USER GUIDE

Syncfusion

DashBoard Platform

Version - v4.1.0.84 | Release Date - July 11,2019

Syncfusion

Overview

The Syncfusion Dashboard Platform is an end-to-end solution for creating, managing and sharing interactive business dashboards. It includes a powerful dashboard server application for easily composing, managing and sharing dashboards.

Key features

Wide variety of Data Sources --- All the commonly used data sources including Microsoft Excel, CSV, XML and JSON or server based data sources like Microsoft SQL Server, MySQL, PostgreSQL, MongoDB, Microsoft Azure Blob Storage or RESTful web services such as JIRA, Twilio, SendGrid, Zendesk are supported.

Business user friendly --- The drag-and-drop friendly dashboard designer application makes it possible for business users to compose and publish dashboards without any help from IT.

Rich selection of widgets --- All the widgets commonly used in business dashboards like a variety of charts, gauges, maps, treemap, heatmap and grids have been included.

HTML5 Powered --- Dashboards are rendered using HTML5 so the only requirement for end users to view dashboards is to have a modern browser installed.

Embed dashboards within your applications --- Dashboards published in Syncfusion Dashboard Server can be embedded within your applications.

Data stays in your servers --- You can host the dashboard server within your local network or on your secure cloud servers so all the data remains secure within your servers.

Cost Effective licensing --- Our licensing is cost effective for both small and large teams. Please contact us for more details.

Create a support incident

If you are still not able to find the information that you are looking for self-help resources mentioned above, please contact us by creating a support ticket.

Quick Start with Syncfusion Dashboard Platform

This quick start guide helps you deploy **Syncfusion Dashboard Server** application in an on-premise server, create a dashboard with your business data, and share it to your team to analyze business insights.

Overview

Syncfusion Dashboard Platform is an end-to-end solution for creating, managing and sharing interactive dashboards. This platform comprises the following products, each of which addresses specific business needs.

Product	Type and used for
Syncfusion Dashboard Server (Windows)	A web application installer to host dashboard server in on-premises web server in Windows environment.
Syncfusion Dashboard Server (Microsoft Azure)	An Azure package to host dashboard server as app-service in Azure environment.

Syncfusion Dashboard Mobile	A mobile app supported in iOS and Android devices to connect dashboard server application hosted in on-premises or in cloud and view dashboards.

Syncfusion Dashboard Server plays a vital role across all these packages. It lets you connect to data, compose dashboards, visualize widgets, organize and share dashboards efficiently with your team, and collaborate. It has a user-friendly editor, called as web designer, which is integrated within Dashboard Server to create and edit dashboards.

Prerequisites

The minimal hardware and software requirements required to deploy and run the Syncfusion Dashboard Server are listed in the following table.

Package	System Requirements	Software Requirements
Syncfusion Dashboard Server (Windows)	• Operating System : Windows Client OS 7+ | Windows Server OS 2008 R2+ • CPU : 1 GHz or faster, 32-bit or 64-bit processor • Memory : 1 GB RAM for 32-bit, 2 GB RAM for 64-bit • Hard drive : 300 MB of free space (only installation files)	• Framework : Microsoft .NET Framework 4.5 • Database : Microsoft SQL Server 2005+ | Azure SQL Database | Oracle | MySQL | PostgreSQL • Web Server : IIS 7.0+ | Microsoft Azure • Web Browser : IE9+ | Microsoft Edge | Mozilla Firefox 22+ | Chrome 17+ | Opera 12+ | Safari 5+ • SMTP Mail Server (Check Email Settings for more detail)

Note: If the web browser is IE11, please ensure the following points:

* Disable the Enterprise mode if it is found enabled. For more details, refer here.

* Turn off the Compatibility Settings for the intranet sites. For more details, refer here.

Deploy Syncfusion Dashboard Server

Download the package

For evaluation purpose, download the required package you need from the Downloads page.

The licensed users can download the package from this page.

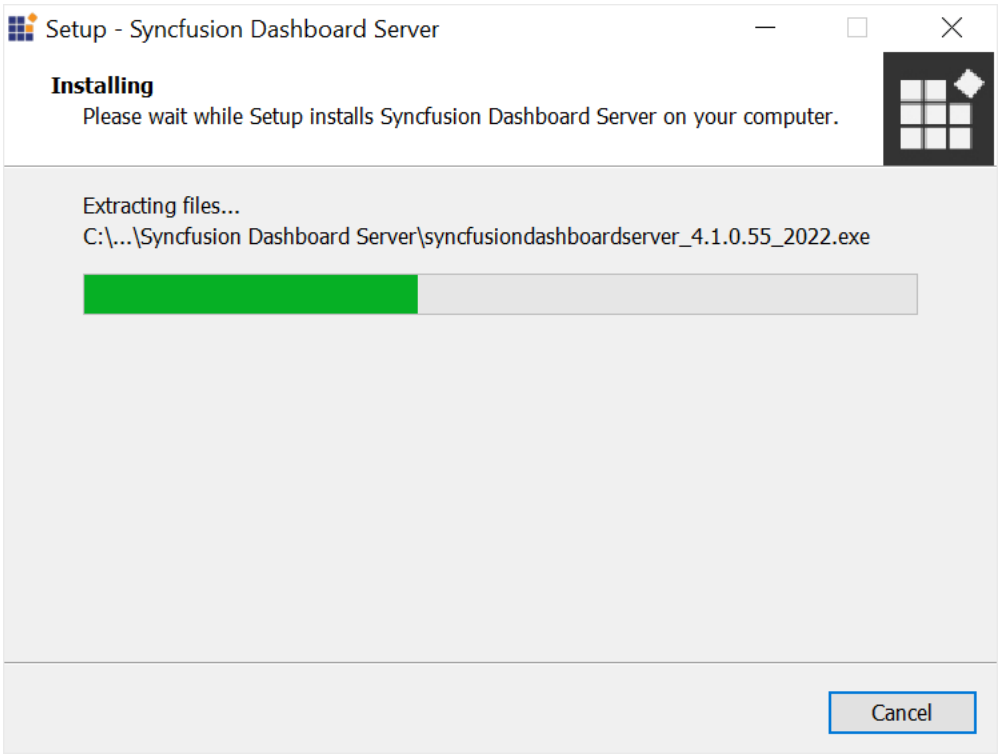
Downloads will be available in the following two formats: **EXE** and **ZIP**. You can choose any of them. If you choose ZIP, you have to extract the zipped folder to get the EXE.

Note: Mobile app can be downloaded directly from Google Play Store for Android devices and App Store for iOS devices.

Install the package

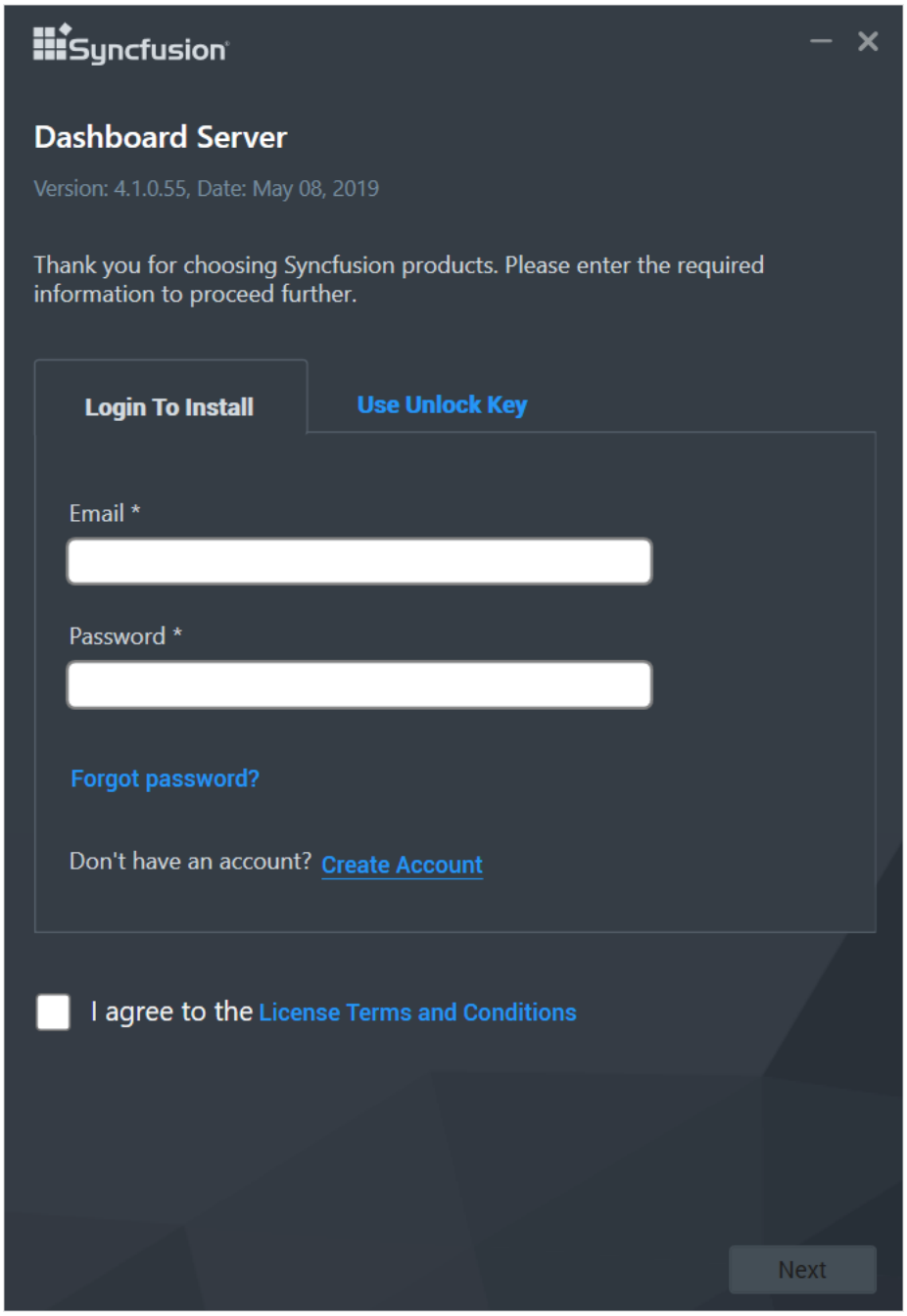
Step 1:

Place the installer EXE in a physical location in the server machine (On-premises or Cloud VM), and double-click to install. Now, the file extraction wizard launches. On its completion, installer wizard opens.



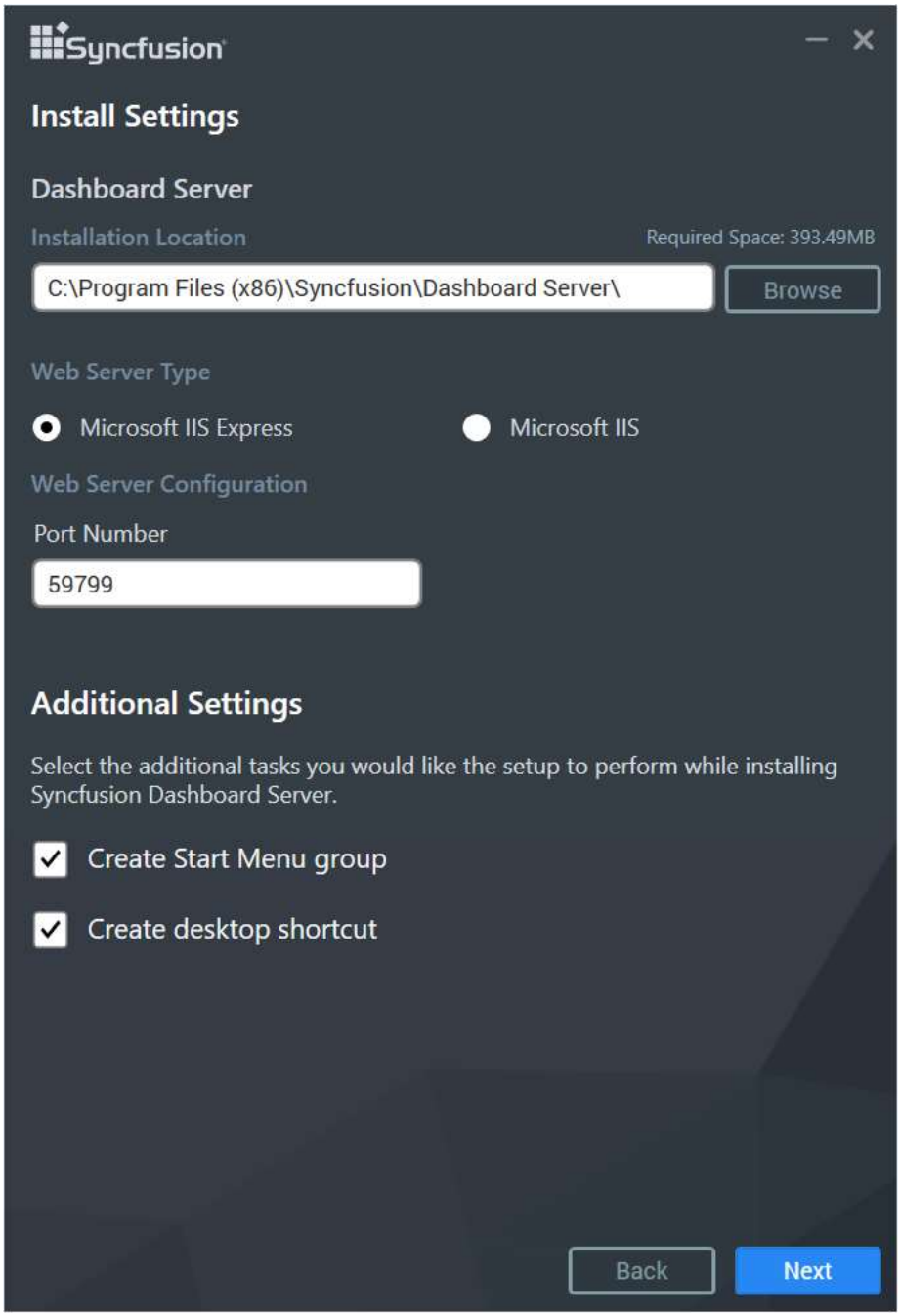
Step 2:

Enter your Syncfusion account credentials to unlock the Setup. Alternatively, you can use the unlock key sent to your mail once you downloaded.



Step 3:

Check and accept license agreement terms and conditions and click **Next**.



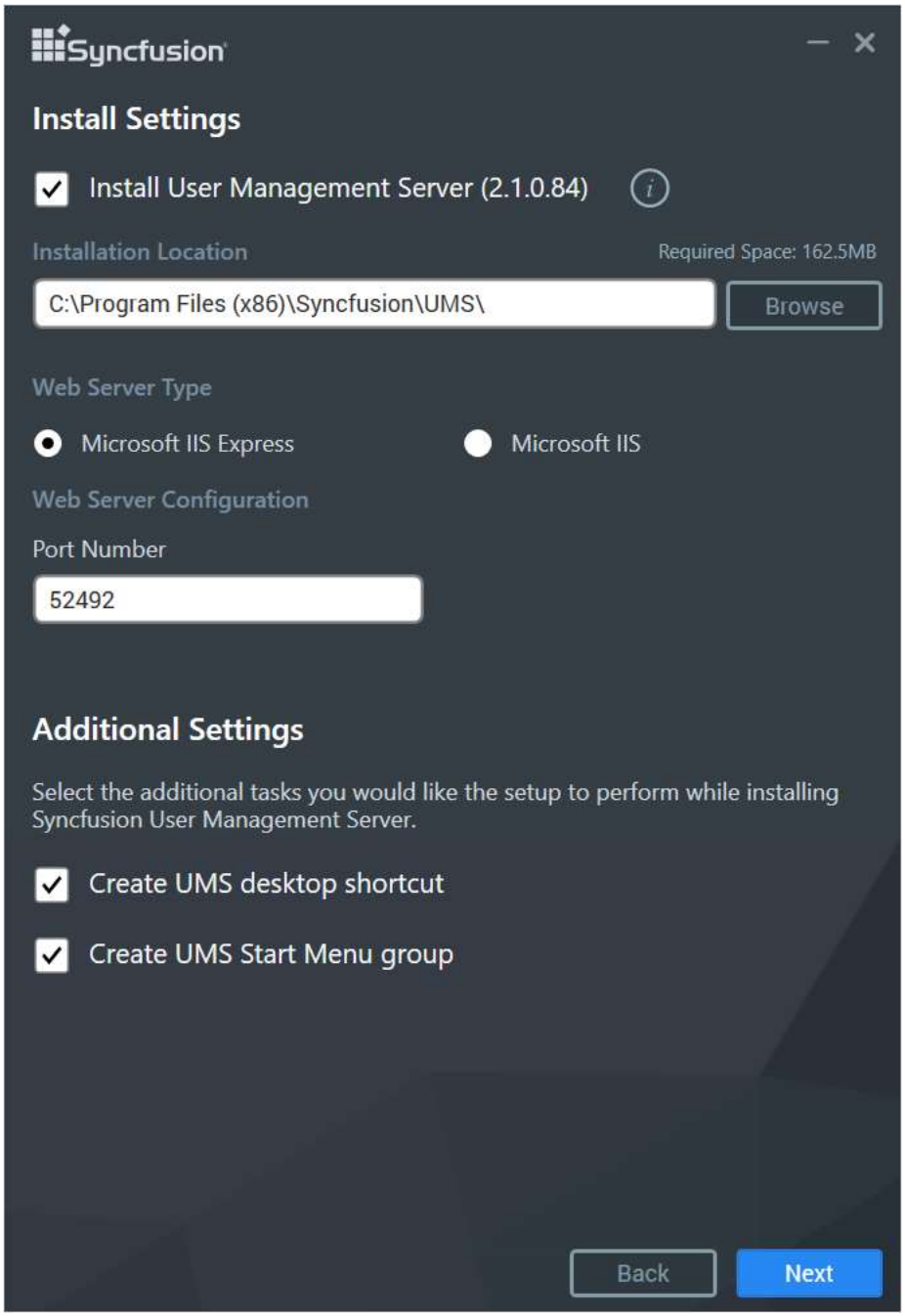
Step 4 (optional):

Configure the installation location by replacing the default location, if needed.

Step 5:

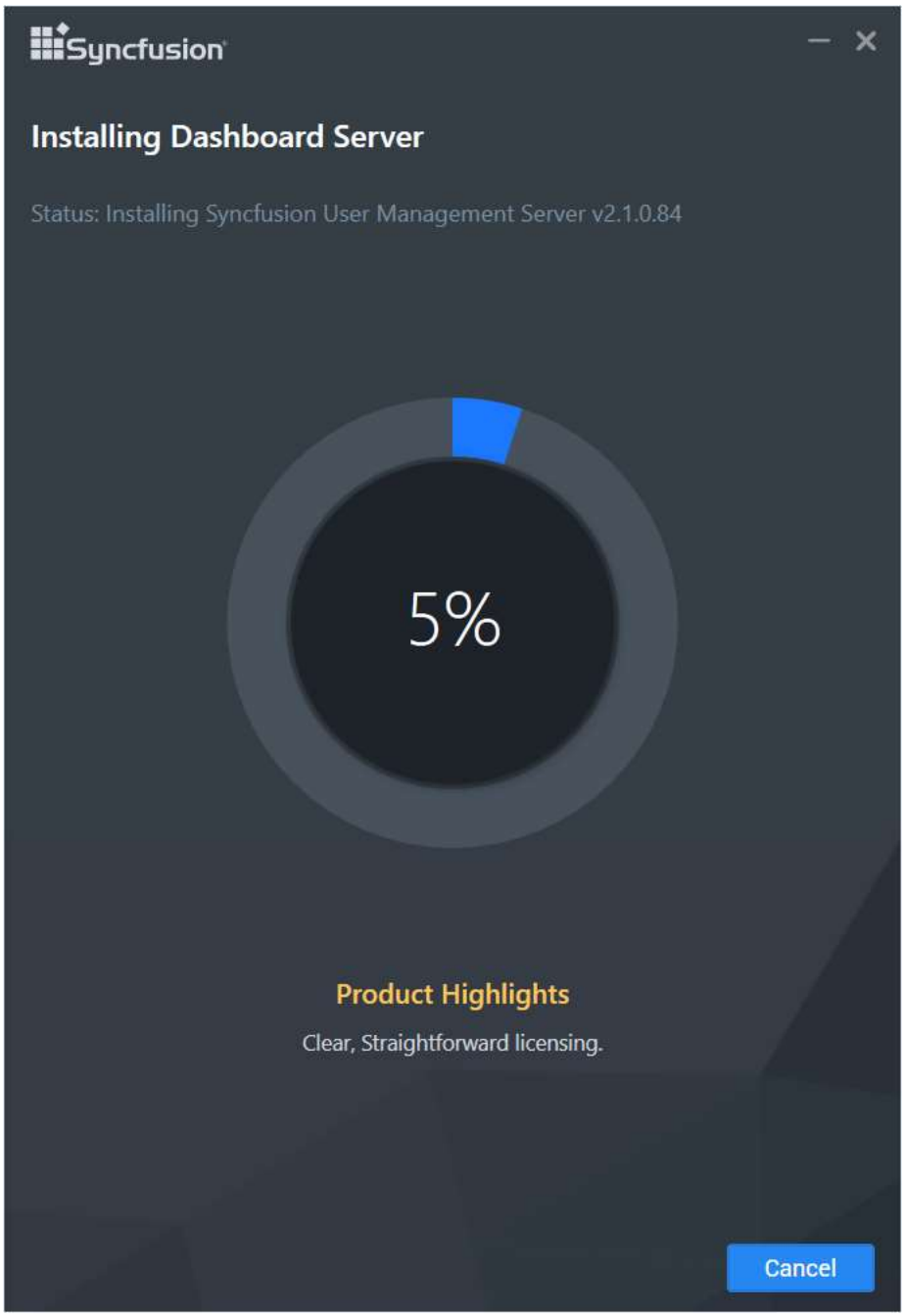
Select the web server (IIS or IIS Express), where the Dashboard Server and User Management Server applications need to be hosted.

Note: If you have User Management Server application installed already here or in a different machine (accessible from this one), you can skip its installation alone by clearing the check box selection near **Install User Management Server (x.x.x.x)** and then choosing **Next**.



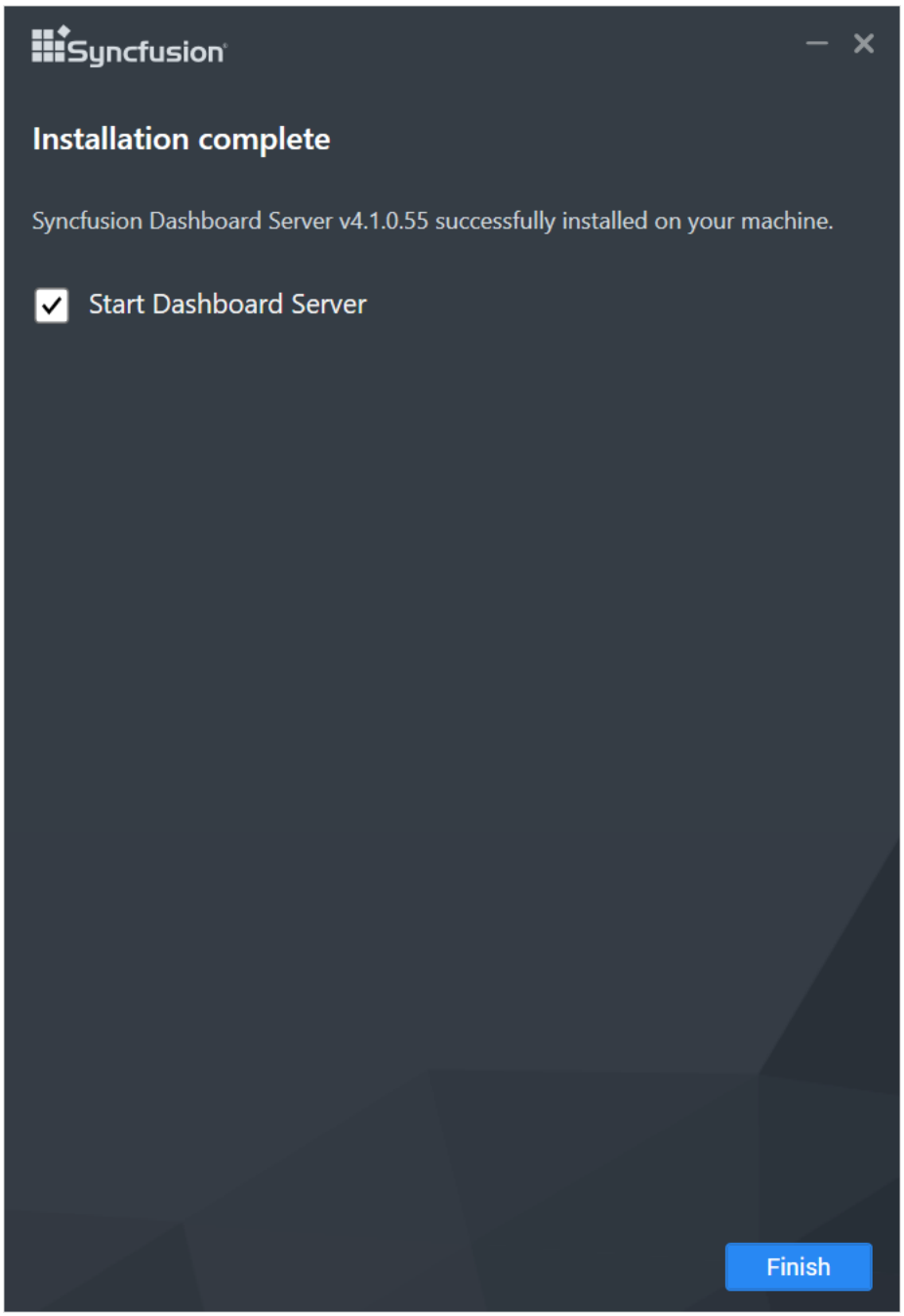
Step 6:

Configure the port number by replacing the default port number, if needed, and then click **Next**. Now, the installation begins.



Step 7:

Finally, the installation completes with the following screenshot. To immediately launch the dashboard server configuration screen in web browser on closing the wizard, select the check box in the wizard screen, and then click **Finish** to close the wizard.

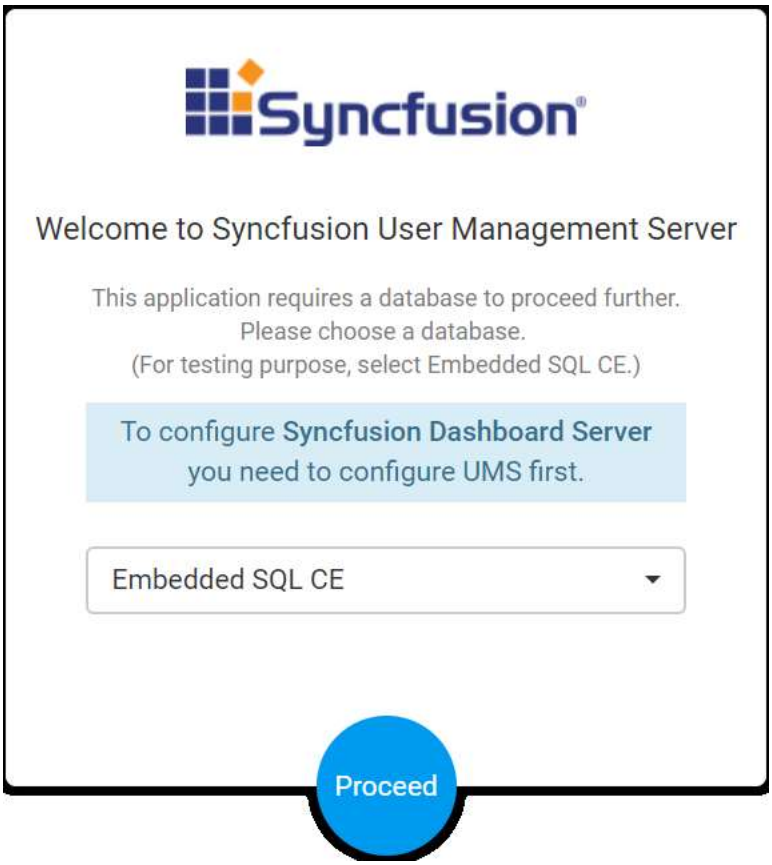


You can also launch the dashboard server later through the program shortcut added in the desktop during installation.

Configure server applications

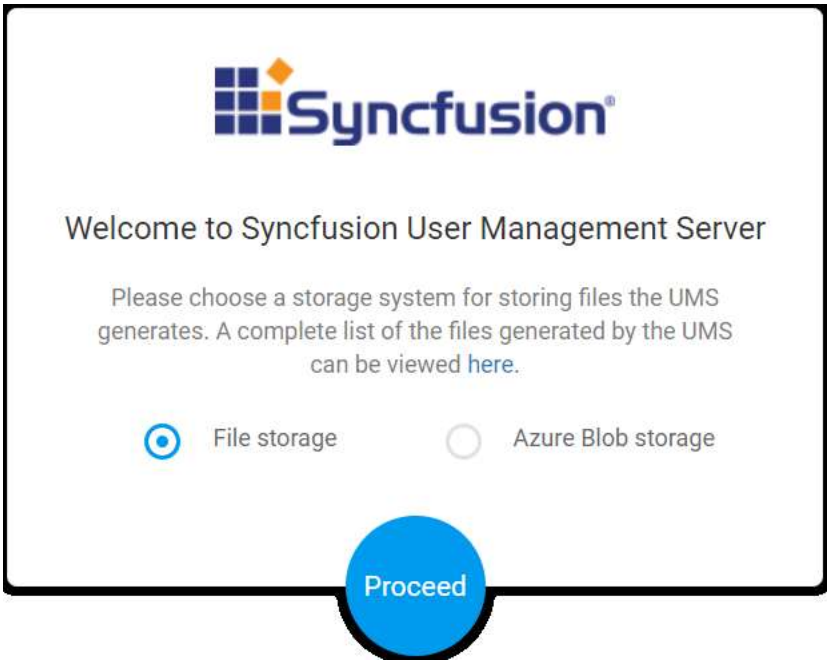
Step 1:

In the dashboard server application launched, configure the database to manage the users, and click **Proceed**.



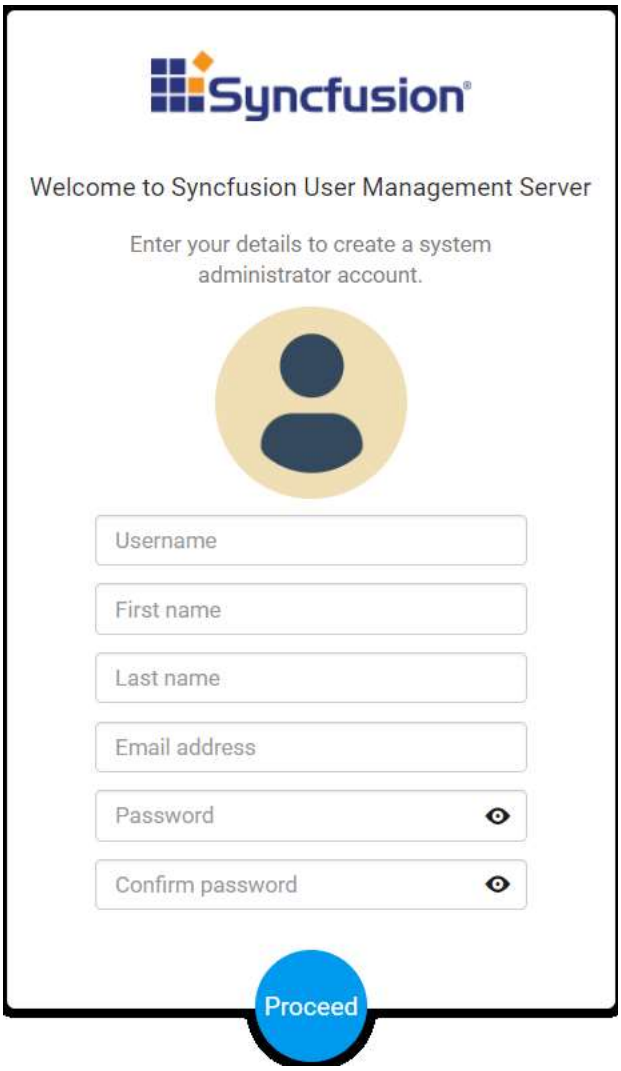
Step 2:

Configure the storage medium (files or blob) to store the resources of User Management Server, and click **Proceed**.



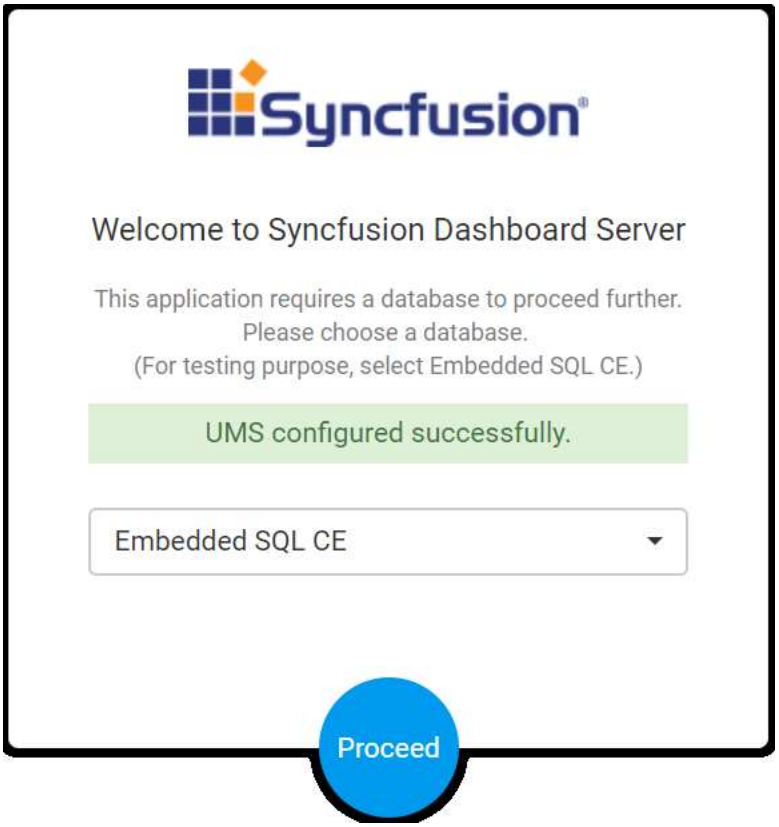
Step 3:

Configure the user information and credential details to create a system administrator account in the User Management Server, and click **Proceed**. Now the user management server configuration is completed.



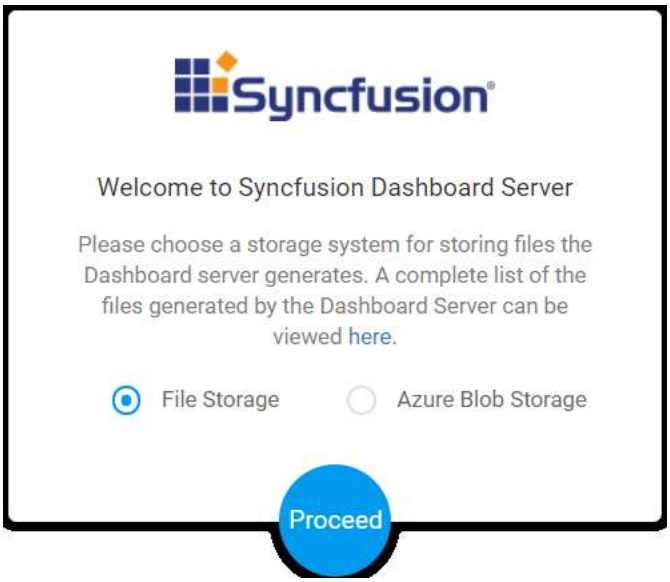
Step 4:

Configure the database to manage the dashboard server resources, and click **Proceed**.



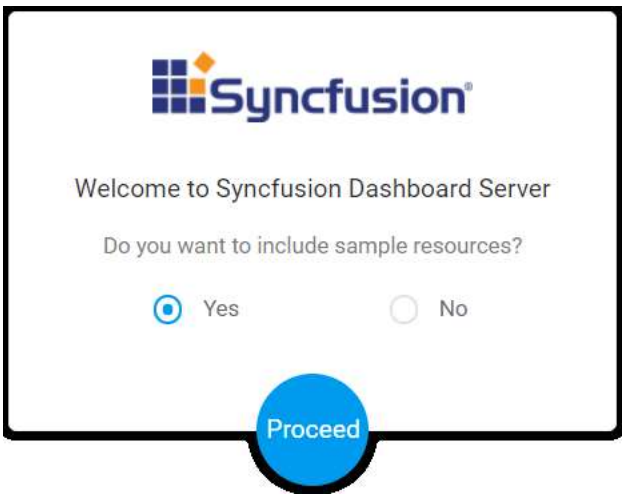
Step 5:

Configure the storage medium (files or blob) to store the resources of dashboard server, and click **Proceed**.

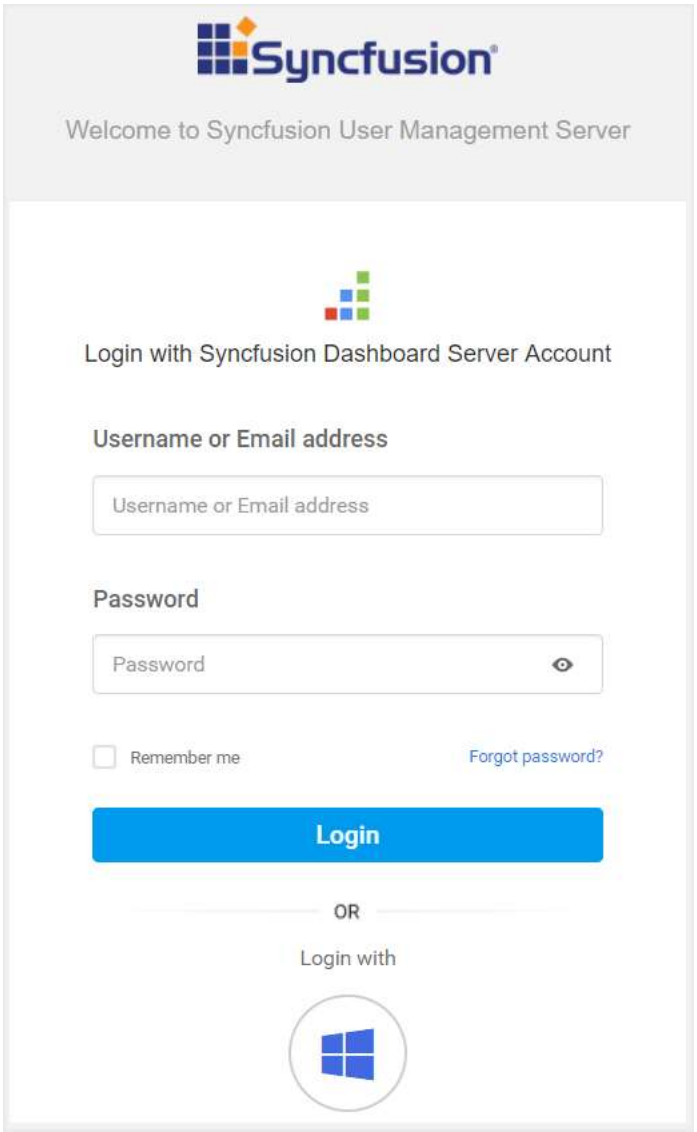


Step 6:

Choose **Yes** to include the sample resources, and click **Proceed**. This includes the sample dashboards and data sources into the server that comes as a part of the installer.

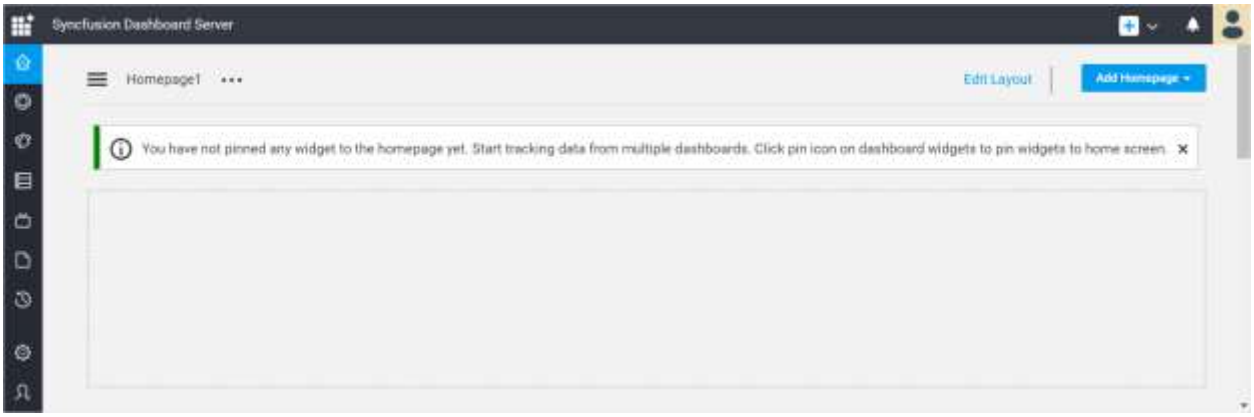


On successful configuration, you will be redirected to the login page.



Step 7:

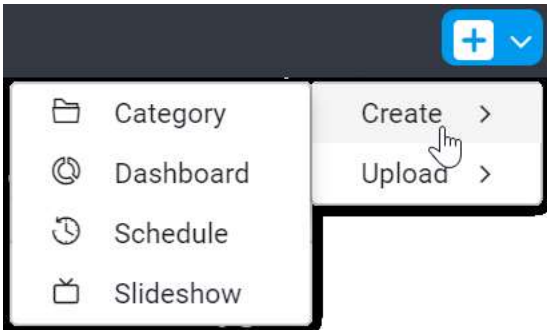
Enter the username or email and password to log on to the dashboard server. Now the dashboard server home page opens.



Create your first dashboard

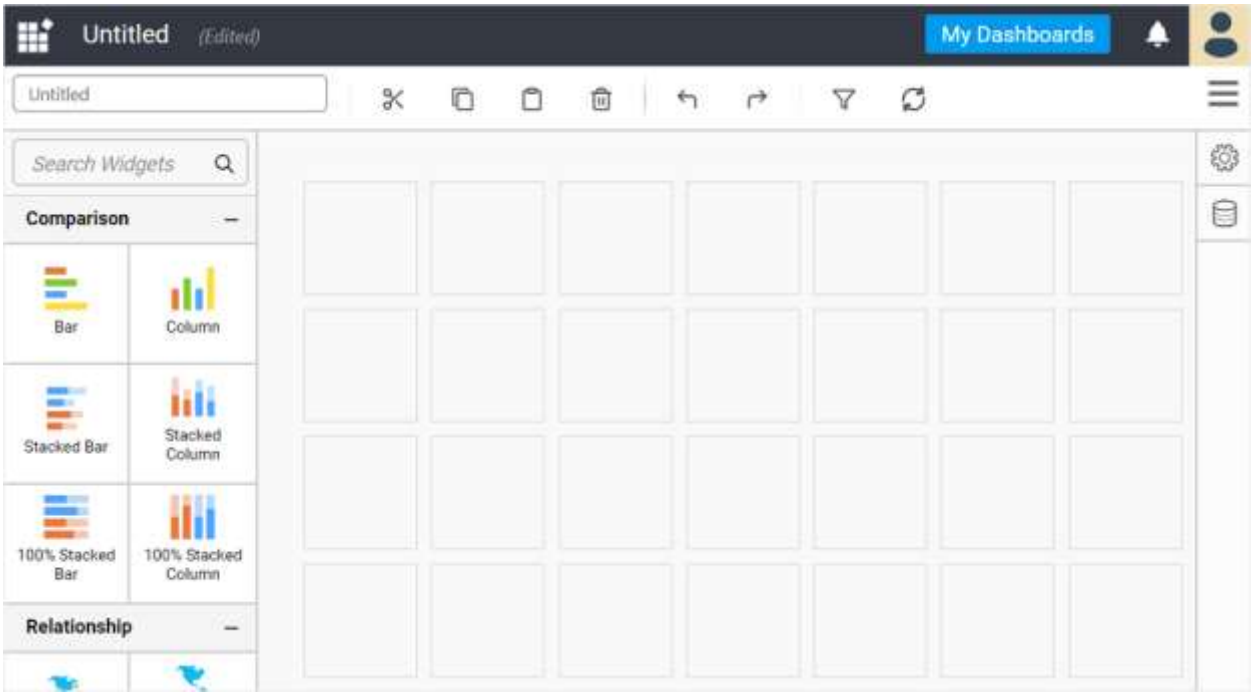
Step 1:

In the homepage, navigate to the top-right corner, and click the down arrow button next to the **+** icon to open a drop-down list.



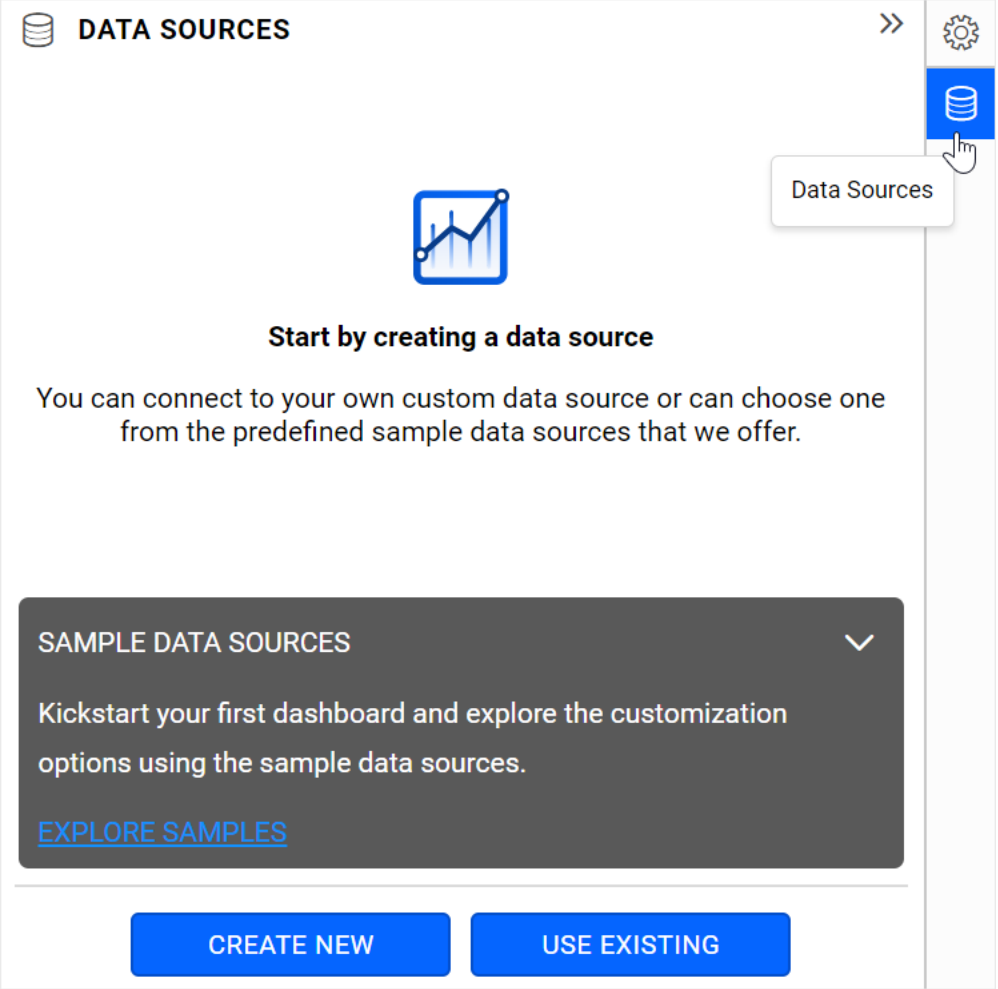
Step 2:

Hover the mouse over the **Create >** menu item to open the submenu, and choose **Dashboard** in it. Now a blank dashboard opens.



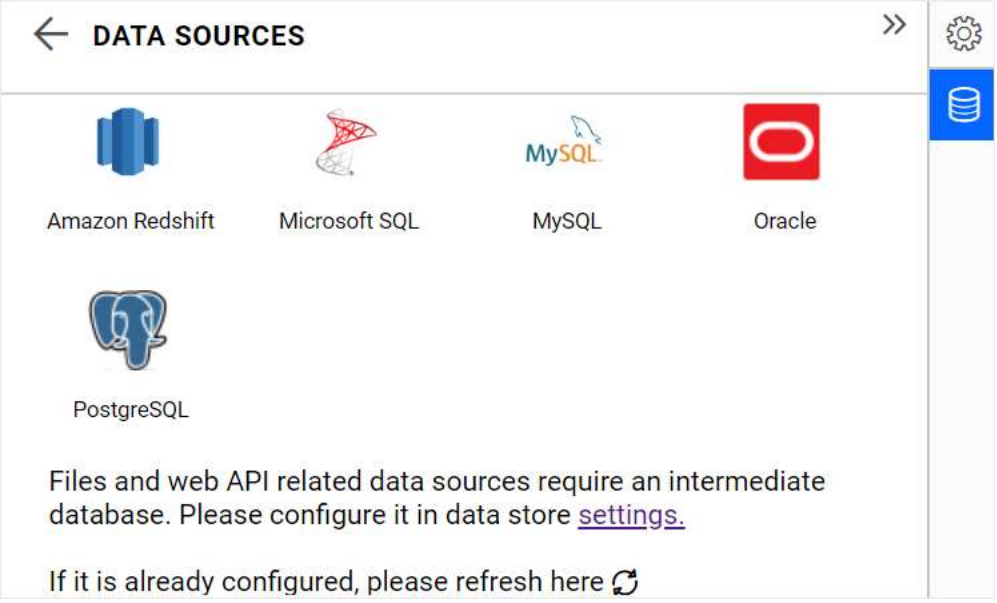
Step 3:

To create a data source for widgets configuration, click the database icon at the right side panel next to the gear icon. Now, the **DATA SOURCES** panel opens.



Step 4:

Select **CREATE NEW** to open the data connections listing panel.



Step 5:

Choose the connection type that matches your data store and configure the database and related details.

Note: For files and web services type connection, you need to configure a Microsoft SQL Server database as intermediate data store for data extraction and processing. To do this,

1. Click My Dashboards at the top to navigate to server page.

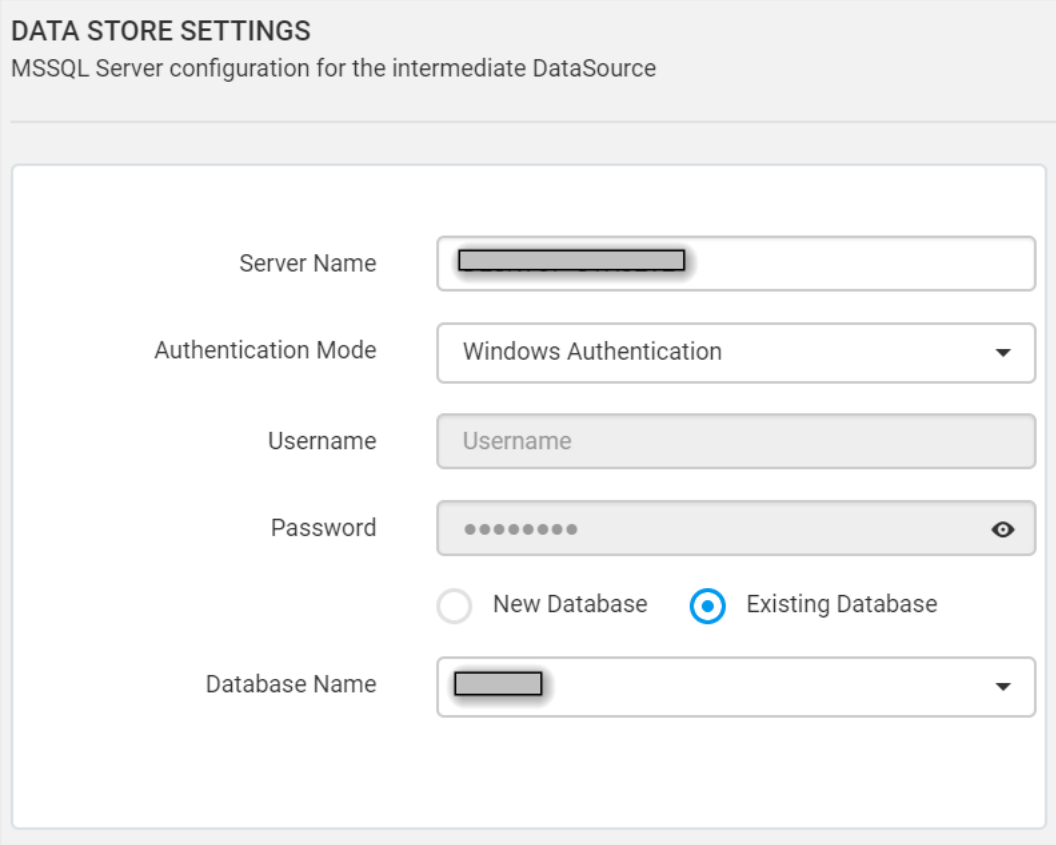
2. Choose the gear icon (Settings) at the bottom of the left side panel. This opens the Site Settings page.

3. Navigate to the Data Store tab, configure the SQL Server database and server detail, and click Save.

4. Click the **Refresh** icon in the **DATA SOURCES** panel in dashboard editor to reflect the supported files and web services in the listing.

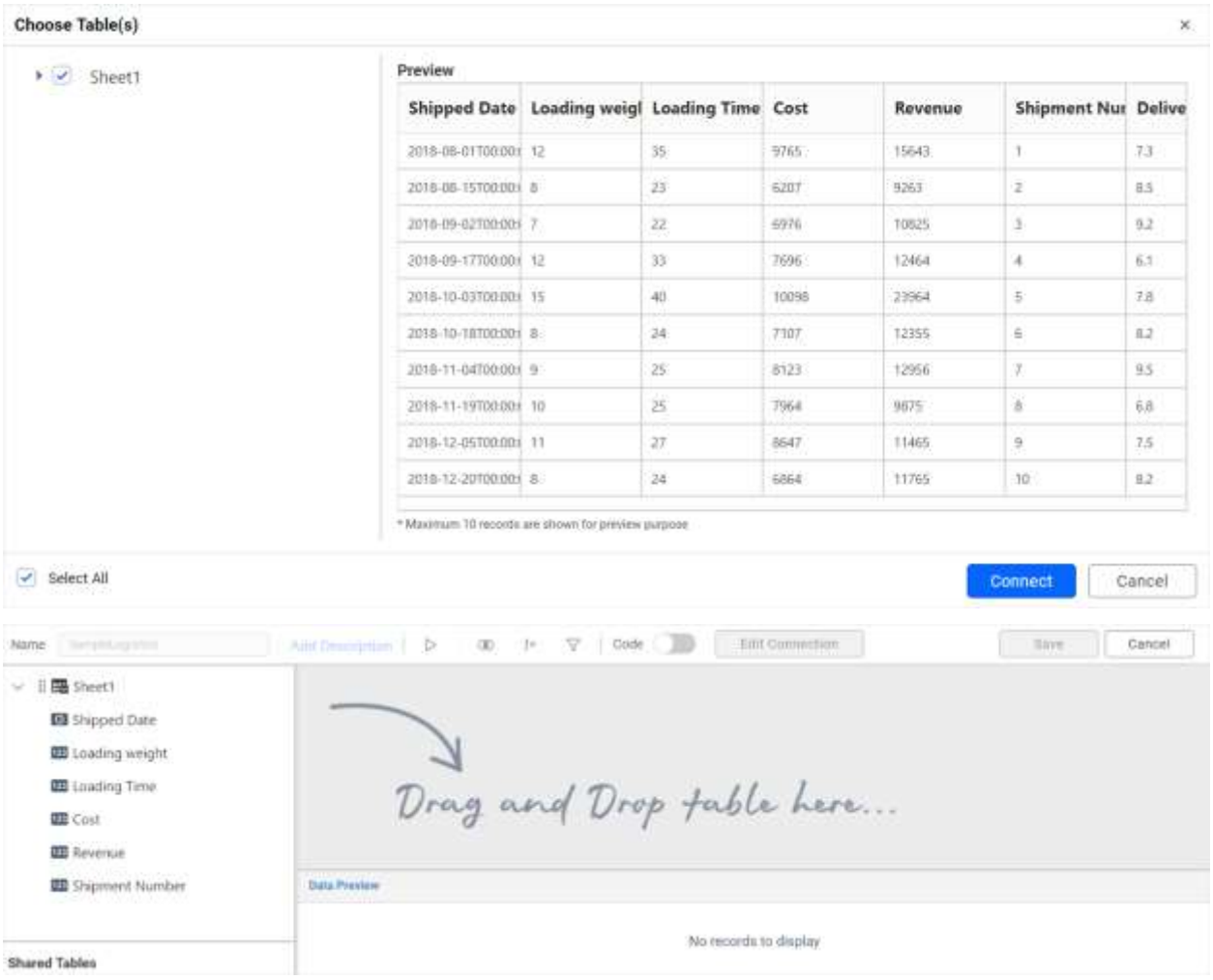
You can also navigate to this page directly by clicking the Settings link at the bottom of the data connections listing panel.

Refer to here for the data connections listing categorized by direct and in-memory modes.



Step 6:

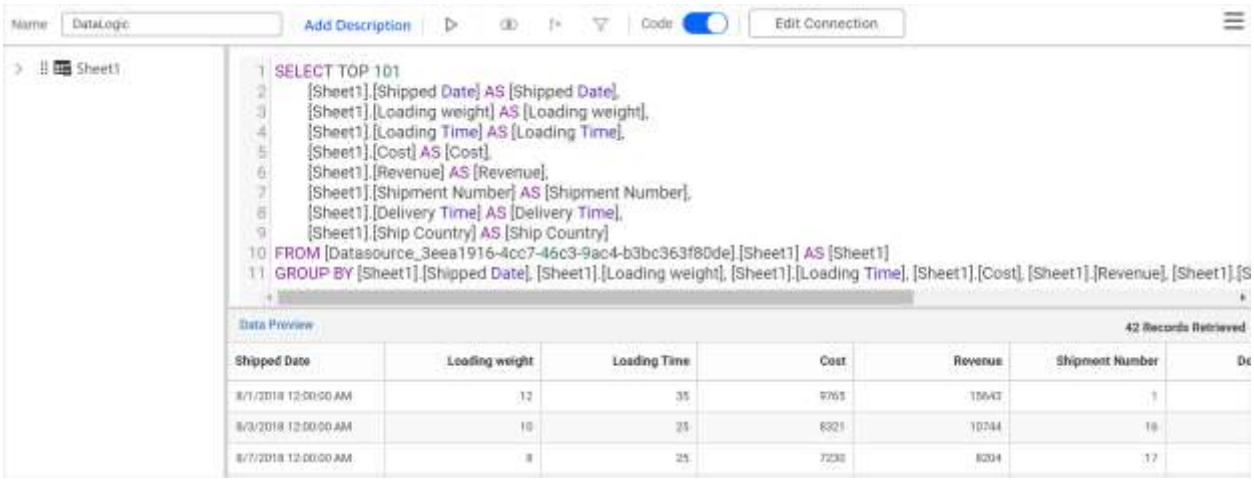
Choose the required tables and select **Connect**. Now, the data source design view window opens with the selected tables listed in tree view panel at the left.



Step 7:

Drag the required table(s) from the tree view panel to the table design view. If you drop more than one table, join editor opens to define the relationship between tables. This editor can also be opened explicitly from the toolbar at top.

Alternatively, you can switch to the code view using the toggle button in toolbar. Paste your select query and execute to fetch the data for data source preparation.



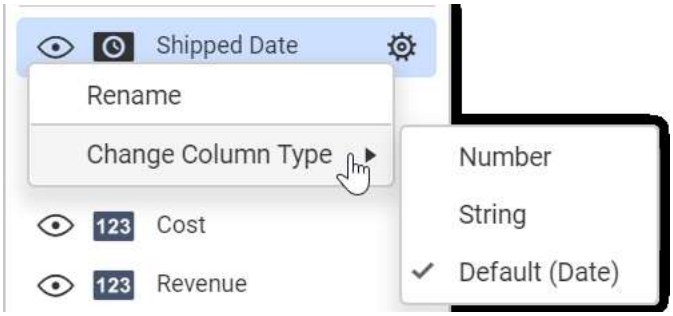
Note: * Switching from design view to code view will retain the current state in design view unless you execute the query through the Play button in toolbar in code view.

* Switching from code view to design view is allowed only when you wipe out the query in the code view window.

* Code view supports native queries for direct mode connections. For in-memory mode connections, Microsoft SQL Server queries alone were supported.

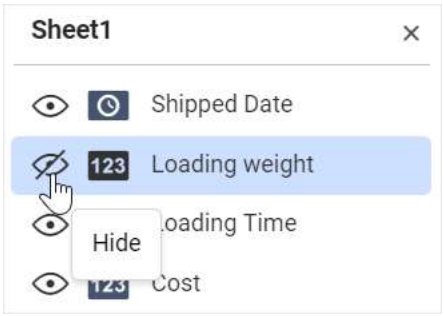
Step 8 (optional):

To rename a column and change its type, click the gear icon near to it in the dropped table and select corresponding menu items.



Step 9 (optional):

To exclude a column from consideration, select the eye icon near to that column in the dropped table to disable.

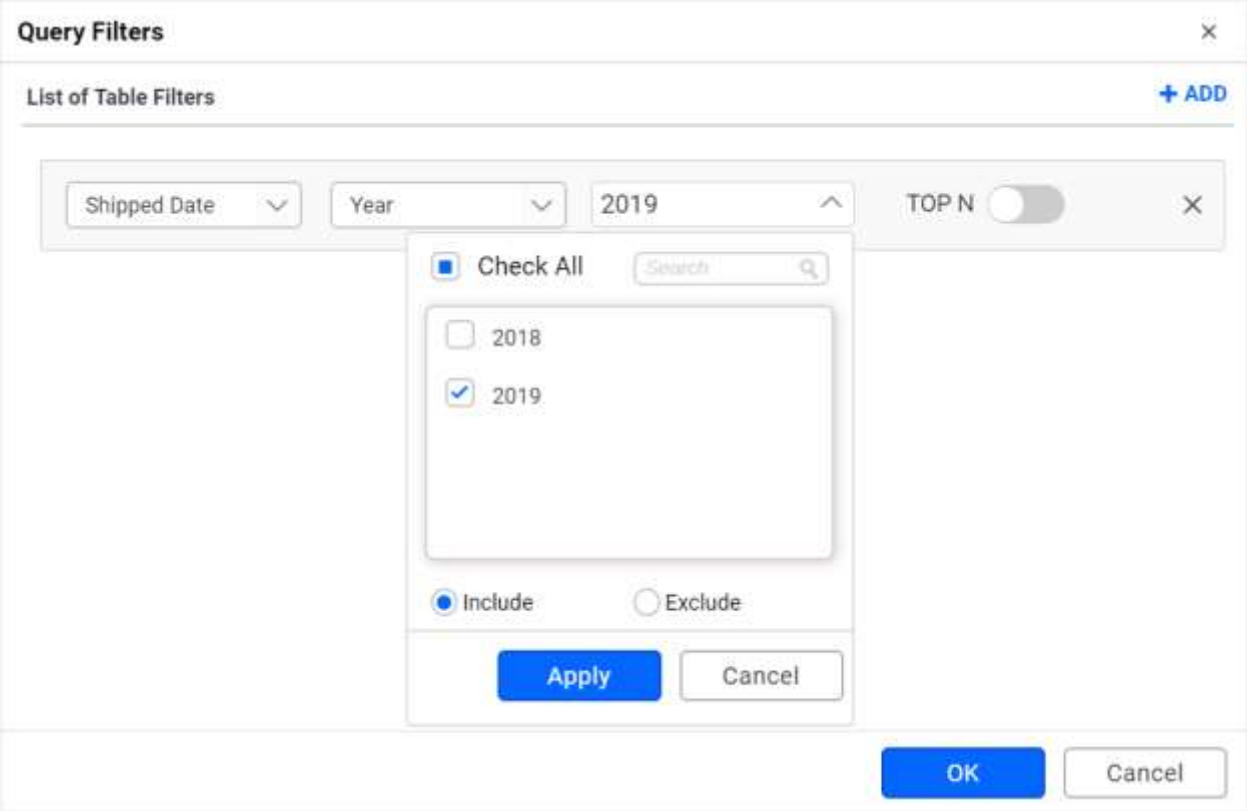


Step 10 (optional):

To create calculated columns, open the expression designer from the toolbar menu item and create columns with or without expression functions.

Step 11 (optional):

To restrict the incoming data based on some filter criteria, open the Filter window from the toolbar menu item and create filter conditions.

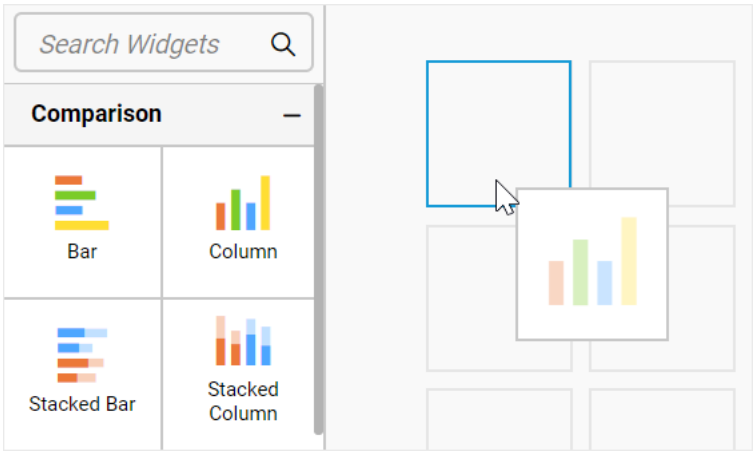


Step 12:

Define a name and a suitable description (optional) for the data source and click **Save** in the toolbar.

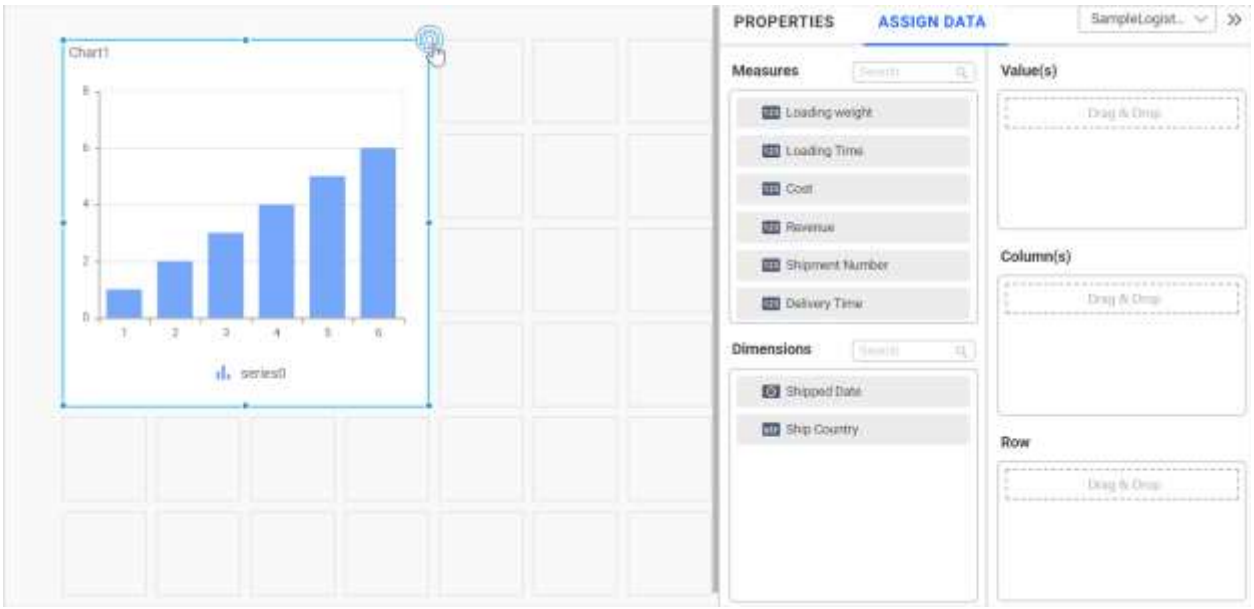
Step 13:

Drag the required widgets from toolbox to the designer surface and resize as needed.



Step 14:

To configure data to a widget, select that widget and click the gear icon at the top-right corner of the focused widget. This opens the **PROPERTIES** panel. Switch to the **ASSIGN DATA** tab.



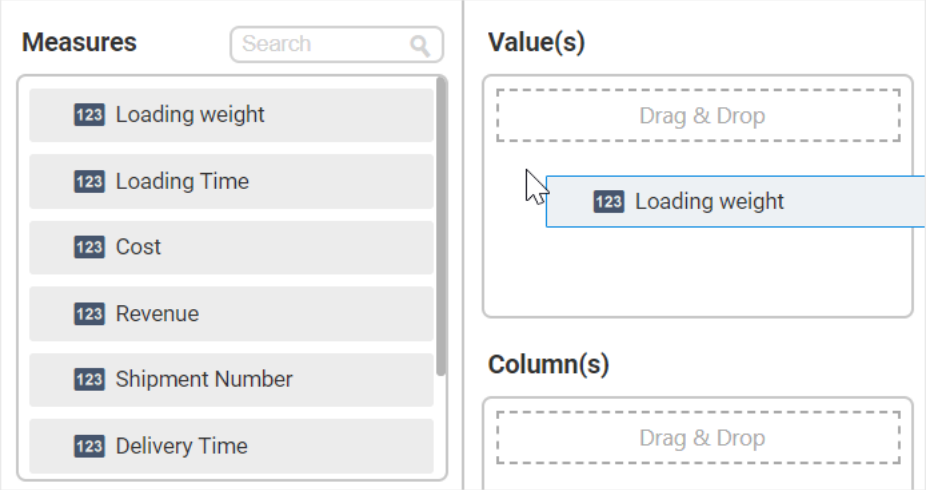
Step 15:

Choose data source from the drop-down list at the top to show its columns for widget configuration.

Step 16:

Configure the appropriate columns in corresponding sections.

Note: The sections change based on the widget type and kind.



Step 17 (optional):

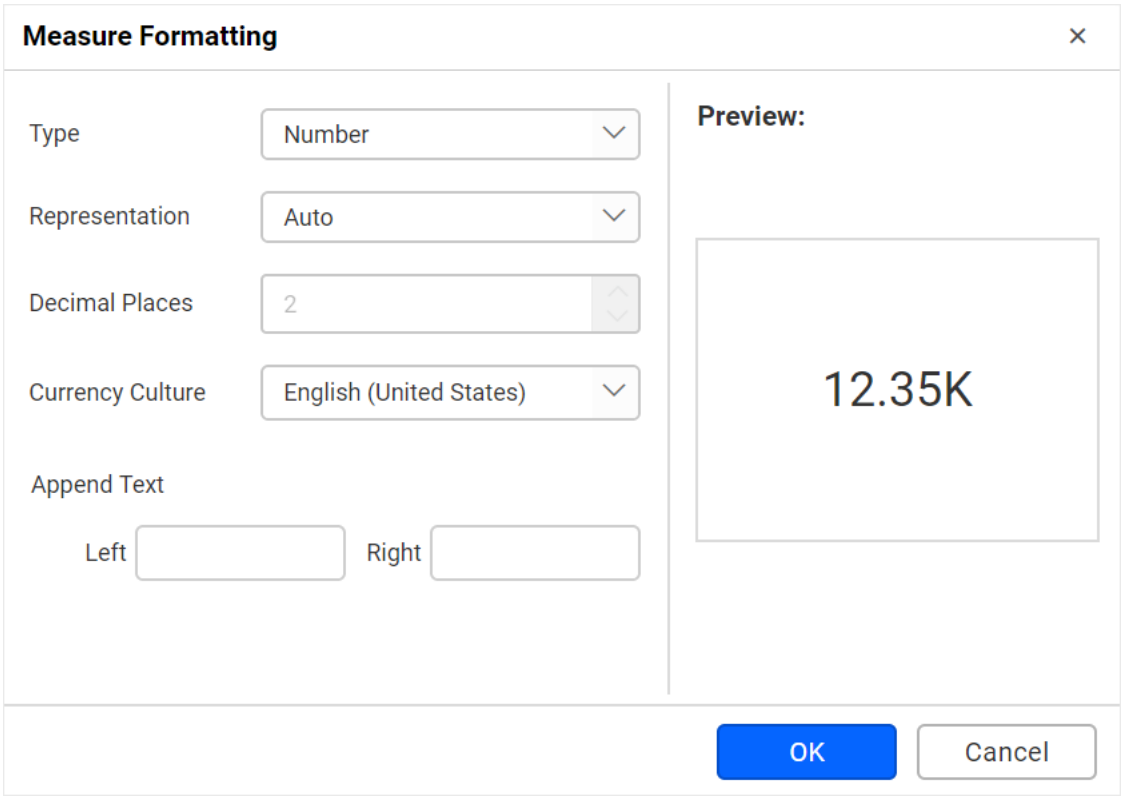
In the configured columns, based on its type (measure or dimension) and the section placed, the following operations can be handled as needed by clicking the gear icon near each column.

Changing aggregation

Filtering data at widget level

Sorting data items

Formatting data display



Step 18:

Configure other required widgets in similar fashion, and click **Preview** at the title bar to visualize the configured widgets in the dashboard. After configured, click **Close Preview** at the top to close the preview window.

Step 19:

Click **Publish** next to the **Preview** button in the title bar to publish the dashboard to the server view part. Now the **Publish As Dashboard** window opens.

Step 20:

Choose a **Category** to publish and a **Name** for the dashboard with optional **Description**.

Note: If you do not have any categories listed, create a new category by pressing the **+** button near the **Category** field.

Step 21:

Define the Privacy Settings to either *Private*, *Public* or *Unlisted* for the dashboard based on the level of its access you need to give others. Click **Publish**.

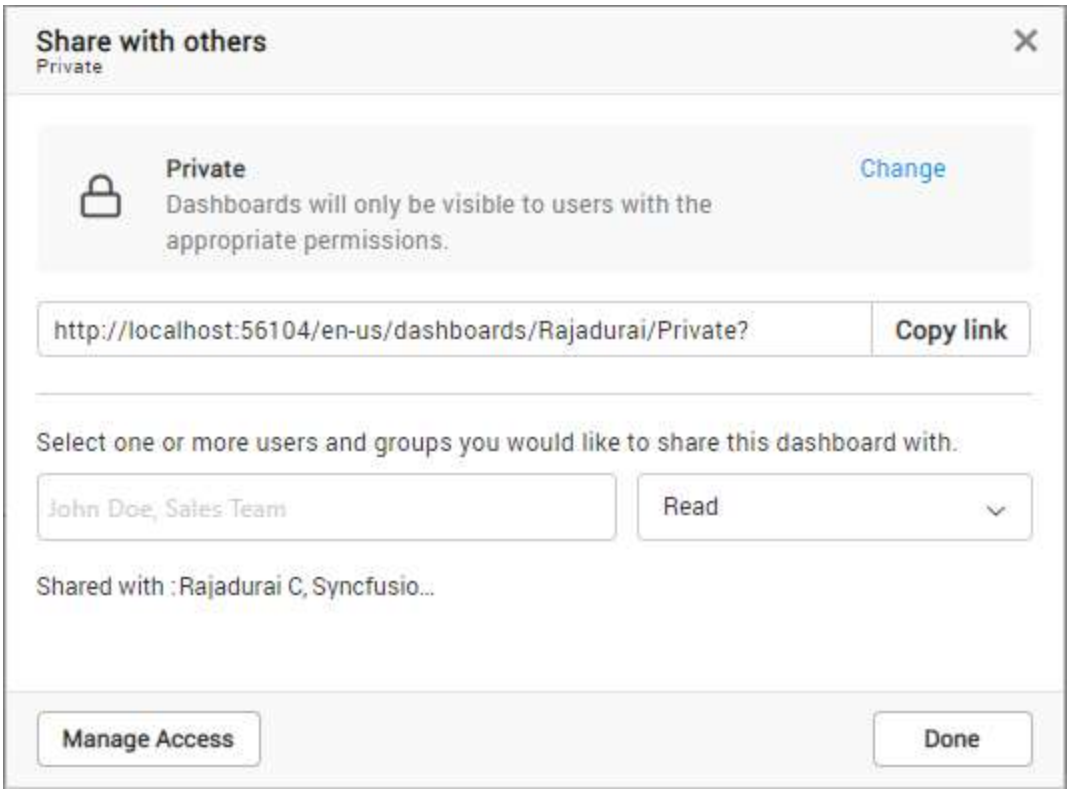
Step 22:

Click **Yes** in the **Confirm Publish Dashboard** message window. Your dashboard will be saved, listed in the **Dashboards** listing in the Server view, and opened.

Share your dashboard

Step 1:

Click **Share** at the top of your dashboard. Now **Share with others** dialog opens.



Step 2:

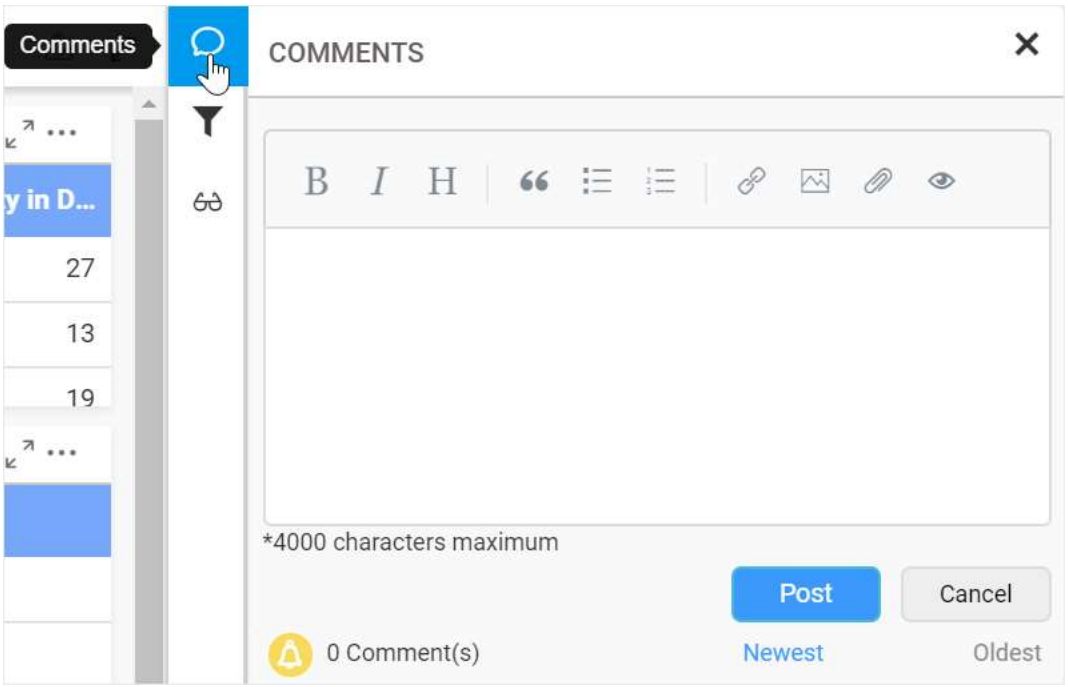
From the auto-complete dropdownlist, select one or more users and groups you would like to share this dashboard with.

Step 3:

Choose the access level (Read or Read, Download or Read, Write, Download or Read, Write, Delete, Download) to be given for the selected users and groups and click **Done**. Now, your dashboard will be shared.

Step 4:

Select the **Comment** icon at the top-right corner (below the user avatar), and start collaborate with others who were given access to your dashboard. For every comment, respective users will get an email.



Next steps

Now, you have learned how to start with Syncfusion Dashboard Platform from creating a new dashboard to a meaningful visualization tool that reveals insights over your business data for you and your team to make better decisions.

Resources

These self-help resources will assist you in getting quick answers for your queries and better understanding of concepts and functionalities supported in Syncfusion Dashboard Platform.

Sample Demos

The different sets of dashboards are designed and published in the demo server website for dashboard platform users to demonstrate various features in the Syncfusion Dashboard Platform. You can view the dashboards from the following link.

https://dashboardserver.syncfusion.com/

Tutorial videos

Short video tutorials help you explore the Syncfusion Dashboard Platform quickly. These can be viewed from the following link:

https://www.syncfusion.com/products/dashboard/videos

Frequently asked questions

Common queries in Dashboard Platform were answered with required explanation and illustrations and organized as Knowledge Base articles. Refer to the following link for KB articles:

https://www.syncfusion.com/kb/dashboard

Also, queries raised by different users along with suggested solutions in Dashboard Platform by Syncfusion team can be viewed from the following link:

https://www.syncfusion.com/forums/dashboard

Technical Blogs

Blogs on topics related to remarkable functionalities, overview of features in a public release, trending features coming sooner in Dashboard Platform can be found from the following link:

https://blog.syncfusion.com/

Road map

Refer to the following link to learn about the road map plan of the upcoming dashboard platform:

https://www.syncfusion.com/products/roadmap/dashboard

Release history

The release information of all the public versions of the Dashboard platform can be referred in the following link.

https://www.syncfusion.com/products/release-history/dashboard

Whats new

The following page holds the features releases in the latest public version of the Dashboard platform:

https://www.syncfusion.com/products/whatsnew/dashboard

Create a support ticket

If you have questions specific to Dashboard Platform and you have not found answers throughout the resources, you can raise a support ticket to Syncfusion. To create a support, refer to the following link:

https://www.syncfusion.com/support/directtrac/incidents/newincident

While creating a support ticket, try to provide detailed explanation of your requirement with appropriate **screenshots** or **videos**. For reporting any bug, consider providing the exact **replication procedure** and necessary **log files**. It will help our support team to reach you earlier with a better solution.

Submit your feedback

Your feedback is valuable to us. Please submit your suggestions or comments for product enhancements in future releases. To provide feedback, refer to the following link:

https://www.syncfusion.com/feedback/dashboard-platform

Dashboard Server

REST API

Using the Syncfusion Dashboard Server REST API, you can manage and change Syncfusion Dashboard Server resources programmatically via HTTP. The API gives you simple access to the functionality behind the resources on a Syncfusion Dashboard Server. You can use this access to create your own custom applications or to script interactions with Syncfusion Dashboard Server resources.

API Versions

Rest API-v1.0	You can manage(retrieve, add, update and delete) groups and users in Syncfusion Dashboard Server using the Rest API.
Rest API-v2.0	You can manage(retrieve, add, update, delete and export) items and permissions, also can perform special operations of groups and users in Syncfusion Dashboard Server using the Rest API.

Rest API-v3.0	You can manage(add and update) files in Syncfusion Dashboard Server using the Rest API.
Rest API-v4.0	You can manage(retrieve, add, update, delete) items and schedule(add,update) in the Syncfusion Dashboard Server using the REST API

REST API Sample

REST API sample client application shows how to invoke the Syncfusion Dashboard Server API programmatically from your own application. You can refer the sample application directly into your application and call the helper methods to make use of it. Please download the sample application here.

Overview

The Syncfusion Dashboard Server lets you efficiently organize and share Dashboards through a web interface. Check the below list of key features in the Syncfusion Dashboard Server.

Key features

Dashboard management --- Dashboards are efficiently organized under the categories. Permission to view the Dashboards can be given to specific users or groups.

Designer integration --- Seamlessly design and publish Dashboards from within the Syncfusion Dashboard Designer application.

Versioning --- All items stored in the Dashboard server are versioned, so it is possible to revert to an older version.

User management --- Users can be easily organized into groups to accurately map the structure of the small and large organizations.

Scheduling --- Dashboards can be exported into an image file and emailed according to a schedule. The scheduling functionality is very flexible.

Flexible permissions --- A flexible permission scheme controls the access to read, write, delete and download Dashboards.

View Dashboards --- The built-in HTML 5 Dashboard Viewer control lets the user to view Dashboards from within the browser.

Export --- Dashboards can be exported to image file formats.

Custom branding --- The Dashboard server has built-in customization capabilities such as allowing you to add your organization's name, logo, welcome note, etc.

Create a support incident

If you are still not able to find the information that you are looking for in the self-help resources mentioned above then please contact us by creating a support ticket.

Setup

Installation and Deployment

This section explains on how to install and deploy the Syncfusion Dashboard Server version 3.1 or lesser.

Download Setup

- You can download the Dashboard Server setup from here

- Licensed customers can download the install from the downloads section

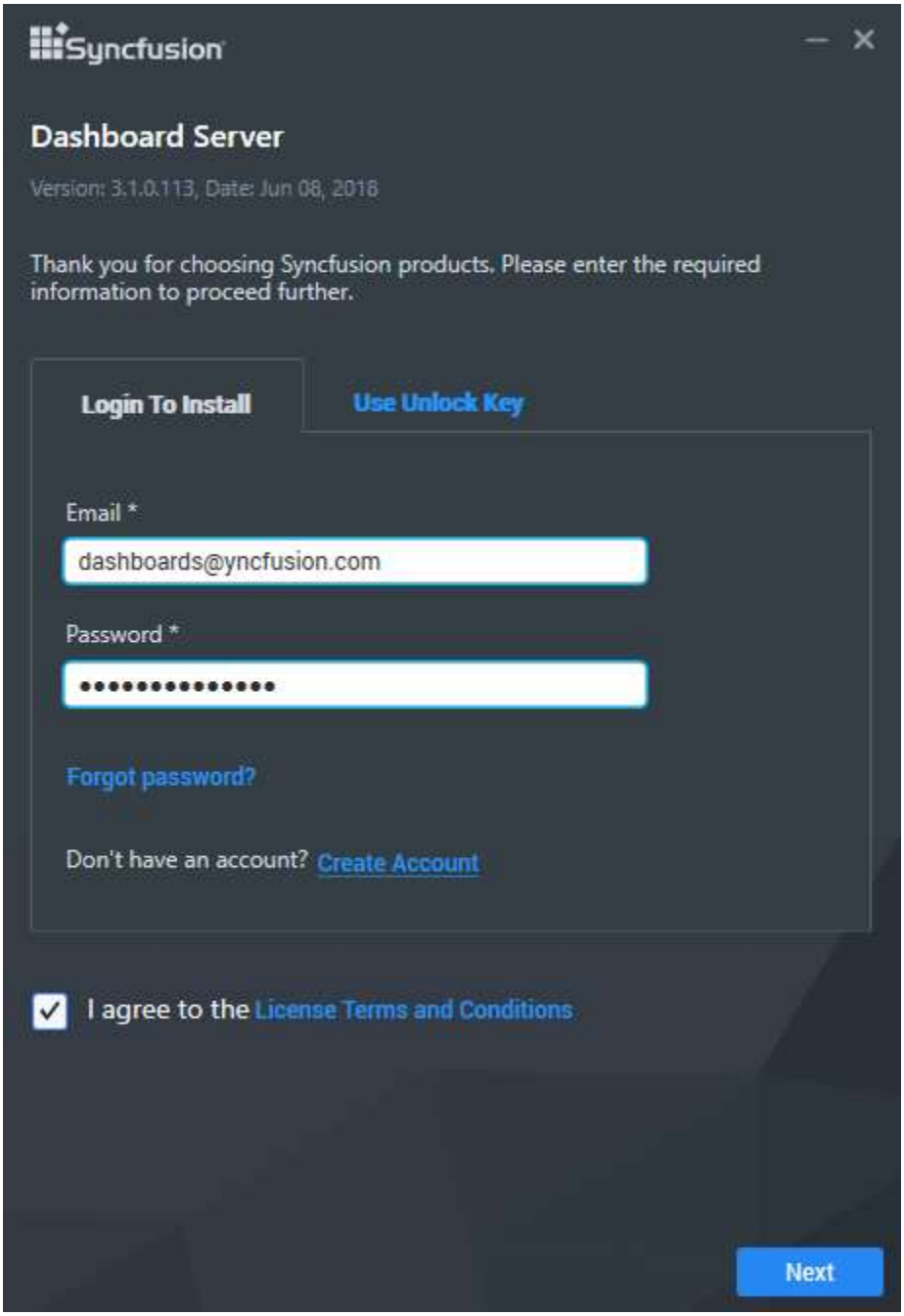
Note: The key to unlock the setup will be sent to your registered e-mail address.

Installation

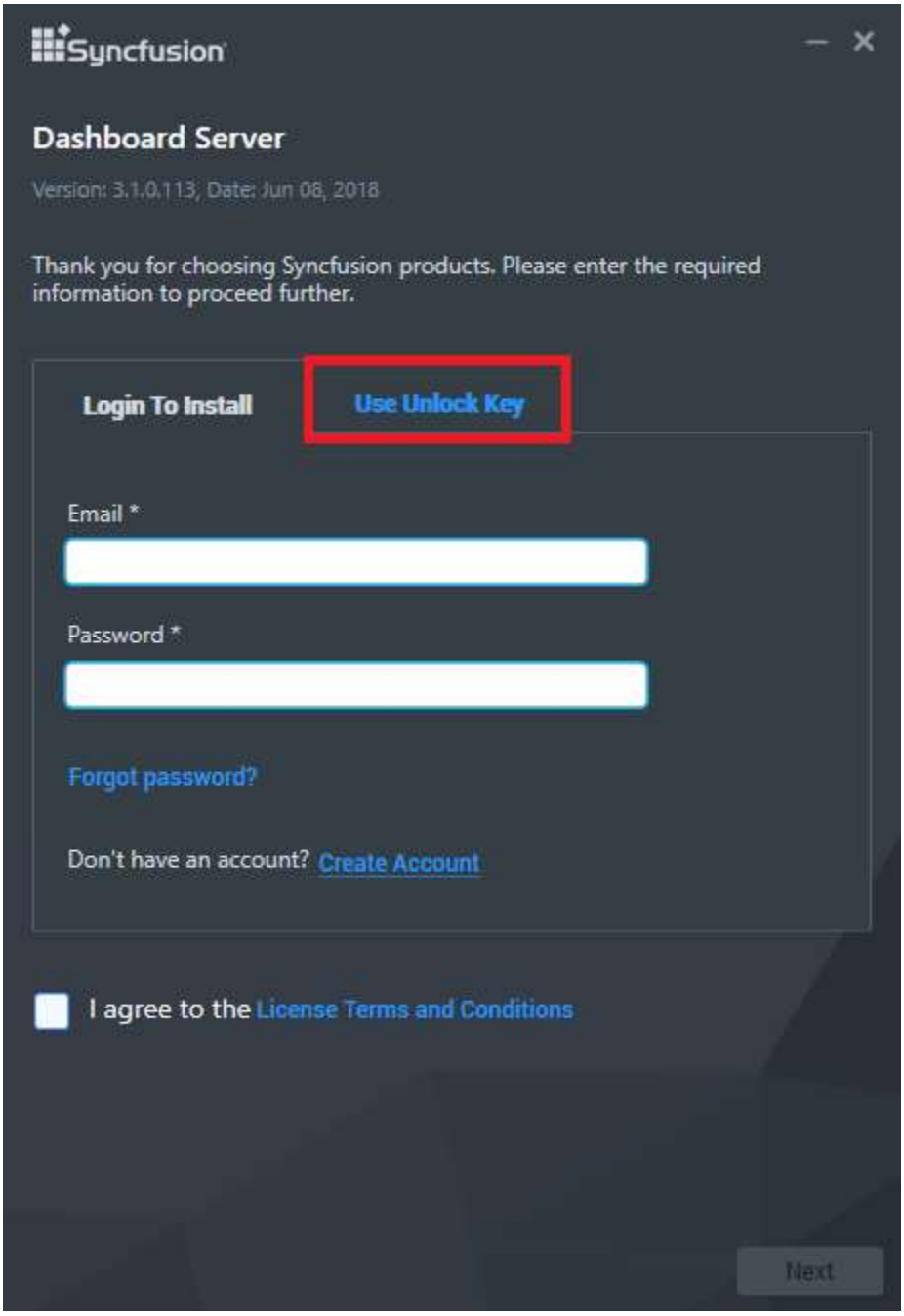
This topic details the steps required to install the Dashboard Server.

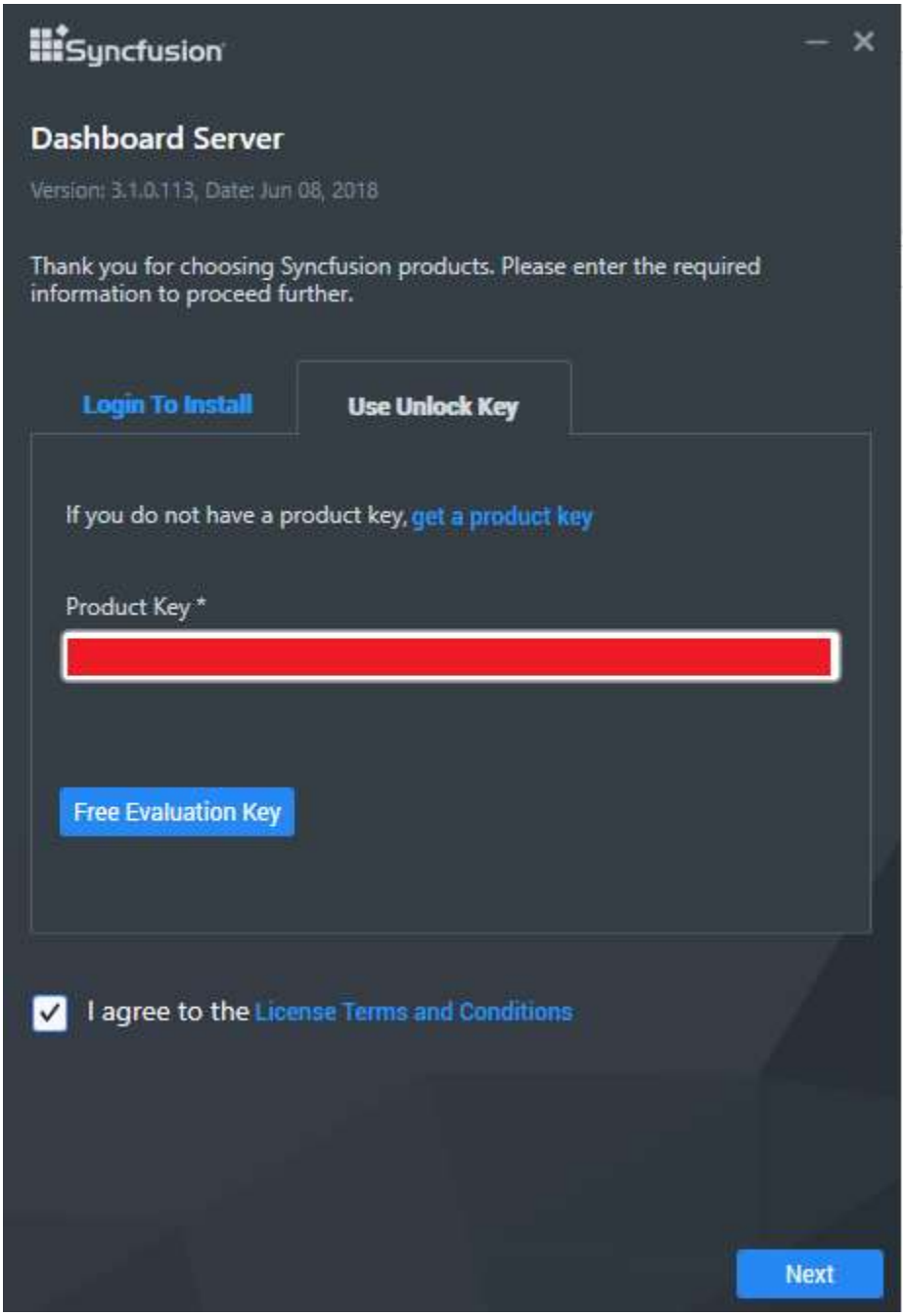
To learn about the system requirements needed to deploy the Dashboard Server in your business environment, see System Requirements.

Run the Dashboard Server Installer and type in the credentials of your Syncfusion account to unlock the setup.



You can alternatively type in the unlock key that has been sent to your registered e-mail address to unlock the setup by selecting the Use Unlock Key option.





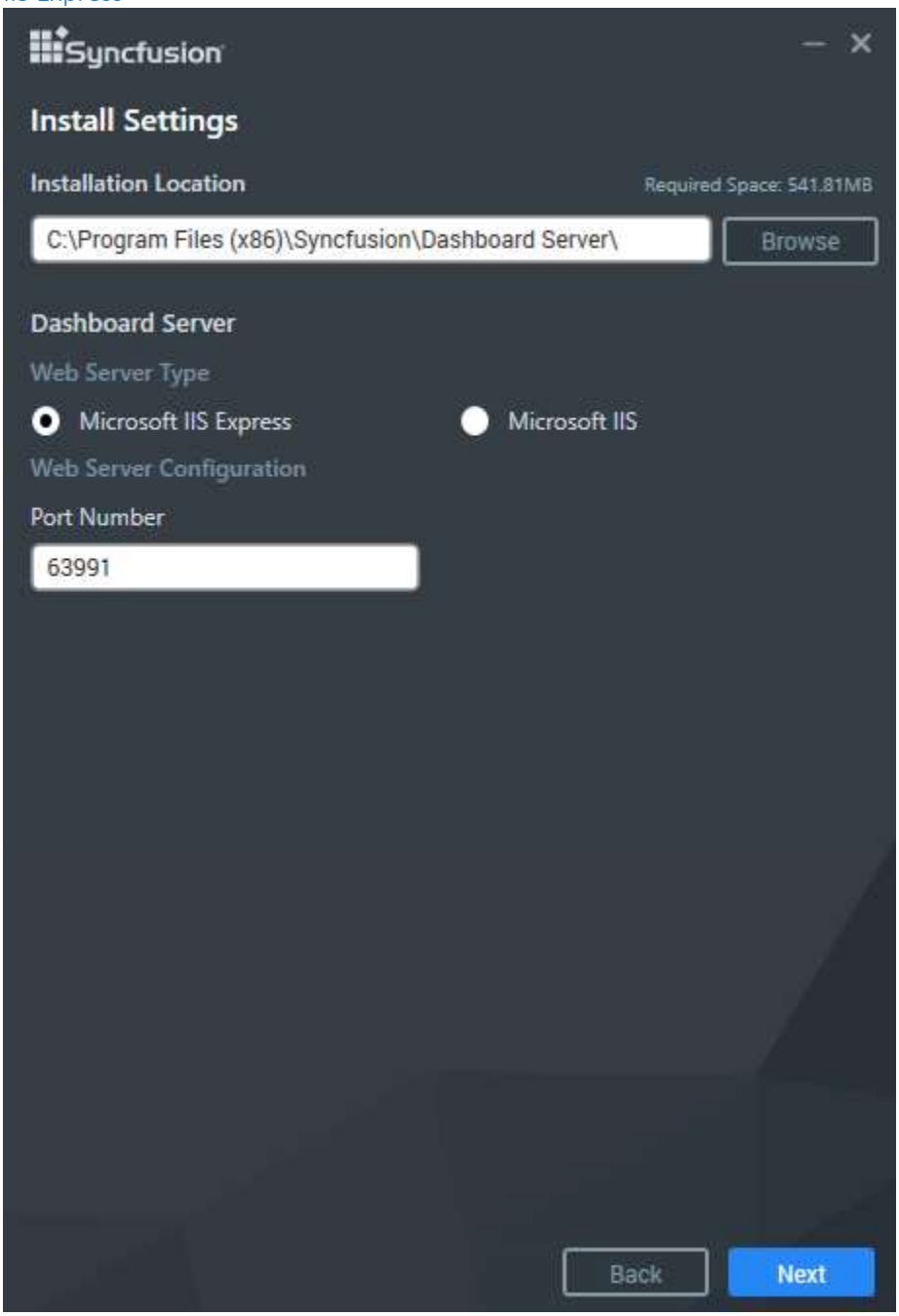
You can check the License Agreement of Dashboard Server by clicking on the License Terms and Conditions.

After you read the license agreement, click on Next to select the installation server type, location and the port number on where the dashboard server has to be hosted.

We have provided the Dashboard Server to be hosted into the following two web server types

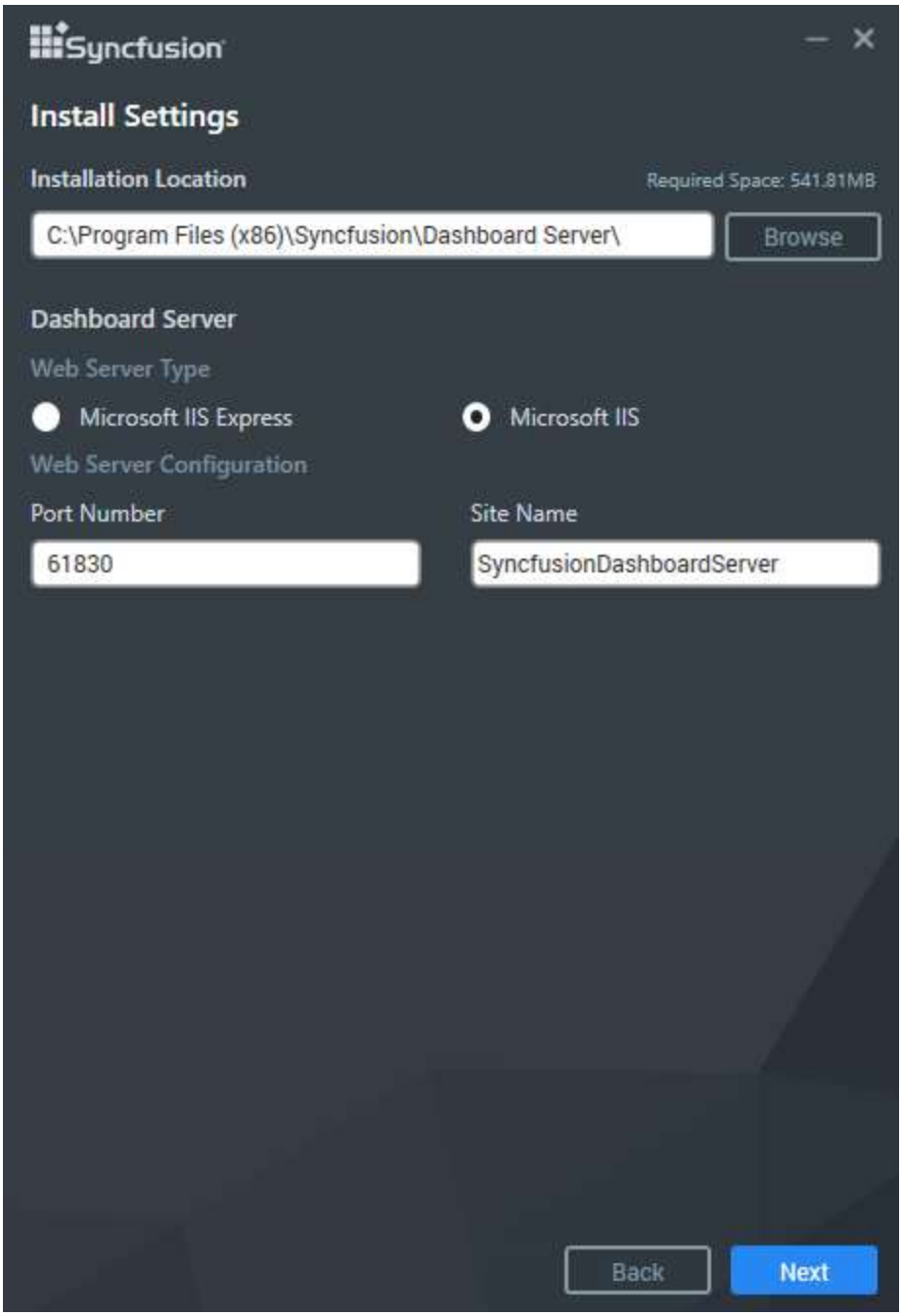
1. IIS Express
2. IIS

IIS Express



IIS

Need to provide the Port number, Location and Site Name to host the Dashboard Server into the IIS.

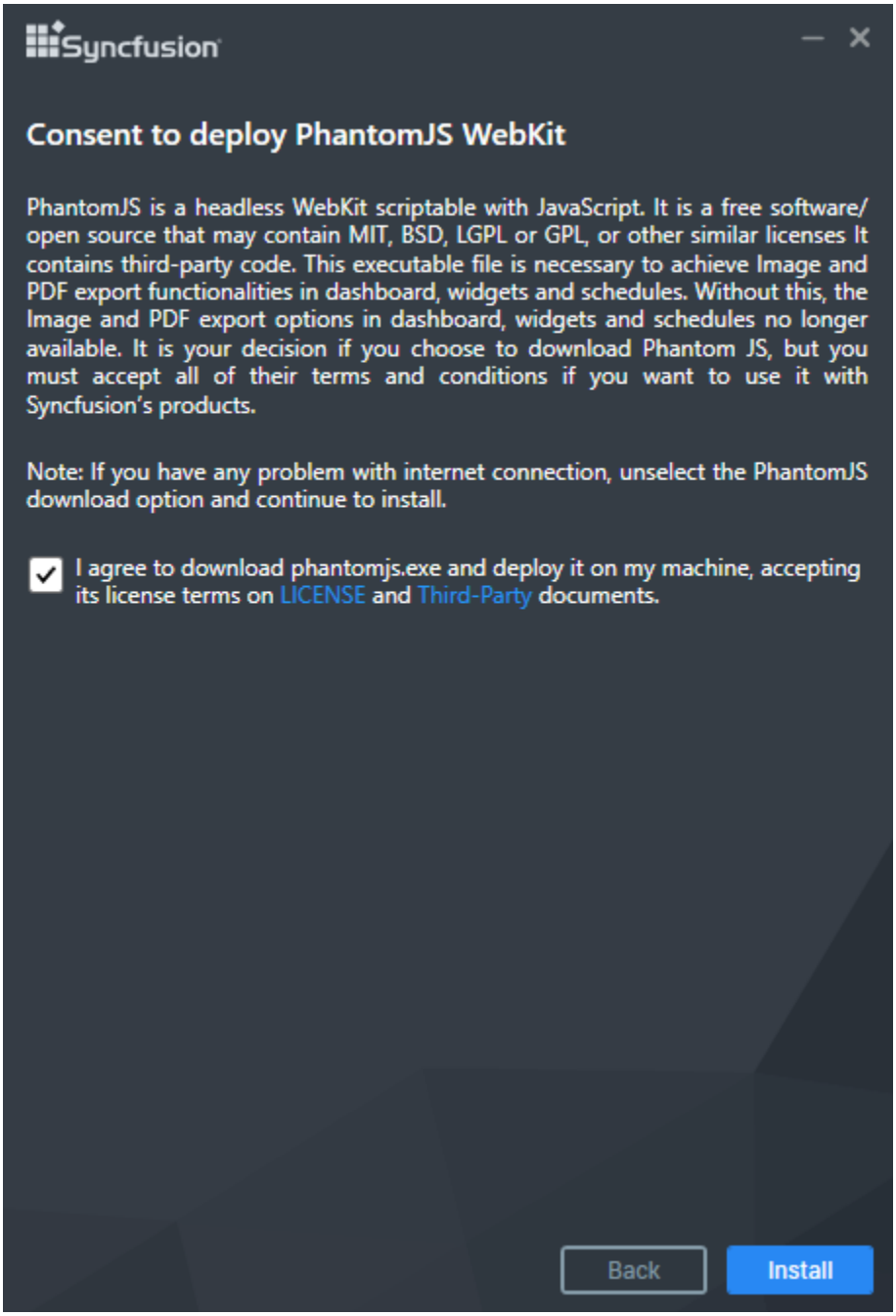


PhantomJs

Information: PhantomJS is a headless WebKit scriptable with JavaScript. This is a free software/open source, and it may contain MIT, BSD, LGPL, or GPL, or other similar licenses that contain third-party code. This executable file is necessary to achieve Image and PDF export functionalities in the Dashboard and widgets. Without this file, the image and PDF export options in the Dashboard and widgets will no longer be available. If you choose to download PhantomJS, must accept all terms and conditions to use it with Syncfusion's products.

Note: If you have any problem with internet connection or do not have internet connection, unselect the PhantomJS download option and continue to install. To manually install the PhantomJS, please refer

this.

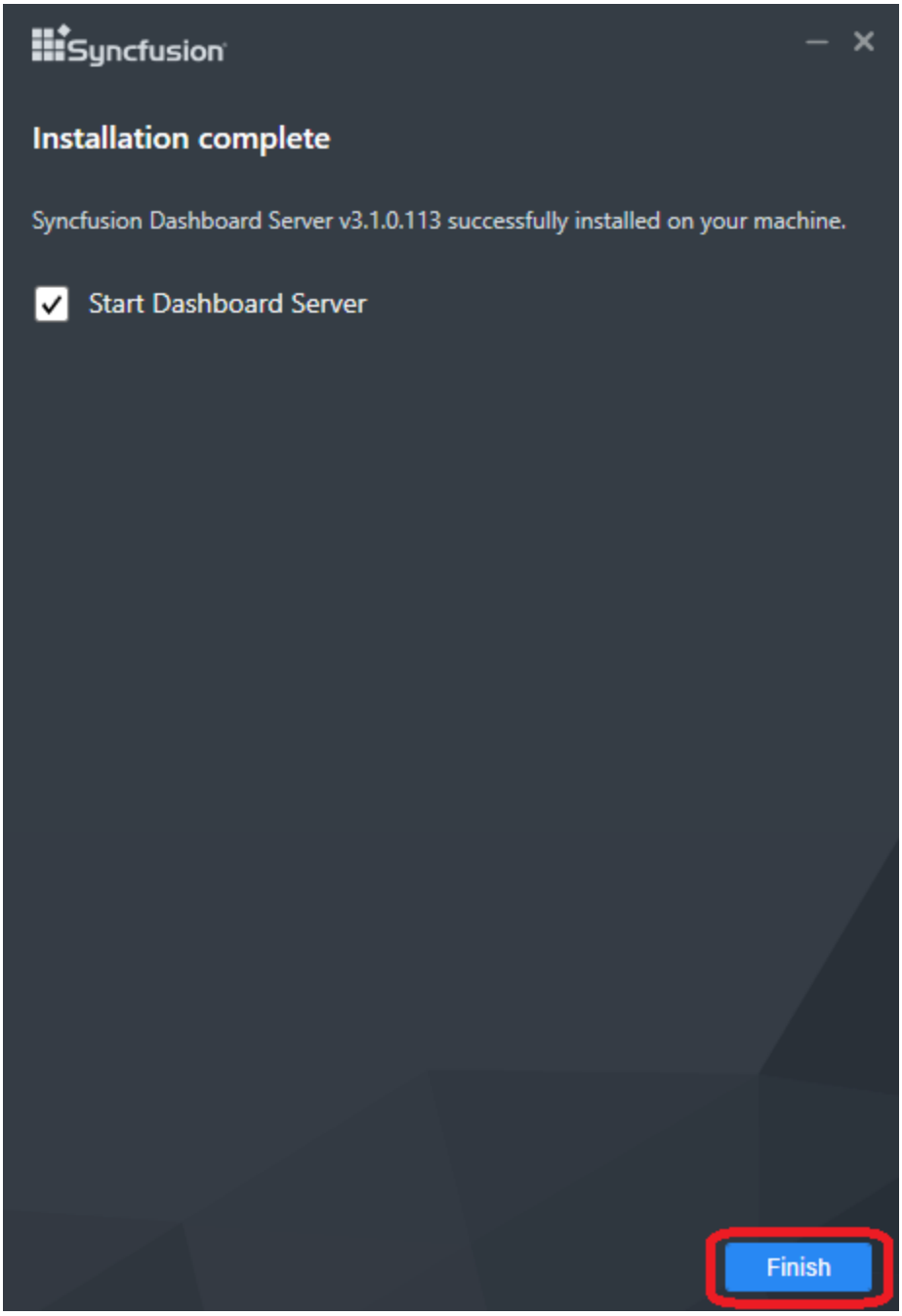


Read and accept the license and third-party terms and conditions through checking the option LICENSE and Third-party for install PhantomJS and click INSTALL.

Dashboard Server will be installed with the below components in the mentioned installation location.

- Dashboard Server web application
- Scheduling Service

Once the installation completes you can start the dashboard server by checking the "Start Dashboard Server" in the last screen and click on finish.



Or you can also start the dashboard server from the shortcuts available in the desktop.

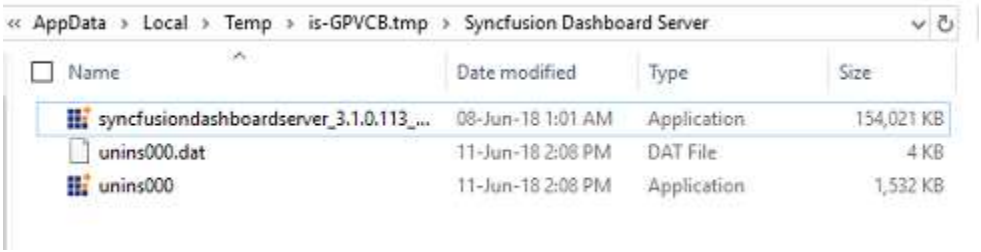
Desktop shortcuts will be provided for starting and stopping the dashboard server and for the dashboard designer.

Note: Dashboard Server does not support multiple versions installed on the same machine.

Note: Dashboard Server cannot be downgraded to the previous version.

Silent Installation

1.Double click the Syncfusion Dashboard Server setup. 2.Syncfusion Dashboard Server setup will be extracted in Temp location (%temp%).



3.Copy the extracted Dashboard Server setup to some other location and cancel the installation. 4.Open the command prompt with administrative privileges and run the extracted Dashboard Server setup with the following arguments.

Arguments:

IIS Express:

/Install silent /InstallPath:{InstallationPath} /pidkey:{unlock*key} /portno:{port*no} /servertype:{server_type} /Log “{LogFilePath\filename.log}”

Example:

/Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard Server /pidkey:@1243453sdffdfvv /portno:54321 /servertype:IISExpress /Log “C:\Program Files (x86)\New\Install.log”

IIS:

/Install silent /InstallPath:{InstallationPath} /pidkey:{unlock*key} /portno:{port*no} /sitename:{site*name} /servertype:{server*type} /Log “{LogFilePath\filename.log}”

Example:

/Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard Server /pidkey:@1243453sdffdfvv /portno:54321 /sitename:SyncfusionDashboardServer /servertype:IIS /Log “C:\Program Files (x86)\New\Install.log”

```
C:\Users\Syncfusion\Desktop\RefreshBuilds>syncfusiondashboardserver_3.1.0.114_2203.exe /Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard Server /pidkey:@1243453sdffdfvv /portno:54321 /sitename:SyncfusionDashbaordServer /servertype:IIS /Log "C:\Program Files (x86)\New\Install.log"
```

Now, Syncfusion Dashboard Server will be installed in silent mode.

Consent to deploy PhantomJS WebKit

PhantomJS is a headless WebKit scriptable with JavaScript. It is a free software/open source that may contain MIT, BSD, LGPL or GPL, or other similar licenses It contains third-party code. This executable file is necessary to achieve Image and PDF export functionalities in dashboard, widgets and schedules. Without this, the Image and PDF export options in dashboard, widgets and schedules no longer available. It is your decision if you choose to download Phantom JS, but you must accept all of their terms and conditions if you want to use it with Syncfusion’s products.

To download PhantomJS application and deploy it on your machine, you should accept it's license terms on LICENSE and Third-Party document. Then, you can download PhantomJS by clicking here.

Once download completed, extract the zip file and then copy the PhantomJS application from the zip extracted location and paste it in the below mentioned install locations.

Install Locations:

1. {InstallPath}\Dashboard Server\DashboardServer.Web\DashboardService
2. {DeploymentPath}\Dashboard Server\DashboardServer.Web\DashboardService

Examples:

1. C:\Program Files (x86)\Syncfusion\Dashboard Server\DashboardServer.Web\DashboardService
2. C:\Syncfusion\Dashboard Server\DashboardServer.Web\DashboardService

Deployment

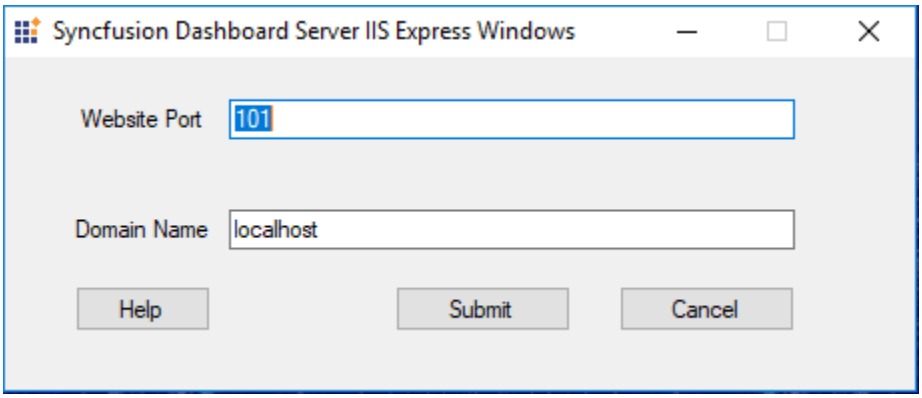
- Dashboard Server web application opens in your default browser with the specified port number at http://localhost:[port_number]/

We have shipped two utilities with the Dashboard Server to host the application in IIS and in IIS Express.

Host as website in IIS Express

1. Run the program ConfigureDashboardServerIISExpress.exe from the following installed location to host the dashboard server in IIS Express.

{Installed_ Location}\Syncfusion\Dashboard Server\Utilities\DashboardServerIISExpress\ConfigureDashboardServerIISExpress.exe



Note: By default, it will show a random port number. An unused port can also be chosen to host in that port.

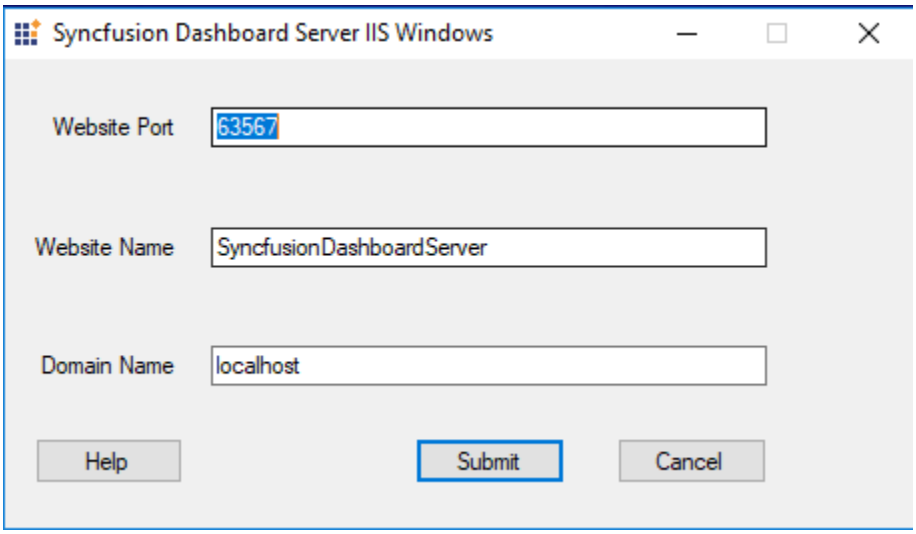
2. Click Submit and this program will host the application in IIS Express and Dashboard Server application will be launched in browser.

Host as website in IIS

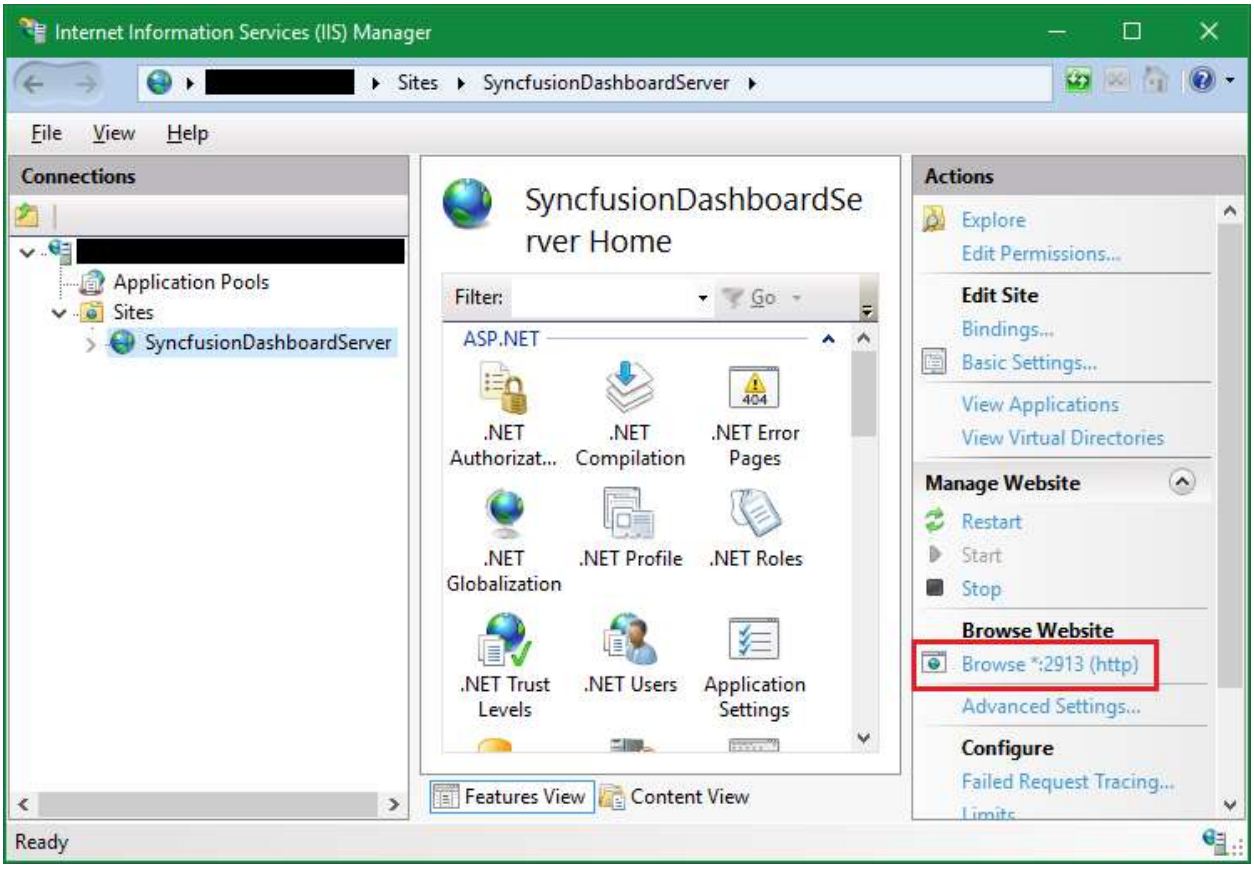
Dashboard Server can also be hosted in IIS by following the below steps.

1. Run the program ConfigureDashboardServerIIS.exe from the following installed location to host the dashboard server in IIS

{Installed_Location}\Syncfusion\Dashboard Server\Utilities\DashboardServerIIS\ConfigureDashboardServerIIS.exe



2. Type in a unused port for the Dashboard Server as like in the above image. This program will host the application in IIS and the Dashboard Server can be opened from the browse button in the IIS.

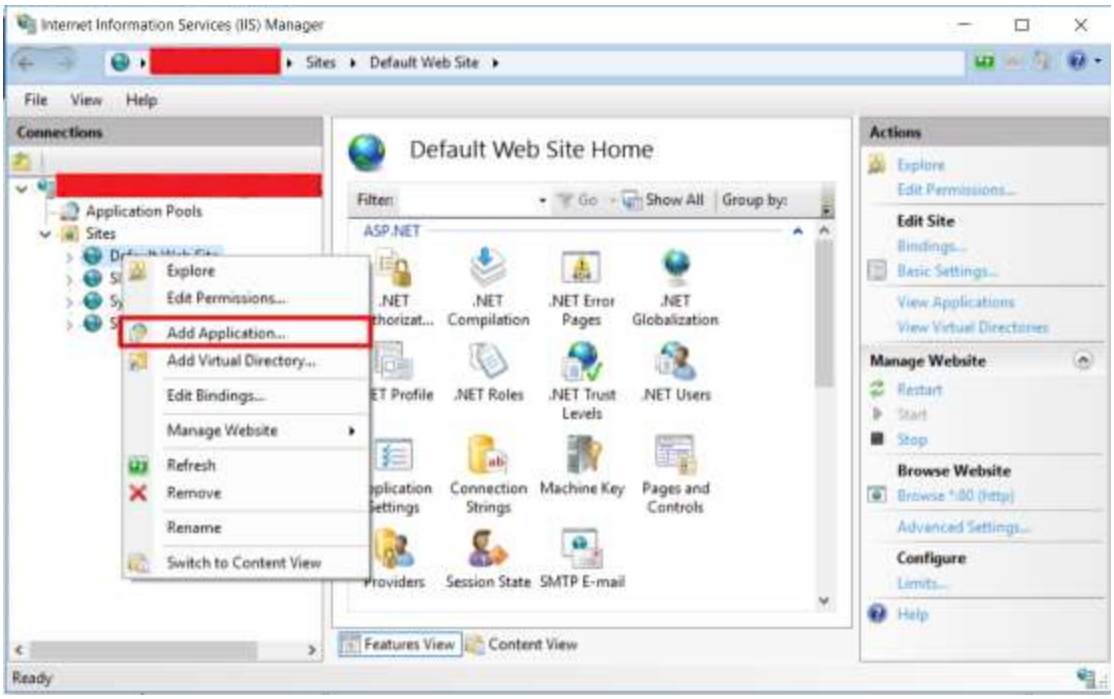


Host as Application in IIS

Dashboard Server can also be hosted as Application in IIS by following the below steps.

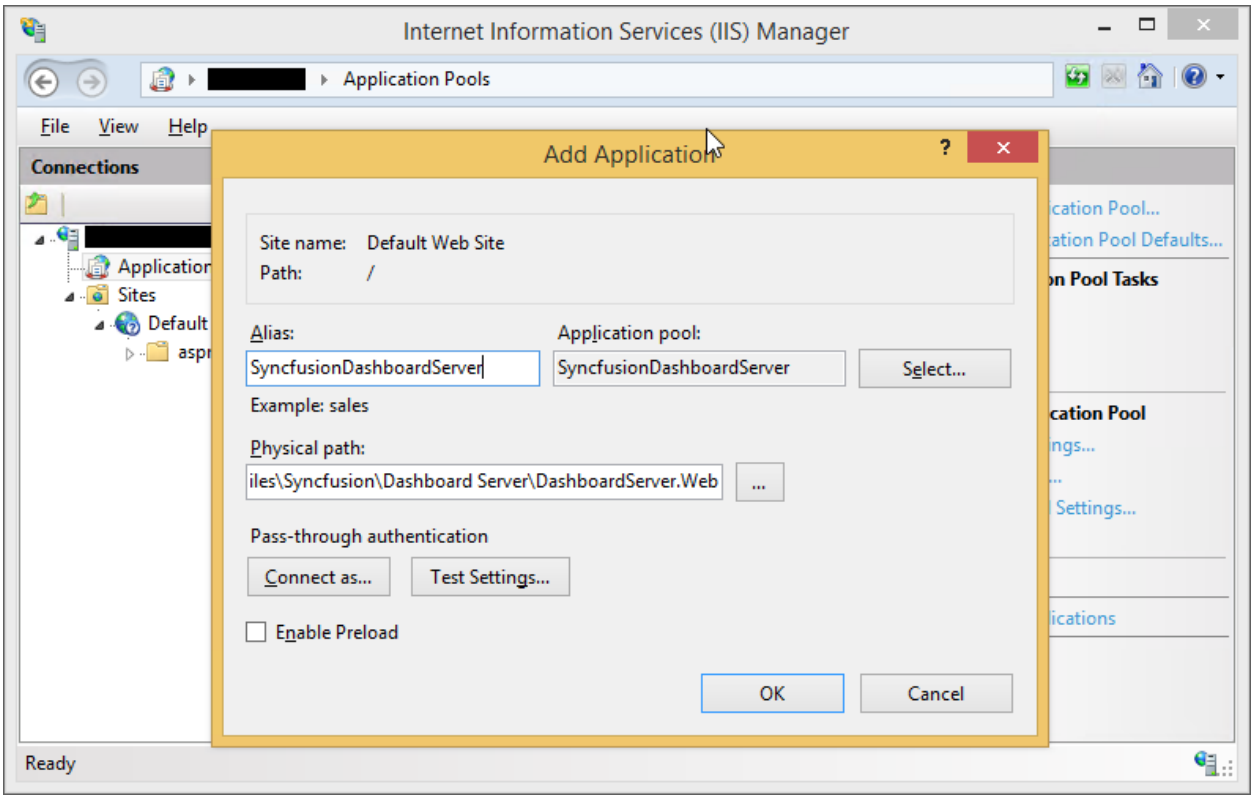
Add Dashboard Server as application

Right click the Website hosted in IIS and choose Add Application as below figure.



And Fill the following details as below figure

1. Alias name
2. Application pool
3. Physical path



Convert the sub folders as application

We have the following folders to be converted as application.

1. API
2. WindowsAuthentication
3. DashboardService

Right click the folder and choose Convert to Application as below figure

![Host Dashboard Server as application in IIS - Convert to sub Application](images/Convert to application.png)

SSL

To enable SSL for the Dashboard Server application, you will need a valid SSL certificate. Please check the below link on how to Obtain an SSL certificate and install it to a website in IIS.

http://www.iis.net/learn/manage/configuring-security/how-to-set-up-ssl-on-iis

Note: If you want to access Dashboard Server from a different machine to the one it's installed on, use the URL http://machine*name:[port*number] or http://machine*ip*address:[port_number]

SSL for Dashboard Service

It is must to enable the SSL for the Dashboard Service if you have configured the SSL for Dashboard Server Application.

To configure SSL for Dashboard Service, run Syncfusion Dashboard Service Configuration Manager application.

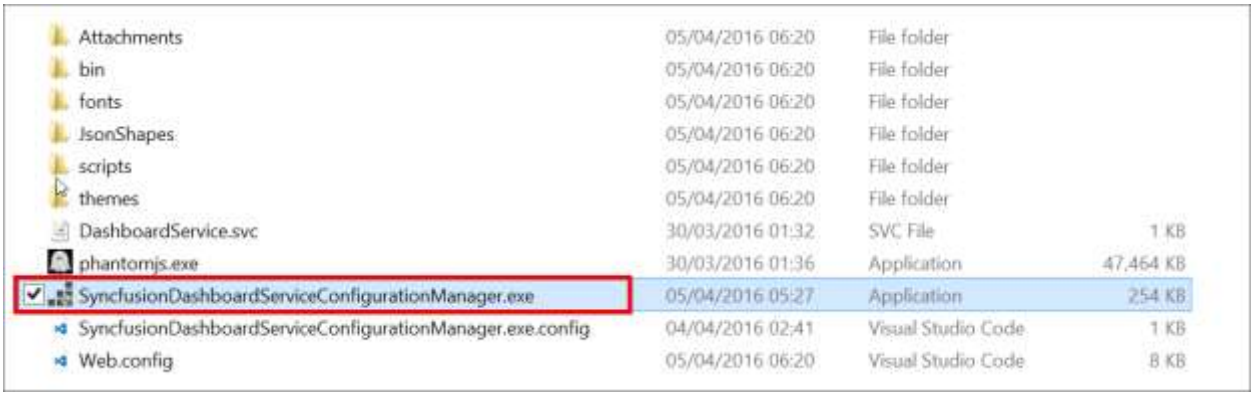
Dashboard Server is deployed in the below location by default.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\

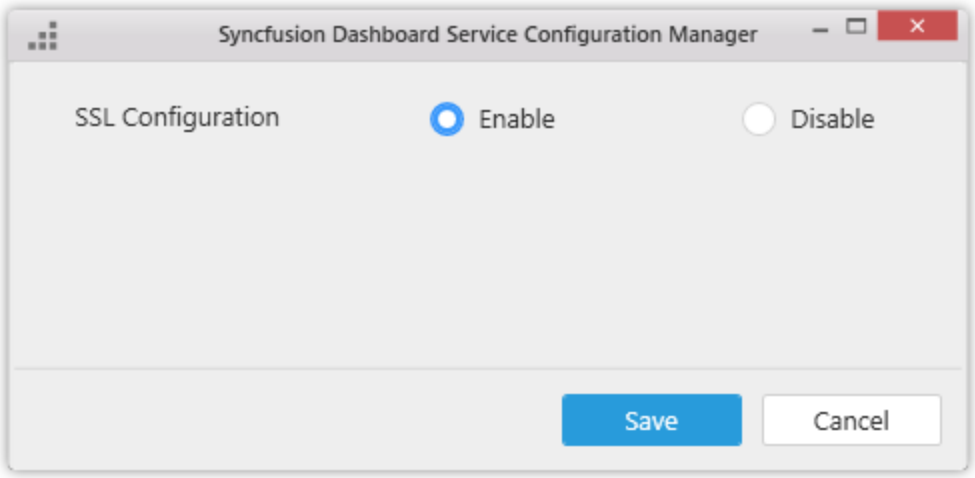
For example, C:\Syncfusion\Dashboard Server\DashboardServer.Web\

We have shipped a utility with the Syncfusion Dashboard Server application in the below location by default.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\DashboardService



Choose the type of configuration required and click on Save button.



Note: SyncfusionDashboardServiceConfigurationManager.exe execution requires administrator mode.

Installation and Deployment of Syncfusion Dashboard Server version 3.2

This section explains on how to install and deploy the Syncfusion Dashboard Server version 3.2 along with the User Management Server.

Download Setup

- You can download the latest Dashboard Server setup from here
- Licensed customers can download the install from the downloads section

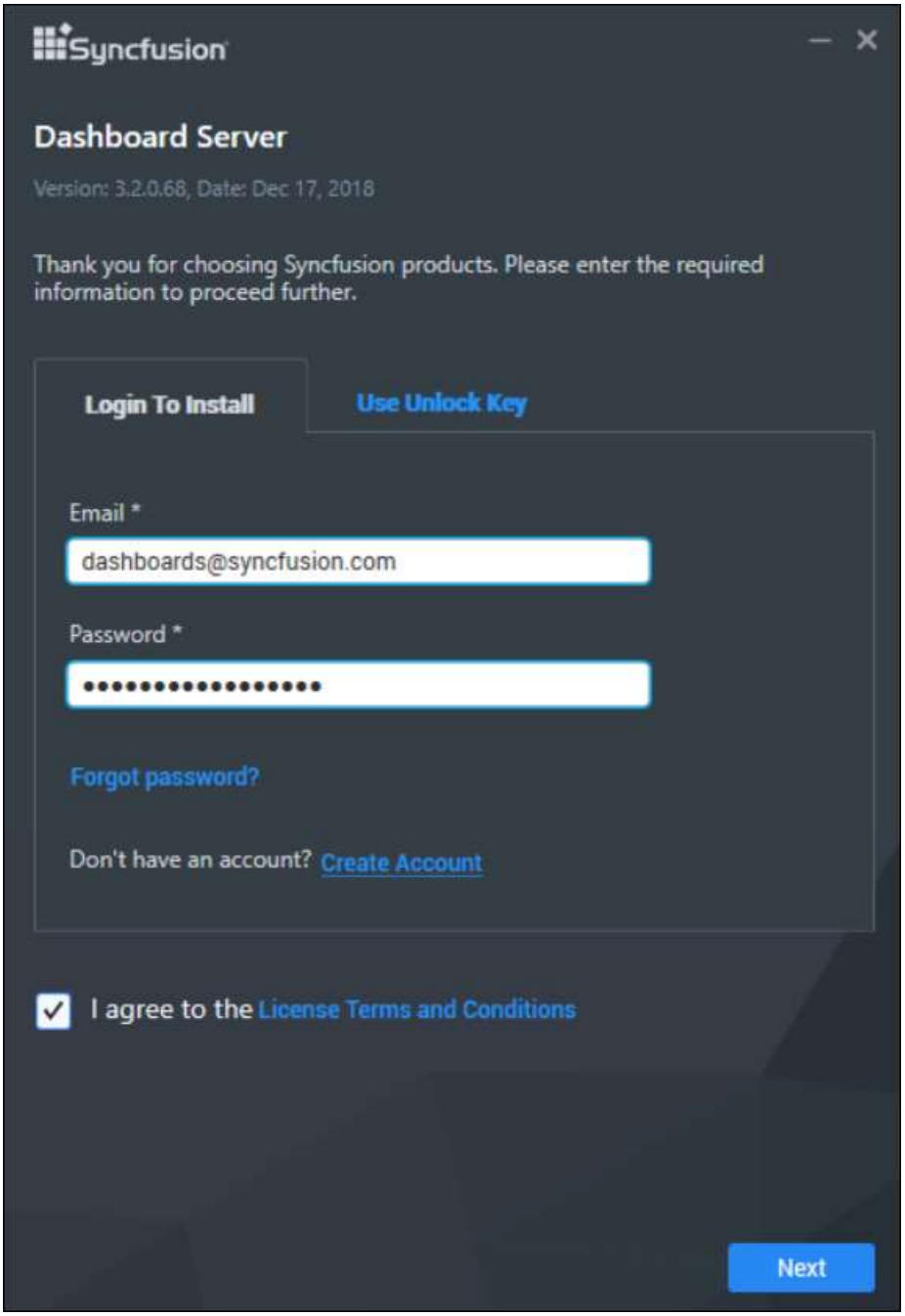
Note: The key to unlock the setup will be sent to your registered e-mail address.

Installation

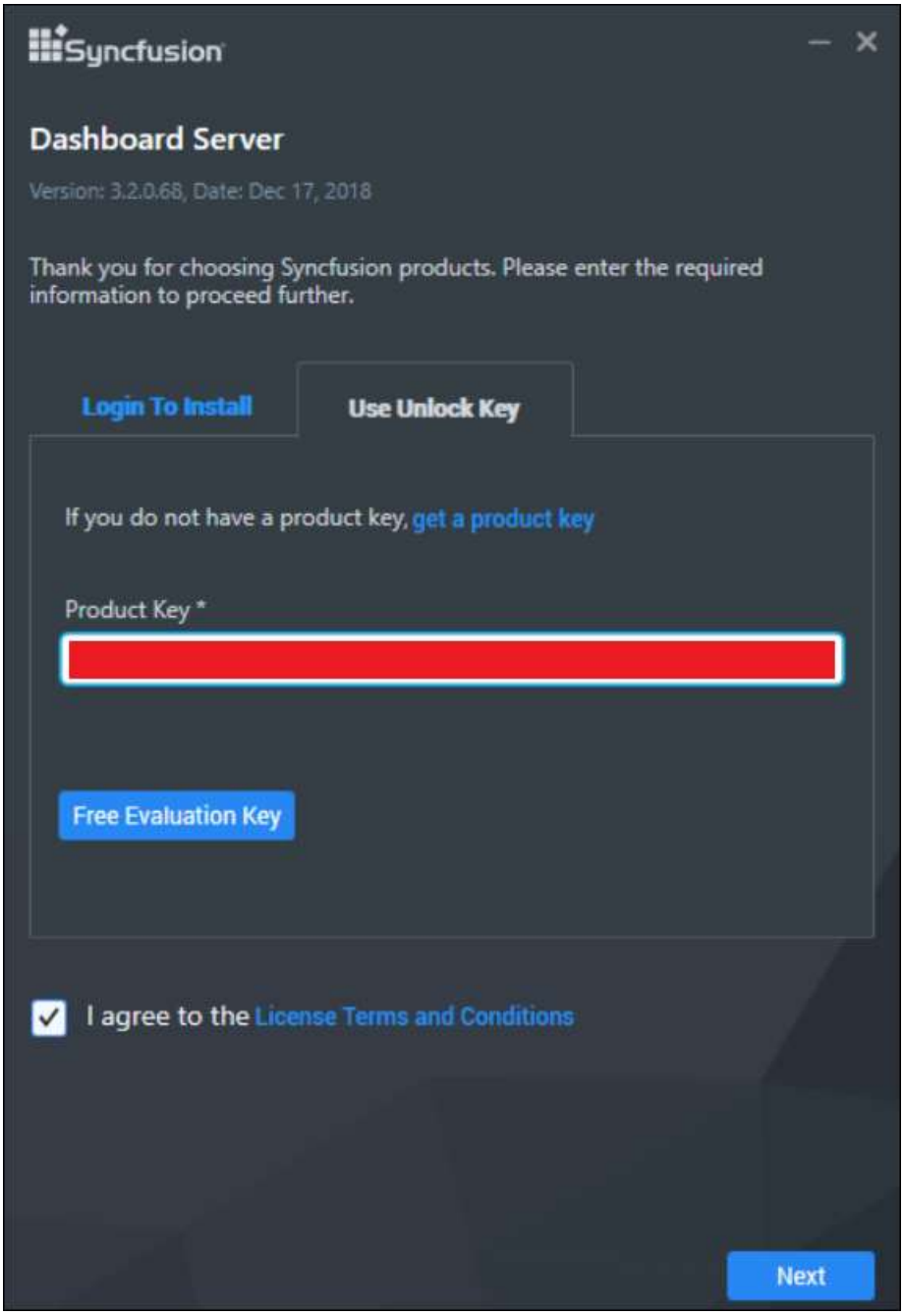
This topic details the steps required to install the Dashboard Server version 3.2 along with User Management Server.

To learn about the system requirements needed to deploy the Dashboard Server and User Management Server in your business environment, see System Requirements.

Run the Dashboard Server Installer and type in the credentials of your Syncfusion account to unlock the setup.



You can alternatively type in the unlock key that has been sent to your registered e-mail address to unlock the setup by selecting the Use Unlock Key option.



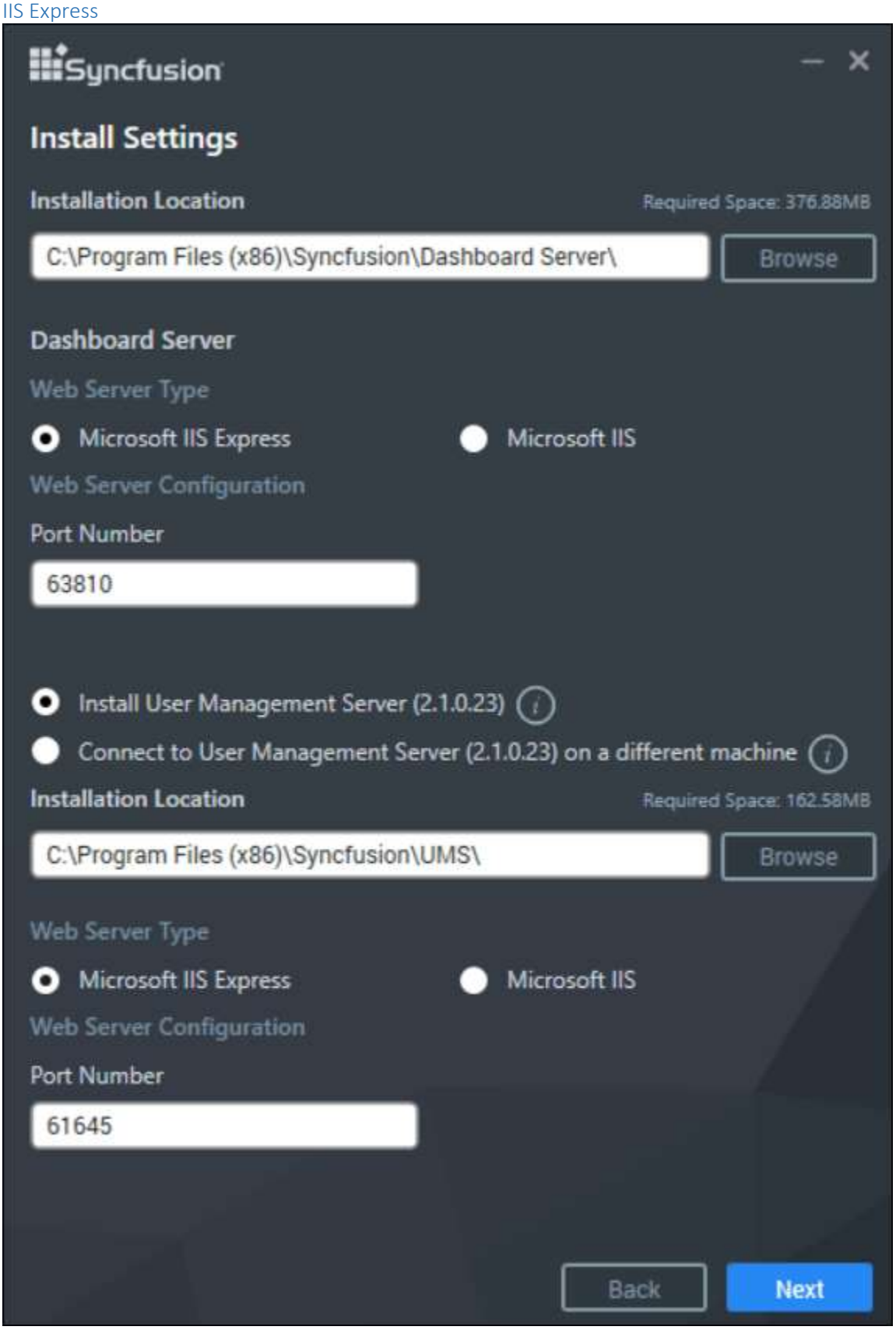
You can check the License Agreement of Dashboard Server by clicking on the License Terms and Conditions.

After you read the license agreement, click on Next to select the installation server type, location and the port number on where the dashboard server and User Management Server has to be hosted.

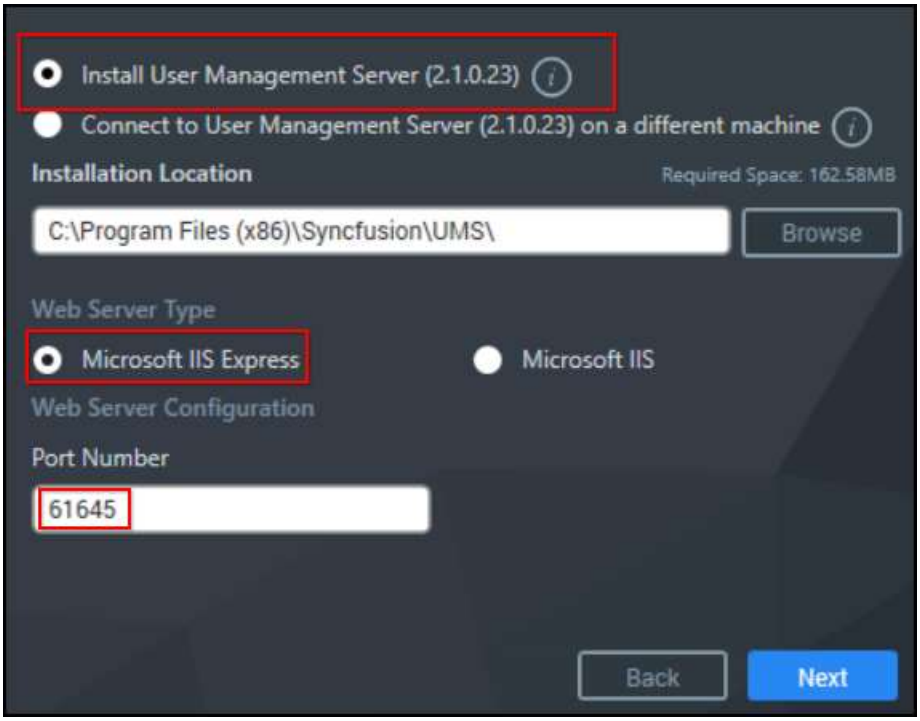
We have provided the Dashboard Server and User Management Server to be hosted into the following two web server types

1. IIS Express
2. IIS

IIS Express



Install User Management Server option will install the User Management Server in mentioned port number.



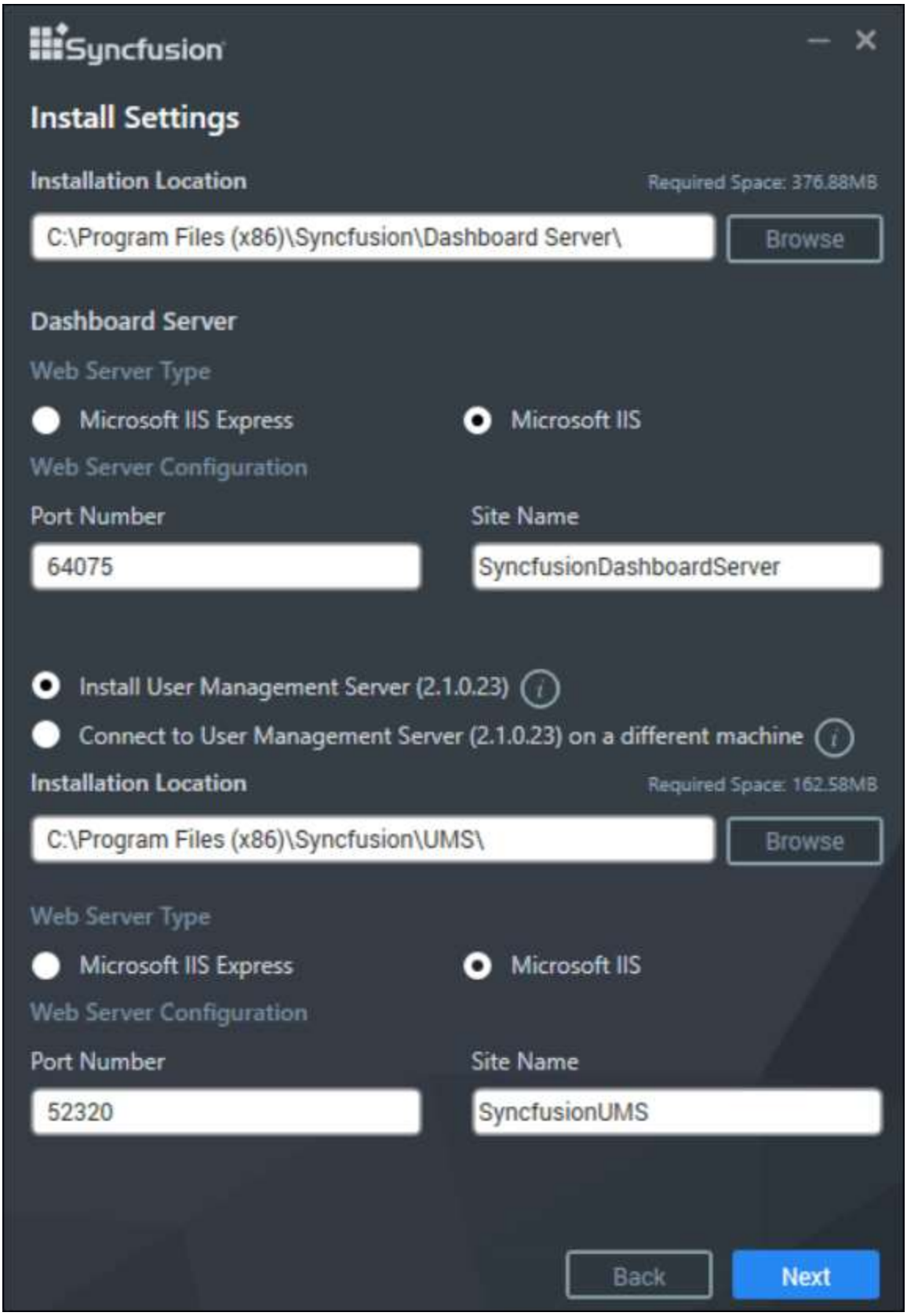
Connect to User Management Server on a different machine option will able to connect Dashboard Server to the latest User Management on another machine. You can find how to connect to User Management Server here.

After you select the installation server type, click on Next.

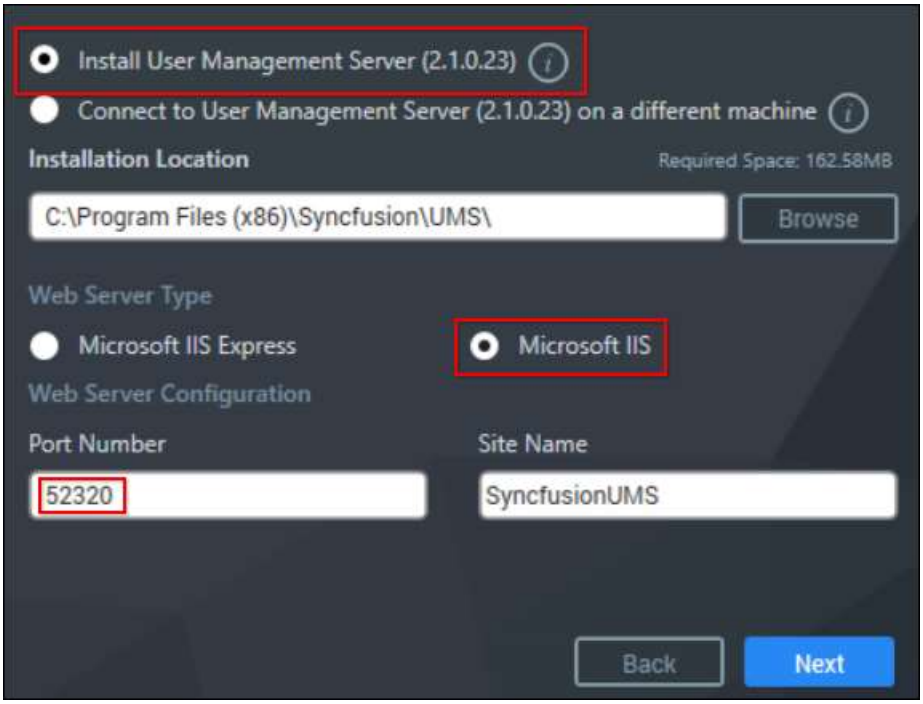
Note: Latest User Management Server is mandatory to run Syncfusion Dashboard Server version 3.2

IIS

Need to provide the Port number, Location and Site Name to host the Dashboard Server and User Management Server into the IIS.



Install User Management Server option will host the User Management Server in mentioned port number.



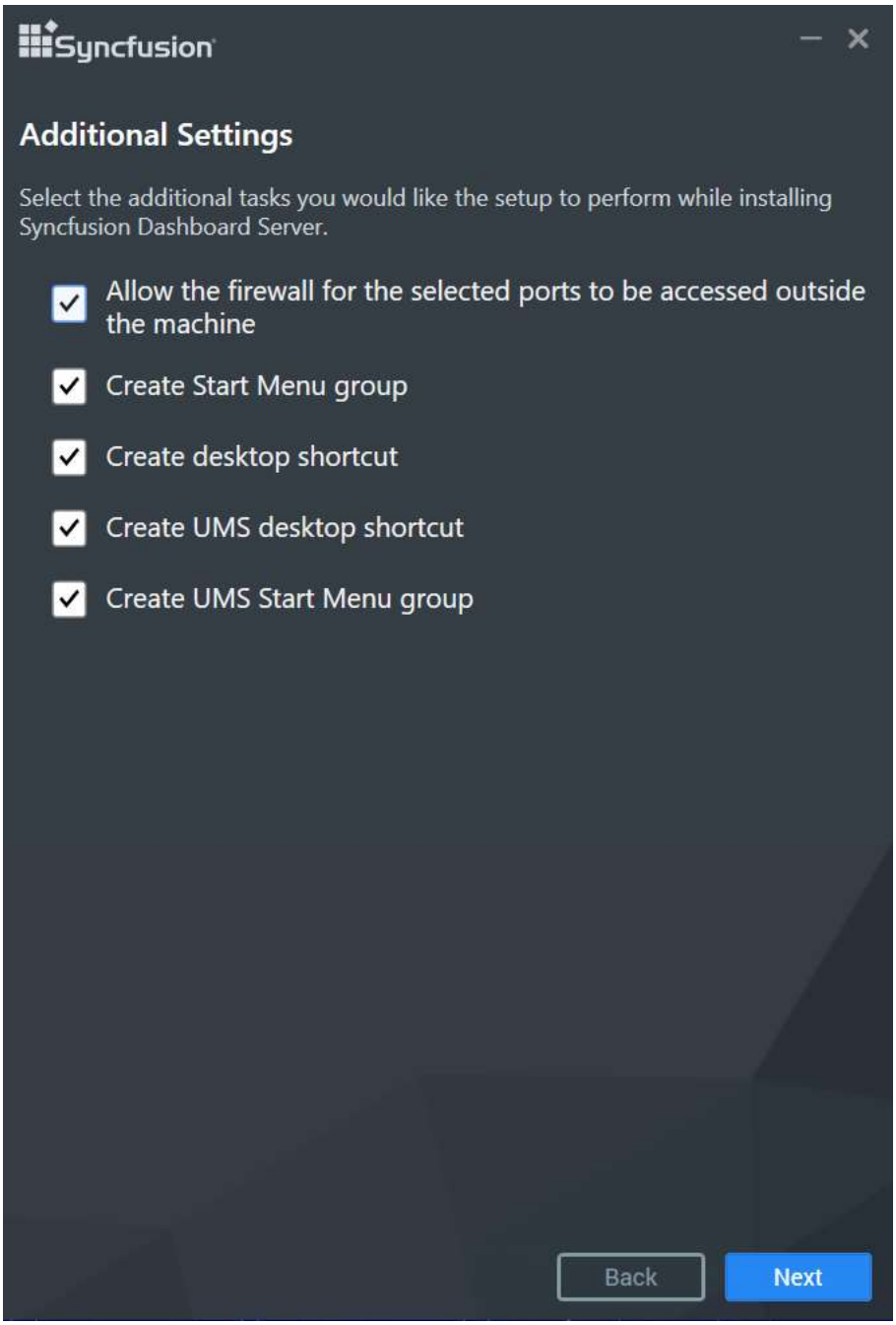
Connect to User Management Server on a different machine option will able to connect Dashboard Server to the latest User Management on another machine. You can find how to connect to User Management Server here.

After you select the installation server type, click on Next.

Note: Latest User Management Server is mandatory to run Dashboard Server version 3.2

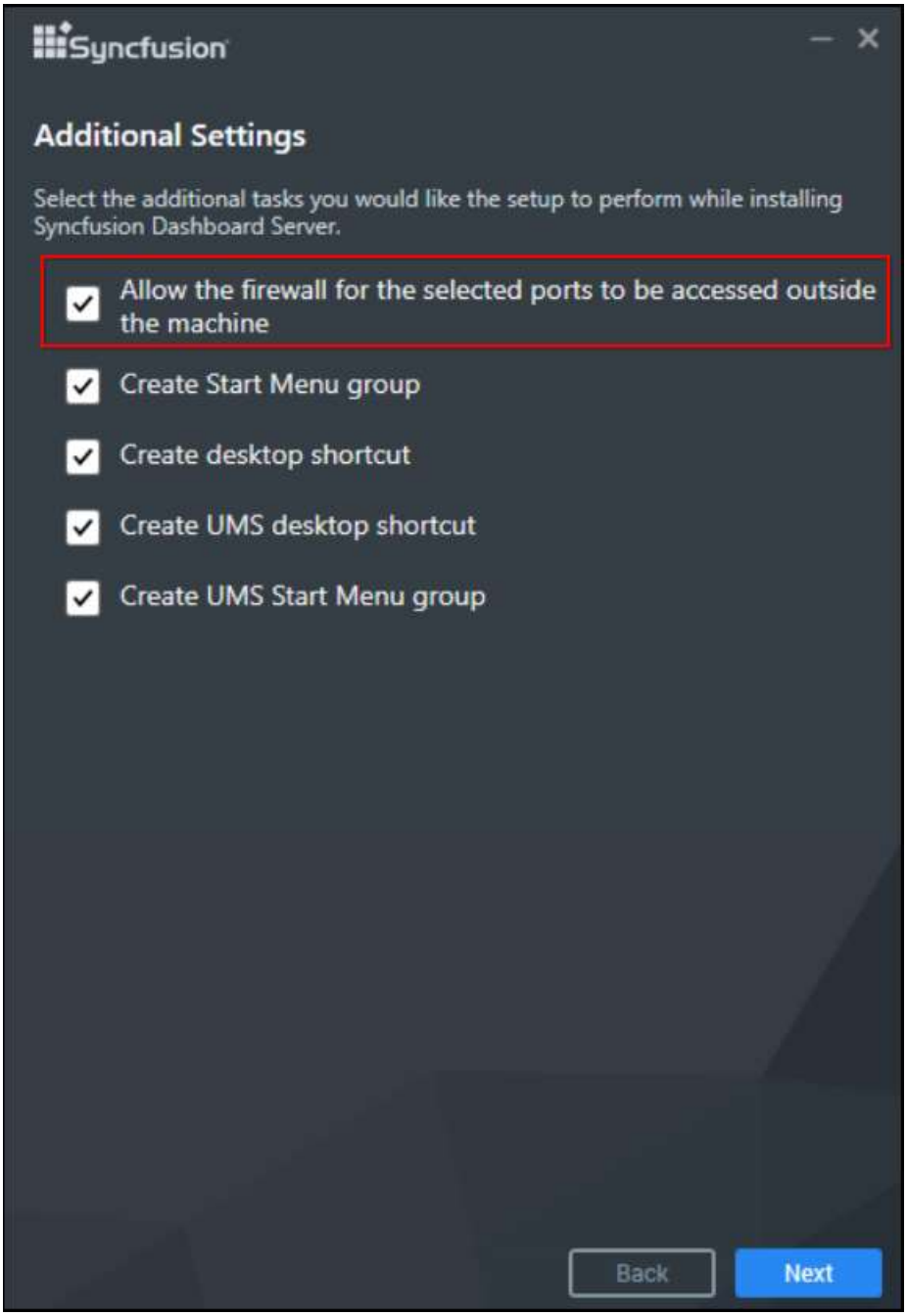
Additional Settings

Perform the additional tasks like desktop shortcuts creation and start menu shortcut creations. If you want to perform the additional tasks, you can check the options. Otherwise, you can uncheck.



Allow firewall for selected ports

This option will allow selected ports of Dashboard Server and User Management Server in firewall to make it accessible outside of the installed machine.

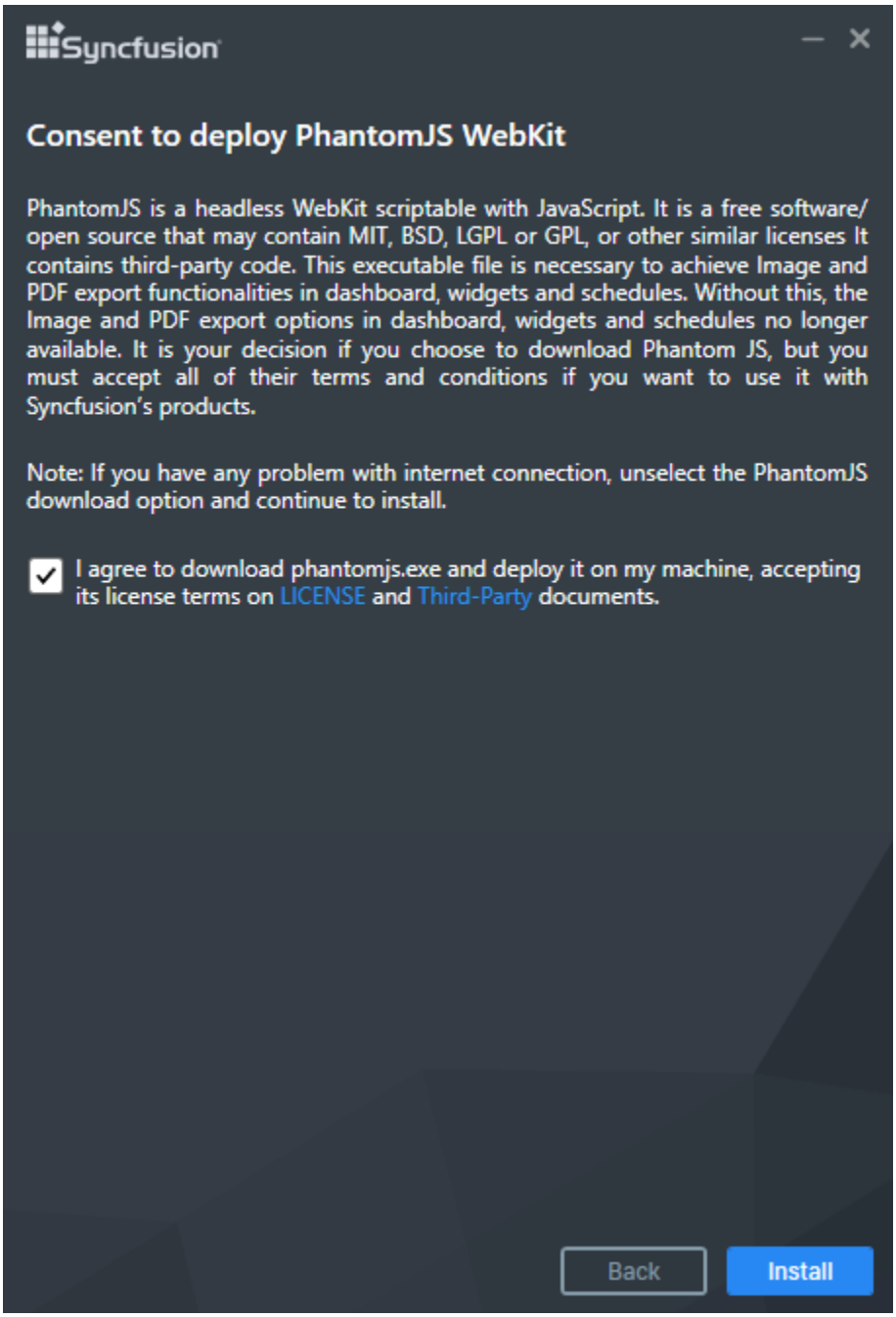


Click Next to proceed further.

PhantomJs

Information: PhantomJS is a headless WebKit scriptable with JavaScript. This is a free software/open source, and it may contain MIT, BSD, LGPL, or GPL, or other similar licenses that contain third-party code. This executable file is necessary to achieve Image and PDF export functionalities in the Dashboard and widgets. Without this file, the image and PDF export options in the Dashboard and widgets will no longer be available. If you choose to download PhantomJS, must accept all terms and conditions to use it with Syncfusion's products.

Note: If you have any problem with internet connection or do not have internet connection, unselect the PhantomJS download option and continue to install. To manually install the PhantomJS, please refer this.



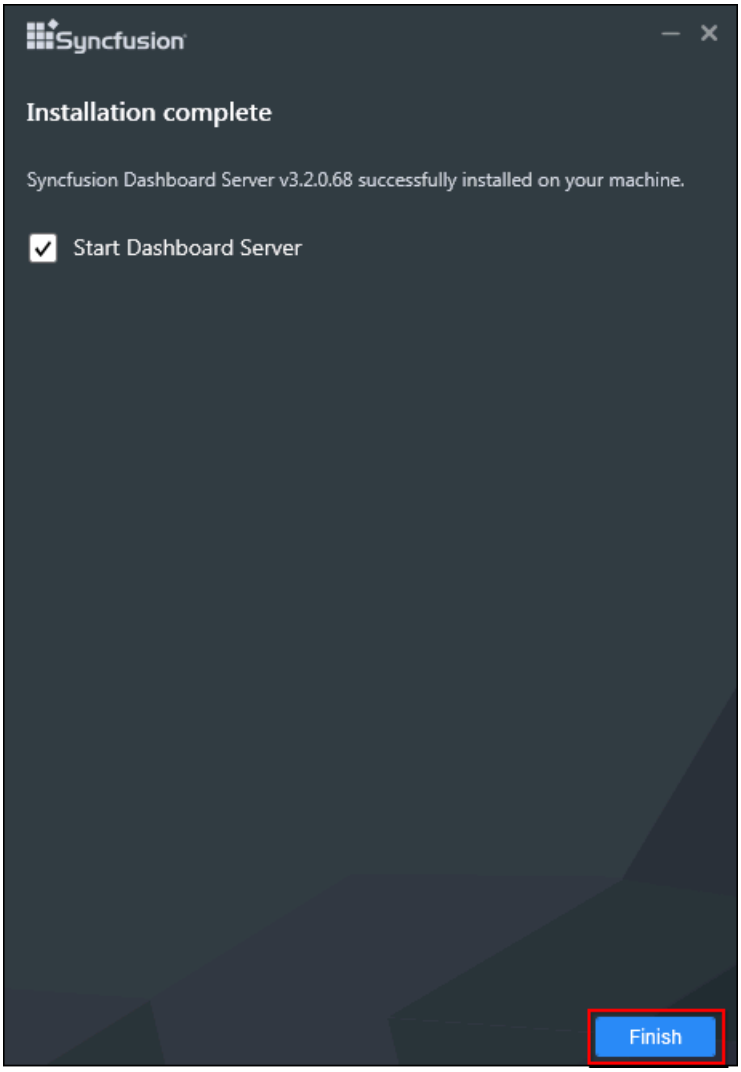
Read and accept the license and third-party terms and conditions through checking the option LICENSE and Third-party for install PhantomJS and click INSTALL.

Dashboard Server will be installed with the below components in the mentioned installation location.

- Dashboard Server web application

- Scheduling Service

Once the installation completes you can start the dashboard server by checking the "Start Dashboard Server" in the last screen and click on Finish.



Or you can also start the dashboard server from the shortcuts available in the desktop.

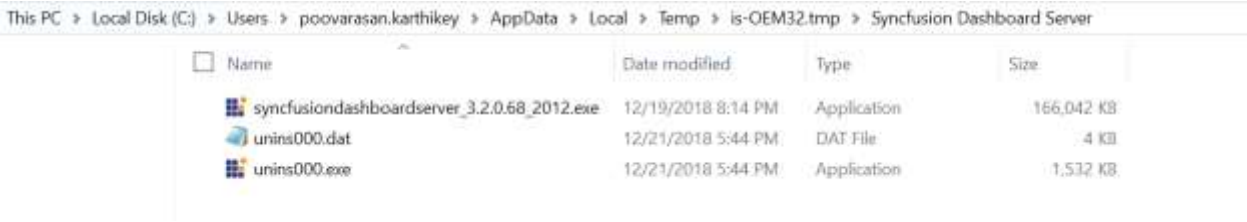
Desktop shortcuts will be provided for starting and stopping the dashboard server and for the dashboard designer.

Note: Dashboard Server does not support multiple versions installed on the same machine.

Note: Dashboard Server cannot be downgraded to the previous version.

Silent Installation

1.Double click the Syncfusion Dashboard Server setup. 2.Syncfusion Dashboard Server setup will be extracted in Temp location (%temp%).



3.Copy the extracted Dashboard Server setup to some other location and cancel the installation. 4.Open the command prompt with administrative privileges and run the extracted Dashboard Server setup with the following arguments.

Arguments:

IIS Express:

/Install silent /InstallPath:{InstallationPath} /pidkey:{unlock*key} /portno:{port*no} /servertype:{server_type} /isstartmenushortcut:{TRUE or FALSE} /issetfirewallrule:{True or False} /Log “{LogFilePath\filename.log}”

Example:

/Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard Server /pidkey:@1243453sdffdfvv /portno:54321 /servertype:IISExpress /isdesktopshortcut:TRUE /isstartmenushortcut:TRUE /issetfirewallrule:True /Log “C:\Program Files (x86)\New\Install.log”

IIS:

/Install silent /InstallPath:{InstallationPath} /pidkey:{unlock*key} /portno:{port*no} /sitename:{site*name} /servertype:{server*type} /isdesktopshortcut:{TRUE or FALSE} /isstartmenushortcut:{TRUE or FALSE} /issetfirewallrule:{True or False} /Log “{LogFilePath\filename.log}”

Example:

/Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard Server /pidkey:@1243453sdffdfvv /portno:54321 /sitename:SyncfusionDashboardServer /servertype:IIS /isdesktopshortcut:TRUE /isstartmenushortcut:TRUE /issetfirewallrule:True /Log “C:\Program Files (x86)\New\Install.log”

```
C:\>C:\Users\labuser\Desktop\ReleaseBuils\syncfusiondashboardserver_3.2.0.68_201
2.exe /Install silent /InstallPath:C:\Program Files (x86)\Syncfusion\Dashboard S
erver /pidkey:@1234fgdrethgsksaln= /portno=54321 /sitename:SyncfusionDashboardSe
rver /servertype:IIS /isdesktopshortcut:TRUE /isstartmenushortcut:TRUE /issetfir
ewallrule:True
```

Now, Syncfusion Dashboard Server will be installed in silent mode.

Consent to deploy PhantomJS WebKit

PhantomJS is a headless WebKit scriptable with JavaScript. It is a free software/open source that may contain MIT, BSD, LGPL or GPL, or other similar licenses It contains third-party code. This executable file is necessary to achieve Image and PDF export functionalities in dashboard, widgets and schedules. Without this, the Image and PDF export options in dashboard, widgets and schedules no longer available. It is your decision if you choose to download Phantom JS, but you must accept all of their terms and conditions if you want to use it with Syncfusion’s products.

To download PhantomJS application and deploy it on your machine, you should accept it's license terms on LICENSE and Third-Party document. Then, you can download PhantomJS by clicking here.

Once download completed, extract the zip file and then copy the PhantomJS application from the zip extracted location and paste it in the below mentioned install locations.

Install Locations:

1. {InstallPath}\Dashboard Server\DashboardServer.Web\API
2. {DeploymentPath}\Dashboard Server\DashboardServer.Web\API

Examples:

1. C:\Program Files (x86)\Syncfusion\Dashboard Server\DashboardServer.Web\API
2. C:\Syncfusion\Dashboard Server\DashboardServer.Web\API

Deployment

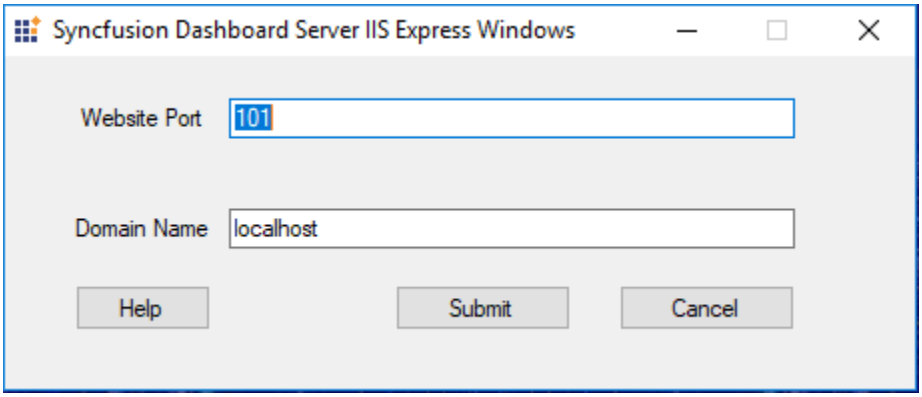
- Dashboard Server web application opens in your default browser with the specified port number given during at the time of installation at http://localhost:[dashboard*server*port_number]/
- User Management Server web application opens in your default browser with the specified port number given during at the time of installation at http://localhost:[user*management*server*port*number]/

We have shipped two utilities with the Dashboard Server and User Management Server to host the application in IIS and in IIS Express.

Host Dashboard Server as website in IIS Express

1. Run the program ConfigureDashboardServerIISExpress.exe from the following installed location to host the dashboard server in IIS Express.

{Installed_ Location}\Syncfusion\Dashboard Server\Utilities\DashboardServerIISExpress\ConfigureDashboardServerIISExpress.exe



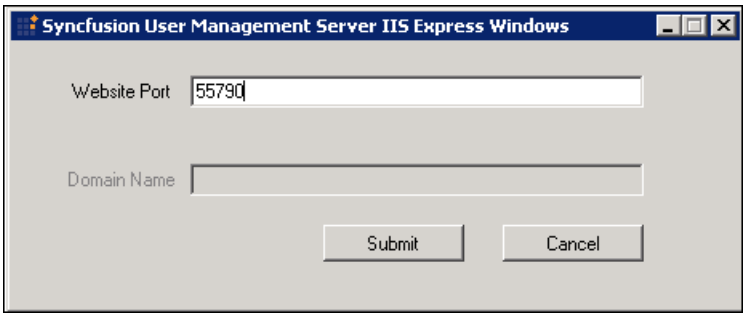
Note: By default, it will show a random port number. An unused port can also be chosen to host in that port.

2. Click Submit and this program will host the application in IIS Express and Dashboard Server application will be launched in browser.

Host User Management Server as website in IIS Express

1. Run the program ConfigureUserManagementServerIISExpress.exe from the following installed location to host the user management server in IIS Express.

{Installed_ Location}\Syncfusion\UMS\Utilities\UserManagementServerIISExpress\ConfigureUserManagementServerIISExpress.exe



Note: By default, it will show a random port number. An unused port can also be chosen to host in that port.

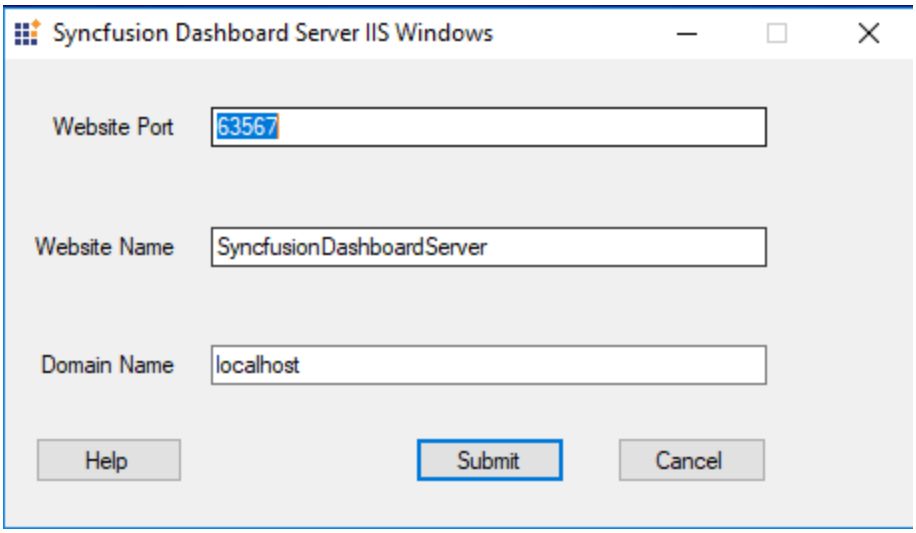
2. Click Submit and this program will host the application in IIS Express and User Management Server application will be launched in browser.

Host Dashboard Server as website in IIS

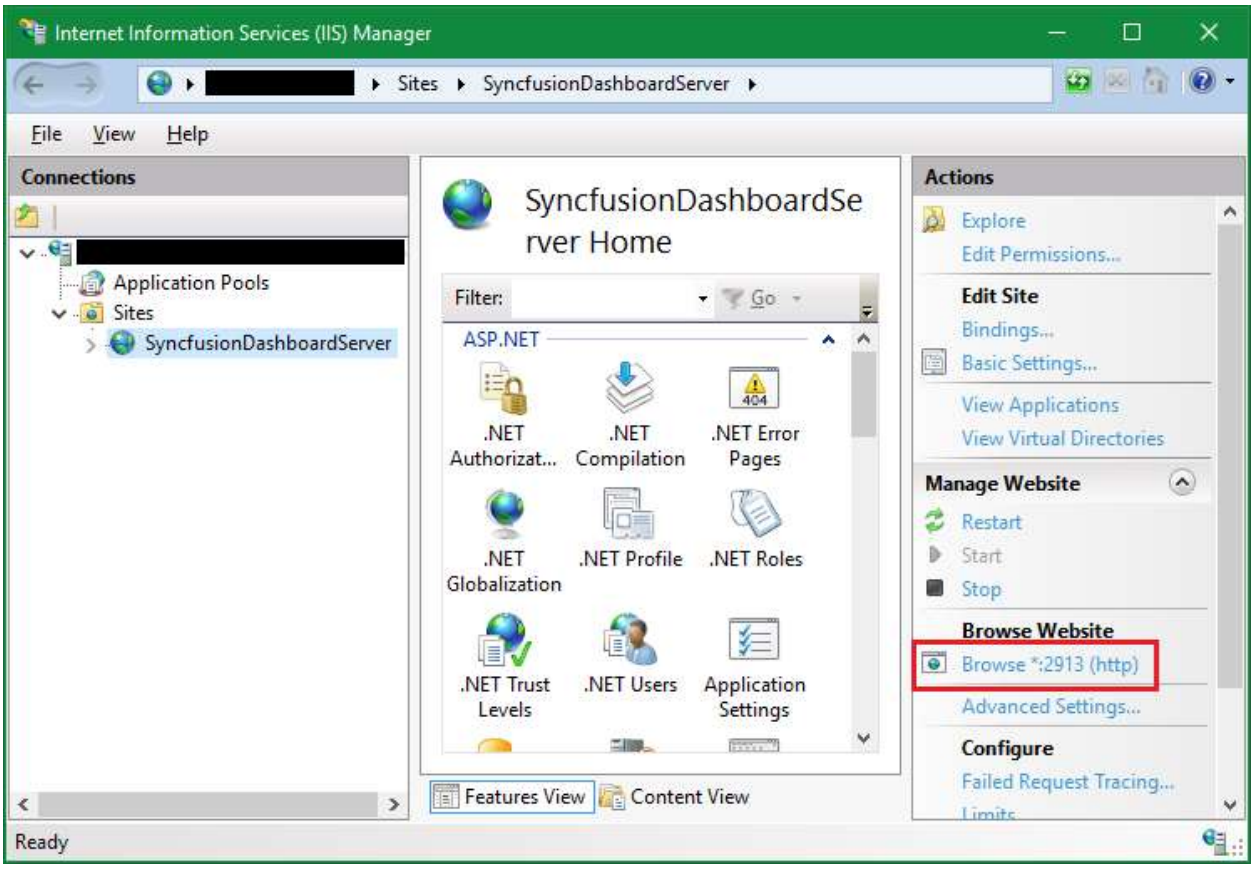
Dashboard Server can also be hosted in IIS by following the below steps.

1. Run the program ConfigureDashboardServerIIS.exe from the following installed location to host the dashboard server in IIS

{Installed_ Location}\Syncfusion\Dashboard Server\Utilities\DashboardServerIIS\ConfigureDashboardServerIIS.exe



2. Type in a unused port for the Dashboard Server as like in the above image. This program will host the application in IIS and the Dashboard Server can be opened from the browse button in the IIS.

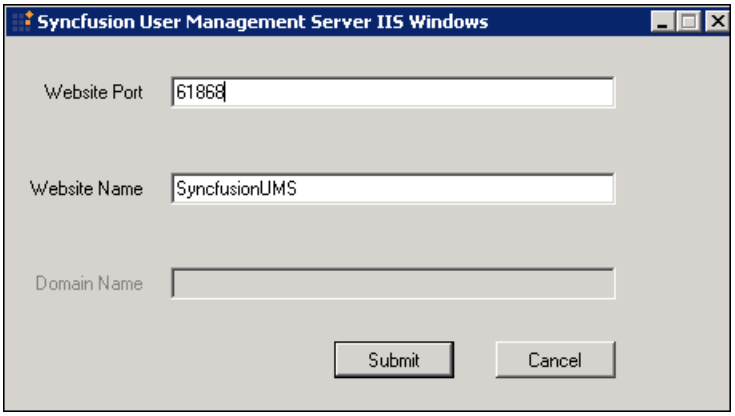


Host User Management Server as website in IIS

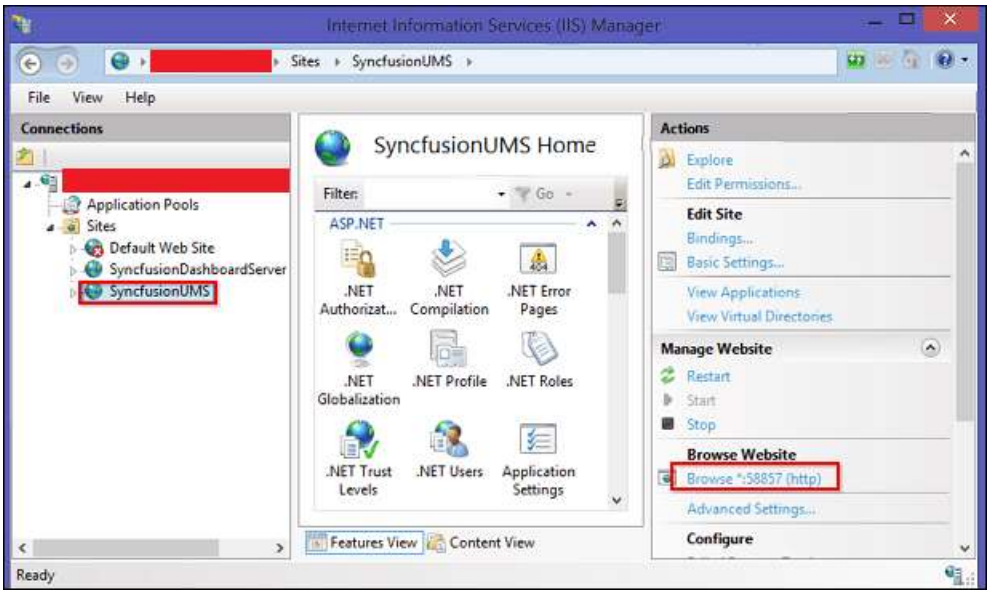
User Management Server can also be hosted in IIS by following the below steps.

1. Run the program ConfigureUserManagementServerIIS.exe from the following installed location to host the user management server in IIS

{Installed_ Location}\Syncfusion\UMS\Utilities\UserManagementServerIIS\ConfigureUserManagementServerIIS.exe



2. Type in a unused port for the User Management Server as like in the above image. This program will host the application in IIS and theUser Management Server can be opened from the browse button in the IIS.

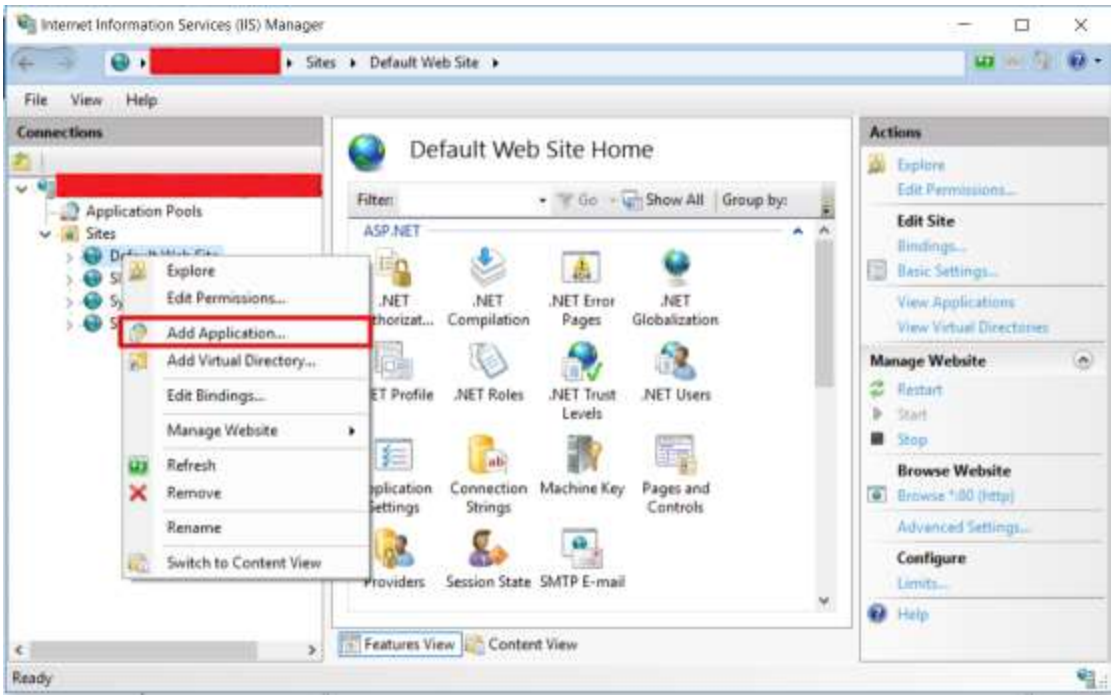


Host Dashboard Server as Application in IIS

Dashboard Server can also be hosted as Application in IIS by following the below steps.

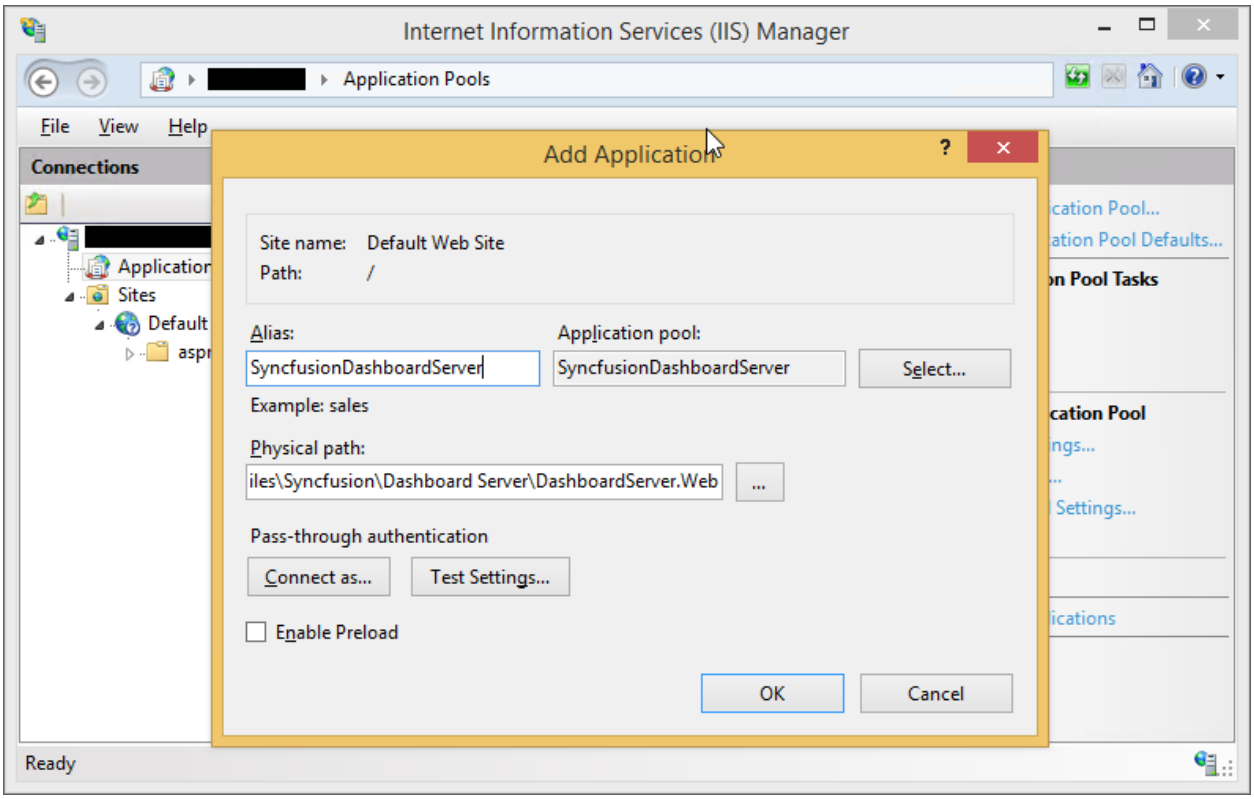
Add Dashboard Server as application

Right click the Website hosted in IIS and choose Add Application as below figure.



And Fill the following details as below figure

1. Alias name
2. Application pool
3. Physical path



Convert the sub folders as application

We have the following folders to be converted as application.

1. API
2. WebDesignerService

Right click the folder and choose Convert to Application as below figure

![Host Dashboard Server as application in IIS - Convert to sub Application](images/Convert to application.png)

SSL for Dashboard Server

To enable SSL for the Dashboard Server application, you will need a valid SSL certificate. Please check the below link on how to Obtain an SSL certificate and install it to a website in IIS.

http://www.iis.net/learn/manage/configuring-security/how-to-set-up-ssl-on-iis

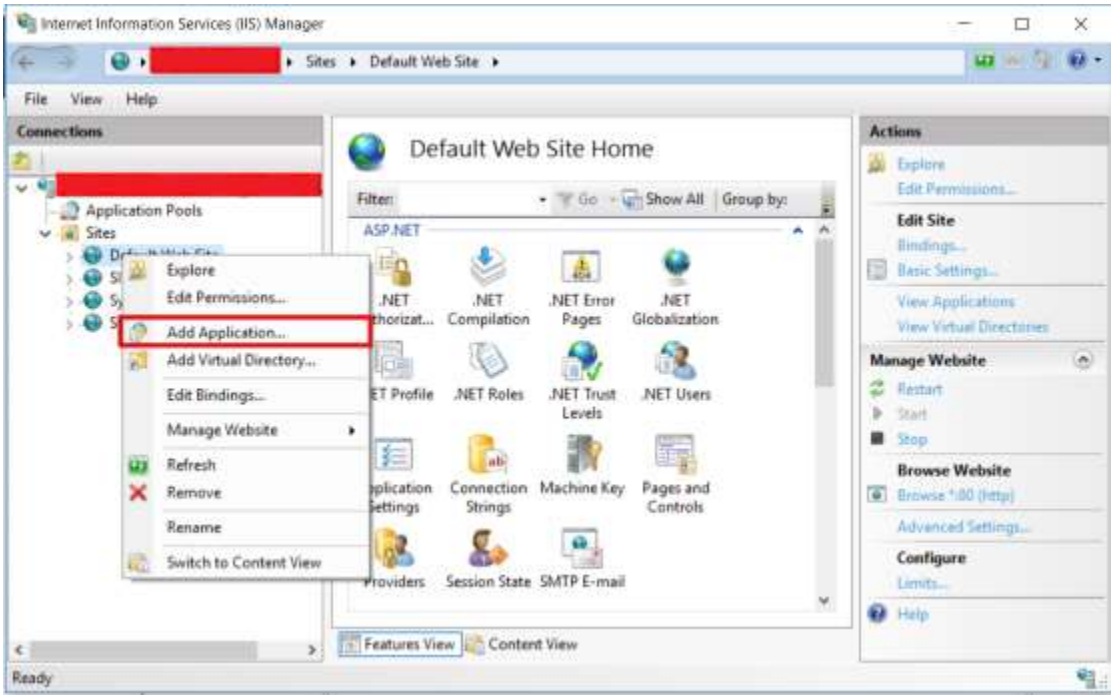
Note: If you want to access Dashboard Server from a different machine to the one it's installed on, use the URL http://machine*name:[dashboard*server*port*number] or http://machine*ip*address:[dashboard*server*port_number]

Host User Management Server as Application in IIS

User Management Server can also be hosted as Application in IIS by following the below steps.

Add User Management Server as application

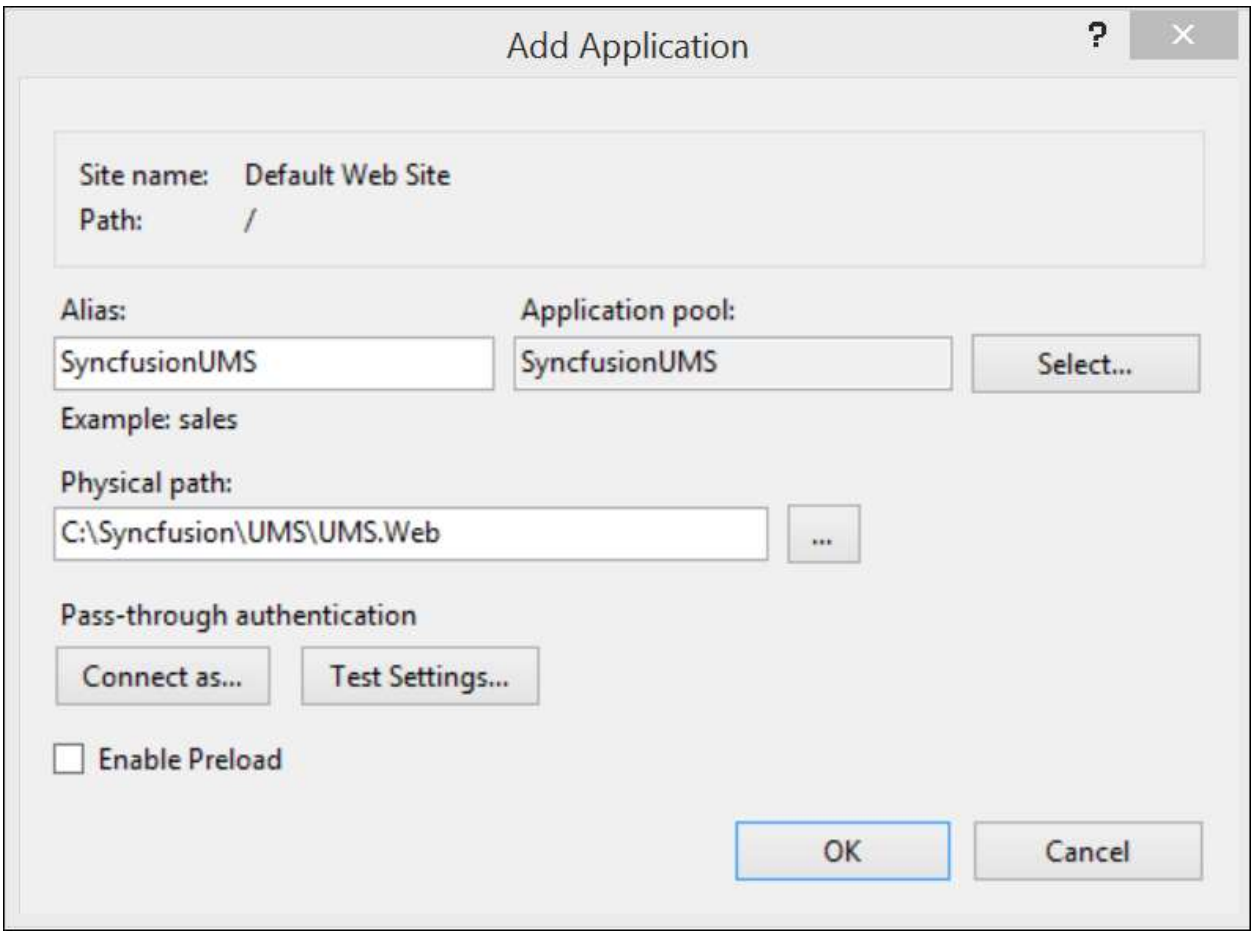
Right click the Website hosted in IIS and choose Add Application as below figure.



And Fill the following details as below figure

1. Alias name
2. Application pool

3. Physical path



Convert the sub folders as application

We have the following folders to be converted as application.

1. API
2. WindowsAuthentication

Right click the folder and choose Convert to Application as below figure

![Host User Management Server as application in IIS - Convert to sub Application](images/Convert to application UMS.png)

SSL for User Management Server

To enable SSL for the User Management Server application, you will need a valid SSL certificate. Please check the below link on how to Obtain an SSL certificate and install it to a website in IIS.

http://www.iis.net/learn/manage/configuring-security/how-to-set-up-ssl-on-iis

Note: If you want to access User Management Server from a different machine to the one it's installed on, use the URL http://machine*name:[user*management*server*port*number] or http://machine*ipaddress:[user*management*server*port_number]

Upgrading Syncfusion Dashboard Server from any version to 3.1

This section explains how to how to upgrade Syncfusion Dashboard Server from any version to 3.1.

Syncfusion releases several major versions in a year. Each version includes new features, bug fixes and other improvements.

Syncfusion Dashboard Server can be upgraded to version 3.1 at any time manually, and there are no automatic updates for Syncfusion Dashboard Server. Before upgrading, you can refer the features and enhancements from the Release Notes.

Upgrading Guidelines

Syncfusion recommends you to follow below guidelines while upgrading the Dashboard Server from an older version to latest version.

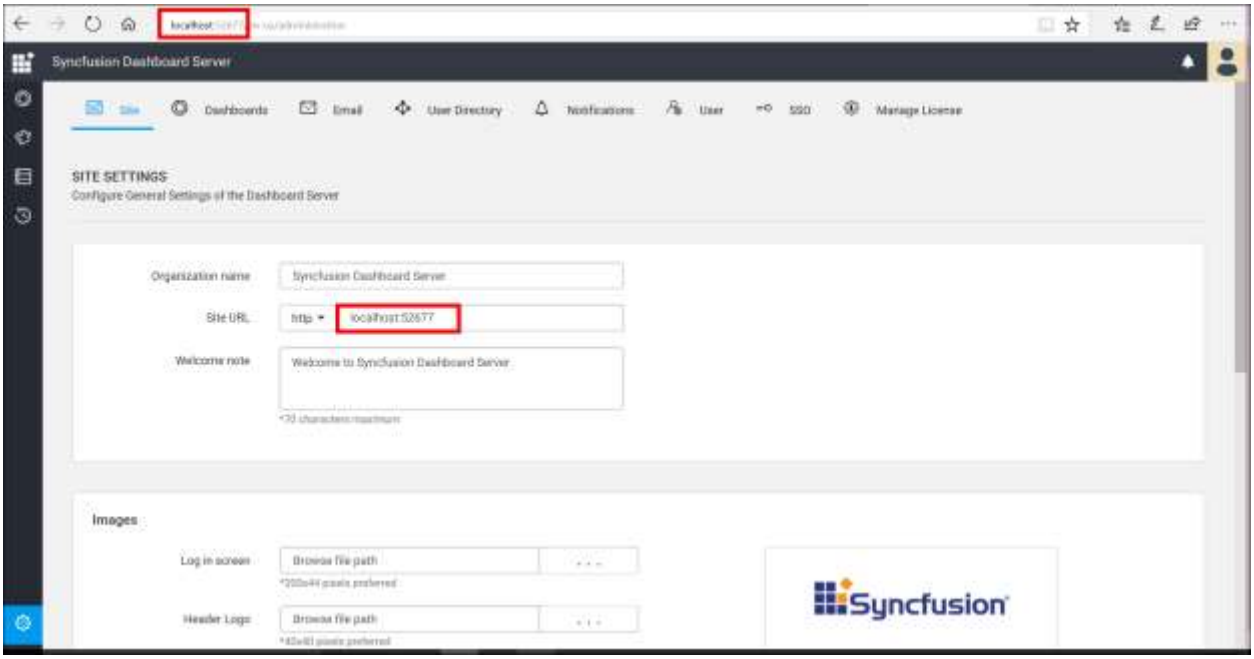
- Save all the open settings and the unsaved items.
- Ensure no one is currently working with dashboards.
- Inform about the maintenance time to the users.
- For SQL, MySQL, Oracle and Postgre SQL databases, make sure you have a valid network connection to the database while upgrading to the new version.
- Download the latest Syncfusion Dashboard Server from here.
- Follow the installation steps from the above section Installation.

The upgrade process will retain all the resources and settings from the previous installation.

Dashboard Server updates the database schema of your current version to the new version.

Please ensure the below steps if the dashboards are not rendered properly.

1. Ensure that the Site URL is updated properly as the hosted Syncfusion Dashboard Server URL.



2. If you have enabled SSL for Dashboard Server in IIS, it is must be enabled in Dashboard service also, Please follow the steps in the below link to enable the SSL in Dashboard service

https://help.syncfusion.com/dashboard-platform/dashboard-sdk/installation-and-deployment#configuring-ssl-for-dashboard-service

Upgrading Syncfusion Dashboard Server from any version to 3.2

This section explains how to upgrade Syncfusion Dashboard Server from any version to 3.2.

Syncfusion releases several major versions in a year. Each version includes new features, bug fixes and other improvements.

Syncfusion Dashboard Server version 3.2 comes with a User Management Server which is a separate application for managing your users and the applications. To know more about User Management Server click here.

Syncfusion Dashboard Server can be upgraded to latest version at any time manually, and there are no automatic updates for Syncfusion Dashboard Server. Before upgrading, you can refer the features and enhancements from the Release Notes.

Upgrading Guidelines

Syncfusion recommends you to follow below guidelines while upgrading the Dashboard Server from an older version to latest 3.2 version.

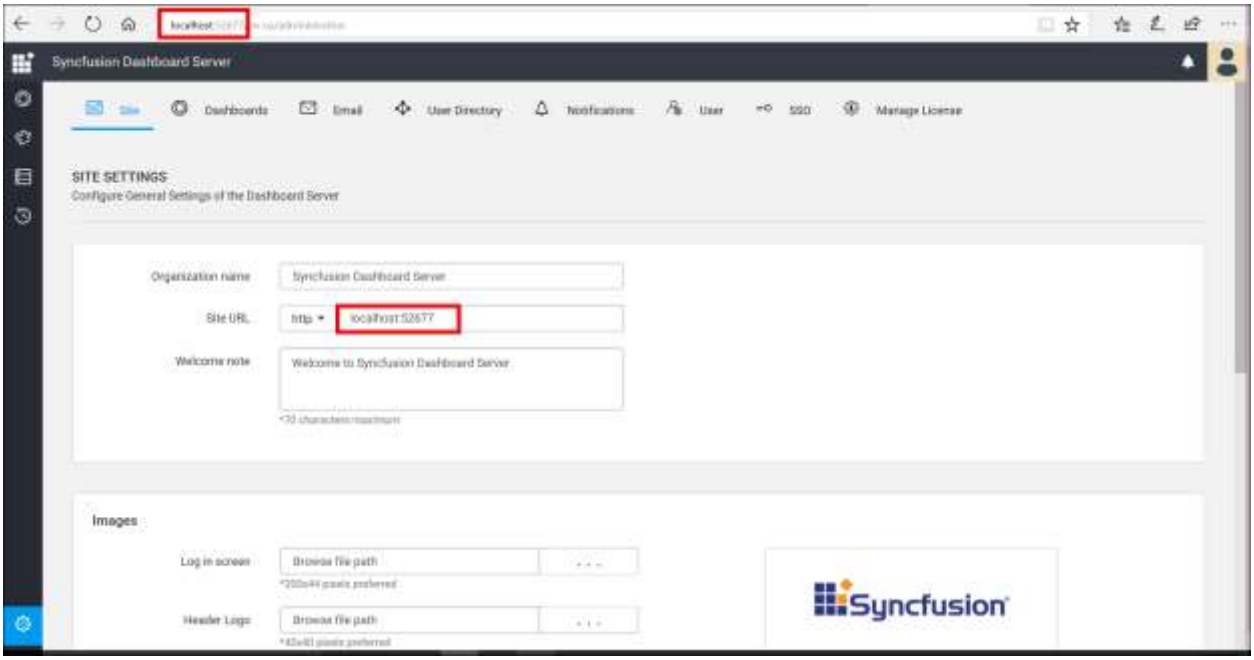
- Save all the open settings and the unsaved items.
- Ensure no one is currently working with dashboards.
- Inform about the maintenance time to the users.
- For SQL, MySQL, Oracle and Postgre SQL databases, make sure you have a valid network connection to the database while upgrading to the new version.
- Download the latest Syncfusion Dashboard Server version 3.2 from here.
- Follow the installation steps from the above section Installation.

The upgrade process will retain all the resources and settings from the previous installation.

Dashboard Server updates the database schema of your current version to the new version.

Please ensure the below steps if the dashboards are not rendered properly.

1. Ensure that the Site URL is updated properly as the hosted Syncfusion Dashboard Server URL.



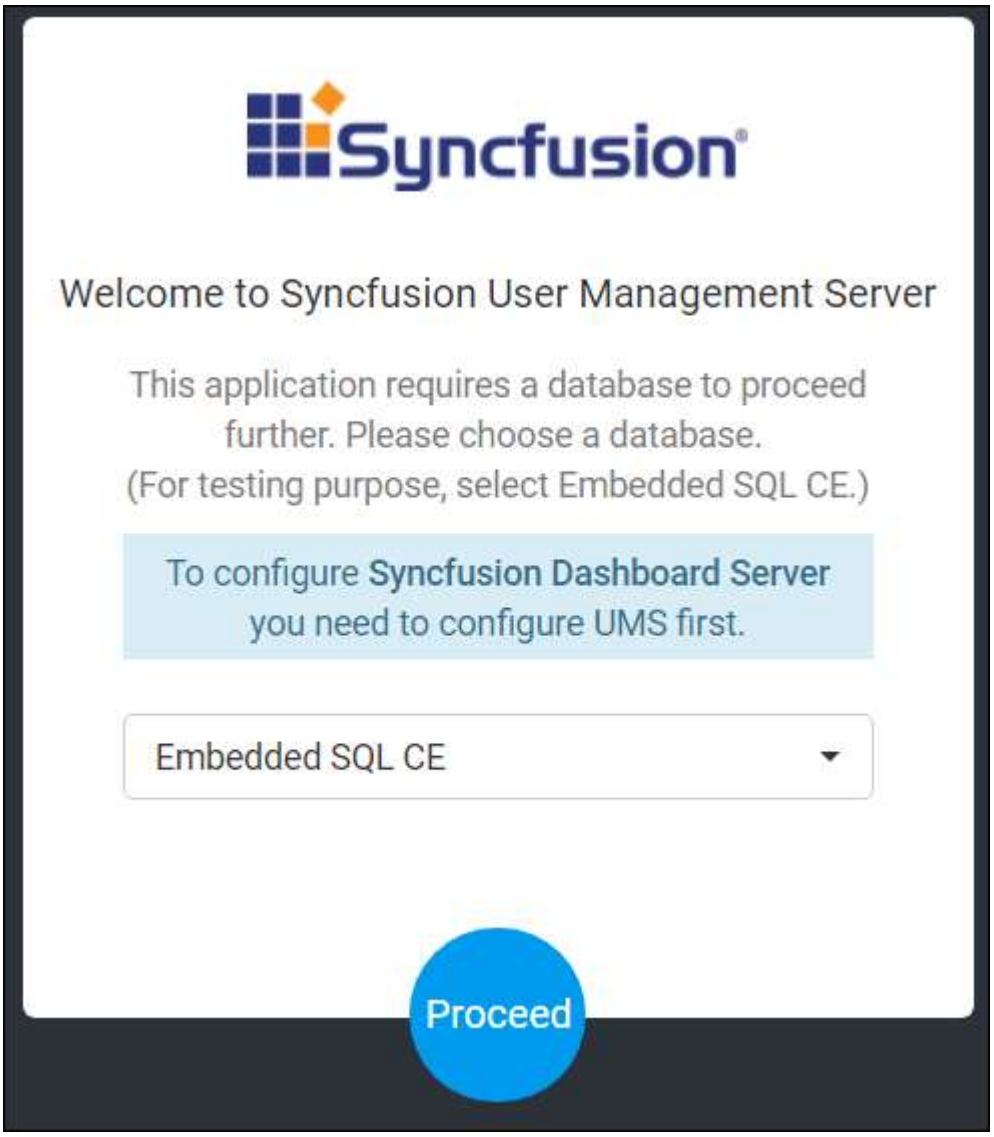
Dashboard Server with User Management Server configuration cases

1. New installation of Dashboard Server and User Management Server
2. Upgrade Dashboard Server with new User Management Server
3. Upgrade Dashboard Server along with User Management Server
4. Connect new Dashboard Server with existing User Management Server

New installation of Dashboard Server and User Management Server

After successful installation of Dashboard Server and User Management Server. Start the Dashboard Server.

To configure Dashboard Server, you must configure the User Management Server first.



Configure User Management Server by providing Storage Options, Storage System and System Administrator details and click proceed

After successful configuration of User Management Server, it will be redirected Dashboard Server startup.

You can find the details on How to configure User Management Server here

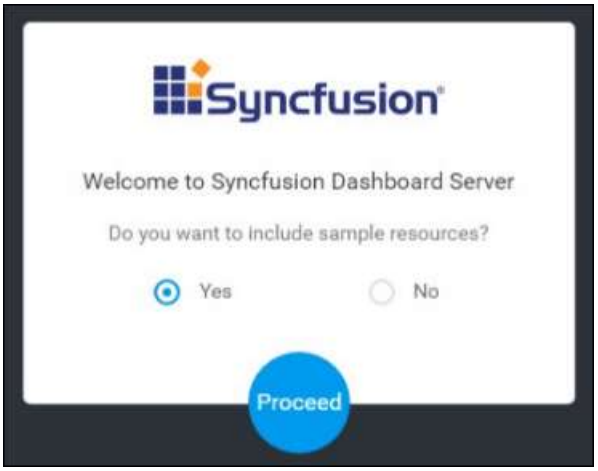
Note: An application of type Dashboard Server will be created automatically in User Management Server and System administrator group for Dashboard server will be created in User Management Server.

Use same database credentials of UMS option will use the same database credentials of User Management Server except the database name.

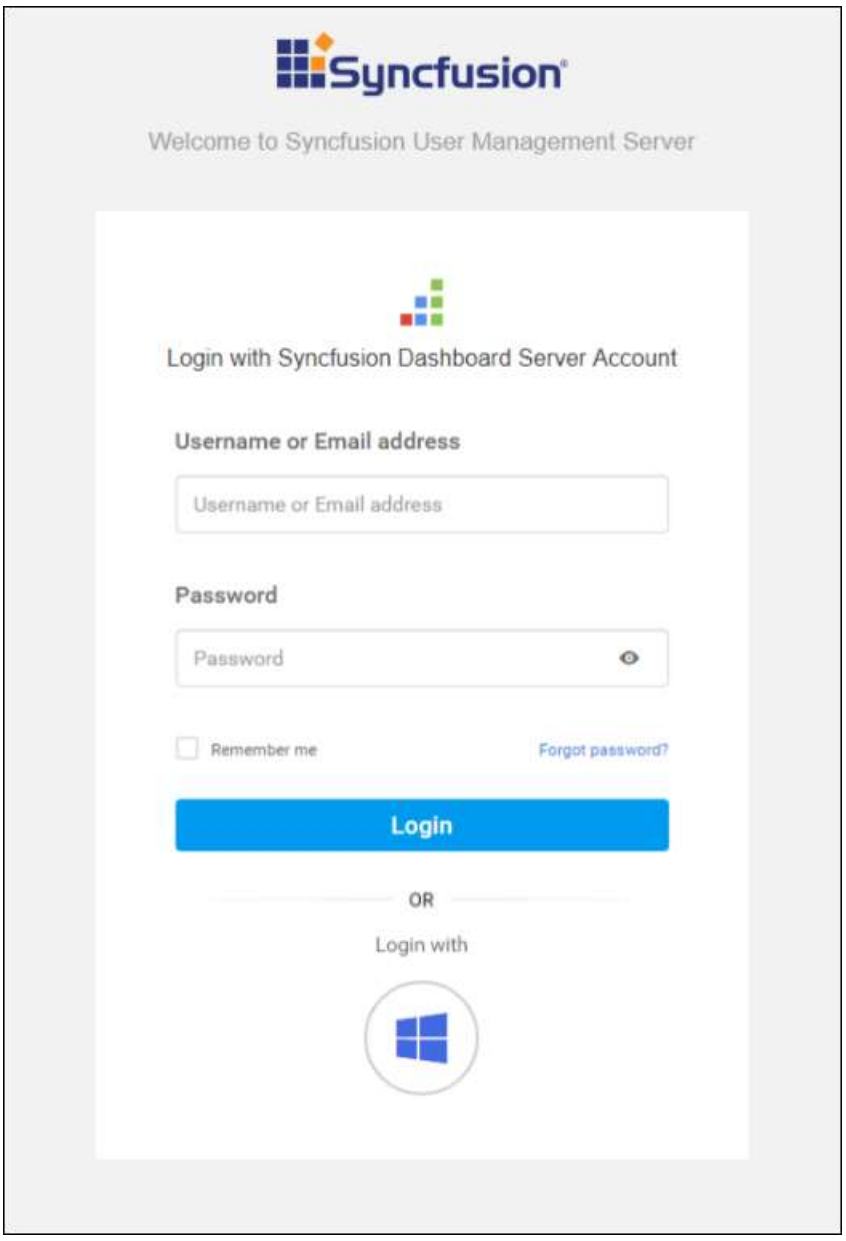
Configure Dashboard Server by providing Storage Options, Storage System.

You can find the details on How to configure Dashboard Server here

Choose option to startup with or without including the sample resources and click proceed.



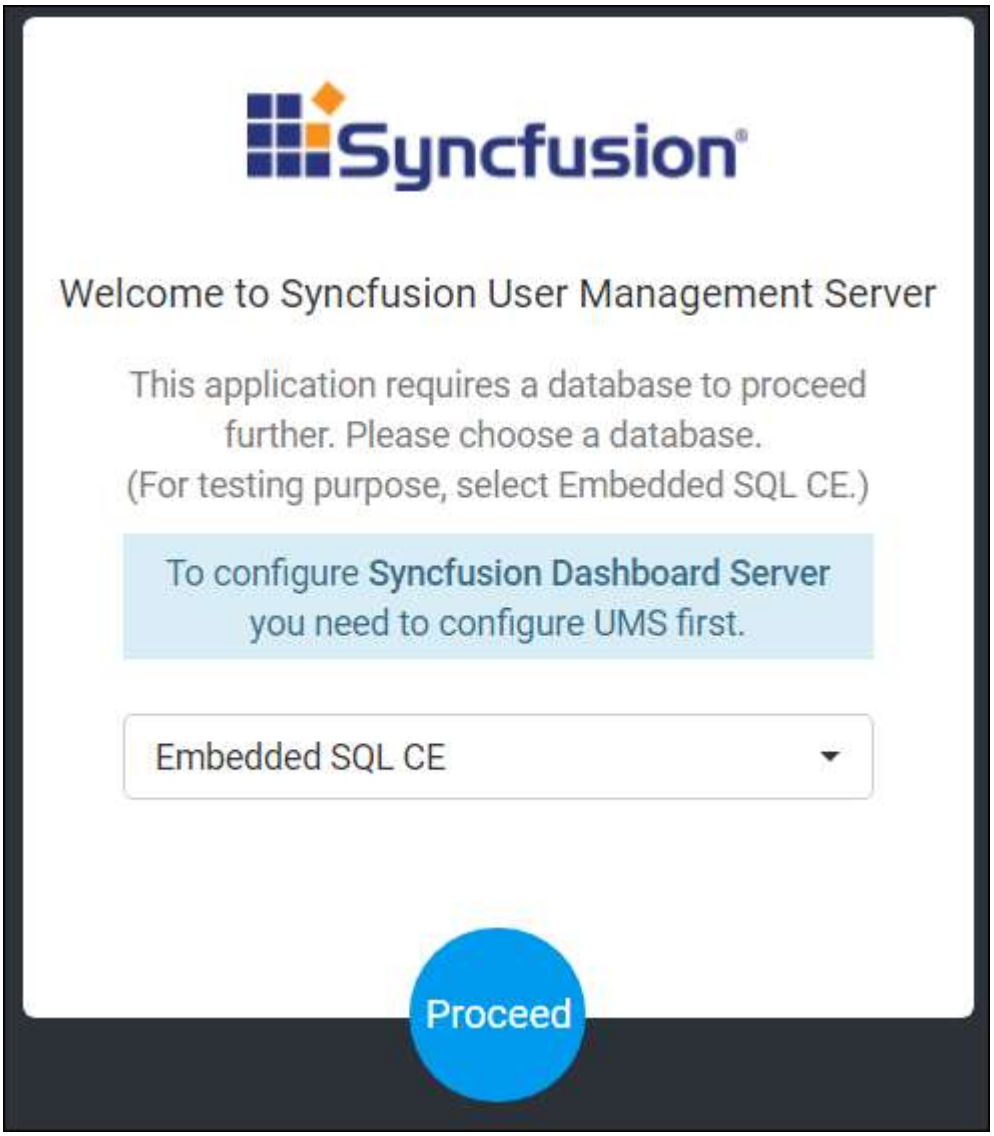
After successful configuration of Dashboard Server, it will be redirected to login page.



Note: Dashboard Server system administrator will be automatically added from User Management Server.

Upgrade Dashboard Server with new User Management Server

After successful upgrade of Dashboard Server and installation of User Management Server. You must configure the User Management Server first to run the Dashboard Server.

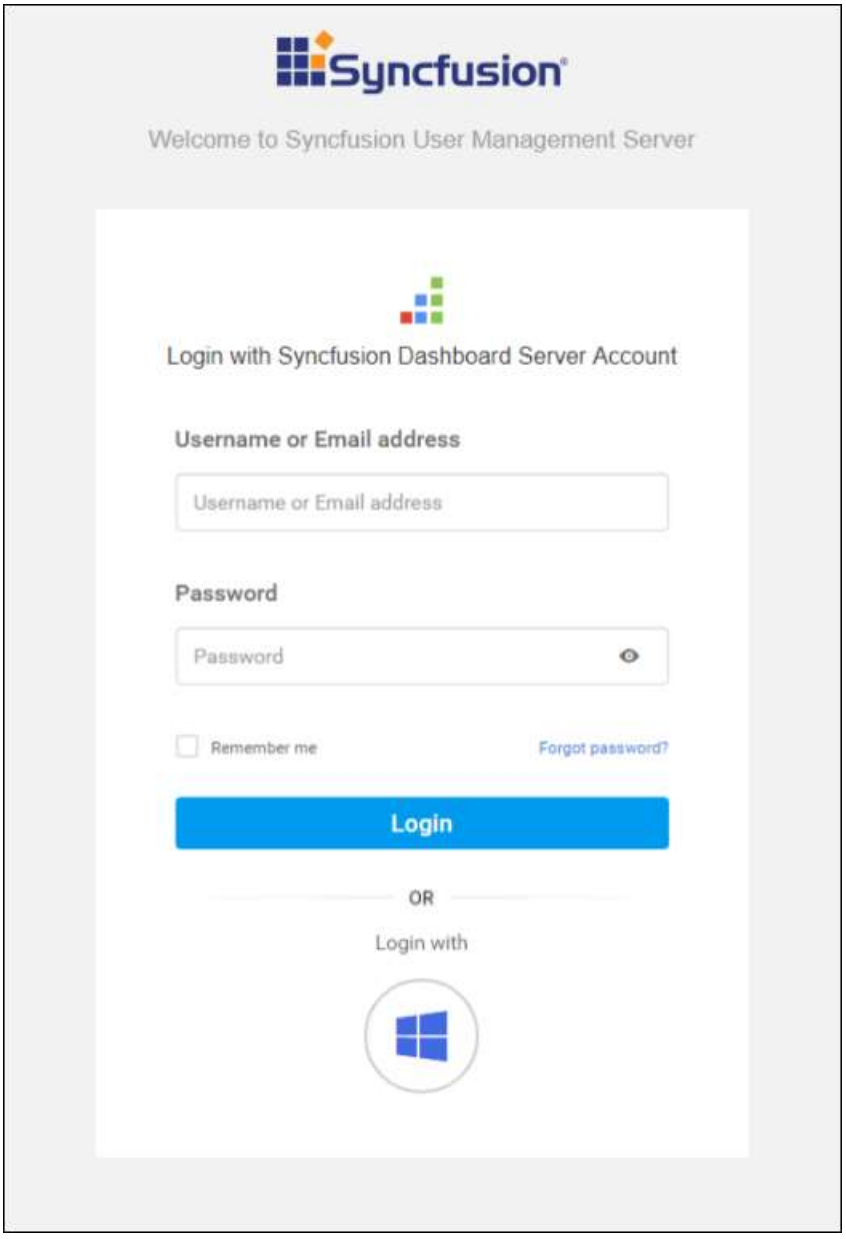


Configure User Management Server by providing Storage Options, Storage System. You can find the details on How to configure User Management Server here

Note: Users and groups of Dashboard Server will be automatically migrated to User Management Server.

Note: An application of type Dashboard Server will be created automatically in User Management Server and by default, system administrators of Dashboard Server will be the system administrators of User Management Server.

After successful configuration of User Management Server, it will be redirected to login page.



Upgrade Dashboard Server along with User Management Server

Note: To connect to the Syncfusion Dashboard Server with the User Management Server you must create a application on User Management Server. You can find the details on how to create Application in User Management Server [here]()

After successful upgrade of Dashboard Server along with User Management Server. Start the Dashboard Server.

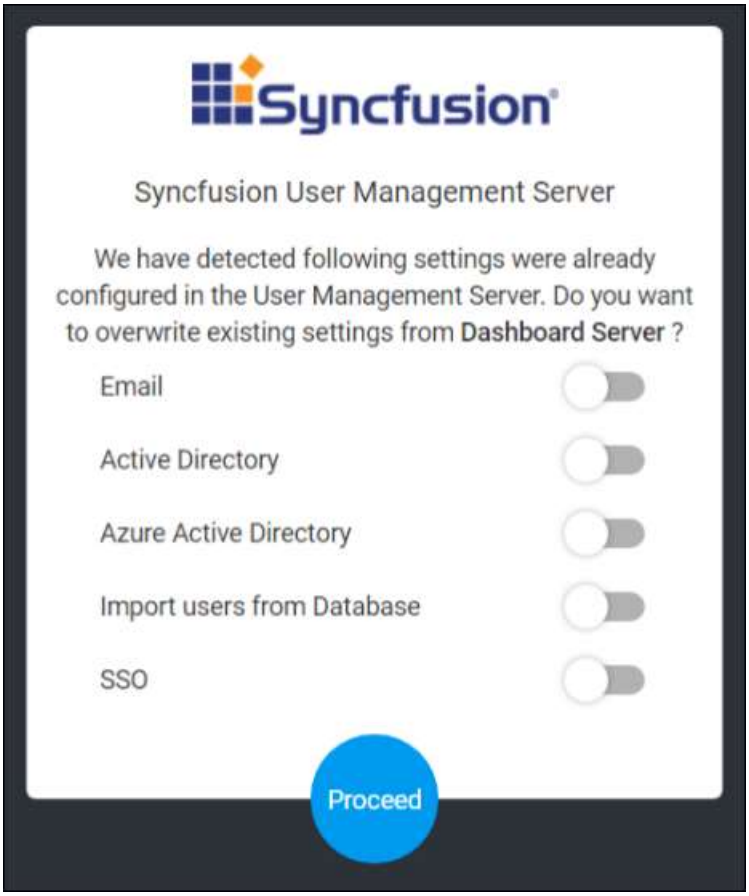
Enter the User Management Server URL, Client Id, Client Secret of the Dashboard Server application that is created on User Management Server.

Application will check for any system settings conflict and users conflict while migrating Data from Dashboard Server to User Management Server.

Resolve settings conflict

If both User Management Server and Dashboard Server application having settings saved. Then application will ask for which settings to retain as like below,

Selected settings will be migrated from Dashboard Server to User Management Server and existing settings of User Management Server will be removed.



Note: This page will not be shown, if there is no settings conflict.

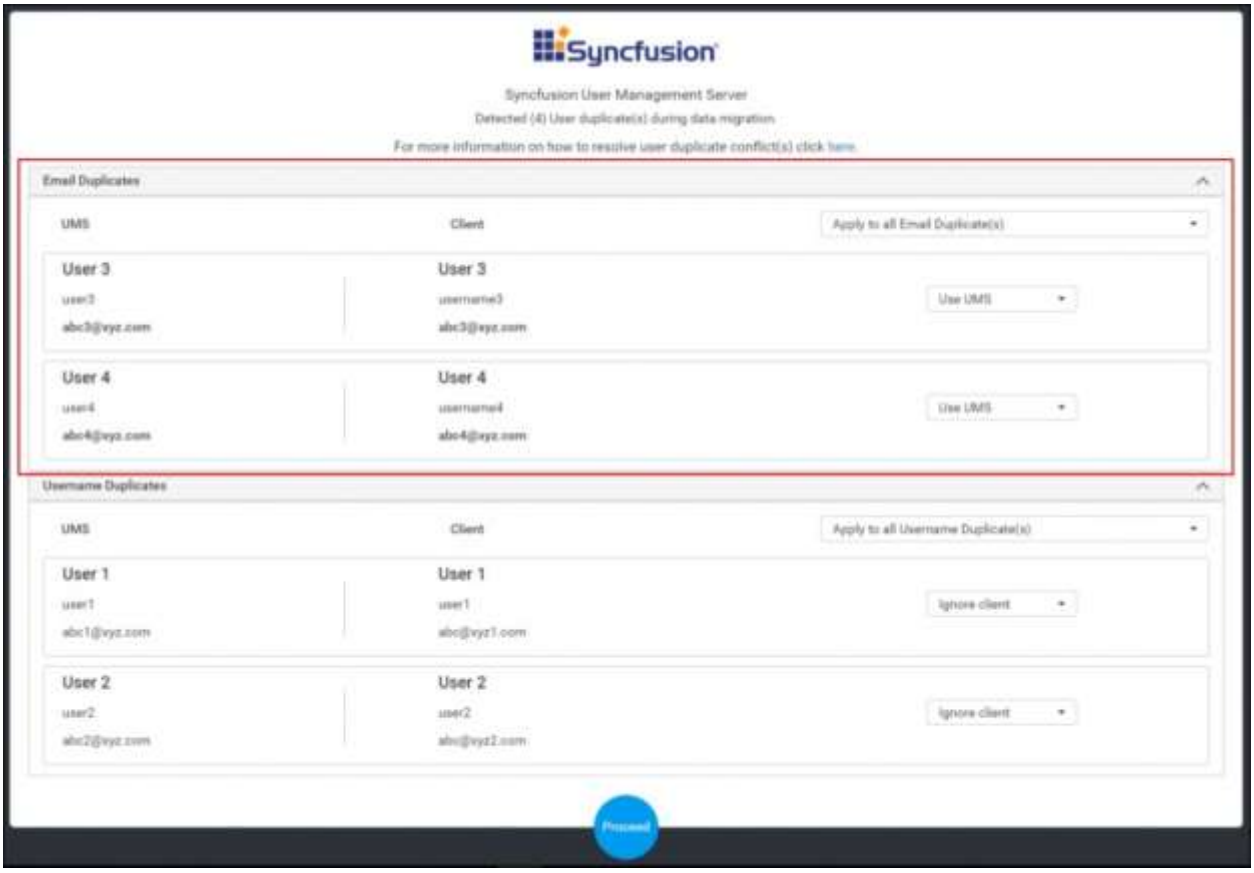
Resolve users conflict

After resolving settings conflict if any, application will check for any user conflicts between User Management Server and Dashboard Server.

If users conflict exists, it will be resolved manually. User conflict may arise with the below two possibilities,

1. Same Email address for user in User Management Server and Dashboard Server

If email address is same, you can able to retain either User Management Server user or Dashboard Server user.



Click on the dropdown menu besides the client user to choose which user you want to retain.

- Use UMS option will retain User Management Server user and ignore the client user.
- Use client option will retain client user and ignore the User Management Server user.

Apply global option will apply the selected option for all users that are listed under email duplication.

2. Same username for a user in User Management Server and Dashboard Server

If username is same, you can able to ignore client or edit ums user or Dashboard Server user.

Click on the dropdown menu besides the client user to choose which user you want to ignore or edit.

- Ignore client will ignore the client user.

- Edit UMS option will enable editing User Management Server username as like below,

- Edit client option will enable editing client username as like below,

- Keep both option will generate a random digit and appends to the username.

Note: If Edit UMS, Edit client, Keep both options are selected, both User Management Server user and client user will be retained.

Apply global option will apply the selected option for all users that are listed under username duplication.

After resolving user duplicates, click proceed.

Resolved users will be migrated and it will be redirected to login page.

Connect new Dashboard Server with existing User Management Server

Note: Latest version of User Management Server is mandatory to run Syncfusion Dashboard Version 3.2

Note: To connect to the Syncfusion Dashboard Server with the User Management Server you must create a application on User Management Server. You can find the details on how to create Application in User Management Server [here]()

After successful installation of Dashboard Server. Start the Dashboard Server.

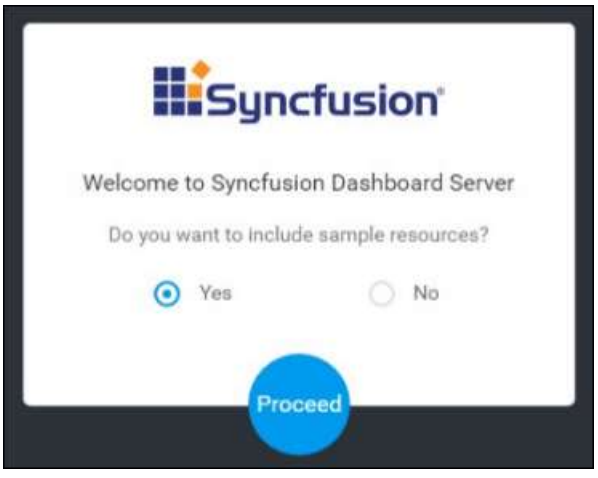
Enter the User Management Server URL, Client Id, Client Secret of the Dashboard Server Application that is created on User Management Server.

Configure Dashboard Server by providing Storage Options, Storage System.

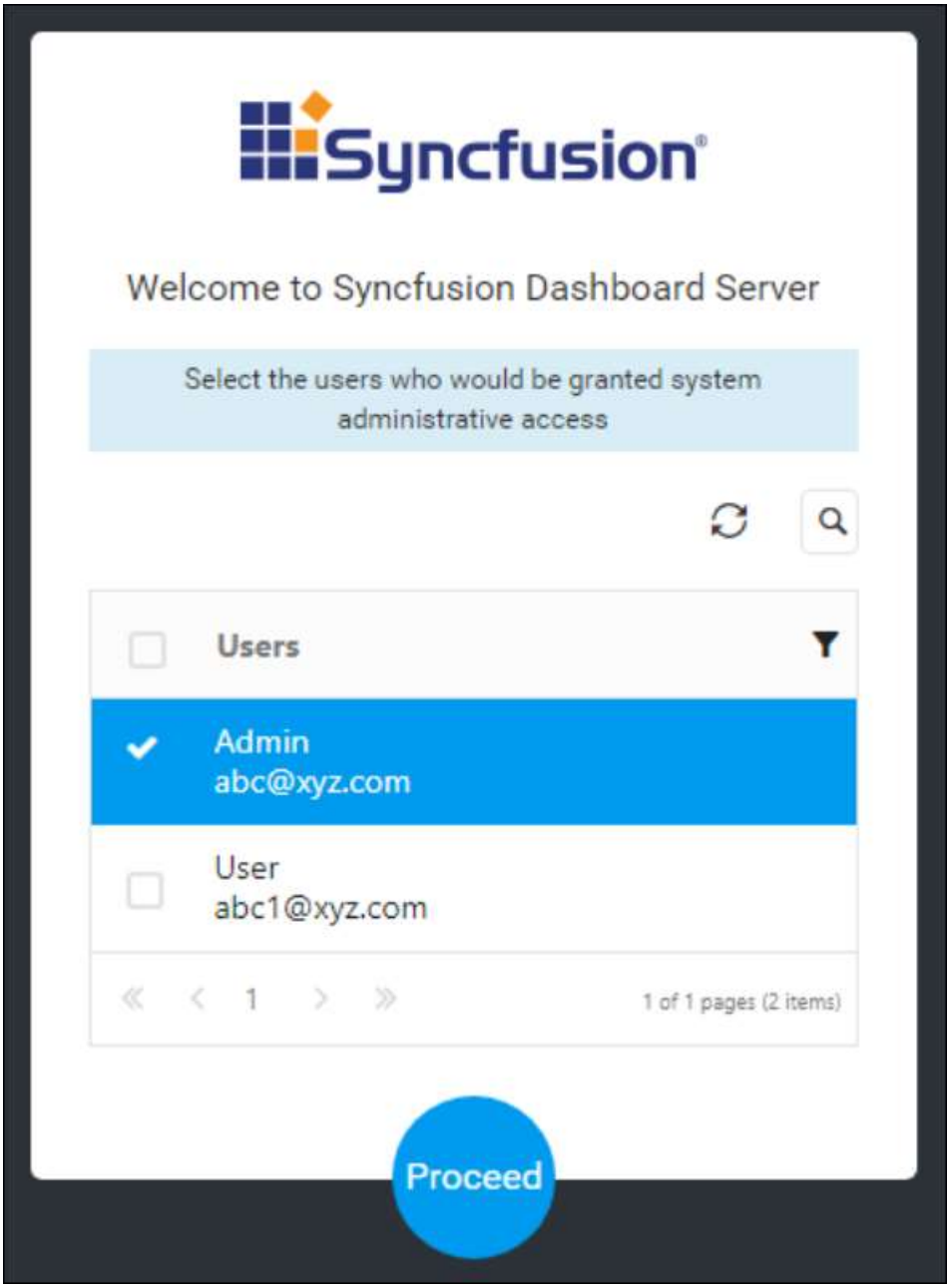
You can find the details on How to configure Dashboard Server [here]

(/dashboard-platform/dashboard-server/application-startup)

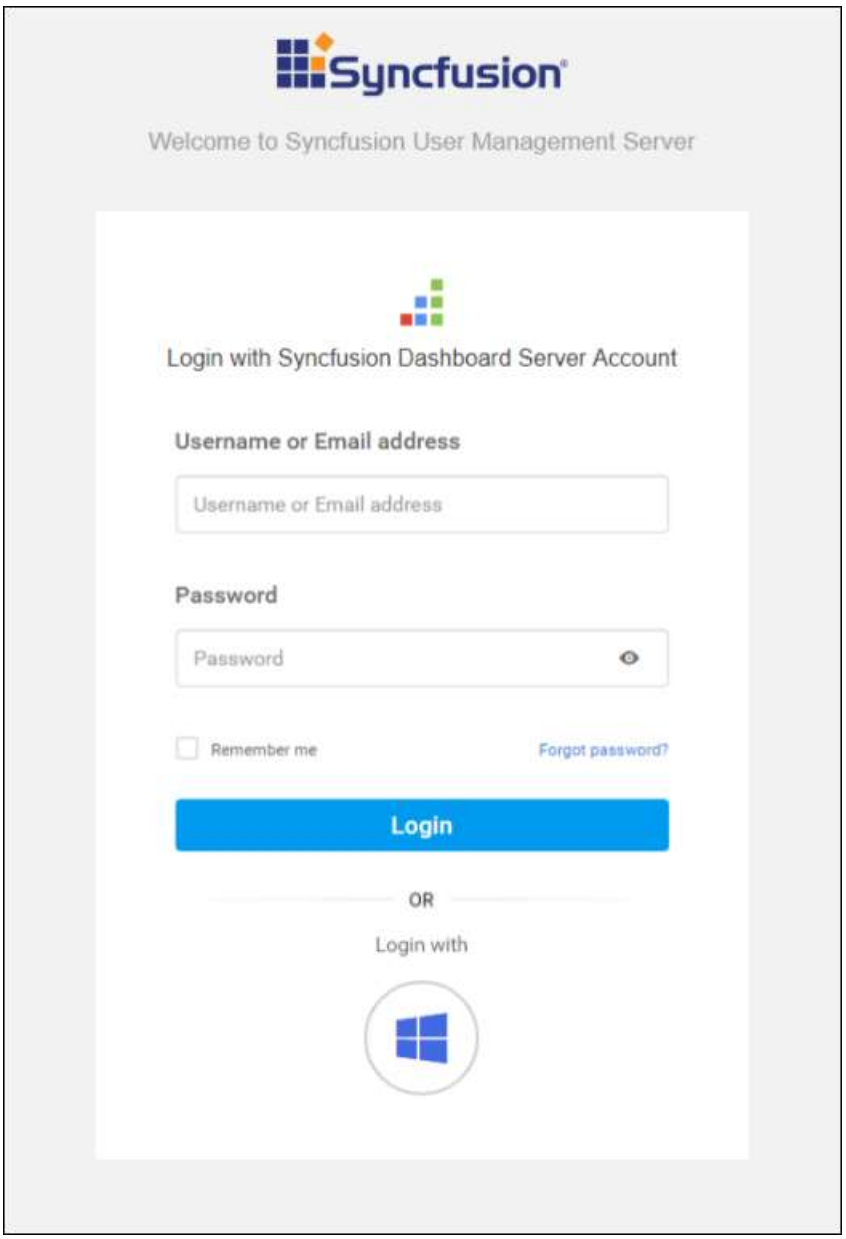
Choose option to startup with or without including the sample resources and click proceed.



Select the administrator users of Dashboard server and click proceed.



After successful configuration of Dashboard Server, it will be redirected to login page.



User Management Server

Azure Deployment

App Service

Dashboard Server and User Management Server Azure App Service Deployment by ARM template

App Service is a Platform as a Service (PaaS) offering of Microsoft Azure. Create the web and mobile apps for any platform or device. Integrate apps with Software as a Service (SaaS) solutions, connect with on-premises applications, and then automate the business processes. Azure runs the apps on fully managed virtual machines (VMs) with the choice of shared VM resources or dedicated VMs. To know more about Azure App Service, click here.

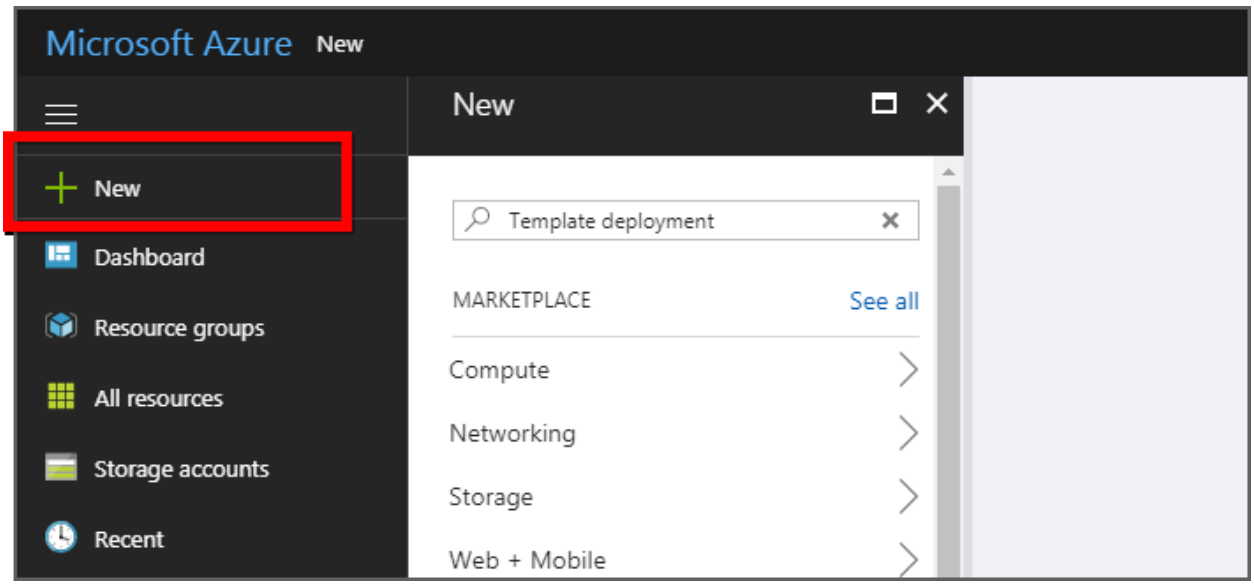
This section explains about how to deploy the latest Dashboard Server version 3.2 in Azure App Service using Azure Resource Manager (ARM) templates.

Create new Dashboard Server Azure App Service

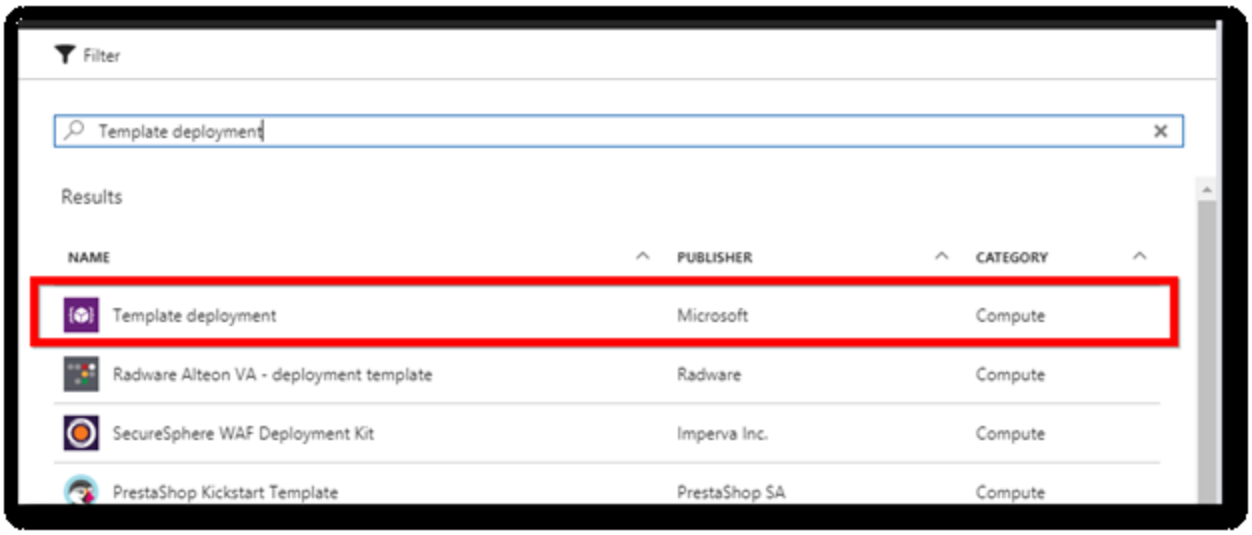
Follow the given steps to create the Syncfusion Dashboard Server Azure App Service using ARM template:

The ARM template is a JavaScript Object Notation (JSON) file that defines one or more resources to deploy to a resource group. It also defines the dependencies between the deployed resources. The template can be used to deploy the resources consistently and repeatedly, and it is based on the declarative syntax. To know more about ARM template, click here.

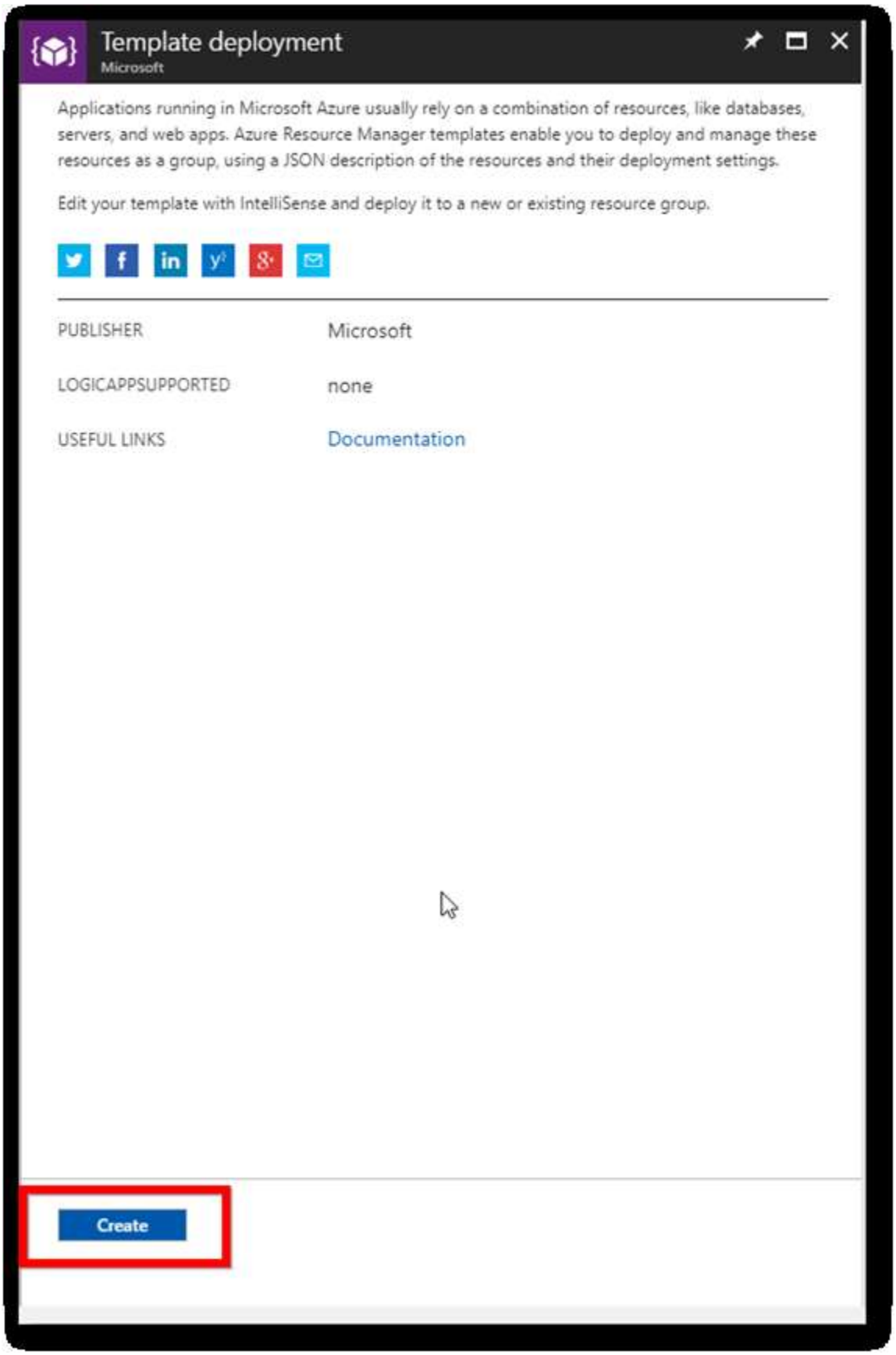
1. Login to Azure portal: .
2. Click New in the left menu to create new resources in the Azure portal.



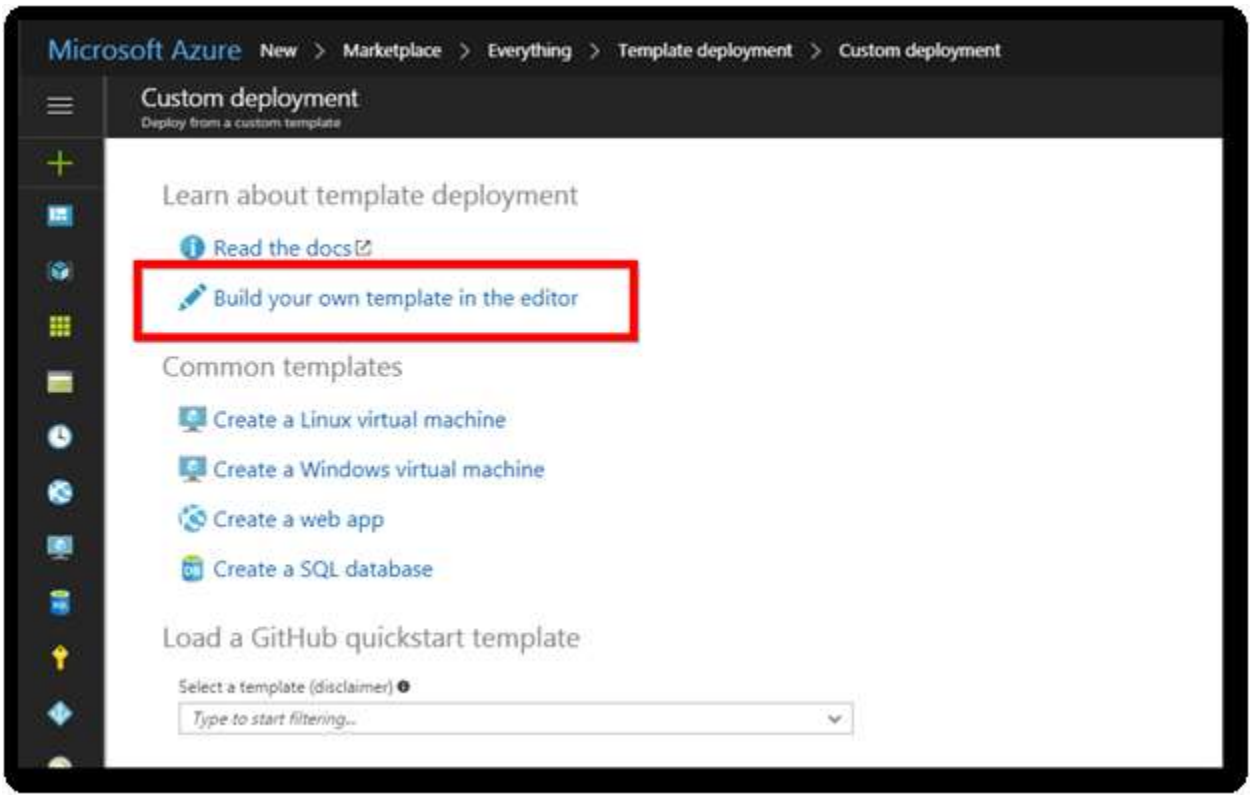
3. Search "Template deployment" in the marketplace and select it. The template deployment option allows you to create a customized template that defines the infrastructure and dependencies of your resources. To learn more about template deployment, click here.



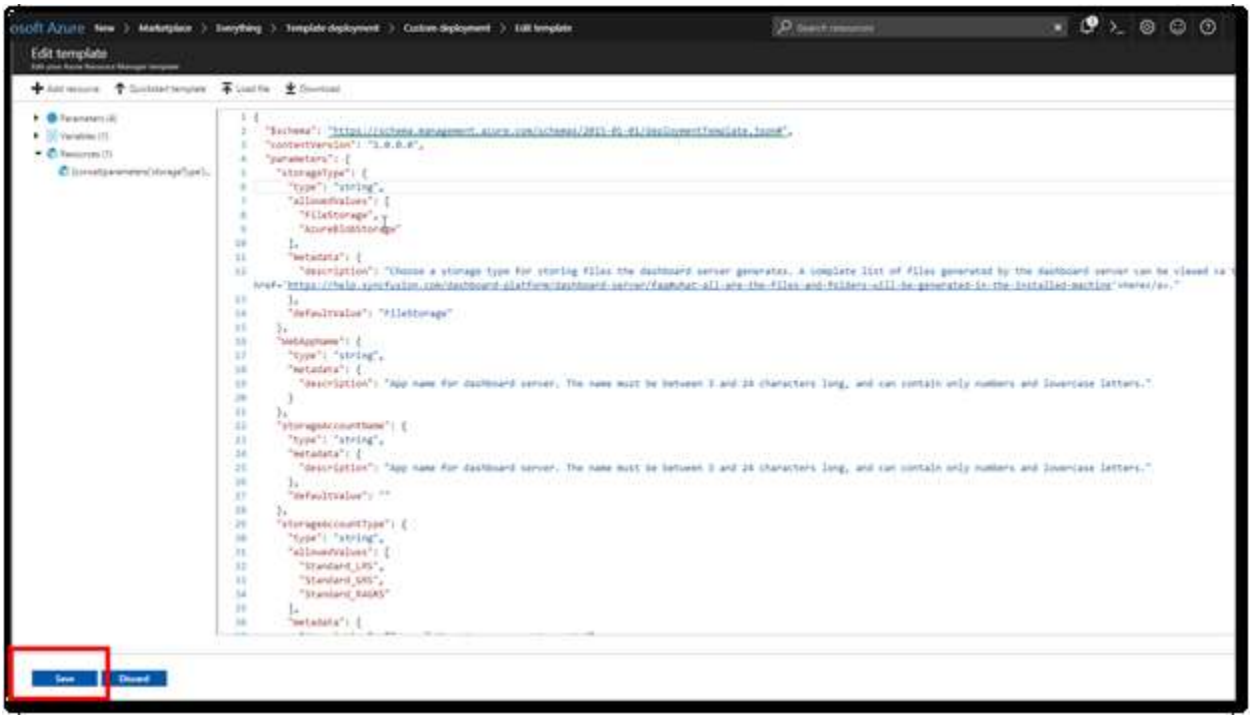
4. Click the create.



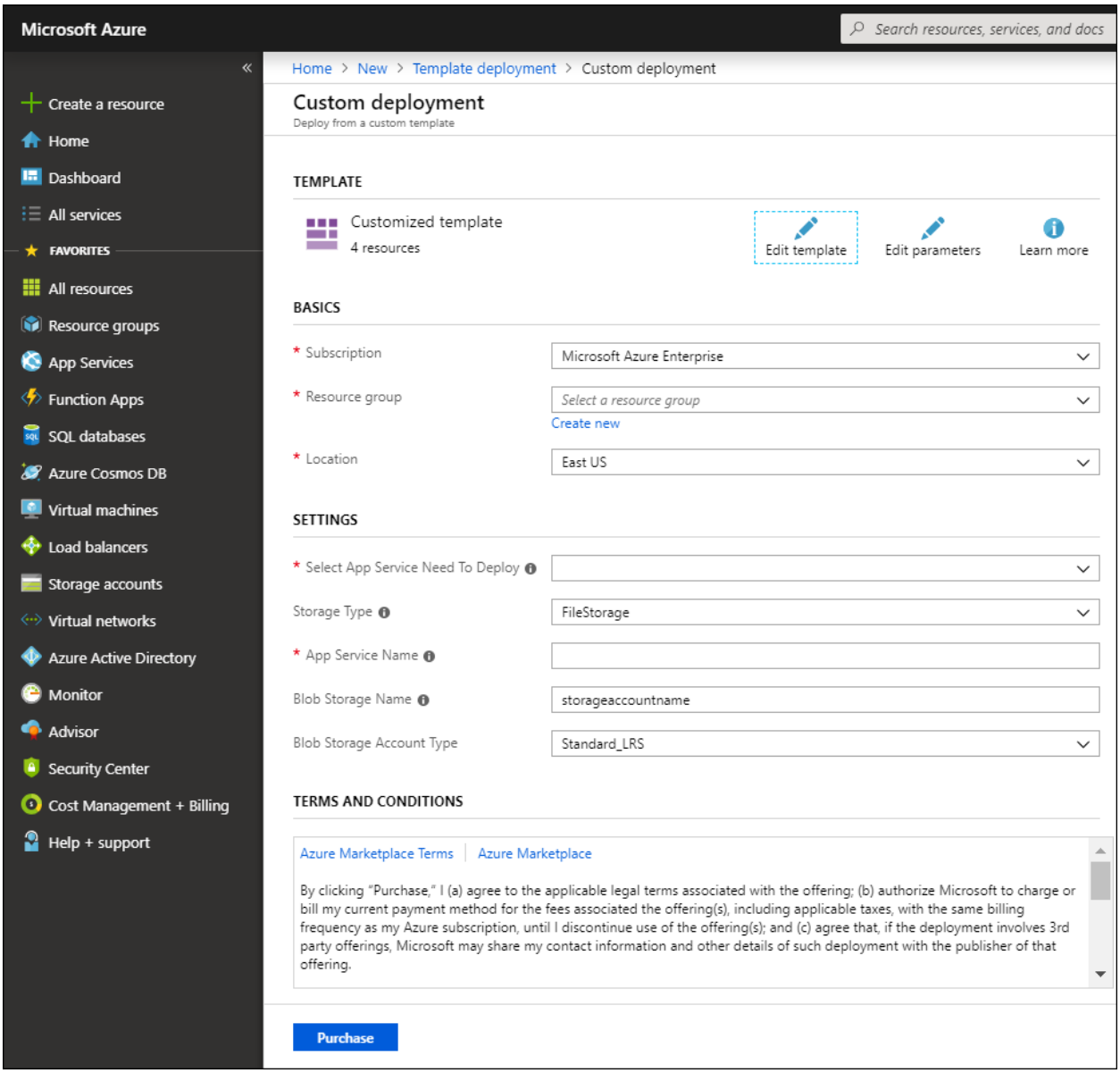
5. Select “Build your own template in the editor” in the "Custom deployment" blade. Now, the "Edit template" blade opens. It is a blank template available for customizing.



6. Click here to download the ARM template file.
7. Copy all the contents of the template file and replace them in the "Edit template" window. Then, click save.

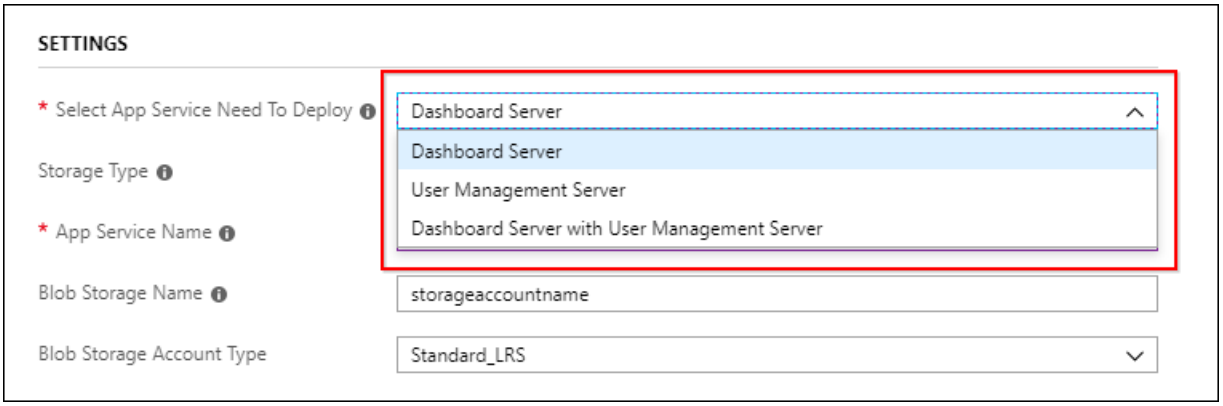


8. Fill the form as instructed below to complete the deployment steps.



- **Subscription**: Choose the subscription that you have with Azure. Learn more about subscription from here.
- **Resource group**: This is a logical group in Azure to group your resources such as a web app, storage account, network, etc. To learn more about resource groups, click here.
- **Location**: Choose the location at which the app to be deployed. “East US” is the recommended location.
- **Select App Service Need To Deploy**:

Note: It is mandatory to have latest User Management Server to run latest Dashboard Server version 3.2

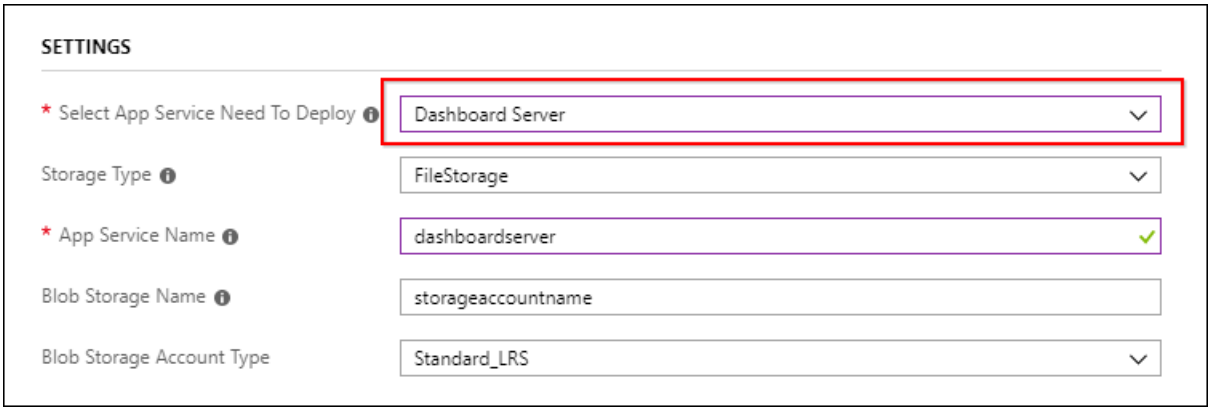


Latest 3.2 version of Dashboard Server comes with the User Management Server. ARM template deployment provides below three options to deploy app service,

- Deploy Dashboard Server individually.
- Deploy User Management Server individually.
- Deploy Dashboard Server along with the User Management Server.

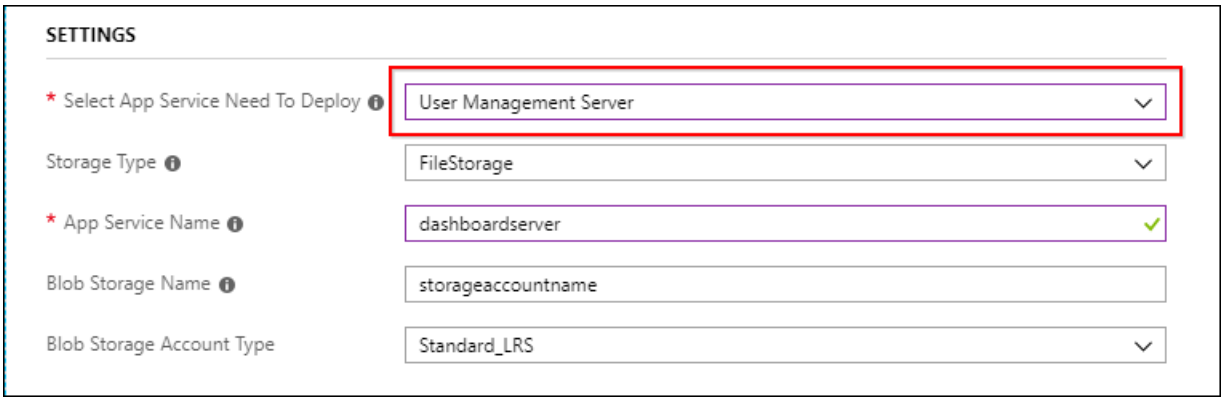
Deploy Dashboard Server individually

This option is preferred when User Management Server is already deployed or running on remote machine and this will deploy Dashboard Server alone with the given App service name inside the resource group.



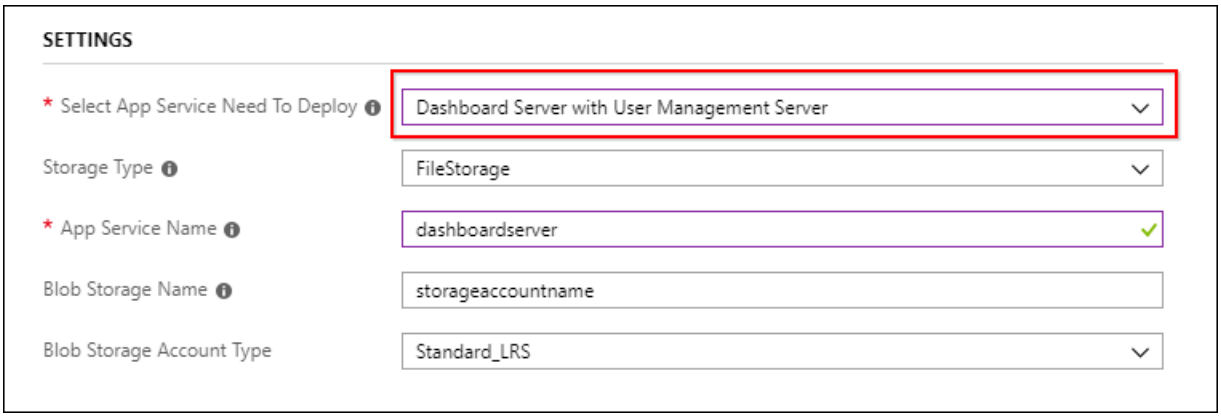
Deploy User Management Server individually

This option is preferred when Dashboard Server is already deployed or running on remote machine and this will deploy User Management Server alone with the given App service name inside the resource group.



Deploy Dashboard Server along with the User Management Server

This option will deploy both Dashboard Server and User Management Server at a single time.



Note: This option will create two app services. One for Dashboard Server and another one for User Management Server. App Service for User Management Server can be created automatically using the App Service name given during deployment

Dashboard Server will be deployed under {GivenAppServiceName} and User Management Server will be deployed under { GivenAppServiceName-**ums**}

- **Storage Type**: The Syncfusion Dashboard Server and User Management Server stores the resources in the file storage or in the blob storage. Choose the storage type for storing files generated by the Dashboard Server and User Management Server.

A complete list of files generated by the dashboard server can be viewed here. (For upgrade, select file storage) and User Management Server can be viewer here.

Note: While deploying Dashboard Server along with the User Management Server with the Storage type **AzureBlobStorage** selected, both Dashboard Server and User Management Server containers are created under given Blob Storage Name.

Dashboard Server container name – **syncfusiondscontainer**

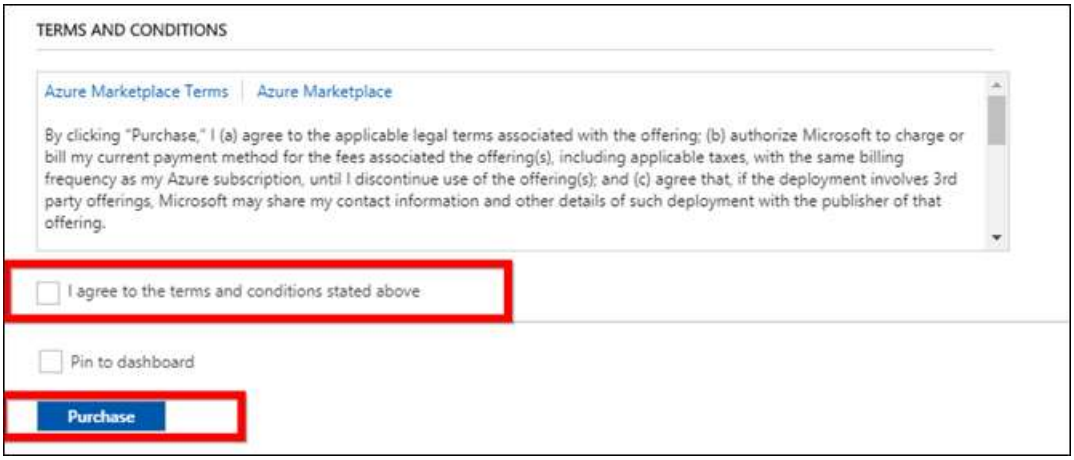
User Management Server container name - **syncfusionumscontainer**

- **Web App name**: This is the name of the Dashboard Server and User Management Server to be provided in the URL; it should be between 3 to 24 characters long, contain only numbers and

lowercase letters, and be globally unique. Deployment process will be failed if this had been presented already. Then, you will have to start once again with another name.

Note: Dashboard Server will be deployed under {GivenAppServiceName} and User Management Server will be deployed under { GivenAppServiceName-**ums**}.

- **Storage account name**: This is **optional** if the file storage had been chosen in storage type and should be between 3 to 24 characters long, contain only numbers and lowercase letters, and should be mandatory for the blob storage. This must be unique as that of the Web App name. Learn more about storage accounts by clicking here.
- **Storage account type**: This is **optional** if the file storage had been chosen in storage type, and this is mandatory for the blob storage. Learn more about storage account types by clicking here.
- Click the agreement checkbox and select the purchase to deploy the Syncfusion Dashboard Server web app.



9. Now, the Syncfusion Dashboard Server App Service (web app) deployment will be started.
10. An App Service Plan is created for the web app which will be in “Basic – B1”, by default. To learn more about App Service Plans, click here. Syncfusion Dashboard Server and User Management Server web app does not support free or shared App service plans.

Syncfusion Dashboard Server and User Management Server supports basic, standard, and premium App Service Plans in the Azure. The minimum recommended App Service Plan to run the application is the basic plan.

To get better performance, you can scale up the App Service Plan from basic to standard or premium plans. To learn about how to scale up and scale out the App Service Plan, refer to the following documentation links:

Scale up: https://docs.microsoft.com/en-us/azure/app-service-web/web-sites-scale

Scale out: https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/insights-how-to-scale

Migration to latest Syncfusion Dashboard Server version 4.1 along with User Management Server

Syncfusion Dashboard Server version 4.1 comes with a User Management Server which is a separate application for managing your users and the applications. To know more about User Management Server click here.

Migration to latest Syncfusion Dashboard Server version 4.1

Follow the below steps to upgrade the Dashboard Server App Service to the latest.

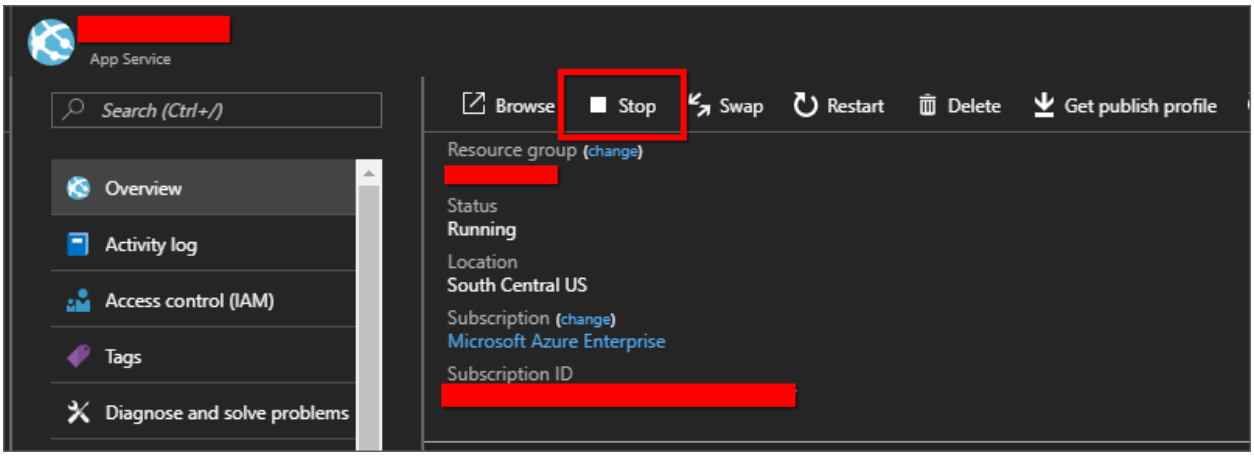
- Stop Syncfusion Dashboard Server Azure App Service.
- Upgrade Dashboard Server to the latest release version.
- Start Syncfusion Dashboard Server App Service.
- Restart WebJobs.

Stop Syncfusion Dashboard Server Azure App Service

Before starting the upgrade/update process, the Dashboard Server App Service should be stopped first.

Follow the below steps to stop the Dashboard Server App Service.

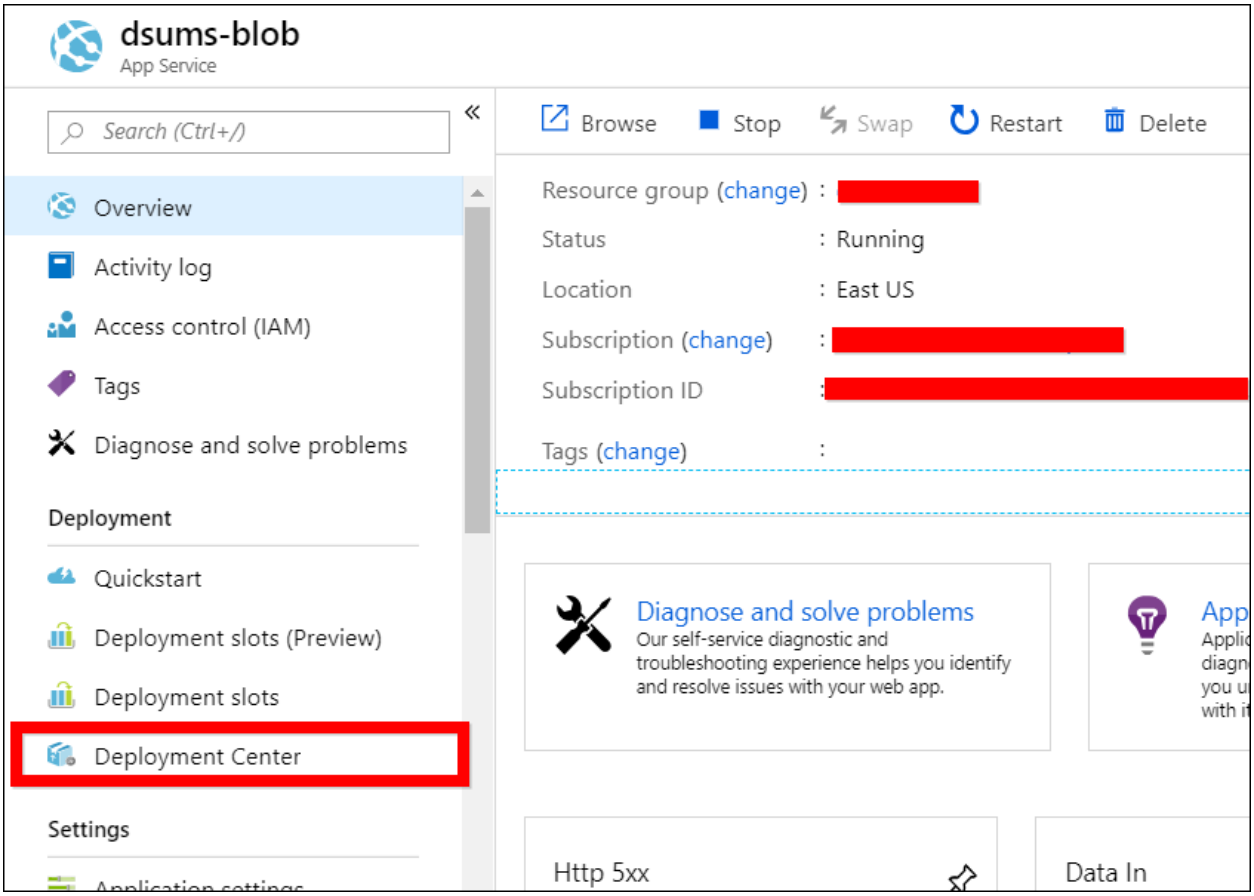
- Login to Azure portal: .
- Select App Services in Microsoft Azure Services.
- Choose the Dashboard Server Azure App Service.
- In the **Overview** section, click the **Stop**.



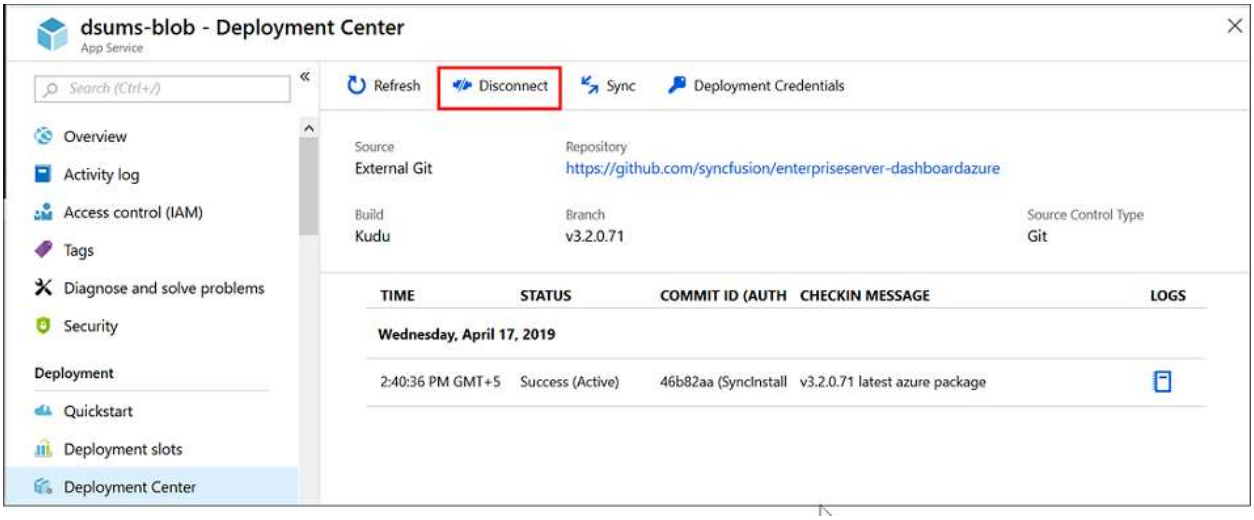
Upgrade Dashboard Server to the latest release version

Follow the below steps to configure the existing Azure blob storage details to the latest Dashboard Server:

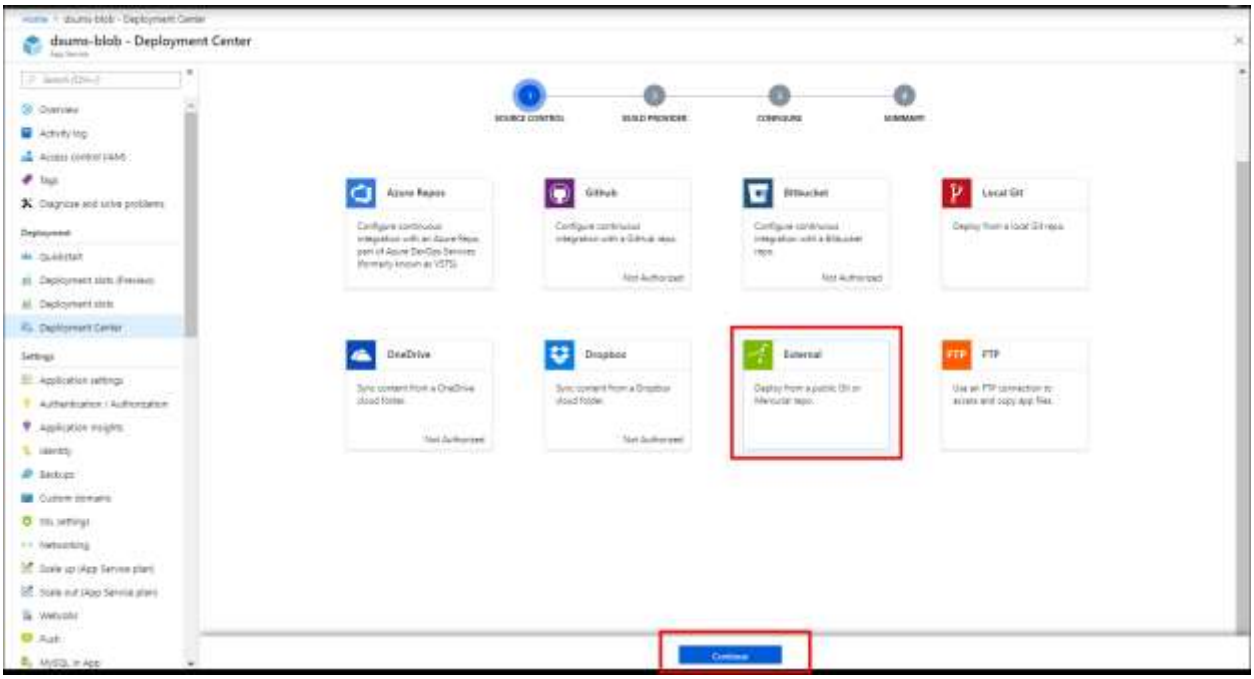
1. Login to Azure portal: .
2. Select Dashboard Server App Services.
3. Choose **Deployment Center** in the left panel of the Dashboard Server App Service.



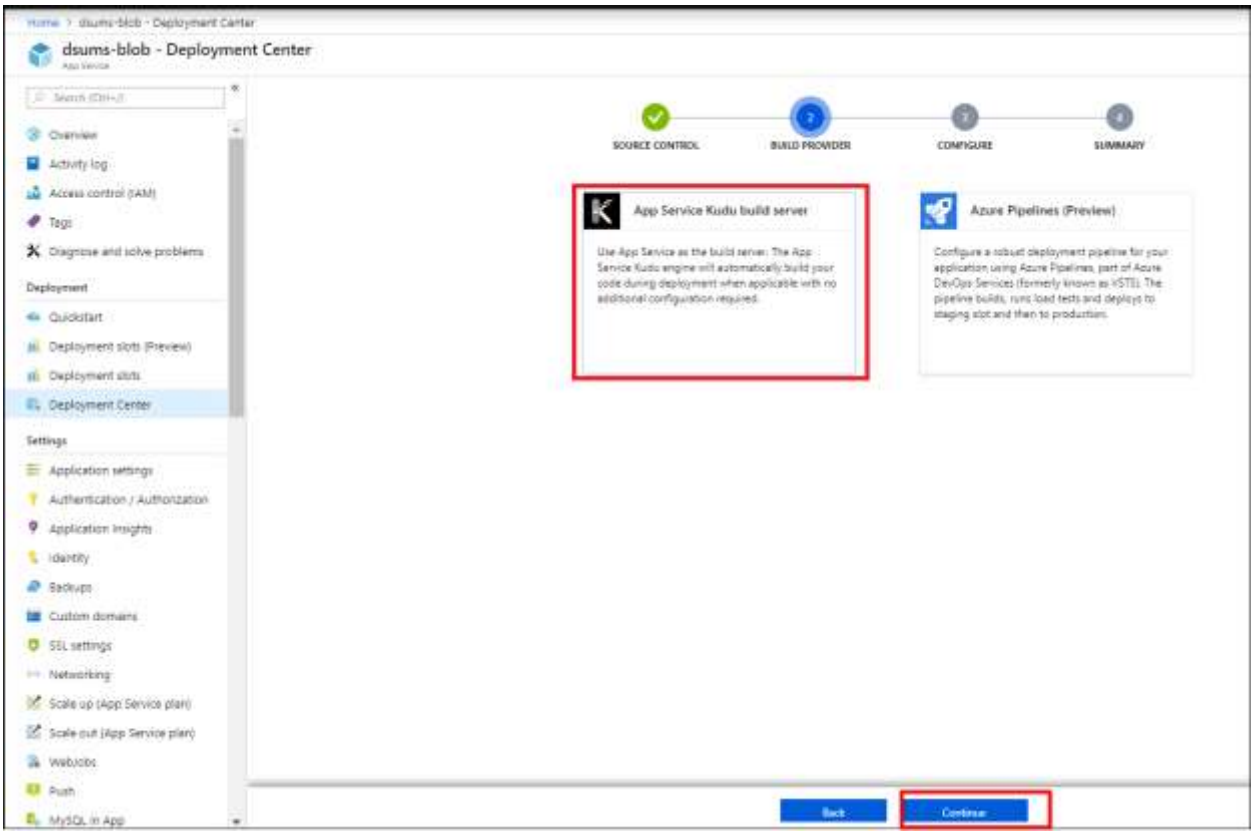
4. Once the deployment center is clicked, click the **Disconnect**. Now the deployment panel will open.



5. Select the **External** in the source control tab and click **Continue**

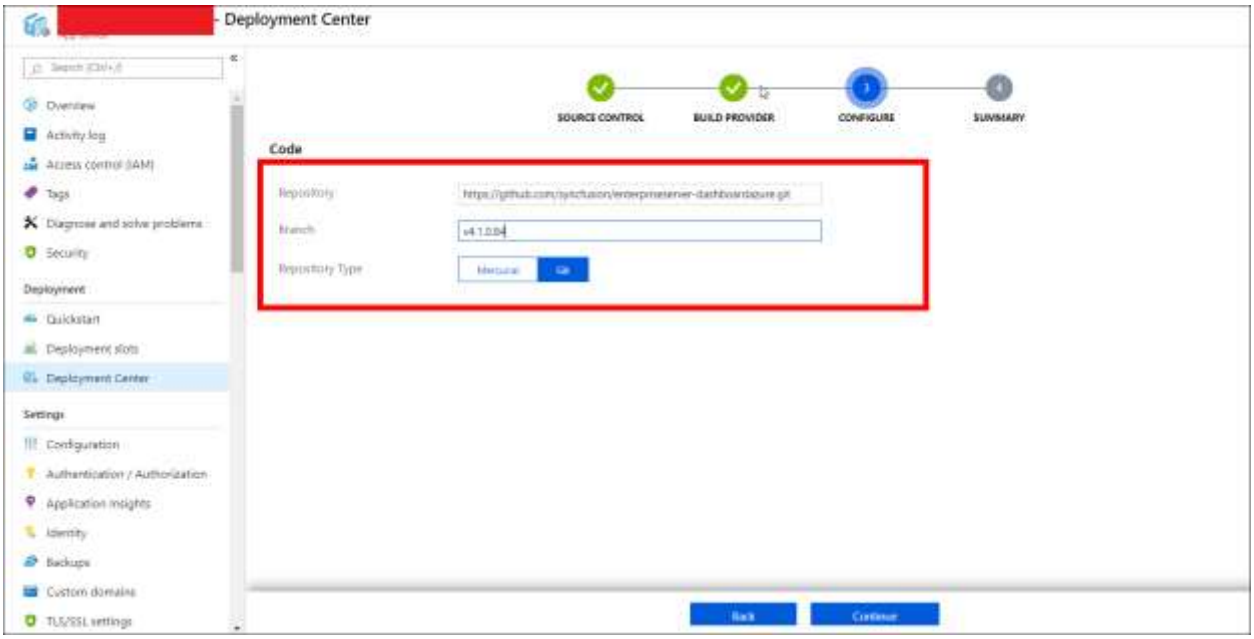


6. Select the **App Service Kudu build server** and click **Continue**.

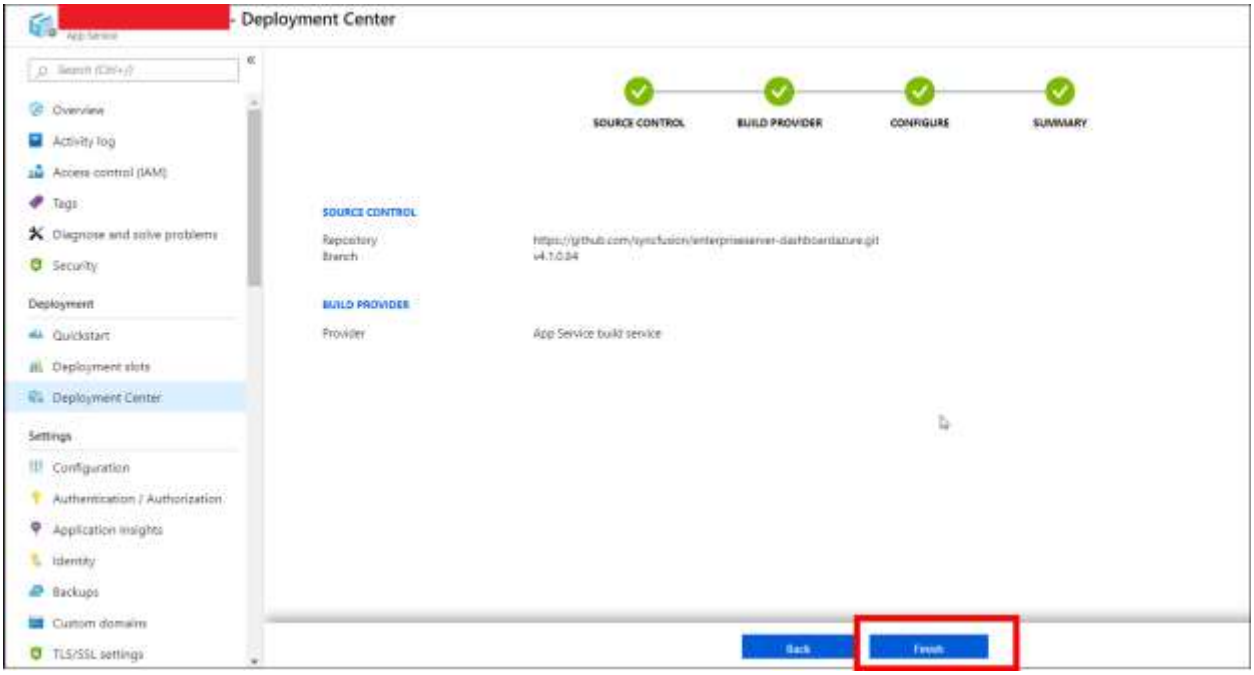


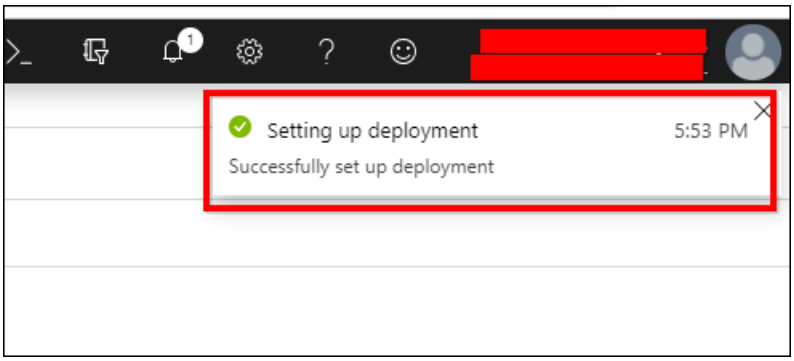
7. Now, fill the below details in the **Configure** tab and click **Continue**.
    - Repository URL -

- Branch - v4.1.0.84
- Repository Type - Git



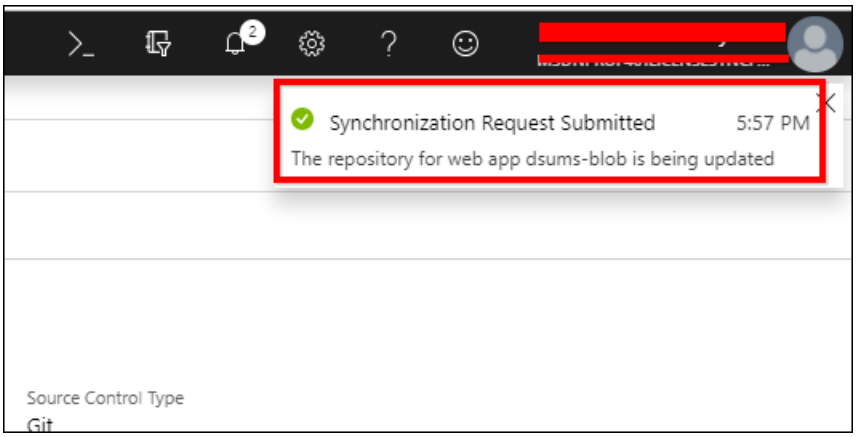
8. Next click on **Finish** option.The "Successfully set up deployment" notification will be shown in the notification blade.





9. Select the **Sync** button in the deployment center blade.

10. Synchronization request submitted notification is shown in the notification blade and synchronization progress is shown in the deployment options blade.



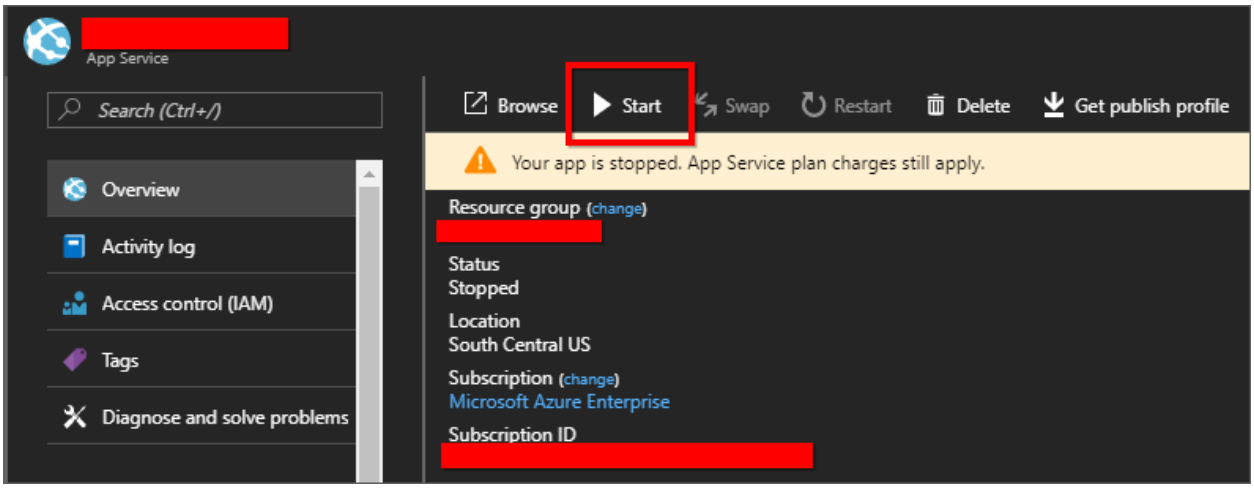
11. Please wait until synchronization is completed.

Start Syncfusion Dashboard Server Azure App Service

Once the Dashboard Server App Service update/upgrade is completed, start the Dashboard Server App Service.

Follow the below steps to start App Service:

- Select App Services in the Microsoft Azure Services.
- Choose Dashboard Server Azure App Service.
- In the **Overview** section, click the **Start**.

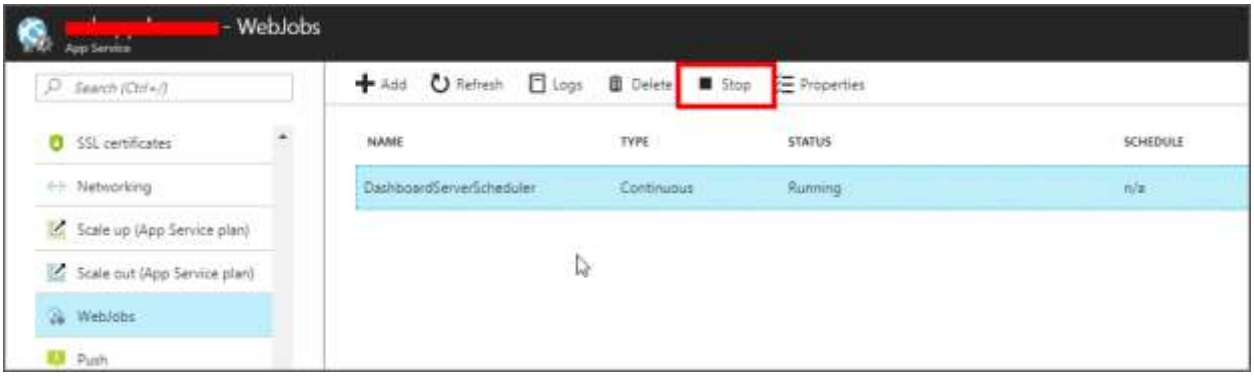


Restart the WebJobs

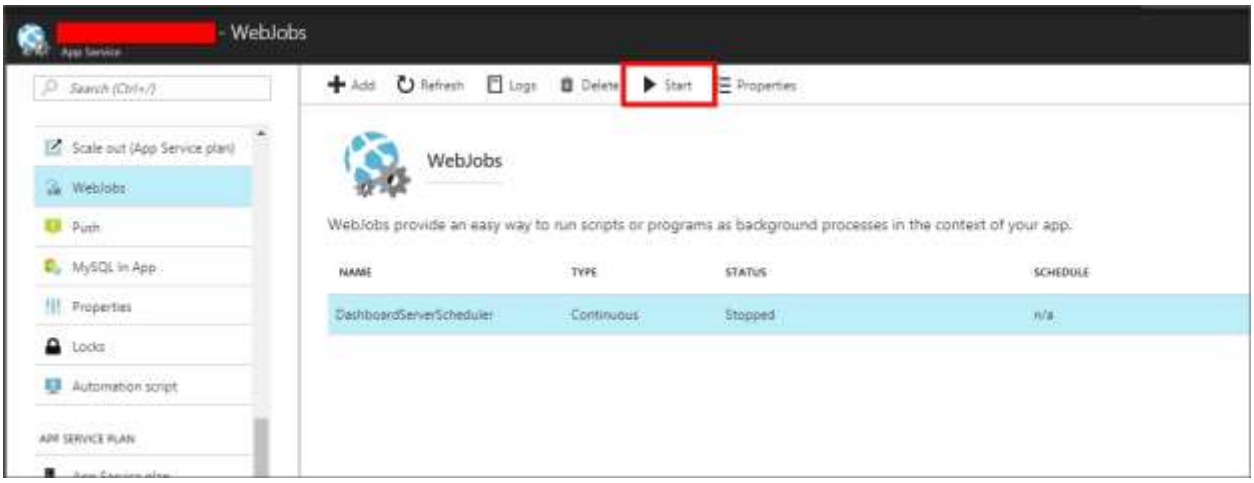
Once the Azure blob storage details are configured in the Dashboard Server application, you must restart the WebJobs.

Follow the below steps to restart the WebJobs:

1. Select **WebJobs** in the left panel of the latest Dashboard Server App Service.
2. Choose **DashboardServerScheduler** WebJobs, and click the stop.



3. Once the status for **DashboardServerScheduler** WebJobs is changed to stopped, click the start.



Deploy Syncfusion User Management Server Azure App Service

1. Click here to deploy Syncfusion User Management Server Azure app service.
2. Connect Dashboard Server with the deployed User Management Server. You can find how to configure Dashboard Server with User Management Server here

Upgrade Dashboard Server Azure App Service from versions less than or equal to v2.1.0.2

Up to Dashboard Platform release version v2.1.0.2, we delivered zip packages which contains the Dashboard Server application to be manually deployed in the Azure. This had many manual steps which consumed more time for the deployment.

To overcome this, we have prepared Azure ARM templates for the Dashboard Server and its applications which could be used to deploy the application into Azure using the Custom Deployment model.

Syncfusion Dashboard Server version 4.1 comes with a User Management Server which is a separate application for managing your users and the applications. To know more about User Management Server click here.

Migration to latest Syncfusion Dashboard Server version 4.1

Follow the below steps to upgrade the Dashboard Server App Service to the latest source from older versions.

- Stop Syncfusion Dashboard Server Azure App Service.
- Get existing Azure blob storage details.
- Update existing Azure blob storage details in App setting section.
- Start Syncfusion Dashboard Server App Service.
- Restart WebJobs.

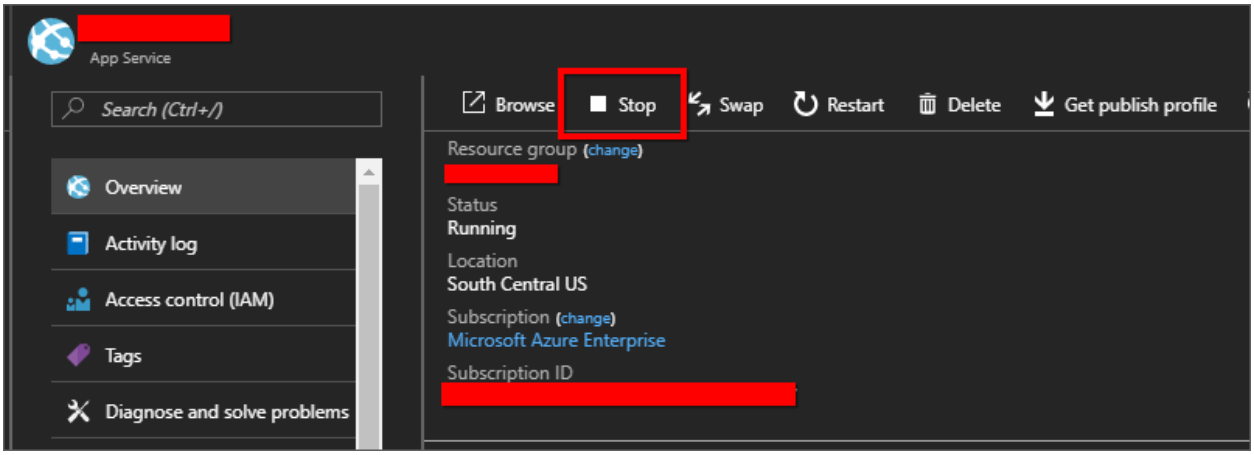
Stop Syncfusion Dashboard Server Azure App Service

Before starting the upgrade/update process, the Dashboard Server App Service should be stopped first.

Follow the below steps to stop the Dashboard Server App Service.

- Login to Azure portal: .
- Select App Services in Microsoft Azure Services.
- Choose the Dashboard Server Azure App Service.

- In the **Overview** section, click the **Stop**.



Get existing Azure blob storage details

In existing Dashboard Server, all resources are stored in the Azure blob storage. All items are needed while upgrading to latest version Dashboard Server App Service.

Previously the Azure blob storage details are configured in the web.config file.

Follow the below steps and get the existing Azure blob storage details from the web.config file:

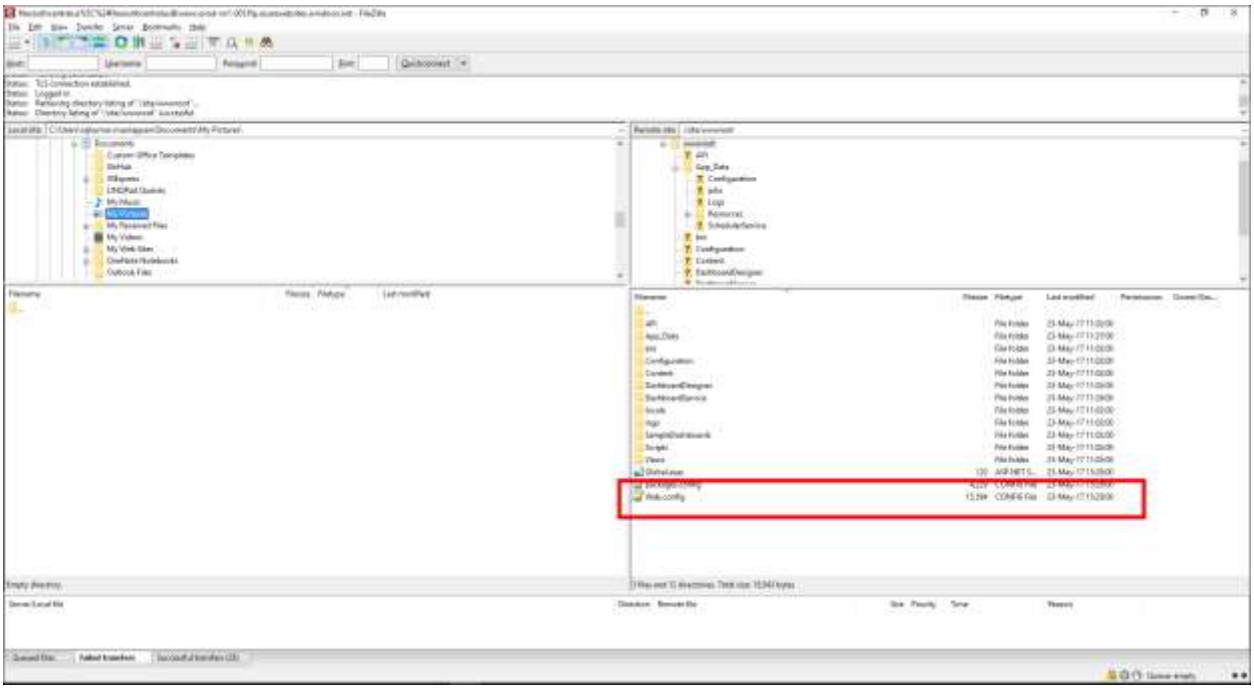
1. Login to Azure portal: .
2. Select App Services in Microsoft Azure Services.
3. Choose the existing Syncfusion Dashboard Server Site and then click the get publish profile.

4. Save the .PublishSettings file and open it.
5. The file contains 2 \ sections for web deploy and FTP.
6. From the FTP \ section, copy the following values:
   - publishUrl
   - userName
   - userPWD


```
<publishData>
    <publishProfile profileName="webappdsserver - Web Deploy" publishMethod="MSDeploy"
    publishUrl="webappdsserver.scm.azurewebsites.net:443"
    msdeploySite="webappdsserver"
    userName="$webappdsserver"
    userPWD="78f9Bjxv9ZzJjefTdGxLAh8hDLREWFG2WwEQv5dNm2jTAw9H4Aq93iJ8WPsL"
    destinationAppUrl="http://webappdsserver.azurewebsites.net"
    SQLServerDBConnectionString="" mySQLDBConnectionString=""
    hostingProviderForumLink="" controlPanelLink="" webSystem="WebSites">
        <databases />
    </publishProfile>
    <publishProfile profileName="webappdsserver - FTP"
    publishMethod="FTP"
    publishUrl="ftp://waws-prod-sn1-085.ftp.azurewebsites.windows.net/site/wwwroot"
    ftpPassiveMode="True"
    userName="webappdsserver\$webappdsserver"
    userPWD="78f9Bjxv9ZzJjefTdGxLAh8hDLREWFG2WwEQv5dNm2jTAw9H4Aq93iJ8WPsL"
    destinationAppUrl="http://webappdsserver.azurewebsites.net"
    SQLServerDBConnectionString="" mySQLDBConnectionString="" hostingProviderForumLink="" controlPanelLink="" webSystem="WebSites">
        <databases />
    </publishProfile>
</publishData>
```

7. Apply the copied credentials to the host, username, and password fields in the FTP client FileZilla as shown here and click the quickconnect.

8. Download the web.config file from "/site/wwwroot" folder in the existing Azure App Service through FileZilla.



9. Get the below Azure blob storage details in the existing App Service web.config file.

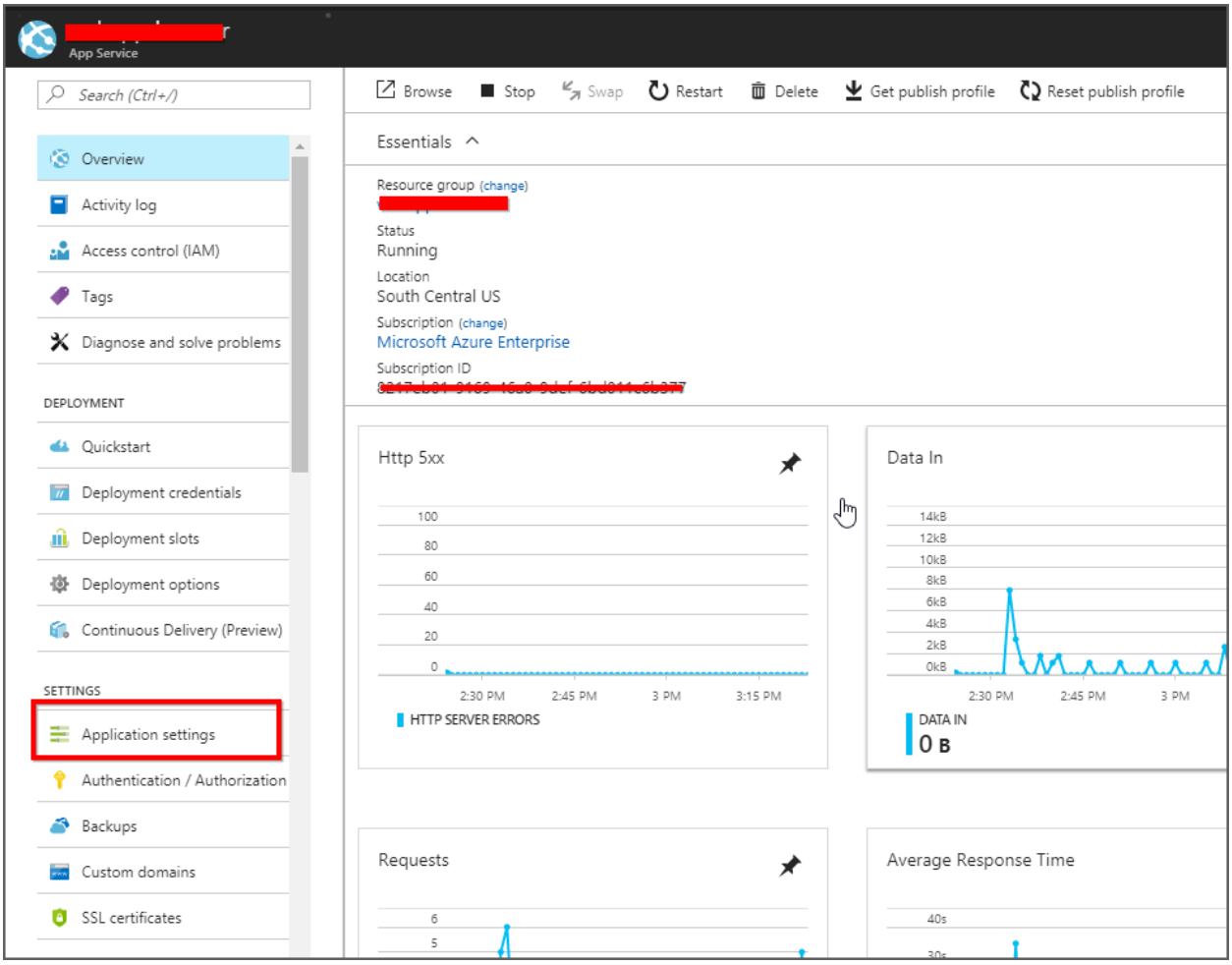
- AzureStorageAccountName
- AzureStorageBlobServiceEndpoint
- AzureStorageAccessKey
- AzureStorageContainerName
- AzureStorageConnectionType
- AzureStorageBlobURL

```
<add key="AzureStorageAccountName" value="syncfusionazurestorage" />
<add key="AzureStorageBlobServiceEndpoint" value="https://syncfusionazurestorage.blob.core.windows.net/" />
<add key="AzureStorageAccessKey" value="+7+j9aGDpqyrf+GSG6/ckxPBsOU8XAXGfIOOdJvqA2gBarOeD/DDMyQ8I0kelHJtl7dPv/uzMssY7aCgF5avMg==" />
<add key="AzureStorageContainerName" value="syncfusionazurecontainer" />
<add key="AzureStorageConnectionType" value="customendpoint" />
<add key="AzureStorageBlobURL" value="https://syncfusionazurestorage.blob.core.windows.net/" />
```

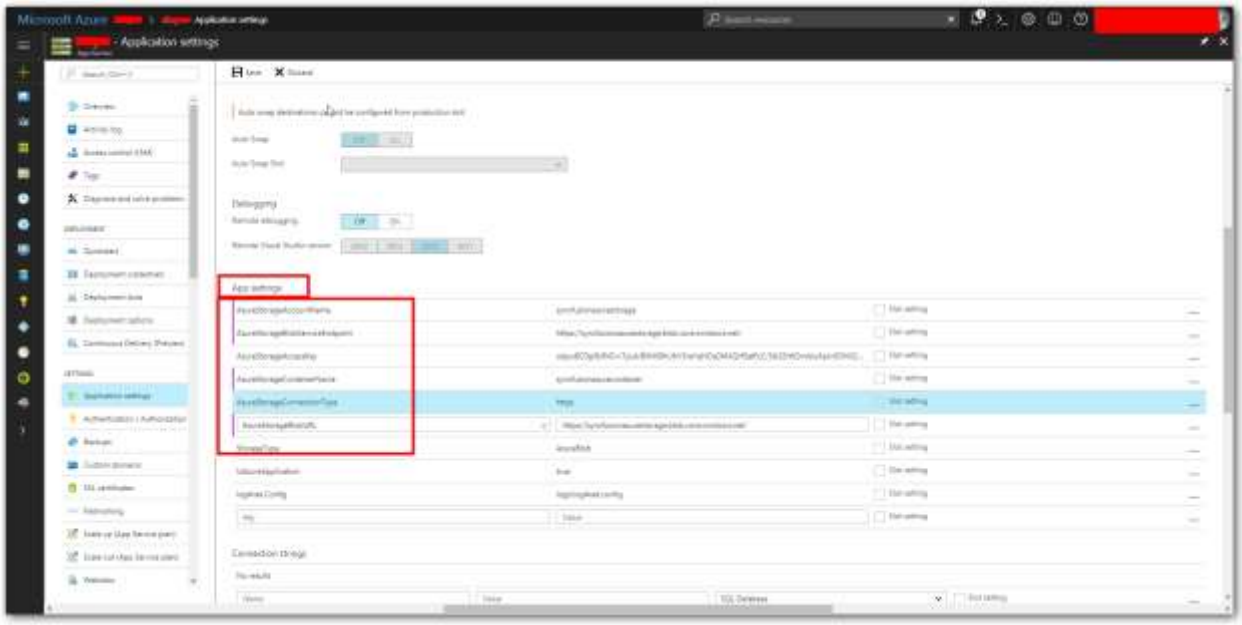
Update existing Azure blob storage details in App setting section

Follow the below steps to configure the existing Azure blob storage details to the latest Dashboard Server:

1. Login to Azure portal: .
2. Select App Services in Microsoft Azure Services.
3. Choose the latest Syncfusion Dashboard Server App Service.
4. Select **Application Settings** in the left panel of the latest Dashboard Server App Service.



5. Configure existing App Service Azure blob storage details in the new App Service, and app setting section in the application settings tab.



6. Add the below details in the app setting section.
    - AzureStorageAccountName
    - AzureStorageBlobServiceEndpoint
    - AzureStorageAccessKey
    - AzureStorageContainerName
    - AzureStorageConnectionType
    - AzureStorageBlobURL
7. Also add the following informations in the app setting section.

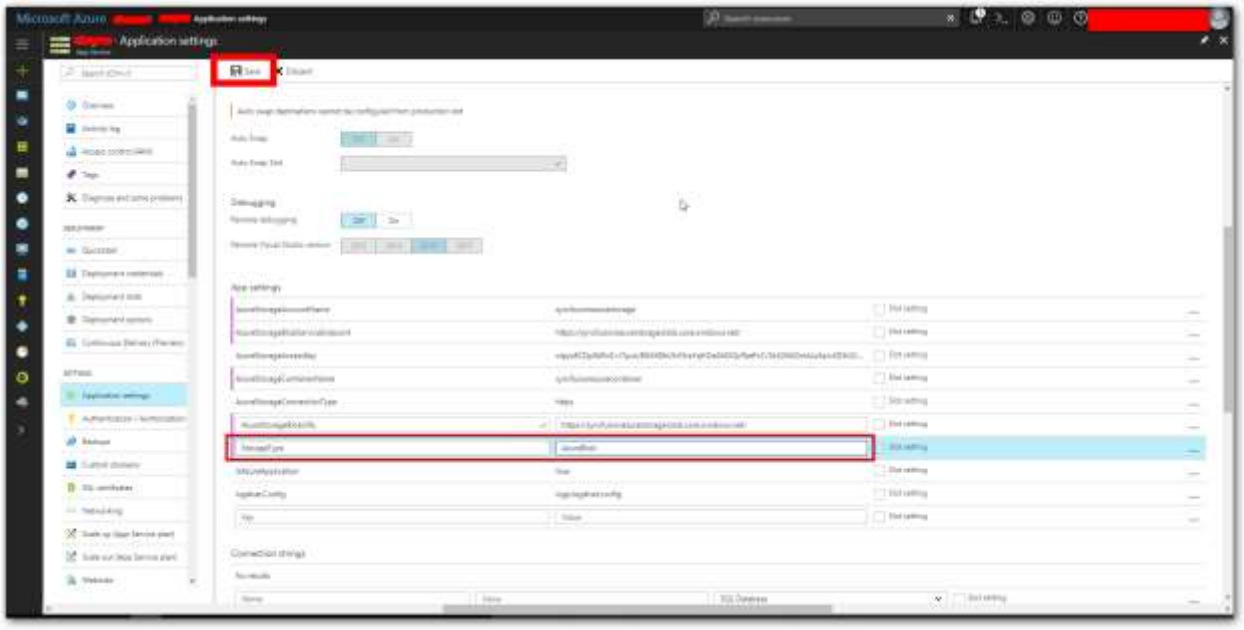
||Key||Value||

|StorageType|AzureBlob|

|IsAzureApplication|true|

|log4net.Config|logs\log4net.config|

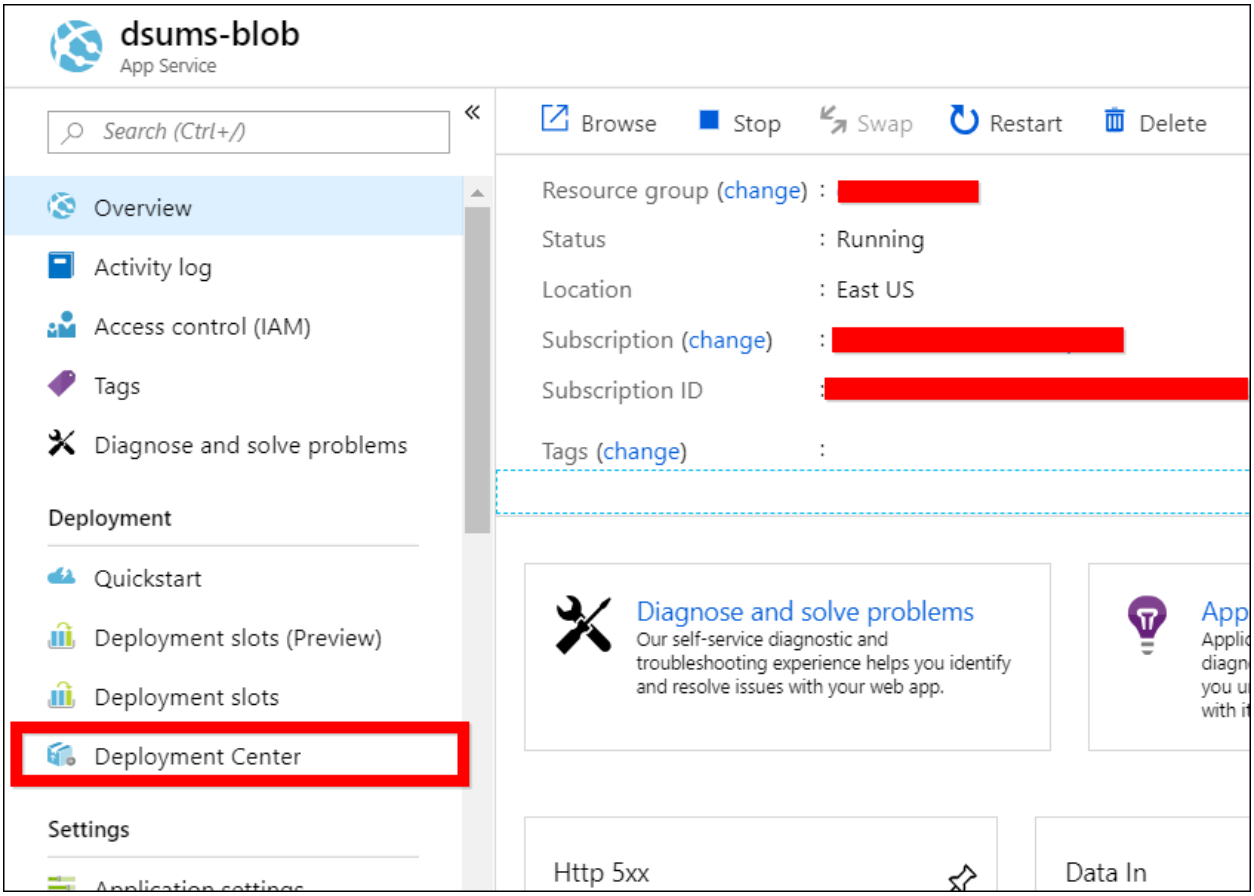
8. Click the save.



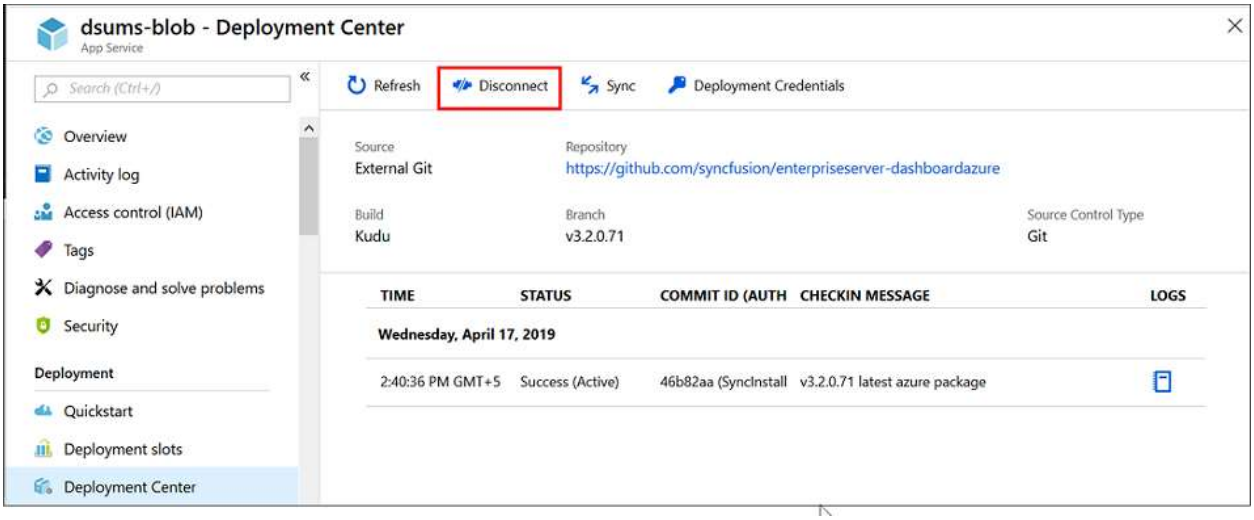
Migration to latest versions

Follow the below steps to configure the existing Azure blob storage details to the latest Dashboard Server:

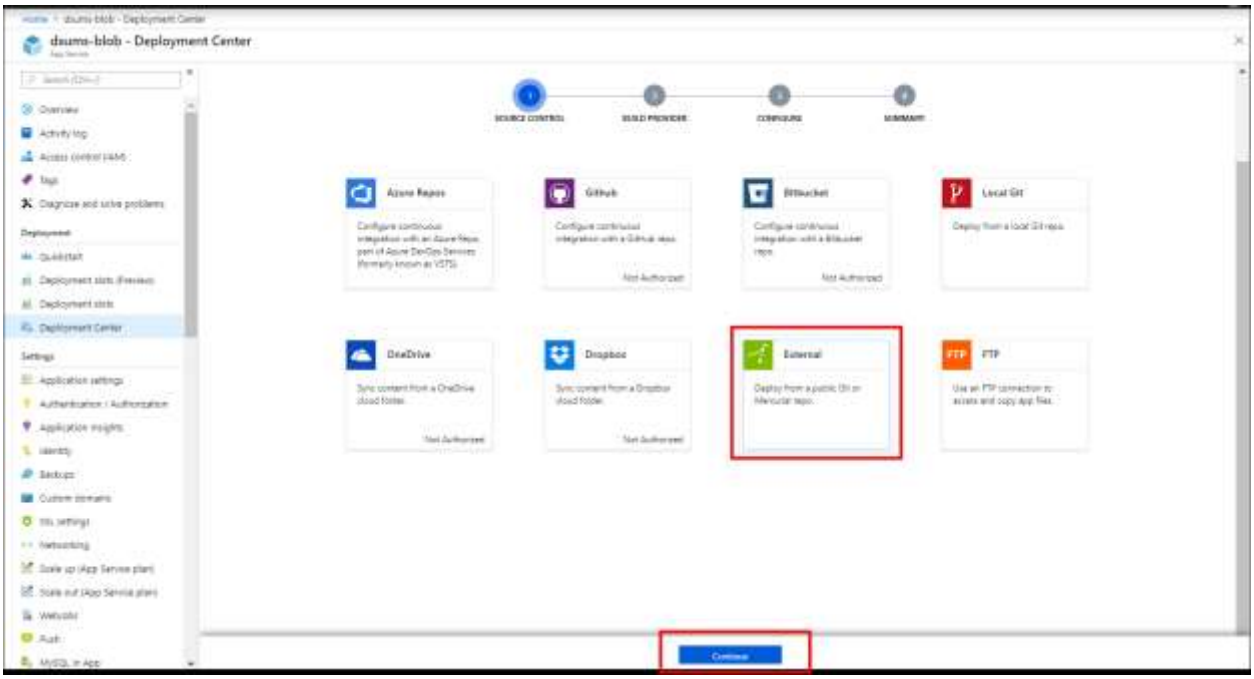
1. Login to Azure portal: .
2. Select Dashboard Server App Services.
3. Choose **Deployment Center** in the left panel of the Dashboard Server App Service.



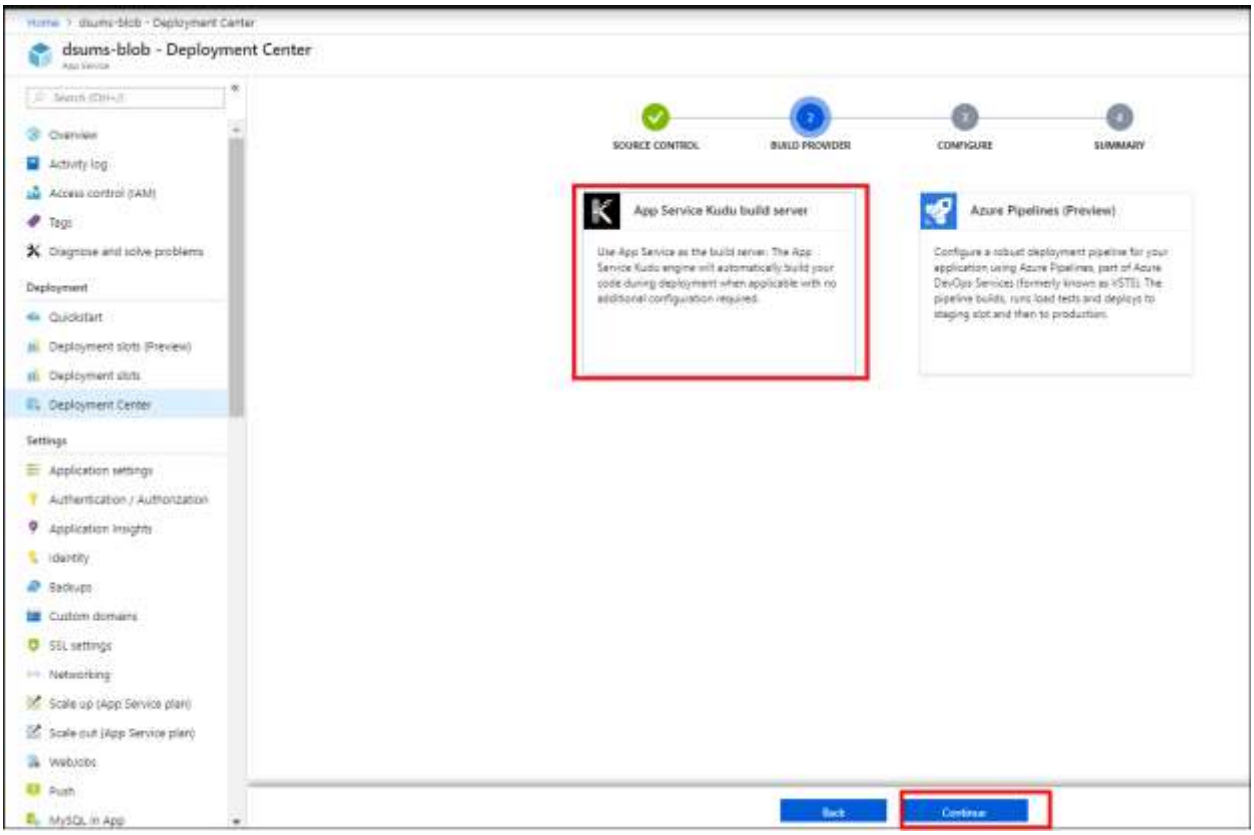
4. Once the deployment center is clicked, click the **Disconnect**. Now the deployment panel will open.



5. Select the **External** in the source control tab and click **Continue**.

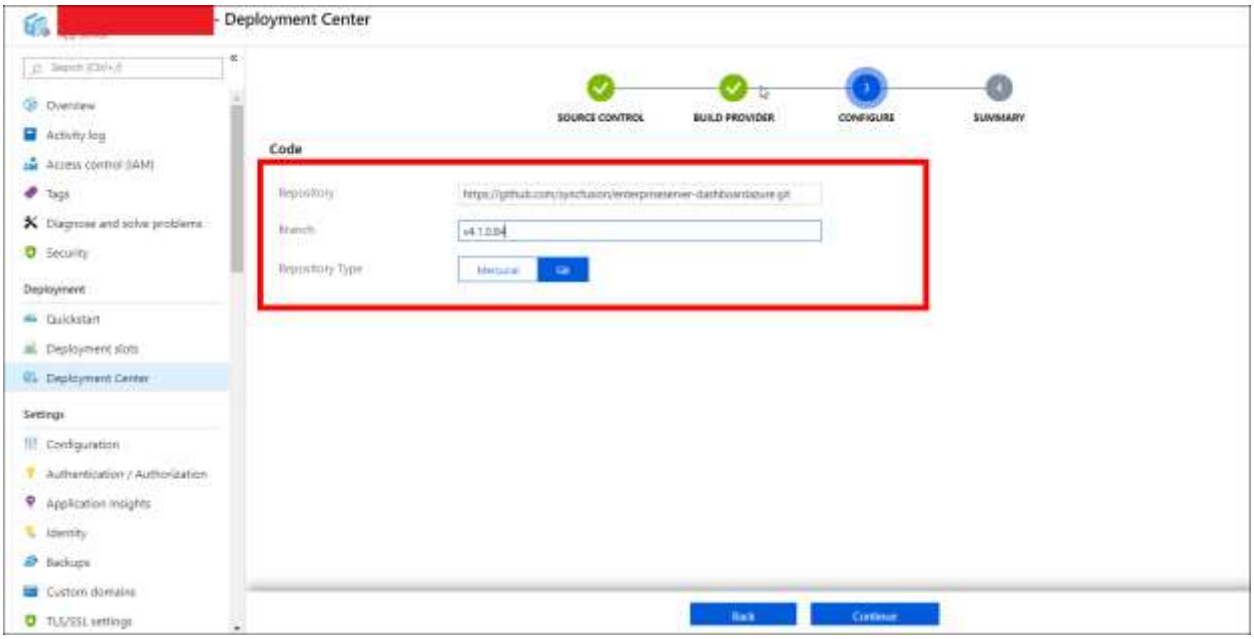


6. Select the **App Service Kudu build server** and click **Continue**.

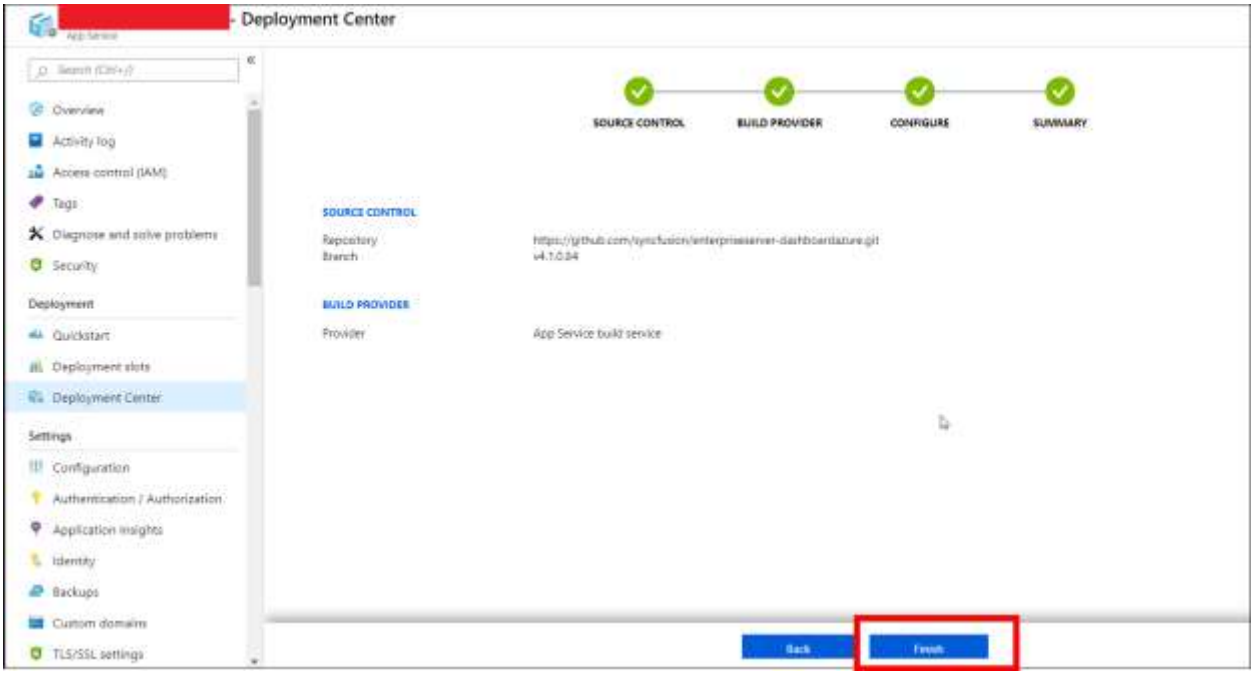


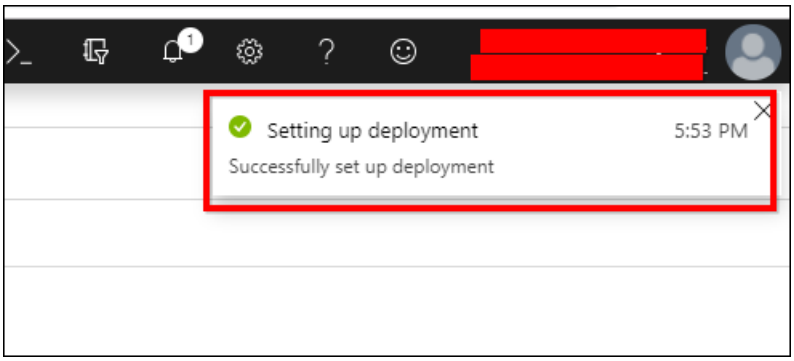
7. Now, fill the below details in the **Configure** tab and click **Continue**.
    - Repository URL -

- Branch - v4.1.0.84
- Repository Type - Git



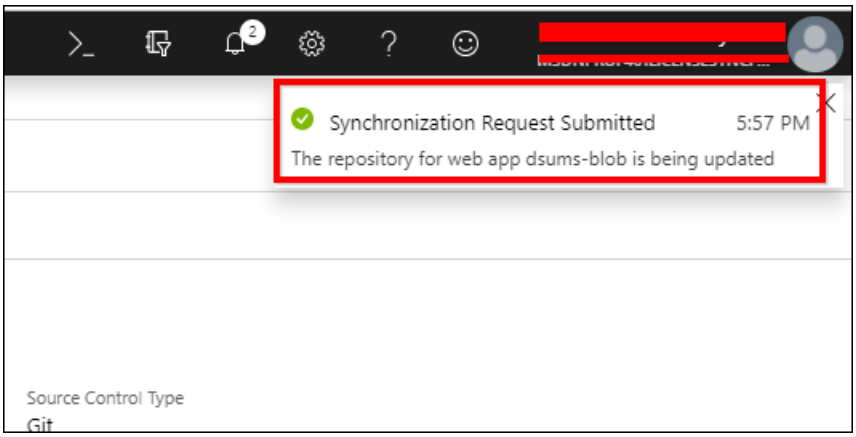
8. Next click on **Finish** option. The "Successfully set up deployment" notification will be shown in the notification blade.





9. Select the **Sync** button in the deployment center blade.

10. Synchronization request submitted notification is shown in the notification blade and synchronization progress is shown in the deployment options blade.



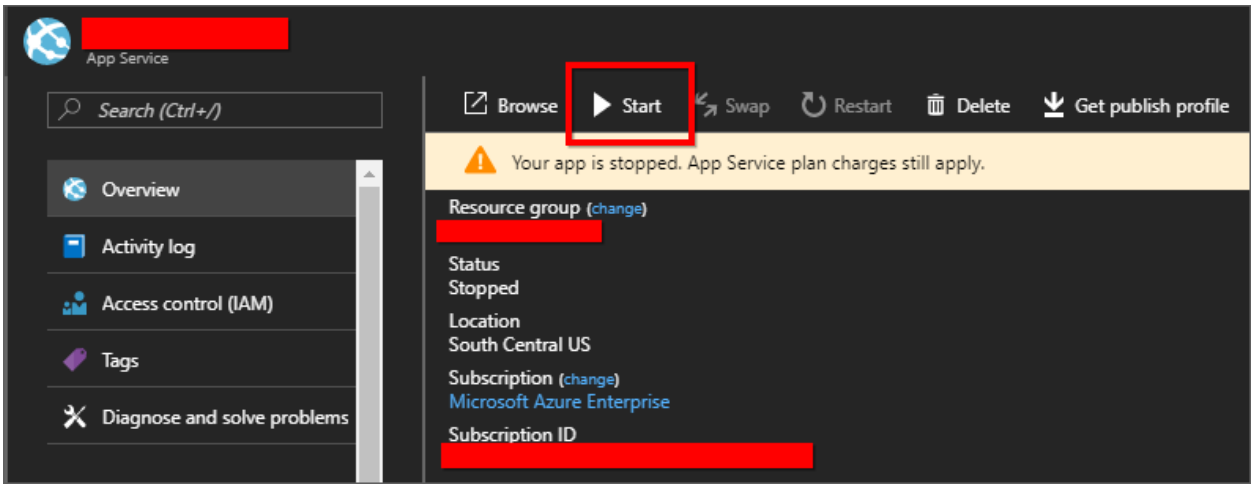
11. Please wait until synchronization is completed.

Start Syncfusion Dashboard Server Azure App Service

Once the Dashboard Server App Service update/upgrade is completed, start the Dashboard Server App Service.

Follow the below steps to start App Service:

- Select App Services in the Microsoft Azure Services.
- Choose Dashboard Server Azure App Service.
- In the **Overview** section, click the **Start**.

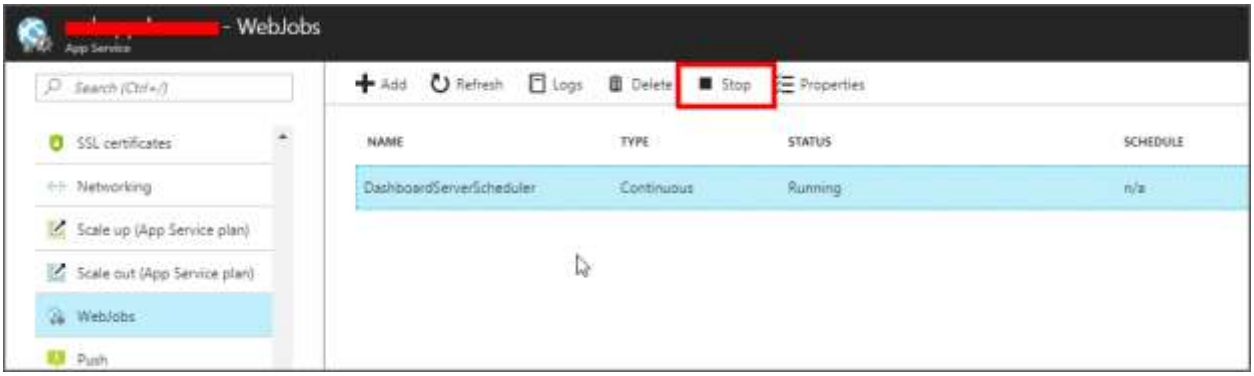


Restart the WebJobs

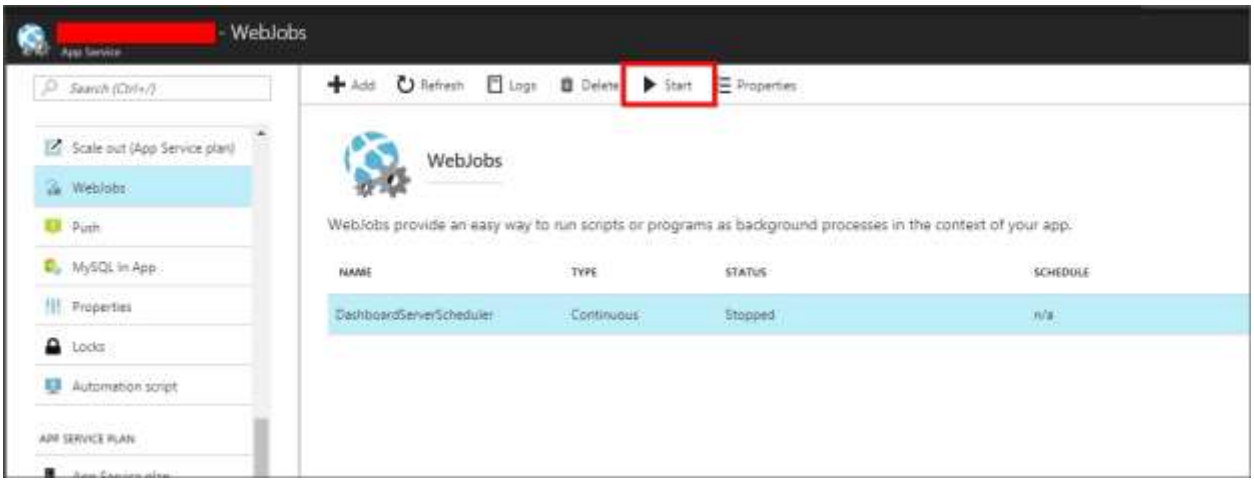
Once the Azure blob storage details are configured in the Dashboard Server application, you must restart the WebJobs.

Follow the below steps to restart the WebJobs:

1. Select **WebJobs** in the left panel of the latest Dashboard Server App Service.
2. Choose **DashboardServerScheduler** WebJobs, and click the stop.



3. Once the status for **DashboardServerScheduler** WebJobs is changed to stopped, click the start.



Deploy Syncfusion User Management Server Azure App Service

1. Click here to deploy Syncfusion User Management Server Azure app service.
2. Connect Dashboard Server with the deployed User Management Server. You can find how to configure Dashboard Server with User Management Server here

VM

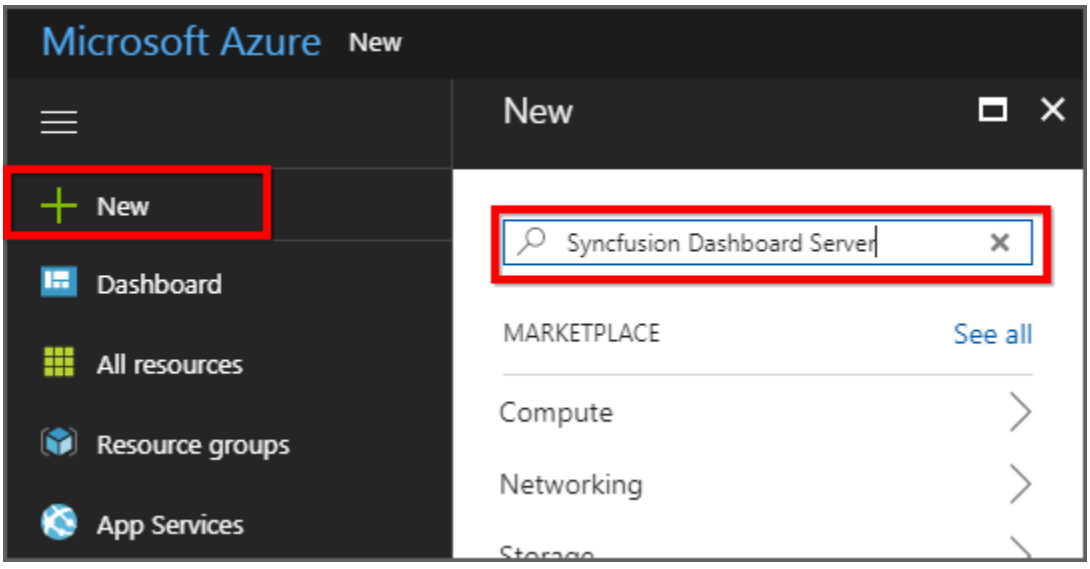
Dashboard server VM with Microsoft Azure

This section explains how to use the dashboard in the Azure portal to select and create the Syncfusion Dashboard Server virtual machine.

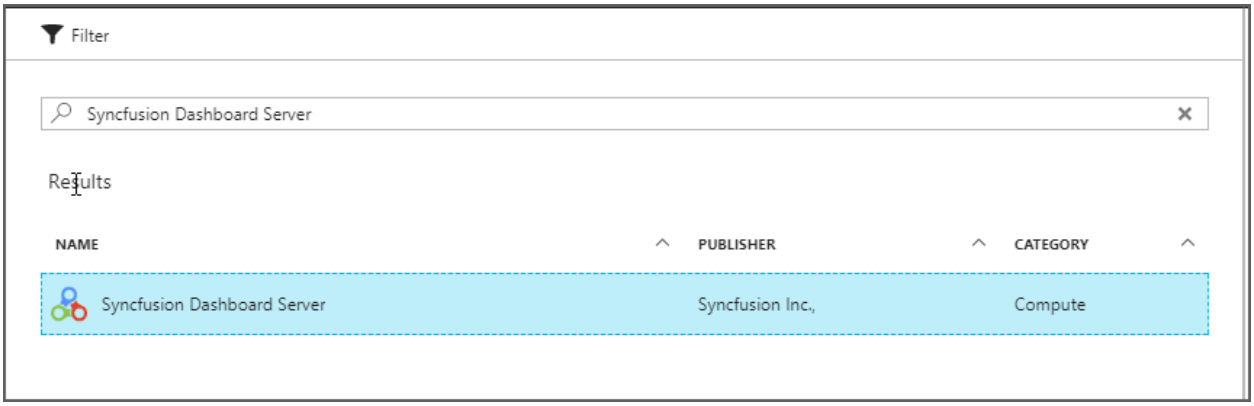
Pre-configured image via Azure Marketplace

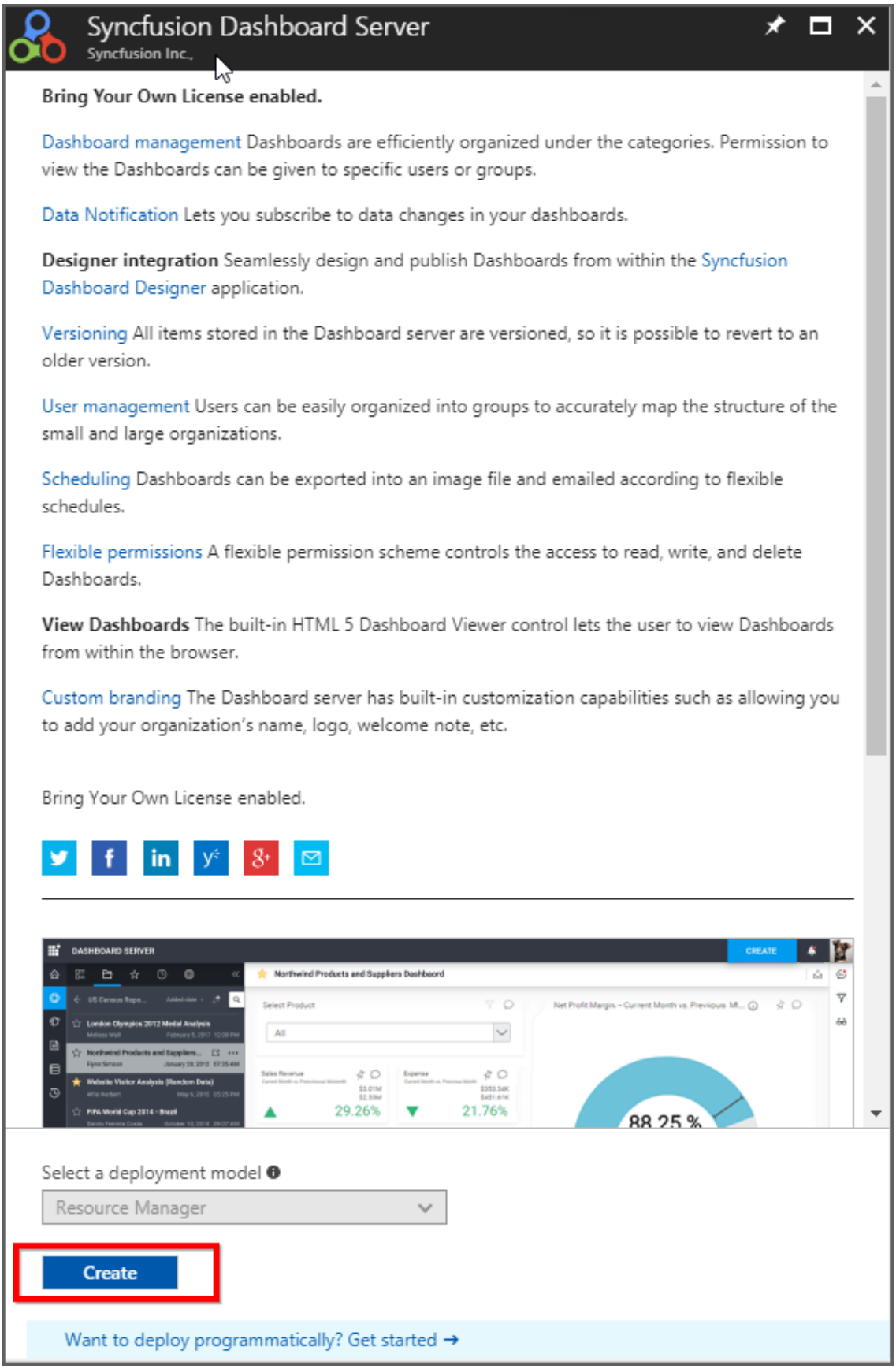
One of the fastest ways to get Dashboard Server up and to run in Azure is based on the pre-configured server image through the Azure Marketplace.

- Sign in to the Azure portal.
- Click new in the upper left side.
- Search for “Syncfusion Dashboard Server”.



- Select the Dashboard Server and click the “Create”.

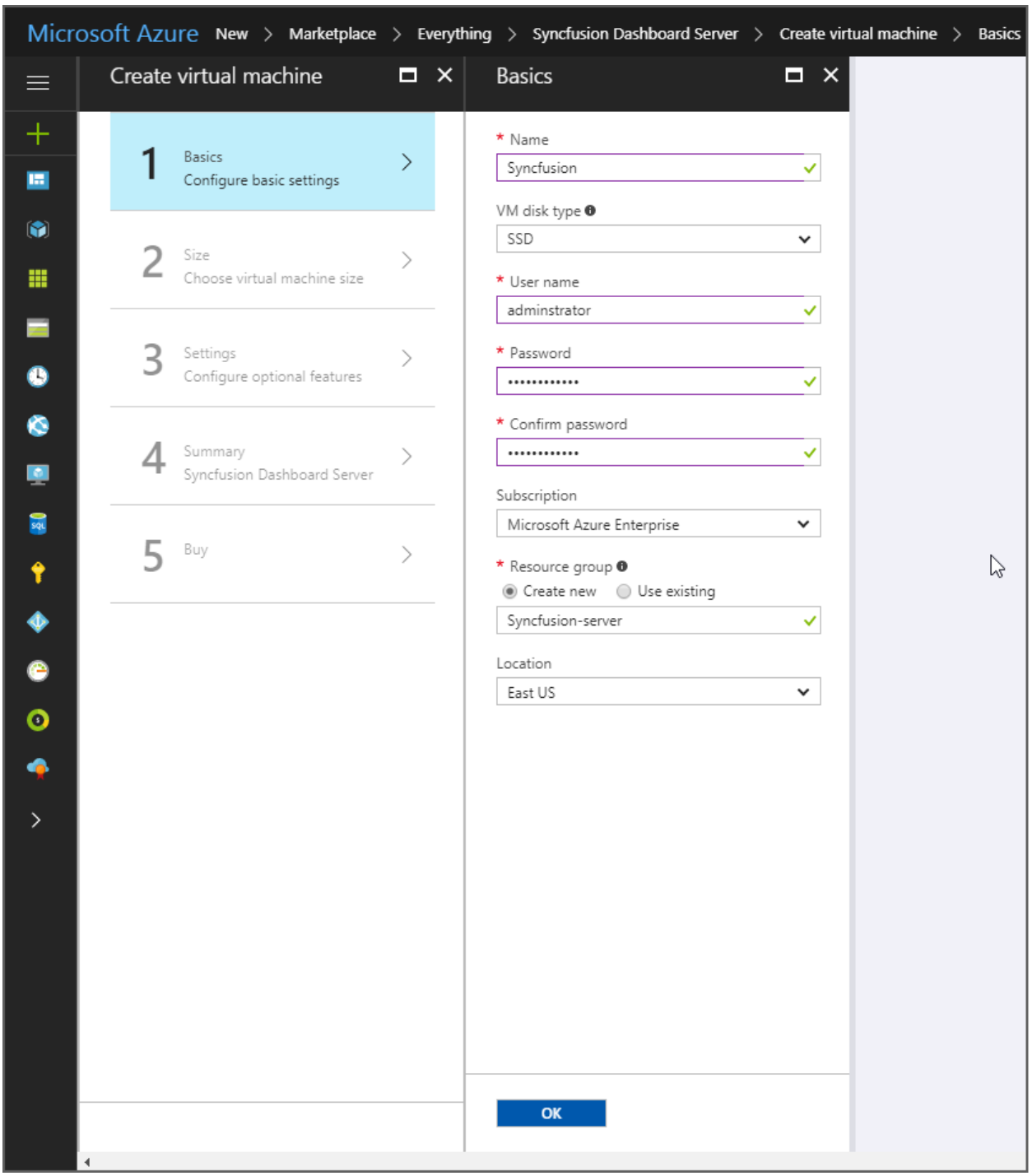




Basics blade

The basics blade requests administrative information for the virtual machine.

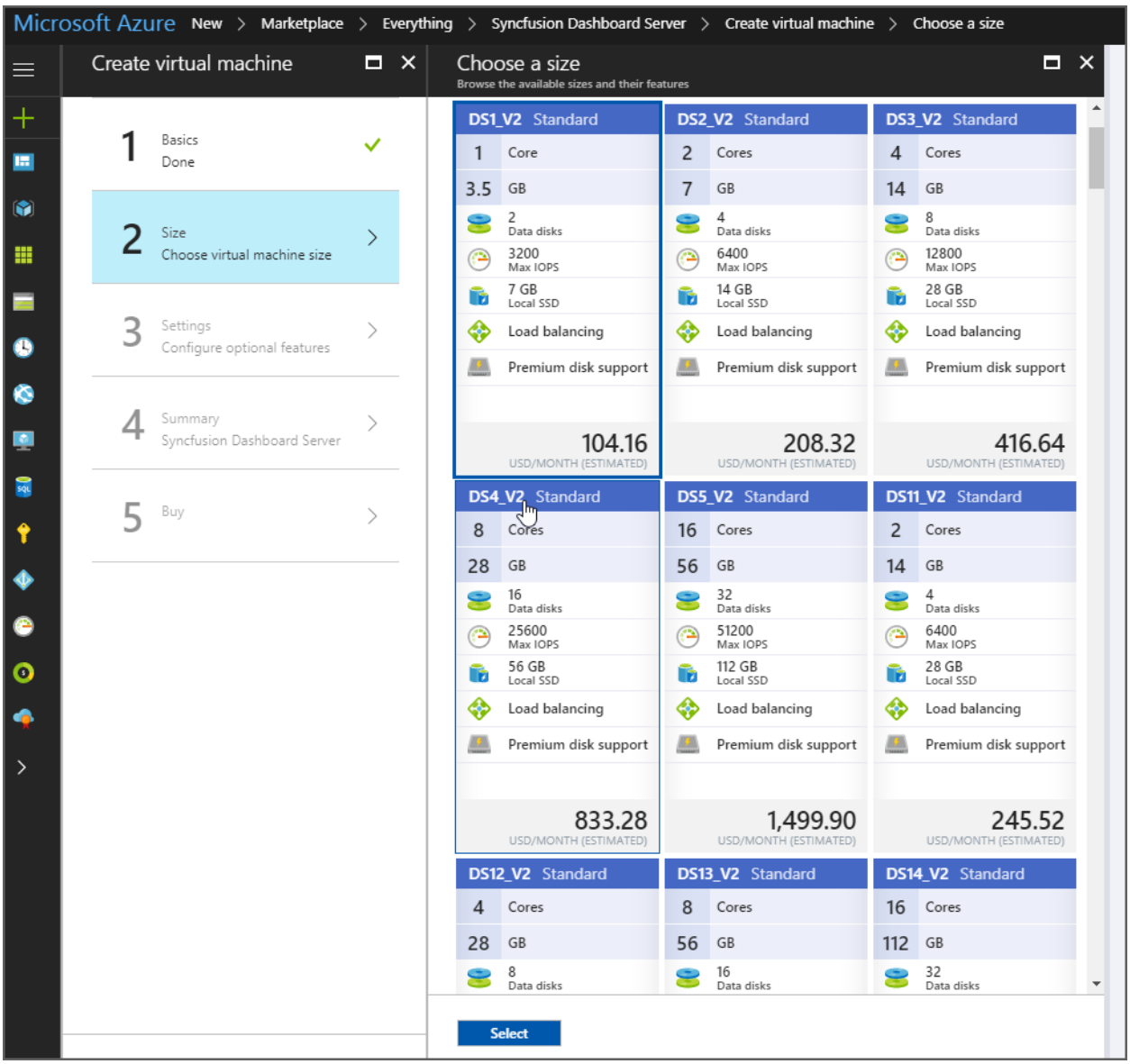
- **Name**: Enter a name for the virtual machine. The name must be within 1-15 characters and it should not contain special characters.
- **User name**: Enter a user name to create a local account on the VM. The local account is used to sign in and manage the VM.
- **Password**: Enter strong password to create a local account on the VM. The local account is used to sign in and manage the VM. The password must be within 8-123 characters and meets three out of the four following complexity requirements: one lower case character, one upper case character, one number, and one special character.
- **Subscription**: The subscription is optional. To learn more about subscription click here.
- **Resource group**: Select an existing resource group or type the name for a new one. To learn more about resource groups click here.
- **Location**: Select an Azure data center location where you want to run the VM.
- When you are done, click next to continue to the next blade.



Size blade

The size blade identifies the configuration details of the VM and lists various choices that include OS, number of processors, disk storage type, and estimated monthly usage costs.

- Choose the VM size, and then click select option to continue.



Settings blade

The settings blade requests storage and network options. You can accept the default settings. Azure creates appropriate entries at required location.

If the required virtual machine size is selected, you can try Azure premium storage by selecting the Premium (SSD) in disk type.

Storage

- **Storage**: Premium disks (SSD) backed by solid state drives are offer consistent and low-latency performance. They provide the best balance between price and performance, and are ideal for I/O-intensive applications and production workloads. Standard disks (HDD) are backed by magnetic drives and are preferable for applications where data is accessed infrequently.

When all the changes are completed, click OK.

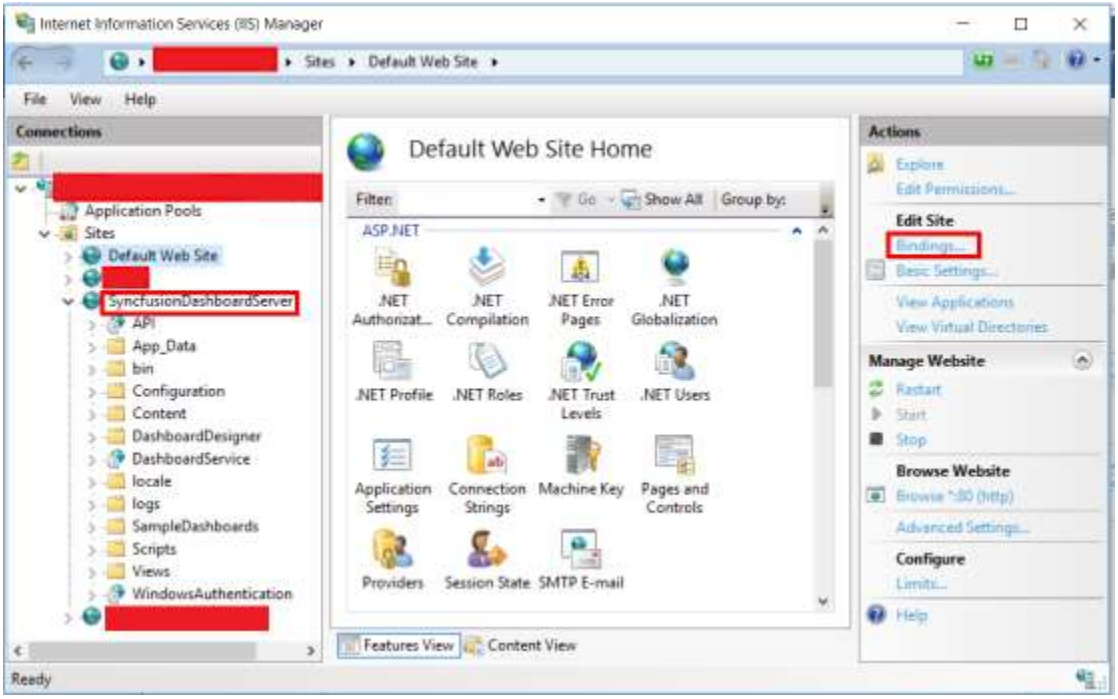
- **Use managed disks**: Enable this feature to automatically manage the availability of disks by Azure to provide data redundancy and fault tolerance, without creating and managing storage accounts on your own. Managed disks can not be available in all regions. To learn more about the managed disk click here.
- **Storage account**: Disks for Azure virtual machines are created in storage accounts. To learn more about storage account click here.

Network

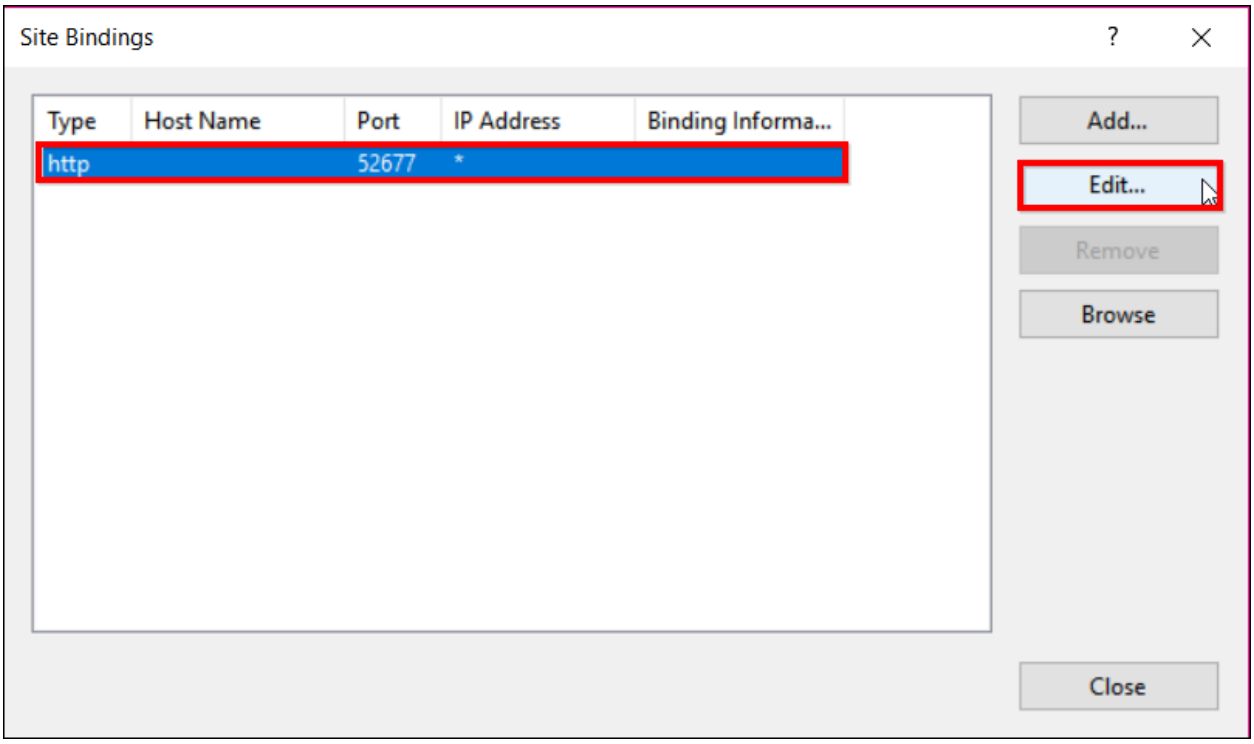
- **Virtual network**: The virtual networks are logically isolated from each other in Azure. You can configure their IP address ranges, subnets, route tables, gateways, and security settings like the traditional network in the data center. Virtual machines in the same virtual network can access each other by default.
- **Subnet**: A subnet is a range of IP addresses in the virtual network, that can be used to isolate virtual machines from each other or from the Internet.
- **Public IP address**: Use the public IP address if you want to communicate with the virtual machine from outside the virtual network.

Use globally accessed IP Address of the virtual machine to bound in IIS as shown below.

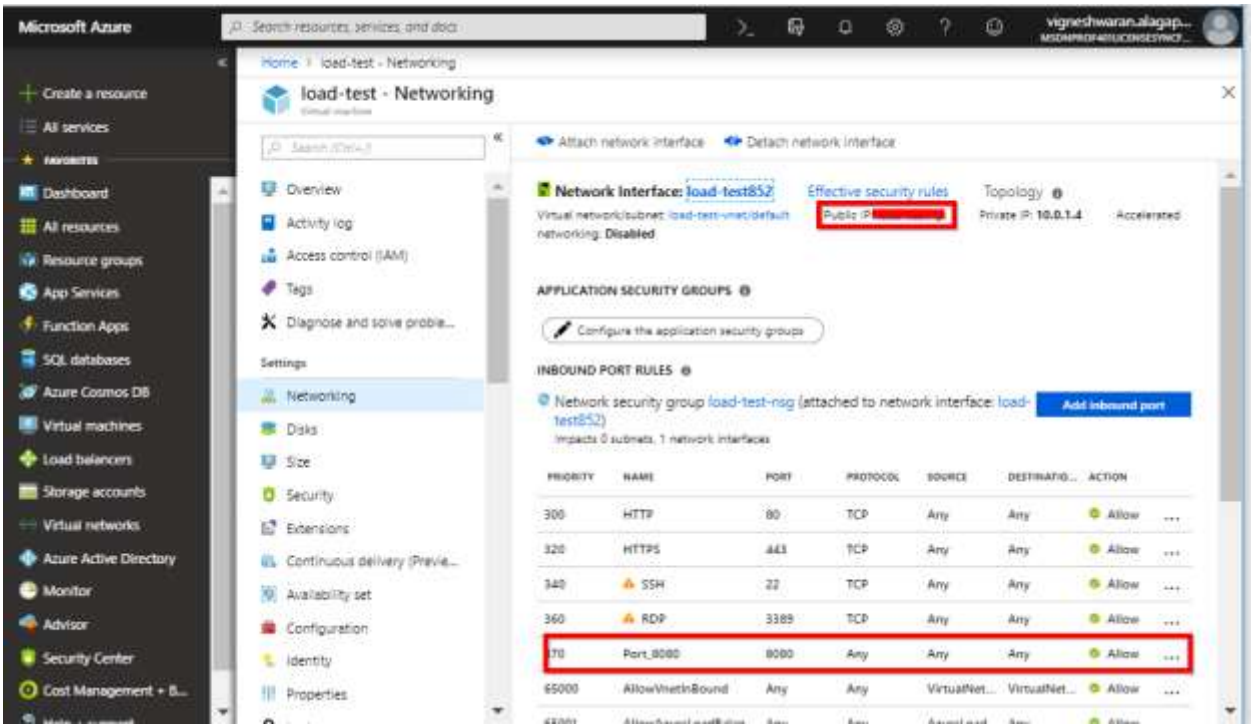
1. Open IIS and click on the Syncfusion Dashboard Server Site, and then select Bindings option.

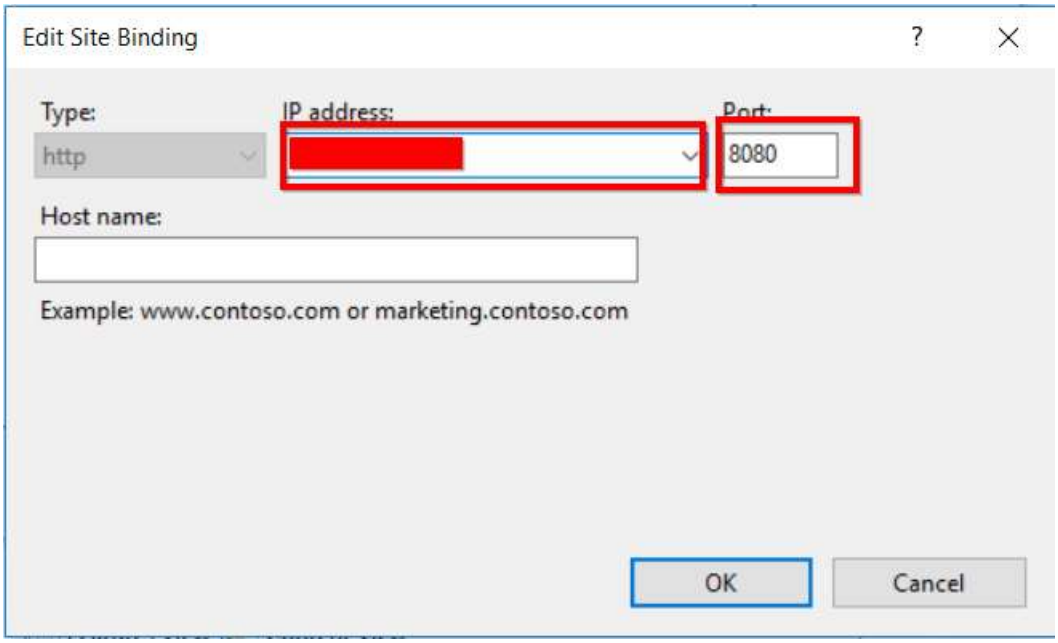


2. Select the site and click the Edit option.



3. Select the virtual machine public IP Address and provide the port number, and then proceed with OK button.

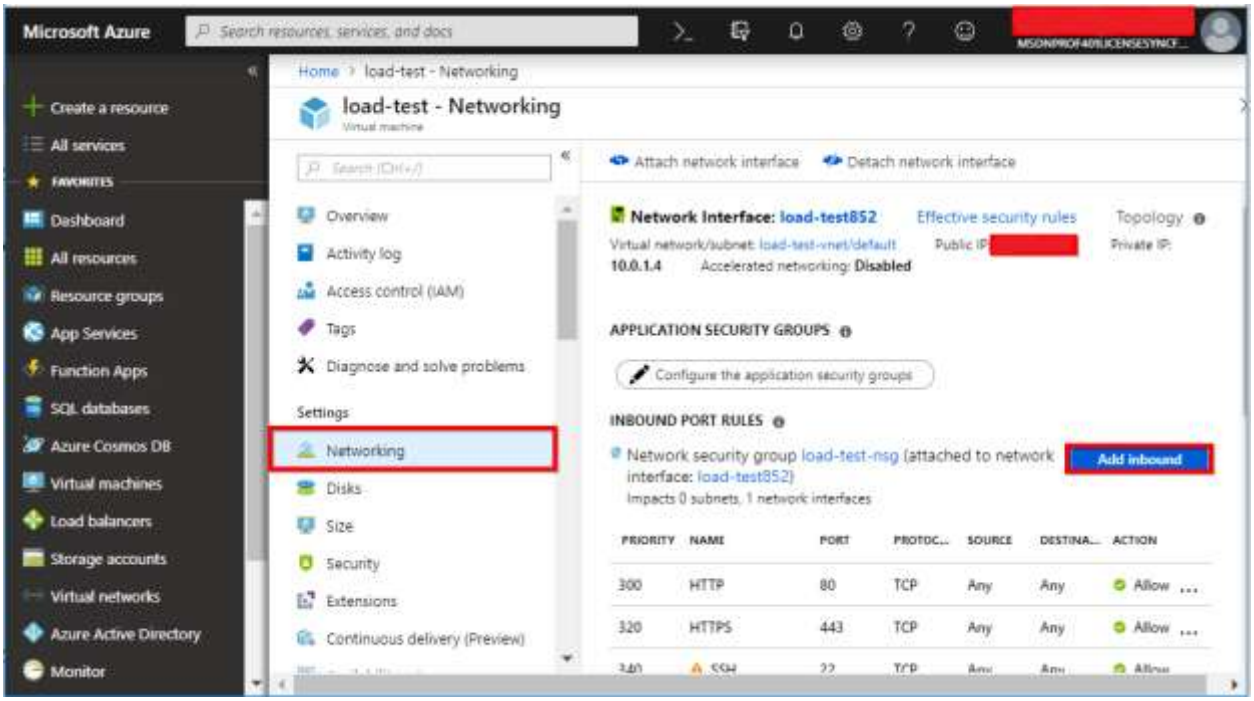




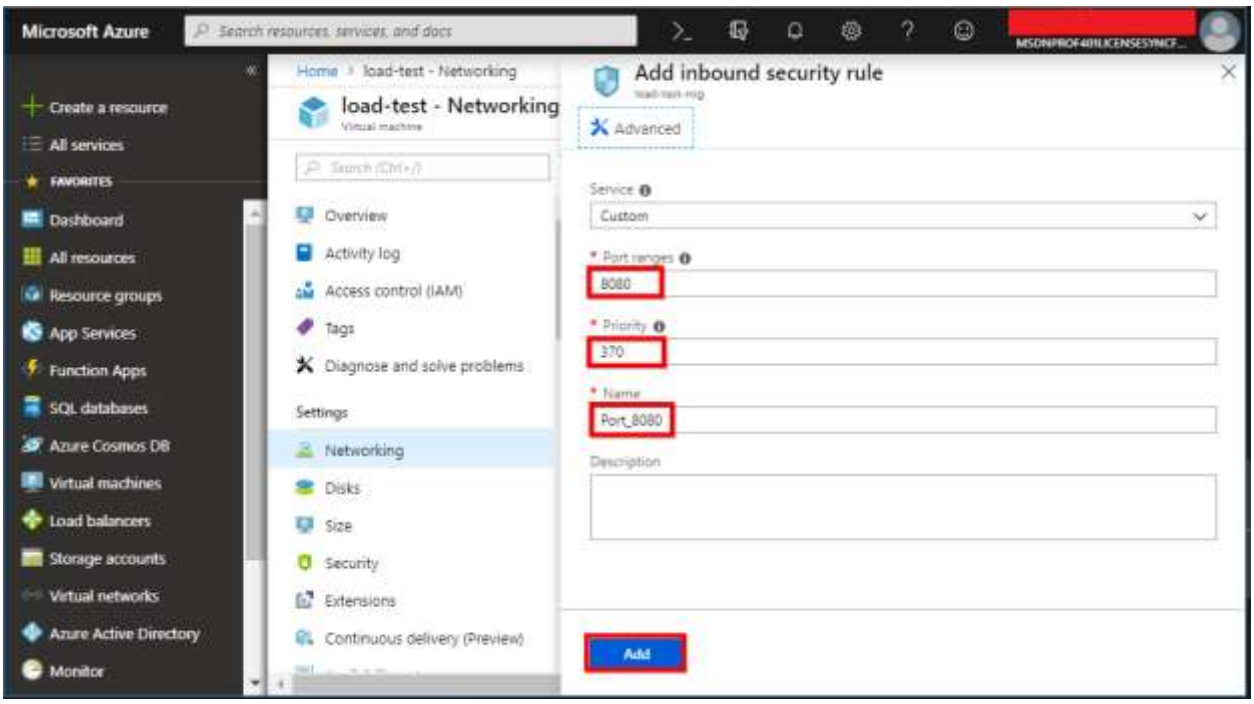
- **Network security group (firewall)**: A network security group is a set of firewall rules that control traffic to and from the virtual machine.

Please follow the below steps to bound the port in the network security group (fireWall):

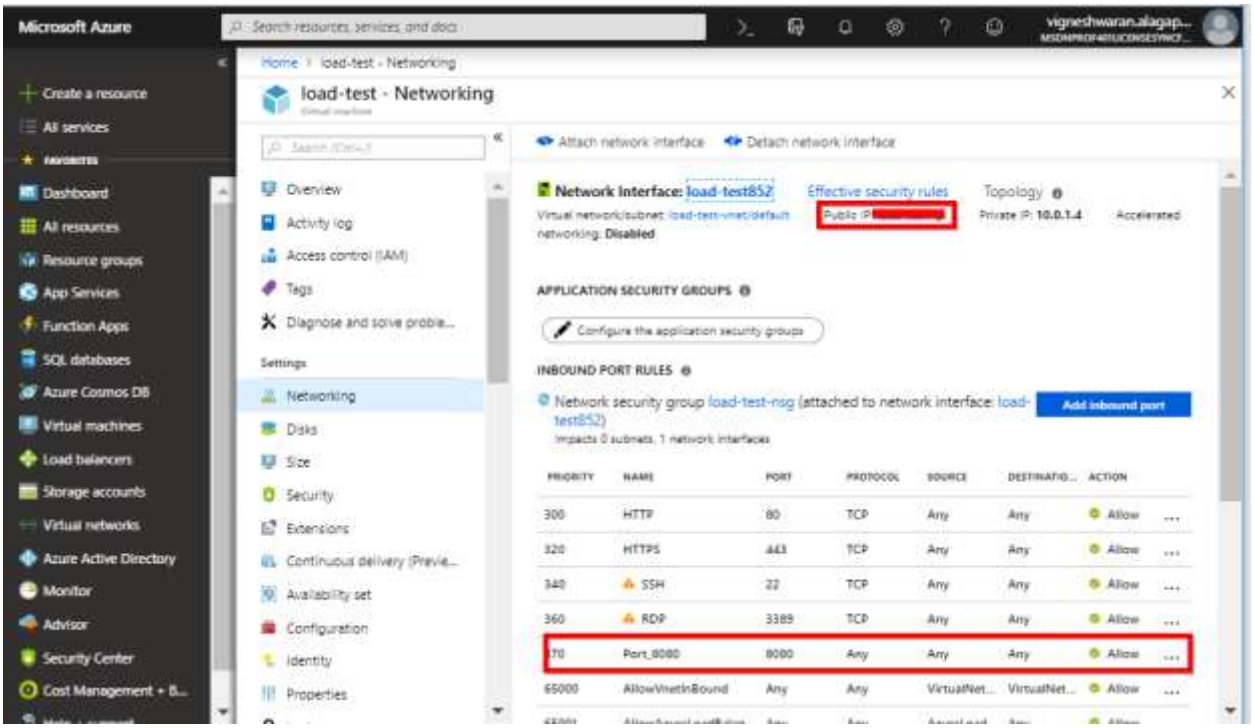
1. Login into the Azure portal and click the virtual machine that is used at present.
2. Select networking option in settings and click the Add Inbound Port Rule.



3. Then provide the port number, priority, and name, and proceed with OK button.



4. Now, the port get added.



Extensions

- **Extensions**: Add new features like configuration management or antivirus protection to the virtual machine using extensions. To learn more about extensions, click here.

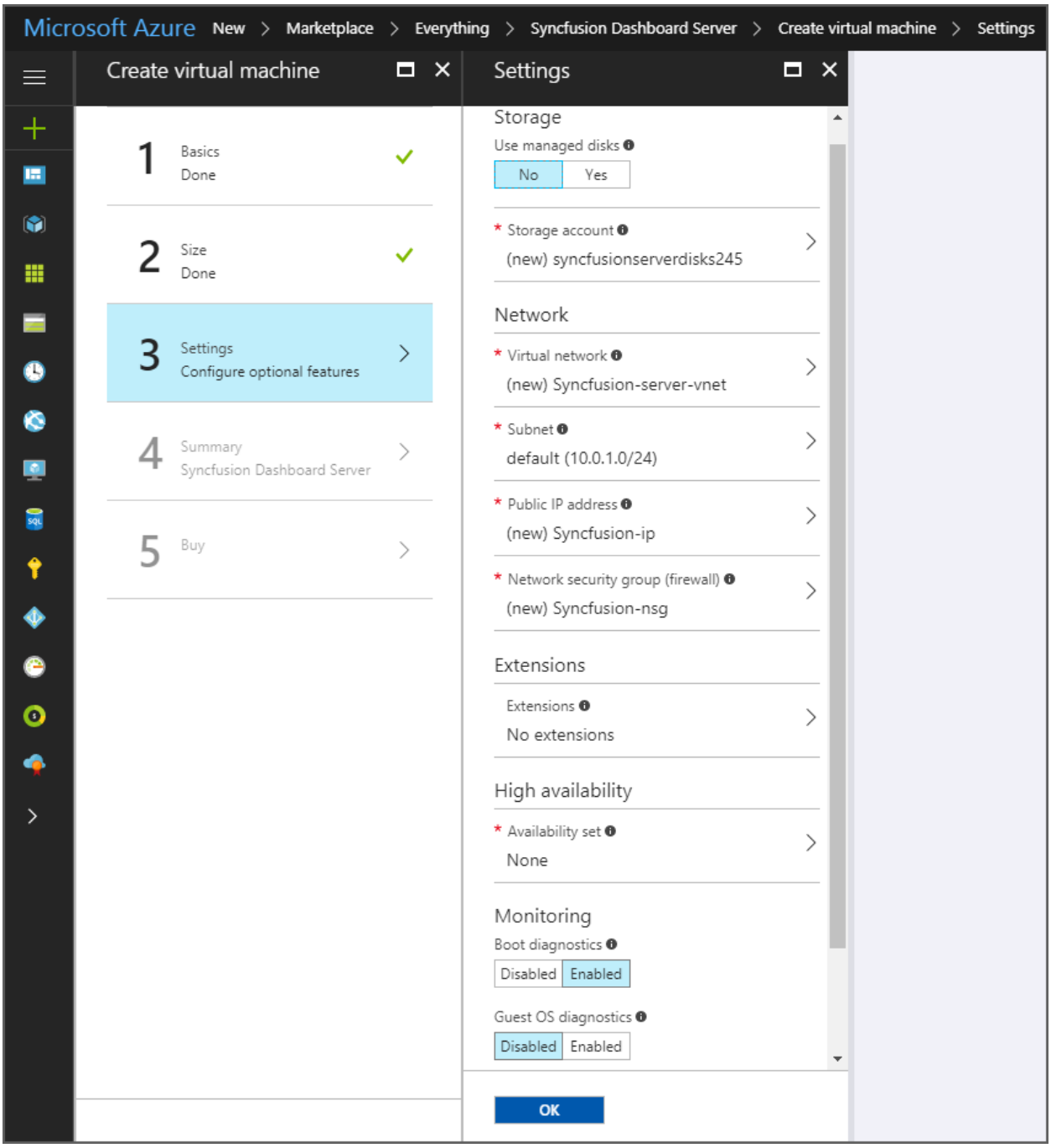
High availability

- **Availability set**: You can group two or more virtual machines in an availability set to provide redundancy to the application. This configuration ensures the availability of at least one virtual machine and meets the 99.95% Azure SLA, during the planned or unplanned maintenance event. The availability set of the virtual machine can not be changed after it is created.

Monitoring

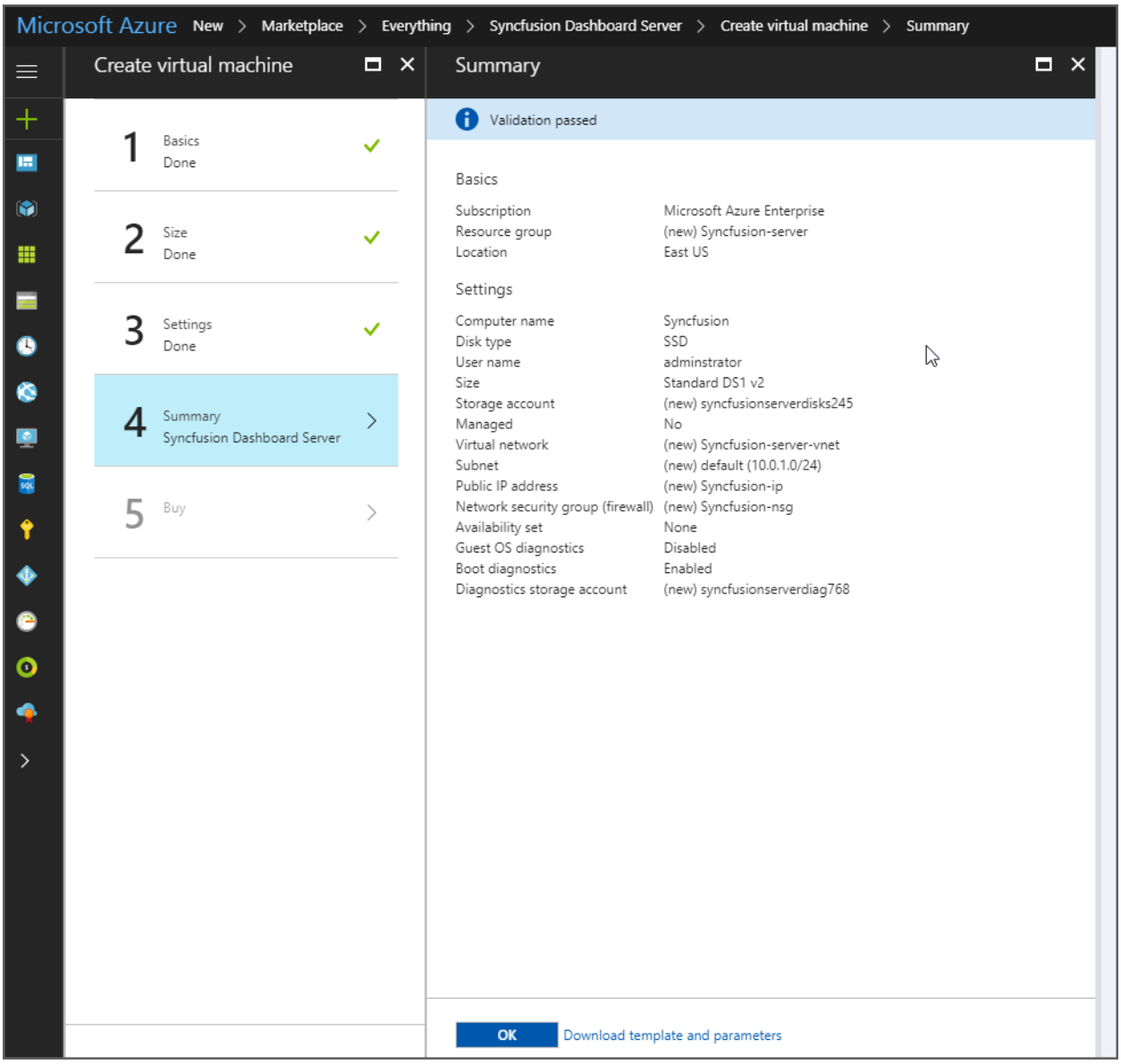
- **Boot diagnostics**: Capture serial console output and screenshots of the virtual machine running on the host to help diagnose startup issues.
- **Guest OS diagnostics**: Get metrics for every minute of the virtual machine. You can use them to create alerts and stay informed on the applications.
- **Diagnostics storage account**: Metrics are written to the storage account, so that you can analyze them with your own tools.

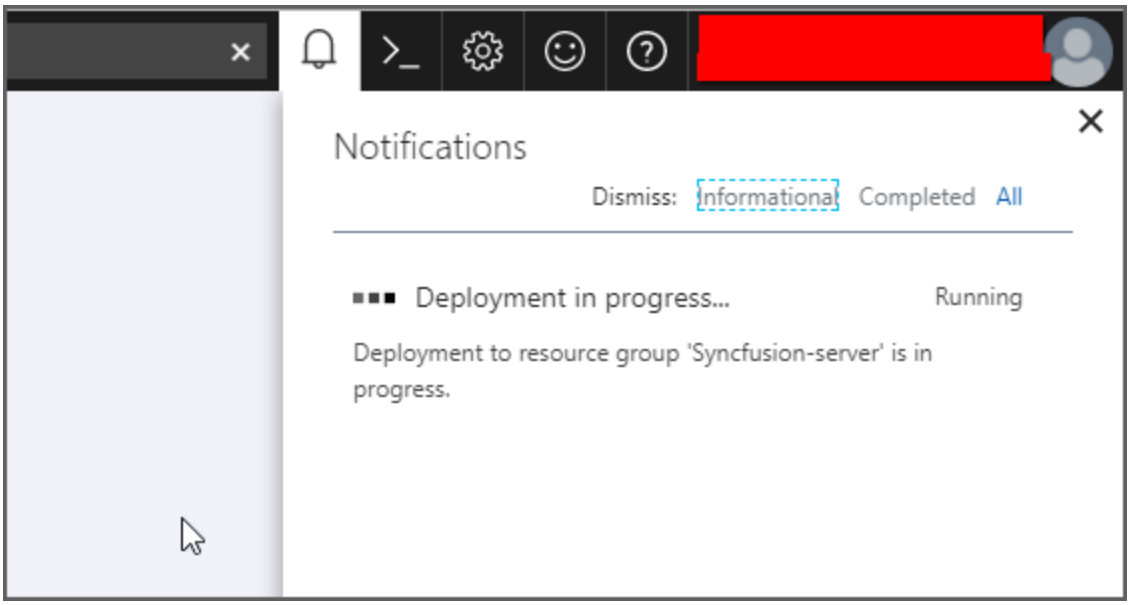
When all the changes are completed, click OK.



Summary blade

- On the summary page, click OK to start the Syncfusion dashboard server virtual machine deployment.



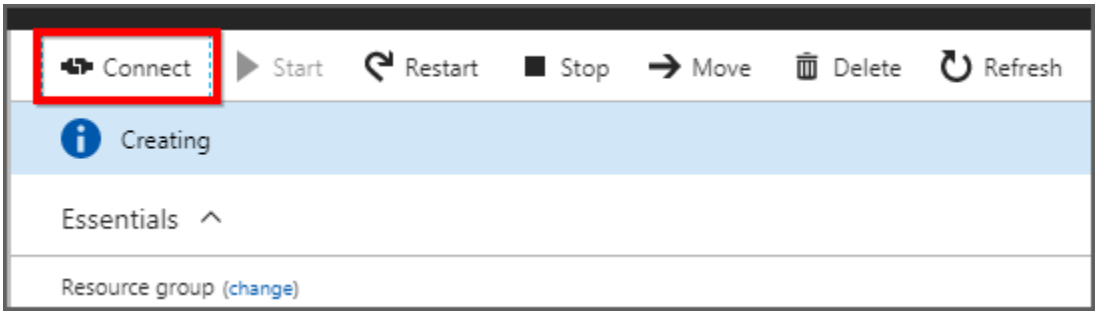


Connect to Syncfusion Dashboard Server virtual machine

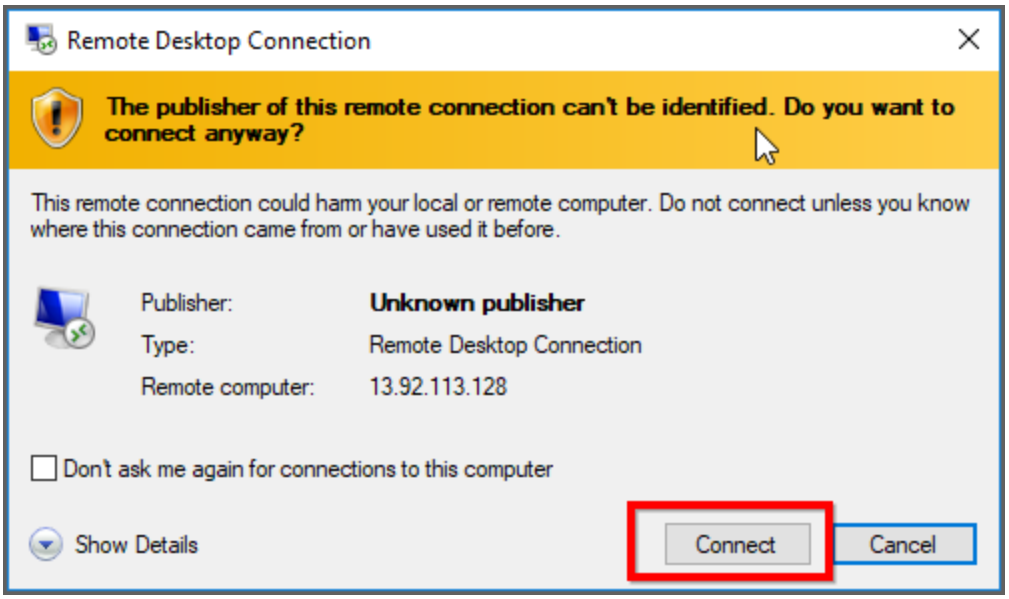
Once the deployment progress is completed, Dashboard Server VM can be connected through a Remote Desktop Connection (RDP).

Follow the below steps to connect to the virtual machine:

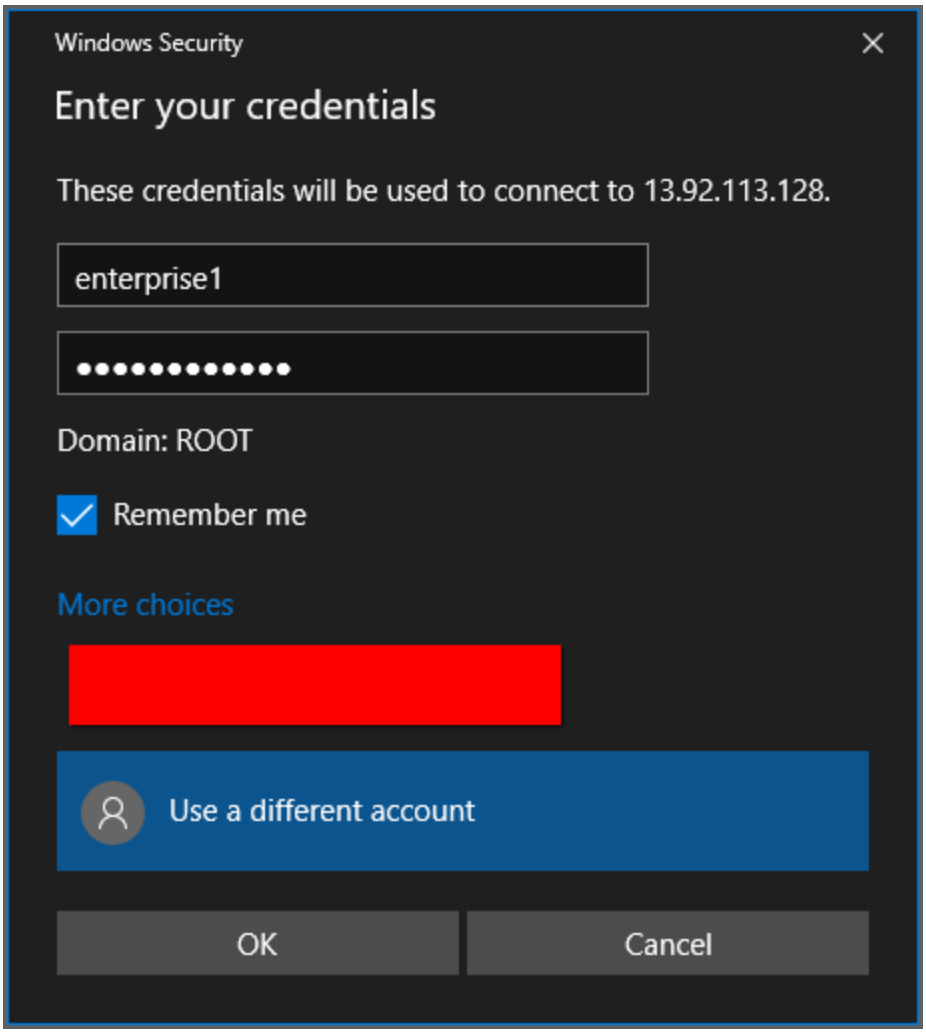
1. Click the connect on the virtual machine overview window. A Remote Desktop Protocol (.rdp) file will be downloaded from the Azure portal.



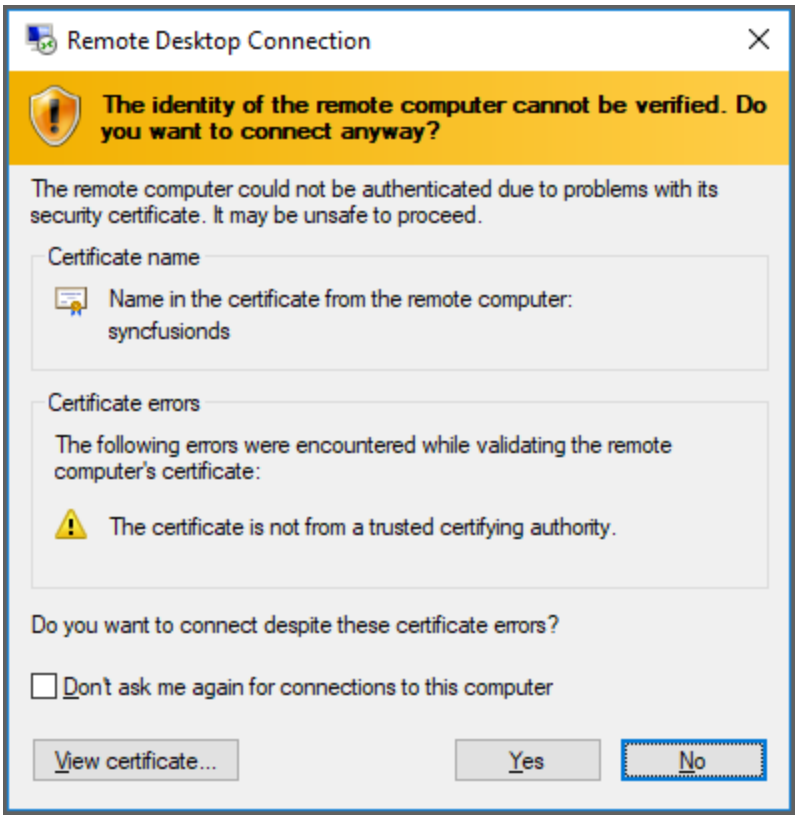
2. Open the .rdp file and click the continue for the unknown publisher warning.



3. Enter the credentials that you have given while deploying the VM as like below and click the OK.



4. On successful connection, the identity verification window will be displayed as shown below. Click OK to accept the certificate problems and connect to the virtual machine.



Run the Dashboard Server

Desktop shortcuts to start and stop the dashboard server can be found once connected to the virtual machine. By default, dashboard server is hosted in the 80 port in the IIS.

Follow the given steps to run the dashboard server:

1. Open the Start Syncfusion Dashboard Server shortcut to run the dashboard server.
2. As the dashboard server is not configured yet, application startup page will be shown.
3. Follow the steps in the link to do the application startup.

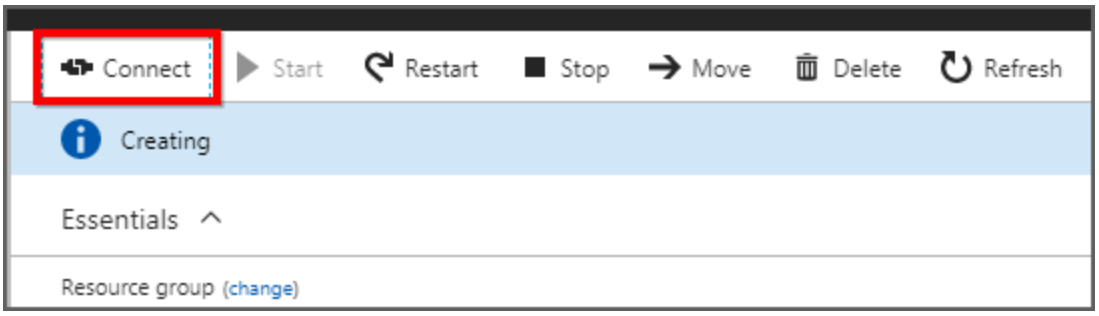
Upgrade to latest Syncfusion Dashboard Server version 3.2 along with User Management Server

Syncfusion Dashboard Server version 3.2 comes with a User Management Server, which is a separate application for managing your users and applications. To learn more about User Management Server, click here.

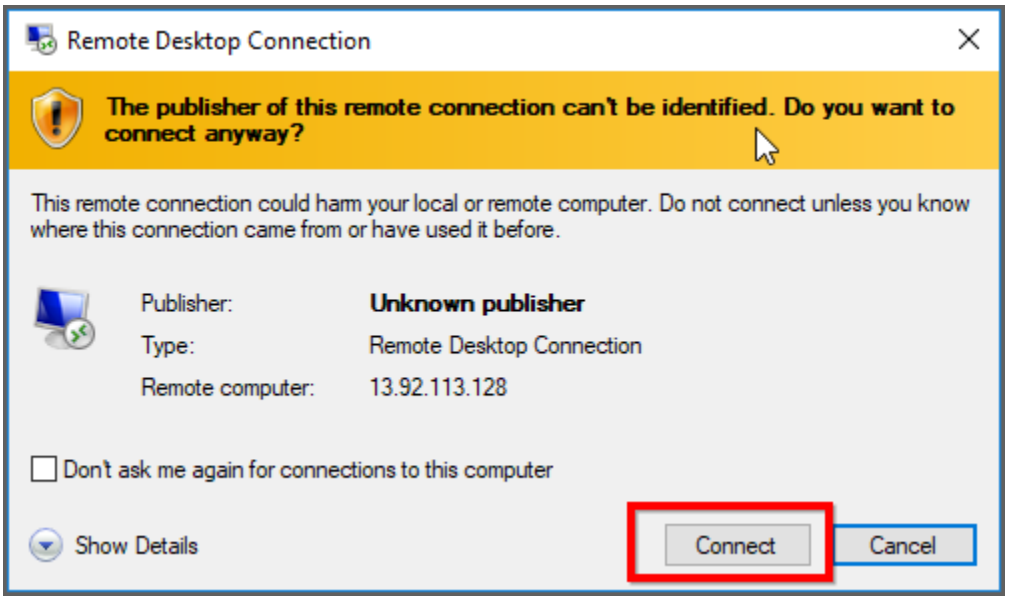
Upgrade to latest Syncfusion Dashboard Server version 3.2

Follow these steps to upgrade the Dashboard Server VM to the latest source from older versions:

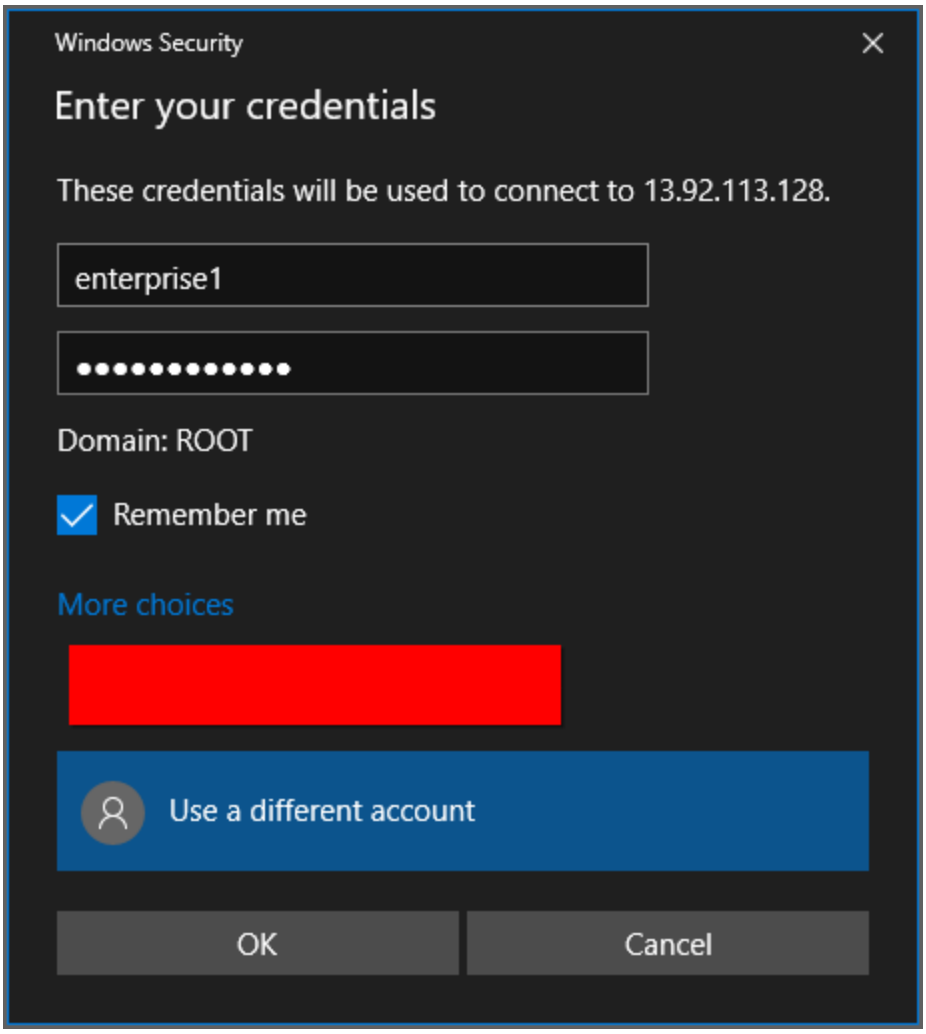
1. Connect Dashboard Server VM through a Remote Desktop Connection (RDP) by clicking **Connect** in the virtual machine window. A Remote Desktop Protocol (.rdp) file will be downloaded from the Azure portal.



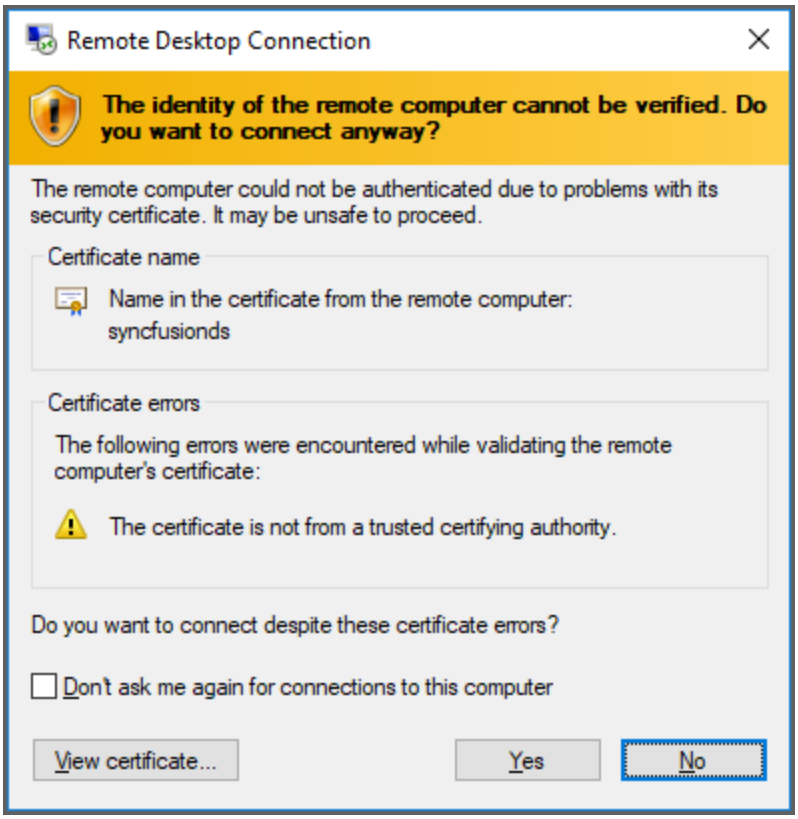
2. Open the .rdp file and click **Continue** for the unknown publisher warning.



3. Enter the credentials that you gave when deploying the VM as follows, and then click **OK**.



4. On successful connection, the identity verification window will be displayed as follows. Click **OK** to accept the certificate problems and connect to the virtual machine.



5. After connected to the virtual machine, follow the steps given here to upgrade the Dashboard Server to 3.2 version.

Application Startup

This topic describes how to configure the Syncfusion Dashboard Server.

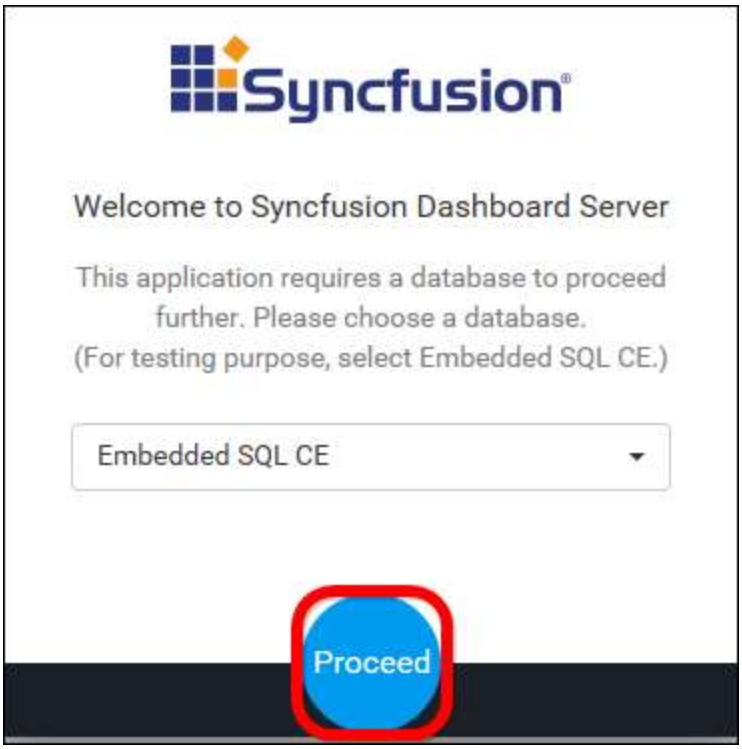
Application startup screen help you configure storage options and to register a new user.

Storage Options

Dashboard Server stores the user management data in the following databases as you select in the first screen.

- SQL CE
- SQL Server
- MySQL
- Oracle
- PostgreSQL, Azure-PostgreSQL
- AWS Aurora (Please click here to know how to configure AWS Aurora)
    1. ### Embedded SQL CE (For Testing purposes only)

Installed along with Dashboard Server Installer to easily set the environment up for testing purposes.

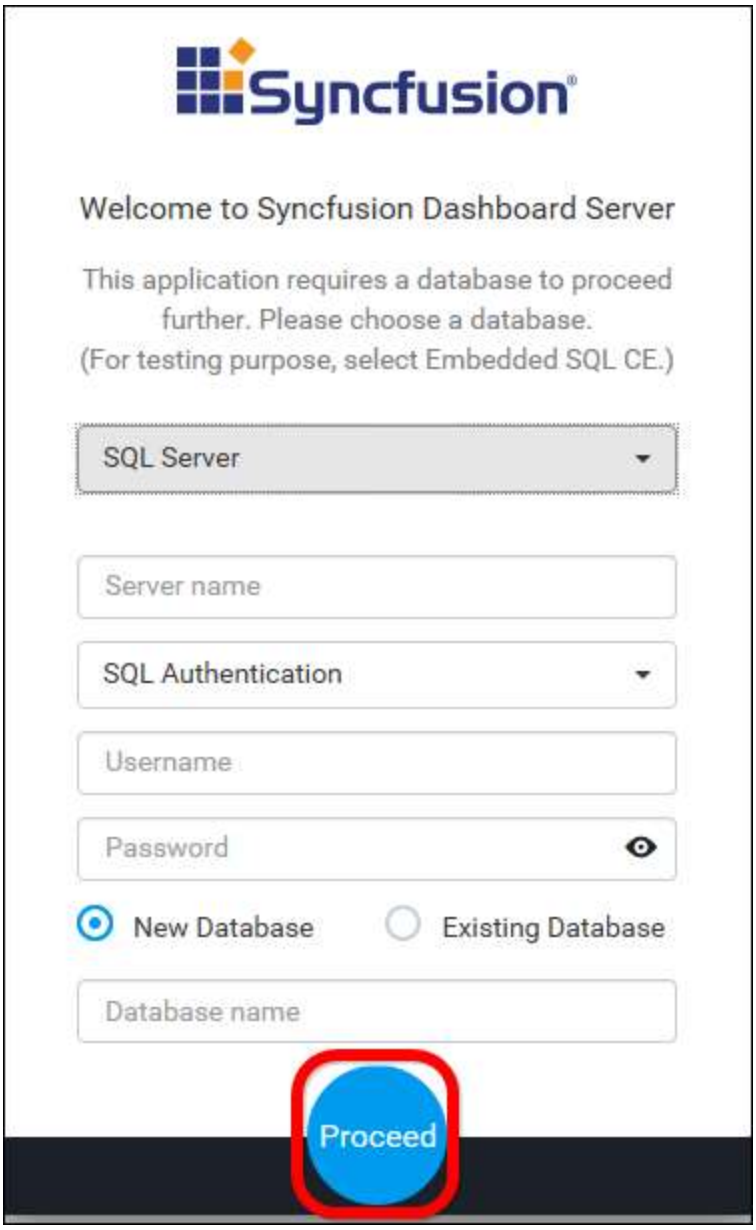


Note: Since it is an Embedded Database, we do not have option to create database from Azure App service.

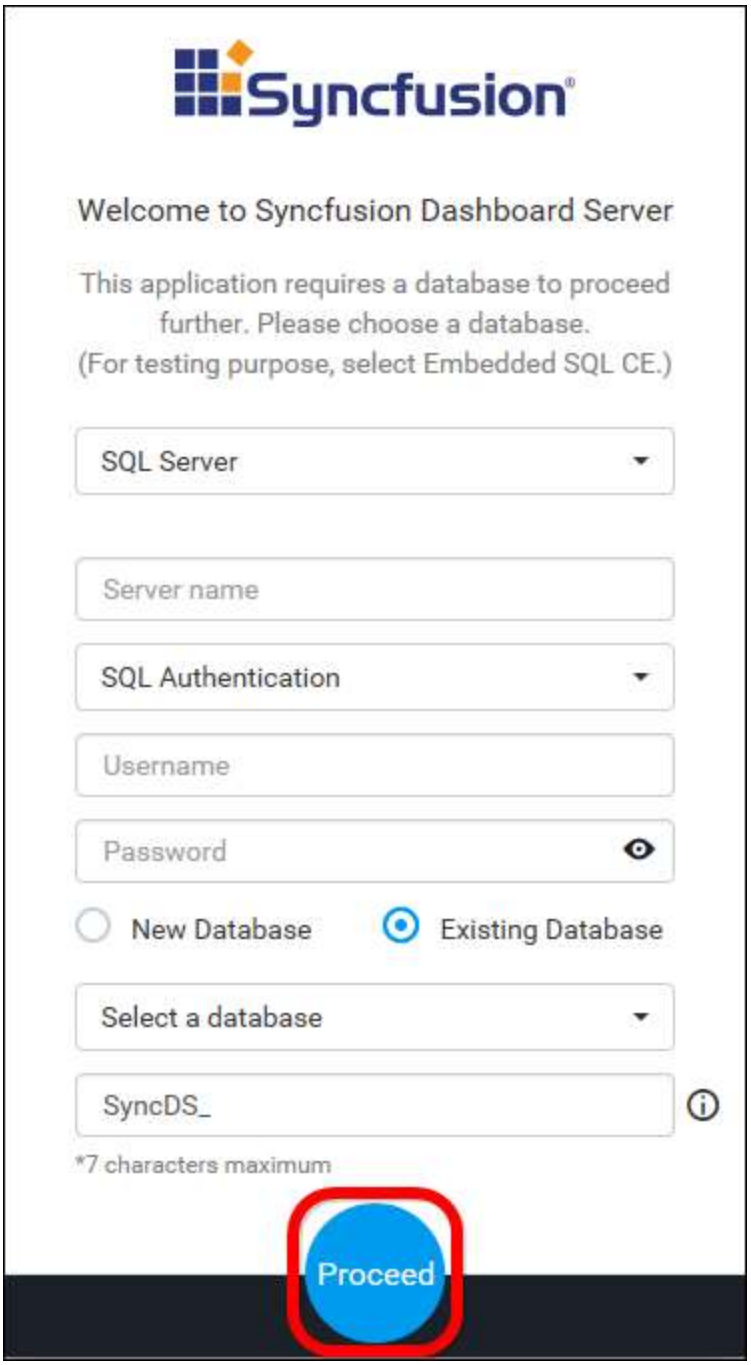
2. ### SQL Server

Can connect to the existing SQL Server instance with the below options.

- Create new Syncfusion Dashboard Server database.



- Use an existing database for Syncfusion Dashboard Server.
- Choose one of the database from Select a Database drop down for creating Dashboard Server tables in that database.
- In order to avoid table name conflicts, we have added a prefix SyncDS *by default. It can also be changed. If the prefix is empty, the default prefix SyncDS* is added.



Note: The credentials that is given to connect to the SQL Server instance must have permissions to

- Create Database
- Create Table
- Insert
- Update Table
- Alter Table
- Select
- Drop Table

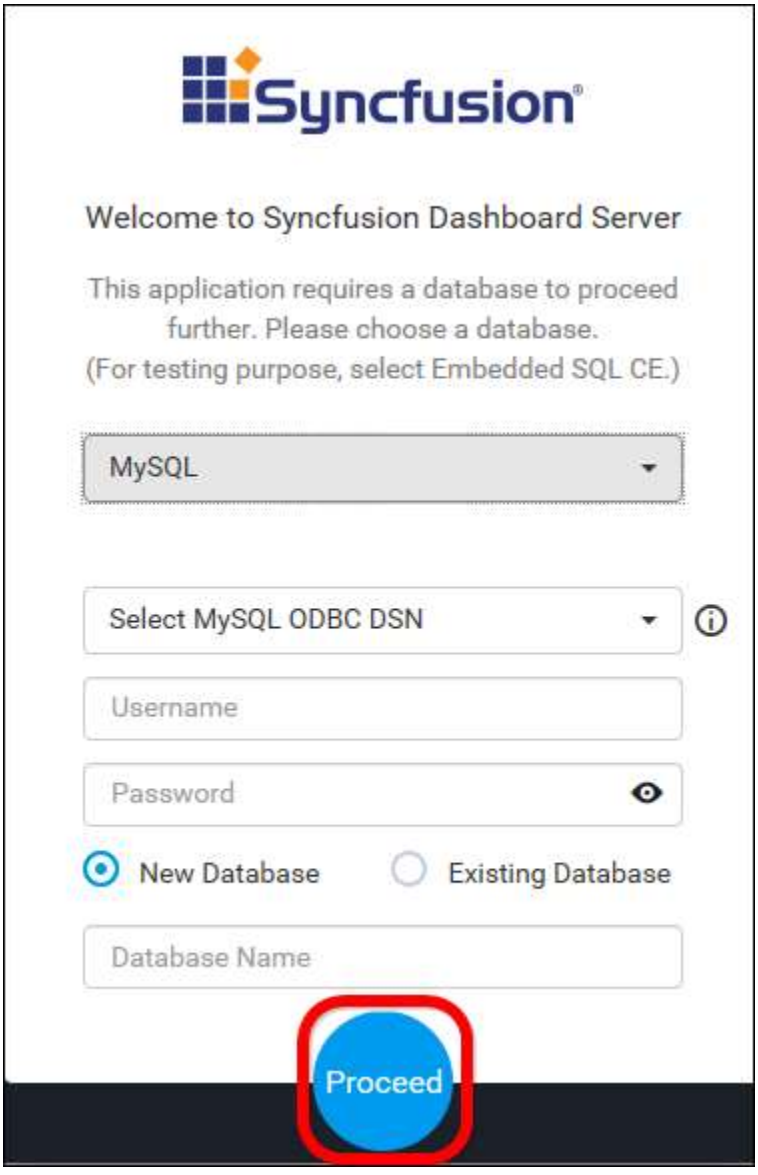
- Drop Database

The SQL Server within an elastic pool can also be supported in the Syncfusion Dashboard Server.

3. ### MySQL

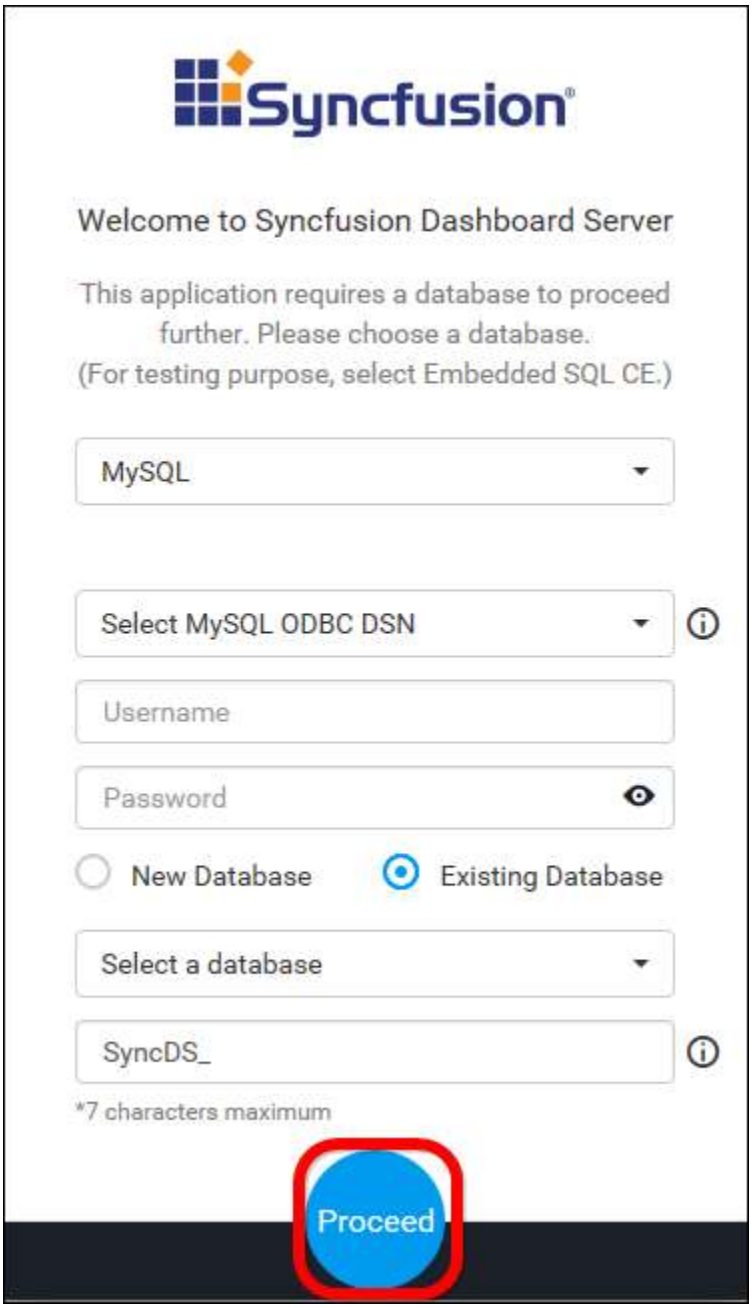
Can connect to the existing MySQL instance with the below options.

- Create new Syncfusion Dashboard Server database.



- Use an existing database for Syncfusion Dashboard Server.
- Choose one of the database from Select a Database drop down for creating Dashboard Server tables in that database.

- In order to avoid table name conflicts, we have added a prefix SyncDS *by default. It can also be changed. If the prefix is empty, the default prefix "SyncDS*" is added.



Note: The credentials that is given to connect to the MySQL instance must have privileges to

- Create
- Delete
- Insert
- Update
- Alter
- Select

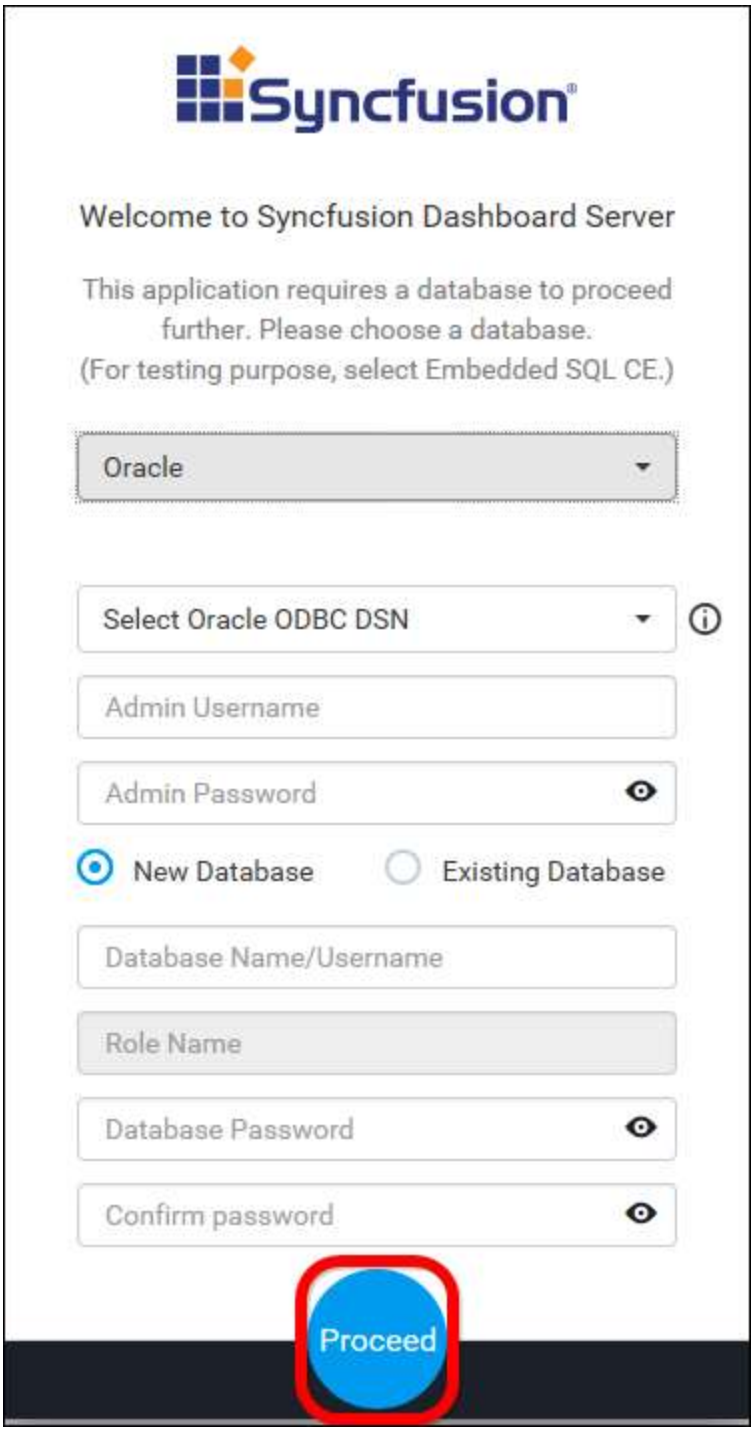
- Drop
- Show Databases

We do not have option to create database from Azure App service.

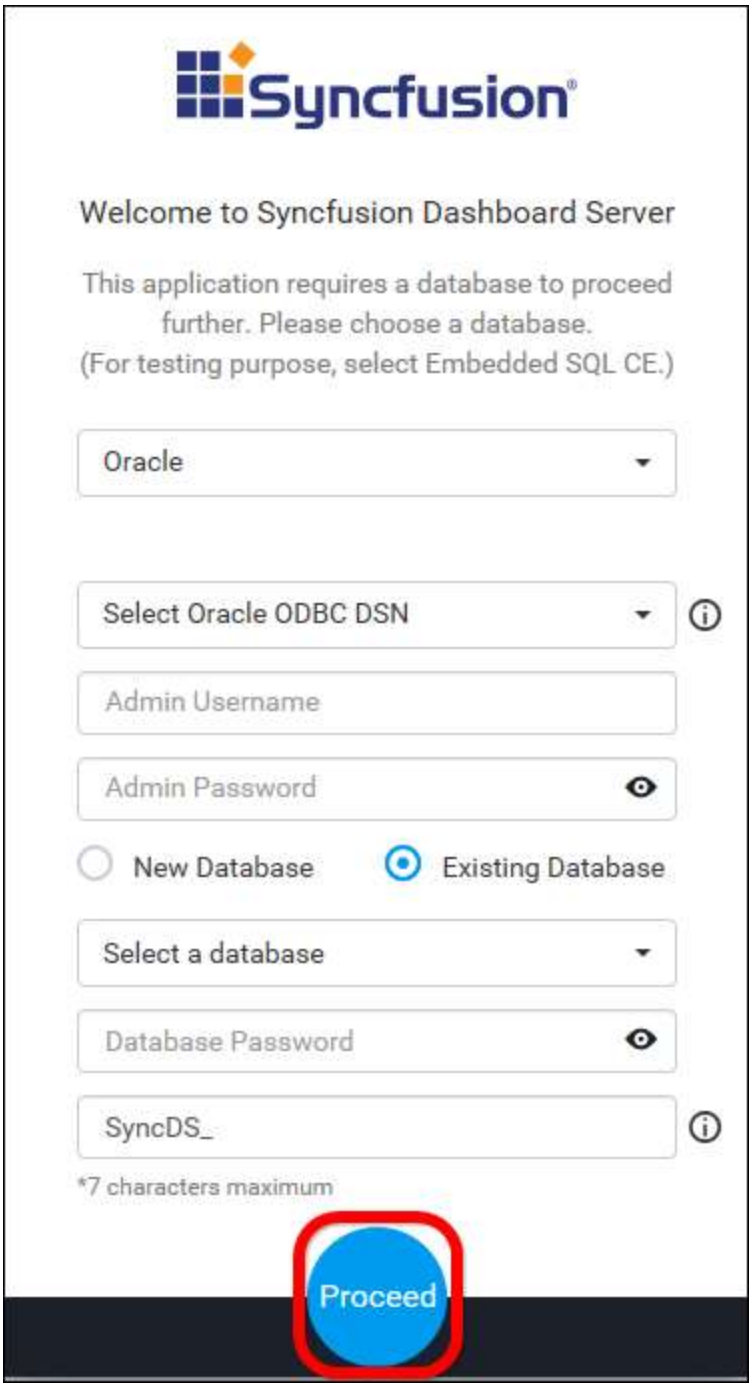
4. ### Oracle

Can connect to the existing Oracle instance with the below options.

- Create new Syncfusion Dashboard Server database.



- Use an existing database for Syncfusion Dashboard Server.
- Choose one of the database from Select a Database drop down for creating Dashboard Server tables in that database.
- In order to avoid table name conflicts, we have added a prefix SyncDS *by default. It can also be changed. If the prefix is empty, the default prefix "SyncDS*" is added.



Note: The credentials that is given to connect to the Oracle instance must have permissions to

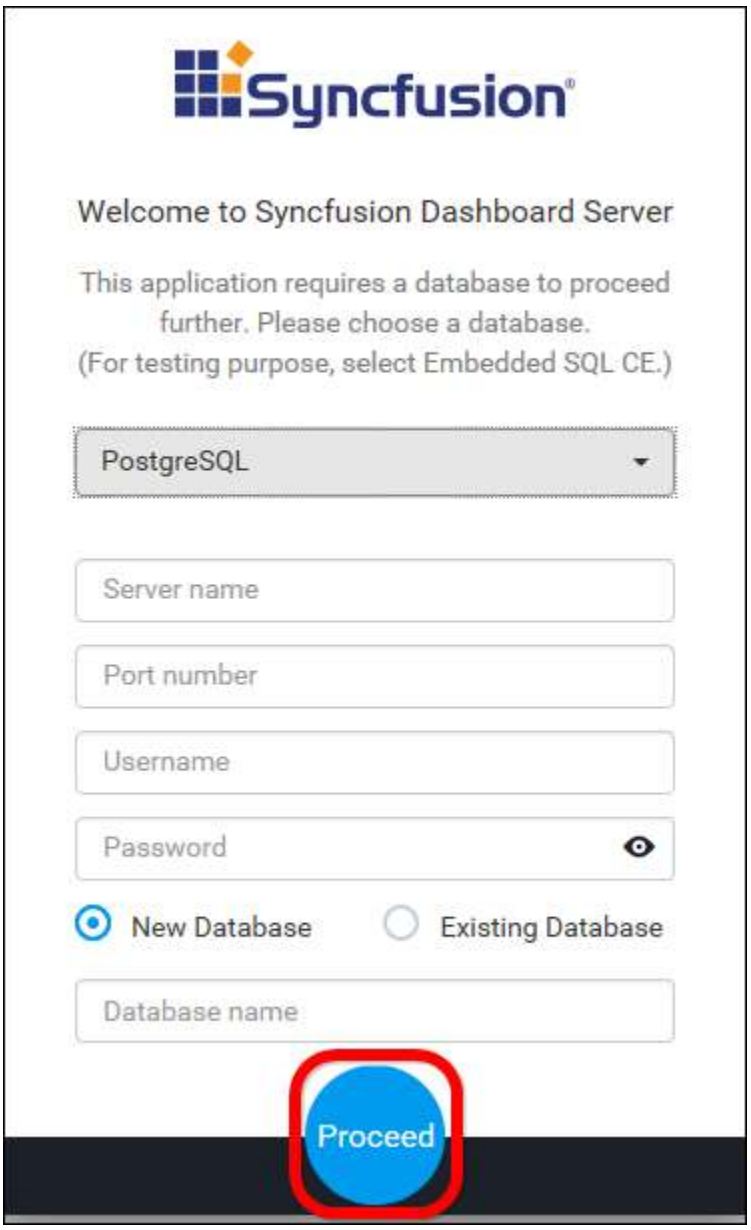
- Create Database
- Create Role
- Grant Role
- Grant Permission
- Drop Database

We do not have option to create database from Azure App service

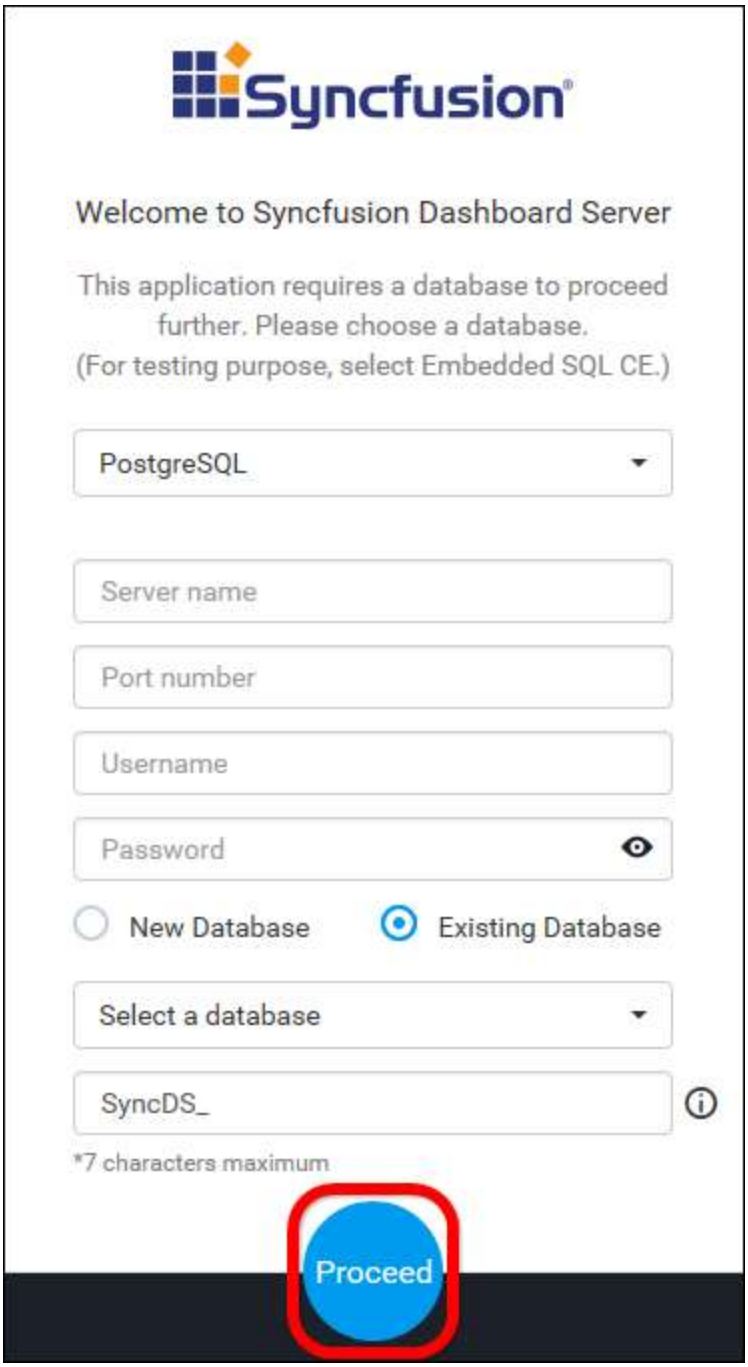
5. ### PostgreSQL

Can connect to the existing PostgreSQL instance with the below options.

- Create new Syncfusion Dashboard Server database.



- Use an existing database for Syncfusion Dashboard Server.
- Choose one of the database from Select a Database drop down for creating Dashboard Server tables in that database.
- In order to avoid table name conflicts, we have added a prefix SyncDS *by default. It can also be changed. If the prefix is empty, the default prefix "SyncDS*" is added.



Note: The credentials that is given to connect to the PostgreSQL instance must have permissions to

- Create Database
- Create Table
- Insert
- Update Table
- Alter Table
- Select
- Drop Table

- Drop Database

Dashboard Server is deployed in the below location by default.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\

For example, C:\Syncfusion\Dashboard Server\DashboardServer.Web\

Dashboard Server stores the dashboard, data source and widget files that are uploaded to the server in the following location as file system.

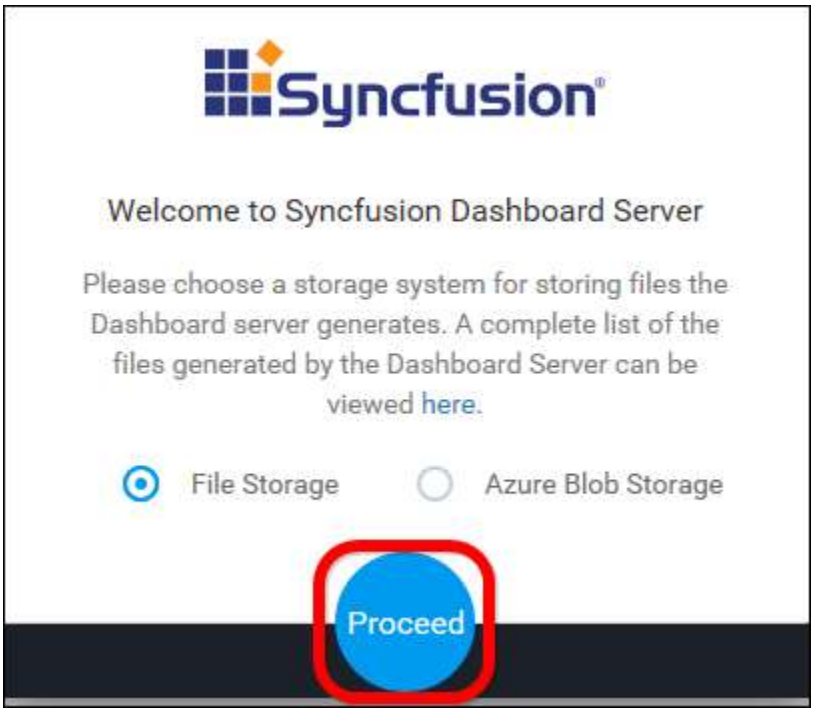
{Windows*Drive}**Syncfusion**Dashboard Server**DashboardServer.Web**App*Data\Resources

Also, an option to create a database from the Azure PostgreSQL has been provided.

Storage System

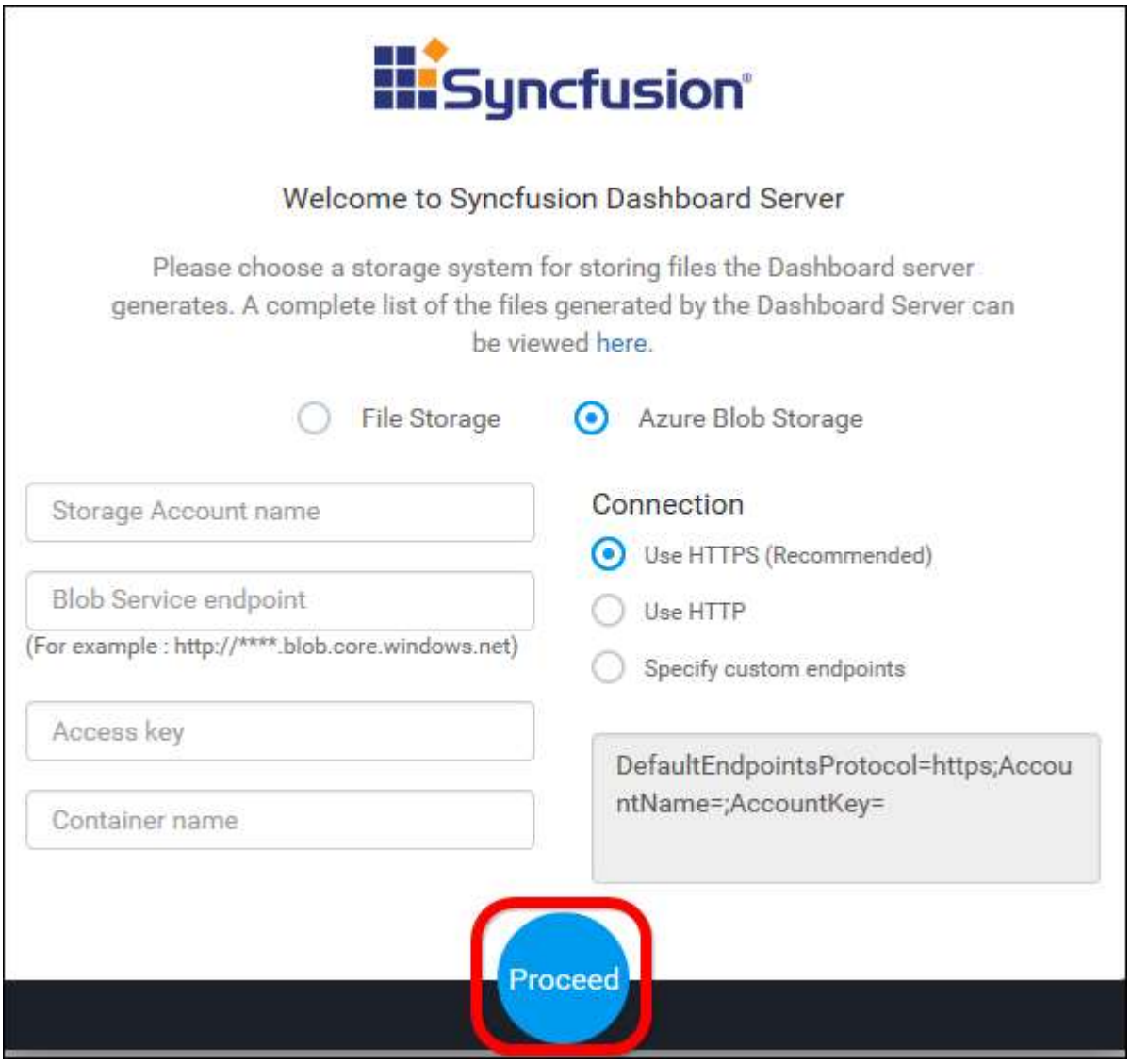
File Storage

The default system is File Storage, in this the Dashboard Server stores the dashboards and data sources that are uploaded to the server in the following location in the installed machine.



Blob Storage

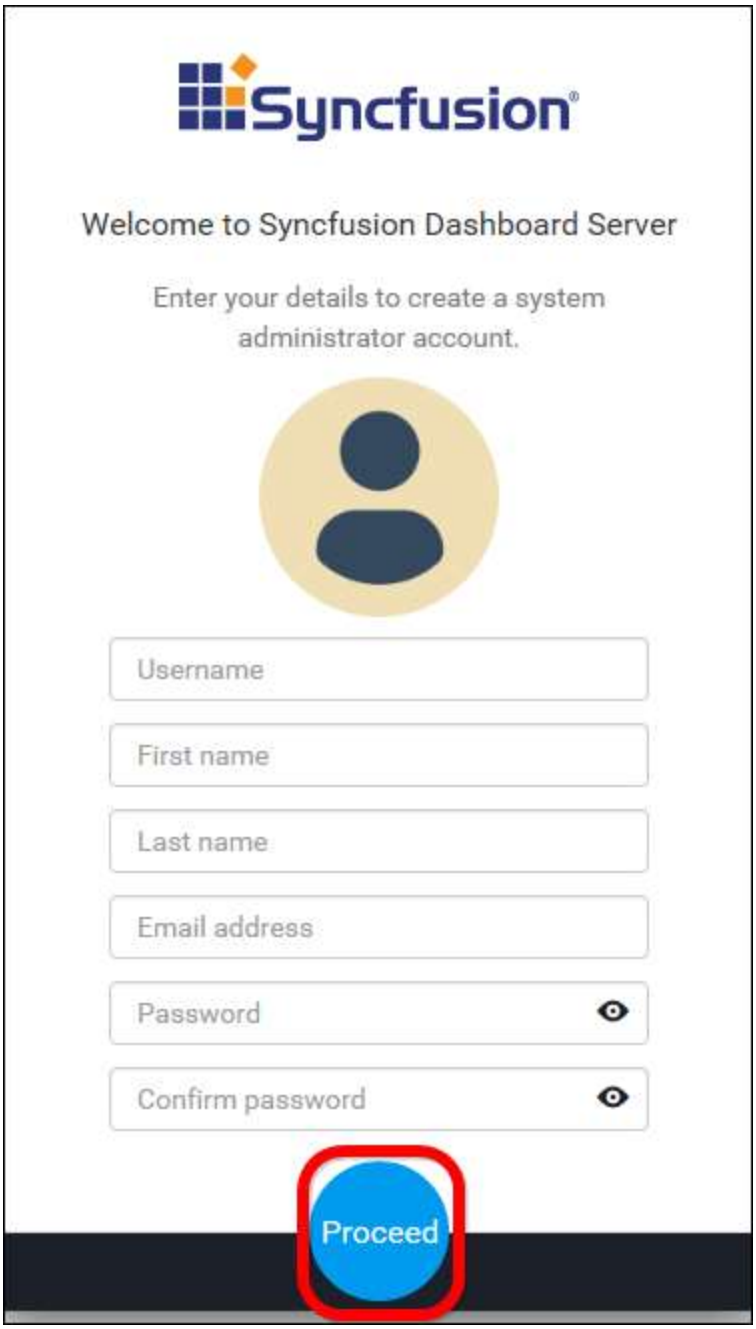
If the Dashboard Server wants to store the dashboards and data sources that are uploaded to the server in the blob storage location, need to provide details shown in below figure.



You can find the details on How to configure the Azure Blob here

New User - System Administrator

New user should be created to access the dashboard server with the details mentioned in the below image.



While creating this new user account, a new group System Administrator is also created.

By default, System Administrator group have permission to do the below

- Create Dashboards
- Create Data Sources
- Create Schedules
- Create Users
- Create Groups
- Manage Permissions for users and groups

The new user account created is assigned to this group by default.

Compose Dashboards

Working with Data Source

Supported Data Connections in Syncfusion Dashboard

This page describes the data connections supported in **Syncfusion Dashboard Server** through direct and extract modes and the data formats supported by REST-based web services.

- **Direct Mode** - Source data connected and processed directly by Syncfusion Dashboard.
- **Extract Mode** - Source data connected and imported to intermediate database and processed from it by Syncfusion Dashboard.

The following tables list the supported data connections categorized by the department/industry and type.

Sales

Data Connections	Direct Mode	Extract Mode	Data Formats
ChartMogul			JSON
Copper			JSON
Freshsales			JSON
Harvest			JSON
Insightly			JSON
Keen IO			JSON
Kissmetrics			JSON
Pipedrive			JSON
PipelineDeals			JSON
ServiceNow			JSON
Zendesk			JSON
Zendesk Sell			JSON

Marketing

Data Connections	Direct Mode	Extract Mode	Data Formats
ActiveCampaign			JSON
Ask Nicely			JSON
Campaign Monitor			JSON
Chartbeat			JSON
GoSquared			JSON
Mixpanel			JSON
Nicereply			JSON

Support

Data Connections	Direct Mode	Extract Mode	Data Formats
CallRail			JSON
Freshdesk			JSON
Freshservice			JSON
Help Scout			JSON
LiveAgent			JSON
Twilio			JSON

SQL

Data Connections	Direct Mode	Extract Mode
Amazon Aurora		
Amazon Redshift		

Azure SQL Data Warehouse		
Google Cloud SQL		
MariaDB		
MemSQL		
Microsoft SQL Server		
MySQL		
Oracle		
PostgreSQL		

NoSQL

Data Connections	Direct Mode	Extract Mode
MongoDB		

Cloud Storage

Data Connections	Direct Mode	Extract Mode	Data Formats
Azure Blob			JSON, XML, CSV
Dropbox			JSON

Files

Data Connections	Direct Mode	Extract Mode
Excel		
CSV		
JSON		

XML		

Finance

Data Connections	Direct Mode	Extract Mode	Data Formats
Chargebee			JSON
Chargify			JSON
Fusebill			JSON
Recurly			XML
Stripe			JSON

Mobile Analytics

Data Connections	Direct Mode	Extract Mode	Data Formats
42 matters			JSON

Advertising

Data Connections	Direct Mode	Extract Mode	Data Formats
Flurry			JSON

Project Management

Data Connections	Direct Mode	Extract Mode	Data Formats
GitHub			JSON
GitLab			JSON
Intervals			JSON
JIRA			JSON
Lighthouse			JSON, XML

Microsoft Outlook Calendar			JSON
New Relic			JSON
Toggl			JSON

Email Campaign

Data Connections	Direct Mode	Extract Mode	Data Formats
MailChimp			JSON
SparkPost			JSON
SendGrid			JSON

REST Services

Data Connections	Direct Mode	Extract Mode	Data Formats
OData			JSON, XML
RSS Feeds			JSON
Web			JSON, XML, CSV

Web Analytics

Data Connections	Direct Mode	Extract Mode	Data Formats
Optimizely			JSON
SEOmonitor			JSON

Social Media Management

Data Connections	Direct Mode	Extract Mode	Data Formats
Sendible			JSON

Survey

Data Connections	Direct Mode	Extract Mode	Data Formats
SurveyGizmo			JSON

Create a support ticket

If you have questions specific to Dashboard Platform and don't find answers throughout the resources, you can raise a support ticket to Syncfusion at:

https://www.syncfusion.com/support/directtrac/incidents/newincident

While creating support ticket, try to provide detailed explanation of your requirement with necessary screenshots or videos. For reporting any bug, consider providing the exact replication procedure and necessary log files. It will help our support team to reach you earlier with a better solution.

Submit your feedback

Your feedback is valuable to us. Please submit your suggestions or comments for product enhancements in future releases at:

https://www.syncfusion.com/feedback/dashboard-platform

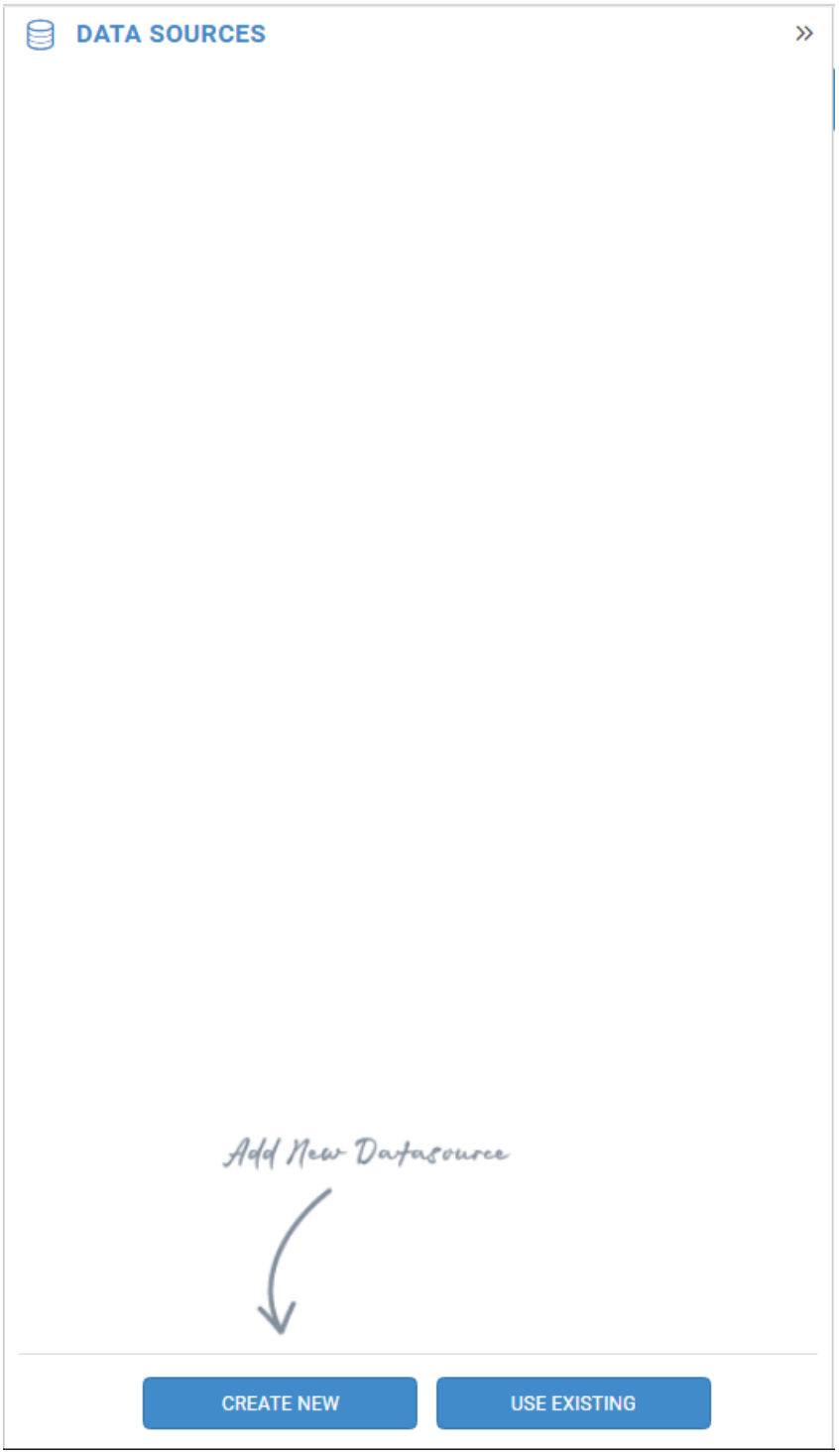
Data Transformation

Creating a New Data Source

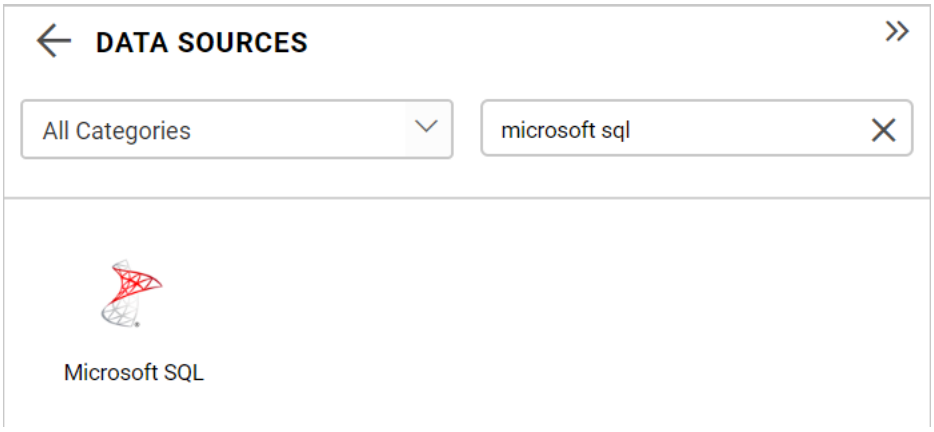
To bind data to a widget, minimum of one data source is required. Follow these steps to create a data source:

1. Click Data Source button in the configuration panel. The data panel opens.

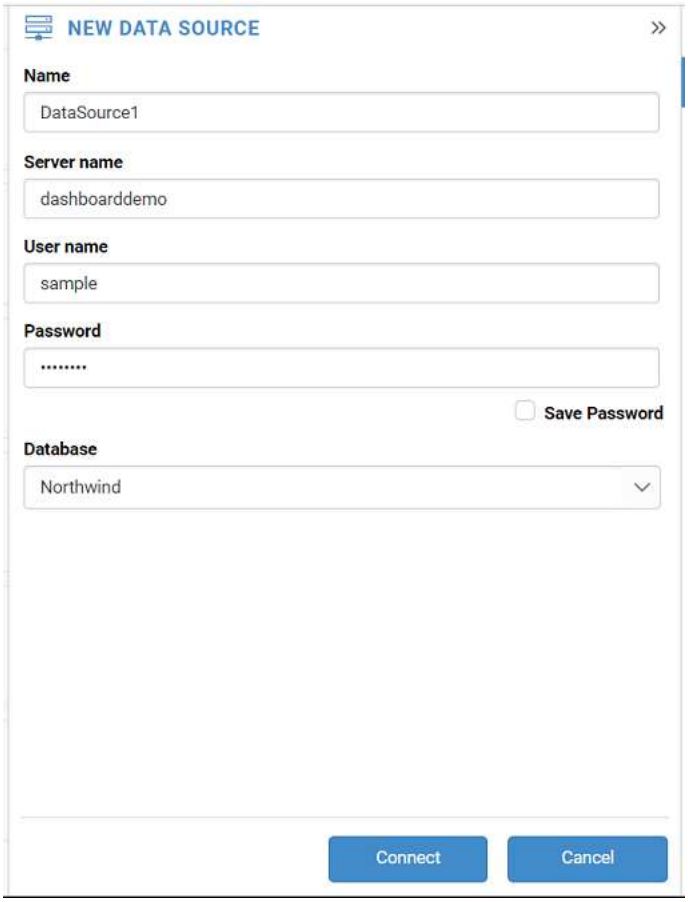
2. Click CREATE NEW to launch a new connection from the connection type panel.



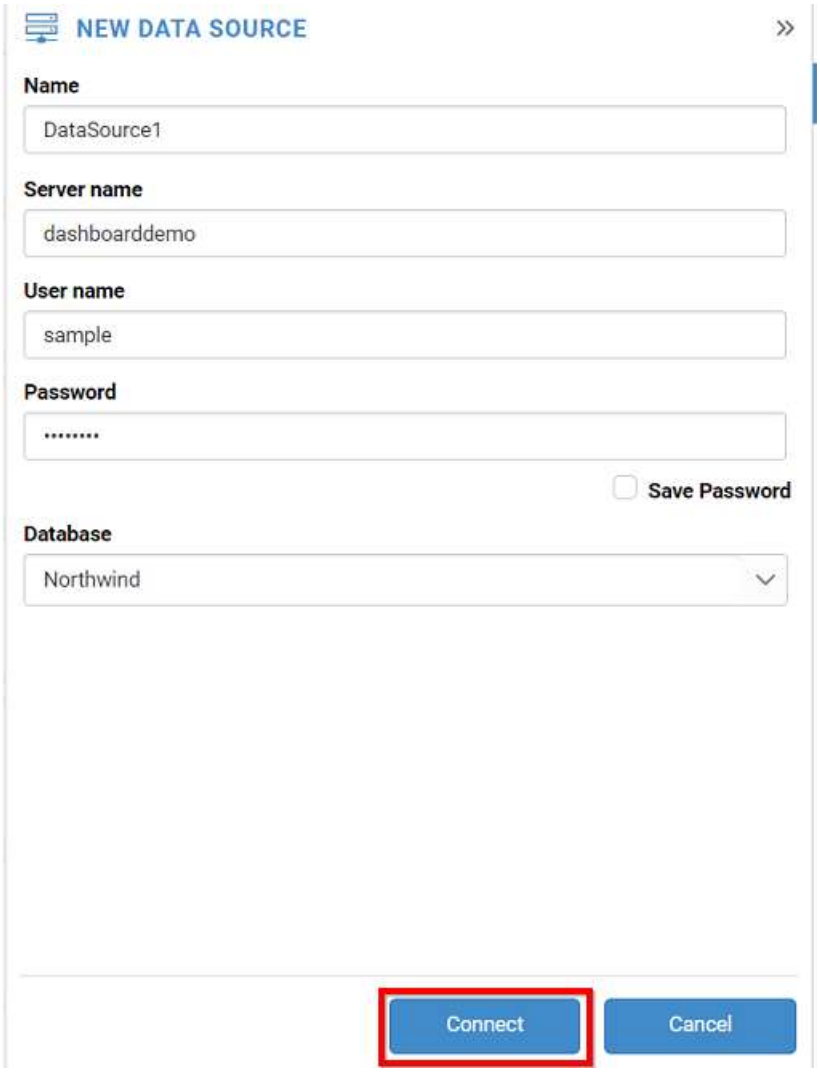
3. In the connection type panel, select any one (say Microsoft SQL connection type for demonstration) of the listed connection types shown in the following image.



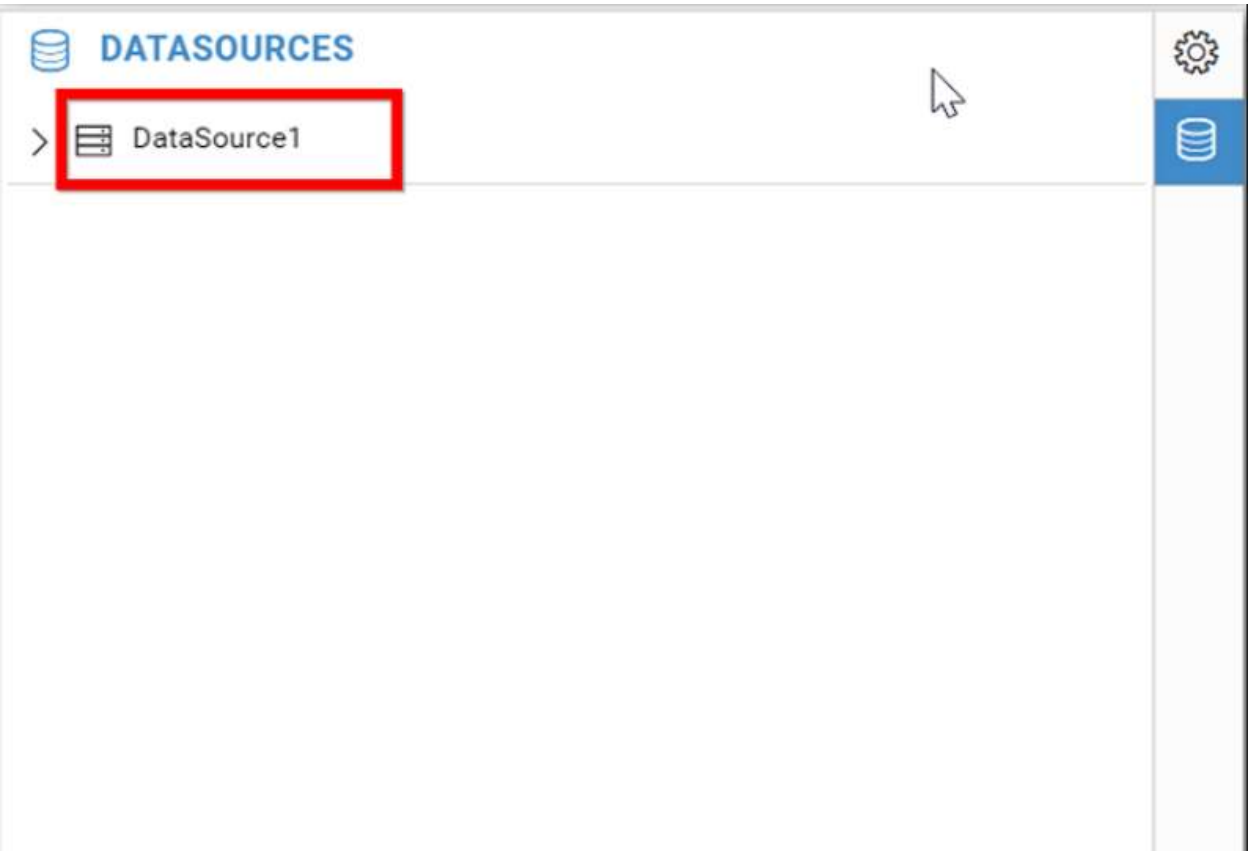
4. In the NEW DATA SOURCE configuration panel, fill the connection type and related details.



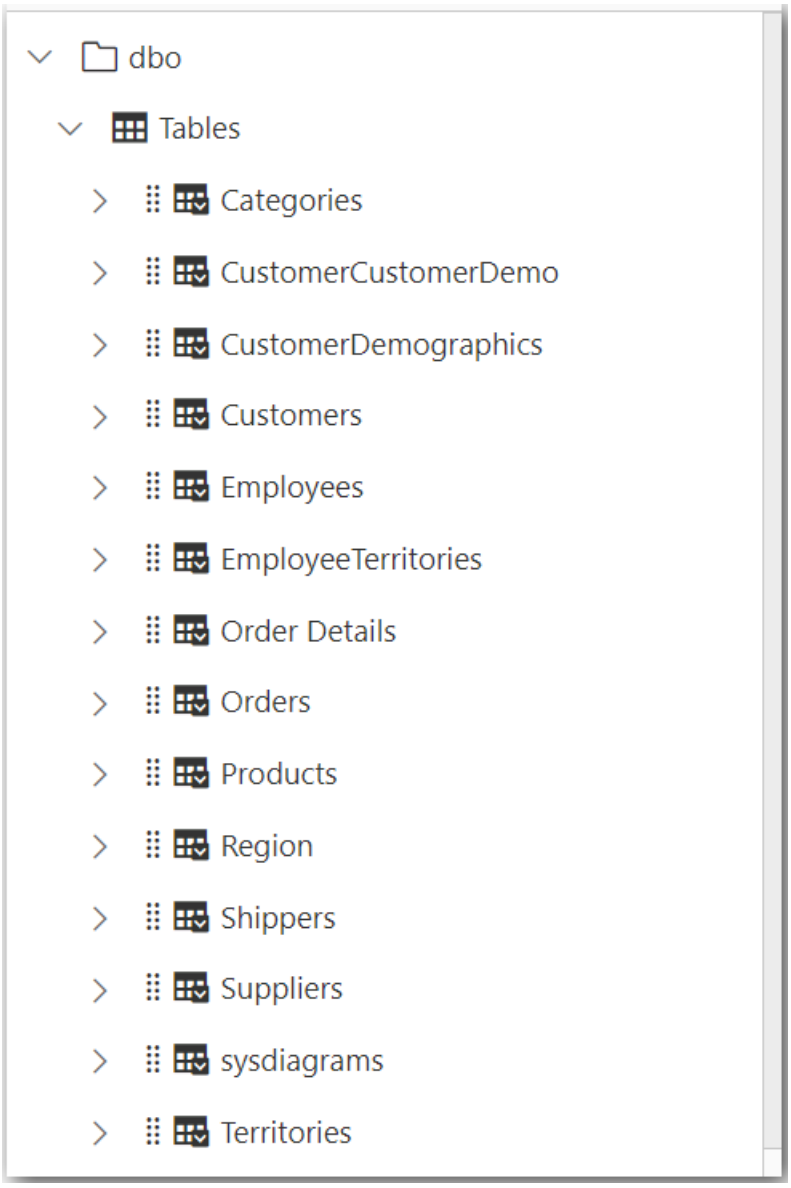
5. Click Connect in the NEW DATA SOURCEconfiguration panel.



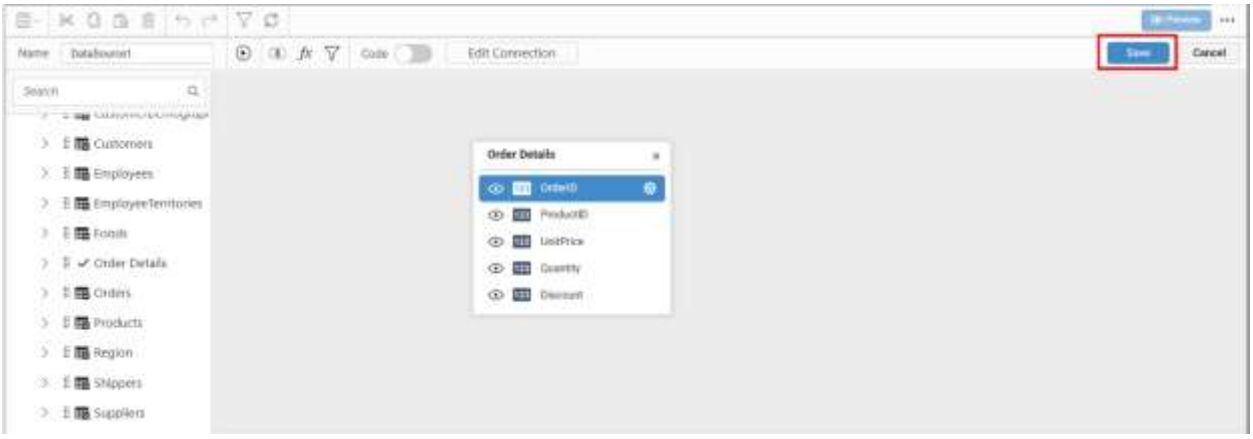
Now, the following view will be displayed.



6. You can drag and drop the tables or views in the data design view by expanding the tree view.



7. Click Save button to save the data source with valid name.



[Learn how to edit a data source]

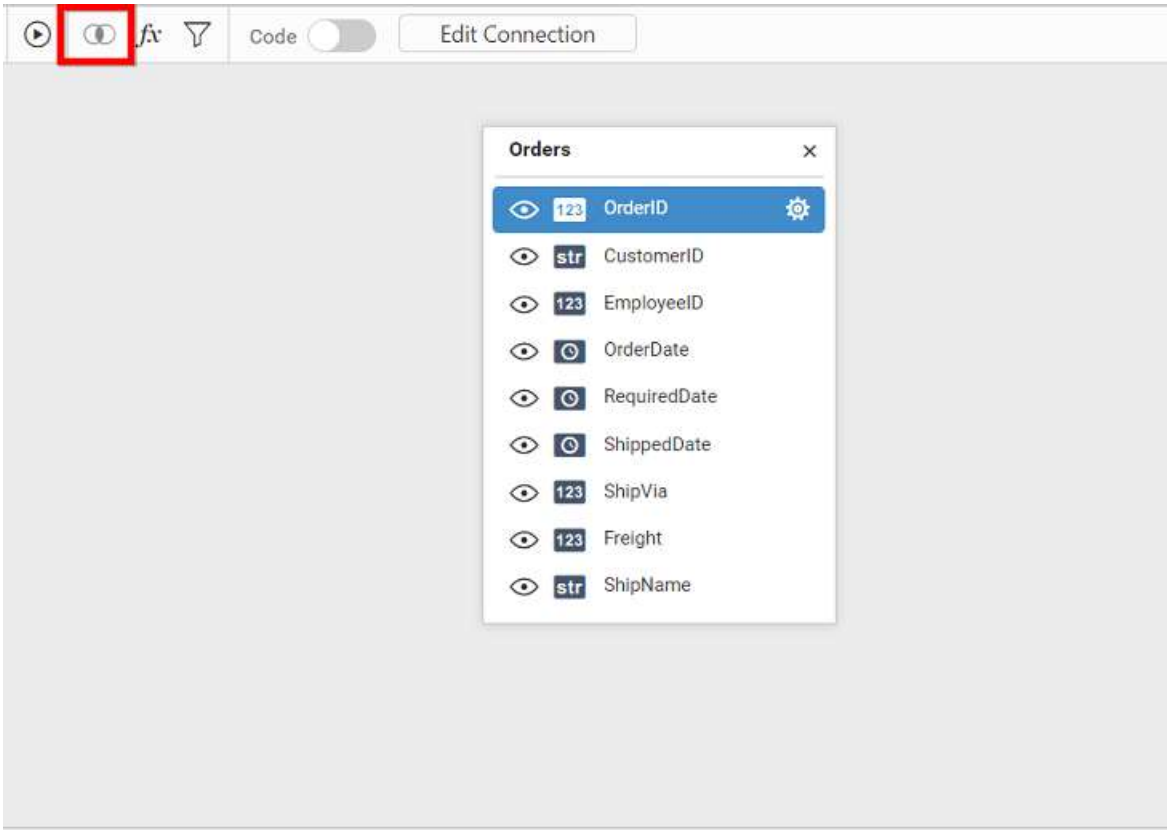
[Learn how to delete a data source]

Post your message

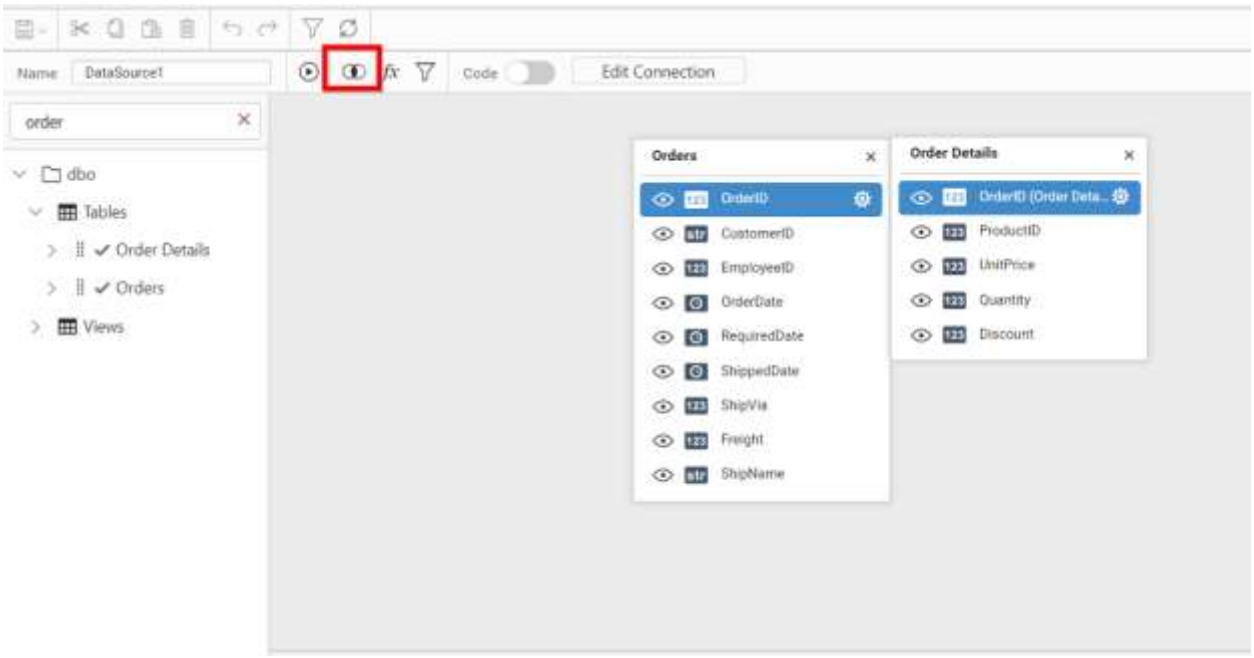
If you are still not able to find the information that you are looking for in the self-help resources mentioned above then please [contact us] and send your requirements.

Joining Tables

Joining of tables is required when you are going to use more than one table in your data source design. The join icon (highlighted below) in the tools pane at data design view will be in disabled state, if there was only one table found dropped in table design view like below:

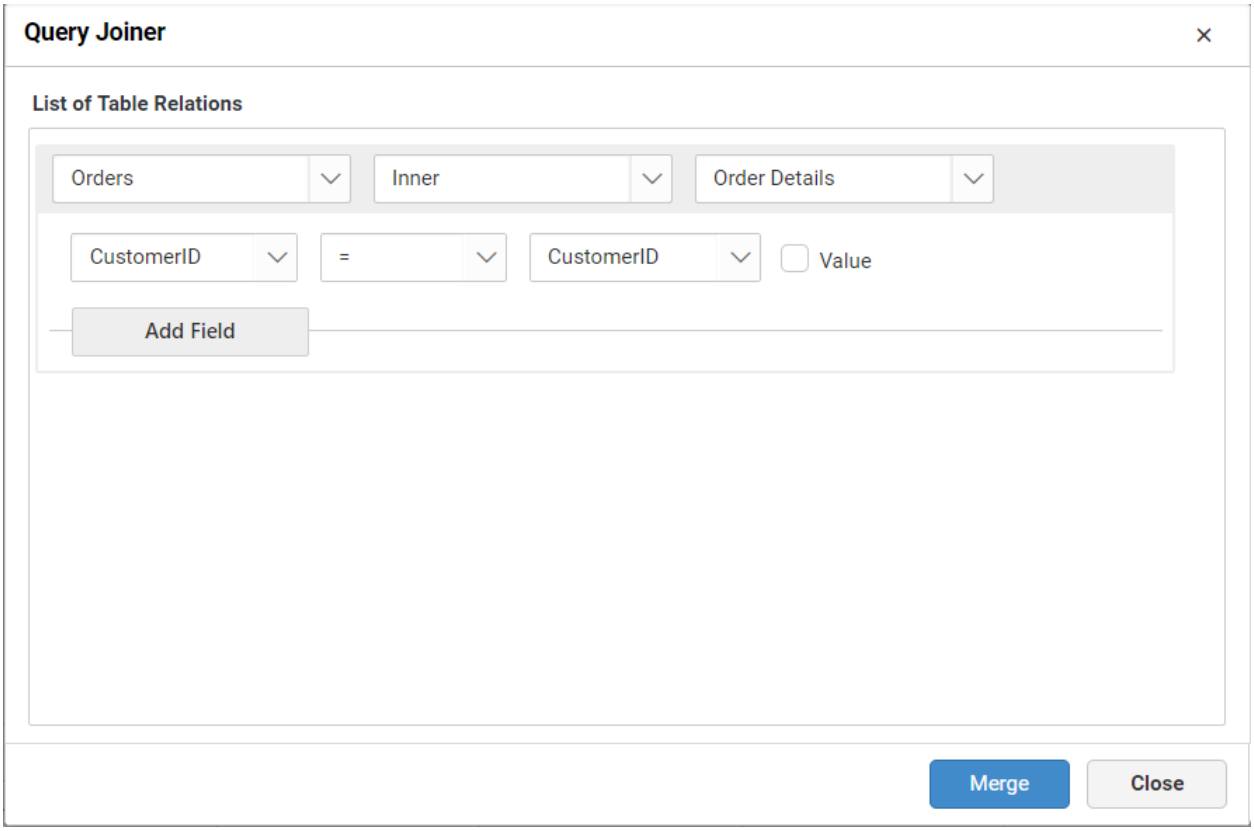


It will get enabled once you drop the 2nd table like below:



Adding a join condition

If the subsequent table being dropped, has any of its column as foreign key in any of the already dropped tables, the joining will take place automatically. Else, it will prompt the join editor like below to let you define the keys (columns) to join between this table and any one of the already dropped tables.

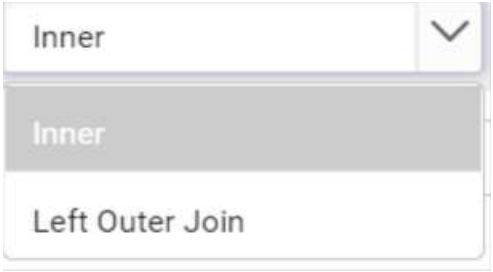


In the above screen shot, the LeftTable illustrate the list of table dropped already. The RightTable illustrate the table which you have dropped recently and that requires to set up a relation with any of the previously dropped tables. The drop-down list below to that, represent the join condition. You can add multiple join condition for single table relation just by click Add Field button.

The join type, compare operator and relational operator to make relationship between the two tables, can be defined through the options available in join editor.

Join Types

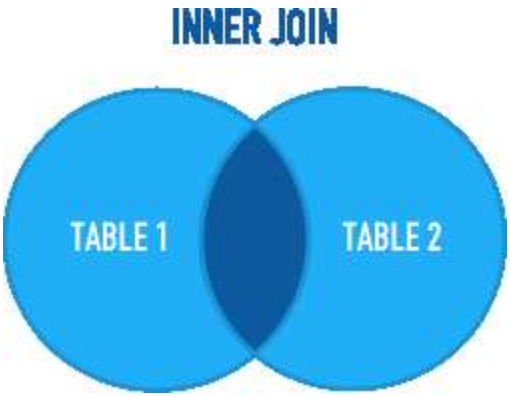
Two types of joins can be made between tables in join editor. They are Left Outer Join and Inner Join.



Inner Join

INNER JOIN will return the records from two or more tables, while records are matching in both the tables.

An inner join of Table1 and Table2 gives the result of Table1 intersect Table2, i.e. the inner part of a Venn diagram intersection.



For example, consider the below two tables.

Table1

| Supplier*Id* | *Supplier*Name |

|---|---|

| 100 | James |

| 101 | John |

| 102 | Robert |

| 103 | Michael |

Table2

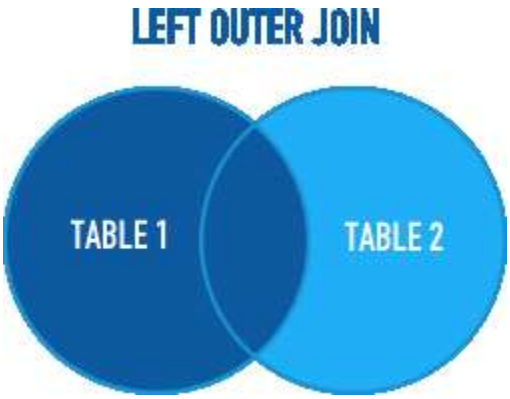
Order*Id* / *Supplier*Id	Order_Date
20125	100	09/21/2017
20126	101	09/22/2017
20127	104	09/23/2017

If we join (INNER JOIN) Table1 and Table2 based on Supplier_Id column and equals (=) as comparison operator, then Syncfusion Dashboard will return the result like below.

Supplier*Id* / *Supplier*Name	Order*Id* / *Supplier*Id(Table2)	Order_Date
100	James	20125	100	09/21/2017
101	John	20126	101	09/22/2017

Left outer join

LEFT OUTER JOIN will return all record from the left table and the matched records from the right table. The result is NULL from the right table, if there is no match.



For example, consider the below two tables.

Table1

Supplier*Id* / *Supplier*Name
100	James
101	John
102	Robert
103	Michael

Table2

Order*Id* / *Supplier*Id	Order_Date
20125	100	09/21/2017

| 20126 | 101 | 09/22/2017 |

| 20127 | 104 | 09/23/2017 |

If we join (LEFT OUTER JOIN) Table1 and Table2 based on Supplier_Id column and equals (=) as comparison operator, then Syncfusion Dashboard will return the result like below.

| Supplier*Id* | *Supplier*Name | Order*Id* | *Supplier*Id(Table2) | Order_Date |

|---|---|---|---|---|

| 100 | James | 20125 | 100 | 09/21/2017 |

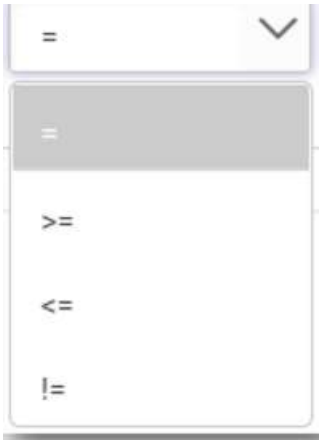
| 101 | John | 20126 | 101 | 09/22/2017 |

| 102 | Robert | | | |

| 103 | Michael | | | |

Join Condition

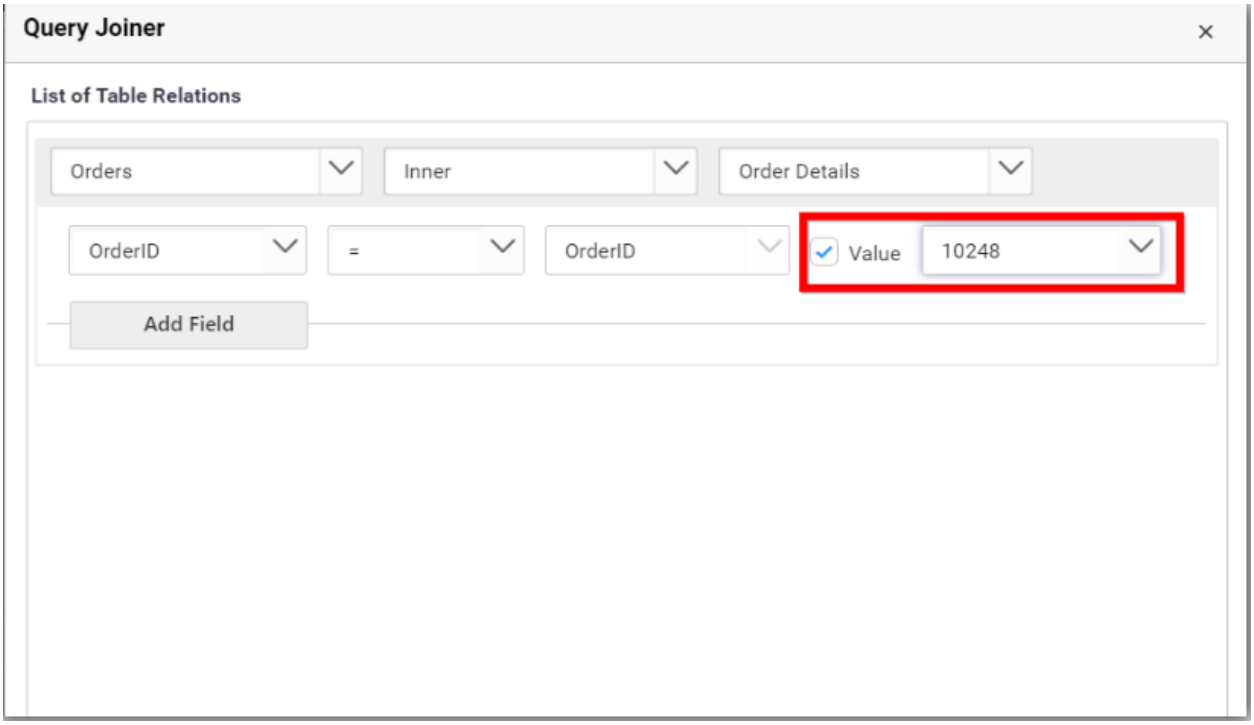
You can define a condition for joining two tables through any of the compare operator for comparing the values of the two columns (one from each table) by which relation between tables need to be made.



Join Condition Relationship

You can define the relationship for joining, with multiple condition, in the condition selection block.

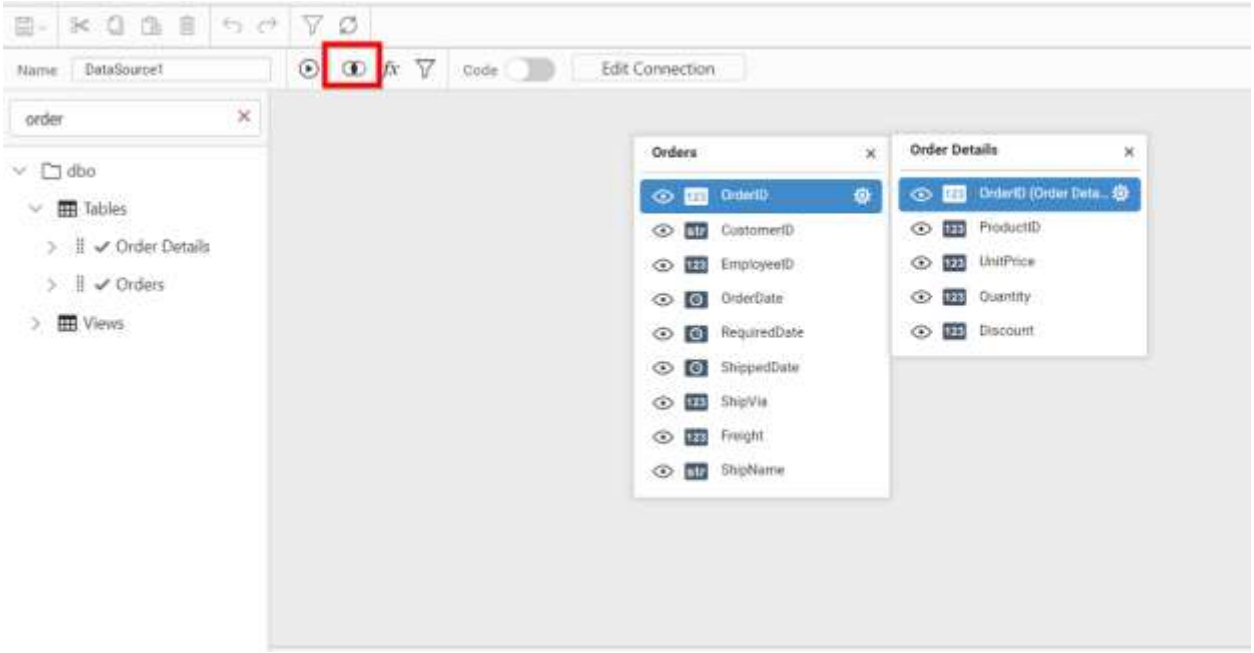
You can also create condition using a constant value instead of choosing column as right operand to join tables.



Updating a join condition

Update an existing join condition through selecting that in the top table and then edit the mapping between columns through interacting with columns list, join type and compare operator.

If you are not at the join editor, it can be invoked through clicking the highlighted icon below in the data design view.



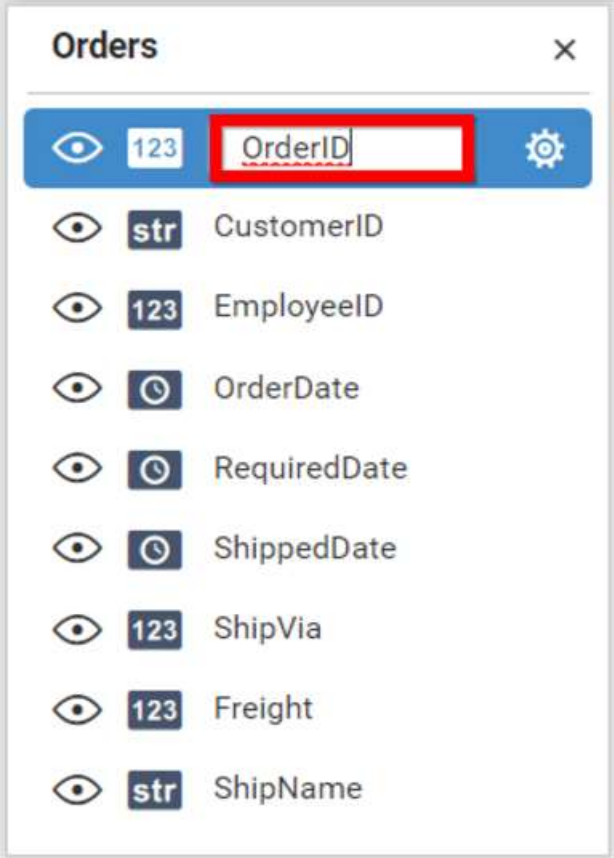
Note: Updating an existing join condition will allow you to edit the column mapping only between those two tables.

Click Save to close the join editor.

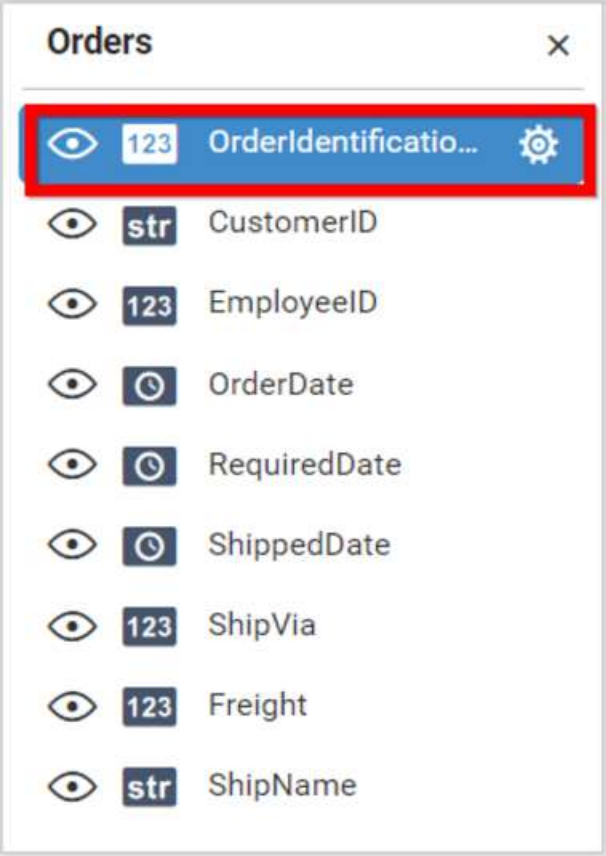
Formatting Columns

Renaming column

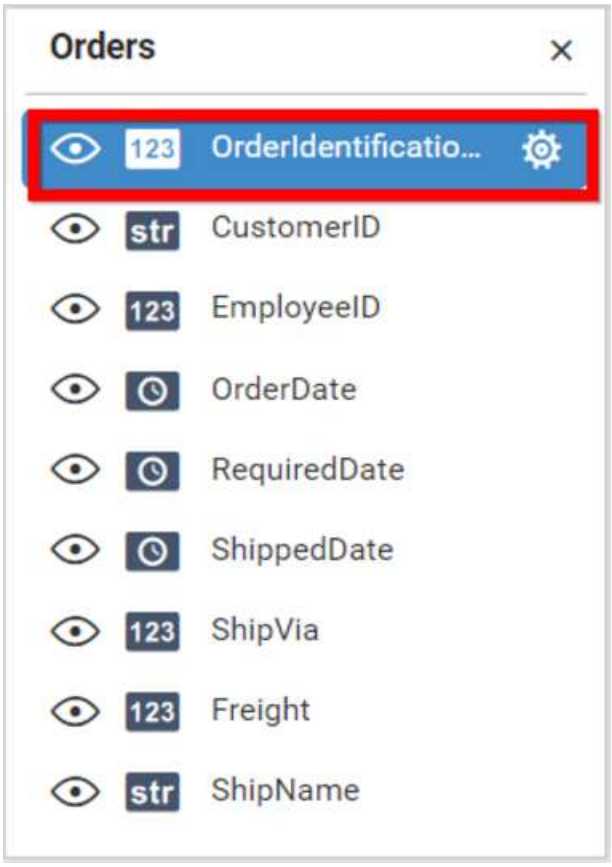
Rename the column, if required, through double clicking the respective column to enable the edit mode and typing the modified name. Press <Enter> key to commit the modification done.



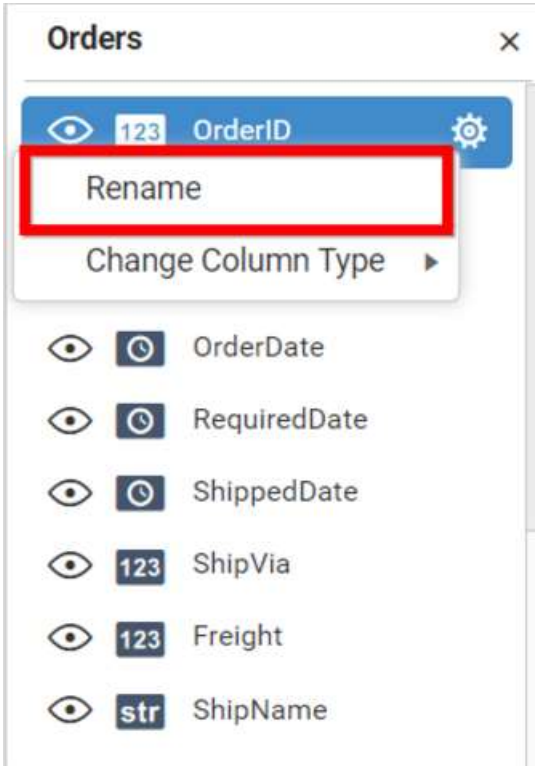
Once modified, the changes will be reflected in the column in the table.



You can also rename the column through the Settings icon which will be displayed like below.

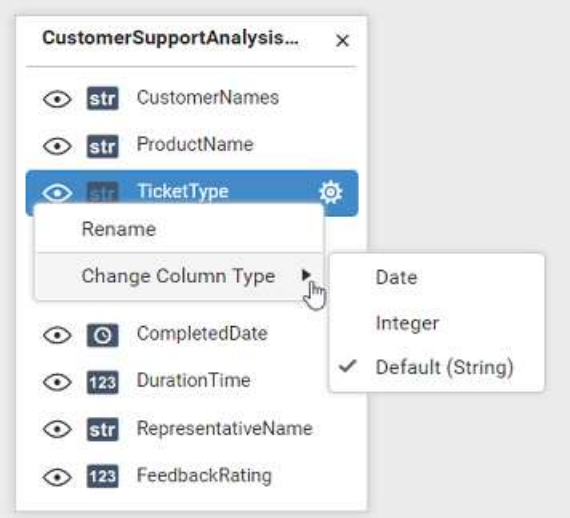


Click this icon to drop down the menu with Rename Column, clicking which the edit mode will be enabled in that column.



Handling column type conversion

Click this icon to drop down the menu with Change Column Type, clicking Change Column Type option allows to convert the column type.



The following table represents the column types and their equivalent convertible types that are supported in Dashboard Designer.

Column Data Type	Equivalent Convertible Types
Date	String
Integer	Date, String
String	Date

Data filters

Configuring Data Filters

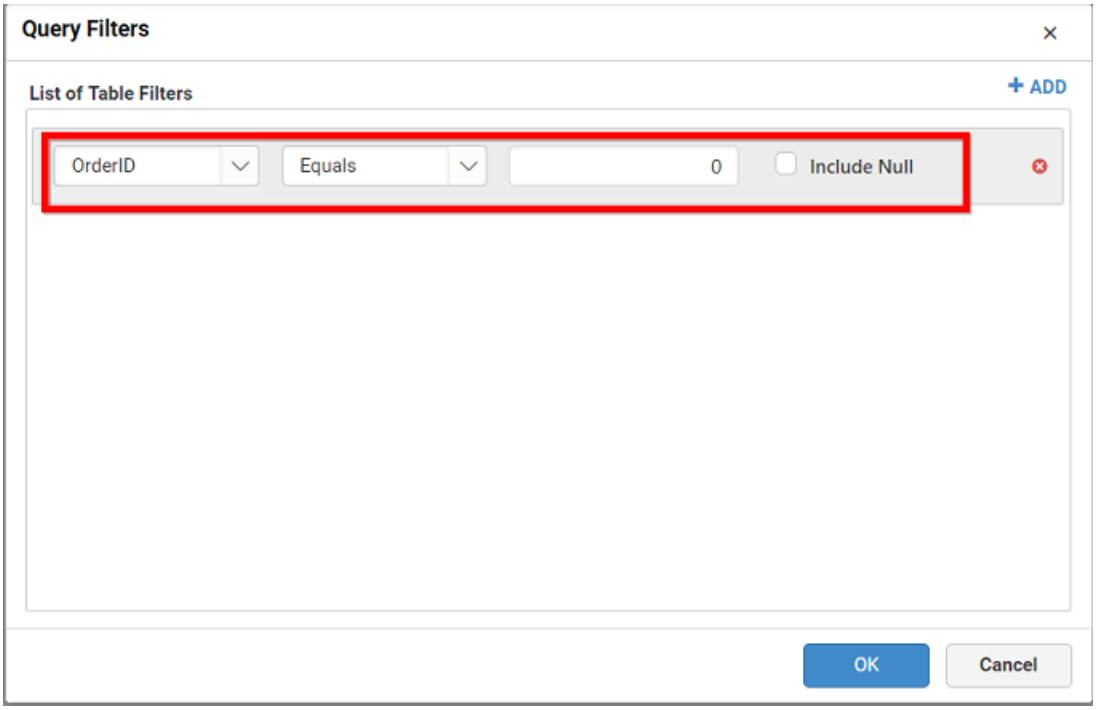
Data Filters can be configured to restrict records visibility based on defined criteria. The configuration can be done by adding and deleting a filter condition.

Adding a filter condition

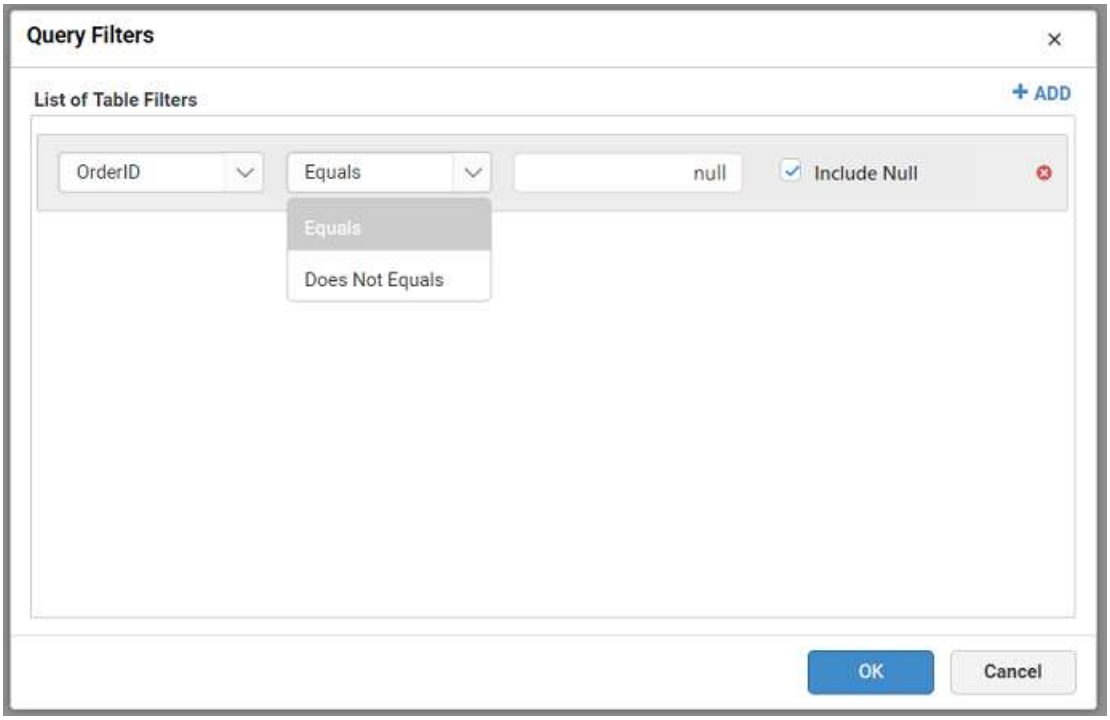
Click Filters option shown in the data design view window. Data filter window opens.

Add a filter condition through clicking the Add button in the Query Filters window.

Now, a filter condition will get added by default like below:

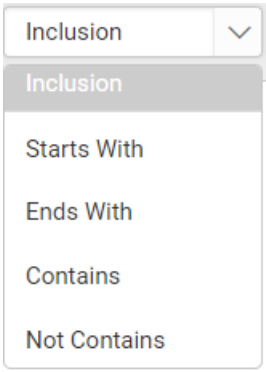


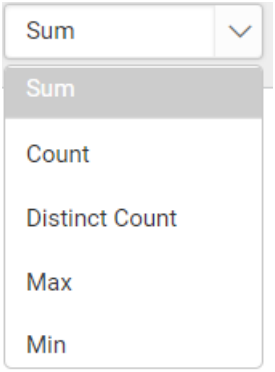
You can filter values with or without Null by checking or unchecking the Include Null option. If you check this option, only Equals and Does Not Equal condition will be shown like below. Based on these conditions, you may filter values with or without Null.

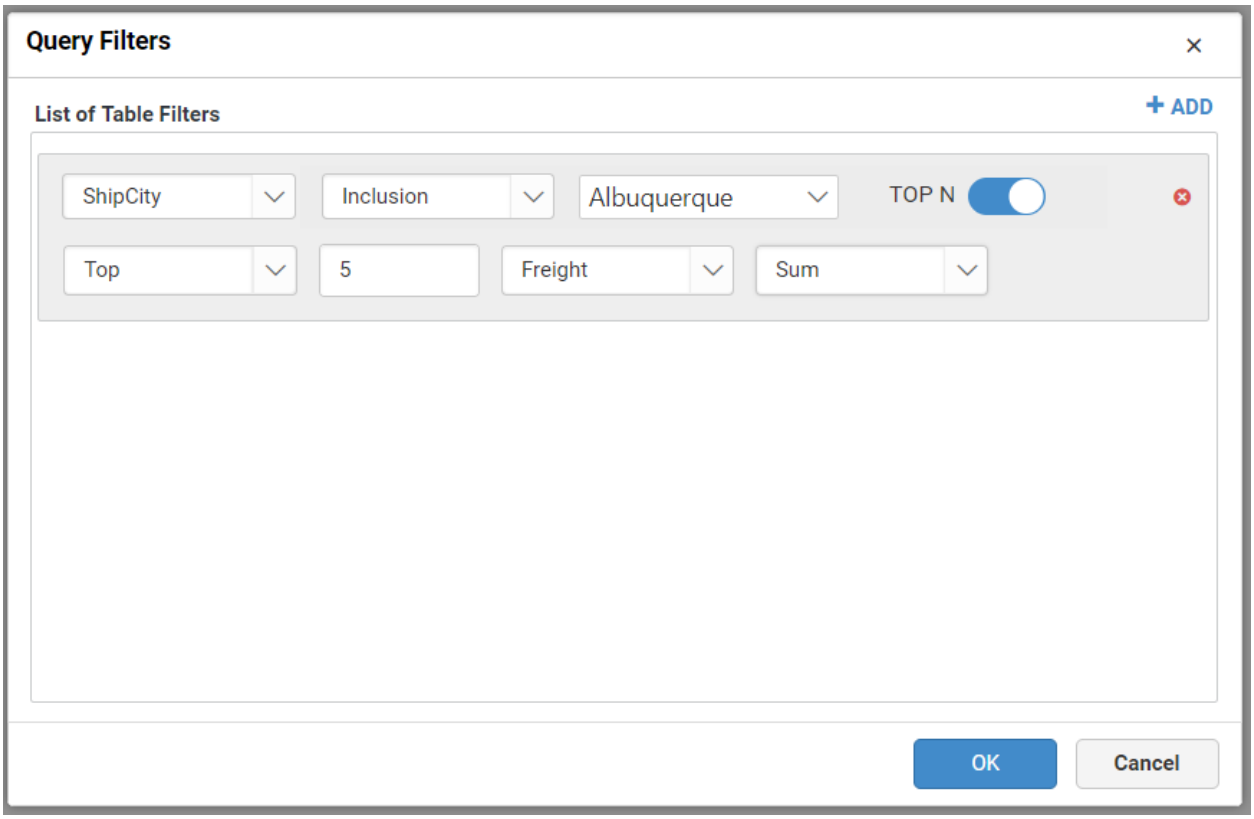


Modify the condition as you require and define criteria. The condition can be defined based on the column. Based on its data type, the parameters to define will differ.

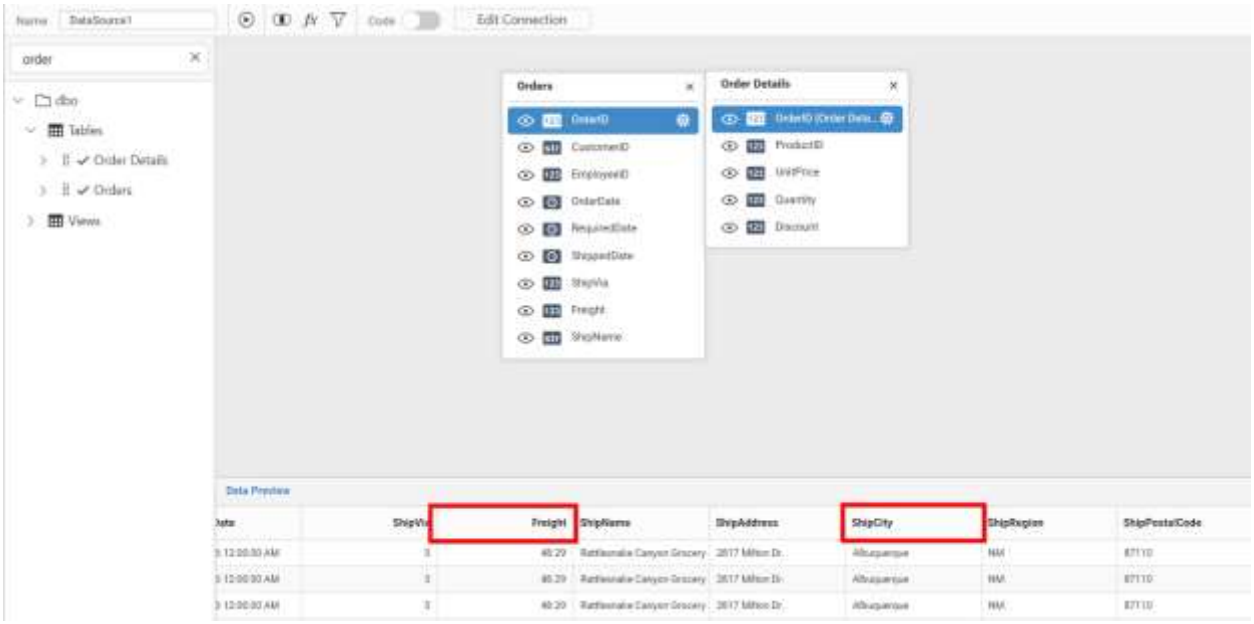
With the date time type or text type column like above, you may get a toggle button TOP ‘N’ at right, to enable the Top N filter to configure the field and the condition by which it has to be applied.



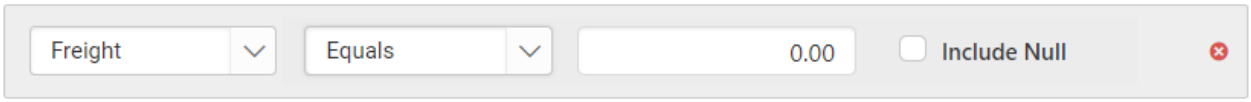


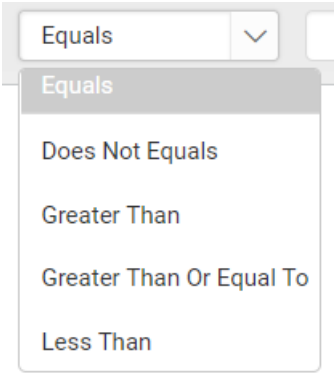


Below example shows top 5 freight data of a city.

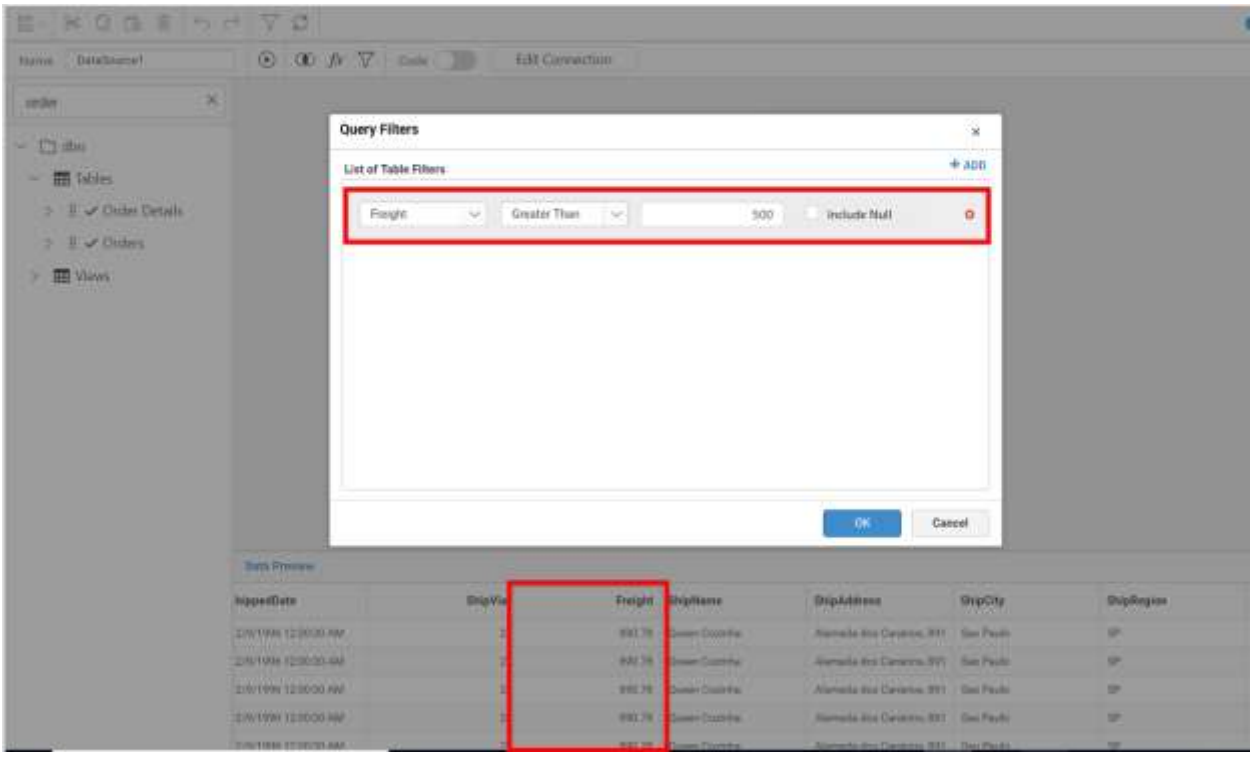


For numeric type column, the parameters will be like below:

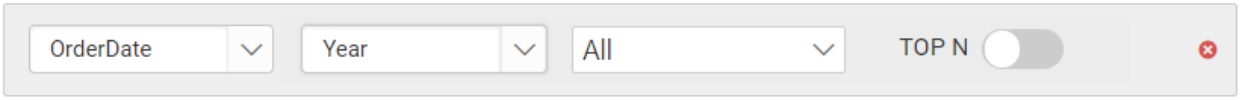


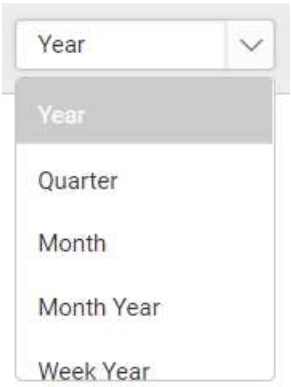


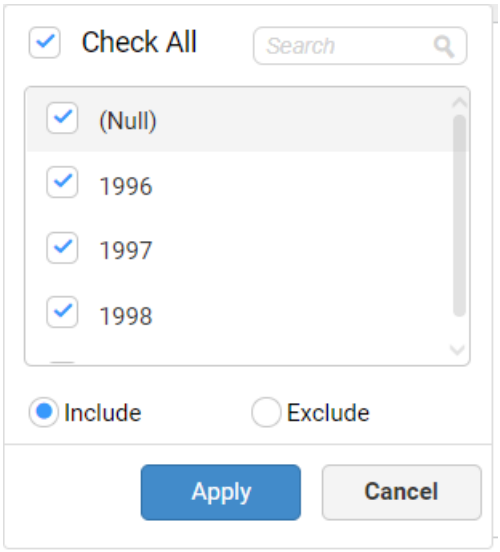
Below example shows data of freight whose values are greater than 500.



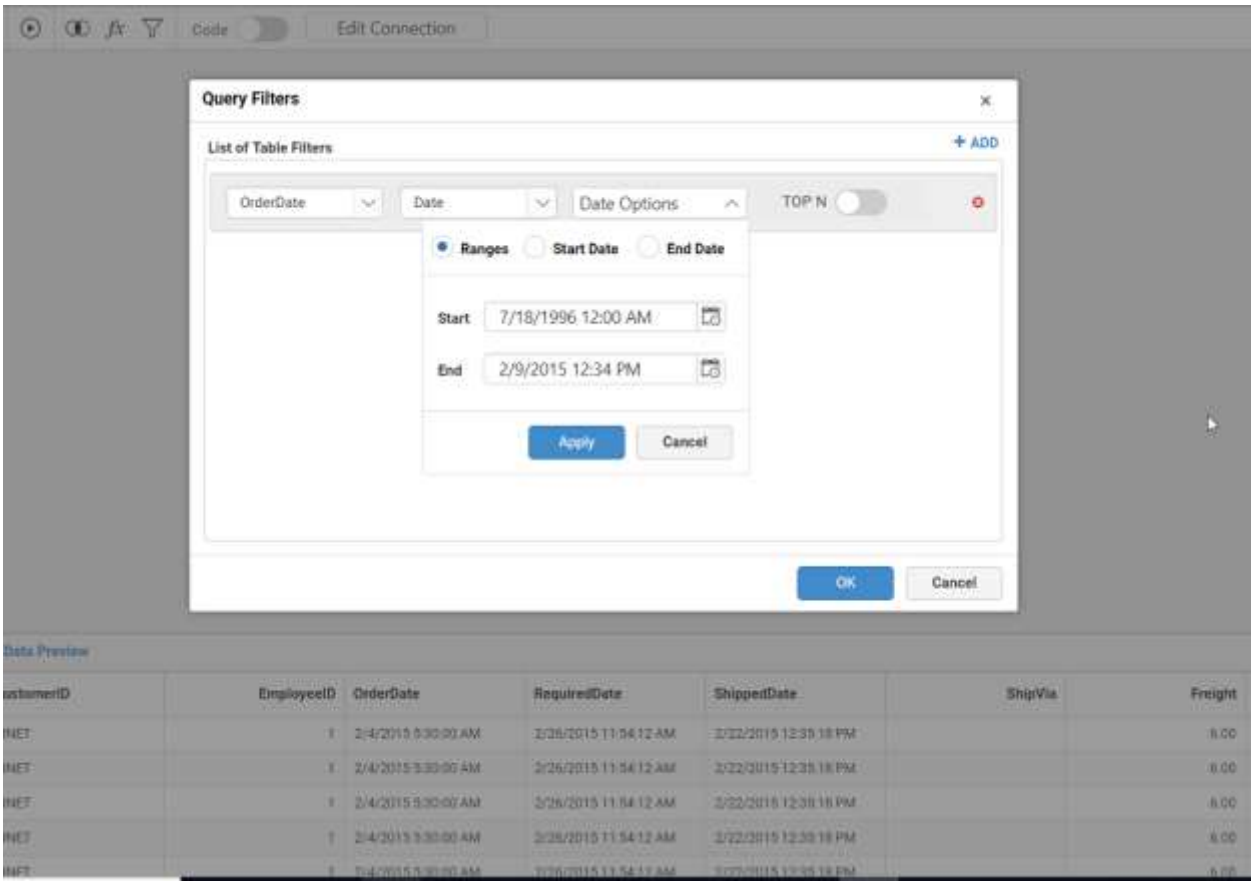
For date time type column, the parameters will be like below:



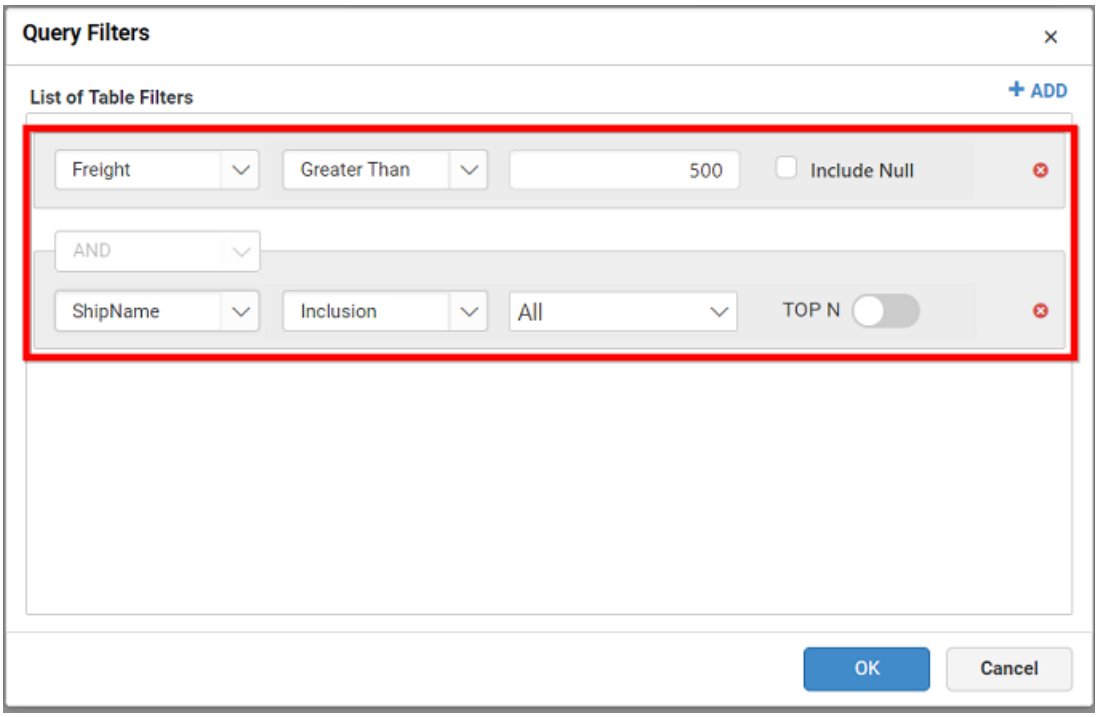




Below example shows data within the applied date range.



Add more than one condition if you prefer, through clicking the Add button.



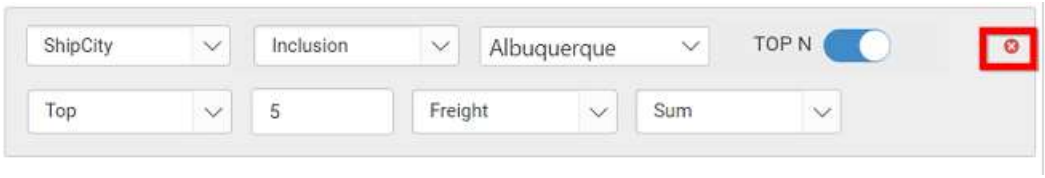
Note: By default, AND operation will be handled in between two conditions. You cannot change the operation between two conditions.

Click OK to save the defined data filter conditions.

Click Close button or the Close icon at top right corner of the window to close the Filters window.

Deleting a filter condition

You can remove a filter condition by clicking the highlighted icon at right of the respective filter condition.



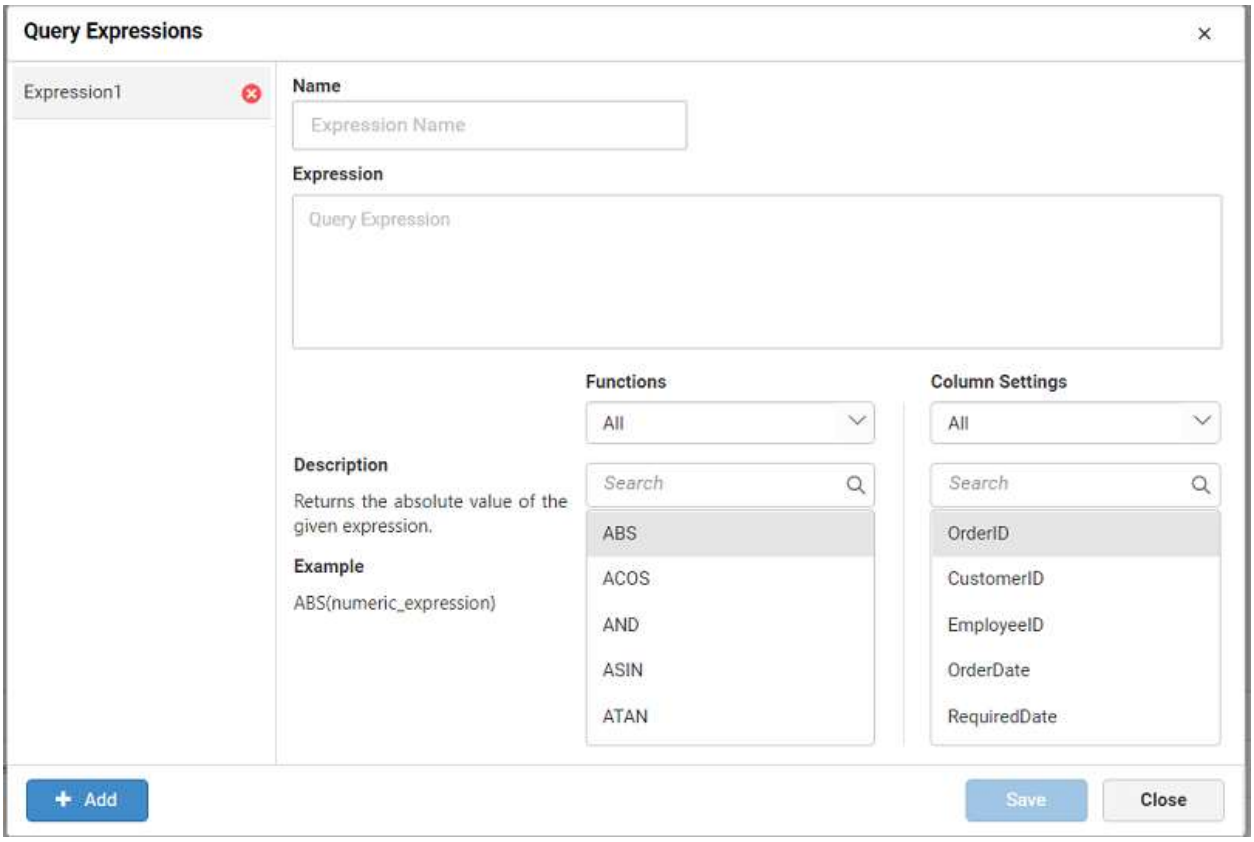
Configuring Expression Columns

An expression column is used to create an expression which is a combination of data columns, operators and built-in functions. This expression column will act as a calculated measure that can be configured to widget like other normal numeric columns as a quantitative measure.

Adding an expression column

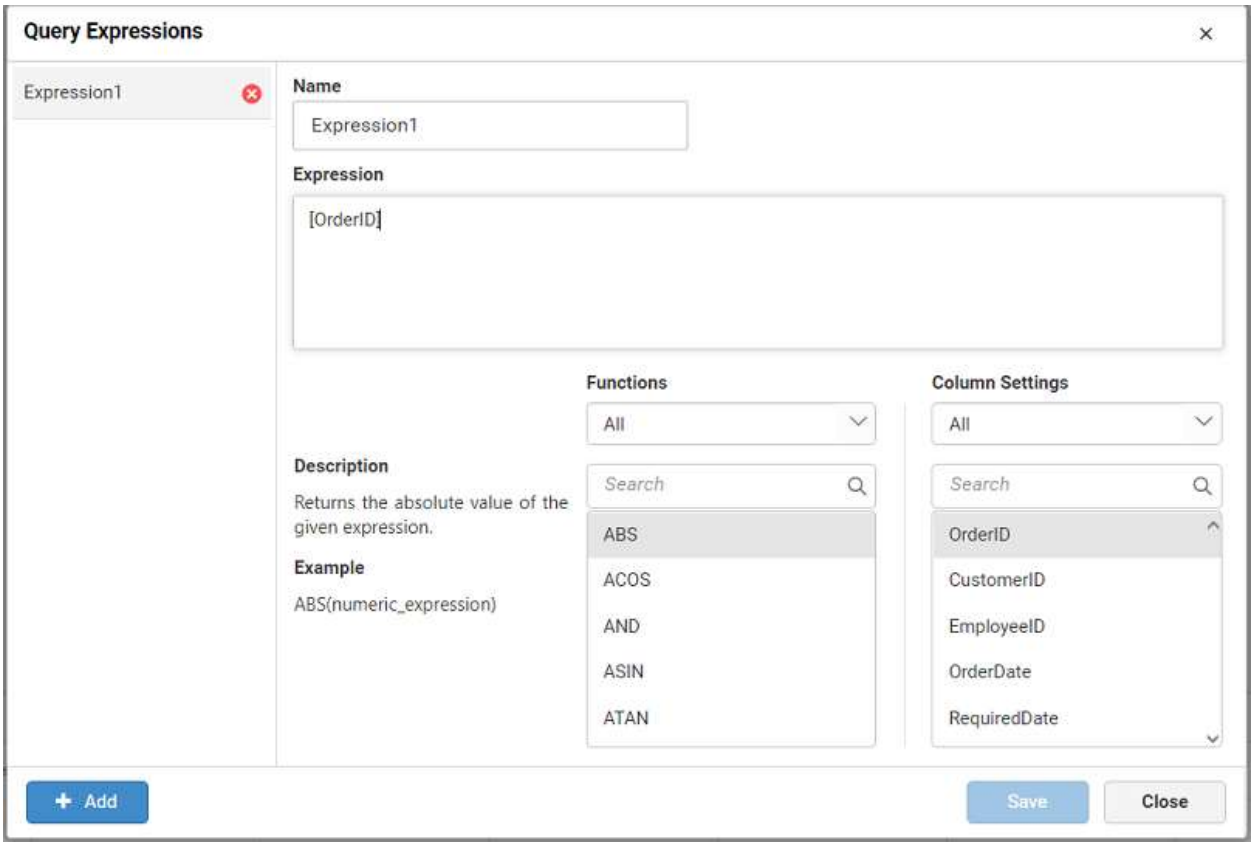
An expression field can be added by clicking Expression menu in the tool bar of the data design view.

Click Add in the Query Expressions window to add a new expression column.



Enter a suitable name for the expression in the Name text area. By default, it will be Expression1.

Enter the expression that you like to define in the Expression text area.



The syntax for defining a simple expression is,

{function name(}[columnname]{operator[columnname]…}

Where, content within curly braces is optional.

Some expressions for reference:

1) YEAR([Order Date]) – To compute year of order date. 2) COUNTD([EmployeeID]) – To compute distinct count of employees. 3) [Freight]+100 – To compute the total with 100 added to Freight.

Following built-in functions are supported in Expression Designer.

Category	Functions	Syntax & Descriptions
Numbers	ABS	Syntax: ABS(numeric_expression) Description: Returns the absolute value of the given expression.
Numbers	ACOS	Syntax: ACOS(numeric_expression) Description: Returns the inverse cosine (also known as arccosine) of the given numeric expression.
Numbers	ASIN	Syntax: ASIN(numeric_expression) Description: Returns the inverse sine (also known as arcsine) of the given numeric expression.
Numbers	ATAN	Syntax: ATAN(numeric_expression) Description: Returns the inverse tangent (also known as arctangent) of the given numeric expression.

Numbers	COS	Syntax: COS(numeric_expression) Description: Returns the cosine of the angle specified in radians, in the given expression.
Numbers	DEGREES	Syntax: DEGREES(numeric_expression) Description: Returns the angle in degrees for the one specified in radians, in the given numeric expression.
Numbers	EXP	Syntax: EXP(numeric_expression) Description: Returns the exponential value of the given expression.
Numbers	LOG	Syntax: LOG(numeric_expression) Description: Returns the logarithm of the given expression to the specified base.
Numbers	PI	Syntax: PI() Description: Returns the constant value of PI.
Numbers	POWER	Syntax: POWER(expression1, expression2) Description: Returns the value of the given expression (expression1) to the specified power (expression2).
Numbers	RADIANS	Syntax: ROUND(numeric_expression) Description: Returns a rounded value.
Numbers	SIGN	Syntax: SIGN(numeric_expression) Description: Returns a value representing the positive (+1), zero (0), or negative (-1) sign of the given numeric expression.
Numbers	SIN	Syntax: SIN(numeric_expression) Description: Returns the sine of the angle specified in radians, in the given expression.
Numbers	SQRT	Syntax: SQRT(numeric_expression) Description: Returns the square root of the given numeric expression.
Numbers	TAN	Syntax: TAN(numeric_expression) Description: Returns the tangent of the given numeric expression.
Aggregation	AVG	Syntax: AVG(numeric_expression) Description: Returns the average of the values in the given expression.
Aggregation	COUNT	Syntax: COUNT(numeric_expression) Description: Returns the number of items in the given expression.
Aggregation	COUNTD	Syntax: COUNTD(numeric_expression) Description: Returns the distinct number of items in the given expression.
Aggregation	MAX	Syntax: MAX(numeric_expression) Description: Returns the maximum value in the given expression.
Aggregation	MIN	Syntax: MIN(numeric_expression) Description: Returns the minimum value in the given expression.
Aggregation	STDEV	Syntax: STDEV(numeric_expression) Description: Returns the standard deviation of values in the given expression.

Aggregation	SUM	Syntax: SUM(numeric_expression) Description: Returns the sum of values in the given expression.
Aggregation	VAR	Syntax: VAR(numeric_expression) Description: Returns the variance of values in the given expression.
Conditional	IF	Syntax: IF(expression, true*part, false*part) Description: Returns either true part or false part, depending on the evaluation of the expression.
Conditional	ISNULL	Syntax: ISNULL(expression) Description: Returns true if the given expression evaluates to null.
Logical	AND	Syntax: (expression1) AND (expression2) Description: Returns true if both the expressions evaluates to true.
Logical	NOT	Syntax: NOT(expression) Description: Returns the reversed logical value of the expression being evaluated.
Logical	OR	Syntax: (expression1) OR (expression2) Description: Returns true if any of the expressions evaluates to true.
Date	DATEADD	Syntax: DATEADD(numeric*expression, date*expression) Description: Adds the number of days to the specified date.
Date	DATESUB	Syntax: DATESUB(numeric*expression, date*expression) Description: Returns the date subtracted from the specified date.
Date	DAY	Syntax: DAY(date_expression) Description: Returns a numeric value representing the day part of the specified date.
Date	DAYDIFF	Syntax: DAYDIFF(start*date*expression, end*date*expression) Description: Returns a numeric value representing the difference between two specified dates.
Date	HOUR	Syntax: HOUR(date_expression) Description: Returns the hour of the given date as an integer.
Date	MINUTE	Syntax: MINUTE(date_expression) Description: Returns a numeric value representing the minute part of the date resulted from specified date expression.
Date	MONTH	Syntax: MONTH(date_expression) Description: Returns a numeric value representing the month part of the date resulted from specified date expression.
Date	NOW	Syntax: NOW() Description: Returns the current date and time.
Date	TODAY	Syntax: TODAY() Description: Returns the current date.
Date	YEAR	Syntax: YEAR(date_expression) Description: Returns a numeric value representing the year part of the date resulting from the specified date expression.

Date	DATENAME	Syntax: DATENAME(date*part, date*expression) Description: Returns a string representing the specified date_part of the given date expression.
Date	DATEPART	Syntax: DATEPART(date*part, date*expression) Description: Returns an integer value representing the specified date_part of the given date expression.
Date	MAX	Syntax: MAX(expression) Description: Returns the maximum value in the given expression.
Date	MIN	Syntax: MIN(expression) Description: Returns the minimum value in the given expression.
String	LEN	Syntax: LEN(string_expression) Description: Returns the number of characters in the given string expression.
String	CHAR	Syntax: CHAR(integer_expression) Description: Converts the given integer ASCII code into a character.
String	CONCAT	Syntax: CONCAT(string*expression1, string*expression2,â€¦, string_expressionN) Description: Returns a string value resulting from the concatenation of two or more string values.
String	CONTAINS	Syntax: CONTAINS(string*expression, substring*expression) Description: Returns true if the given string expression contains the specified substring expression.
String	ENDSWITH	Syntax: ENDSWITH(string*expression substring*expression) Description: Returns true if the given string expression ends with the specified substring expression.
String	LEFT	Syntax: LEFT(string*expression, numeric*expression) Description: Returns the specified number of characters from start of the given string expression.
String	LOWER	Syntax: LOWER(string_expression) Description: Returns a lower case converted string value from a given string expression.
String	LTRIM	Syntax: LTRIM(string_expression) Description: Returns the string value with any leading blanks removed from string expression.
String	MAX	Syntax: MAX(expression) Description: Returns the maximum value in the given expression.
String	MIN	Syntax: MIN(expression) Description: Returns the minimum value in the given expression.
String	RIGHT	Syntax: RIGHT(string*expression, numeric*expression) Description: Returns the specified number of characters from end of the given string expression.

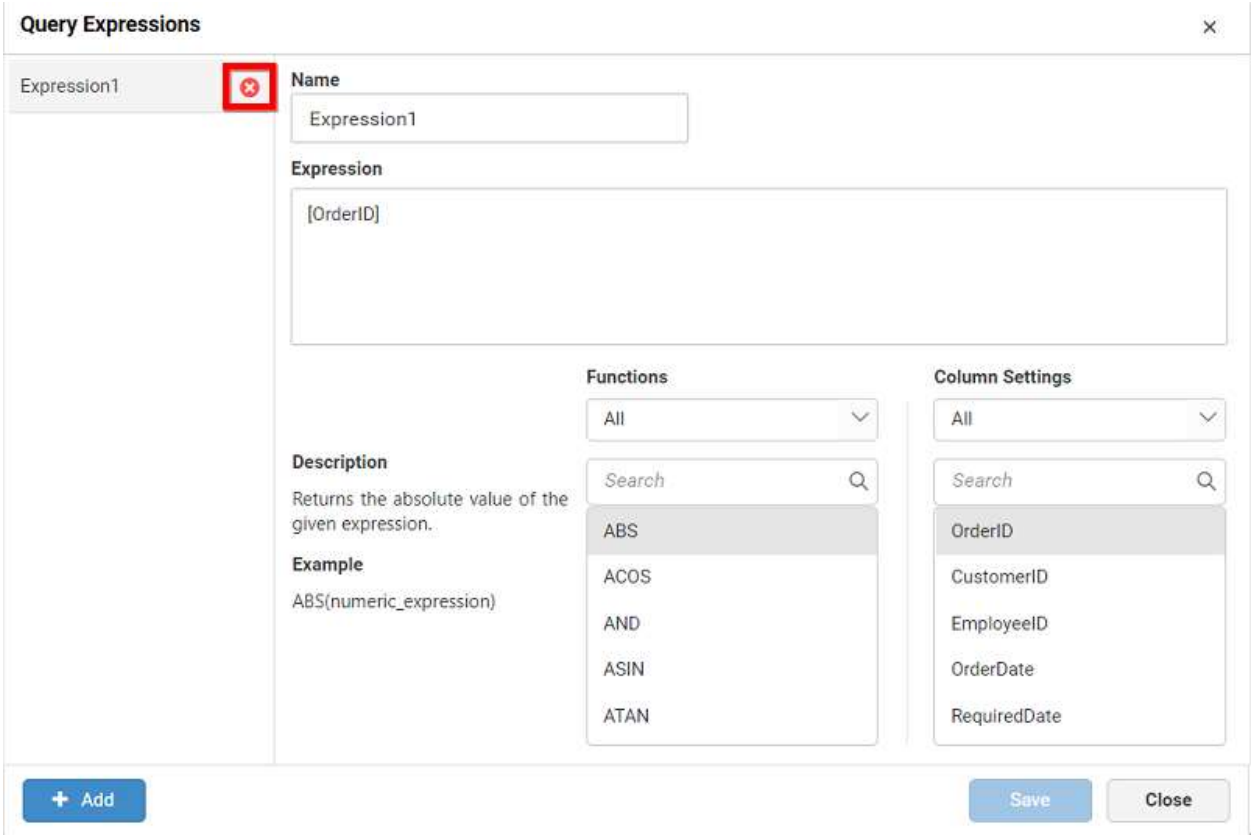
String	RTRIM	Syntax: RTRIM(string_expression) Description: Returns the string value with any trailing blanks removed from string expression.
String	STARTSWITH	Syntax: STARTSWITH(string*expression, substring*expression) Description: Returns true if the given string expression starts with the specified substring expression.
String	SUBSTR	Syntax: SUBSTR(string*expression, starting*index, length*of*the_string) Description: Returns a specific length of string starting from specific index from the given string expression.
String	UPPER	Syntax: UPPER(string_expression) Description: Returns an upper case converted string value from a given string expression.

Once framing an expression, click Save in Query Expression window.

Deleting an expression column

Select an expression column in left pane.

Click Delete icon to remove the selected expression column.



Updating an expression column

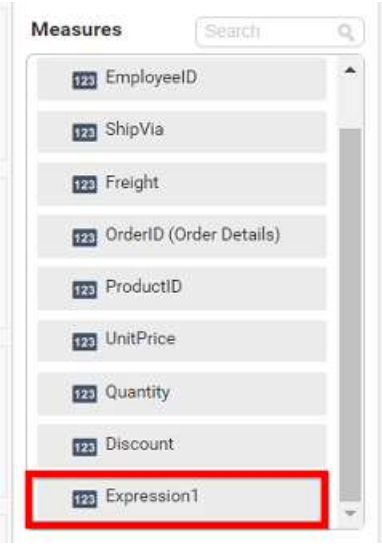
Select an expression column in left pane that you need to update.

Edit the Name and Expression text areas, if required.

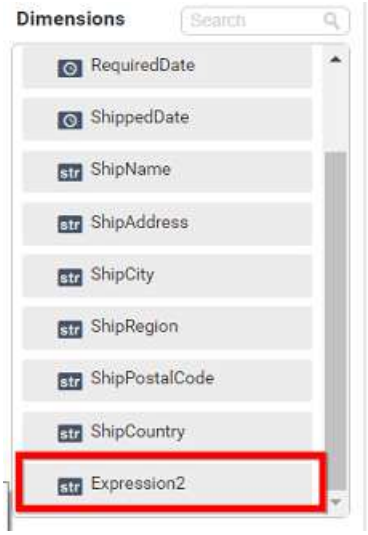
Click Save in Query Expression window to save the modifications handled.

Configuring expression column in widgets

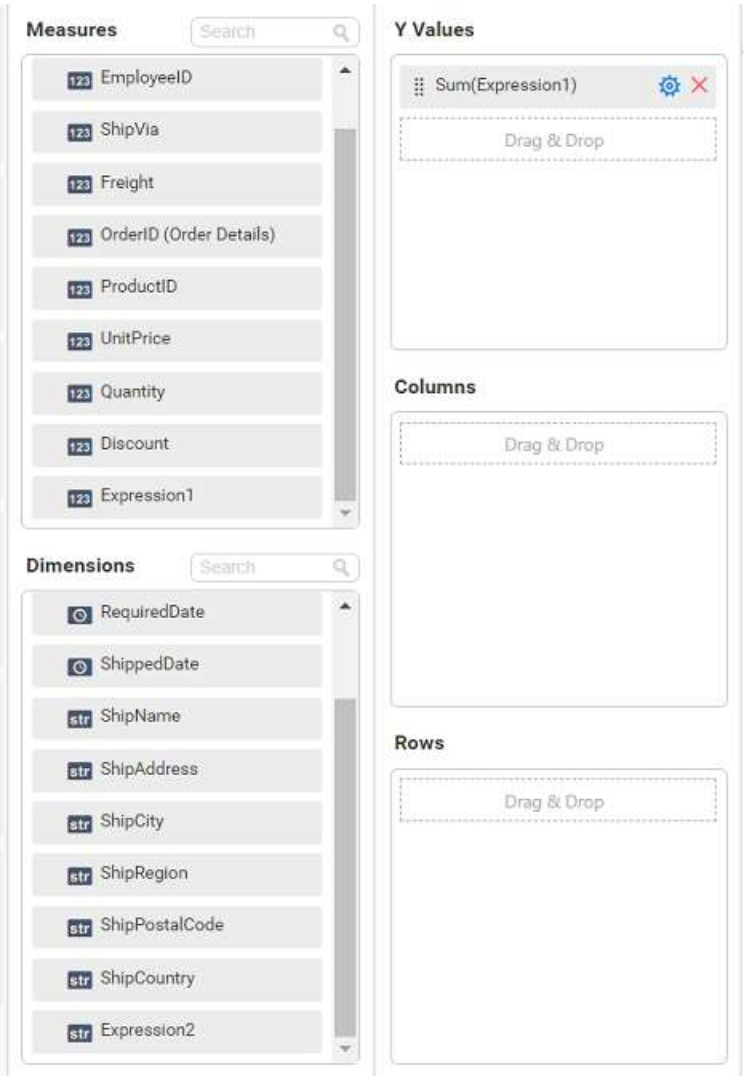
Saved measure expression will be shown in Measure Columns section of ASSIGN DATA tab like below.



Saved dimension expression will be shown in Dimension Columns section of ASSIGN DATA tab like below.



You can also drag and drop expression column into widgets from measure or dimension fields or both.



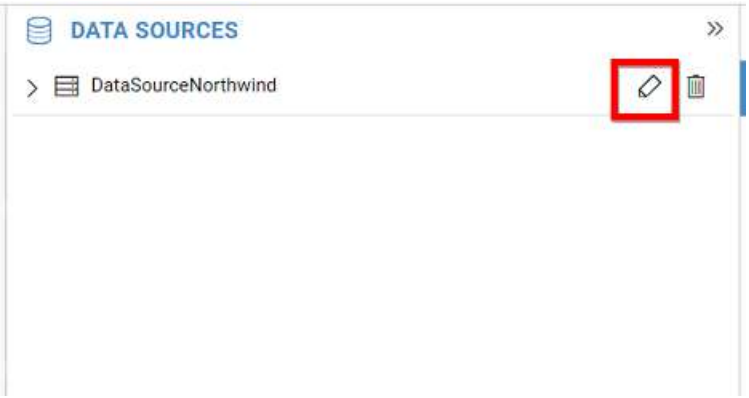
You can also apply filters for expression column which is used in widget. For numeric expressions, you can apply filter just like a measure filter. For string and date expressions, you can apply filter just like a dimension filter.

Editing a Data Source

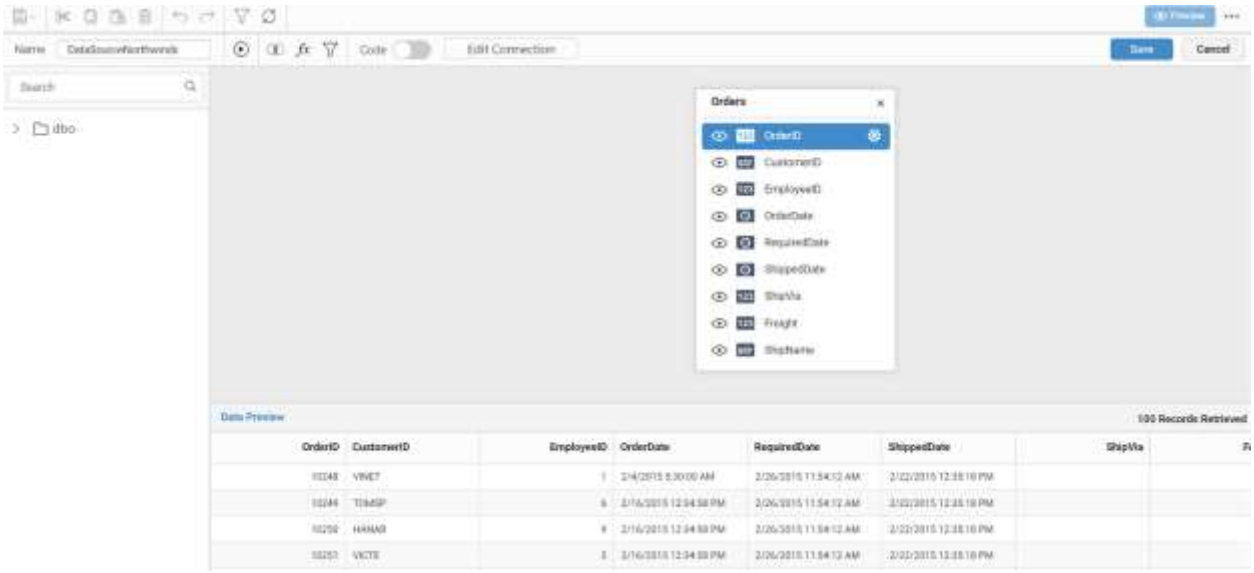
You can edit a data source using the following steps:

1. Click Data Source button in the configuration panel.

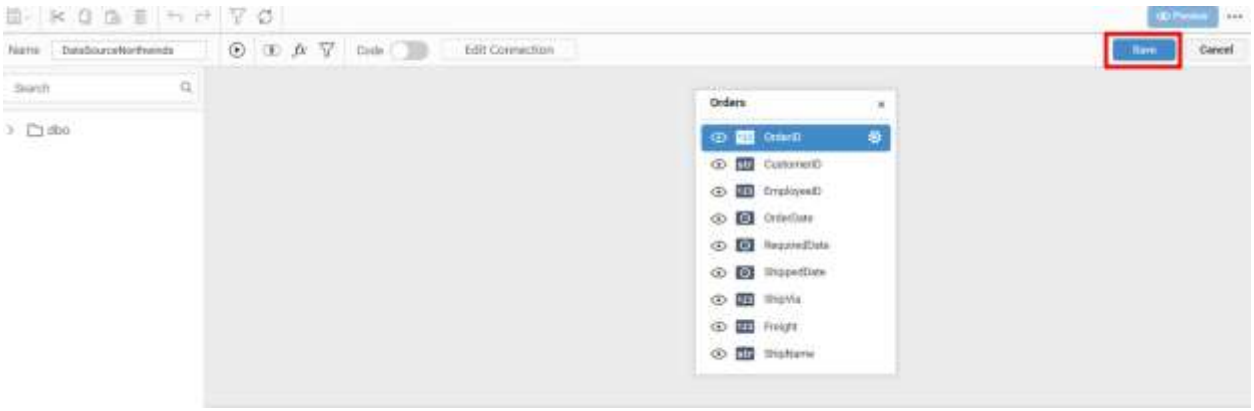
2. Select a data source listed in the data panel that you need to edit.
3. Click the highlighted icon to edit the selected data source.



Now, the respective data source will be opened in the data design view to handle the modification.



4. After modification, click Save in the tools pane in data design view to navigate to the dashboard design view as highlighted in the following image.



[Learn how to rename a data source]

Post your message

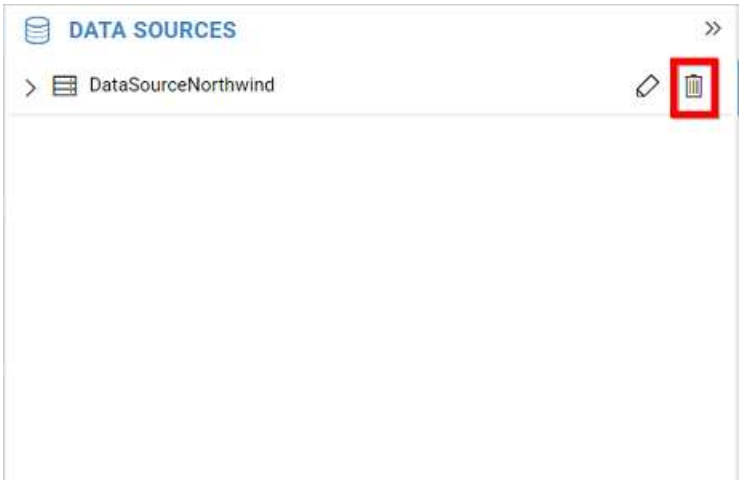
If you are still not able to find the information that you are looking for in the self-help resources mentioned above then please contact us and send your requirements.

Deleting a Data Source

You can delete an existing data source by using the following steps:

1. Click Data Source button in the configuration panel.
2. Select a data source listed in the data panel that you need to delete.

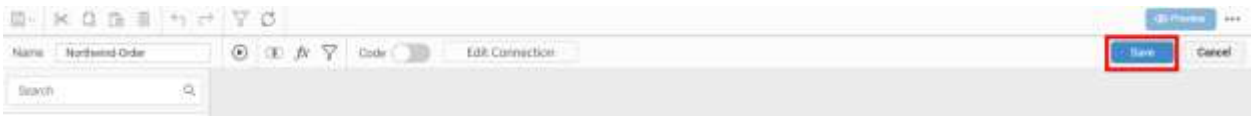
3. Click the highlighted icon to delete the selected data source.



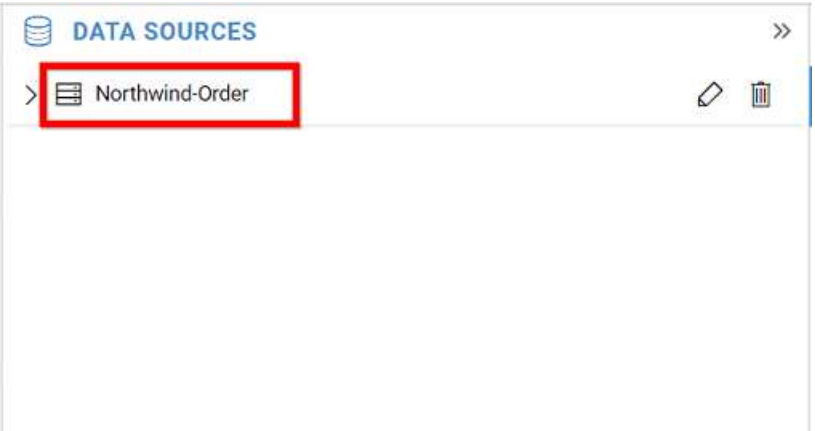
Renaming a Data Source

You can rename a data source using the Name field in the tools pane in data design view as shown (highlighted) in the following image.

Click Save button in the data design view.



After renaming the data source, changes will be reflected in the data panel as shown in the following image.

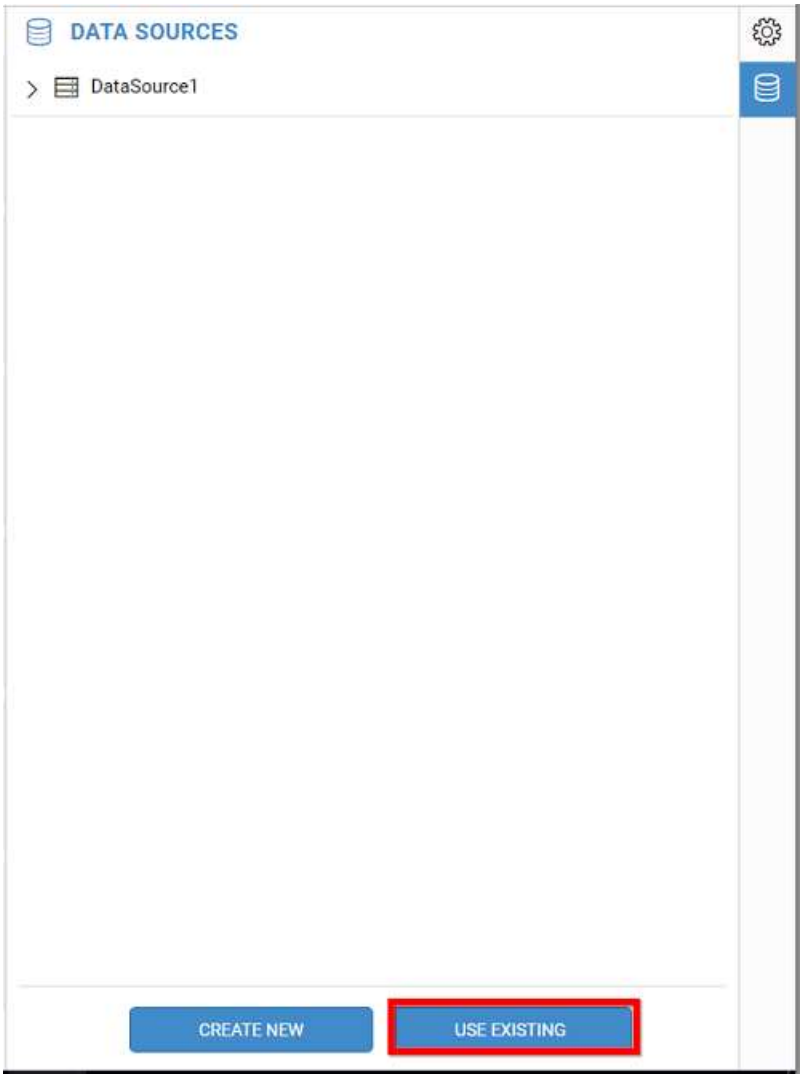


Using an Existing Data Source

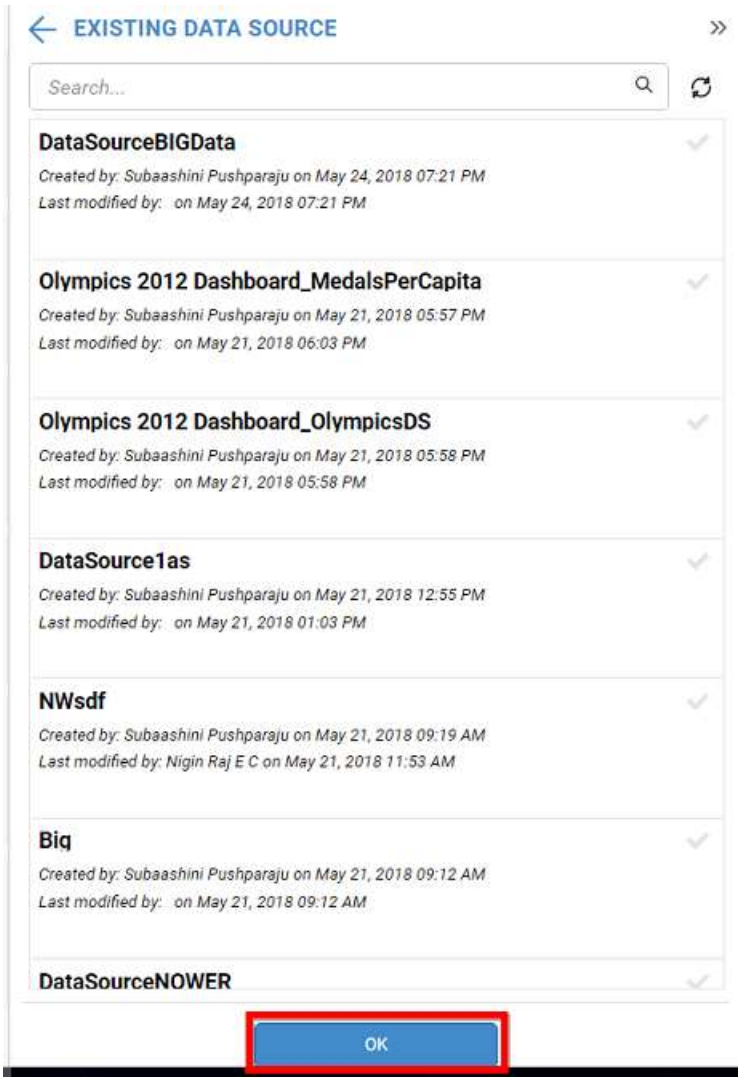
An existing data source can be used in the current dashboard using the Dashboard Designer as follows:

1. Click Data Source button in the configuration panel. The data panel opens.

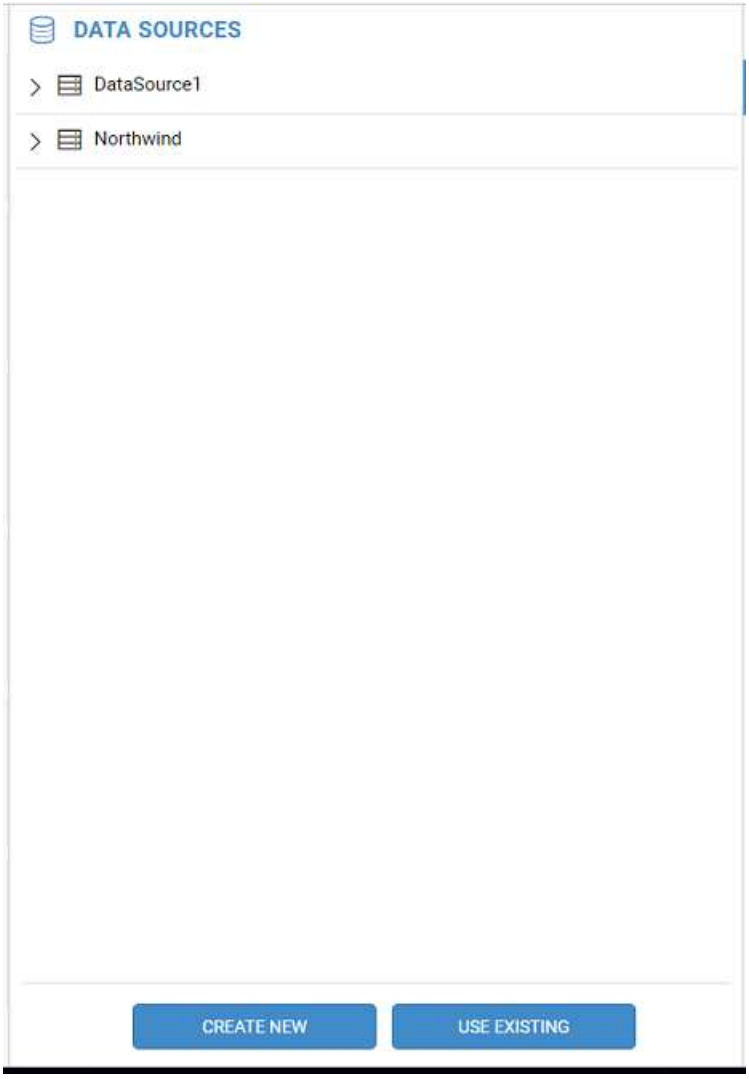
2. Click USE EXISTING in the data panel window. The EXISTING DATA SOURCE window opens.



3. Select the required data source to include in the current dashboard and click OK.



Now, the respective data source(s) will be included in the DATA SOURCES window as follows.



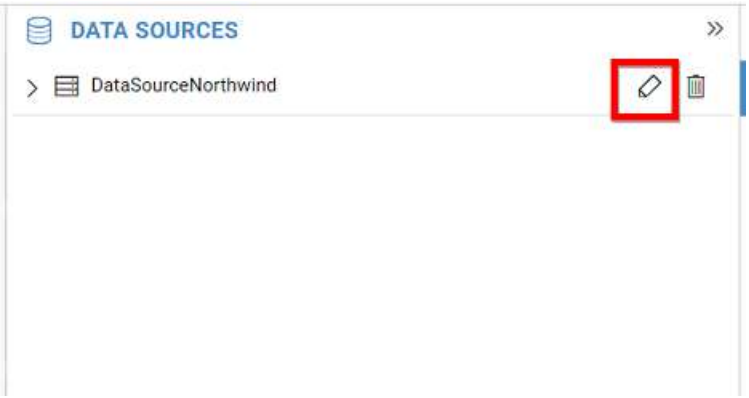
Note: If you include data source with same name, the warning message will be shown as follows.

Editing a Data Connection

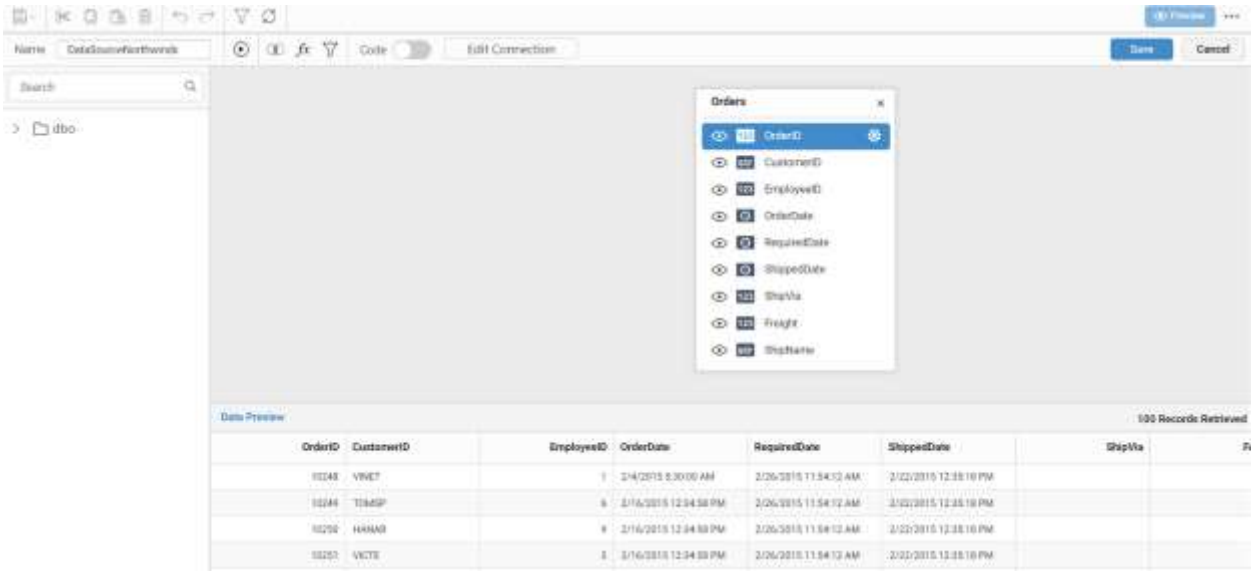
You can edit a data connection using the following steps:

1. Click the Data Source button in the configuration panel.

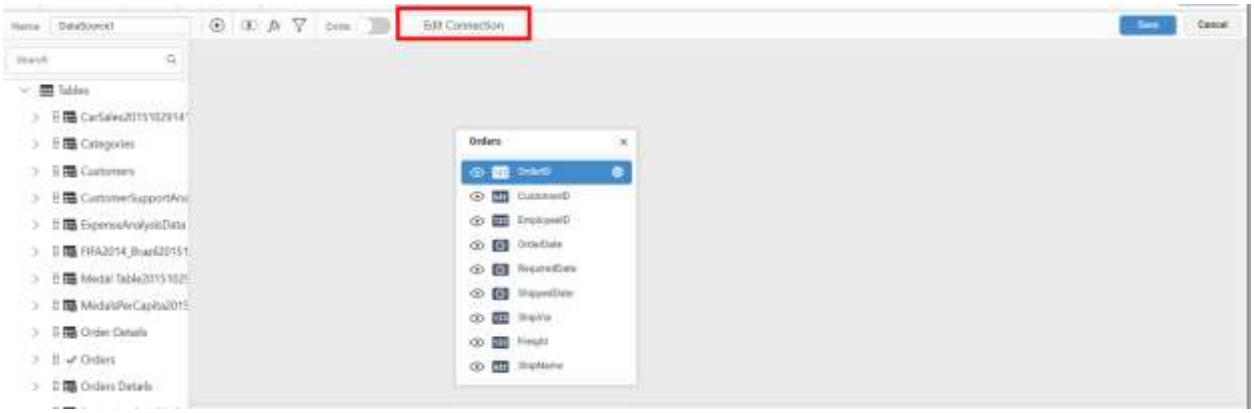
2. Select a data source listed in the data panel that you need to edit.
3. Click the highlighted icon to edit the selected data source connection.



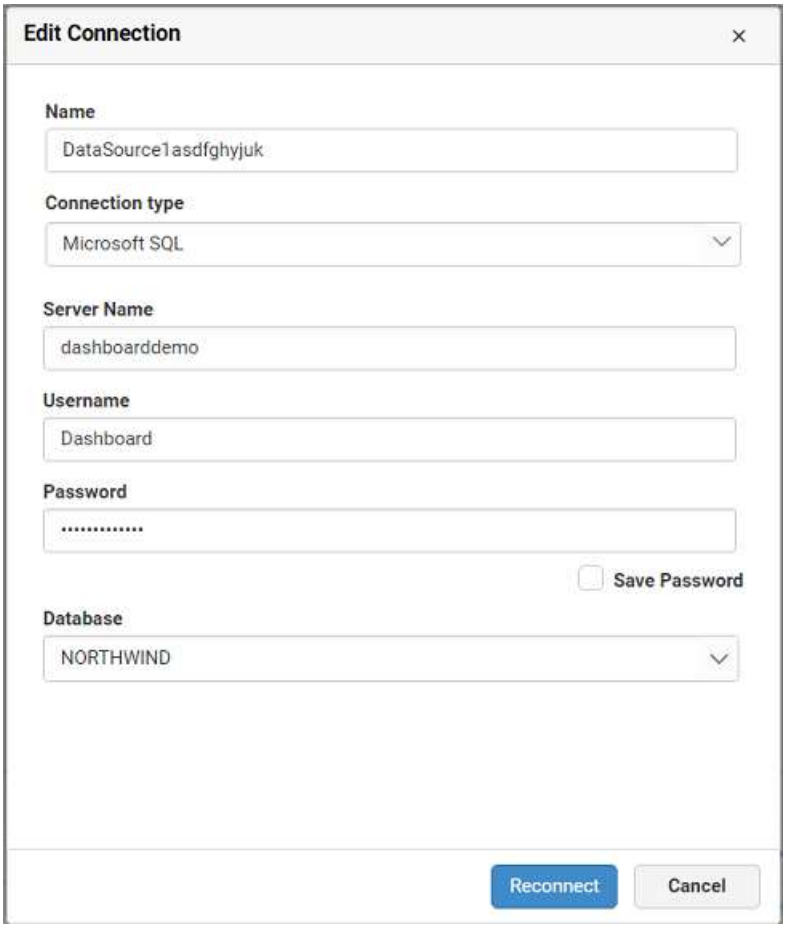
Now, the respective data source will be opened in the data design view to handle the modification.



4. Click the Edit Connection in the data design window toolbar.



Now, the Edit Connection dialog opens.



5. Make the preferred changes and click Reconnect.

Information: Reconnection will persist the dropped table(s), table relationships, data filters, and data configuration to widgets, unless the schema differs from the previous data connection. i.e., the reconnected database should have similar schema such as the previously connected database, which may exist in same or different location. If the reconnected database does not have a column that is available in previous one, reconnection will just ignore that column and its related settings and persist others. Beyond that level, reconnection will drop previous settings entirely.

Note: Cross-data source filter configuration handled in Filters Configuration window can not be maintained on reconnection even the schema is similar.

Configuring Date Parameters

You can configure the URL of the API request with templates containing date queries. The queries will be updated with respective date values.

Syntax{{“{{“}}:today(){{}}}}:It is used for single calendar related method.

Or

Syntax{{“{{“}}:today().adddays(1){{}}}}:It is used when more than one method is added.

For example,a dummy API is used to explain here

Now, this URL will be parsed, and templates will be matched, hence the templates will be replaced with dates accordingly. This helps you to fetch data between the start date and end date.

Functions supported in date parameters

You can configure parameters as numeric values for the following functions: AddMinutes, Addhours, AddDays, AddWeeks, AddMonths, AddYears, AddQuarters, SetDayStart, and SetMonthStart.

All the Add methods should not have parameter as 0, the SetDayStart should have a numerical value between 0 and 6, and the SetMonthStart should have a numerical value between 1 and 12.

String parameters are used in the next set of functions which are start, end, format, and SetTimeZone. Both start and end functions support four string parameters which are week, month, quarter, and year. The format function is used to change the format of date and time, and the parameter is matched with the date and time format supported in C#. A support for epoch time is also made available in the format function. The SetTimeZone is used to change the time zone of the date and the parameter is matched with TimeZoneInfo IDs present in C#.

Today function does not hold any parameters.

Here for example concern we are using today()as 11/16/2018 12:17

Function Name	Type(s) Used	Description	Example(s)	Result
Today	No Parameter	Sets date and time to current date and time.	{{"{{"}}:today(){{}}}}	11/16/2018 12:17
AddMinutes	Numerical	Updates the date and time by changing the number of minutes.	{{"{{"}}:today().addminutes(10){{}}}}	11/16/2018 12:27

AddHours	Numerical	Updates the date and time by changing the number of hours.	{{"{{"}}:today().addminutes.addhours(2){{}}}}	11/16/2018 14:29
AddDays	Numerical	Updates the date and time by changing the number of days.	{{"{{"}}:today().adddays(2){{}}}}	11/18/2018 12:17
AddWeeks	Numerical	Updates the date time by adding a date with the numerical parameter considered as 7 days.	{{"{{"}}:today().addweeks(1){{}}}}	11/23/2018 12:17
AddMonths	Numerical	Updates date and time by adding months with numerical parameter.	{{"{{"}}:today().addmonths(2){{}}}}	1/16/2018 12:17
AddYears	Numerical	Updates date and time by adding years with the numerical parameter.	{{"{{"}}:today().addyears(3){{}}}}	11/16/2020 12:17
AddQuarters	Numerical	Updates date and time by adding months with a	{{"{{"}}:today().adddays(10).addquarters(2){{}}}}	5/26/2019 12:17

		numerical parameter, where parameter value 1 means 3 months.		
Start	String	Sets the date and time values to the start of the given string parameter.	{{"{{"}}:today.addweeks(2).start(week){{}}}}	11/25/2018 00:00
End	String	Sets the date and time value to the end of the given string parameter.	{{"{{"}}:today().addmonths(4).end(year{{}})}}	12/31/2019 00:00
Format	String	Formats the date and time to the correct date format entered as string parameter.	{{"{{"}}:today().start(week).format(MM/dd/yyyy){{}}}}	11/11/2018
SetTimeZone	String	Changes the time zone to the time zone entered as string parameter.	{{"{{"}}:today().settimezone(New Zealand Time Zone){{}}}}	11/16/2018 19:47
SetDayStart	Numerical	Updates the date to the day of the week based on the entered numerical parameter.	{{"{{"}}:today().Setdaystart(1){{}}}}	11/12/2018 12:17

SetMonthStart	Numerical	Updates the date to change the month based on the entered numerical parameter	{{"{{"}}:today().SetMonthStart(10){{}}}}	10/16/2018 12:17

Note: Each template should mandatorily start with today function.

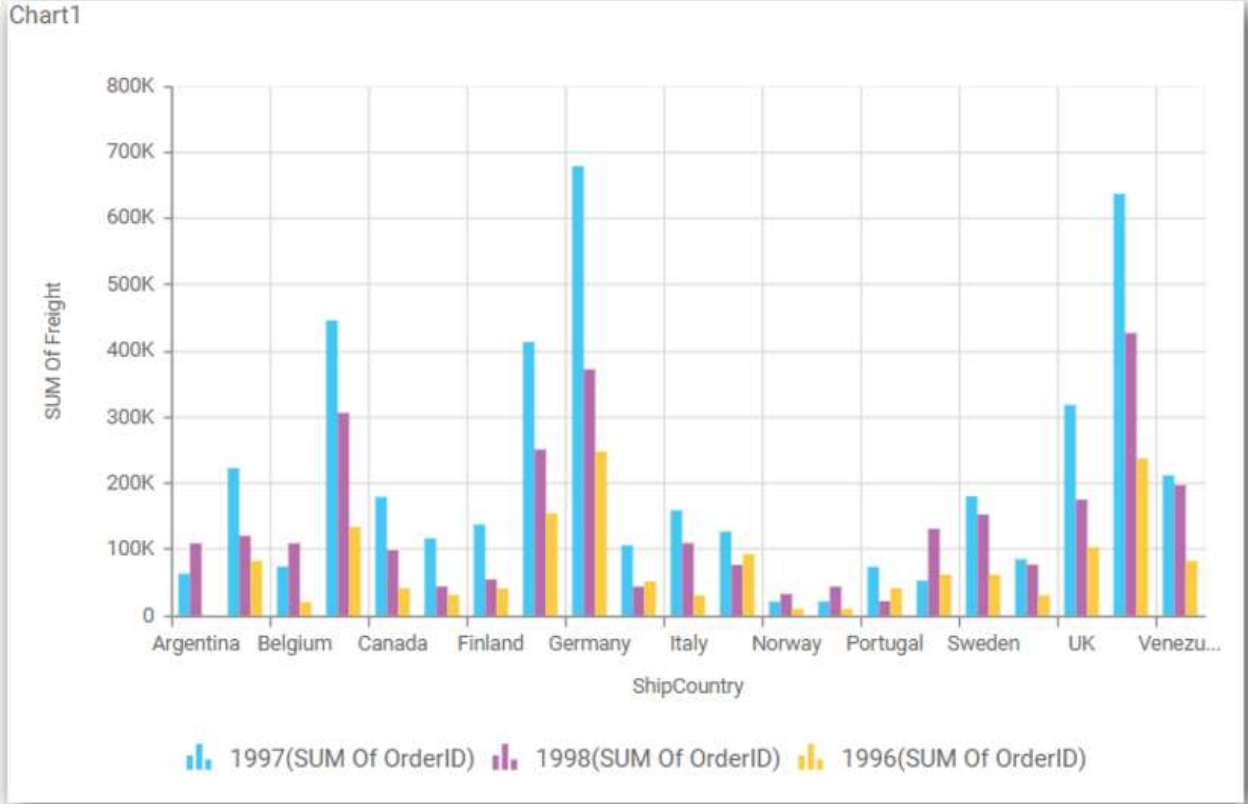
For a template, if the format function is used, it should be the last function call.

Visualize Data

Configure Data Visualizations

Column Chart

Column Chart allows you to compare values for a set of unordered items across categories through vertical bars ordered horizontally.

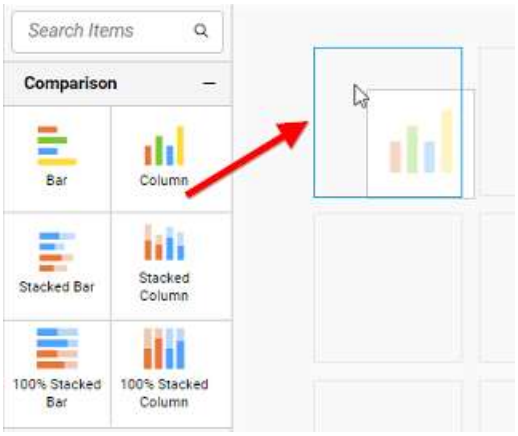


How to configure table data to column chart?

Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

Follow the steps to configure data to column chart

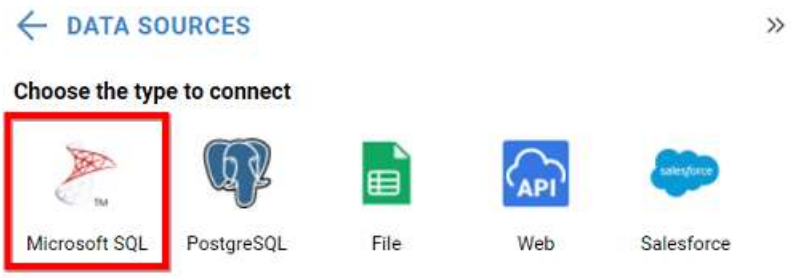
Drag and drop the column chart widget into canvas and resize it to your required size.



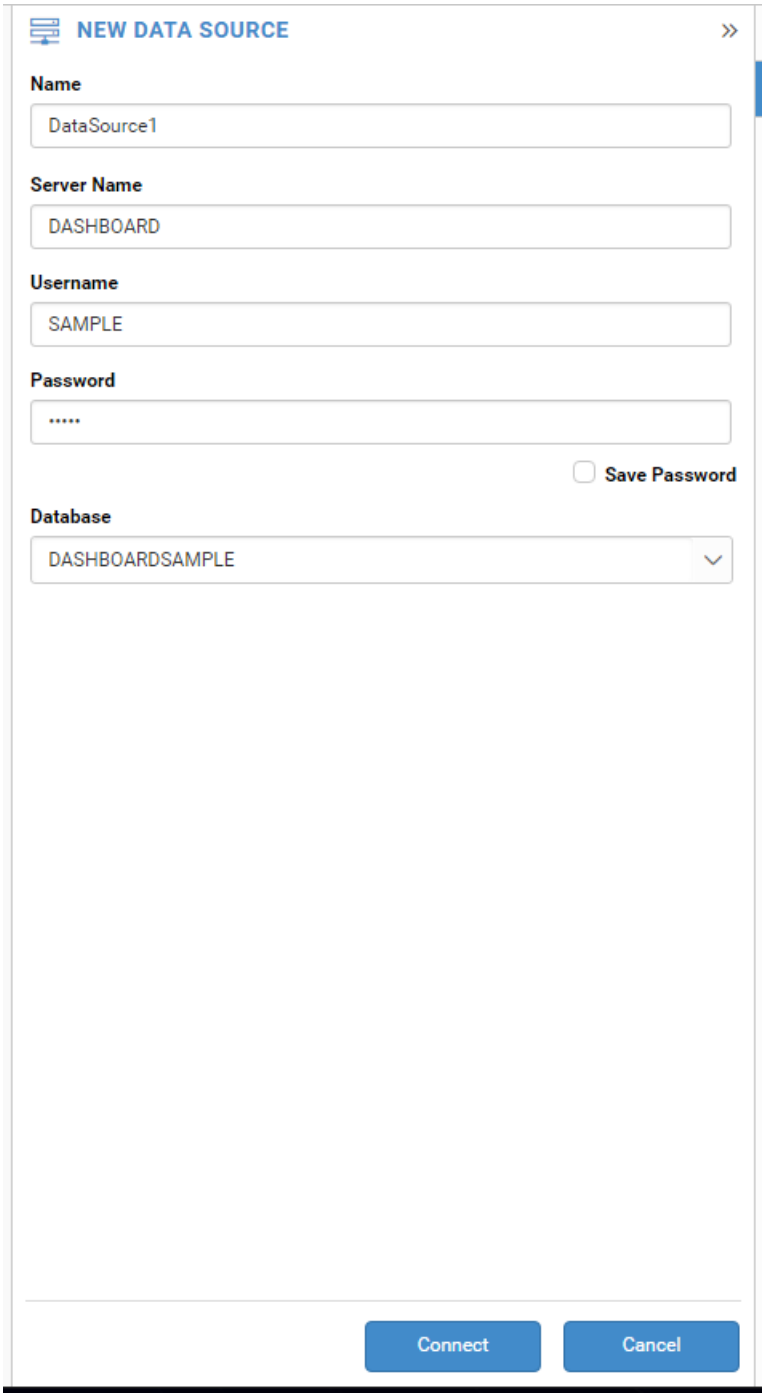
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a New Connection from connection type panel.

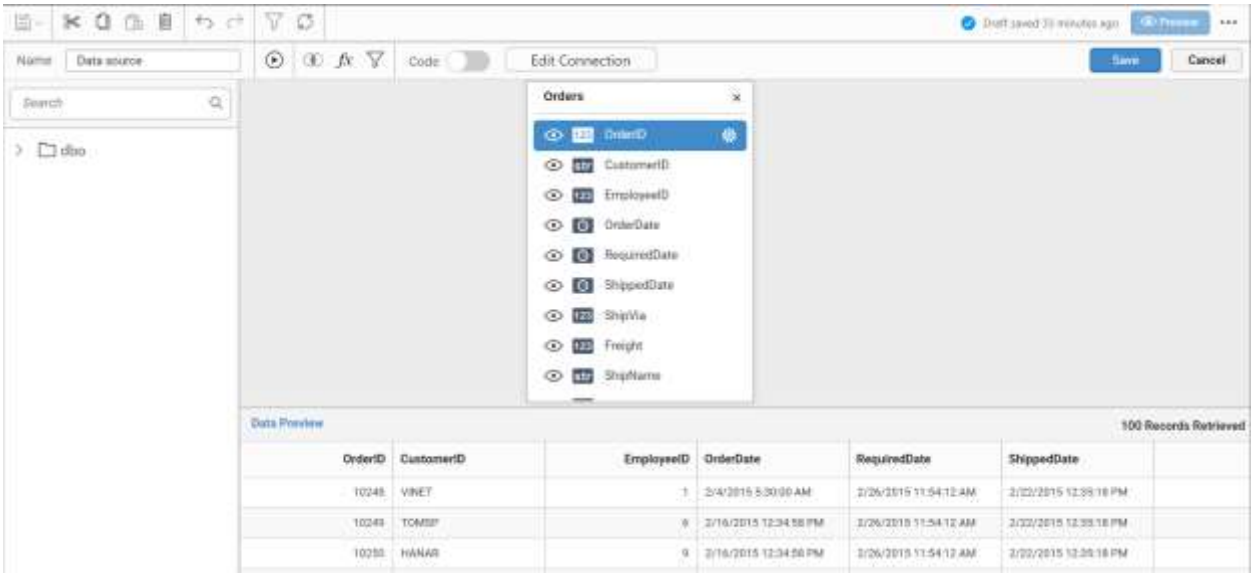
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



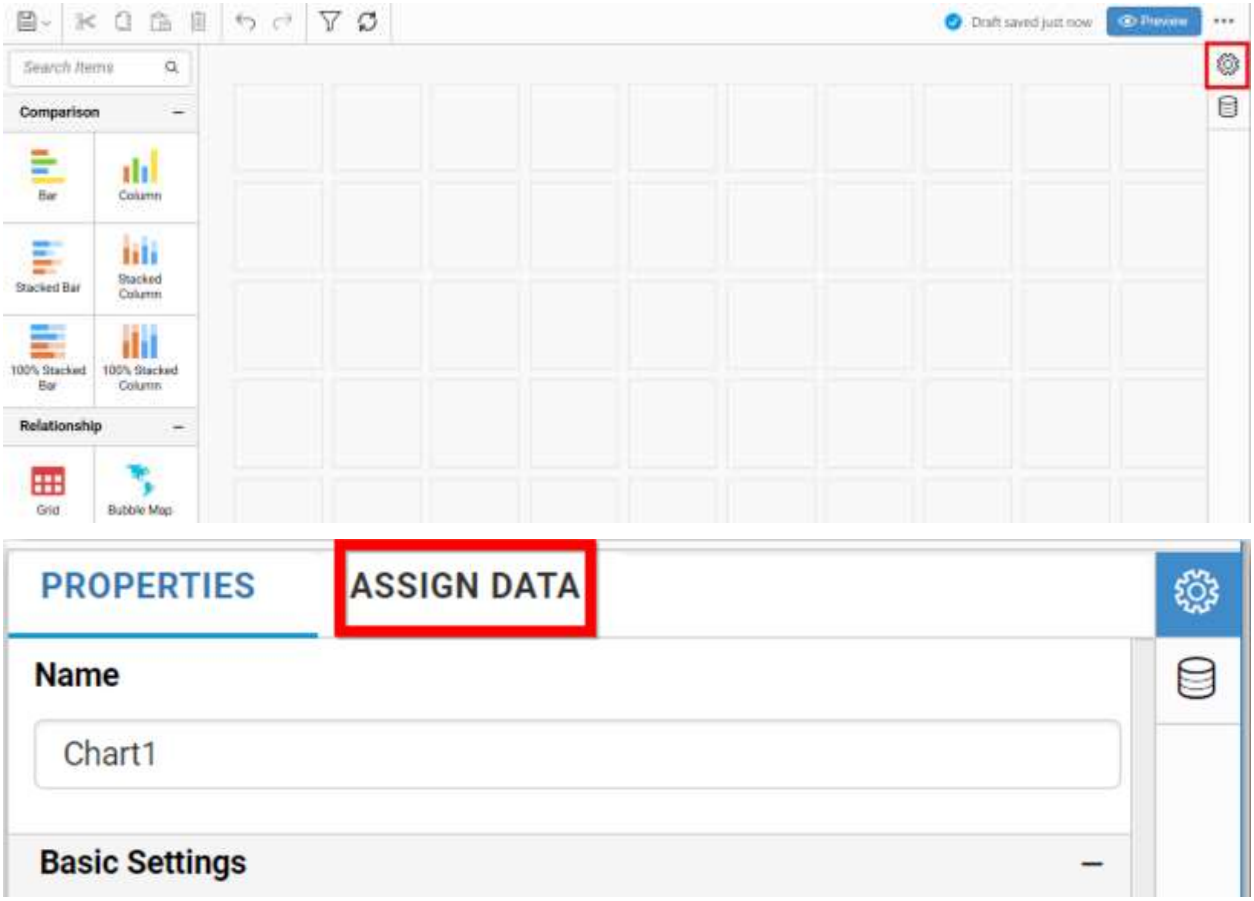
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



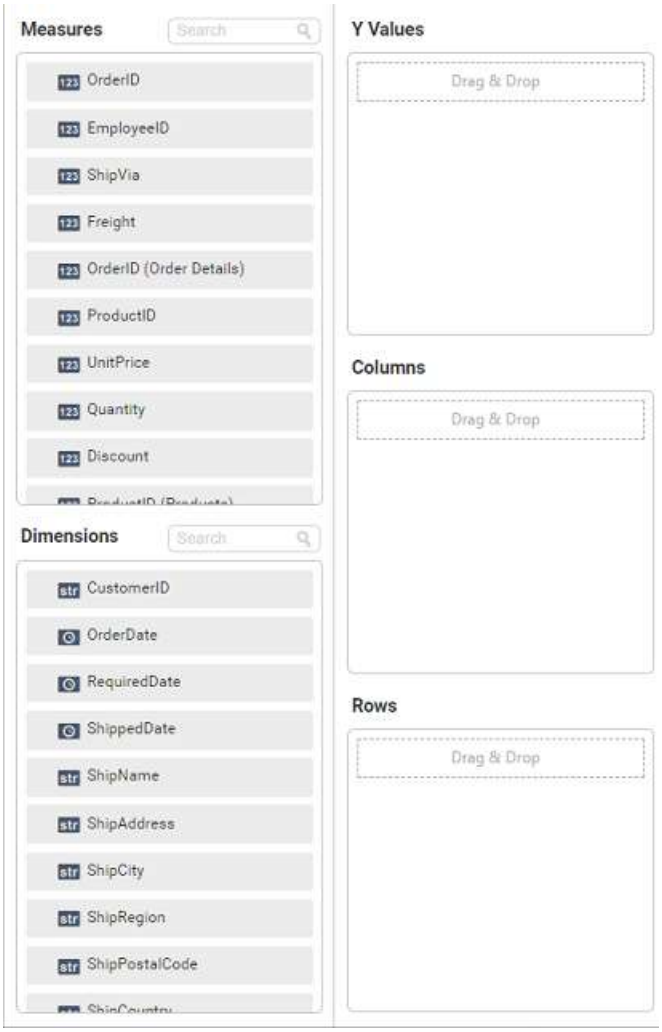
Drag your preferred table or view from the left pane from data design view, click Save button.



Click Properties button in Configuration panel, Property pane opens. Now, Switch to ASSIGN DATA tab.



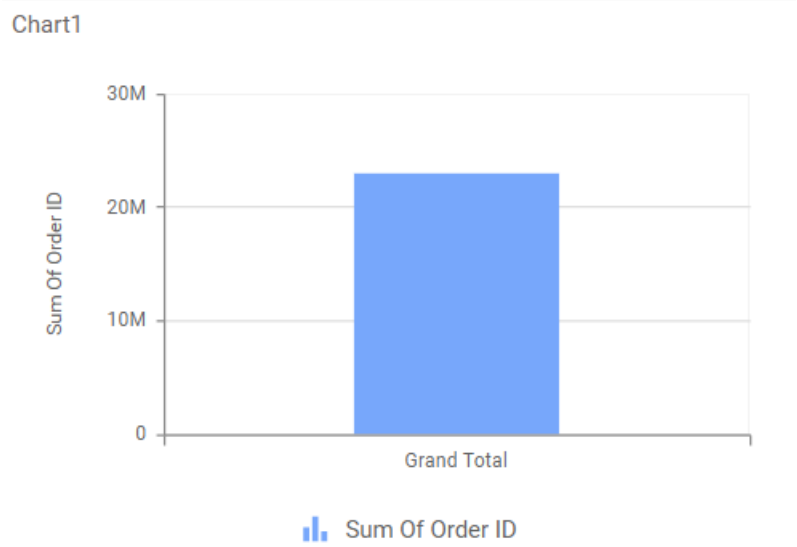
The data tab will be opened with available measures and dimensions from the connected data source



Adding Y Values

You can add more than one Measures into Y Values field by drag and drop the required measure.

Now the chart will be rendered like this

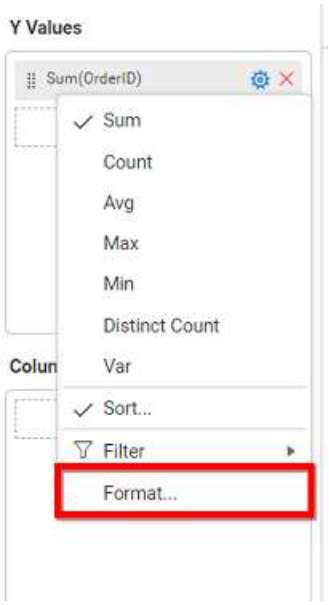


Click the Settings option to change required summary type from the available summary types shown in Settings.

You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

You can filter the data to be displayed in chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

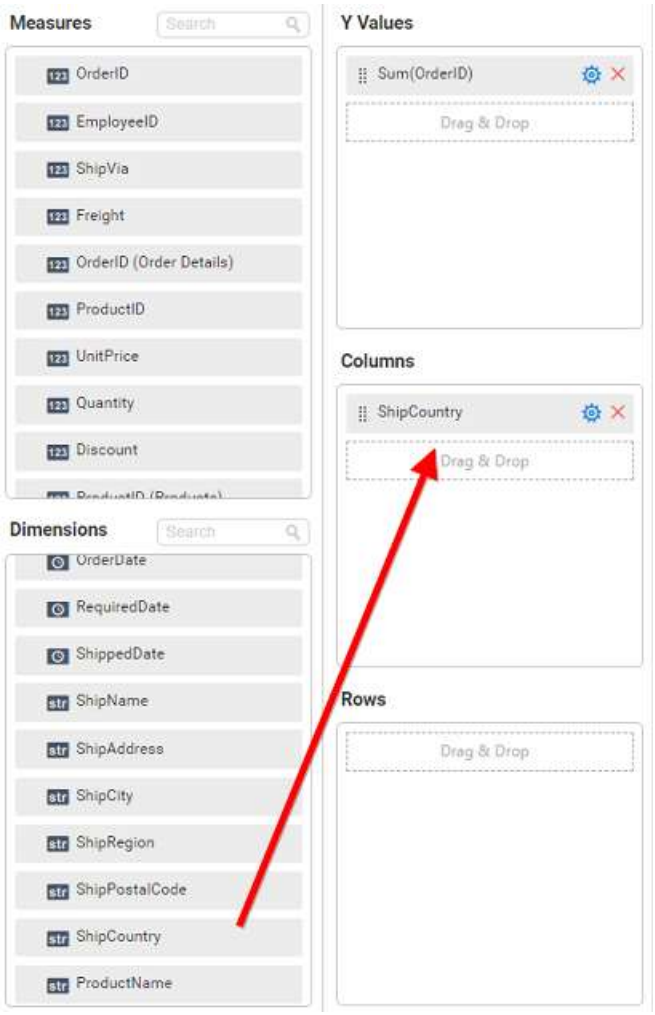


To remove the added value fields click the highlighted button.

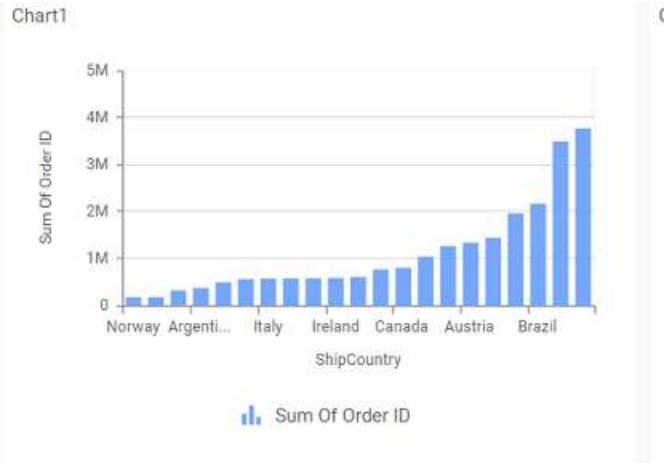
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



Column chart will be rendered like this



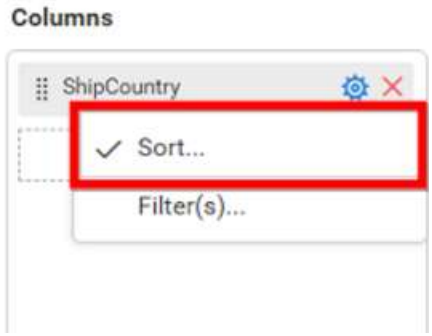
Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

Note: If you click No, single value will be added to the Columns field.

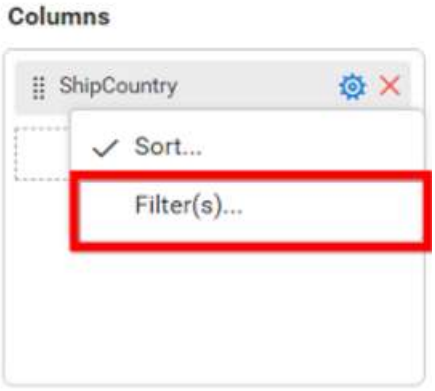
The drilled view of the chart region selected.

You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filter in settings.For more details, refer filter.

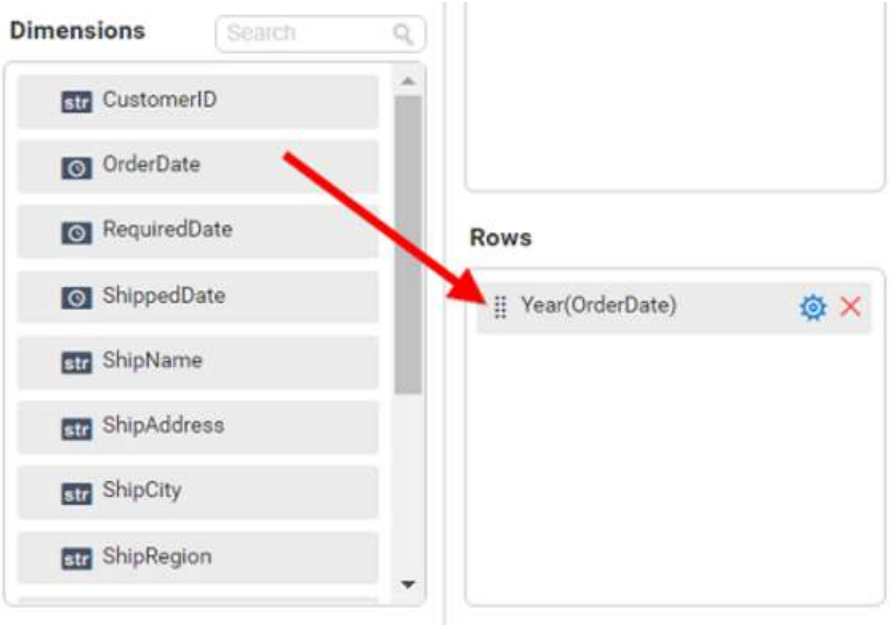


To show all records click on Show All Records.

Similarly you can add the Measures and Expressions into columns field.

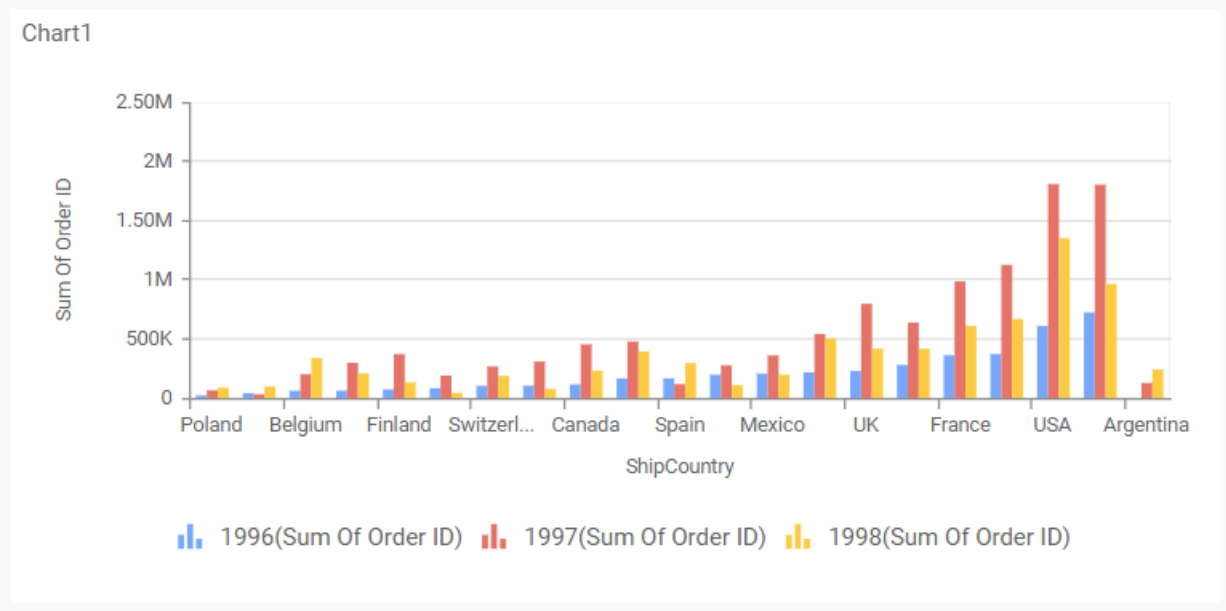
Adding Rows

You can drag and drop the Measure or Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render chart in series.

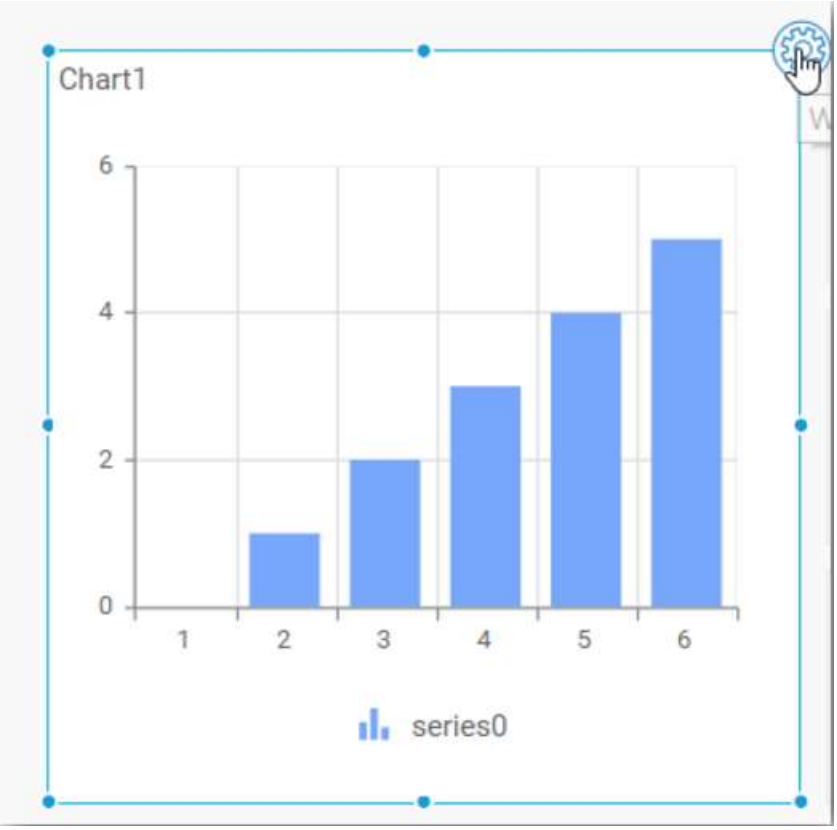


How to format column chart?

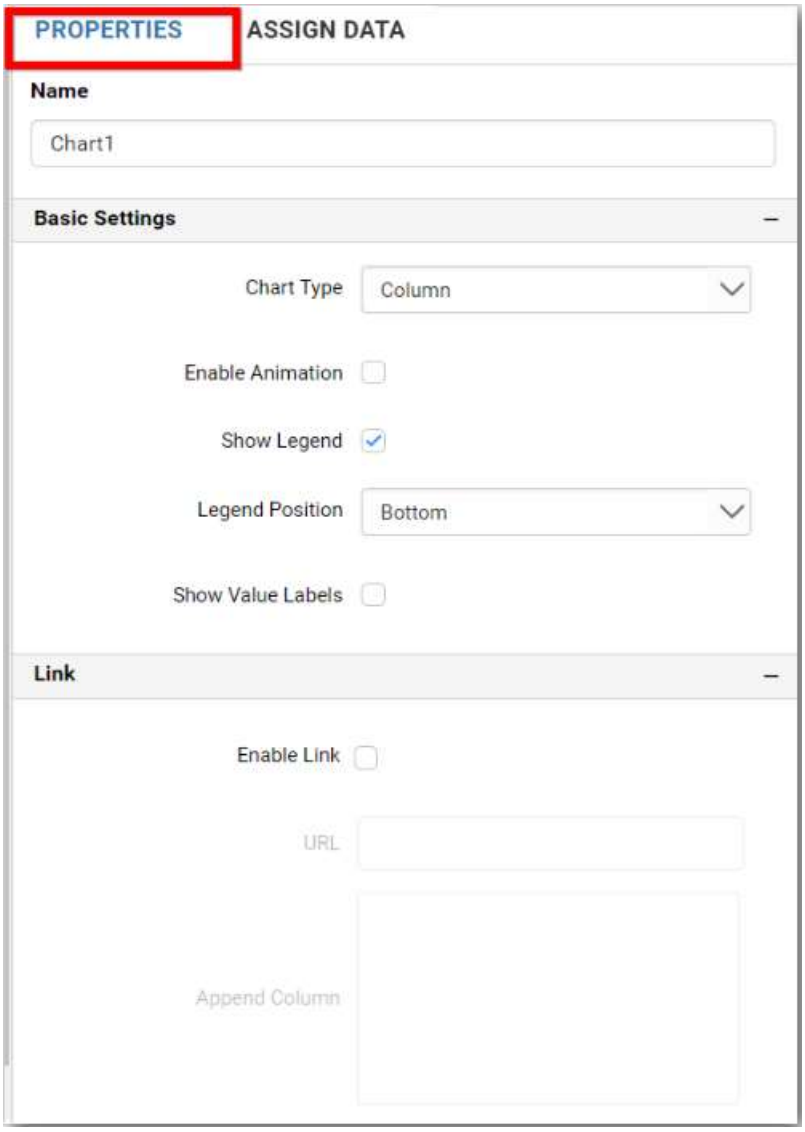
You can format the column chart for better illustration of the view that you require, through the settings available in Properties tab.

To format bar chart follow the steps

1. Drag and drop the column chart into canvas and resize it to your required size. 2. Configure the data into column chart. 3. Focus on the column chart and Click on Widget Settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this column chart widget

Basic Settings

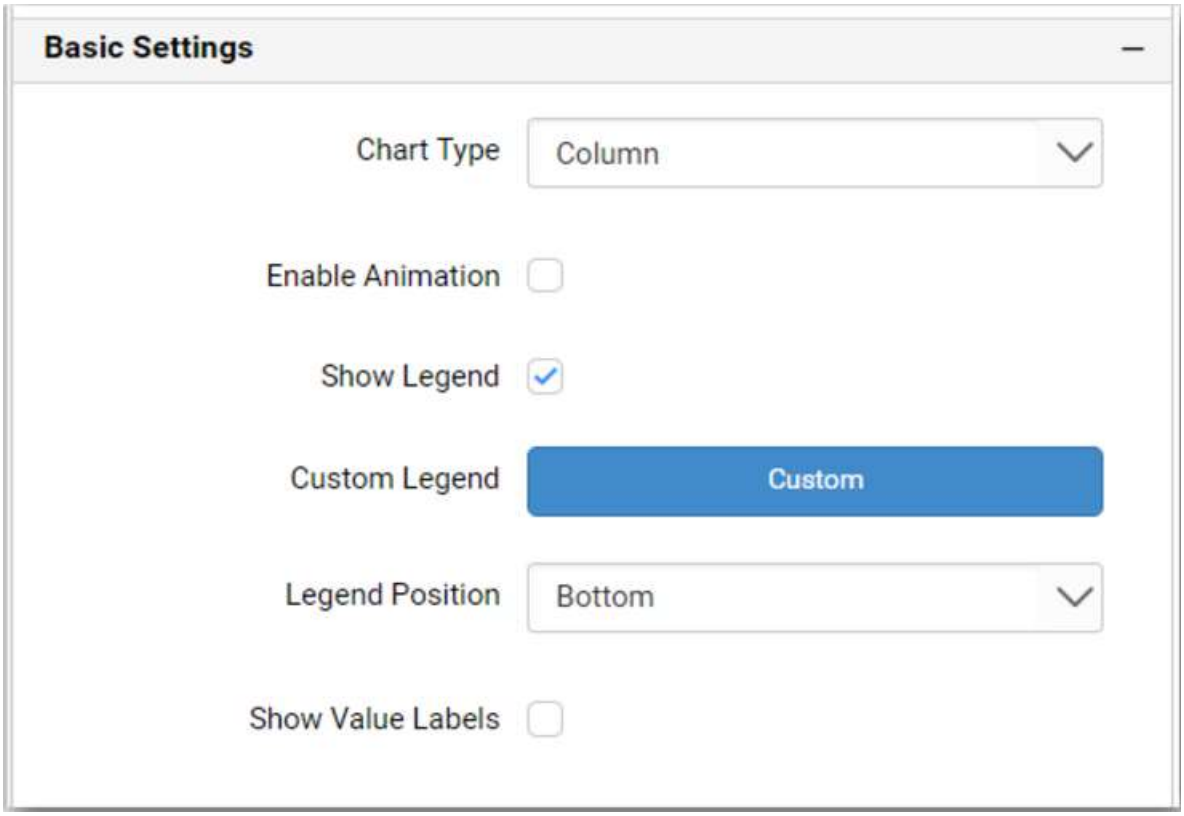


Chart Type

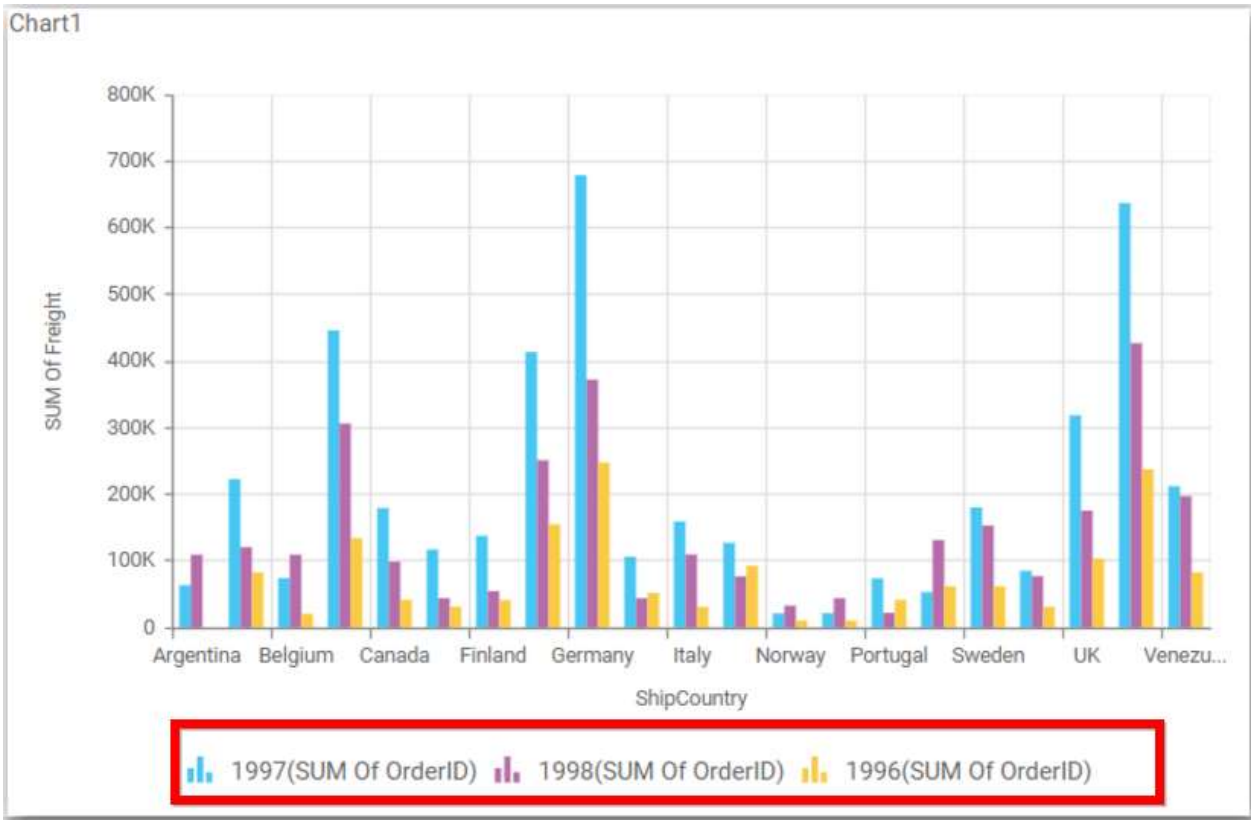
This allows you to switch the widget view from current chart type to another convertible chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

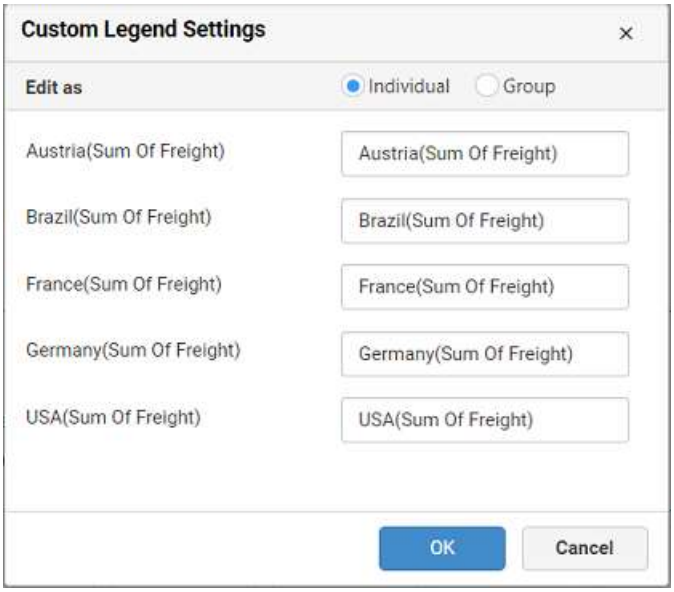
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

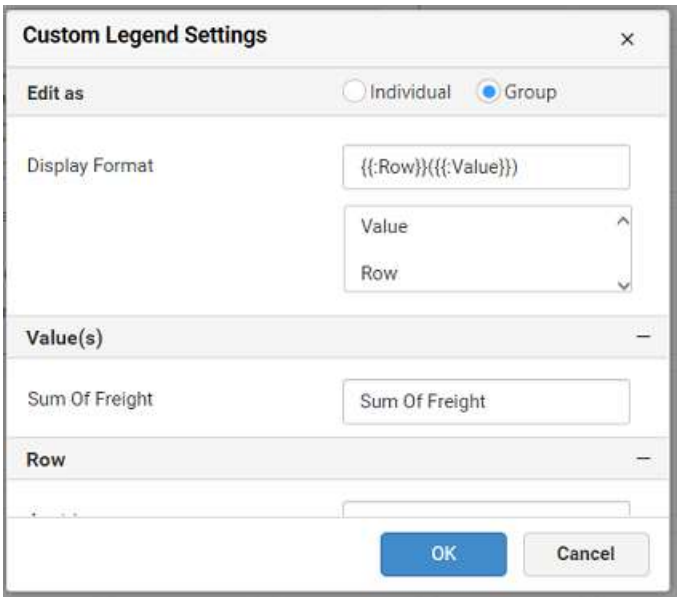
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.

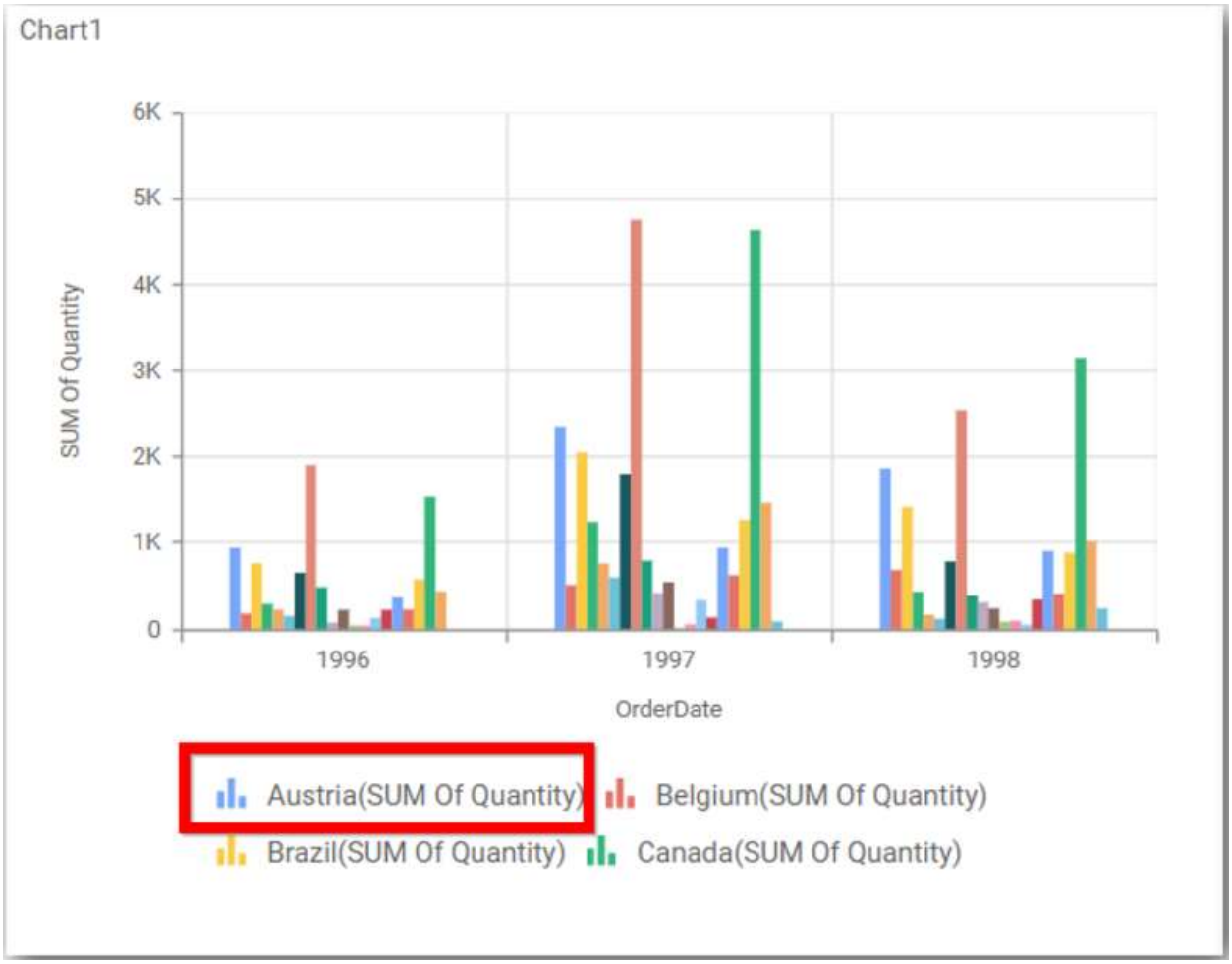


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

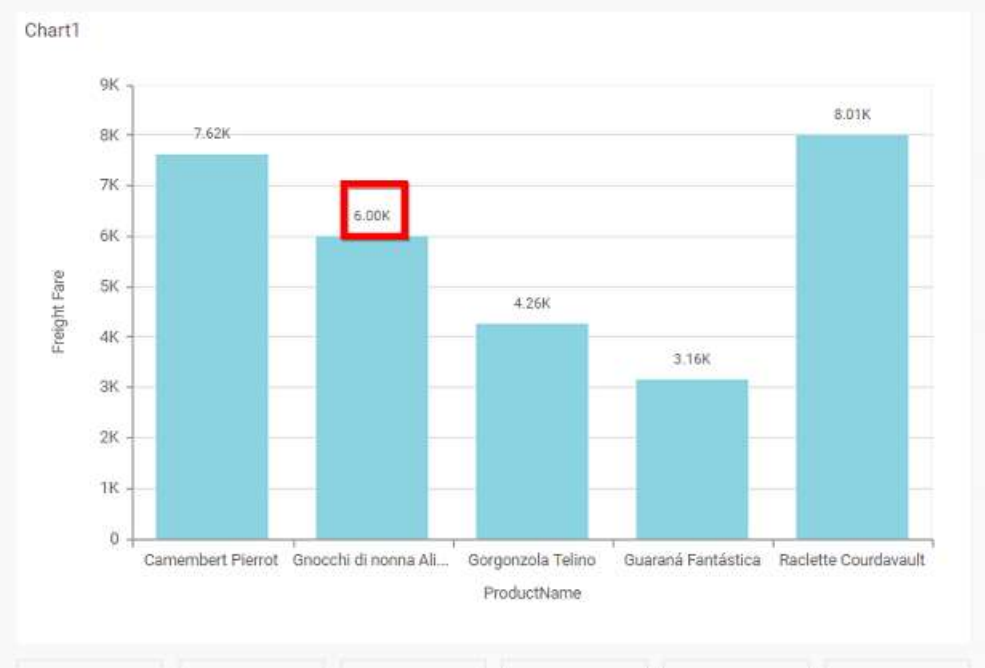


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Austria (Sum of Freight)



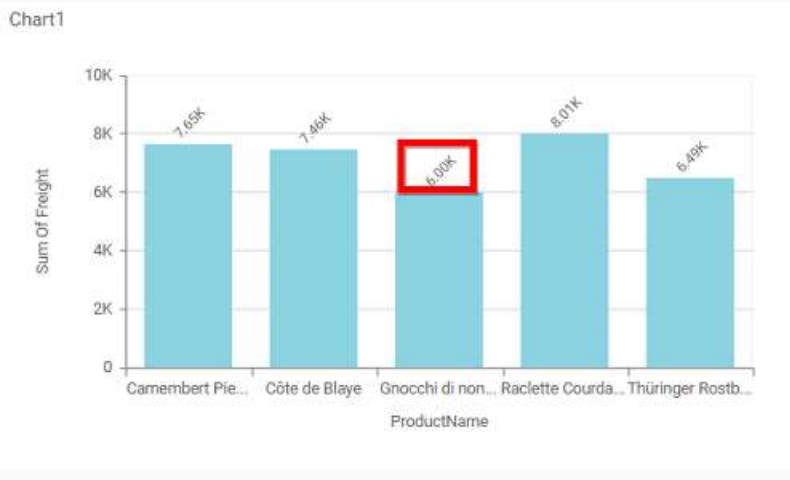
Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

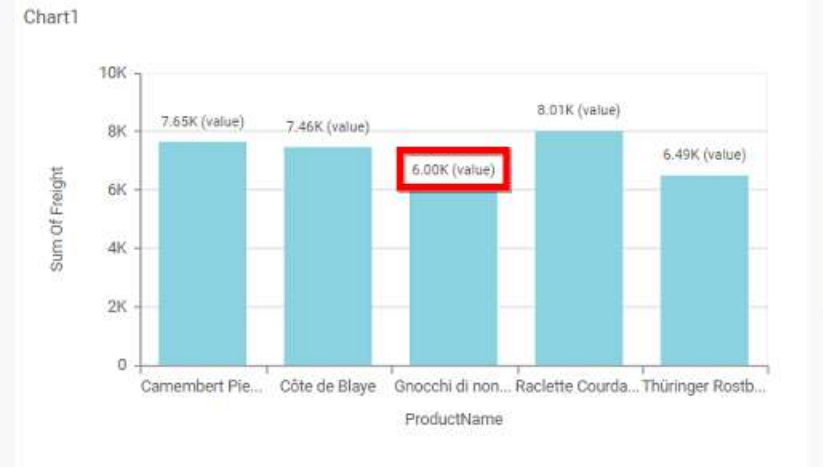


Value Label Suffix

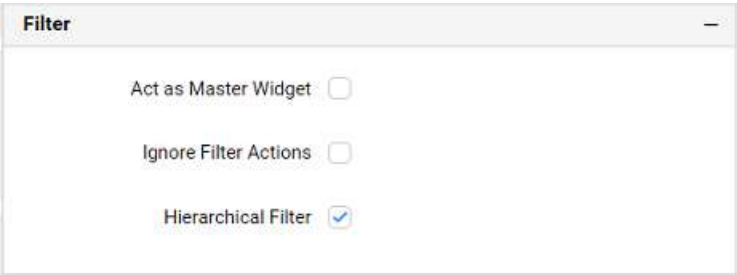
Allows you to enable the Suffix value text to the value labels.

Suffix Value

Allows you to set\edit suffix value to the value labels.



Filter



Act as Master Widget

This allows you to define this column chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

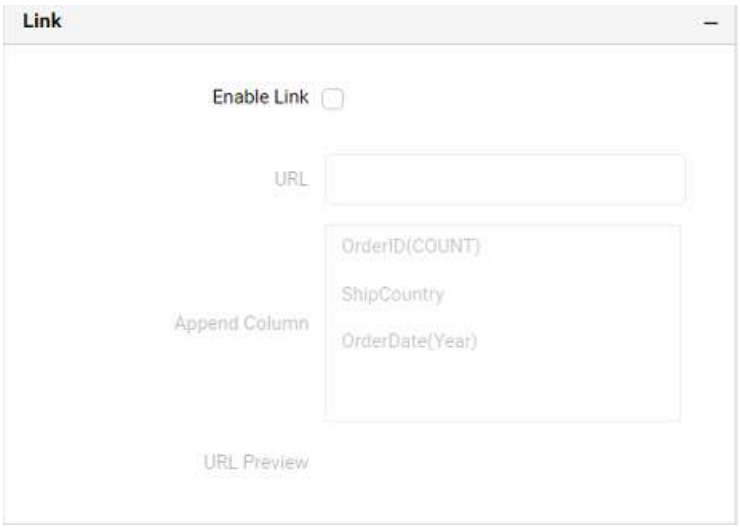
This allows you to define this column chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

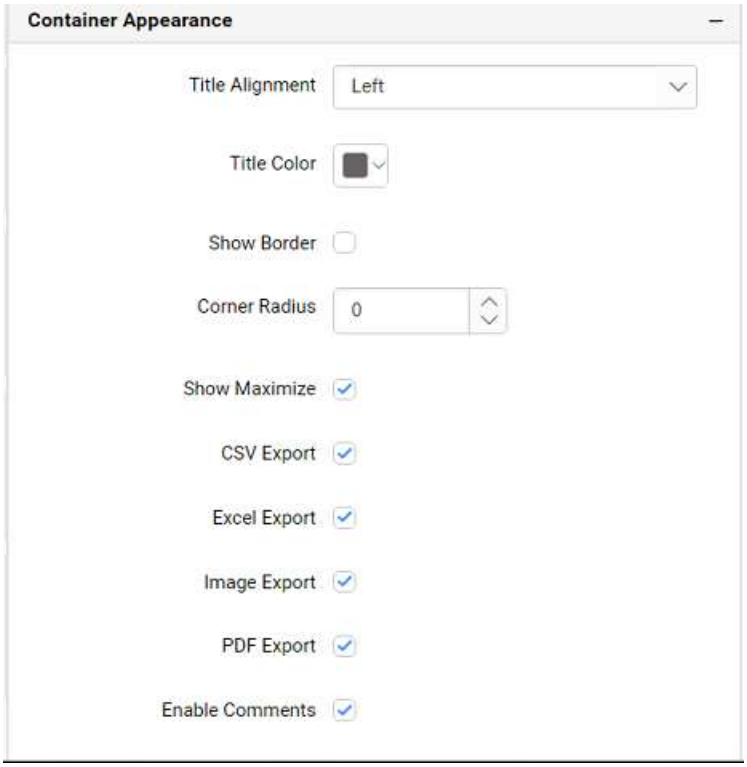
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this column chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this column chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this column chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

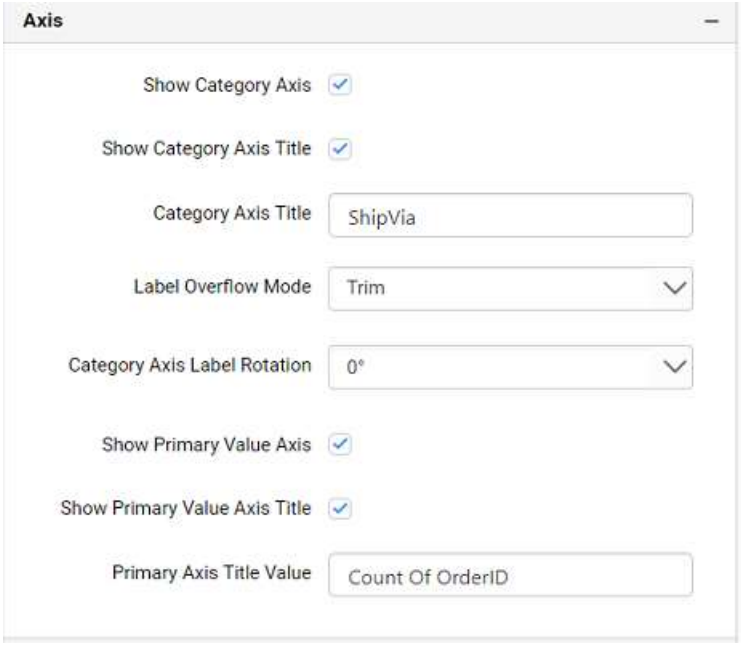
Image Export

This allows you to enable/disable the image export option for this column chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

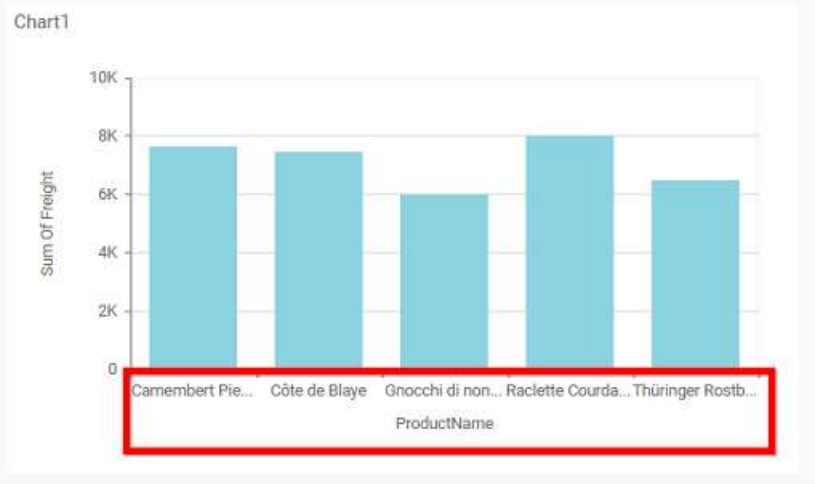
Axis



This section allows you to customize the axis settings in chart.

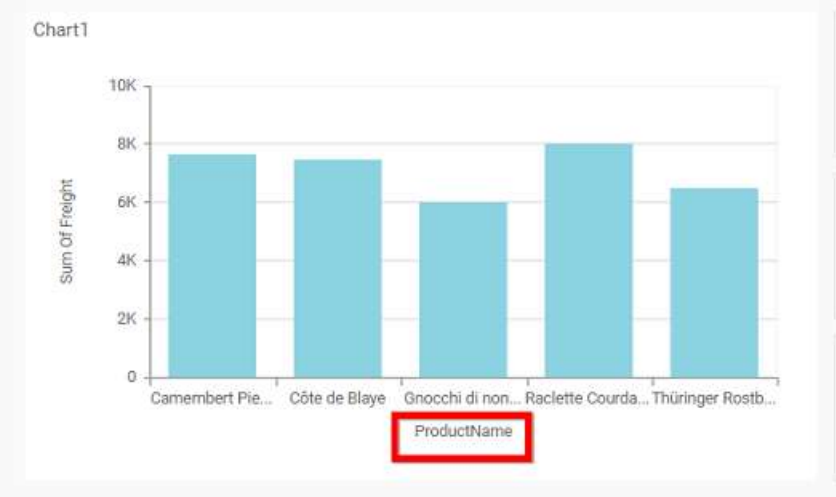
Show Category Axis

This allows to enable the visibility of Category Axis.



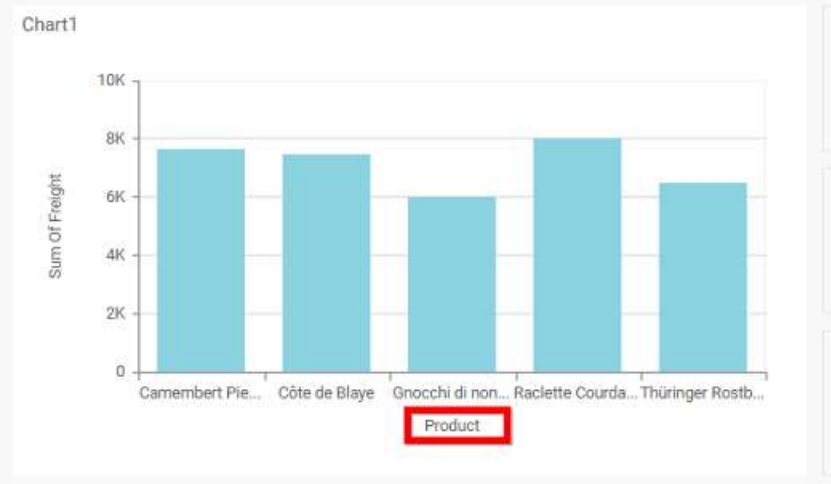
Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

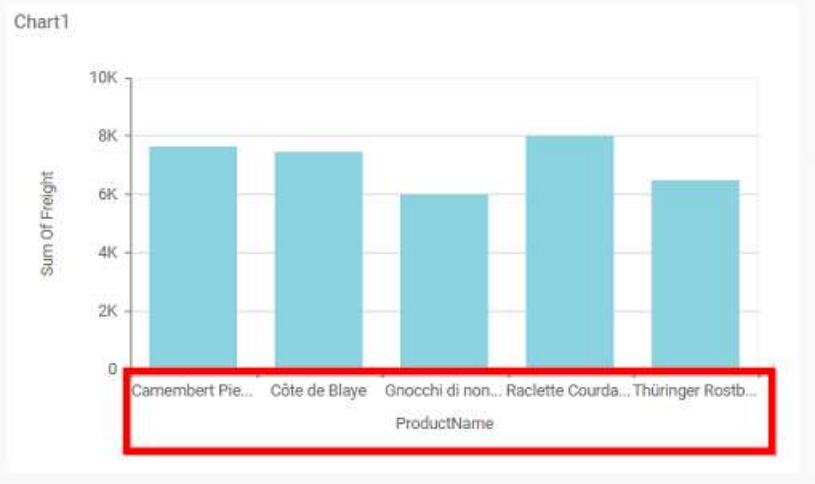


Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

Trim

This option trims the end of overlapping label in the axis.



Hide

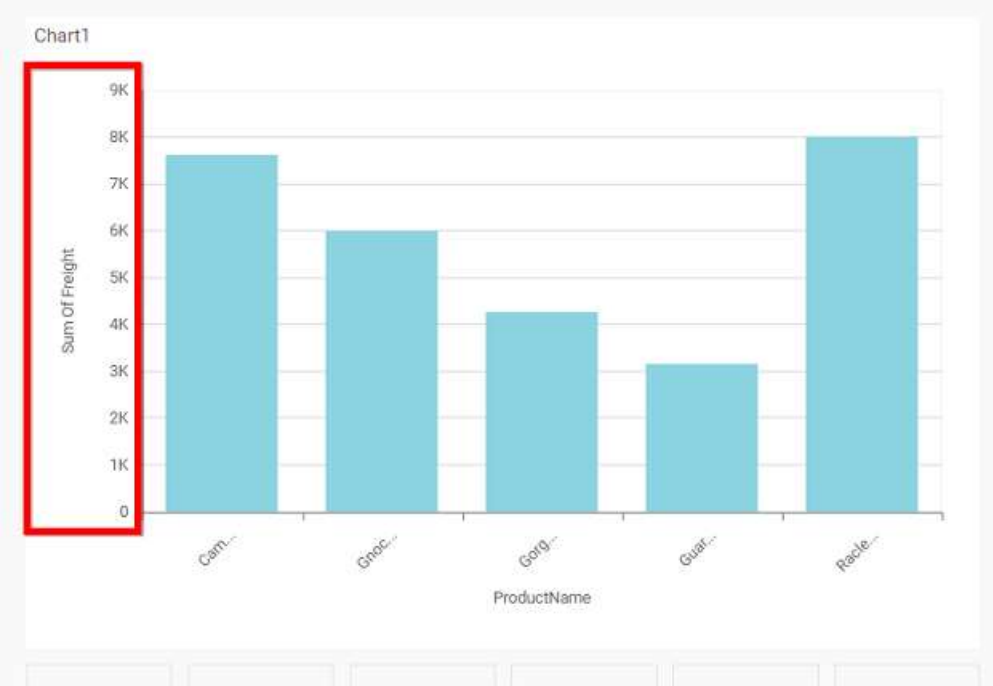
This option hides the overlapping label in the axis.

Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

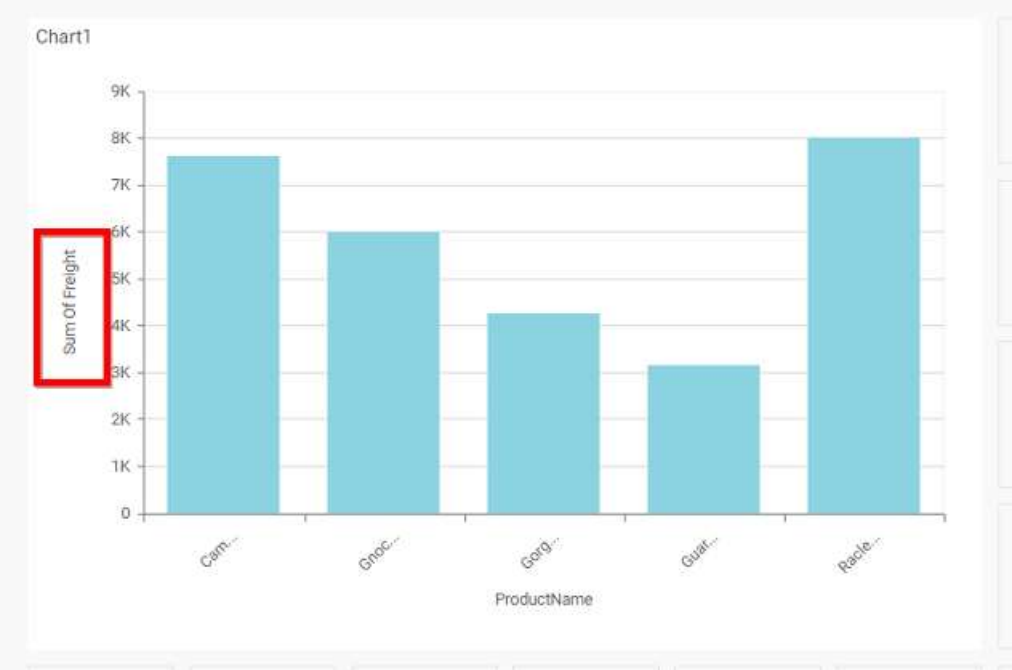
Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.



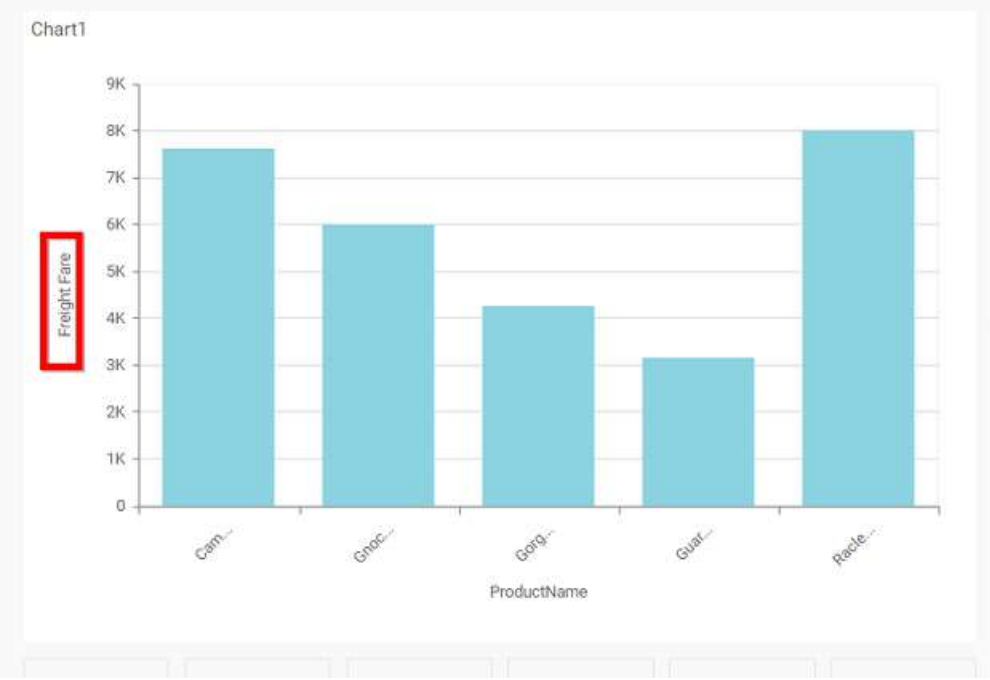
Show Primary Value Axis Title

This allows you to enable the visibility of Primary Value Axis title of chart.

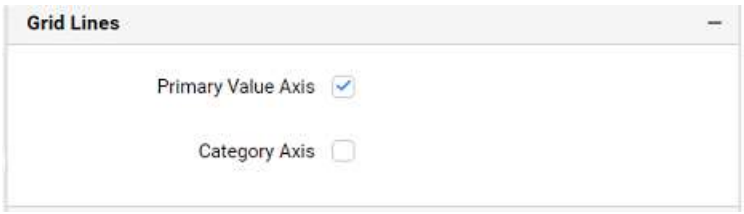


Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

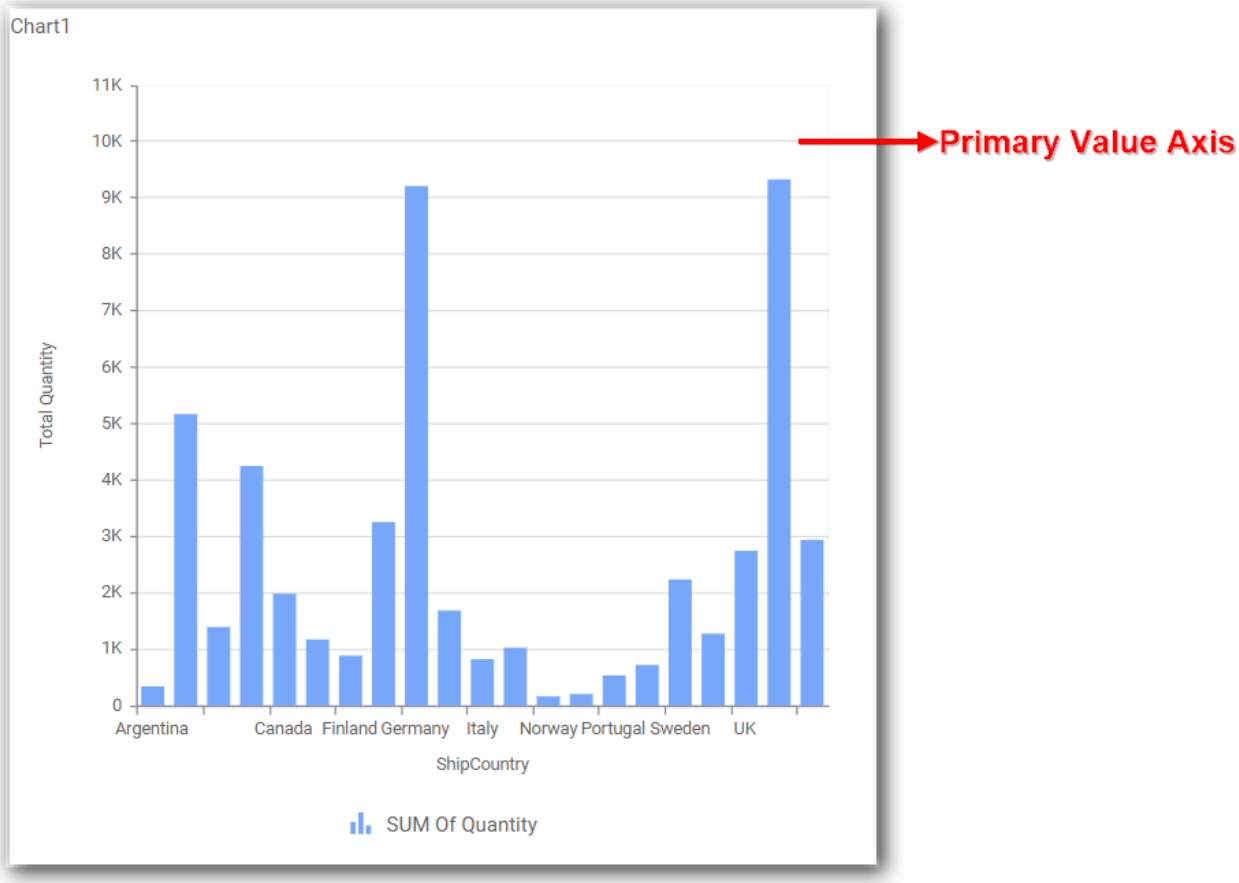


Grid Lines



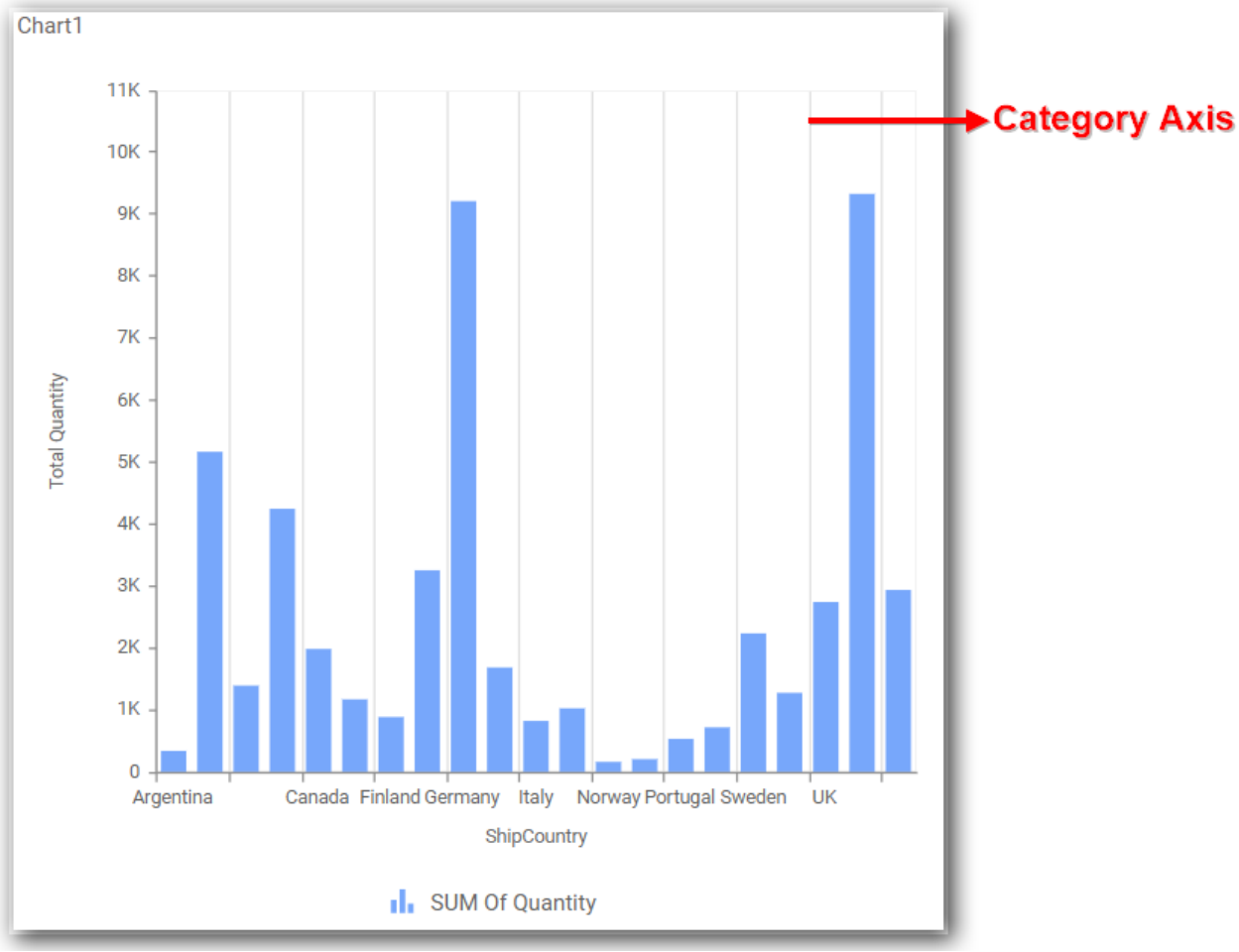
Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the column chart



Category Axis

This allows you to enable the Category Axis gridlines for the column chart.

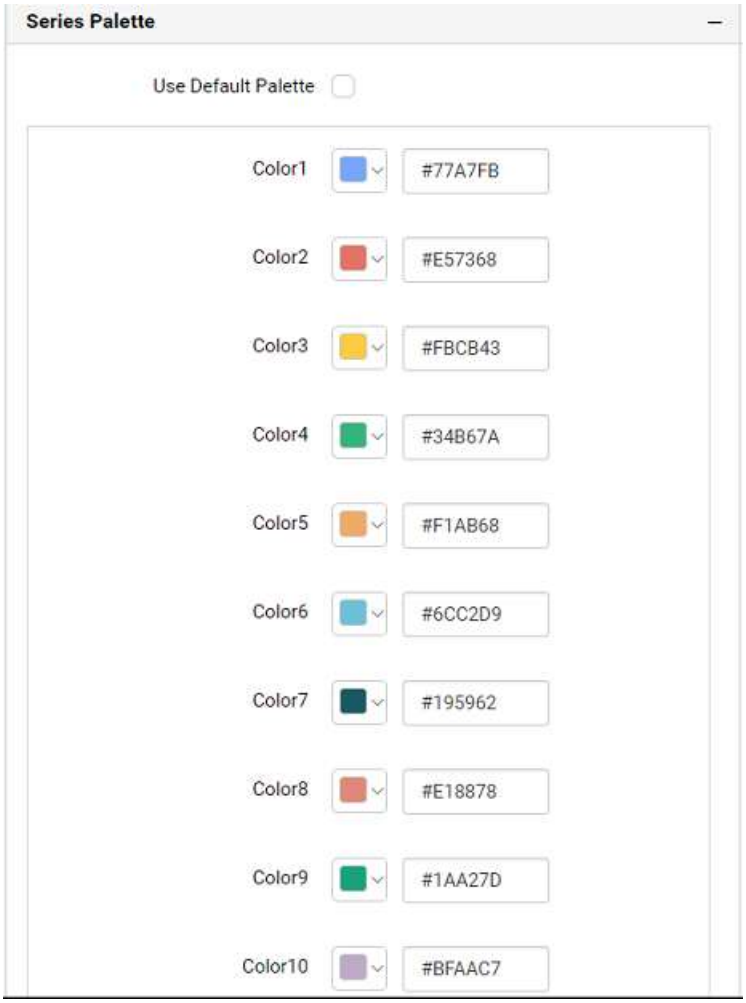


Series Palette

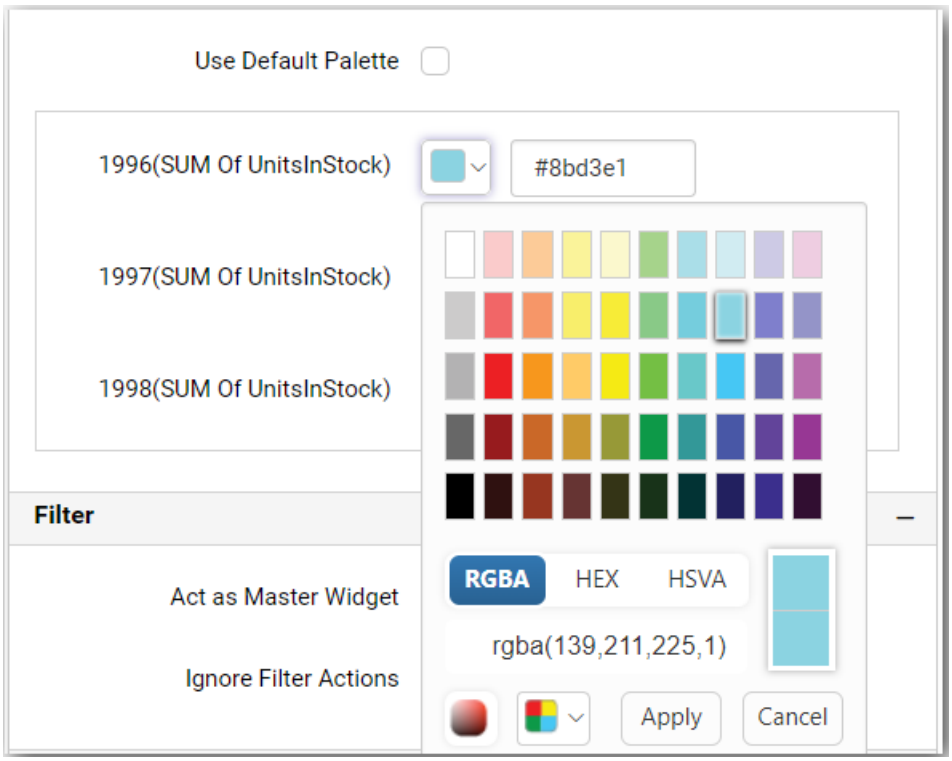
This allows you to customize the chart series color through Series Palette section.

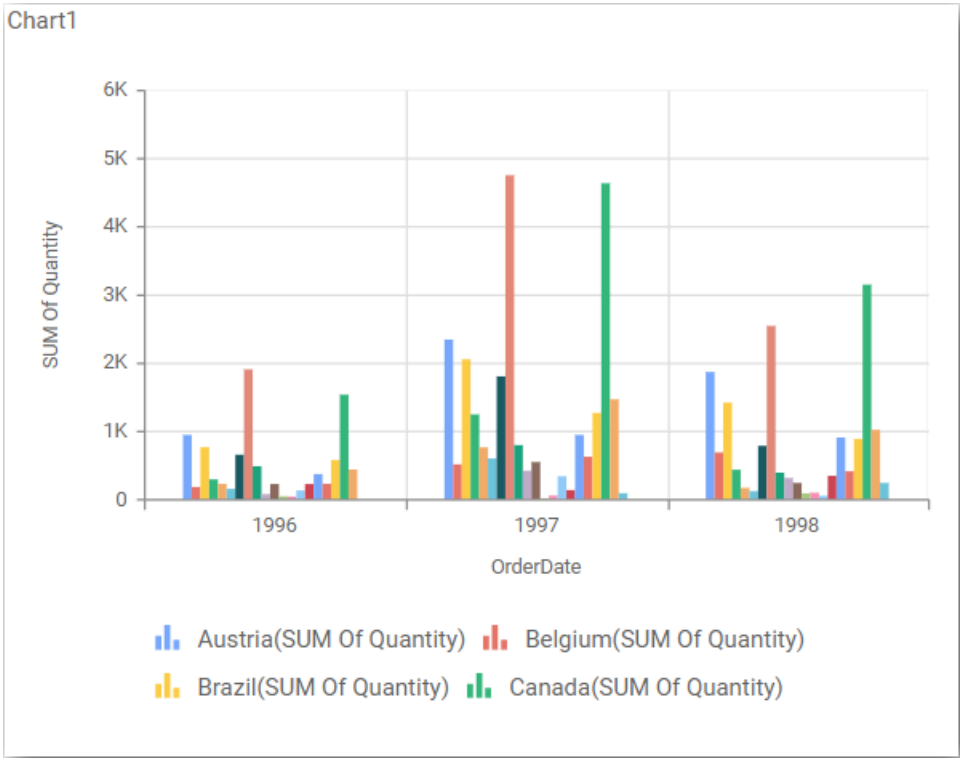
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





How to apply conditional formatting?

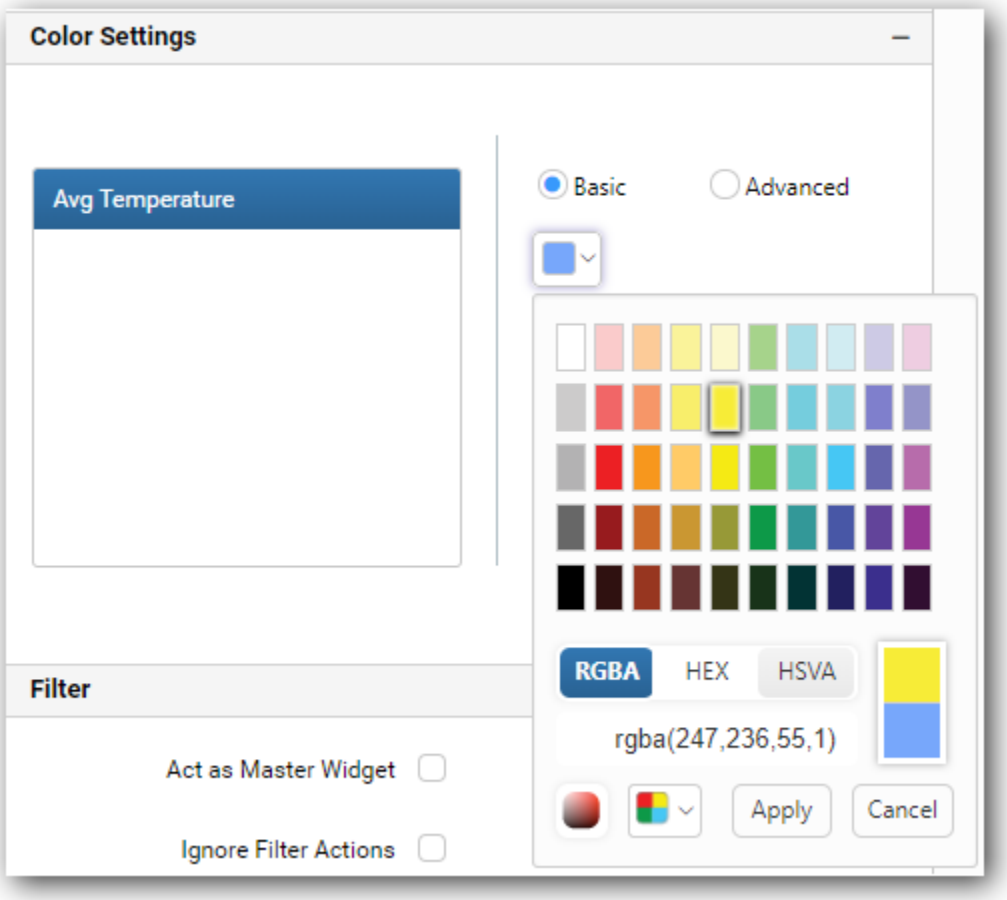
Color in column chart widget can be customized using the color settings which is available in the properties section. This will allow the user to improve the visualization in column chart and to distinguish the data based on conditional range values that will let the visualizer to understand what is shown in data.

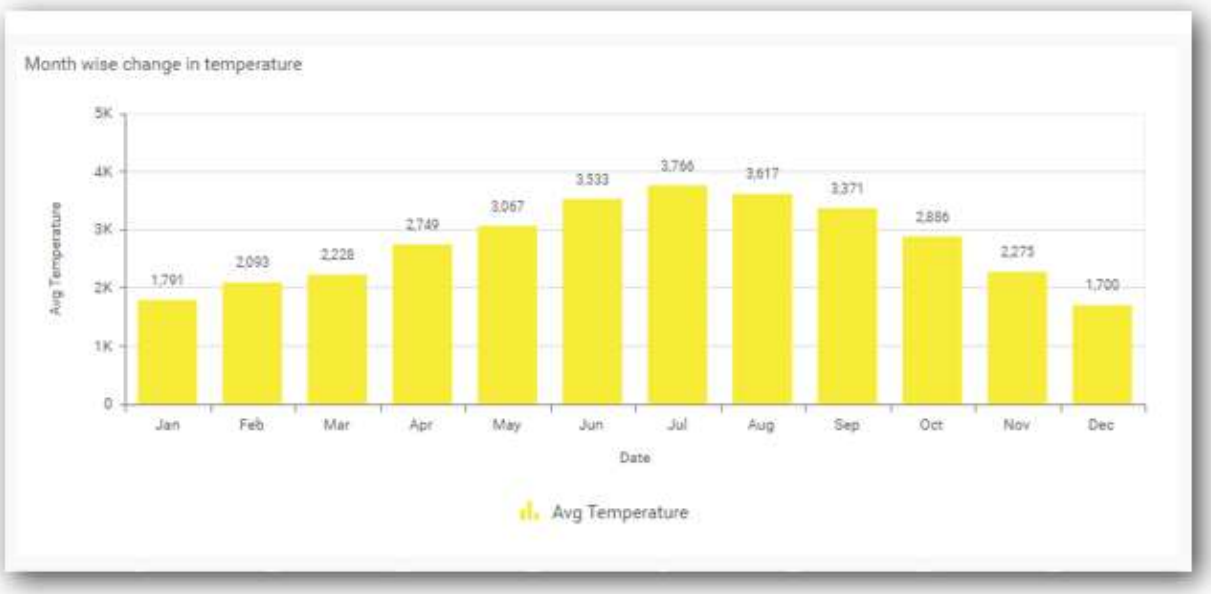
Color Settings

This allows you to customize chart series color based on user provided conditions.

Basic Settings

This section allows you to customize the series color with series label on the left-hand side and corresponding series color on the right-hand side. You can choose a color. And, you can also change the series color by changing the corresponding color value in the right-hand side.

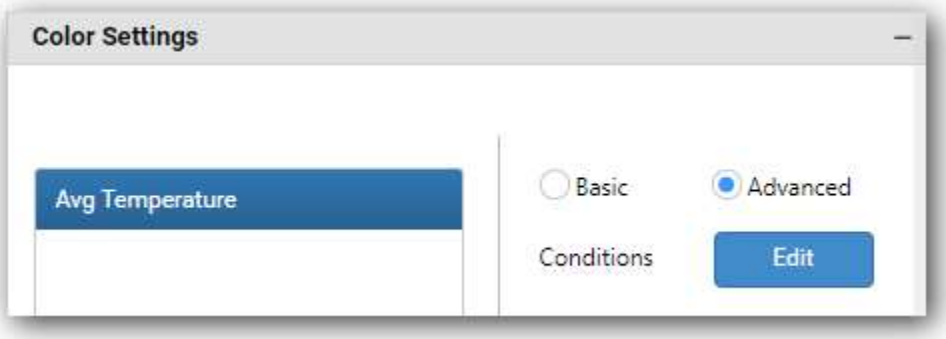


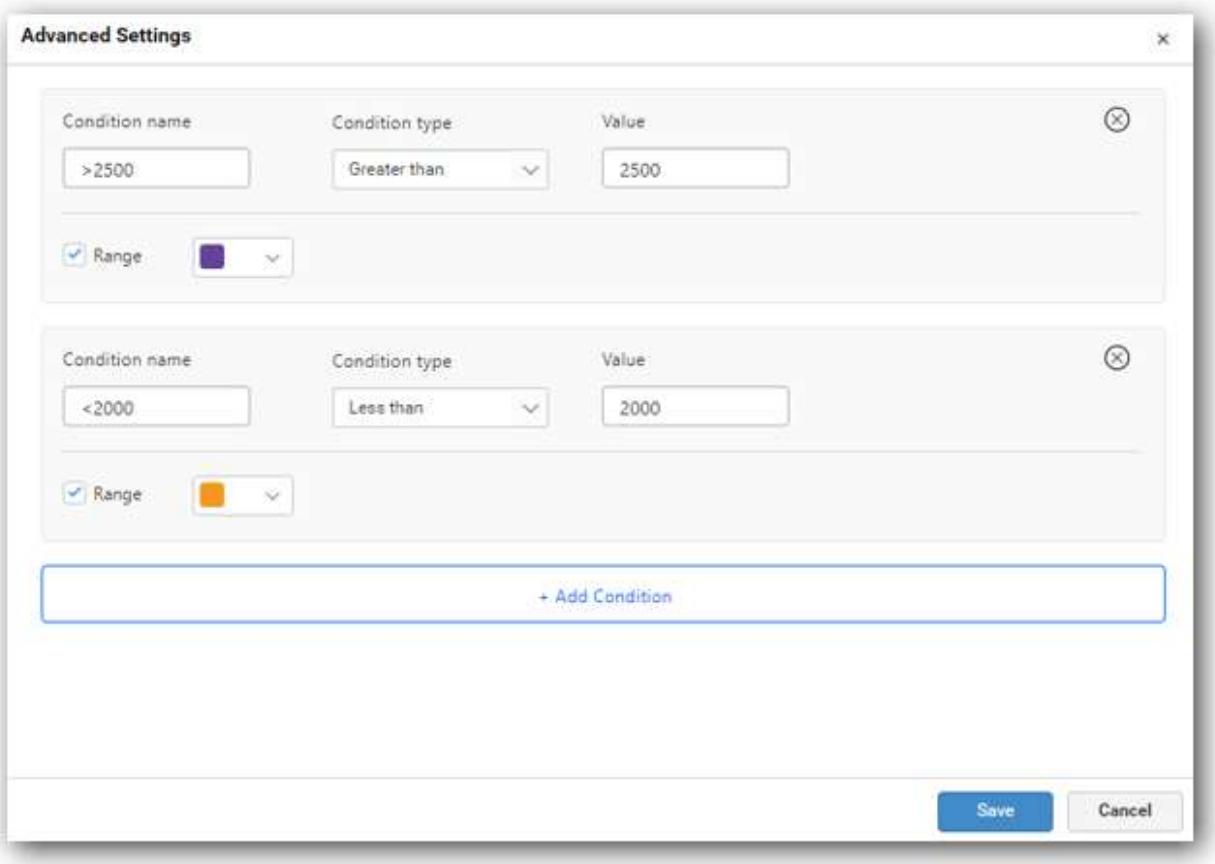


Advanced Settings

This allows you to provide colors for the selected measure based on single or multiple conditions available.

You can choose a series color using multiple condition sets such as Greater than, Less than, Equal to, Not Equal to, Between, Not between, Greater than or equal to, Less than or equal to.





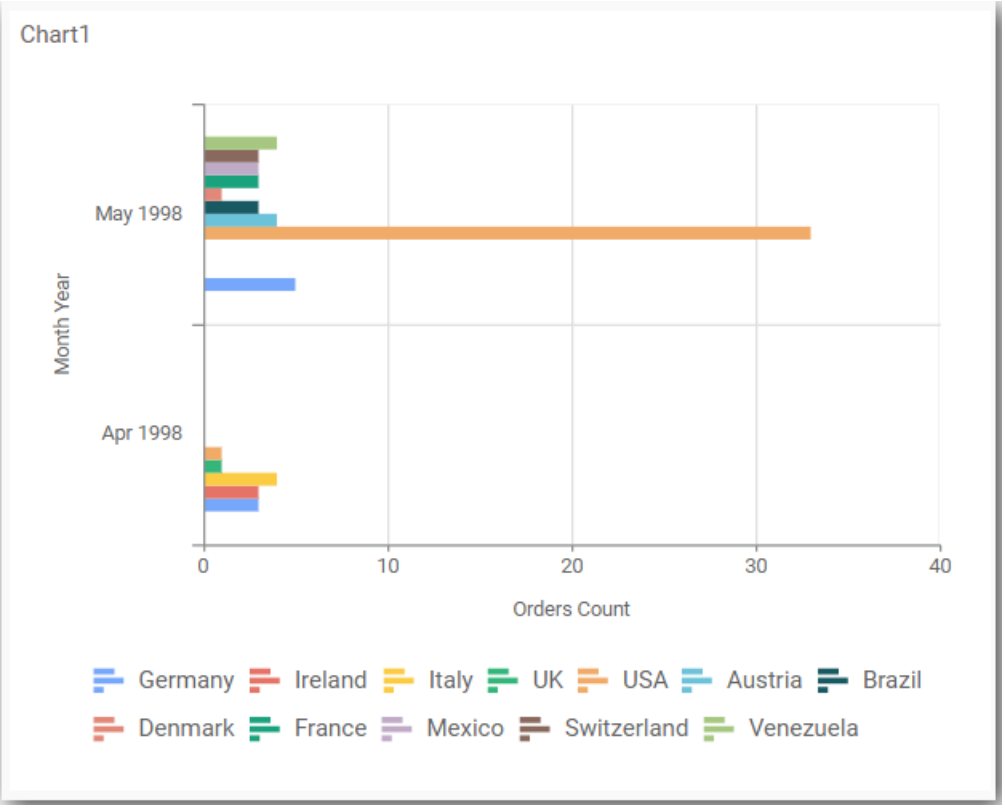
The chart will render with series color based on the provided conditions and the series which is outside the applied conditional range, will have basic color applied.

You can also customize the legend name by providing necessary values inside condition name.

Note: Advanced setting will be available only for chart with single measure value.

Bar Chart

Bar Chart allows you to compare values for a set of unordered items across categories through horizontal bars ordered vertically.

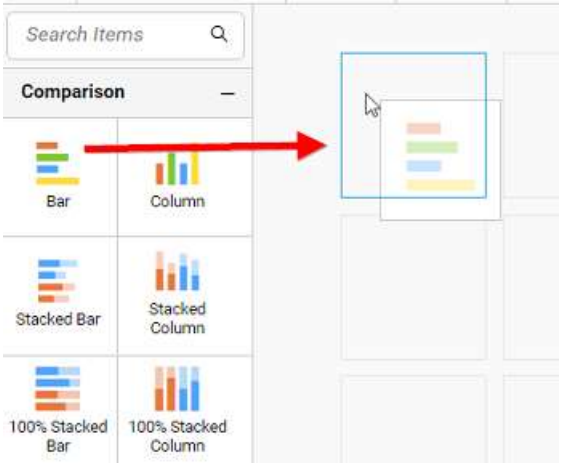


How to configure the table data to bar chart?

Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure column or expression column that you would like to analyze can be dropped into Y Values block. The dimension column that you would like to categorize the measure column, can be dropped onto Columns block. If you would like to categorize based on a series column, then the respective dimension column can be dropped onto Rows block in addition. These blocks are composed into Data pane.

Follow the steps to configure data into bar chart widget.

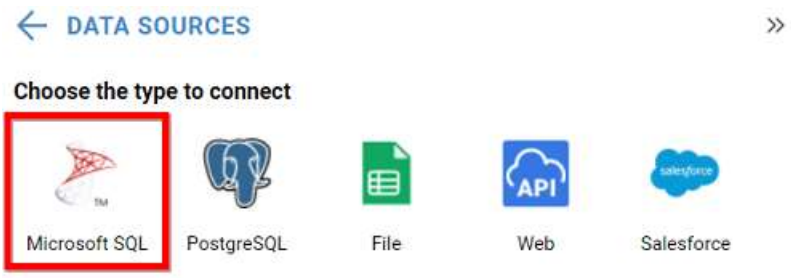
Drag and drop the bar chart widget into canvas and resize it to your required size.



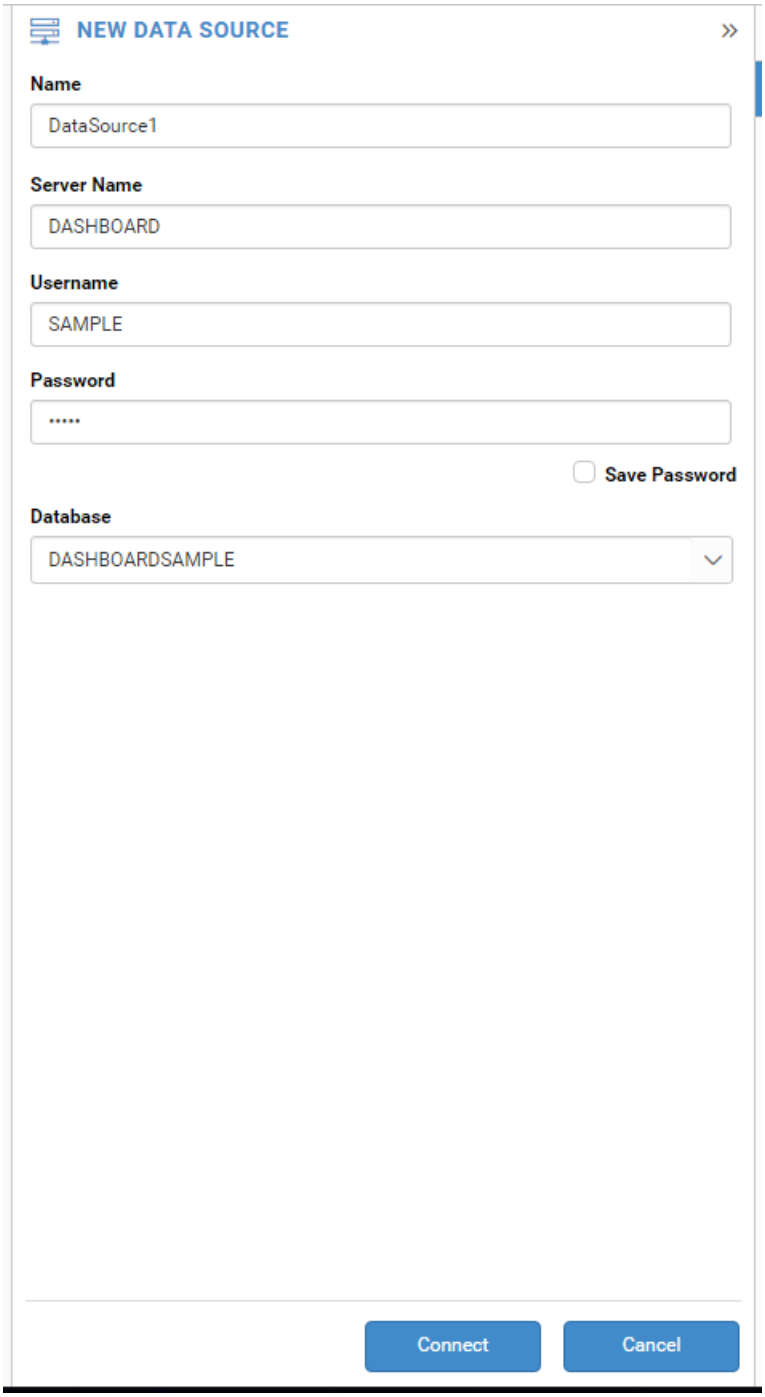
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a New Connection from connection type panel.

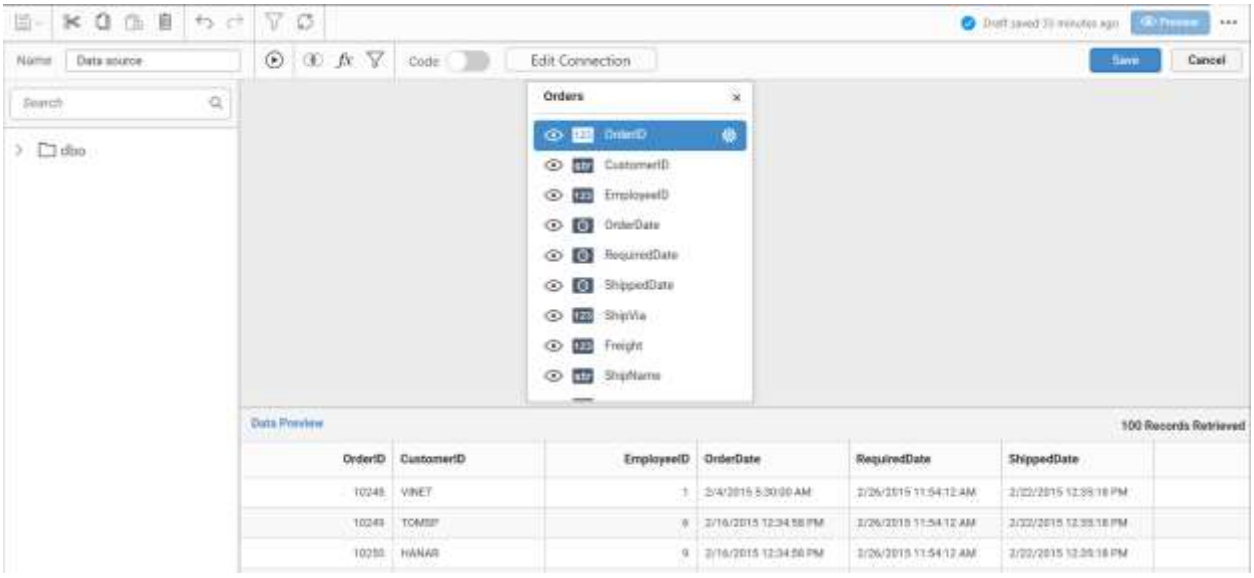
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



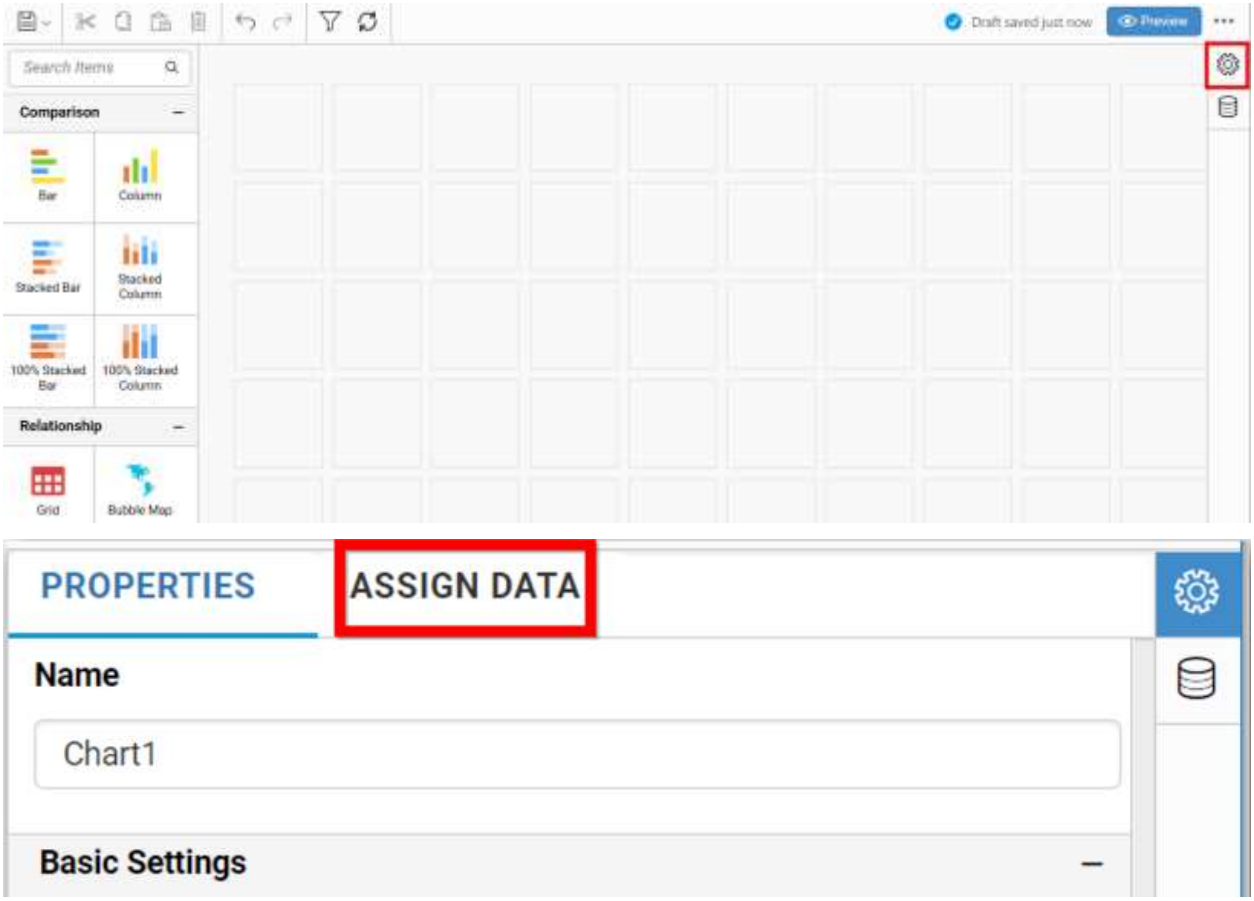
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



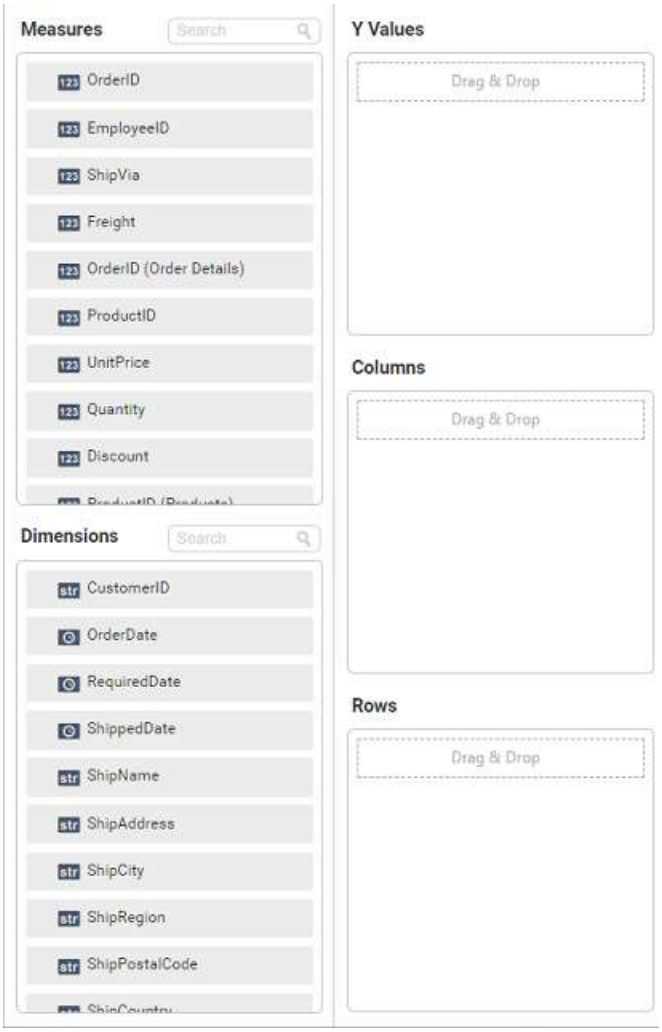
Drag your preferred table or view from the left pane from data design view, click Save button.



Click Properties button in Configuration panel, Property pane opens. Now, Switch to ASSIGN DATA tab.



The data tab will be opened with available measures and dimensions from the connected data source



Adding Y Values

You can add more than one Measures into Y Values field by drag and drop the required measure.

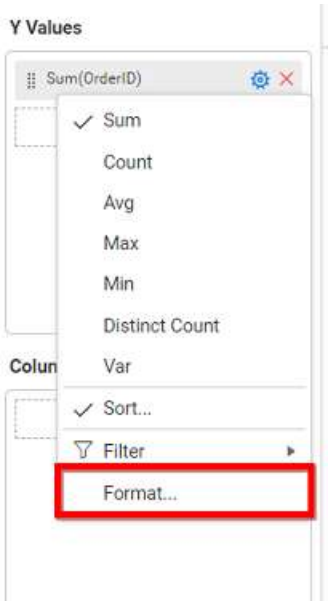
Now the chart will be rendered like this

Click the Settings option to change required summary type from the available summary types shown in Settings.

You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

You can filter the data to be displayed in chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format



To remove the added value fields click highlighted button.

Similarly you can add the Measures and Expression Columns into column field.

Adding Columns

You can add more than one value into Columns field.

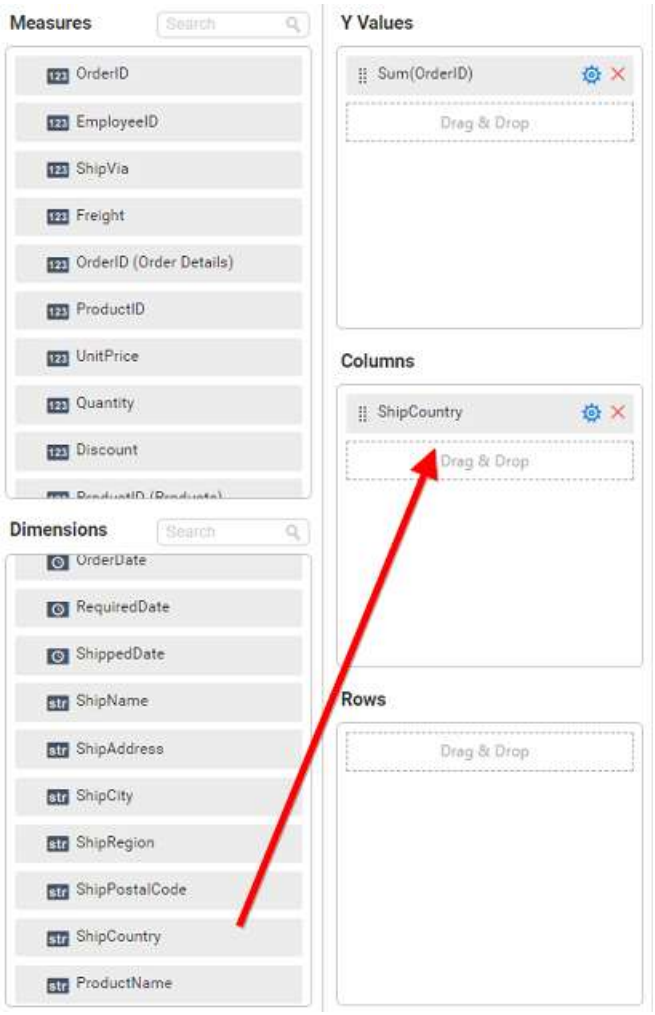
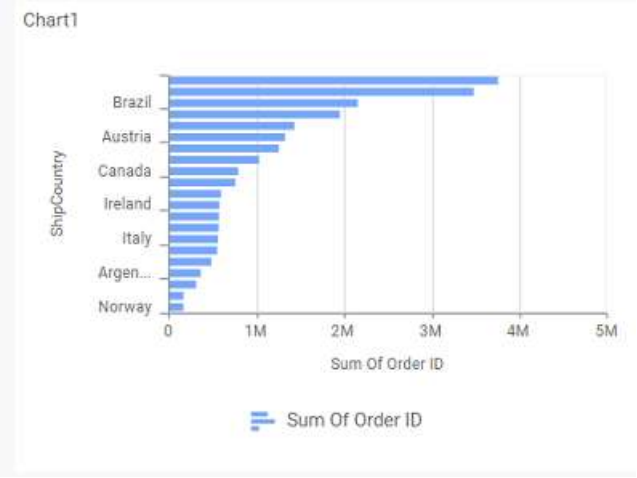


Chart will be rendered like this



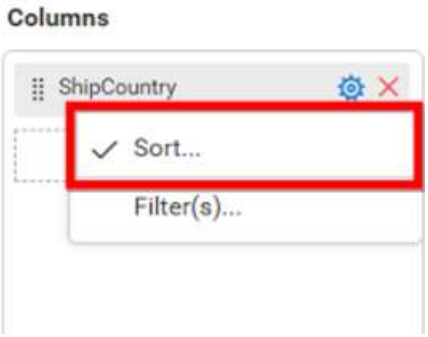
Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

Note: If you click No, single value will be added to the Columns field.

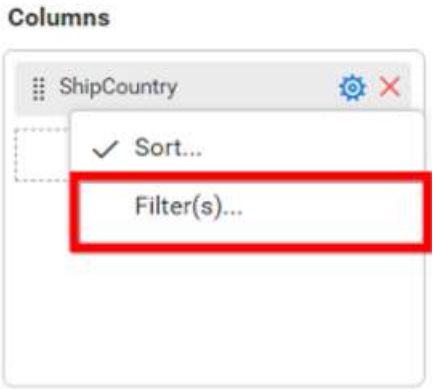
The drilled view of the chart region selected.

You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

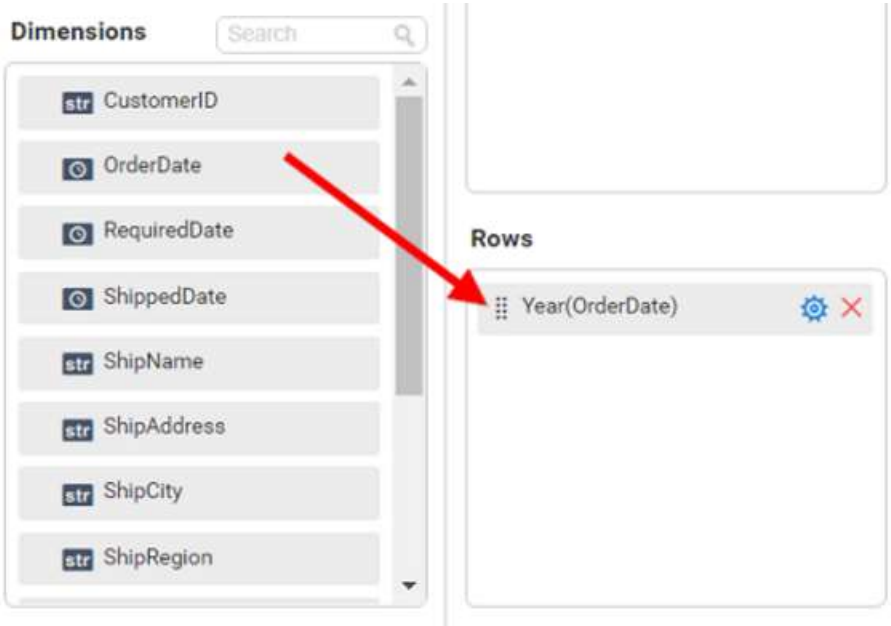


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

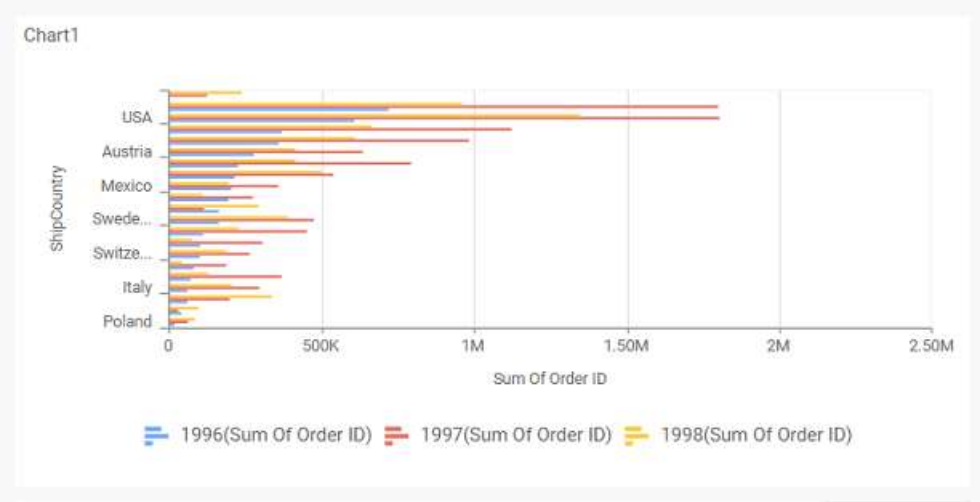
Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render chart in series.

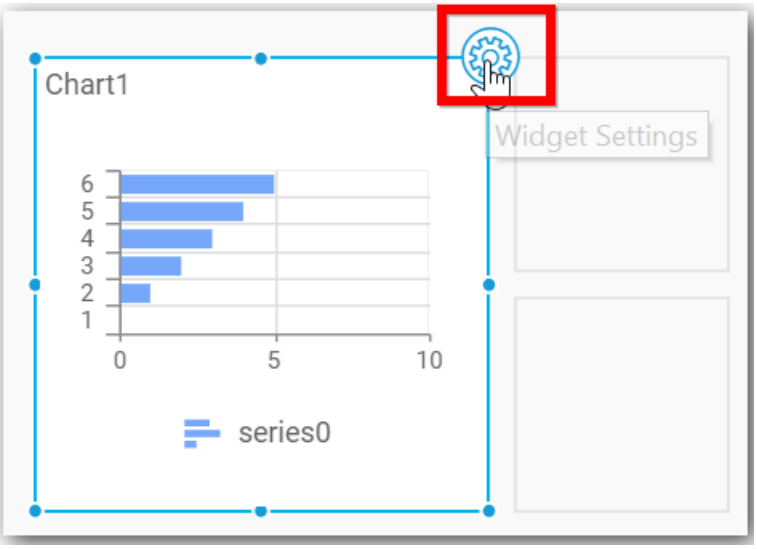


How to format bar chart?

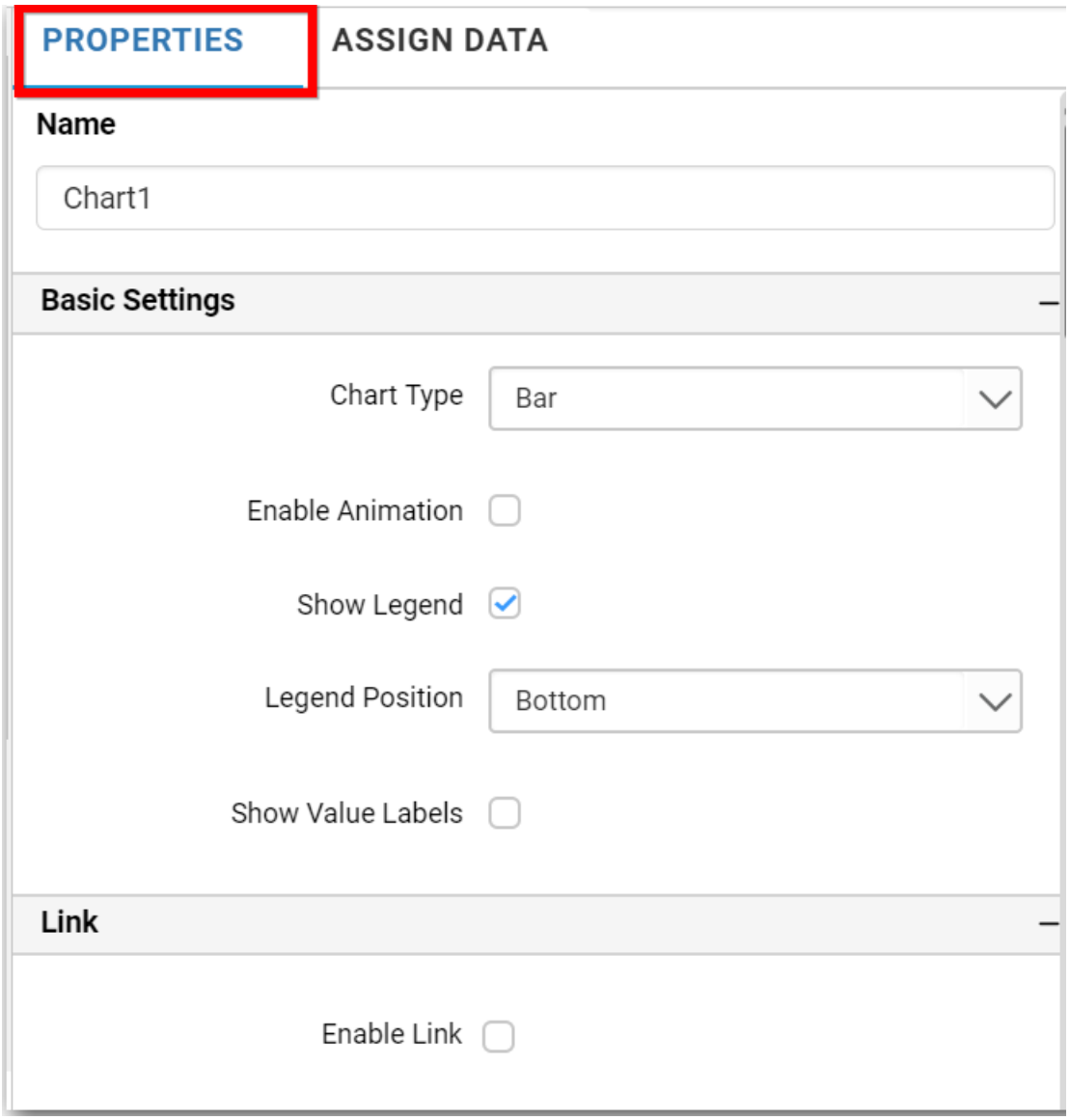
You can format the bar chart for better illustration of the view that you require, through the settings available in Properties tab.

To format bar chart follow the steps

1. Drag and drop the bar chart into canvas and resize it to your required size. 2. Configure the data into bar chart. 3. Focus on the bar chart and Click on Widget Settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this bar chart widget

Basic Settings

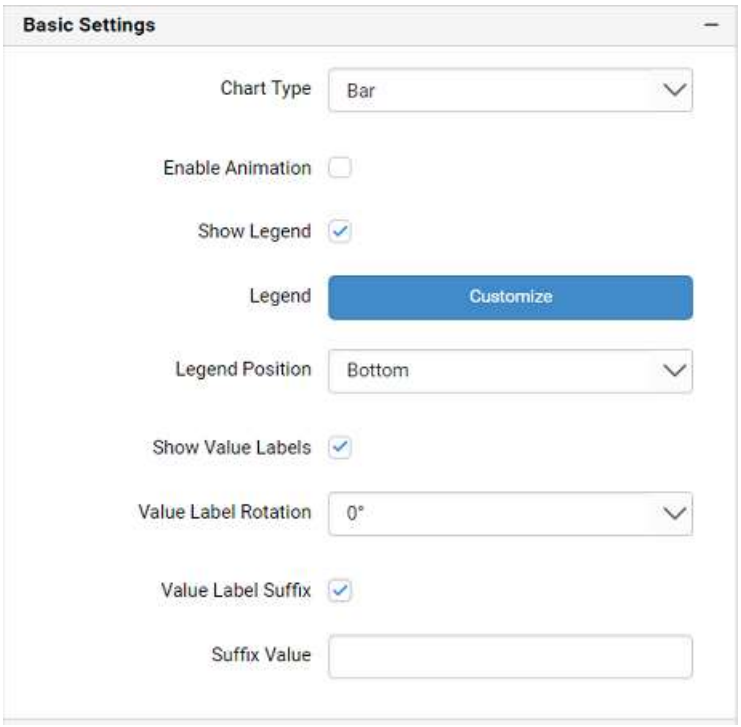


Chart Type

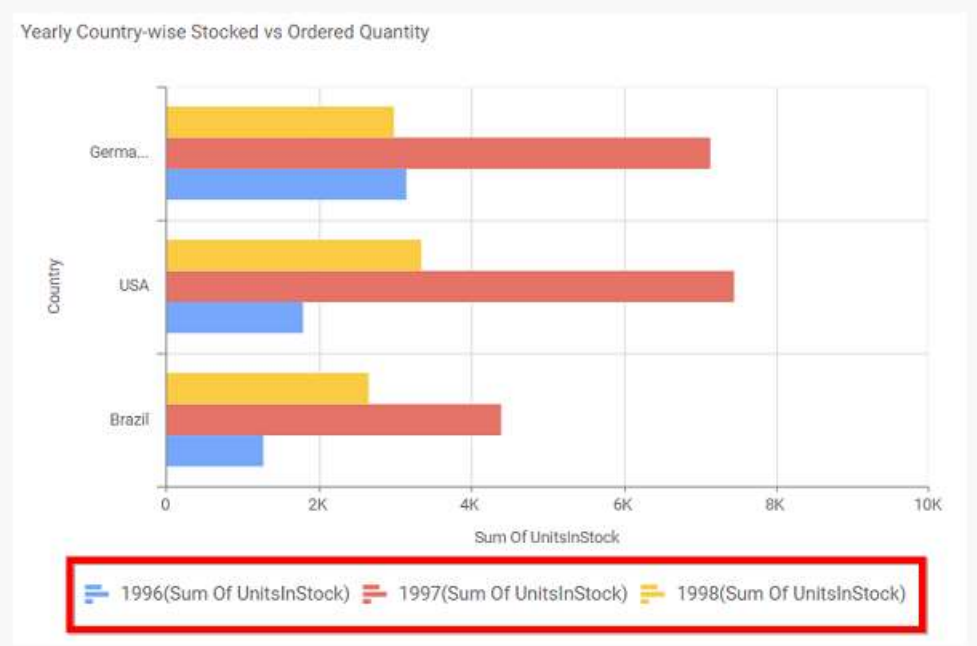
This allows you to switch the widget view from current chart type to another convertible chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

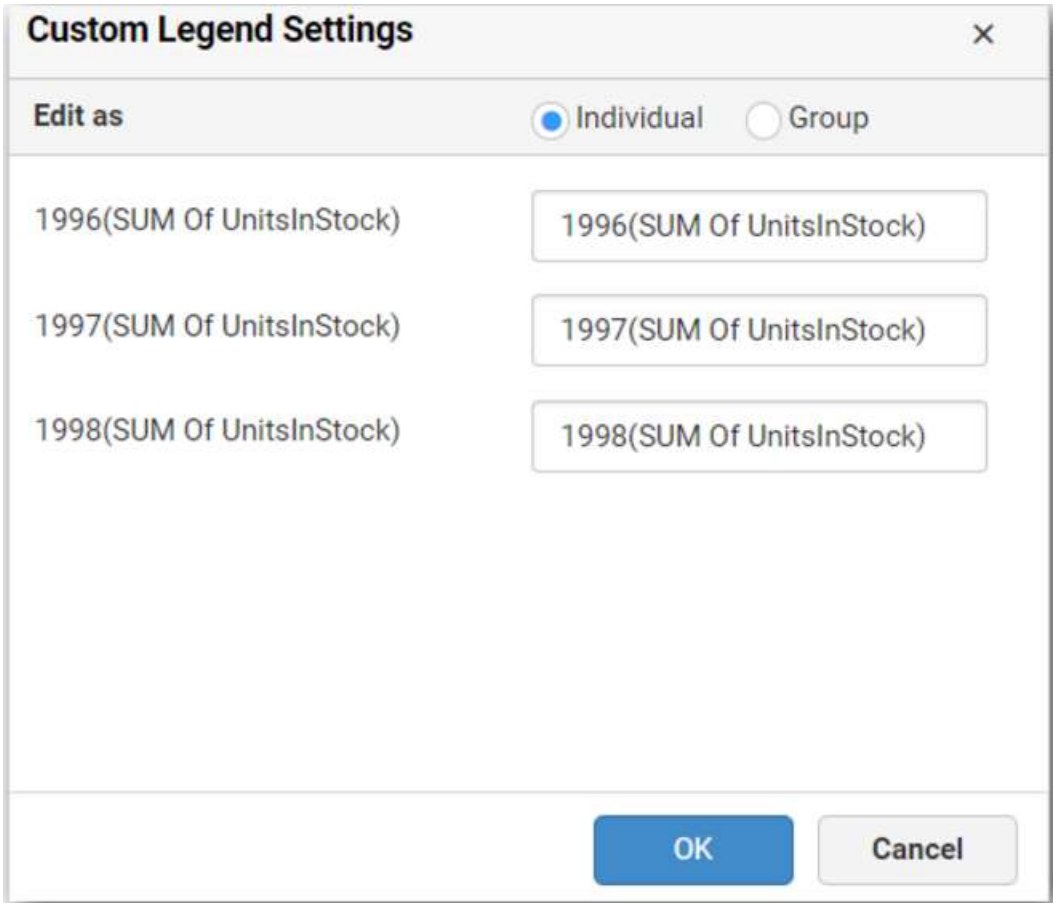
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

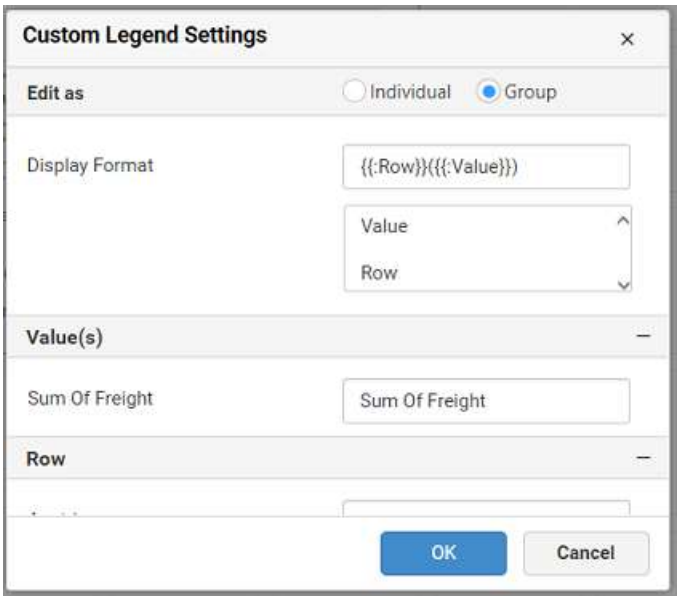
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.

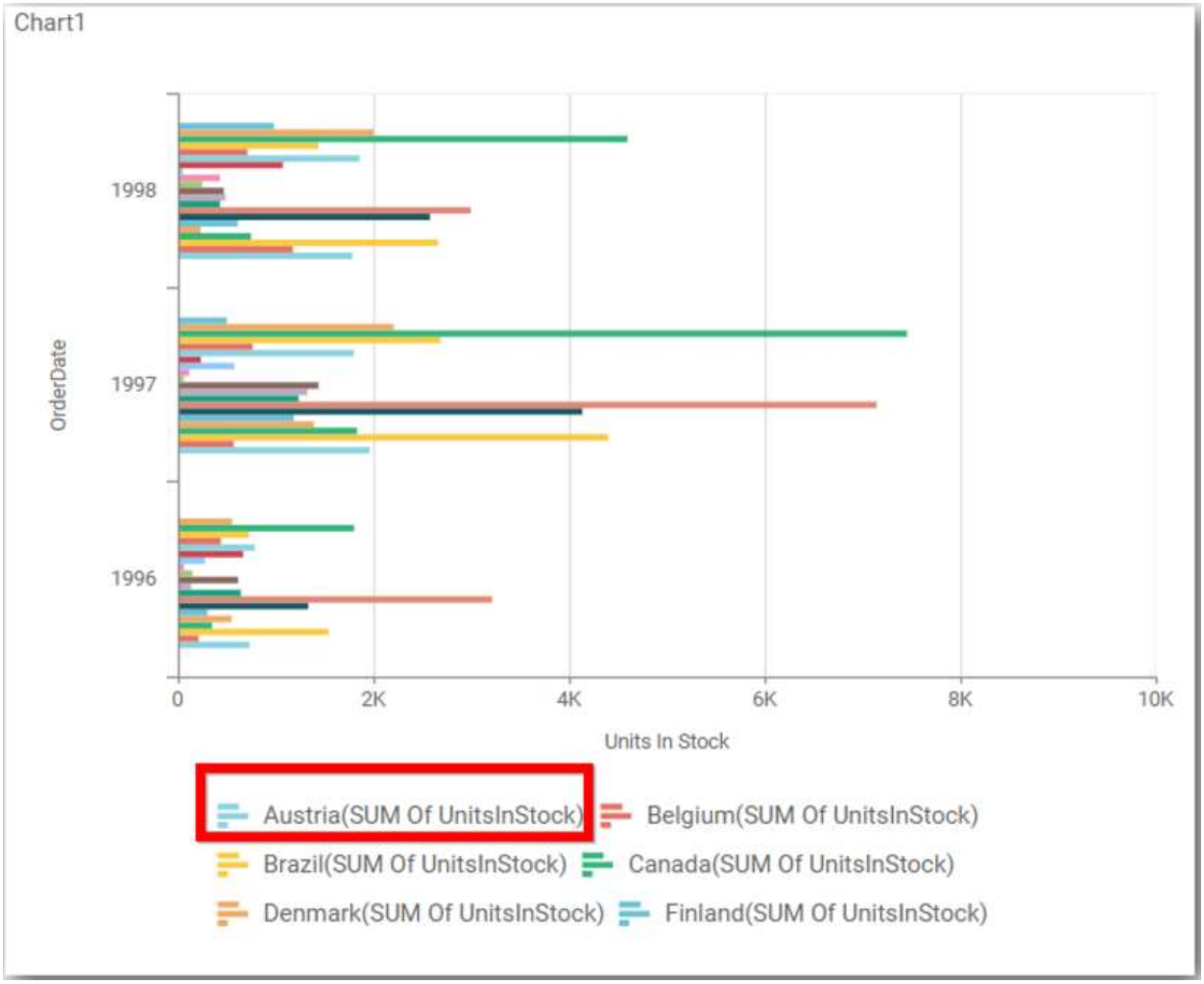


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

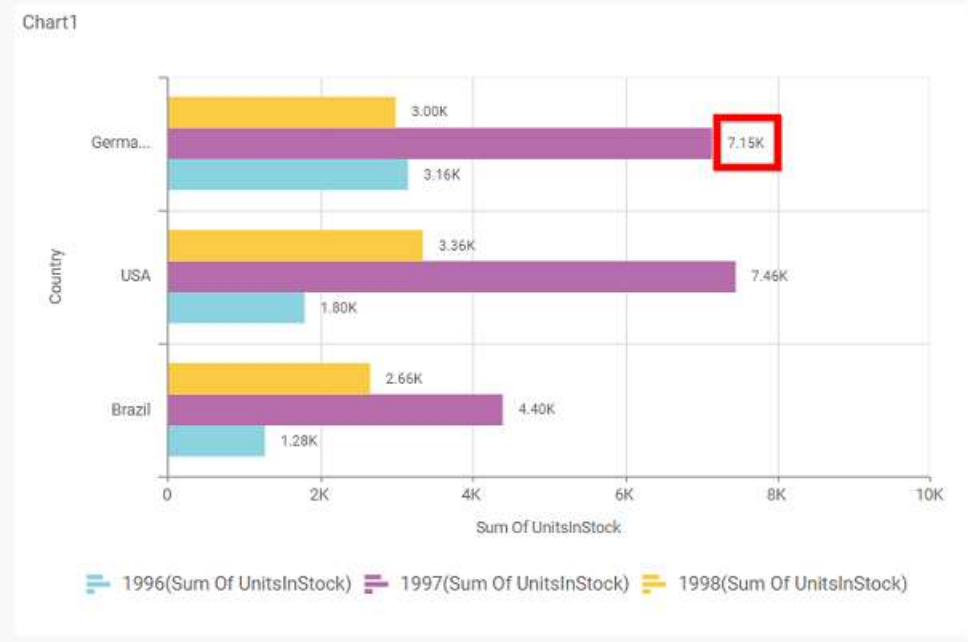


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Austria (Sum of Freight)



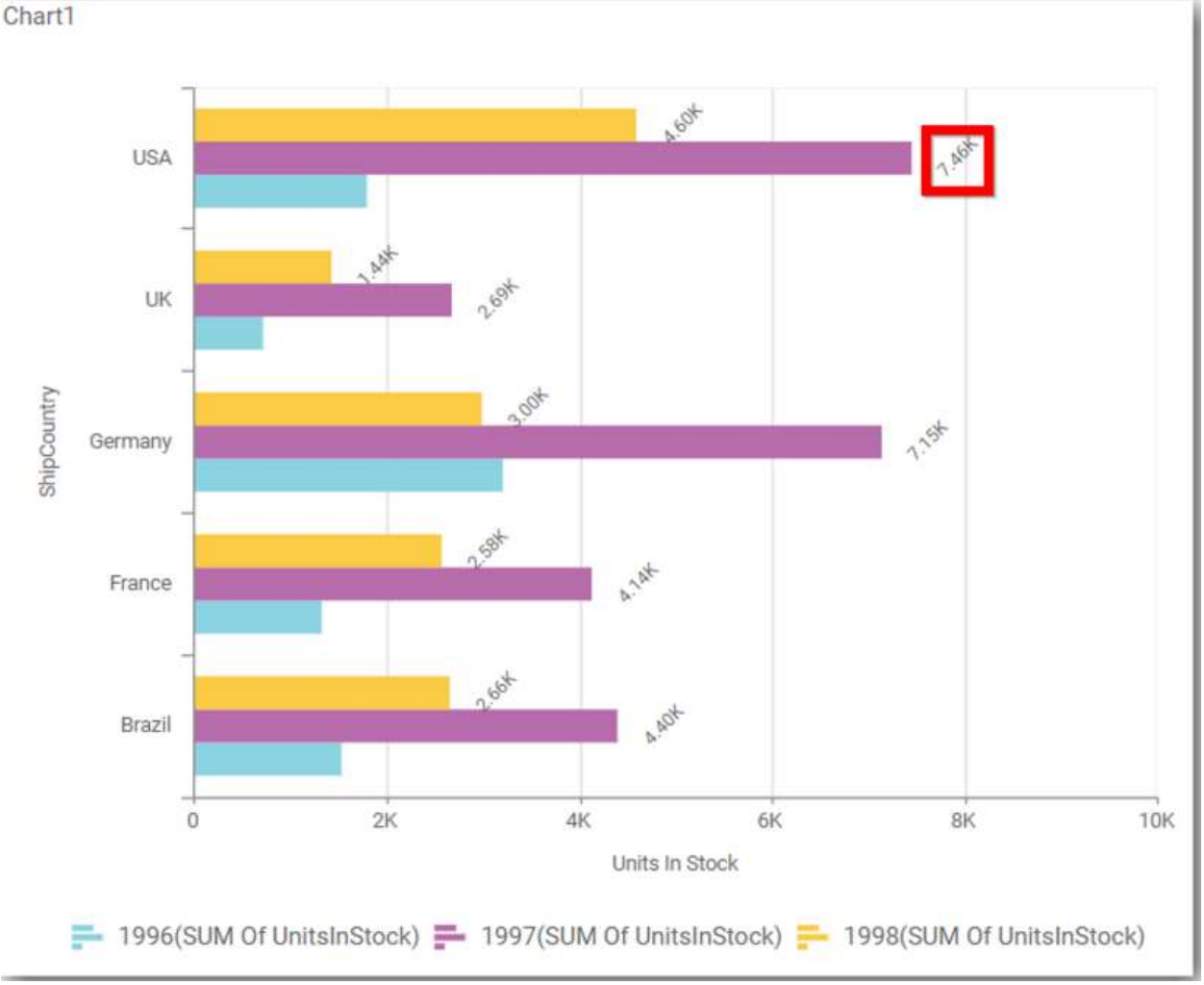
Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

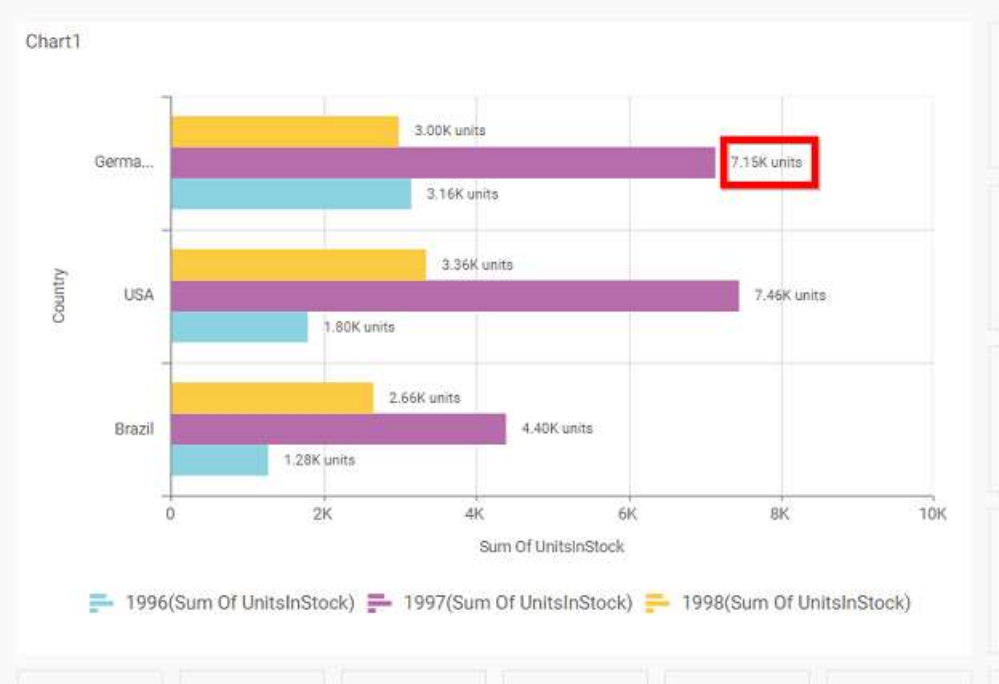


Value Label Suffix

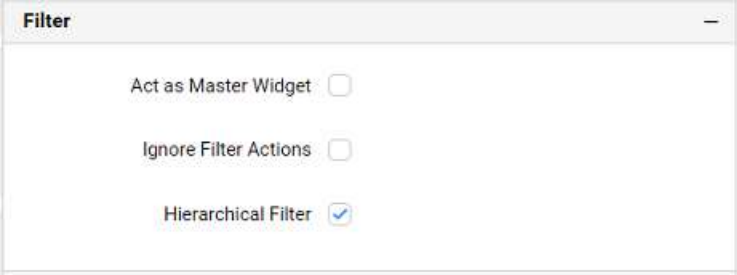
Allows you to enable the Suffix value text to the value labels.

Suffix Value

Allows you to set\edit suffix value to the value labels.



Filter



Act as Master Widget

This allows you to define this bar chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

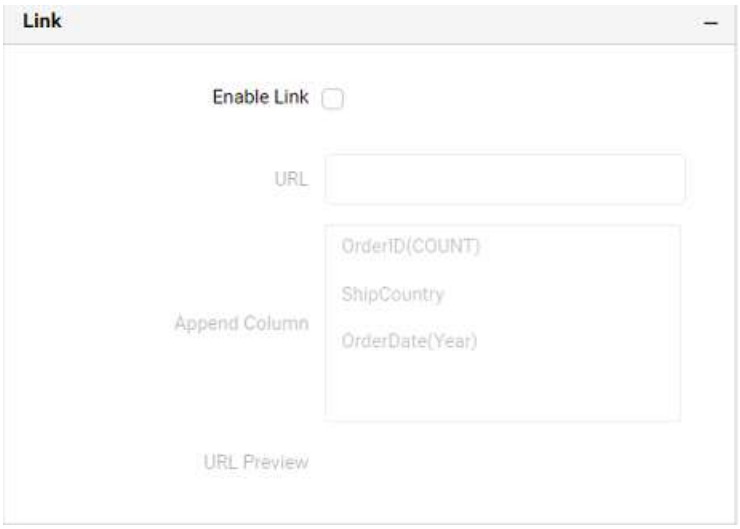
This allows you to define this bar chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

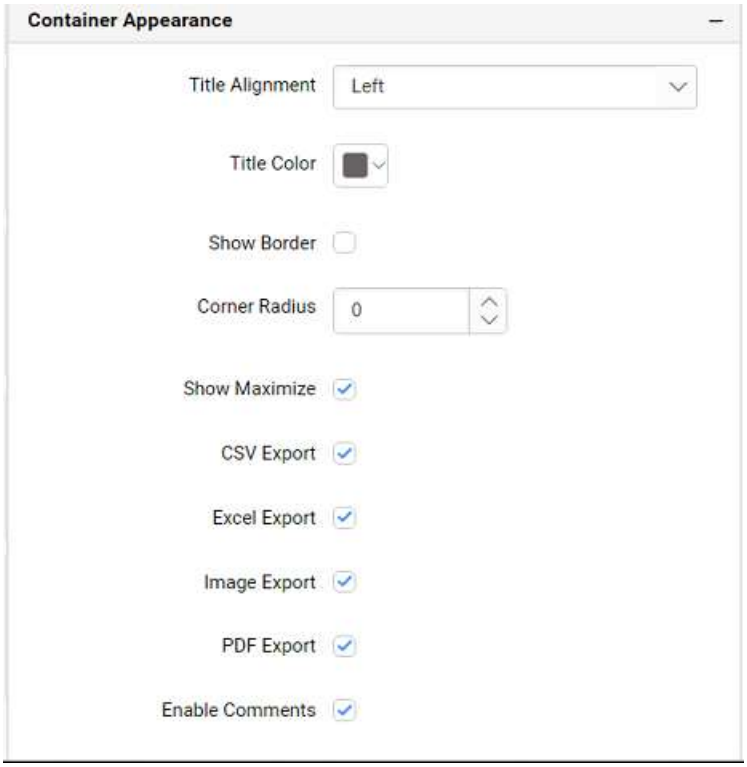
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you the handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this bar chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this bar chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this bar chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

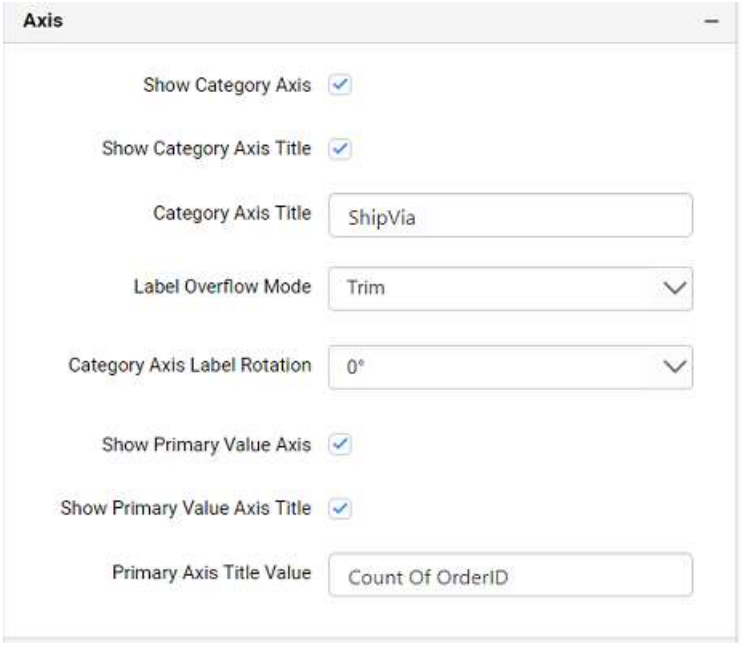
Image Export

This allows you to enable/disable the image export option for this bar chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

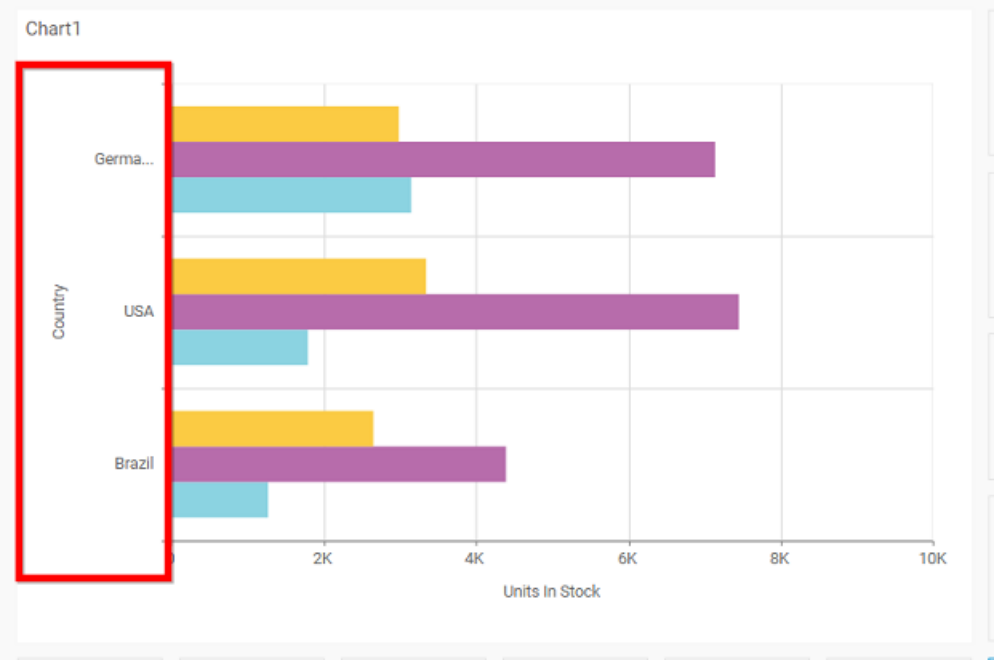
Axis



This section allows you to customize the axis settings in chart.

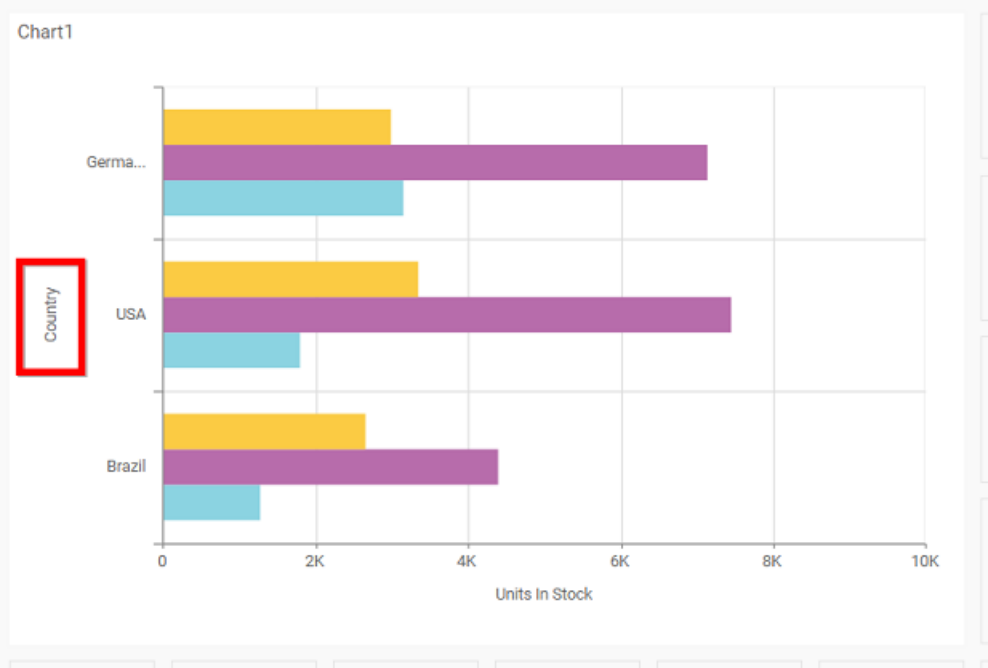
Show Category Axis

This allows to enable the visibility of Category Axis.



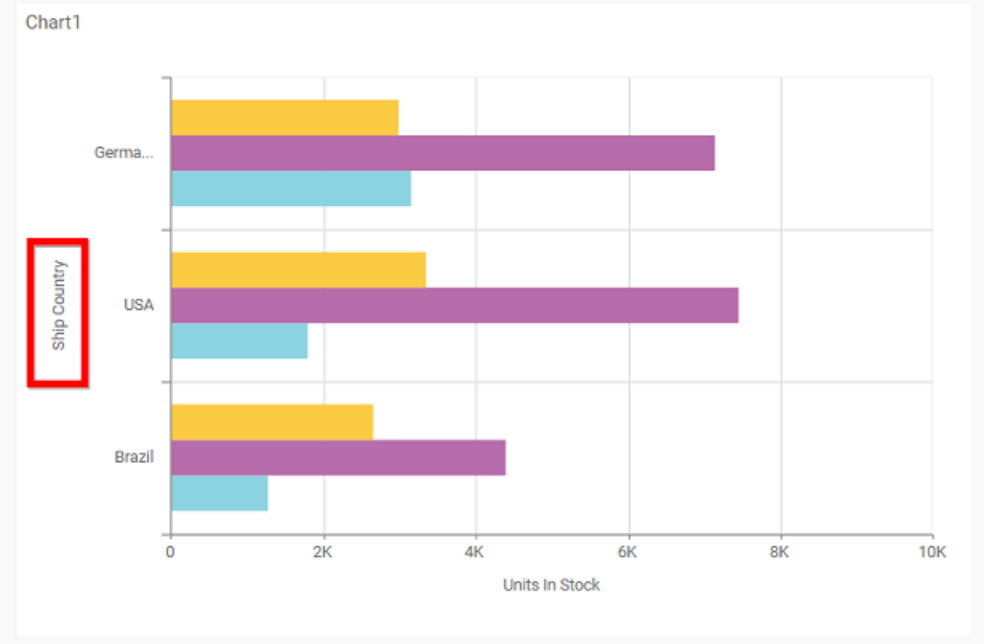
Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

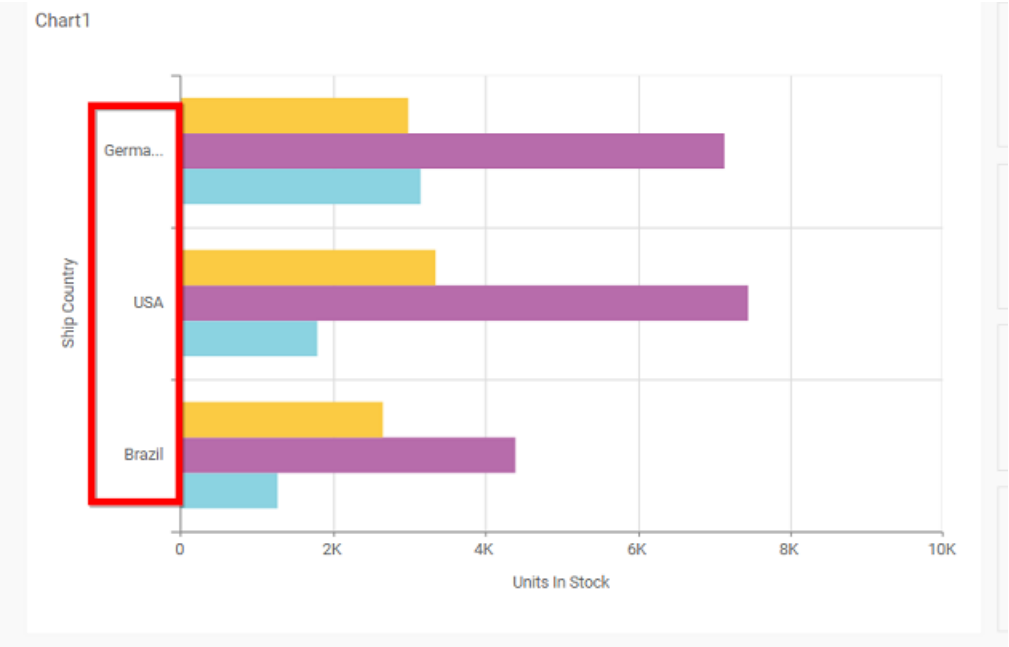


Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

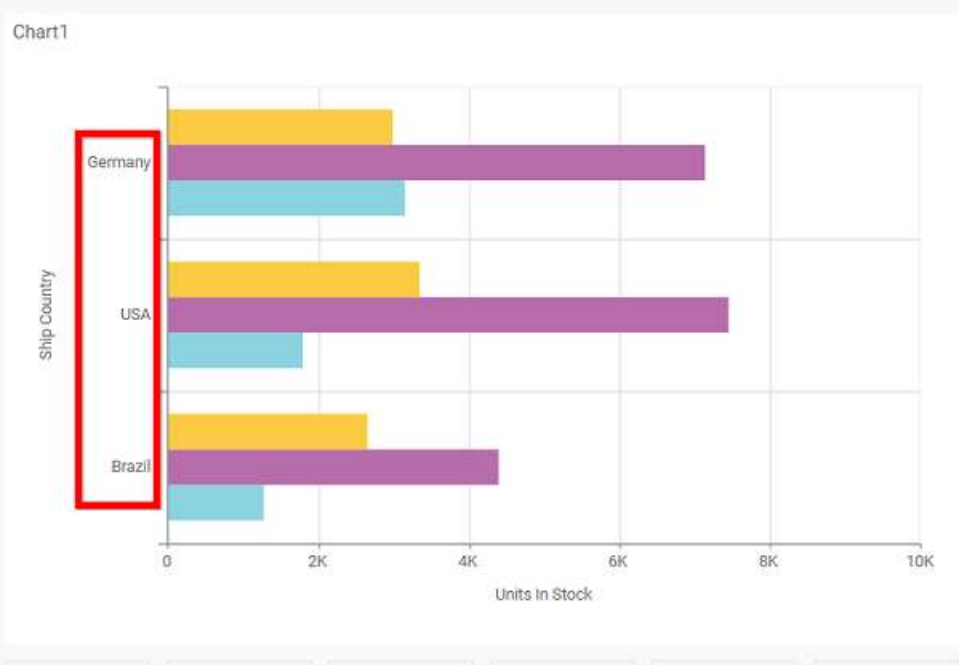
Trim

This option trims the end of overlapping label in the axis.



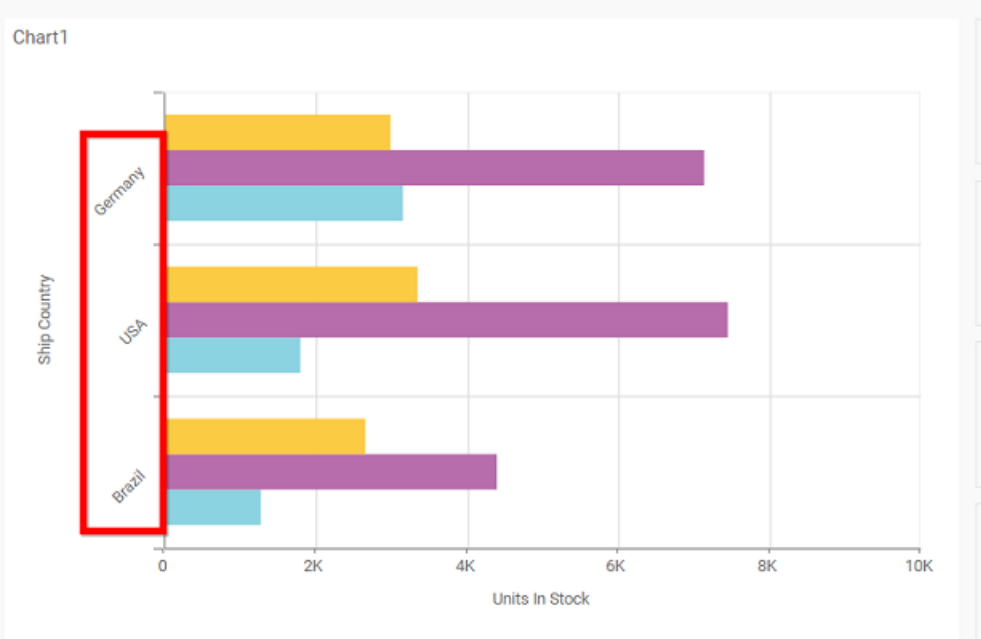
Hide

This option hides the overlapping label in the axis.



Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

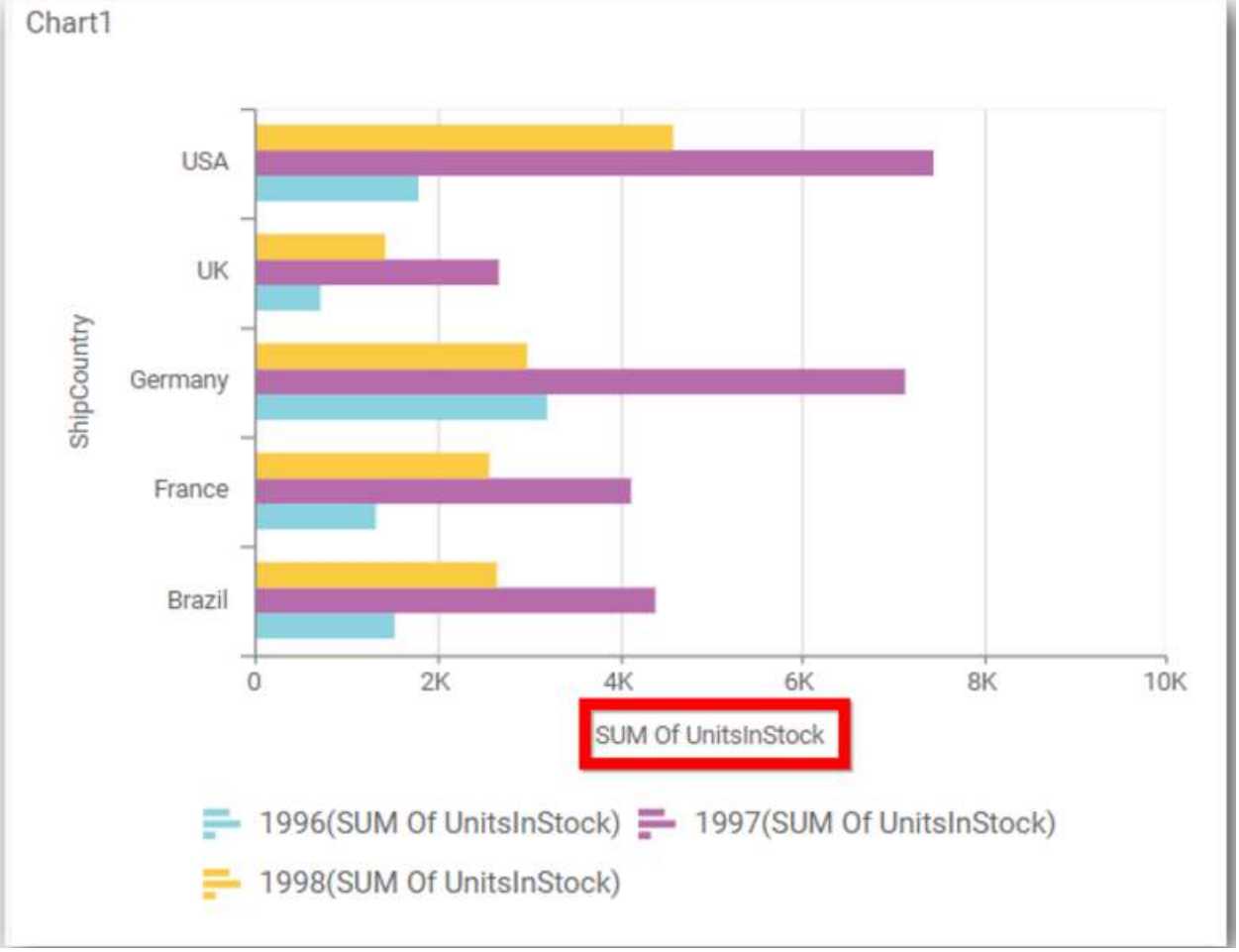


Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.

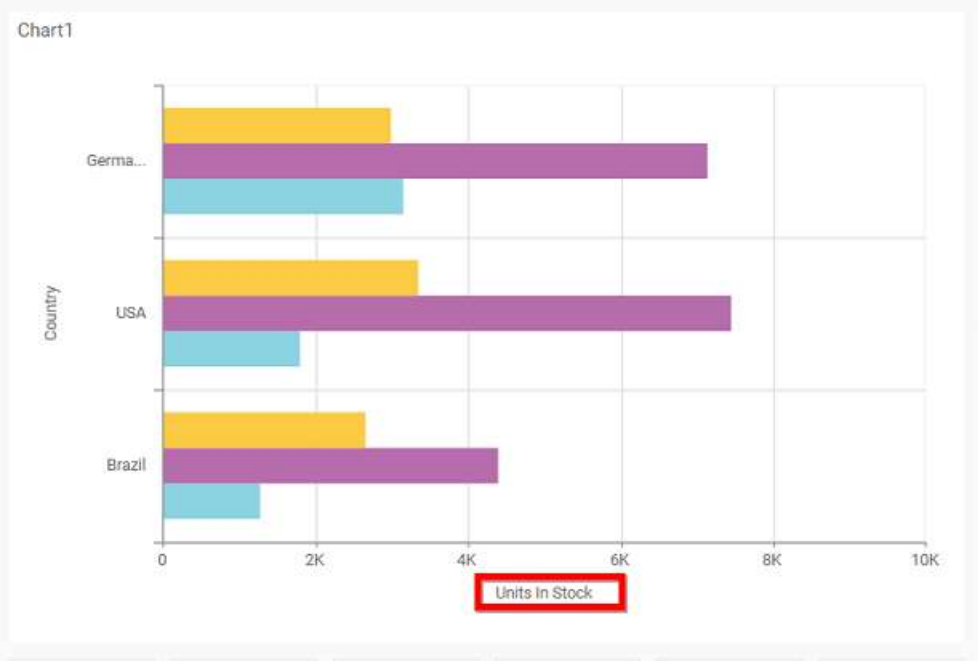
Show Primary Value Axis Title

This allows you to enable the visibility of Primary Value Axis title of chart.

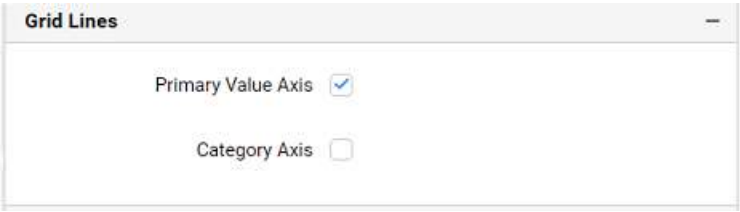


Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

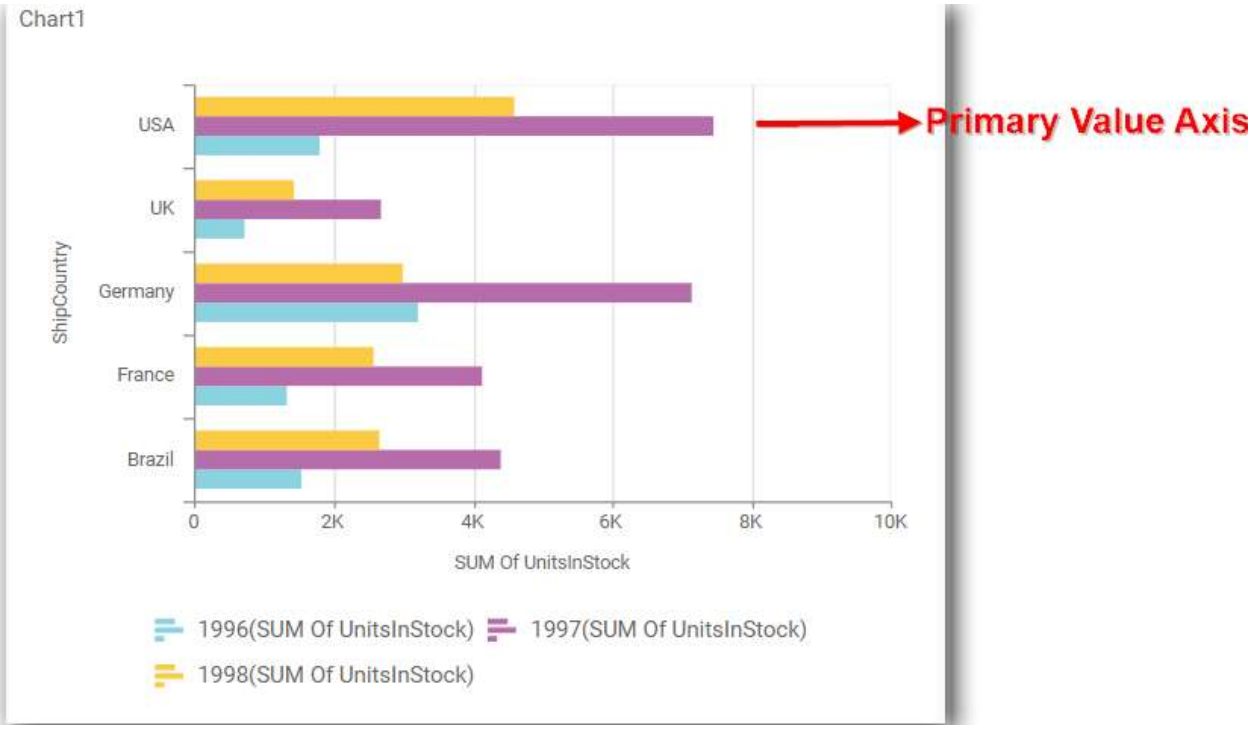


Grid Lines



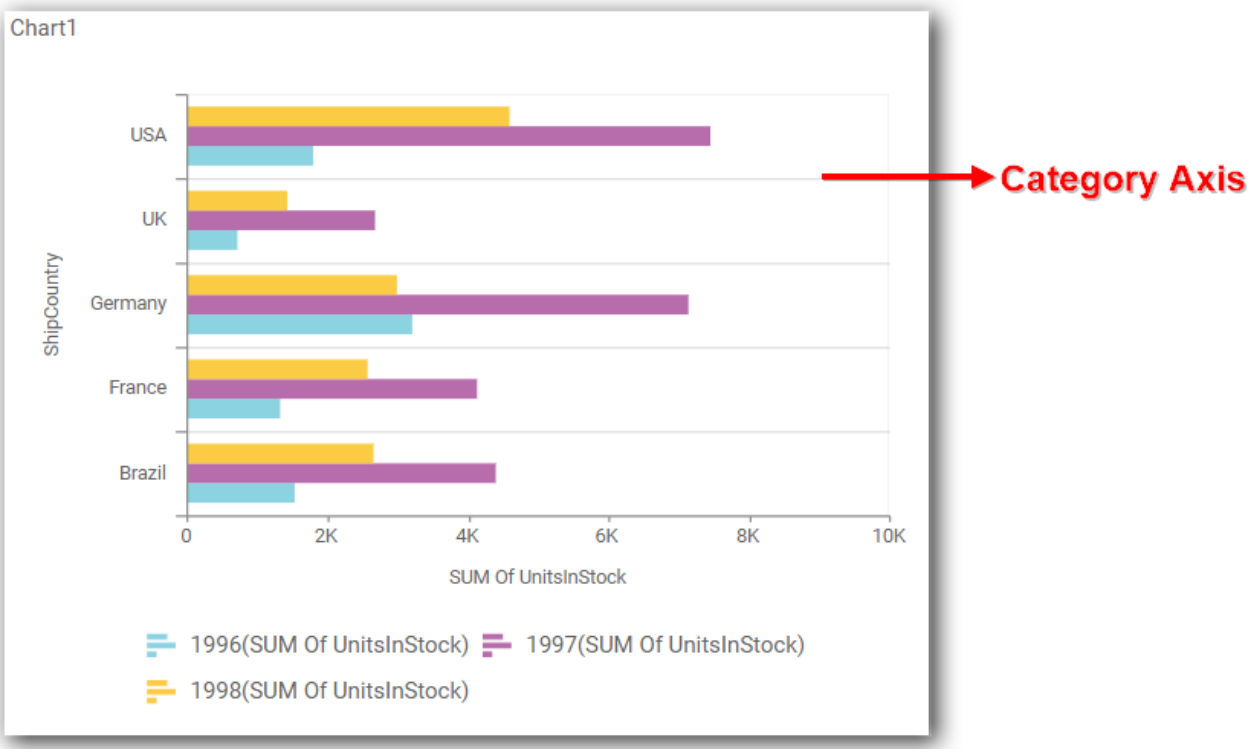
Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the bar chart.



Category Axis

This allows you to enable the Category Axis gridlines for the bar chart.

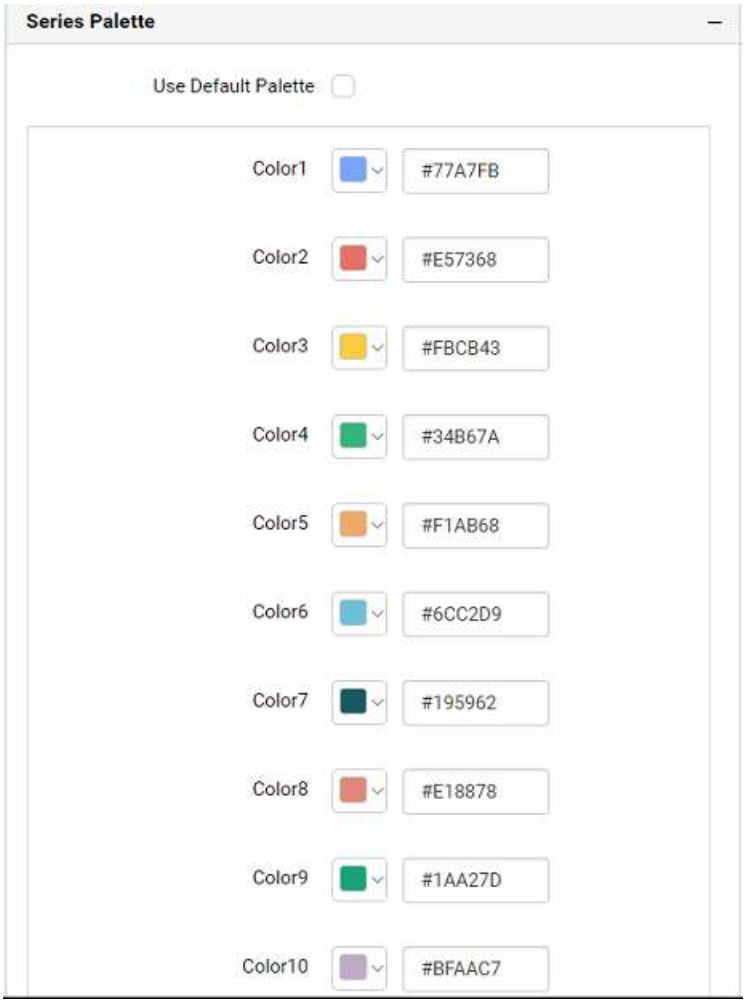


Series Palette

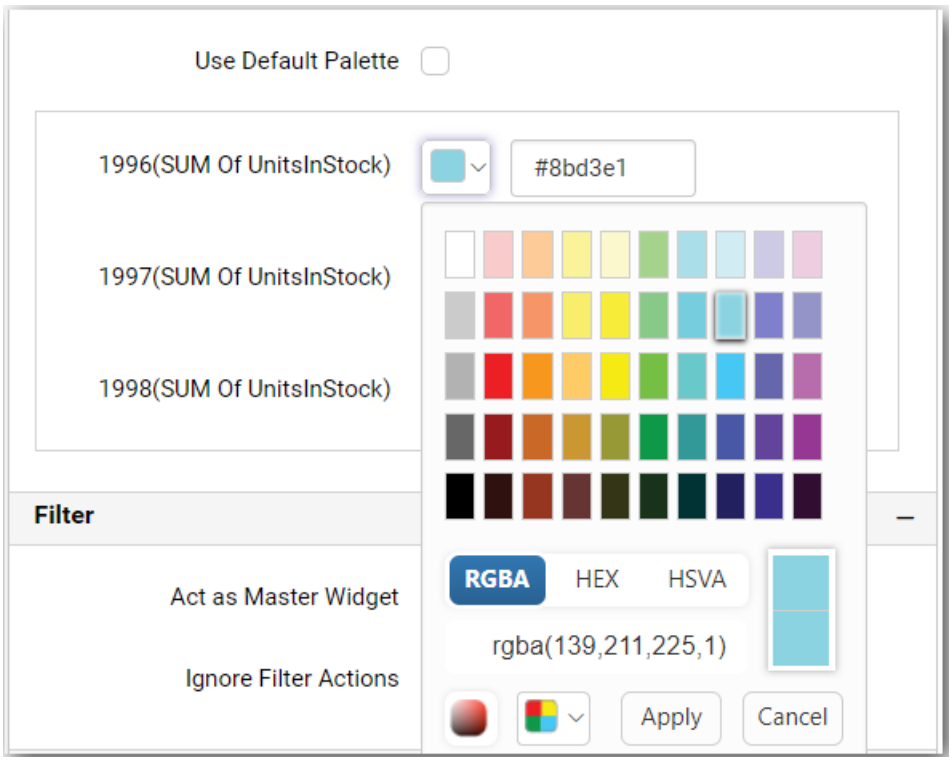
This allows you to customize the chart series color through Series Palette section.

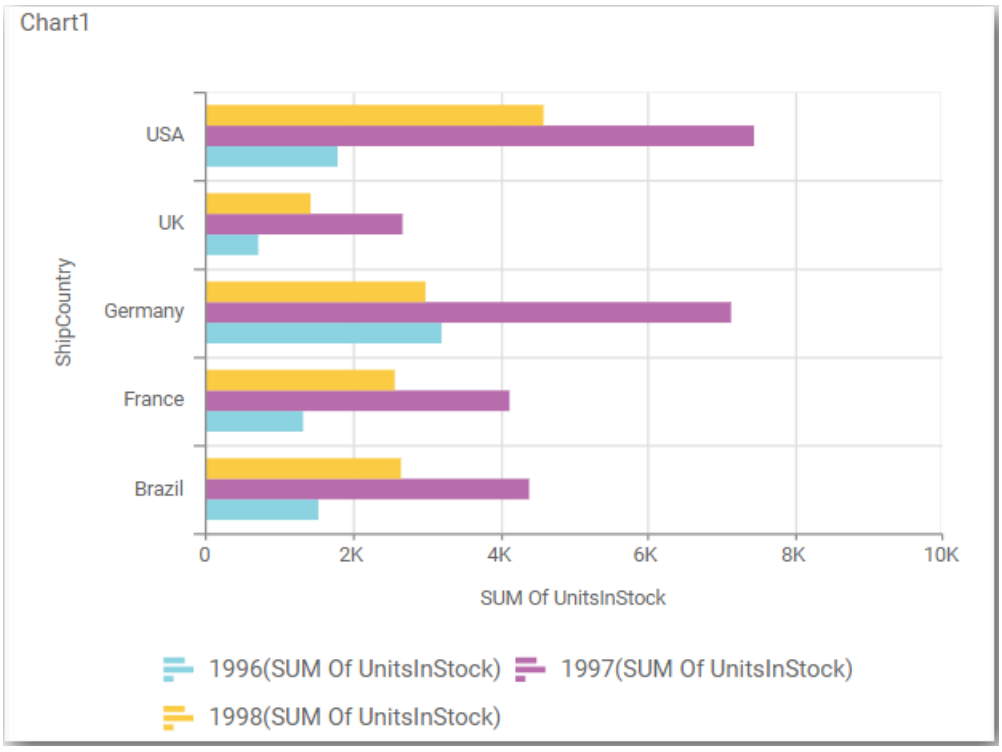
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





How to apply conditional formatting?

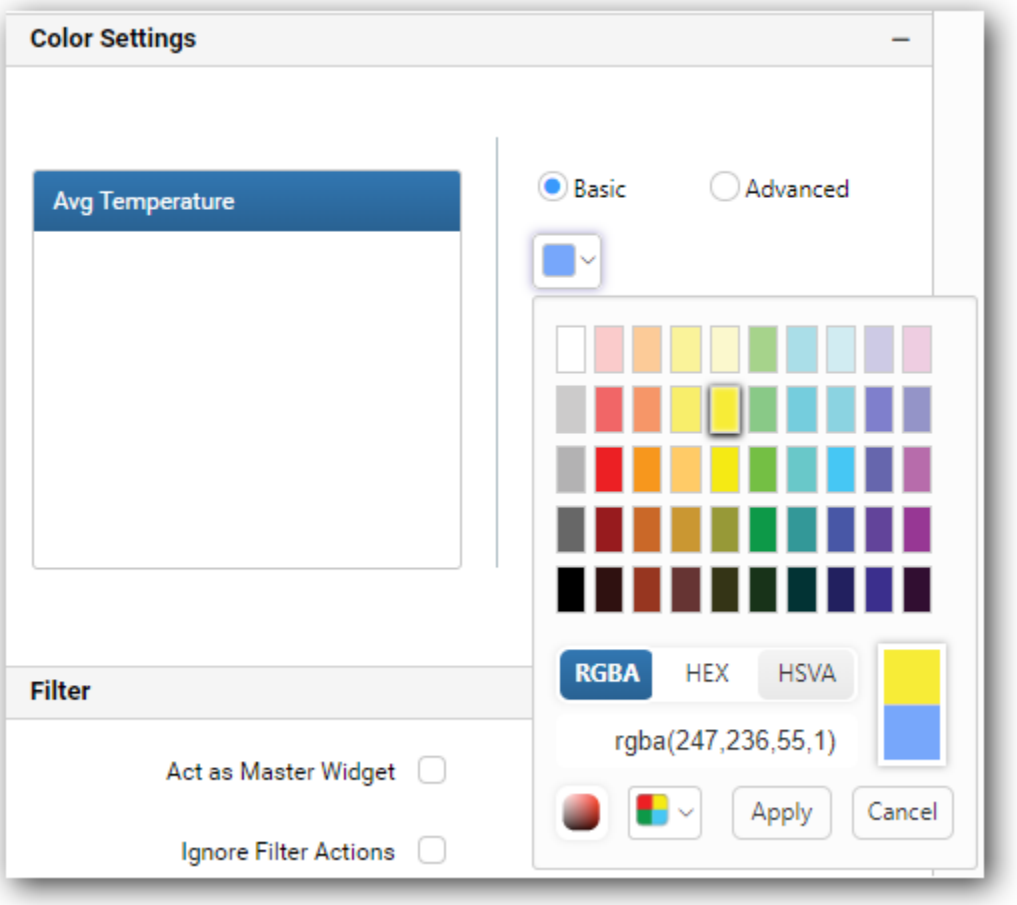
Color in bar chart widget can be customized using the color settings which is available in the properties section. This will allow the user to improve the visualization in bar chart and to distinguish the data based on conditional range values that will let the visualizer to understand what is shown in data.

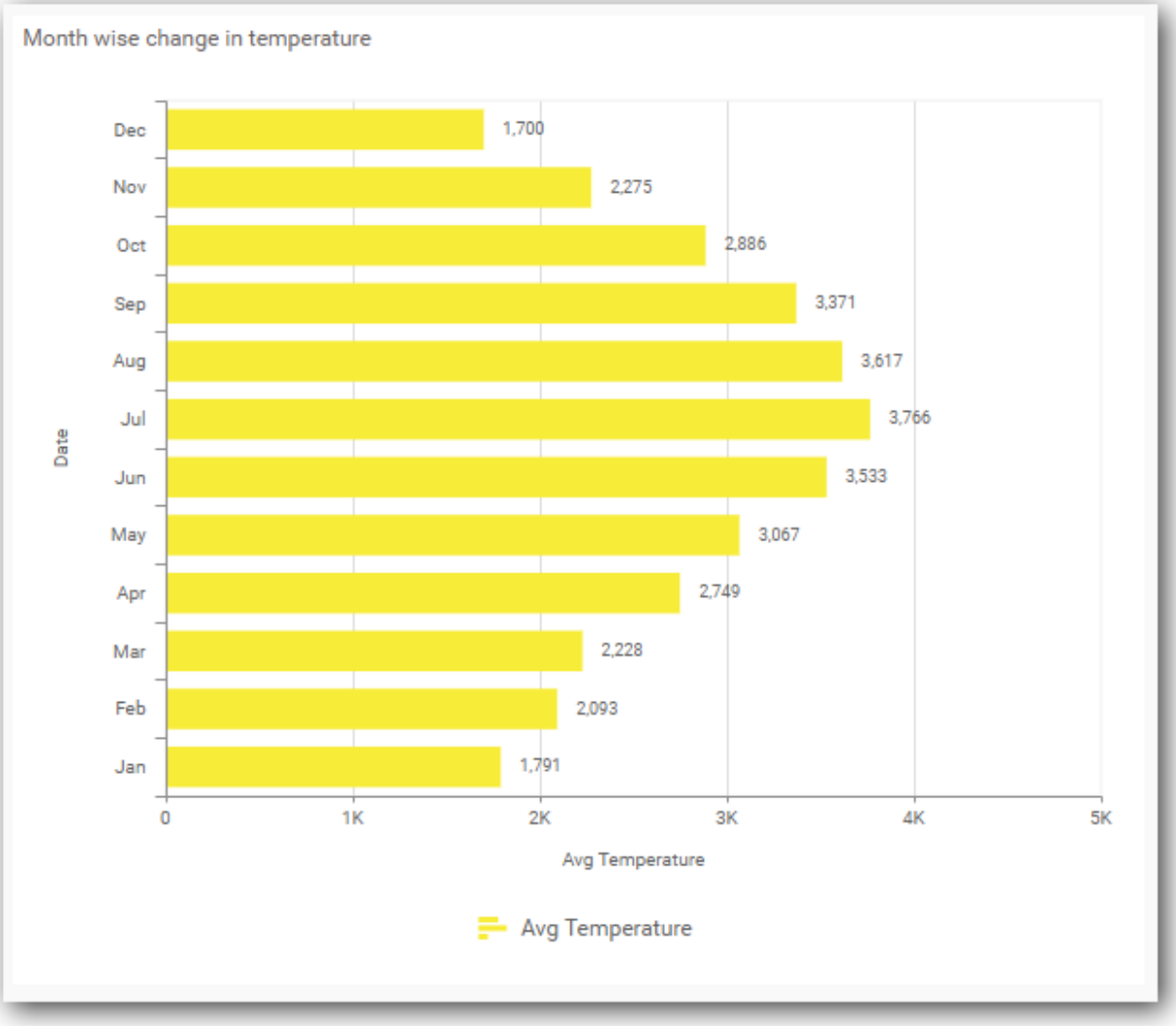
Color Settings

This allows you to customize chart series color based on user provided conditions.

Basic Settings

This section allows you to customize the series color with series label on the left-hand side and corresponding series color on the right-hand side. You can choose a color. And, you can also change the series color by changing the corresponding color value in the right-hand side.

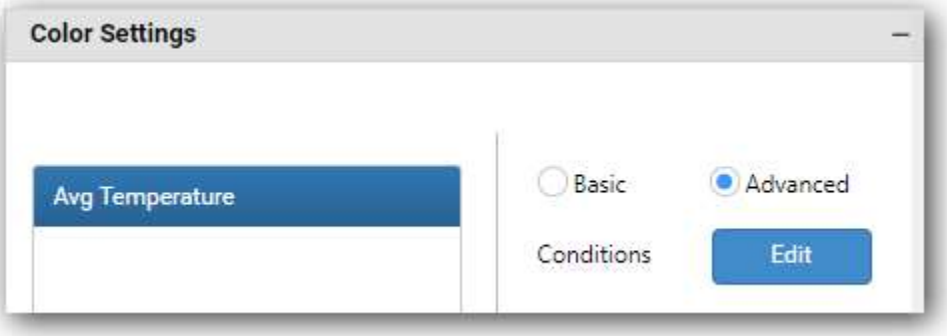


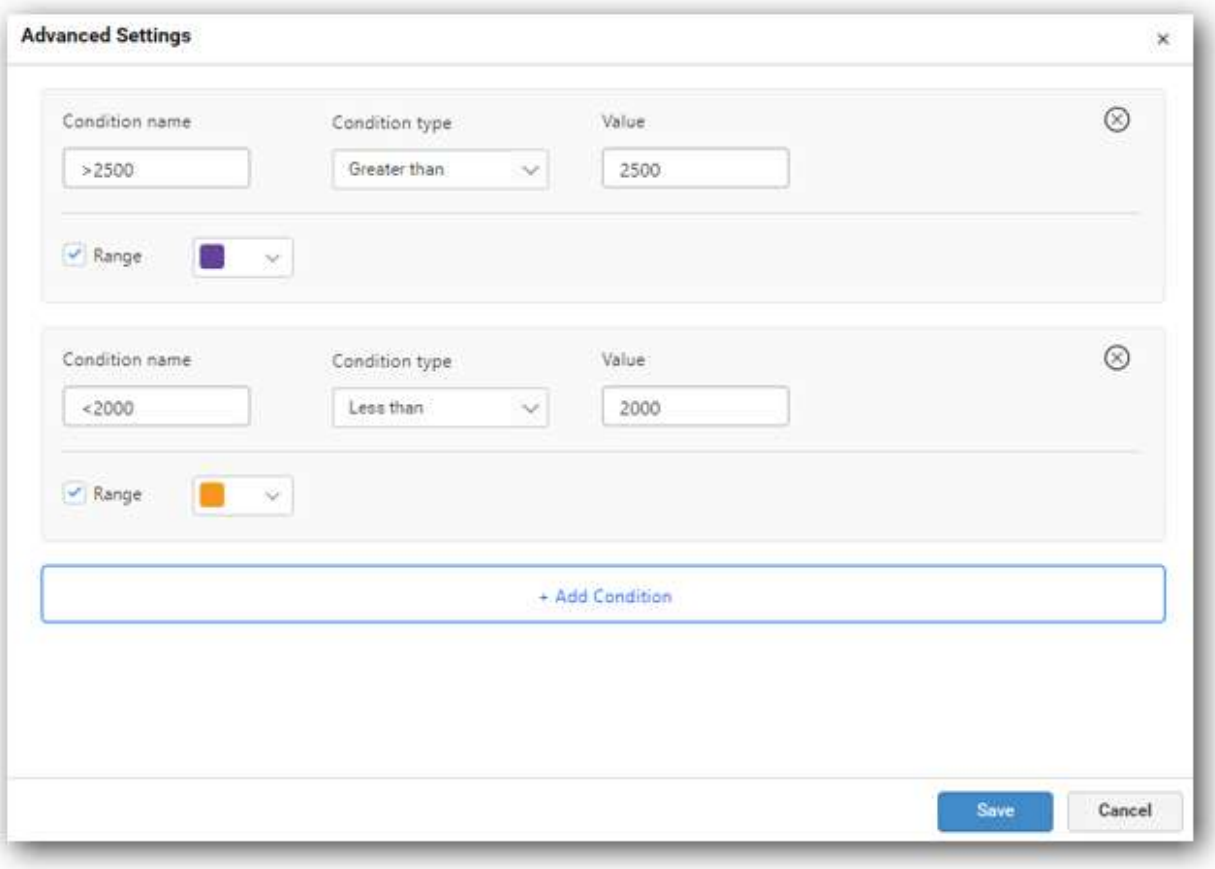


Advanced Settings

This allows you to provide colors for the selected measure based on single or multiple conditions available.

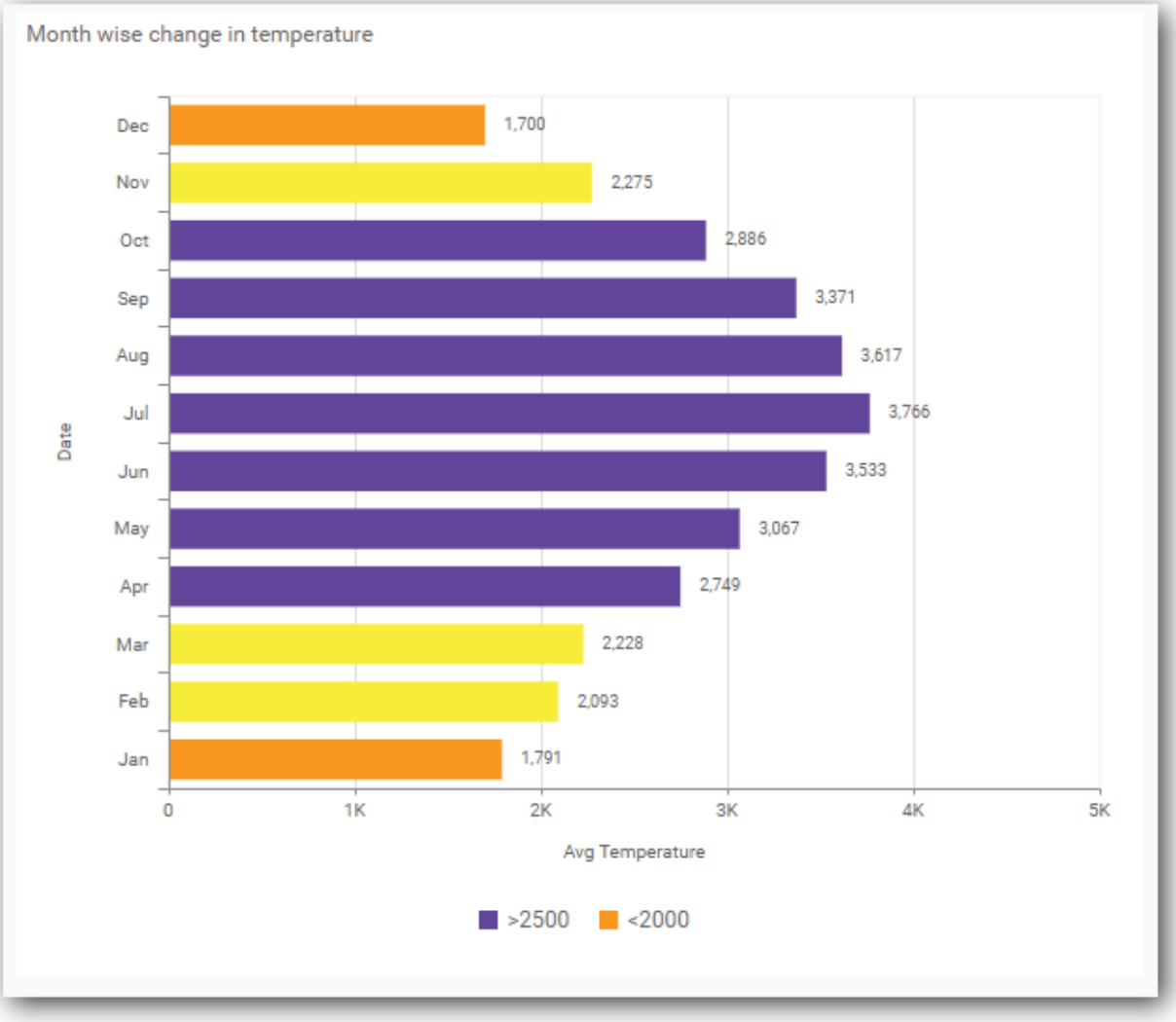
You can choose a series color using multiple condition sets such as Greater than, Less than, Equal to, Not Equal to, Between, Not between, Greater than or equal to, Less than or equal to.





The chart will render with series color based on the provided conditions and the series which is outside the applied conditional range, will have basic color applied.

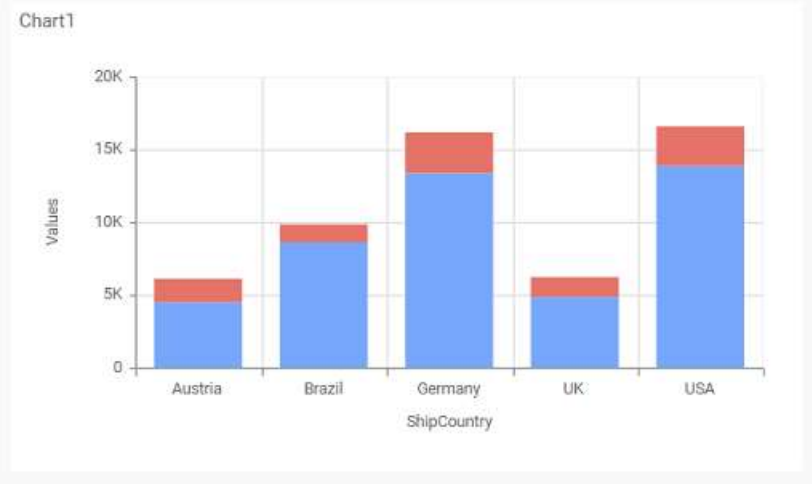
You can also customize the legend name by providing necessary values inside condition name.



Note: Advanced setting will be available only for chart with single measure value.

Stacked Column Chart

Stacked Column Chart allows you to compare multiple measures through bars stacked one after the other vertically.



How to configure the table data to stacked column chart?

Stacked Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

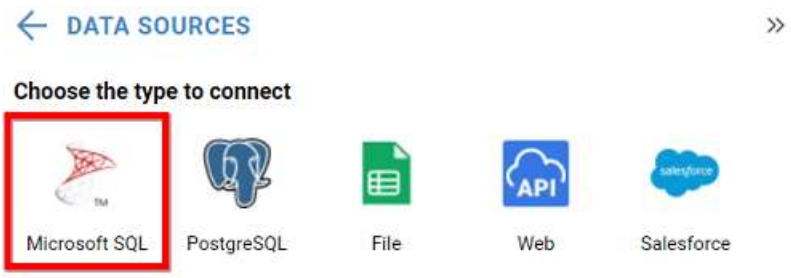
Follow the steps configure data to stacked column chart

Drag and drop the stacked column chart to canvas and resize it to your required size.

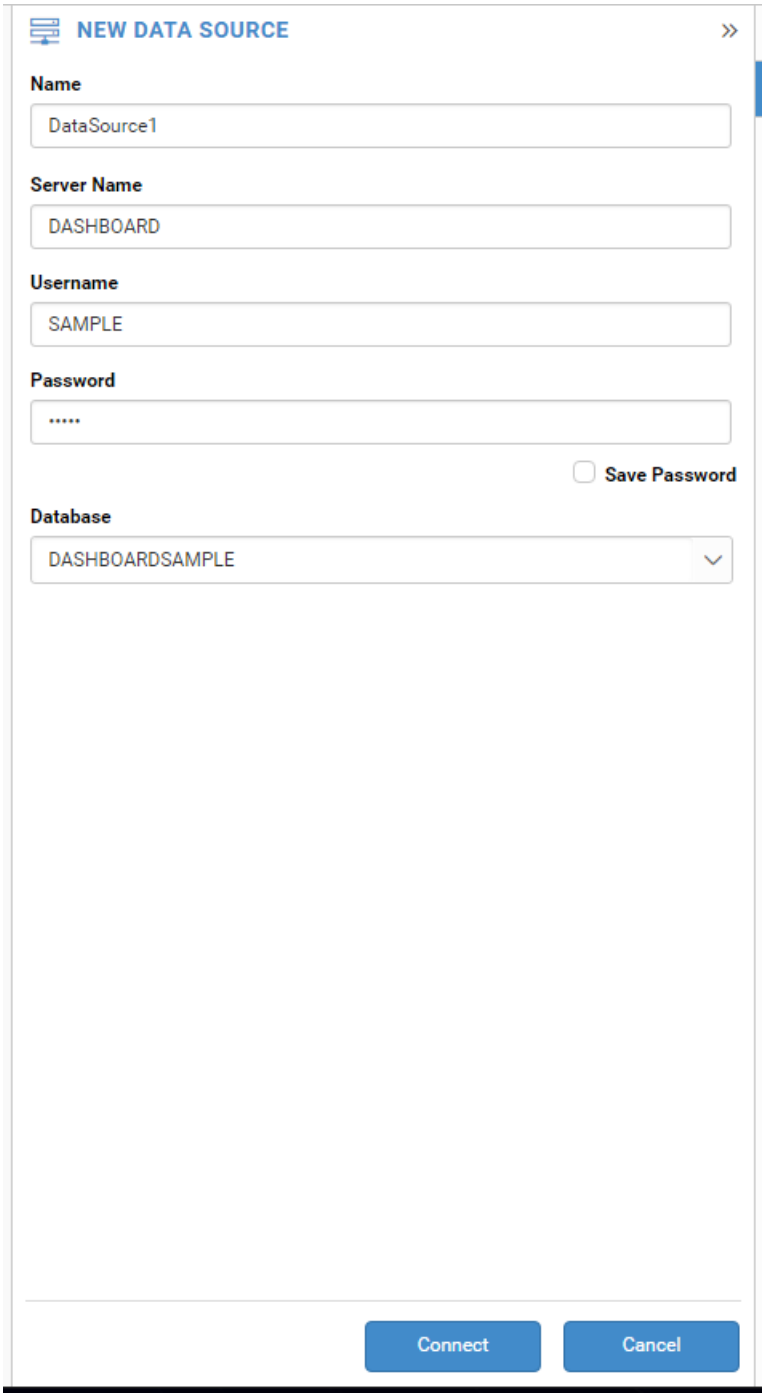
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

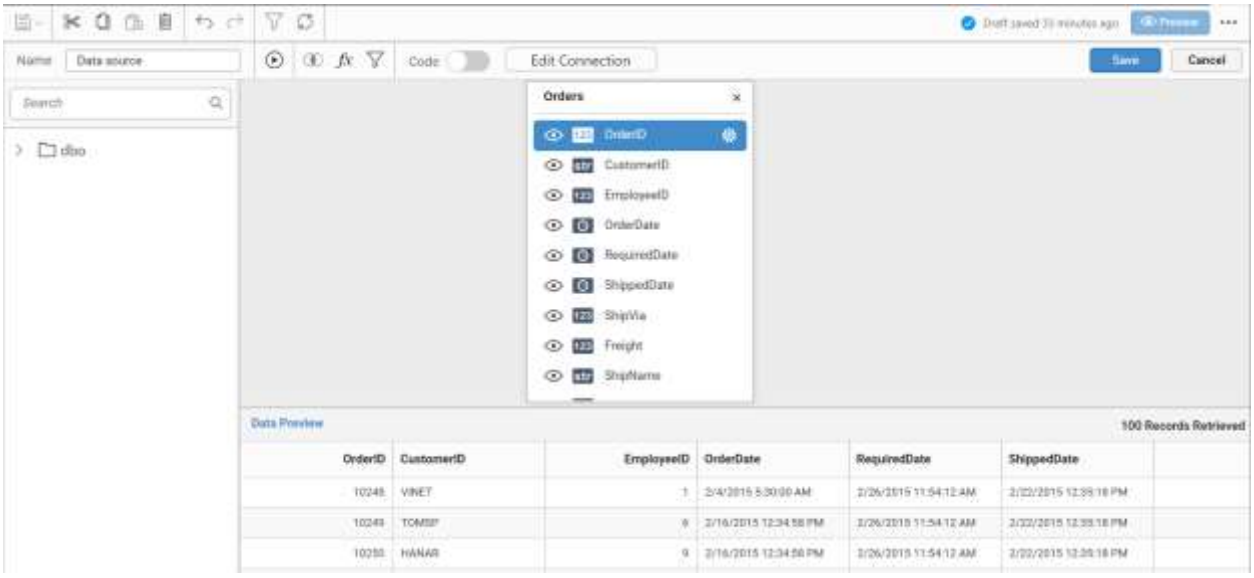
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



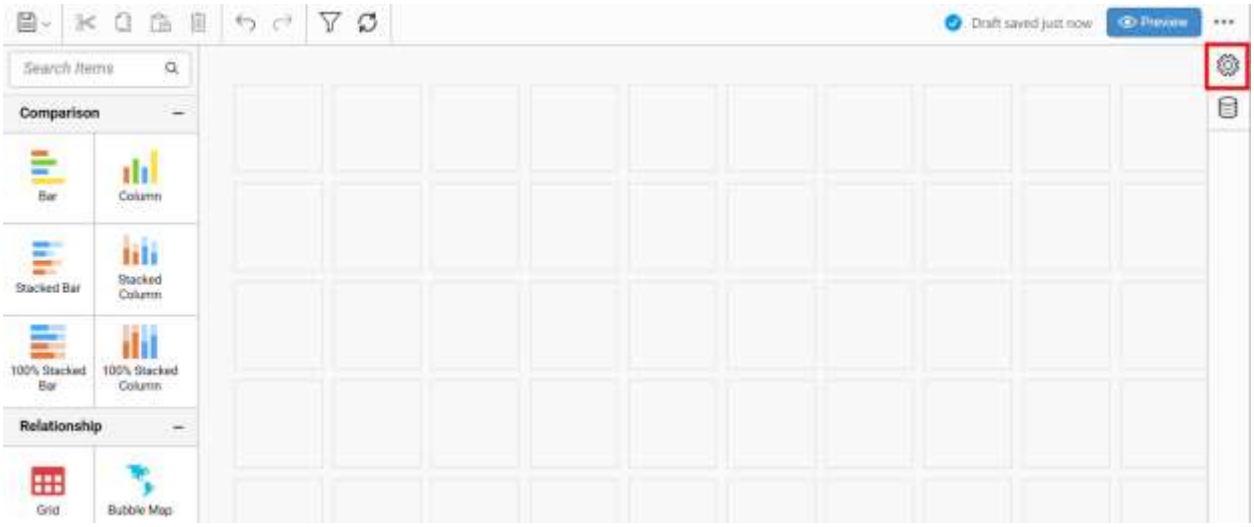
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

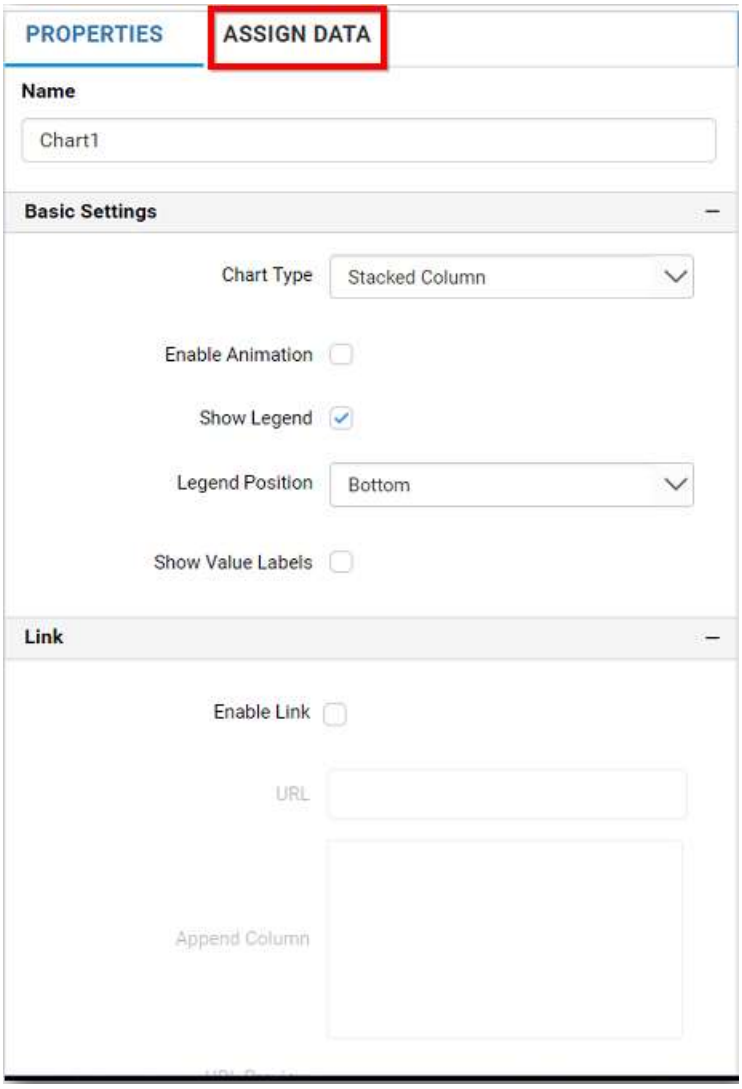


Drag your preferred table or view from the left pane from data design view, click Save button.

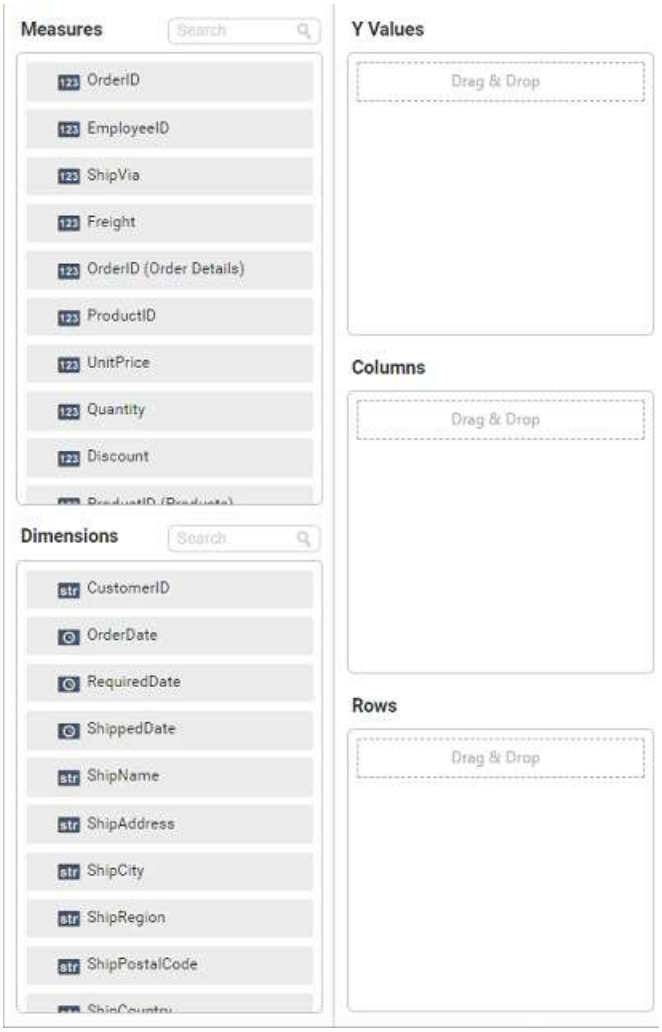


Click Properties button in configuration panel, property pane opens. Now, switch to ASSIGN DATA tab.





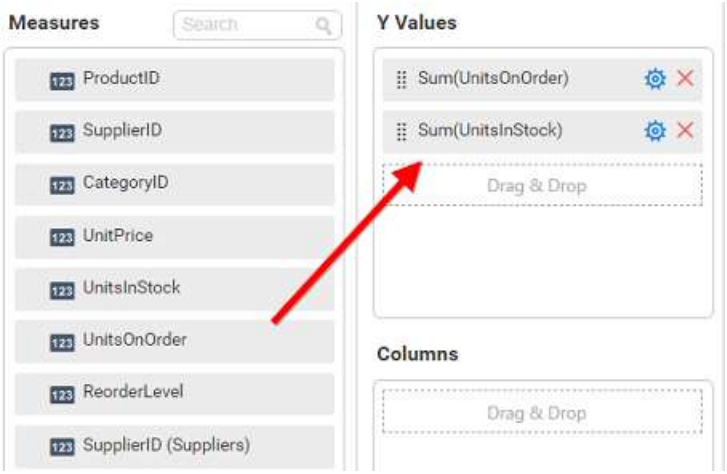
The data tab will be opened with available measures and dimensions from the connected data source



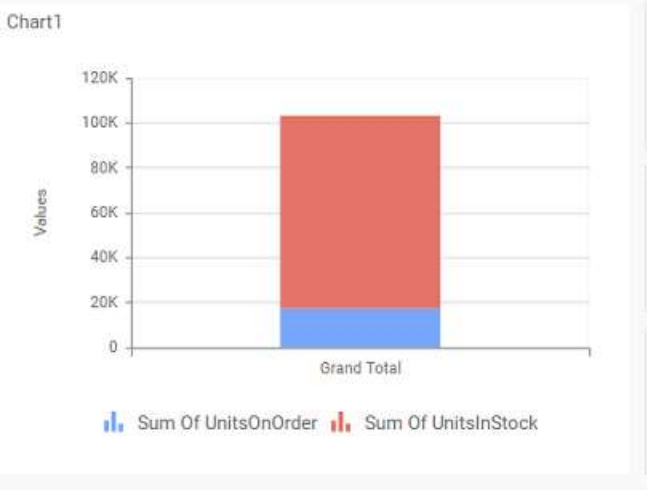
You can add the required data from Measures and Dimensions into required field.

Adding Y Values

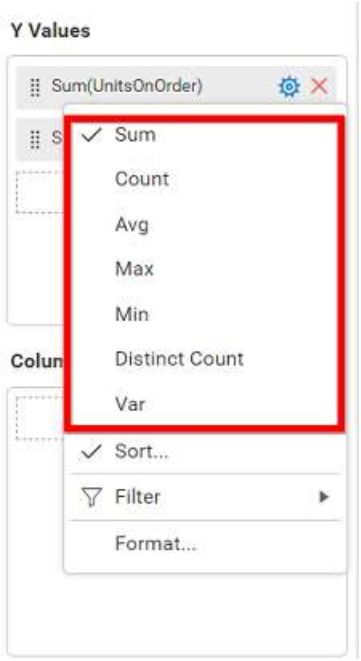
You can add more than one Measures into Y Values field by drag and drop the required measure.



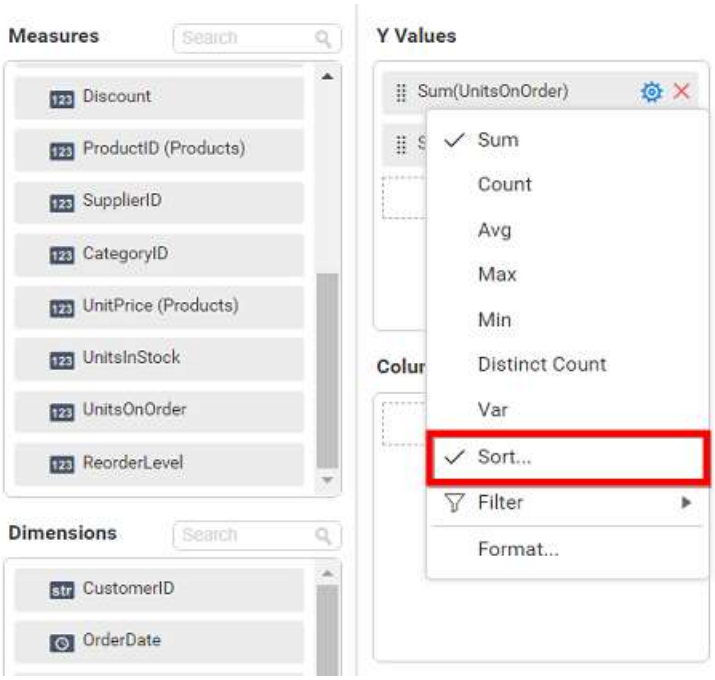
Now the chart will be rendered like this



Click the Settings option to change required summary type from the available summary types shown in Settings.

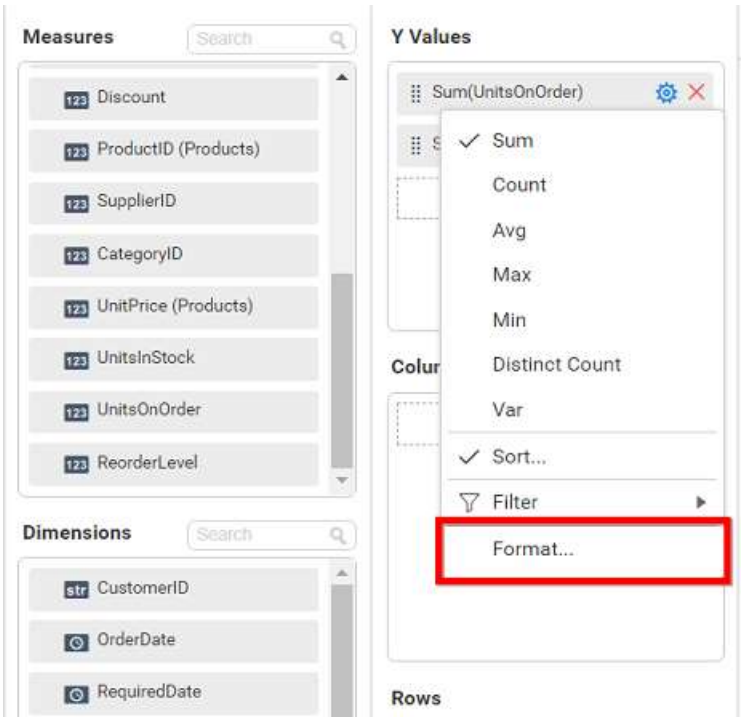


You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort



You can filter the data to be displayed in chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format



To remove the added value fields click highlighted button.

You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.

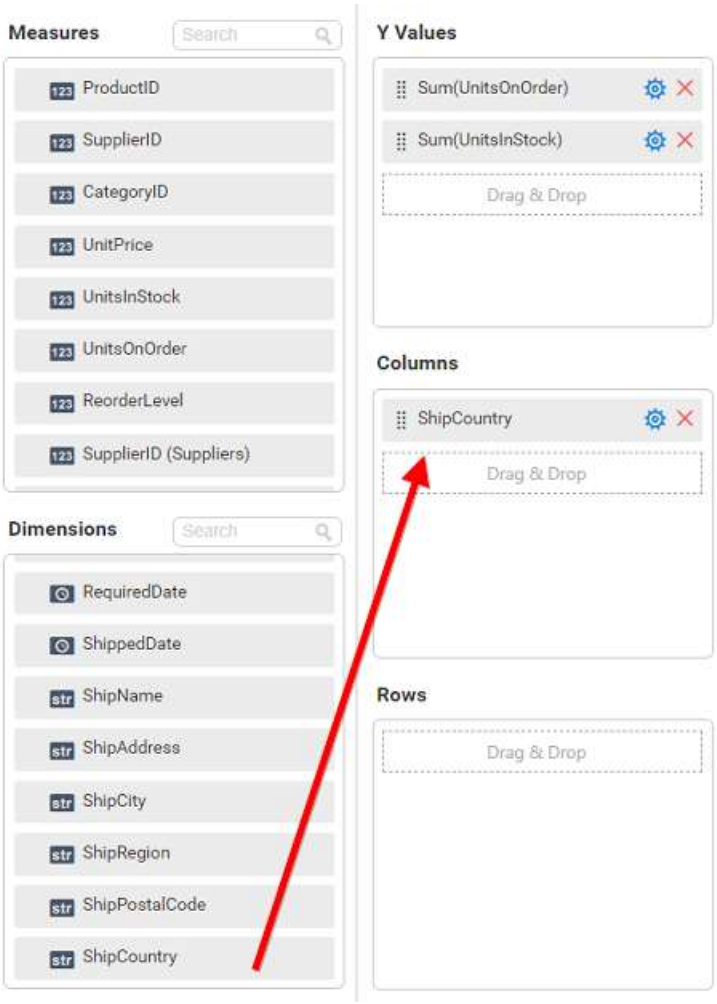
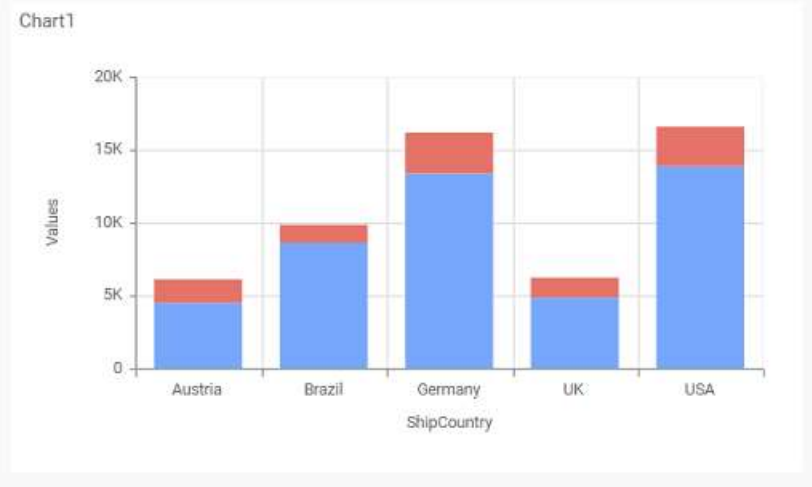


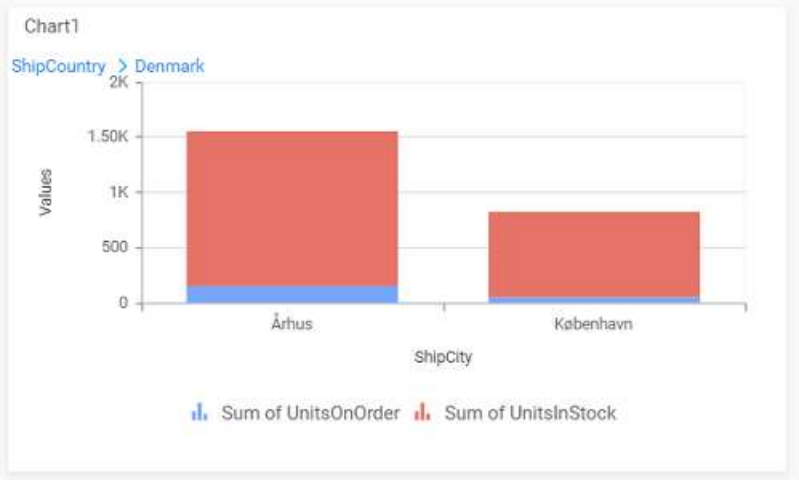
Chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

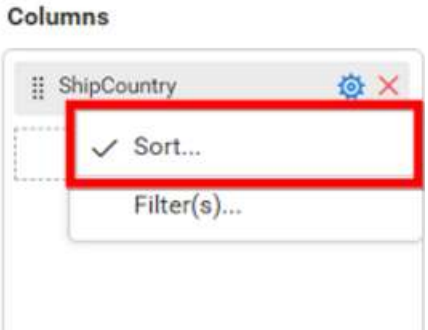
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.

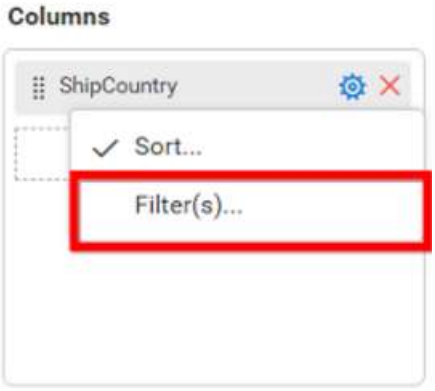


You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

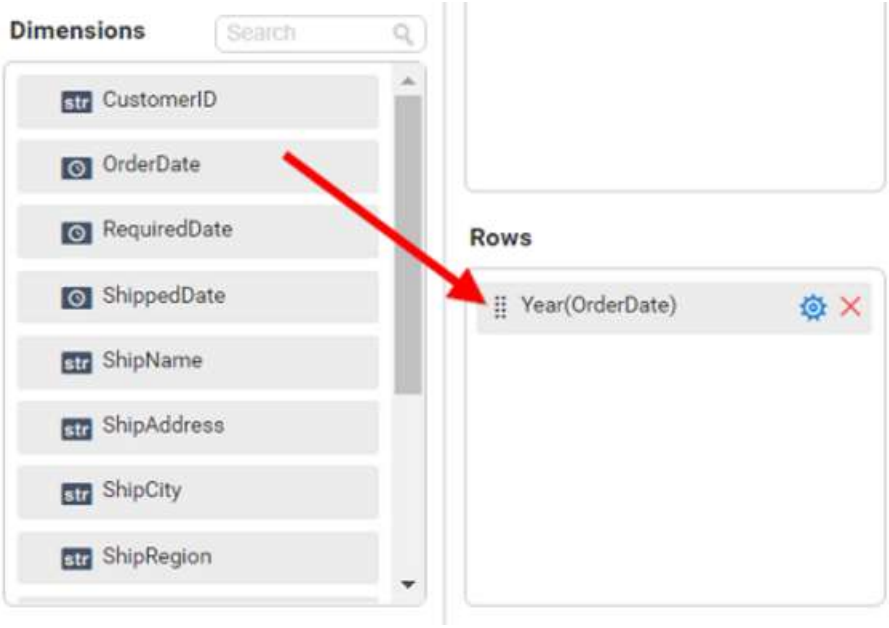


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

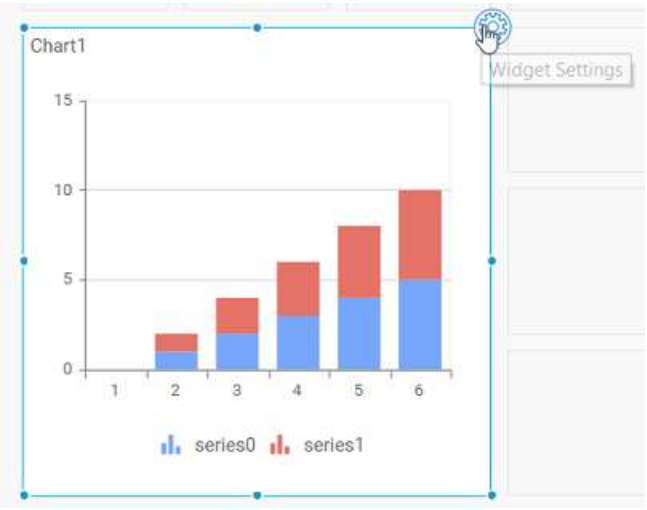
This will render chart in series.

How to format stacked column chart?

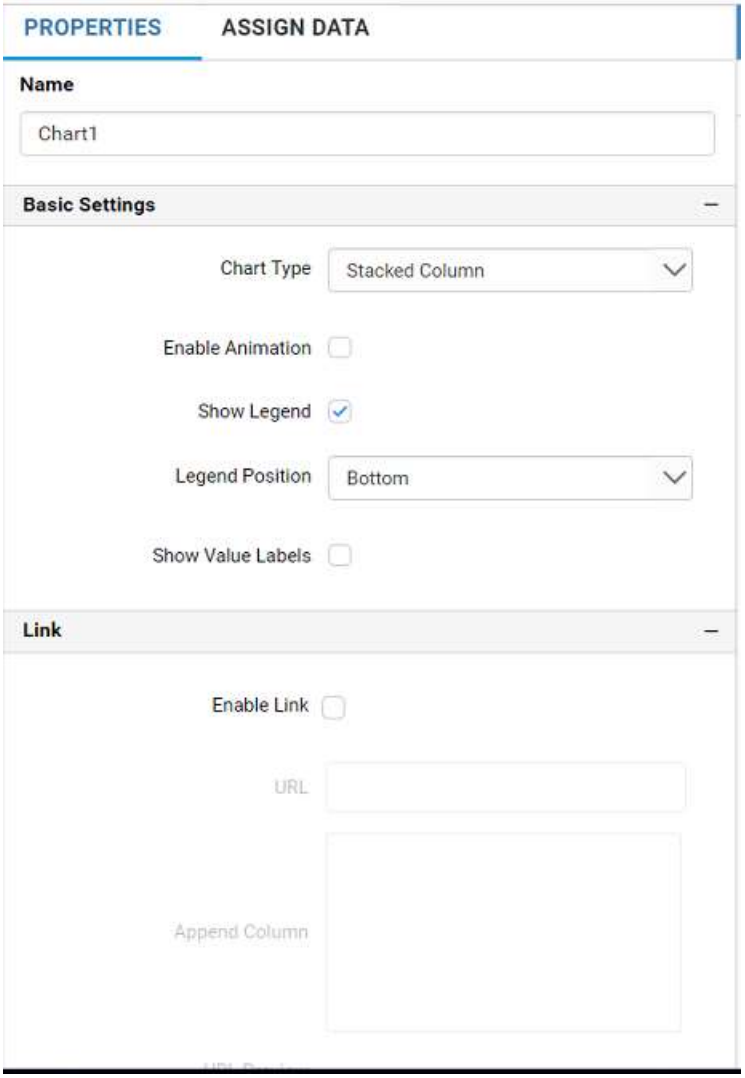
You can format the stacked column chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into stacked column chart follow the steps

1. Drag and drop the stacked column chart into canvas and resize it to your required size.
2. Configure the data into stacked column chart.
3. Focus on the stacked column chart and click on widget settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this stacked column chart widget

Basic Settings

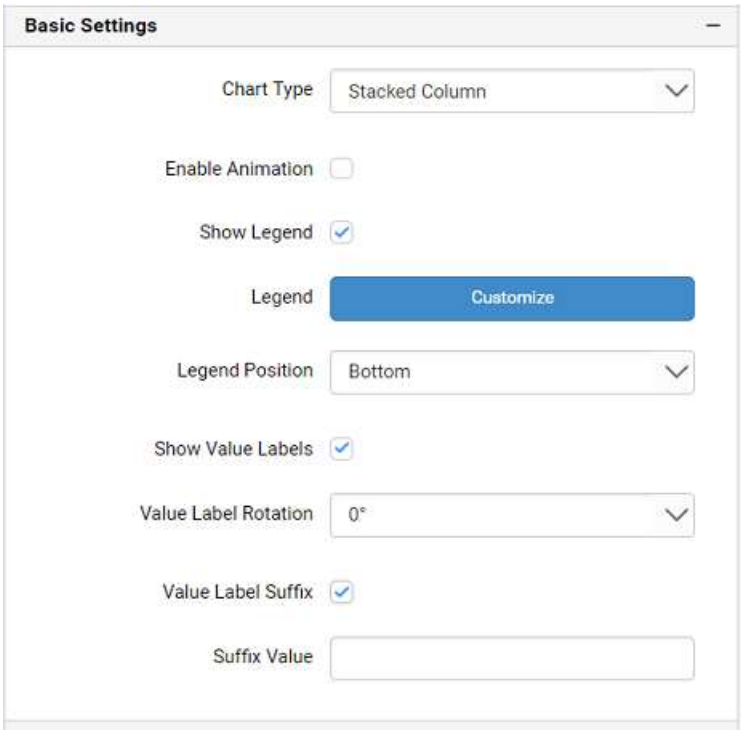


Chart Type

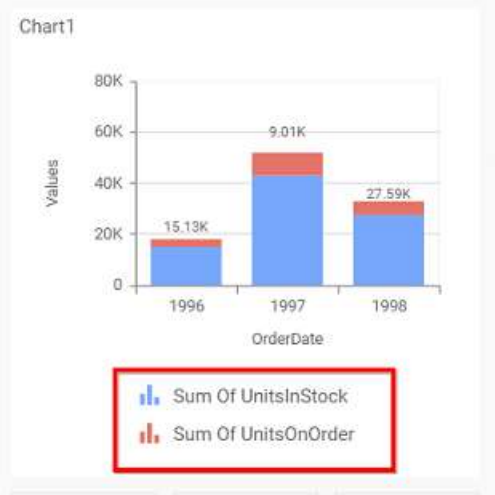
This allows you to switch the widget view from current chart type to another convertible chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

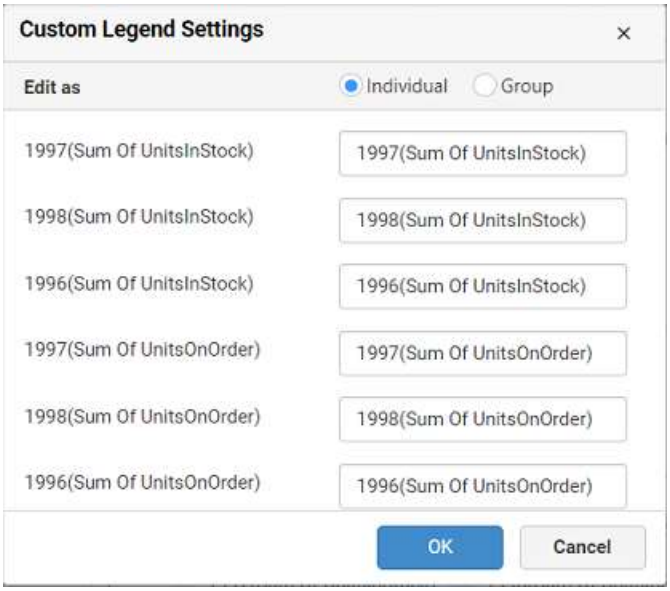
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

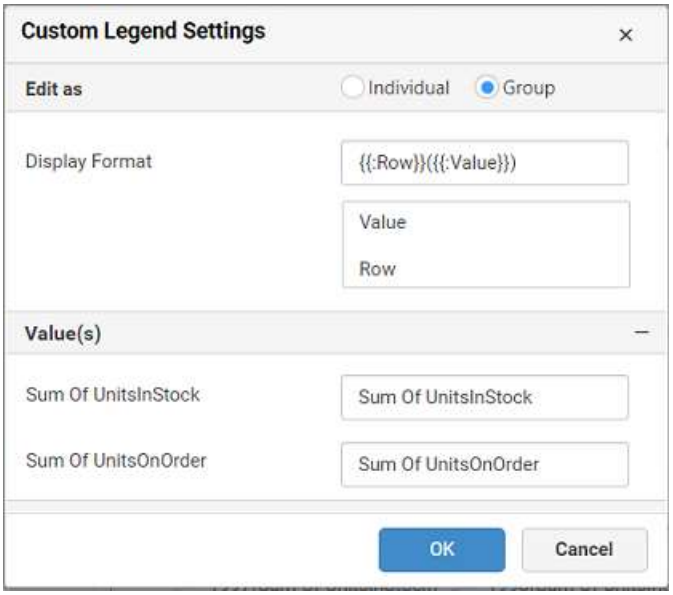
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.

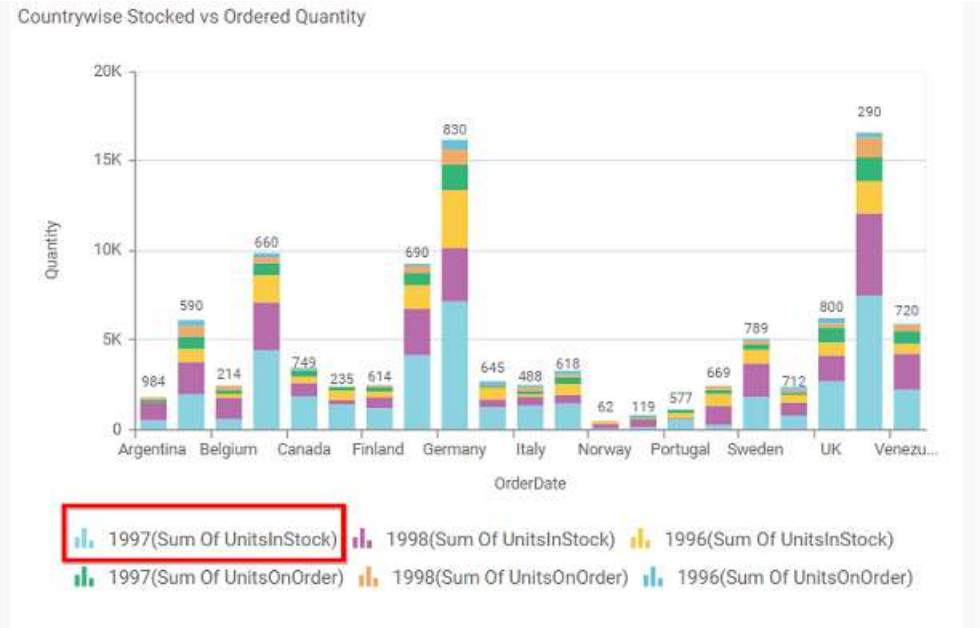


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

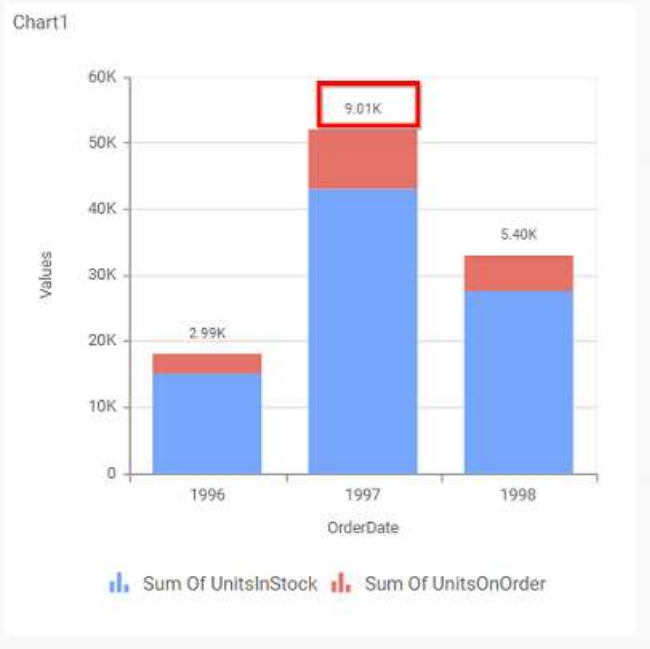


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like 1997(Sum of UnitInStock)



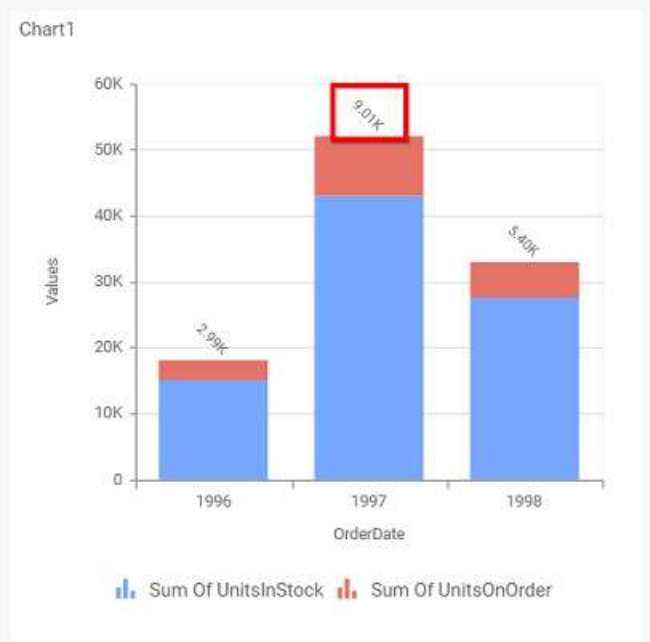
Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

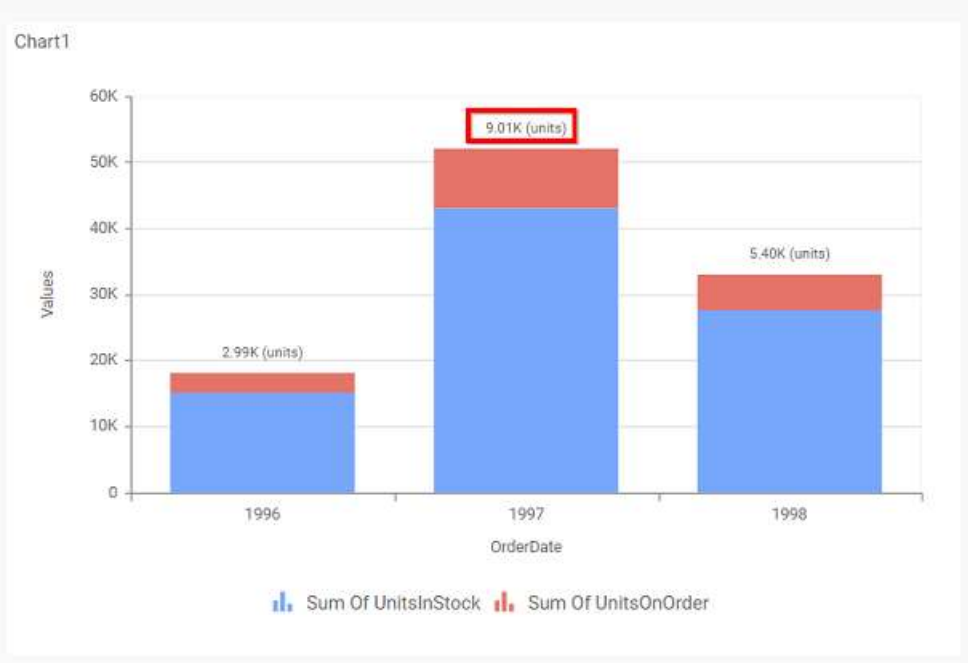


Value Label Suffix

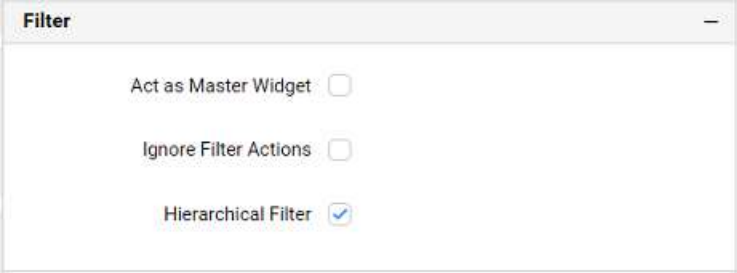
Allows you to enable the Suffix value text to the value labels.

Suffix Value

Allows you to set\edit suffix value to the value labels.



Filter



Act as Master Widget

This allows you to define this stacked column chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

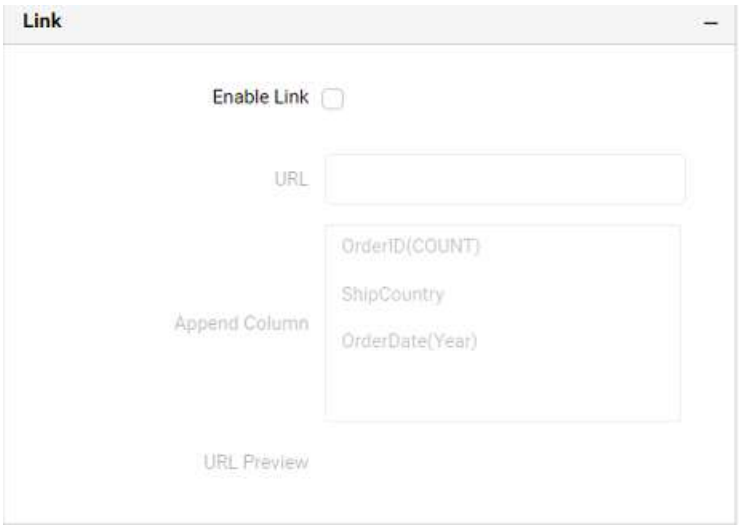
This allows you to define this stacked column chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

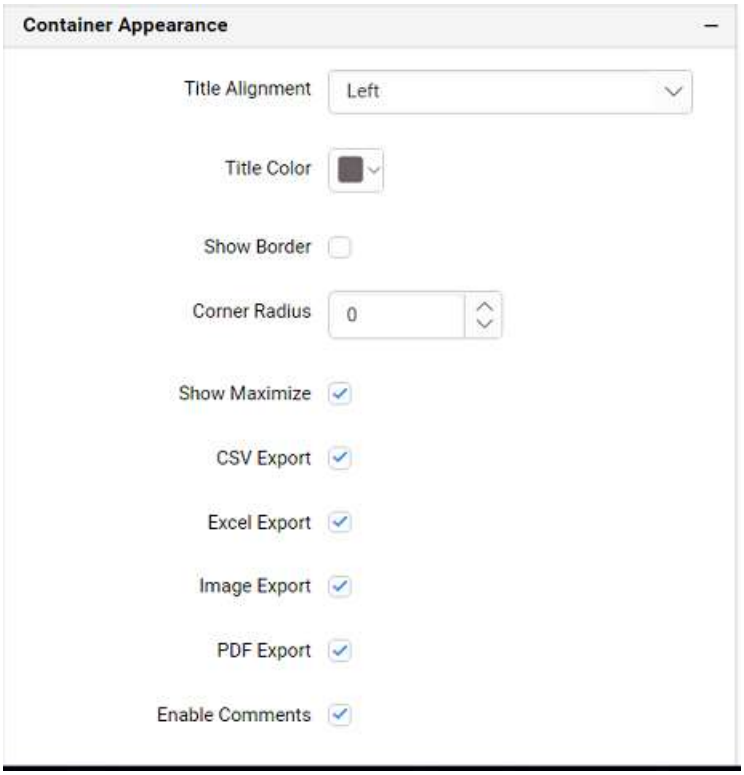
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this stacked column chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this stacked column widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this stacked column widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

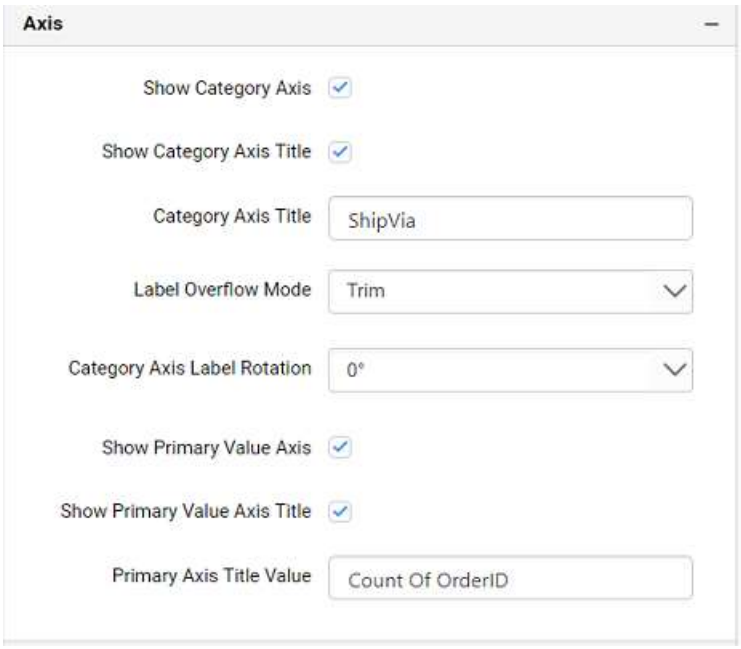
Image Export

This allows you to enable/disable the image export option for this stacked column chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

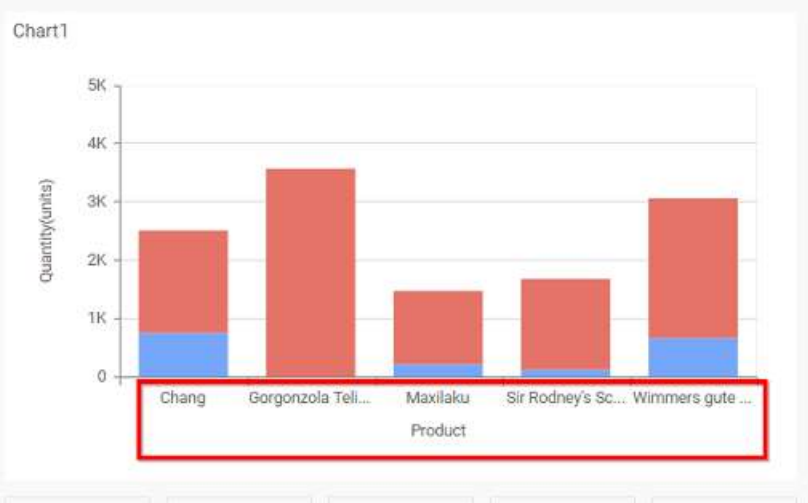
Axis



This section allows you to customize the axis settings in chart.

Show Category Axis

This allows to enable the visibility of Category Axis.



Show Category Axis Title

This allows you to enable the visibility of Category Axis title.

Category Axis Title

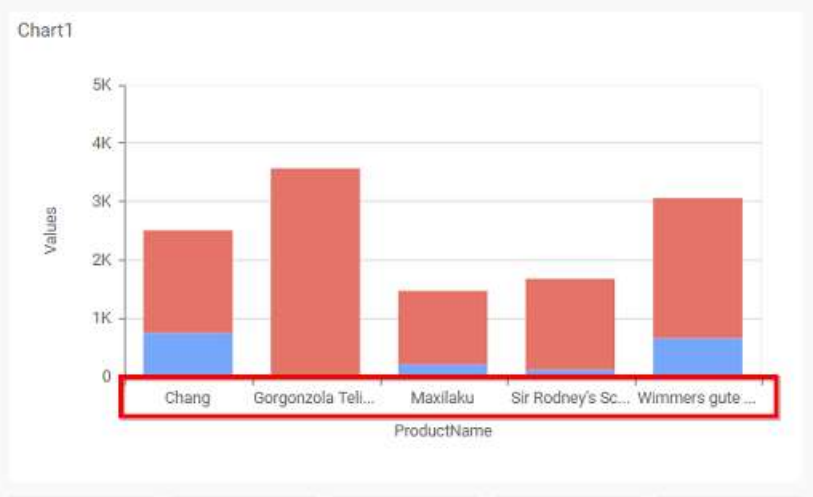
This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

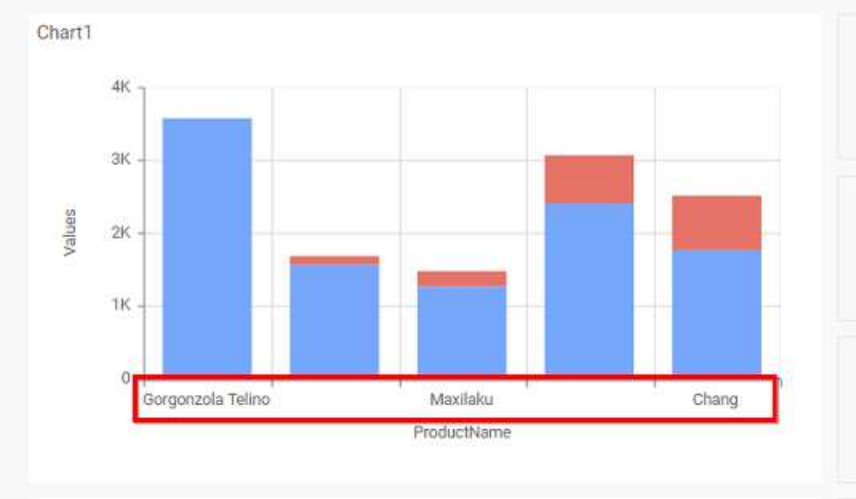
Trim

This option trims the end of overlapping label in the axis.



Hide

This option hides the overlapping label in the axis.

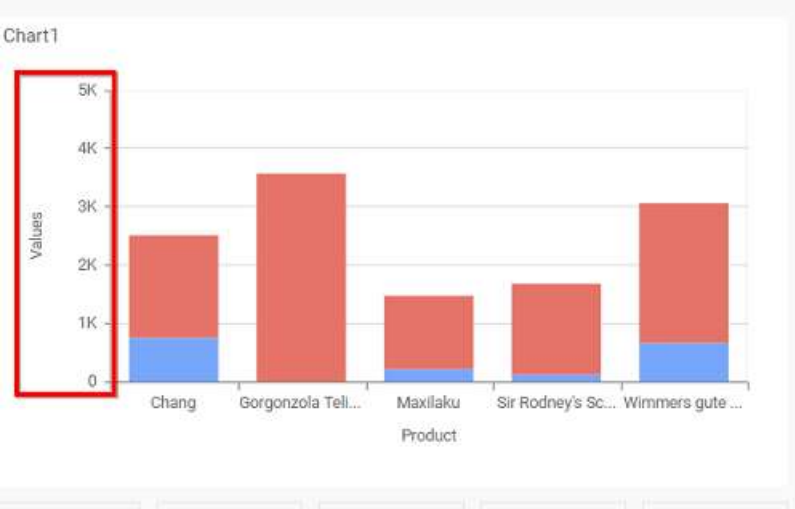


Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

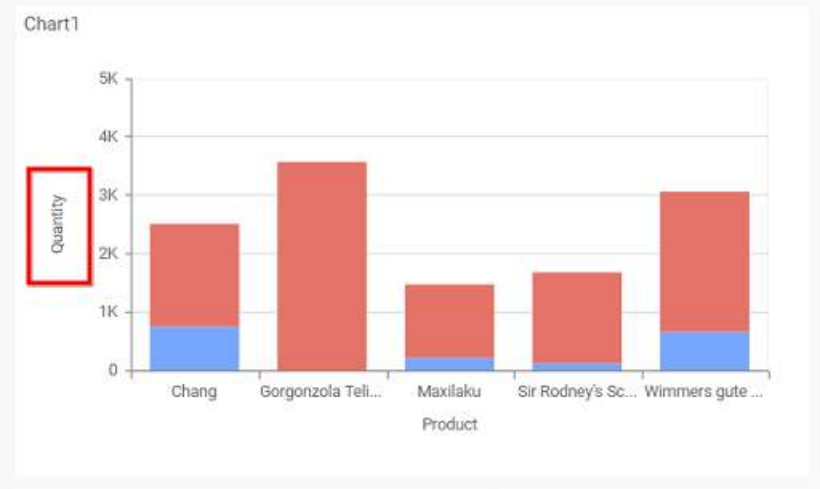
Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.



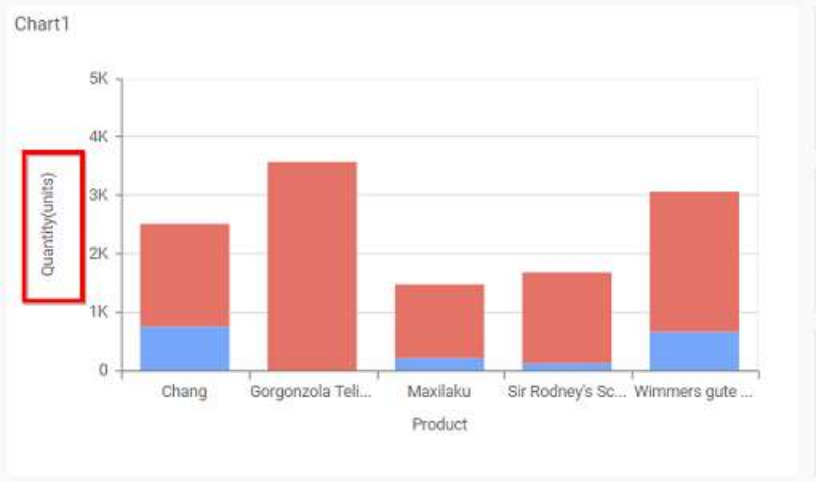
Show Primary Value Axis Title

This allows you to enable the visibility of Primary Value Axis title of chart.

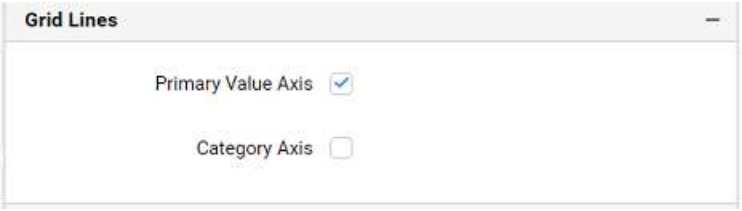


Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.



Grid Lines



Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the stacked column chart.

Category Axis

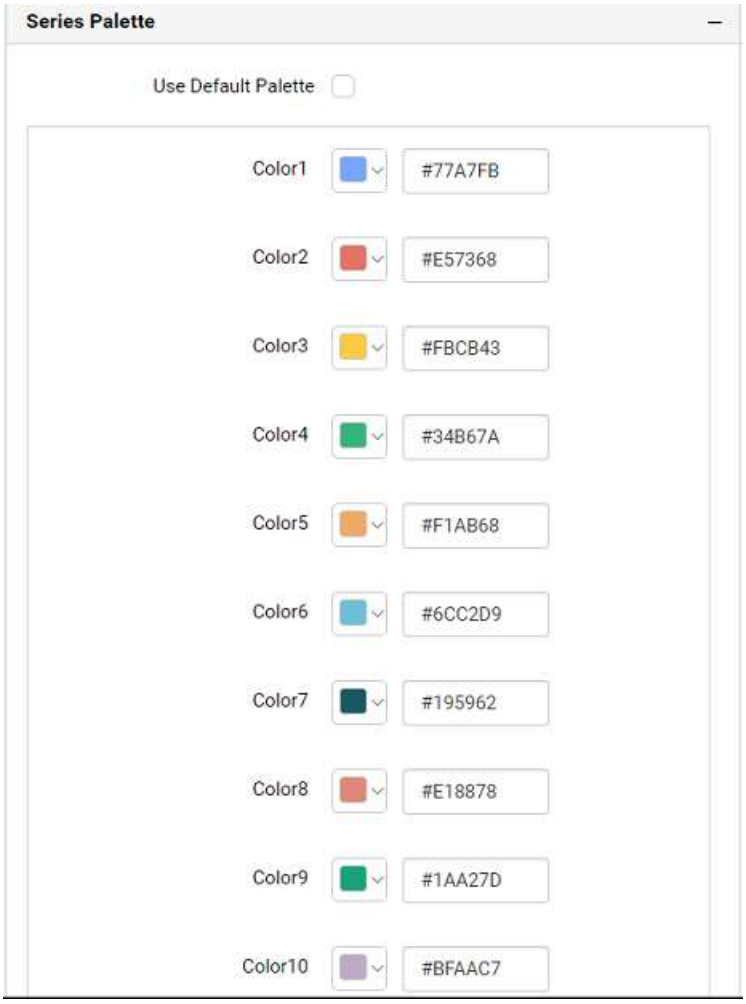
This allows you to enable the Category Axis gridlines for the stacked column chart.

Series Palette

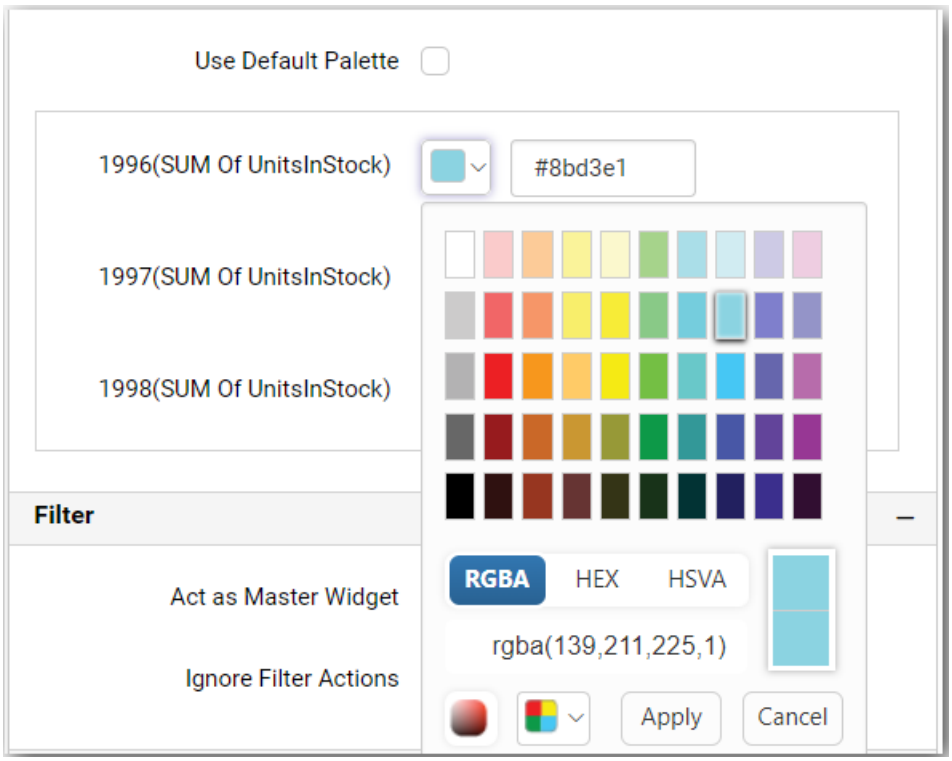
This allows you to customize the chart series color through Series Palette section.

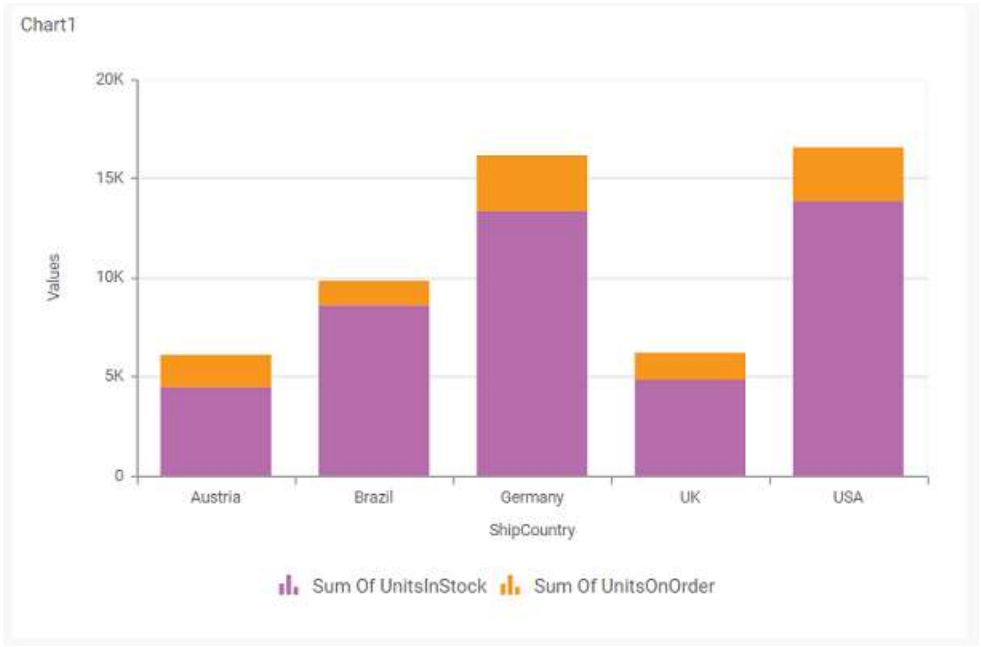
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



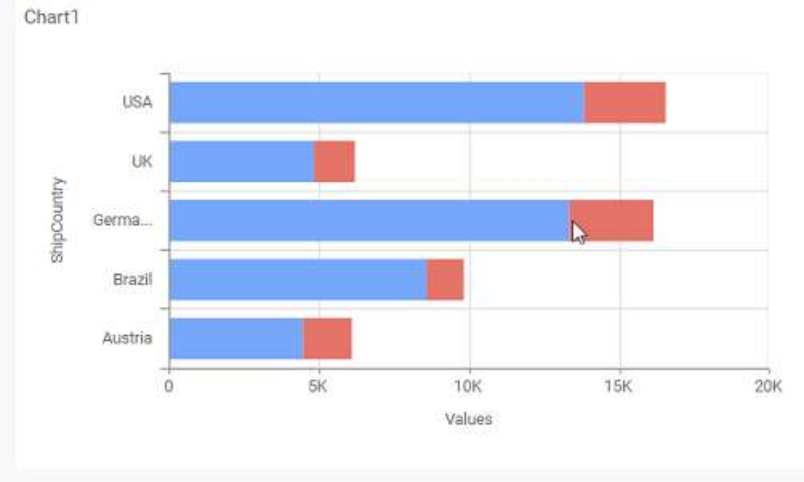
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Stacked Bar Chart

Stacked Bar Chart allows you to compare multiple measures through bars stacked one after the other horizontally.

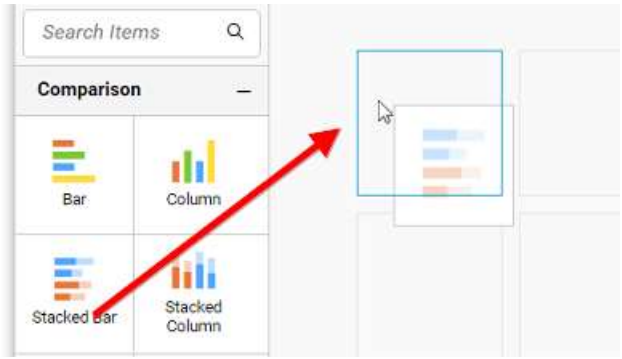


How to configure the flat table data to stacked bar chart?

Stacked Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

Follow the steps to configure data to stacked bar chart.

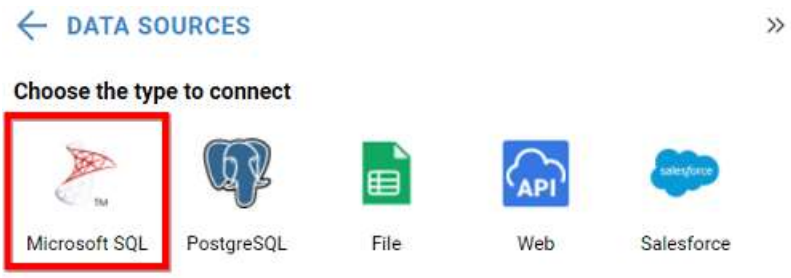
Drag and drop the stacked bar chart into canvas and resize it your required size.



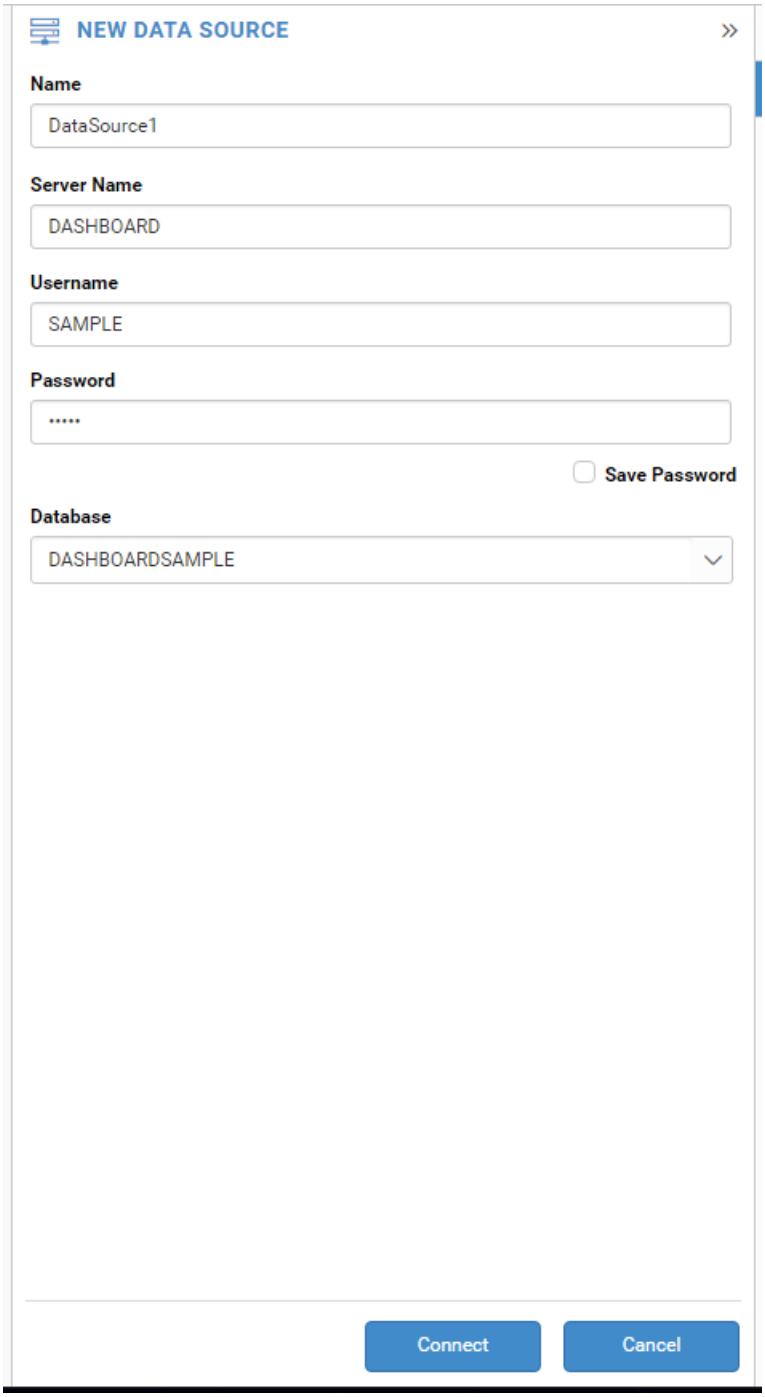
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

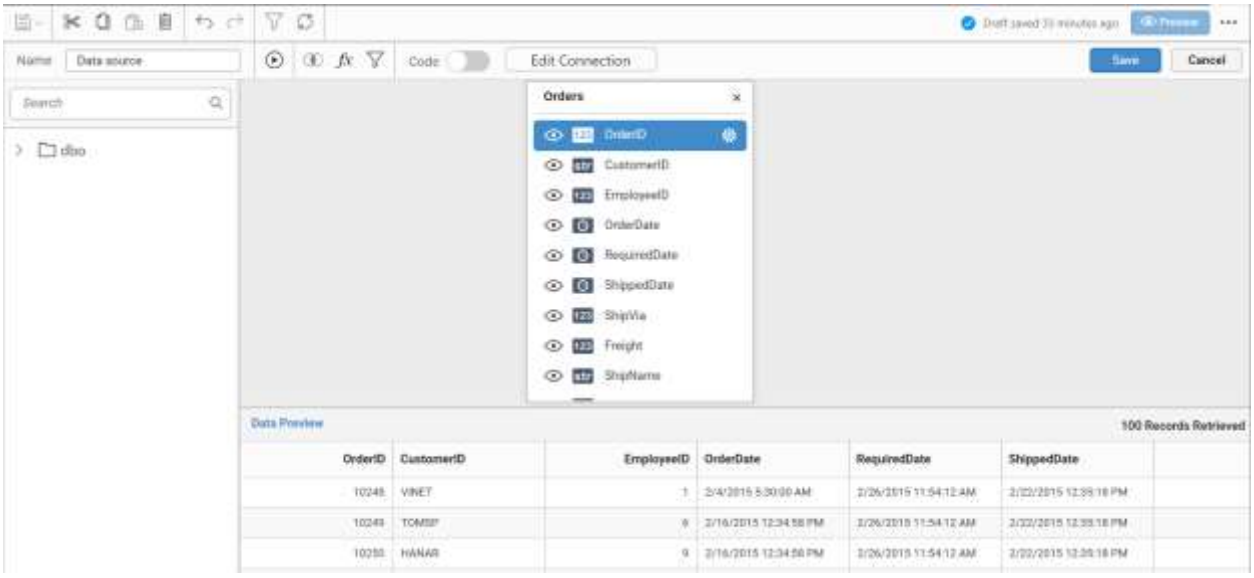
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



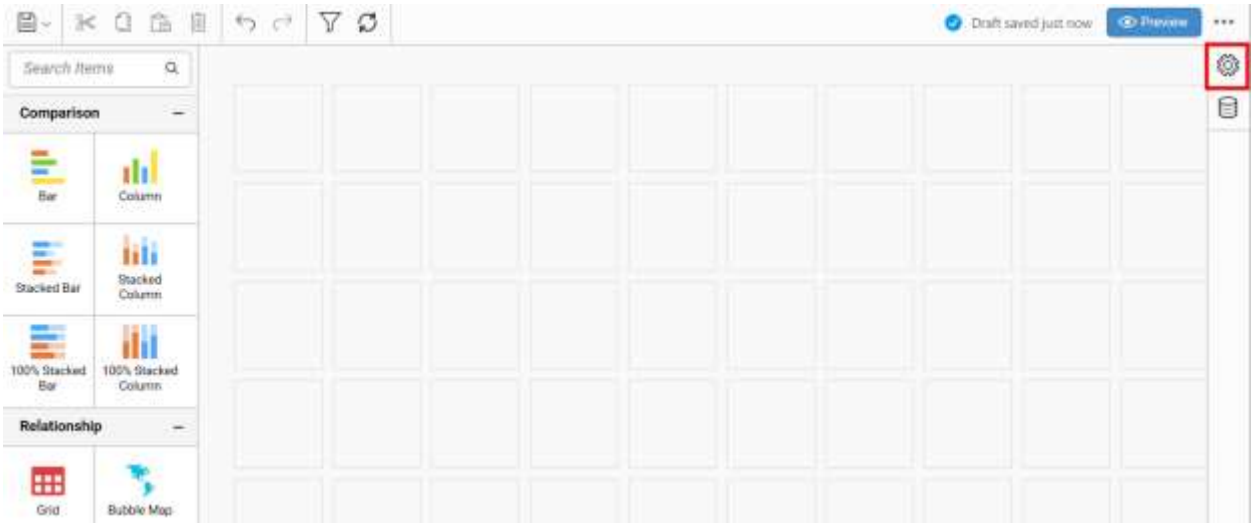
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

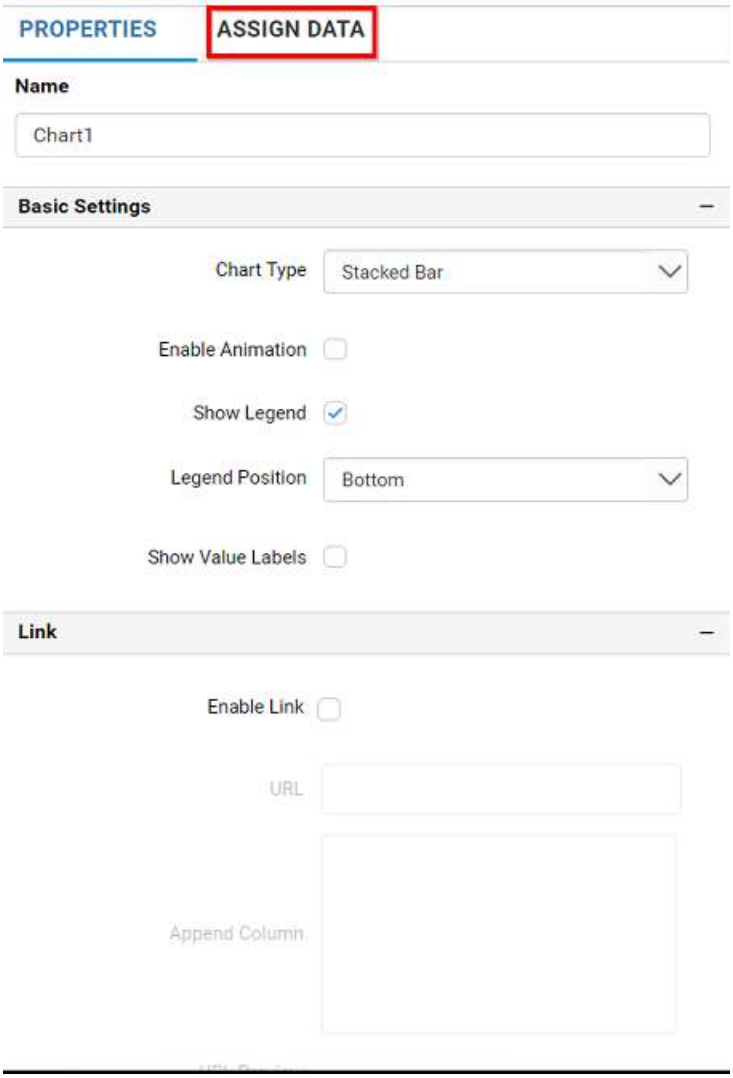


Drag your preferred table or view from the left pane from data design view, click Save button.

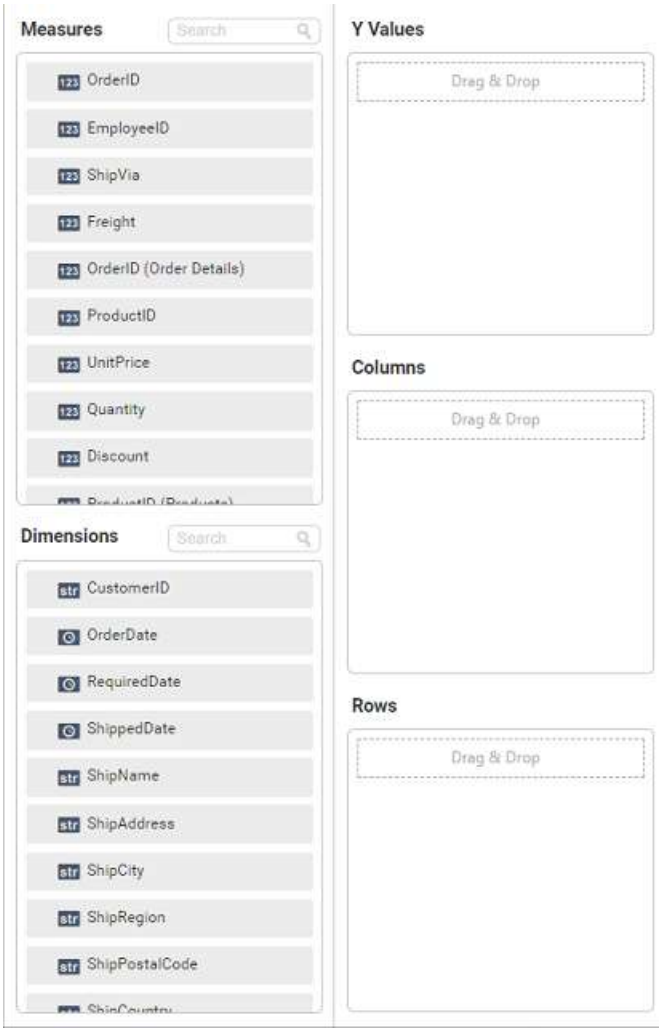


Click Properties button in configuration panel, property pane opens. Now, switch to ASSIGN DATA tab.





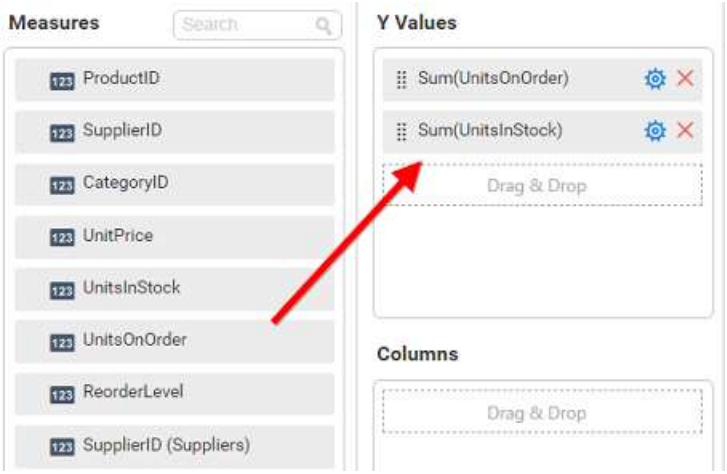
The data tab will be opened with available measures and dimensions from the connected data source



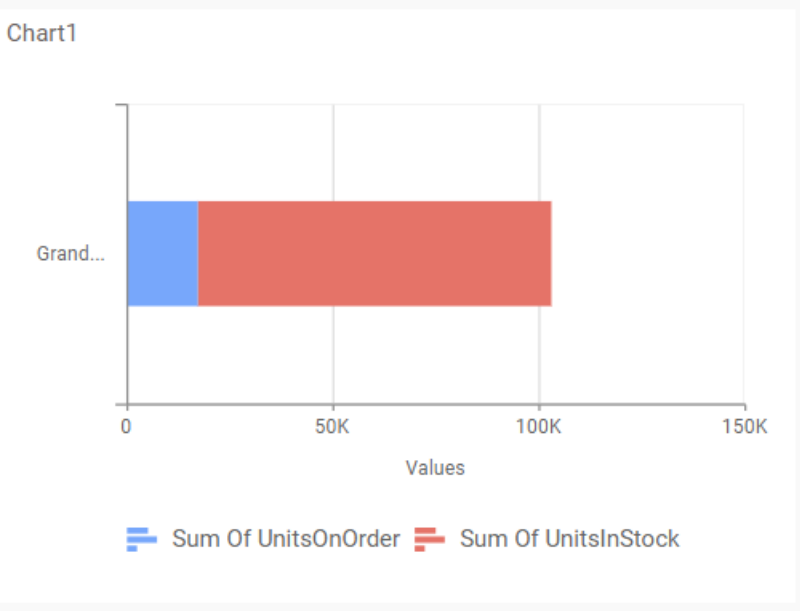
You can add the required data from Measures and Dimensions into required field.

Adding Y Values

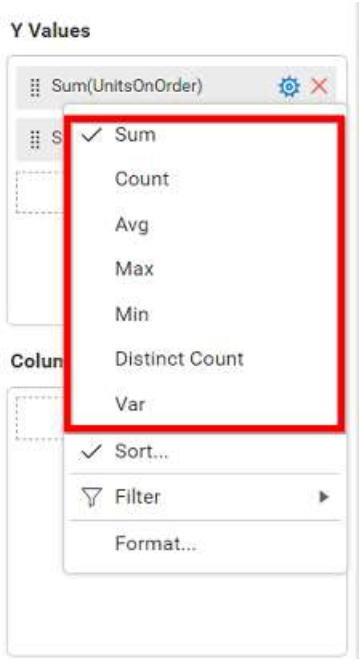
You can add more than one Measures into Y Values field by drag and drop the required measure.



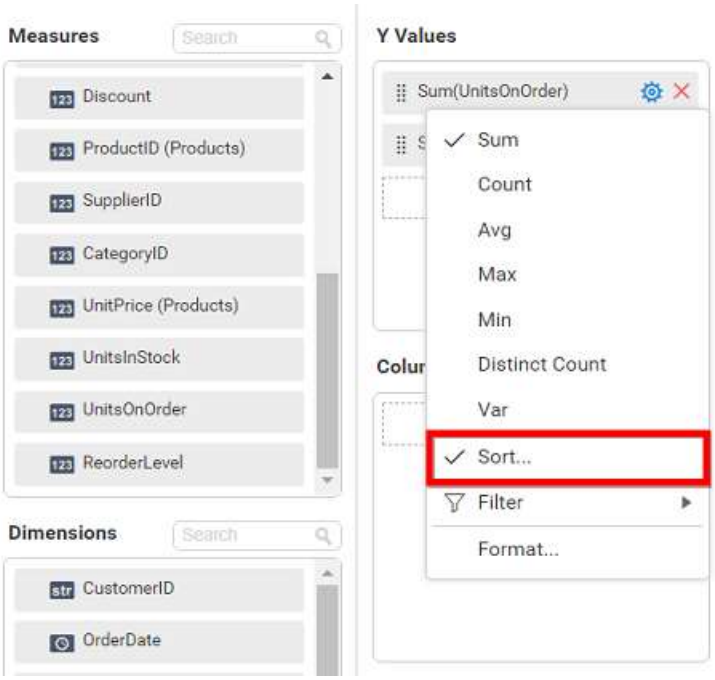
Now the stacked bar chart will be rendered like this



Click the Settings option to change required summary type from the available summary types shown in Settings.

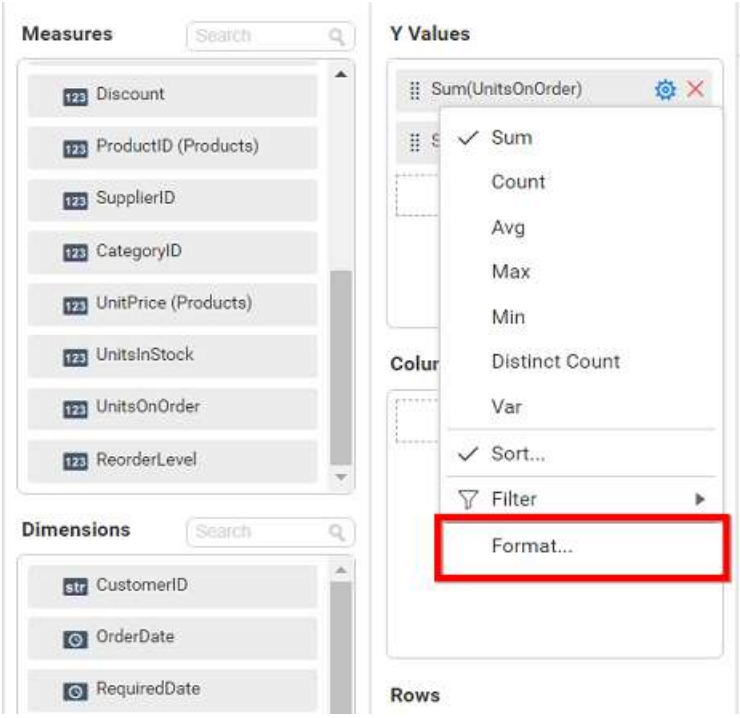


You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort



You can filter the data to be displayed in funnel chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

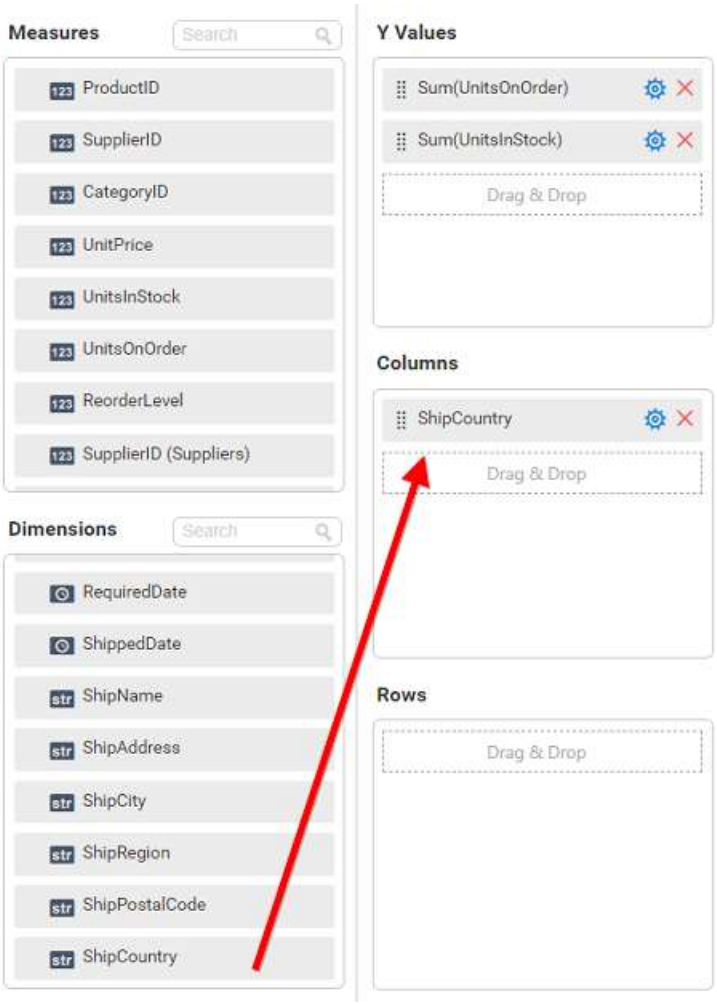


To remove the added value fields click highlighted button.

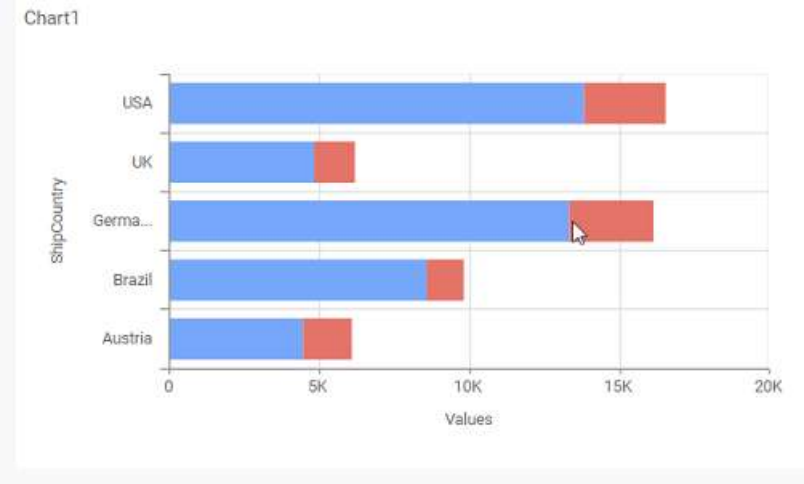
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field. If you add more than one column, then drill down option enabled automatically.



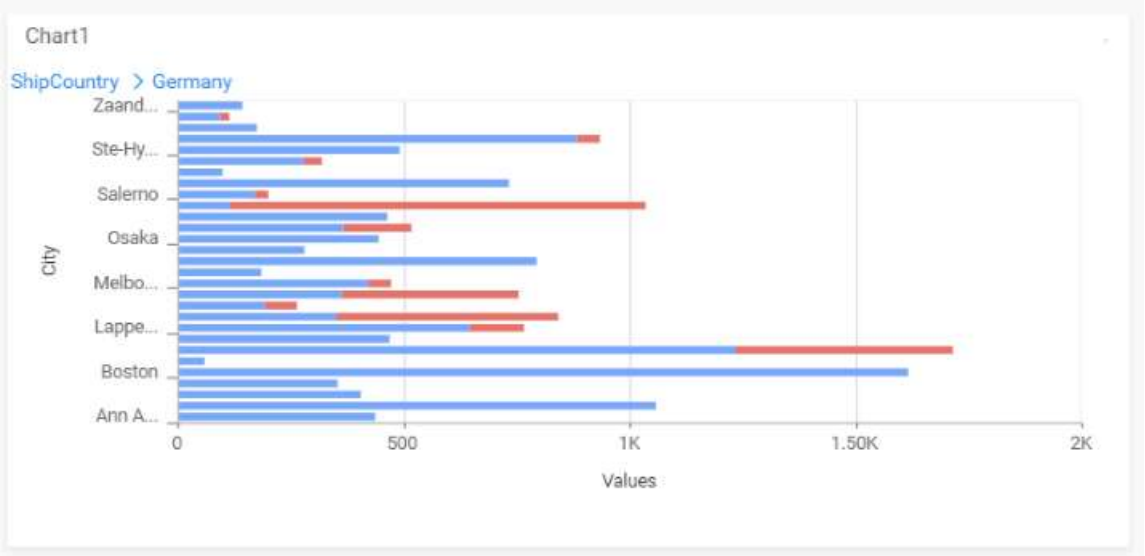
Stacked bar chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

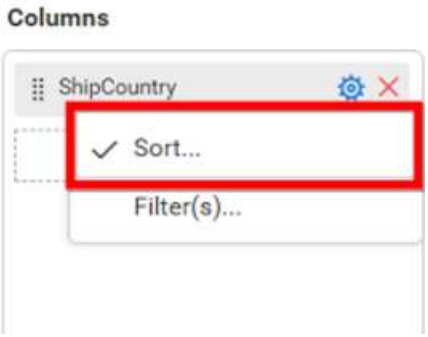
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.

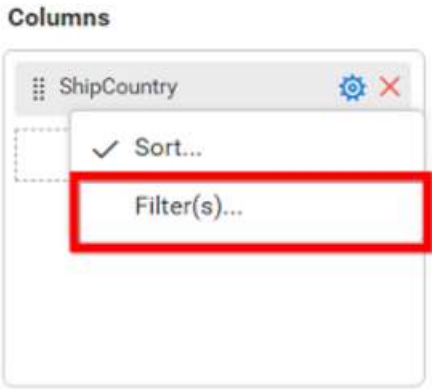


You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply Sorting for the data, refer Sort.



You can apply filters by selecting filter in settings. For more details, refer filter.

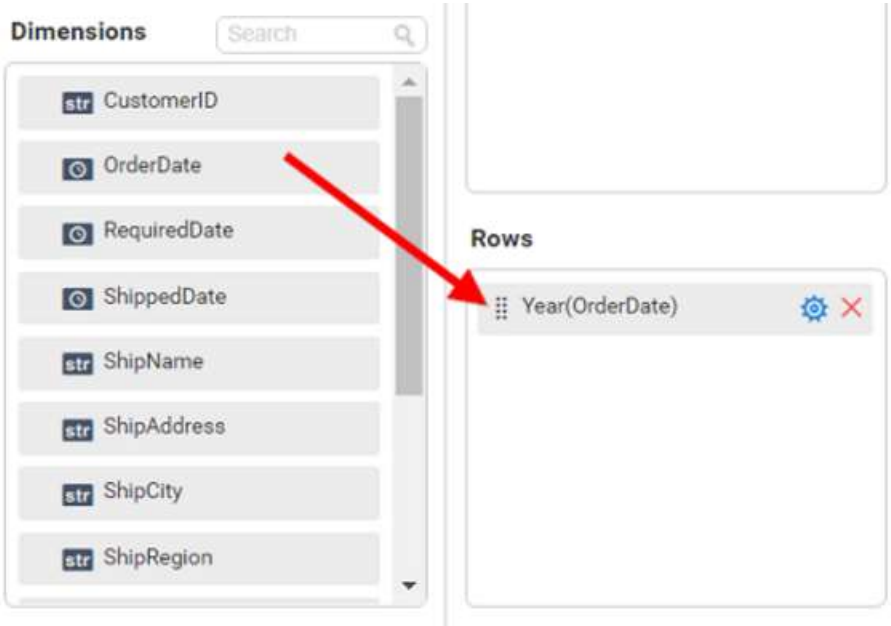


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

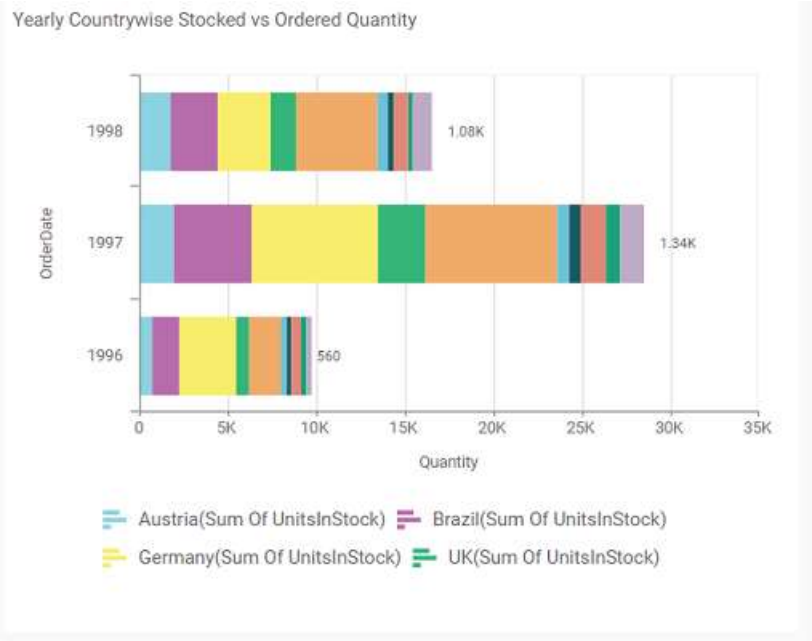
Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render stacked bar chart in series.



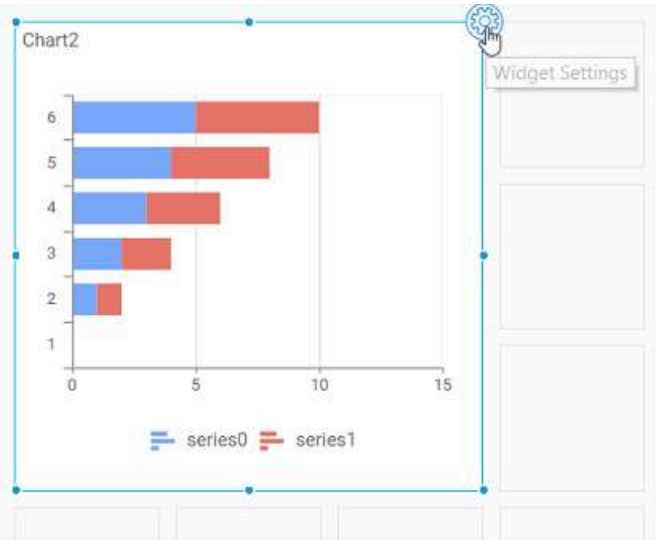
How to format stacked bar chart?

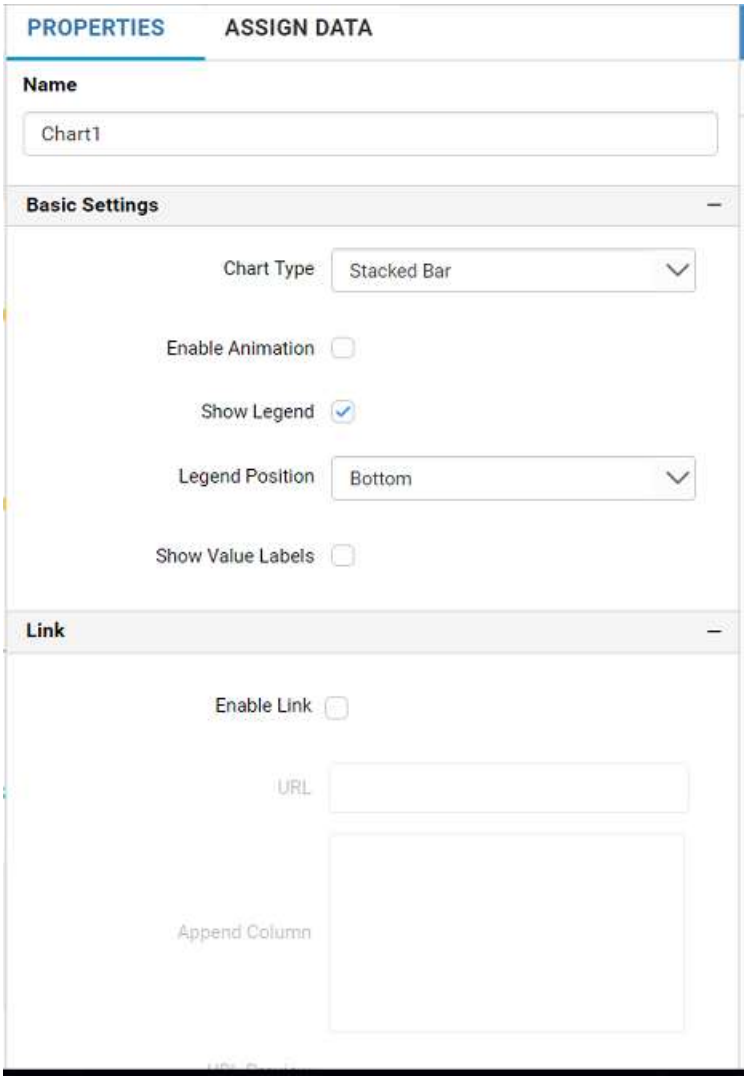
You can format the stacked bar chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into stacked bar chart follow the steps

1. Drag and drop the stacked bar chart into canvas and resize it to your required size.
2. Configure the data into stacked bar chart.
3. Focus on the stacked chart and click on widget settings.

The property window will be opened.





You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this stacked bar chart widget

Basic Settings

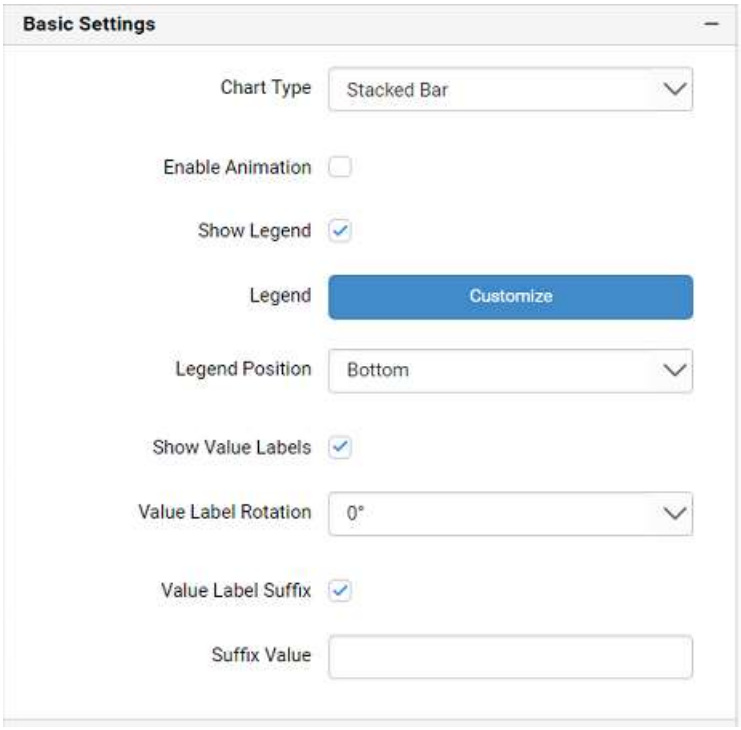


Chart Type

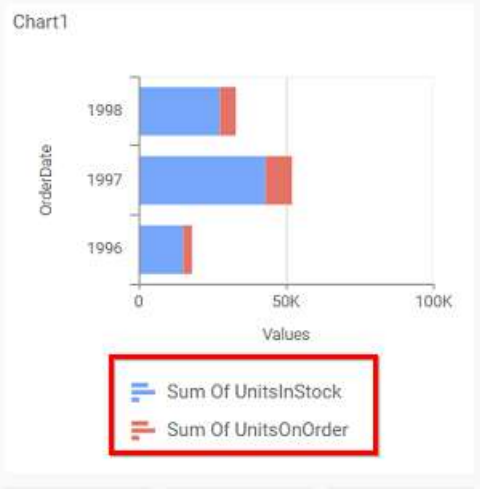
This allows you to switch the widget view from current chart type to another convertible chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

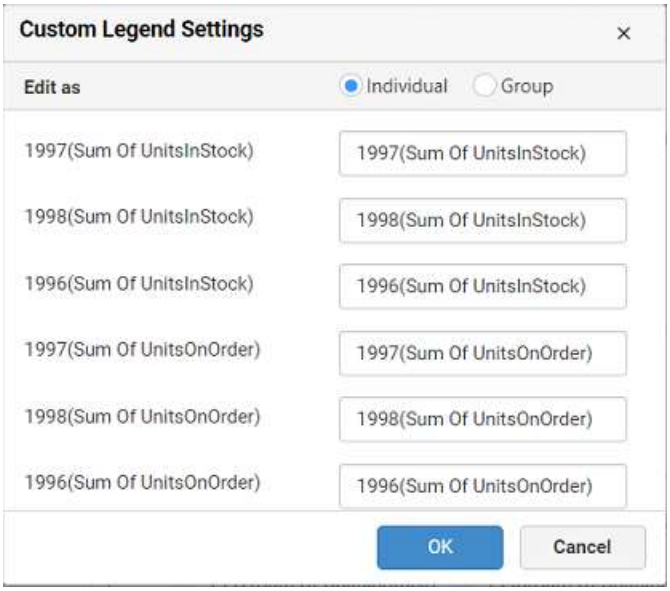
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

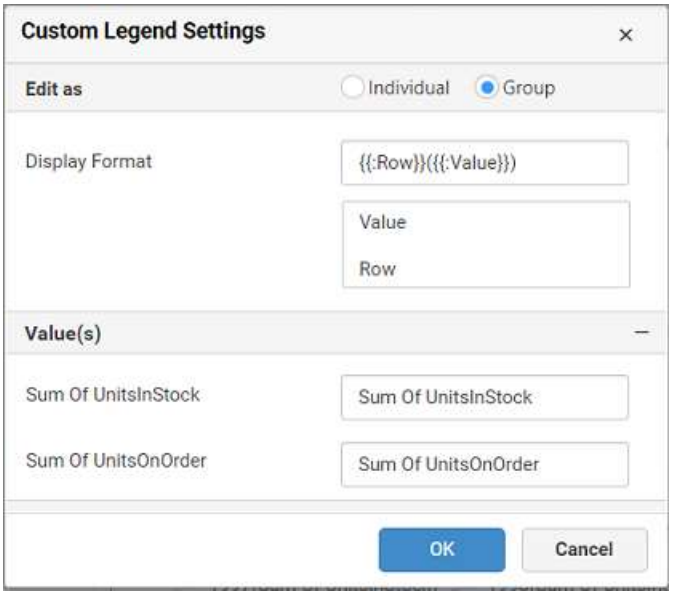
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.

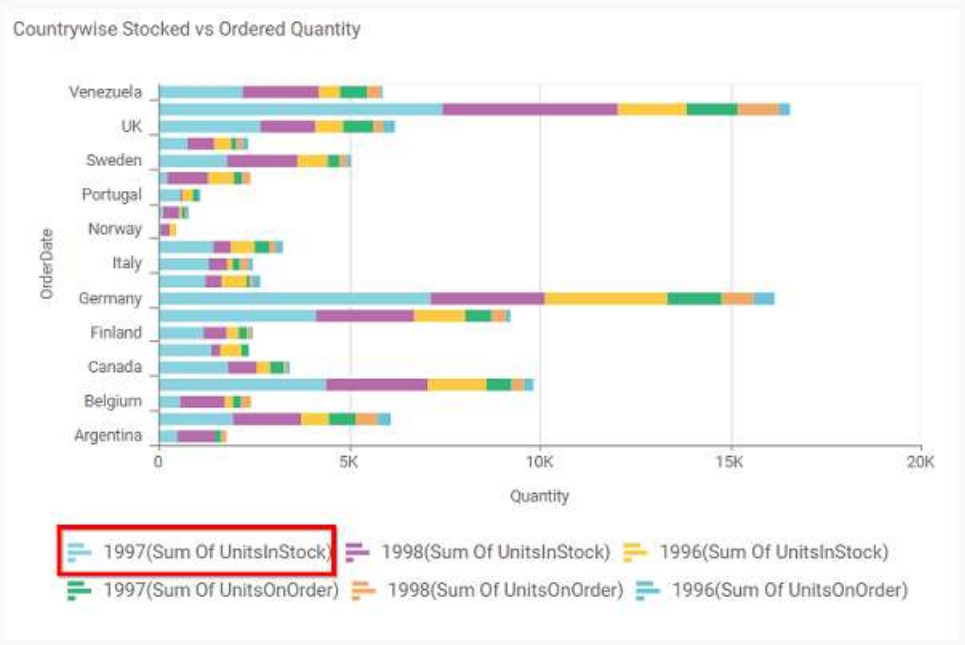


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

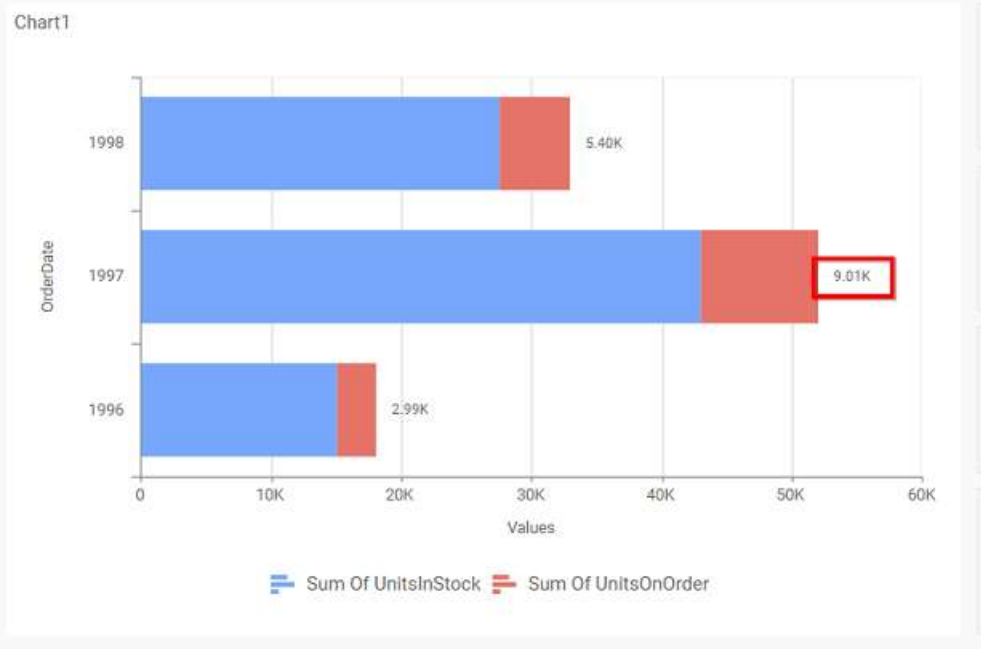


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like 1997(Sum of UnitInStock)



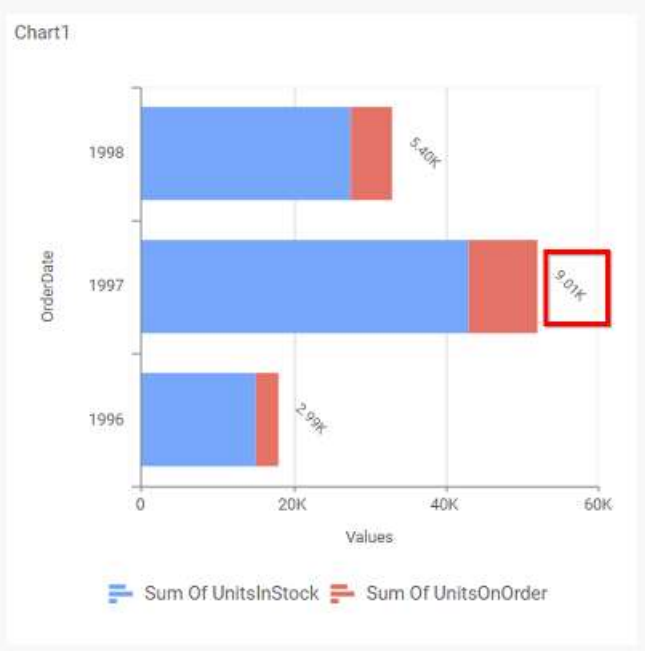
Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

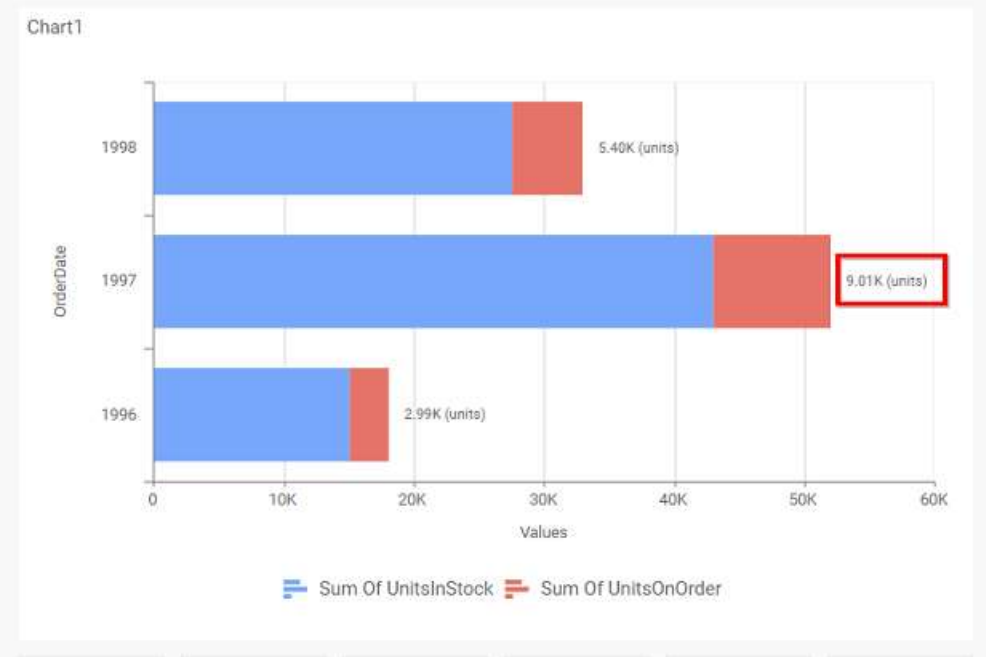


Value Label Suffix

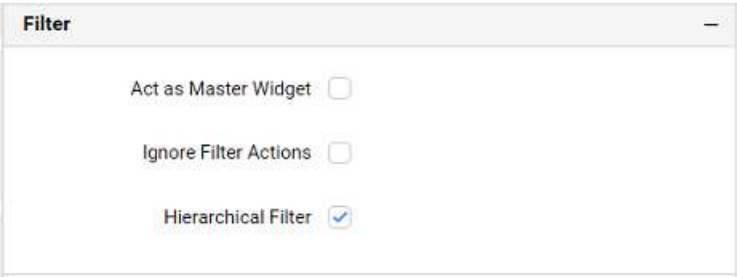
Allows you to enable the Suffix value text to the value labels.

Suffix Value

Allows you to set\edit suffix value to the value labels.



Filter



Act as Master Widget

This allows you to define this stacked bar chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

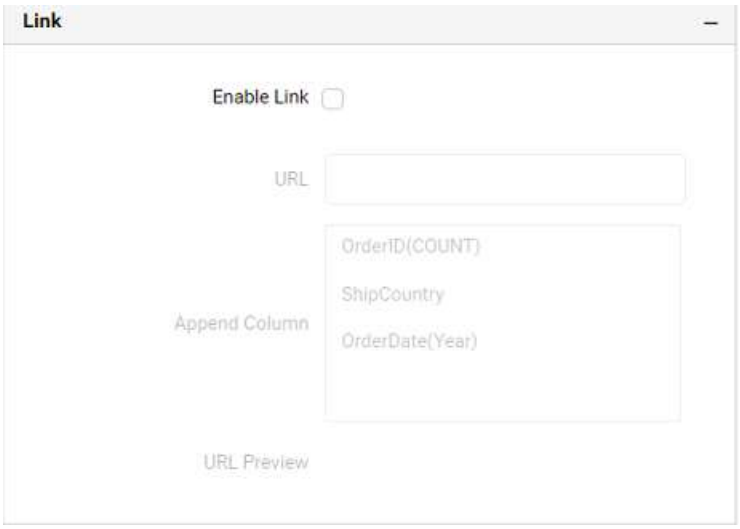
This allows you to define this stacked bar chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

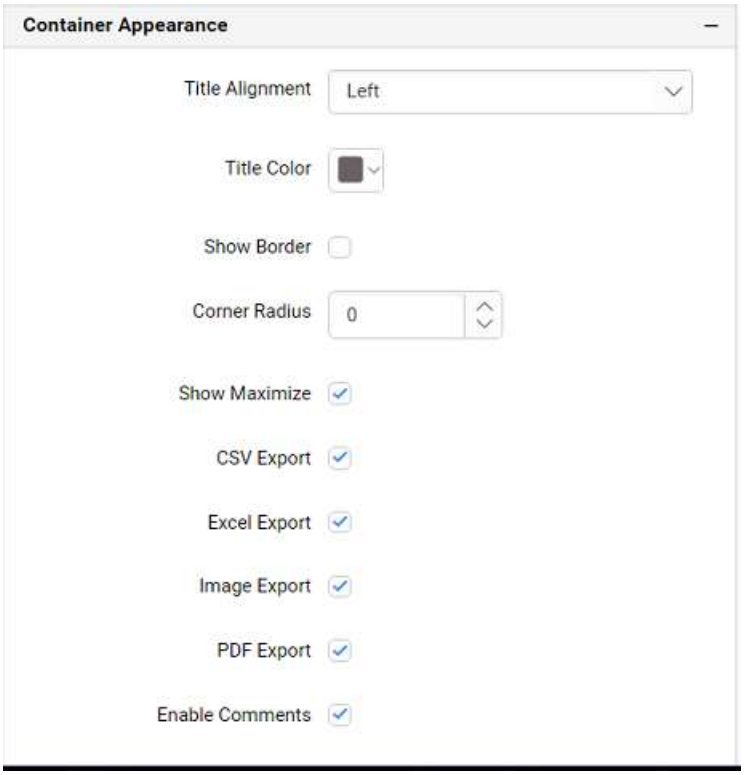
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this stacked bar chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this stacked bar chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this stacked bar chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

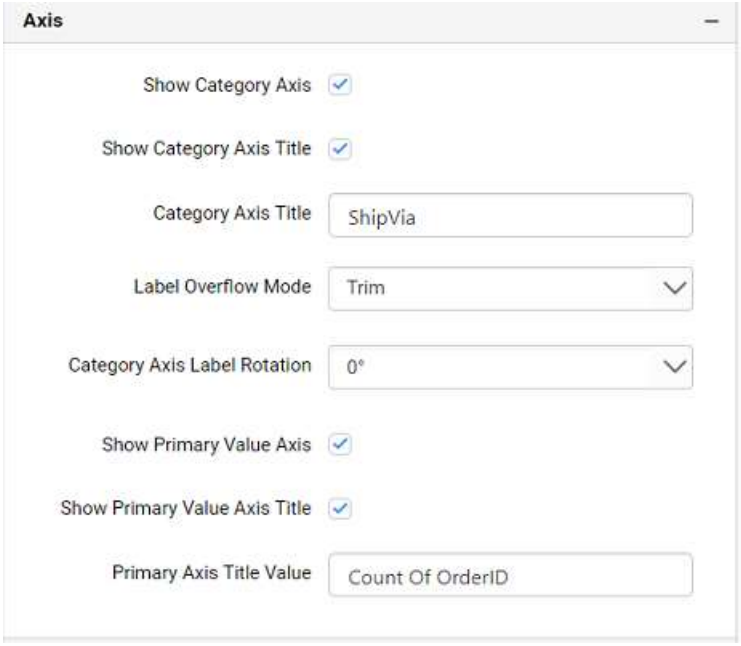
Image Export

This allows you to enable/disable the image export option for this stacked bar chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

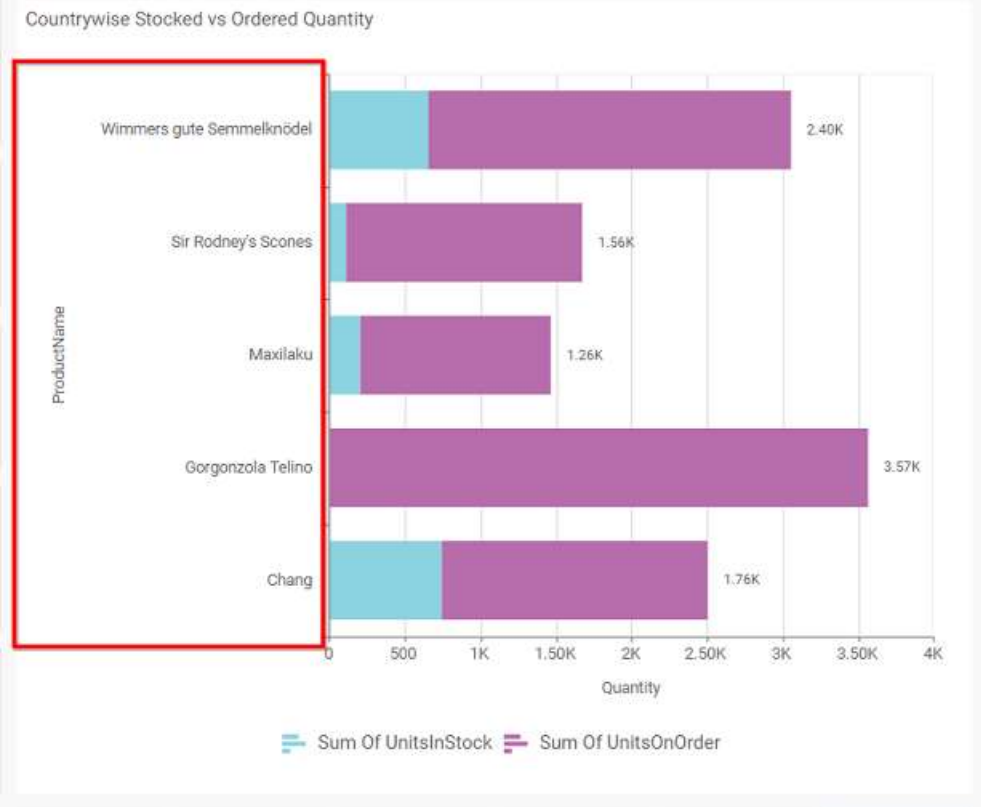
Axis



This section allows you to customize the axis settings in chart.

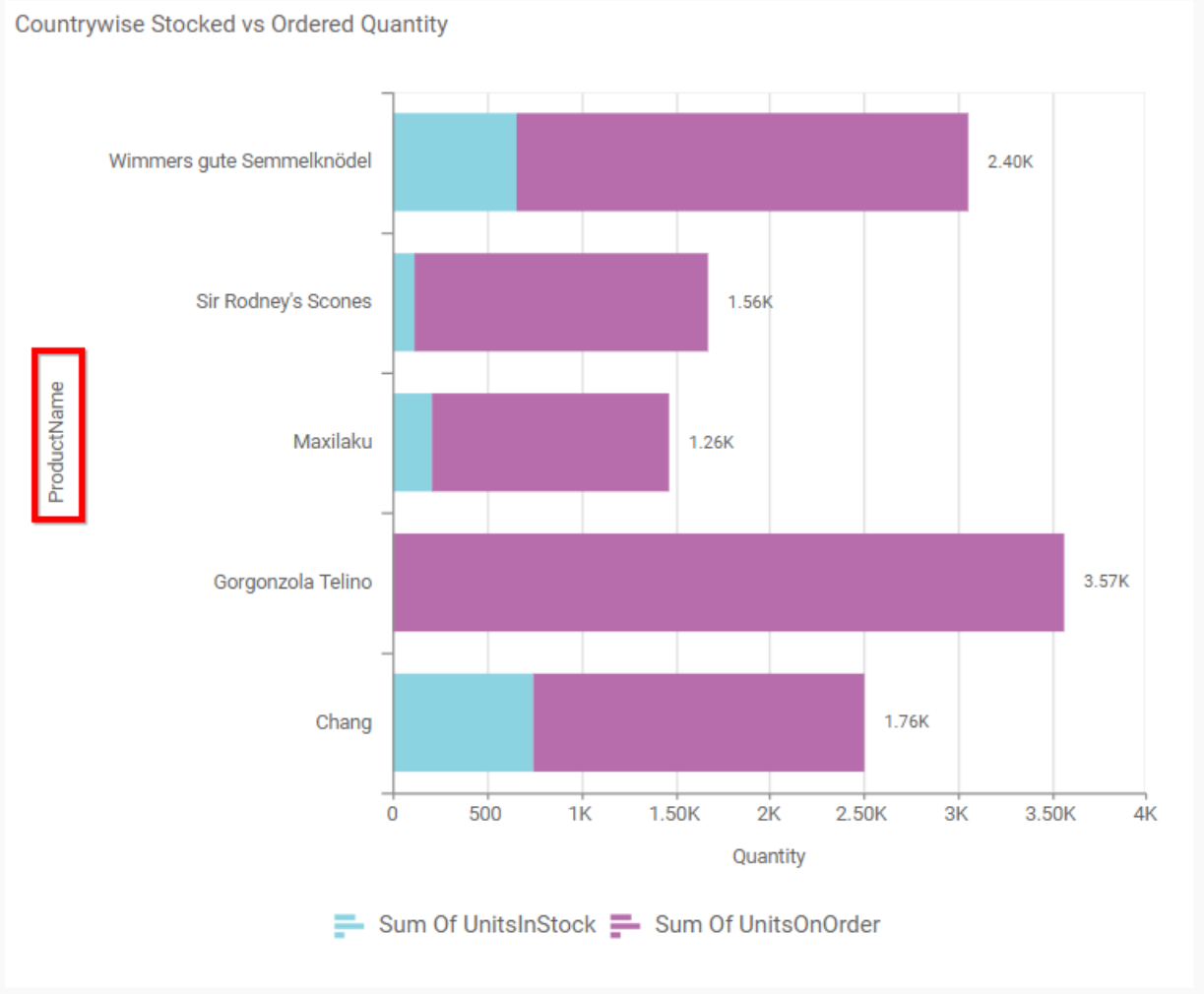
Show Category Axis

This allows to enable the visibility of Category Axis.



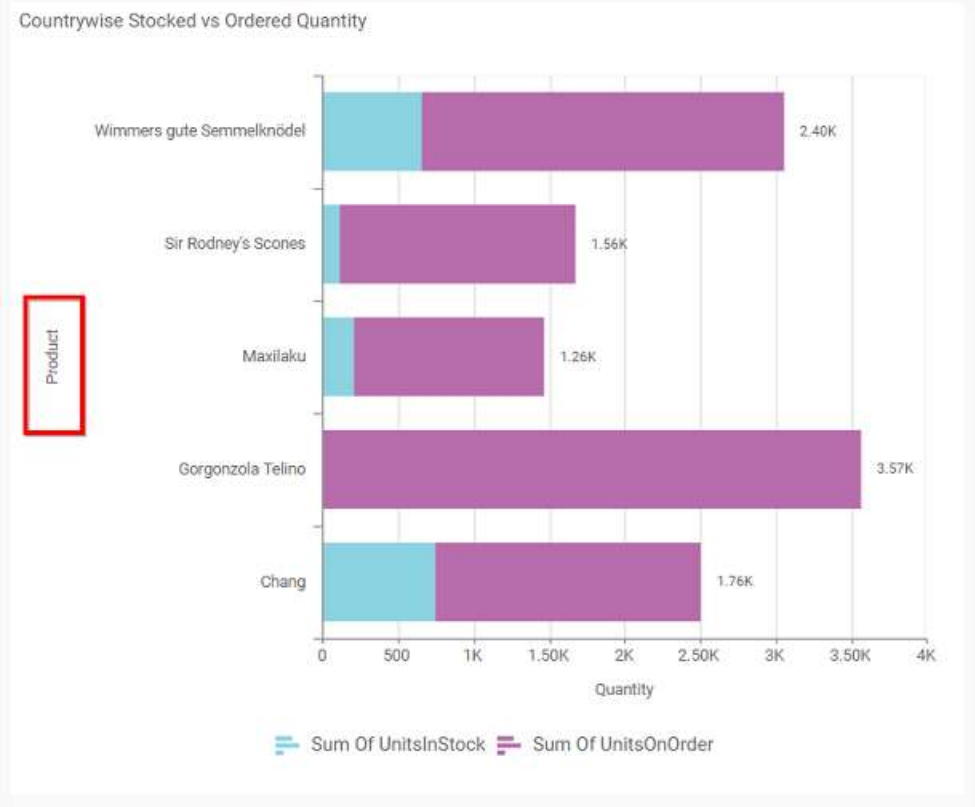
Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

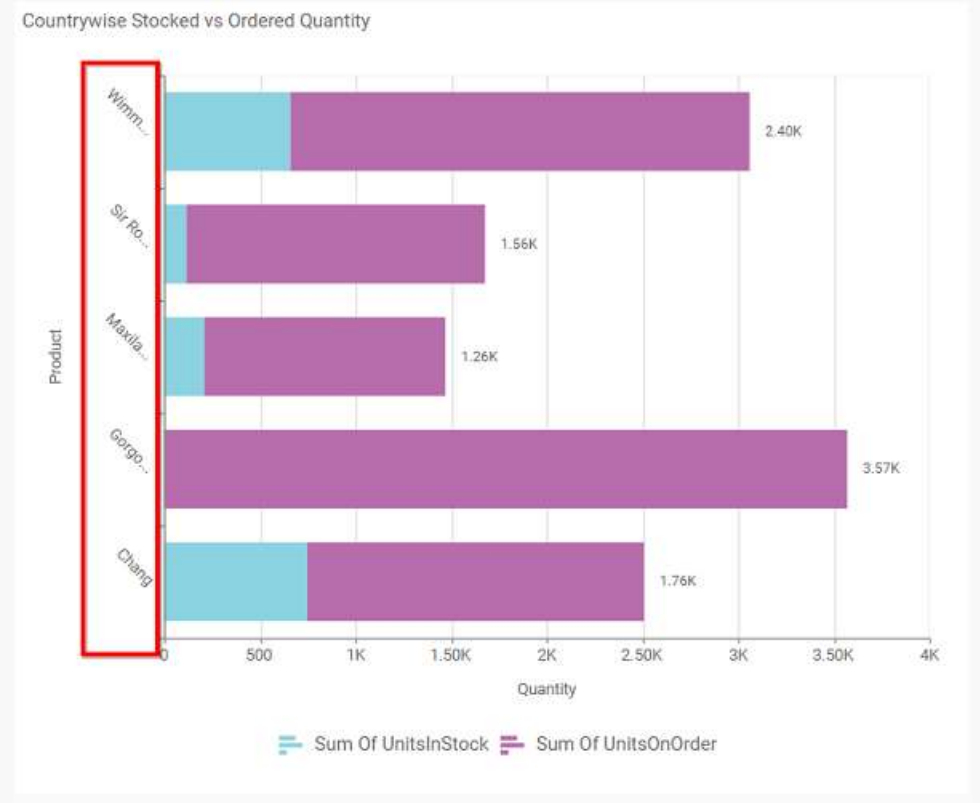


Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

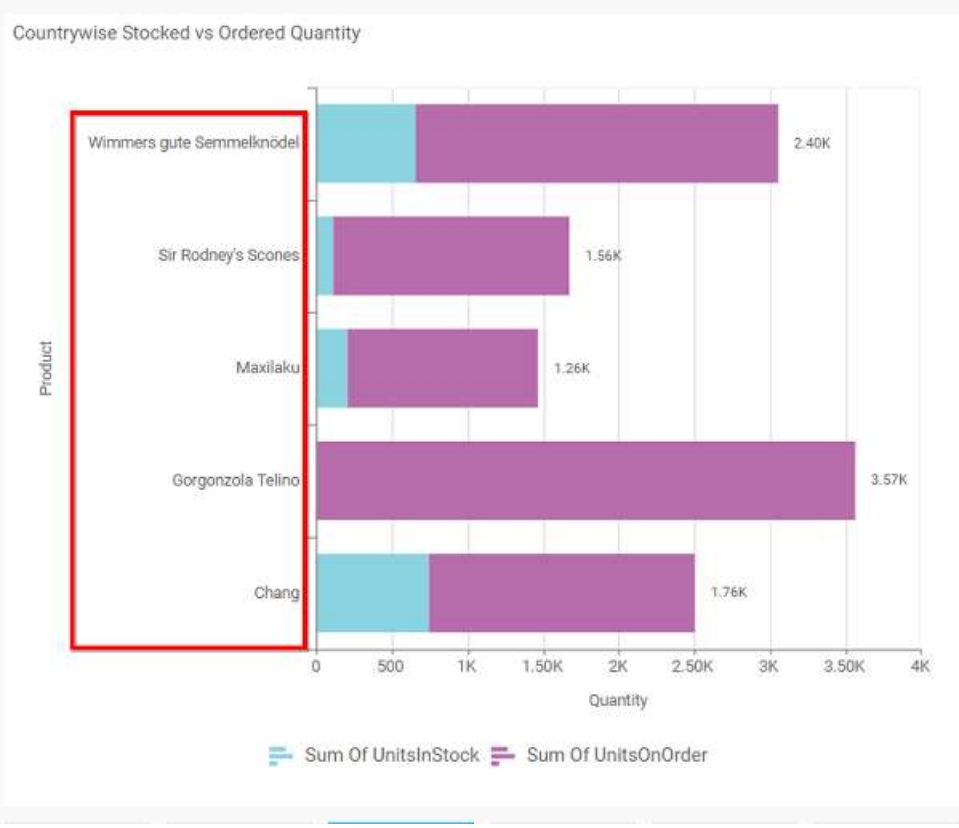
Trim

This option trims the end of overlapping label in the axis.



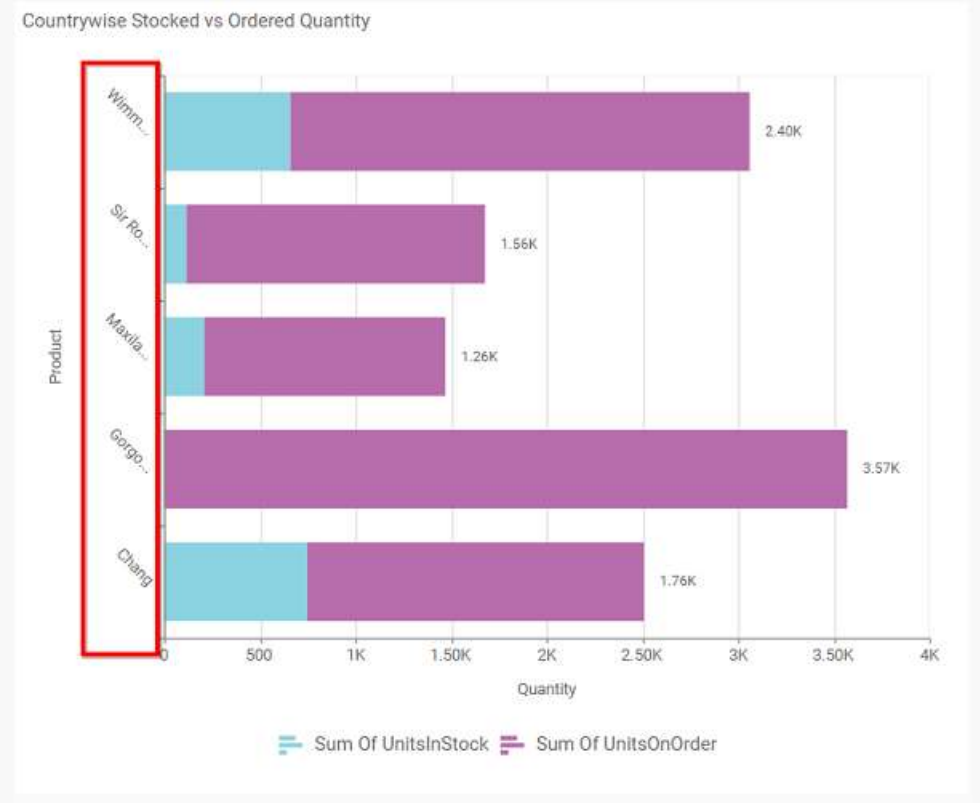
Hide

This option hides the overlapping label in the axis.



Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.

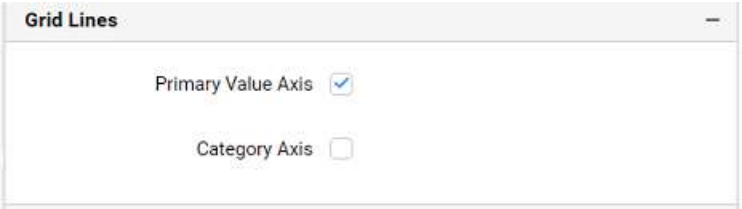
Show Primary Value Axis Title

This allows you to enable the visibility of Primary Value Axis title of chart.

Primary Value Axis Title

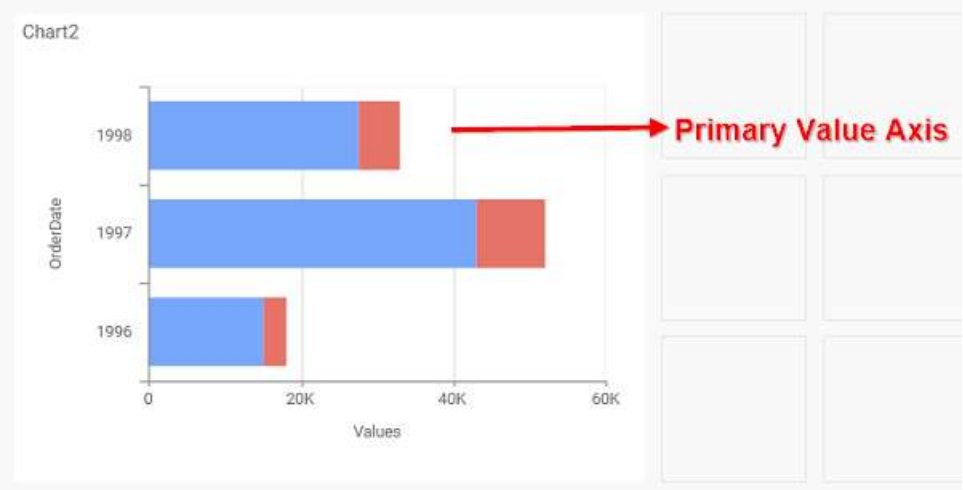
This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

Grid Lines



Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the stacked bar chart.



Category Axis

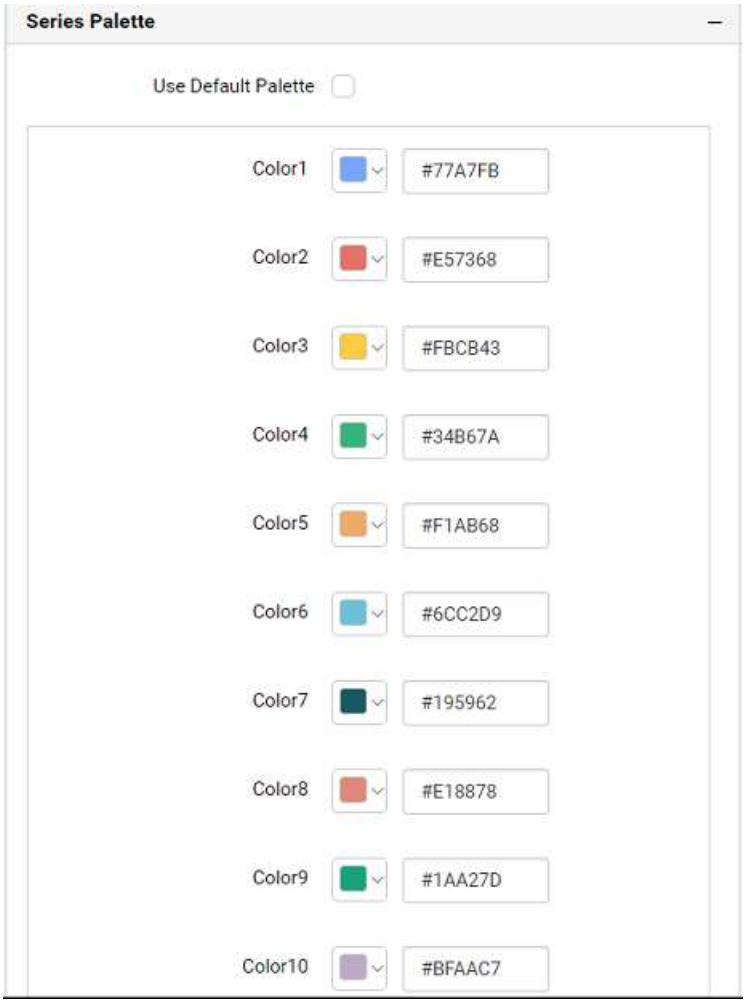
This allows you to enable the Category Axis gridlines for the stacked bar chart.

Series Palette

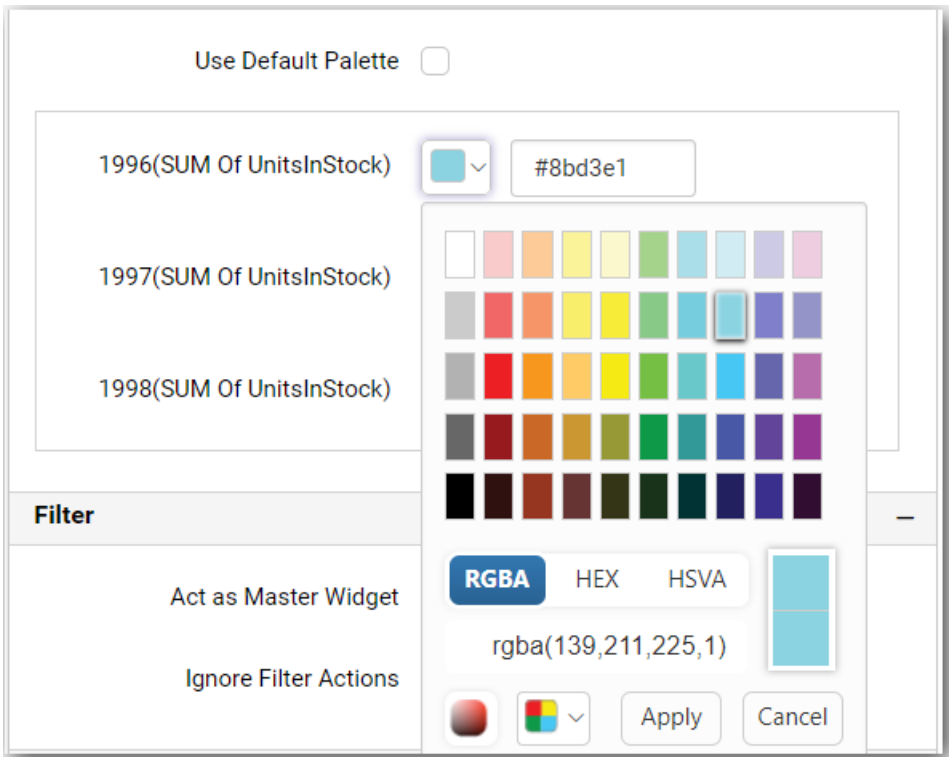
This allows you to customize the chart series color through Series Palette section.

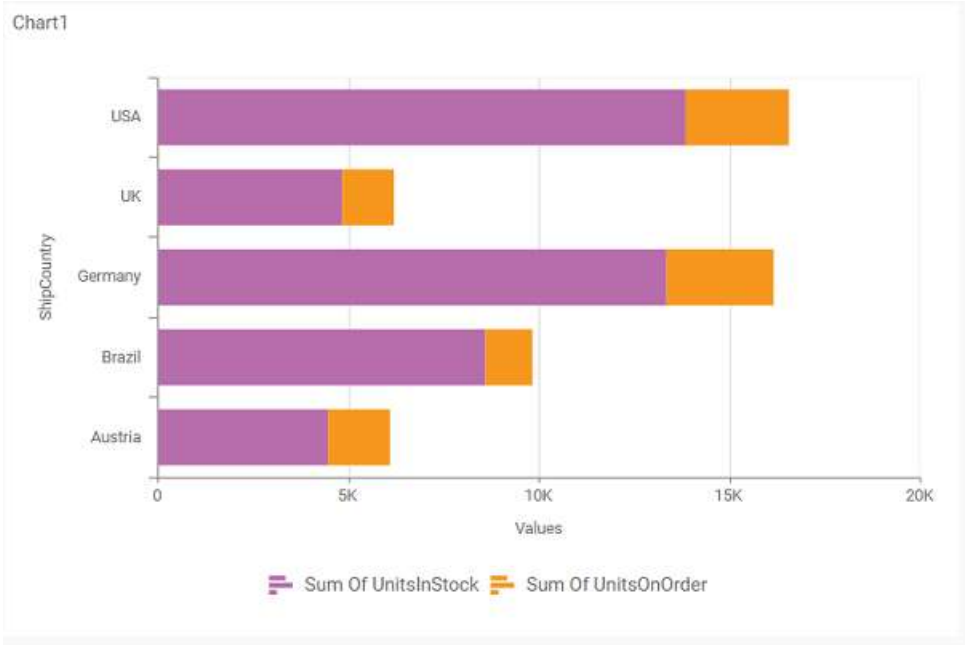
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





100% Stacked Column Chart

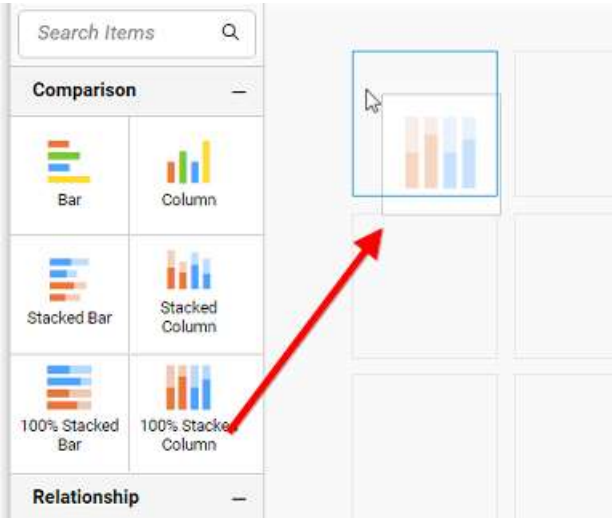
100% Stacked Column Chart allows you to compare multiple measures through bars stacked one after the other vertically.

How to configure the table data to 100% stacked column chart?

100% Stacked Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

Follow the steps configure data to 100% stacked column chart

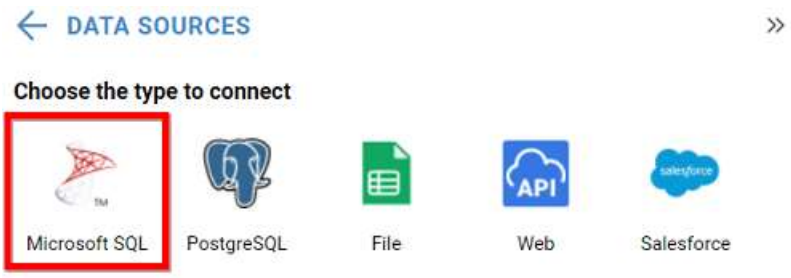
Drag and drop the 100% stacked column chart to canvas and resize it to your required size.



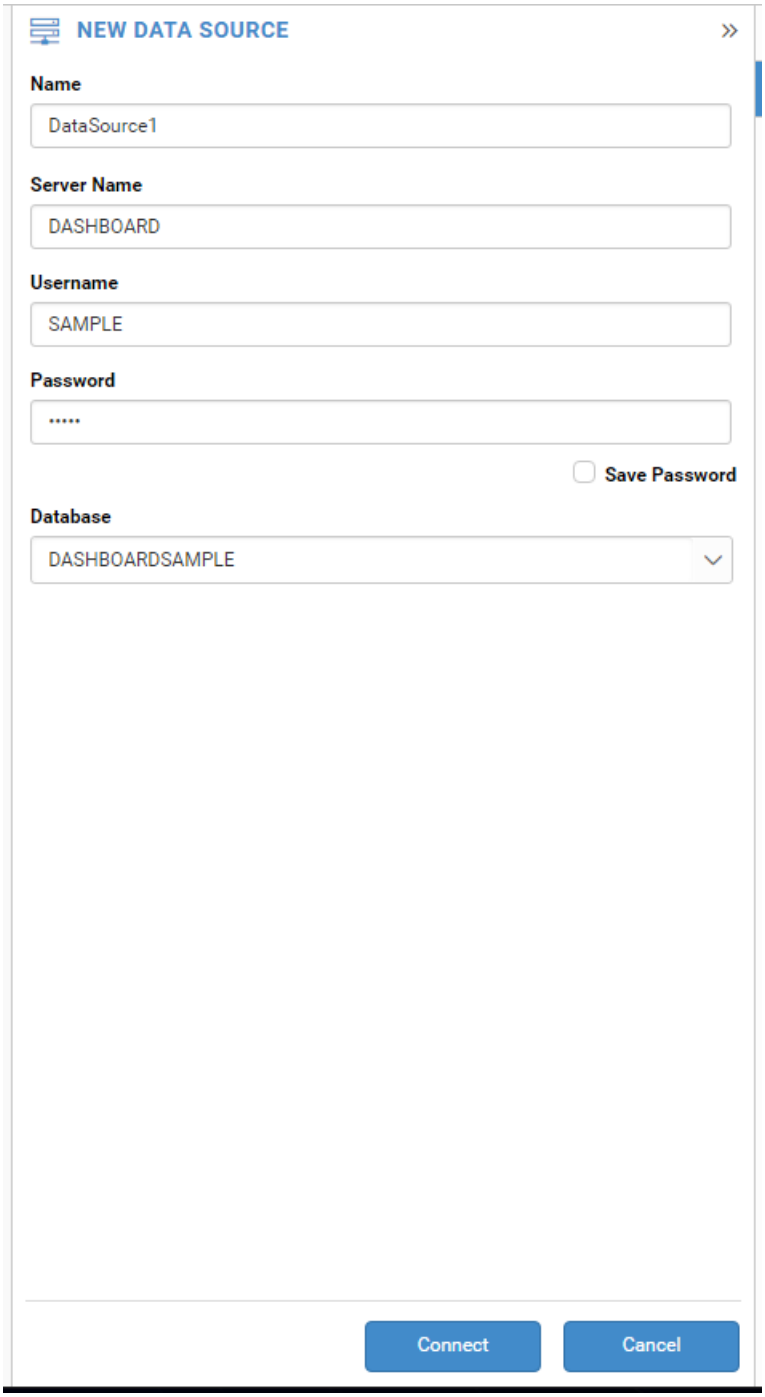
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

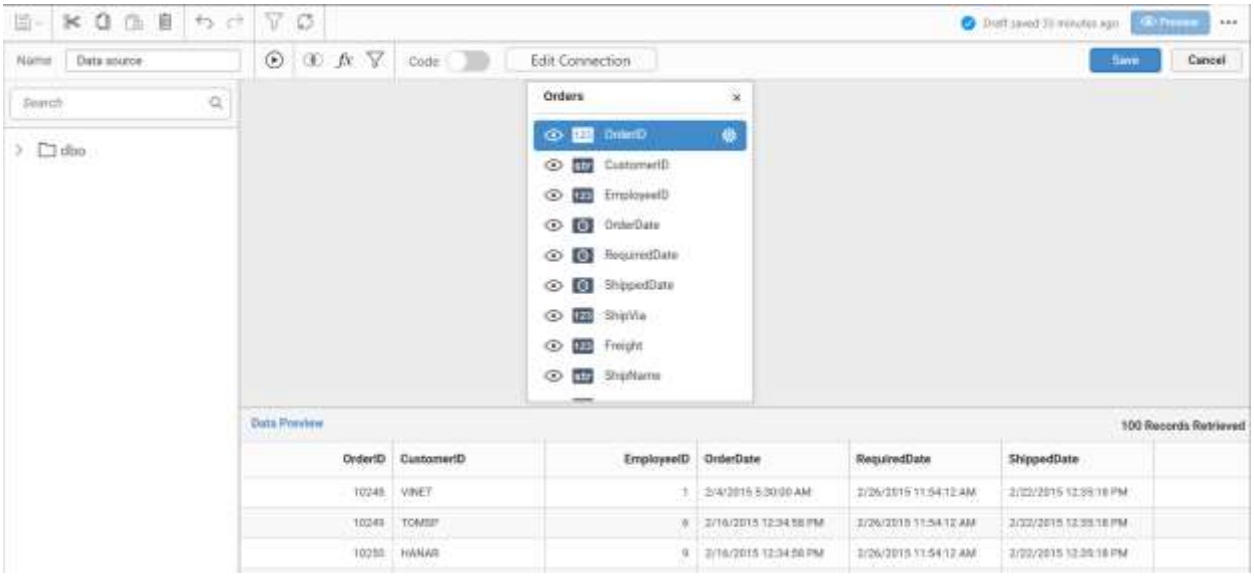
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



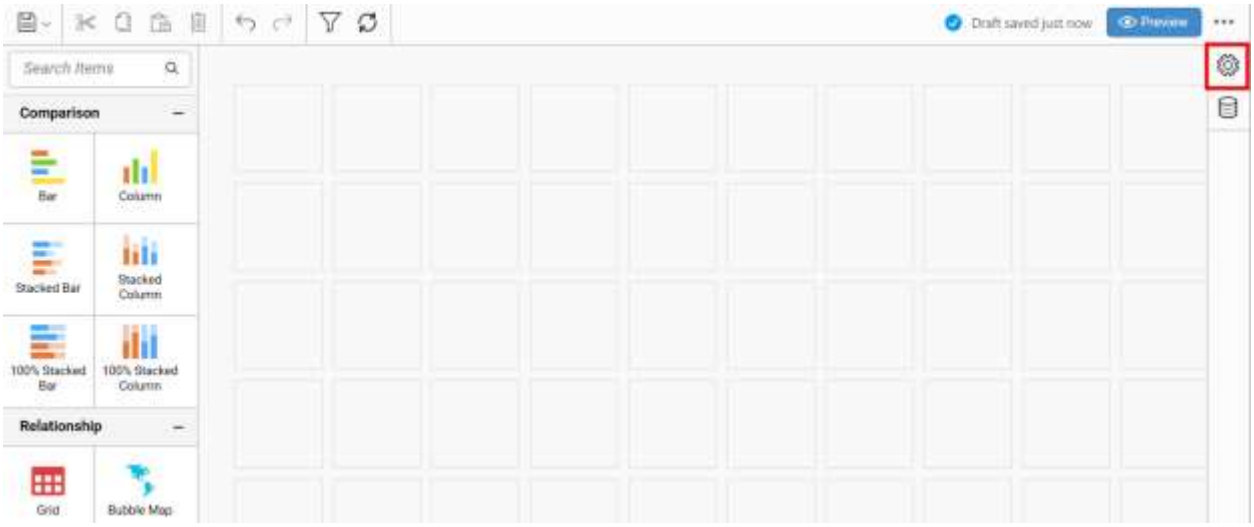
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

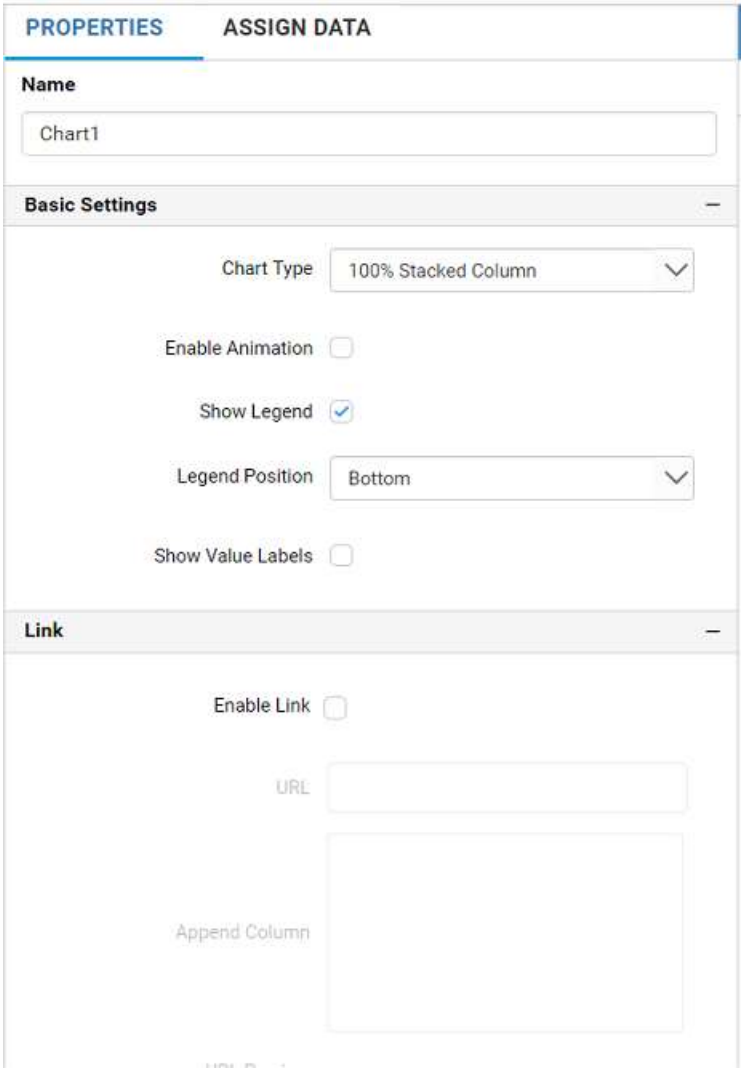


Drag your preferred table or view from the left pane from data design view, click Save button.

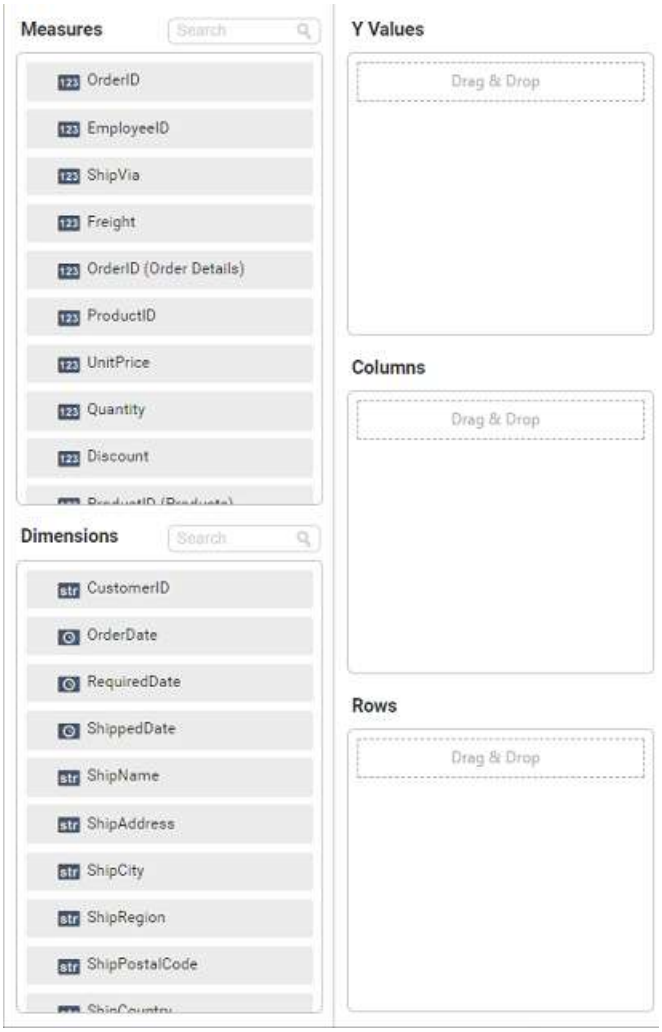


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.





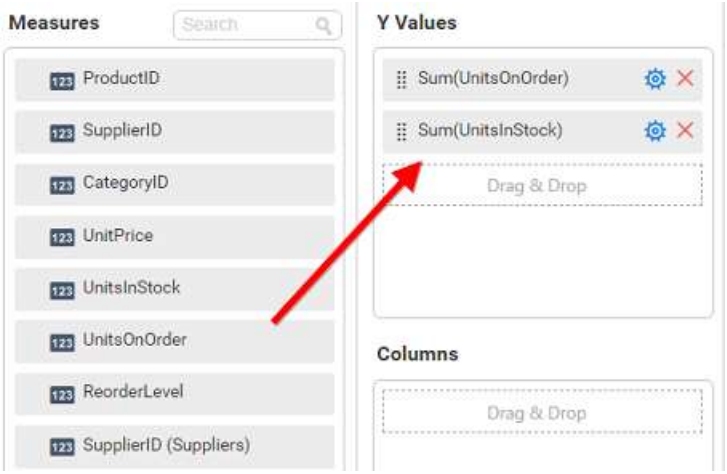
The data tab will be opened with available measures and dimensions from the connected data source



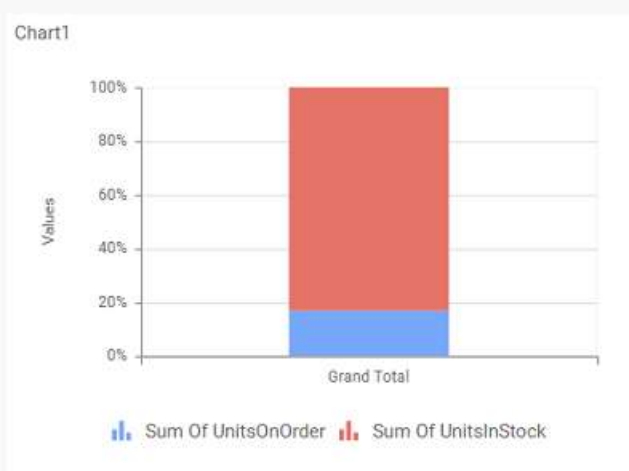
You can add the required data from Measures and Dimensions into required field.

Adding Y Values

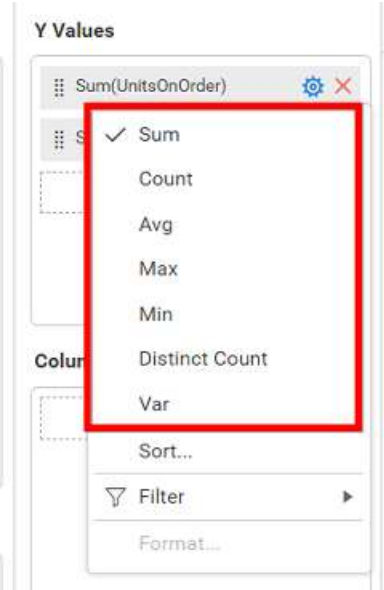
You can add more than one Measures into Y Values field by drag and drop the required measure.



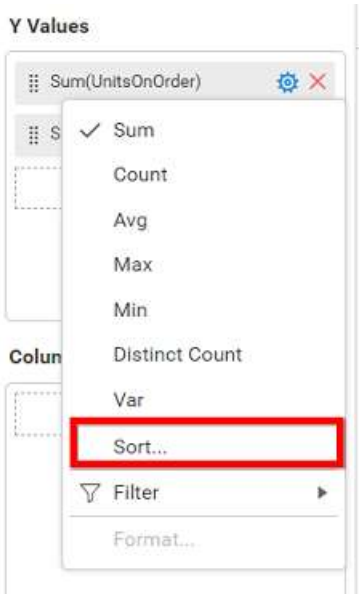
Now the 100% stacked column chart will be rendered like this



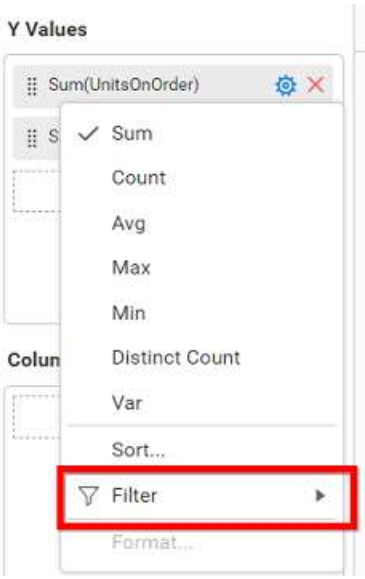
Click the Settings option to change required summary type from the available summary types shown in Settings.



You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort



You can filter the data to be displayed in chart by using filter option. For more details, refer filter.

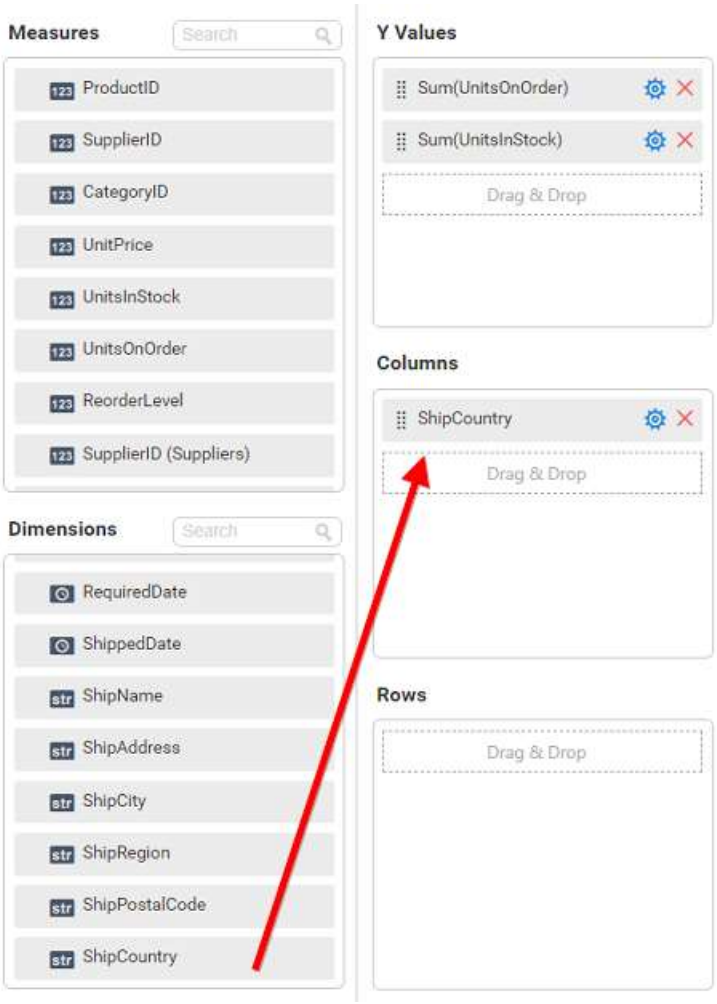


To remove the added value fields click highlighted button.

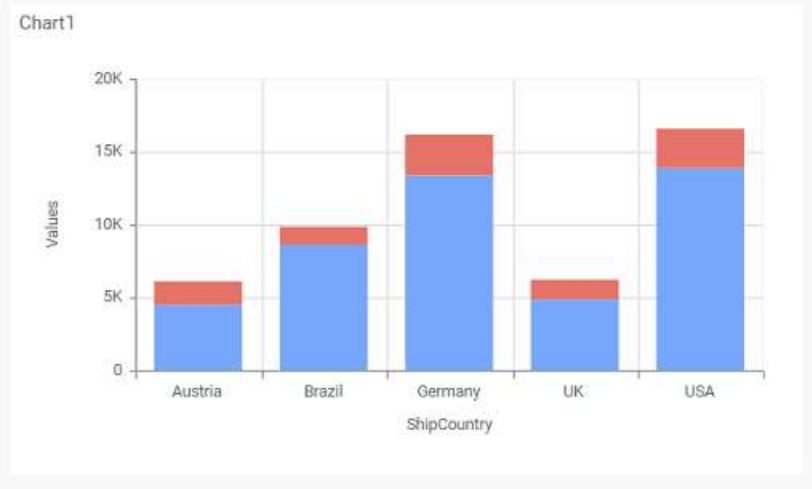
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



100% stacked column chart chart will be rendered like this



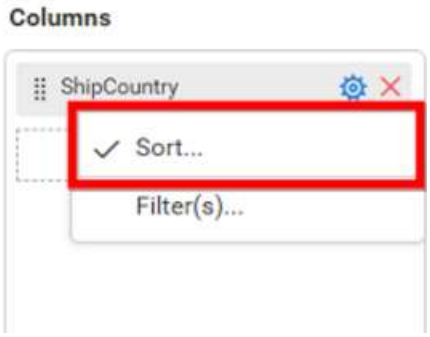
Add more than one value to Columns field, the alert message will be shown to enable the drill down option. Click Yes to enable the option.

Note: If you click No, single value will be added to the Columns field.

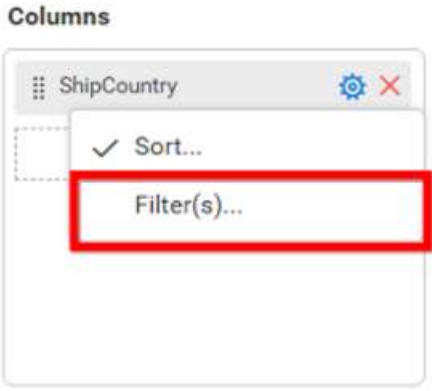
The drilled view of the chart region selected.

You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

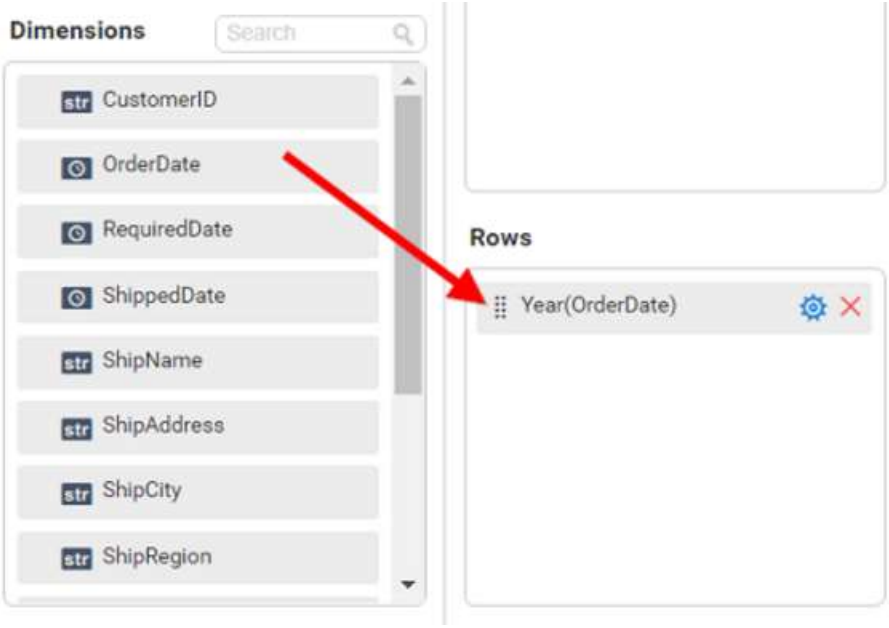


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

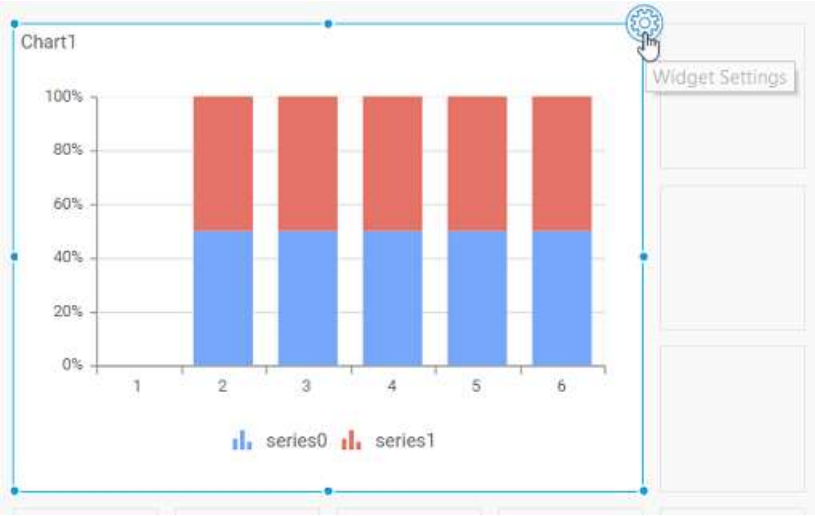
This will render chart in series.

How to format 100% stacked column chart?

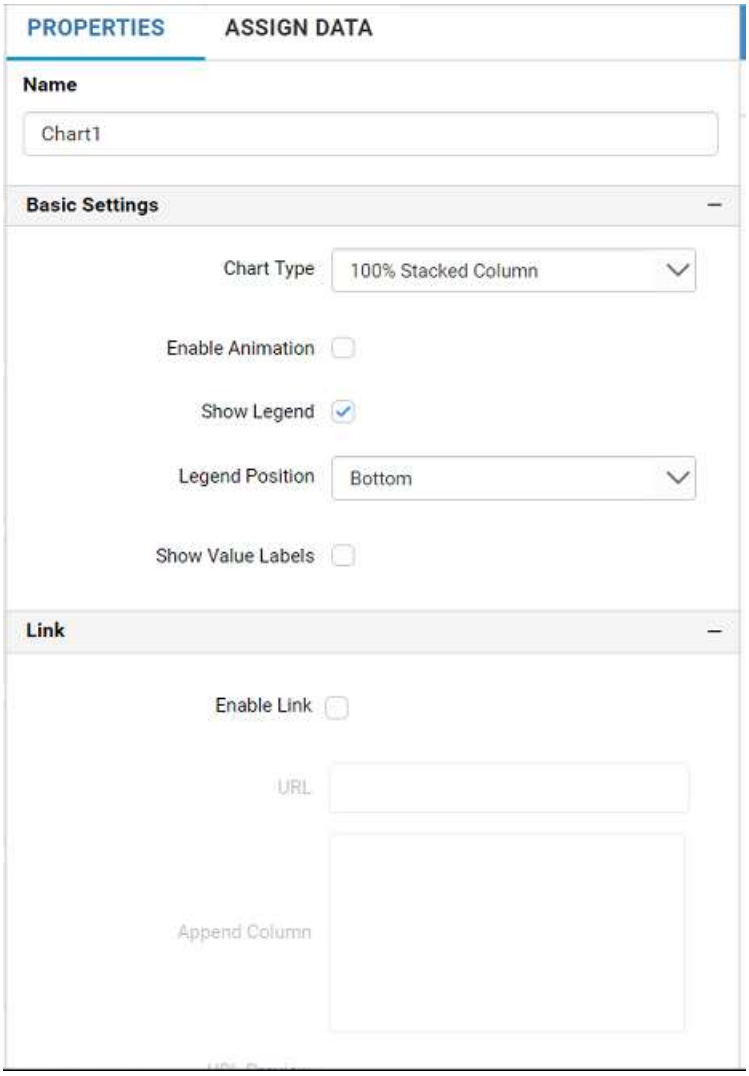
You can format the 100% stacked column chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into 100% stacked column chart follow the steps

1. Drag and drop the 100% stacked column chart into canvas and resize it to your required size.
2. Configure the data into 100% stacked column chart.
3. Focus on the 100% stacked column chart and click on widget settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this 10% stacked column chart widget

Basic Settings

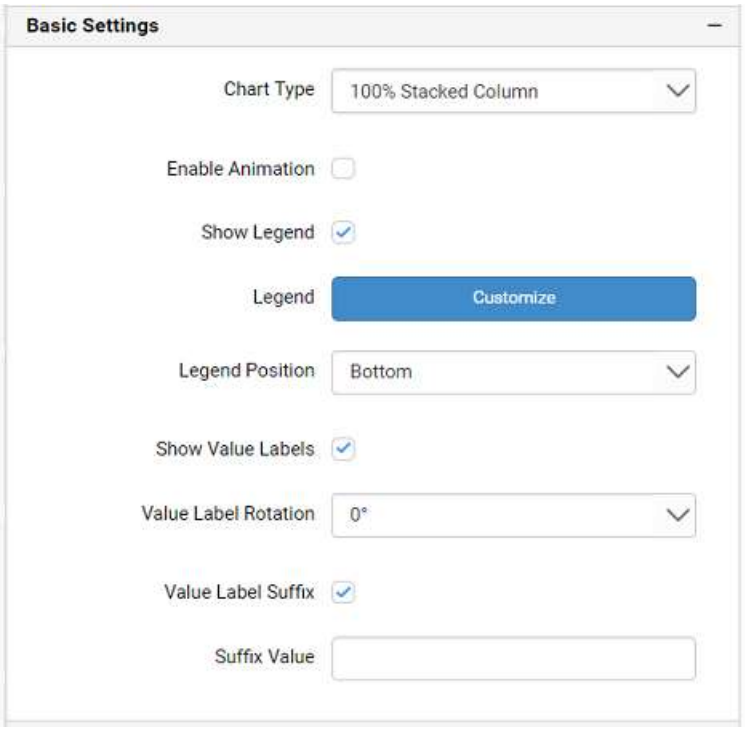


Chart Type

This allows you to switch the widget view from current chart type to another convertible chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).

Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

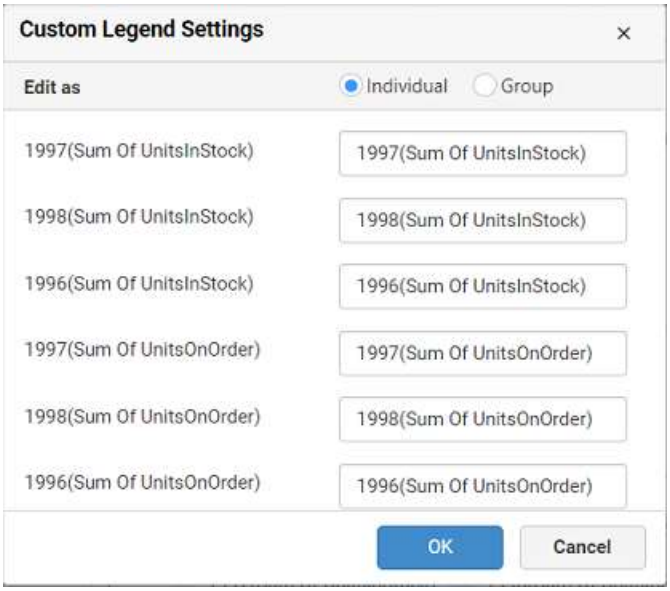
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

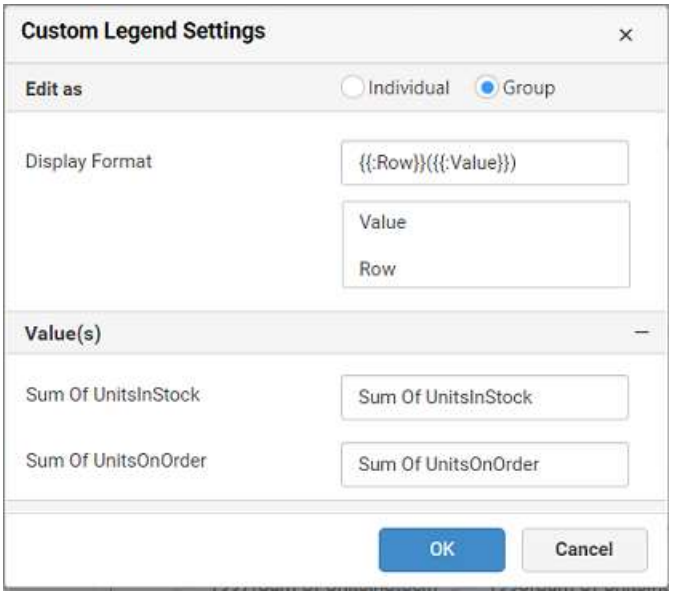
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.



Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like 1997(Sum of UnitsInStock)

Show Value Labels

This allows you to toggle the visibility of value labels.

Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

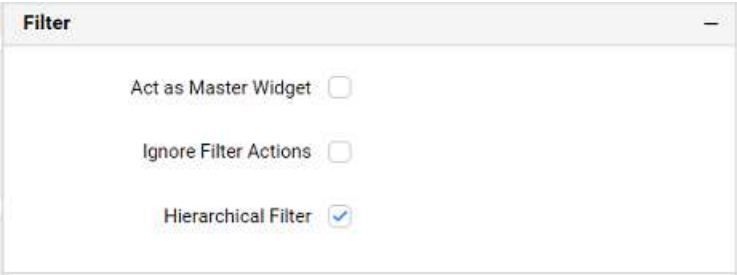
Value Label Suffix

Allows you to enable the Suffix value text to the value labels.

Suffix Value

Allows you to set\edit suffix value to the value labels.

Filter



Act as Master Widget

This allows you to define this 100% stacked column chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

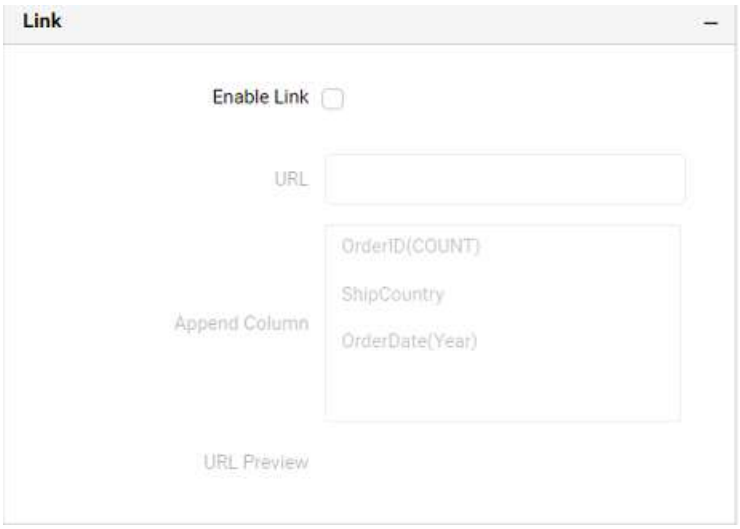
This allows you to define this 100% stacked column chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

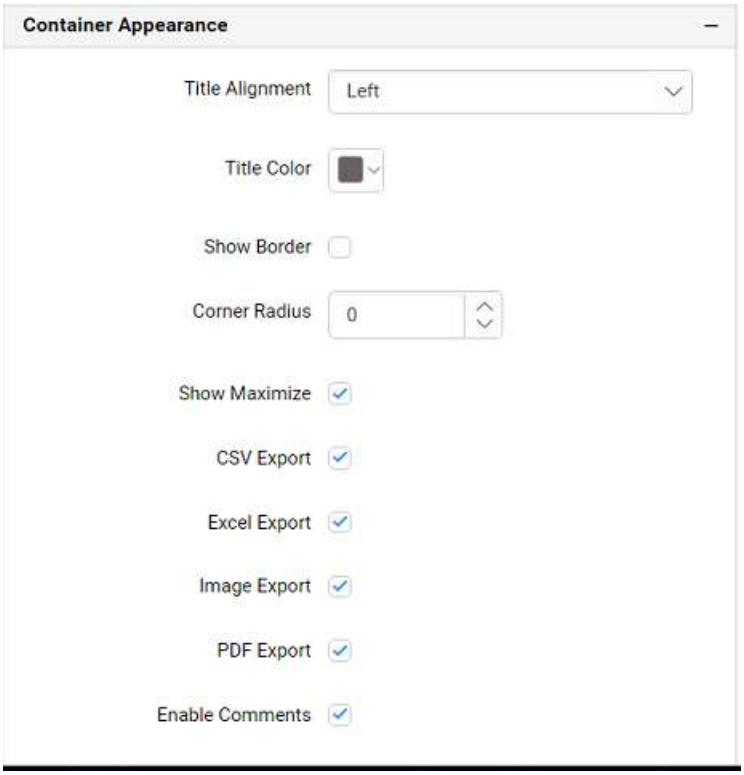
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this 100% stacked column chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this 100% stacked column widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this 100% stacked column widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

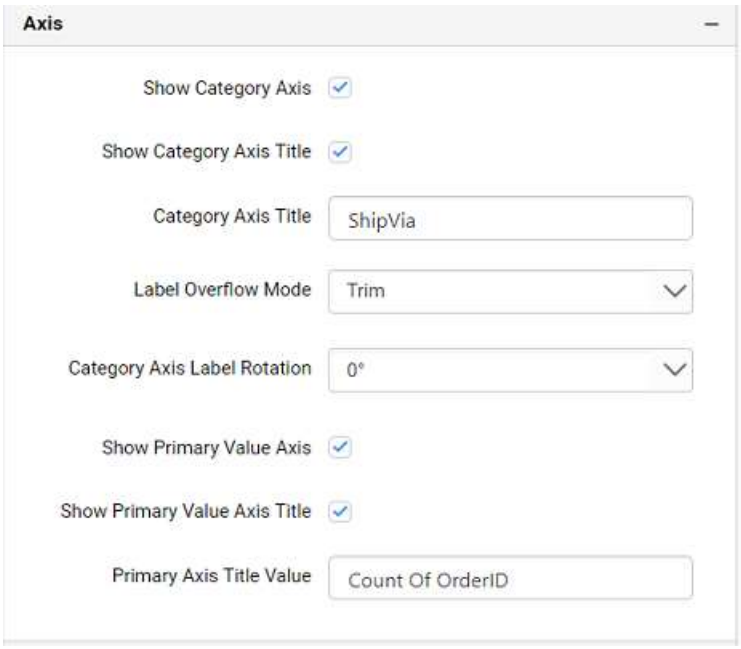
Image Export

This allows you to enable/disable the image export option for this 100% stacked column chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

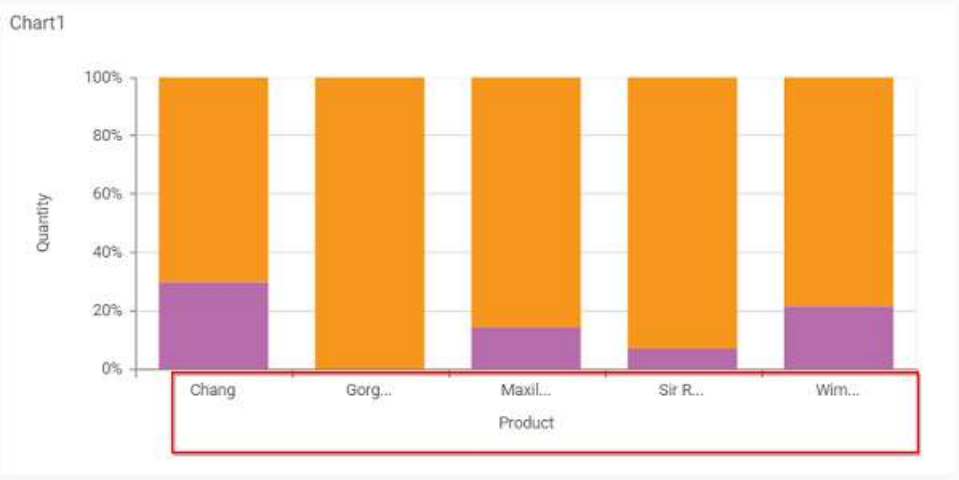
Axis



This section allows you to customize the axis settings in chart.

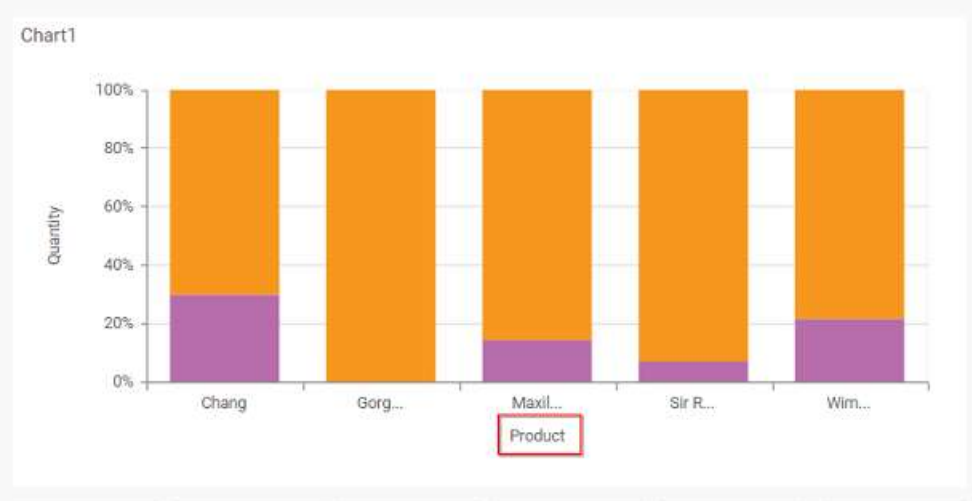
Show Category Axis

This allows to enable the visibility of Category Axis.



Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

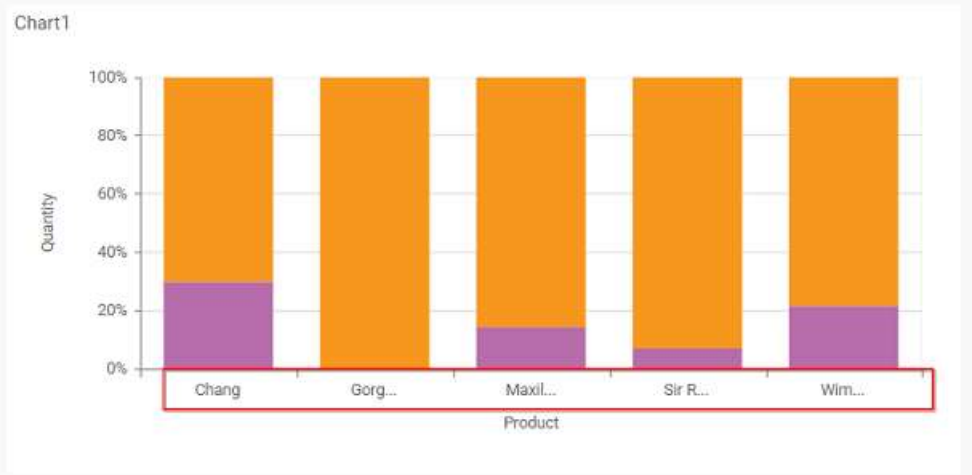
This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

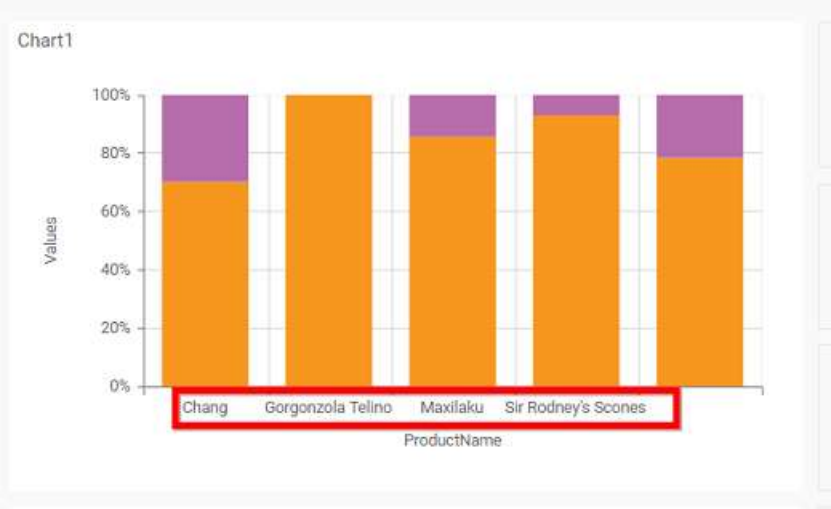
Trim

This option trims the end of overlapping label in the axis.



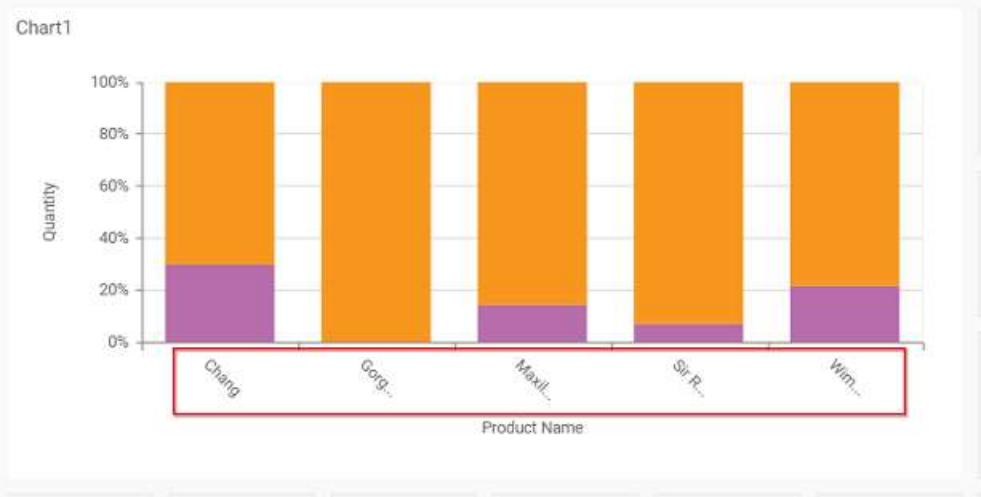
Hide

This option hides the overlapping label in the axis.



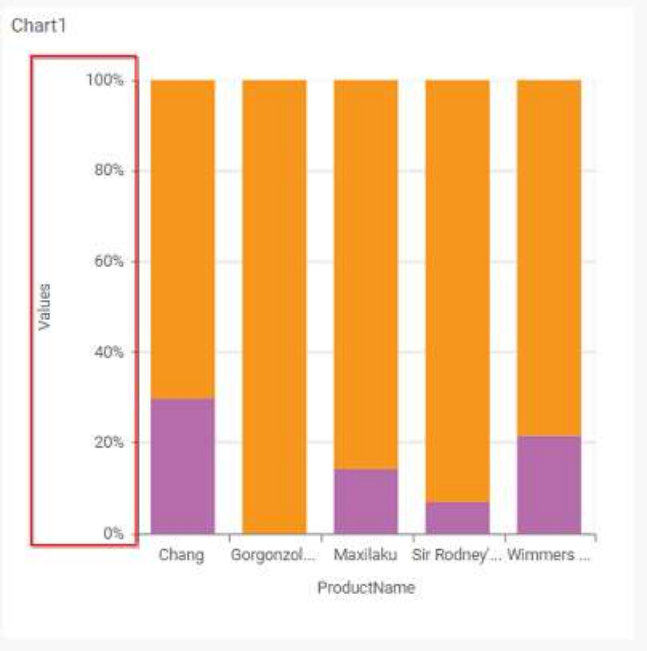
Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.

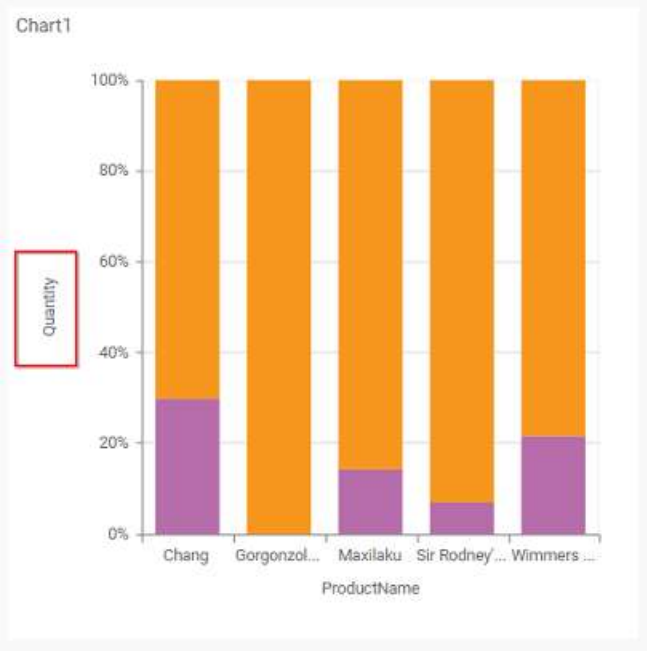


Show Primary Value Axis Title

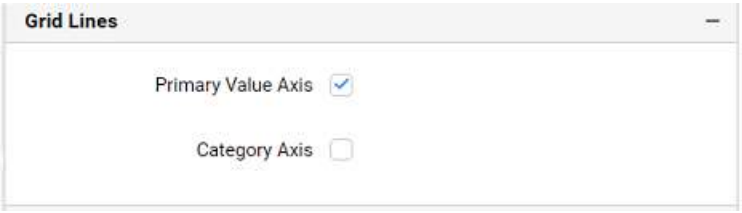
This allows you to enable the visibility of Primary Value Axis title of chart.

Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

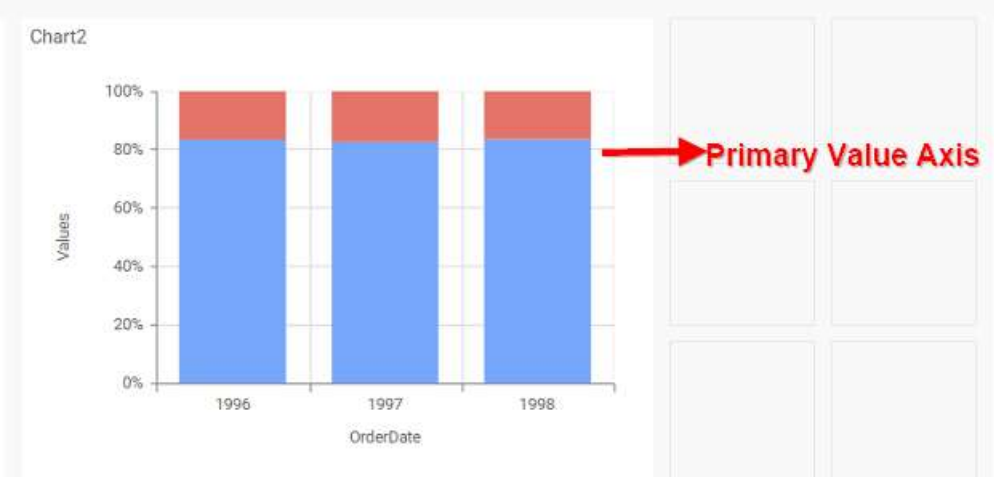


Grid Lines



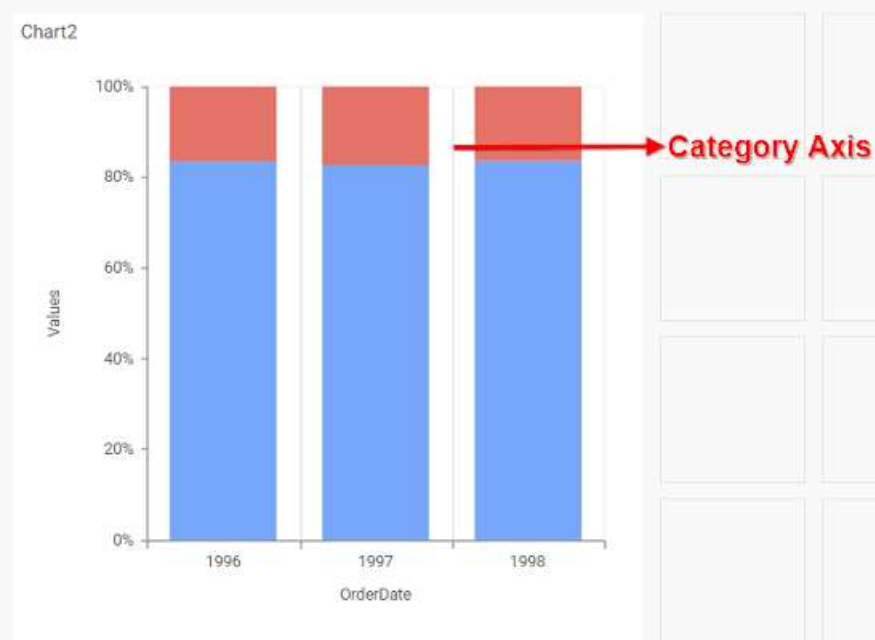
Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the 100% stacked column chart.



Category Axis

This allows you to enable the Category Axis gridlines for the 100% stacked column chart.

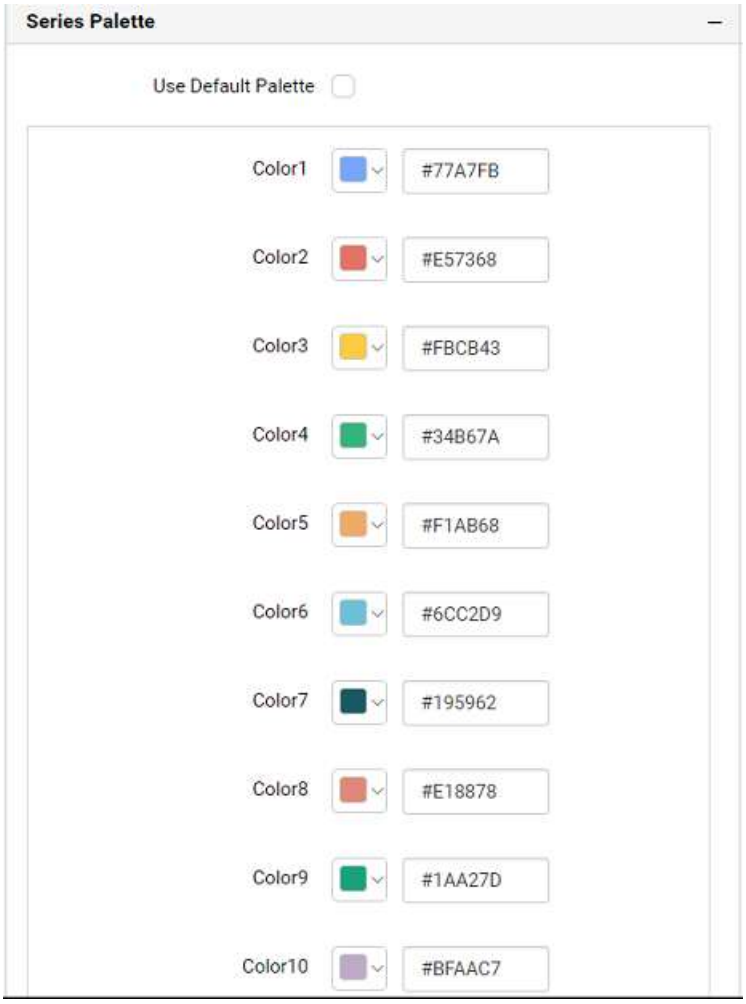


Series Palette

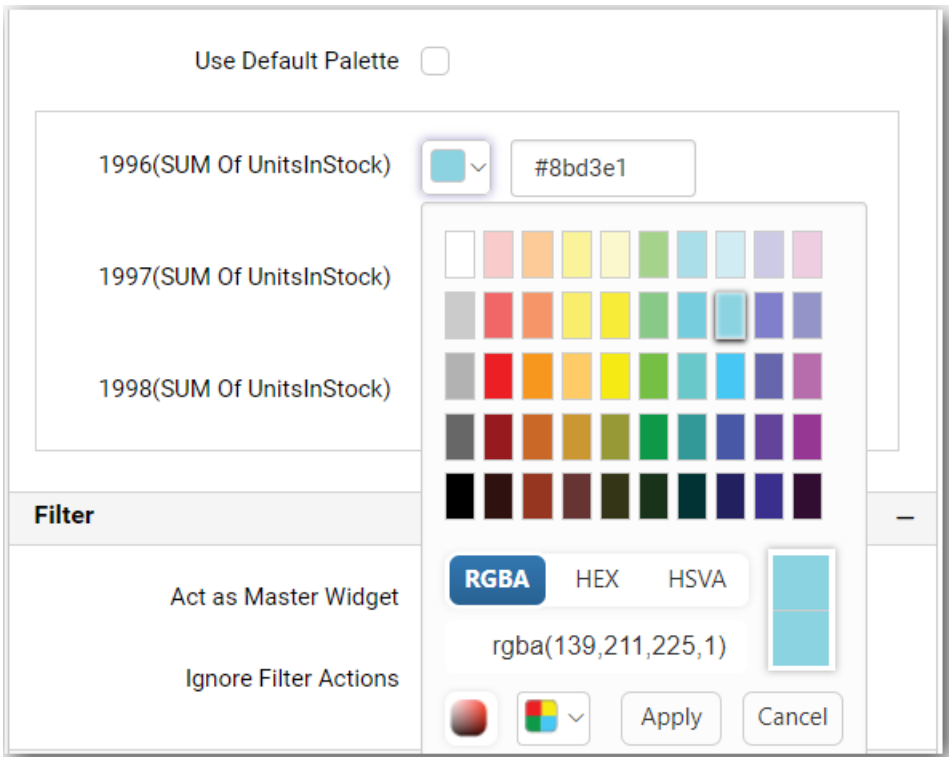
This allows you to customize the chart series color through Series Palette section.

Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.

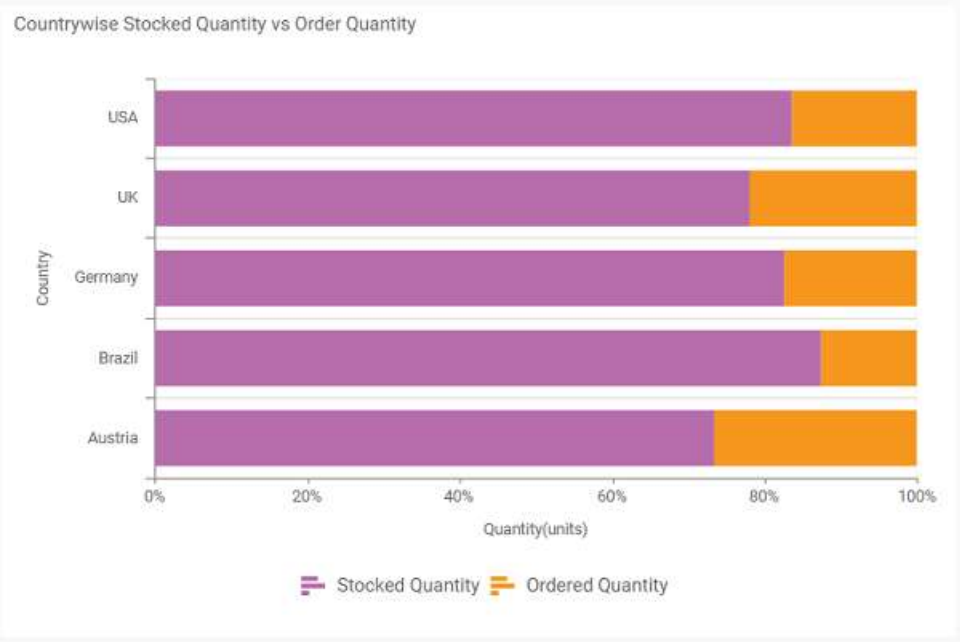


By toggle off the Use Default Palette you can customize the series colors. This section shows, list of series labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.



100% Stacked Bar Chart

100% Stacked Bar Chart allows you to compare multiple measures through bars stacked one after the other horizontally.

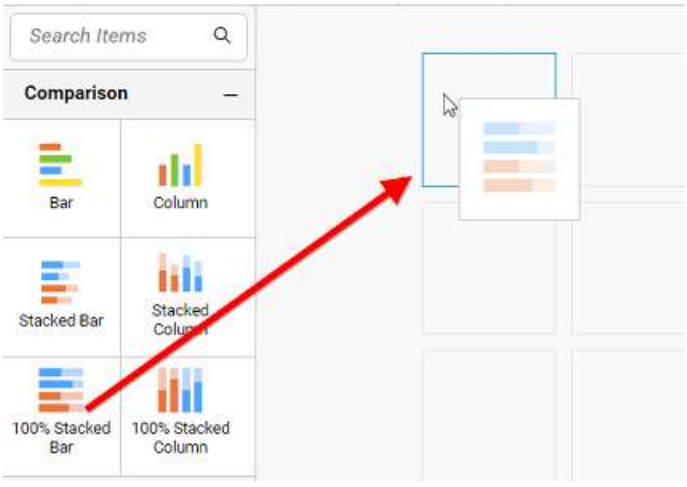


How to configure the table data to 100% stacked bar chart?

100% Stacked Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

Follow the steps to configure data to 100% stacked bar chart.

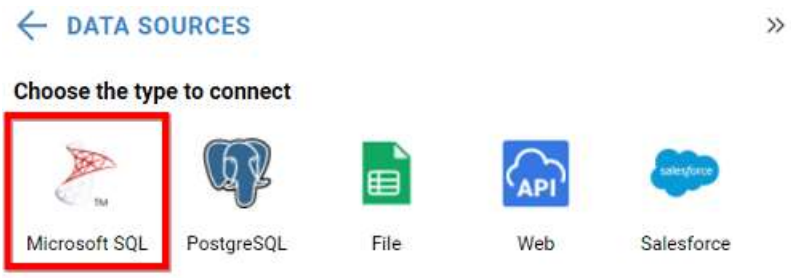
Drag and drop the 100% stacked bar chart into canvas and resize it your required size.



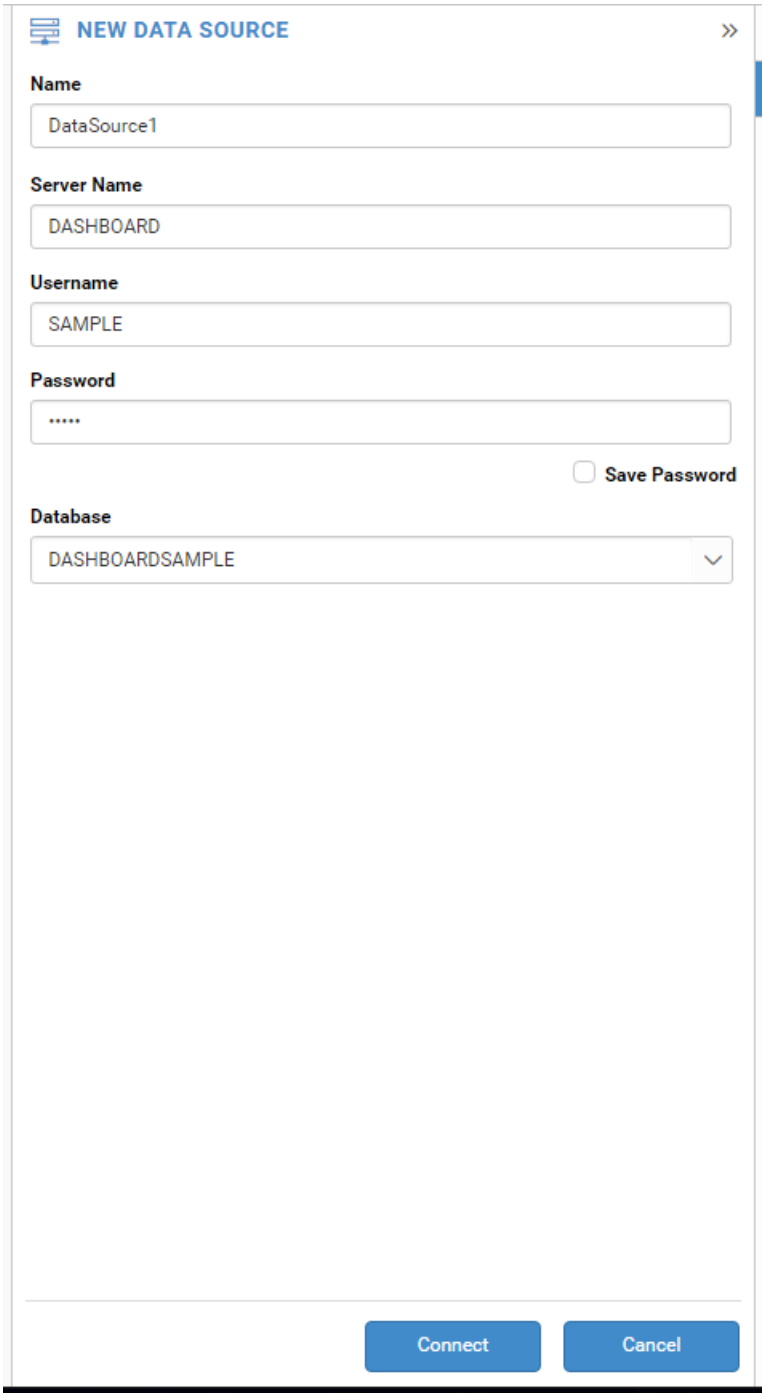
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

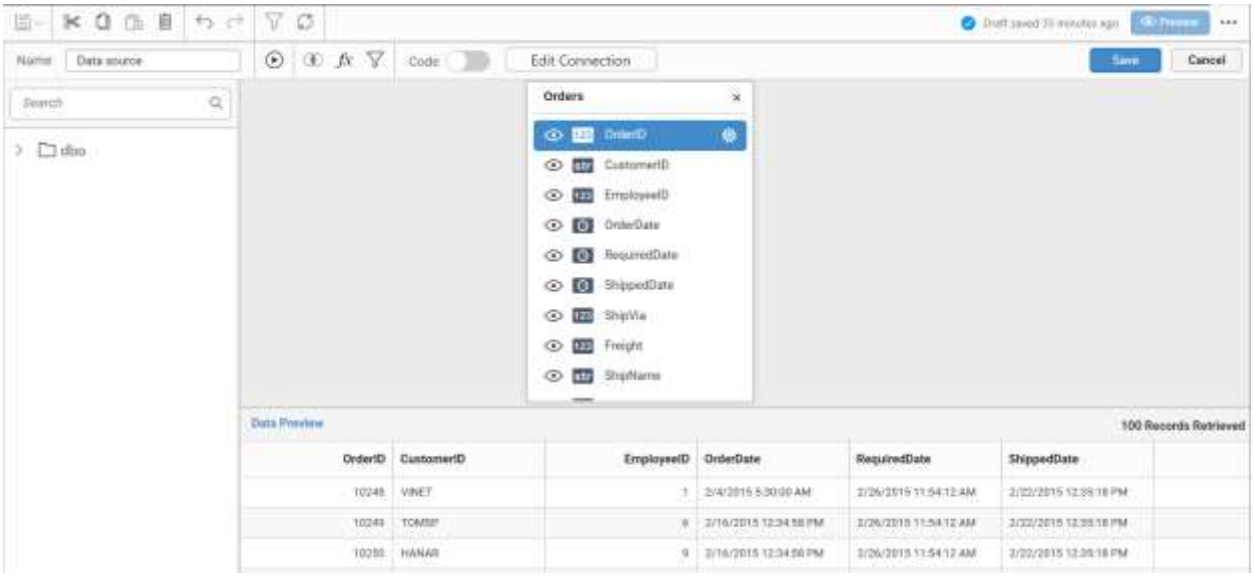
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



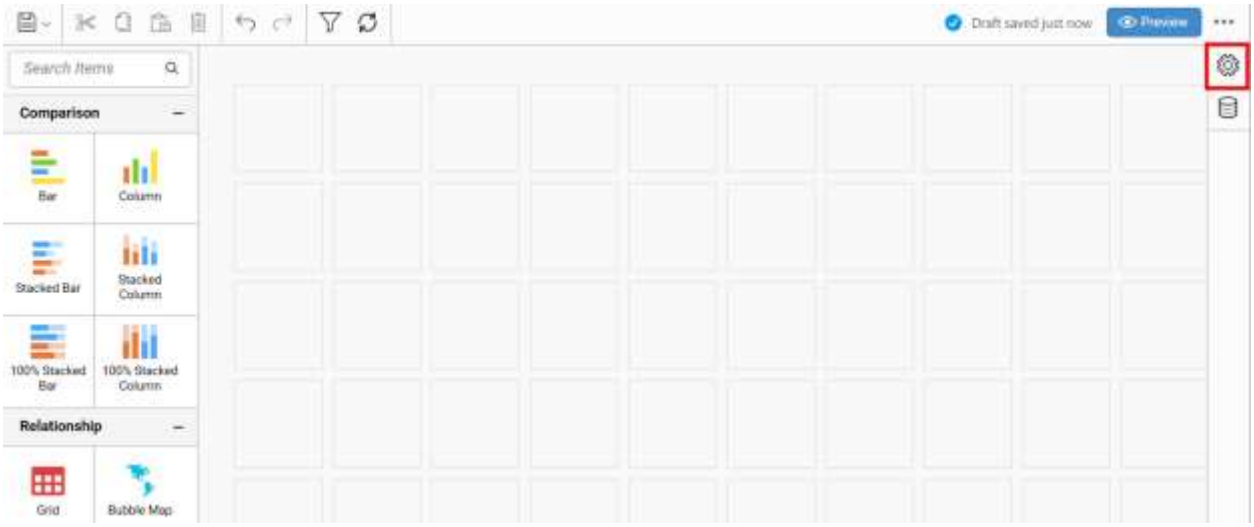
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

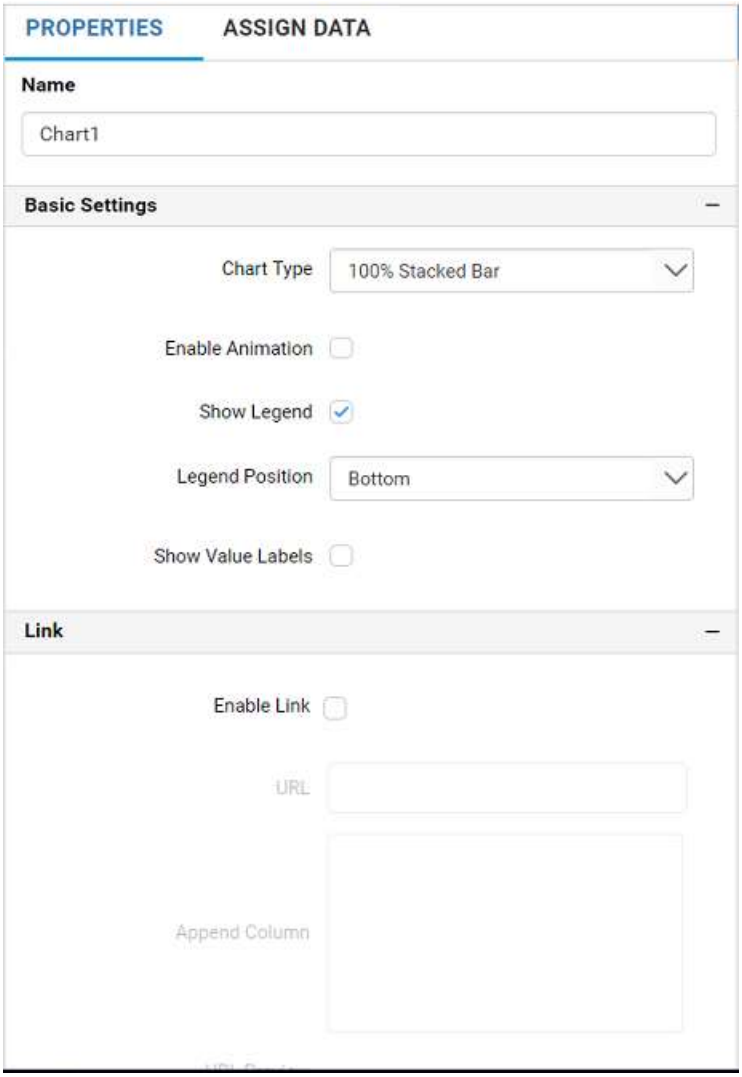


Drag your preferred table or view from the left pane from data design view, click Save button.

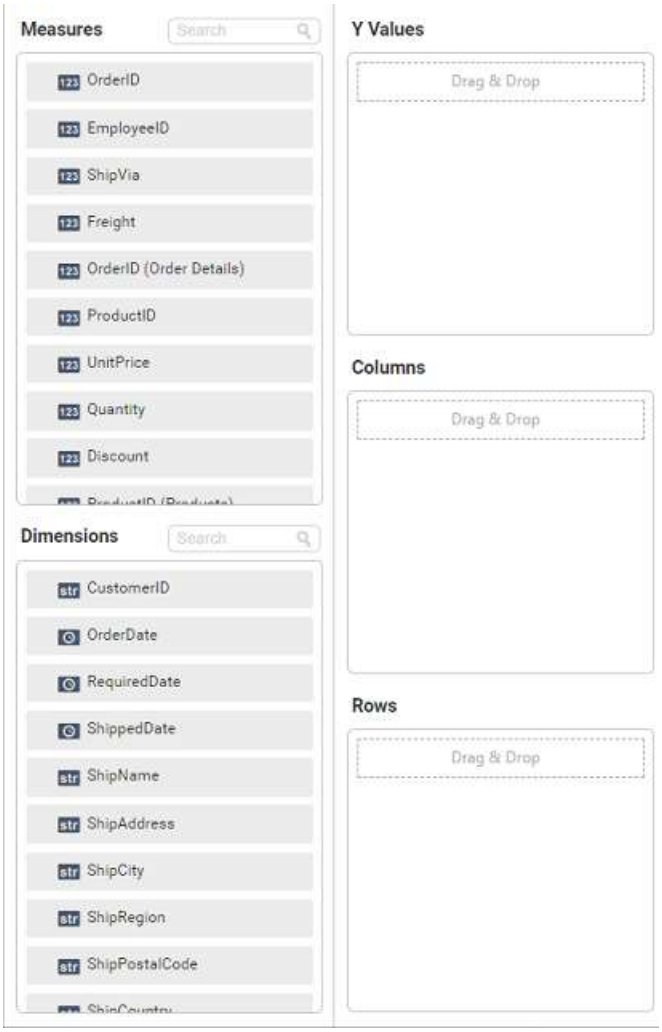


Click Properties button in configuration panel, property pane opens. Now, switch to ASSIGN DATA tab.





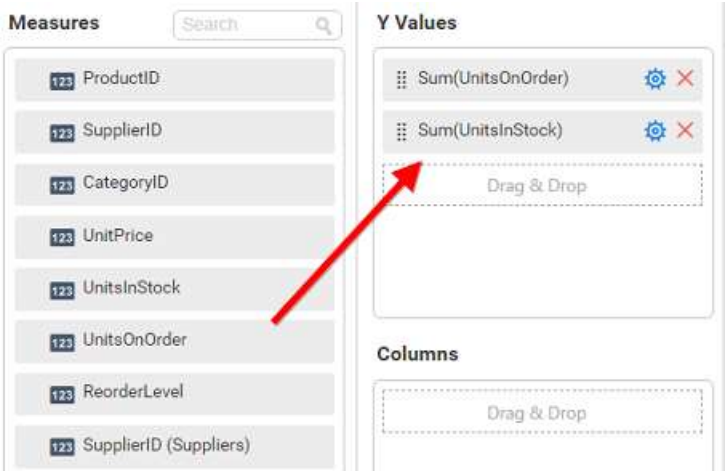
The data tab will be opened with available measures and dimensions from the connected data source



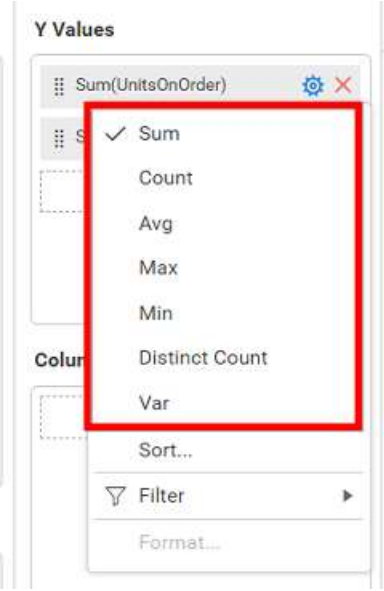
You can add the required data from Measures and Dimensions into required field.

Adding Y Values

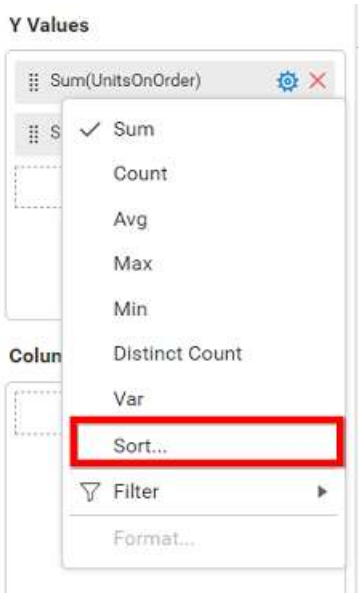
You can add more than one Measures into Y Values field by drag and drop the required measure.



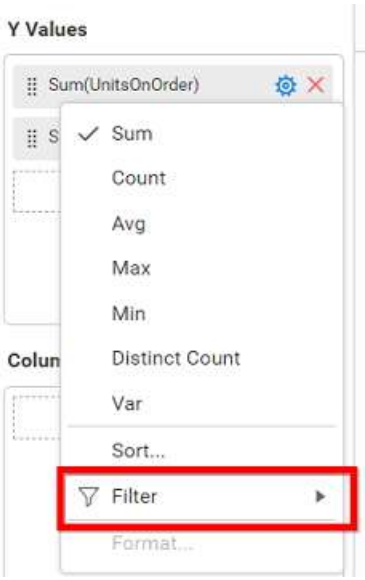
Click the Settings option to change required summary type from the available summary types shown in Settings.



You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort



You can filter the data to be displayed in chart by using filter option. For more details, refer filter.

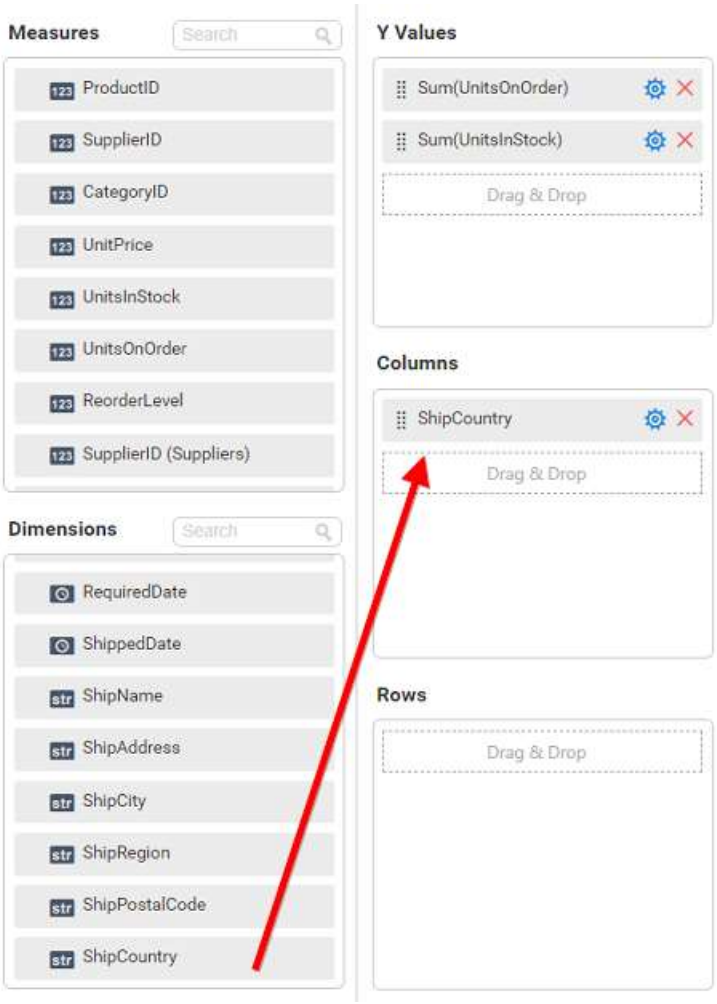


To remove the added value fields click highlighted button.

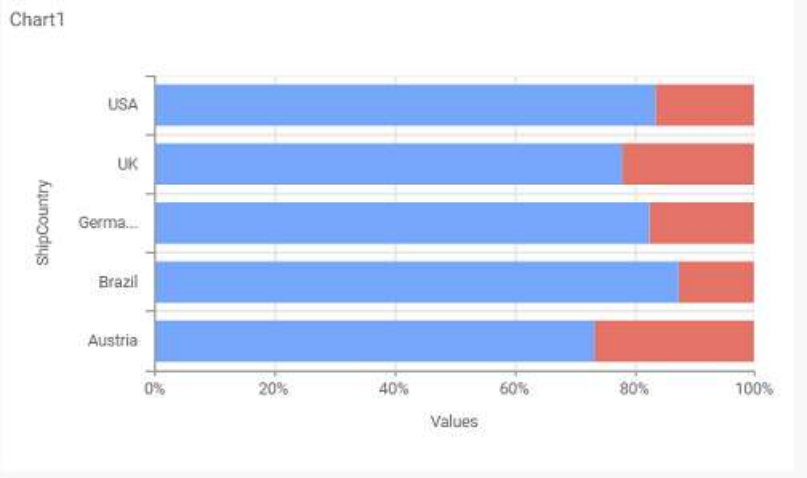
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



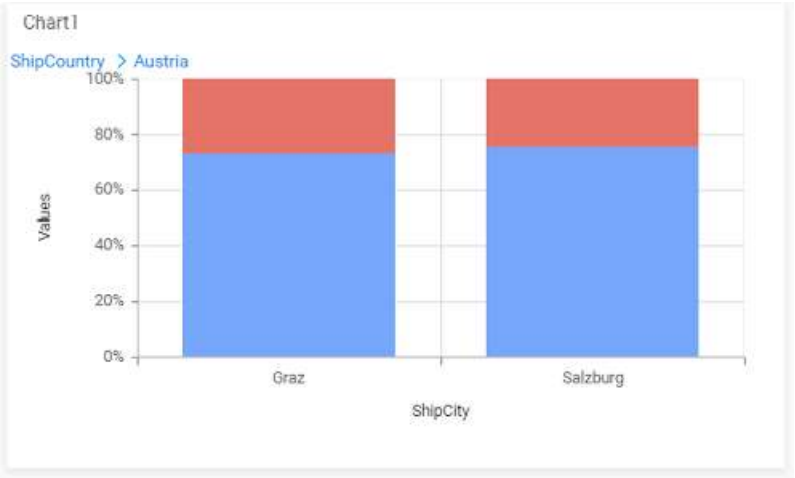
100% stacked bar chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

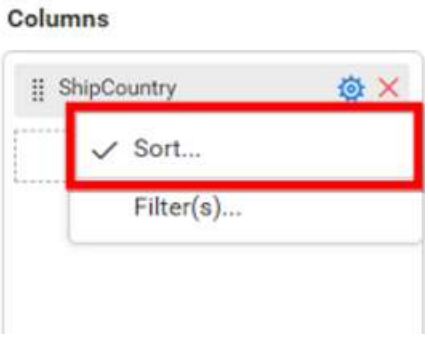
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.

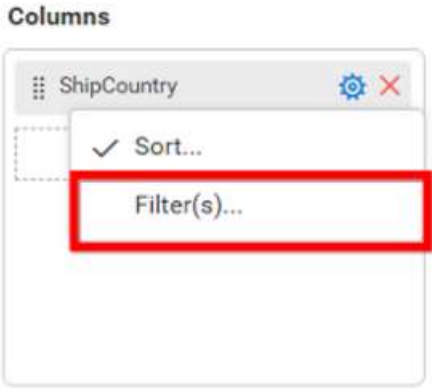


You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

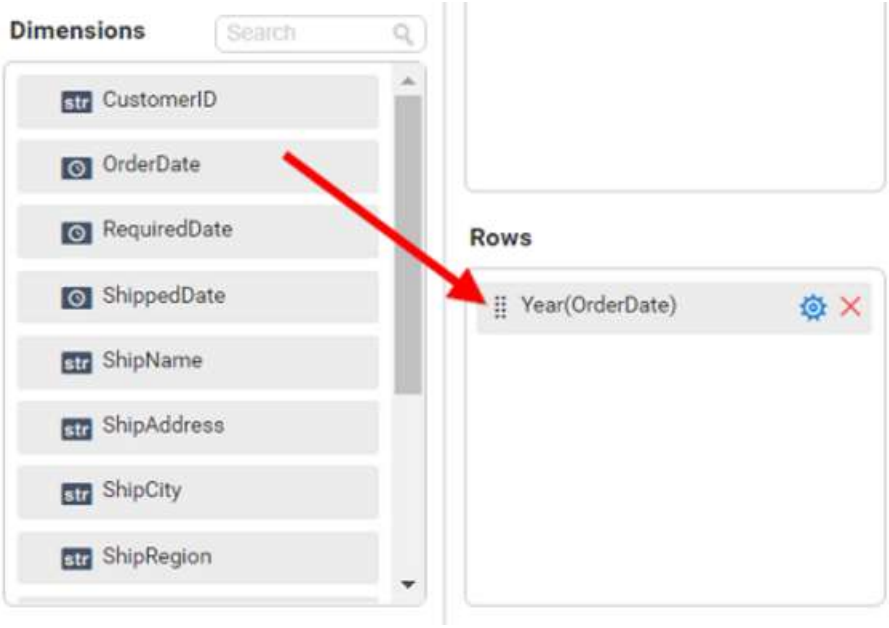


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

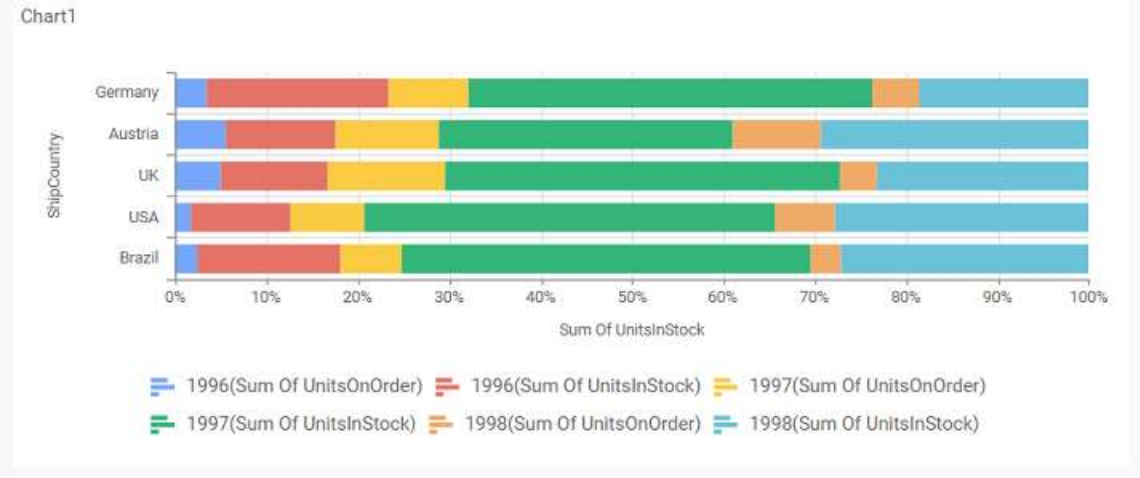
Adding Rows

You can drag and drop the Measure or Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render chart in series.

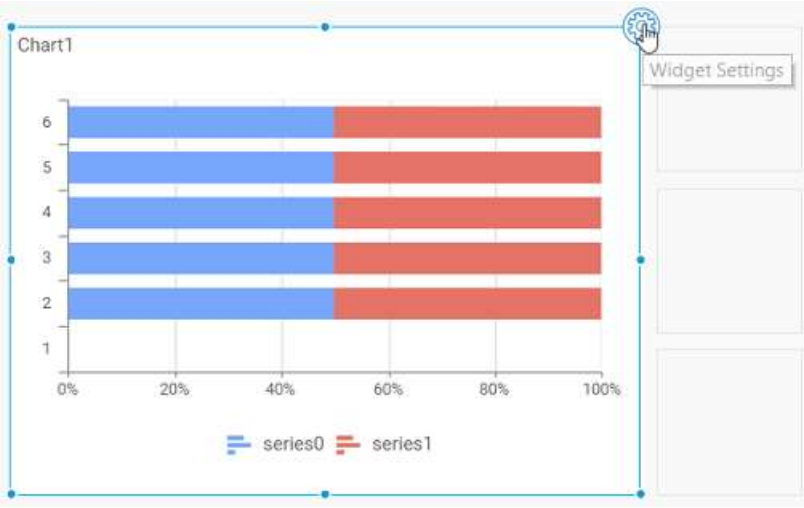


How to format 100% stacked bar chart?

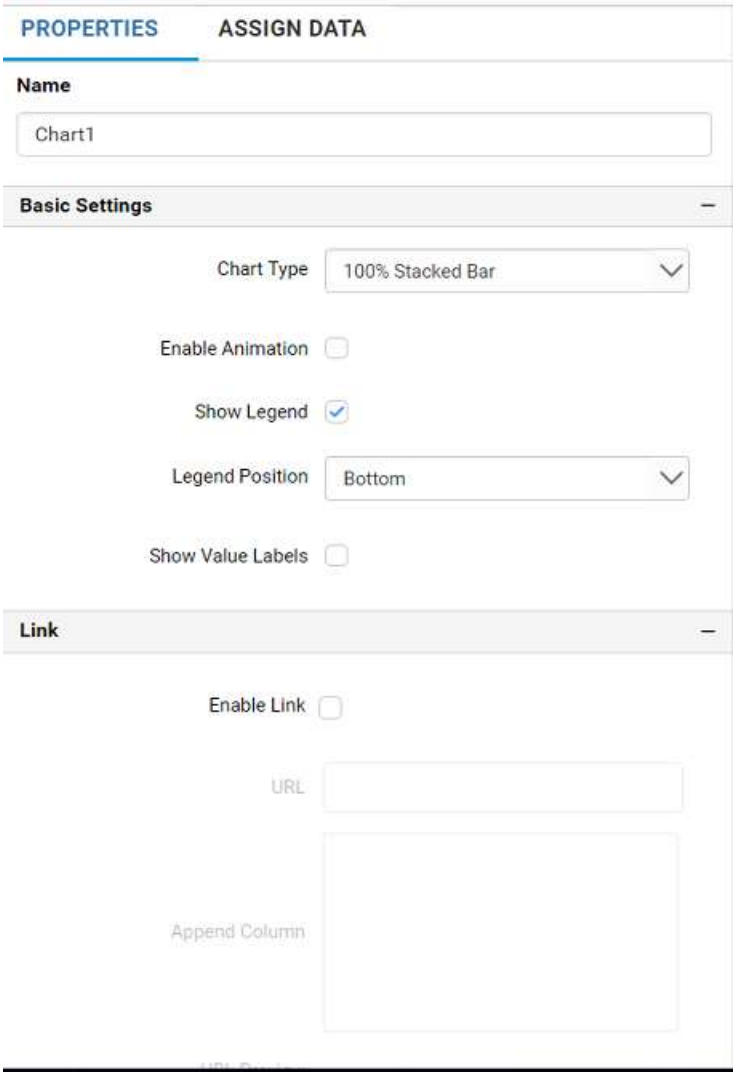
You can format the 100% stacked bar chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into 100% stacked bar chart follow the steps

1. Drag and drop the 100% stacked bar chart into canvas and resize it to your required size.
2. Configure the data into 100% stacked bar chart.
3. Focus on the 100% stacked bar chart and click on widget settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this 100% stacked bar chart widget

Basic Settings

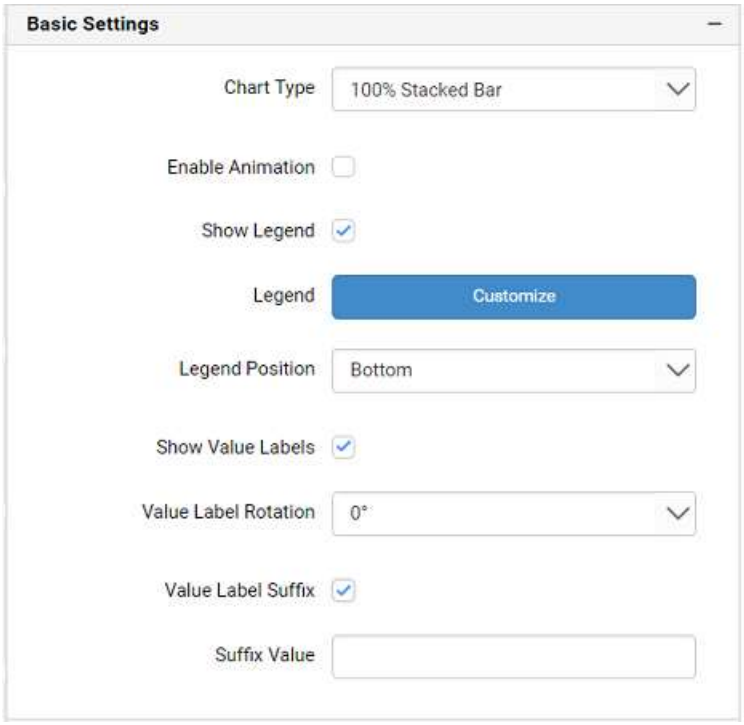


Chart Type

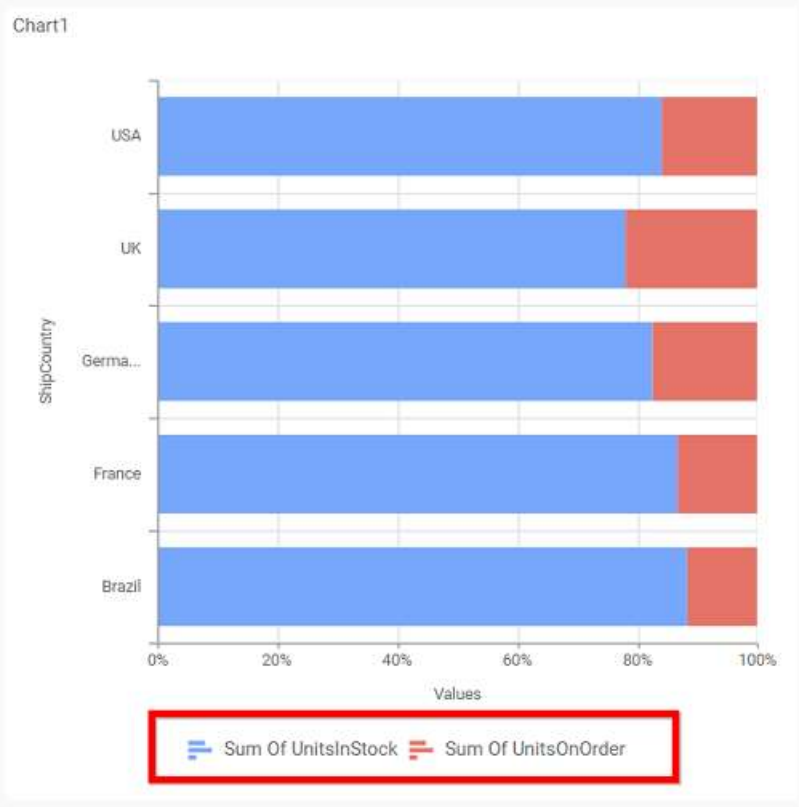
This allows you to switch the widget view from current chart type to another convertible chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

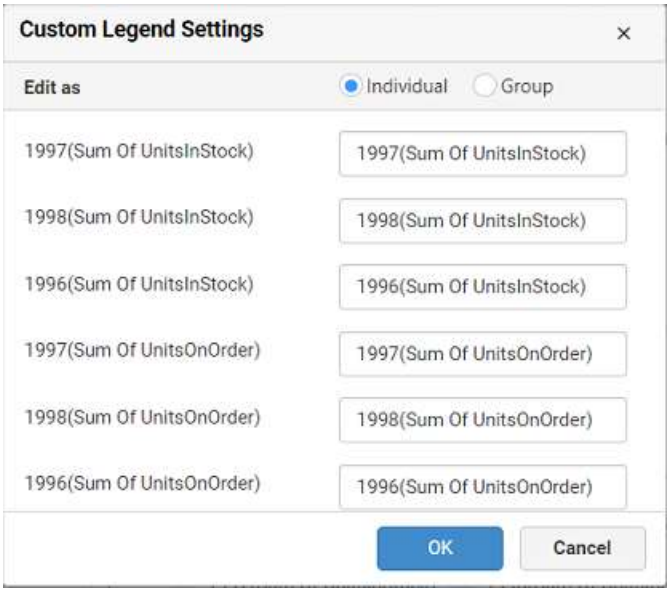
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

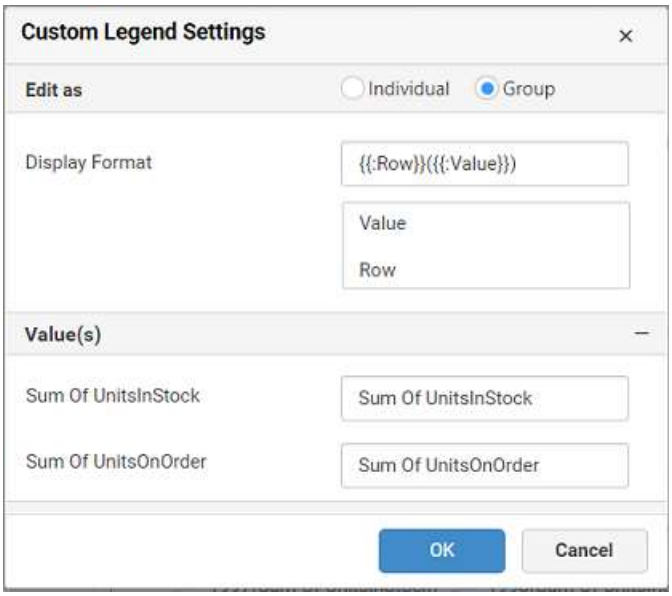
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.

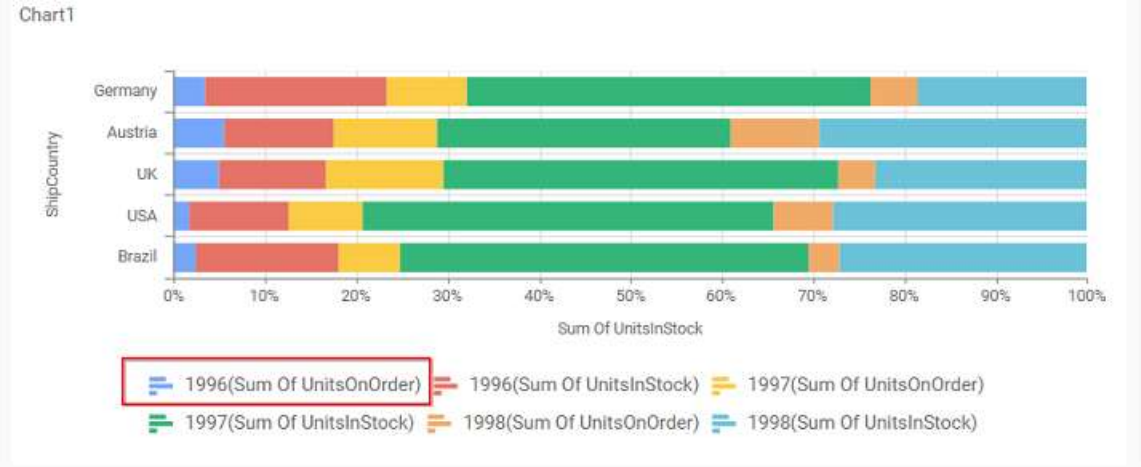


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

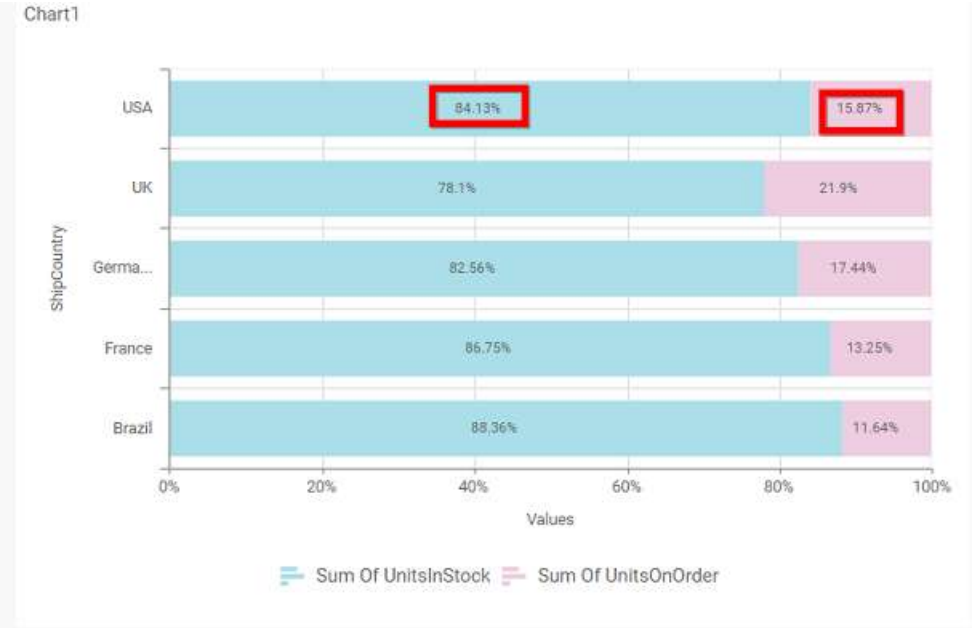


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like 1996(Sum of UnitsInOrder)



Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

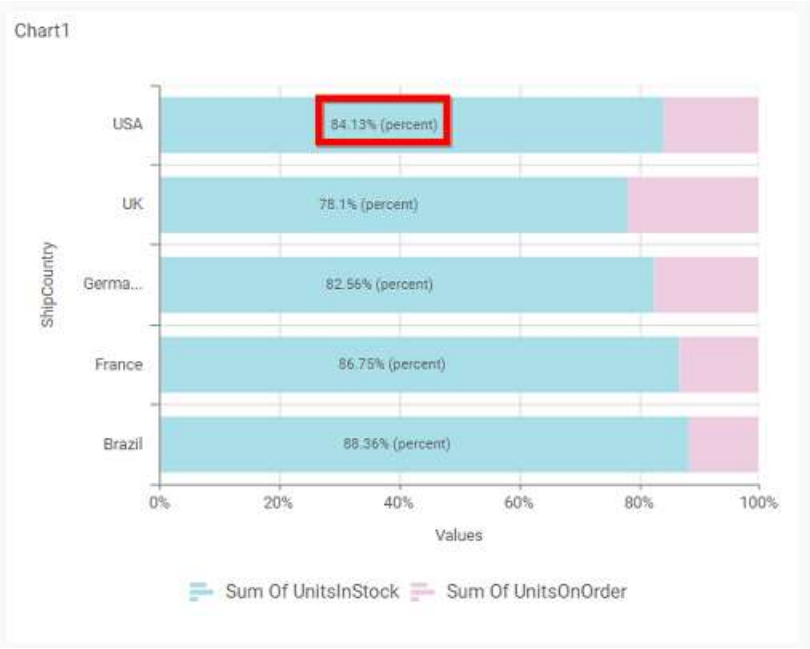
This allows you to define the rotation angle for the value labels to display.

Value Label Suffix

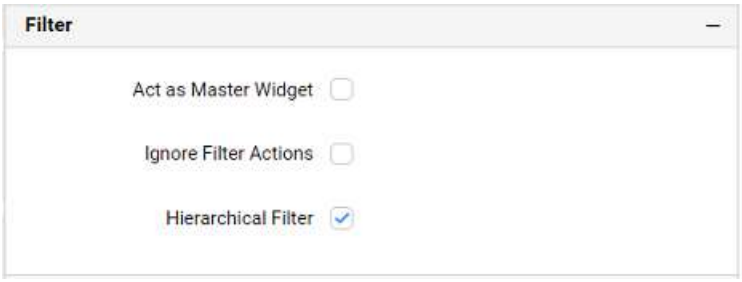
Allows you to enable the Suffix value text to the value labels.

Suffix Value

Allows you to set\edit suffix value to the value labels.



Filter



Act as Master Widget

This allows you to define this 100% stacked bar chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

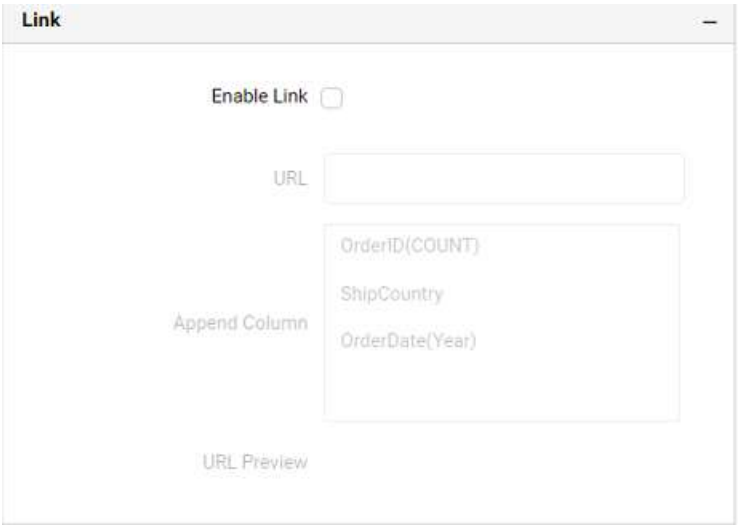
This allows you to define this 100% stacked bar chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

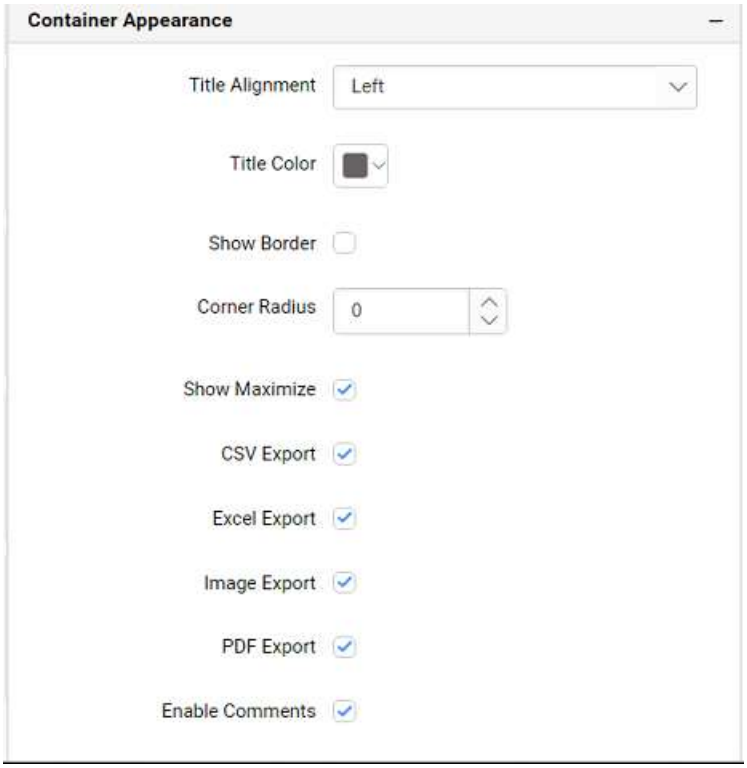
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this 100% stacked bar chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this 100% stacked bar chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this 100% stacked bar chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

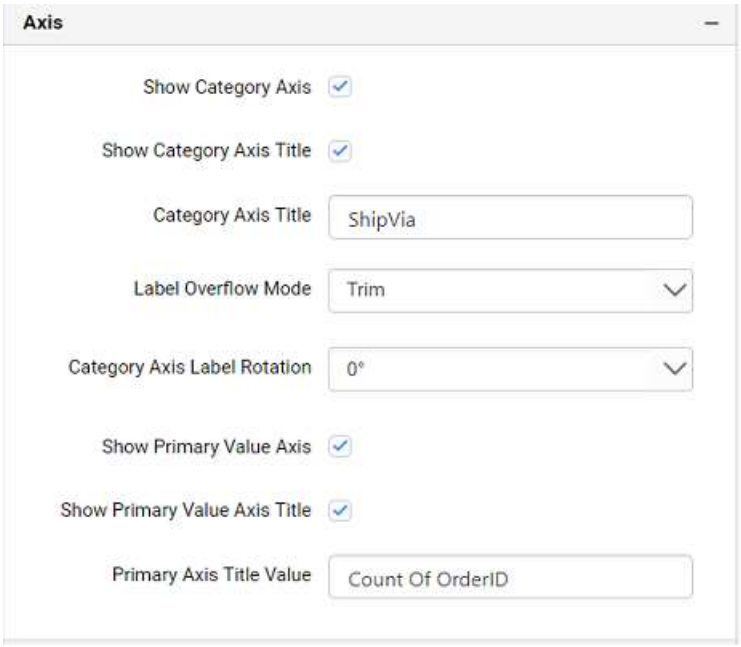
Image Export

This allows you to enable/disable the image export option for this 100% stacked bar chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

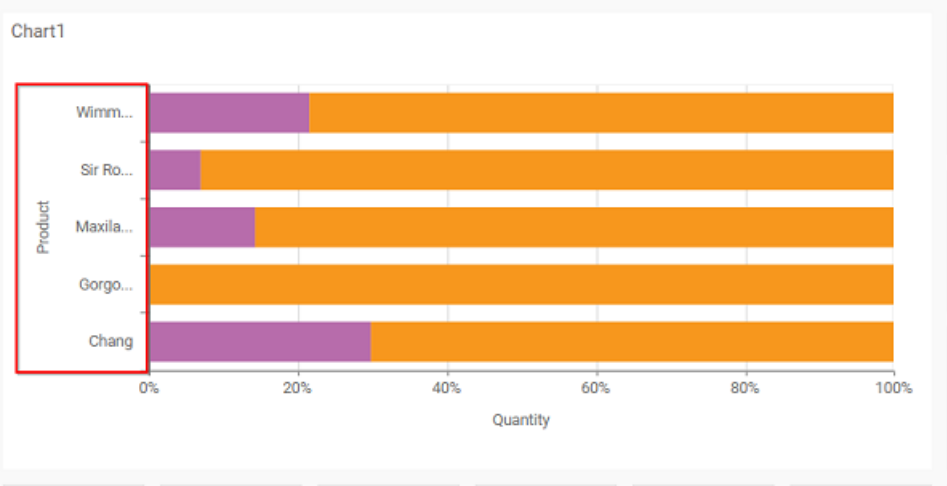
Axis



This section allows you to customize the axis settings in chart.

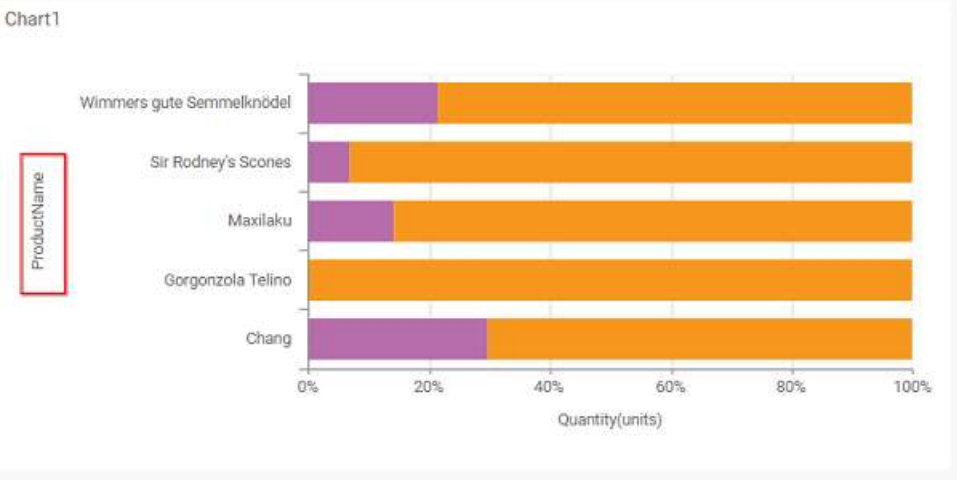
Show Category Axis

This allows to enable the visibility of Category Axis.



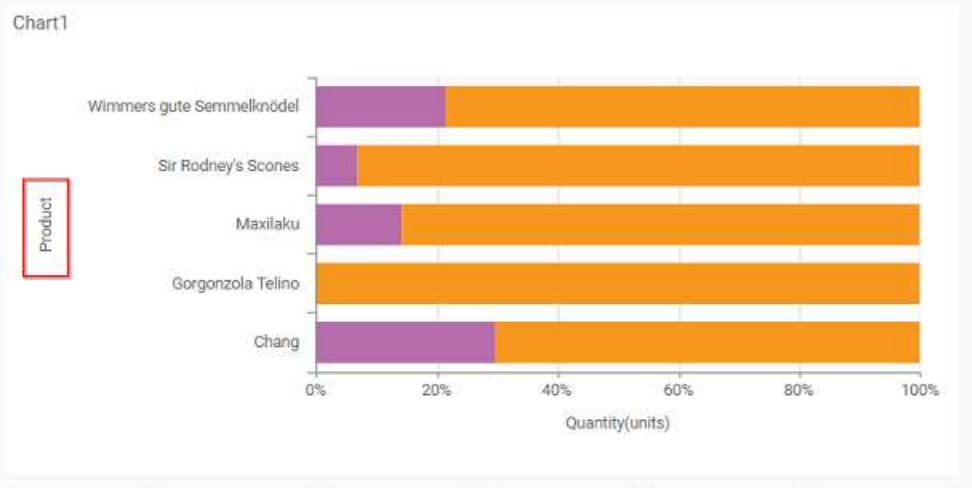
Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

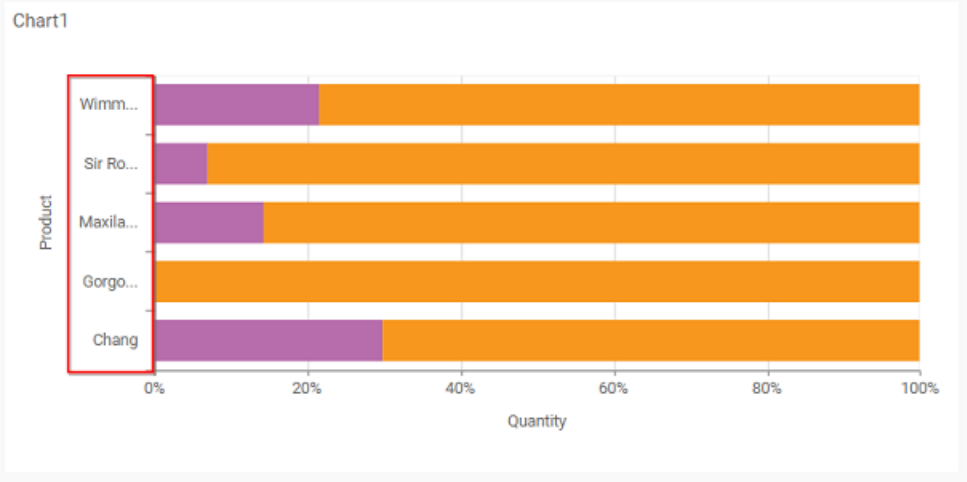


Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

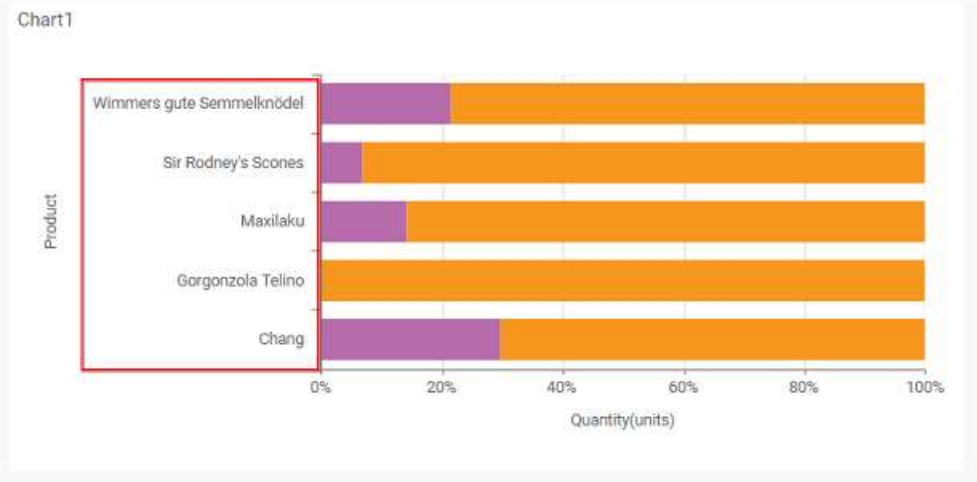
Trim

This option trims the end of overlapping label in the axis.



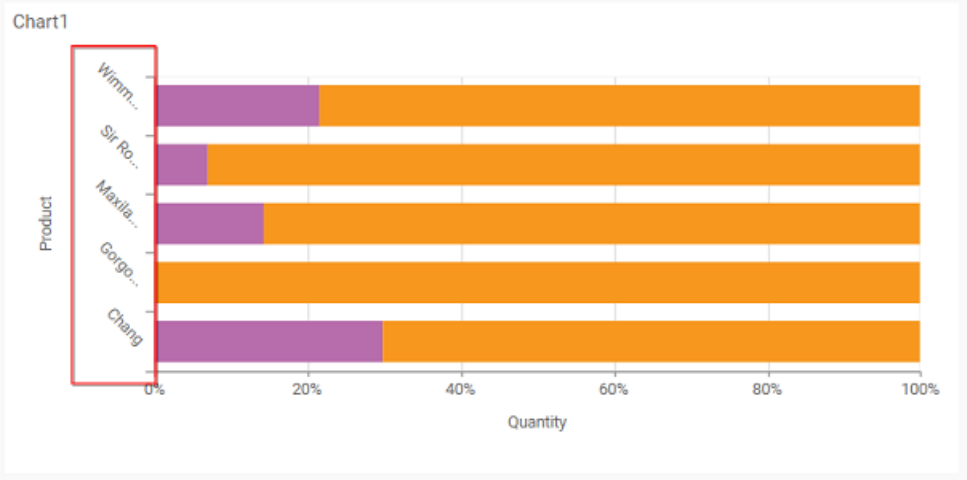
Hide

This option hides the overlapping label in the axis.



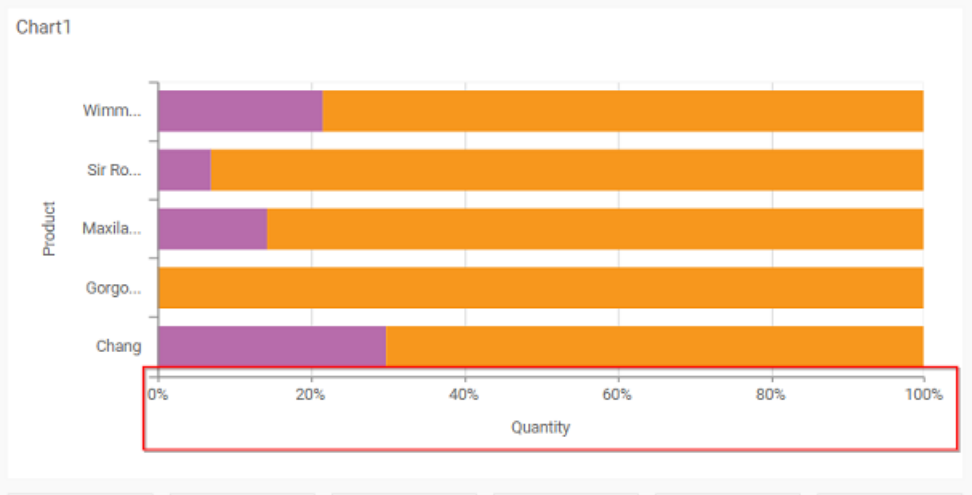
Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



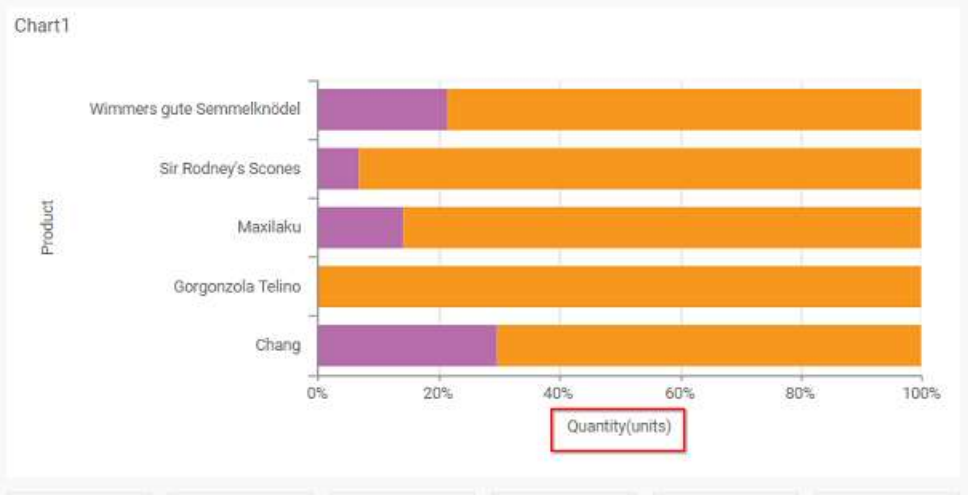
Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.



Show Primary Value Axis Title

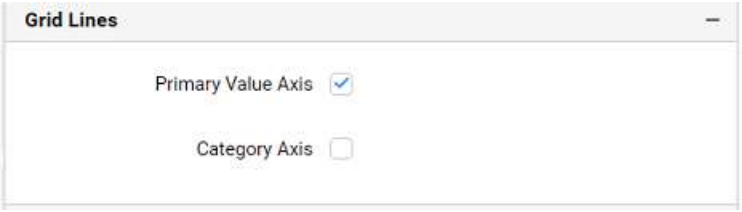
This allows you to enable the visibility of Primary Value Axis title of chart.



Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

Grid Lines

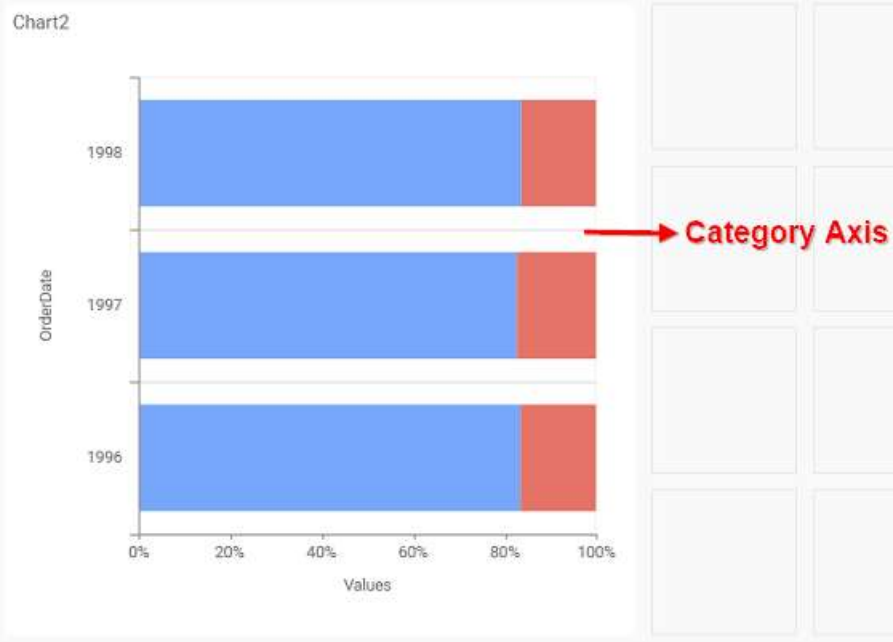


Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the 100% stacked bar chart.

Category Axis

This allows you to enable the Category Axis gridlines for the 100% stacked bar chart.

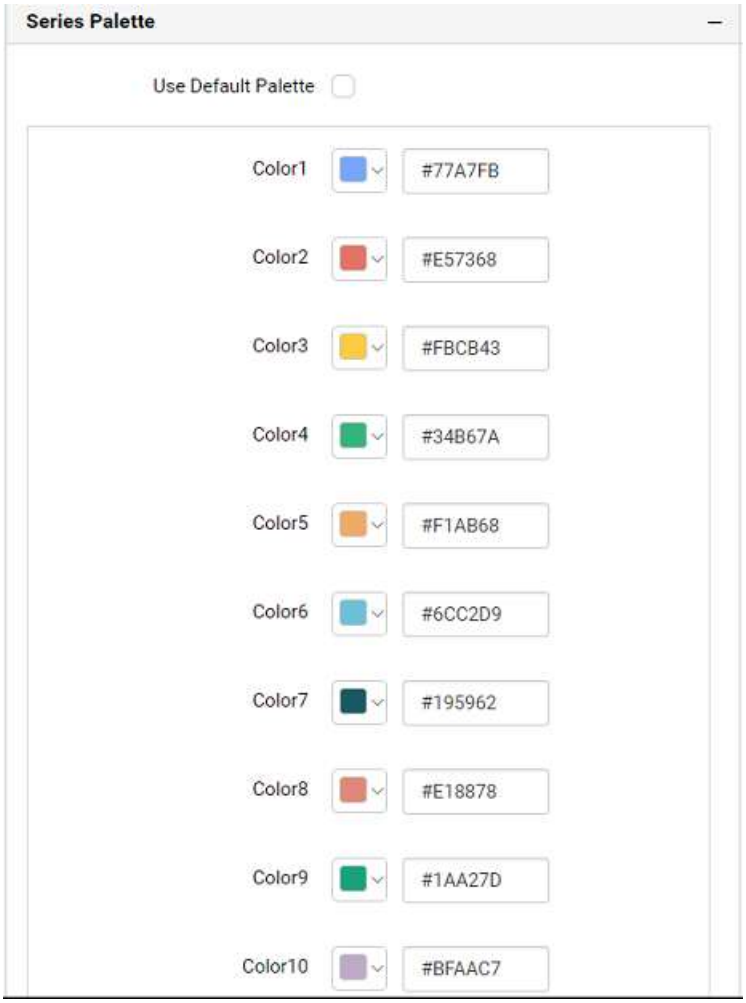


Series Palette

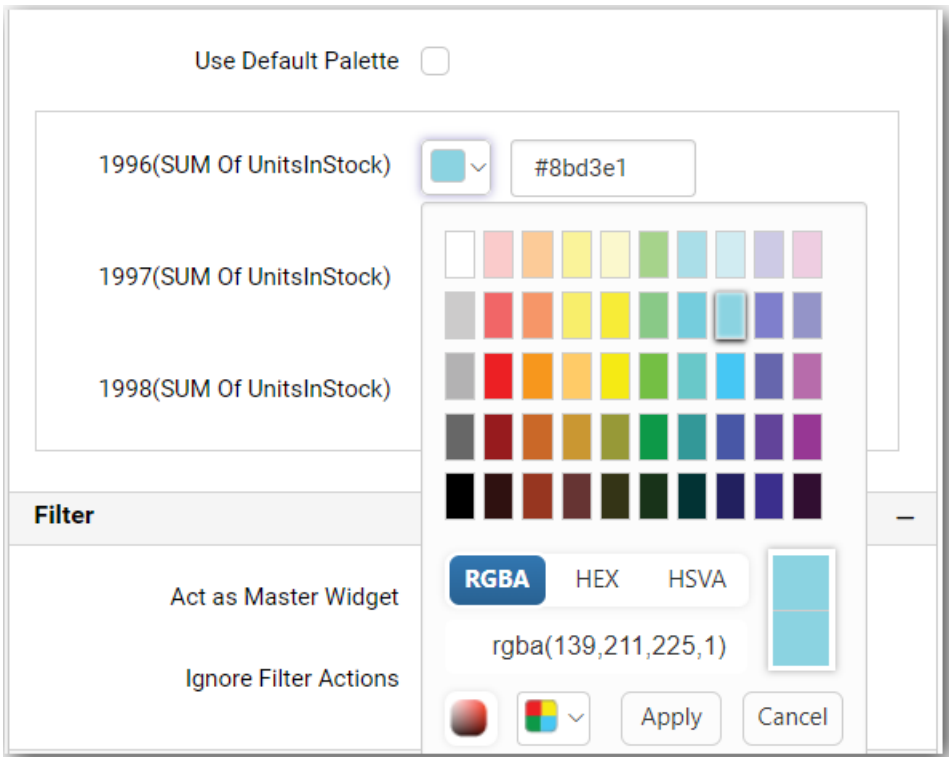
This allows you to customize the chart series color through Series Palette section.

Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.

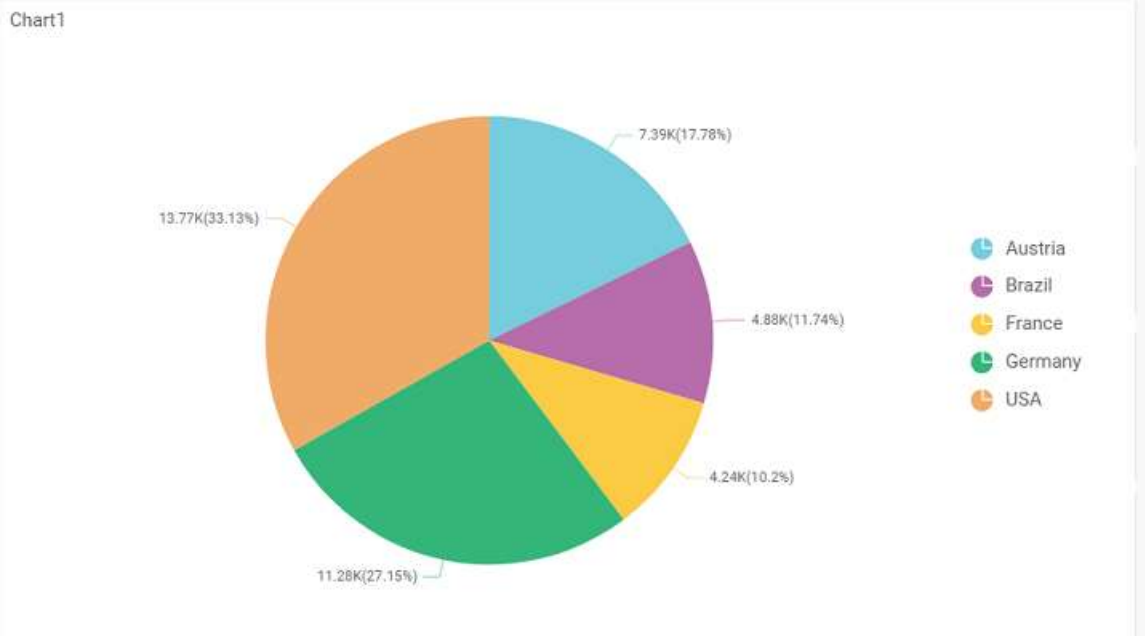


By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.



Pie Chart

Pie Chart allows you to showcase proportionality of each item to the total in the form of pie-slices.

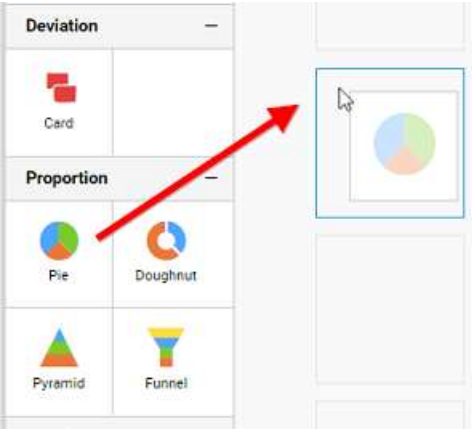


How to configure table data to Pie Chart?

Pie Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

To configure data into pie chart follow the steps

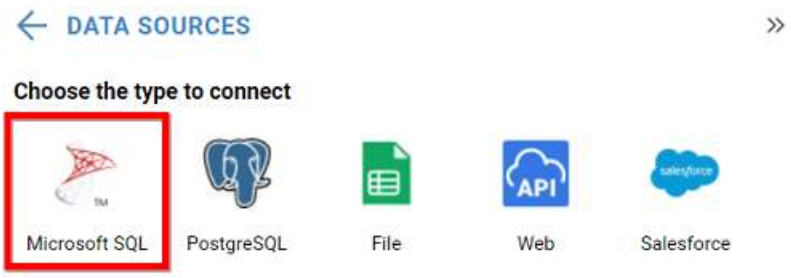
Drag and drop the pie chart into canvas and resize it to your required size.



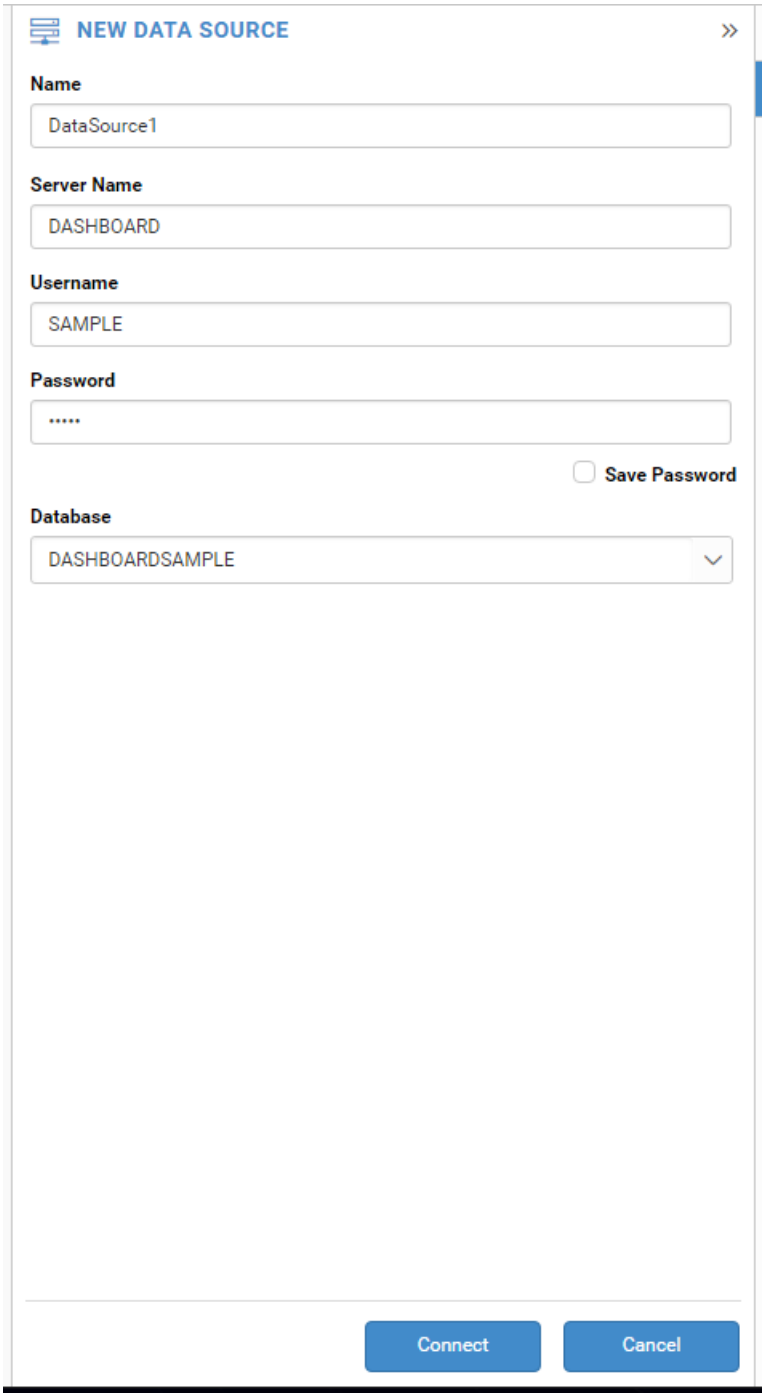
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

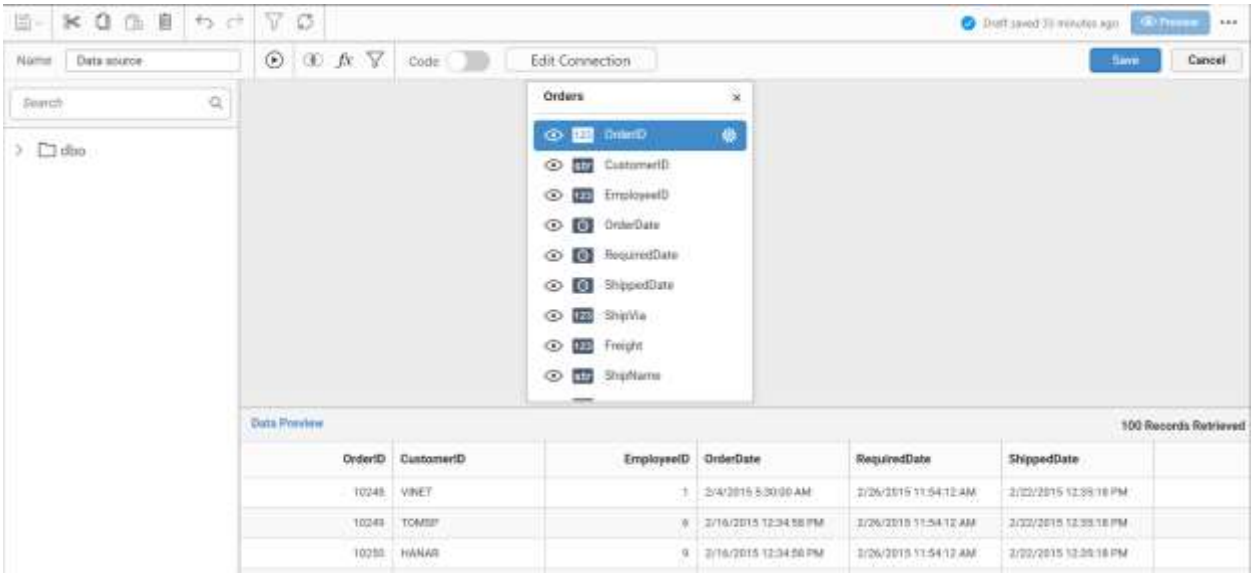
In the connection type panel, click any one (Here SQL Connection type is selected for demonstration) of the listed connection type button shown.



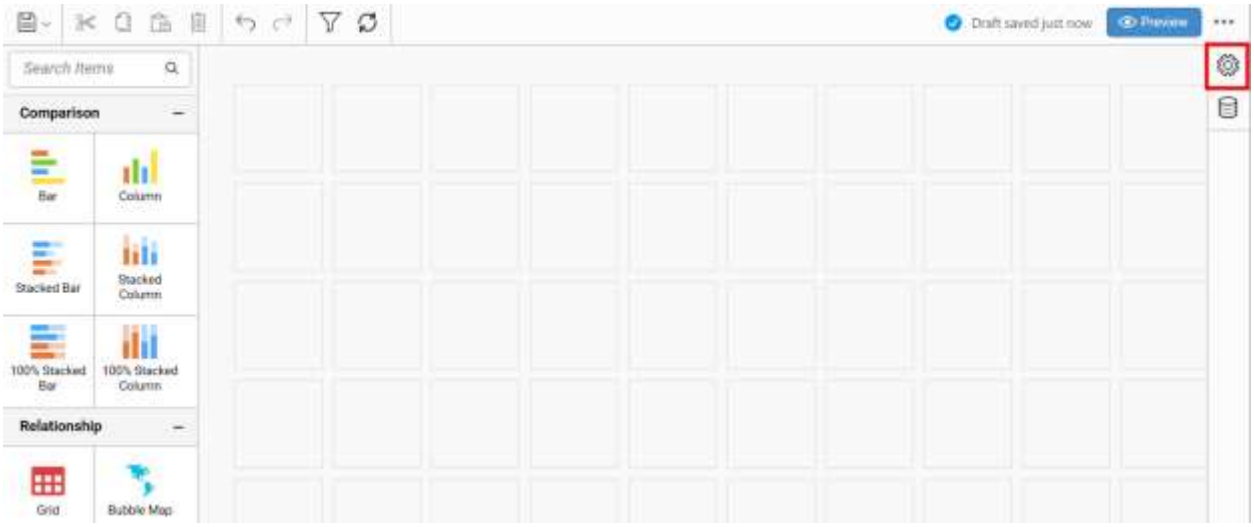
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



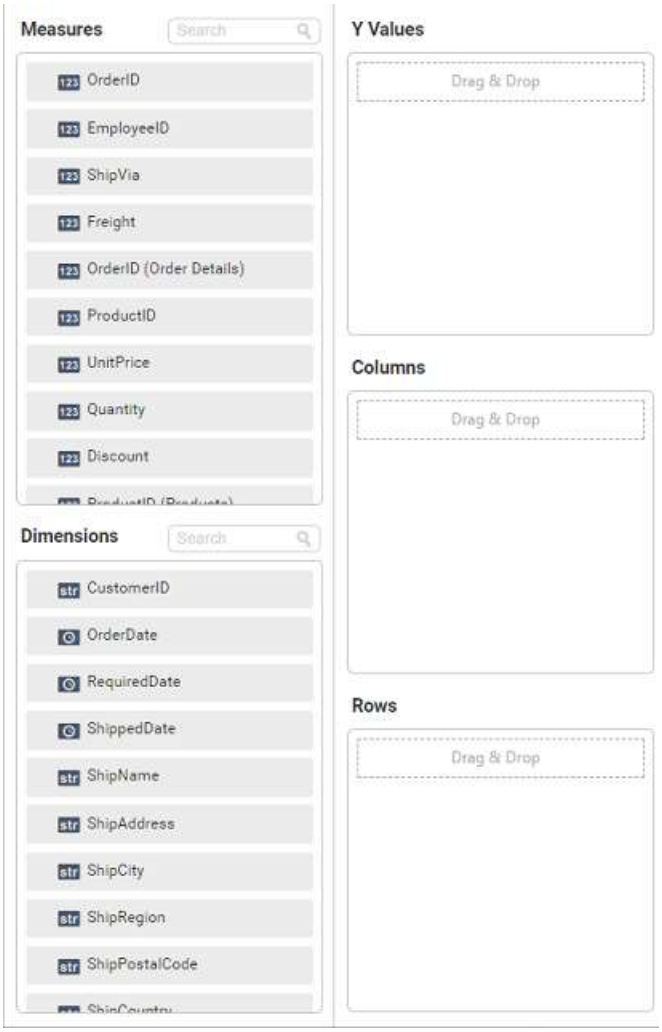
Drag your preferred table or view from the left pane from data design view, click Save button.



Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.



The data tab will be opened with available measures and dimensions from the connected data source

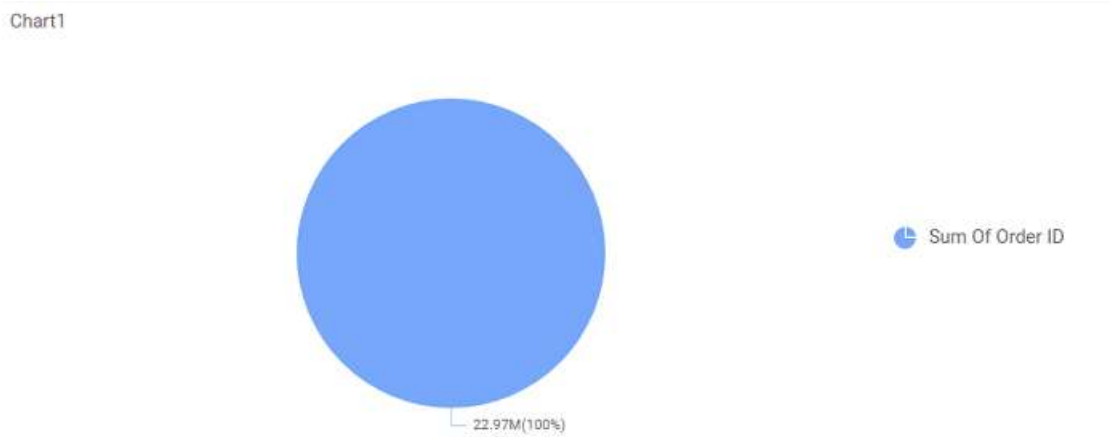


You can add the required data from Measures and Dimensions into required field.

Adding Y Values

You can add single or multiple columns from Measures into Y Values field.

Now the Pie chart will be rendered like this.

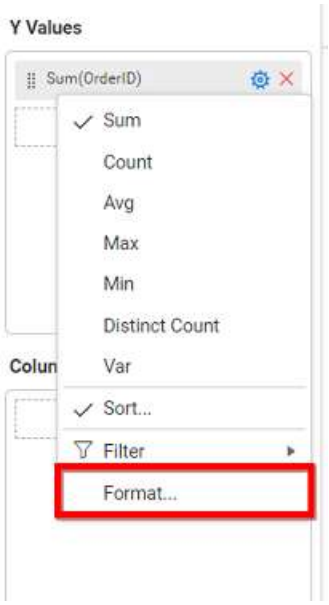


Click the Settings option to change required summary type from the available summary types shown in Settings.

You can Sort the data using Sort option shown under Settings menu list. To Sort the measure data, refer Sort

You can filter the data to be displayed in chart by using filter option. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

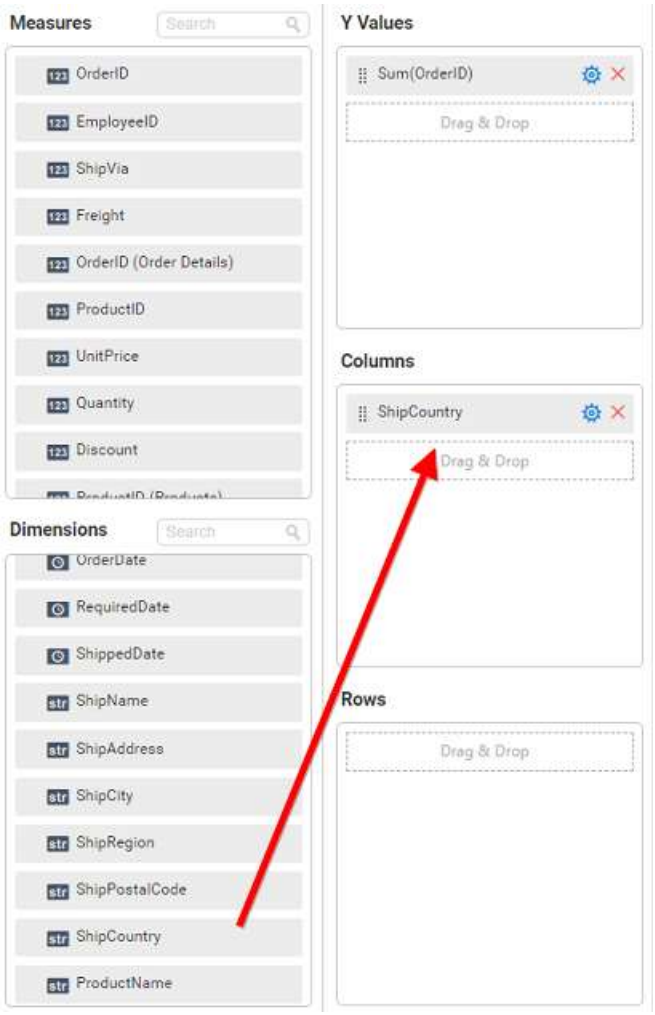


To remove the added value fields click highlighted button.

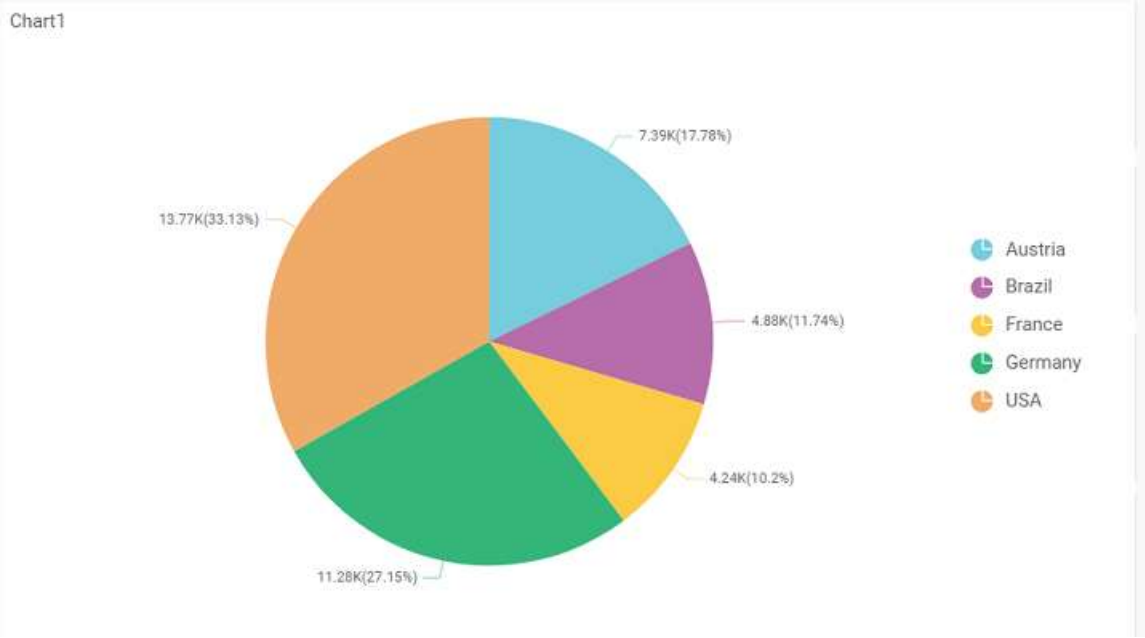
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



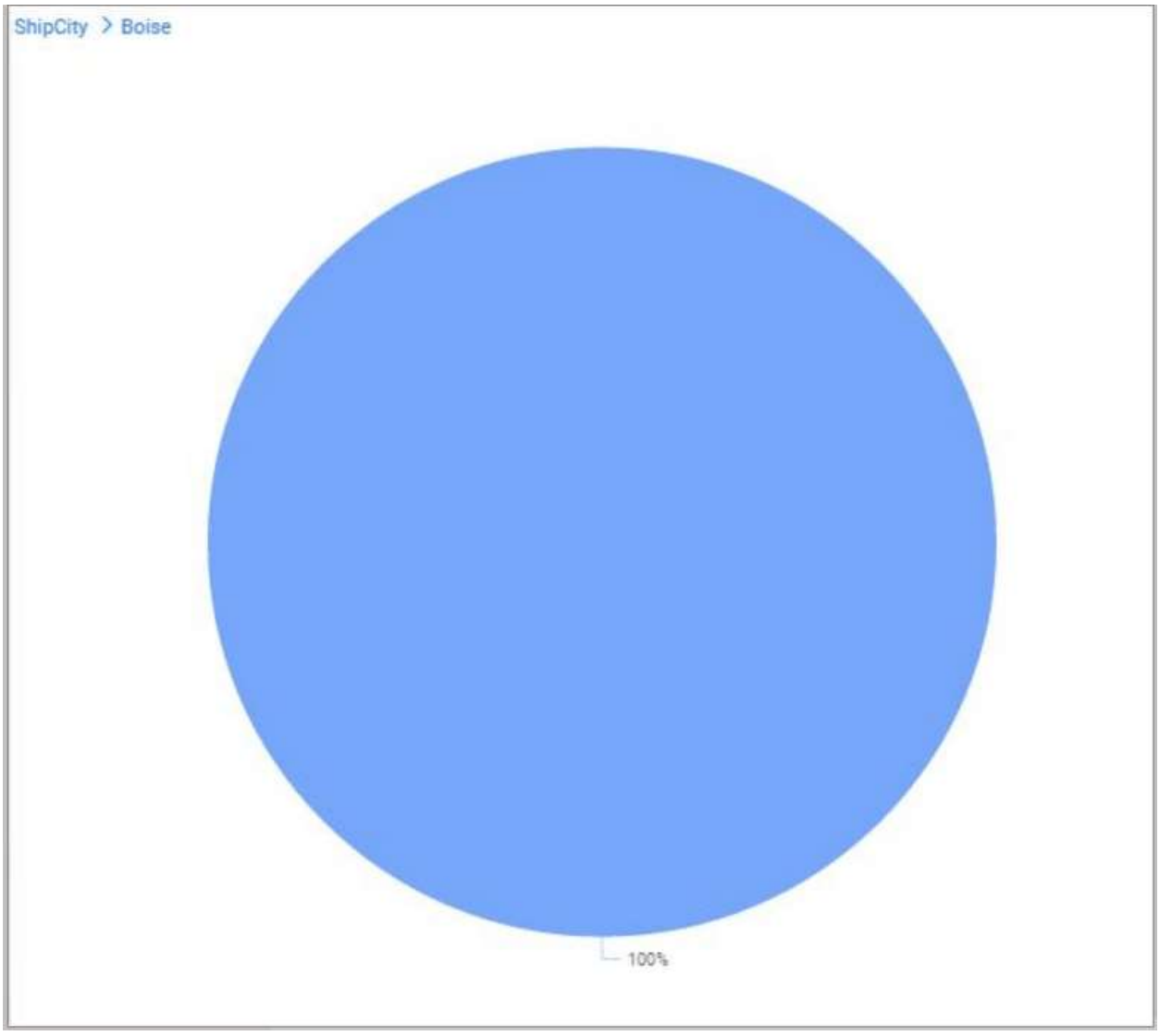
Pie chart will be rendered like this.



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

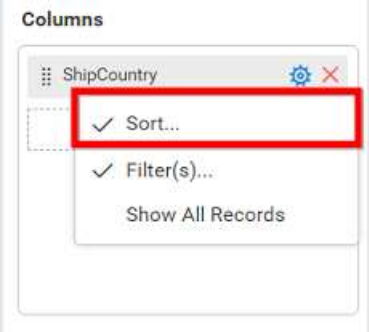
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.



You can change the Settings.

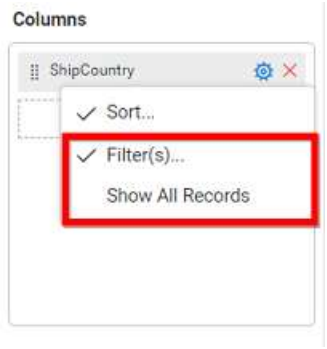
You can Sort the dimension data using sort option under Settings menu list. To apply sorting for the data, refer Sort.



Note: Filter will be set by default for top 5 records.

You can apply filters by selecting filters option in settings. For more details, refer filter.

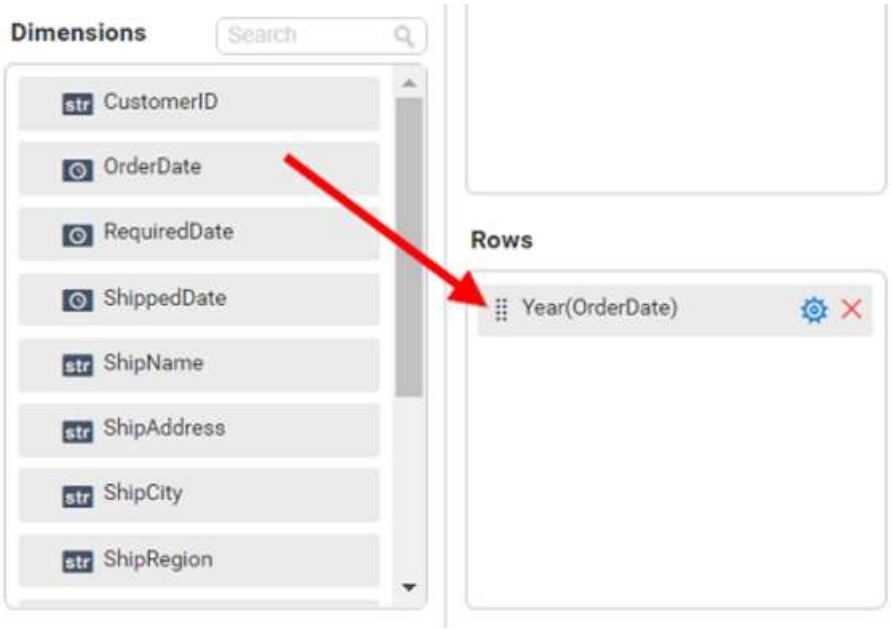
To show all records click on Show All Records.



Similarly you can add the Measures and Expressions into columns field.

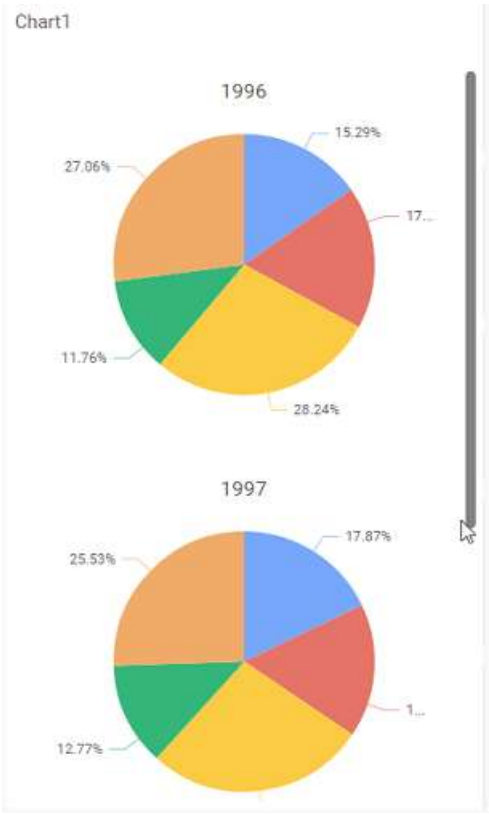
Adding Row

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render pie chart in series.



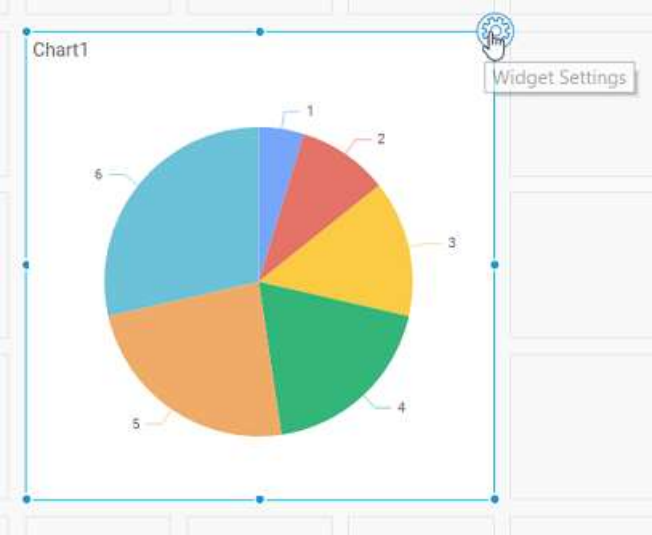
Scroll down to see all charts.

How to format pie chart?

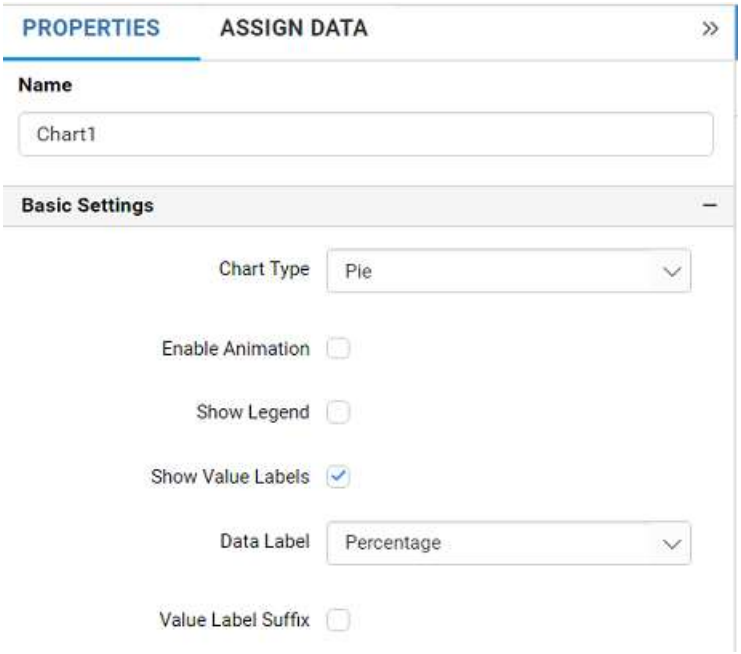
You can format the pie chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into pie chart follow the steps

1. Drag and drop the pie chart into canvas and resize it to your required size.
2. Configure the data into pie chart.
3. Focus on the pie chart and click on widget settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this pie chart widget.

Basic Settings

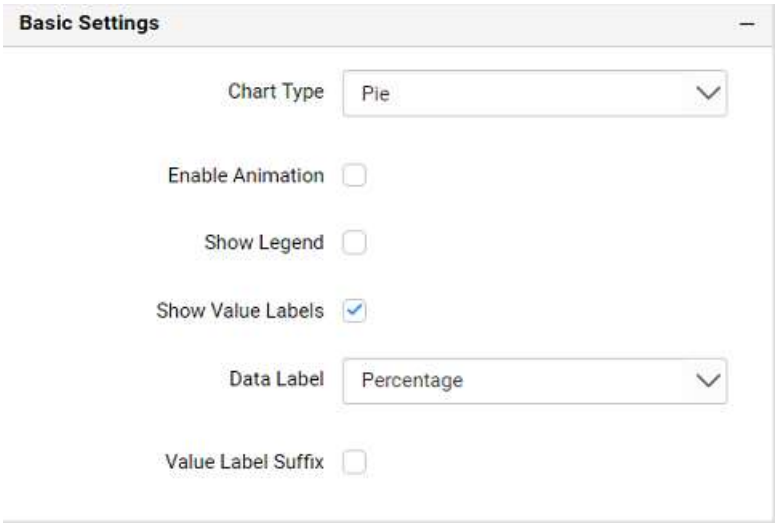


Chart Type

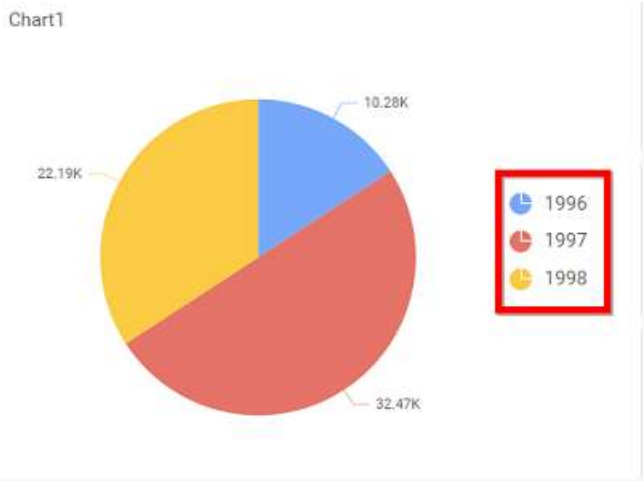
This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the series rendering in animated mode.

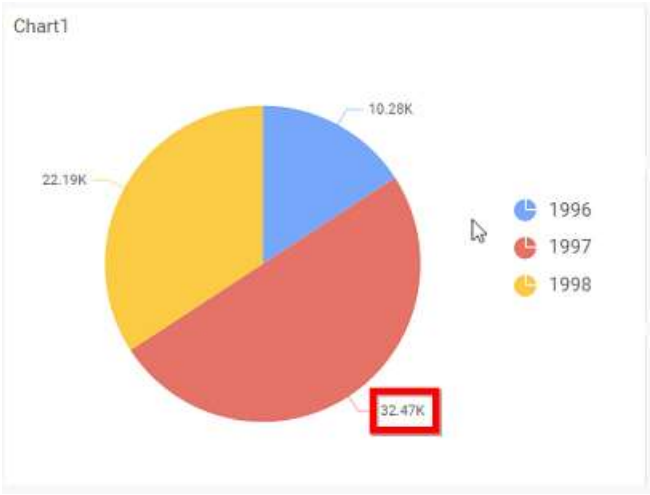
Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box). Enabling this option of Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.



Show Value Label

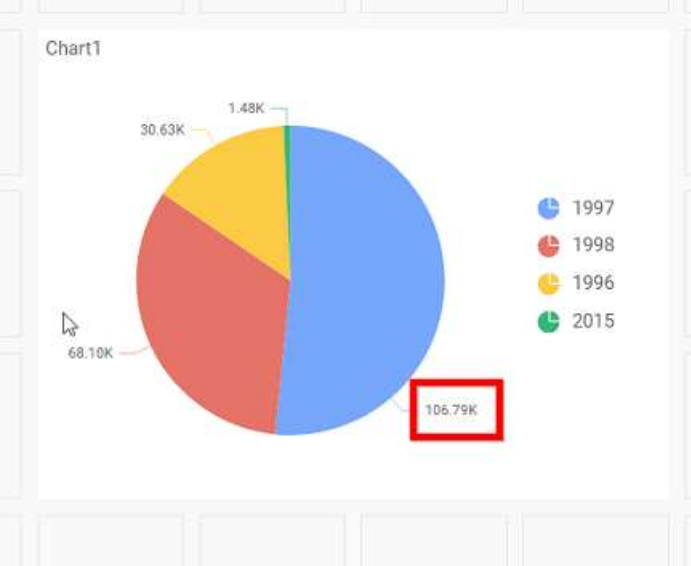
This allows you to toggle the visibility of value labels.



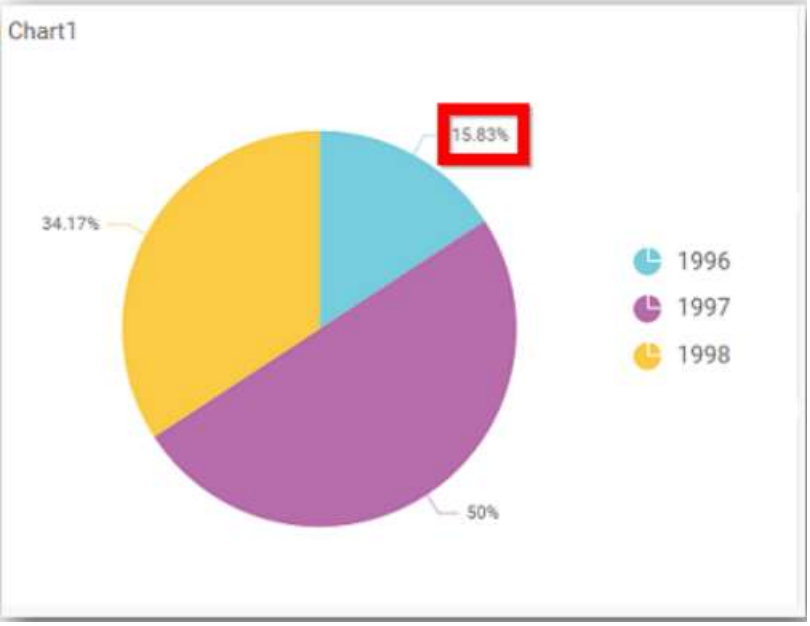
Data Label

This allows you to define the display format either as value or as percentage or both.

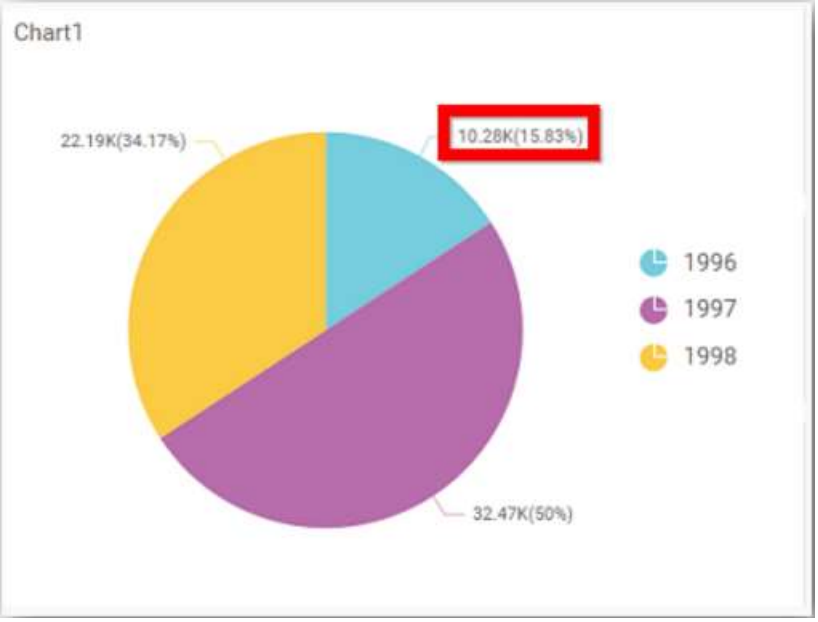
Value



Percentage

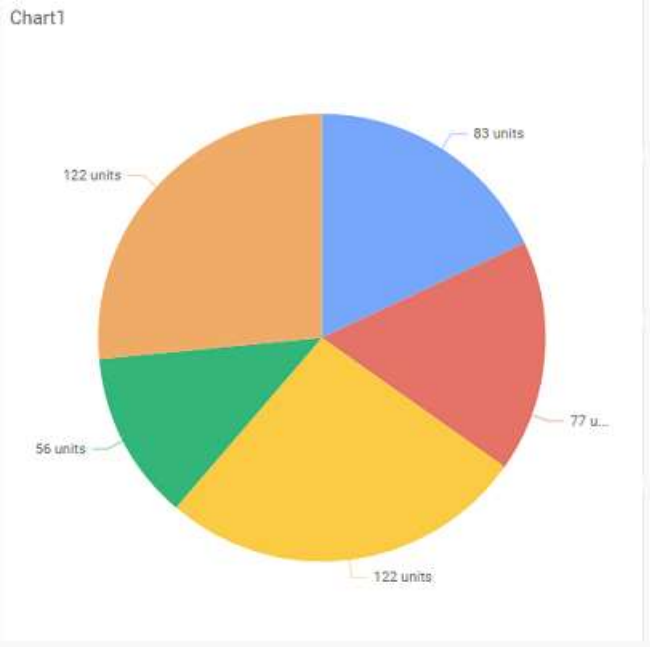


Value and Percentage

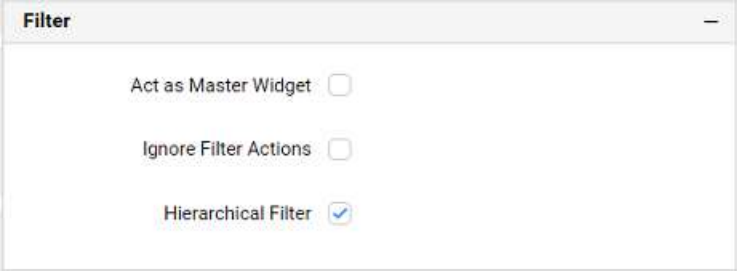


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter



Act as Master Widget

This allows you to define this pie chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

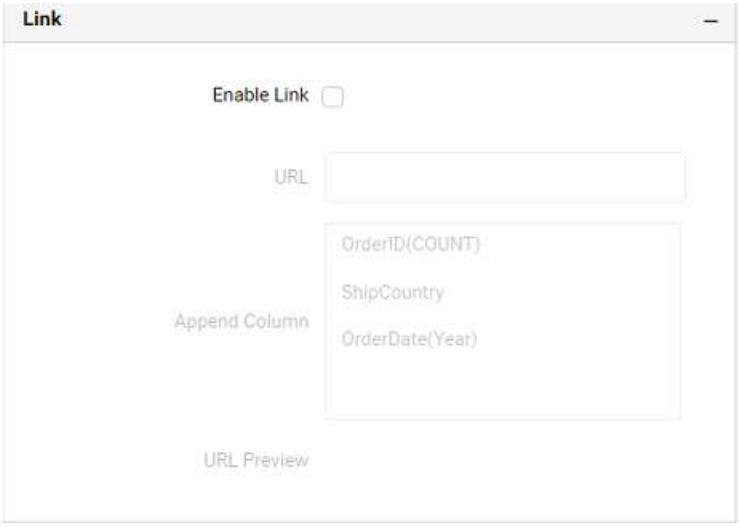
This allows you to define this pie chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

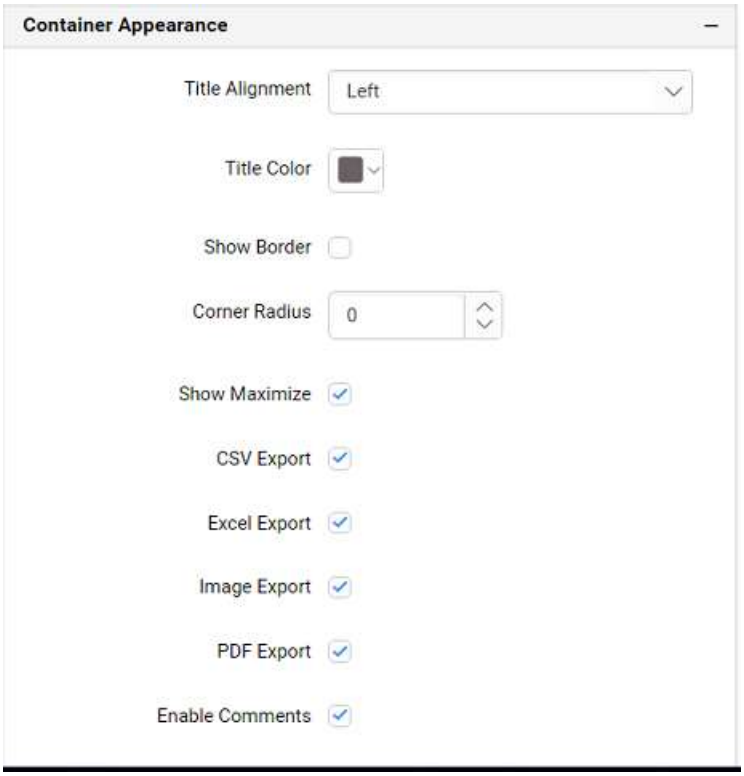
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply different text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if **Show Border** property is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this pie chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this pie chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this pie chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

Image Export

This allows you to enable/disable the image export option for this pie chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

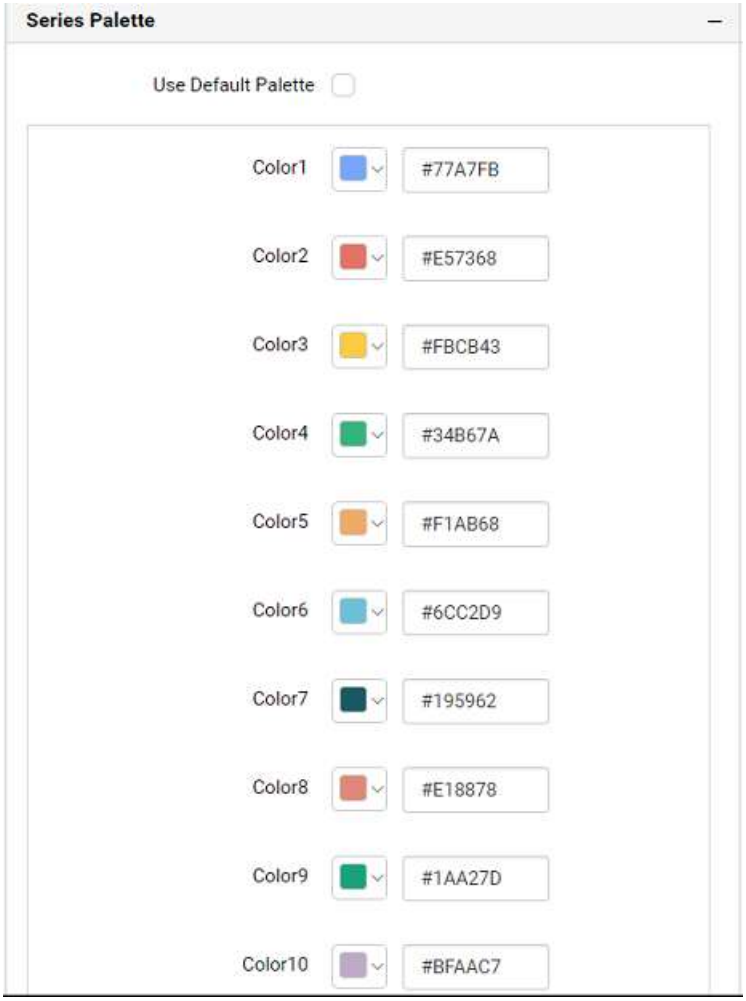
This allows you to enable/disable comment for dashboard widget. For more details refer here

Series Palette

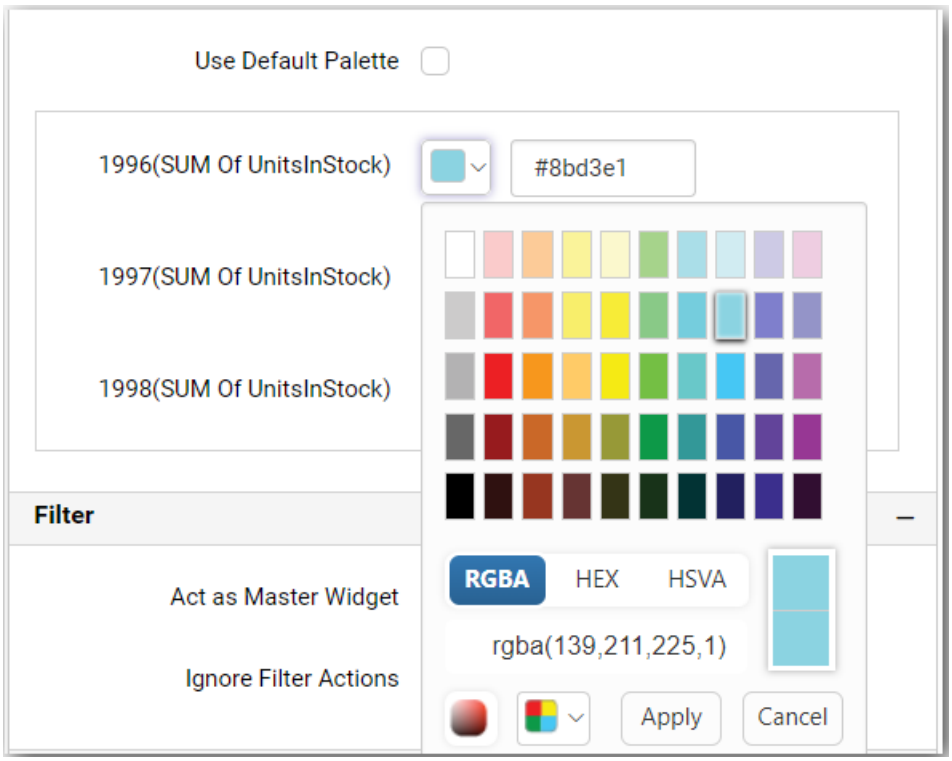
This allows you to customize the chart series color through Series Palette section.

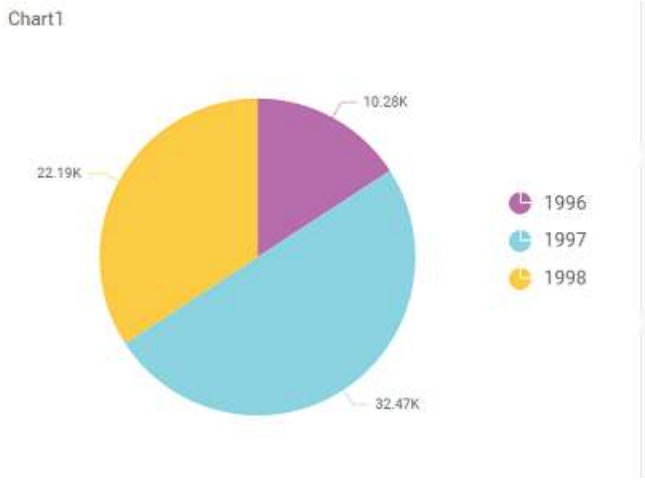
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to proportion series segments.



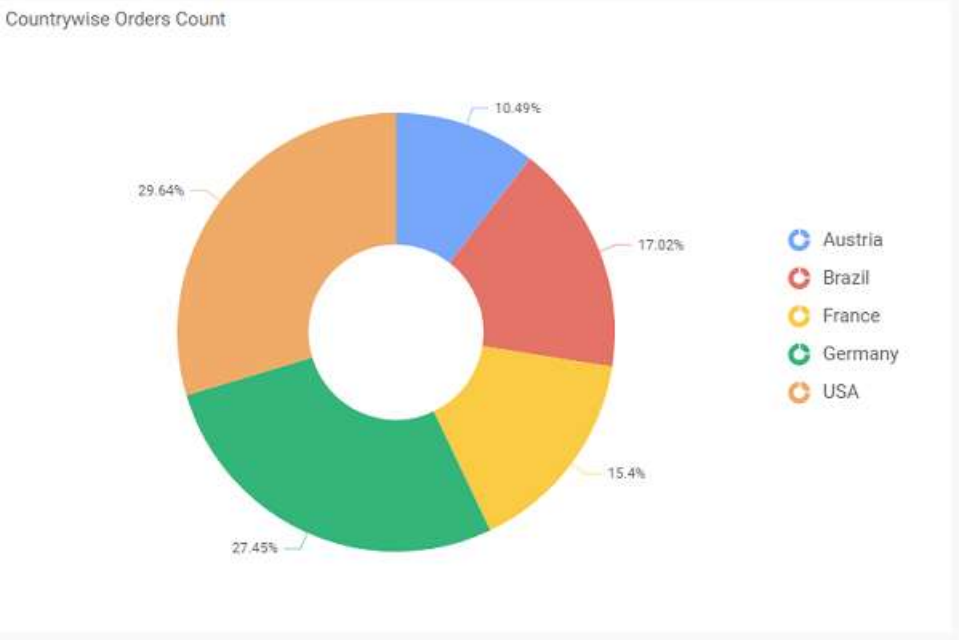
By toggle off the Use Default Palette, you can customize the proportion series segments' colors. This section shows a palette of colors. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Doughnut Chart

Doughnut Chart allows you to showcase proportionality of each item to the total in the form of donut-slices. To plot a doughnut chart, a minimum requirement of 1 value and 1 column is needed.

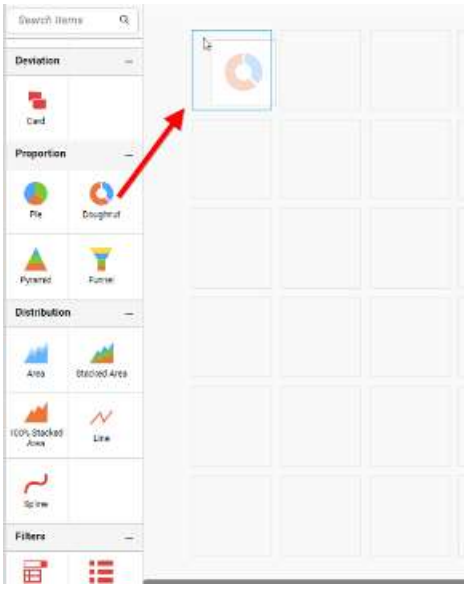


How to configure table data to doughnut chart?

Doughnut Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

To configure data into doughnut chart follow the steps

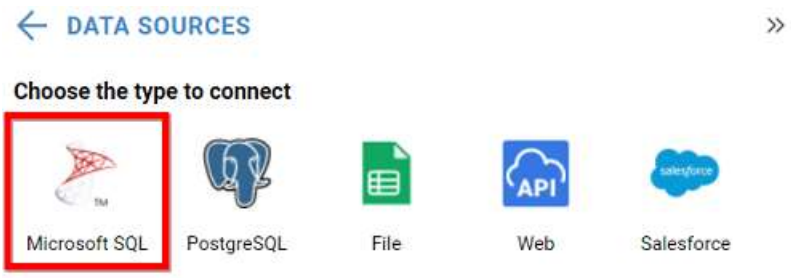
Drag and drop the doughnut chart into canvas and resize it to your required size.



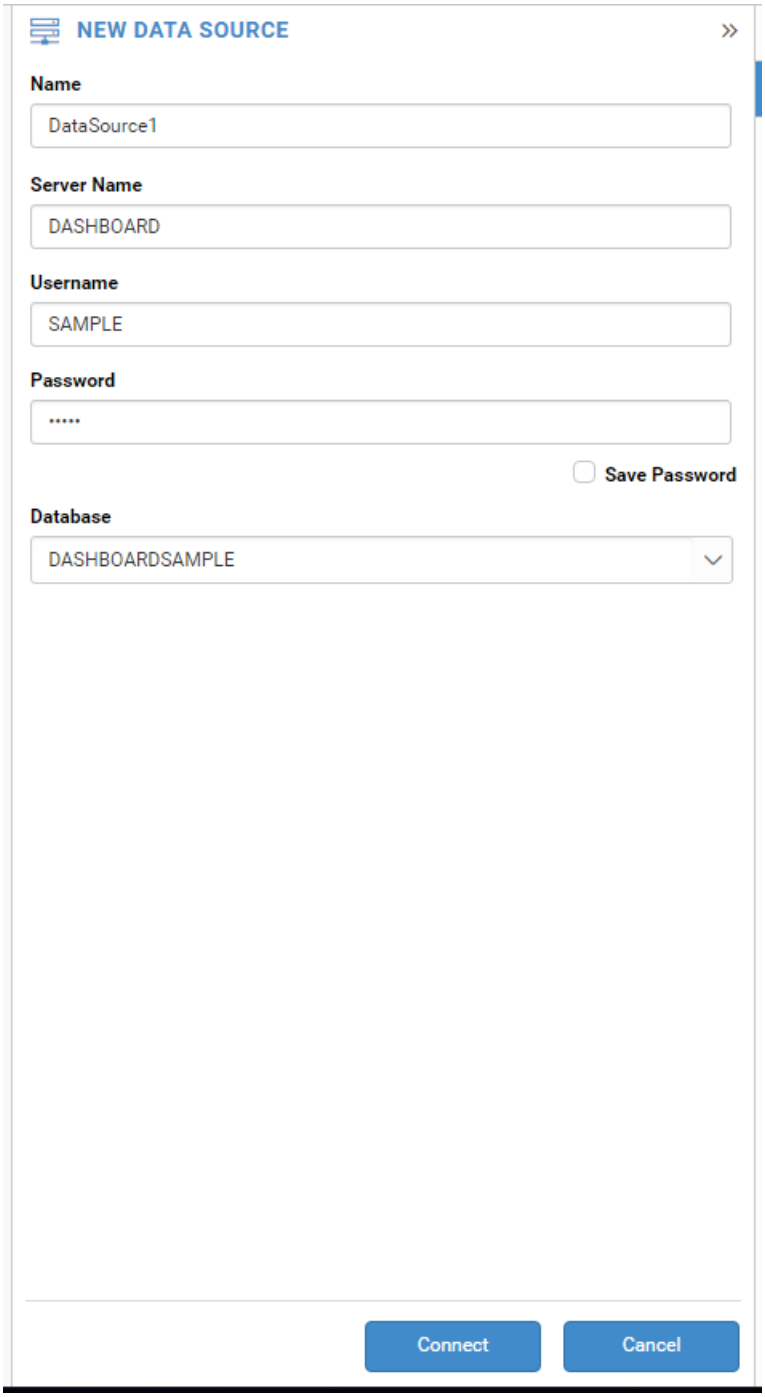
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

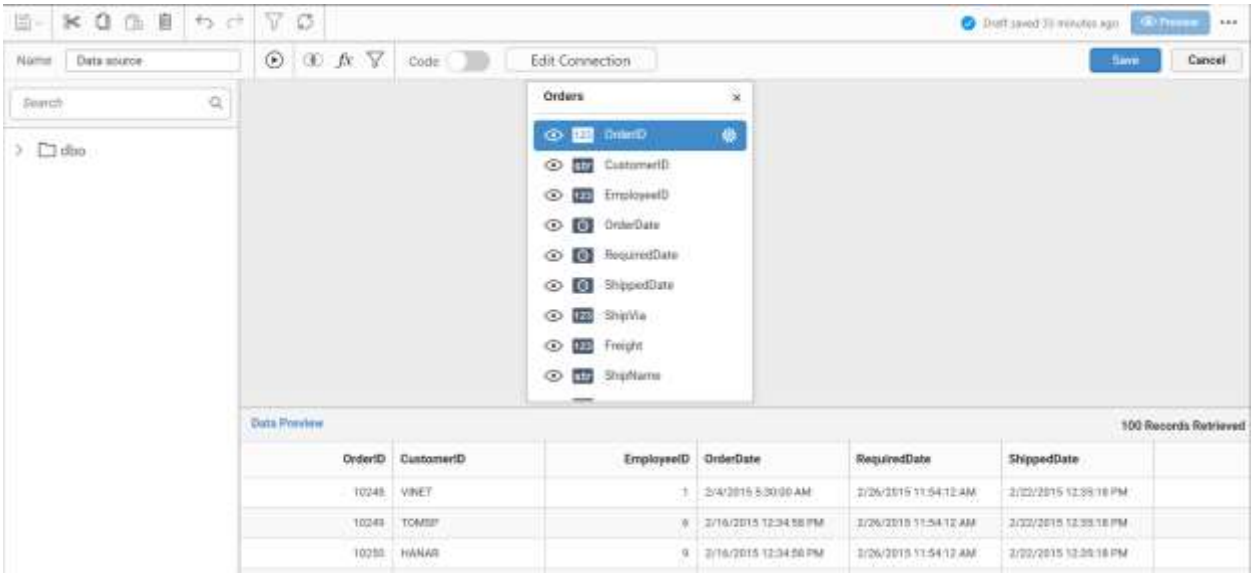
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



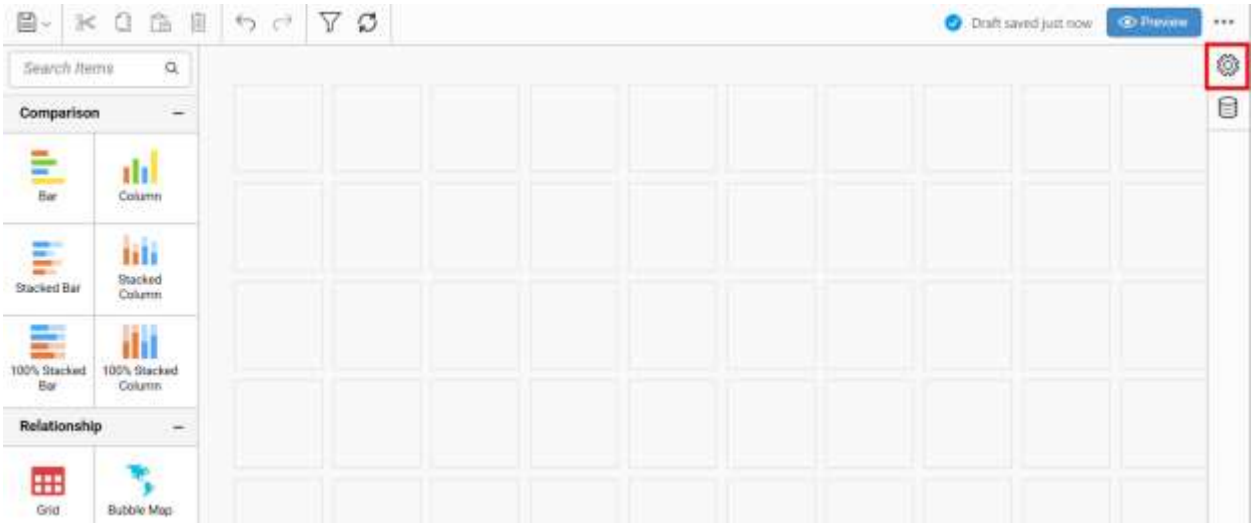
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



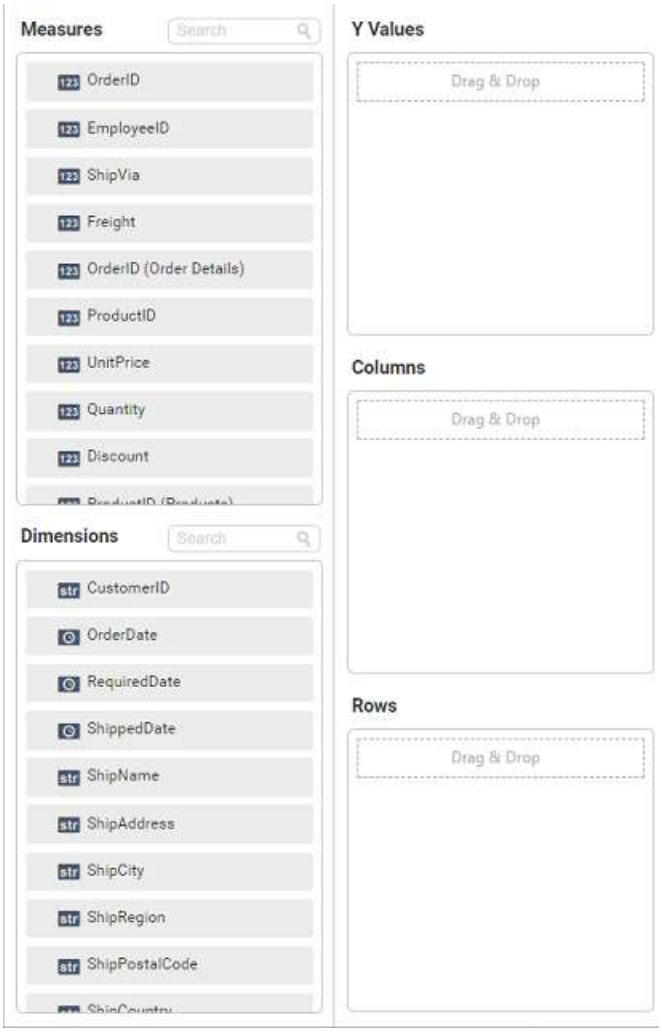
Drag your preferred table or view from the left pane from data design view, click Save button.



Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.



The data tab will be opened with available measures and dimensions from the connected data source

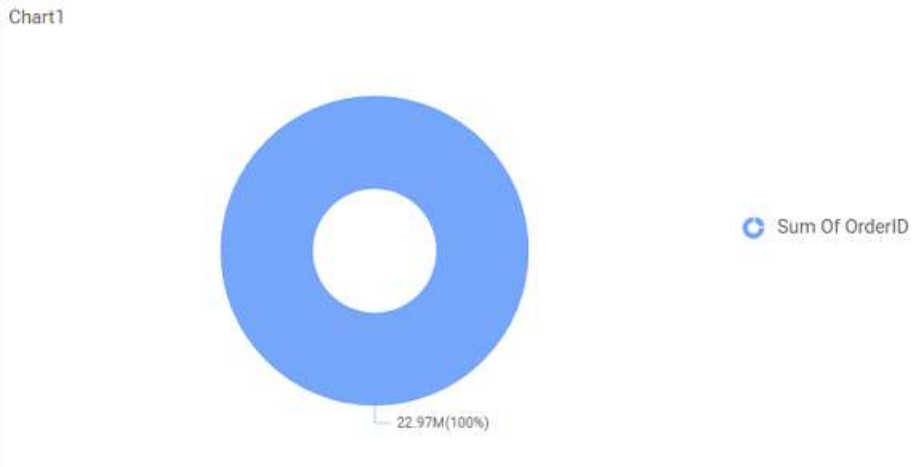


You can add the required data from Measures and Dimensions into required field.

Adding Y Values

You can add more than one Measures into Values field by drag and drop the required measure.

Now, the doughnut chart will be rendered like this

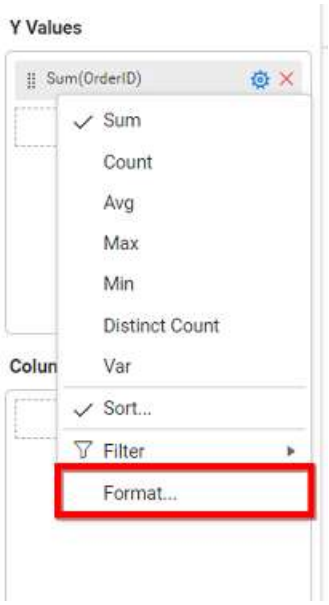


Click the Settings option to change required summary type from the available summary types shown in Settings.

You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

You can filter the data to be displayed in chart by using filter option. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

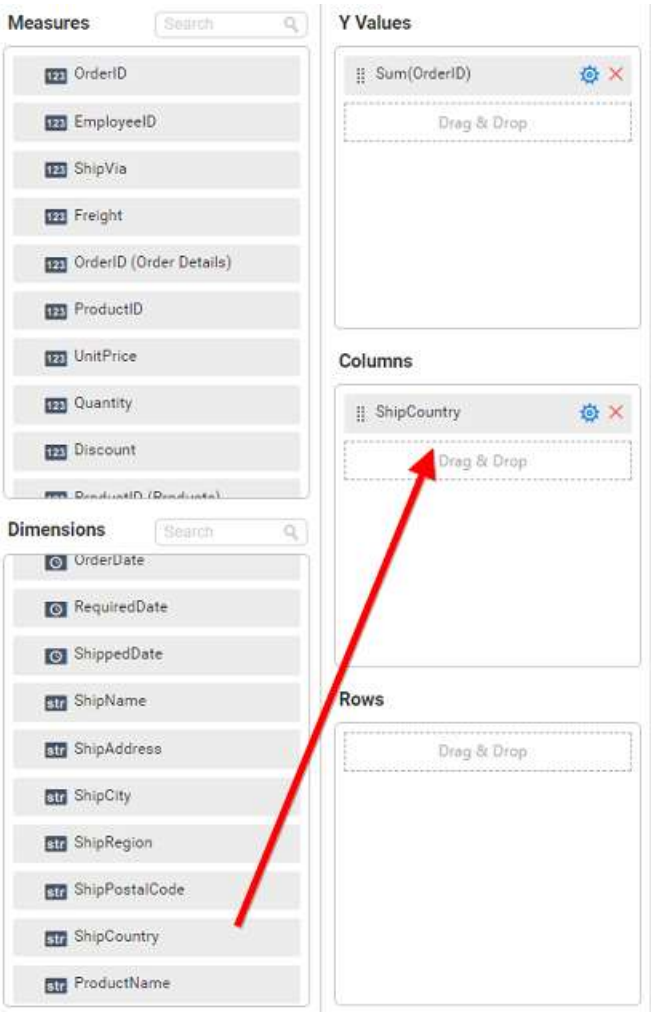


To remove the added value fields click highlighted button.

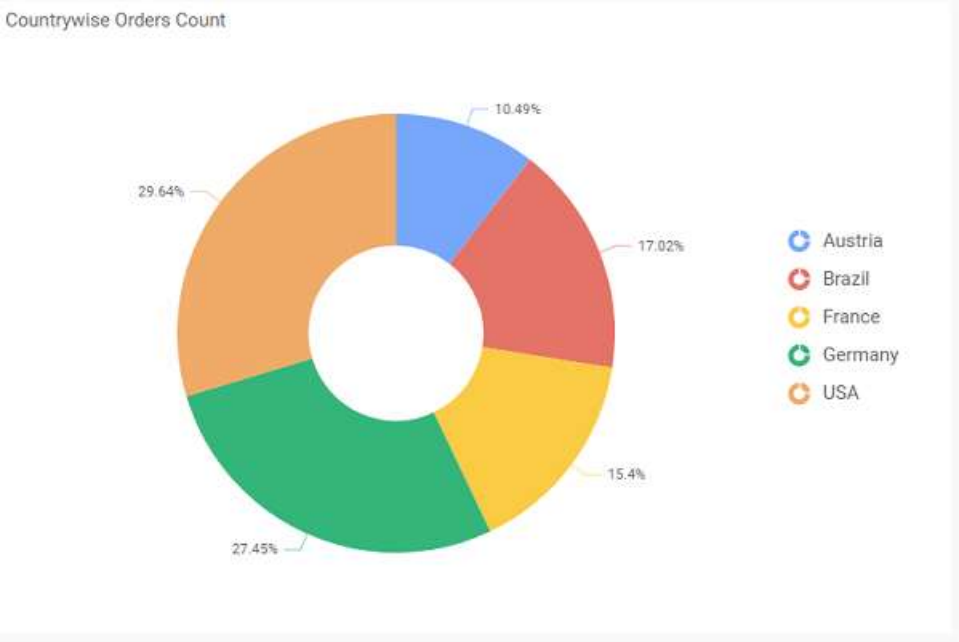
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field. .



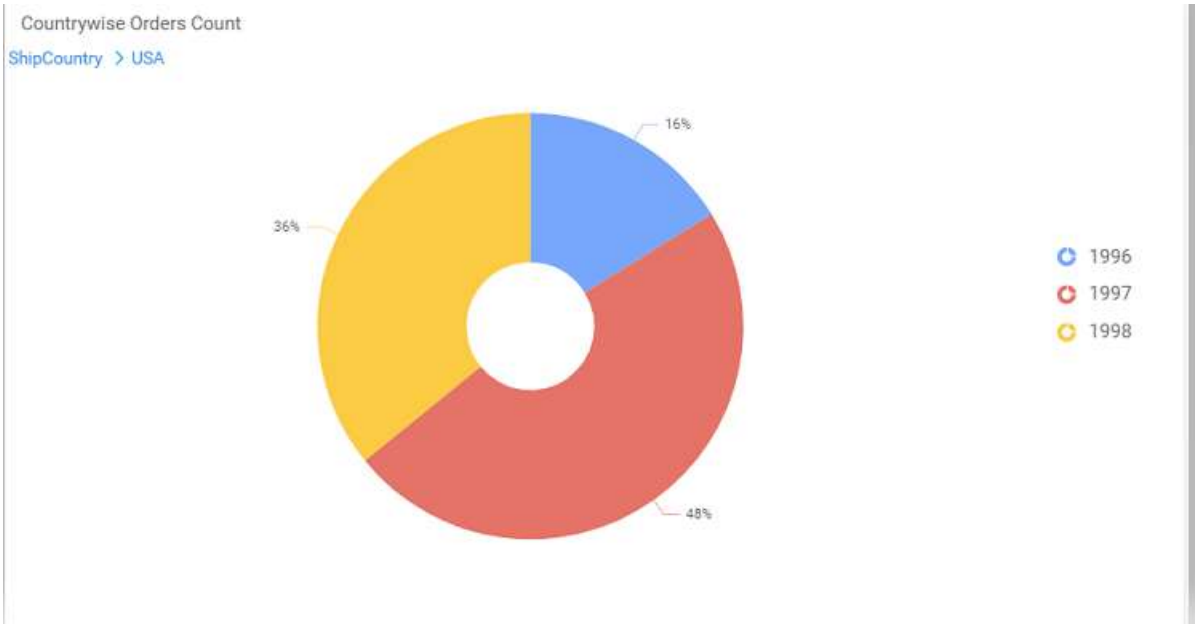
Doughnut chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

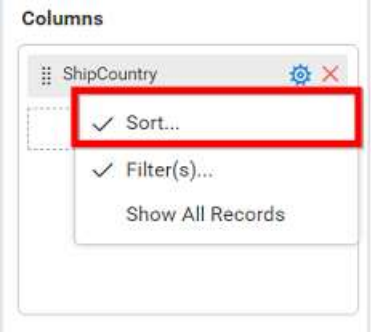
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.



You can change the Settings.

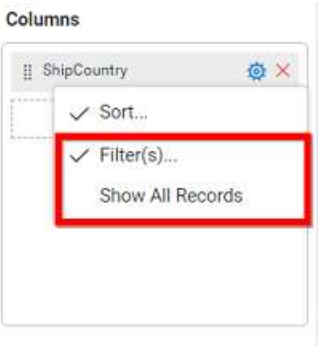
You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



Note: Filter will be set by default for top 5 records.

You can apply filters by selecting filters option in settings. For more details, refer filter.

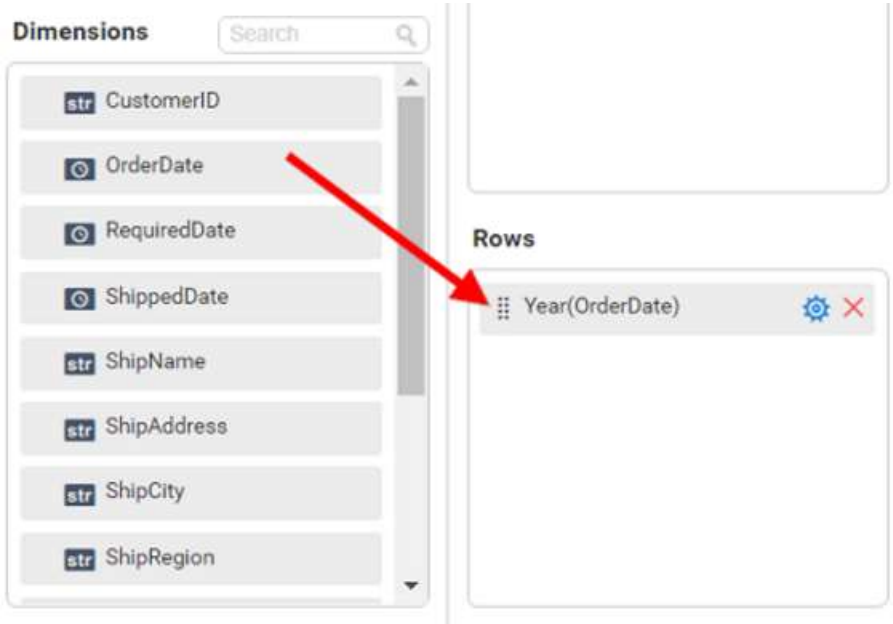
To show all records click on Show All Records.



Similarly you can add the Measures and Expression Columns into column field.

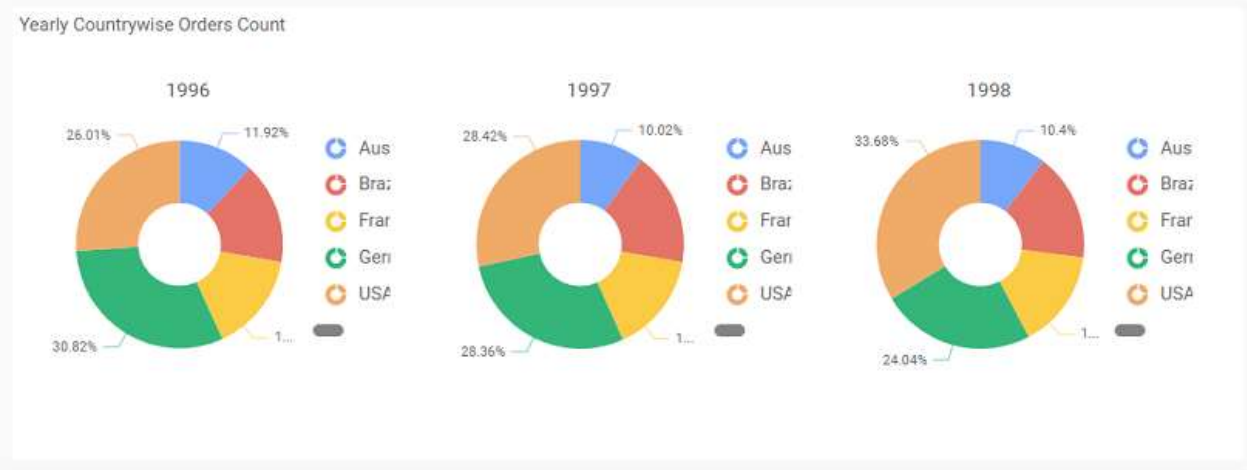
Assigning Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render doughnut chart in series.



Scroll down to see all charts.

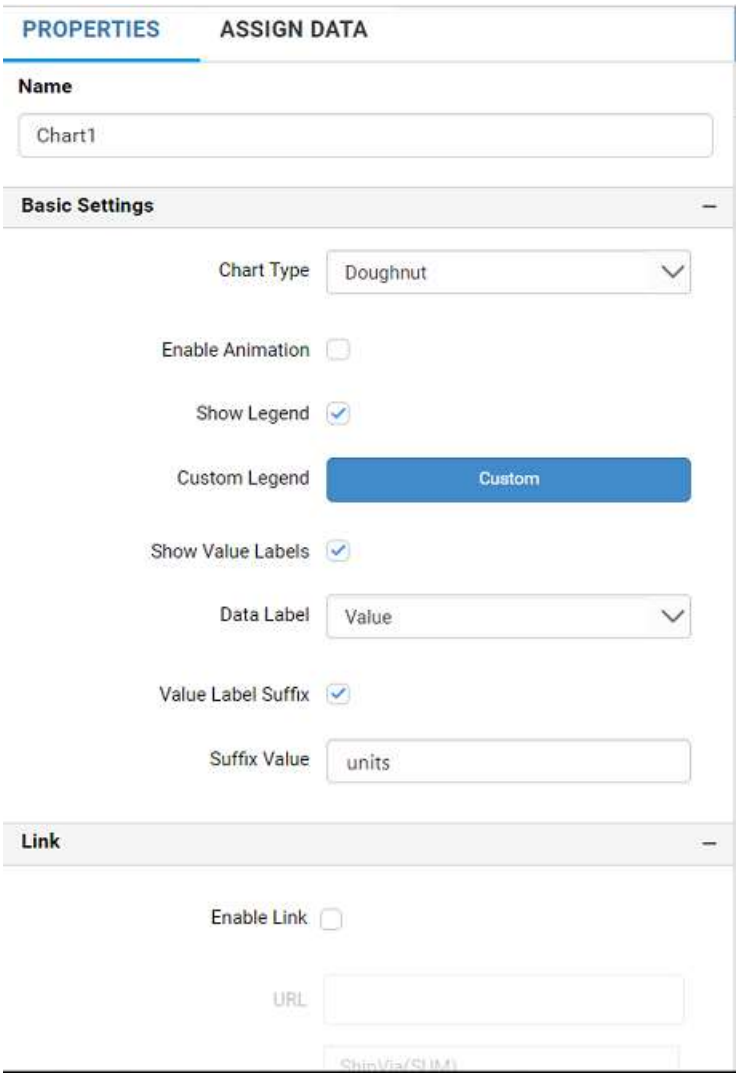
How to format doughnut chart?

You can format the doughnut chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into doughnut chart follow the steps

1. Drag and drop the doughnut chart into canvas and resize it to your required size.
2. Configure the data into doughnut chart.
3. Focus on the doughnut chart and click on widget settings.

The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to set title for this doughnut chart widget.

Basic Settings

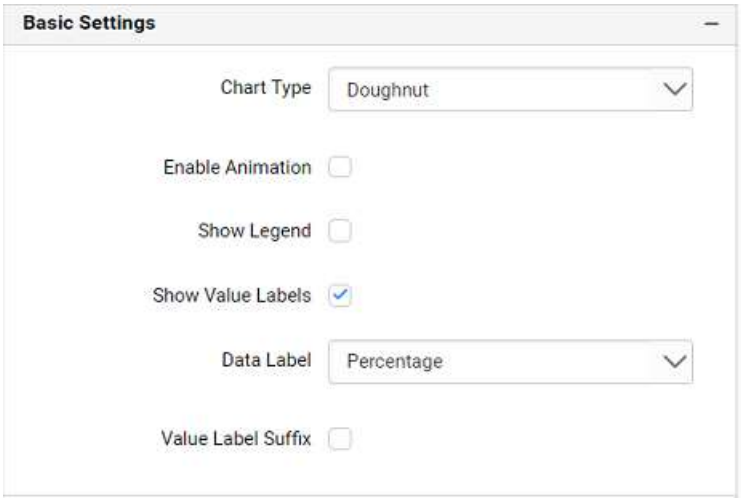


Chart Type

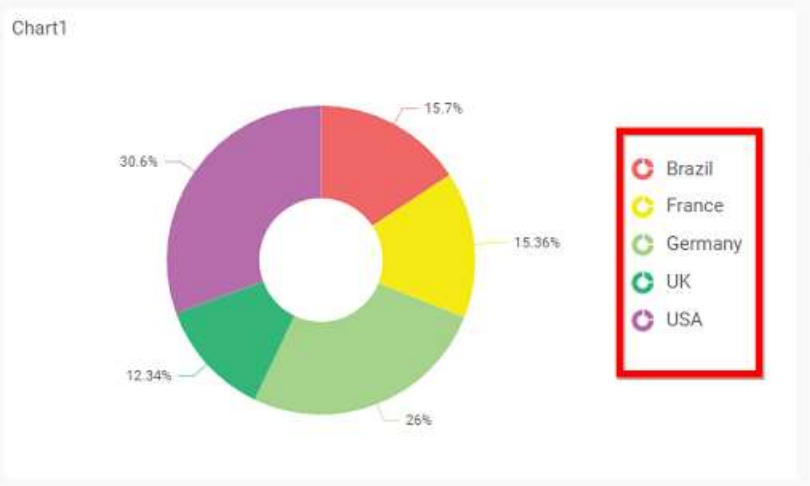
This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the series rendering in animated mode.

Show Legend

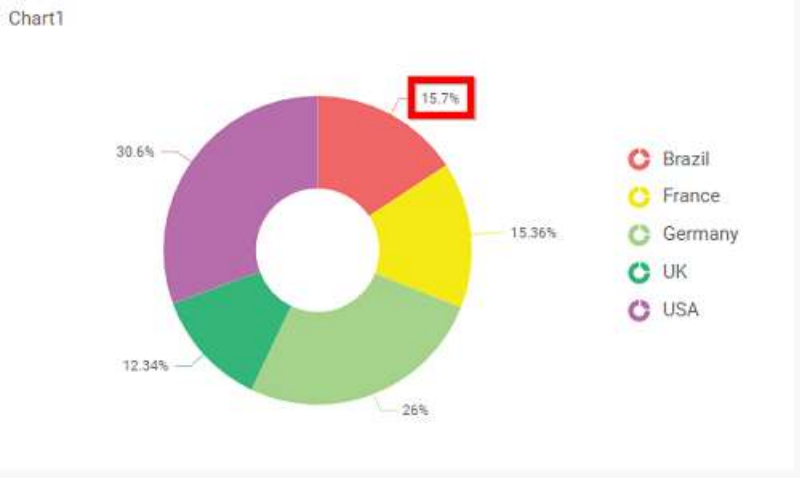
A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Show Value Labels

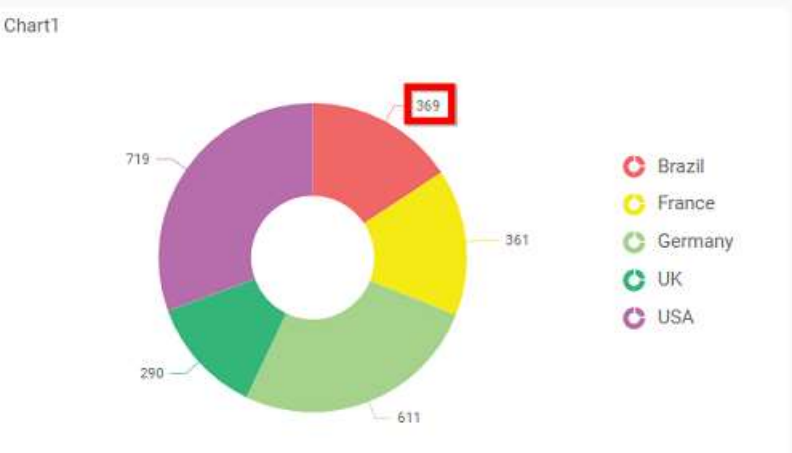
This allows you to toggle the visibility of value labels.



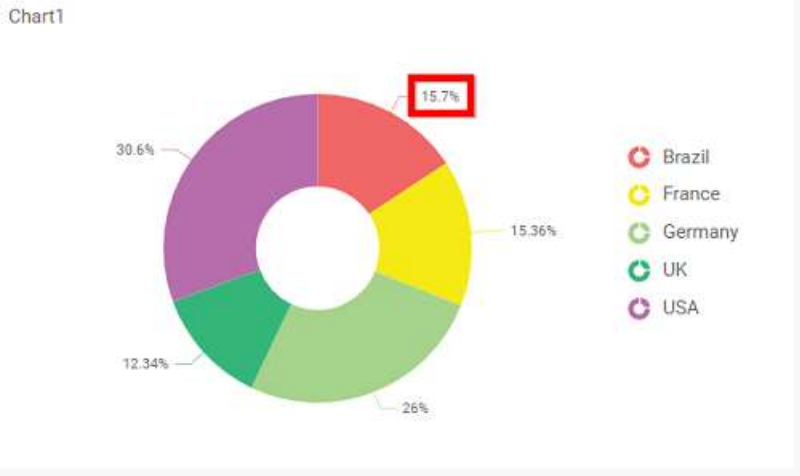
Data Label

This allows you to define the display format either as value or as percentage or both.

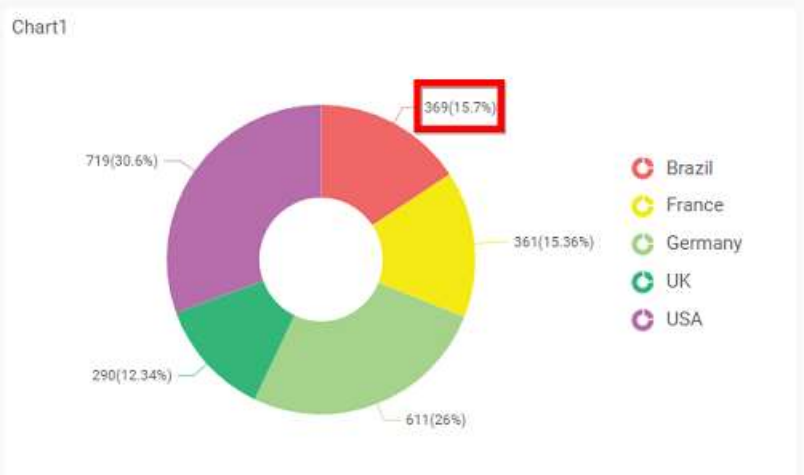
Value



Percentage

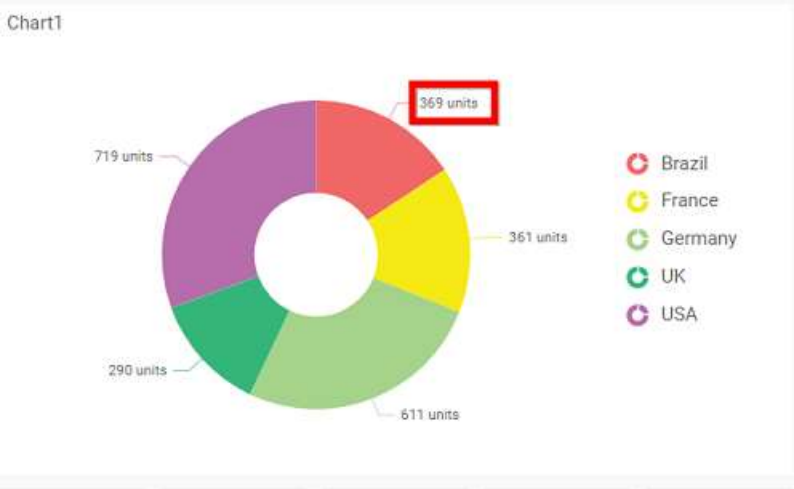


Value and Percentage

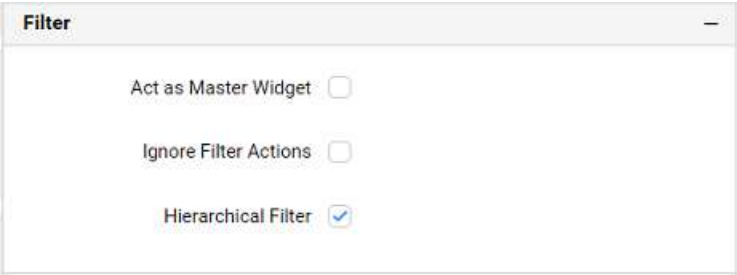


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter



Act as Master Widget

This allows you to define this doughnut chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

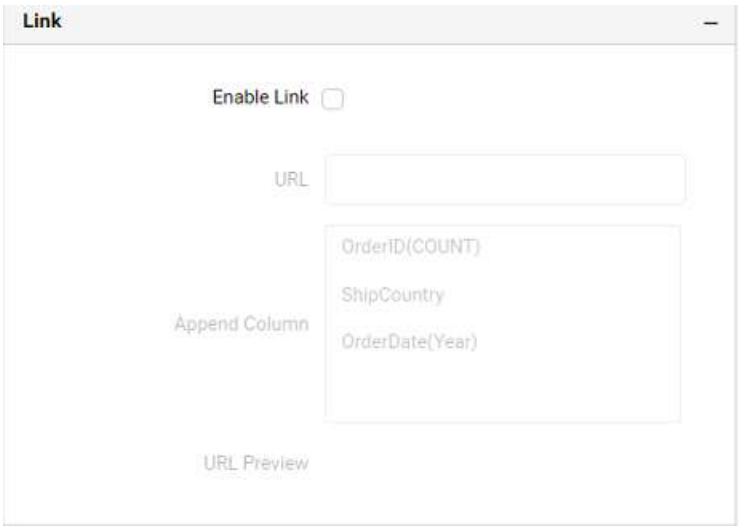
This allows you to define this doughnut chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

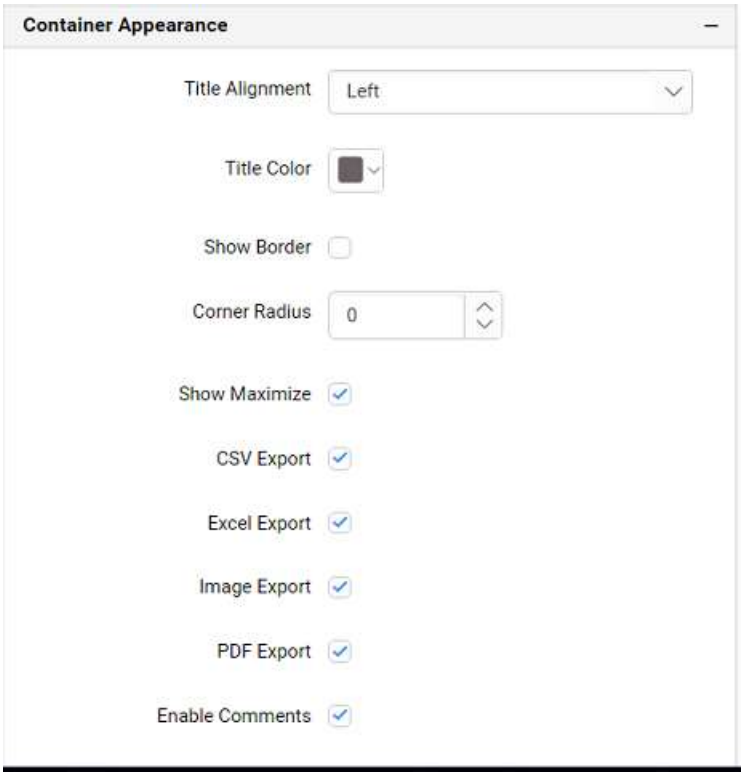
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if **Show Border** property is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this doughnut chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this doughnut chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this doughnut chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

Image Export

This allows you to enable/disable the image export option for this doughnut chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

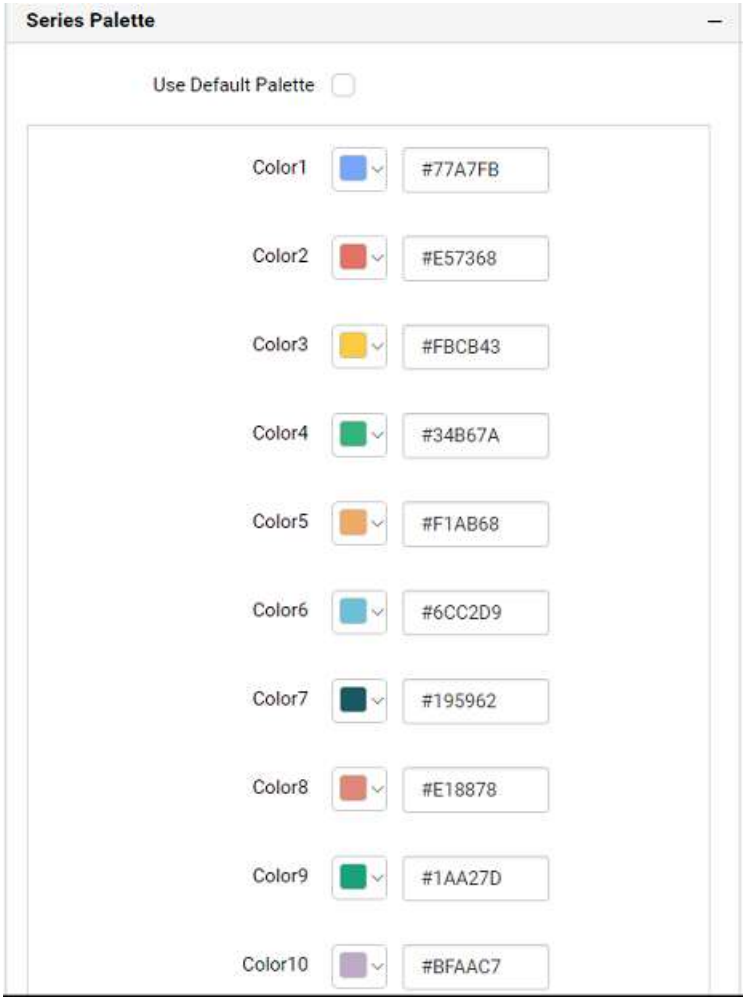
This allows you to enable/disable comment for dashboard widget. For more details refer here

Series Palette

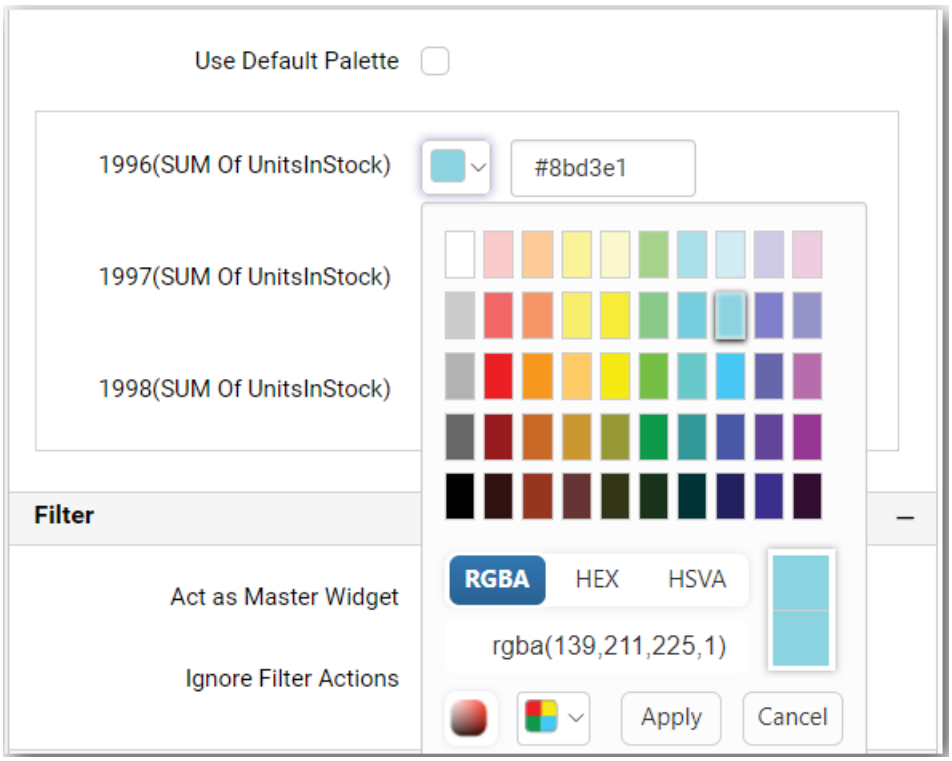
This allows you to customize the chart series color through Series Palette section.

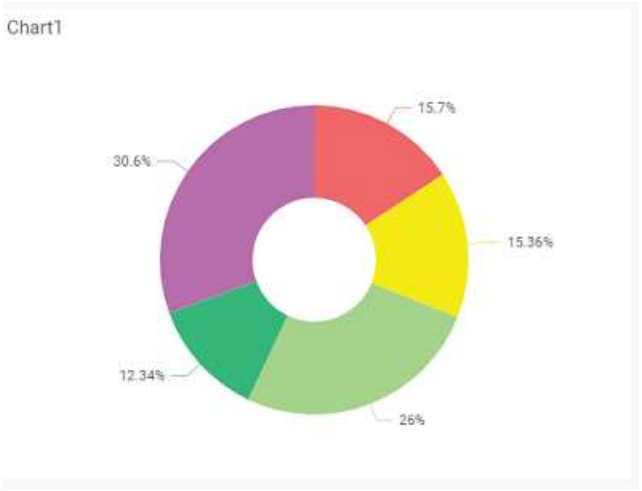
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to proportion series segments.



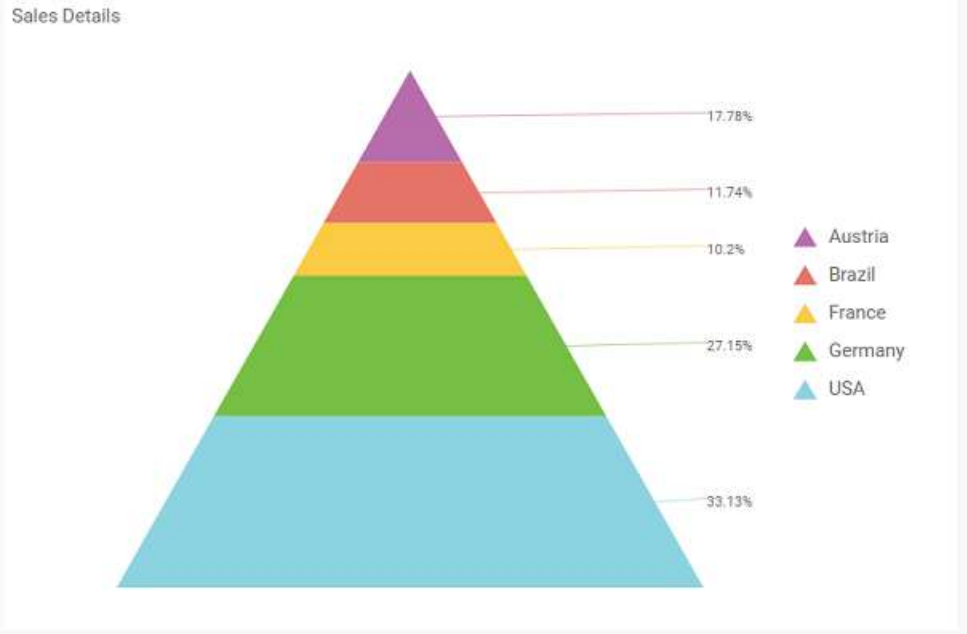
By toggle off the Use Default Palette, you can customize the proportion series segments colors. This section shows a palette of colors. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Pyramid Chart

Pyramid Chart allows you to make proportional comparison between values showcased as progressively increasing manner. To plot a pyramid chart, a minimum requirement of 1 value and 1 column is needed.

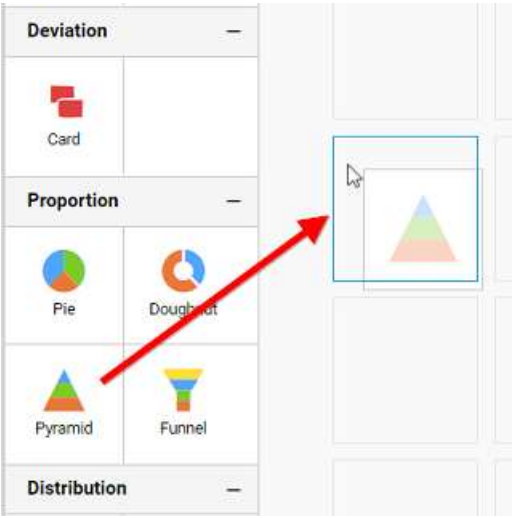


How to configure table data to pyramid Chart?

Pyramid Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

To configure data into pyramid chart follow the steps

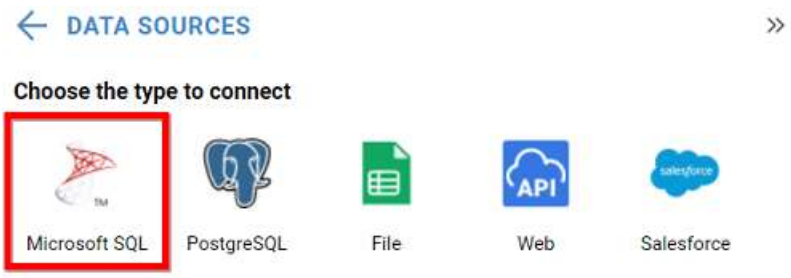
Drag and drop the Pyramid chart into canvas and resize it to your required size.



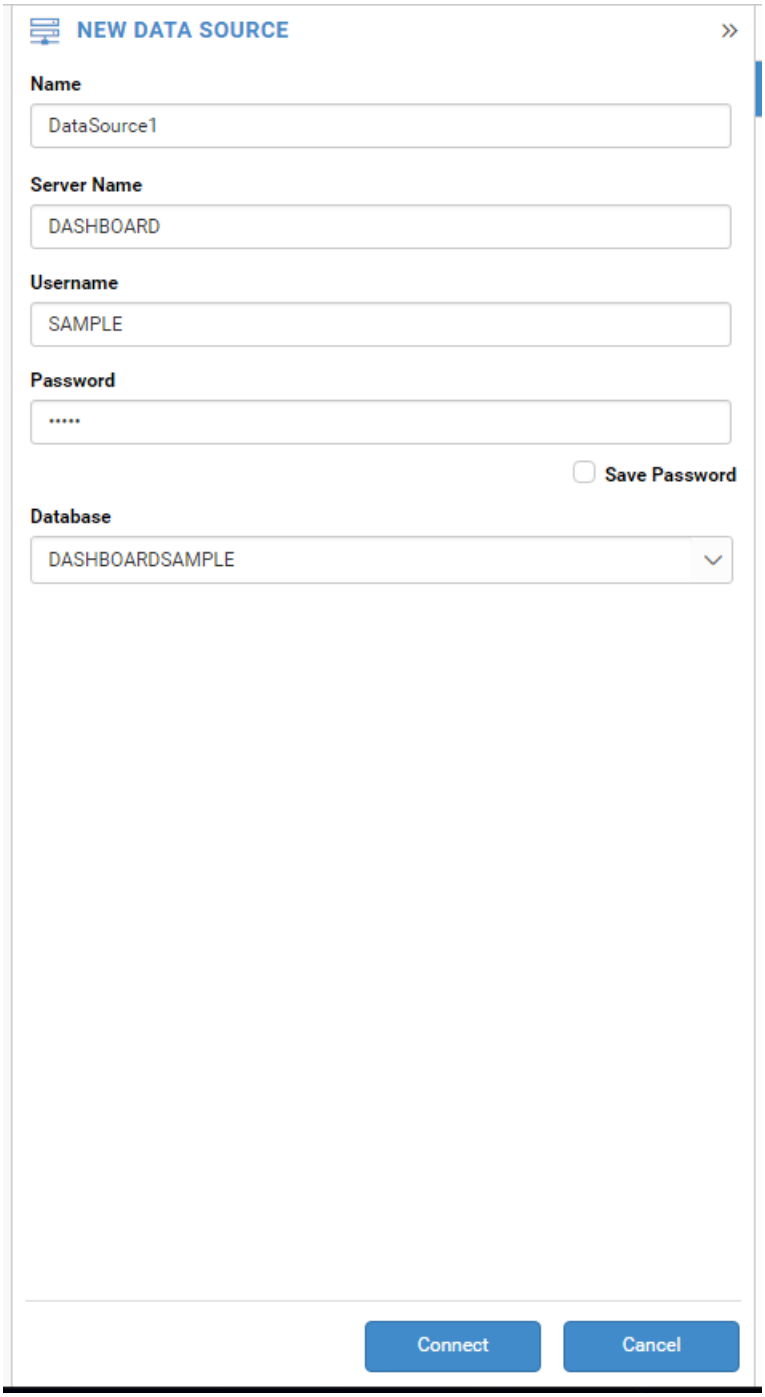
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

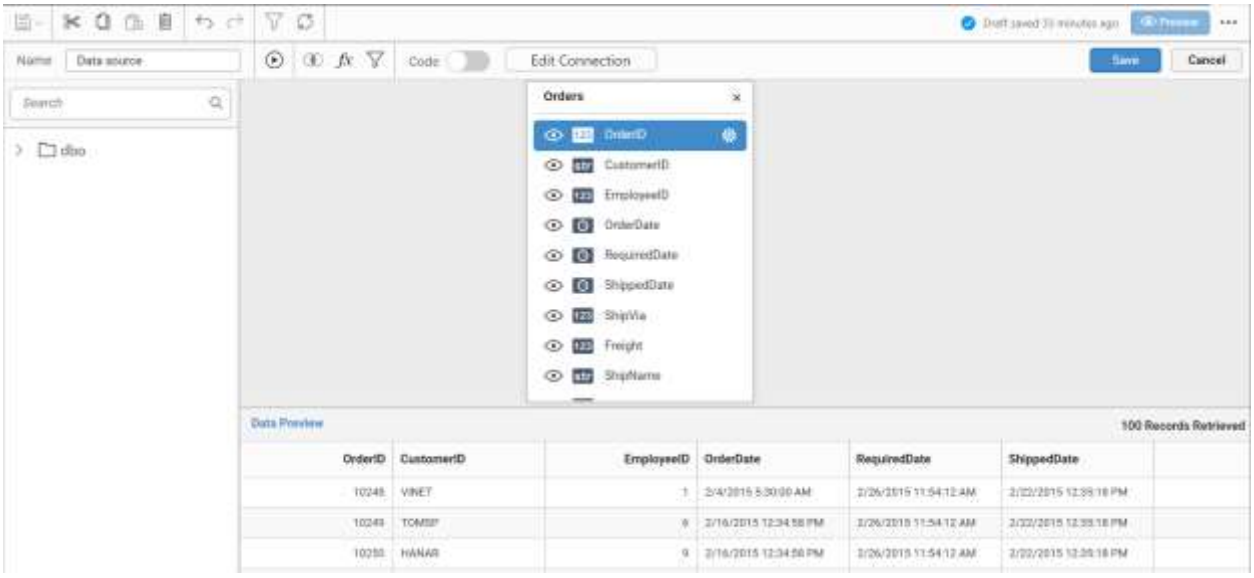
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



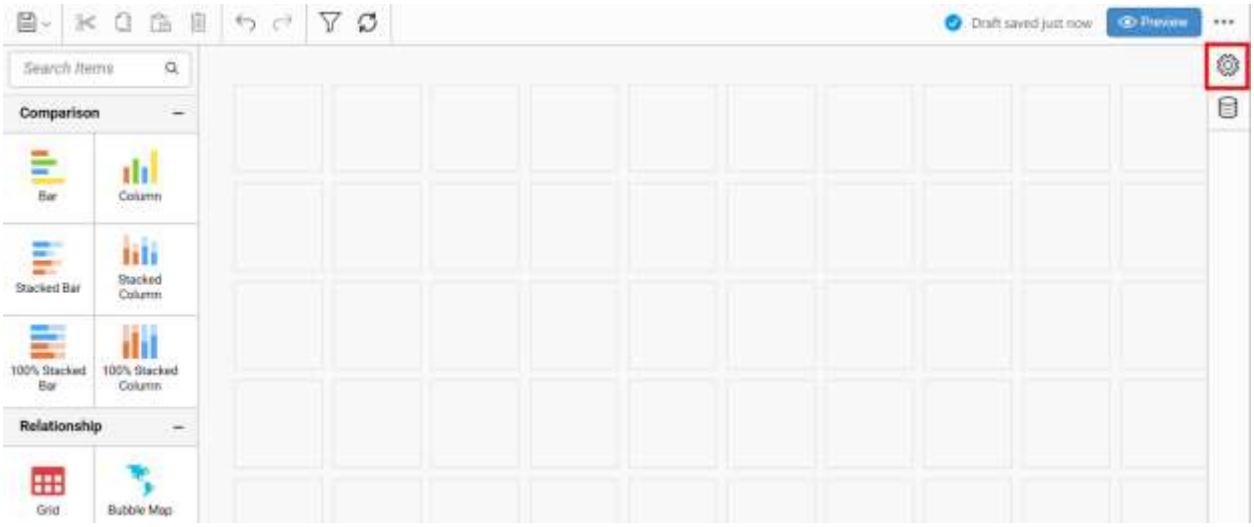
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



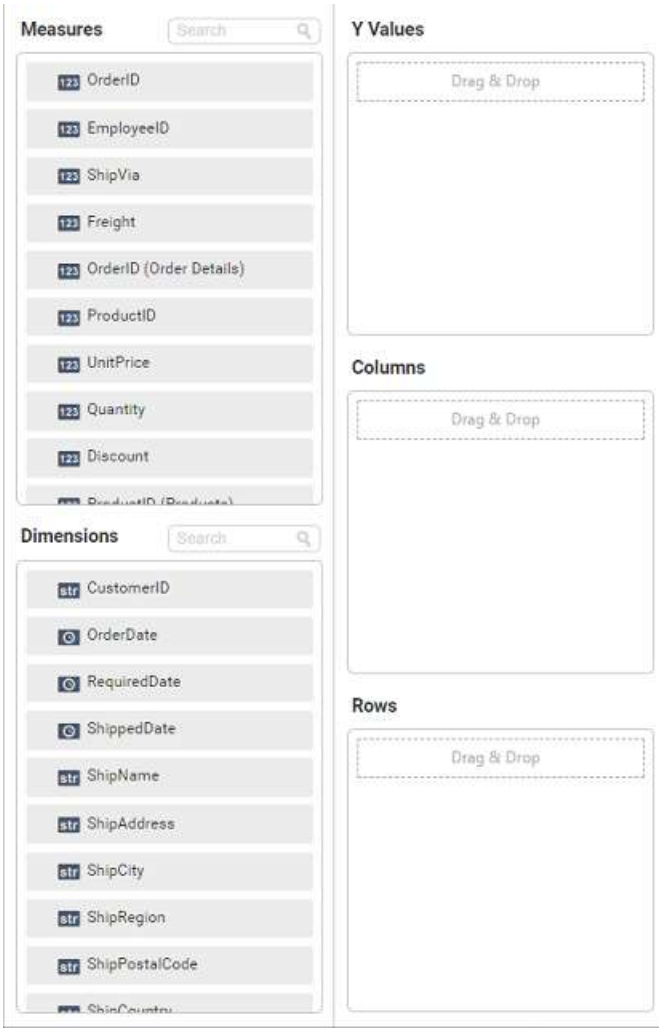
Drag your preferred table or view from the left pane from data design view, click Save button.



Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.



The data tab will be opened with available measures and dimensions from the connected data source

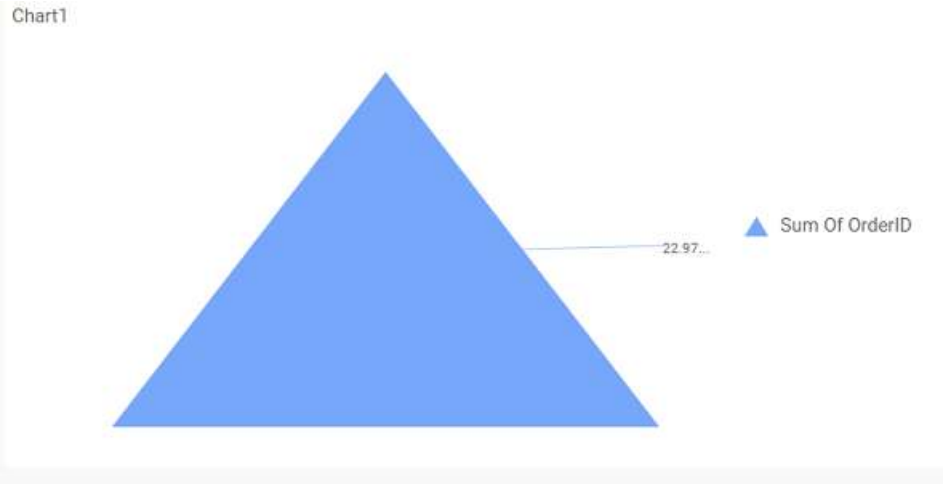


You can add the required data from Measures and Dimensions into required field.

Adding Y Values

You can add more than one Measures into Values field by drag and drop the required measure.

Now, the pyramid chart will be rendered like this

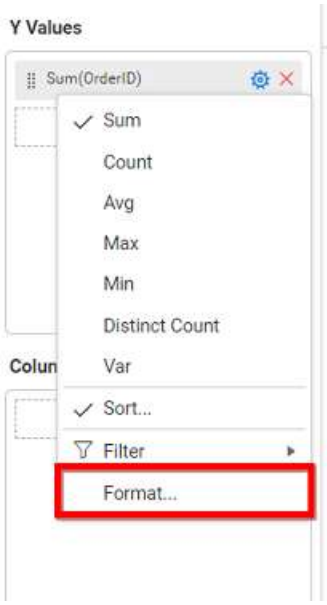


Click the Settings option to change required summary type from the available summary types shown in Settings.

You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

You can filter the data to be displayed in funnel chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

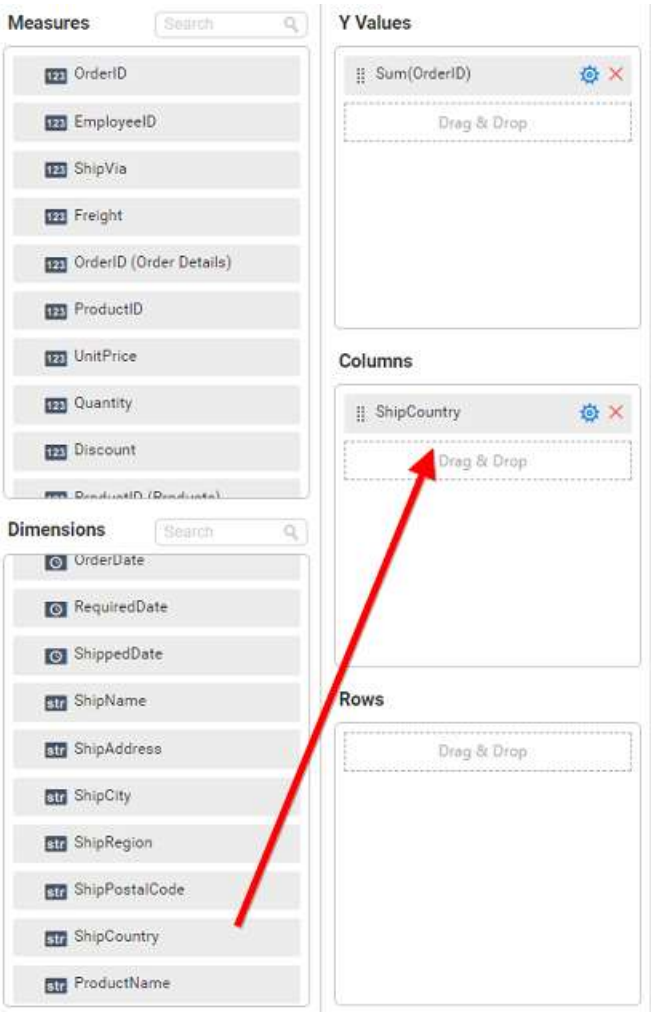


To remove the added value fields click highlighted button.

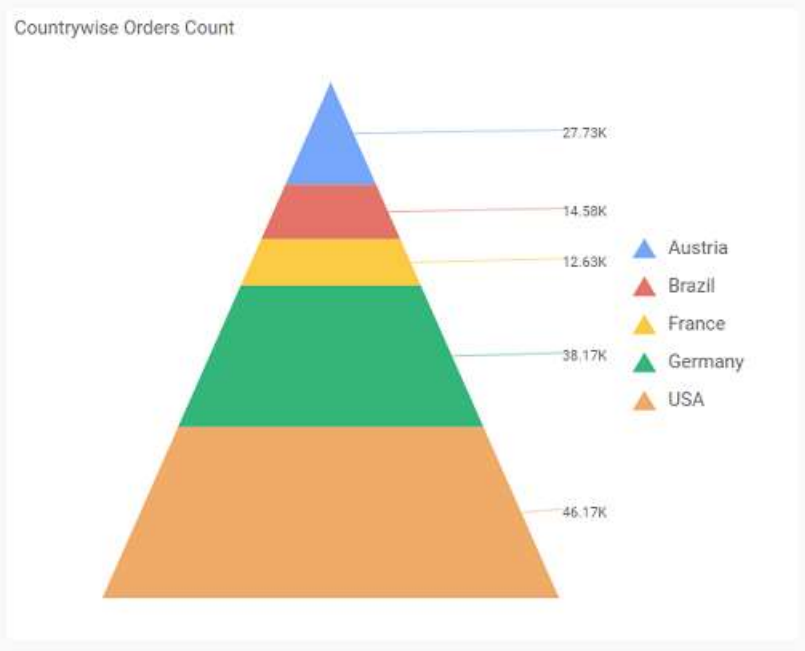
You can add more than one column from Dimensions field into Values field.

Adding Columns

You can add more than one value into Columns field.



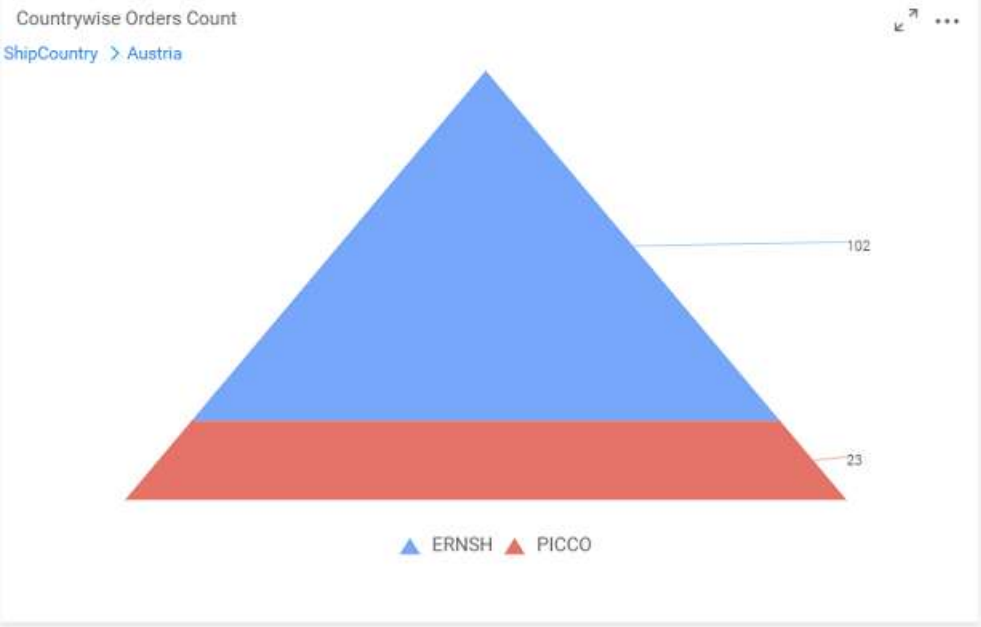
Pyramid chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.



You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.

You can apply filters by selecting filter in settings

Note: Filter will be set by default for top 5 records.

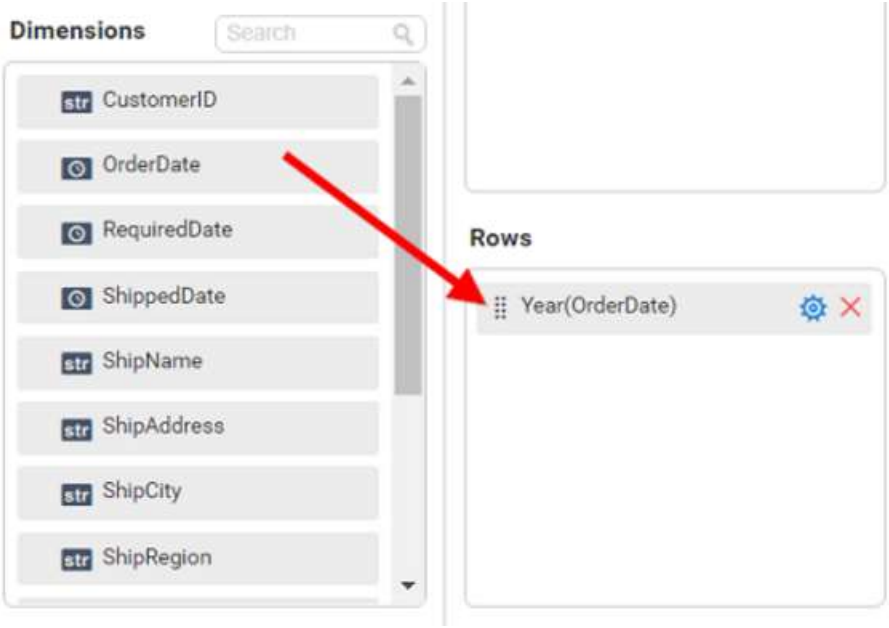
You can apply filters by selecting filters option in settings. For more details, refer filter.

To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

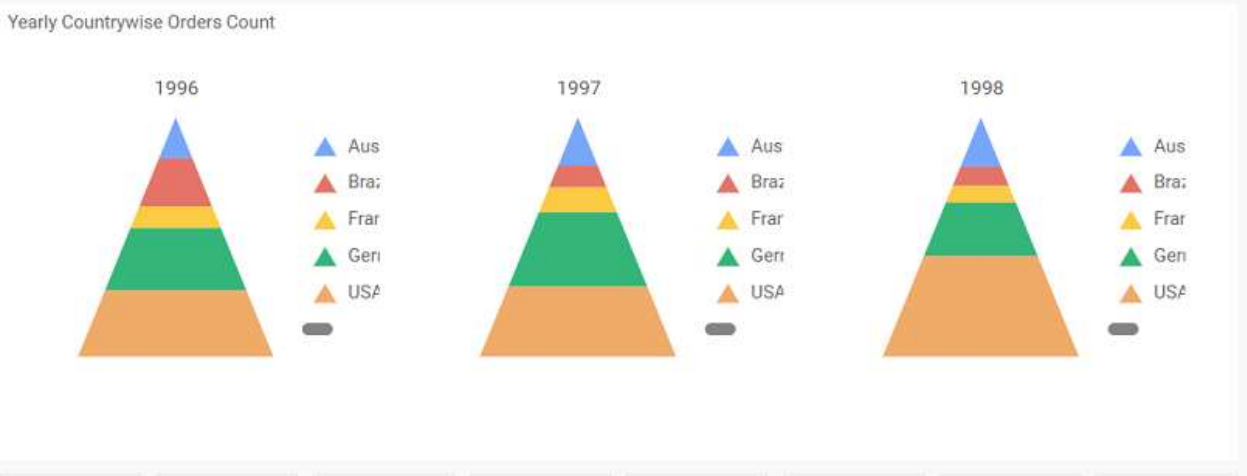
Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render pyramid chart in series.



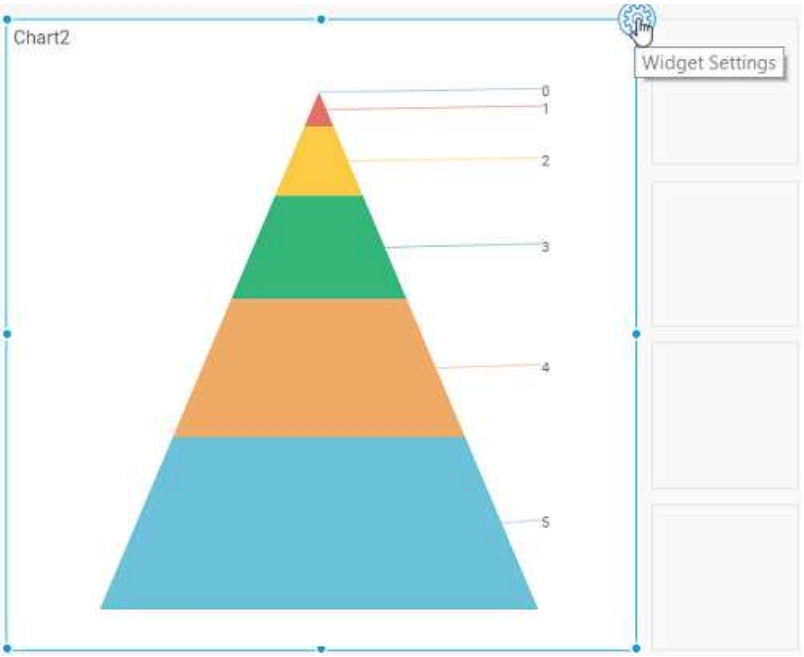
Scroll down to see all charts.

How to format pyramid chart?

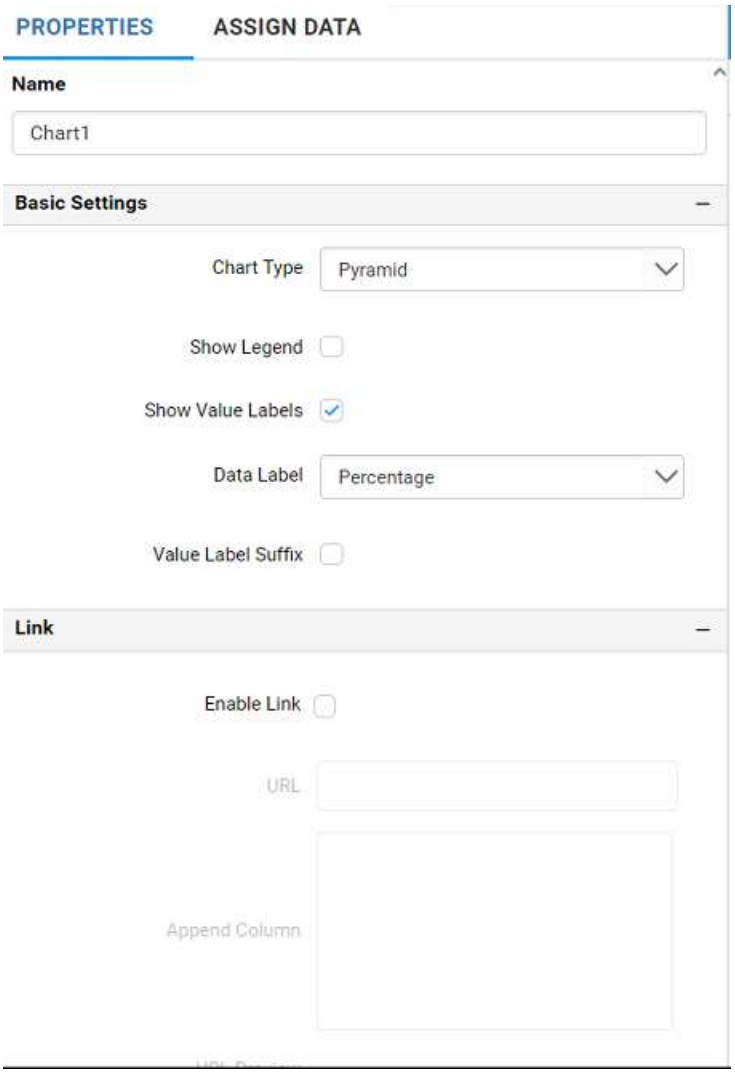
You can format the pyramid chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into pyramid chart follow the steps

1. Drag and drop the pyramid chart into canvas and resize it to your required size.
2. Configure the data into pyramid chart.
3. Focus on the pyramid chart and click on widget settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this pyramid chart widget.

Basic Settings

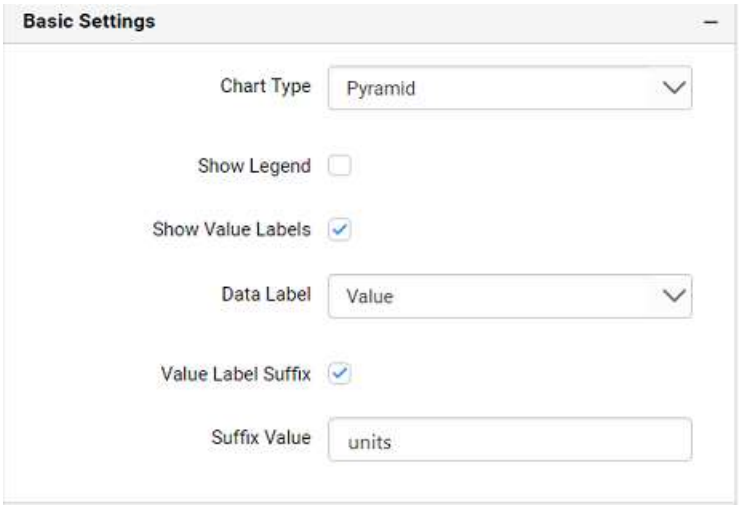


Chart Type

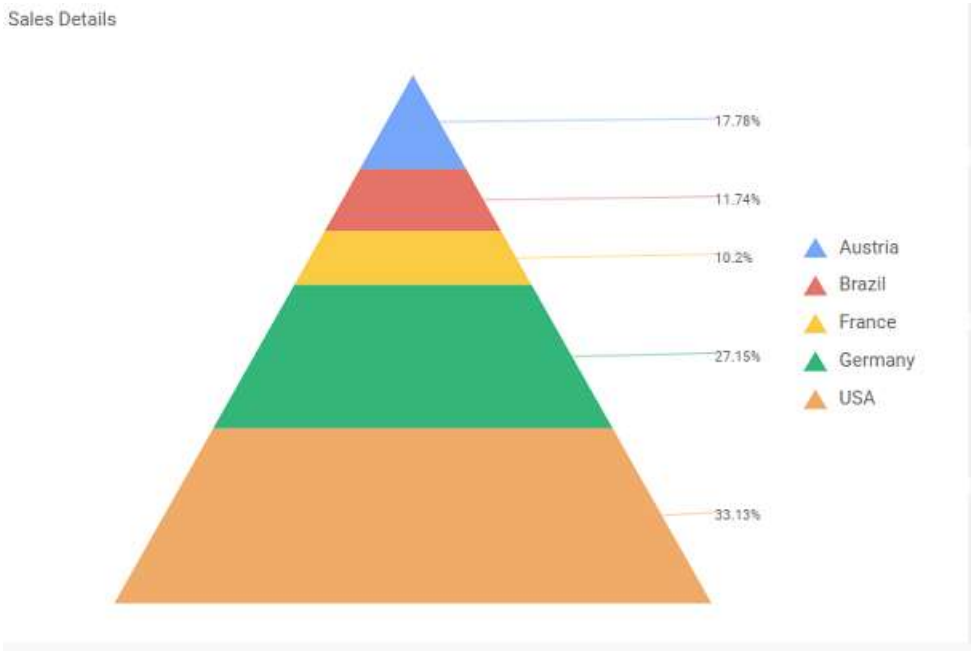
This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the series rendering in animated mode.

Show Legend

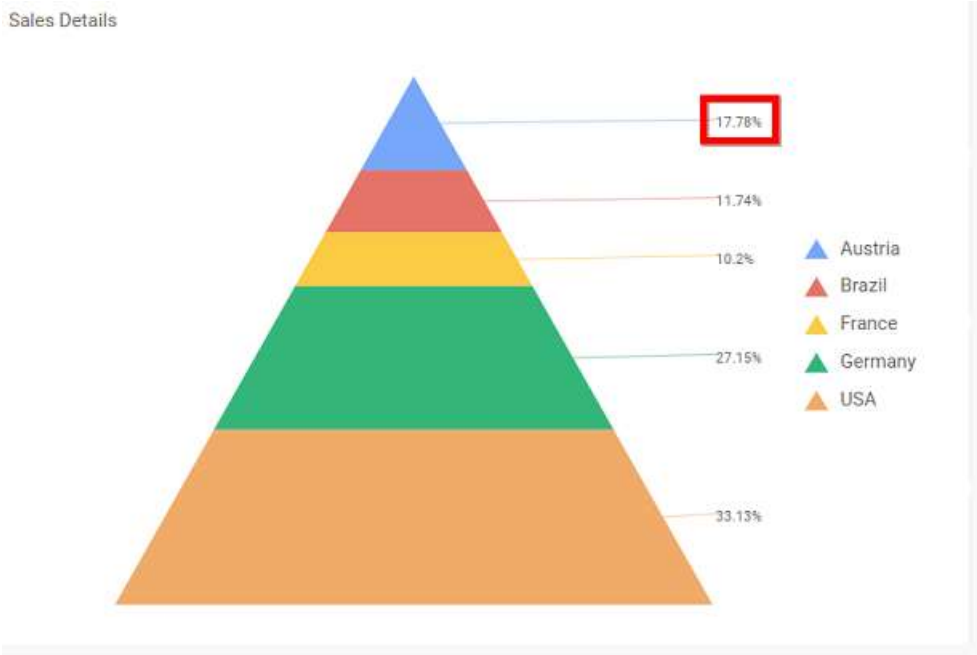
A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Show Value Labels

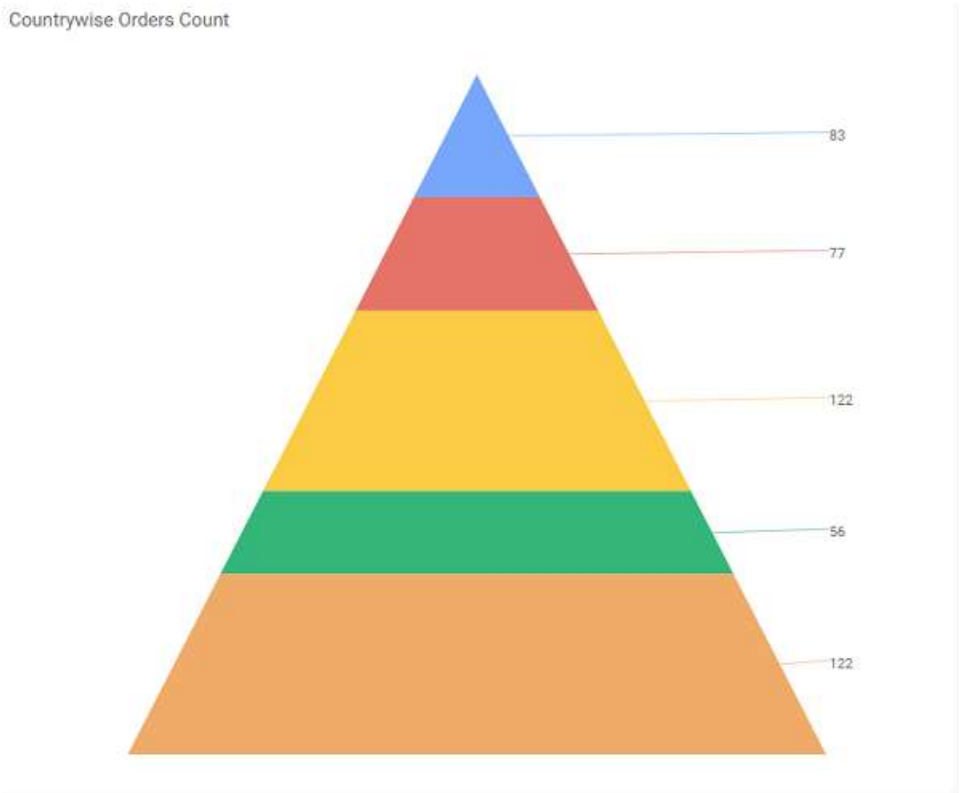
This allows you to toggle the visibility of value labels.



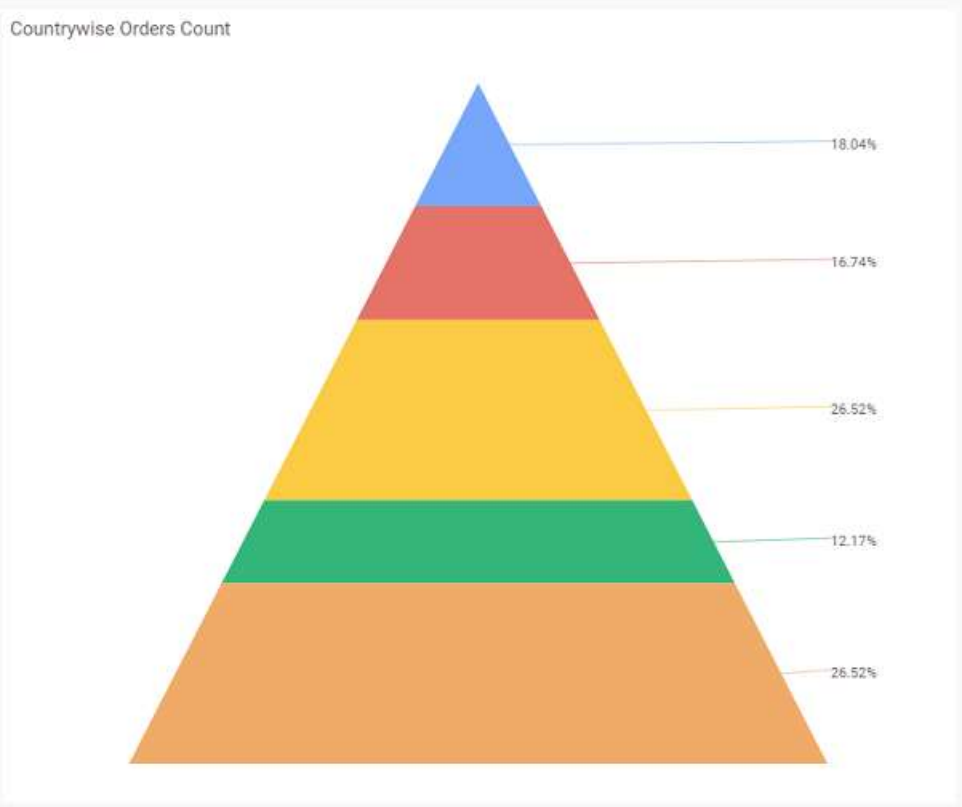
Data Label Value

This allows you to define the display format either as value or as percentage or both.

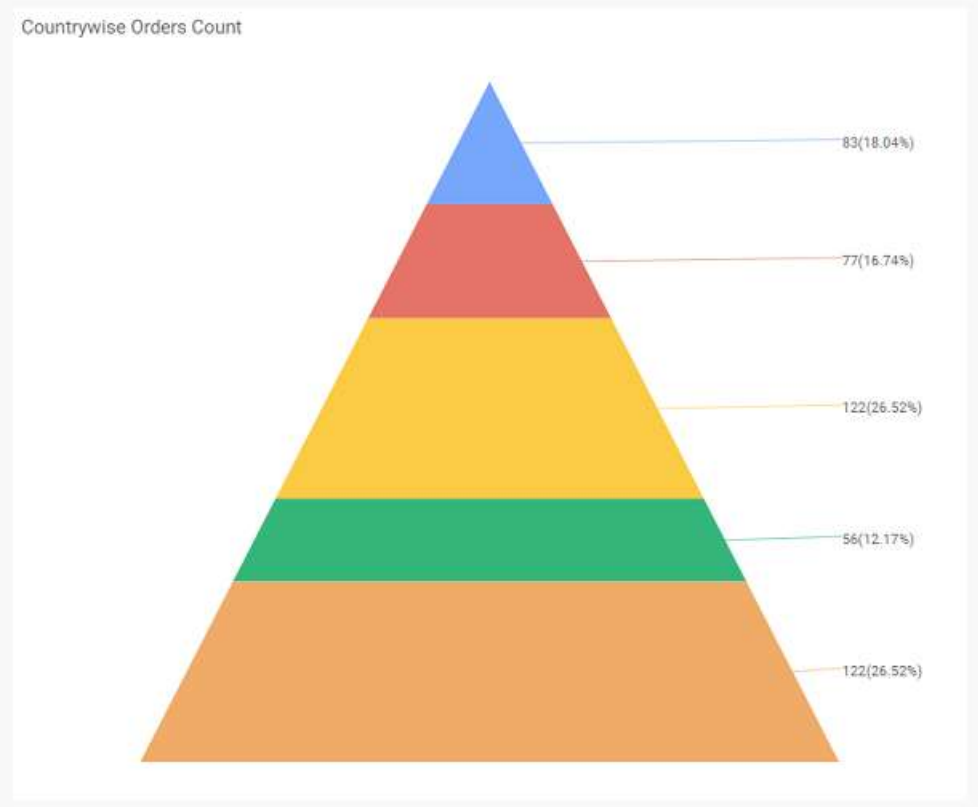
Value



Percentage



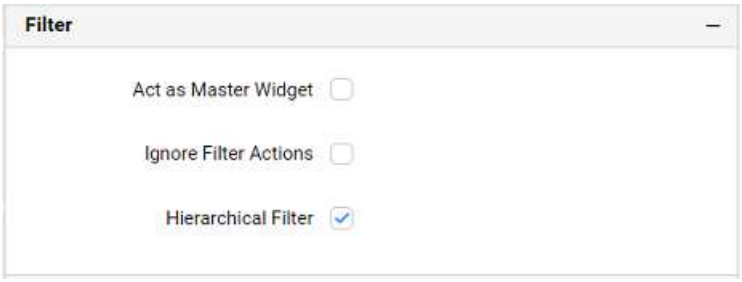
Value and Percentage



Value Labels Suffix

Allows you to set suffix to the value labels.

Filter



Act as Master Widget

This allows you to define this pyramid chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

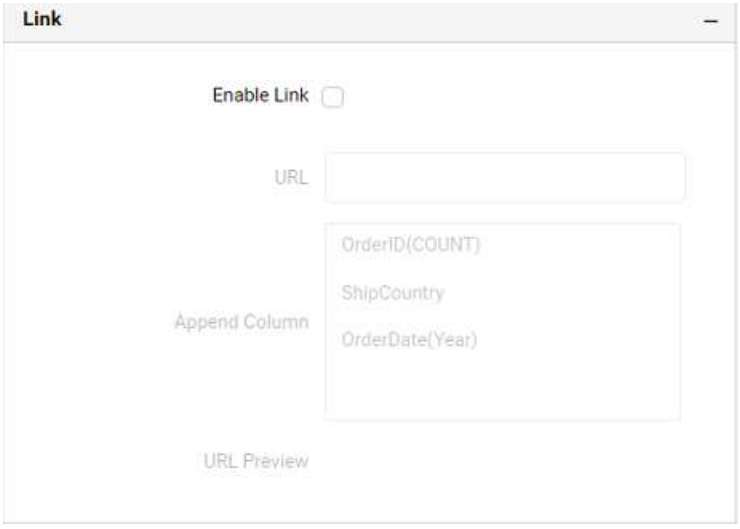
This allows you to define this pyramid chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

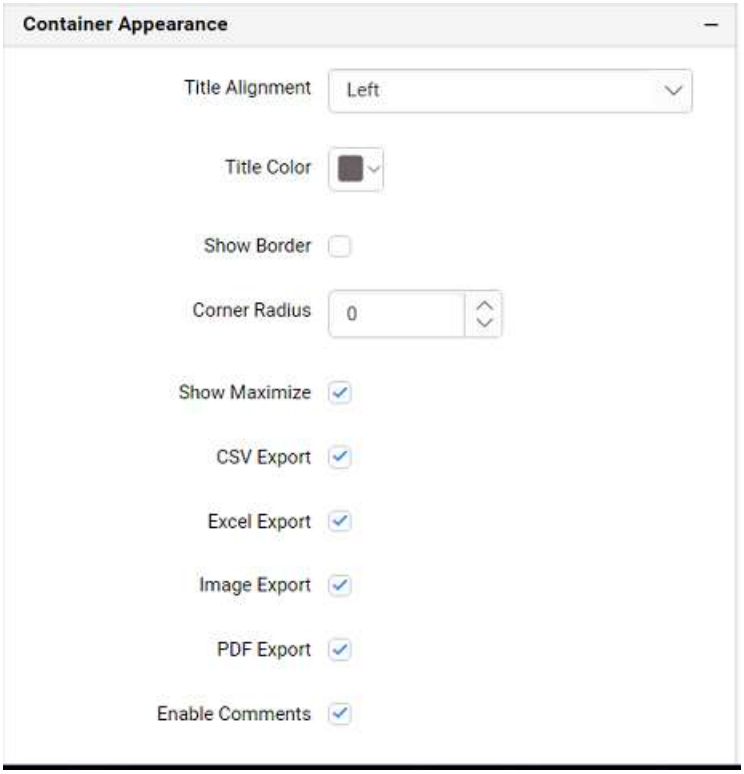
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply different text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if **Show Border** property is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this pyramid chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this pyramid chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this pyramid chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

Image Export

This allows you to enable/disable the image export option for this pyramid chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

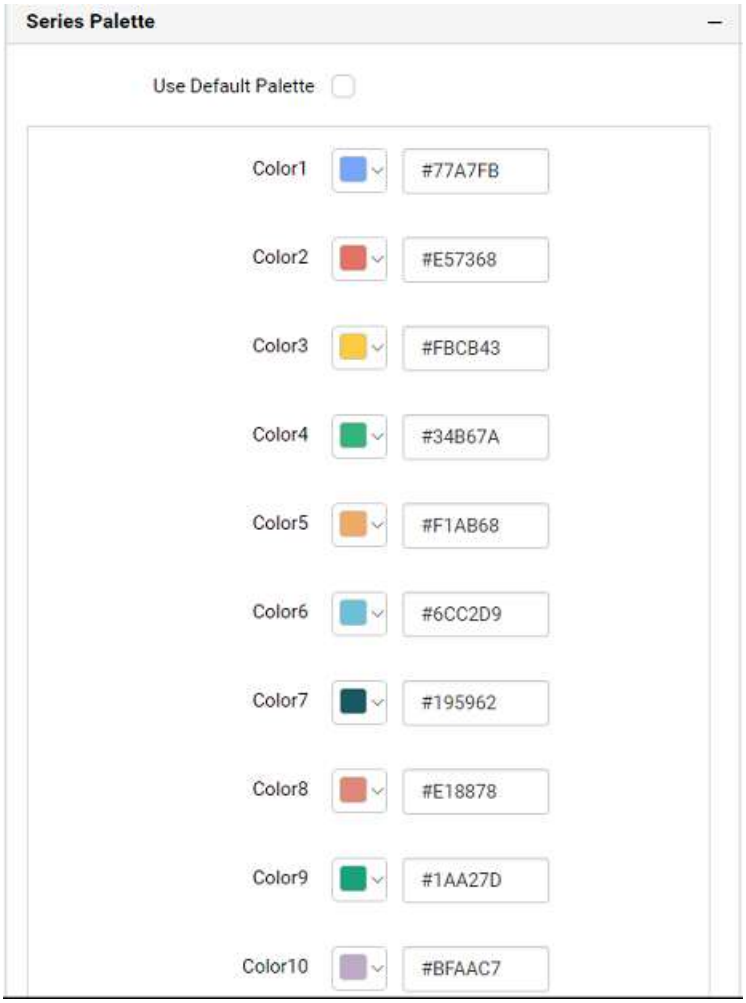
This allows you to enable/disable comment for dashboard widget. For more details refer here

Series Palette

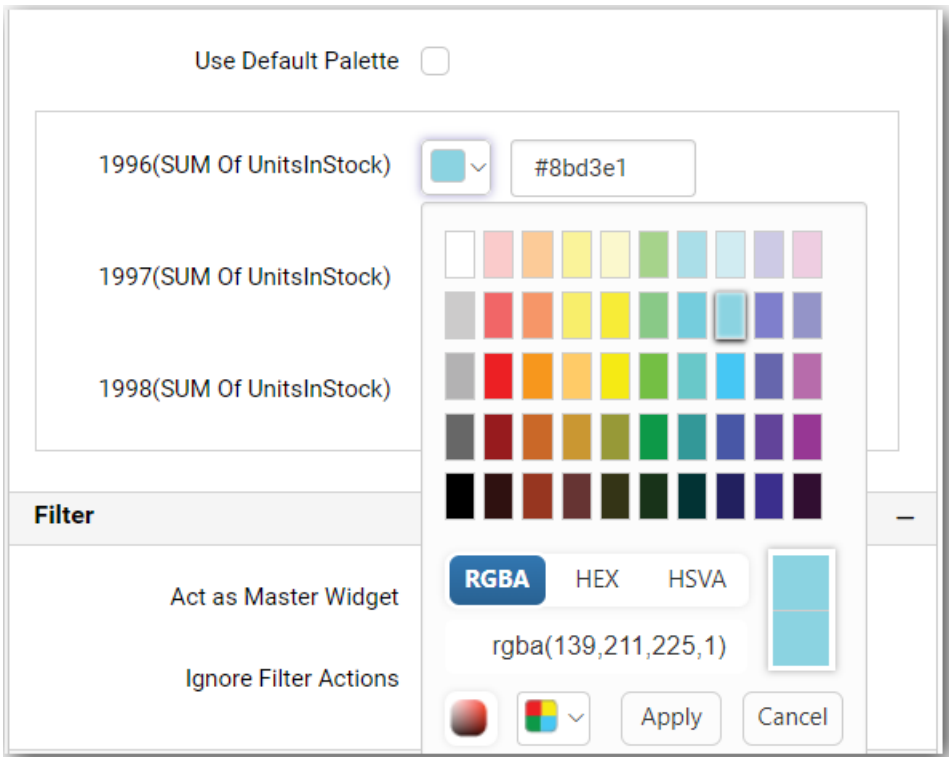
This allows you to customize the chart series color through Series Palette section.

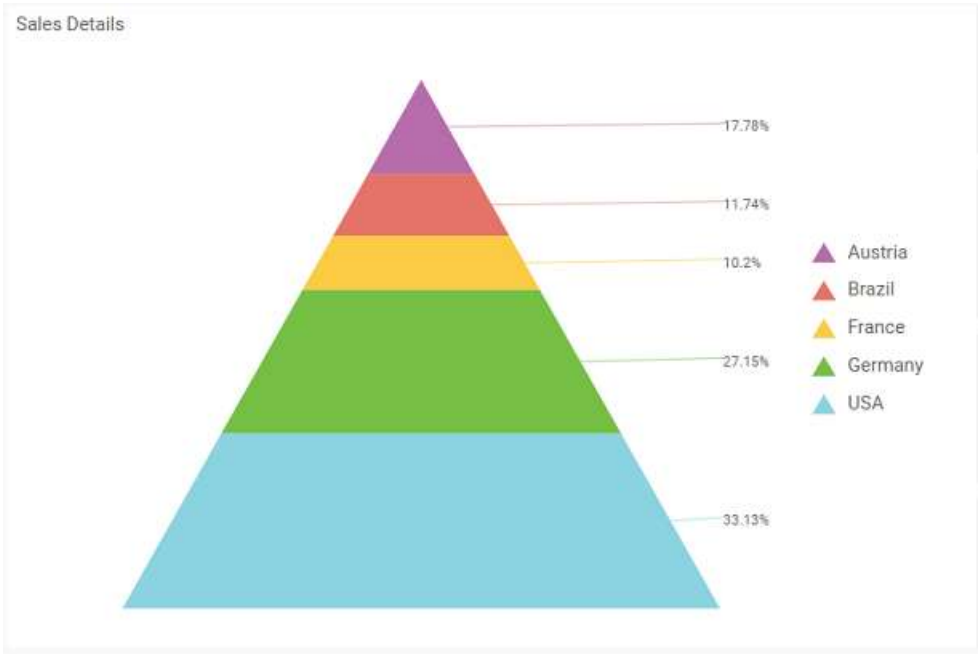
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to proportion series segments.



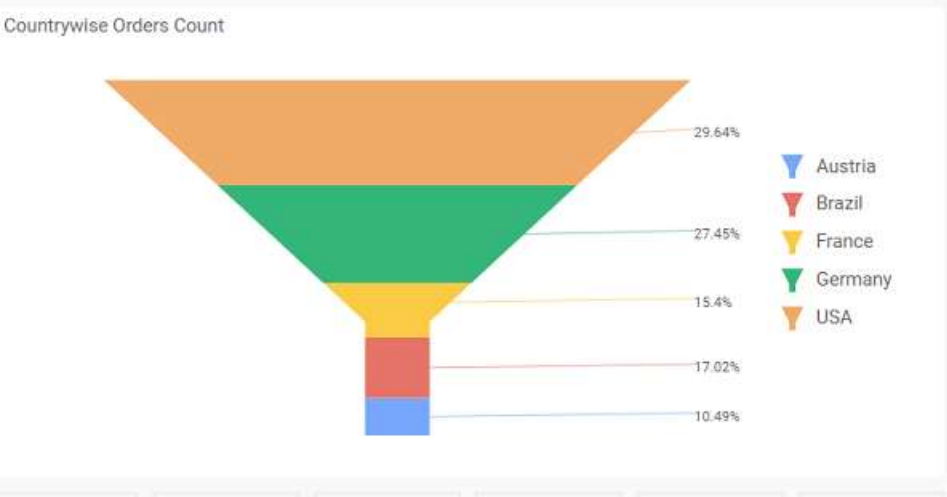
By toggle off the Use Default Palette, you can customize the proportion series segments colors. This section shows a palette of colors. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Funnel Chart

Funnel Chart allows you to make proportional comparison between values showcased as progressively decreasing manner. To plot a funnel chart, a minimum requirement of 1 value and 1 column is needed.

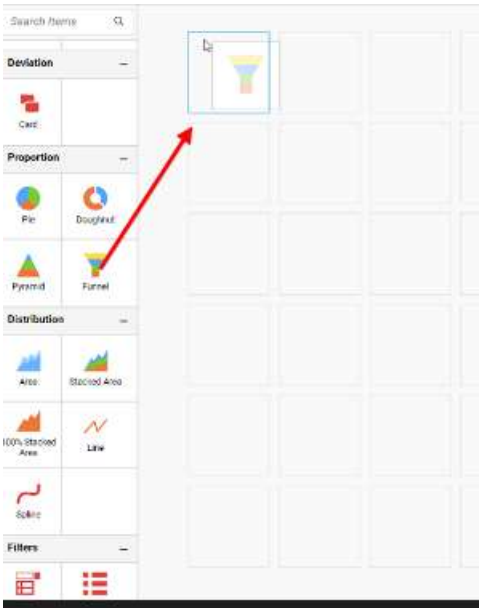


How to configure table data to funnel chart?

Funnel Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

To configure data into funnel chart follow the steps

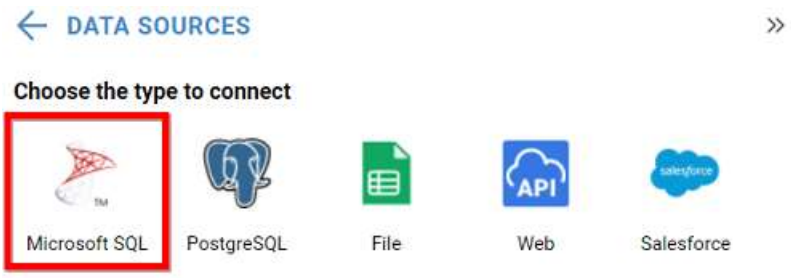
Drag and drop the funnel chart into canvas and resize it to your required size.



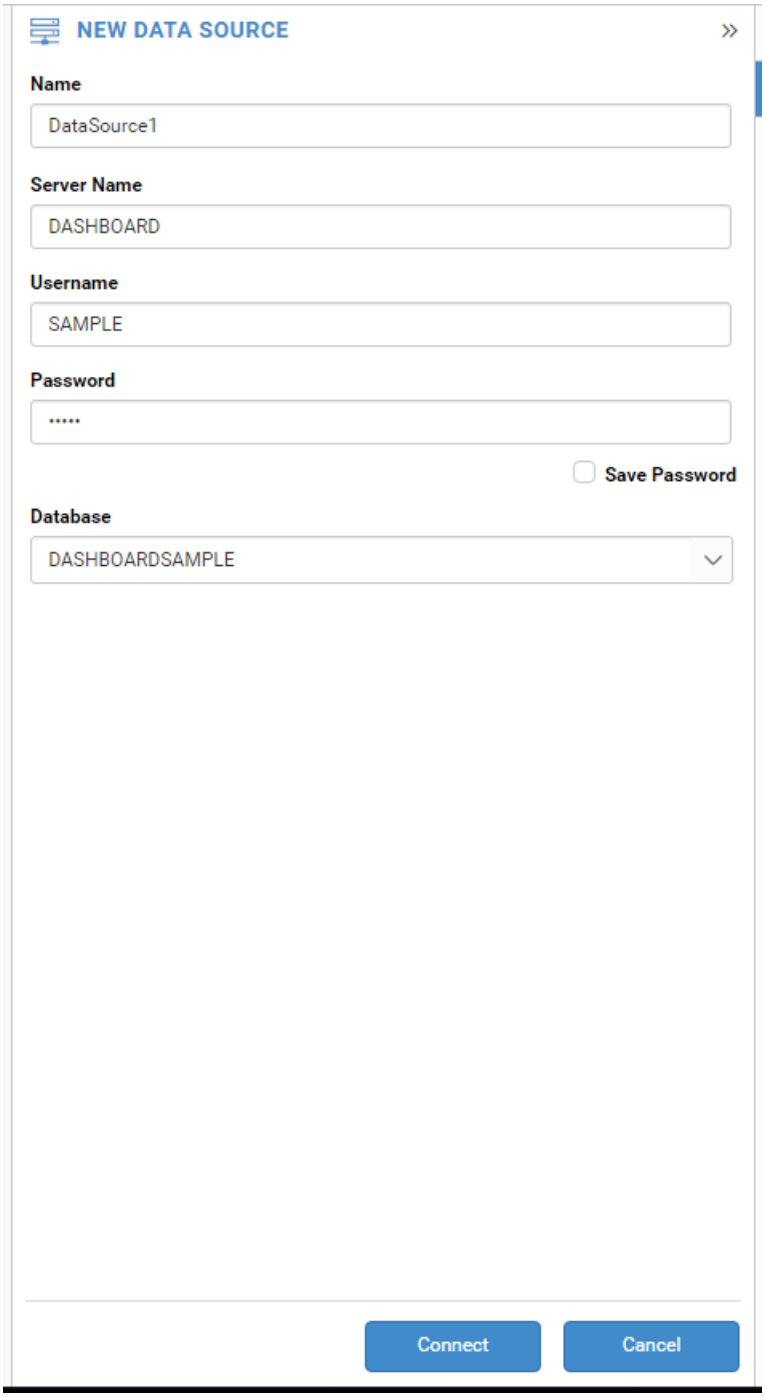
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

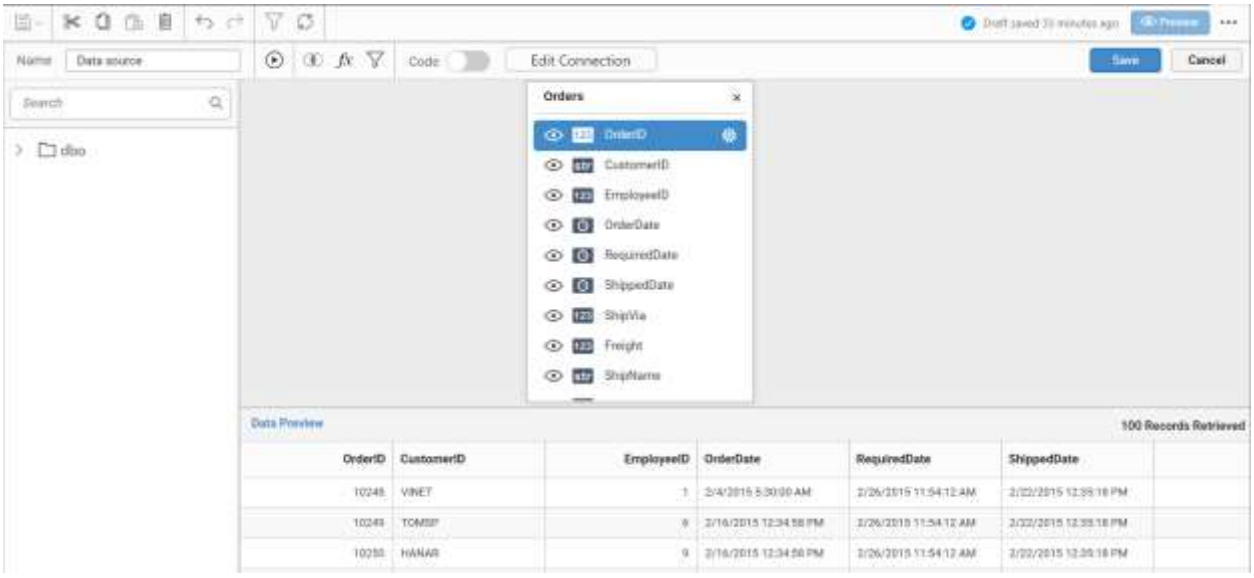
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



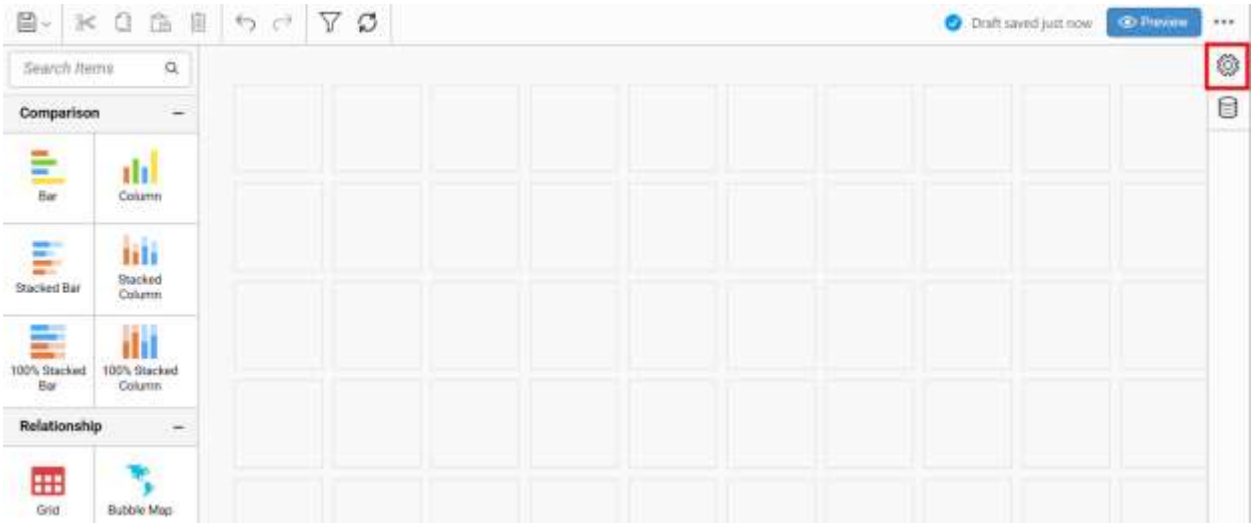
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



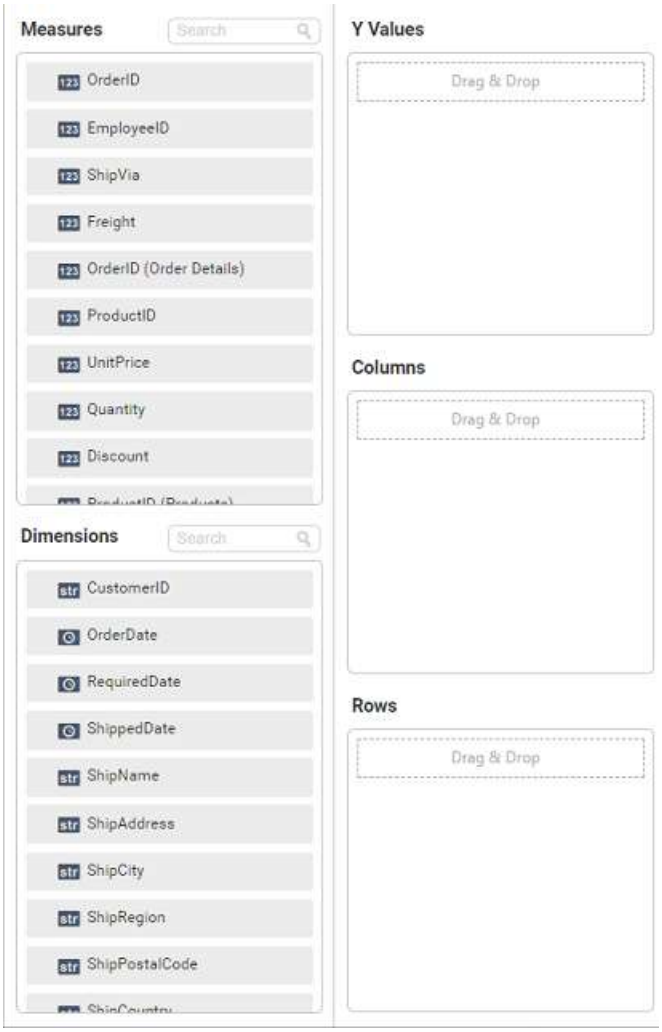
Drag your preferred table or view from the left pane from data design view, click Save button.



Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.



The data tab will be opened with available measures and dimensions from the connected data source

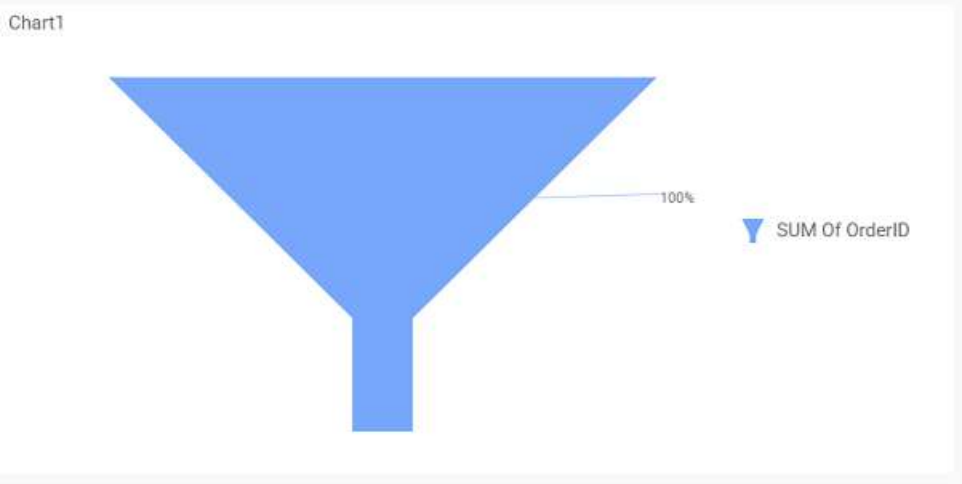


You can add the required data from Measures and Dimensions into required field.

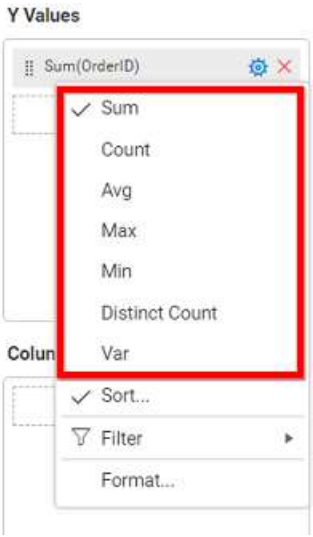
Adding Y Values

You can add more than one Measures into Y Values field by drag and drop the required measure.

Now, the funnel chart will be rendered like this



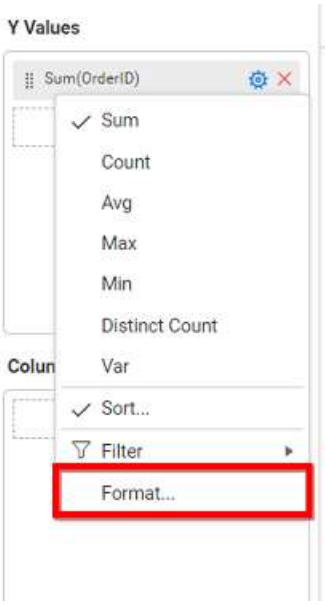
Click the Settings option to change required summary type from the available summary types shown in Settings.



You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

You can filter the data to be displayed in funnel chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

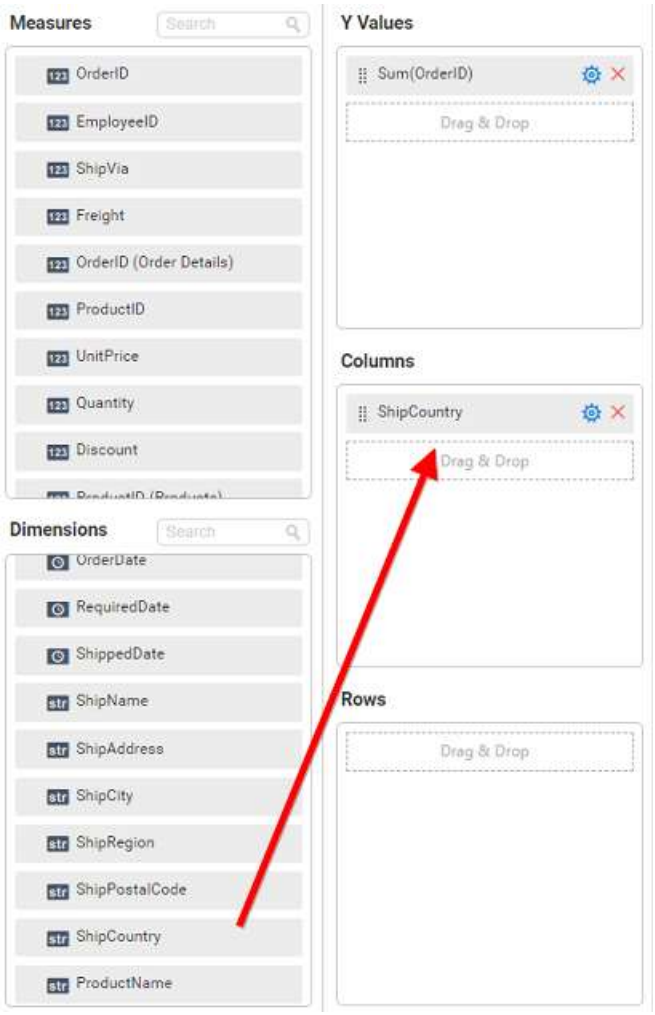


To remove the added value fields click highlighted button.

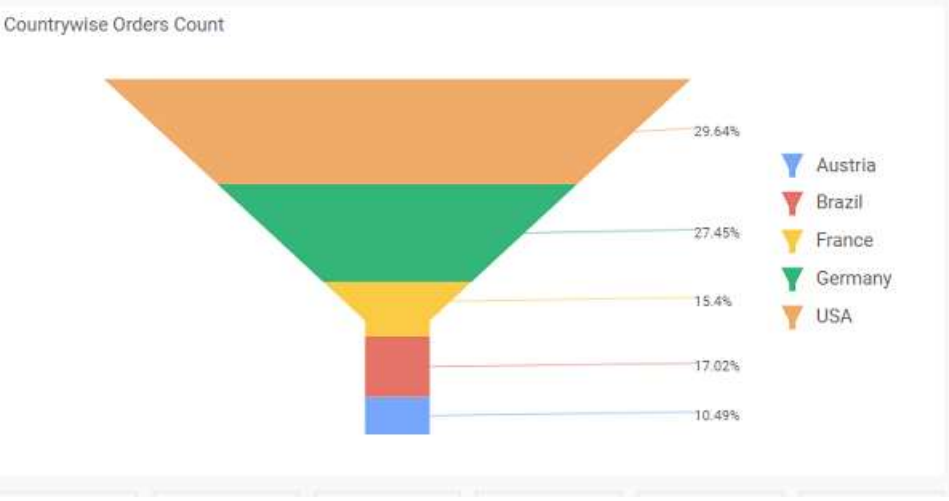
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



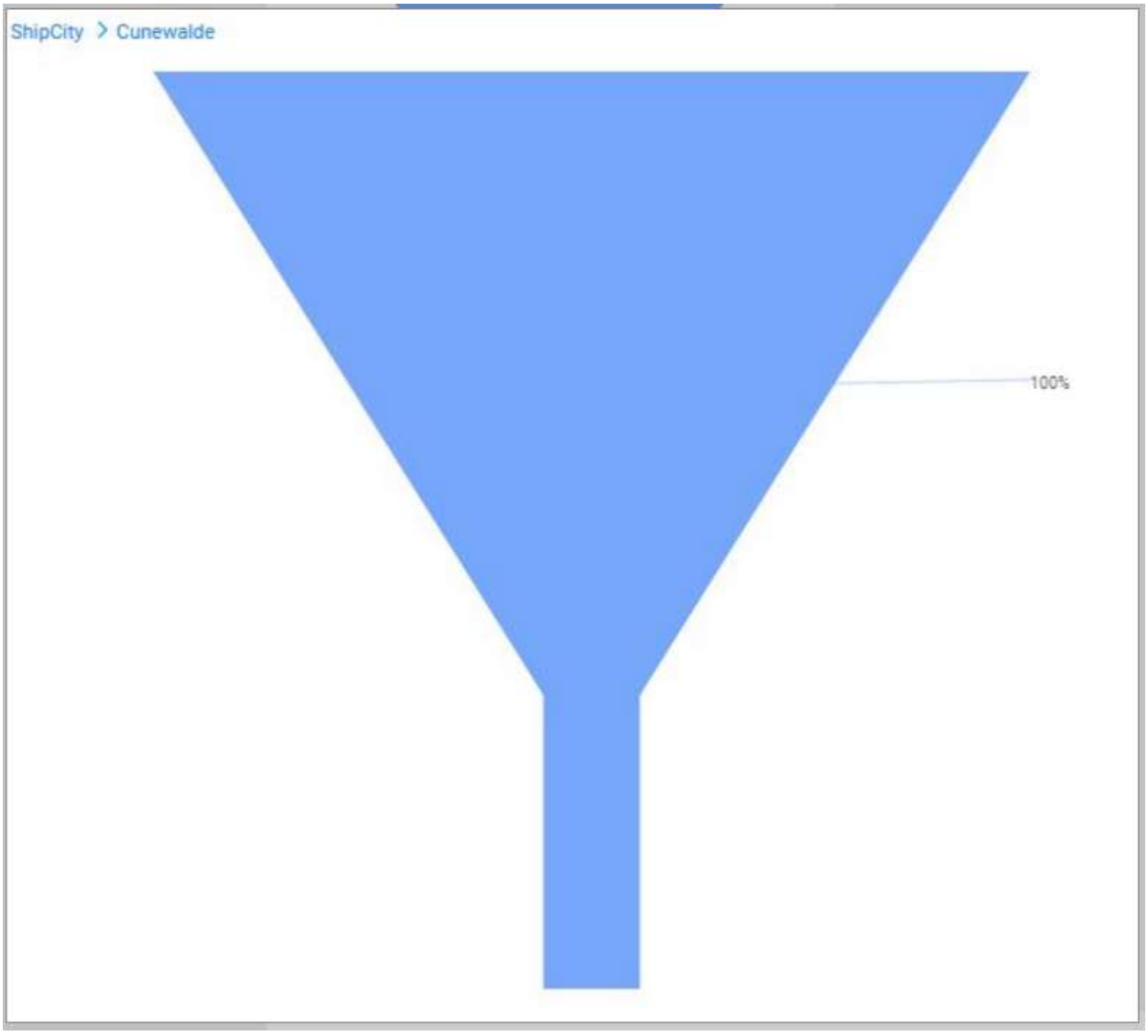
Funnel chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

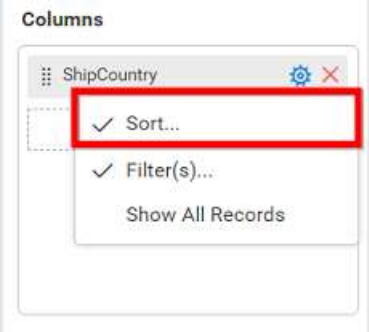
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.



You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.

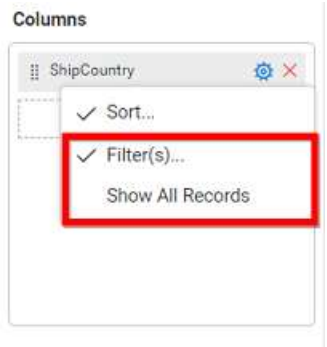


You can apply filters by selecting filter in settings

Note: Filter will be set by default for top 5 records.

You can apply filters by selecting filters option in settings. For more details, refer filter.

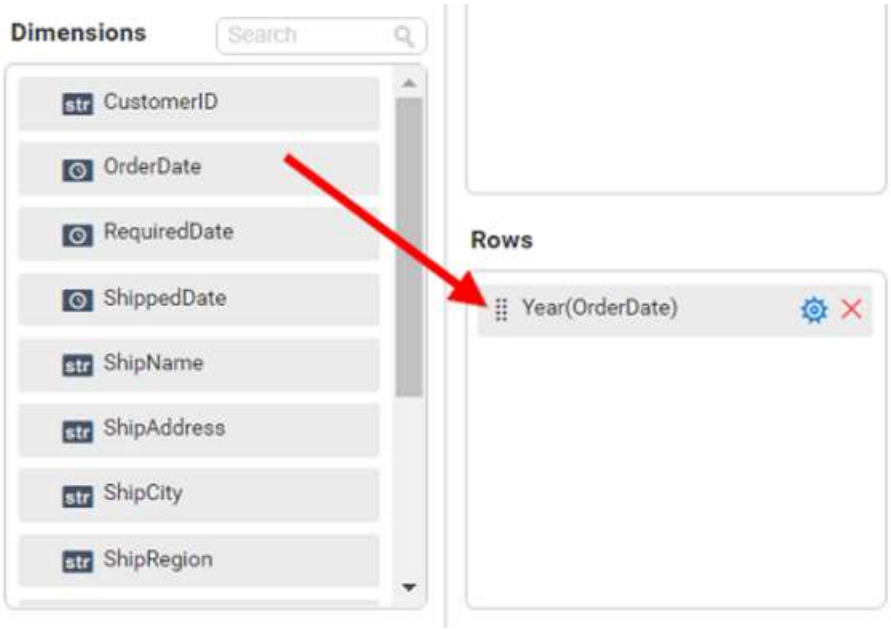
To show all records click on Show All Records.



Similarly you can add the Measures and Expression Columns into column field.

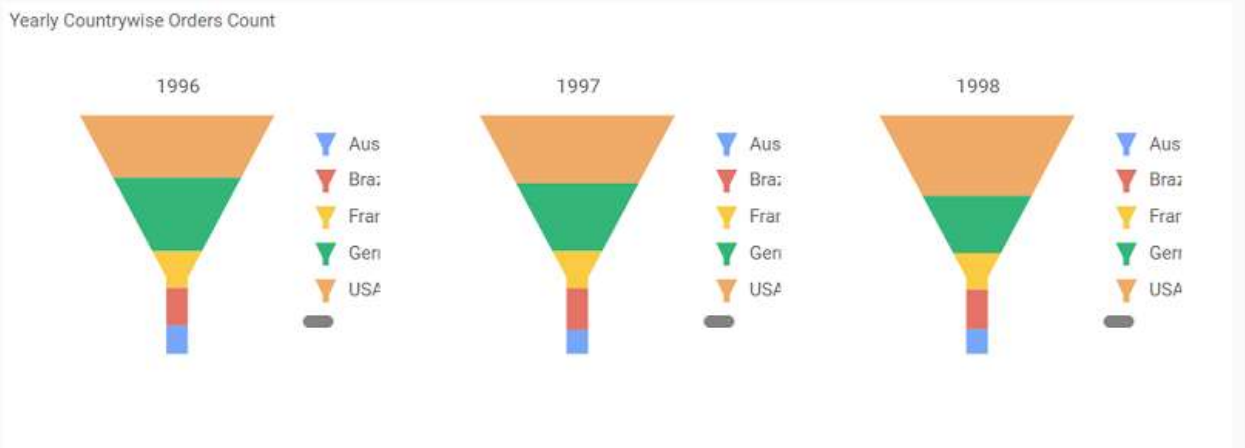
Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render funnel chart in series.



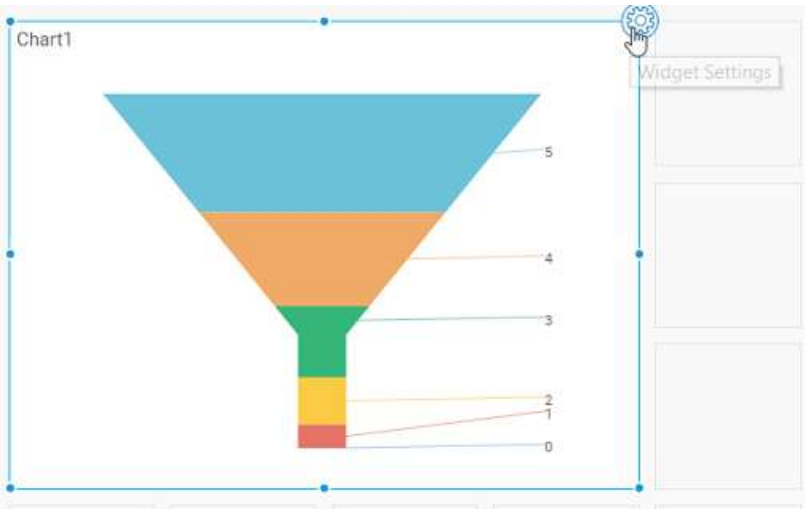
Scroll down to see all charts.

How to format funnel chart?

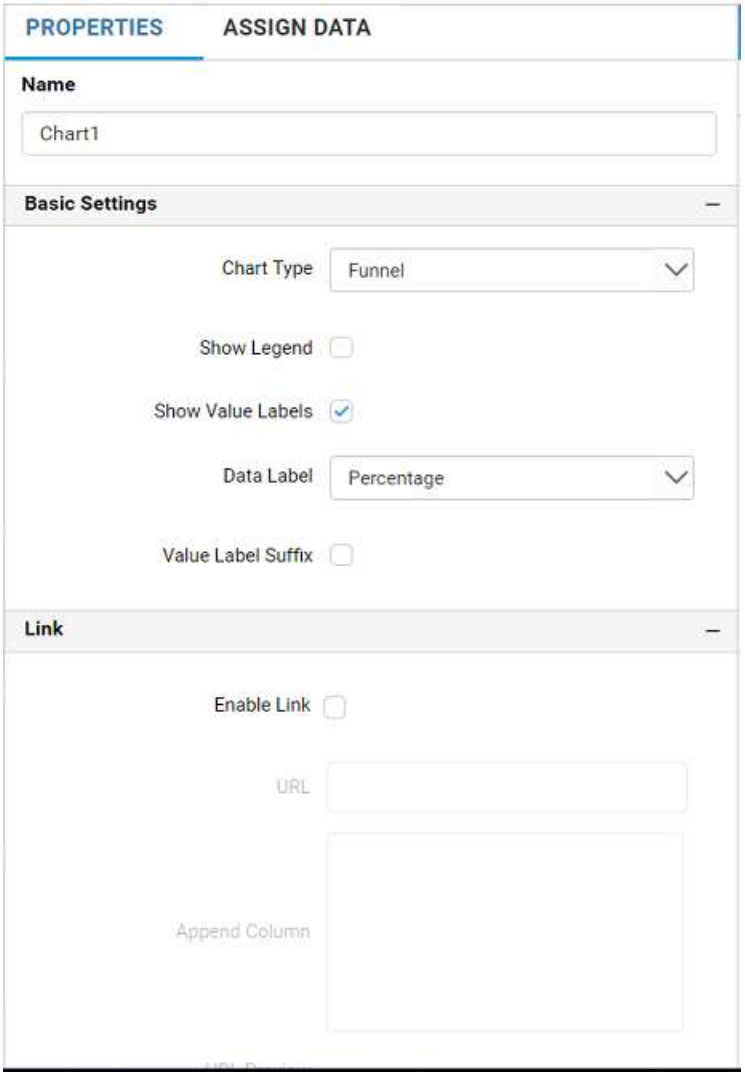
You can format the funnel chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into funnel chart follow the steps

- Drag and drop the funnel chart into canvas and resize it to your required size.
- Configure the data into Funnel chart.
- Focus on the funnel chart and click on widget settings.

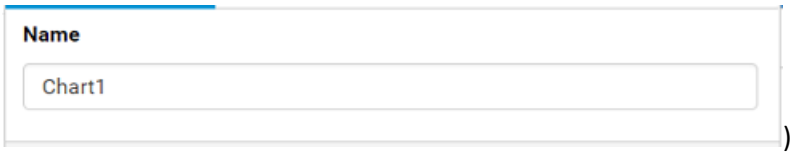


The property window will be opened



You can see the list of properties available for the widget with default value.

General Settings



)

Name

This allows you to change the title for this funnel chart widget.

Basic Settings

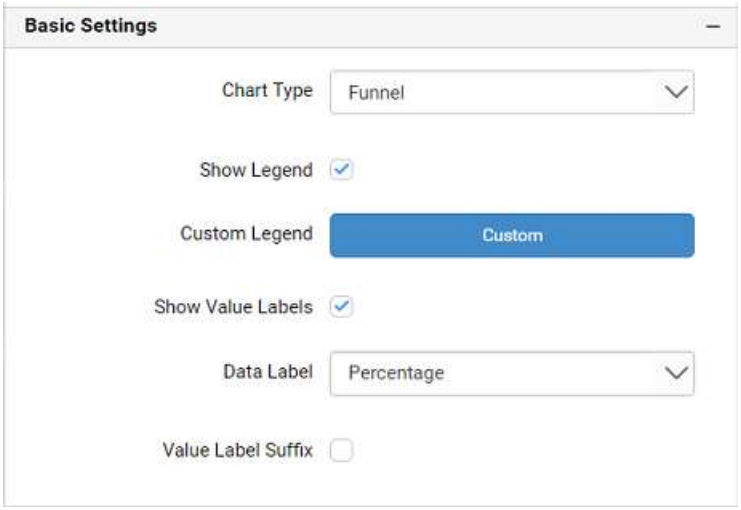


Chart Type

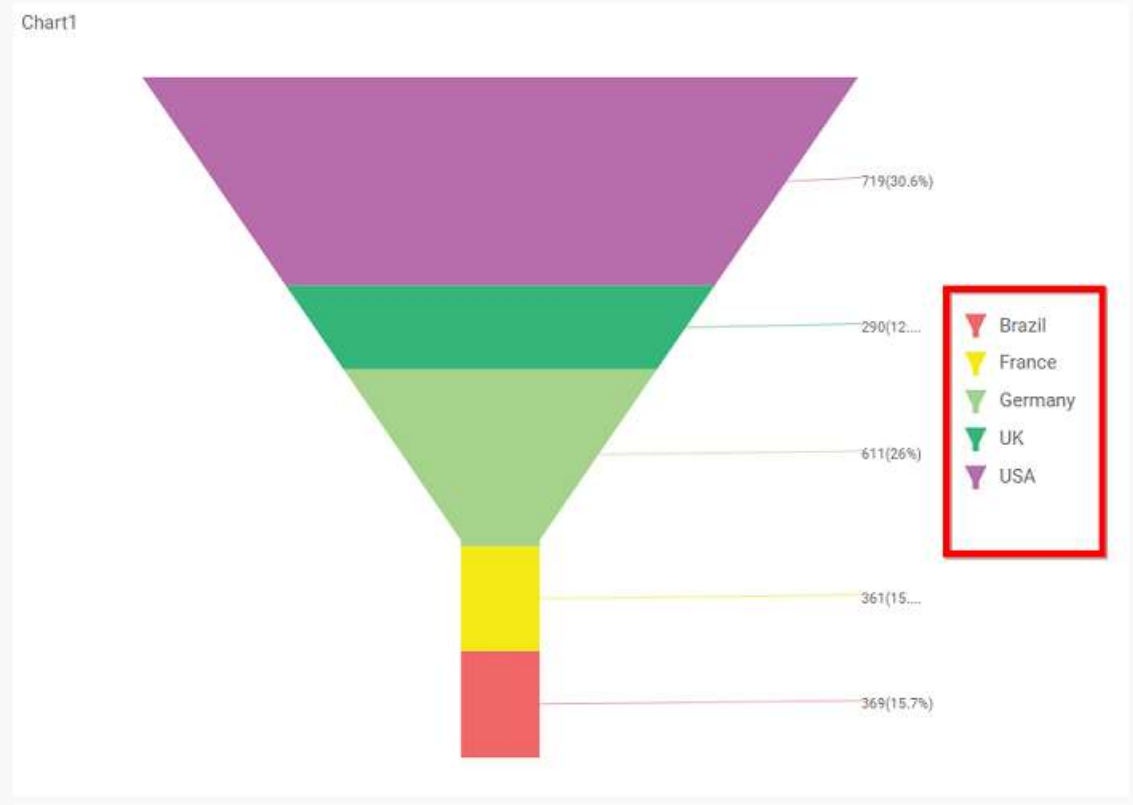
This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the series rendering in animated mode.

Show Legend

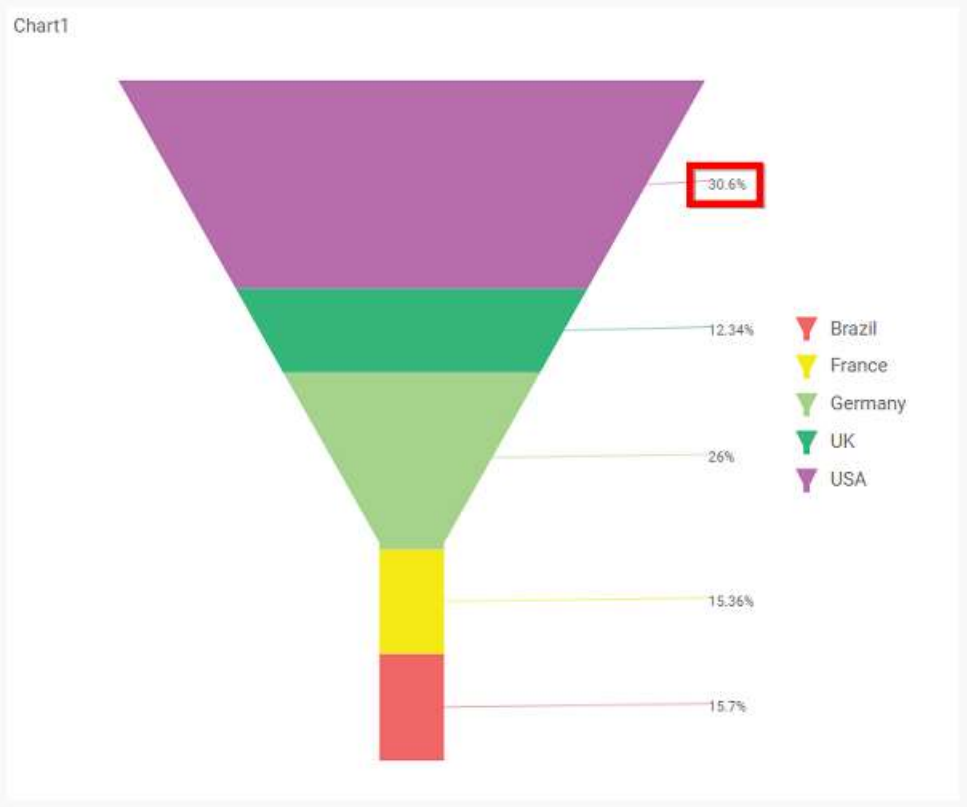
A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Show Value Labels

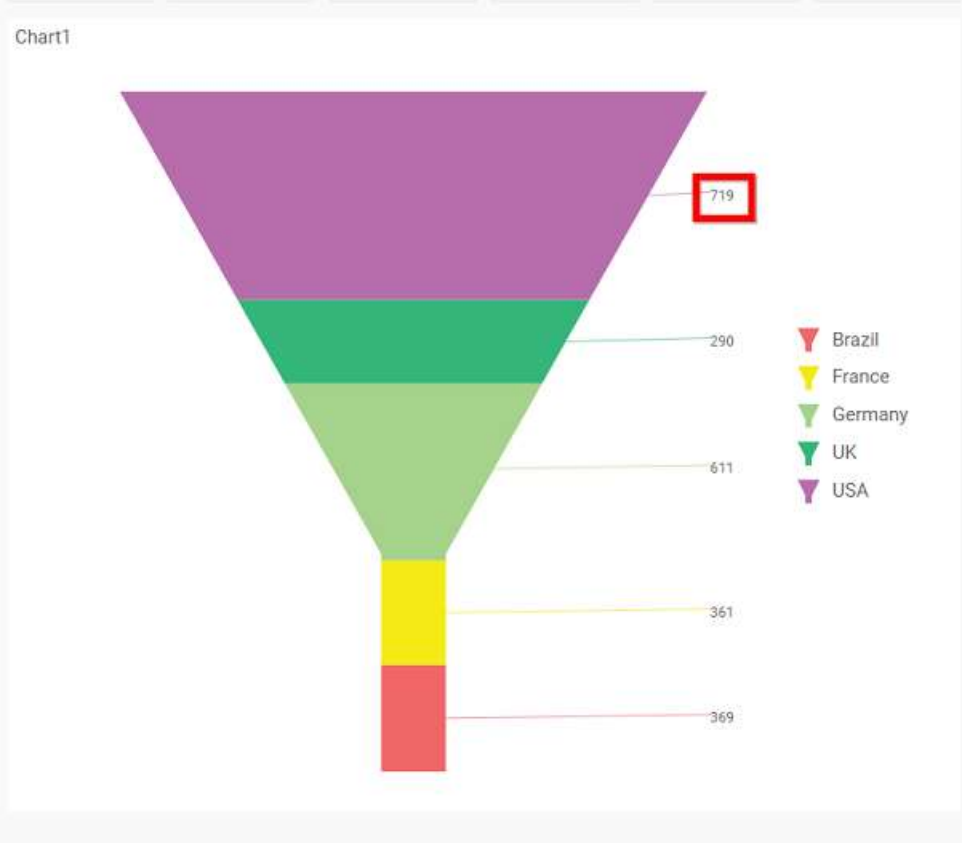
This allows you to toggle the visibility of value labels.



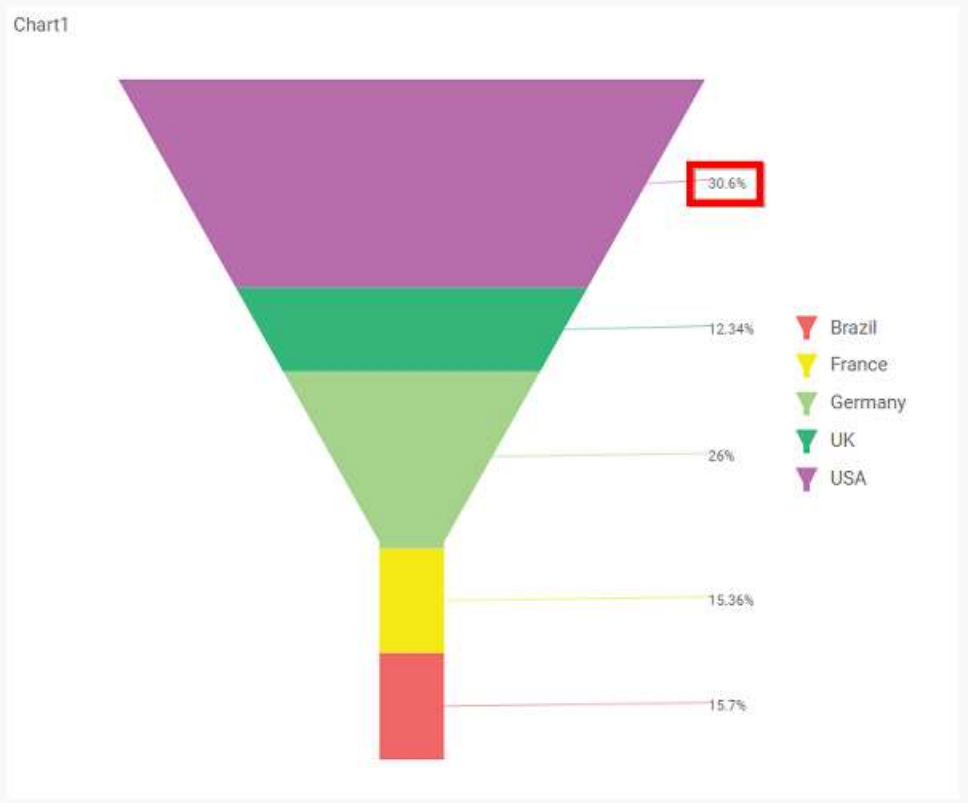
Data Label

This allows you to define the display format either as value or as percentage or both.

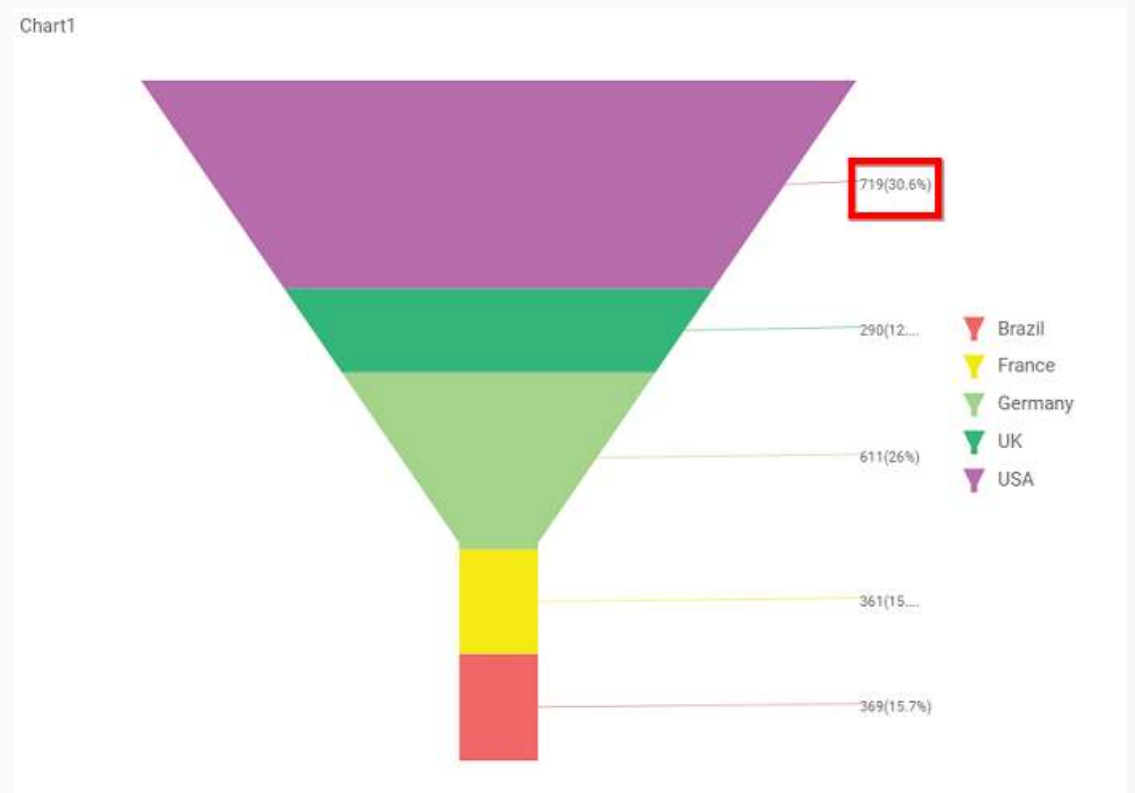
Value



Percentage

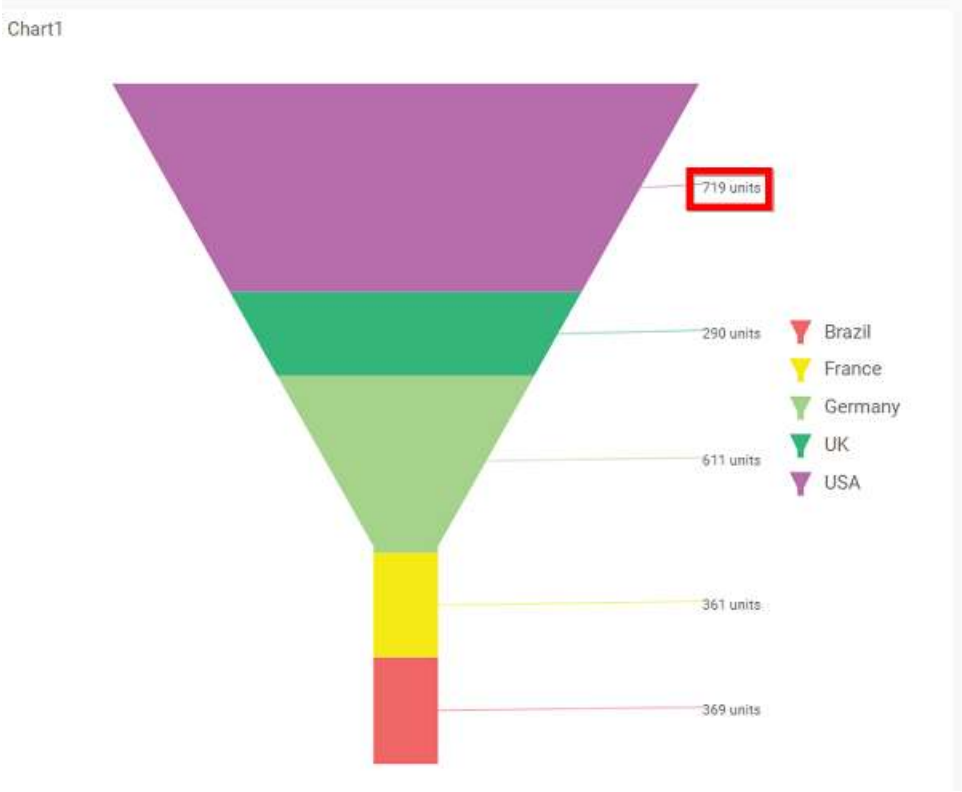


Value and Percentage

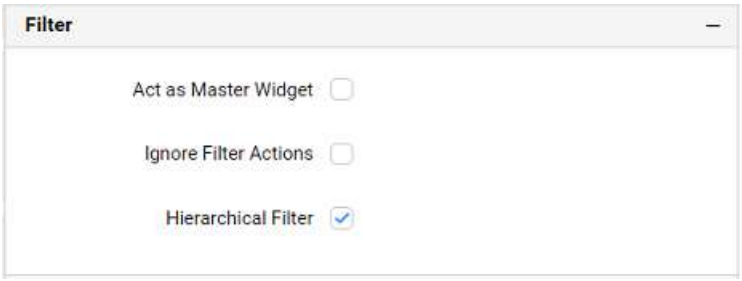


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter



Act as Master Widget

This allows you to define this funnel chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

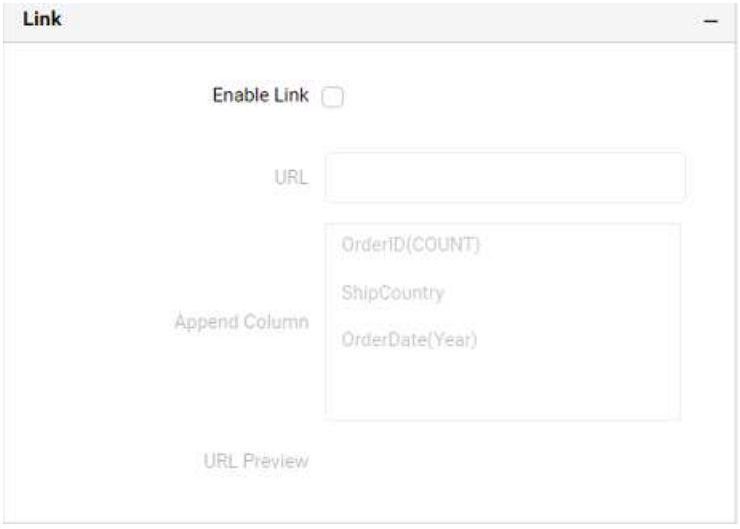
This allows you to define this funnel chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

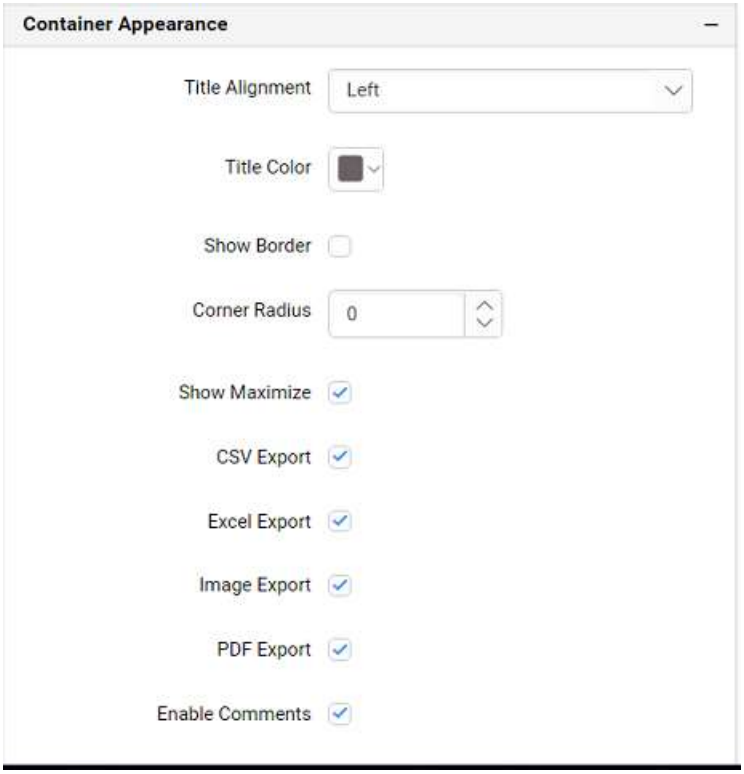
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if **Show Border** property is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this funnel chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this funnel chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this funnel chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

Image Export

This allows you to enable/disable the image export option for this funnel chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

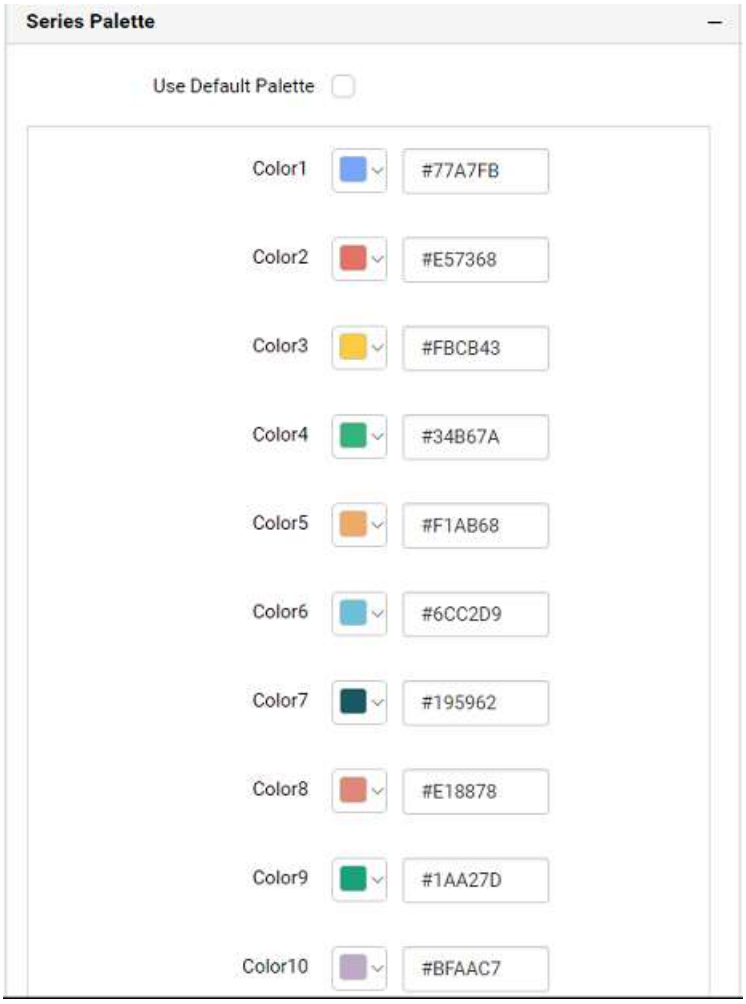
This allows you to enable/disable comment for dashboard widget. For more details refer here

Series Palette

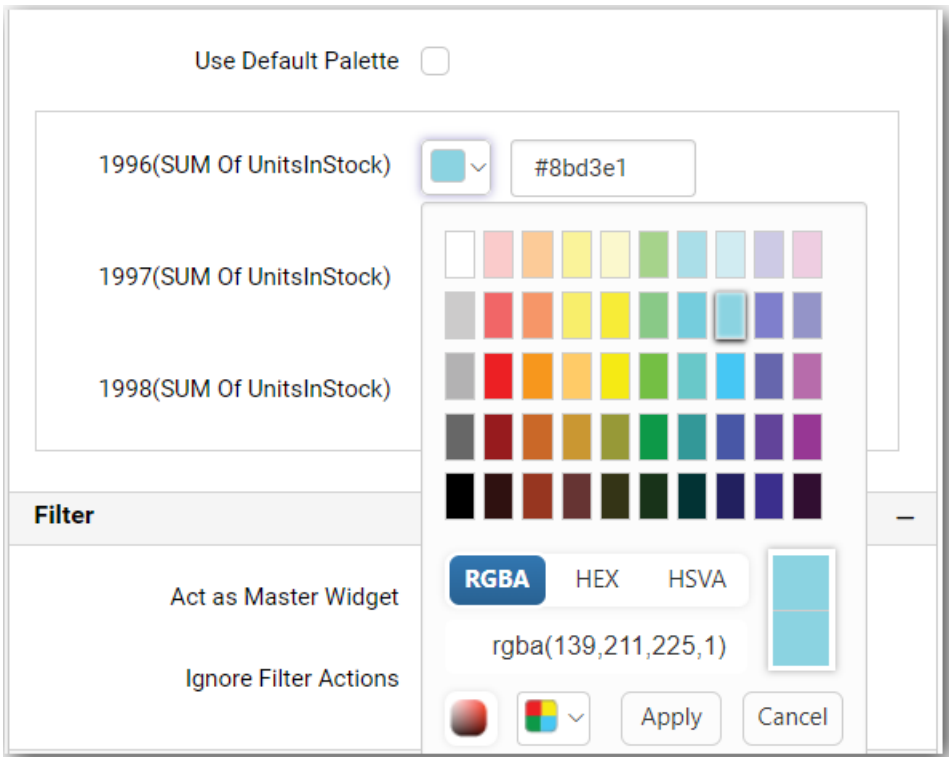
This allows you to customize the chart series color through Series Palette section.

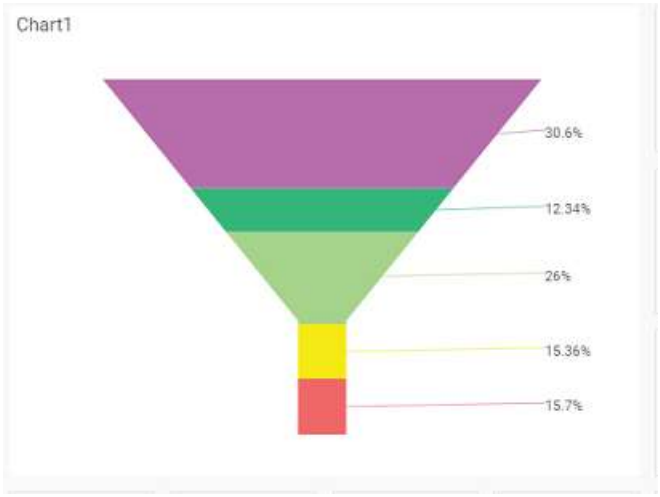
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to proportion series segments.



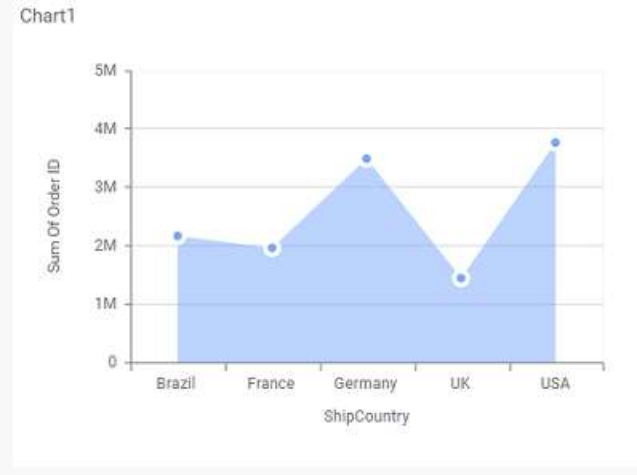
By toggle off the Use Default Palette, you can customize the proportion series segments colors. This section shows a palette of colors. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Area Chart

Area Chart allows you to compare values for a set of unordered items across categories through filled curves ordered vertically.

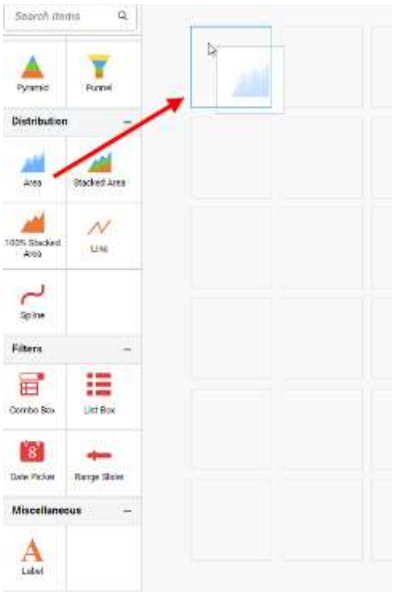


How to configure the table data to area chart?

Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

Following chart illustrates about how to configure data to area chart

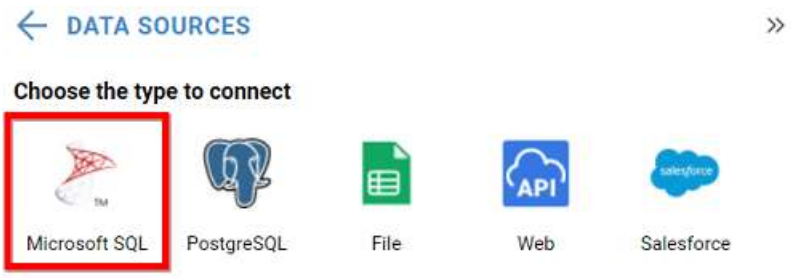
Drag and drop the area chart widget into canvas and resize into your required size.



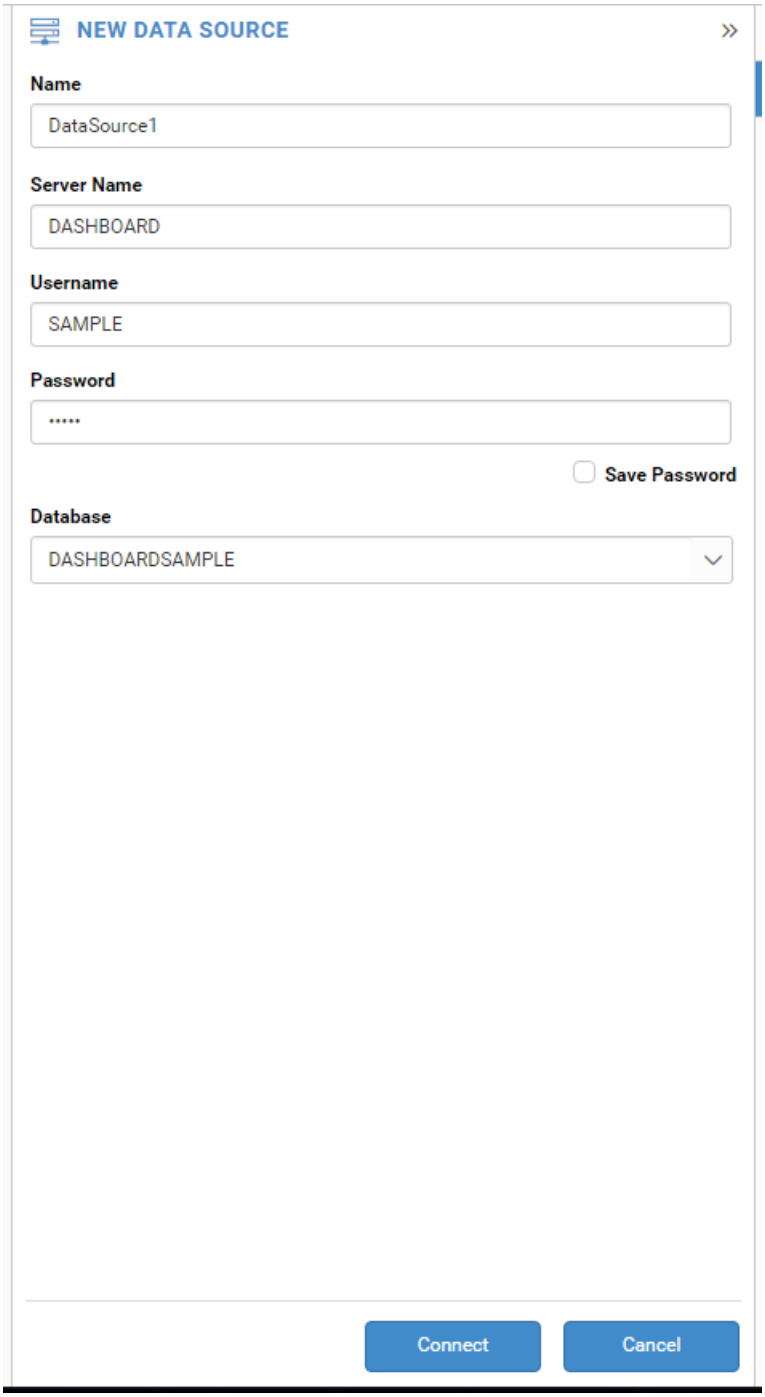
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

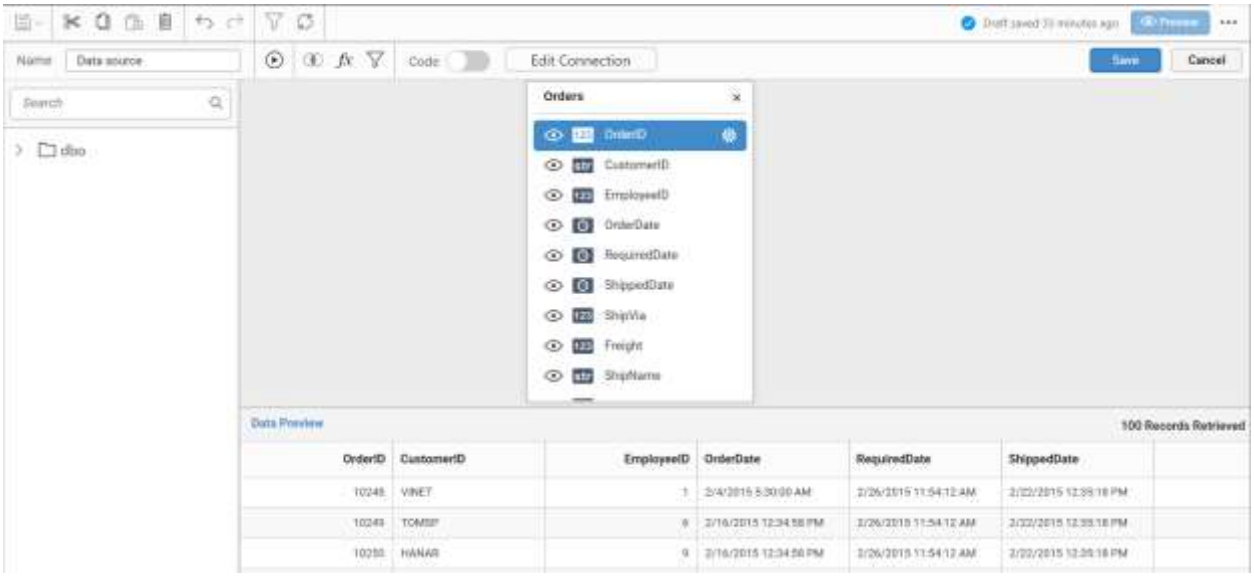
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



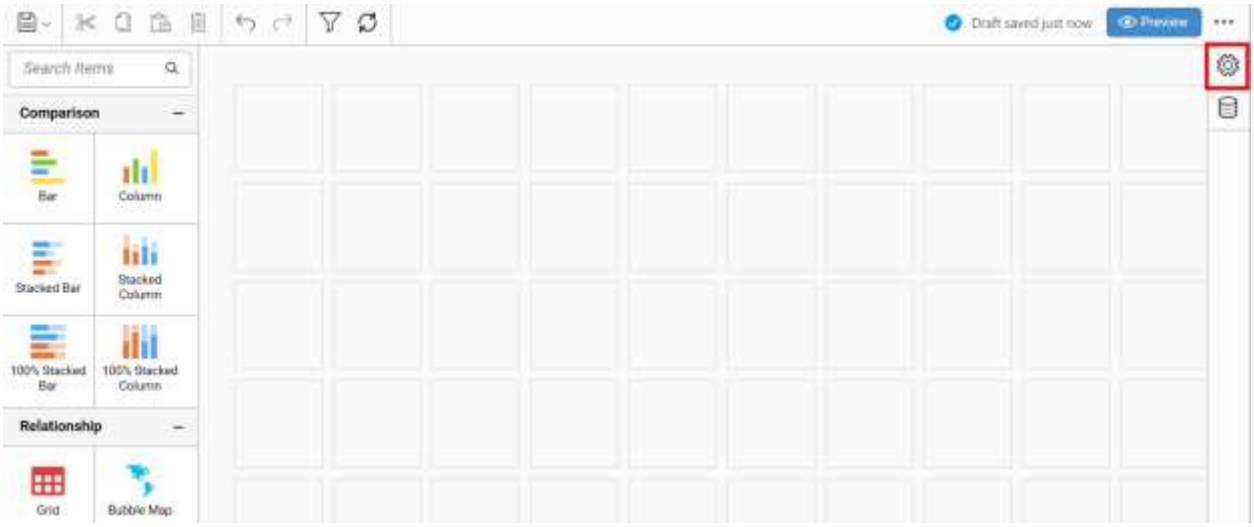
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

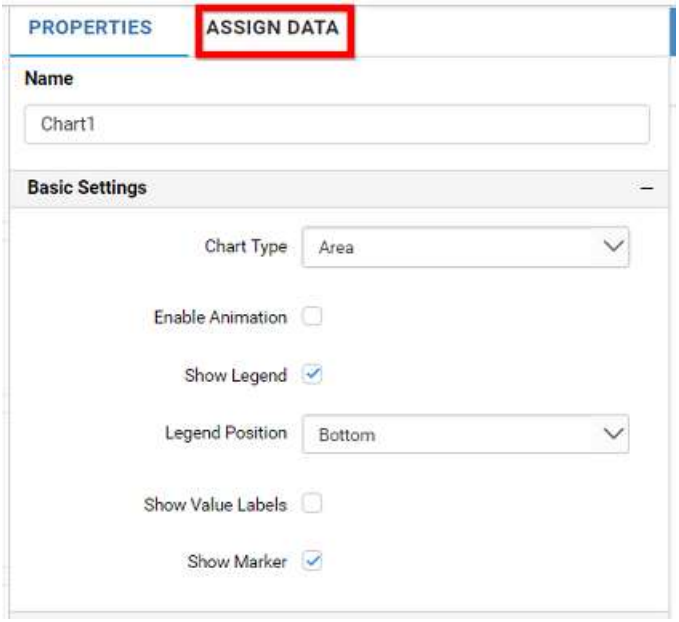


Drag your preferred table or view from the left pane from data design view, click Save button.

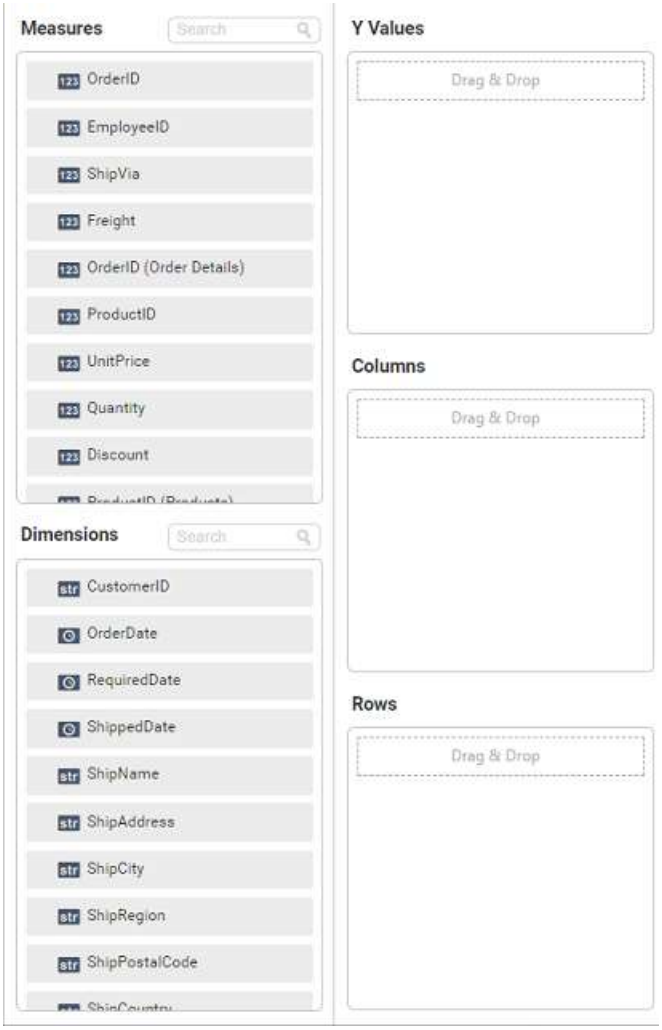


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.





The data tab will be opened with available measures and dimensions from the connected data source

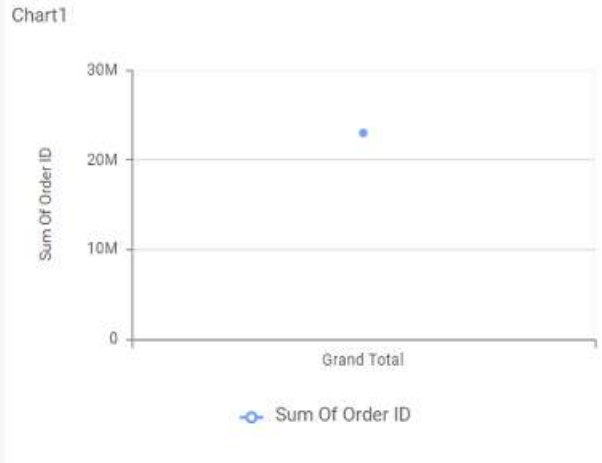


You can add the required data from Measures and Dimensions into required field.

Adding Y Values

You can add more than one Measures into Y Values field by drag and drop the required measure.

Now the area chart will be rendered like this

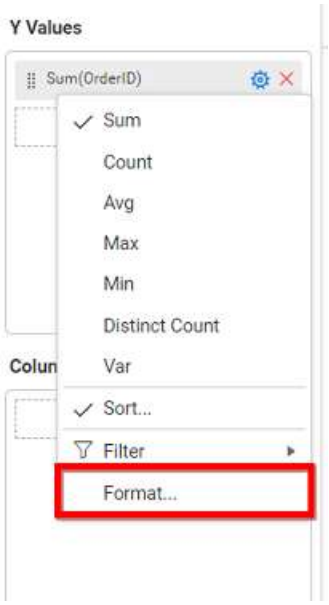


Click the Settings option to change required summary type from the available summary types shown in Settings.

You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

You can filter the data to be displayed in chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

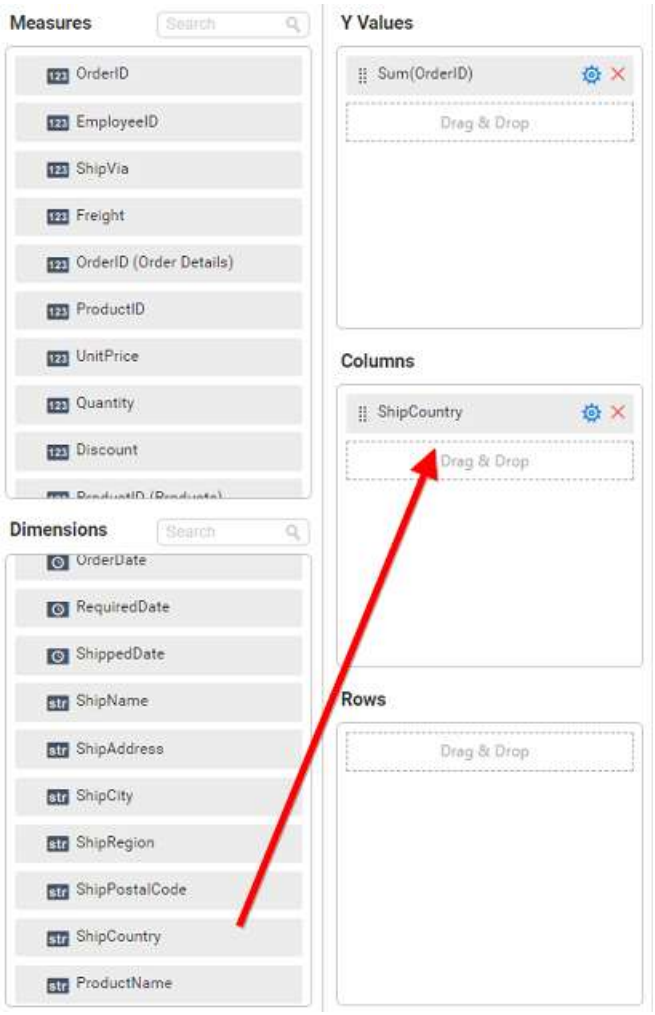


To remove the added value fields click highlighted button.

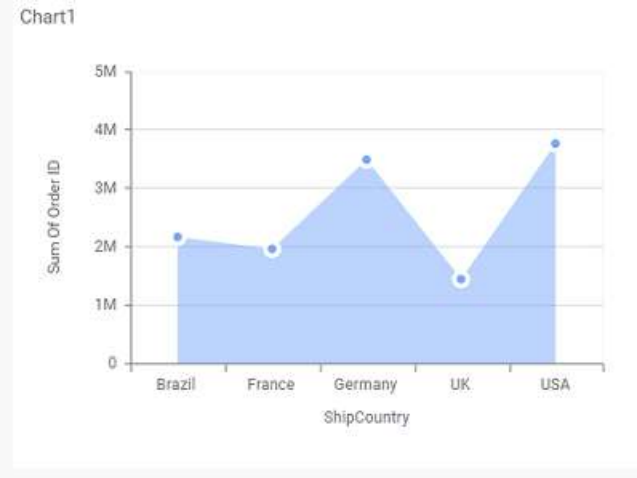
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



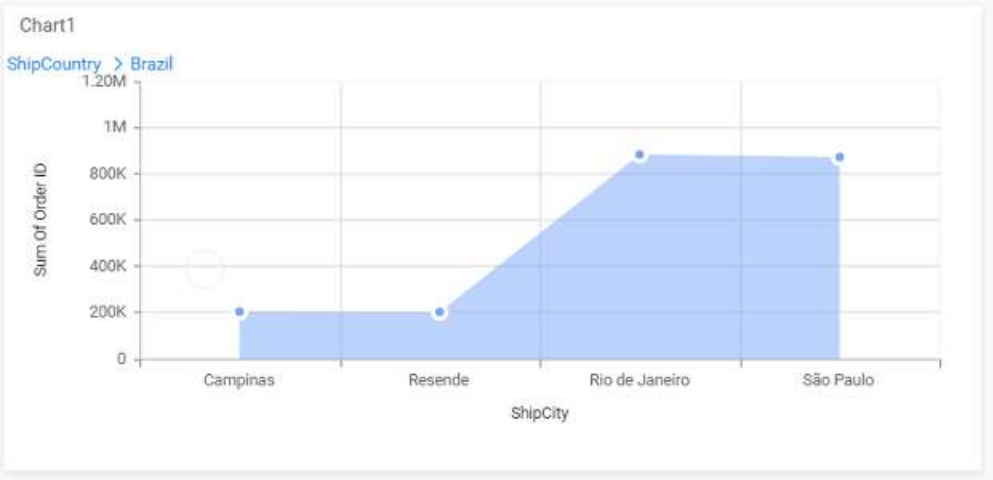
Area chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

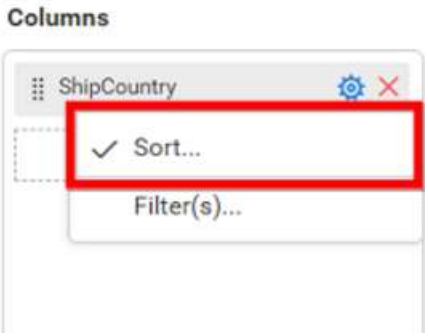
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.

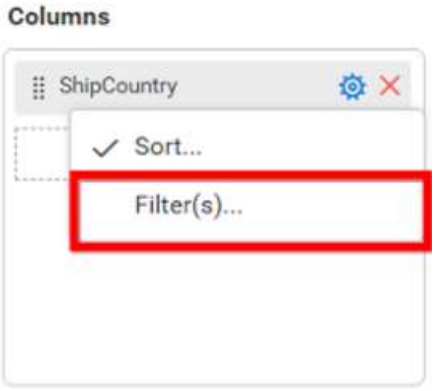


You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

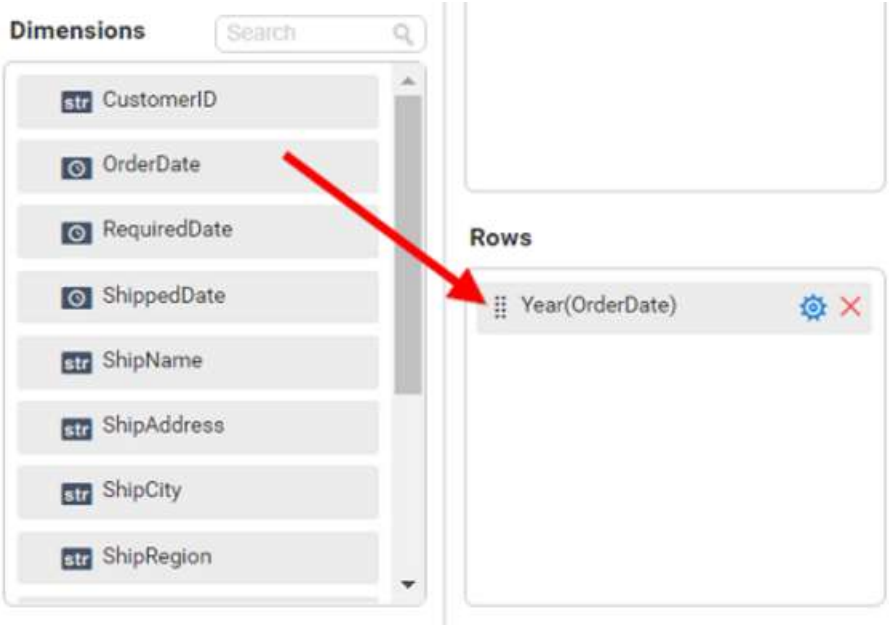


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

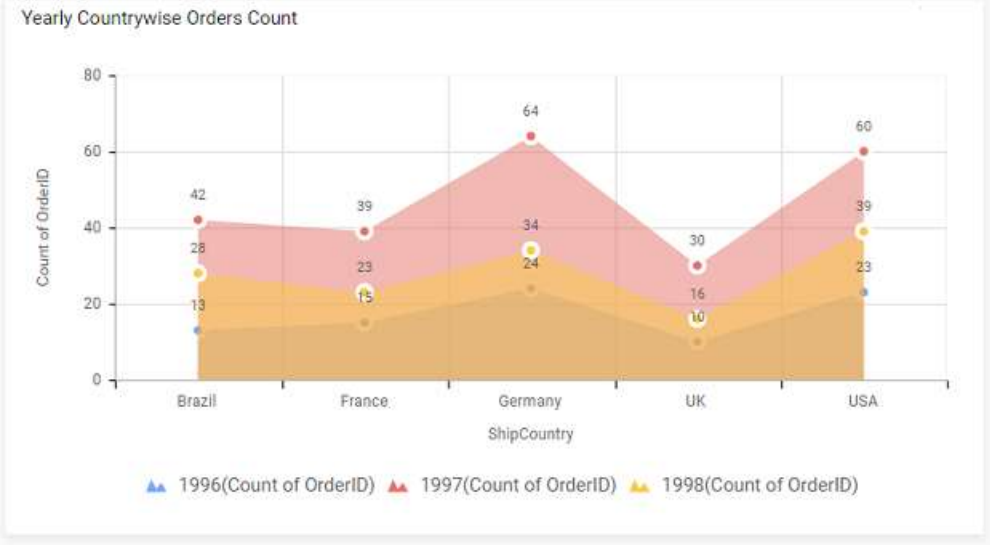
Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render area chart in series.



How to format area chart?

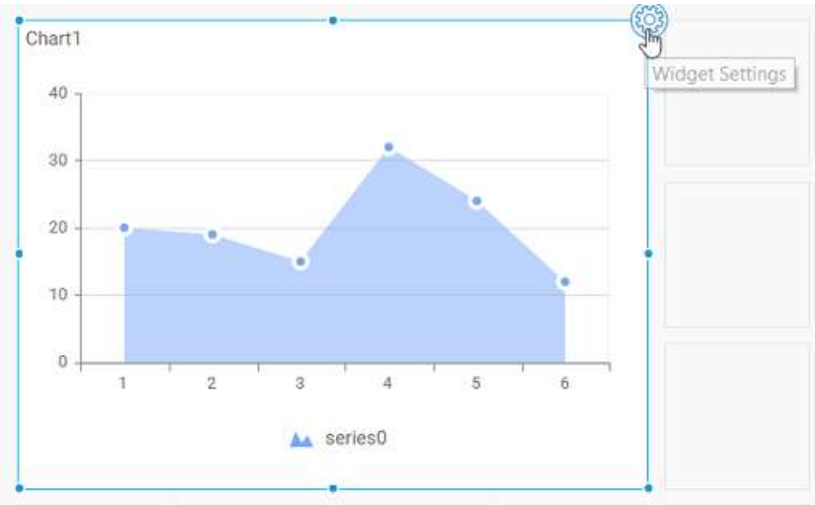
You can format the area chart for better illustration of the view that you require, through the settings available in Properties tab.

To format area chart follow the steps

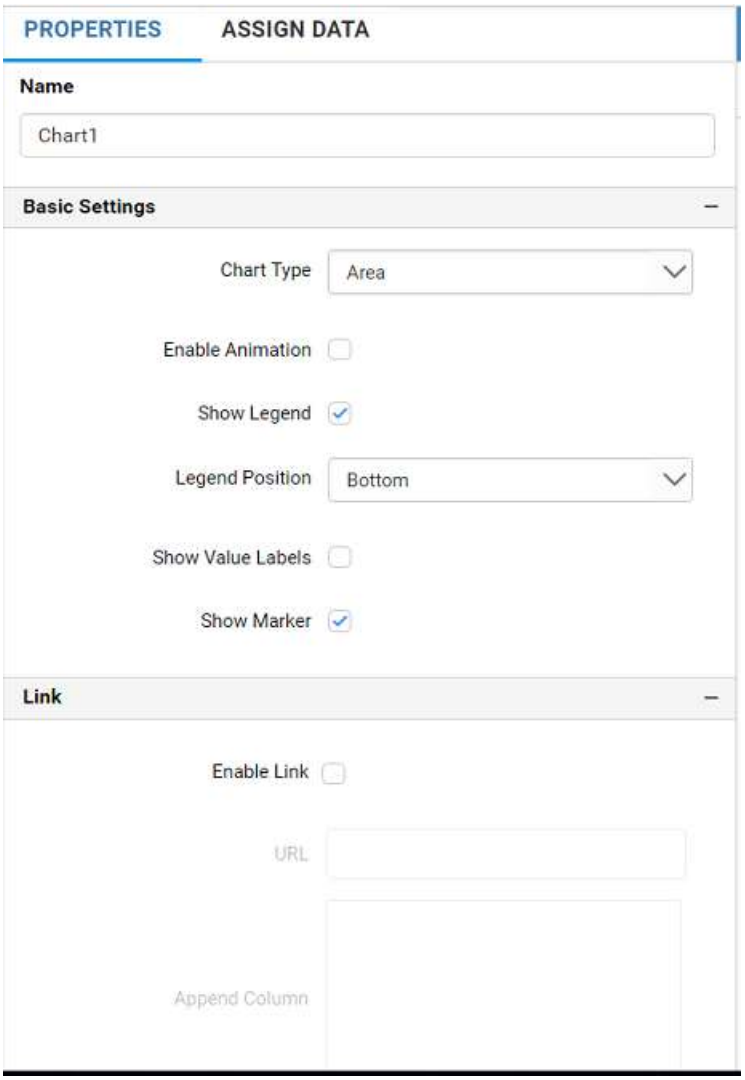
Drag and drop the area chart into canvas and resize it to your required size.

Configure the data into area chart.

Focus on the area chart and click on widget settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this area chart widget.

Basic Settings

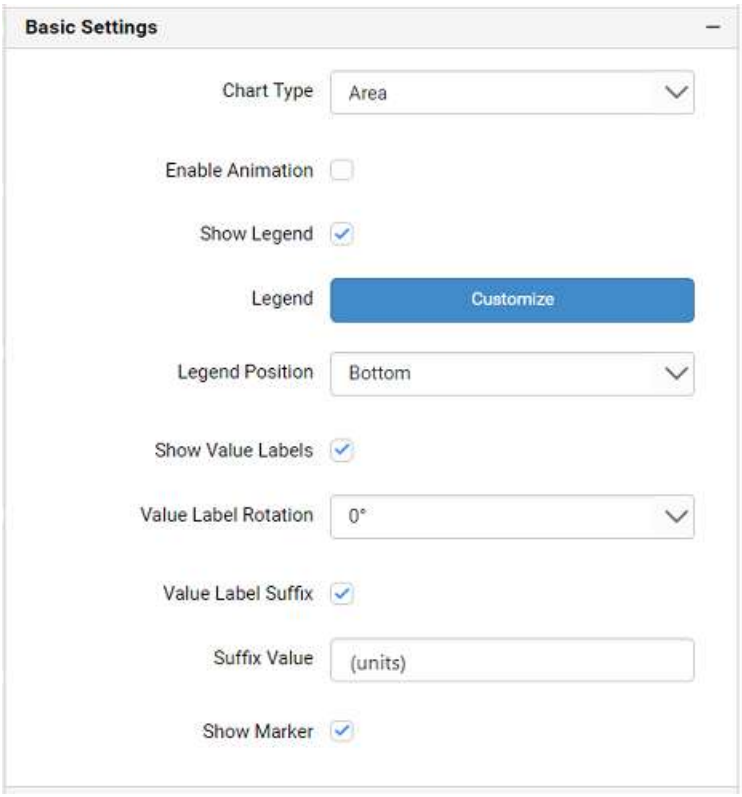


Chart Type

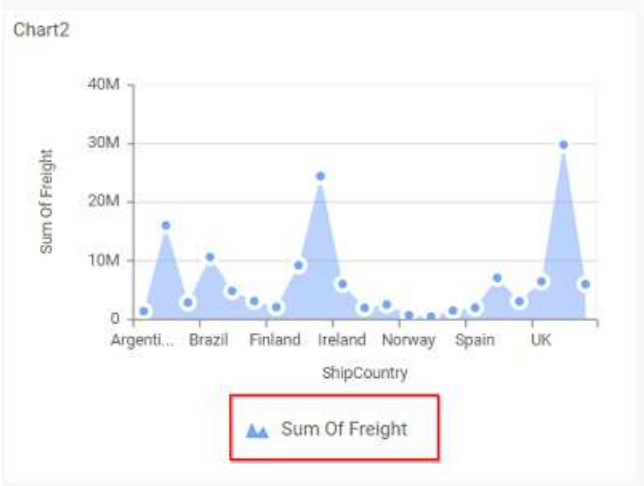
This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

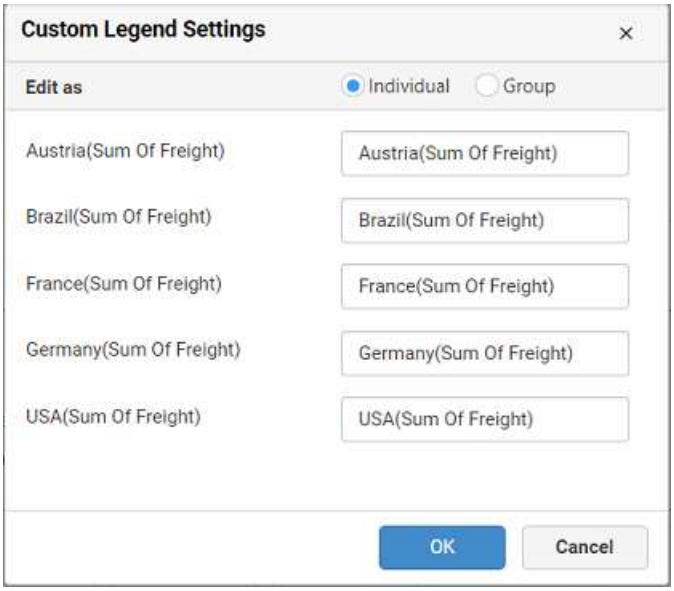
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

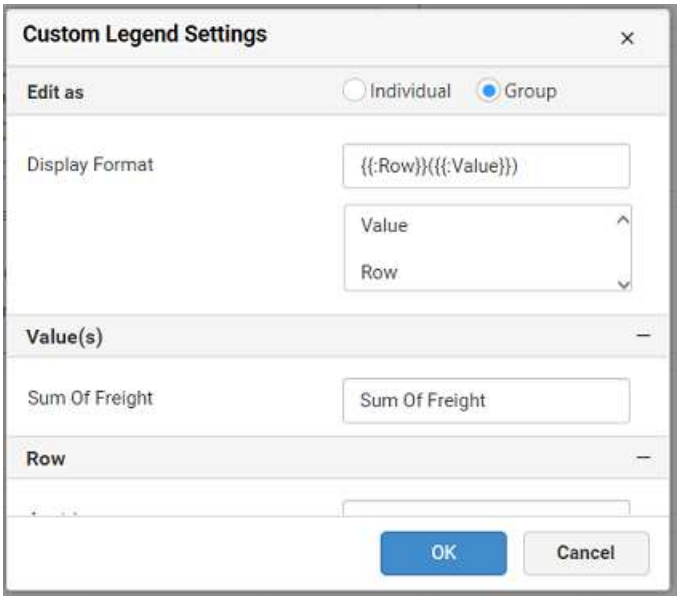
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.



Group

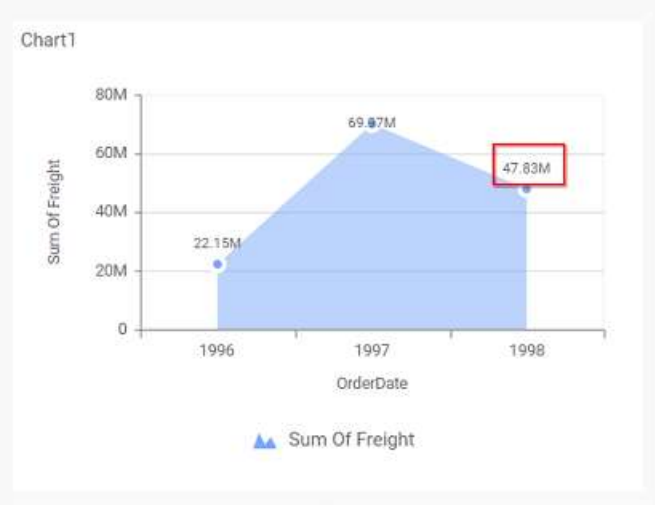
Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Austria (Sum of Freight)

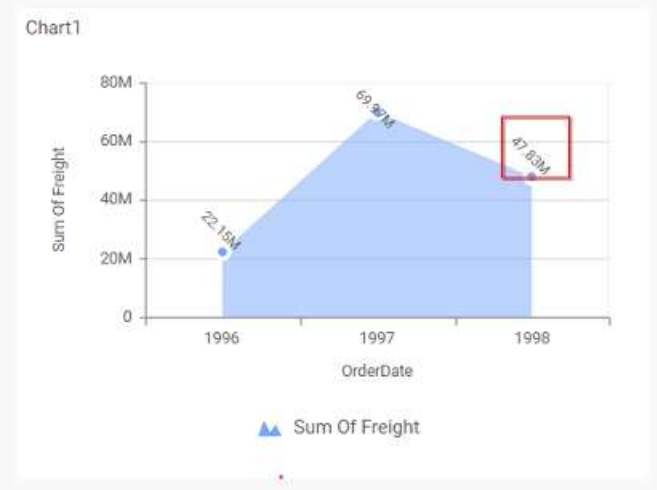
Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

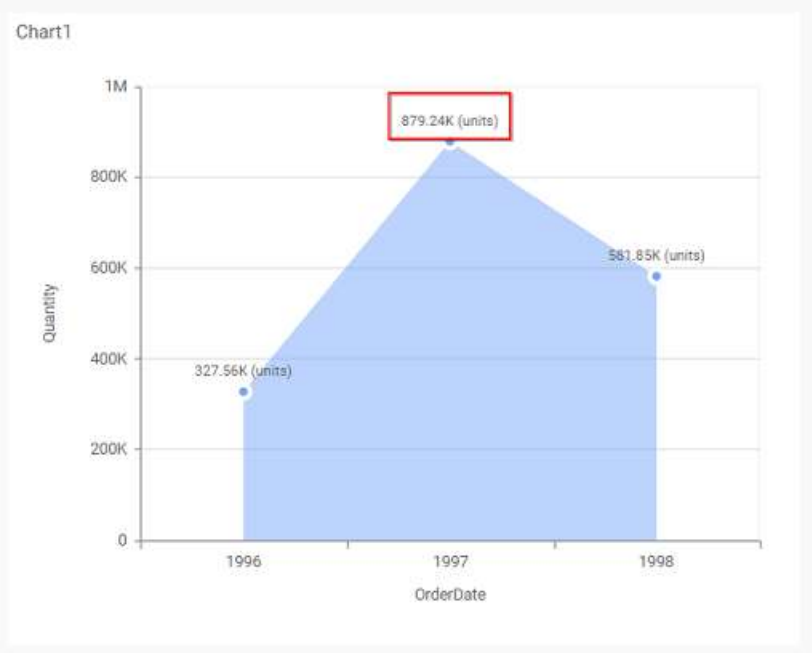


Value Label Suffix

Allows you to enable the Suffix value text to the value labels.

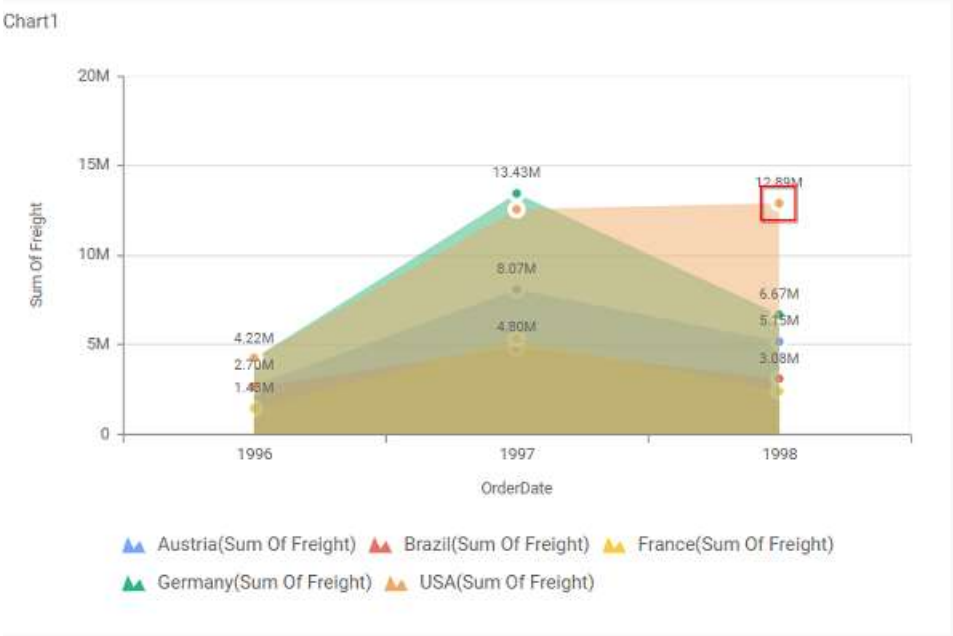
Suffix Value

Allows you to set\edit suffix value to the value labels.

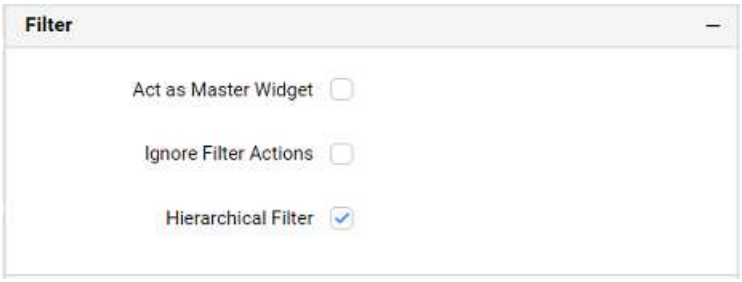


Show Marker

This allows you to toggle the visibility of marker from label to adorn each data point in chart series.



Filter



Act as Master Widget

This allows you to define this area chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

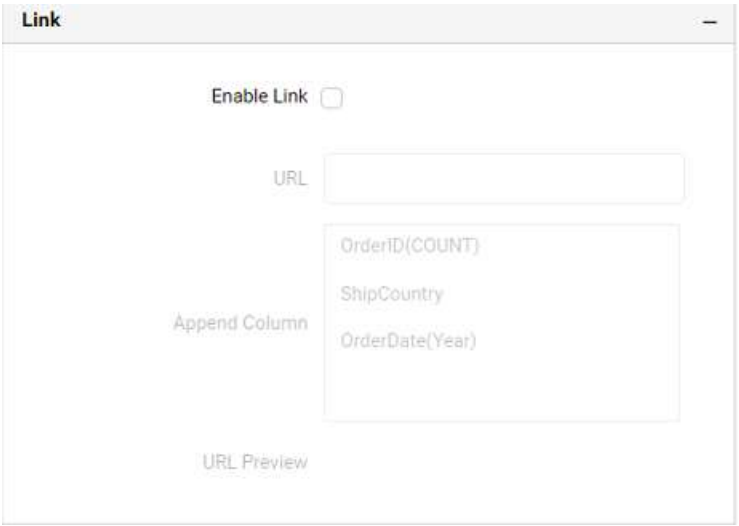
This allows you to define this area chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

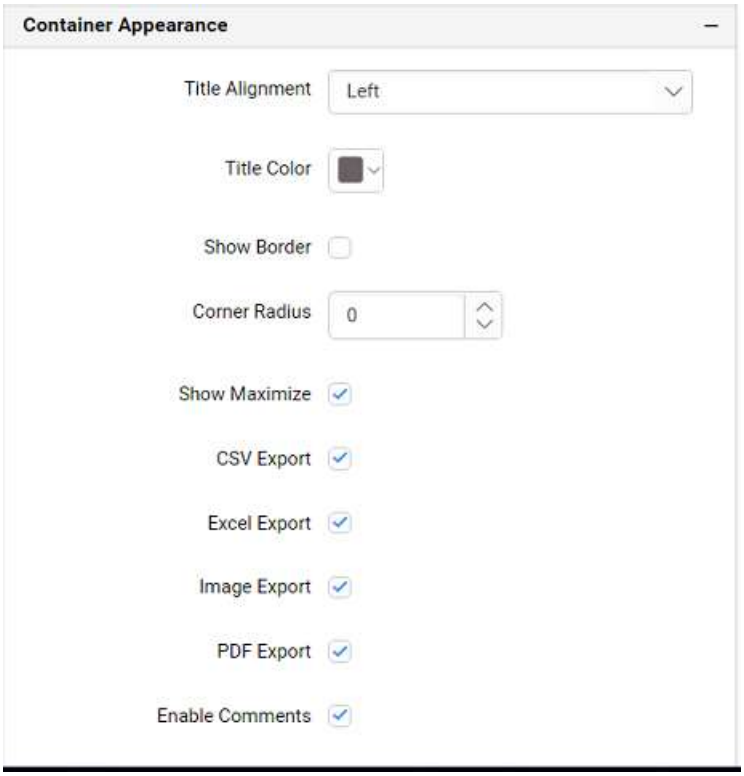
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Maximized View

This allows you to enable/disable the maximized mode of this area chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this area chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this area chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

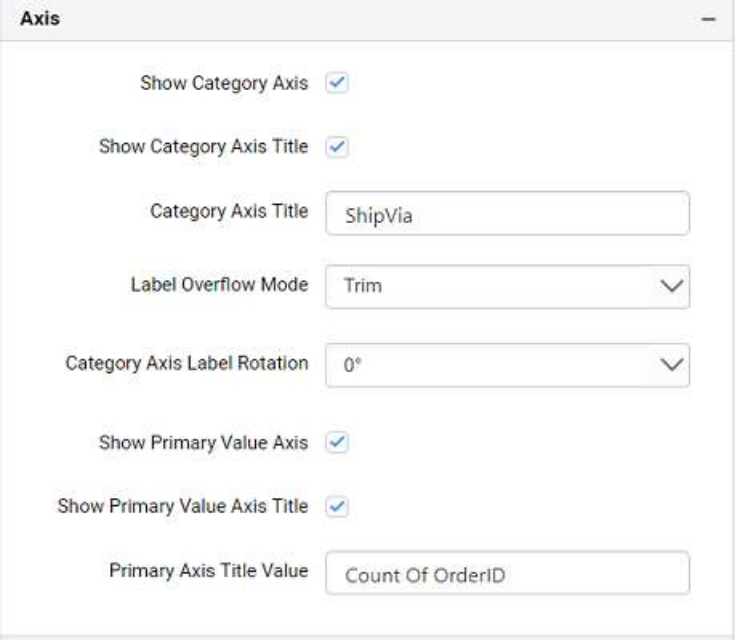
Image Export

This allows you to enable/disable the image export option for this area chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

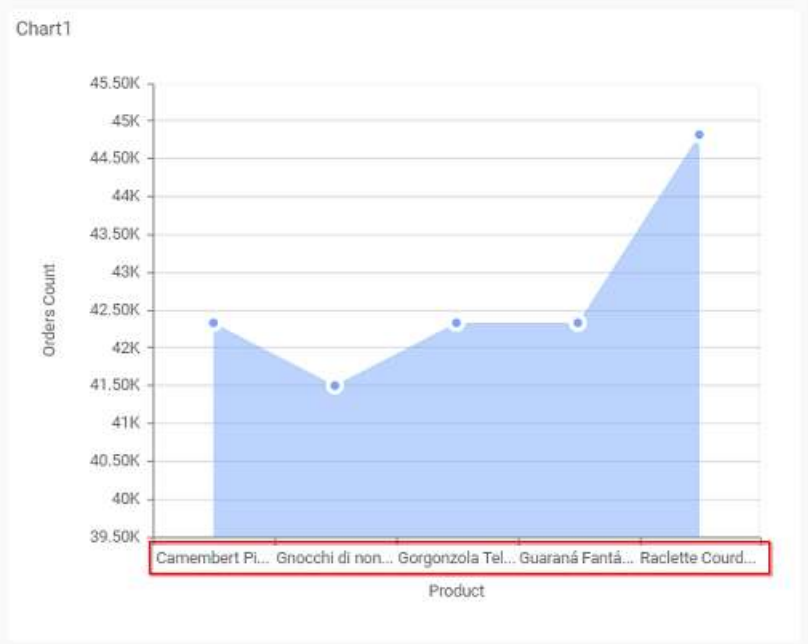
Axis



This section allows you to customize the axis settings in chart.

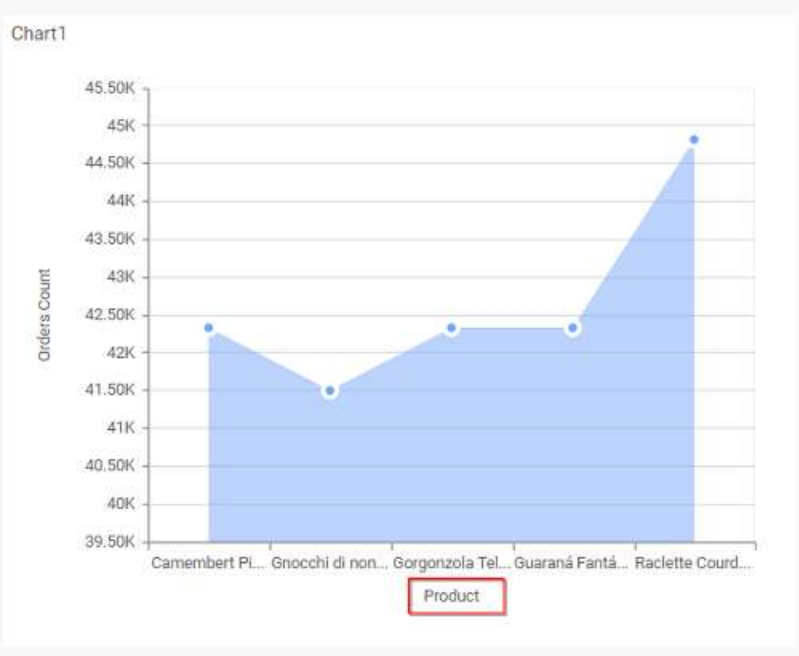
Show Category Axis

This allows to enable the visibility of Category Axis.



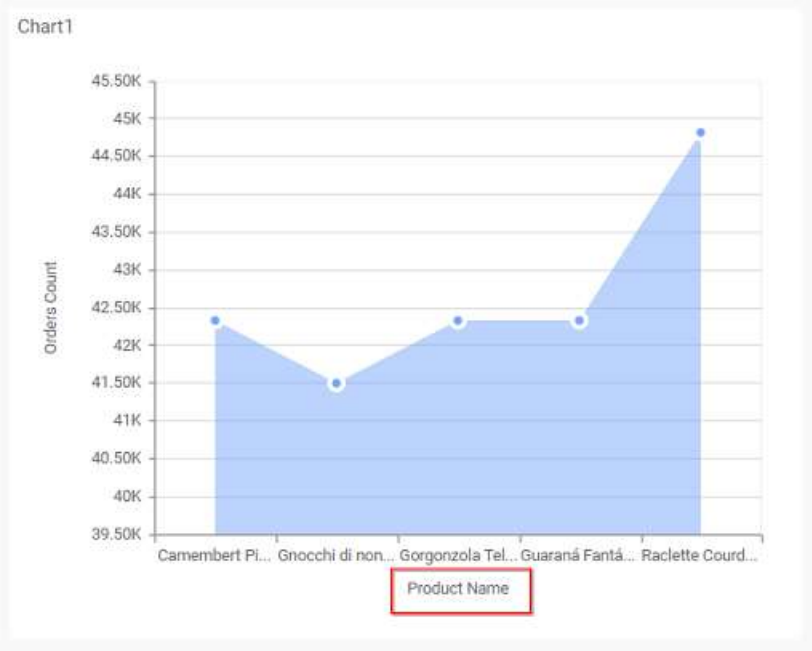
Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

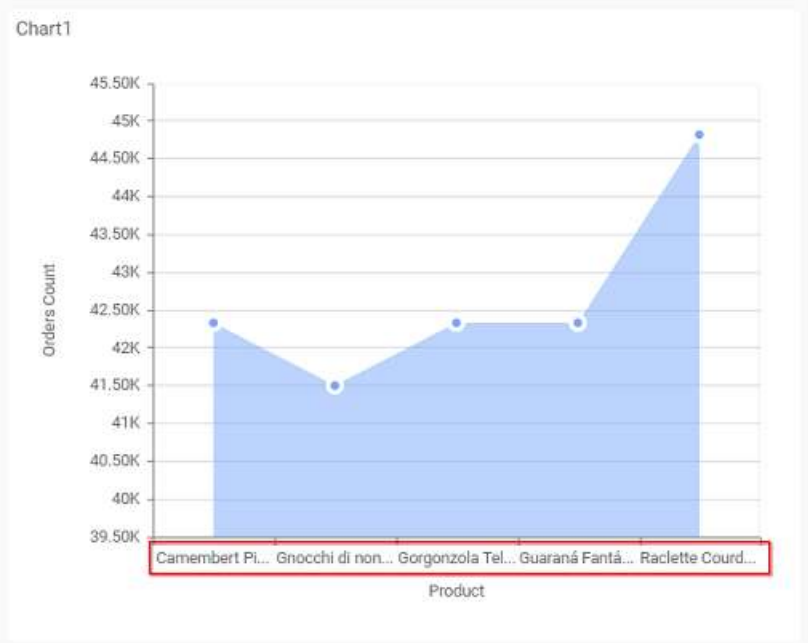


Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

Trim

This option trims the end of overlapping label in the axis.

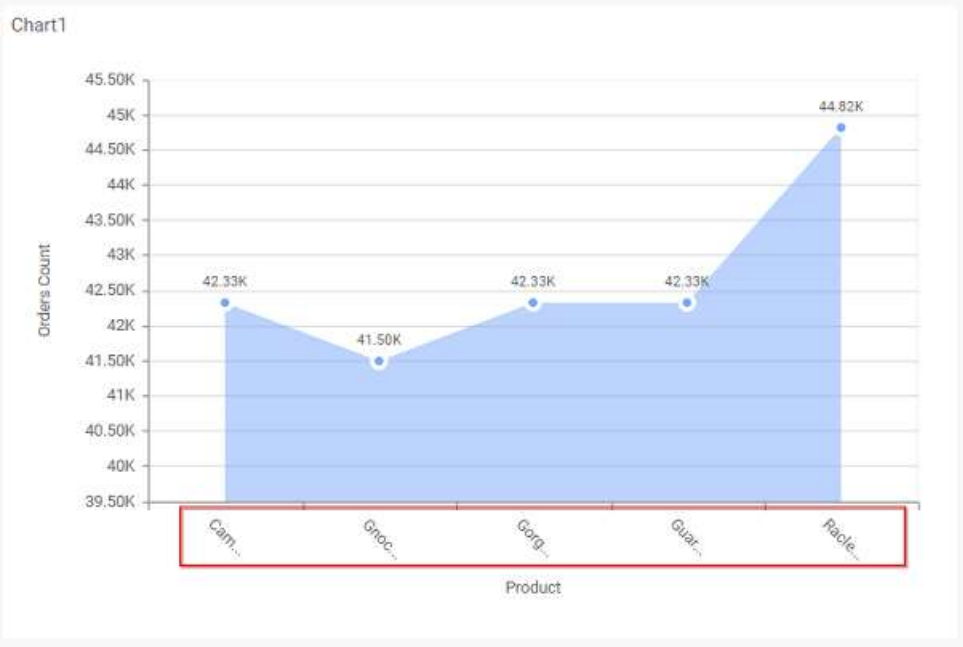


Hide

This option hides the overlapping label in the axis.

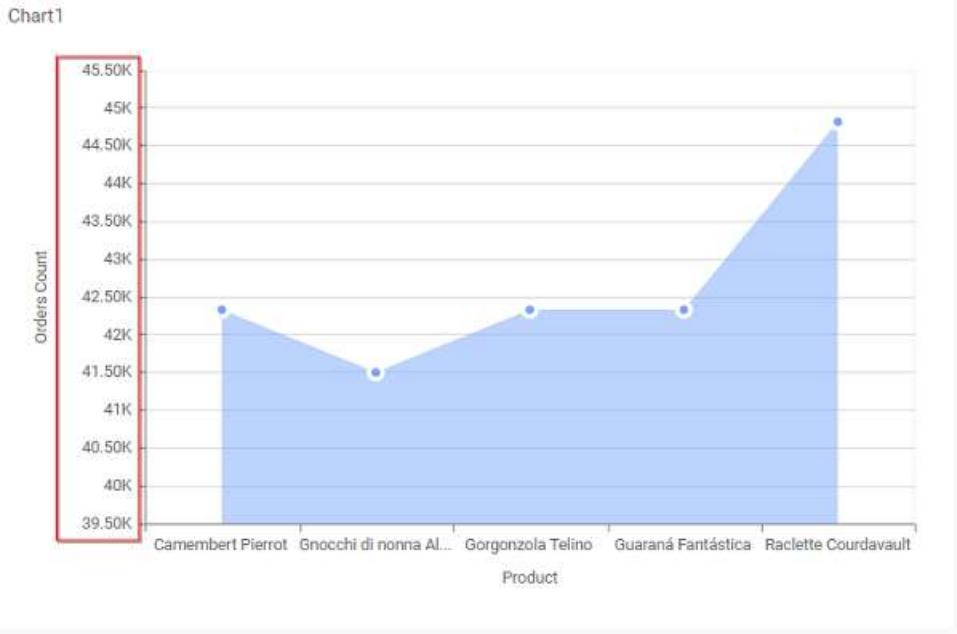
Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



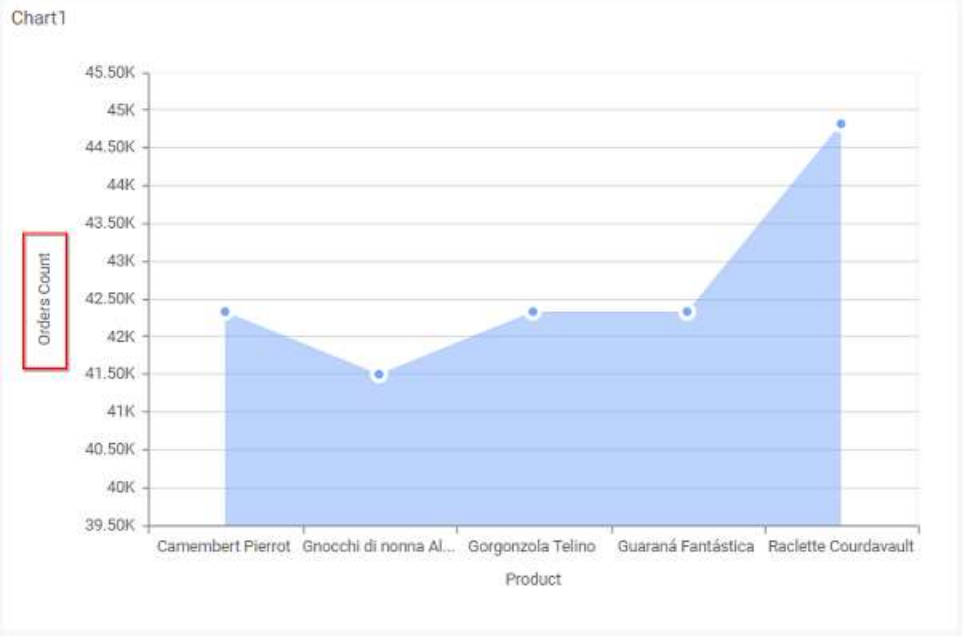
Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.



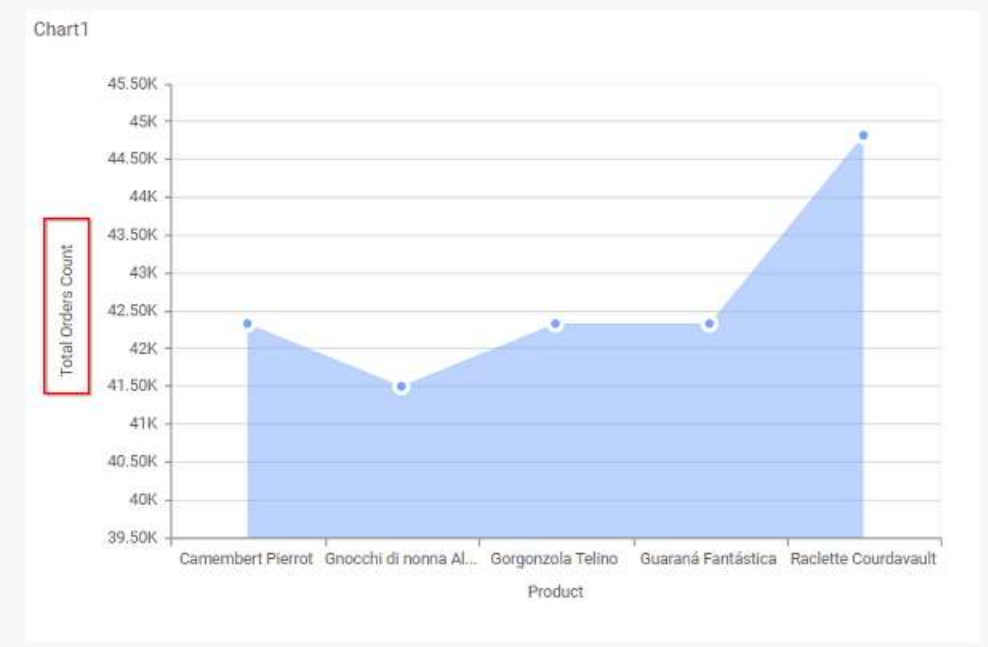
Show Primary Value Axis Title

This allows you to enable the visibility of Primary Value Axis title of chart.

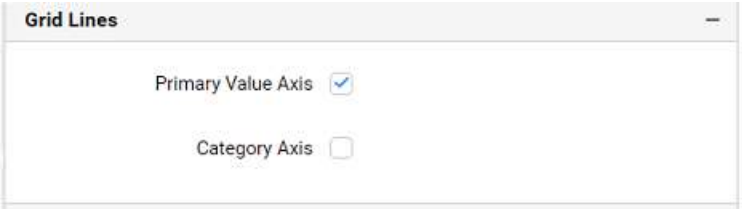


Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

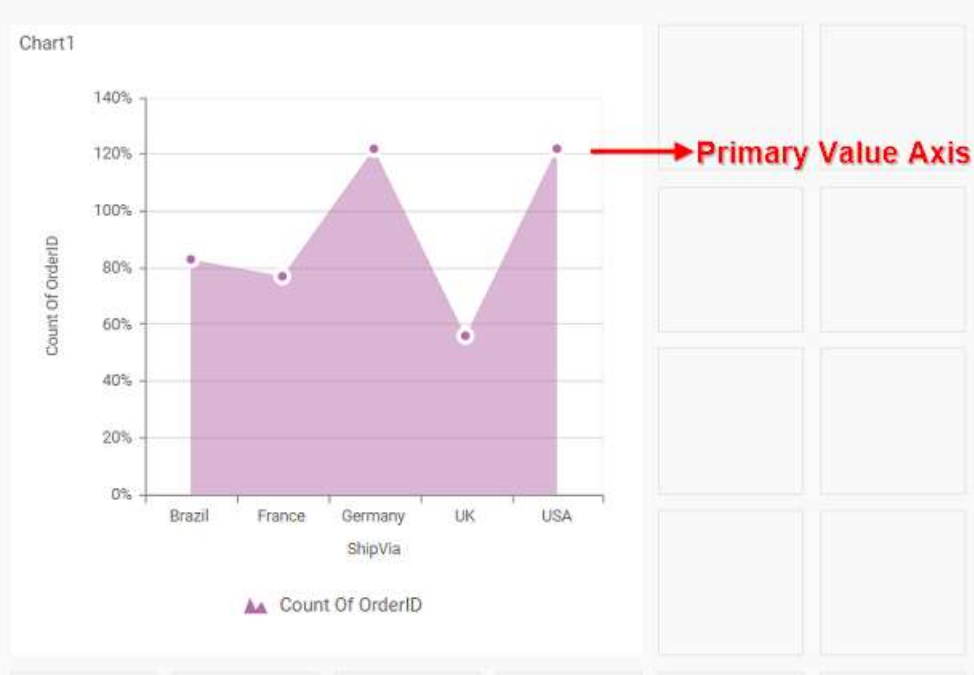


Grid Line



Primary Value Axis

This allows you to enable the Primary Value Axis gridlines for the area chart.



Category Axis

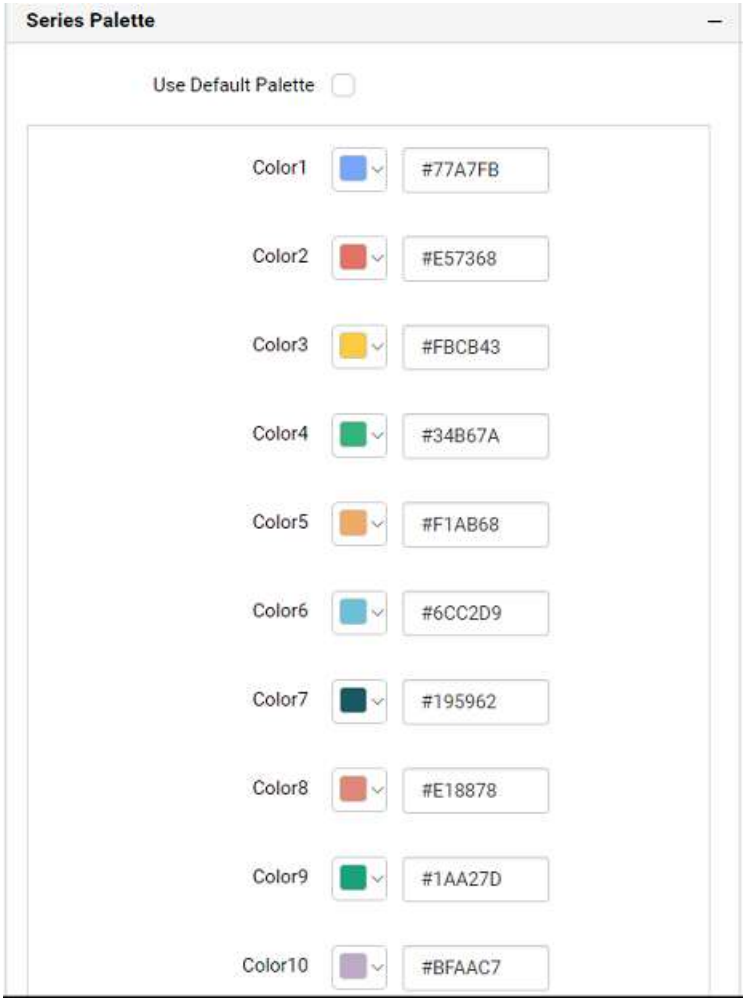
This allows you to enable the Category Axis gridlines for the area chart.

Series Palette

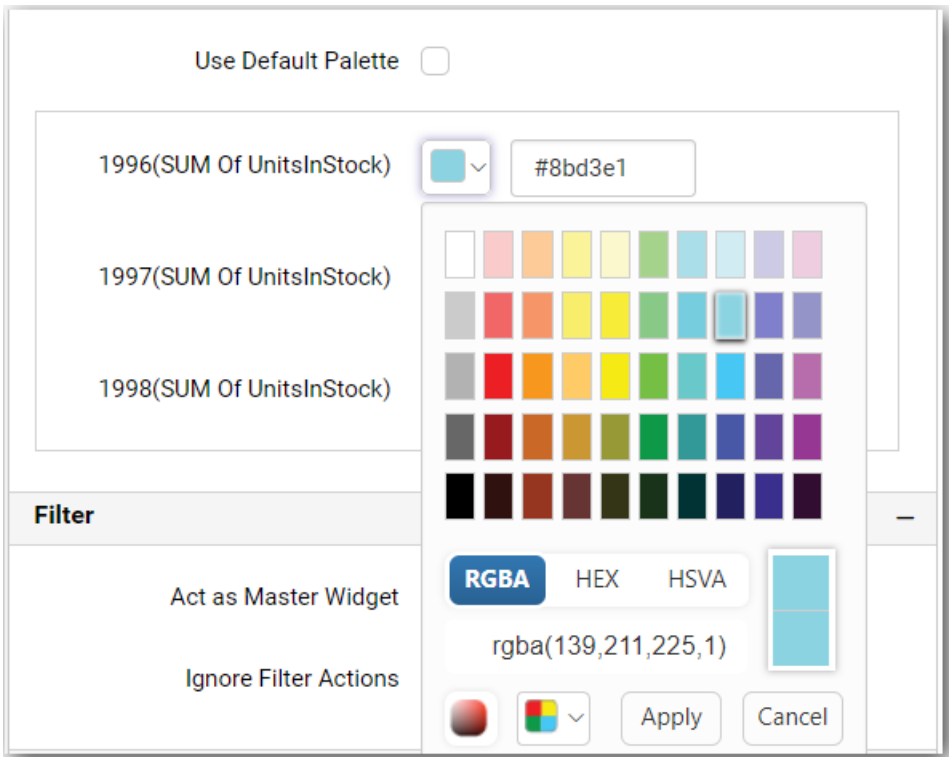
This allows you to customize the chart series color through Series Palette section.

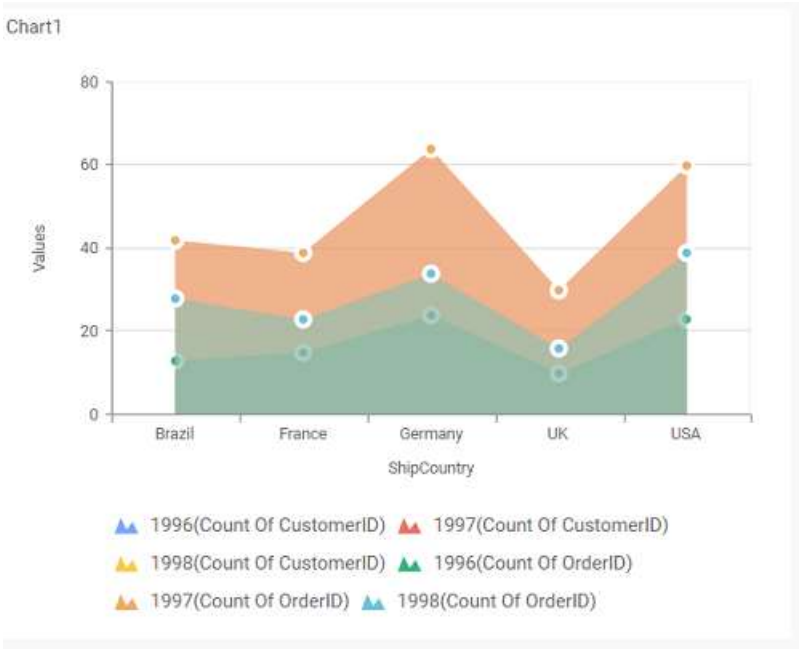
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



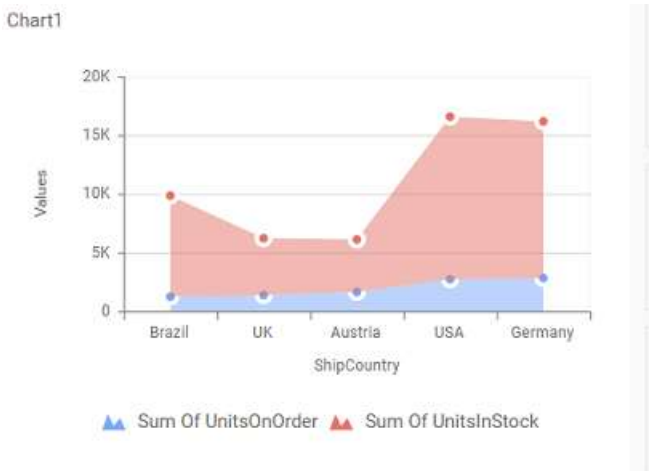
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Stacked Area Chart

Stacked Area Chart allows you to compare multiple measures through filled curves stacked one after the other vertically.

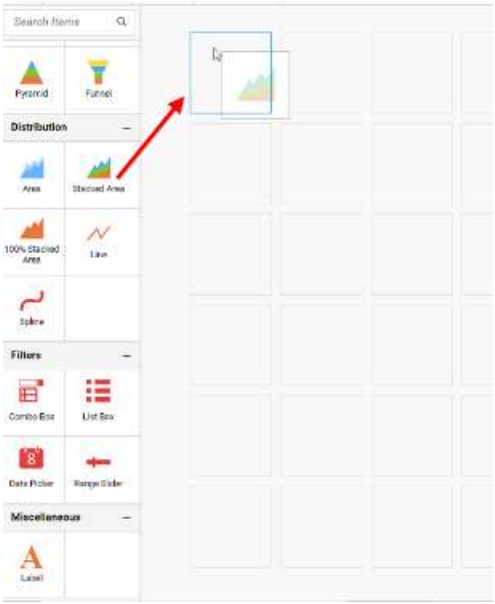


How to configure the table data to stacked area chart?

Stacked Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

Follow the steps to configure data to stacked area chart

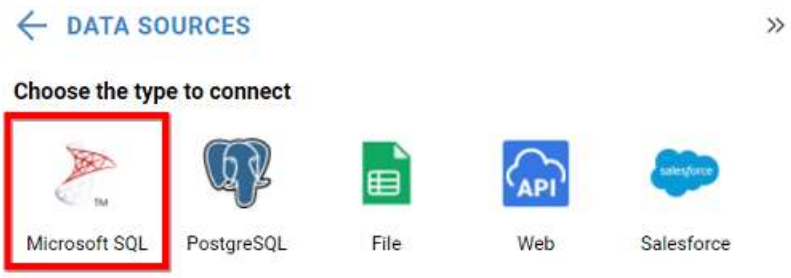
Drag and drop the stacked area chart into canvas and resize it to your required size.



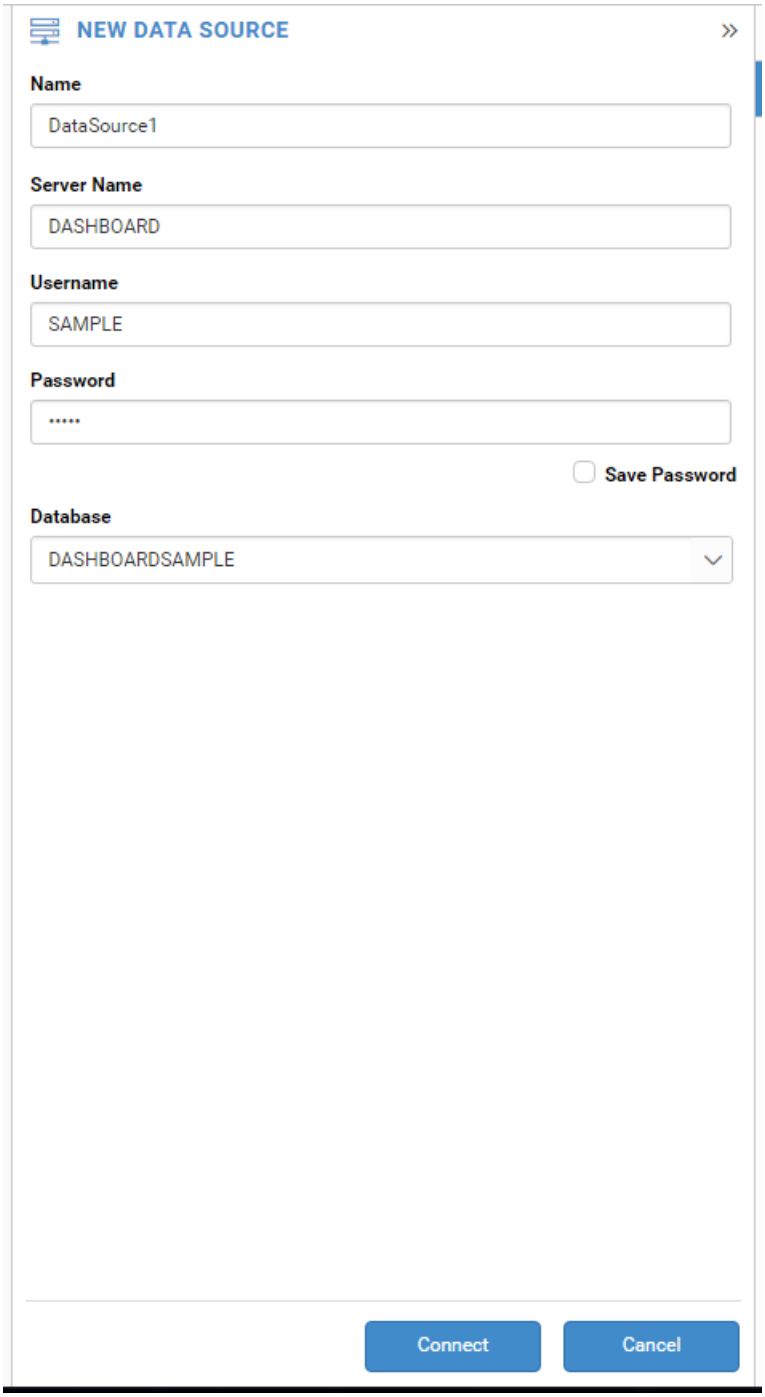
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

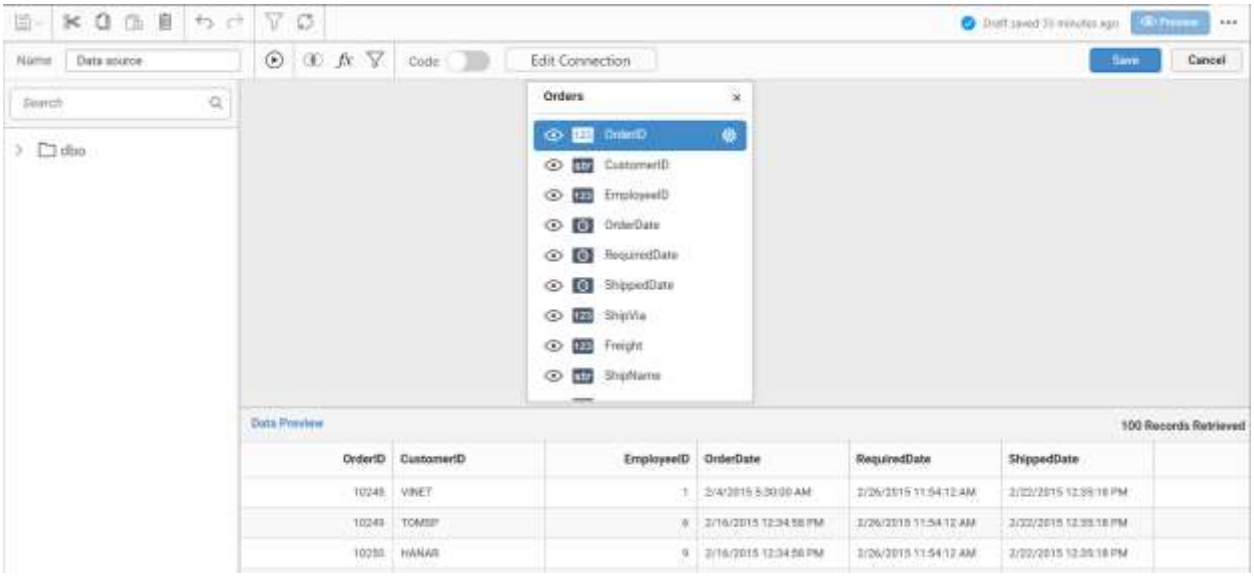
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



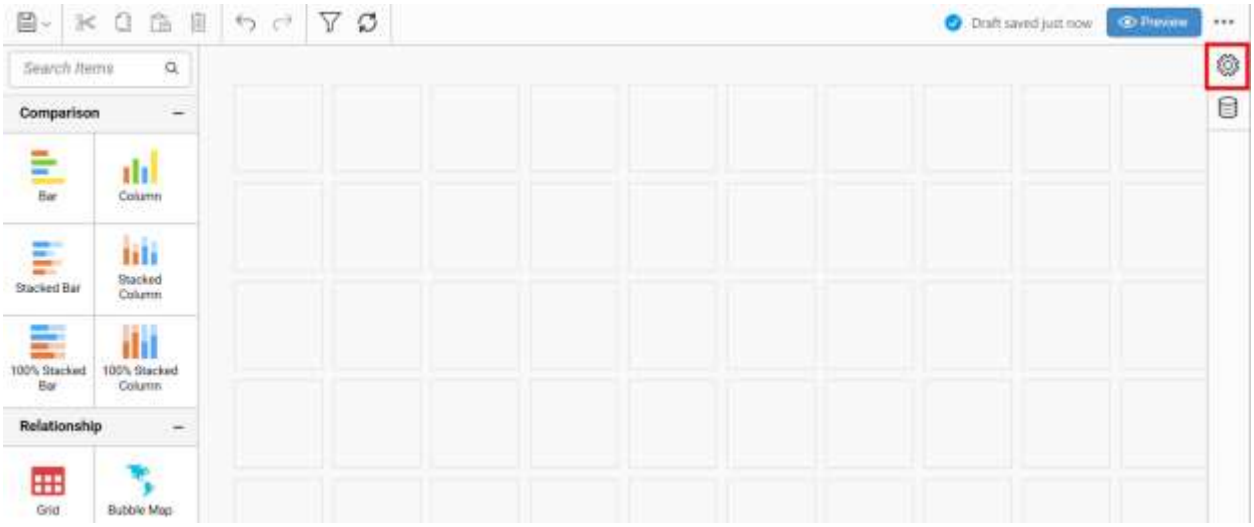
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

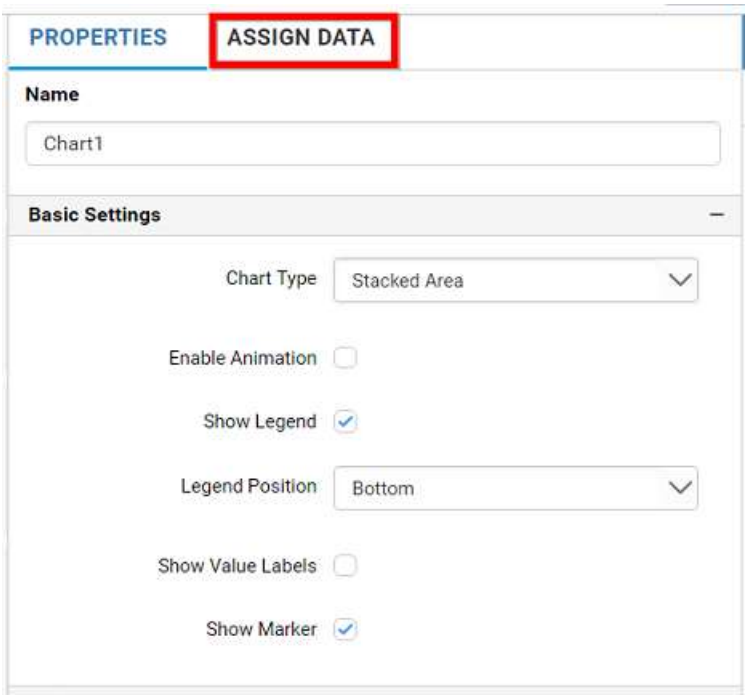


Drag your preferred table or view from the left pane from data design view, click Save button.

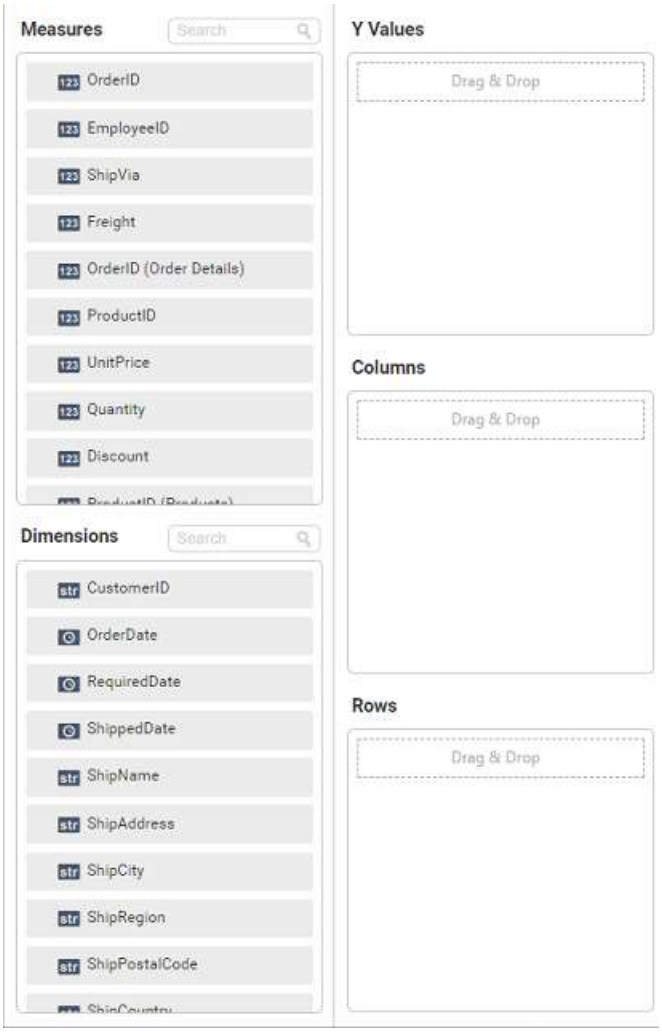


Click Properties button in configuration panel, property pane opens. Now, switch to ASSIGN DATA tab.





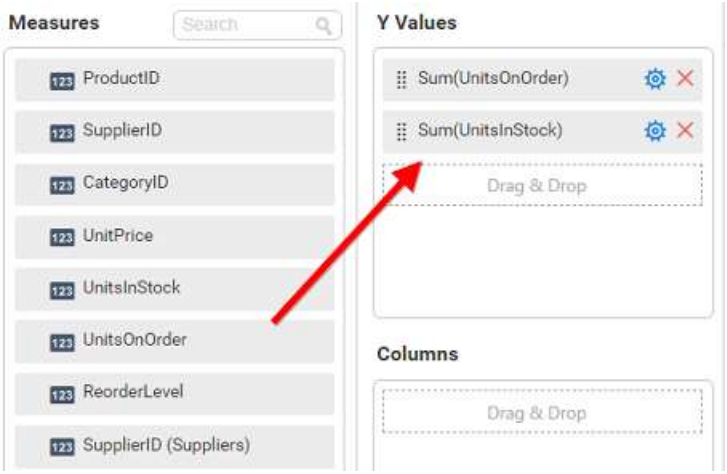
The data tab will be opened with available measures and dimensions from the connected data source



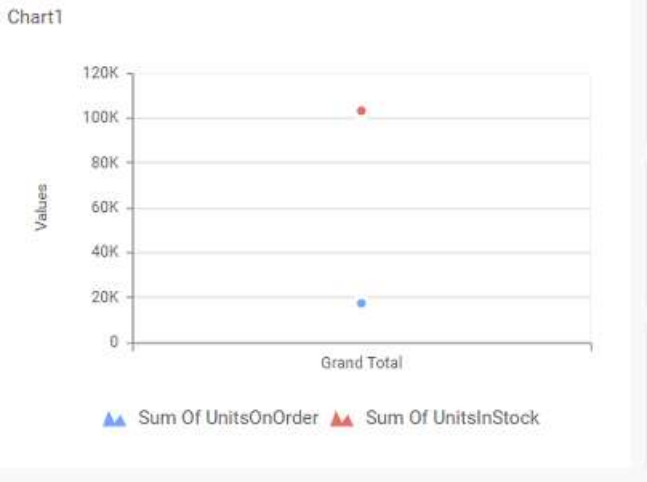
You can add the required data from Measures and Dimensions into required field.

Adding Y Values

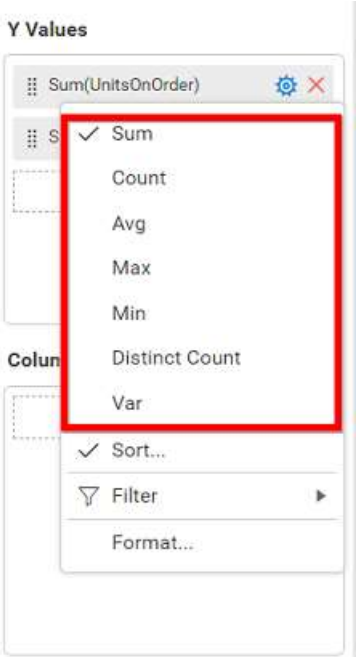
You can add more than one Measures into Y Values field by drag and drop the required measure.



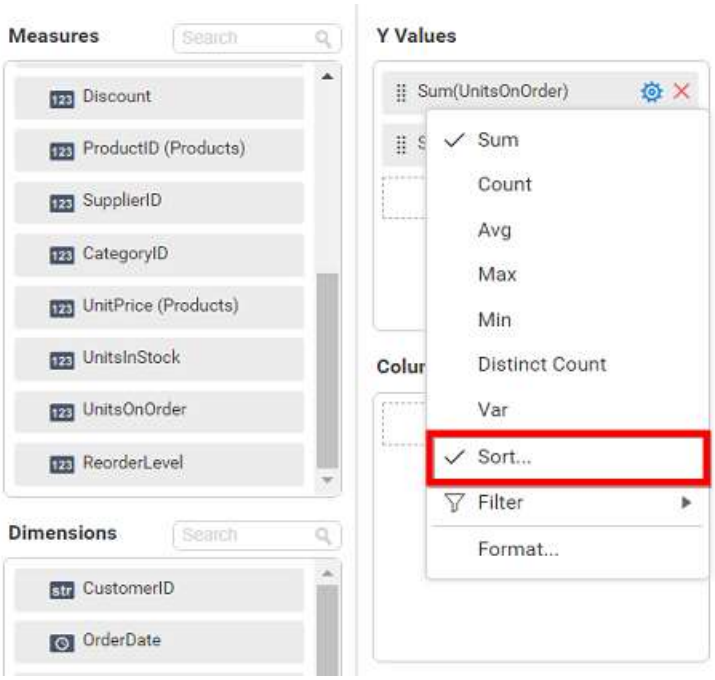
Now the stacked area chart will be rendered like this



Click the Settings option to change required summary type from the available summary types shown in Settings.

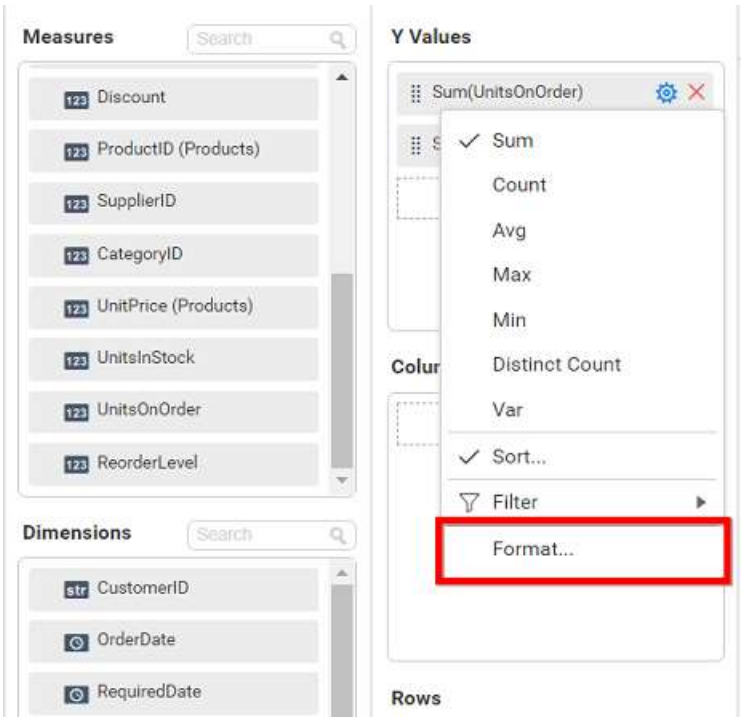


You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort



You can filter the data to be displayed in funnel chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

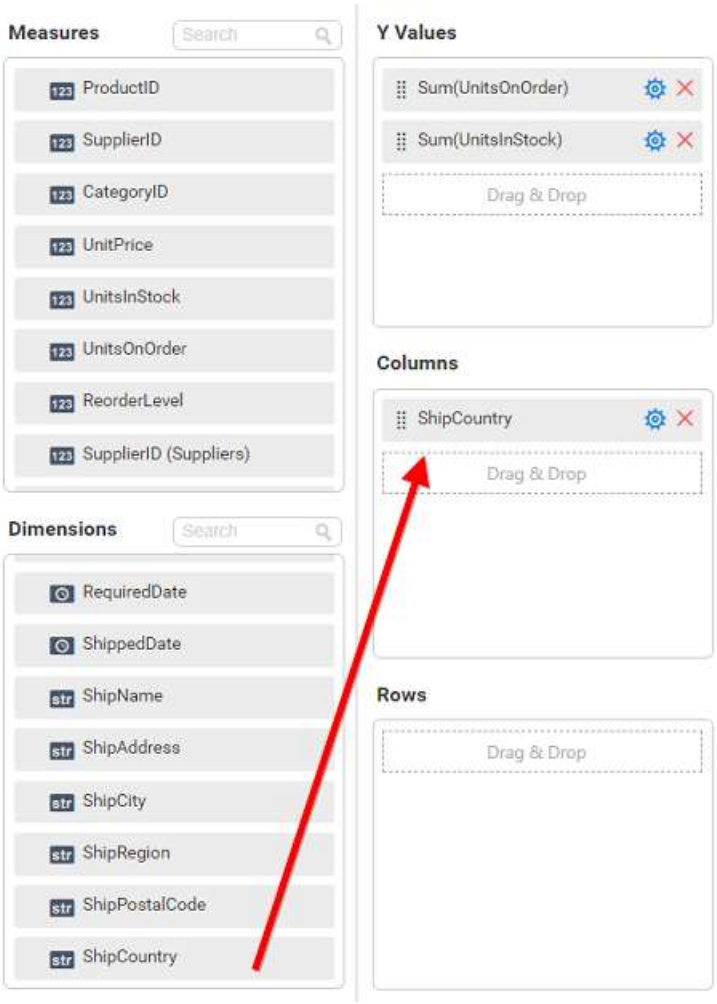


To remove the added value fields click highlighted button.

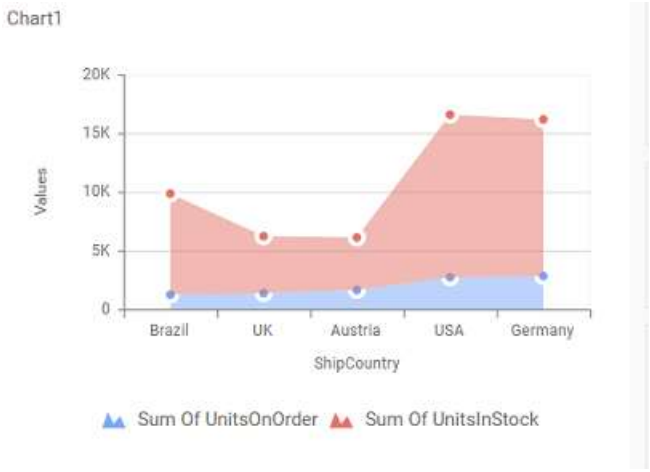
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



Stacked area chart will be rendered like this



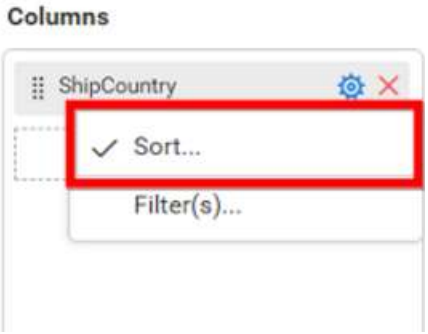
Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

Note: If you click No, single value will be added to the Columns field.

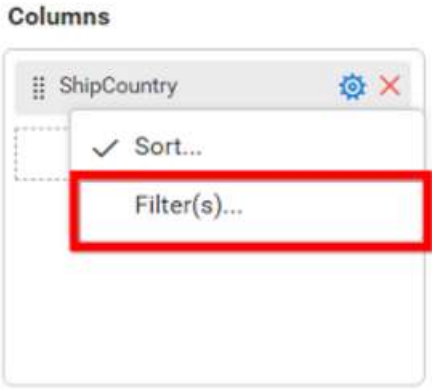
The drilled view of the chart region selected.

You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

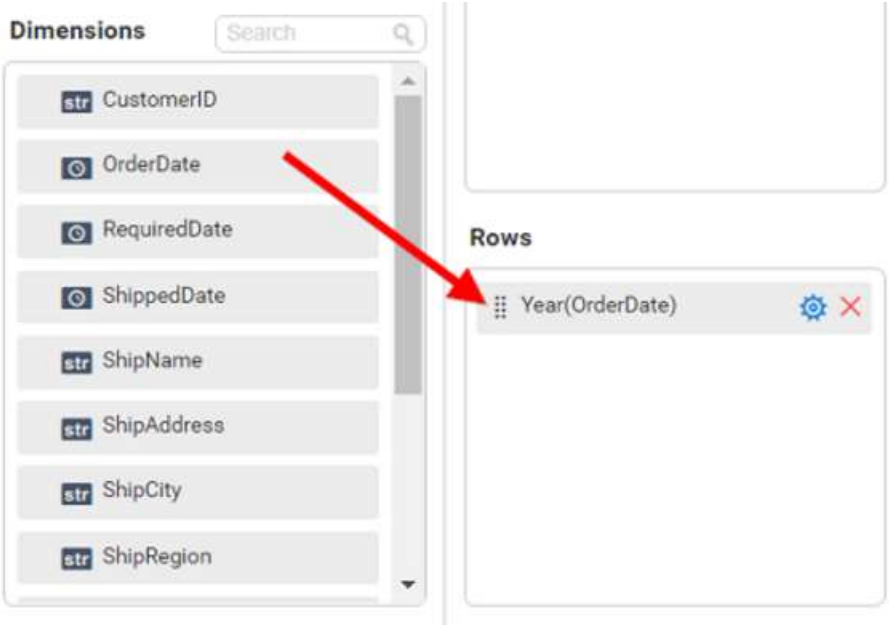


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

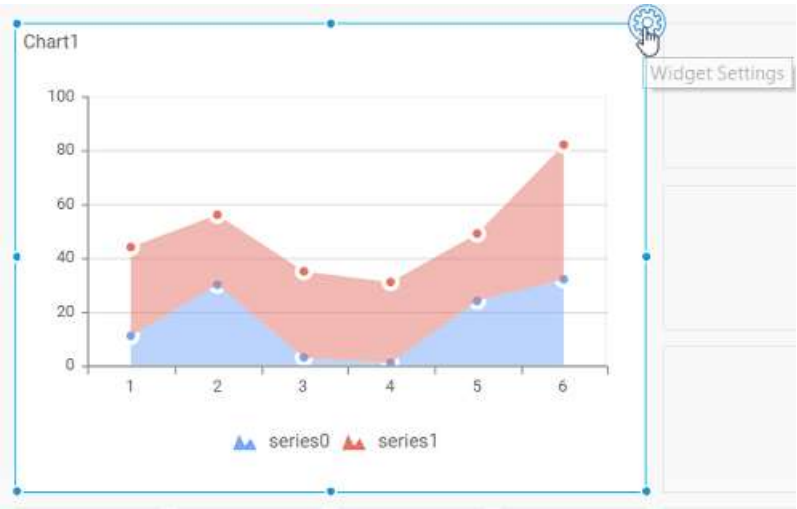
This will render stacked area chart in series.

How to format stacked area chart?

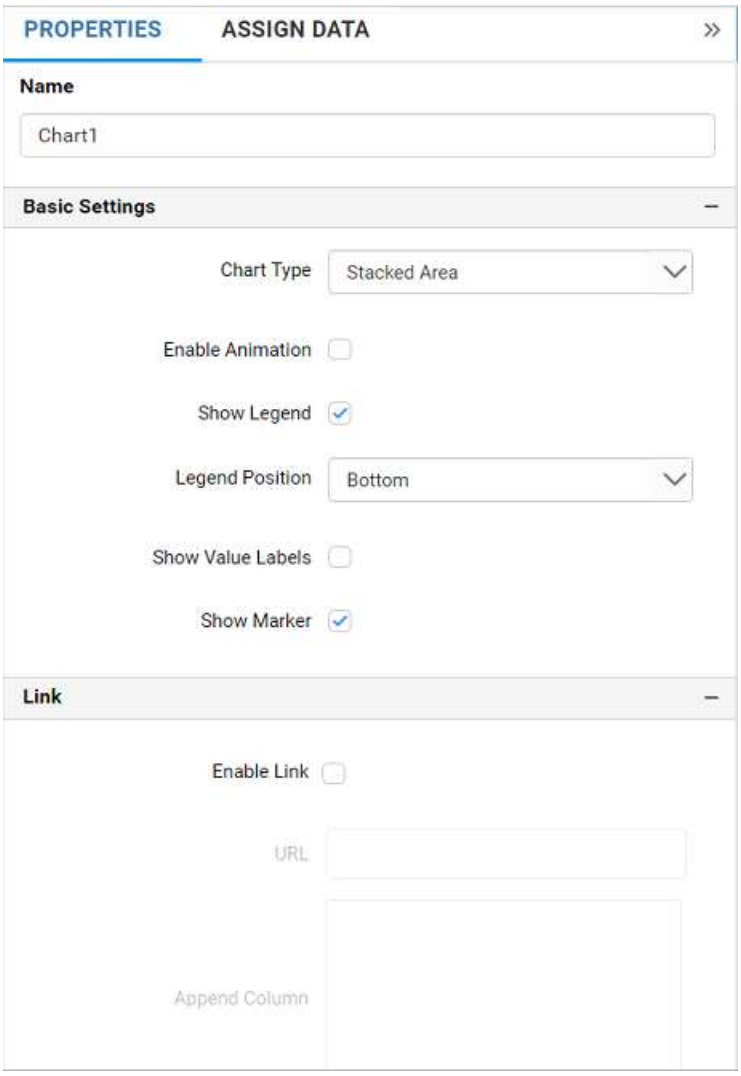
You can format the stacked area chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into stacked area chart follow the steps

1. Drag and drop the stacked area chart into canvas and resize it to your required size.
2. Configure the data into stacked area chart.
3. Focus on the stacked area chart and click on widget settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this stacked area chart widget.

Basic Settings

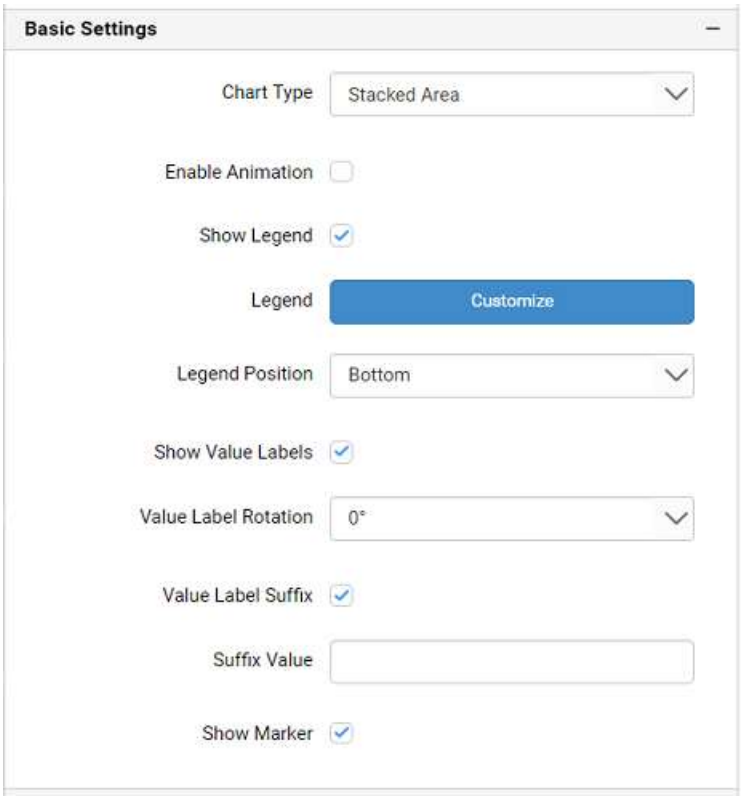


Chart Type

This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).

Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

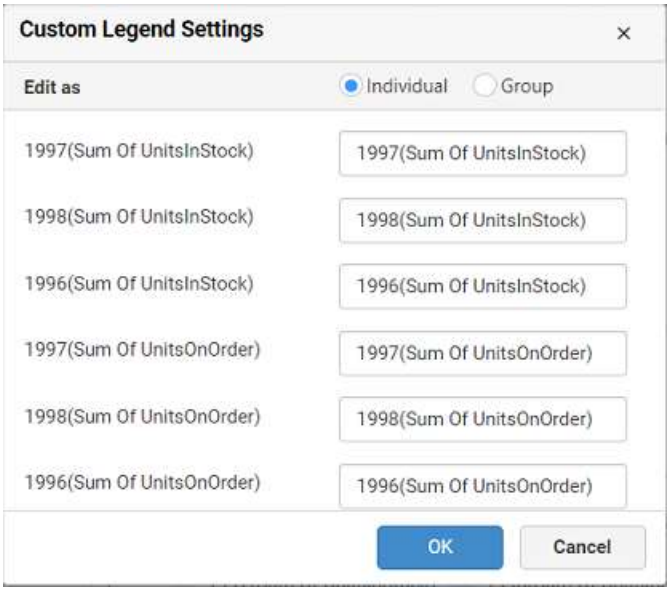
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

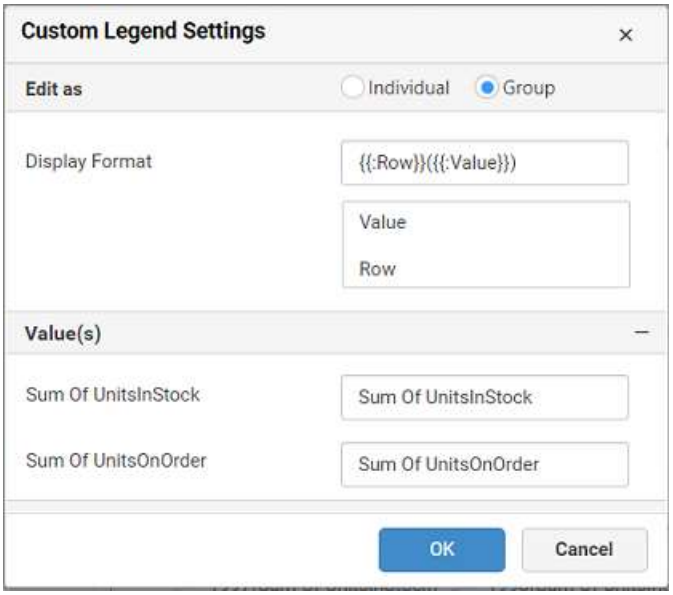
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.



Group

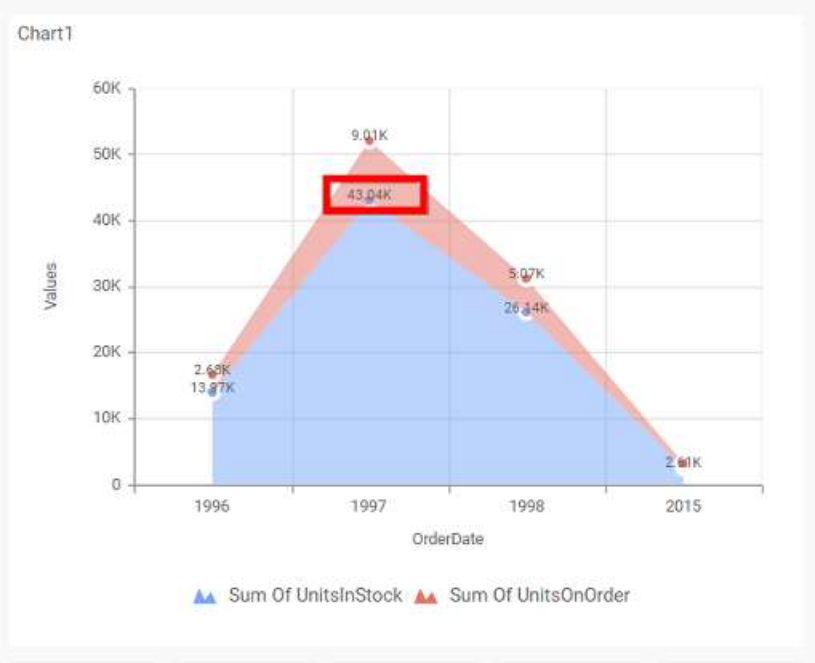
Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like 1996 (Sum Of UnitsInStock)

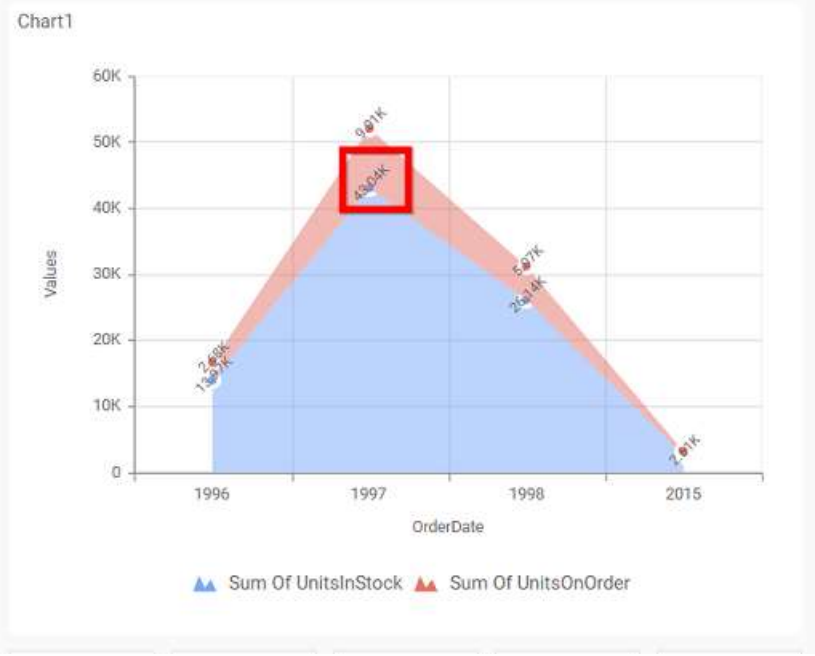
Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

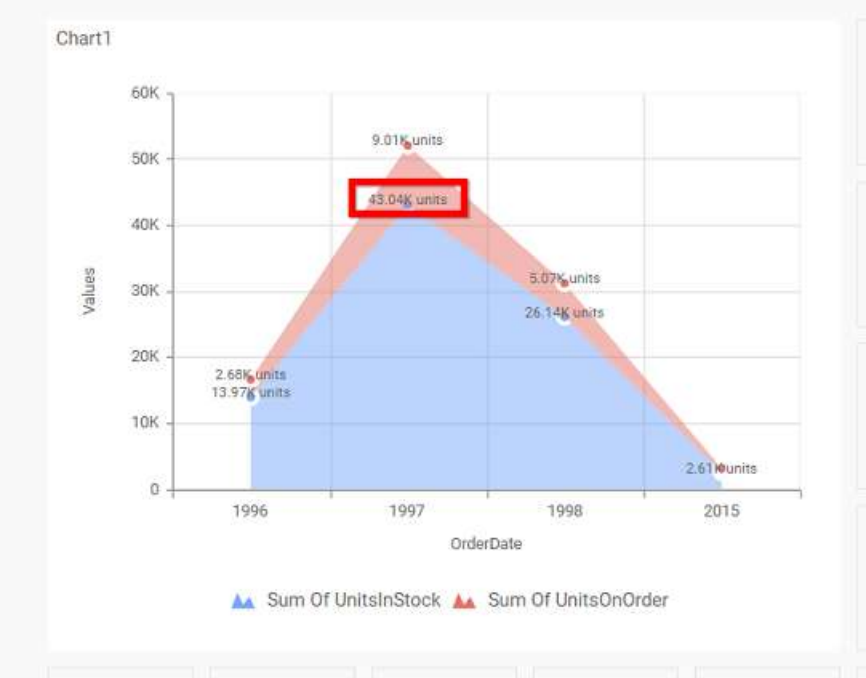


Value Label Suffix

Allows you to enable the Suffix value text to the value labels.

Suffix Value

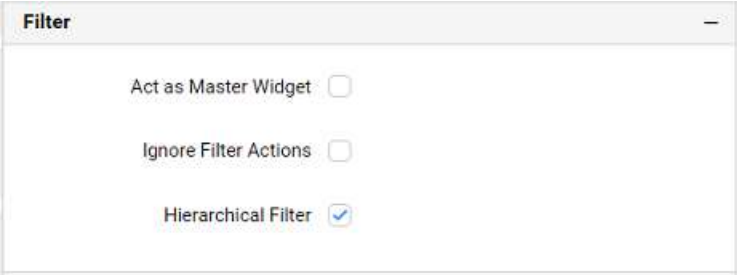
Allows you to set\edit suffix value to the value labels.



Show Marker

This allows you to toggle the visibility of marker from label to adorn each data point in chart series.

Filter



Act as Master Widget

This allows you to define this 100% stacked area chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

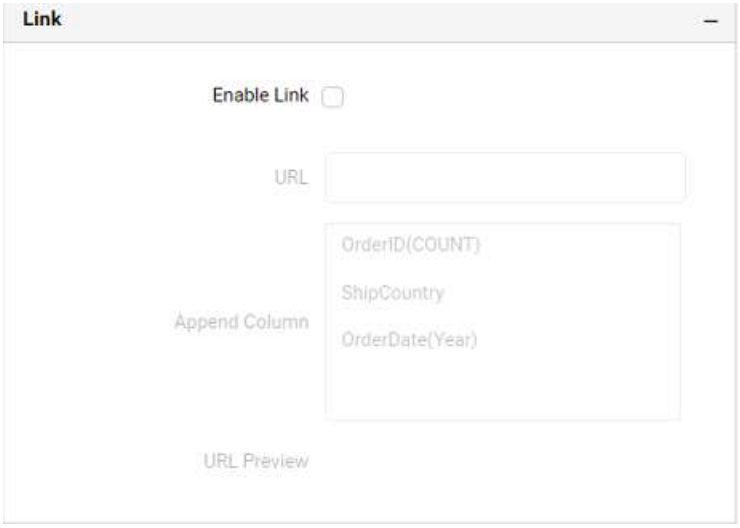
This allows you to define this 100% stacked area chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

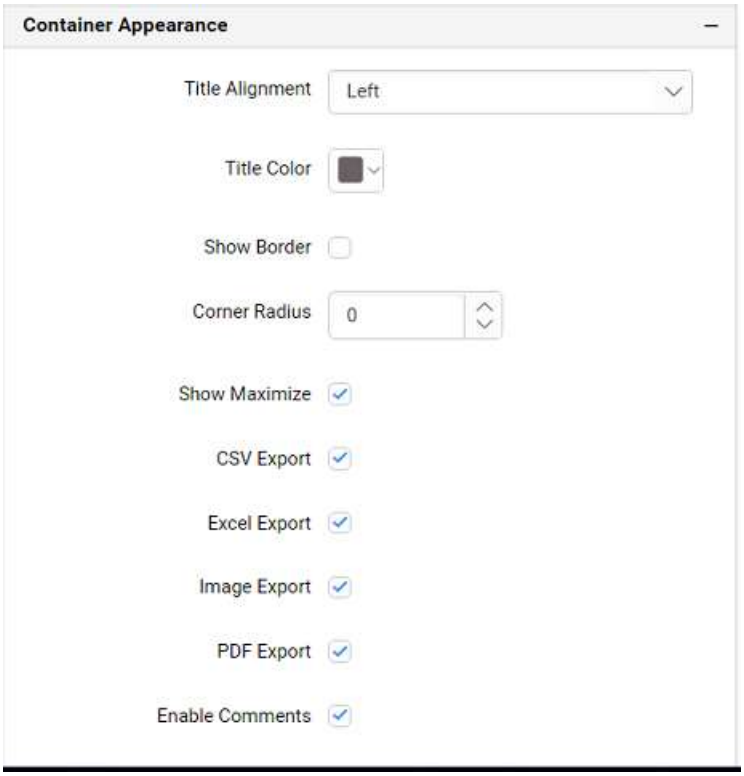
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to enable the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** property is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this stacked area chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this stacked area chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this stacked area chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

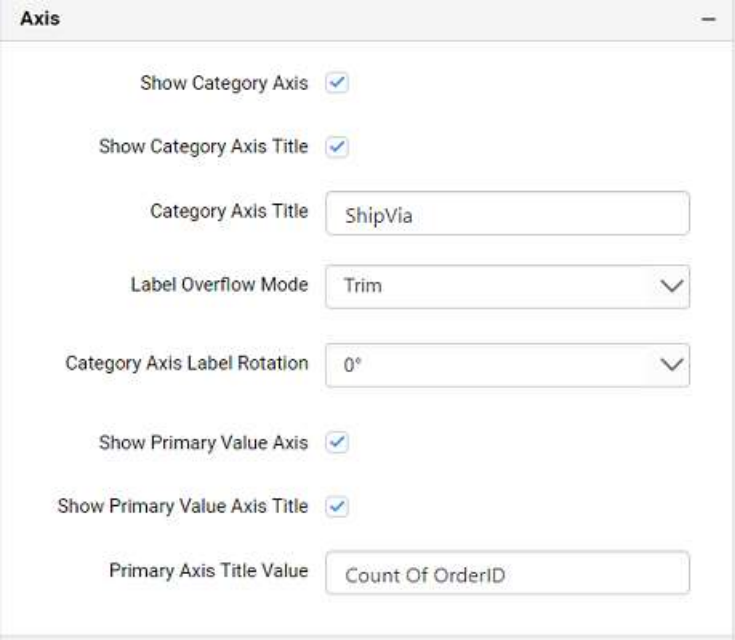
Image Export

This allows you to enable/disable the image export option for this stacked area chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

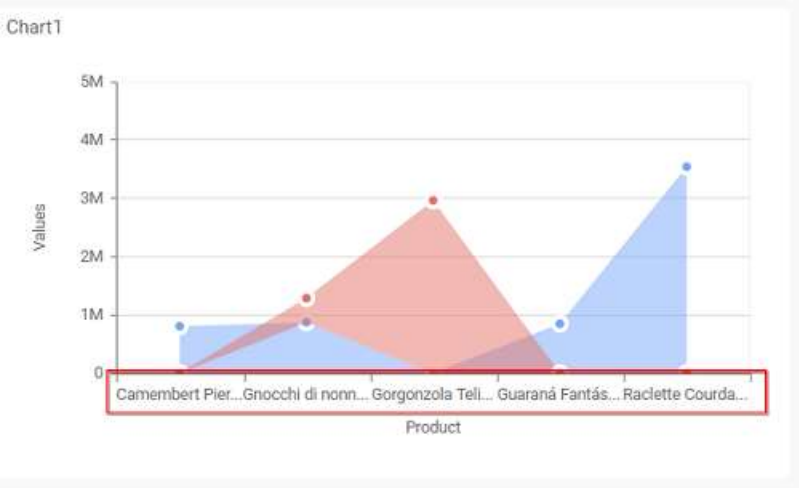
Axis Settings



This section allows you to customize the axis settings in chart.

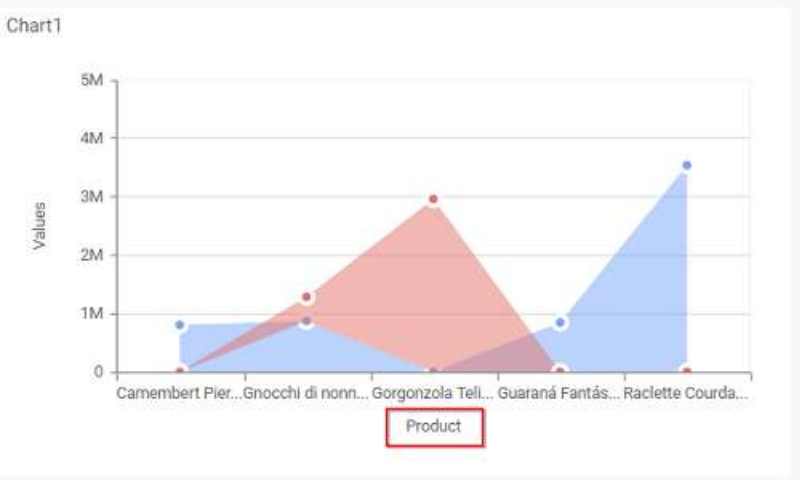
Show Category Axis

This allows to enable the visibility of Category Axis.



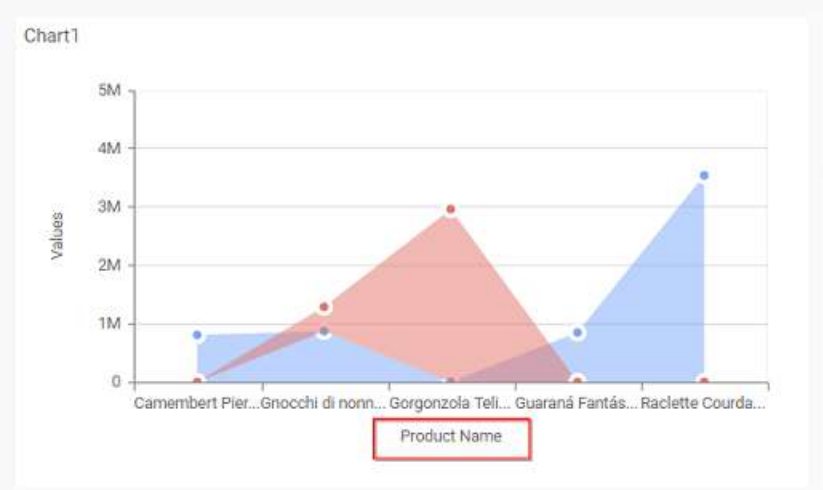
Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

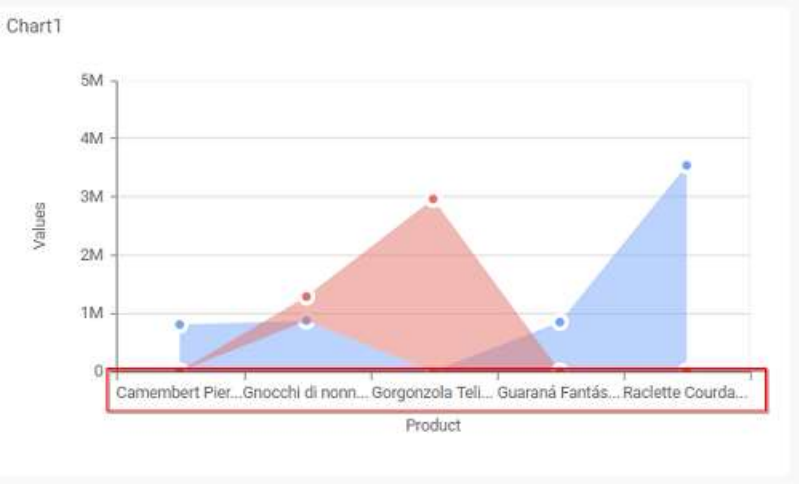


Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

Trim

This option trims the end of overlapping label in the axis.

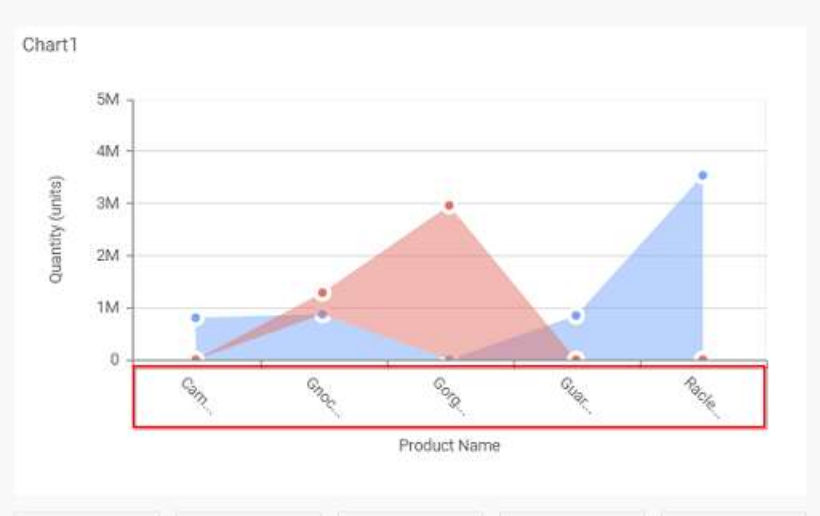


Hide

This option hides the overlapping label in the axis.

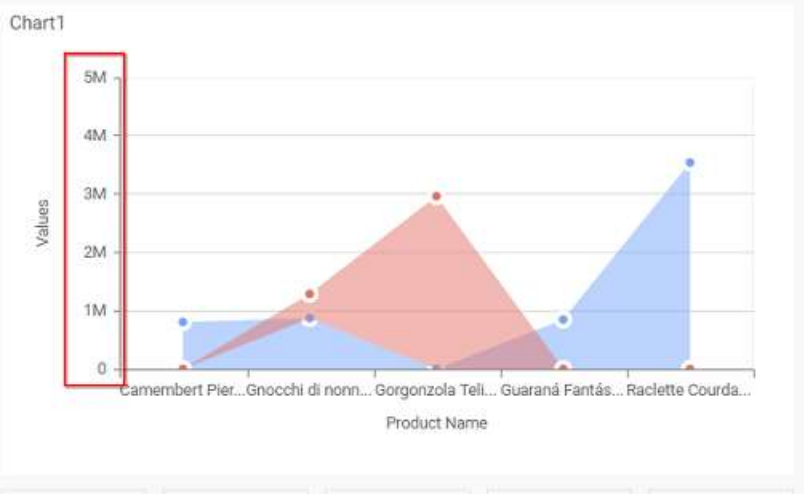
Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



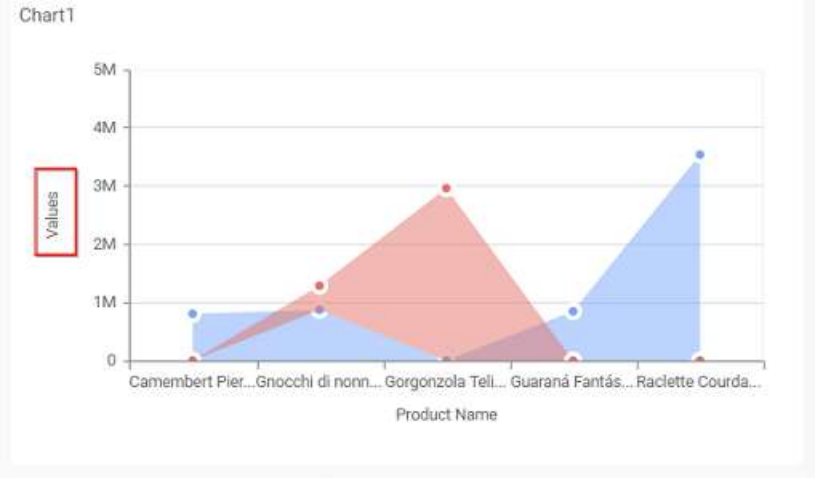
Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.



Show Primary Value Axis Title

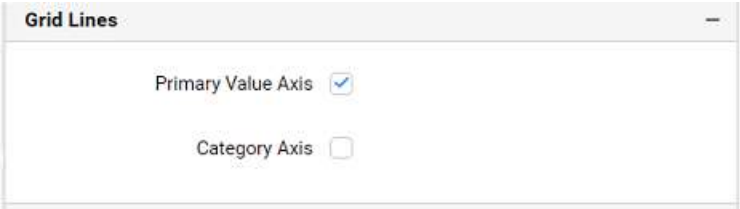
This allows you to enable the visibility of Primary Value Axis title of chart.



Primary Value Axis Title

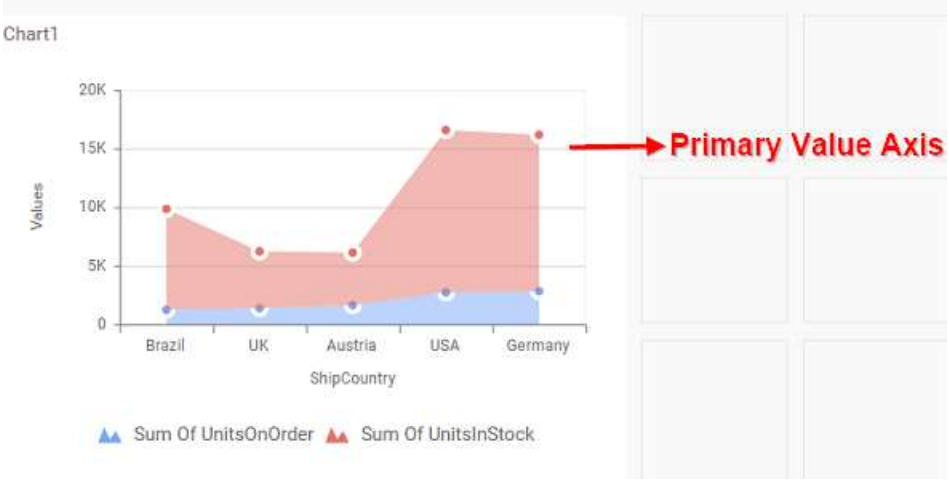
This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

Grid Line



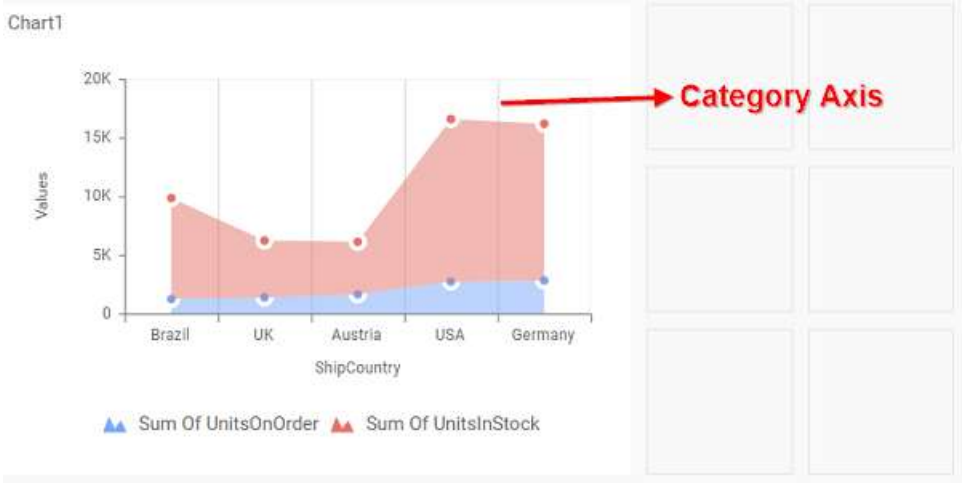
Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the stacked area chart.



Category Axis

This allows you to enable the Category axis gridlines for the stacked area chart.

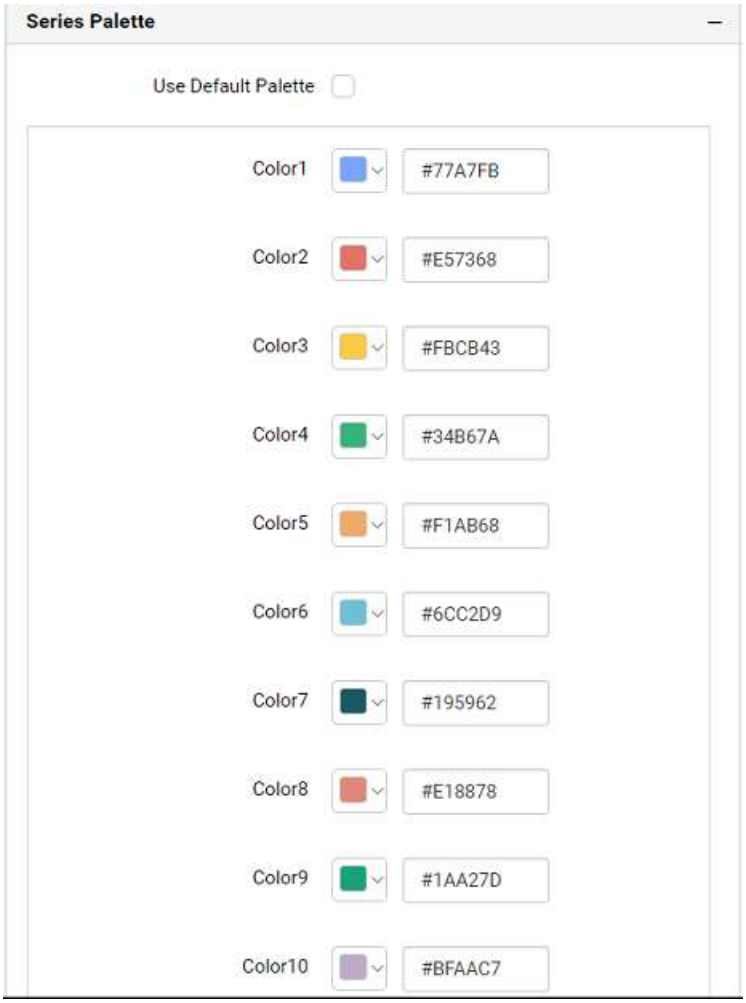


Series Palette

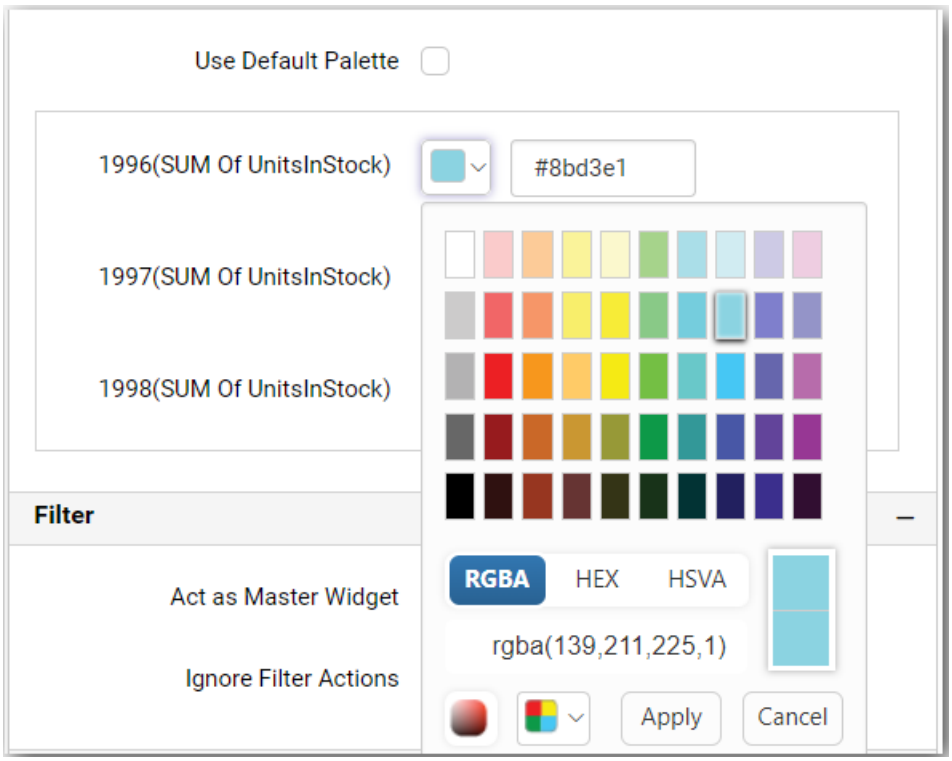
This allows you to customize the chart series color through Series Palette section.

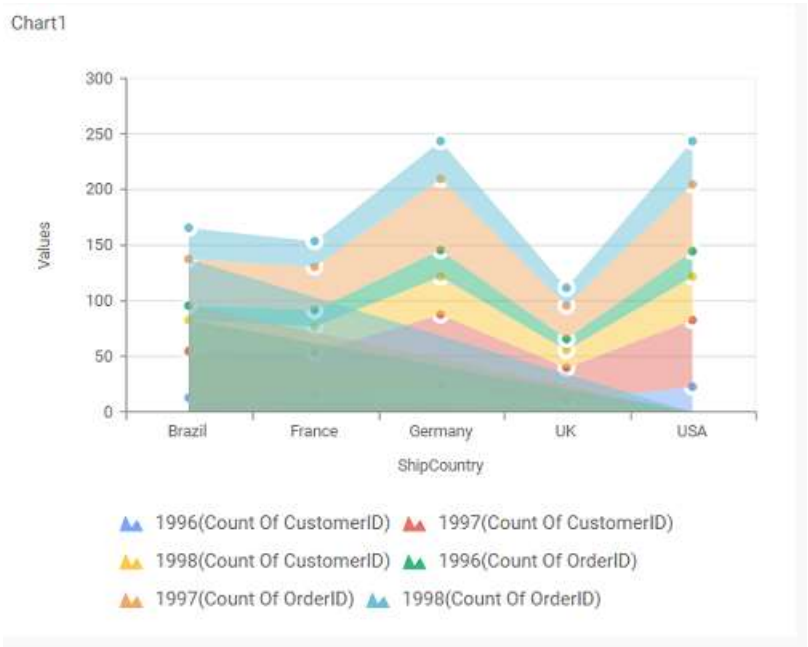
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





100% Stacked Area Chart

100% Stacked Area Chart allows you to compare multiple measures through filled curves stacked one after the other vertically.

How to configure the table data to 100% stacked area chart?

100% Stacked Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

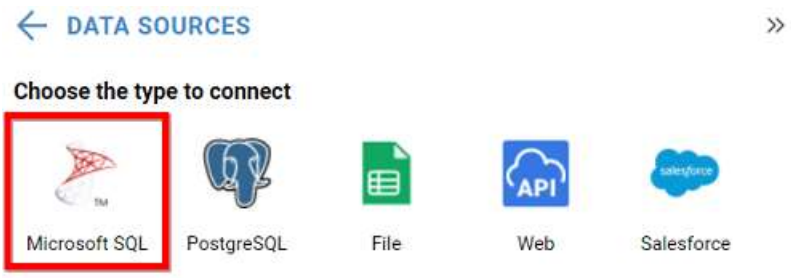
Follow the steps to configure data to 100% stacked area chart

Drag and drop the 100% stacked area chart into canvas and resize it to your required size.

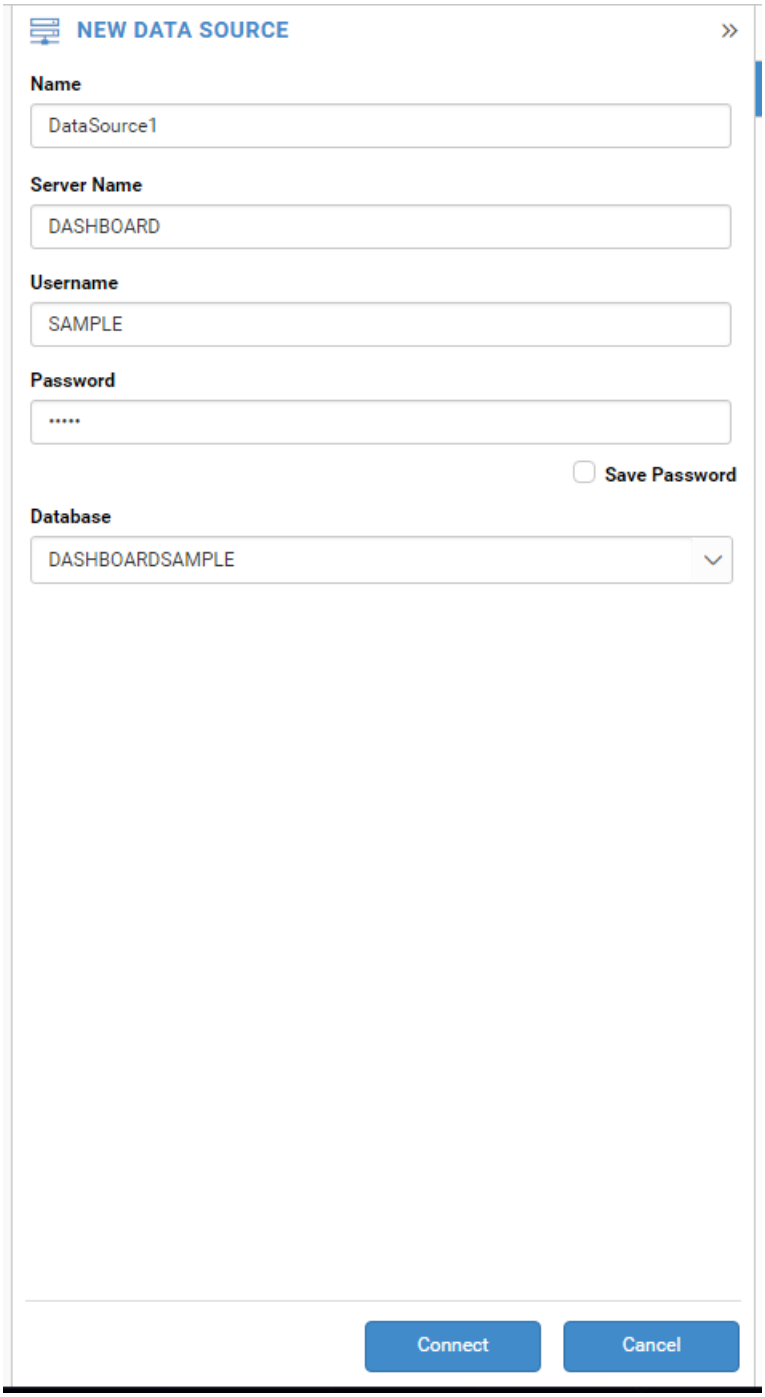
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

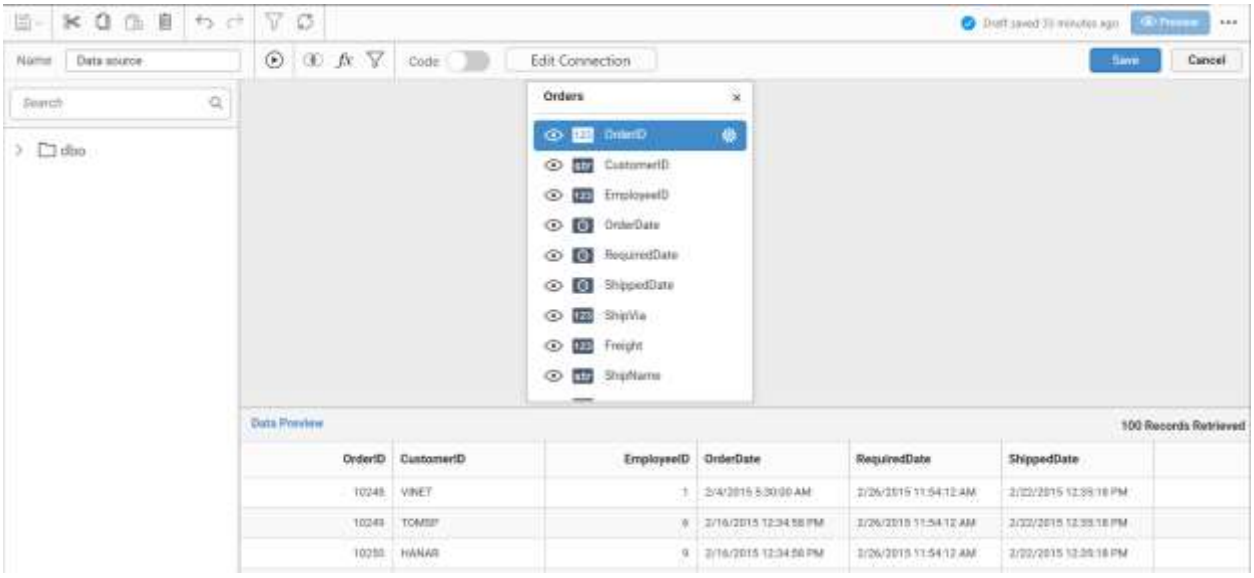
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



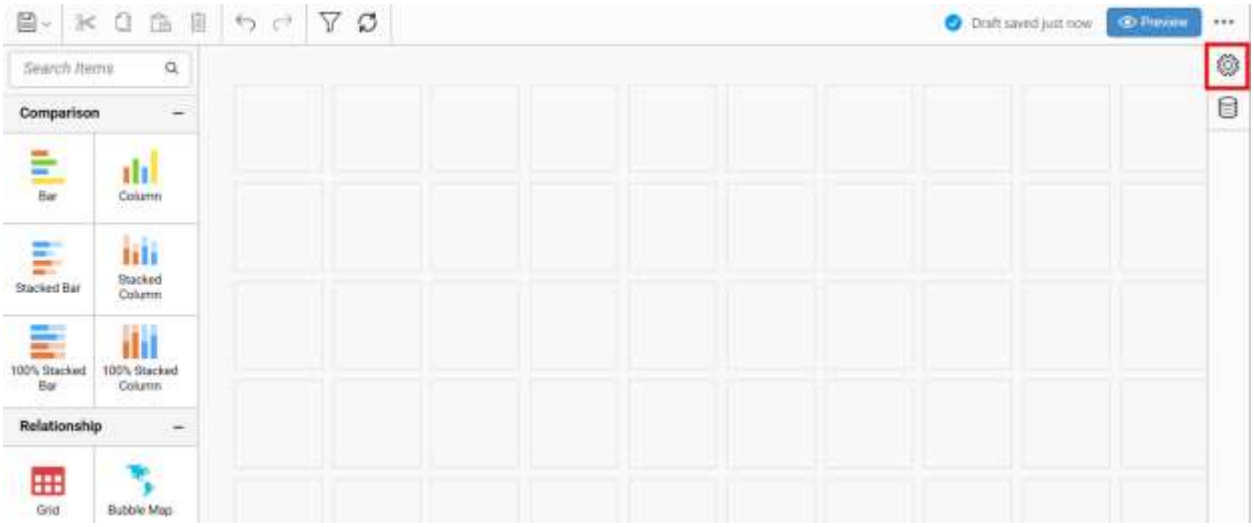
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

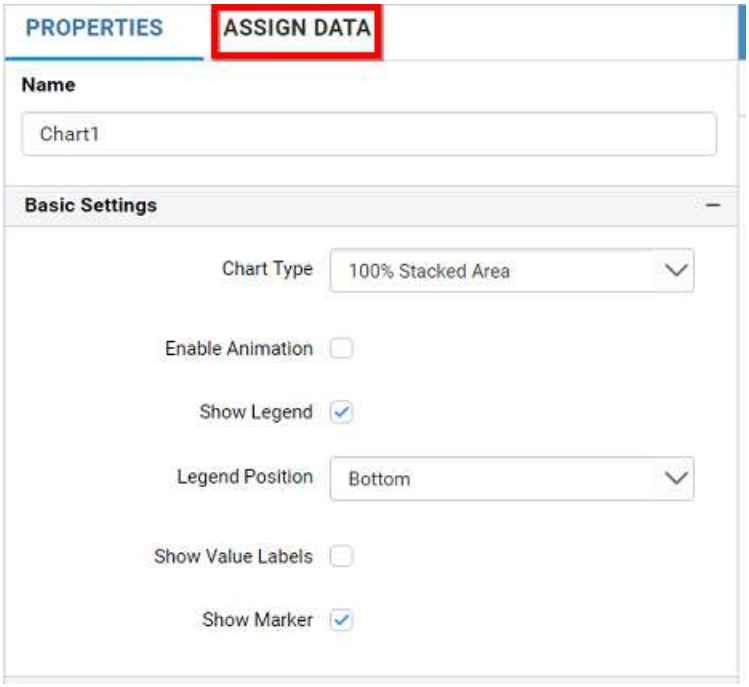


Drag your preferred table or view from the left pane from data design view, click Save button.

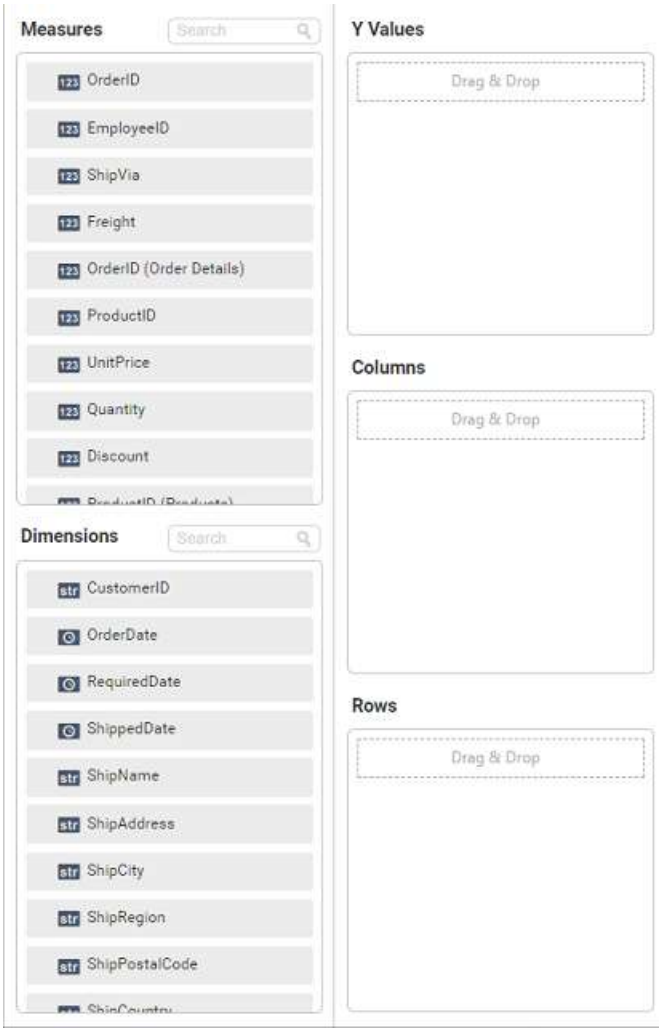


Click Properties button in configuration panel, property pane opens. Now, switch to ASSIGN DATA tab.





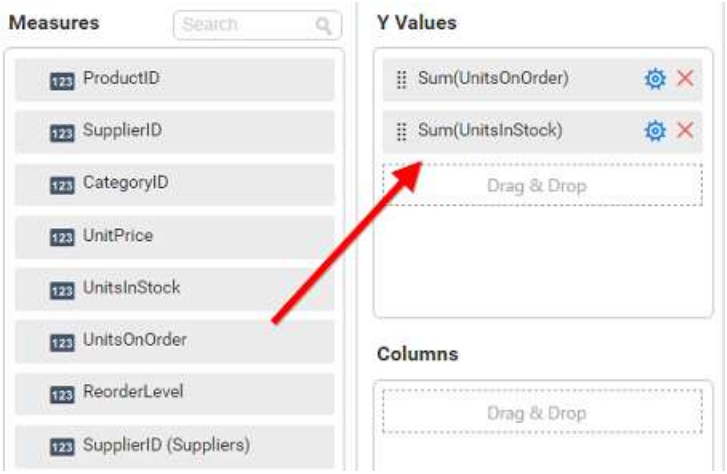
The data tab will be opened with available measures and dimensions from the connected data source



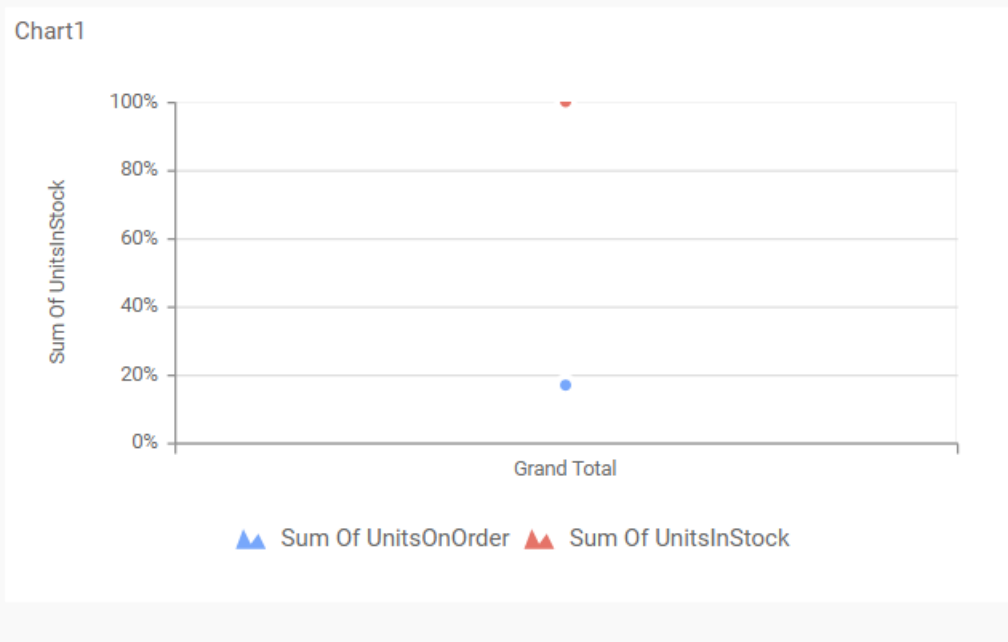
You can add the required data from Measures and Dimensions into required field.

Adding Y Values

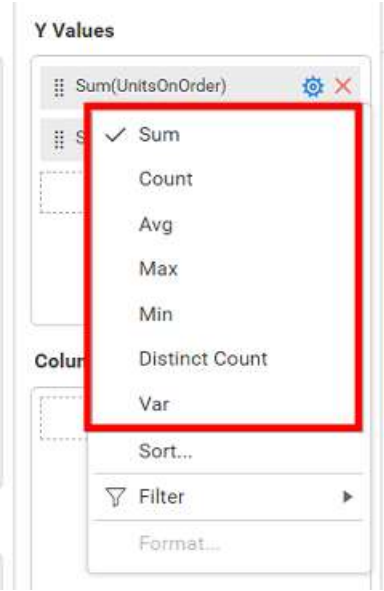
You can add more than one Measures into Y Values field by drag and drop the required measure.



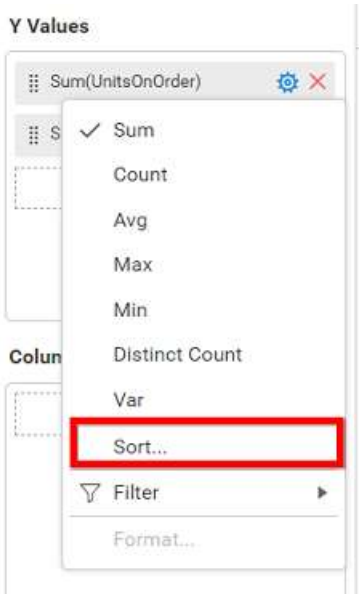
Now the chart will be rendered like this



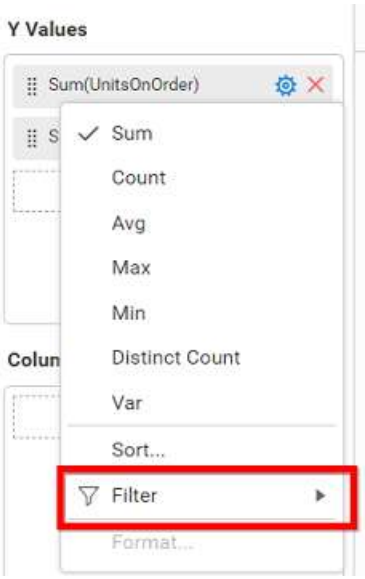
Click the Settings option to change required summary type from the available summary types shown in Settings.



You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort



You can filter the data to be displayed in chart by using filter option. For more details, refer filter.



To remove the added value fields click highlighted button.

You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.

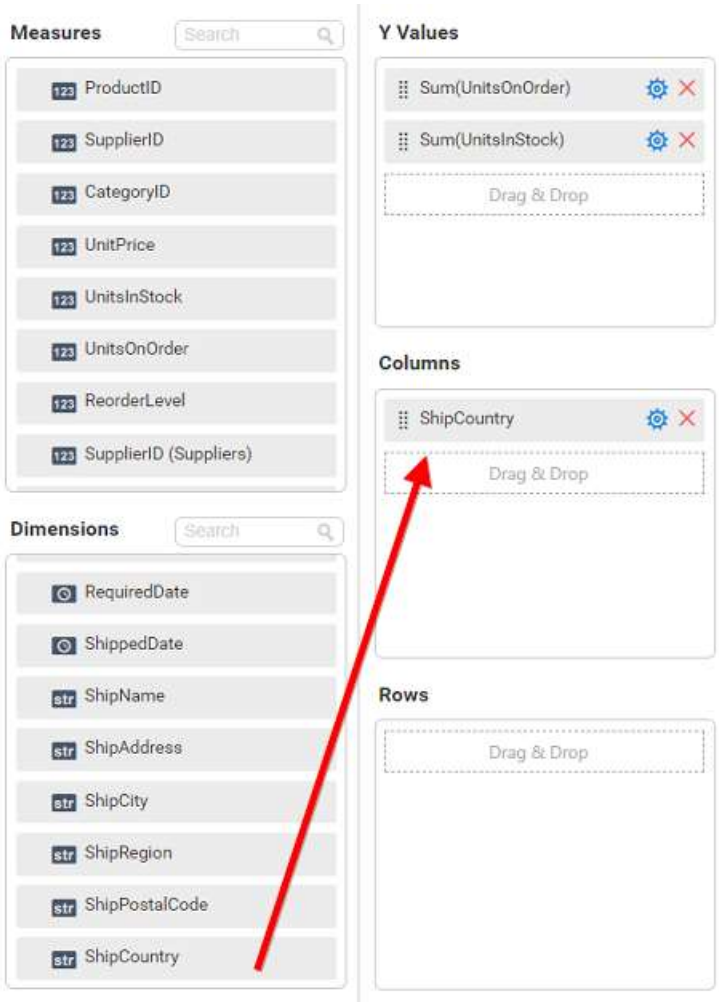
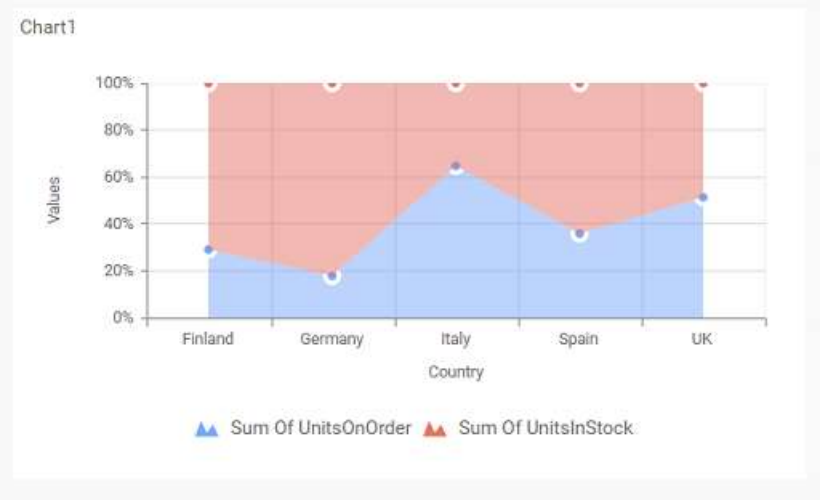


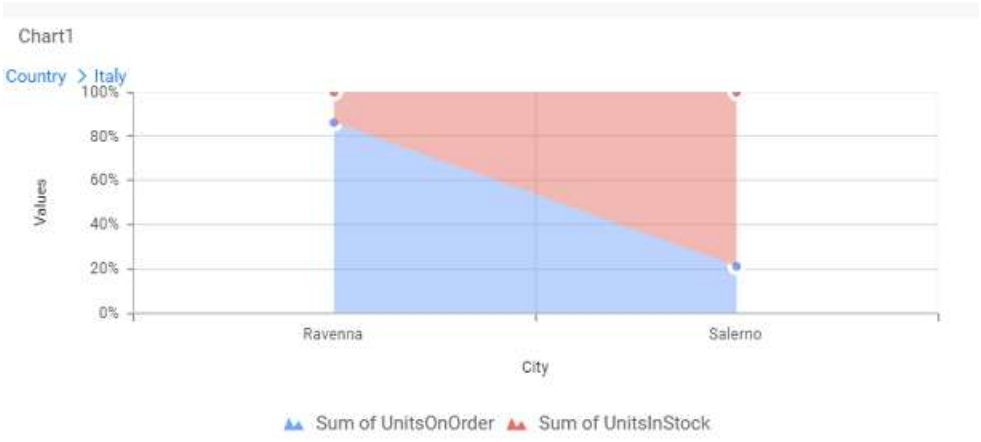
Chart will be rendered like this



Add more than one value to Columns field, the alert message will be shown to enable the drill down option. Click Yes to enable the option.

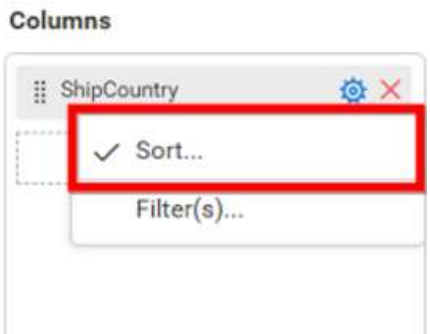
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.

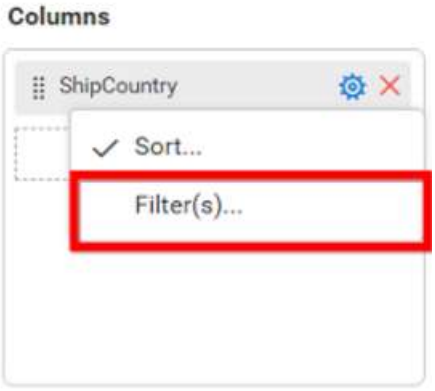


You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

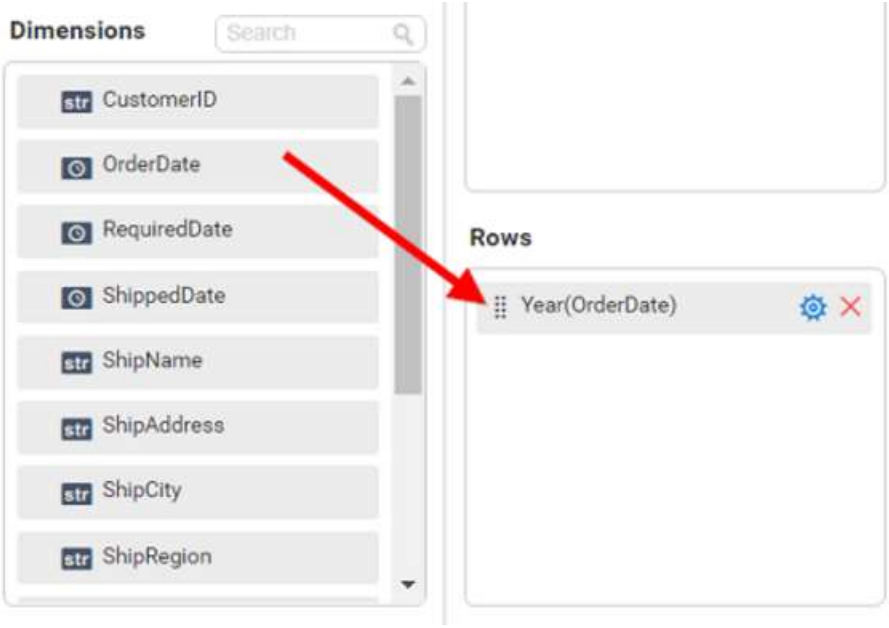


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

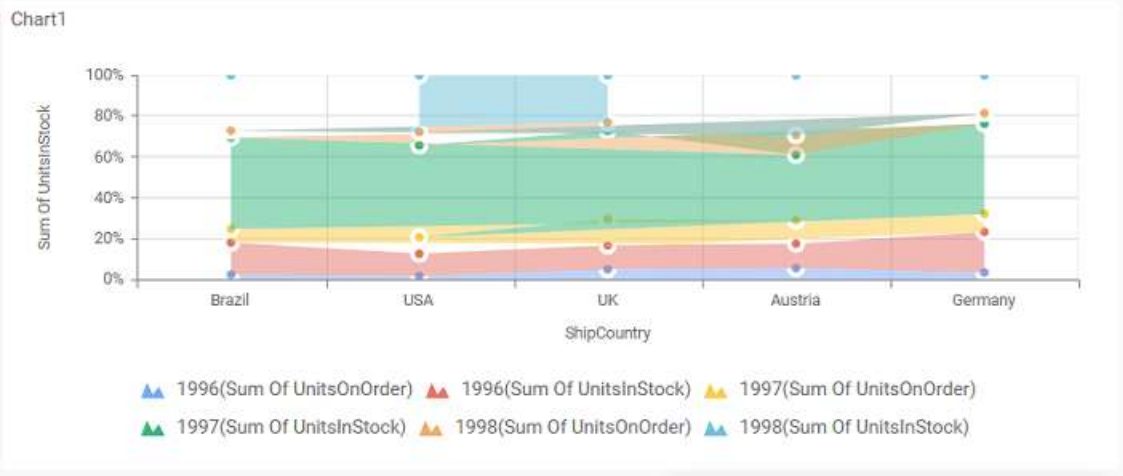
Adding Rows

You can drag and drop the Measure or Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render chart in series.



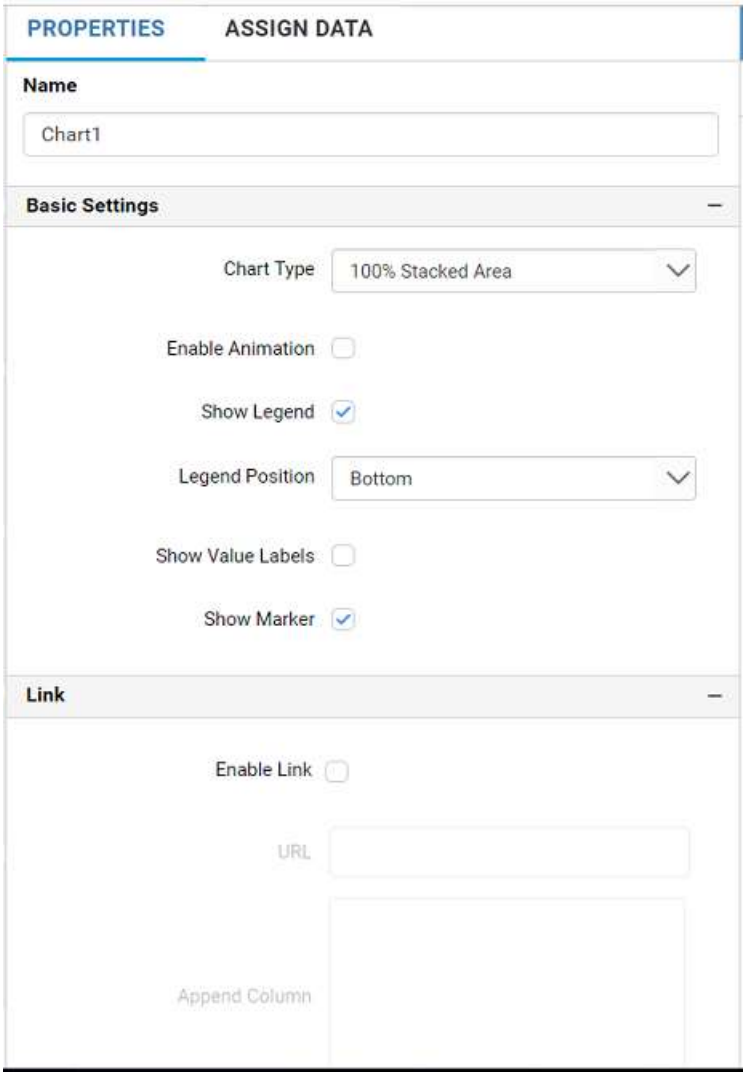
How to format 100% stacked area chart?

You can format the 100% stacked area chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into 100% stacked area chart follow the steps

1. Drag and drop the 100% stacked area chart into canvas and resize it to your required size.
2. Configure the data into 100% stacked area chart.
3. Focus on the 100% stacked area chart and click on widget settings.

The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this 100% stacked area chart widget.

Basic Settings

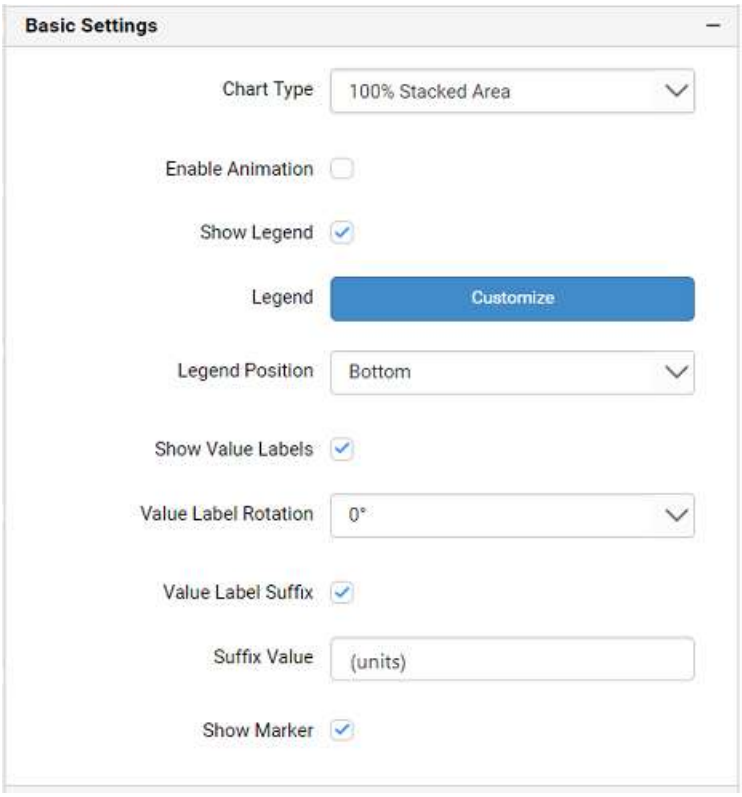


Chart Type

This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).

Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

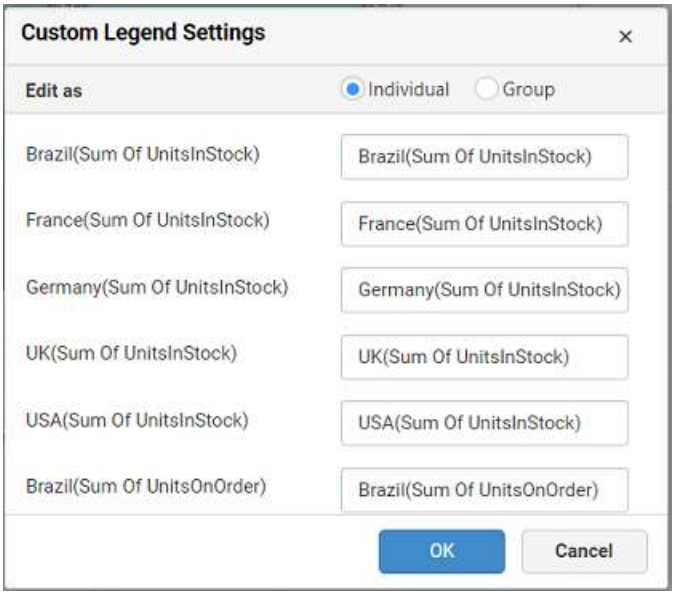
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

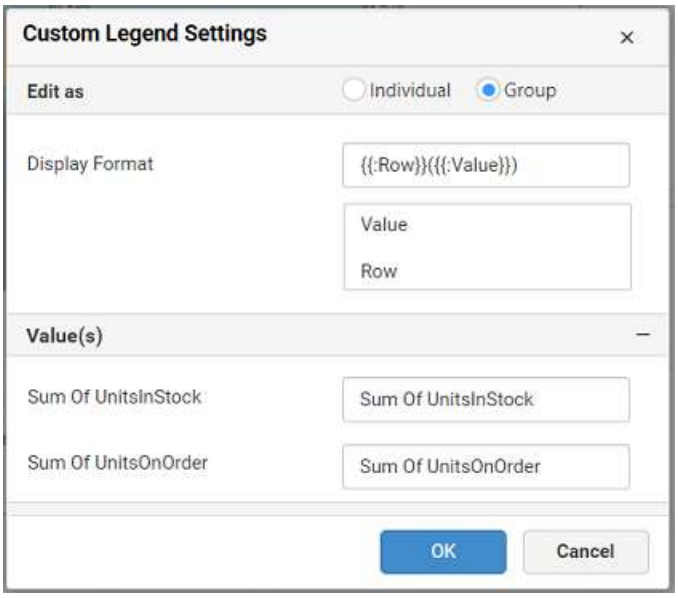
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Rows represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.

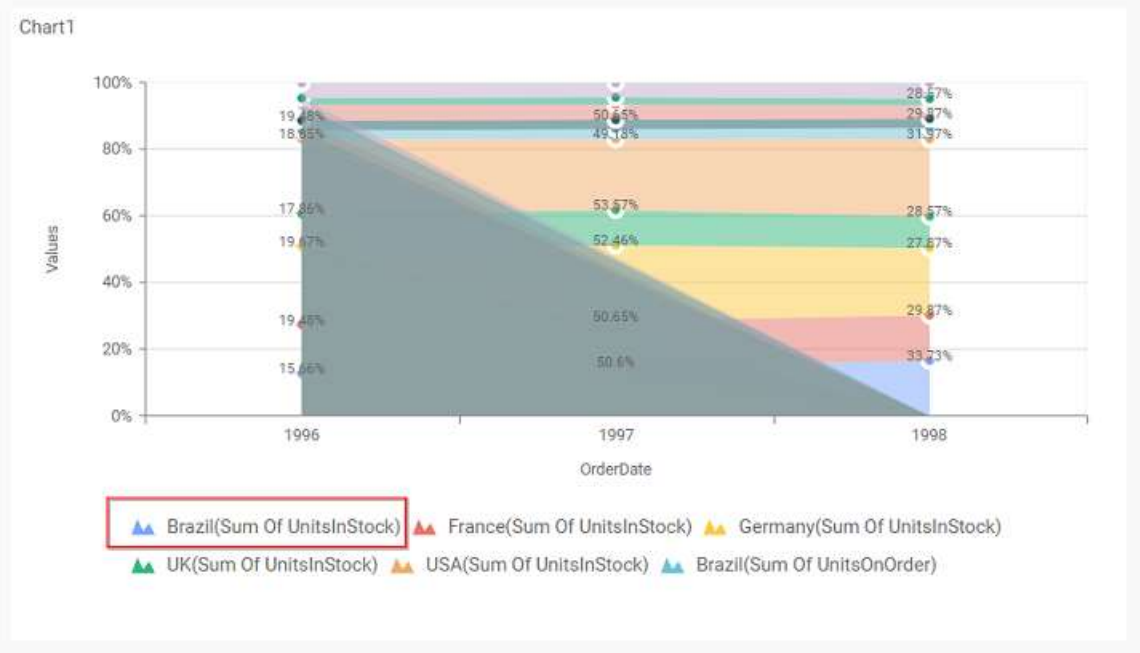


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

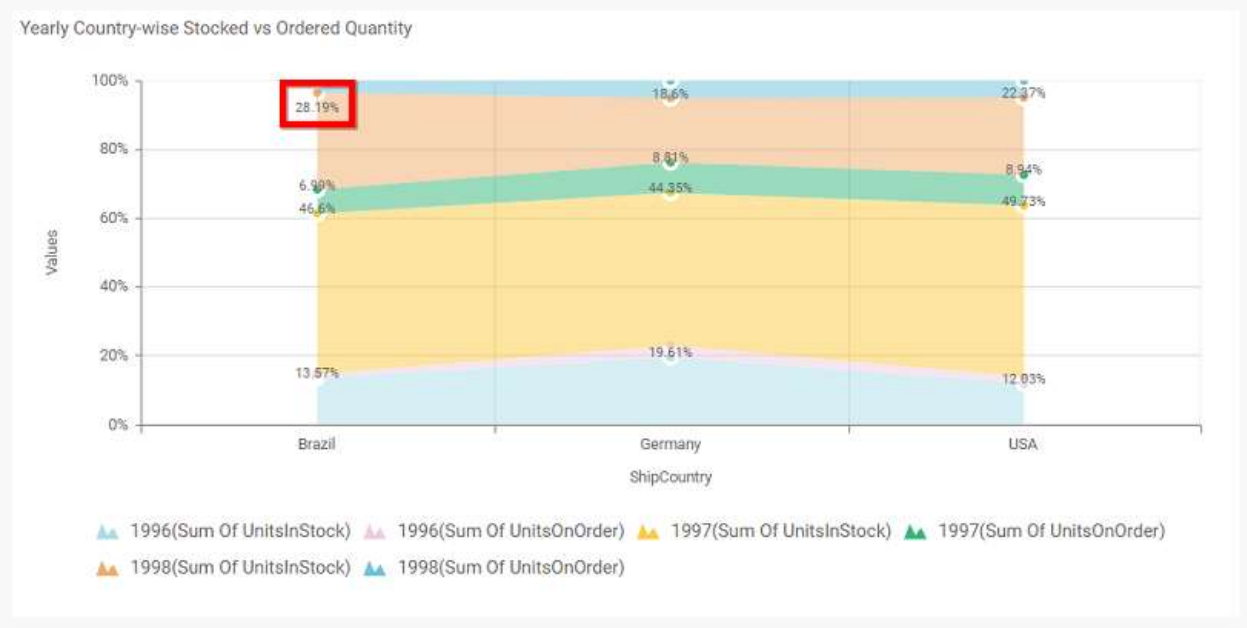


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Brazil (Sum of UnitsInStock)



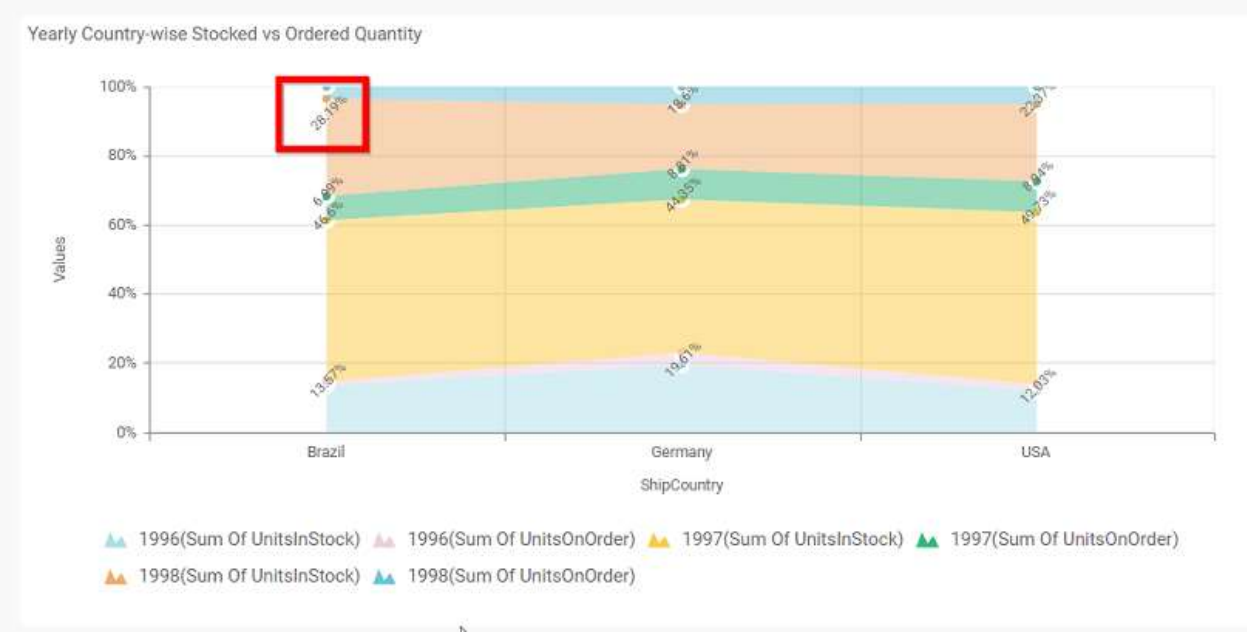
Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



Value Label Suffix

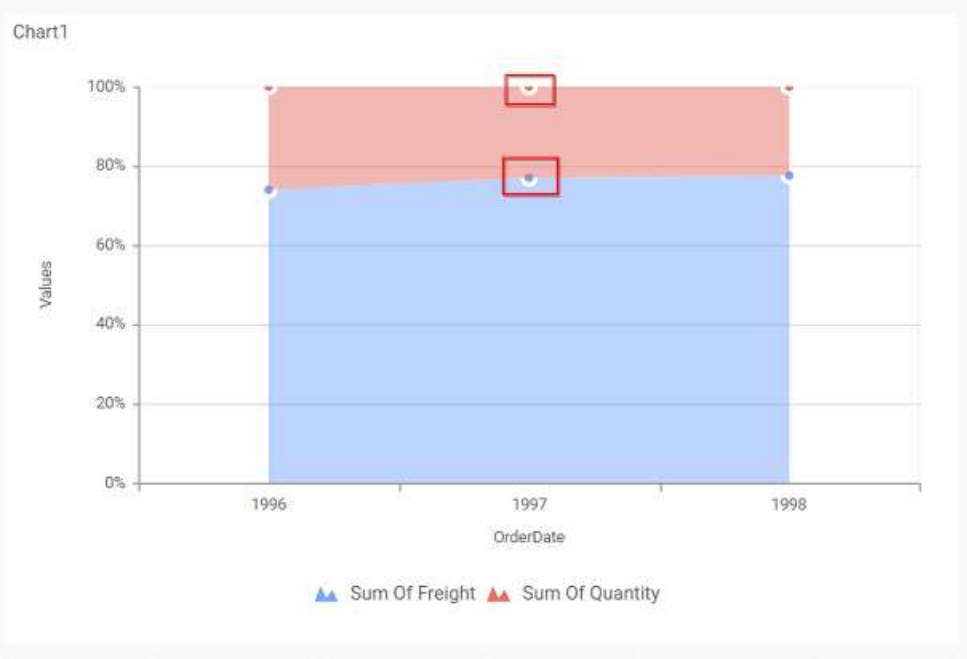
Allows you to enable the Suffix value text to the value labels.

Suffix Value

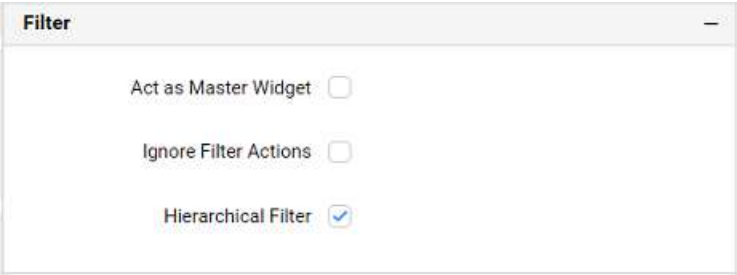
Allows you to set\edit suffix value to the value labels.

Show Marker

This allows you to toggle the visibility of marker from label to adorn each data point in chart series.



Filter



Act as Master Widget

This allows you to define this 100% stacked area chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

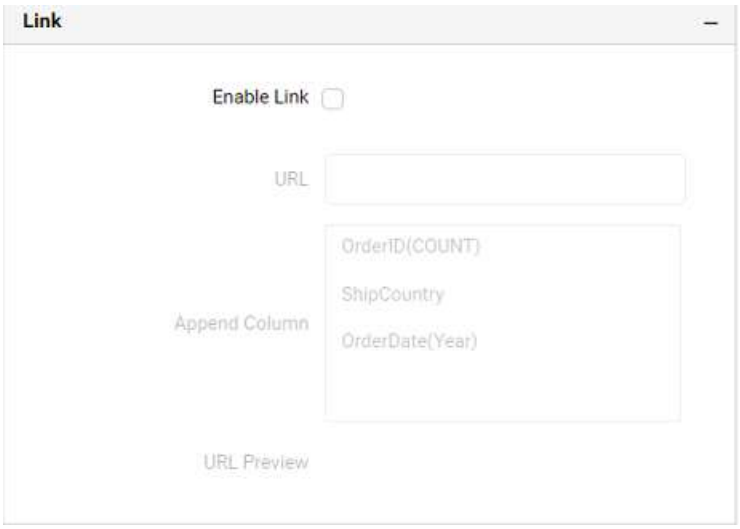
This allows you to define this 100% stacked area chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

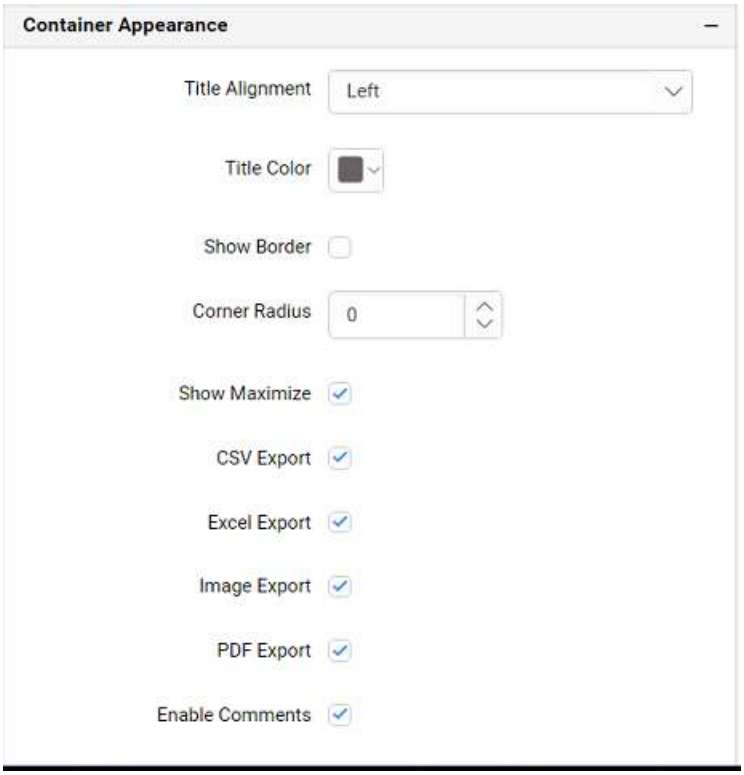
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this 100% stacked area chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this 100% stacked area chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this 100% stacked area chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

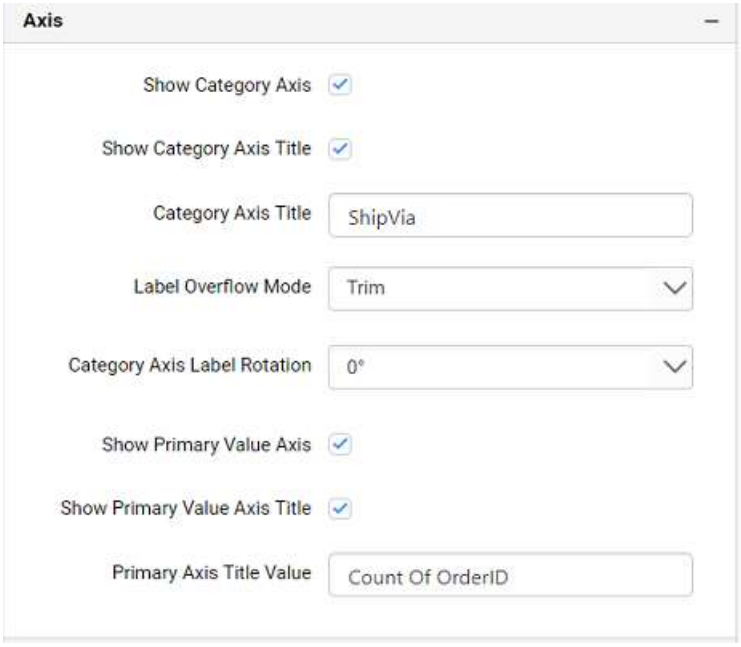
Image Export

This allows you to enable/disable the image export option for this 100% stacked area chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

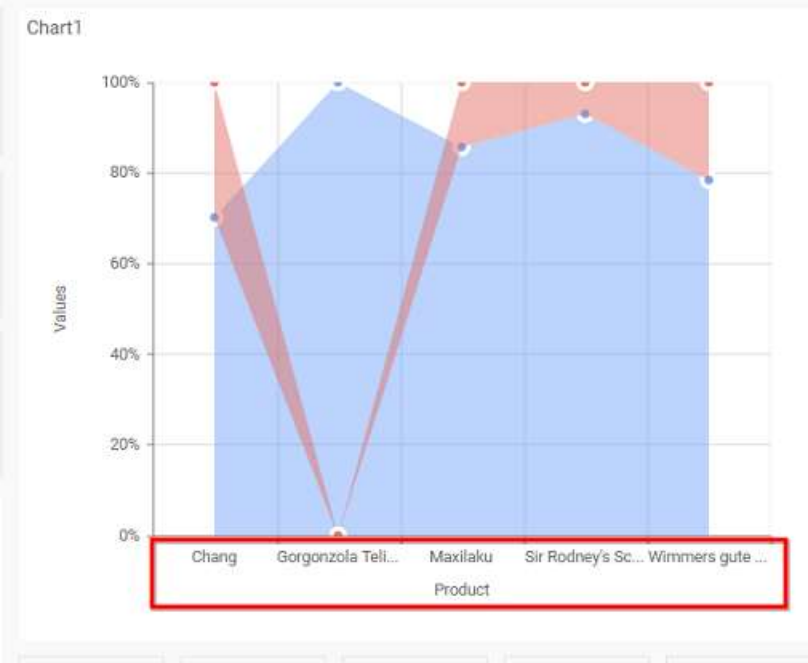
Axis Settings



This section allows you to customize the axis settings in chart.

Show Category Axis

This allows to enable the visibility of Category Axis.



Show Category Axis Title

This allows you to enable the visibility of Category Axis title.

Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

Trim

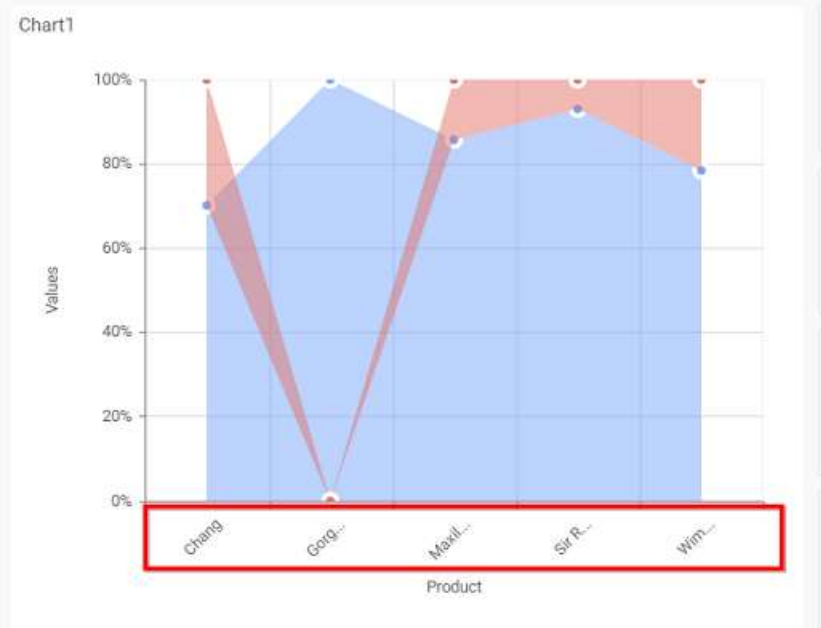
This option trims the end of overlapping label in the axis.

Hide

This option hides the overlapping label in the axis.

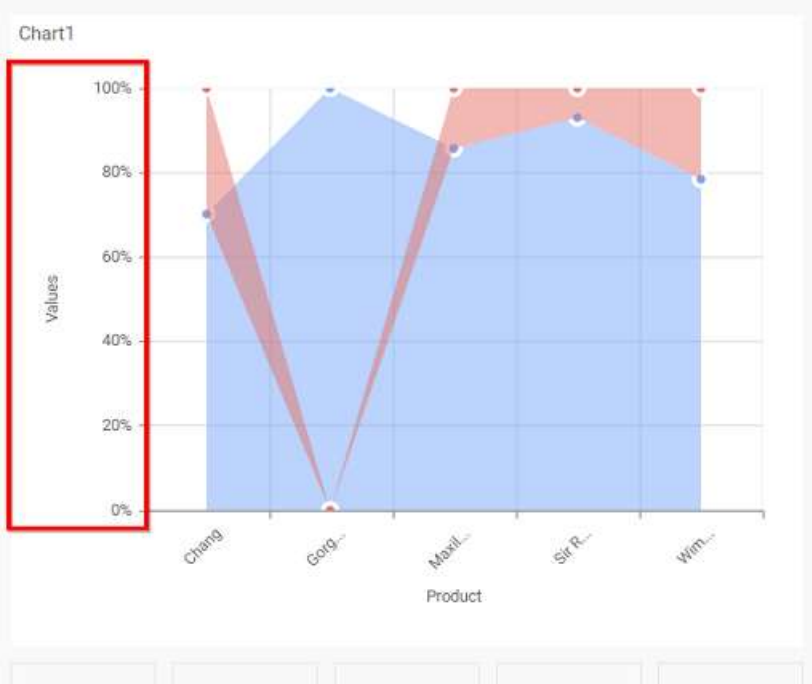
Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



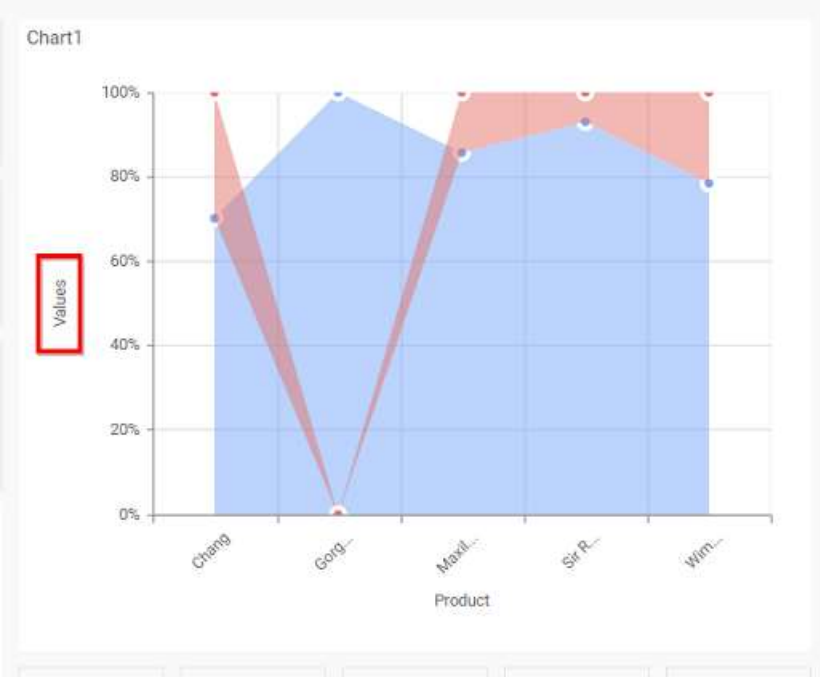
Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.



Show Primary Value Axis Title

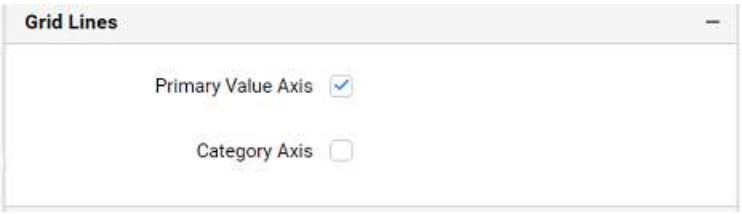
This allows you to enable the visibility of Primary Value Axis title of chart.



Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

Grid Line



Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the 100% stacked area chart.

Category Axis

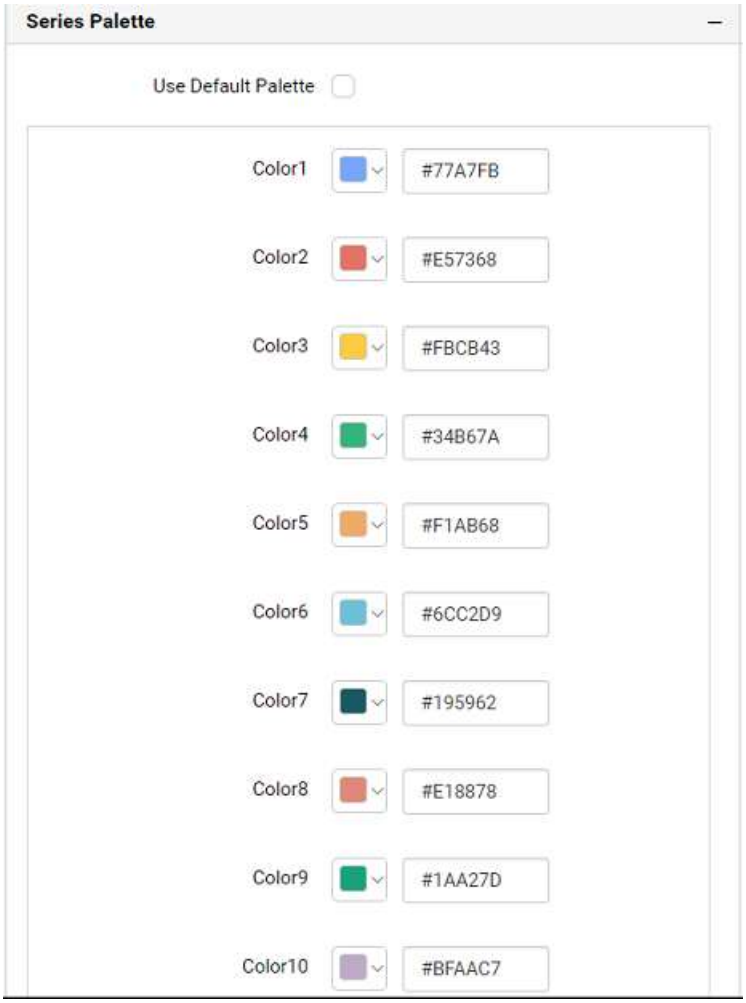
This allows you to enable the Category axis gridlines for the 100% stacked area chart.

Series Palette

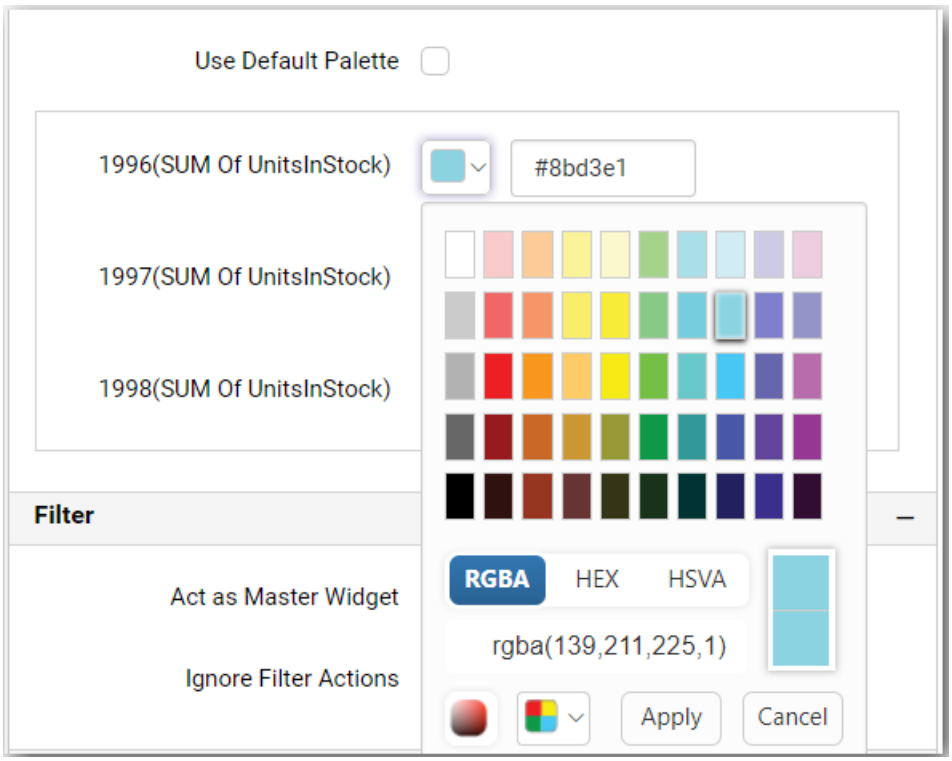
This allows you to customize the chart series color through Series Palette section.

Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.

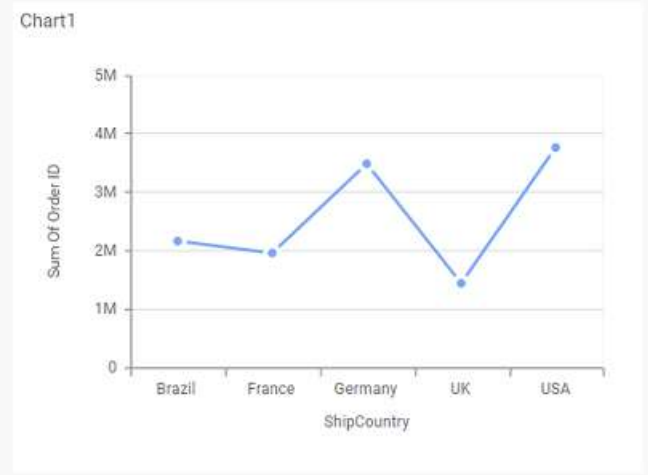


By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.



Line Chart

Line Chart allows you to showcase trends for analysis over a time period with data points connecting using straight lines.

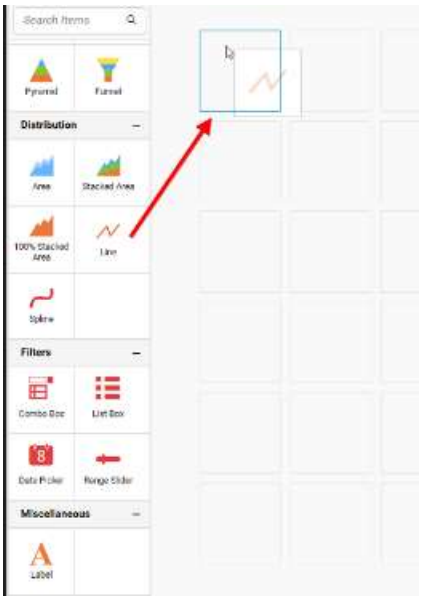


How to configure the table data to line chart?

Line Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

Follow the steps to configure data to Line chart

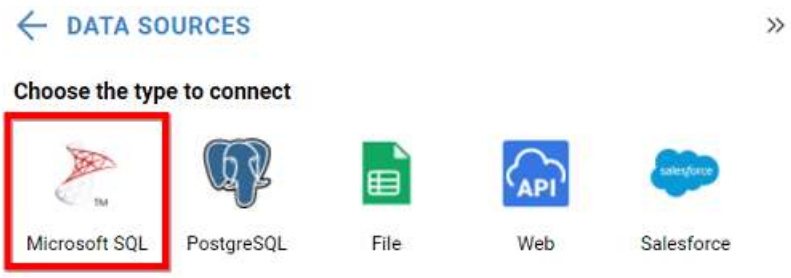
Drag and drop the control to canvas and resize it to your required size.



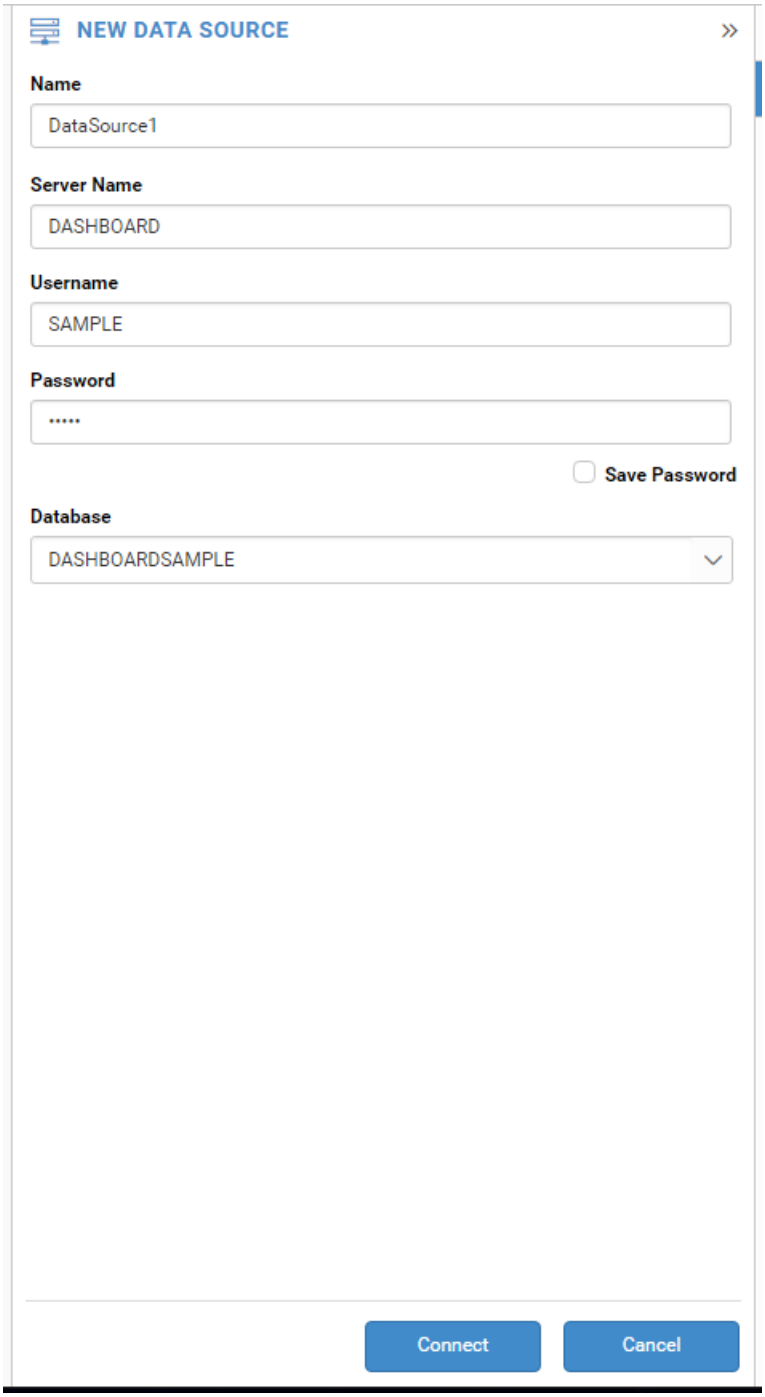
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

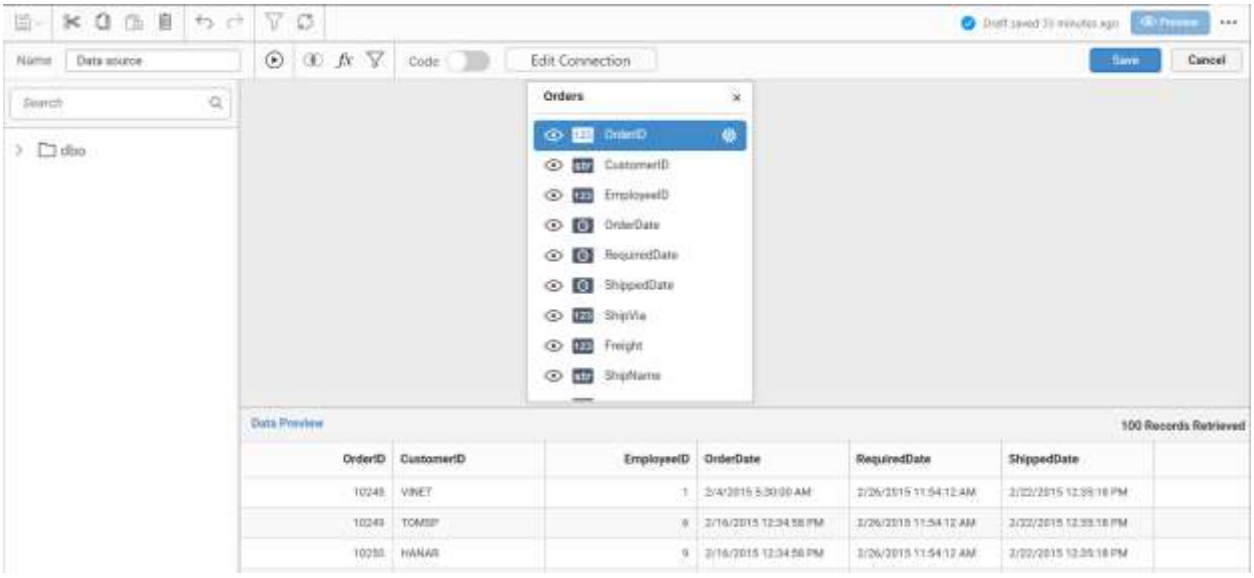
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



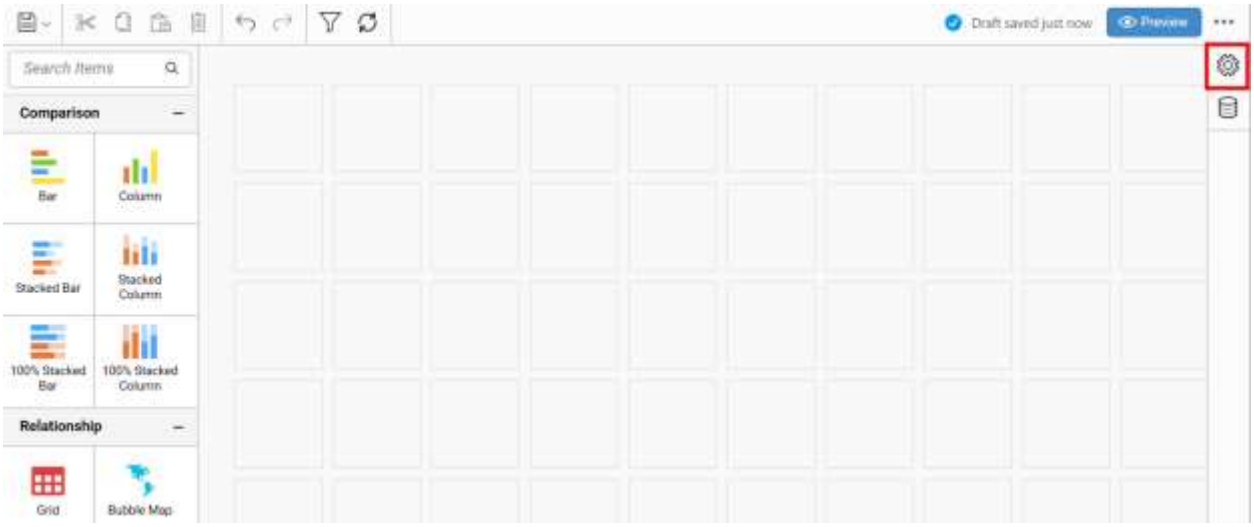
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

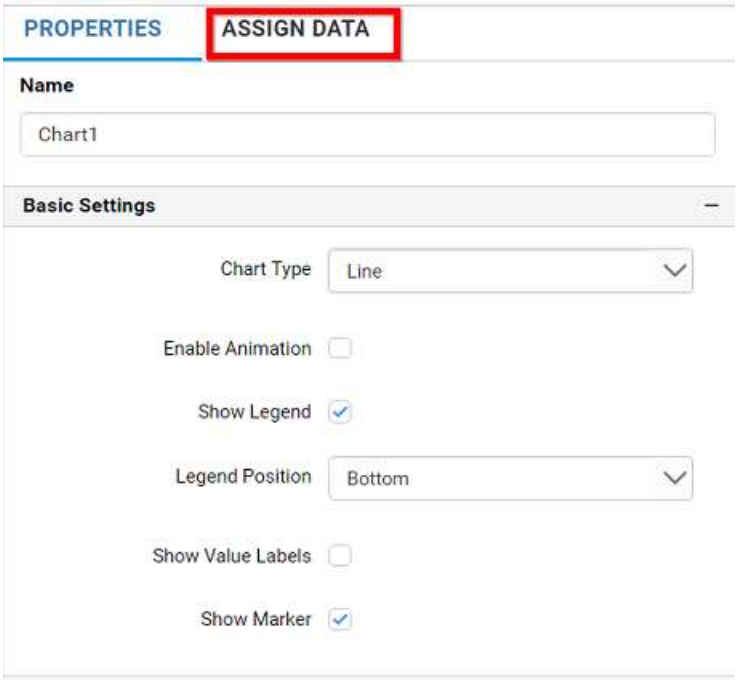


Drag your preferred table or view from the left pane from data design view, click Save button.

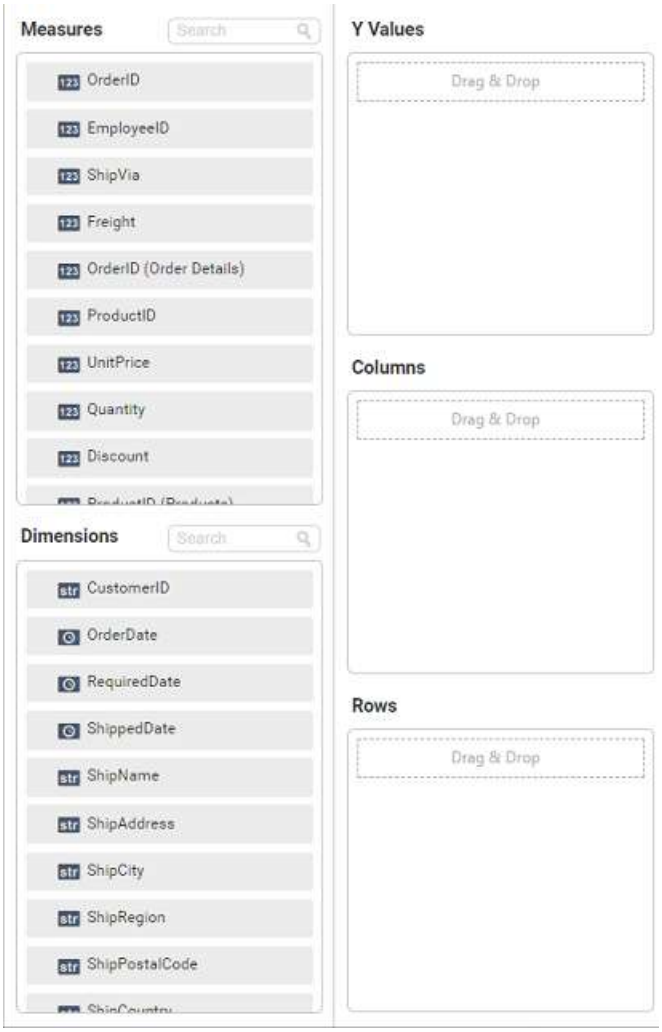


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.





The data tab will be opened with available measures and dimensions from the connected data source



You can add the required data from Measures and Dimensions into required field.

Adding Y Values

You can add more than one Measures into Y Values field by drag and drop the required measure.

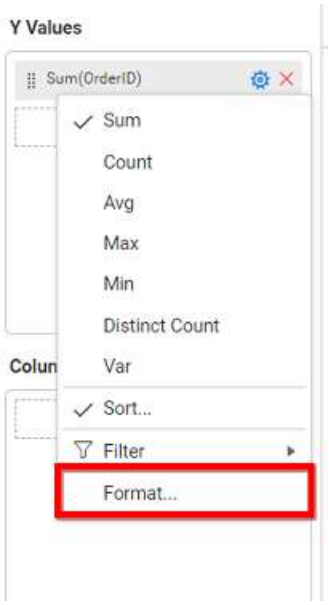
Now, the line chart will be rendered like this

Click the Settings option to change required summary type from the available summary types shown in Settings.

You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

You can filter the data to be displayed in funnel chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

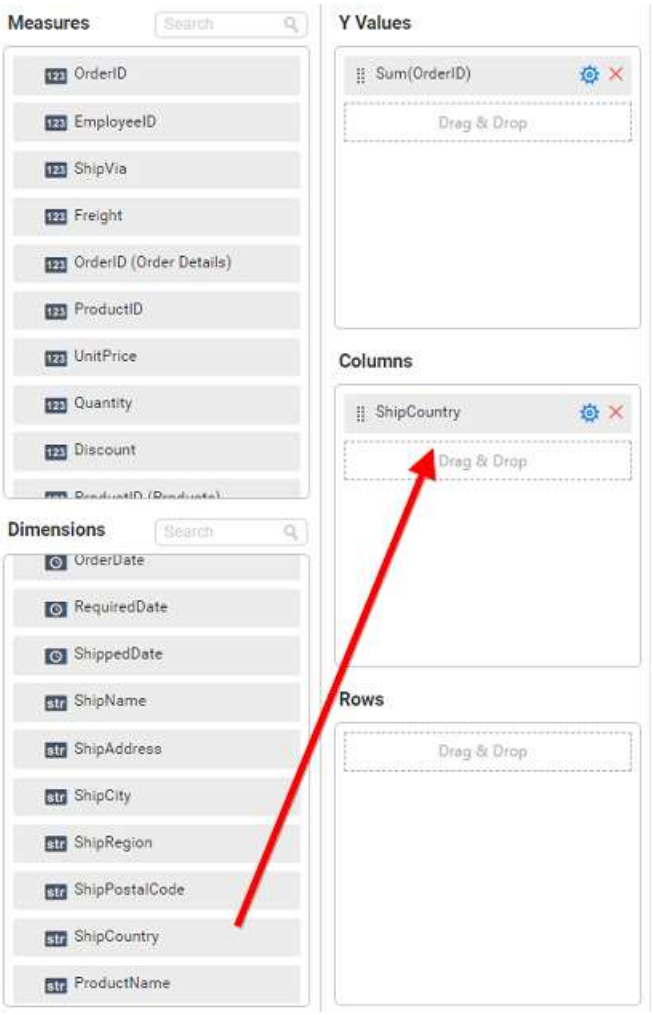


To remove the added value fields click highlighted button.

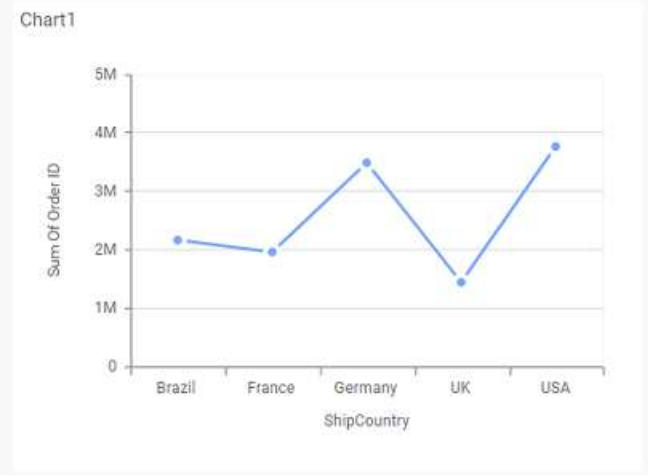
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



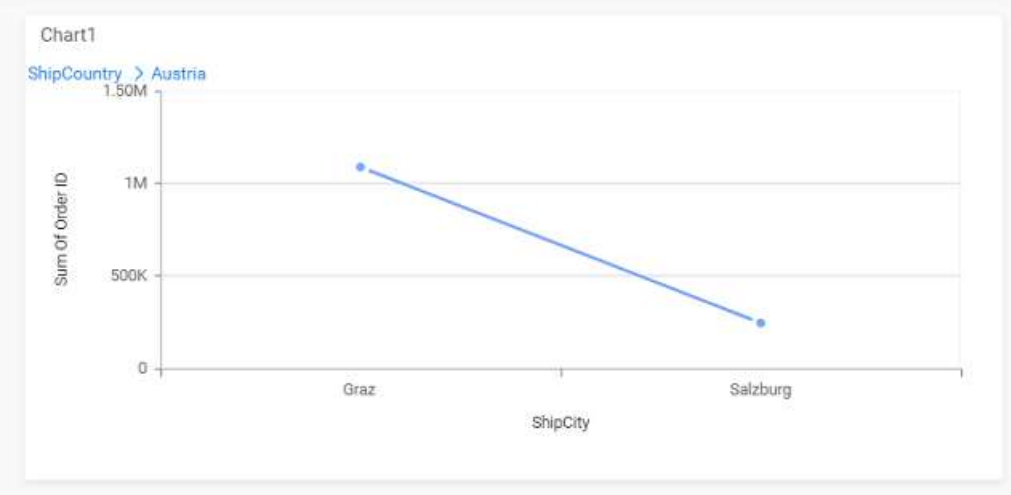
Line chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

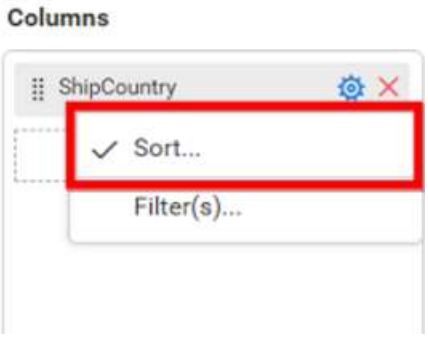
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.

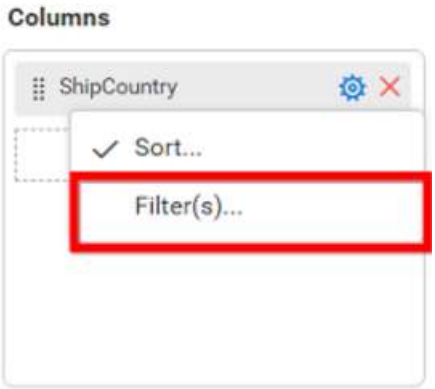


You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

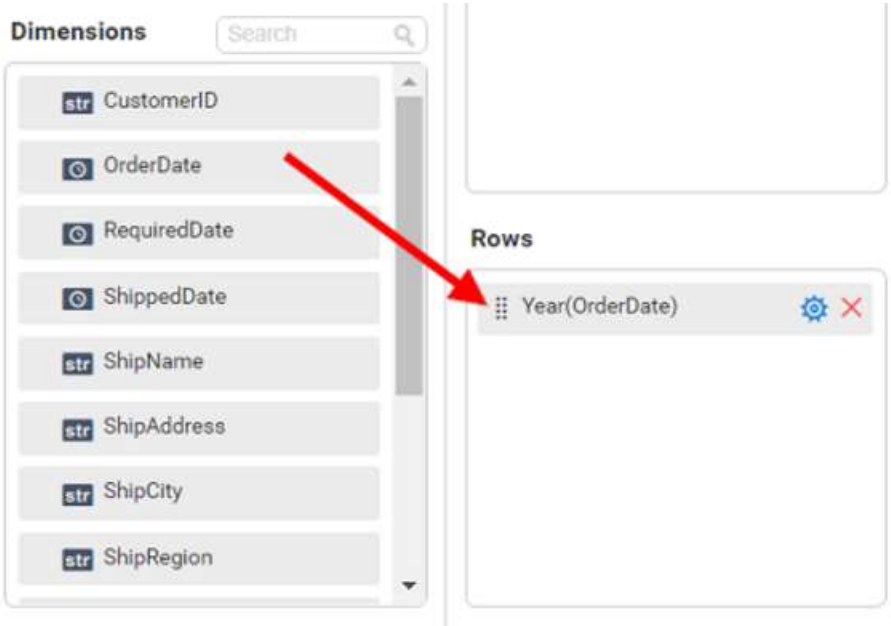


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

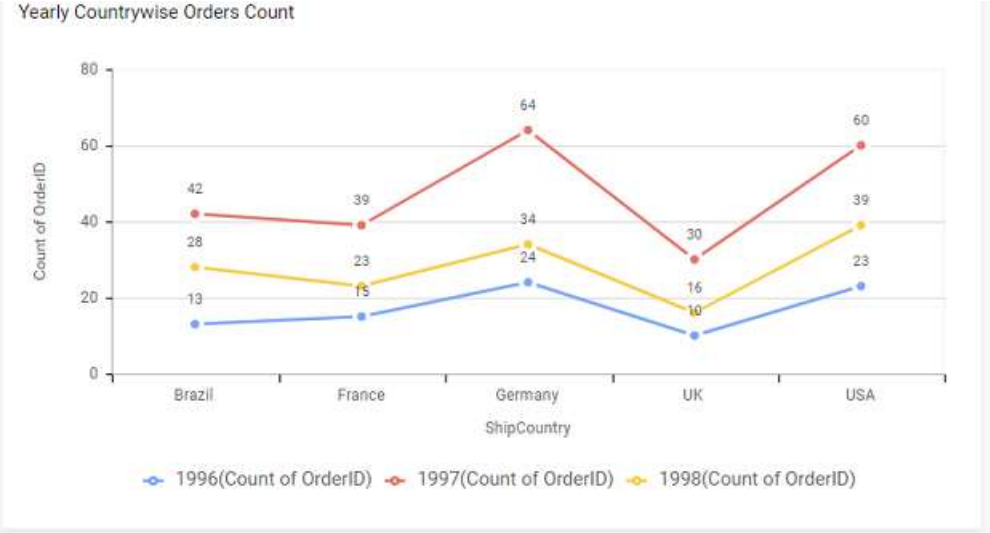
Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render line chart in series.

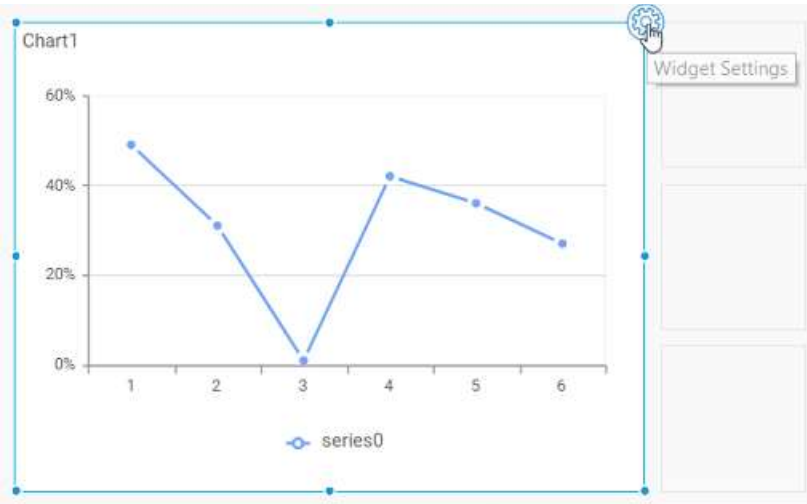


How to format line chart?

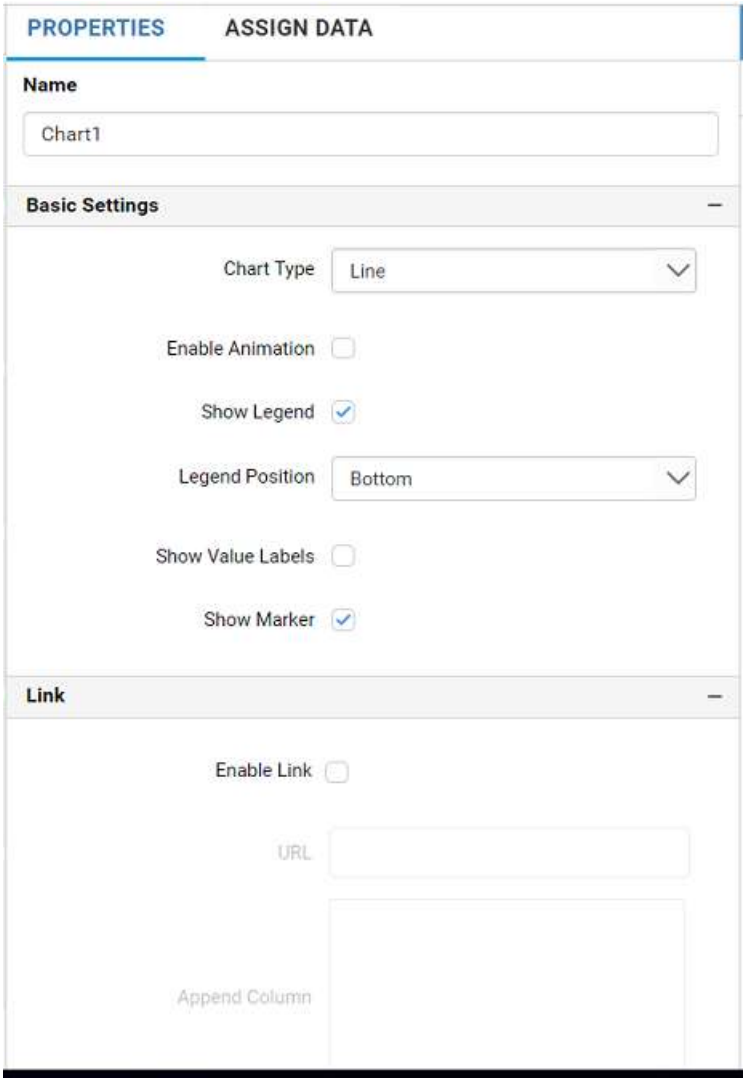
You can format the line chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into line chart follow the steps

1. Drag and drop the line chart into canvas and resize it to your required size.
2. Configure the data into line chart.
3. Focus on the line chart and click on widget settings.



The property window will be opened



General Settings

Name

This allows you to change the title for this line chart widget.

Basic Settings

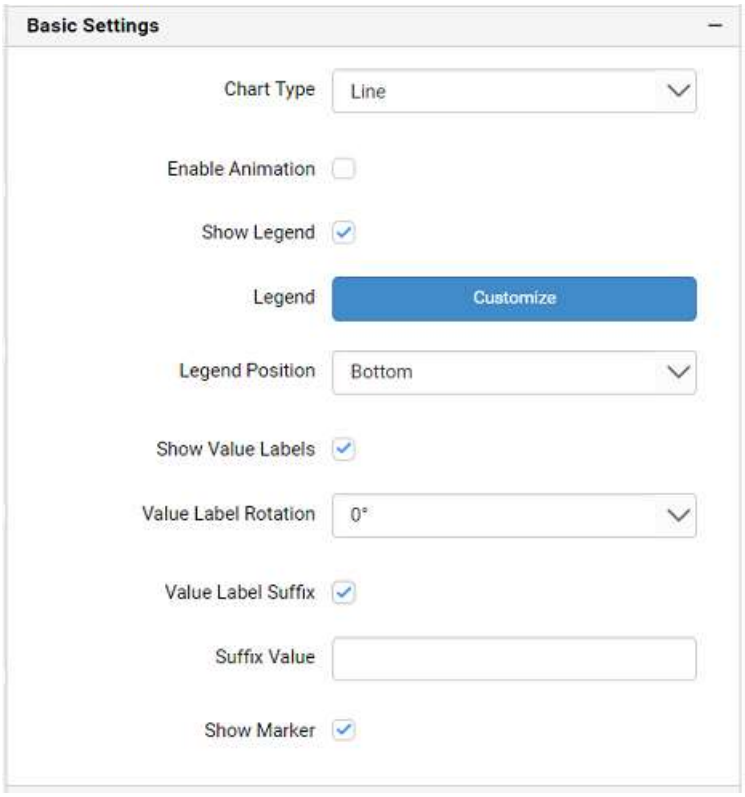


Chart Type

This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).

Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

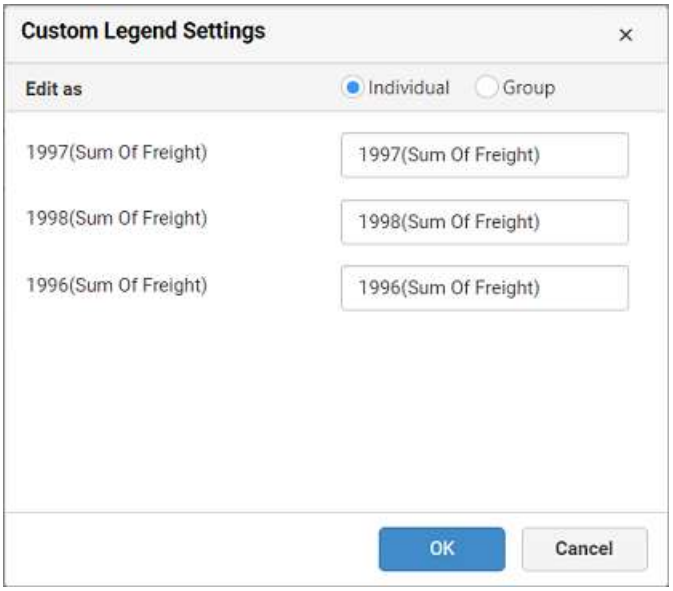
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

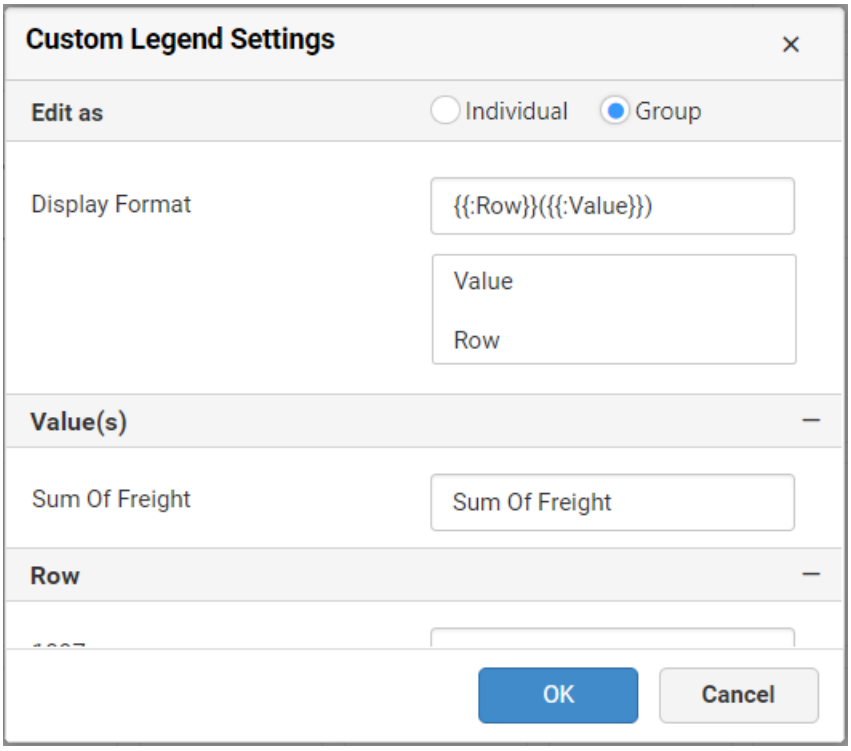
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.

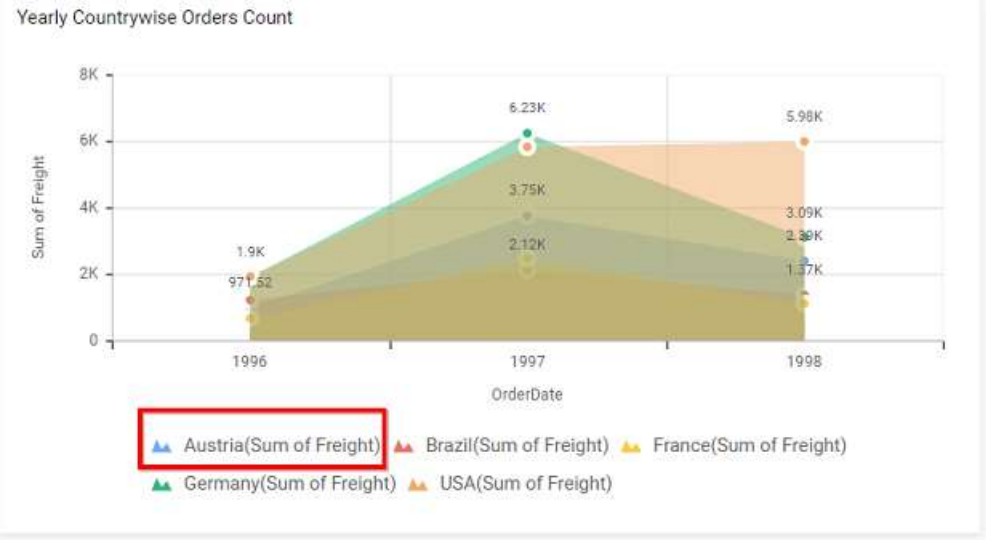


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Austria (Sum of Freight)

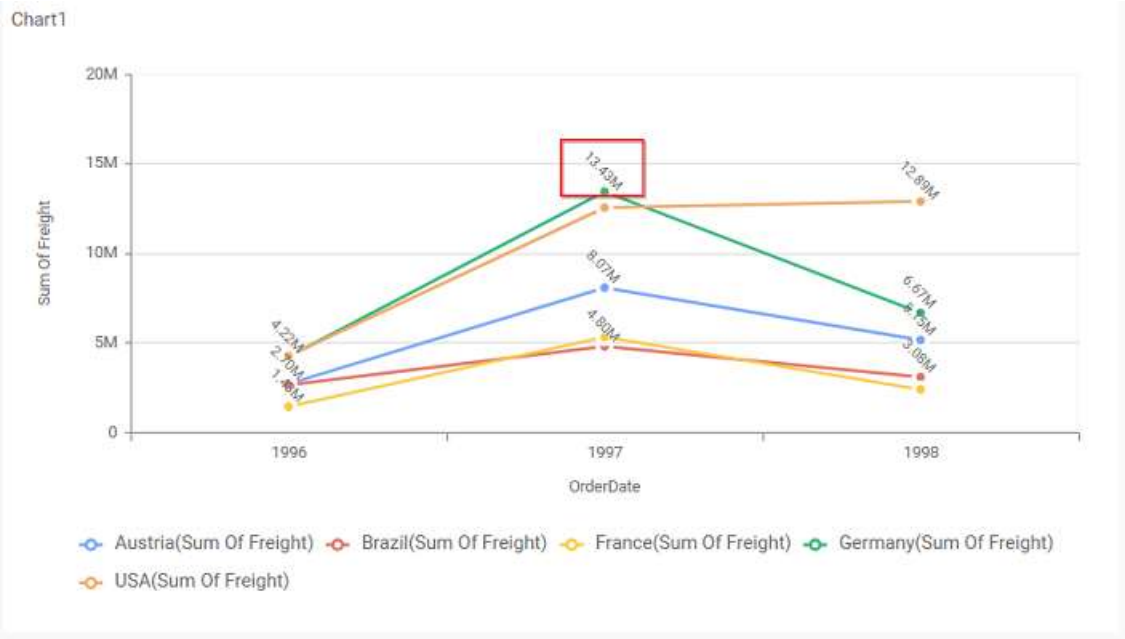


Show Value Labels

This allows you to toggle the visibility of value labels.

Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

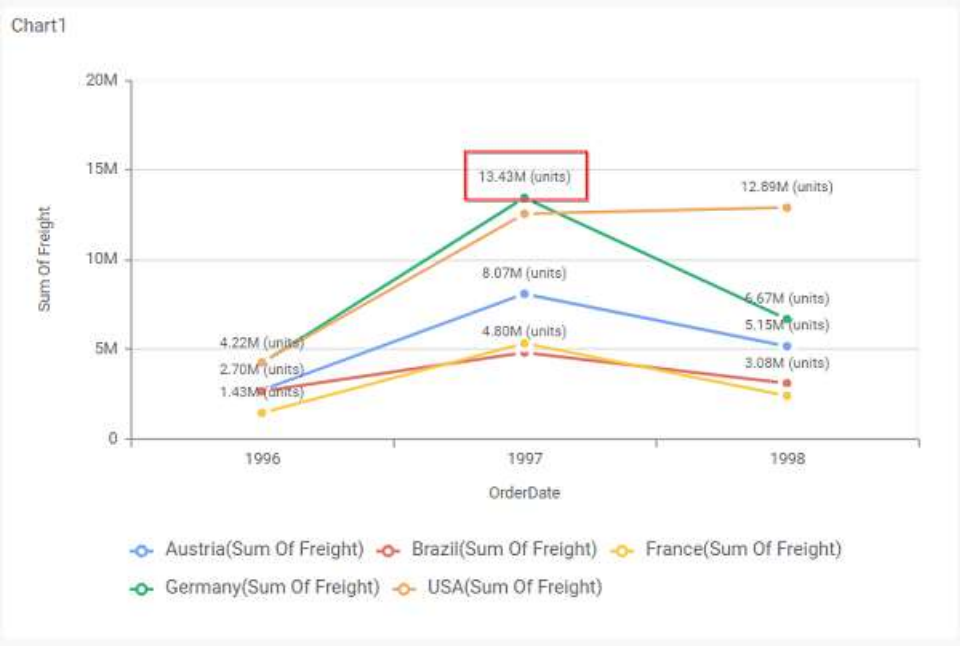


Value Label Suffix

Allows you to enable the Suffix value text to the value labels.

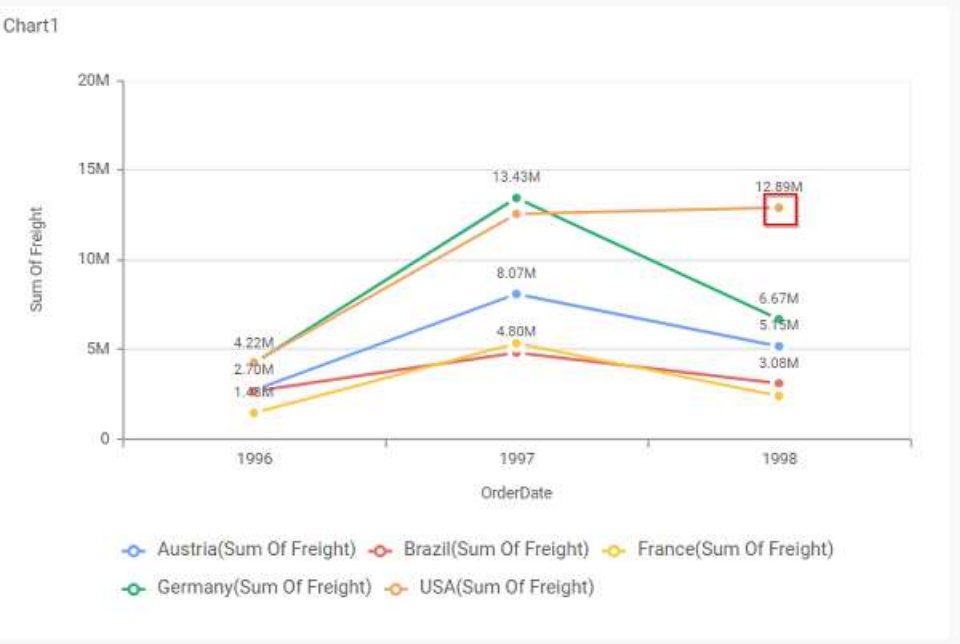
Suffix Value

Allows you to set\edit suffix value to the value labels.

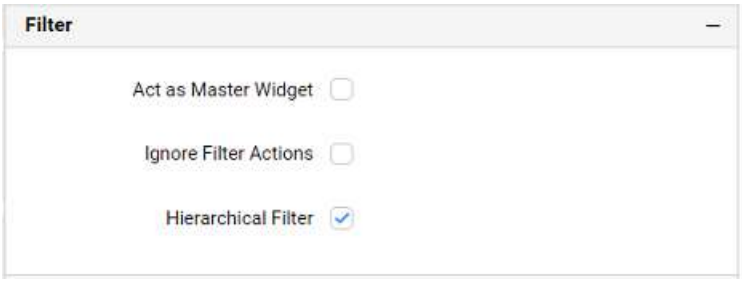


Show Marker

This allows you to toggle the visibility of marker from label to adorn each data point in chart series.



Filter



Act as Master Widget

This allows you to define this chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

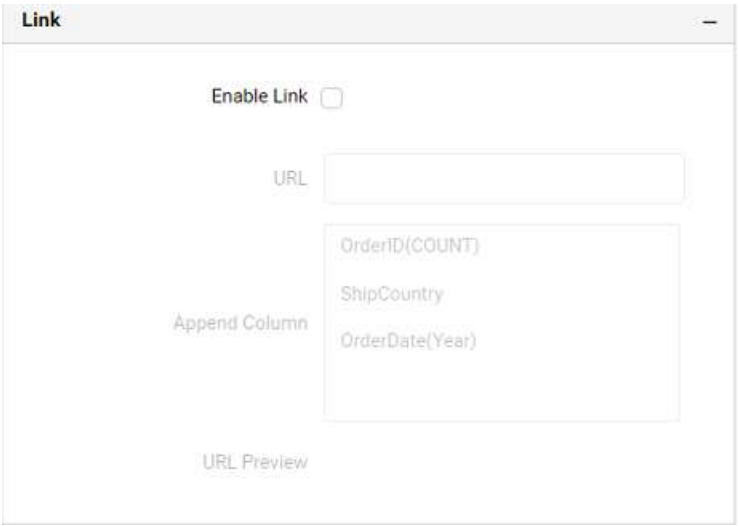
This allows you to define this chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

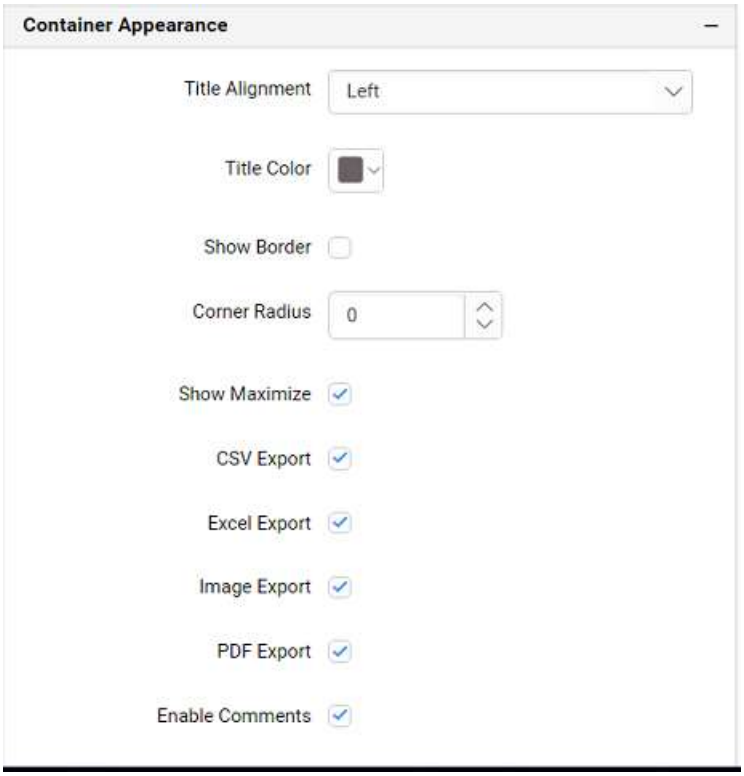
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this line chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this line chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this line chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer.

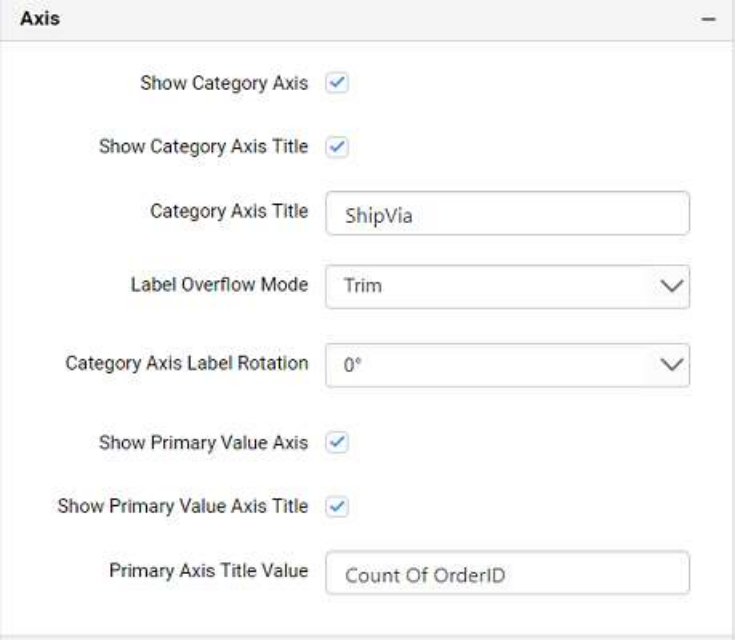
Image Export

This allows you to enable/disable the image export option for this line chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

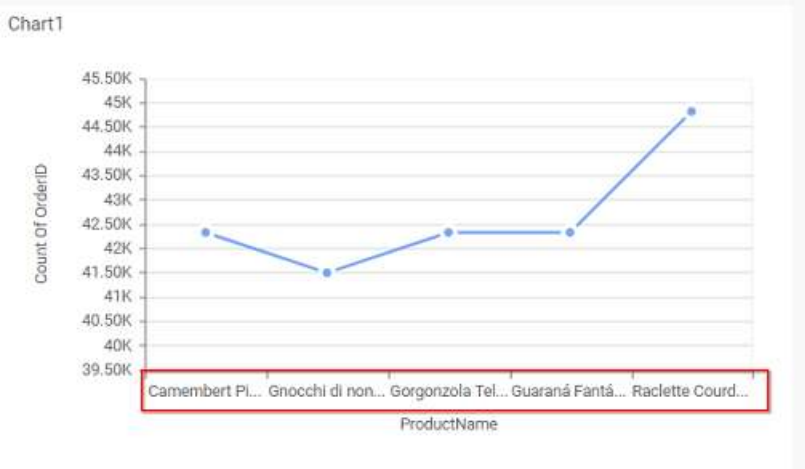
Axis Settings



This section allows you to customize the axis settings in chart.

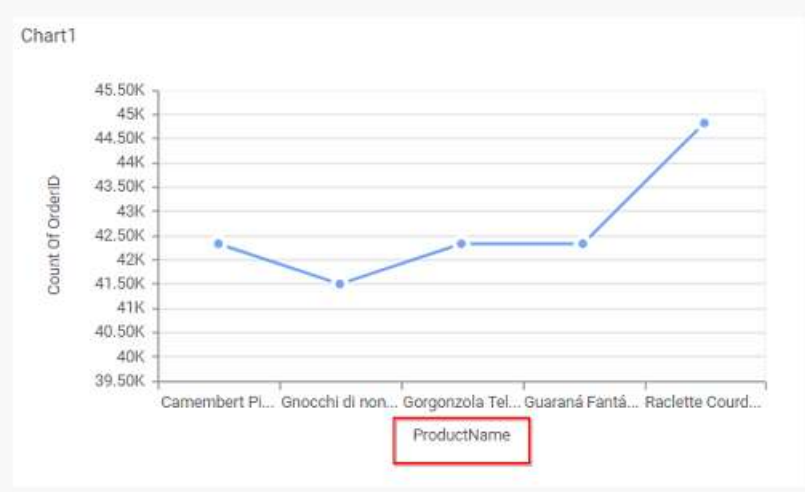
Show Category Axis

This allows to enable the visibility of Category Axis.



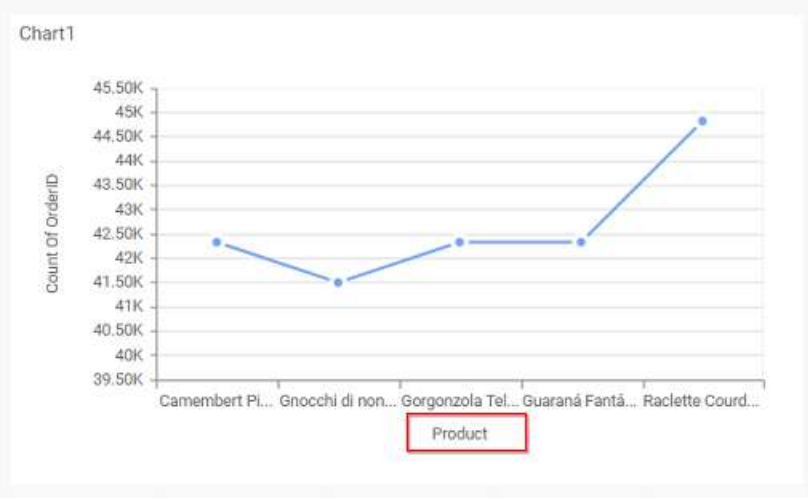
Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.



Label overflow mode

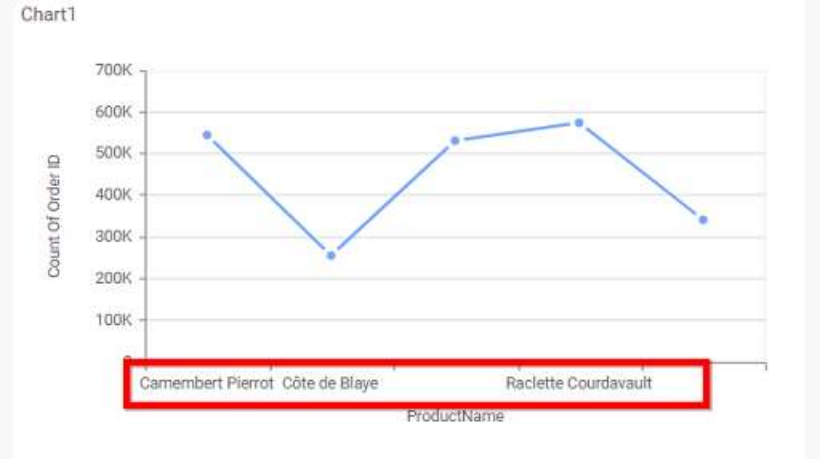
This allows you to handle the display mode of the overlapping labels in the Category Axis.

Trim

This option trims the end of overlapping label in the axis.

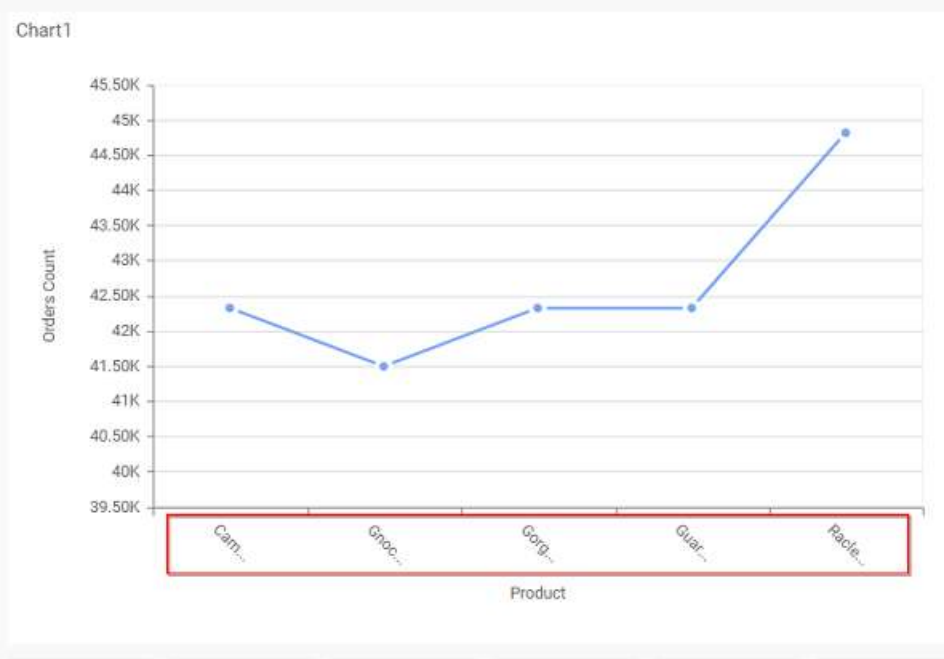
Hide

This option hides the overlapping label in the axis.



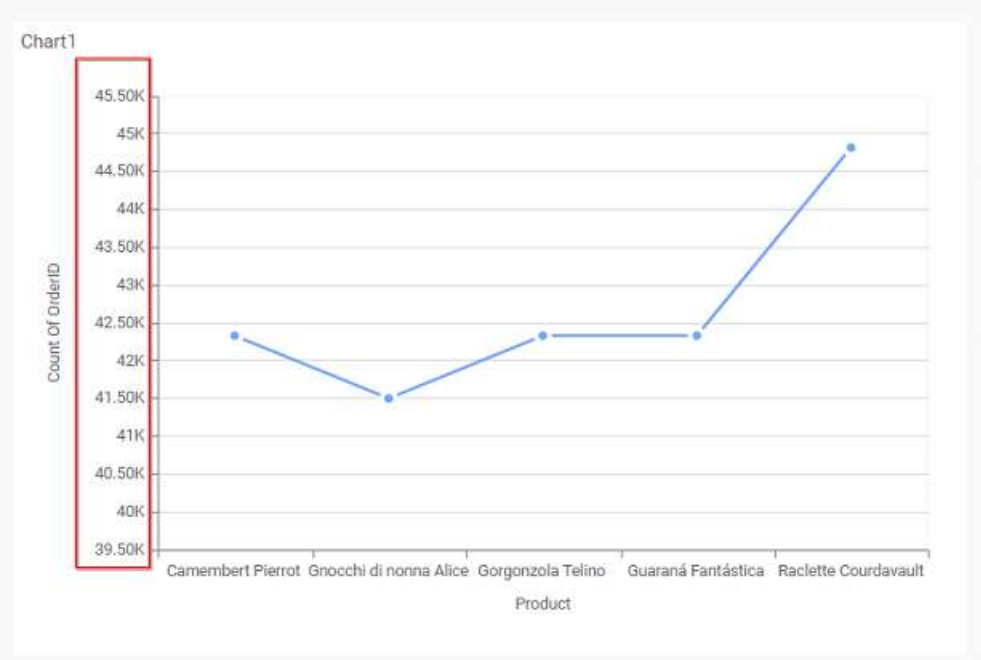
Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



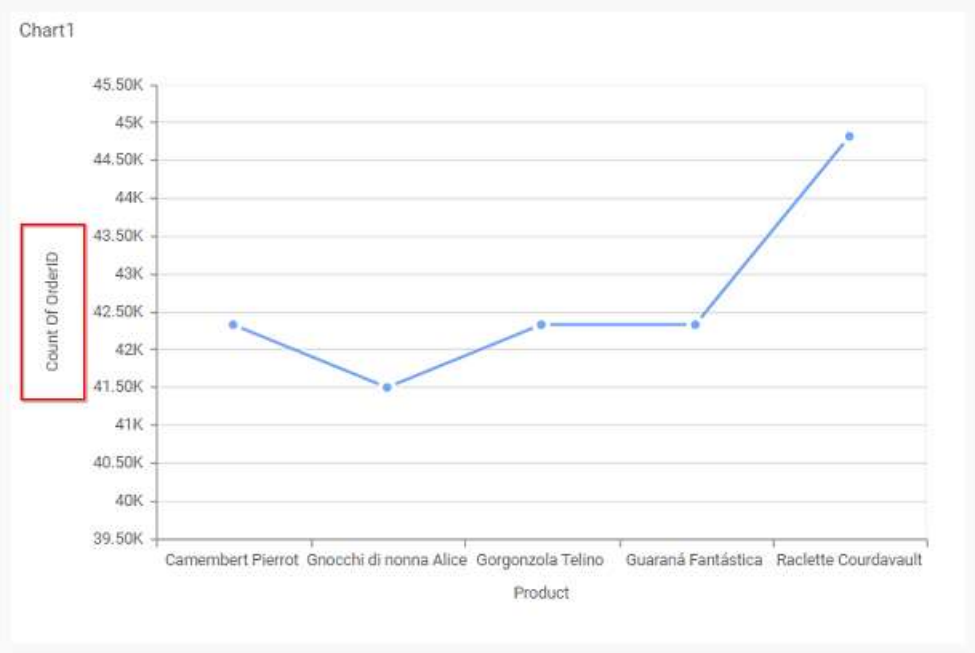
Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.



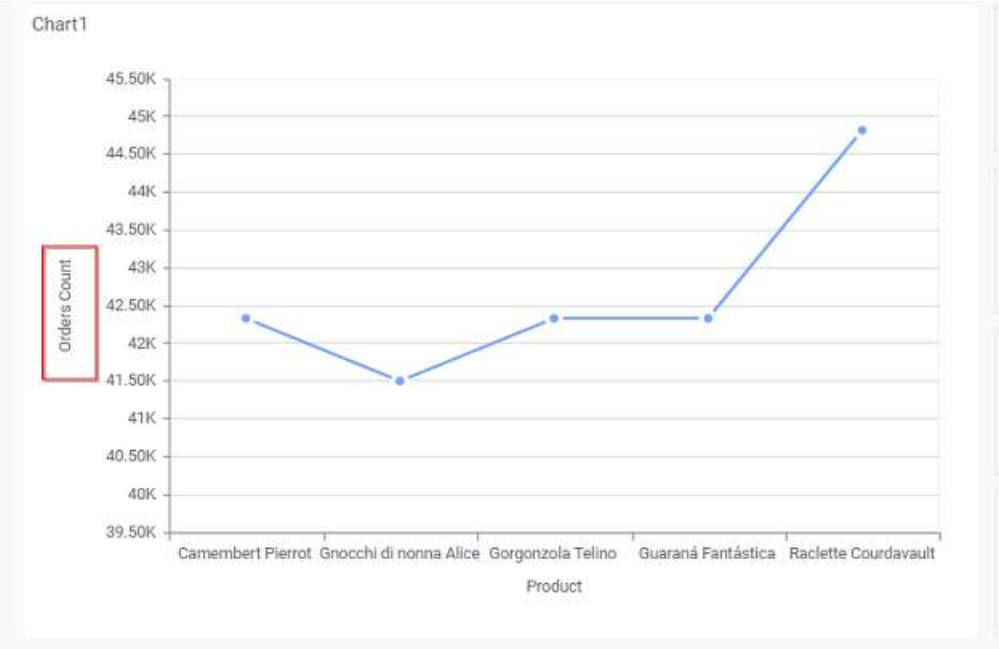
Show Primary Value Axis Title

This allows you to enable the visibility of Primary Value Axis title of chart.

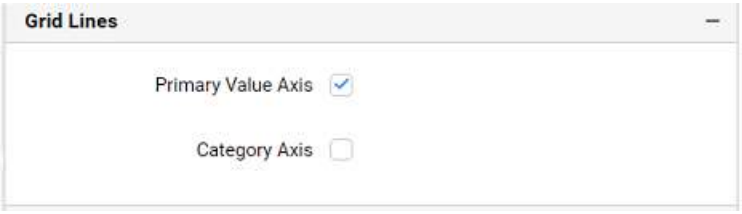


Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

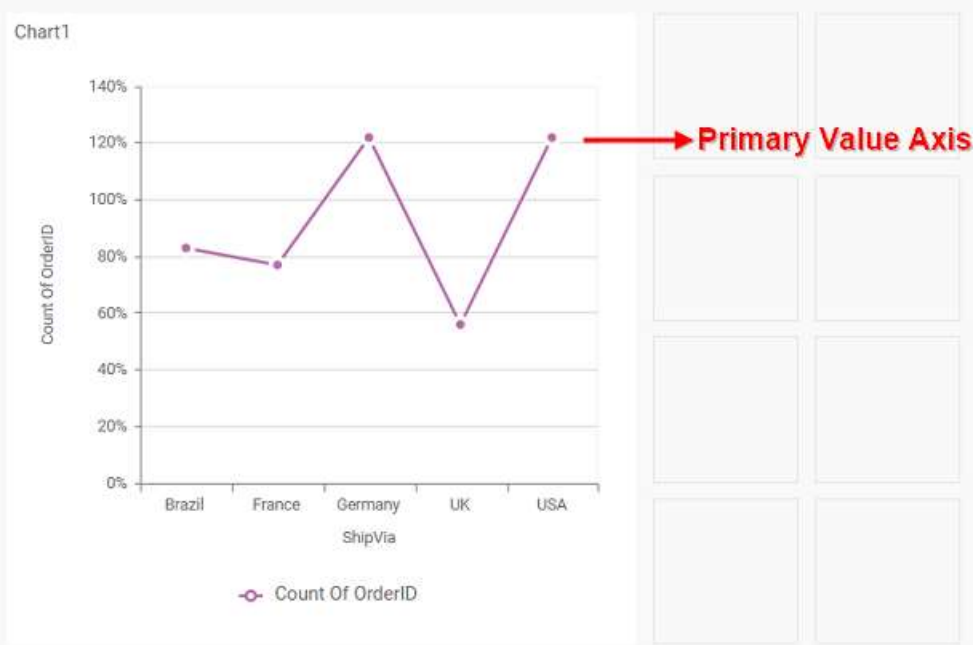


Grid Line



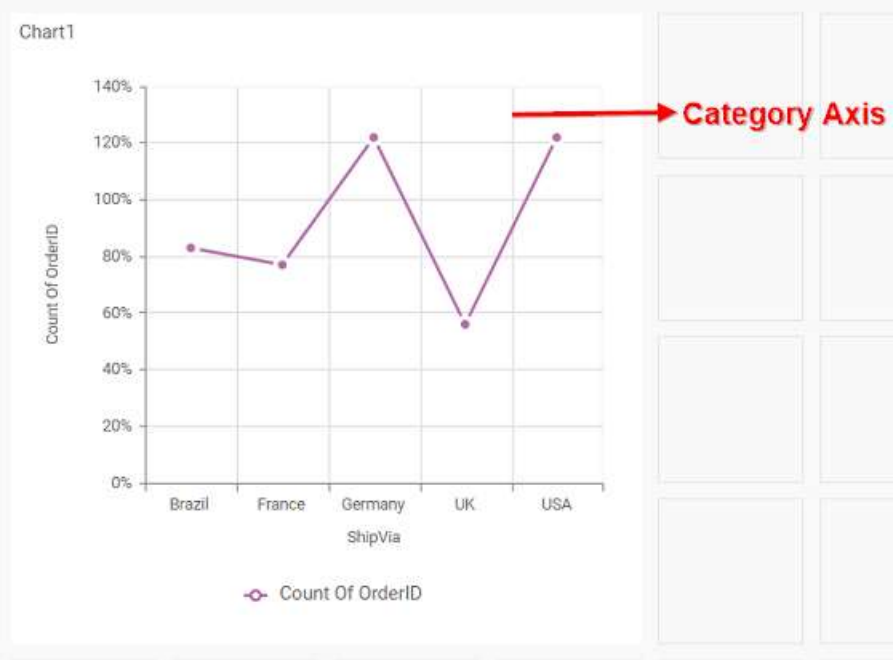
Primary Value Axis

This allows you to enable the Primary Value Axis gridlines for the line chart.



Category Axis

This allows you to enable the Category Axis gridlines for the line chart.

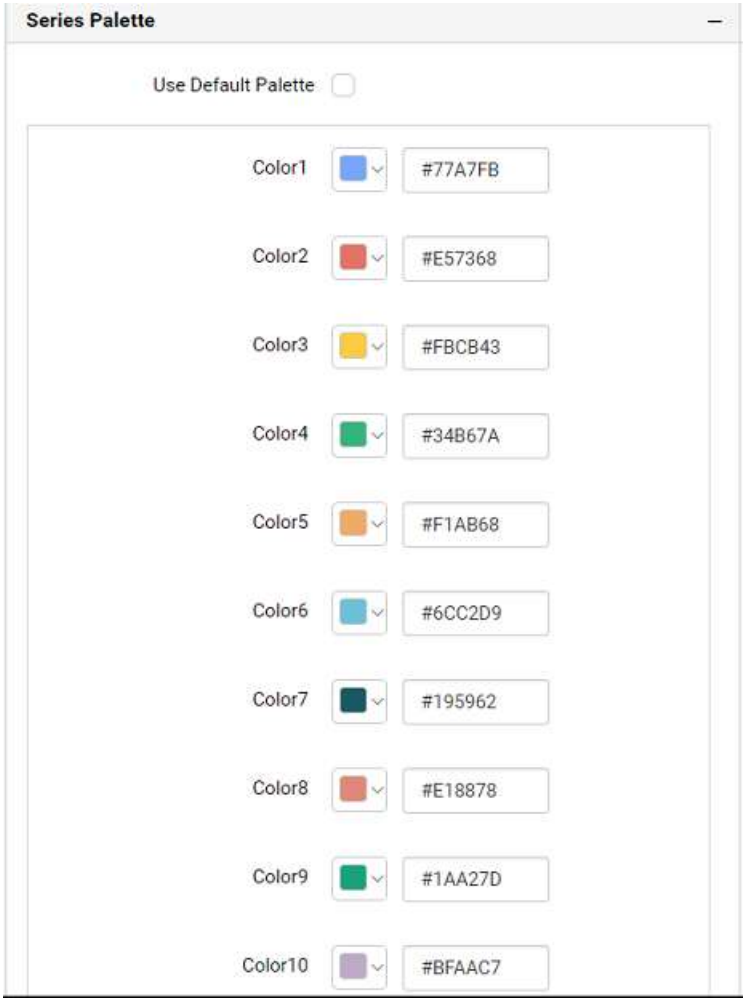


Series Palette

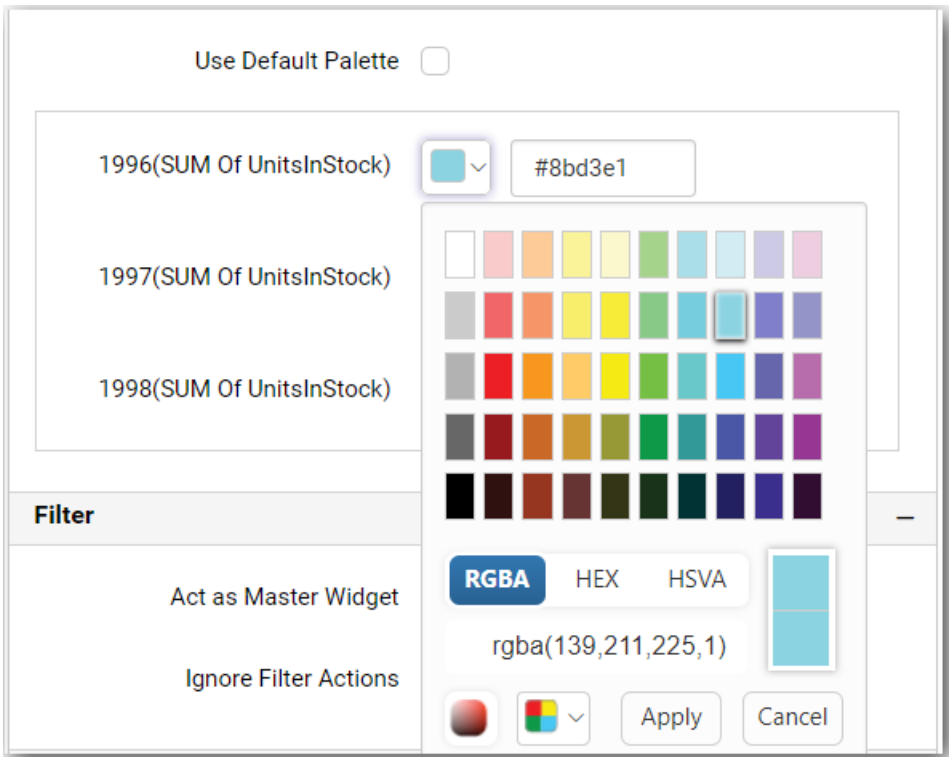
This allows you to customize the chart series color through Series Palette section.

Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.

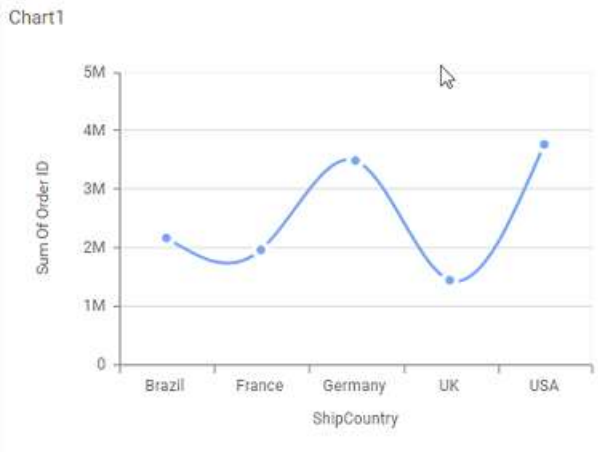


By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.



Spline Chart

Spline Chart allows you to showcase trends for analysis over a time period with data points connected using splines.

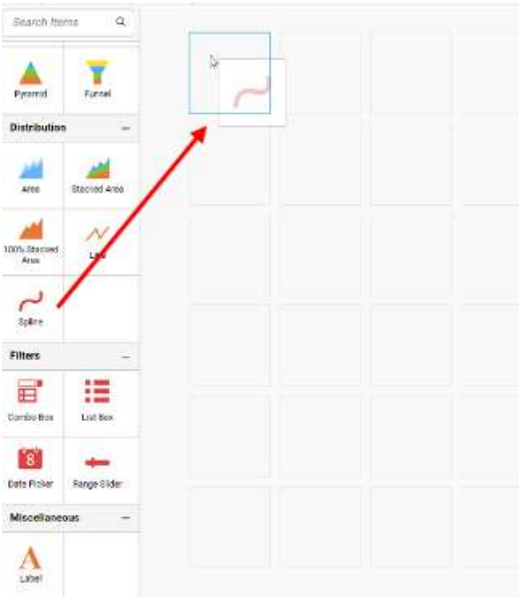


How to configure the table data to spline Chart?

Spline Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Y Values block. The dimension that you would like to categorize the measure, can be dropped onto Columns block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Rows block in addition.

Follow the steps to configure data to spline charts

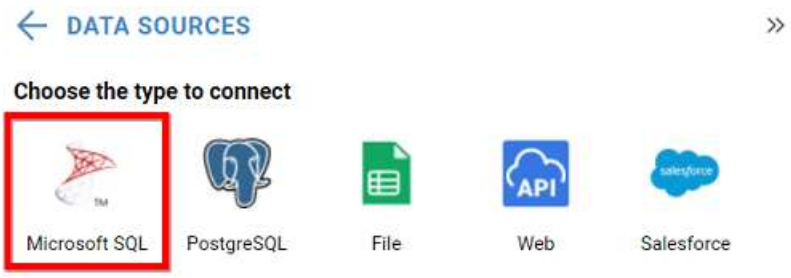
Drag and drop the spline chart into canvas and resize it into required size.



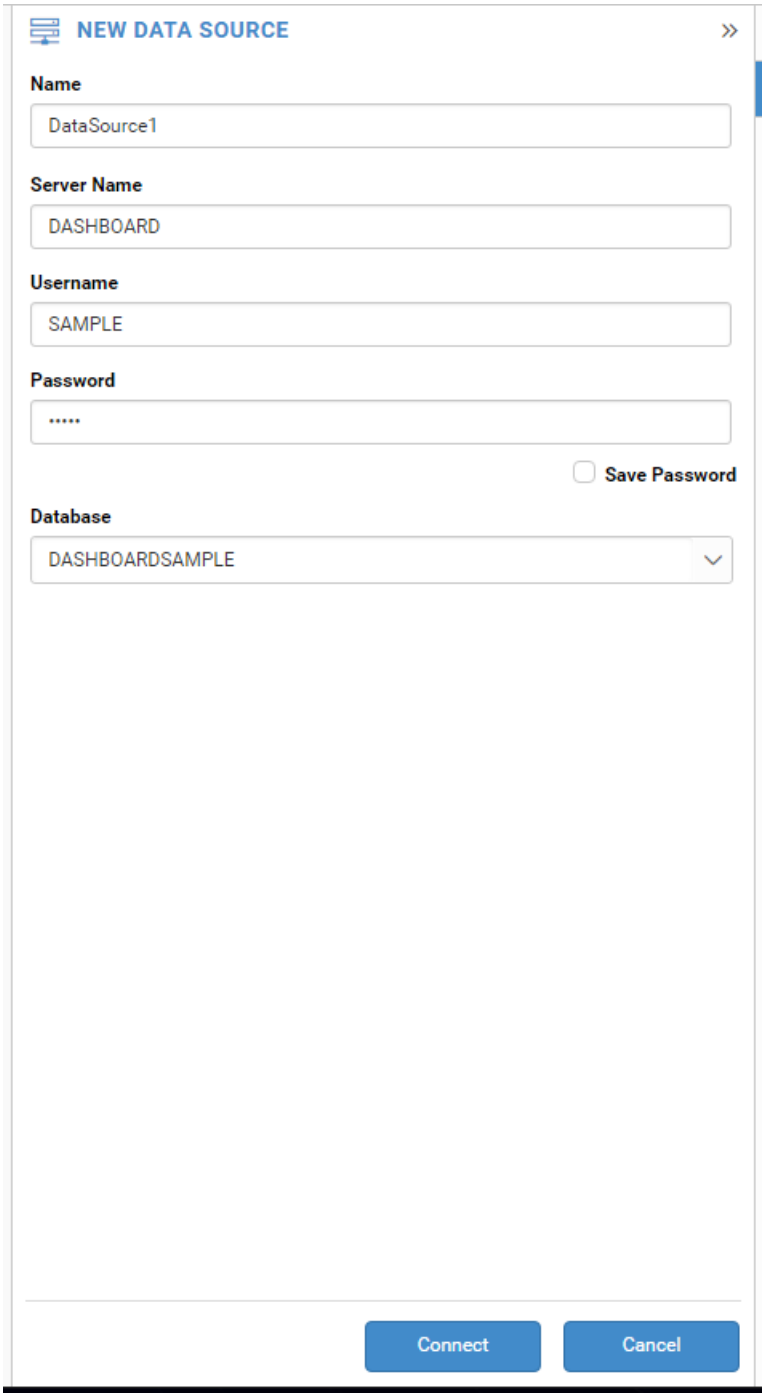
Click Data button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

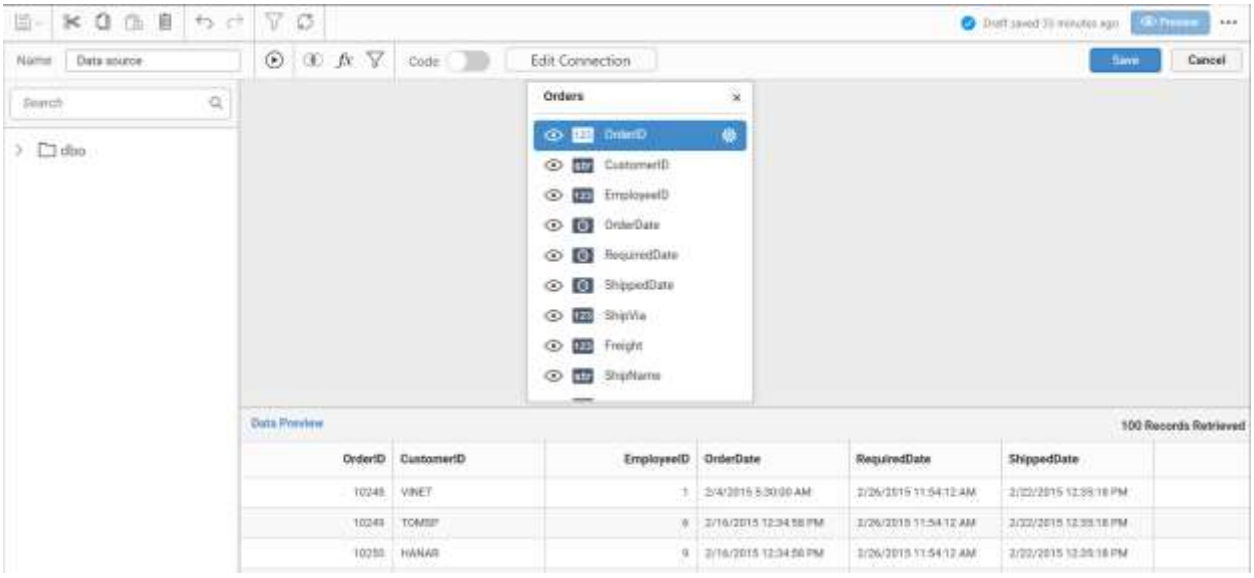
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



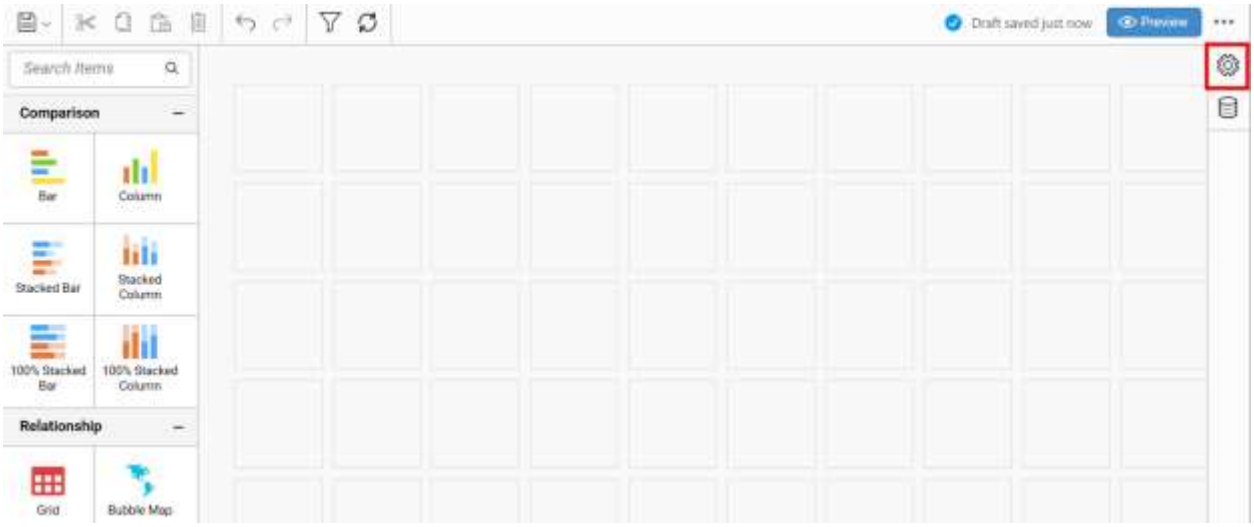
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

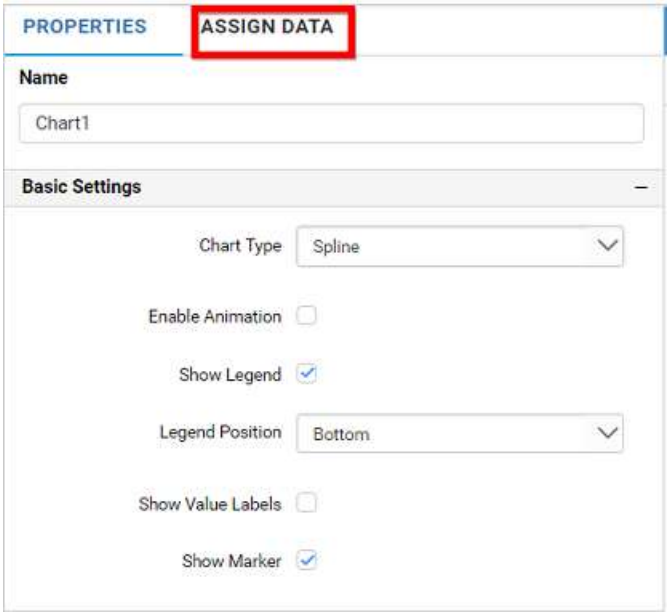


Drag your preferred table or view from the left pane from data design view, click Save button.

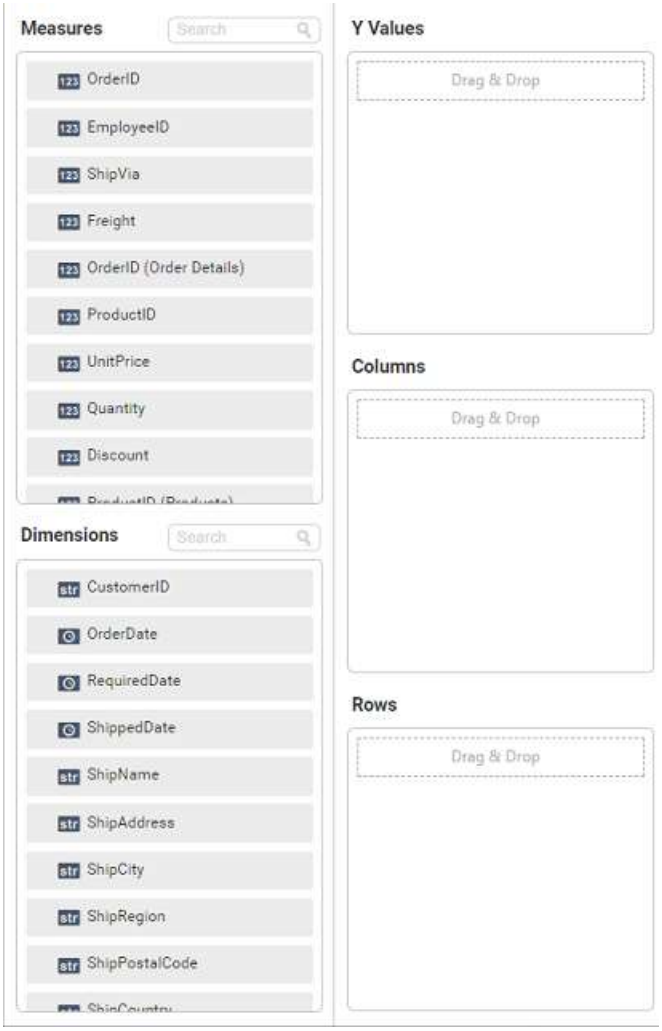


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.





The data tab will be opened with available measures and dimensions from the connected data source



You can add the required data from Measures and Dimensions into required field.

Adding Y Values

You can add more than one Measures into Y Values field by drag and drop the required measure.

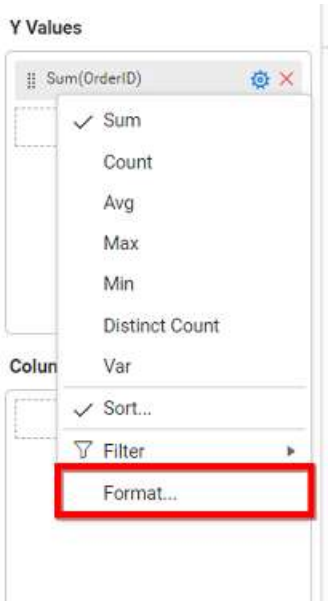
Now the spline chart will be rendered like this

Click the Settings option to change required summary type from the available summary types shown in Settings.

You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

You can filter the data to be displayed in funnel chart by using filter. For more details, refer filter.

You can format the data to be displayed in the chart by using format option. For more details, refer measure format

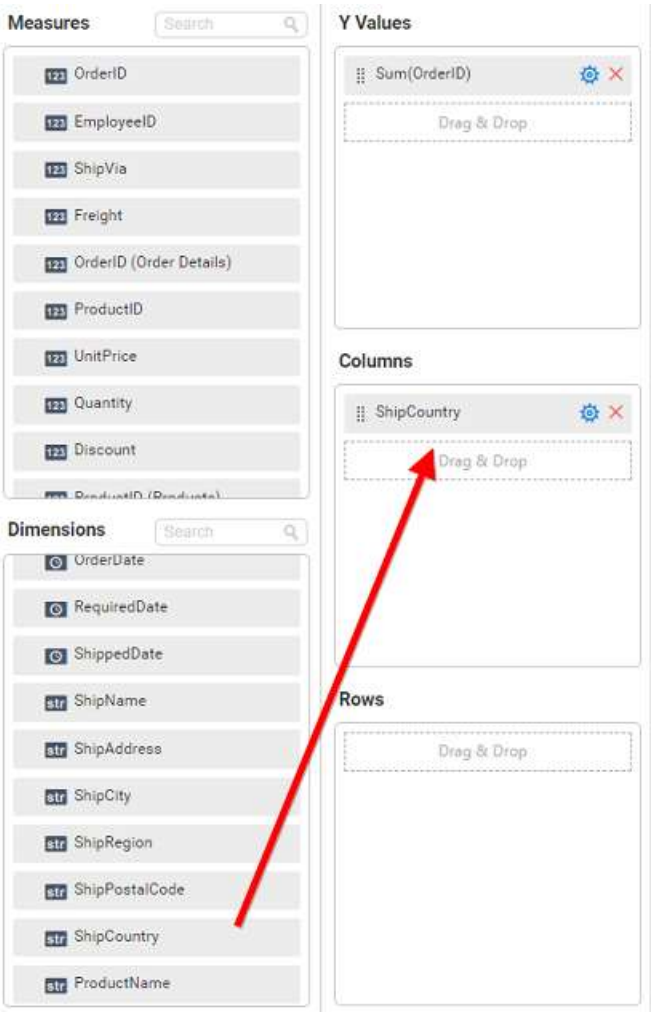


To remove the added value fields click highlighted button.

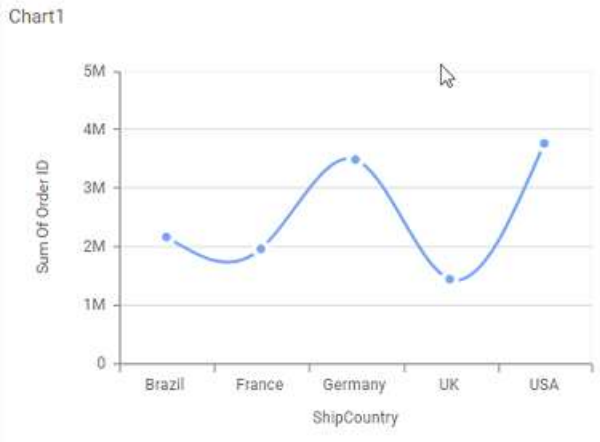
You can add more than one column from Dimensions field into Y Values field.

Adding Columns

You can add more than one value into Columns field.



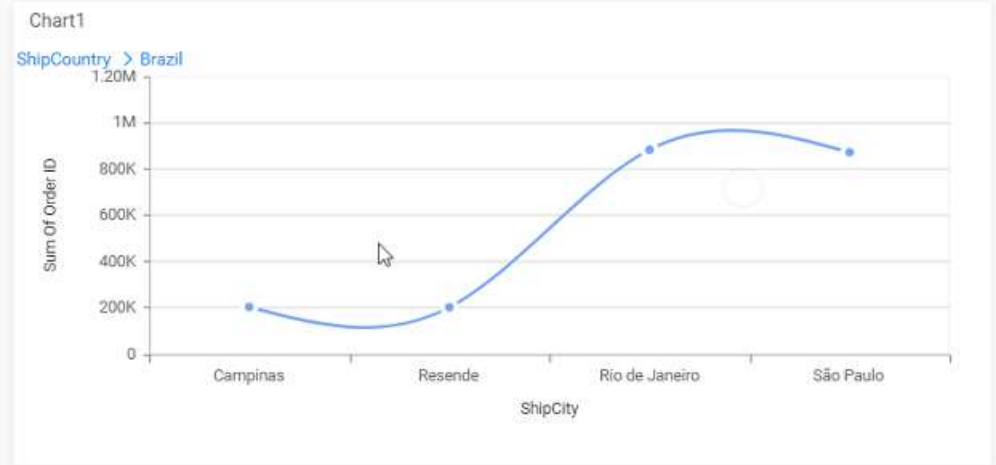
Spline chart will be rendered like this



Add more than one value to Columns field, an alert message will be shown. Click Yes to enable the option.

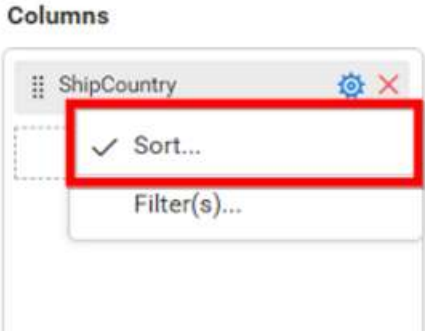
Note: If you click No, single value will be added to the Columns field.

The drilled view of the chart region selected.

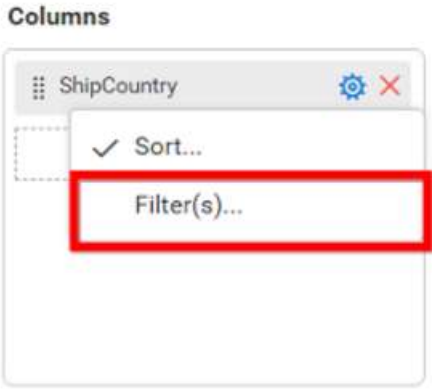


You can change the Settings.

You can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



You can apply filters by selecting filters option in settings. For more details, refer filter.

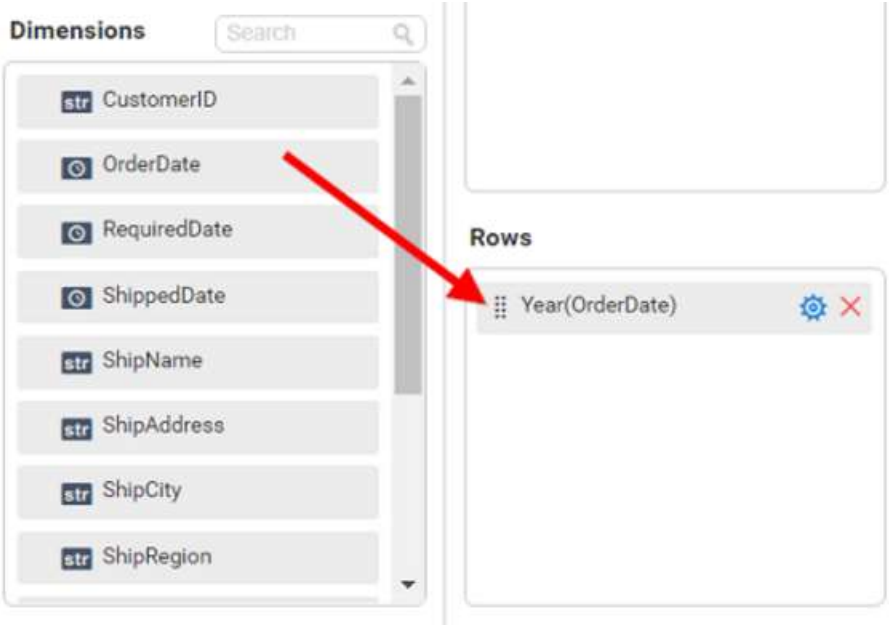


To show all records click on Show All Records.

Similarly you can add the Measures and Expression Columns into column field.

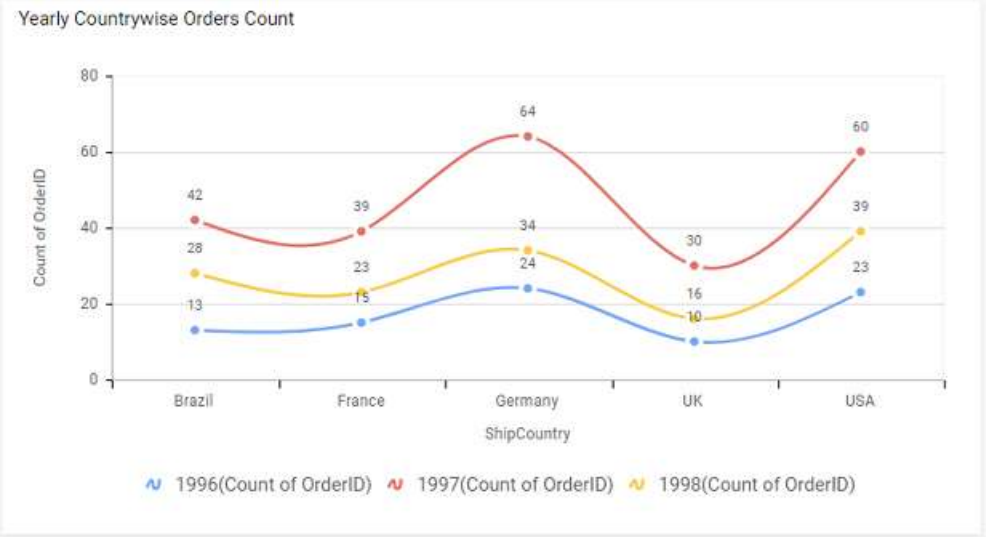
Adding Rows

You can drag and drop the Dimension into the Rows field.



You can apply filter and sort option for the rows field, if required.

This will render spline chart in series.

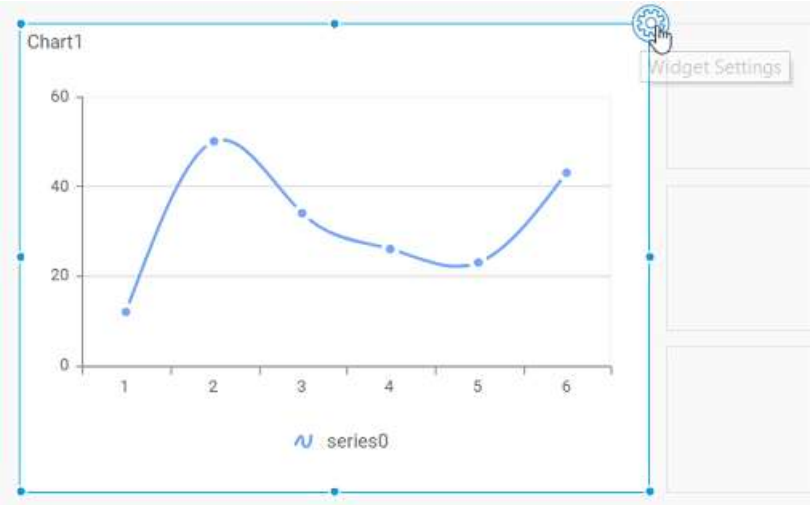


How to format spline Chart?

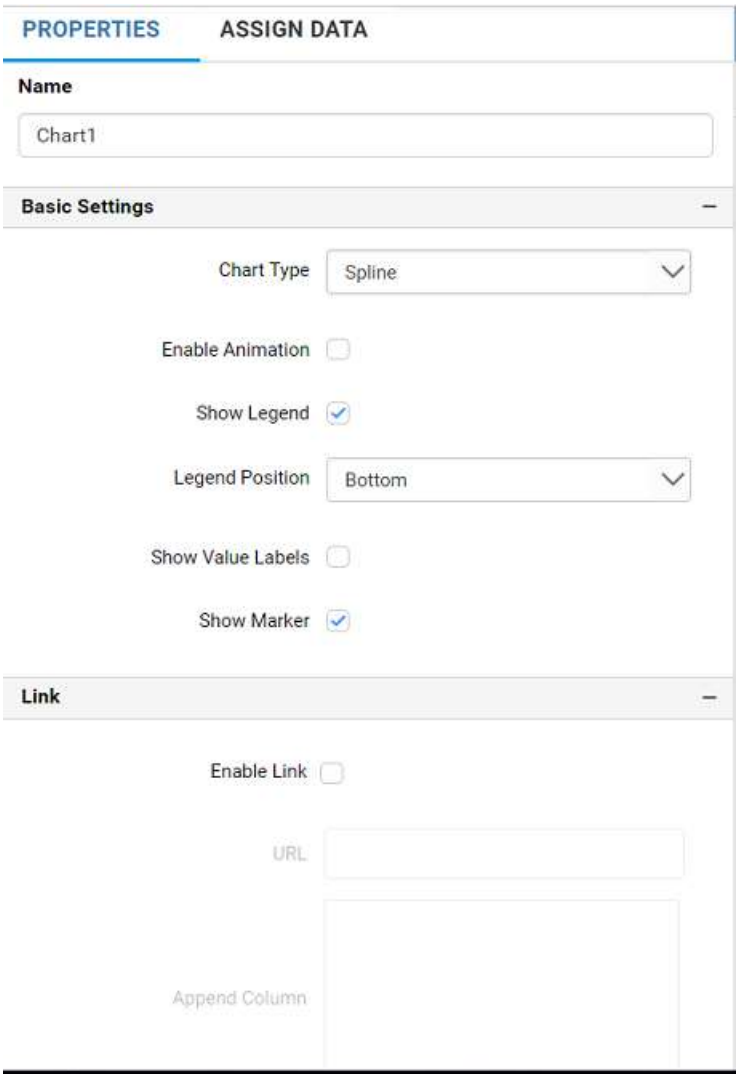
You can format the spline chart for better illustration of the view that you require, through the settings available in Properties tab.

To configure data into spline chart follow the steps

1. Drag and drop the spline chart into canvas and resize it to your required size.
2. Configure the data into spline chart.
3. Focus on the spline chart and click on widget settings.



The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings

Name

This allows you to change the title for this spline chart widget.

Basic Settings

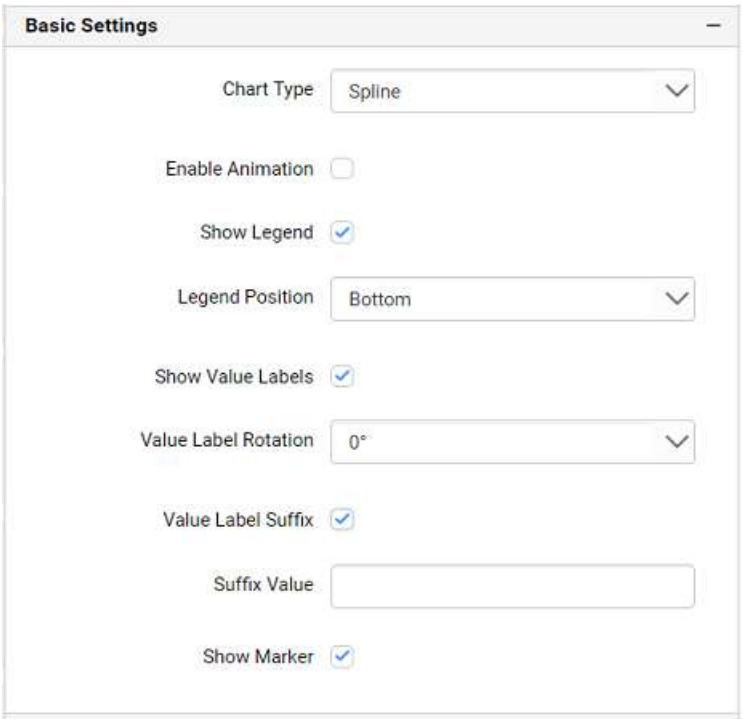


Chart Type

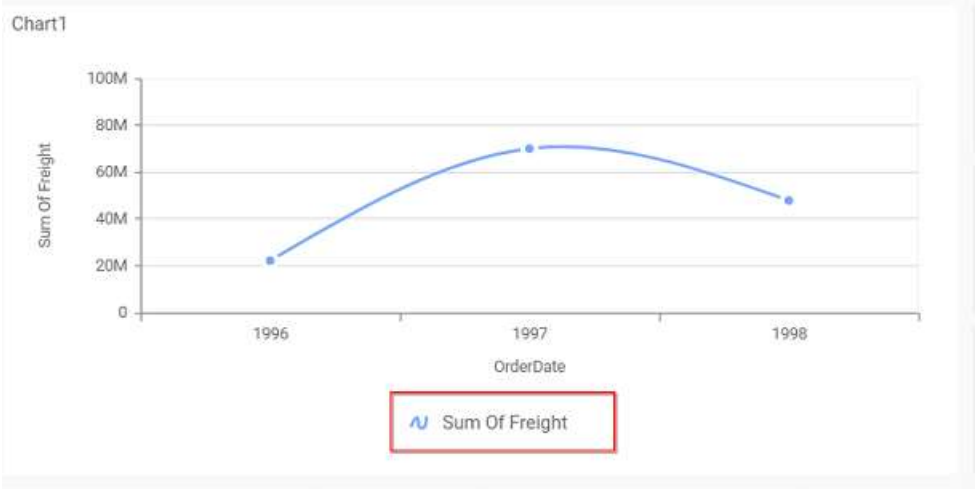
This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

A Legend is a text used to describe the data plotted. This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

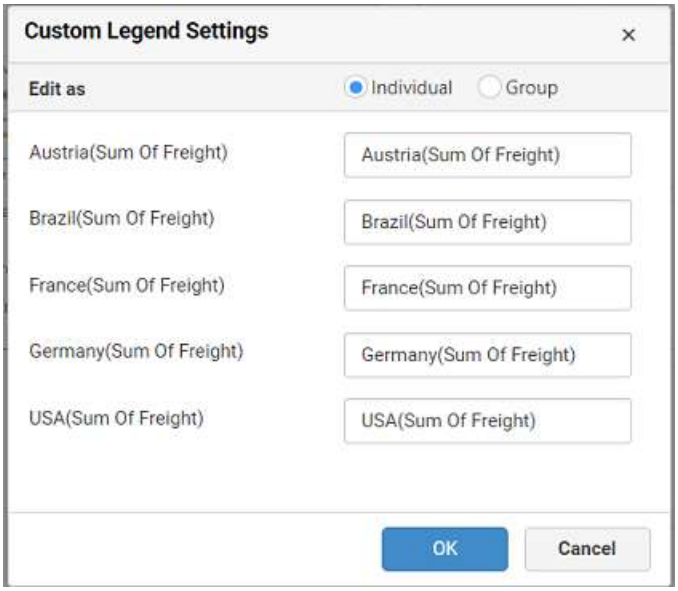
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

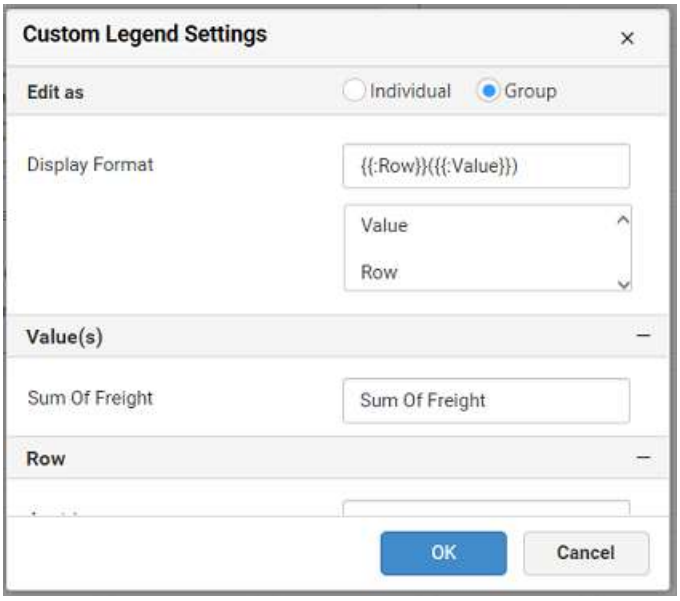
{{"{{"}} : Row {{}}}} ({{"{{"}} : Y Value {{}}}})

Where, Row represents the value of dimension column added to Rows section and Value represents the value of the measure column added to Y Values section.

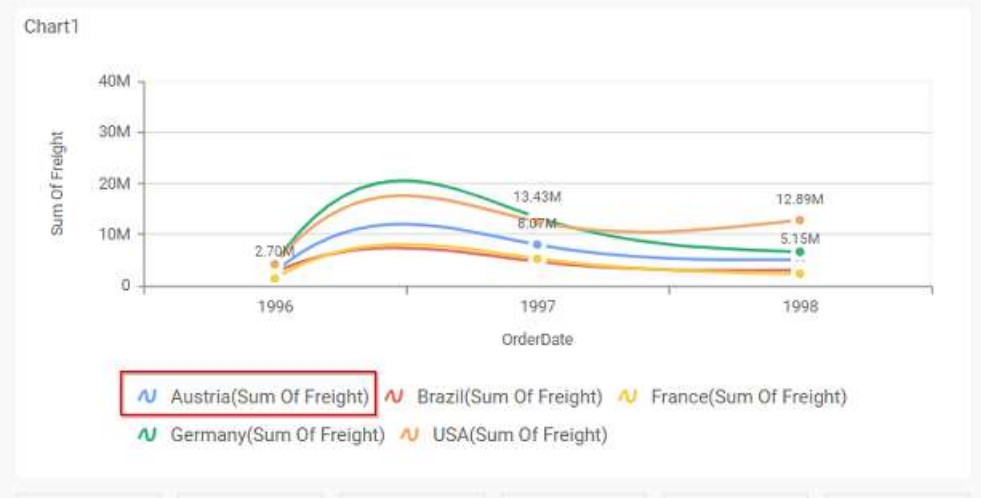


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

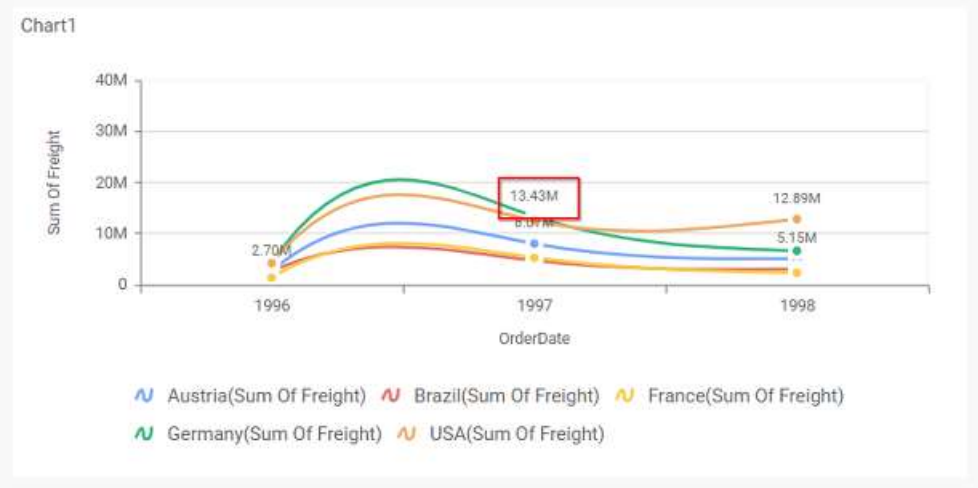


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Austria (Sum of Freight)



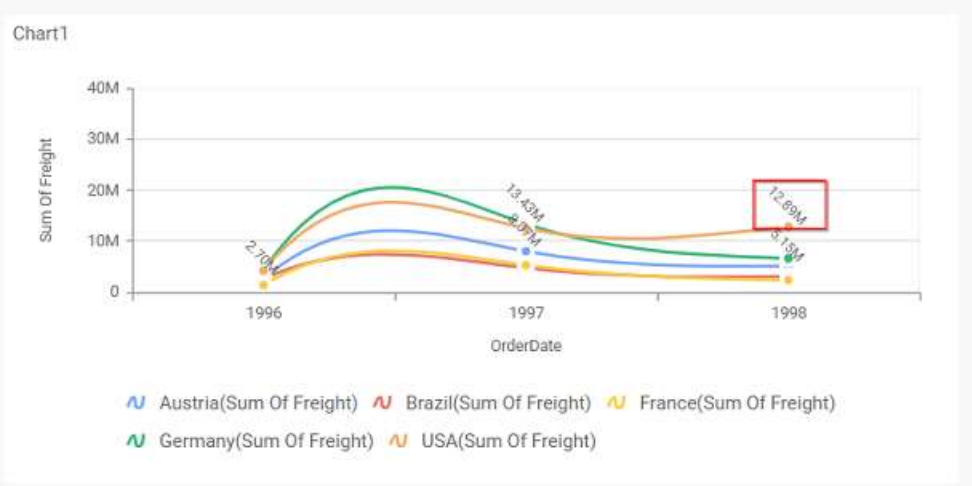
Show Value Labels

This allows you to toggle the visibility of value labels.



Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

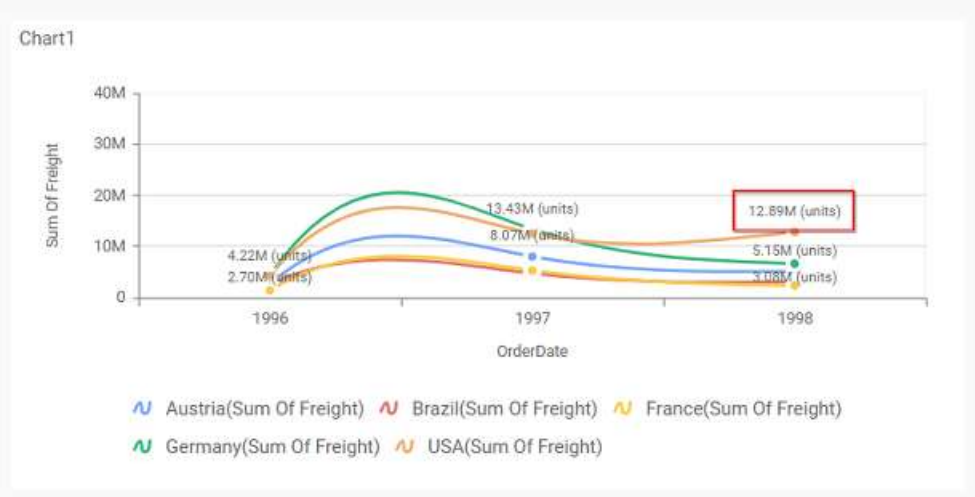


Value Label Suffix

Allows you to enable the Suffix value text to the value labels.

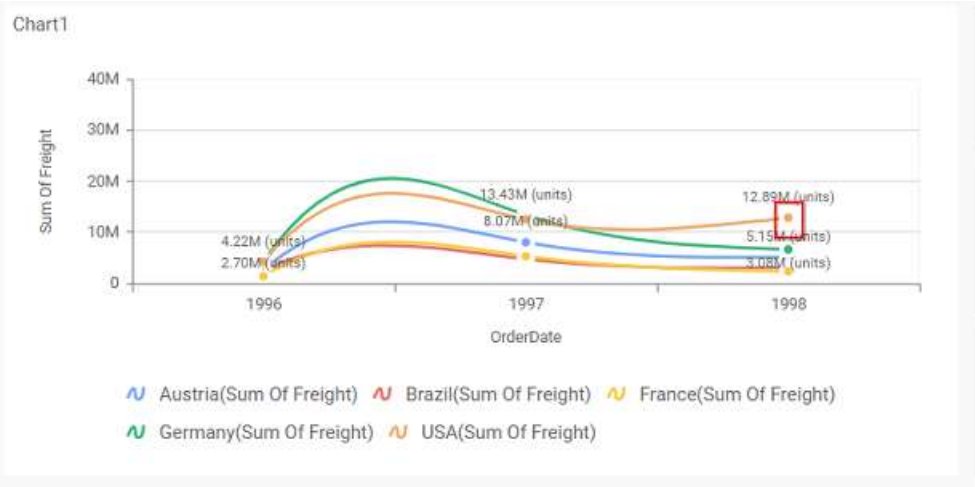
Suffix Value

Allows you to set\edit suffix value to the value labels.

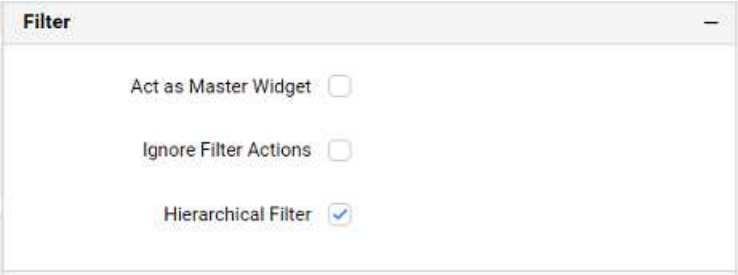


Show Marker

This allows you to toggle the visibility of marker from label to adorn each data point in chart series.



Filter



Act as Master Widget

This allows you to define this chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

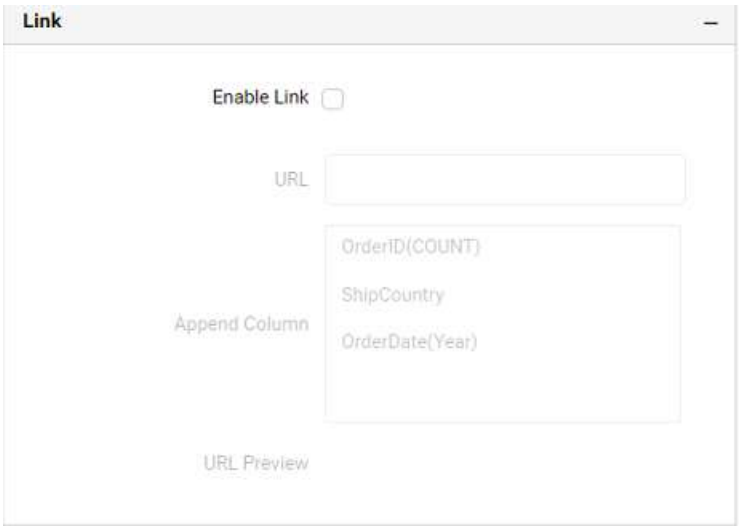
This allows you to define this chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

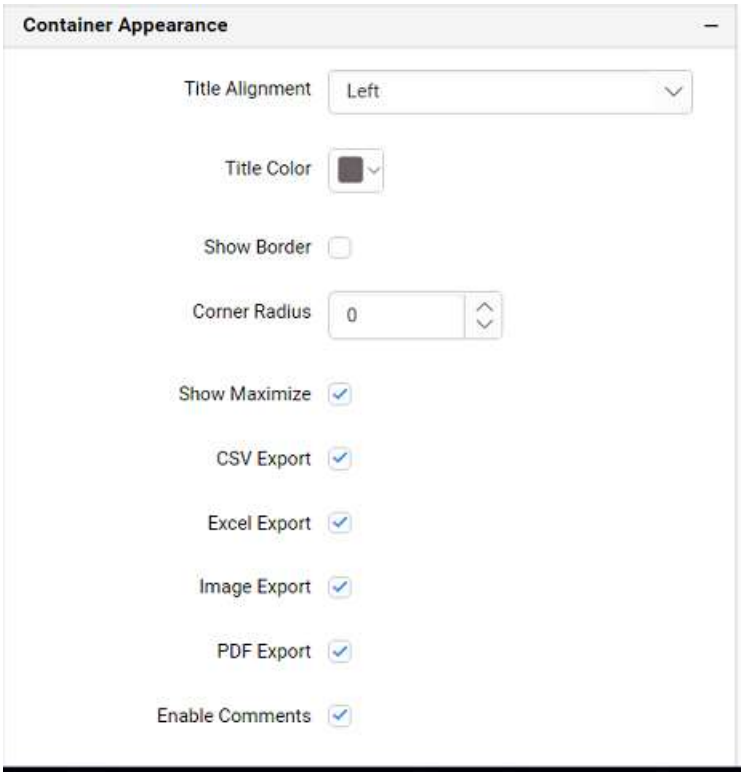
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if the **Show Border** is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this spline chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this spline chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer .

Excel Export

This allows you to enable/disable the Excel export option for this spline chart widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format in viewer .

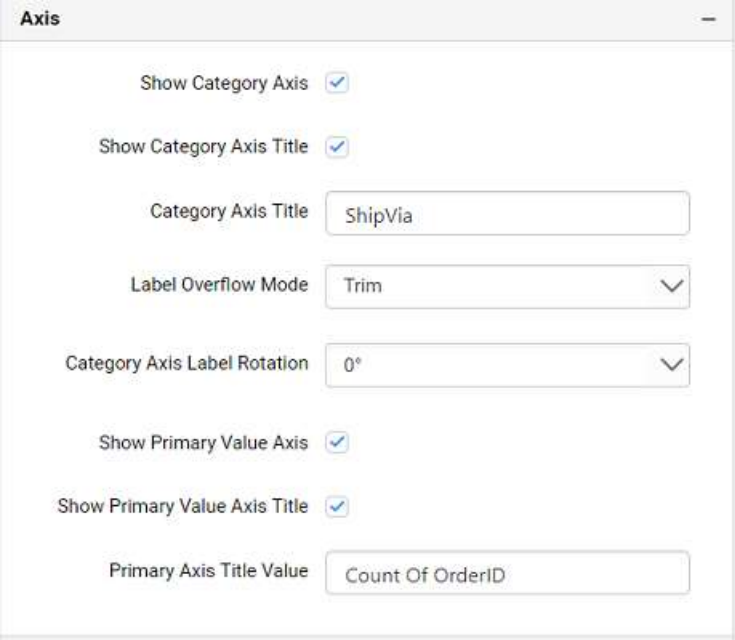
Image Export

This allows you to enable/disable the image export option for this spline chart widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

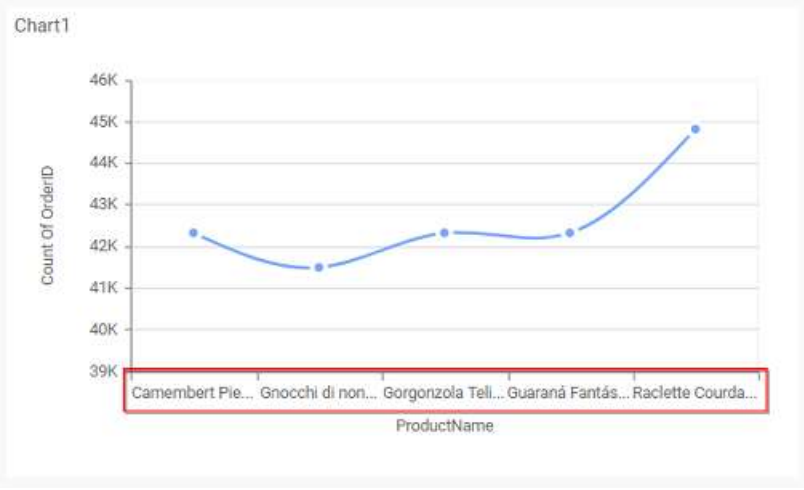
Axis Settings



This section allows you to customize the axis settings in chart.

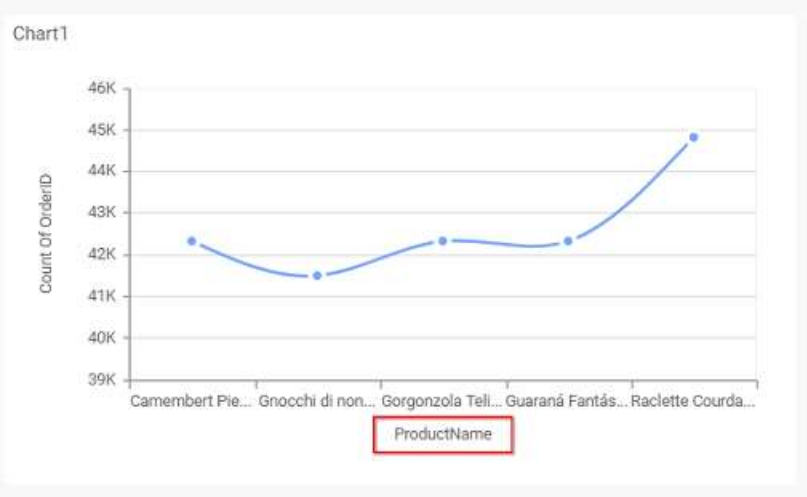
Show Category Axis

This allows to enable the visibility of Category Axis.



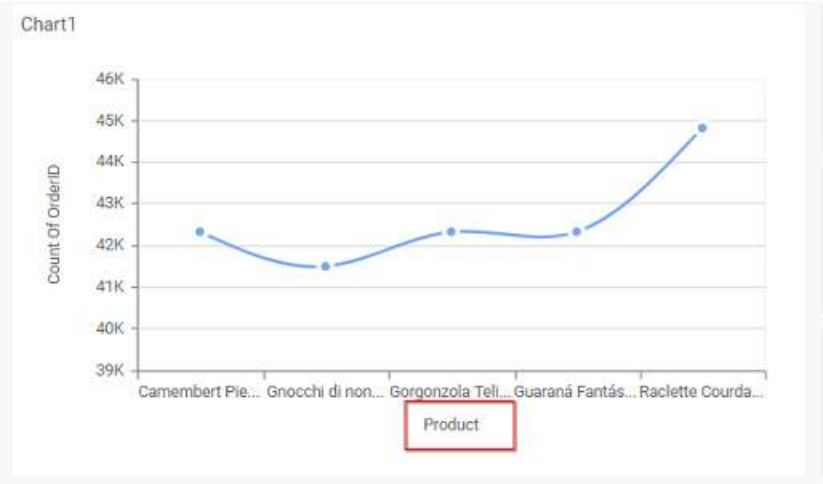
Show Category Axis Title

This allows you to enable the visibility of Category Axis title.



Category Axis Title

This allows you to edit the Category Axis title for chart. It will reflect in x-axis name of chart.

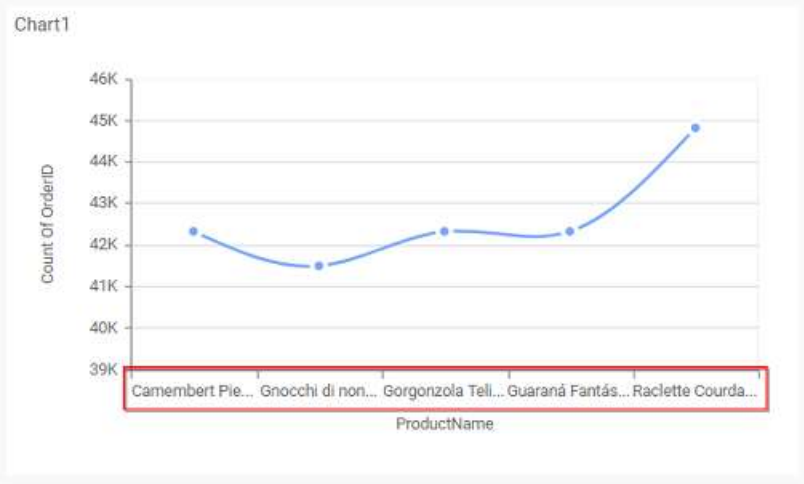


Label overflow mode

This allows you to handle the display mode of the overlapping labels in the Category Axis.

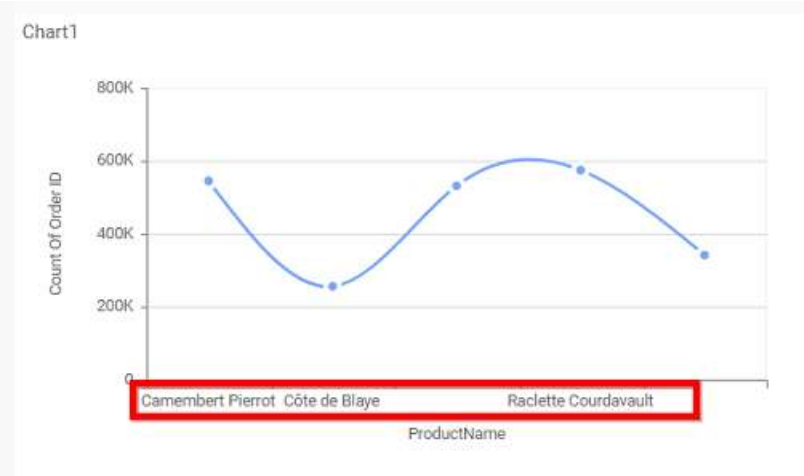
Trim

This option trims the end of overlapping label in the axis.



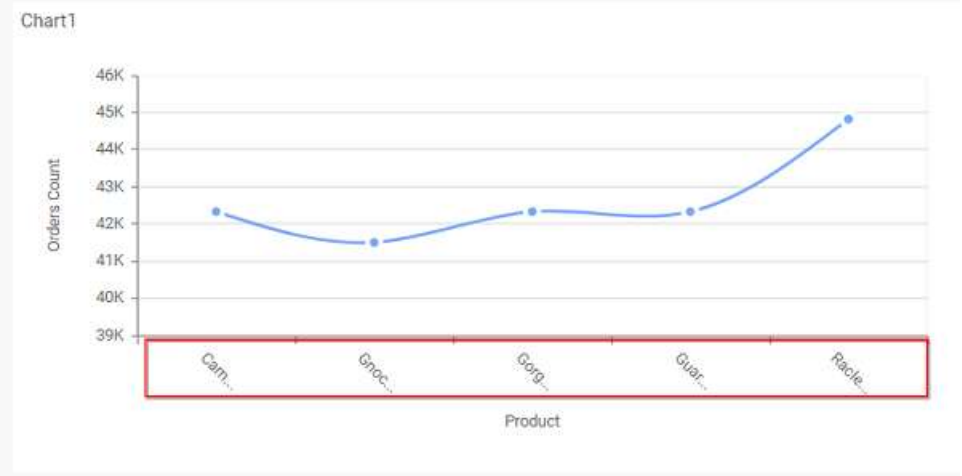
Hide

This option hides the overlapping label in the axis.



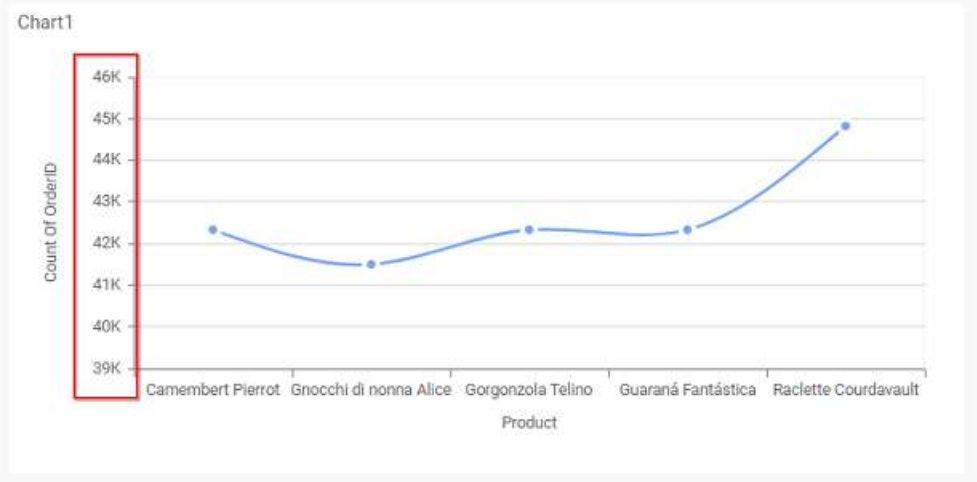
Category Axis Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



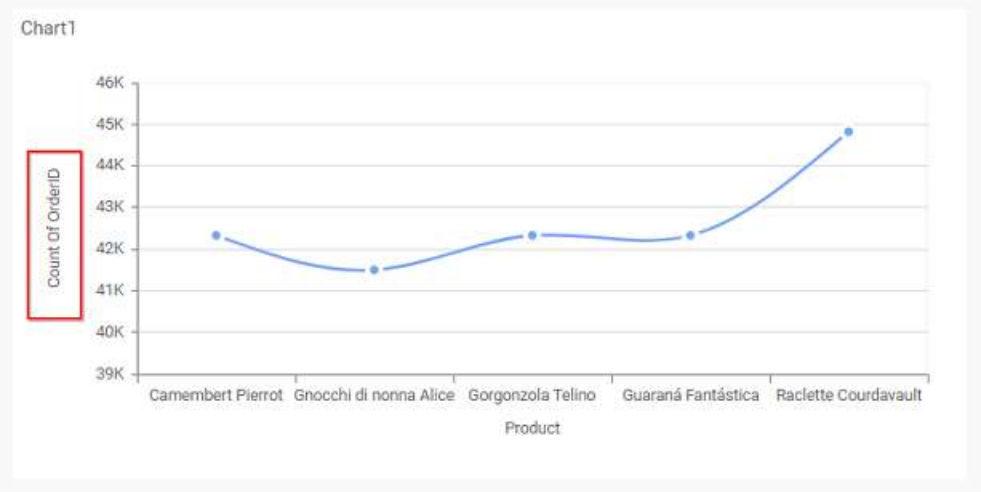
Show Primary Value Axis

This allows you to enable the Primary Value Axis for chart.



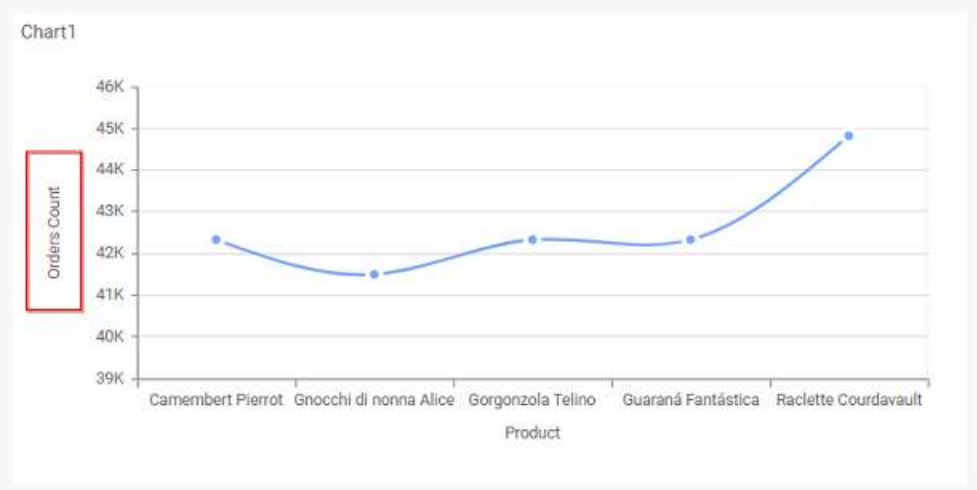
Show Primary Value Axis Title

This allows you to enable the visibility of Primary Value Axis title of chart.

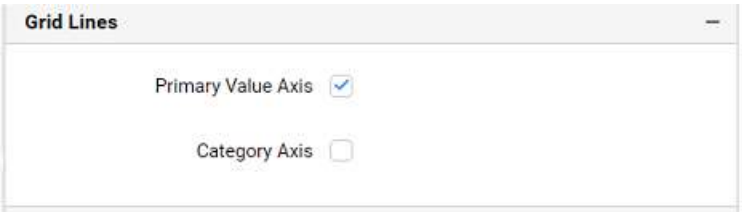


Primary Value Axis Title

This allows you to edit the Primary Value Axis title. It will reflect in y-axis name of chart.

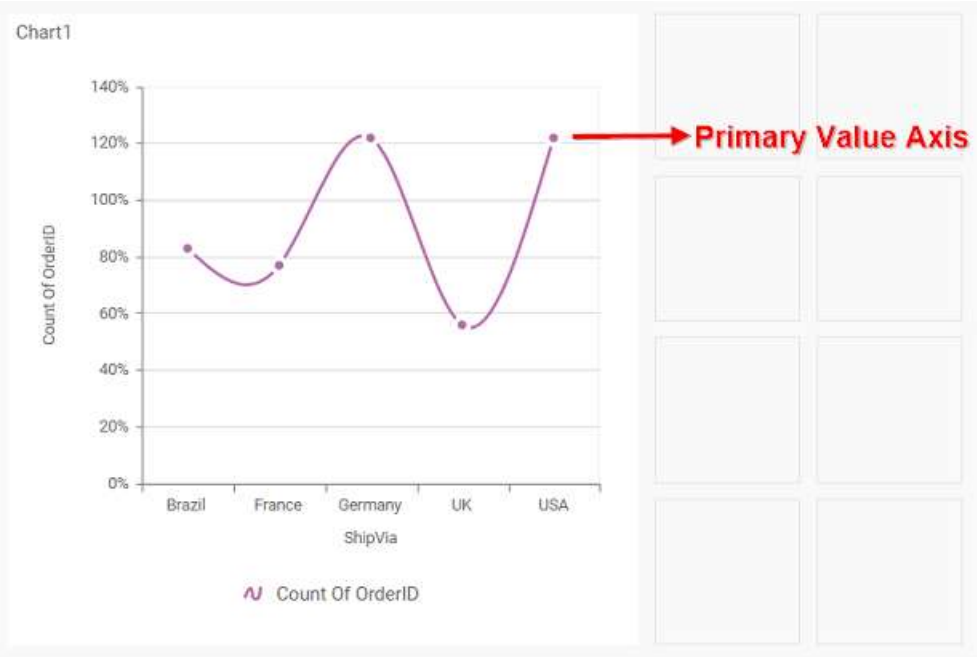


Grid Line



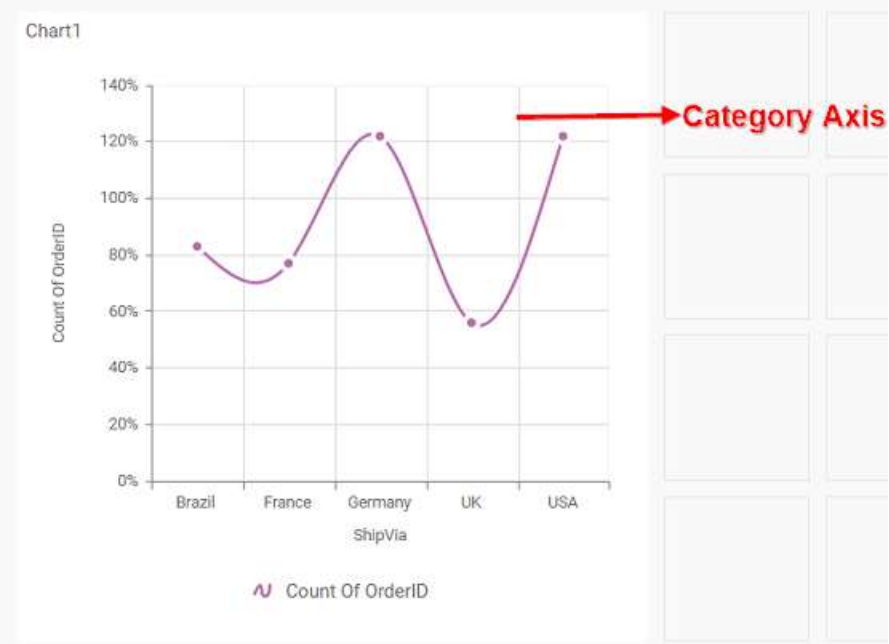
Primary Value Axis

This allows you to enable the Primary Value Axis gridlines for the spline chart.



Category Axis

This allows you to enable the Category Axis gridlines for the spline chart.

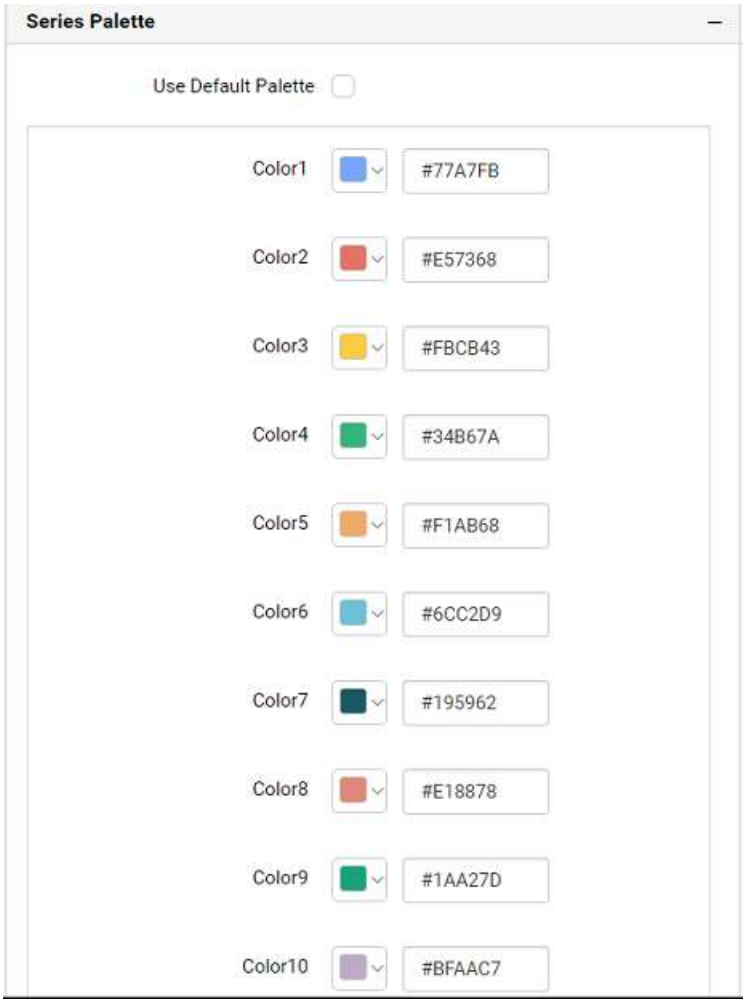


Series Palette

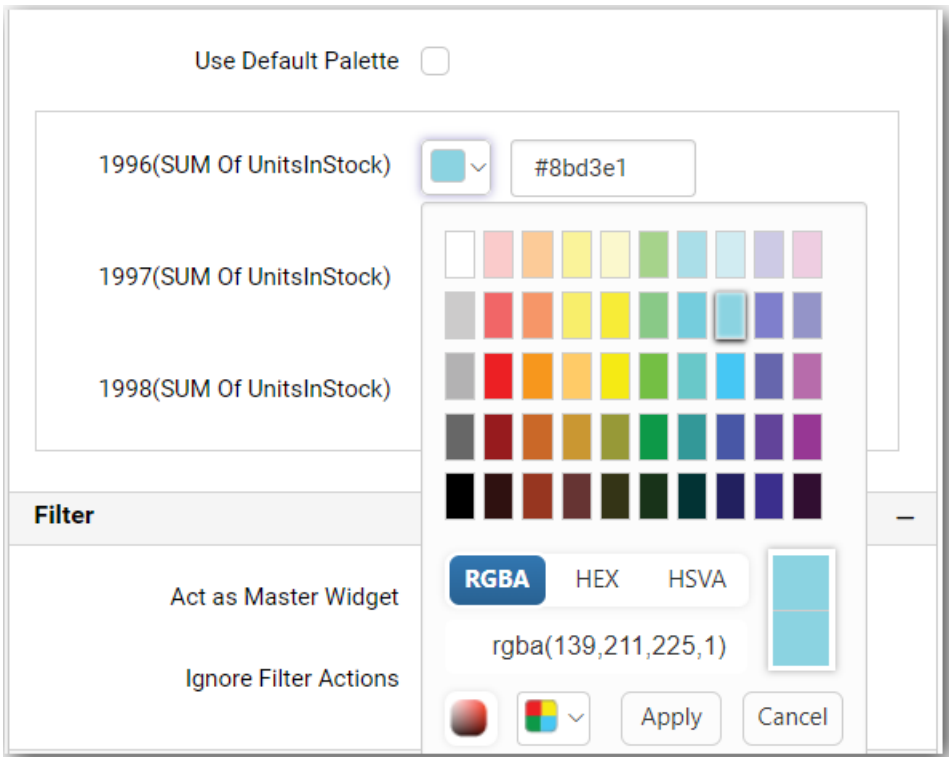
This allows you to customize the chart series color through Series Palette section.

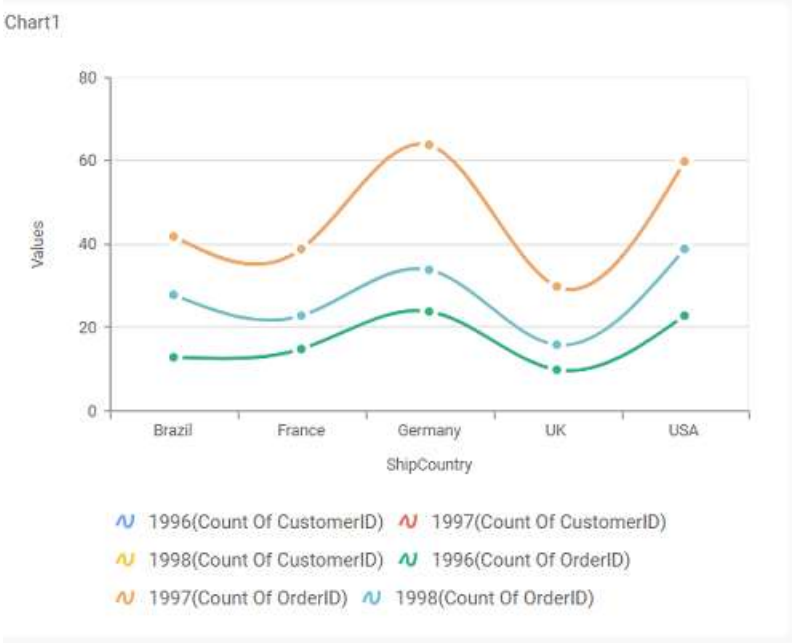
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Grid

Grid allows you to showcase ranking relationship through vertical arrangement of items, ordered from top to bottom.

Grid1

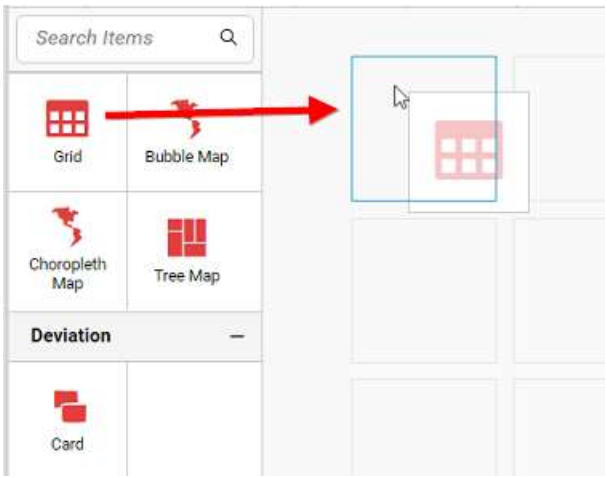
Product	Quantity	UnitsInStock	UnitsOnOrder	ReorderLevel
Aniseed Syrup	328	156	840	300
Chang	1,057	748	1,760	1,100
Chocolade	138	90	420	150
Gnocchi di non...	1,263	1,050	500	1,500
Gorgonzola Teli...	1,397	0	3,570	1,020
Gravad lax	125	66	300	150
Ipoh Coffee	580	476	280	700
Longlife Tofu	297	52	260	65
Louisiana Hot S...	239	32	800	160
Mascarpone Fa...	297	135	600	375
Maxilaku	520	210	1,260	315
Outback Lager	817	585	390	1,170
Queso Cabrales	706	836	1,140	1,140
Rogede sild	508	70	980	210

How to configure the table data to Grid?

To construct a grid, a minimum requirement of 1 column is needed. You can visualize both measure, calculated measure and dimension column data in grid control. You can also add a column that is hidden from the view by adding the column in the hidden columns section. The data of these columns will be hidden from the view but can be used for filtering other widgets in the dashboard.

The following procedure illustrates data configuration of grid.

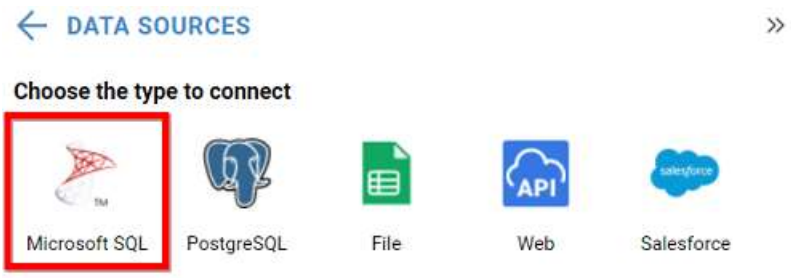
Drag and drop Grid control icon from the tool box into design panel. You can find control in tool box by search.



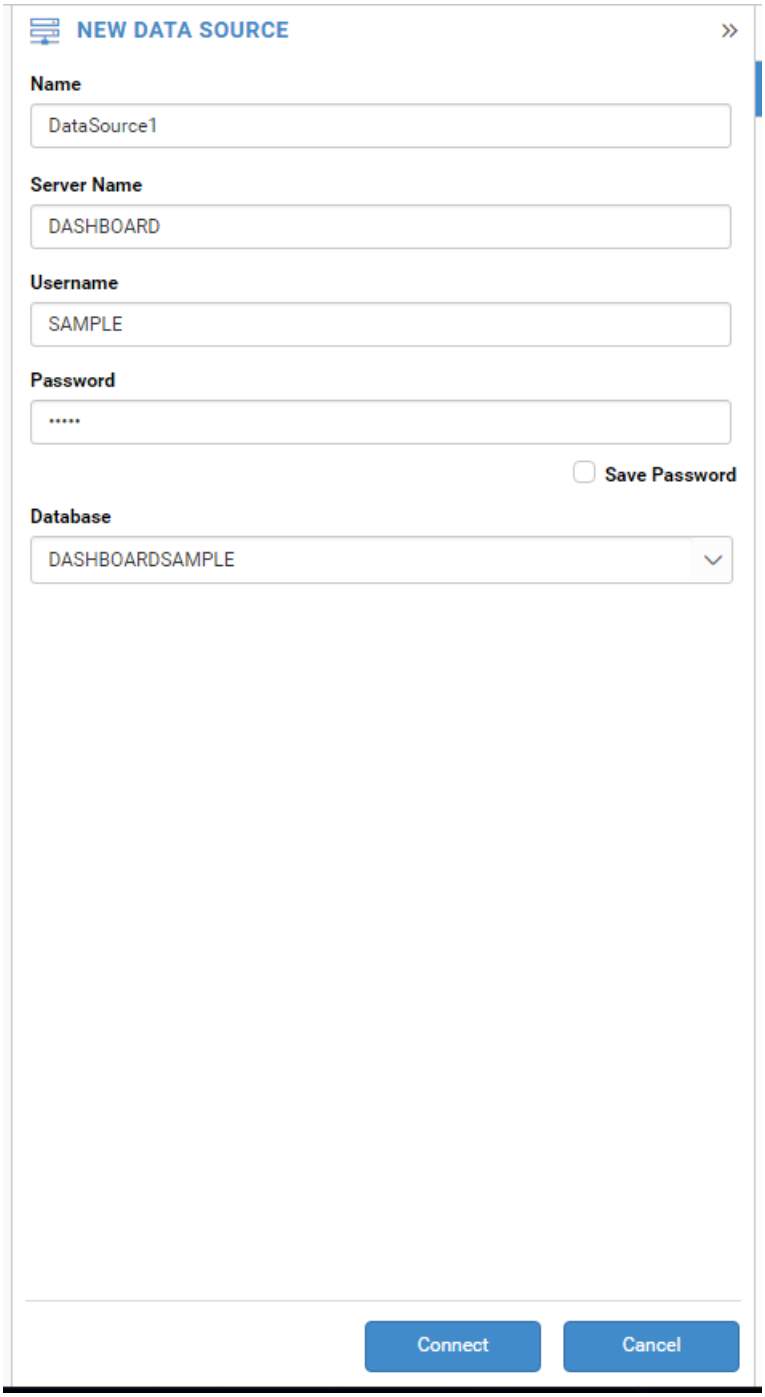
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

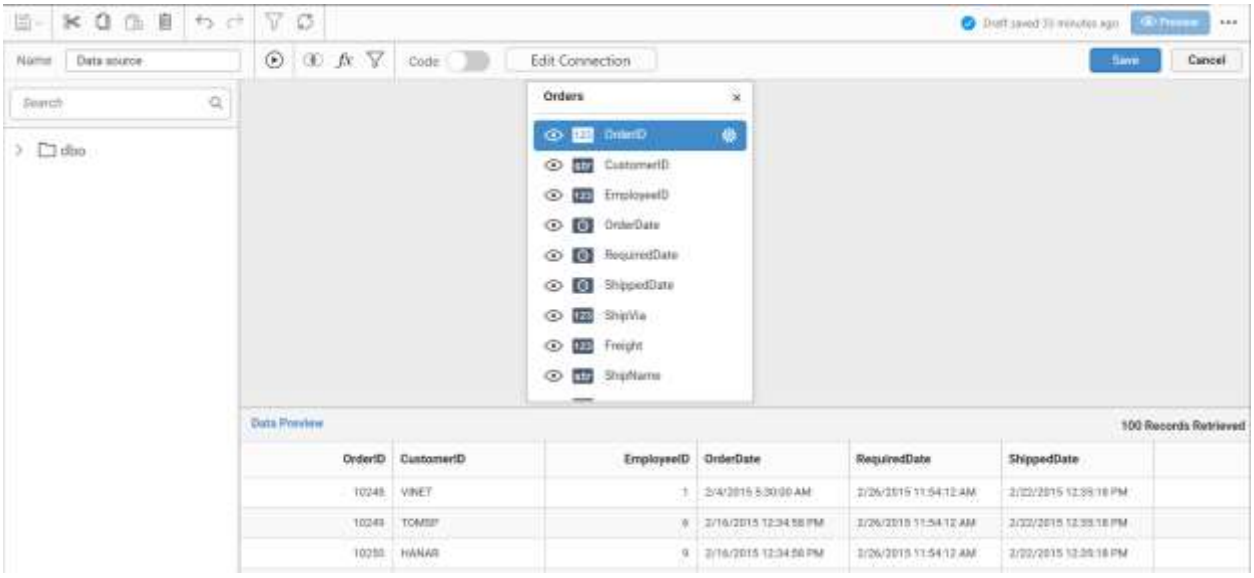
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



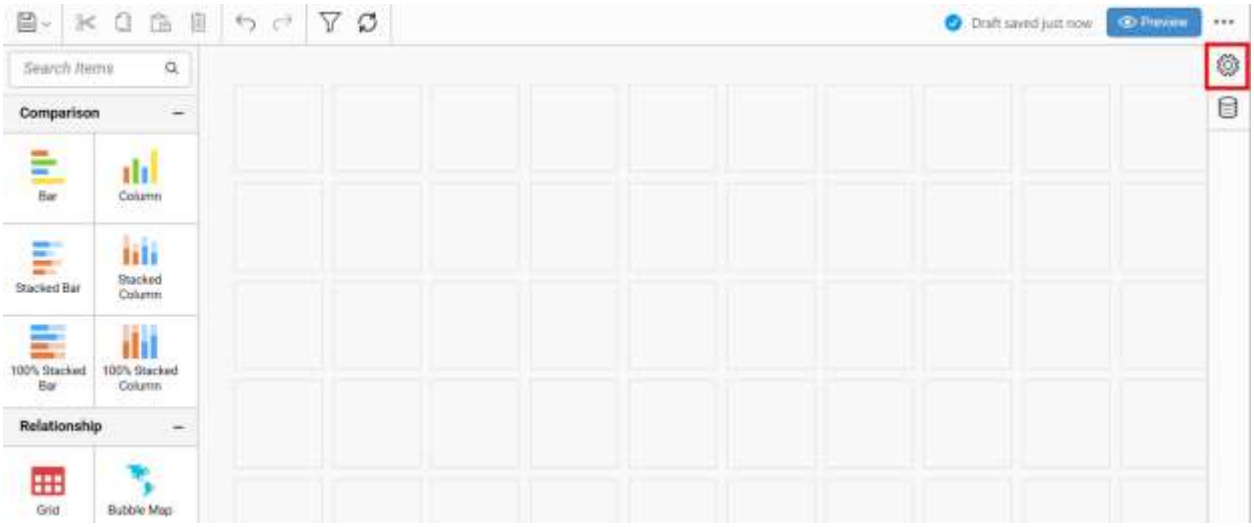
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

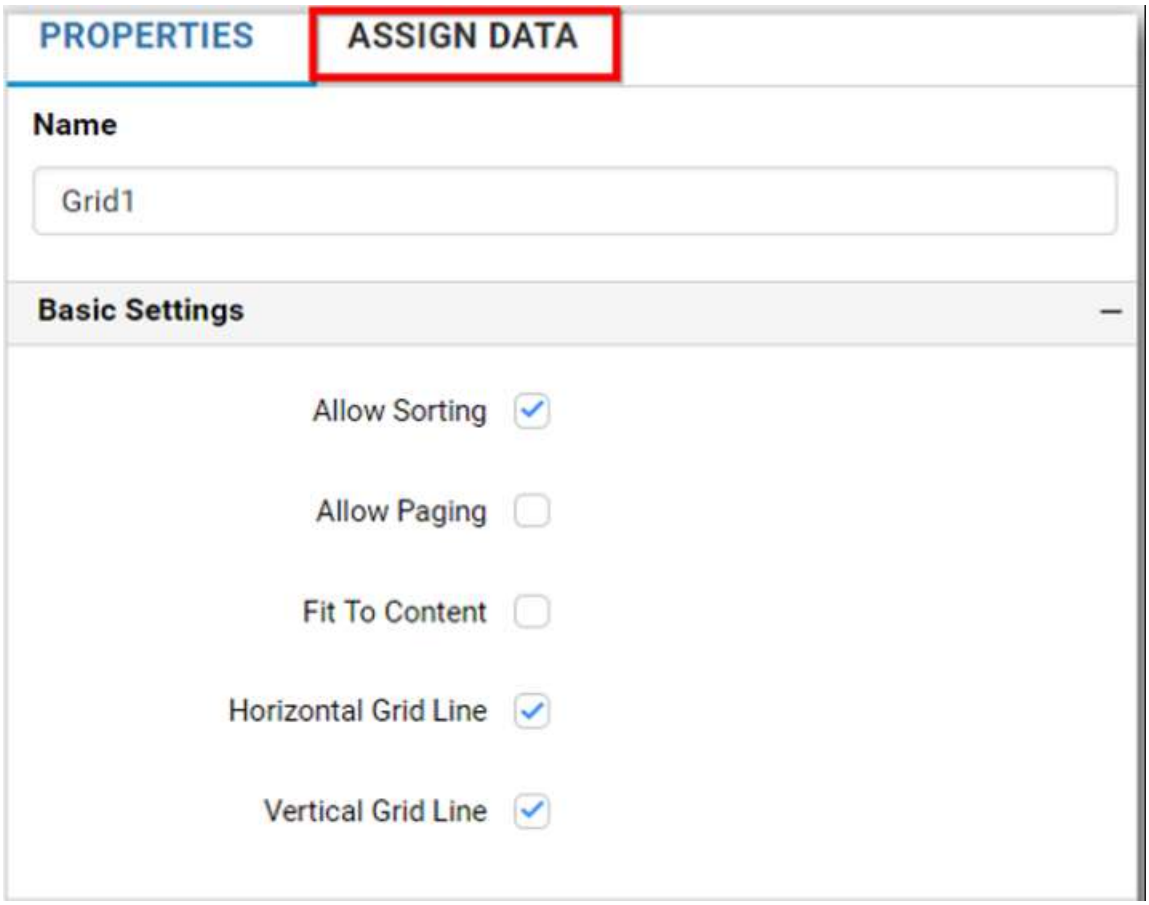


Drag your preferred table or view from the left pane from data design view, click Save button.

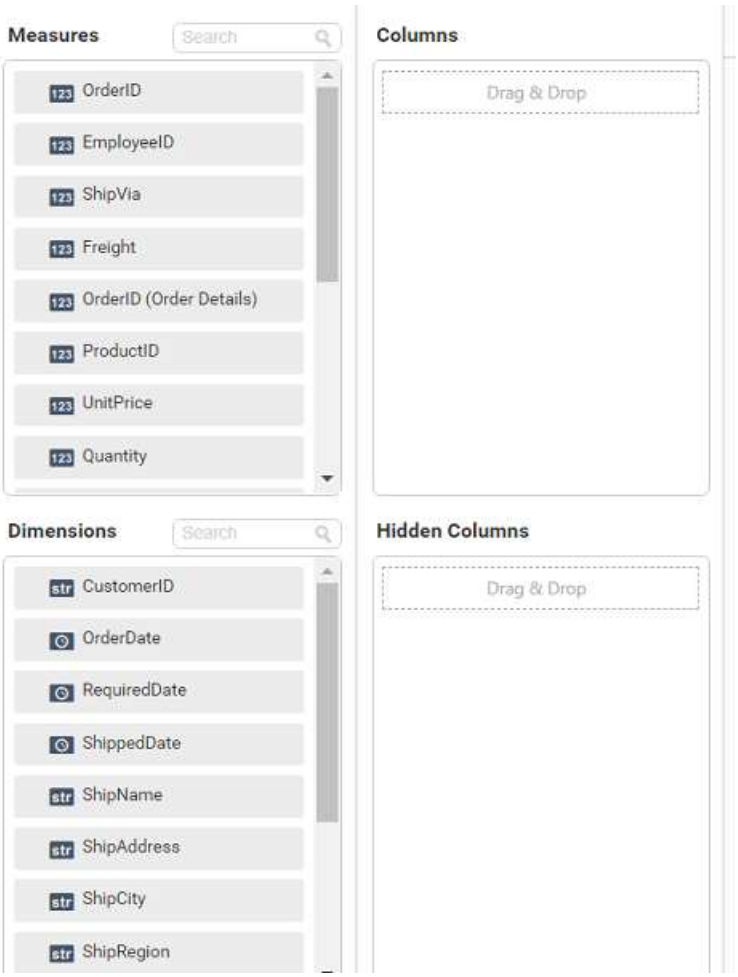


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.

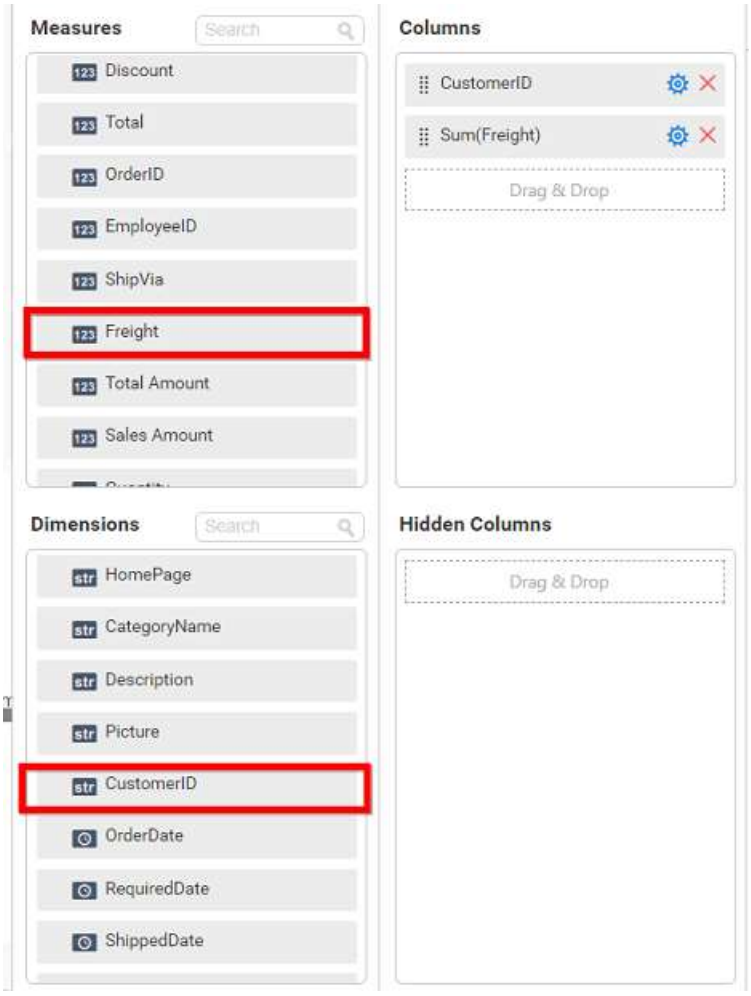




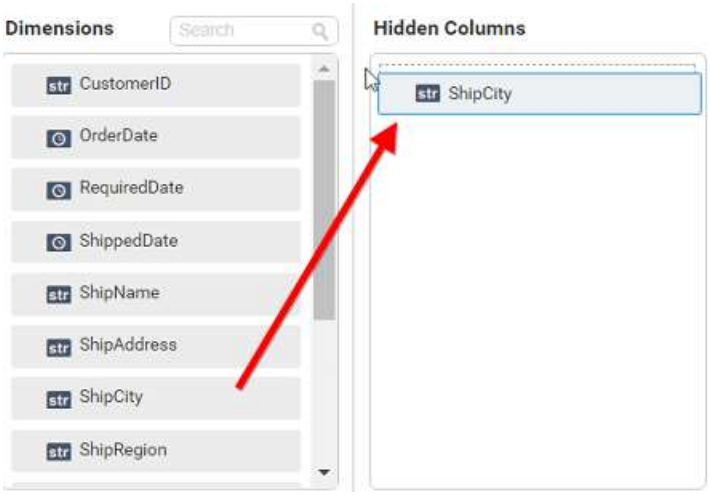
The data tab will be opened with available columns from the connected data source



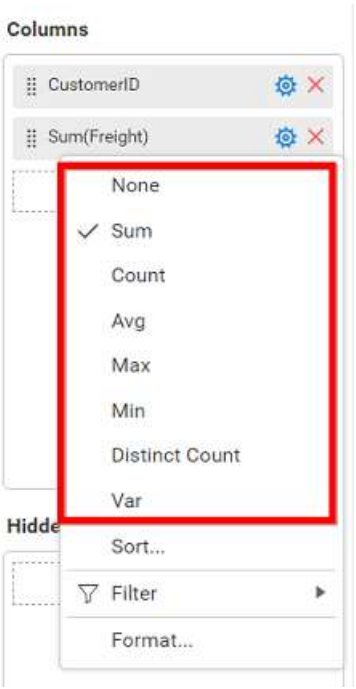
Bind column through drag and drop element from Measures or Dimension section to Columns section.



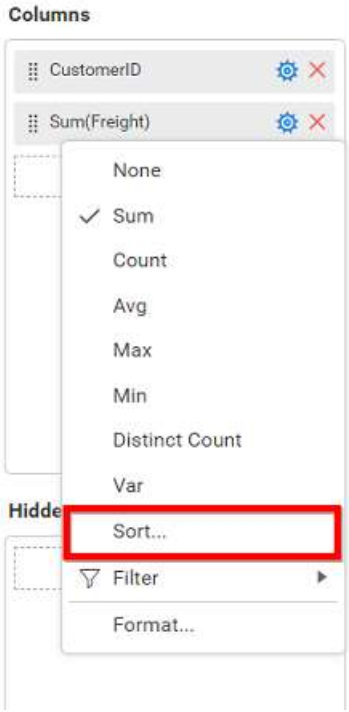
You can drag and drop the elements to Hidden Columns if required. Based on the hidden columns elements the values will be shown in grid widget.



You can use aggregate function to change the column values by clicking the settings.

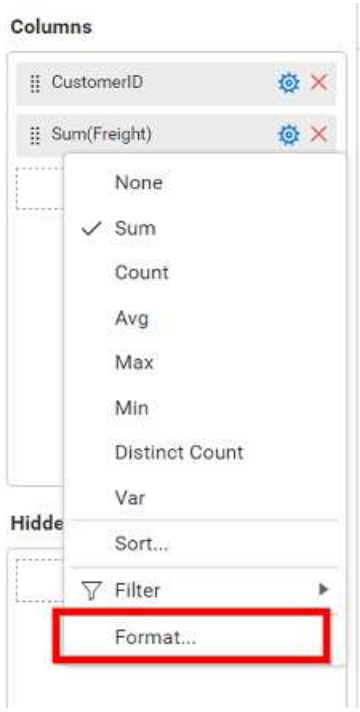


You can Sort the data using Sort option shown under Settings menu list. To sort the measure data, refer Sort

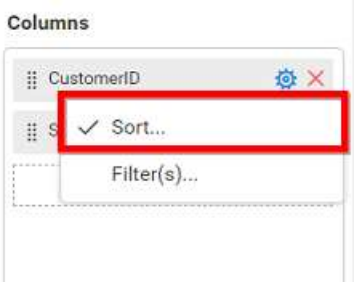


You can use Filters to change the values by selecting the Filter option. For more details, refer filter

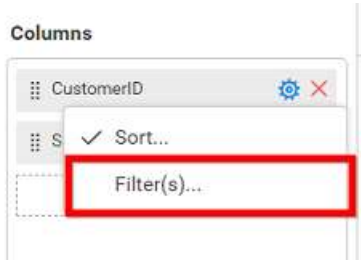
You can format the elements by selecting the Format option. For more details, refer measure format.



For dimension field, you can Sort the dimension data using Sort option under Settings menu list. To apply sorting for the data, refer Sort.



For dimension field, you can apply filters by selecting filters option in settings. For more details, refer filter.



Here is an illustration,

Grid1

Customer ID	Freight Charge
ALFKI	$225.6
ANATR	$97.4
ANTON	$268.5
AROUT	$472.0
BERGS	$1,559.5
BLAUS	$168.3
BLONP	$623.7
BOLID	$191.2
BONAP	$1,357.9
BOTTM	$794.0

How to format grid widget?

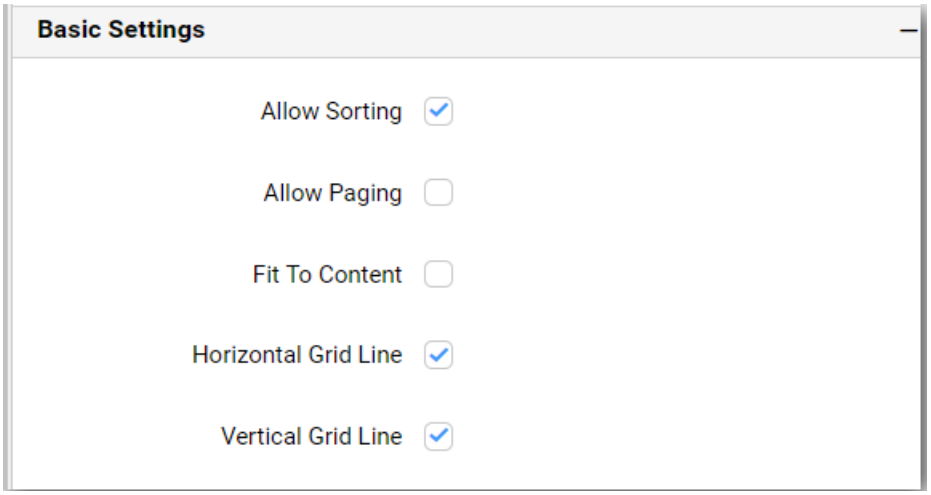
You can format the grid for better illustration of the view that you require, through the settings available in Properties tab.

General Settings

Name

This allows you to set title for this grid widget.

Basic Settings



Allow Sorting

You can toggle the interactive sorting of columns in grid control using this. This option is enabled by default.

Fit To Content

The columns in the grid can be made to auto size based on the length of the content of the column. This option is not enabled by default

Horizontal Grid Line

You can enable/ disable horizontal grid lines in grid control. This option is enabled by default.

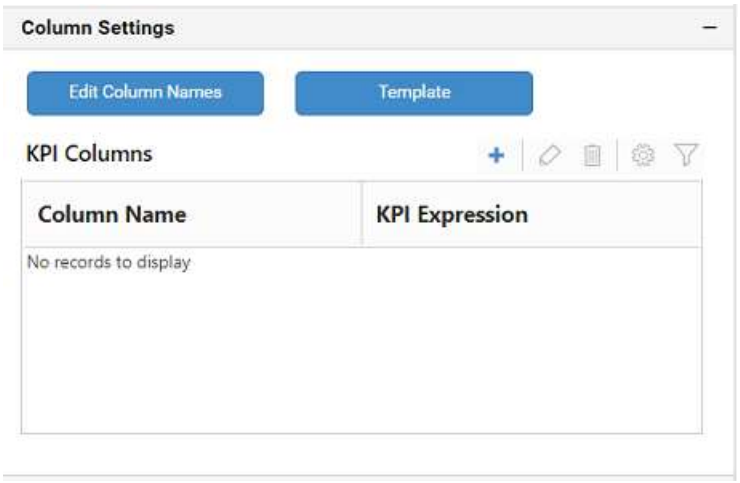
Vertical Grid Line

You can enable/ disable vertical lines in grid control. This option is enabled by default.

Allow Paging

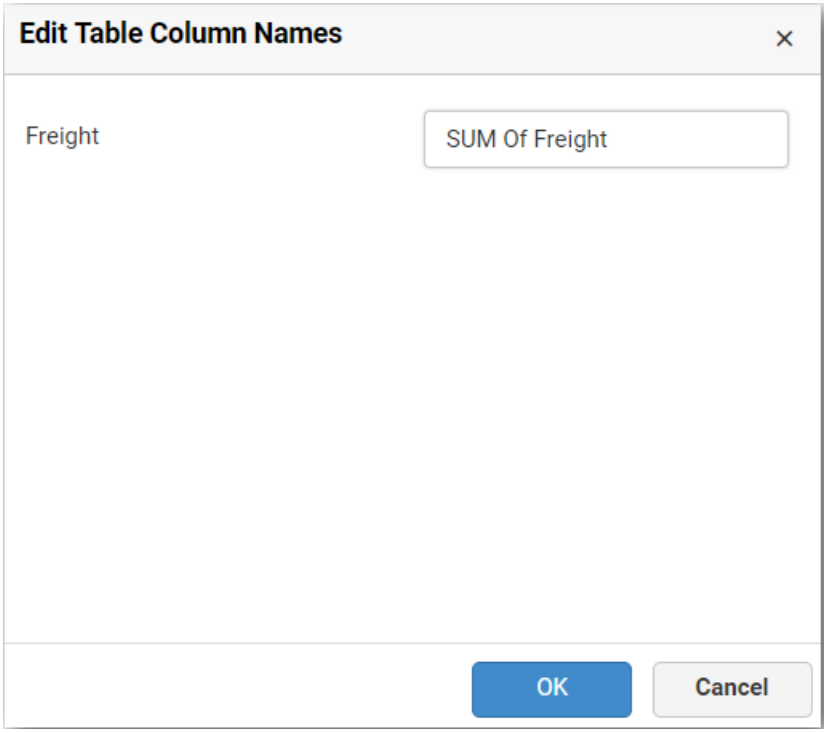
You can toggle the paging option in the grid to show number of pages. This option is not enabled by default

Column Settings



Edit Column Names

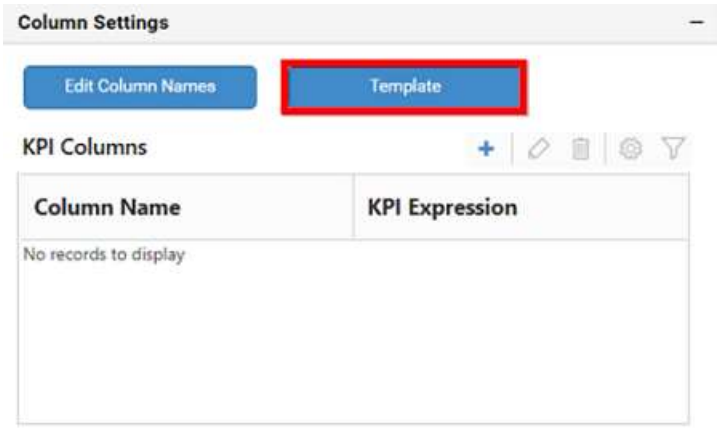
This allows you to customize the column names displayed in the grid by navigating to edit columns dialog from property panel. Through Edit Table Column Names window, a different name can be set to individual columns.



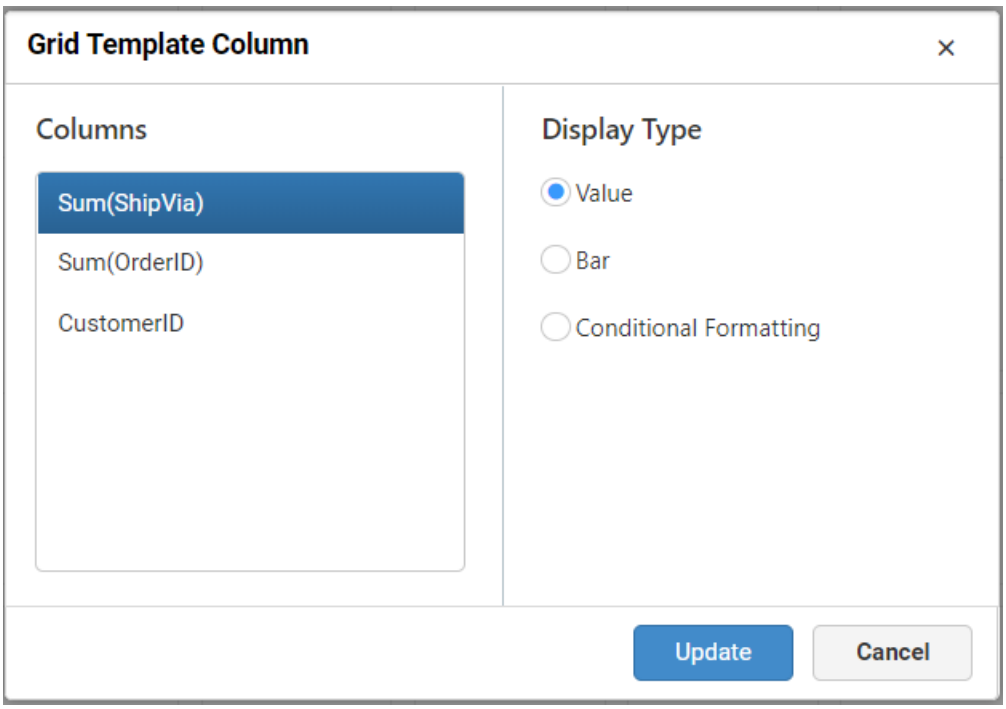
Once the changes are made, you can save by clicking the OK button.

Template

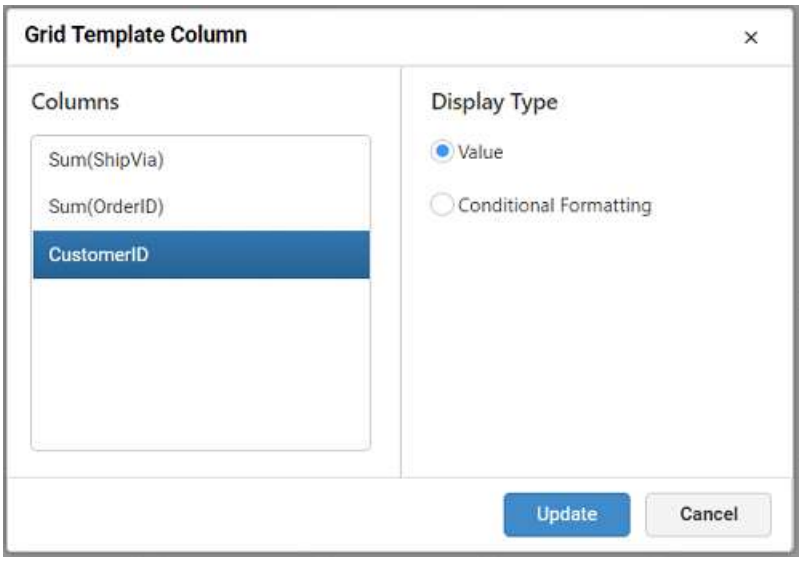
You can define the column value represented as text, bar or condition based coloring. Click Template to launch the Grid Template Column dialog. This lists out the columns added to the grid widget. For each of those columns, the value representation can be configured through options displayed at right.



For measure type column,



For dimension type column,



To define the value representation for a column, select the respective column from the Columns list in the left pane and select the display type for the same at right pane.

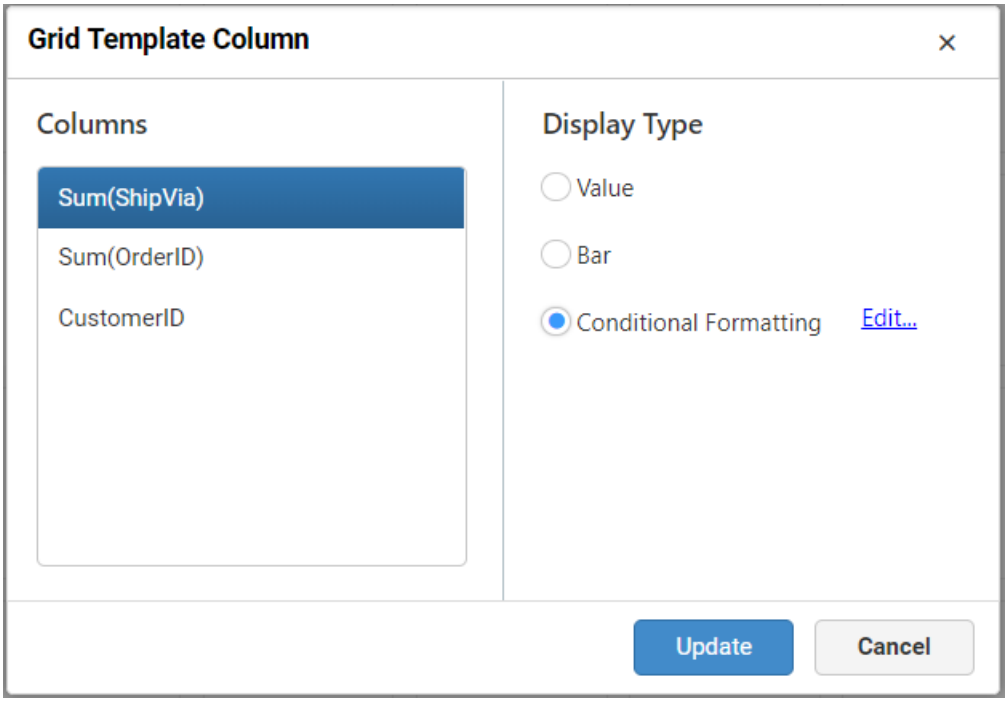
Select Value as display type, to get the column values represented as it is.

Product Name	Company Name	Sales Amount
Côte de Blaye	Aux joyeux ecclésiastiques	164,424
Mishi Kobe Niku	Tokyo Traders	108,640
Carnarvon Tigers	Pavlova, Ltd.	39,000
Gustaf's Knäckebröd	PB Knäckebröd AB	31,584
Ikura	Tokyo Traders	29,760

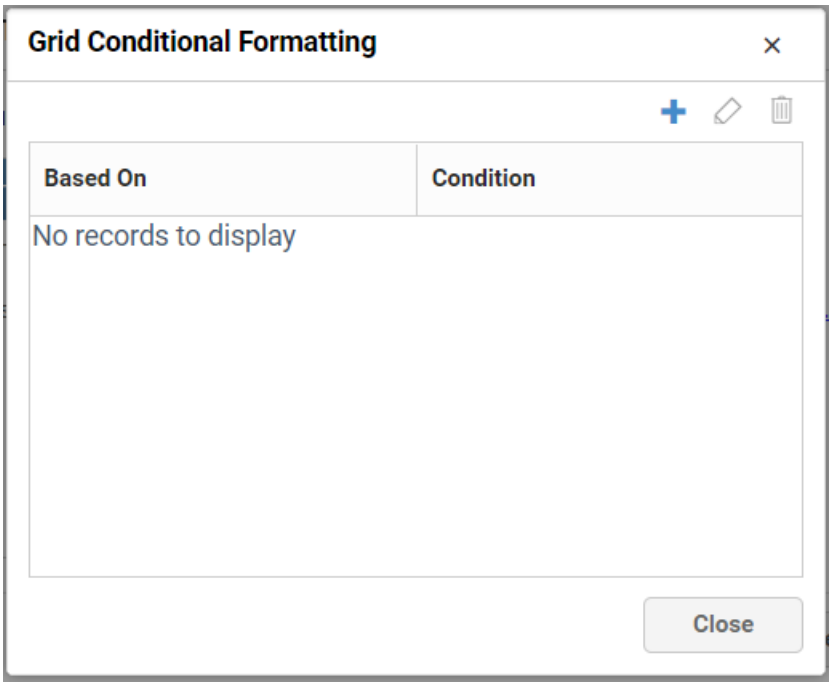
Select Bar as display type, to get the column values represented as progress bar.

Product Name	Company Name	Sales Amount
Côte de Blaye	Aux joyeux ecclésiastiques	
Mishi Kobe Niku	Tokyo Traders	
Carnarvon Tigers	Pavlova, Ltd.	
Gustaf's Knäckebröd	PB Knäckebröd AB	
Ikura	Tokyo Traders	

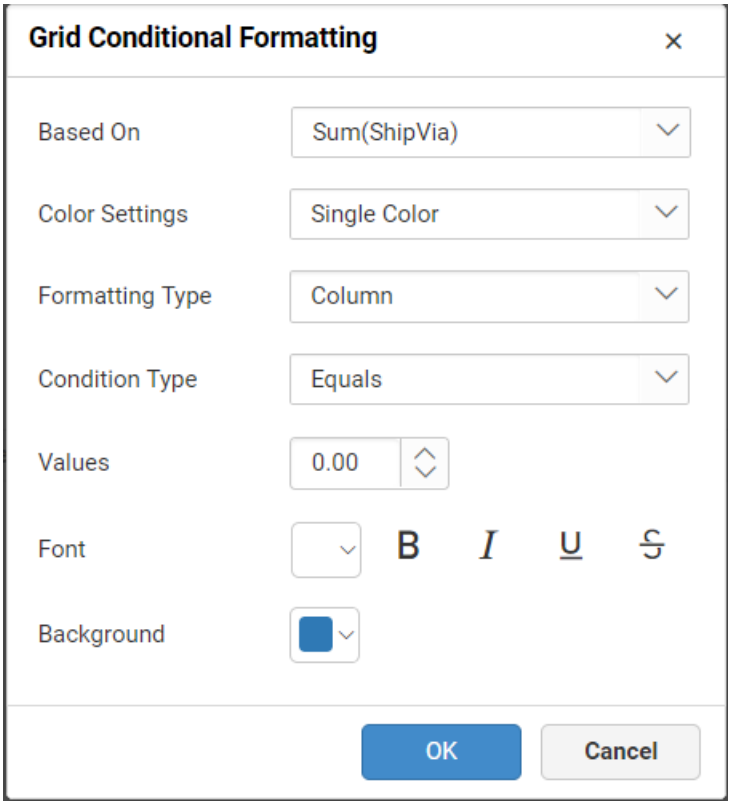
Select Conditional Formatting to configure conditions and apply color to the cells based on that.



The Grid Conditional Formatting dialog can be opened through clicking the Edit link button.



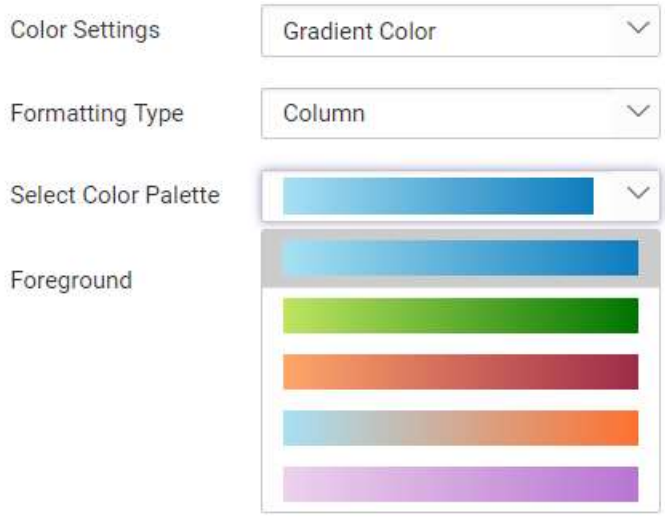
Add one or more conditions through clicking the icon.



In this dialog,

Based On – Shows added column list out of which a column can be selected over which the condition need to be defined and applied to the column selected for template display.

Color Settings – It can be Single Color or Gradient Color. Selecting Single Color will apply the selected color to the rows/column that meets the filter criteria. Selecting Gradient Color will apply the selected palette to the rows/column that meets the filter criteria.



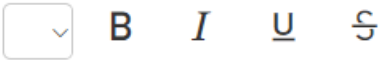
Formatting Type – It can be Row or Column which need to be applied with the formatting.

Column Formatted Grid

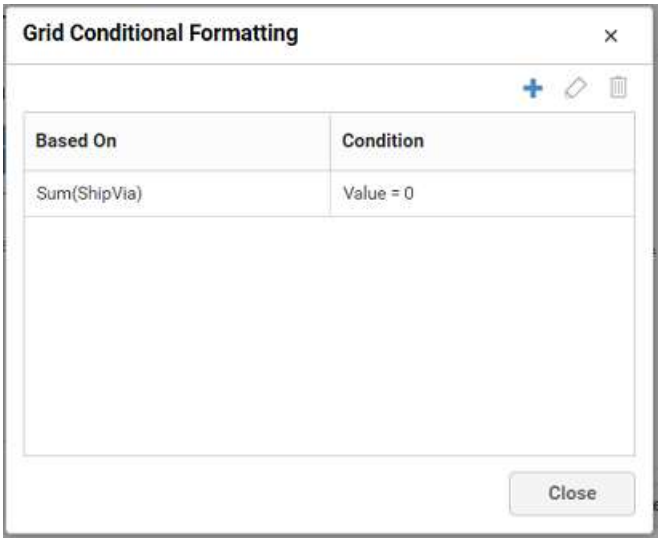
Row Formatted Grid

Condition Type – The compare operator can be set to compare values against.

Other font settings like color, style can be set.



Click OK to save the condition in Grid Conditional Formatting window. You can also edit an already added condition through selecting the respective condition and click the Edit icon highlighted below.



Key Performance Indicator (KPI)

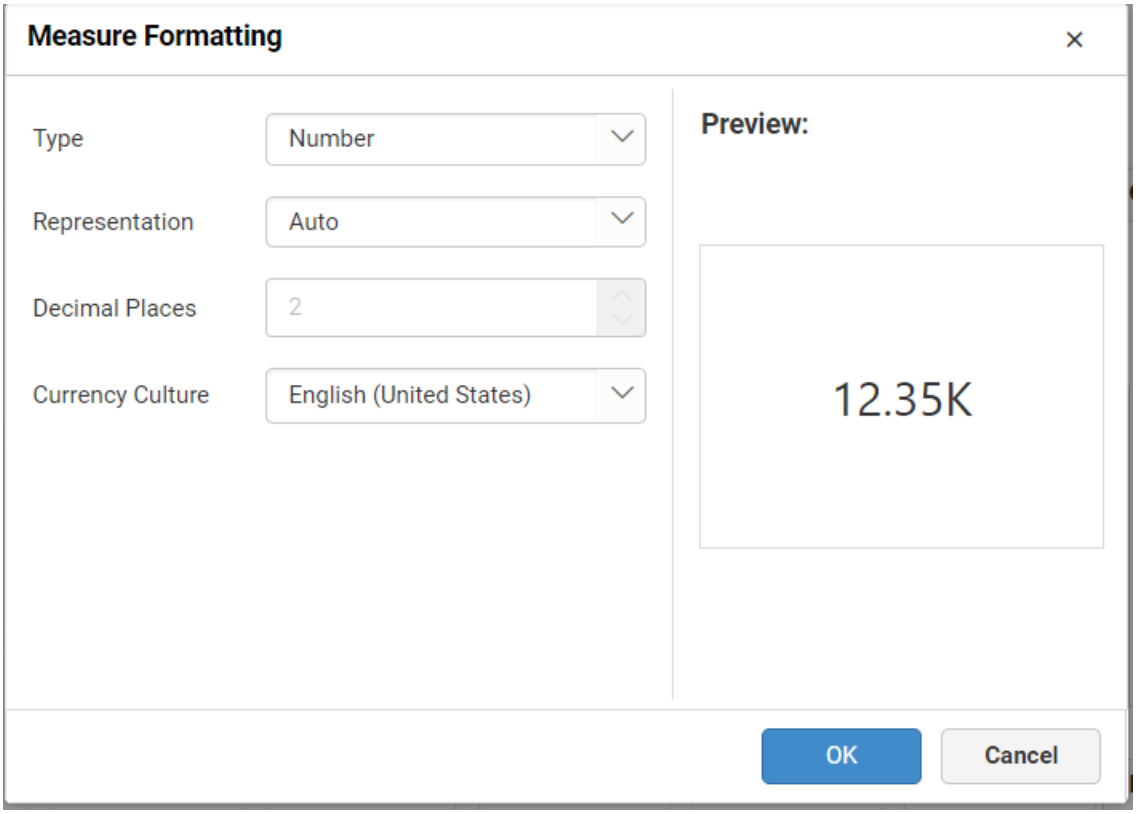
You can add Key Performance Indicator (KPI) columns in grid control by navigating to KPI Expression window by clicking Add KPI button from property panel at top.

From the KPI Expression dialog, you can specify the column whose values need to be considered as actual value and the column that need to be considered as target. The value type can be set based on which the KPI will be calculated. The following value types are available.

- Actual Variation (Default)
- Actual Value
- Percentage of variation
- Percentage of target
- Value and Percentage

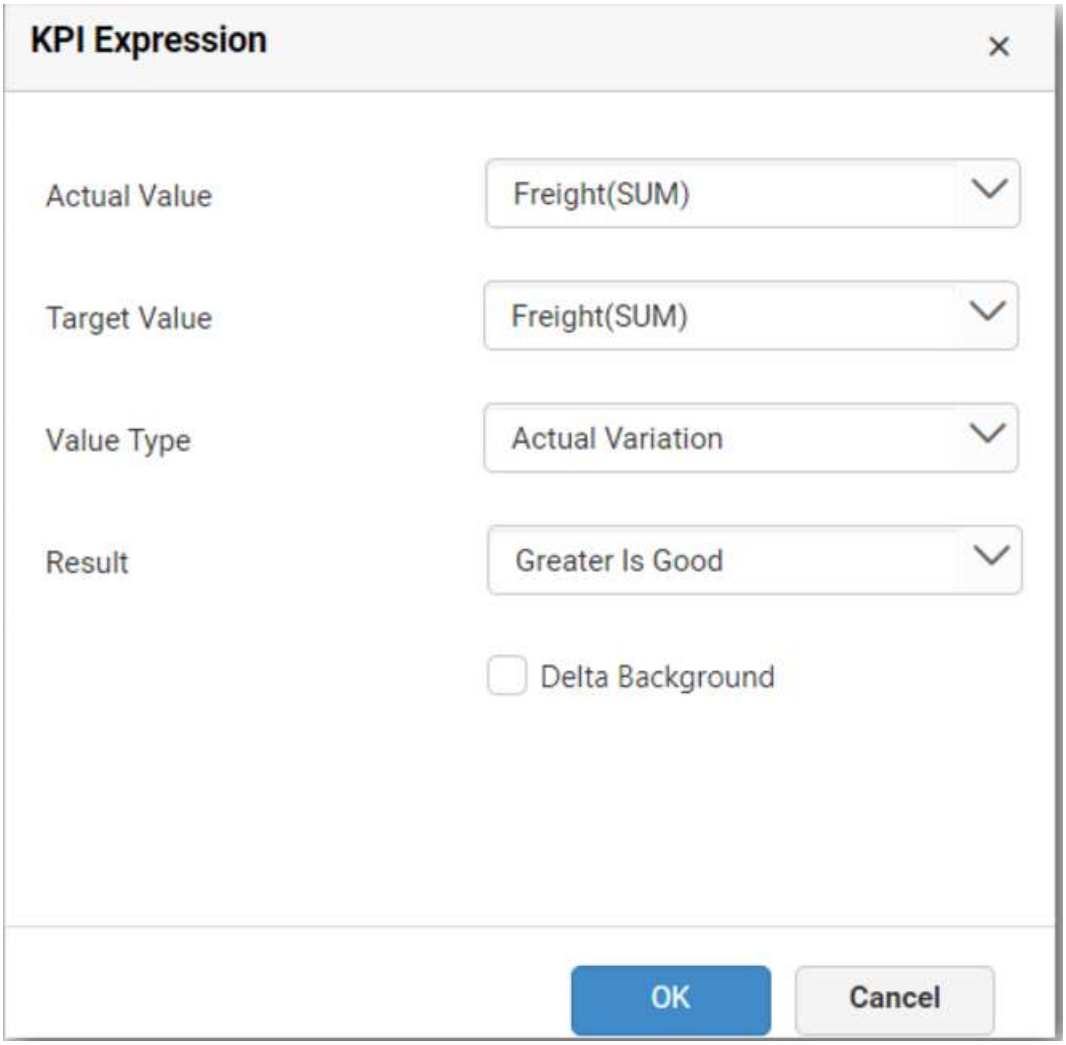
The Result can be set to showcase the result as gain or loss based on which the value will be visualized. You can choose the type as value or graphical bar to showcase the data in the column.

The KPI column can be filtered based on the measure values bind to the KPI. You can set measure filter by clicking the filter icon button, which will open the Measure Filter dialog from where you can specify the column and the condition for filtering the data showcased.

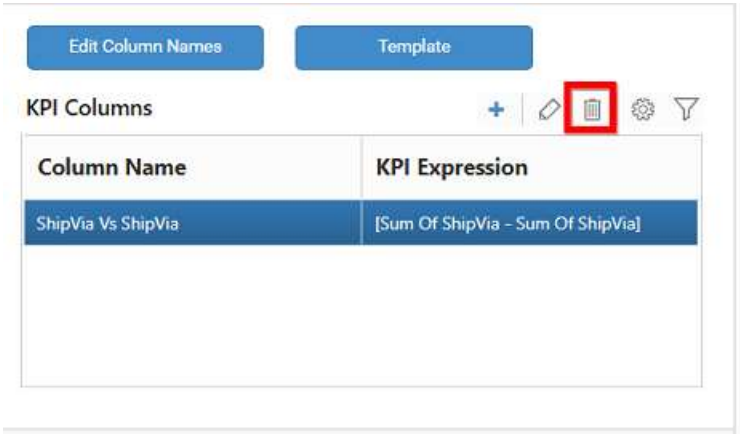


The values showcased in KPI column can be formatted just like any other measure column. You can open the Measure Formatting dialog box by clicking the format KPI column button. This allows you to handle different formatting options like display type, representation, decimal places and currency culture to the respective KPI column added.

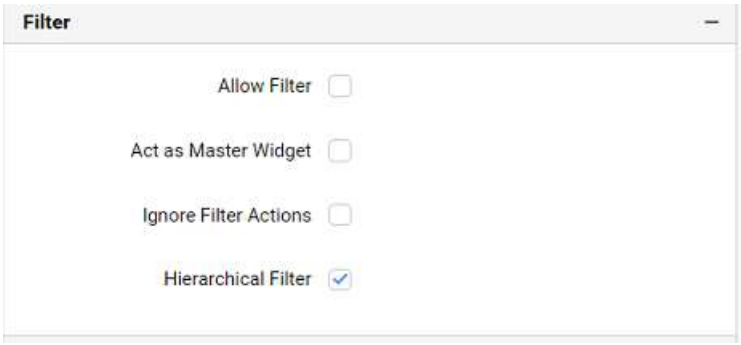
You can Edit KPI column by clicking the Edit KPI column icon.



You can delete KPI column by clicking the Delete KPI column.



Filter



Allow Filter

This allows you to enable a filter box for each column in the grid for easy filtering of data through this option.

Act as Master Widget

This allows you to define this grid widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

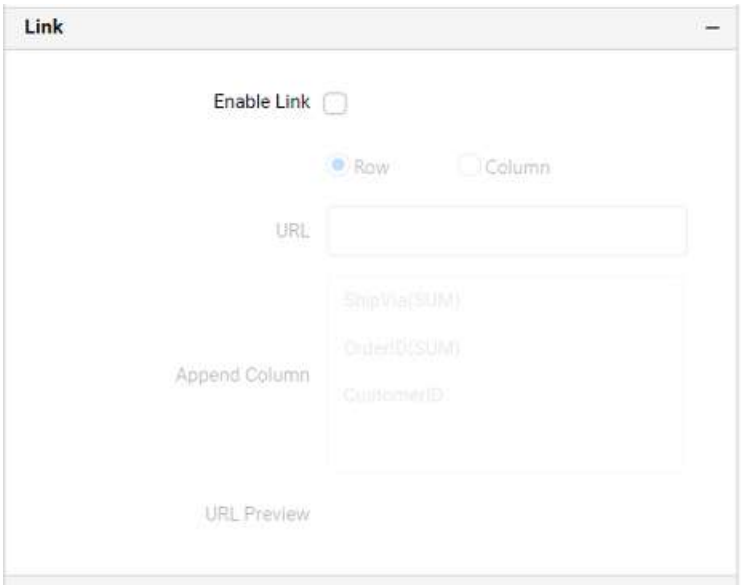
This allows you to define this grid widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

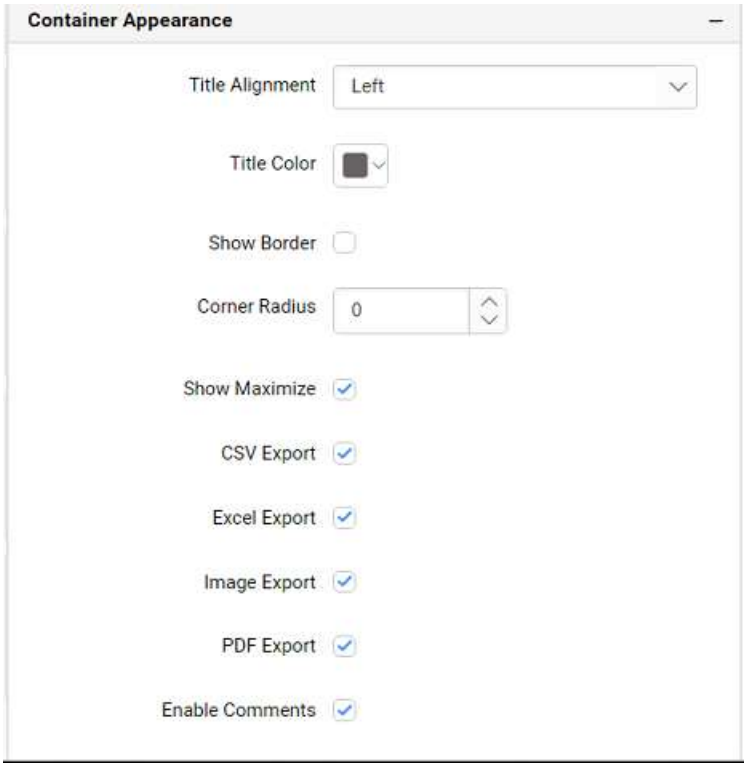
When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if **Show Border** property is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this grid widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the grid widget.

CSV Export

This allows you to enable/disable the CSV export option for this grid widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this grid widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format.

Image Export

This allows you to enable/disable the image export option for this grid widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

Bubble Map

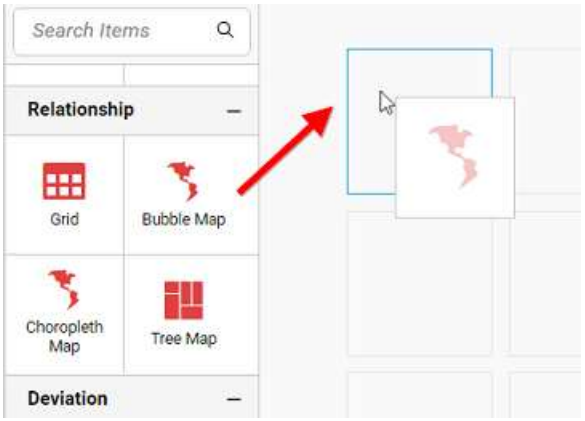
Bubble Map allows you to showcase quantitative values encoded through bubble size.

How to configure the table data to bubble map?

To plot a bubble map, a minimum requirement of 1 value and 1 shape is needed.

The following procedure illustrates data configuration of bubble Map.

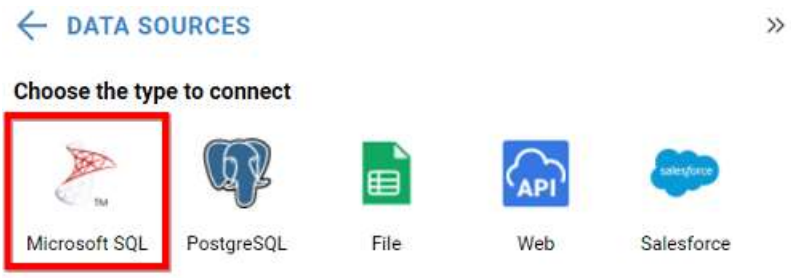
Drag and drop Bubble Map control icon from the tool box into design panel. You can find control in tool box by search.



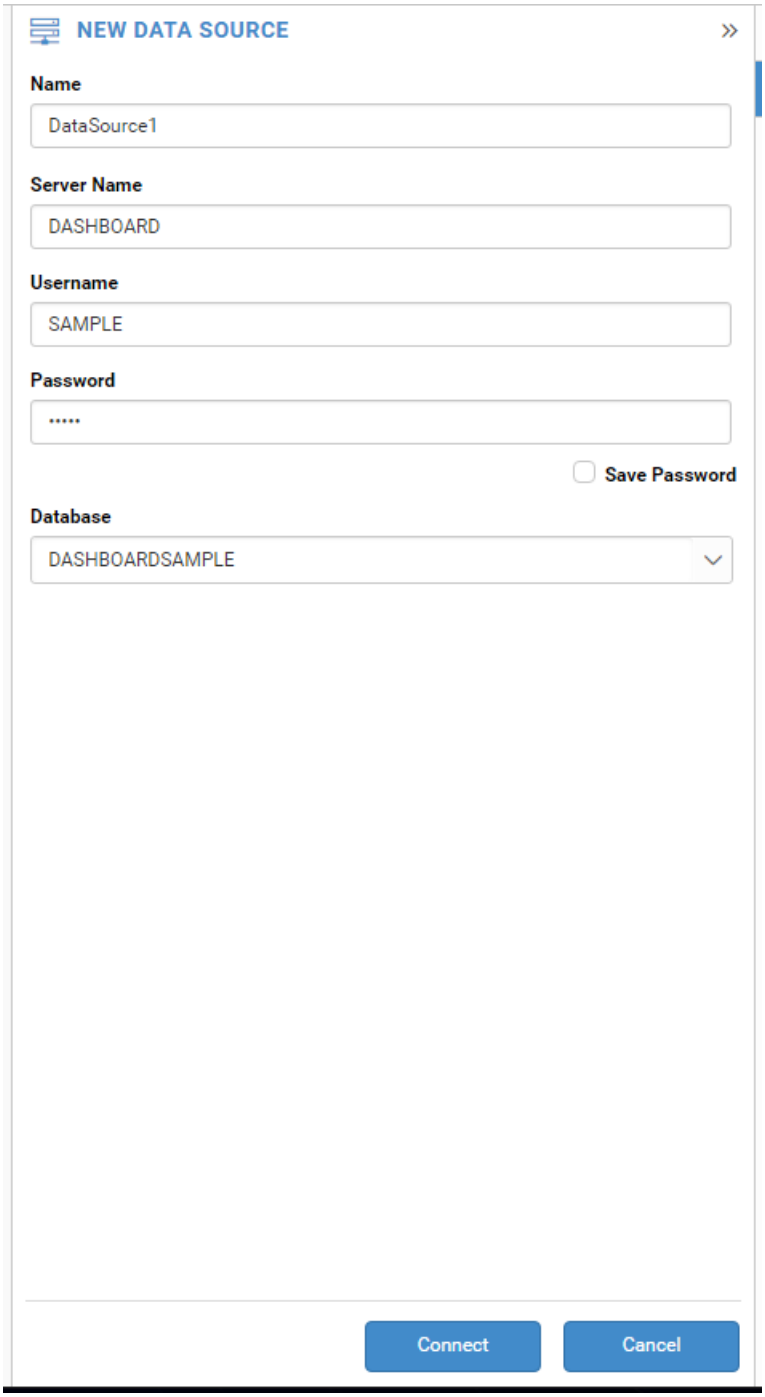
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

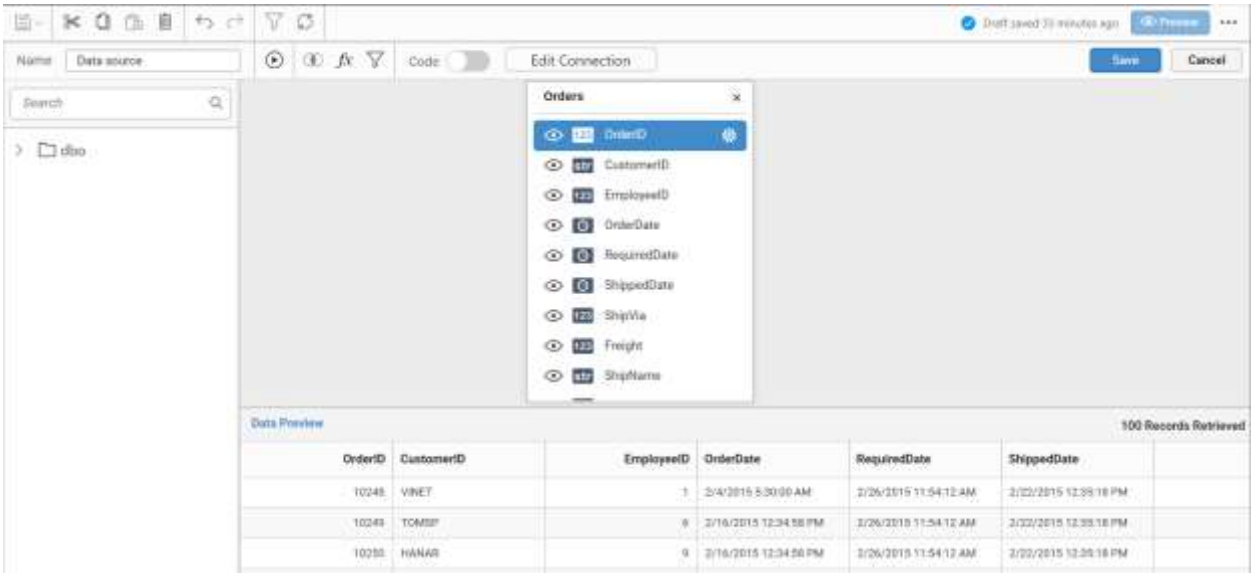
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



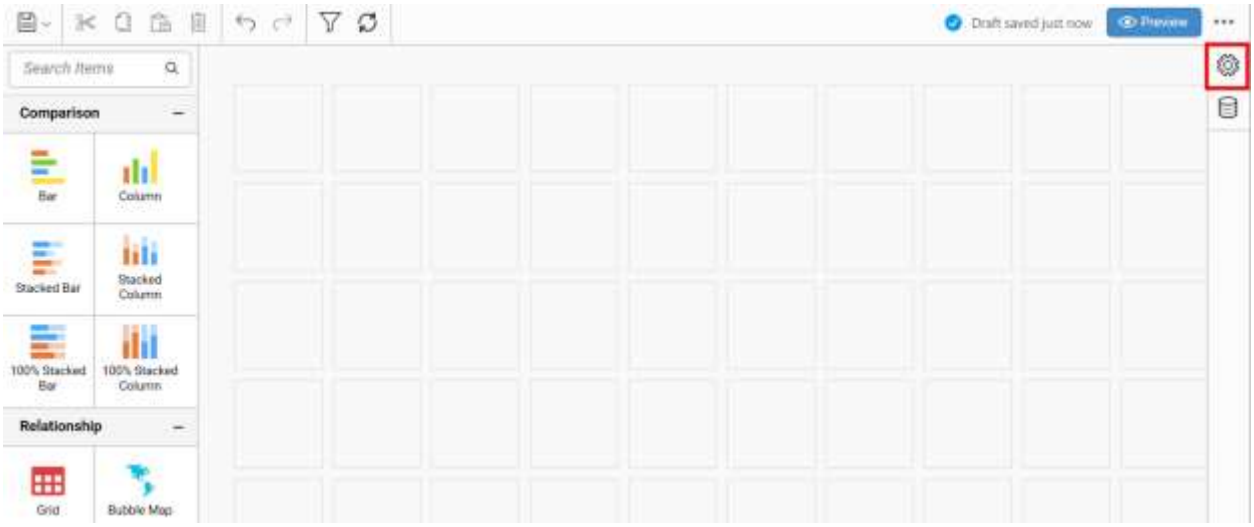
In the NEW DATA SOURCEconfiguration panel, fill the connection type and related details. Click Connect button

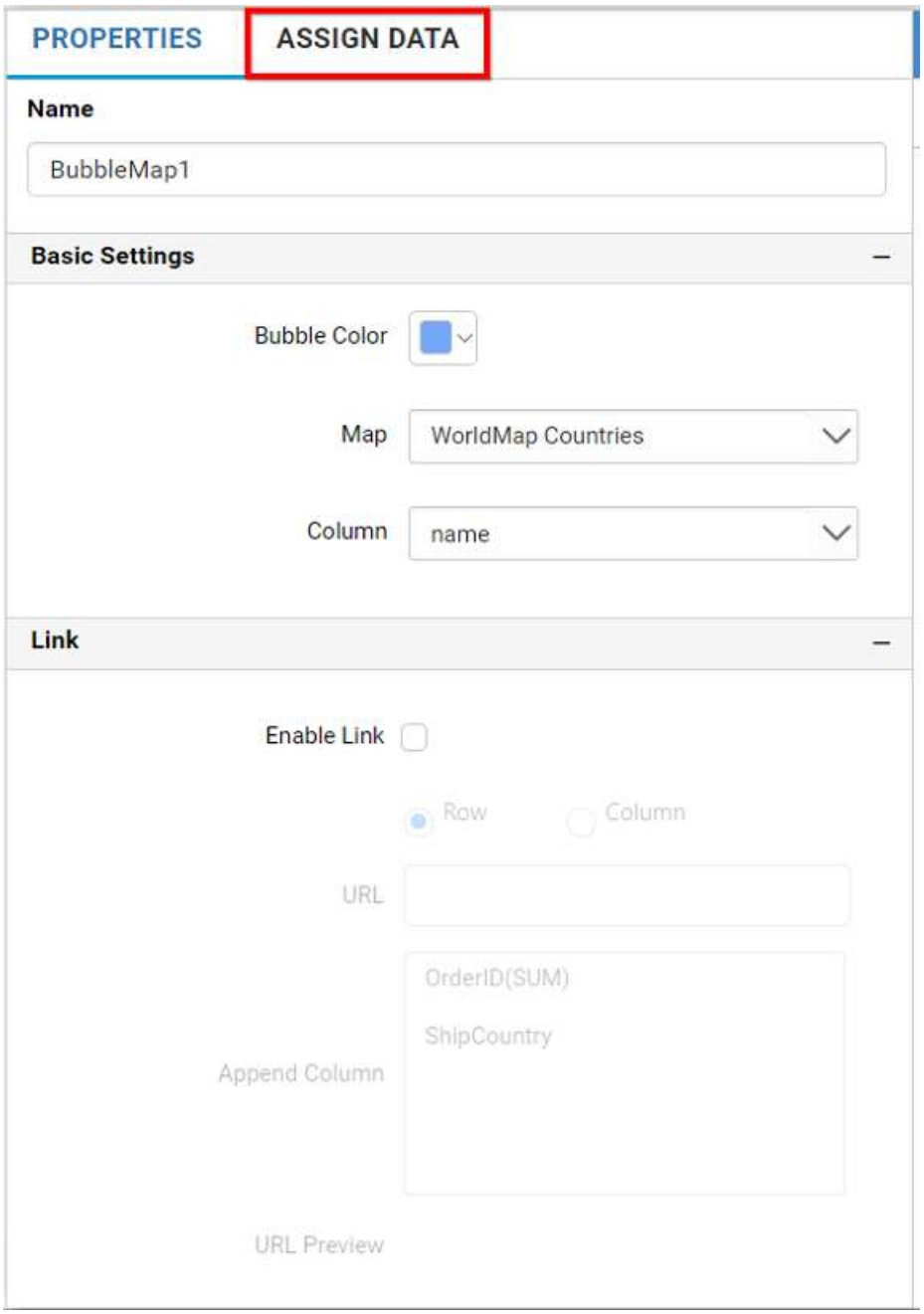


Drag your preferred table or view from the left pane from data design view, click Save button.

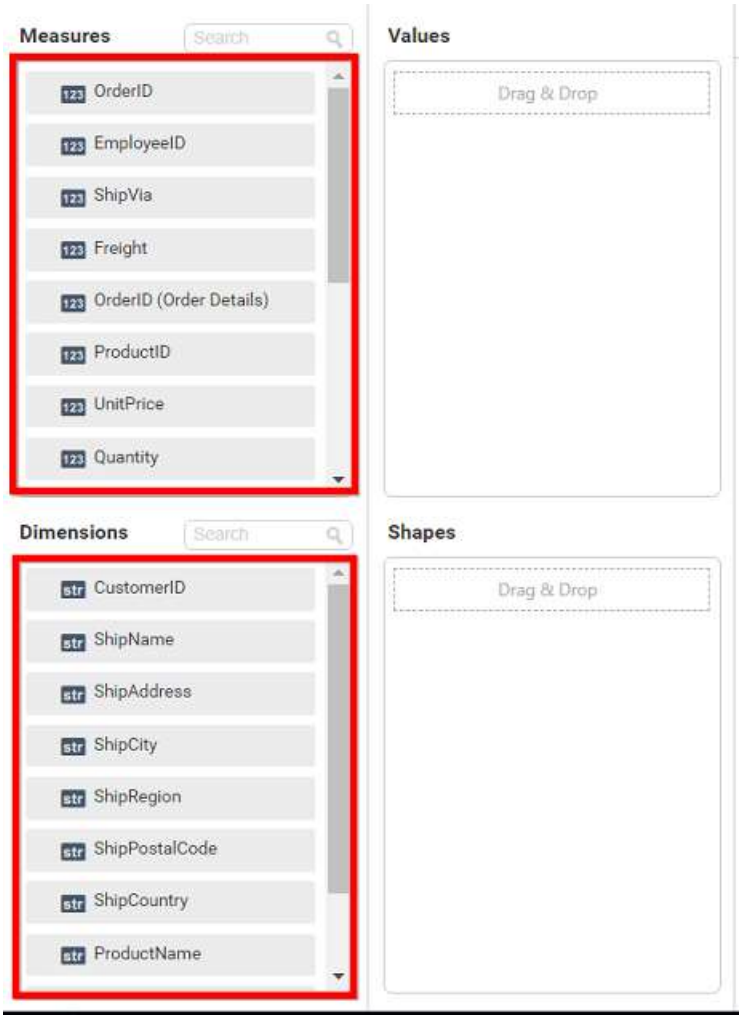


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.

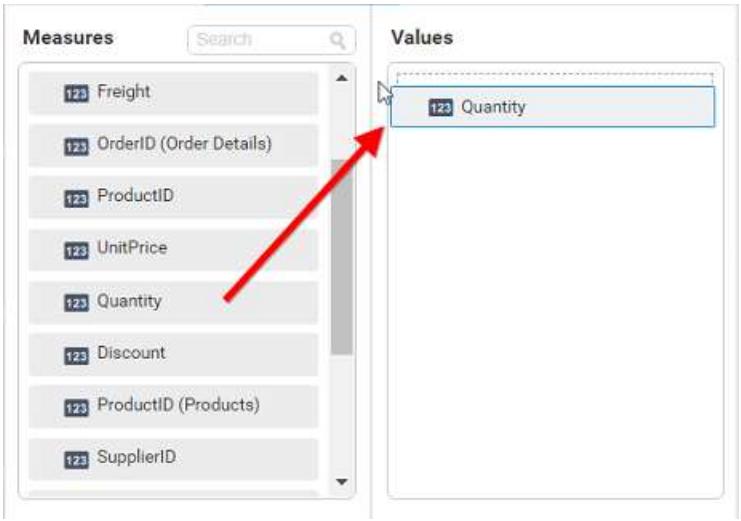




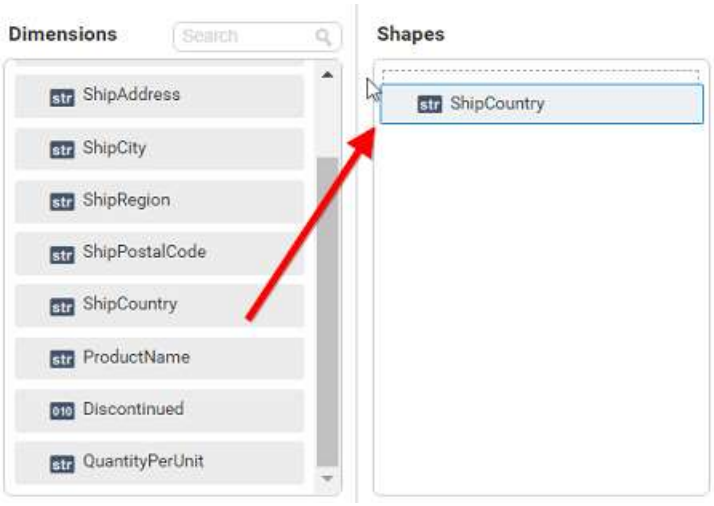
The data tab will be opened with available measures and dimensions from the connected data source



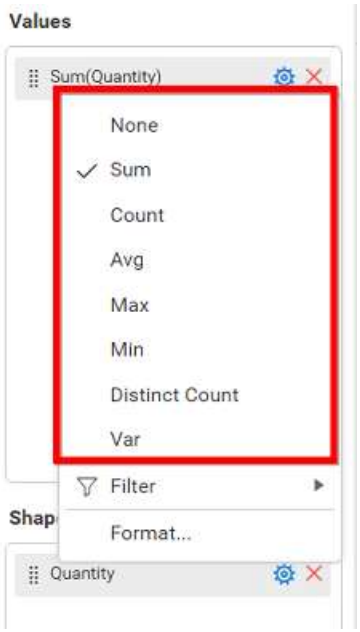
Bind column through drag and drop element from Measures section to Values.



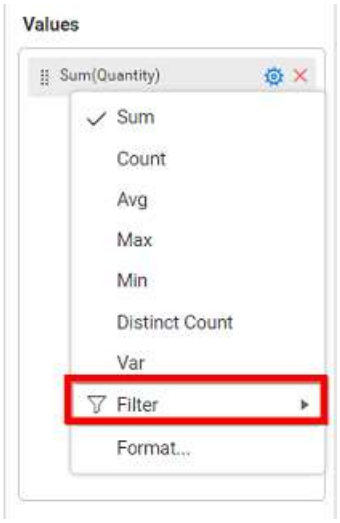
Drag and Drop the elements from sections to Shapes.



You can use the aggregation function to change the Values of the column.

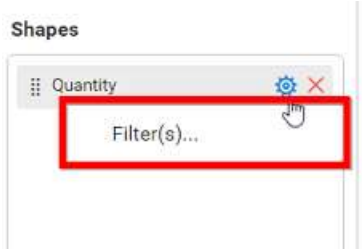


You can filter the data using Filter option. For more details, refer filter.



You can format the values by selecting the Format option. For more details, refer measure format

You can filter the data by selecting the Filter(s)... option. For more details, refer filter.



You can clear the filters by selecting the Show All Records options.

Here is an illustration,

How to format bubble map widget?

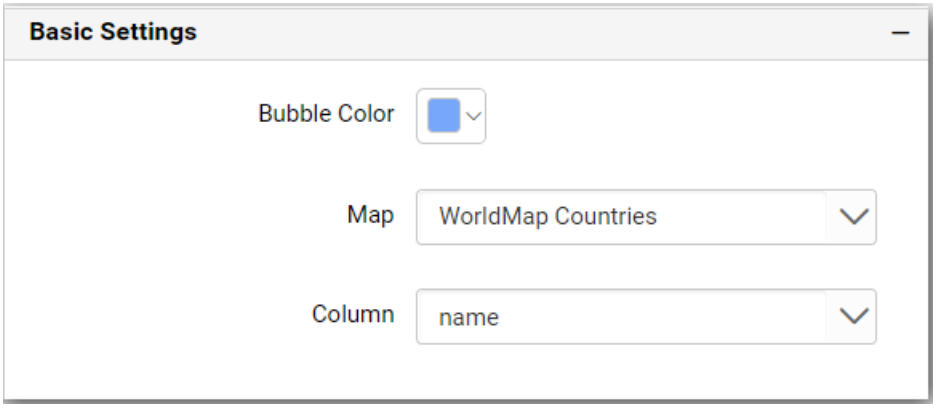
You can format the bubble map for better illustration of the view that you require, through the settings available in Properties tab.

General Settings

Name

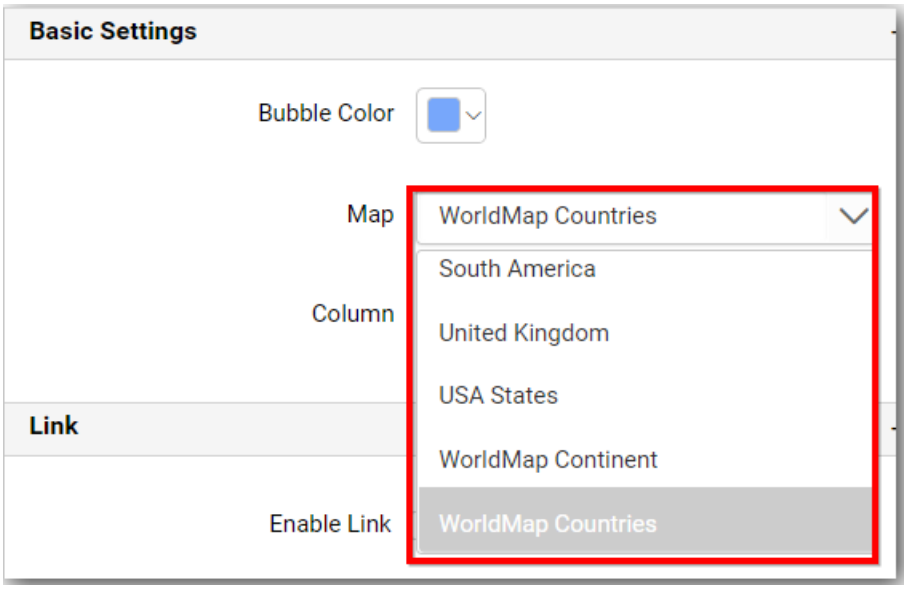
This allows you to set title for this bubble map widget.

Basic Settings



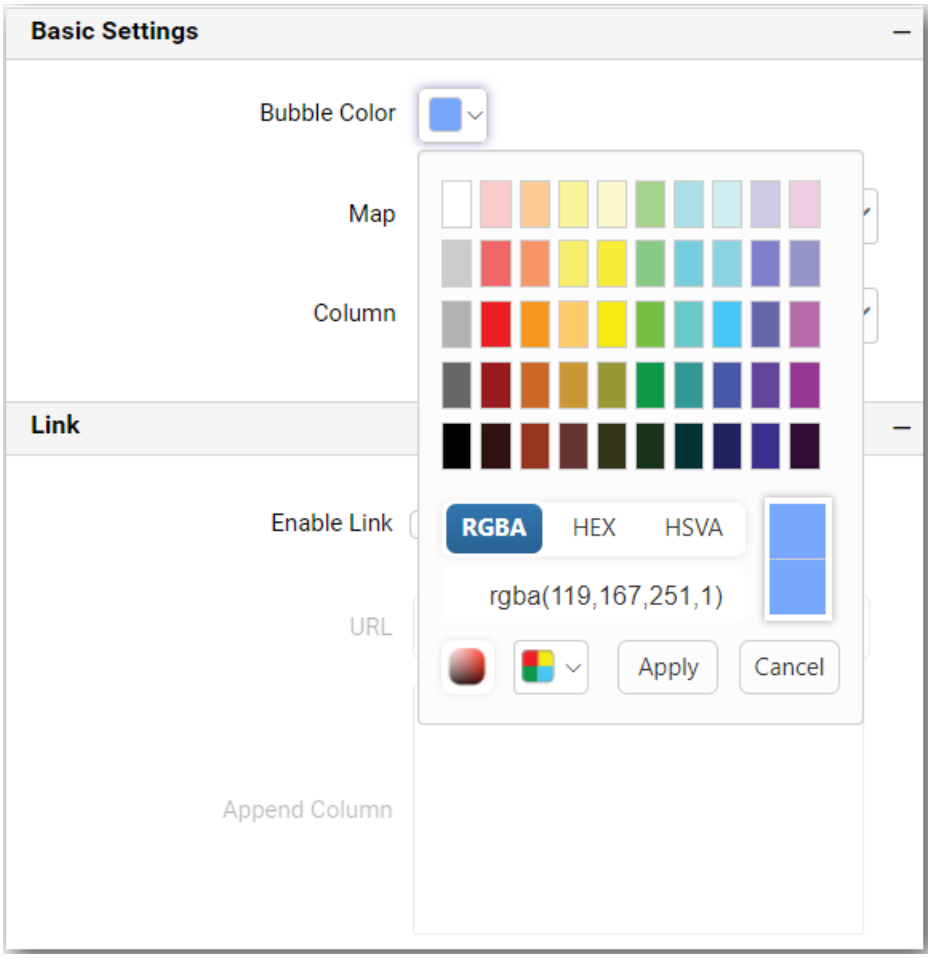
Map

You can choose map listed in the combo box for bubble map.



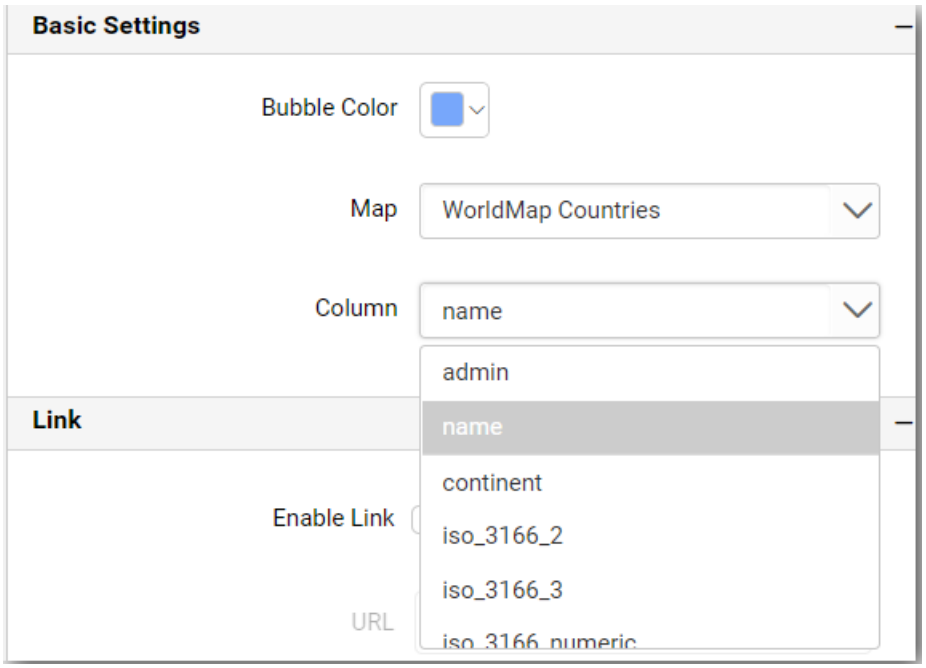
Bubble Color

You can customize the bubble color from the set of predefined color palette.

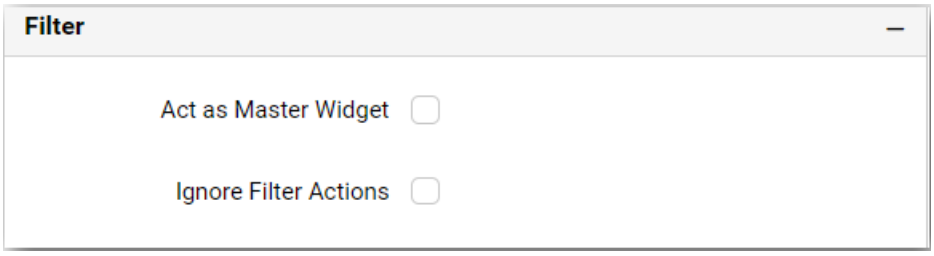


Column

You can choose from the list of column names supported by the loaded map shape file that can be mapped with the data source connected.



Filter



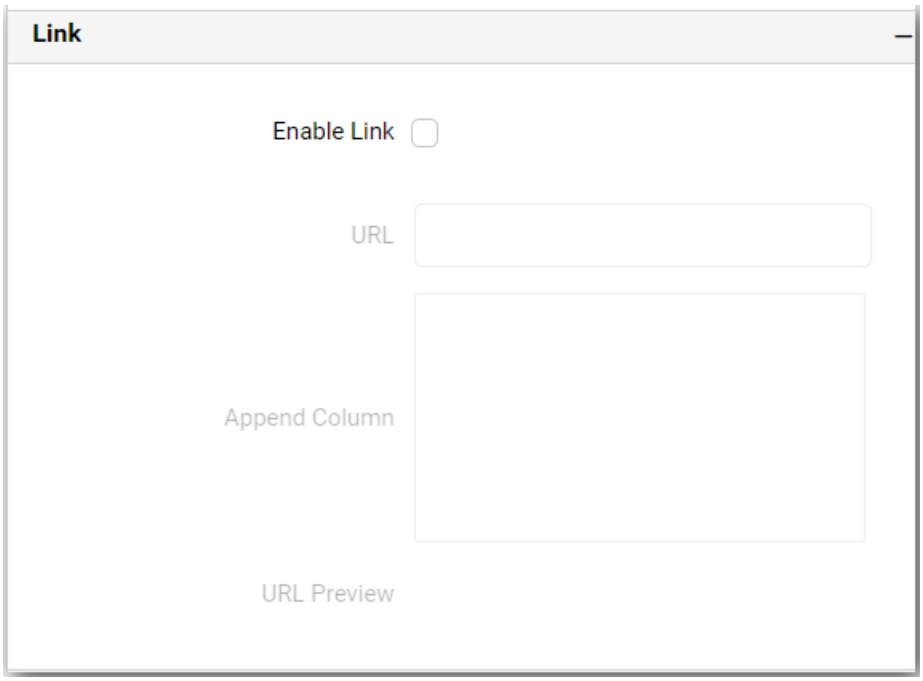
Act as Master Widget

This allows you to define this bubble map widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

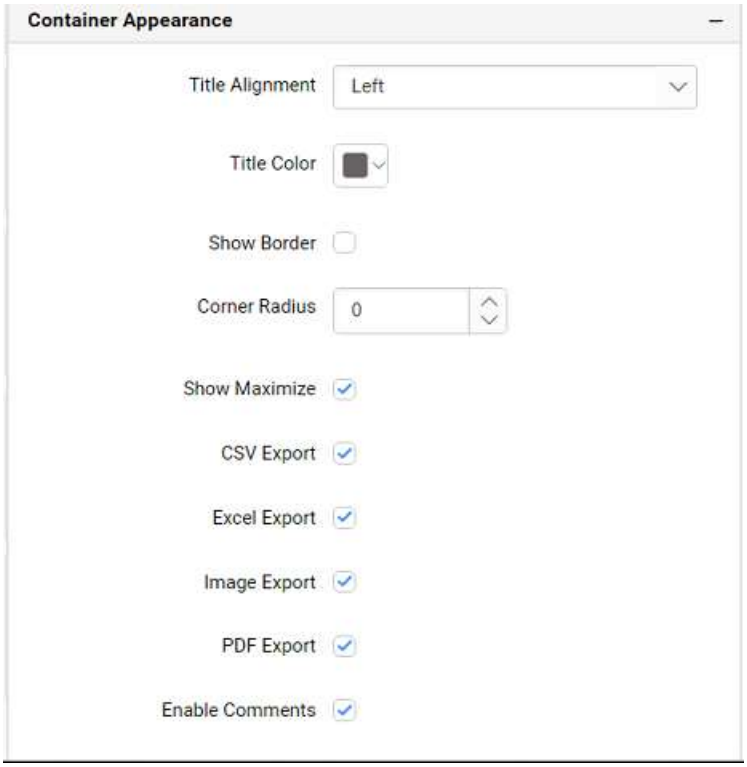
This allows you to define this bubble map widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if **Show Border** property is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this bubble map widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the bubble map widget.

CSV Export

This allows you to enable/disable the CSV export option for this bubble map widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this bubble map widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format.

Image Export

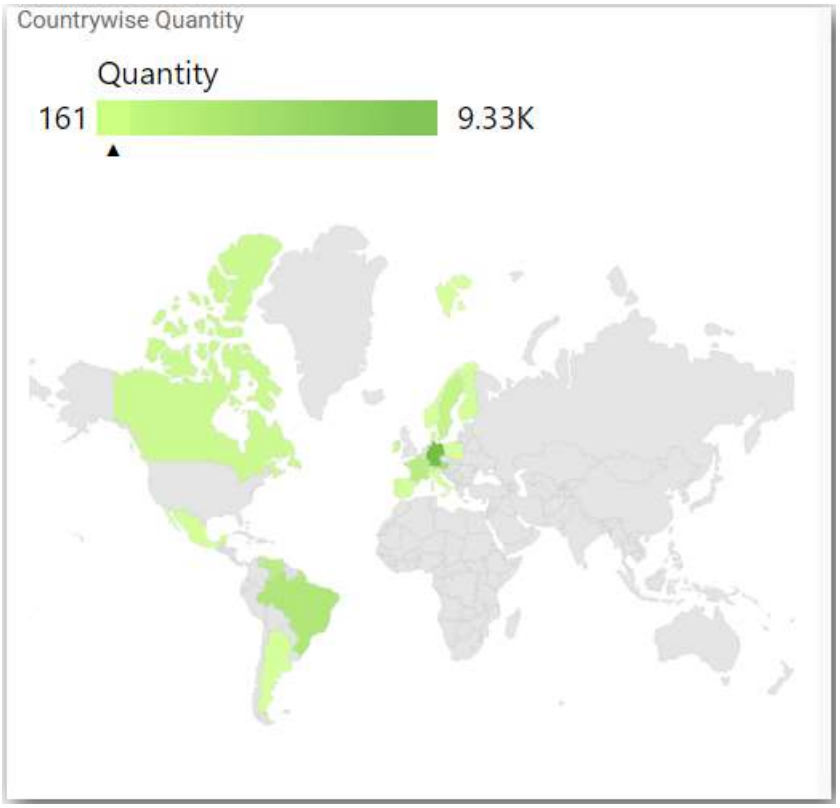
This allows you to enable/disable the image export option for this bubble map widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

Choropleth Map

Choropleth Map allows you to showcase quantitative values encoded through color scale.



How to configure the table data into choropleth map?

To plot a choropleth map, a minimum requirement of 1 value and 1 shape is needed. Dropping a dimension will display each region split by each of its item.

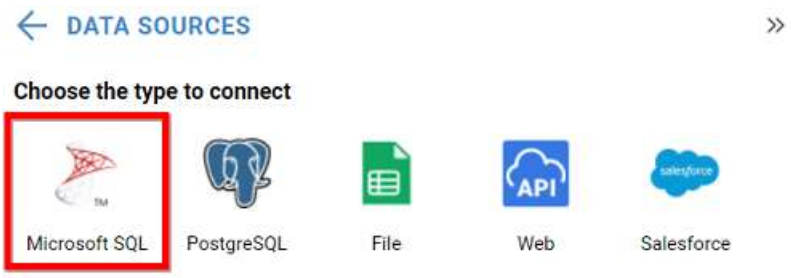
The following procedure illustrates data configuration of choropleth Map.

Drag and drop Choropleth Map control icon from the tool box into design panel. You can find control in tool box by search.

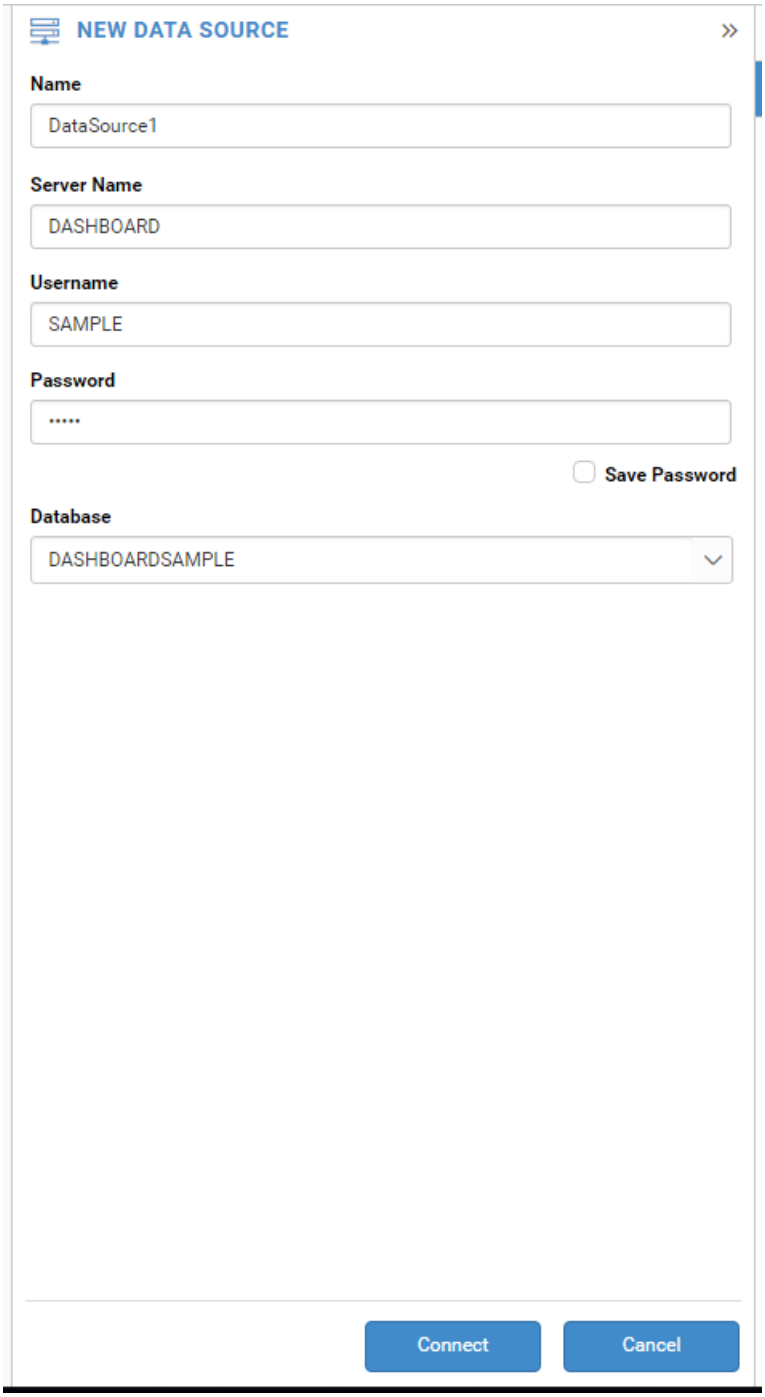
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

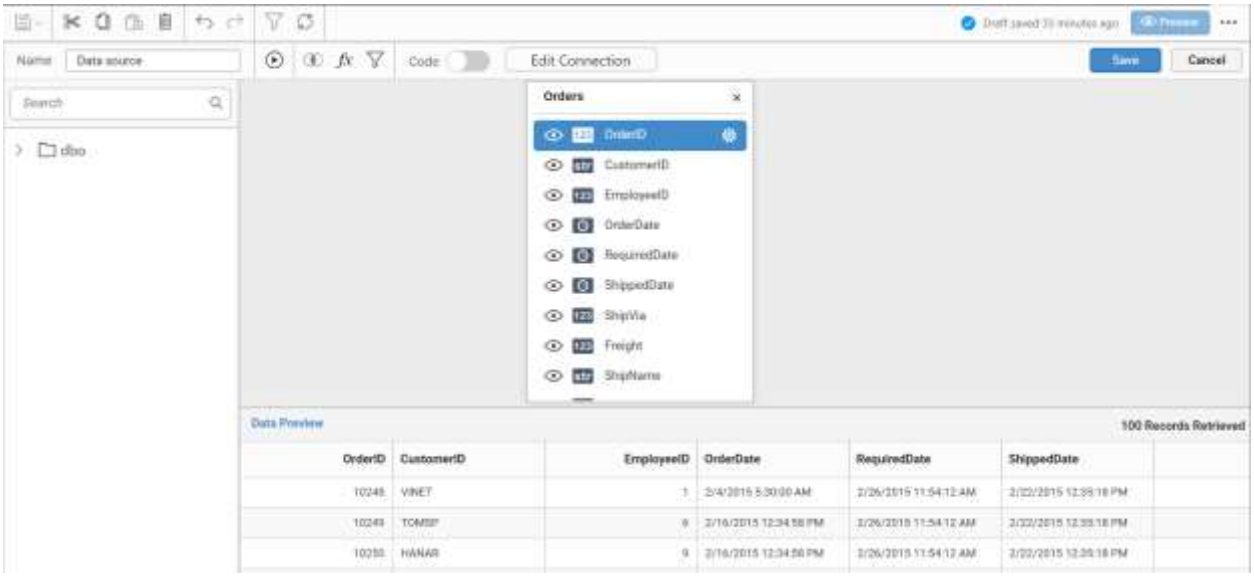
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



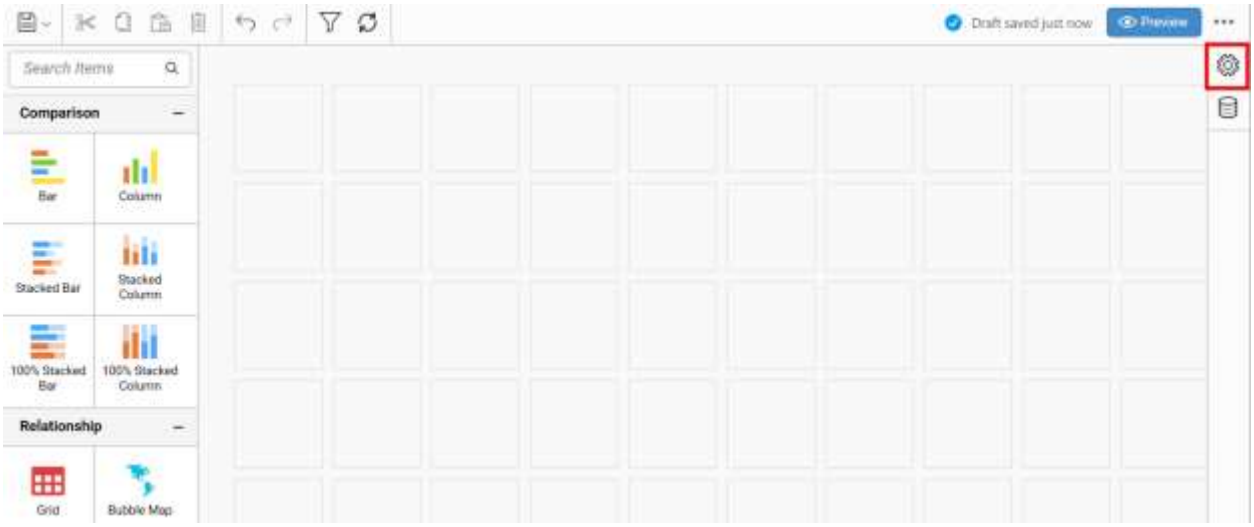
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

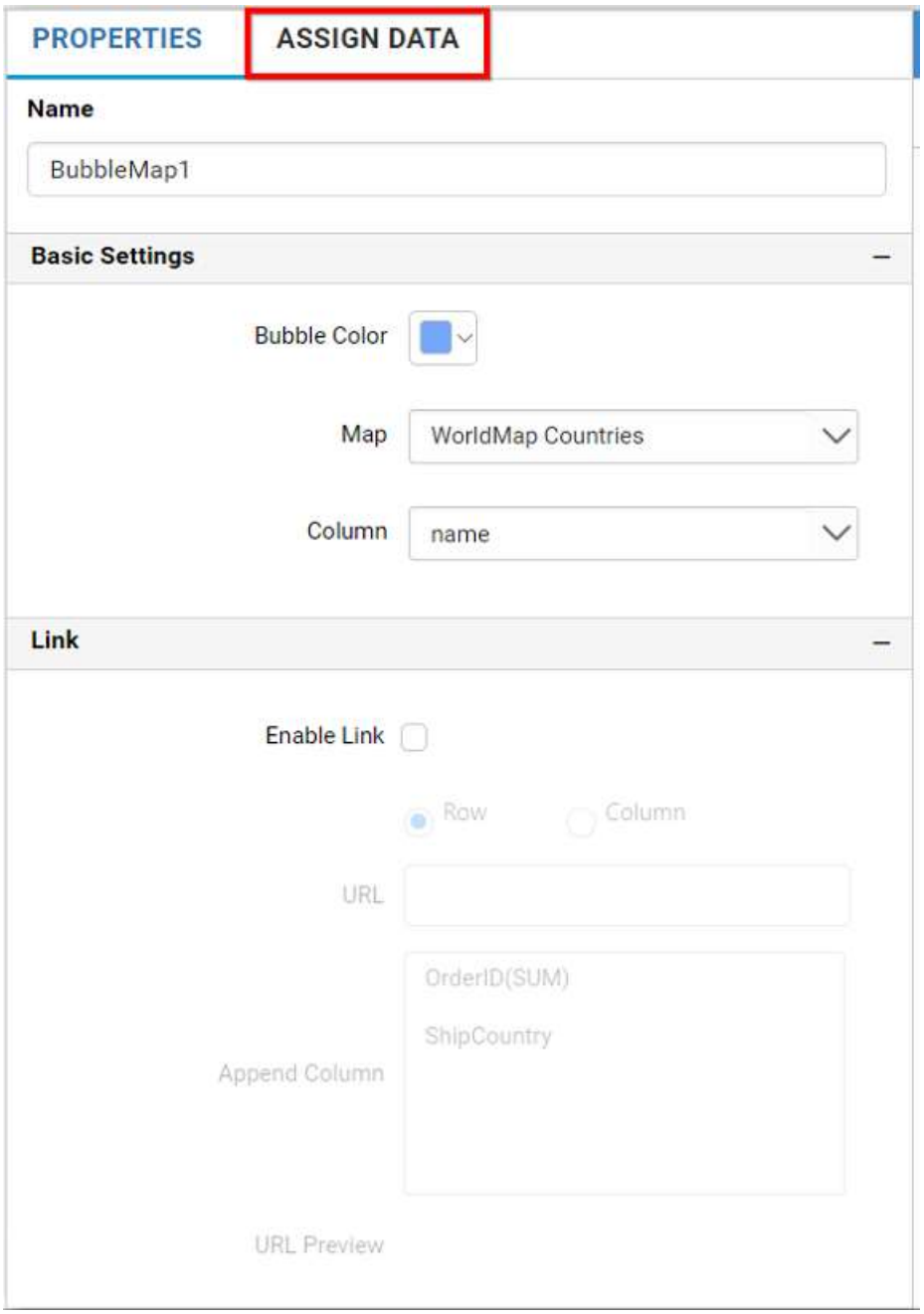


Drag your preferred table or view from the left pane from data design view, click Save button.

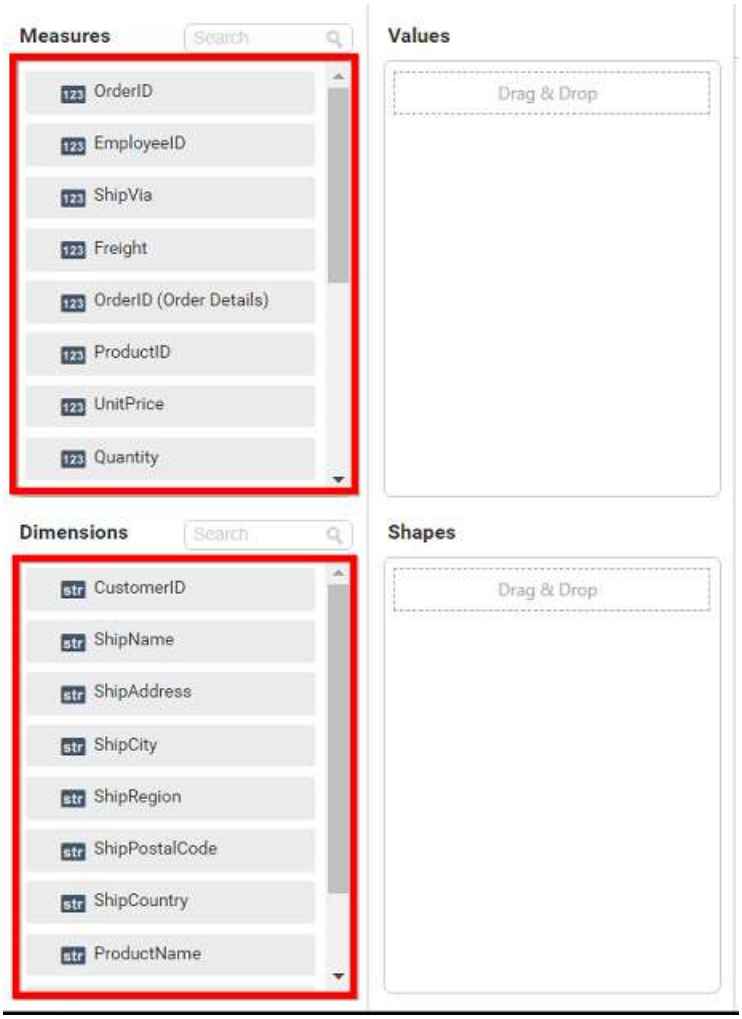


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.

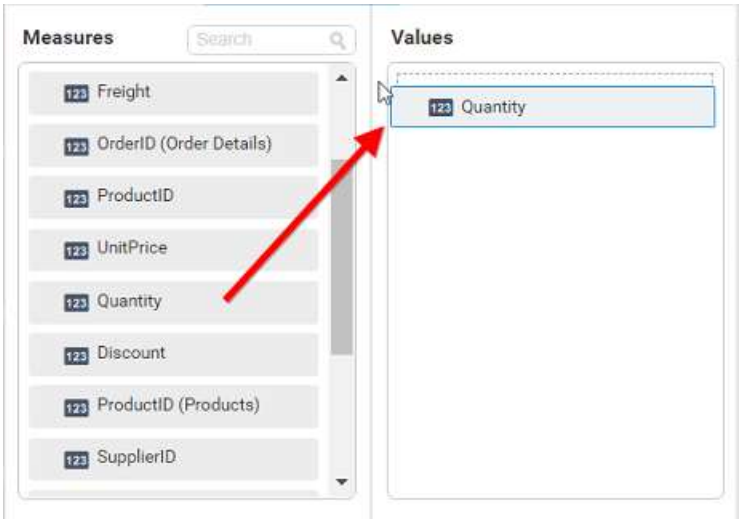




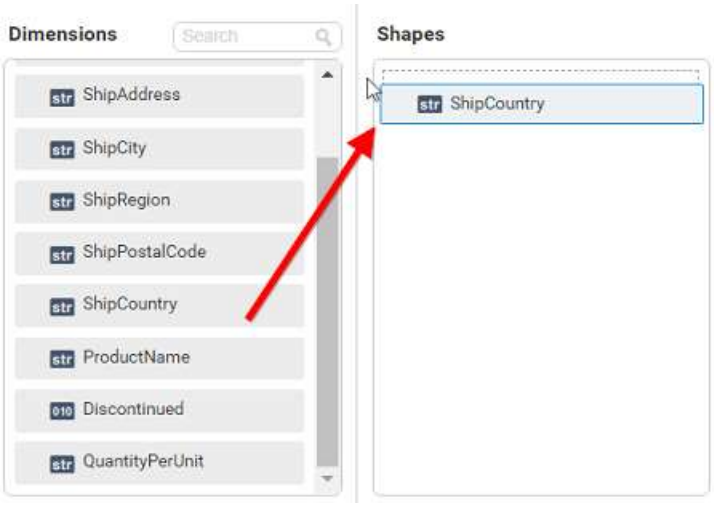
The data tab will be opened with available measures and dimensions from the connected data source



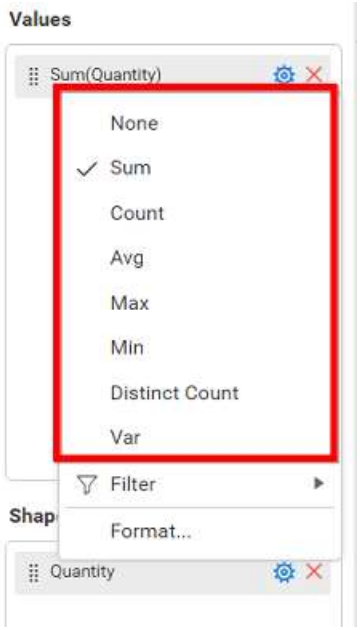
Bind column through drag and drop element from Measures section to Values.



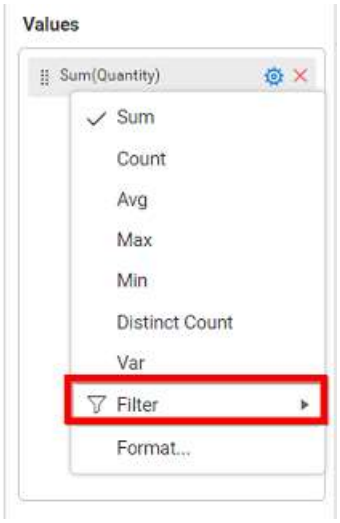
Drag and Drop the elements from sections to Shapes.



You can use the aggregation function to change the Values of the column.

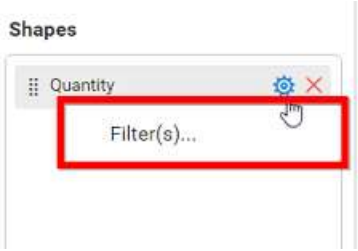


You can use the Filter option to filter the data. For more details, refer filter.



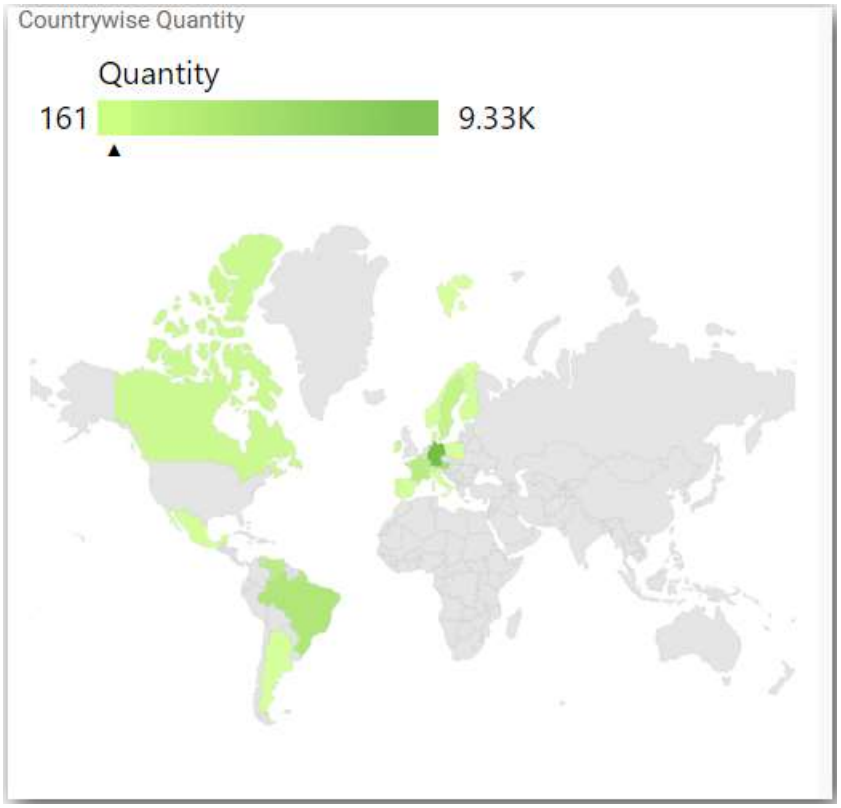
You can format the values by selecting the Format option. For more details, refer measure format

You can use the Filter(s) option to filter the data. For more details, refer filter.



You can clear the filters by selecting the Show All Records options.

Here is an illustration,



How to format choropleth map?

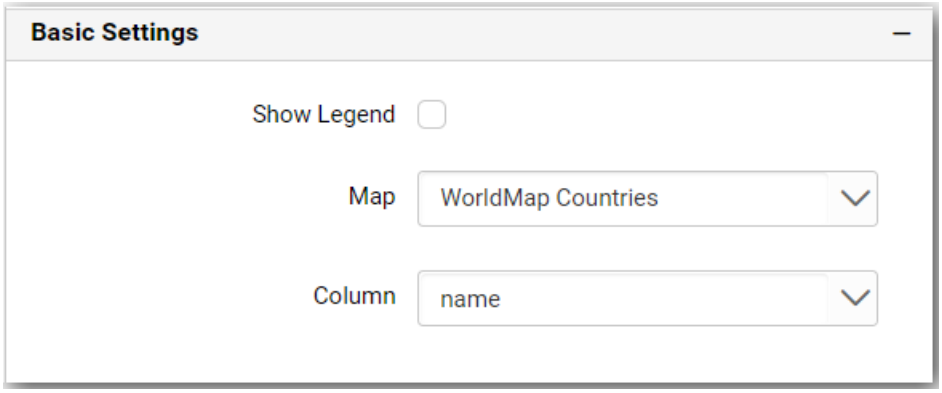
You can format the choropleth map for better illustration of the view that you require, through the settings available in Properties tab.

General Settings

Name

This allows you to set title for this choropleth map widget.

Basic Settings

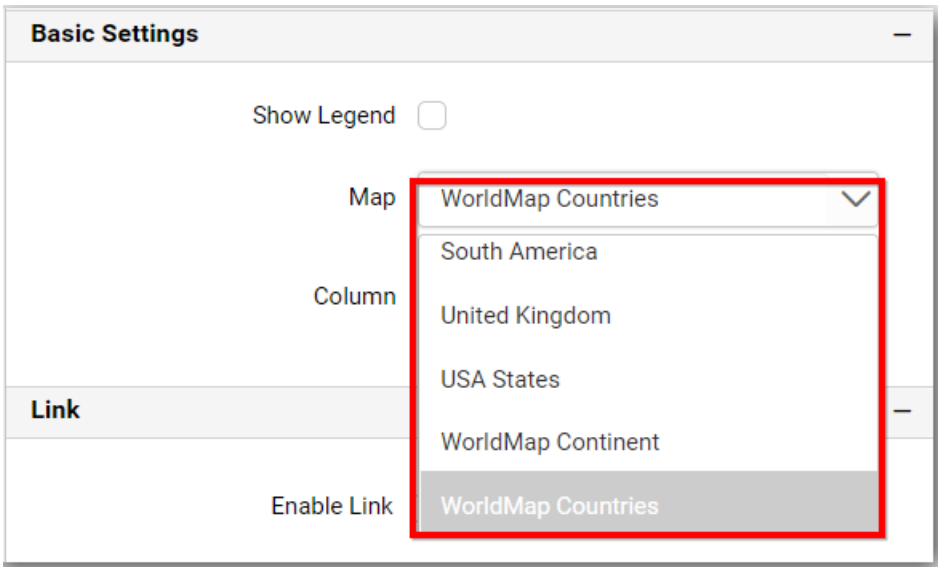


Show Legend

You can toggle the visibility of legend in choropleth map.

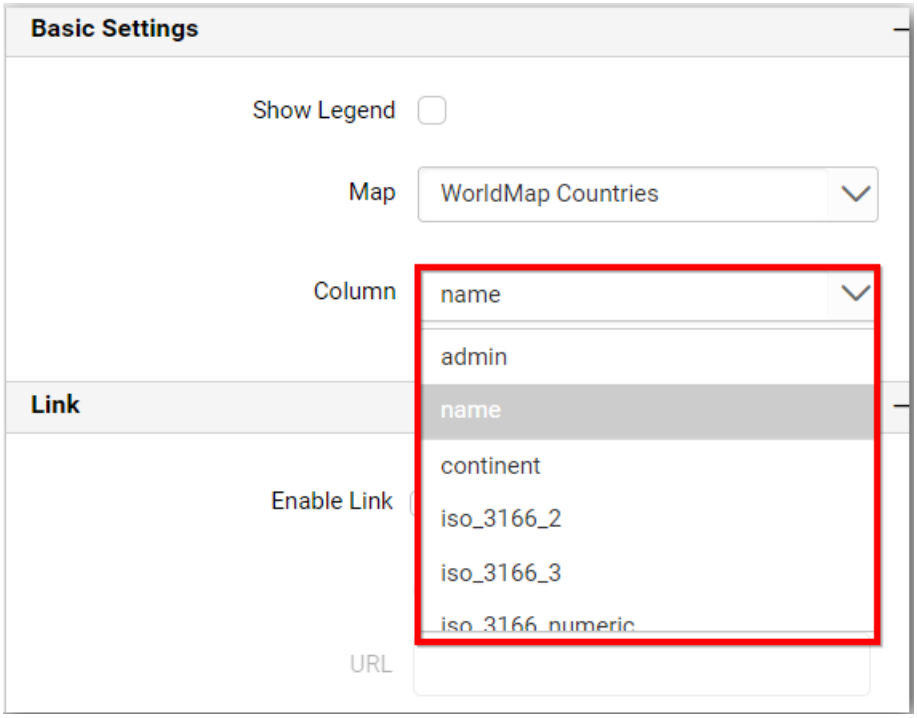
Map

You can choose map listed in the combo box in bubble map.

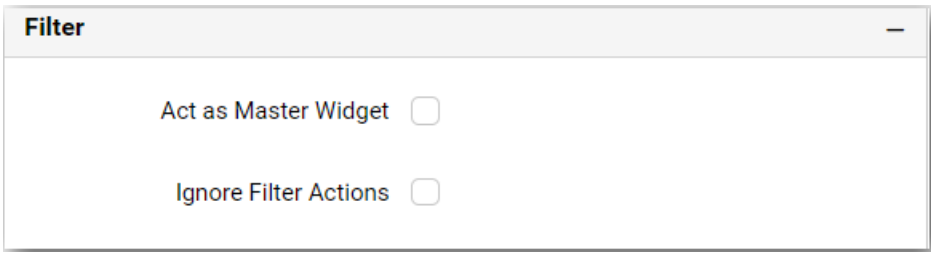


Column

You can choose from the list of field names supported by the loaded map shape that can be mapped with the data source connected.



Filter



Act as Master Widget

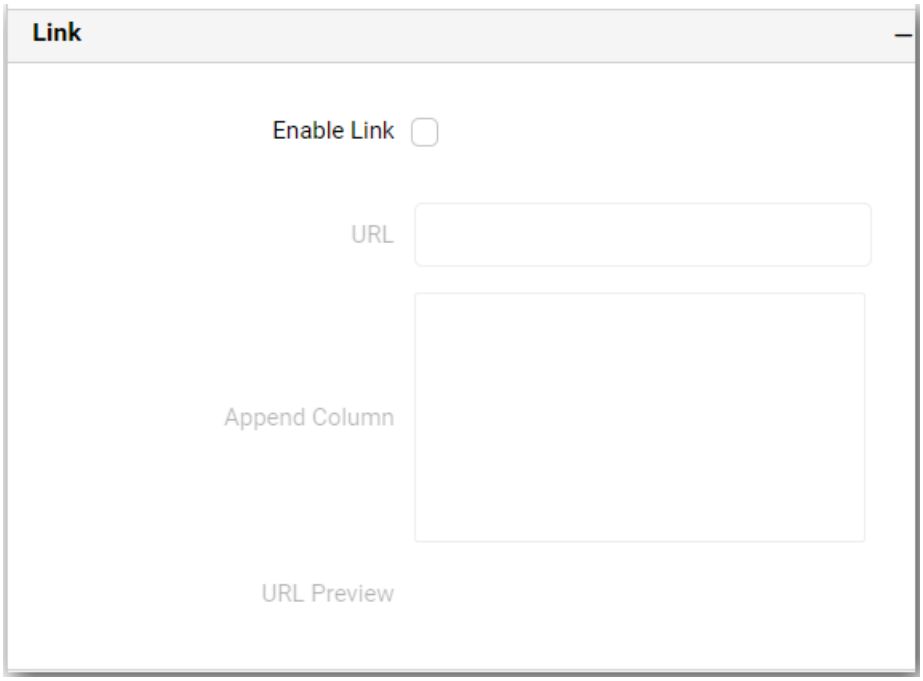
This allows you to define this choropleth map widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

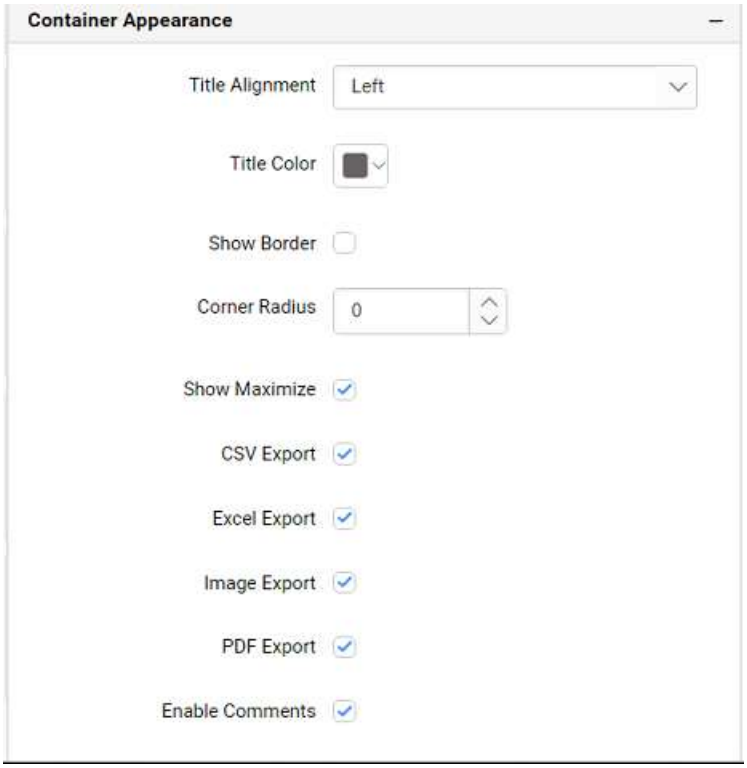
This allows you to define this choropleth map widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link

You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.



Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to apply the specified radius to the widget corners, if **Show Border** property is enabled. Value can be between 0 and 10.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this choropleth map widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the choropleth map widget.

CSV Export

This allows you to enable/disable the CSV export option for this choropleth map widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this choropleth map widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format.

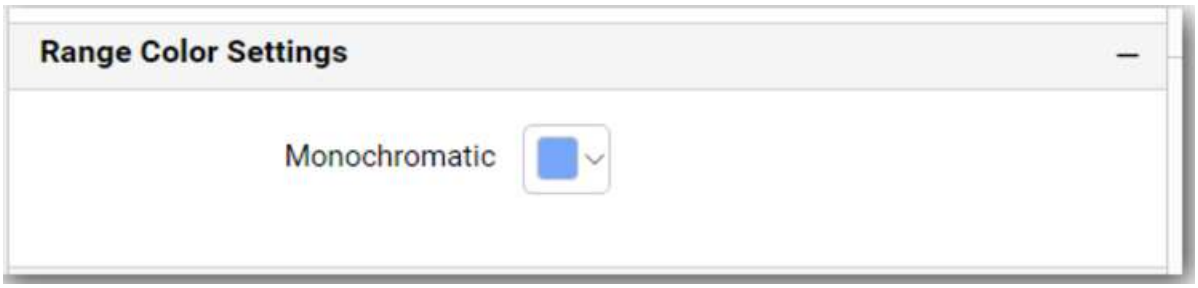
Image Export

This allows you to enable/disable the image export option for this choropleth map widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

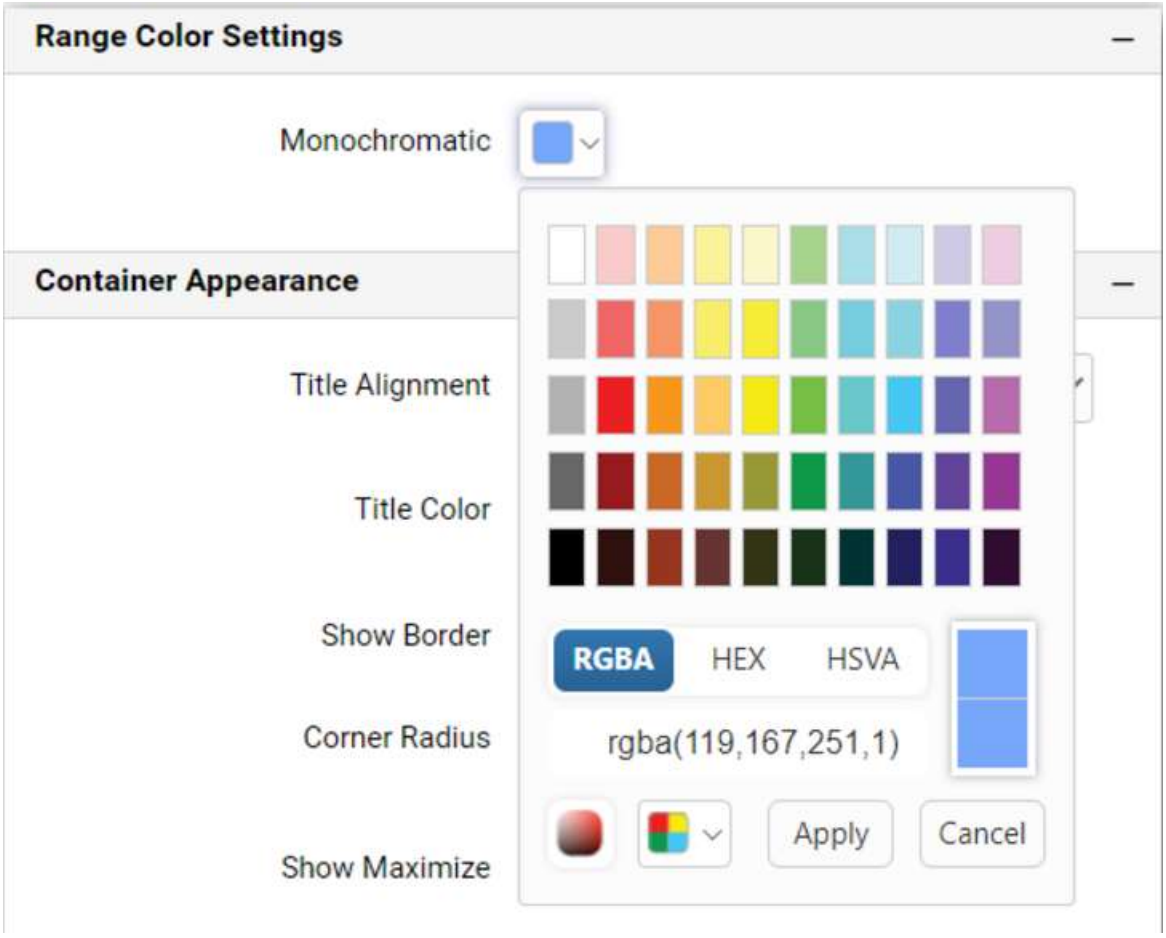
This allows you to enable/disable comment for dashboard widget. For more details refer here

Range Color Settings

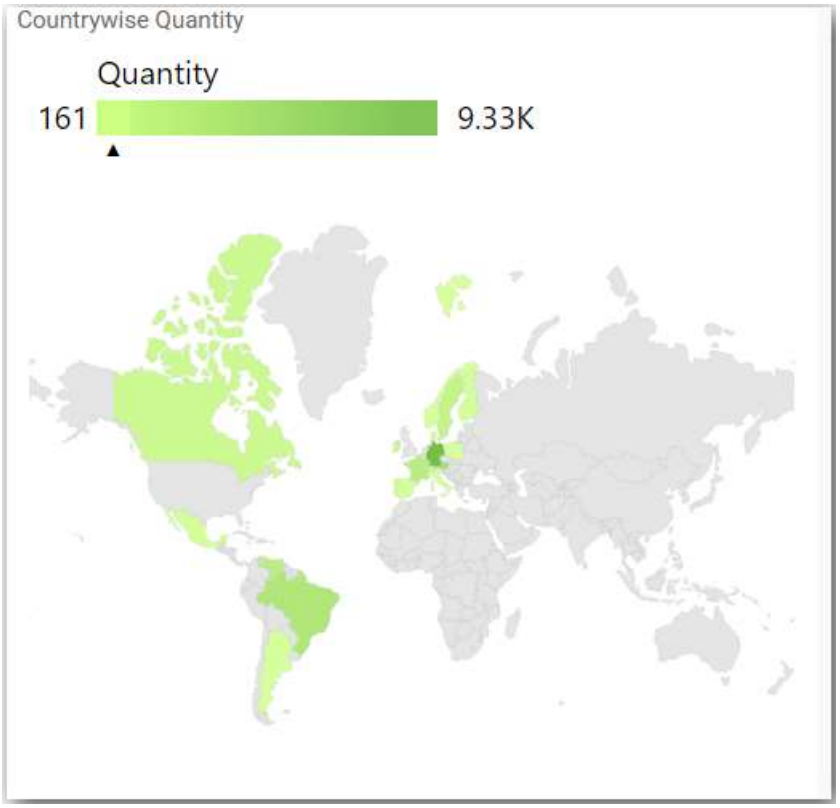


Monochromatic

You can configure a single color palette whose saturation differs based on the value density.

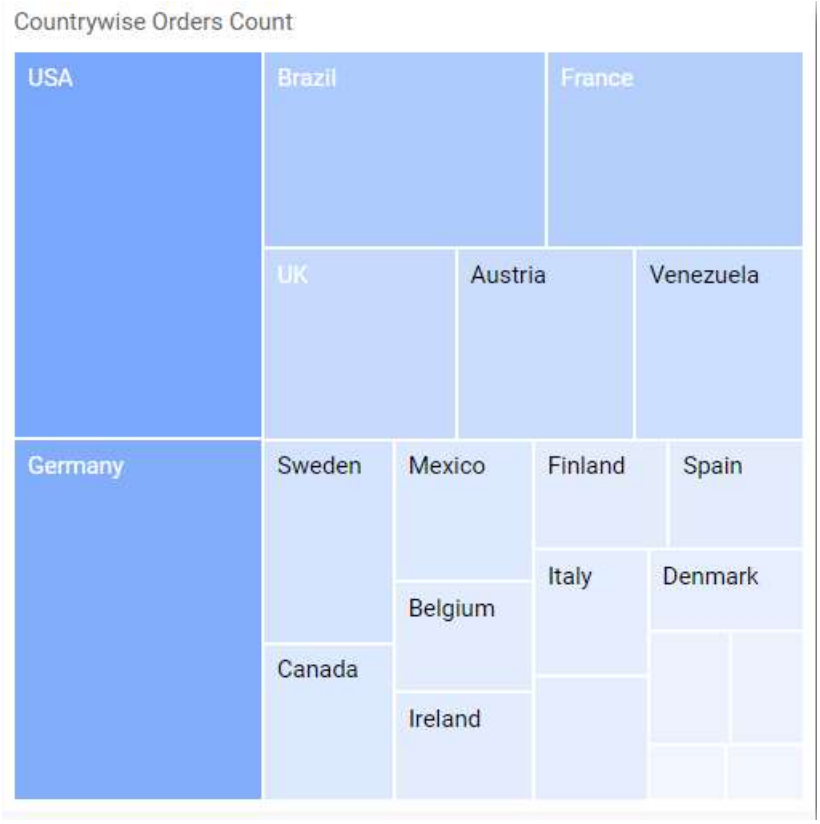


Here is an illustration,



Tree Map

Tree Map allows you to visualize large data through its proportional shelves and color scales.

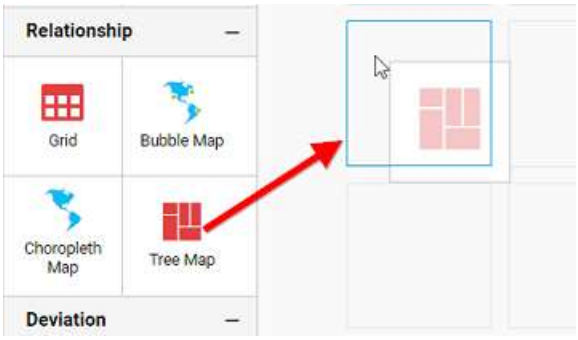


How to configure the table data to tree map widget?

To showcase a tree map, a minimum requirement of 1 values and 1 groups by field is needed.

The following procedure illustrates data configuration of tree Map.

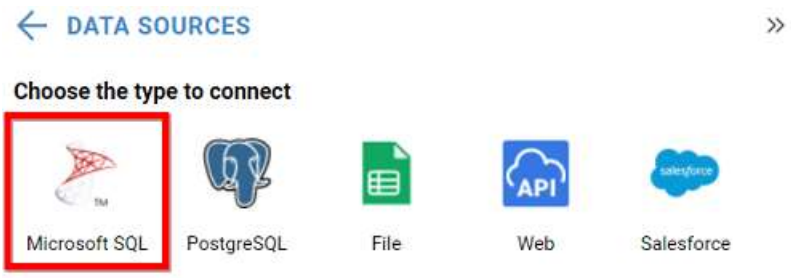
Drag and drop Tree Map control icon from the tool box into design panel. You can find control in tool box by search.



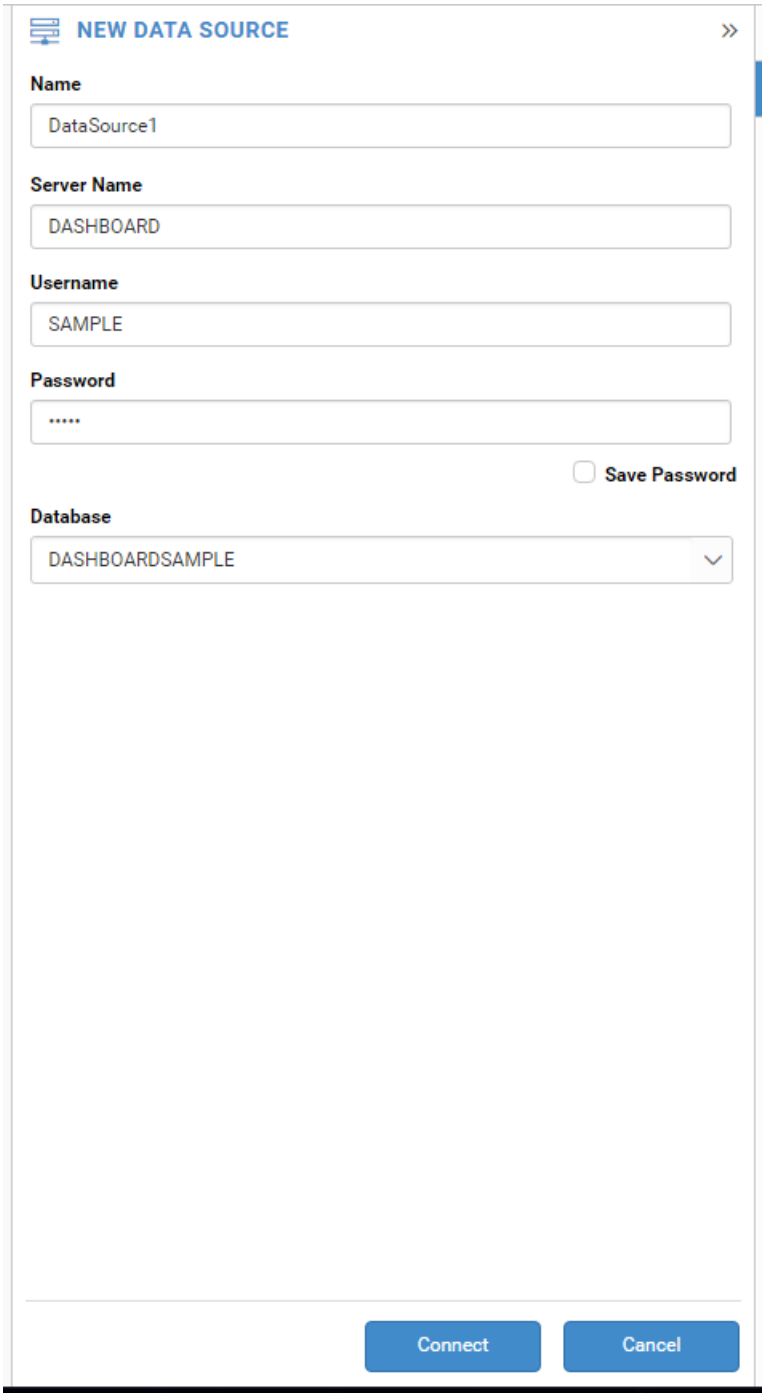
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

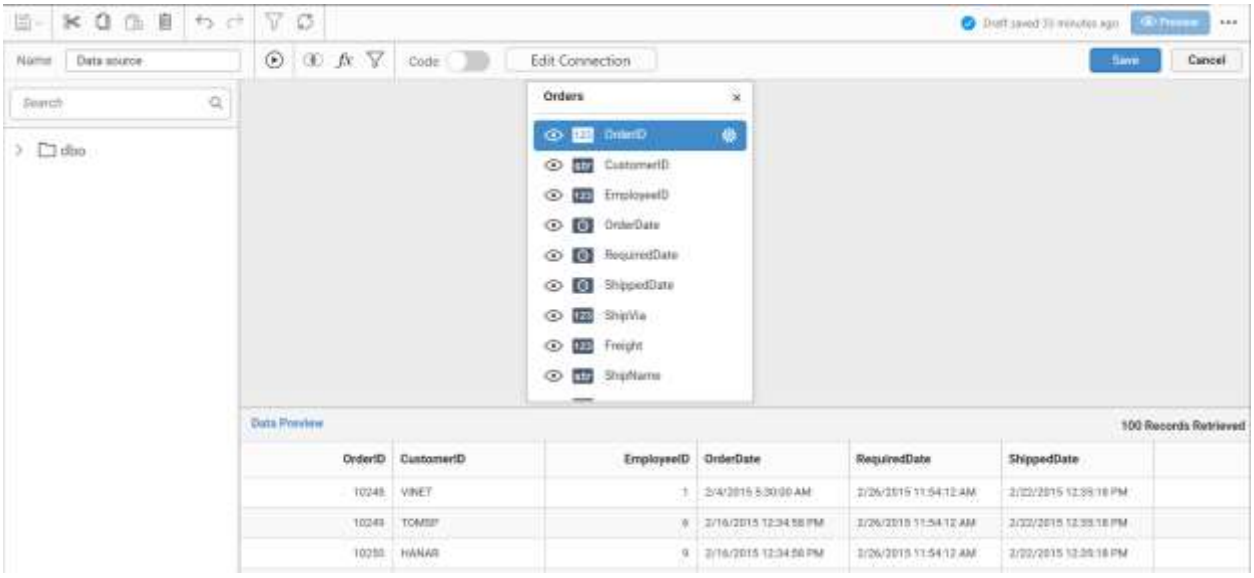
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



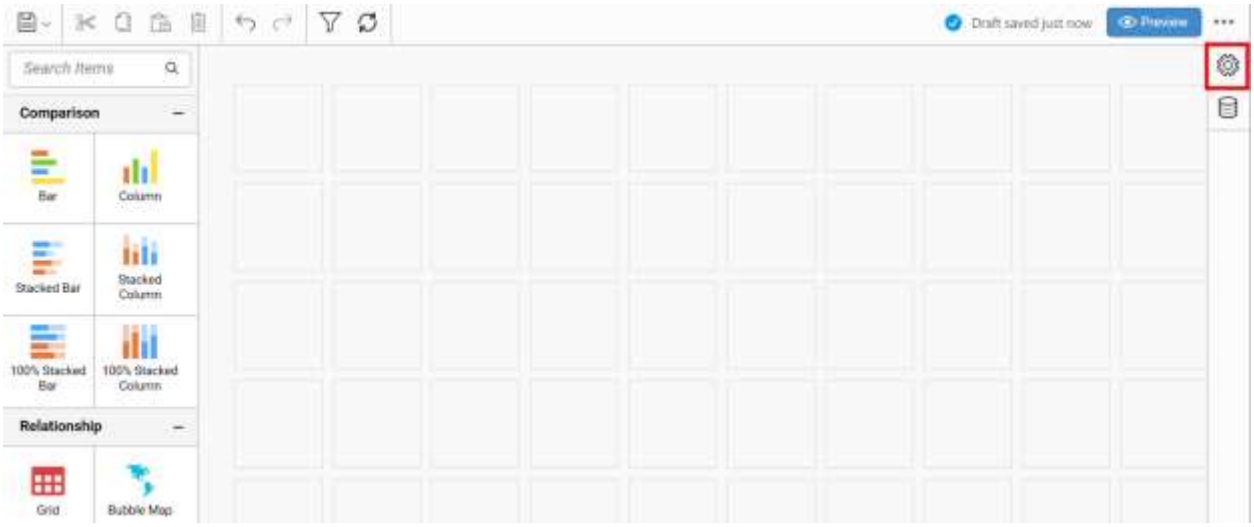
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

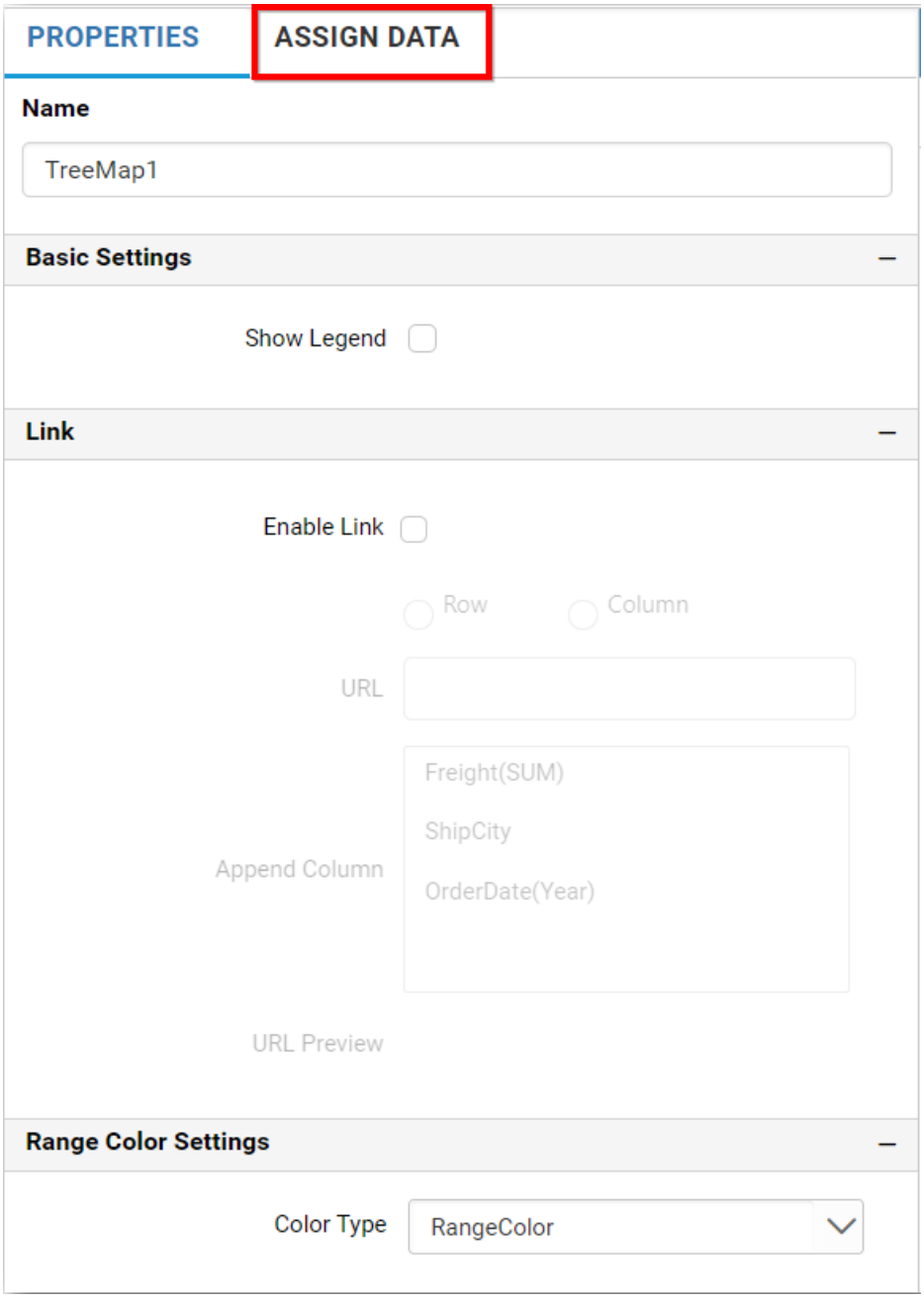


Drag your preferred table or view from the left pane from data design view, click Save button.

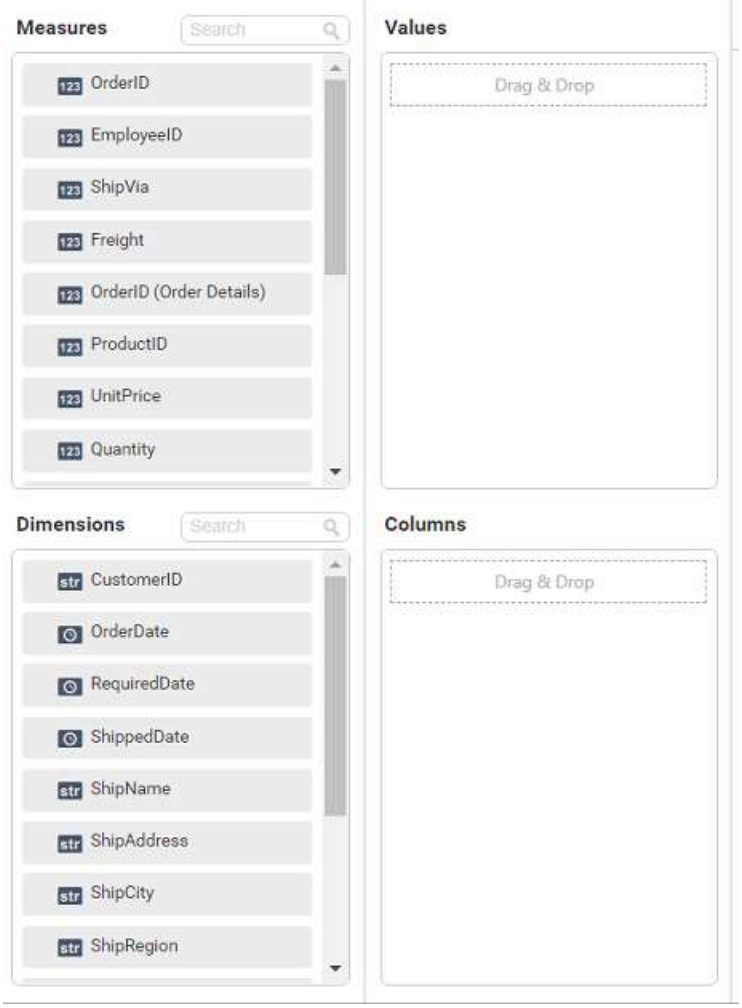


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.

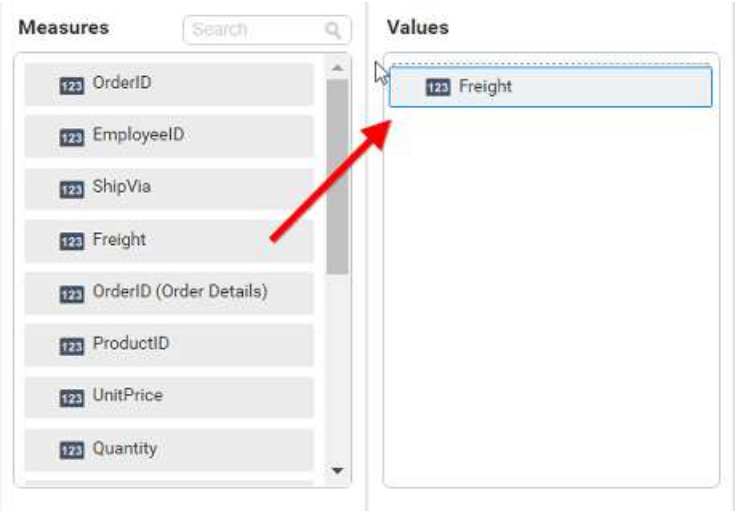




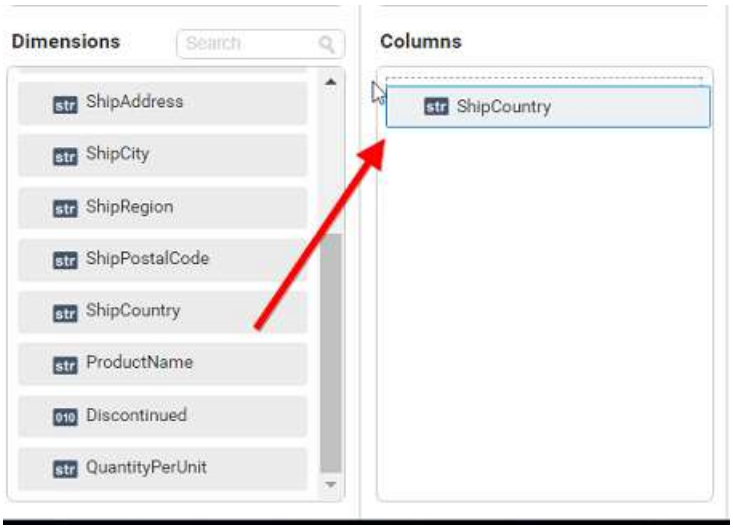
The data tab will be opened with available measures and dimensions from the connected data source



Bind column through drag and drop element from sections to Values.

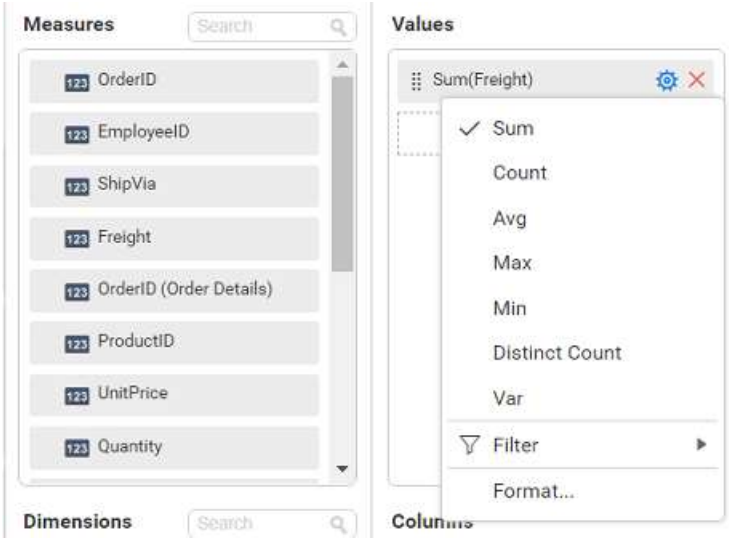


Drag and Drop the elements from sections to Columns.

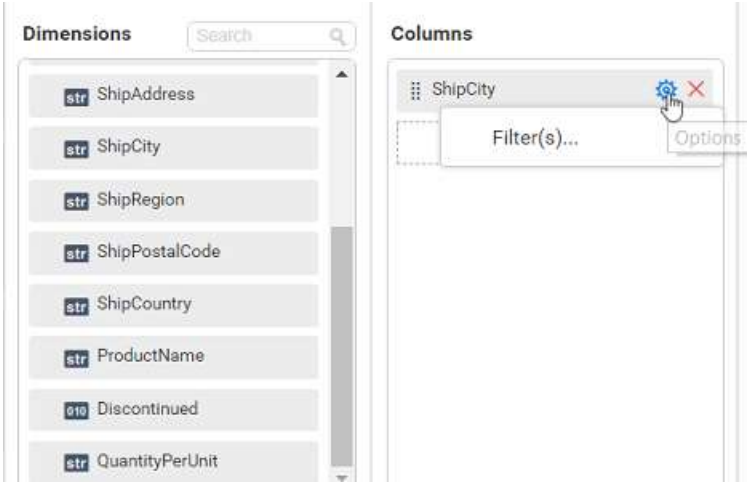


You can use aggregate function to change the Values of the column.

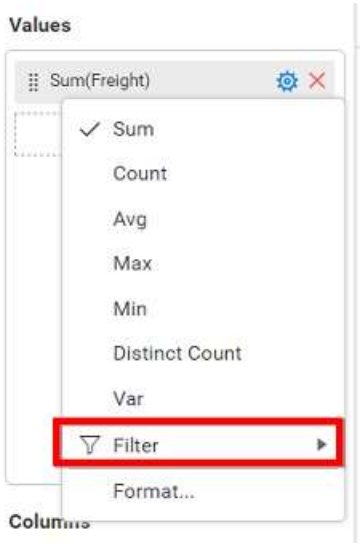
If measure column is binded following function list will be shown,



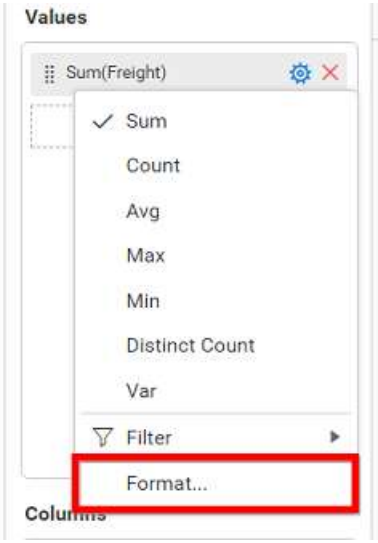
If dimension column is binded following function list will be shown,



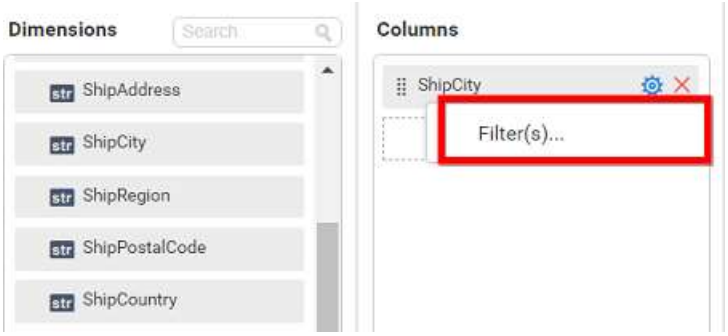
You can use Filter option to filter the data by specifying the filter condition. For more details, refer filter.



You can format the values by selecting the Format option. For more details, refer measure format

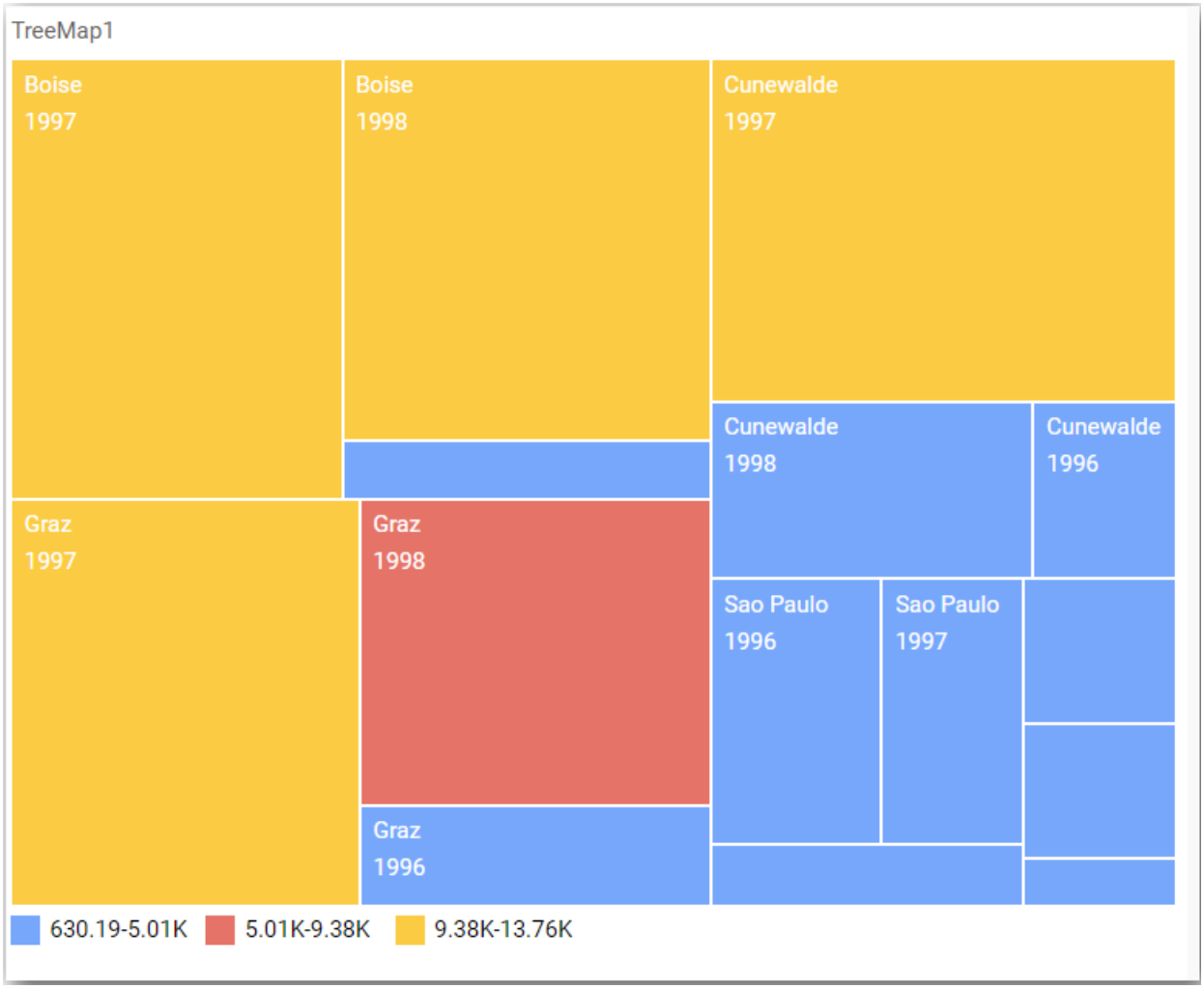


For Columns field, you can use the filters by selecting the Filter(s) option to rank to the elements. For more details, refer filter.



You can clear filters by selecting the Show All Records.

Here is an illustration,



How to format tree map widget?

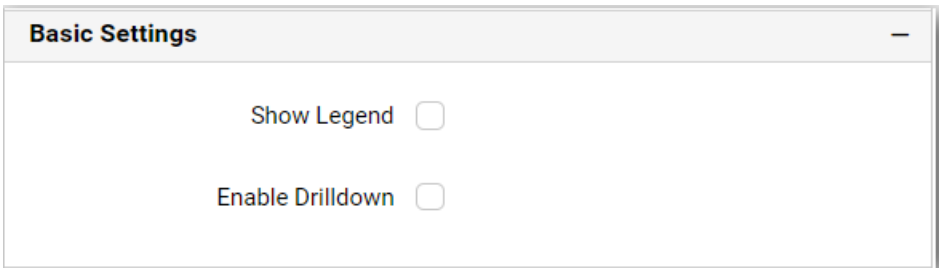
You can format the tree map for better illustration of the view that you require, through the settings available in Properties tab.

General Settings

Name

This allows you to set title for this tree map widget.

Basic Settings



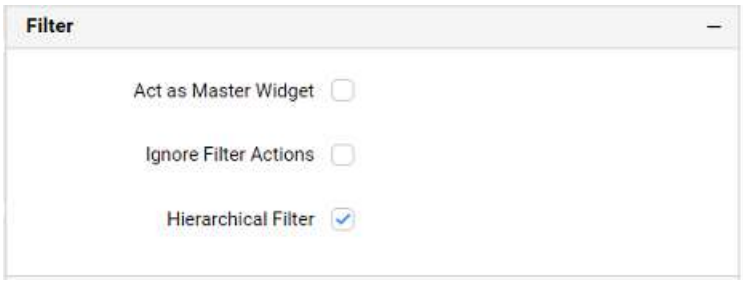
Show Legend

This allows you to toggle the visibility of legend.

Enable Drilldown

In case of hierarchical view, multiple levels will get rendered in the same view. This option will be visible, if you bind more than one column in the Columns section. This can be switched to drill down view through enabling this setting.

Filter



Act as Master Widget

This allows you to define this tree map widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Action

This allows you to define this tree map widget to ignore responding to the filter actions applied on other widgets in dashboard.

Hierarchical Filter

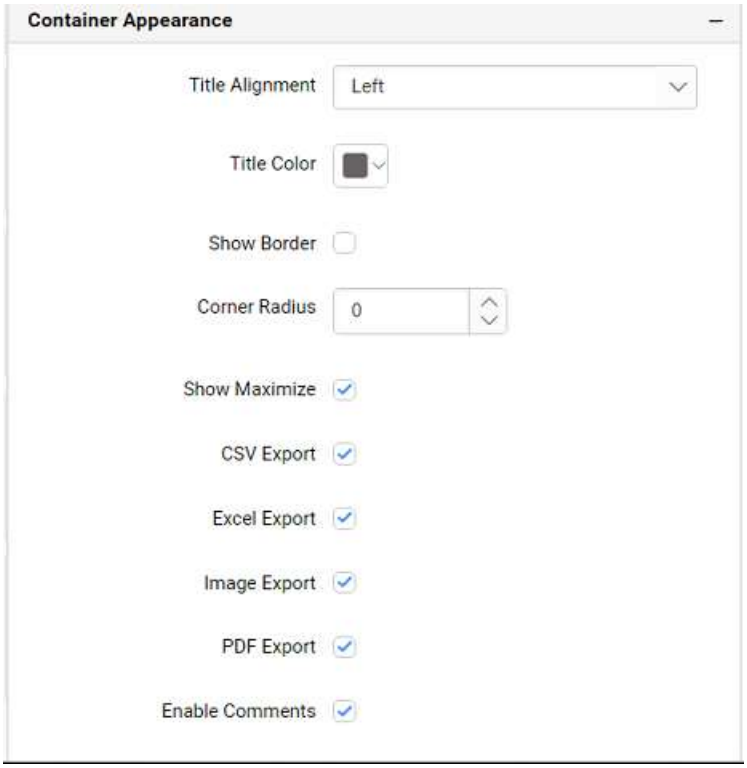
Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

When Hierarchical Filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Link

You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners, if **Show Border** property is enabled. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this tree map widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the tree map widget.

CSV Export

This allows you to enable/disable the CSV export option for this tree map widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this tree map widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format.

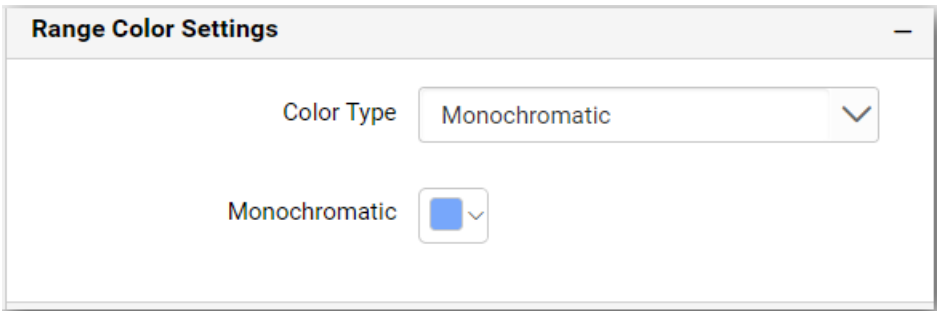
Image Export

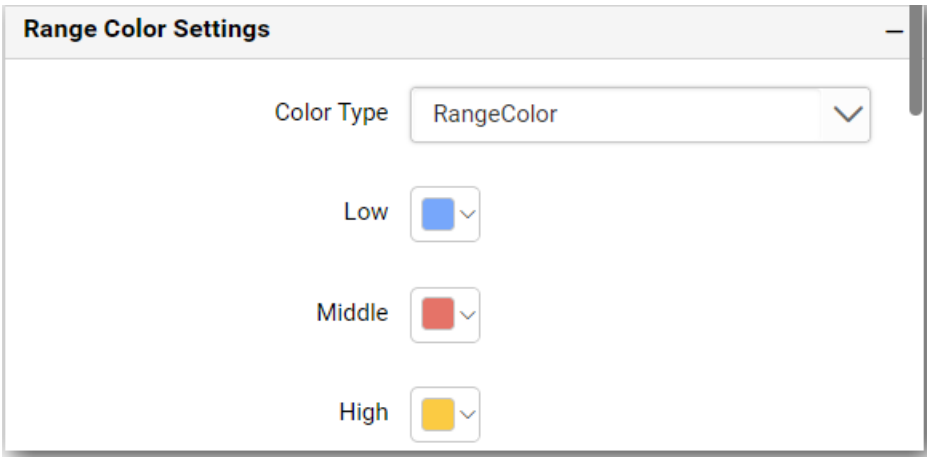
This allows you to enable/disable the image export option for this tree map widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

Range Color Settings





Color Type

This allows you to choose the color type to be applied for the tree map.

Monochromatic

This allows you to configure a single color palette whose saturation will be varied based on the value density.

Range Color

This allows you to configure three different colors that allocates itself to the value range accordingly.

Low

Let you choose single color for low values when Range Color is selected.

Middle

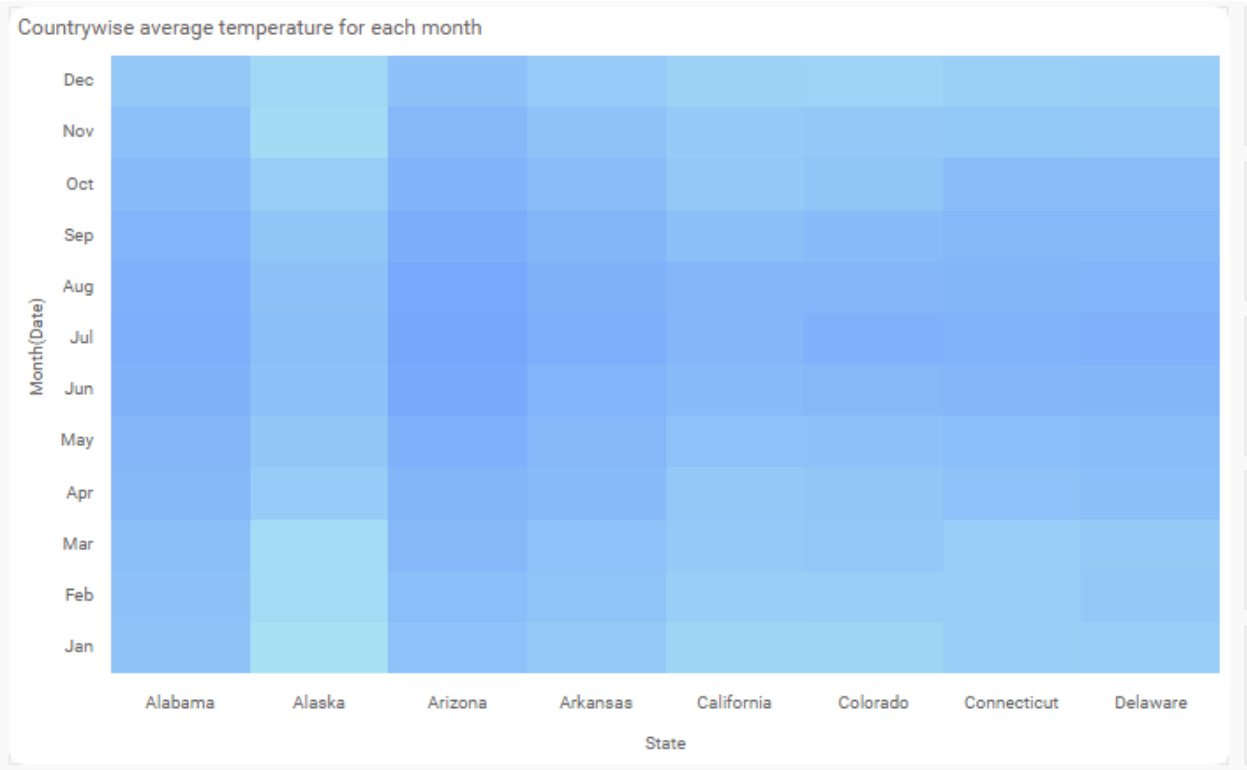
Let you choose single color for middle values when Range Color is selected.

High

Let you choose single color for high values when Range Color is selected.

HeatMap

HeatMap allows you to visualize large amounts of data as clustered rectangles with a color scale.

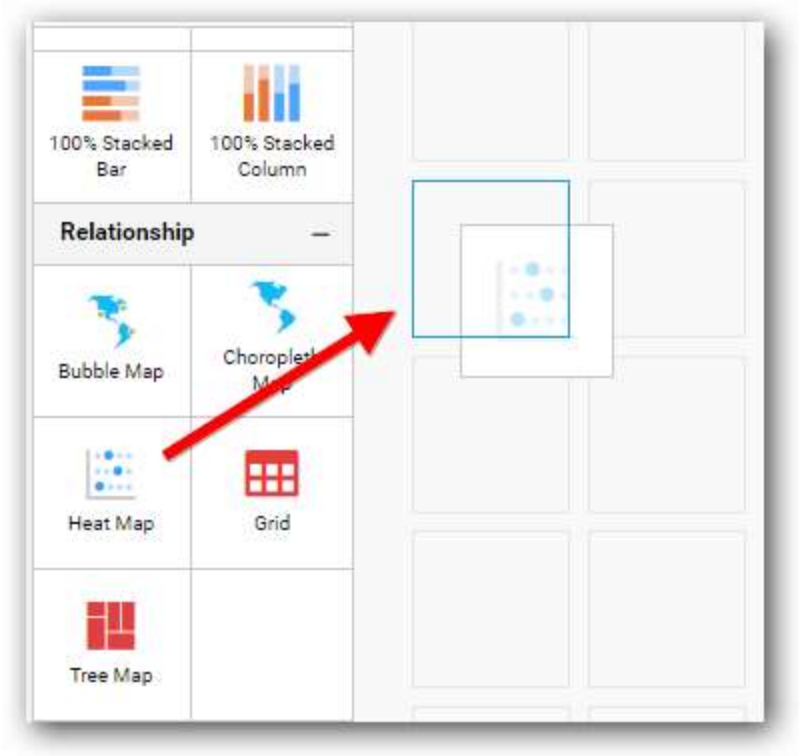


How to configure the data table to HeatMap widget

To showcase a HeatMap, a minimum requirement of 1 value and two groups by the field is needed.

The following procedure illustrates the data configuration of the HeatMap.

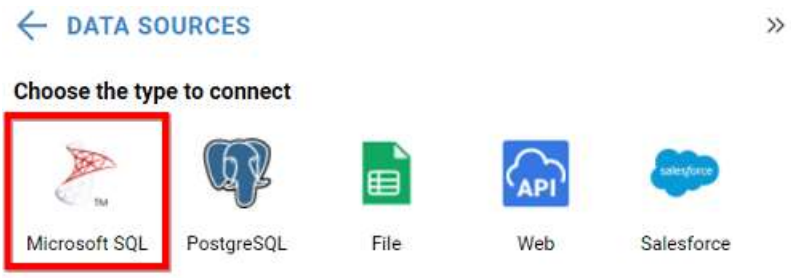
Drag and drop the **HeatMap** control icon from the toolbox into the design panel. You can find the control in the toolbox by search.



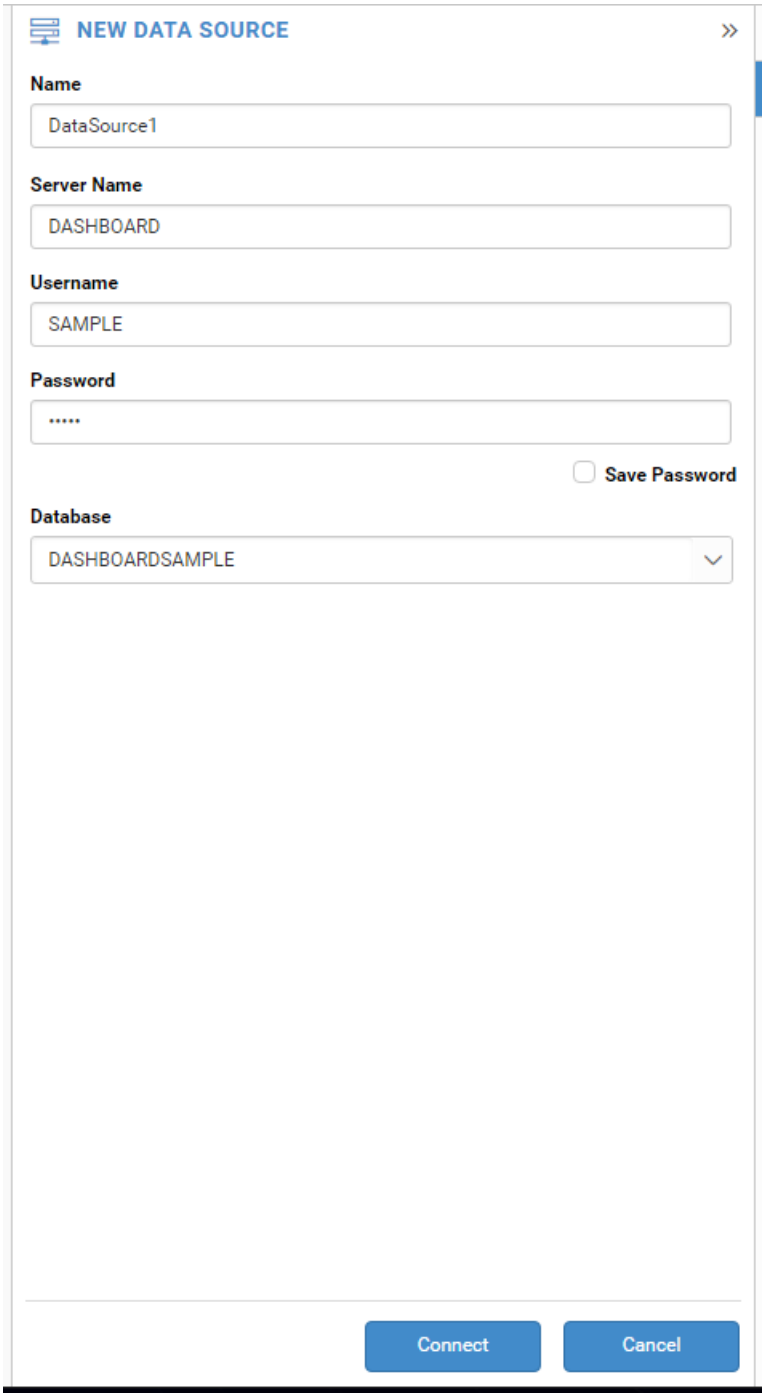
Click the **Data Source** button in the configuration panel.

Click **CREATE NEW** to launch a new connection from the connection type panel.

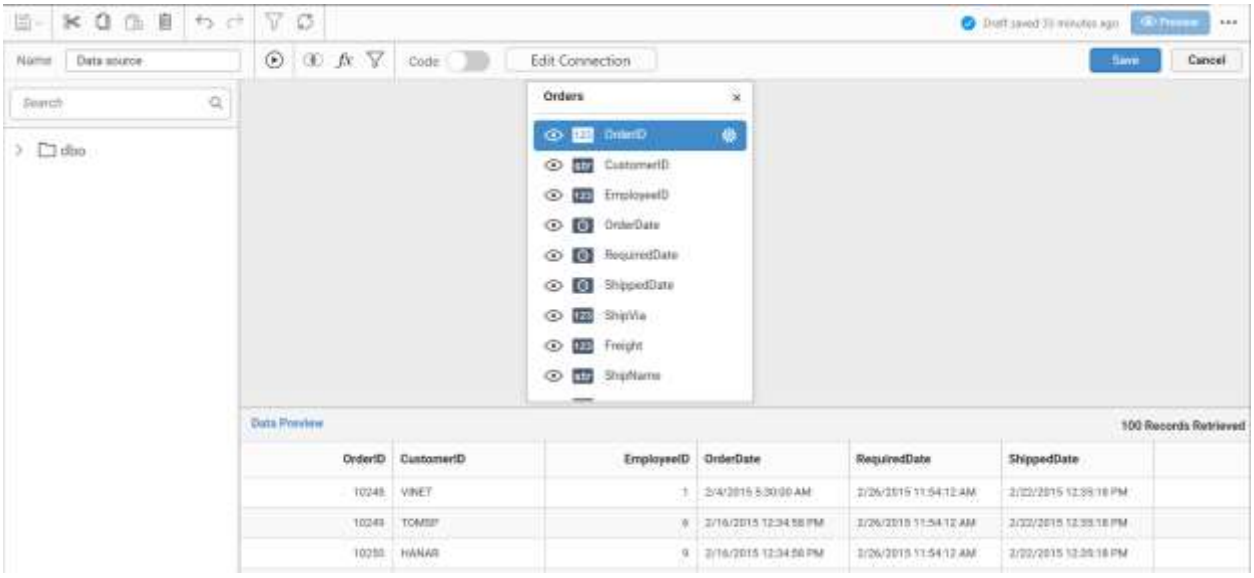
In the connection type panel, click any one (Here, **Microsoft SQL** connection type is selected for demonstration) of the listed connection type buttons shown below.



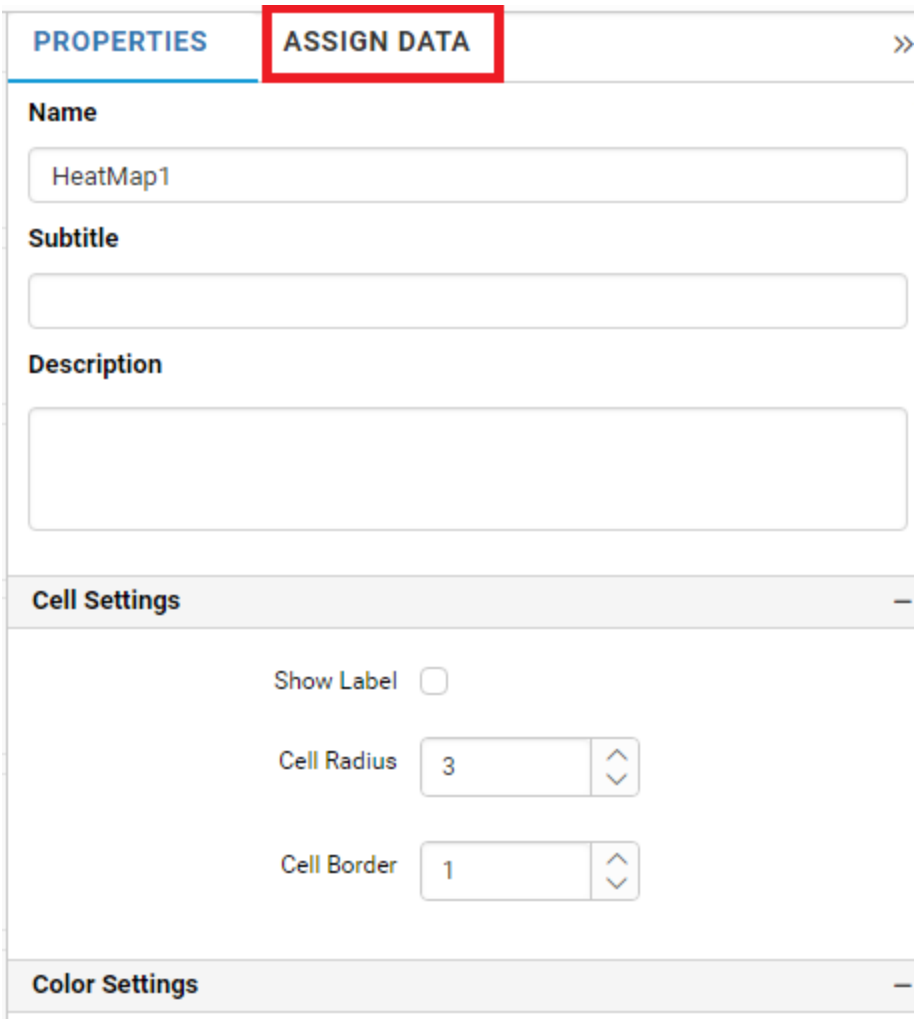
In the **NEW DATA SOURCE** configuration panel, fill the connection type and related details, and click **Connect**.



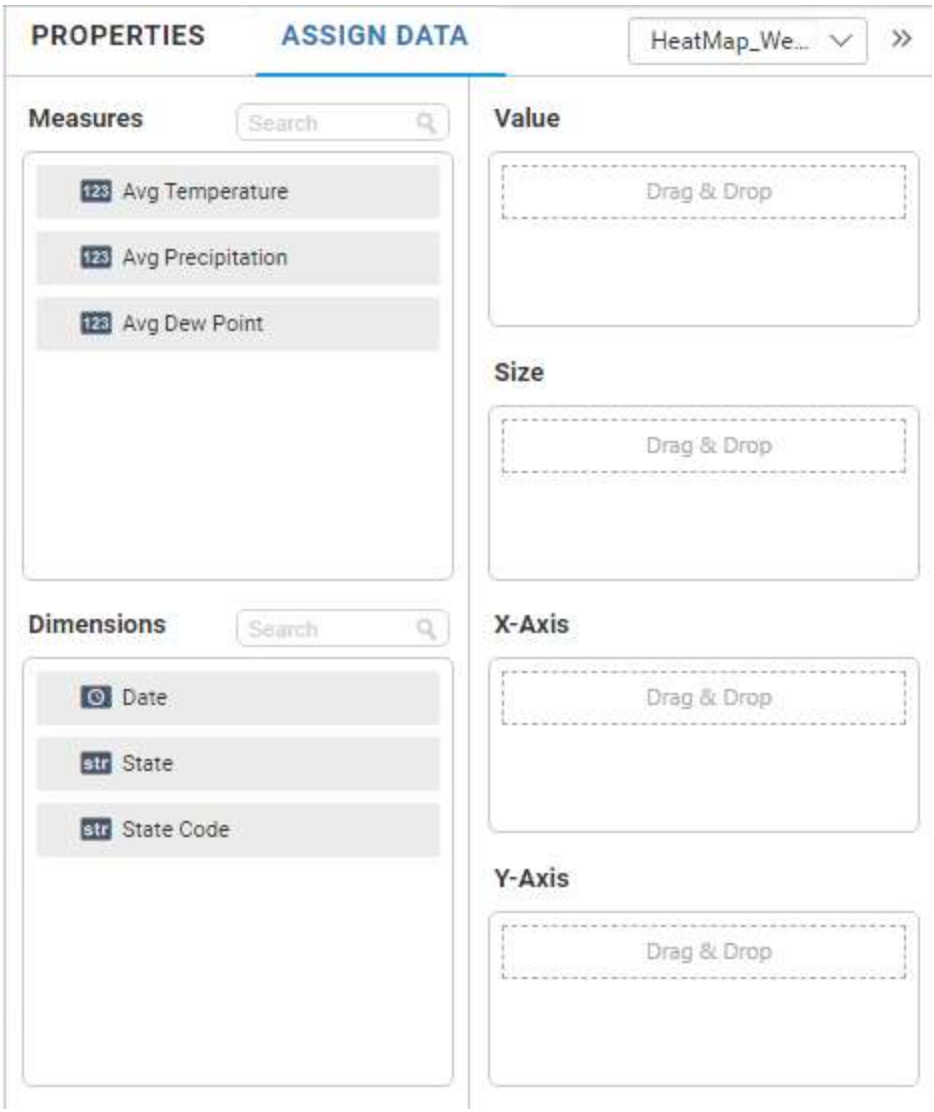
Drag your preferred table or view from the left pane in the data design view and click **Save**.



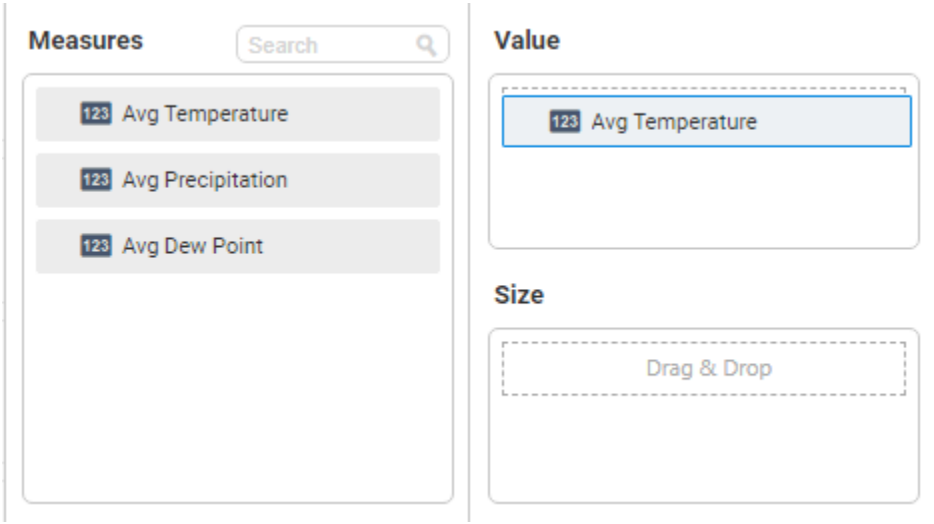
Click **Properties** in the configuration panel, the property pane opens. Now, switch to the **ASSIGN DATA** tab.



The data tab will be opened with available measures and dimensions from the connected data source.

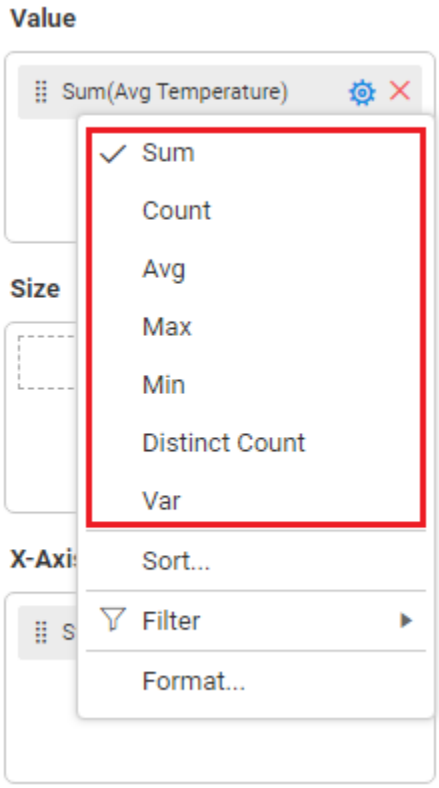


Bind a column through the drag and drop element from the measure section to value.

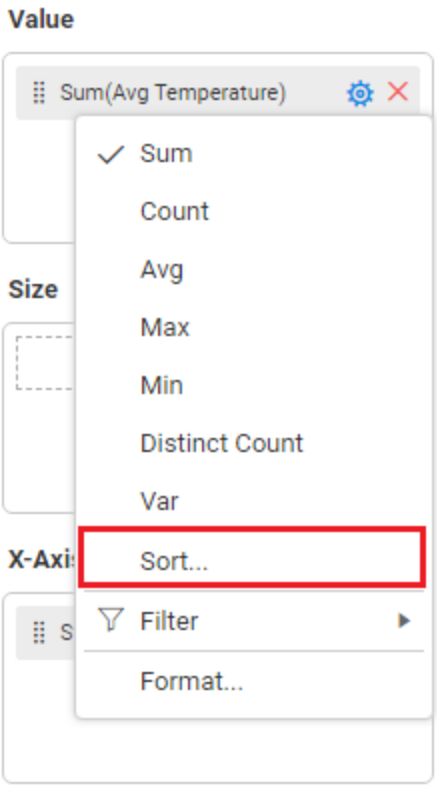


Drag the elements from dimension sections to x-axis and y-axis.

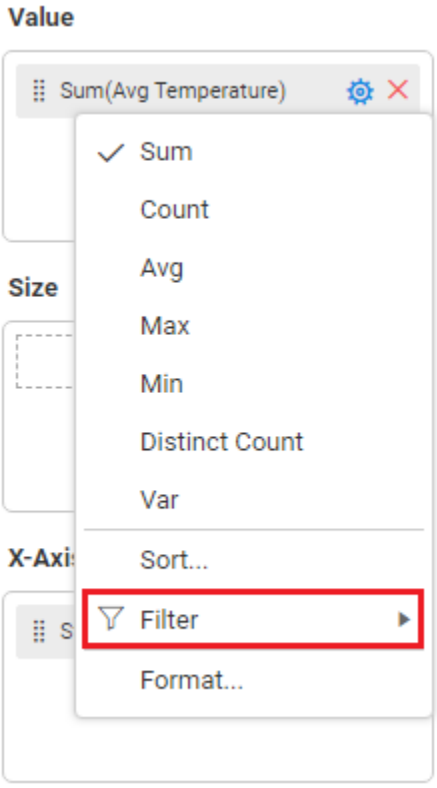
You can use the aggregation functions to change the values of the column.



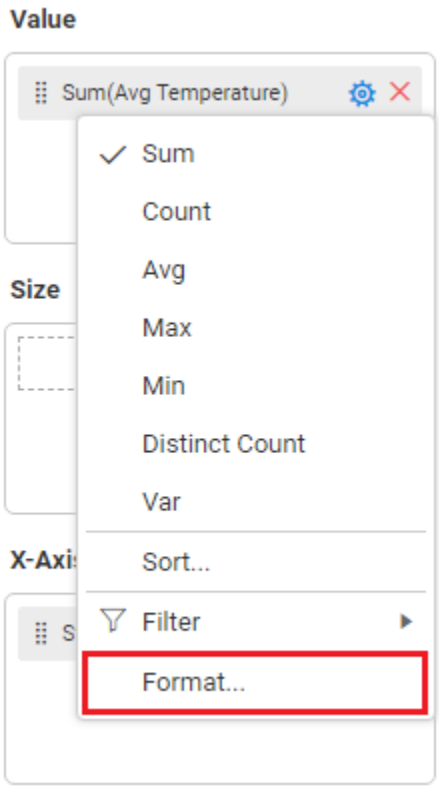
You can sort the data using the **Sort...** option shown under the settings menu list. For more details, refer to Sort.



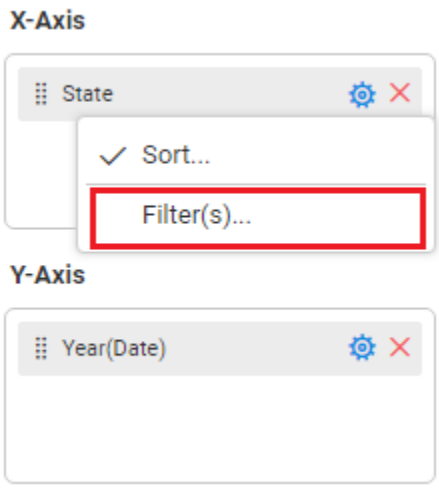
You can filter the data using the **Filter** option. For more details, refer to filter.



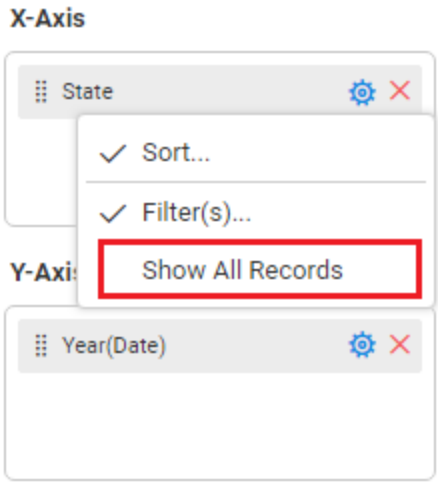
You can format the values by selecting the **Format...** option. For more details, refer to measure format.



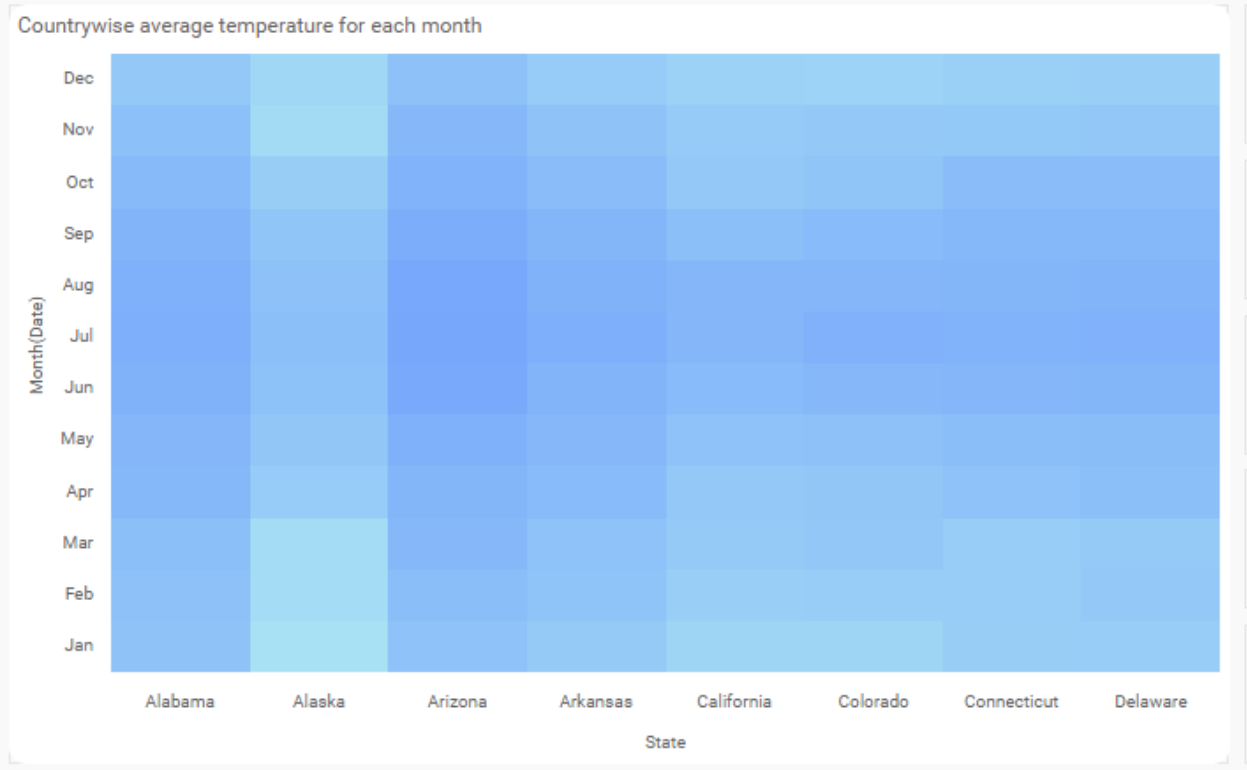
You can filter the data by selecting the **Filter(s)...** option. For more details, refer to filter.



You can clear the filters by selecting the **Show All Records** options.



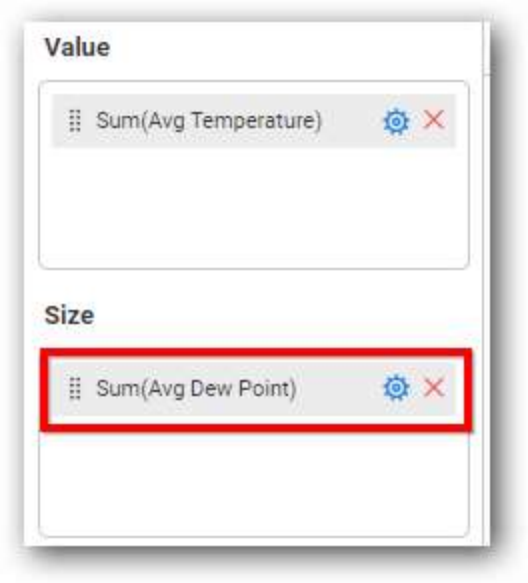
Here is an illustration,



Assigning Size value

HeatMap bubble size will be calculated based on its corresponding size column values.

Drag the elements from measure section to size.



Now the heatmap rendered like this

How to format HeatMap widget

You can format the HeatMap for better illustration of the view that you require, through the settings available in the **Properties** tab.

General settings

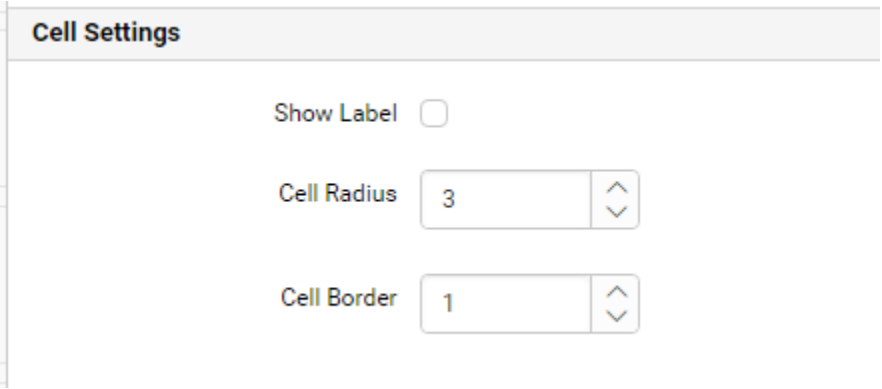
Name

HeatMap1

Name

This allows you to set a title for the HeatMap widget.

Cell settings

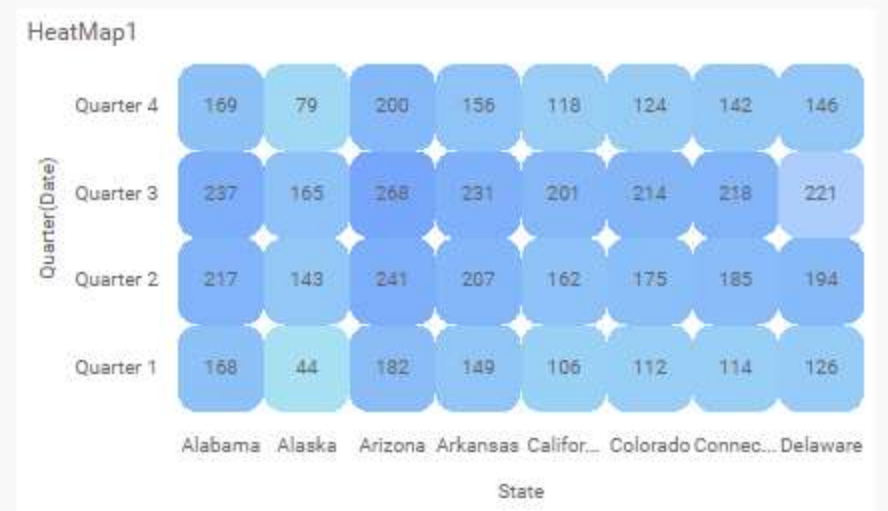


Show label

This allows you to toggle the visibility of value labels.

Cell radius

This allows you to apply the specified radius to cell corners. Value ranges from 0 to 10.



Cell border

This allows you to toggle the visibility of border surrounding the cell. Value ranges from 0 to 10.

Color settings

This allows you to customize a single color palette whose saturation will be varied based on the value density for minimum and maximum value.

Min color

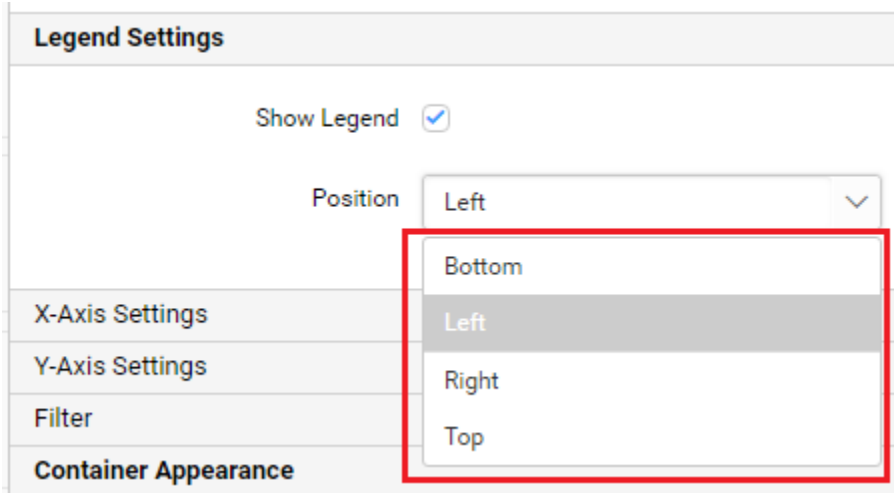
Let you choose single color for minimum values.

Max color

Let you choose single color for maximum values.

Here is an illustration,

Legend settings

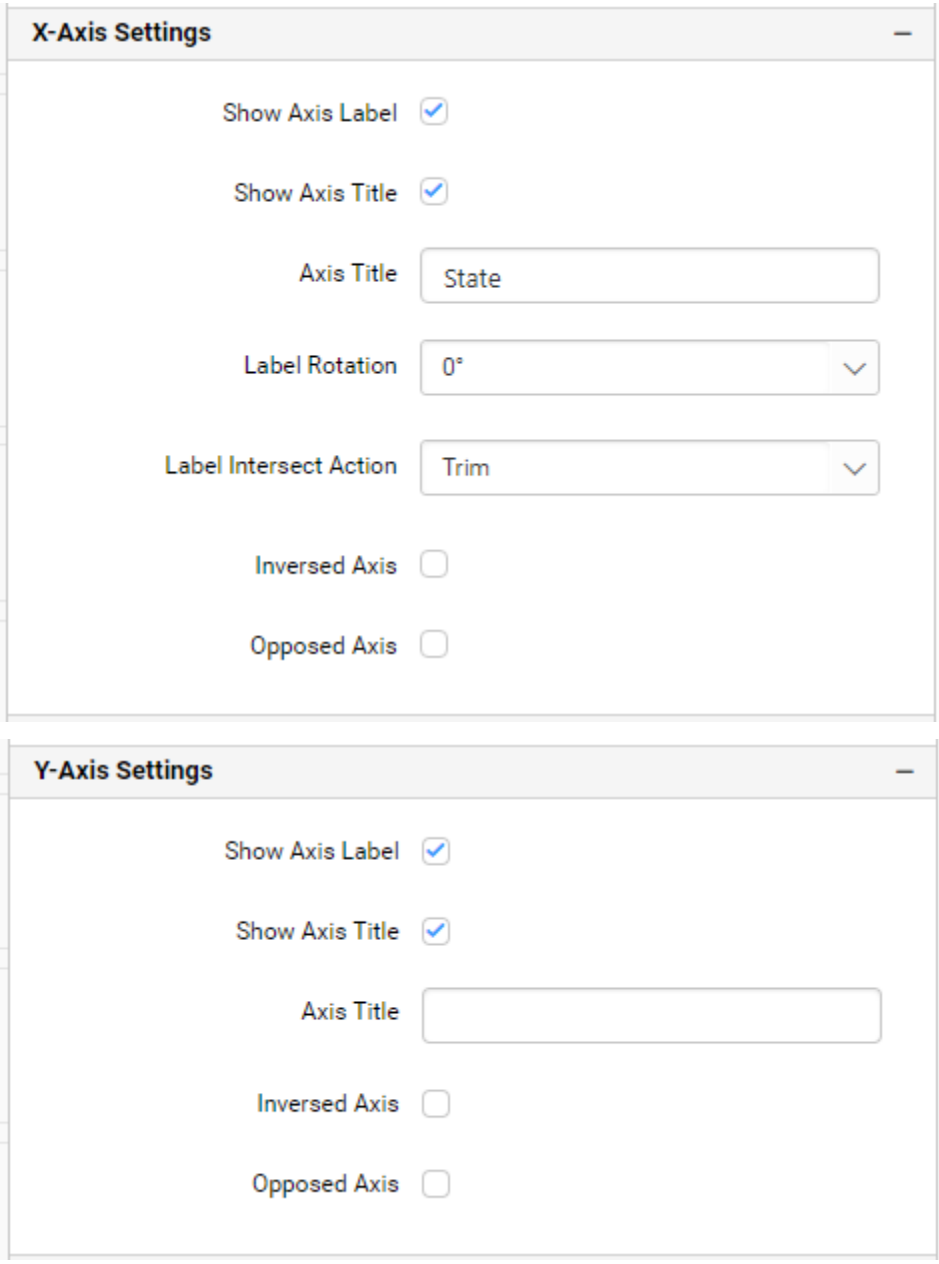


Show legend

This allows you to toggle the visibility of legend in the HeatMap and also can change the legend position (selecting through the combo box).

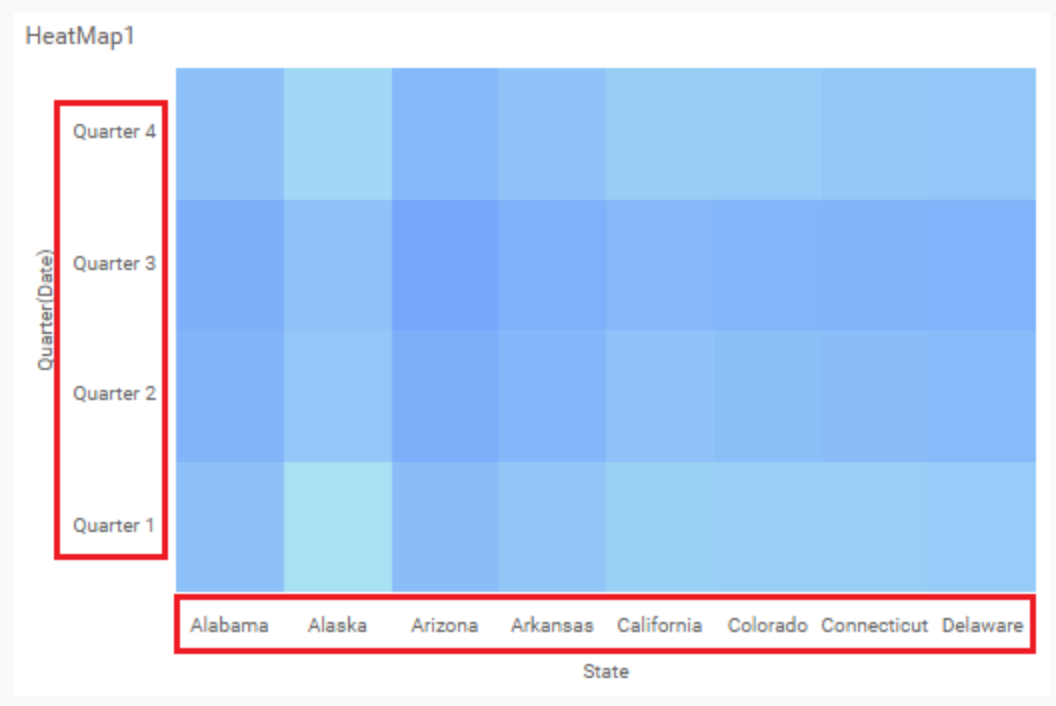
Axis

This section allows you to customize the axis settings in the HeatMap.



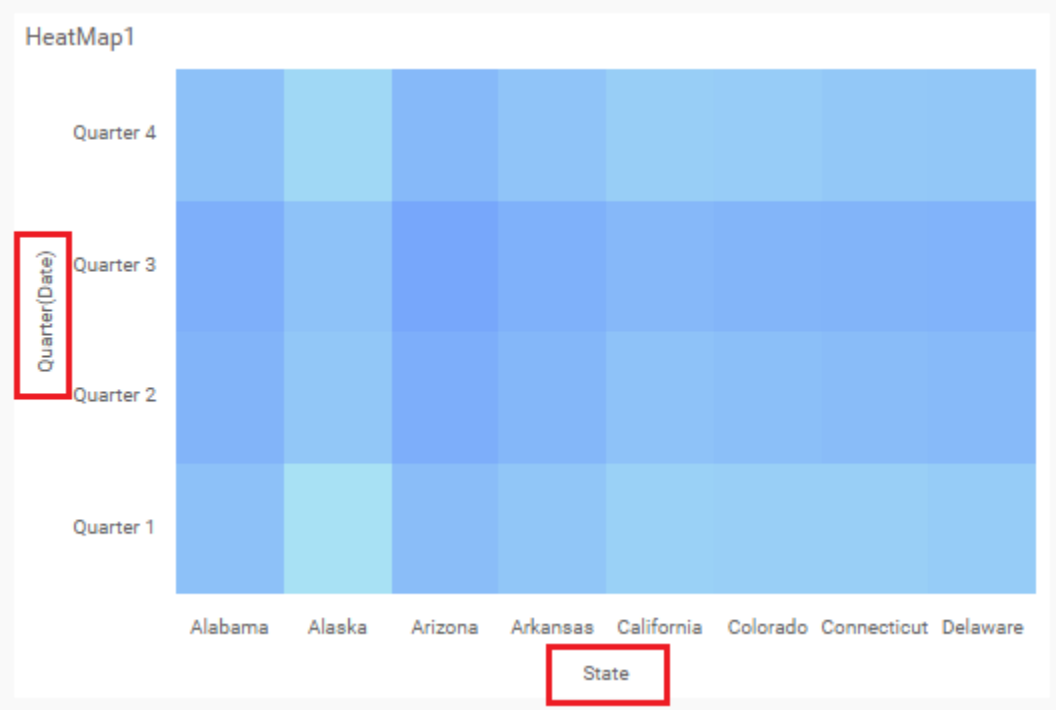
Show axis label

This allows you to enable the visibility of x-axis and y-axis labels.



Show axis title

This allows you to enable the visibility of x-axis and y-axis title.

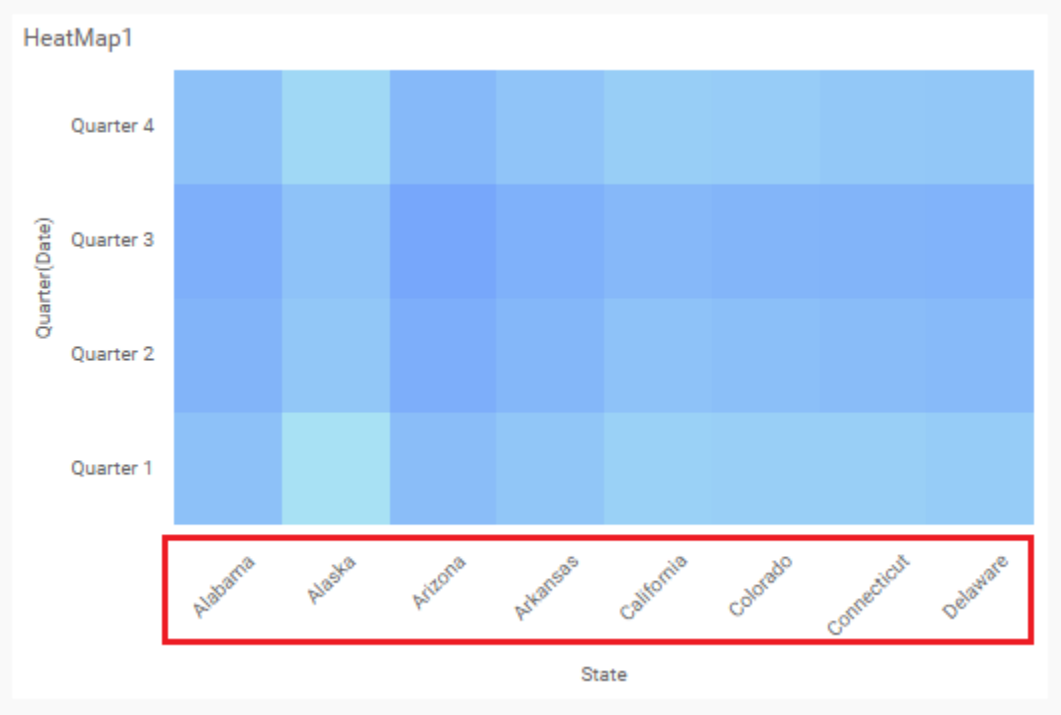


Axis title

This allows you to edit the x-axis and y-axis title for the HeatMap. It will reflect in the x-axis and y-axis title of the HeatMap.

Label rotation

This allows you to define the rotation angle for x-axis labels to display.

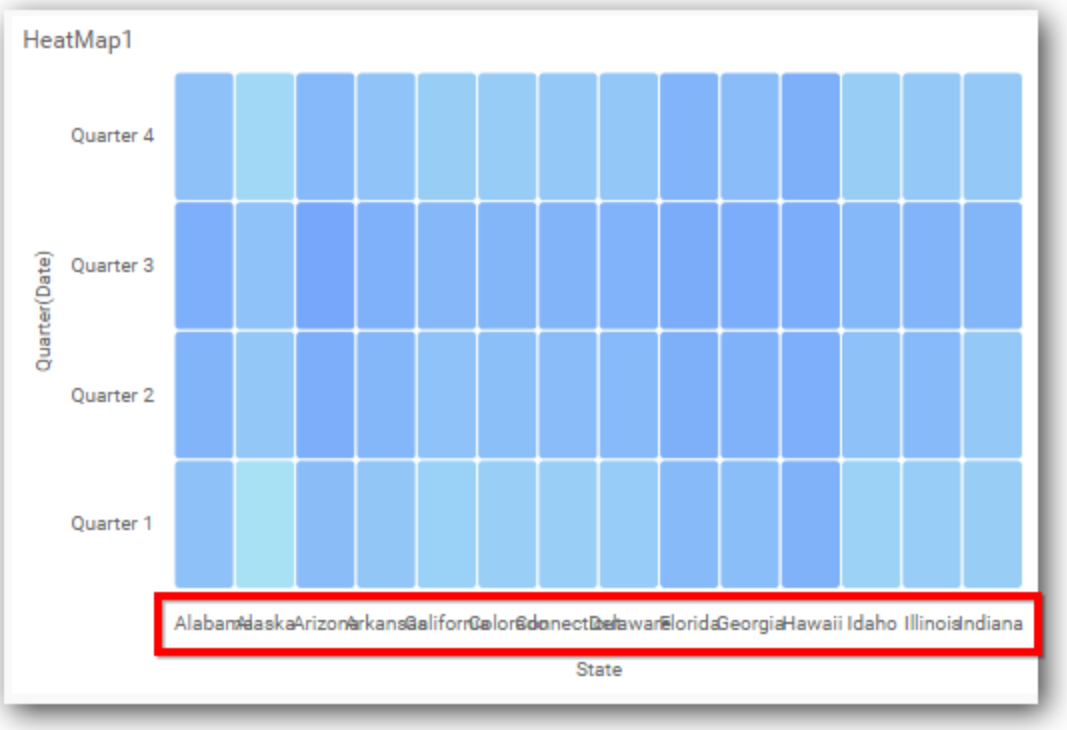


Label intersect action

This allows you to handle the display mode of overlapping labels in the x-axis.

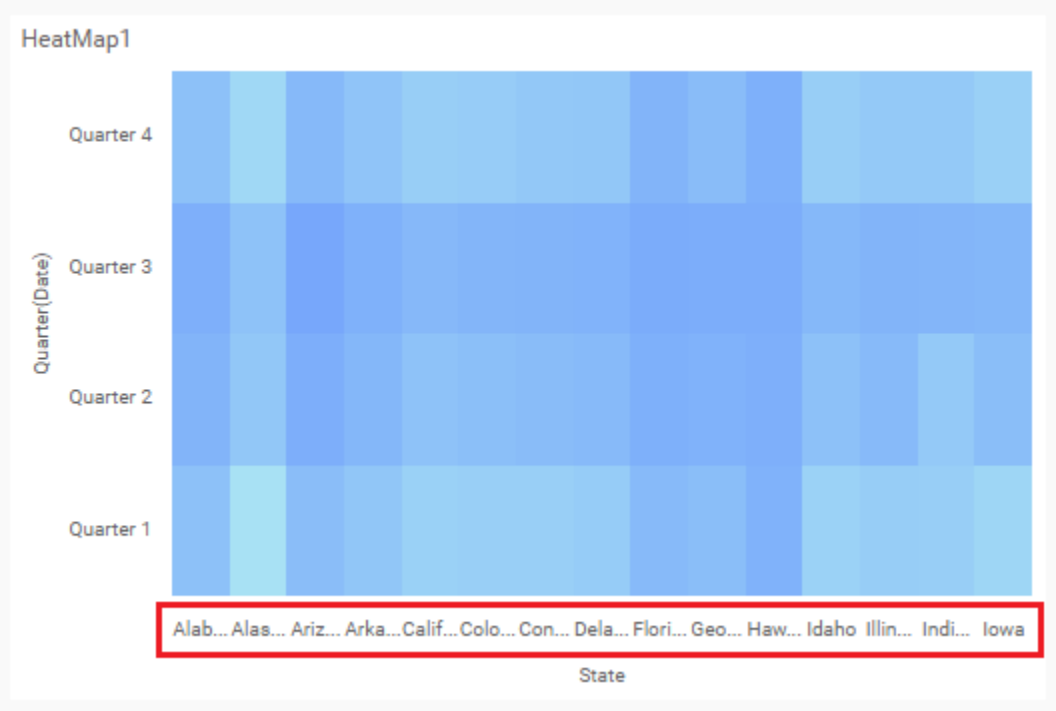
None

This option didn't trims the end of overlapping label in the axis.



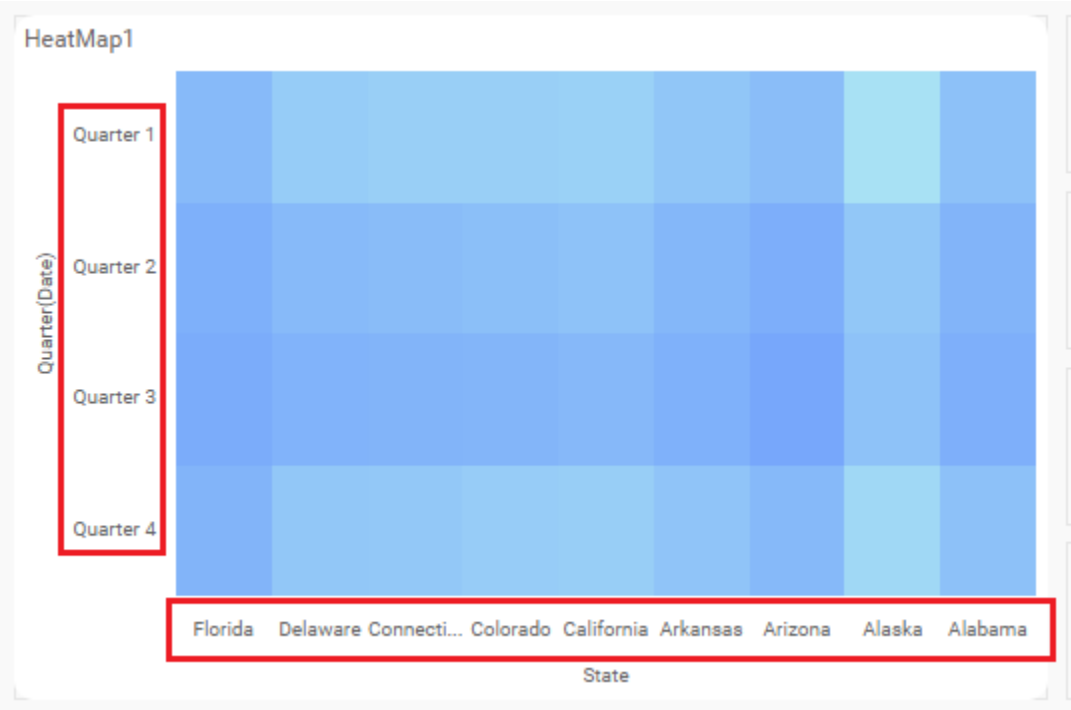
Trim

This option trims the end of overlapping label in the axis.



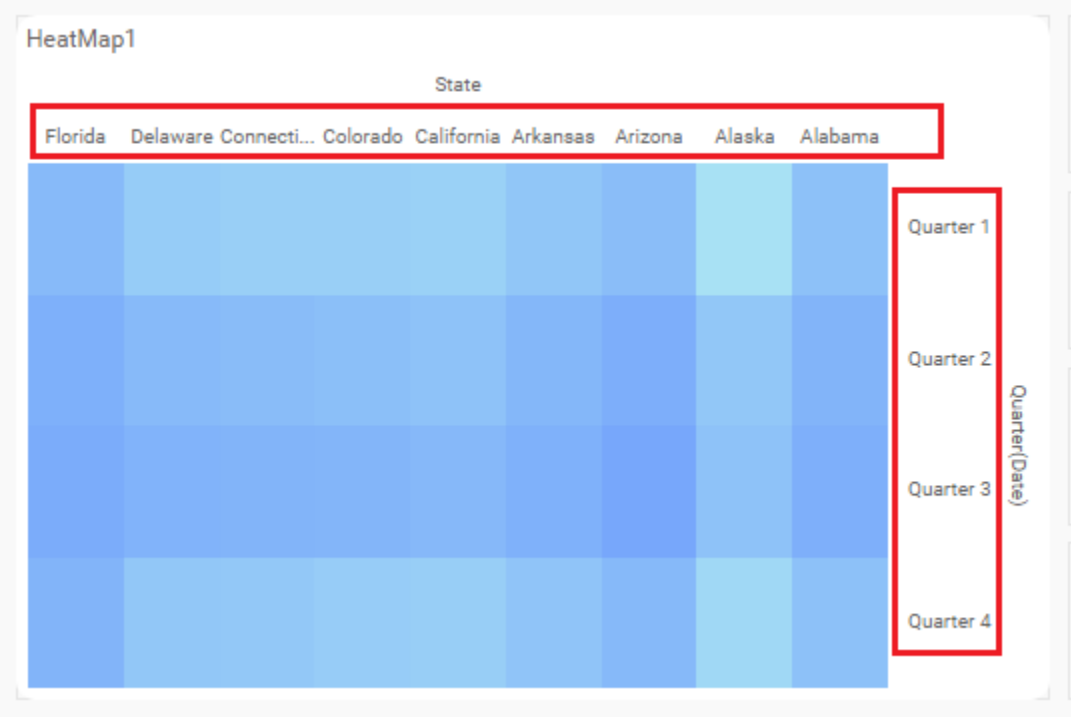
Inversed Axis

This allows you to change the axis label's placement order from left to right in the x-axis and axis label's placement order from bottom to top in the y-axis.

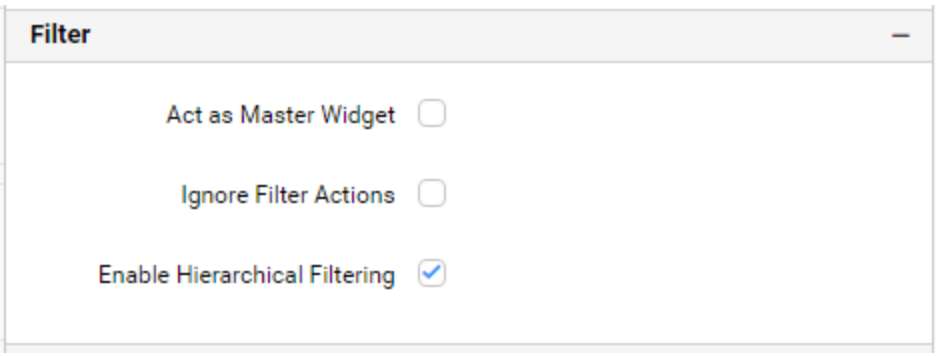


Opposed axis

This allows you to change the axis position from bottom to top in the x-axis and axis position from left to right in the y-axis.



Filter



Act as master widget

This allows you to define the HeatMap widget as a master widget, such that its filter action can be listened by other widgets in the Dashboard.

Ignore filter action

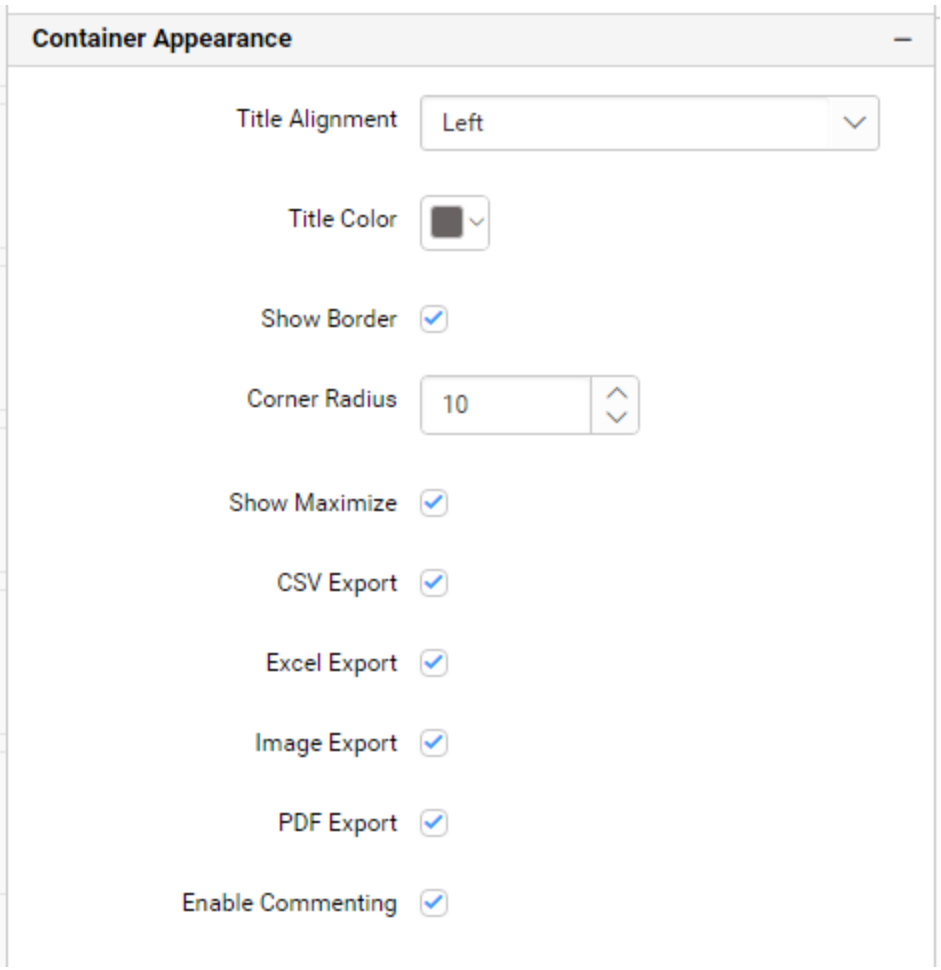
This allows you to define the HeatMap widget to ignore responding filter actions applied on other widgets in the Dashboard.

Hierarchical filter

Using this option, you can enable/disable the hierarchical Top N filtering. While applying Top N filter with multiple dimension columns, the returned data can be customized based on whether the filtering need to be done as flat or hierarchy of added dimension columns.

When the hierarchical filter option is enabled, the Top N will be applied for each individual column separately based on the number set for each column.

Container appearance



Title alignment

This allows you to handle the alignment of widget title to either left, center, or right.

Title color

This allows you to apply text color to the widget title.

Show border

This allows you to toggle the visibility of border surrounding the widget.

Corner radius

This allows you to apply the specified radius to widget corners, if the Show Border property is enabled. Value ranges from 0 to 10.

Show maximize

This enables/disables the maximized mode of the HeatMap widget. The visibility of the maximize icon in widget header will be defined based on this setting. Click this icon in the viewer to obtain the maximized view of the Heatmap widget.

CSV export

This enables/disables the CSV export option for the HeatMap widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel export

This enables/disables the Excel export option for the HeatMap widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format.

Image export

This enables/disables the image export option for the HeatMap widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in the viewer.

PDF export

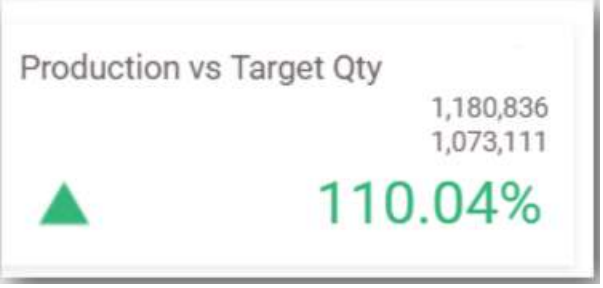
This enables/disables the PDF export option for the HeatMap widget. Enabling this allows you to export the view of the widget to pdf format in the viewer.

Enable comments

This enables/disables the comment for the dashboard widget. For more details, refer to here.

Card

Card allows you to measure trends through key performance indicators (KPIs) like value and goal.



How to configure table data to card widget?

To showcase a card, a minimum requirement of 1 actual and/or target values is needed.

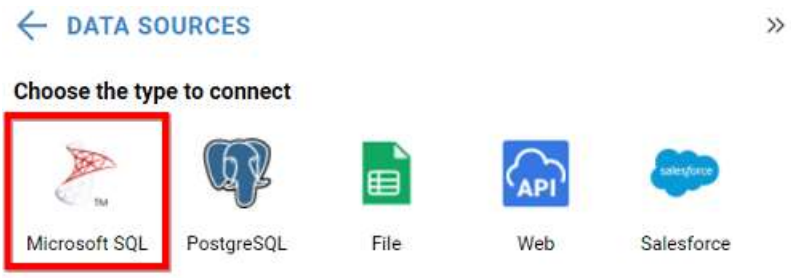
The following procedure illustrates data configuration of card.

Drag and drop Card control icon from the Tool box into design panel. You can find control in tool box by search.

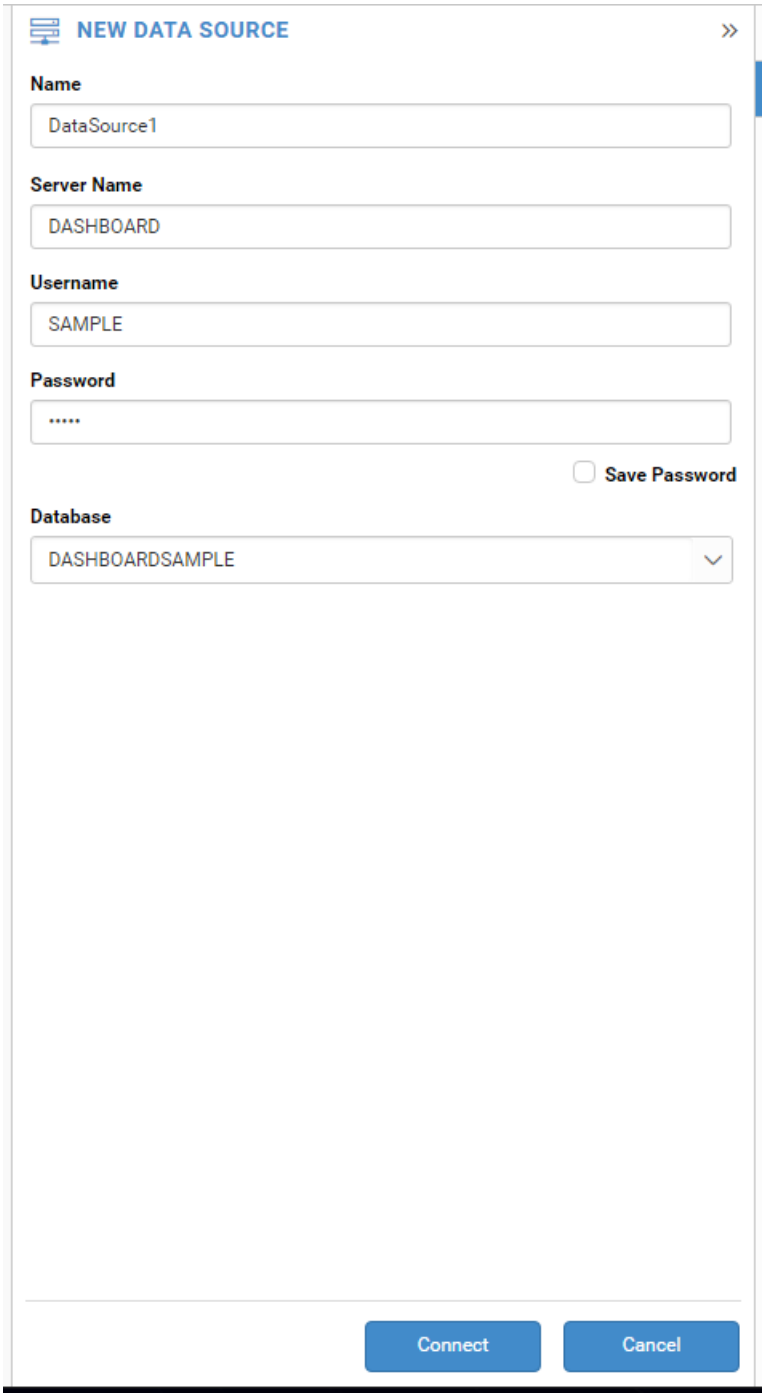
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

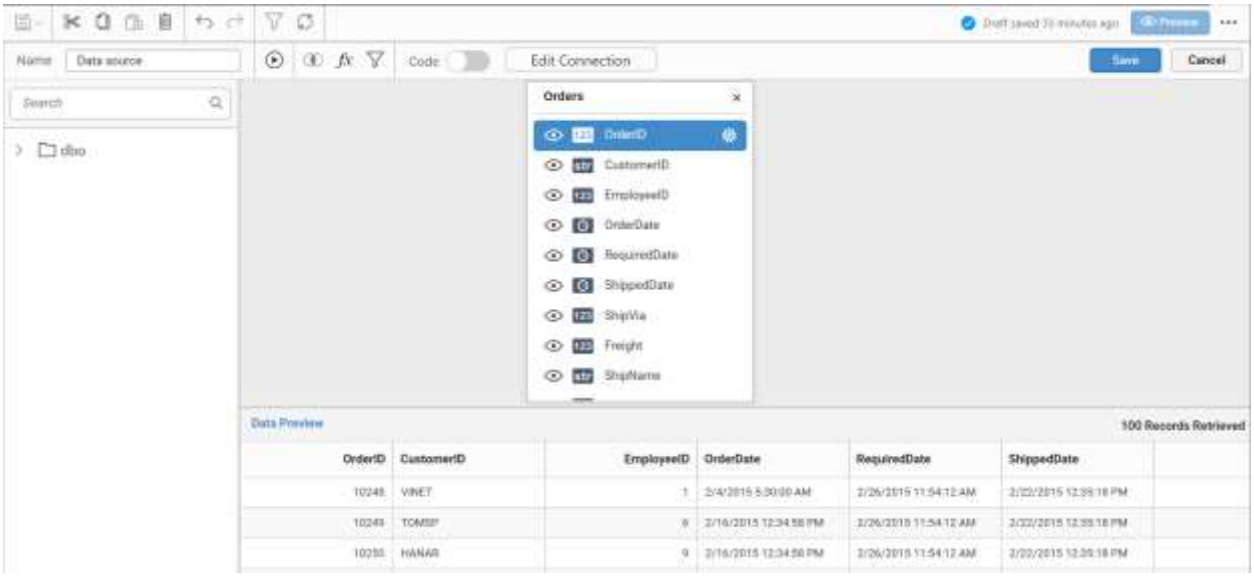
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



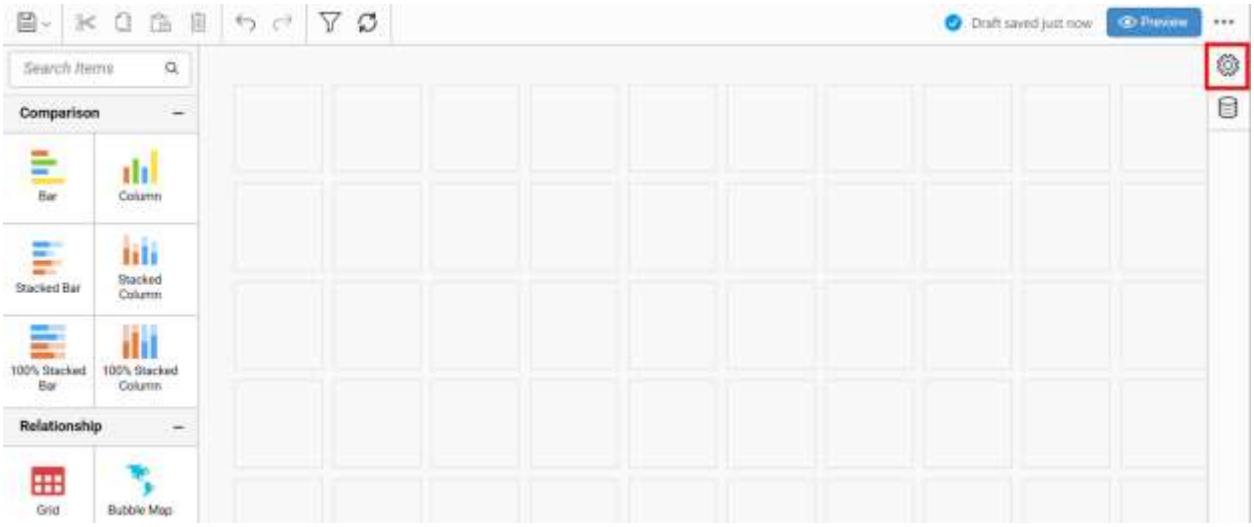
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



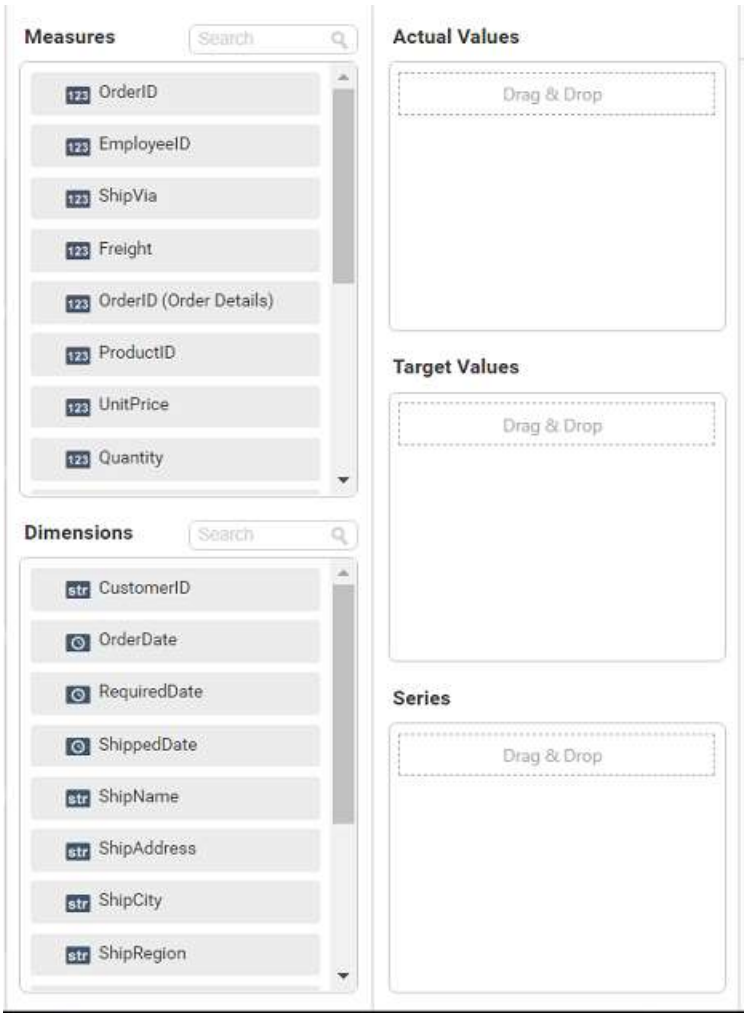
Drag your preferred table or view from the left pane from data design view, click Save button.



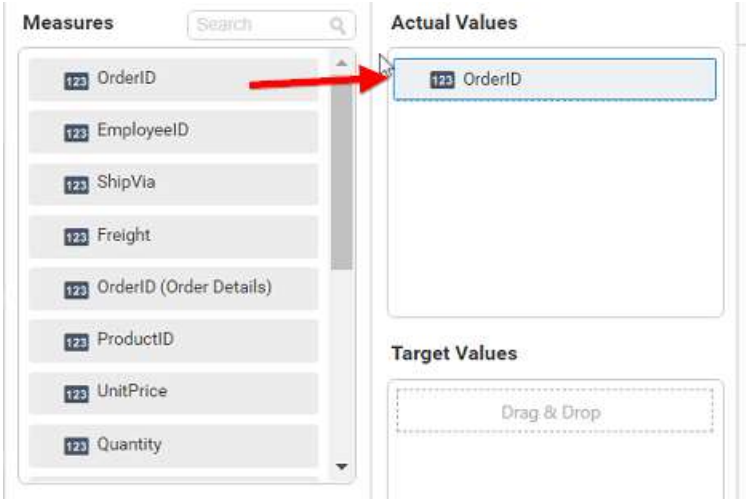
Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.



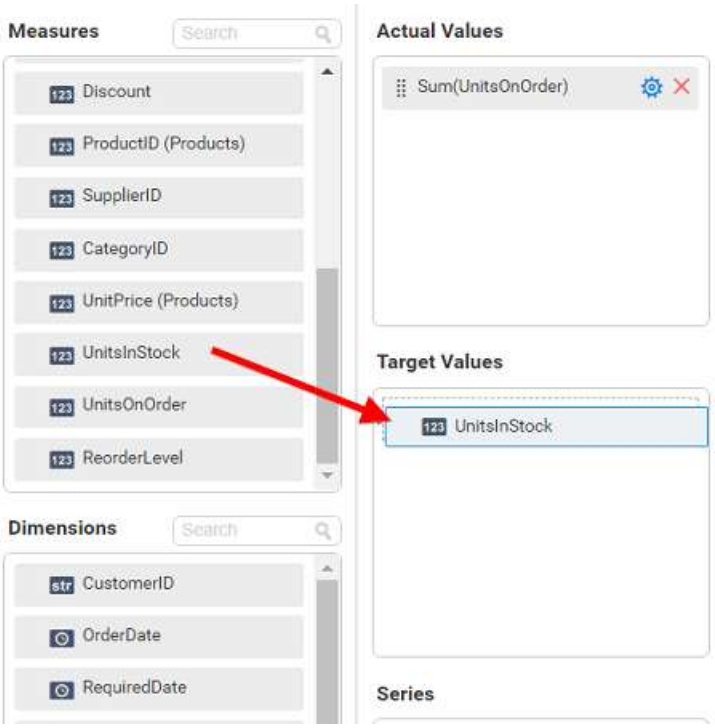
The data tab will be opened with available measures and dimensions from the connected data source



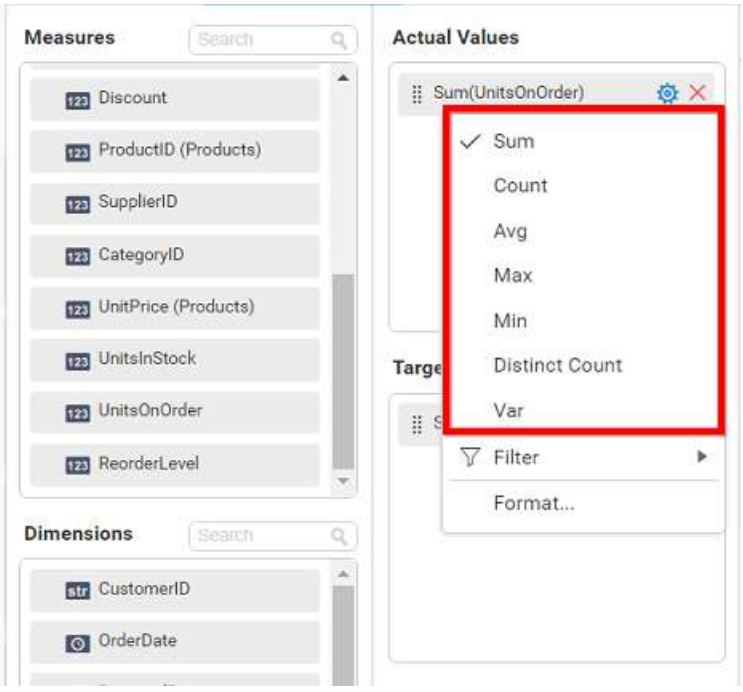
Bind column through drag and drop element from Measures section to Actual Values.



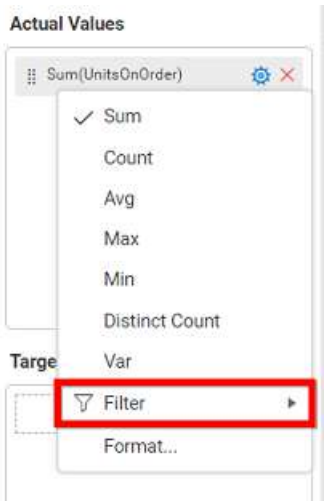
Drag and Drop the elements to Target Values.



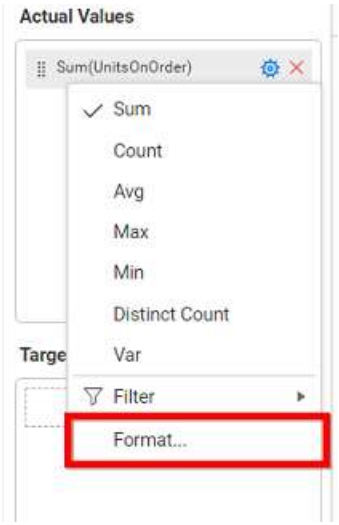
You can use aggregate function to change the Actual Values of the card.



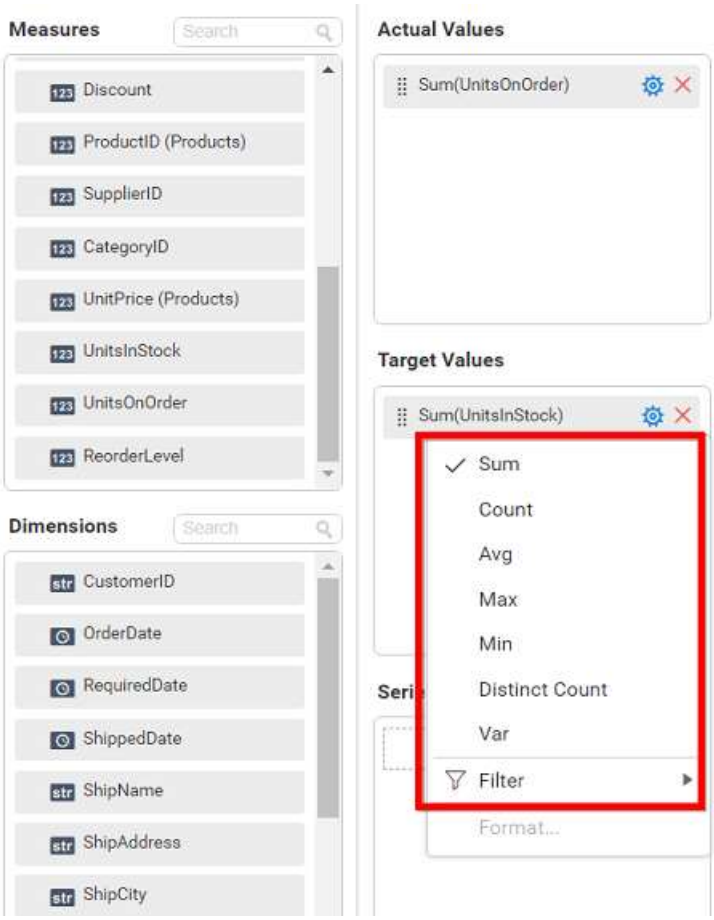
You can use Filter option to filter the data by specifying the filter condition. For more details, refer filter.



You can format the data to be displayed in the card by using format option. For more details, refer [measure format]

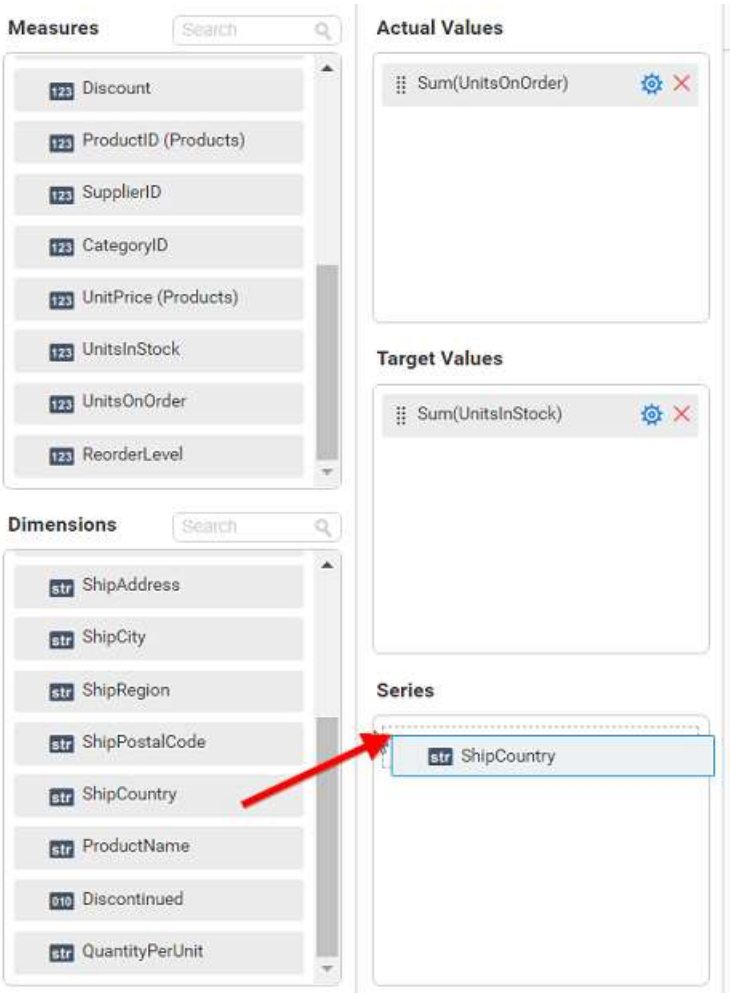


You can the change the Target values by selecting aggregate function.



You can use Filter option to filter the data by specifying the filter condition. For more details, refer filter.

Drag and Drop the elements from sections to Series.

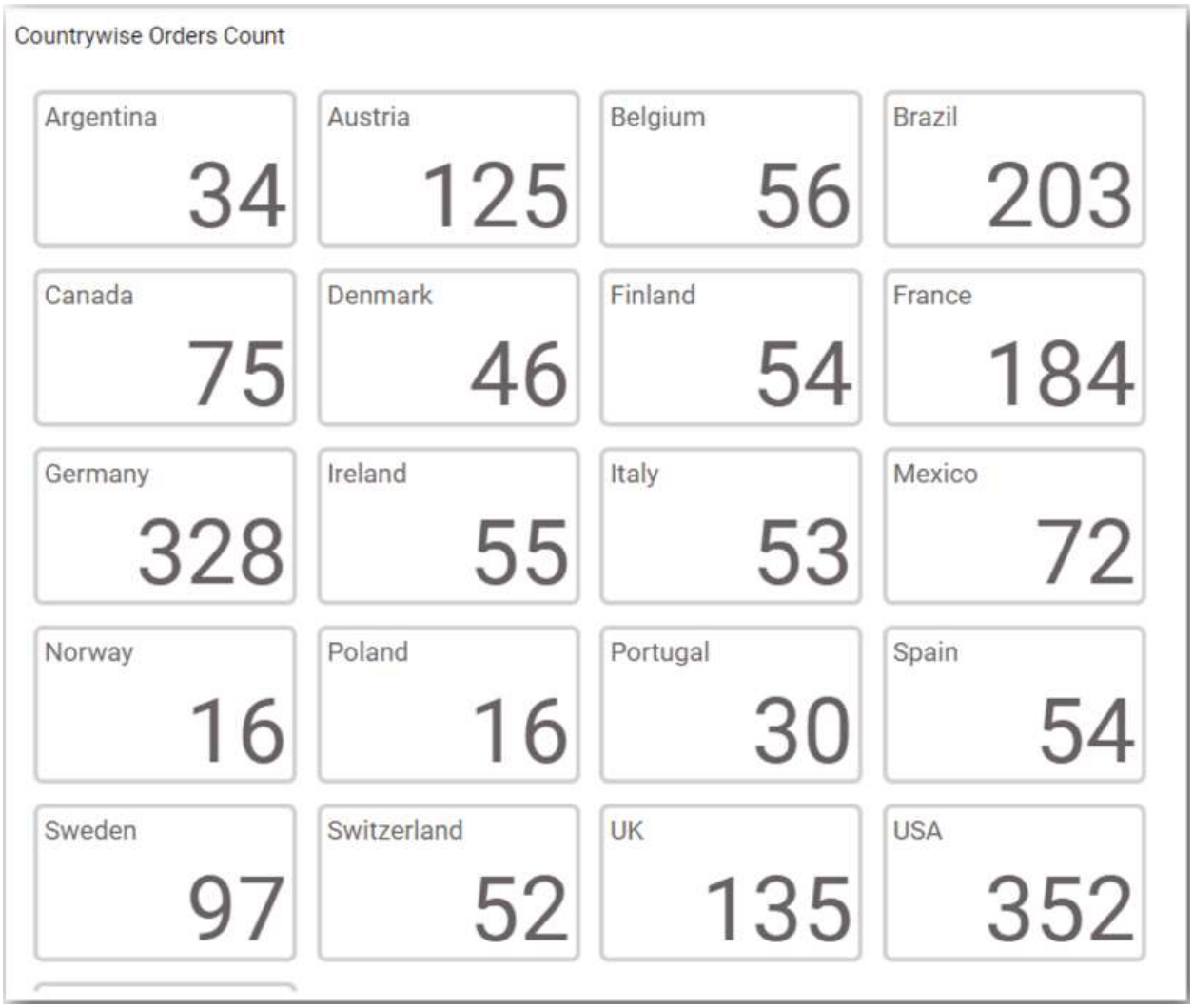


You can change the Series value of the card by changing the setting.

You can apply filter and sort option for the series field, if required.

You can clear filter by selecting Show All Records for Series section.

Here is an illustration for series card,



How to format card widget?

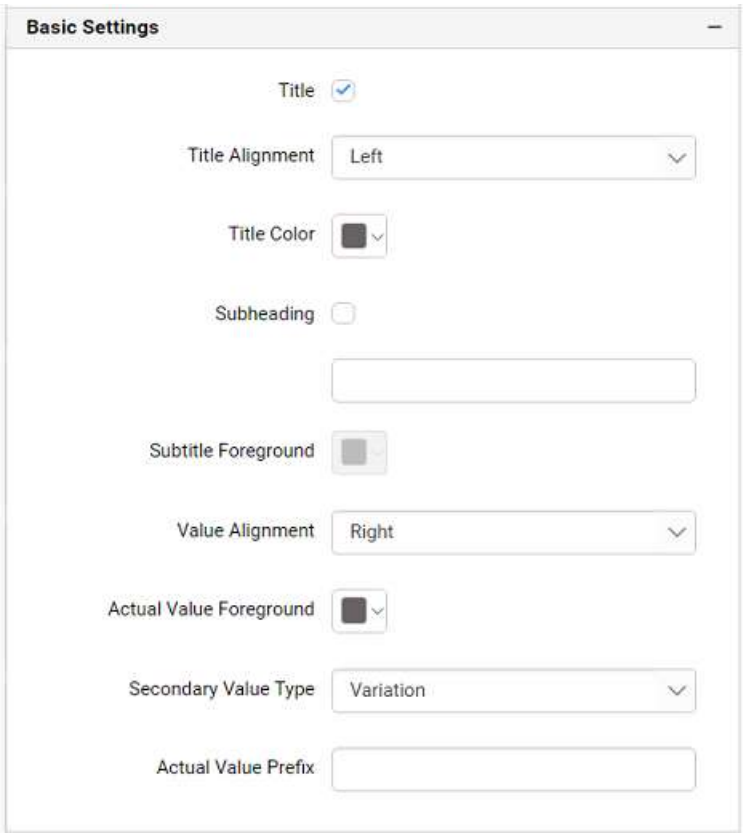
You can format the card for better illustration of the view that you require, through the settings available in Properties tab.

General Settings

Name

This allows you to set title for this card widget.

Basic Settings



Title

You can set a custom name as card title.

Title Alignment

The title can be aligned left, center, or right.

Title Color

The color of title text can be customized.

Subheading

A Subheading can be added to the card control providing a suitable text. Subheading text will be displayed at the bottom of the title text.

Subtitle Foreground

The text color of subtitle text can be customized.

Value Alignment

You can customize the card value alignment.

Actual Value Foreground

You can customize the color of card value.

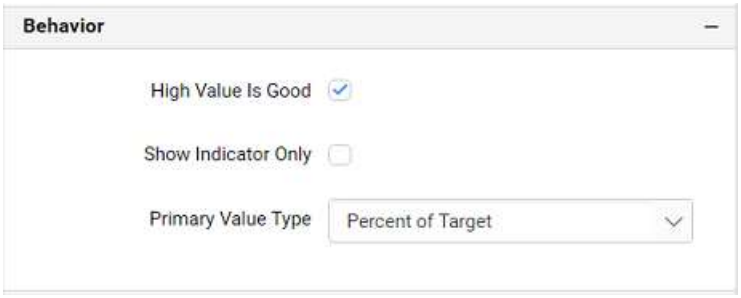
Secondary Value Type

This allows you to set the Secondary Value Type based on the Variation and Target Value.

Actual Value Prefix

This allows you to set the actual value prefix.

Behavior Settings



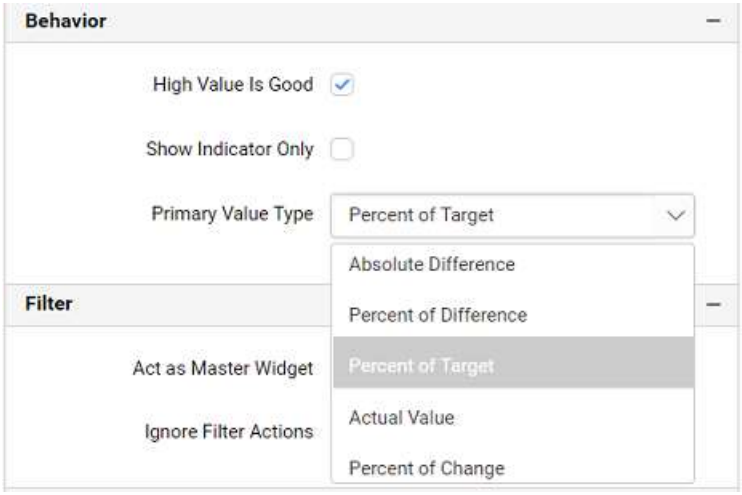
High Value is Good

The card visualization can be customized through specifying whether higher value should be treated as good or bad.

Show Indicator Only

Enabling this allows you to show indicator representation alone in card. This option will be available only if columns are added to both Actual Values and Target Values sections.

Primary Value Type



This allows you to customize the data showcased in card control by switching the available Primary Value Types.

- Absolute difference
- Percent of difference
- Percent of target
- Actual Value
- Percent of Change

Absolute difference

Absolute difference = Actual Value – Target Value

Percent of difference

Percent of difference = [((Actual Value - Target Value) / ([Actual Value / Target Value]/2)) 100]*

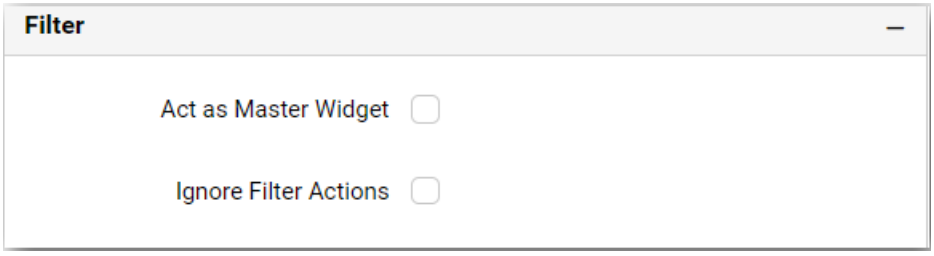
Percent of target

Percent of target= [([Actual Value / Target Value] 100)]*

Percent of Change

Percent of Change = [((Actual Value - Target Value) / Target Value) 100]*

Filter



Act as Master Widget

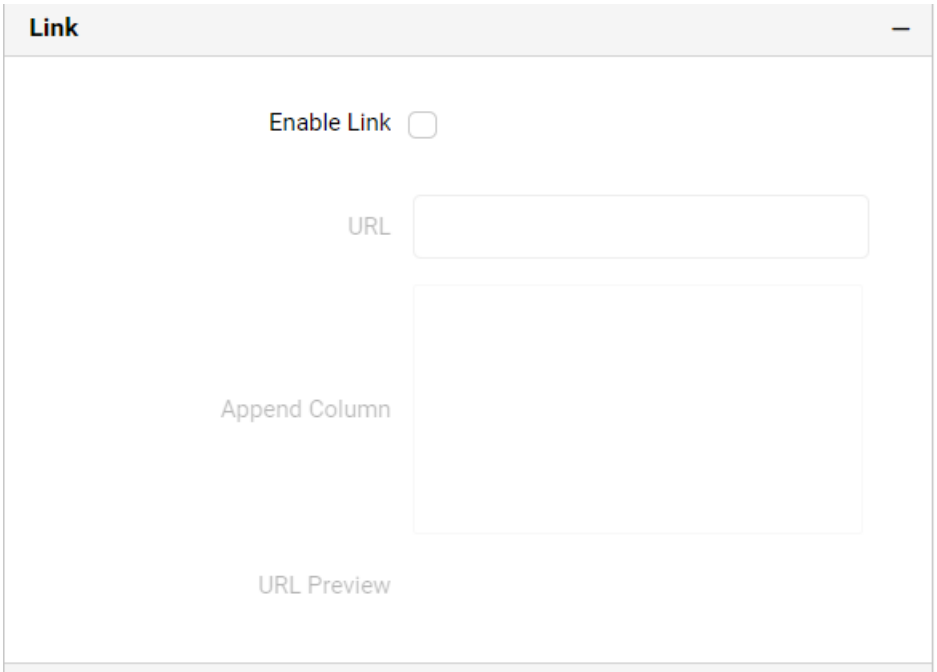
This allows you to define this widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Note: For single card, you can cannot enable Act as Master Widget option.

Ignore Filter Actions

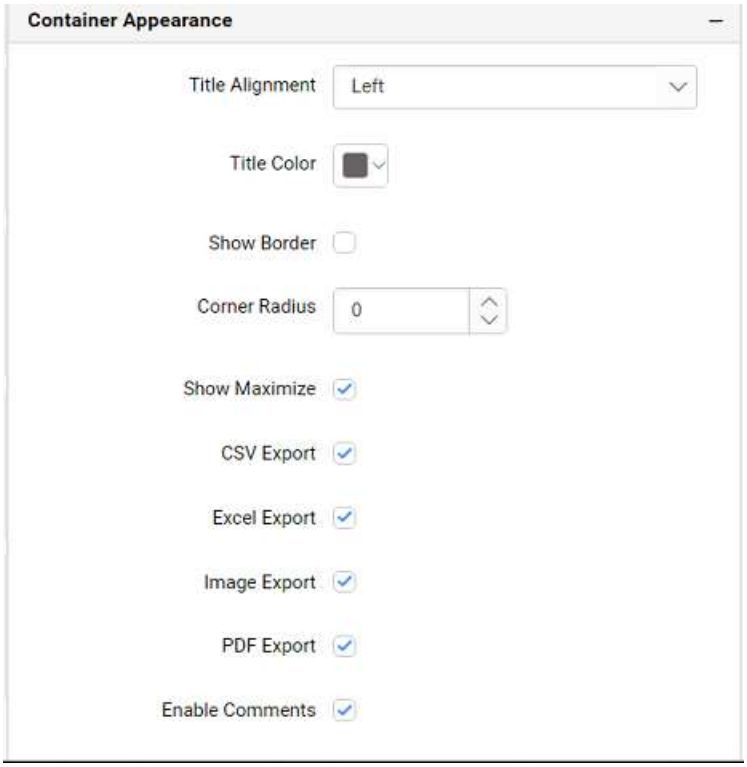
This allows you to define this widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link



You can enable linking and configure to navigate to a general URL with or without parameters. For more details, refer Linking

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this card widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the grid widget.

CSV Export

This allows you to enable/disable the CSV export option for this card widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this card widget. Enabling this allows you to export the summarized data of the widget view to (.xlsx or .xls) format.

Image Export

This allows you to enable/disable the image export option for this card widget. Enabling this allows you to export the view of the widget to image format (.jpg), (.png), or (.bmp) in viewer.

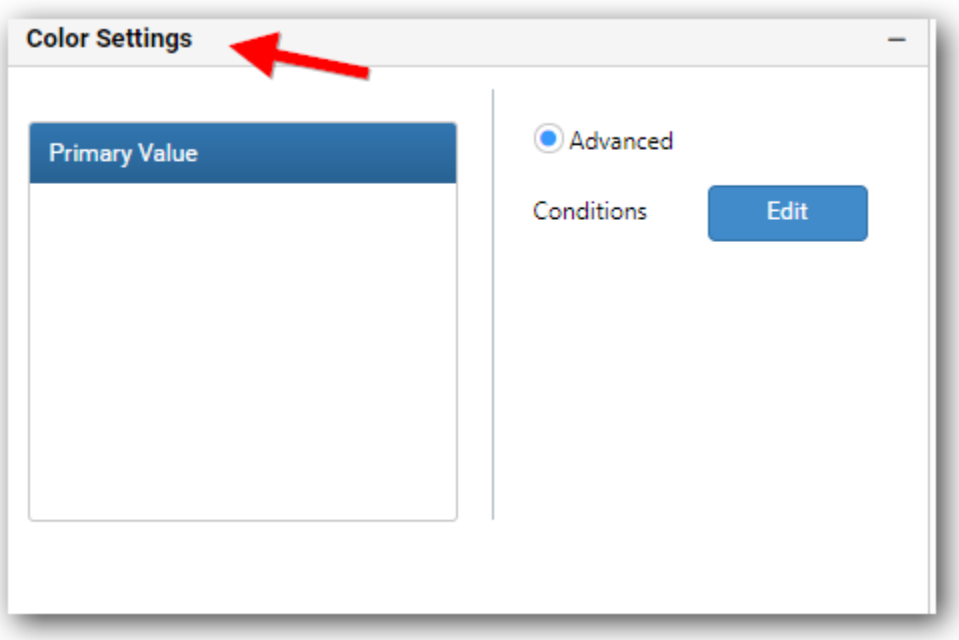
Note: Maximize and Export options are not available in the single card.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

How to apply conditional formatting?

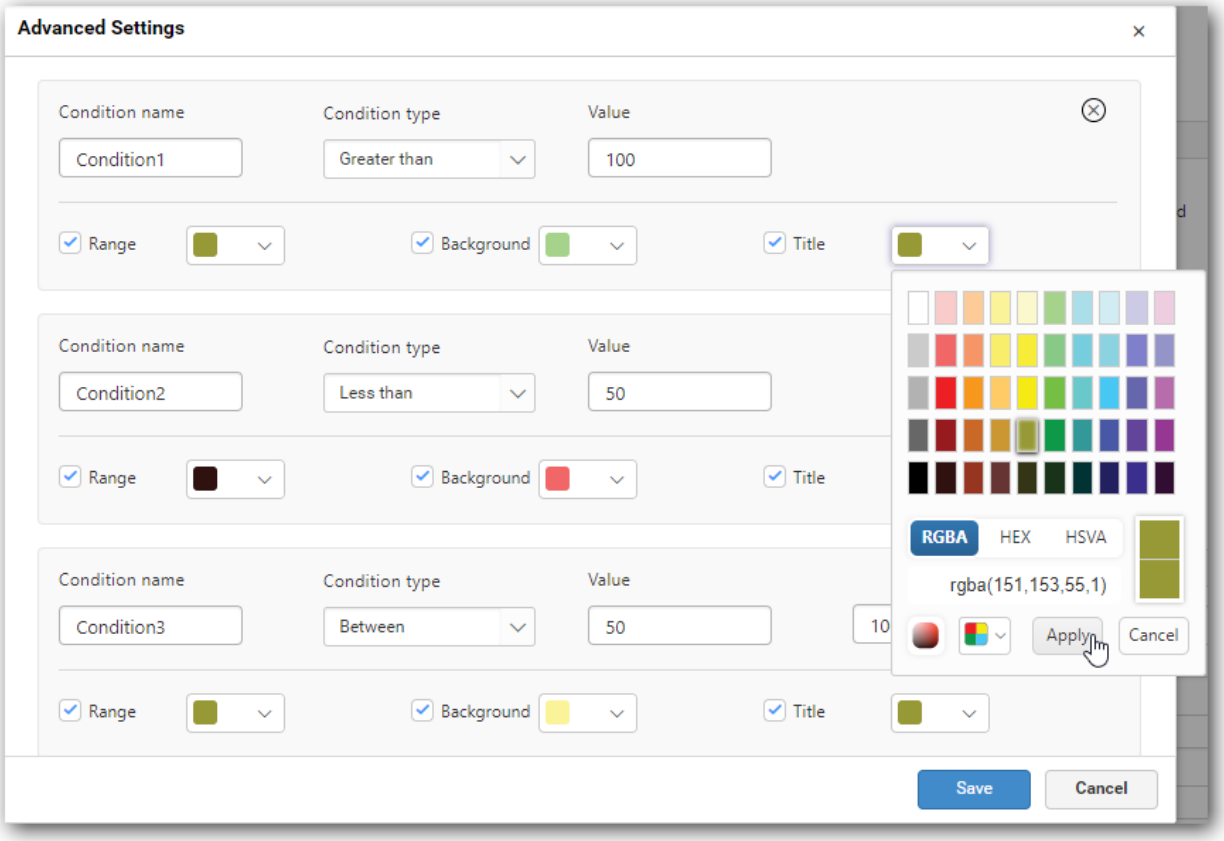
Color of elements in card widget can be customized using conditional formatting support which is available in the properties section of the card widget. This will allow the user to improve the visualization in card and to distinguish the data based on conditional range values that will let the visualizer to understand what is shown in data.



Advanced Settings

This allows you to provide colors for the selected measure based on single or multiple conditions available.

You can choose a card color using multiple condition sets such as Greater than, Less than, Equal to, Not Equal to, Between, Not between, Greater than or equal to, Less than or equal to.



As shown in the above image, there are three color pickers namely Range, Background and Title that will allow us to choose the colors for primary value and indicator, card background and title respectively.

On clicking the save button, color will be applied based on the conditions configured.

Note: The card widget need to be configured with actual and target values to apply the conditional formatting. Since, the color is customized based on the primary value.

Combo Box

Combo Box enables you to filter based on single or multiple items selection in dropdown list. To bind a combo box, a minimum requirement of 1 column is needed.

How to configure table data to combo box?

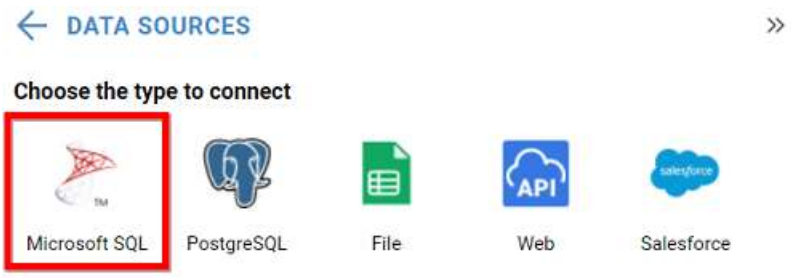
The following procedure illustrates data configuration of Combo Box.

Drag and drop Combo Box widget from the tool box into design panel and resize into your required size. You can find widget in tool box by search.

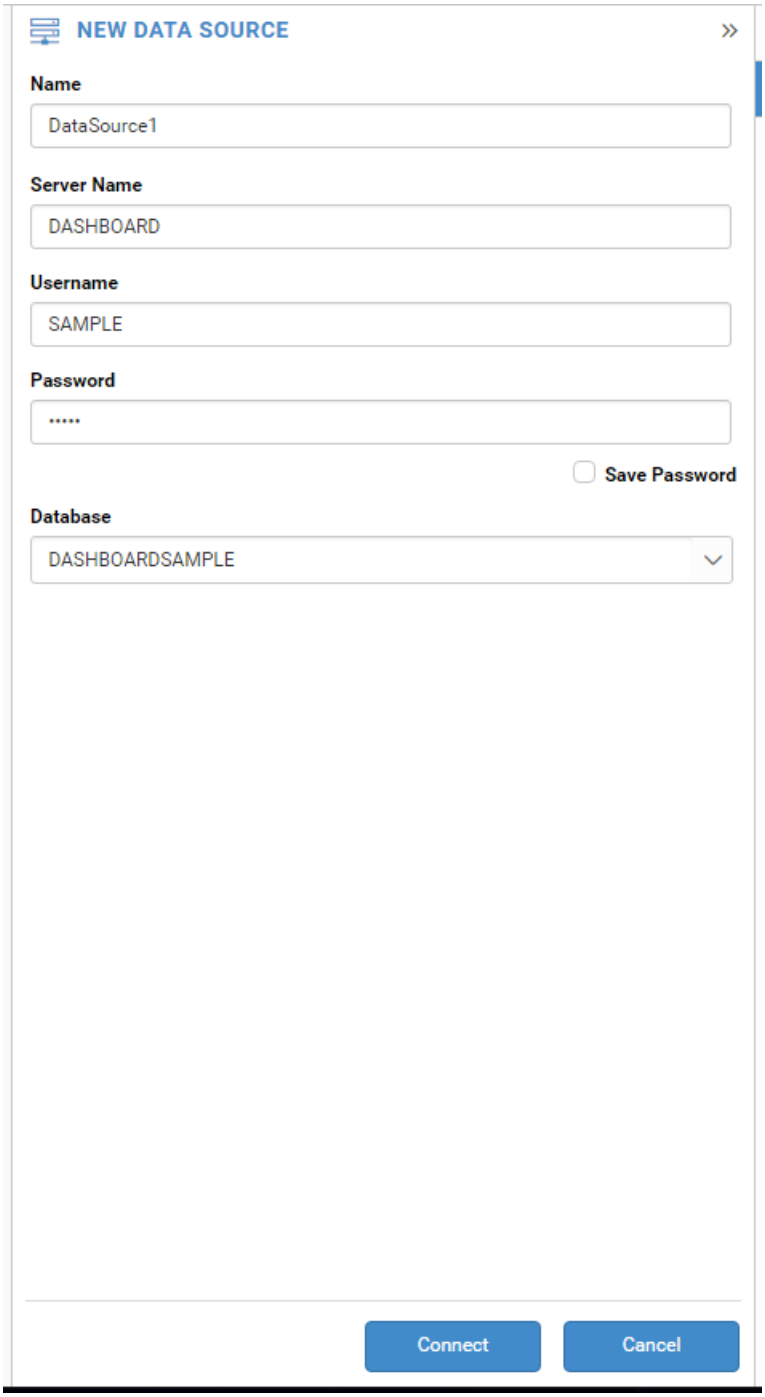
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

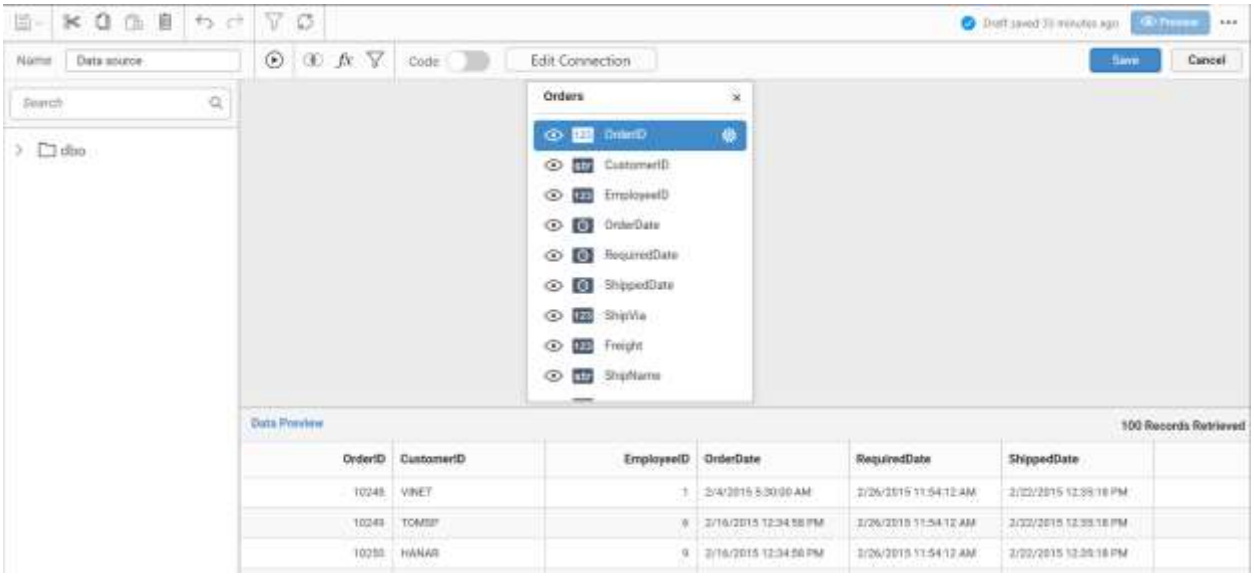
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



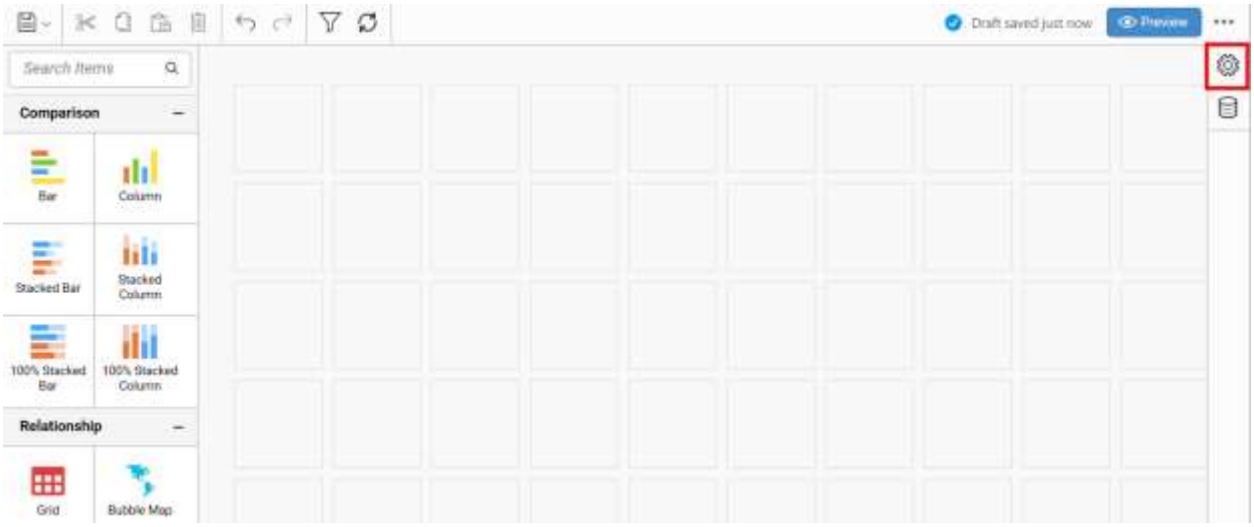
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



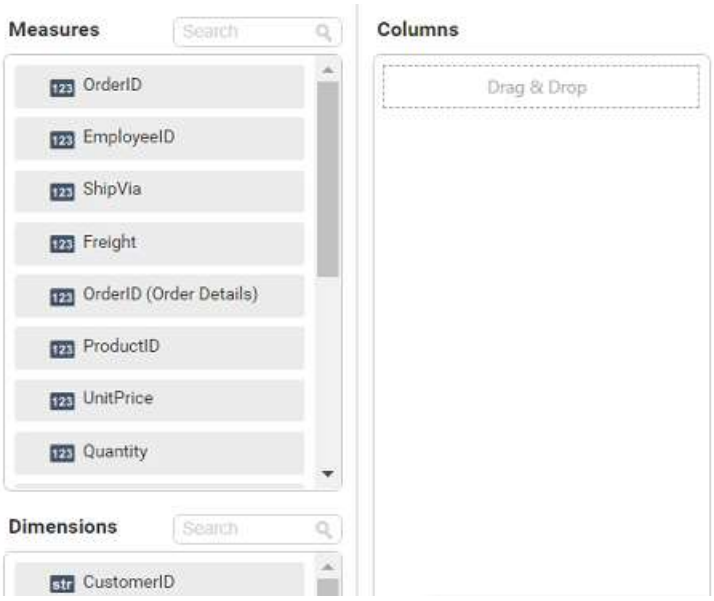
Drag your preferred table or view from the left pane from data design view, click Save button.



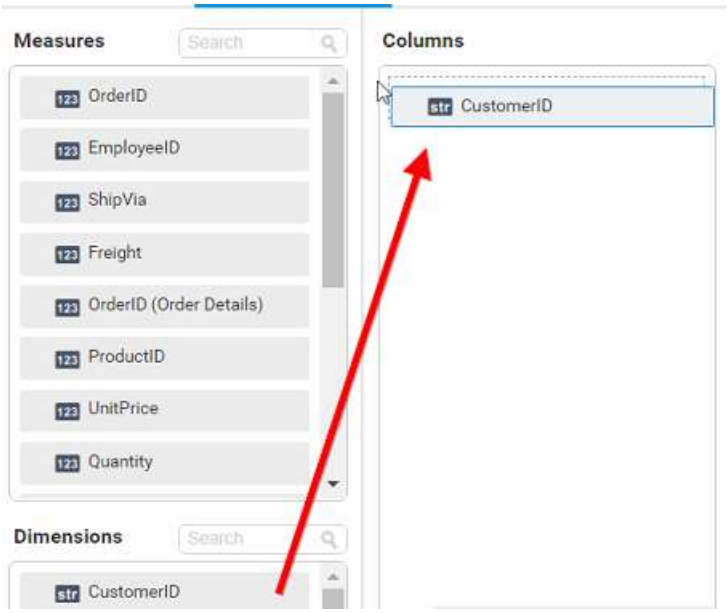
Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.



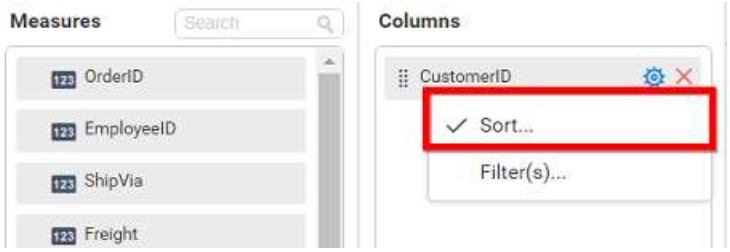
The data tab will be opened with available measures and dimensions from the connected data source



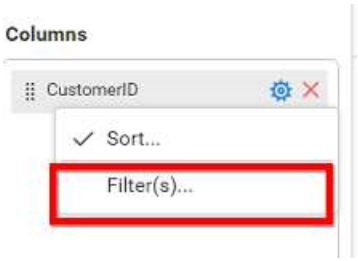
Drag and drop a column from Measures or Dimensions to Columns section.



Define the Sort of the dropped column through Sort option in the Settings drop down menu. For more details refer Sort.



Define filter criteria through the Filter(s) menu item in the Settings drop down menu. For more details, refer filter.



Clear the filters by selecting the Show All Records in the Settings dropdown menu.

How to format combo box?

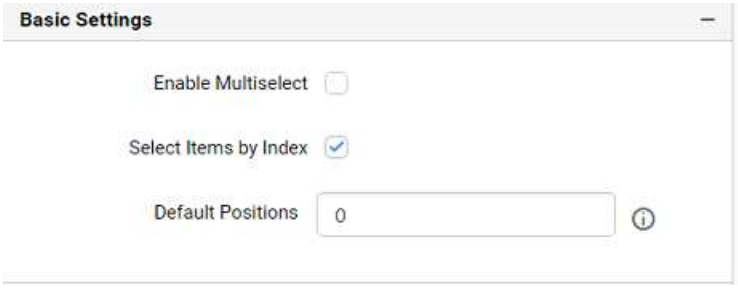
You can format the combo box for better illustration of the view that you require, through the settings available in Properties tab. This pane can be opened from design view through clicking the Settings icon at top right corner of the widget.

General Settings

Name

This allows you to set title for this combo box widget.

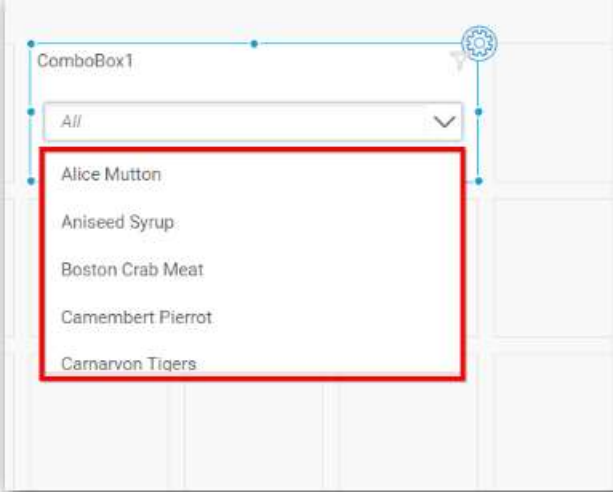
Basic Settings



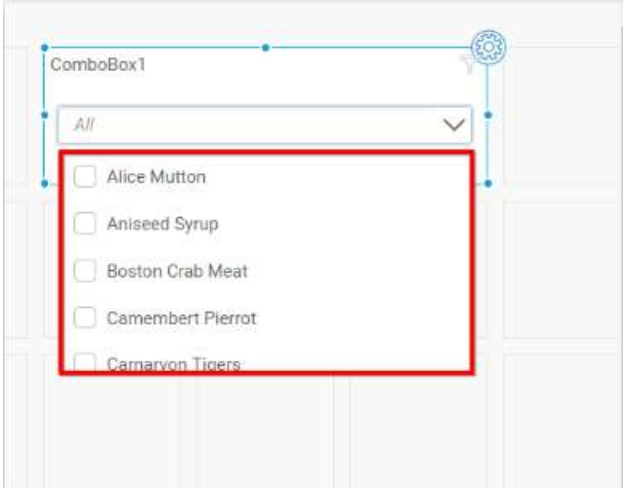
Enable Multi-select

This allows you to define single/multiple item selection in dropdown list.

Single Selection



Multiple Selection



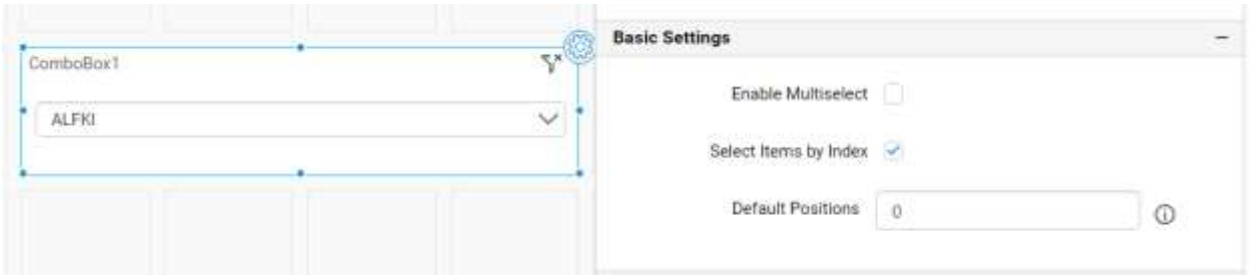
Select First Item By Default

This allows you to select the data value present in drop-down and it will be applicable only for single selection.

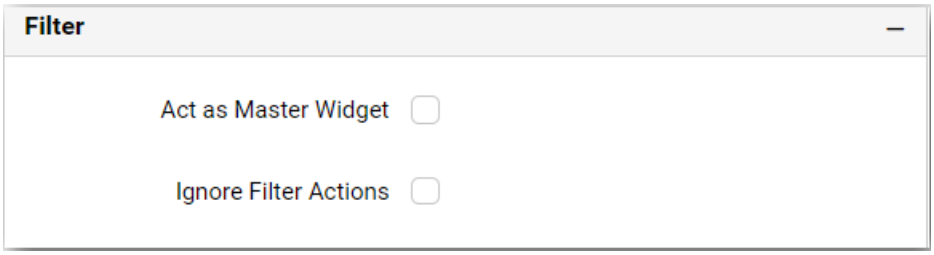
Default Positions

This allows you to set single index values for data in the drop-down. By default, zeroth index will be set.

Note: Supports only positive integer values starting from 0.



Filter



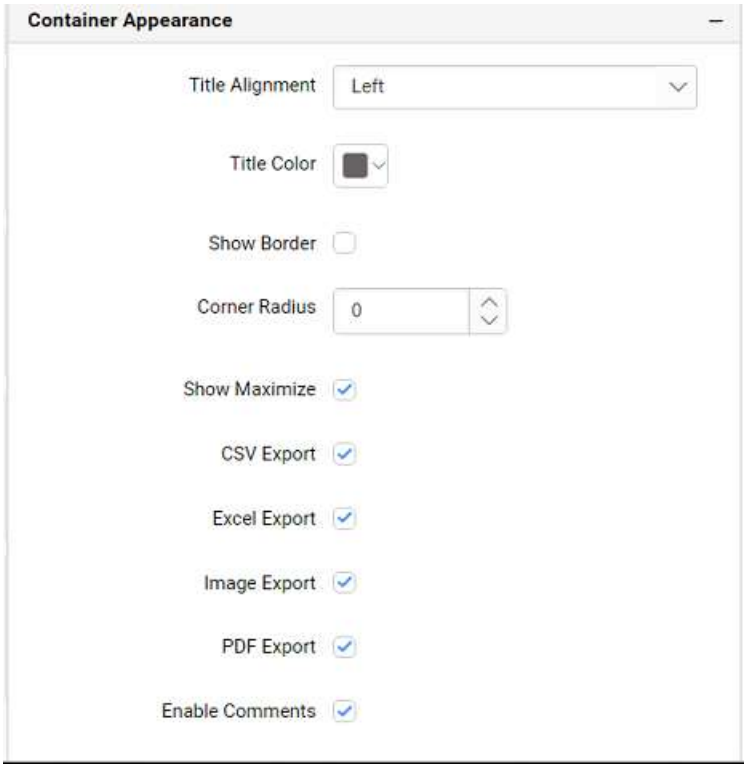
Act as Master Widget

This allows you to define this combo box widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this combo box widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

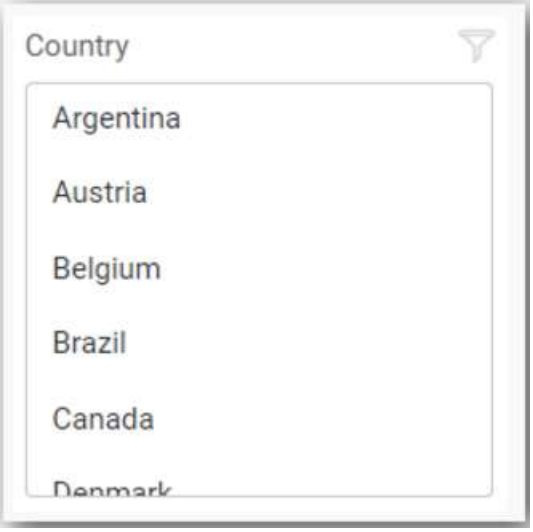
This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

List Box

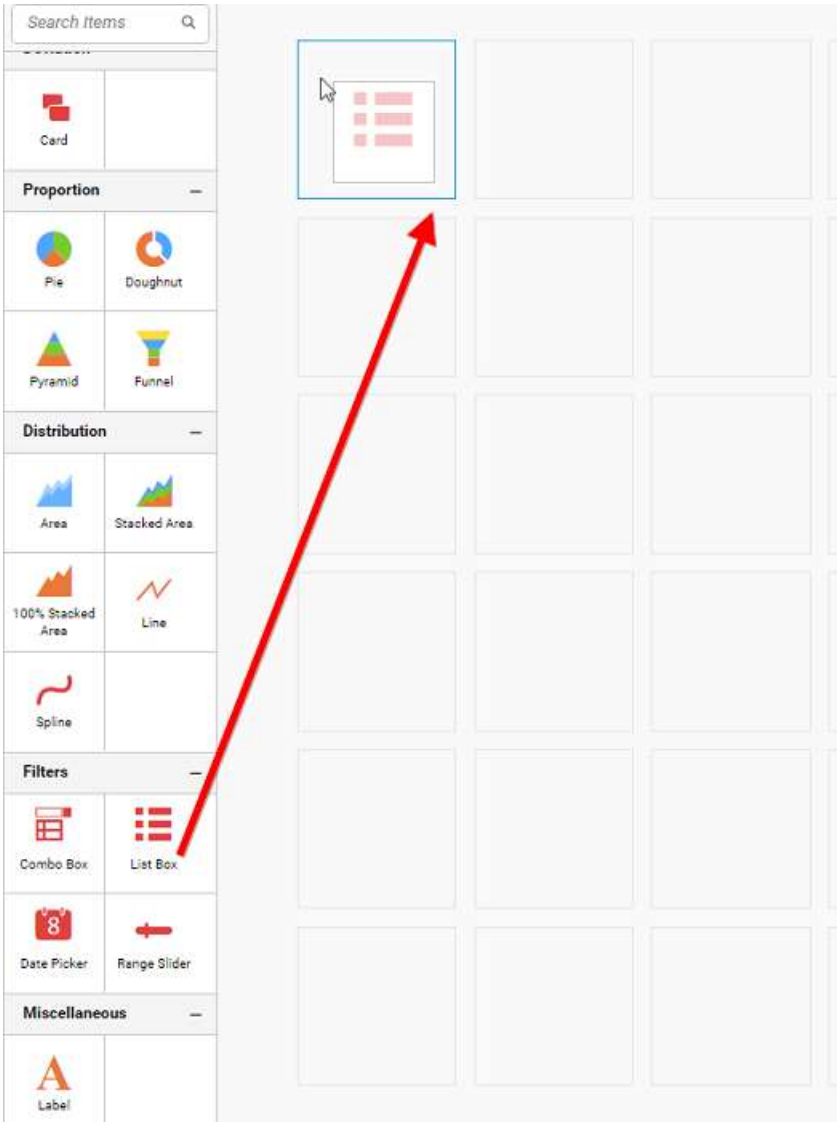
List Box enables you to filter based on single or multiple items selection in a list. To configure a list box, a minimum requirement of 1 column is needed.



How to configure the table data to list box?

The following procedure illustrates data configuration of List Box.

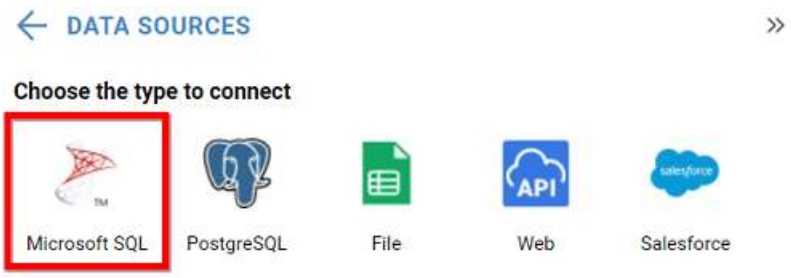
Drag and drop List Box control icon from the tool box into design panel. You can find control in tool box by search.



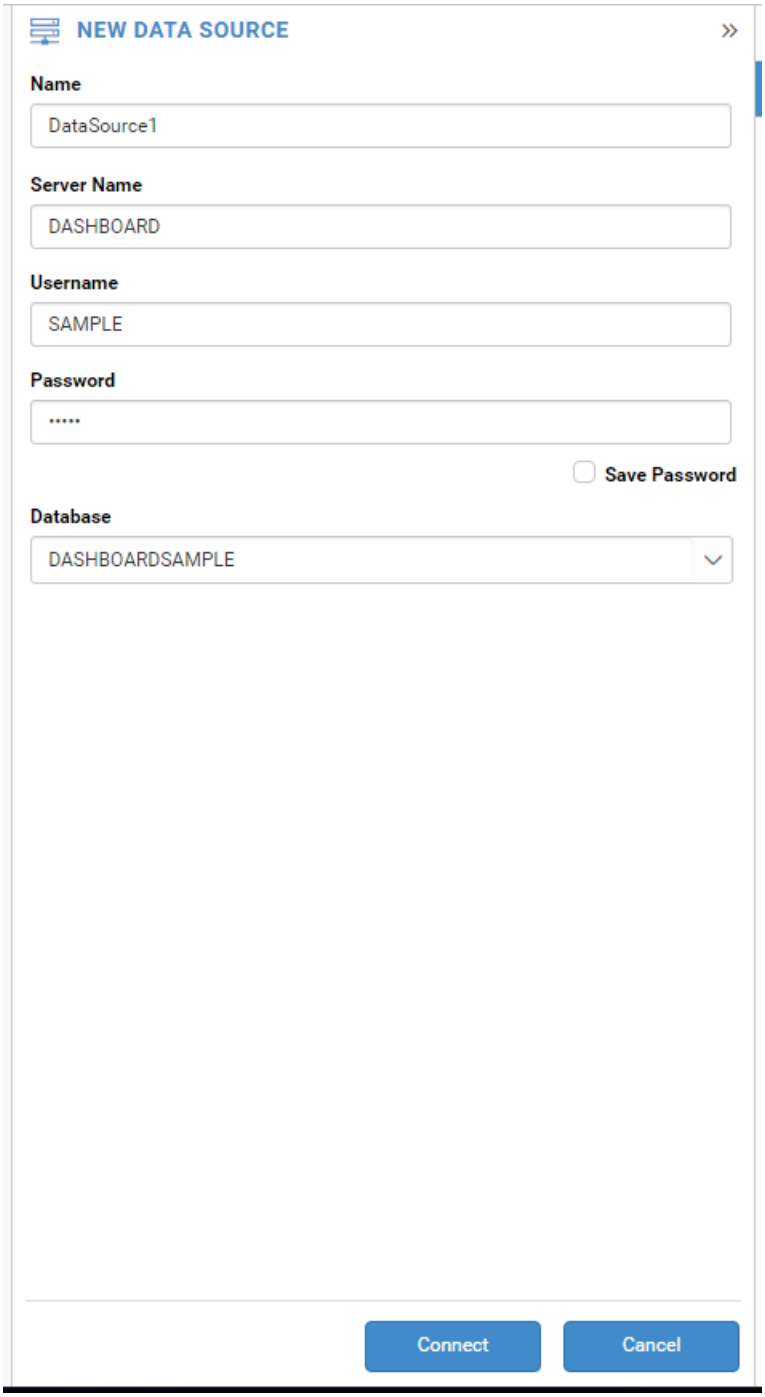
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a New Connection from connection type panel.

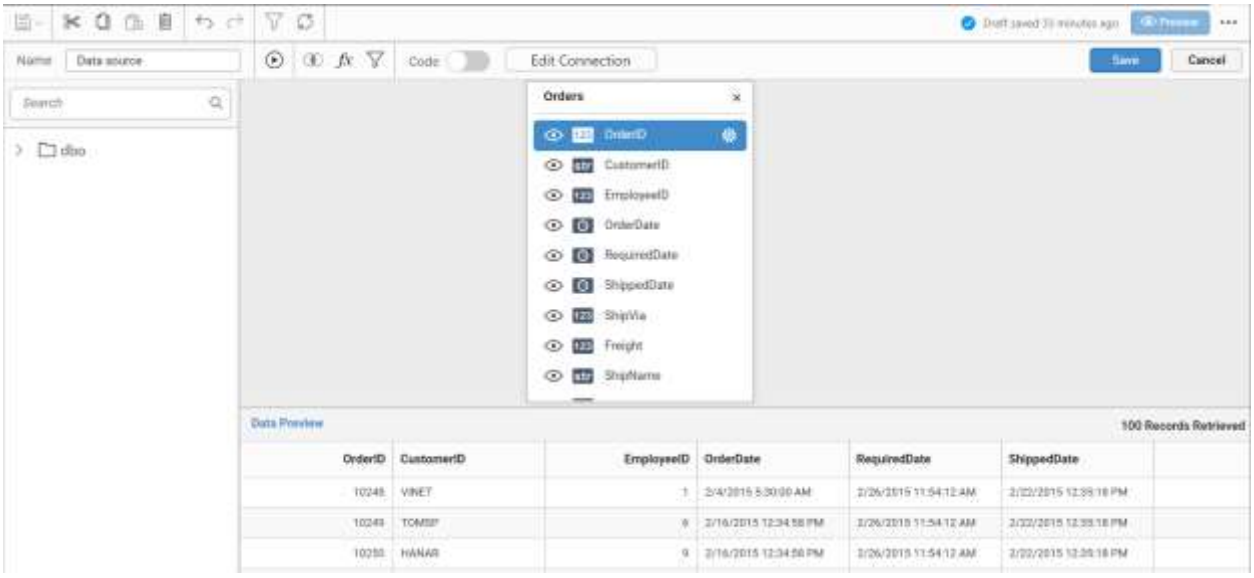
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



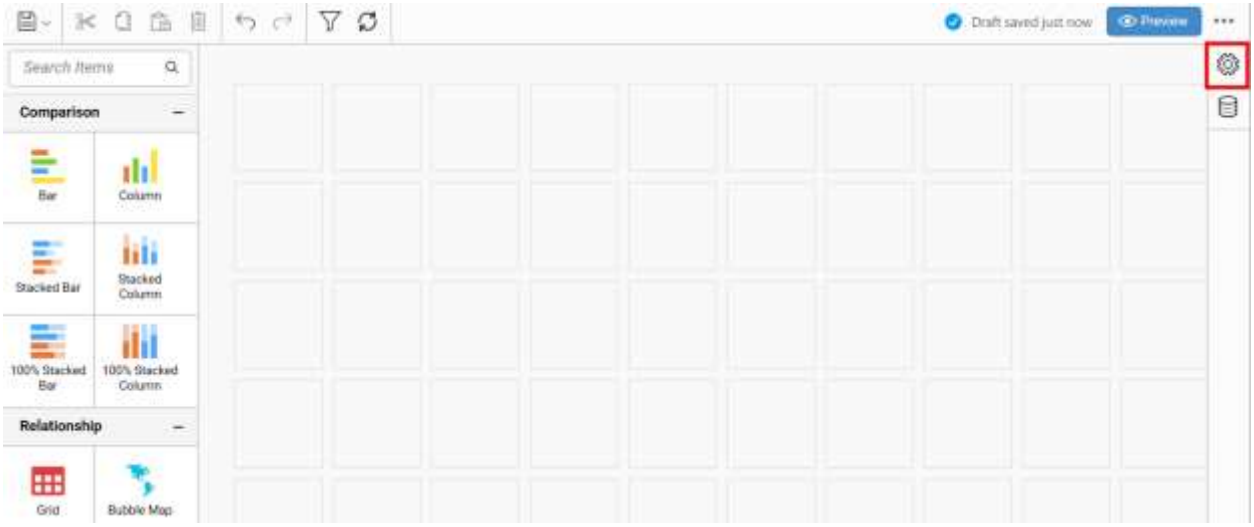
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button



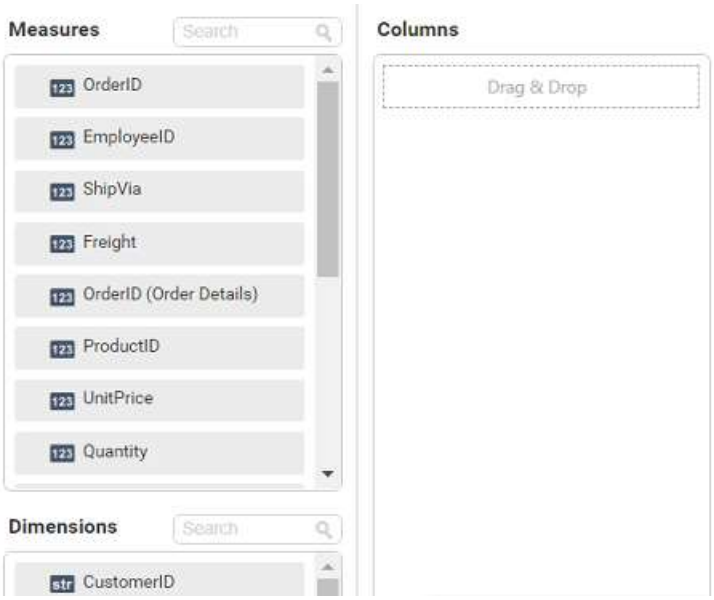
Drag your preferred table or view from the left pane from data design view, click Save button.



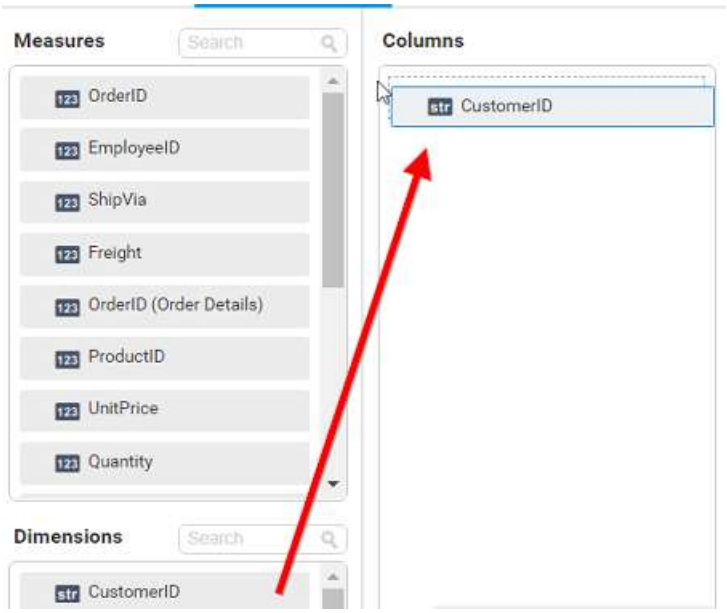
Click Properties button in Configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.



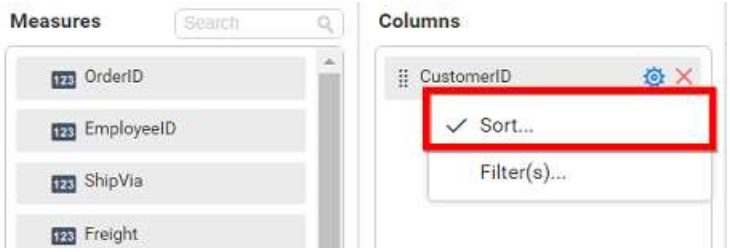
The data tab will be opened with available measures and dimensions from the connected data source



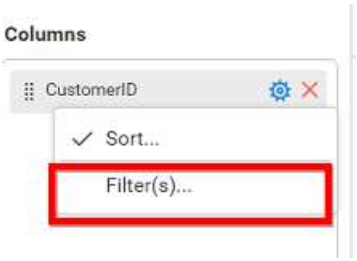
Drag and drop a column from Measures or Dimensions to Columns section.



Define the Sort of the dropped column through the Sort option in the Settings drop down menu. For more details refer Sort



You can use the filters by selecting the Filter(s) option. For more details, refer filter.



You can clear the filters by selecting the Show All Records option.

How to format list box?

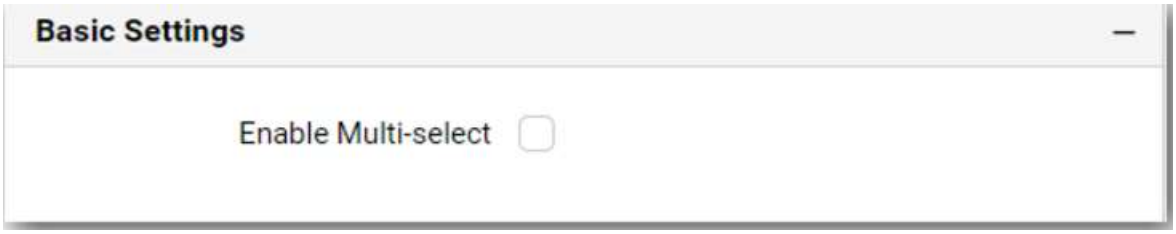
You can format the List box for better illustration of the view that you require, through the settings available in Properties tab. This pane can be opened from design view through clicking the Settings icon at top right corner of the widget.

General Settings

Name

This allows you to set title for this list box widget.

Basic Settings

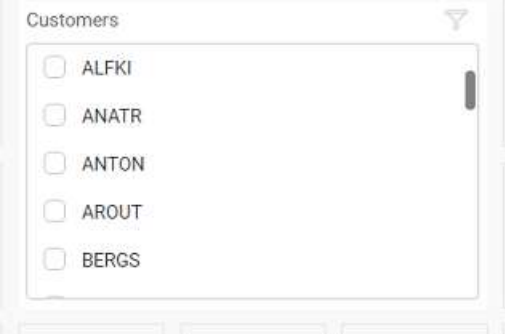


Enable Multi-select

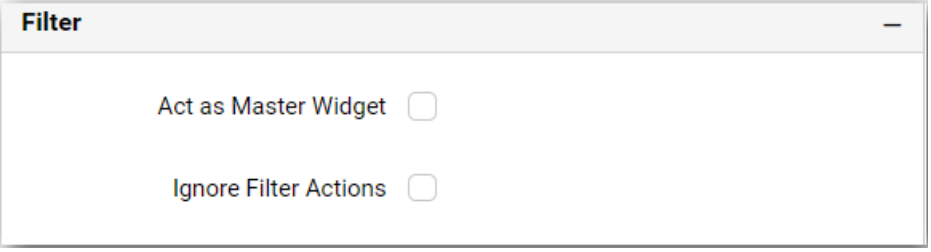
This allows you to define single/multiple item selection in List Box.

Single Selection

Multiple Selection



Filter



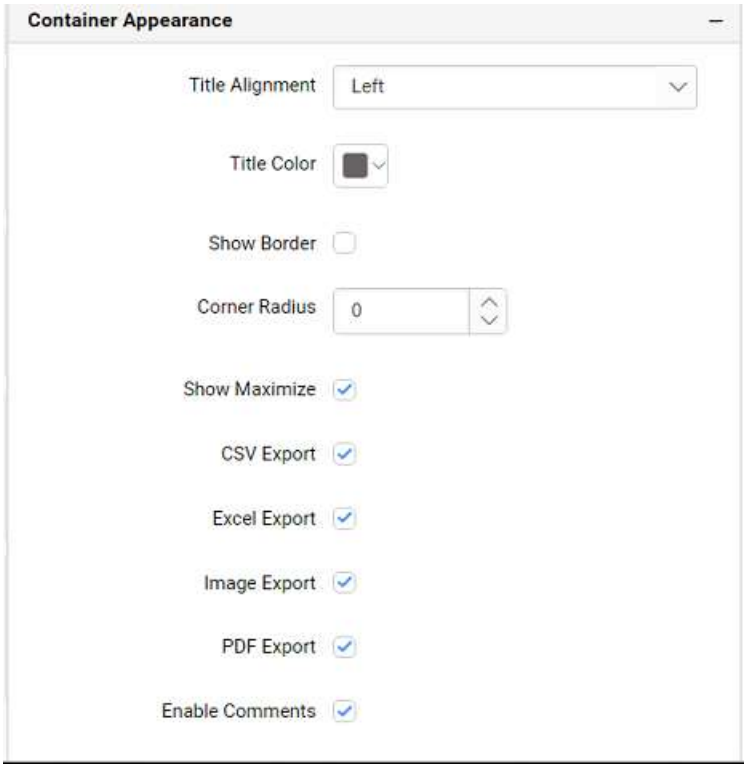
Act as Master Widget

This allows you to define this list box widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this list box widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comments

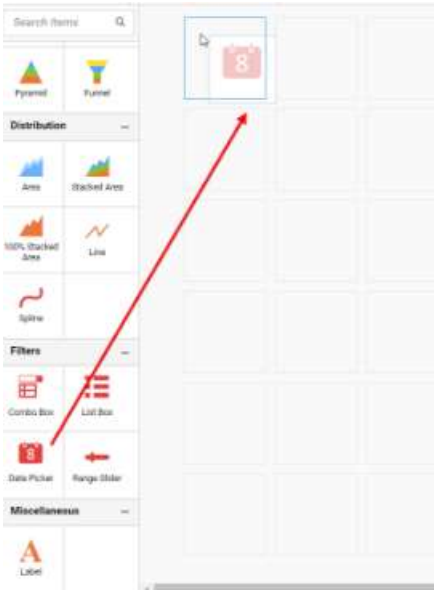
This allows you to enable/disable comment for dashboard widget. For more details refer here

Date Picker

Date Picker enables you to filter based on the single or range of date selection.

How to configure table data to date picker?

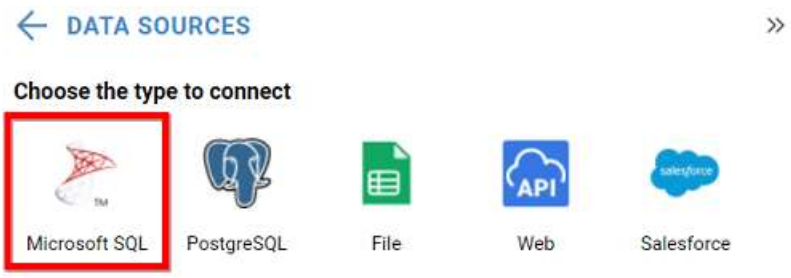
Drag and drop the Date Picker from toolbox at left into design canvas and resize it to your required size.



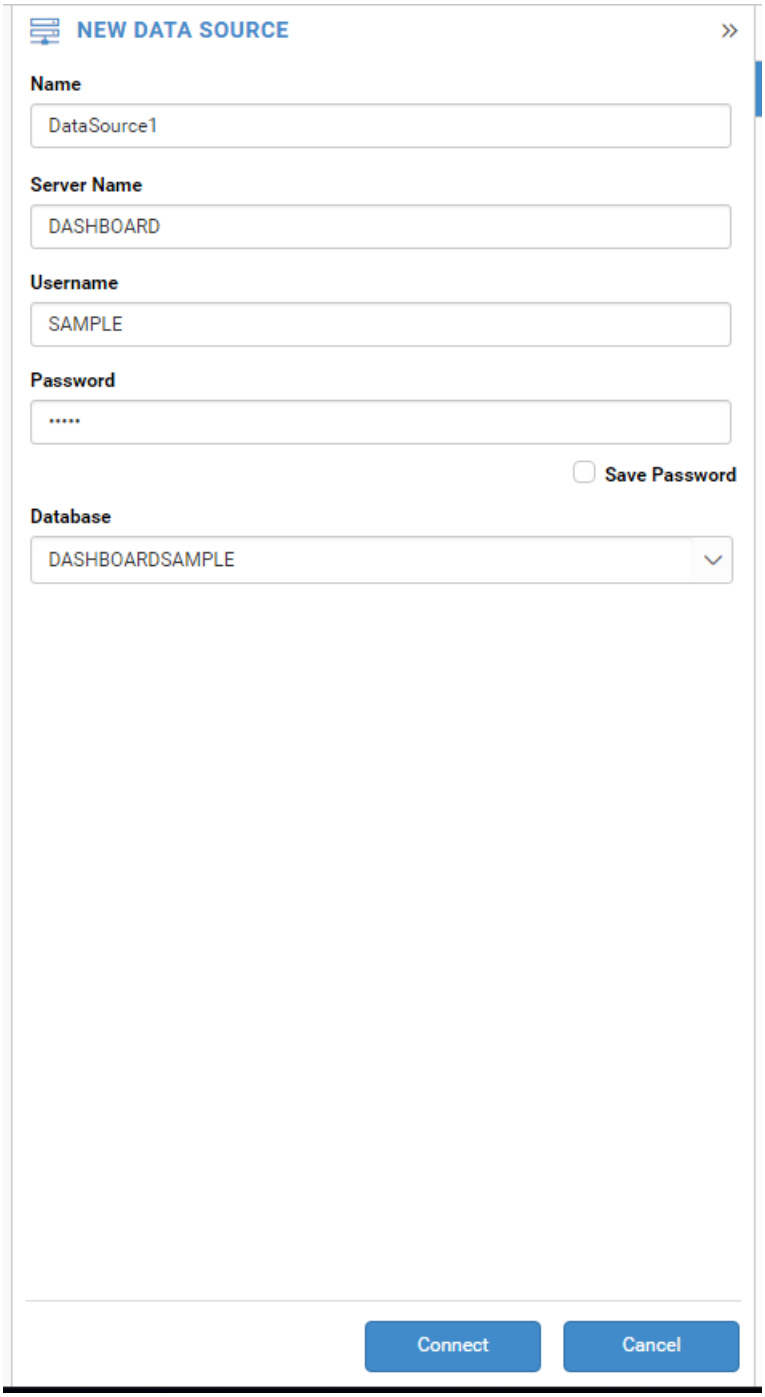
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

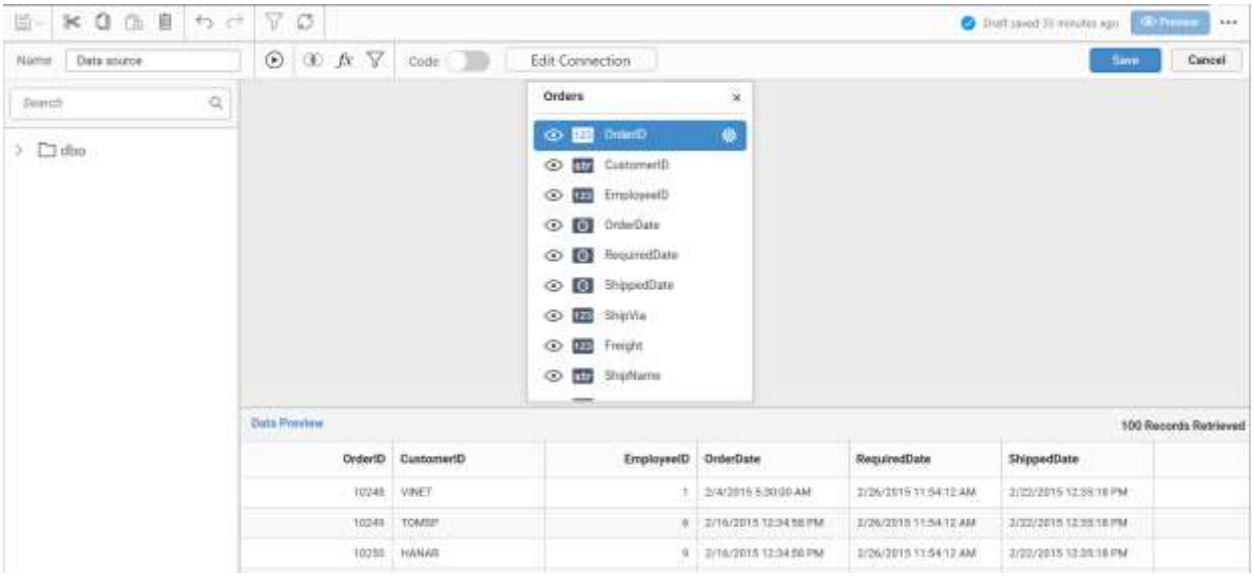
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



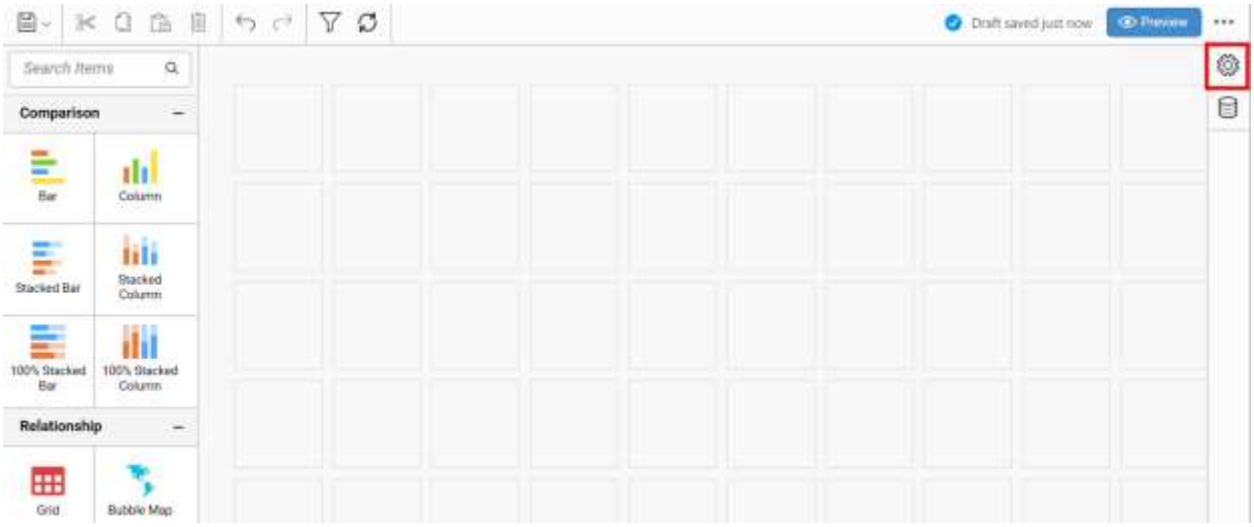
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

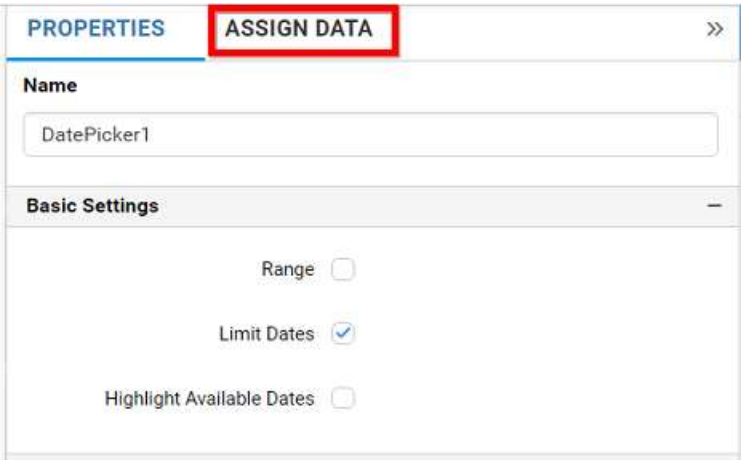


Drag your preferred table or view from the left pane from data design view, click Save button.

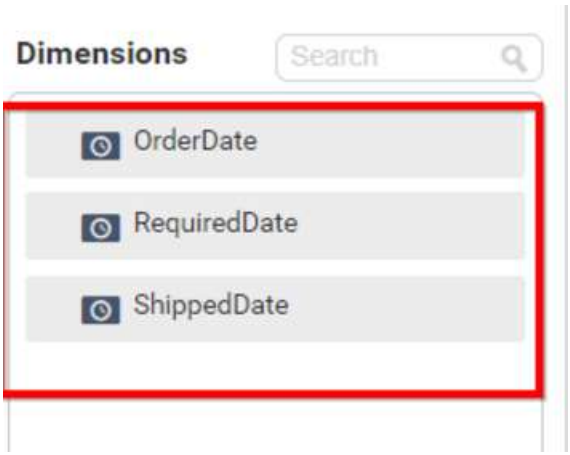


Click Properties button in configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.

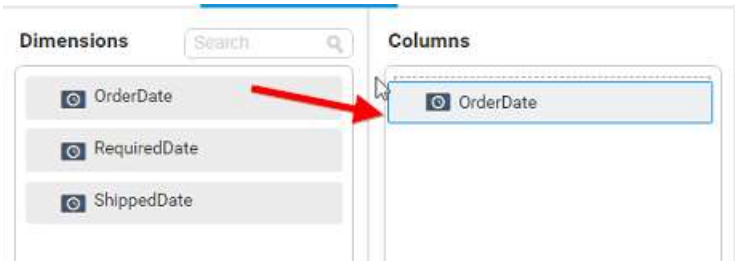




The data tab will be opened with available measures and dimensions from the connected data source

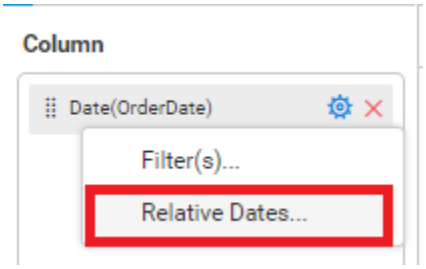


Drag and drop a date field from Dimensions into Columns section.

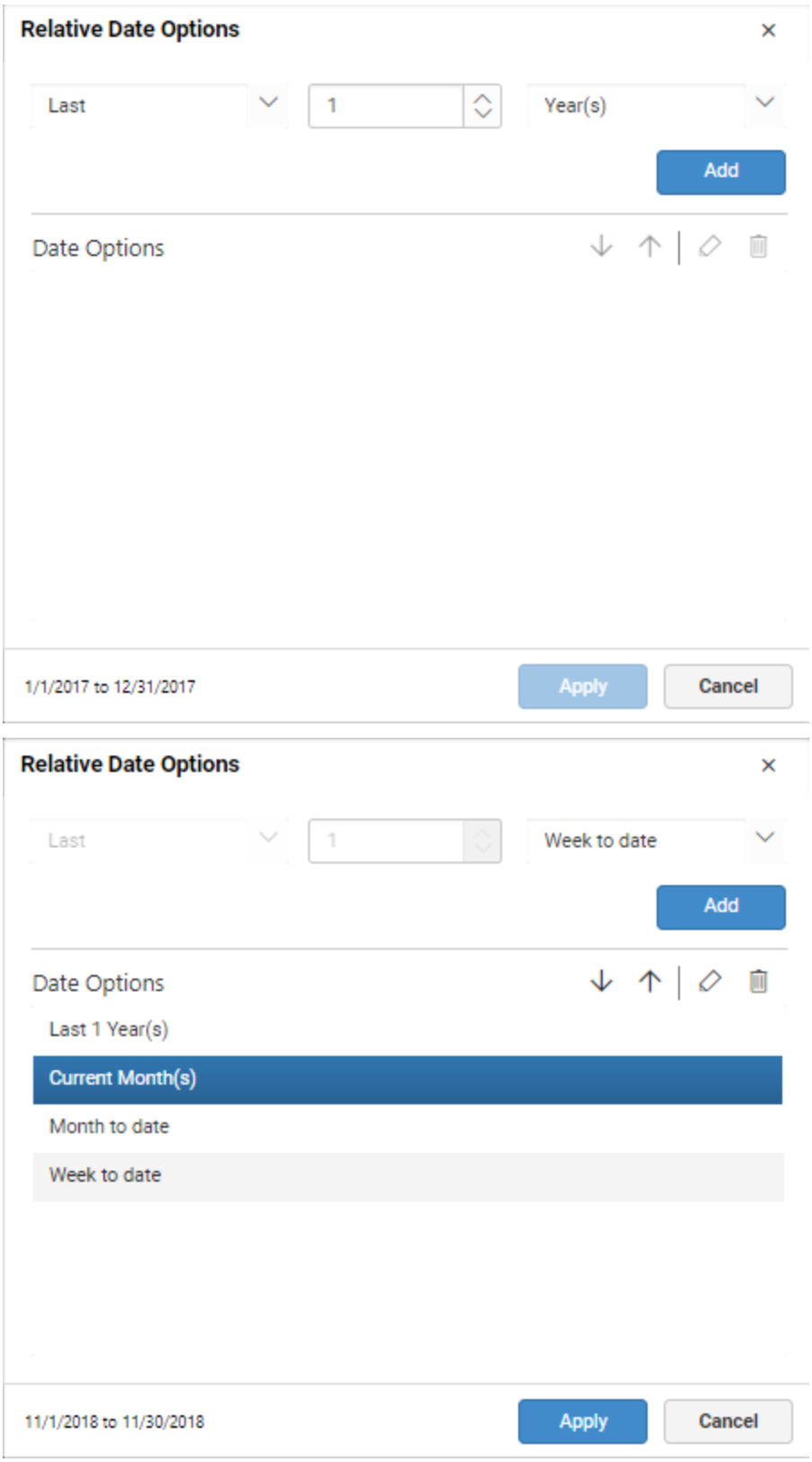


How to configure relative dates to DatePicker?

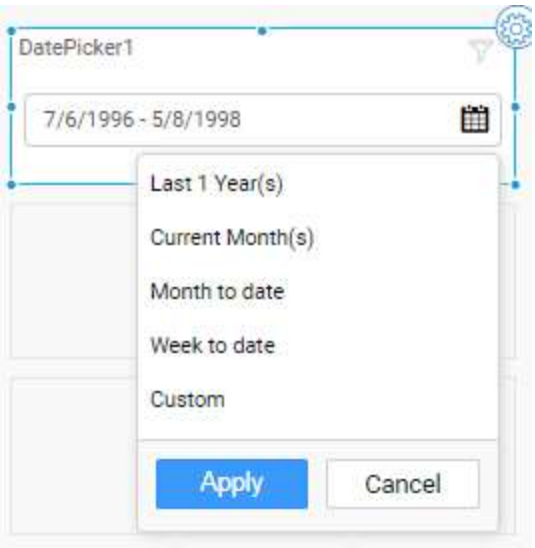
Switch to the Properties pane and set the Selection Type as Range; Switch back to the Data pane; Click the Settings icon in the dropped date column and select Relative Dates... in the drop-down menu.



In the launched Relative Date Options window, configure the relative date and click Add. Repeat this process till the required set of relative dates are added.

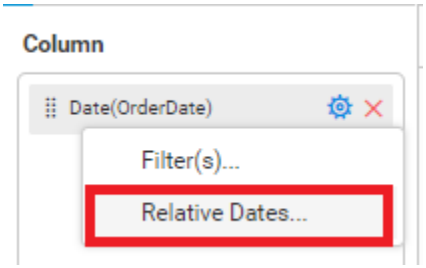


You can see the added relative dates in the DatePicker as follows.

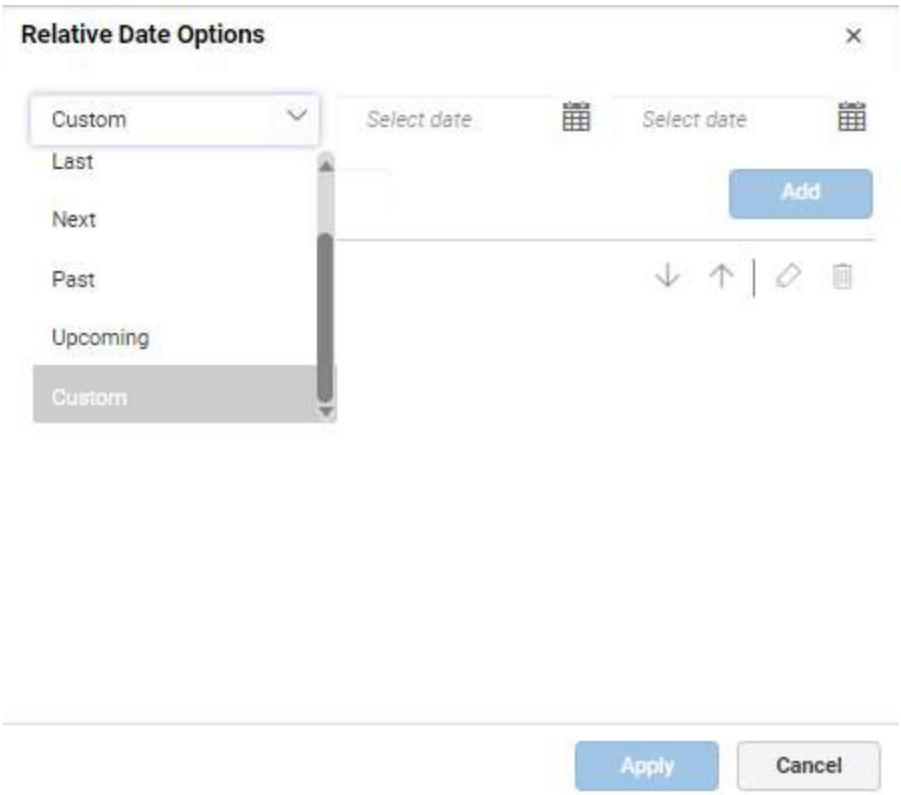


How to configure custom relative dates to the DatePicker?

Switch to the Properties pane and set the Selection Type as Range, and then switch back to the Data pane; Click the Settings icon in the dropped date column and select Relative Dates... in the drop-down menu.

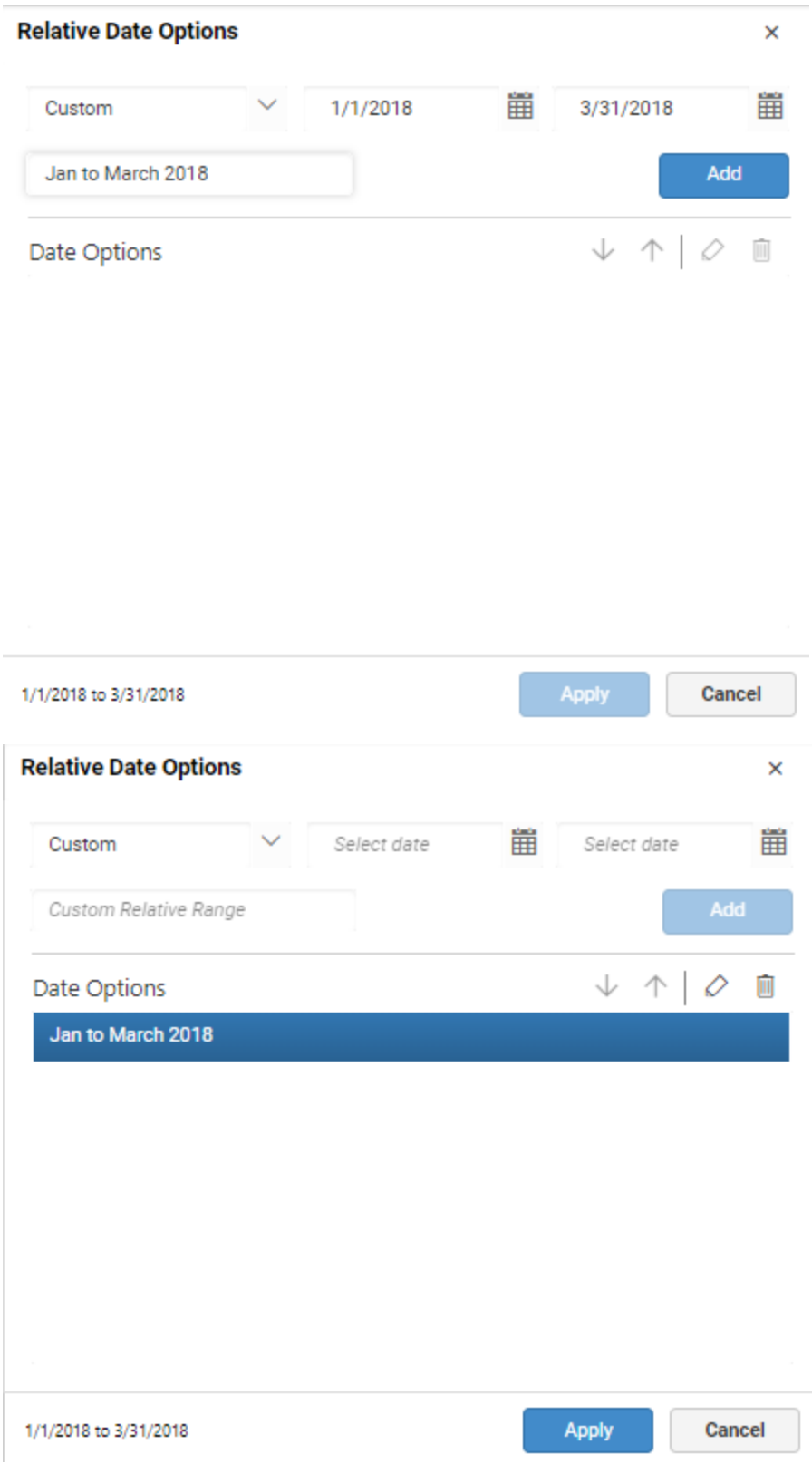


In the launched Relative Date Options window, select the Custom option from drop-down List.

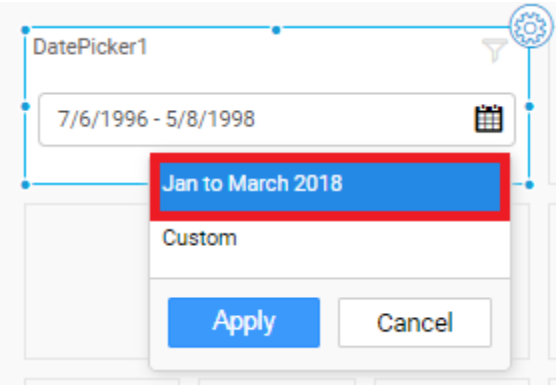


Choose the start and end dates you like to set as custom range, set a name for the custom range, and then click Add. Repeat this process till the required set of custom

relative dates are added.



You can see the added custom relative dates in the DatePicker as follows.



How to format date picker?

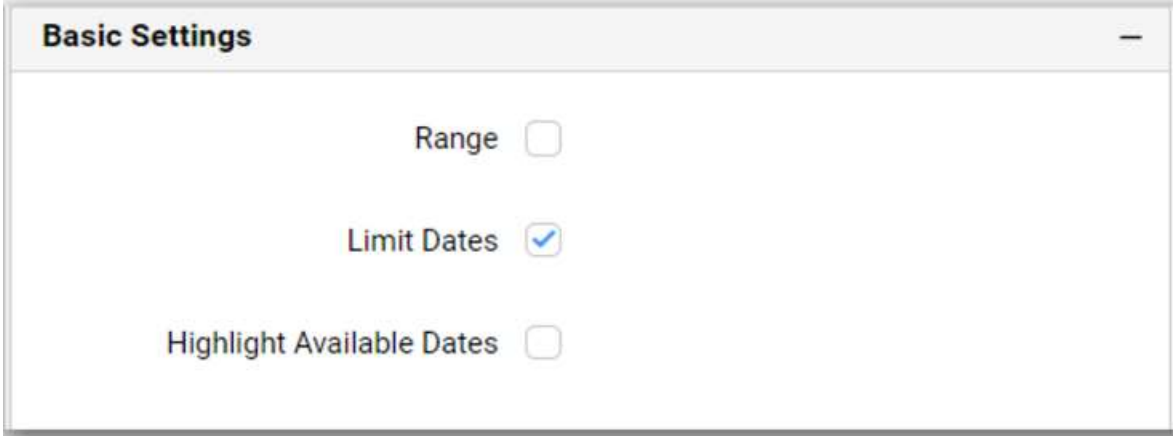
You can format the Date Picker for better illustration of the view that you require, through the settings available in Properties tab.

General Settings

Name

This allows you to set title for this Date Picker widget.

Basic Settings



Range

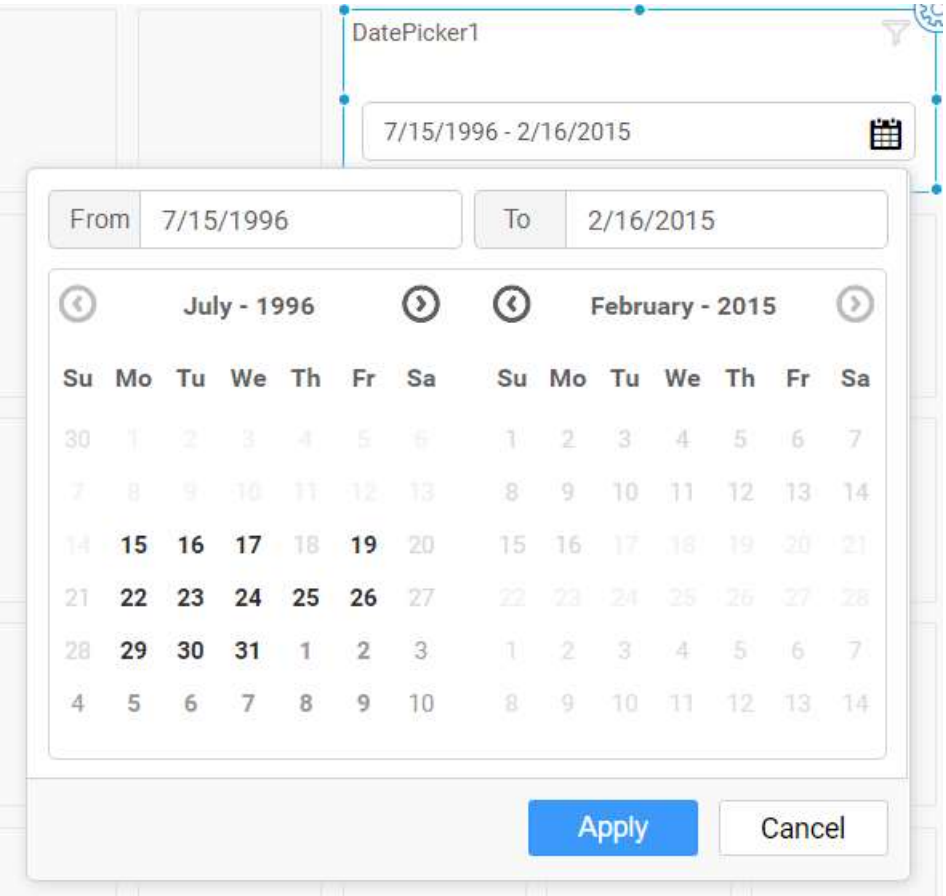
This allows you to toggle the selection type of Range.

Single – Single date can be bounded, if you disable the Range selection option.

Range – A range of dates (two dates) can be bounded,if you enable the Range selection option.

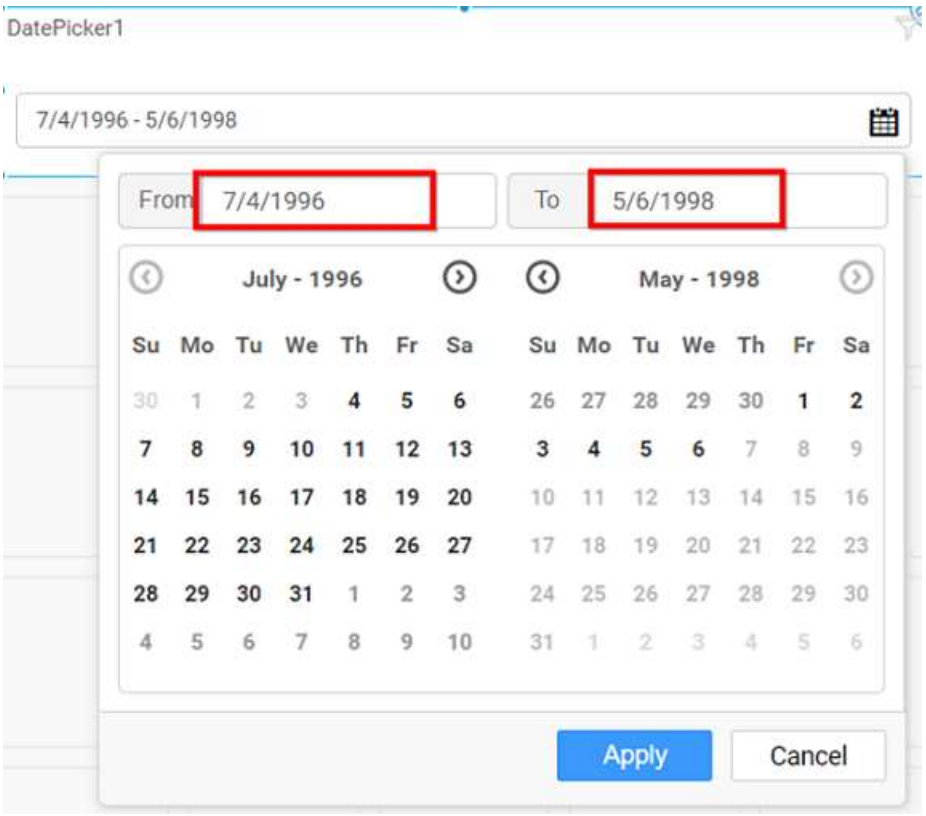
Highlight Available Dates

This allows you to enable the highlighting of available dates in date picker.

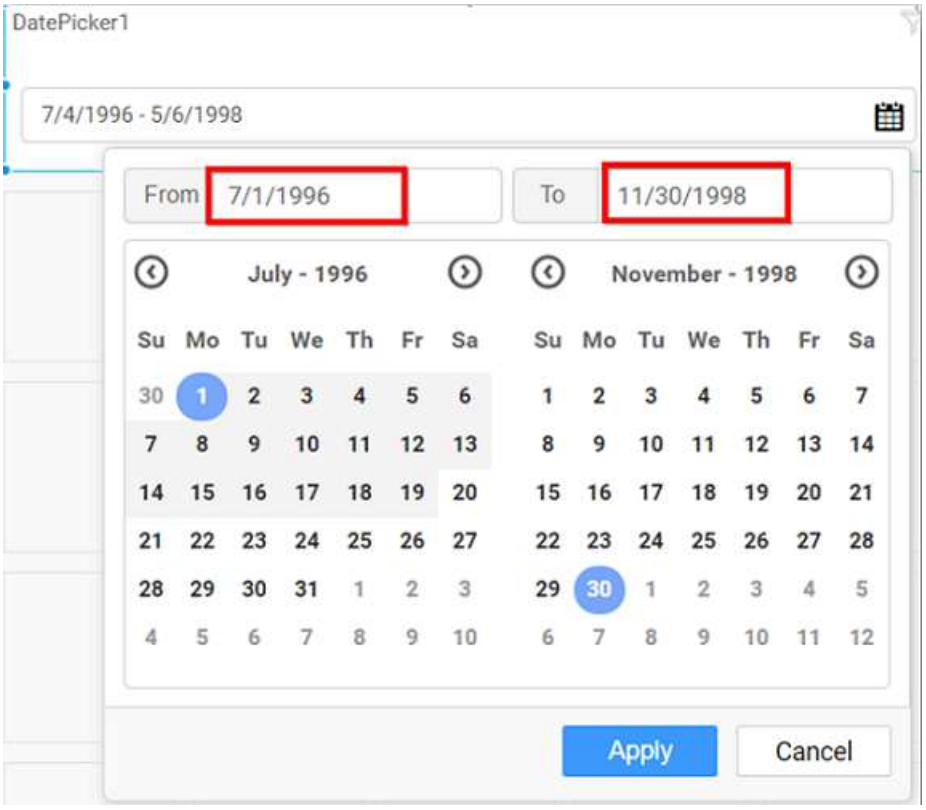


Limit Dates

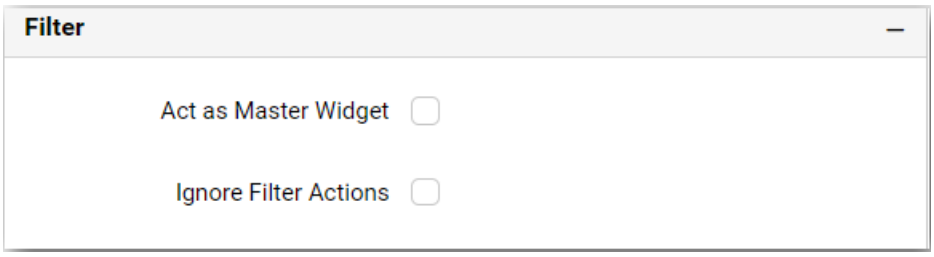
This allows you to display only the limited dates (dates in the datasource) in date picker, by default option will be enabled. If option is enabled you can select the dates available in the data source.



If you disable the Limit Dates option, there is no limit for the date range.



Filter



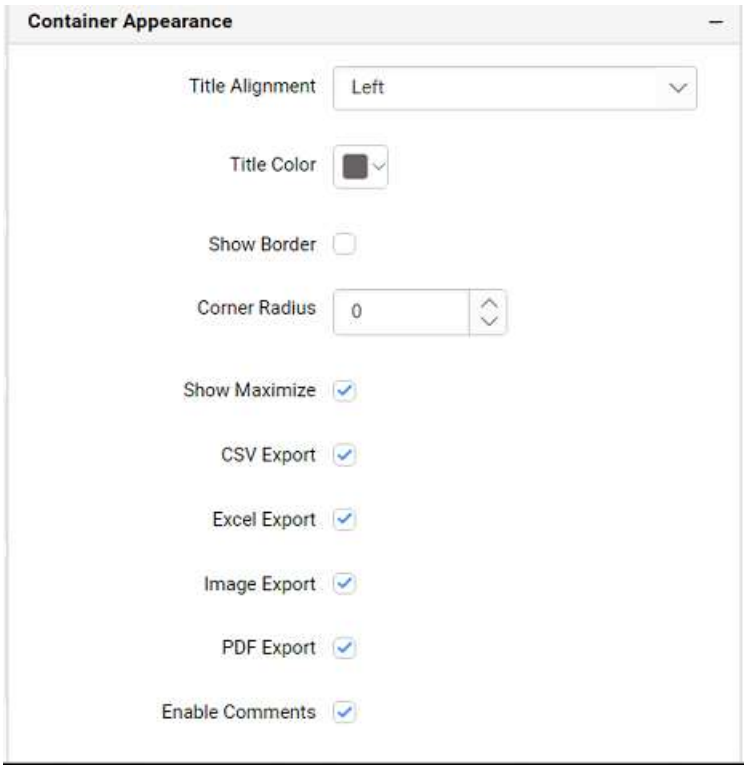
Act as Master Widget

This allows you to define this date picker widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this date picker widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

Range Slider

Range Slider enables you to filter based on value or date range set through sliders. To configure a range slider, a minimum requirement of 1 column is needed.

How to configure table data to range slider?

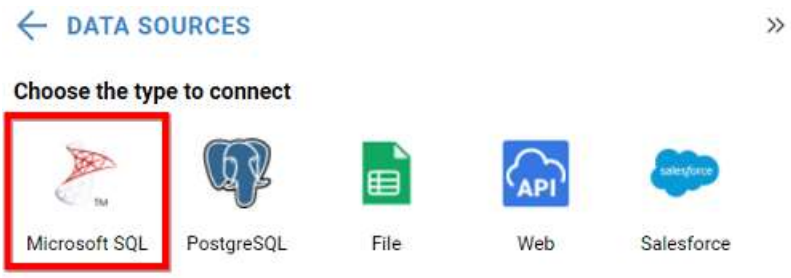
The following procedure illustrates data configuration of Range Slider.

Drag and drop Range Slider control icon from the tool box into design panel. You can find control in tool box by search.

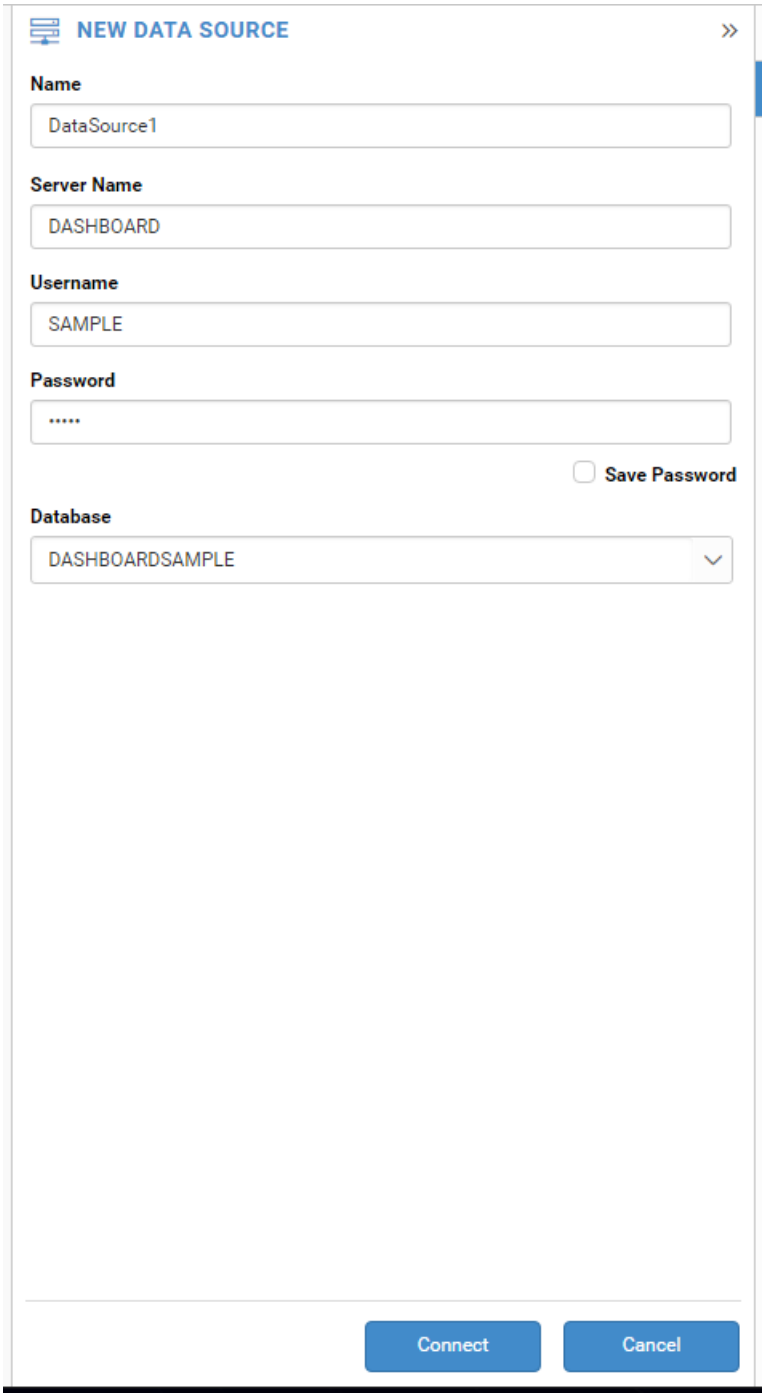
Click Data Source button in configuration panel.

Click CREATE NEW button to launch a new connection from connection type panel.

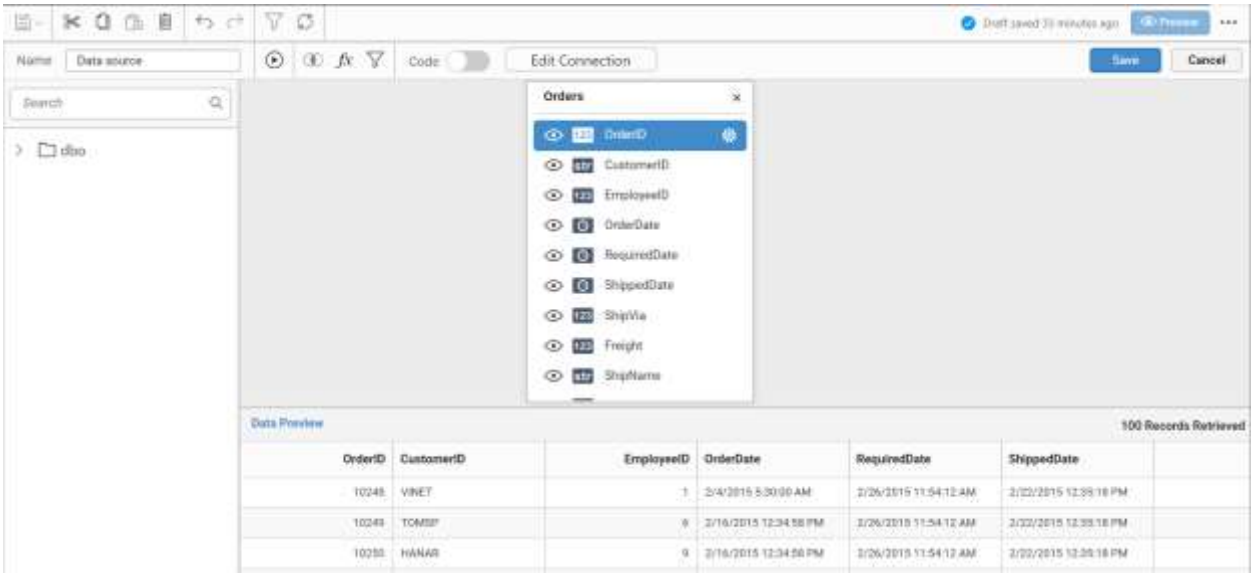
In the connection type panel, click any one (Here Microsoft SQL Connection type is selected for demonstration) of the listed connection type button shown.



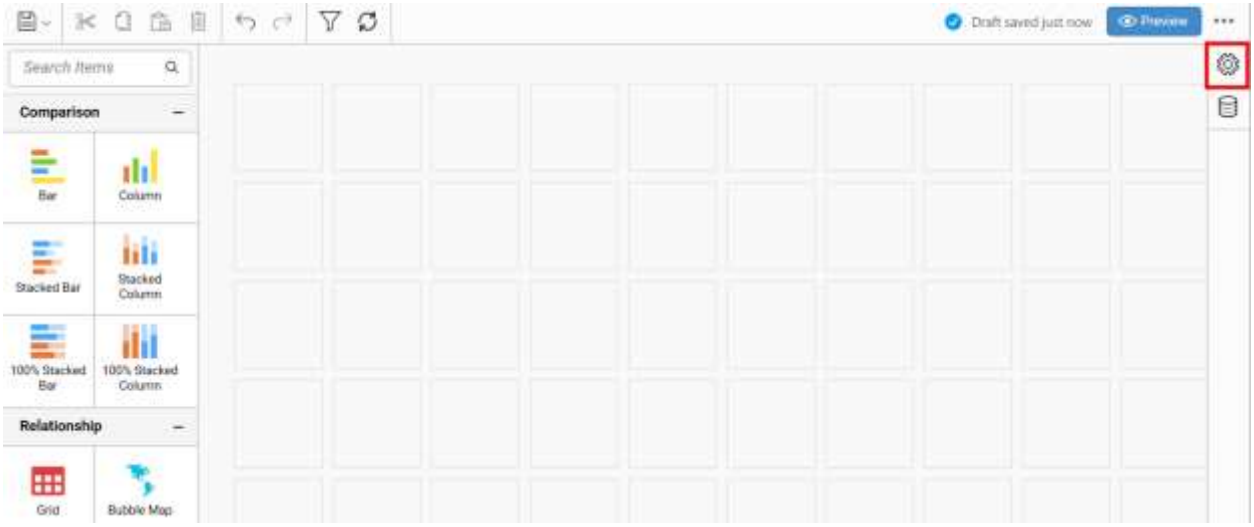
In the NEW DATA SOURCE configuration panel, fill the connection type and related details. Click Connect button

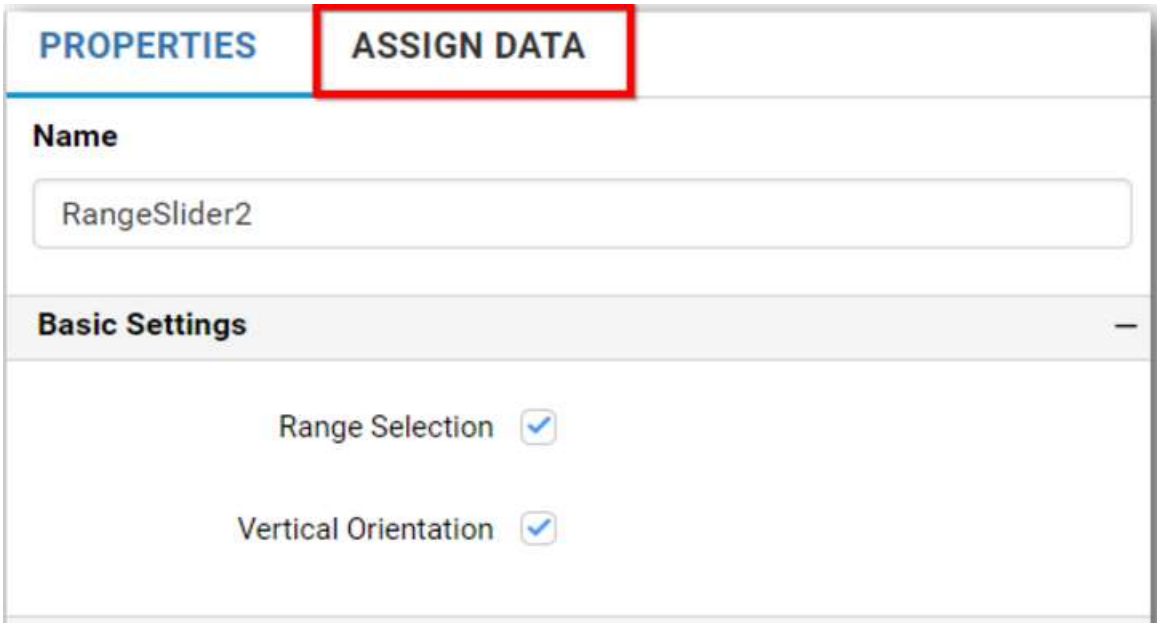


Drag your preferred table or view from the left pane from data design view, click Save button.

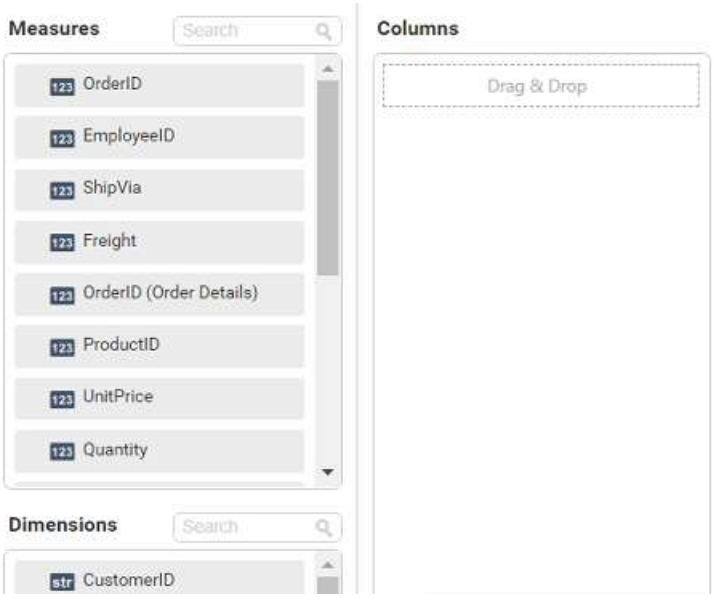


Click Properties button in Configuration panel, property pane opens. Now, Switch to ASSIGN DATA tab.

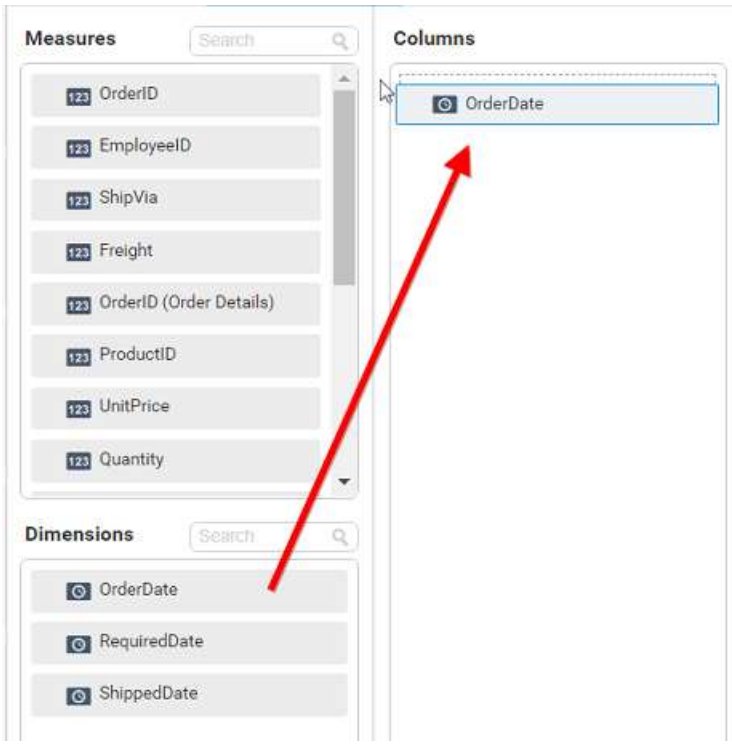




The data tab will be opened with available measures and dimensions from the connected data source



Bind column through drag and drop element from sections to Columns section.



Here is an illustration,

How to format range slider?

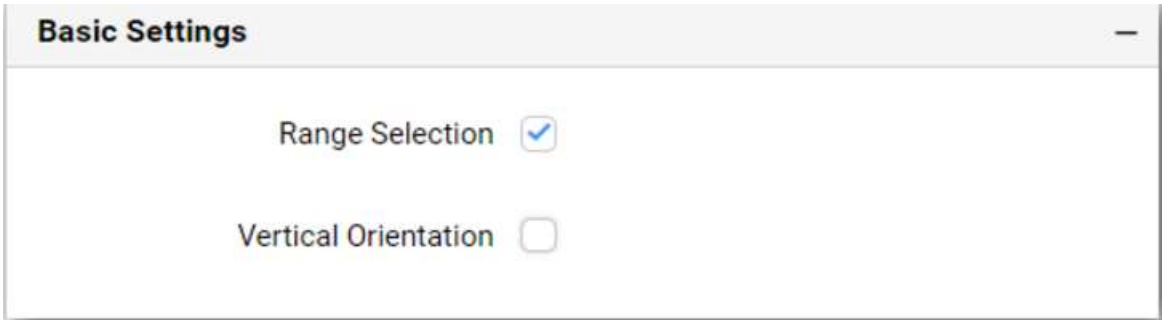
You can format the Range Slider for better illustration of the view that you require, through the settings available in Properties tab.

General Settings

Name

This allows you to set title for this range slider widget.

Basic Settings

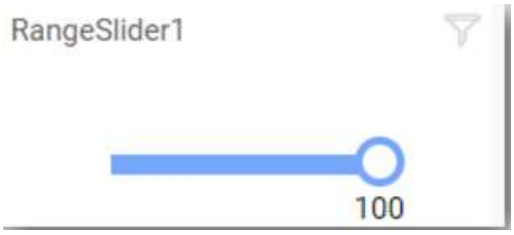


Range Selection

This allows you to toggle the selection type of Range Selection.

Single – Single value can be bounded, if you disable the Range Selection option.

Range Slider with Single Pointer



Range – A range (two values) can be bounded, if you enable the Range Selection option.

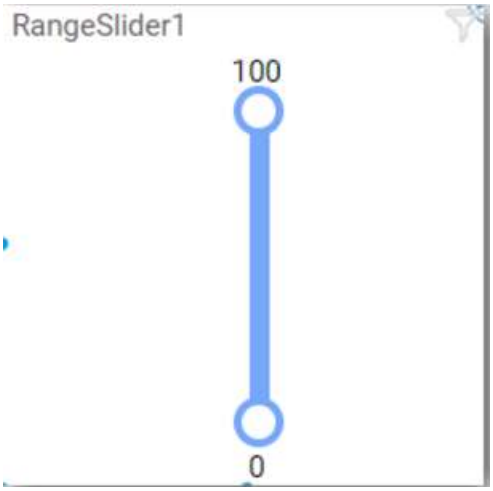
Range Slider with Range Pointer

Vertical Orientation

This allows you to toggle the orientation type of Vertical Orientation.

Vertical – Enable the Vertical Orientation option for vertical orientation.

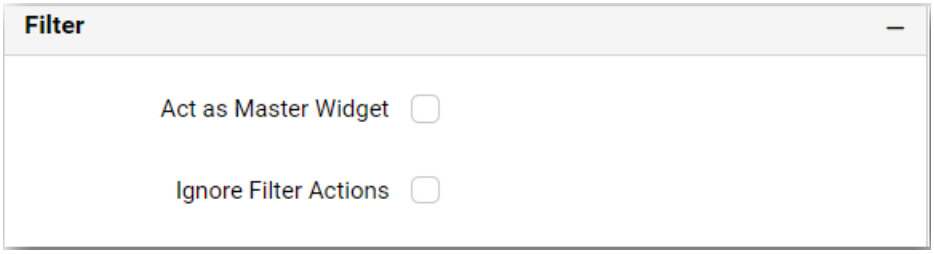
Range Slider with Vertical Orientation



Horizontal – Disable the Vertical Orientation option for horizontal orientation.

Range Slider with Horizontal Orientation

Filter



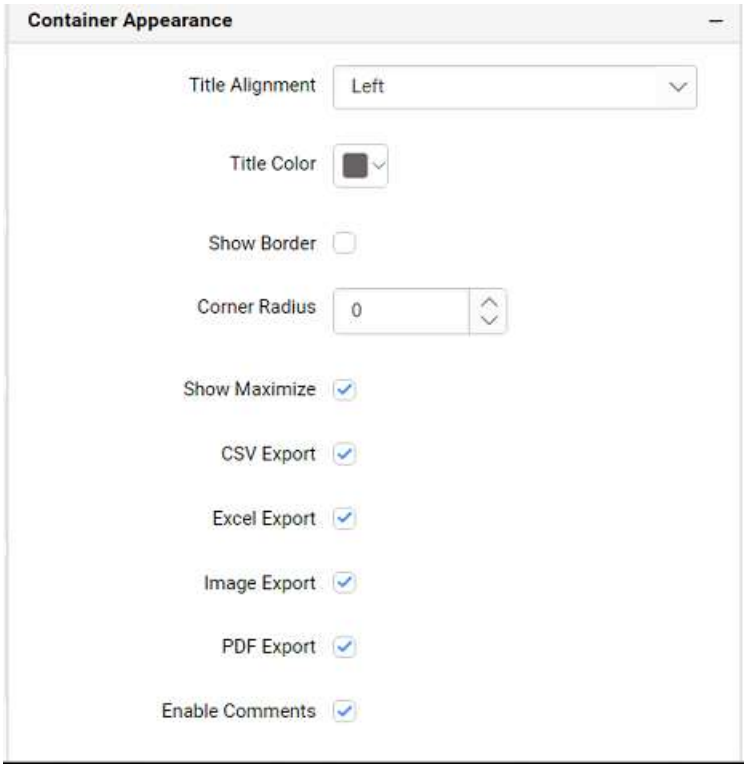
Act as Master Widget

This allows you to define this range slider widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this range slider widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comments

This allows you to enable/disable comment for dashboard widget. For more details refer here

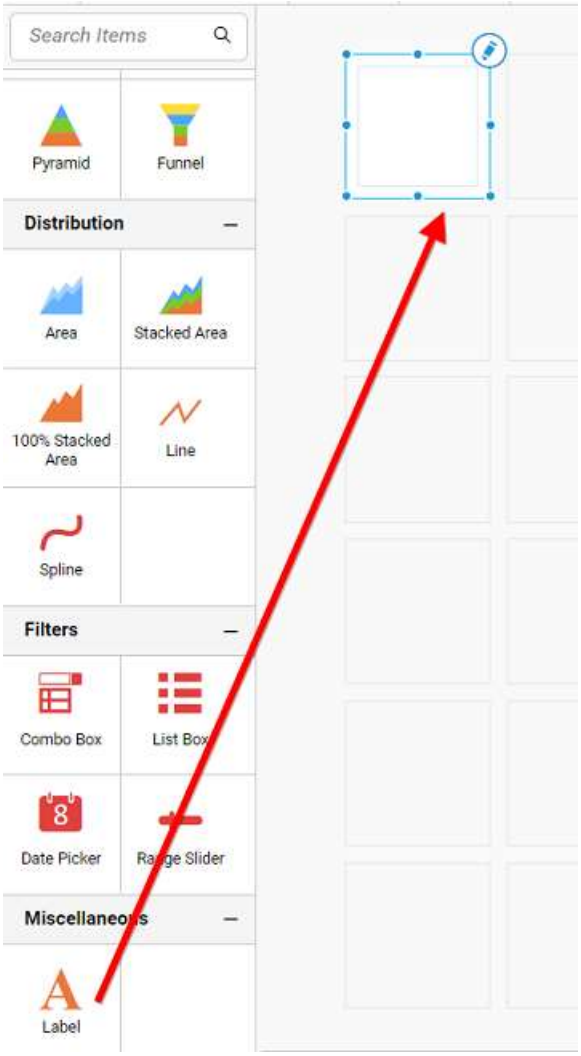
Label

Label allows you to display value set in rich text format through parameter placeholders.

How to add text and format label?

The following procedure illustrates data configuration of Label.

Drag and drop Label control icon from the tool box into design panel. You can find control in tool box by search.

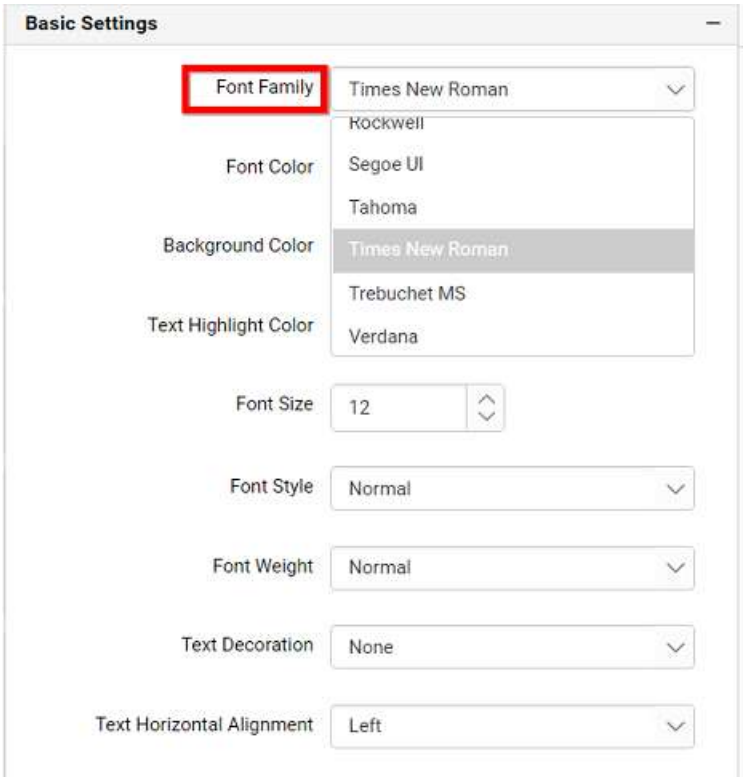


After control added in design panel, focus on label widget edit the label.

You can click widget setting icon to open the properties tab of label widget.

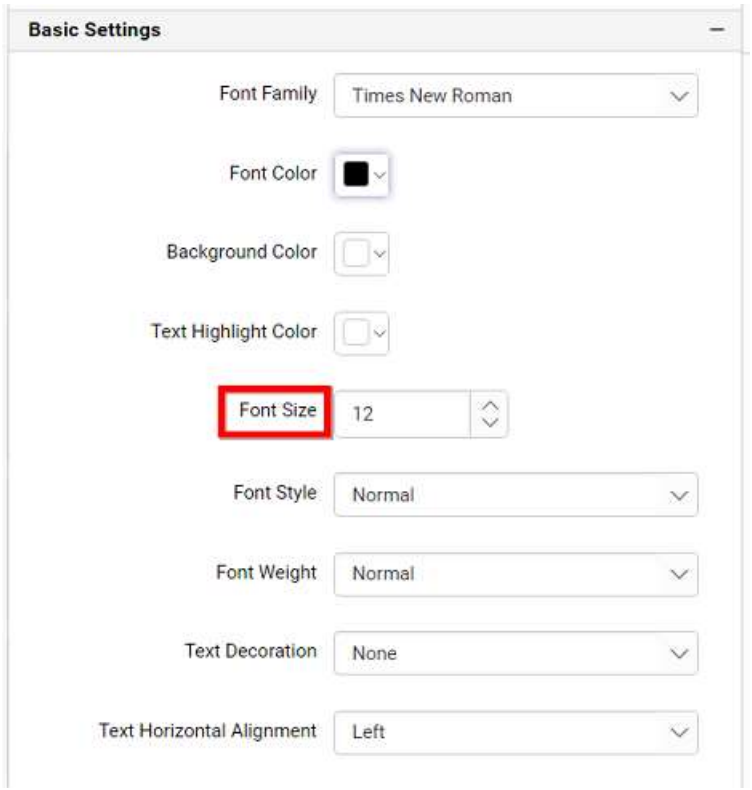
Font Family

You can change font family of the text in the label widget. By default, font family will be Roboto.



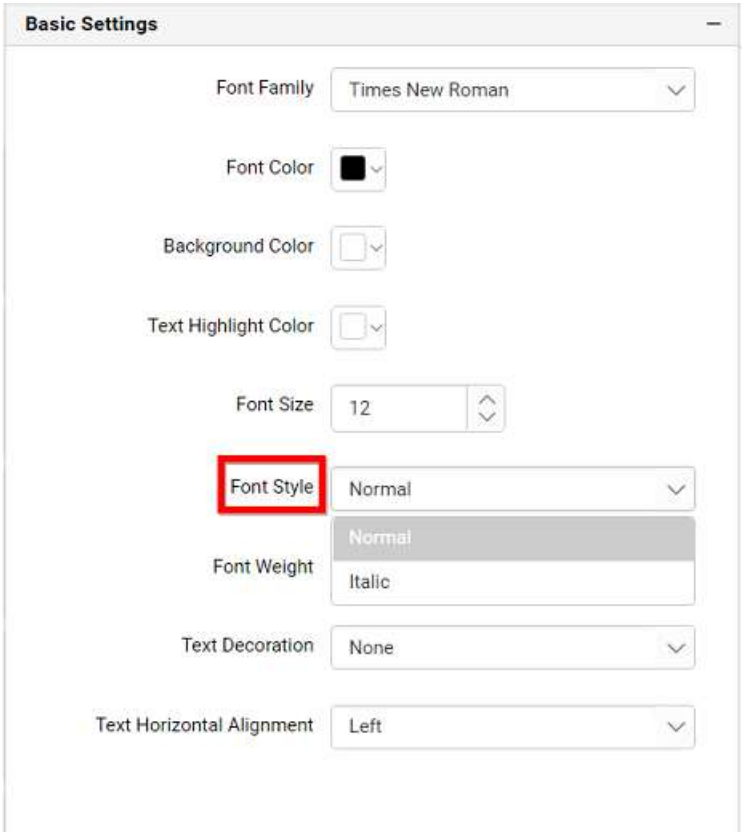
Font Size

You can change font size of the text in the label widget. By default, font size will be 12.



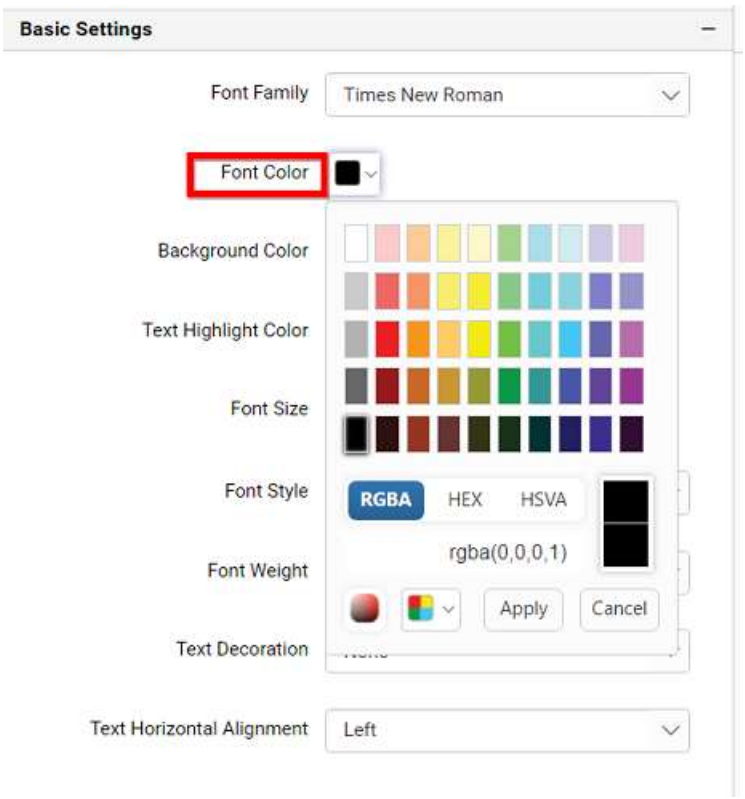
Font Style

You can able to change font style of the text. By default, font style will be normal.



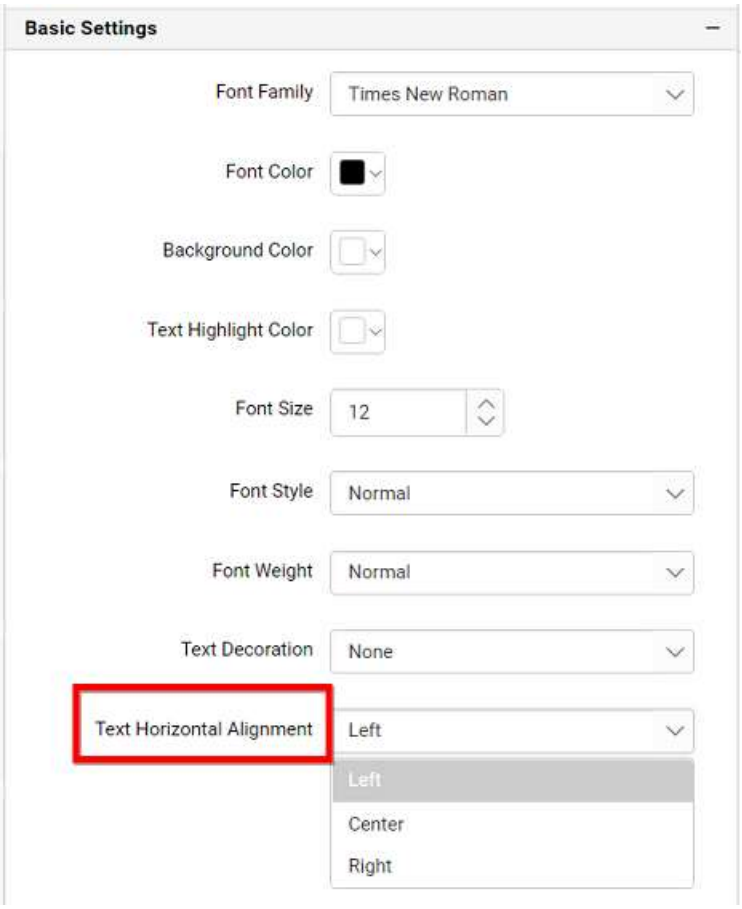
Font Color

You can customize the color of the text in the label widget.



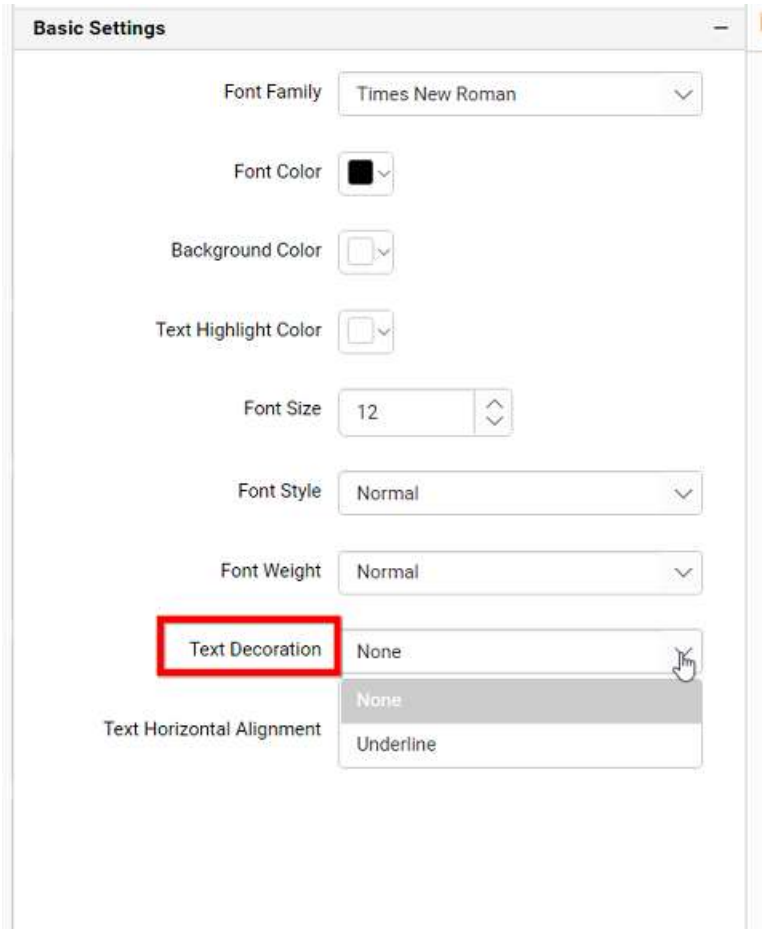
Text Horizontal Alignment

You have options(Left, Right, Center) to change the alignment of the text in the label widget. By default, text will be aligned in Left.



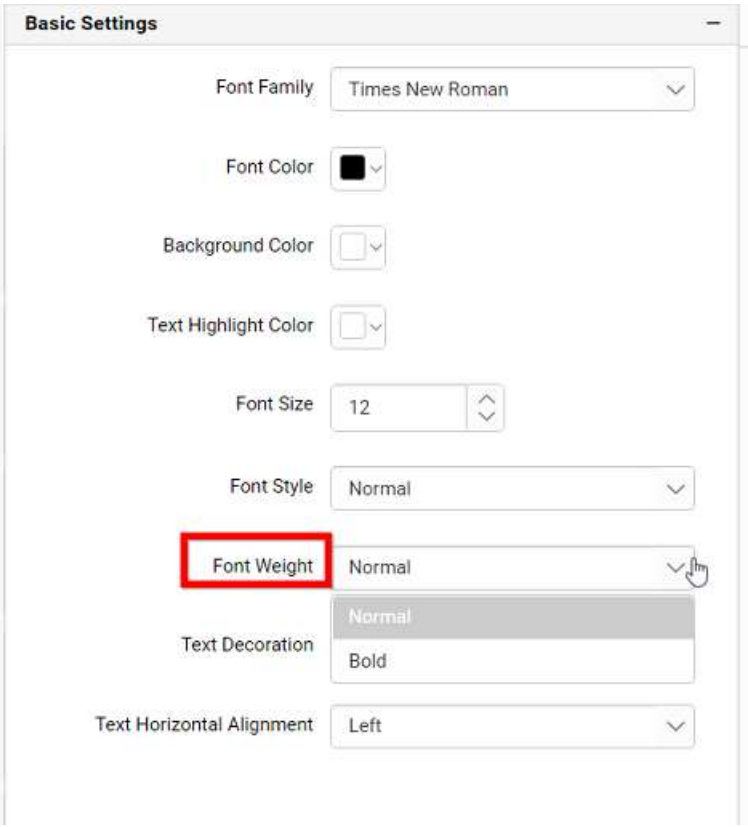
Text Decoration

You can able to decorate the text in the label widget using text decoration property. By default, text decoration will be none.



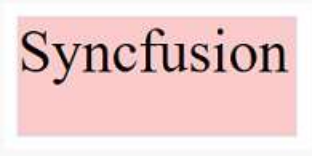
Font Weight

You can able to change font weight of the text. By default, font weight will be normal.



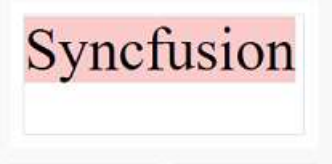
Background Color

You can customize the color of the background of the label widget.



Text Highlight Color

You can highlight the color of the text in the label widget.



Image

Image allows you to display a both static and dynamic image within defined mode (default, fill, uniform and uniform to fill).

You may add image of supported formats including, BMP, JPG, JPEG, GIF, EMF, JFIF, JPE, PNG, RLE, TIF, TIFF, WMF, DIB, and ICO from your local machine or column binding to the image widget.

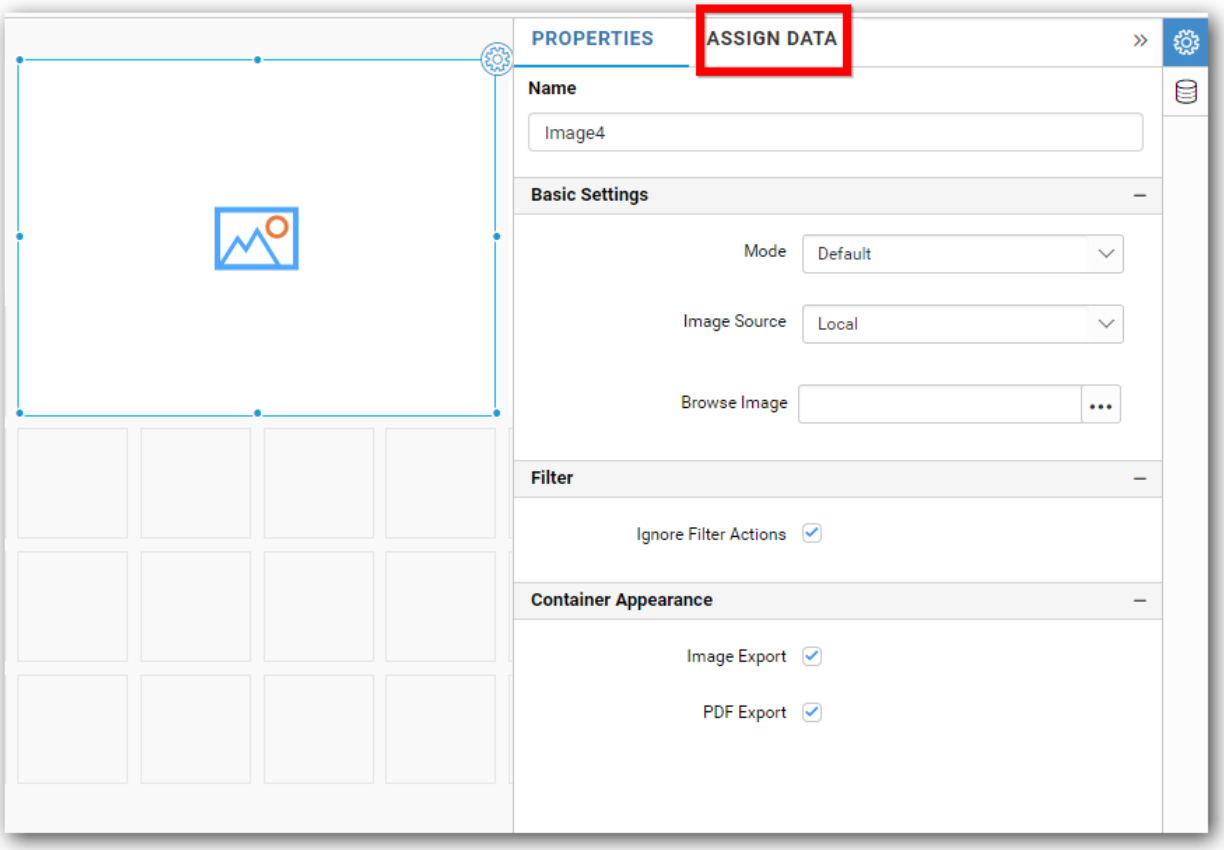
The following steps represents to add Image to dashboard.

Drag and drop the image widget into the Canvas.

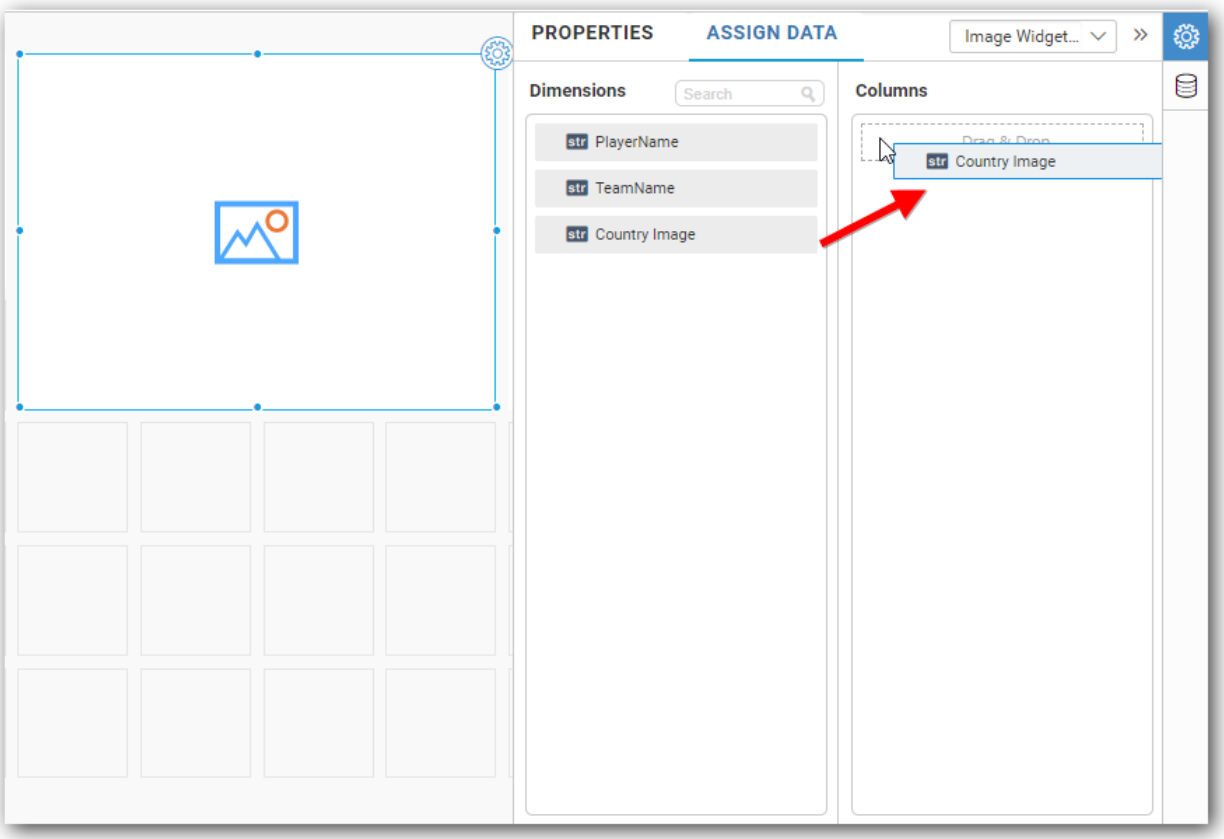
How to configure the data to Image widget?

You may browse the image or bind a datasource column that contains the image URL.

You may assign a data by clicking **Assign Data** button.



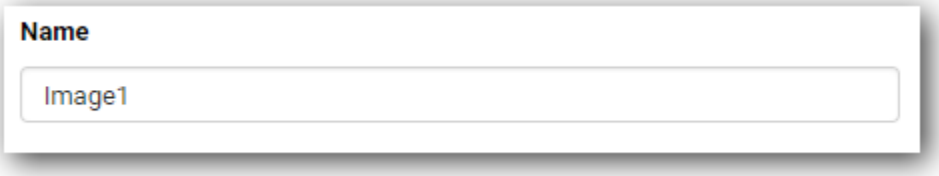
Drag and drop image column from dimension section to Column(s) section.



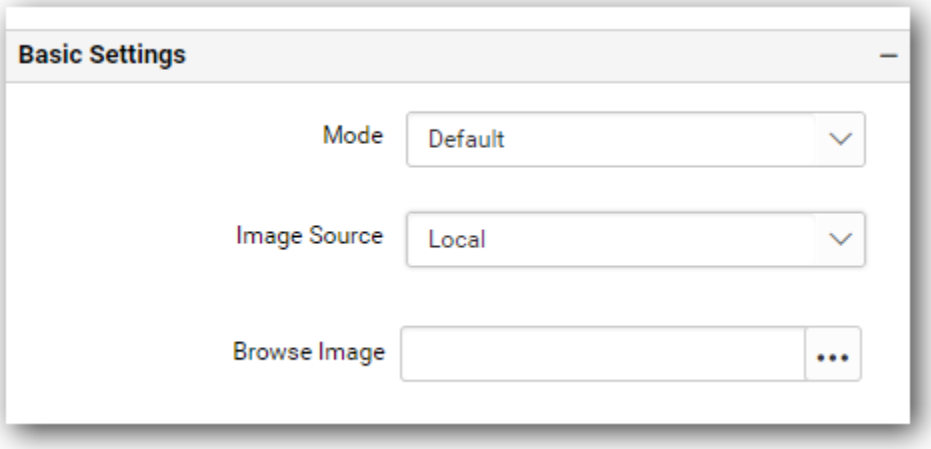
You can format the image for better illustration of the view that you require, through the settings available in Properties pane.

Name

This allows you to set title for this Image widget.



Basic Settings



Mode

You can customize the image showcase style through Mode setting in the Design Tools pane or properties pane.

Default

The image will be displayed in its original size.

Fill

The image will be filled in the available space.

Uniform to Fill

The image will be uniformly occupying the space but gets clipped, if it is larger than control

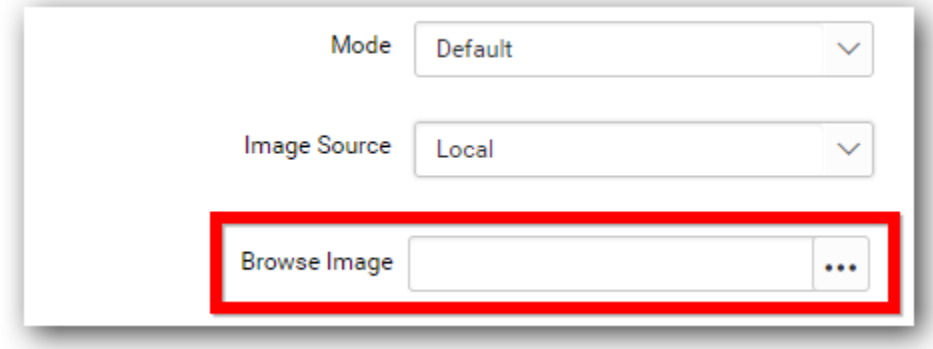
Uniform

The image sizes proportionally (without clipping) to best fit to the widget area.

Image Source

Local

You can browse the image from your local system.

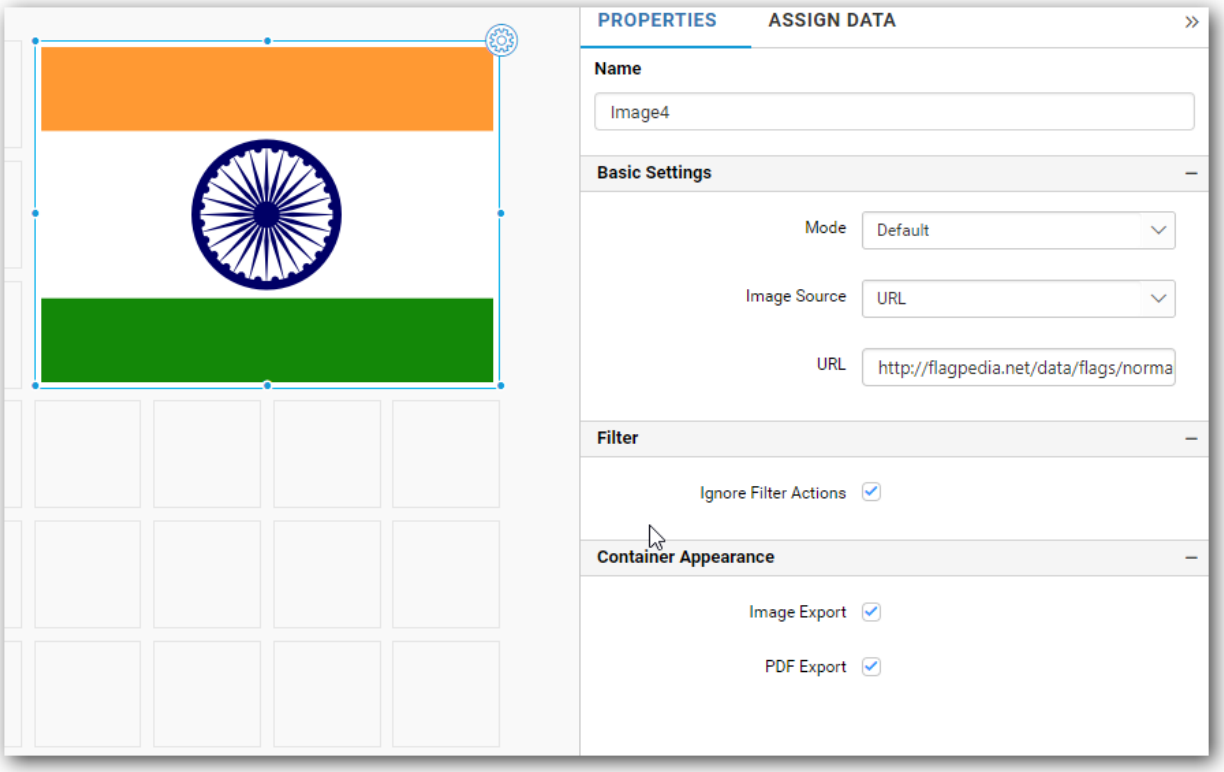


Note: Image that having special characters in the file name is not supported in Dashboard Application.

URL

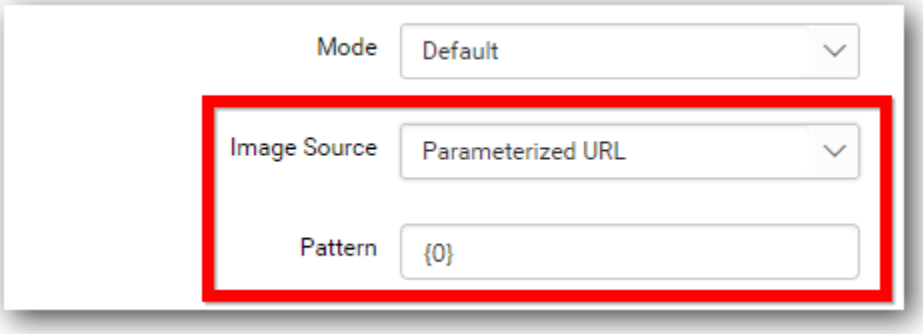
You can give the url of the image which must be a valid url.

Ex: http://flagpedia.net/data/flags/normal/in.png



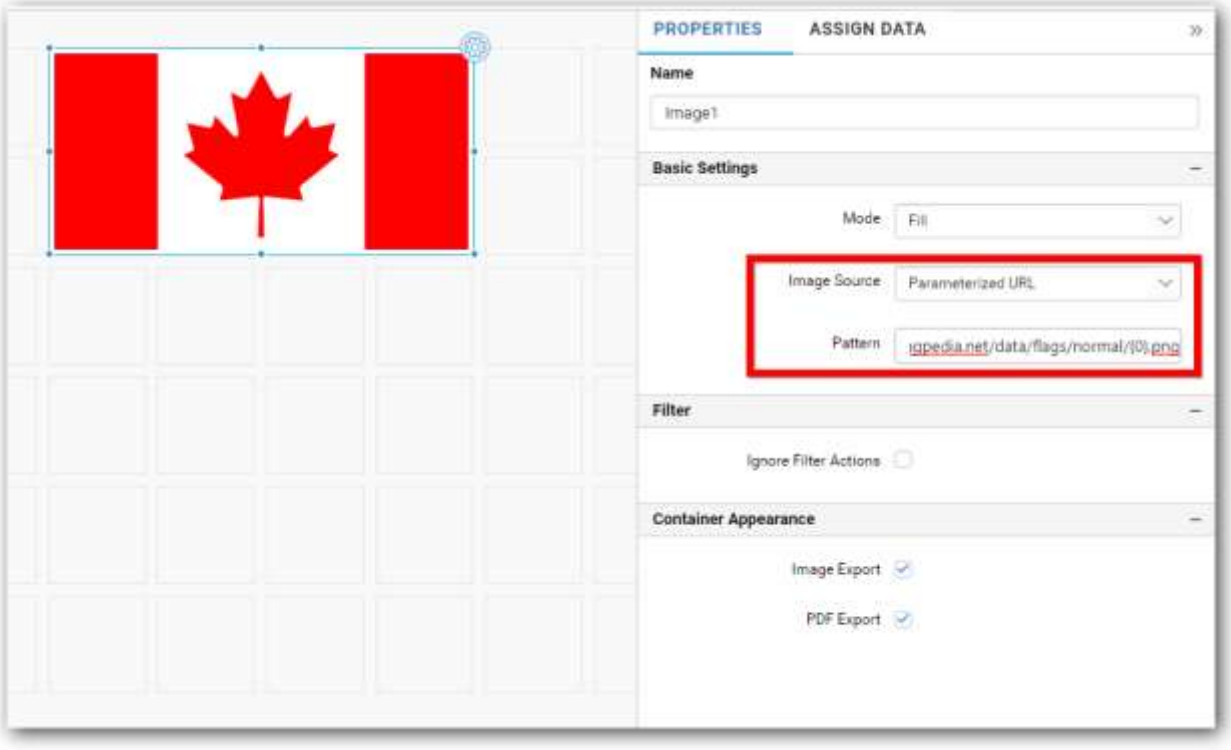
Parameterized URL

Parameterized URL images can be bind through this option and we can also able to add different columns value as a placeholder through the placeholder textbox.



Forming URI through placeholder

EX: http://flagpedia.net/data/flags/normal/{0}.png



Filter Settings

IgnoreFilterActions

You can ignore the filter actions by enabling IgnoreFilterActions property. Browse Image will not act as a slave widget.

Container Appearance

Image Export

You can export the image widget as image.

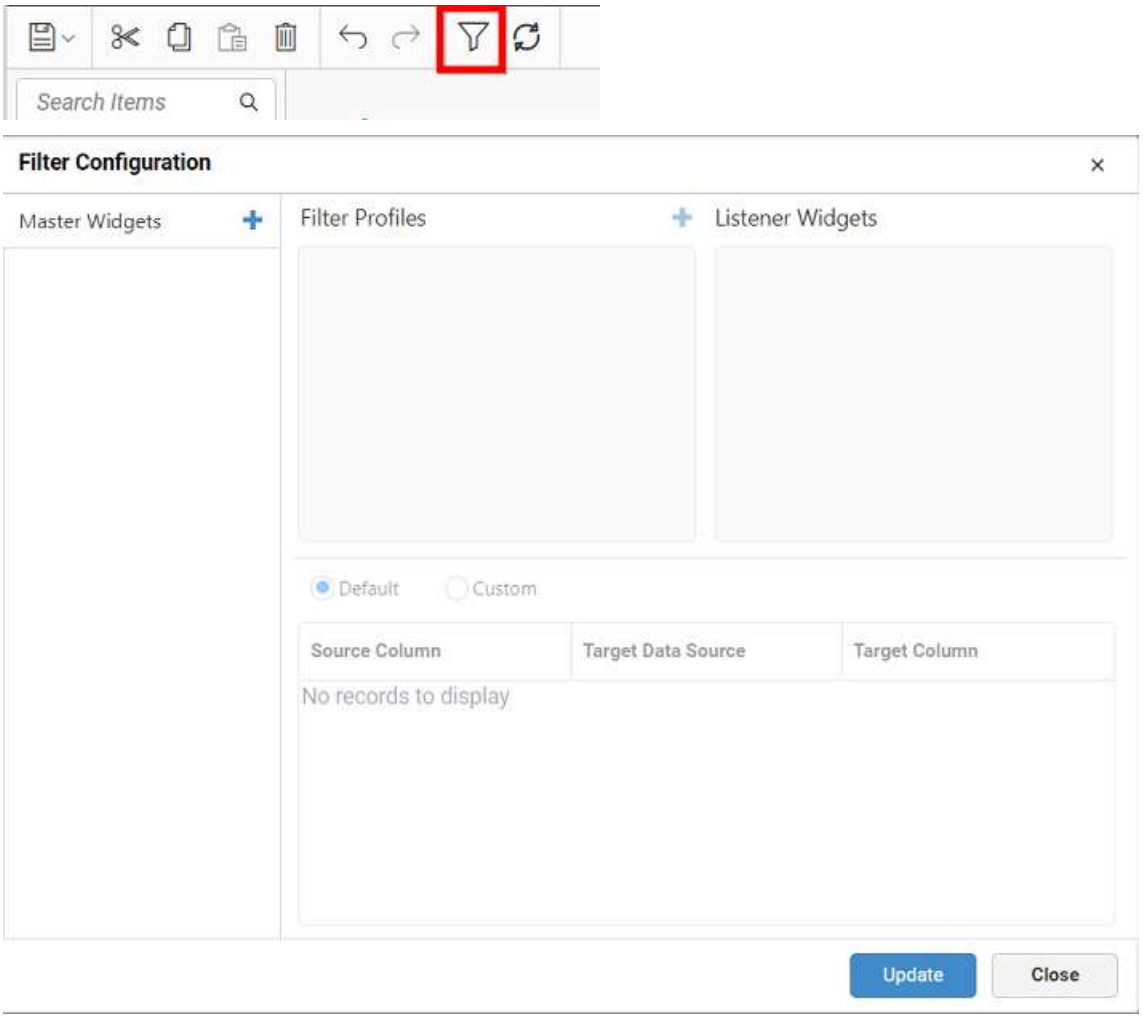
PDF Export

You can export the image widget as PDF.

Configuring Dashboard Filters

Dashboard filters allow you to control the interdependency of widgets in a dashboard with respect to dynamic user interactions.

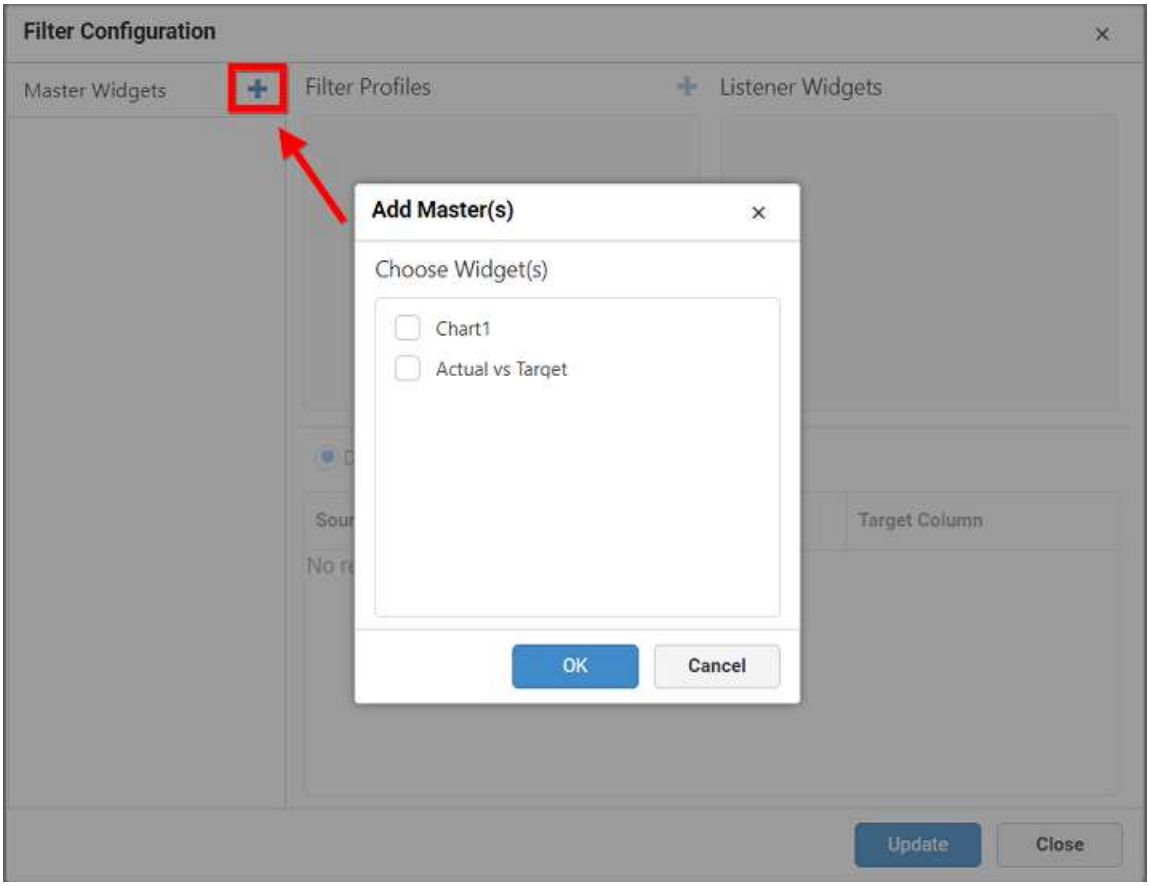
You can configure the dashboard filters through the Filter Configuration window that is launched by selecting the Filter menu in the tool bar.



Master widgets

In this window, the master widgets section holds the names of widgets whose Act as Master Widget property setting is enabled. If the widget is added under this section, it is subjected to have its filter effect on user interaction. The remaining widgets that are not under the master widget section are marked as listeners through the listener widgets section.

You can also add a widget into this section explicitly by clicking the highlighted icon.



Selecting a widget show its associated filter profiles and listener widgets in respective sections; they are customized.

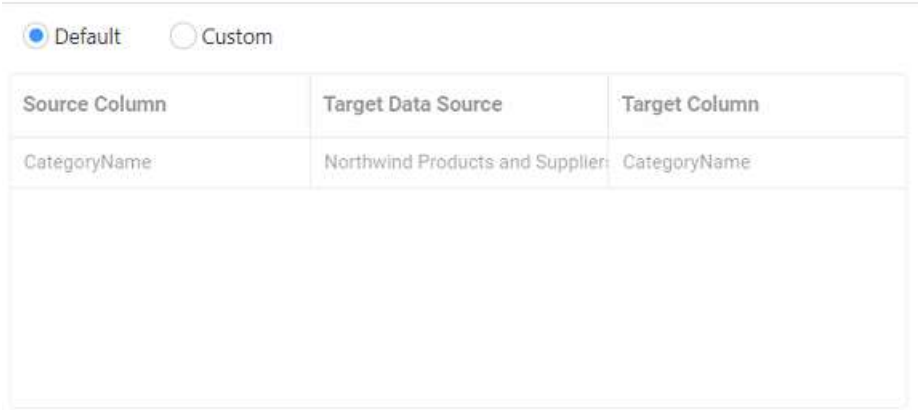
You can remove the selected widget from the master widgets section through the Remove in its header.

Note: Filter type widgets will get added automatically for user convenience, by default. You can remove it, if not required.

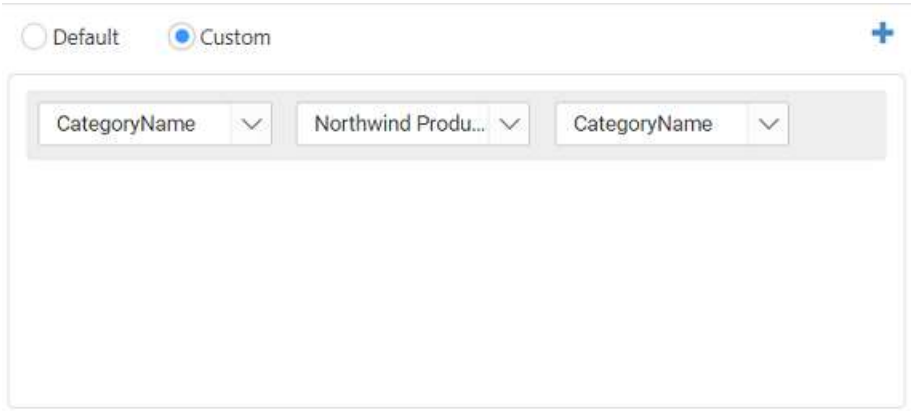
Filter profiles

The filter profiles section holds a default profile generated automatically for the widget that is added in the master widgets section. This profile holds the detail about the filter criteria and listener widgets to be affected based on that criteria.

Filter criteria can be set through the bottom pane configuration.

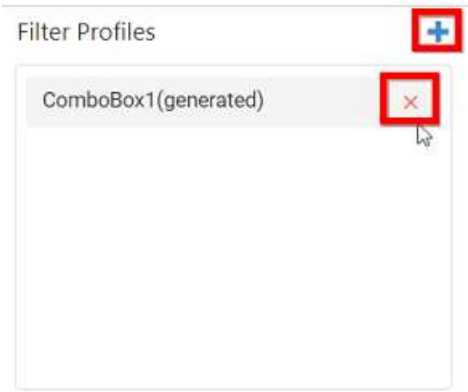


This pane holds the detail about mapping of a column in the current data source with the column in target data source, which is same by default, based on which the user interaction filtering works. You can also customize this default filter criteria by switching to Custom option, such as add, edit, and remove.



You can modify the default profile setting or remove the same or add a new profile through respective options in its header.

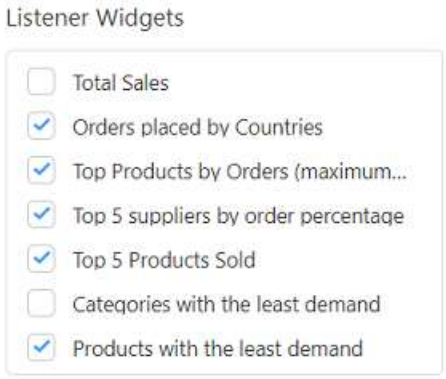
You can rename the profile by double-clicking it.



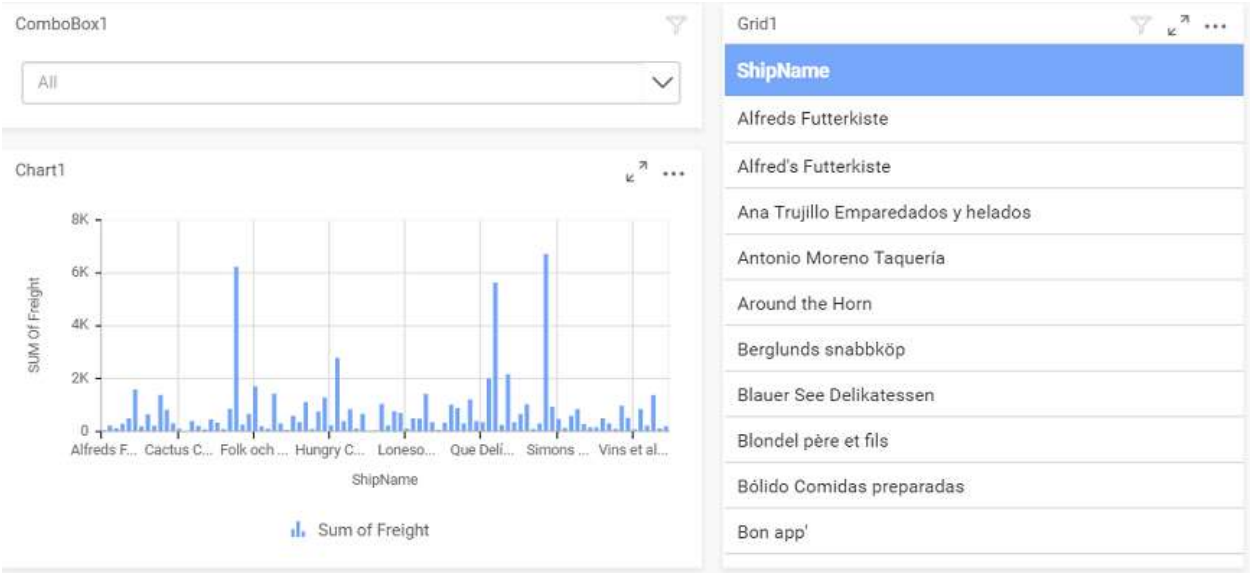
Note: If you define more than one filter profile for a master widget, it gets the interaction effect, i.e., all those filter profiles that match filter criteria will be filtered.

Listener widgets

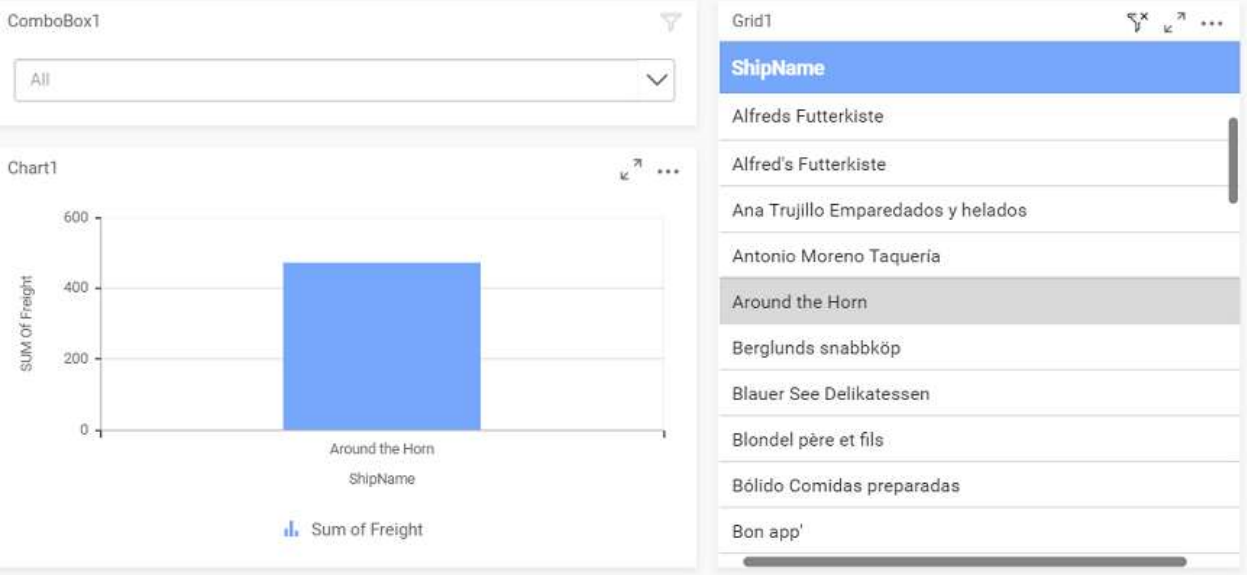
Listener Widgets section holds the name list of widgets other than the master widget. Select the check box that besides to the respective widget name to map it to the respective master widget under the specified filter profile. To respond to user interaction in master widget, deselect the one that you do not want to respond to.



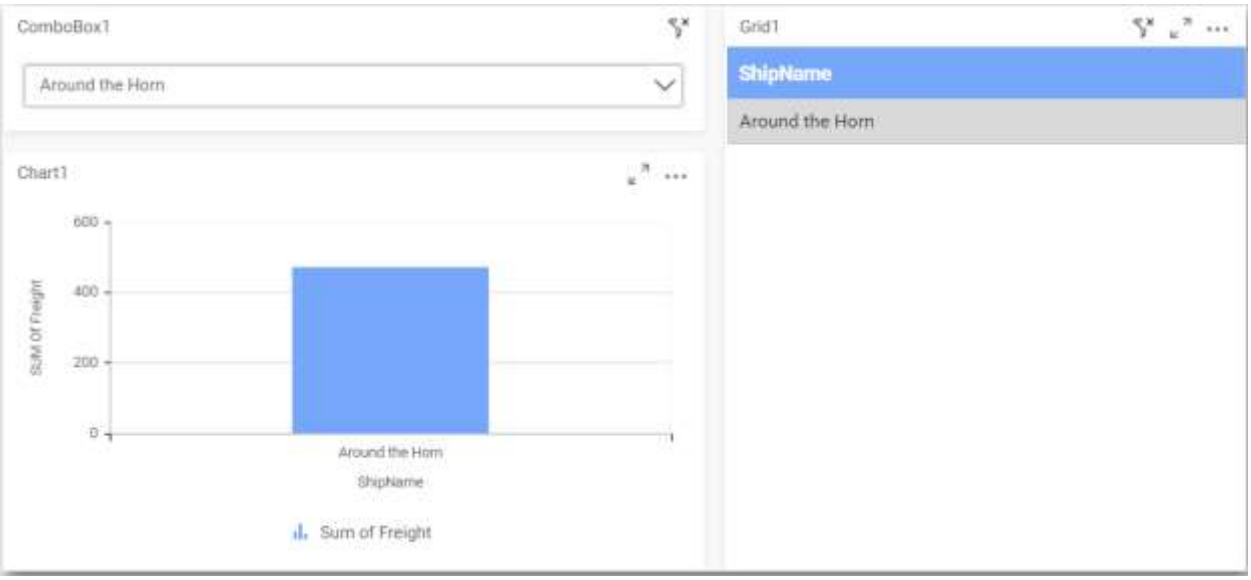
For example, Consider Chart*1 widget is marked as a listener widget to the Grid*1 and ComboBox*1 widgets, Grid*1 widget is marked as listener widget to the ComboBox_1 widget.



While applying the filter in the grid widget, the chart widget will be filtered. Now, the chart widget contains the information about the selected ship name in the grid.



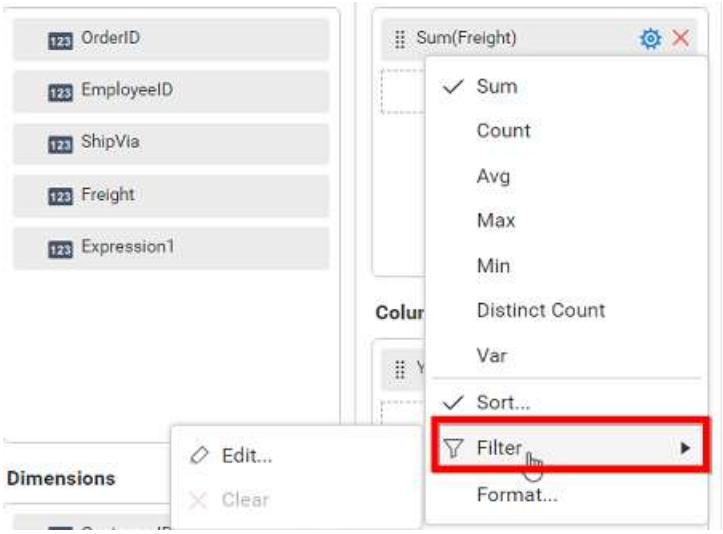
On selecting the data in the combo box, the chart widget and grid widget show the particular detail.



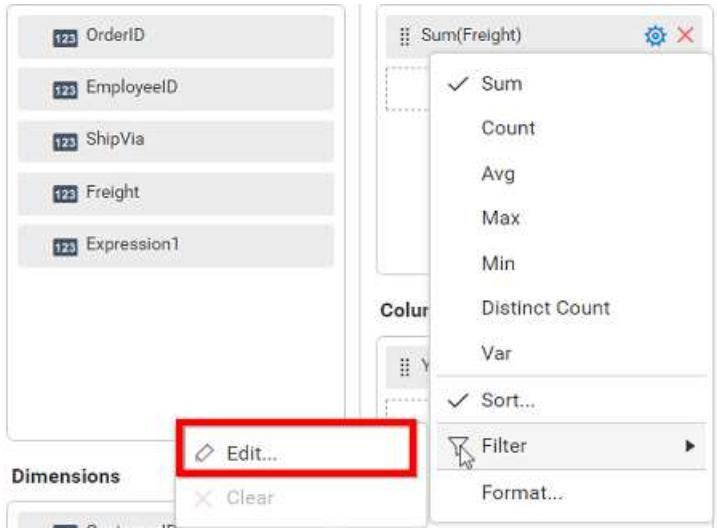
Configuring Widget Filters

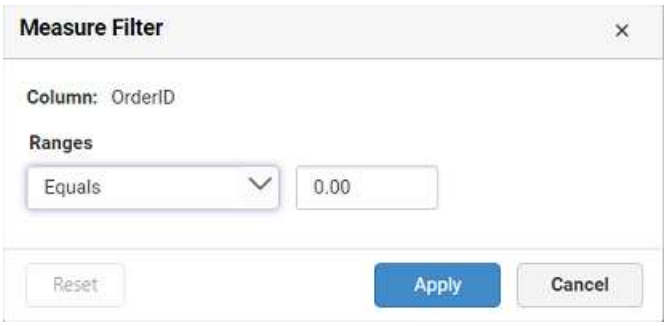
Configuring filter for measure column

The filter for measure column can be configured by opening the Measure Filter dialog from the Filter option in the Settings drop-down menu.



By clicking the Edit... menu item, the measure filter dialog will open as follows.

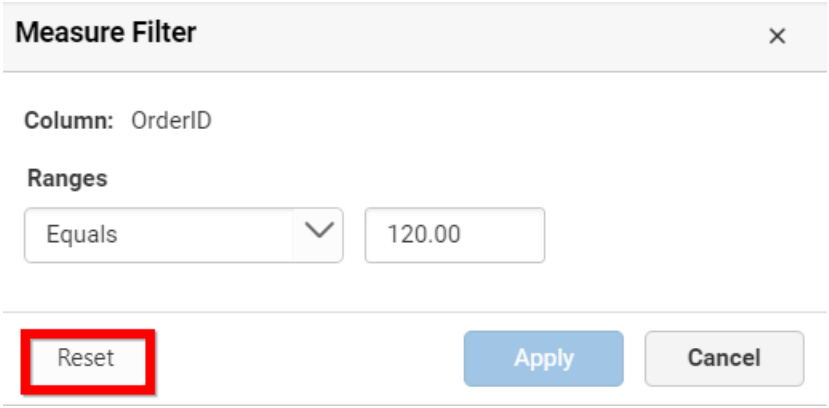




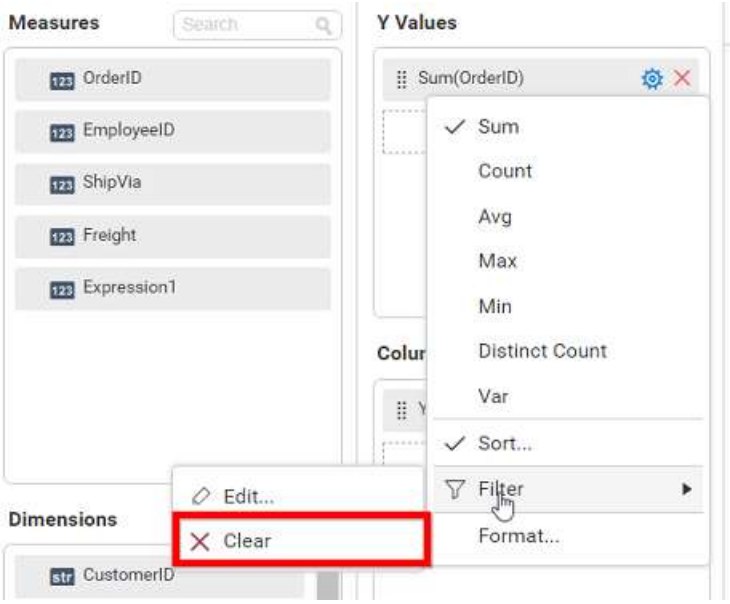
Configure the compare operator and the value to be compared against the selected column values. Click Apply to apply the filter settings to the widget. Now, the applied settings are saved, and these settings will be retained on reopening this dialog.

Click Reset to reset the changes made in the dialog. This will also reset the filters applied before to that column.

Note: Reset will be in enabled state only when the filter is applied already. Apply will be in enabled state only when there are pending changes in the dialog to save.

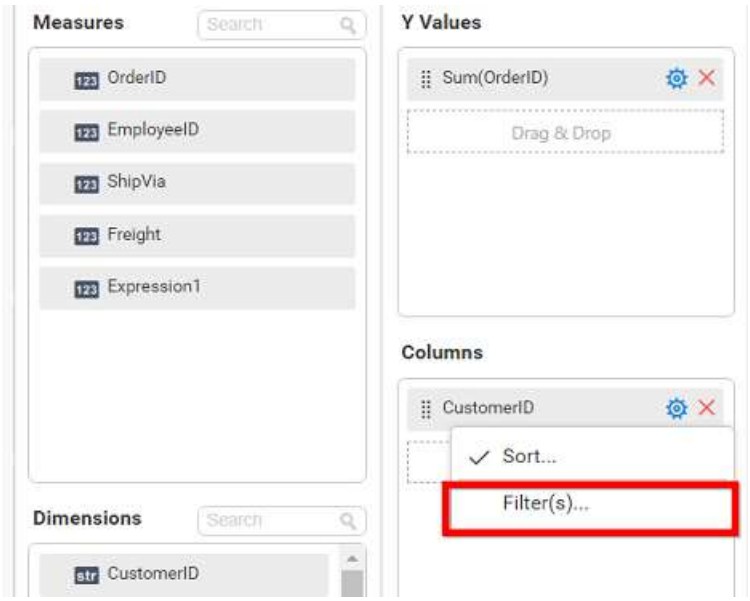


To clear the filter applied to the measure column, click Clear in the settings drop-down menu. This menu item will be in enabled state only when the filter is configured already to that column.

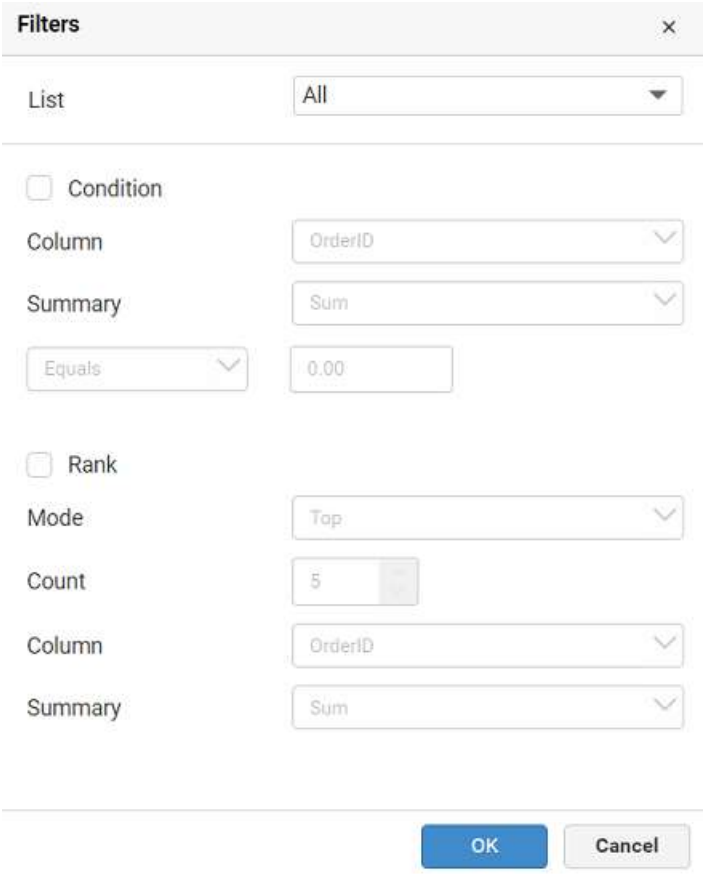


Configuring filter for dimension column

The filter for dimension column can be configured by opening the Filters dialog from the Filters... option in the Settings drop-down menu.



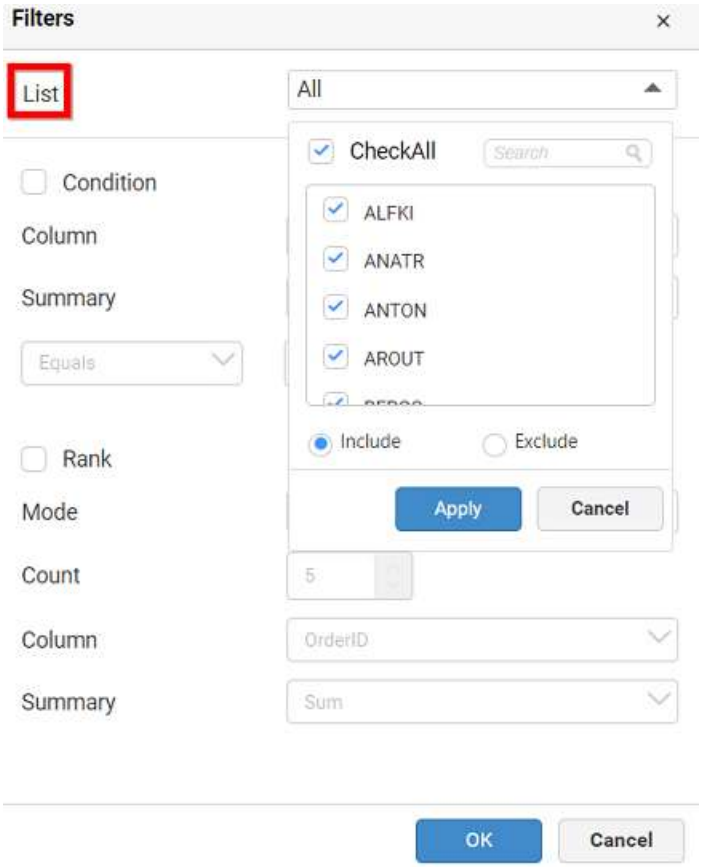
The filters dialog will open as follows.



This dialog consists of three different filters that can be applied individually or in combined manner.

Item-based filtering

Through this filtering, you can filter the specific items from consideration.



Click the list at top-right corner to drop-down the list holding the individual values of that column. You can check or uncheck each value or as a whole using the CheckAll.

Include and Exclude options allow you to choose whether to consider checked items for inclusion or exclusion in filter respectively.

The Search text box helps you to filter the items from the large list and search for specific one.

Click Apply to save the changes made in the filter drop-down list.

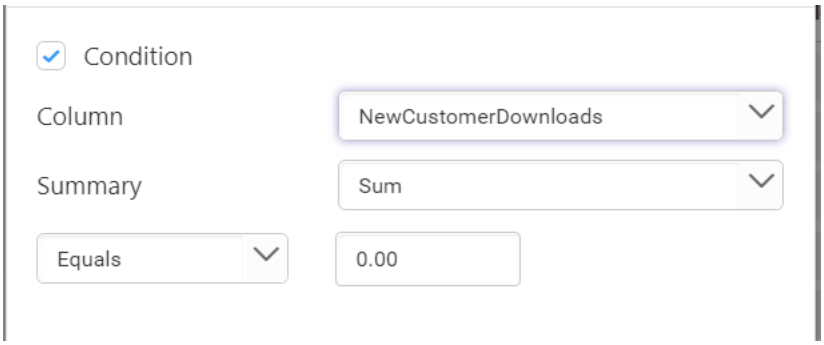
Click Cancel to cancel the changes made in the filter drop-down list, if required.

Click OK in the filters dialog to save the changes made with respect to item-based filtering.

Click Cancel to ignore the changes made in the filters dialog, if required.

Condition-based filtering

Through this filtering, you can impose a condition based on which the filter need to be applied. This filtering option is disabled, by default. You can enable it by clicking the Condition check box.



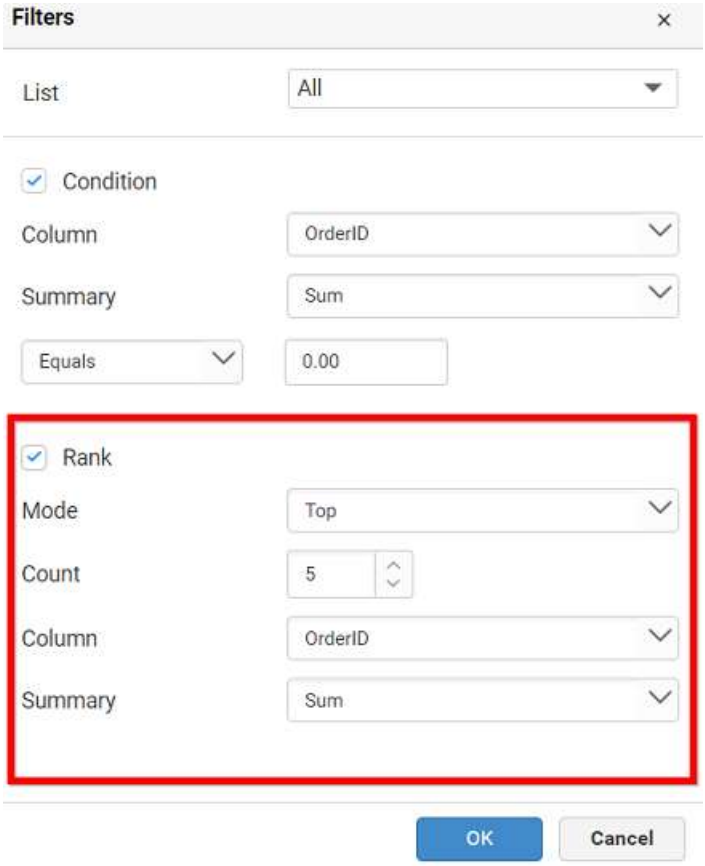
Set the column near the Column label, by which the filter criteria need to be defined. Set the summary type near Summary label, based on which aggregation need to be applied over the selected column. Set the compare operator and the value to compare against the column values.

Click OK in the filters dialog to save the changes made with respect to condition-based filtering.

Click Cancel in the Filters dialog to ignore the changes, if required.

Rank-based Filtering

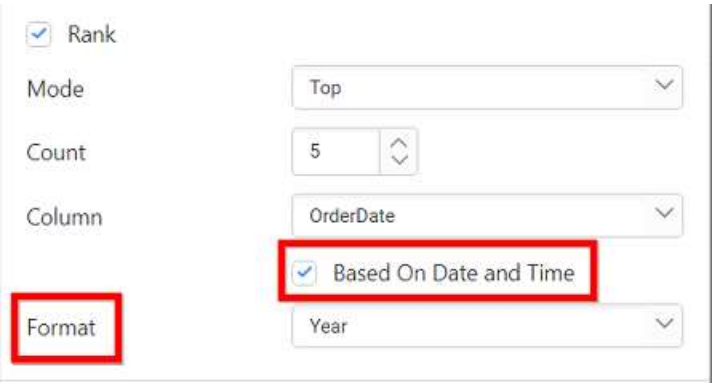
You can filter the n items by setting the mode, count, column, and summary fields through the rank-based filtering. You can enable it by clicking the Rank check box.



Set the top or bottom mode in the Mode field, and set the number of records to filter in the Count field. Set the column name in the Column field based on which the filter need to be applied.

Set the summary type in the Summary field based on which aggregation need to be handled whose output should have been compared with corresponding value in widget bound data.

For date Column field, you have additional option to format the Column field by enabling Based On Date and Time. Set the format type in the Format field need to be handled whose output should have been compared with corresponding value in widget bound data.



Click OK in the filters dialog to save the changes made with respect to rank-based filtering.

Click Cancel in the filters dialog to ignore the changes, if required.

Note: When all these three filters are configured and applied, the records that satisfy the criteria of three filters will be considered by the widget.

Configuring Label Parameters

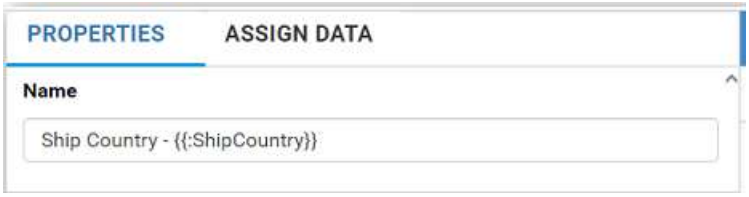
You can configure the label parameters by using the field name in the Name of widget. Use the following format to configure the label parameter.

Syntax: {{“{{“}}:Column_Name{{}}}} **when single data source is present**

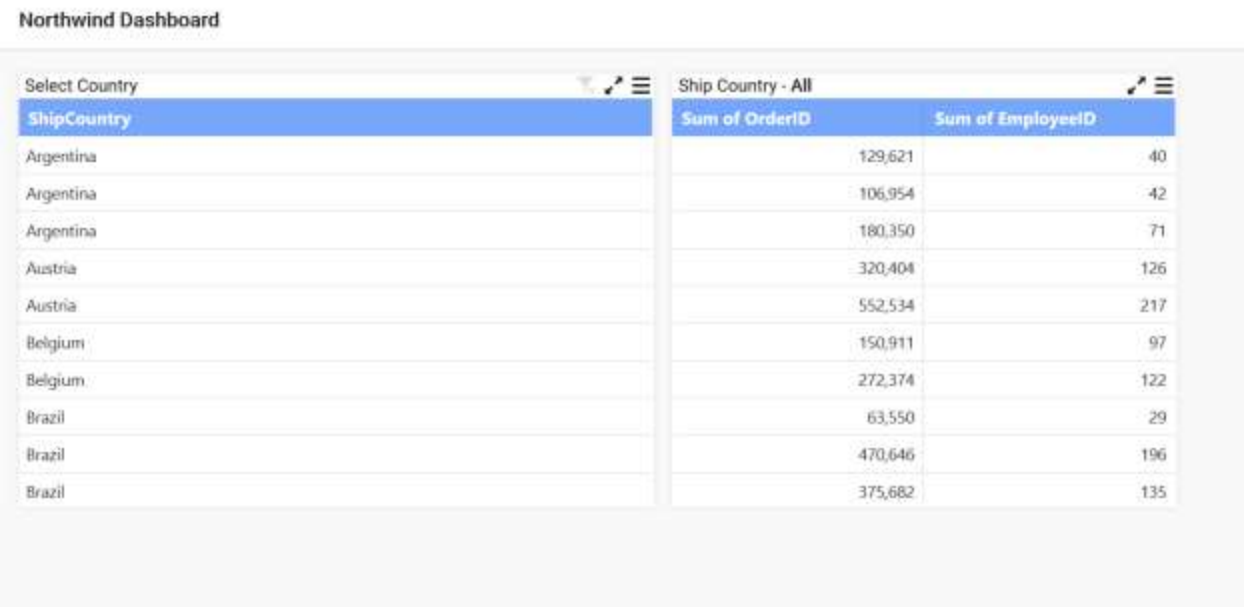
Or

Syntax: {{“{{“}}:DataSource*Name.Column*Name{{}}}} **when more than one data source is present**.

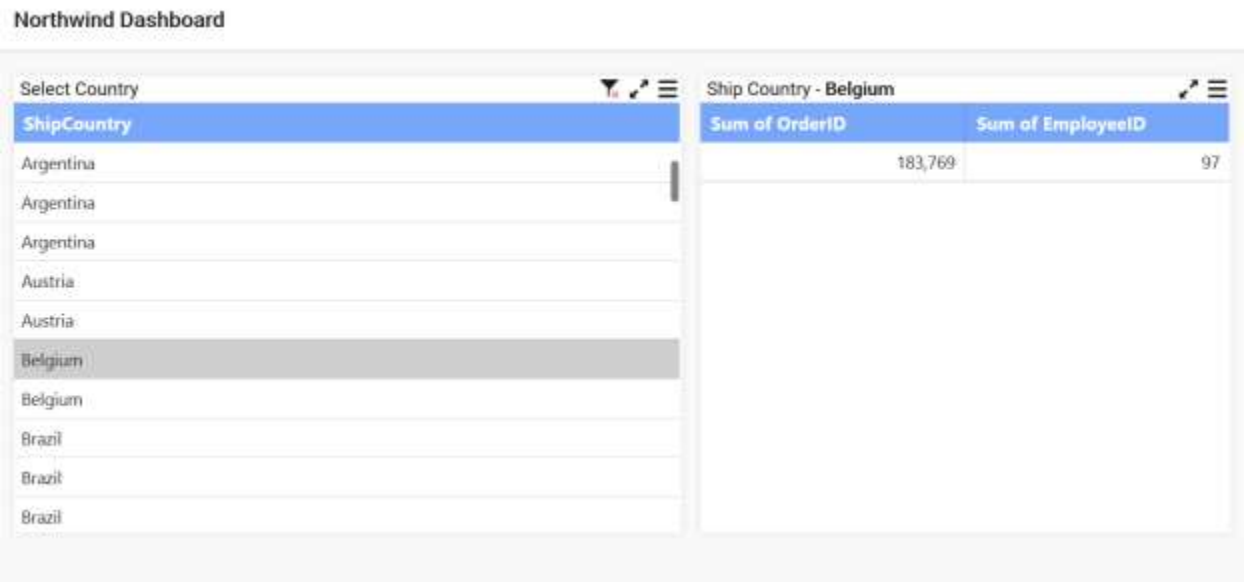
For example, the name text of the Grid widget as: Ship Country - {{"{{"}}:ShipCountry{{}}}}.



Now, the dashboard will show the label parameter for all countries like Ship Country - All

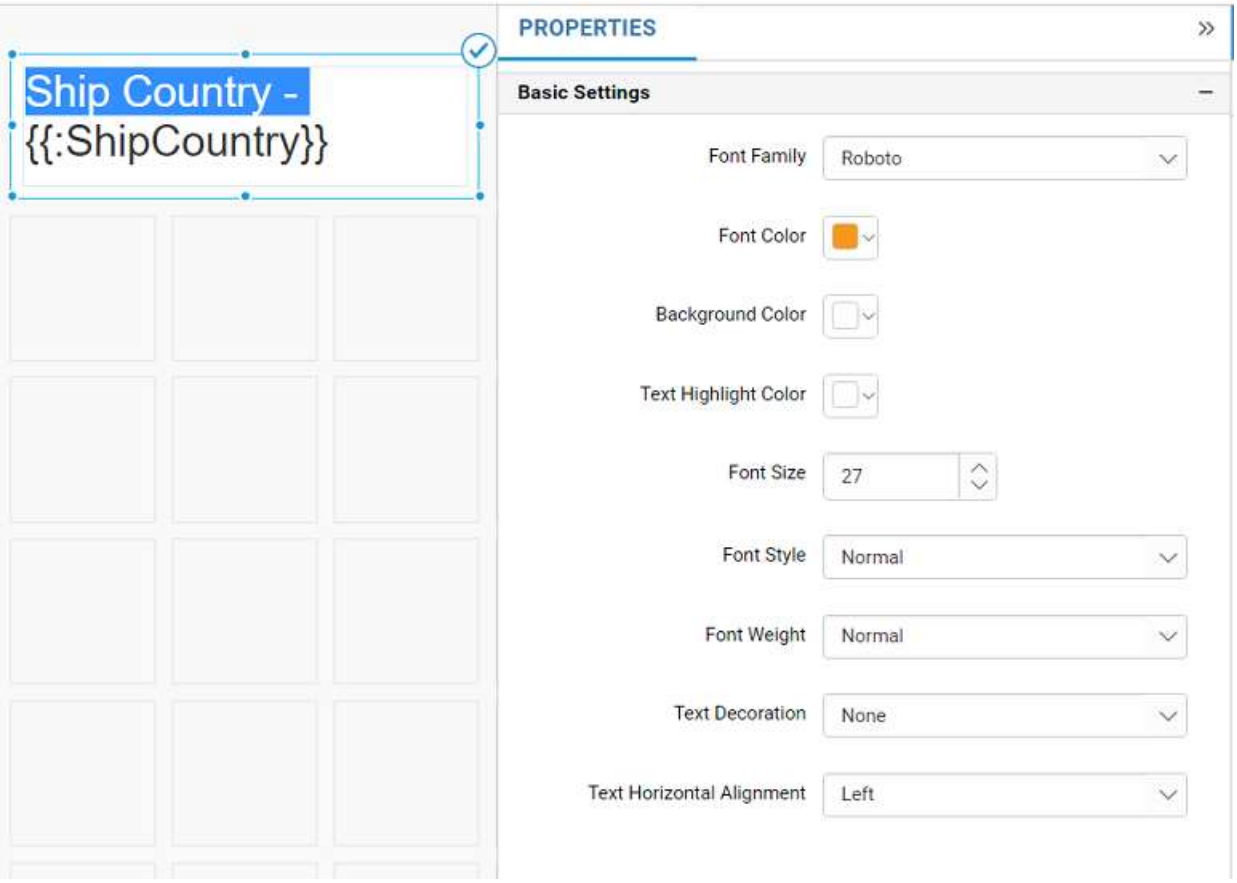


You can select the required country to display by selecting the country name in the grid widget. Based on the selected country, the values will be displayed on the widget.

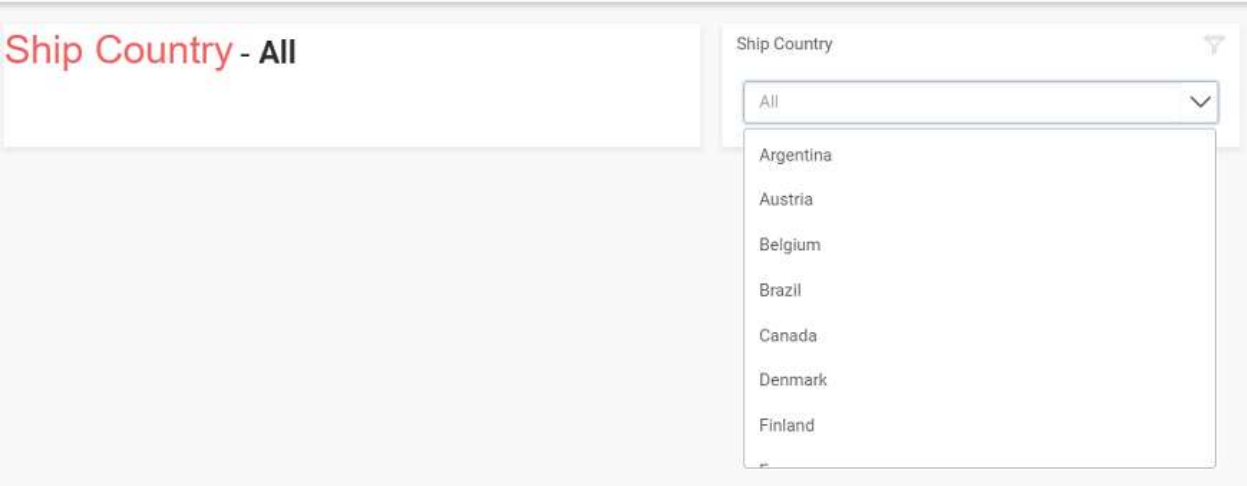


You can also use label widget to configure the label parameters.

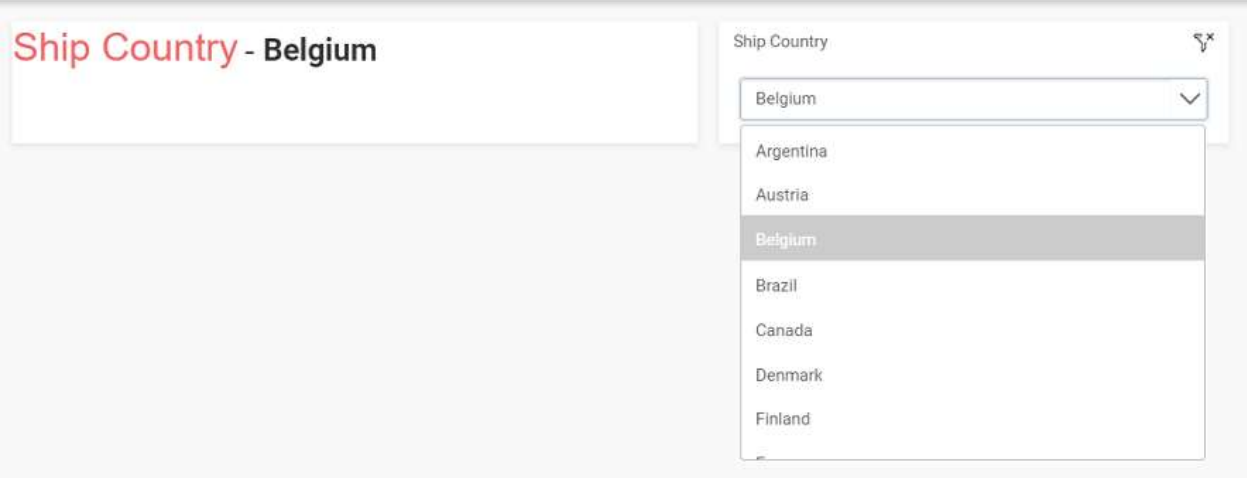
Drag and drop the label widget into the design pane and click the edit label to add the label parameters.



While previewing the dashboard, initially the parameter of the ship country will be shown as All in the label widget.

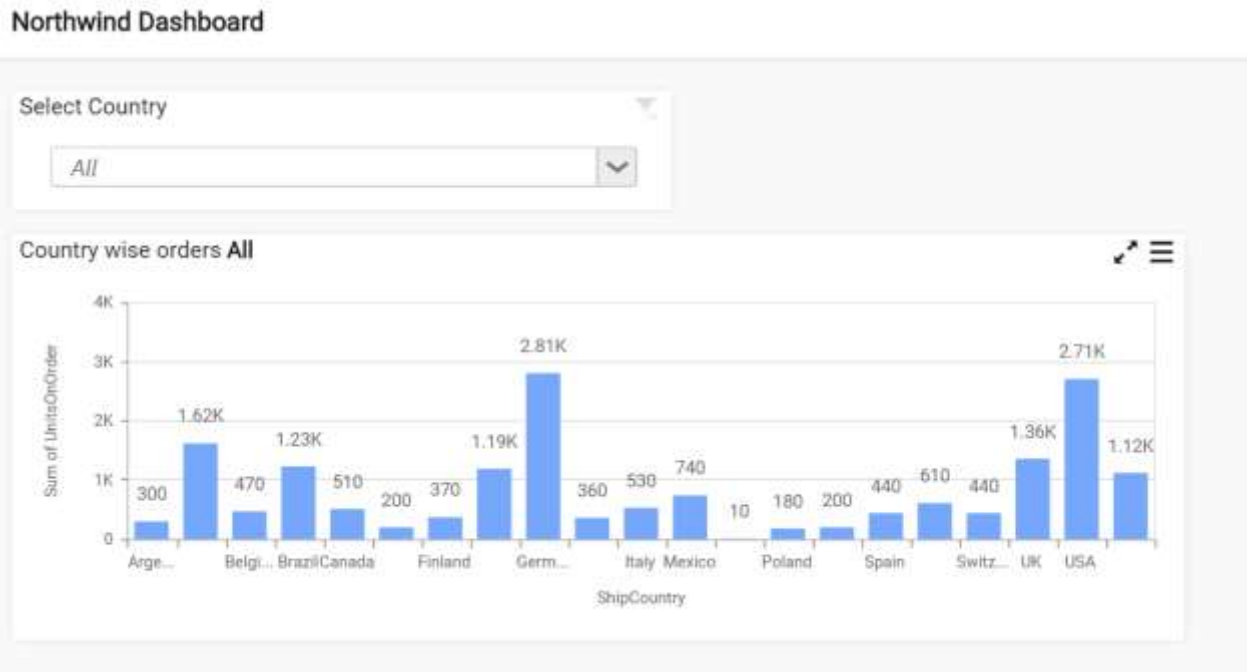


While selecting the particular country, it will show that country name in the label widget.

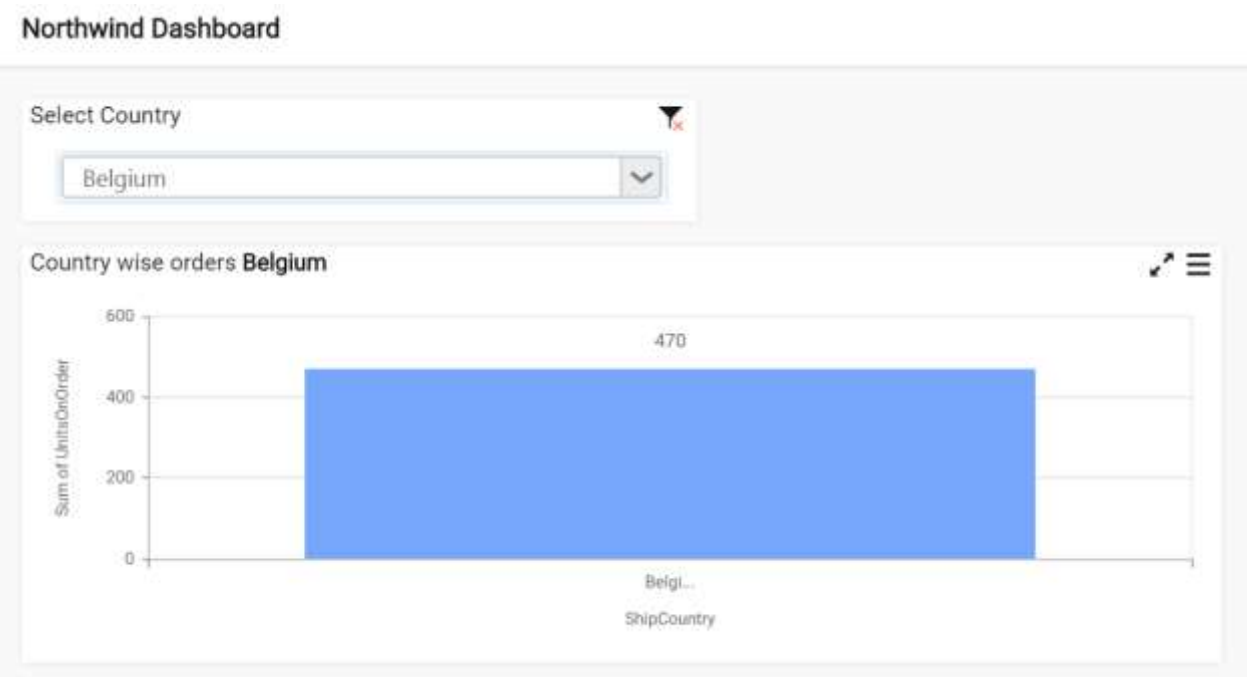


You can add the data source before the parameters in the Name of the properties tab, when more than one data source is present in the dashboard.

While previewing the dashboard, initially the values of parameters will be shown as All.



While selecting the particular country, you can show that country name in the widget.



Functions supported in label parameters

You can configure the label parameters using the column name with functions such as Sum, Count, Average, Min, Max, Var, and Distinct Count for numeric values.

For Date Time type, the supported functions are Date, Year, Month, Quarter, Day, Day Month Year, Minutes, Second, Month Year, Date Hour, Day of Week, and Week of Year.

For Text type, the supported functions are Count and Distinct Count.

Use the following format, to configure the label parameter.

Syntax: {{“{{“}}:function(Column_Name){{}}}} **when single data source is present**

Or

Syntax: {{“{{“}}:function(DataSource*Name.Column*Name){{}}}} **when more than one data source is present**.

Function Name	Supported Type(s)	Description	Example(s)
Sum	Number	This function returns the summation of the given column in number format.	Total Quantity - {{"{{"}}:Quantity{{}}}} or Total Quantity - {{"{{"}}:sum(Quantity){{}}}}
Average	Number	This function returns the average of the given column in number format.	Average freight amount is {{"{{"}}:avg(Freight){{}}}} or {{"{{"}}:average(Freight){{}}}}

Count	Number, Text, Date Time	This function returns the count of the given column in numeric format.	Number of records : {{"{{"}}:count(Quantity){{}}}} or Number of records : {{"{{"}}:count(ShipCountry){{}}}} or Total Transactions : {{"{{"}}:count(InvoiceDate){{}}}}
DistinctCount	Number, Text, Date Time	This function returns the count of distinct values in the given column in number format.	Number of unique records are {{"{{"}}:dcount(OrderID){{}}}} or {{"{{"}}:distinctcount(OrderID){{}}}} Countries count is {{"{{"}}:dcount(ShipCountry){{}}}} or {{"{{"}}:distinctcount(ShipCountry){{}}}}Total unique transaction count is, {{"{{"}}:dcount(InvoiceDate){{}}}}
Minimum	Number	This function returns the minimum value in the given column in number format.	Minimum value of Quantity: {{"{{"}}:min(Quantity){{}}}}
Maximum	Number	This function returns the highest value of the given column in number format.	Maximum value of Quantity: {{"{{"}}:max(Quantity){{}}}}
Date	Date Time	This function returns the date value as string formatted based on the current system culture.	Sales done on {{"{{"}}:date(ShippedDate){{}}}}
DayMonthYear	Date Time	This function returns the date value as string formatted in DD/MM/YYYY format.	Sales done on - {{"{{"}}:daymonthyear(ShippedDate){{}}}}
MonthDayYear	Date Time	This function returns the date value as string formatted in MM/DD/YYYY format.	Sales done on - {{"{{"}}:monthdayyear(ShippedDate){{}}}}
Year	Date Time	This function returns the year in number format.	Revenue for the Year - {{"{{"}}:year(ShippedDate){{}}}}
Month	Date Time	This function returns the month name as MMM format on the selected row.	Weather in NYC - {{"{{"}}:monthname(ShippedDate){{}}}}
Quarter	Date Time	This function returns the â€˜Quarter 1/2/3/4â€™	Total sales for the Quarter - {{"{{"}}:quarter(ShippedDate){{}}}}

		based on calendar year on the selected row.	
Day	Date Time	This function returns the day value (1-28/29/30/31) on the selected row.	Store rate for the day - {{"{{"}}:day(ShippedDate){{}}}}
Minutes	Date Time	This function returns the minute value (00-59) on the selected row.	Stock rate in - {{"{{"}}:minutes(ShippedDate){{}}}} minutes
QuarterYear	Date Time	This function returns the â€˜Quarter 1/2/3/4â€™ and the year based on calendar year on the selected row.	Total sales in - {{"{{"}}:quarteryear(ShippedDate){{}}}}
MonthYear	Date Time	This function returns the month and year as MMM YYYY format for the selected record.	Weather in NYC on {{"{{"}}:monthyear(ShippedDate){{}}}}
DateHour	Date Time	This function returns the date and hour values for the selected record.	Total sales in - {{"{{"}}:datehour(ShippedDate){{}}}}
DayOfWeek	Date Time	This function returns the day value of the week (Sunday to Saturday) for the selected record.	Weather in NYC on {{"{{"}}:dayofweek(ShippedDate){{}}}}
WeekOfYear	Date Time	This function returns the week value of respective year as number in the selected record.	Week of the year is, {{"{{"}}:weekofyear(ShippedDate){{}}}}

Note: Function name of label parameters are case insensitive.

Aggregating Value Columns Based on Type

Each value column configured to widget can be aggregated individually based on the type you define. Following table illustrates the aggregation types and their use.

Aggregation type	Result
Sum	Calculates the sum of values of all members.
Count	Returns the count of all members.
Average	Calculates the average of values of all members.

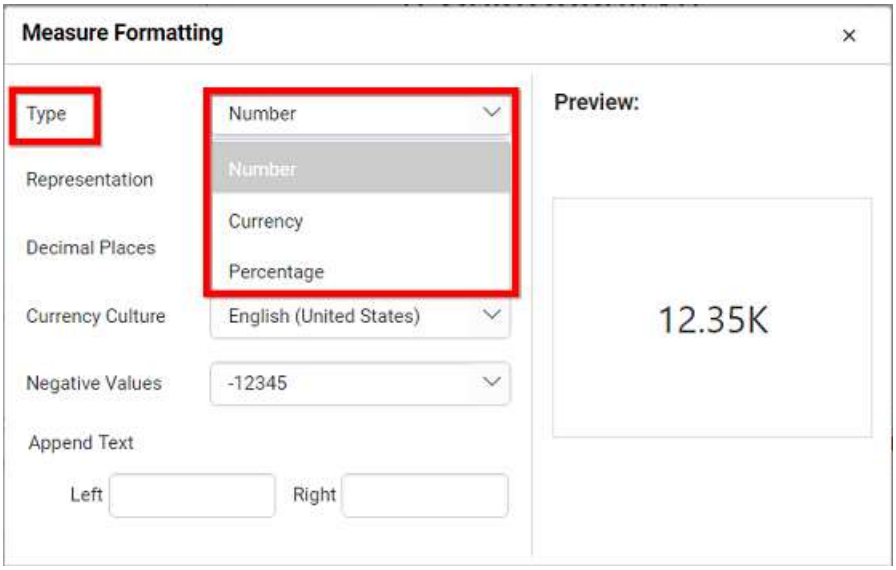
Max	Returns the highest value for all members.
Min	Returns the lowest value for all members.
Var	Calculates the variance of all members.
DistinctCount	Returns the distinct count of all members.

Formatting Measure Type Column

Measure type column values can be formatted based on the following different options:

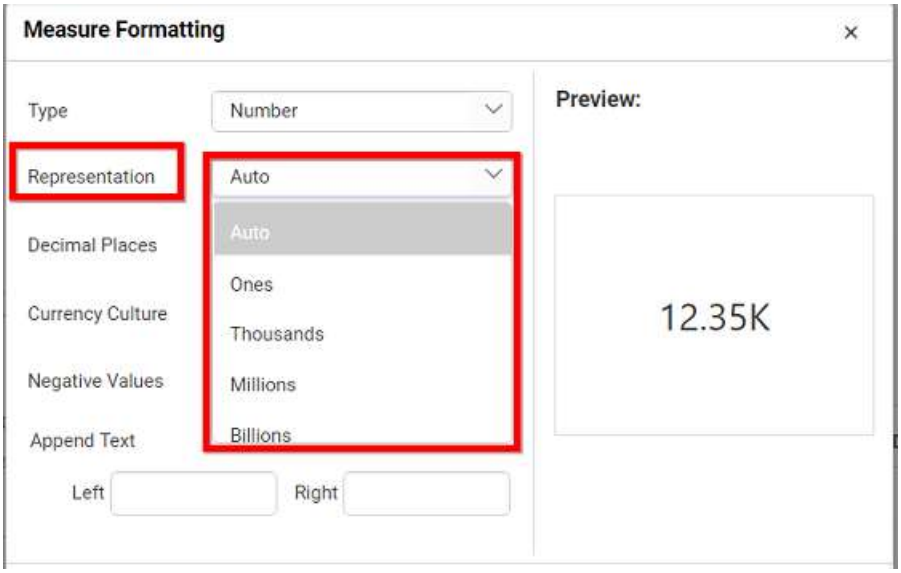
Type

The value display type of measure column can be defined based on the displayed data. For example, if you display the sales amount column, then the type can be defined as Currency.



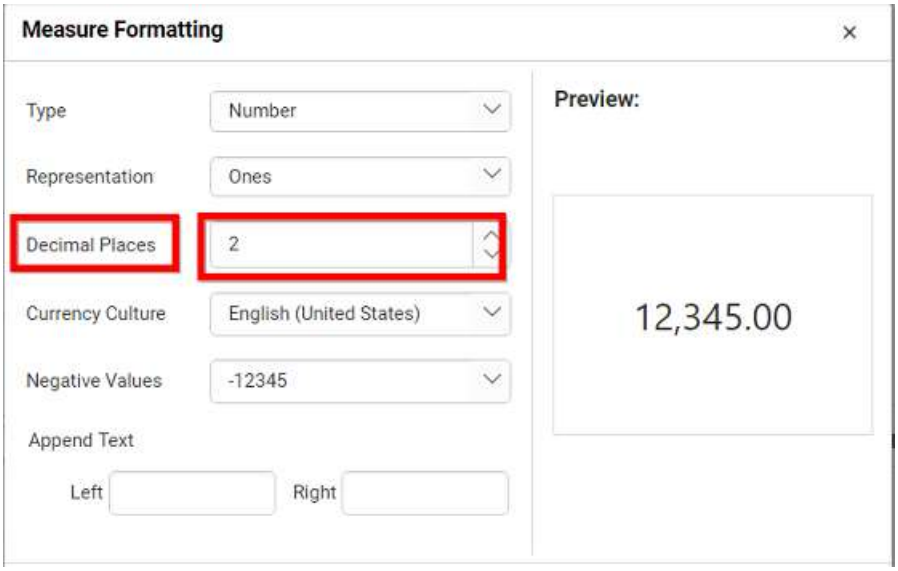
Representation

The value display format can be defined by the representation. For example, by selecting thousands, the value 10,000 will be displayed as 10K.



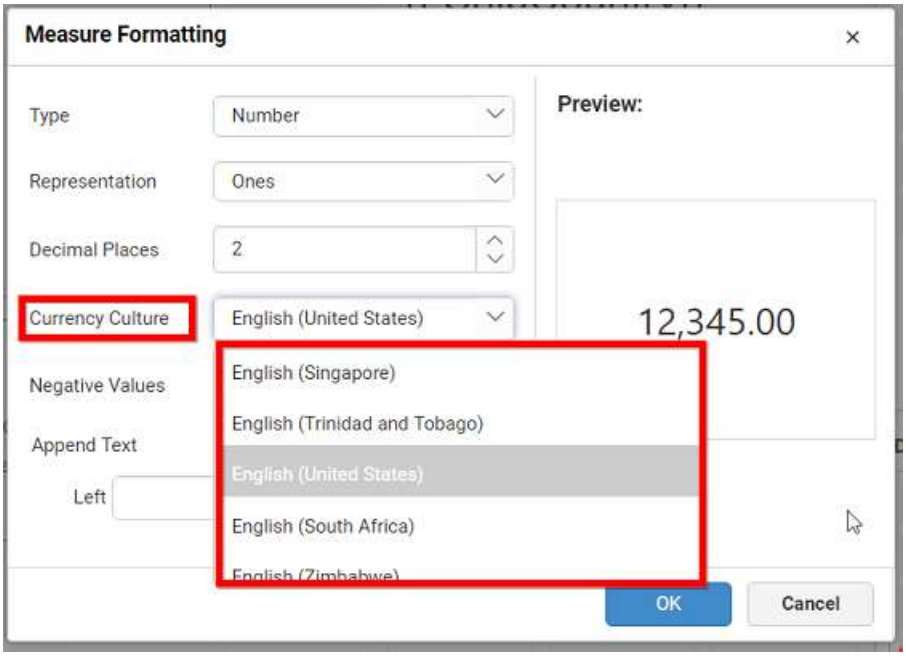
Decimal places

You can set the decimal places explicitly when the representation is set with options other than Auto.



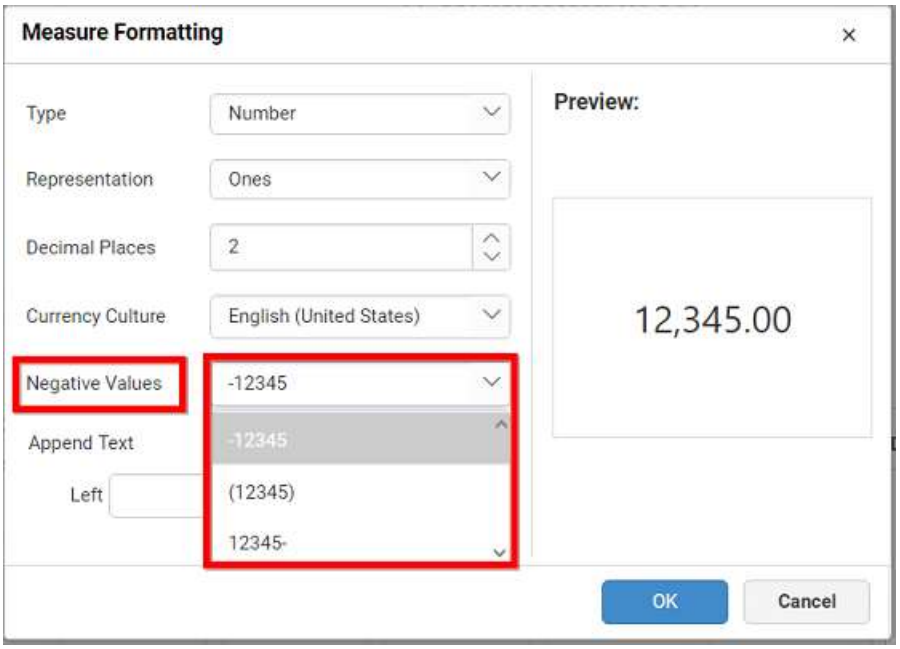
Currency culture

You can set the currency value culture when the value display type is set as Currency.



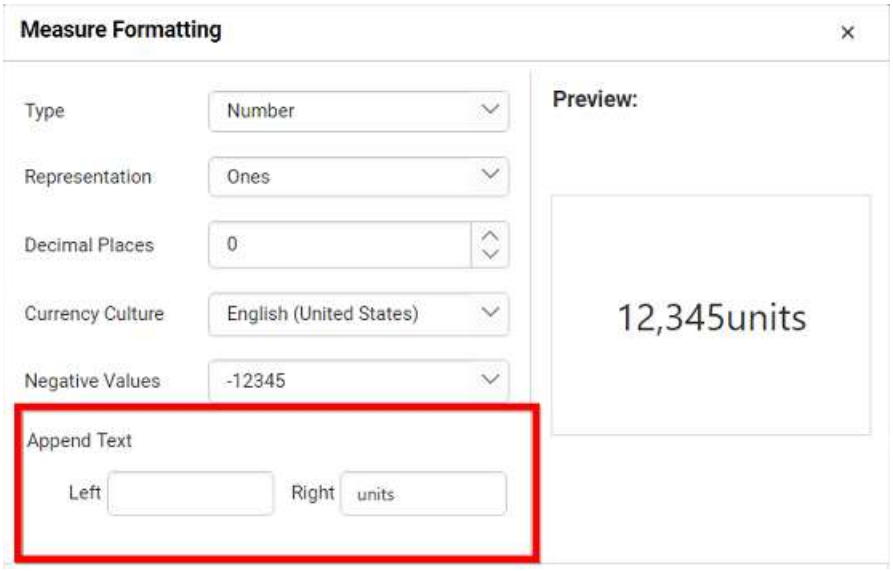
Negative values

You can set the negative value display format for number representation.



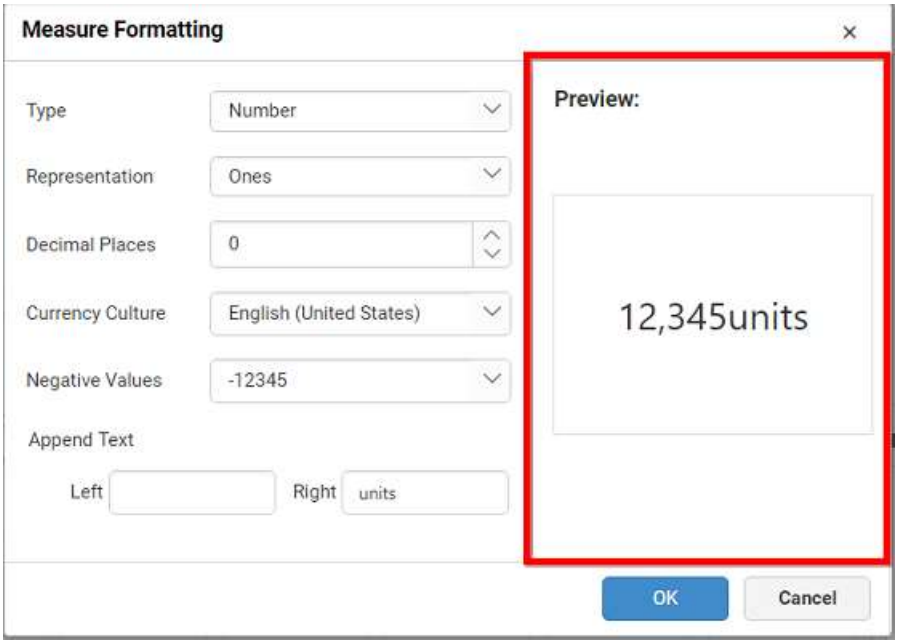
Append text

You can append text, character, number, or symbol either at start or at the end of the values.



Preview

This field provides the preview of display value in the Measure Formatting dialog based on the settings.



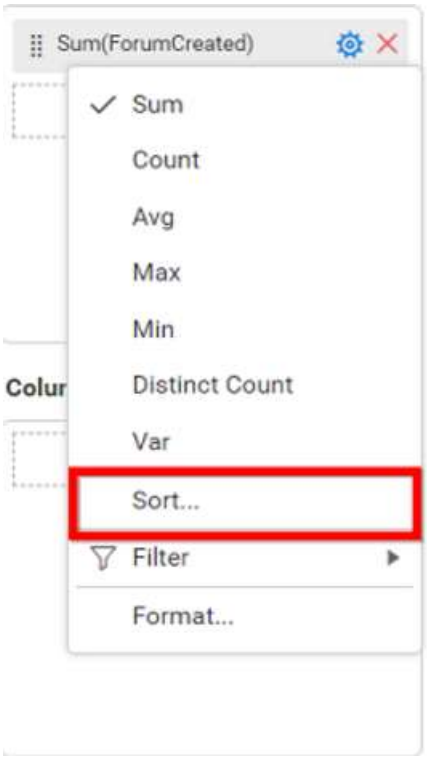
Sorting

You can customize the sorting behavior of dimension and measure fields in each widget. You can order them based on alphabet or value, data source (default), or field.

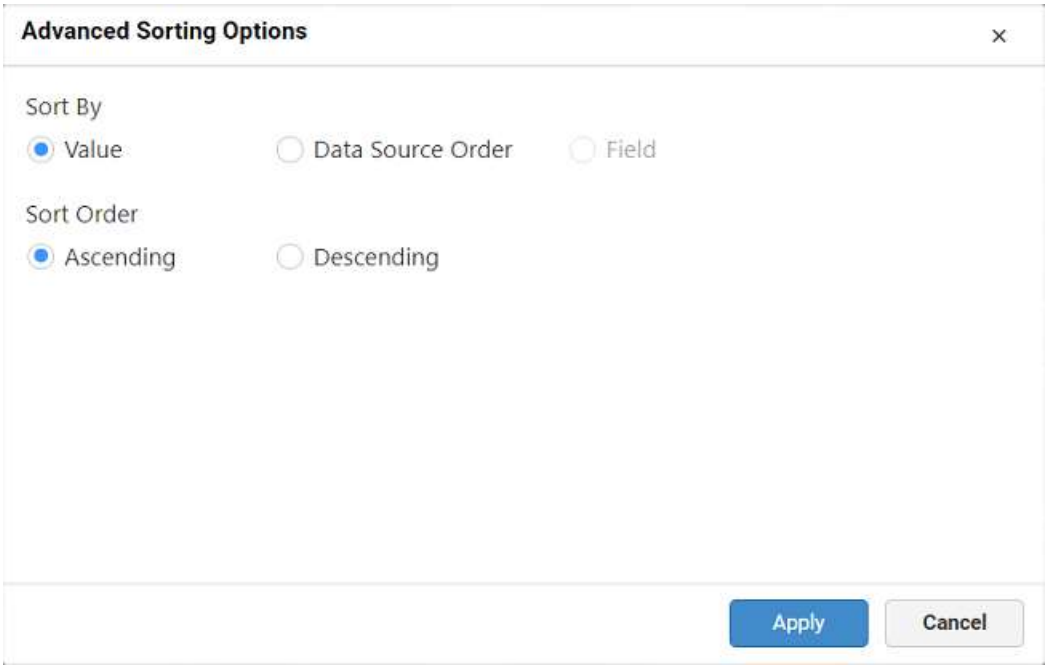
Measure column

To customize the sorting behavior of measure field, drag and drop the Measure Field into the control designer.

Click the Settings icon available in the Y Values section and select Sort option from the context menu.



The Advanced Sorting Options dialog will open as shown in the following image.



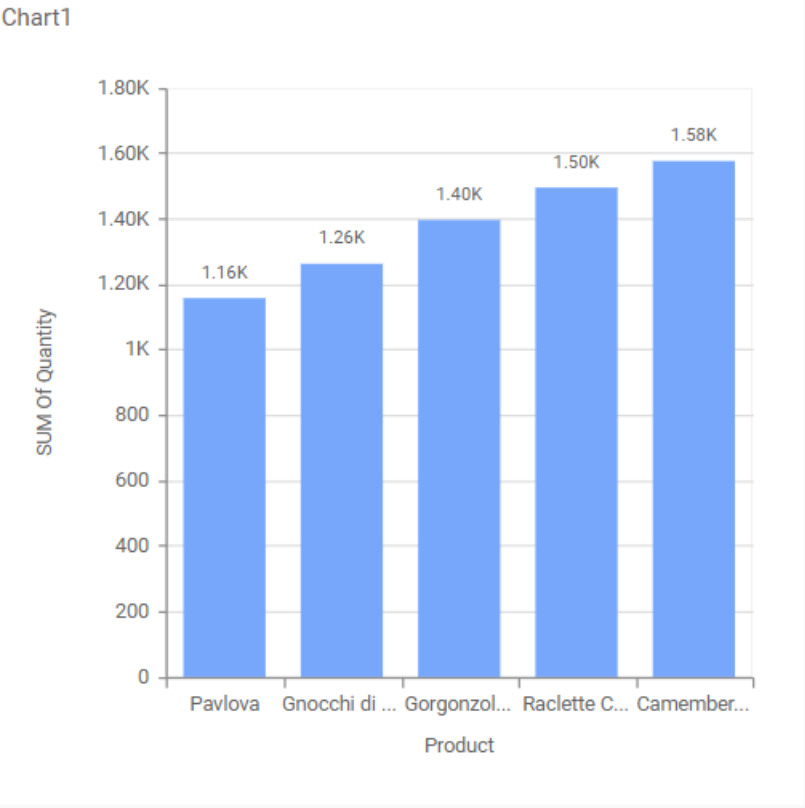
The options available in the Advanced Sorting Options dialog are:

Sort order

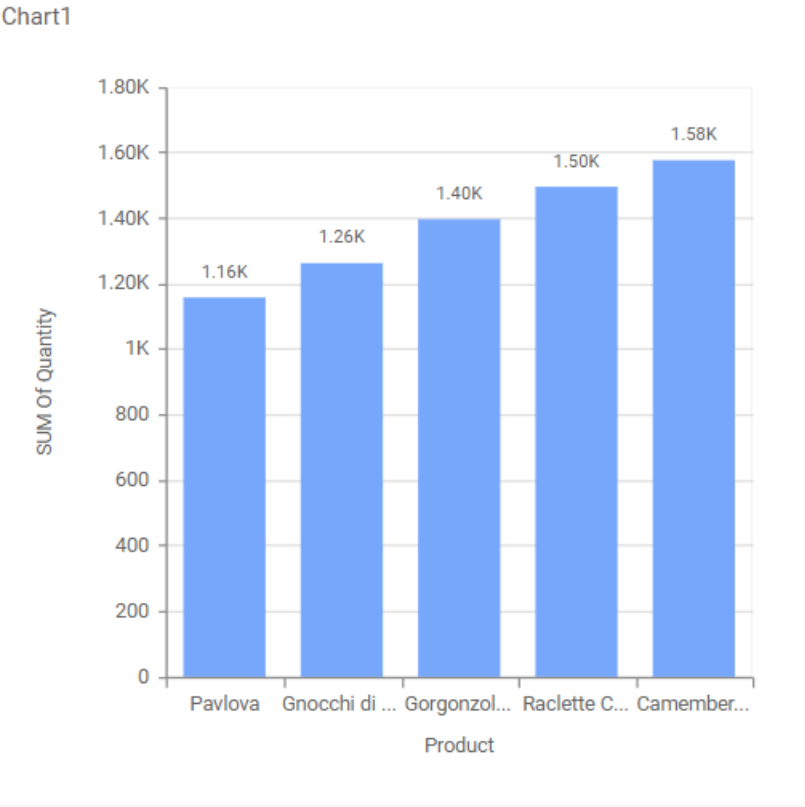
- Ascending: Displays the sorted results in the ascending order.
- Descending: Displays the sorted results in the descending order.

Sort by

- Value: Orders the data in either ascending or descending order based on values. You can apply this sorting for more than one number of fields. Here, the data has been ordered in hierarchical pattern as follows.



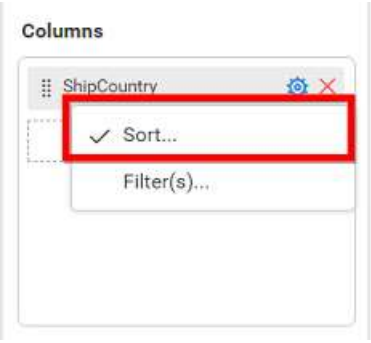
- Data source: Places resultant data from the data source on query execution, i.e., without performing any additional operations such as ascending or descending as follows.



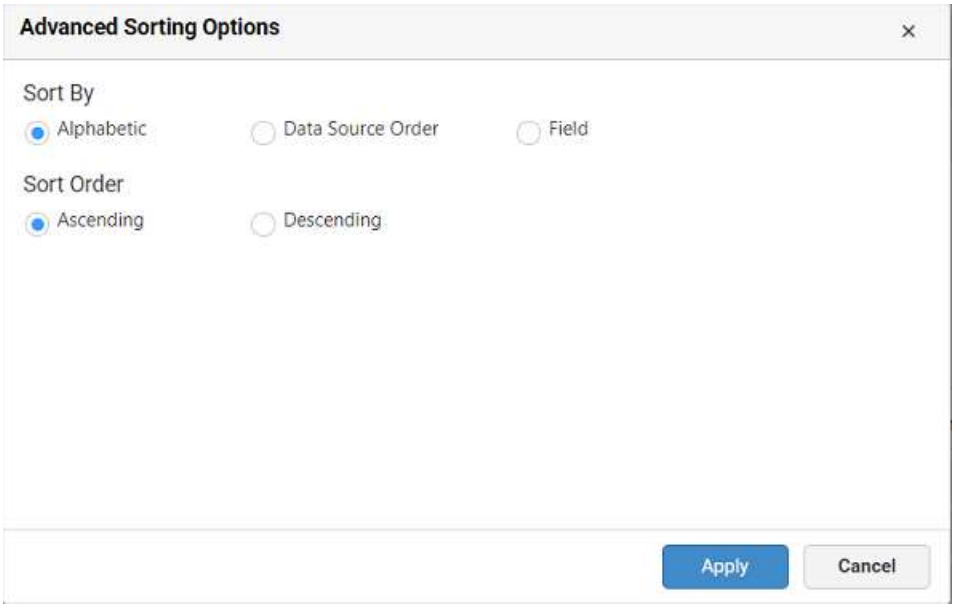
Dimension column

To customize the sorting behavior of dimension field, drag and drop the Dimension Field into the control designer.

Click the Settings icon available in the Columns or Rows or Series section, and select Sort from the context menu.



The Advanced Sorting Options dialog will open as shown in the following image.



The options available in the Advanced Sorting Options dialog are:

Sort order

- Ascending: Displays the sorted results in the ascending order.
- Descending: Displays the sorted results in the descending order.

Sort by

- Alphabetic: Orders the data either in ascending or descending order based on initial alphabet. You can apply this sorting for more than one string field. Here, the data has been ordered in hierarchical pattern as follows.

Grid1

ShipCountry	CustomerID	Freight Charge
Argentina	CACTU	73
Argentina	OCEAN	307
Argentina	RANCH	219
Austria	ERNSH	6,205
Austria	PICCO	1,186
Belgium	MAISD	459
Belgium	SUPRD	821
Brazil	COMMI	188
Brazil	FAMIA	233
Brazil	GOURL	322
Brazil	HANAR	725

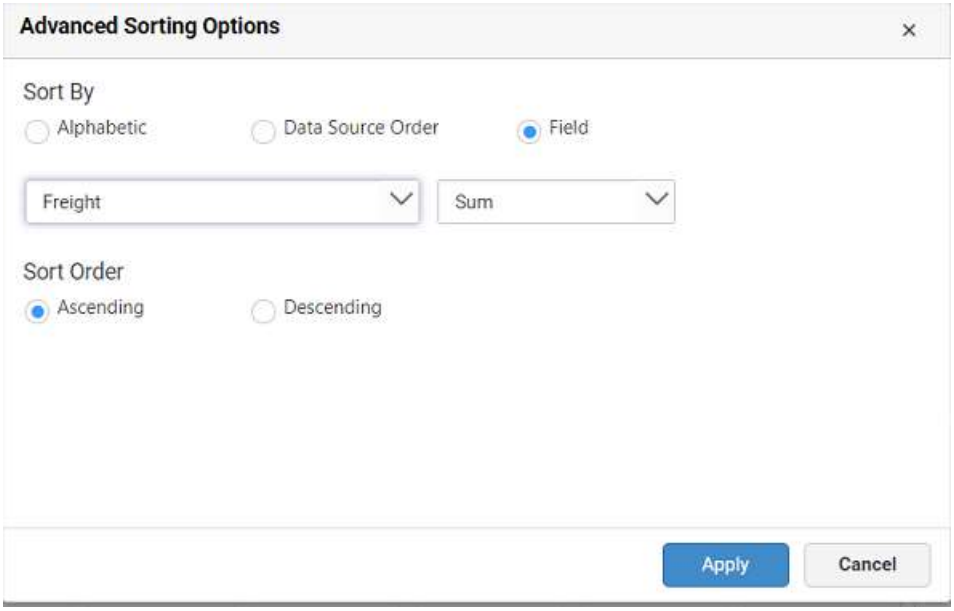
- Data source: Places resultant data from the data source on query execution, i.e., without performing any additional operations such as ascending or descending as follows.

Grid1

ShipCountry	CustomerID	Freight Charge
Germany	ALFKI	226
Mexico	ANATR	97
Mexico	ANTON	269
UK	AROUT	472
Sweden	BERGS	1,560
Germany	BLAUS	168
France	BLONP	624
Spain	BOLID	191
France	BONAP	1,358
Canada	BOTTM	794
UK	BSBEV	281

- Field: Orders the data based on the associated values of another measure or dimension field. For example, you can order several countries based on their freight values.

For example, the sort dialog box shown below is configured to sort the "Ship Country" field based on sum of the "Freight" measure in the descending order. The results will be displayed in such a way that the "Ship Country" with lowest "Freight" value is displayed first and the "Ship Country" with the second lowest "Freight" value is displayed second, and so on.



Grid1

ShipName	CustomerID	SUM Of Freight
Centro comercial ...	CENTC	3
Laughing Bacchu...	LAUGB	10
Lazy K Kountry St...	LAZYK	19
Alfreds Futterkiste	ALFKI	29
North/South	NORTS	38
Galería del gastro...	GALED	38
Consolidated Hol...	CONSH	54
Vins et alcools C...	VINET	58
Du monde entier	DUMON	64
Romero y tomillo	ROMEY	64
GROSELLA-Resta...	GROSR	68

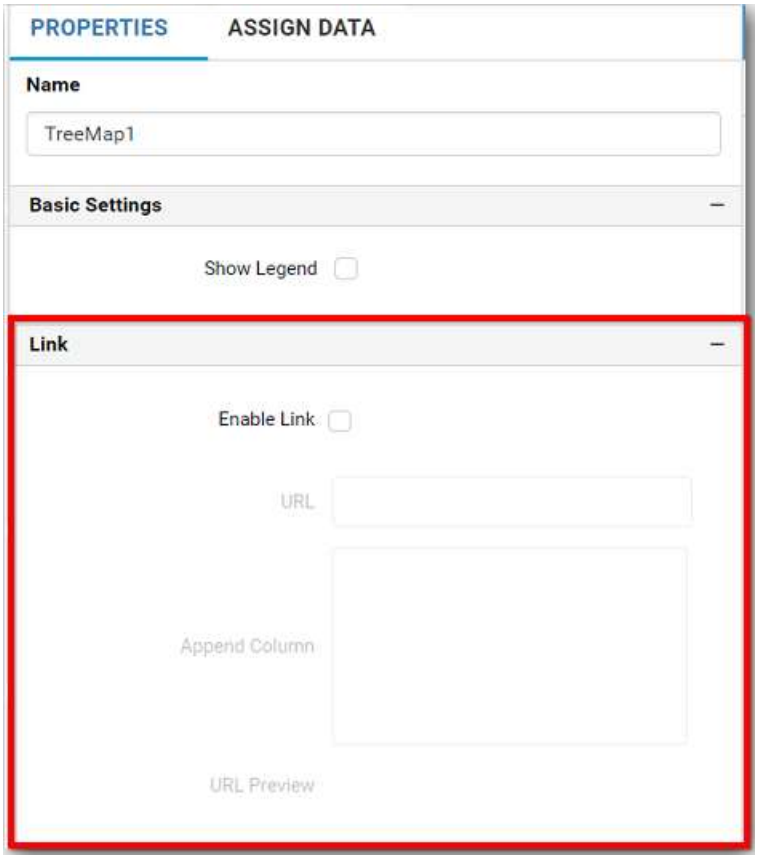
Note: Advanced sorting is not available for "date" type, "date time" type, raw data, and proportional charts such as pie, doughnut, pyramid, and funnel.

Linking URLs

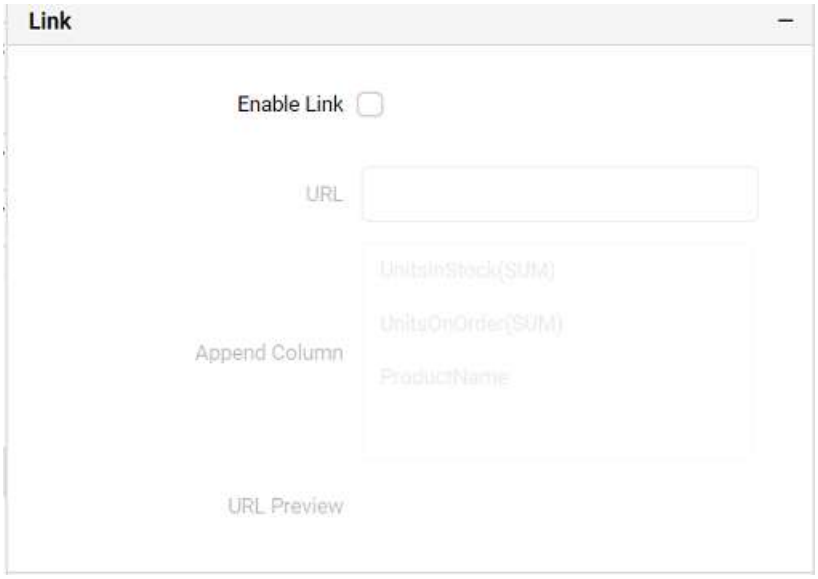
URL linking allows you to link the dashboard with valid web URLs.

You can link URLs to a visualization widget by enabling the Enable Link property.

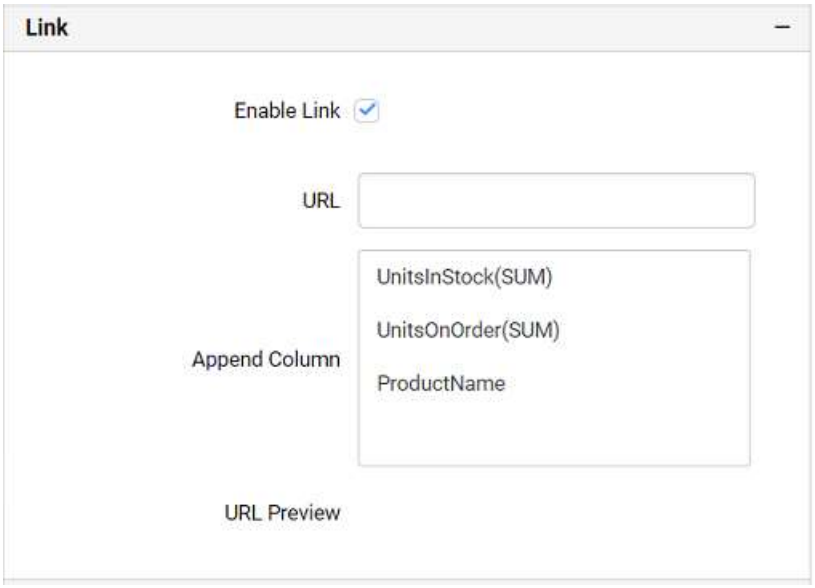
The Enable Link option is available in the Properties tab of the widgets.



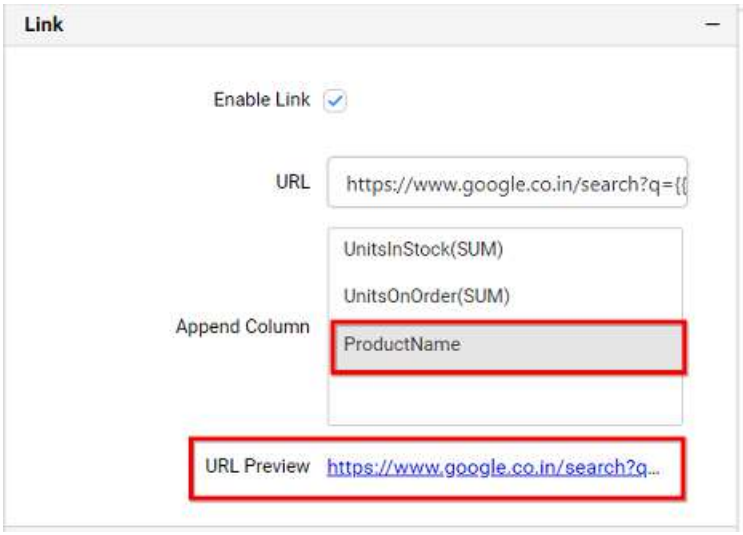
By default Enable Link property will remain unchecked.



To enable linking select the Enable Link checkbox.



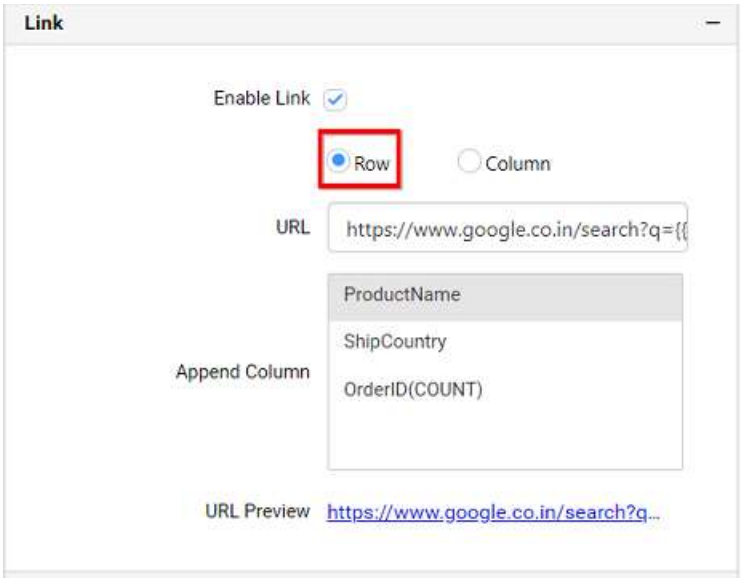
Enter the Web URL in the URL text box. If you click on the column names listed in the Append Column name list , it will be appended to the URL entered in the URL text box.



For Grid widget, you can get URL based on Row and Column.

Row

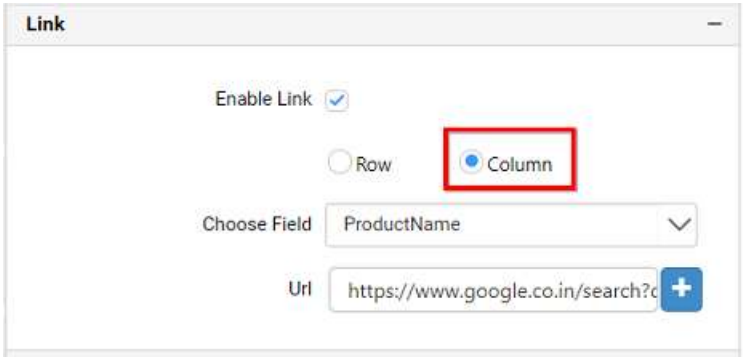
Enter the Web URL in the URL text box. If you click on the column names listed in the Append Column name list , it will be appended to the URL entered in the URL text box.



You can preview the linked URL using the URL preview option. If you click the preview URL link,it will be opened in a browser.

Column

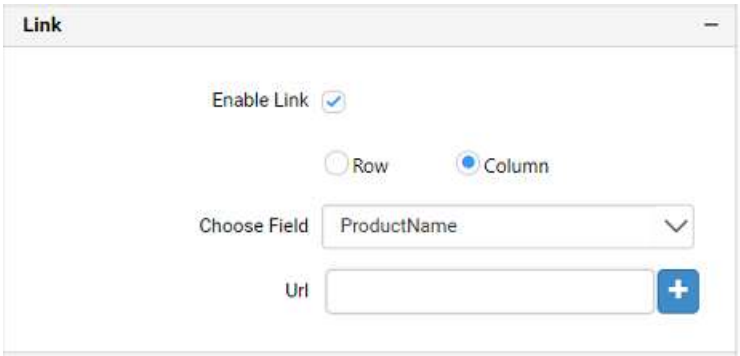
You can select the column.

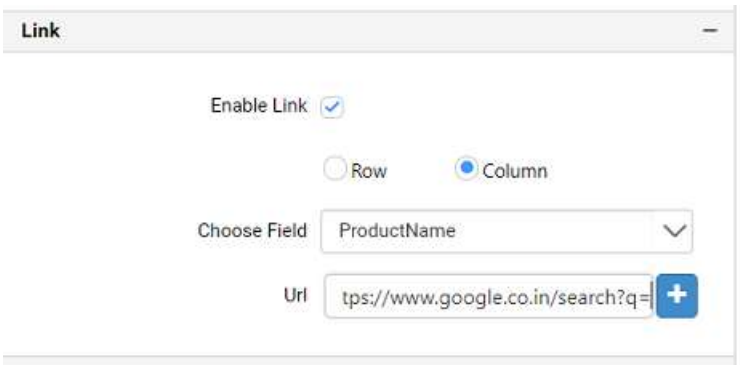


Choose the field from Choose Field property listed in the combo box.

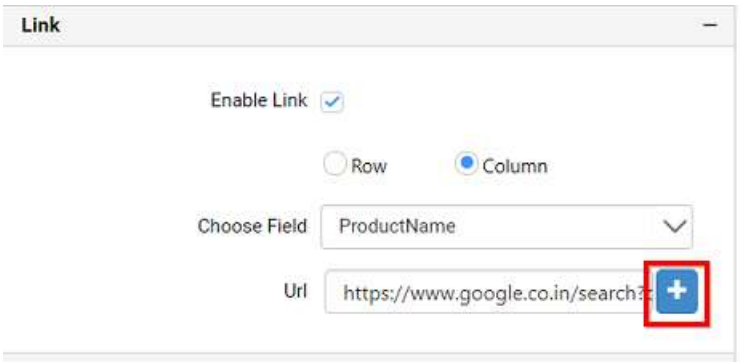
!Linking09[](images/Linking09.png)

Add the web Url in the URL text box.

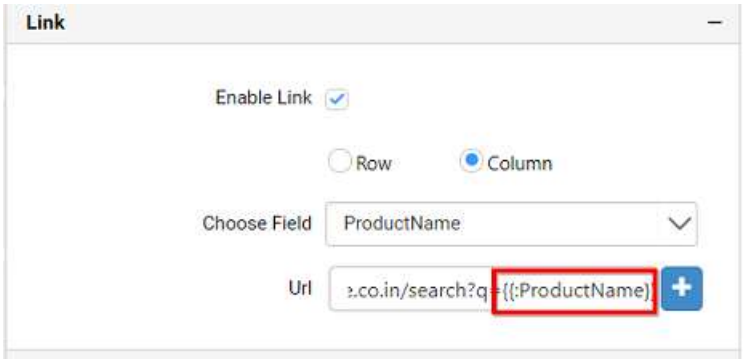




Click Add button shown as below to add the field.



Now, the field will be added with URL.

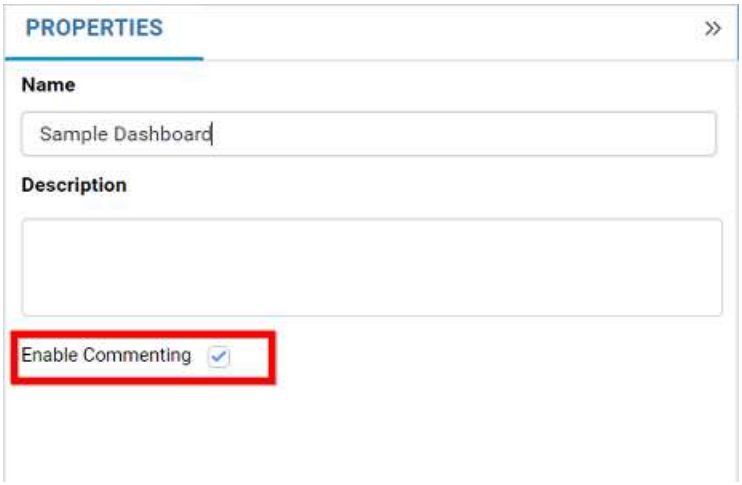


Commenting Dashboard and Widget

You can comment over a dashboard and/or its individual widgets in the site. You can toggle this setting through the Enable Commenting option exposed in the properties tab of the dashboard. For widget, toggle the Enable Comments option in the properties tab of individual widgets.

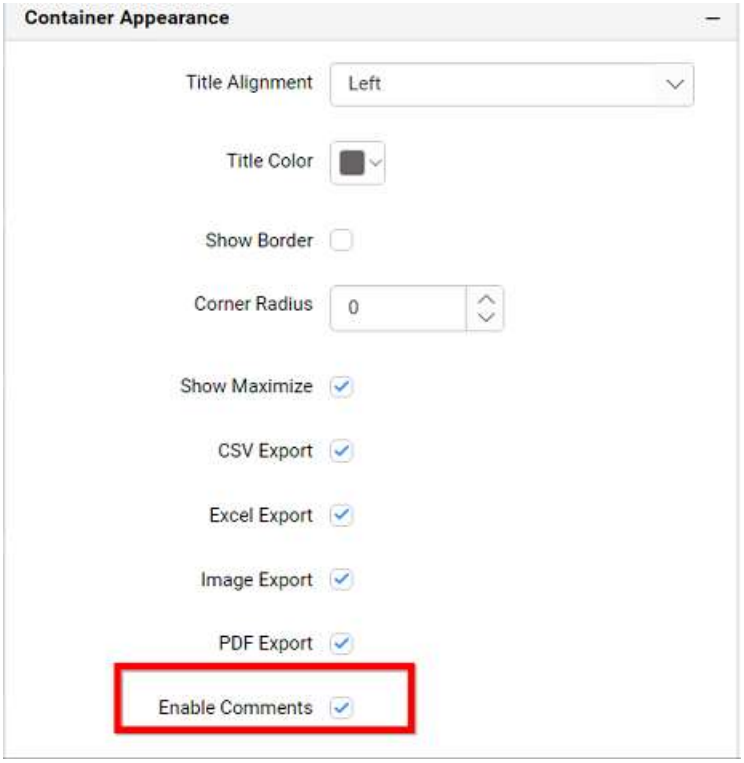
Commenting a dashboard

To enable comments on dashboard, go to the Properties Tab and enable the Enable Commenting option. By default, this option is enabled.



Commenting a widget

To enable comments for widget, select that widget after opening the Properties Tab, and select the Enable Comments under the Container Appearance section. By default, this option is enabled.



Note: Label widget do not have commenting support.

Desktop Designer Compatibility

Dashboards created with version 2.3.0.32 can be viewed and edited in the web designer (except the unsupported elements). Dashboards created in earlier versions of the desktop designer can be viewed in dashboard designer, but cannot be edited.

Unsupported data connections

      1. JSON type under File Connection

      2. Microsoft SQL Server Analysis Services

      3. SQLite

      4. Hive

      5. Spark SQL

      6. Microsoft Azure Table Storage

      7. Twitter

      8. Facebook

      9. LinkedIn

      10. Yahoo

      11. Instagram

      12. Dropbox

      13. GitHub

      14. ODBC – Microsoft SQL Server, MySQL, Microsoft Access, Other (ANSI SQL), Hive

      15. MySQL

      16. Oracle

Unsupported widgets

      1. Pivot Grid

      2. Bubble Chart

      3. Scatter Chart

      4. Combo Chart

      5. Range Navigator

      6. Radio Button

      7. Check Box

      8. Image

      9. Custom Widget

Unsupported features

      1. Multi-tabbed Dashboard.

      2. Dashboard Parameter.

      3. Relative Date Filter.

      4. Fiscal Year Start.

      5. Custom Date Time Formatting.

      6. Dashboard and Internal linking.

      7. Weighted Average and Standard deviation in Summary type.

      8. Functions like EMAIL, CURRENTUSER, FULLNAME are not supported for creating the expression.

      9. Manual Sorting is not supported in Custom Sorting.

      10. User-based filter.

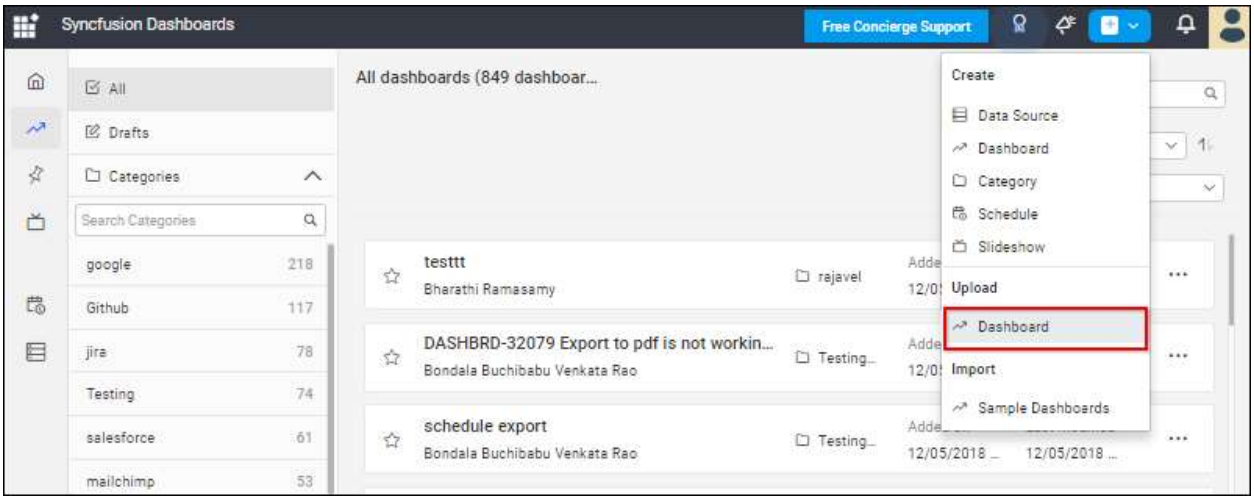
      11. Dashboard Filter Panel.

      12. Label parameter for Dashboard Title.

      13. Calculations section for Choropleth Map is not supported.

      14. Initial selection is not supported for widgets expect filter controls.

Note: To make use of the latest features, create the dashboards with the latest desktop designer and upload the dashboard to the dashboard server application using the upload option as shown in the following screenshot.



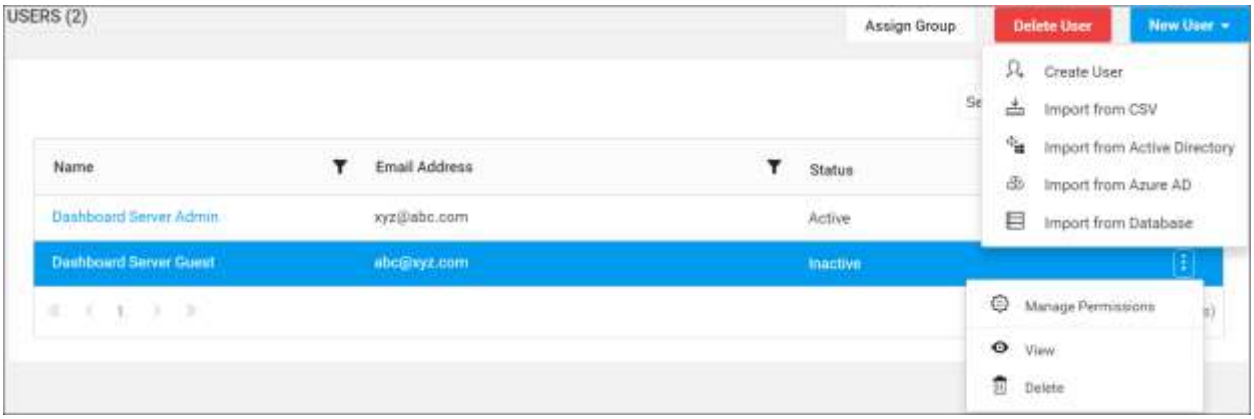
Administration

Users

Manage Users

This section explains on how to add, edit, activate, deactivate, delete users and also on how to manage the permissions and assign users to groups in the Syncfusion Dashboard Server.

Users can only be added/edited/deleted by the users, belonging to the System Administrator group.



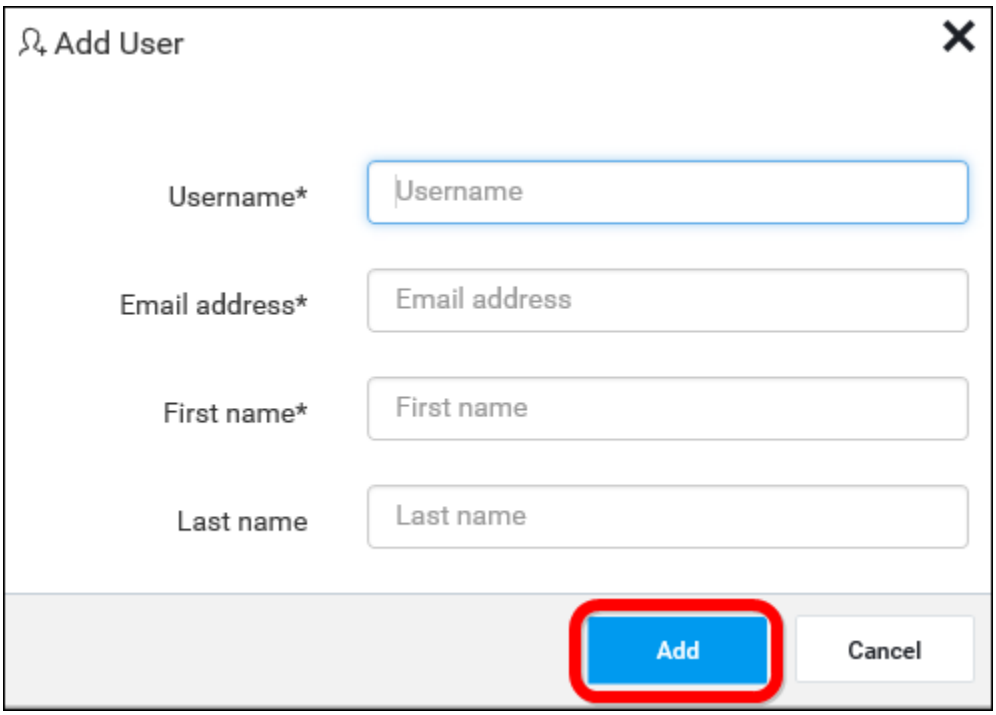
Add new users

New users can be added to the Dashboard Server individually or in bulk using CSV import

Add individual users

To add new users to the dashboard server, click on New User and then Create User from the User Management page.

The Add User dialog will be shown as like in the image below.



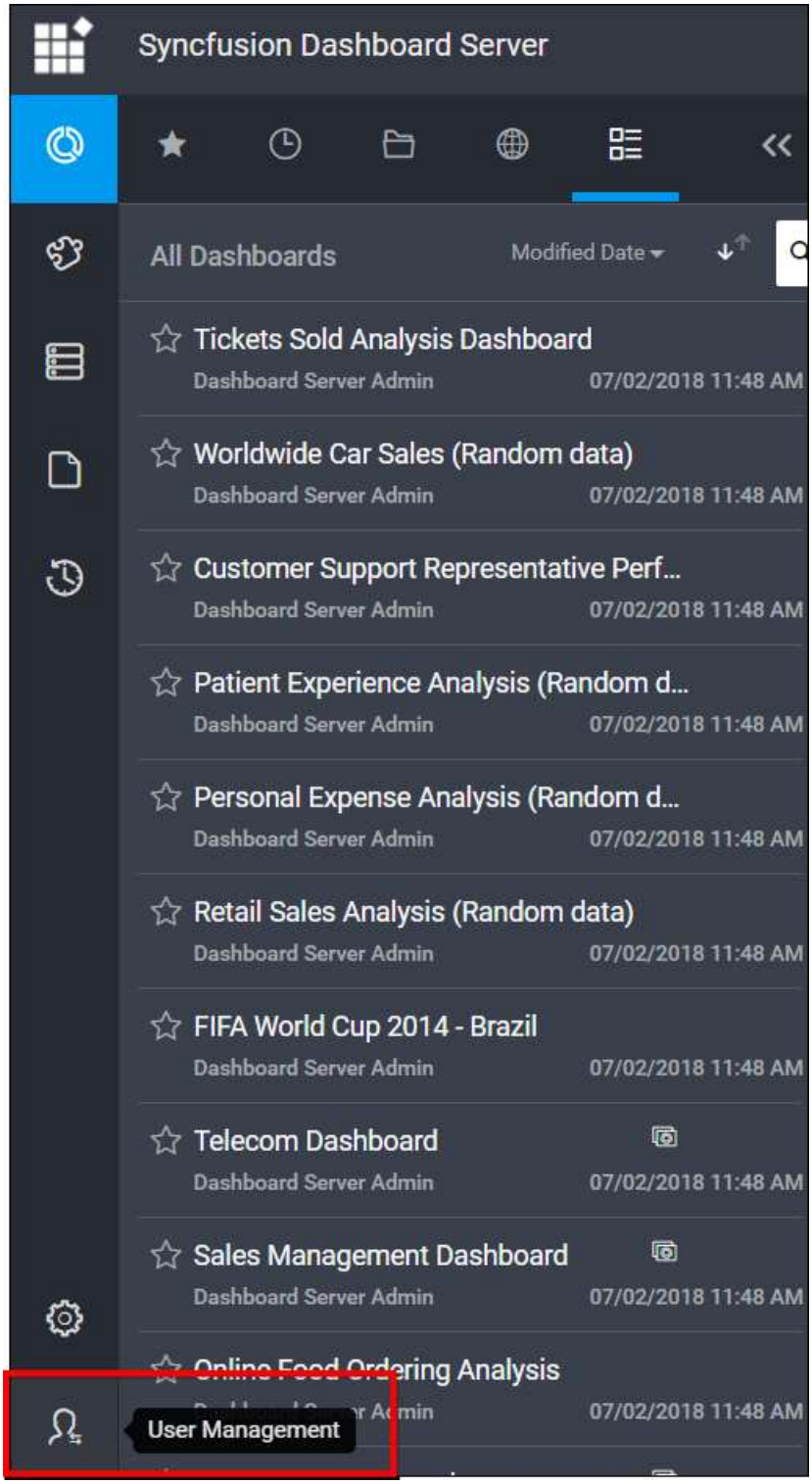
Fill the form with Username, Email address, First name and Last name and click on Add.

New account will be created for users in the Dashboard Server. Regarding account activation, refer to User Settings section for more details.

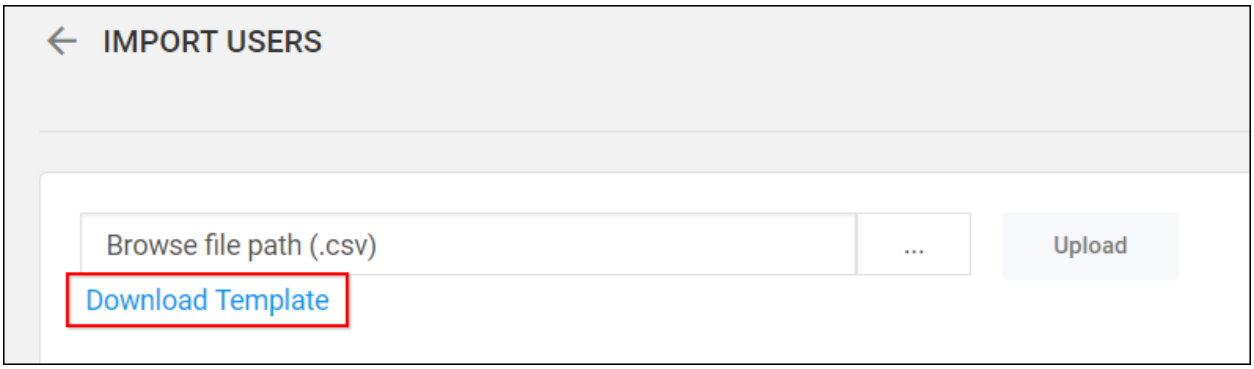
Import users from CSV

To automate the process of adding large number of users to a Syncfusion Dashboard Server, you can download the CSV template file and add the users in it and then import the file.

You can navigate to user management page by click User Management drop down under the Admin menu as below.

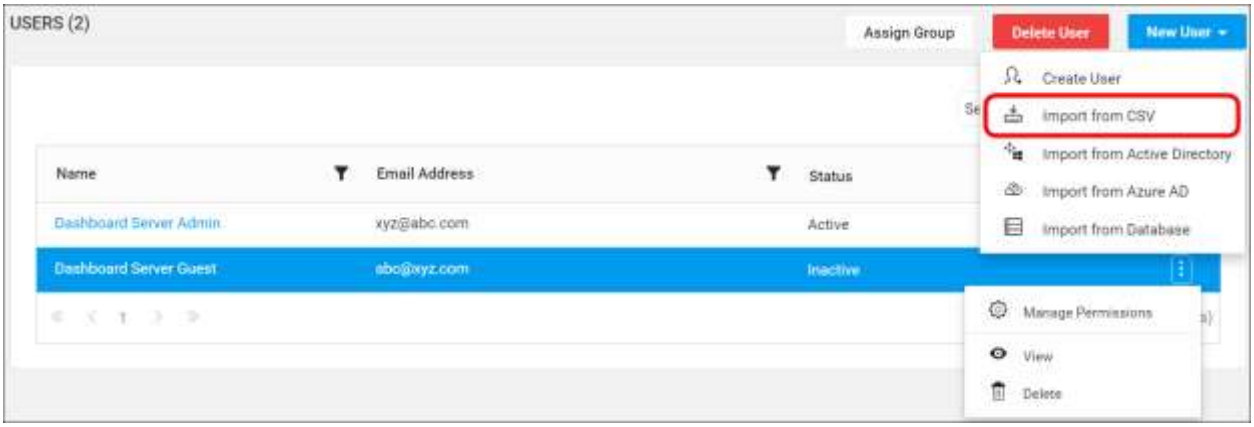


Note: The section will explain on how to import the users from CSV file which can be downloaded from below location in Dashboard Server application.



Add users from CSV file

In Dashboard Server, click Import from csv.



CSV file Requirements

The first row in the CSV template represents the column heading. Syncfusion Dashboard Server assumes that the data from the second line in the file represents the user.

We have the following two types of User account activation.

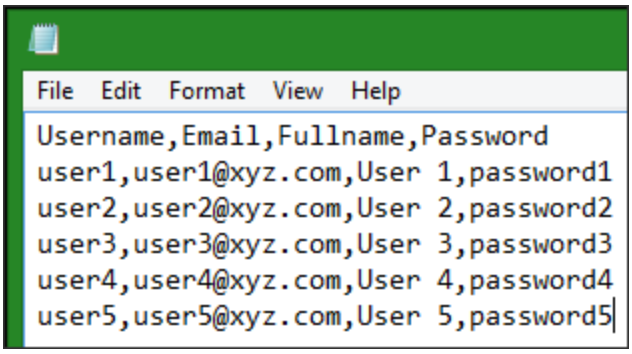
- Automatic
- Email

We have the following columns are considered as mandatory in the downloaded CSV file.

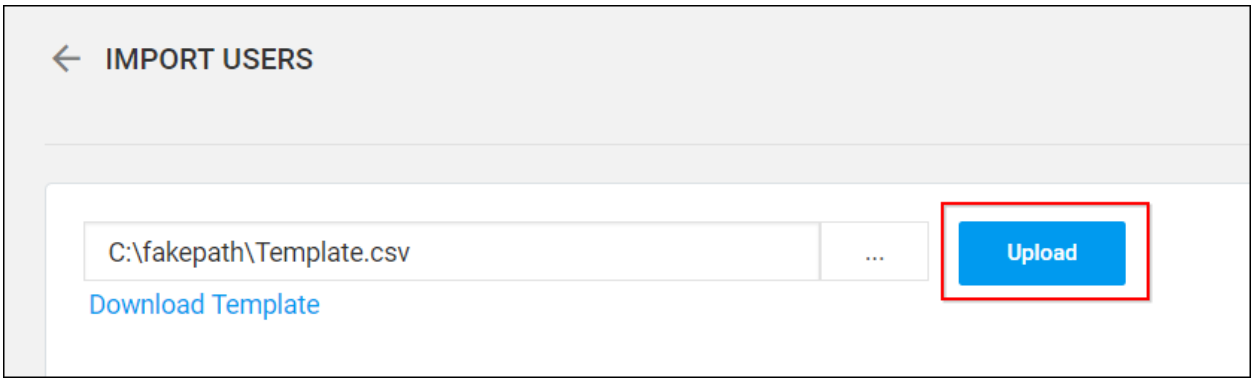
- Username
- Email address
- Full Name
- Password : If the Syncfusion Dashboard Server configured with Automatic account activation, password field should be filled. Otherwise we can leave the password field as empty.

Follow the below steps to add users using the CSV template

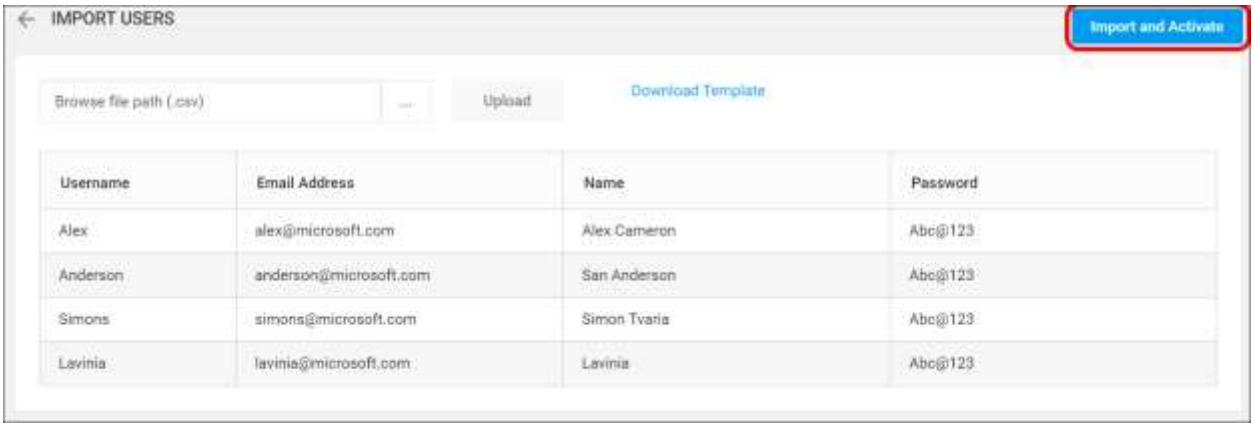
1. Download CSV template.
2. Add users in the CSV file.



3. Save the CSV file and upload it.

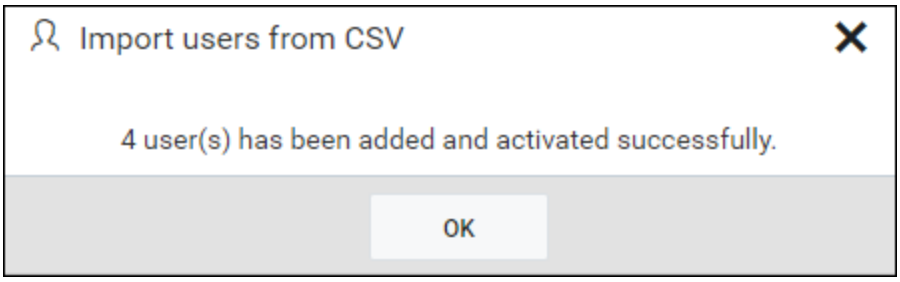


4. Once the file is uploaded the user details will be shown in the grid as like in the below image.



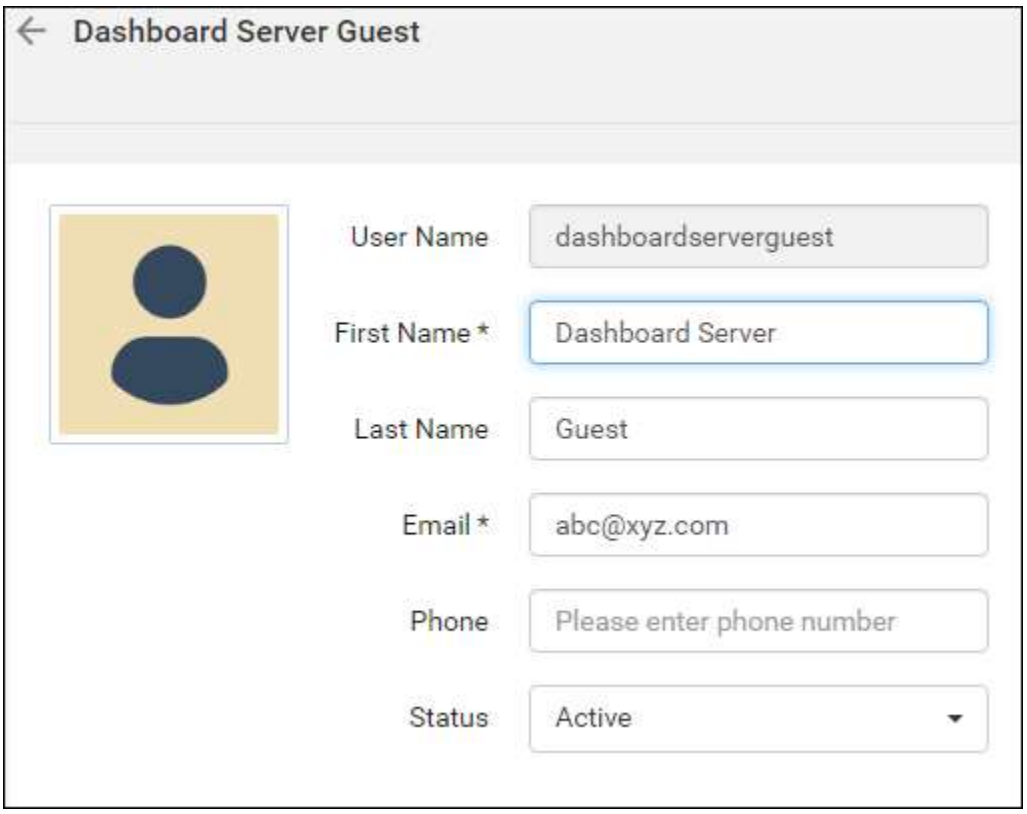
If the Syncfusion Dashboard Server configured with Email activation, the activation mail will be send to the user's mail Id. If the Syncfusion Dashboard Server configured with Automatic activation, the user will be automatically activated.

5. After uploaded the users in Dashboard Server the results are displayed as below.

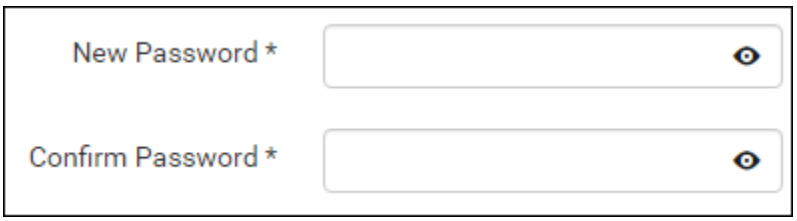


Edit users

User profile details can be edited from the users edit page as shown in the below image.



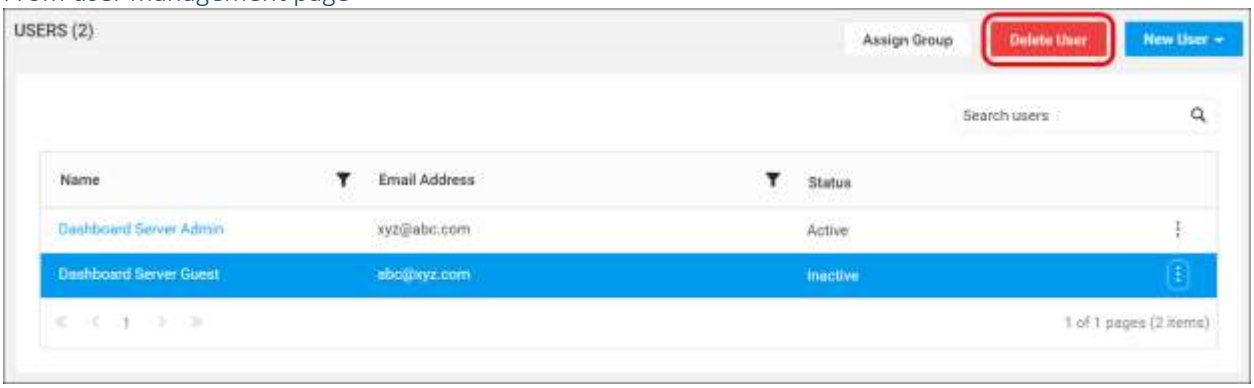
First Name, Last Name, Email address, Phone number and profile picture and the login password for the user can be edited by the user belonging to the 'System Administrator' group.



Delete users

Users can be deleted from the Dashboard Server when the user no longer requires the access. Users can be deleted from the user management page or from the edit page.

From user management page

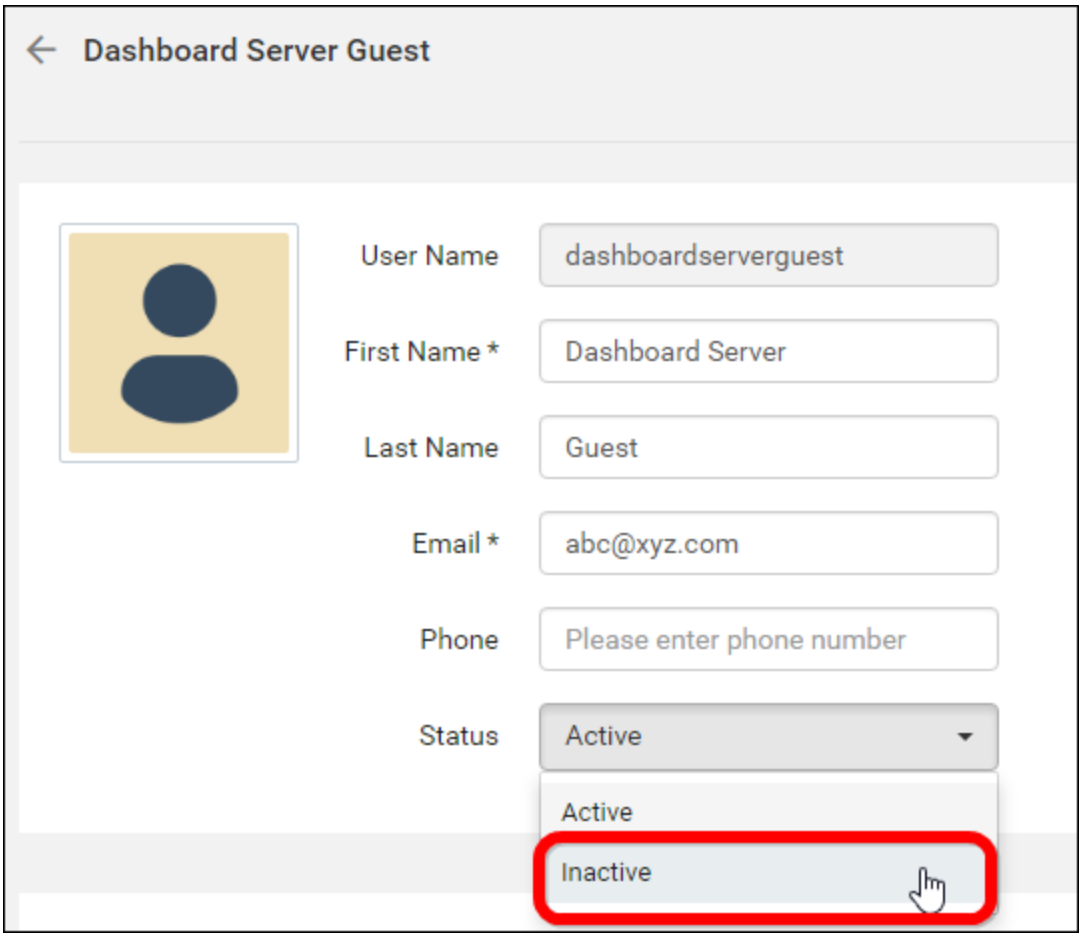


From user edit page

Deactivate users

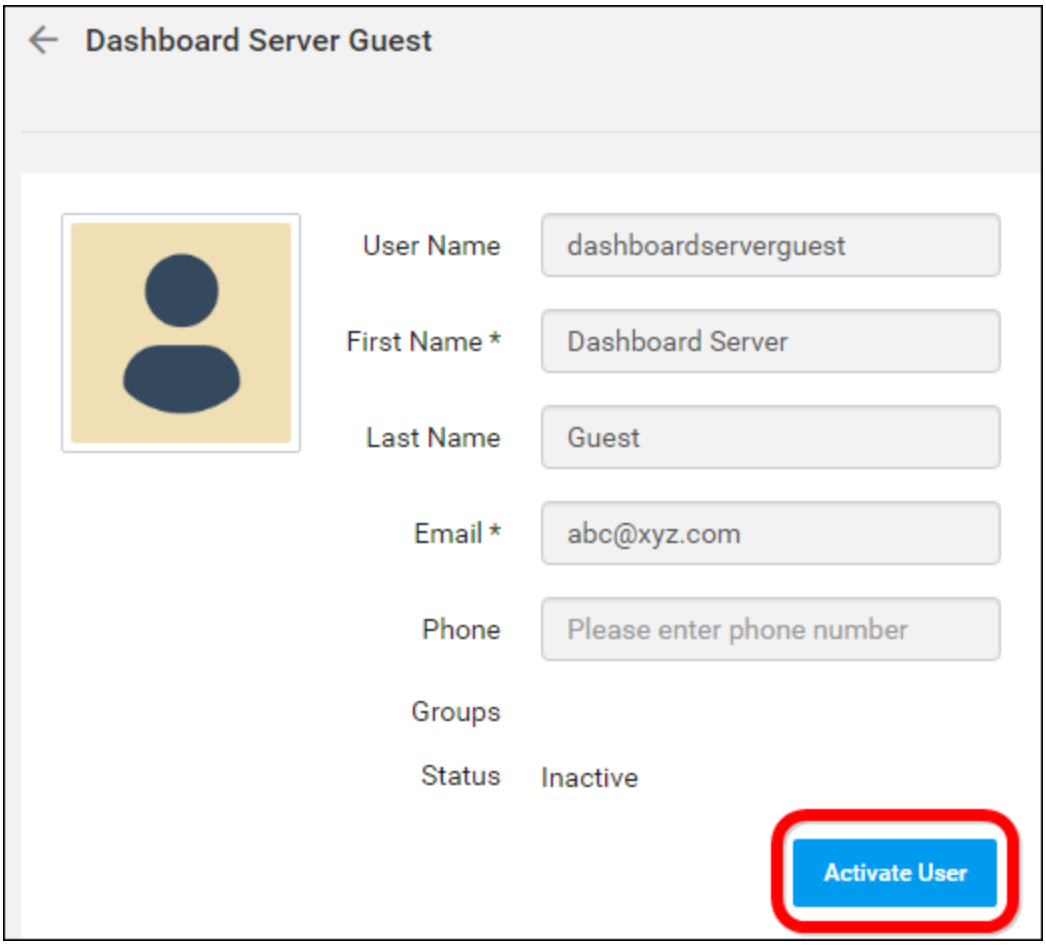
Users can be deactivated at any time. Once deactivated, the user cannot log into the Dashboard Server.

To deactivate a user, select inactive from the status dropdown in the user edit page.



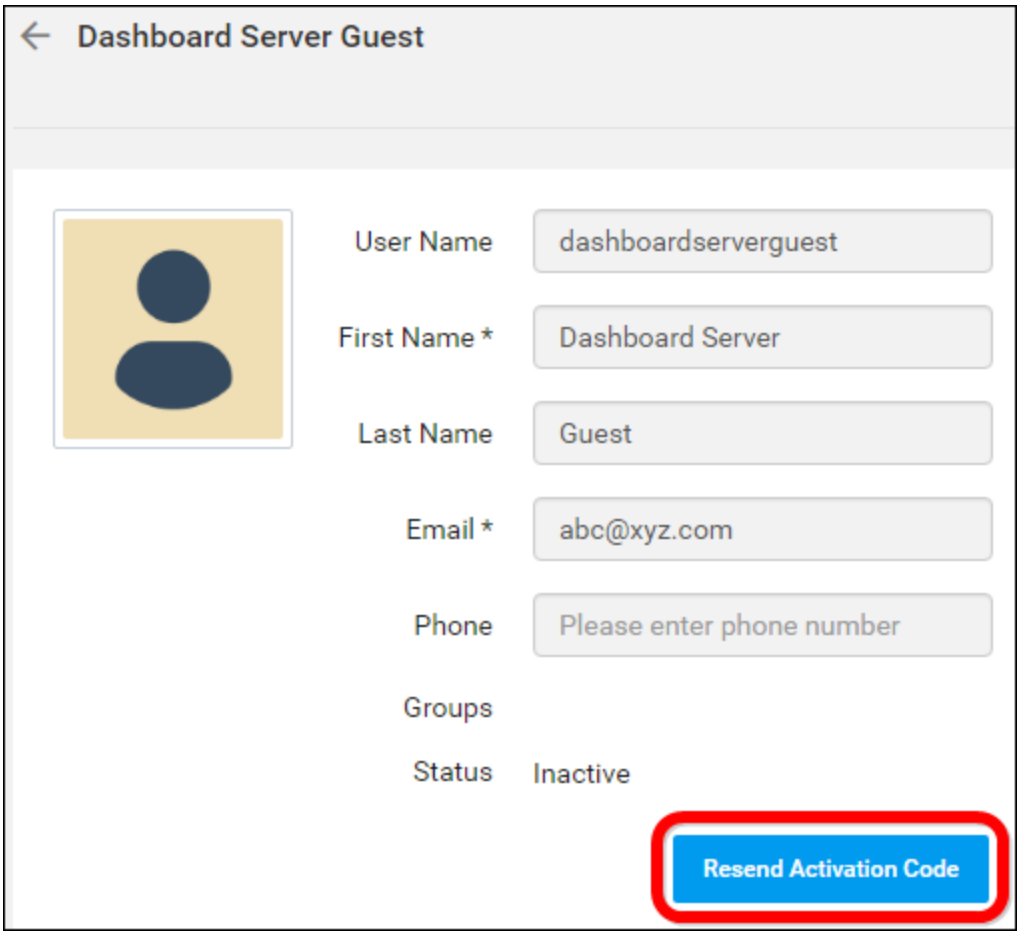
Activate users

Inactive users can be activated by clicking on the Activate User button in the user edit page.



This will send an account activation email to the user with an activation link to activate the account and again this activation link will be valid only for 3 days.

If the user has not received the activation email within 3 days or missed to activate the account, the System Administrator has to resend the activation email to the user.



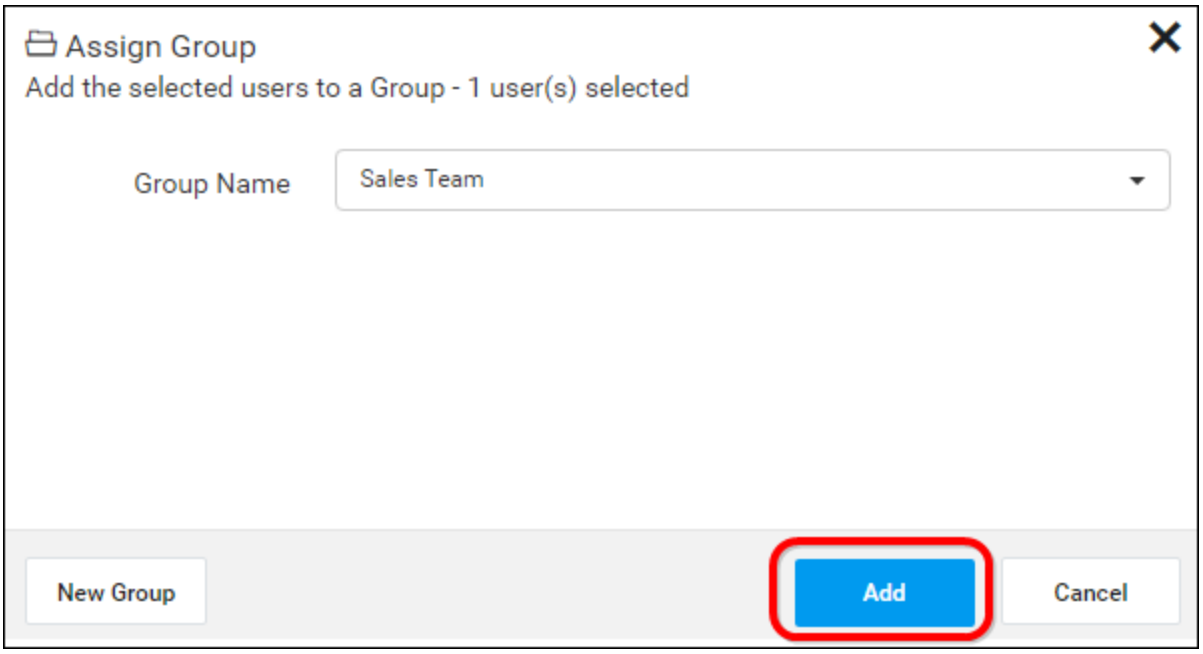
Manage permissions

Check the Manage Permissions section to learn how to manage permissions to an user.

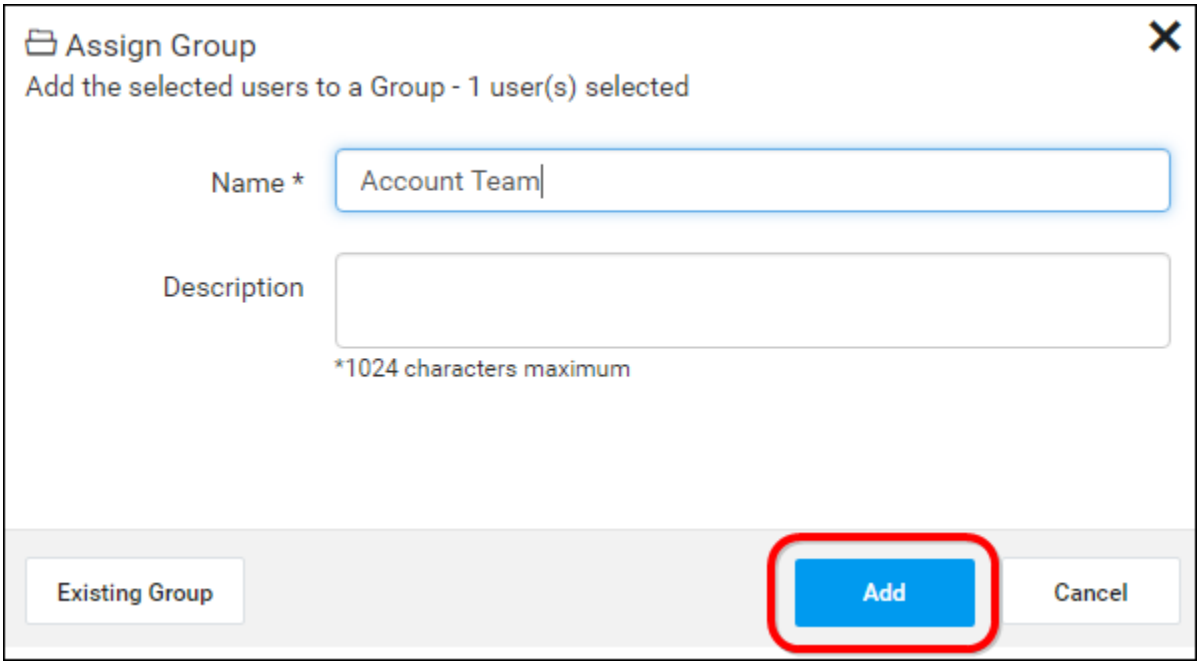
Assign users to group

Users can be assigned to one or many groups from the user management page.

Users can be assigned to an existing group.



A new group can also be created at this time and the selected users can be assigned to the new group.



Note: All the users in the group will have the permissions of assigned group.

Active Directory User Import

This section explains how to search and import users from Active Directory into the Syncfusion Dashboard Server.

Note: Active Directory connection has to be configured in the Active Directory Settings in the General page for importing users.

Users belonging to the System Administrator group only can import Active Directory users into the Dashboard Server.

Search Users

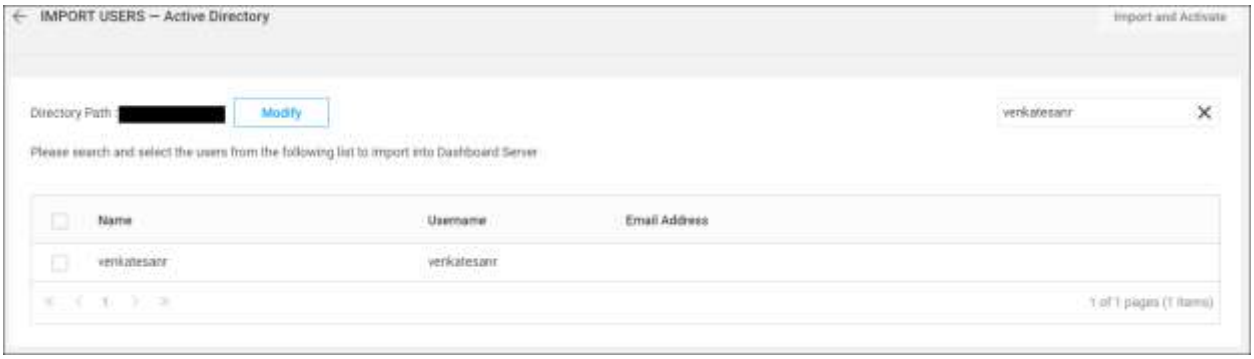
Initially, any Active Directory users cannot be displayed until searching for the user.

You can search the Active Directory users with any one of the below properties and choose them to import into the Dashboard Server.

- User name
- First name
- Last name
- Email Address
- Display name

A maximum of 1000 users will be searched and pulled from Active Directory in a single request.

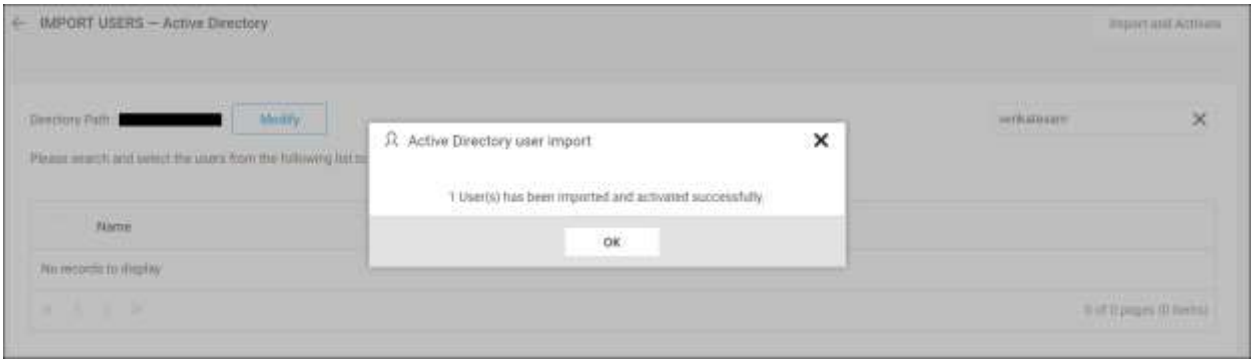
Dashboard Server will list the search results of the users in the grid as shown in the below figure.



Import Users

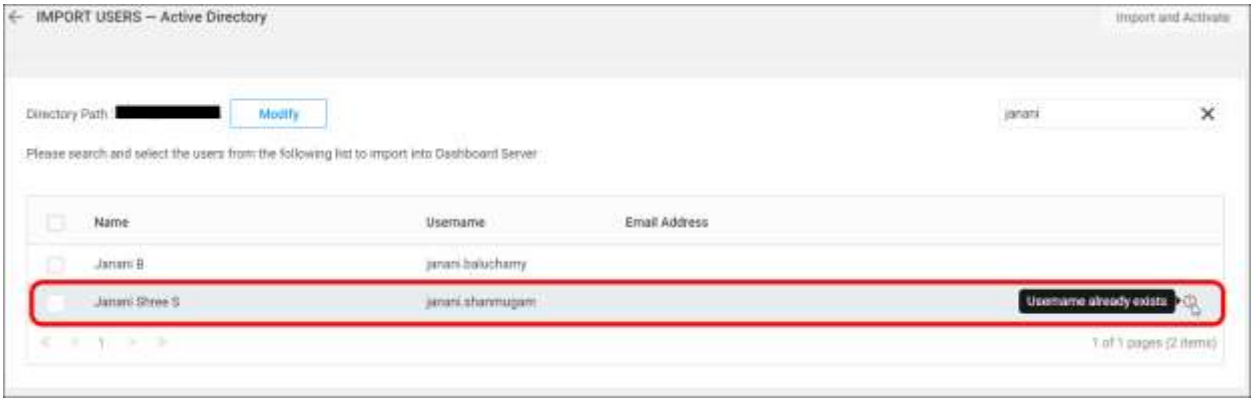
To import the Active Directory users into the Dashboard Server, you have to choose the users from the list and click on the Import and Activate button at the top right corner.

Dashboard Server will import the chosen users and a confirmation message will be displayed as shown in the below figure.



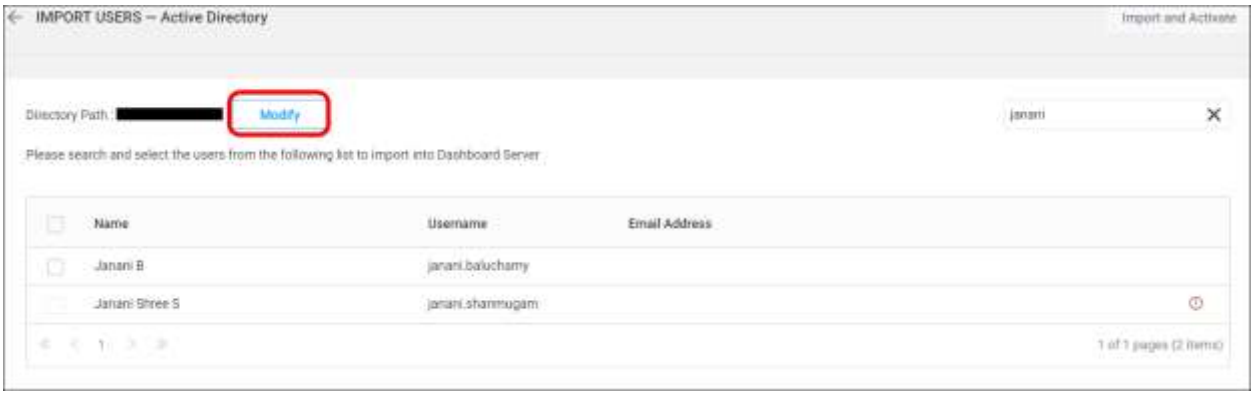
Duplicate Users

Active Directory users who has the same username or email address as that of the Dashboard Server users(who are already present) will be marked as duplicate users and will not be allowed to import into Dashboard Server.



Modify Active Directory Connection

To modify Active Directory configuration settings, click on the Modify link as below



Active Directory User Synchronization

This section explains how to synchronize the imported Active Directory users details with the Active Directory.

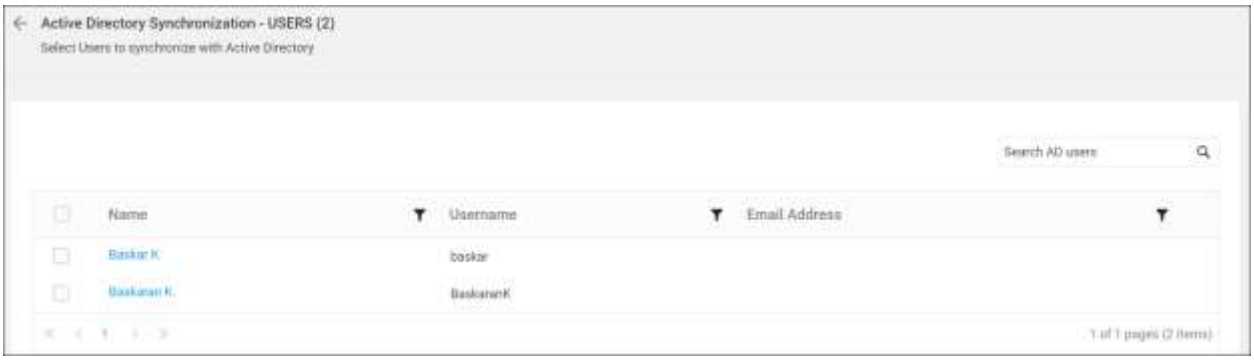
Note: Before synchronizing the Active Directory users, follow the given steps:

1. Configure Active Directory Settings
2. Import users from the Active Directory to the Syncfusion Dashboard Server by referring the following link Active Directory User Import.

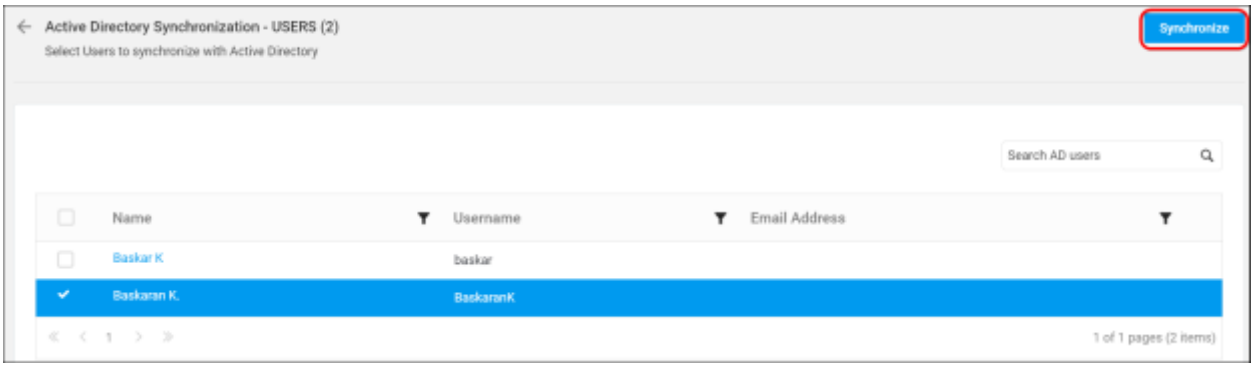
You can navigate to the user synchronization page from users page as shown in the below figure.

Synchronize Users

Dashboard Server will list the Active Directory users that are already imported as shown in the below figure.



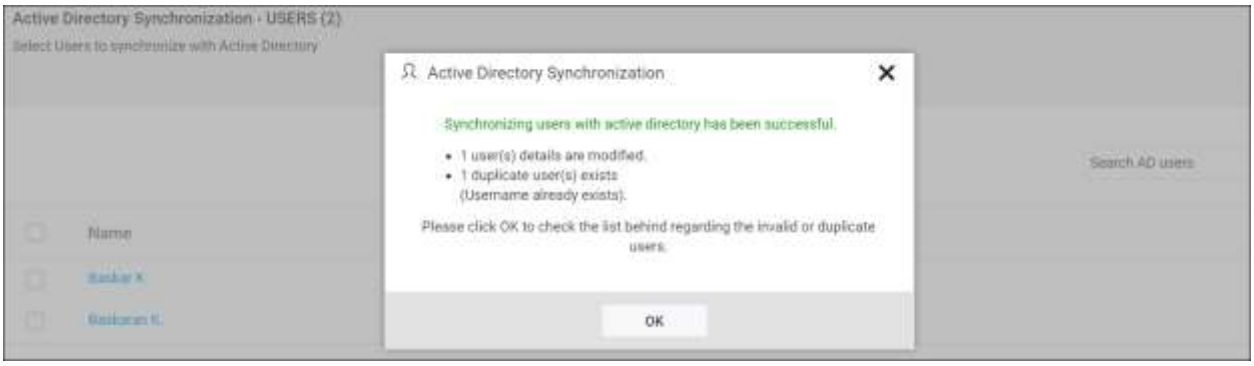
Choose the users you want to synchronize and click on Synchronize at the top.



Synchronization procedure

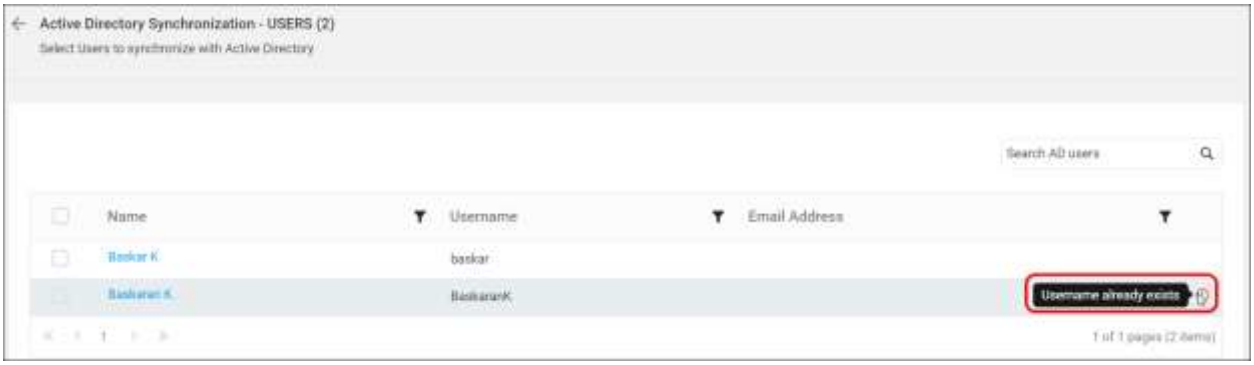
- Dashboard Server will synchronize the user details - username, first name, last name, email address, contact number with the Active Directory Server.
- Dashboard Server will delete the user if the user has deleted from the Active Directory Server.

After synchronization completes, the number of users modified, deleted, duplicated will be shown in the success message box as shown in the below figure.



Duplicate Users

Active Directory users who has the same username or email address as that of the Dashboard Server users(who are already present) will be marked as duplicate users and will not be allowed to synchronize into Active Directory.



User Import from a Database

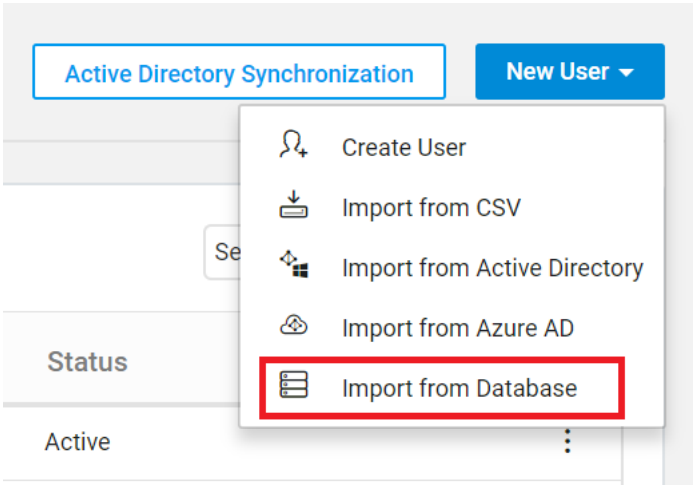
This section explains how to import users from Database into the Syncfusion Dashboard Server.

Note: Account Activation type should be E-mail Activation and E-mail settings has to be configured in the E-mail Settings in the General page for importing users from Database.

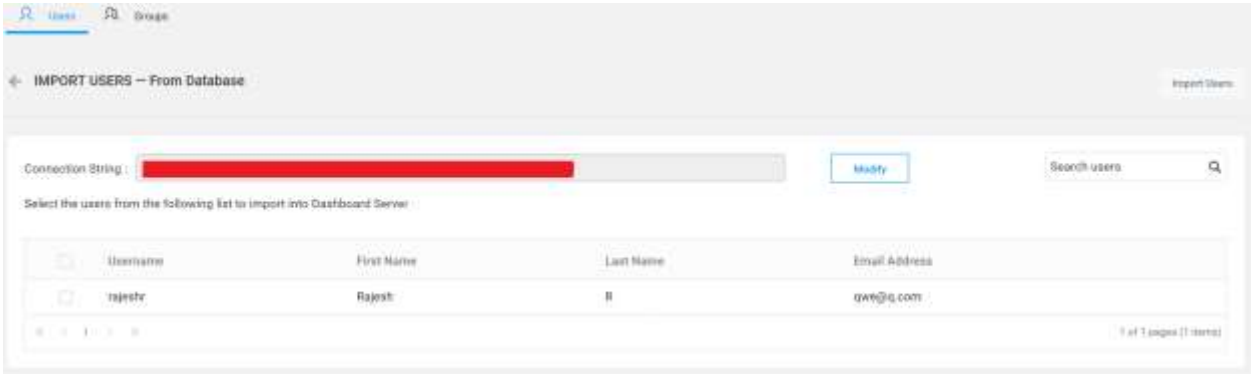
Users belonging to the System Administrator group only can import users from database into the Dashboard Server.

Listing Database Users

To add new users to the Dashboard Server, click on New User and then Import from Database from the User Management page.

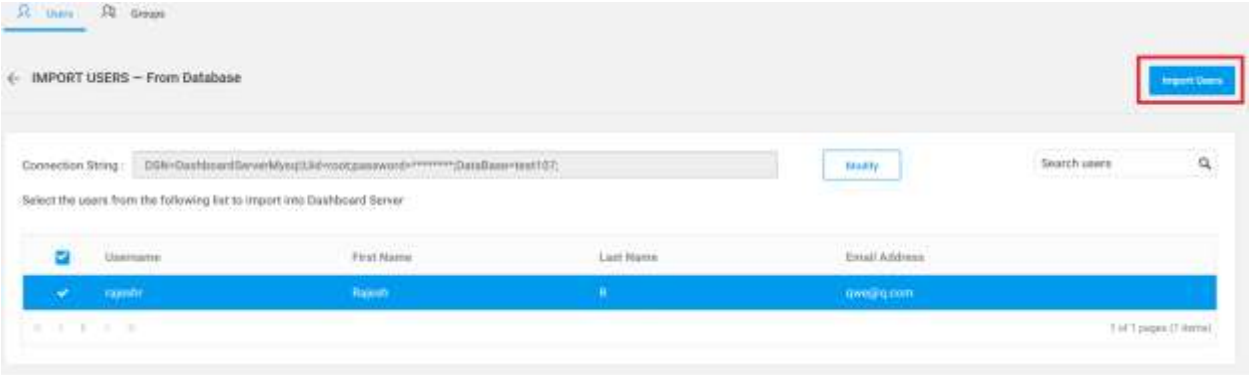


The link will redirect to another page that will look like below.

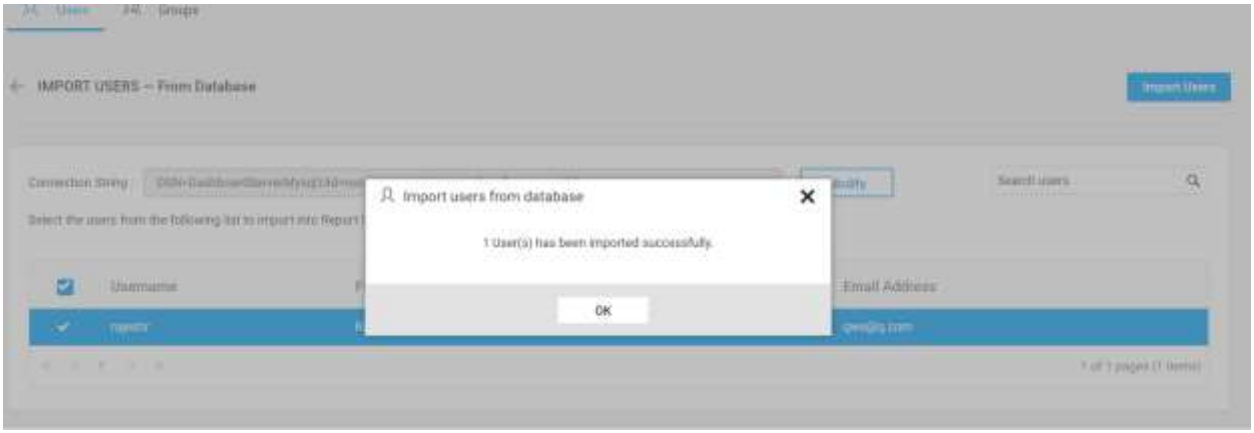


Select Users and Import

After selecting columns the data retrieved from database will be shown in Grid. Select the users to be imported and click on Import Users to import the users.



Dashboard Server will import the chosen users and a confirmation message will be displayed as shown in the below image.



Modify Existing Database Connection

To modify Existing Database configuration settings, click on the Modify link as below

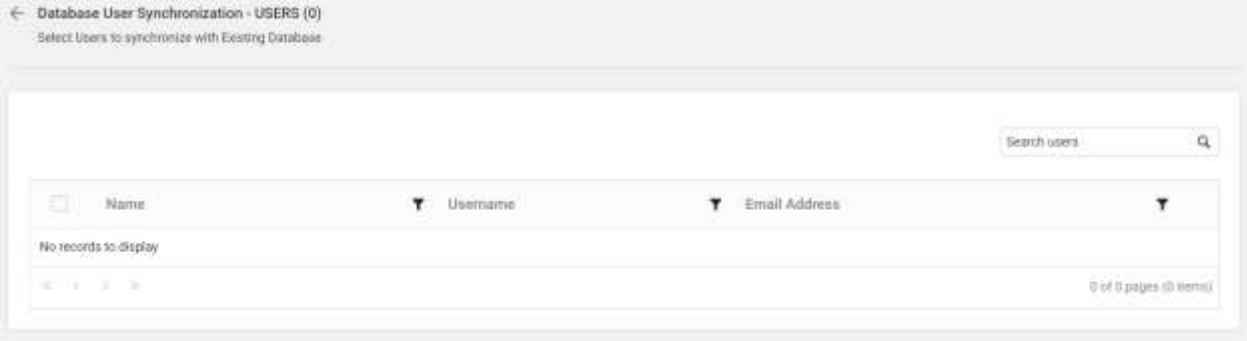
Synchronization of Imported Users From the Existing Database

This section explains how to synchronize the imported existing database users details with the Existing database.

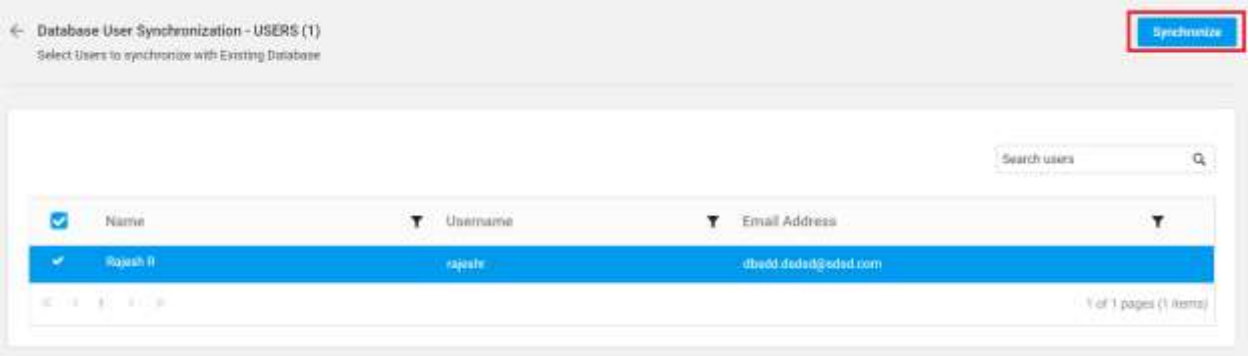
You can navigate to the user synchronization page from users page as shown in the below figure.

Synchronize Users

Dashboard Server will list the Imported Database users that are already imported as shown in the below figure.



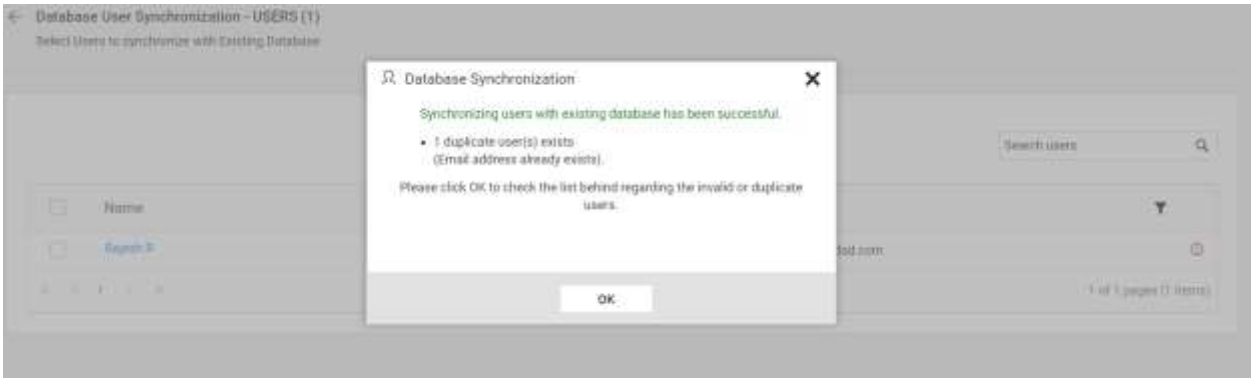
Choose the users you want to synchronize and click on Synchronize at the top.



Synchronization procedure

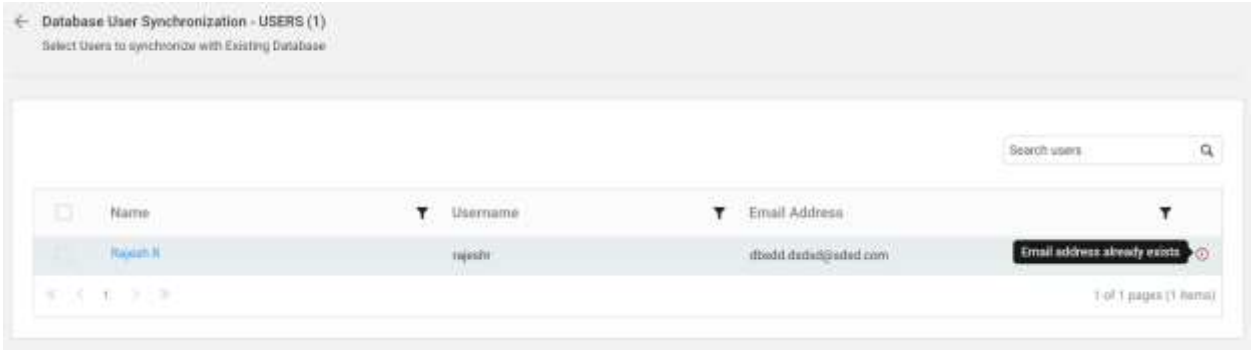
- Dashboard Server will synchronize the user details - username, first name, last name, email address, contact number with the Existing Database.
- Dashboard Server will delete the user if the user has deleted from the Existing Database.

After synchronization completes, the number of users modified, deleted, duplicated will be shown in the success message box as shown in the below figure.



Duplicate Users

Existing Database users who has the same username or email address as that of the Dashboard Server users(who are already present) will be marked as duplicate users and will not be allowed to synchronize with the imported existing database users.



Azure Active Directory User Import

This section explains how to search and import users from Azure Active Directory into the Syncfusion Dashboard Server.

Note: Azure Active Directory connection has to be configured in the Azure Active Directory Settings in the General page for importing users.

Users belonging to the System Administrator group only can import Azure Active Directory users into the Dashboard Server.

Search Users

Initially, any Active Directory users cannot be displayed until searching for the user.

You can search the Azure Active Directory users with any one of the below properties and choose them to import into the Dashboard Server.

- User name
- Email Address
- Display name

A maximum of 1000 users will be searched and pulled from Azure Active Directory in a single request.

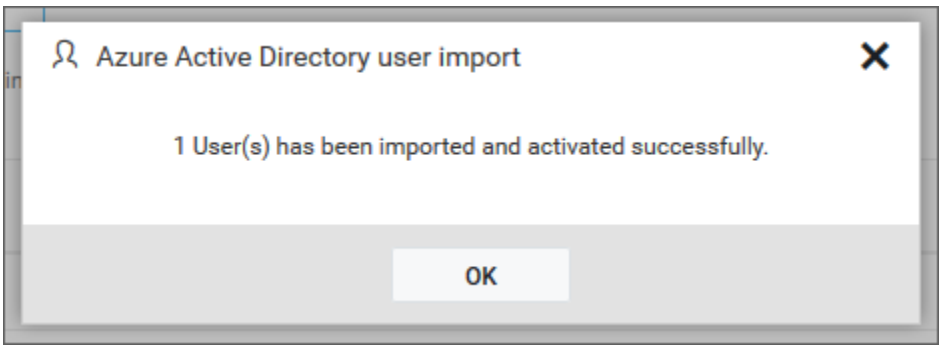
Dashboard Server will list the search results of the users in the grid as shown in the below figure.

Note: The search result will be based on "starts with" query.

Import Users

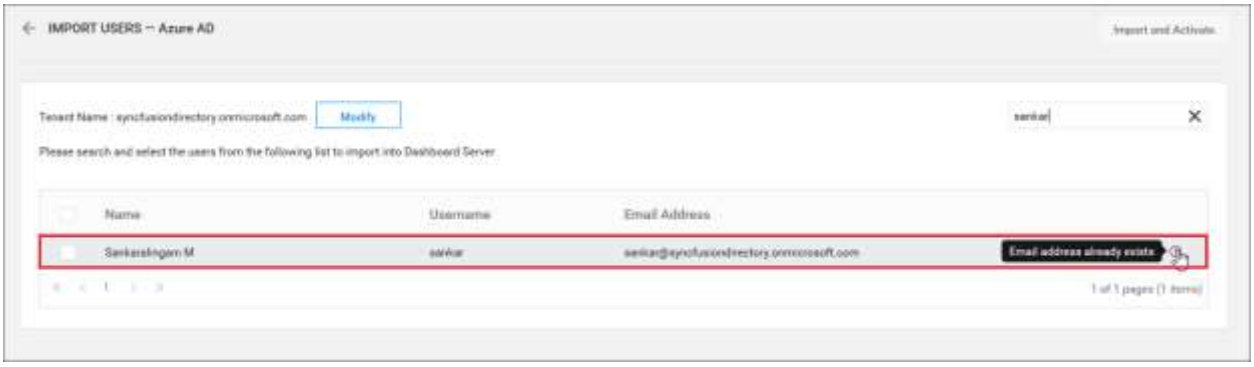
To import the Azure Active Directory users into the Dashboard Server, you have to choose the users from the list and click on the Import and Activate button at the top right corner.

Dashboard Server will import the chosen users and a confirmation message will be displayed as shown in the below figure.



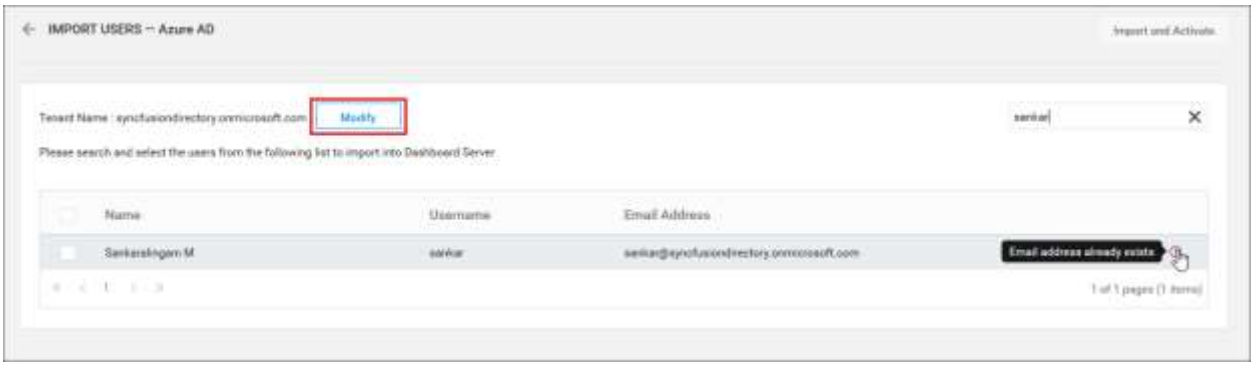
Duplicate Users

Azure Active Directory users who has the same username or email address as that of the Dashboard Server users(who are already present) will be marked as duplicate users and will not be allowed to import into Dashboard Server.



Modify Azure Active Directory Connection

To modify Azure Active Directory configuration settings, click on the Modify link as below



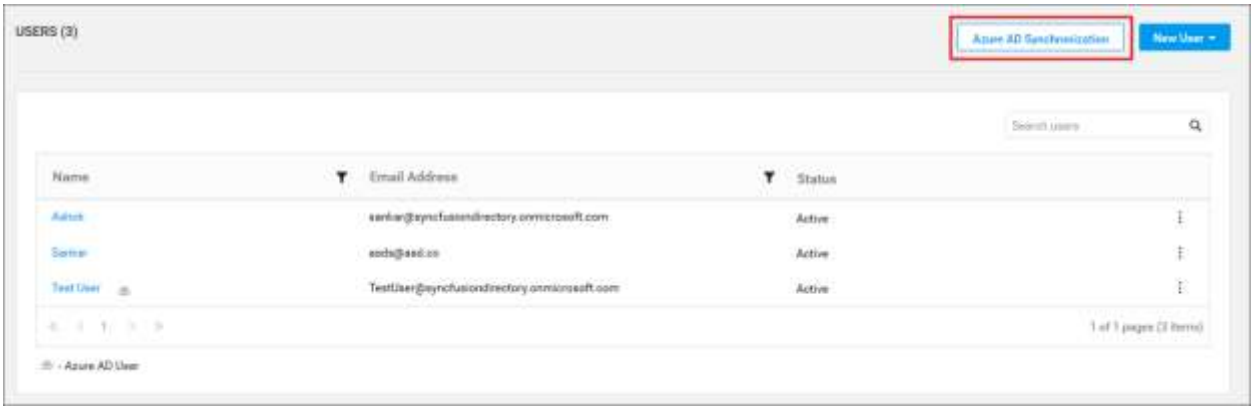
Azure Active Directory User Synchronization

This section explains how to synchronize the imported Azure Active Directory users details with the Azure Active Directory.

Note: Before synchronizing the Azure Active Directory users, follow the given steps:

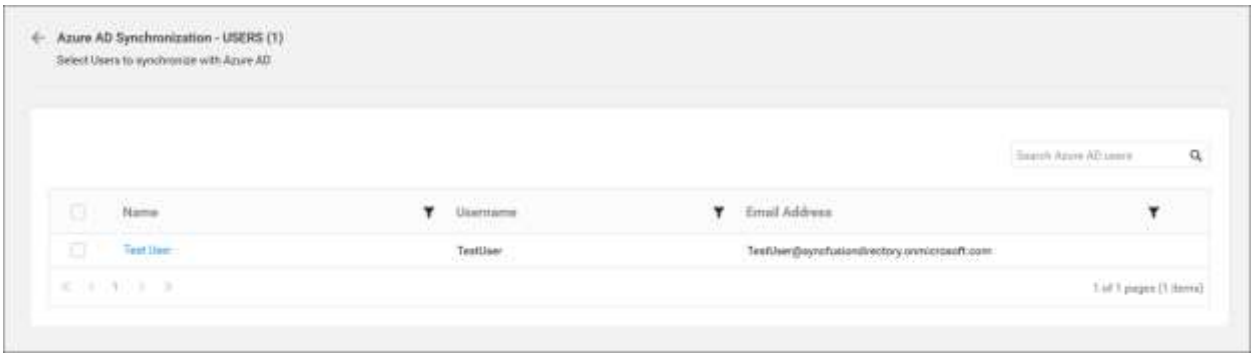
1. Configure Azure Active Directory Settings.
2. Import users from the Azure Active Directory to the Syncfusion Dashboard Server by referring the following link Active Directory Group Import.

You can navigate to the user synchronization page from users page as shown in the below figure.

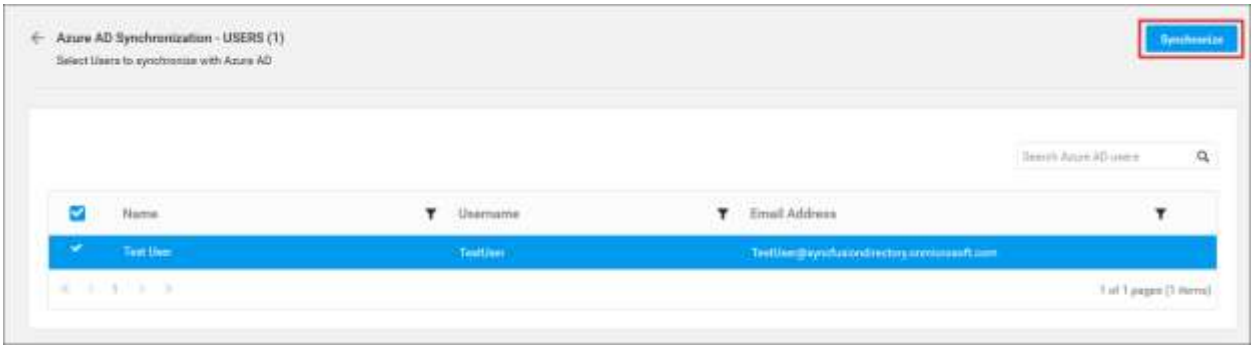


Synchronize Users

Dashboard Server will list the Azure Active Directory users that are already imported as shown in the below figure.



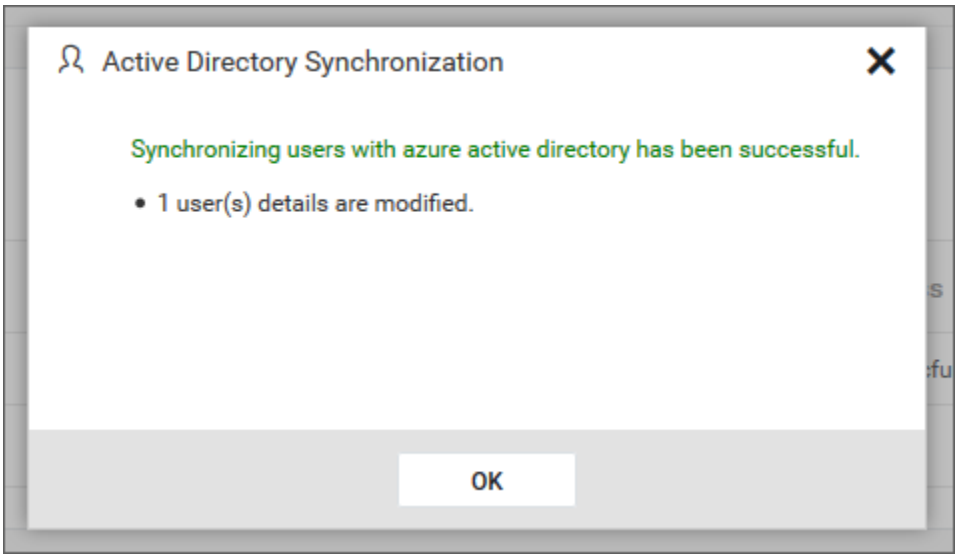
Choose the users you want to synchronize and click on Synchronize at the top.



Synchronization procedure

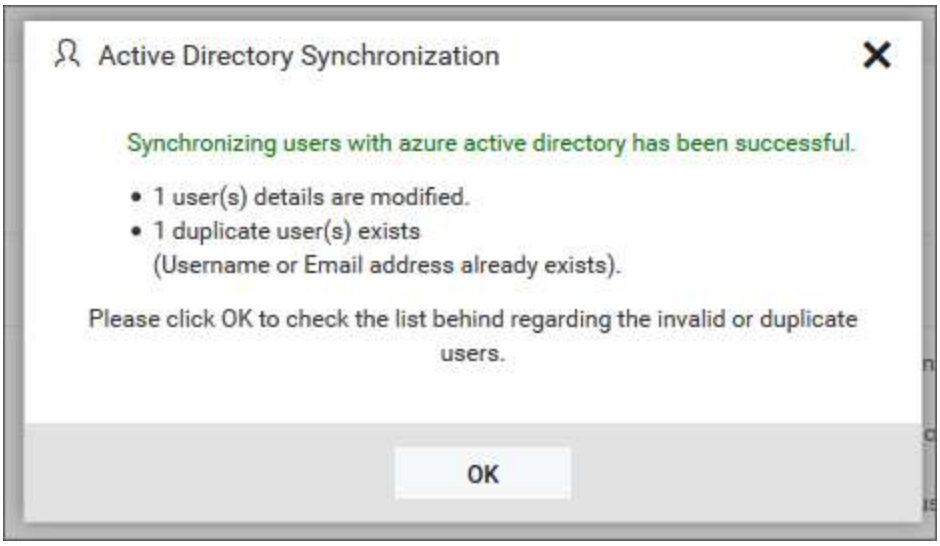
- Dashboard Server will synchronize the user details - username, first name, last name, email address, contact number with the Azure Active Directory Server.
- Dashboard Server will delete the user if the user has deleted from the Azure Active Directory Server.

After synchronization completes, the number of users modified, deleted, duplicated will be shown in the success message box as shown in the below figure.



Duplicate Users

Azure Active Directory users who has the same username or email address as that of the Dashboard Server users(who are already present) will be marked as duplicate users and will not be allowed to synchronize into Active Directory.

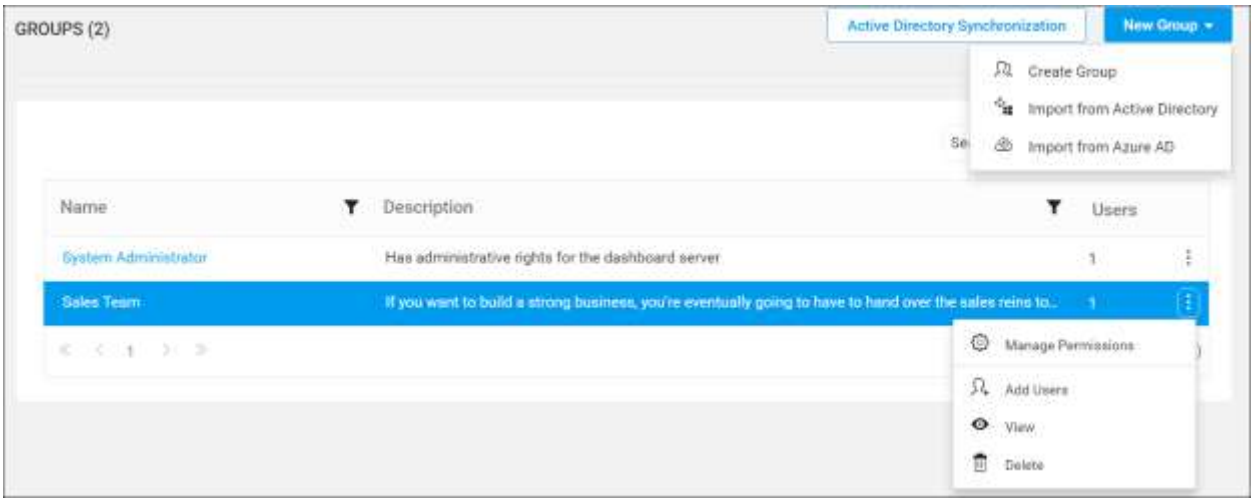


Groups

Manage Groups

This section explains on how to add, edit, delete groups and also on how to assign users and manage permissions to groups in the Syncfusion Dashboard Server.

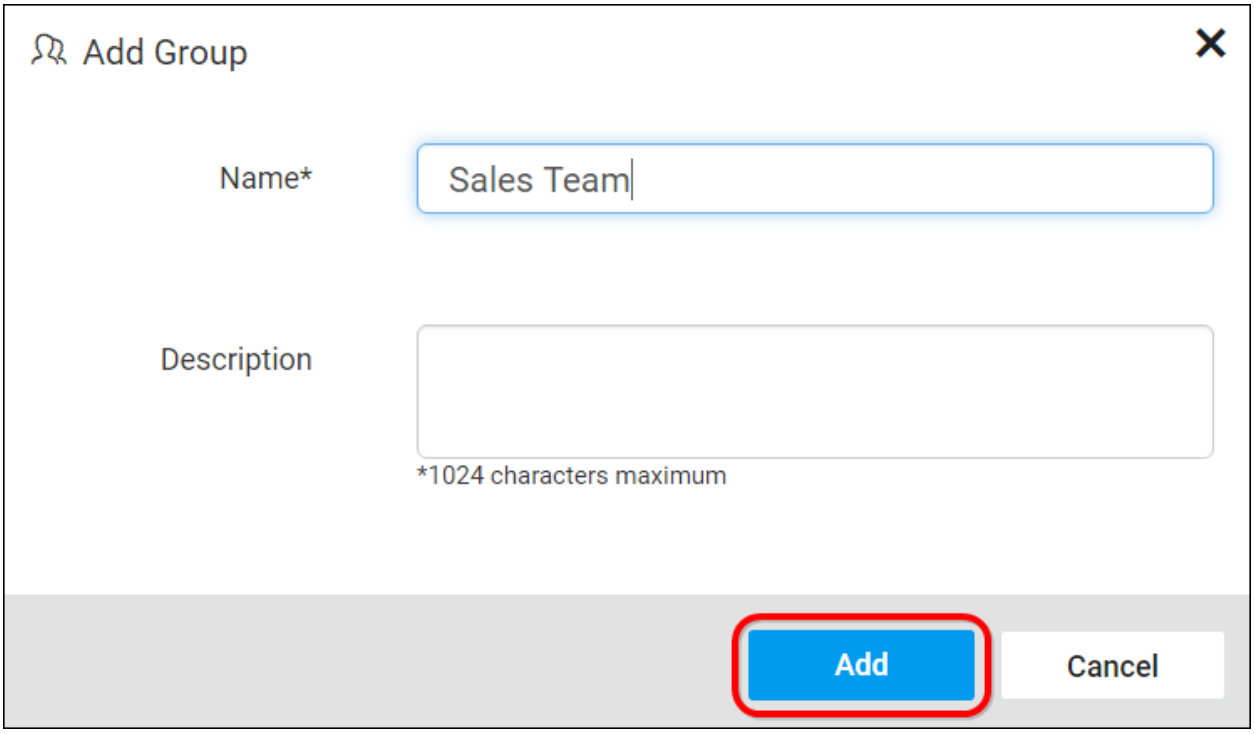
Groups is a collection of users to which permissions can be assigned.



Add new group

To add new group to the dashboard server, click on New Group in the groups management page.

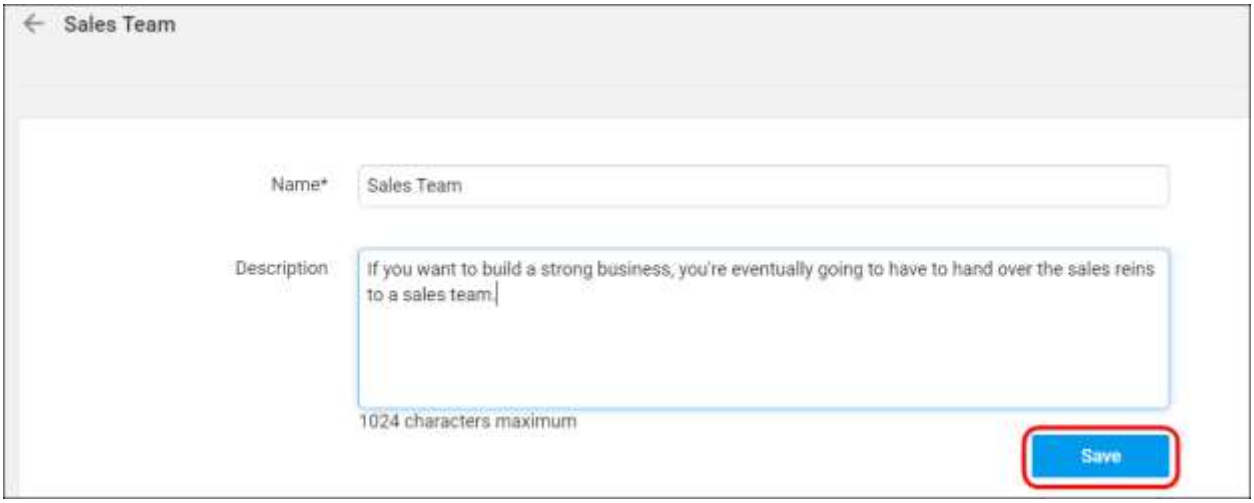
New groups can be added by providing name and description(optional) for the group.



Fill the form with name and description and click on Add. New group will be created and you can add users or manage permissions for it.

Edit group

Group Information can be edited from the group's edit page.

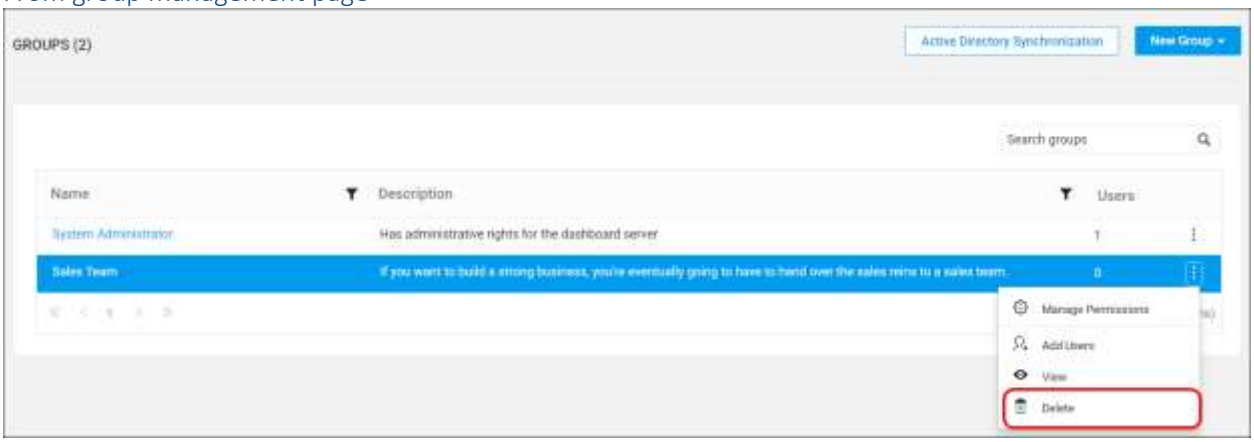


Group name and description can be edited in the group edit page. In addition to that, users can also be assigned or removed from the group in this page.

Delete group

Groups can be deleted if it is no longer needed. You cannot delete the System Administrator group.

From group management page

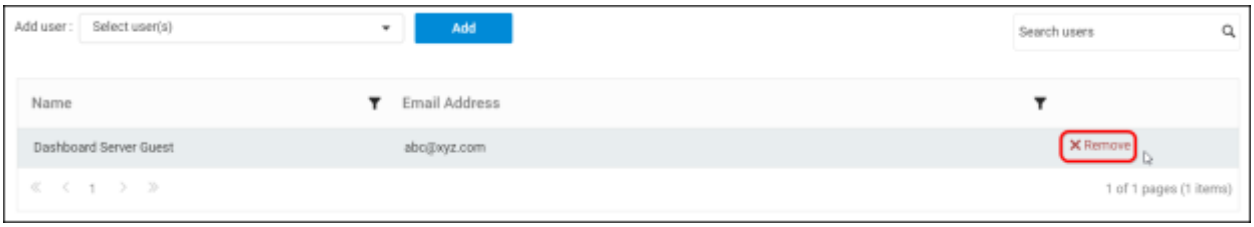


From group edit page

Assign users

Users can be assigned to the selected group there by assigning the permissions of the group to the users.

Users can also be removed from the group if the user no longer needs the permissions of the group. Click on Remove next to the user in the group edit page to remove the user from the group.



Manage permissions

Check the Manage Permissions section to learn how to manage permissions to a group.

Active Directory Group Import

This section explain on how to search and import groups from Active Directory into the Syncfusion Dashboard Server.

Note: Active Directory connection has to be configured in the Active Directory Settings in the General page for importing groups.

Users belonging to the System Administrator group only can import Active Directory groups into the Syncfusion Dashboard Server.

Search Groups

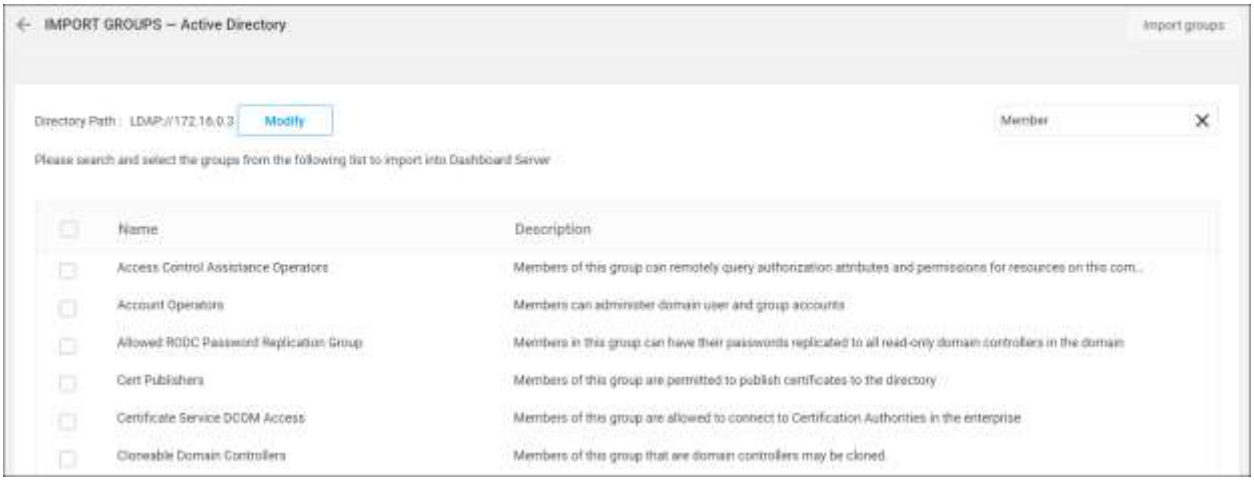
Initially, any Active Directory groups cannot be displayed until searching the group.

You can search the Active Directory groups with any one of the below properties and choose them to import into Syncfusion Dashboard Server.

- Group name
- Group description

A maximum of 1000 groups will be searched and pulled from Active Directory in a single request.

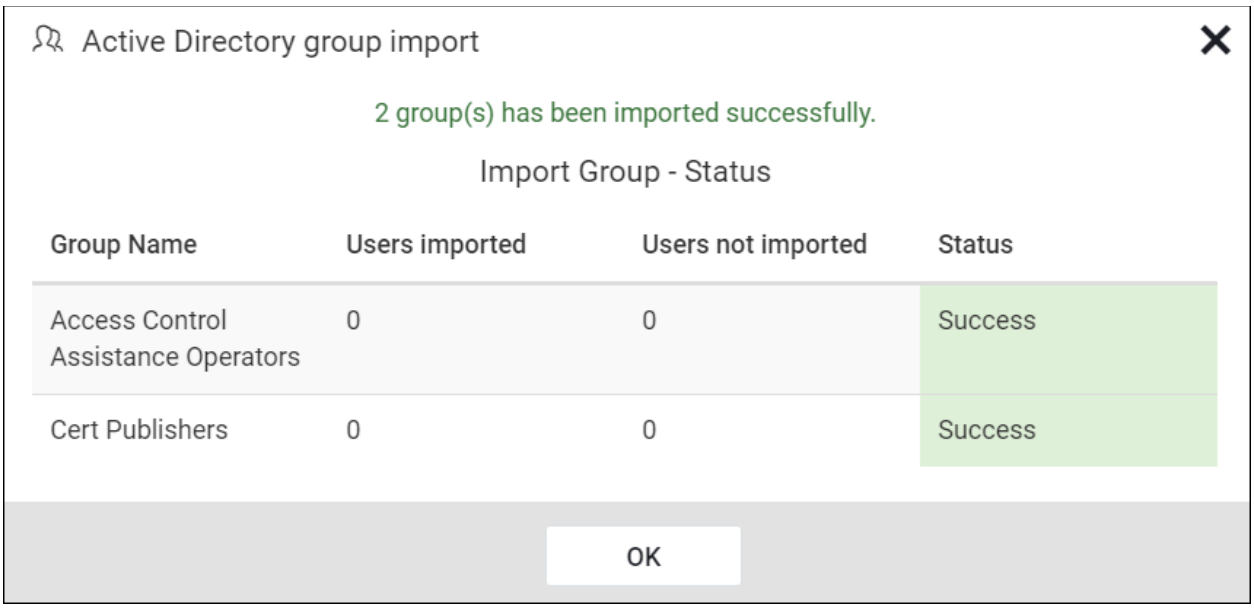
Dashboard Server will list the search results of the groups in the grid as shown in the below figure.



Import Groups

To import the Active Directory groups into the Dashboard Server, you have to choose the groups from the list and click on the Import groups button at the top right corner.

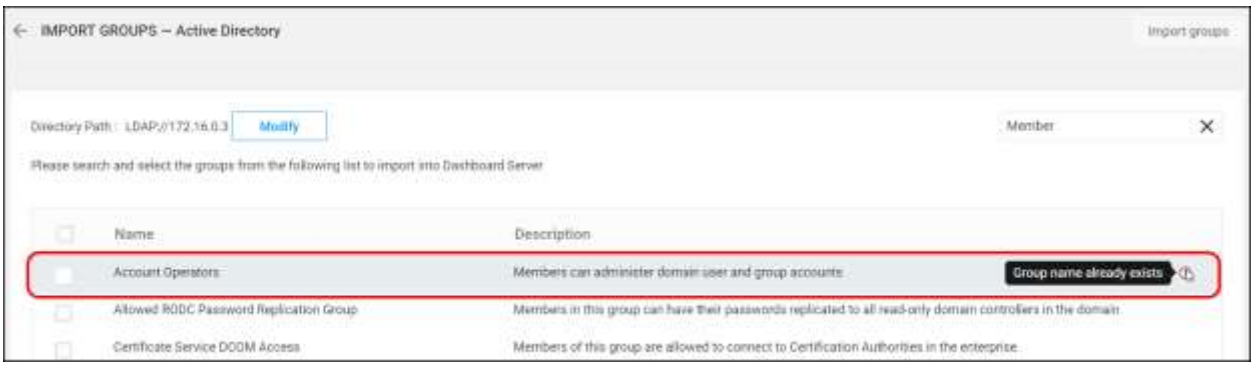
Dashboard Server will import the chosen groups and a confirmation message will be displayed as shown in the below figure.



The success message box explains the users who all are get imported/not imported into the Dashboard server.

Duplicate Groups

Active Directory groups who has the same groupname as that of the Dashboard Server groups(which are already present) will be marked as duplicate groups and will not be allowed to import into Dashboard Server.



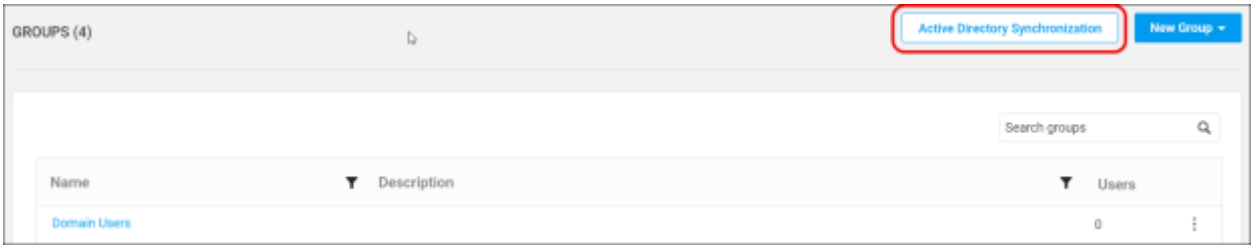
Active Directory Group Synchronization

This section explains how to synchronize the imported Active Directory group and its users with the Active Directory.

Note: Before synchronizing the Active Directory groups, follow the given steps:

1. Configure Active Directory Settings
2. Import groups from the Active Directory to the Syncfusion Dashboard Server by referring the following link Active Directory Group Import.

You can navigate to the group synchronization page from groups page as shown in the below figure.



Synchronize Groups

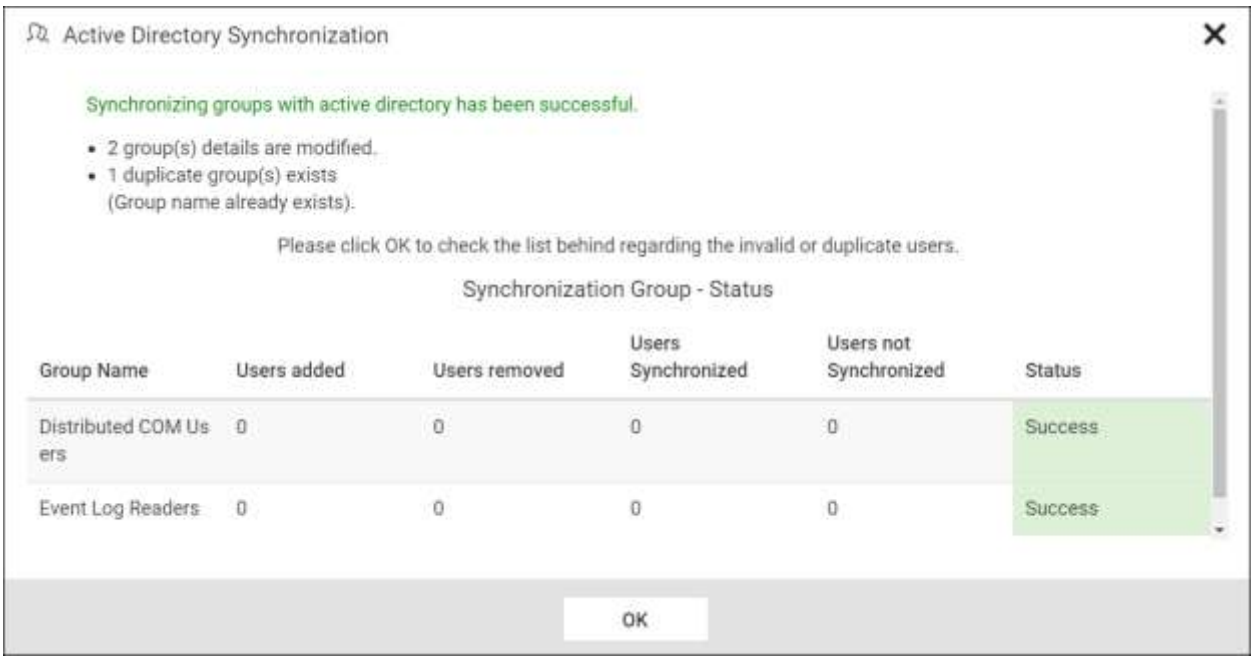
Dashboard Server will list the Active Directory groups that are already imported as shown in the below figure.

Choose the groups you want to synchronize and click on Synchronize at the top.

Synchronization procedure

- Dashboard Server will update the group's name and description from the Active Directory Server.
- Dashboard Server will delete the groups if the group has been deleted from the Active Directory Server.
- Dashboard Server will delete the user from Dashboard Server group, if the user has been deleted from Active Directory Server group. Dashboard Server will add the user into Dashboard Server, if a new user is added into the Active Directory group. If the new user is not present in the Dashboard Server, then a new user account will be created in the Dashboard Server and will be added into the group.

After synchronization completes, the number of groups modified, deleted, duplicated will be shown in the success message box as shown in the below figure.



Duplicate Groups

Active Directory groups who has the same group name as that of the Dashboard Server groups(which are already present) will be marked as duplicate groups and will not be allowed to synchronize with Active Directory.

Azure Active Directory Group Import

This section explain on how to search and import groups from Azure Active Directory into the Syncfusion Dashboard Server.

Note: Azure Active Directory connection has to be configured in the Azure Active Directory Settings in the General page for importing groups.

Users belonging to the System Administrator group only can import Azure Active Directory groups into the Syncfusion Dashboard Server.

Search Groups

Initially, any Active Directory groups cannot be displayed until searching for the group.

You can search the Azure Active Directory groups with any one of the below properties and choose them to import into Syncfusion Dashboard Server.

- Group name

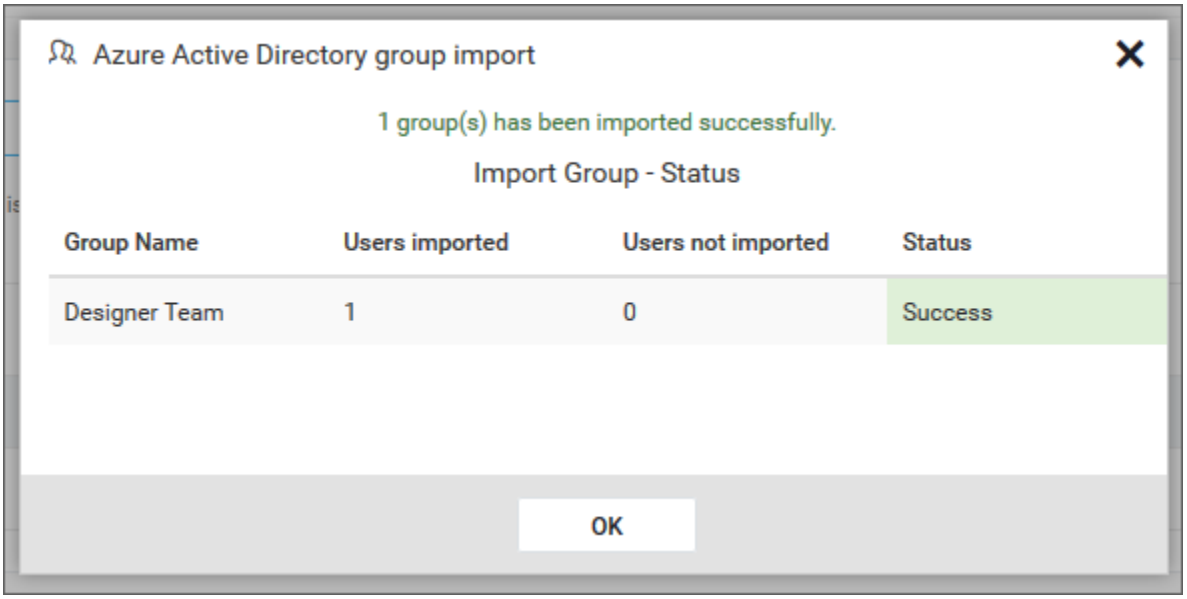
A maximum of 1000 groups will be searched and pulled from Azure Active Directory in a single request.

Dashboard Server will list the search results of the groups in the grid as shown in the below figure.

Import Groups

To import the Azure Active Directory groups into the Dashboard Server, you have to choose the groups from the list and click on the Import groups button at the top right corner.

Dashboard Server will import the chosen groups and a confirmation message will be displayed as shown in the below figure.



The success message box explains the users who all are get imported/not imported into the Dashboard server.

Duplicate Groups

Azure Active Directory groups who has the same groupname as that of the Dashboard Server groups(which are already present) will be marked as duplicate groups and will not be allowed to import into Dashboard Server.

Azure Active Directory Group Synchronization

This section explains how to synchronize the imported Azure Active Directory group and its users with the Azure Active Directory.

Note: Before synchronizing the Azure Active Directory groups, follow the given steps:

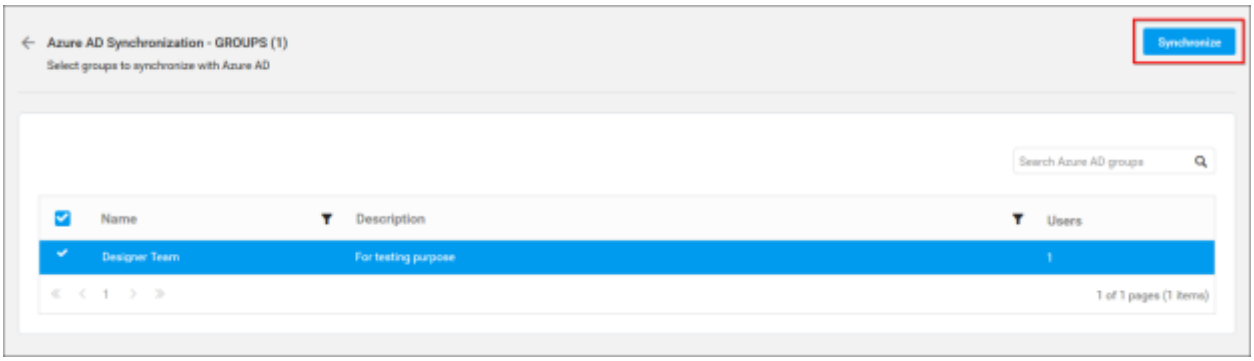
1. Configure Azure Active Directory Settings
2. Import groups from the Azure Active Directory to the Syncfusion Dashboard Server by referring the following link Active Directory Group Import.

You can navigate to the group synchronization page from groups page as shown in the below figure.

Synchronize Groups

Dashboard Server will list the Azure Active Directory groups that are already imported as shown in the below figure.

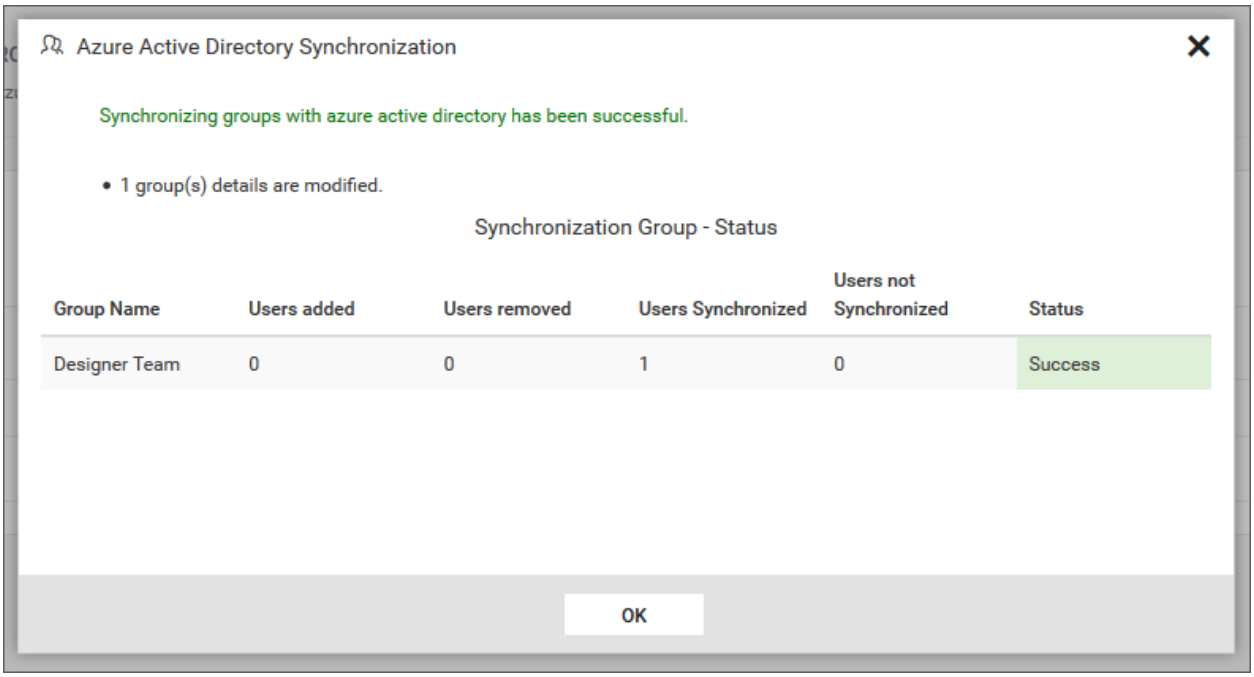
Choose the groups you want to synchronize and click on Synchronize at the top.



Synchronization procedure

- Dashboard Server will update the group's name and description from the Azure Active Directory Server.
- Dashboard Server will delete the groups if the group has been deleted from the Azure Active Directory Server.
- Dashboard Server will delete the user from Dashboard Server group, if the user has been deleted from Azure Active Directory Server group. Dashboard Server will add the user into Dashboard Server, if a new user is added into the Azure Active Directory group. If the new user is not present in the Dashboard Server, then a new user account will be created in the Dashboard Server and will be added into the group.

After synchronization completes, the number of groups modified, deleted, duplicated will be shown in the success message box as shown in the below figure.



Duplicate Groups

Azure Active Directory groups who has the same group name as that of the Dashboard Server groups(which are already present) will be marked as duplicate groups and will not be allowed to synchronize with Azure Active Directory.

Manage Permissions

This section explains the access modes, entities & scopes and how to manage the permissions for the users and groups.

Permissions can only be managed by the users belonging to the System administrator group.

Permission can be directly added to both users and groups. Permissions are classified in the following structure.

Access Mode – Entity – Scope

Access Modes

- Read – Provides read permission for the chosen entity.
- Read and Write – Provides read and write permission for the chosen entity.
- Read, Write and Delete – Provides read, write and delete permission for the chosen entity.
- Read, Download – Provides read and download permission for the chosen entity.
- Read, Write and Download – Provides read, write and download permission for the chosen entity.
- Read, Write, Delete and Download – Provides read, write, delete and download permission for the chosen entity.
- Create – Provides permission to create the chosen entity.

Entity

- All Dashboards – Provides permission to access all dashboards with the chosen access mode.
- Dashboards in Category – Provides permission to access dashboards in a specific category with chosen access mode.
- Specific Dashboard – Provides permission to access a specific dashboard with the chosen access mode.
- All Data Sources – Provides permission to access all data sources with the chosen access mode.
- Specific Data Source – Provides permission to access a specific data source with the chosen access mode.
- All Categories – Provides permission to access all categories with the chosen access mode.
- Specific Category – Provides permission to access a specific category with the chosen access mode.
- All Schedules – Provides permission to access all schedules with the chosen access mode.
- Specific Schedule – Provides permission to access a specific schedule with the chosen access mode.

Scope

Scopes can be chosen for the below entities only. Other entities do not require to specify the scope.

- Dashboards in Category – A specific category has to be chosen to provide access to the dashboards in that category.
- Specific Dashboard – A specific dashboard has to be chosen to provide access to it.
- Specific Data Source – A specific data source has to be chosen to provide access to it.
- Specific Category – A specific category has to be chosen to provide access to it.
- Specific Schedule – A specific schedule has to be chosen to provide access to it.

Note: Create access can only have the scopes, All Dashboards, Dashboards in Category, All Data Sources, All Schedules and All Categories.

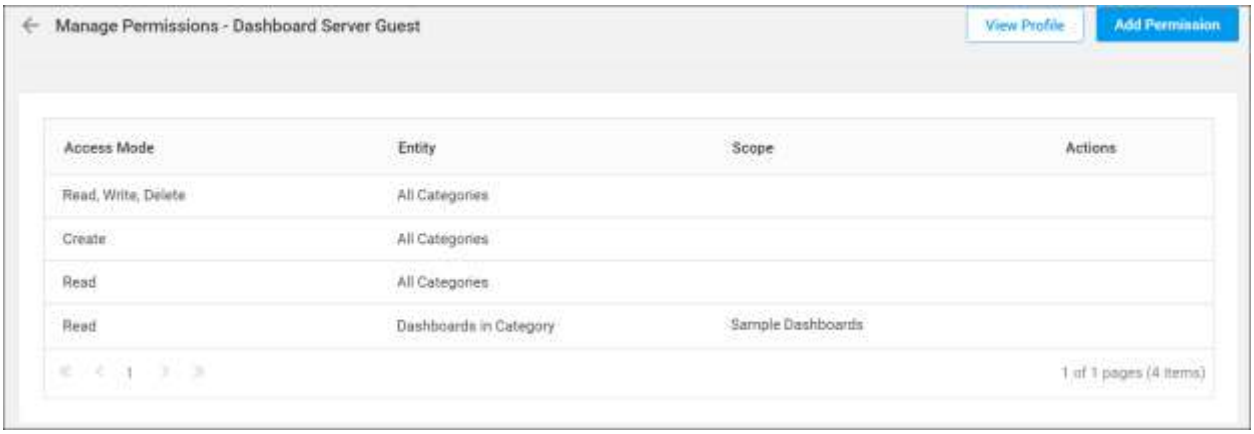
Manage Permissions - users

Manage Permissions page for the user can be accessed from any one of the following places.

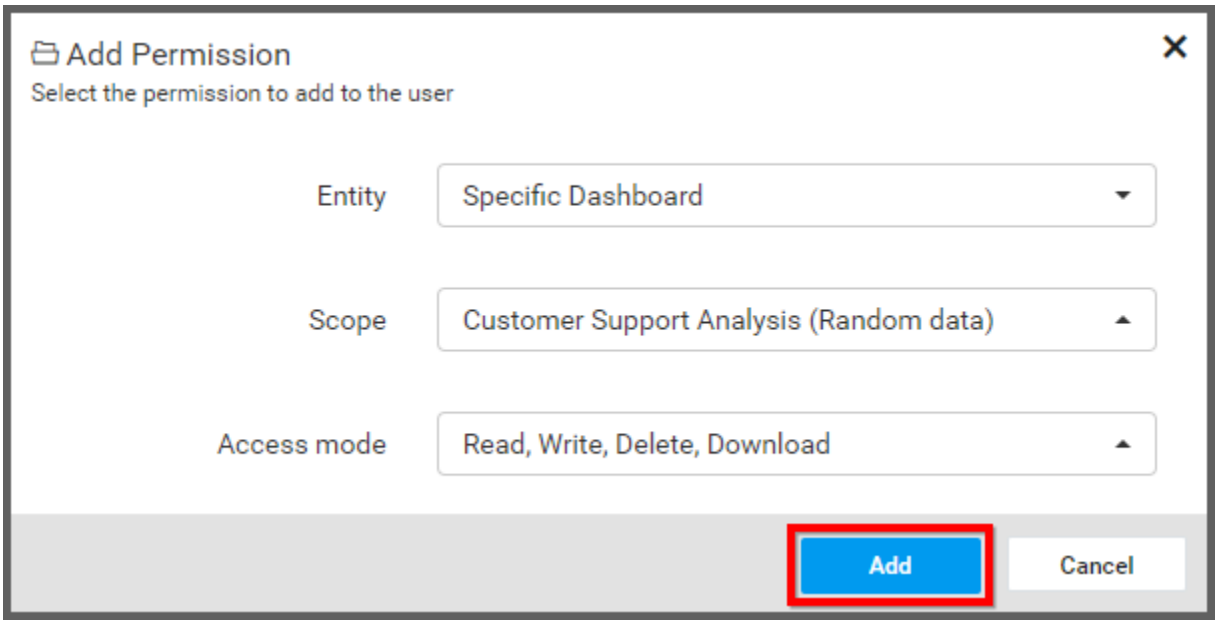
1. Context menu of the respective user in the users grid on the user management page

2. On top right corner of the user profile edit page

Here you will find both the permissions assigned directly to the user and the permissions that the user got inherited from the groups assigned with.



Click on Add Permission to add permissions to the user. Add Permission dialog box is shown below.



Steps to add permission to the user

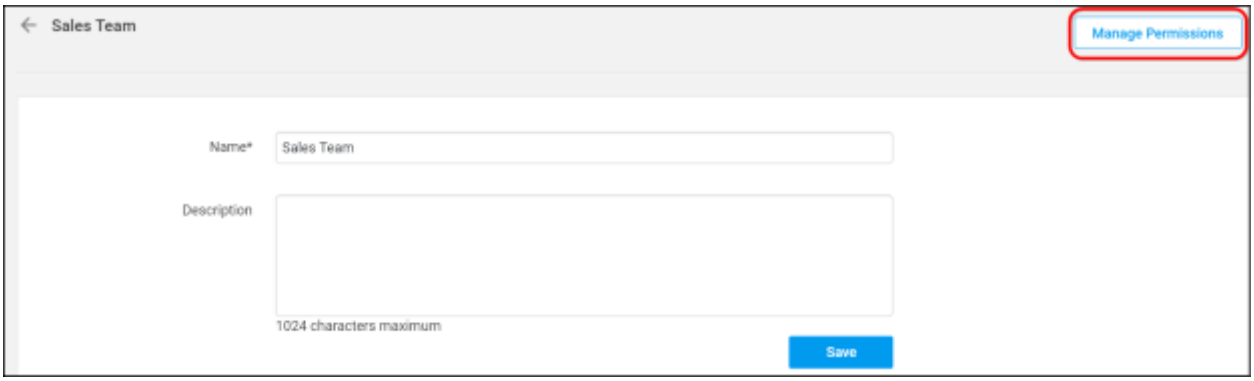
1. Select the entity.
2. Select the scope if the entity is not All item type.
3. Select the access mode.
4. Click on Add to add the framed permission to the user.

Manage Permissions - groups

Manage Permissions page for the group can be accessed from any one of the following places.

1. Context menu of the respective group in the groups grid on the group management page

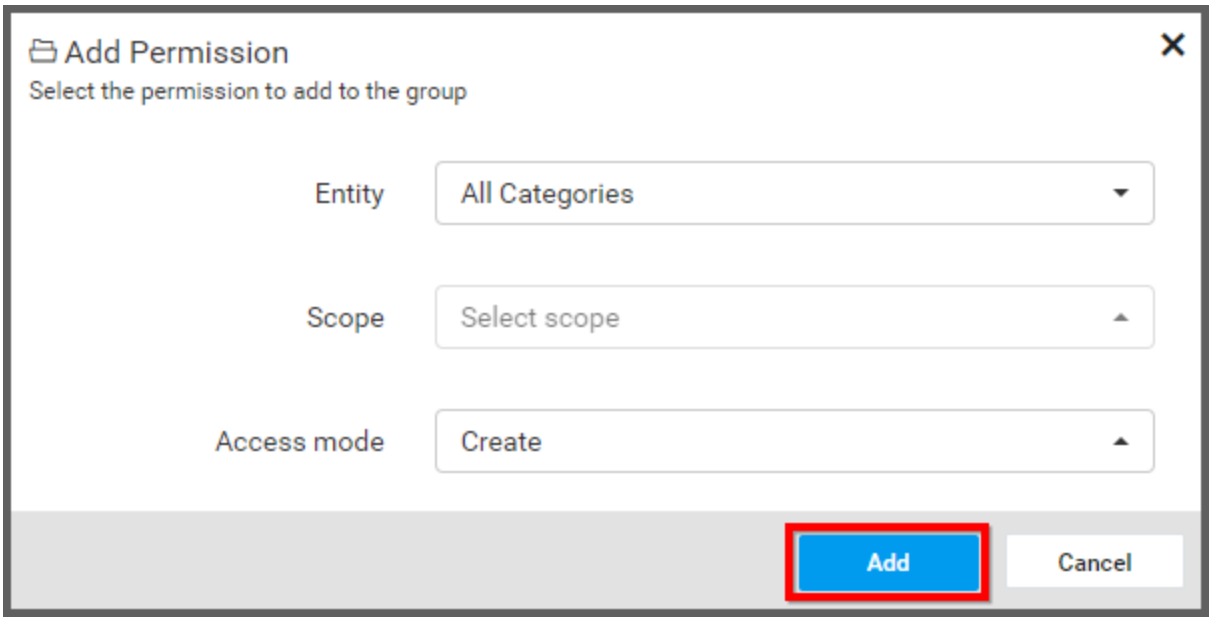
2. On top right corner of the group edit page



Here you will find the permissions assigned directly to the group.

Please refer the below screenshot for the Manage Permissions for the user page.

Click on Add Permission to add permissions to the group. Add Permission dialog box is shown below.



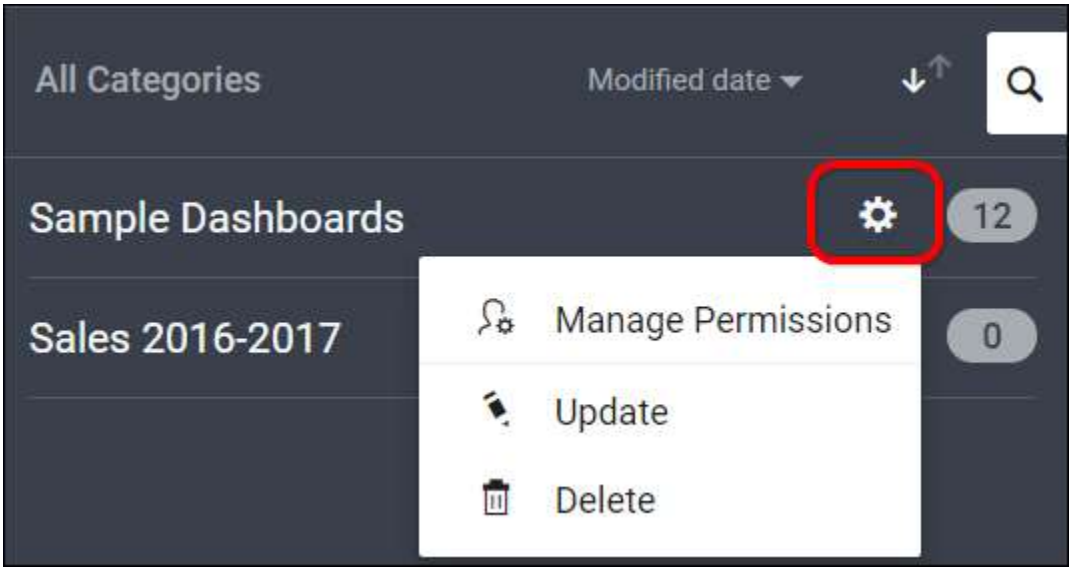
Steps to add permission to the group

1. Select the entity.
2. Select the scope if the entity is not All item type.
3. Select the access mode.
4. Click on Add to add the framed permission to the group.

Manage Categories

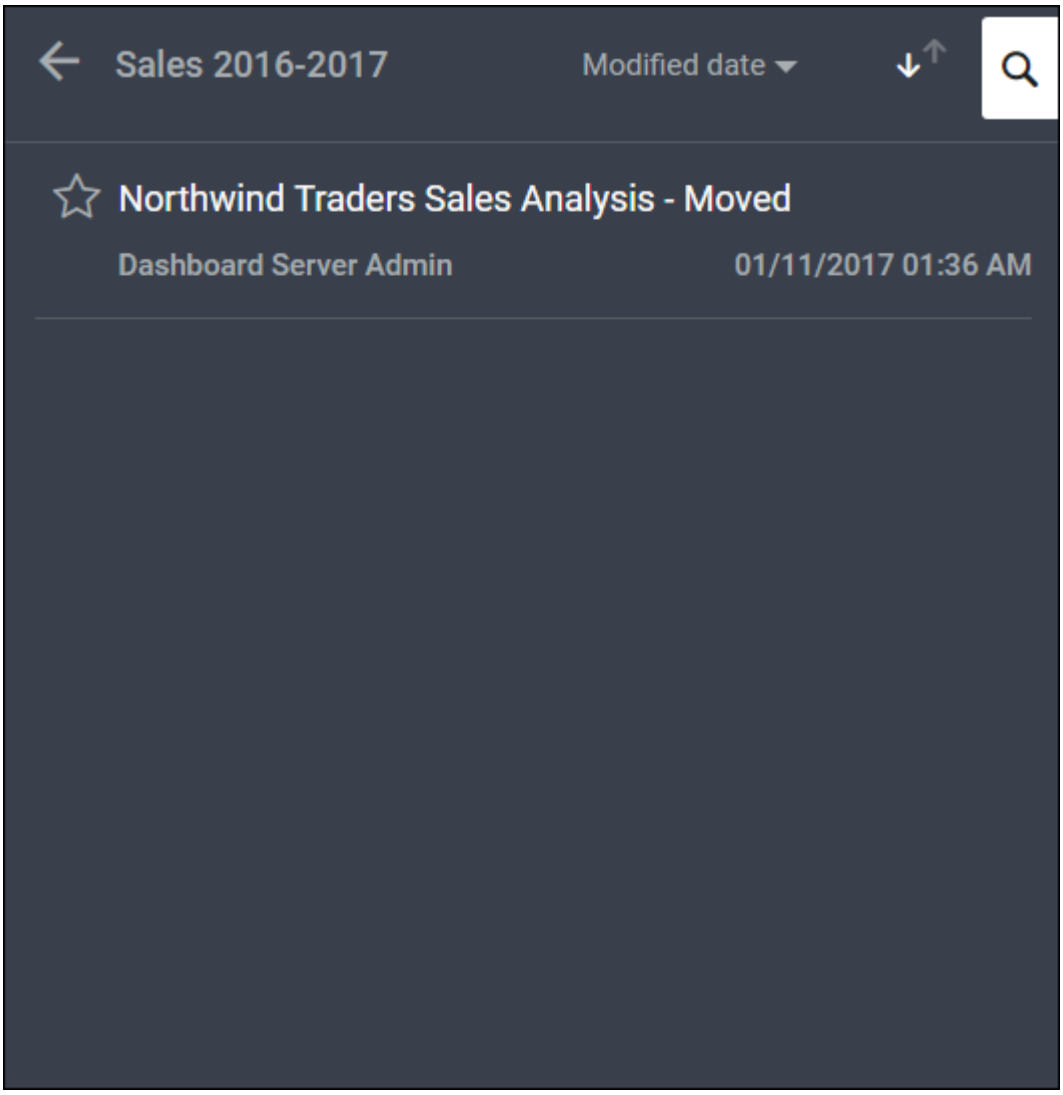
This section explains on how to open, add, update, share and delete categories in the Syncfusion Dashboard Server.

Categories are used to group and manage the dashboards. Categories that are accessible by the user depends upon the user's permission and the categories whose dashboards the user has access are displayed in the left panel in the dashboards page.



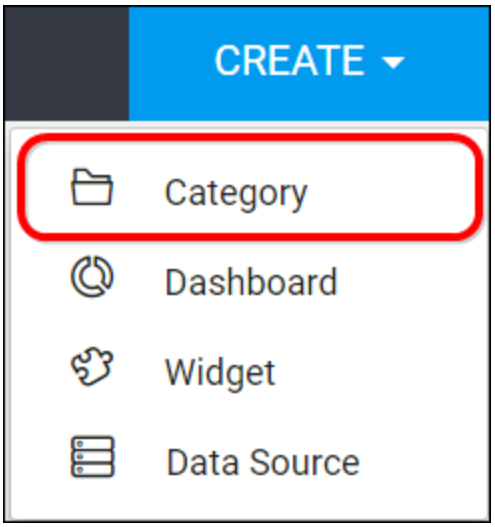
Open Category

Click on any category in the left panel to view the dashboards grouped with it.

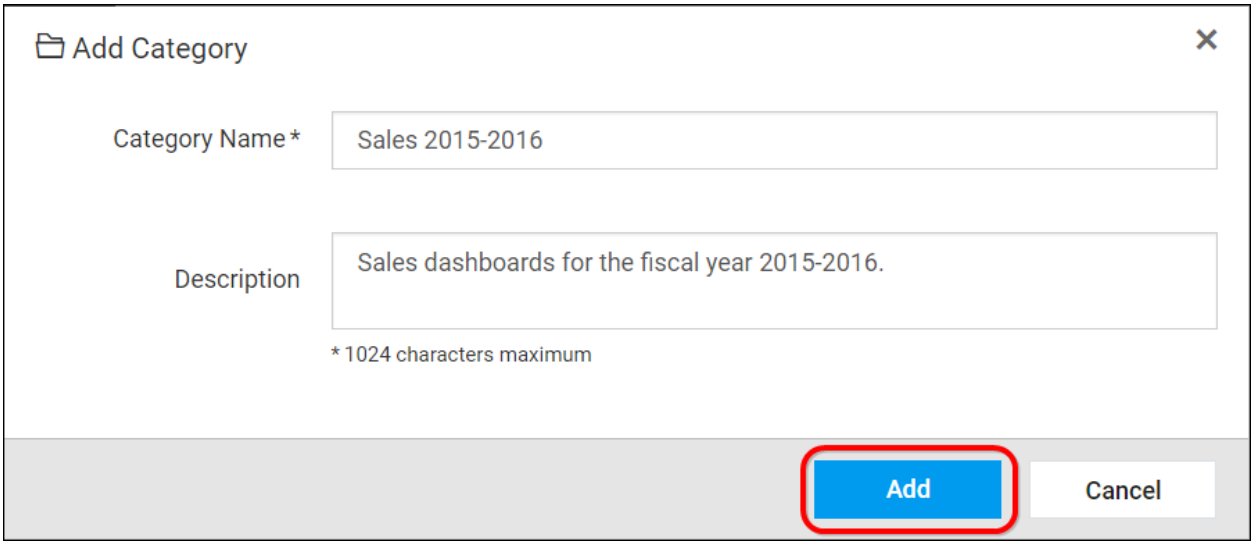


Add Category

To add a new category you must have Create All Categories permission. Click on the Create button in the menu and select Category to create a category.



New categories can be added by providing name and description(optional) for the category.

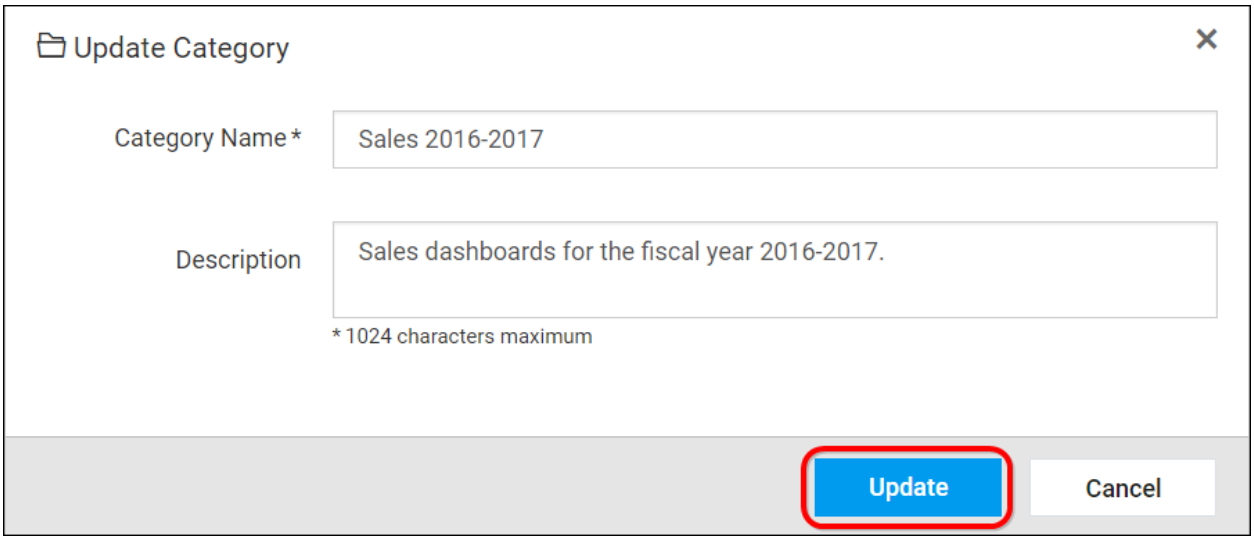


Fill the form with name, description and click on Add.

Note: Read Write Delete permission for that Specific Category will be added for the user who created the category.

Update Category

Category can be updated from the context menu with its name and description.

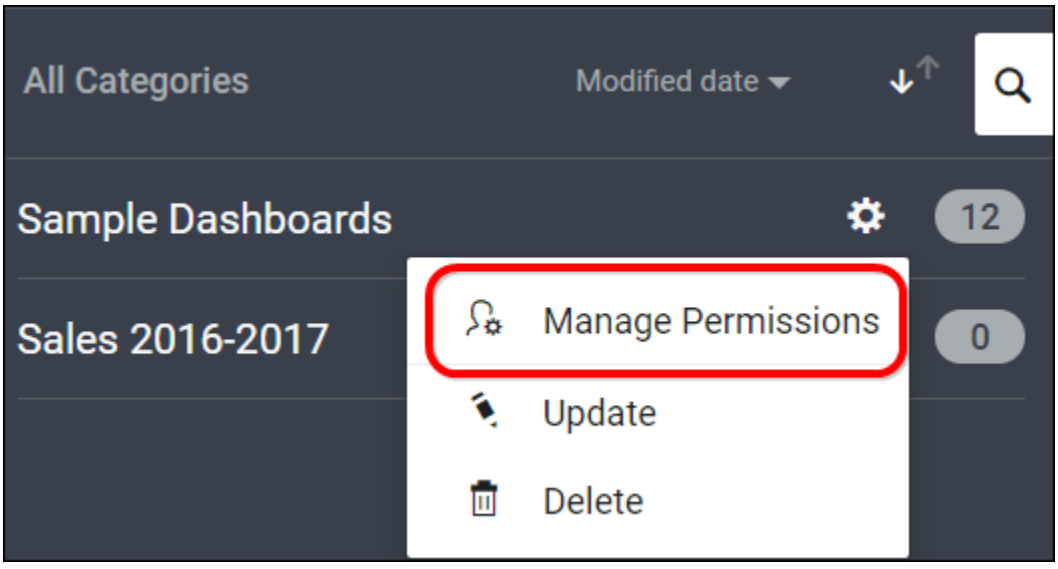


Share Category

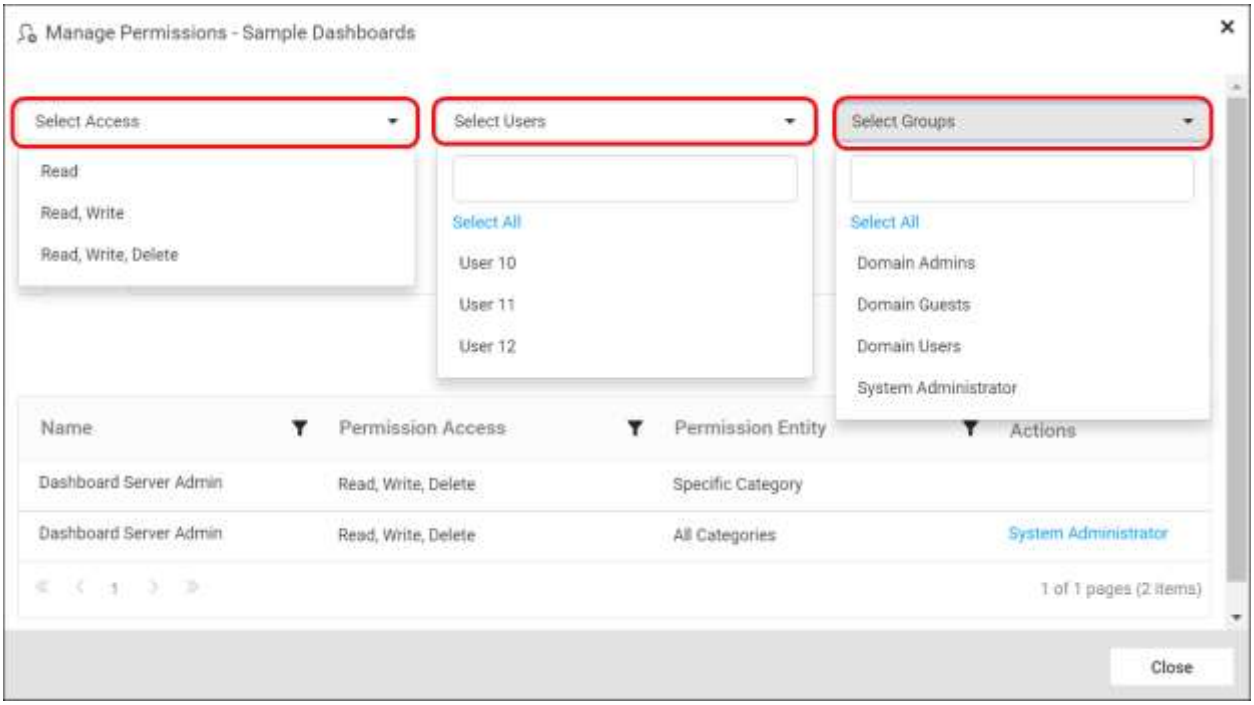
This section explains on how to share categories with the other users in the Dashboard Server.

Steps to share a category

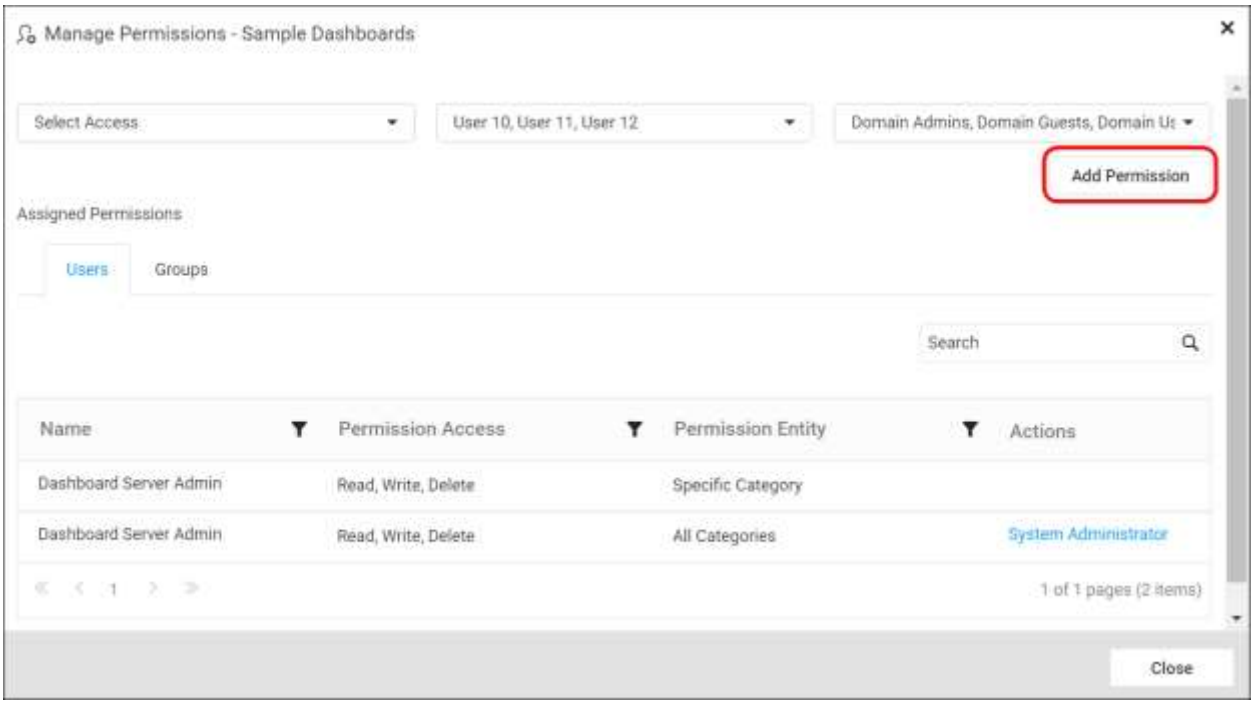
1. Click the Actions button in the category list context menu and select Manage Permissions option.



2. Select the permission access from the Select Access dropdown and select the users or groups to share the category.



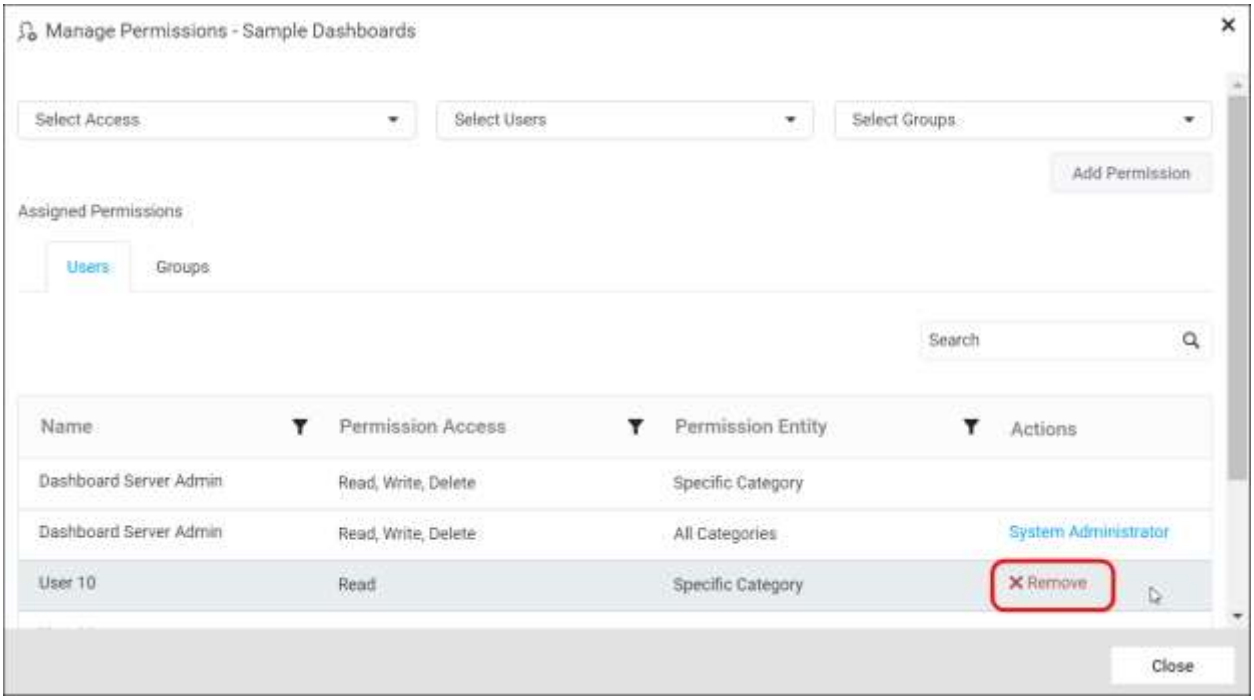
3. After selecting the access and users or groups, click on the Add Permission button.



Note: Only the user who created the category and the Administrator can share the category with other Dashboard Server users.

Remove Permission

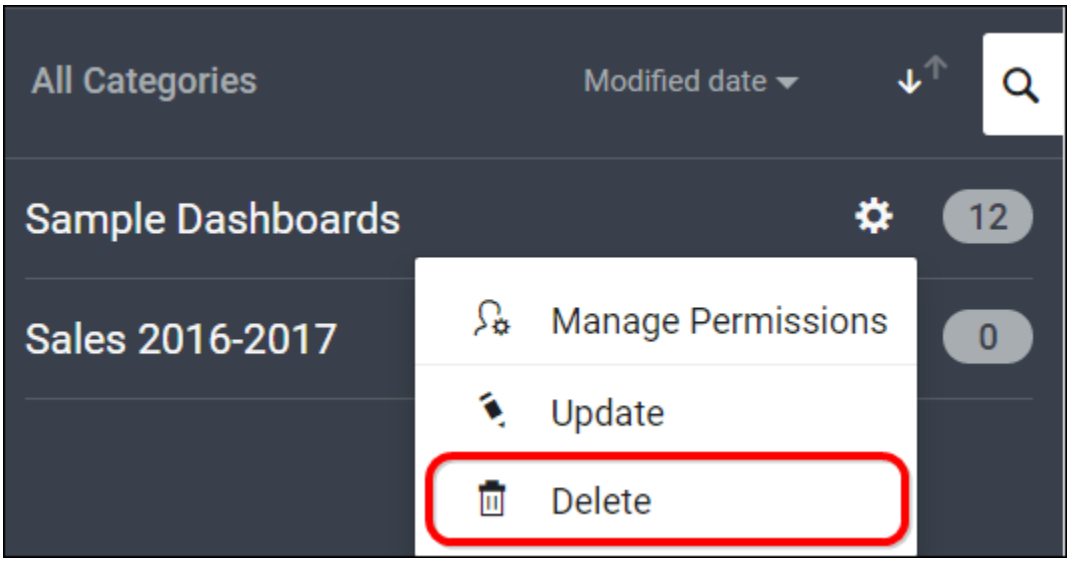
The user who created the category and the Administrator can remove the shared category permissions using the Remove option in the Actions column of the each permissions.



Delete Category

Category can also be deleted from the Dashboard Server when they are no longer required.

Click on delete in the context menu for the category to be deleted.



Note: Category cannot be deleted when it has dashboards grouped in it.

Manage Dashboards

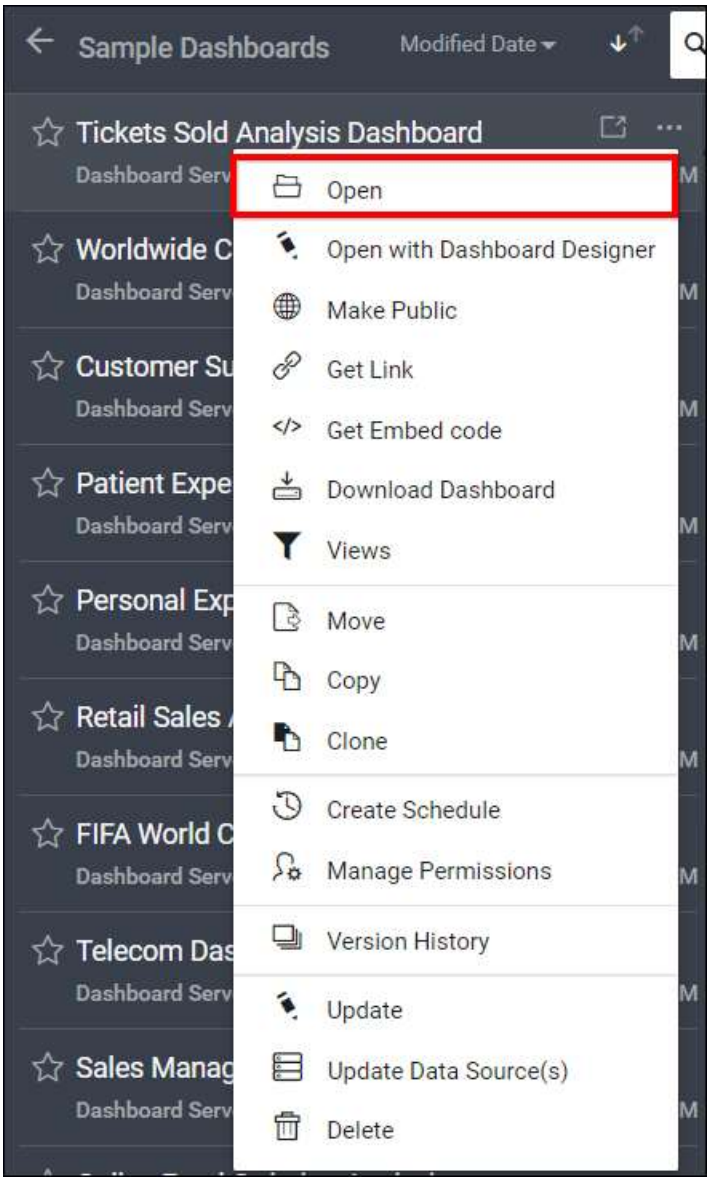
How to open Dashboards

This section explains on how to open Dashboards in the Dashboard Server.

Dashboards that are accessible by the user depending on the user's permission is displayed in the Dashboards page.

Open Dashboard

Click on the Dashboard Name in the list to open it.

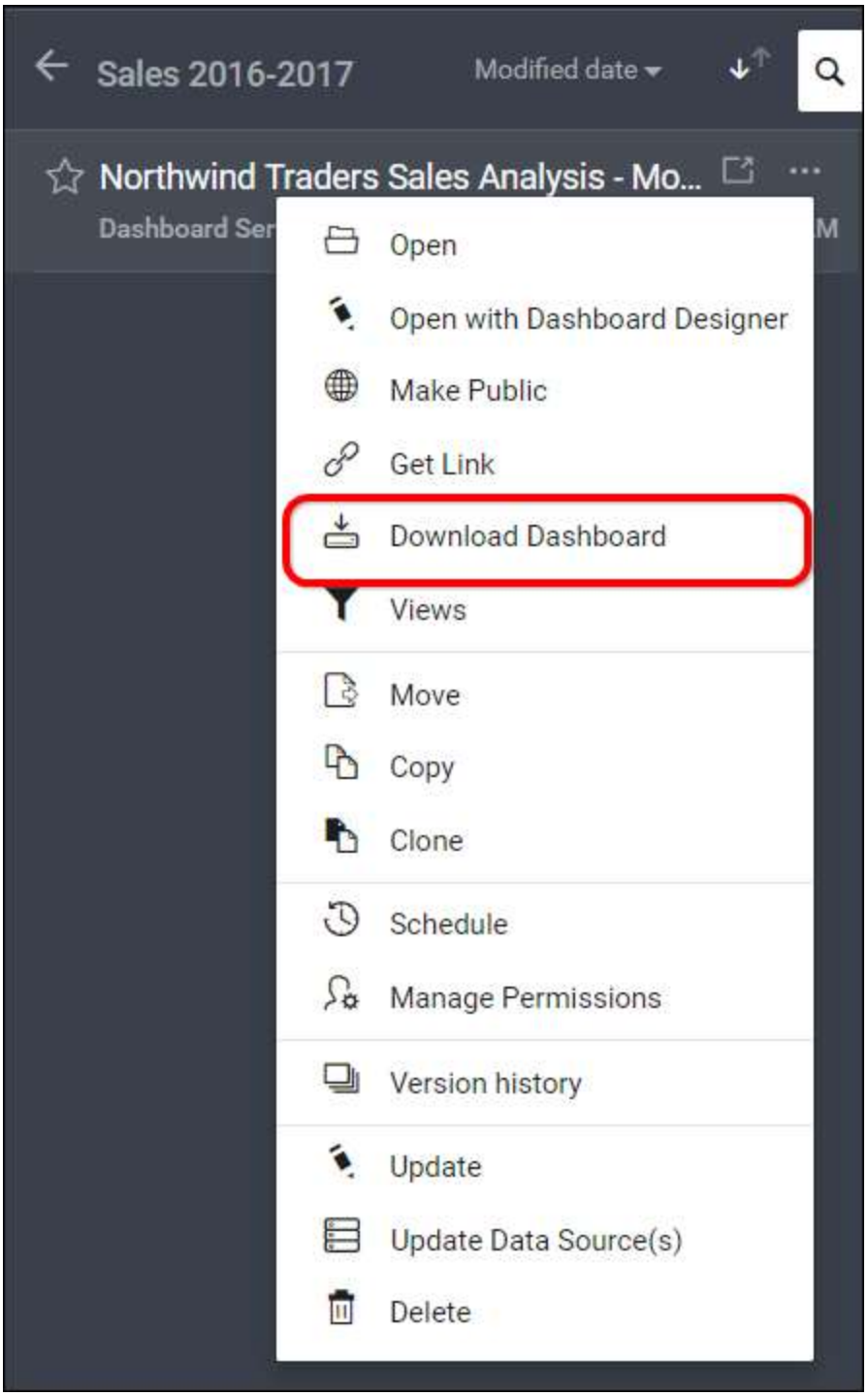


Dashboards are opened in our embedded Dashboard Viewer and Dashboards can also be exported in image format.

Download Dashboards

This section explains on how to download Dashboards from the Syncfusion Dashboard Server.

Click the Actions button in the Dashboards grid context menu and select Download to download the Dashboard in .sydx format.



Downloaded Dashboard can be loaded in our Dashboard Designer.

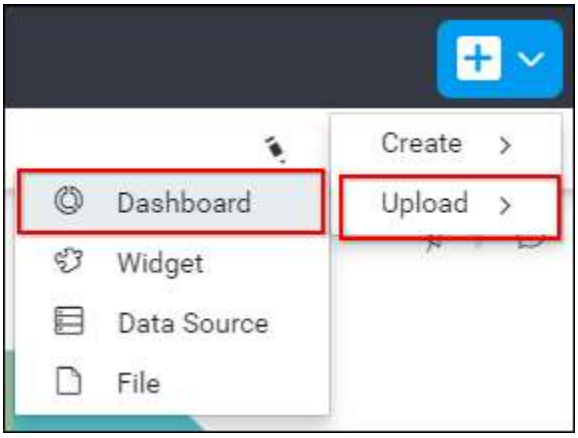
Note: This Download option will be shown only for the dashboards which are created using Desktop Dashboard Designer.

Add Dashboards

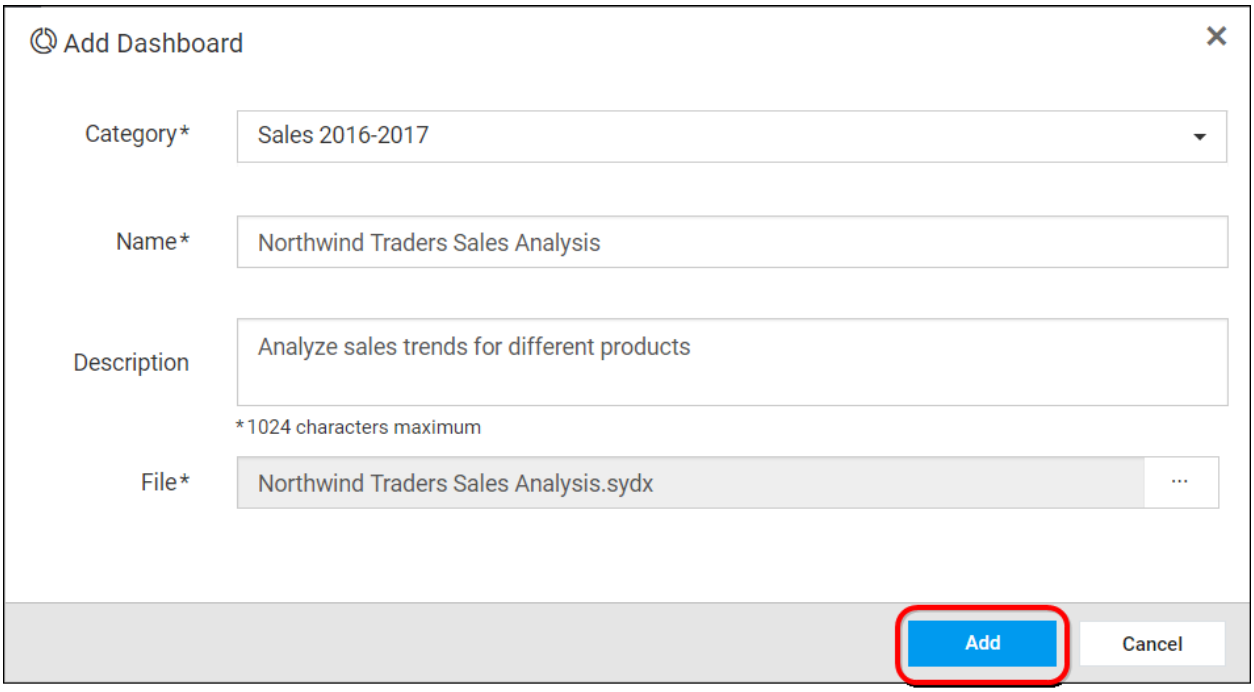
This section explains on how to add Dashboards in the Syncfusion Dashboard Server.

Steps to add a Dashboard

1. Click on the Upload button in the menu and select Dashboards to add a Dashboard.



2. Select a category for the Dashboard and fill in the name and description of the Dashboard and upload the Dashboard file(.sydx) in the Add Dashboard dialog box.



3. After filling the form, the Dashboard can be saved to be added in the Dashboard server.

Note: Read Write Delete Download permission for that Specific Dashboard will be added for the user who created the Dashboard.

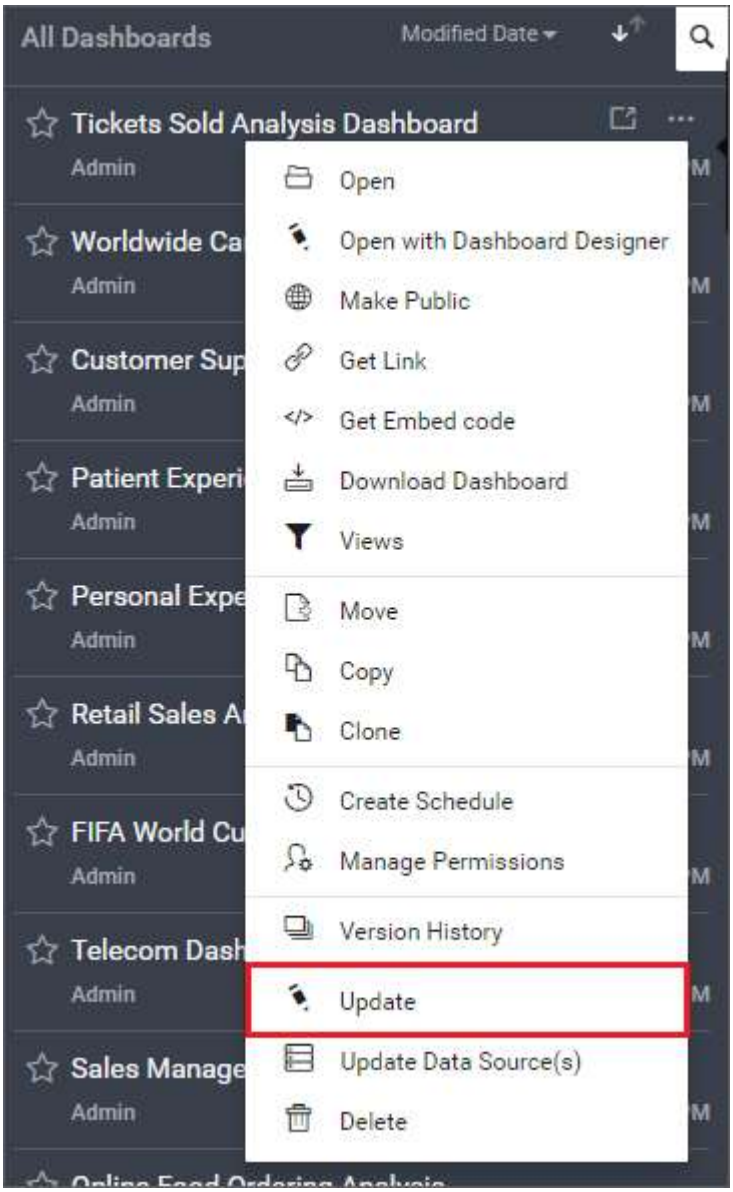
Update Dashboards

This section explains on how to add Dashboards in the Syncfusion Dashboard Server.

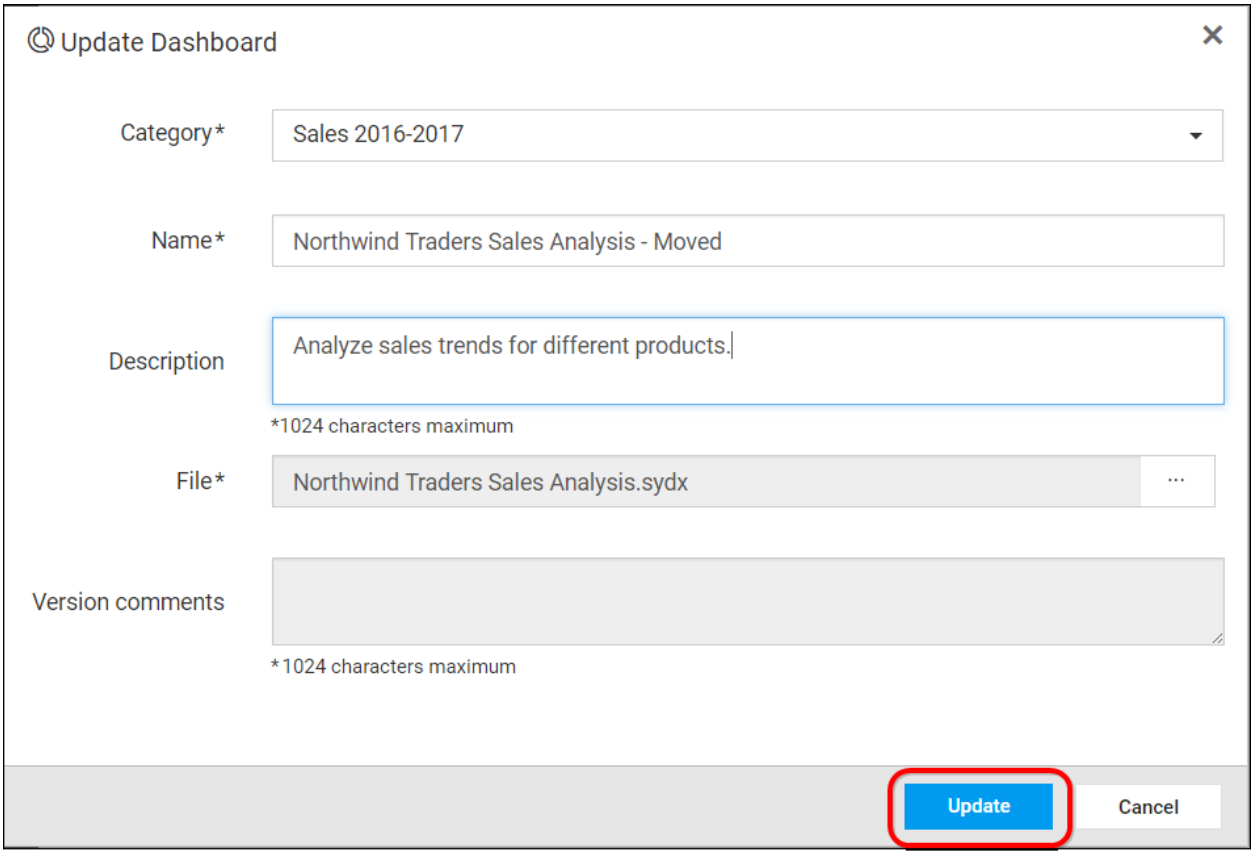
Steps to update a Dashboard

Dashboards can be updated to move the Dashboard to a different category. Name, description and the Dashboard file(.sydx) can be changed for the Dashboard in the update Dashboard dialog box.

1. Click on the Update option in the context menu of the respective Dashboard.



2. Click on the Update button in the Update Dashboard dialog box after making changes to the Category, Name, Description or to the Dashboard file(.sydx). Comments can also be added if there is a change in the Dashboard file(.sydx) to maintain as Version Comments.



Schedule Dashboards

Dashboards can be made to run at scheduled times and export them as images to mail them to the users in the Dashboard server.

(Discussed more in the Manage Schedules section)

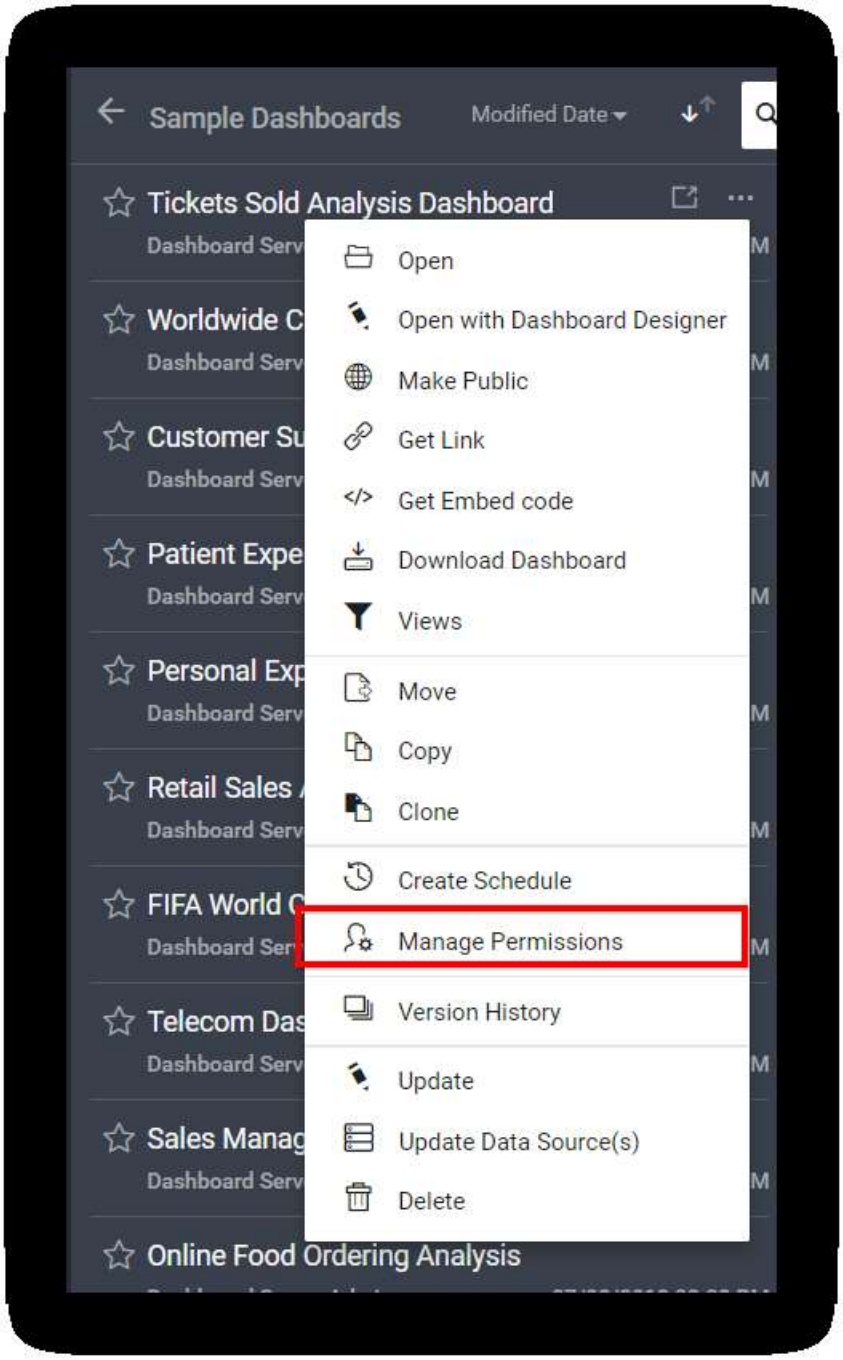
Share Dashboards

This section explains on how to share dashboards with the other users in the Dashboard Server.

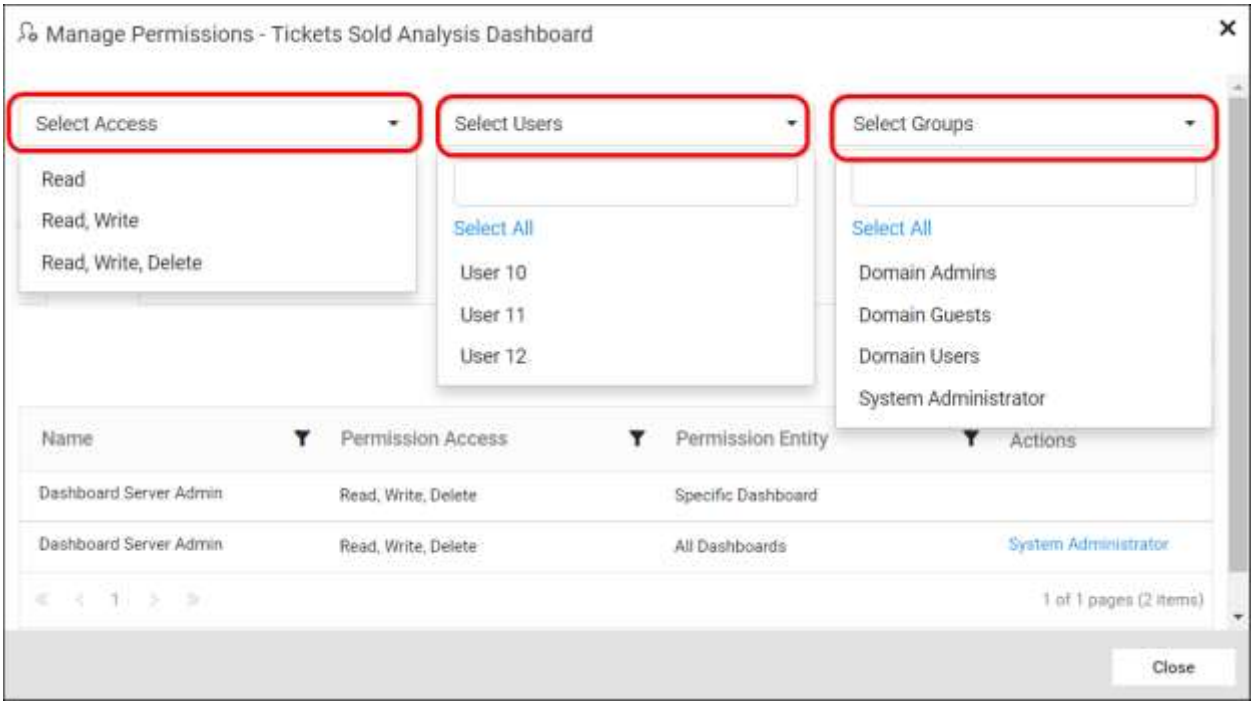
Note: Only the user who created the dashboard and the Administrator can share the dashboard with other Dashboard Server users.

Steps to share a Dashboard

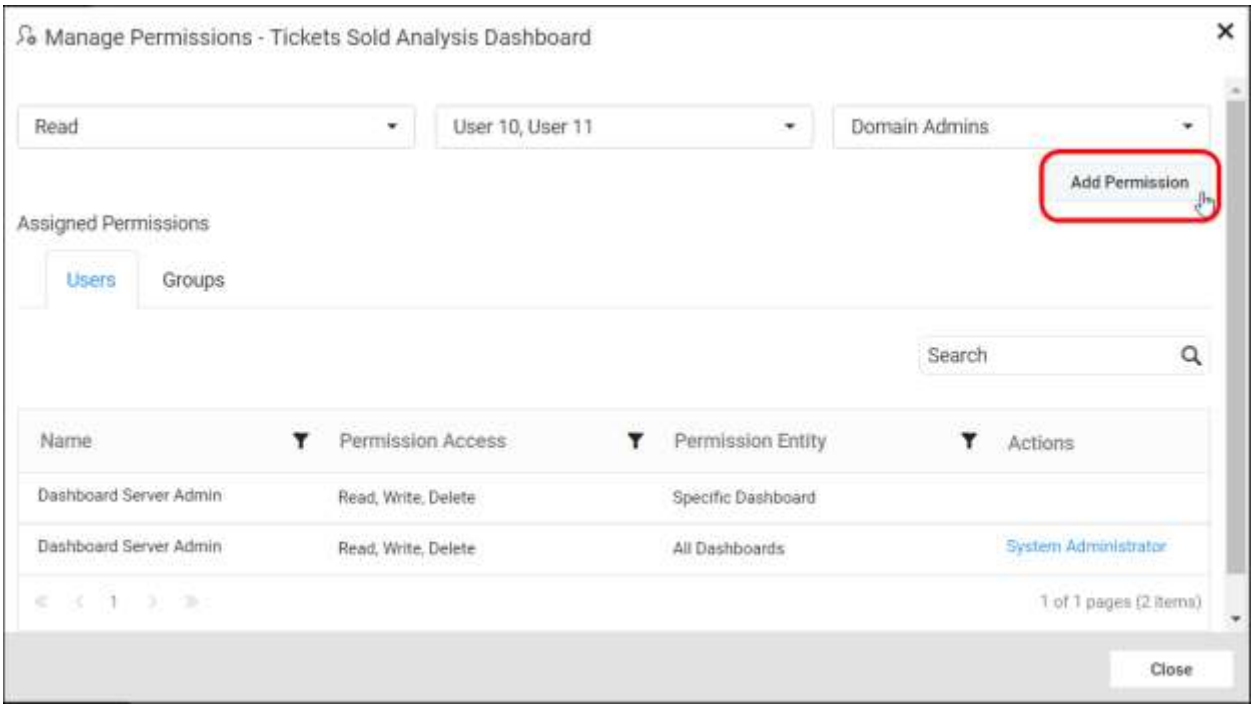
1. Click the Actions button in the Dashboards grid context menu and select Manage Permissions option.



2. Select the permission access from the Select Access dropdown and select the users or groups to share the dashboard.

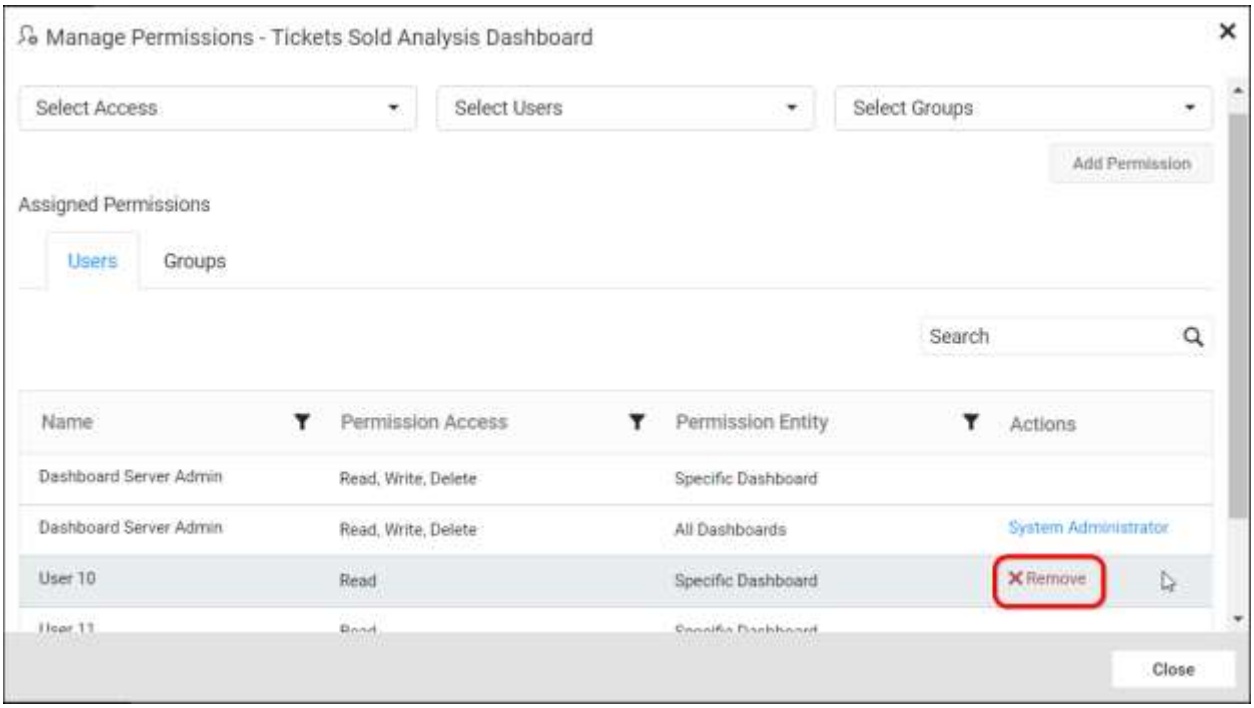


3. After selecting the access and users or groups, click on the Add Permission button.



Remove Permission

The user who created the dashboard and the Administrator can remove the shared dashboard permissions using the Remove option in the Actions column of the each permissions.



Favorite Dashboard

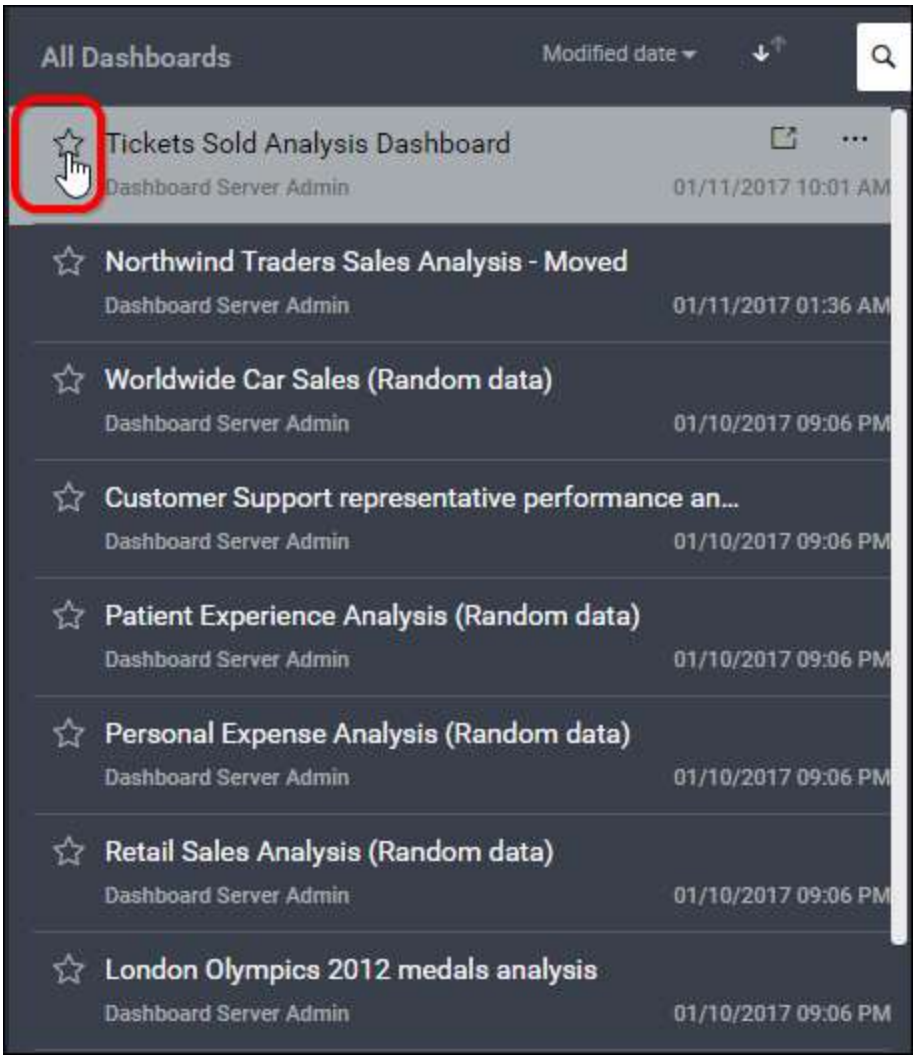
This section explains about how to mark Dashboards as favorites, remove a particular Dashboard from favorites and view the list of favorite Dashboards in the Syncfusion Dashboard Server.

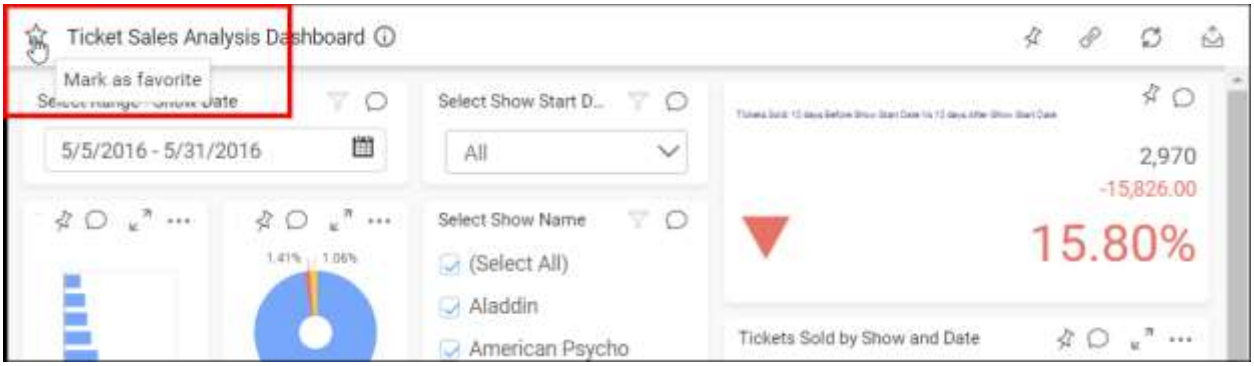
Mark a Dashboard as favorite

Dashboards can be marked as favorite to view them in the Favorite Dashboards category instead of searching them in the Categories or using keywords in the Dashboards list.

To mark a Dashboard as favorite, click on the star icon near the Dashboard name.

On clicking, star icon is filled with color to indicate that it is added as favorite Dashboard.

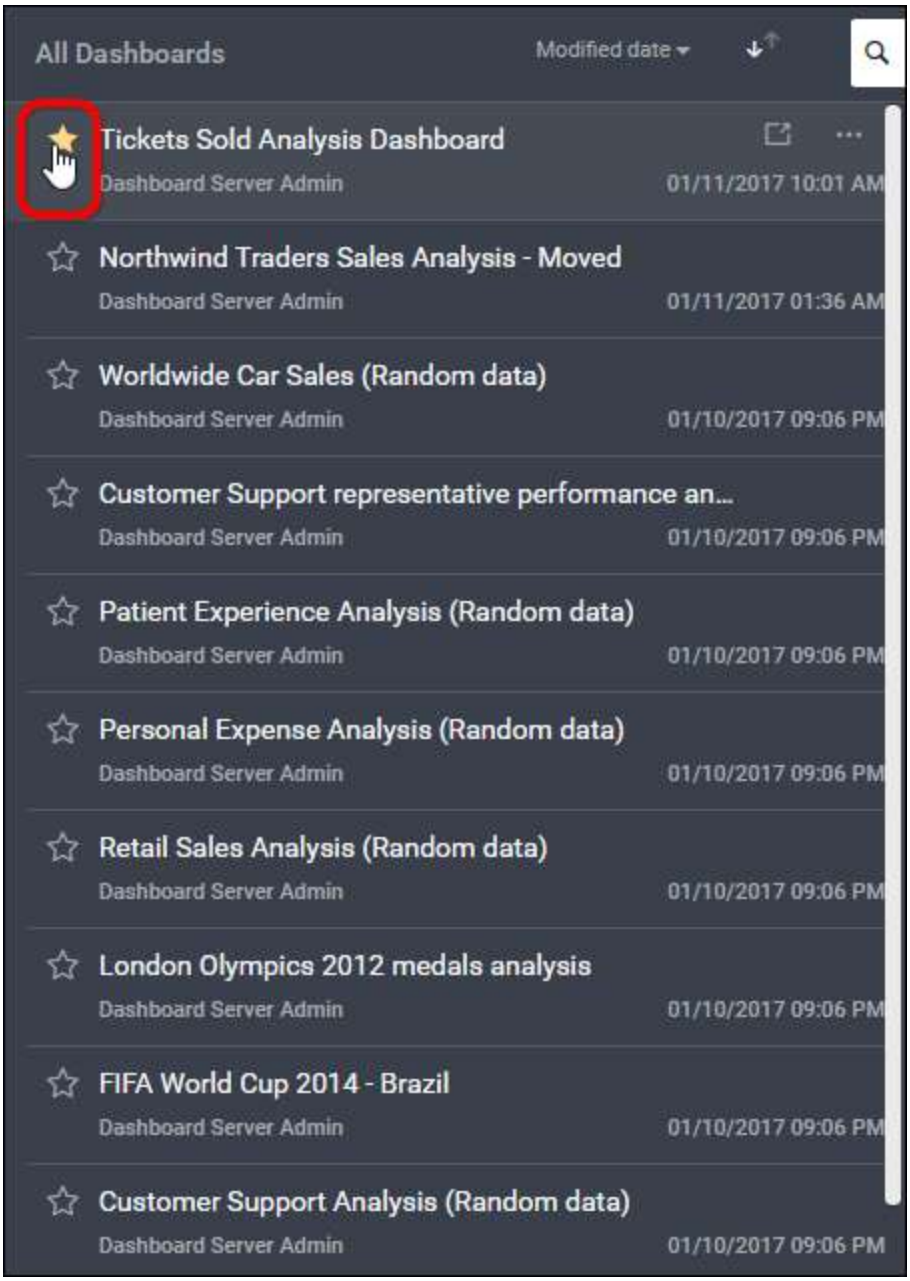


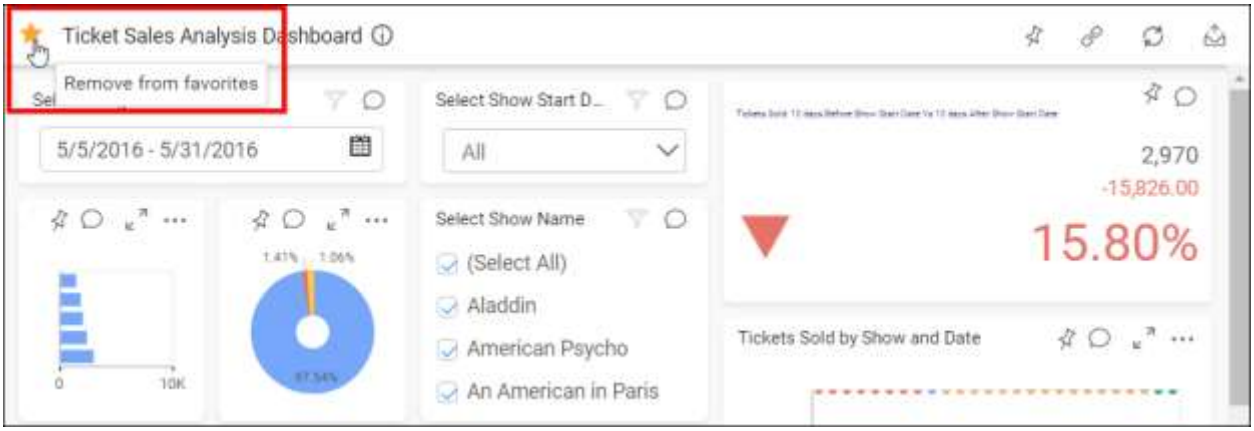


Remove a Dashboard from favorites

To remove a dashboard from favorites, click on the star icon near the Dashboard name.

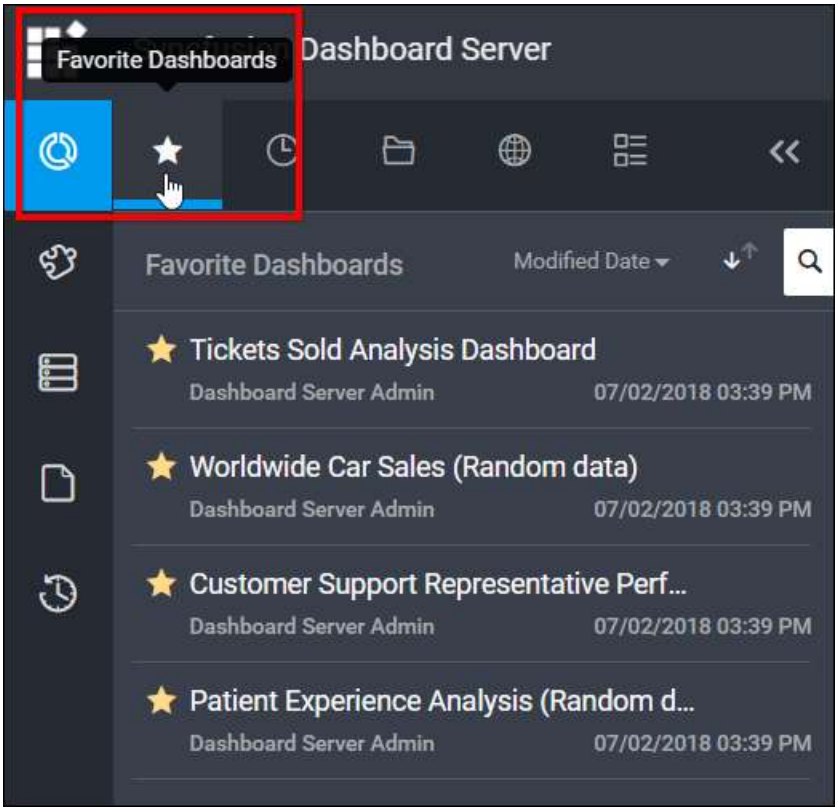
On clicking, star icon color is emptied to indicate that it is removed from favorites.


All Dashboards
Modified date
Tickets Sold Analysis Dashboard
Dashboard Server Admin
01/11/2017 10:01 AM
Northwind Traders Sales Analysis - Moved
Dashboard Server Admin
01/11/2017 01:36 AM
Worldwide Car Sales (Random data)
Dashboard Server Admin
01/10/2017 09:06 PM
Customer Support representative performance an...
Dashboard Server Admin
01/10/2017 09:06 PM
Patient Experience Analysis (Random data)
Dashboard Server Admin
01/10/2017 09:06 PM
Personal Expense Analysis (Random data)
Dashboard Server Admin
01/10/2017 09:06 PM
Retail Sales Analysis (Random data)
Dashboard Server Admin
01/10/2017 09:06 PM
London Olympics 2012 medals analysis
Dashboard Server Admin
01/10/2017 09:06 PM
FIFA World Cup 2014 - Brazil
Dashboard Server Admin
01/10/2017 09:06 PM
Customer Support Analysis (Random data)
Dashboard Server Admin
01/10/2017 09:06 PM


Ticket Sales Analysis Dashboard
Remove from favorites
5/5/2016 - 5/31/2016
Select Show Start D...
All
2,970
-15,826.00
15.80%
Select Show Name
(Select All)
Aladdin
American Psycho
An American in Paris
Tickets Sold by Show and Date

Favorite Dashboards Category

Dashboards that are marked as favorite can be viewed under Favorite Dashboards category.



Privacy Settings of Dashboard

This section explains on how to make the Dashboards public, private and unlisted.

Private Dashboards are accessible to the registered users in the Dashboard Server who has appropriate permissions.

Public Dashboards are accessible to anonymous users who has the Dashboard link.

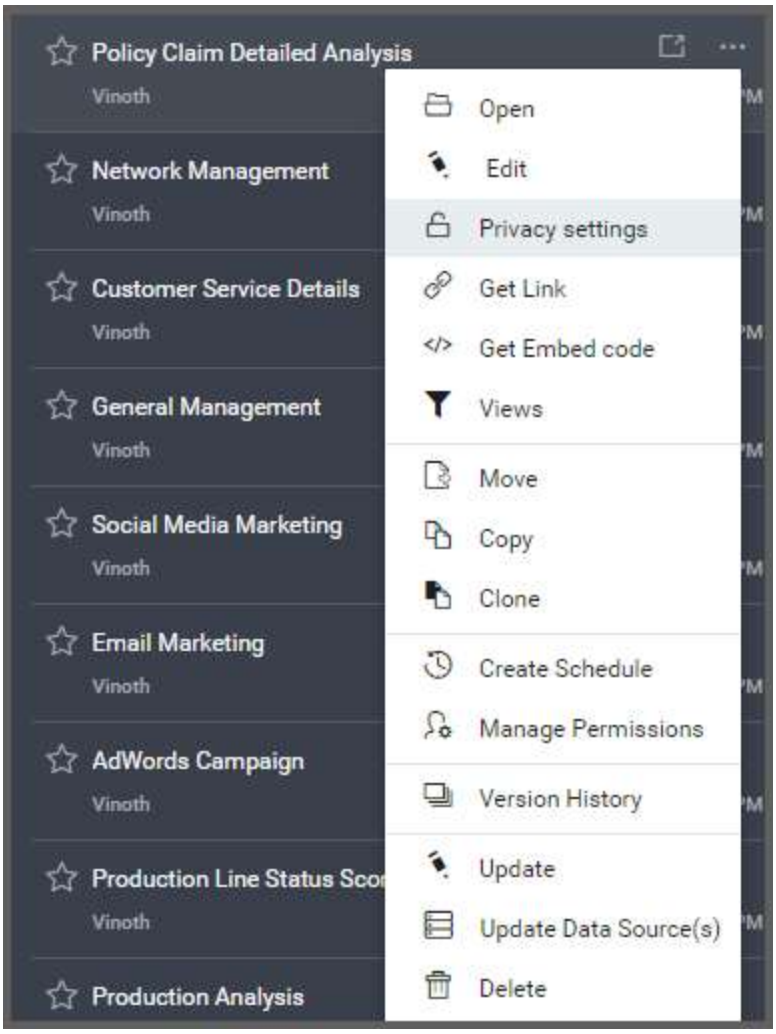
Unlisted Dashboards are accessible to anonymous users who has the Dashboard link with unlisted code. Only dashboard owners can see them in their dashboards list.

Privacy Settings option is available only to the owner of the Dashboard.

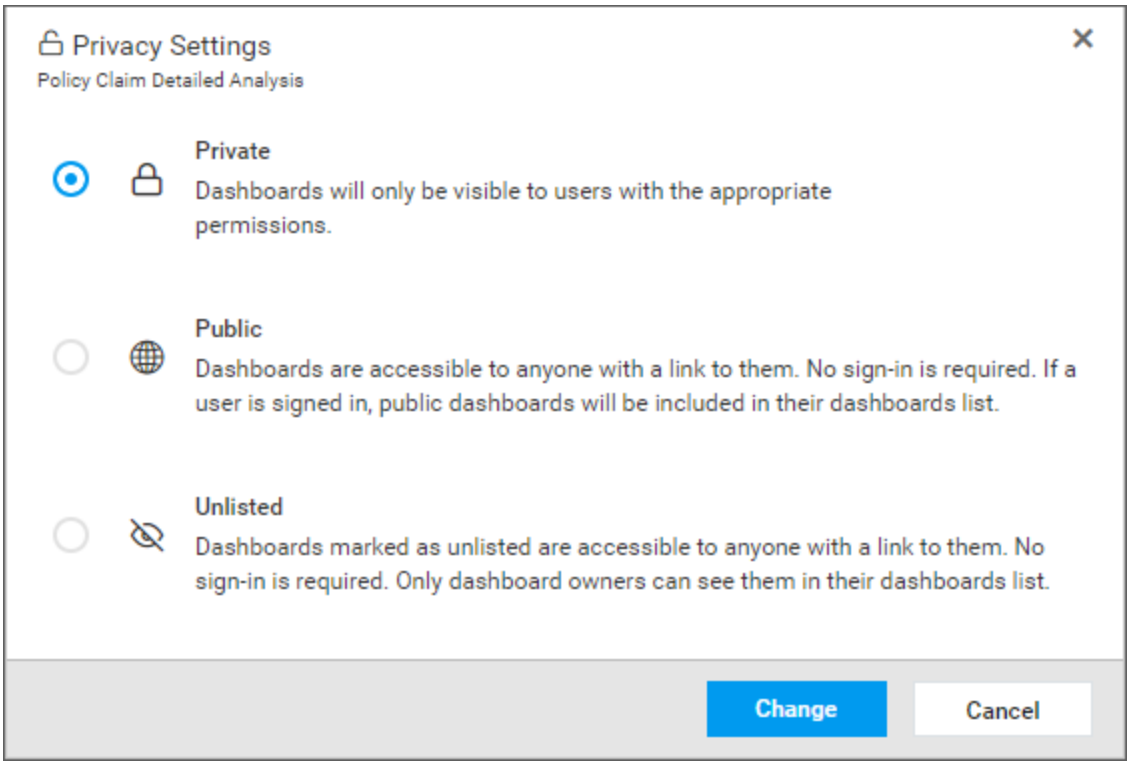
Make Private

By default all the dashboards are sets as Private. If you want to revert the Dashboard as Private from Public or Unlisted state, Please follow the steps below to make the Dashboards accessible only to the users in the Dashboard Server who has appropriate permissions.

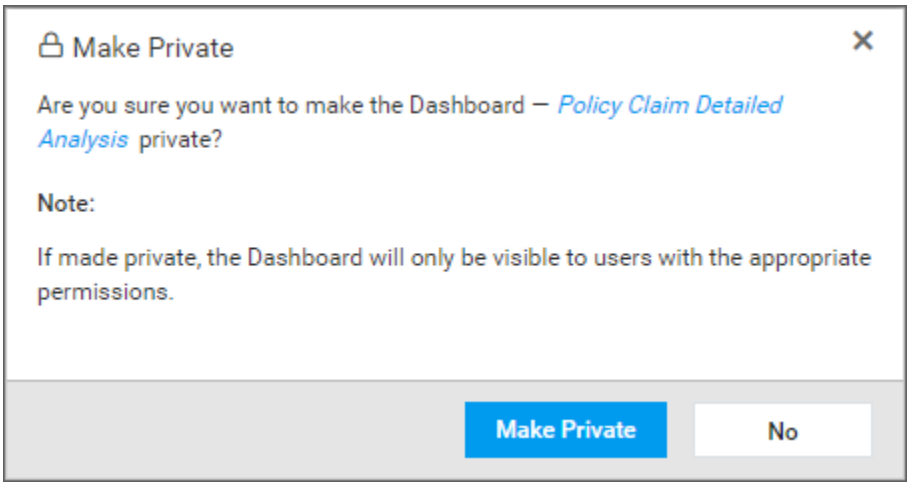
1. Click on the context menu of the respective Dashboard and choose Privacy Settings option.



2. Choose the Private option and click the Change button in the following dialog.



2. Click on Make Private in the following confirmation dialog box.



Once the Dashboard is made private, dialog box with the confirmation message is displayed.

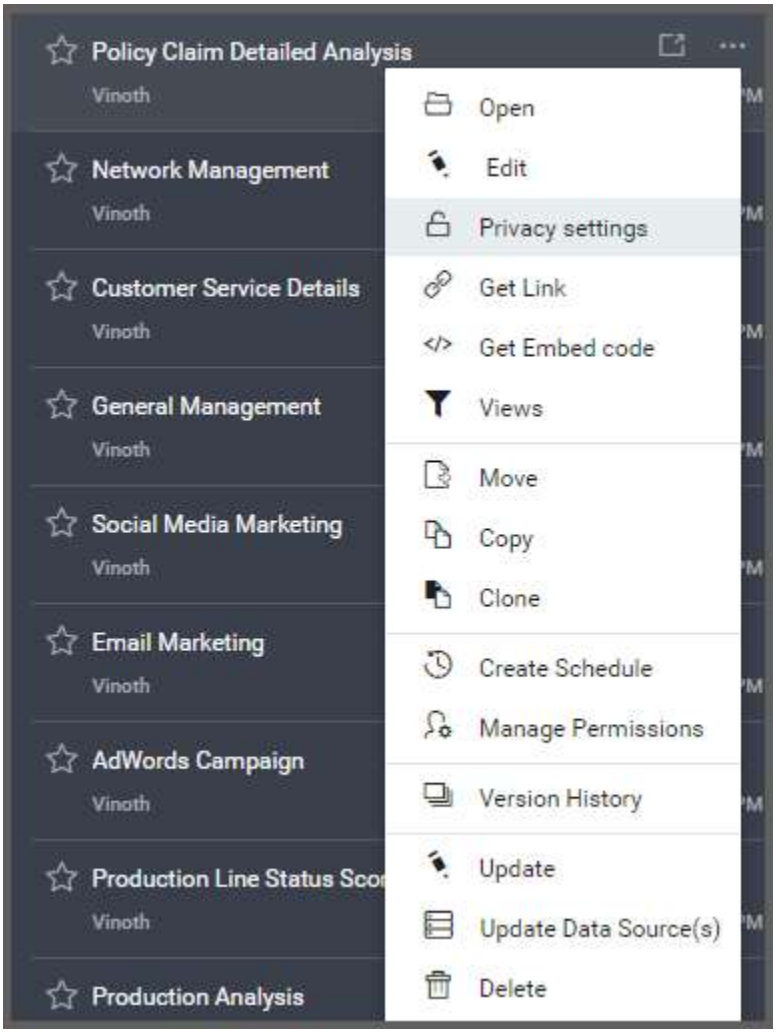
Note:

1. Anonymous user couldn't render the Private Dashboard.
2. Logged user can only render the Dashboard and do the Commenting, Filtering and Views actions on it, who has appropriate permissions.

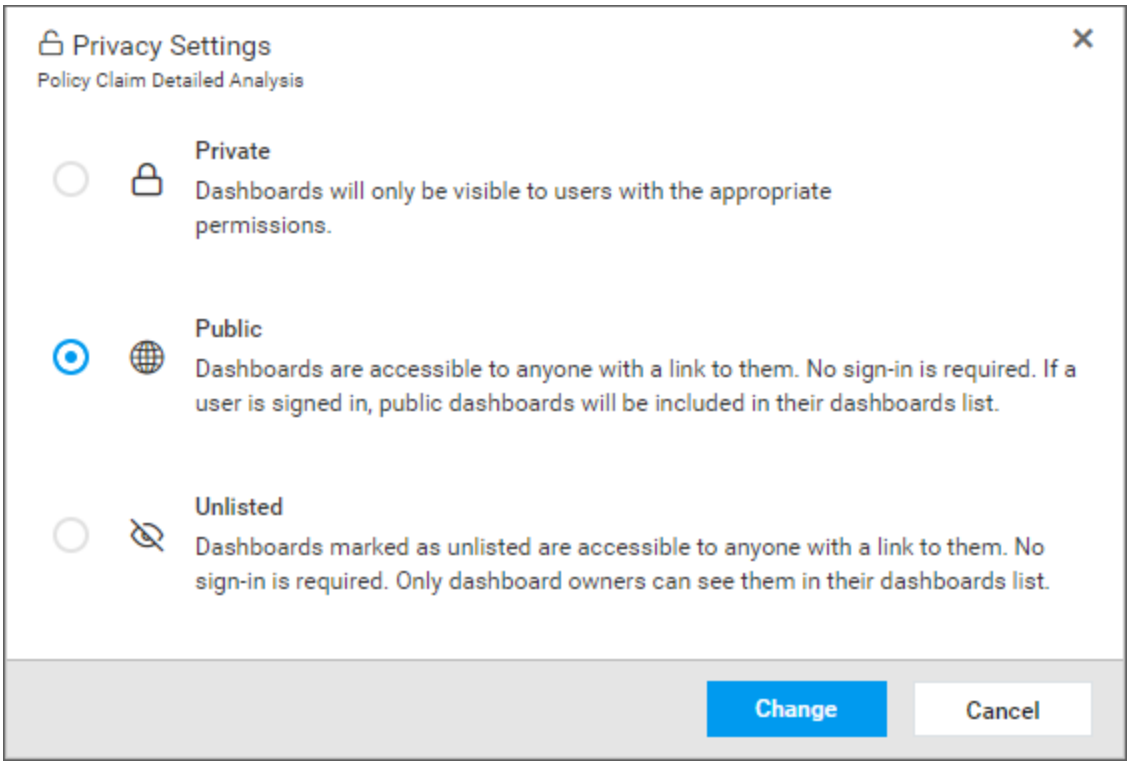
Make Public

If you want to change the Dashboard as Public from Private or Unlisted state, Please follow the steps below to make the Dashboards accessible to anonymous users.

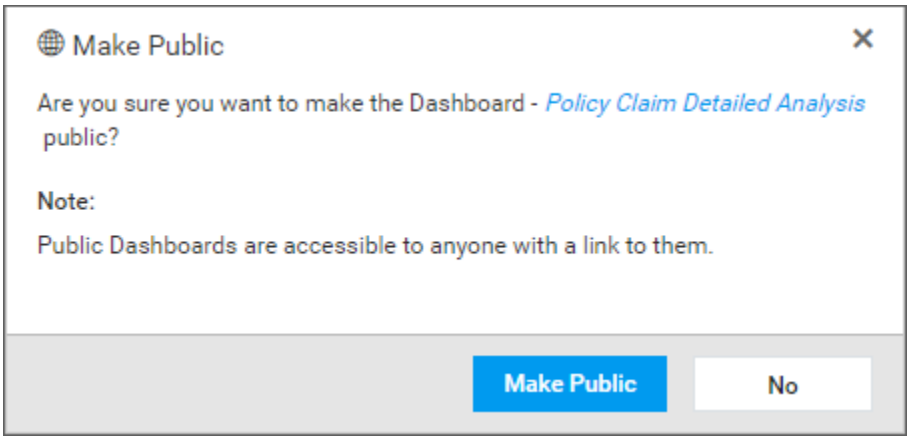
1. Click on the context menu of the respective Dashboard and choose Privacy Settings option.



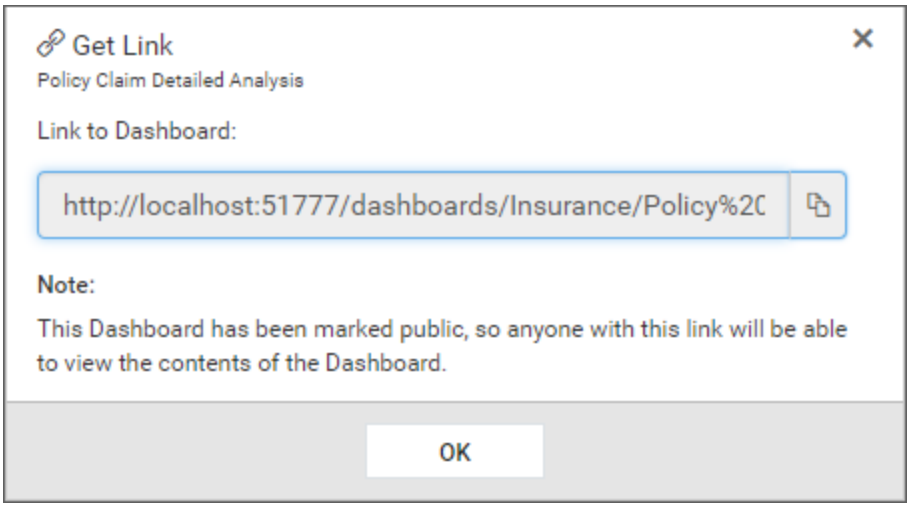
2. Choose the Public option and click the Change button in the following dialog.



3. Click the Make Public option in the following confirmation dialog box.



Once the Dashboard is made public, below dialog box with the Dashboard link is displayed.



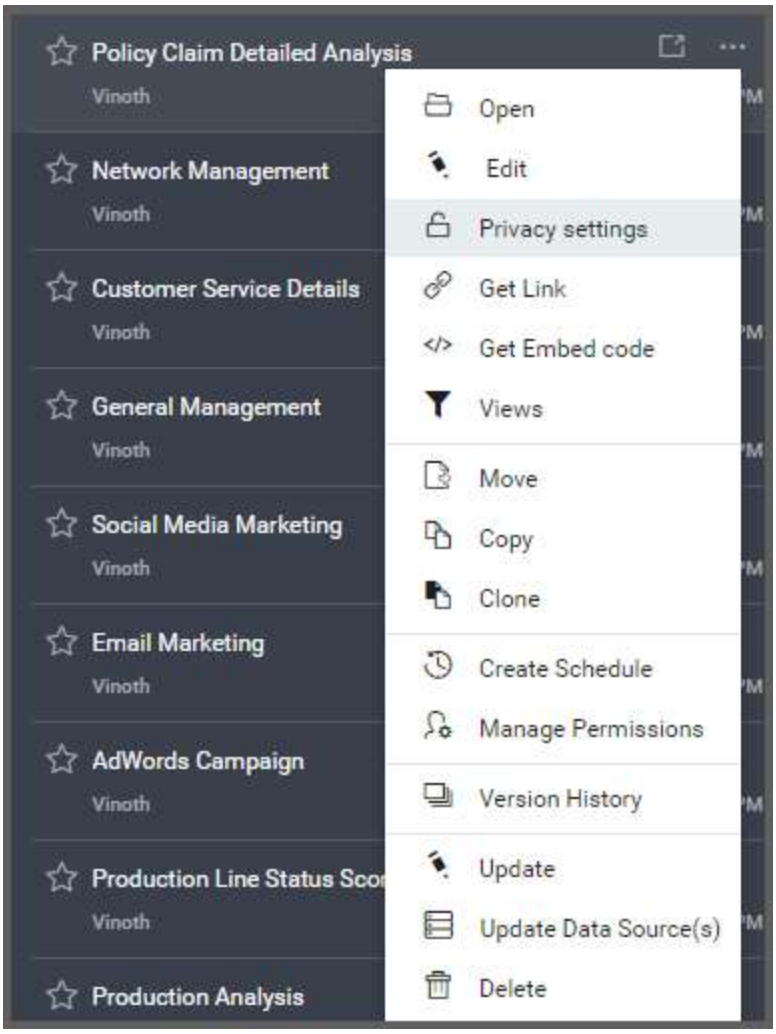
Note:

1. Anonymous user can only render the Public Dashboard but couldn't Comment, Filter on it.
2. Logged user only can do the Commenting, Filtering and Views actions on the Dashboard who has appropriate permissions.

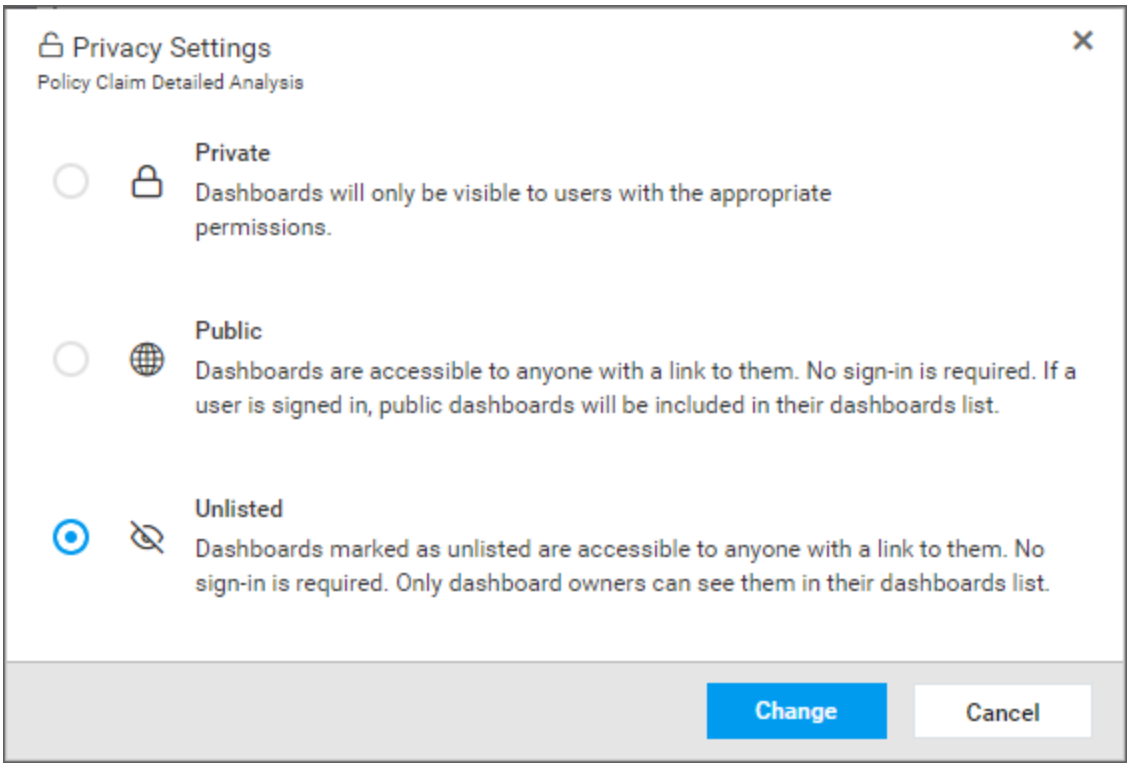
Make Unlisted

If you want to change the Dashboard as Unlisted from Private or Public state, Please follow the steps below to make the Dashboards accessible to anonymous users.

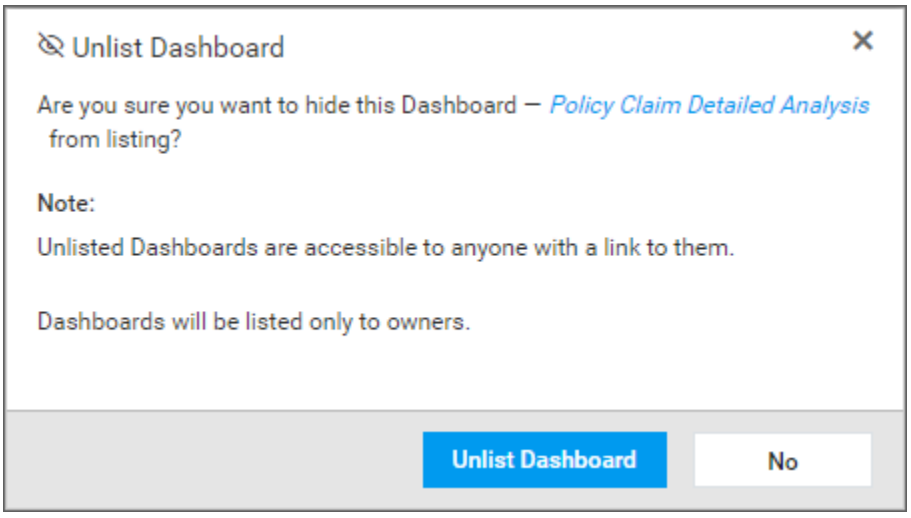
1. Click on the context menu of the respective Dashboard and choose Privacy Settings option.



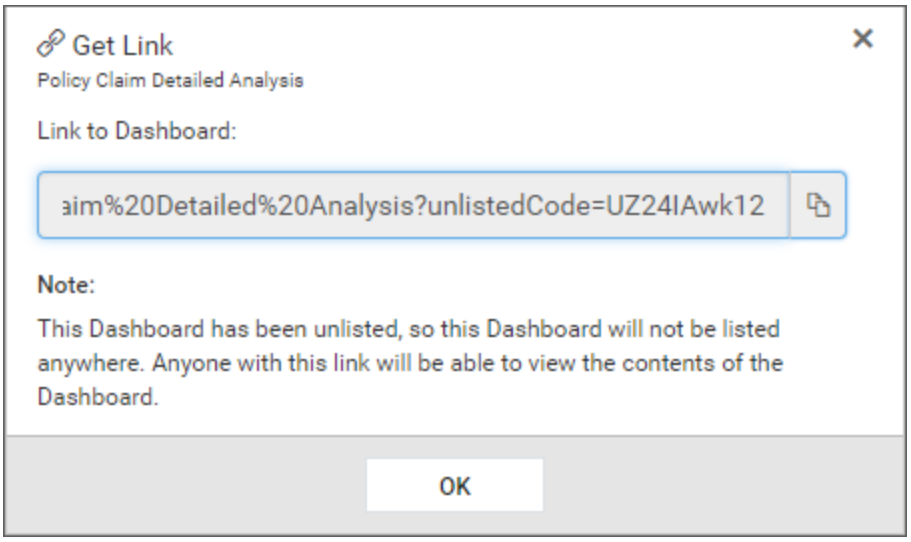
2. Choose the Unlisted option and click the Change button in the following dialog.



3. Click the Unlist Dashboard option in the following confirmation dialog box.



Once the Dashboard is made unlisted, below dialog box with the Unlisted Dashboard link is displayed. So this Dashboard will be listed only on owner's Dashboards list.

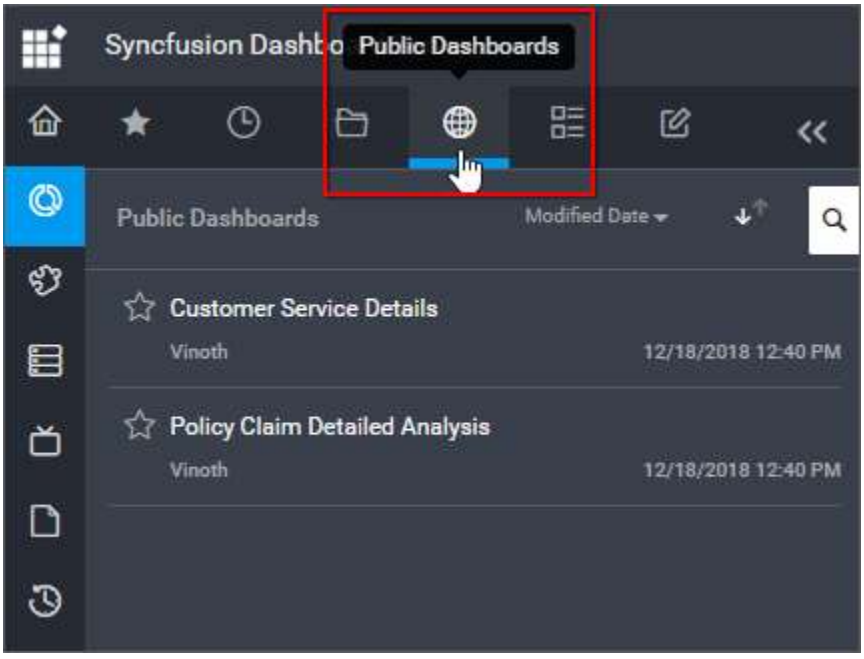


Note:

1. Anonymous user can only render the Unlisted Dashboard but couldn't comment and filter on it.
2. Unlisted Dashboard link will have a unlisted code along with the link.
3. Unlisted Dashboard link will be generated every time when the owner unlist it.
4. If a new Unlisted Code is generated, the link with the existing Unlisted Code will be expired and that Dashboard couldn't render.
5. Unlisted code is not necessary for logged owner of that dashboard.

Public Dashboards

Public dashboards are listed in the below icon section in the home page of Syncfusion Dashboard Server.



Note: Click here to get more details about the public - allows/restricts switch.

Get Dashboard Link

This section explains on how to get link to the Dashboards in the Syncfusion Dashboard Server.

These links are used to navigate to the Dashboard and can be shared with others.

If the Dashboard is public, anyone with this link will be able to view its contents.

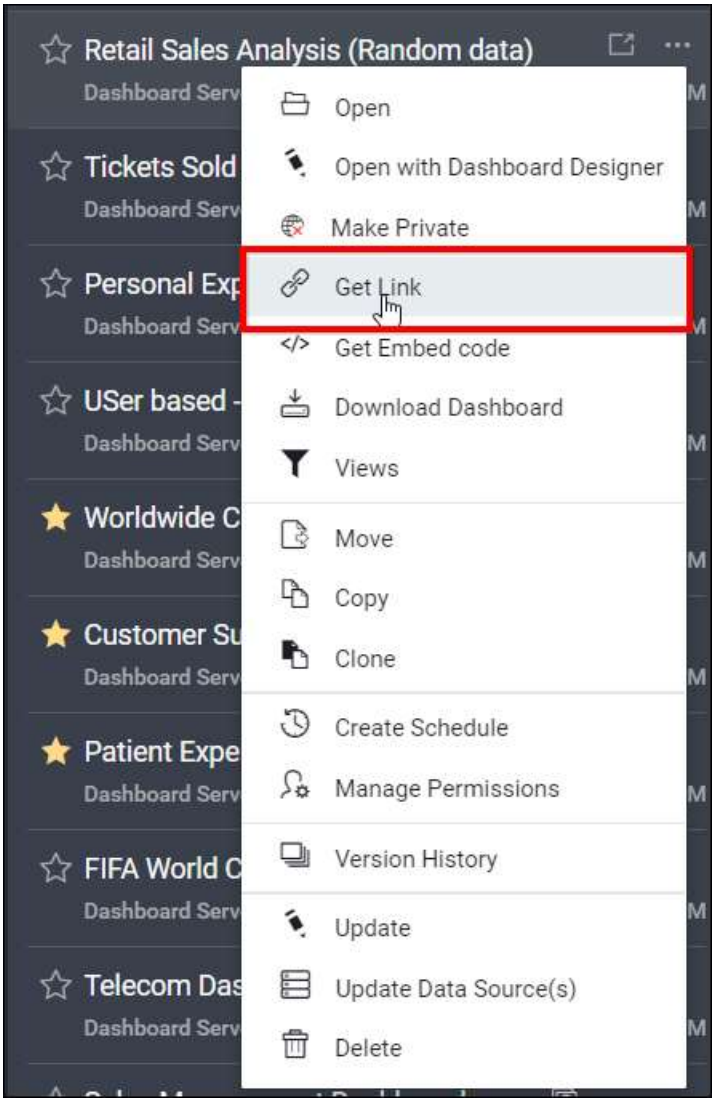
If the Dashboard is private, anyone with this link can navigate to the Dashboard, but only users with the appropriate permissions will be able to view its contents.

Get Link

Get Link option is available for all the Dashboards.

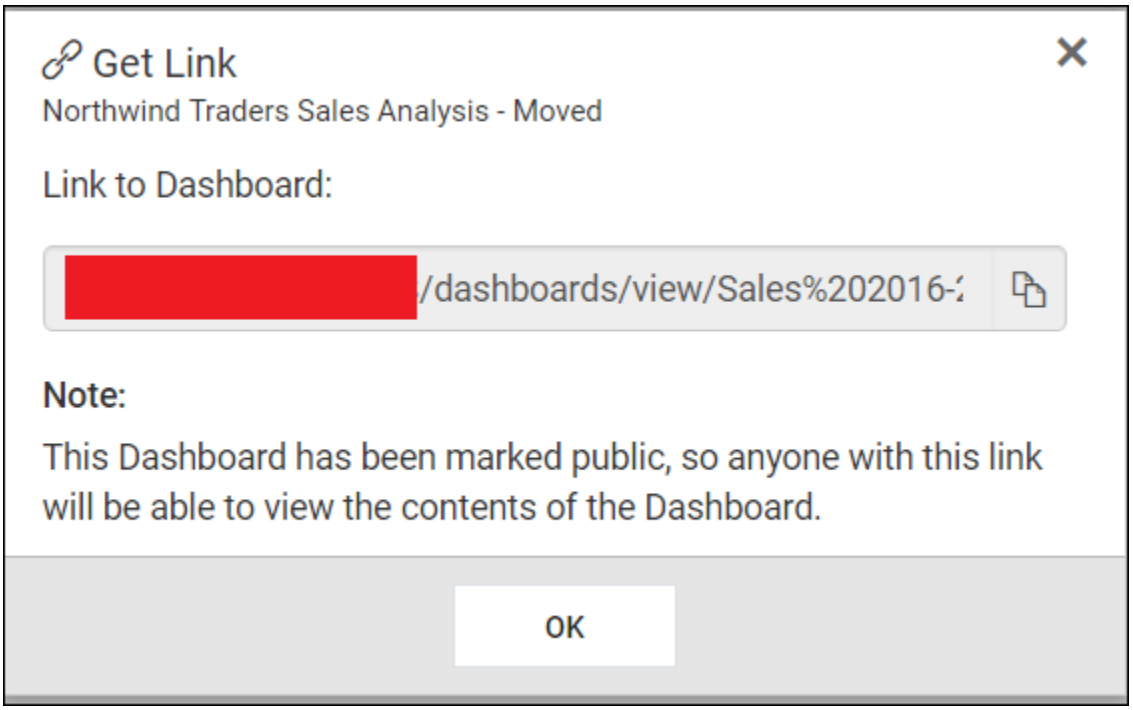
Follow the steps below to get the Dashboards link.

1. Click on the context menu of the respective Dashboard and choose Get Link option.

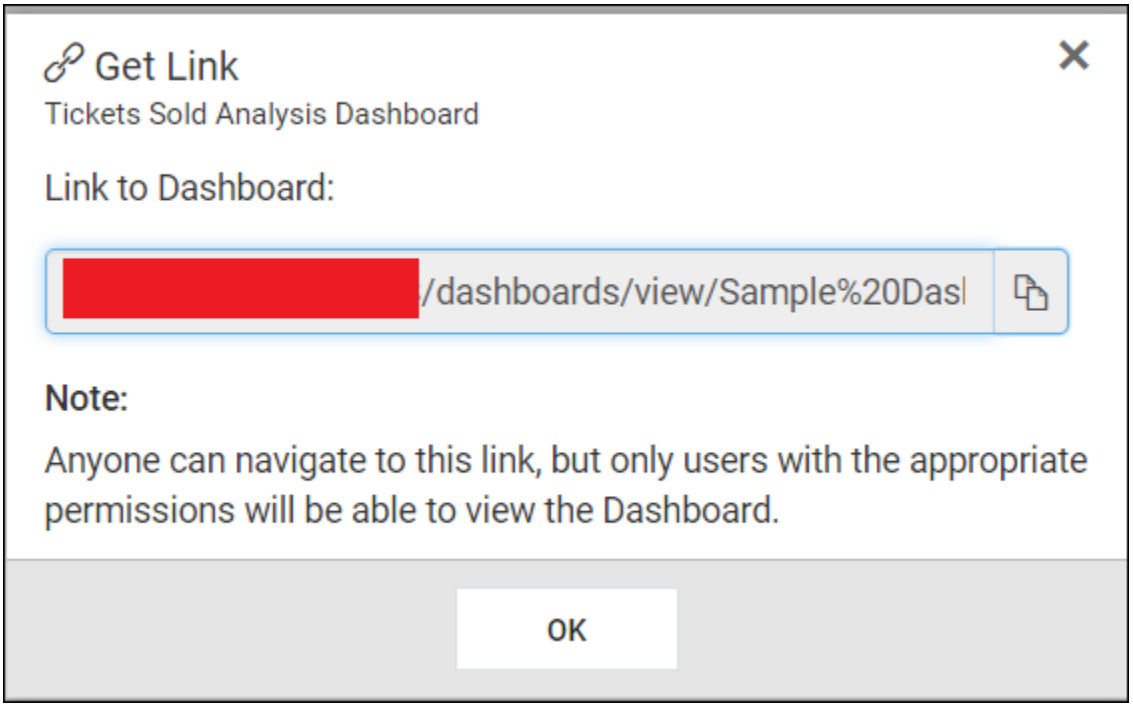


2. Respective Dashboard link will be show in the Get Link dialog box.

For Public Dashboards



For Private Dashboards

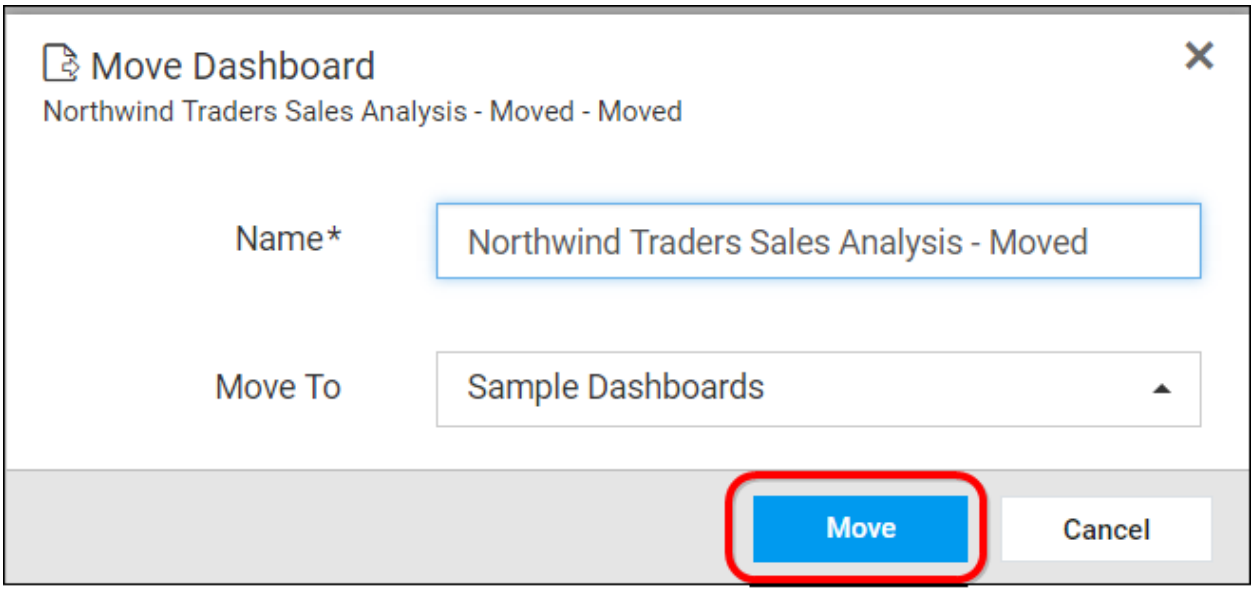


Move, Copy and Clone Dashboards

Dashboards can be moved, copied or cloned from one category to another category.

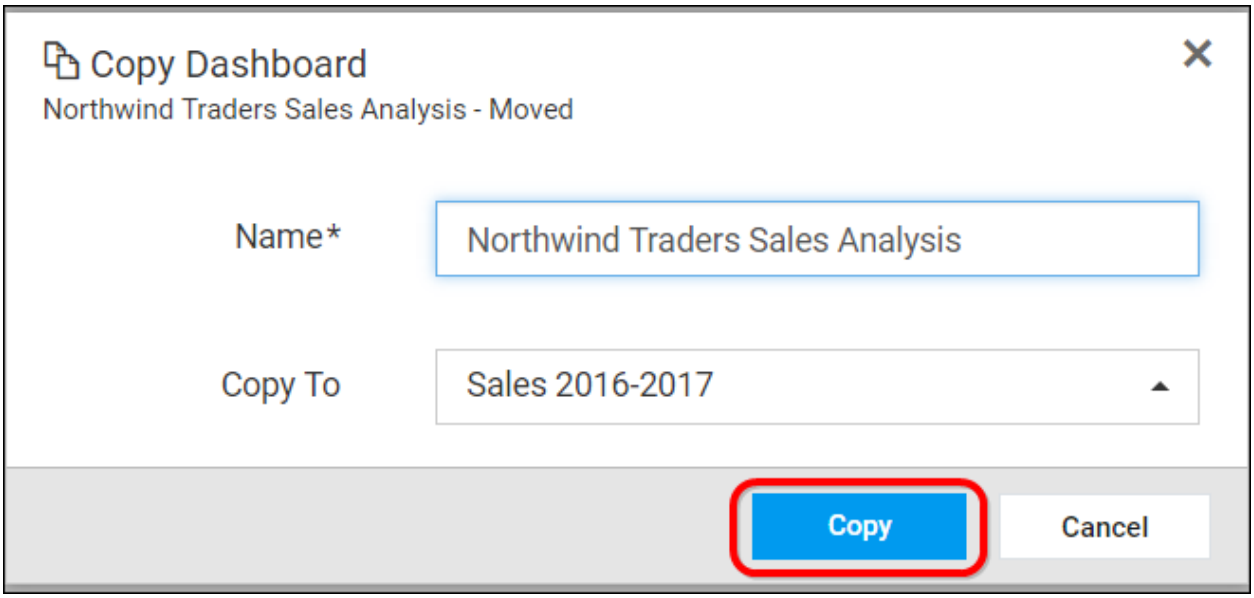
Move Dashboards

Moves the Dashboard from one to another category.



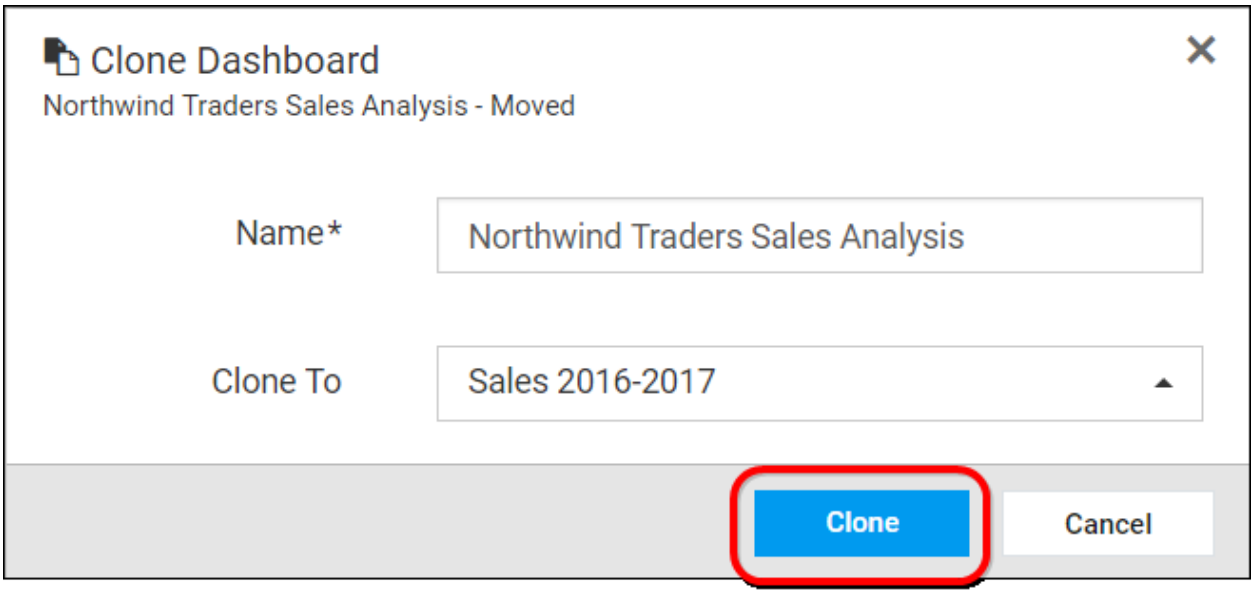
Copy Dashboards

Copies the Dashboard from one to another category.



Clone Dashboards

Creates a reference of the Dashboard to destination category. When the Dashboard `.sydx` file is changed, then it affects the Dashboards in both the categories.

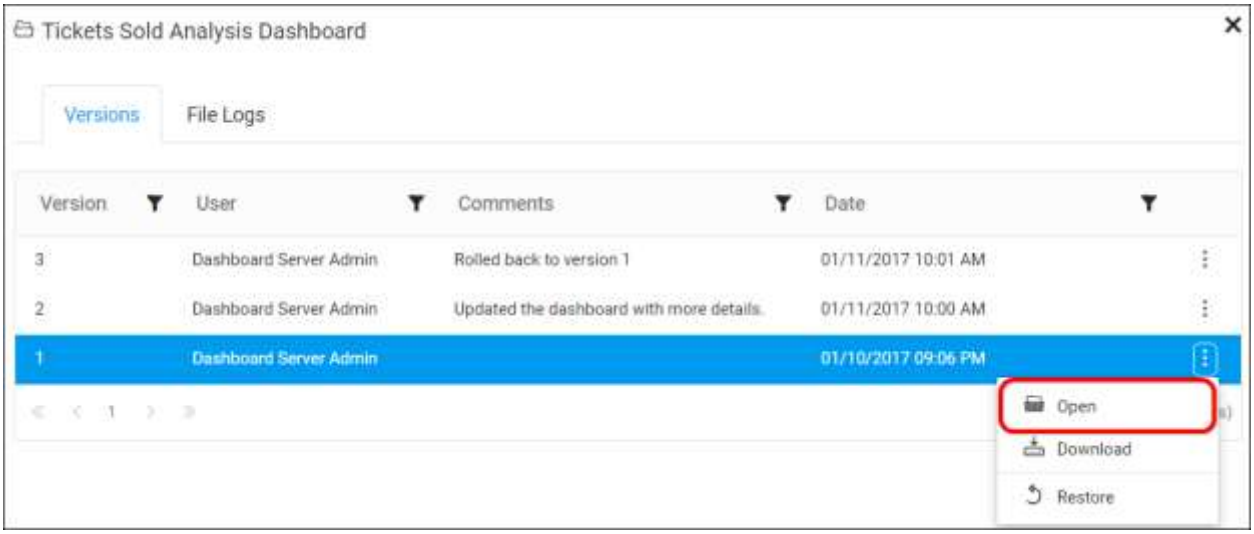


Version History

Versions and file logs for each Dashboard will be maintained in the Dashboard server for every changes in the Dashboard.

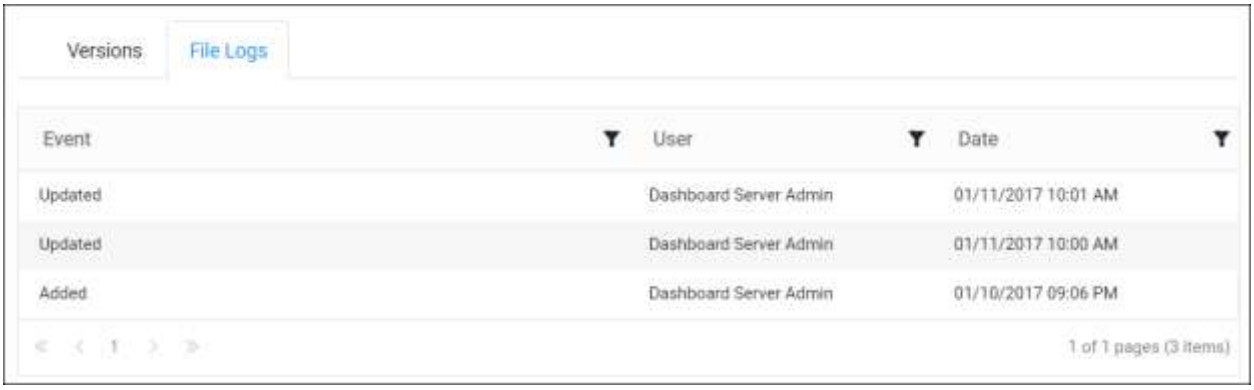
Versions

For each change in the .sydx file, a new version will be created. All versions can be individually opened. At any time, the Dashboard can be rolled back to an older version.



File logs

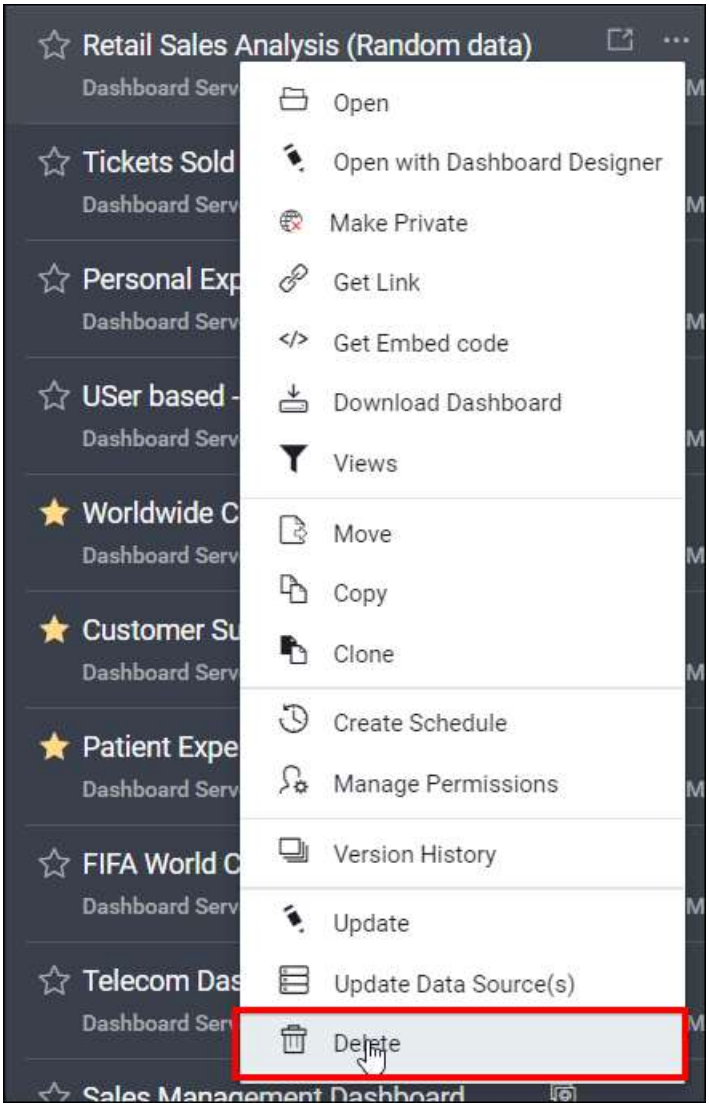
For each change in the Dashboard including changes in the name, description, category and .sydx file, Dashboard server logs the changes done in the file logs.



Delete Dashboards

Dashboards can also be deleted from the Dashboard server when they are no longer required.

Click the Actions button in the Dashboards grid context menu and select Delete to delete the Dashboard.



Note: Dashboards cannot be deleted when they are scheduled by an user.

Shared Data Sources

This section explains about how to use shared Data Sources in existing Dashboard in the Syncfusion Dashboard Server.

Data sources are allowed to connect dashboards to different types of databases or middle-tier business objects.

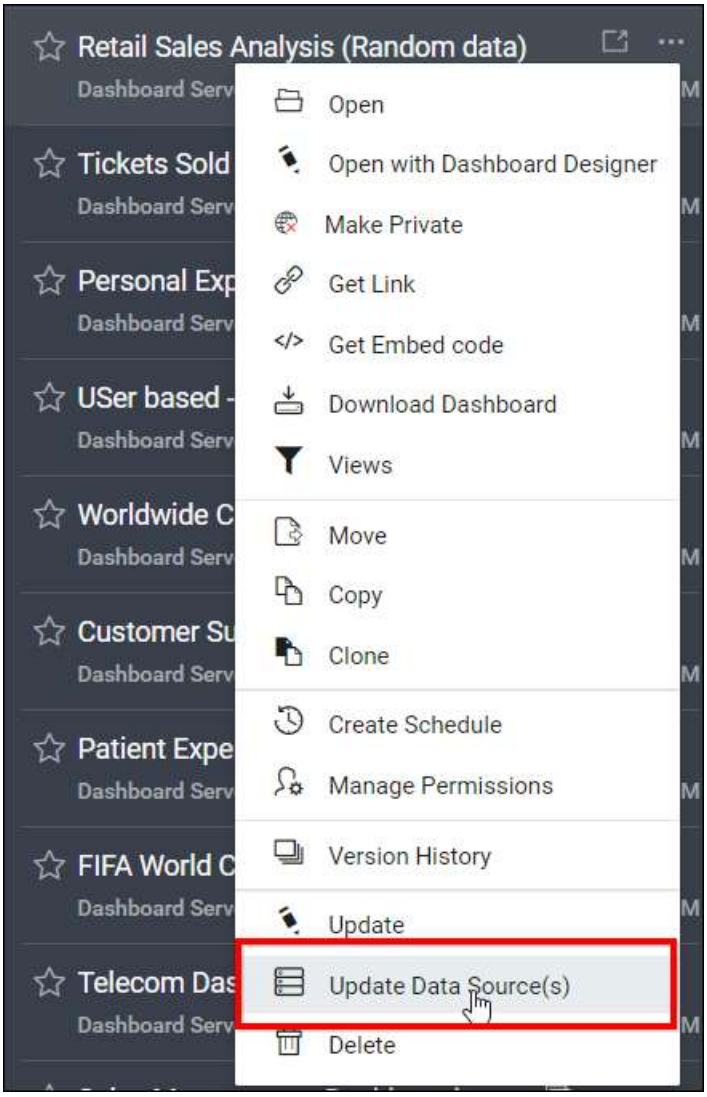
Shared data source the term indicates that the data sources can be shared with multiple dashboards. Examples of shared data sources are as follows,

1. Used to switch between testing and production servers instantly
2. Data sources can be shared across multiple Dashboards. If the database has moved or renamed or database server credentials has updated, it can be easier to update the data source which is shared with the multiple dashboards.

Shared data sources are optional for Dashboards and dashboards can be rendered with the embedded data source itself.

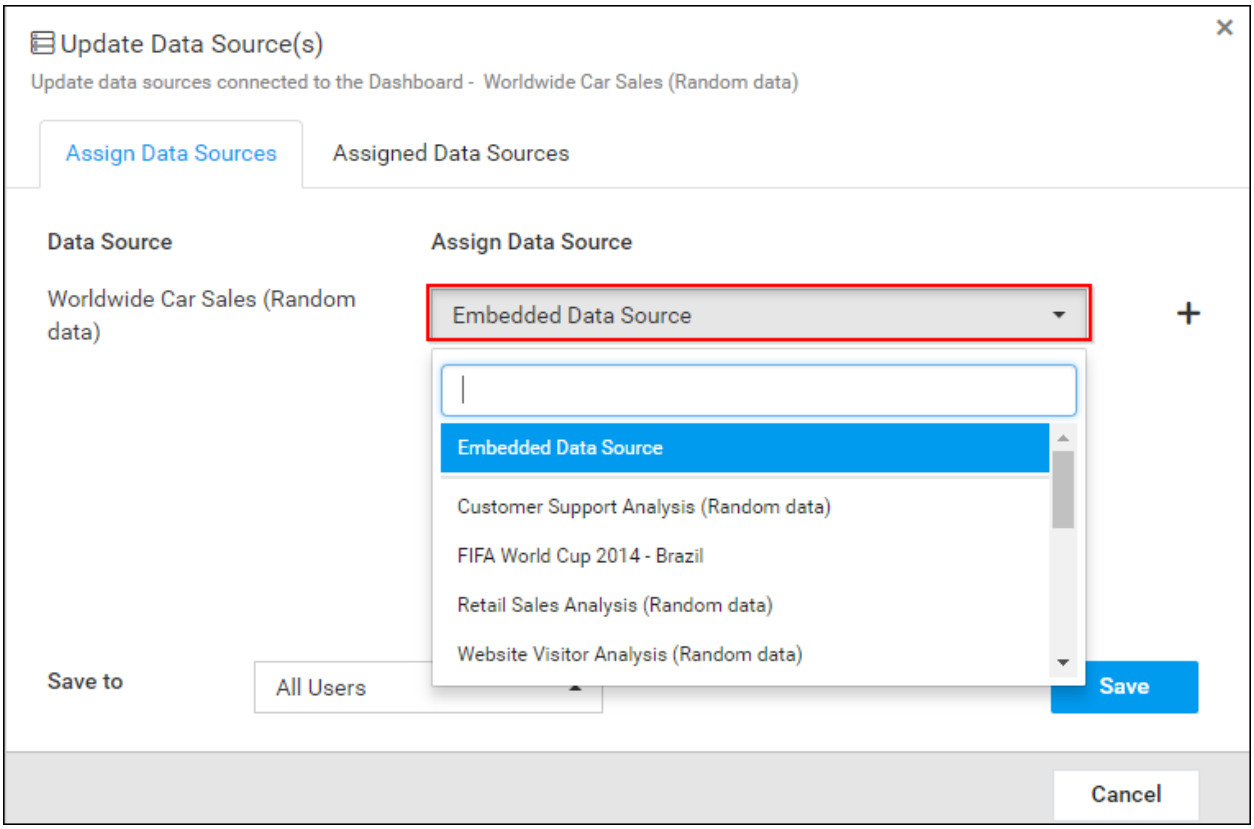
Update Shared Data Sources

To use shared Data Sources for a Dashboard, click on the Update Data Source(s) option from the context menu.

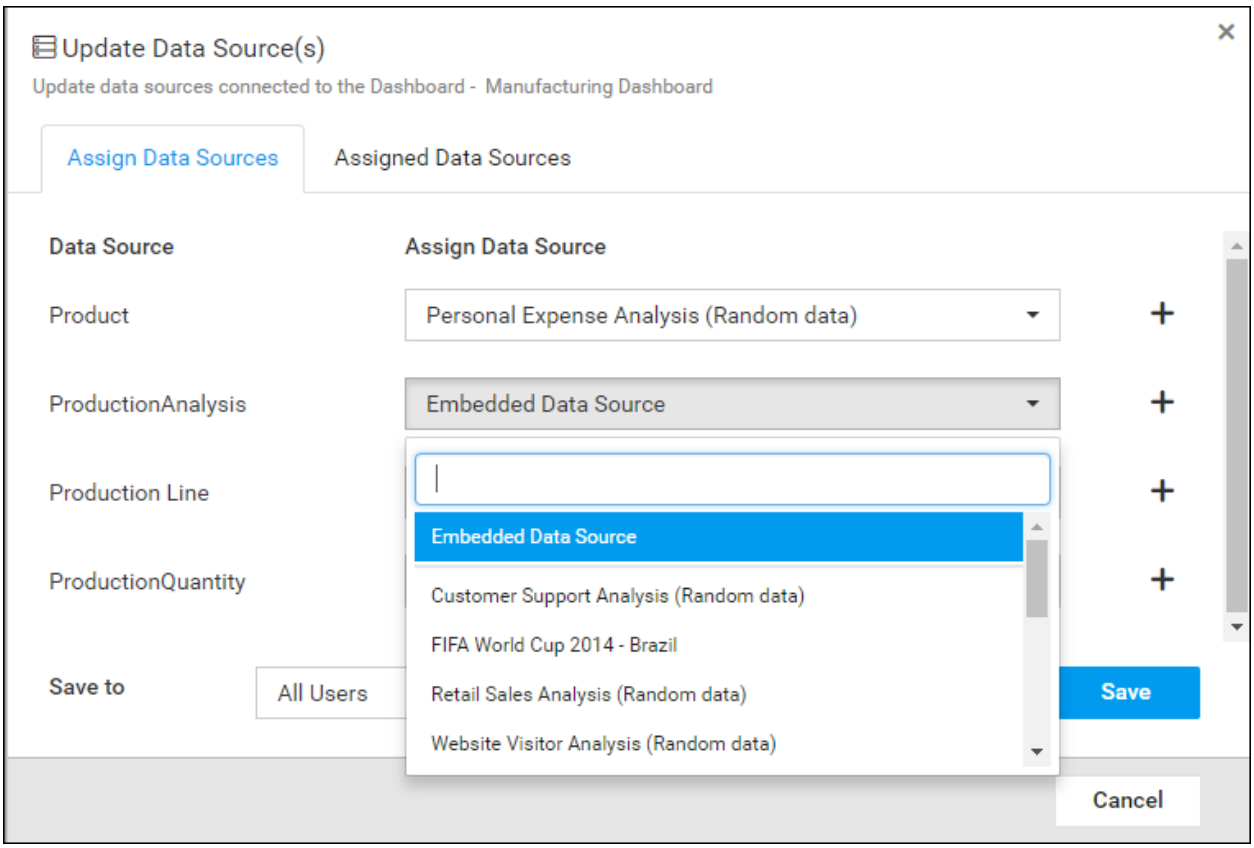


Data Sources already used in the Dashboard along with the available Data Sources in the Server to choose are shown in the dialog box.

By default, all Dashboards use embedded Data Source.



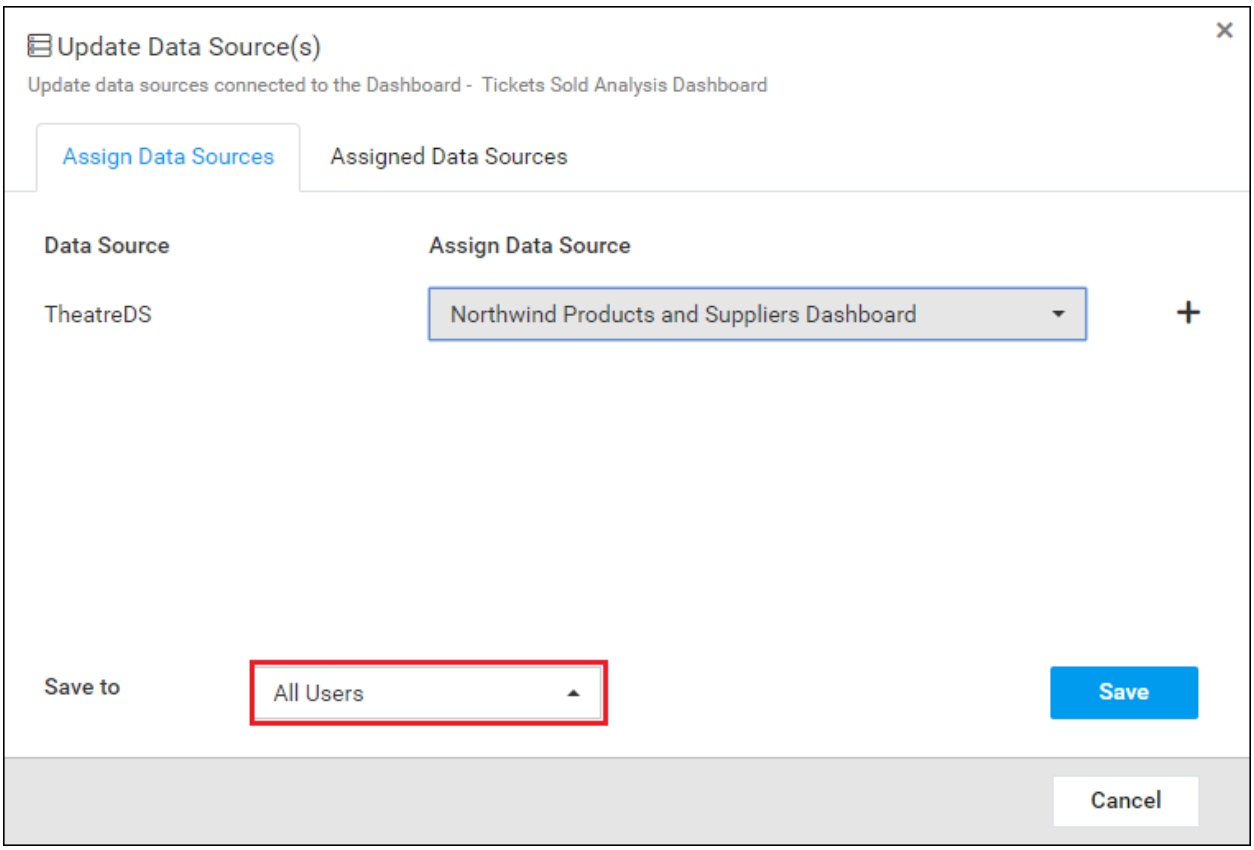
Change the Data Source for Data Source Names available in the Dashboard by selecting it from drop down list or by adding new Data Source.



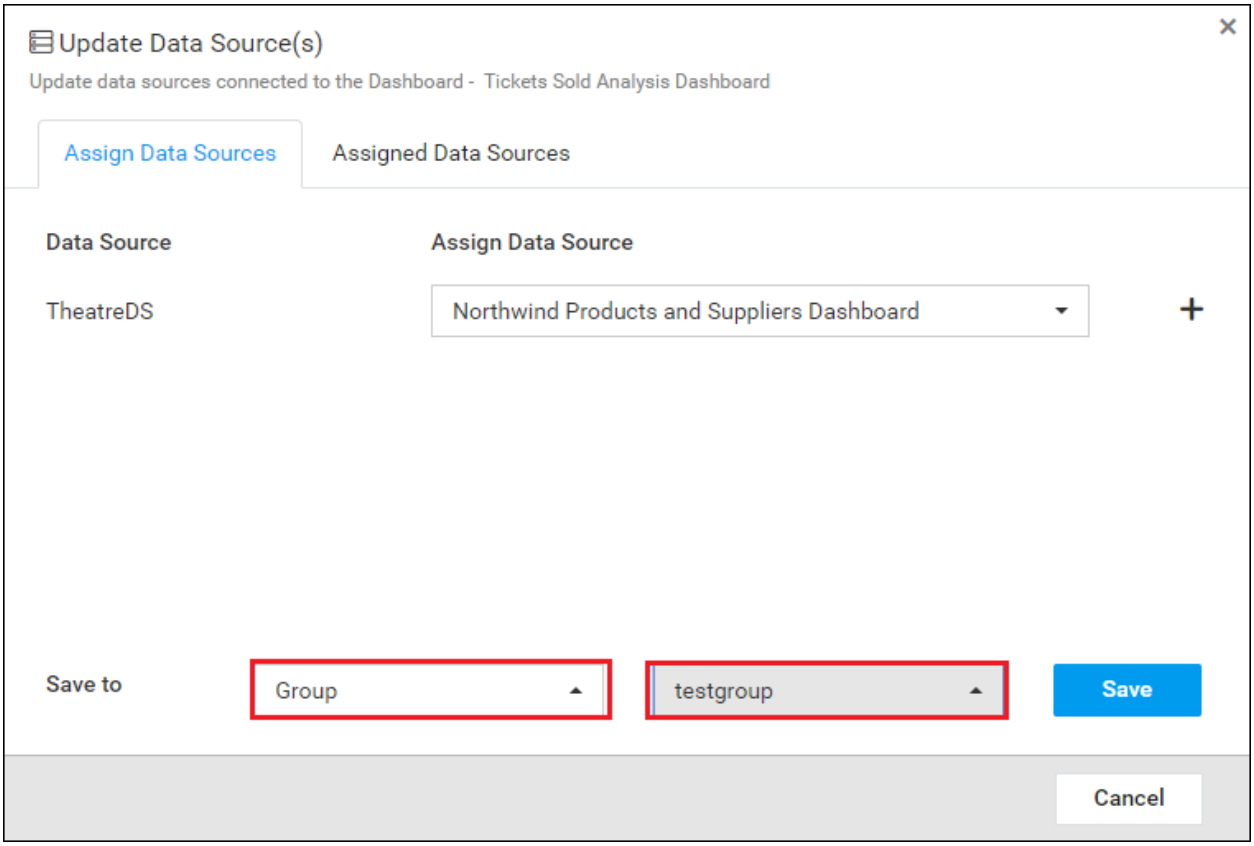
Dynamic Data Source Mapping

This can be achieved by assigning data sources for groups. The data source can be loaded dynamically based on the group of the current user.

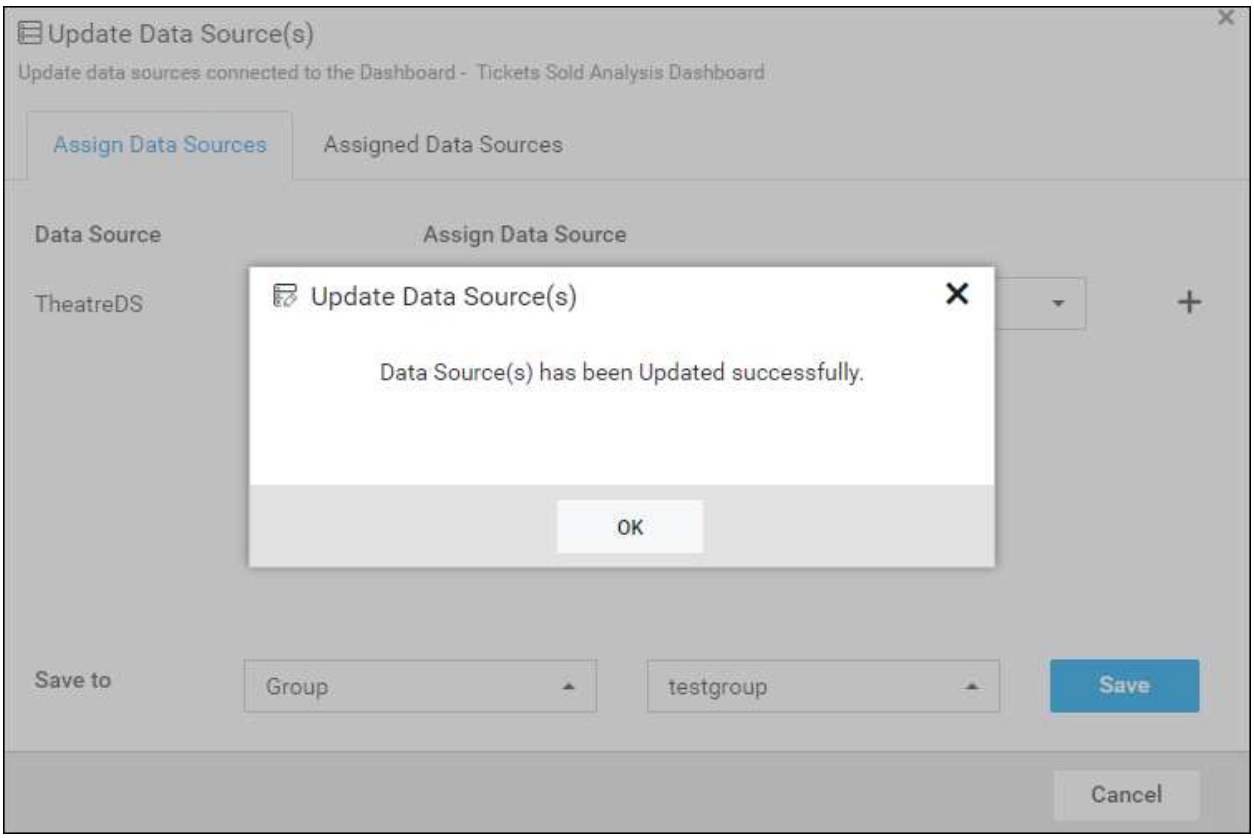
1. If you select, All users option, then the data source you have added will be shared to all the users.



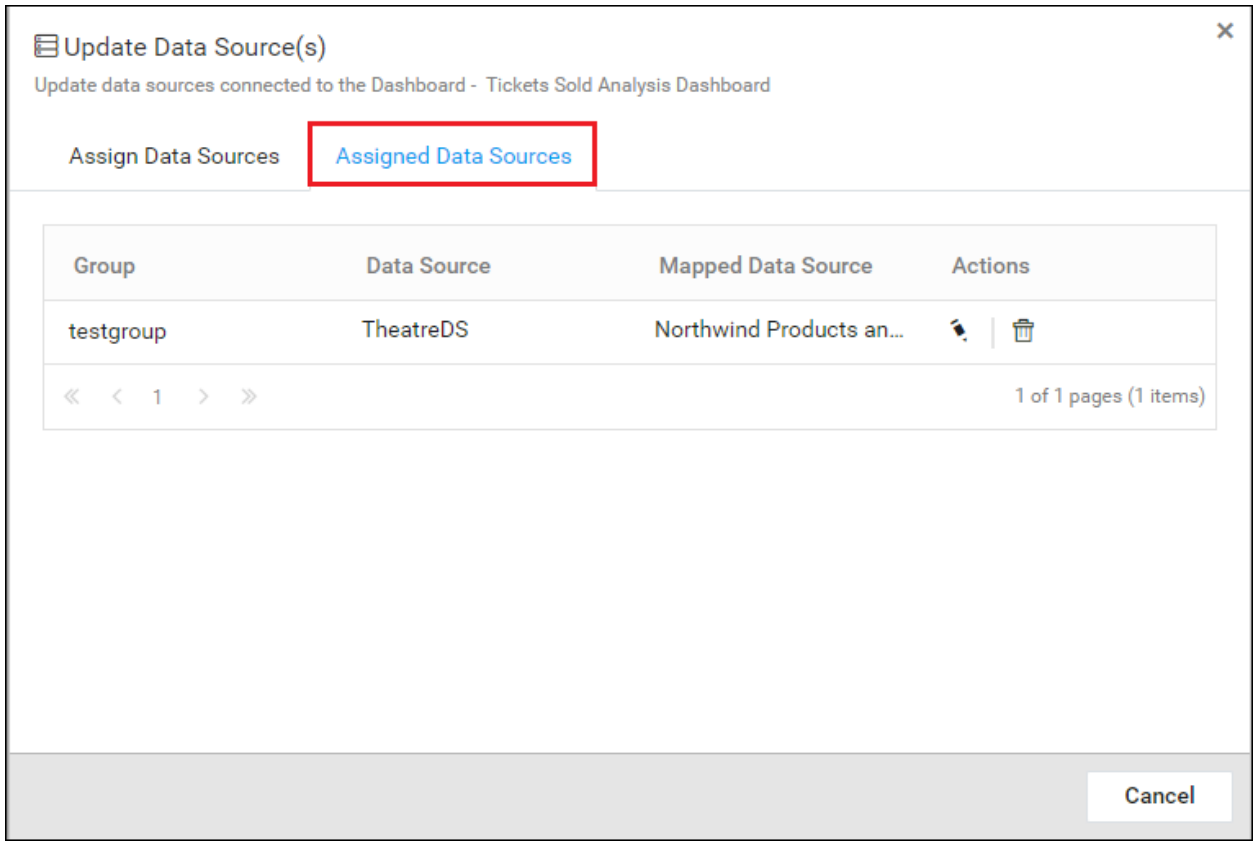
2. If you select Group option, then you can see now an another dropdown which list all the active groups. From the dropdown you can map the data source to the group.



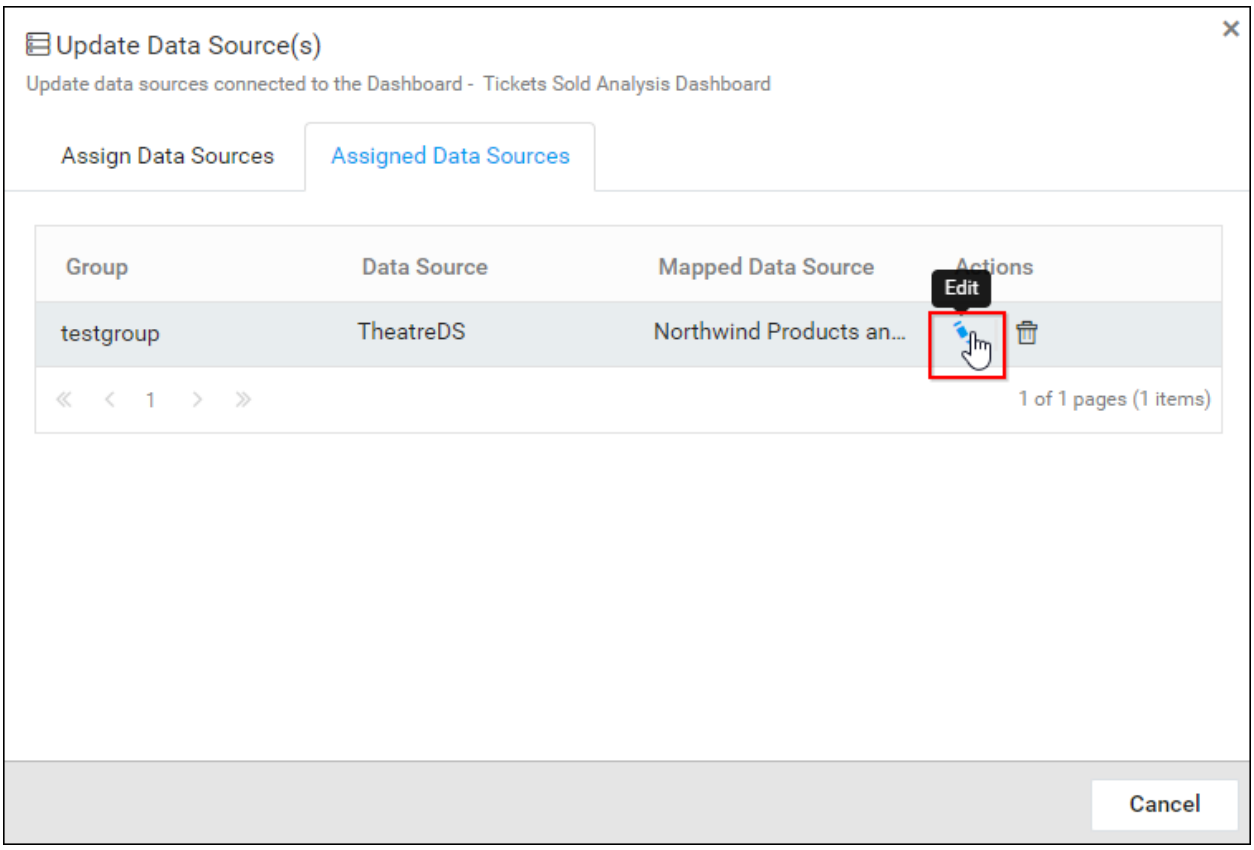
3. Now click the save button to save the assigned data source for groups.Now the Dashboard will be loaded based on the the data source assigned to the group of the current user.



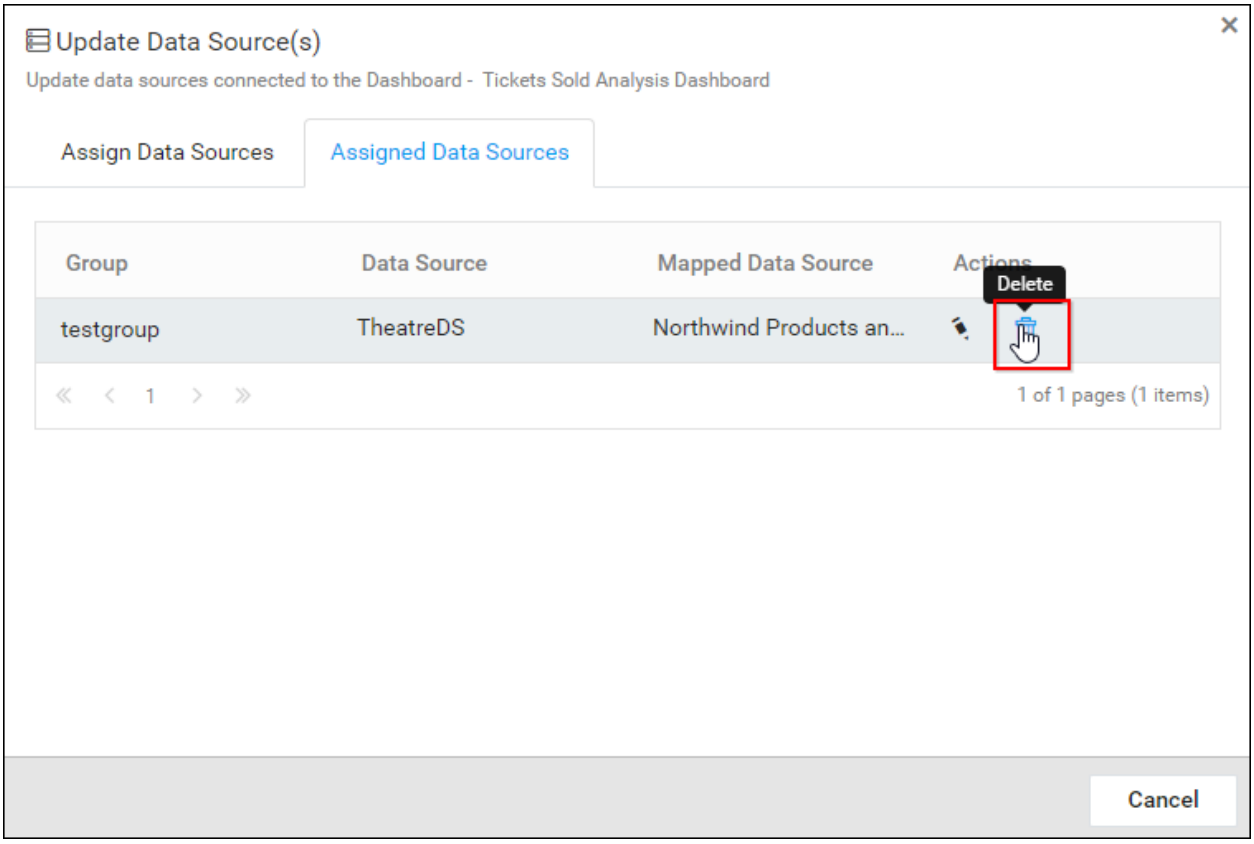
3. Then the selected data source will be mapped with the particular group(s).You can see assigned groups in an assigned data sources tab.

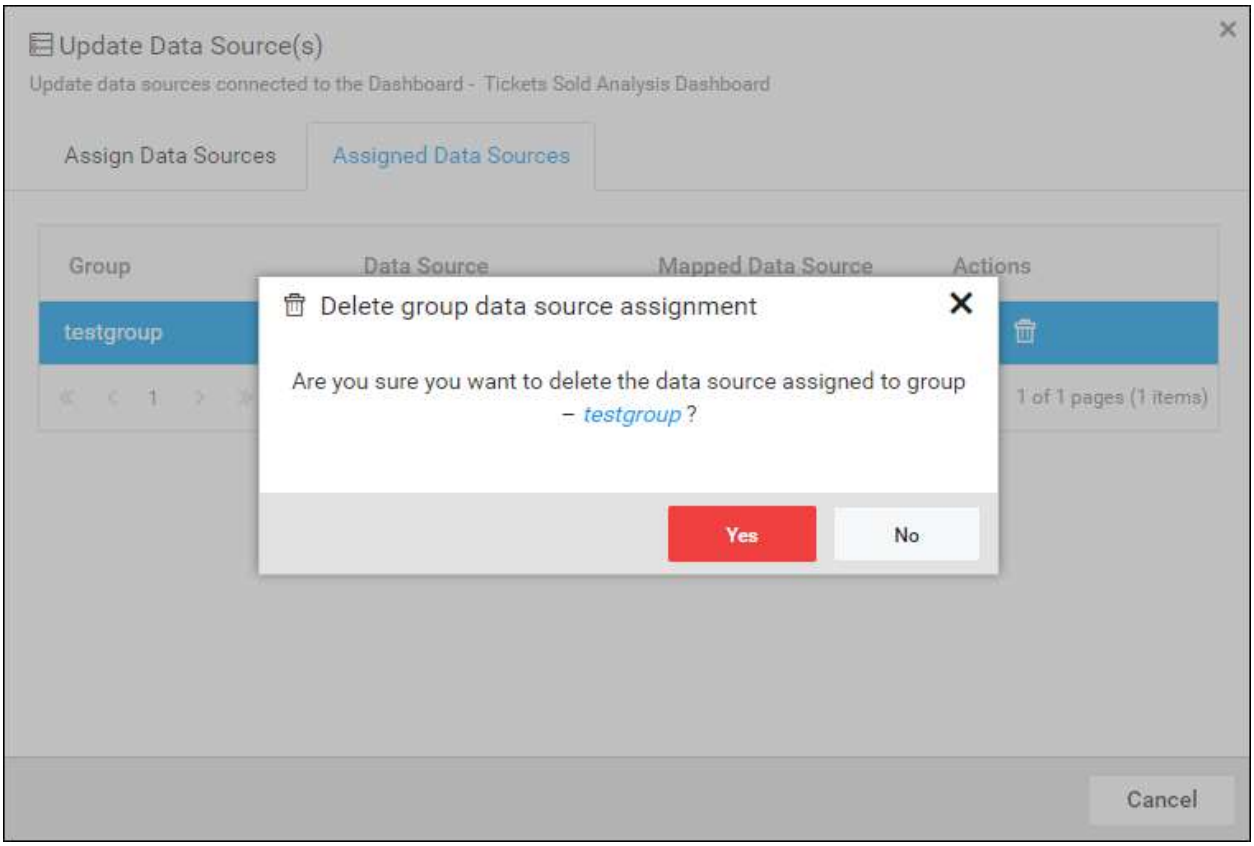


4. You can edit the assigned data source for that particular group, using the Edit button, it will get you back to assign data source tab content.



5. You can delete the assigned group for a data source(s),using the Delete button it will delete the assigned group and the group permission.



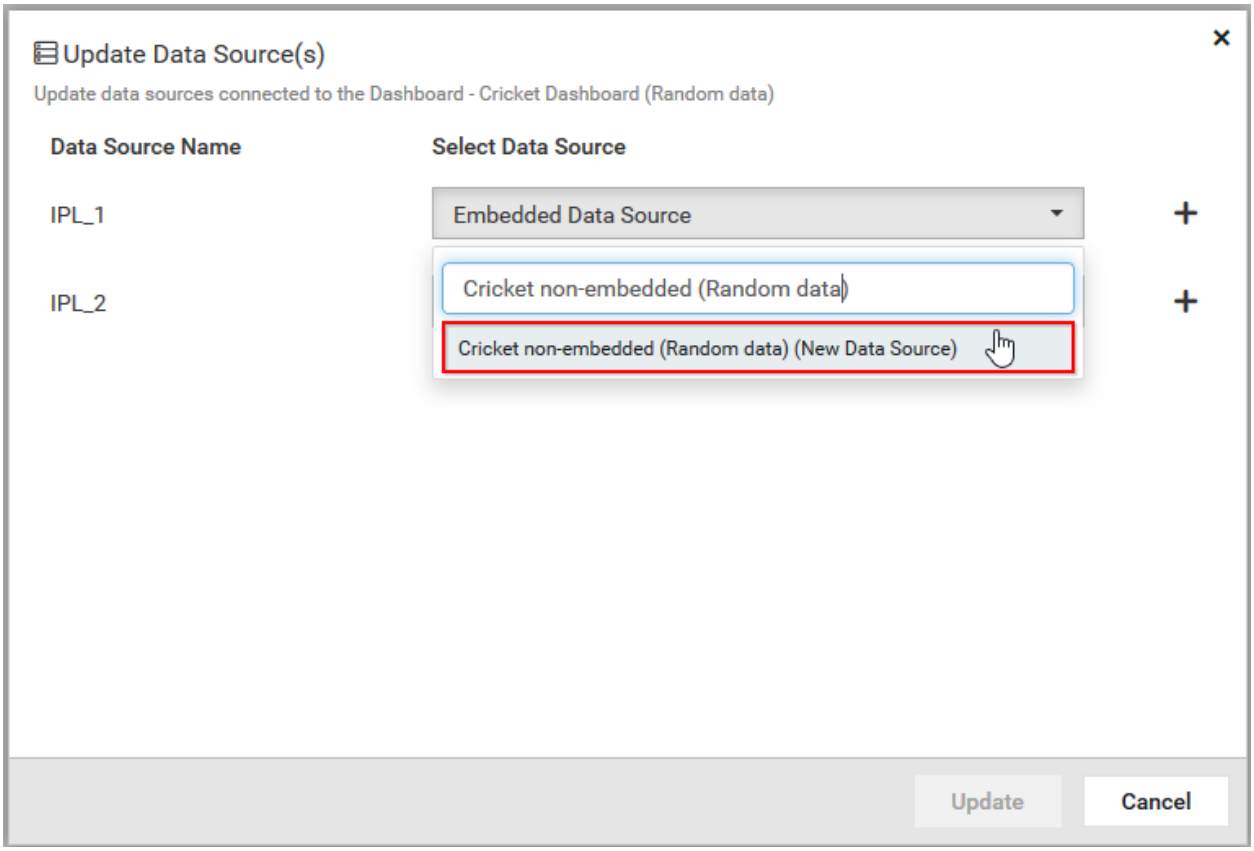


Note: If the shared data sources does not have permission for the current user, then the dashboard will not load.

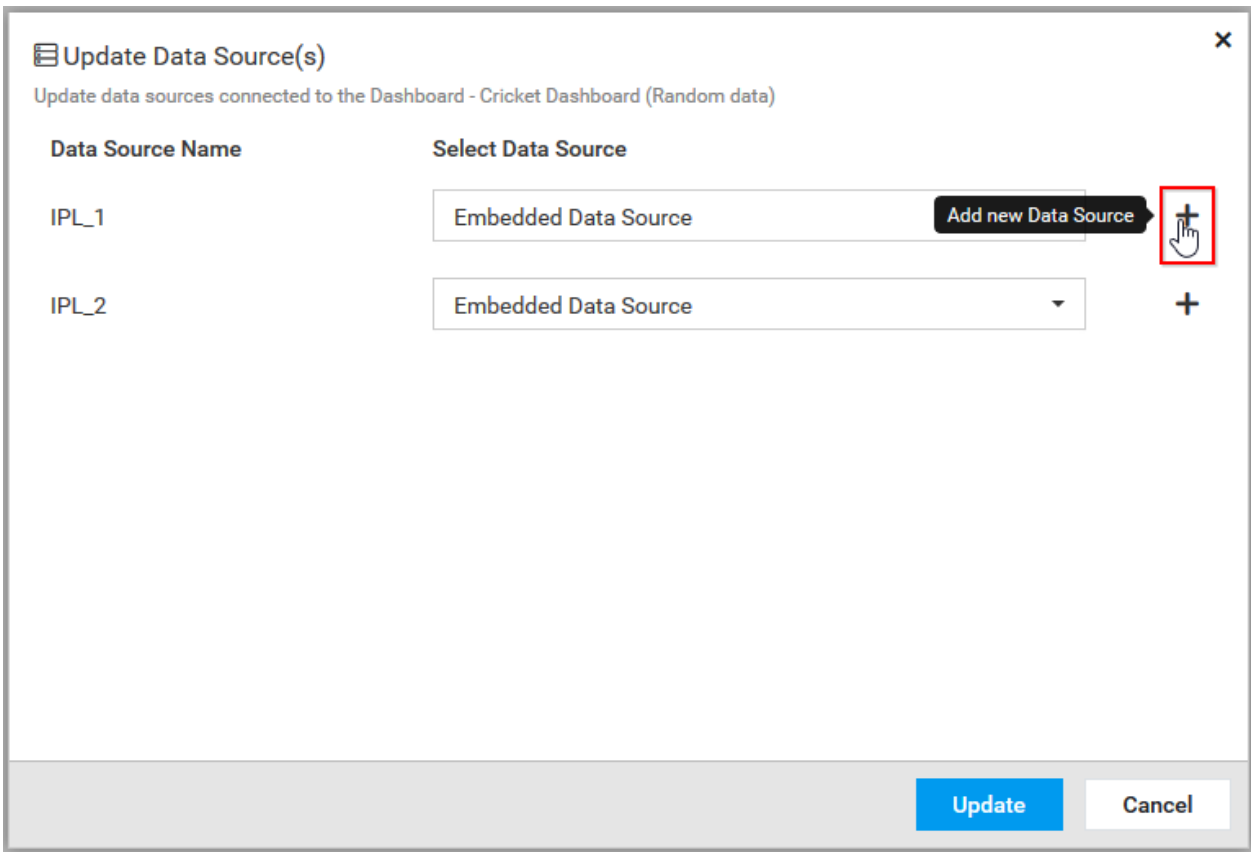
Add new Data Source

New Data Source can be added by two ways,

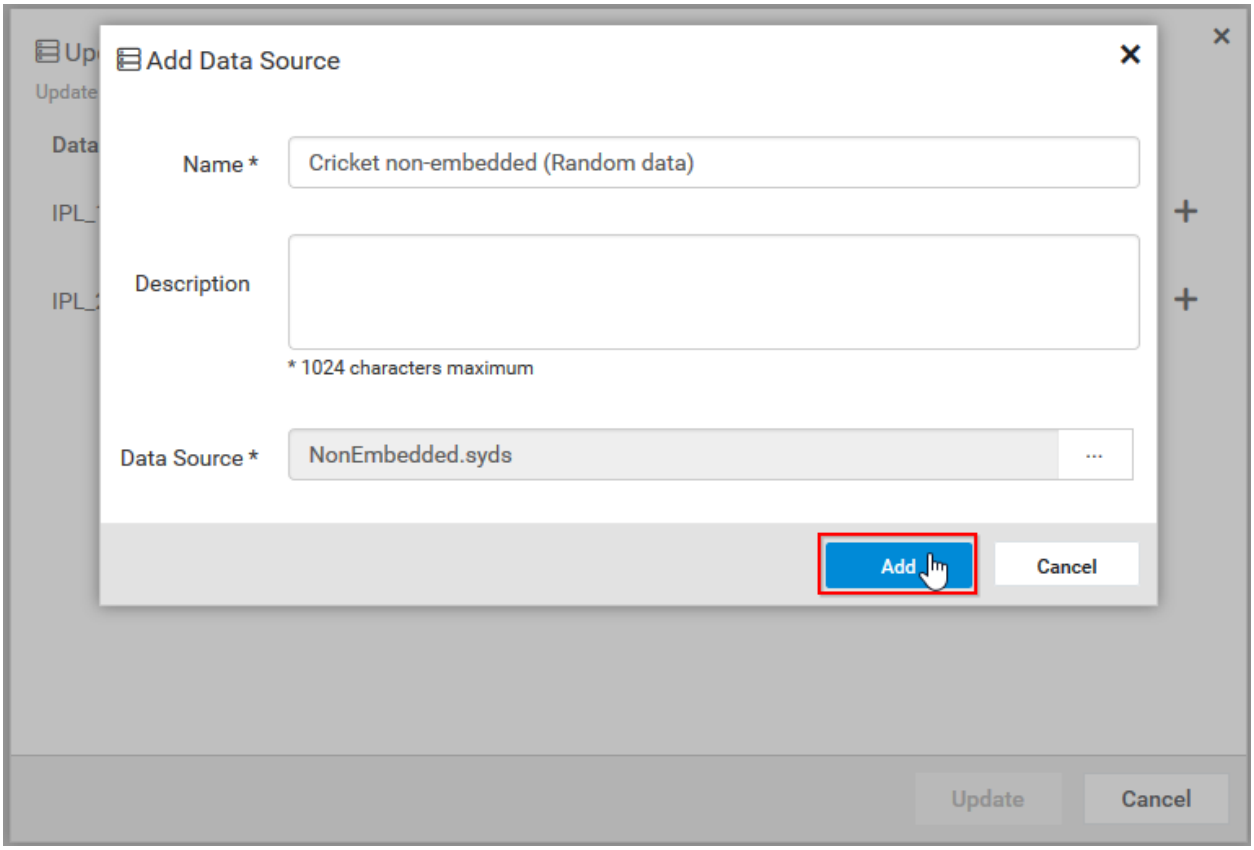
1. By searching with the Data Source names in the drop down. It suggests to add new Data Source, if the searched Data Source is not in the list. On clicking the (New Data Source) option opens the new Data Source dialog box.



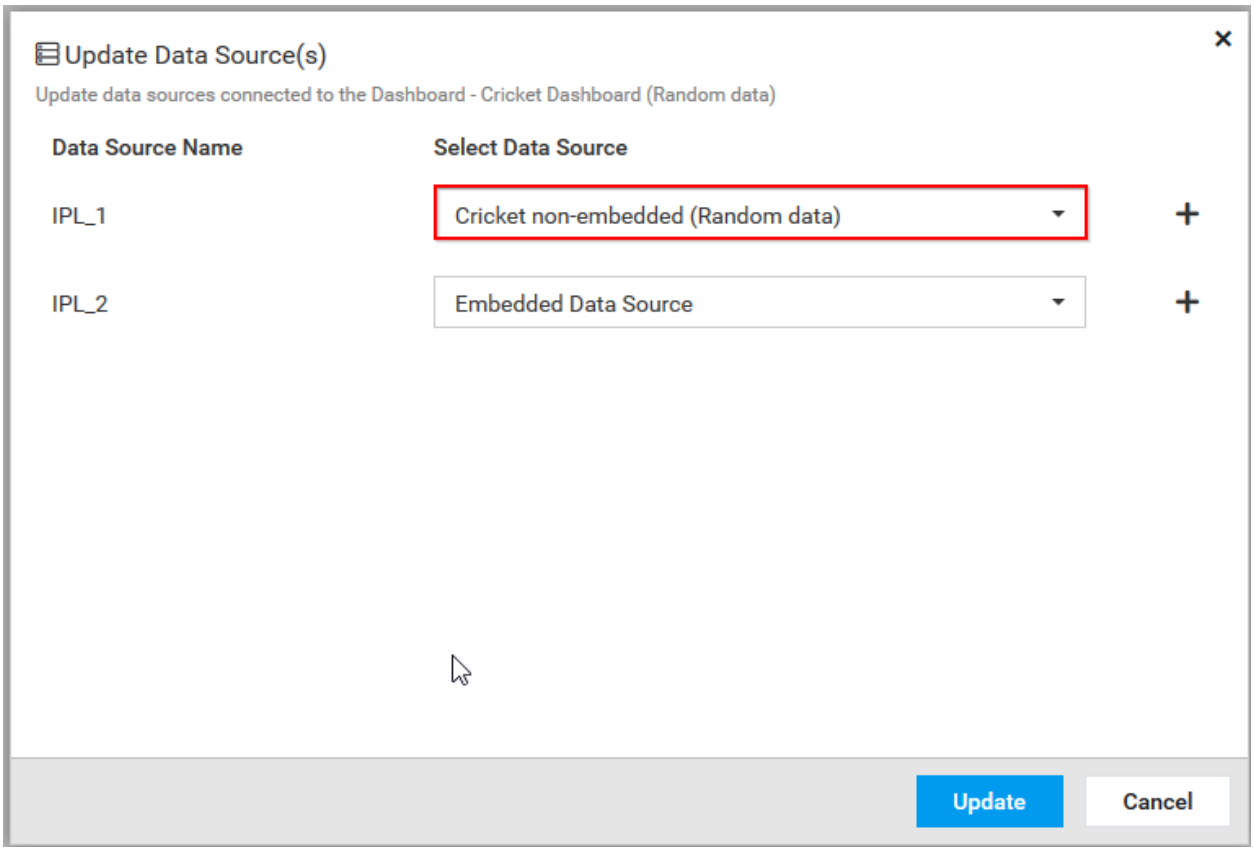
2. Click the plus (+) icon to open the new Data Source dialog box.



Select a Data Source file and add it.



The newly added Data Source is automatically selected for the specific Data Source Name.

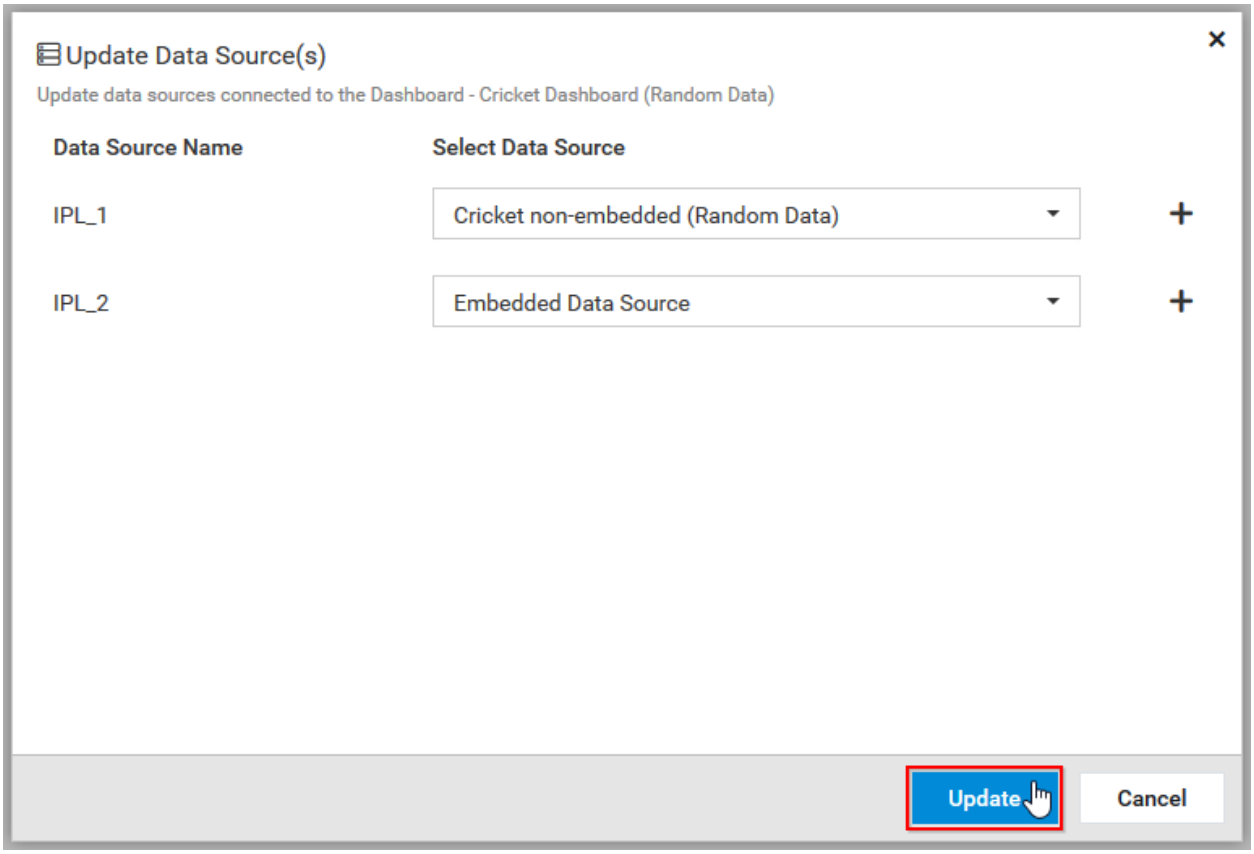


On clicking the update button, the shared Data Sources are updated for the specific dashboard.

Here,

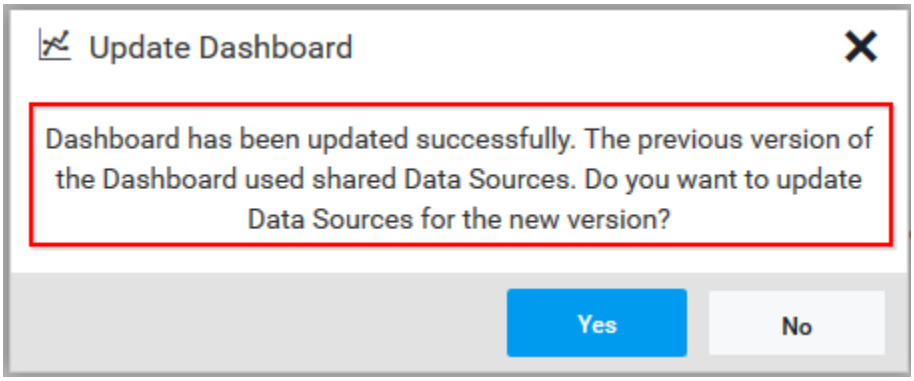
Data Source Name IPL_1 uses non-embedded Data Source.

Data Source Name IPL_2 uses embedded Data Source.



Update Dashboard with Shared Data Sources

On updating the version of the Dashboard with the Shared Data Source, a message box is displayed like below,



On clicking Yes, displays update Data Source dialog for current version of Dashboard.

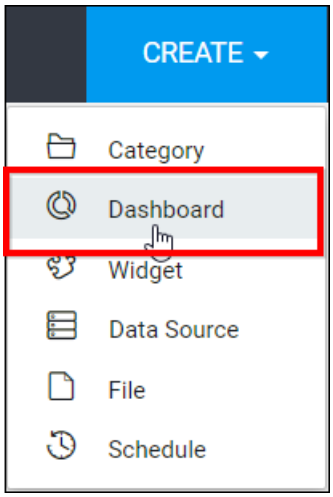
On clicking No, the embedded Data Source is used for the current version.

Multi-Tabbed Dashboards

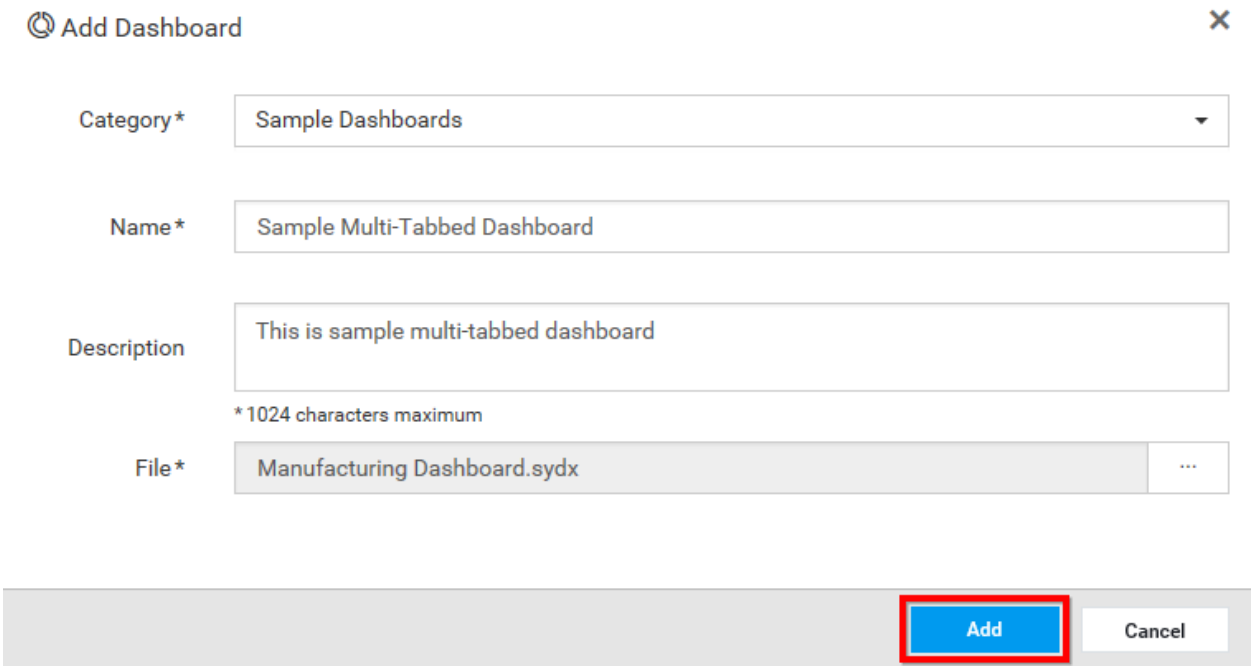
This section explains about the Multi-Tabbed Dashboards in the Syncfusion Dashboard Server.

Add Multi-Tabbed Dashboards

1. Click on the Create button in the menu and select Dashboard to add a Multi-Tabbed Dashboard.



2. Select a category for the Dashboard and fill in the name and description of the Dashboard and upload the Multi-Tabbed Dashboard file(.sydx) in the Add Dashboard dialog box.



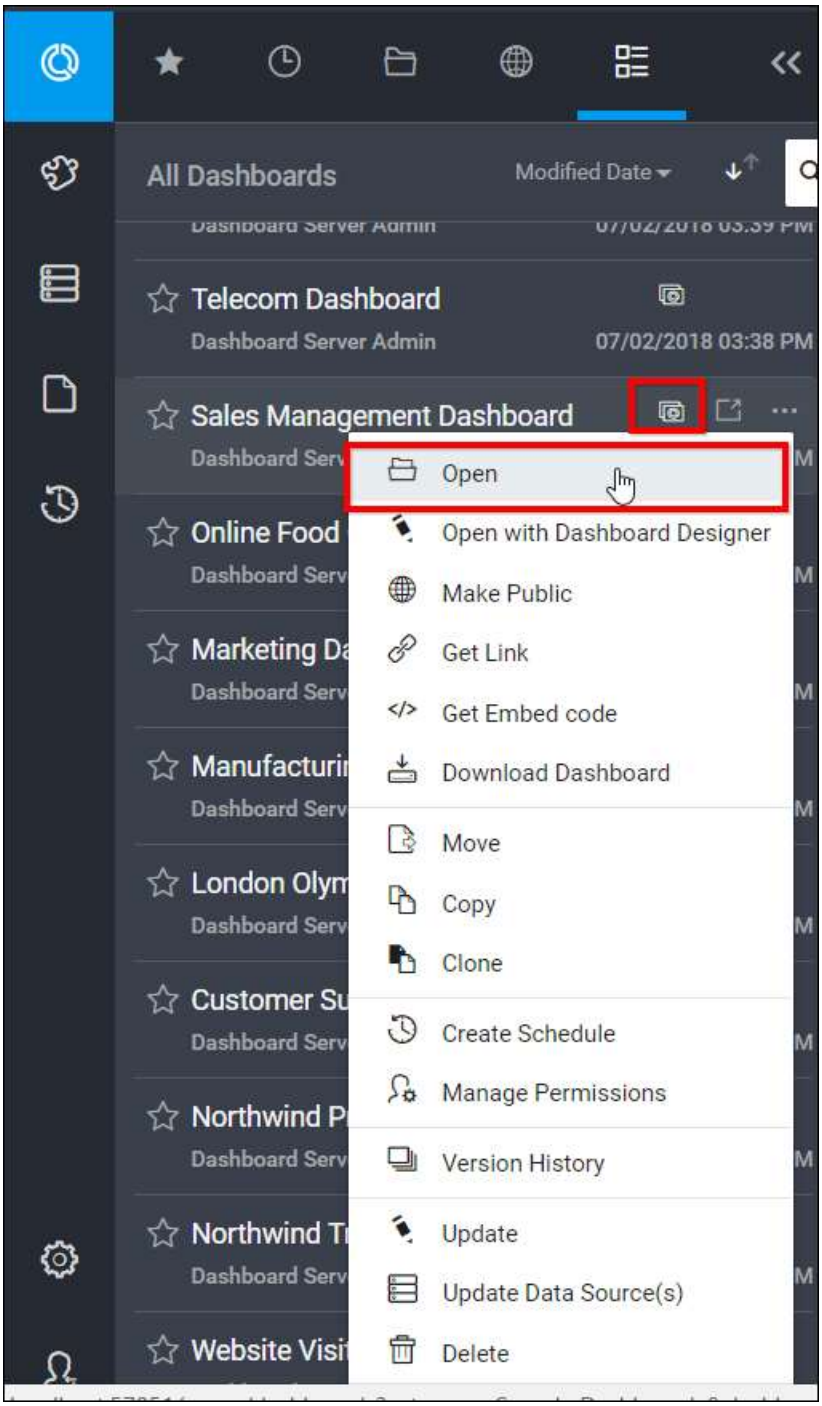
3. After filling the form, the Dashboard can be saved to be added in the Dashboard server.

Note: Read Write Delete permission for that Specific Dashboard will be added for the user who created the Dashboard.

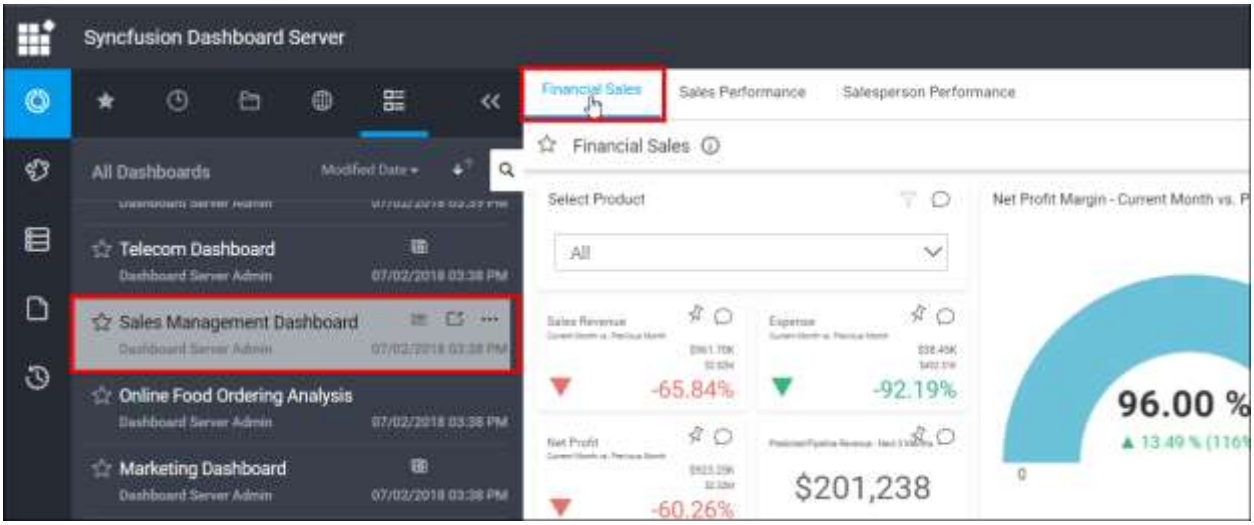
Multi-Tabbed Dashboards are represented by an icon to easily identify them.

Open Multi-Tabbed Dashboards

Click on the Dashboard Name which has Multi-Tabbed dashboard icon in the list to open it.



Dashboards are opened and by default, the first tab will be selected in the Multi-Tabbed dashboard and it can be exported in image, excel and pdf format.

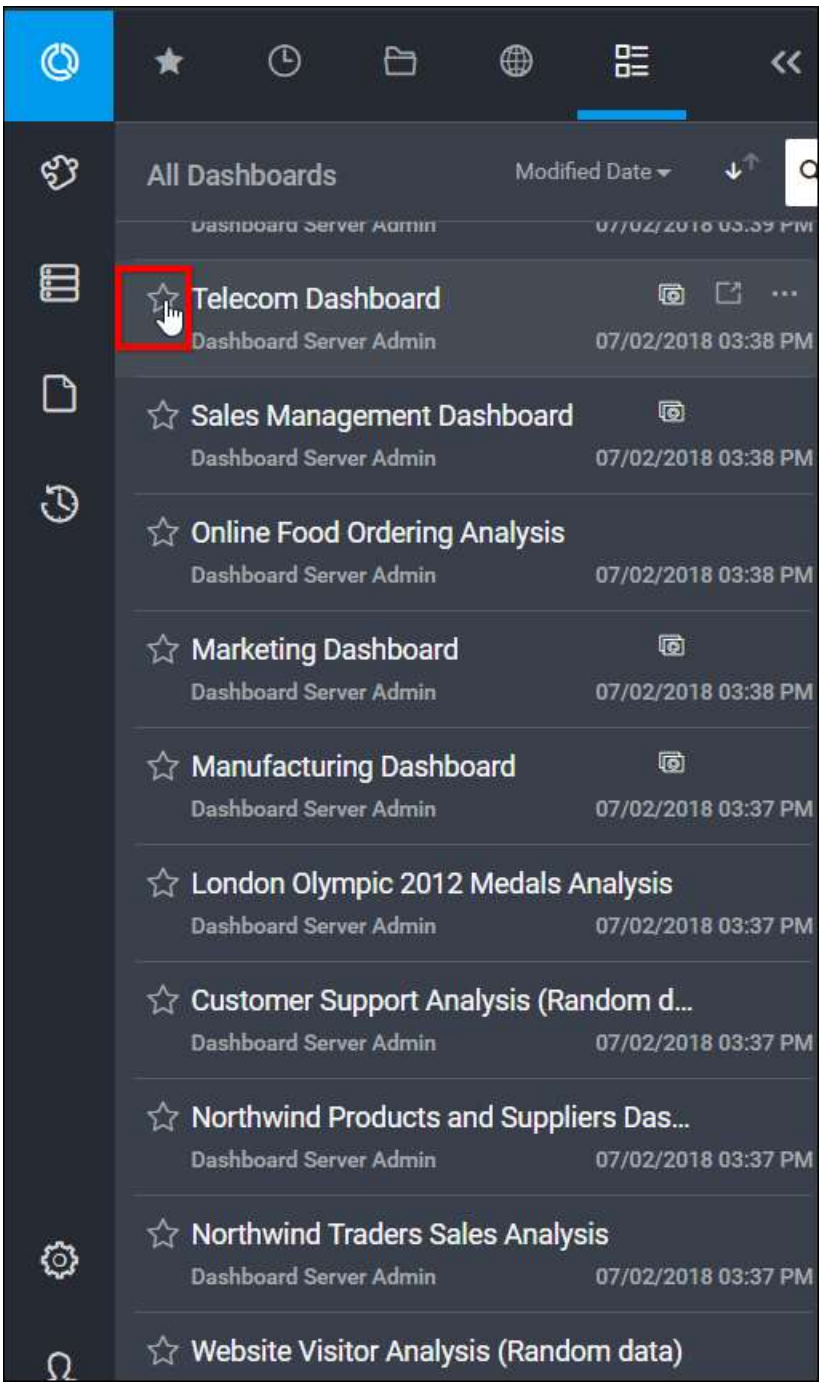


Mark as favorite

In Multi-Tabbed Dashboard, Dashboard and tabs also can be marked as favorite.

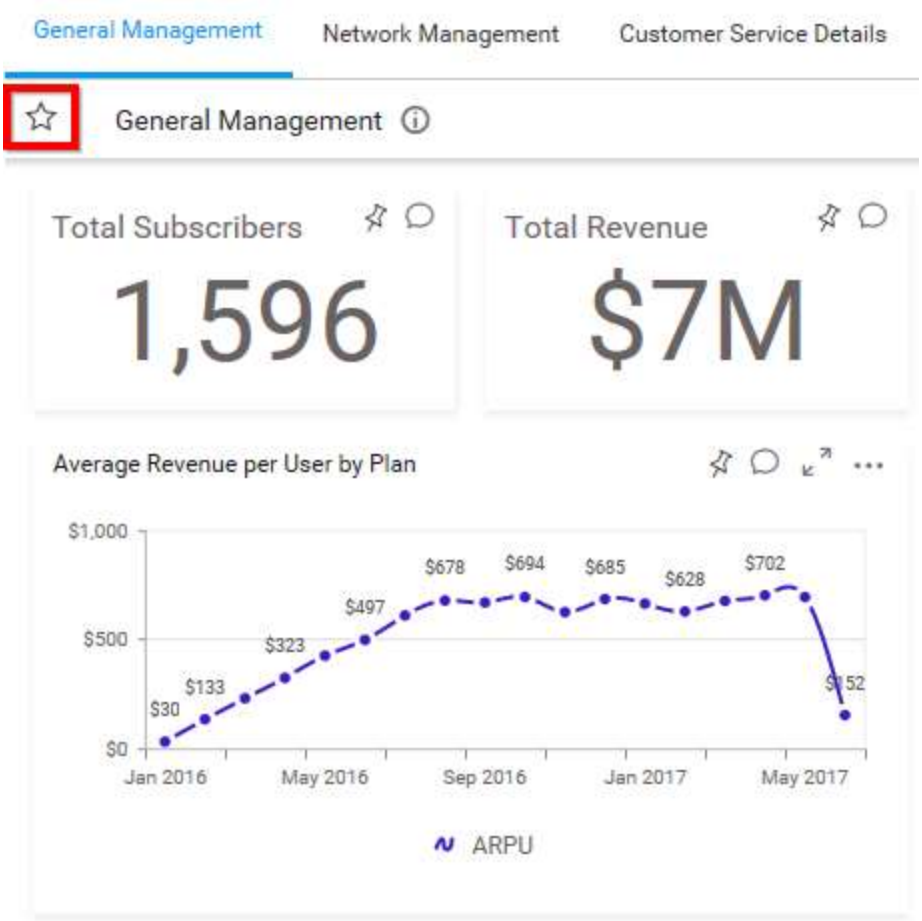
Mark Dashboard as favorite

The Multi-Tabbed dashboard can be marked as favorite from the Dashboard listing as below,

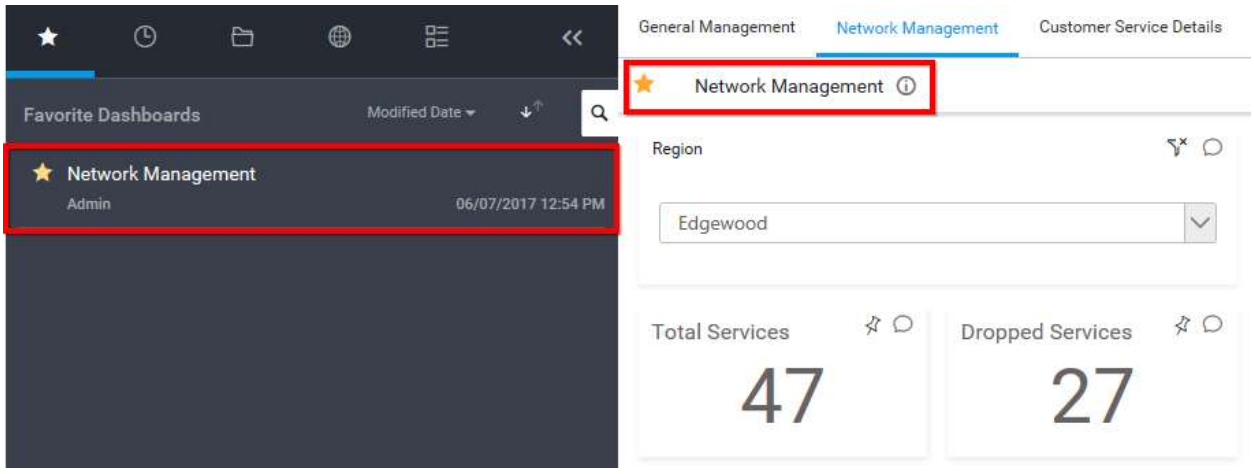


Mark Tab as favorite

Dashboards inside the Multi-Tabbed Dashboard can be marked as favorite as shown below,



After marked as favorite, the dashboards inside the multi-tabbed dashboards are listed in the favorite section as below,

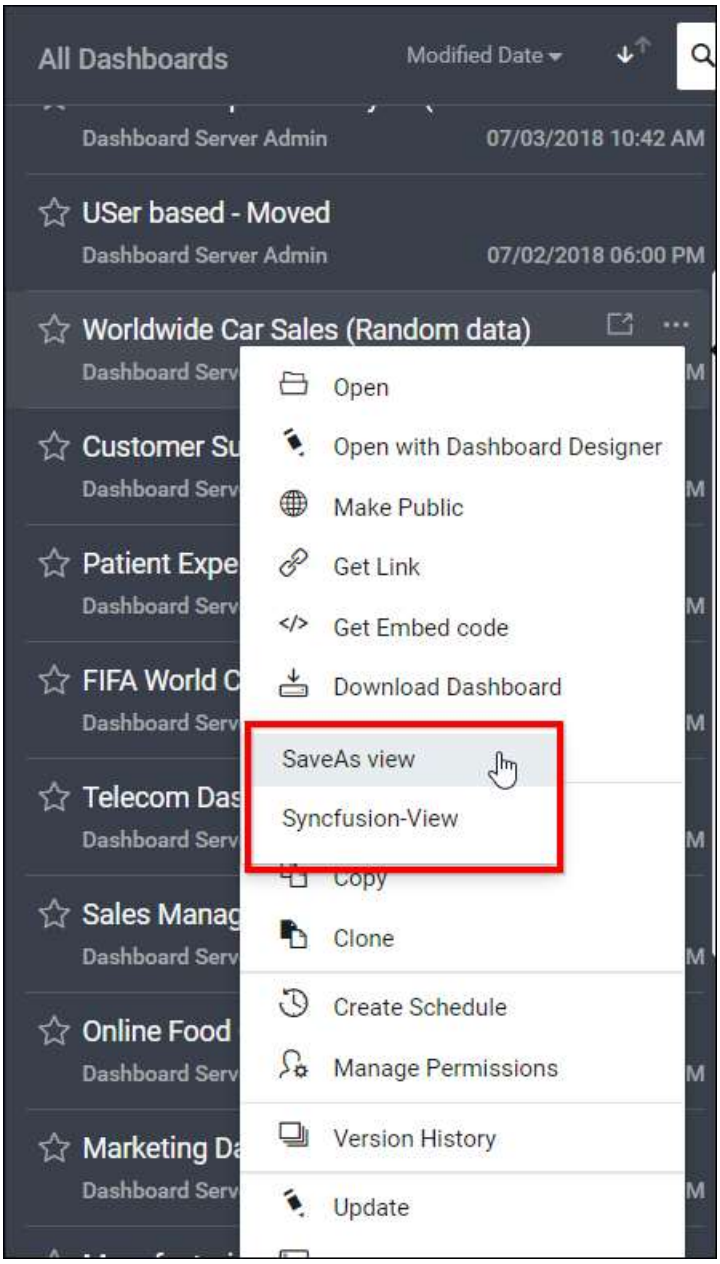


Click here to know more about how to create Multi-Tabbed Dashboard in Dashboard Designer.

Manage Dashboards Views

This section explains on how to open, add, update, share, delete Dashboard views in the Syncfusion Dashboard Server.

Dashboard Views that are accessible by the user depending on the user's permission is displayed in the Dashboards page.



Open Dashboard Views

Dashboard Views are opened in our embedded Dashboard Viewer itself as Dashboards.

Add Dashboard Views

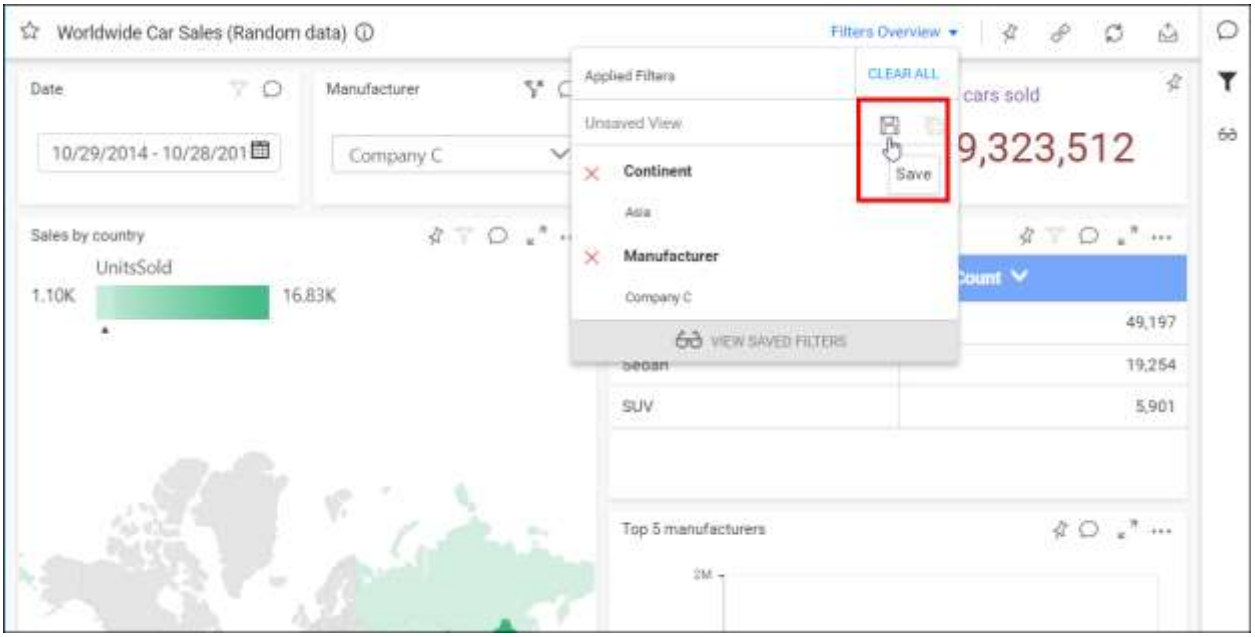
- If the user has Read Specific Dashboards permission, then the user can create Dashboard Views in any dashboard.
- The created Dashboard Views cannot be updated/deleted by other users.
- The owner of the Dashboard View only has share permission on the Dashboard View.

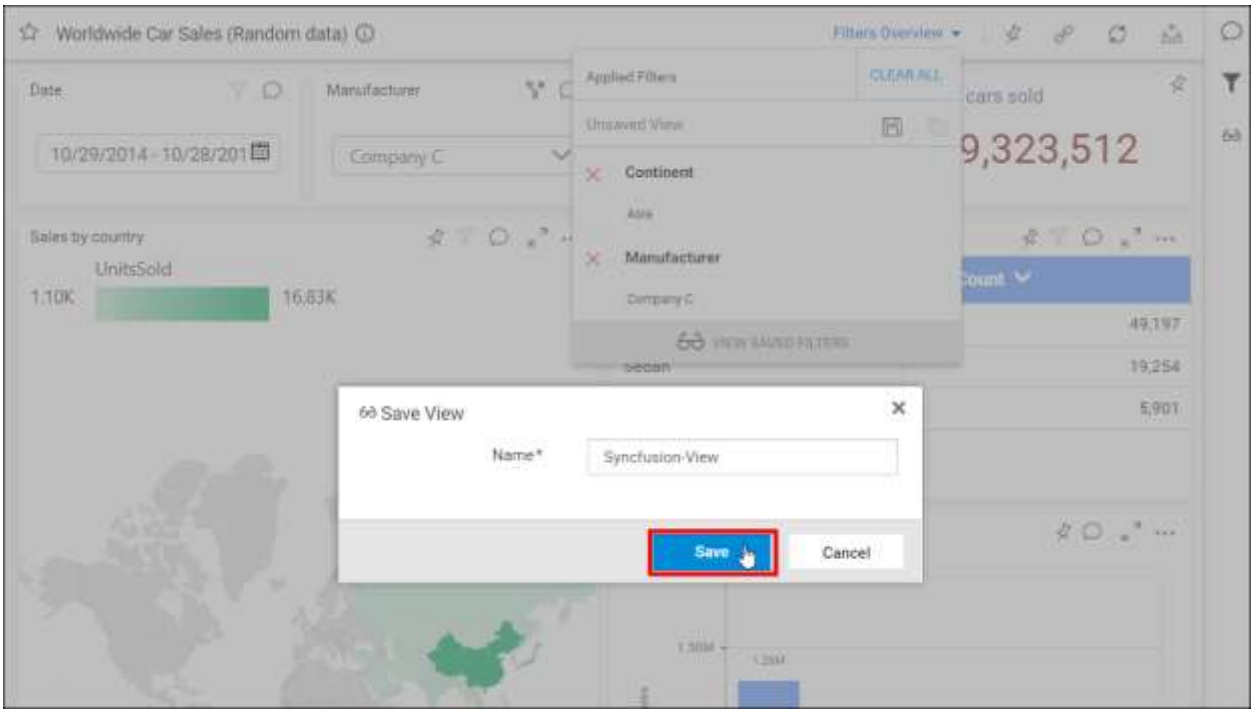
- The shared Dashboard Views can be modified and saved as another view as the users cannot modify the actual ones.

Steps to add a Dashboard View

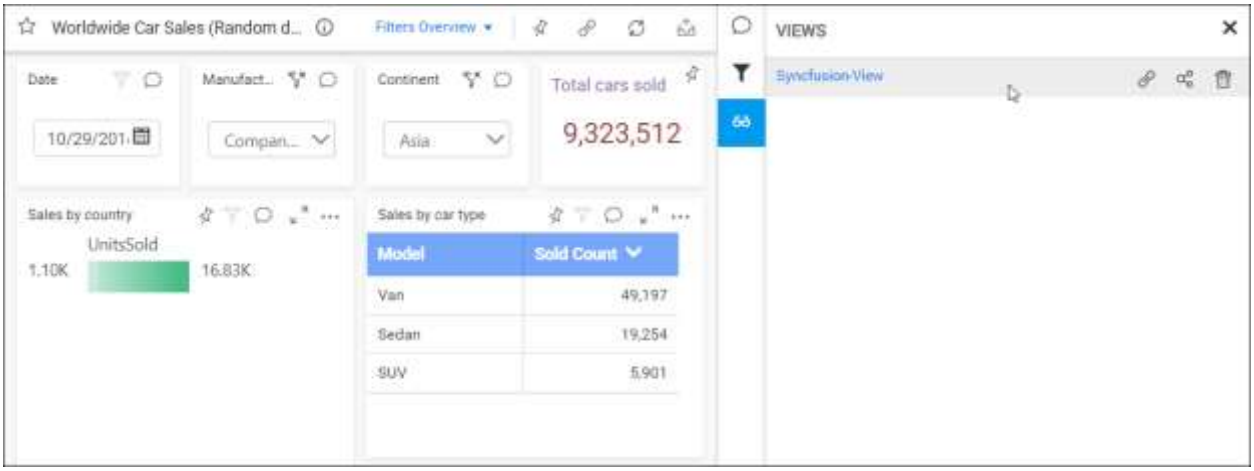
1. Filters applied in the Dashboards are summarized in the Filter Overview section under Applied Filters.

To save the filters, click on Save icon, a popup is opened to type the name for the Dashboard View and submit it.

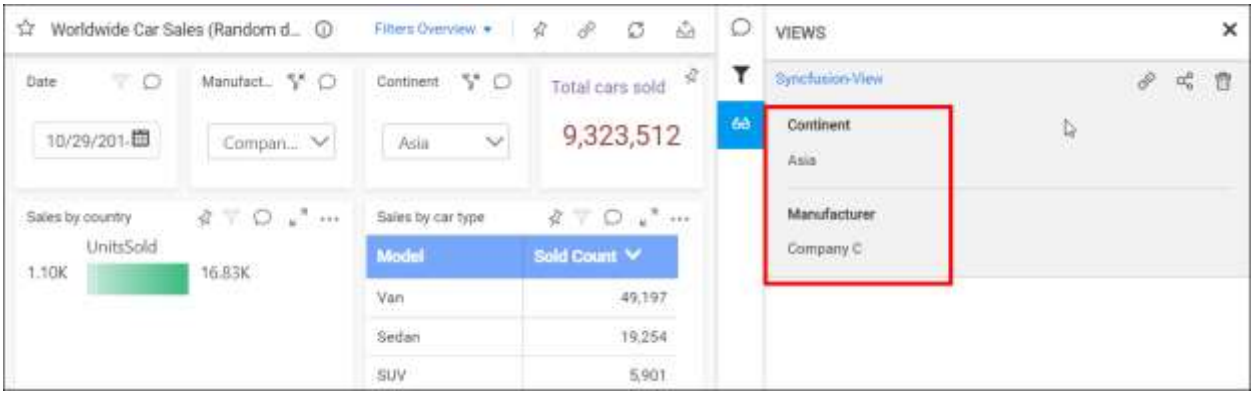




2. Saved Dashboard Views are shown in Dashboard Views panel under VIEWS.



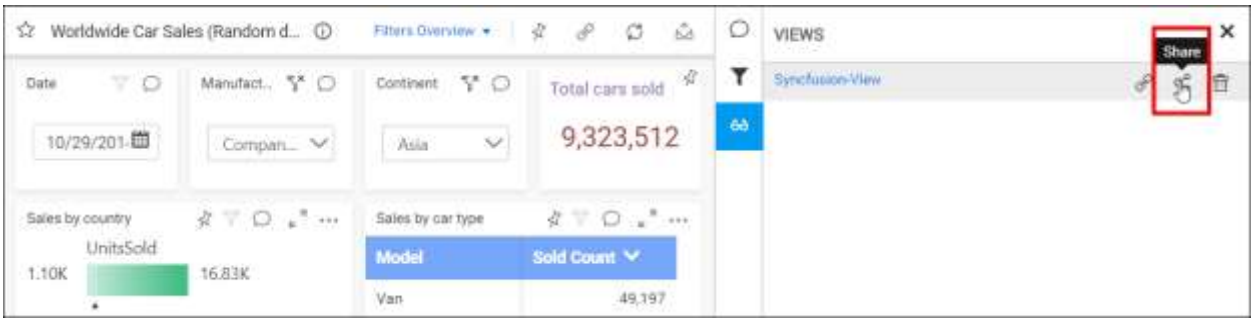
3. Click on saved views shown in Dashboard Views panel, filters applied in that view is displayed as accordion.



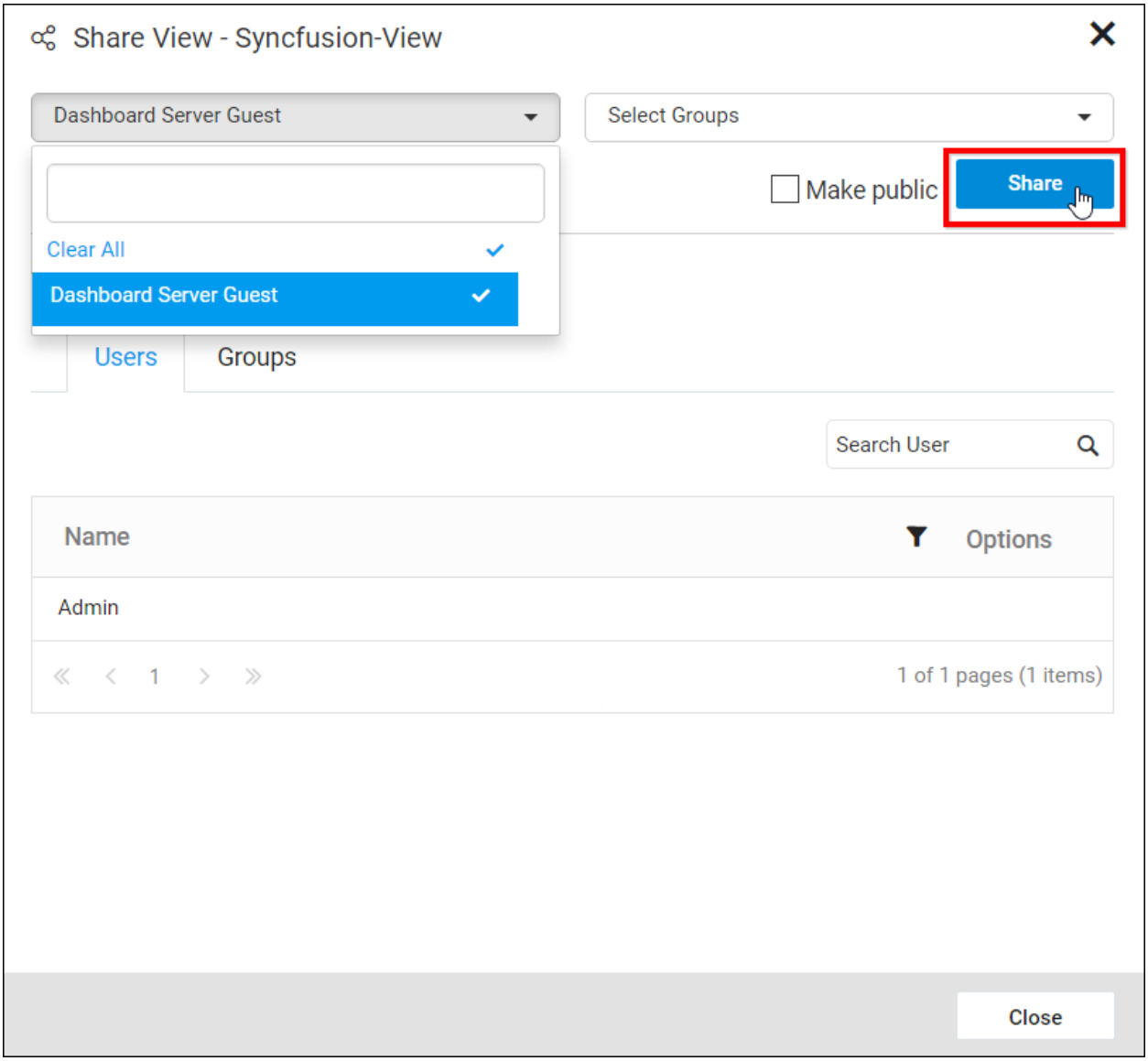
Note: Read Write Delete permission for the Specific Dashboard View is added to the users by the person who created the Dashboard View.

Share Dashboard Views

Dashboard Views can be shared to the users, who have permission for the Specific Dashboard.

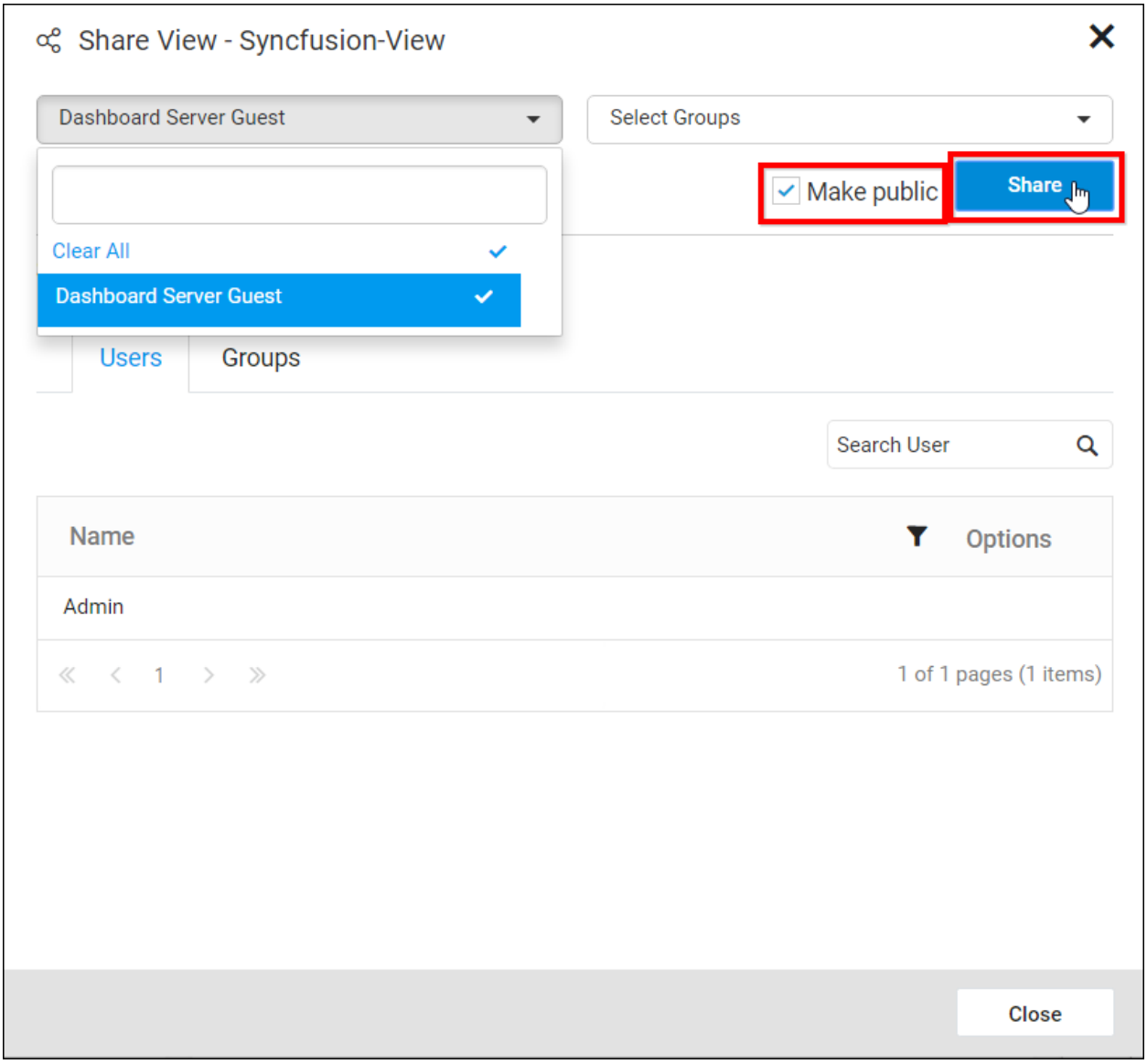


Choose the users and groups from the dropdown and click on Share button.



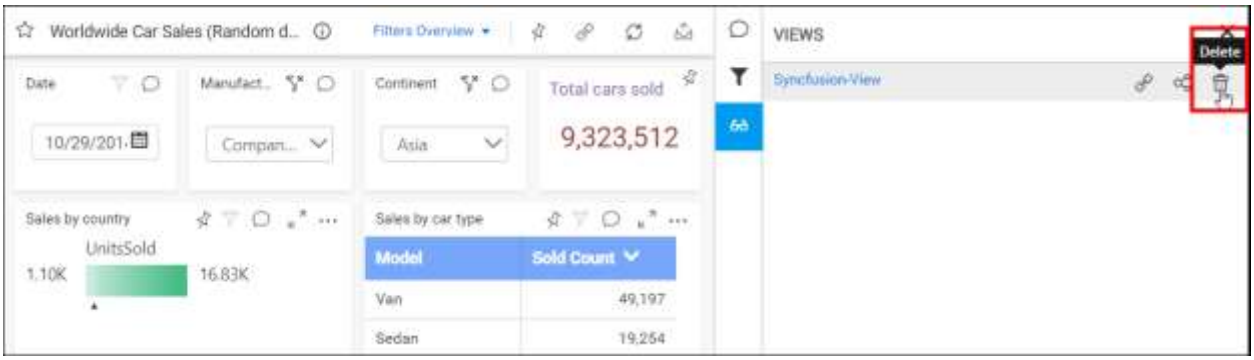
Make Public

Dashboard Views can be marked as public to let anyone access it.



Delete Dashboard Views

Choose the Delete option to delete the Dashboard View from Dashboard Server.

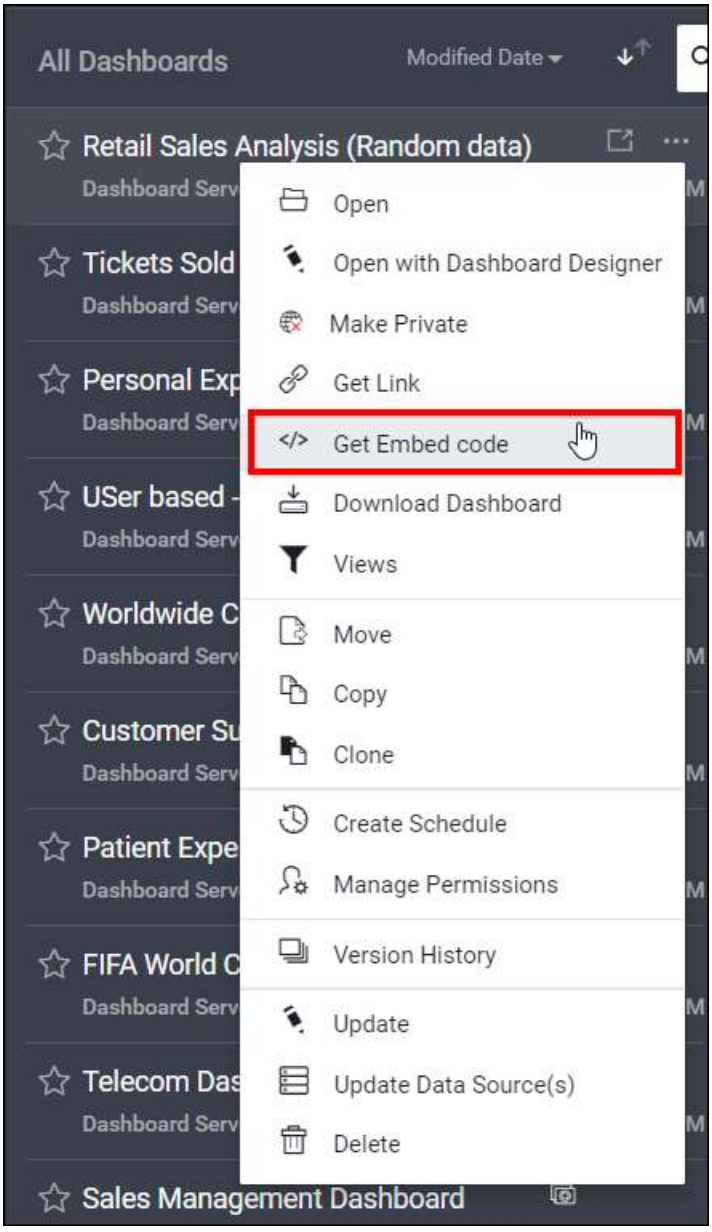


Embed Server Dashboards

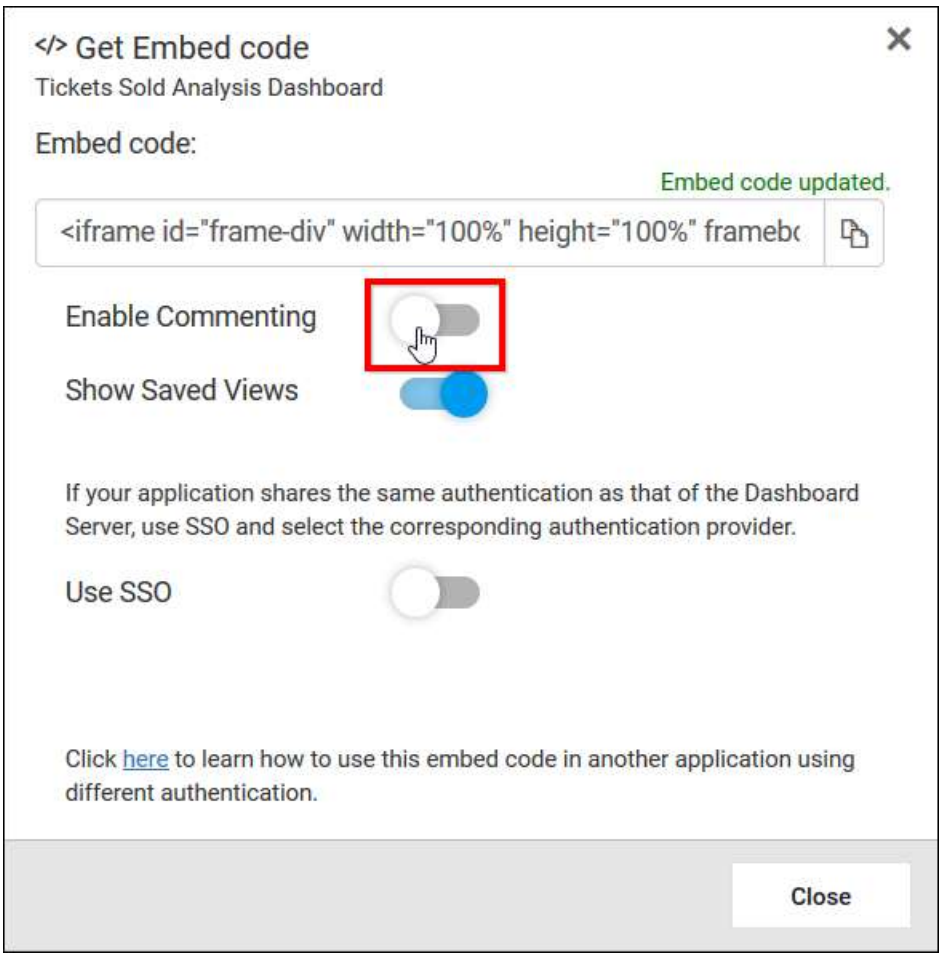
Embed Server Dashboards provides option to render the dashboards into other web applications. Follow the below steps to embed server dashboards into other web applications.

Getting Embed URL

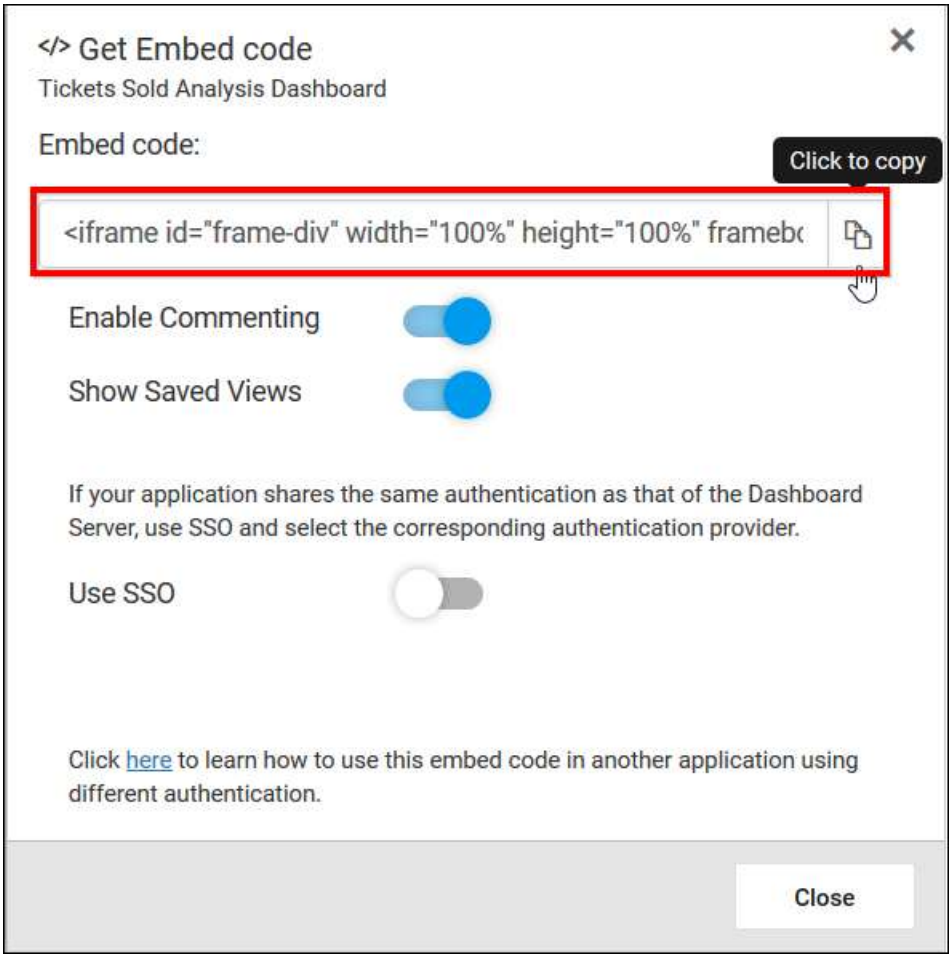
1. Click the Actions button in the dashboards grid context menu and select Get Embed code of the corresponding dashboards to embed.



3. By Default, Saved Views and Comments are enabled. You can able to disable the Saved Views or Comments by sliding the Saved Views or Comments button as below.



2. Copy the embed URL by clicking the copy icon from the Get Embed code dialog.



Embed using Server Authentication

Embed the copied URL into other web application with the help of the following code snippet.

HTML

```
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.1/jquery.min.js"></scri
pt>
<body>
<iframe id="frame-div" width="100%" height="100%" frameborder="0"
src="http://localhost:50290/dashboards/embed/Sample%20Dashboards/Customer%20
Support%20designer
%20publish?hascomments=true&hasviews=true&hassso=false"
onload="apiCall()"></iframe>
<script>
$(document).ready(function() {
jQuery.support.cors = true;
});
function apiCall() {
var dataValue = "";
var element = document.getElementById("frame-div").contentWindow;
var apiRequest = new Object();
var baseURL = "http://localhost:50290"; //Replace here with the launched
Dashboard Server site URL
```



```
apiRequest.validationkey = ""; //Find the Validation key in Machinekey node
of the Dashboard Server web.config file
apiRequest.decryptionKey = ""; //Find the Decryption key in Machinekey node
of the Dashboard Server web.config file
apiRequest.userid = "admin"; //Username/Userid of the user who has
permission to the particular Dashboard
apiRequest.password = "admin"; //Password of the user.Its optional when
validationkey and decryptionkey is used.
$.ajax({
type: "POST",
url: baseURL + "/api/get-user-key",
data: apiRequest,
success: function(data) {
dataValue = data.Token;
element.postMessage(dataValue, "*");
},
error: function(data) {
element.postMessage("", "*");
}
});
}
</script>
</body>
</html>
```

Please find the below changes that have to be done to render the dashboard into the web application.

1. Paste the copied Embed URL instead of the above iframe.
2. Add the apiCall() function invoke in iframe onload event.
3. Can able to get the token using either by passing the below three type of properties.

a. Username and Password

b. UserId, Validationkey and Decryptionkey

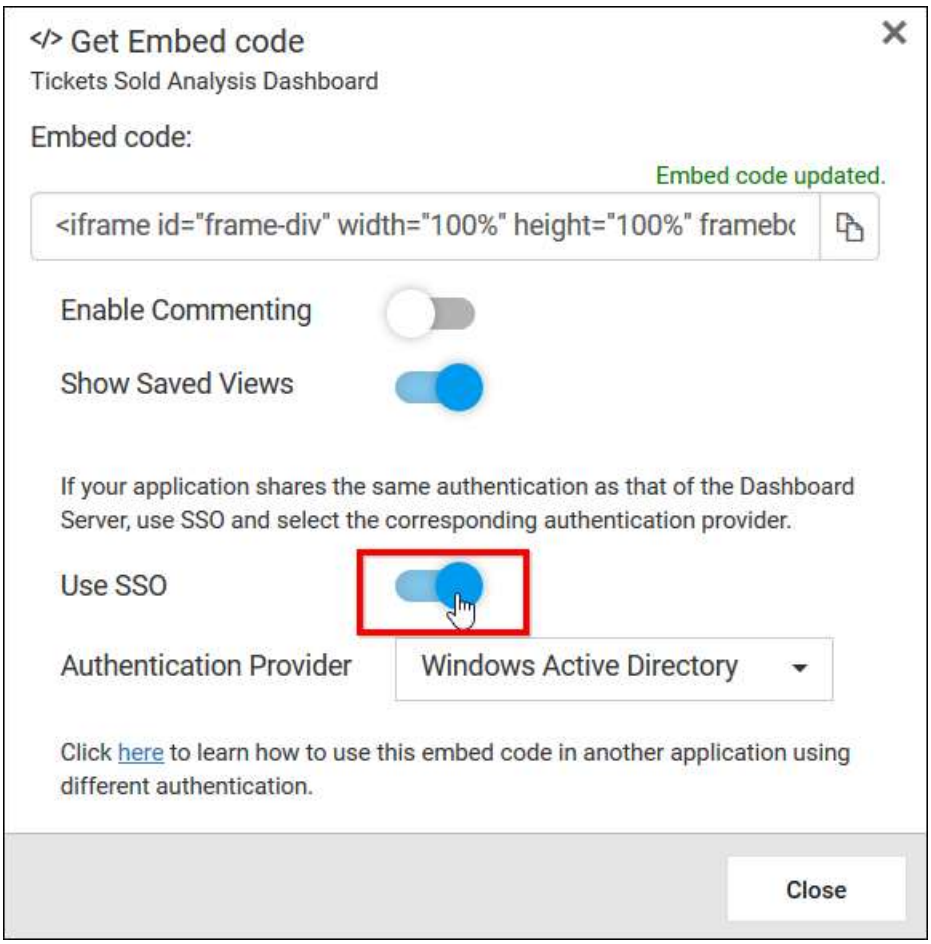
c. Username, Validationkey and Decryptionkey

Note: Public dashboards does not need token to authenticate with Dashboard server.

Embed using Windows AD Authentication

If the Dashboard server deployed into the machine which has configured with the Windows Active Directory, then follow the below steps to get the embed code with the Windows Active Directory SSO settings.

1. Click the Actions button in the dashboards grid context menu Get Embed code and enable SSO and select Windows Active Directory as shown below.

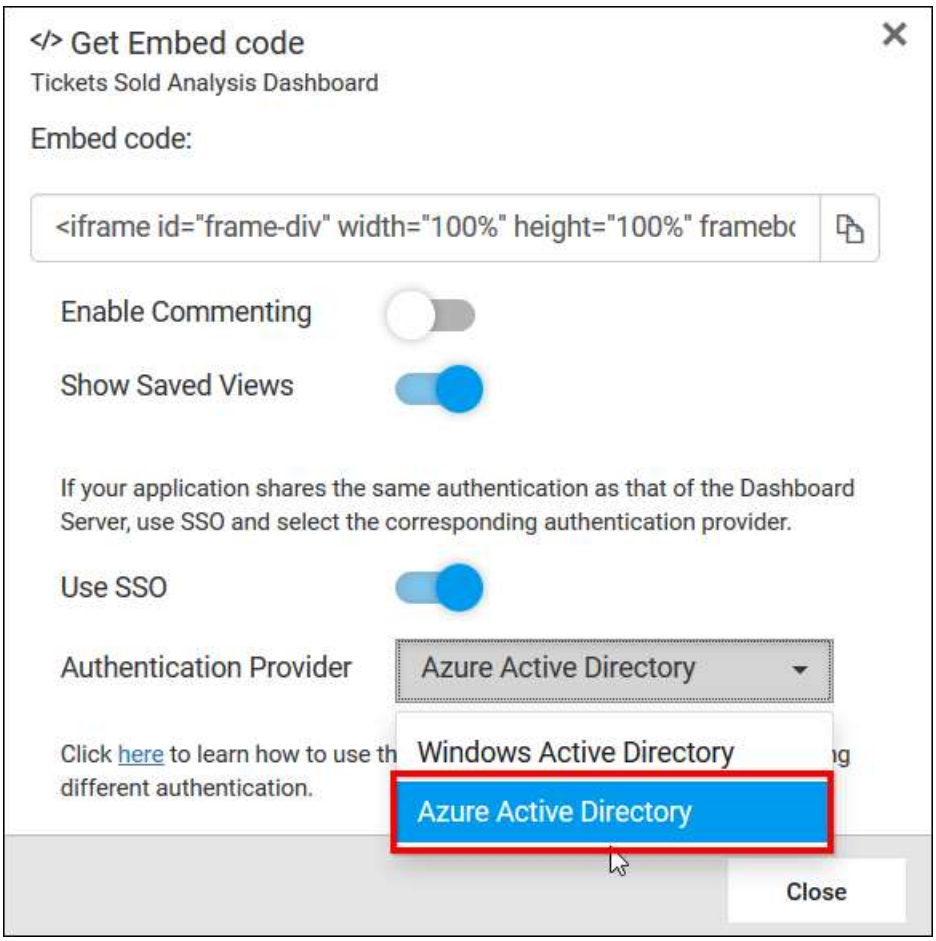


2. Paste the embed URL into an web application which has Windows Active Directory configuration.
3. Run the web application and the dashboards will render automatically.

Embed using Azure AD Authentication

If the Dashboard server deployed into the machine which has configured with the Azure Active Directory, you can follow the below steps to get the embed code with the Azure Active Directory SSO settings.

1. Copy the embed URL from the dashboards context menu Get Embed code and enable SSO and select Azure Active Directory as shown below.



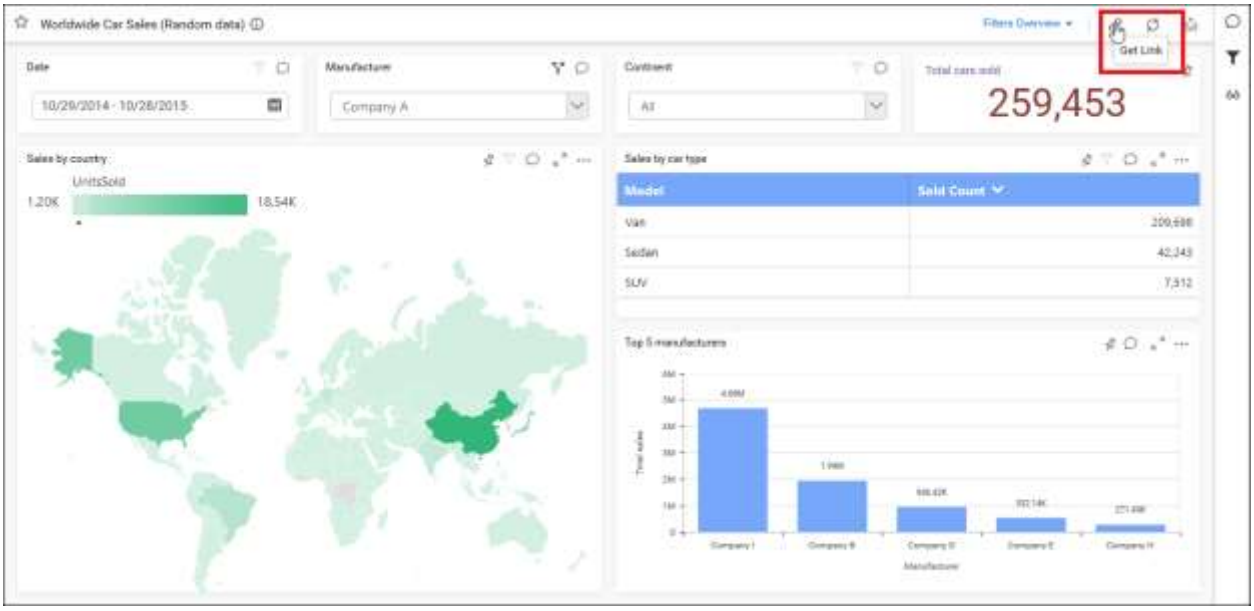
2. Paste the embed URL into an web application which has Azure Active Directory configuration.
3. Run the web application and the dashboards will render automatically.

Share filtered dashboards without save views

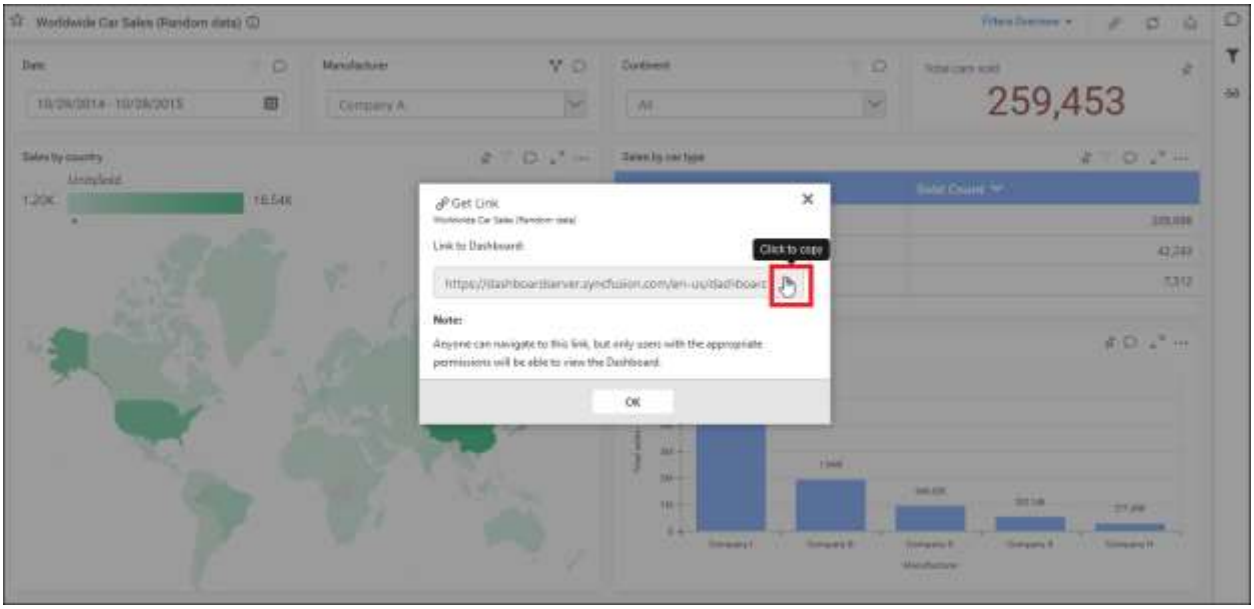
This section explains how to share the current dashboard filtered view without saving the filters as Dashboard Views.

Steps to be followed for sharing the filtered dashboard without saving it as view.

1. Open any dashboard and apply filters. The filters are written into browser history and you can navigate back and forth to apply or remove them.
2. To share the current state of the dashboard, click on the Get Link icon.



3. Get the URL of the dashboard from this dialog box. You can share the dashboard in its current state by copying the URL of the dashboard.



Manage Homepages

Add Homepage

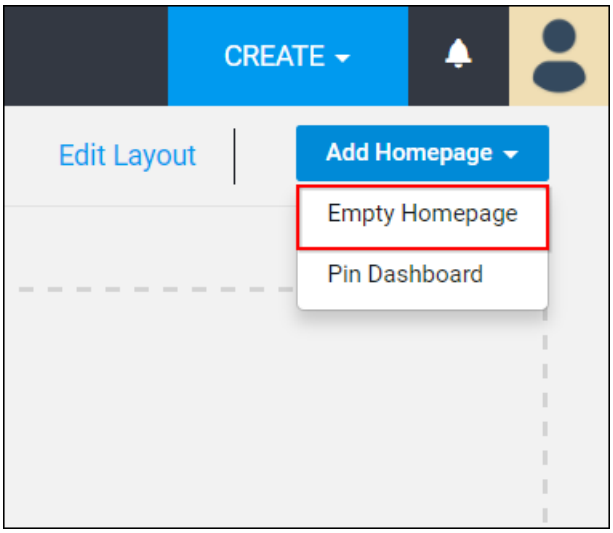
This section explains on how to add Homepage in the Syncfusion Dashboard Server.

Homepage is a collection of either individual widgets from various Dashboards or entire Dashboard pinned to it.

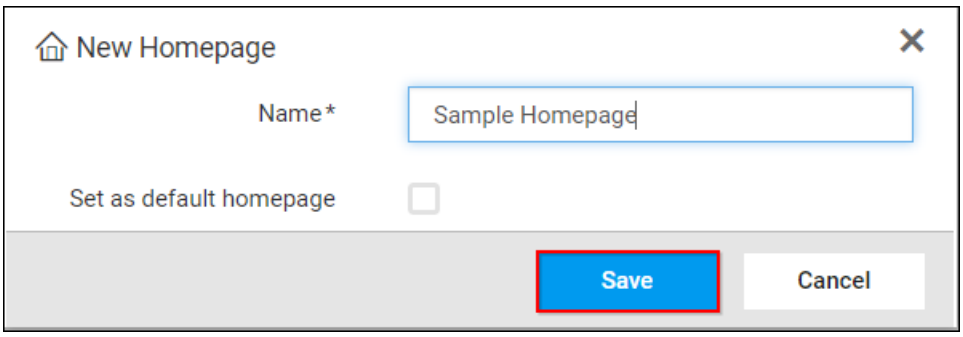
A virtual Homepage named as Homepage1 has been added by default for all users to illustrate how a Homepage looks like. It will be saved if the layout is changed or a widget is added to it.

Add empty Homepage

1. Click on Add Homepage button in the menu and select Empty Homepage to add a new empty Homepage.



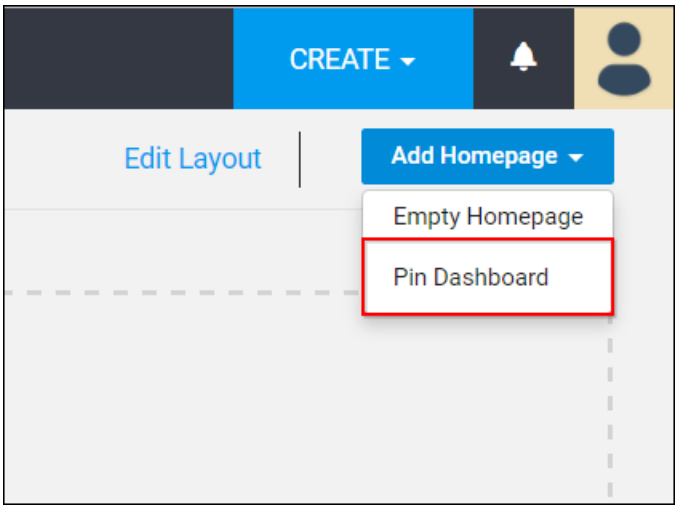
2. Enter the name of the Homepage and save it.



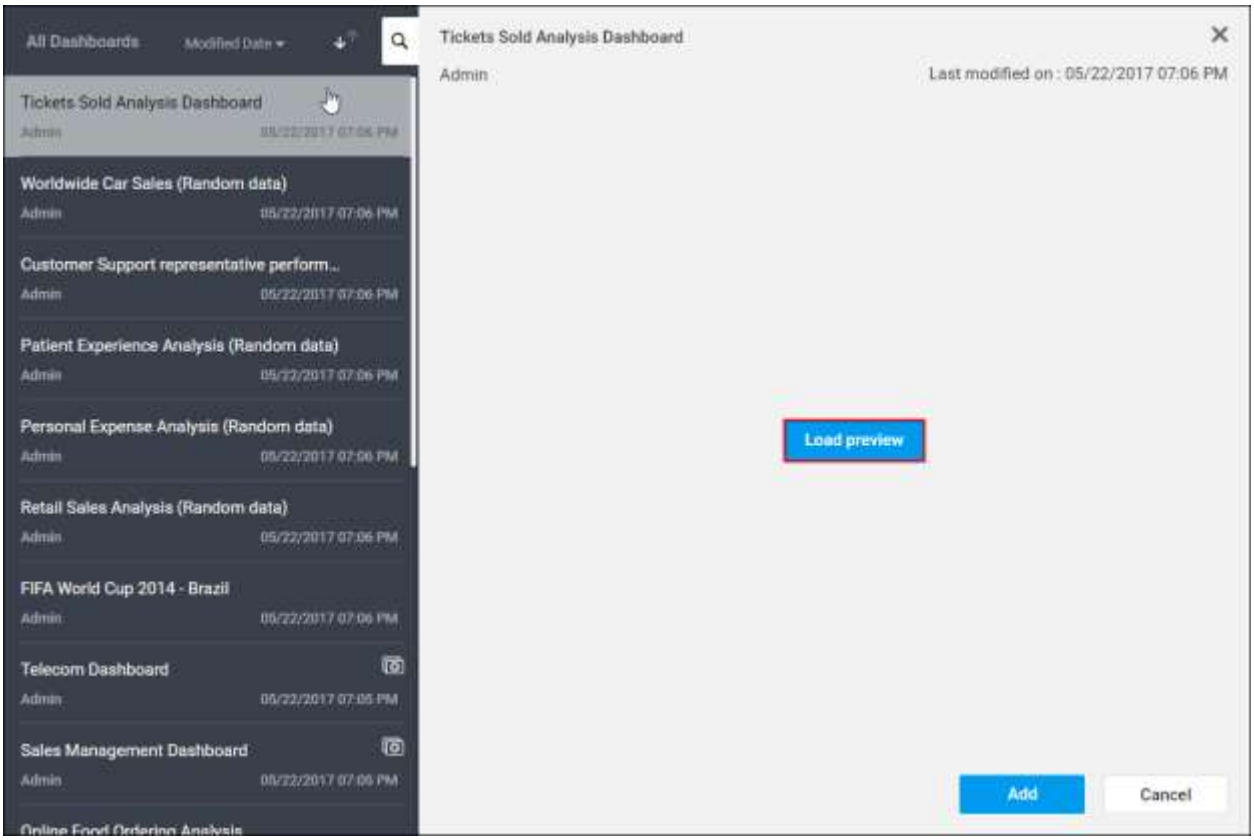
3. After saving, the Homepage will be added into Dashboard Server and the newly added Homepage will be loaded.

Add Dashboard Homepage

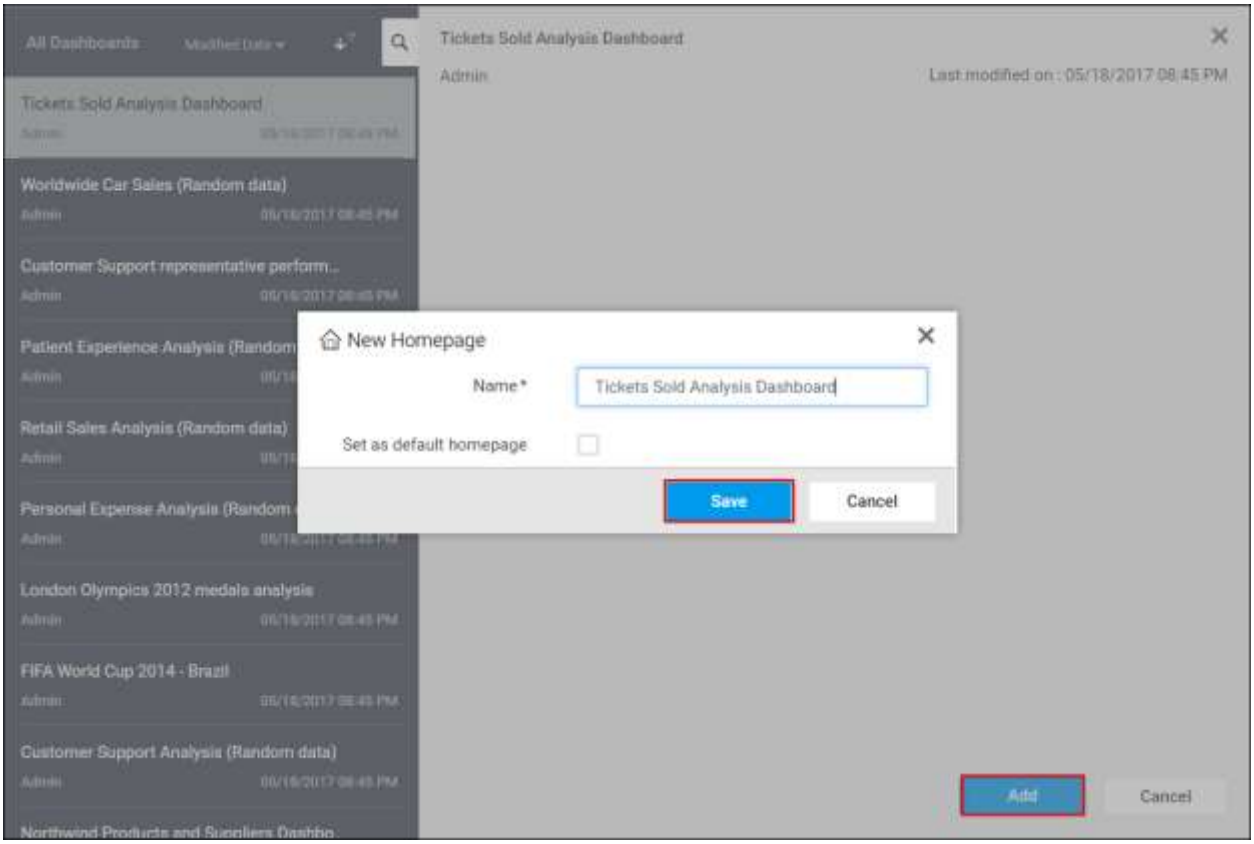
1. The Dashboard can be added into the Homepage by clicking Pin Dashboard drop down option from Add Homepage button.



2. Select any Dashboard from the list of available Dashboards and click Load Preview to preview the selected Dashboard.



3. While adding, the name of the Dashboard will be replaced as the Homepage name in a popup window, and it can be editable.



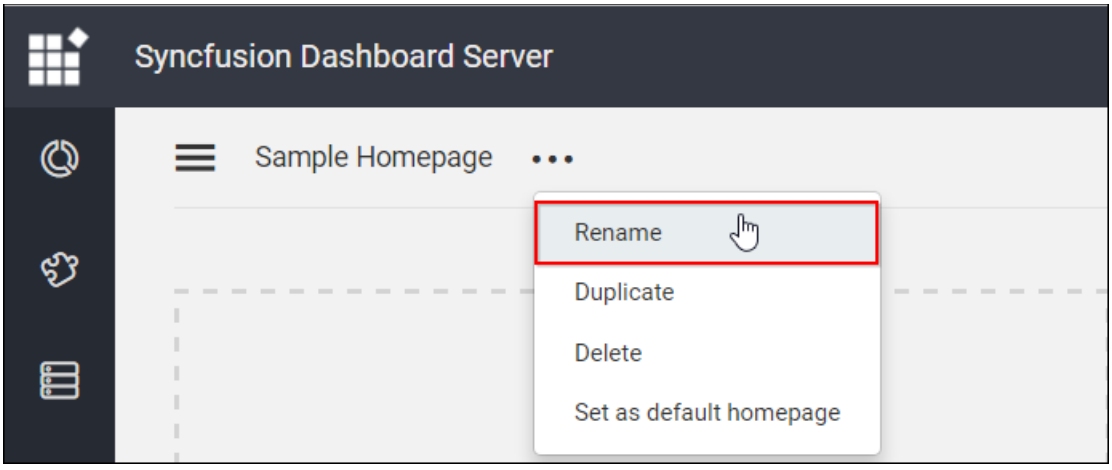
4. After saving, the Homepage will be added into Dashboard Server and the newly added Homepage will be loaded.

Rename Homepage

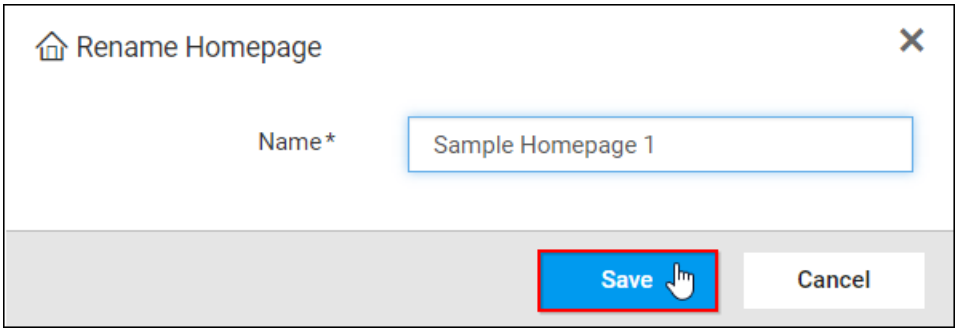
This section explains on how to rename Homepage in the Syncfusion Dashboard Server.

Steps to rename Homepage

1. Click on the Rename option from the context menu.



2. Enter the new name for the Homepage and save it.

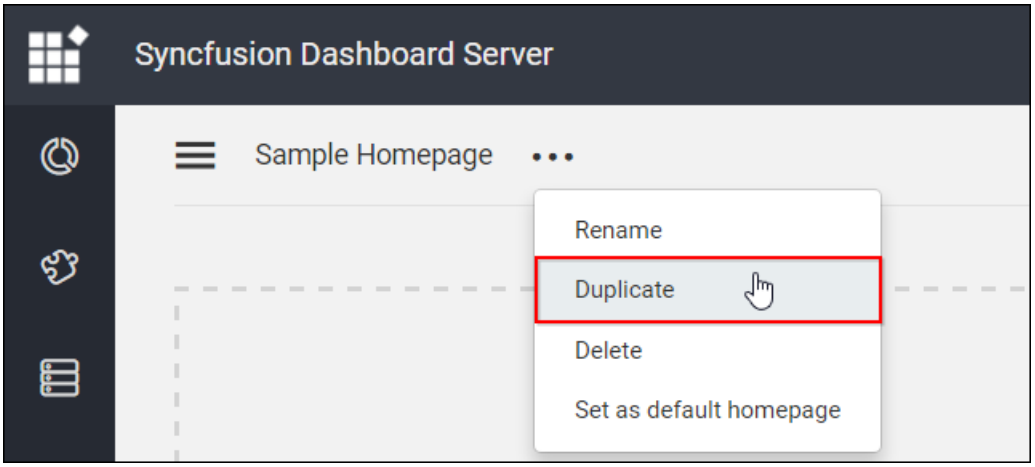


Duplicate Homepage

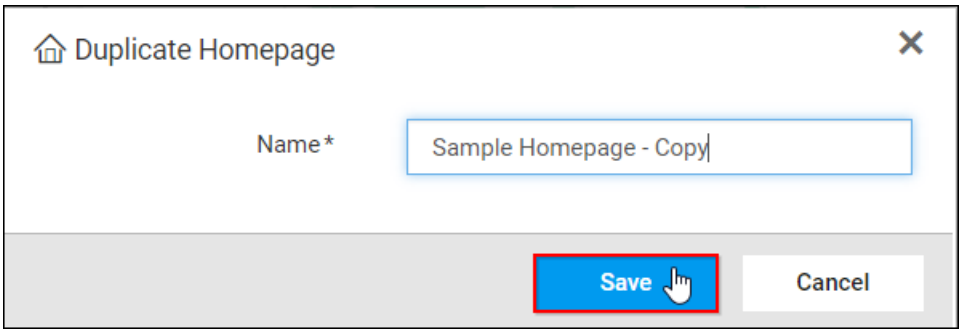
This section explains on how to duplicate Homepage in the Syncfusion Dashboard Server.

Steps to duplicate a Homepage

1. Click on the Duplicate option from the context menu.



2. Enter the new name and save it.



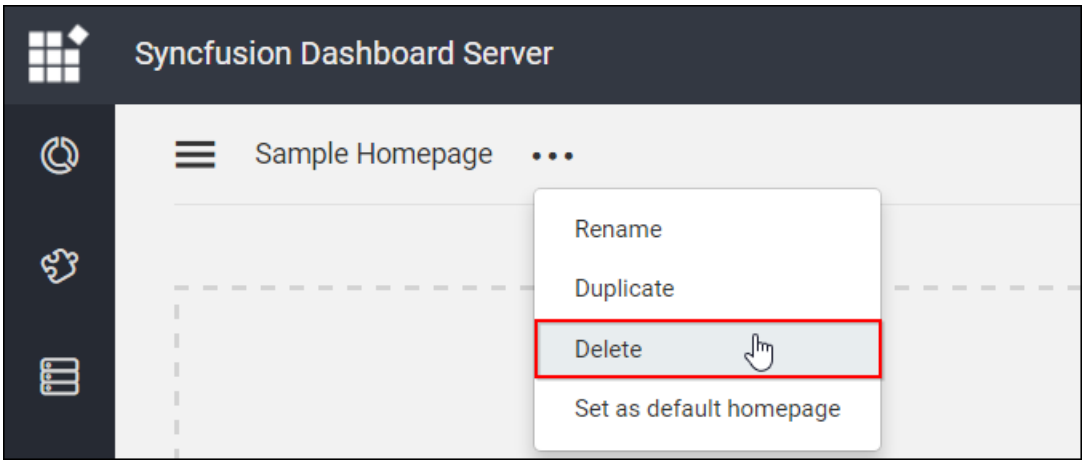
3. A duplicated Homepage have been saved and it will be loaded automatically.

Delete Homepage

This section explains on how to delete Homepage from Syncfusion Dashboard Server.

Steps to delete a Homepage

1. Click on the Delete option from the context menu as shown below,



2. After deleting, default Homepage will be loaded.

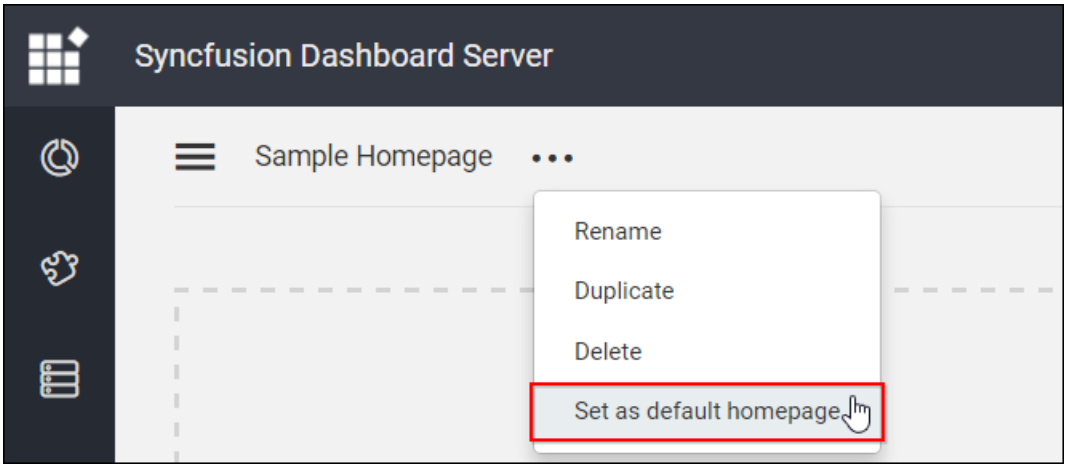
Set default Homepage

This section explains on how to set default Homepage in the Syncfusion Dashboard Server.

Steps to set default Homepage

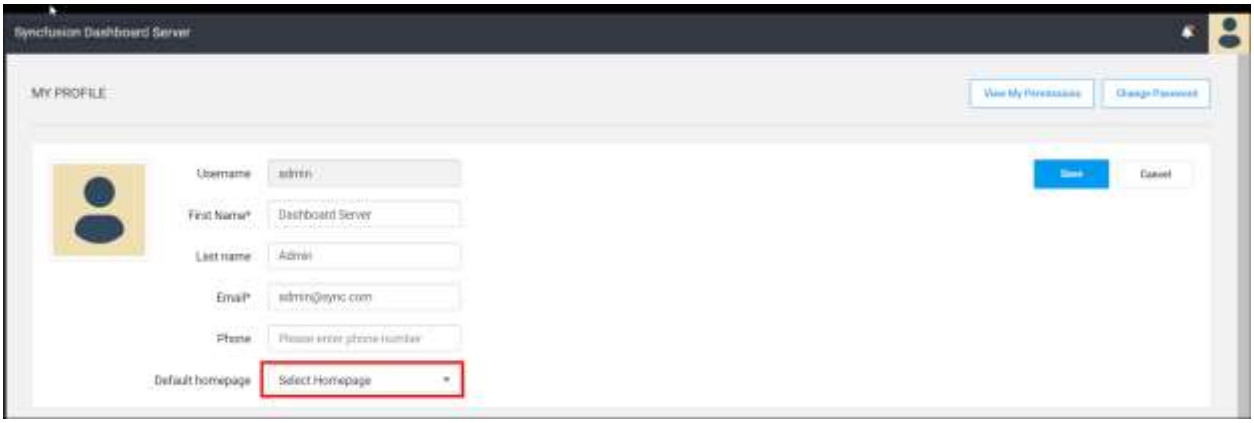
Default Homepage will be loaded by default when the user navigates to the Homepage.

1. Click on the Set as default homepage option from the context menu.

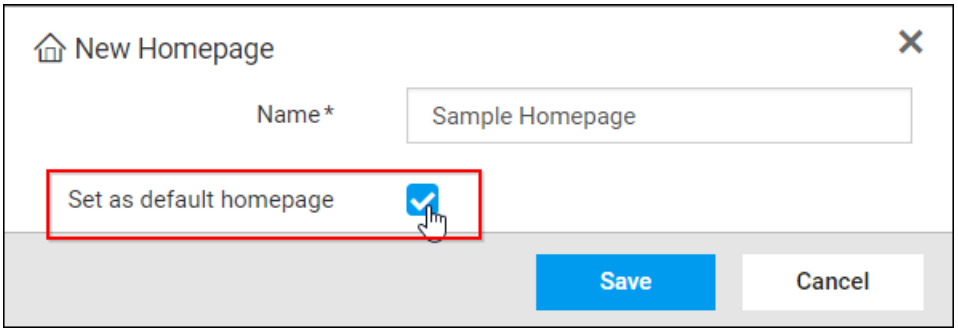


2. Default Homepage can also be selected from the user profile settings.
    - Select Edit Profile from the user profile page.

- Select the default Homepage from the dropdown list and click Save



3. Default Homepage can also be created while creating the new Homepage.
    - Check the Set as default homepage option and the newly created Homepage will be the default Homepage.

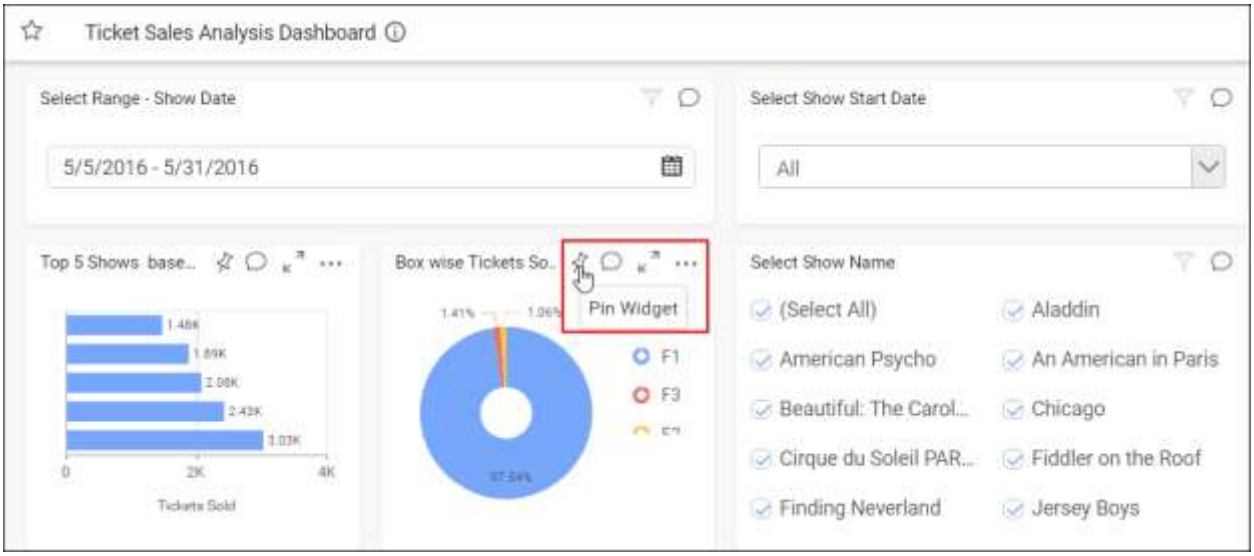


Pin Widgets from Dashboard

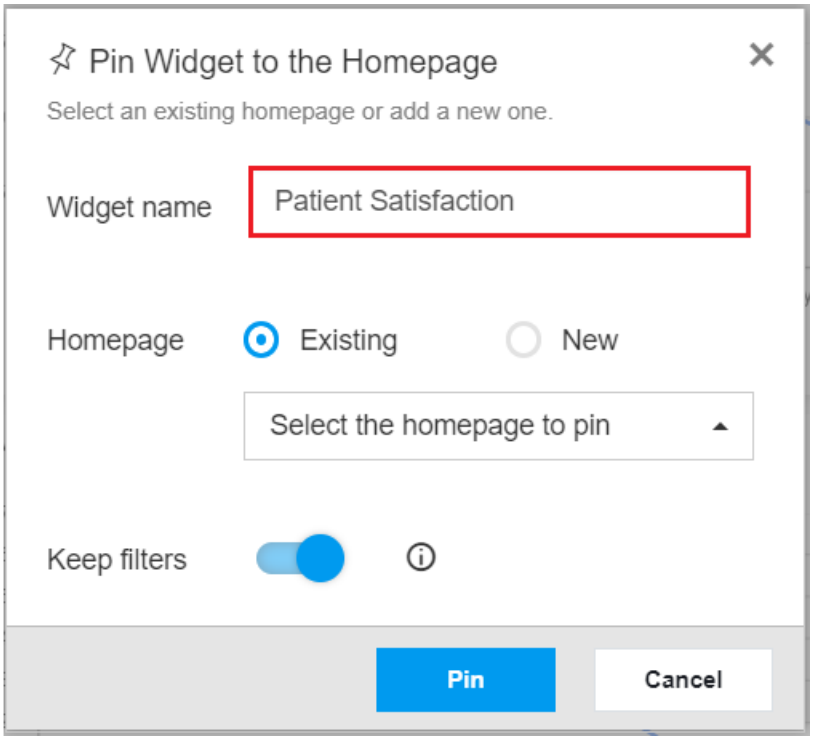
This section explains on how to pin widgets to Homepage in the Syncfusion Dashboard Server.

Steps to pin Widget to Homepage

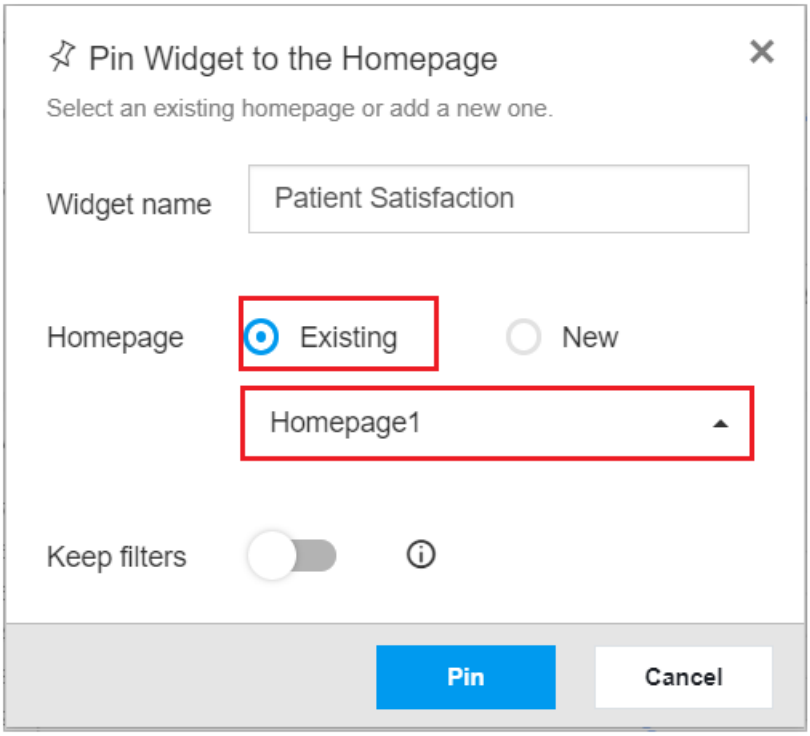
1. Widget can be pinned to the Homepage by clicking pin icon on the specific widget.



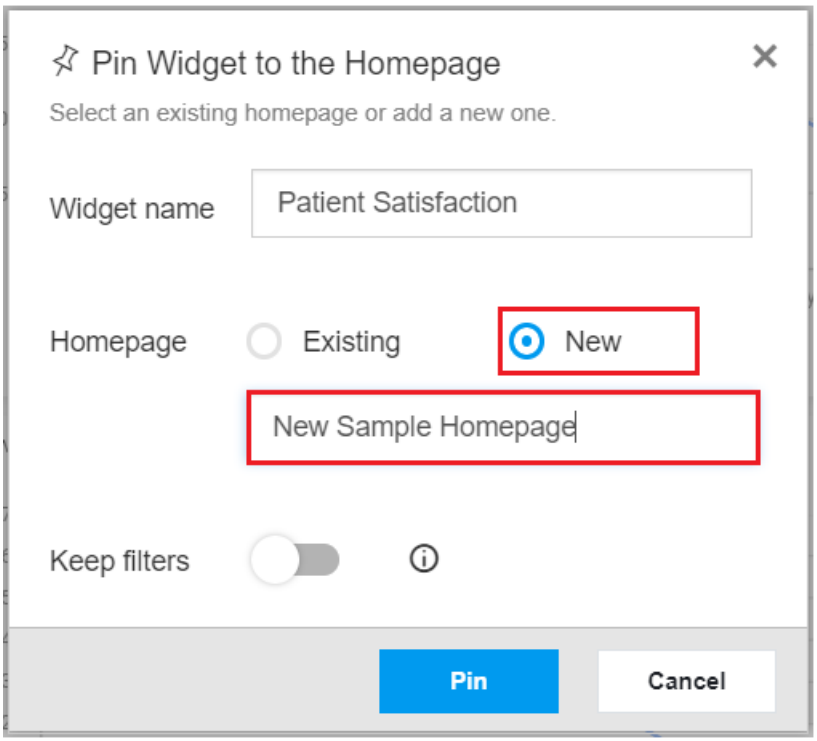
2. Enter the name of the widget.



3. Widget can be pinned either to the Existing or New Homepage.
   - Clicking Existing Homepage will list the available Homepages which is already saved by the user.

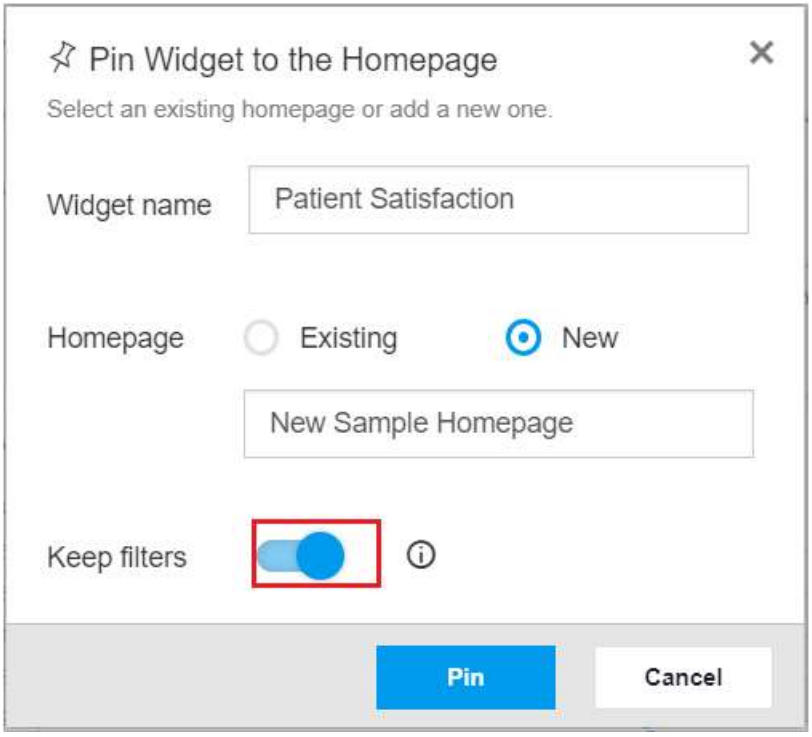


- Enter the new Homepage name for pinning to the New Homepage.

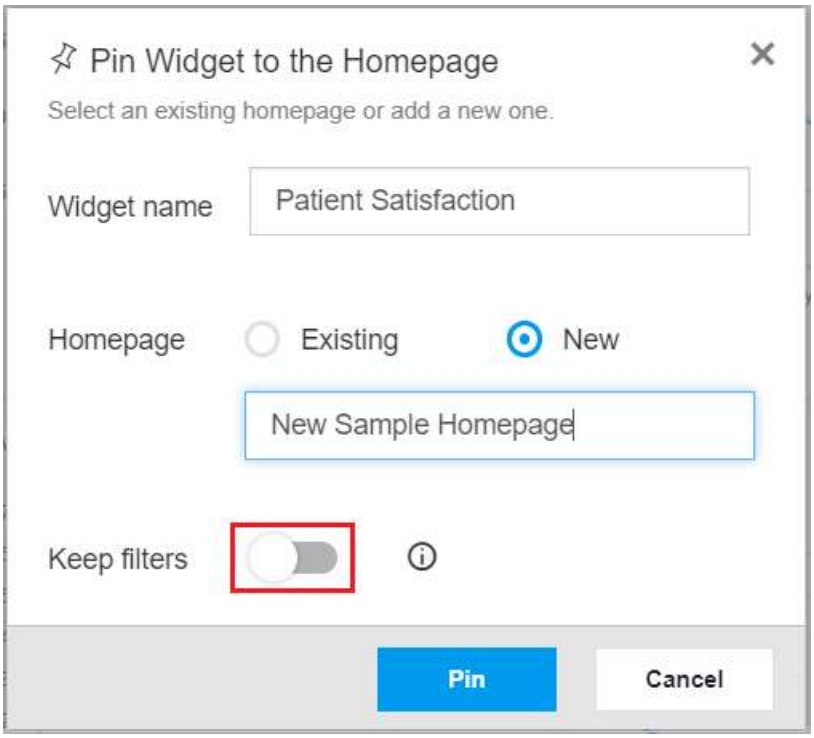


4. Widget can be pinned along with the applied filters.

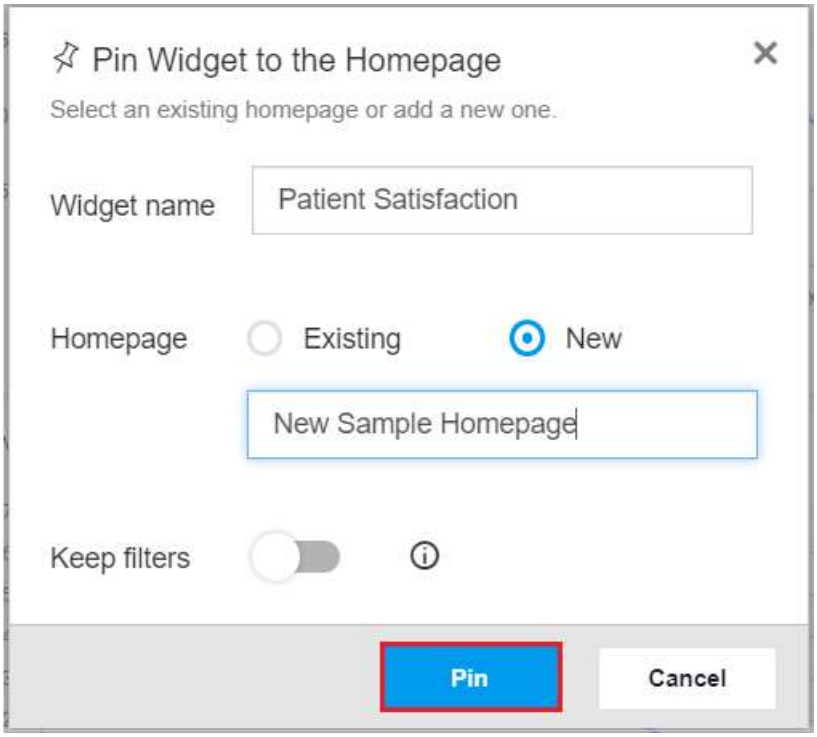
- When we enable the Keep filters switch, Widget will be pinned along with the applied filters.



- When we disable the Keep filters switch, Widget will be pinned without the applied filters.



4. Click Pin and it will pin the specific widget to the Homepage.

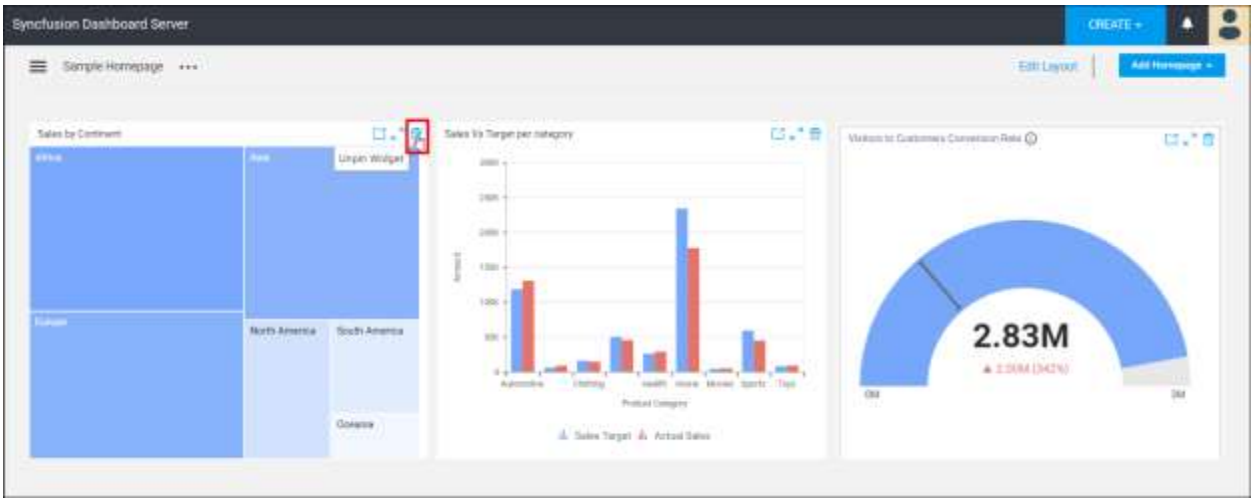


Unpin Widgets

This section explains on how to unpin Widgets from Homepage in the Syncfusion Dashboard Server.

Steps to unpin Widget from the Homepage

Pinned widget can be unpinned from the Homepage by clicking the unpin icon on the specific widget.



Edit Layout

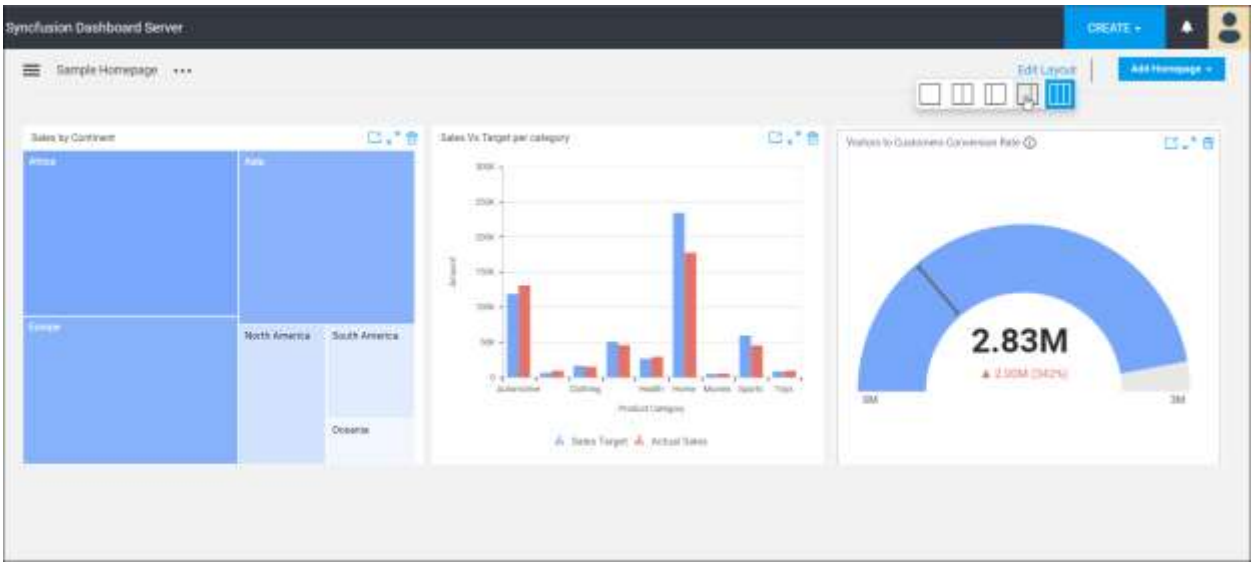
This section explains on how to edit Homepage layout in the Syncfusion Dashboard Server.

Homepage is based on column layout.

Steps to edit layout in Homepage

To change the layout of the Homepage,

1. Click on the Edit Layout option from the Homepage menu and change the layout of the Homepage from the available layout options as below,

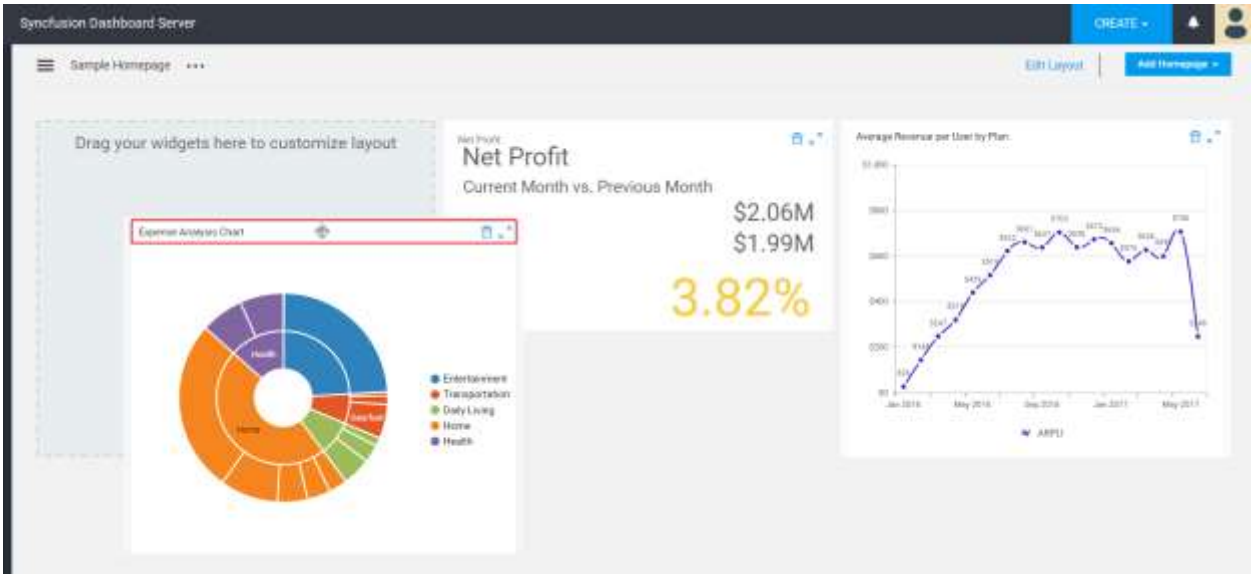


Drag and Drop

This section explains on how to drag and drop widgets between columns inside Homepage in the Syncfusion Dashboard Server.

Steps to drag and drop widgets in Homepage

Widgets inside the Homepage can be dragged and dropped between its columns by using its headers.

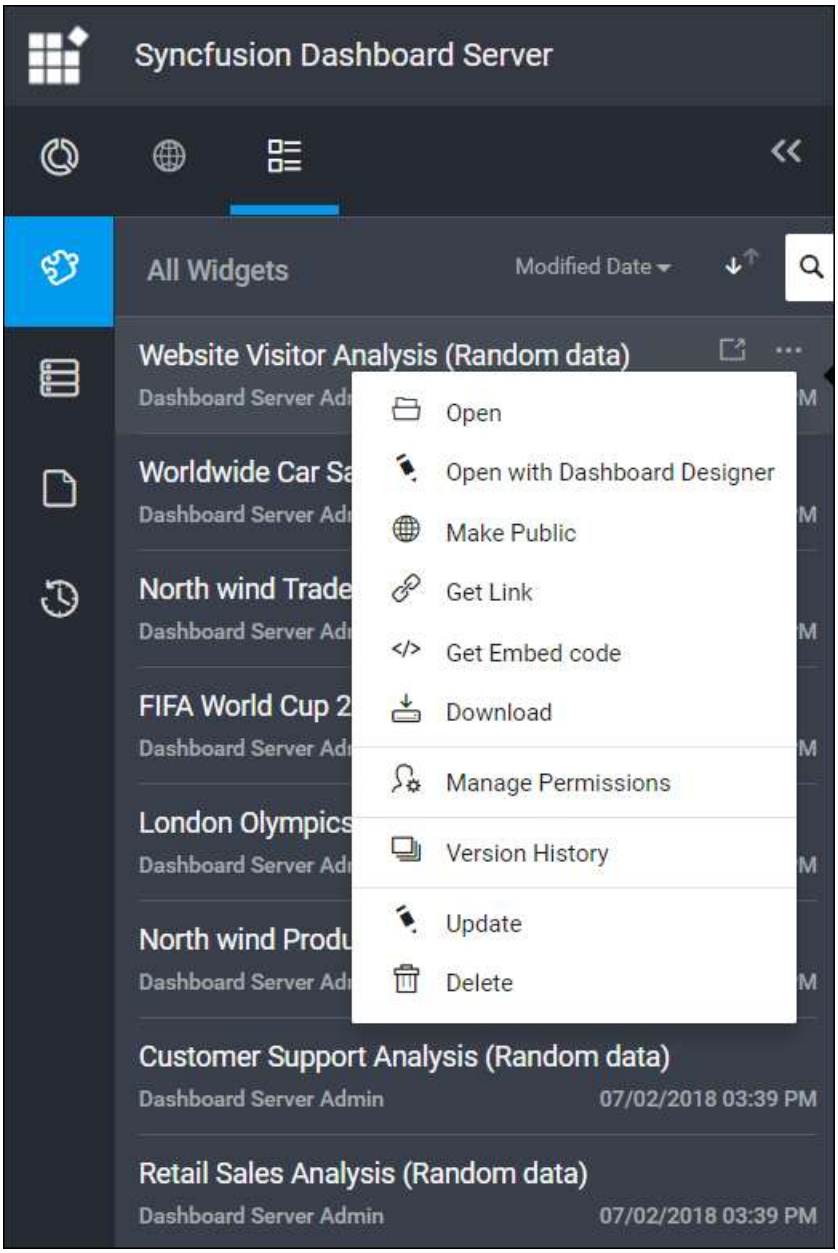


Note: Dashboards cannot be draggable in Homepage.

Manage Widgets

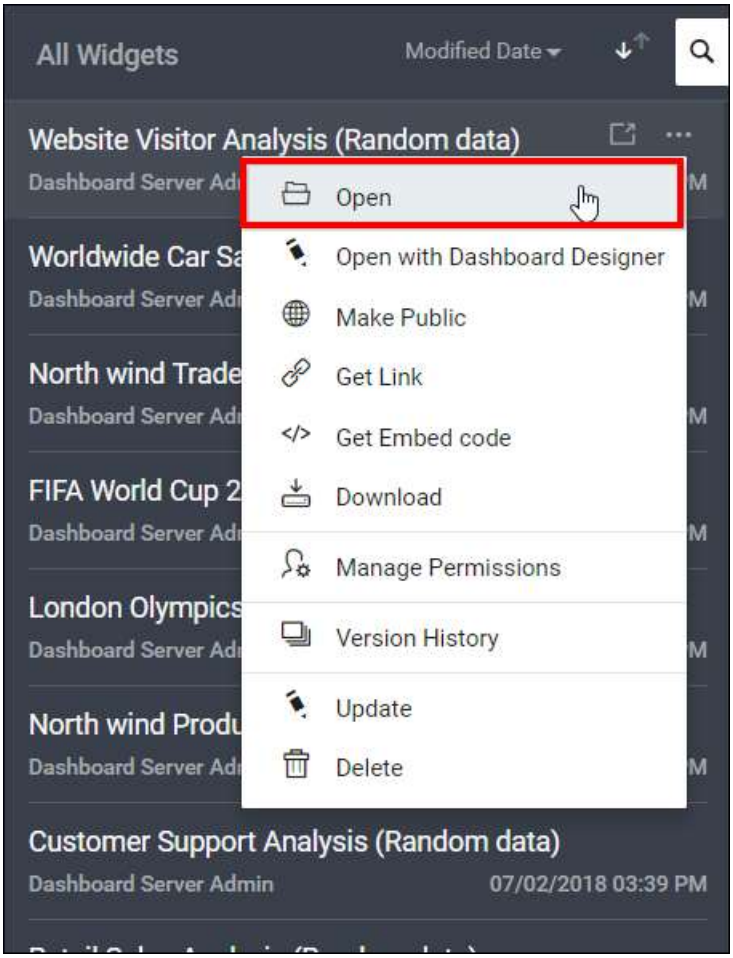
This section explains on how to open, add, update, share, download, delete Widgets and also on how to view version history for Widgets in the Syncfusion Dashboard Server.

Widgets that are accessible by the user depending on the user's permission is displayed in the Widgets page.



Open Widget

Widgets are opened in our embedded Dashboard Viewer.

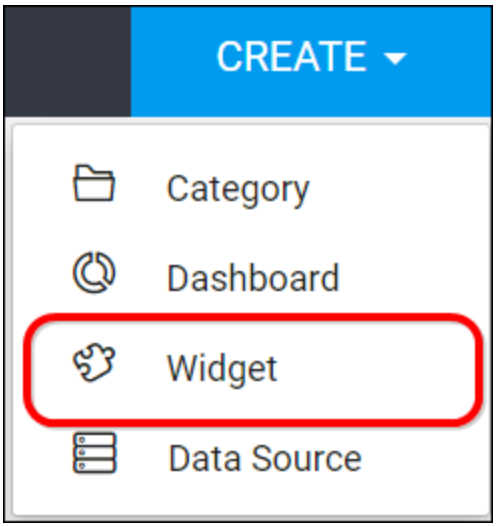


Add Widgets

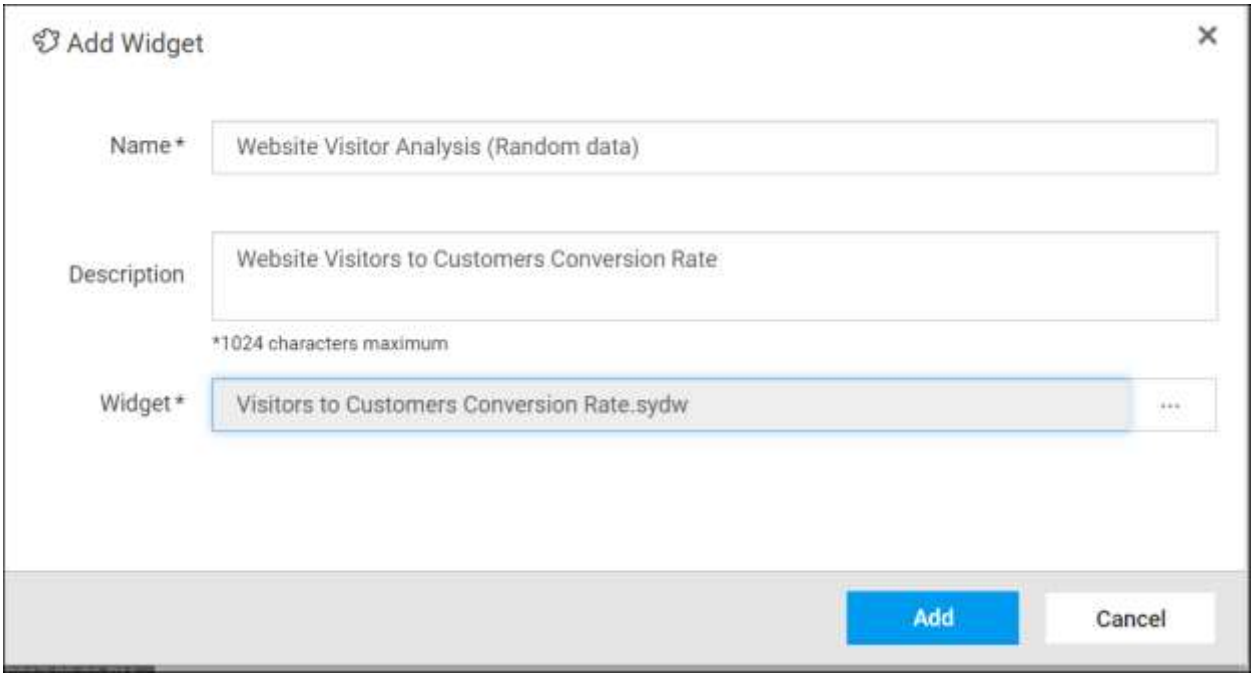
- Widgets can be created only if the user has Create All Widgets permission.
- Widgets can be designed in the Dashboard Designer and then added in the Dashboard Server. For client users, Dashboard Designer can be downloaded from the Dashboard Server.

Steps to add a Widget

1. Click on the Create button in the menu and select Widgets to add a Widget.



2. Fill in the name and description of the Widget and upload the Widget file(.sydw) in the Add Widget dialog box.

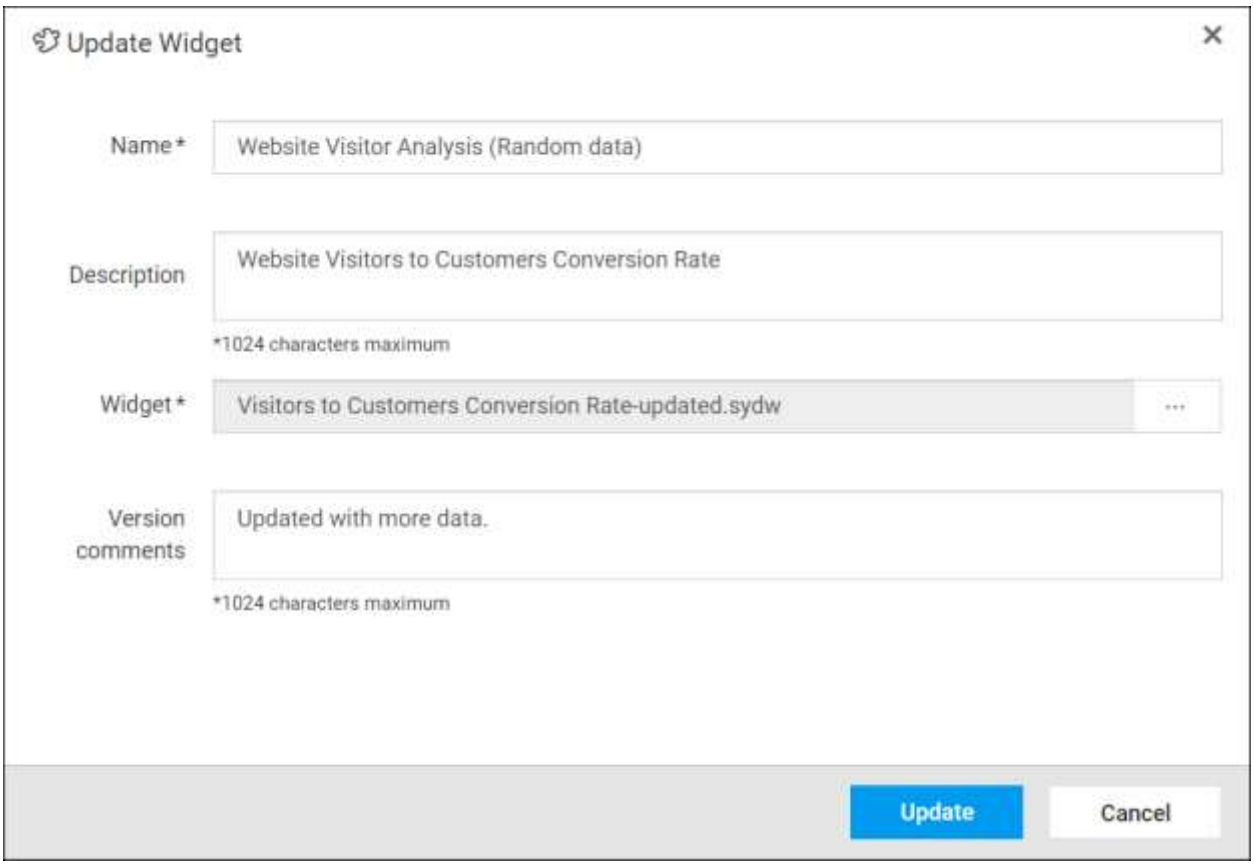


3. After filling the form, the Widget can be saved to be added in the Dashboard server.

Note: Read Write Delete Download permission for that Specific Widget will be added for the user who created the Widget.

Update Widgets

Name, description and the Widget file(.sydw) can be changed for the Widget in the update Widget dialog box.

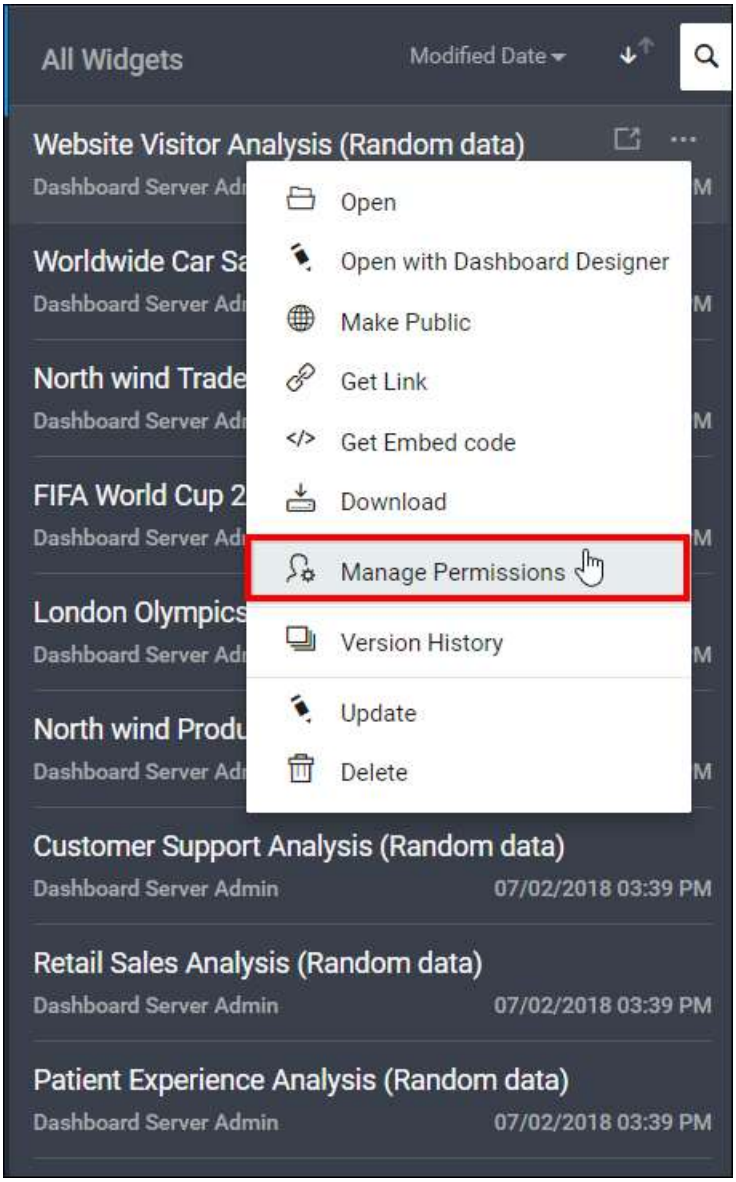


Share Widgets

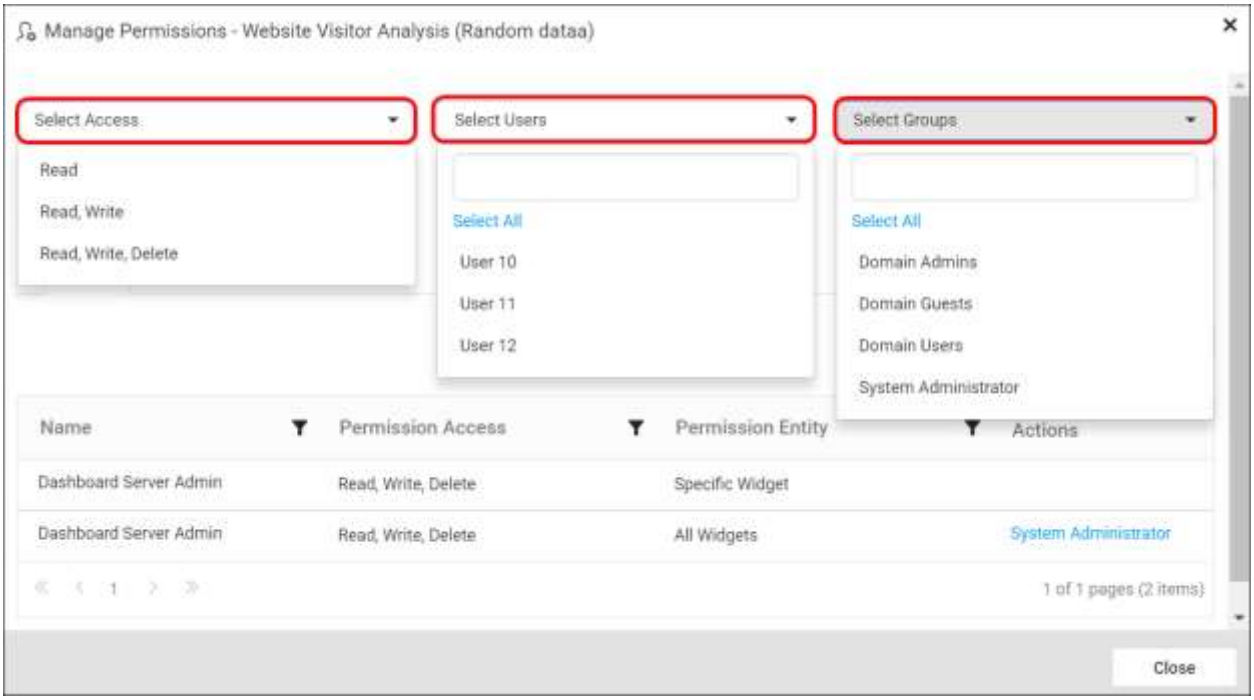
This section explains on how to share widgets with the other users in the Dashboard Server.

Steps to share a Widget

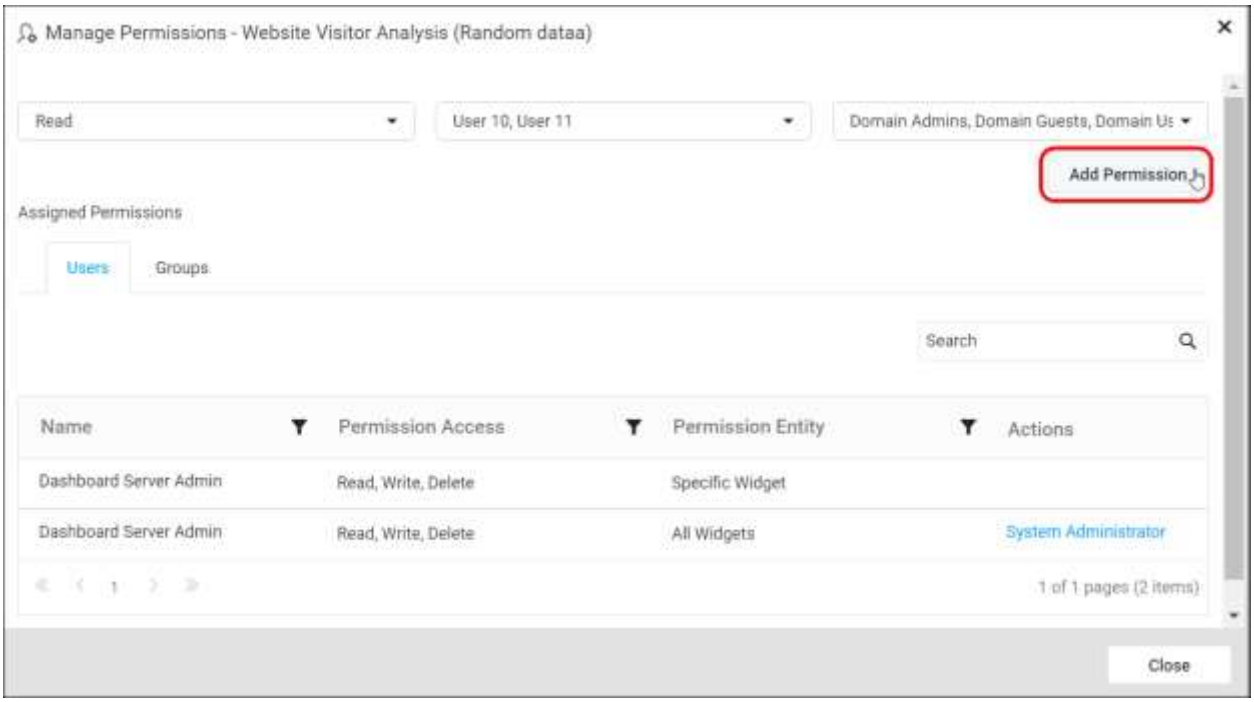
1. Click the Actions button in the Widgets grid context menu and select Manage Permissions option.



2. Select the permission access from the Select Access dropdown and select the users or groups to share the widget.



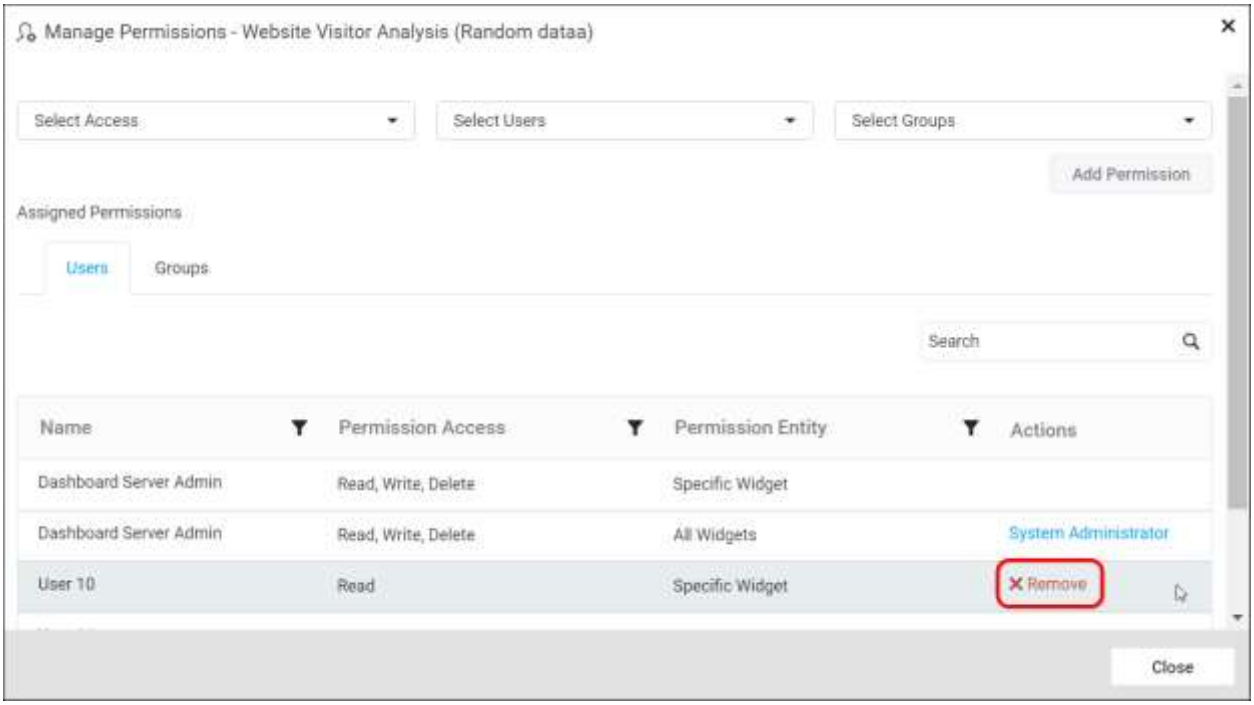
3. After selecting the access and users or groups, click on the Add Permission button.



Note: Only the user who created the widget and the Administrator can share the widget with other Dashboard Server users.

Remove Permission

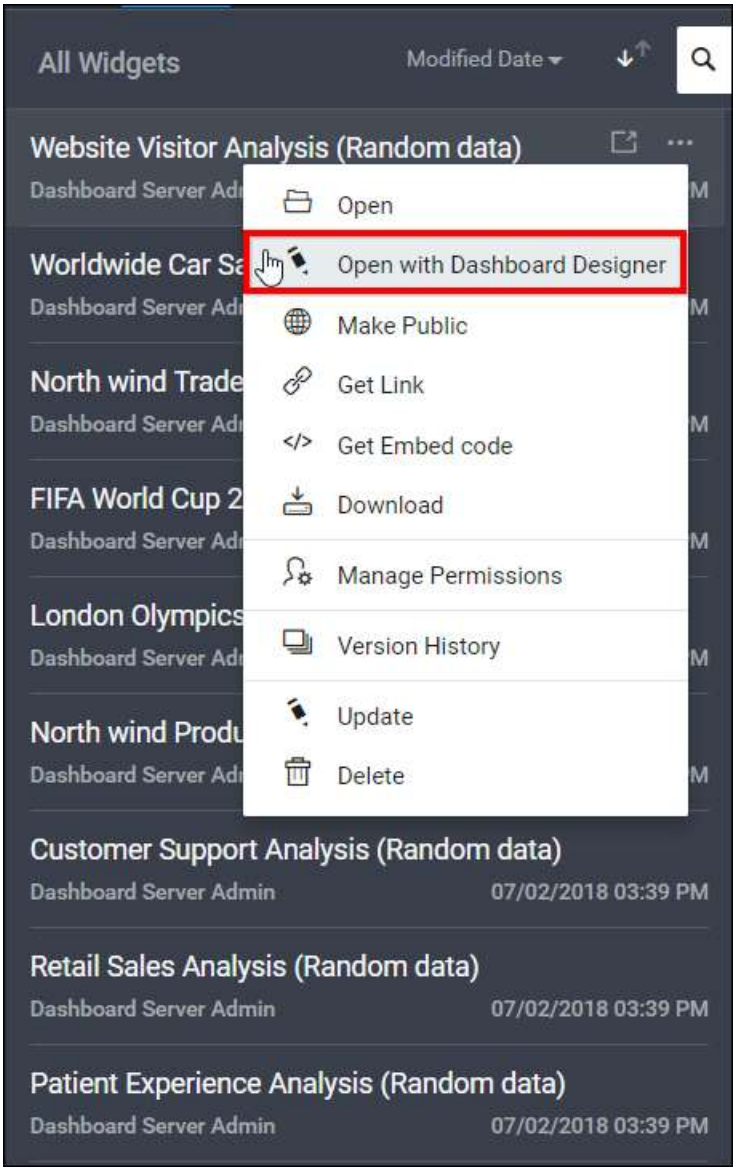
The user who created the widget and the Administrator can remove the shared widget permissions using the Remove option in the Actions column of the each permissions.



Open with Dashboard Designer

Widgets can be launched directly in the Dashboard Designer from the Dashboard Server.

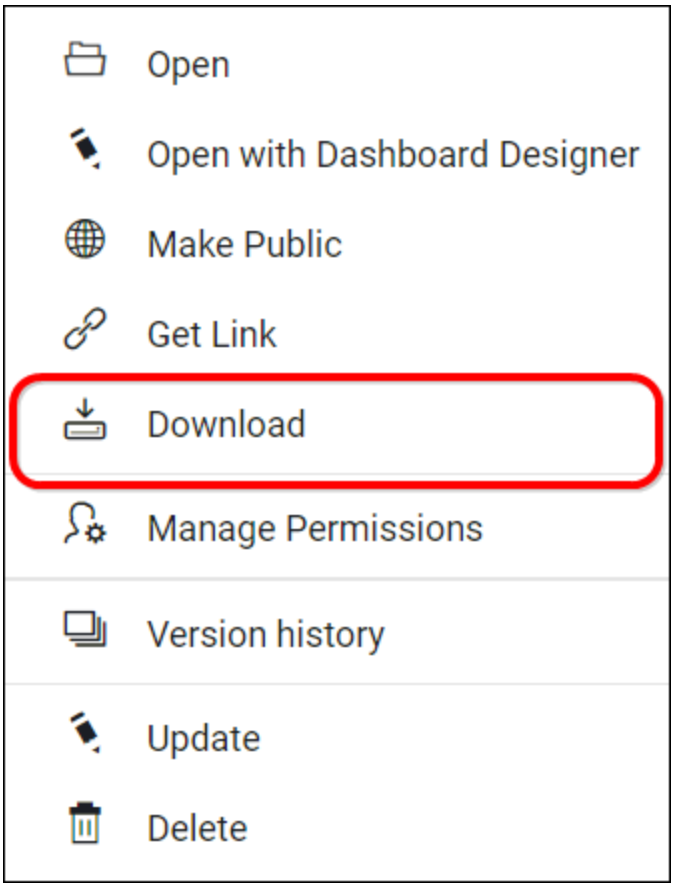
Click the Actions button in the Widgets grid context menu and select Open with Dashboard Designer to open the Widget in the Dashboard Designer if it is already installed in the client machine.



If Dashboard Designer is not already installed in the client machine, then Dashboard Designer will be downloaded in the client machine for the user to install.

Download Widgets

Click the Actions button in the Widgets grid context menu and select Download to download the Widget in .sydw format.



Downloaded Widget can be loaded in our Dashboard Designer.

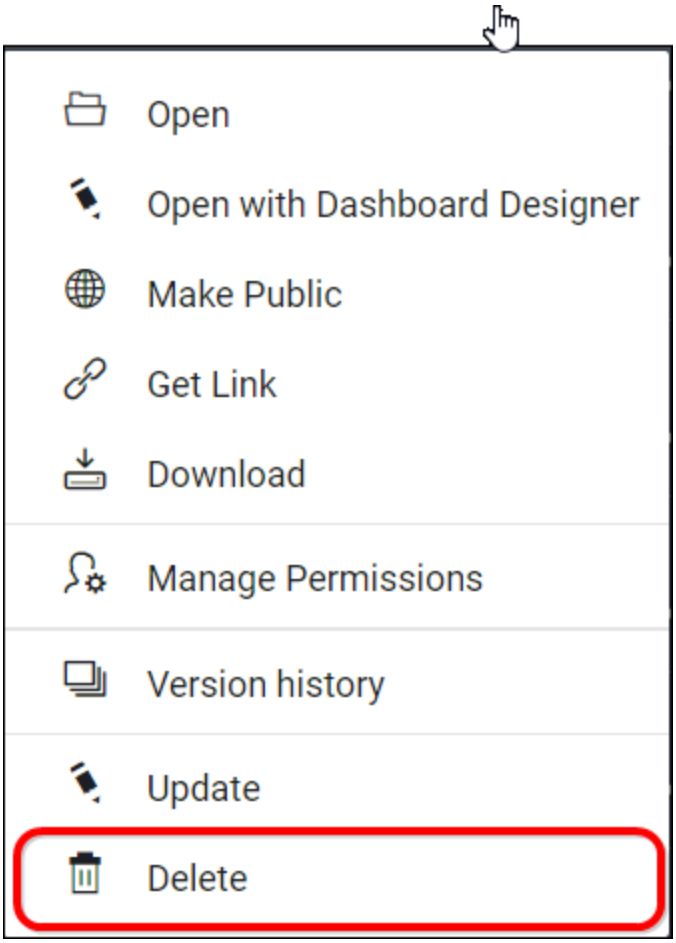
Version History

Versions and file logs for each Widget are maintained in the dashboard server for every changes in the Widget. Check Version History section under Manage Dashboards for more details.

Delete Widgets

Widgets can also be deleted from the Widget server when they are no longer required.

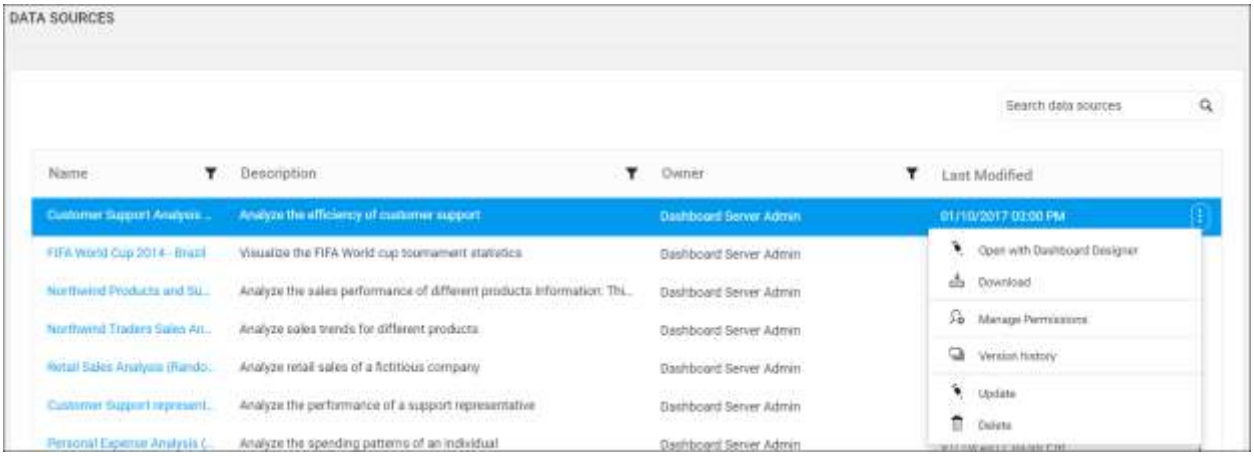
Click the Actions button in the Widgets grid context menu and select Delete to delete the Widget.



Manage Data Sources

This section explains on how to add, update, share, download, delete data sources and also on how to view the version history of data sources in the Syncfusion Dashboard Server.

Data Sources that are accessible by the user depending on the user's permission is displayed in the data sources page.

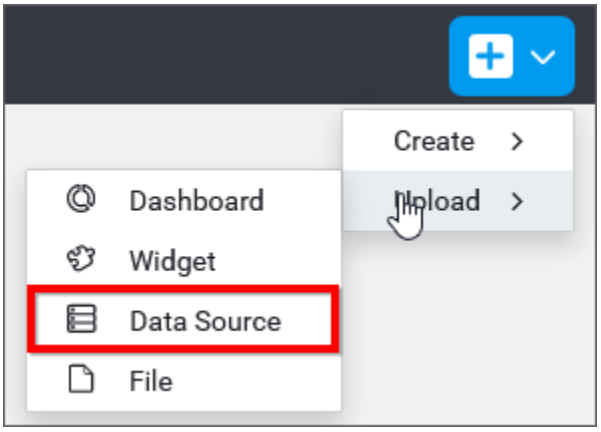


Add Data Sources

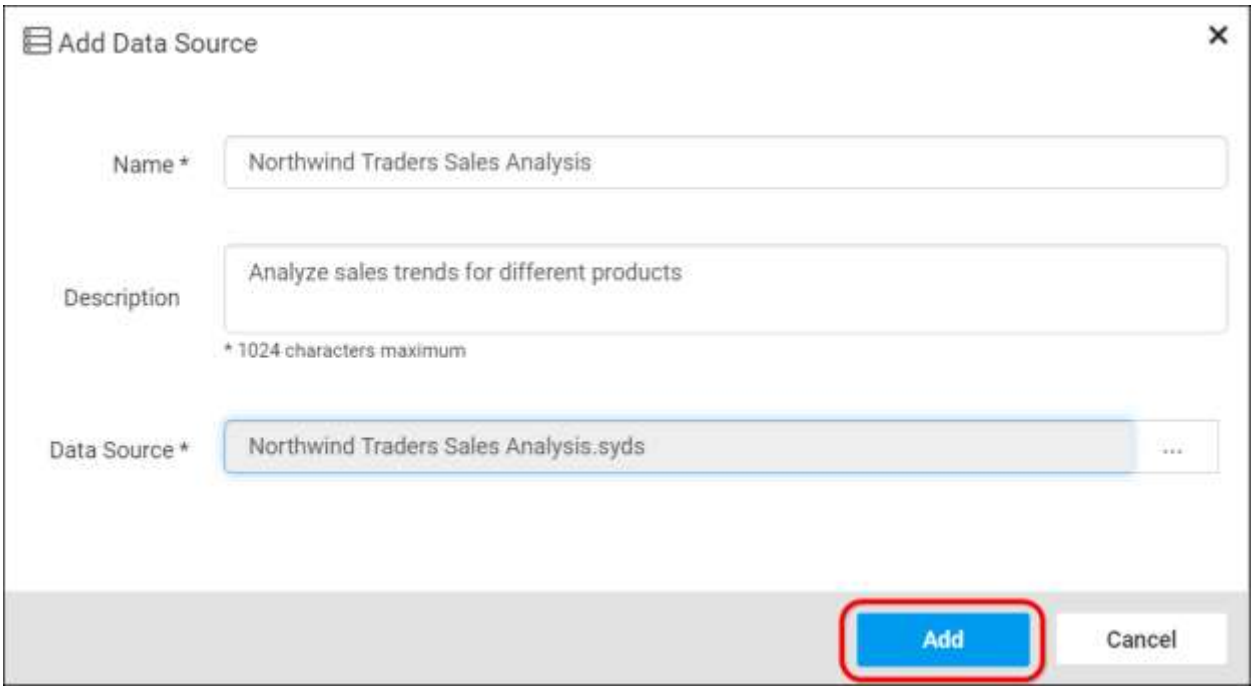
Data source can be created only if the user has Create All Data Sources permission.

Steps to add a data source

1. Click on the Upload button in the menu and select Data Source to add a data source.



2. Fill in the form with name and description of the data source and upload the data source file(.syds).

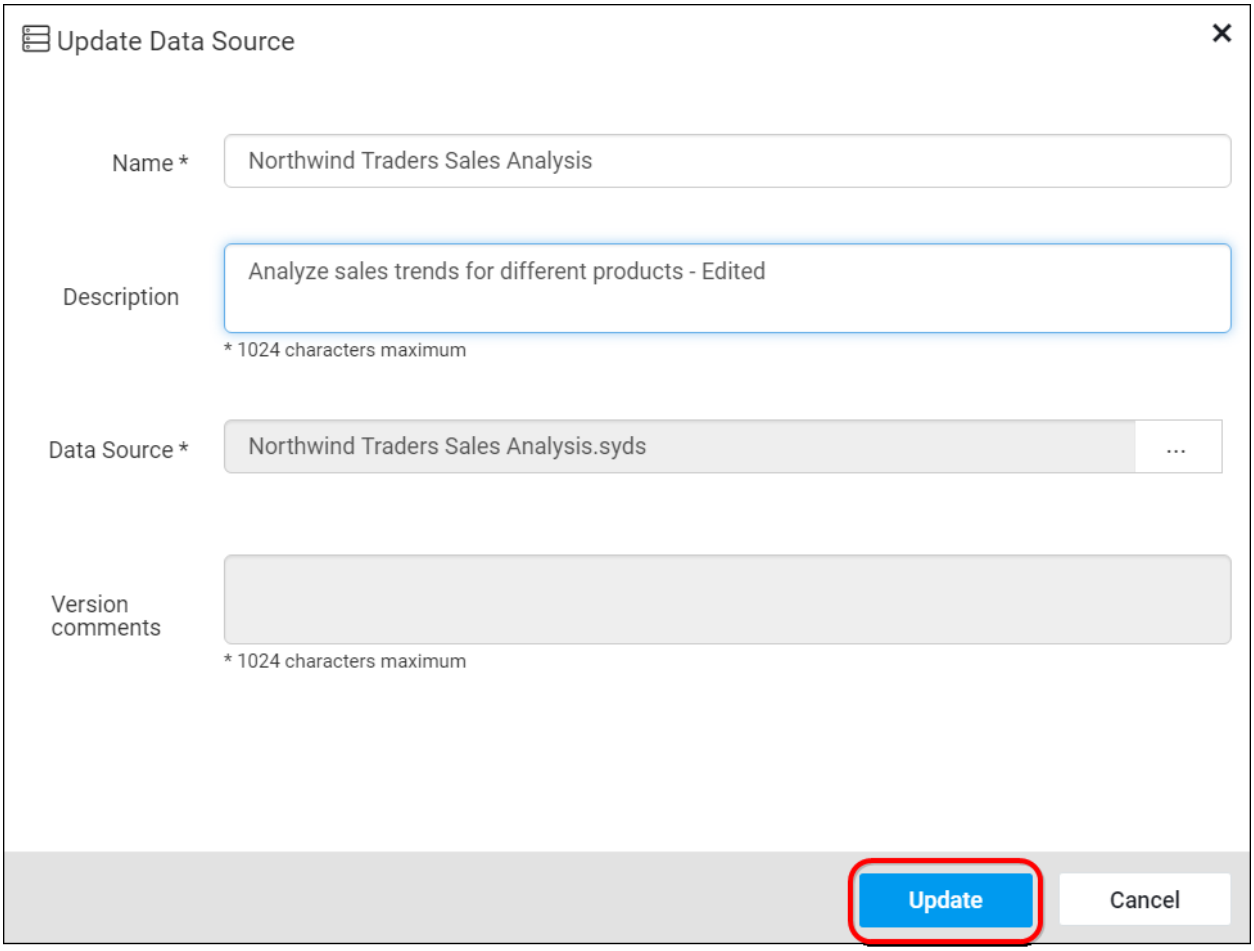


3. When clicking on Add, the data source will be added to the dashboard server.

Note: Read Write Delete Download permission for that Specific Data Source will be added for the user who created the data source.

Update Data Sources

Name, description and the data source file(.syds) can be changed in the update data source dialog box.

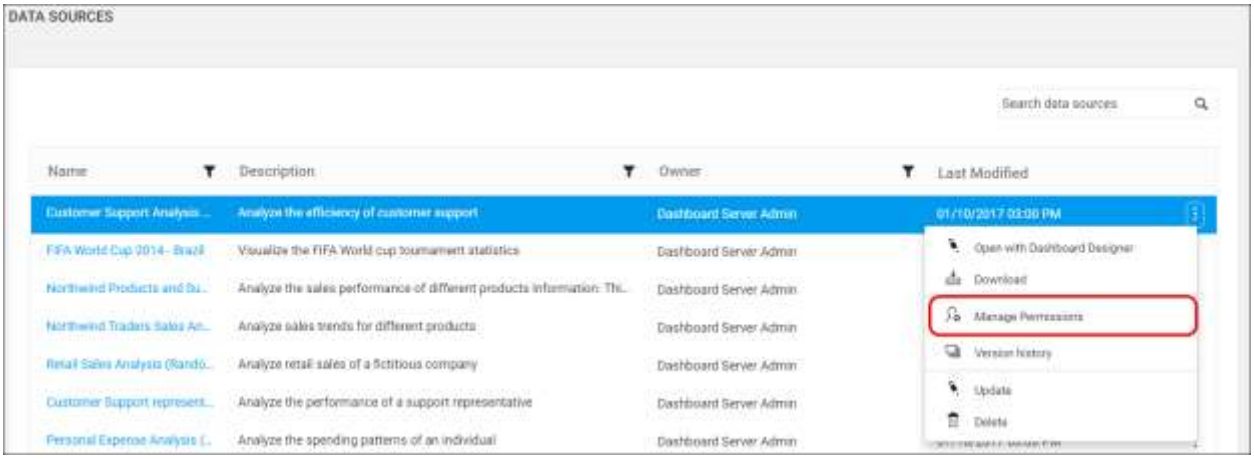


Share Data sources

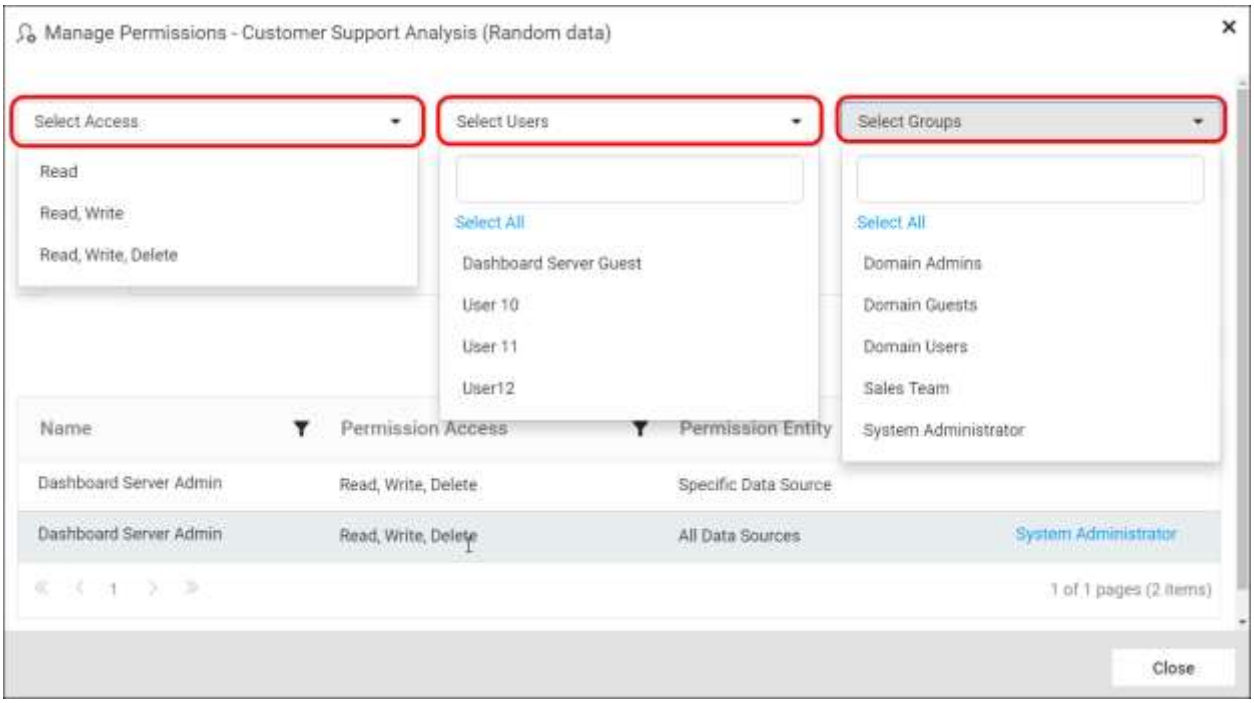
This section explains on how to share data sources with the other users in the Dashboard Server.

Steps to share a data source

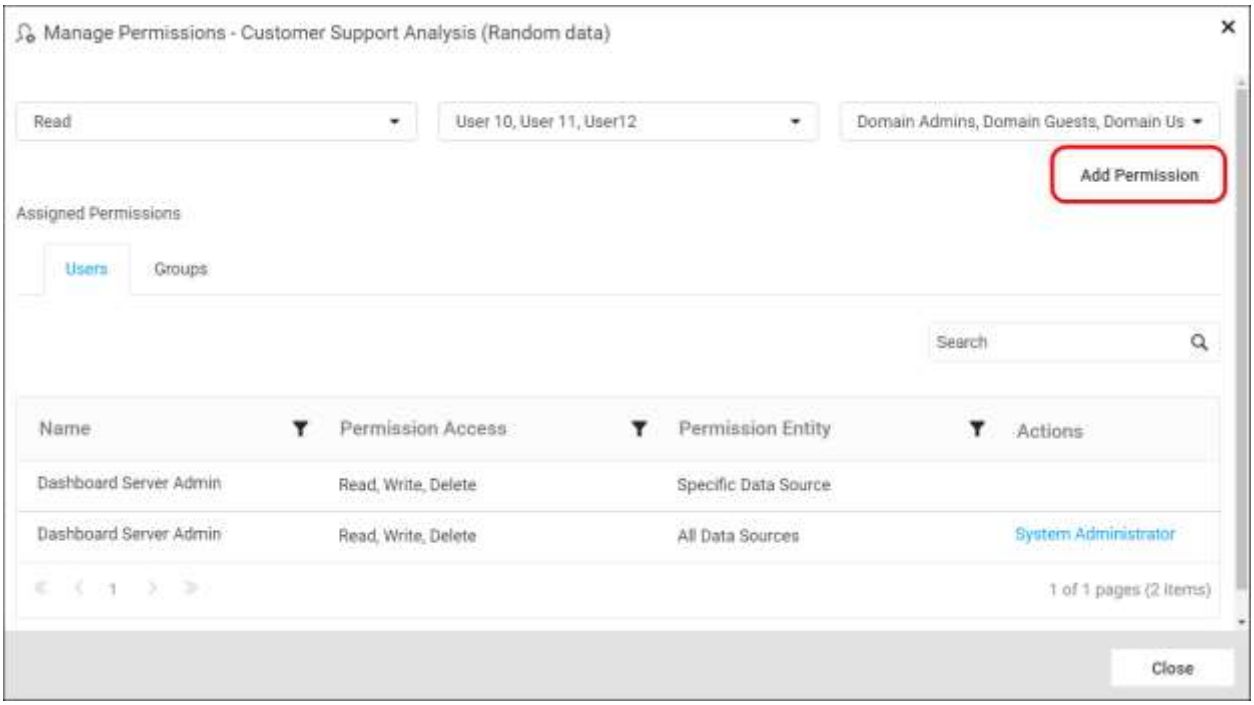
1. Click the Actions button in the Data sources grid context menu and select Manage Permissions option.



2. Select the permission access from the Select Access dropdown and select the users or groups to share the data source.



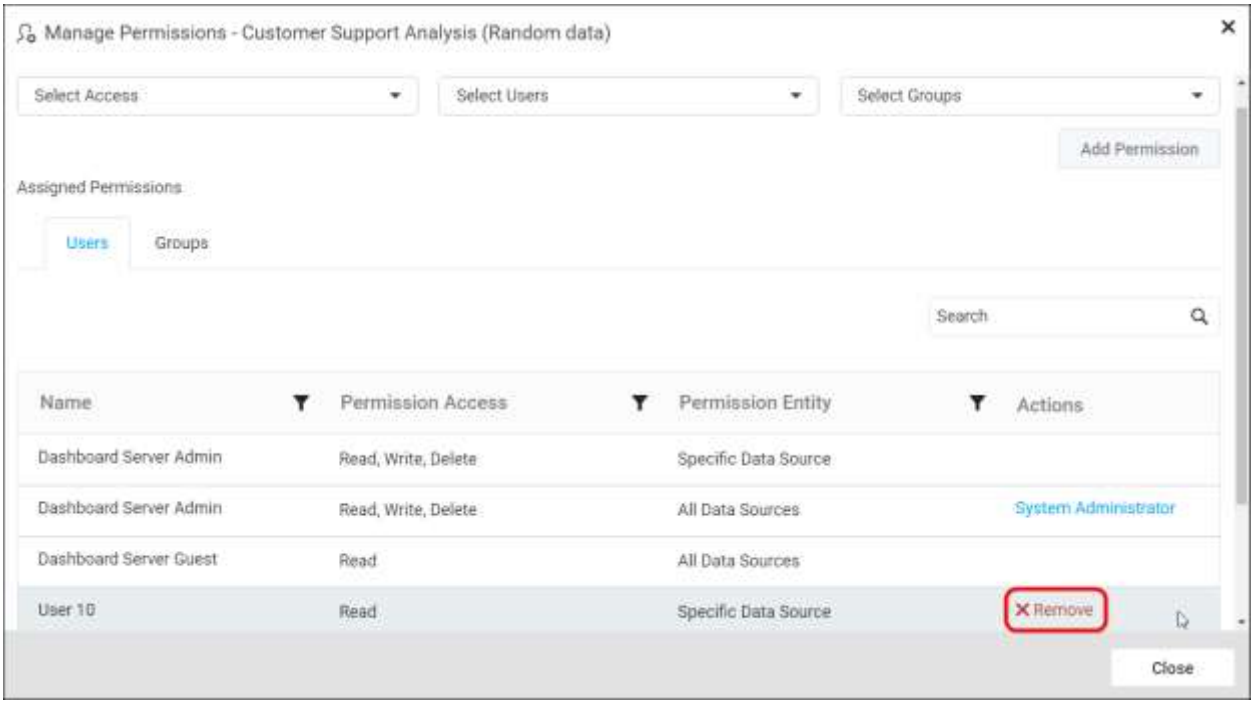
3. After selecting the access and users or groups, click on the Add Permission button.



Note: Only the user who created the data source and the Administrator can share the data source with other Dashboard Server users.

Remove Permission

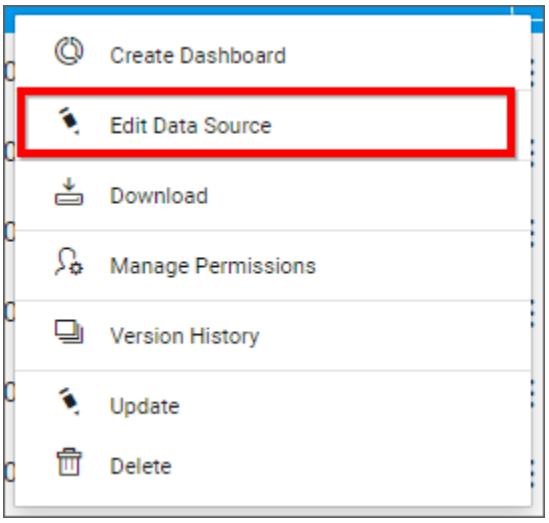
The user who created the data source and the Administrator can remove the shared data source permissions using the Remove option in the Actions column of the each permissions.



Edit Data sources

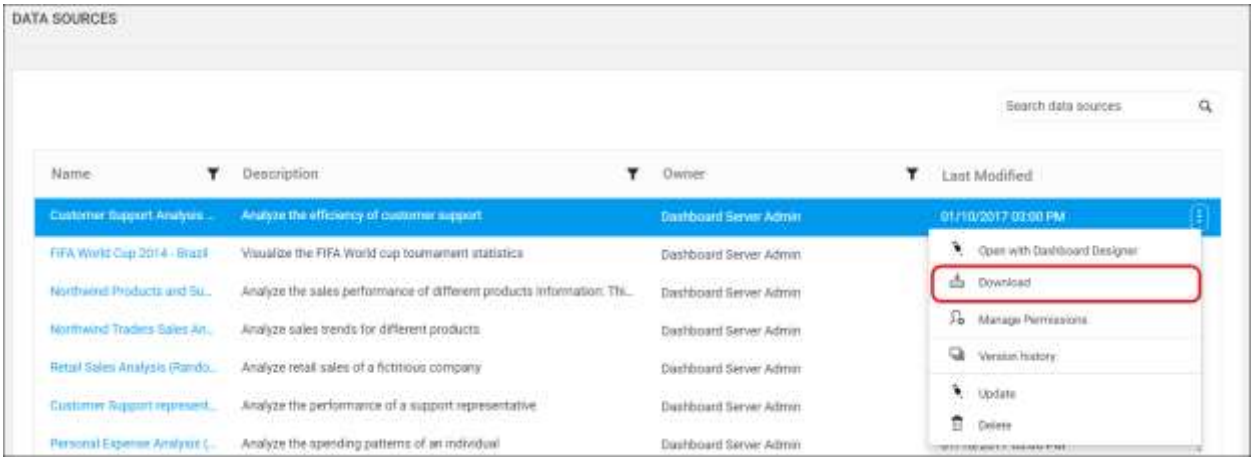
Data Sources can be modified from the Dashboard Server, if the user has Write permission for the particular data source.

Click the Actions button in the Data Sources grid context menu and select Edit Data source to modify the Data Source.



Download Data Sources

Click the Actions button in the data sources grid context menu and select Download to download the data source in .syds format.



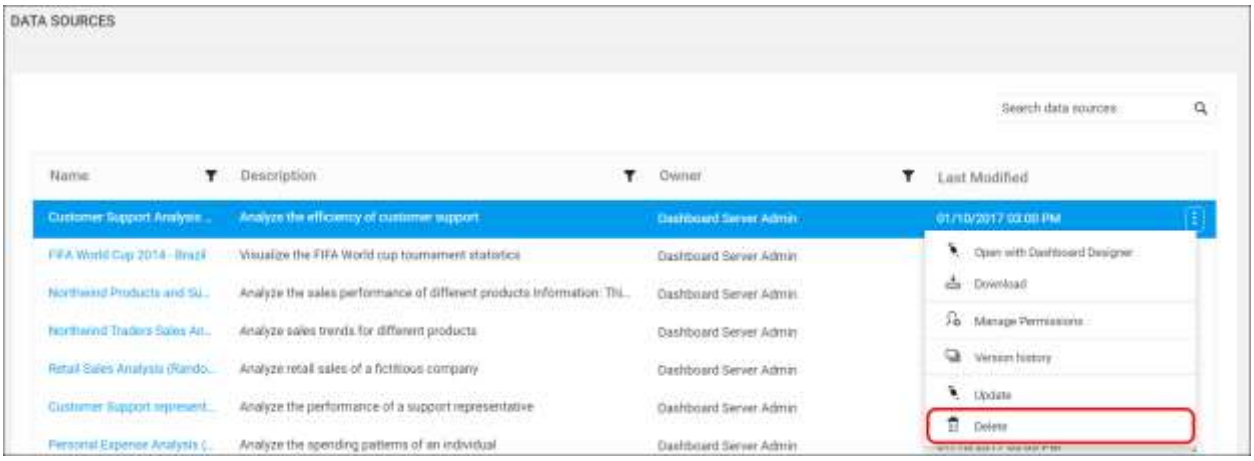
Version History

Versions and file logs for each data source are maintained in the dashboard server for every changes in the data source. Check Version History section under Manage Dashboards for more details.

Delete Data Sources

Data Sources can also be deleted from the dashboard server when they are no longer required.

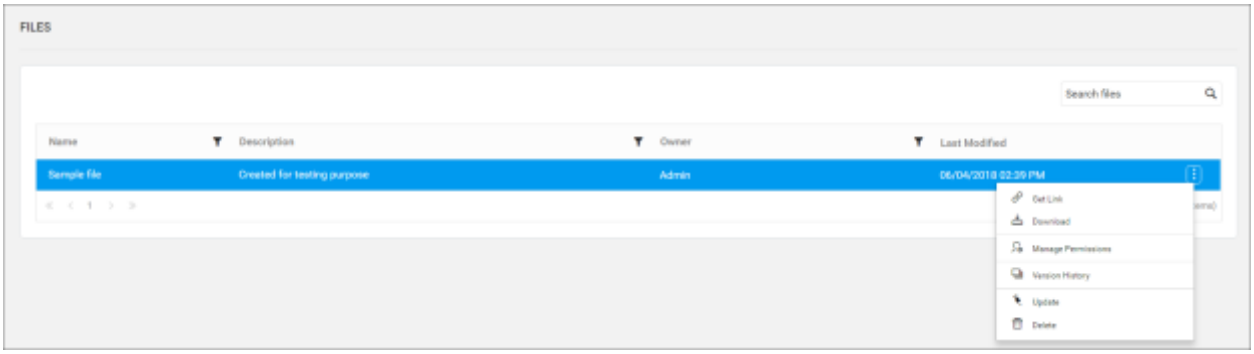
Click the Actions button in the data sources grid context menu and select Delete to delete the data source.



Manage Files

This section explains on how to add, update, download, delete files and also on how to view version history of files in the Syncfusion Dashboard Server.

Files that are accessible by the user depending on the user's permission is displayed in the files page.

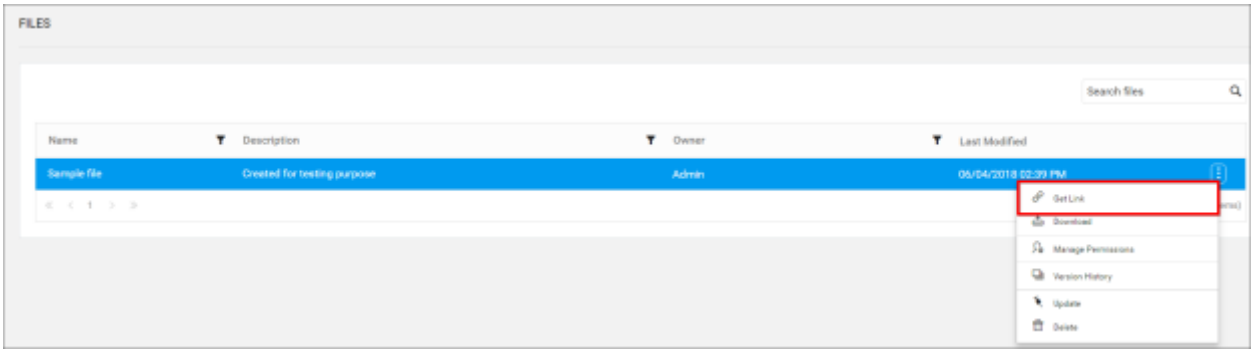


Get Link

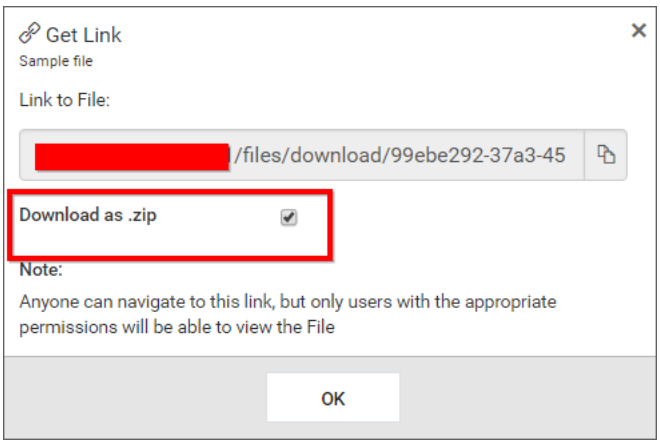
Files can be either downloaded in .zip format or viewed in the browser using Get Link option which is available for all the files in the grid.

Follow the below steps to get the link of a file.

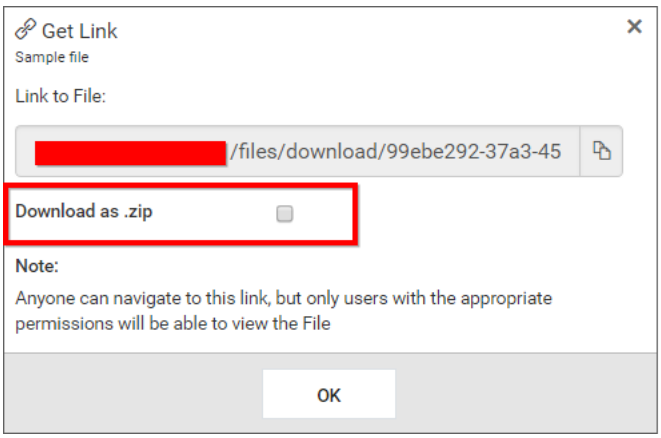
1. Click on the context menu of the respective file and choose Get Link option.



2. The Get Link dialog will be opened. Check the Download as .zip to download the file in .zip format.



To view the respective file in browser, uncheck the Download as .zip option in the dialog.

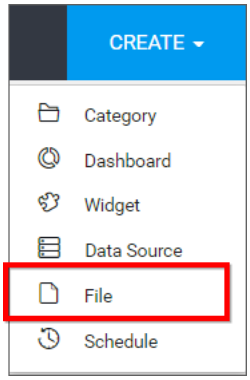


Add Files

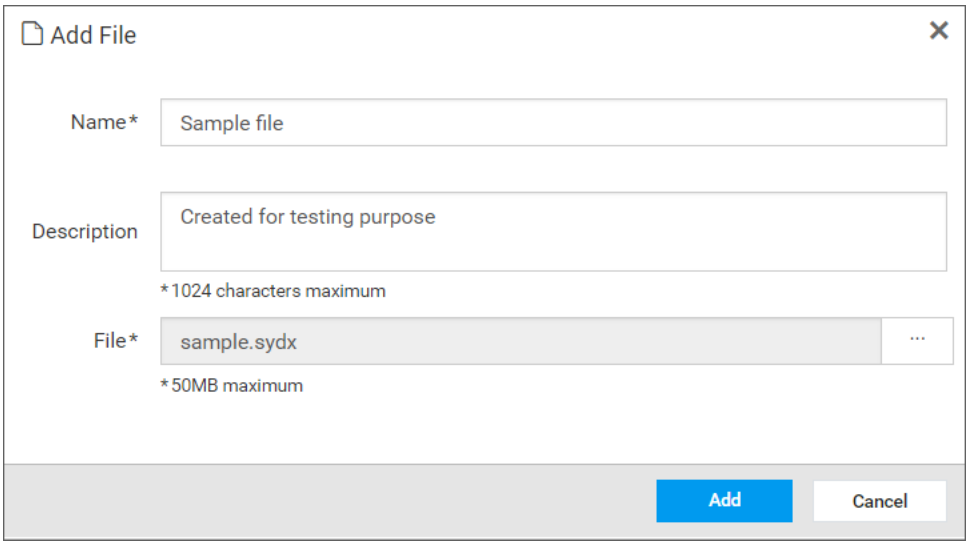
Files can be added in the Dashboard Server only when you have Create All Files permission.

Steps to add a file

1. Click on the Create button in the menu and select File to add a file.



2. Fill the form with name, description of file and upload the file in the Add File dialog box. Any file can be uploaded into the Dashboard Server and the file can be linked/added to the dashboards.

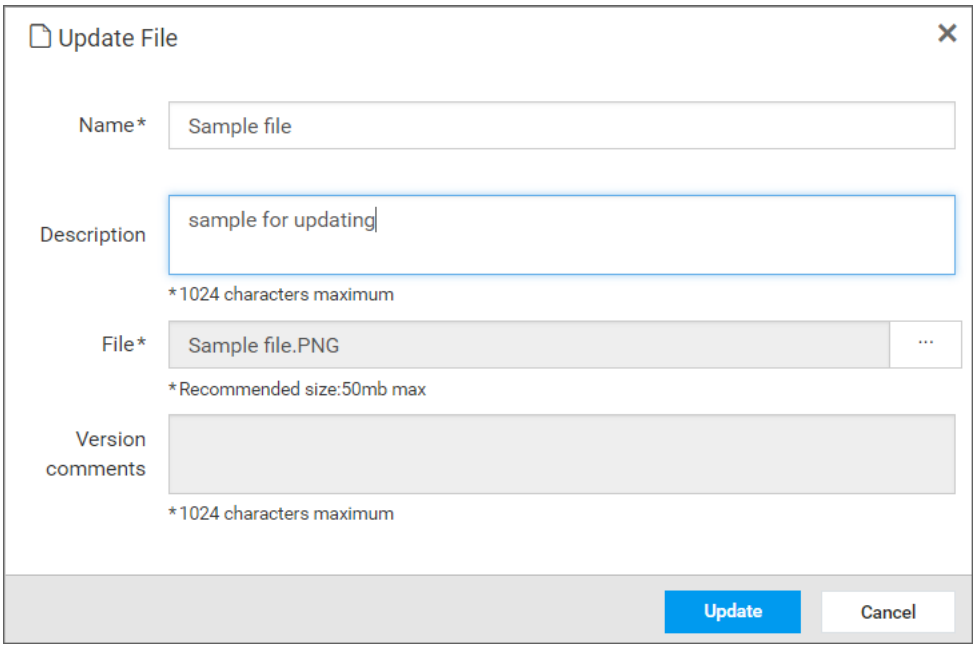


3. When clicking on Add, the file will be added to the Dashboard Server and it can be used in any one of the Dashboards.

Note: Read Write Delete permission for that Specific File will be added for the user who created the file.

Update Files

Name, description and the physical file can be changed in the update file dialog box.

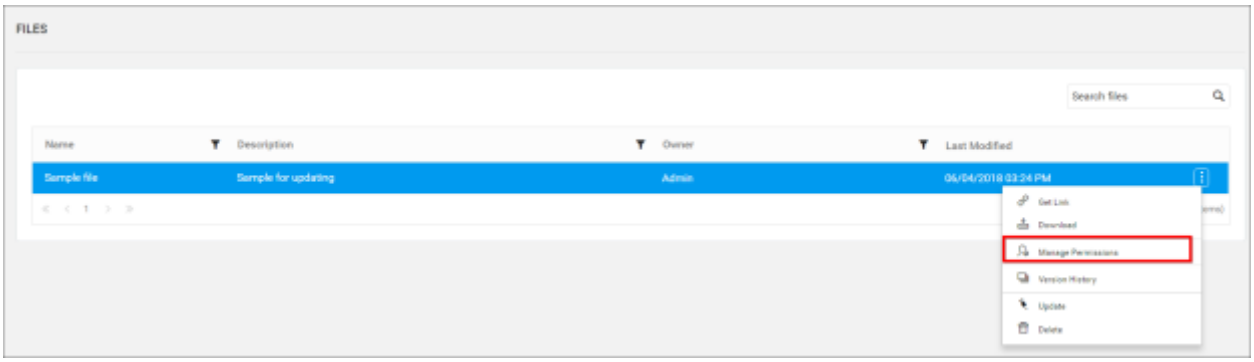


Share Files

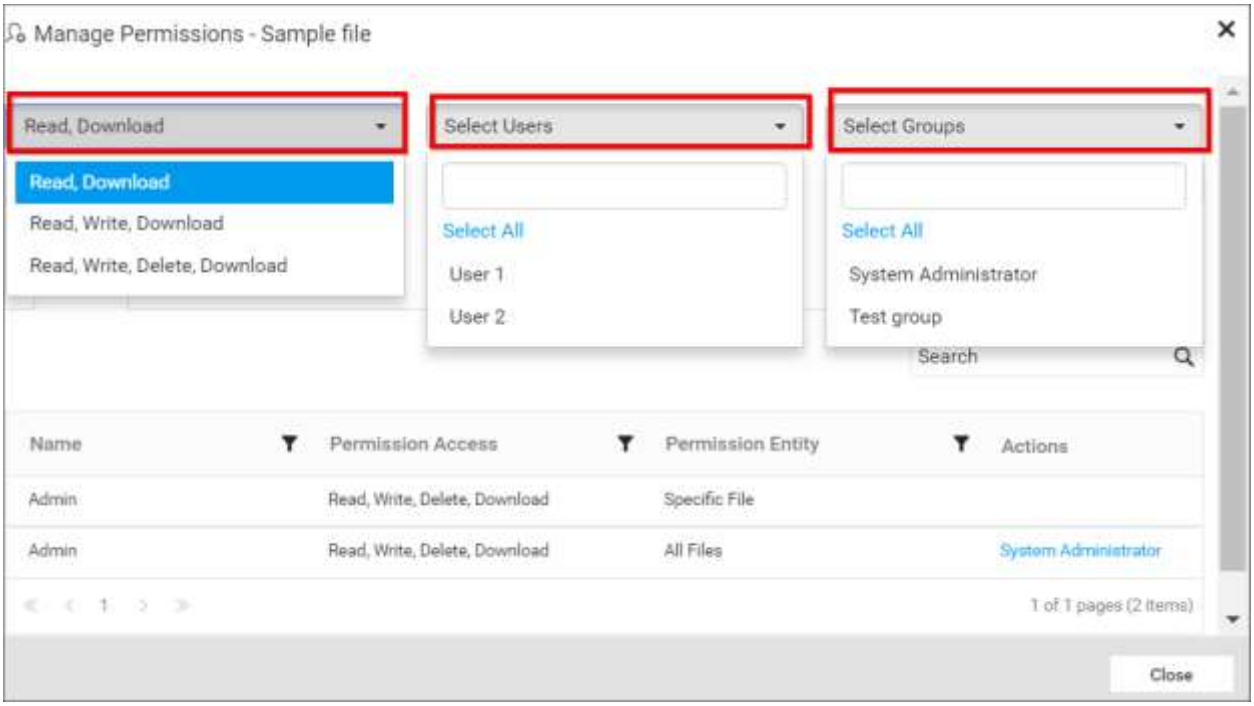
This section explains on how to share files with the other users in the Dashboard Server.

Steps to share a file

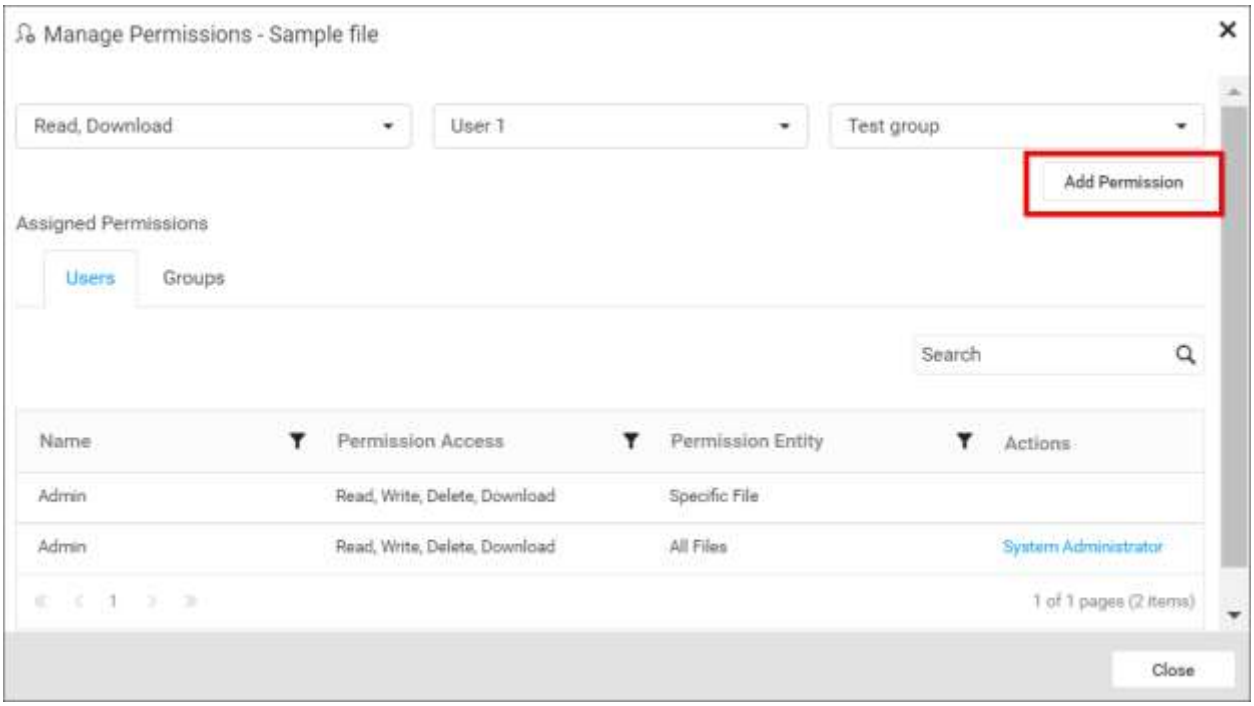
1. Click the Actions button in the Files grid context menu and select Manage Permissions option.



2. Select the permission access from the Select Access dropdown and select the users or groups to share the files.



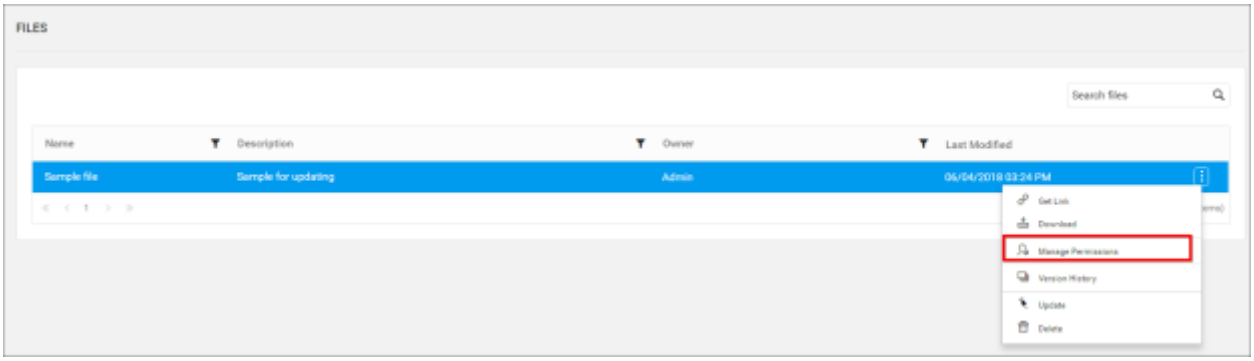
3. After selecting the access and users or groups, click on the Add Permission button.



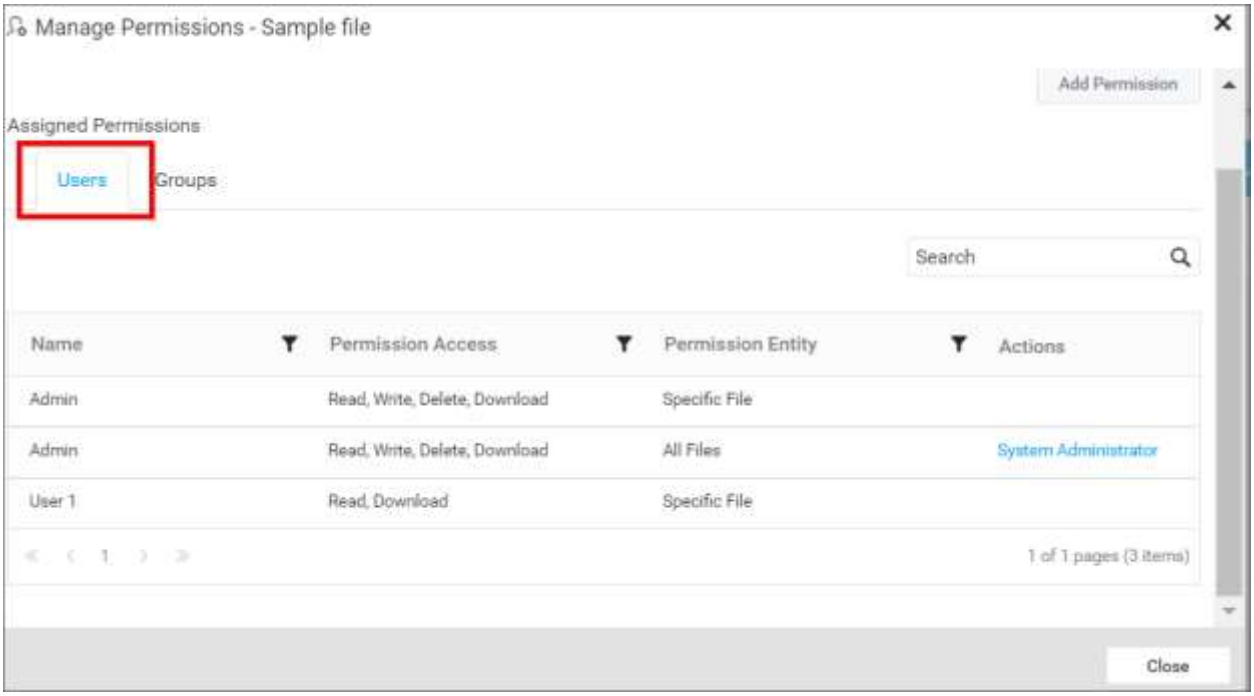
Note: Only the user who created the file can share the file with other Dashboard Server users.

View Permission

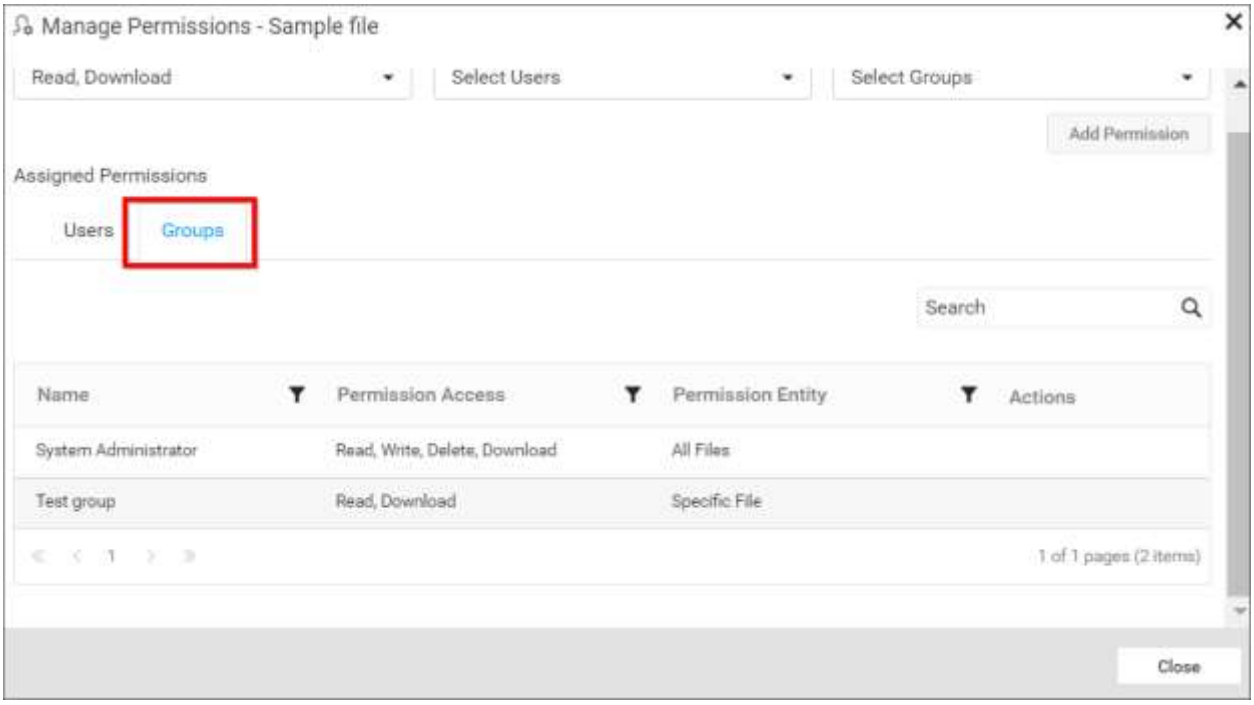
If the user is not an owner of the File, user can view the assigned permissions of the file by clicking the Manage Permissions option in the Files grid context menu.



The permission availed to the users can be viewed in the Users tab.

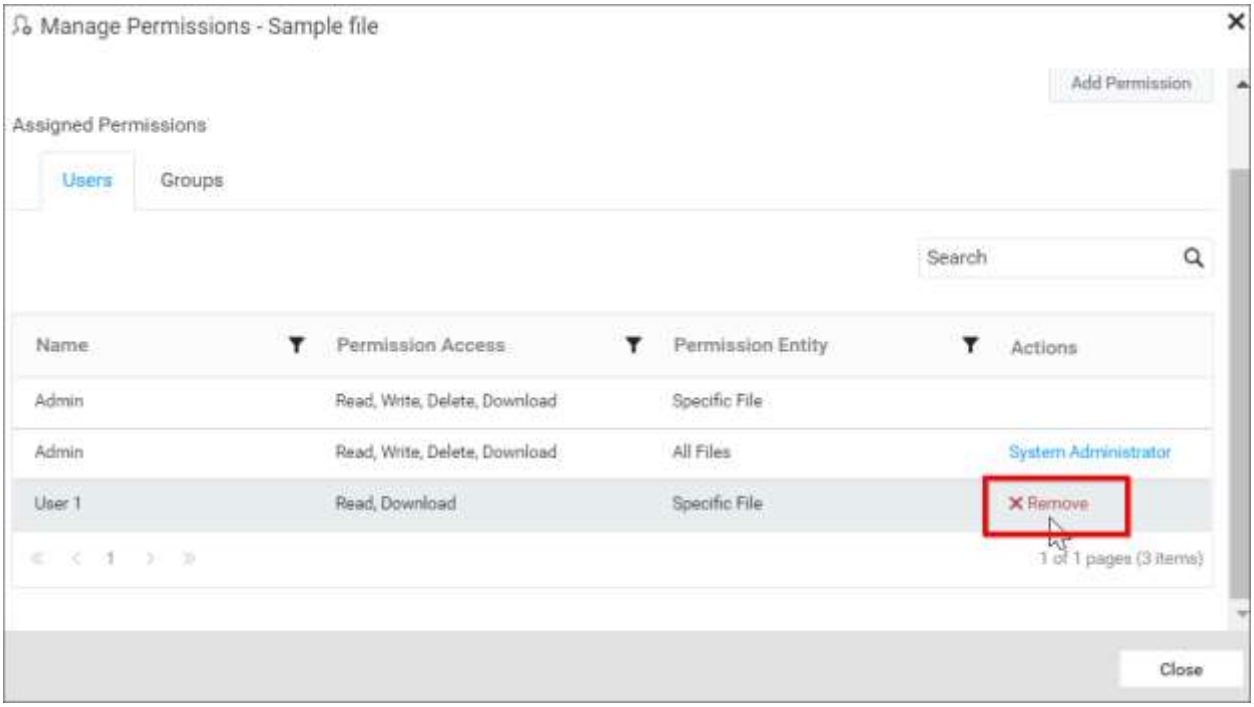


The permission availed to the groups can be viewed in the Groups tab.



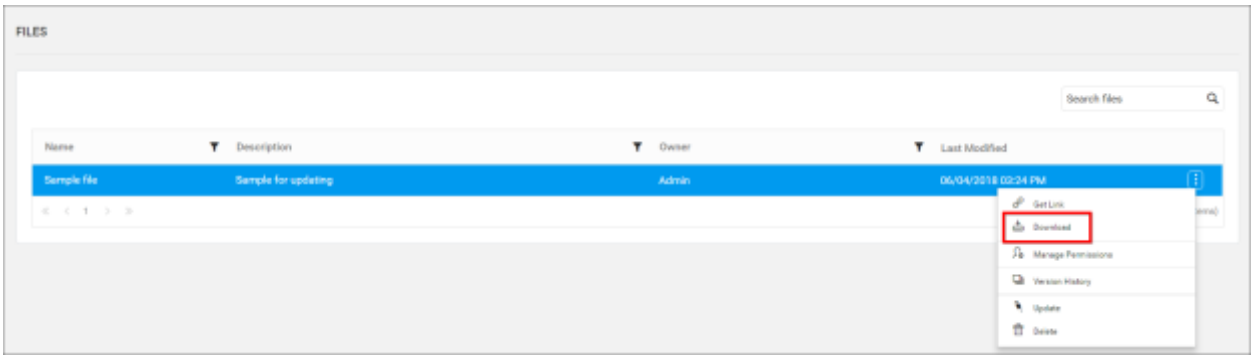
Remove Permission

The user who created the file can remove the shared file permissions using the Remove option in the Actions column of the each permissions.



Download Files

Click the Actions button in the files grid context menu and select Download to download the file compressed in a .zip format.



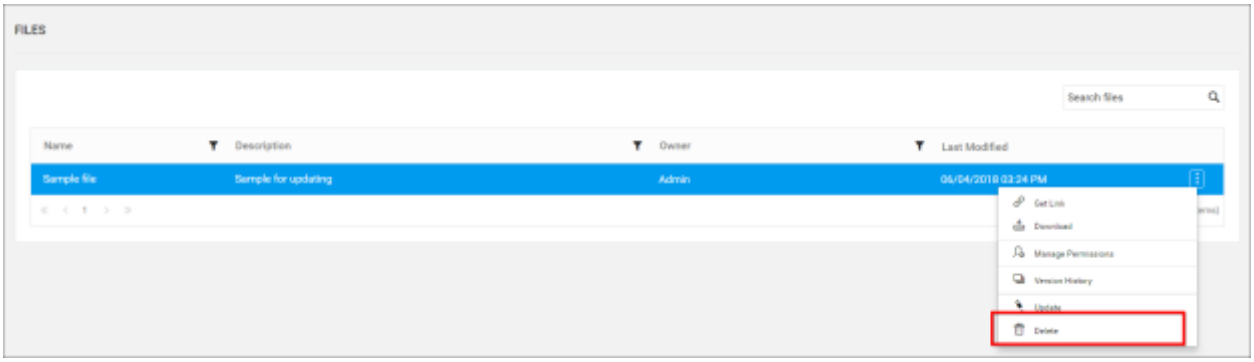
Version History

Versions and file logs for each file are maintained in the Dashboard Server for every changes in the file. Check Version History section in Manage Dashboards for more details.

Delete Files

Files can also be deleted from the Dashboard Server when they are no longer required.

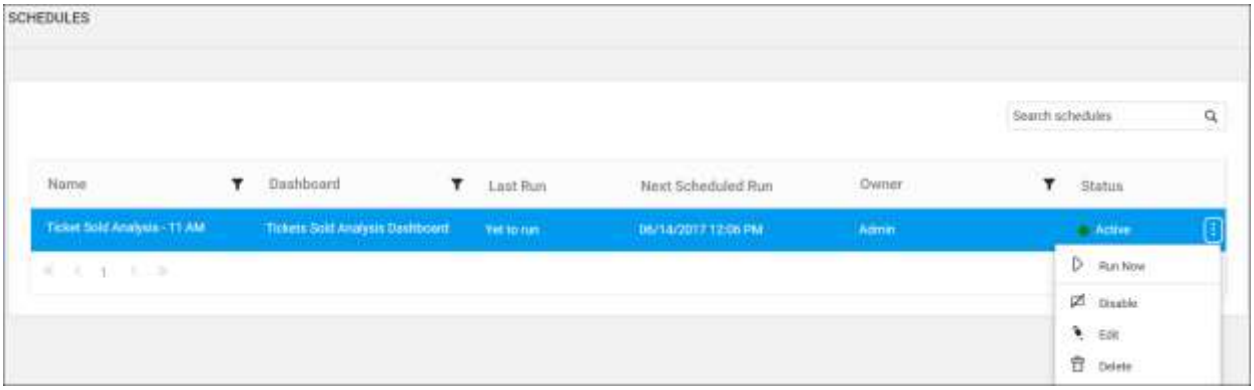
Click the Actions button in the files grid context menu and select Delete to delete the file.



Manage Schedules

This section explains on how to add, edit, delete schedules and also on how to run the schedules on demand and enable or disable schedules in the Syncfusion Dashboard Server.

Schedules' page displays the schedules that are accessible by the user depending on the user's permission.



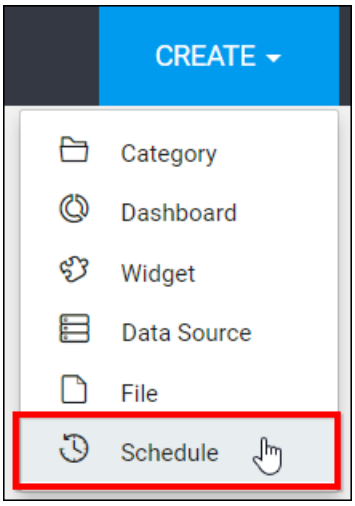
Add Schedules

Schedules can be created only if the user has Create All Schedules permission. Schedules can be created in two ways,

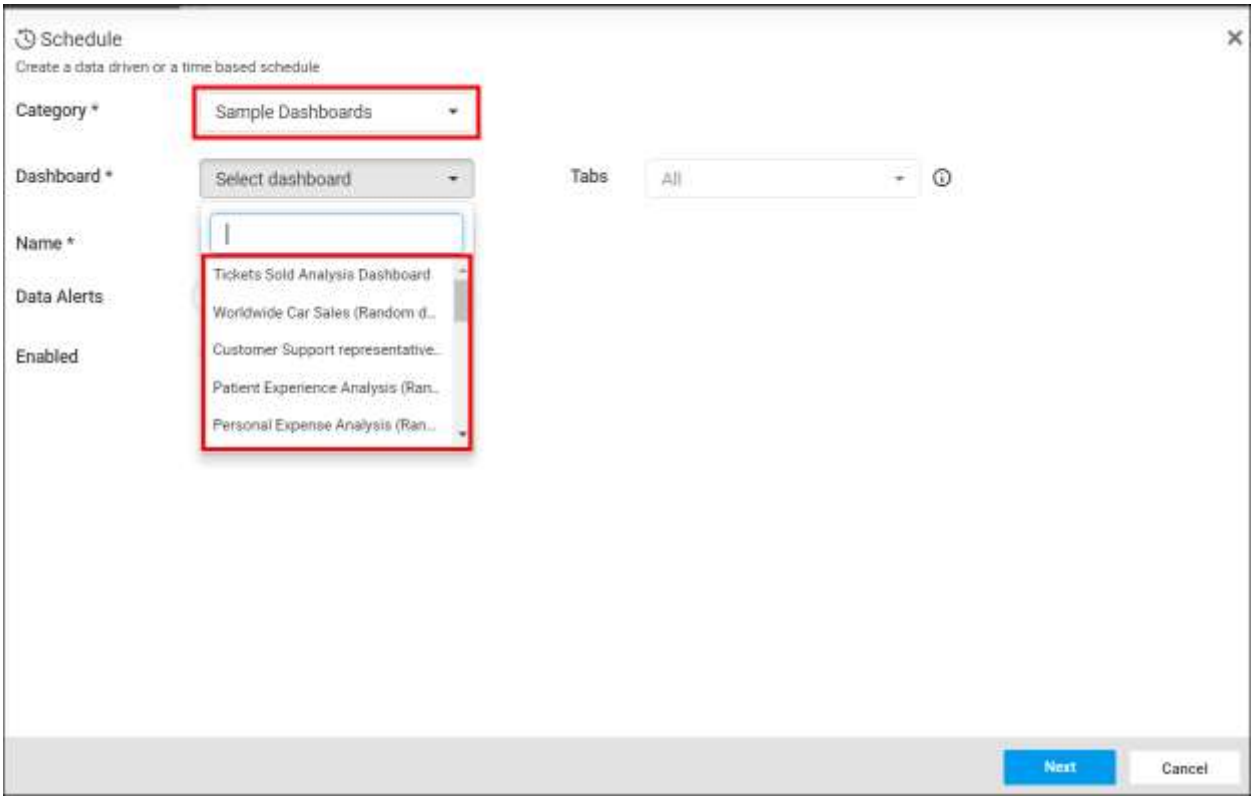
1.Add schedule from create menu. 2.Add schedule from context menu of the respective dashboards.

Add schedule from create menu

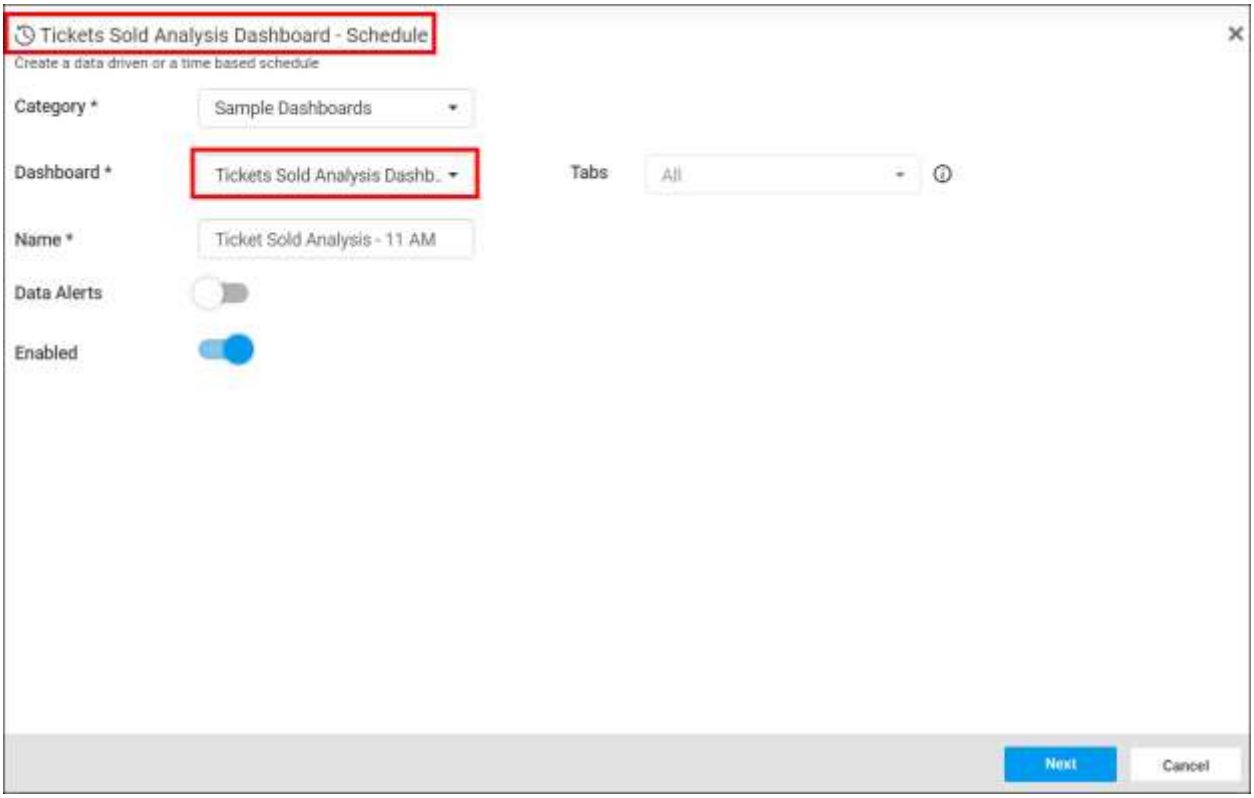
- Click on Create button in the menu and select Schedule option to schedule the dashboards.



- Select the required category from Category dropdown. After selecting the category, corresponding dashboards under that selected category will be displayed in the Dashboard dropdown,

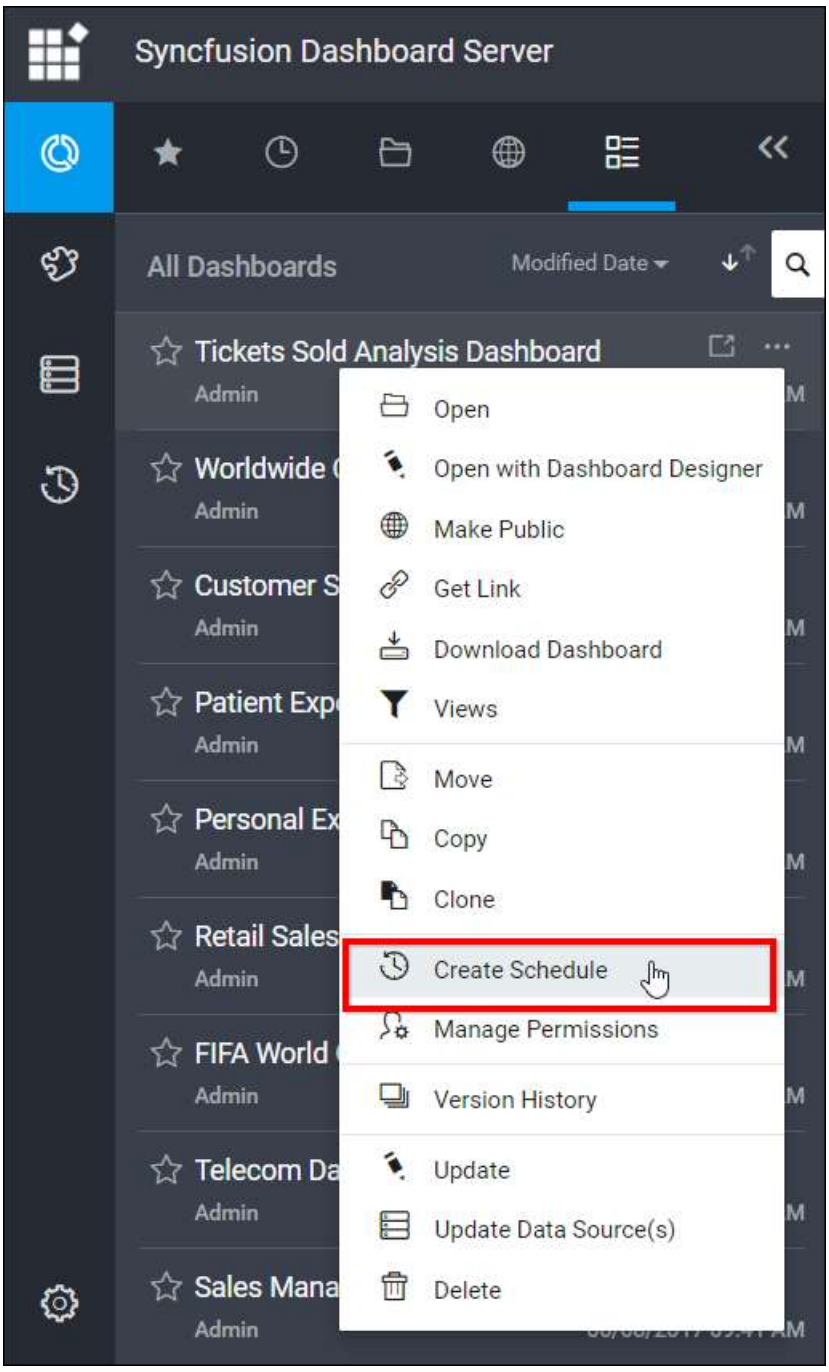


- Select the required dashboard from the dropdown.

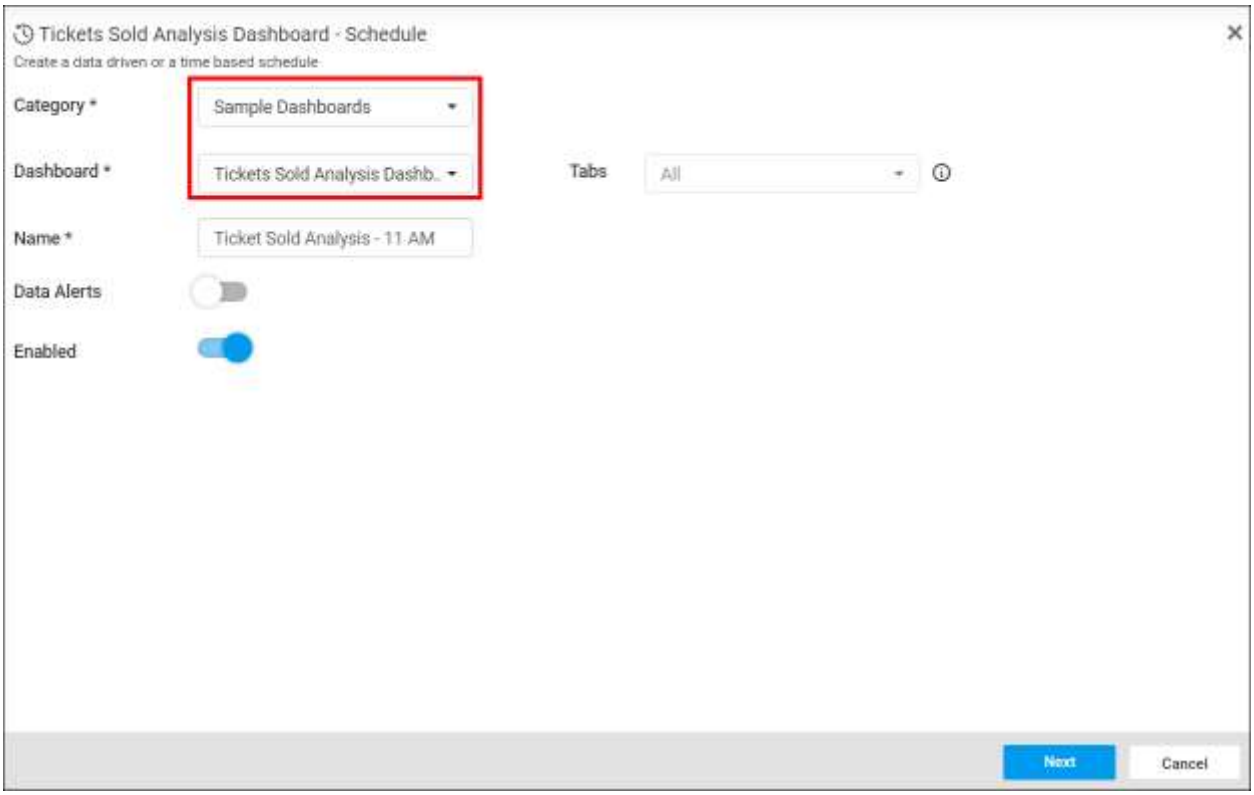


Add schedule from context menu of the respective dashboards

- Click on Actions button in the schedules grid context menu and select Create Schedule to schedule the corresponding dashboard.

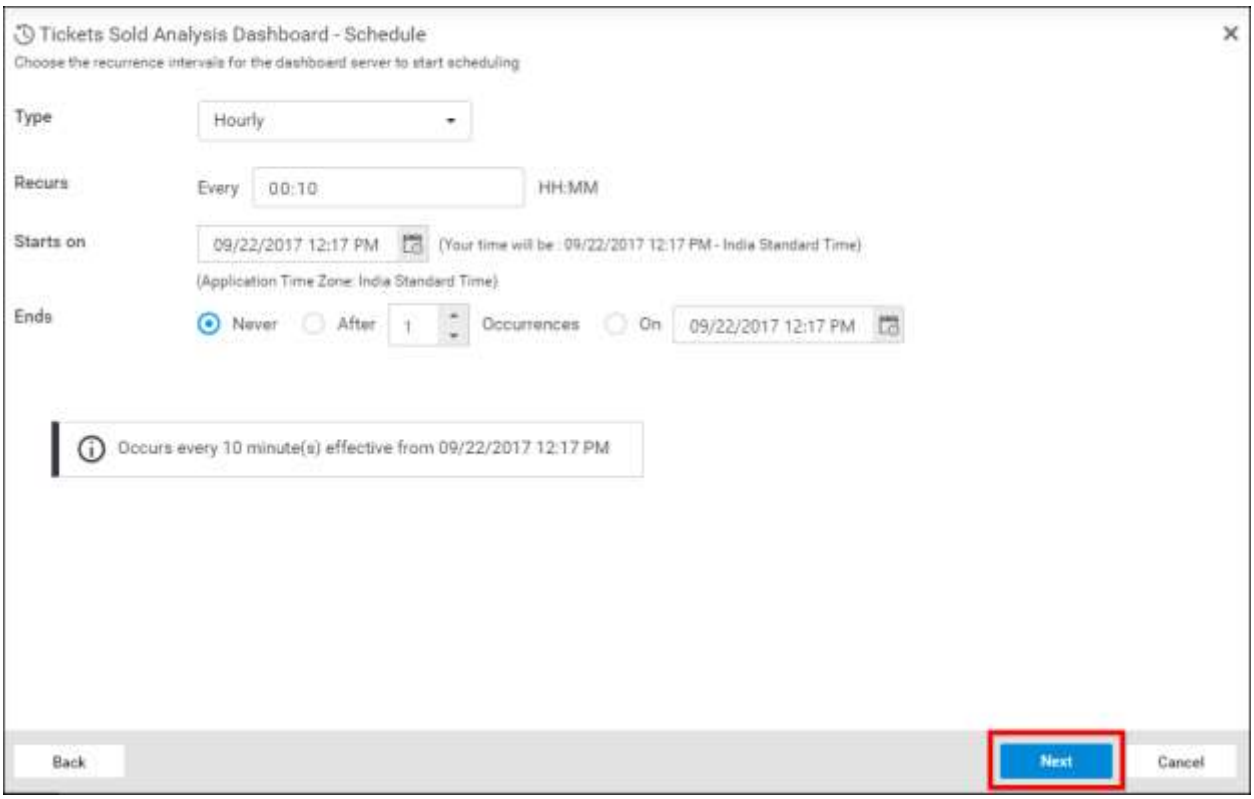


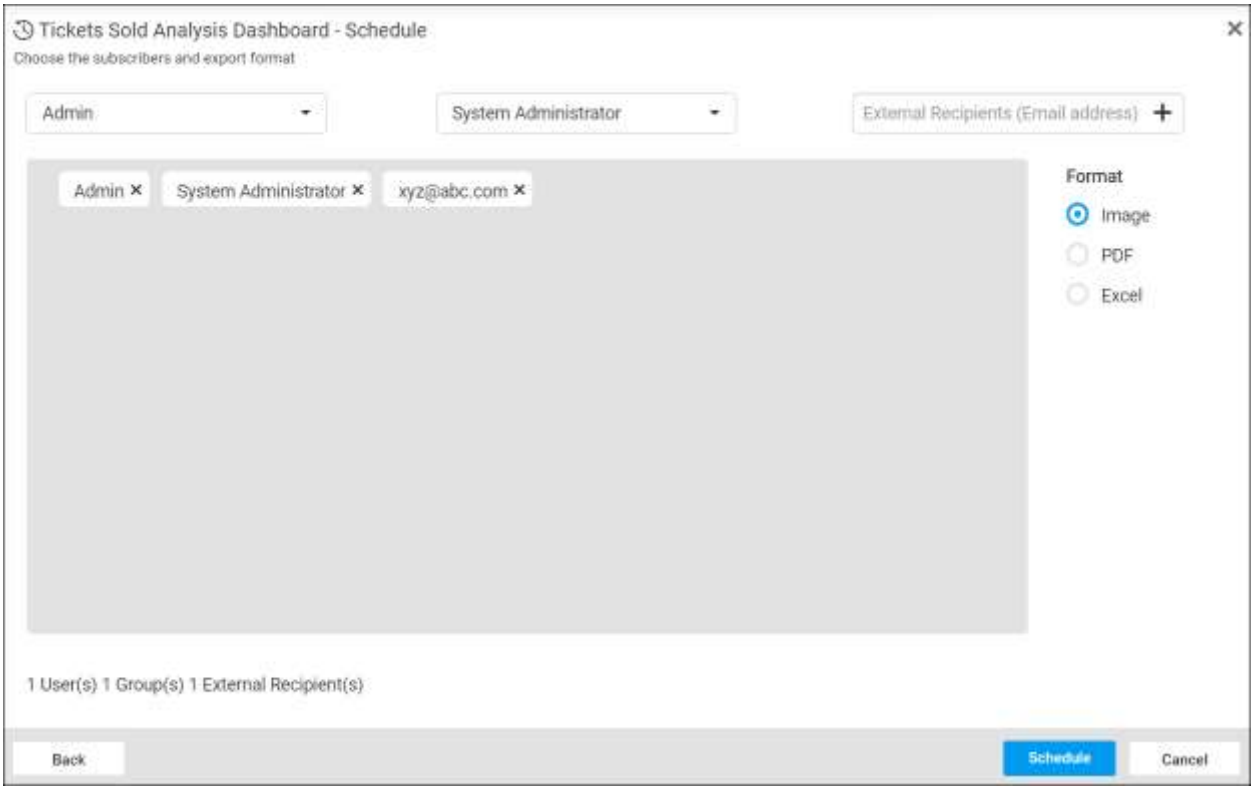
- Once dialog was opened, the category and dashboard values are selected by default,



Note: Categories or Dashboards can be changed from schedule dialog box itself.

- After adding schedule details, click on the Next button in schedule dialog.
- Select the recurrence type, recurrence, start and end dates, export formats and the users to which the exported dashboards have to be emailed in the Add Schedule dialog box.
- Dashboards can be scheduled hourly, daily, weekly, monthly and yearly
- Please refer the link to know more about the export formats of dashboard.
- Application Time Zone is displayed below the date picker. Start time of the schedule is converted to client Time Zone and shown in the right side for the user's convenience
- Exported dashboards can be sent to individual users or groups or to external recipients along with the link the dashboard in the dashboard server through emails.





- When clicking on Schedule, the dashboard is scheduled in the selected recurrence.

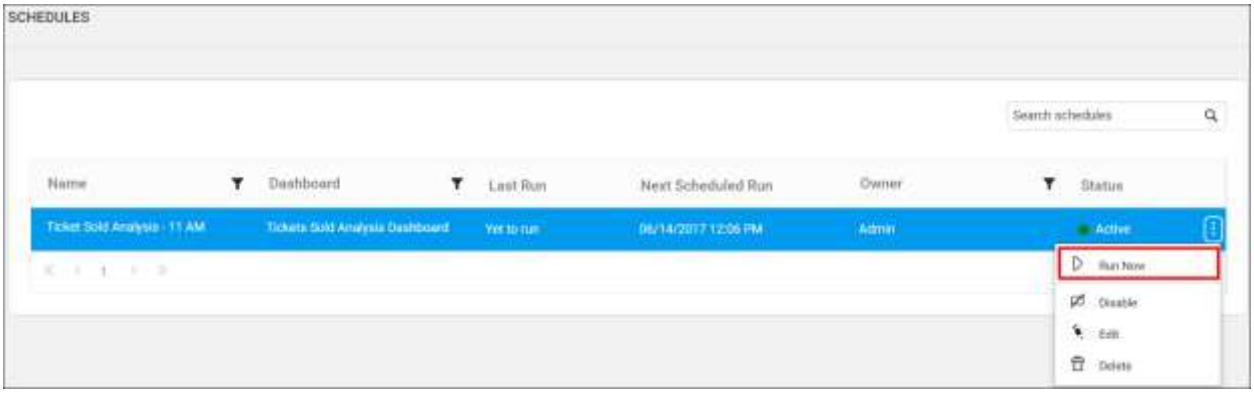
Note: Read Write Delete permission for that Specific Schedule is added to the users by the person who created the schedule.

Edit Schedules

Category, dashboard, name, recurrence type, recurrence, start and end dates, export format and the recipients can be changed in the Edit Schedule dialog box.

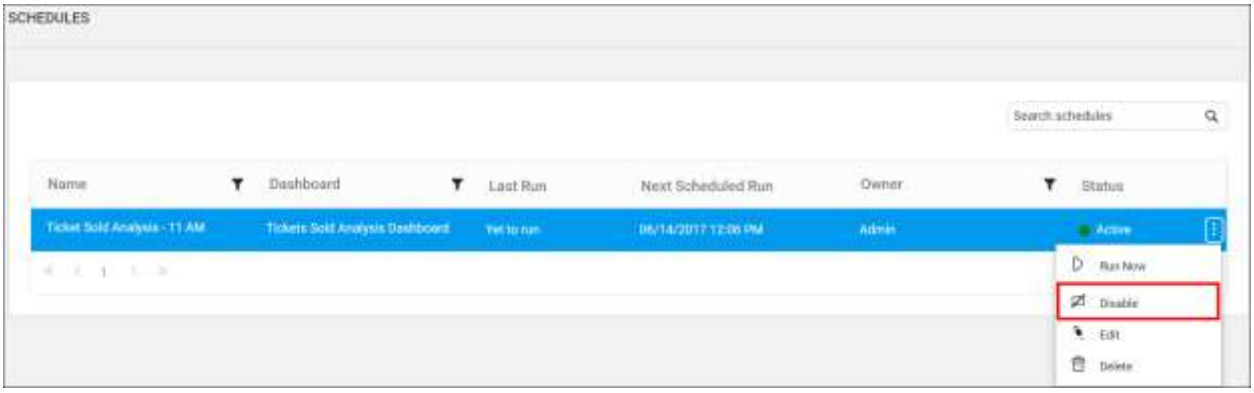
Run Now

Schedules can be made to run on demand by using Run Now option in the schedule grid context menu. Dashboard get exported in the format specified and sent to the recipients along with the link the dashboard in the dashboard server through emails.



Enable or Disable Schedule

Schedules can be disabled at any time, which ignores any next occurrences. When enabled, it finds the next occurrence and run accordingly.

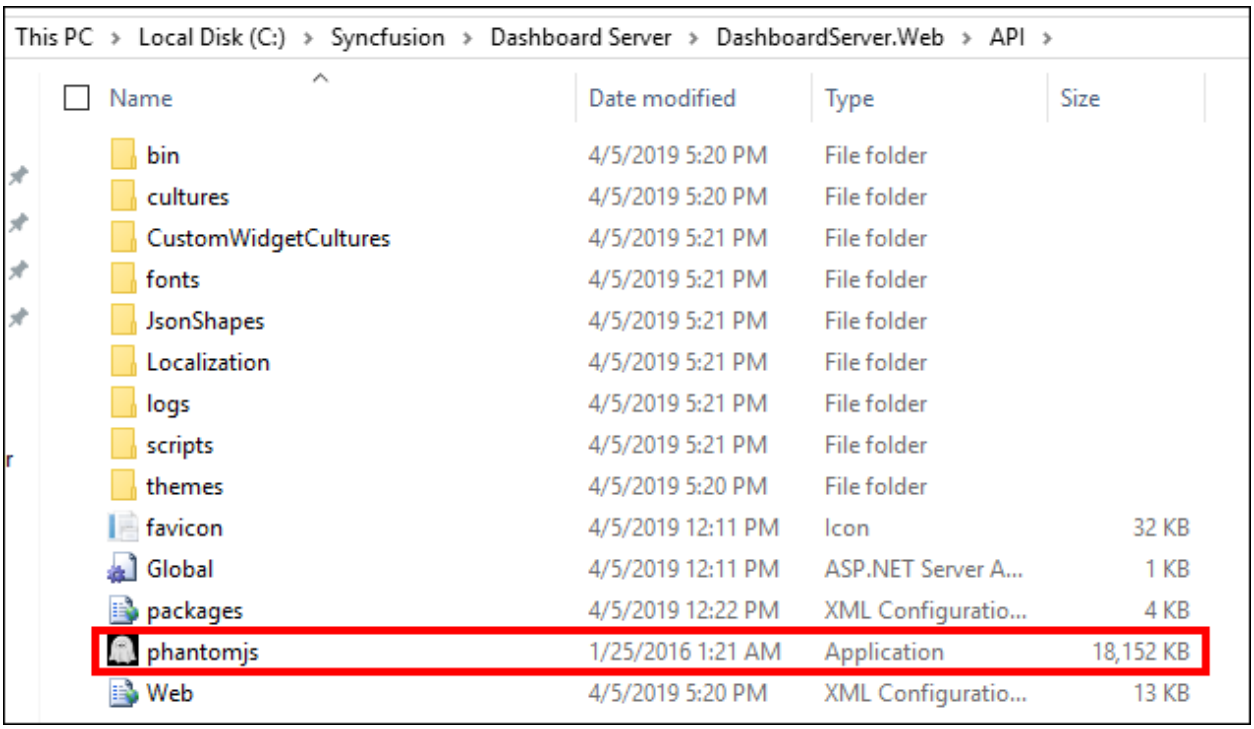


Export Format

Dashboards can be exported as Image, PDF and excel.

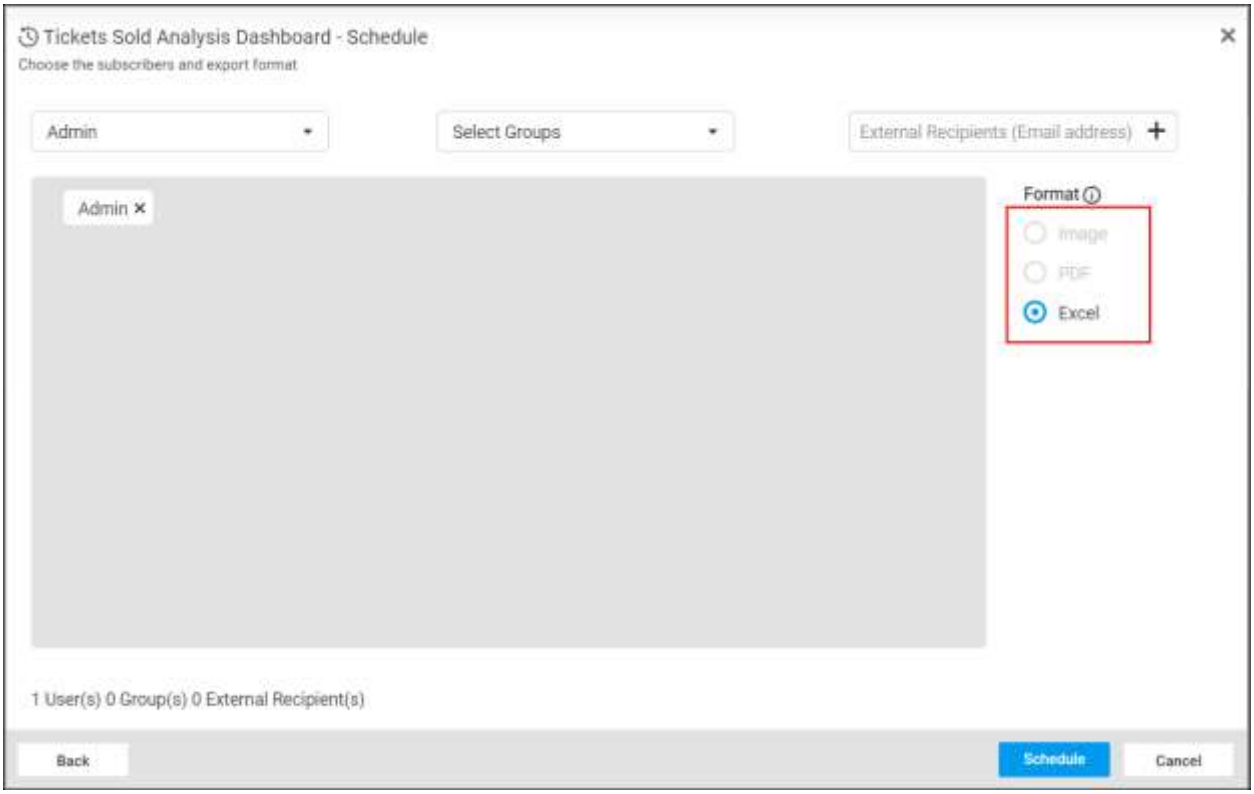
phantomjs.exe is a third party utility, which is mandatory to export the Dashboards as Image or PDF format, which can be found in the following location

"{windows_drive}\Syncfusion\Dashboard Server\DashboardServer.Web\API".



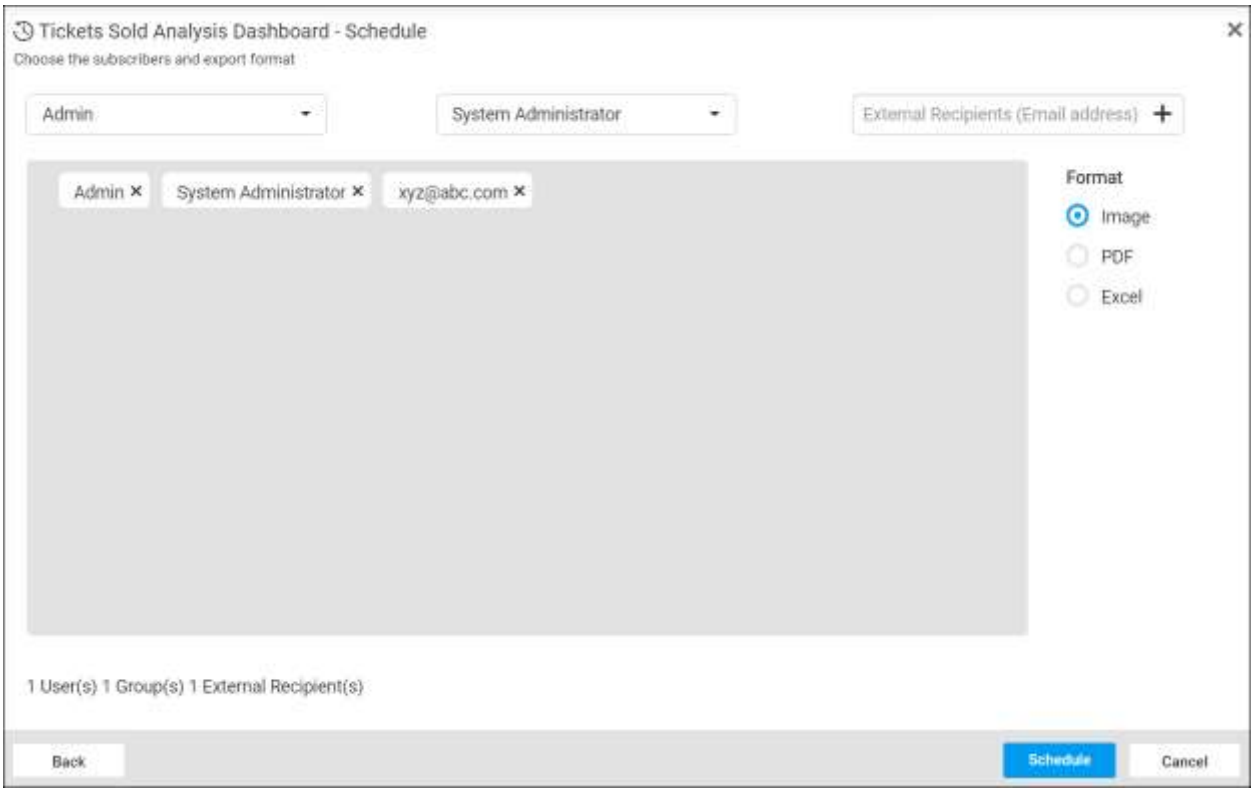
If it does not found in the below location the export options for Image and PDF will be disabled, only the Excel option will be enabled in the dialog box.

"{windows_drive}\Syncfusion\Dashboard Server\DashboardServer.Web\API"



If it is found in the below location, the dashboard can be exported in Image and PDF. The dashboards can be exported as Excel irrespective of the phantomjs.exe.

"{windows_drive}\Syncfusion\Dashboard Server\DashboardServer.Web\API"

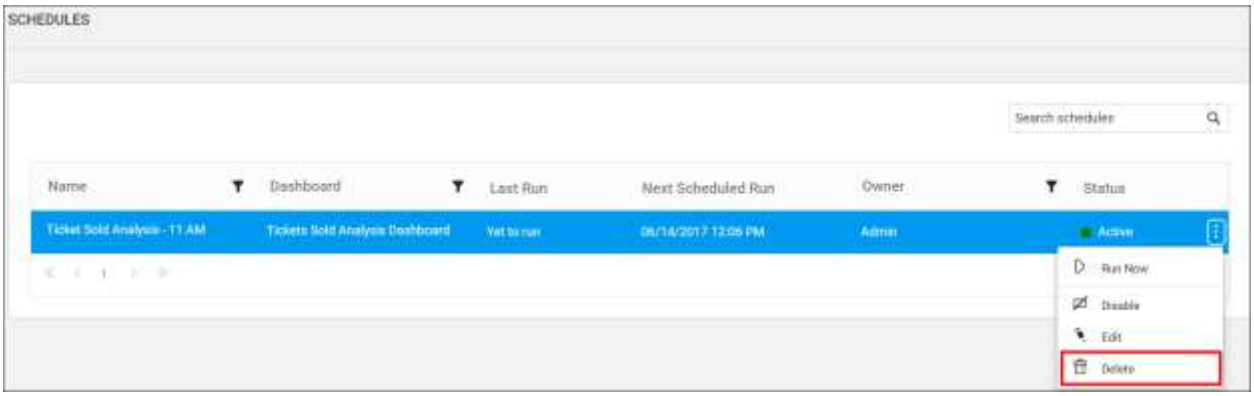


For local and Virtual Machines, the phantomjs.exe can be downloaded from here For Azure app services, the phantomjs.exe can be downloaded from here.

Delete Schedules

Schedules can be deleted from the dashboard server when it is no longer required.

Click the Actions button in the schedules grid context menu and select Delete to delete the schedule.



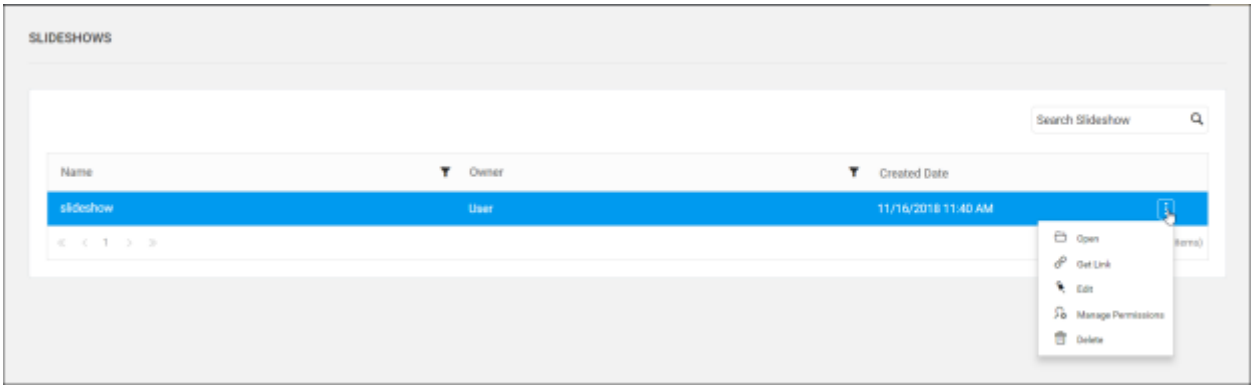
Slideshows

Slideshows are a presentation of dashboards/widgets that can rotate them with regular intervals.

Manage Slideshows

This section explains on how to add, view, share, update and delete slideshows in the Syncfusion Dashboard Server.

Slideshows that are accessible by the user depending on the user's permission will be displayed in the slideshows page.

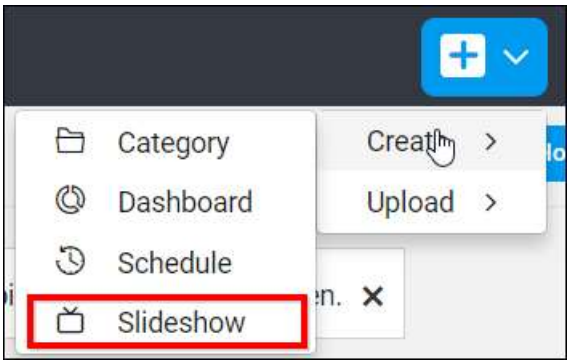


Add Slideshows

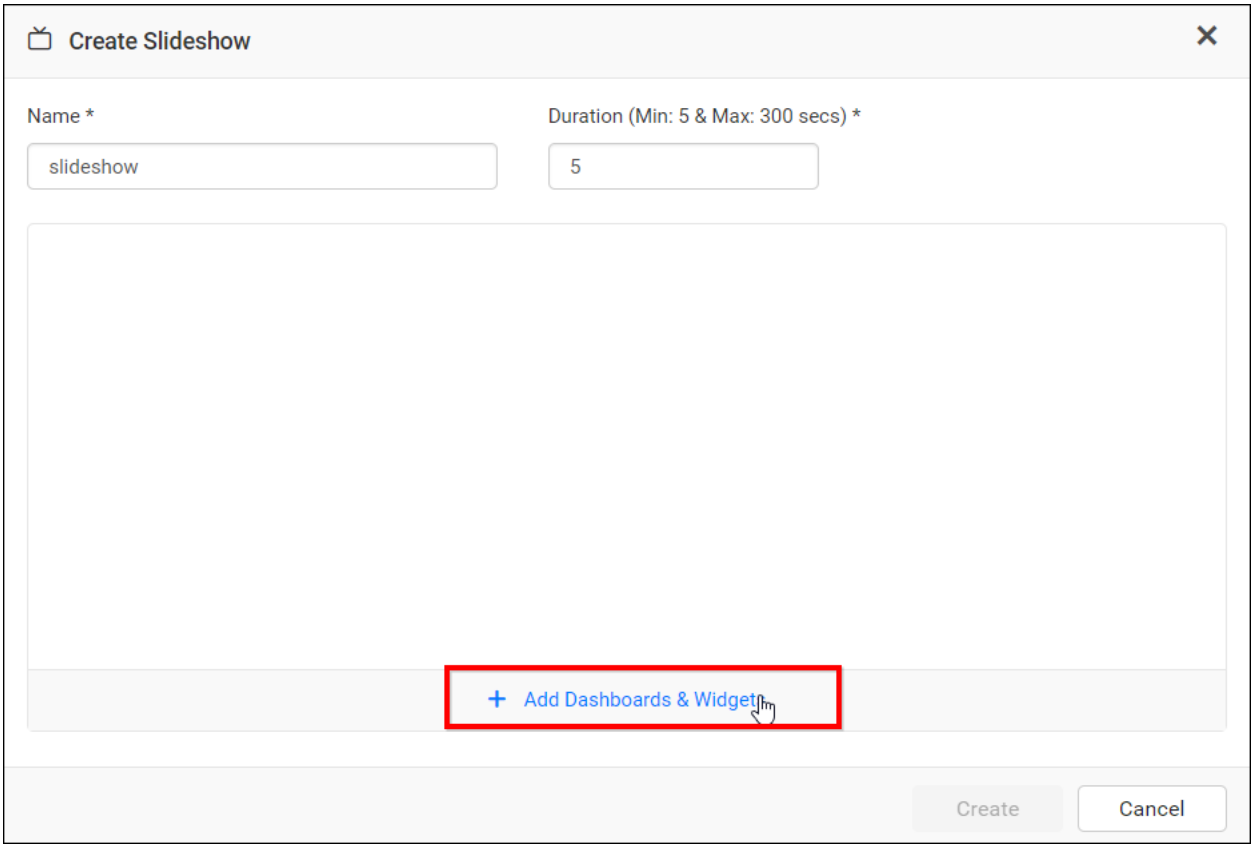
This section explains on how to add slideshow in the Syncfusion Dashboard Server. Slideshow can be added in the Dashboard Server only when you have Create permission All Slideshow.

Steps to add a Slideshow

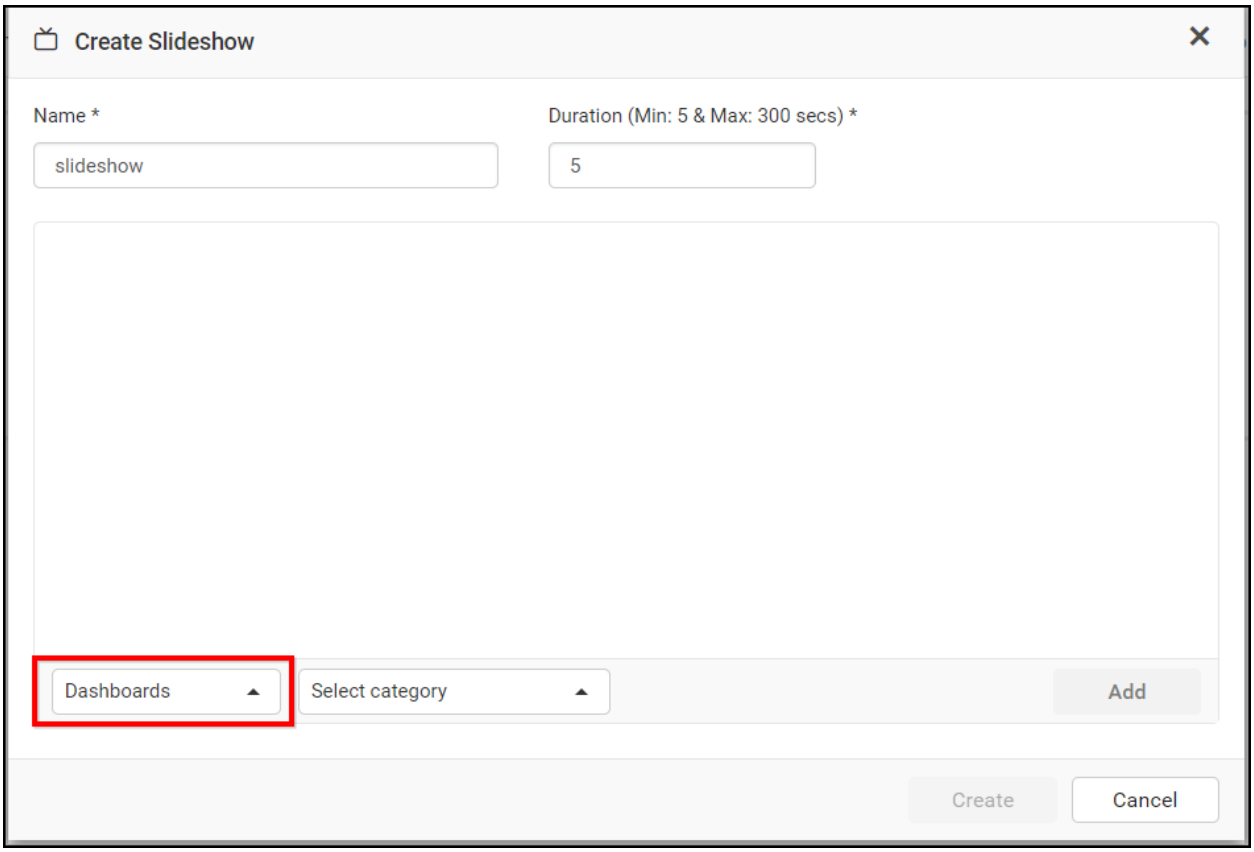
1. Click on the Create button in the menu and select Slideshow to add a slideshow.

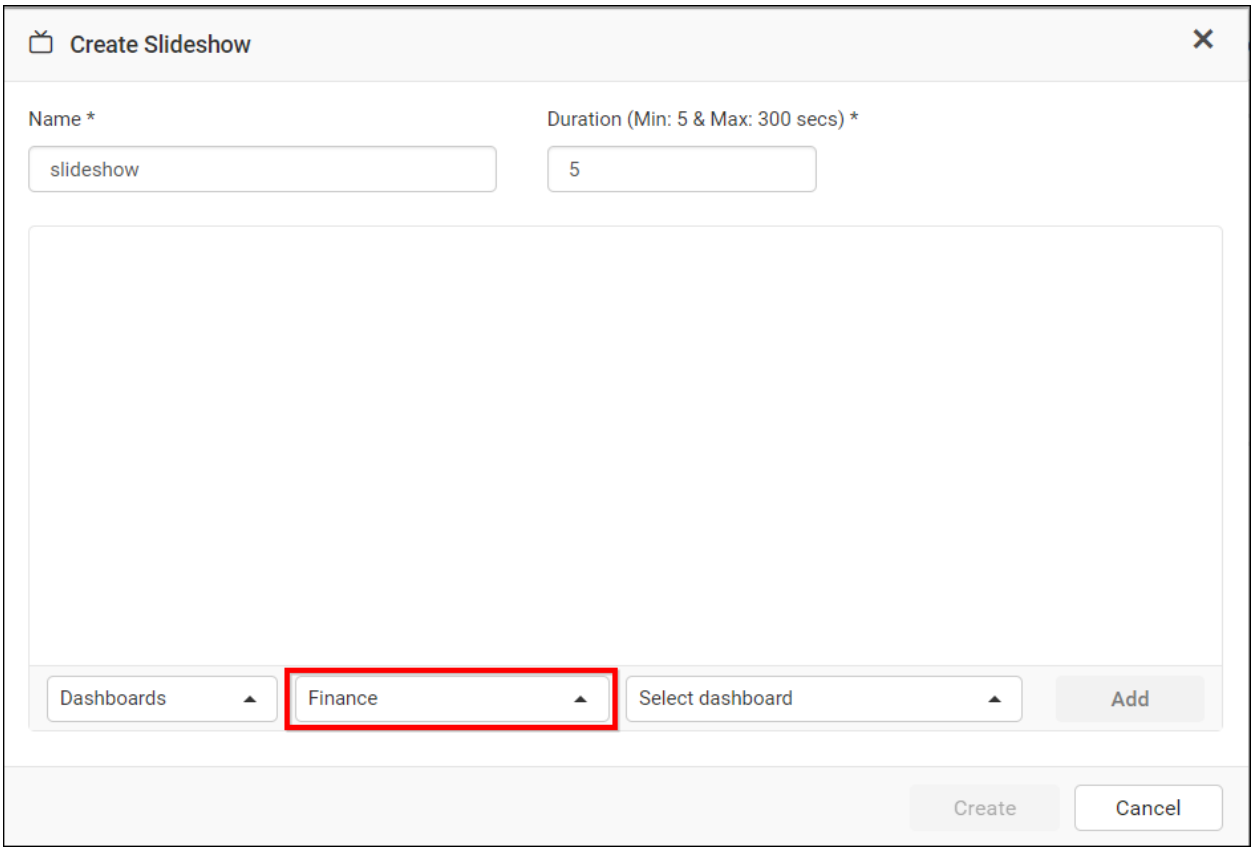


2. Fill the form with Name and Duration for slideshow and click on the Add Dashboards & Widgets in the Create Slideshow dialog box.

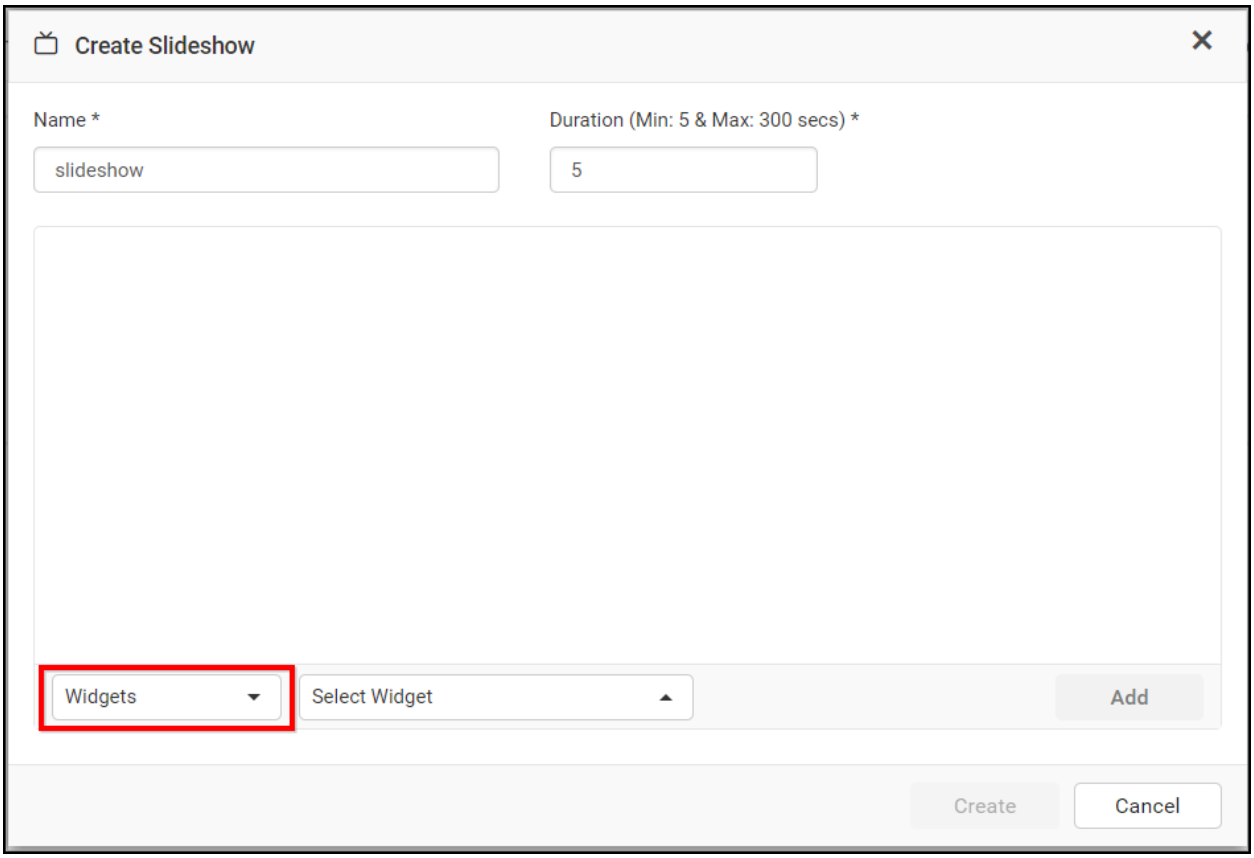


3. Select the required type Dashboards or Widgets from dropdown.
4. Dashboards - If the Dashboards type is chosen, after that select the category from Select Category dropdown and corresponding dashboards under that selected category will be displayed in the Select Dashboards dropdown.

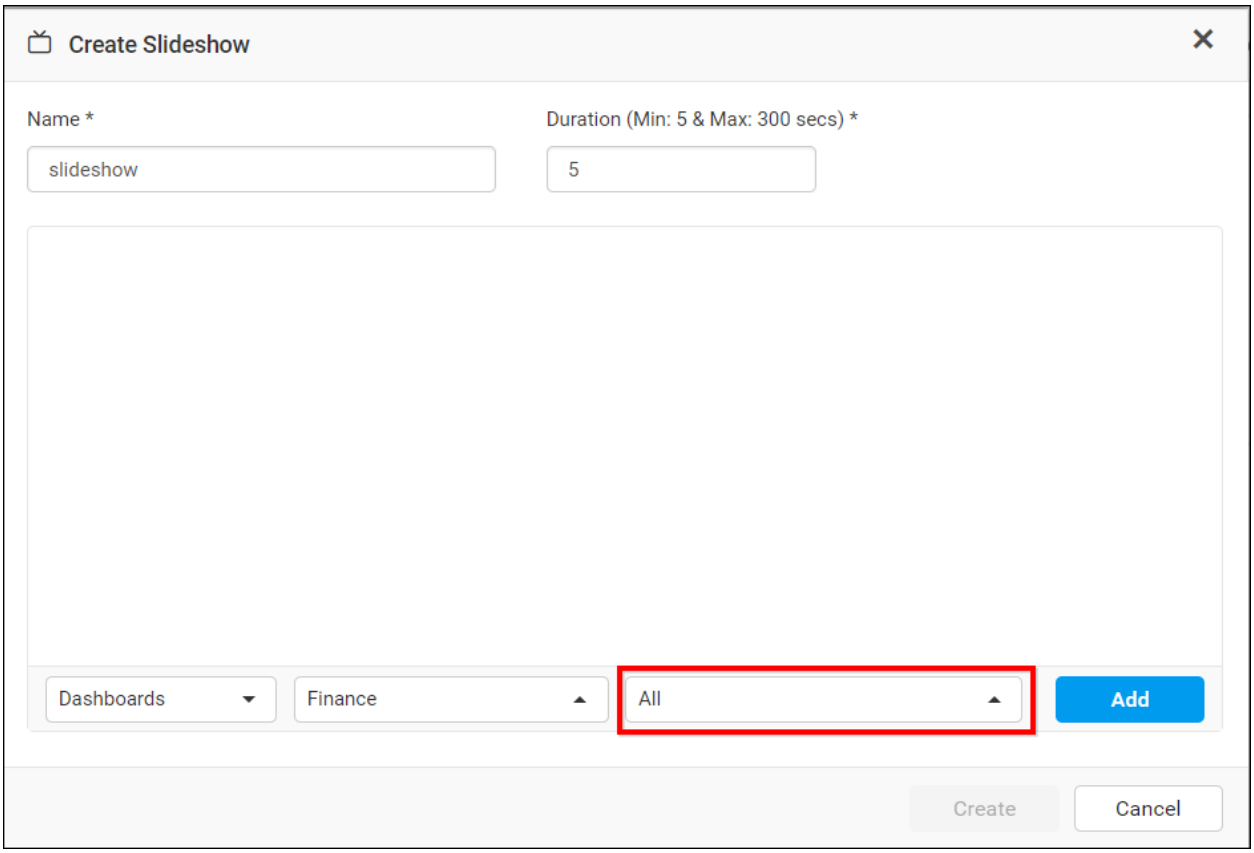




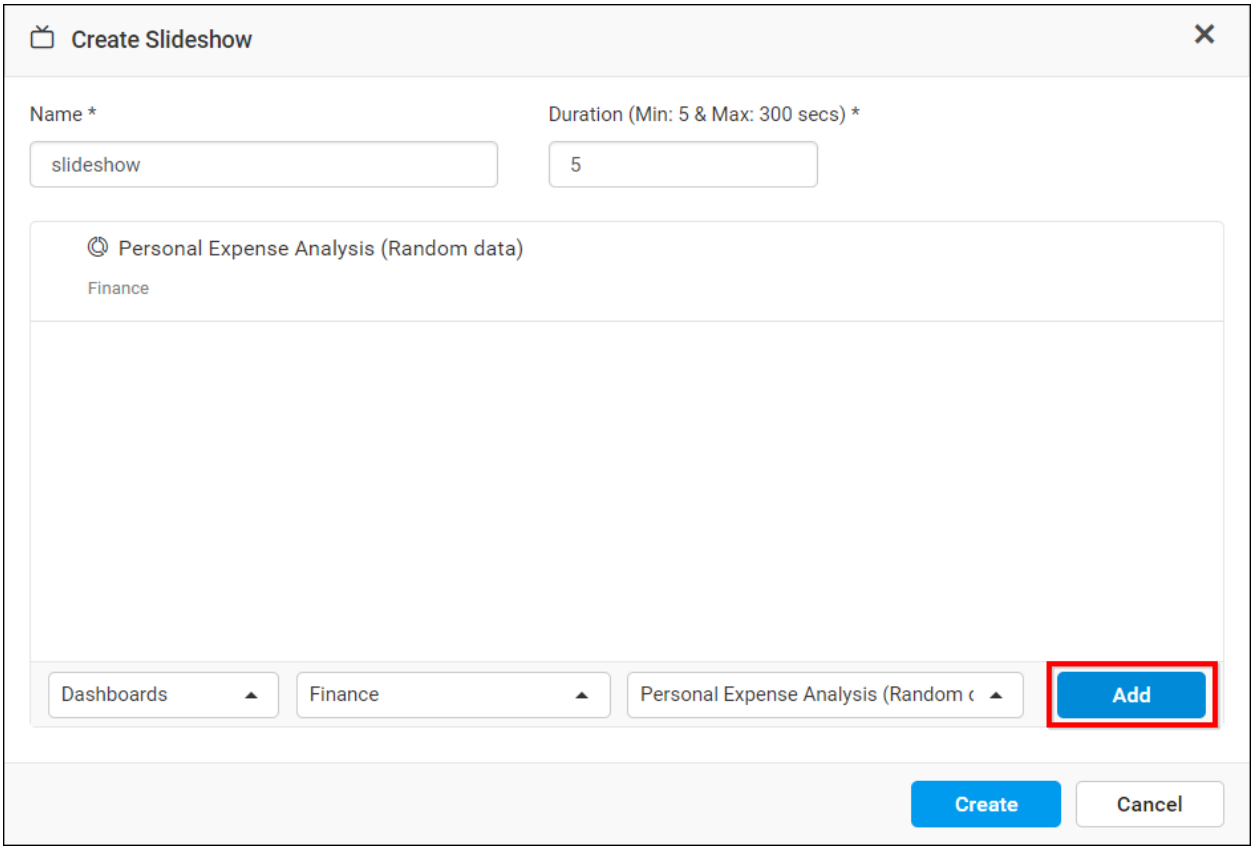
2. Widgets - If the Widgets type is chosen, after that select the widgets from Select Widgets dropdown.



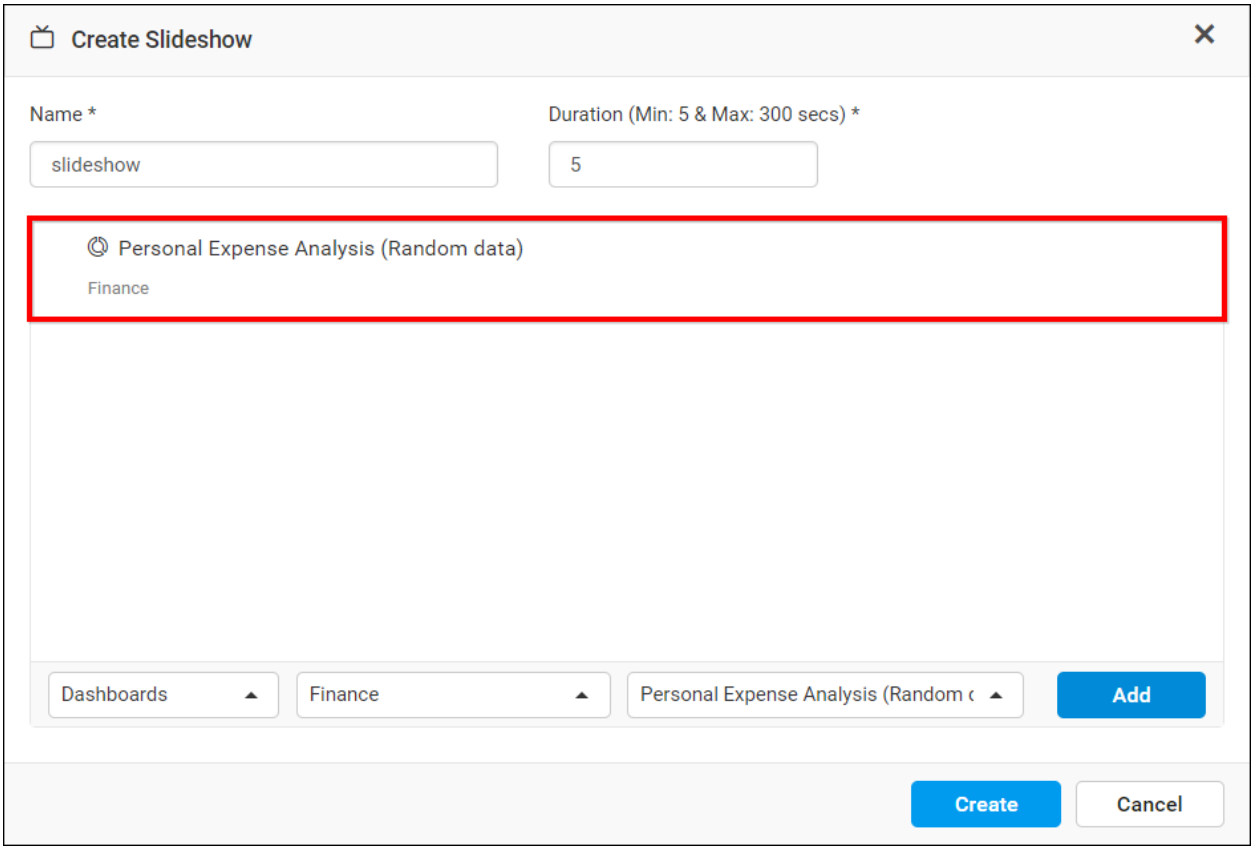
4. After selecting the dashboards, widgets from Select Dashboards and Select Widgets dropdown respectively Add button will enable. In the Select Dashboards, if All option is chosen, it will add all dashboards in that category. In the Select Widgets, if All option is chosen, it will add all widgets to the slideshow.



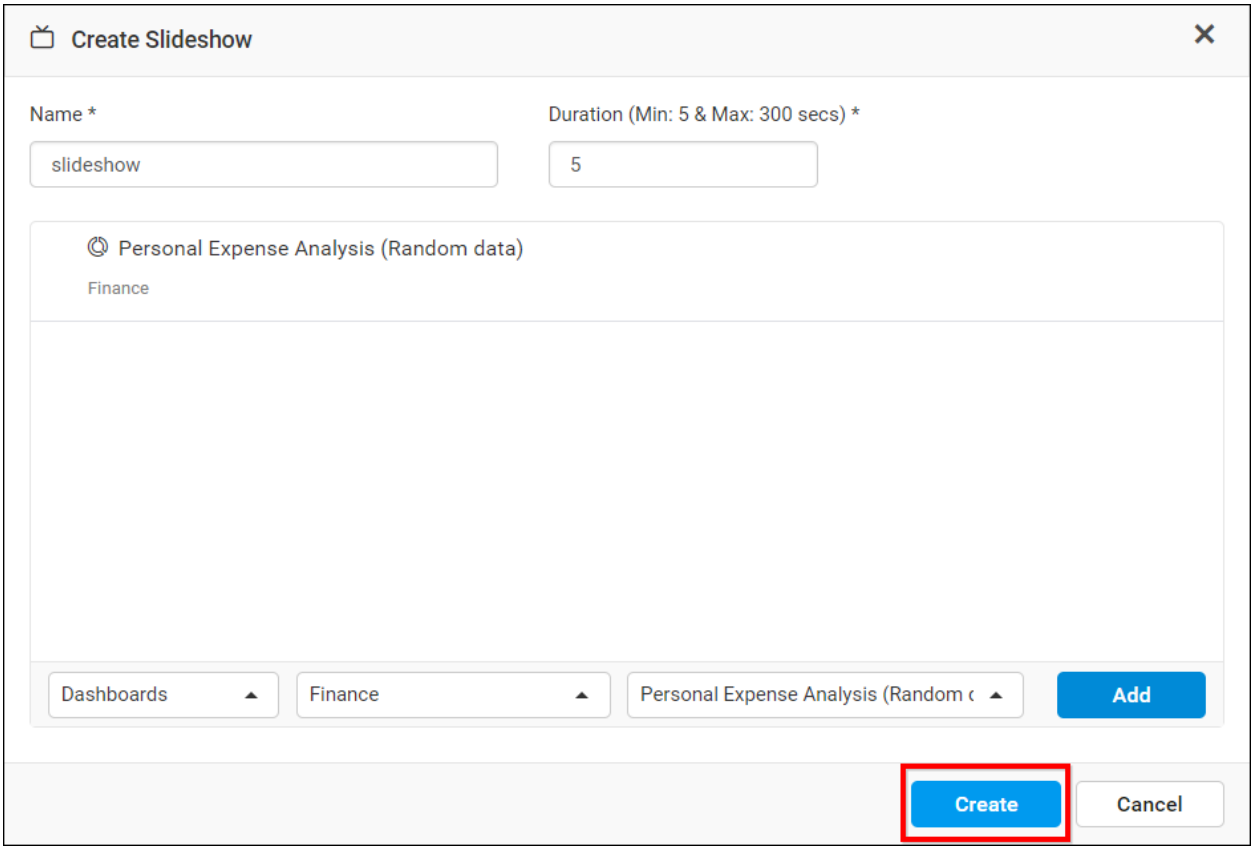
5. Click on the Add button.



6. Multiple dashboards or widgets can be added for creating slideshow. It will display the added dashboards/widgets for slideshow as shown below,



7. After adding dashboards/widgets for slideshow. Click on the Create button, it will add the slideshow into Syncfusion Dashboard Server.

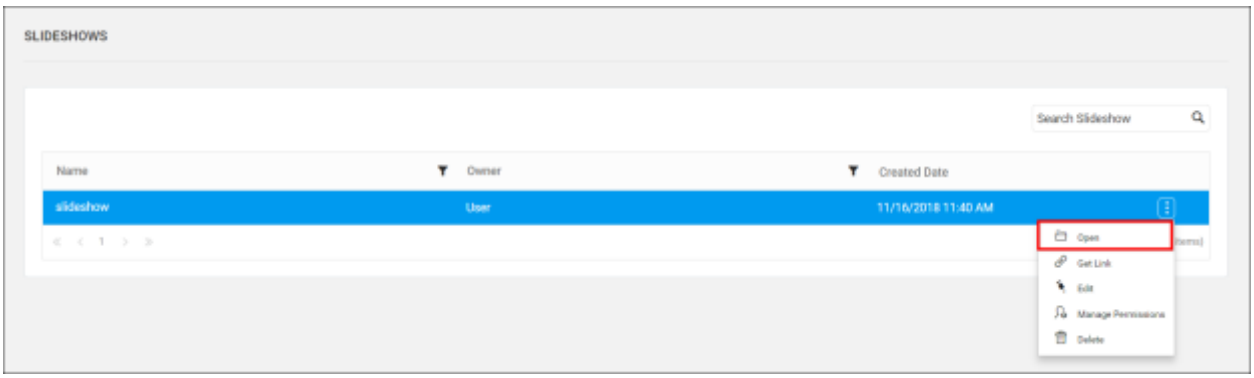


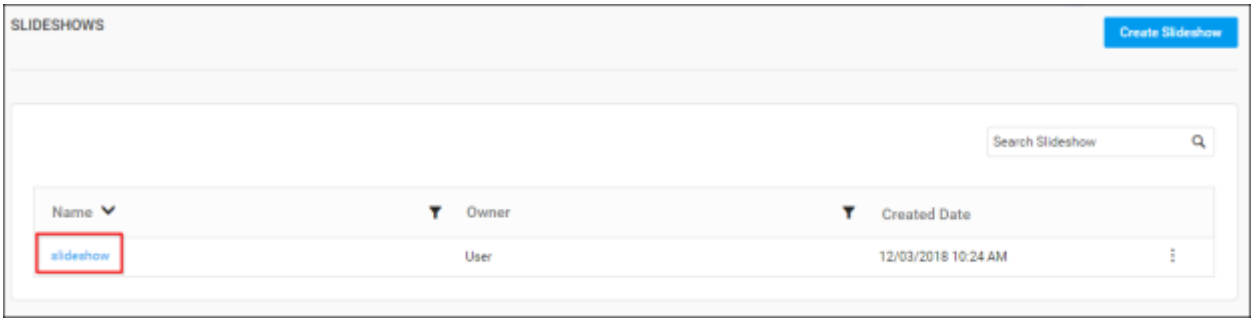
Note: Read Write Delete permission for that Specific Slideshow will be added for the user who created the slideshow.

View Slideshows

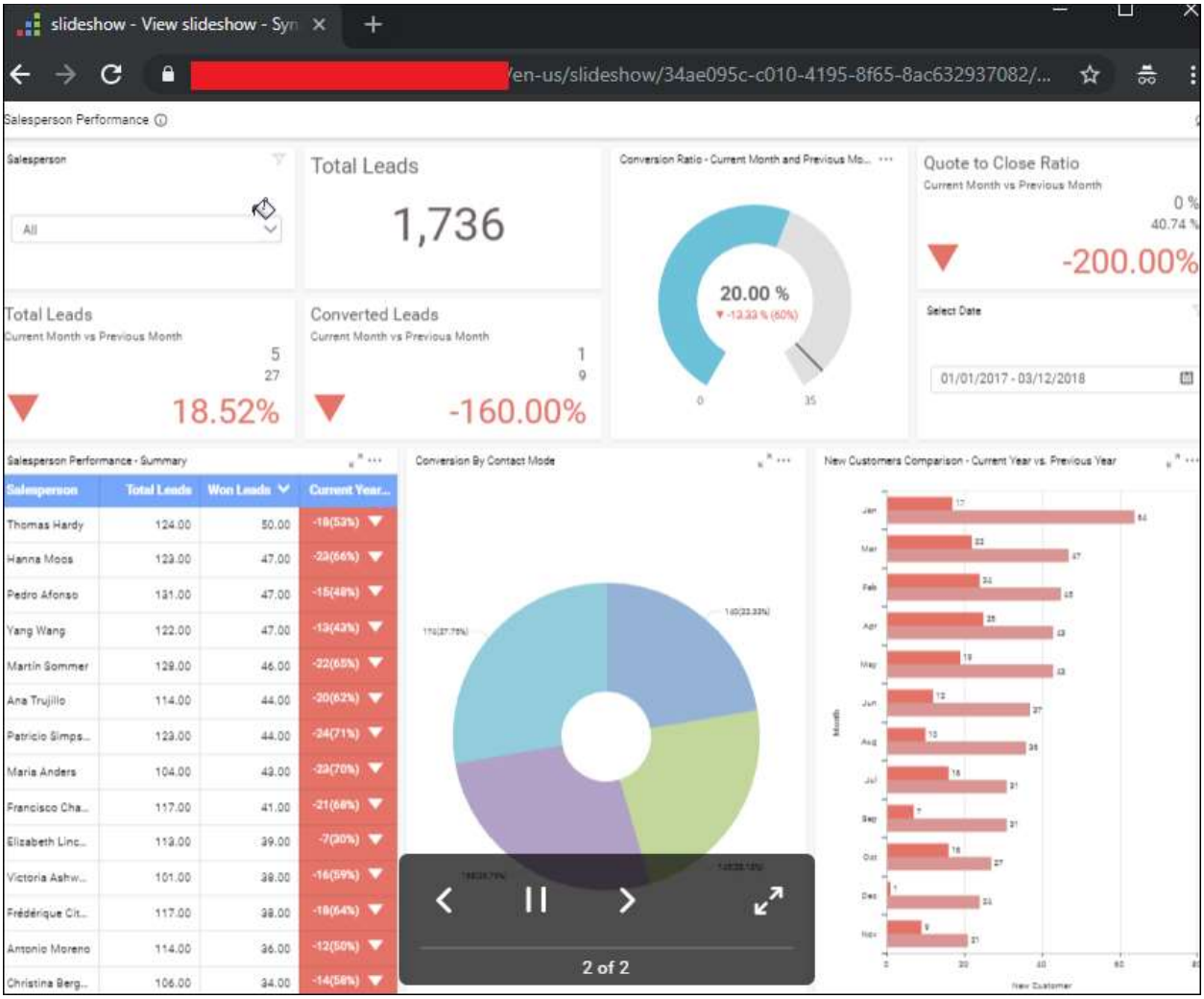
This section explains on how to view slideshows in the Syncfusion Dashboard Server.

Created slideshows can be viewed by clicking on either the Actions button in the slideshow grid context menu and select Open or the name from slideshow listing page.





Once it is opened, selected dashboards/widgets are viewed as presentation, changing one another in given time interval like below,



Options to view the previous dashboard/widget in presentation, to pause the presentation, to view the next dashboard/widget in presentation and to view the presentation in full screen are provided.

Share Slideshows

Slideshows must be shared for users or groups, so that it can be accessed by them.

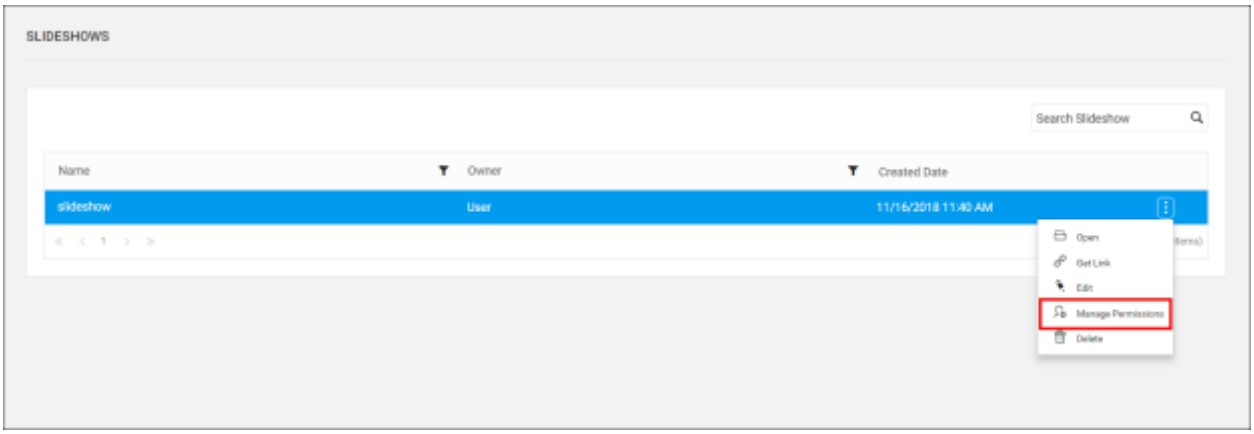
If the slideshow contains public dashboards, anyone with this link will be able to view its contents, only if the slideshows are shared to them.

If the slideshow contains private dashboards, only users who have access to that slideshow can navigate to it using the link, but users with appropriate permissions alone will be able to view the items in slideshow.

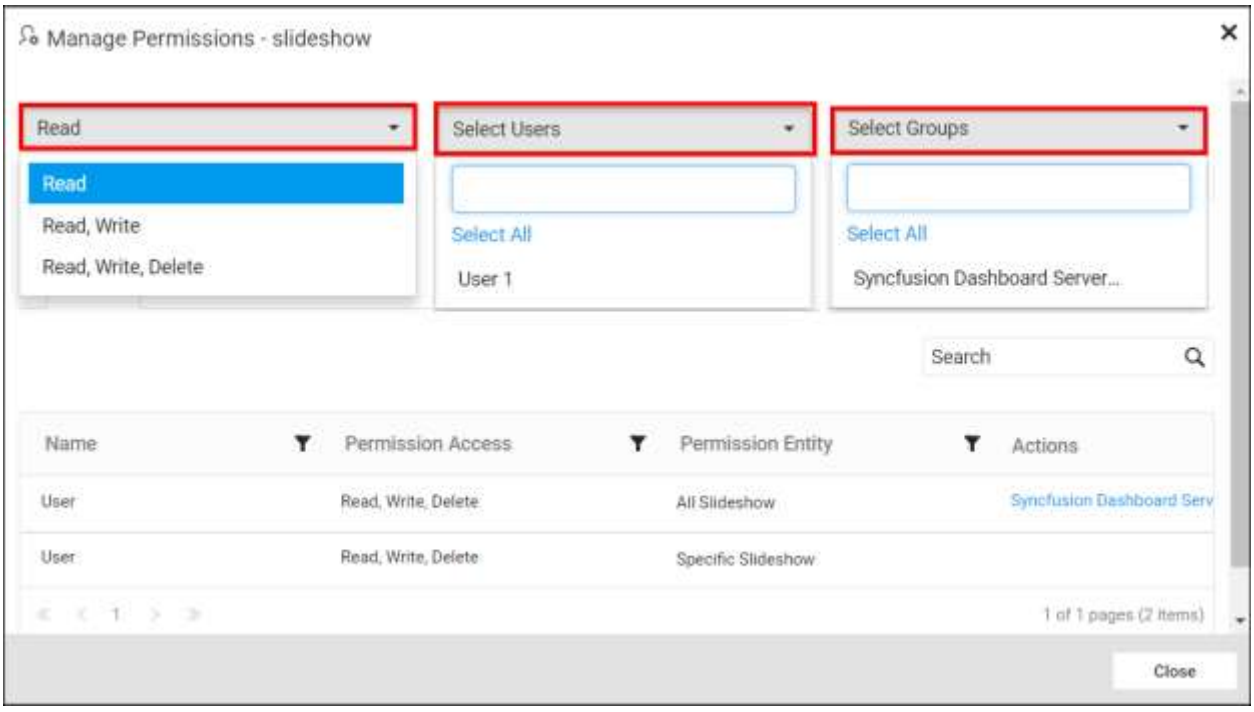
This section explains on how to share slideshows with the other users in the Dashboard Server.

Steps to share a Slideshow

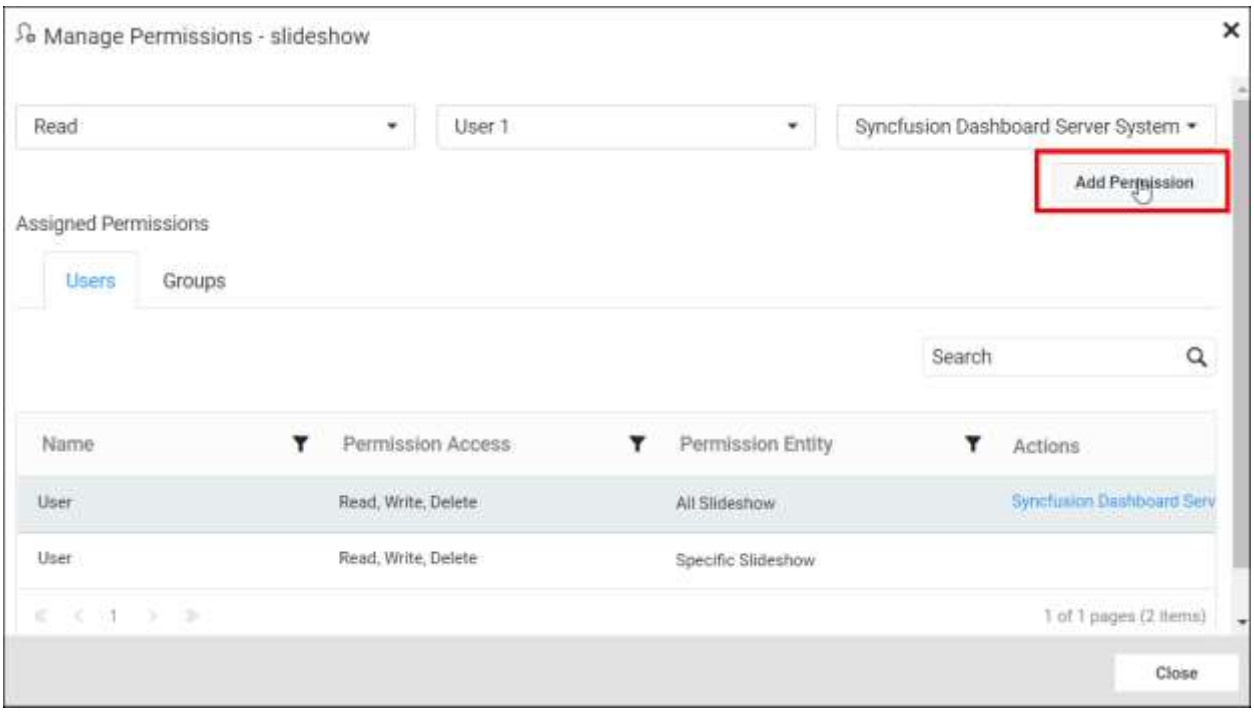
1. Click the Actions button in the slideshows grid context menu and select Manage Permissions option.



2. Select the permission access from the Select Access dropdown and select the users or groups to share the slideshow.



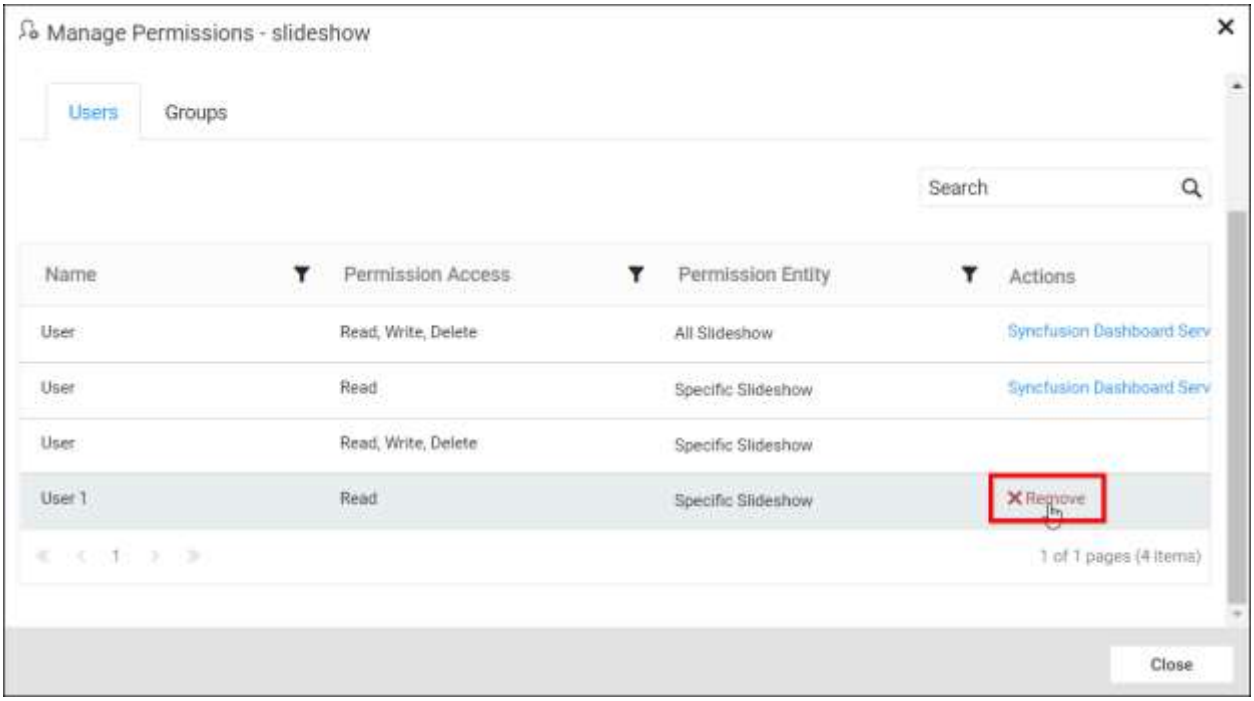
3. After selecting the access and users or groups, click on the Add Permission button.



Note: Only the user who created the slideshow and the administrator can share the slideshow with other Dashboard Server users.

Remove Permission

The user who created the slideshow and the administrator can remove the shared slideshow permissions using the Remove option in the Actions column of the each permissions.



Link to Slideshows

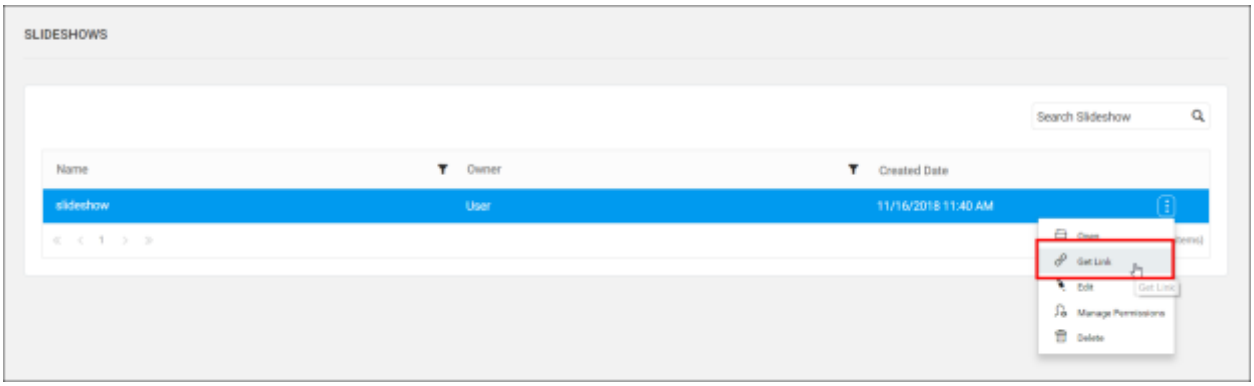
This section explains on how to get link to the slideshows in the Syncfusion Dashboard Server.

These links are used to navigate to the slideshow and can be shared with others.

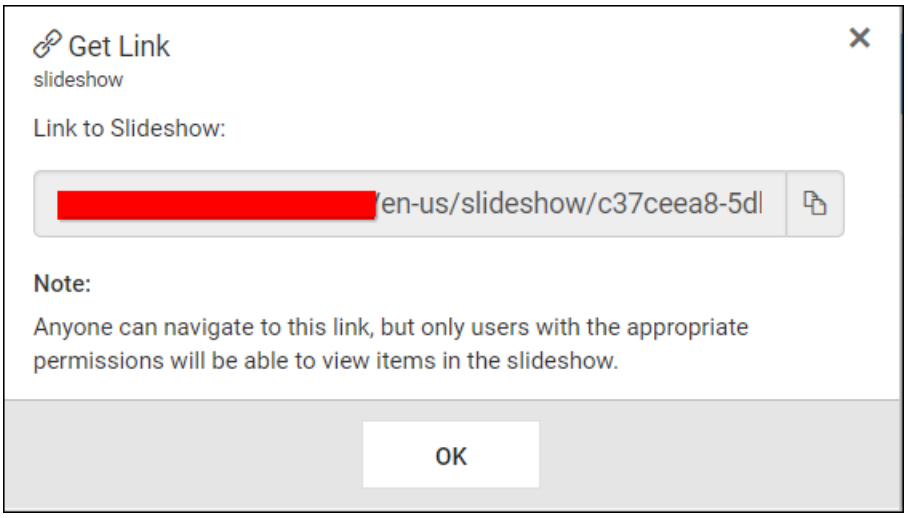
Get Link option is available for all the slideshows in the grid.

Follow the below steps to get the link of a slideshow.

1. Click on the context menu of the respective slideshow and choose Get Link option.



2. The Get Link dialog will be opened. Click on Click to Copy to get the get link.



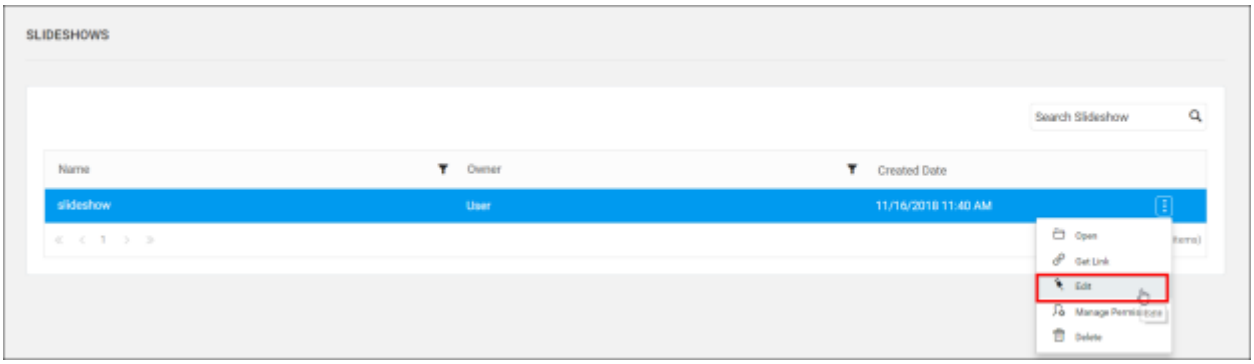
Update Slideshows

This section explains on how to modify existing slideshows in the Syncfusion Dashboard Server.

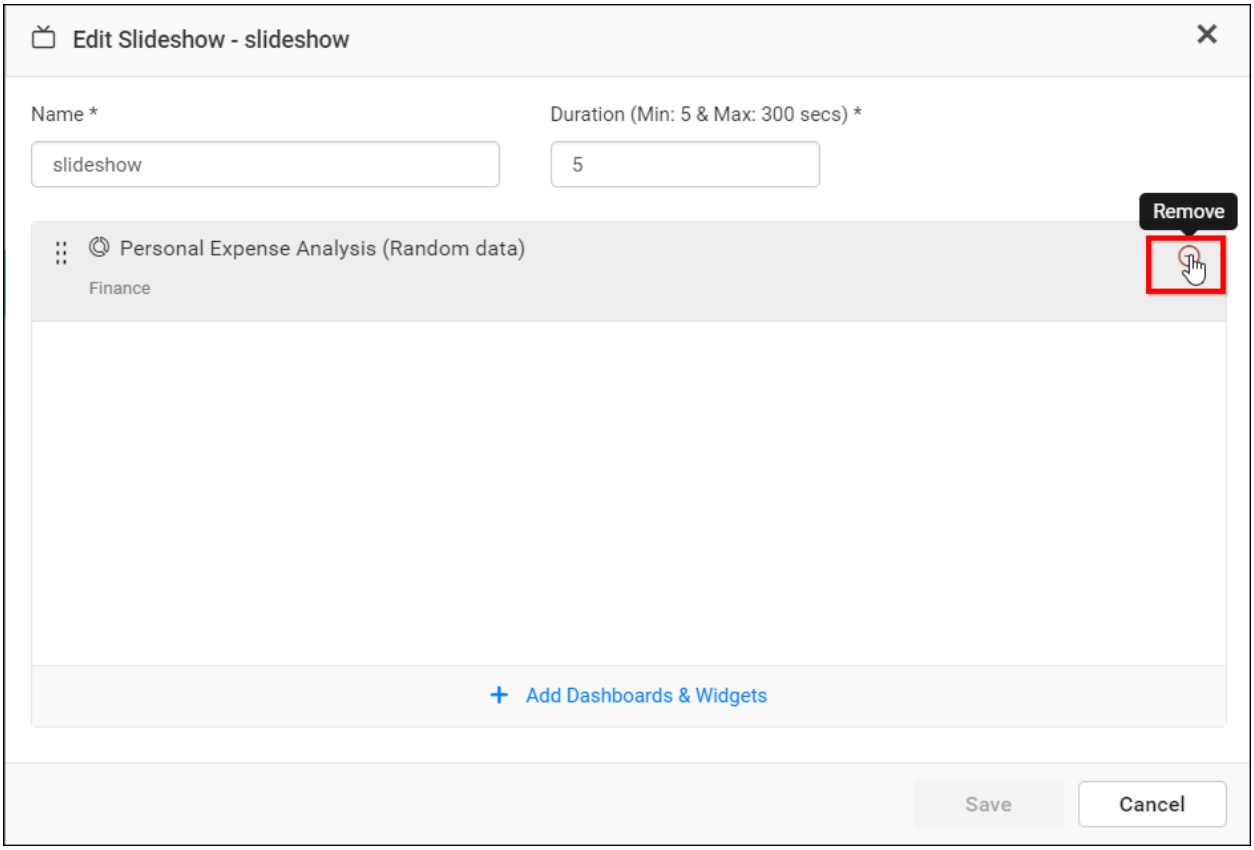
Steps to update a Slideshow

Name, Duration and the Dashboards/Widgets listed for slideshow can be changed in the Edit Slideshow dialog box.

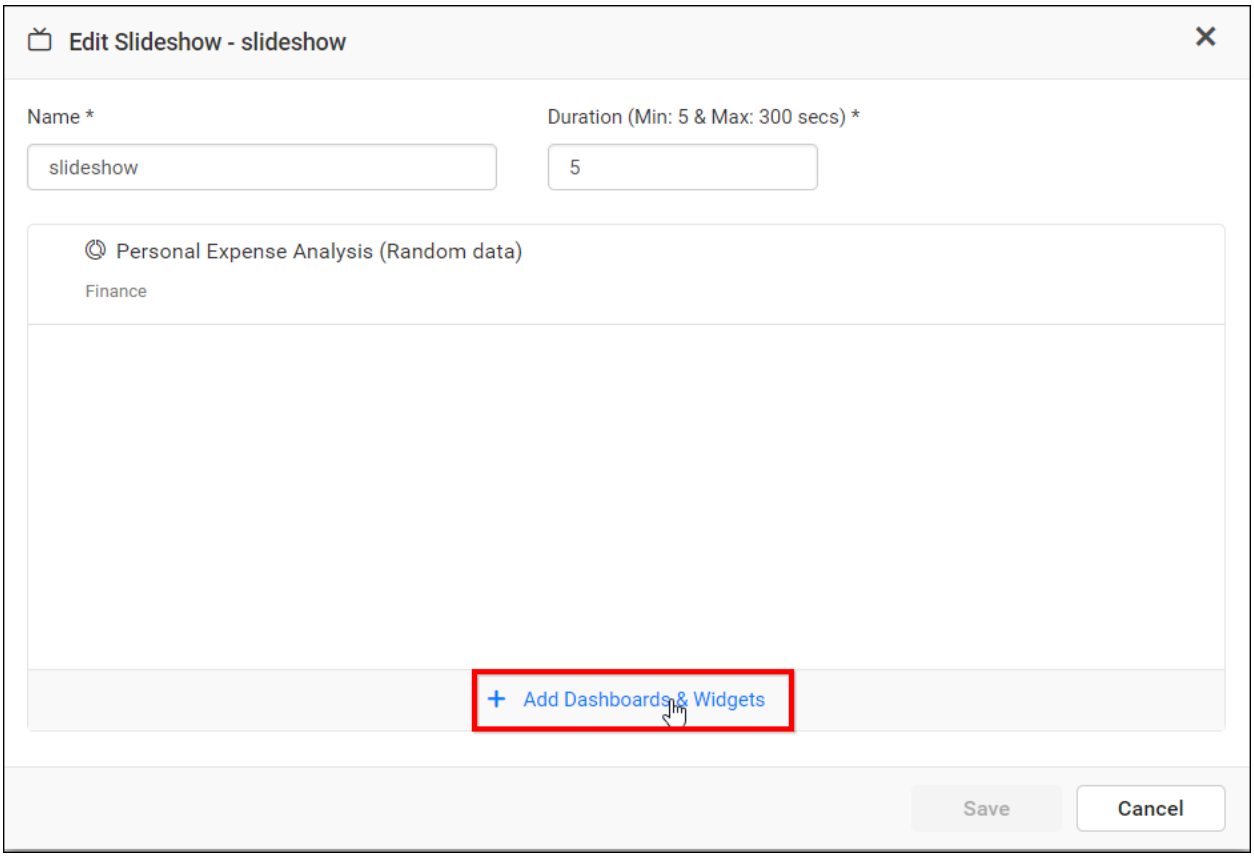
1. Click on the Edit option in the context menu of the respective slideshow.



2. Dashboards/Widgets for the slideshow can be added or removed.
3. To remove the listed Dashboards/Widgets added for the slideshow, click on the icon as shown below,

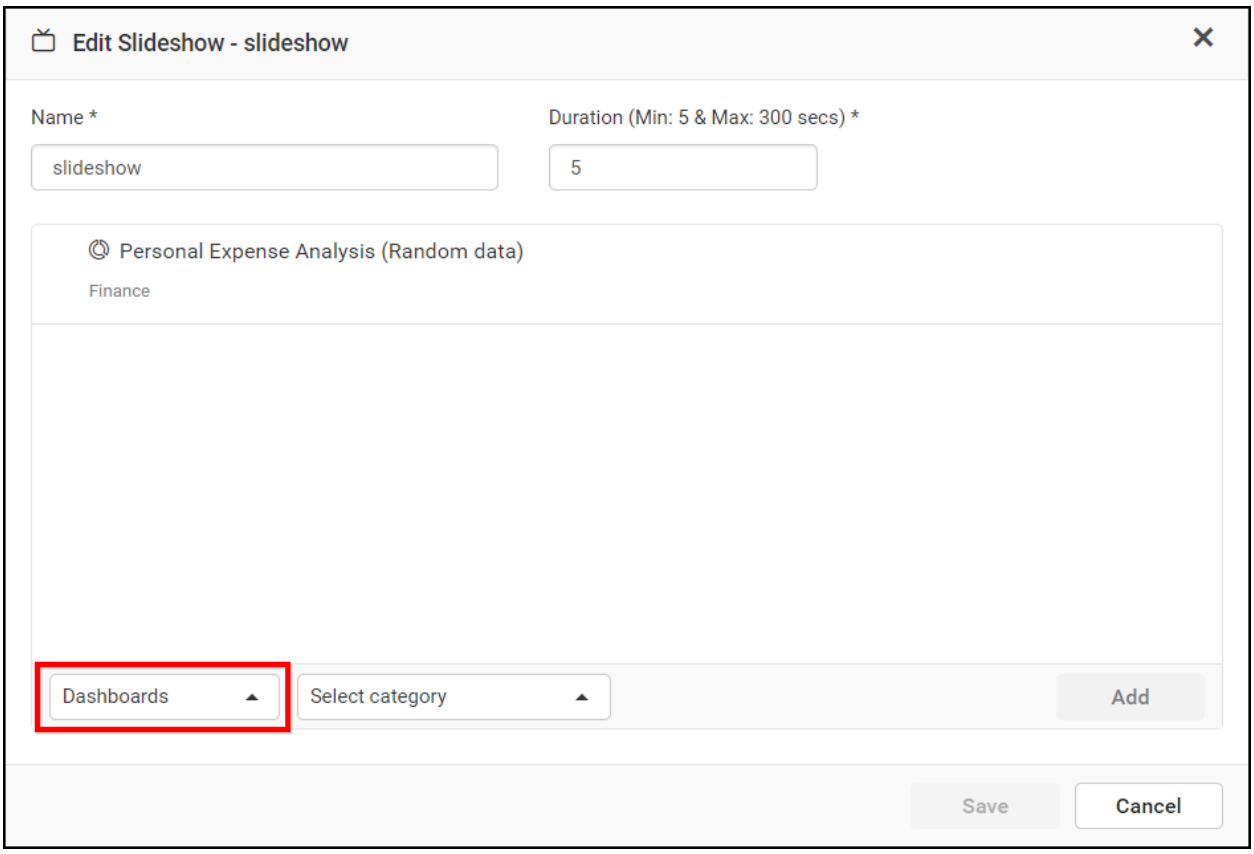


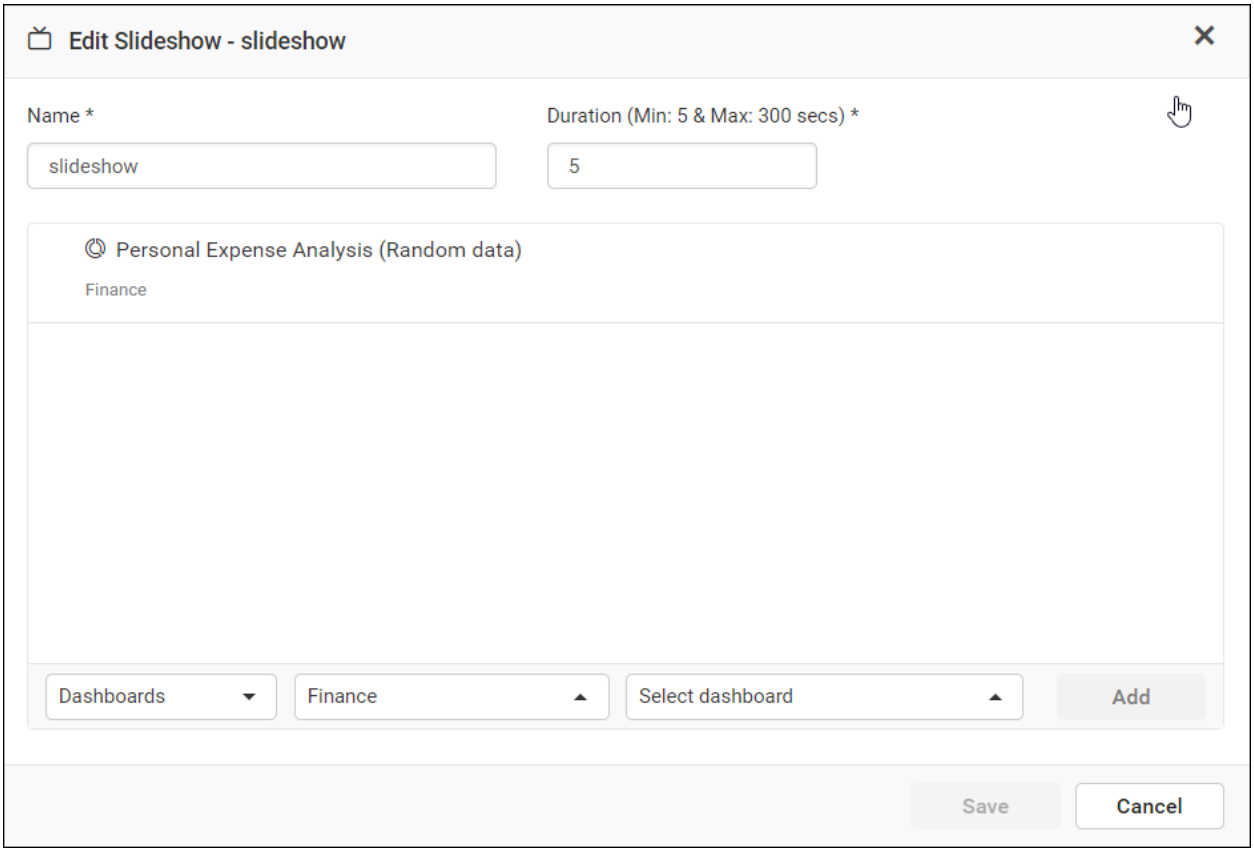
2. To add the dashboard/widget for the slideshow, click on the Add Dashboards & Widgets in the Edit Slideshow dialog box.



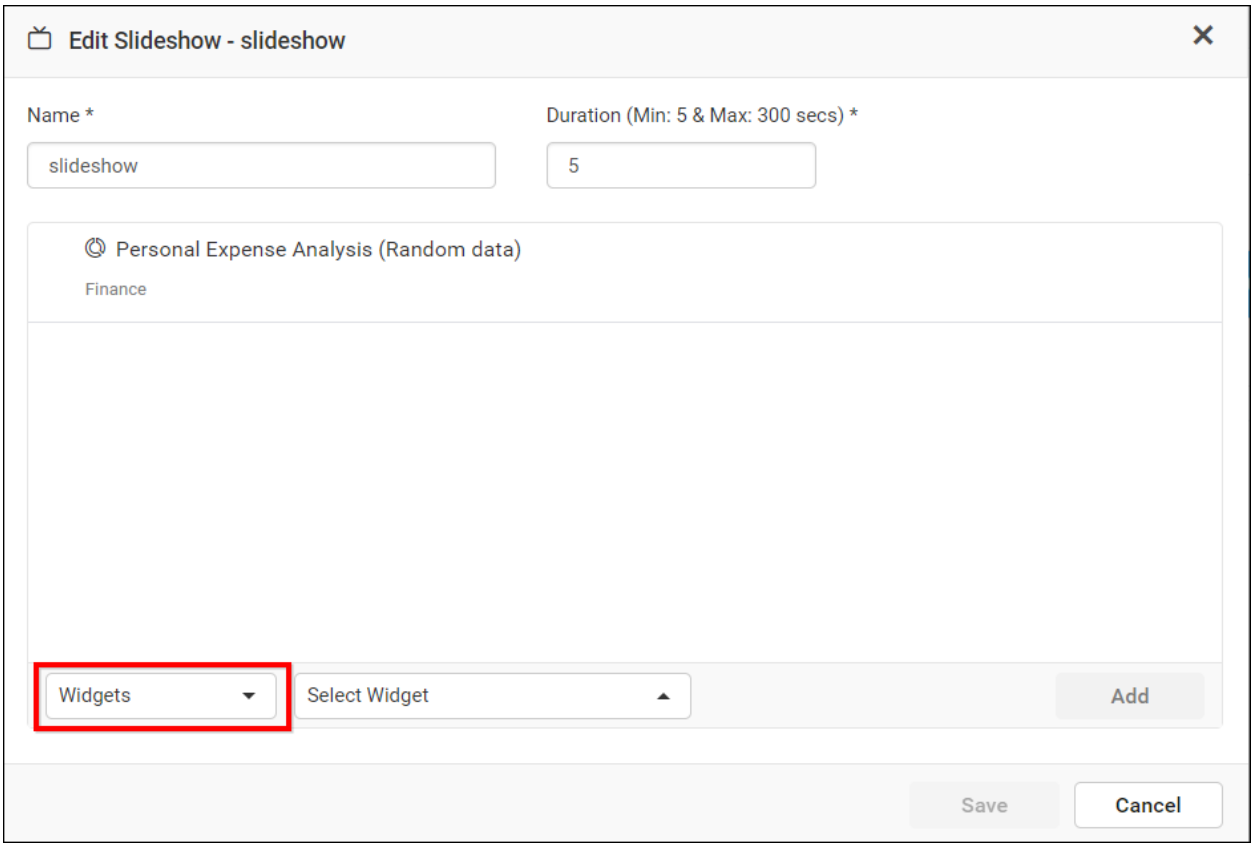
- Select the required type Dashboards or Widgets from dropdown.

Dashboards - If the Dashboards type is chosen, after that select the category from Select Category dropdown and corresponding dashboards under that selected category will be displayed in the Select Dashboards dropdown.

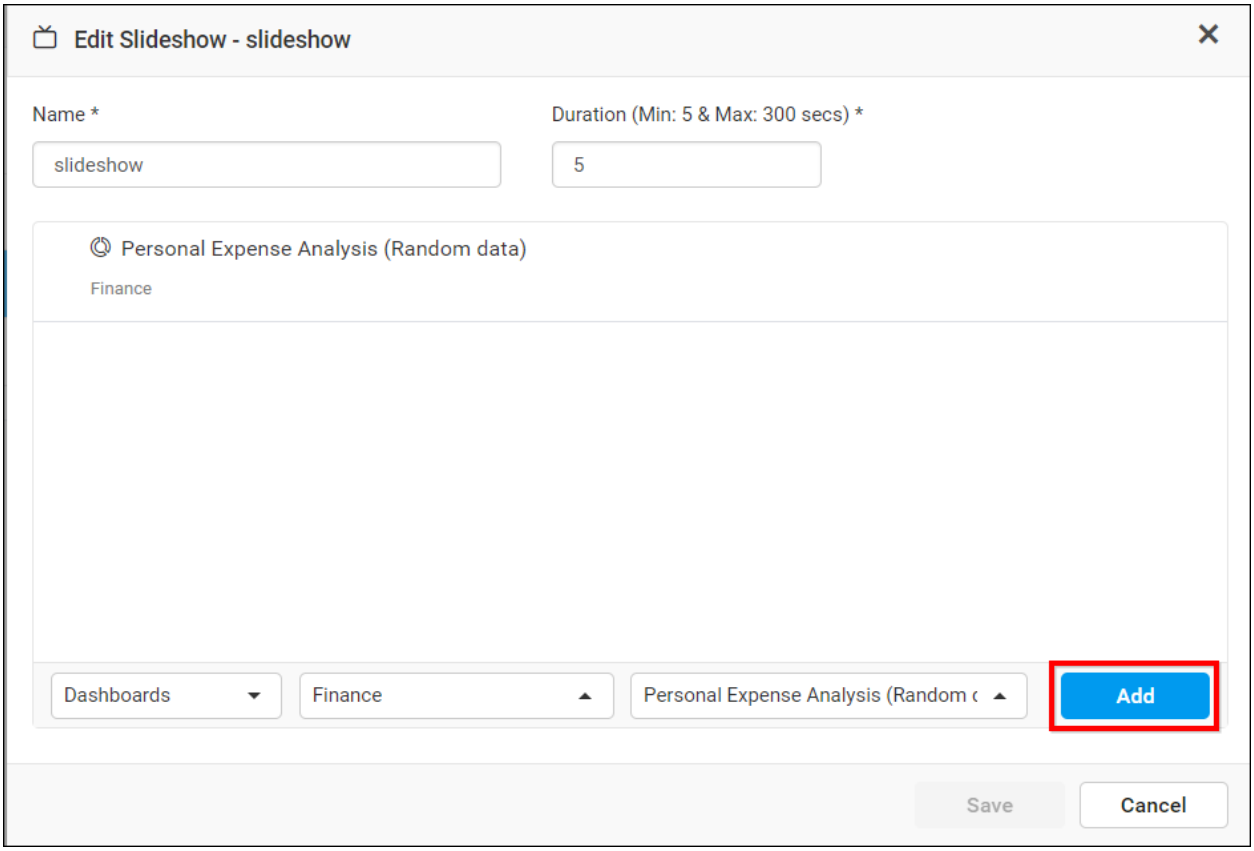




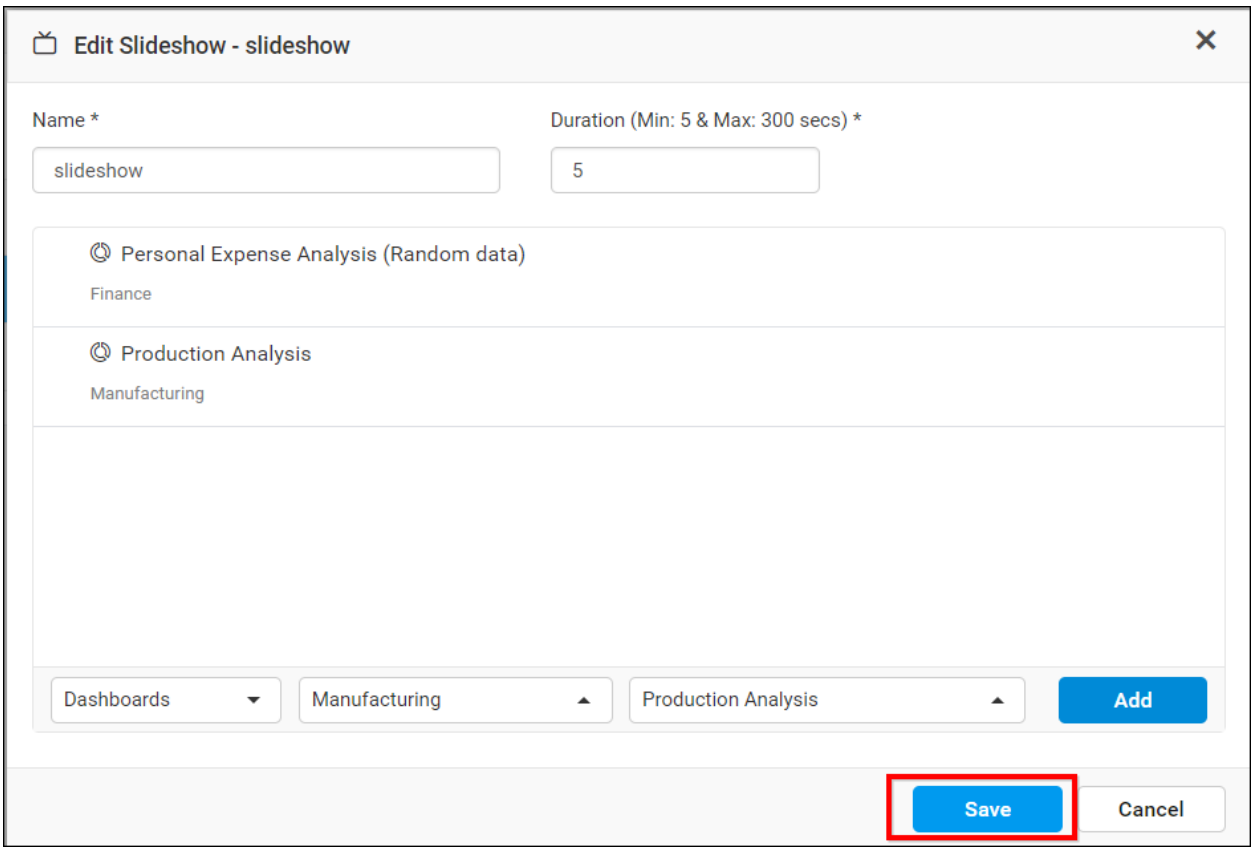
Widgets - If the Widgets type is chosen, after that select the widgets from Select Widgets dropdown.



- After selecting the dashboards, widgets from select dashboards and select widgets dropdown respectively Add button will enable and click on the Add button.



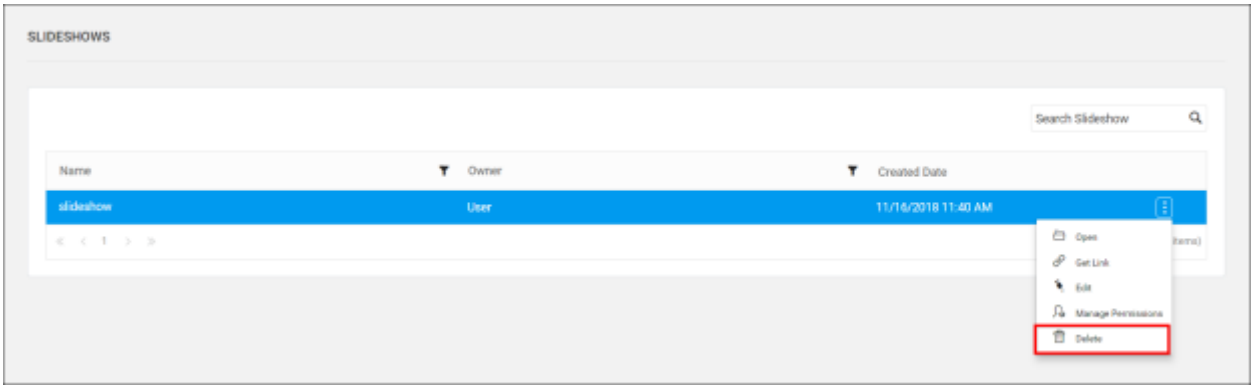
3. After making changes to the Name, Duration and Dashboards/Widgets listed for slideshow, Click on the Save button in the Update Slideshow dialog box .



Delete Slideshows

Slideshow can also be deleted from the Dashboard server when they are no longer required.

Click the Actions button in the slideshows grid context menu and select Delete to delete the slideshow.



Data Alerts

Data Alert provides the users to track certain conditions and respond it to a specified recipients via mail when a certain pre-defined threshold is met.

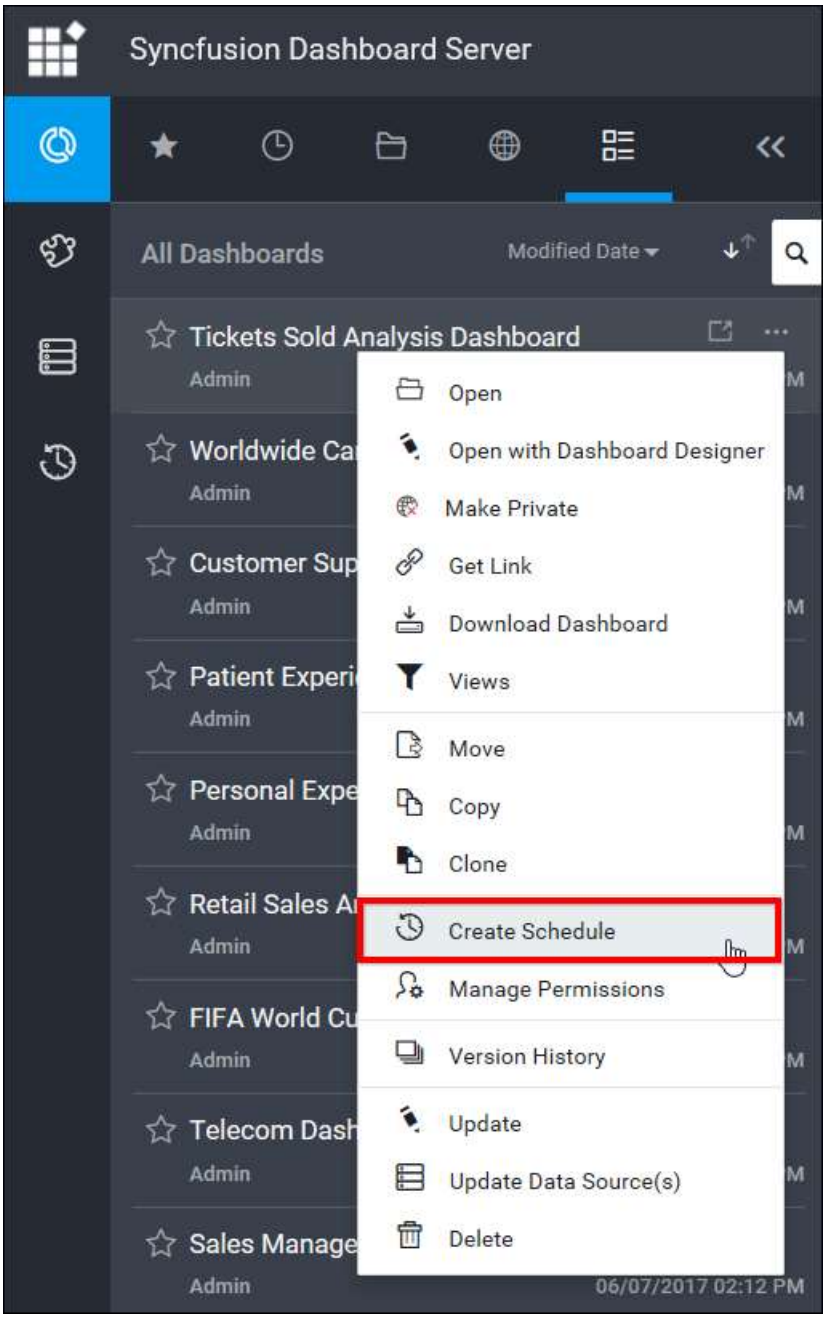
Add Data Alert

Follow the below steps to create the new Data Alert with the desired dashboard.

Steps to create a schedule

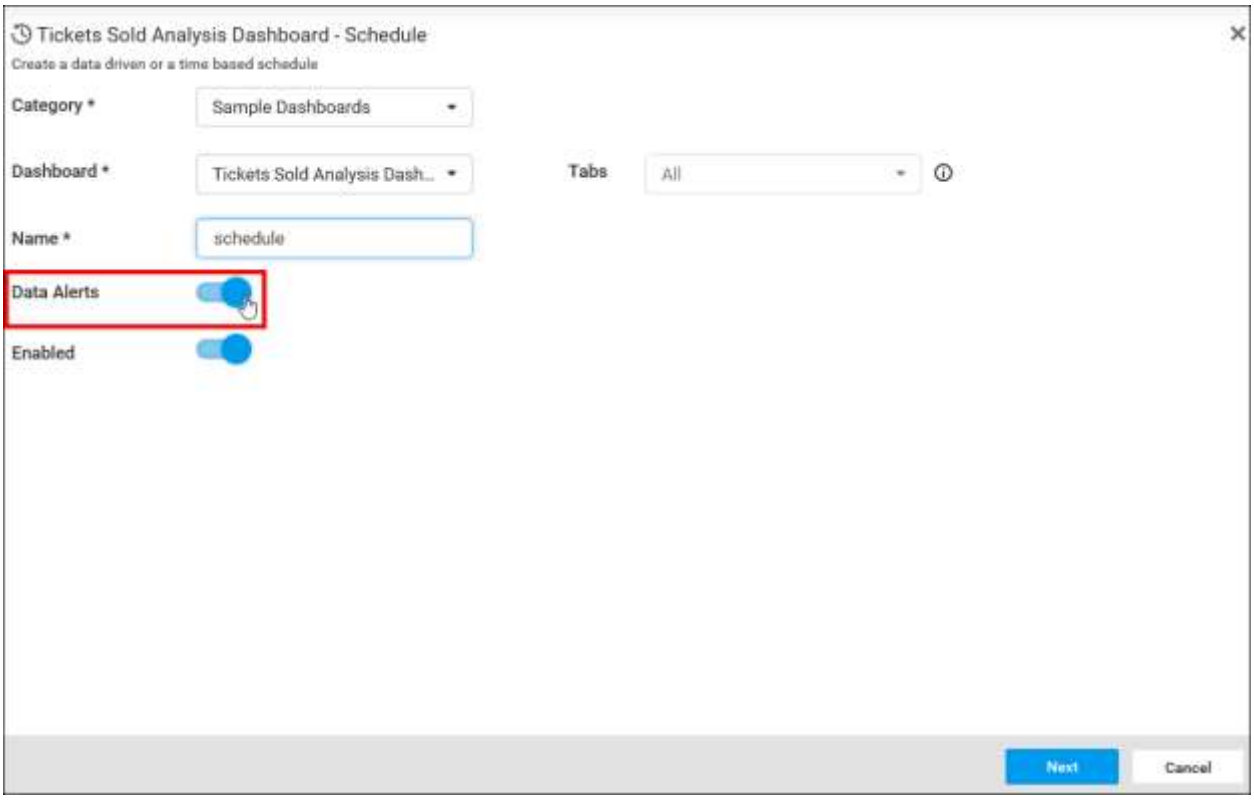
Select Dashboard

Click the Actions button in the dashboard grid context menu and select Create Schedule to schedule the corresponding dashboard.



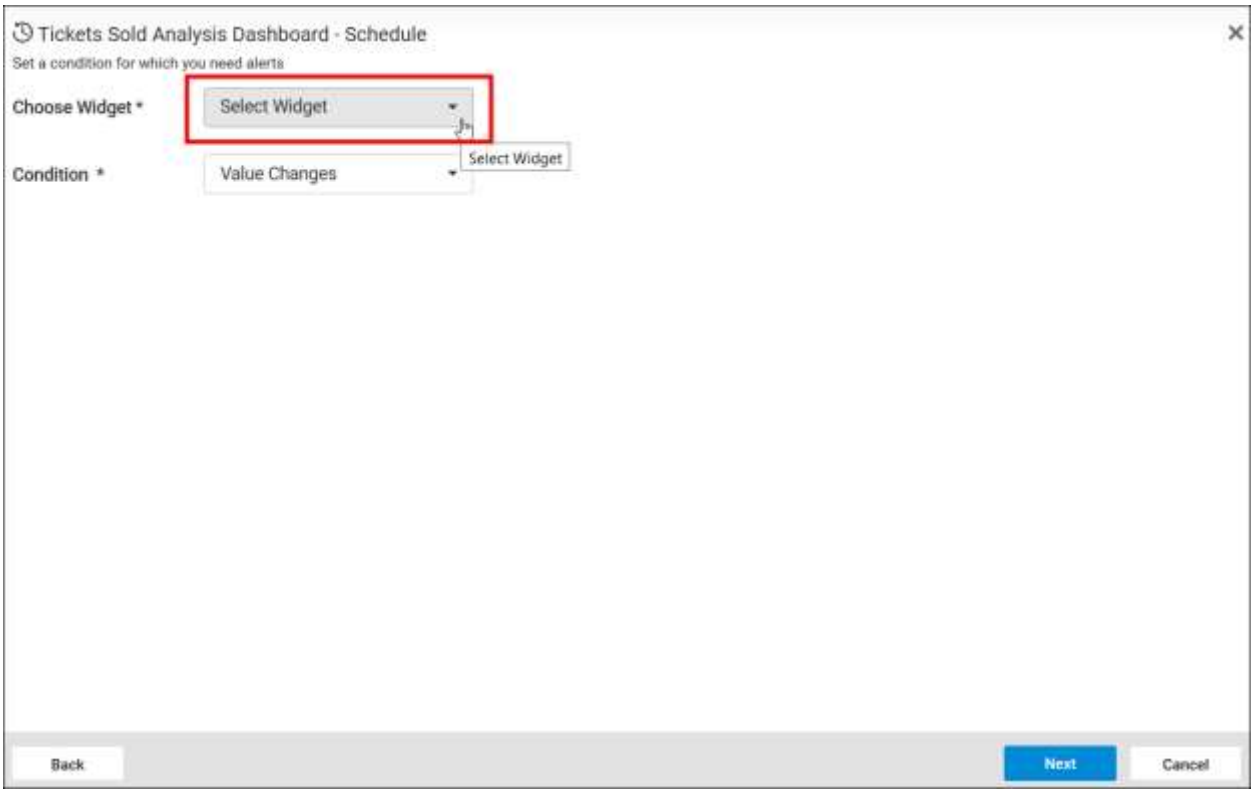
Otherwise, click on the Schedule option from the menu to create a schedule,

Schedules can be created for both single dashboard as well as multi-tabbed dashboard. For multi-tabbed dashboard, Tabs dropdown will be enabled and the dashboards inside the multi-tabbed dashboards are listed in it. Enter the schedule Name and then Data Alerts needs to be enabled.



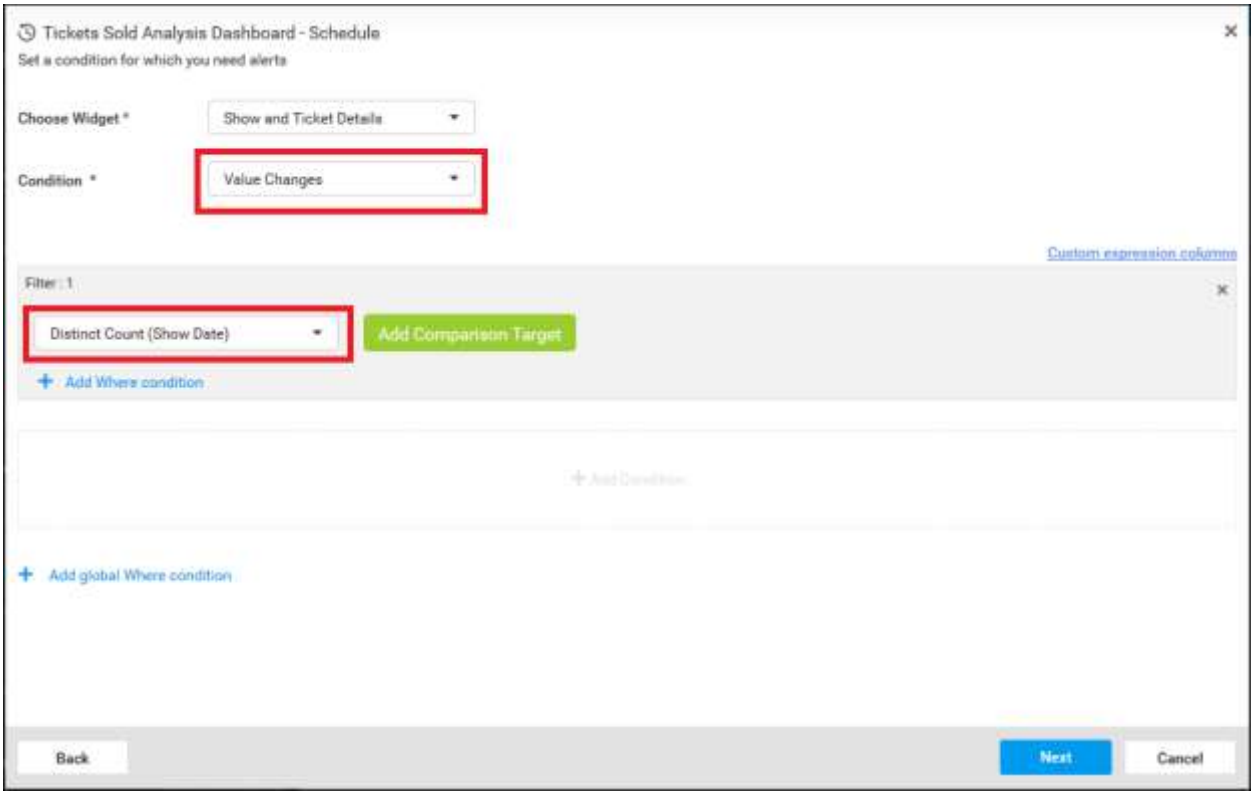
Data Alert

Data Alert screen has the option to add conditions for the desired dashboards. Each Condition has the Source ColumnName, Aggregation, and Target column to compare. A Where Condition can be added as globally or for each condition. After the successful validation, the user can be navigating to the next screen, otherwise, the error message displayed at the top of the dialog and the screen remains exists until the successful validation.



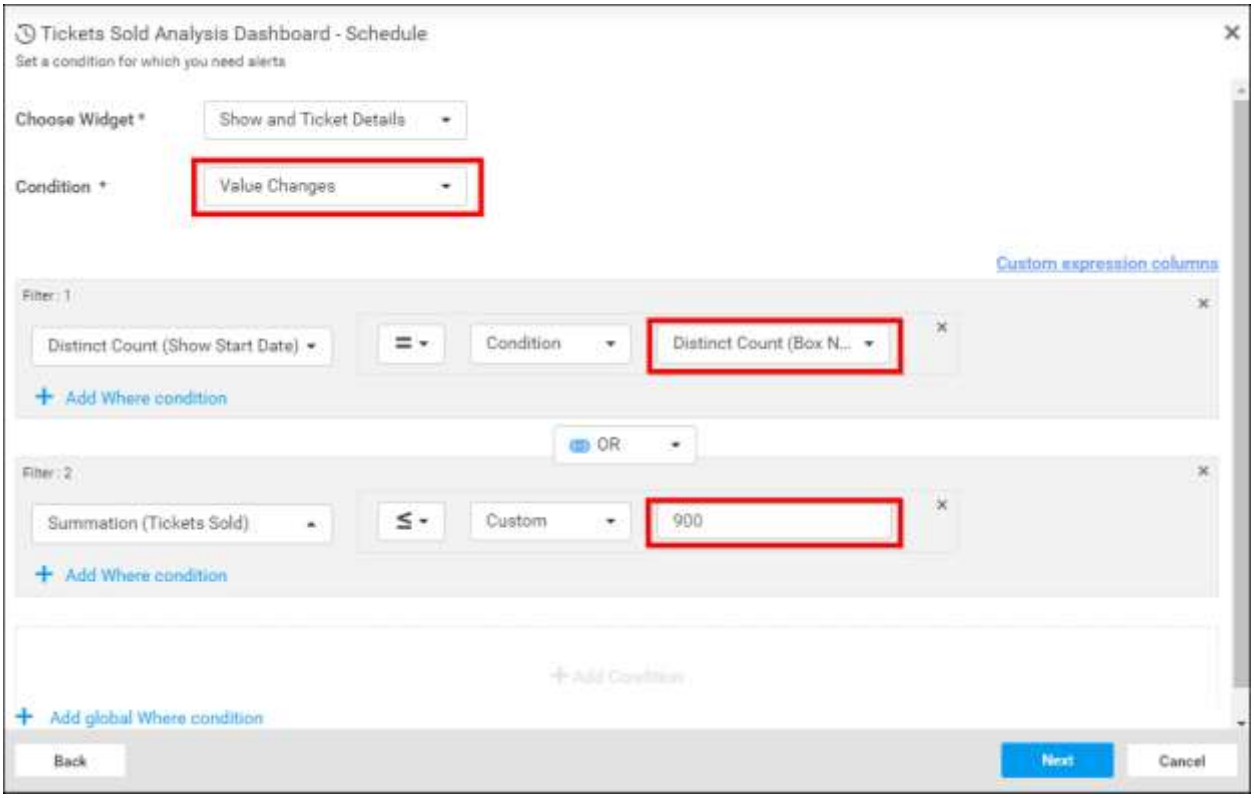
- Select the Widget which you want to add threshold condition and select any one of the Condition. By Default, the Condition will be Value Changes.
    1. Value Changes - Checks whether the filtered value changes.

Value Changes can be evaluated either Expression or a single column.

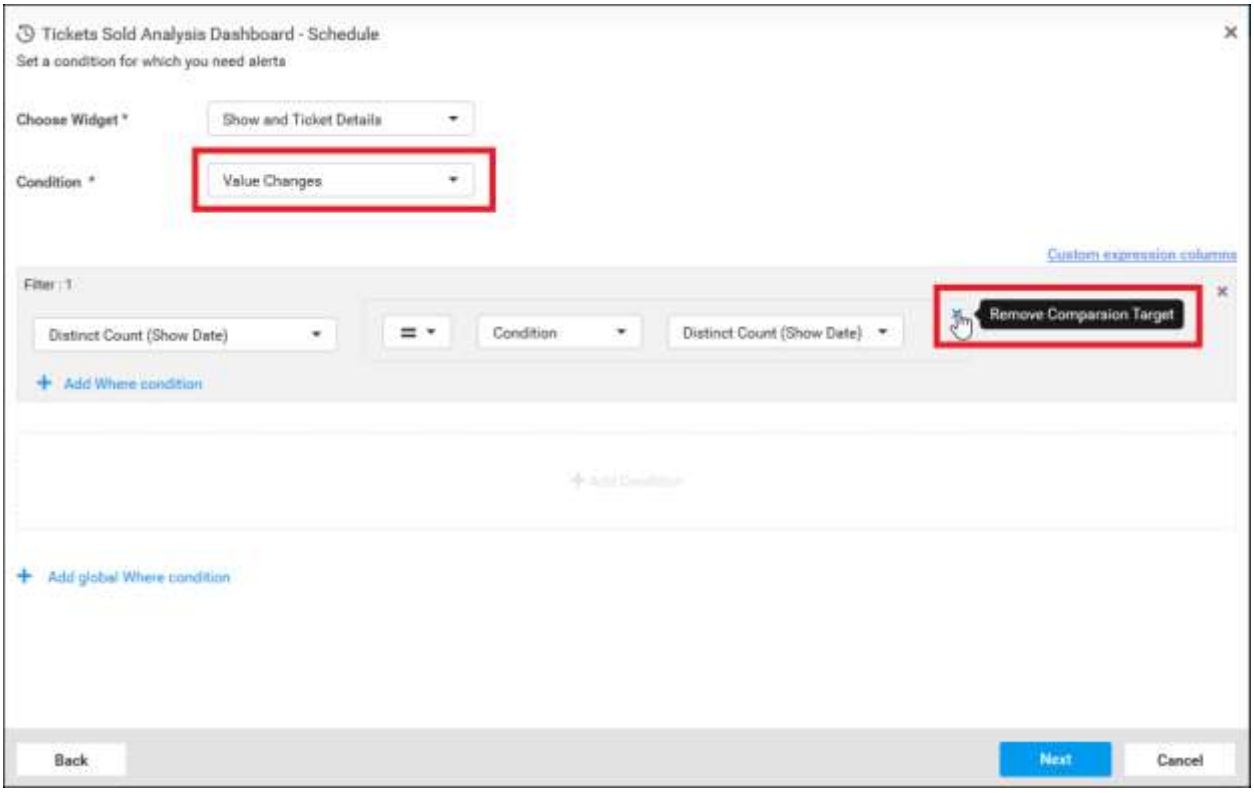


If it is an Expression, the user can be notified when it satisfies the condition.

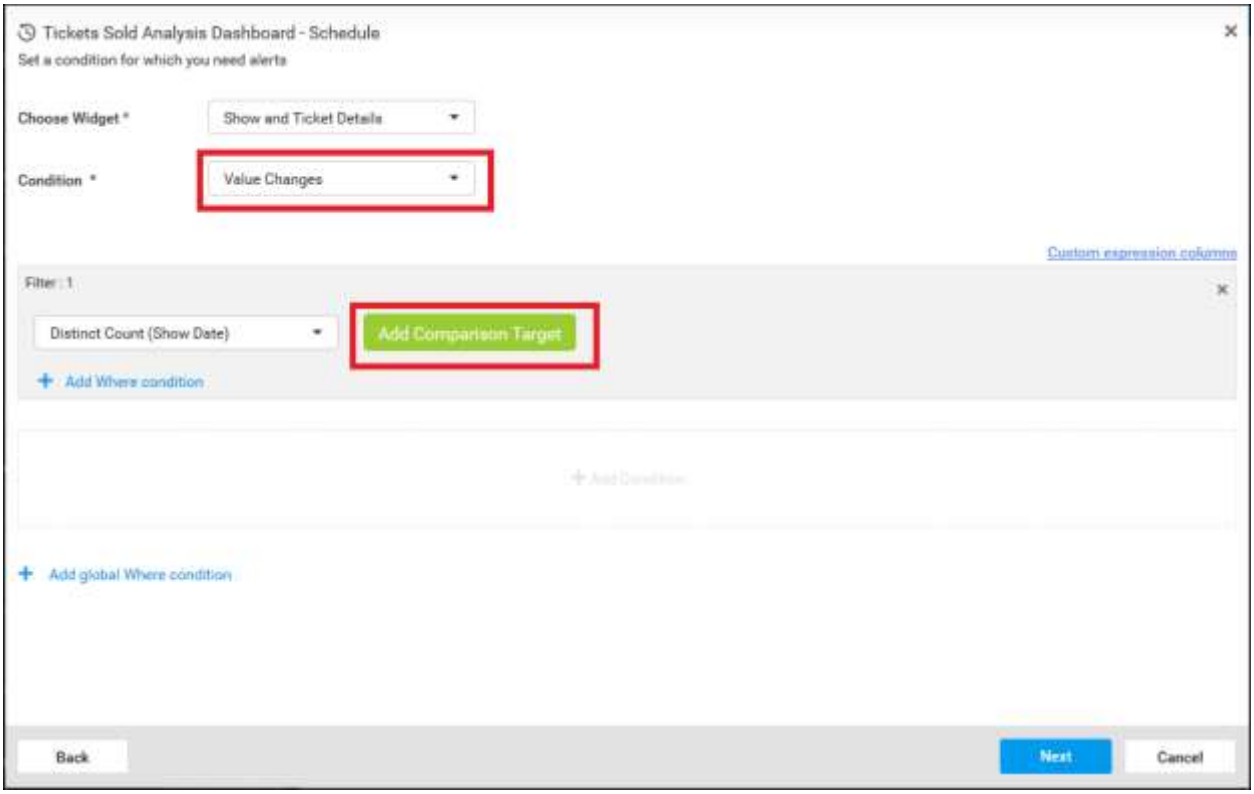
1. The user can add the expression as per their need as below,



2. To switch the expression to a single column click on the close button as below,

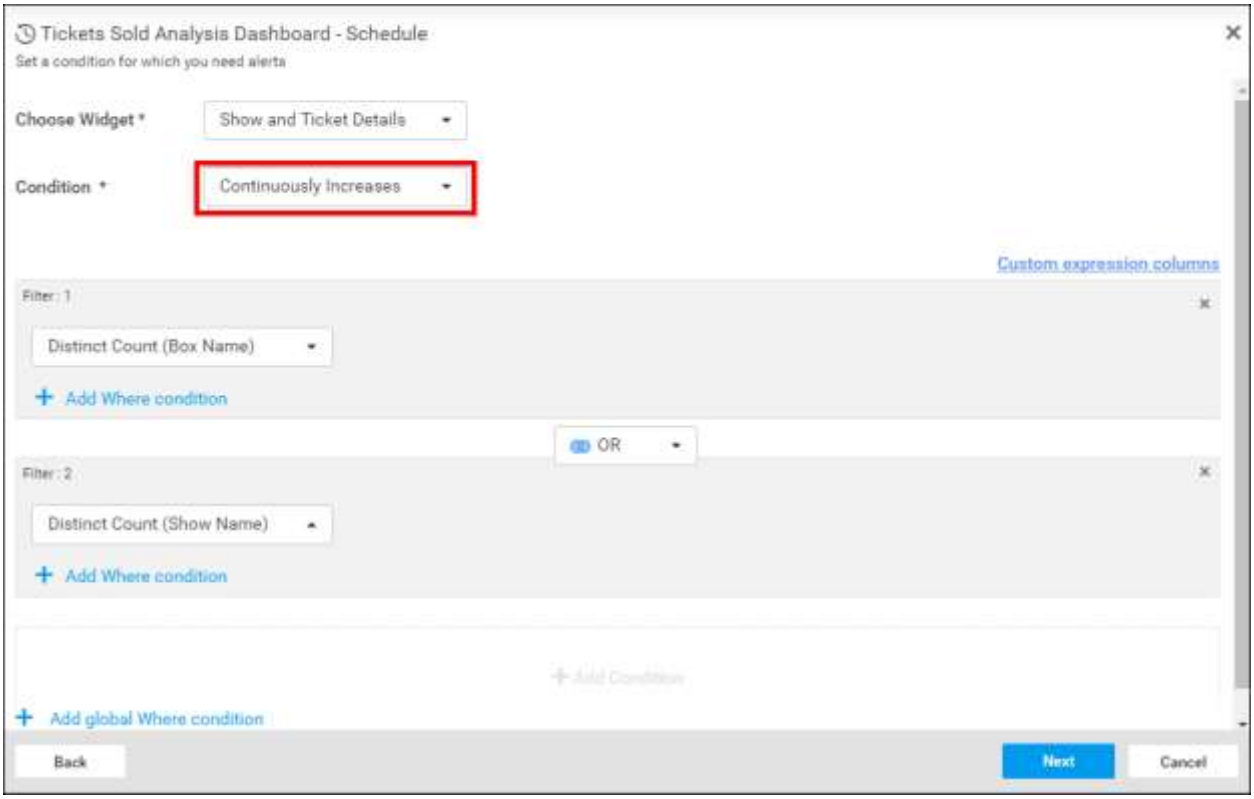


3. Click on the button Add Comparison Target to add expression,

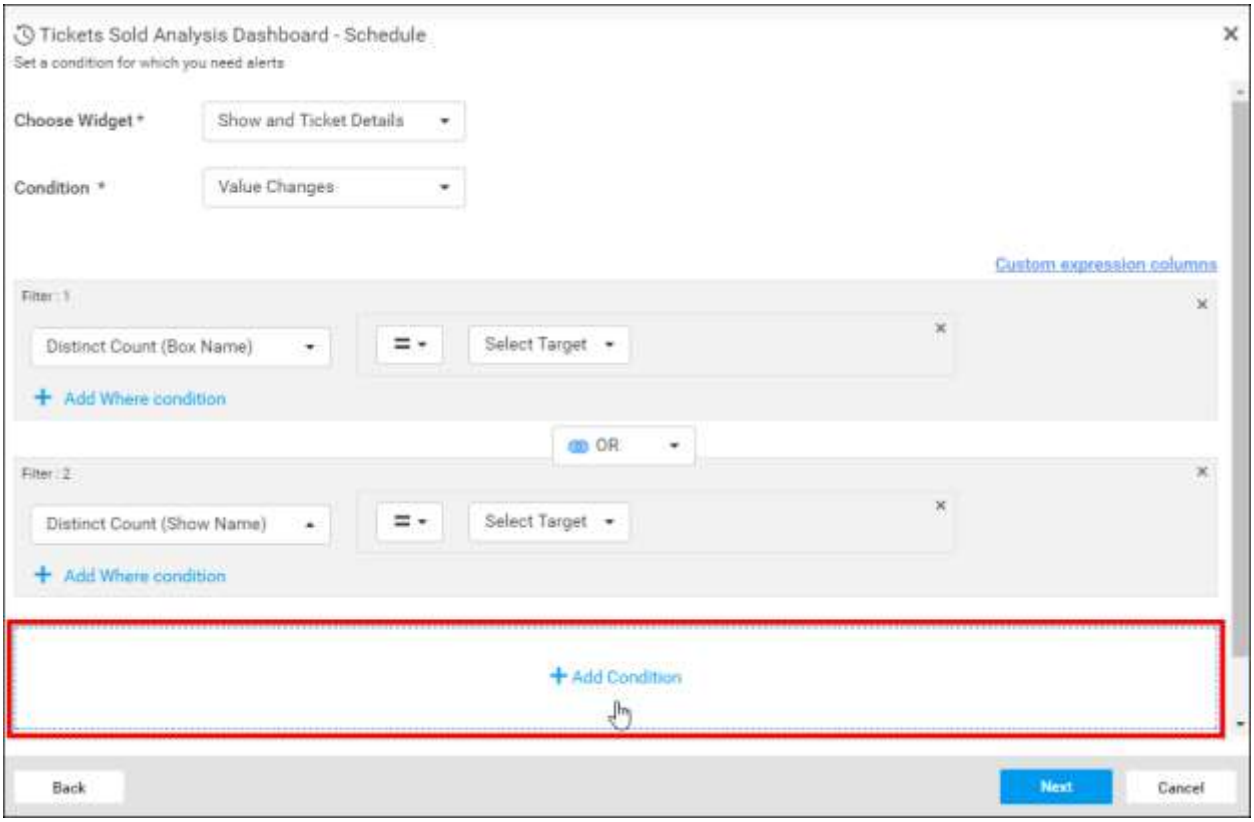


If it is a single column, the user can be notified when there is a change in the column value compared to the previous value.

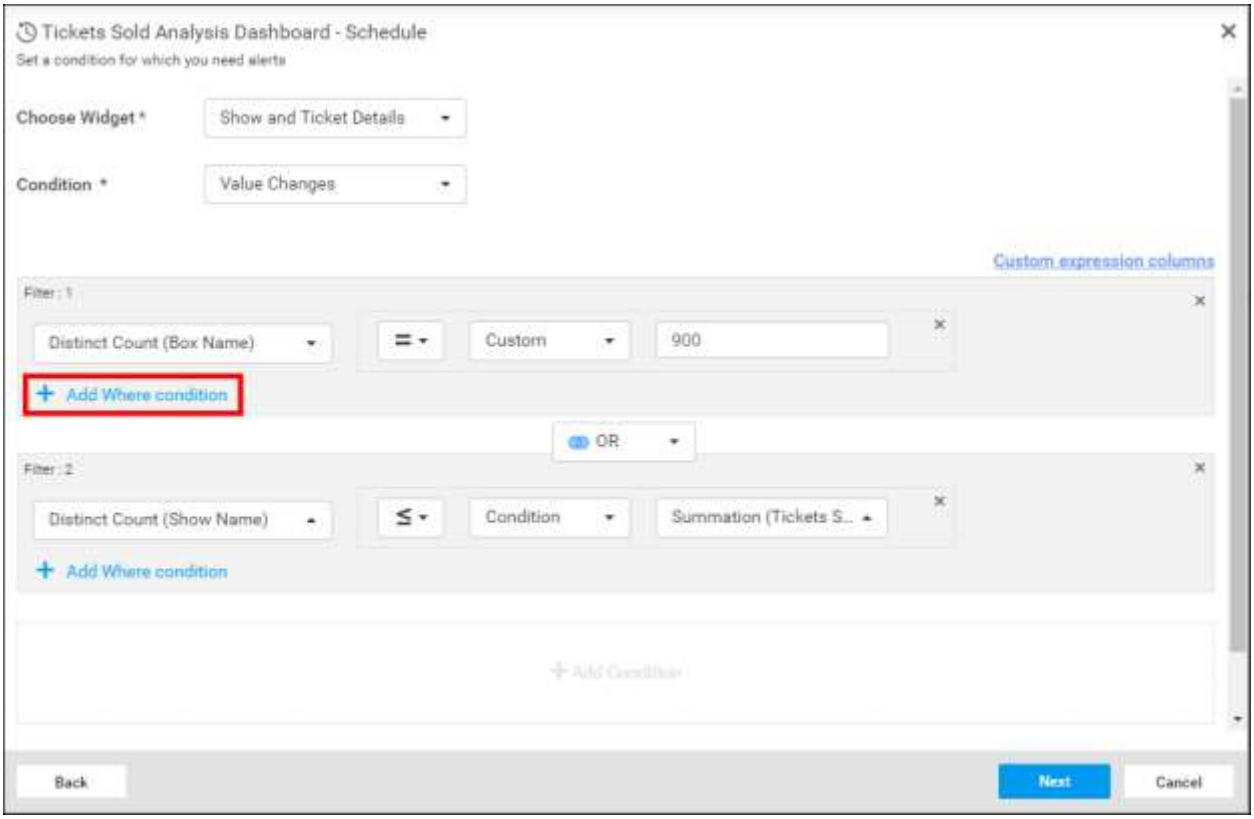
2. Increases - Checks whether the filtered value increases once.
3. Continuously Increases - Checks whether the filtered value increases continuously.
4. Decreases - Checks whether the filtered value decreases once.
5. Continuously Decreases - Checks whether the filtered value decreases continuously.



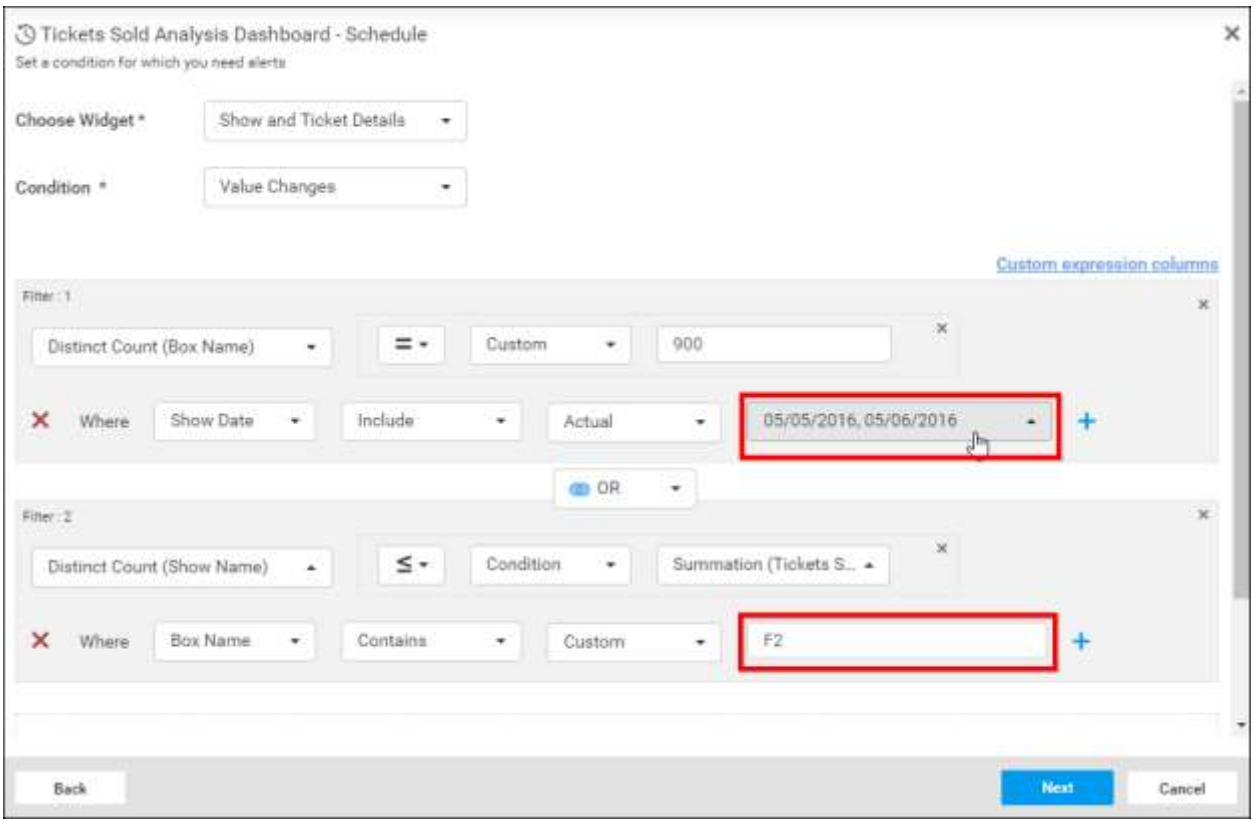
- Add more conditions by click on Add Condition button.

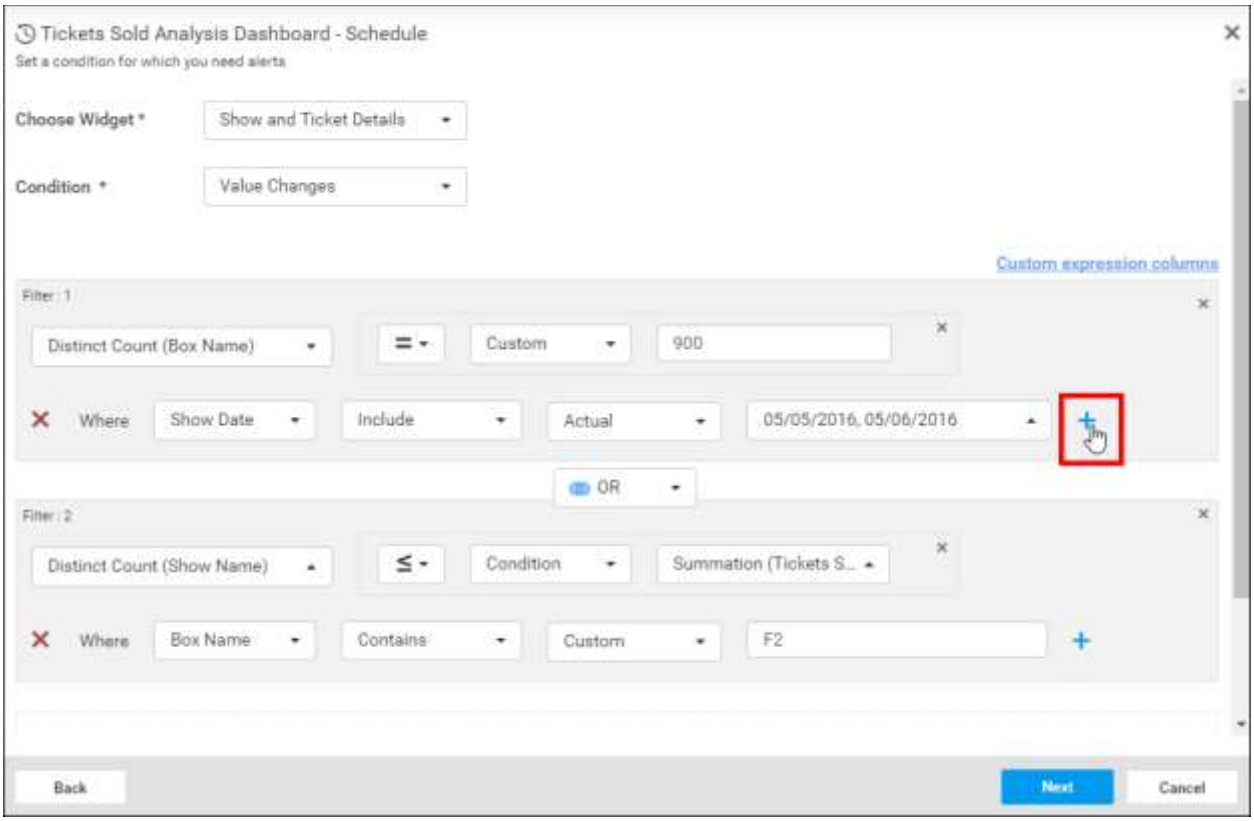


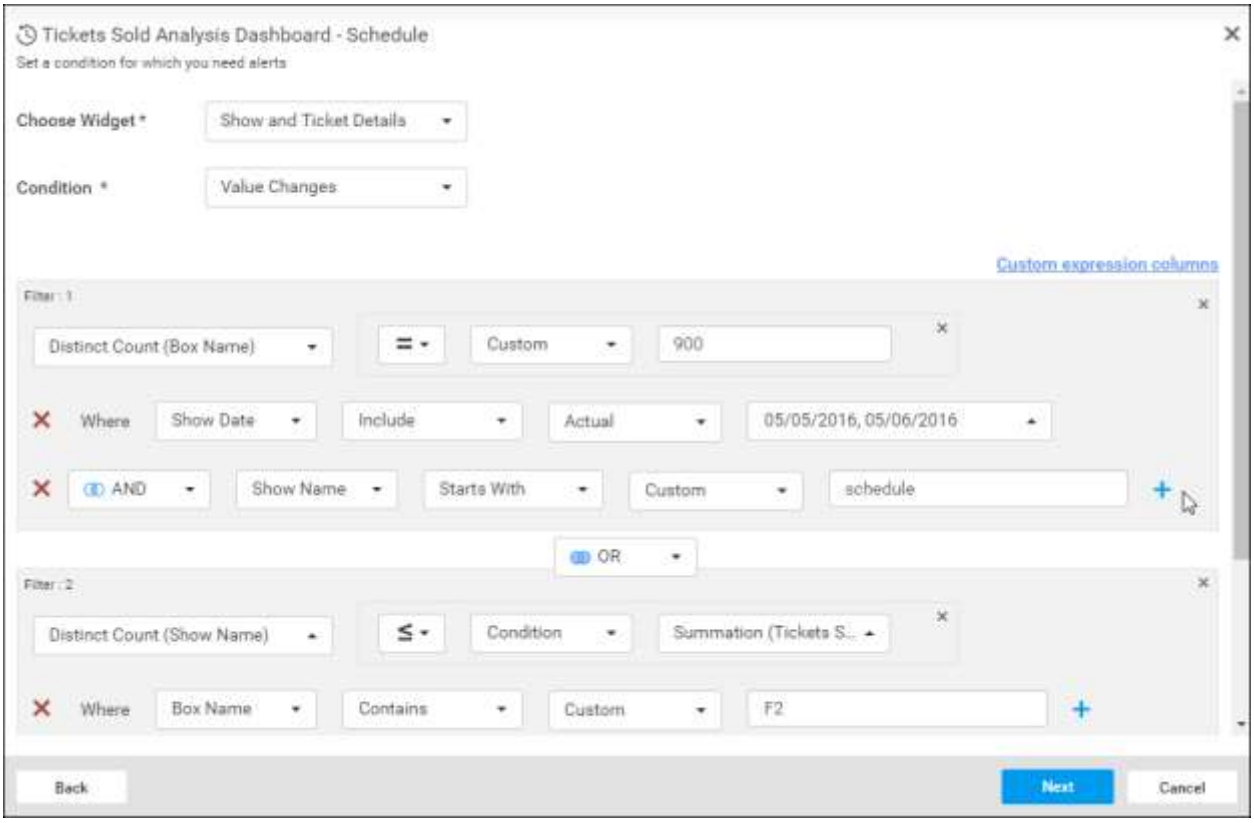
- Add Where condition for each filter by click on Add Where Condition.



- The column values can be compared with custom or actual values. Actual values are listed from the database and the custom values are the input provided by the user.





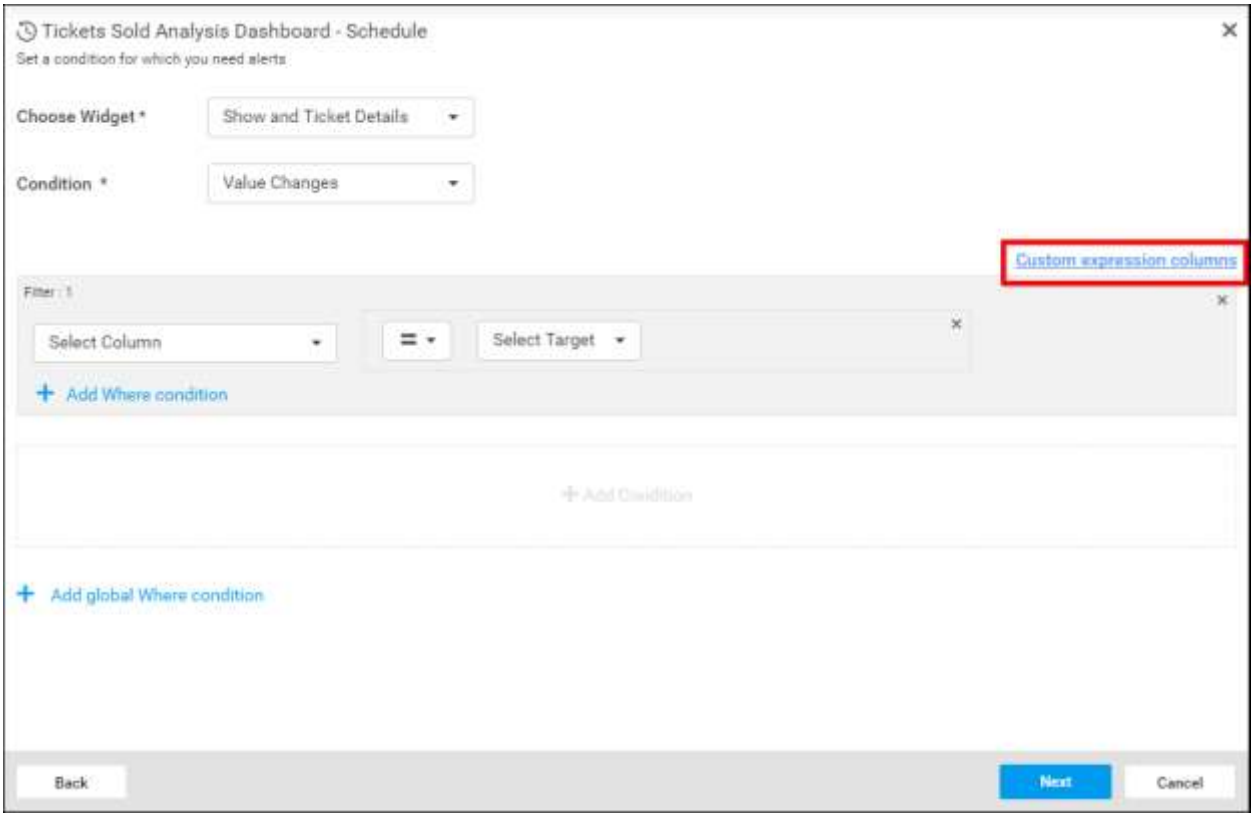


Custom Expression

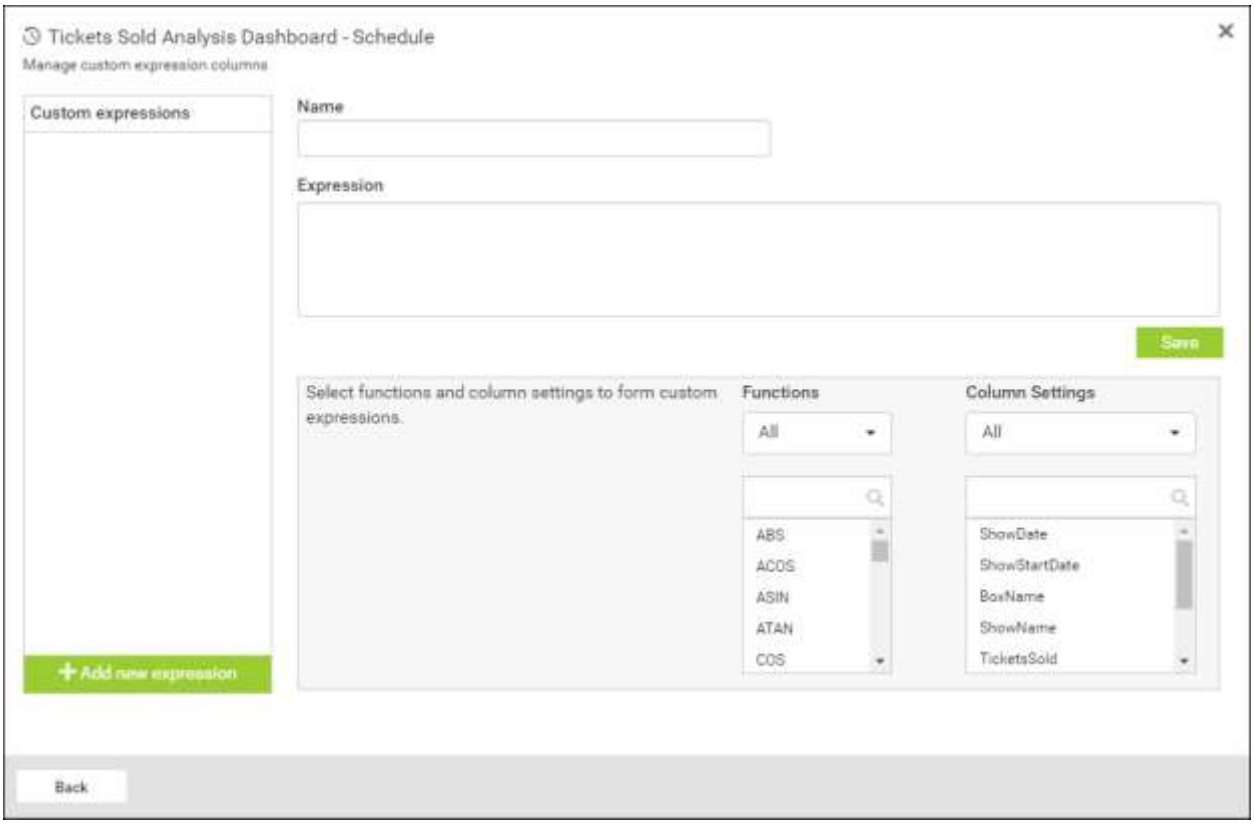
In Data alert, a new option is included to create expressions with the combinations of unbounded data source columns of selected widget and multiple functions. For instance, [Sum(Column1) - Sum(Column2)].

Steps to add custom expression

- After selecting the widget, click on the Custom expression columns label at the top of filter in data alerts screen.



- Custom expression dialog opens as like below,

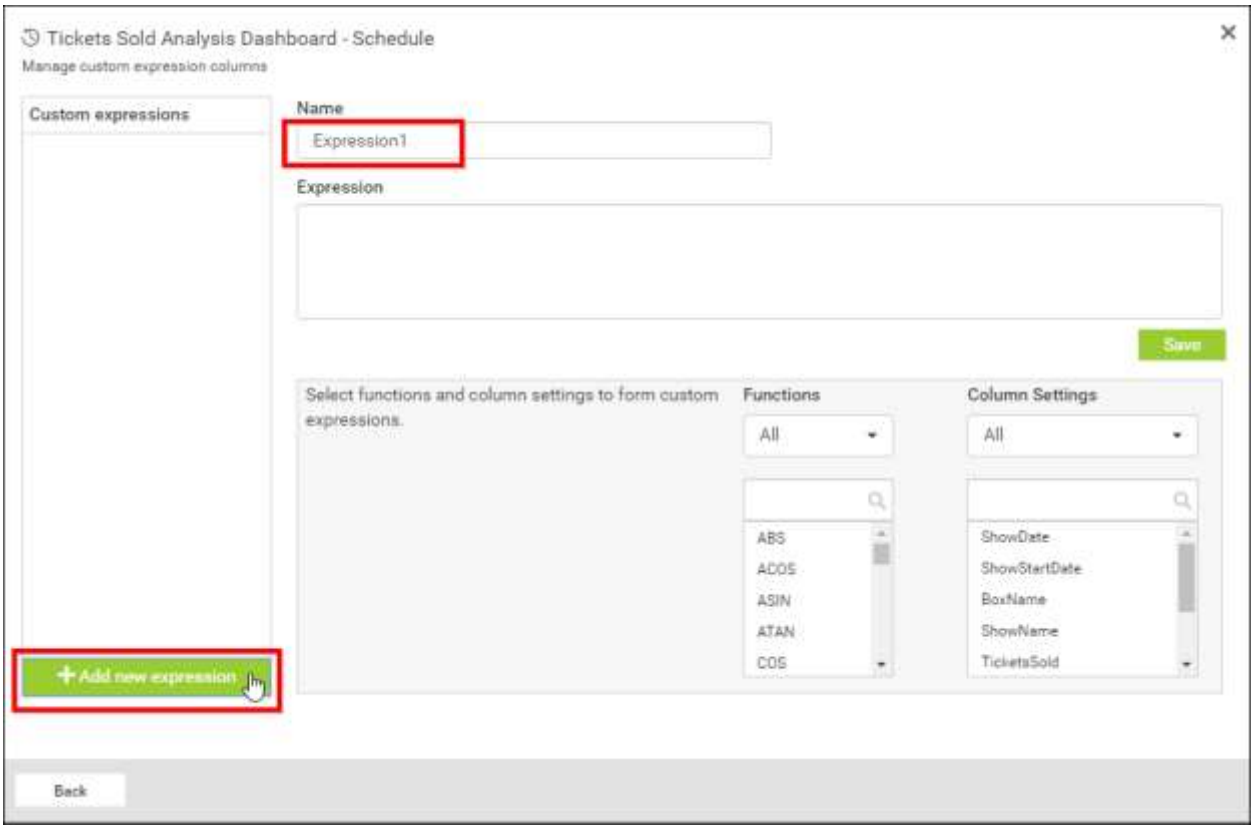


In above dialog, Custom expressions column lists all the saved expressions of the appropriate datasource.

You can add new expression by providing values to name and expression input fields.

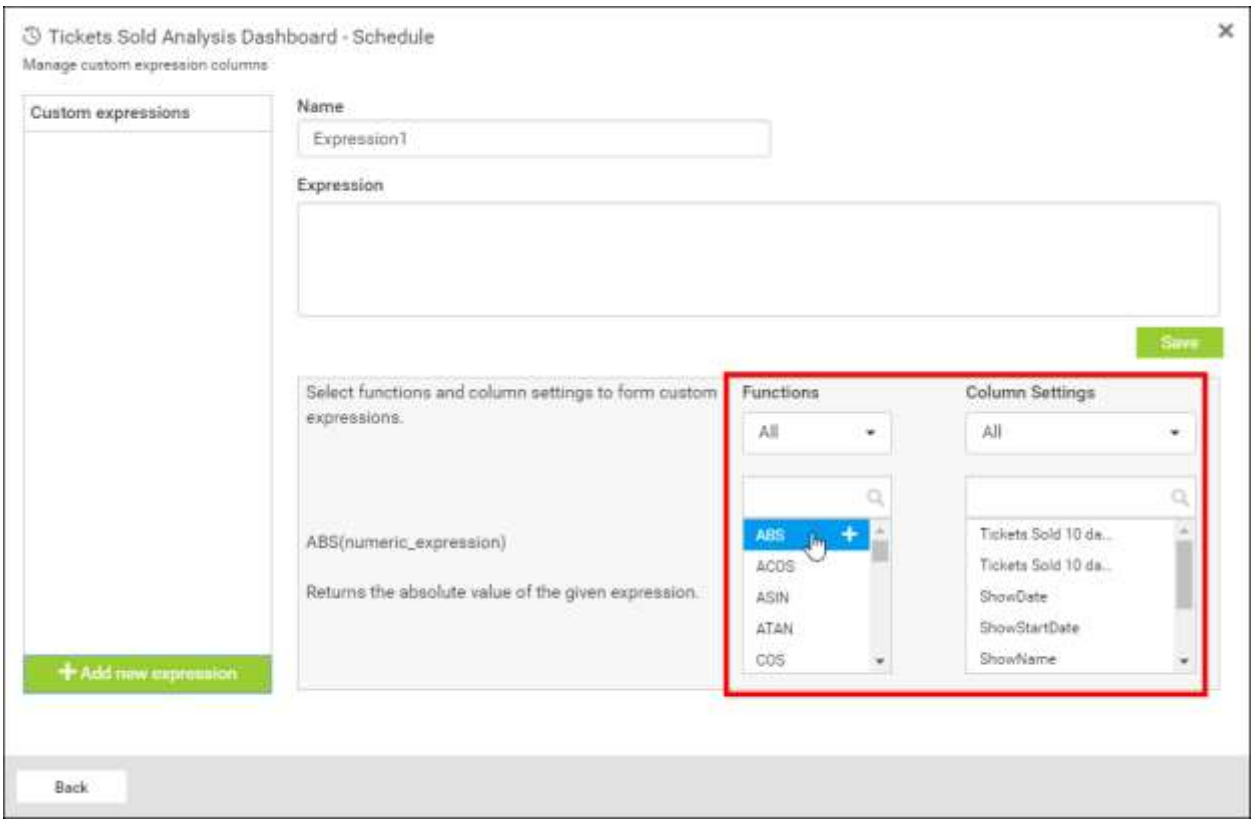
Functions and column settings drop downs provides multiple functions and datasource columns of selected widget to create expressions.

- Click on the Add new expression button to add default expression name in name input field.



Expression name can also be changed.

- Expressions can be formed with the combinations of functions and column names, from the functions and column settings drop downs.

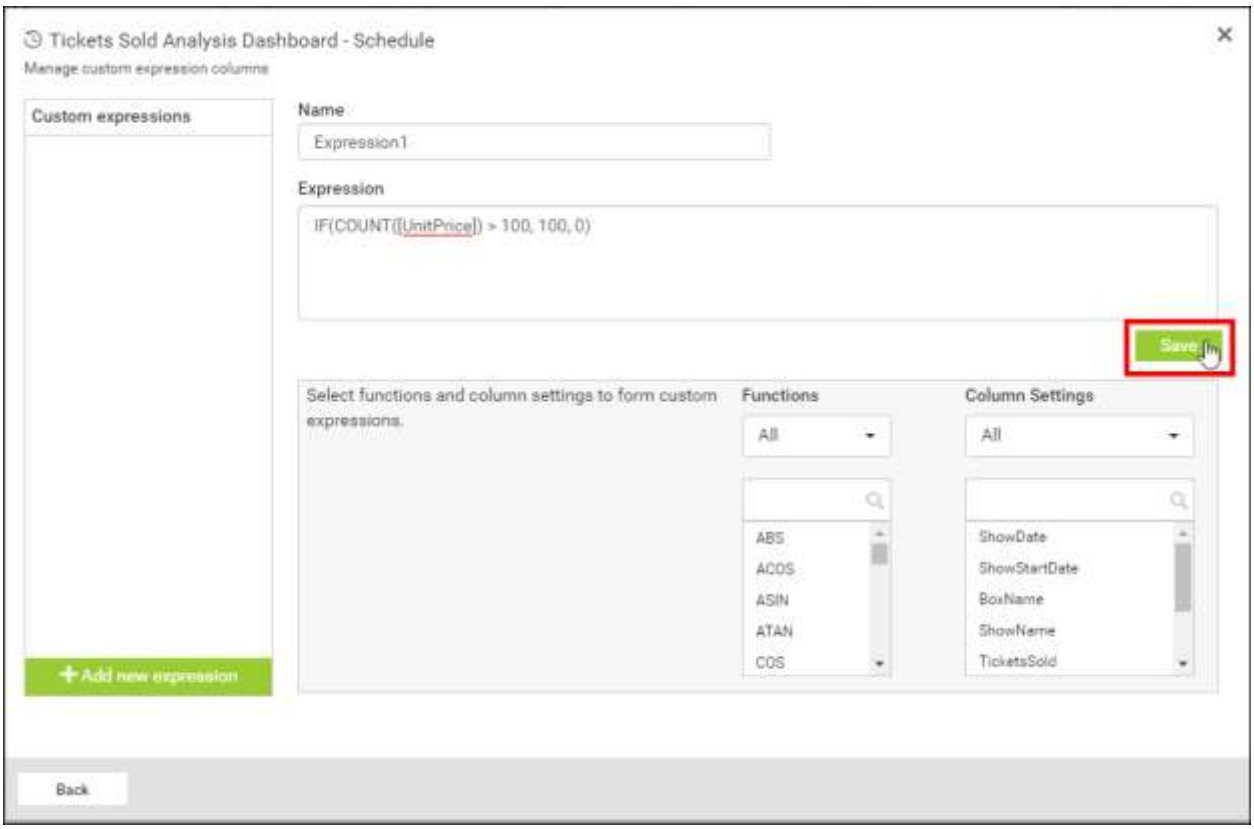


Functions and column settings are categorized under various types and they can also be filtered by the first dropdowns underneath its headings.

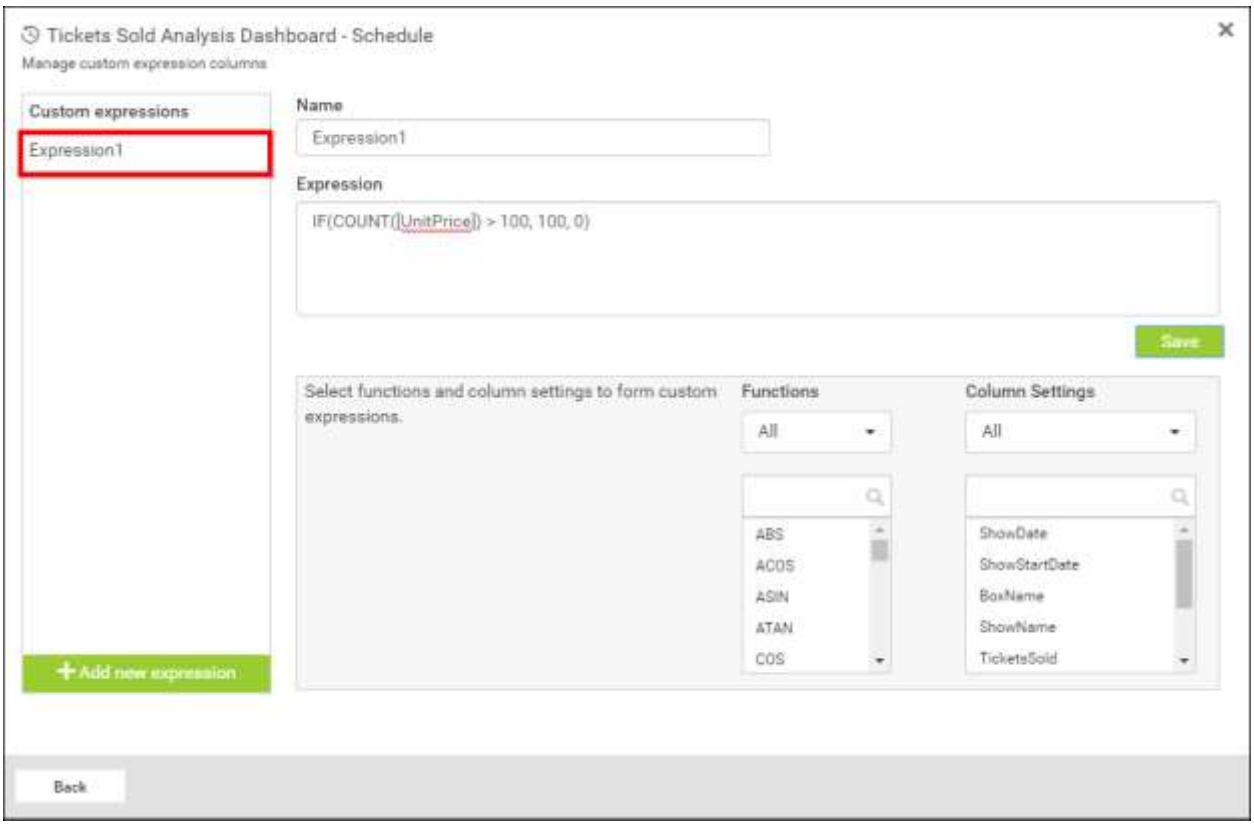
While hovering over the functions, the syntax and short description of them are displayed to its left.

Expression can be inserted into its input field by clicking on the required values from the functions and column settings drop downs, the inserted values are added in appropriate cursor position.

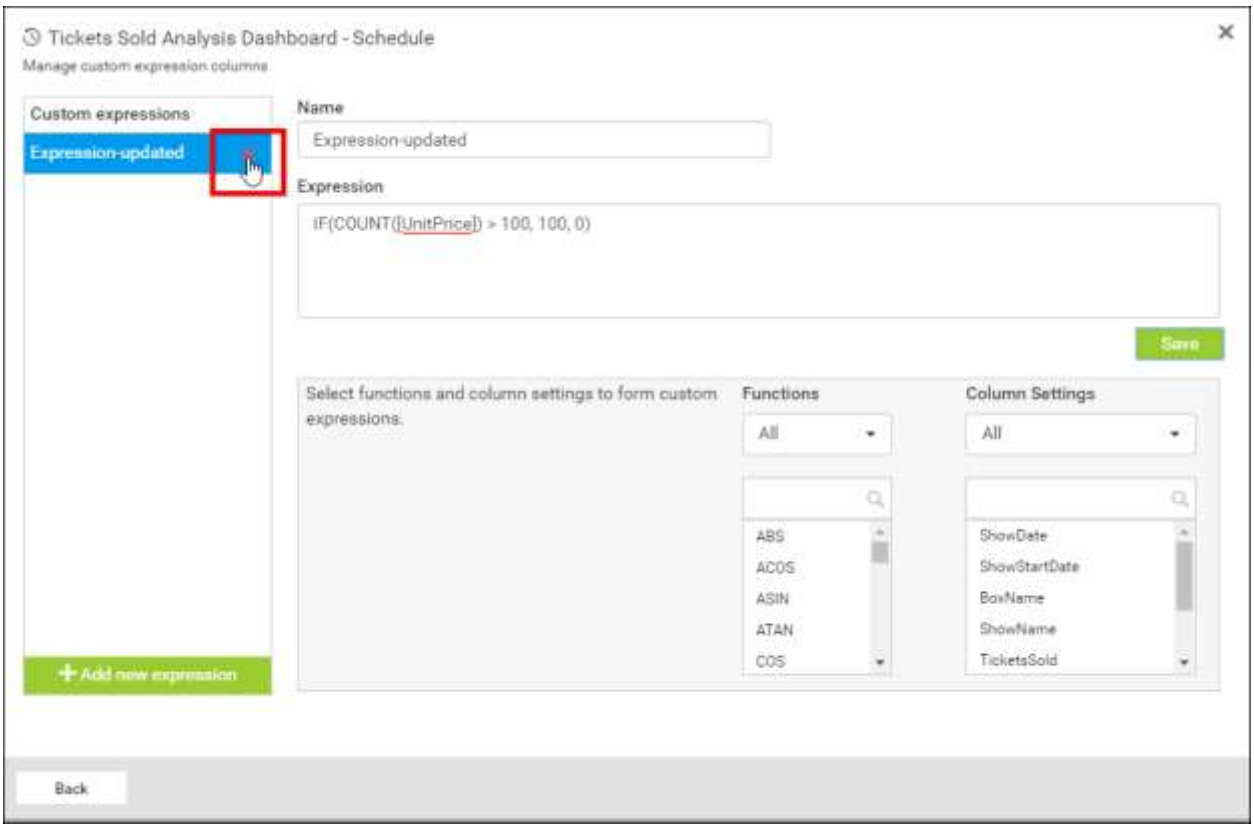
- Created expression can be added by clicking on the save button.



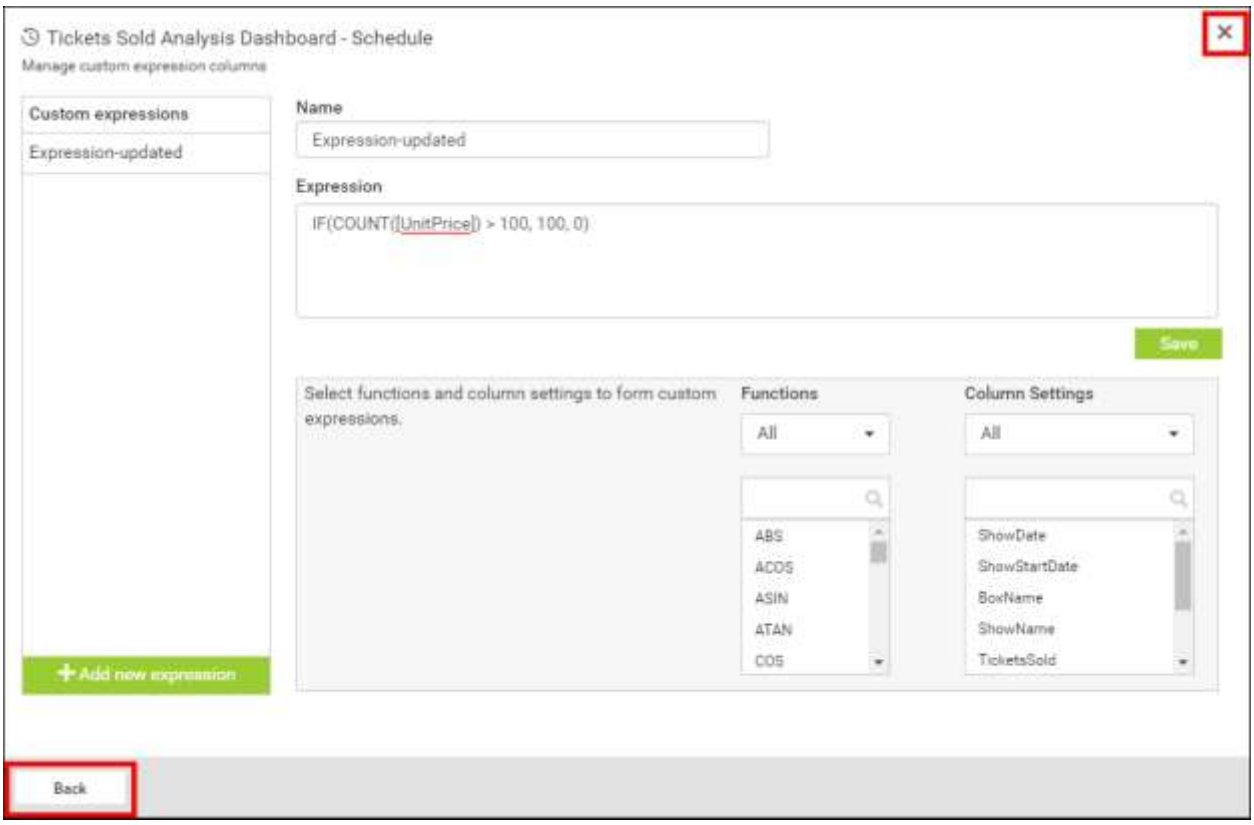
- Saved expression is listed under the custom expressions column as like below,



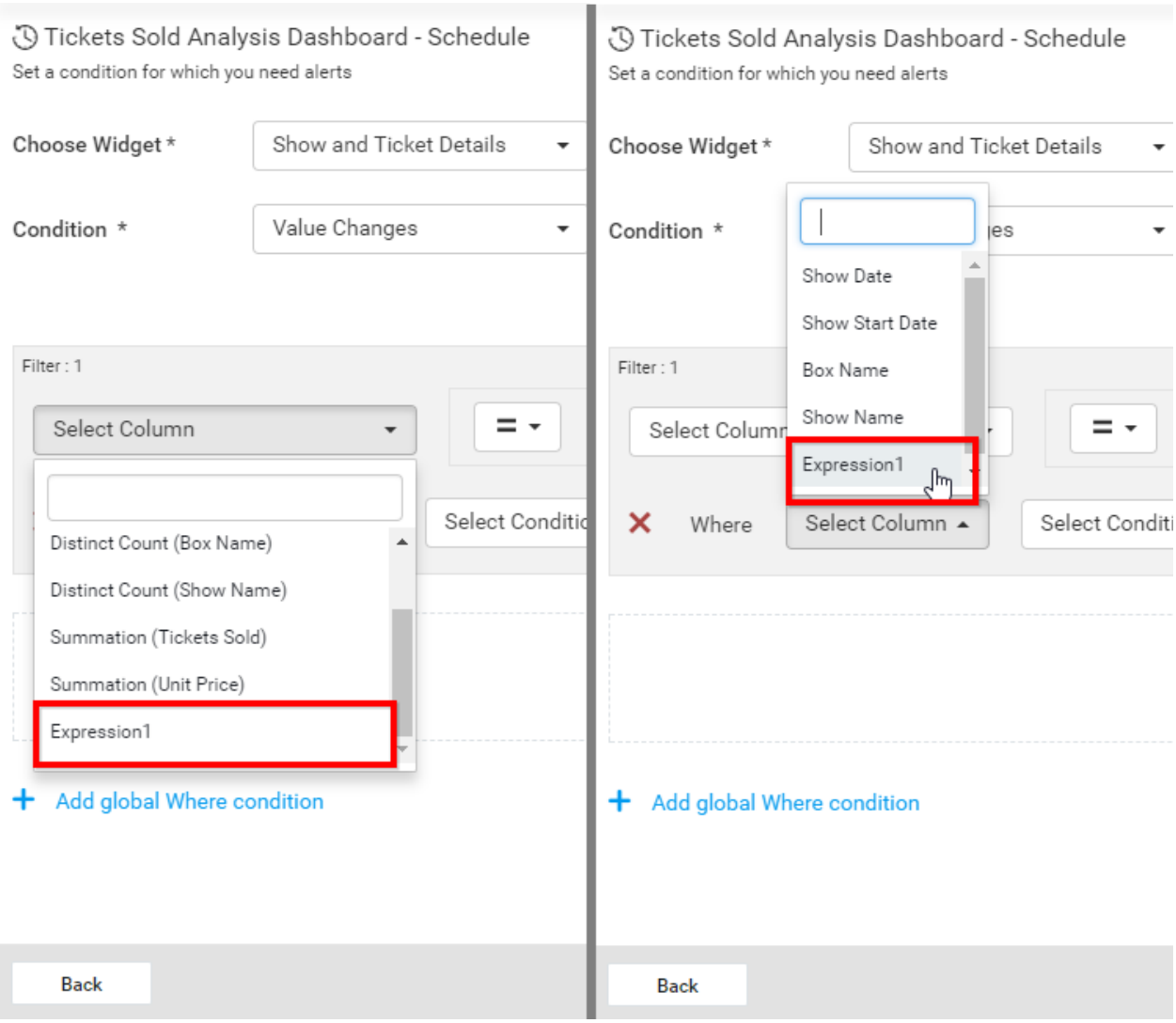
- When clicked on any expression in this list, the appropriate name and expression is filled in the right panel. You can update this saved expression and can be saved again.
- Expressions can be deleted, by clicking on the close icon, which is visible on hovering the custom expressions list.



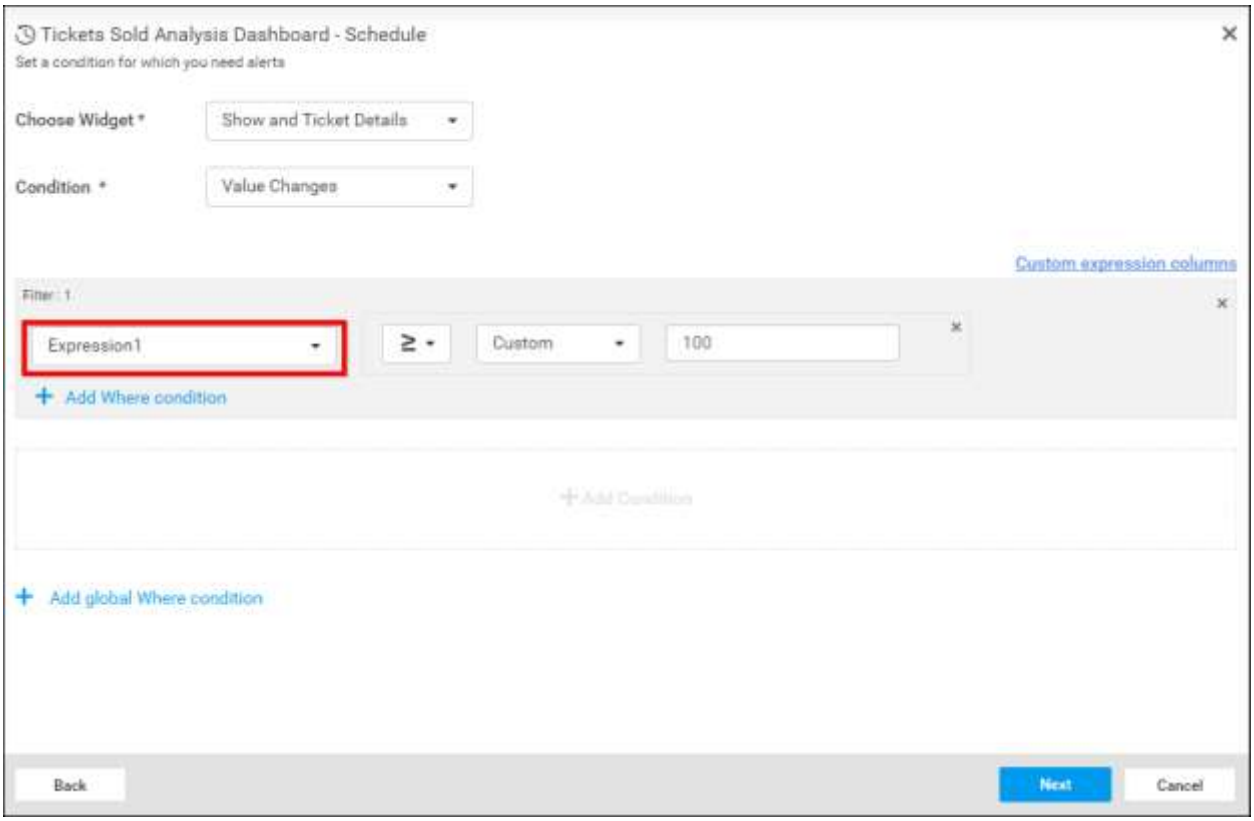
- Click on the back button to close the custom expression dialog.



- Once the dialog is closed, all the saved expressions are listed under all the select column drop downs of both measure and dimensions in data alerts screen as like below.

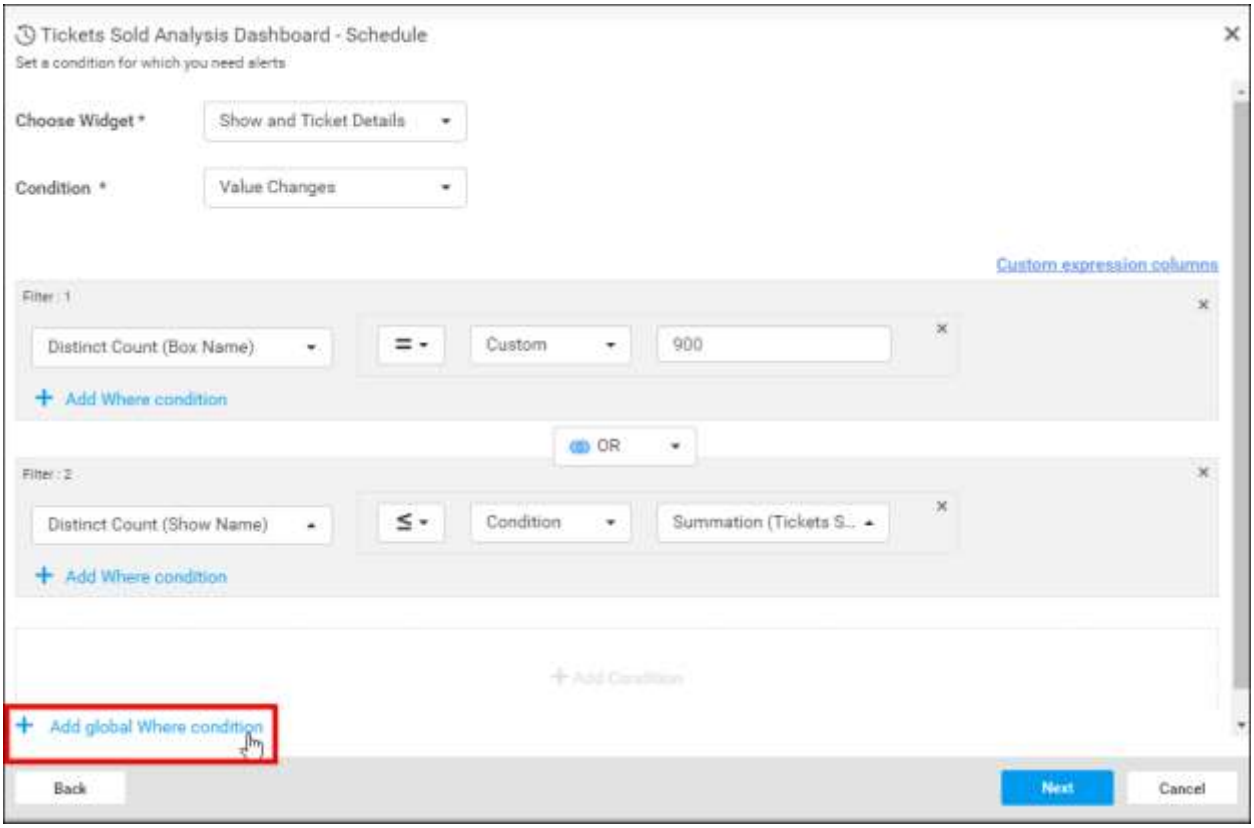


- You can use the saved expressions like database columns in data alerts screen as like below.



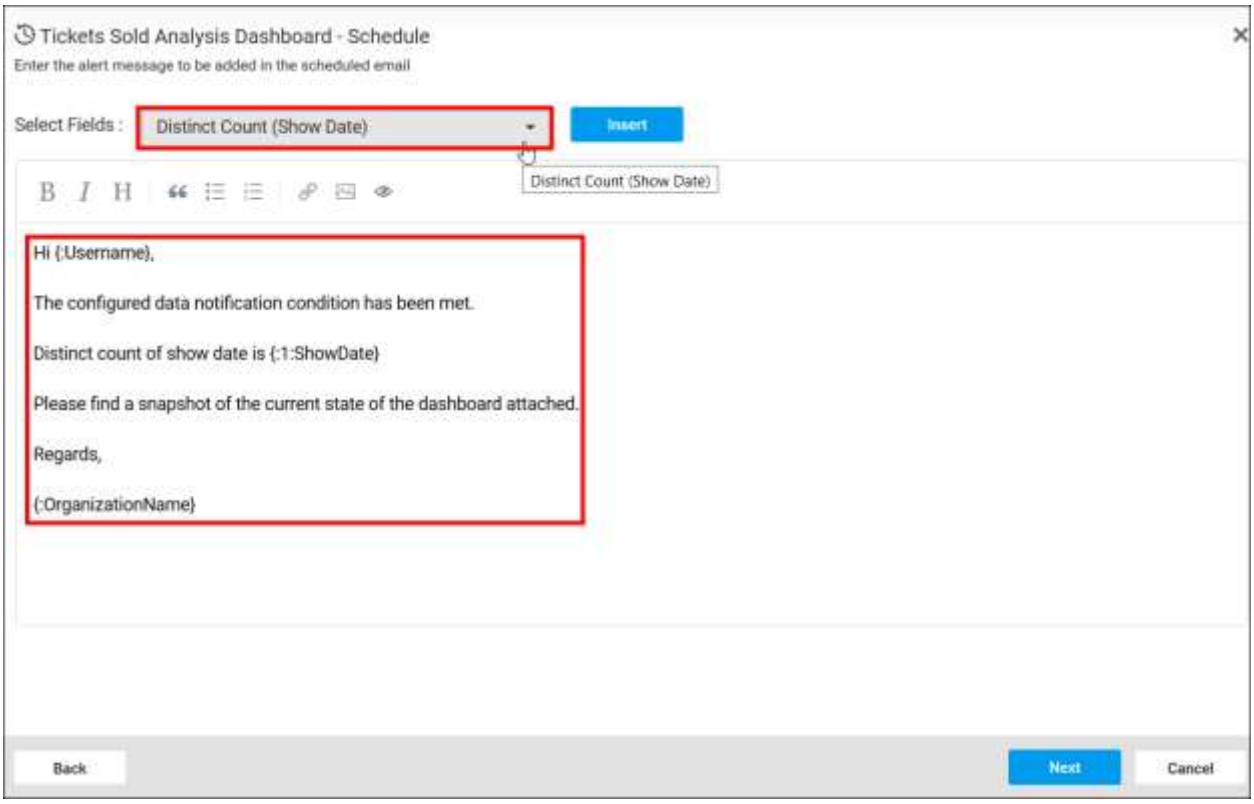
Global Where Condition

Add common Where condition for two or more filters by click on Add Global Where Condition.



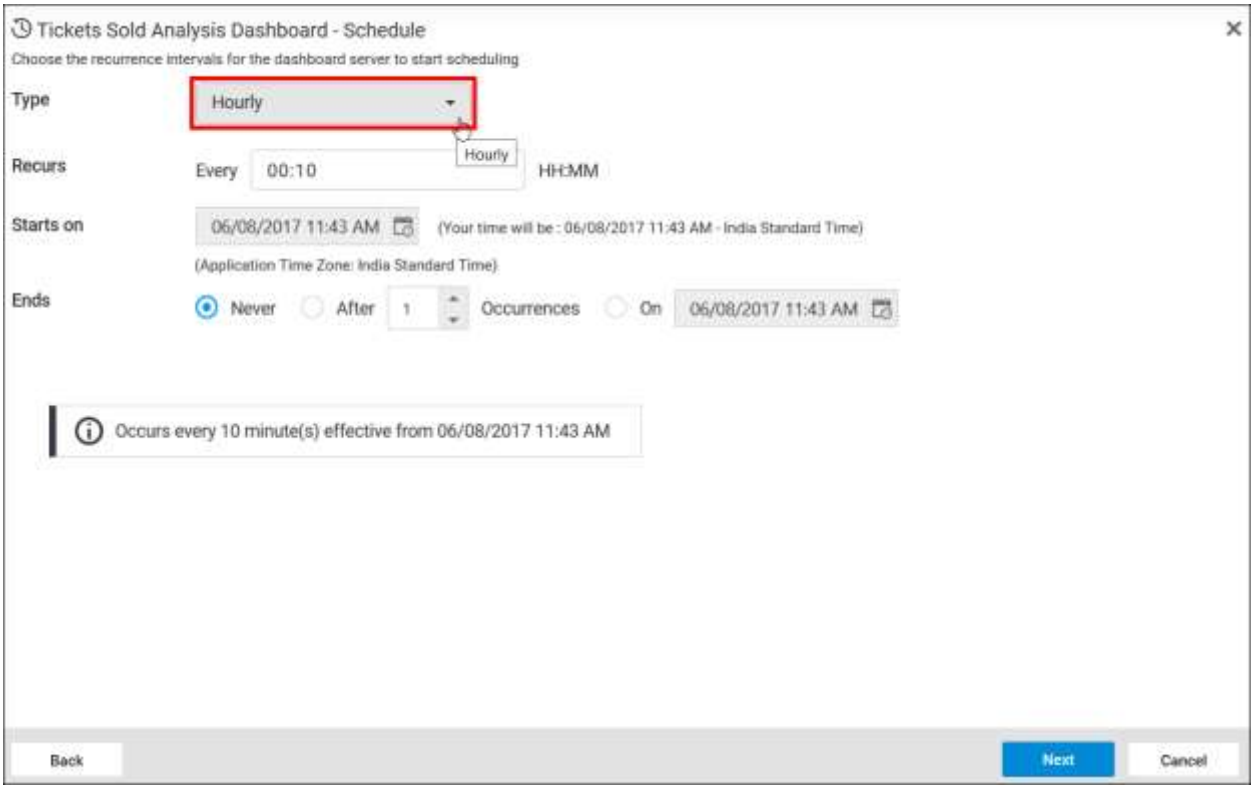
Email Editor

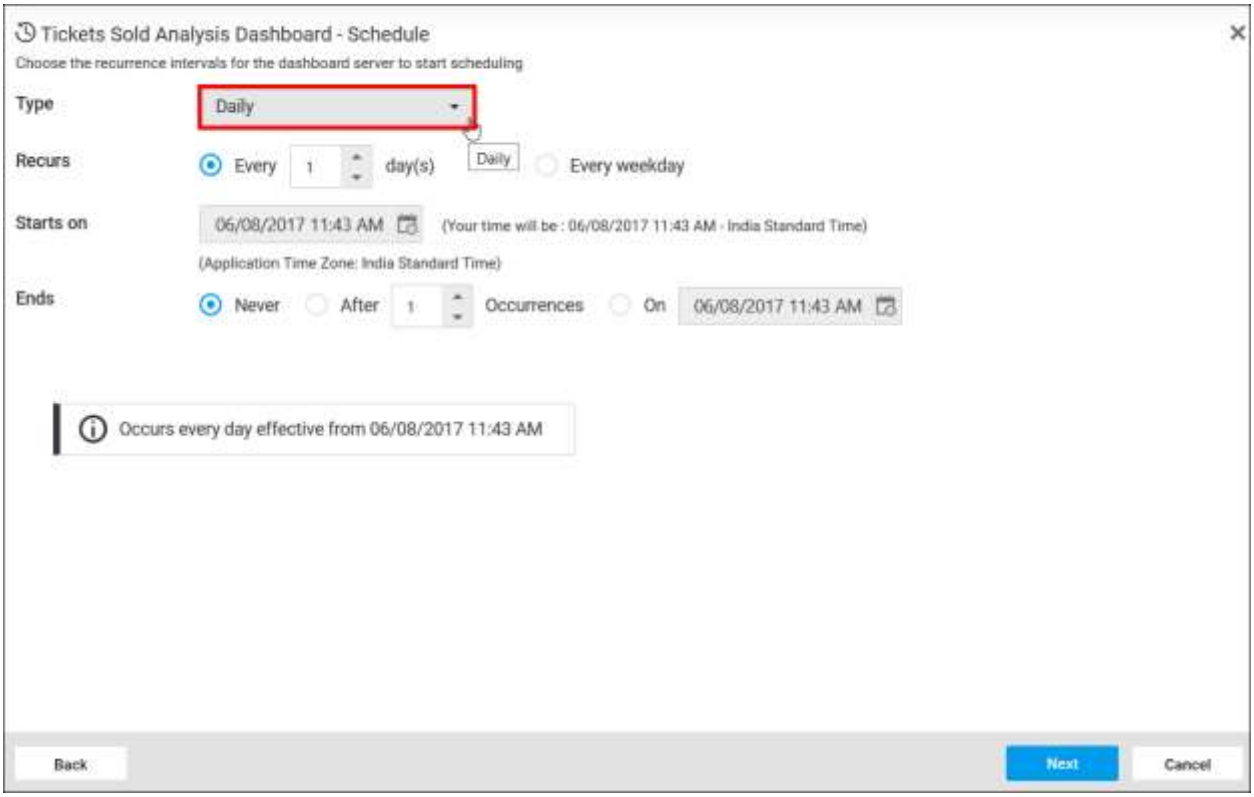
Email Editor provides an option to edit the Email body of the Data Alert mail and embed the selected database column values into the email body. Select Field describes the database columns that are chosen in Data Alert screen. The values of each inserted columns can be replaced while sending mail by Dashboard Server.



Recurrence Interval

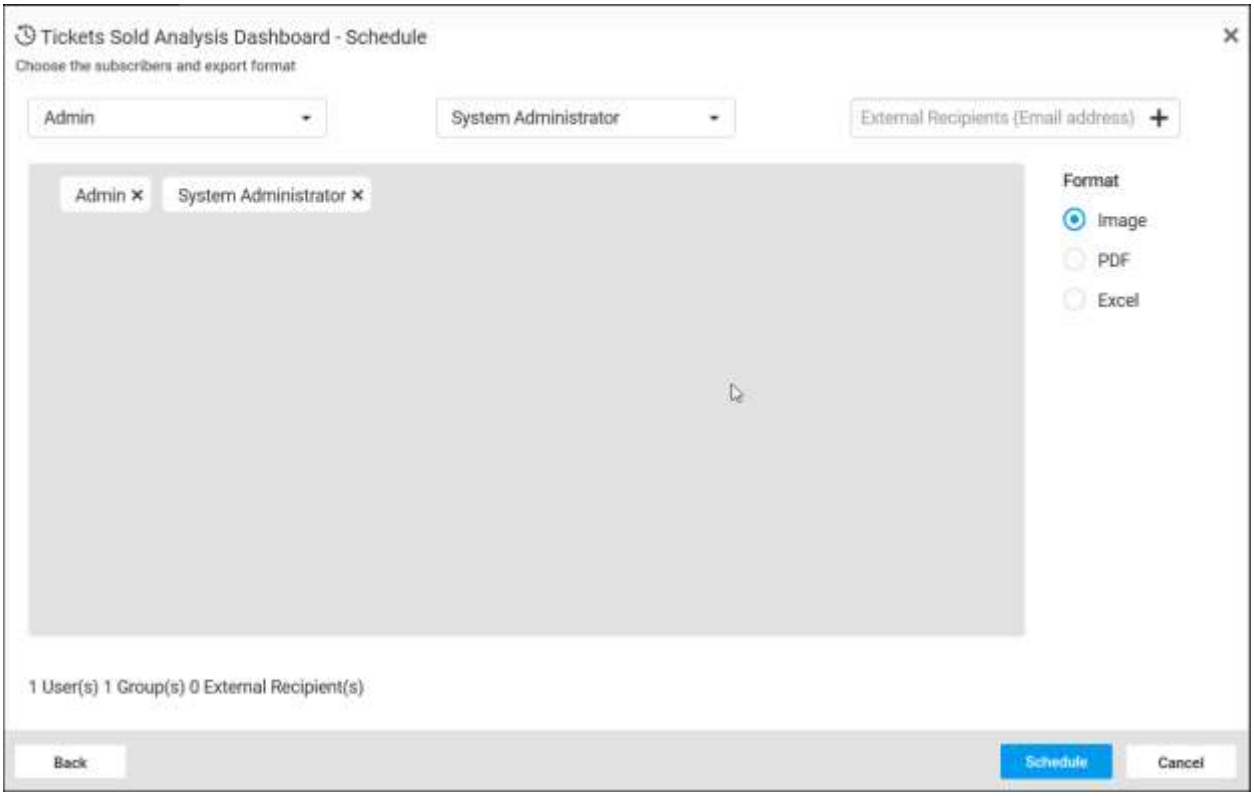
Dashboards can be scheduled on hourly, daily, weekly, monthly and yearly.





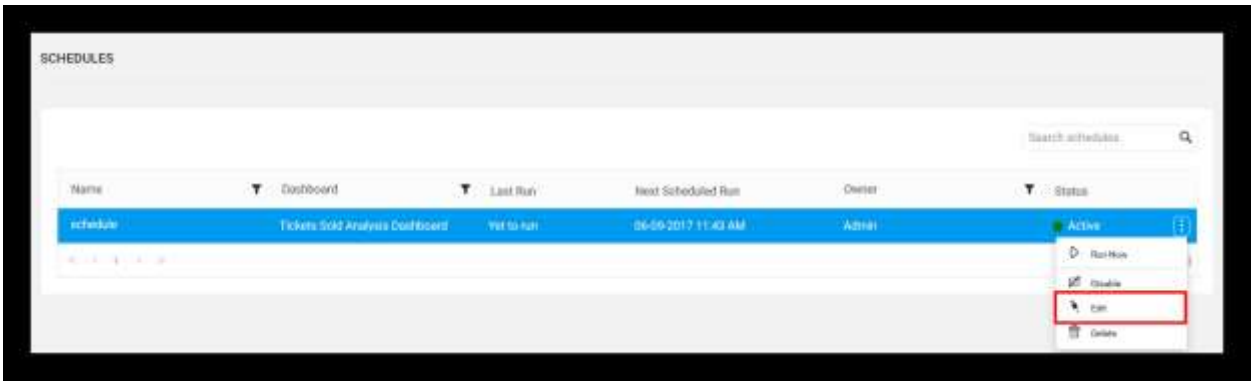
Choose Subscribers

Dashboards can be exported as Image, PDF and Excel. Exported dashboards can be sent to individual users or groups or to external recipients.



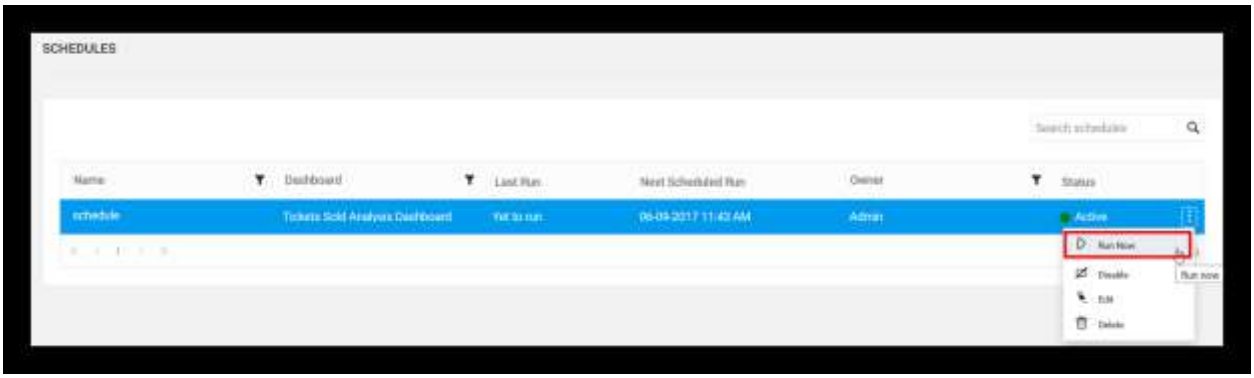
Edit Data Alert

Name, Data Alert, Schedule Type, Email Content, Export Format and the recipients can be changed in the Edit Schedule dialog box.



Run Now

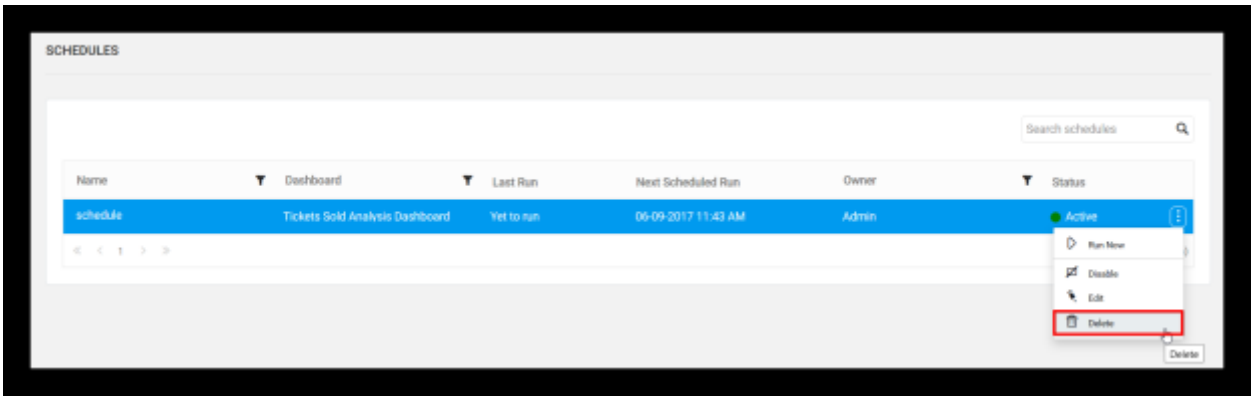
Schedules can be made to run on demand by using Run Now option in the schedule grid context menu. Dashboard get exported in the specified format and sent to the recipients if the threshold condition met.



Delete Data Alert

Data Alert can be deleted from the dashboard server when it is no longer required.

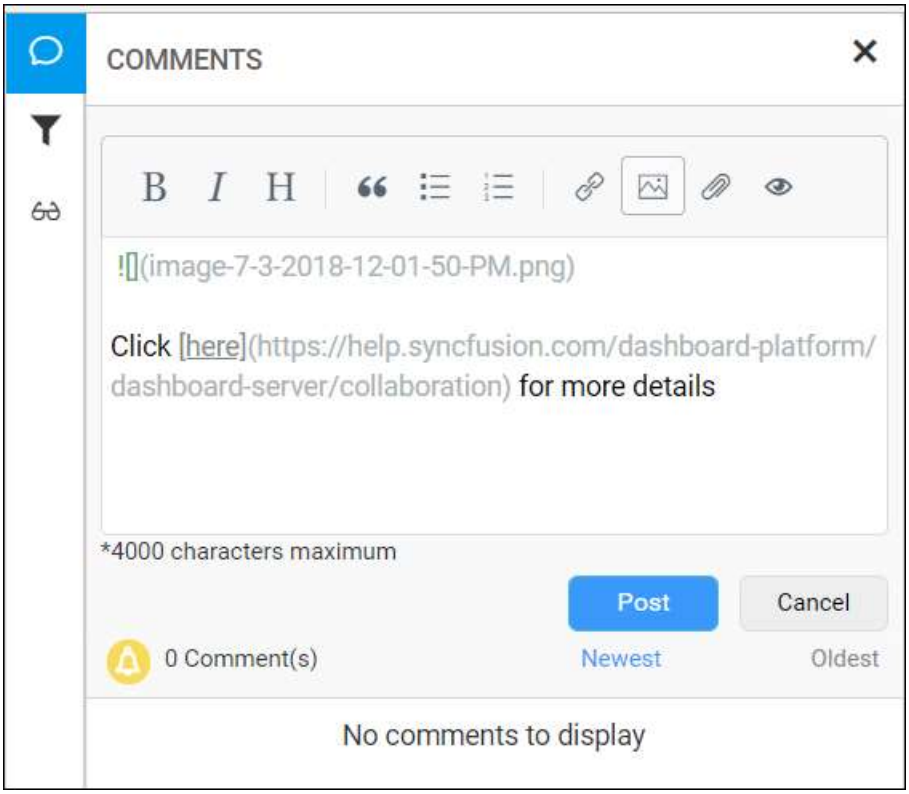
Click the Actions button in the schedules grid context menu and select Delete to delete the Data Alert.

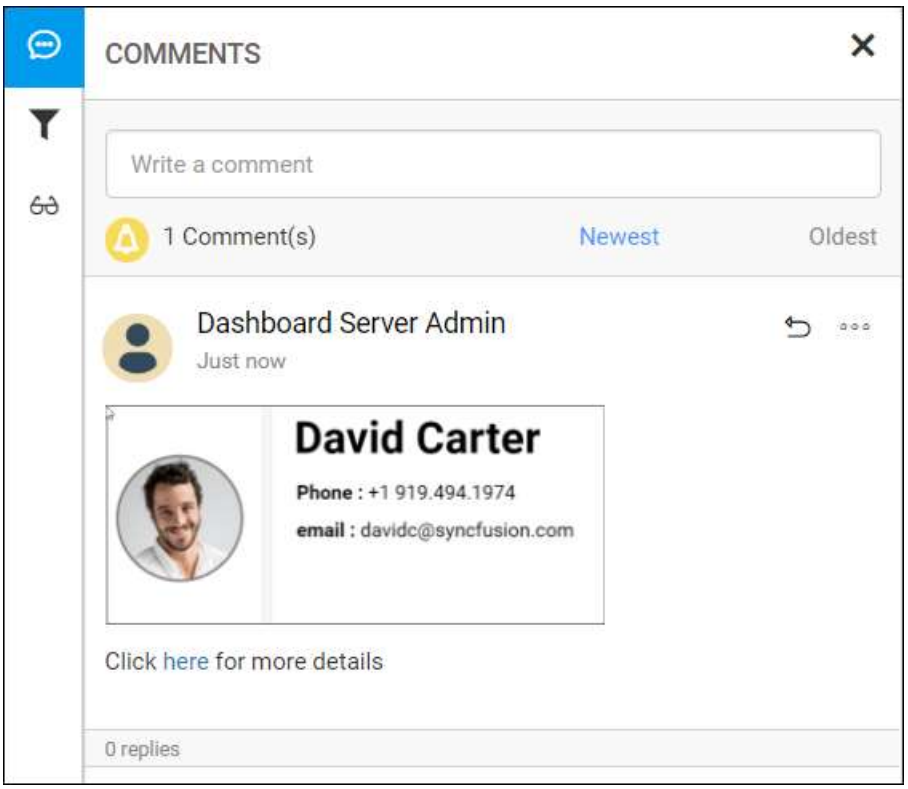


Collaboration

This section explains on how to collaborate with other users in the Syncfusion Dashboard Server by commenting on Dashboards and Widgets.

Collaboration feature in the Dashboard Server allows users to write text and add image comments on a Dashboard or widget to share with other users who have access to the Dashboard or widget. It is used to track events and provide insights into those events. Users can also add links to other Dashboards or widgets or any other external websites.

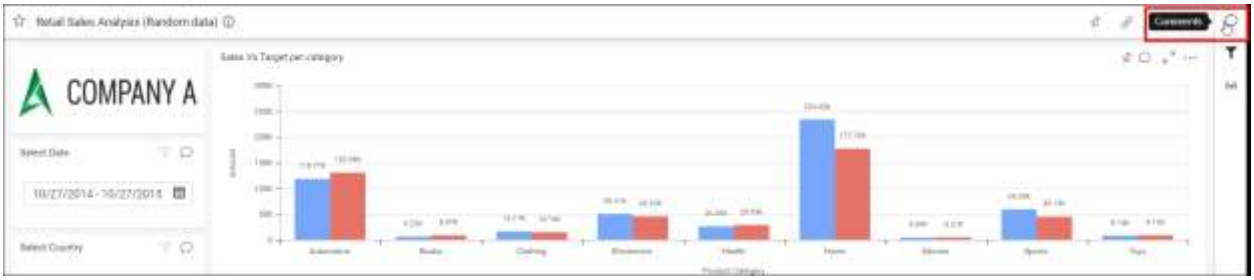




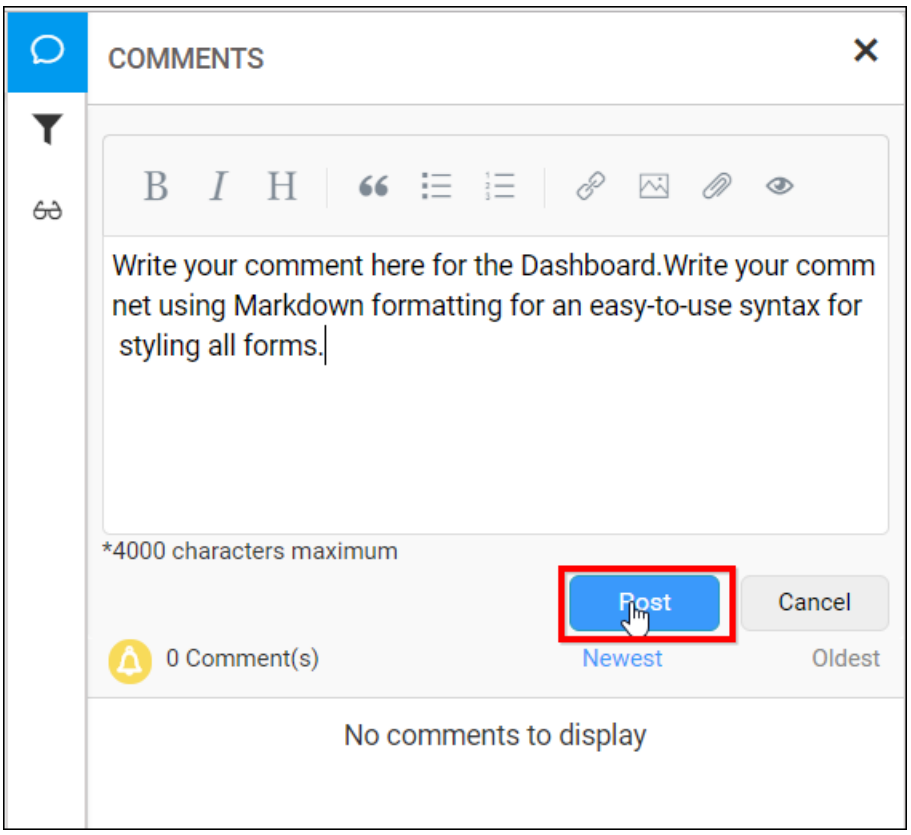
Anyone with access to a Dashboard can write comments. When a comment has been added to a Dashboard, users who have notifications enabled in their profile, receives notifications through email and system notifications. Learn more about Notifications in this section.

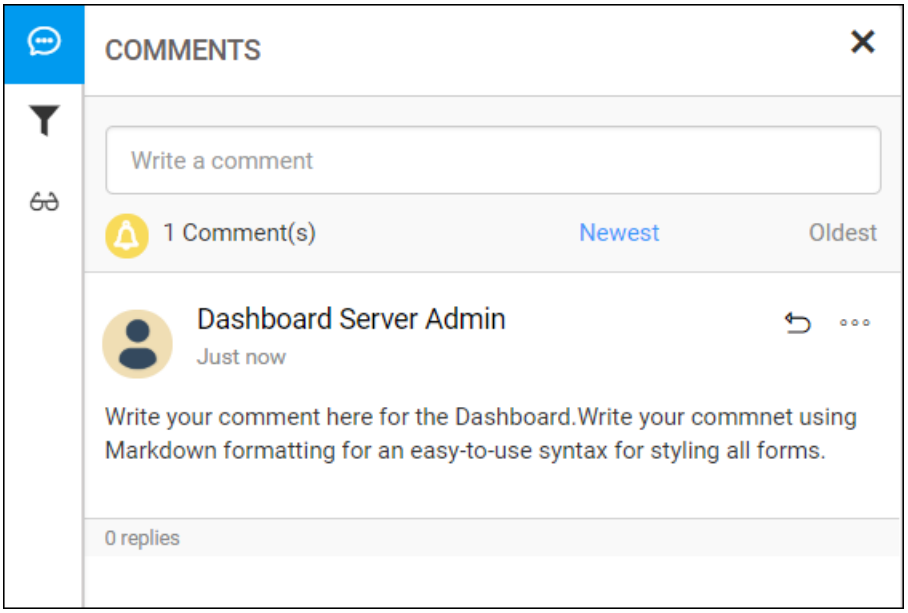
Post a new comment

To post a new comment, open the Dashboard and click on the comment icon in the top right corner as shown in the below image.



Type the comment in the text area and click on Post to save the comment for the Dashboard.

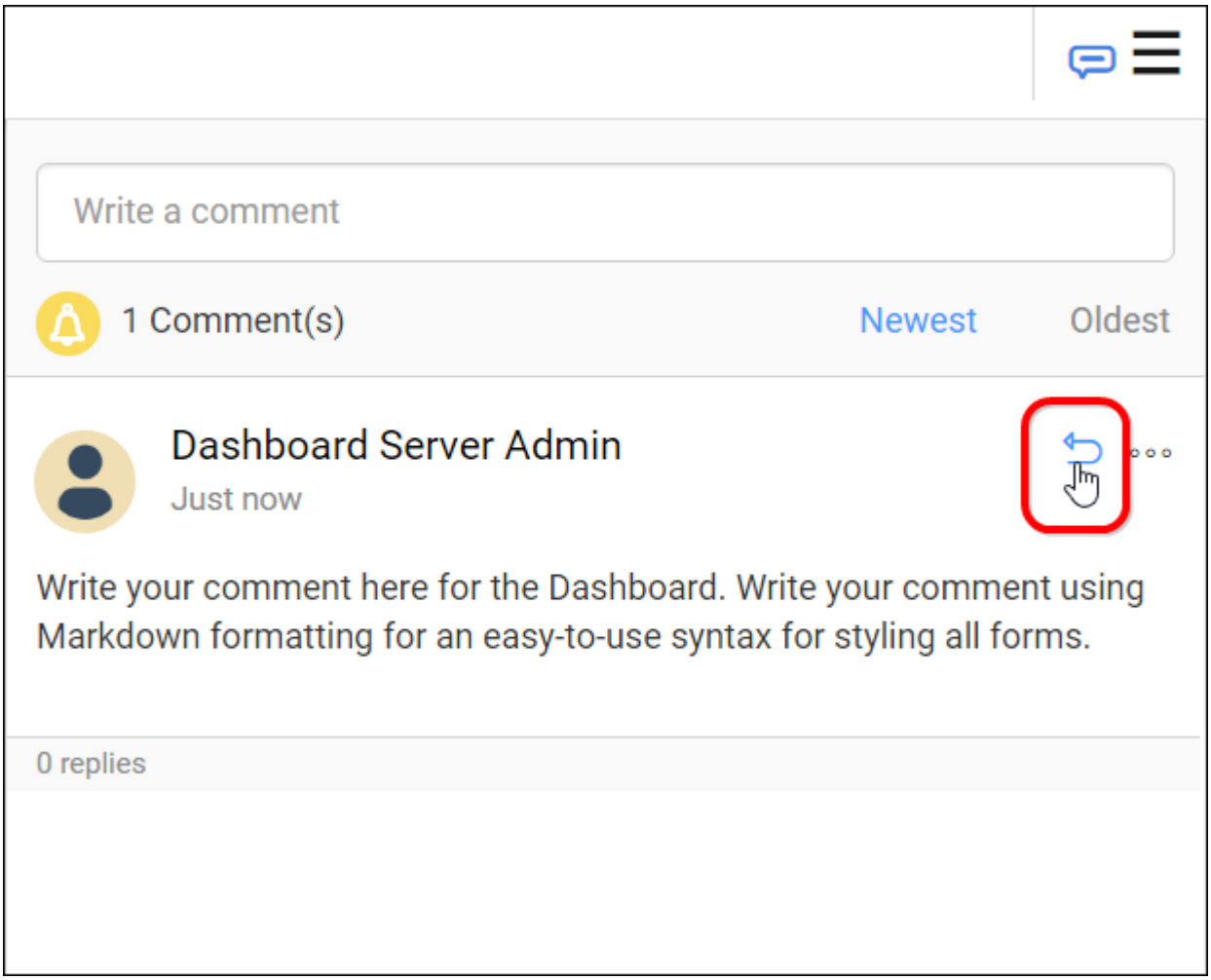




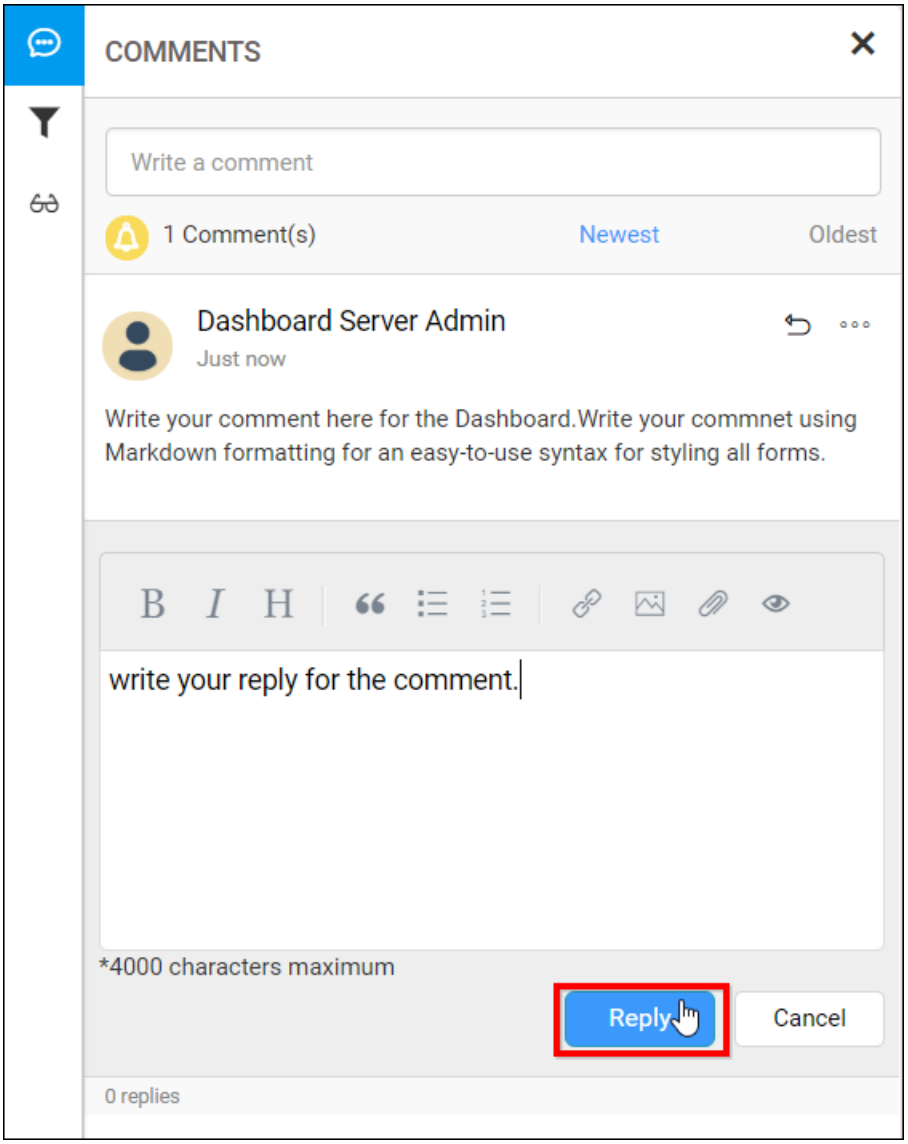
Note: Clipboard images can also be added along with the comments by simply copying an image and pasting in the text area.

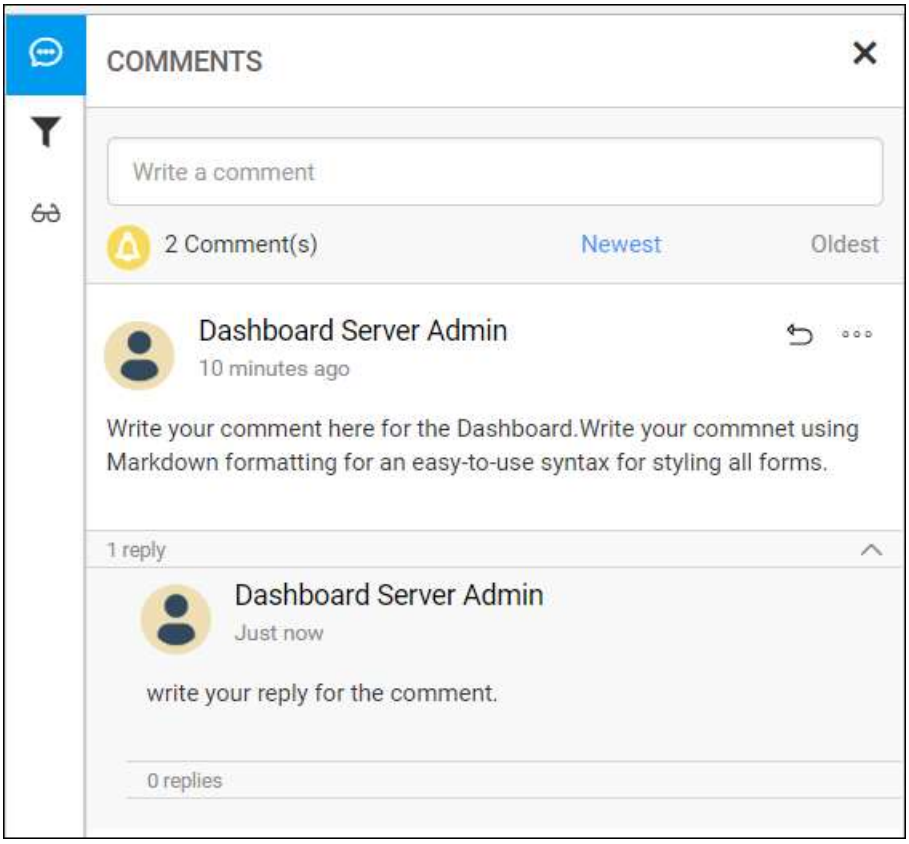
Reply to a comment

To reply to a comment, click on the Reply icon in the comment as shown in the below image.



Type the reply in the text area and click on Reply to save the reply for the comment on the Dashboard.

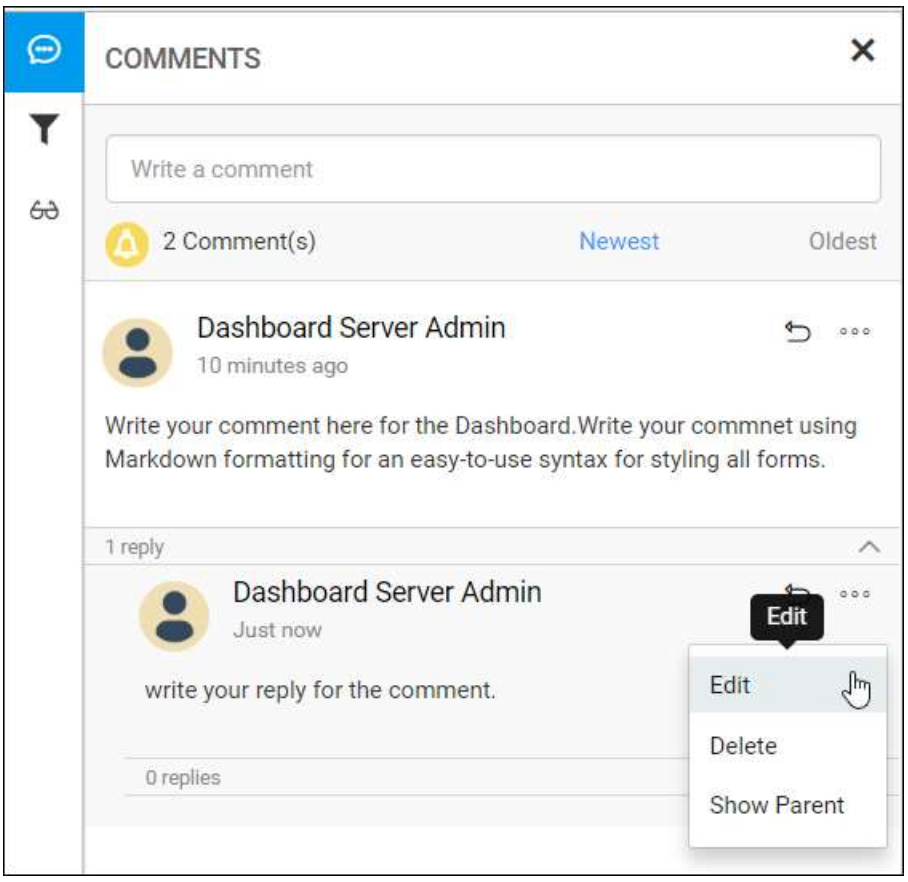




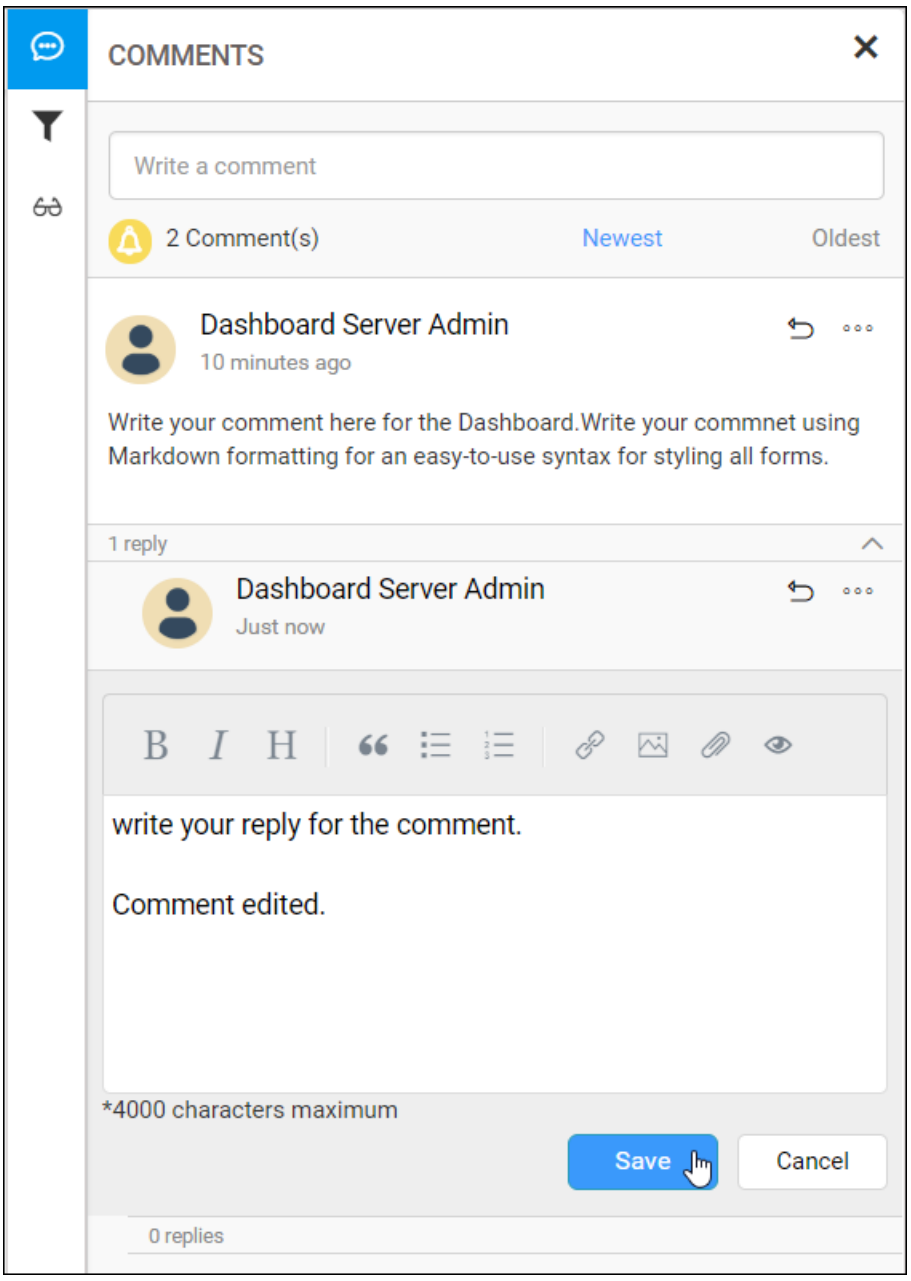
Note: You can also reply to a reply of a comment. This can be repeated a number of times.

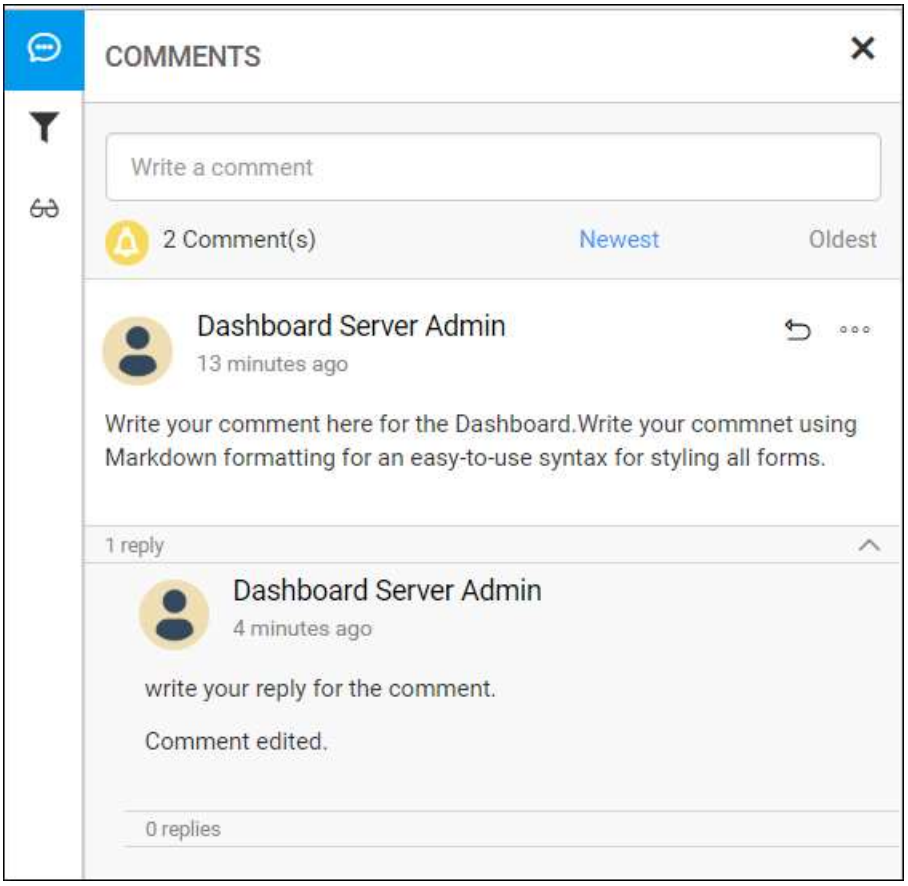
Edit a comment

To edit a comment, click on the option Actions button to get more options for a comment or a reply and click on the Edit button as shown in the below image.



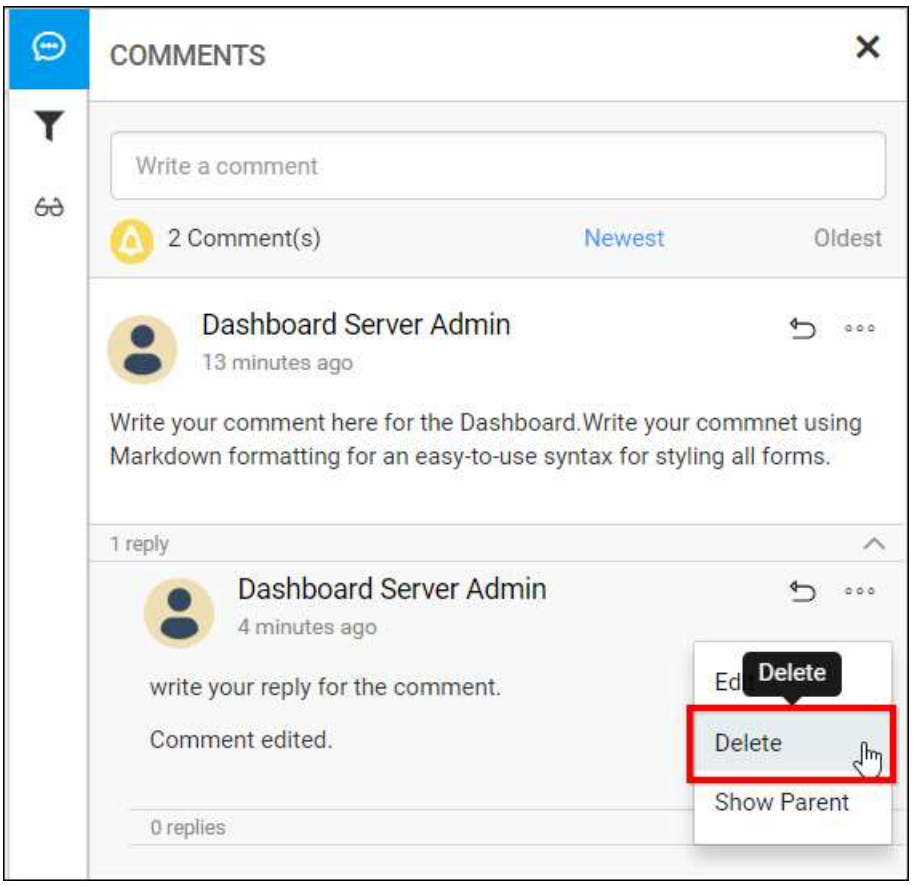
Edit the comment and click on Save to save it.





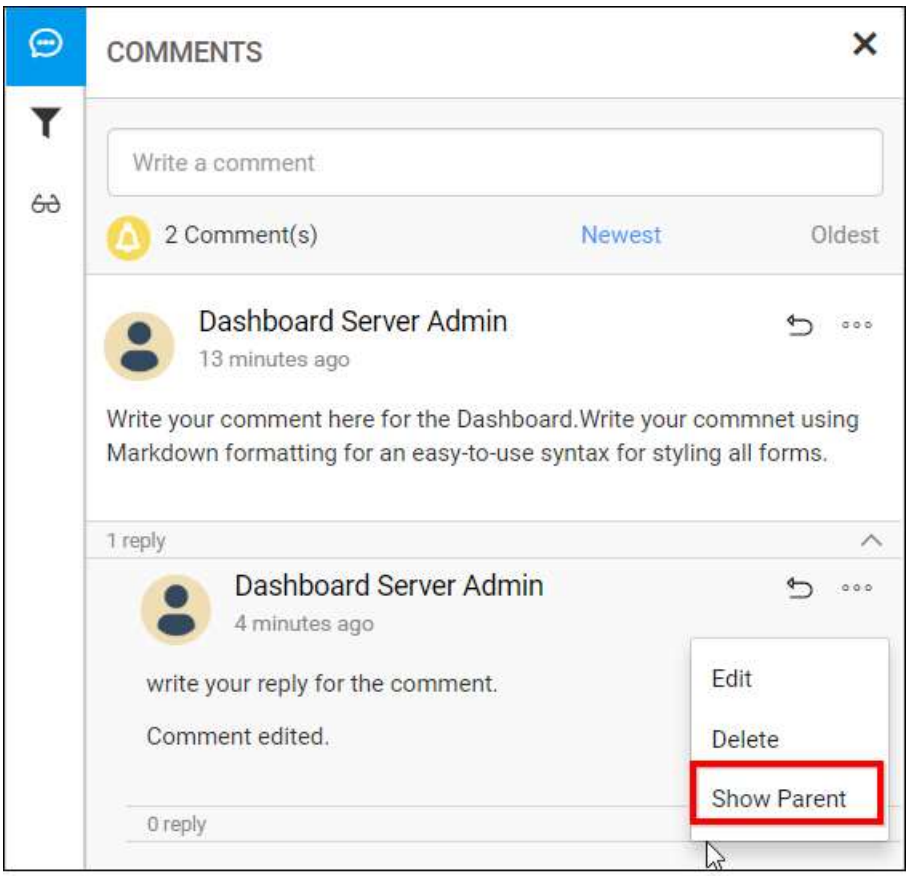
Delete a comment

To delete a comment, click on the option Actions button to get more options for a comment or a reply and click on the Delete button as shown in the below image.

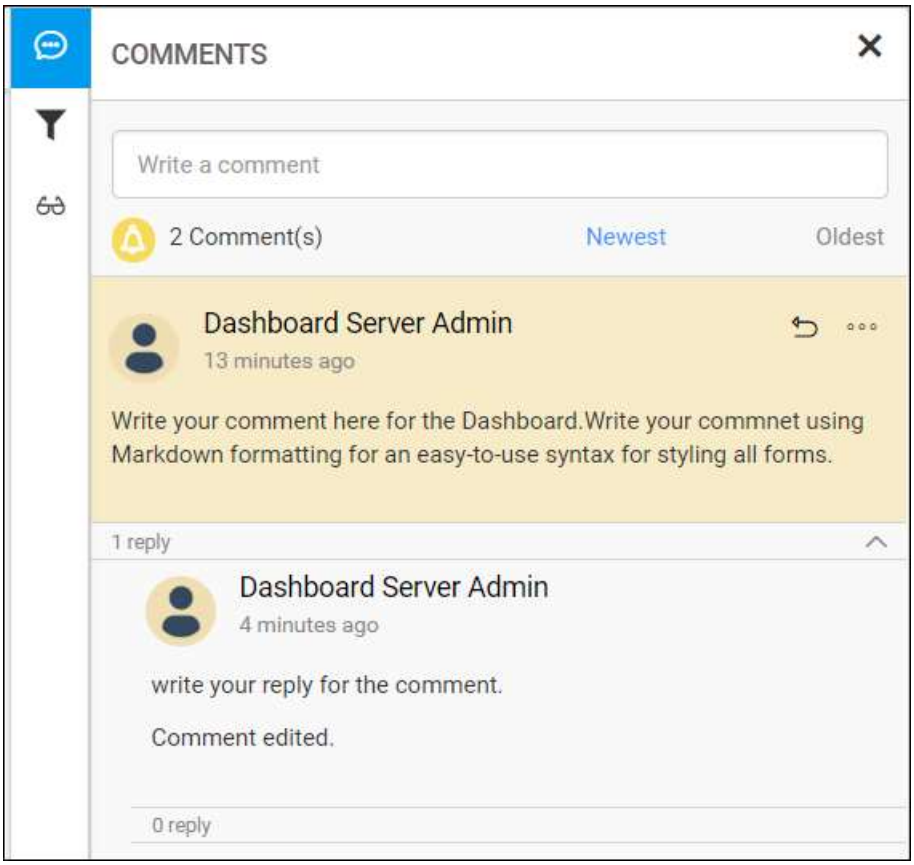


Show parent comment of a reply

To know the parent comment of a reply or to know which comment the reply has been posted, click on the option Actions button and click on the Show Parent button as shown in the below image.



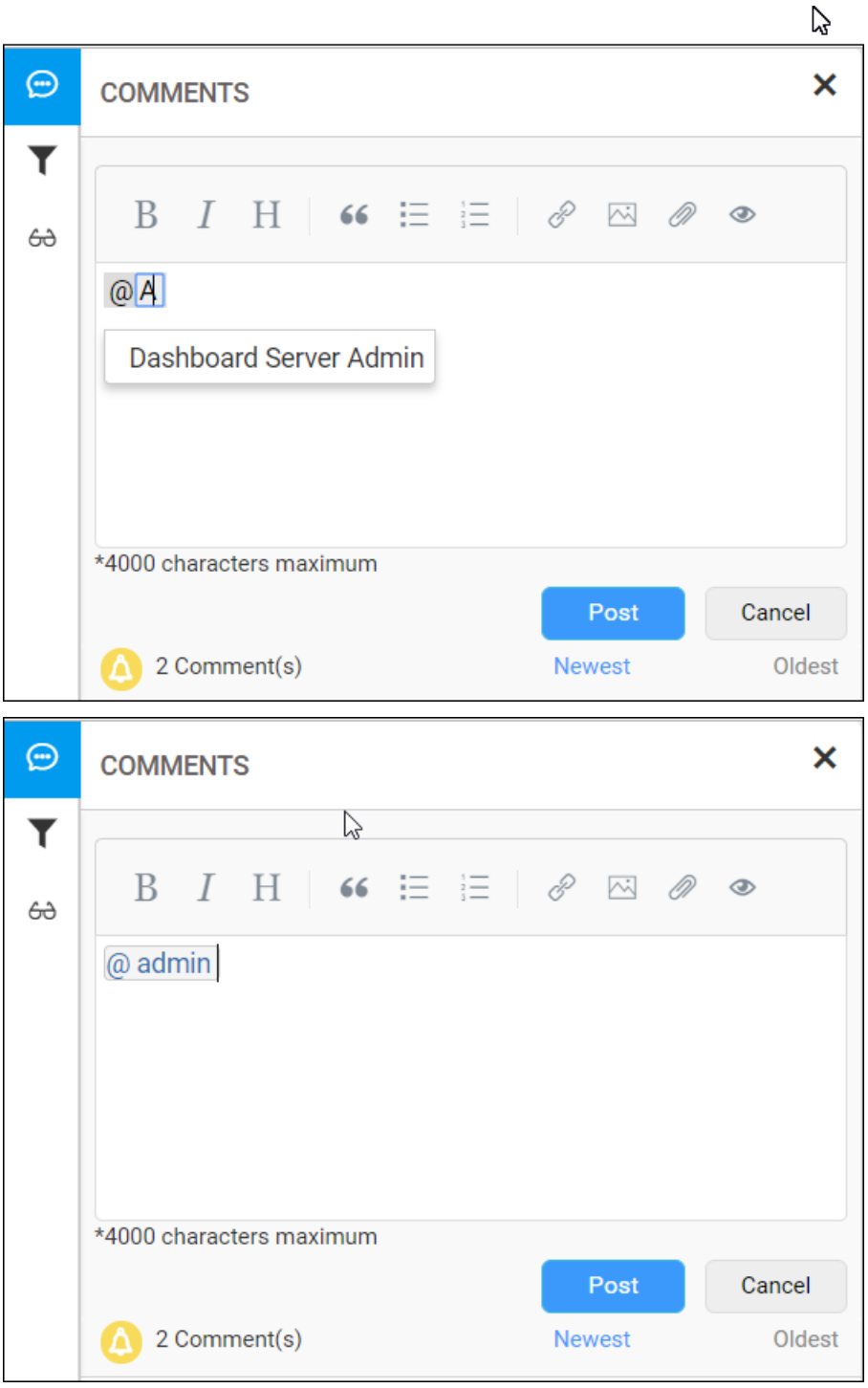
On clicking, the parent comment is highlighted for the reply as like below.



Mention Users in the comment

Users can be mentioned in the comments to notify them about the comment through email.

Type `@` followed by the user's name and from the list of possible names select the user to mention them in the comment.

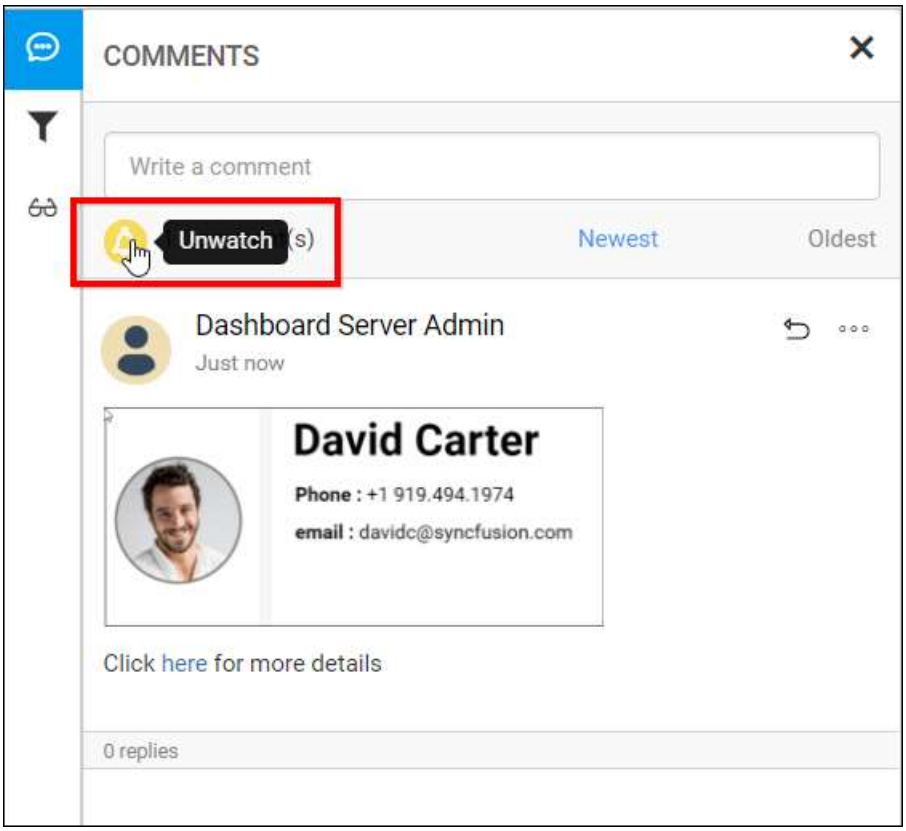


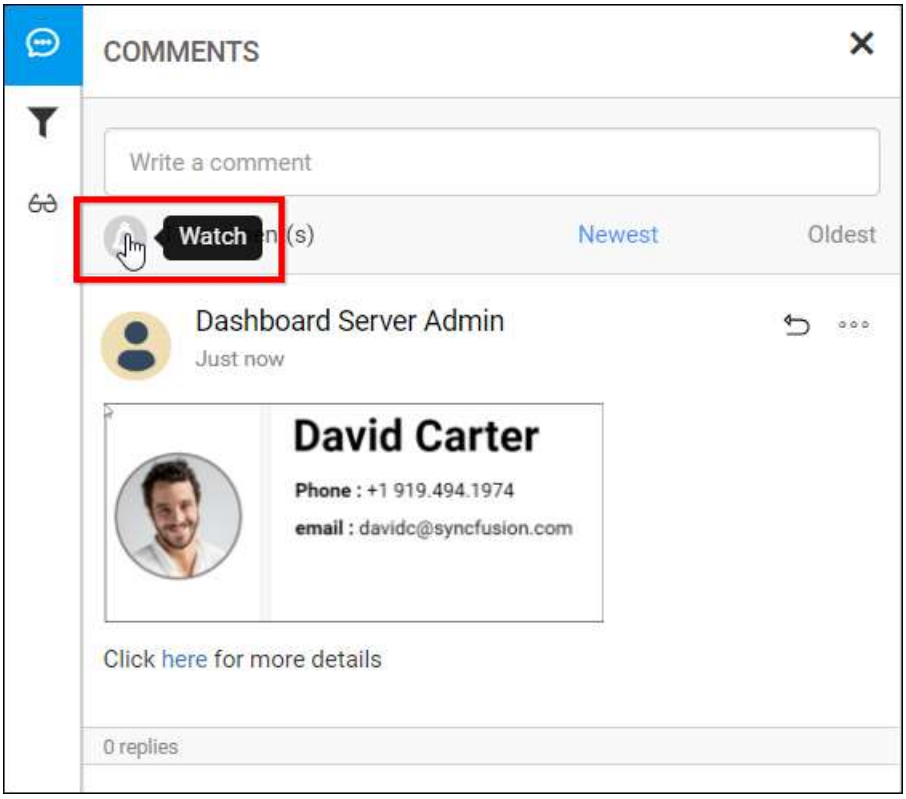
Watch and Unwatch comment

Using this option, users can watch and unwatch the notification of specific dashboard.

- If the watch comment of the dashboard is marked, user will get the notification from the specific dashboard.
- If the unwatch comment of the dashboard is marked, user will not get any notification from the specific dashboard.

You can change watch and unwatch comments of the specific dashboard using the following option.





Note: The above options - Post a new comment, reply to a comment, edit a comment, delete a comment and show parent comment of a reply applies to Widgets inside a Dashboard and to each Widget in the Widgets page (click here for more details about Dashboard and Widget commenting) too.

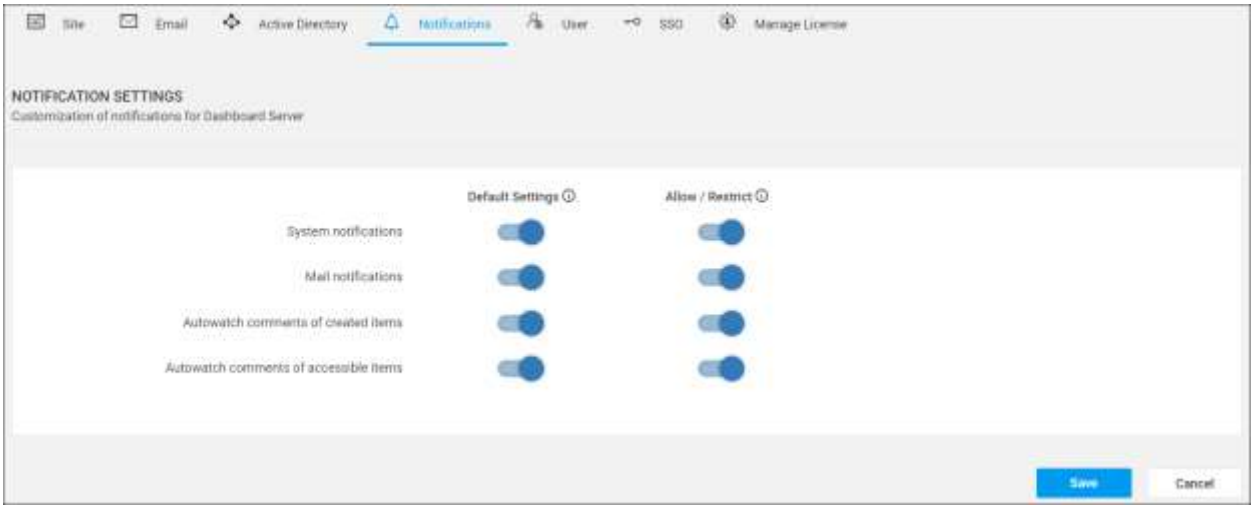
Notifications

This section explains on how to configure notifications to notify the users for comments on the Dashboards and Widgets in the Syncfusion Dashboard Server.

Notifications can be configured by both the System Administrator and the user.

Admin notification settings

Configure how the users receive notifications for the comments from the admin notification settings page.



Let's look at the notification settings one by one.

System notifications

System notifications are the ones that will appear in the top right before the user name once the user logged into the Dashboard Server as like in the below image.

![System notifications](images/notifications-system notifications.png)

Mail notifications

Users will also be notified through email for comments.

Autowatch comments of created items

This is a switch for watching the comments of the items that the users have created. Enabling this will send notifications for comments on all the items that the users have created.

Autowatch comments of accessible items

This is a switch for watching the comments of the items that the users have access. Enabling this will send notifications for comments on all the items that the users have access.

Now, let's look at the default and allowable notification settings configuration.

Default Settings

This is the default settings applied to the user while the user is added into the Dashboard Server. Users can change switch from this setting and make their own or choose to inherit this setting anytime in their profile edit page

Allow/Restrict Settings

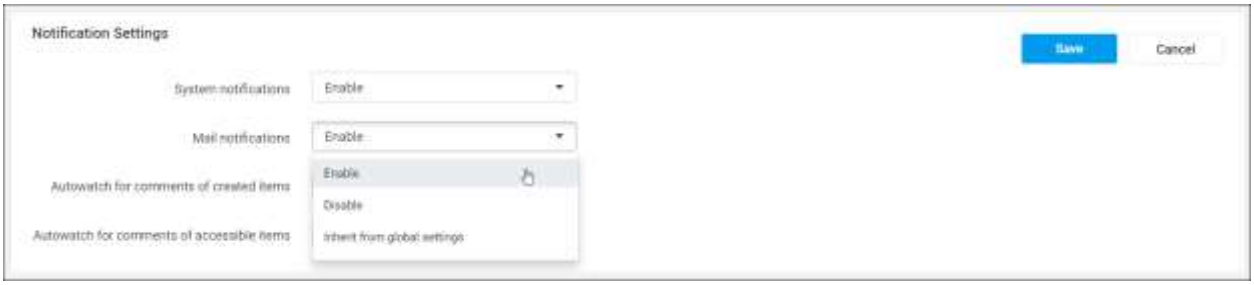
This is the master settings for the Dashboard Server. Upon enabling or disabling any setting in here will enable or disable it in the Dashboard Server. This will override the default and user settings.

User Notification Settings

Configure how the current user receive notifications for the comments from the user notification settings page.

Users can navigate to this page from the profile edit page as shown in the below image.

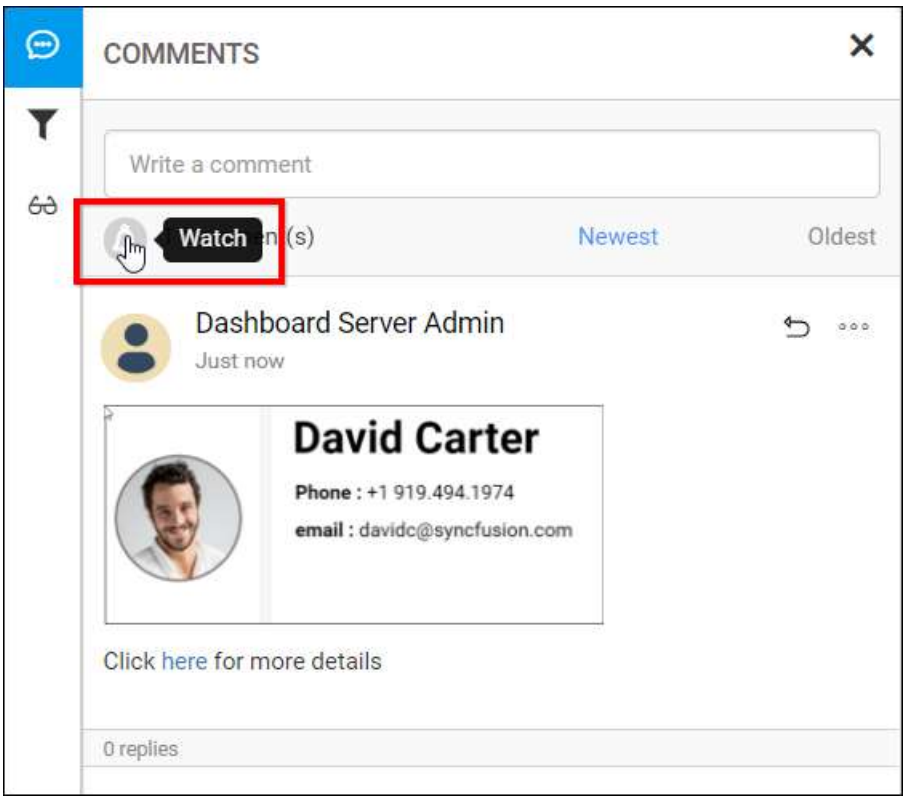
Refer to the below image for changing the notification settings for the current user.



Settings can be enabled or disabled or inherited from global settings which is the default settings of the Dashboard Server.

Specific watch

Apart from autowatch of created and accessible item settings, users can also watch on a specific item if they want to watch an item specifically.



Users can toggle between watch and unwatch for a Dashboard comment anytime.

Localization

Localization is the process of adapting a website into different linguistic and cultural contexts - involving much more than the simple translation of text.

Syncfusion Dashboard Server is released with localization support.

The default language is English “en-US”.

Read the below documentation on how to add new localizations and how to edit existing localizations.

Syncfusion Dashboard Server

How to add new localization

How to edit existing localizations

Syncfusion Dashboard Viewer

How to add new localization

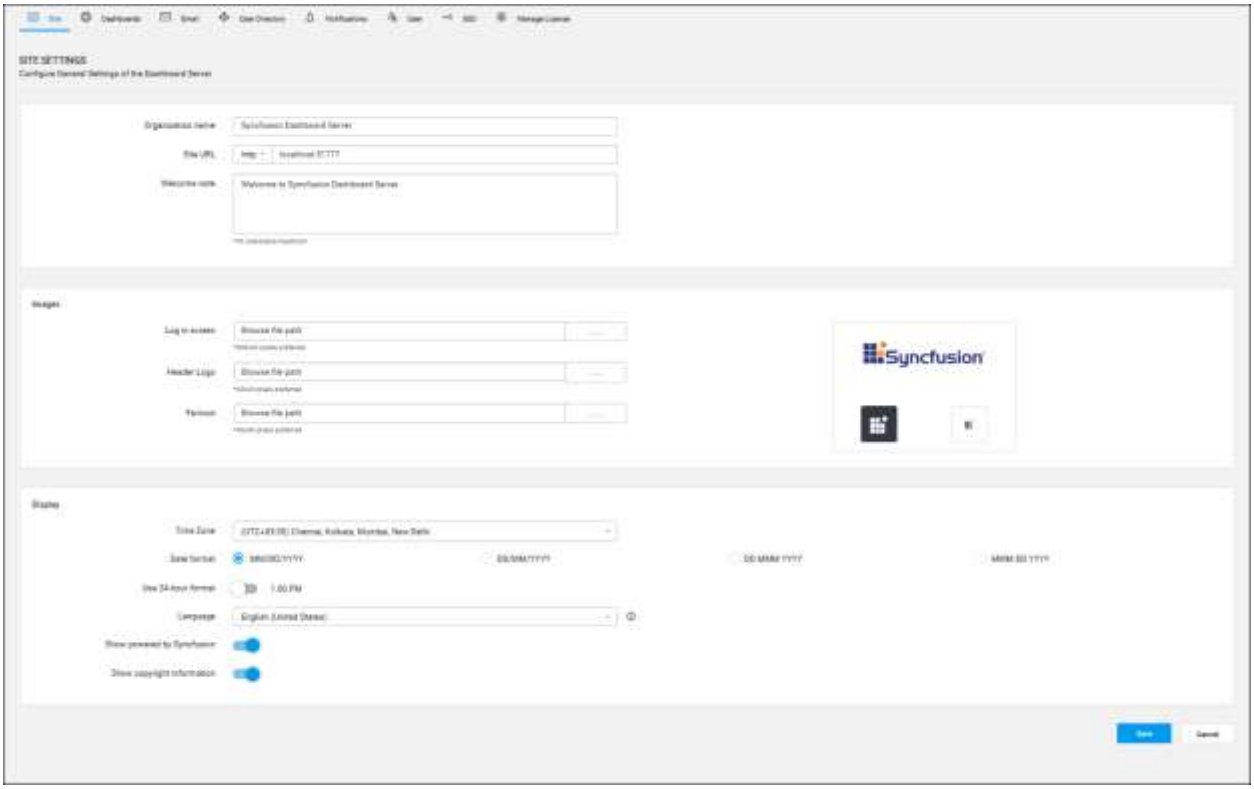
How to edit existing localizations

Site Settings

Custom Rebranding

This section explains on how to customize the Syncfusion Dashboard Server by changing the organization name, site URL, login screen logo and welcome note text, main screen logo, favorite icon and time zone and date time display formats.

Dashboard Server can be rebranded with Organization name, site URL, login screen logo and welcome text, main screen logo, favicon, time zone and date time format.

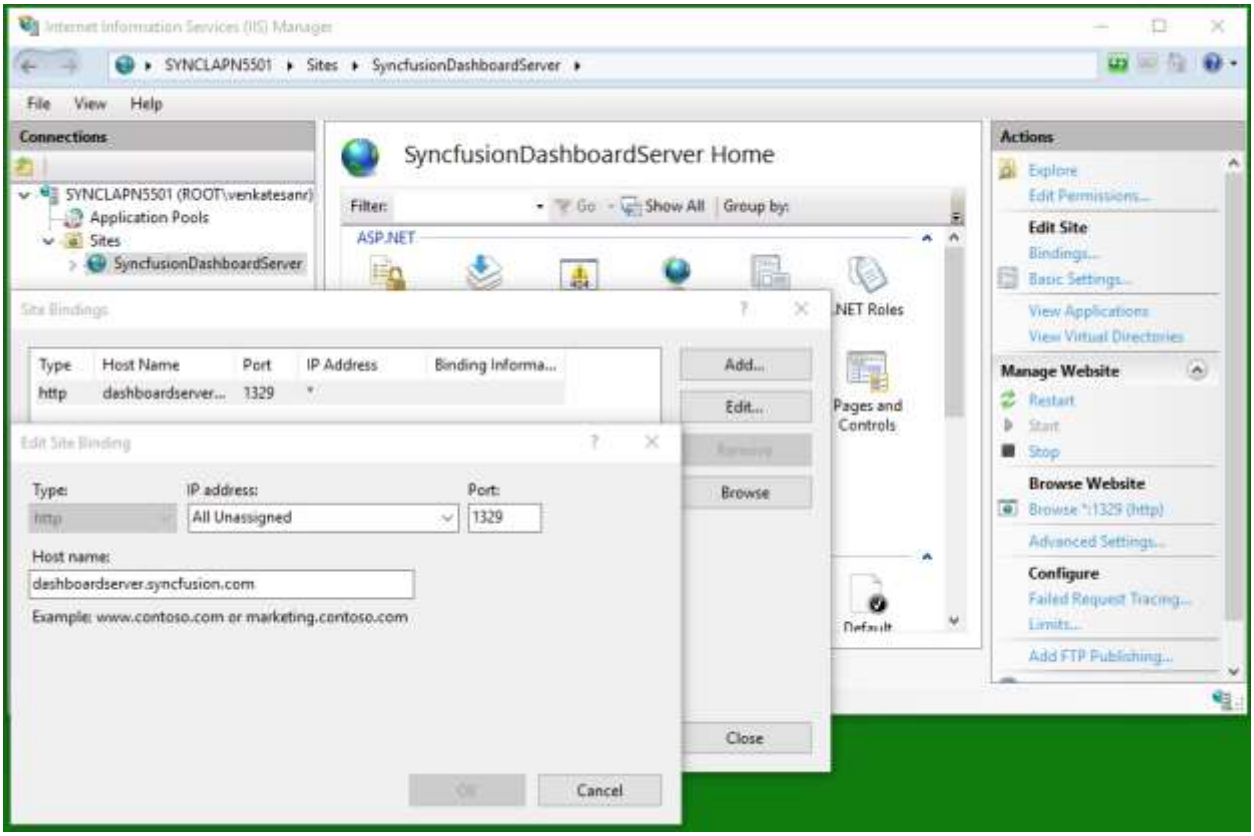


Organization Name

Name of the Dashboard Server can be changed at any time and this is in the title bar of the browser.

Site URL

Dashboard Server URL can be changed at any time in the Server Settings page and to get this change reflected you need to configure the same in the IIS. Check the Host in IIS section on how to host the dashboard server in IIS. After hosting is done, add the same URL in the site binding in IIS as like below.



Dashboard Server can also run under SSL for which you will need to select the https from the Site URL dropdown.

Login Screen

- Login page logo image can be changed and the preferred image size is 240x120 pixels. Dashboard Server will have Syncfusion logo as default login logo.
- Welcome note can be changed and the maximum characters is 70. Dashboard Server will have "Welcome to Syncfusion Dashboard Server" as default login welcome text.

Main Screen

Main screen logo image can be changed and the preferred image size is 40x40 pixels. Dashboard Server will have Syncfusion logo as default main screen logo.

Favicon

Favicon for the Dashboard Server can be changed and the preferred image size is 16x16 pixels. Dashboard Server will have Syncfusion favicon as default favicon.

Display

Time zone

Time zone for the Dashboard Server can be changed. Dashboard Server sets the time zone of the system where it is installed by default.

Date format

Date format of the Dashboard Server can be changed. Dashboard Server will have "MM/dd/yyyy" as the default date format.

Time format

Time format of the Dashboard Server can be changed. Dashboard Server will have "12 hour" as the default time format.

Language

Option to localize the Dashboard Server to any culture. Please click here to learn how to add new localization or edit existing localization in the Dashboard Server.

Powered by Syncfusion

Option to show/hide Powered by Syncfusion in the footer of the Dashboard Server. By default, this will be shown.

Copyright information

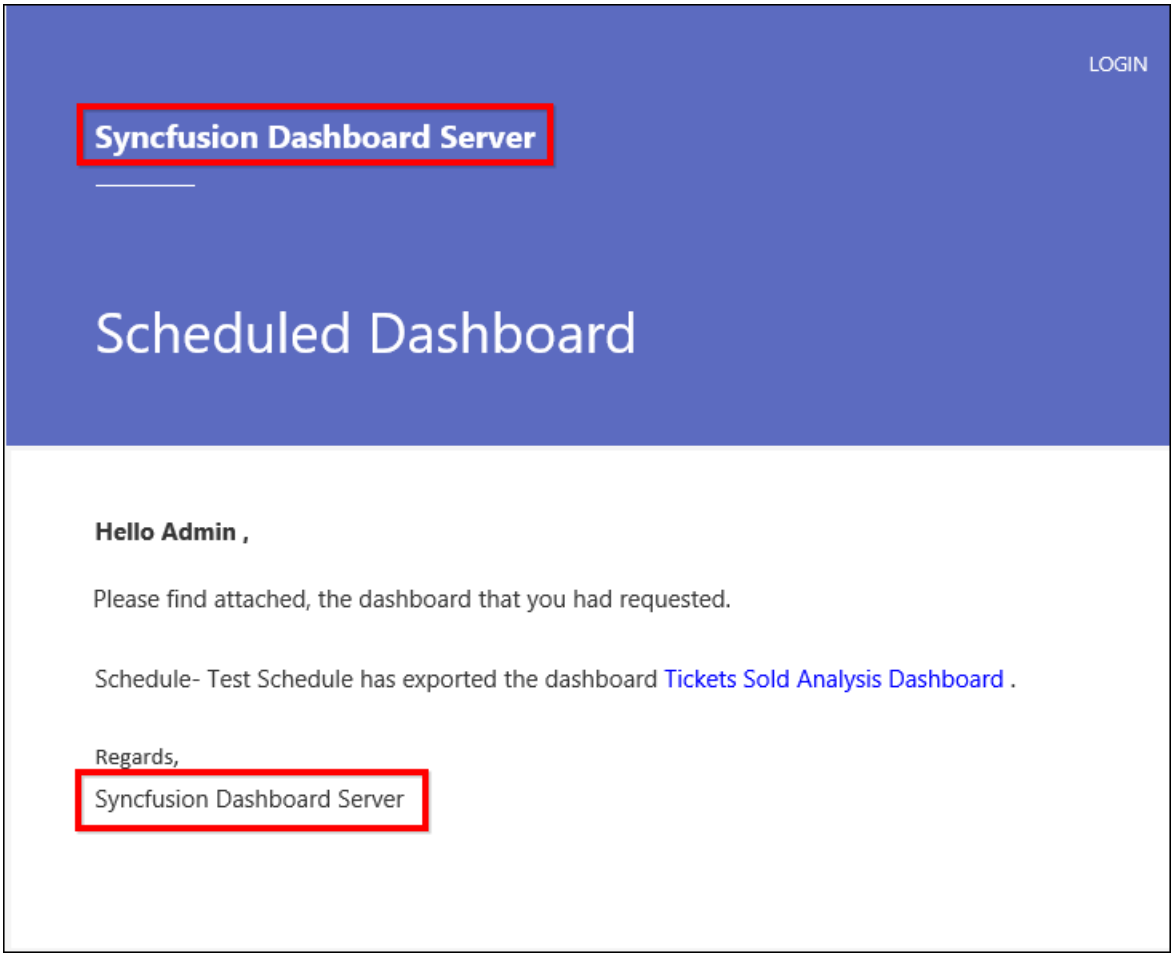
Allows to show/hide Copyright Information in the footer of the Dashboard Server. By default, this will be shown.

Email templates

Email templates define the mail content to be sent via email to the Dashboard Server users. Email templates can be customized by the following ways.

Organization name

Organization name can be changed in the site settings page of the Dashboard Server which will shown in the email template as below.



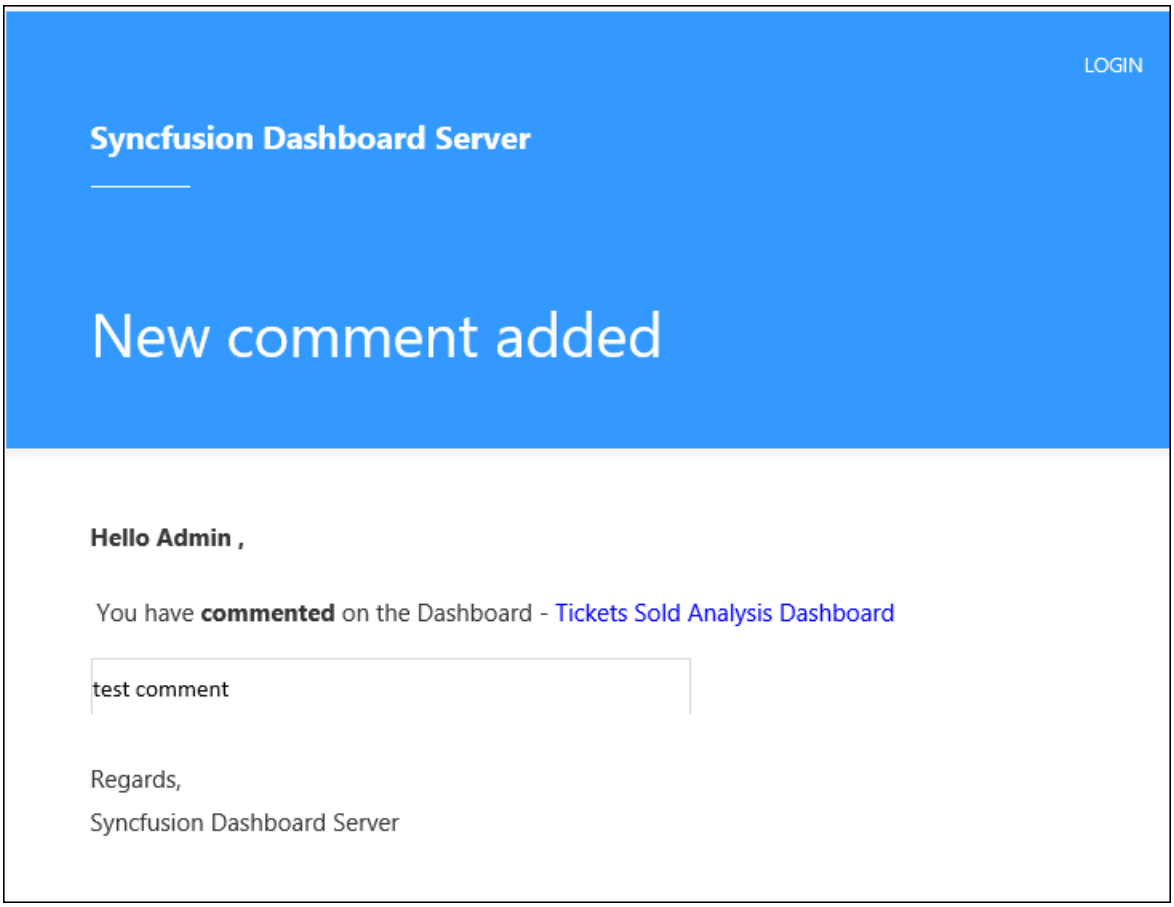
Mail content

The mail body content can be changed by editing the email templates provided in the following location:

{Installed_ Location}\Syncfusion\Dashboard Server\DashboardServer.Web\Content\EmailTemplates

Comments

Users can add/edit/reply/delete comments for dashboards/widgets. An email notification will be sent to users with comment details.



Delete comment notification

If the comment has been deleted, notification mail will be sent to the user with the deleted comment.

New comment notification

If a new comment is added to the dashboard/widget, mail will be sent to the user with the added comment.

Reply comment notification

If any user replies to the added comment, mail will be sent to the user who initiated the comment.

Update comment notification

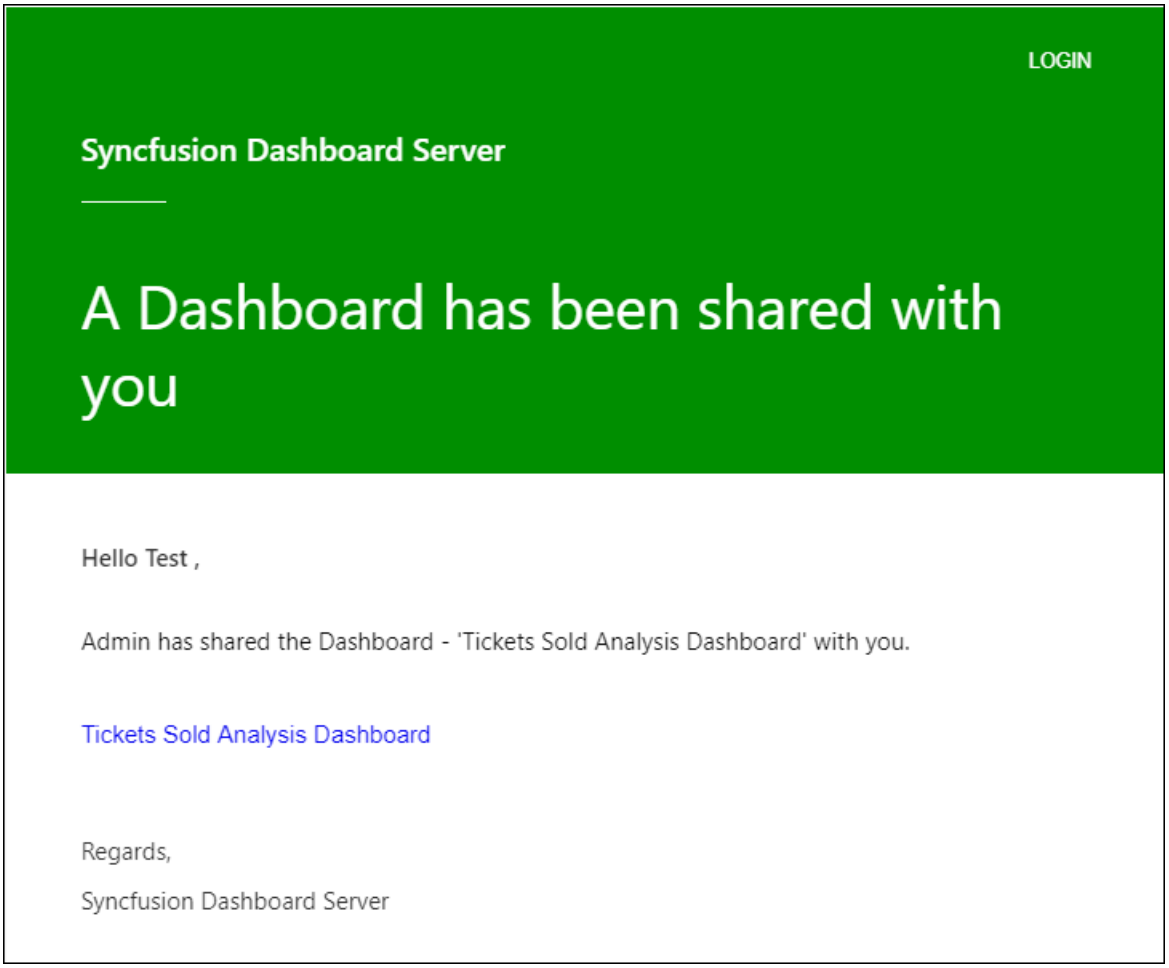
If the comment is modified, mail will be sent to users with the modified comment.

UserMention comment notification

You can mention any user in the comment then that user will be notified with the mail notification.

Item share

Item owners can share the dashboard/widget/datasource/view with other users. An email notification will be sent to users with shared item details.



All item share in category

When a user is granted permission to access all items in the specific category, the notification mail will be sent to the user with the shared item and category details.

All item share

When a user is granted permission to access all items (dashboard/widget/datasource), the notification will be sent to the user through mail.

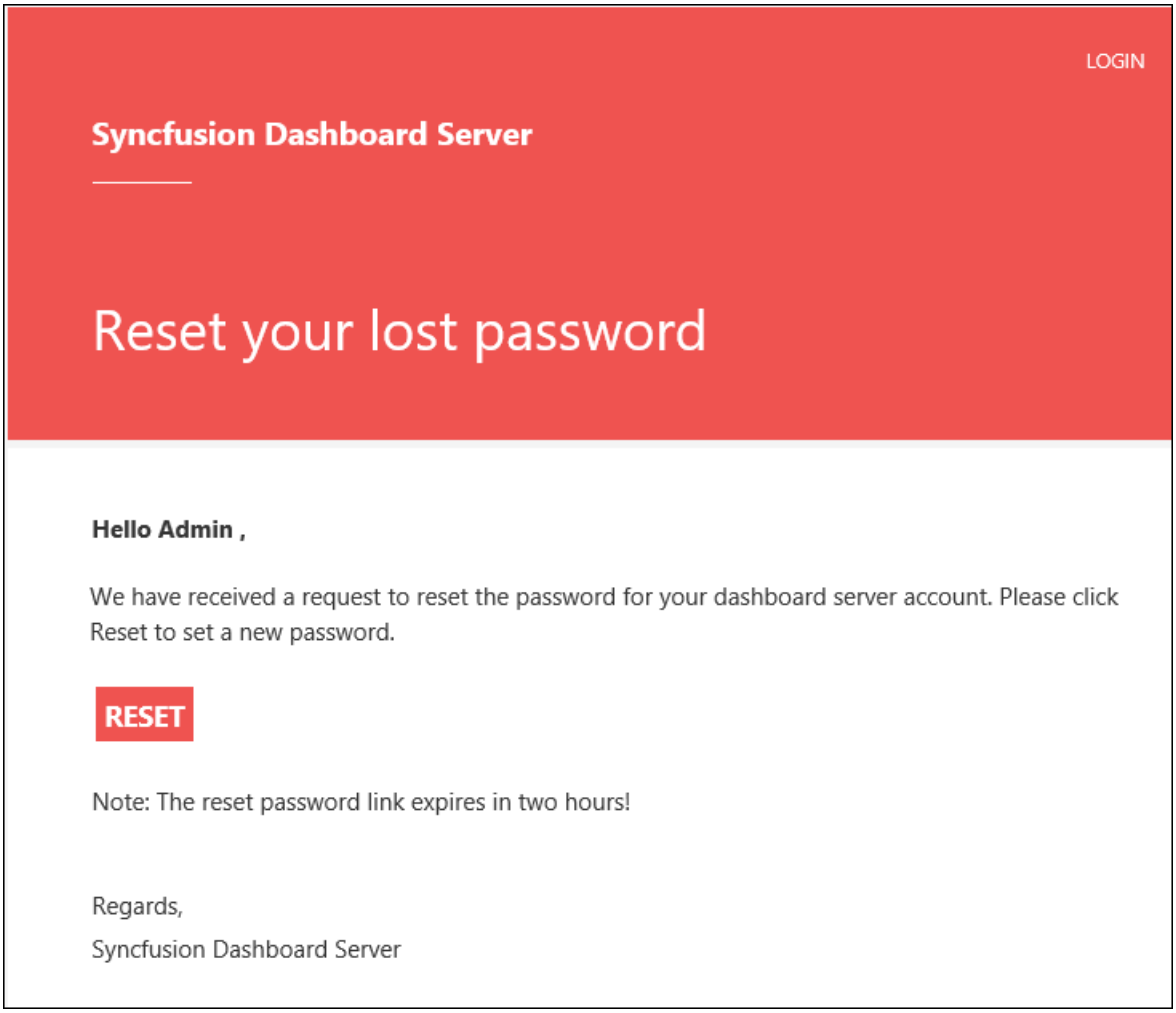
Specific item share

When the user is granted permission to access specific item (dashboard/widget/datasource/view), the notification mail will be sent to the user with the item detail.

Password

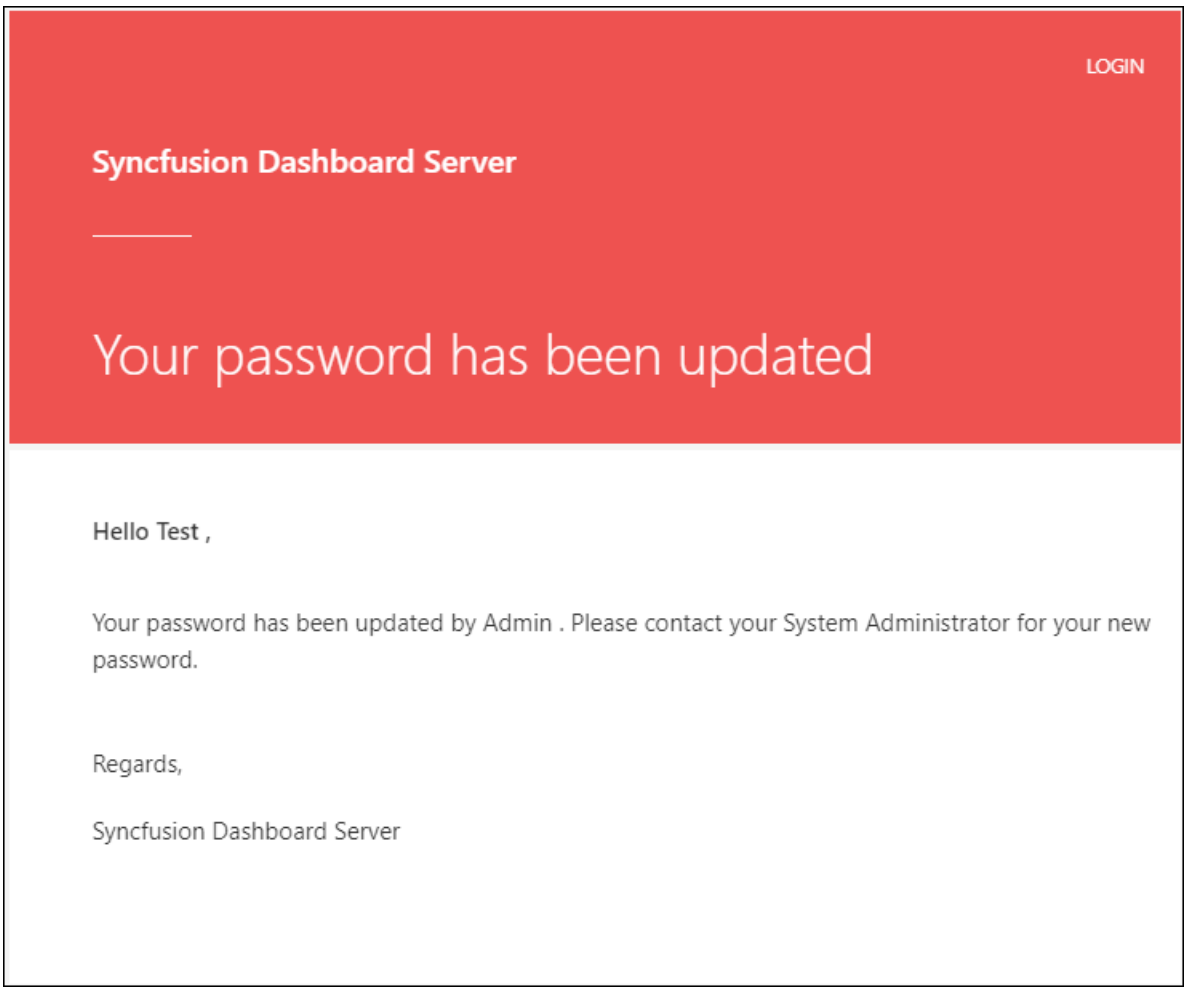
Forgot password

When a user uses the forgot password option to reset the password, the mail will be sent to the user with the reset password link.



Password update

When user/admin resets the password, the user/admin will be notified that the password has been changed.



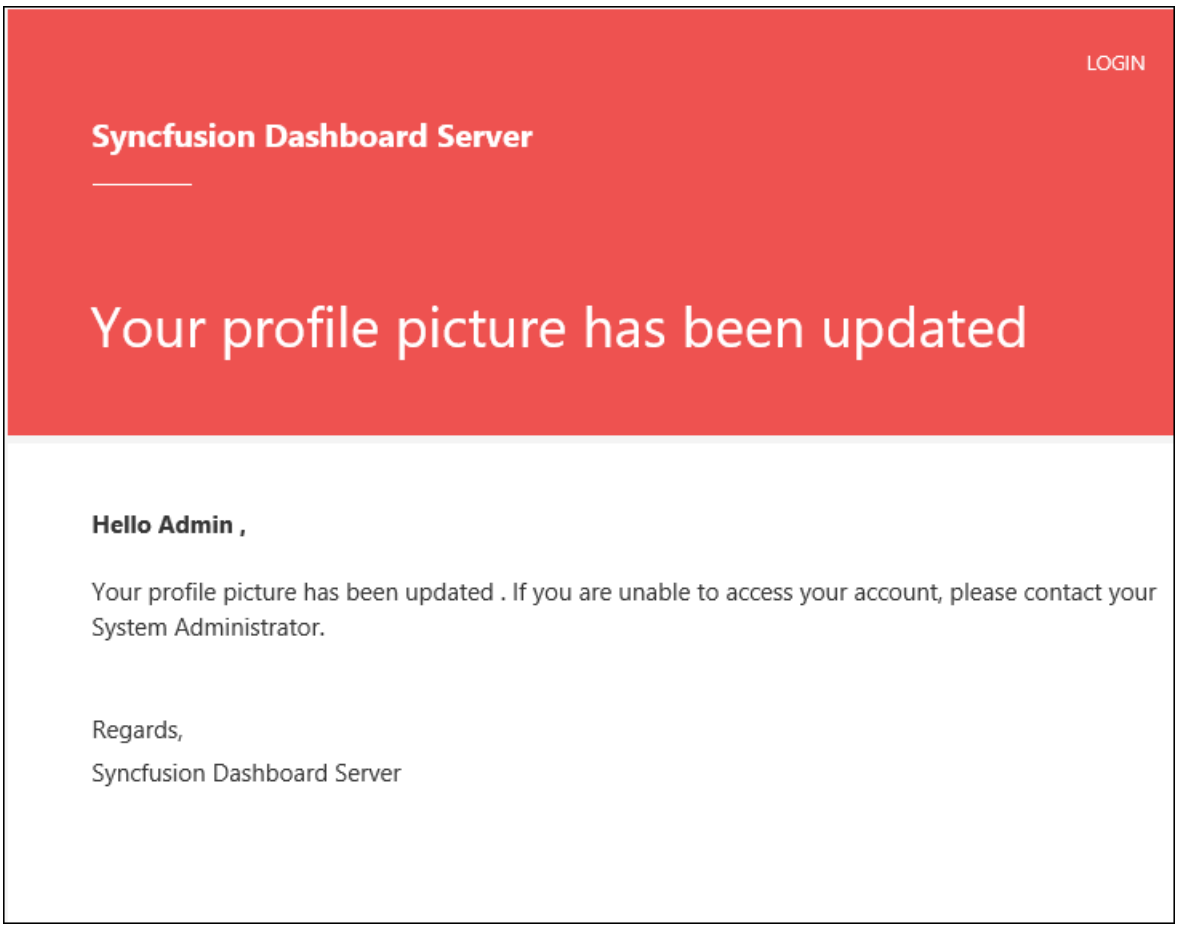
User profile

Profile update

When user/admin changes the user profile information, the user/admin will be notified with the changes made in the profile.

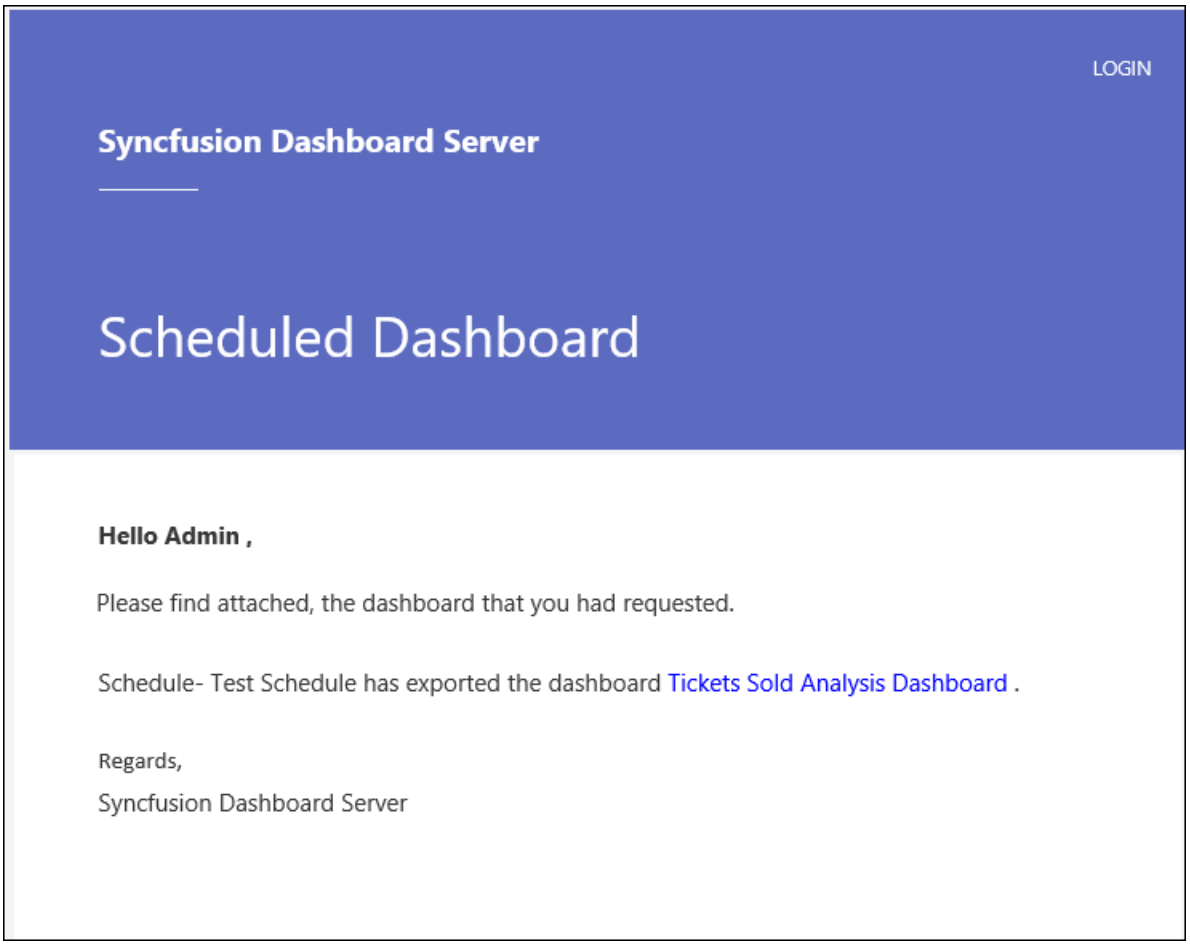
Profile picture update

When user/admin changes the profile picture, the user/admin will be notified that the profile picture has been changed.



Schedule

Users can schedule the dashboards and users/groups synchronization in Active directory/Azure Active directory/Database.



On demand schedule

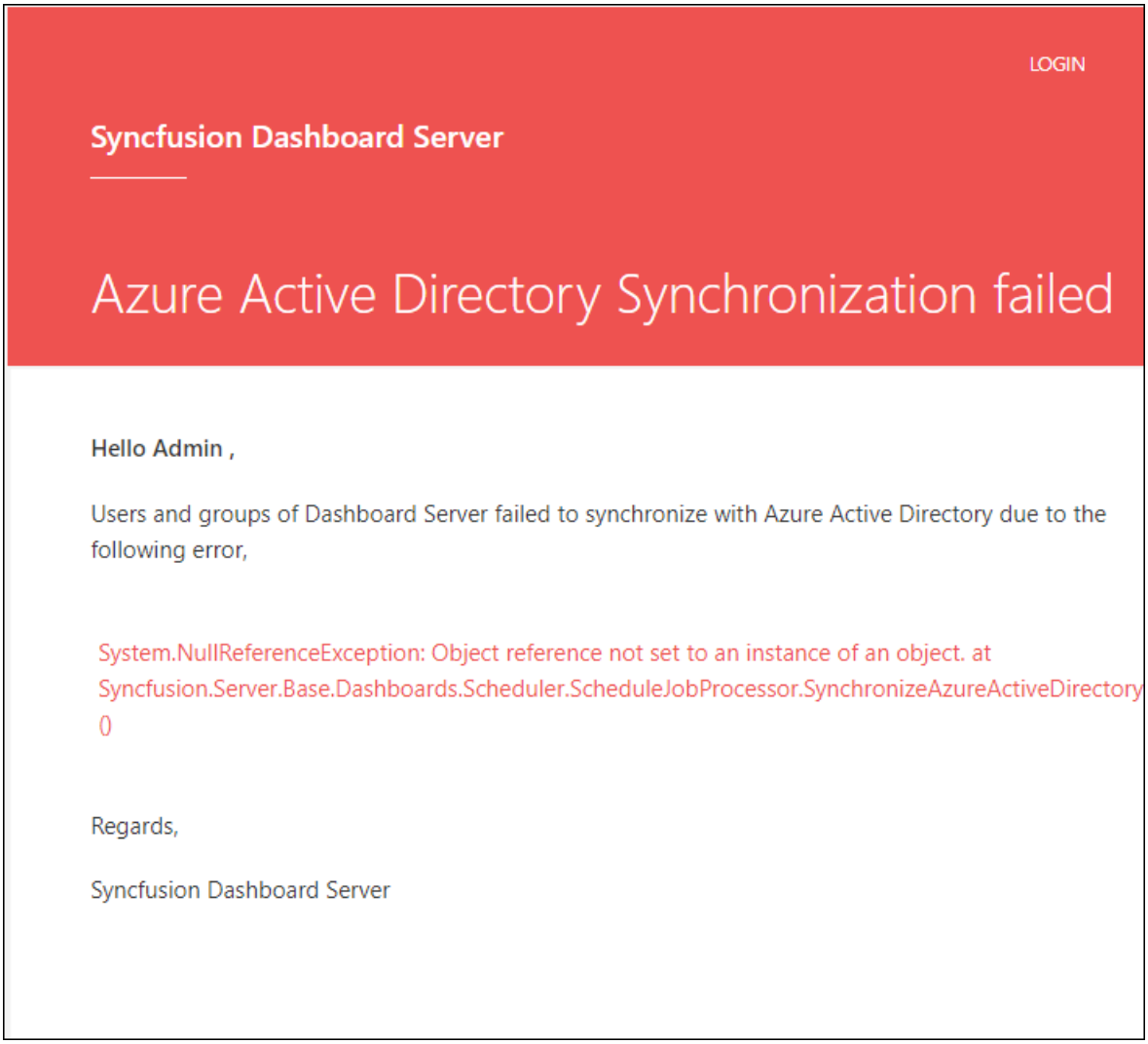
When user starts the schedule on demand, the dashboard will be sent to the user via mail.

Process schedule

Scheduled dashboards are sent via mail in user specified dashboard format (Image/PDF/Excel).

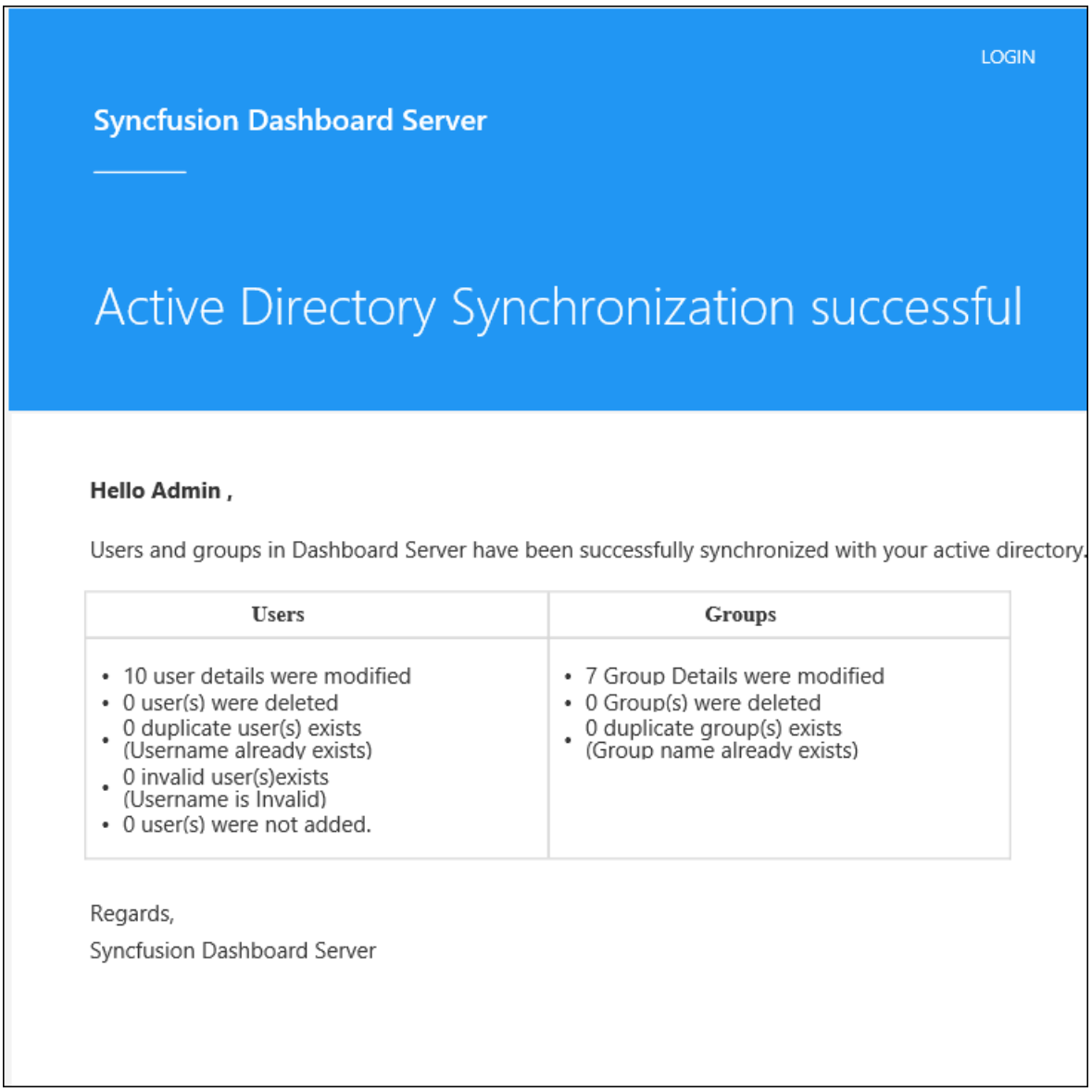
Synchronization failed

When Active directory/Azure Active directory/Database synchronization is failed, the user will be notified with error details.



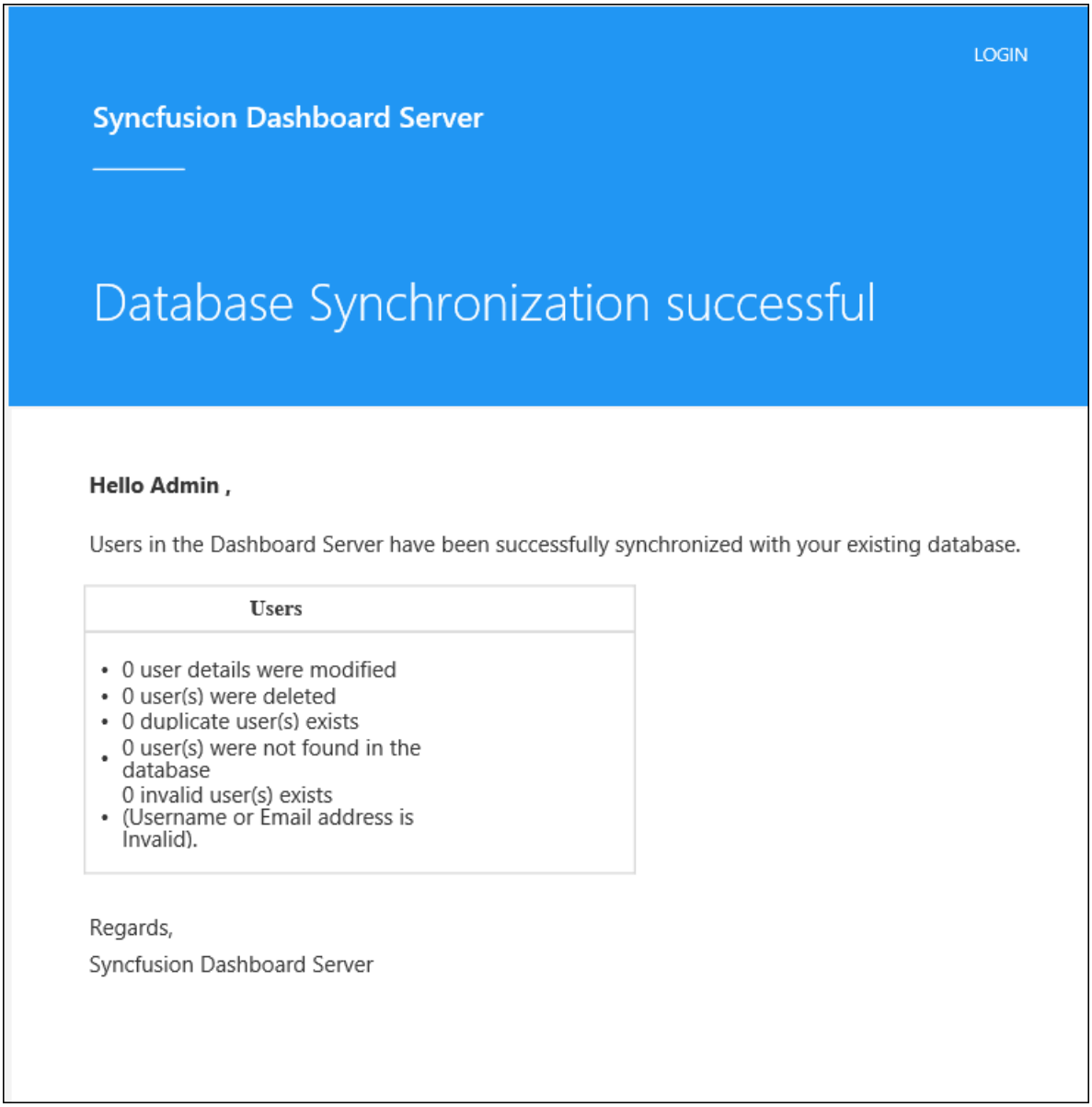
Synchronized Active Directory

On successful Active Directory/Azure Active Directory synchronization, user will be notified with the user count.



Synchronized user import

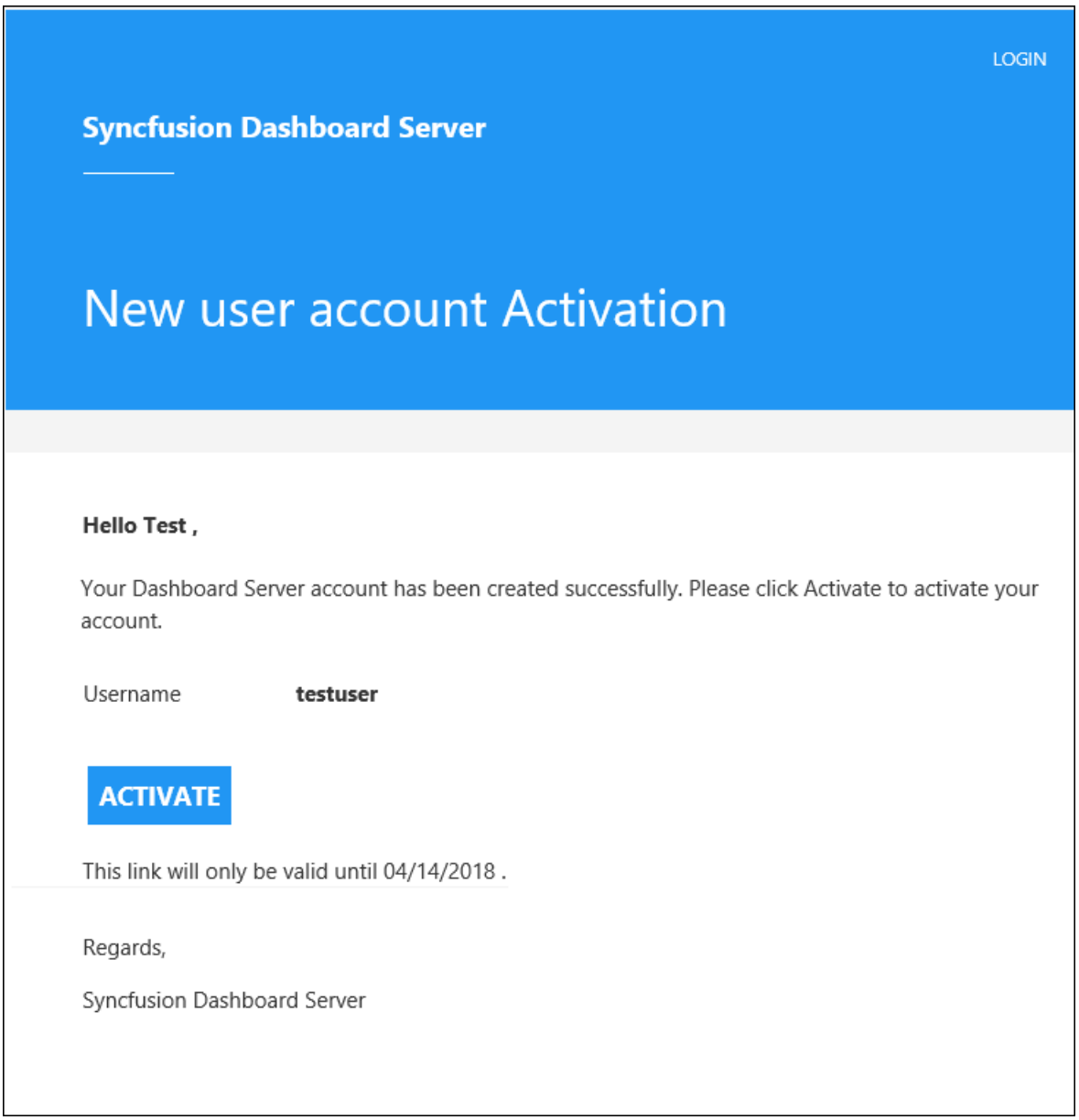
On successful user synchronization of the database, the user will be notified with the user count.



Users

User activation

When a new user is added to the Dashboard Server in Email Activation mode, an activation mail will be sent to the user with the Activation link.



Email Settings

This section explains on how to configure the SMTP details in the Syncfusion Dashboard Server to send emails.

SMTP Email Settings are required to perform the following operations

Account Activation --- Sends user account activation email

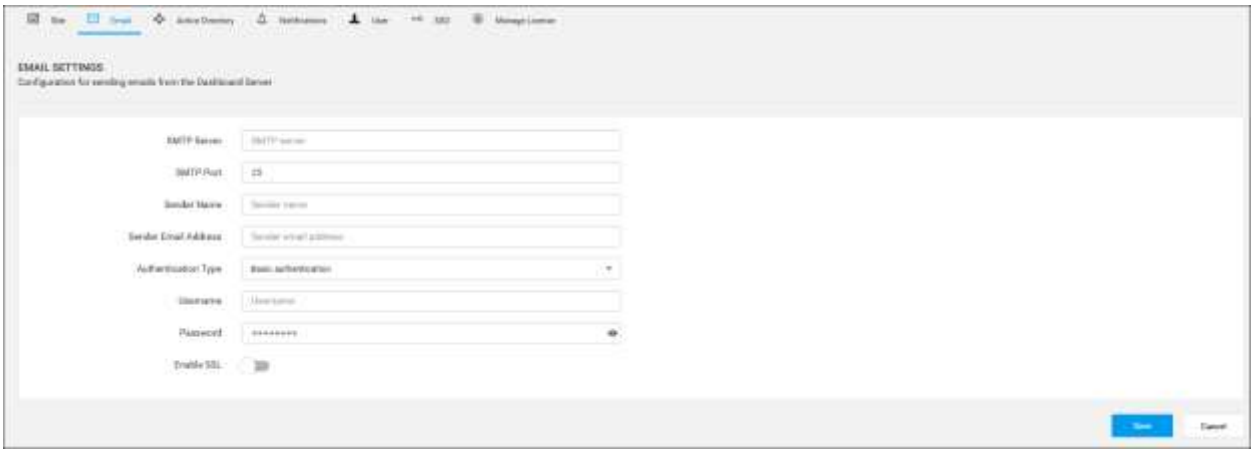
Forgot Password --- Sends request links to reset the password when the user has forgotten the password

Reset Password --- Sends links to reset the password

Scheduled Dashboards --- Sends the exported dashboard to the scheduled recipients

The following SMTP details are required to send email from the dashboard server.

- SMTP Server
- SMTP Port
- Sender Name
- Sender email
- Password
- SSL/TLS



Mark Dashboards and Widgets as Public - Allow/Restrict Switch

This section explains about how to allow/restrict marking the Dashboards/Widgets as public from the administration dashboards settings page.

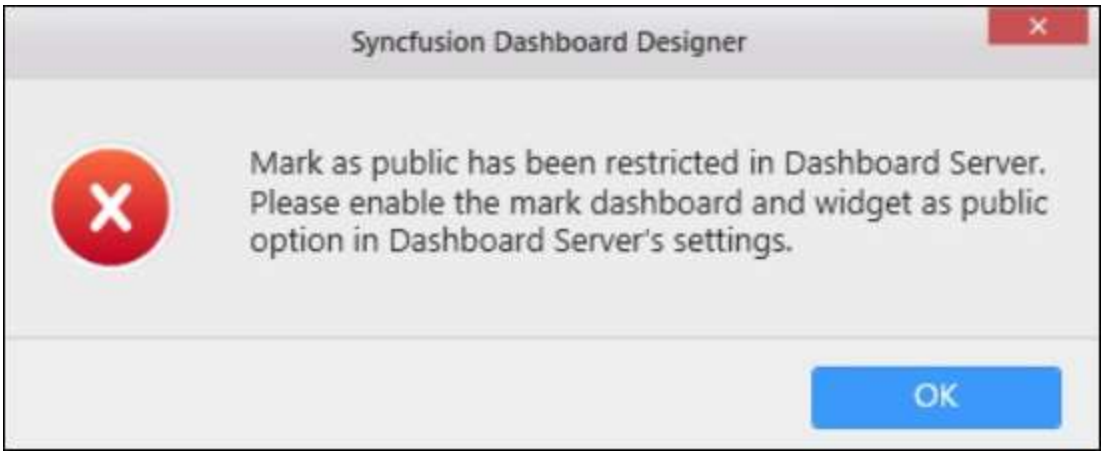
You can control the user to make the Dashboards/Widgets as public through the Dashboards settings tab in settings page. The user can able to mark the Dashboards/Widgets as public, if the administrator allows the Mark dashboards and widgets as public.

Take control over the public Dashboards/Widgets

If the administrator has allowed, then the user can able to mark the Dashboards/Widgets as public and other users can able to see the public Dashboards/Widgets.

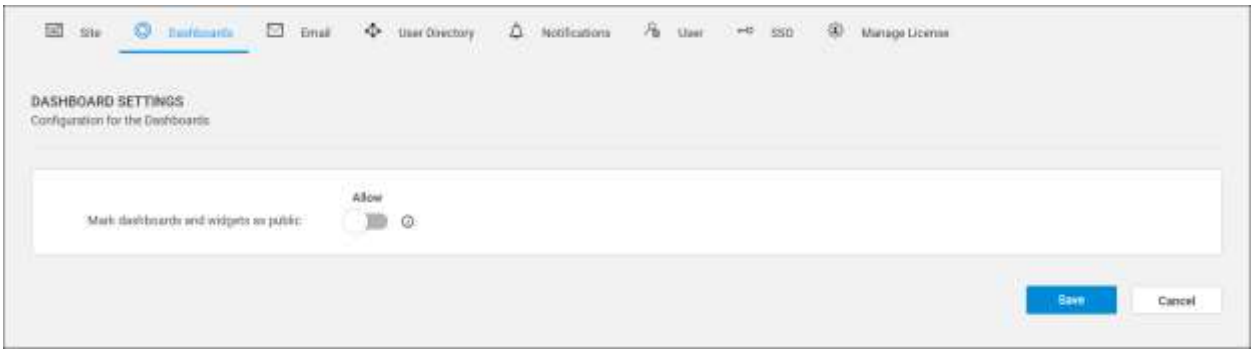
If the administrator has restricted, the public Dashboards/Widgets cannot be rendered by the user until they hold the permission for the Dashboards/Widgets.

When the user tries to publish the dashboard from the designer by enabling Make public, the Dashboard Server will allow it if the administrator has allowed the Mark Dashboards and Widgets as public in the Dashboards tab of the settings page. Otherwise, the error message will show as follows:

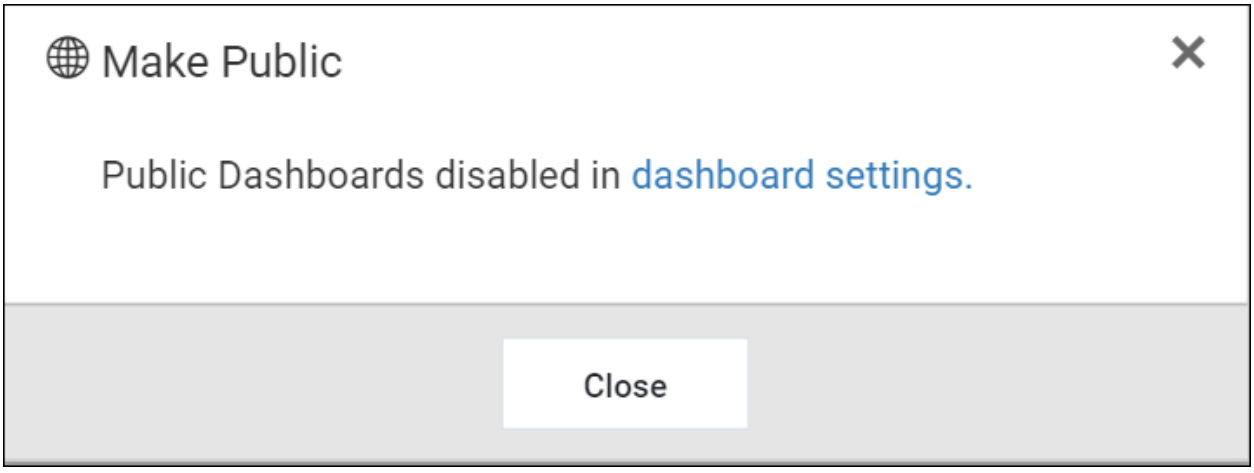


Mark dashboards and widgets as public switch to allow/restrict public Dashboards/Widgets

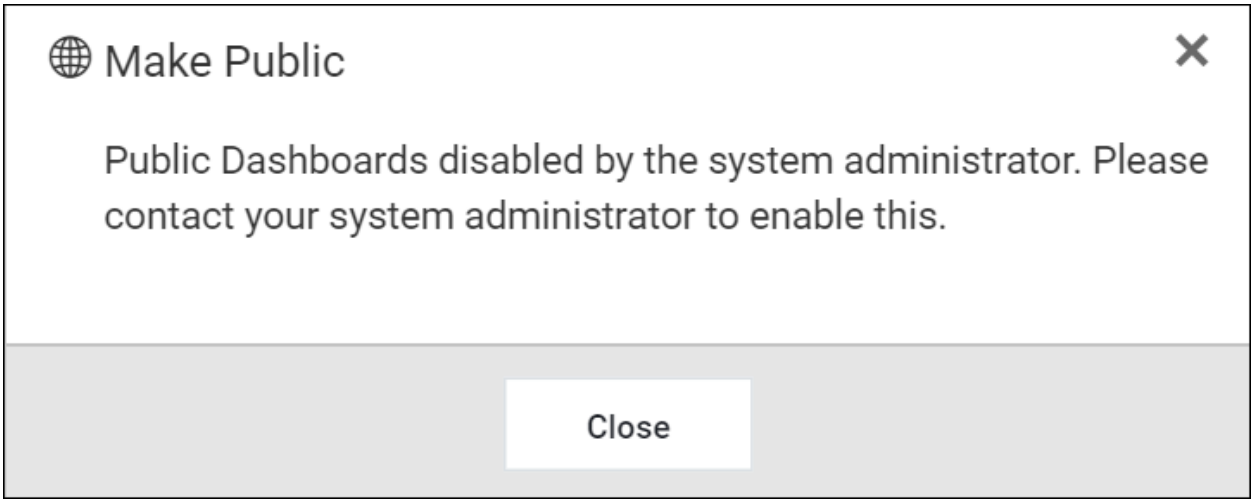
After restricting the Mark dashboards and widgets as public in dashboards settings page, the following image will be displayed.



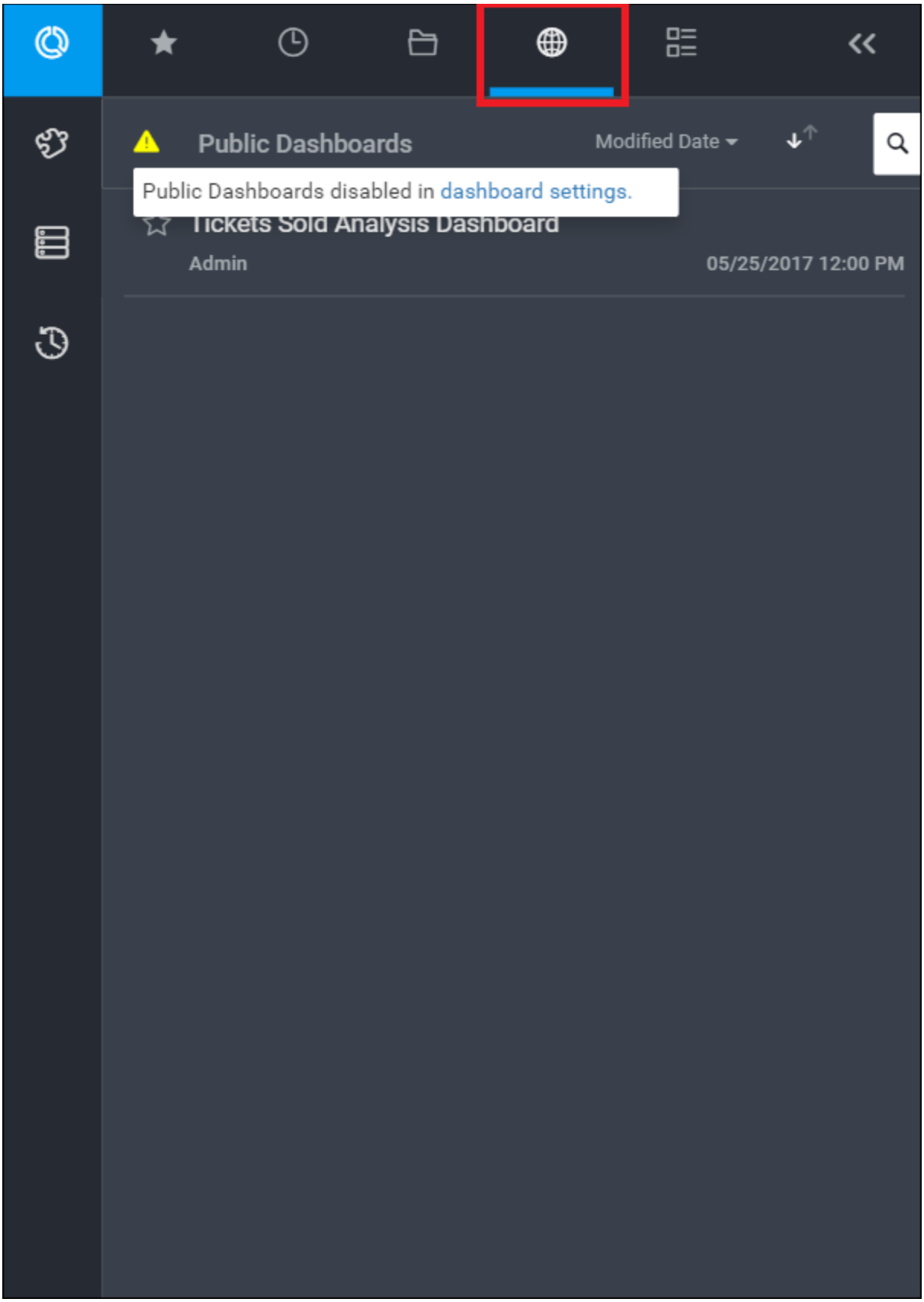
When you click the Make Public from the context menu of the respective Dashboards/Widgets, the following message will be shown to the administrator.



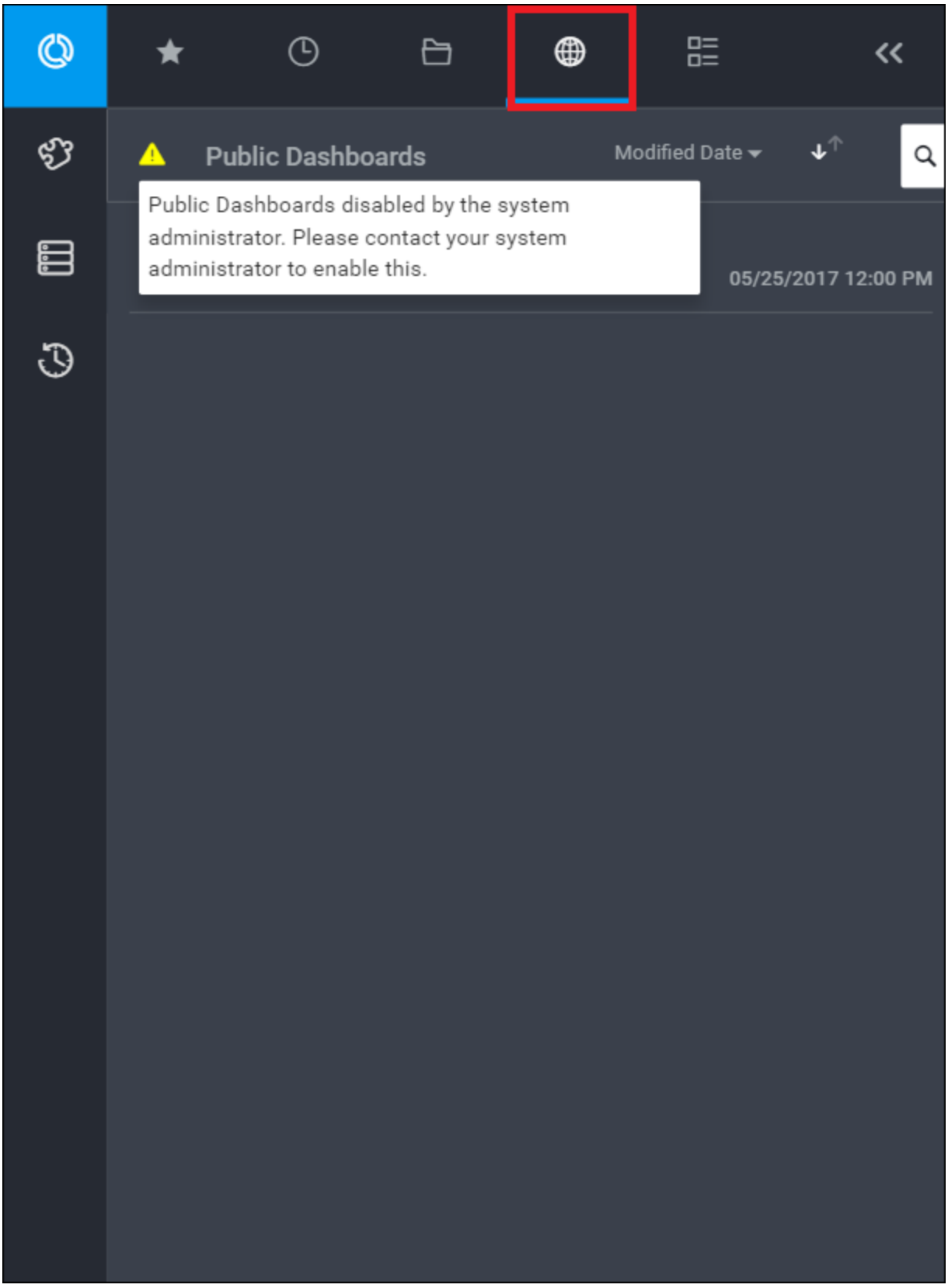
When you click the Make Public from the context menu of the respective Dashboards/Widgets, the following message will be shown to the user.



After restricting the Mark dashboards and widgets as public, if you click the Public Dashboards or Public Widgets tab the following message will be shown to the administrator.



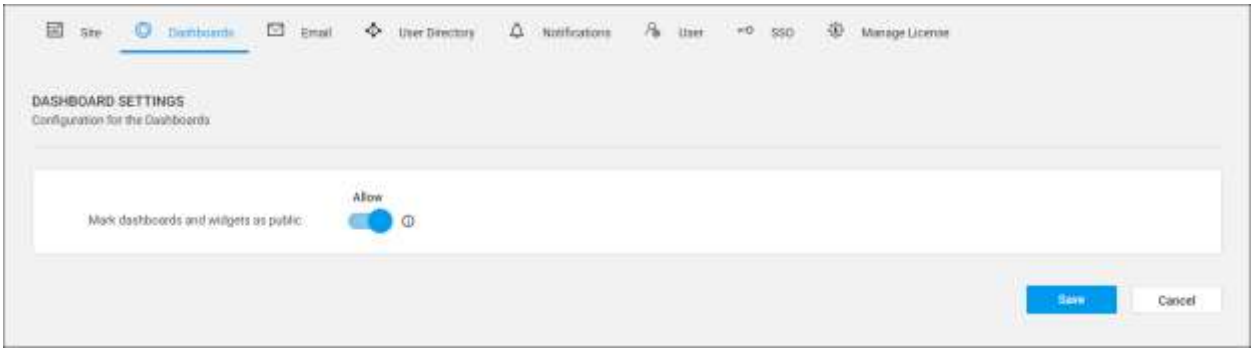
After restricting the Mark dashboards and widgets as public, if you click the Public Dashboards or Public widgets tab the following message will be shown to the user.



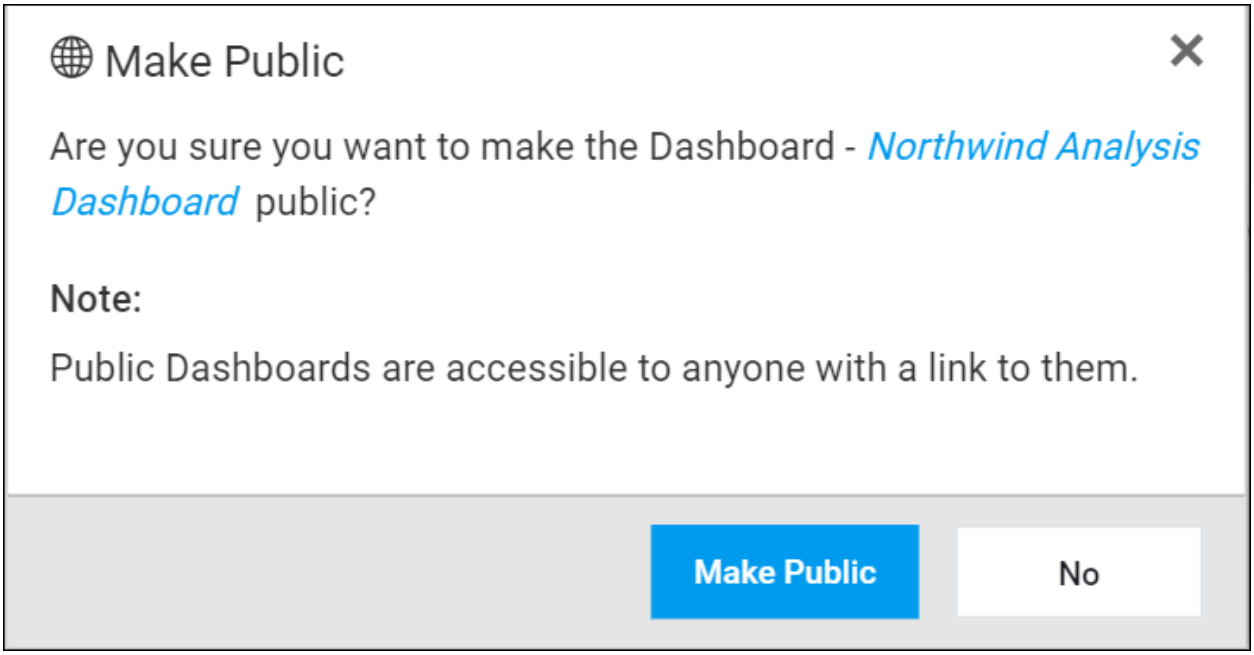
Make public

Follow the below steps to make the Dashboards/Widgets accessible to anonymous users.

1.Allow the Mark dashboards and widgets as public in dashboards settings page.



2.Click the Make Public from the context menu of the respective Dashboard/Widget.



3.Click the Make public in the confirmation dialog box to mark Dashboards/Widgets as public.

Note: Click here to get more details about public dashboards.

DataStore Settings

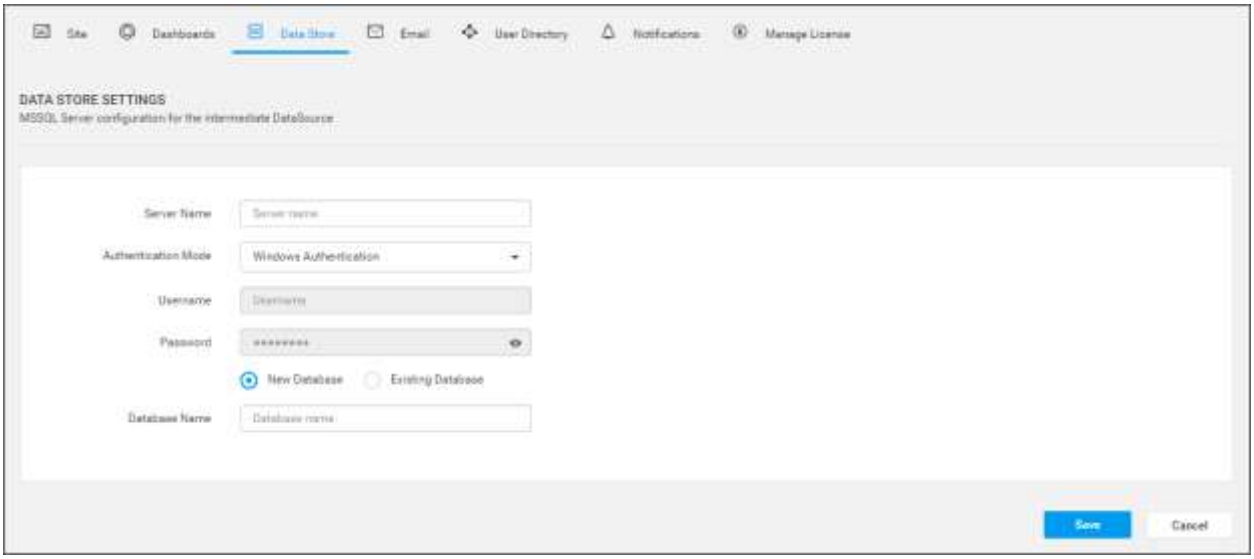
This section explains on how to configure the Data store settings for storing extracted data from data sources.

DataStore is an intermediate database that is used to store imported data from web data sources and statistic files, there you will have a scheduled refresh option to update data.

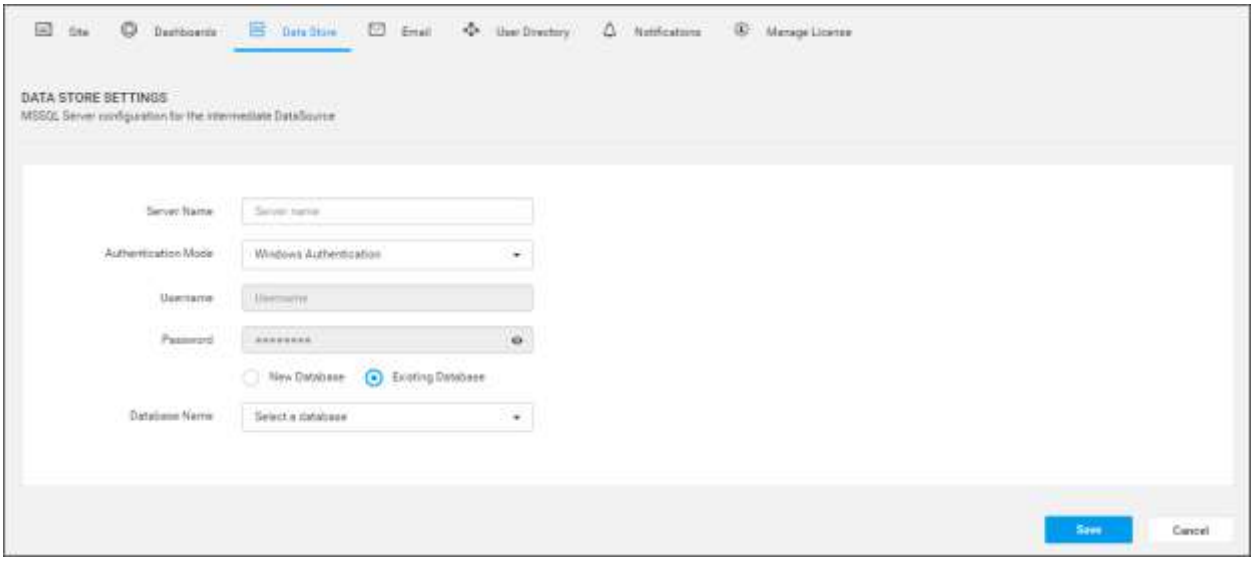
Database configuration

We can connect to the existing SQL Server instance with the below actions

- Create new database for intermediate database.



- Use an existing database for intermediate database.
- Choose one of the database from Select a Database drop down for creating Data Store tables in that database.



Note: The credentials that is given to connect to the SQL Server instance must have permissions to

- Create Database
- Create Table
- Insert
- Update Table
- Alter Table
- Select
- Drop Table
- Drop Database

User Profile

This section explains on how to view and edit profile, how to view my permissions in the profile and also on how to change password for the user profile in the Syncfusion Dashboard Server.

User can view the profile and edit the profile details, my permission details and can change the password.

View Profile

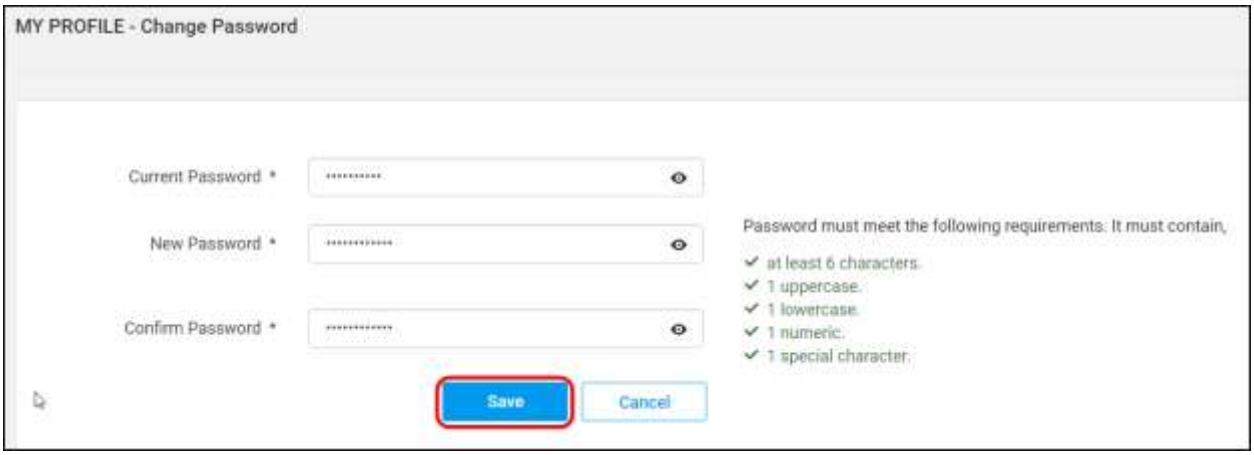
You can view your profile details in this page.

Edit Profile

You can change the first name, last name, email address, phone number, profile picture, and default homepage in the edit profile page.

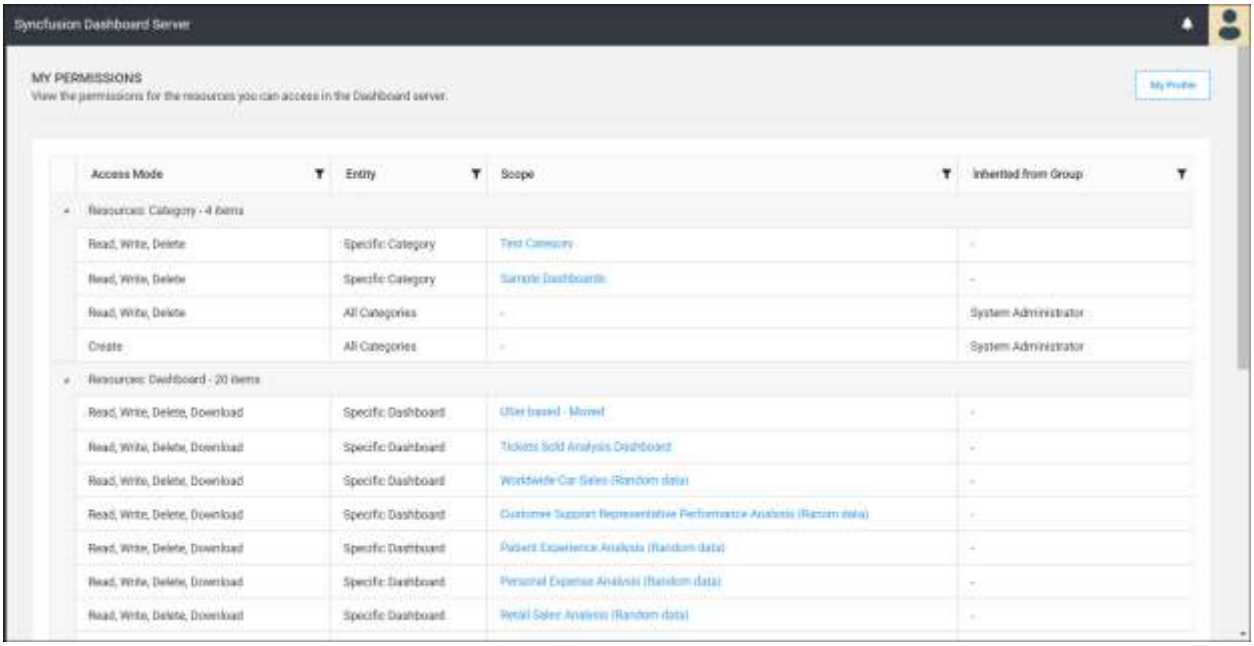
Change Password

Password to log in to the dashboard server can be changed in the change password page.



My Permissions

Users can view their access permission list for each resources like Dashboards, Data Sources, Widgets and Schedules in the Dashboard Server.



Utilities

Database Backup

This section explains on how to back up your resources and databases of Dashboard Server.

Note: This utility cannot be worked outside of the native folder.

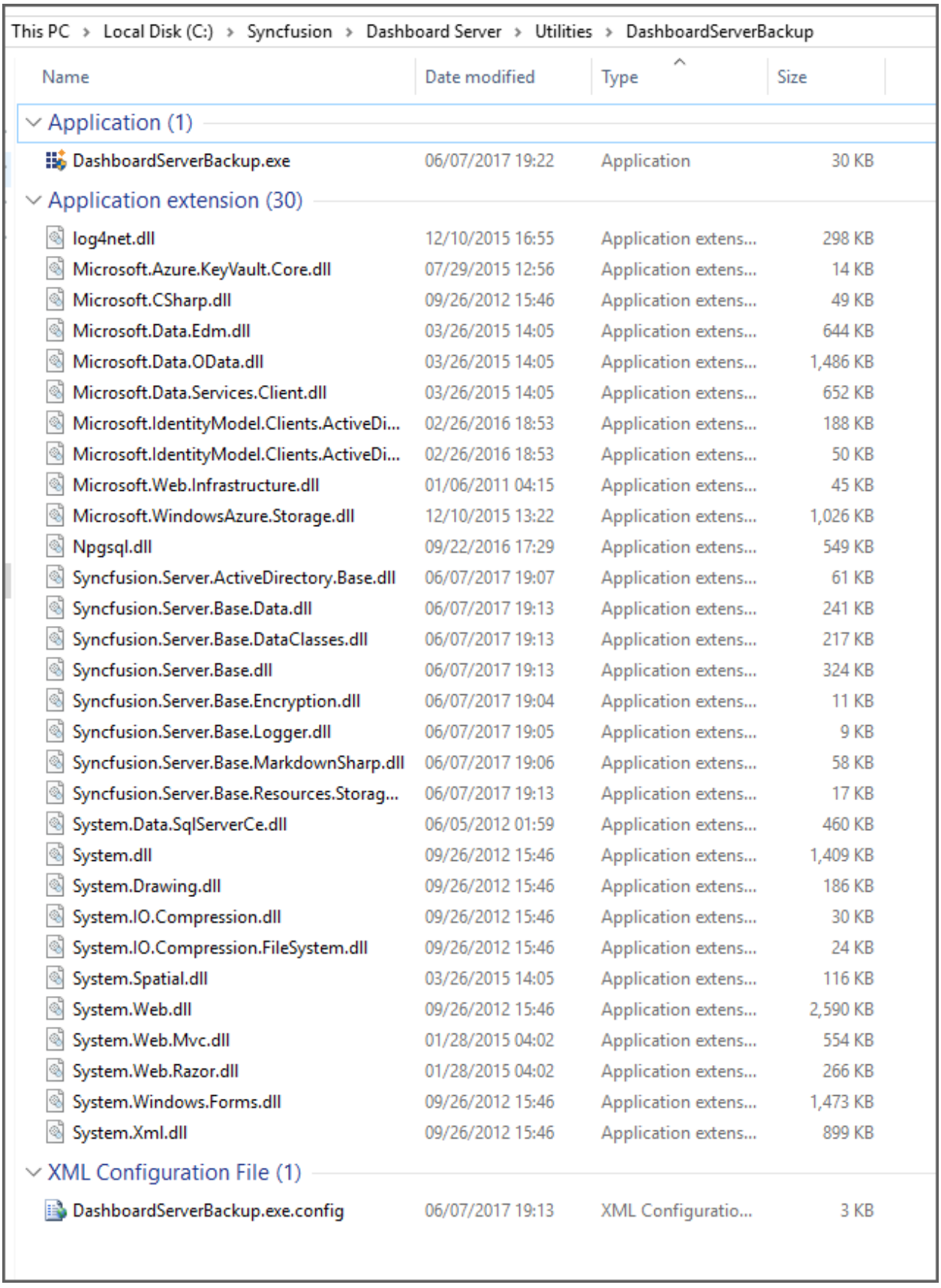
Dashboard Server is deployed in the below location by default.

`{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\`

For example, C:\Syncfusion\Dashboard Server\DashboardServer.Web\

We have shipped a utility with the Syncfusion Dashboard Server application in the below location by default.

`{Windows_Drive}\Syncfusion\Dashboard Server\Utilities\DashboardServerBackup`



This PC › Local Disk (C:) › Syncfusion › Dashboard Server › Utilities › DashboardServerBackup

Name	Date modified	Type	Size
Application (1)			
DashboardServerBackup.exe	06/07/2017 19:22	Application	30 KB
Application extension (30)			
log4net.dll	12/10/2015 16:55	Application extens...	298 KB
Microsoft.Azure.KeyVault.Core.dll	07/29/2015 12:56	Application extens...	14 KB
Microsoft.CSharp.dll	09/26/2012 15:46	Application extens...	49 KB
Microsoft.Data.Edm.dll	03/26/2015 14:05	Application extens...	644 KB
Microsoft.Data.OData.dll	03/26/2015 14:05	Application extens...	1,486 KB
Microsoft.Data.Services.Client.dll	03/26/2015 14:05	Application extens...	652 KB
Microsoft.IdentityModel.Clients.ActiveDi...	02/26/2016 18:53	Application extens...	188 KB
Microsoft.IdentityModel.Clients.ActiveDi...	02/26/2016 18:53	Application extens...	50 KB
Microsoft.Web.Infrastructure.dll	01/06/2011 04:15	Application extens...	45 KB
Microsoft.WindowsAzure.Storage.dll	12/10/2015 13:22	Application extens...	1,026 KB
Npgsql.dll	09/22/2016 17:29	Application extens...	549 KB
Syncfusion.Server.ActiveDirectory.Base.dll	06/07/2017 19:07	Application extens...	61 KB
Syncfusion.Server.Base.Data.dll	06/07/2017 19:13	Application extens...	241 KB
Syncfusion.Server.Base.DataClasses.dll	06/07/2017 19:13	Application extens...	217 KB
Syncfusion.Server.Base.dll	06/07/2017 19:13	Application extens...	324 KB
Syncfusion.Server.Base.Encryption.dll	06/07/2017 19:04	Application extens...	11 KB
Syncfusion.Server.Base.Logger.dll	06/07/2017 19:05	Application extens...	9 KB
Syncfusion.Server.Base.MarkdownSharp.dll	06/07/2017 19:06	Application extens...	58 KB
Syncfusion.Server.Base.Resources.Storag...	06/07/2017 19:13	Application extens...	17 KB
System.Data.SqlServerCe.dll	06/05/2012 01:59	Application extens...	460 KB
System.dll	09/26/2012 15:46	Application extens...	1,409 KB
System.Drawing.dll	09/26/2012 15:46	Application extens...	186 KB
System.IO.Compression.dll	09/26/2012 15:46	Application extens...	30 KB
System.IO.Compression.FileSystem.dll	09/26/2012 15:46	Application extens...	24 KB
System.Spatial.dll	03/26/2015 14:05	Application extens...	116 KB
System.Web.dll	09/26/2012 15:46	Application extens...	2,590 KB
System.Web.Mvc.dll	01/28/2015 04:02	Application extens...	554 KB
System.Web.Razor.dll	01/28/2015 04:02	Application extens...	266 KB
System.Windows.Forms.dll	09/26/2012 15:46	Application extens...	1,473 KB
System.Xml.dll	09/26/2012 15:46	Application extens...	899 KB
XML Configuration File (1)			
DashboardServerBackup.exe.config	06/07/2017 19:13	XML Configuratio...	3 KB

This utility can take both the resources and database backups for the below types of Databases.

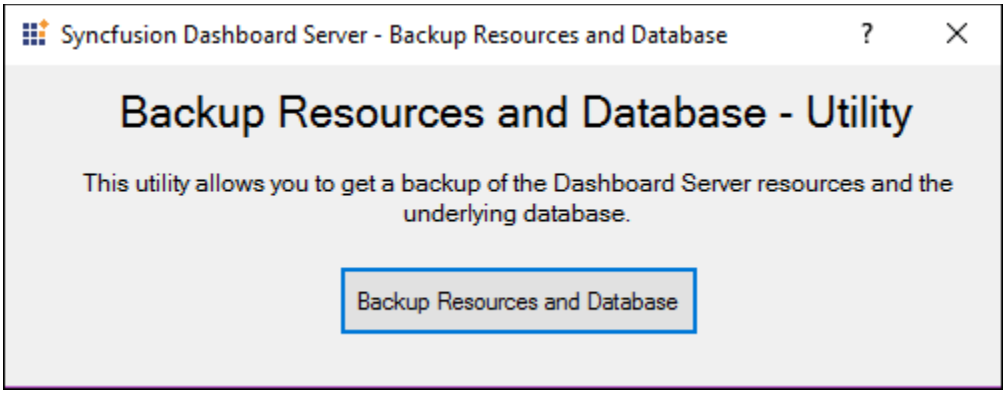
1. SQL CE
2. SQL Server

This utility can take only the resources backup for the below types of Databases.

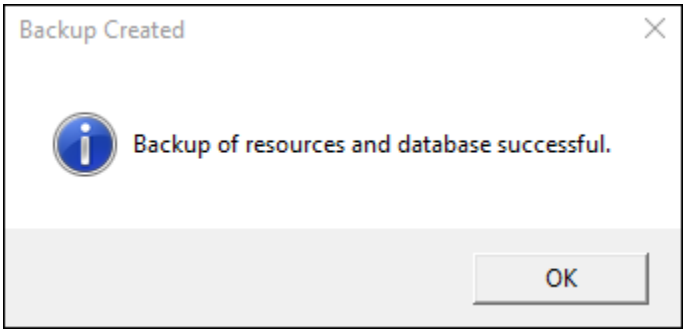
1. MySQL
2. Oracle
3. PostgreSQL

Embedded SQL CE (For Testing purposes only)

Backup utility layout for Embedded SQL CE database.

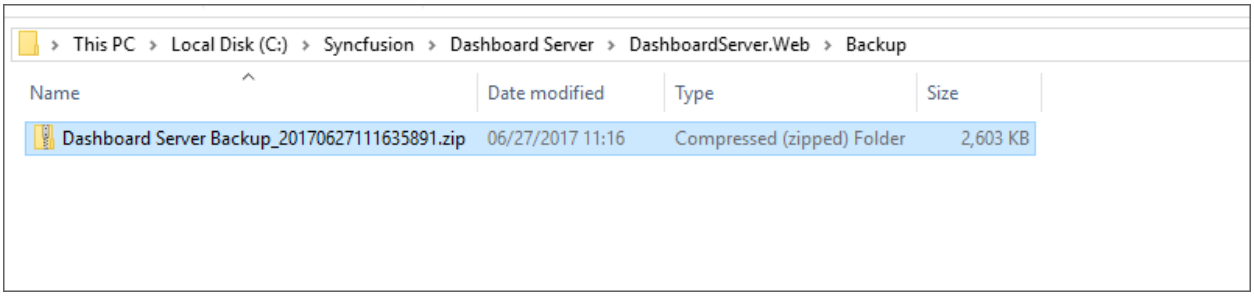


On clicking the Backup Resources and Database button, backup is created and success message is displayed.



Created backup is stored as zip file in deployed folder as below.

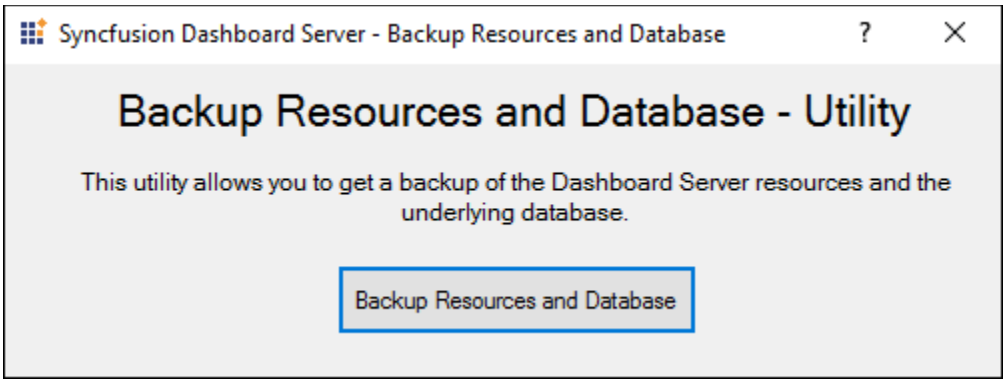
{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\Backup



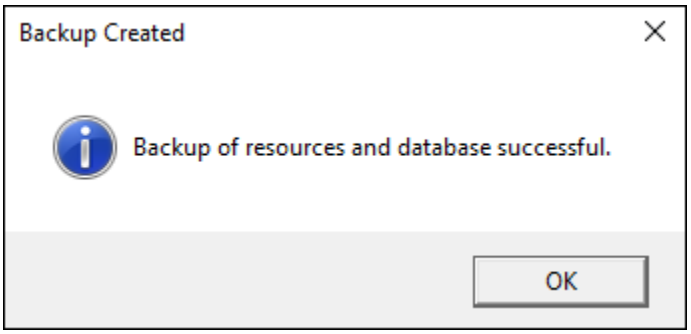
Created zip files contents are as like below,

SQL Server

Backup utility layout for SQL Server database.

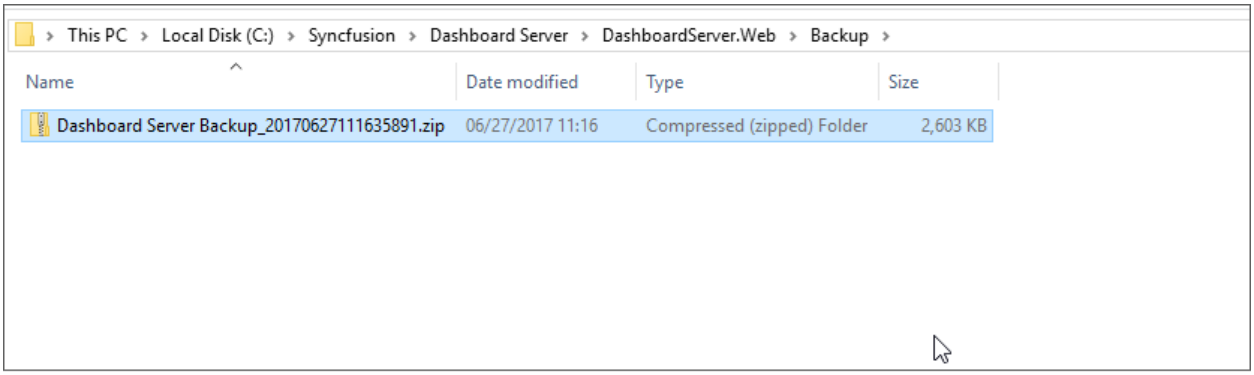


On clicking the Backup Resources and Database button, backup is created and success message is displayed.



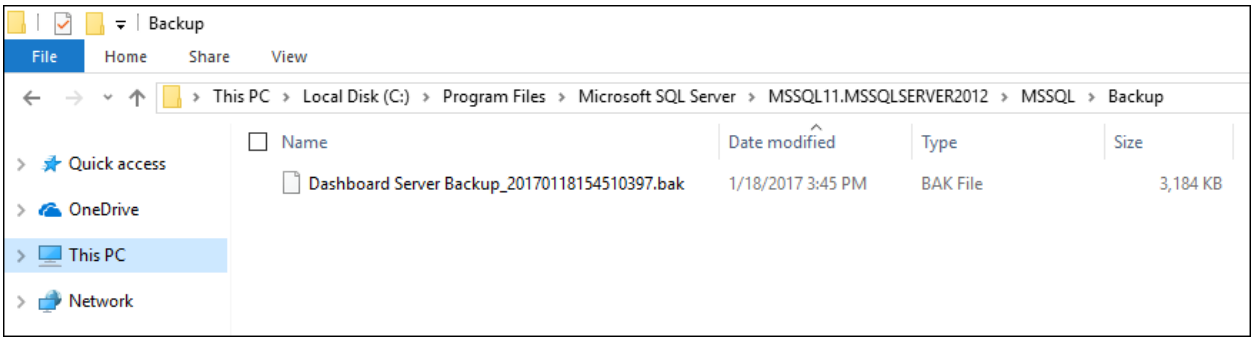
Backup of resources is stored as zip file in deployed folder as below.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\Backup



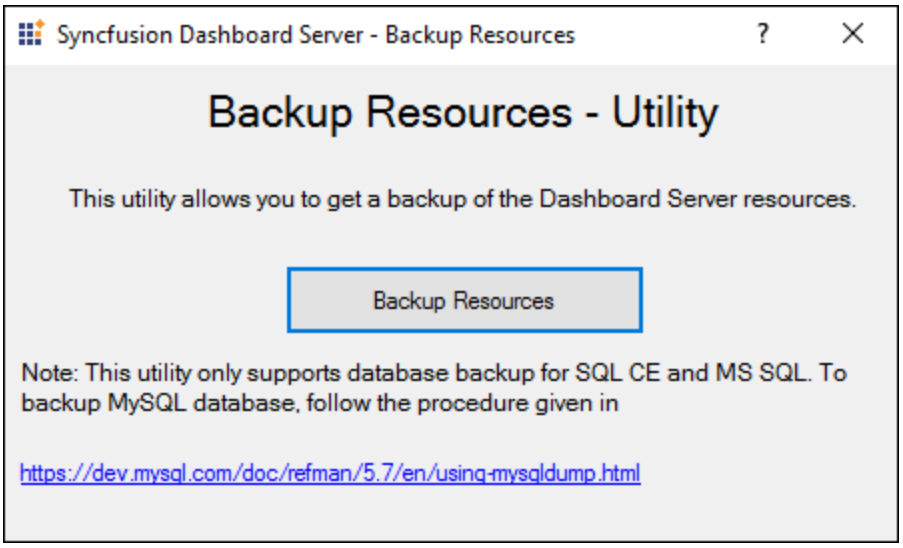
Created zip files contents are as like below,

Backup of database is stored as bak file in C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER2012\MSSQL\Backup

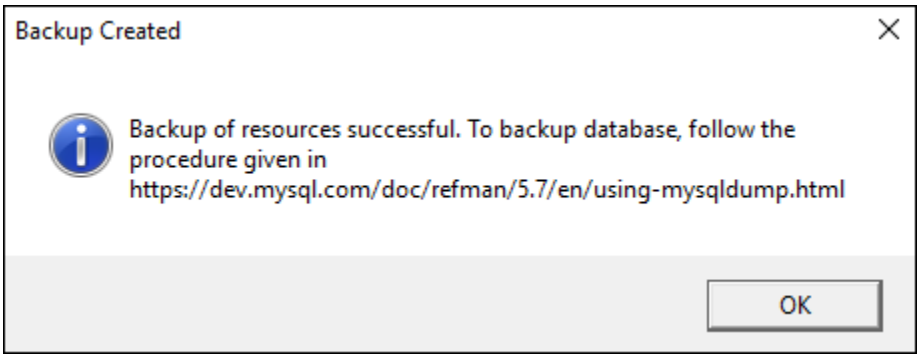


MySQL

Backup utility layout for MySQL database.

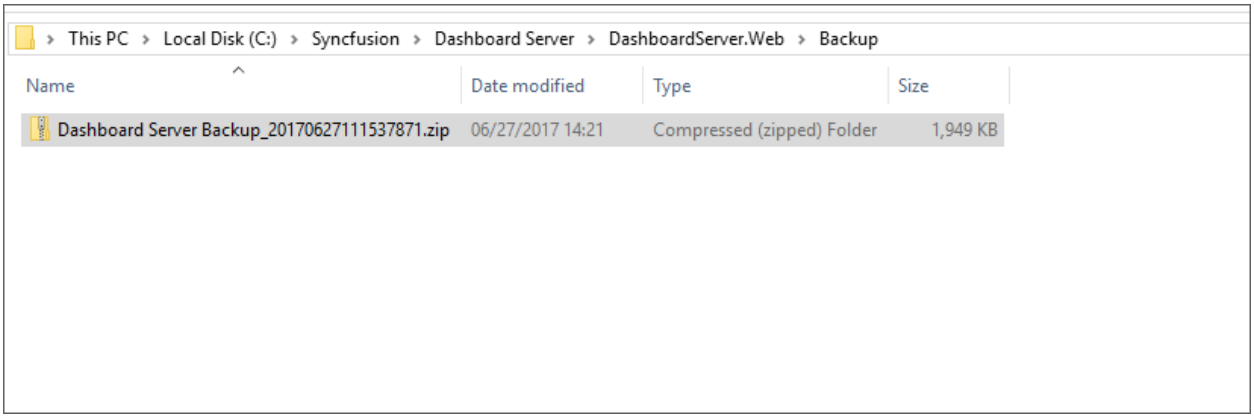


On clicking the Backup Resources button, backup is created and success message is displayed.



Backup of resources is stored as zip file in deployed folder as below.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\Backup

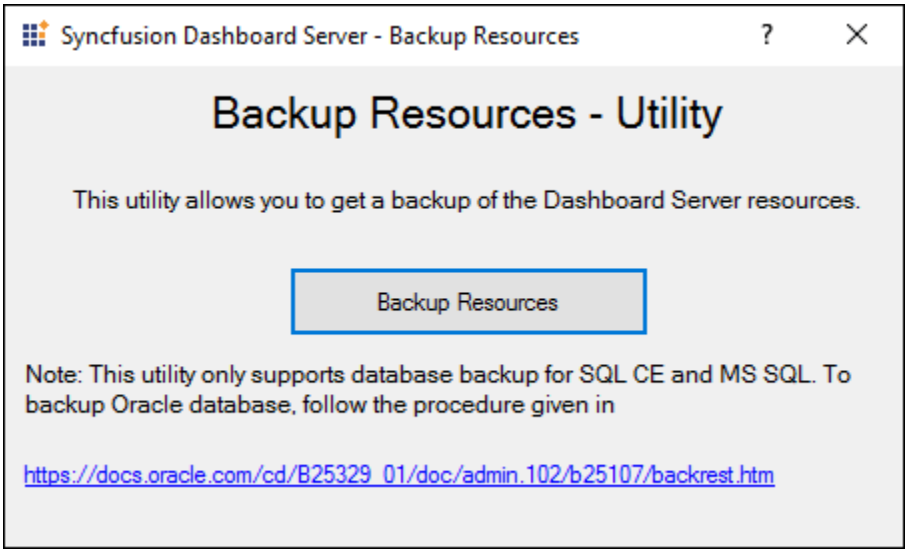


Created zip files contents are as like below,

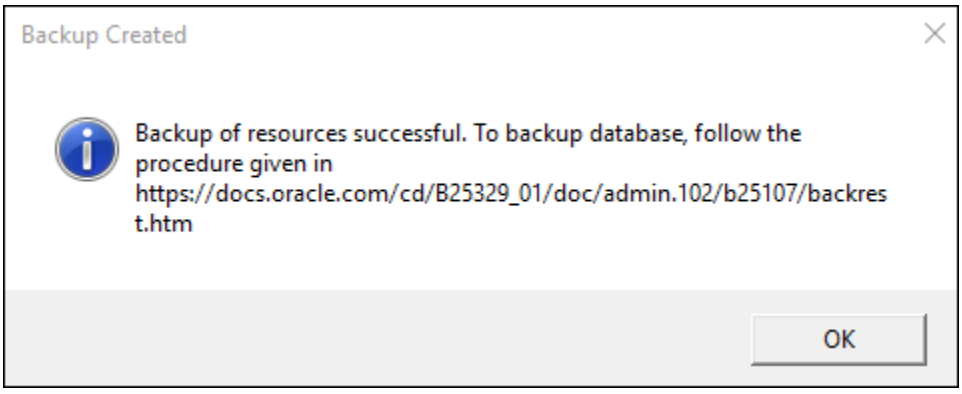
Note: This utility only supports database backup for SQL CE and MS SQL. Database backup can be taken by following the procedure given in https://dev.mysql.com/doc/refman/5.7/en/using-mysqldump.html

Oracle

Backup utility layout for Oracle database.

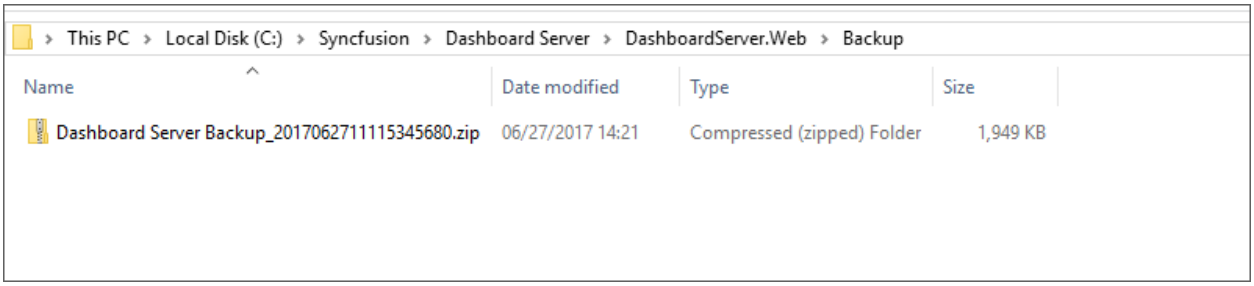


On clicking the Backup Resources button, backup is created and success message is displayed.

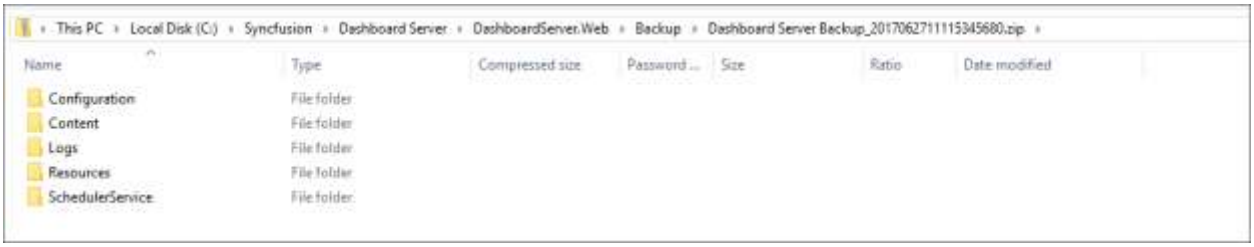


Backup of resources is stored as zip file in deployed folder as below.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\Backup



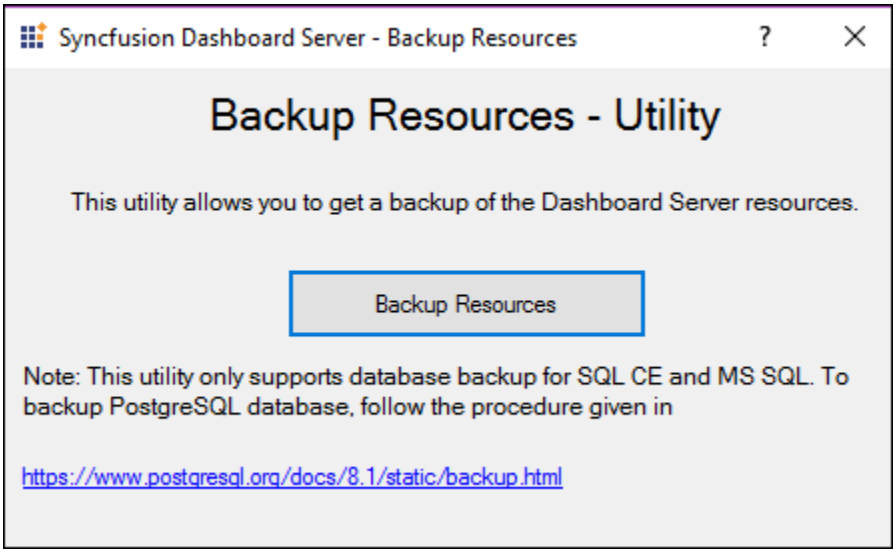
Created zip files' contents are as like below,



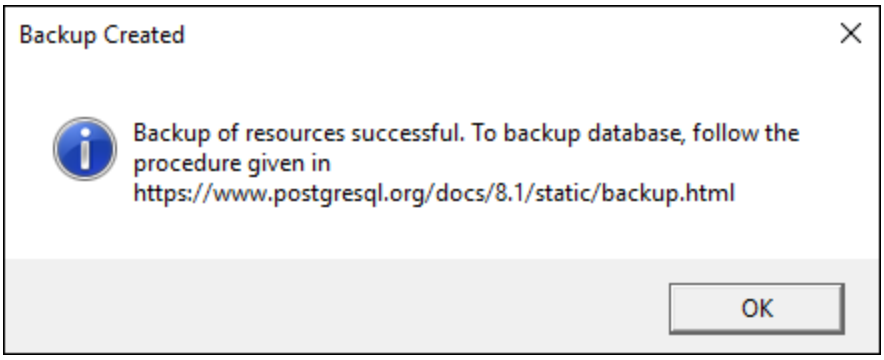
Note: This utility only supports database backup for SQL CE and MS SQL. Database backup can be taken by following the procedure given in https://docs.oracle.com/cd/B25329_01/doc/admin.102/b25107/backrest.htm

PostgreSQL

Backup utility layout for PostgreSQL database.

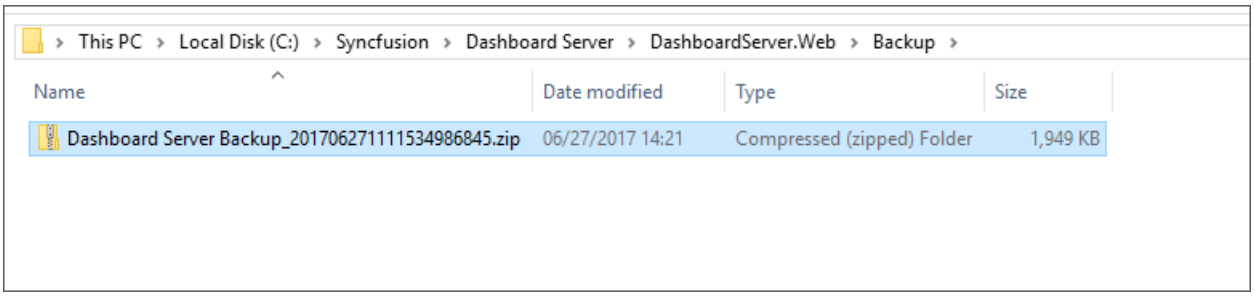


On clicking the Backup Resources button, backup is created and success message is displayed.

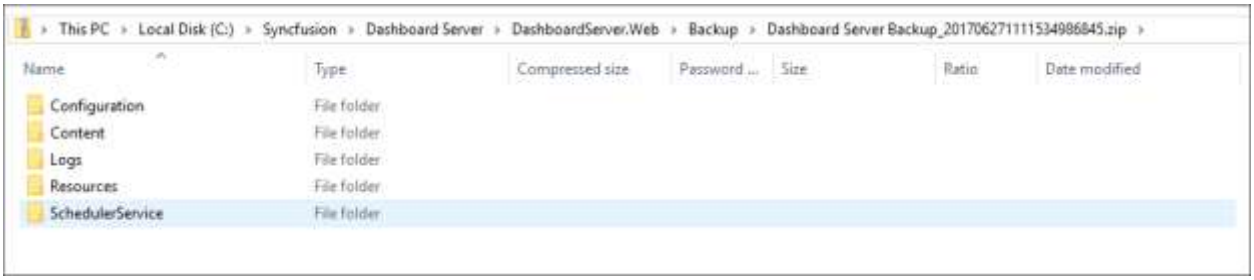


Backup of resources is stored as zip file in deployed folder as below.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\Backup



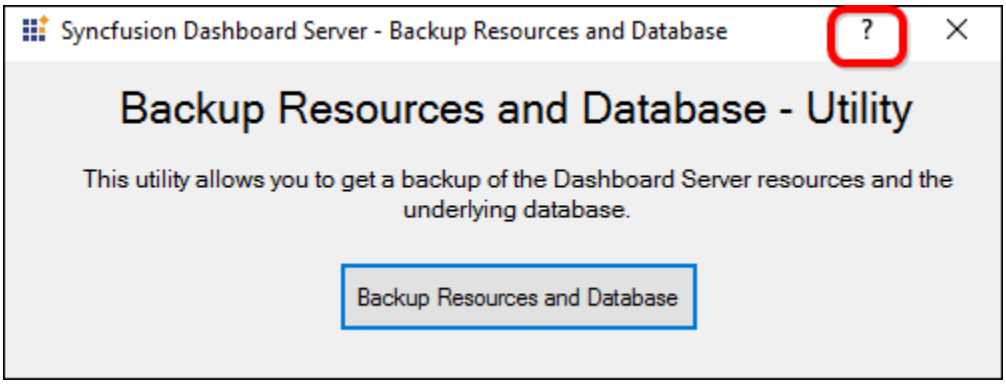
Created zip files contents are as like below,



Note: This utility only supports database backup for SQL CE and MS SQL. Database backup can be taken by following the procedure given in https://www.postgresql.org/docs/8.1/static/backup.html

Help Link

UG Documentation help link is provided at the title bar of the utility for all the database types.



FAQ

The frequently asked questions in Dashboard Server

What all are the files and folders will be generated in the installed machine?

Resources

Dashboards, Data Sources, Widgets that are added in the Dashboard Server are maintained here.

Scheduler Service

Exported Dashboards that are to be emailed and schedule configuration are maintained here.

Content

This folder holds profile pictures and application images.

Logs

You can log errors and debug information from the Dashboard Server application for troubleshooting.

SQLCE Database

If you are choosing SQLCE database as the underlying database for the Dashboard Server, then a .SDF database file will be created and maintained.

Will Azure charge for Syncfusion Dashboard Server license?

No. Azure will not charge for Syncfusion Dashboard Server license. However, you will be charged for the Azure resources used by the Syncfusion Dashboard Server.

- Azure App Service - Based on the selected Azure App Service Plan. Default plan is Basic.
- Storage account - If Blob Storage is selected as your storage account.

How to update the credentials of the Oracle database in Dashboard Server ?

The credentials used to connect the Dashboard Server database to the Syncfusion Dashboard Server can be changed at any time.

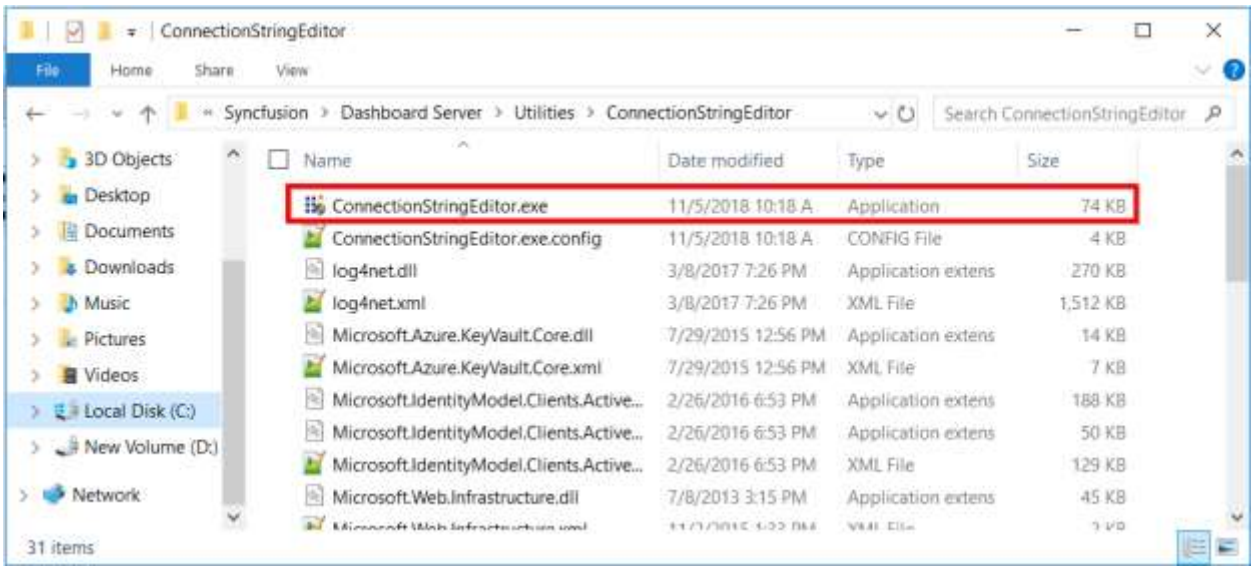
The Dashboard Server is deployed in the following location, by default.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\

For example, C:\Syncfusion\Dashboard Server\DashboardServer.Web\

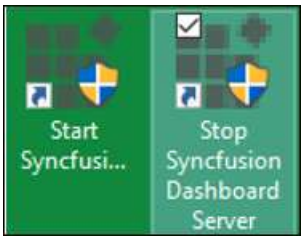
By default, a utility with the Syncfusion Dashboard Server application can be shipped in the following location.

{Windows_Drive}\Syncfusion\Dashboard Server\Utilities\ConnectionStringEditor

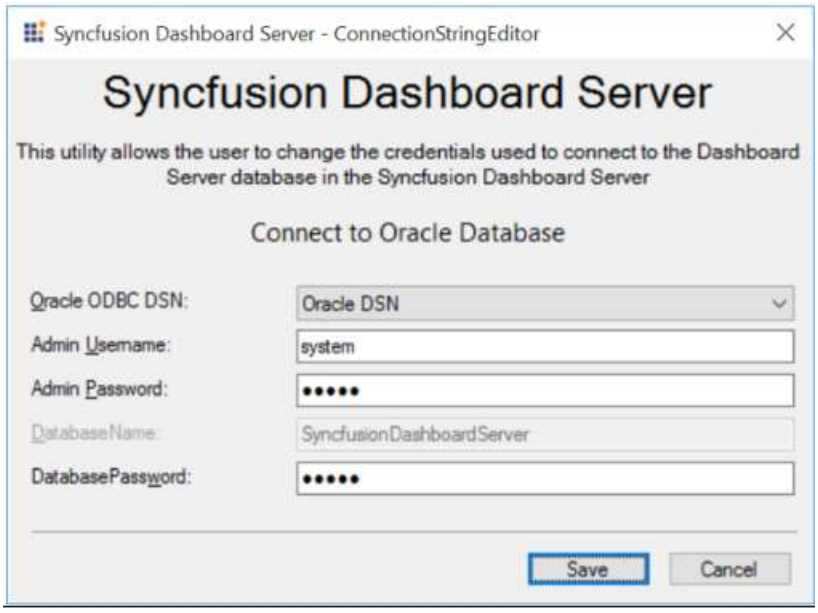


Follow the given steps to change the Database Credentials,

Step 1: Stop the Dashboard Server from the Desktop shortcut – “Stop Syncfusion Dashboard Server”.

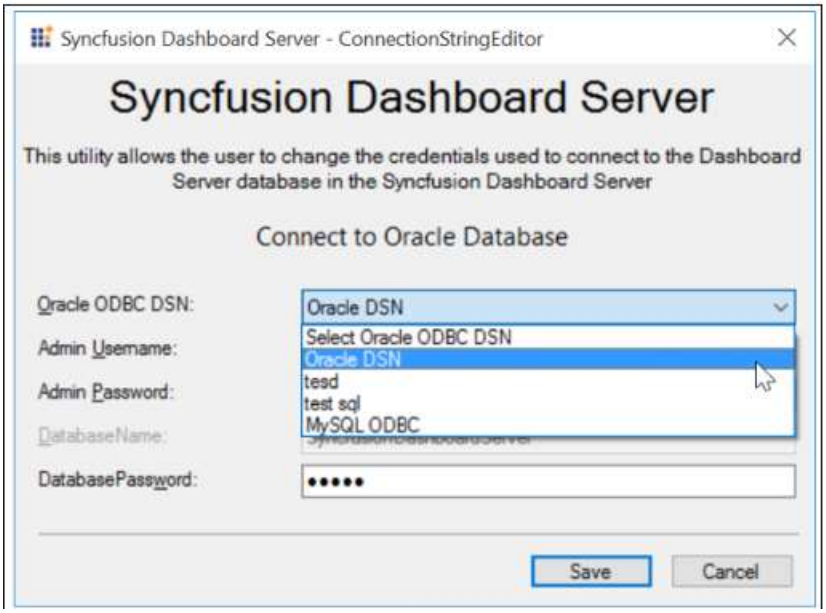


Step 2: Open the ConnectionStringEditor.exe tool in an installed location.



Step 3: The connection details that are supplied while starting up the Dashboard Server application will be populated in the utility.

The DSN, Username, and the password can be changed for the Oracle database.

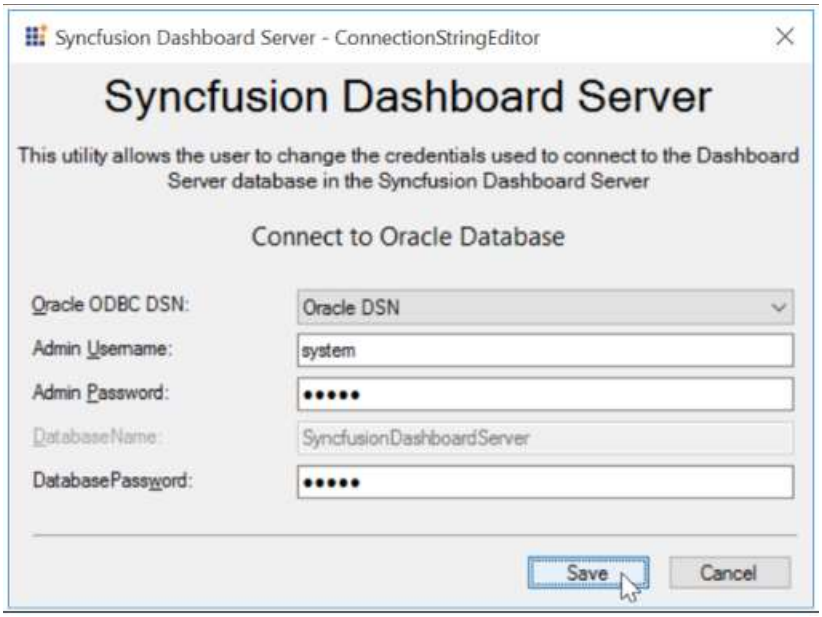


Note: This utility cannot be used when SQL CE database is chosen as the Dashboard Server database in the Syncfusion Dashboard Server.

Step 4: This utility allows to change the Username, Password, and Database Password of the Oracle database.

It can be log in with different user accounts. After filling the connection details, proceed to save.

The utility tests the connection to the database with given details. If the connection test passes, the new connection details will be updated in the connection string of the Dashboard Server database.



The tool changes the Database Credentials and returns a “success” message if the process was completed successfully.

How to update the credentials of the PostgreSQL database in Dashboard Server ?

The credentials that are used to connect to the Dashboard Server database in the Syncfusion Dashboard Server can be changed at any time.

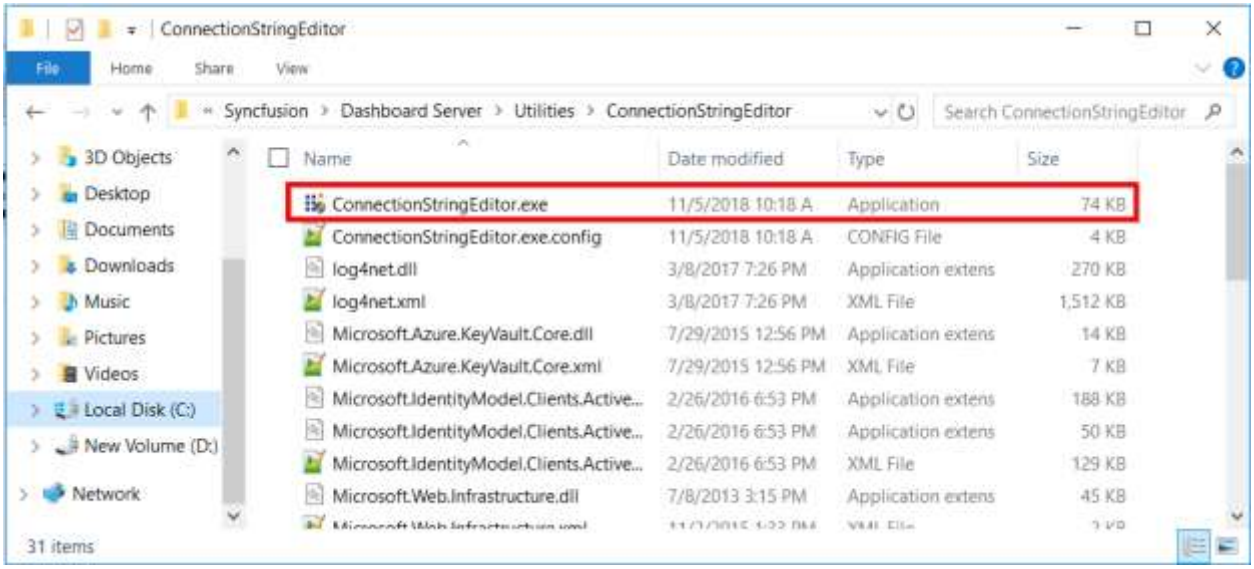
The Dashboard Server is deployed in the following location by default.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\

For example, C:\Syncfusion\Dashboard Server\DashboardServer.Web\

A utility has been shipped with the Syncfusion Dashboard Server application in the following location by default.

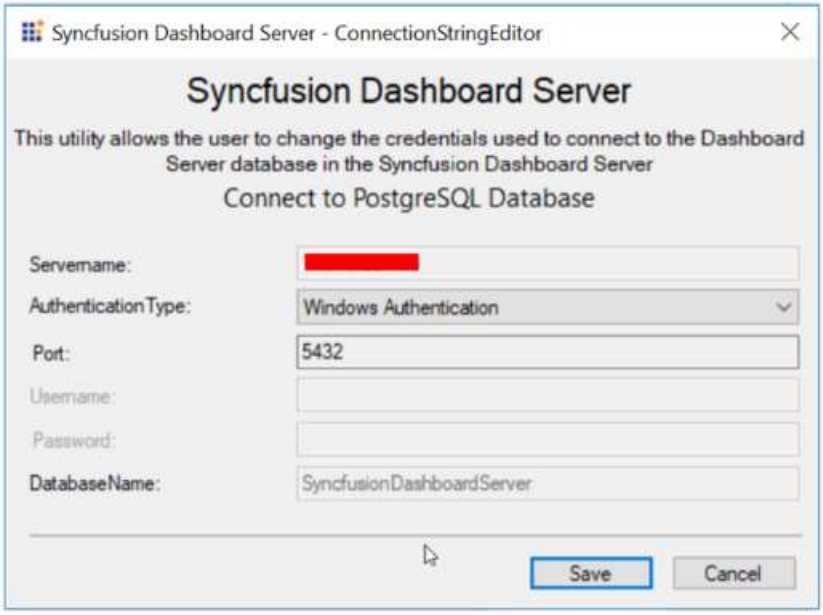
{Windows_Drive}\Syncfusion\Dashboard Server\Utilities\ConnectionStringEditor



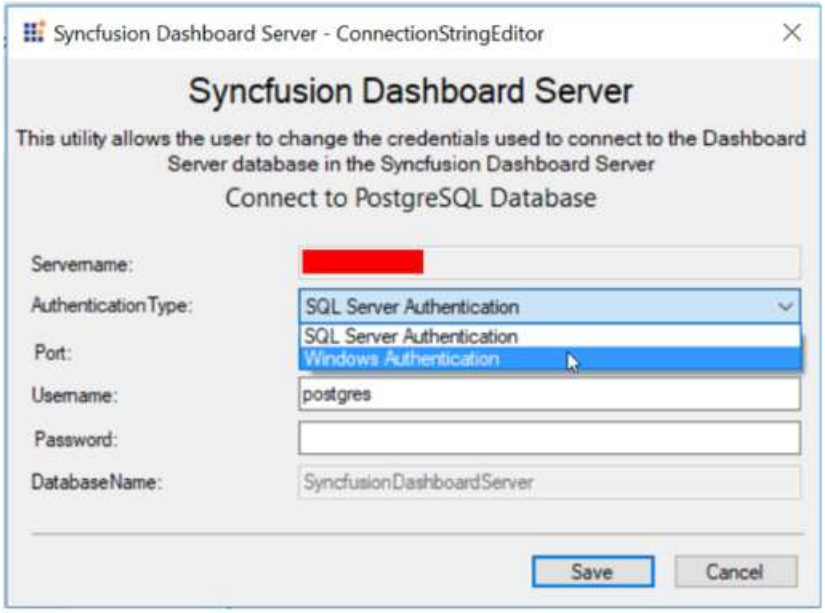
Follow the given steps to change the Database Credentials:

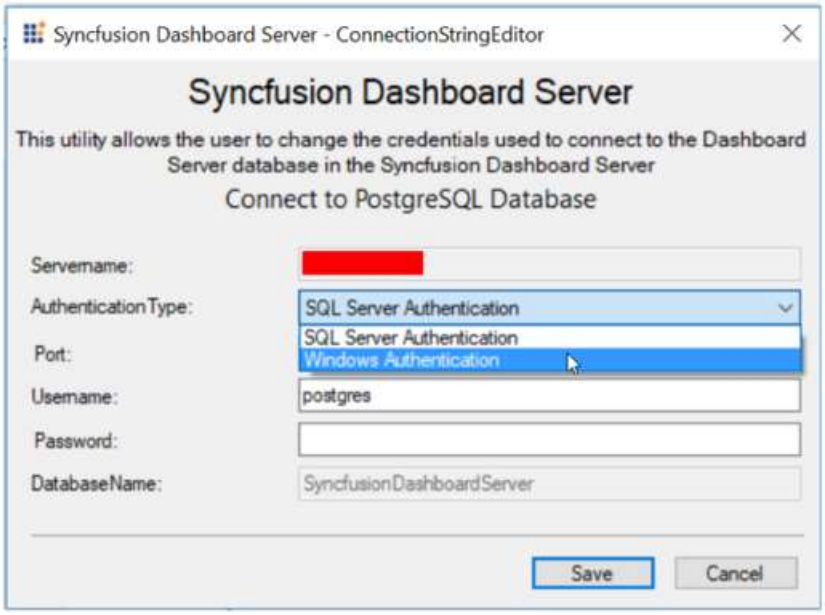
Step 1: Stop the Dashboard Server through the Desktop shortcut – “Stop Syncfusion Dashboard Server”.

Step 2: Open the ConnectionStringEditor.exe tool in the installed location.



Step 3: Run the utility. The connection details that are supplied while starting up the Dashboard Server application will be populated in the utility. You can change the Authentication Type and Server, and you cannot change the Database name in the utility.

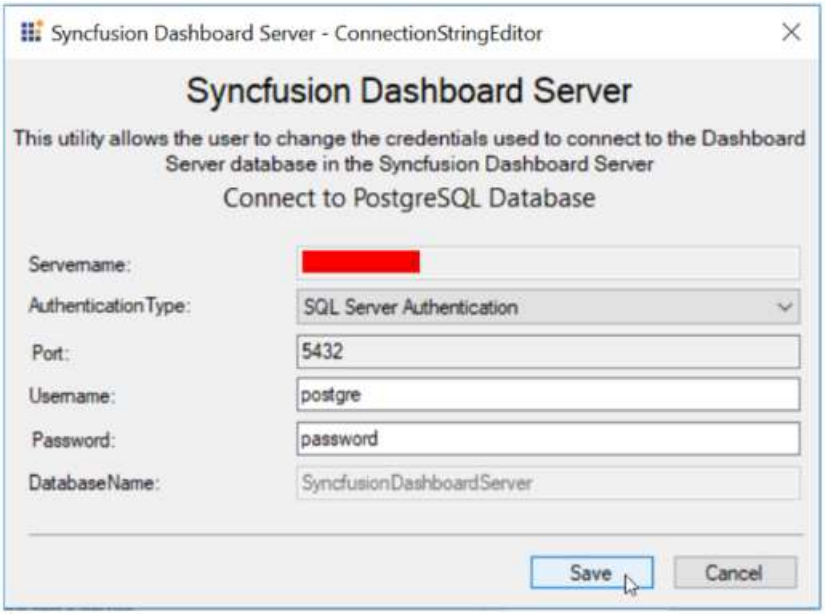




Note: This utility cannot be used when SQL CE database is chosen as the Dashboard Server database in the Syncfusion Dashboard Server.

Step 4: Enter username and password to fill if “SQL Server Authentication” type has been selected. For “Windows Authentication” type, it is not needed. After filling the connection details, proceed to save.

The utility will test the connection to the database with the given details. If the connection test is passed, the new connection details will be updated in the connection string of the Dashboard Server database.



The tool changes the Database Credentials and return a “success” message if the process was completed successfully.

If there is an error occurred in the process, the tool will generate an error log in text format in the following location and will display the error log file name.

Error log file location: %temp%/Syncfusion Dashboard Server/{file_name}

Please create a support incident and attach the error log file for us to check the issue and give a solution.

How to update the credentials of the MSSQL database in Dashboard Server ?

The credentials that are used to connect to the Dashboard Server database in the Syncfusion Dashboard Server can be changed at any time.

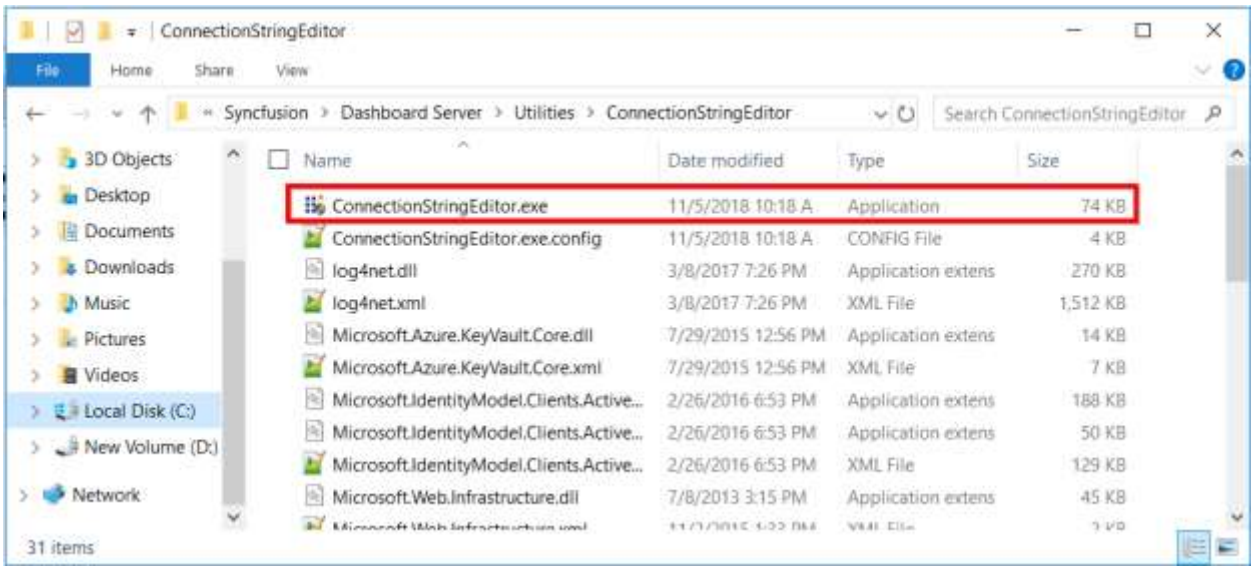
The Dashboard Server is deployed in the following location by default.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\

For example, C:\Syncfusion\Dashboard Server\DashboardServer.Web\

A utility has been shipped with the Syncfusion Dashboard Server application in the following location.

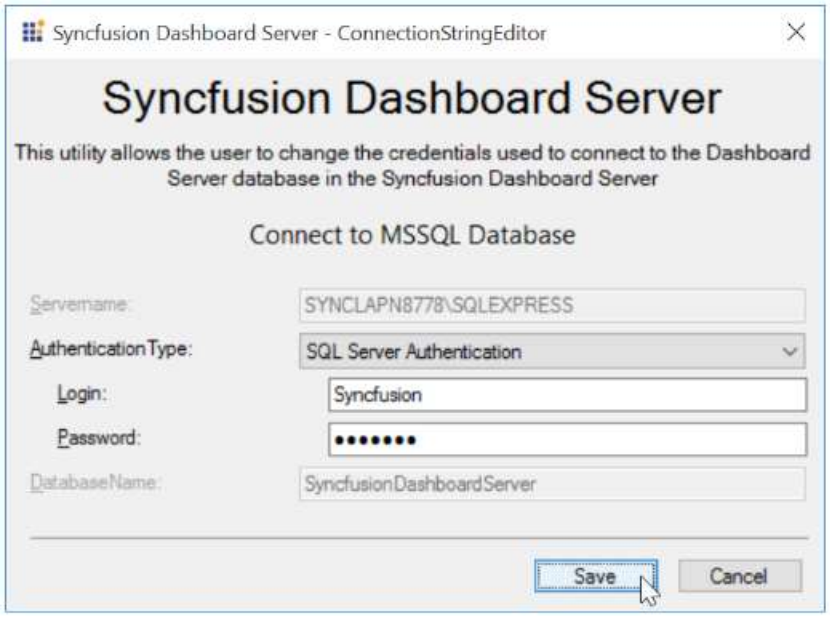
{Windows_Drive}\Syncfusion\Dashboard Server\Utilities\ConnectionStringEditor



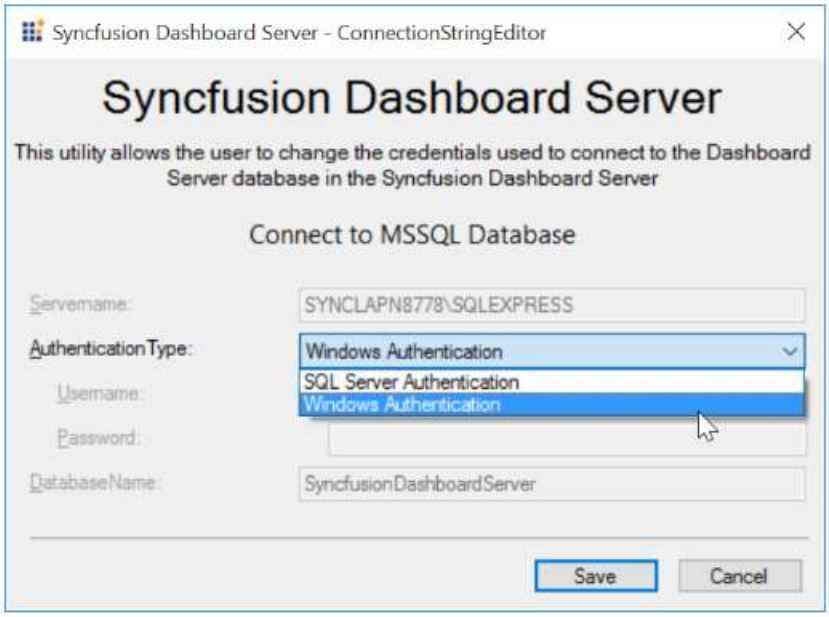
Follow the given steps to change the Database Credentials:

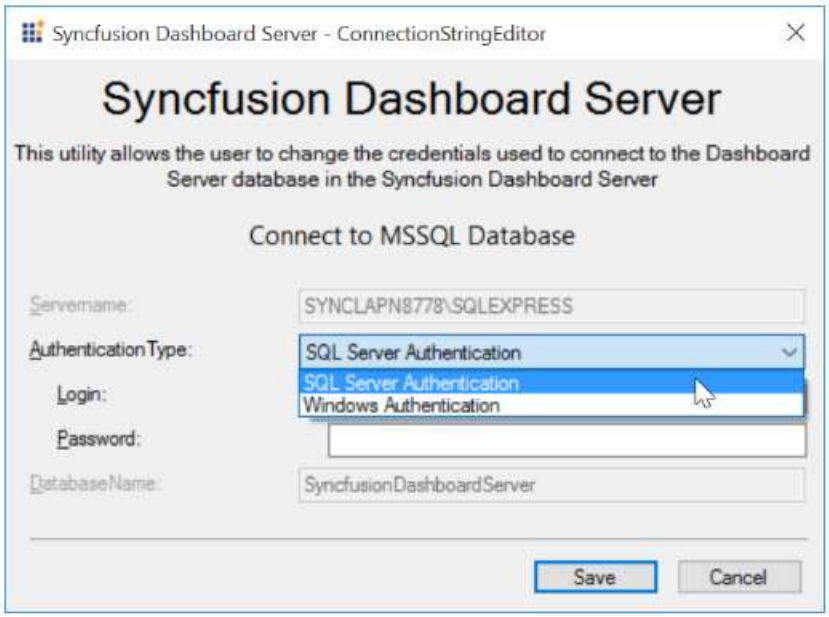
Step 1: Stop the Dashboard Server through the Desktop shortcut – “Stop Syncfusion Dashboard Server”.

Step 2: Open the ConnectionStringEditor.exe tool in the installed location.



Step 3: Run the utility. The connection details that are supplied while starting up the Dashboard Server application will be populated in the utility. You can change the Authentication Type and the Server, and you cannot change the Database name in the utility.

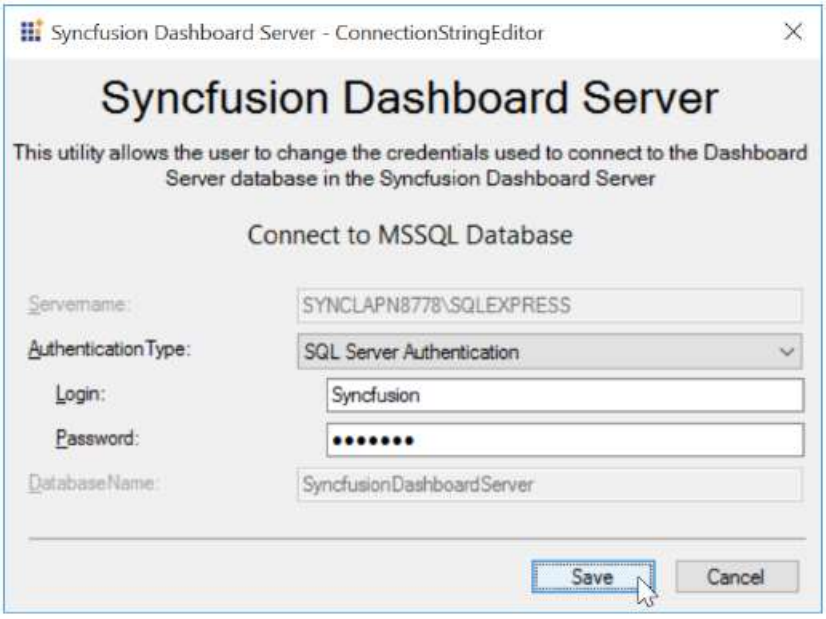




Note: This utility cannot be used when SQL CE database is chosen as the Dashboard Server database in the Syncfusion Dashboard Server.

Step 4: Enter username and password to fill if “SQL Server Authentication” type has been selected. For “Windows Authentication” type, it is not needed. After filling the connection details, proceed to save.

The utility tests the connection to the database with the given details. If the connection test is passed, the new connection details will be updated in the connection string of the Dashboard Server database.



The tool changes the Database Credentials and returns a “success” message if the process was completed successfully.

If there is an error occurred in the process, the tool will generate an error log in text format in the following location and display the error log file name.

Error log file location: %temp%/Syncfusion Dashboard Server/{file_name}

Please create a support incident and attach the error log file for us to check the issue and give a solution.

How to update the credentials of the MySQL database in Dashboard Server ?

The credentials that are used to connect to the Dashboard Server database in the Syncfusion Dashboard Server can be changed at any time.

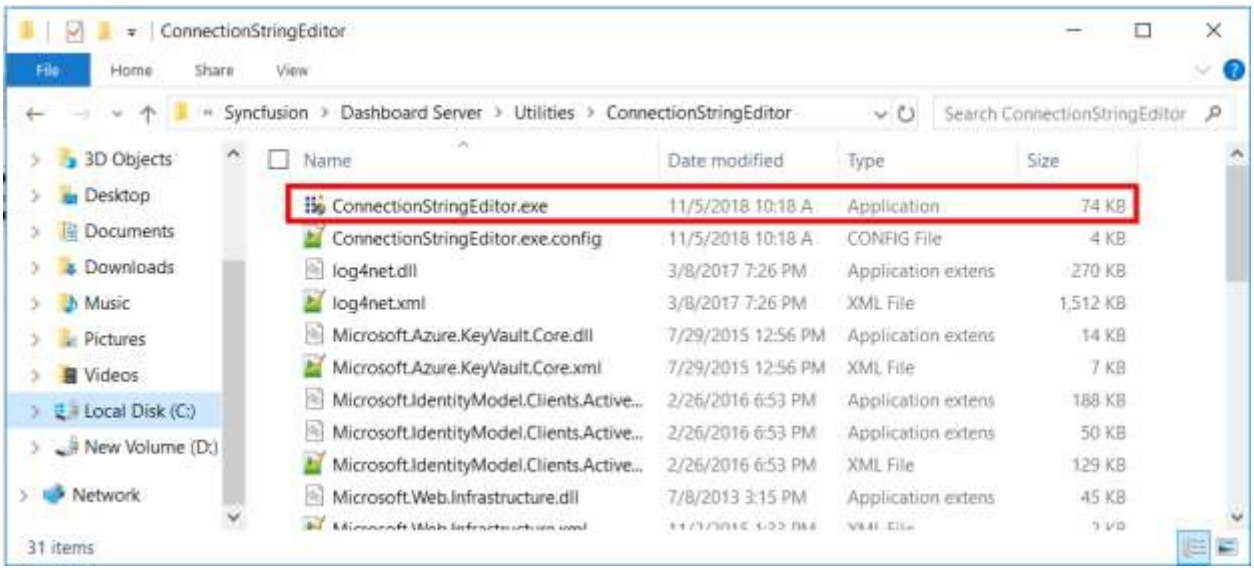
The Dashboard Server is deployed in the following location by default.

{Windows_Drive}\Syncfusion\Dashboard Server\DashboardServer.Web\

For example, C:\Syncfusion\Dashboard Server\DashboardServer.Web\

A utility has been shipped with the Syncfusion Dashboard Server application in the following location.

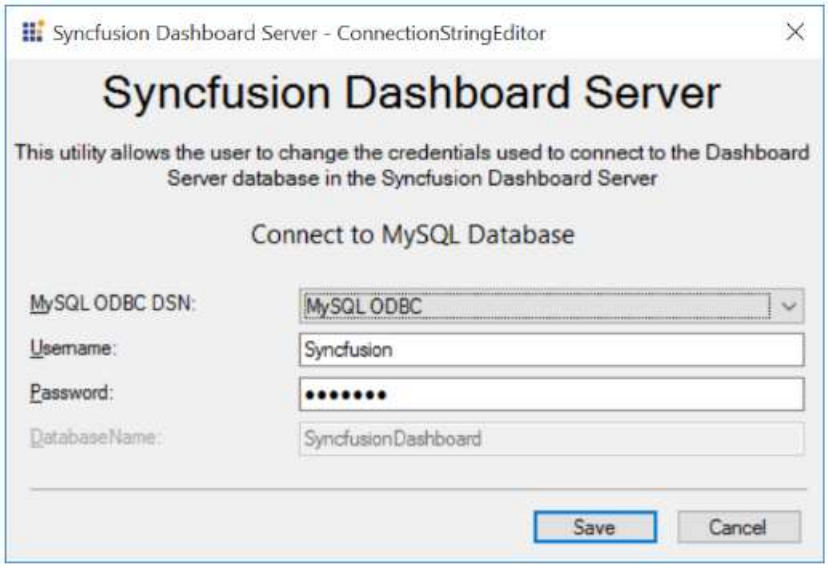
{Windows_Drive}\Syncfusion\Dashboard Server\Utilities\ConnectionStringEditor



Follow the given steps to change the Database Credentials:

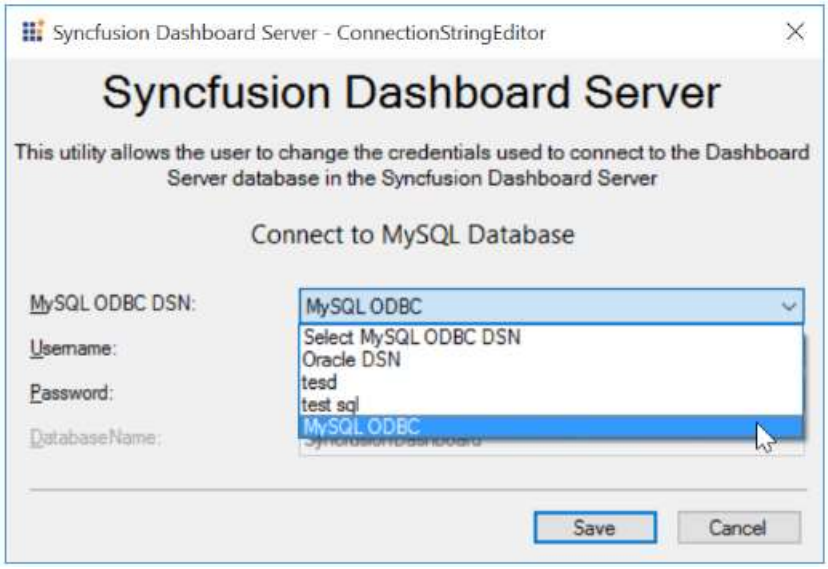
Step 1: Stop the Dashboard Server through the Desktop shortcut – “Stop Syncfusion Dashboard Server”.

Step 2: Open the ConnectionStringEditor.exe tool in the installed location.



Step 3: Run the utility. The connection details that are supplied while starting up the Dashboard Server application will be populated in the utility.

The DSN, Username, and Password can be changed for the MySQL database.



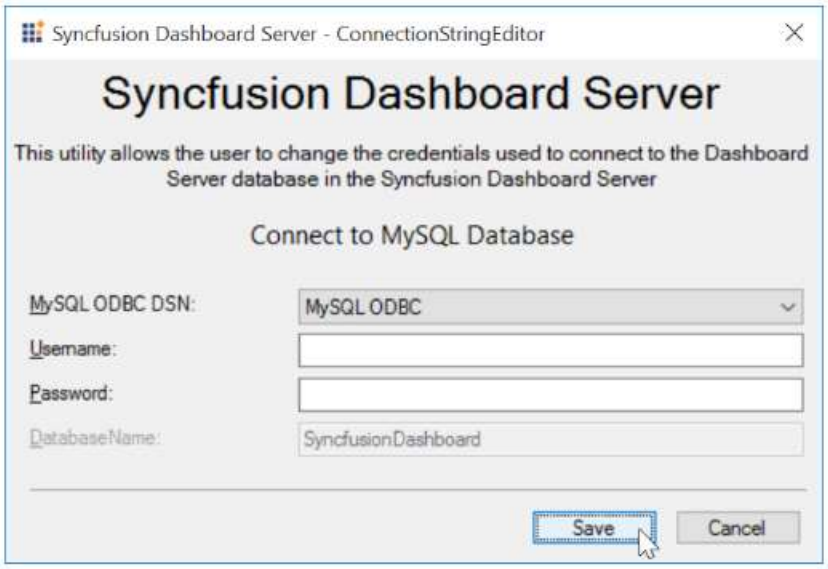
Note: This utility cannot be used when SQL CE database is chosen as the Dashboard Server database in the Syncfusion Dashboard Server.

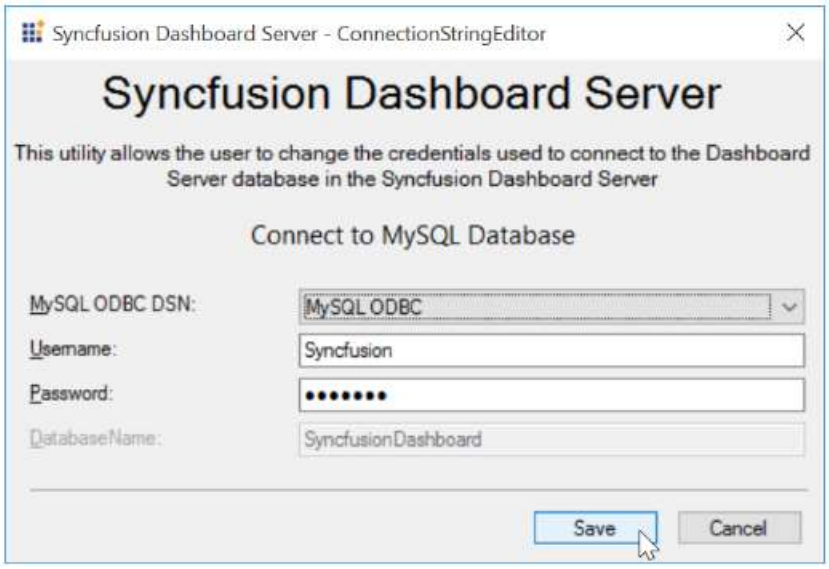
If the Username and Password for the database has already been given in the ODBC driver, no need to fill it in this utility.

In this case, leave the Username and Password field as empty and directly save the connection details.

If the login credentials are not given in the ODBC driver, this utility needs a Username and Password field to fill.

The utility tests the connection to the database with the given details. If the connection test is passed, the new connection details will be updated in the connection string of the Dashboard Server database.





The tool changes the Database Credentials and returns a “success” message if the process was completed successfully.

If there is an error occurred in the process, the tool will generate an error log in text format in the below location and display the error log file name.

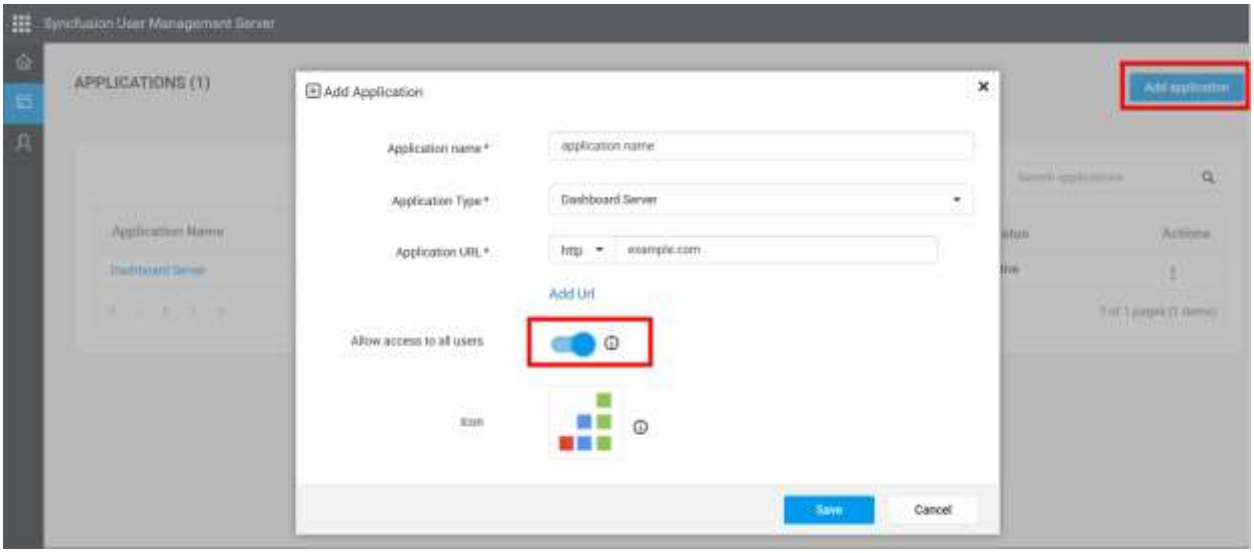
Error log file location: %temp%/Syncfusion Dashboard Server/{file_name}

Please create a support incident and attach the error log file for us to check the issue and give a solution.

How to grant access to all users for Dashboard Server?

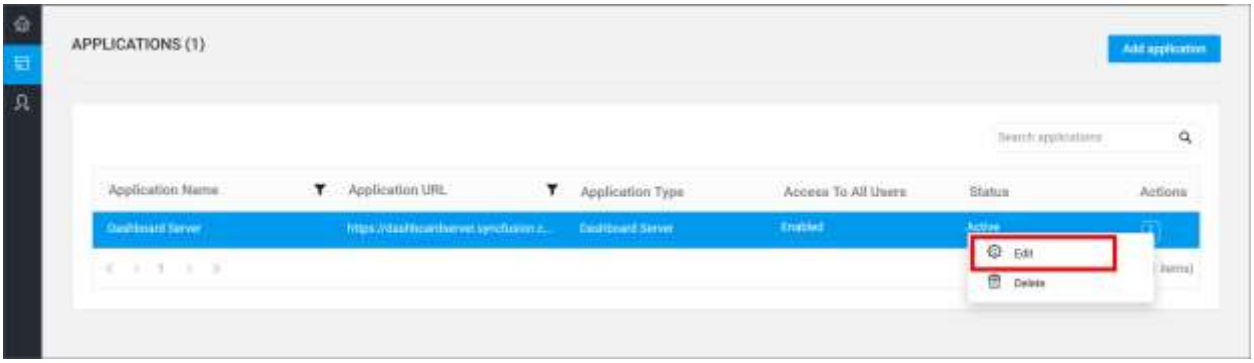
If you going to add a new application and want grant access to all users means enable “Allow access to all users” button in Add Application dialog box of User Management Server.

Go to Application Management -> Add Application -> Enable “Allow access to all users”.



If you want to grant access to all users for the existing applications.

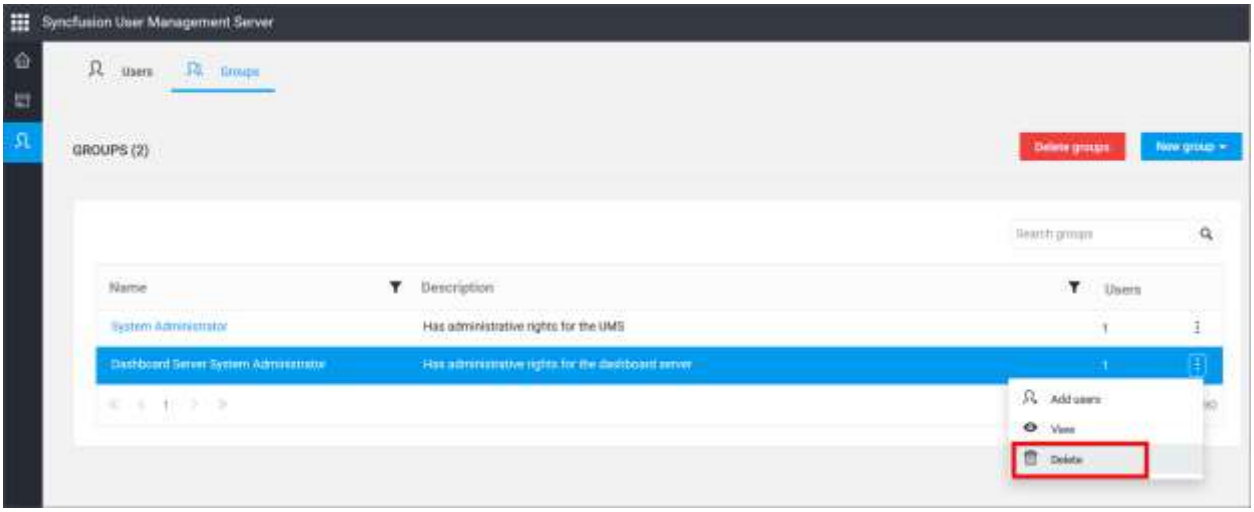
Go to Application Management -> Choose your application -> Choose edit option in the context menu of Actions button -> Enable “Allow access to all users”.



Blank page appears after deleting App Data of Dashboard Server and reconfigured. How to resolve this?

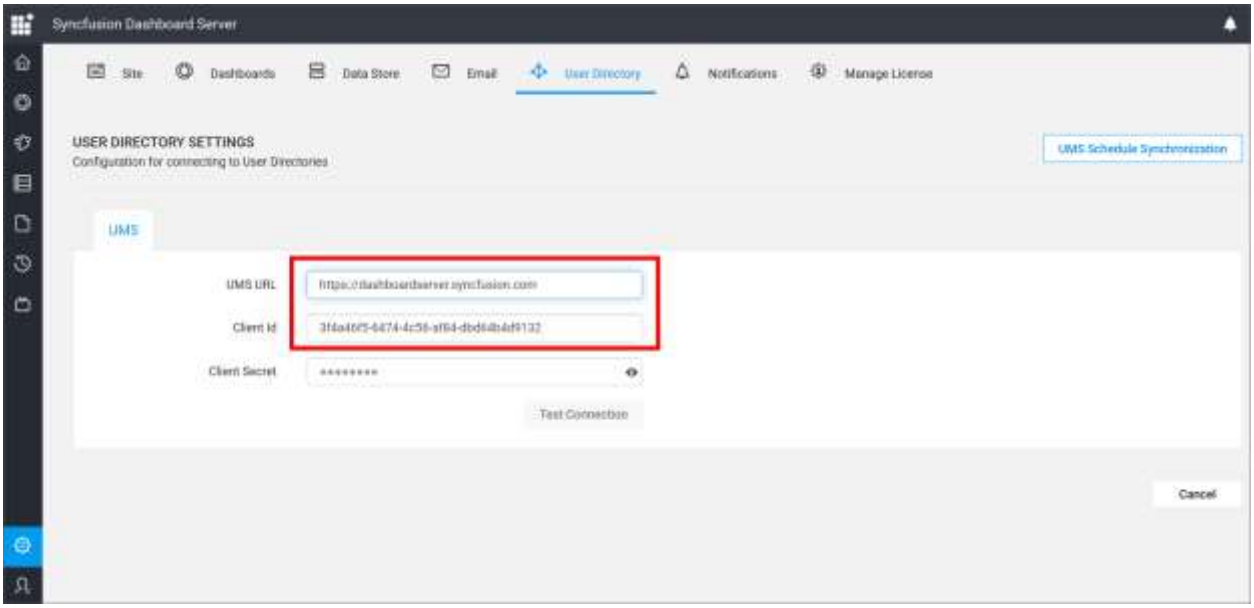
This occurs due to the Dashboard Server trying to create a same group name at the time of server configuration. To resolve follow the below steps.

1. Go to Groups page in User Management Server. 2. Delete the Admin group of respective application.



How to find the current User Management Server details in Dashboard Server?

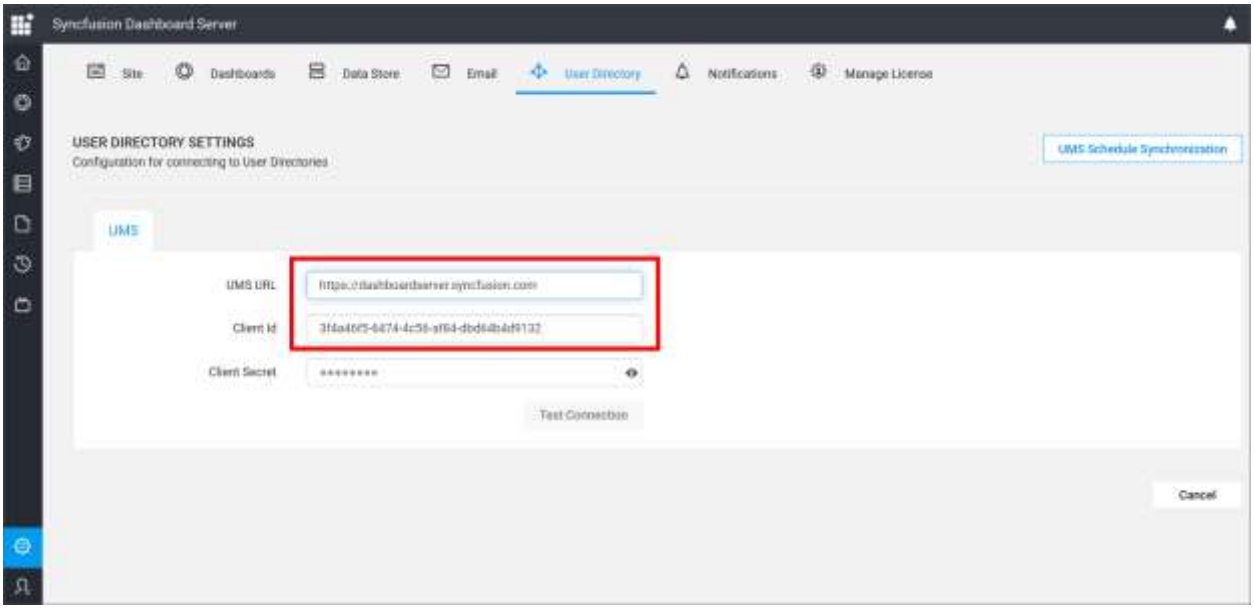
You can find the currently pointing User Management Server URL and Client ID inside Settings -> User Directory settings.



How to change UMS URL, Client Id and Client Secret in Dashboard Server?

You can change the UMS URL, Client Id and Client Secret in config.xml files of Dashboard Server.

Before adding credentials in config file, you can test the connections in the Settings -> User Directory settings page.



For adding credentials follow below steps.

1. Find the Dashboard Server config files in below locations, *C:\Syncfusion\Dashboard Server\DashboardServer.Web\Configuration\Config.xml* C:\Syncfusion\Dashboard Server\DashboardServer.Web\App_Data\Configuration\Config.xml 2. Open Config.xml files from above locations and find the ClientId, ClientSecret and UmsUrl nodes. 3. Update the nodes with valid credentials. Now login in to Dashboard Server, It will update the latest values from config file to Dashboard Server Database if the connection is valid one.

How the Group conflicts in Data Migration is handled programmatically?

While upgrading the Dashboard Server to v3.2, The Server simply updates the group name with application name if any group conflicts occurs among User Management Server and Dashboard Server.

How To

How to create DSN for MySQL

This topic describes how to create DSN for MySQL that can be used to connect MySQL database from Syncfusion Dashboard Server.

An ODBC Data Source Name (DSN) stores information for establishing a connection to a database on a remote database server. A system DSN provides access to multiple users, rather than only the user who created it.

Download and Install MySQL ODBC Driver

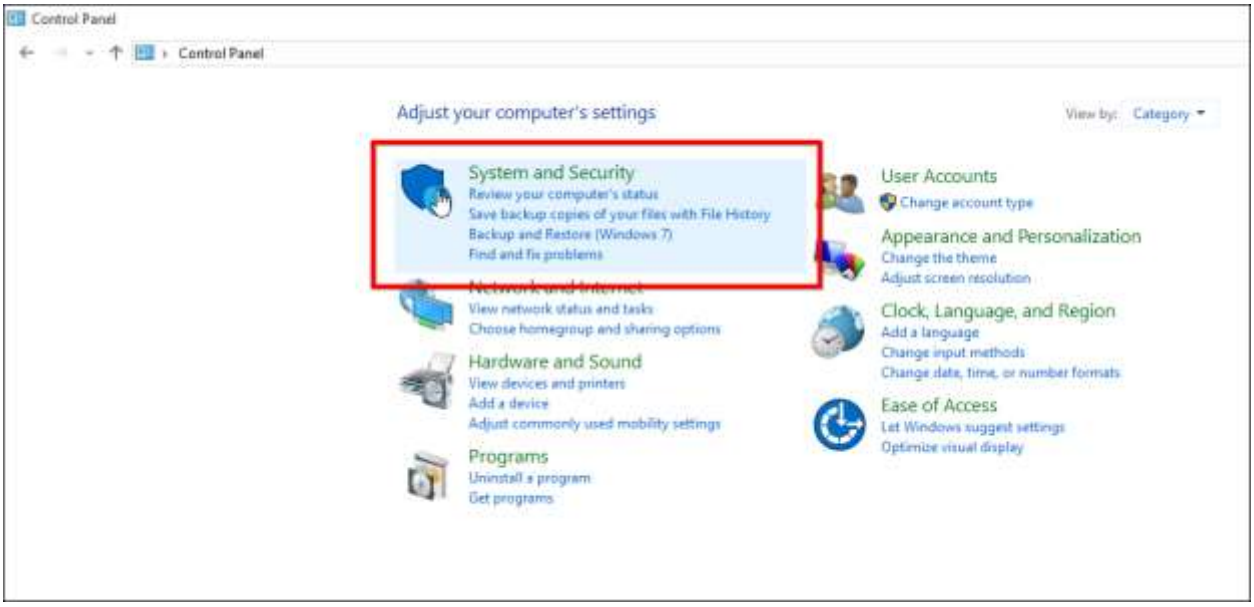
Download and install the 32-bit Connector/ODBC driver from the Downloads section of the MySQL website .

Create DSN

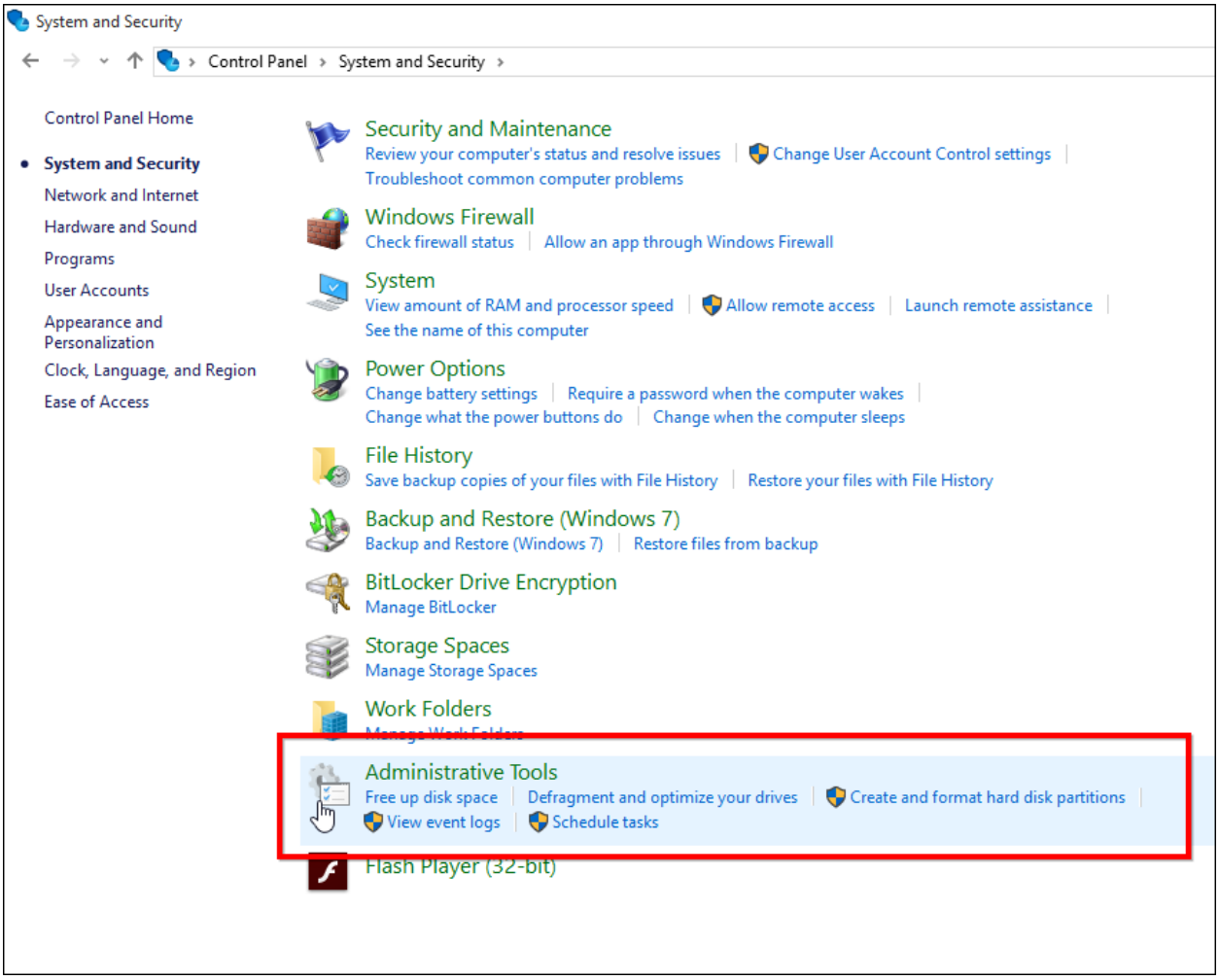
After installing MySQL ODBC driver follow the below steps to create DSN for MySQL database.

Open ODBC Data Source Administration Tool

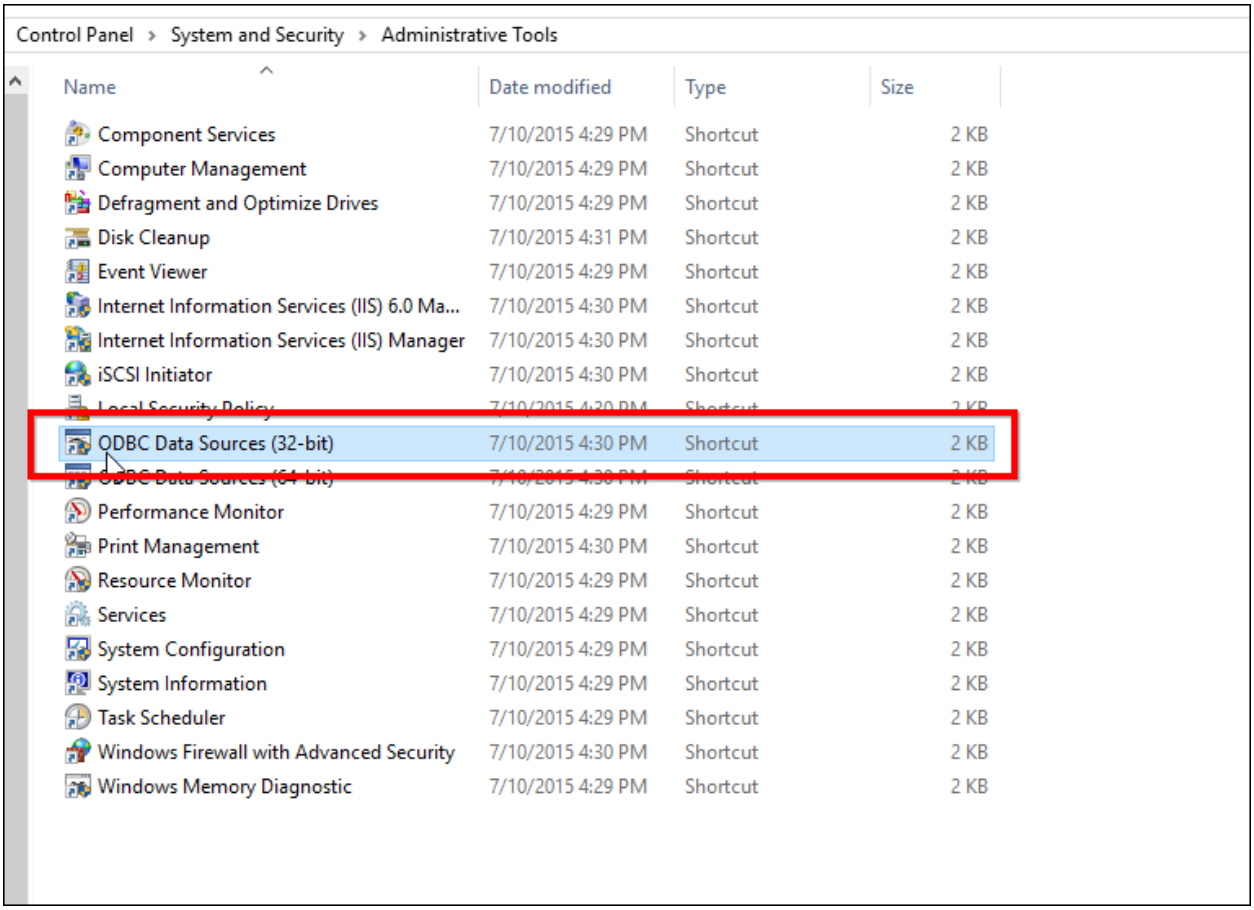
Open Control Panel and select System and Security.



Select Administrative Tools from the list of options.

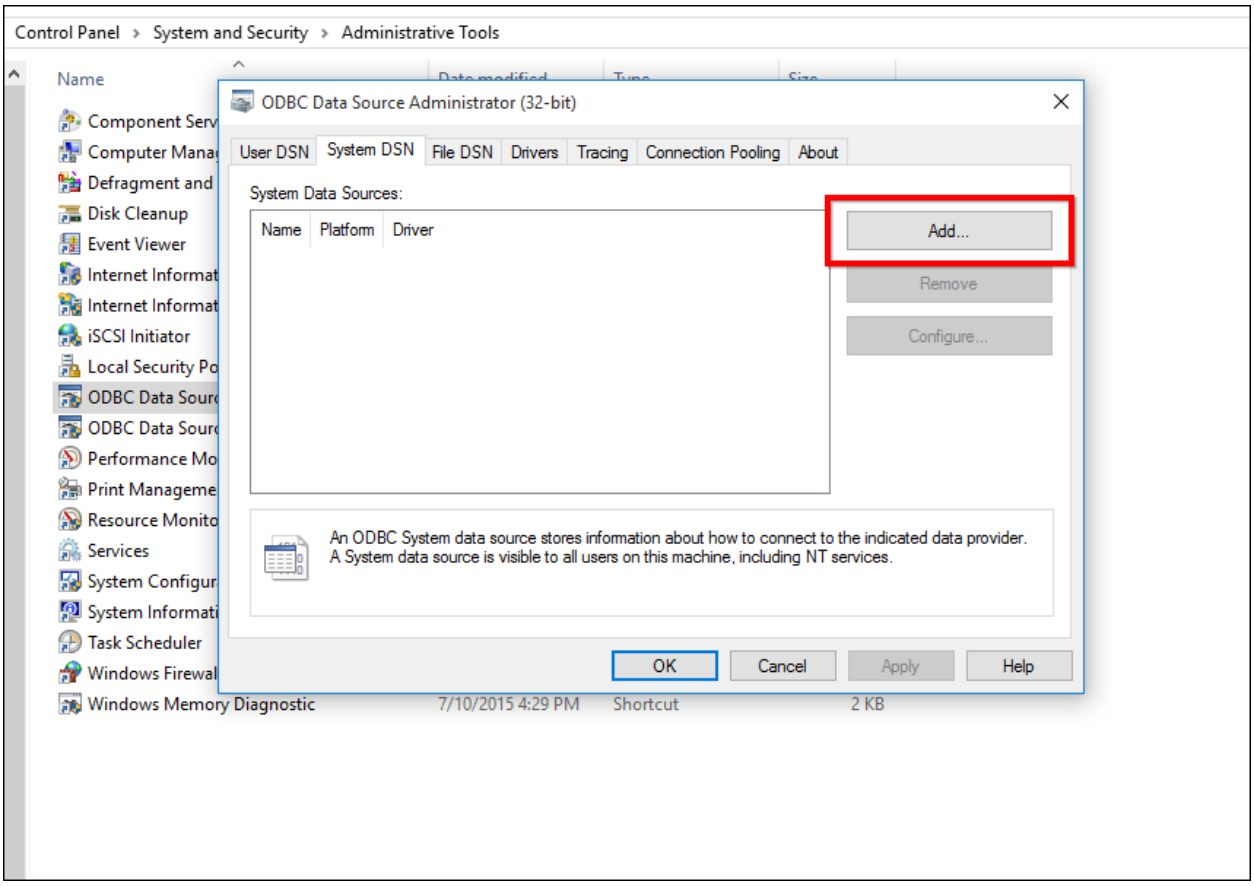


Select ODBC Data Sources (32-bit) from the list of options.

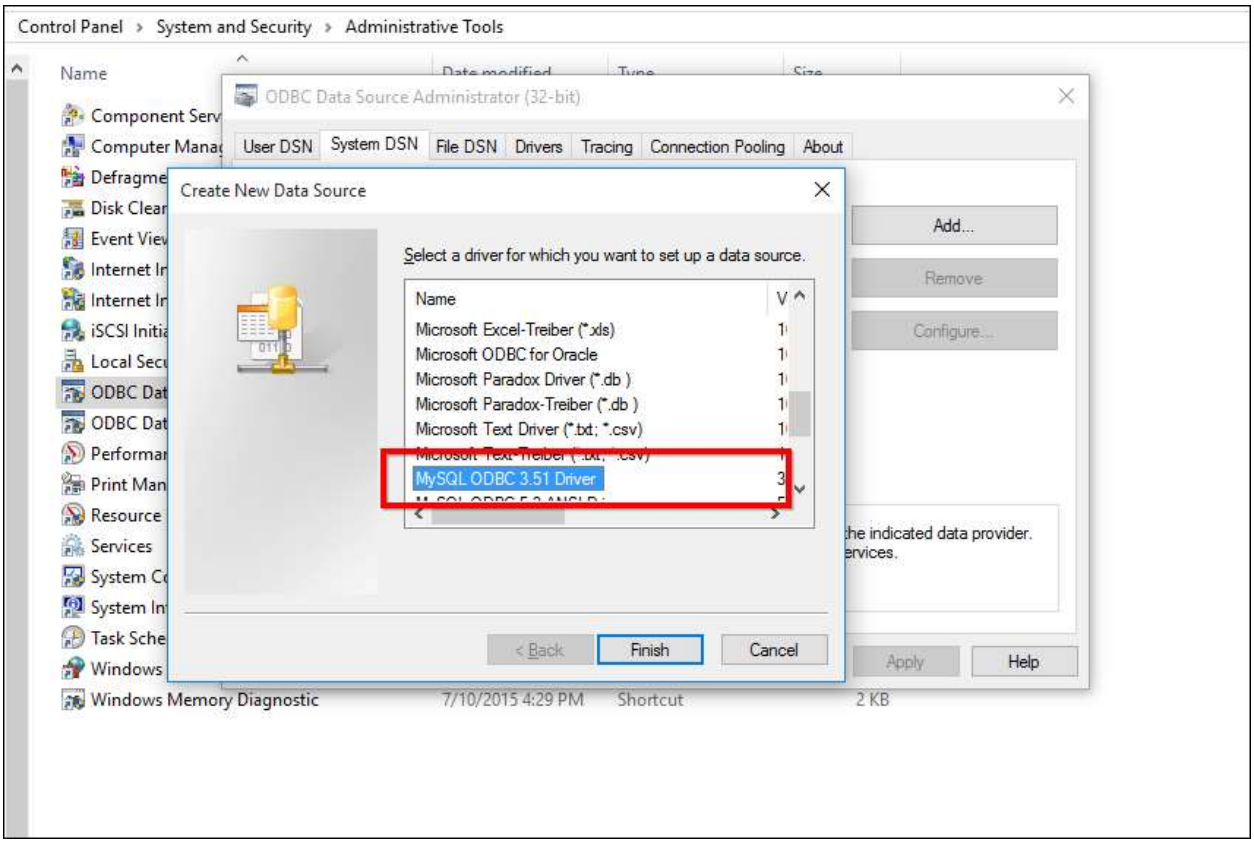


Create DSN with MySQL ODBC Driver

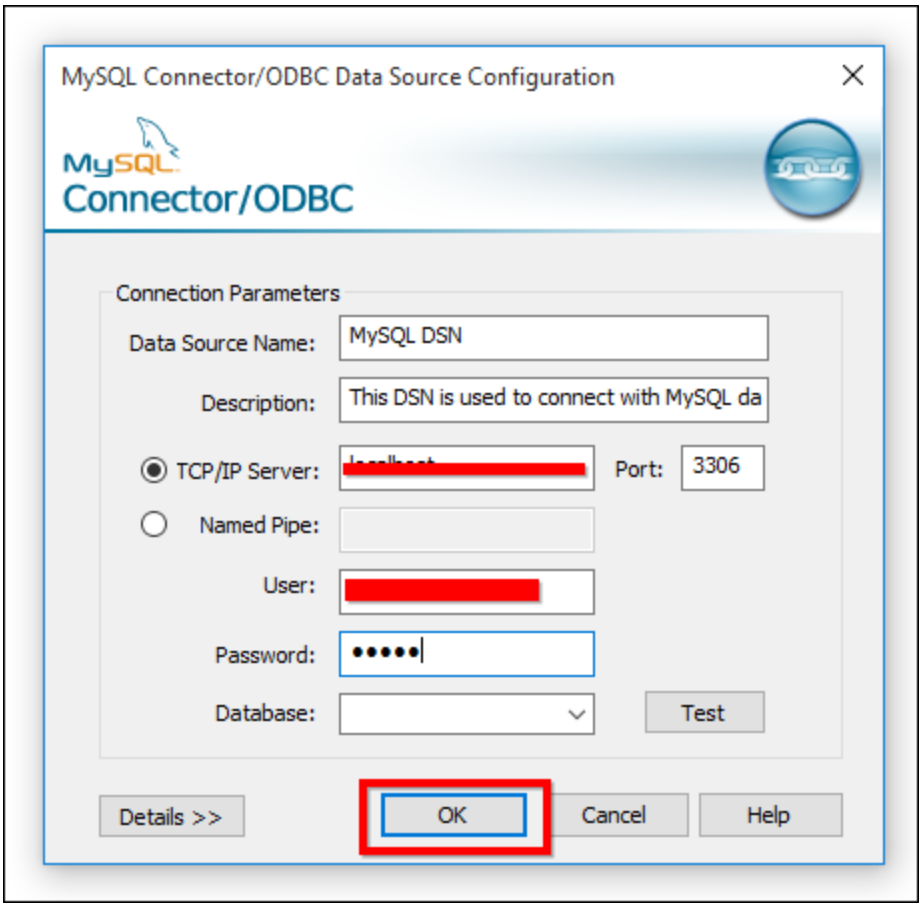
In the ODBC Datasource Administration (32-bit) Tool navigate to System DSN and click on Add to add a new DSN.



Select MySQL ODBC Driver from the list of drivers and then click on Finish.

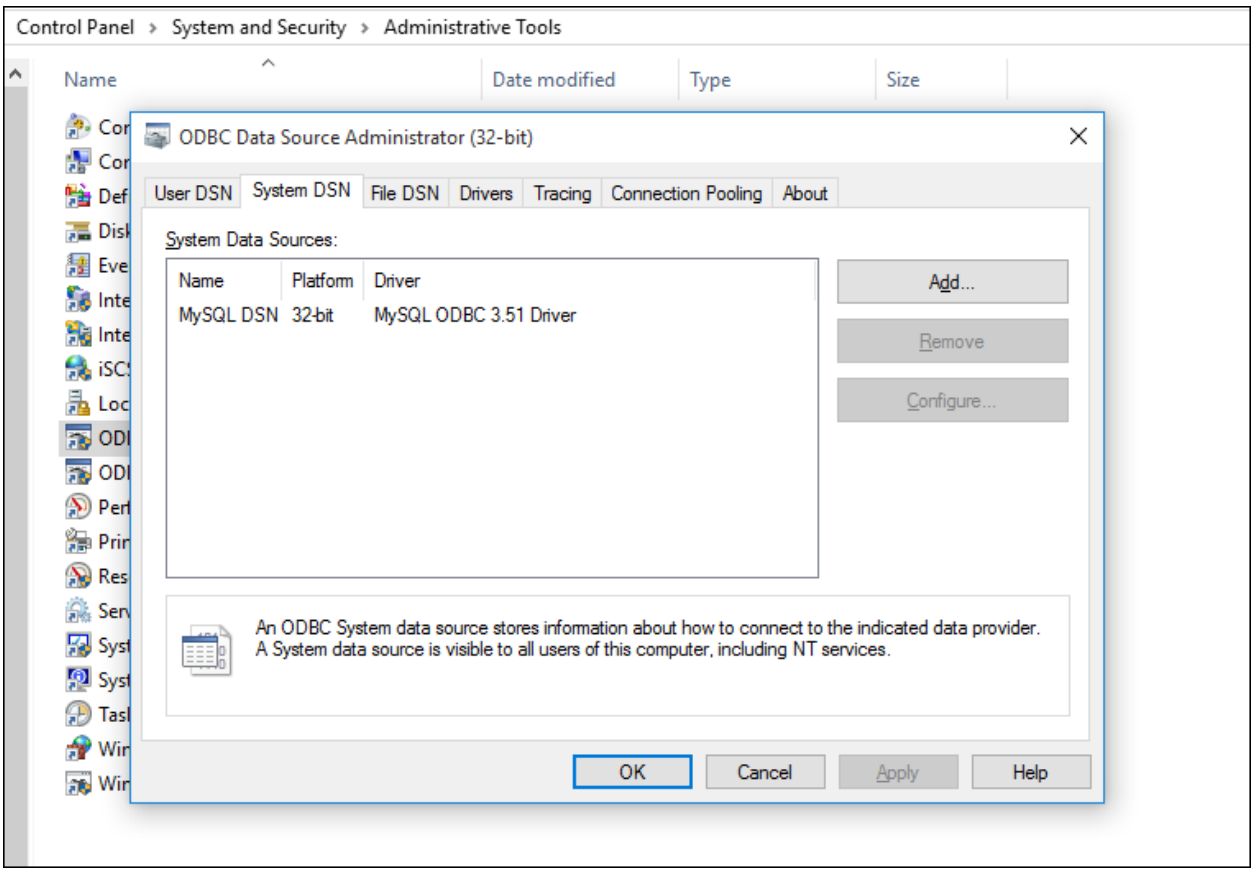


The MySQL Connector/ODBC Configuration tool will be opened and fill the requested details in the tool.



- Data Source Name – Name of the DSN that will be used to connect MySQL database from Syncfusion Dashboard Server.
- Description – Description of the DSN(Optional)
- TCP/IP Server – Server that holds MySQL Database.
- Port - Port number provided for MySQL Database.
- User – Username that used to connect MySQL.
- Password - Password of the corresponding user.

The created DSN will be listed in the ODBC Data Source Administration (32-bit) tool.



How to Create DSN for Oracle

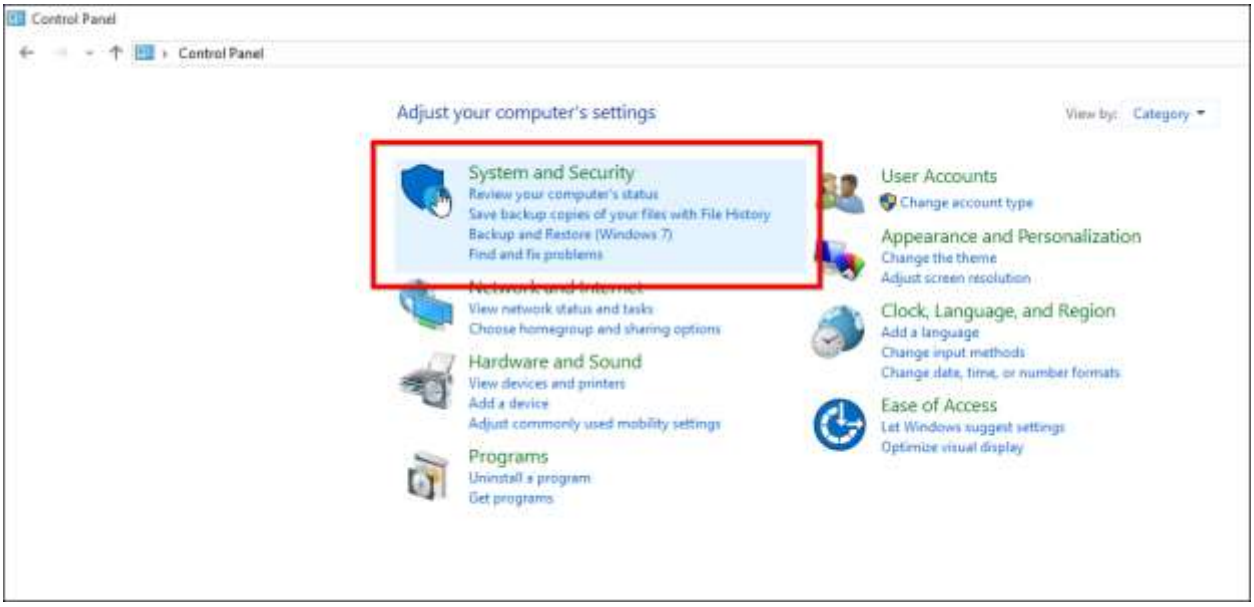
This section explains on how to create DSN for Oracle.

An ODBC Data Source Name (DSN) stores information for establishing a connection to a database on a remote database server. A system DSN provides access to multiple users, rather than only the user who created it.

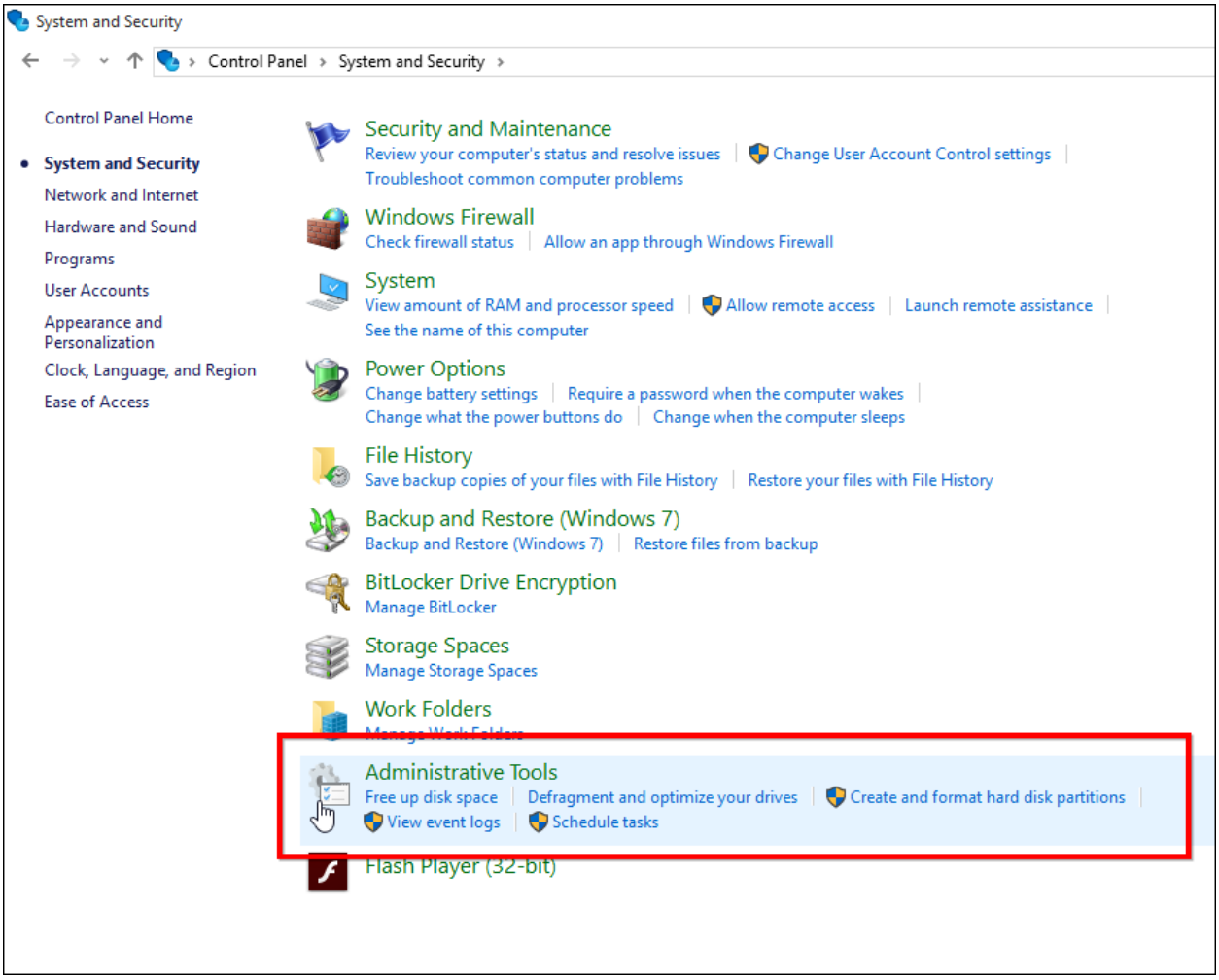
Steps to create DSN for Oracle

Open ODBC Data Source Administration Tool

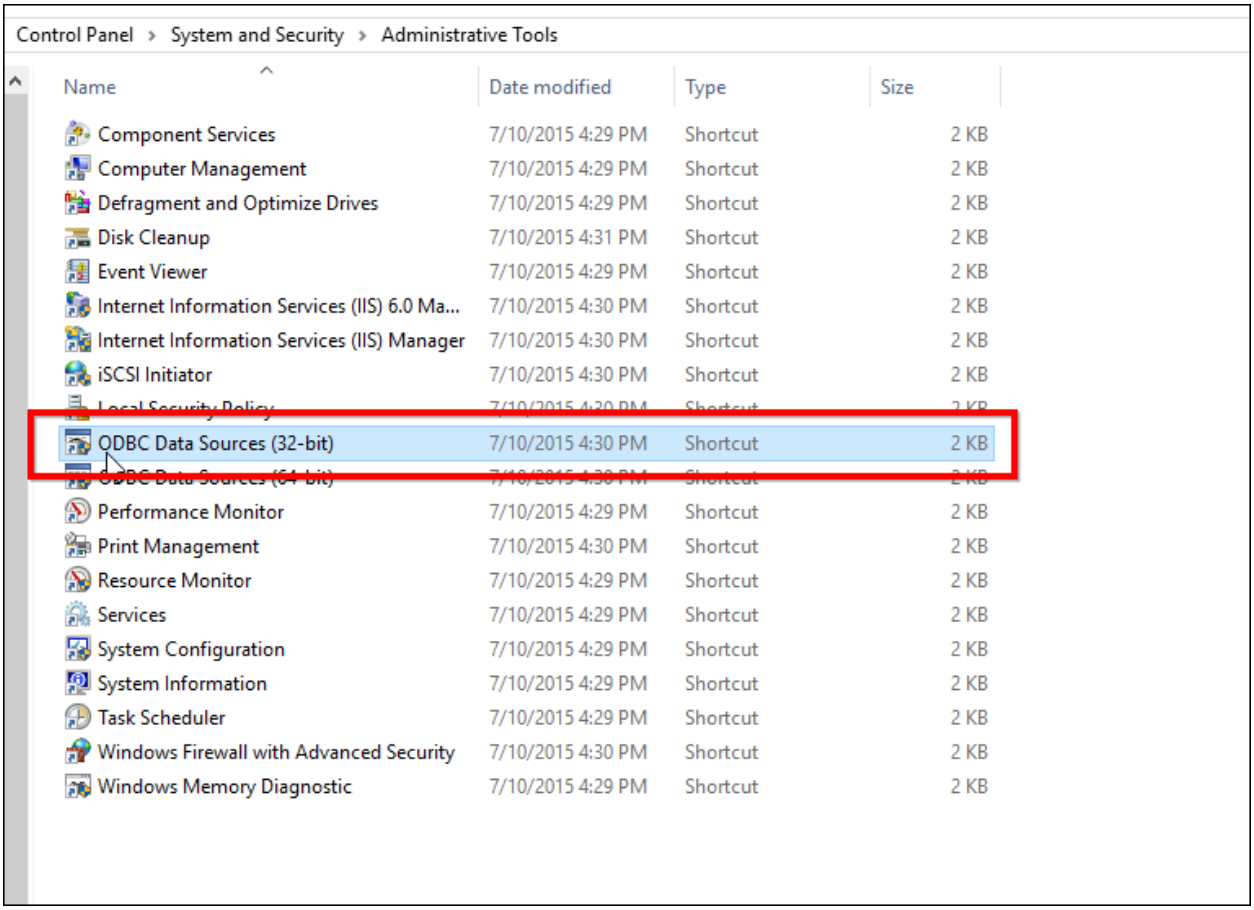
Open Control Panel and select System and Security.



Select Administrative Tools from the list of options.

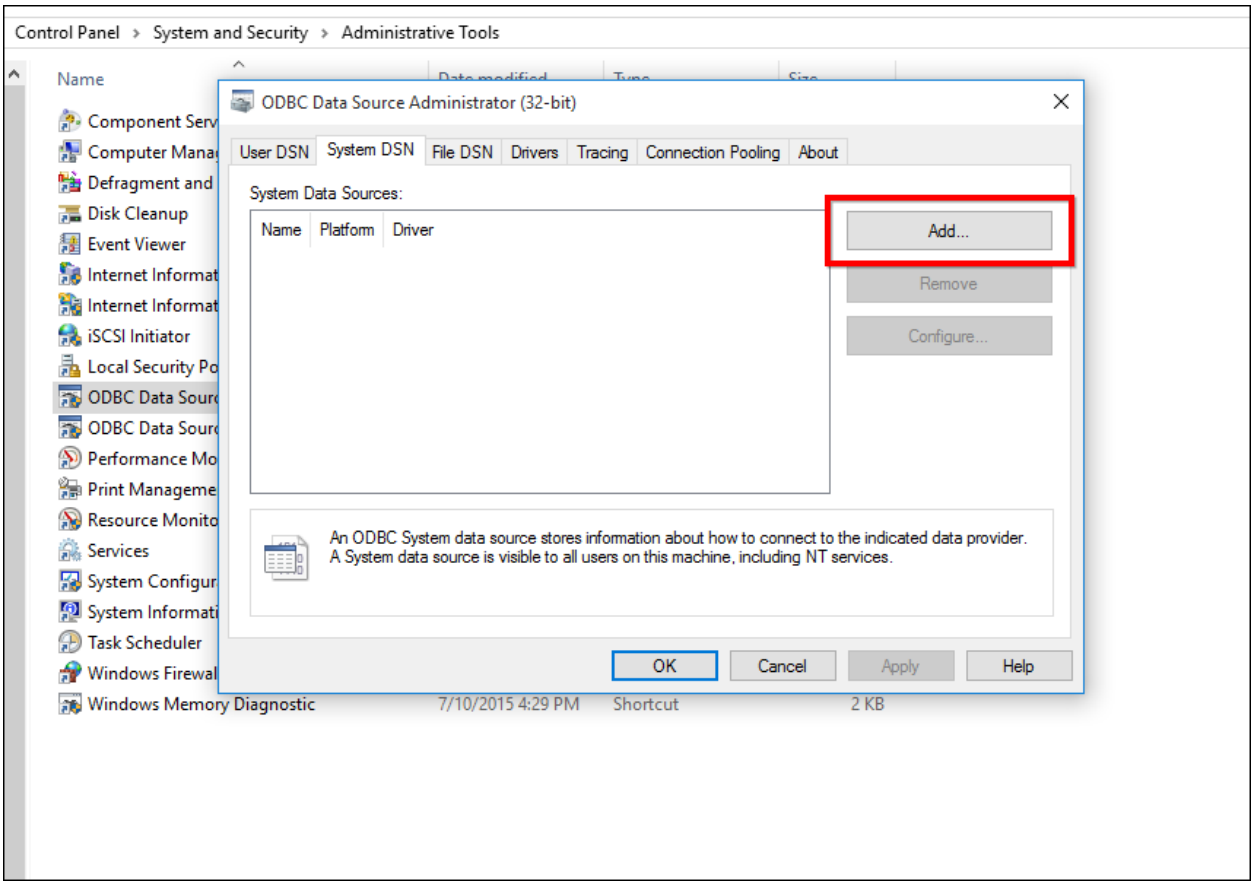


Select ODBC Data Sources (32-bit) from the list of options.

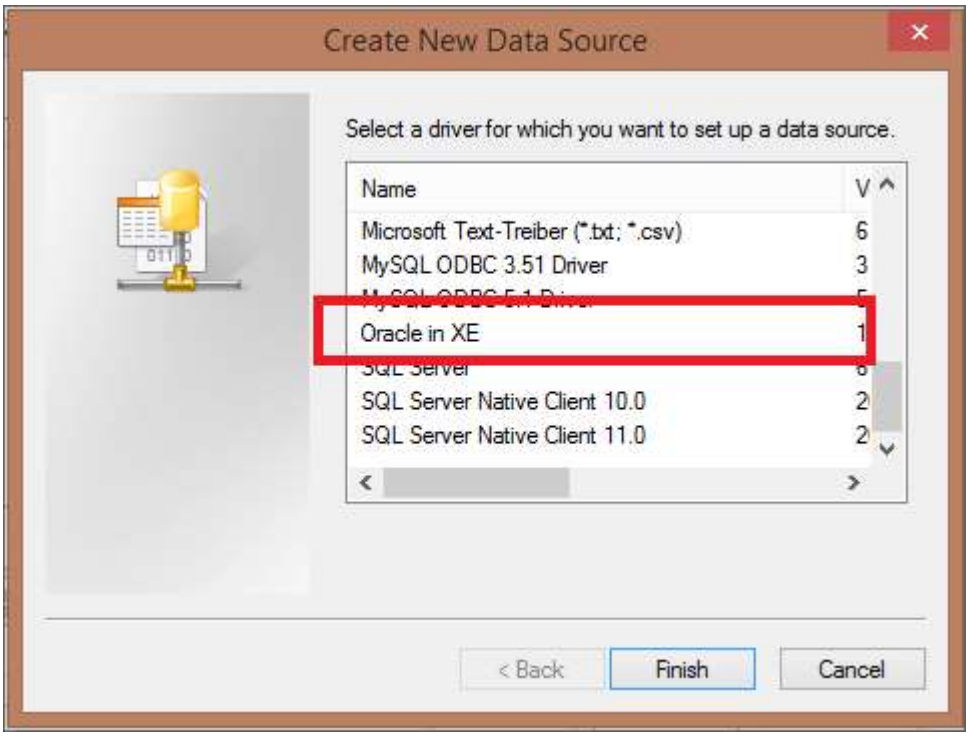


Create DSN with Oracle ODBC Driver

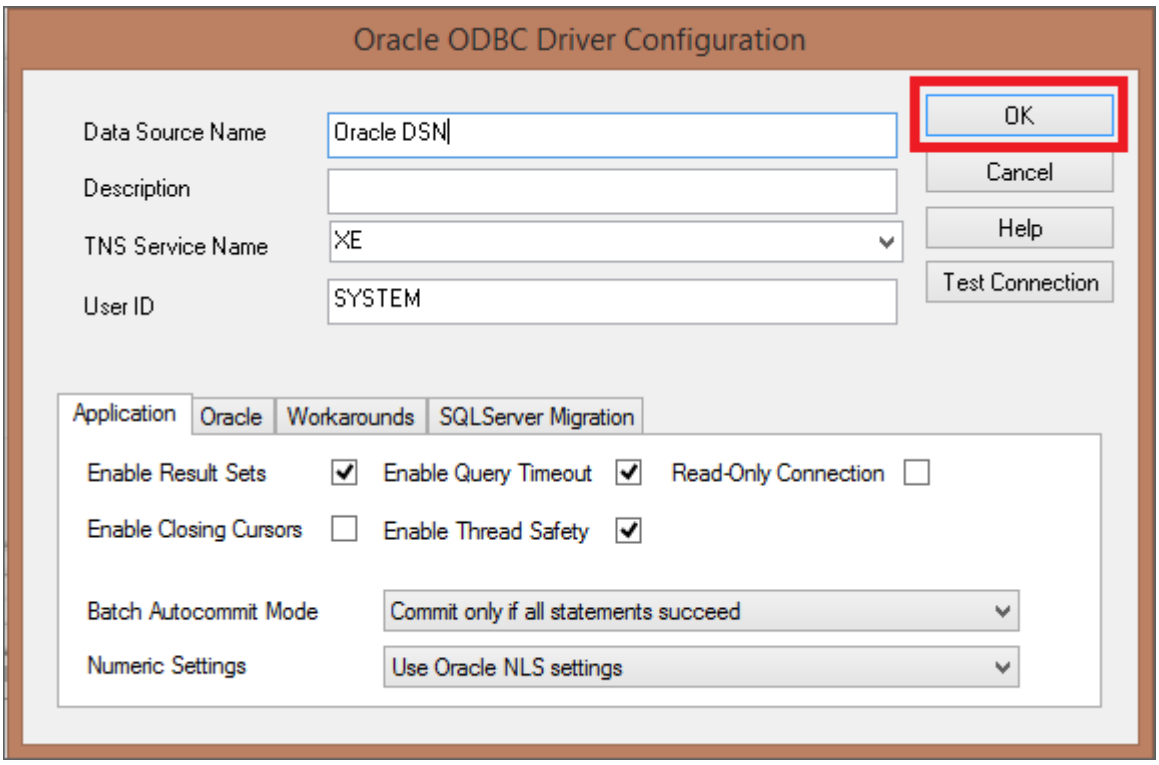
In the ODBC Datasource Administration (32-bit) Tool navigate to System and click on Add to add a new DSN.



Select the Oracle Driver from the list of drivers and then click on Finish.



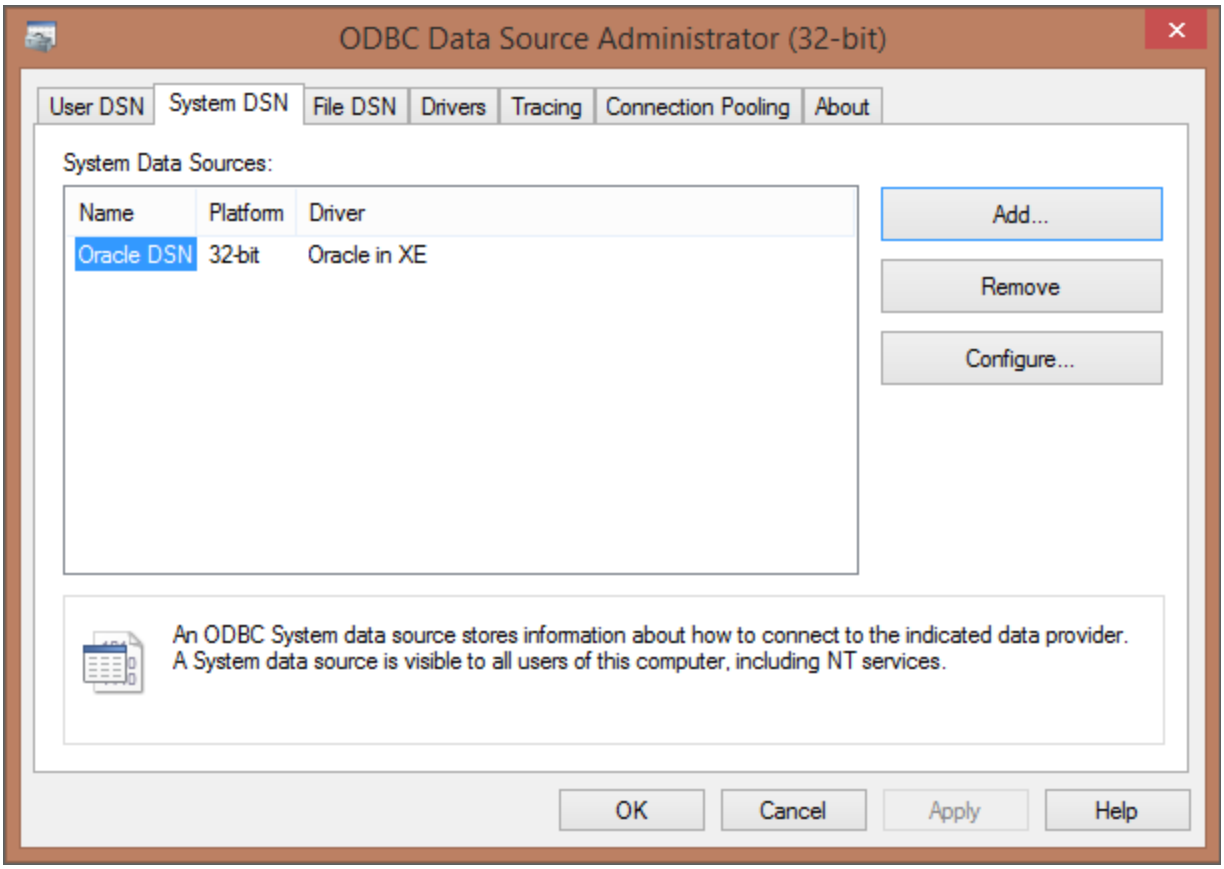
Click Finish. An ODBC driver setup window opens. Fill the requested details in the tool.



- Data Source Name – Type a name to display in the DSN field on the Dashboard Server application for the Oracle database type.
- Description – Description of the DSN(Optional)
- TNS Service name – Select the TNS Service Name for the database your workspace repositories will be stored in. If no choices are shown, or if you are unsure which name to select, contact your DBA.
- User ID – Enter the database User ID.

Note: While initializing application, the Username and Password given in the start up page will overwrite the Username and Password of the DSN.

The created DSN will be listed in the ODBC Data Source Administration (32-bit) tool.



How to Set up Azure Active Directory to perform authentication using Single Sign-On

This section explains how to perform Single Sign-On (SSO) for users in the Azure Active Directory with Syncfusion Dashboard Server and Dashboard Designer.

Note: This configuration has been done using the Azure Portal.

Steps to set up Azure Active Directory

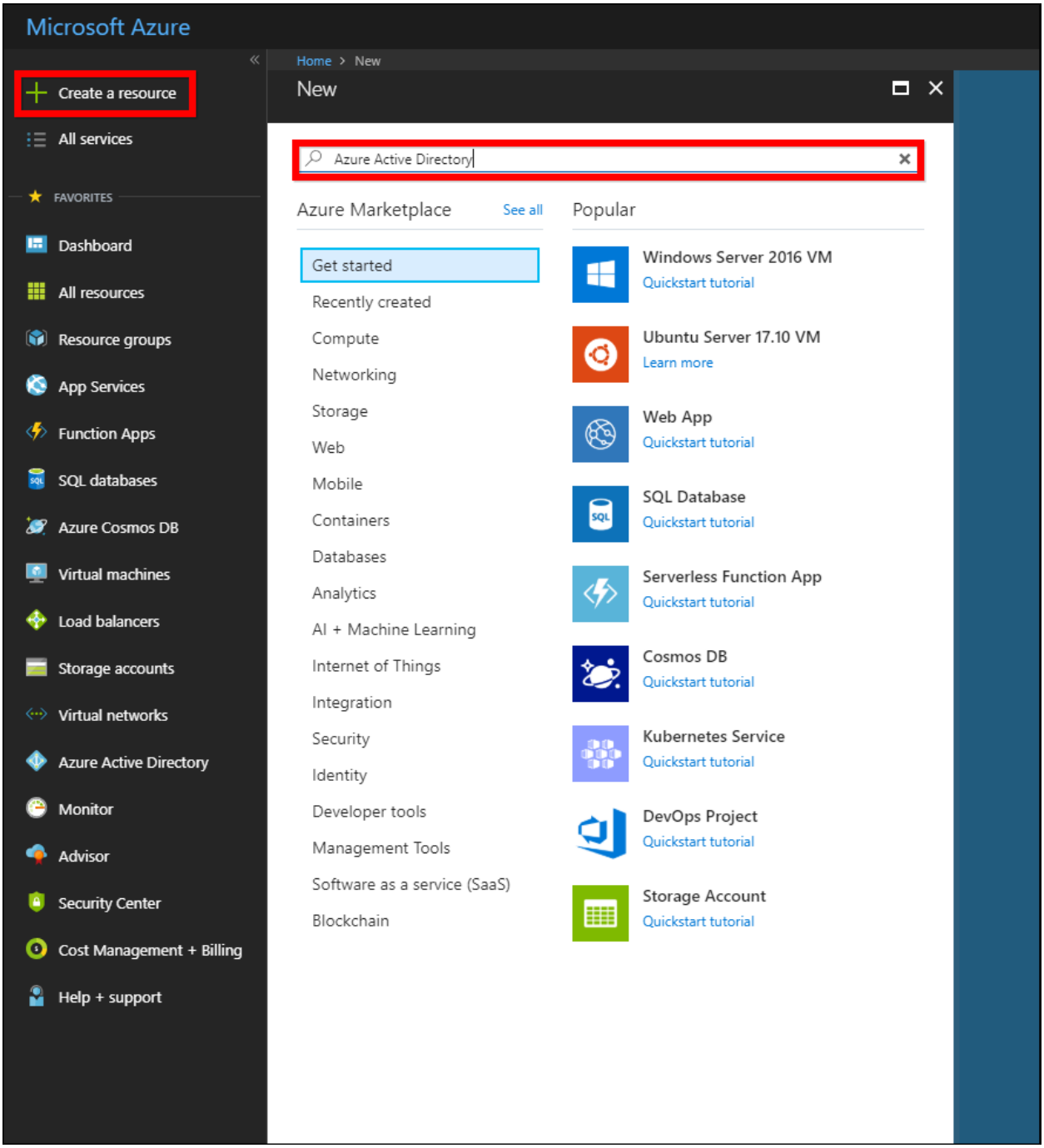
Prerequisites

- An Azure account with Active Directory support.
- Install Syncfusion Dashboard Server and log on with administrator account.

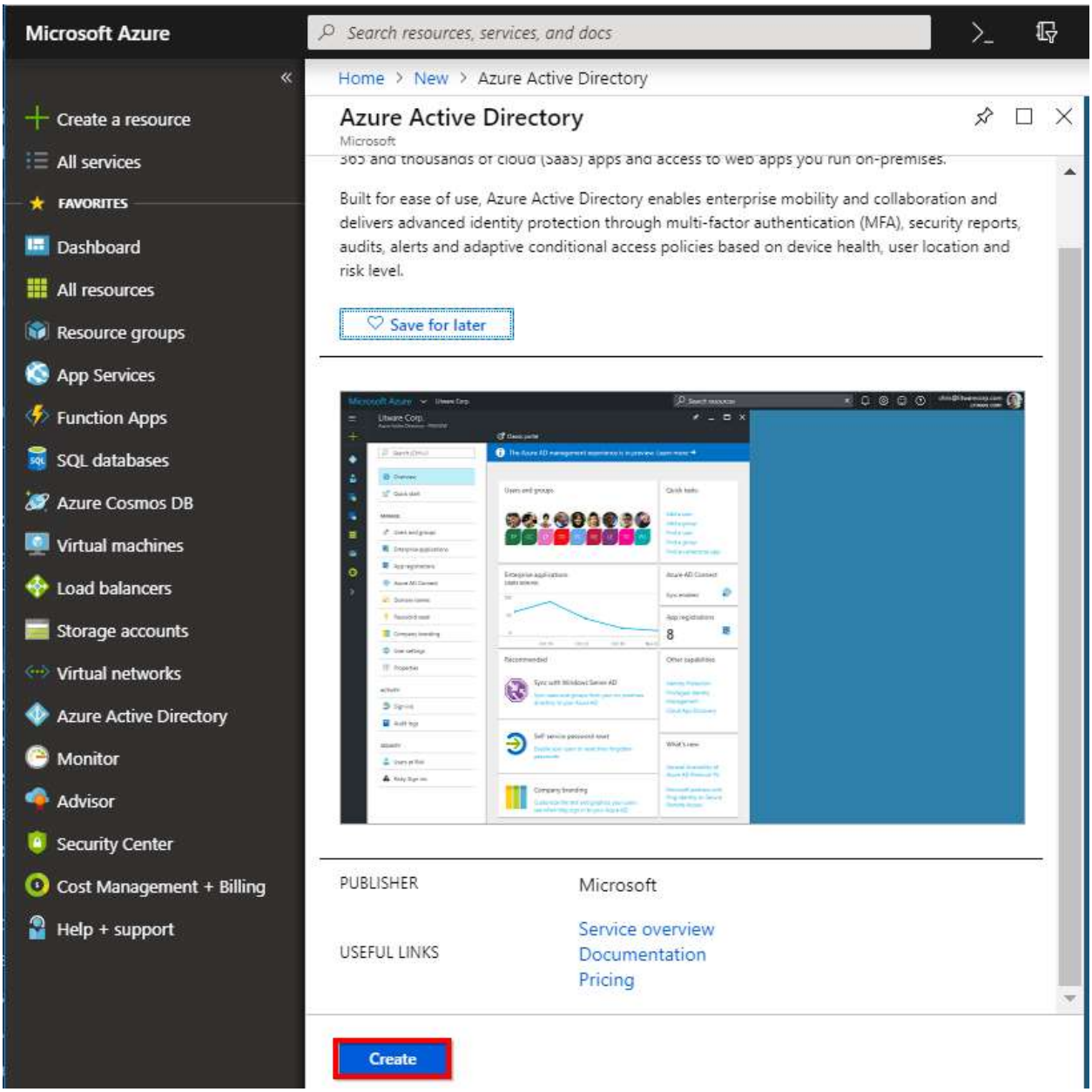
Setup Azure Active Directory application

Log on to the Azure portal to create an Azure Active Directory.

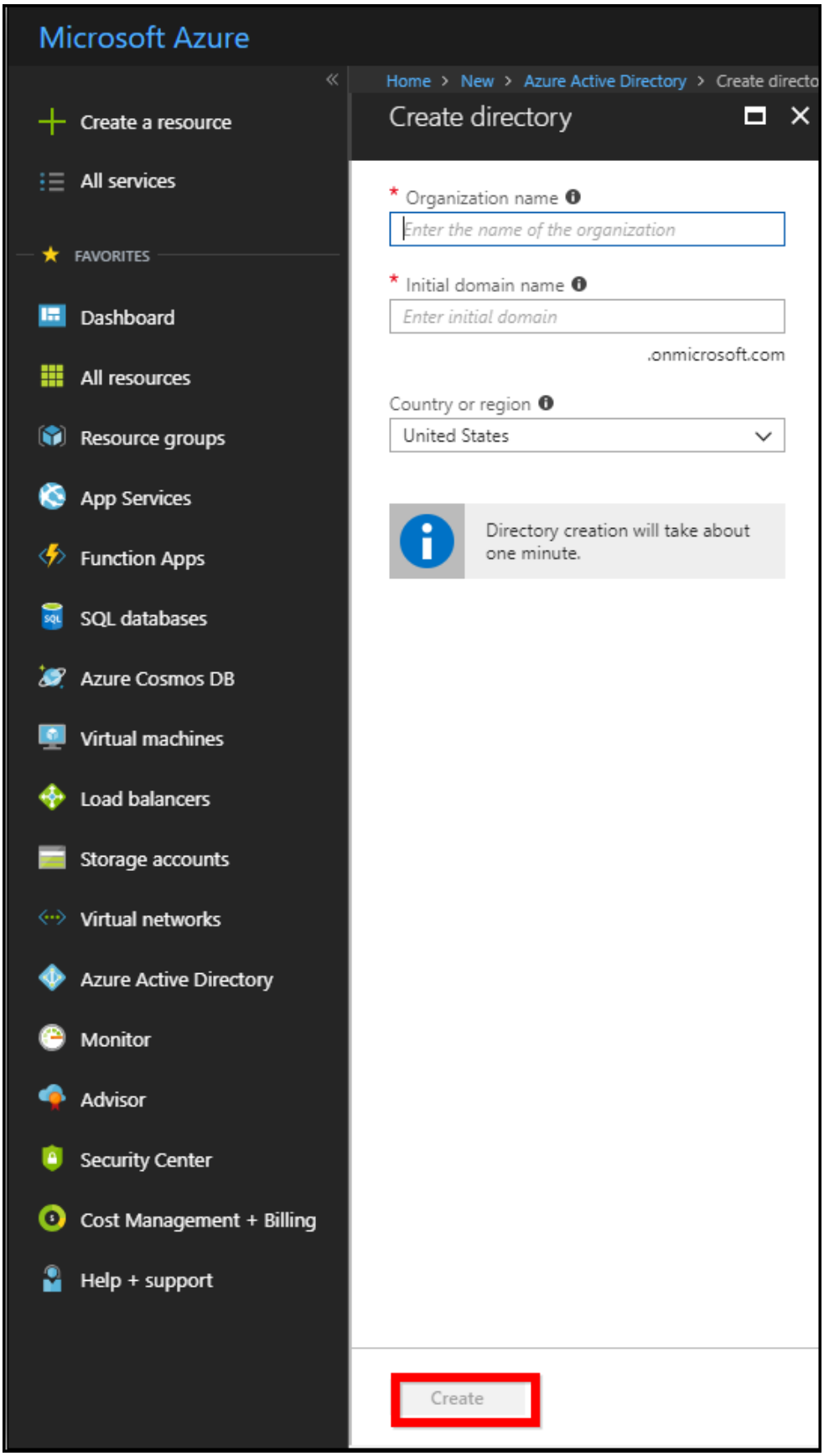
1. Click Create a resource and search Azure Active Directory as follows.



2. Click Create in the following screenshot.



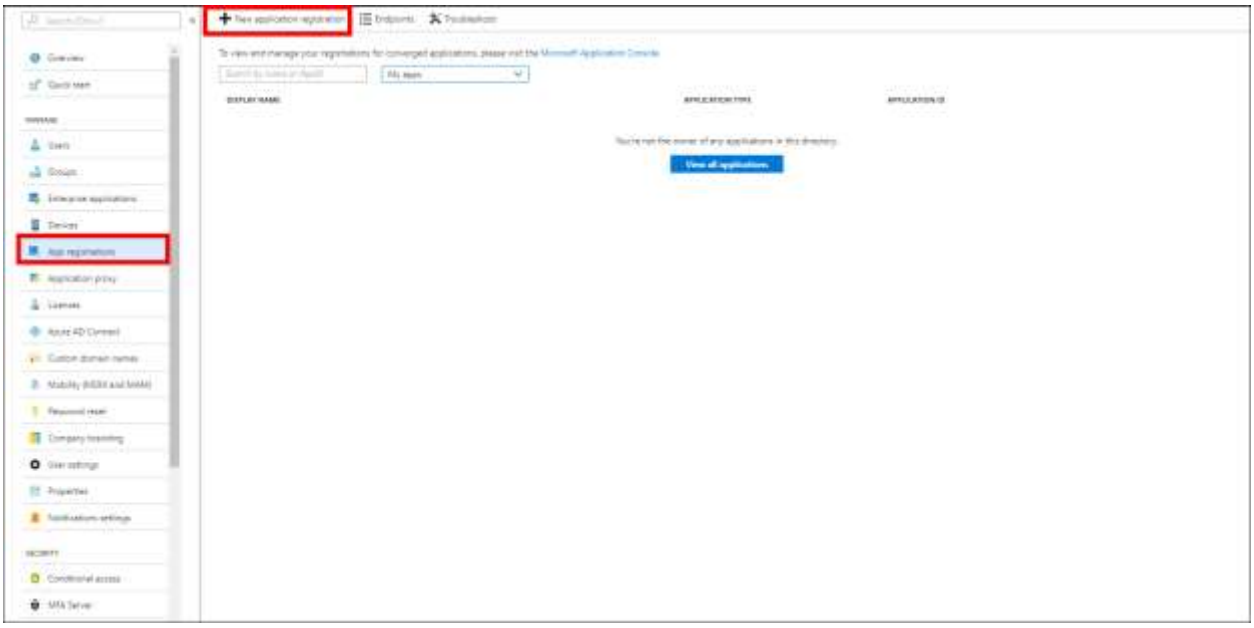
3. In the dialog box, enter the Name, Domain Name, and choose the Country or Region, and then click Create.


Microsoft Azure
Create a resource
All services
FAVORITES
Dashboard
All resources
Resource groups
App Services
Function Apps
SQL databases
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
Azure Active Directory
Monitor
Advisor
Security Center
Cost Management + Billing
Help + support
Home > New > Azure Active Directory > Create directo
Create directory
* Organization name
Enter the name of the organization
* Initial domain name
Enter initial domain
.onmicrosoft.com
Country or region
United States
Directory creation will take about one minute.
Create

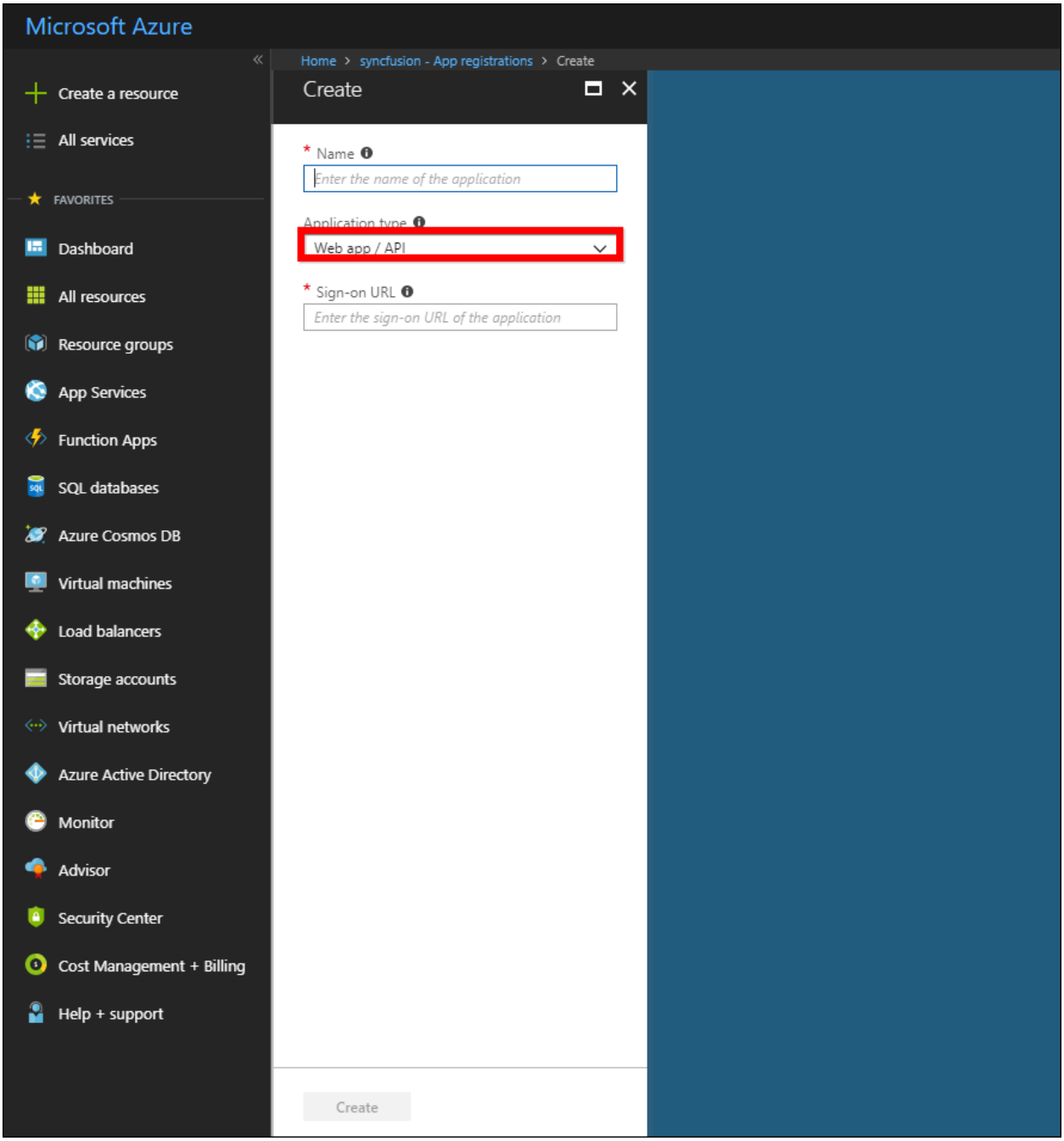
The application will be added to the directory and you can view the details of the application in the App registrations.

In this directory, you should add three applications. An application acts as a Web API Server (Dashboard Server), and the other two applications act as native client applications (Dashboard Designer and Syncfusion Dashboards mobile app).

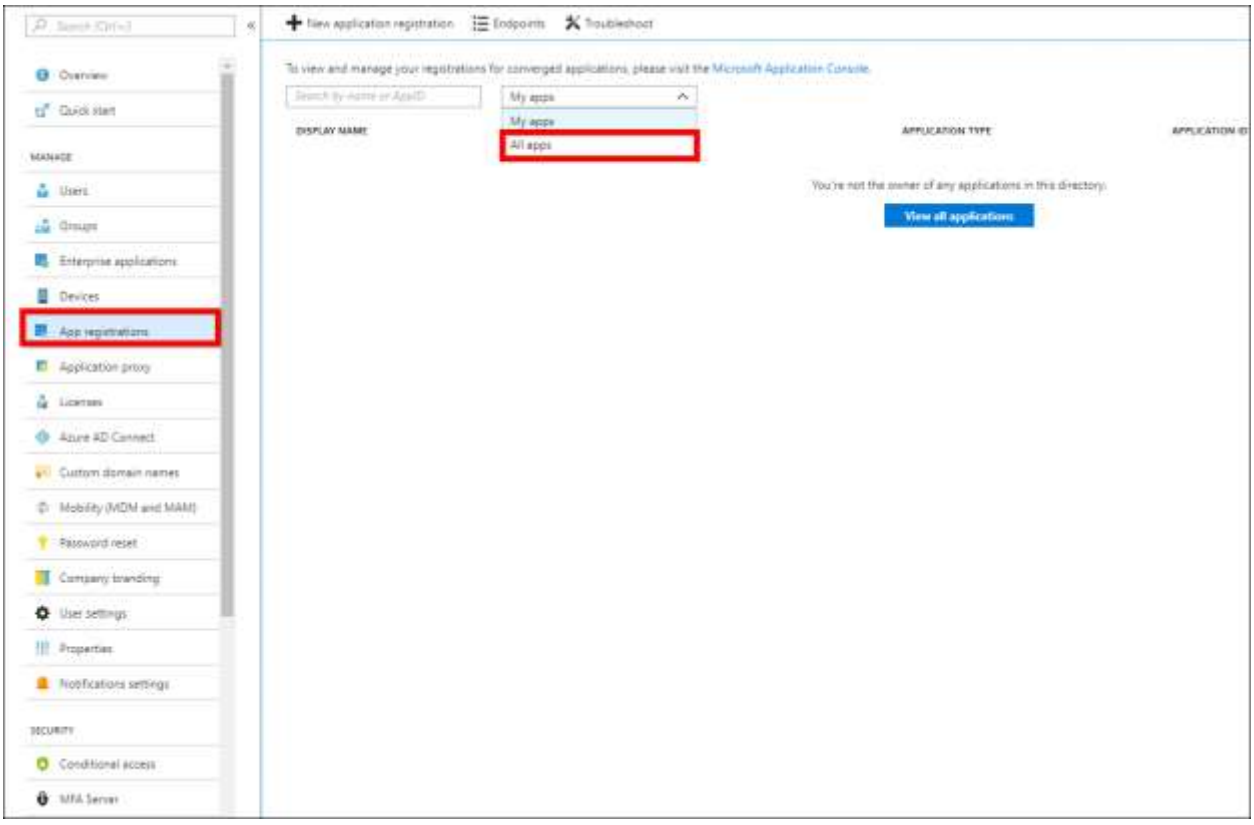
4. Enter into the created directory and click Azure Active Directory, and then select App registrations.
5. Now, click New application registration to add a new application.



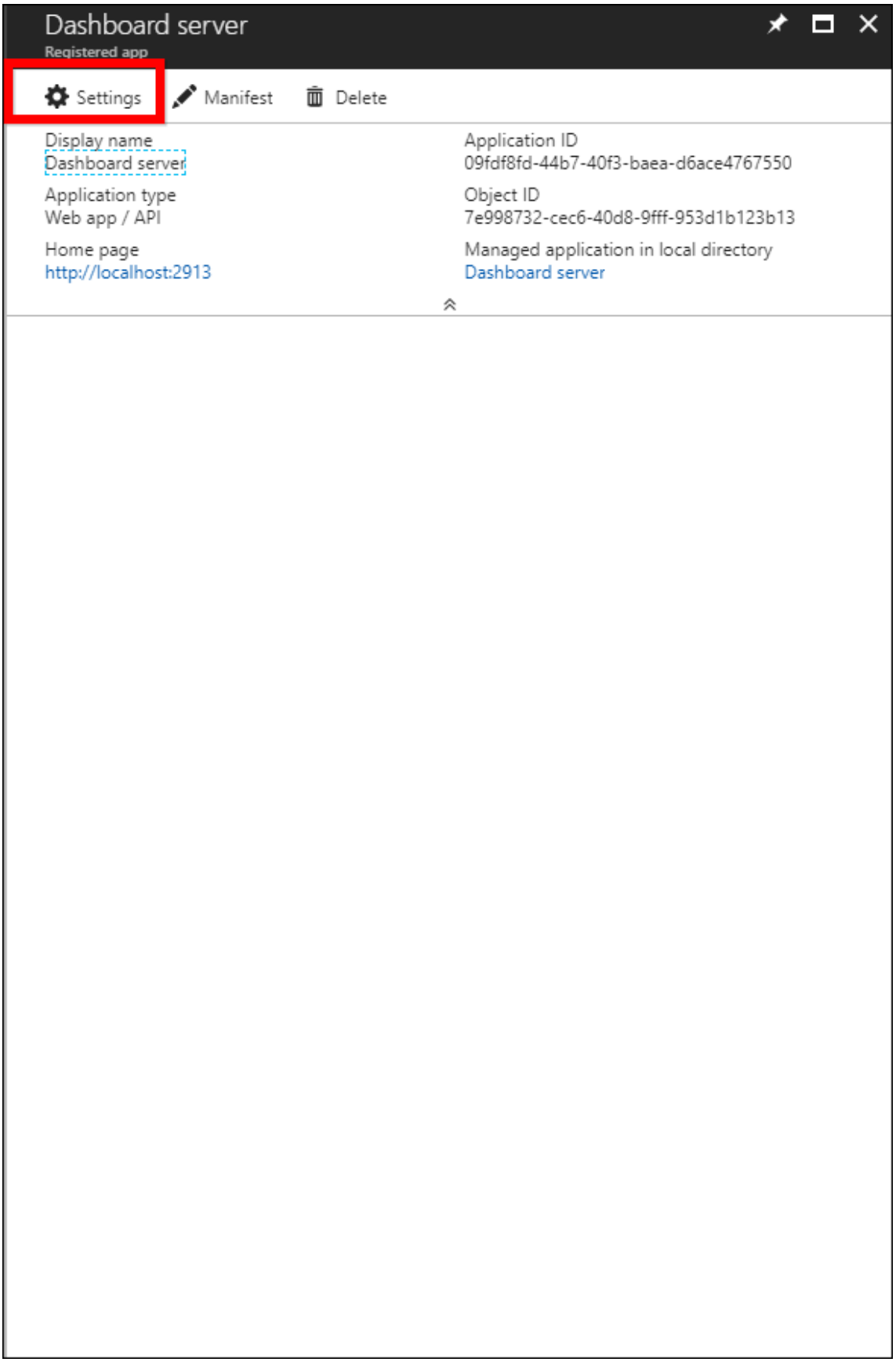
6. Enter the name of the application and choose Web app/API as Application type and enter the sign-on URL.



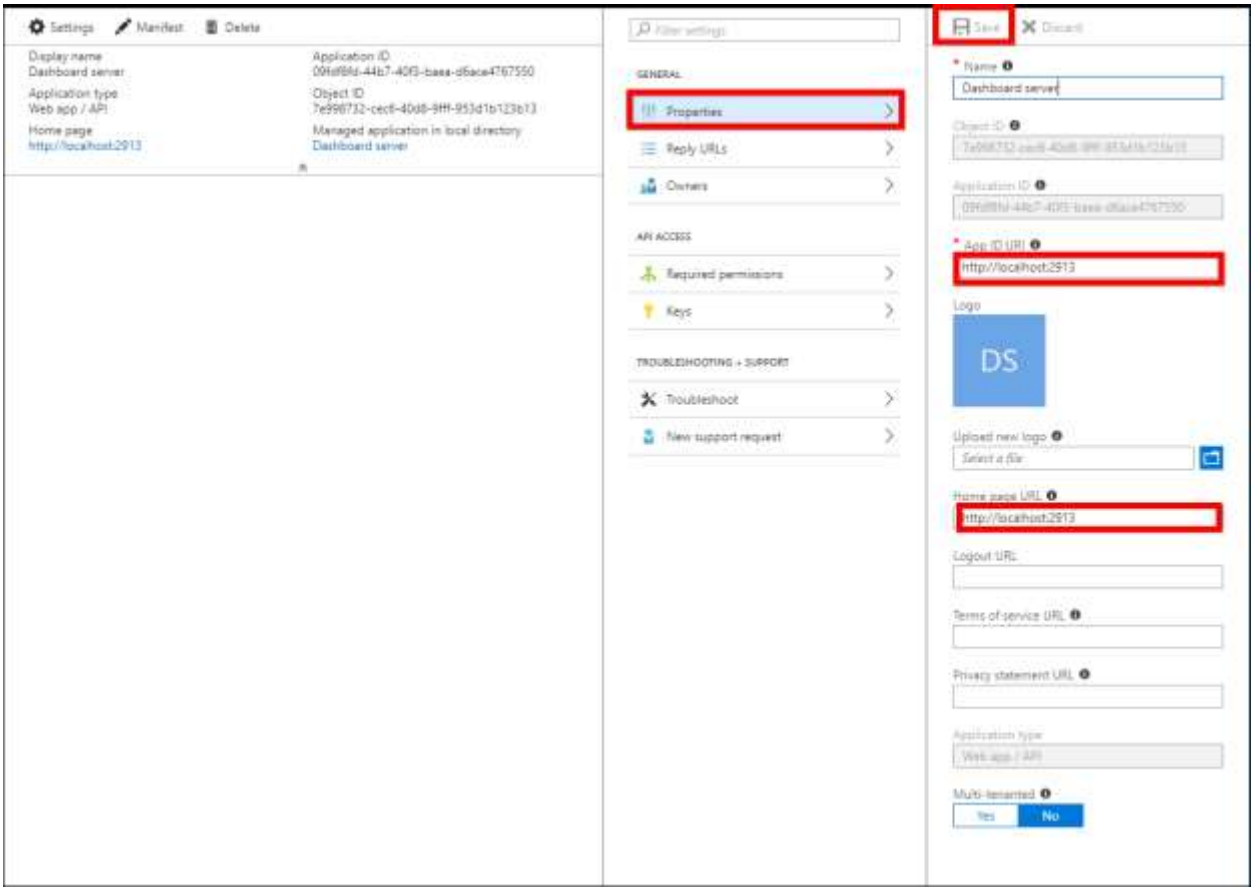
7. To view all registered applications, click Azure Active Directory and select App registrations, and then choose All apps.



8. Choose the registered application and click Settings.



9. Select Properties and enter the App Id URI and Home page URL.

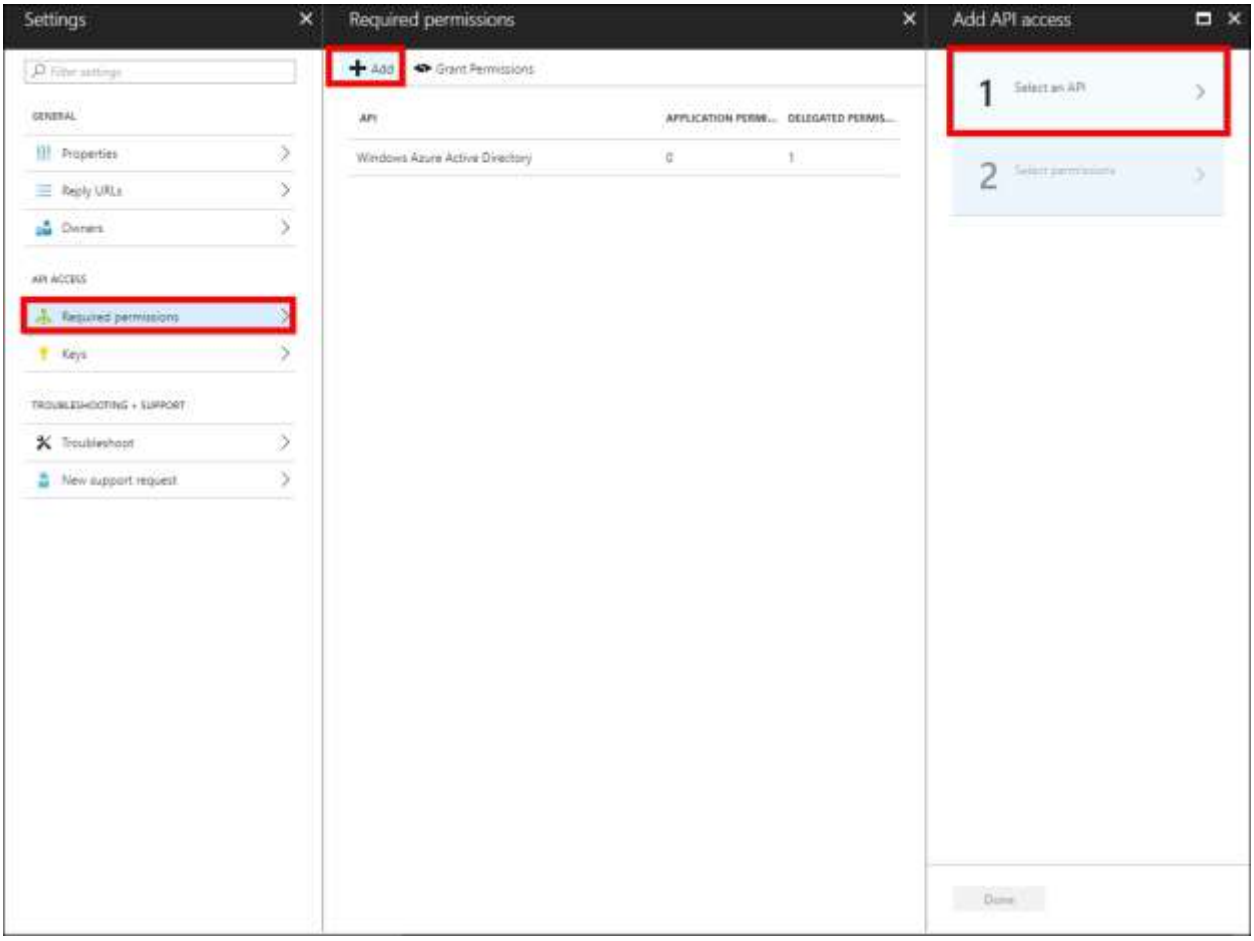


10. Click Save as highlighted in the above screenshot.

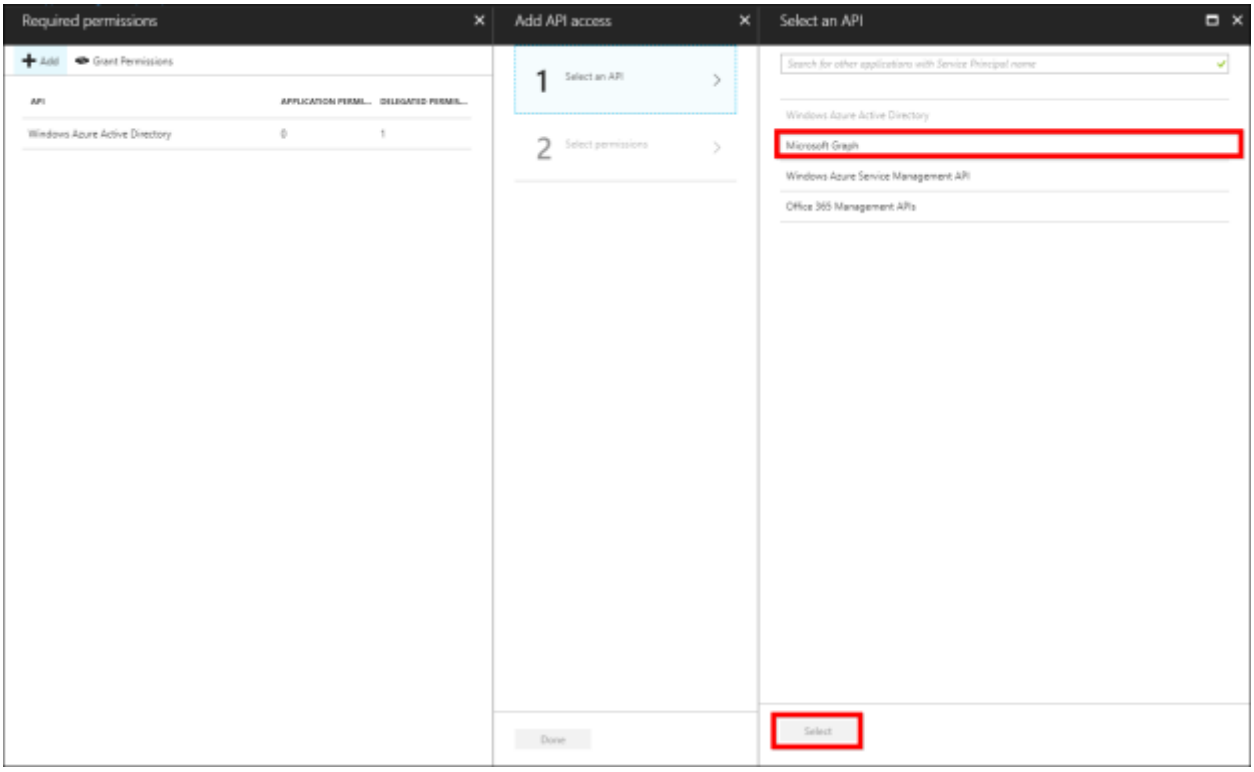
Note: The sign-on URL, App ID URI, and Home page URL should be the URL of the Syncfusion Dashboard Server application.

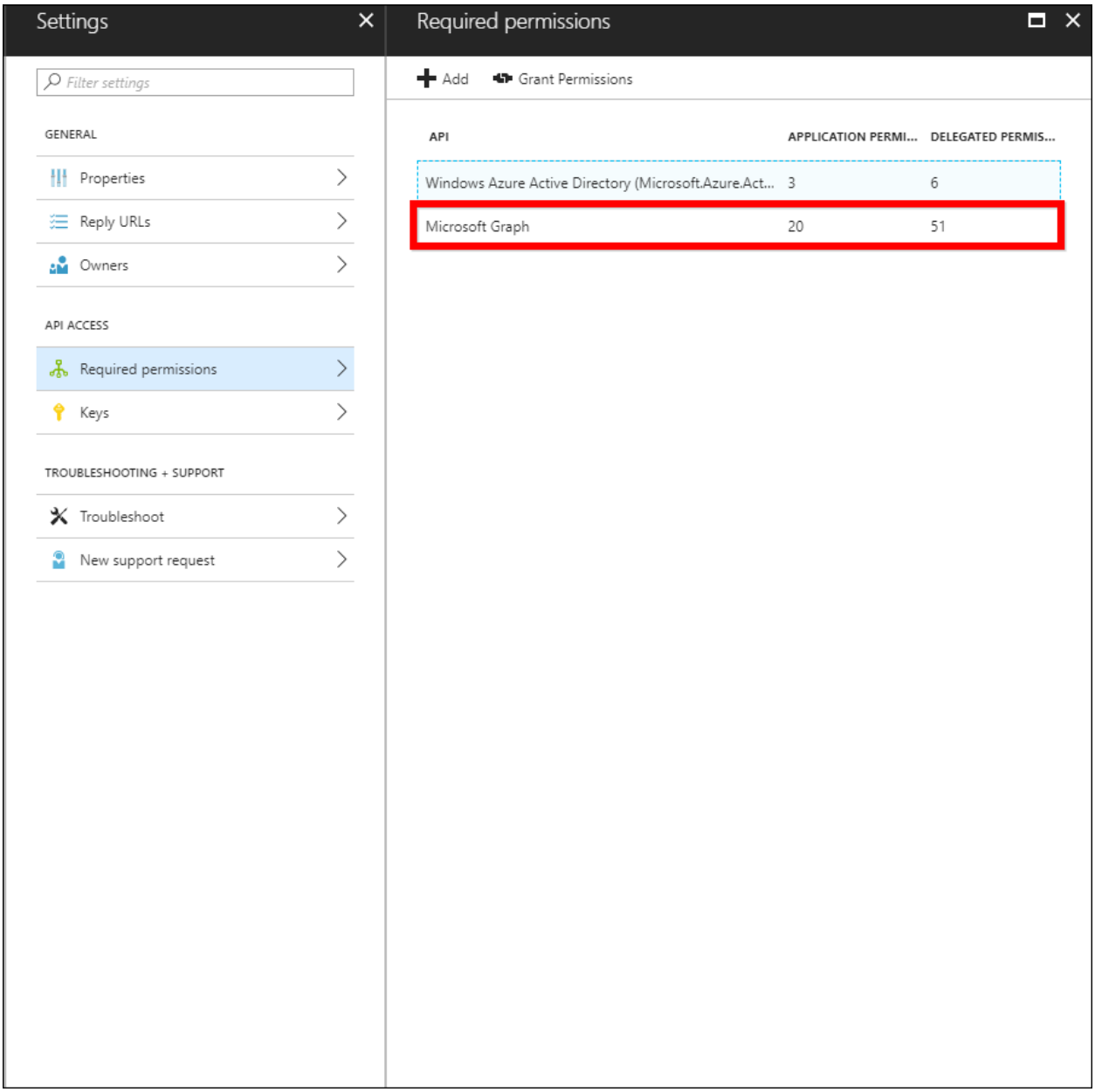
Now, you can add Microsoft Graph application to your application to import the users and groups into the Syncfusion Dashboard Server.

11. Go to application, click Settings, and select the Required permissions. Then click Add and click Select an API.



12. Select Microsoft Graph from the list and click Select.


Required permissions
Add
Grant Permissions
API
APPLICATION PERMI...
DELEGATED PERMI...
Windows Azure Active Directory
0
1
Add API access
1
Select an API
2
Select permissions
Done
Select an API
Search for other applications with Service Principal name
Windows Azure Active Directory
Microsoft Graph
Windows Azure Service Management API
Office 365 Management APIs
Select



13. Enable following permissions for dashboard server application
    - Microsoft Graph Application and Delegated Permissions

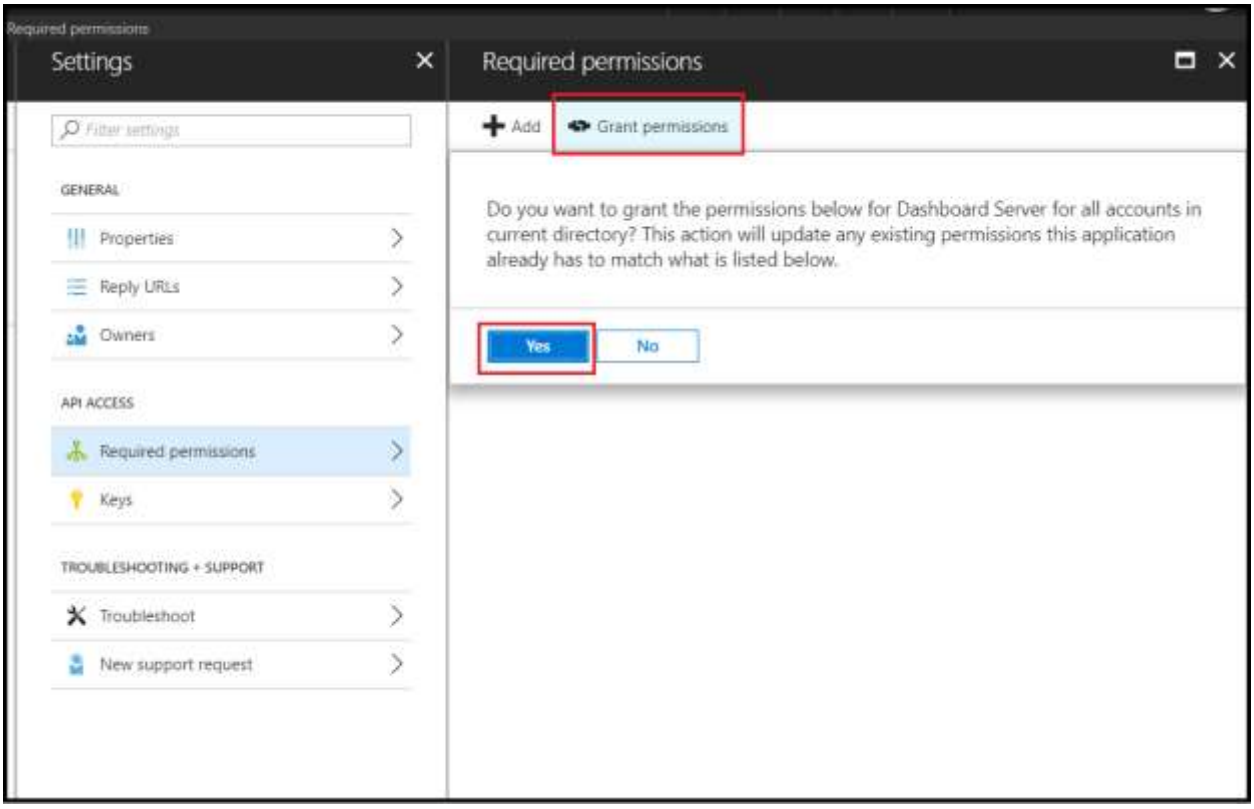
Application Permissions
Read directory data
Delegated Permissions
1. Read directory data
2. Read all groups

3. Sign in and read user profile
4. Access directory as the signed in user

- Windows Azure Active Directory Application and Delegated Permission

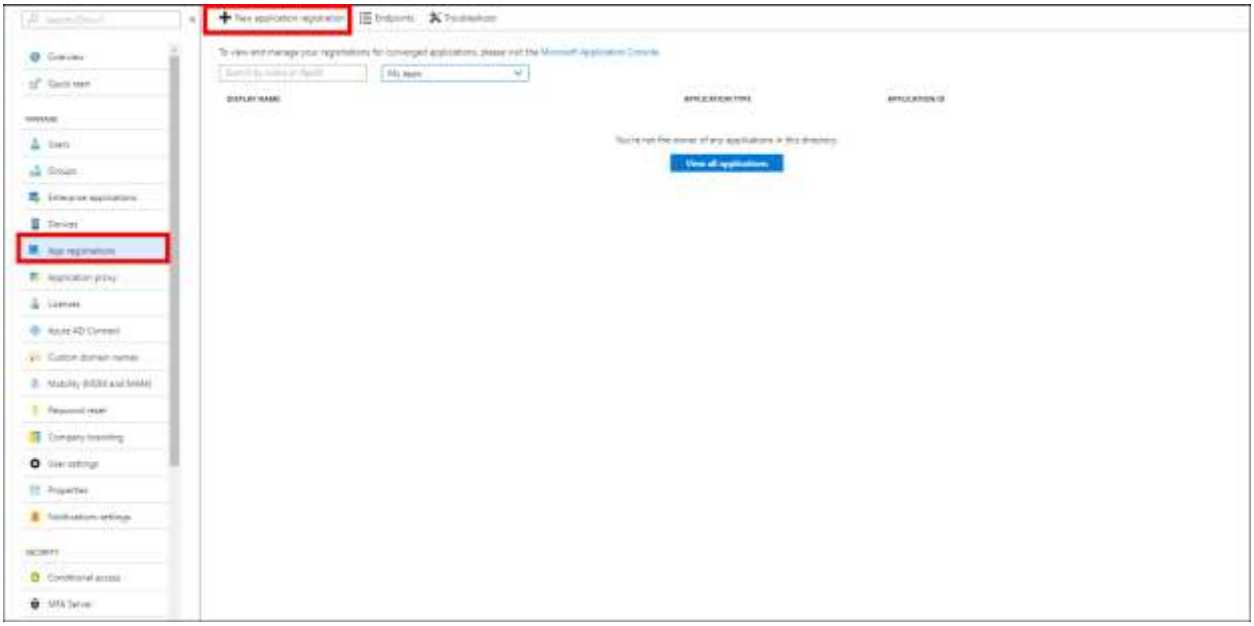
Application Permissions
Read directory data
Delegated Permissions
1. Read directory data
2. Sign in and read user profile

14. After adding the permission, click the Grant Permission from the Required permissions section of the application page and select yes as below.

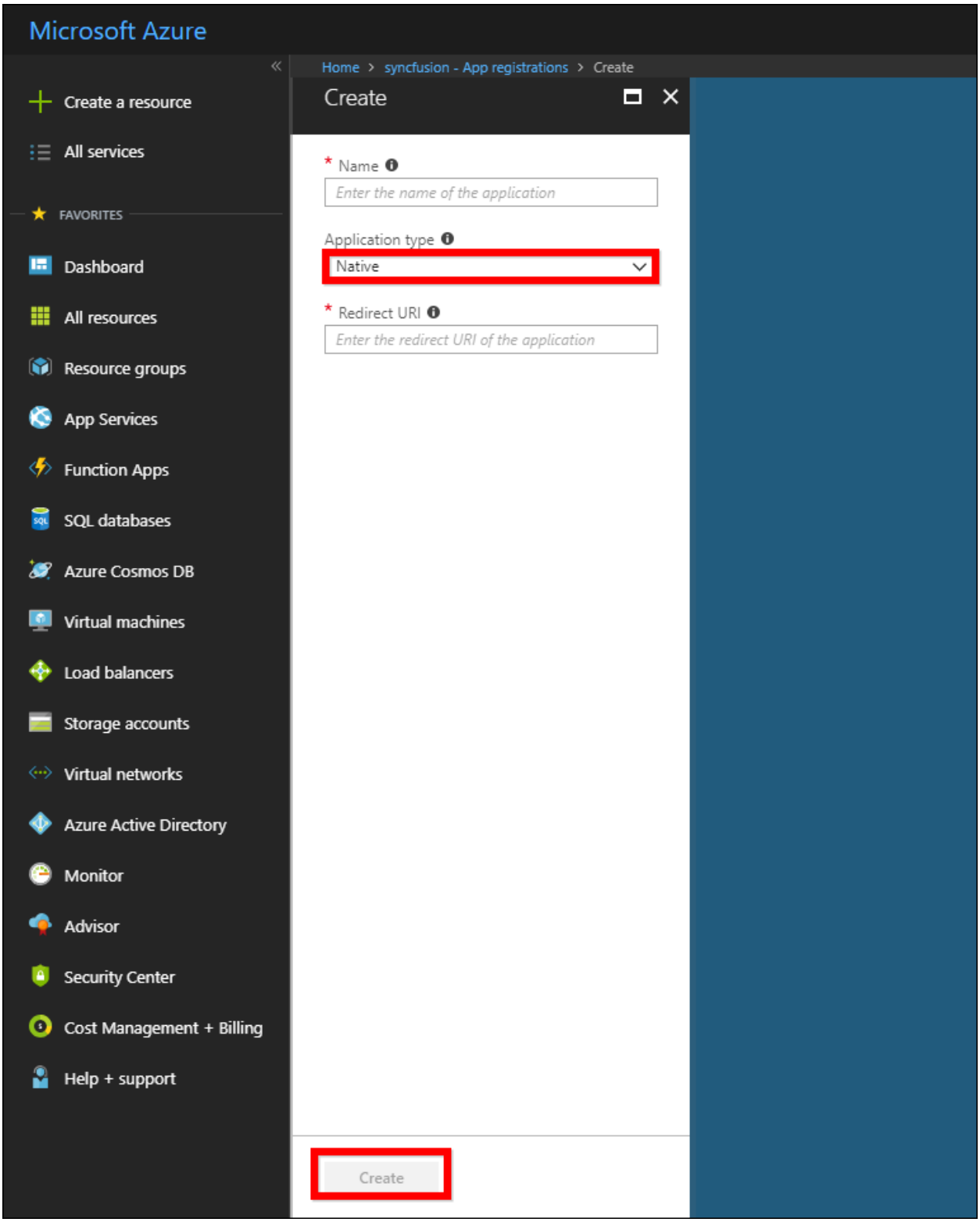


Configure Azure Active Directory to perform Single Sign-On in Dashboard Designer application

1. Enter into the created directory and click the Azure Active Directory. Then, select App registrations and click the New application registration to add a new application.



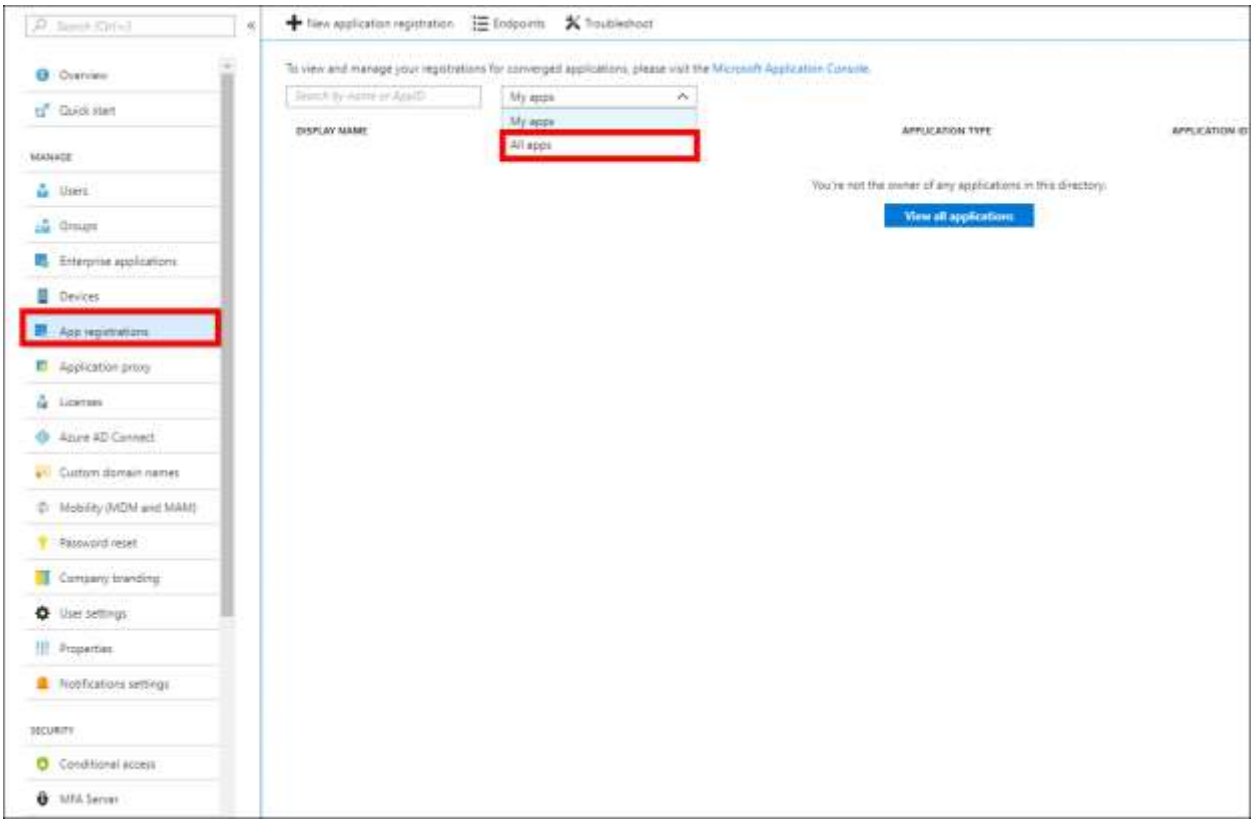
2. Enter the name of the client application and choose the Application type as Native and enter the Redirect URI.



3. Click Create. The client application will be added to the directory and you can view the details of the application in the App registrations.

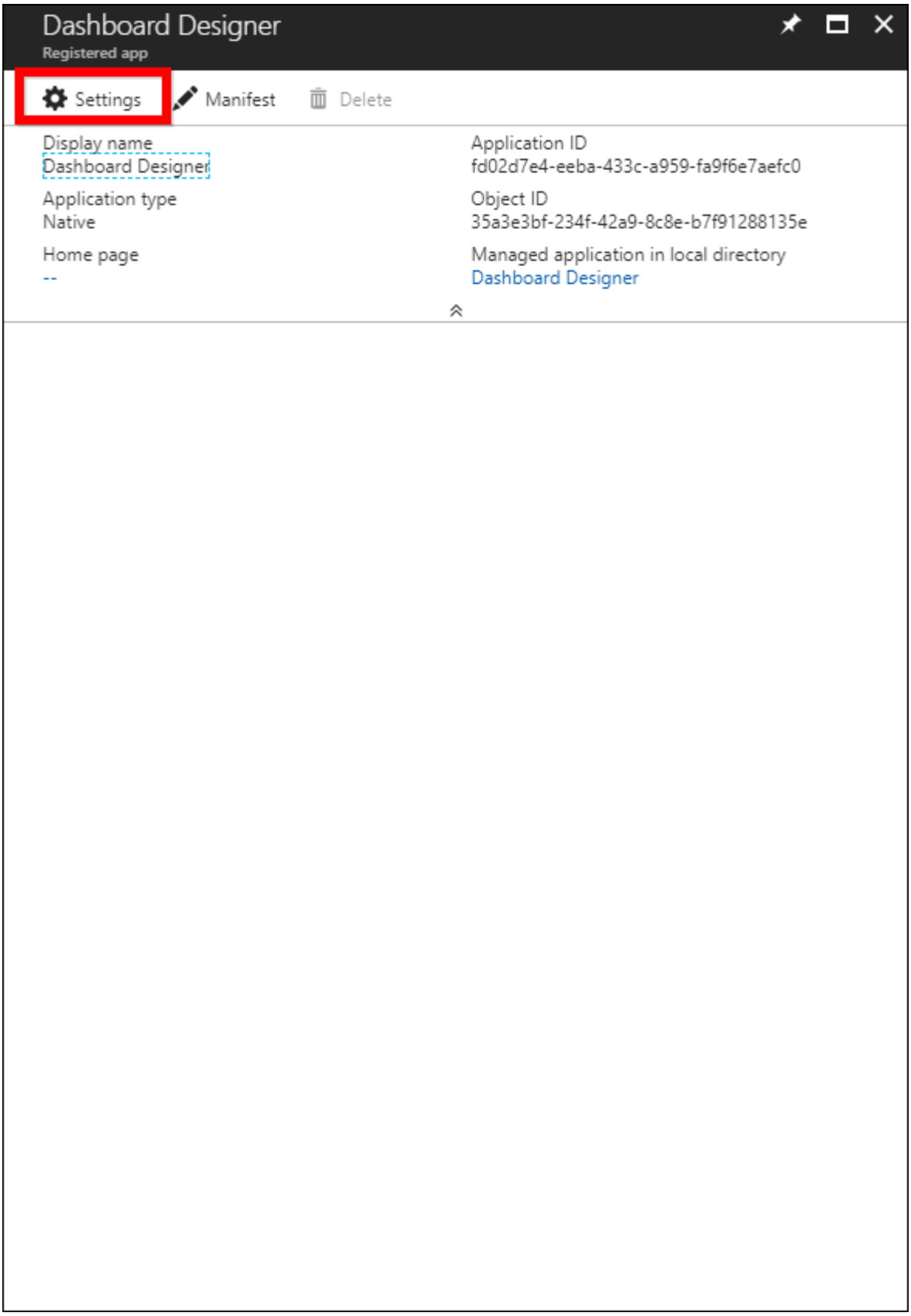
Note: Redirect URI should be the URL of the Syncfusion Dashboard Server application.

4. To view all registered applications, click the Azure Active Directory and select App registrations, and then choose All apps.

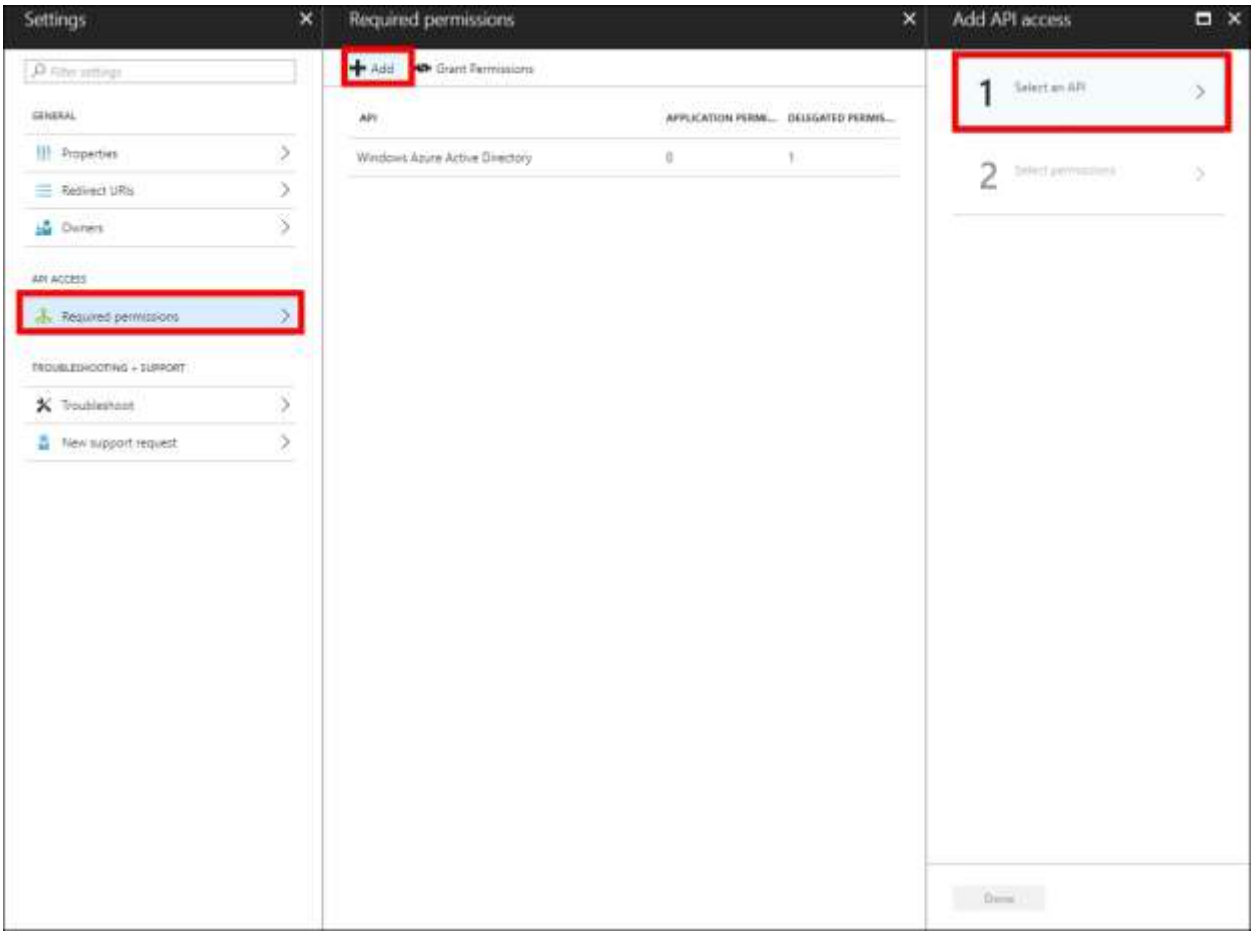


Now, you can add Dashboard Server application to the client application to enable Single Sign-On in native client application.

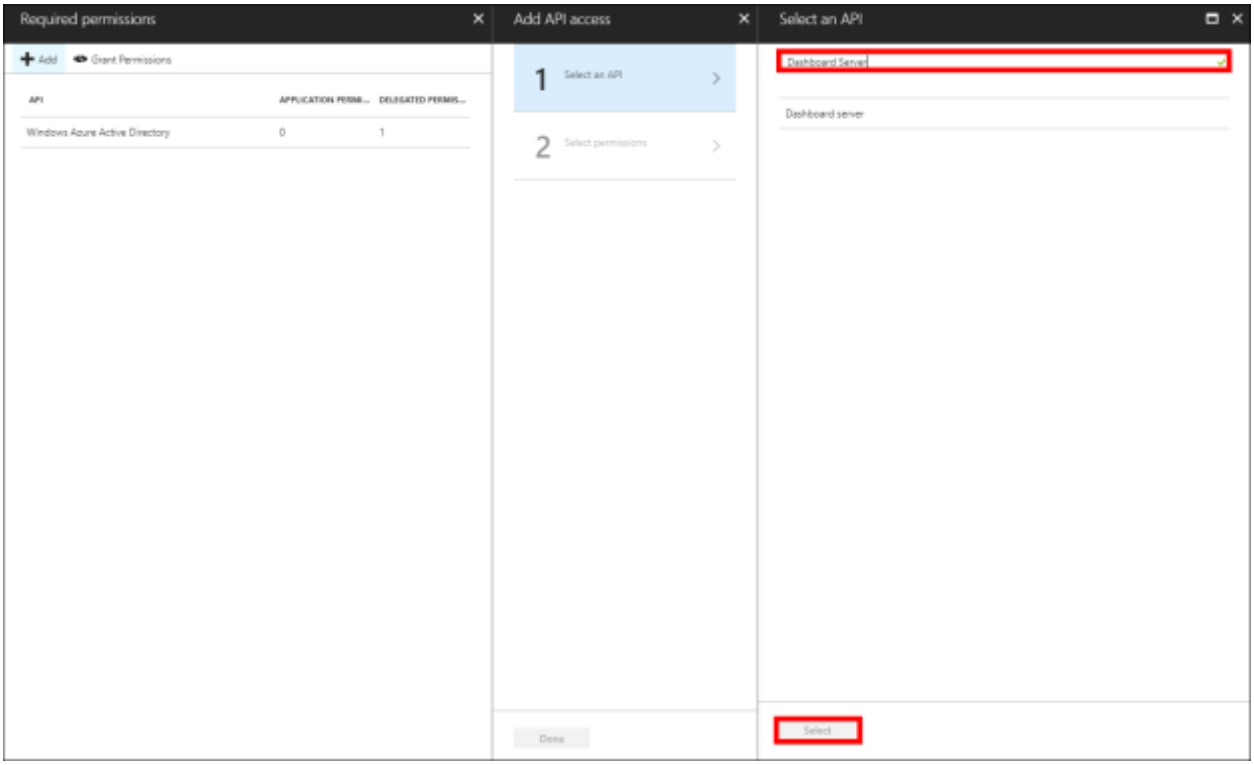
5. Choose the registered application and click the Settings.


Dashboard Designer
Registered app
Settings
Manifest
Delete
Display name
Dashboard Designer
Application type
Native
Home page
--
Application ID
fd02d7e4-eeba-433c-a959-fa9f6e7aefc0
Object ID
35a3e3bf-234f-42a9-8c8e-b7f91288135e
Managed application in local directory
Dashboard Designer

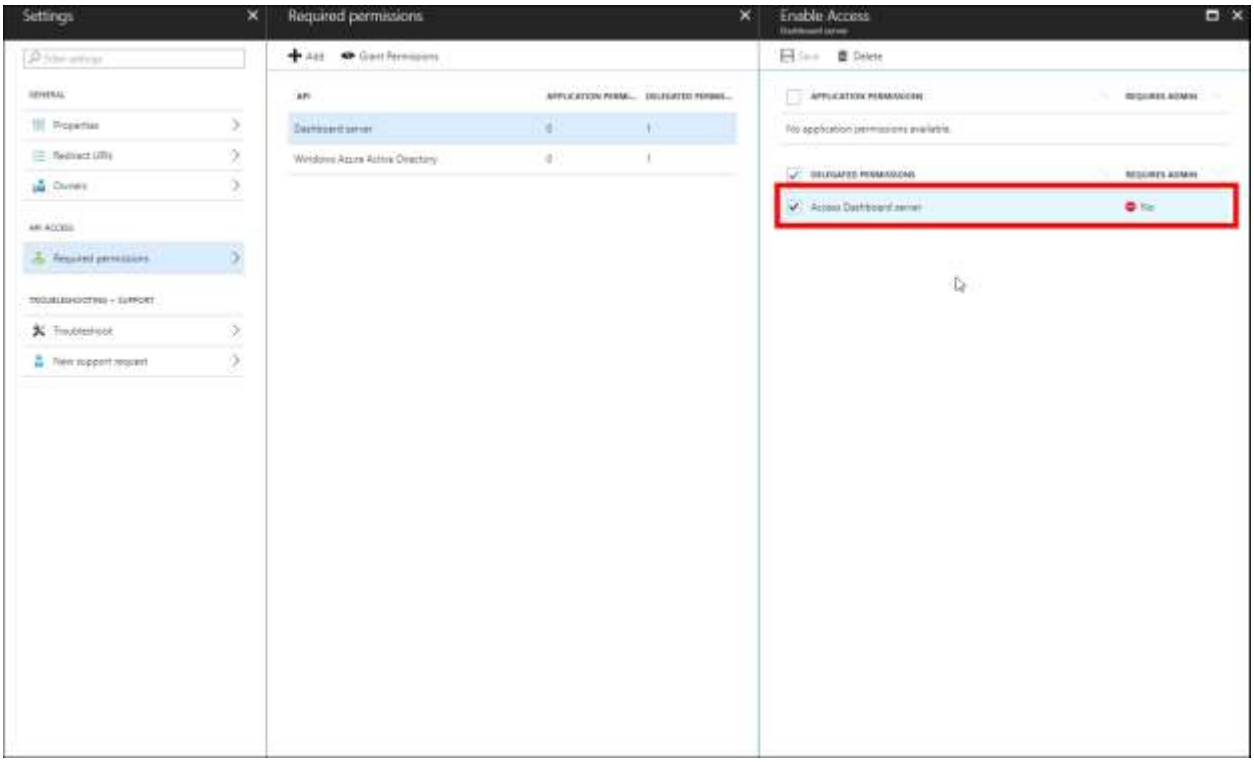
6. Go to application, click Settings, and select Required permissions. Then click Add, and then choose Select an API.



7. Select the Dashboard Server from the list and click Select.

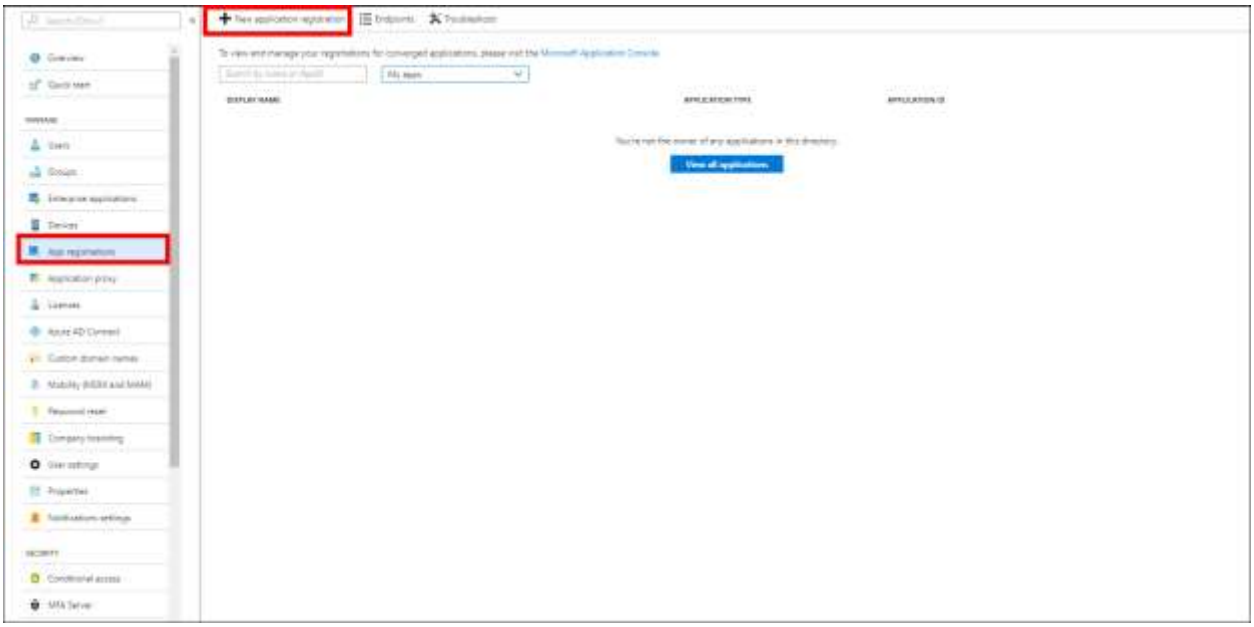


8. Select the delegated permission for accessing Dashboard Server and save it.

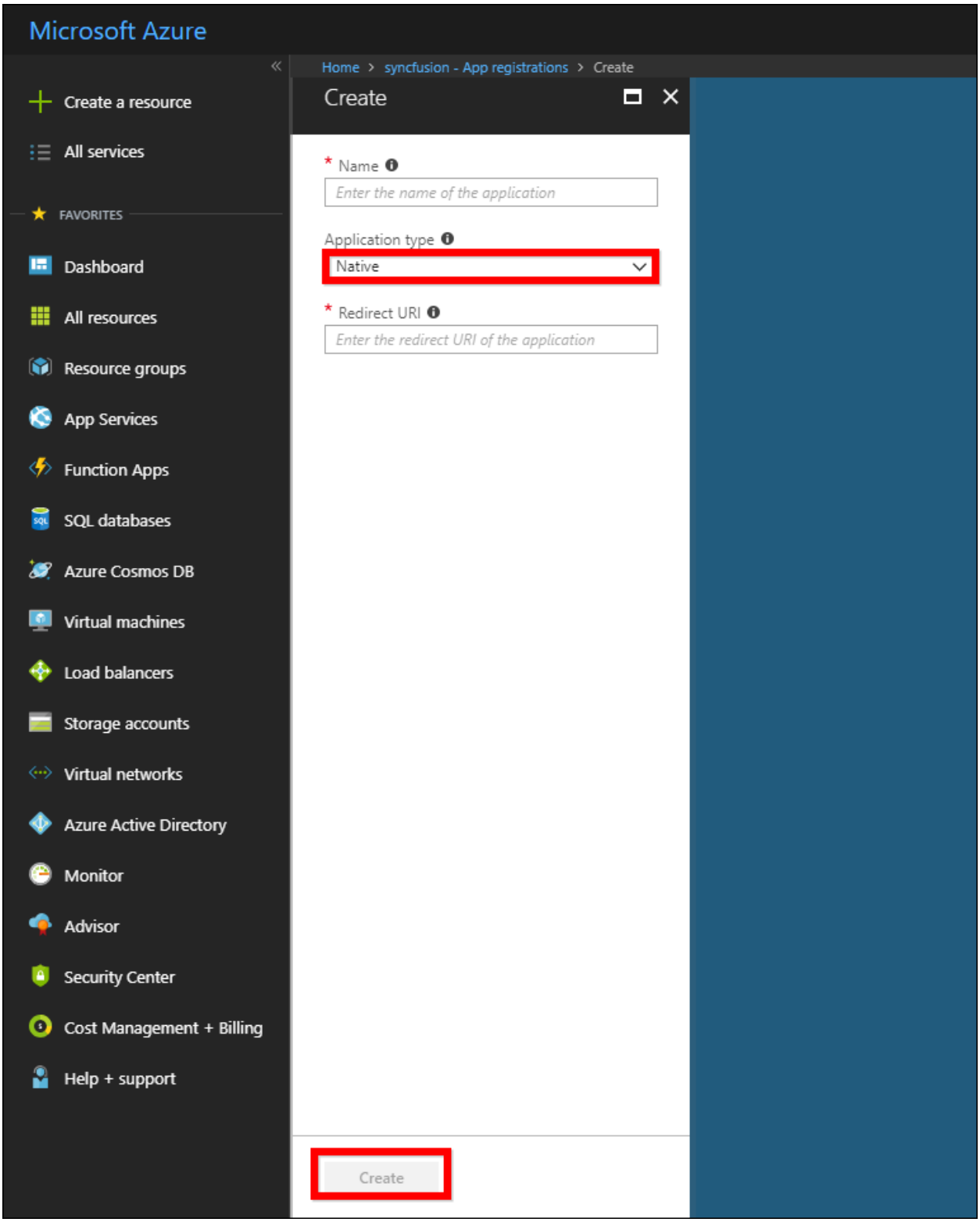


Configure Azure Active Directory to perform Single Sign-On in Syncfusion Dashboard Mobile application

1. Enter into the created directory and click Azure Active Directory. Select the App registrations and click New application registration to add a new application.



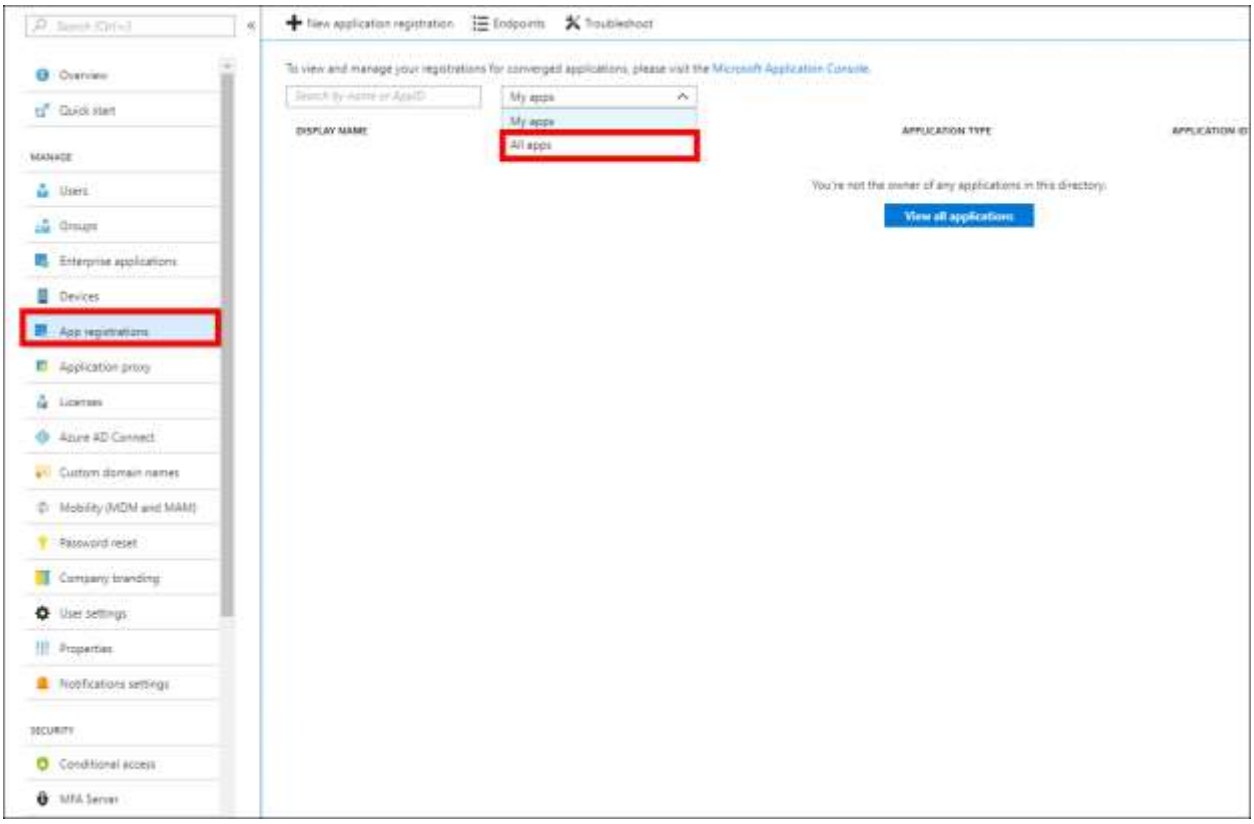
2. Enter the name of the client application and choose the Application type as Native, and then enter the Redirect URI.



3. Click Create, the client application will be added to the directory and you can view the details of the application in the App registrations.

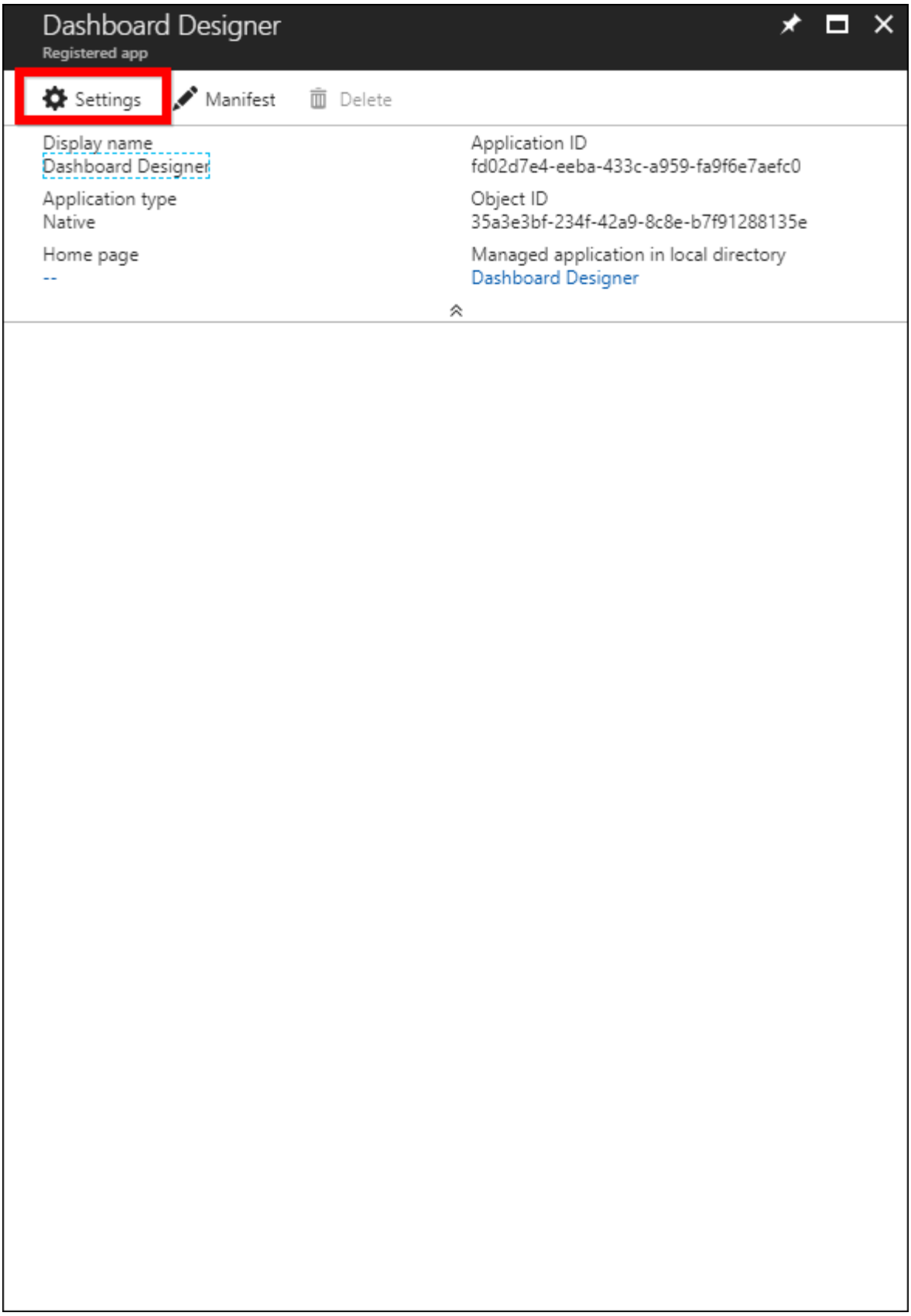
Note: Redirect URI should be the URL of the Syncfusion Dashboard Server application.

4. To view all registered applications, click the Azure Active Directory and select App registrations, and then choose All apps.

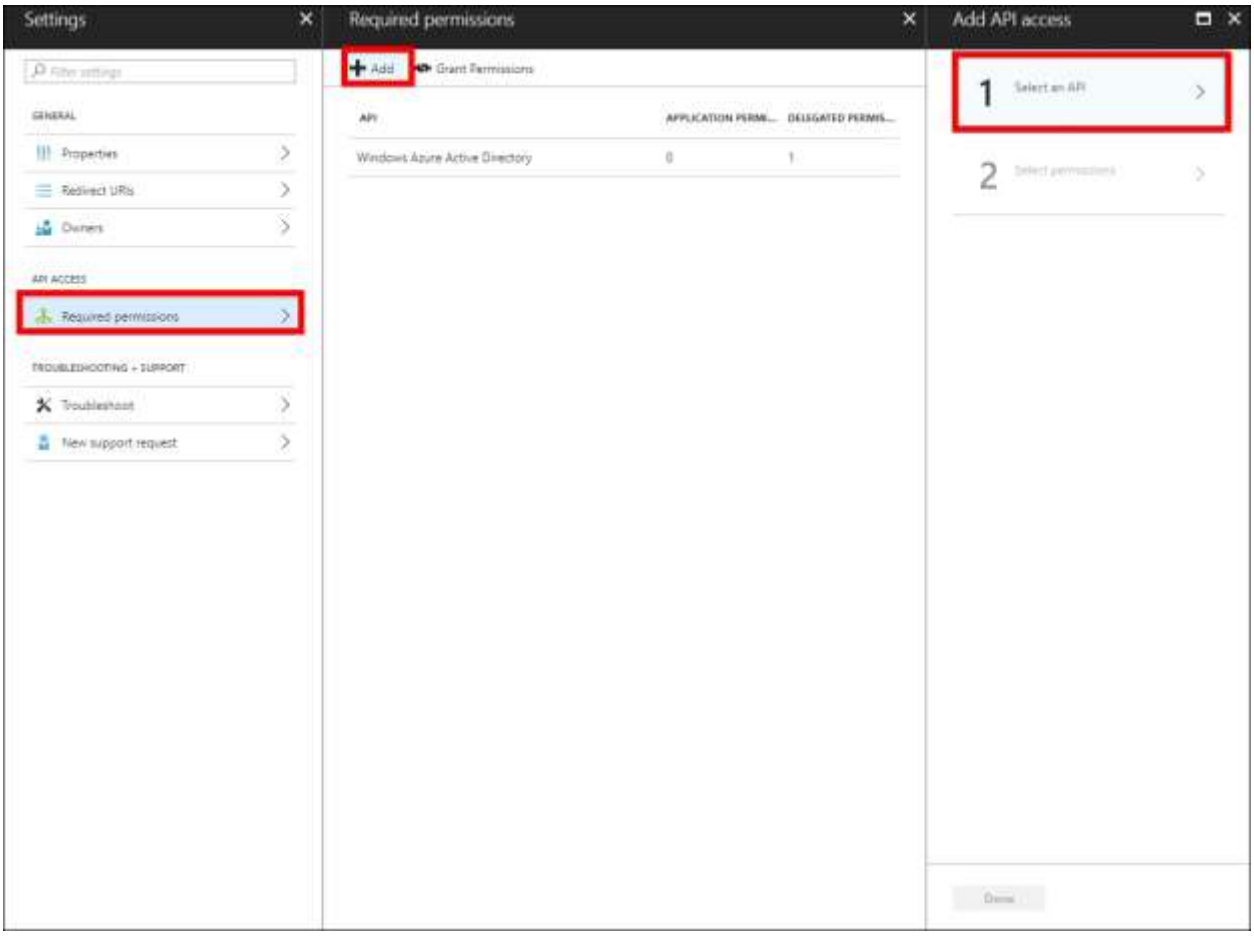


Now, you can add Dashboard Server application to the client application to enable Single Sign-On in native client application.

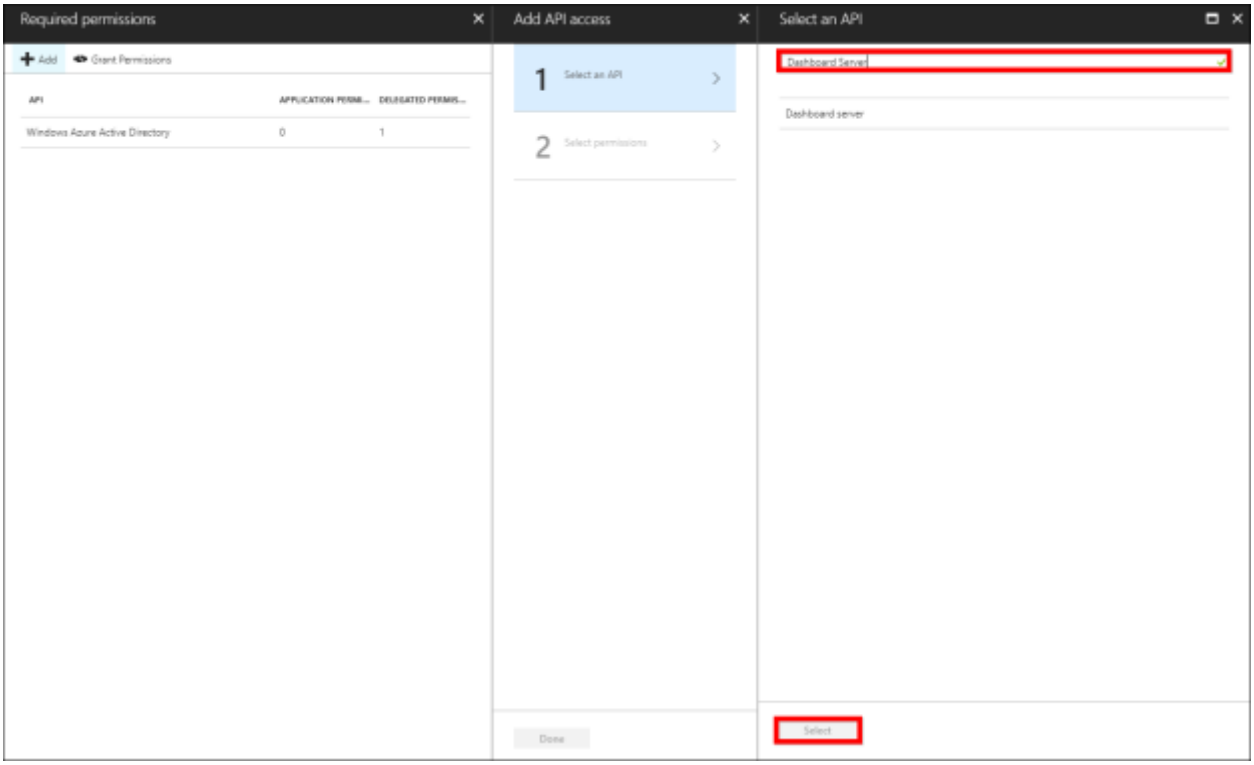
5. Choose the registered application and click the Settings.



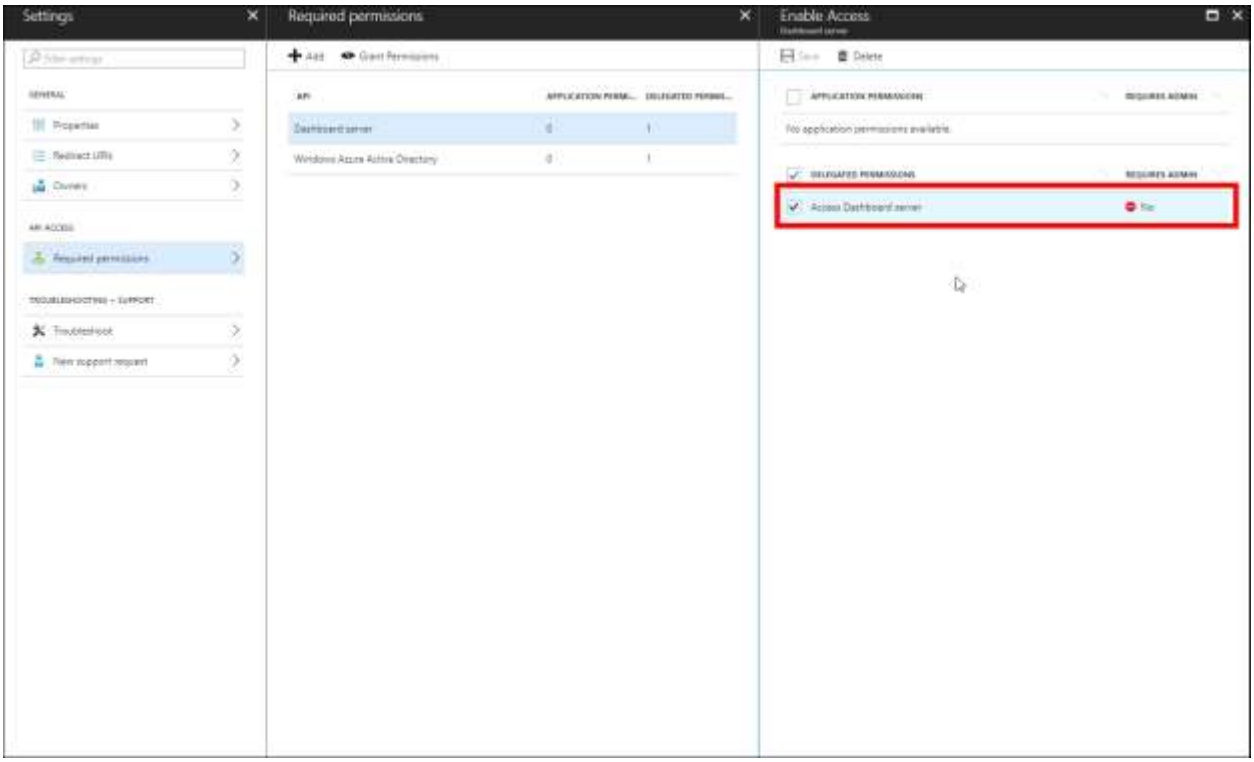
6. Go to application, click the Settings, and select Required permissions. Then click Add, and then choose Select an API.



7. Select the Dashboard Server from the list and click Select.



8. Select the delegated permission for accessing the Dashboard Server and saving it.



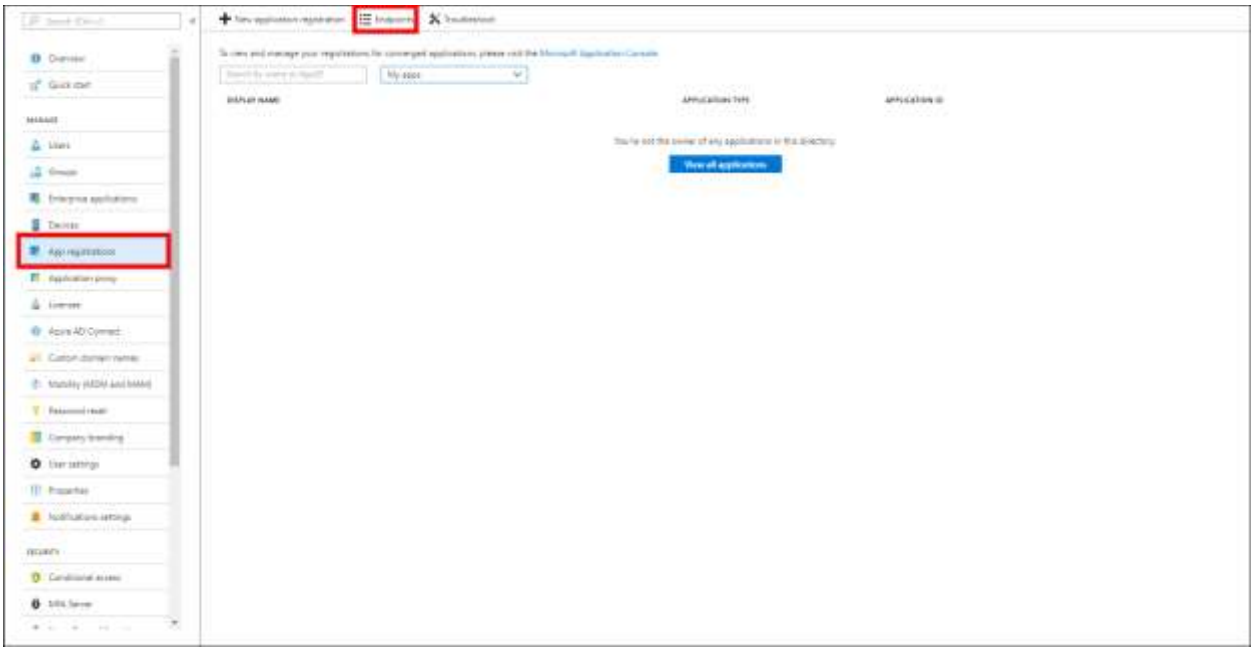
Setup Azure Active Directory users and groups

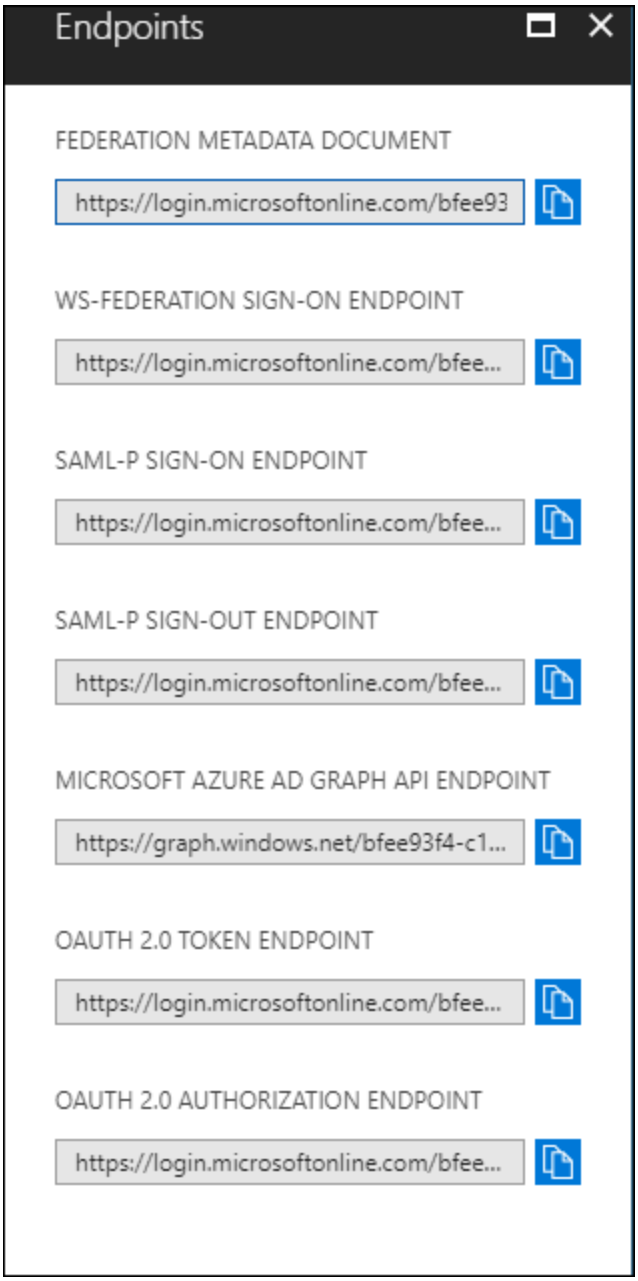
By default, a root user sourced from the Microsoft account is added to the directory. You can add users to this directory and later it will be imported to the Syncfusion Dashboard Server to perform the Single Sign-On.

Setup Syncfusion Dashboard Server to perform Single Sign-On

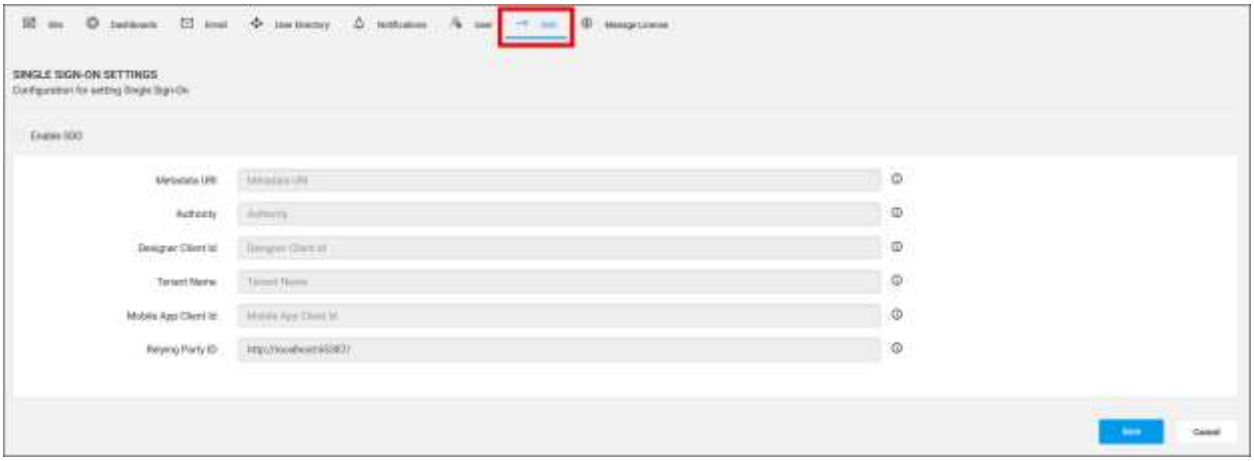
Configure the settings in Syncfusion Dashboard Server to perform Single Sign-On.

1. When you are in the same Azure Active Directory application (Dashboard Server) page, go to App registrations, and click Endpoints at the top, and a pop-up will be appeared as follows.



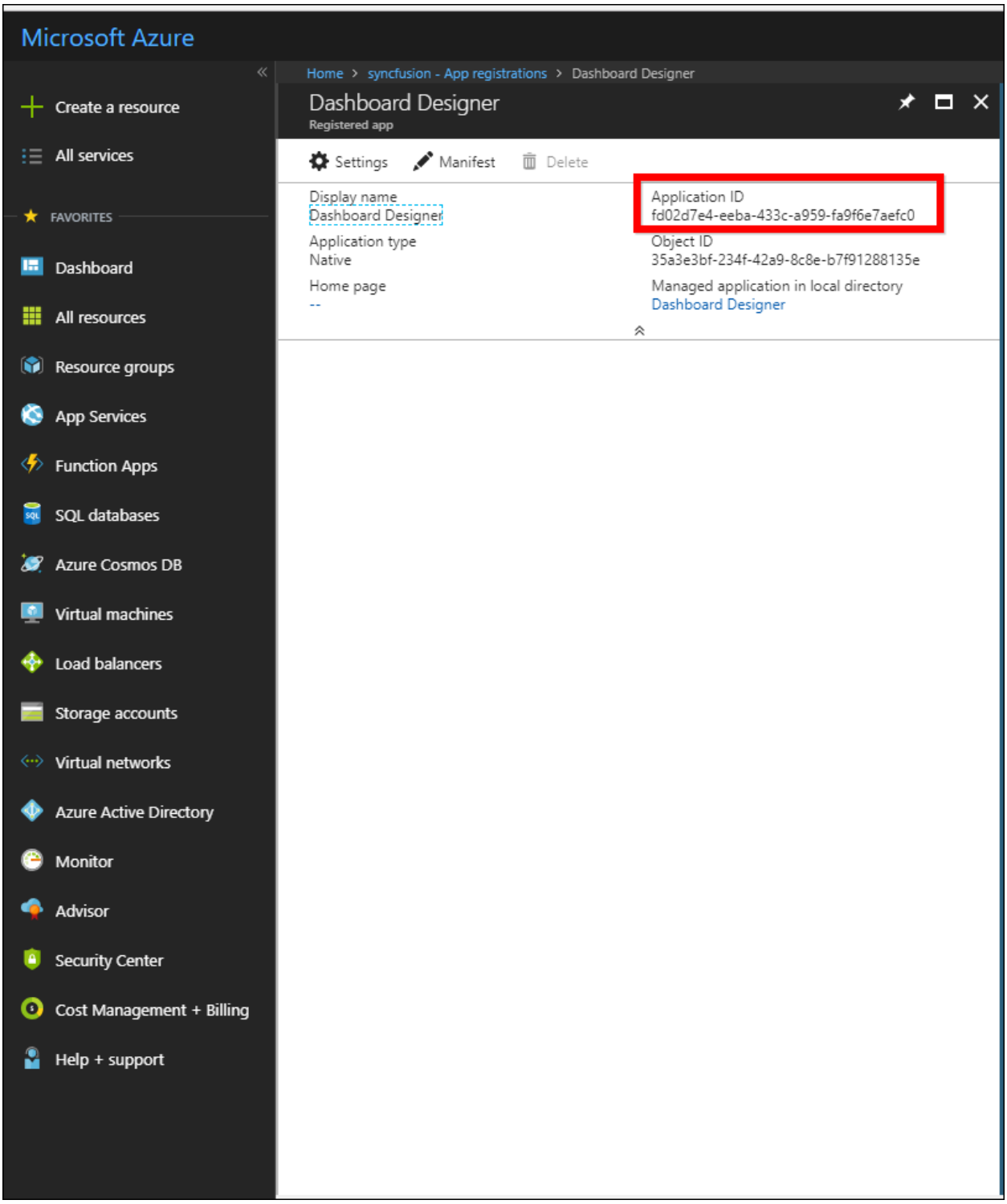


2. Start Syncfusion Dashboard Server and log on with administrator account. Click the Settings icon in the bottom-left corner and select the SSO settings.



3. Configure the following fields in the Syncfusion Dashboard Server to perform Single Sign-On with Dashboard Server.
    - Metadata URI: Copy the text in the first textbox named FEDERATION METADATA DOCUMENT and paste it.
    - Relying Party ID: The default site URL is already defined in this field. Copy this URL and go to configure menu of the server application created in the Azure. Paste the URL in Sign-on URL, App ID URI, and Reply URL and save the application.
4. Configure the following fields in the Syncfusion Dashboard Server to perform Single Sign-On with Dashboard Designer.
    - Authority: From the Azure application, click the view endpoints. A pop-up will be displayed. Copy the text in the second textbox named WS-Federation Sign-On Endpoint and paste it.
    - Tenant Name: Go to the created Azure Active Directory and copy the domain name by clicking it as shown in the following image.

- Designer client ID: Go to the registered application and click the Settings. Then, copy the Application Id and paste it.

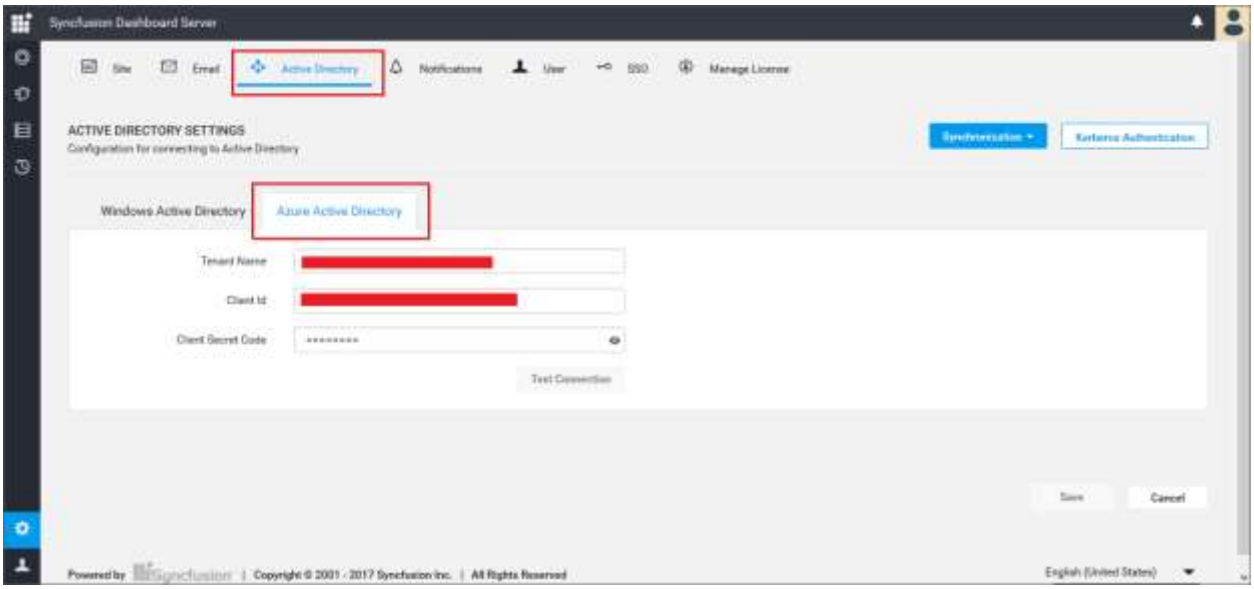


- Mobile App Client Id: The client ID of the Syncfusion Dashboards client application is created in the Azure Active Directory.

5. Now, click save. After the values are saved, the application is Restarted to apply the settings.

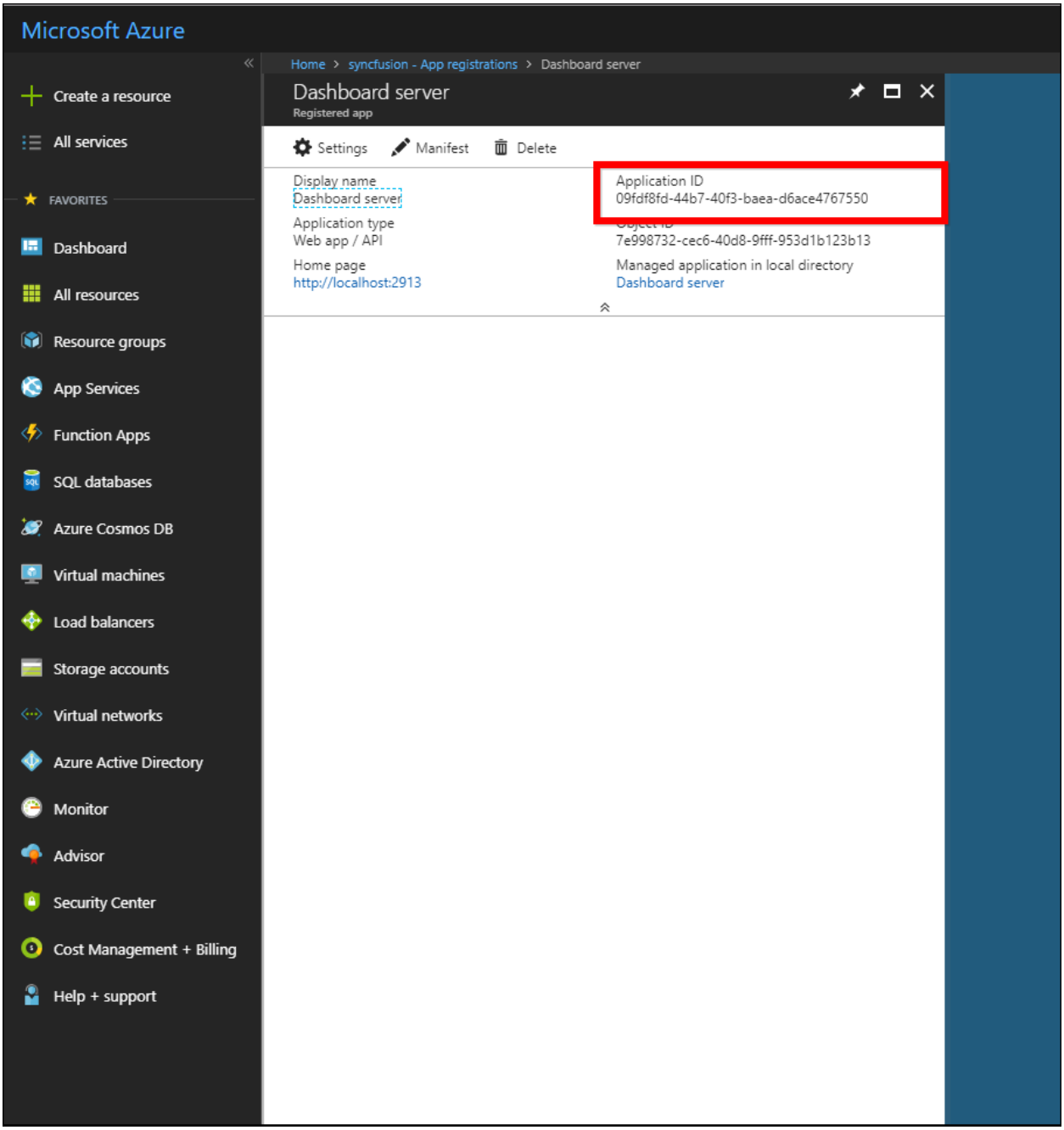
Setup Syncfusion Dashboard Server to import Azure Active Directory users and groups

1. Go to the Active Directory Settings page in the Syncfusion Dashboard Server and click the Azure Active Directory tab.

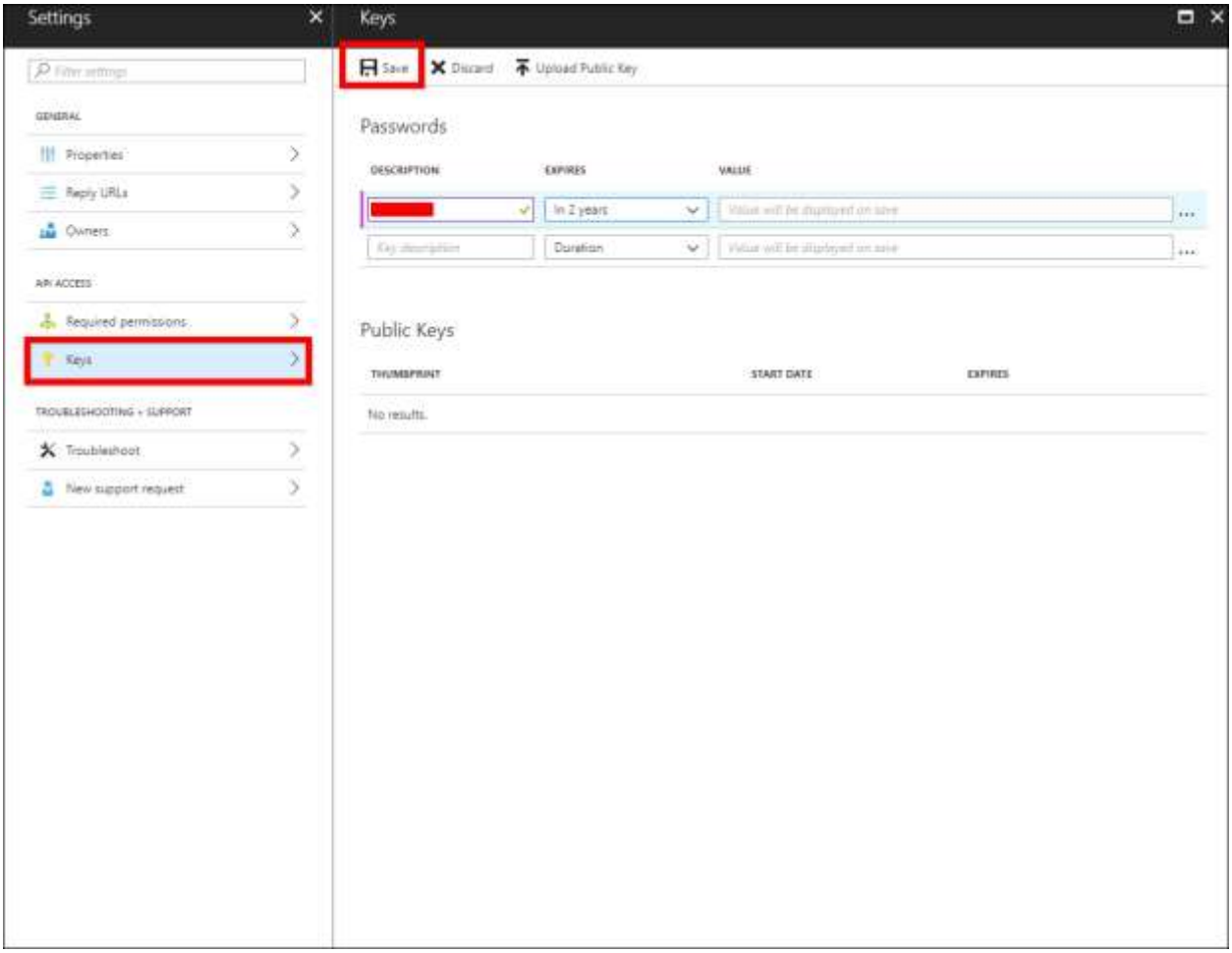


2. Configure the following fields in the Syncfusion Dashboard Server to import Azure users and groups.
    - Tenant Name: Go to the created Azure Active Directory and copy the domain name by clicking it as follows.

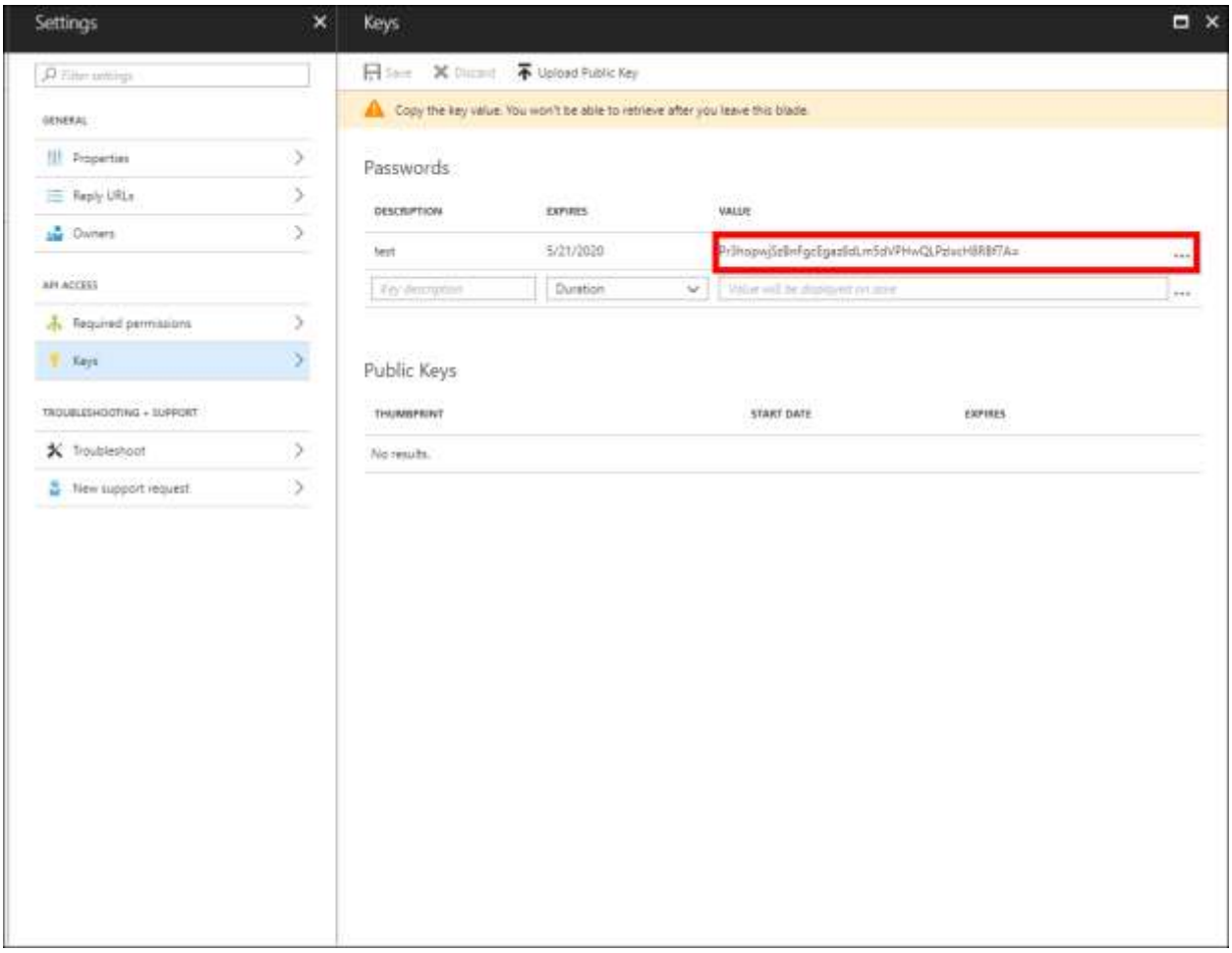
- Client ID: Go to the registered application and click the Settings, and then copy the Application Id and paste it.



- Client secret code: Go to the Settings and click Keys, and then enter the Description and choose the Duration under Passwords.



3. Click Save. The client secret will be generated, and then copy and paste it into the text box.

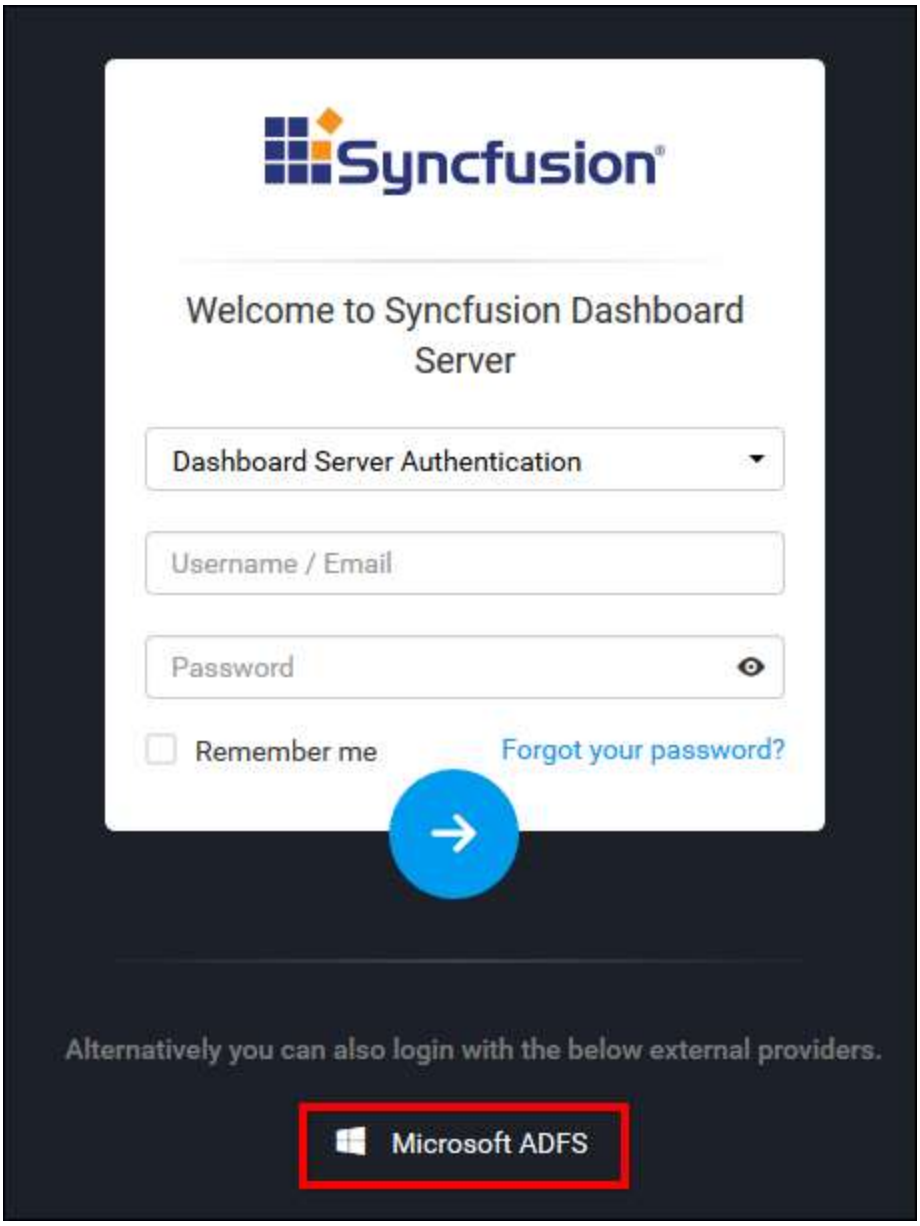


4. Now, test the connection. If the connection is valid, the success message is displayed. Save the settings.

The Azure user can be imported into the Syncfusion Dashboard Server. Refer to the following link to Import Azure Active Directory Users and Import Azure Active Directory Groups.

Login with Azure ADFS

After the Single Sign-On settings are saved and the Azure users are imported to the Syncfusion Dashboard Server, you can logout from the application. Now, the login page is provided with the additional button named Microsoft ADFS, which opens the external authentication provider login window, as follows.



After sign in with the Azure username and password, you can log on to the Syncfusion Dashboard Server.

Note: To log on to the Syncfusion Dashboard Server with Azure ADFS, the particular user should be imported to the application. If the user is not imported, it redirects to the login page.

How to make Dashboard Server accessible from outside of the installed machine through IIS?

To make the Dashboard Server accessible from outside of the installed machine, host as an application in the IIS if it is not chosen to be hosted in the IIS while installing.

If the application is already hosted in the IIS, skip the below step and go to Accessing the application outside the installed machine.

Steps for hosting the application in IIS:
https://help.syncfusion.com/dashboard-platform/dashboard-server/setup/installation-and-deployment-for-3-2#host-dashboard-server-as-application-in-iis

After hosting the application, replace the site URL of Dashboard Server in the Site Settings page of the Dashboard Server.

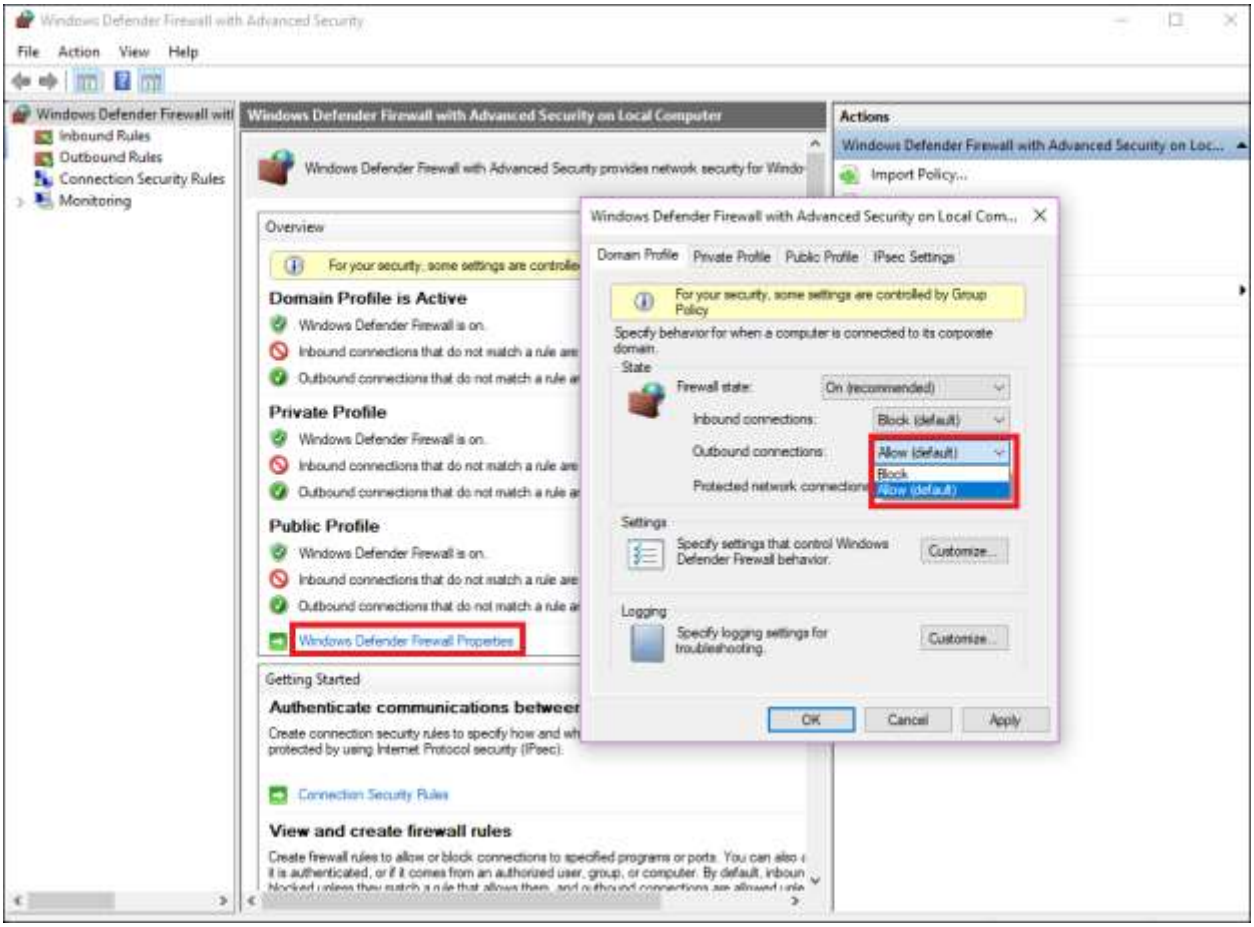
Reference help link:https://help.syncfusion.com/dashboard-platform/dashboard-server/site-settings/custom-rebranding#site-url

Accessing the application outside the installed machine:
To access the site outside of the installed machine, add the inbound rule to the hosted port number in the firewall by following the instructions in the below link.

https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-firewall/create-an-inbound-port-rule

If the site is still not accessible, check whether the firewall has blocked the connection.

1. Open Windows Firewall with Advanced Security.
2. Choose Windows Firewall Properties from the center pane.
3. Next to Outbound connections, choose Allow. Then, click OK.



If the site is still not accessible, check whether the machine's internet has been connected via proxy server.

Disable the proxy and try to access the Dashboard Server from outside of the installed machine. If it works, try the following configuration changes to access the Dashboard Server with the proxy connection.

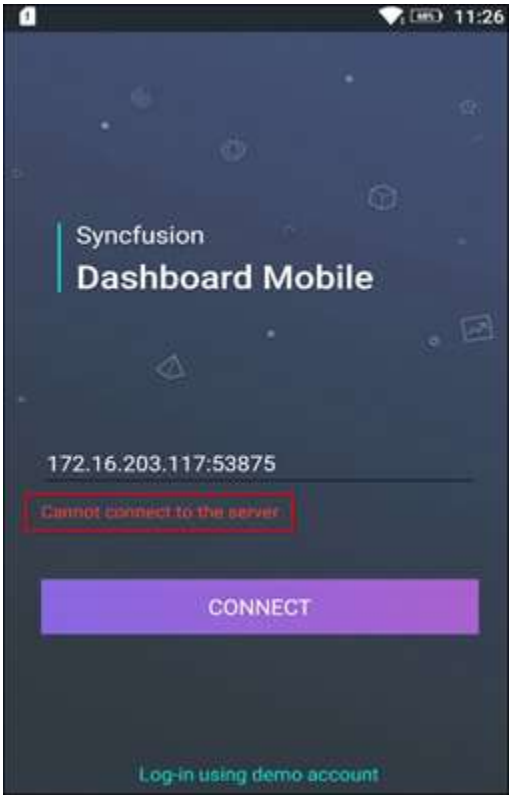
https://social.technet.microsoft.com/Forums/en-US/0face535-3c7a-4658-be34-6c376322ca34/microsoft-edge-cant-open-local-domains?forum=win10itpronetworking

How to resolve Syncfusion Dashboard Mobile app login issue

Syncfusion Dashboard Mobile app allows you access your Syncfusion dashboards. You can get insights into your data by navigating through your dashboards and staying up to date with the latest data. To list and view the dashboards in Syncfusion Dashboard Mobile app, you should provide the Syncfusion Dashboard Server URL along with the login credentials.

Symptom

An issue 'Cannot connect to the server' may occur when configuring Syncfusion Dashboards in Syncfusion Dashboard Mobile application as follows.



This section explains how to resolve this login issue.

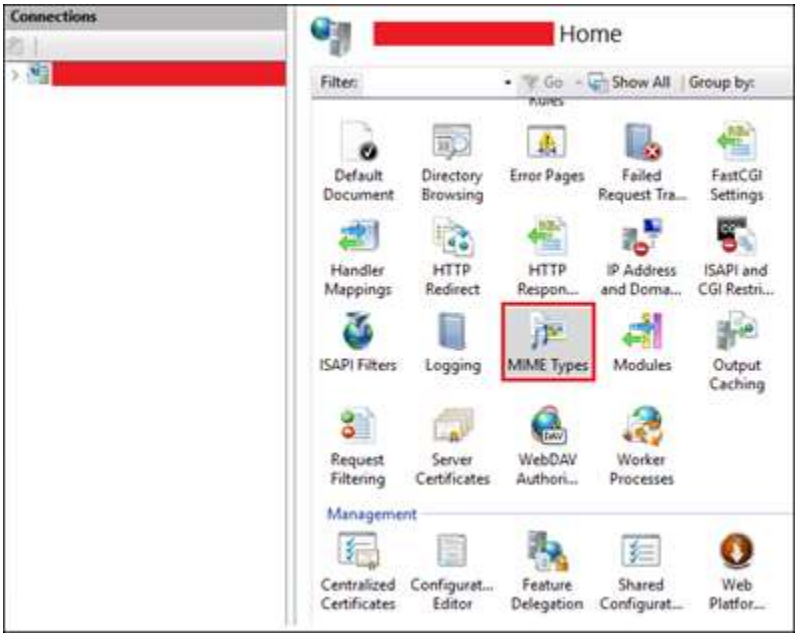
Cause

Syncfusion Dashboard Server uses fonts with '.woff' extension in its dashboards. When the MIME type is not added, these fonts will not be downloaded. Hence, the above reported issue occurs.

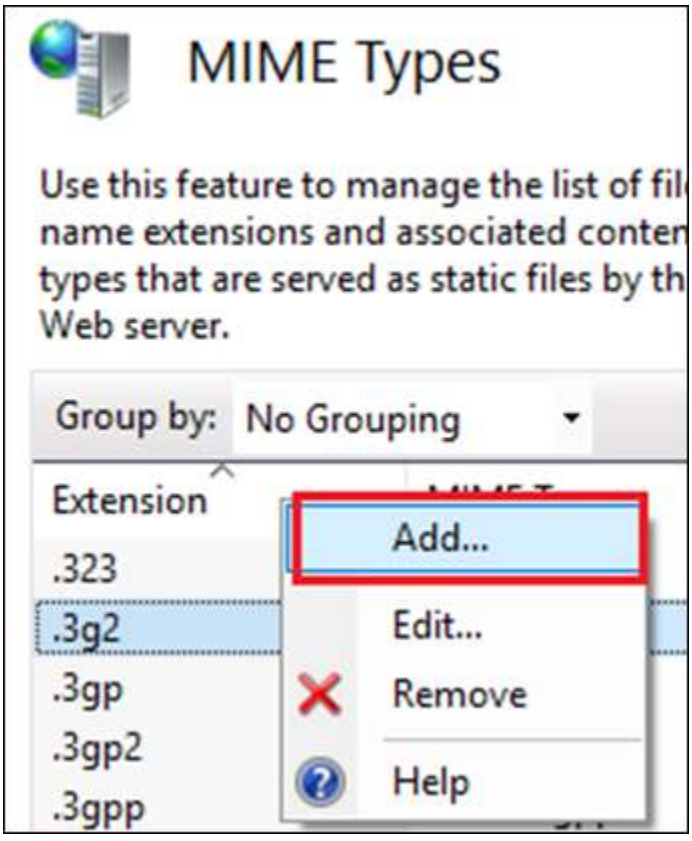
Solution

Adding MIME type for the font with '.woff' extension resolves this issue. Follow these steps to add the MIME type in IIS:

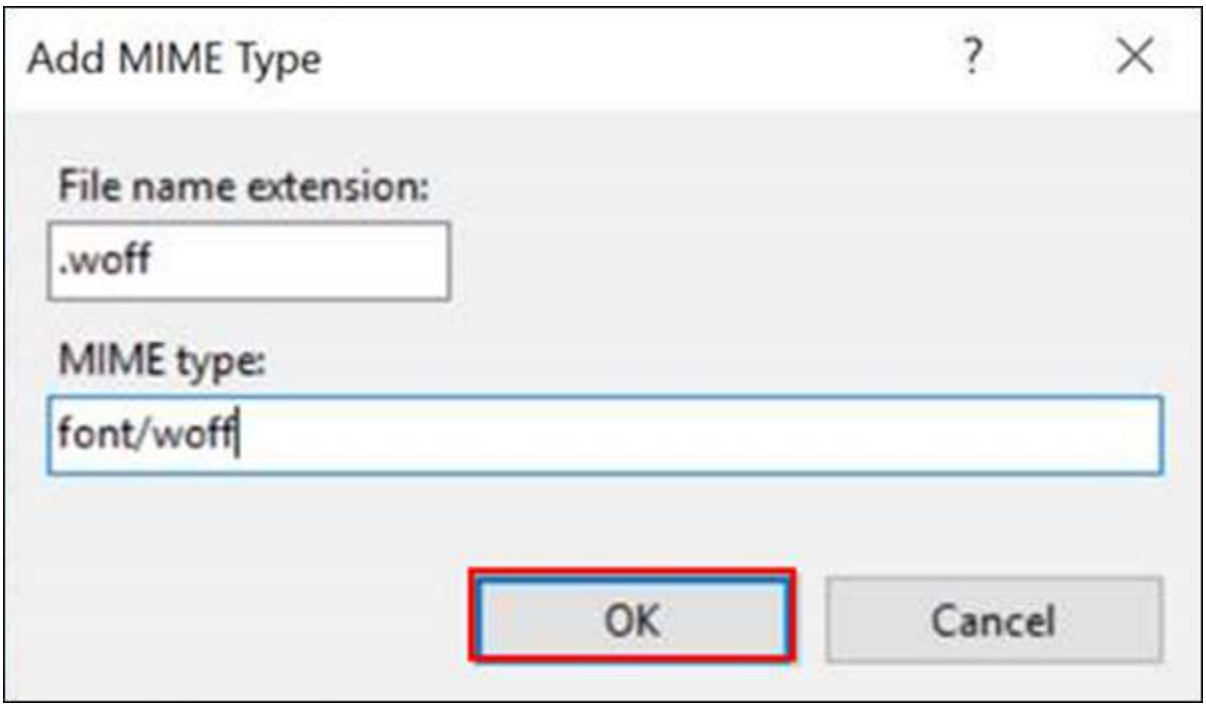
1. Open Internet Information Services (IIS).
2. Click **MIME Types** from the home section as follows.



3. Right-click and select **Add** option in MIME Types page.



4. In Add MIME Type dialog, give '.woff' extension as follows and click **OK**.



Now, all the fonts with '.woff' extension can be downloaded from the Dashboard Server application and the above issue will be resolved in Syncfusion Dashboard Mobile application.

Dashboard Mobile

Overview

Syncfusion Dashboard Mobile app lets you access your Syncfusion dashboards on the go. Get insights into your data by navigating through your dashboards and stay up to date with the latest data. Check the below list of key features in the Syncfusion Dashboard Mobile.

Key features

Dashboard management — Dashboards are efficiently organized under the categories like favorite, recent, public and all dashboards.

View Dashboards --- The built-in Dashboard Viewer control lets the user to view Dashboards.

Export --- Dashboards can be exported to image, PDF, Excel and CSV file formats.

Supported Operating Systems

- iOS 9.0 or later, Compatible with iPhone, iPad and iPod touch.
- Android – Android versions 4.1 or later.

Get Syncfusion Dashboard Mobile

To download the Syncfusion Dashboard Mobile App for your device, click the relevant link below according to your device OS.

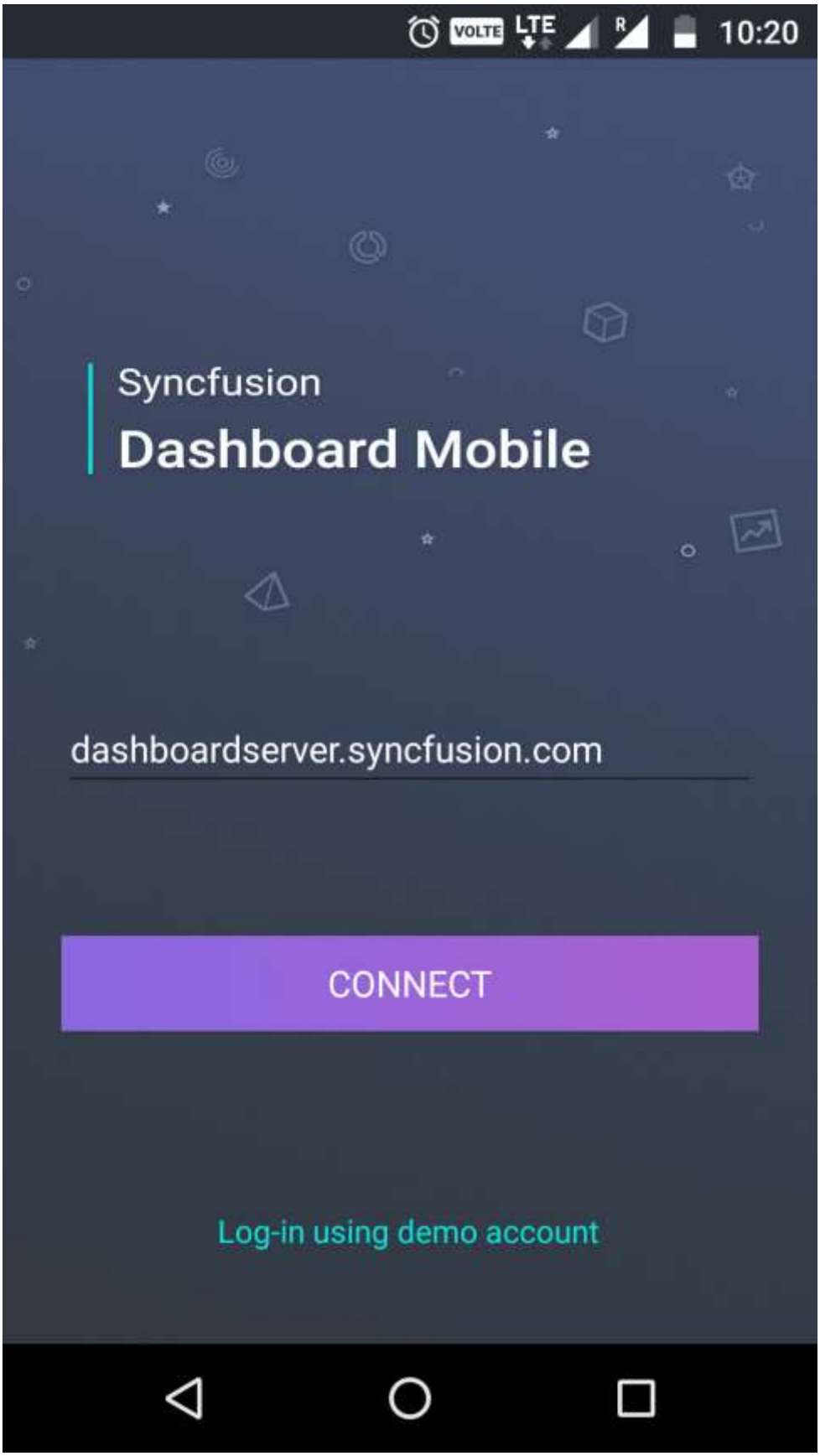
Create a support incident

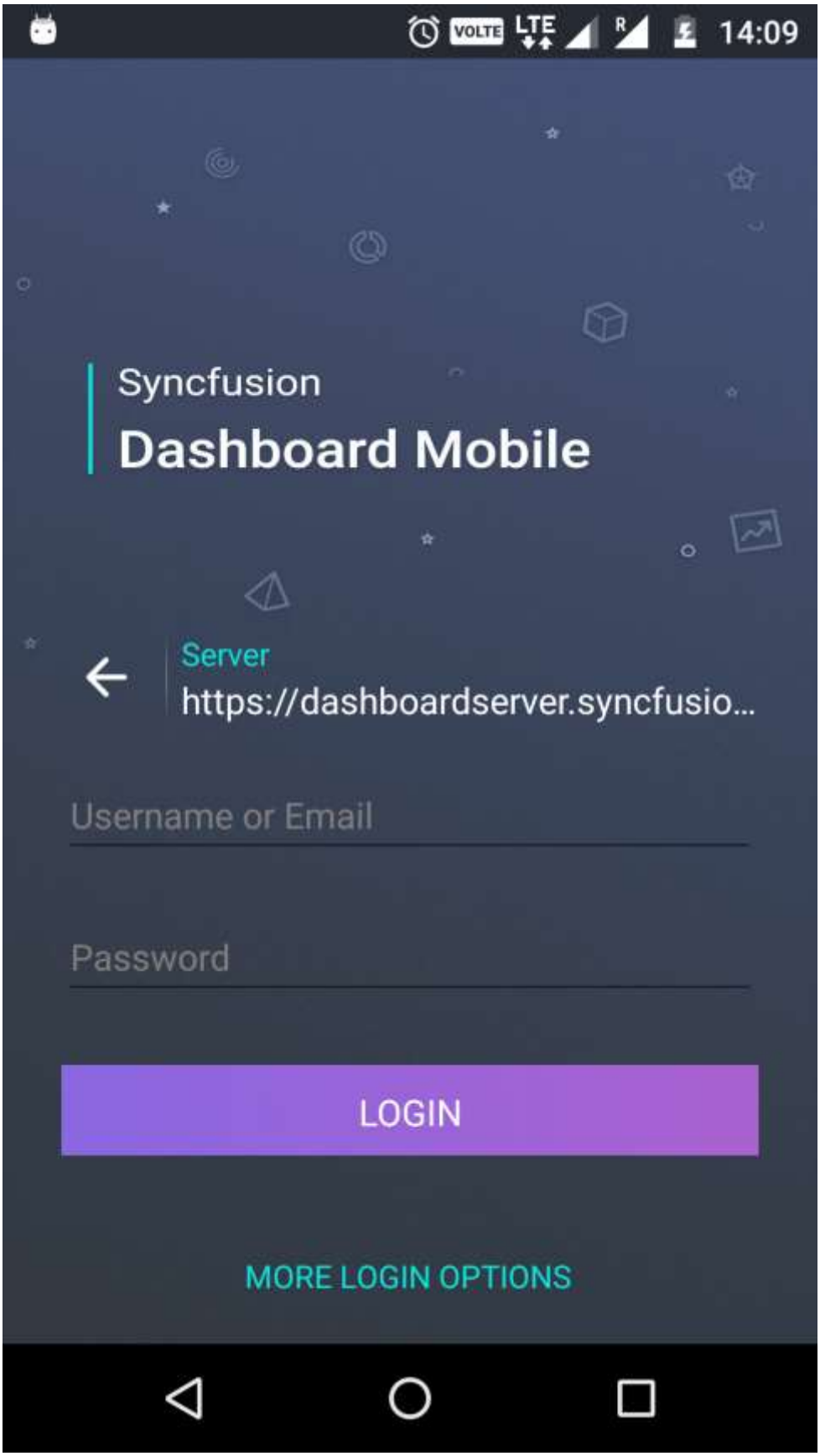
If you are still not able to find the information that you are looking for in the self-help resources mentioned above then please contact us by creating a support ticket.

To list and view the dashboards in Syncfusion Dashboard Mobile app you need to have the Syncfusion Dashboard Server URL.

Configure and Login in Syncfusion Dashboard Mobile app

If this is your first time logging in, type in the server URL and then enter your username and password to log in.





After the successful login you can see the dashboards listed in different tabs.

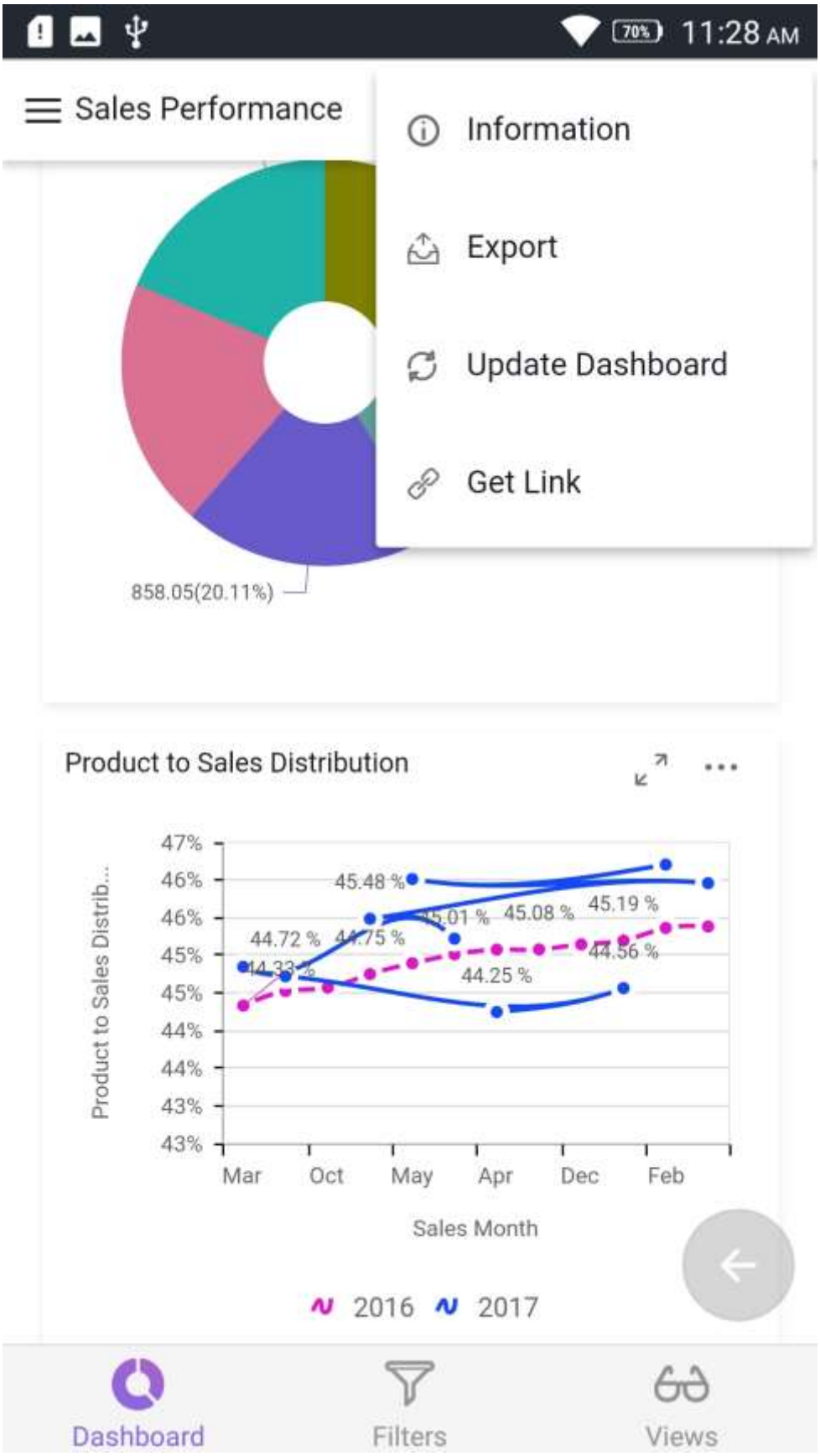
List Dashboards

After the successful login you can see the dashboards listed in different tabs like favorite, recent, category, public and all dashboards.

Searching, Filtering and Sorting - You can search and sort the dashboards based on Owner name, Modified date and Category, to quickly view the dashboard.

View Dashboards

You can view and interact with the dashboards using Syncfusion Dashboard Mobile app.



Export Dashboards - Dashboards can be exported to image, PDF, Excel and CSV file formats.

Update Dashboard - Dashboard data can be updated easily with real time data.

Filter Dashboard - Dashboard data can be filtered and viewed easily.

Expand and view the widget - You can expand the widgets also export the dashboard widgets individually.

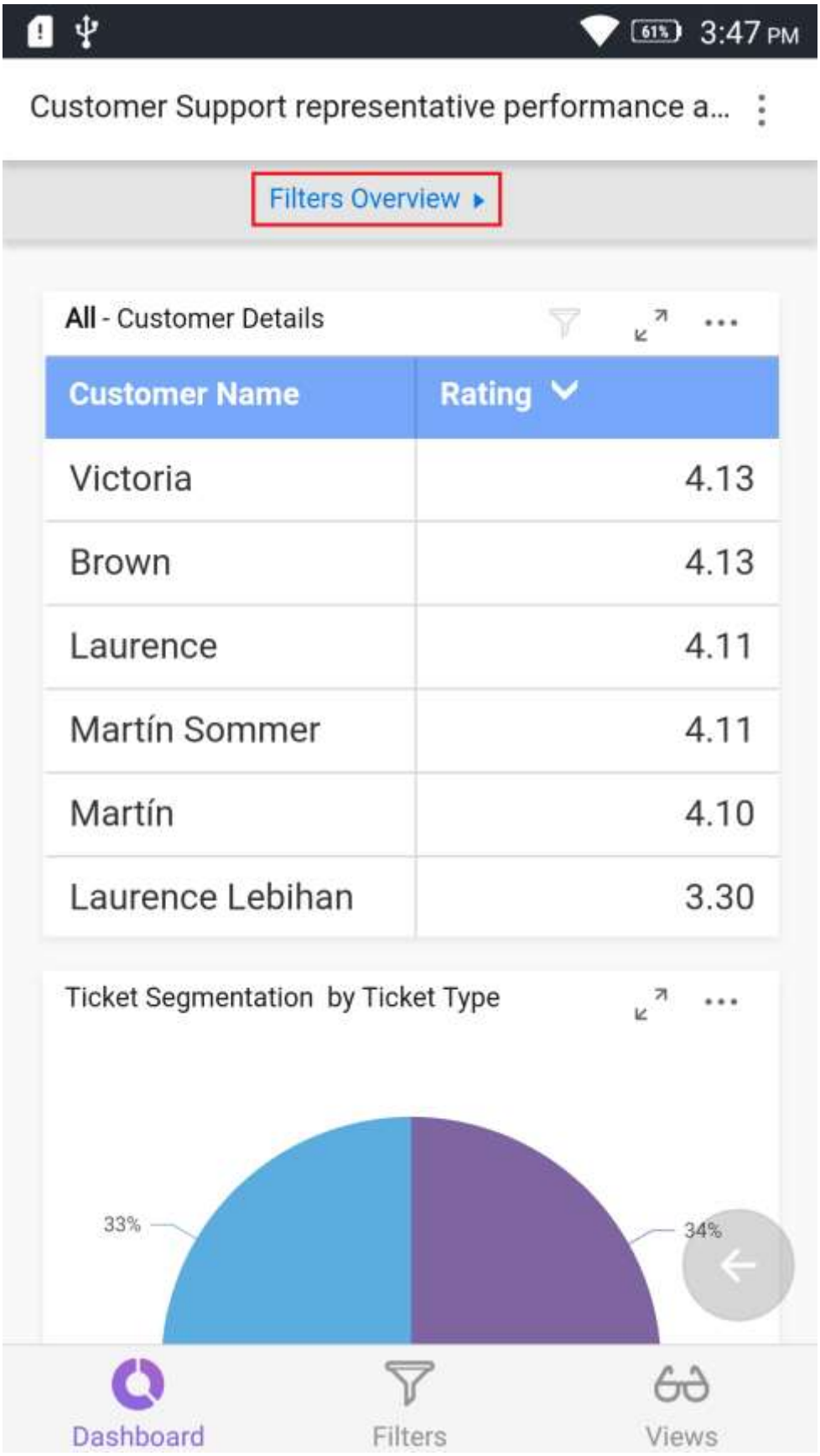
Get link - The get link helps you to get the dashboard link with the current view data along with the filters.

Dashboard Views

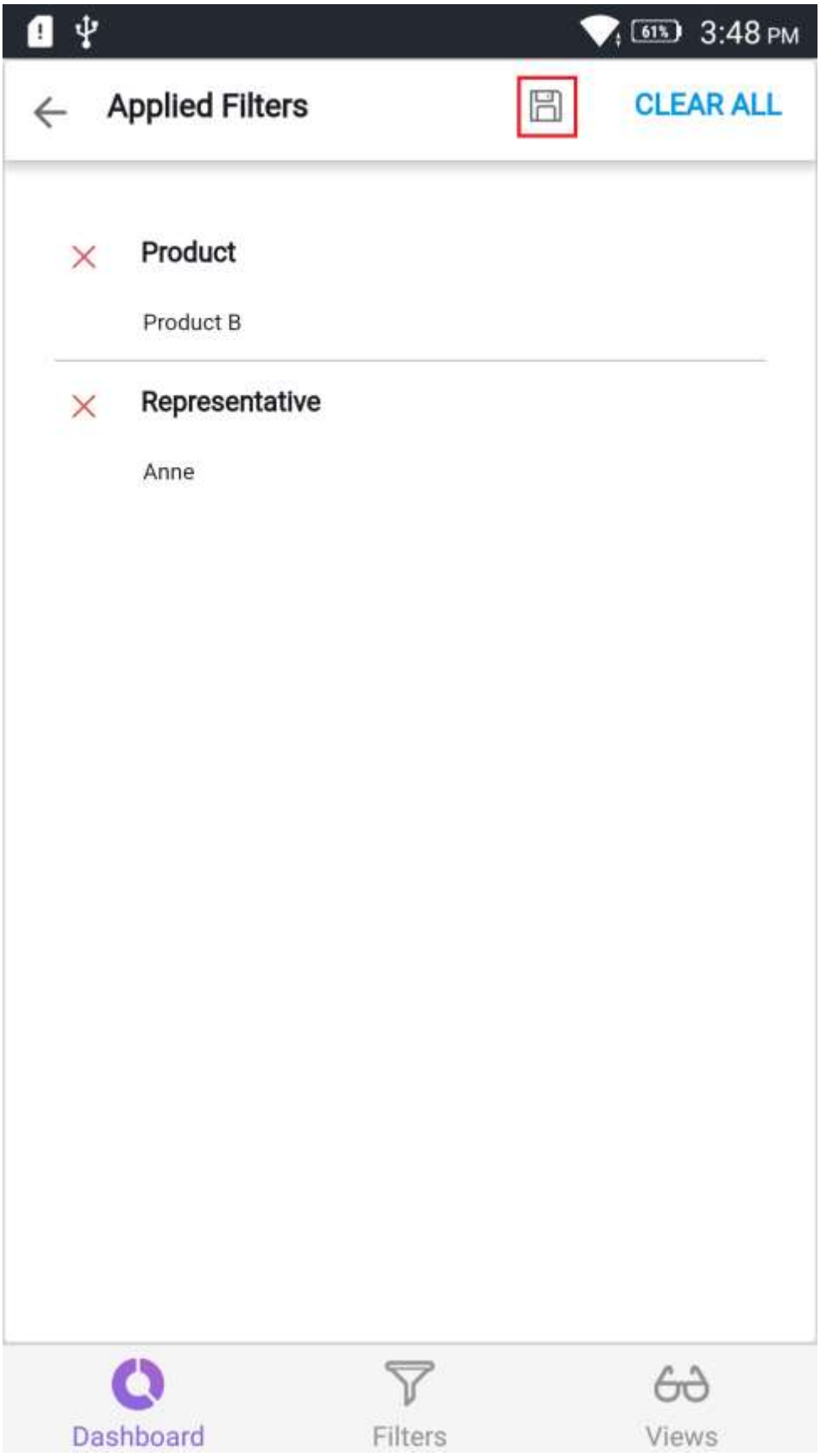
If the user has permission to access the dashboard, then the user can create the Dashboard Views in that dashboard from the mobile app.

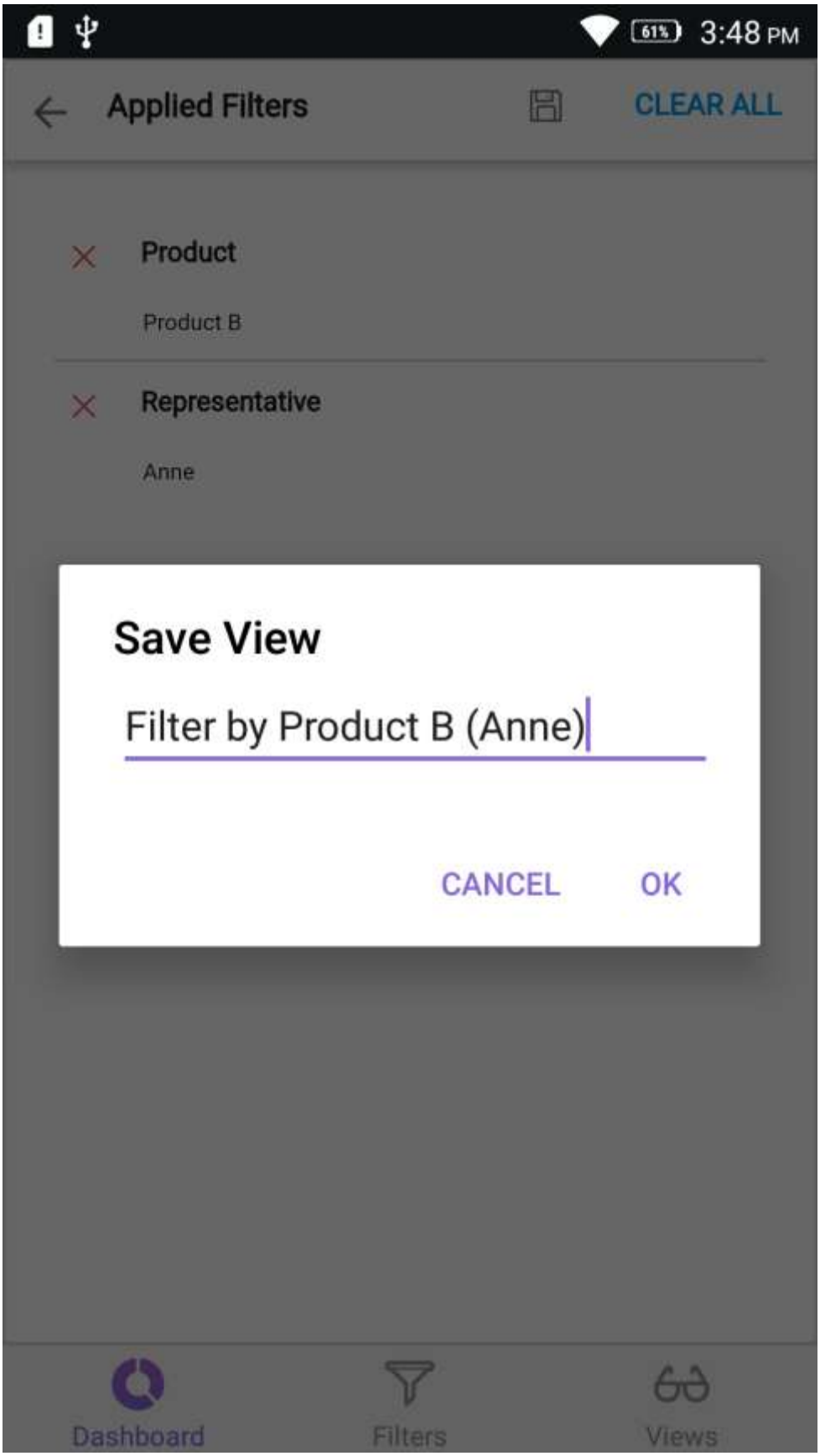
Add Dashboard Views

Filters applied in the dashboards are summarized in the Filter Overview section under Applied Filters.

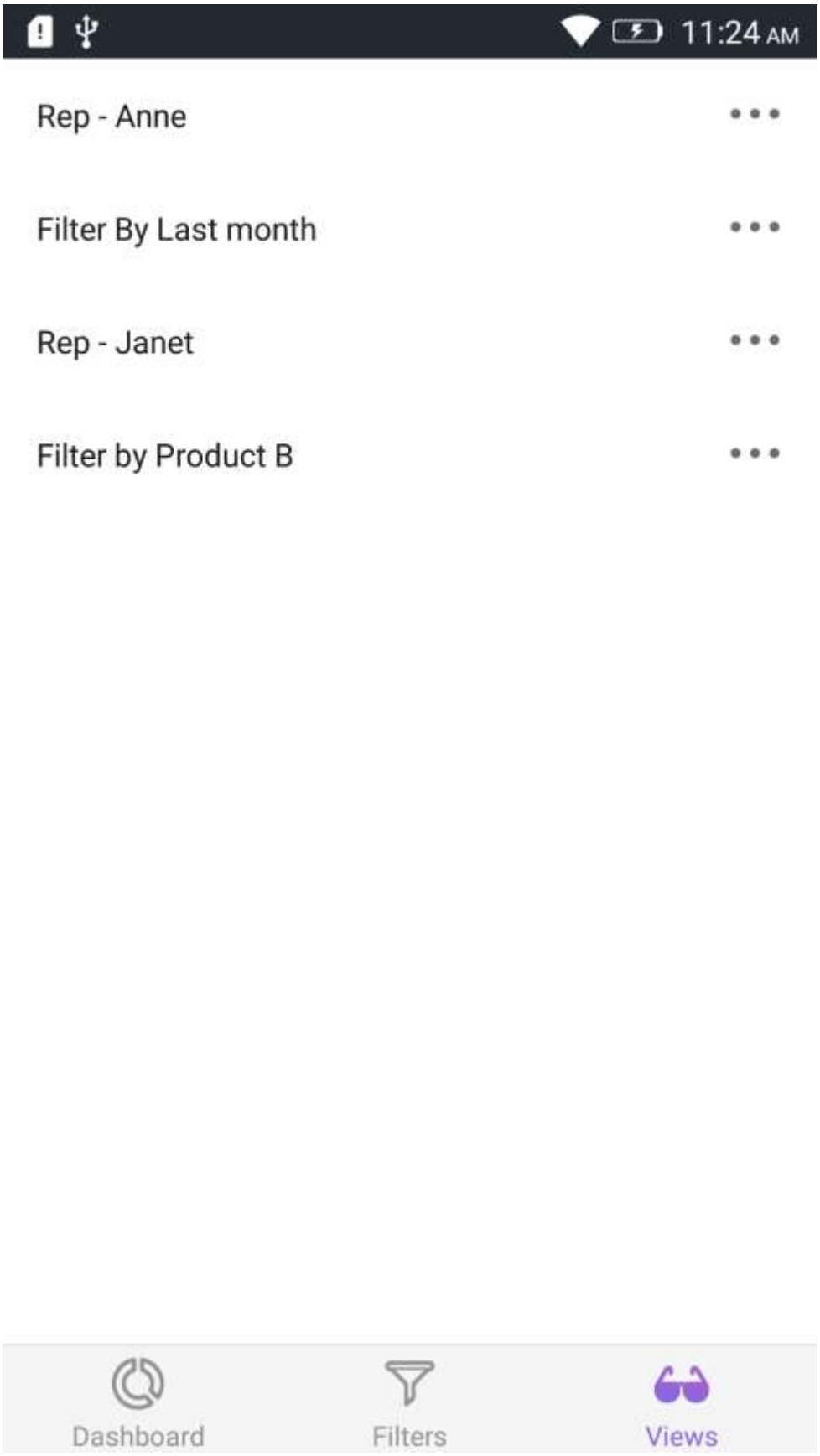


To save the filters, click on Save icon, a popup is opened to type the name for the Dashboard View and submit it.



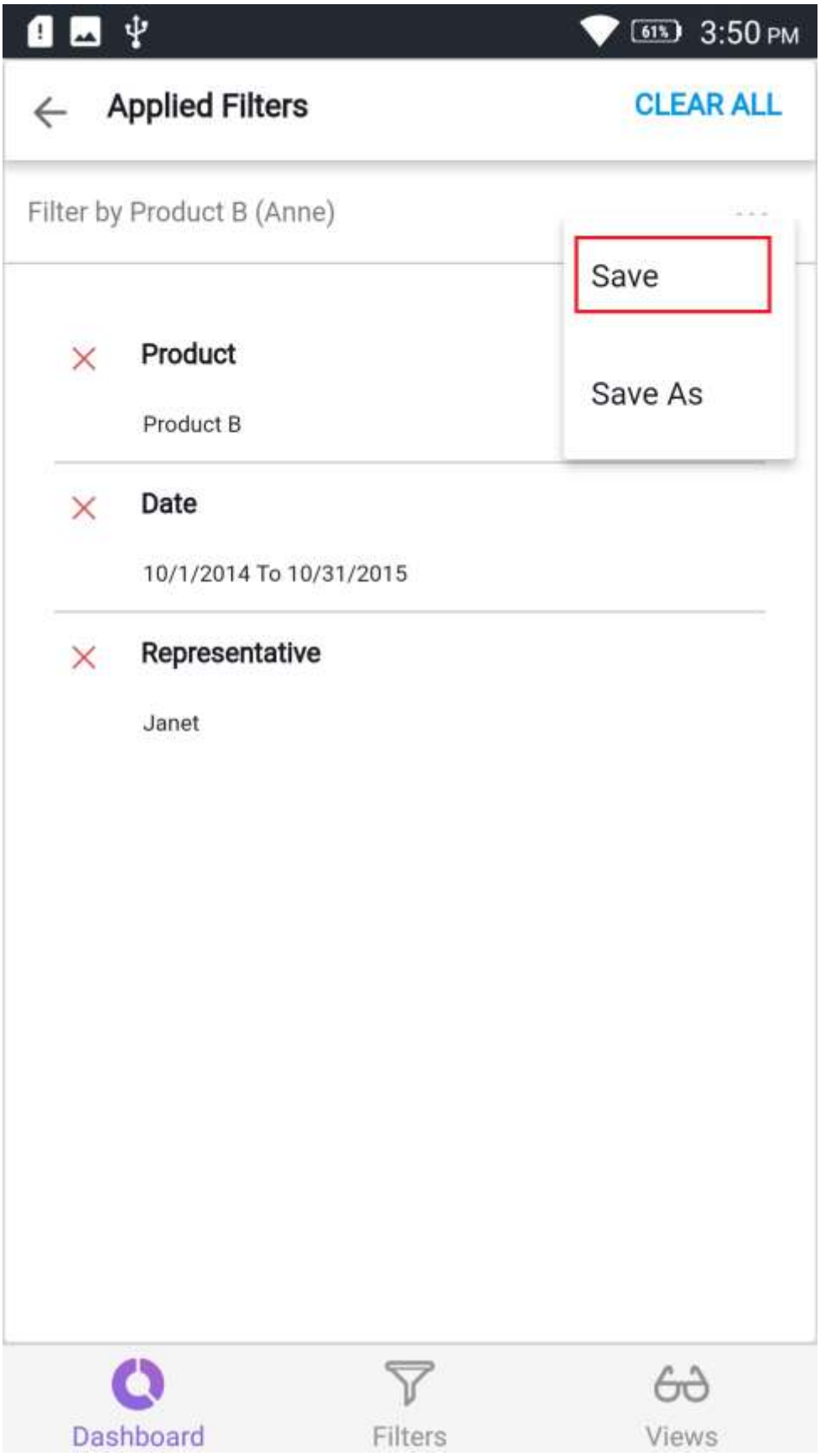


User can check their saved views in the Views tab.



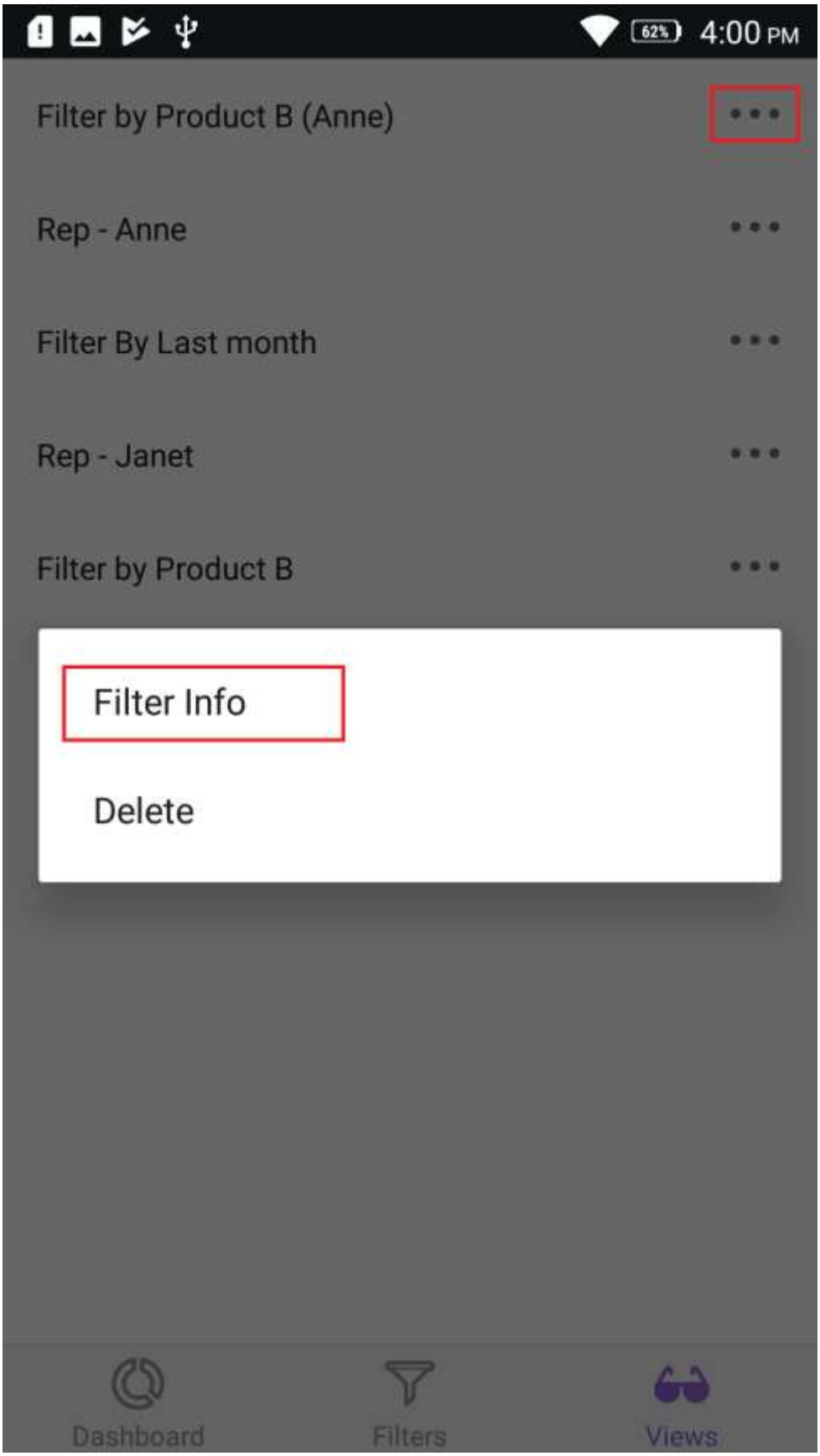
Update Dashboard Views

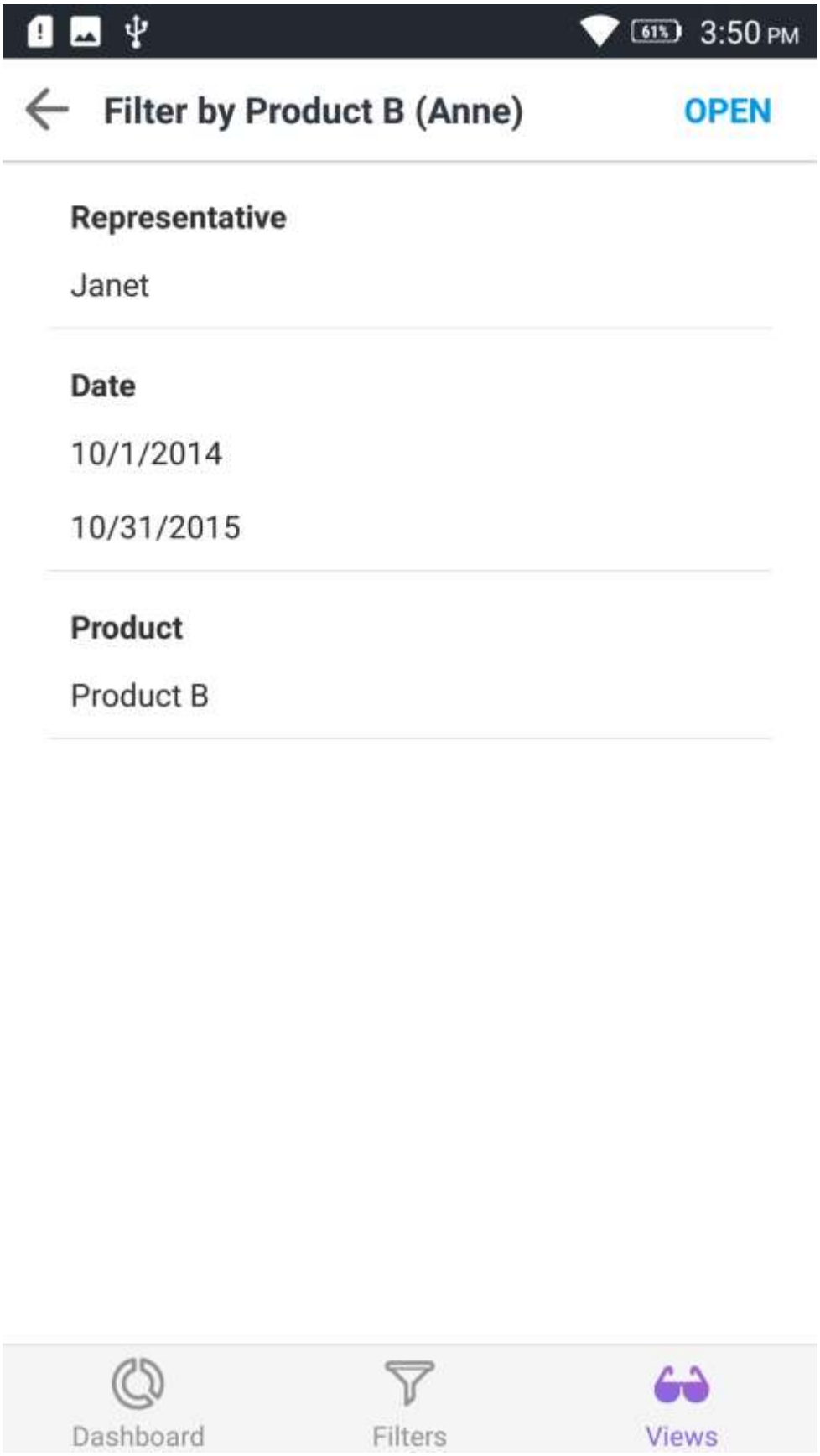
After saving a Dashboard View, you can change the filters in the dashboard and update the changes in the same Dashboard View or you can save it as a new Dashboard View.



View Dashboard View Info

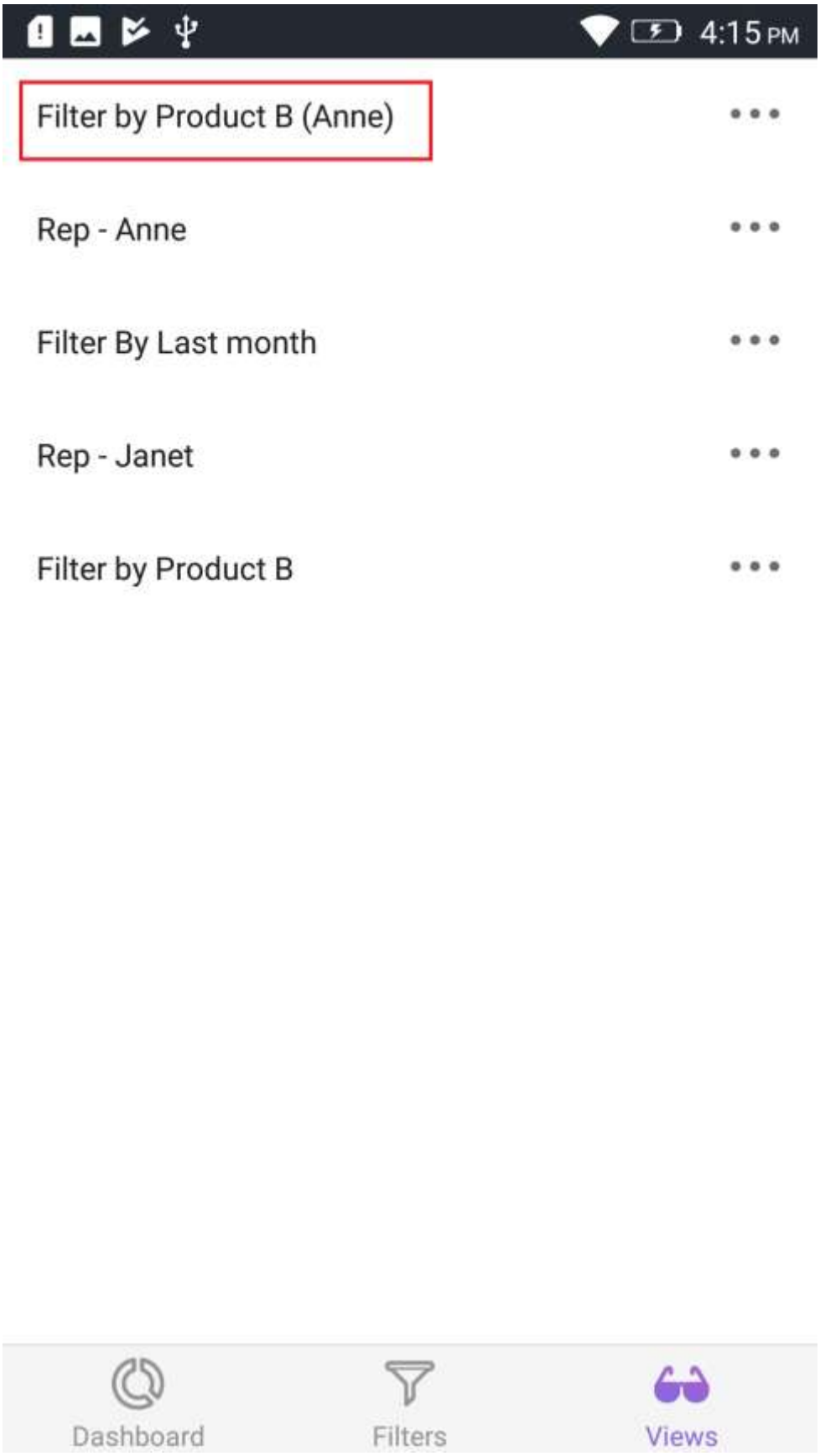
User can view the filter information of the saved views.



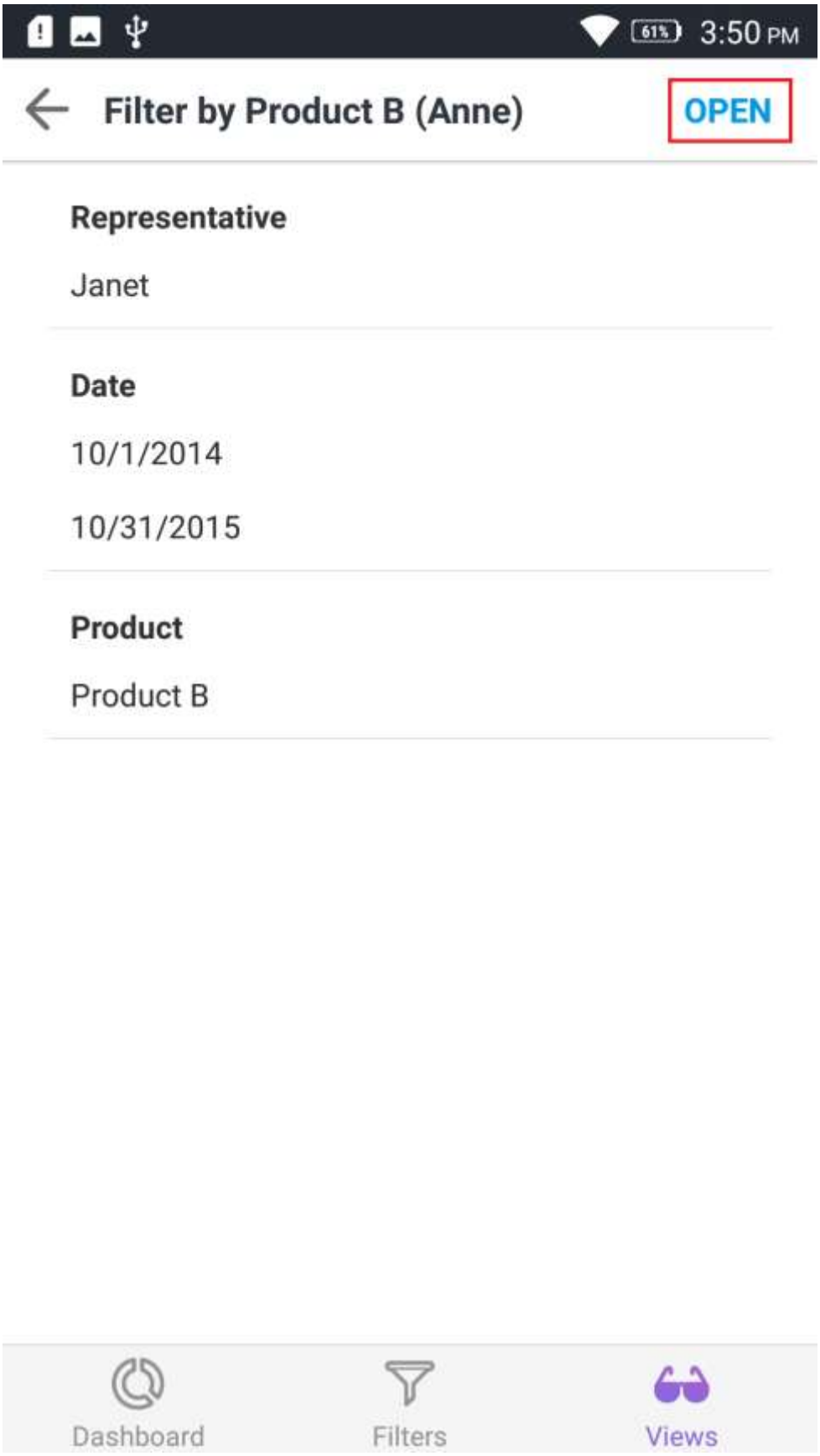


Apply Views

You can apply the views into the current dashboard by selecting the view in the Dashboard Views list

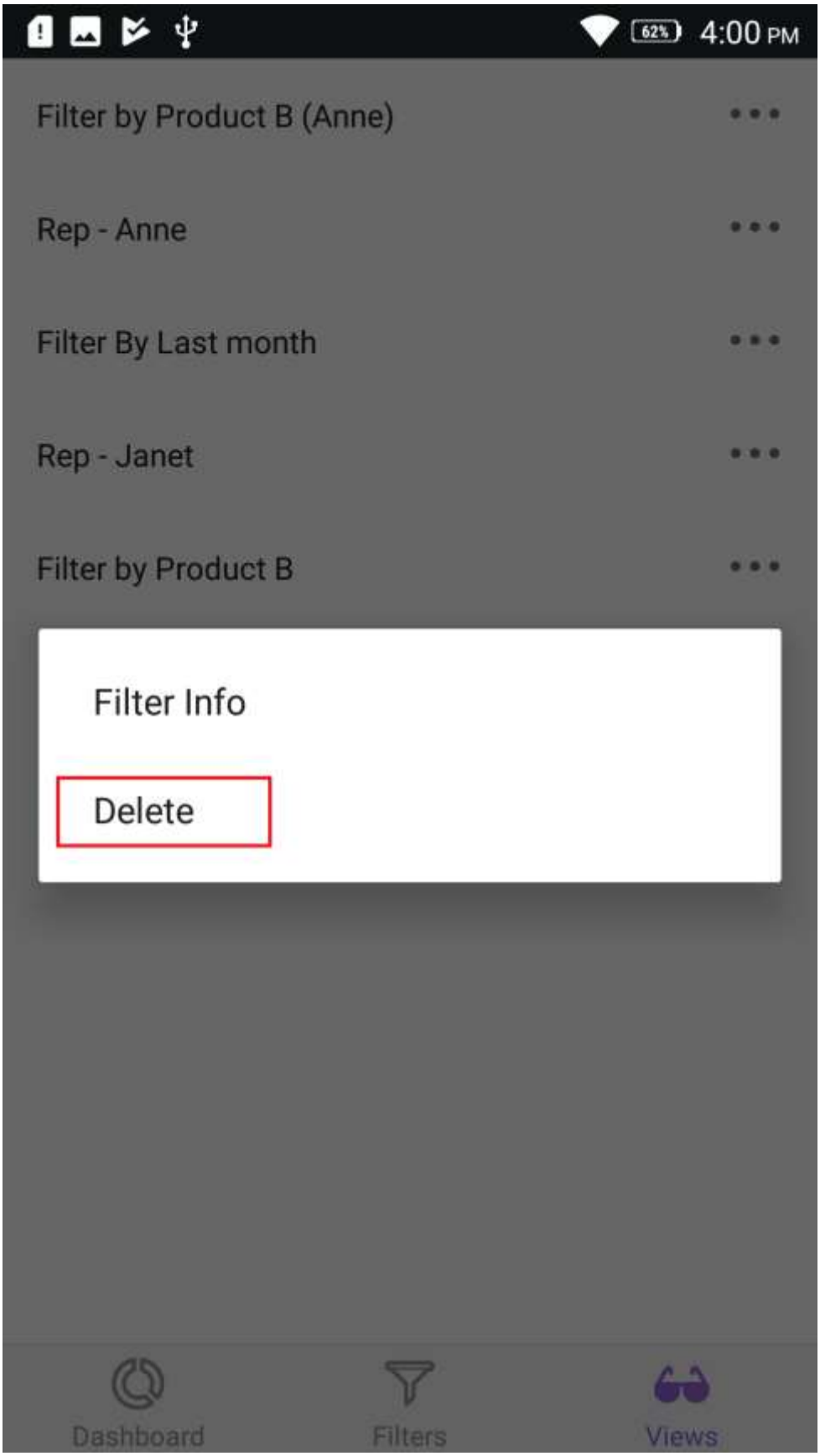


or you can apply it by tabbing Open in the filter information.



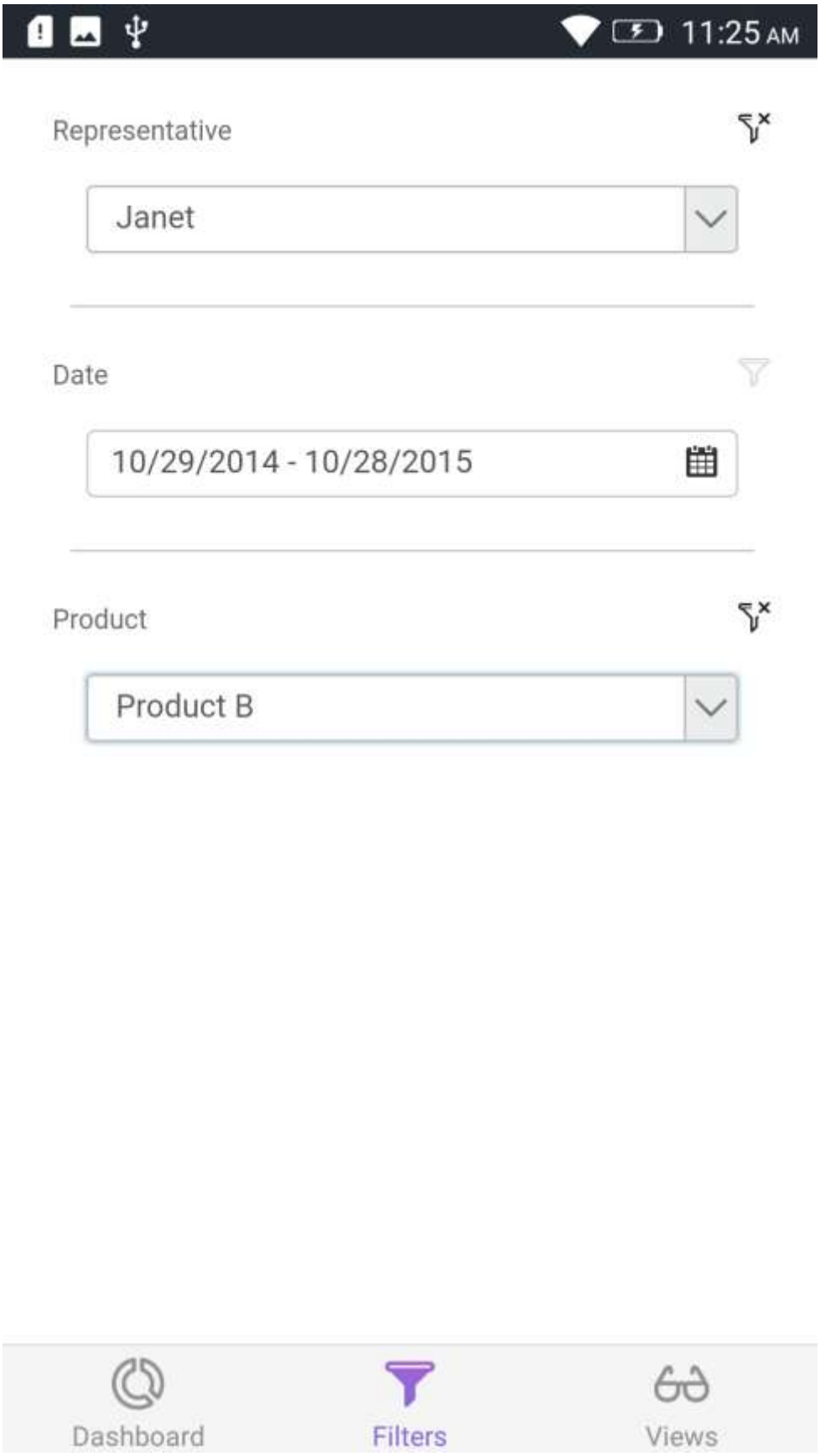
Delete Dashboard Views

Choose the Delete option in the context menu to delete the Dashboard View from the dashboard.



Dedicated filter

The dashboard filters will be displayed in the separate Filters tab for the dashboards which have dedicated filters configured.



List Widgets

Widgets are listed in the widgets page in the Syncfusion Dashboard Mobile app.

Searching, Filtering and Sorting - You can search and sort the widgets based on Owner name and Modified date, to quickly view the dashboard.

View Widgets

You can view and interact with the widgets using Syncfusion Dashboard Mobile app.

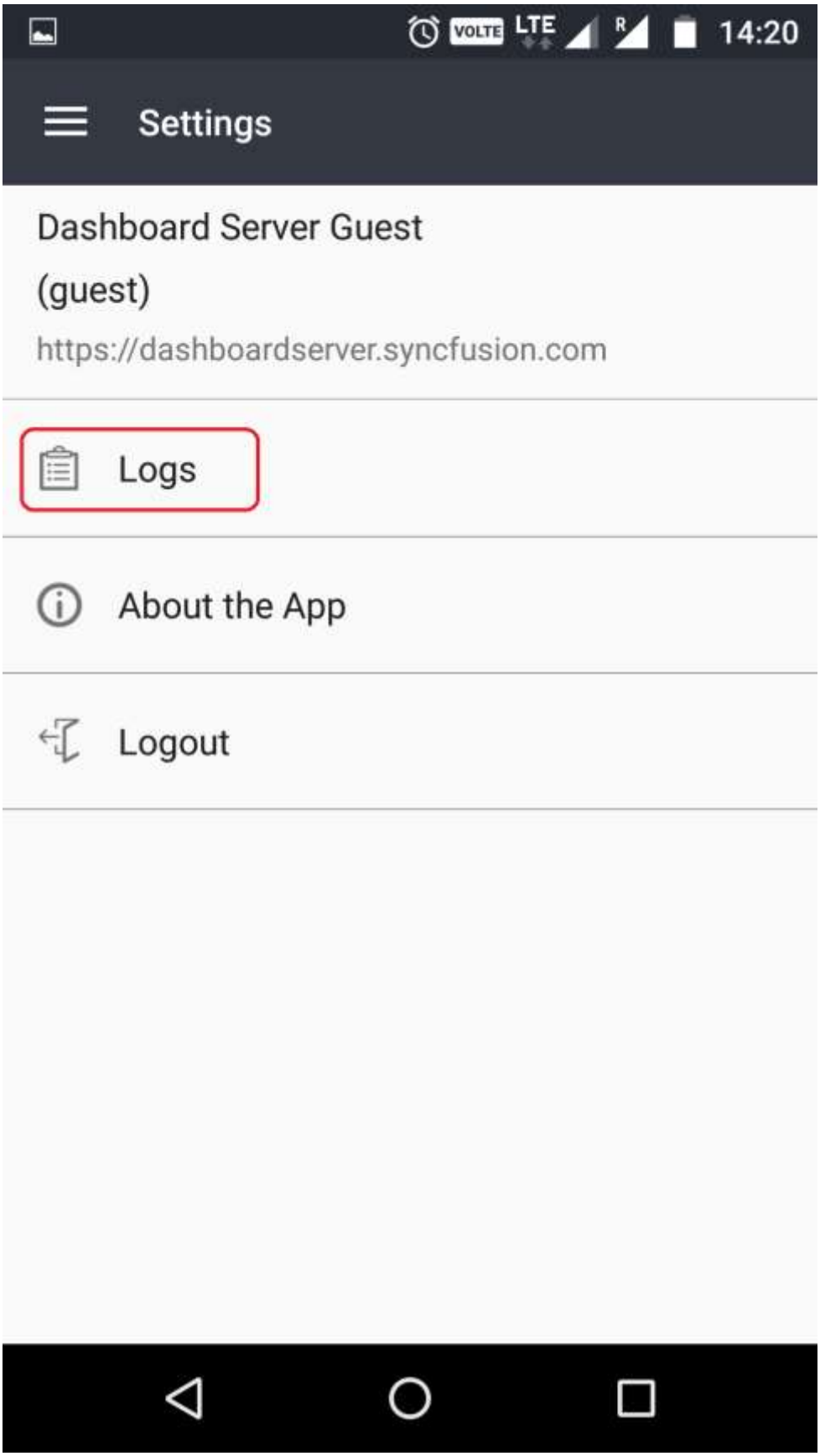
Export Widgets - Widgets can be exported to image, PDF, Excel and CSV file formats.

Settings

The settings page will have information about the user, error logs and logout option.

Logs

This page will have the error logs from the starting of the application. Also we have an option to clear the logs.



Here, you can share the error logs with our development team. Please send us your feedback and suggestion to support@syncfusion.com.

Dashboard Platform SDK

Overview

The Syncfusion Dashboard Platform SDK includes a Dashboard Viewer HTML 5 control that can be embedded within your applications.

Key features

Embed dashboards within your applications --- You can embed dashboard within your application which may be ASP.NET,ASP.NET Core,ASP.NET MVC,Angular 1,Angular 2,TypeScript,Aurelia,LightSwitch HTML,PHP,UWP, Windows Forms and WPF. using Dashboard Viewer HTML 5 control.

Cost effective licensing --- Our licensing is cost effective for both small and large teams. Please contact us for more details.

Create a support incident

If you are still not able to find the information that you are looking for in the self-help resources mentioned above then please contact us by creating a support ticket.

System Requirements

This section explains the system requirements to run Syncfusion Dashboard Platform SDK.

Hardware Requirements

The following minimum hardware requirements are necessary to run the Syncfusion Dashboard Platform SDK:

- Processor - Dual Core 32-bit CPU (x86)
- Hard disk - 2 GB
- RAM - 2 GB

Software Requirements

The following minimum software requirements are necessary to run the Syncfusion Dashboard Platform SDK:

- Operating System - Windows 7, 8+
- Microsoft Visual Studio 2012 or later
- IIS Express
- Browser - Internet Explorer 9+, Microsoft Edge, Mozilla Firefox 22+, Chrome 17+, Opera 12+, Safari 5+

Note: In Internet Explorer 11, if you enabled the Enterprise mode, make sure that you have disabled it. For more details, follow the link.

Note: In the Internet Explorer, make sure that you have turned off the Compatibility Settings for the intranet sites. To know how to turn off the Compatibility Settings, follow the link.

The following minimum software requirements are necessary in a data server (can be local or remote) for respective server connection types:

- Microsoft SQL Server - Microsoft SQL Server 2005+
- Microsoft SQL Server Analysis Services - Microsoft SQL Server 2012+
- PostgreSQL – PostgreSQL Server 9.x+

- Spark SQL - Syncfusion Big Data Cluster Manager (Spark Server should be started through Service Manager after installation. It will be running on port 10001 by default.)
- Hive - Syncfusion Big Data Cluster Manager (Hive Server should be started through Service Manager after installation. It will be running on port 10000 by default.)
- SQL through ODBC Connection - SQL Native Client or SQL Server Native Client 10.0 (Any one of the these drivers need to be installed and database need to be setup as discussed here).
- MySQL through ODBC Connection - MySQL ODBC 5.3 Unicode Driver (The driver need to be installed and database need to be setup as discussed here).
- Oracle through ODBC Connection - Microsoft ODBC for Oracle or Oracle in OraClient 11g_home1 (Any one of the these drivers need to be installed and database need to be setup as discussed here).
- Access through ODBC Connection - 2007 Office System Driver: Data Connectivity Components (The driver need to be installed and database need to be setup as discussed here).

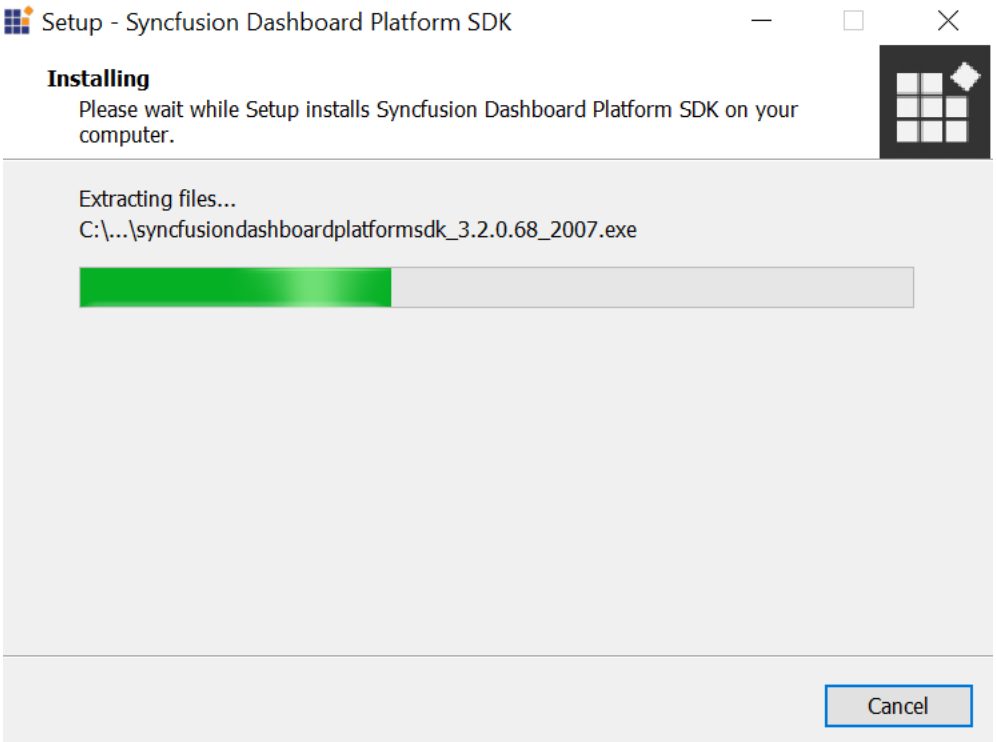
Installation and Deployment

This section briefly illustrates the installation and deployment procedure of Syncfusion Dashboard Platform SDK.

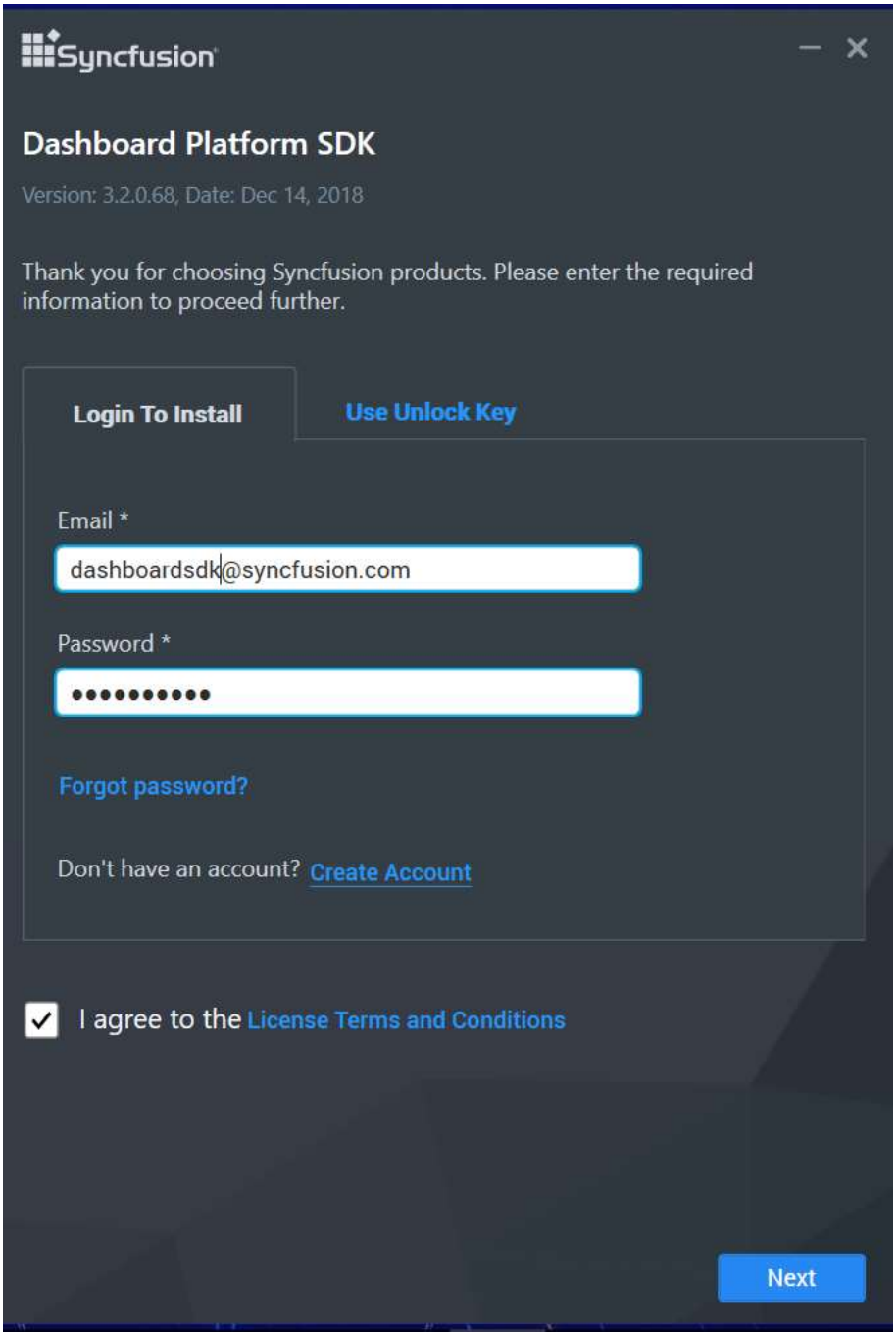
Installing Dashboard Platform SDK

To learn about the system requirements needed to install the Syncfusion Dashboard Platform SDK in your machine, see System Requirements section.

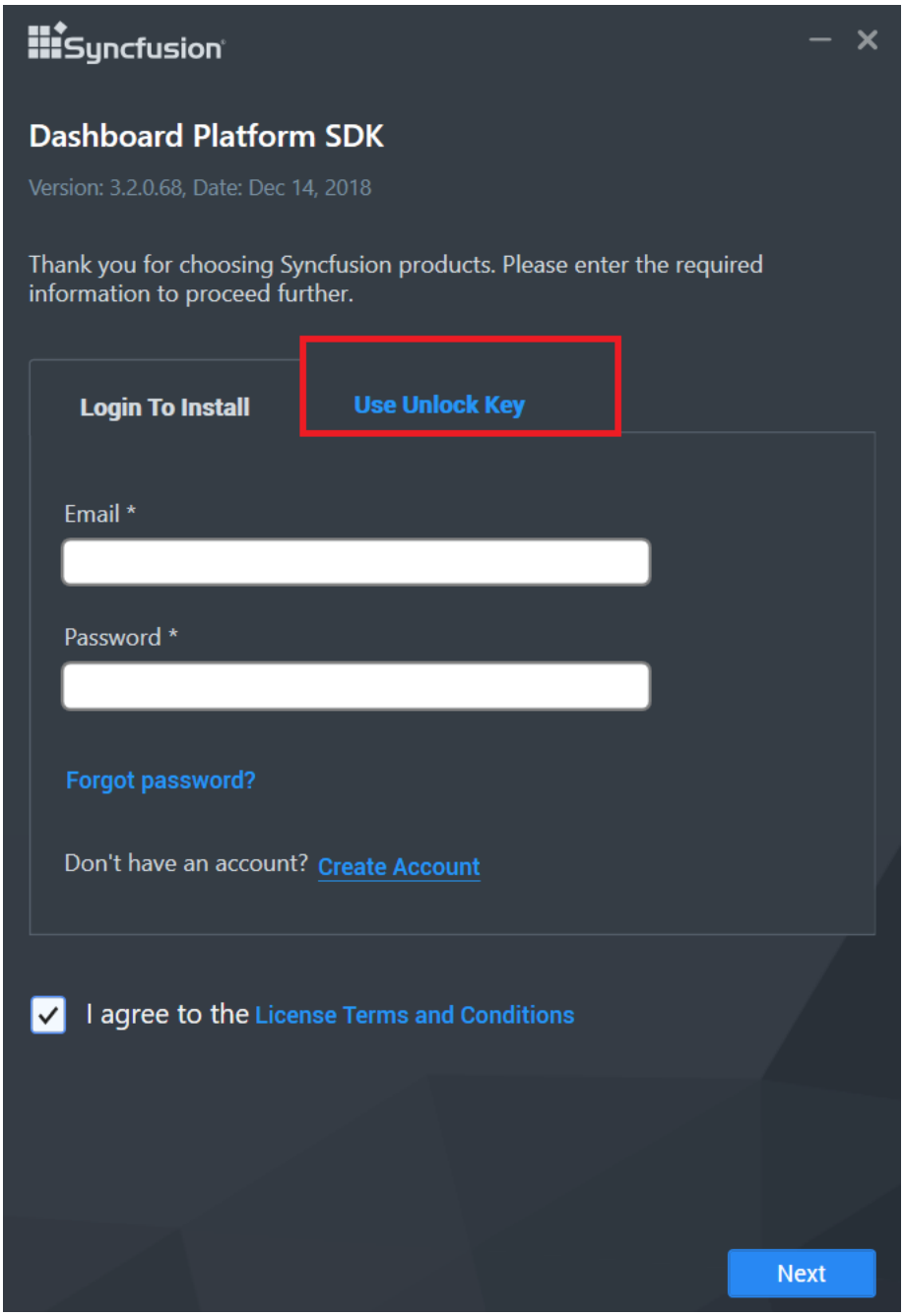
Run the saved installer either through clicking the Run button or by double-clicking the EXE file from the saved location.

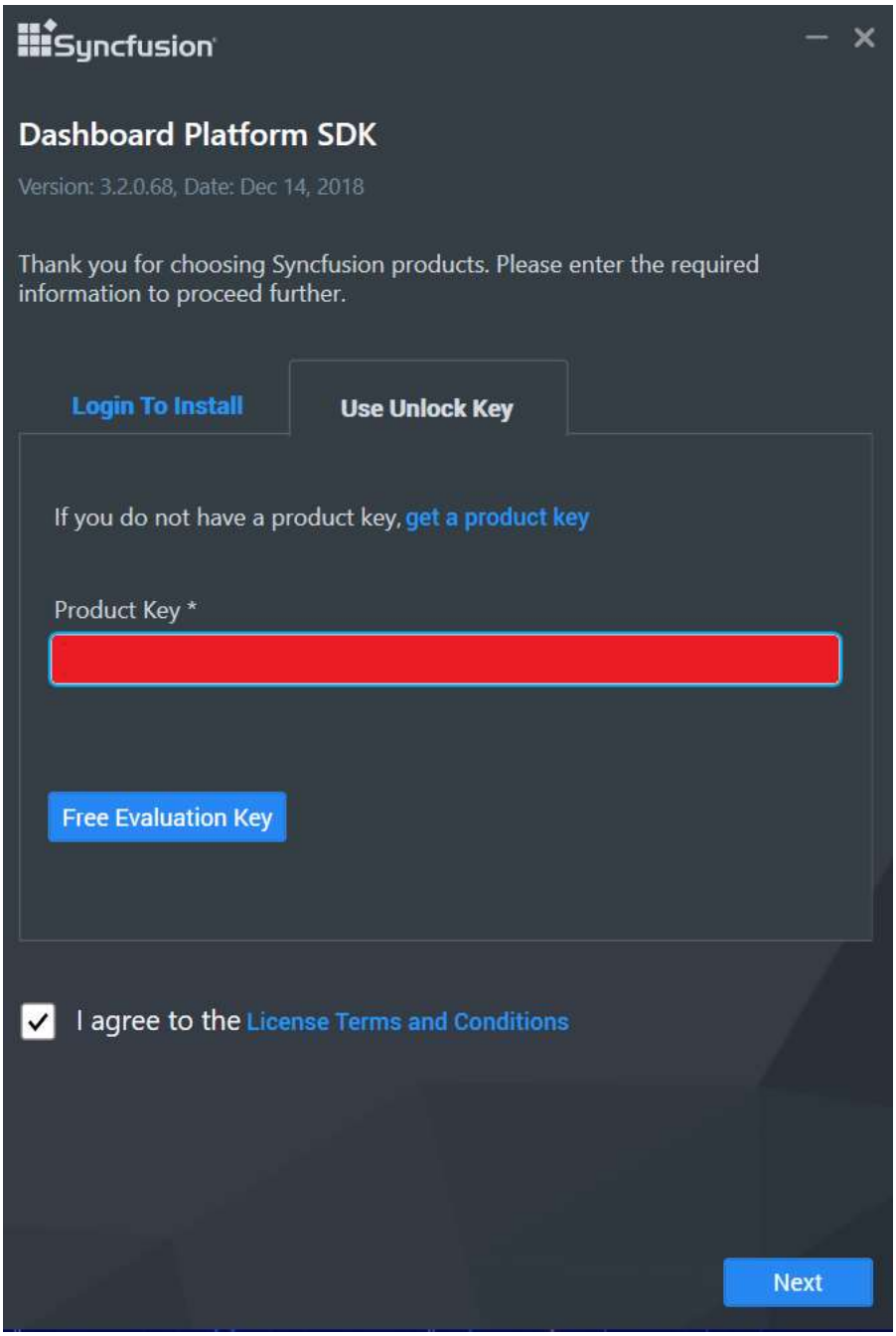


Run the Dashboard Platform SDK Installer and type in the credentials of your Syncfusion account to unlock the setup.

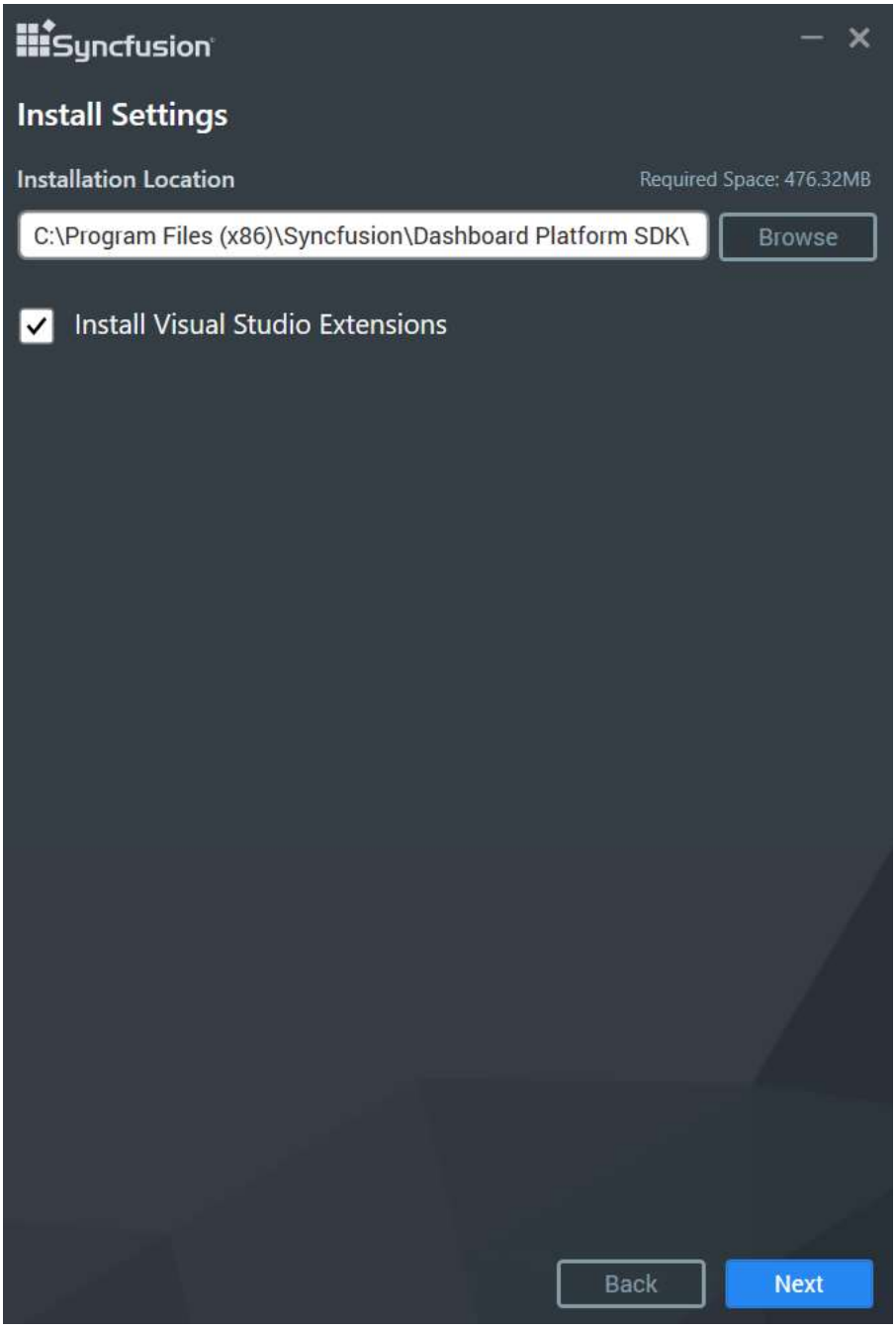


You can alternatively type in the unlock key that has been sent to your registered e-mail address to unlock the setup by selecting the Use Unlock Key option.



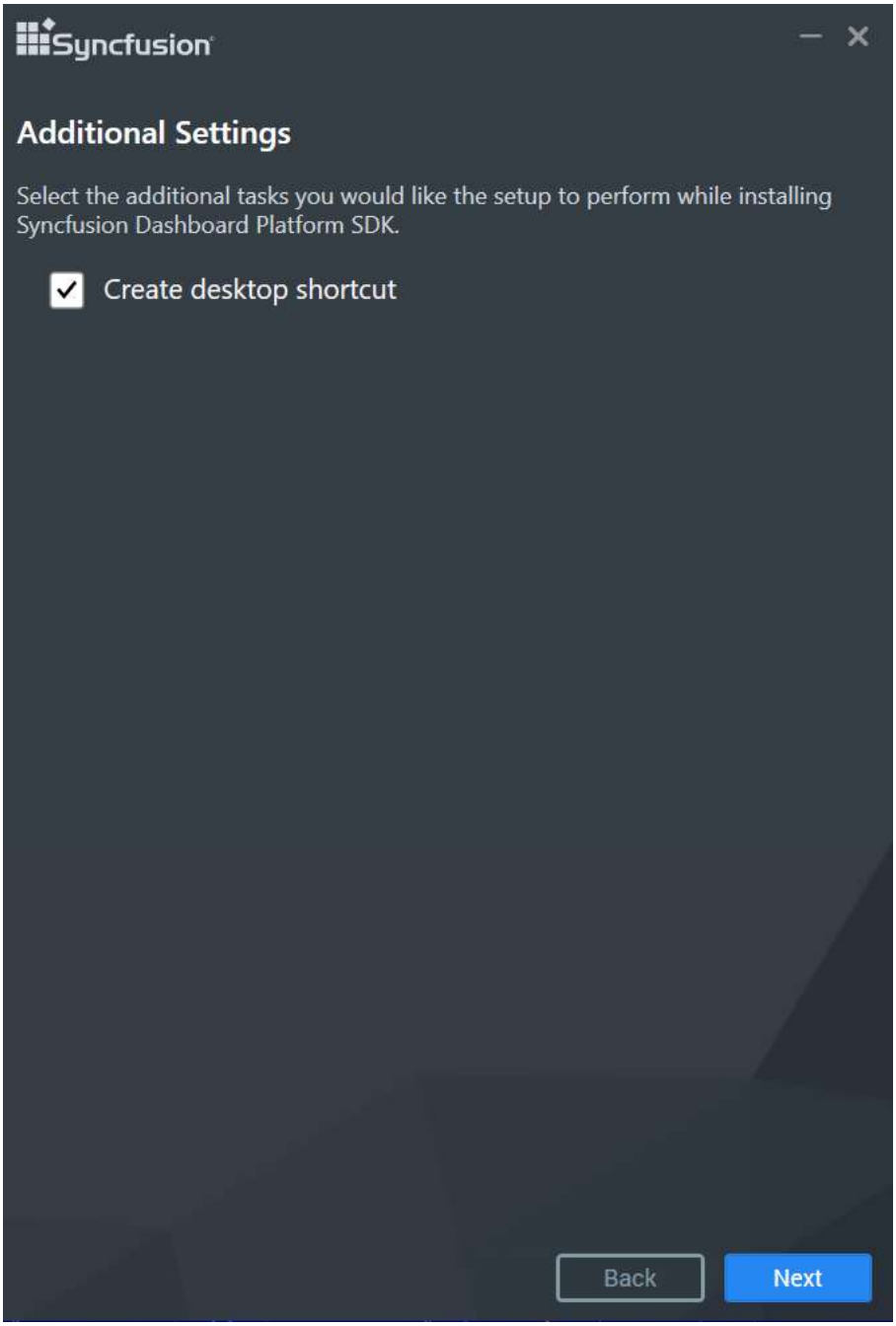


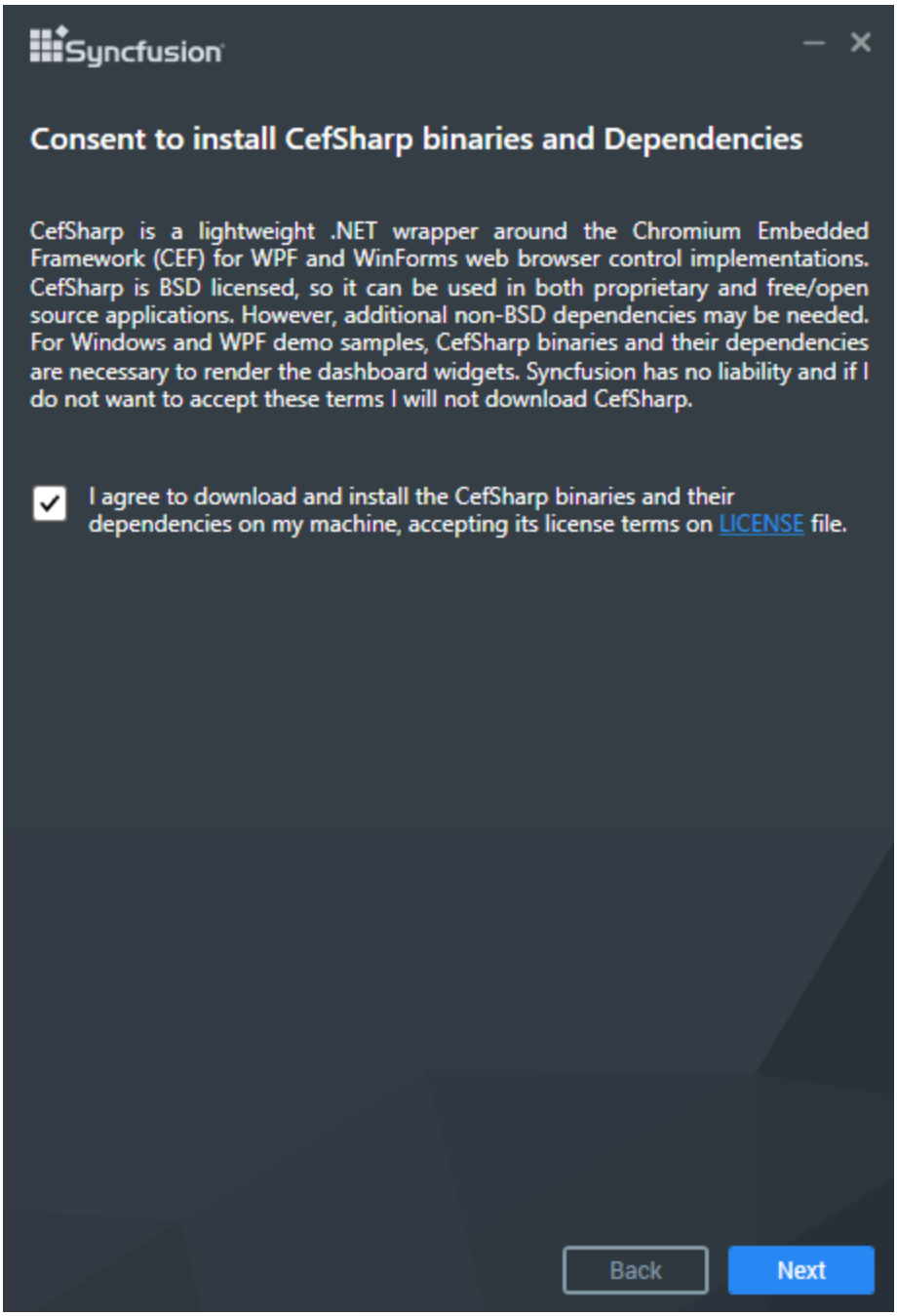
You can check the License Agreement of Dashboard Platform SDK by clicking on the License Terms and Conditions.



Browse to the location where you would like to install the Dashboard Platform SDK application and click NEXT.

Perform the additional task desktop shortcuts creation. If you want to perform the additional tasks, you can check the option. Otherwise, you can uncheck it and click NEXT.

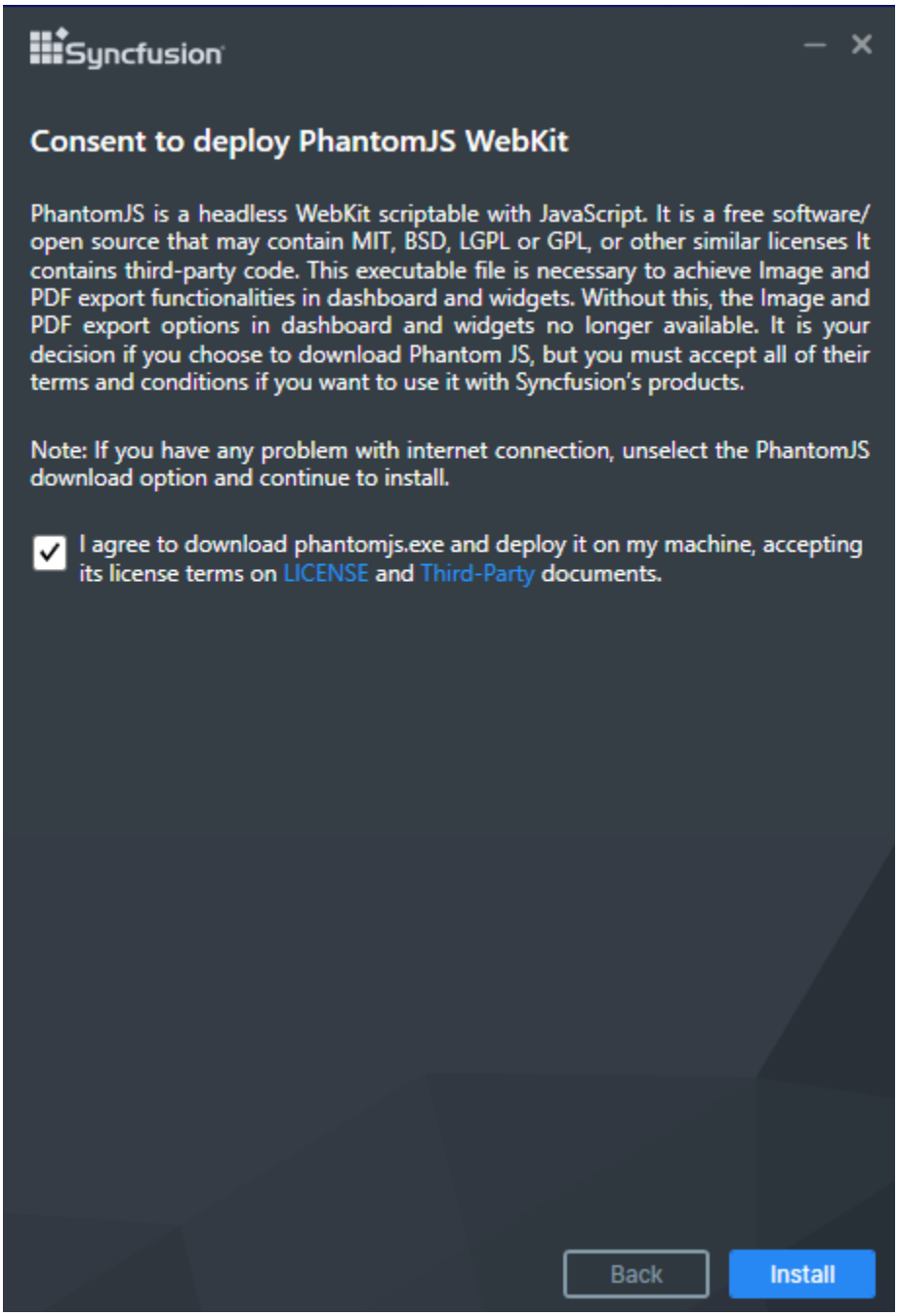




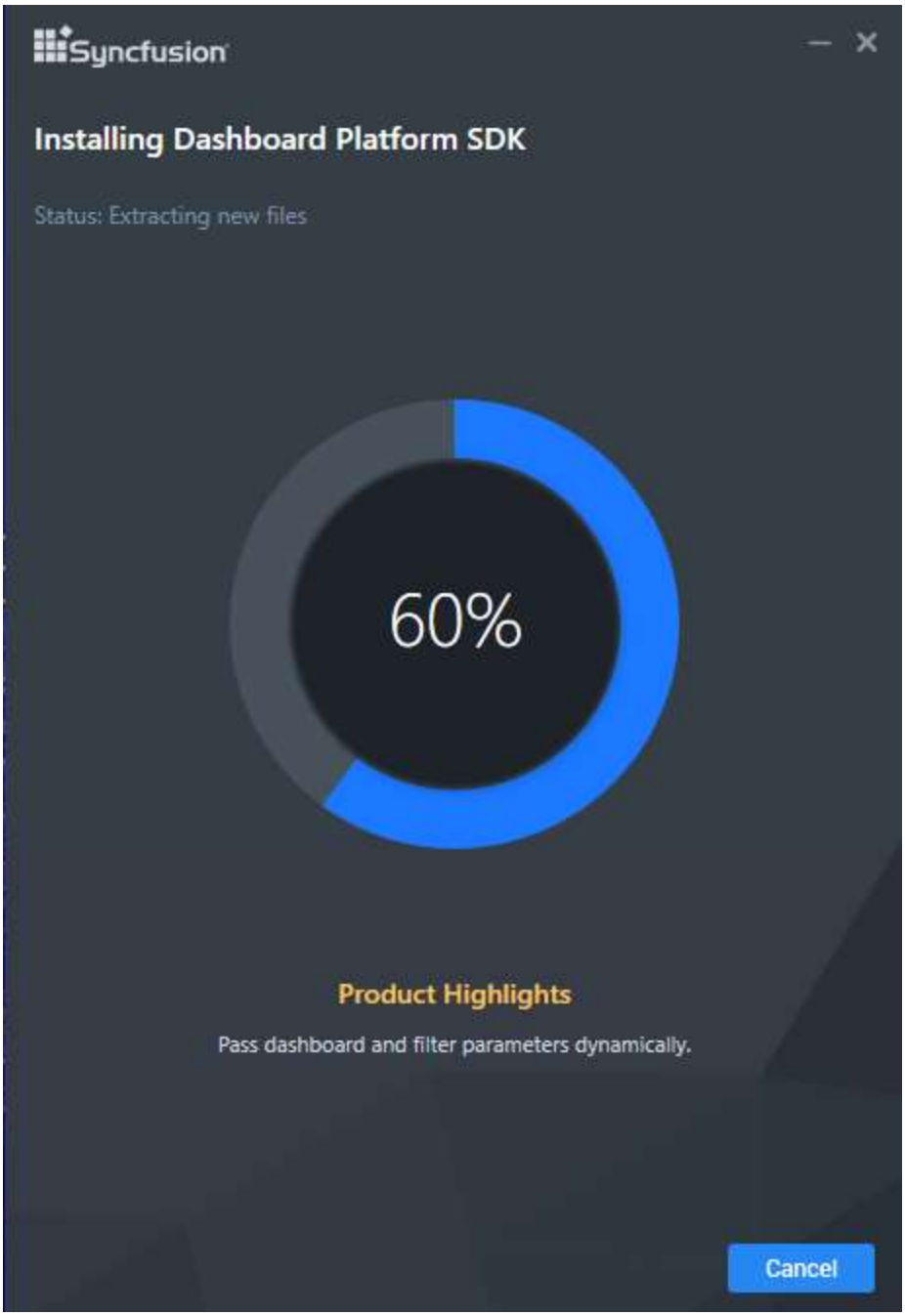
Read and accept the license terms through checking the option LICENSE for download and install CefSharp and click NEXT.

Information: PhantomJS is a headless WebKit scriptable with JavaScript. This is a free software/open source, and it may contain MIT, BSD, LGPL, or GPL, or other similar licenses that contain third-party code. This executable file is necessary to achieve Image and PDF export functionalities in the Dashboard and widgets. Without this file, the image and PDF export options in the Dashboard and widgets will no longer be available. If you choose to download PhantomJS, must accept all terms and conditions to use it with Syncfusion's products.

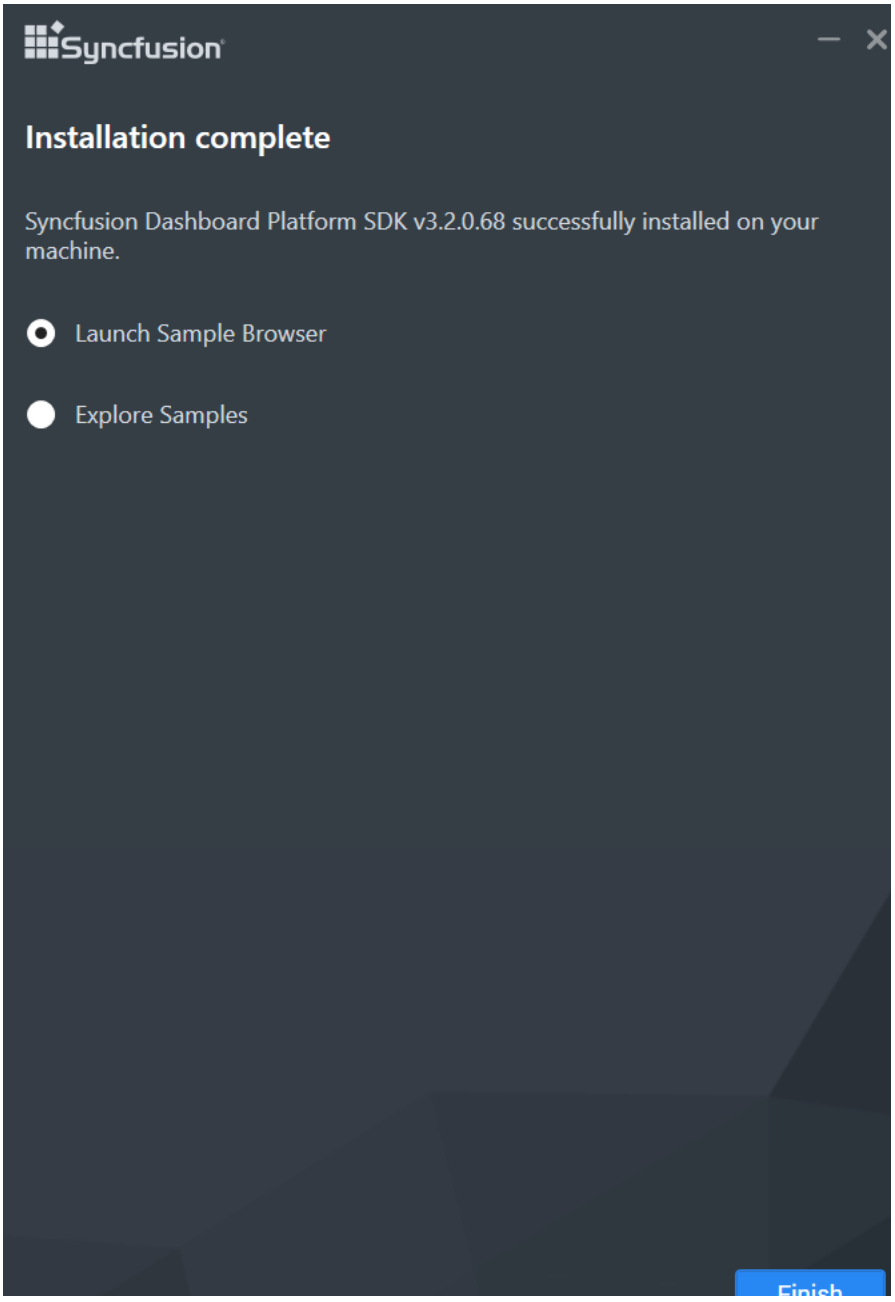
Note: If you have any problem with internet connection or do not have internet connection, unselect the PhantomJS download option and continue to install. To manually install the PhantomJS, please refer this.



Read and accept the license and third-party terms and conditions through checking the option LICENSE and Third-party for install PhantomJS and click INSTALL.



Now the installation begins. You can cancel the installation anytime through pressing CANCEL, if you prefer.



On successful installation, the above screen appears. Click Finish to close the installation wizard and check any one radio button for run Sample Browser or Explore Samples.

Silent Installation

1.Double click the Syncfusion Dashboard Platform SDK setup. 2.Syncfusion Dashboard Platform SDK setup will be extracted in Temp location (%temp%).

3.Copy the extracted Dashboard Platform SDK setup to some other location and cancel the installation.
4.Open the command prompt with administrative privileges and run the extracted Dashboard Platform SDK setup with the following arguments.

Arguments:

Dashboard Platform SDK V3.1:

/Install silent /InstallPath:{InstallationPath} /pidkey:{unlock_key} /Log "{LogFilePath\filename.log}"

Example:

/Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard /pidkey:@1243453sdffdfvv /Log "C:\Program Files (x86)\New\Install.log"

```
D:\>syncfusiondashboardplatformsdk_3.1.0.113_2133.exe /Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard
/pidkey:@1243453sdffdfvv  /Log "C:\Program Files (x86)\New\Install.log"
```

Dashboard Platform SDK V3.2:

/Install silent /InstallPath:{InstallationPath} /pidkey:{unlock_key} /isdesktopshortcut:TRUE /Log "{LogFilePath\filename.log}"

Example:

/Install silent /InstallPath:C:\Program Files (x86)\New\DashboardPlatformSDK /pidkey:@1243453sdffdfvv /isdesktopshortcut:TRUE /Log “C:\Program Files (x86)\New\Install.log”

```
C:\>C:\Users\labuser\Desktop\SilentSetup\syncfusiondashboarddesigner_3.2.0.68_20
07.exe /Install silent /InstallPath:C:\Program Files (x86)\New\DashboardPlatform
SDK /pidkey:@1243453sdffdfvv /isdesktopshortcut:TRUE /Log "C:\Program Files (x86
)\New\Install.log"
```

Now, Syncfusion Dashboard Platform SDK will be installed in silent mode.

Consent to deploy PhantomJS WebKit

PhantomJS is a headless WebKit scriptable with JavaScript. It is a free software/open source that may contain MIT, BSD, LGPL or GPL, or other similar licenses It contains third-party code. This executable file is necessary to achieve Image and PDF export functionalities in dashboard, widgets and schedules. Without this, the Image and PDF export options in dashboard, widgets and schedules no longer available. It is your decision if you choose to download Phantom JS, but you must accept all of their terms and conditions if you want to use it with Syncfusion’s products.

To download PhantomJS application and deploy it on your machine, you should accept it's license terms on LICENSE and Third-Party document. Then, you can download PhantomJS by clicking here.

Once download completed, extract the zip file and then copy the PhantomJS application from the zip extracted location and paste it in the below mentioned install locations.

Install Locations:

1. {InstallPath}\Dashboard Platform SDK\Utilities\Windows Service

2. %localappdata%\Syncfusion\Dashboard\Samples\Common\Service

Examples:

1. C:\Program Files (x86)\Syncfusion\Dashboard Platform SDK\Utilities\Windows Service
2. C:\Users\John\AppData\Local\Syncfusion\Dashboard\Samples\Common\Service

Samples deployment

By installing the Syncfusion Dashboard Platform SDK, the samples are installed in the following location.

`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples`

The scripts, styles, and fonts that are required to run the dashboard application will be placed in the following location.

`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html`

This folder contains samples of different platforms such as ASP.NET, ASP.NET Core, ASP.NET MVC, Angular 1, Angular 2, TypeScript, Aurelia, LightSwitch HTML, PHP, UWP, Windows Forms, and WPF.

Note: Internet connection is required to run sample dashboards as their data need to be retrieved from a remote data server.

Service installation

With the installation of Syncfusion Dashboard Platform SDK, a windows service called `Syncfusion Dashboard Windows Service` is installed and started automatically to run as a background process. This service is essential for the Dashboard Viewer to render the dashboard. You can also stop and restart the service through the Task Manager, if required.

Hosting dashboard service as Windows service background process

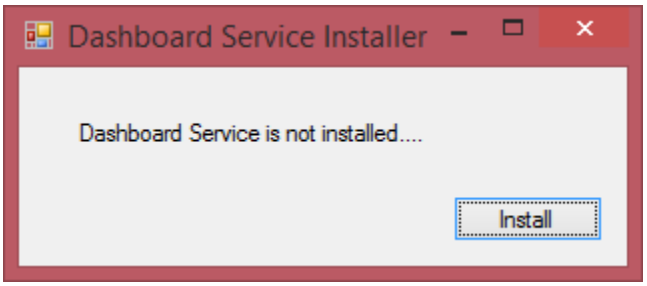
To install Dashboard Service as windows service, copy the folder `Windows Service` with its contents from the following location, and paste in a desired location and run the `Syncfusion Dashboard Service Installer.exe`.

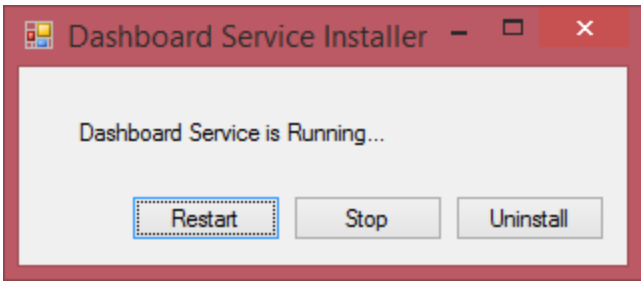
`C:\Program Files (x86)\Syncfusion\Dashboard Platform SDK\Utilities`

Note: 1. Running Syncfusion Dashboard Service Installer.exe requires administrator access for the current user.

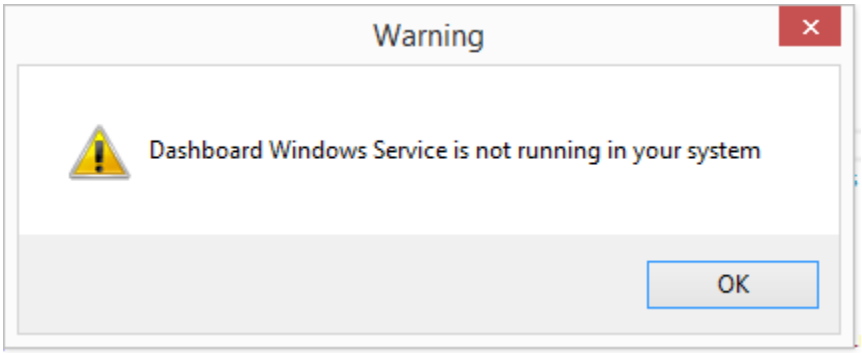
Note: 2. Dashboard service can be accessible only at local machine as this service is running as a Windows service background.

Note: 3. Dashboard Service as windows service is not recommended for the production environment. Refer here to learn about how to host the Dashboard Service in IIS.





You can stop/restart/uninstall the service through this EXE when you deal with any error related to service as follows.



Hosting dashboard service in IIS express

To host Syncfusion dashboard service in IIS express and run on deployment machines, copy the folders DashboardServiceInstaller and Service along with their contents from the following location, and paste in a desired location in the deployment machine and run the SyncfusionDashboardServiceInstaller-IISExpress.exe available in the DashboardServiceInstaller folder.

```
%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common
```

Hosting dashboard service in IIS

To host Syncfusion Dashboard Service in IIS and run on deployment machines, copy over the folders `DashboardServiceInstaller` and Service along with their contents from the following location, and paste in a desired location in the deployment machine and run the `SyncfusionDashboardServiceInstaller-IIS.exe` available in the `DashboardServiceInstaller` folder.

```
%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common
```

`SyncfusionDashboardServiceInstaller-IIS.exe` can be executed using command prompt to host the service in the specified port after copying the required file.

Command

```
C:\Windows\system32>[Path of SyncfusionDashboardServiceInstaller-IIS.exe] [Portnumber]
```

Example:

```
C:\Windows\system32> C:\Users\administrator\AppData\Local\Syncfusion\Dashboard Platform SDK\DashboardServiceInstaller\SyncfusionDashboardServiceInstaller-IIS.exe 400
```

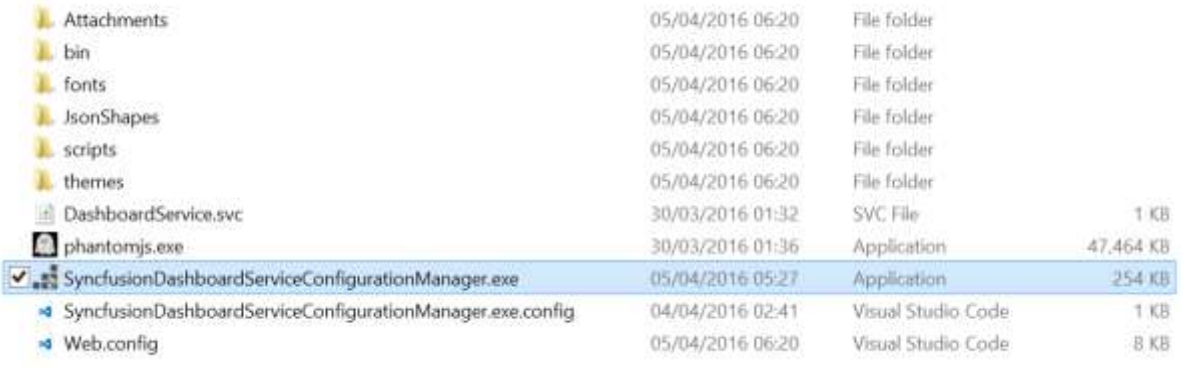
Note: 1. SyncfusionDashboardServiceInstaller-IIS.exe execution requires administrator mode.

2. Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.Refer here to learn about how to host the Dashboard Service in IIS manually.

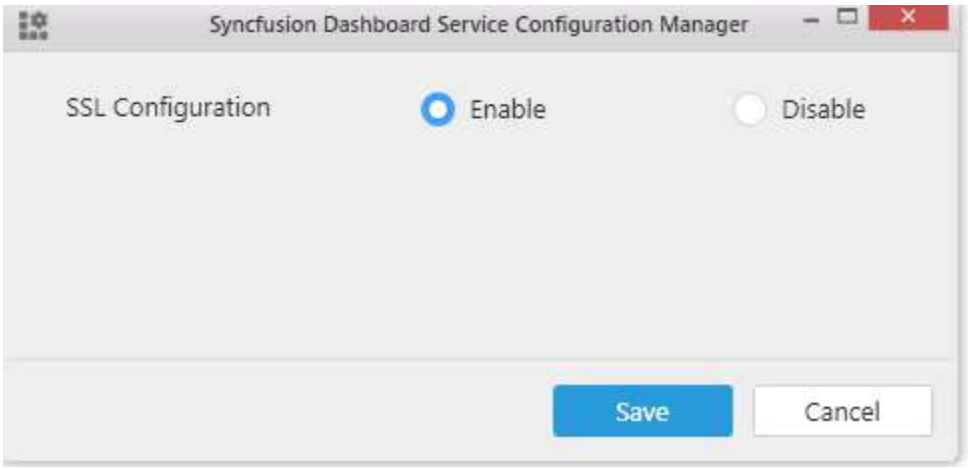
Configuring SSL for Dashboard Service

To configure SSL for Dashboard Service, run the Syncfusion Dashboard Service Configuration Manager application from the following location.

```
%localappdata%\Syncfusion\Dashboard Platform SDK\Service
```



Choose the required type of configuration and click Save.



Note: SyncfusionDashboardServiceConfigurationManager.exe execution requires administrator mode.

Restart the service in IIS Installer after configuring the SSL. The previous steps are not applicable for Window service.

Configuring Syncfusion Dashboard Viewer JavaScript Bower Packages

Overview

Bower is a package manager for the web. Syncfusion Dashboard JavaScript Components allows you to use the Syncfusion Dashboard Viewer Control in an efficient way.

Information: Syncfusion Dashboard JavaScript Components bower package is available as public Git Repository and also registered as Syncfusion-Dashboard-Javascript in the Bower registry.

Bower Installation

To configure the Bower in your machine, install node, npm and git. For more information to configure the Bower package please refer the official site for bower.

Syncfusion Dashboard JavaScript Components Bower package can be configured in the following ways.

1. Using command prompt
2. Using bower.json file
3. From local directory

Using command prompt

Perform the following steps to install Syncfusion Dashboard JavaScript Components using command prompt in your web application:

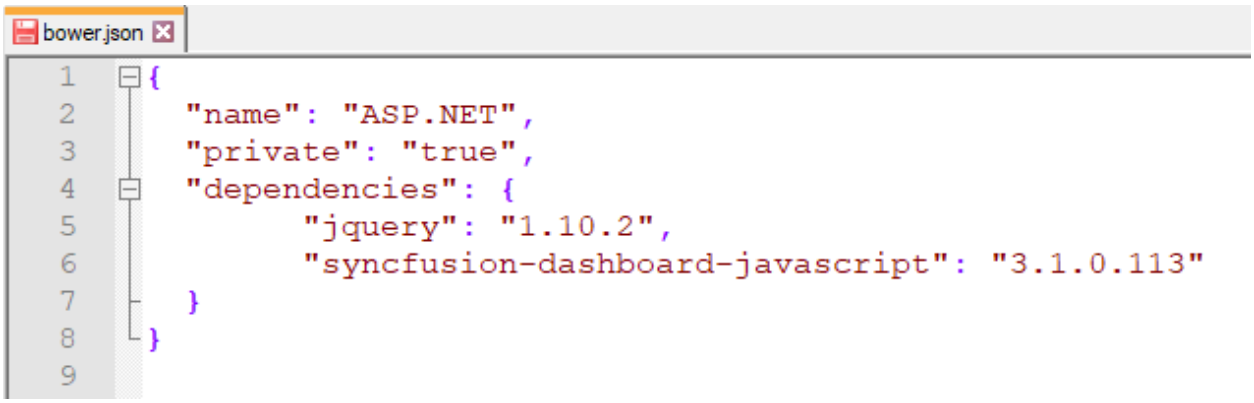
1. Open your web project's location in a command prompt window.
2. Run the command Bower install .

HTML

```
bower install syncfusion-dashboard-javascript
```

Using bower.json file

You can add the packages to the bower.json file by specifying the package name. This will install/restore the packages to your project. Refer to the following image.



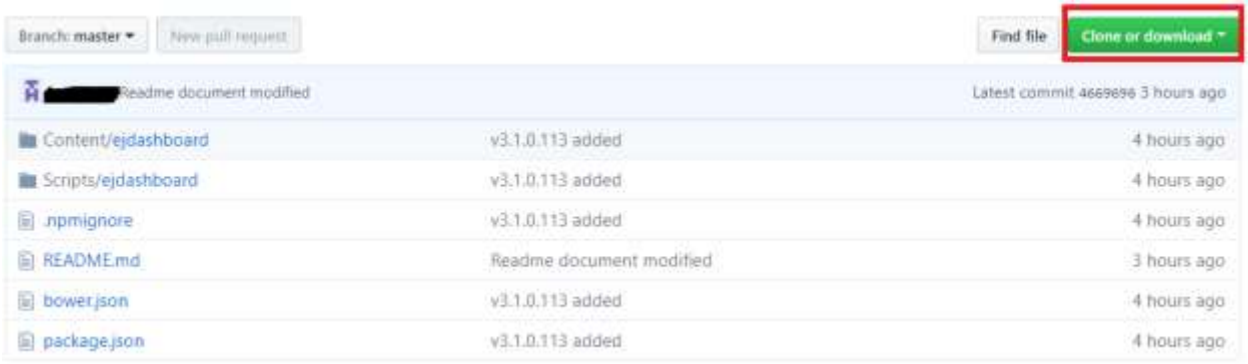
Note: ASP.NET 5 (preview) projects have bower.json file by default. If your project does not have bower.json file, run the following command from your project directory by Command prompt.

'bower init'

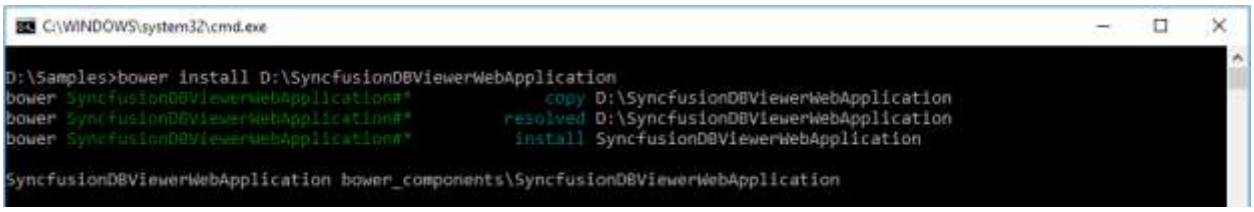
From local directory

To refer the Syncfusion Dashboard JavaScript Components from local directory, follow the given steps below:

1. Navigate to the Syncfusion Dashboard JavaScript Components repository on GitHub and download the repository as zip by clicking the "Download ZIP" button and extract the contents in your computer's local directory.



2. Run the install command by providing the package content's location.



Bower Update

To update the installed Bower packages, run the command Bower update .

HTML

```
bower update syncfusion-dashboard-javascript
```

Self-help Resources

The Syncfusion Dashboard team provides different self-help resources for making the dashboard embed process simpler and quicker. You can get quick answers for your basic queries about the features from existing resources such as KB articles and forums.

Samples Demos

The different sets of dashboards are designed and published in the demo server website for dashboard platform users to demonstrate various features in the Syncfusion Dashboard Platform SDK through its APIs. You can view the dashboards from the following link.

https://dashboardsdk.syncfusion.com/

Videos

The following page lists all available videos about getting started and using the Syncfusion Dashboard Platform SDK related features.

Videos Link: https://www.syncfusion.com/products/dashboard/videos

Frequently asked questions

You can instantly find answers for the most frequently asked questions about the Syncfusion Dashboard platform through the KB articles published by our Syncfusion team and public forums.

Syncfusion Knowledge Base Articles: https://www.syncfusion.com/kb/dashboard/dashboardsdk

Syncfusion Forums: https://www.syncfusion.com/forums/dashboard

Syncfusion technical blogs

Refer to the technical blog website to get the detailed technical blogs about Syncfusion Dashboard features.

https://blog.syncfusion.com/

Release history

The release information of all the public versions of the Dashboard platform can be referred in the following link.

https://www.syncfusion.com/products/release-history/dashboard

Create a support incident

If you are still not able to find the information that you are looking for self-help resources mentioned above, please contact us by creating a support ticket.

While creating the support incidents, try to provide detailed explanation of your requirement with necessary screenshots or videos. For reporting any defect, consider providing the exact replication procedure and necessary log files. It will help our support team to reach you earlier with a better solution.

Getting Started

Overview

Syncfusion Dashboard Platform SDK includes AngularJS directives within the ej.widget.angular.min.js script file. All these directives have been encapsulated into a single module called ejangular.

Getting Started with DashboardViewer in AngularJS Application

To get start with the DashboardViewer control in AngularJS Framework, the following list of external dependencies are mandatory.

- jQuery - 1.10.2 and later versions
- AngularJS - angular latest versions

The required AngularJS script files such as angular.min.js and ej.widget.angular.min.js can be referred from the below CDN links:

- angular.min.js - http://cdn.syncfusion.com/js/assets/external/angular.min.js
- ej.widget.angular.min.js - http://cdn.syncfusion.com/{{ site.releaseversion }}/js/common/ej.widget.angular.min.js

These files can also be accessed from the following installed location.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JavaScript\ng1 app

Preparing HTML document

Create a new folder named angular1 app.

Create a new HTML file say dashboardviewer.html inside angular1 app folder and include the below initial code.

HTML

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta charset="utf-8" />
<title> </title>
<!--  Script & css reference -->
</head>
<body>
<!-- Place div element to create DashboardViewer -->
<script>
<!-- Add your custom scripts here -->
</script>
</body>
</html>
```

Adding Script and CSS references

Add the scripts and CSS references in the order mentioned in the following code example.

HTML

```
<!DOCTYPE html>
<html ng-app="DashboardCtrl">
<head>
<title>DashboardViewer Angular1 Getting Started</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0"
charset="utf-8" />
<link href="https://cdn.syncfusion.com/3.1.x.x/css/default-
theme/ej.dashboardViewer.all.min.css" rel="stylesheet" />
<script src="http://cdn.syncfusion.com/js/assets/external/jquery-
1.10.2.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
easing/1.4.1/jquery.easing.min.js"></script>
<script
src="http://cdn.syncfusion.com/js/assets/external/angular.min.js"></script>
<script
src="https://cdn.syncfusion.com/3.1.x.x/js/ej.dashboardViewer.all.min.js"></
script>
<script
src="http://cdn.syncfusion.com/15.1.0.41/js/common/ej.widget.angular.min.js"
type="text/javascript"></script>
<style>
body, html, #dashboard {
overflow: hidden! Important;
height: 100%;
width: 100%;
}
</style>
</head>
```

```
<body ng-controller="DashboardViewerCtrl" >
<!-- place div elements to create DashboardViewer -->
</script>
<!--Add custom scripts here -->
</script>
</body>
</html>
```

Control Initialization

Initialize DashboardViewer control inside div element using ej-dashboardviewer AngularJS directive. Its properties can be defined using e-prefix followed by the property name (For example, e-serviceUrl).

HTML

```
<!--Place div element to create DashboardViewer-->
<div>
<div id="dashboard" ej-dashboardviewer e-
serviceUrl="https://dashboardsdk.syncfusion.com/DashboardService/DashboardSe
rvice.svc" e-
dashboardPath="https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCar
SalesDashboard.sydx" >
</div>
</div>
```

Note: Provided online Dashboard Service URL and Dashboard Path are for demo purpose.

Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

Declare dependency for ejangular module within your AngularJS application through adding the following script code.

JS

```
<!--Add custom scripts here -->
<script>
angular.module('DashboardCtrl', ['ejangular'])
.controller('DashboardViewerCtrl', function ($scope) {
});
</script>
```

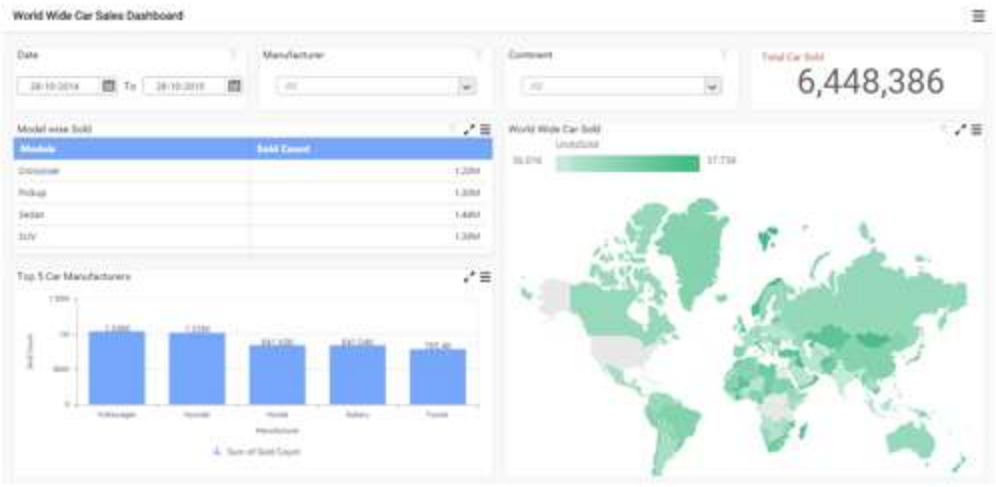
Binding Dashboard Service

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

On running the application, dashboard will be rendered like below.



Two Way Binding

When a widget model (ng-model) attribute is bound to a scope variable, it can reflect the changes in both ways (i.e. can change the value of the variable and as well display the value of the variable).

HTML

```
<!DOCTYPE html> <html ng-app="DashboardCtrl">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0"
charset="utf-8" />
<!--  Script & css reference -->
<style>
body, html, #dashboard {
overflow: hidden! Important;
height: 100%;
width: 100%;
}
body {
background-image: url("../common-images/images/right side BG
texture.png")!important;
}
.row .cols-prop-area {
height: 100px !important;
min-height: 406px;
width: 24.9146%;
}
</style>
</head>
<body style="height: 100%; width: 100%;" ng-
controller="DashboardViewerCtrl">
<div class="content-container-fluid" style="height: 100%; width: 100%;" >
<div class="row" style="height: 100%; width: 100%;" >
<div class="cols-sample-area" style="height: 100%; width: 73.5%; float:
left;" >
<div style="height: 100%; width: 100%;" id="dashboard" ej-dashboardviewer e-
serviceUrl="url" e-dashboardPath="dashboardPath" e-size-width="955" e-size-
height="587" >
</div>
</div>
```



```
<div id="sampleProperties" >
<div class="prop-grid">
<b>Dashboards:</b>
<br/>
<div class="row">
<div class="col-md-6">
Change Dashboard
</div>
<div class="col-md-5 aligntop">
<select ng-model="dashboardPath" data-ng-options="a.value as a.name for  a
in dashboards">
</select>
</div>
</div>
</div>
</div>
</div>
</div>
<script>
angular.module('DashboardCtrl', ['ejangular'])
.controller('DashboardViewerCtrl', function ($scope)
{
$scope.dashboardPath="https://dashboardsdk.syncfusion.com:3011//Dashboards//
WorldWideCarSalesDashboard.sydx";
$scope.url="https://dashboardsdk.syncfusion.com:3011/DashboardService/Dashbo
ardService.svc";
$scope. Dashboards =
[
{ id: 1, value:
"https://dashboardsdk.syncfusion.com:3011//Dashboards//Northwind Website
Analysis.sydx", name: "WebSite Dashboard" },
{ id: 2, value:
"https://dashboardsdk.syncfusion.com:3011//Dashboards//Northwind Products
and Suppliers.sydx", name: "Suppliers Dashboard" },
{ id: 3, value:
"https://dashboardsdk.syncfusion.com:3011//Dashboards//Northwind Traders
Sales Analysis.sydx", name: "Traders Dashboard" },
{ id: 4, value:
"https://dashboardsdk.syncfusion.com:3011//Dashboards//NorthWind Product
Details.sydx", name: "Product Dashboard" },
{ id: 5, value:
"https://dashboardsdk.syncfusion.com:3011//Dashboards//WorldWideCarSalesDash
board.sydx", name: "CarSales Dashboard" }
];
});
$("#sampleProperties").ejPropertiesPanel();
</script>
</body>
</html>
```

The following properties of ejDashboardViewer control supports model binding.

serviceUrl dashboardPath

Overview

Syncfusion Dashboard Platform SDK supports Angular Framework for the dashboard viewer widget.

Prerequisites

- Node JS (v6.x.x or higher).
- NPM (3.x.x or higher).
- Angular-Cli (1.5.X or higher).

Getting started With Webpack

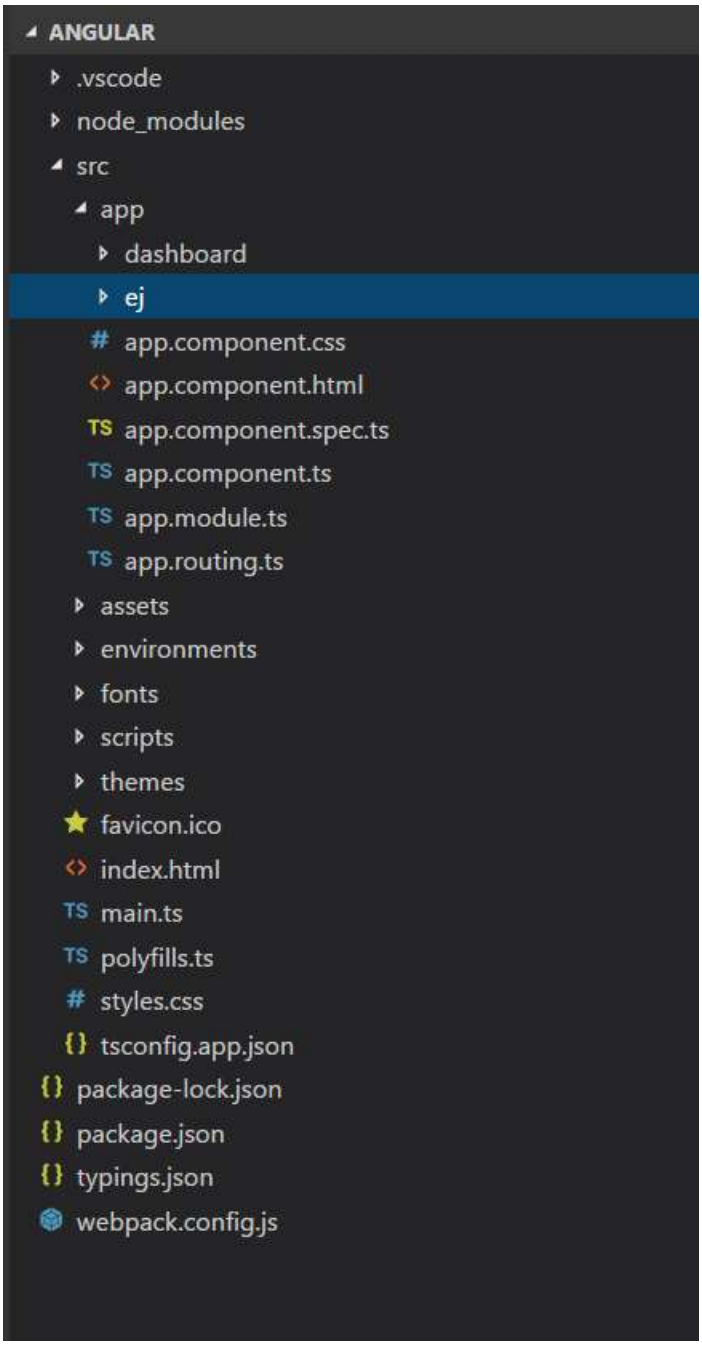
Webpack is a popular module bundler, a tool for bundling application source code in convenient chunks and for loading that code from a server into a browser.

This section guides you to start with Angular framework for the dashboard viewer through step-by-step instructions.

Create a new folder ANGULAR for creating an Angular application using web pack

under the Application folder create a files and folders based on the below structure

The following image illustrates the ANGULAR folder structure.



Copy and paste the following dependencies in the package.json file.

JSON

```
{
"name": "dashboard-angular-4",
"version": "1.2.1",
"repository": {},
"scripts": {
"start": "webpack-dev-server --port=4200",
"build": "webpack"
},
```



```
"private": true,
"dependencies": {
"@angular/animations": "4.3.1",
"@angular/common": "4.3.1",
"@angular/compiler": "4.3.1",
"@angular/core": "4.3.1",
"@angular/forms": "4.3.1",
"@angular/http": "4.3.1",
"@angular/platform-browser": "4.3.1",
"@angular/platform-browser-dynamic": "4.3.1",
"@angular/platform-server": "4.3.1",
"@angular/router": "4.3.1",
"bootstrap": "3.3.5",
"core-js": "2.4.1",
"ej-angular2": "^16.1.26",
"jquery": "1.12.4",
"jquery-validation": "1.16.0",
"jsrender": "^0.9.90",
"latest-version": "^3.1.0",
"require": "^2.4.20",
"rxjs": "5.1.0",
"web-animations-js": "2.2.2",
"syncfusion-javascript": "^16.1.37",
"zone.js": "0.8.4"
},
"devDependencies": {
"@angular/cli": "1.4.2",
"@angular/compiler-cli": "4.3.1",
"@types/bootstrap": "3.3.32",
"@types/ej.web.all": "^16.1.1",
"@types/jasmine": "2.5.38",
"@types/jquery": "^1.10.31",
"@types/node": "^6.0.73",
"codelyzer": "3.0.1",
"jasmine-core": "2.6.2",
"jasmine-spec-reporter": "4.1.0",
"karma": "1.7.0",
"karma-chrome-launcher": "2.1.1",
"karma-cli": "1.0.1",
"karma-coverage-istanbul-reporter": "1.2.1",
"karma-jasmine": "1.1.0",
"karma-jasmine-html-reporter": "0.2.2",
"protractor": "5.1.2",
"ts-node": "3.0.4",
"typescript": "2.3.3",
"webpack-dev-server": "~2.7.1",
"webpack": "~3.5.5",
"autoprefixer": "^6.5.3",
"css-loader": "^0.28.1",
"cssnano": "^3.10.0",
"exports-loader": "^0.6.3",
"file-loader": "^0.10.0",
"html-webpack-plugin": "^2.29.0",
"less-loader": "^4.0.5",
"postcss-loader": "^1.3.3",
"postcss-url": "^5.1.2",
"raw-loader": "^0.5.1",
```



```
"sass-loader": "^6.0.3",
"source-map-loader": "^0.2.0",
"istanbul-instrumenter-loader": "^2.0.0",
"style-loader": "^0.13.1",
"stylus-loader": "^3.0.1",
"url-loader": "^0.5.7",
"circular-dependency-plugin": "^3.0.0",
"webpack-concat-plugin": "1.4.0",
"copy-webpack-plugin": "^4.0.1"
}
}
```

Adding JavaScript and CSS reference

Copy the scripts, themes, and fonts folder from the Dashboard Platform SDK build installed location which is mentioned below and paste them in the src folder under the newly created angularDemo folder.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JavaScript\angular\src

Create a new folder as dashboard under the src/app folder. In the dashboard folder, create a new component as dashboard.component.ts and import the scripts/themes references in the order mentioned in the following code example.

TS

```
import 'jsrender';
import '../../scripts/jquery.easing.min';
import '../../../node_modules/syncfusion-
javascript/Scripts/ej/web/ej.web.all.min';
import '../../scripts/ej.dashboardViewer.all.min';
import '../../themes/default-theme/ej.dashboardViewer.all.min.css';
```

In the Index.html file, add the following code snippet in the <head> tag.

HTML

```
<style>
.e-dbrd-layout-wrapper{
height: 100% !important;
}
.e-dbrd-tab-wrapper {
height:100% !important;
}
#dashboard {
overflow: hidden !important;
height: 100% !important;
width: 100% !important;
position: absolute !important;
margin: 0;
}
</style>
```

Initializing and configuring the widget

Create a new folder as ej in the src/app folder.

Copy the angular dashboardviewer source files from the following Dashboard Platform SDK installed location and paste them into the ej folder.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JavaScript\angular\src\app\ej

Create another dashboard.component.html view file in the dashboard folder and initialize the ejDashboardViewer Angular component using the following code example.

HTML

```
<div>
<ej-dashboardviewer  id="dashboard" [serviceUrl]="url"
[dashboardPath]="dashboardPath"></ej-dashboardviewer>
</div>
```

Copy and paste the below code in dashboardviewer.component.ts file under the dashboard folder.

TS

```
import { Component } from '@angular/core';
import 'jsrender';
import '../../scripts/jquery.easing.min';
import '../../scripts/ej.dashboardViewer.all.min';
import '../../themes/default-theme/ej.dashboardViewer.all.min.css';
declare var $:any;
@Component({
selector: 'app-root',
templateUrl: './dashboard.component.html'
})
export class DashboardComponent {
public url; report;dashboardPath;dashboardParametersSettings;
constructor() {
this.url =
"https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc";
//Service URL
this.dashboardPath
="https://dashboardsdk.syncfusion.com//Dashboards//NorthWind Product
Details.sydx"; //path of the Dashboard
}
}
```

Note: The online dashboard service URL and dashboard path are provided for the demo purpose.

Make sure whether the given dashboard path should be accessible with the given Dashboard Service URL.

Binding dashboard service

To initiate the dashboard service instance, you can follow anyone of the following methods:

1. Hosting dashboard service in IIS.
2. Hosting dashboard service in IIS Express.
3. Hosting dashboard service as Windows service background process.

Information: Hosting dashboard service at IIS is recommended to production environment for object management and other memory management features.

Configuring routes for the router

Before adding router configuration for the above created EJ*DASHBOARDVIEWER*COMPONENTS component, refer to the Angular Routing configuration.

Create a new dashboard.module.ts file in the dashboard folder and paste the following code snippet in it.

TS

```
import { NgModule } from '@angular/core';
import { RouterModule } from '@angular/router';
import { CommonModule } from '@angular/common';
import { DashboardComponent } from './dashboard.component';
import { DashboardRoutes } from './dashboard.routing';
import { EJ_DASHBOARDVIEWER_COMPONENTS} from '../ej/web.all';
@NgModule({
imports: [
CommonModule,
RouterModule.forChild(DashboardRoutes),
],
declarations: [DashboardComponent,EJ_DASHBOARDVIEWER_COMPONENTS]
})
export class DashboardModule {
}
```

Edit the app.component.html file under the src folder for router configuration.

HTML

```
<router-outlet></router-outlet>
```

Import and declare the Syncfusion source component and EJ*DASHBOARDVIEWER*COMPONENTS component in the src\app\app.module.ts file using the following code snippet.

TS

```
import { NgModule } from '@angular/core';
import { BrowserAnimationsModule } from '@angular/platform-
browser/animations'; // this is needed!
import { RouterModule } from '@angular/router';
import { HttpModule } from '@angular/http';
import { APP_BASE_HREF } from '@angular/common';
import { EJ_DASHBOARDVIEWER_COMPONENTS } from
'./ej/dashboardviewer.component';
import { AppComponent }   from './app.component';
import { DashboardComponent } from './dashboard/dashboard.component';
import { AppRoutes } from './app.routing';
@NgModule({
imports: [
BrowserAnimationsModule,
RouterModule.forRoot(AppRoutes),
HttpModule
],
```



```
declarations: [
AppComponent
],
bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

Create a app.routing.ts file under the src\app folder and define the router in the src/app/app.routing.ts file.

TS

```
import { Routes } from '@angular/router';
export const AppRoutes: Routes = [{
path: '',
redirectTo: 'dashboard',
pathMatch: 'full',
},{
path: '',
children: [{
path: '',
loadChildren: './dashboard/dashboard.module#DashboardModule'
}]
}
];
```

Create a app.component.spec.ts file under the src\app folder and define the router in the src/app/app.component.spec.ts file.

TS

```
import { TestBed, async } from '@angular/core/testing';
import { AppComponent } from './app.component';
describe('AppComponent', () => {
beforeEach(async(() => {
TestBed.configureTestingModule({
declarations: [
AppComponent
],
}).compileComponents();
}));
it('should create the app', async(() => {
const fixture = TestBed.createComponent(AppComponent);
const app = fixture.debugElement.componentInstance;
expect(app).toBeTruthy();
}));
it(`should have as title 'app works!'`, async(() => {
const fixture = TestBed.createComponent(AppComponent);
const app = fixture.debugElement.componentInstance;
expect(app.title).toEqual('app works!');
}));
it('should render title in a h1 tag', async(() => {
const fixture = TestBed.createComponent(AppComponent);
fixture.detectChanges();
const compiled = fixture.debugElement.nativeElement;
```



```
expect(compiled.querySelector('h1').textContent).toContain('app works!');
}));
});
```

Also, create a dashboard.routing.ts file under the dashboard folder and place the following code snippet.

TS

```
import { Routes } from '@angular/router';
import { DashboardComponent } from './dashboard.component';
export const DashboardRoutes: Routes = [{
path: '',
children: [{
path: 'dashboard',
component: DashboardComponent
}]
}];
```

Refer the below codes to the required files created in the application

JAVASCRIPT

```
//In webpack.Config.js
const fs = require('fs');
const path = require('path');
const ConcatPlugin = require('webpack-concat-plugin');
const CopyWebpackPlugin = require('copy-webpack-plugin');
const ProgressPlugin = require('webpack/lib/ProgressPlugin');
const CircularDependencyPlugin = require('circular-dependency-plugin');
const HtmlWebpackPlugin = require('html-webpack-plugin');
const autoprefixer = require('autoprefixer');
const postcssUrl = require('postcss-url');
const cssnano = require('cssnano');
const { NoEmitOnErrorsPlugin, SourceMapDevToolPlugin, NamedModulesPlugin } =
require('webpack');
const { InsertConcatAssetsWebpackPlugin, NamedLazyChunksWebpackPlugin,
BaseHrefWebpackPlugin } = require('@angular/cli/plugins/webpack');
const { CommonsChunkPlugin } = require('webpack').optimize;
const { AotPlugin } = require('@ngtools/webpack');
const nodeModules = path.join(process.cwd(), 'node_modules');
const realNodeModules = fs.realpathSync(nodeModules);
const genDirNodeModules = path.join(process.cwd(), 'src', '$$_gendir',
'node_modules');
const entryPoints = ["inline","polyfills","sw-
register","styles","vendor","main"];
const minimizeCss = false;
const baseHref = "";
const deployUrl = "";
const postcssPlugins = function () {
// safe settings based on: https://github.com/ben-
eb/cssnano/issues/358#issuecomment-283696193
const importantCommentRe =
/@preserve|@license|[@#]\s*source(?:Mapping)?URL|^!/i;
const minimizeOptions = {
autoprefixer: false,
```

```
safe: true,
mergeLonghand: false,
discardComments: { remove: (comment) => !importantCommentRe.test(comment) }
};
return [
postcssUrl({
url: (URL) => {
// Only convert root relative URLs, which CSS-Loader won't process into
require().
if (!URL.startsWith('/') || URL.startsWith('//')) {
return URL;
}
if (deployUrl.match(/:\/\//)) {
// If deployUrl contains a scheme, ignore baseHref use deployUrl as is.
return `${deployUrl.replace(/\/$/, '')}${URL}`;
}
else if (baseHref.match(/:\/\//)) {
// If baseHref contains a scheme, include it as is.
return baseHref.replace(/\/$/, '') +
`/${deployUrl}/${URL}`.replace(/\/\/+/g, '/');
}
else {
// Join together base-href, deploy-url and the original URL.
// Also dedupe multiple slashes into single ones.
return `/${baseHref}/${deployUrl}/${URL}`.replace(/\/\/+/g, '/');
}
}
}),
autoprefixer(),
].concat(minimizeCss ? [cssnano(minimizeOptions)] : []);
};
module.exports = {
"resolve": {
"extensions": [
".ts",
".js"
],
"modules": [
"./node_modules",
"./node_modules"
],
"symlinks": true
},
"resolveLoader": {
"modules": [
"./node_modules",
"./node_modules"
]
},
"entry": {
"main": [
"./src\\main.ts"
],
"polyfills": [
"./src\\polyfills.ts"
],
"styles": [
```

```
"./src\\themes\\default-theme\\ej.dashboardViewer.all.min.css"
]
},
"output": {
"path": path.join(process.cwd(), "dist"),
"filename": "[name].bundle.js",
"chunkFilename": "[id].chunk.js"
},
"module": {
"rules": [
{
"enforce": "pre",
"test": /\.js$/,
"loader": "source-map-loader",
"exclude": [
/(\\|\/)node_modules(\\|\/)/
]
},
{
"test": /\.html$/,
"loader": "raw-loader"
},
{
"test": /\.(eot|svg|cur)$/,
"loader": "file-loader?name=[name].[hash:20].[ext]"
},
{
"test": /\.(jpg|png|webp|gif|otf|ttf|woff|woff2|ani|GIF)$/,
"loader": "url-loader?name=[name].[hash:20].[ext]&limit=10000"
},
{
"exclude": [
path.join(process.cwd(), "src\\themes\\default-
theme\\ej.dashboardViewer.all.min.css")
],
"test": /\.css$/,
"use": [
"exports-loader?module.exports.toString()",
{
"loader": "css-loader",
"options": {
"sourceMap": false,
"importLoaders": 1
}
},
{
"loader": "postcss-loader",
"options": {
"ident": "postcss",
"plugins": postcssPlugins
}
}
]
},
{
"exclude": [
```

```
path.join(process.cwd(), "src\\themes\\default-
theme\\ej.dashboardViewer.all.min.css")
],
"test": /\.scss$|\.sass$/,
"use": [
"exports-loader?module.exports.toString()",
{
"loader": "css-loader",
"options": {
"sourceMap": false,
"importLoaders": 1
}
},
{
"loader": "postcss-loader",
"options": {
"ident": "postcss",
"plugins": postcssPlugins
}
},
{
"loader": "sass-loader",
"options": {
"sourceMap": false,
"precision": 8,
"includePaths": []
}
}
]
},
{
"exclude": [
path.join(process.cwd(), "src\\themes\\default-
theme\\ej.dashboardViewer.all.min.css")
],
"test": /\.less$/,
"use": [
"exports-loader?module.exports.toString()",
{
"loader": "css-loader",
"options": {
"sourceMap": false,
"importLoaders": 1
}
},
{
"loader": "postcss-loader",
"options": {
"ident": "postcss",
"plugins": postcssPlugins
}
},
{
"loader": "less-loader",
"options": {
"sourceMap": false,
"paths": []
```



```
}
}
]
},
{
"exclude": [
path.join(process.cwd(), "src\\themes\\default-
theme\\ej.dashboardViewer.all.min.css")
],
"test": /\.styl$/,
"use": [
"exports-loader?module.exports.toString()",
{
"loader": "css-loader",
"options": {
"sourceMap": false,
"importLoaders": 1
}
},
{
"loader": "postcss-loader",
"options": {
"ident": "postcss",
"plugins": postcssPlugins
}
},
{
"loader": "stylus-loader",
"options": {
"sourceMap": false,
"paths": []
}
}
]
},
{
"include": [
path.join(process.cwd(), "src\\themes\\default-
theme\\ej.dashboardViewer.all.min.css")
],
"test": /\.css$/,
"use": [
"style-loader",
{
"loader": "css-loader",
"options": {
"sourceMap": false,
"importLoaders": 1
}
},
{
"loader": "postcss-loader",
"options": {
"ident": "postcss",
"plugins": postcssPlugins
}
}
```



```
]
},
{
"include": [
path.join(process.cwd(), "src\\themes\\default-
theme\\ej.dashboardViewer.all.min.css")
],
"test": /\.scss$|\.sass$/,
"use": [
"style-loader",
{
"loader": "css-loader",
"options": {
"sourceMap": false,
"importLoaders": 1
}
},
{
"loader": "postcss-loader",
"options": {
"ident": "postcss",
"plugins": postcssPlugins
}
},
{
"loader": "sass-loader",
"options": {
"sourceMap": false,
"precision": 8,
"includePaths": []
}
}
]
},
{
"include": [
path.join(process.cwd(), "src\\themes\\default-
theme\\ej.dashboardViewer.all.min.css")
],
"test": /\.less$/,
"use": [
"style-loader",
{
"loader": "css-loader",
"options": {
"sourceMap": false,
"importLoaders": 1
}
},
{
"loader": "postcss-loader",
"options": {
"ident": "postcss",
"plugins": postcssPlugins
}
},
{
```



```
"loader": "less-loader",
"options": {
"sourceMap": false,
"paths": []
}
}
]
},
{
"include": [
path.join(process.cwd(), "src\\themes\\default-
theme\\ej.dashboardViewer.all.min.css")
],
"test": /\.styl$/,
"use": [
"style-loader",
{
"loader": "css-loader",
"options": {
"sourceMap": false,
"importLoaders": 1
}
},
{
"loader": "postcss-loader",
"options": {
"ident": "postcss",
"plugins": postcssPlugins
}
},
{
"loader": "stylus-loader",
"options": {
"sourceMap": false,
"paths": []
}
}
]
},
{
"test": /\.ts$/,
"loader": "@ngtools/webpack"
}
]
},
"plugins": [
new NoEmitOnErrorsPlugin(),
new ConcatPlugin({
"uglify": false,
"sourceMap": true,
"name": "scripts",
"fileName": "[name].bundle.js",
"filesToConcat": [
"**\\node_modules\\jquery\\dist\\jquery.js",
"**\\node_modules\\bootstrap\\dist\\js\\bootstrap.js"
]
}),
```



```
new InsertConcatAssetsWebpackPlugin([
"scripts"
]),
new CopyWebpackPlugin([
{
"context": "**\\src/",
"to": "",
"from": {
"glob": "assets/**/*",
"dot": true
}
},
{
"context": "**\\src/",
"to": "",
"from": {
"glob": "favicon.ico",
"dot": true
}
}
], {
"ignore": [
".gitkeep"
],
"debug": "warning"
}),
new ProgressPlugin(),
new CircularDependencyPlugin({
"exclude": /(\\|\/)node_modules(\\|\/)/,
"failOnError": false
}),
new NamedLazyChunksWebpackPlugin(),
new HtmlWebpackPlugin({
"template": "./src\\index.html",
"filename": "./index.html",
"hash": false,
"inject": true,
"compile": true,
"favicon": false,
"minify": false,
"cache": true,
"showErrors": true,
"chunks": "all",
"excludeChunks": [],
"title": "Webpack App",
"xhtml": true,
"chunksSortMode": function sort(left, right) {
let leftIndex = entryPoints.indexOf(left.names[0]);
let rightindex = entryPoints.indexOf(right.names[0]);
if (leftIndex > rightindex) {
return 1;
}
else if (leftIndex < rightindex) {
return -1;
}
else {
return 0;
```

```
}
}
}),
new BaseHrefWebpackPlugin({}),
new CommonsChunkPlugin({
"name": [
"inline"
],
"minChunks": null
}),
new CommonsChunkPlugin({
"name": [
"vendor"
],
"minChunks": (module) => {
return module.resource
&& (module.resource.startsWith(nodeModules)
|| module.resource.startsWith(genDirNodeModules)
|| module.resource.startsWith(realNodeModules));
},
"chunks": [
"main"
]
}),
new SourceMapDevToolPlugin({
"filename": "[file].map[query]",
"moduleFilenameTemplate": "[resource-path]",
"fallbackModuleFilenameTemplate": "[resource-path]?[hash]",
"sourceRoot": "webpack:///"
}),
new CommonsChunkPlugin({
"name": [
"main"
],
"minChunks": 2,
"async": "common"
}),
new NamedModulesPlugin({}),
new AotPlugin({
"mainPath": "main.ts",
"replaceExport": false,
"hostReplacementPaths": {
"environments\\environment.ts": "environments\\environment.ts"
},
"exclude": [],
"tsConfigPath": "src\\tsconfig.app.json",
"skipCodeGeneration": true
})
],
"node": {
"fs": "empty",
"global": true,
"crypto": "empty",
"tls": "empty",
"net": "empty",
"process": true,
"module": false,
```

```
"clearImmediate": false,
"setImmediate": false
},
"devServer": {
"historyApiFallback": true
}
};
```

TS

```
//in polyfills.ts
import 'core-js/es6/reflect';
import 'core-js/es7/reflect';
import 'zone.js/dist/zone';
```

TS

```
//In main.ts
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { AppModule } from './app/app.module';
import { environment } from './environments/environment';
if (environment.production) {
enableProdMode();
}
platformBrowserDynamic().bootstrapModule(AppModule);
```

JSON

```
// In tsconfig.json
{
"compilerOptions": {
"sourceMap": true,
"declaration": false,
"moduleResolution": "node",
"emitDecoratorMetadata": true,
"experimentalDecorators": true,
"target": "es5",
"lib": [
"es2016",
"dom"
],
"outDir": "../out-tsc/app",
"module": "es2015",
"baseUrl": "",
"typeRoots": [
"node_modules/@types/"
],
"types": [
"jquery",
"ej.web.all"
]
},
"exclude": [
"**/*.spec.ts"
```



```
]
}
```

Running the application

- To run the application, execute the following command.

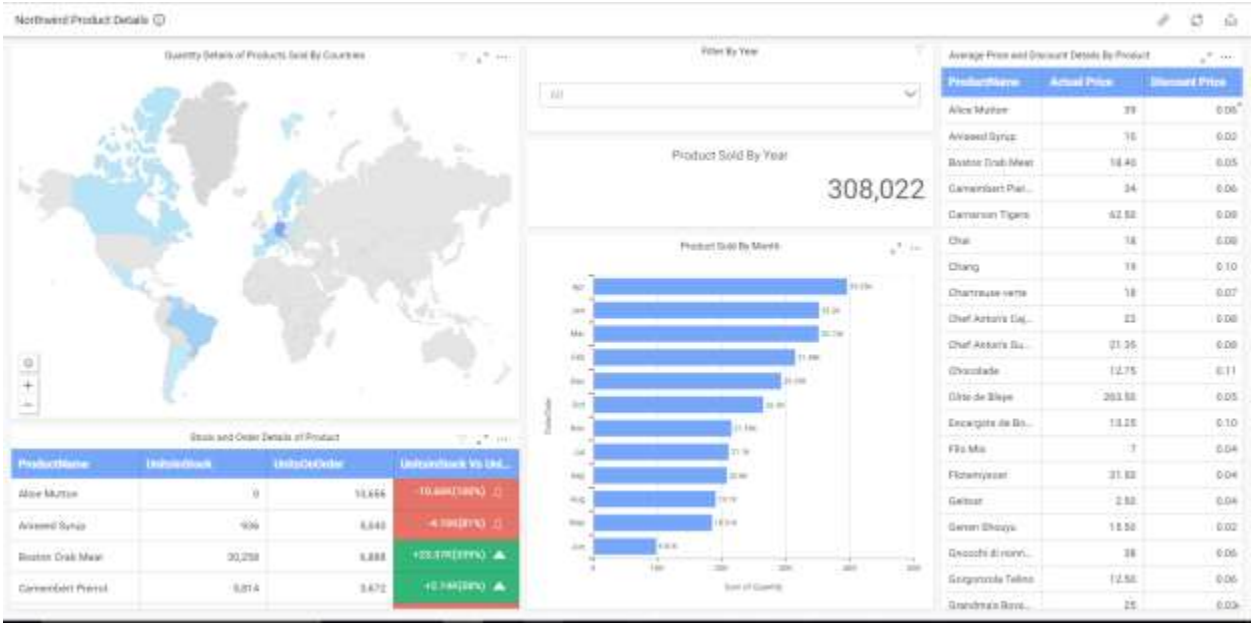
JAVASCRIPT

```
npm install
npm start
```

Note: While running the angular sample for the first time, npm install command is required to install the required packages in the angular followed by npm start to launch the sample.

- Browse to http://localhost:4200, to see the application. The component is rendered as shown in the below screenshot. You can make changes in the code found under the src folder and the browser should auto-refresh itself while saving the files.

*If you face any issue, delete the node_modules folder and again run npm install. After completing the installation process, run the npm start --port <“port”> command, for example: npm start --port 6200.

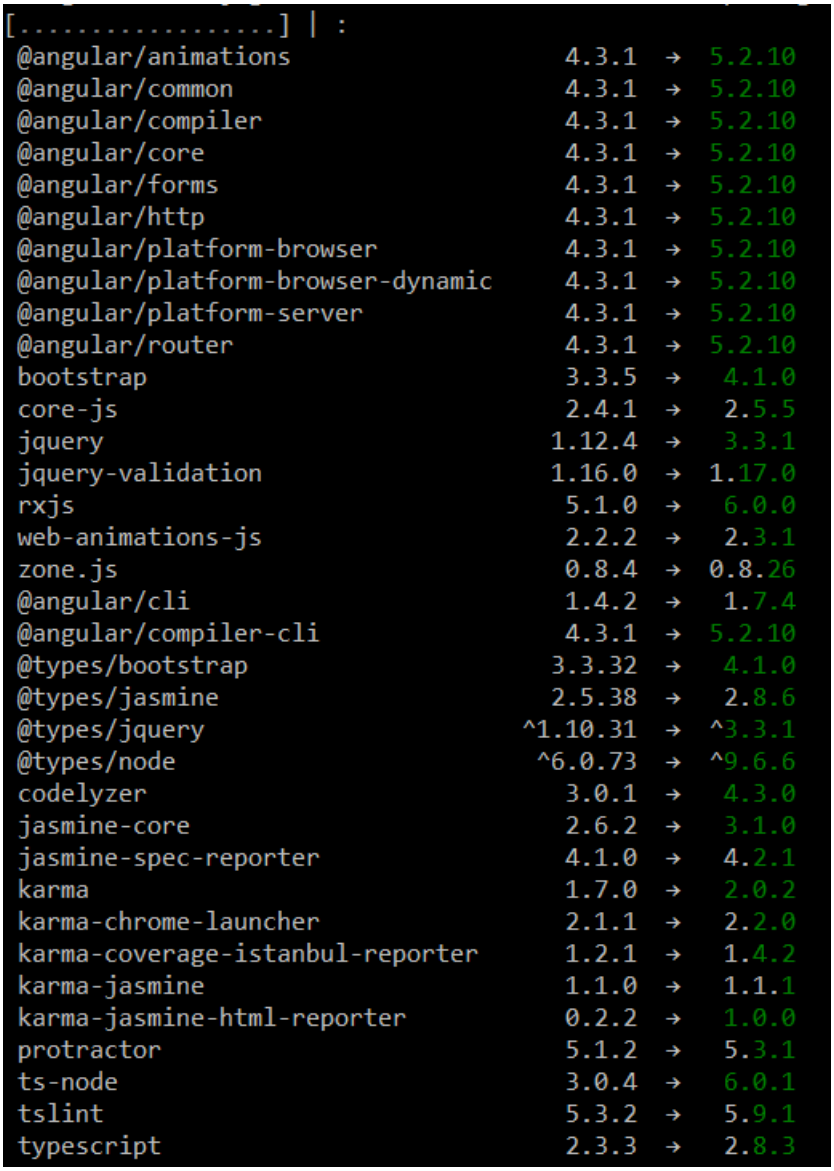


Upgrade Angular 4 app to Latest Version

Update the Angular packages, using the following commands.

JAVASCRIPT

```
npm install -g npm-check-updates
ncu -u
```



If you get any error while running these commands, update your node and npm version. Follow the instructions given in this link to upgrade the node and npm.

Finally, delete the previous `node_modules` folder and run `npm install` command.

Now, your sample is updated to the Latest Angular version.

Binding dashboard service

To initiate the dashboard service instance, follow anyone of the following methods.

1. Hosting dashboard service in IIS.
2. Hosting dashboard service in IIS Express.
3. Hosting dashboard service as Windows service background process.

Information: Hosting dashboard service at IIS is recommended to production environment for object management and other memory management features.

The following properties of ejDashboardViewer widget support model binding:

serviceUrl dashboardPath

ASP.NET Web Forms

Getting Started with ASP.NET Web Forms Application

This section describes how to create an ASP.NET Web Forms application with embedded dashboard viewer.

Project Creation

Create a new ASP.NET Web application project using Microsoft Visual Studio IDE 2012 or higher. Refer here for more details.

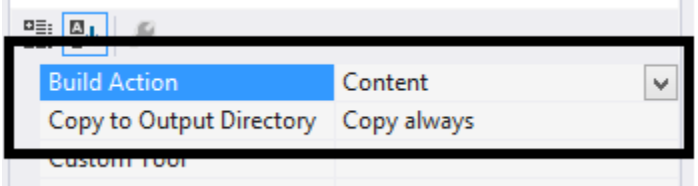
Adding files and references

Add the scripts, styles and refer fonts that are required for the dashboard from the following location to the application project.

`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html`

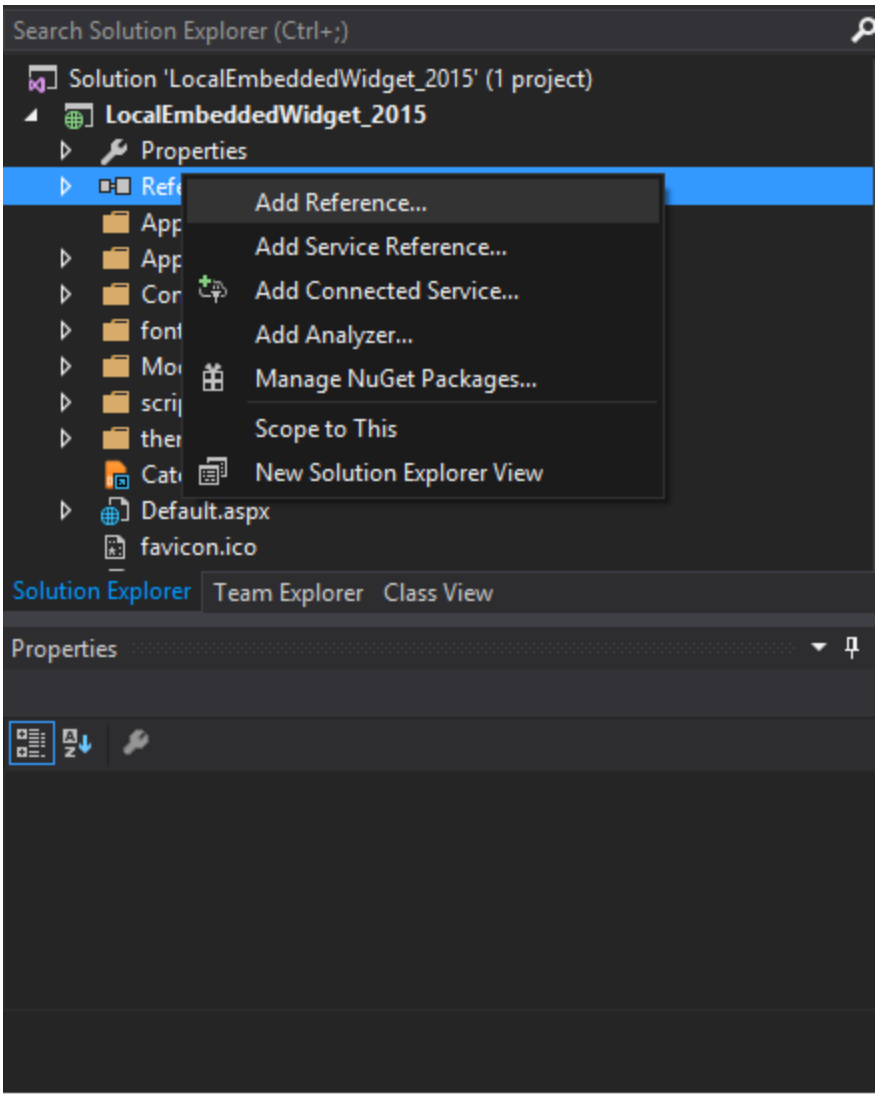
Include the dashboard file (*.sydx) into the project.

Set the `Build Action` property to `Content` and the `Copy to Output Directory` property to `Copy always` as shown in the following image for all the files added to the project.



Adding Dashboard Viewer Assembly References

Right-click the project and add the following assembly references through choosing `Add > Reference...` as shown below,

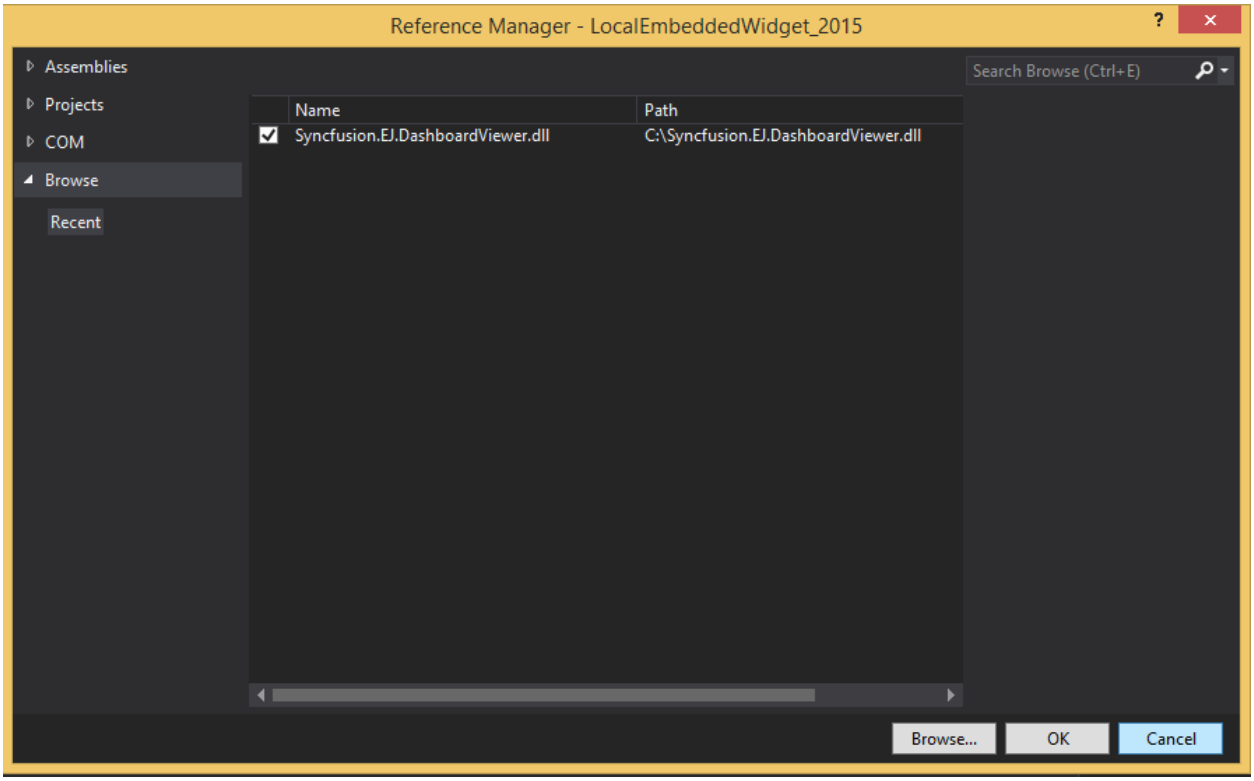


selecting the below mentioned assembly in dialog shown, which allows you to use any of the Syncfusion Dashboard components within it.

- Syncfusion.EJ.DashboardViewer

The above mentioned assembly files can be found in the following Dashboard SDK samples location:

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Precompiled Assemblies



Registering Syncfusion Assemblies within the Web.config

In your application's web.config file, add the below assembly information within the tag.

HTML

```
<system.web>
<compilation debug="true" targetFramework="4.5" >
<assemblies>
<add assembly="Syncfusion.Dashboard.EJ"/>
<add assembly="Syncfusion.Dashboard.EJ.Web"/>
<add assembly="Syncfusion.EJ.DashboardViewer"/>
</assemblies>
</compilation>
</system.web>
```

Adding Dashboard Viewer Resources

Select any one of the way to add the resources for Dashboard Viewer:

- External Resources
- Embedded Resources

External Resources:

Add the below script references in the Site.Master file.

ASPX-CS

```
<%@ Master Language="C#" %>
<!DOCTYPE html>
<html lang="en">
<head id="Head1" runat="server">
```



```
<meta charset="utf-8" />
<title>Getting Started</title>
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet" />
<script src='<%= Page.ResolveClientUrl("scripts/jquery-1.10.2.min.js")%>'
type="text/javascript"></script>
<script src='<%=
Page.ResolveClientUrl("scripts/jquery.easing.1.3.min.js")%>'
type="text/javascript"></script>
<script src='<%=
Page.ResolveClientUrl("scripts/ej.dashboardViewer.all.min.js")%>'
type="text/javascript"></script>
<script src='<%=
Page.ResolveClientUrl("scripts/ej.dashboard.webform.min.js")%>'
type="text/javascript"></script>
</head>
<body style="height:100%;width:100%;">
<form id="Form1" runat="server" style="height:100%;width:100%;">
<asp:ScriptManager runat="server" ID="scriptmgr"  ></asp:ScriptManager>
<div id="body" style="height:100%;width:100%;">
<asp:ContentPlaceHolder runat="server" ID="MainContent" />
</div>
</form>
</body>
</html>
```

Embedded Resources:

Dashboard Viewer will load the resources from its assembly. Refer Embedded Resources for more detail.

Control Initialization

To create the Dashboard Viewer instance, use the below code snippet inside Default.aspx page.

ASPX-CS

```
<%@ Register Assembly="Syncfusion.EJ.DashboardViewer"
Namespace="Syncfusion.Dashboard.JavaScript.Web" TagPrefix="ej" %>
<asp:Content runat="server" ID="BodyContent"
ContentPlaceHolderID="MainContent">
<ej:DashboardViewer ID="dashboard" runat="server" Size-Height="100%" Size-
Width="100%">
</ej:DashboardViewer>
</asp:Content>
```

Binding Dashboard Service and Dashboard File

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Create a global application class file named Global.asax.

Define the properties with public access level for handling dashboard path and service URL through the following code definitions under Global class in Global.asax file.

C#

```
public static string DashboardPath;
public static string ServiceUrl;
```

VB.NET

```
Public Shared DashboardPath As String
Public Shared ServiceUrl As String
```

Set the dashboard path and service URL using the following code under Application_Start method in Global.asax file.

C#

```
DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\\", "\\\\")
+ "bin\\\\WorldWideCarSalesDashboard.sydx"; // Or use the remote (online)
Dashboard Path. For example,
https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDashboard.
sydx
ServiceUrl = "http://localhost:3002/DashboardService.svc"; // Or use the
remote (online) Dashboard Service. For example,
https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc
```

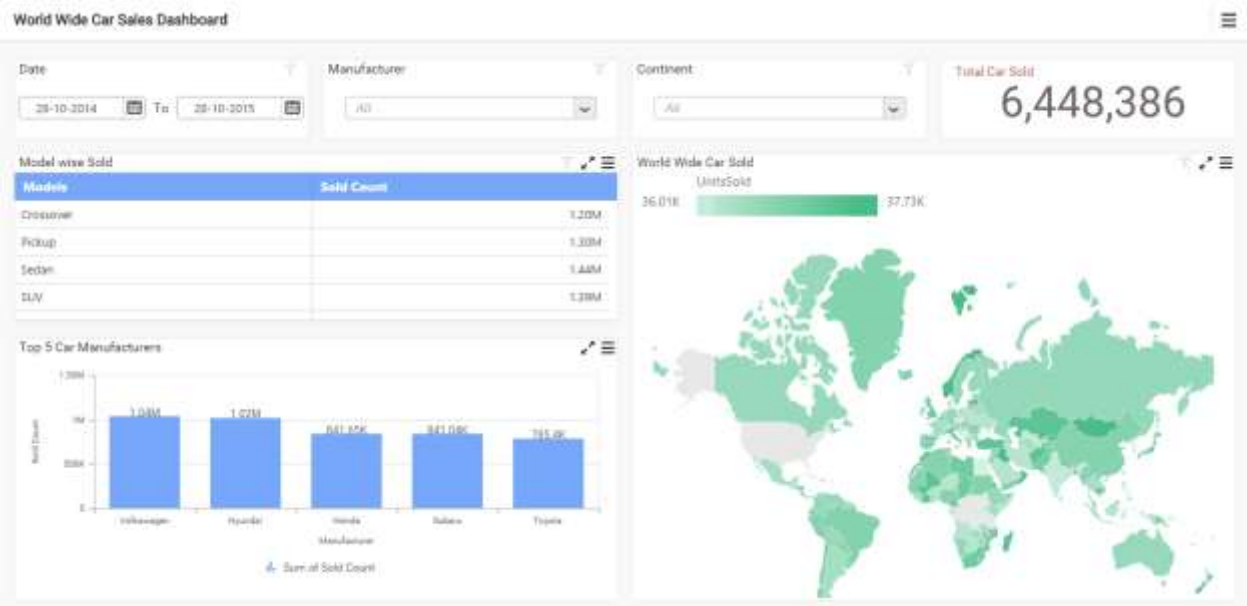
VB.NET

```
DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\", "\\") &
"bin\\WorldWideCarSalesDashboard.sydx" '// Or use the remote (online)
Dashboard Path. For example,
https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDashboard.
sydx
ServiceUrl = "http://localhost:3002/DashboardService.svc" '// Or use the
remote (online) Dashboard Service. For example,
https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc
```

Note: Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

Note: Provided online Dashboard Service URL above, only for demo purpose.

Build and run the application to view the dashboard.



How to Use the Windows Service as Dashboard Service

To use the window service as a background process refer,

[Run Windows Dashboard Service as a background process]

Refer the below code snippet to configure the Dashboard Viewer with Windows Dashboard Service:

Create a class named `DashboardWindowsServiceInfo` and add the below code within the class.

C#

```
using System;
using System.IO;
using System.Net;
using Microsoft.Win32;
public class DashboardWindowsServiceInfo
{
private readonly string _environmentFolder =
AppDomain.CurrentDomain.BaseDirectory;
string version =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
();
public string ServiceUrl;
public string ErrorMessage;
public DashboardWindowsServiceInfo()
{
#region Pick Windows Dashboard Service Url
ServiceUrl = GetWindowsServiceUrl();
#endregion
#region Pick IISExpress or IIS  Dashboard Service Url if Windows Dashboard
Service is not running
if (ValidateDashboardService(ServiceUrl))
{
DashboardServiceSerialization serializer = new
DashboardServiceSerialization();
DashboardServicePreviewSettings settings = new
DashboardServicePreviewSettings();
```

```
string dashboardServiceSettingPath =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) +
@"\Syncfusion\Dashboard Platform SDK\" + version +
@"\DashboardServiceSetting.xml";
if (File.Exists(dashboardServiceSettingPath))
{
settings = serializer.Deserialize(dashboardServiceSettingPath);
if (!ValidateDashboardService(settings.ServiceURL))
ServiceUrl = settings.ServiceURL;
else
{
ServiceUrl = string.Empty;
ErrorMessage; = "Dashboard Service is not running. Run
DashboardServiceInstaller.exe file to start Dashboard Service in IIS
Express";
}
}
else
{
ErrorMessage; = "Dashboard Service is not running. Run
DashboardServiceInstaller.exe file to start Dashboard Service in IIS
Express";
ServiceUrl = string.Empty;
}
}
#endregion
}
/// <summary>
/// Used to pick the Windows Dashboard Service URL
/// </summary>
/// <returns>ServiceURL of Windows Dashboard Service</returns>
private string GetWindowsServiceUrl()
{
string url = string.Empty;
try
{
RegistryKey key =
Registry.LocalMachine.OpenSubKey(@"Software\SyncfusionDashboard\Syncfusion
Dashboard Service");
if (key == null)
key =
Registry.LocalMachine.OpenSubKey(@"Software\Wow6432Node\SyncfusionDashboard\
Syncfusion Dashboard Service");
if (key != null)
{
url = (string)key.GetValue("ServiceURL");
key.Close();
}
}
catch (Exception)
{
}
return url;
}
/// <summary>
/// Validate whether Dashboard Service is running in the Url
/// </summary>
```

```
/// <param name="dashboardServiceUrl">Dashboard Service Url</param>
/// <returns>returns whether valid dashboard service</returns>
private static bool ValidateDashboardService(string dashboardServiceUrl)
{
bool errorOccurred = false;
try
{
if (string.IsNullOrWhiteSpace(dashboardServiceUrl))
{
return true;
}
if (!dashboardServiceUrl.Contains("http://") &&
!dashboardServiceUrl.Contains("https://"))
dashboardServiceUrl = "http://" + dashboardServiceUrl + @"/IsServiceExists";
else
dashboardServiceUrl = dashboardServiceUrl + @"/IsServiceExists";
WebRequest request = WebRequest.Create(new Uri(dashboardServiceUrl,
UriKind.Absolute));
request.Method = "GET";
using (WebResponse response = request.GetResponse())
{
using (StreamReader reader = new StreamReader(response.GetResponseStream()))
{
string text = reader.ReadToEnd();
if
(!text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetB
ytes("DashboardServiceExists"))))
{
errorOccurred = true;
}
}
}
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
}
catch (Exception e)
{
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
errorOccurred = true;
}
return errorOccurred;
}
}
```

VB.NET

```
Imports System
Imports System.IO
Imports System.Net
Imports Microsoft.Win32
Public Class DashboardWindowsServiceInfo
Private ReadOnly _environmentFolder As String =
AppDomain.CurrentDomain.BaseDirectory
Private version As String =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
()
Public ServiceUrl As String
```

```
Public ErrorMessage; As String
Public Sub New()
'Pick Dashboard Windows Service URL
ServiceUrl = GetWindowsServiceUrl()
'Pick IISExpress or IIS  Dashboard Service Url if Windows Dashboard Service
is not running
If ValidateDashboardService(ServiceUrl) Then
Dim serializer As New DashboardServiceSerialization()
Dim settings As New DashboardServicePreviewSettings()
Dim dashboardServiceSettingPath As String =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) &
"\Syncfusion\Dashboard Platform SDK\" & version &
"\DashboardServiceSetting.xml"
If File.Exists(dashboardServiceSettingPath) Then
settings = serializer.Deserialize(dashboardServiceSettingPath)
If Not ValidateDashboardService(settings.ServiceURL) Then
ServiceUrl = settings.ServiceURL
Else
ServiceUrl = String.Empty
ErrorMessage; = "Dashboard Service is not running. Run
DashboardServiceInstaller.exe file to start Dashboard Service in IIS
Express"
End If
Else
ErrorMessage; = "Dashboard Service is not running. Run
DashboardServiceInstaller.exe file to start Dashboard Service in IIS
Express"
ServiceUrl = String.Empty
End If
End If
End Sub
''' <summary>
''' Used to pick the Windows Dashboard Service URL
''' </summary>
''' <returns>ServiceURL of Windows Dashboard Service</returns>
Private Function GetWindowsServiceUrl() As String
Dim url As String = String.Empty
Try
Dim key As RegistryKey =
Registry.LocalMachine.OpenSubKey("Software\SyncfusionDashboard\Syncfusion
Dashboard Service")
If key Is Nothing Then
key =
Registry.LocalMachine.OpenSubKey("Software\Wow6432Node\SyncfusionDashboard\S
yncfusion Dashboard Service")
End If
If key IsNot Nothing Then
url = CStr(key.GetValue("ServiceURL"))
key.Close()
End If
Catch e1 As Exception
End Try
Return url
End Function
''' <summary>
''' Validate whether Dashboard Service is running in the Url
''' </summary>
```

```
''' <param name="dashboardServiceUrl">Dashboard Service Url</param>
''' <returns>returns whether valid dashboard service or not.</returns>
Private Shared Function ValidateDashboardService(ByVal dashboardServiceUrl
As String) As Boolean
Dim errorOccurred As Boolean = False
Try
If String.IsNullOrWhiteSpace(dashboardServiceUrl) Then
Return True
End If
If Not dashboardServiceUrl.Contains("http://") AndAlso Not
dashboardServiceUrl.Contains("https://") Then
dashboardServiceUrl = "http://" & dashboardServiceUrl & "/IsServiceExists"
Else
dashboardServiceUrl = dashboardServiceUrl & "/IsServiceExists"
End If
Dim request As WebRequest = WebRequest.Create(New Uri(dashboardServiceUrl,
UriKind.Absolute))
request.Method = "GET"
Using response As WebResponse = request.GetResponse()
Using reader As New StreamReader(response.GetResponseStream())
Dim text As String = reader.ReadToEnd()
If Not
text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetByt
es("DashboardServiceExists"))) Then
errorOccurred = True
End If
End Using
End Using
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
Catch e As Exception
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
errorOccurred = True
End Try
Return errorOccurred
End Function
End Class
```

Add a class named DashboardServicePreviewSettings and add the below code within the class.

C#

```
using System;
using System.Collections.Generic;
public class DashboardServicePreviewSettings
{
public string ServiceURL { get; set; }
public List<Guid> DashboardServiceInstances { get; set; }
public DashboardServicePreviewSettings()
{
DashboardServiceInstances = new List<Guid>();
}
}
```

VB.NET

```
Imports System
```



```
Imports System.Collections.Generic
Public Class DashboardServicePreviewSettings
Public Property ServiceURL() As String
Public Property DashboardServiceInstances() As List(Of Guid)
Public Sub New()
DashboardServiceInstances = New List(Of Guid)()
End Sub
End Class
```

Create a class named DashboardServiceSerialization and add the below code within the class to serialize and deserialize the DashboardService URL when Dashboard Service is running in IIS Express.

C#

```
using System;
using System.IO;
using System.Xml.Serialization;
public class DashboardServiceSerialization
{
static readonly XmlSerializer previewSerializer = new
XmlSerializer(typeof(DashboardServicePreviewSettings));
public void Serialize(DashboardServicePreviewSettings settings, string path)
{
try
{
using (StreamWriter writer = new StreamWriter(path))
{
previewSerializer.Serialize(writer, settings);
}
}
catch (Exception)
{
}
}
public DashboardServicePreviewSettings Deserialize(string path)
{
DashboardServicePreviewSettings settings = new
DashboardServicePreviewSettings();
try
{
using (StreamReader reader = new StreamReader(path))
{
settings =
(DashboardServicePreviewSettings)previewSerializer.Deserialize(reader);
}
}
catch (Exception)
{
}
return settings;
}
}
```

VB.NET

```
Imports System
```

```
Imports System.IO
Imports System.Xml.Serialization
Public Class DashboardServiceSerialization
Private Shared ReadOnly previewSerializer As New
XmlSerializer(GetType(DashboardServicePreviewSettings))
Public Sub Serialize(ByVal settings As DashboardServicePreviewSettings,
ByVal path As String)
Try
Using writer As New StreamWriter(path)
previewSerializer.Serialize(writer, settings)
End Using
Catch e1 As Exception
End Try
End Sub
Public Function Deserialize(ByVal path As String) As
DashboardServicePreviewSettings
Dim settings As New DashboardServicePreviewSettings()
Try
Using reader As New StreamReader(path)
settings = CType(previewSerializer.Deserialize(reader),
DashboardServicePreviewSettings)
End Using
Catch e1 As Exception
End Try
Return settings
End Function
End Class
```

Embedded Resources

we registered our scripts and themes manually in application level to render our Dashboard Viewer component. Now you can also embed the resources from assembly to reduce the work. Hence forth user can render Dashboard platform SDK Dashboard Viewer component without any manual configuration.

How it works?

The Dashboard platform SDK Dashboard Viewer resources and dependent scripts are encapsulated as embedded in our assembly and registered those resources in ASP Script Manager. Afterwards ASP Script Manager will take appropriate process to access the resources from assembly as usual once components loaded on that page.

For themes, necessary stylesheets will be added in header section of current page dynamically to render theDashboard platform SDK Dashboard Viewer component. To get the embed resources in your application, you should register an App Key in Web.Config file.

Access Embedded Resource

The following key settings will configure in web.config file. Please refer below code snippet:

XML

```
<appSettings>
<add key="LoadDashboardResourcesFromAssembly" value="true" />
<add key="DashboardResources" value="jqueryeasing:true;themes:true;" />
</appSettings>
```


Where 'LoadDashboardResourcesFromAssembly' key denotes that whether resources are referred from assembly or not and 'DashboardResources' key used to get scripts and themes from assembly.

Note: The above key setting we should configure it manually in web.config file.

What are the resources will embed from assembly?

By default, the following resources shipped as embedded resources from assembly.

jQuery Easing v1.3 Default Theme

And component related scripts will embed dynamically from assembly in your application.

CDN integration with Embedded Resources

You can get these embedded resources from CDN (Content Delivery Networks) also. To achieve this behavior you should enable the **EnableCdn** property in ASP script manager control. Please refer below code snippet:

ASPX-CS

```
<asp:scriptmanager id="ScriptManager1" runat="server" EnableCdn="true">
</asp:scriptmanager>
```

You can embed the resources from assembly when CDN (Content Delivery Network) is unavailable. To achieve this behavior you should enable **EnableCdnFallback** property in script manager. Please refer the below code snippet:

ASPX-CS

```
<asp:ScriptManager runat="server" EnableCdn="true" EnableCdnFallback="true">
</asp:scriptmanager>
```

Note: 1. EnableCdnFallback property is supported from 4.5 and above frameworks.

2. EnableCdnFallback is not applicable for theme file.

3. Both script and style resources can also be accessed through HTTPS (secure connection) from CDN

Getting Started with ASP.NET MVC Application

This section describes how to create an ASP.NET MVC application with an embedded dashboard viewer.

Project Creation

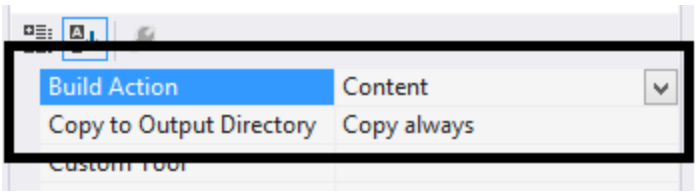
Create a new empty ASP.NET MVC project using Microsoft Visual Studio IDE 2012 or higher. Refer here for more details.

Adding files and references

Add the scripts, styles and refer fonts that are required for the dashboard from the following location to the application project.

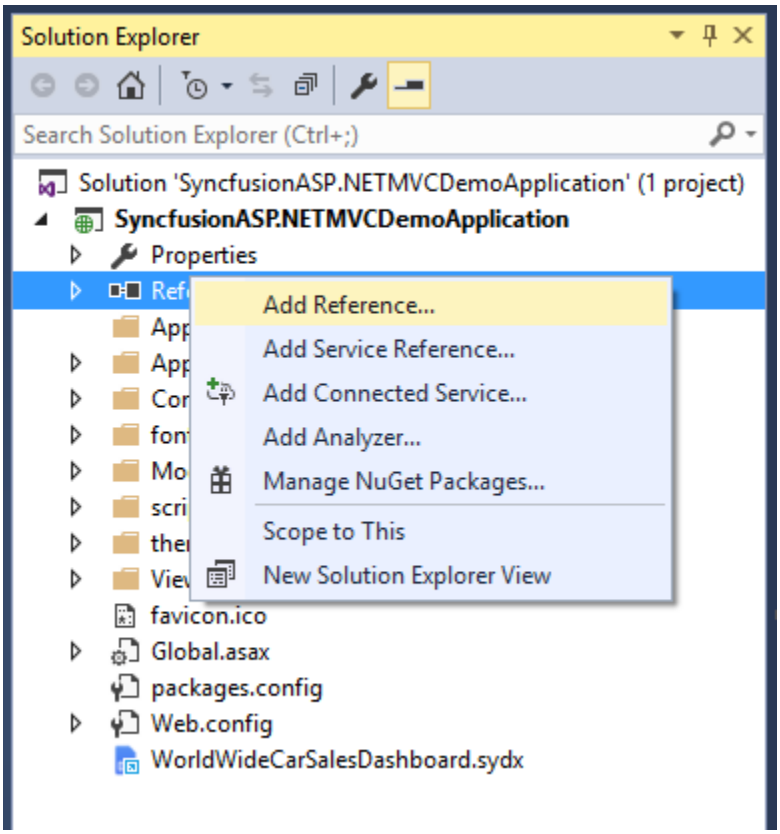
`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html`

Include the dashboard file (*.sydx) in the project and set the `Build Action` and `Copy to Output Directory` properties to `Content` and `Copy always`, respectively.



Adding Dashboard Viewer Assembly References

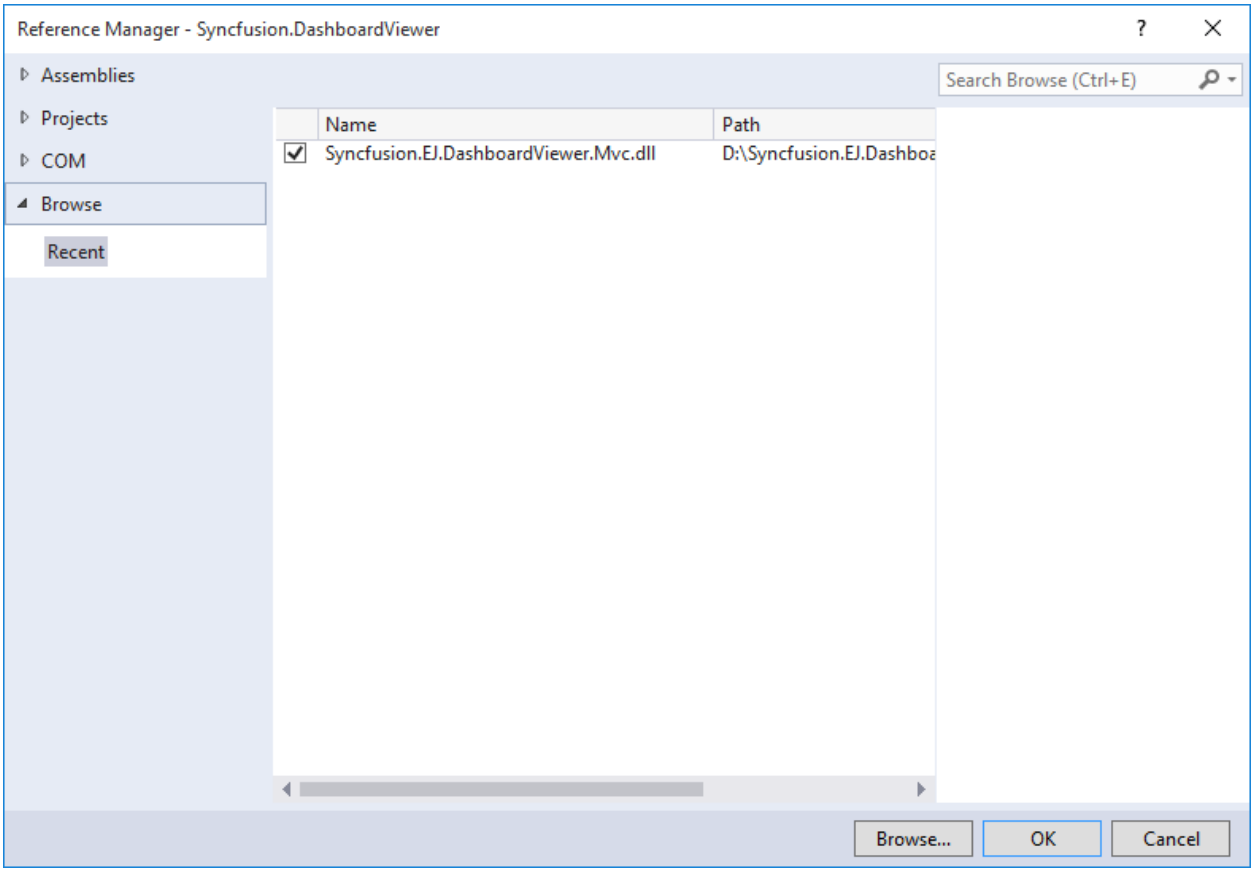
Right-click the project and add the following assembly references through choosing Add > Reference... as shown below,



select the "Syncfusion.EJ.DashboardViewer.Mvc" assembly in the dialog shown, which allows you to use any of the Syncfusion Dashboard components within it.

The above mentioned assembly files can be found in the following Dashboard SDK samples location:

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Precompiled Assemblies



Binding Dashboard Service

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process
4. Hosting Dashboard Service as Dll

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Creating Model, View and Controller

Create a model class DashboardViewerController and add the below code within the class.

C#

```
using System;
using System.Web.Mvc;
public class DashboardViewerController : Controller
{
public ActionResult Index()
{
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\\",
"\\\\") + "bin\\\\WorldWideCarSalesDashboard.sydx"; // Or use the remote
(online) Dashboard Path. For example,
```



```
https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDashboard.sydx
ViewBag.ServiceUrl = "http://localhost:3002/DashboardService.svc"; // Or use the remote (online) Dashboard Service. For example, https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc;
return View();
}
}
```

Imports System

Imports System.Web.Mvc

Public Class DashboardViewerController

Inherits Controller

Public Function Index() As ActionResult

ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\", "\\") & "bin\\WorldWideCarSalesDashboard.sydx" '// Or use the remote (online) Dashboard Path. For example, https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDashboard.sydx

ViewBag.ServiceUrl = "http://localhost:3002/DashboardService.svc"; '// Or use the remote (online) Dashboard Service. For example, https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc;

Return View()

End Function

End Class

{% endhighlight %}

Note: Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

Provided online Dashboard Service URL above, only for demo purpose.

Add a Razor view (Index.cshtml file) with the following template included.

ASPX-CS

```
<!DOCTYPE html>
<html>
<head>
<title>Getting Started</title>
<link href="@Url.Content("~/themes/default-theme/ej.dashboardViewer.all.min.css")" rel="stylesheet">
<script src="@Url.Content("~/scripts/jquery-1.10.2.min.js")"></script>
<script src="@Url.Content("~/scripts/jquery.easing.1.3.min.js")"></script>
<script src="@Url.Content("~/scripts/ej.dashboardViewer.all.min.js")"></script>
<script src="@Url.Content("~/scripts/ej.unobtrusive.min.js")"></script>
</head>
<body>
</body>
</html>
```


Configuring Web Settings

In your project's Web.config file, replace the below assembly information under the system.web section.

ASPX-CS

```
<httpRuntime targetFramework="4.5" />
<compilation debug="true" targetFramework="4.5" >
<assemblies>
<add assembly="Syncfusion.EJ.DashboardViewer.Mvc"/>
</assemblies>
</compilation>
<pages>
<namespaces>
<add namespace="System.Web.Helpers" />
<add namespace="System.Web.Mvc" />
<add namespace="System.Web.Mvc.Ajax" />
<add namespace="System.Web.Mvc.Html" />
<add namespace="System.Web.Routing" />
<add namespace="System.Web.WebPages" />
</namespaces>
</pages>
```

Add the below settings under the appSettings section.

ASPX-CS

```
<add key="UnobtrusiveJavaScriptEnabled" value="true" />
```

In your project's web.config file which is under Views folder, add the below settings under the system.web.webPages.razor section.

ASPX-CS

```
<host factoryType="System.Web.Mvc.MvcWebRazorHostFactory, System.Web.Mvc,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
<pages pageBaseType="System.Web.Mvc.WebViewPage">
<namespaces>
...
<add namespace="Syncfusion.Dashboard.JavaScript"/>
<add namespace="Syncfusion.Dashboard.MVC.EJ"/>
<add namespace="Syncfusion.Dashboard.JavaScript.Shared"/>
</namespaces>
</pages>
```

Control Initialization

Add the below code under the body section of index.cshtml page to initialize the Dashboard Viewer control.

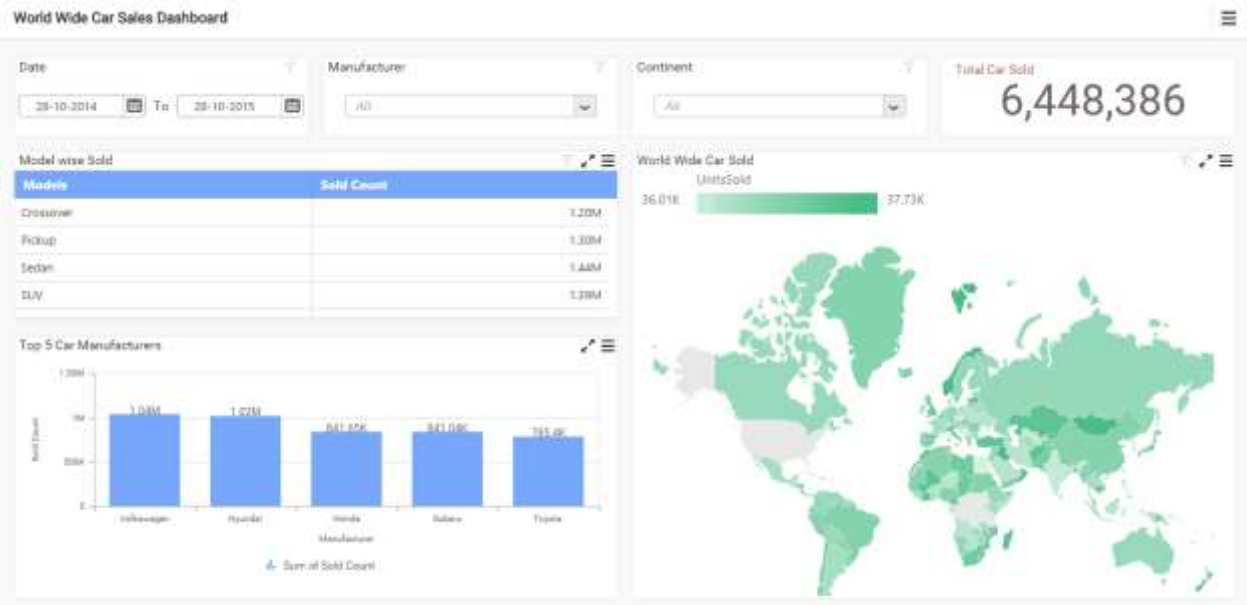
ASPX-CS

```
@Html.EJDashboard().DashboardViewer("dashboard").ServiceUrl(@ViewBag.Service
Url).DashboardPath(@ViewBag.DashboardPath)
```


Configuring Route

In the RegisterRoutes method in the RouteConfig class in the App_Start folder, set the controller name as DashboardViewer.

Build and run the application to view the dashboard.



How to Use the Windows Service as Dashboard Service

To use the window service as a background process refer,

[Run Windows Dashboard Service as a background process]

Refer the below code snippet to configure the Dashboard Viewer with Windows Dashboard Service:

Create a model class DashboardViewer and add the below code within the class.

C#

```
using System;
using System.IO;
using System.Net;
using Microsoft.Win32;
public class DashboardViewer
{
private readonly string _environmentFolder =
AppDomain.CurrentDomain.BaseDirectory;
string Version =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
();
public string ServiceUrl;
public string ErrorMessage;
public DashboardViewer()
{
#region Pick Dashboard Windows Service Url
ServiceUrl = GetWindowsServiceUrl();
#endregion
#region Pick IISExpress or IIS  Dashboard Service Url if Windows Dashboard
Service is not running
```

```
if (ValidateDashboardService(ServiceUrl))
{
DashboardServiceSerialization serializer = new
DashboardServiceSerialization();
DashboardServicePreviewSettings settings = new
DashboardServicePreviewSettings();
string dashboardServiceSettingPath =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) +
@"\Syncfusion\Dashboard Platform SDK\" + Version +
@"\DashboardServiceSetting.xml";
if (File.Exists(dashboardServiceSettingPath))
{
settings = serializer.Deserialize(dashboardServiceSettingPath);
if (!ValidateDashboardService(settings.ServiceURL))
ServiceUrl = settings.ServiceURL;
else
{
ServiceUrl = string.Empty;
ErrorMessage = "Dashboard Service is not running. Run
DashboardServiceInstaller.exe file to start Dashboard Service in IIS
Express";
}
}
else
{
ErrorMessage = "Dashboard Service is not running. Run
DashboardServiceInstaller.exe file to start Dashboard Service in IIS
Express";
ServiceUrl = string.Empty;
}
}
#endregion
}
/// <summary>
/// Validate whether Dashboard Service is running in the provided Url
/// </summary>
/// <param name="dashboardServiceUrl">Dashboard Service Url</param>
/// <returns>true, if valid; false otherwise</returns>
private static bool ValidateDashboardService(string dashboardServiceUrl)
{
bool errorOccurred = false;
try
{
if (string.IsNullOrWhiteSpace(dashboardServiceUrl))
{
return true;
}
if (!dashboardServiceUrl.Contains("http://") &&
!dashboardServiceUrl.Contains("https://"))
dashboardServiceUrl = "http://" + dashboardServiceUrl + @"/IsServiceExists";
else
dashboardServiceUrl = dashboardServiceUrl + @"/IsServiceExists";
WebRequest request = WebRequest.Create(new Uri(dashboardServiceUrl,
UriKind.Absolute));
request.Method = "GET";
using (WebResponse response = request.GetResponse())
{
```

```
using (StreamReader reader = new StreamReader(response.GetResponseStream()))
{
string text = reader.ReadToEnd();
if
(!text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetB
ytes("DashboardServiceExists"))))
{
errorOccurred = true;
}
}
}
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
}
catch (Exception e)
{
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
errorOccurred = true;
}
return errorOccurred;
}
/// <summary>
/// Gets the Dashboard Windows Service URL
/// </summary>
/// <returns>Service URL of Dashboard Windows Service</returns>
private string GetWindowsServiceUrl()
{
string url = string.Empty;
try
{
RegistryKey key =
Registry.LocalMachine.OpenSubKey(@"Software\SyncfusionDashboard\Syncfusion
Dashboard Service");
if (key == null)
key =
Registry.LocalMachine.OpenSubKey(@"Software\Wow6432Node\SyncfusionDashboard\
Syncfusion Dashboard Service");
if (key != null)
{
url = (string)key.GetValue("ServiceURL");
key.Close();
}
}
catch (Exception)
{
}
return url;
}
}
```

VB.NET

```
Imports System
Imports System.IO
Imports System.Net
Imports Microsoft.Win32
Public Class DashboardViewer
```

```
Private ReadOnly _environmentFolder As String =
AppDomain.CurrentDomain.BaseDirectory
Private Version As String =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
()
Public ServiceUrl As String
Public ErrorMessage As String
Public Sub New()
'Pick Dashboard Windows Service Url
ServiceUrl = GetWindowsServiceUrl()
'Pick IISExpress or IIS  Dashboard Service Url if Windows Dashboard Service
is not running
If ValidateDashboardService(ServiceUrl) Then
Dim serializer As New DashboardServiceSerialization()
Dim settings As New DashboardServicePreviewSettings()
Dim dashboardServiceSettingPath As String =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) &
"\Syncfusion\Dashboard Platform SDK\" & Version &
"\DashboardServiceSetting.xml"
If File.Exists(dashboardServiceSettingPath) Then
settings = serializer.Deserialize(dashboardServiceSettingPath)
If Not ValidateDashboardService(settings.ServiceURL) Then
ServiceUrl = settings.ServiceURL
Else
ServiceUrl = String.Empty
ErrorMessage = "Dashboard Service is not running. Run
DashboardServiceInstaller.exe file to start Dashboard Service in IIS
Express"
End If
Else
ErrorMessage = "Dashboard Service is not running. Run
DashboardServiceInstaller.exe file to start Dashboard Service in IIS
Express"
ServiceUrl = String.Empty
End If
End If
End Sub
''' <summary>
''' Validate whether Dashboard Service is running in the Url
''' </summary>
''' <param name="dashboardServiceUrl">Dashboard Service Url</param>
''' <returns>true, if valid; false otherwise</returns>
Private Shared Function ValidateDashboardService(ByVal dashboardServiceUrl
As String) As Boolean
Dim errorOccurred As Boolean = False
Try
If String.IsNullOrWhiteSpace(dashboardServiceUrl) Then
Return True
End If
If Not dashboardServiceUrl.Contains("http://") AndAlso Not
dashboardServiceUrl.Contains("https://") Then
dashboardServiceUrl = "http://" & dashboardServiceUrl & "/IsServiceExists"
Else
dashboardServiceUrl = dashboardServiceUrl & "/IsServiceExists"
End If
Dim request As WebRequest = WebRequest.Create(New Uri(dashboardServiceUrl,
UriKind.Absolute))
```



```
request.Method = "GET"
Using response As WebResponse = request.GetResponse()
Using reader As New StreamReader(response.GetResponseStream())
Dim text As String = reader.ReadToEnd()
If Not
text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetByt
es("DashboardServiceExists"))) Then
errorOccurred = True
End If
End Using
End Using
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
Catch e As Exception
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
errorOccurred = True
End Try
Return errorOccurred
End Function
''' <summary>
''' Used to pick the Dashboard Windows Service URL
''' </summary>
''' <returns>Service URL of Dashboard Windows Service</returns>
Private Function GetWindowsServiceUrl() As String
Dim url As String = String.Empty
Try
Dim key As RegistryKey =
Registry.LocalMachine.OpenSubKey("Software\SyncfusionDashboard\Syncfusion
Dashboard Service")
If key Is Nothing Then
key =
Registry.LocalMachine.OpenSubKey("Software\Wow6432Node\SyncfusionDashboard\S
yncfusion Dashboard Service")
End If
If key IsNot Nothing Then
url = CStr(key.GetValue("ServiceURL"))
key.Close()
End If
Catch e1 As Exception
End Try
Return url
End Function
End Class
End Namespace
```

Add a class DashboardServicePreviewSettings and add the below code within the class.

C#

```
using System;
using System.Collections.Generic;
public class DashboardServicePreviewSettings
{
public string ServiceURL { get; set; }
public List<Guid> DashboardServiceInstances { get; set; }
public DashboardServicePreviewSettings()
{
```

```
DashboardServiceInstances = new List<Guid>();
}
}
```

VB.NET

```
Imports System
Imports System.Collections.Generic
Public Class DashboardServicePreviewSettings
Public Property ServiceURL() As String
Public Property DashboardServiceInstances() As List(Of Guid)
Public Sub New()
DashboardServiceInstances = New List(Of Guid)()
End Sub
End Class
```

Add a model class DashboardServiceSerialization and add the below code within the class to serialize and deserialize the DashboardService URL when Dashboard Service is running in IIS Express.

C#

```
using System;
using System.IO;
using System.Xml.Serialization;
public class DashboardServiceSerialization
{
static readonly XmlSerializer previewSerializer = new
XmlSerializer(typeof(DashboardServicePreviewSettings));
public void Serialize(DashboardServicePreviewSettings settings, string path)
{
try
{
using (StreamWriter writer = new StreamWriter(path))
{
previewSerializer.Serialize(writer, settings);
}
}
catch (Exception)
{
}
}
public DashboardServicePreviewSettings Deserialize(string path)
{
DashboardServicePreviewSettings settings = new
DashboardServicePreviewSettings();
try
{
using (StreamReader reader = new StreamReader(path))
{
settings =
(DashboardServicePreviewSettings)previewSerializer.Deserialize(reader);
}
}
catch (Exception)
{
}
```



```
return settings;
}
}
```

VB.NET

```
Imports System
Imports System.IO
Imports System.Xml.Serialization
Public Class DashboardServiceSerialization
Private Shared ReadOnly previewSerializer As New
XmlSerializer(GetType(DashboardServicePreviewSettings))
Public Sub Serialize(ByVal settings As DashboardServicePreviewSettings,
ByVal path As String)
Try
Using writer As New StreamWriter(path)
previewSerializer.Serialize(writer, settings)
End Using
Catch e1 As Exception
End Try
End Sub
Public Function Deserialize(ByVal path As String) As
DashboardServicePreviewSettings
Dim settings As New DashboardServicePreviewSettings()
Try
Using reader As New StreamReader(path)
settings = CType(previewSerializer.Deserialize(reader),
DashboardServicePreviewSettings)
End Using
Catch e1 As Exception
End Try
Return settings
End Function
End Class
```

How to embed dashboard service as dll

To host Syncfusion Dashboard Service as dll, refer the below steps to configure the Dashboard Viewer with Embedded Dashboard Service.

1. Add the following references from the Service -> bin folder (Syncfusion\Dashboard Platform SDK\Service) of Dashboard Platform SDK sample installed location to your application references.
    - Syncfusion.Dashboard.Base
    - Syncfusion.Dashboard.Compression.Base
    - Syncfusion.Dashboard.Encryption
    - Syncfusion.Dashboard.Json.Base
    - Syncfusion.Dashboard.Serialization
    - Syncfusion.Dashboard.Thrift
    - Syncfusion.Dashboard.ThriftHive.Base
    - Syncfusion.Dashboard.XlsIO.Base
    - Syncfusion.DashboardDesigner.Security
    - Syncfusion.DashboardService.Base

- Syncfusion.DashboardService
- Newtonsoft.json (10.0.3)
- Microsoft.Data.Edm
- Microsoft.Data.OData
- Microsoft.WindowsAzure.Storage
- Microsoft.WindowsAzure.StorageClient
- Microsoft.AnalysisServices.AdomdClient
- Microsoft.AnalysisServices.Core
- Microsoft.AnalysisServices

2. Create a new controller DashboardServiceController in your application and add the following code snippets to it.

C#

```
If your project sample was in MVC, use the below code :
using System.Net.Http;
using System.Web.Mvc;
using Syncfusion.DashboardService.Base;
public class DashboardServiceController : Controller
{
[HttpPost]
public ActionResult ProcessRequestForPost(DashboardServiceArguments
arguments)
{
DashboardServiceBase serviceBase = new DashboardServiceBase();
return Json(serviceBase.GetServiceResponse(arguments));
}
[HttpGet]
public HttpResponseMessage ProcessRequestForGet(string arguments)
{
DashboardServiceBase serviceBase = new DashboardServiceBase();
var stream = serviceBase.GetServiceResponse(arguments);
return serviceBase.GetResponseFromStream(stream);
}
[HttpGet]
public string IsServiceExists()
{
DashboardServiceBase serviceBase = new DashboardServiceBase();
return serviceBase.IsServiceExists();
}
[HttpGet]
public string Version()
{
DashboardServiceBase serviceBase = new DashboardServiceBase();
return serviceBase.Version();
}
}
If your project sample was in Web API, use the below code :
using System.Net.Http;
using System.Web.Mvc;
using Syncfusion.DashboardService.Base;
public class DashboardServiceController : ApiController
{
[Route("api/DashboardService/ProcessRequestForPost")]
[HttpPost]
```



```
public DashboardServiceBase.ServiceResponse
ProcessRequestForPost(DashboardServiceArguments arguments)
{
DashboardServiceBase serviceBase = new DashboardServiceBase();
return serviceBase.GetServiceResponse(arguments);
}
[Route("api/DashboardService/ProcessRequestForGet")]
[HttpGet]
public HttpResponseMessage ProcessRequestForGet(string arguments)
{
DashboardServiceBase serviceBase = new DashboardServiceBase();
var stream = serviceBase.GetServiceResponse(arguments);
return serviceBase.GetResponseFromStream(stream);
}
[Route("api/DashboardService/IsServiceExists")]
[HttpGet]
public string IsServiceExists()
{
DashboardServiceBase serviceBase = new DashboardServiceBase();
return serviceBase.IsServiceExists();
}
[Route("api/DashboardService/Version")]
[HttpGet]
public string Version()
{
DashboardServiceBase serviceBase = new DashboardServiceBase();
return serviceBase.Version();
}
}
```

VB.NET

```
If your project sample was in MVC, use the below code :
Imports Syncfusion.DashboardService.Base
Imports System.Net.Http
Namespace EmbeddingDashboardServiceAsDLL.Controllers
Public Class DashboardServiceController Inherits Controller
<HttpPost>
Public Function ProcessRequestForPost(ByVal arguments As
DashboardServiceArguments) As ActionResult
Dim serviceBase As New DashboardServiceBase()
Return Json(serviceBase.GetServiceResponse(arguments))
End Function
<HttpGet>
Public Function ProcessRequestForGet(ByVal arguments As String) As
HttpResponseMessage
Dim serviceBase As New DashboardServiceBase()
Dim stream = serviceBase.GetServiceResponse(arguments)
Return serviceBase.GetResponseFromStream(stream)
End Function
<HttpGet>
Public Function IsServiceExists() As String
Dim serviceBase As New DashboardServiceBase()
Return serviceBase.IsServiceExists()
End Function
<HttpGet>
```



```
Public Function Version() As String
Dim serviceBase As New DashboardServiceBase()
Return serviceBase.Version()
End Function
End Class
End Namespace
If your project sample was in Web API, use the below code :
Imports System.Net.Http
Imports System.Web.Mvc
Imports Syncfusion.DashboardService.Base
Namespace EmbeddingDashboardServiceAsDLL.Controllers
Public Class DashboardServiceController Inherits ApiController
<Route("api/DashboardService/ProcessRequestForPost")>
<HttpPost>
Public Function ProcessRequestForPost(ByVal arguments As
DashboardServiceArguments) As DashboardServiceBase.ServiceResponse
Dim serviceBase As DashboardServiceBase = New DashboardServiceBase()
Return serviceBase.GetServiceResponse(arguments)
End Function
<Route("api/DashboardService/ProcessRequestForGet")>
<HttpGet>
Public Function ProcessRequestForGet(ByVal arguments As String) As
HttpResponseMessage
Dim serviceBase As DashboardServiceBase = New DashboardServiceBase()
Dim stream = serviceBase.GetServiceResponse(arguments)
Return serviceBase.GetResponseFromStream(stream)
End Function
<Route("api/DashboardService/IsServiceExists")>
<HttpGet>
Public Function IsServiceExists() As String
Dim serviceBase As DashboardServiceBase = New DashboardServiceBase()
Return serviceBase.IsServiceExists()
End Function
<Route("api/DashboardService/Version")>
<HttpGet>
Public Function Version() As String
Dim serviceBase As DashboardServiceBase = New DashboardServiceBase()
Return serviceBase.Version()
End Function
End Class
End Namespace
```

3. Change the ServiceUrl defined in the 'DashboardViewerController' as in the below code snippet

C#

```
If your project sample was in MVC, use the below code :
public class DashboardViewerController : Controller
{
public ActionResult Index()
{
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\\",
"\\\\") + "bin\\\\Northwind Traders Sales Analysis.sydx";
##ViewBag.ServiceUrl = new UriBuilder(HttpContext.Request.Url.Scheme,
HttpContext.Request.Url.Host,
```



```
HttpContext.Request.Url.Port).ToString().TrimEnd('/') +
("/DashboardService");
return View();
}
public ActionResult Dashboard()
{
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\\",
"\\\\") + "bin\\\\Northwind Traders Sales Analysis.sydx";
ViewBag.ServiceUrl = new UriBuilder(HttpContext.Request.Url.Scheme,
HttpContext.Request.Url.Host,
HttpContext.Request.Url.Port).ToString().TrimEnd('/') +
("/DashboardService");
return View();
}
}
If your project sample was in Web API, use the below code :
public class DashboardViewerController : Controller
{
public ActionResult Index()
{
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\\",
"\\\\") + "bin\\\\Northwind Traders Sales Analysis.sydx";
ViewBag.ServiceUrl = Request.Url.AbsoluteUri + ("api/DashboardService");
return View();
}
public ActionResult Dashboard()
{
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\\",
"\\\\") + "bin\\\\Northwind Traders Sales Analysis.sydx";
ViewBag.ServiceUrl = Request.Url.AbsoluteUri + ("api/DashboardService");
return View();
}
}
```

VB.NET

```
If your project sample was in MVC, use the below code :
Public Class DashboardViewerController Inherits Controller
Public Function Index() As ActionResult
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\",
"\\") & "bin\\Northwind Traders Sales Analysis.sydx"
Return View()
End Function
Public Function Dashboard() As ActionResult
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\",
"\\") & "bin\\Northwind Traders Sales Analysis.sydx"
ViewBag.ServiceUrl = New UriBuilder(HttpContext.Request.Url.Scheme,
HttpContext.Request.Url.Host,
HttpContext.Request.Url.Port).ToString().TrimEnd("/"c) &
("/DashboardService")
Return View()
End Function
End Class
If your project sample was in Web API, use the below code :
Public Class DashboardViewerController Inherits Controller
Public Function Index() As ActionResult
```

```
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\",
"\\") & "bin\\Northwind Traders Sales Analysis.sydx"
ViewBag.ServiceUrl = Request.Url.AbsoluteUri & ("api/DashboardService")
Return View()
End Function
Public Function Dashboard() As ActionResult
ViewBag.DashboardPath = AppDomain.CurrentDomain.BaseDirectory.Replace("\",
"\\") & "bin\\Northwind Traders Sales Analysis.sydx"
ViewBag.ServiceUrl = Request.Url.AbsoluteUri & ("api/DashboardService")
Return View()
End Function
End Class
```

4. Add the highlighted code snippet in the Web.config of your sample, if its an MVC application. Add the System.Runtime assembly version according to the System.Object dll version of your MVC application framework.

```
<system.web>
  <httpRuntime targetFramework="4.5" />
  <compilation debug="true" targetFramework="4.5">
  <assemblies>
    <add assembly="Syncfusion.EJ.DashboardViewer.Mvc" />
    <add assembly="System.Runtime, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
  </assemblies>
  </compilation>
  <pages>
    <namespaces>
      <add namespace="System.Web.Helpers" />
      <add namespace="System.Web.Mvc" />
      <add namespace="System.Web.Mvc.Ajax" />
      <add namespace="System.Web.Mvc.Html" />
      <add namespace="System.Web.Routing" />
      <add namespace="System.Web.WebPages" />
    </namespaces>
  </pages>
</system.web>
```

ASP.NET CORE

ASP.NET Core 2.x Application

Overview

Dashboard Viewer can be embedded within your ASP.NET Core 2.x Web application through Syncfusion Dashboard Platform SDK.

System Requirements:

To work with ASP.NET Core 2.x, you need to make sure that you have installed the following software on your machine

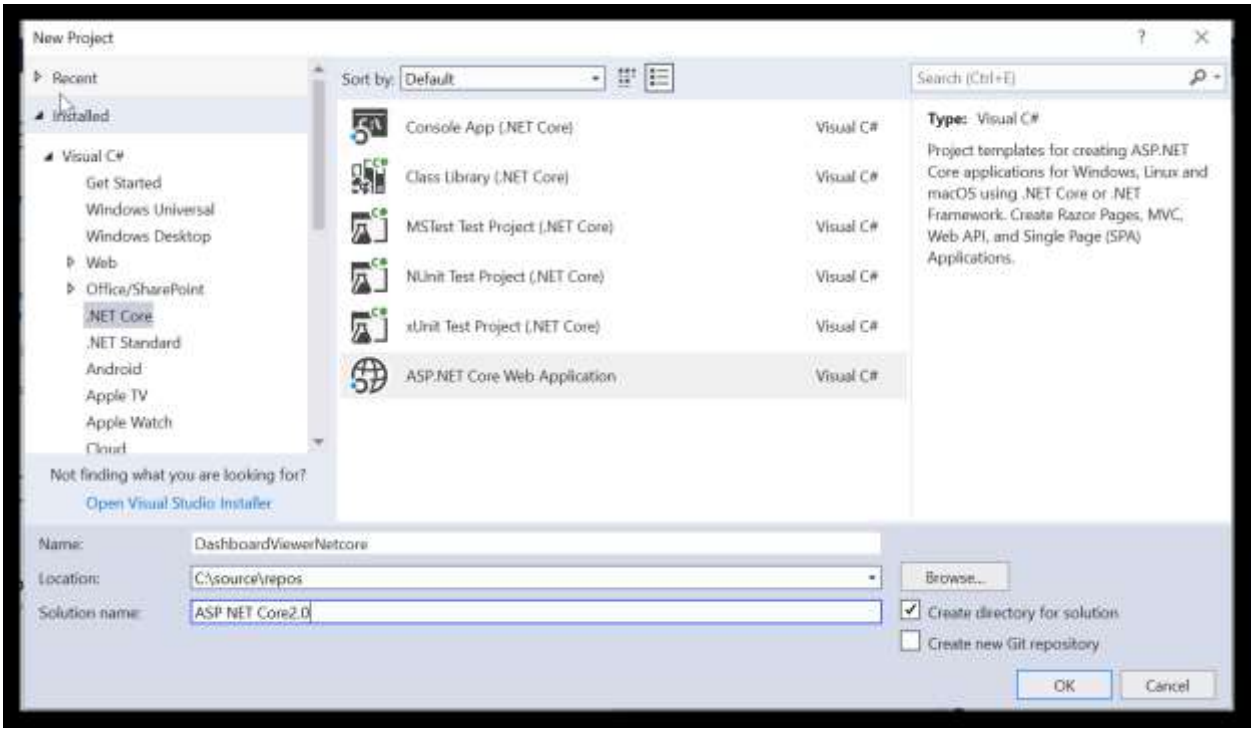
- Microsoft Visual Studio 2017 version 15.9.0 or higher.
- DotNetCore 2.0 or DotNetCore 2.1.

Configuring Syncfusion Dashboard Viewer Component in ASP.NET Core 2.x Application:

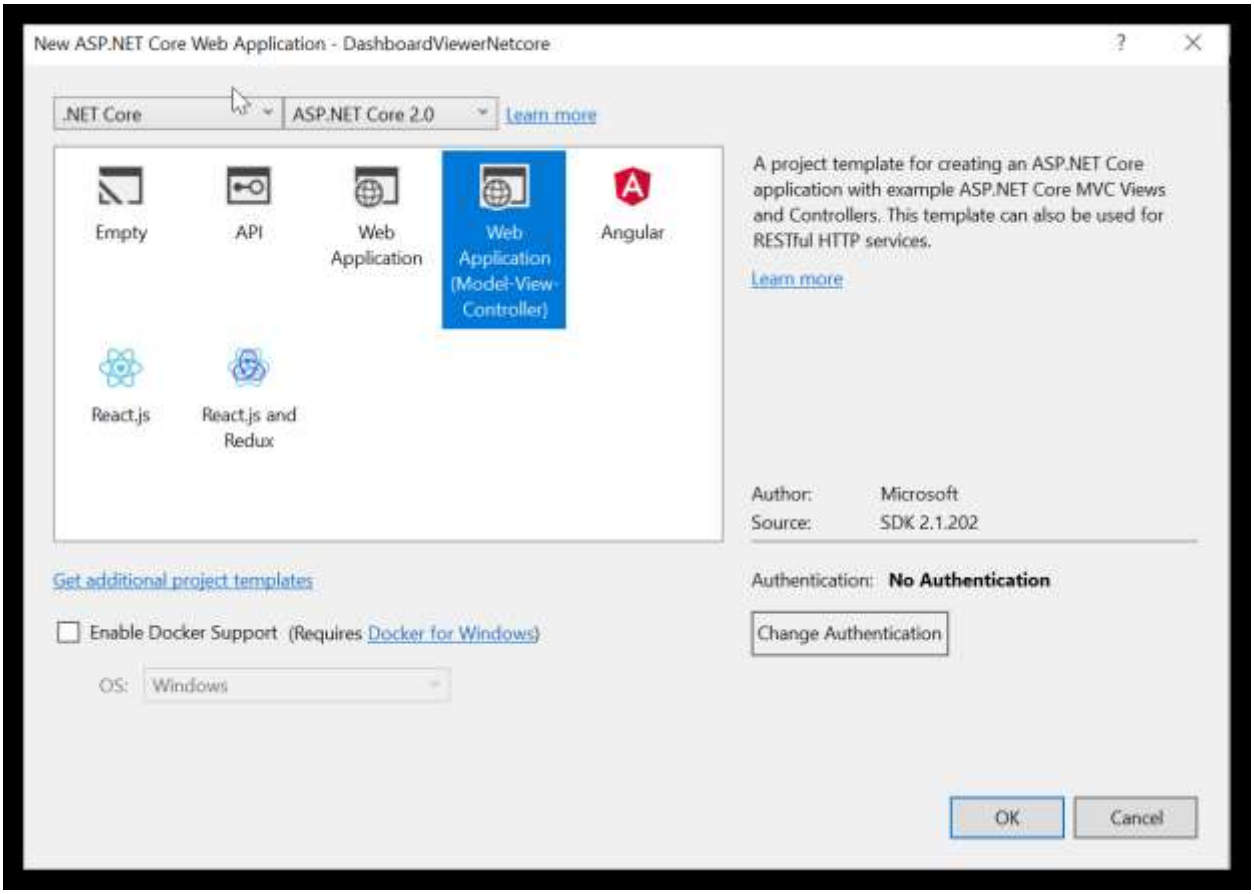
The following steps helps you to create an ASP.NET Core 2.x web application to embed your Dashboard Viewer component.

- Open Visual Studio 2017 or higher to create **ASP.NET 2.x Core web application**.

from File | New | Project and save it with a meaningful name as shown below (Select the .NET Core option, which is available by default in the listed Templates on the left side pane in the new project



Now select the web application(Model-View-Controller) and select the ASP.NET Core Version as in the below image



Refer here for more details for getting started with ASP.NET Core.

- After project creation, open your project file(csproj) to add our Syncfusion assembly packages (highlighted below).To open your project

file right click on the project file in Visual Studio and click the option Edit application.csproj

ASPX-CS

```
<PackageReference Include="Syncfusion.EJ.DashboardViewer.AspNet.Core"
Version="3.2.0.69" />
```

Adding Scripts and Style References

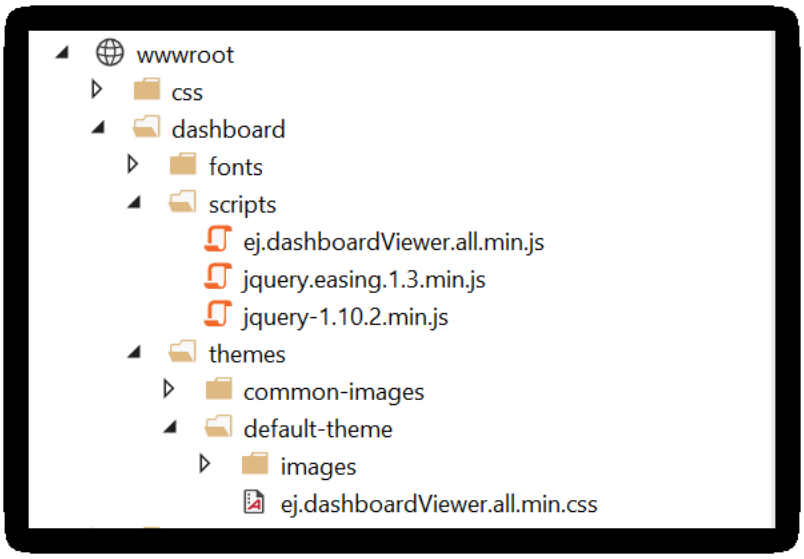
Create a new folder say, dashboard under wwwroot folder.

Note: Default project template consists of the wwwroot folder.

Copy the required scripts, themes and fonts into wwwroot\dashboard folder in your new ASP.NET Core 2.x Web application for rendering the ejDashboardViewer.

These can be availed from the Dashboard Platform SDK build installed location mentioned below:

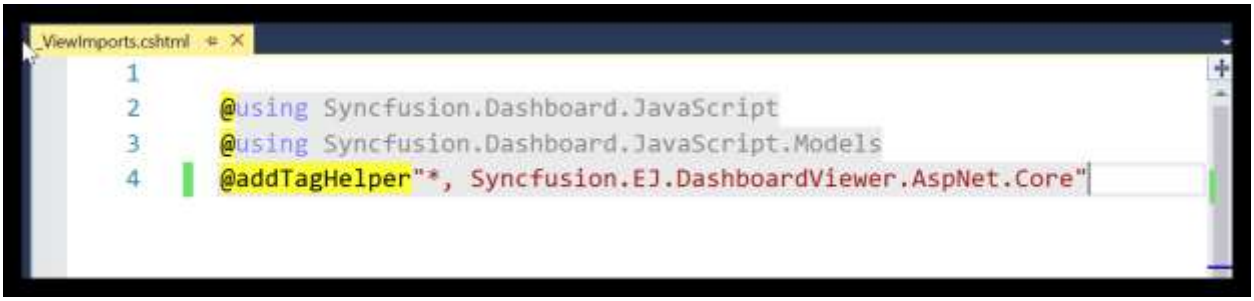
%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html



Now open _ViewImports.cshtml file from the Views folder and add the following namespace for components references and Tag Helper support.

ASPX-CS

```
@using Syncfusion.Dashboard.JavaScript
@using Syncfusion.Dashboard.JavaScript.Models
@addTagHelper"*, Syncfusion.EJ.DashboardViewer.AspNet.Core"
```



Refer the necessary scripts and CSS files in your _Layout.cshtml page from wwwroot -> dashboard folder.

ASPX-CS

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>AspNetCore - DashboardViewer</title>
<link rel="stylesheet" href="~/dashboard/themes/default-
theme/bootstrap.min.css" />
<link rel="stylesheet" href="~/dashboard/themes/default-
theme/ej.dashboardViewer.all.min.css" />
<script src="~/dashboard/scripts/jquery-1.10.2.min.js"></script>
<script src="~/dashboard/scripts/jquery.easing.1.3.min.js"></script>
<script src="~/dashboard/scripts/ej.dashboardViewer.all.min.js"></script>
```



```
</head>
</html>
```

Add ScriptManager to the bottom of the _Layout.cshtml page. The ScriptManager is used to place our control initialization script in the page.

ASPX-CS

```
<body style="height:100%;width:100%;padding:0;">
<div class="container body-content" style="height:100%;width:100%;">
@RenderBody()
<ej-script-manager></ej-script-manager>
</div>
@RenderSection("scripts", required: false)
</body>
```

- Now open your view page to render our Syncfusion Dashboard Viewer component in Tag Helper syntax.

ASPX-CS

```
@{
ViewData["Title"] = "DashboardViewer ASP.NET CORE 2.x Support";
}
<style>
body,  html,#dashboard {
overflow: hidden !important;
height:100%;
width:100%;
}
</style>
<ej-dashboardviewer id="dashboard" service-
url="https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.s
vc" dashboard-
path="https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDash
board.sydx" ></ej-dashboardviewer>
```

Note: Provided online Dashboard Service URL and Dashboard Path are for demo purpose.

Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

Configuring Route

In the Startup.cs file inside the method configure use the below code to configure the route

C#

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env,
ILoggerFactory loggerFactory)
{
loggerFactory.AddConsole(Configuration.GetSection("Logging"));
loggerFactory.AddDebug();
if (env.IsDevelopment())
{
app.UseDeveloperExceptionPage();
```



```
app.UseBrowserLink();
}
else
{
app.UseExceptionHandler("/Home/Error");
}
app.UseStaticFiles();
app.UseMvc(routes =>
{
routes.MapRoute(
name: "default",
template: "{controller=Home}/{action=Index}/{id?}");
});
}
```

Configuring Startup Class

ASP.NET Core app uses startup class so in the program.cs file make the below code changes

C#

```
public static void Main(string[] args)
{
var host = new WebHostBuilder()
.UseKestrel()
.UseContentRoot(Directory.GetCurrentDirectory())
.UseIISIntegration()
.UseStartup<Startup>()
.Build();
host.Run();
}
```

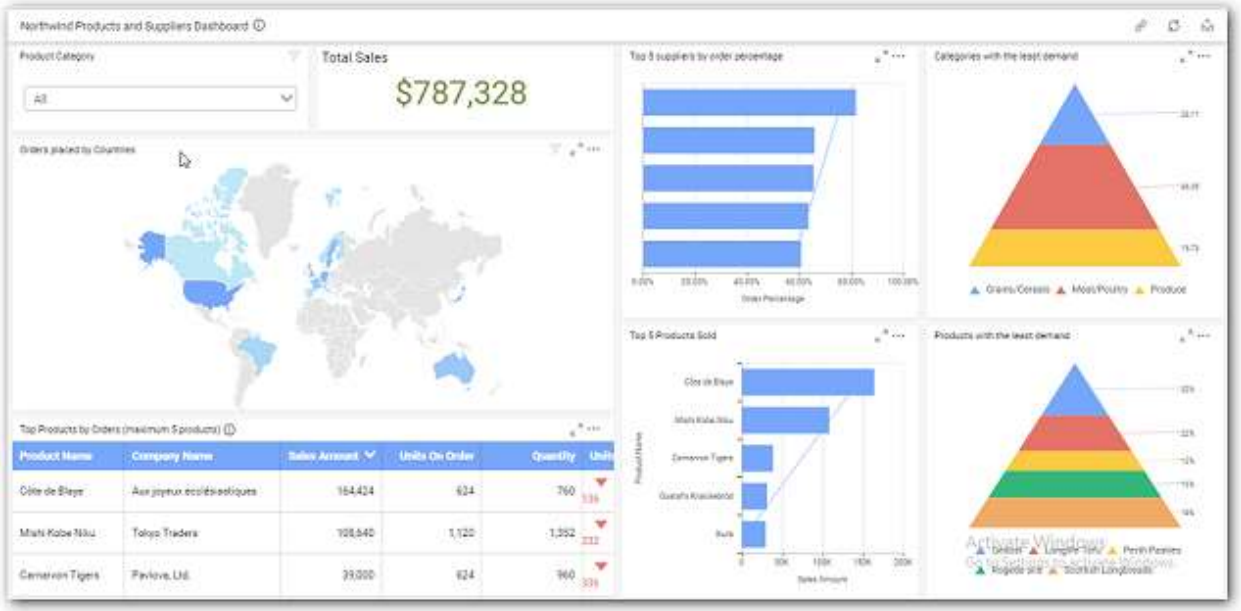
Binding Dashboard Service

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

- Finally compile your project. After successful compilation, press F5 key to deploy your project.

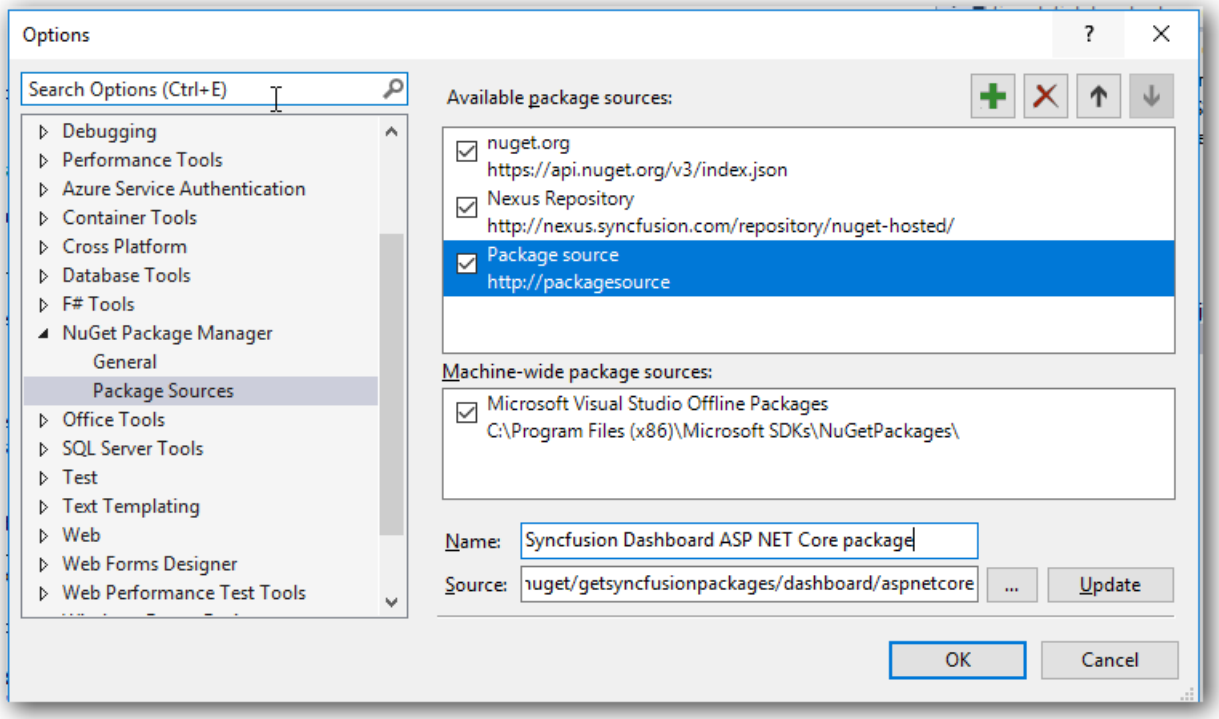


Configuring and Installing Syncfusion NuGet Packages in Visual Studio

NuGet Configuration

The steps to install the Syncfusion Dashboard ASP.NET Core NuGet Packages in Visual Studio are as follows,

1. In Visual Studio, navigate to Tools | NuGet Package Manager | Package Manager Settings, the options dialog will appear on the screen as shows below,

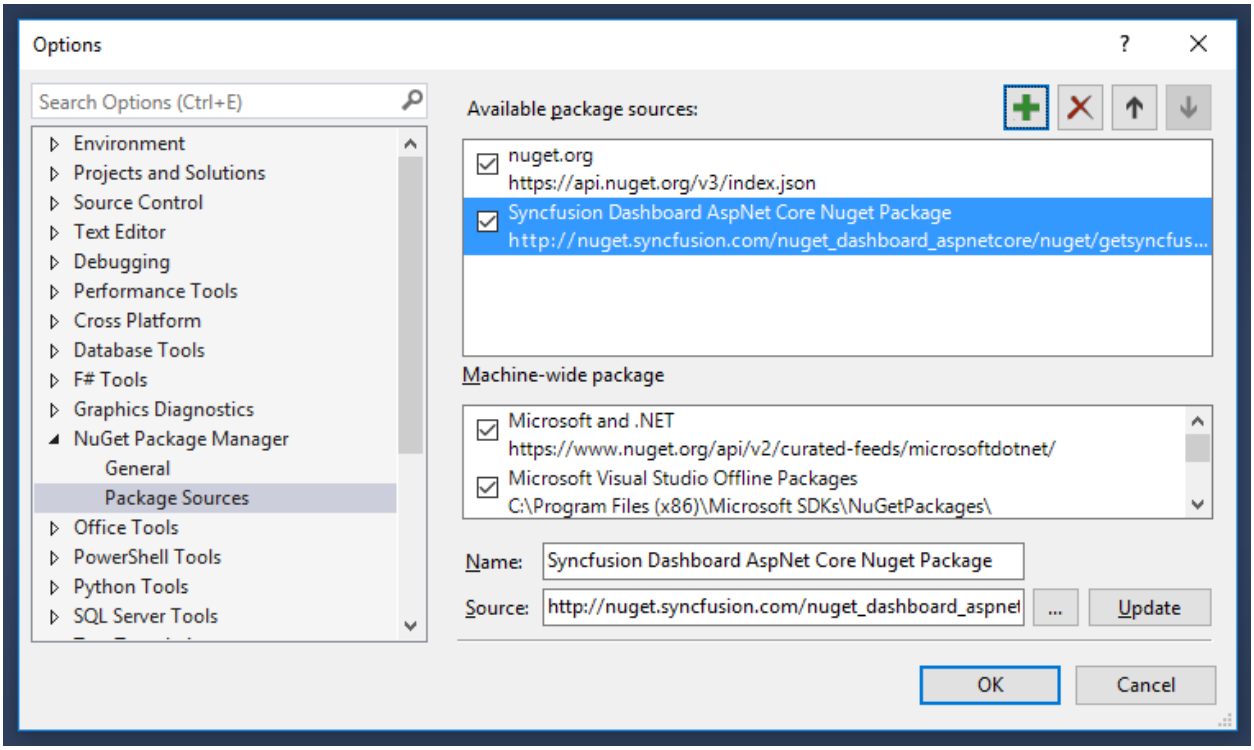


2. Select NuGet Package Manager | Package Sources and click Add button to add the Package Name and Package Source of Syncfusion Dashboard NuGet Packages.

Name: Name of the package that listed in Available package sources

Source: Syncfusion Dashboard ASP.NET Core NuGet Package feed [URL]

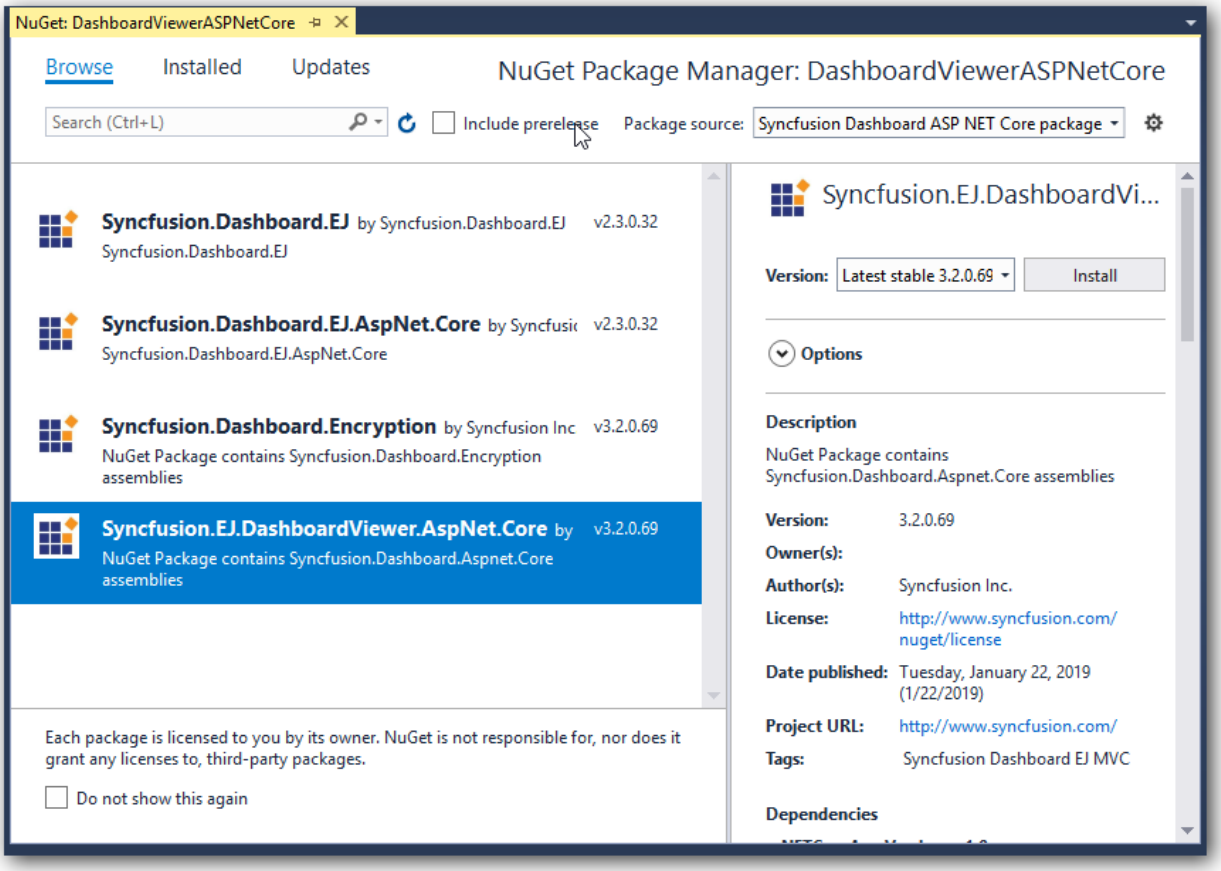
(http://nuget.syncfusion.com/nuget*dashboard*aspnetcore/nuget/getsyncfusionpackages/dashboard/aspnetcore/)



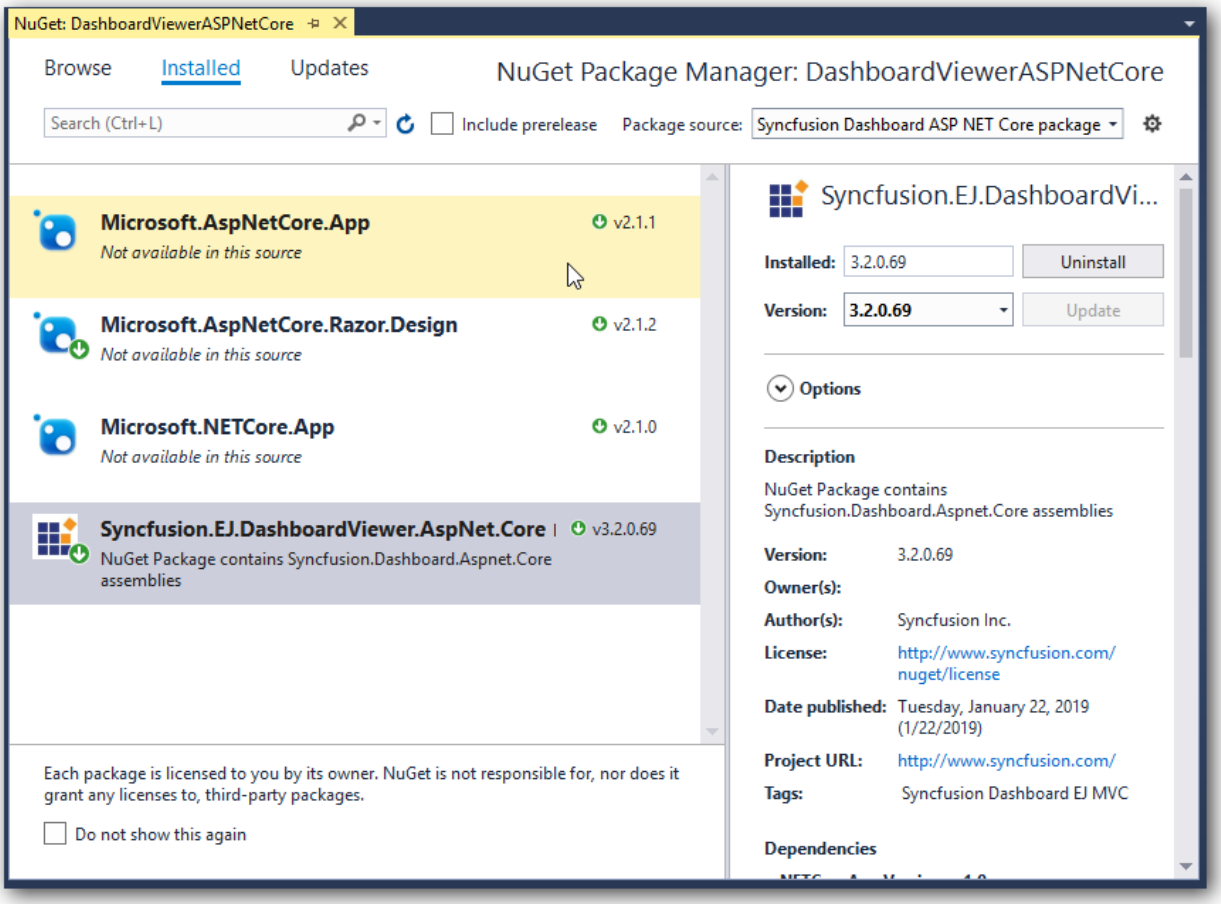
NuGet Installation

Syncfusion ASP.NET Core NuGet can install once configured the package source. The NuGet installation steps as below,

1. Once configured the Package source with Syncfusion NuGet Packages, right click on project and choose Manage NuGet Packages | Browse.



2. The NuGet Packages are listed which are available in package source location. Install the required packages to your application by clicking Install button. 3. Now you can find the installed NuGet package under the Installed tab list as shown below.



ASP.NET Core 1.0.1 Application

Overview

The Syncfusion Dashboard Platform SDK includes a Dashboard Viewer HTML 5 widget that can be embedded within your ASP.NET Core Web application.

System Requirements:

To work with ASP.NET Core 1.0.1, you need to make sure whether you have installed the following software on your machine

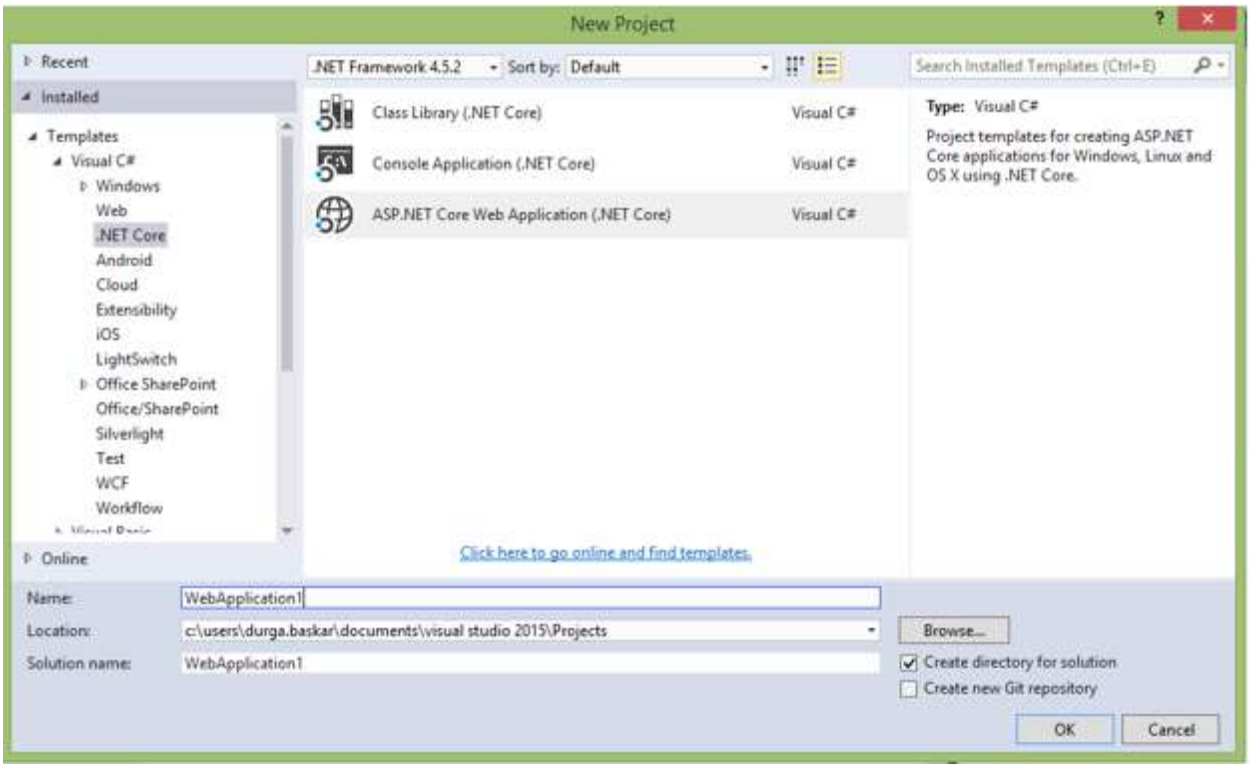
- Microsoft Visual Studio 2015 Update 3 or higher.
- DotNetCore 1.0.1 (Preview 2).

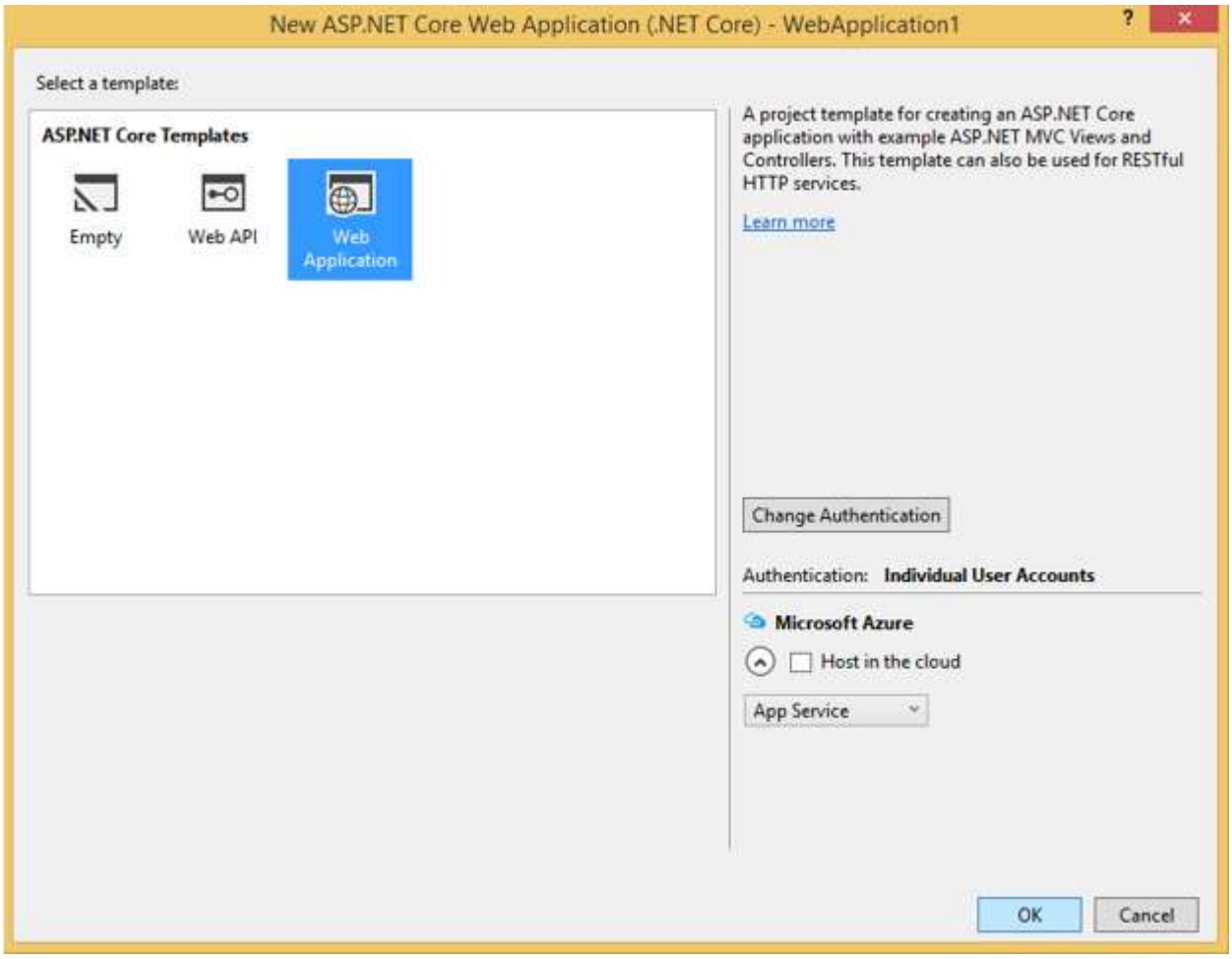
Configuring Syncfusion Dashboard Viewer Component in ASP.NET Core Application:

The following steps helps you to create an ASP.NET Core web application to configure your Dashboard Viewer component.

- Open Visual Studio 2015 or higher to create **ASP.NET Core web application**.

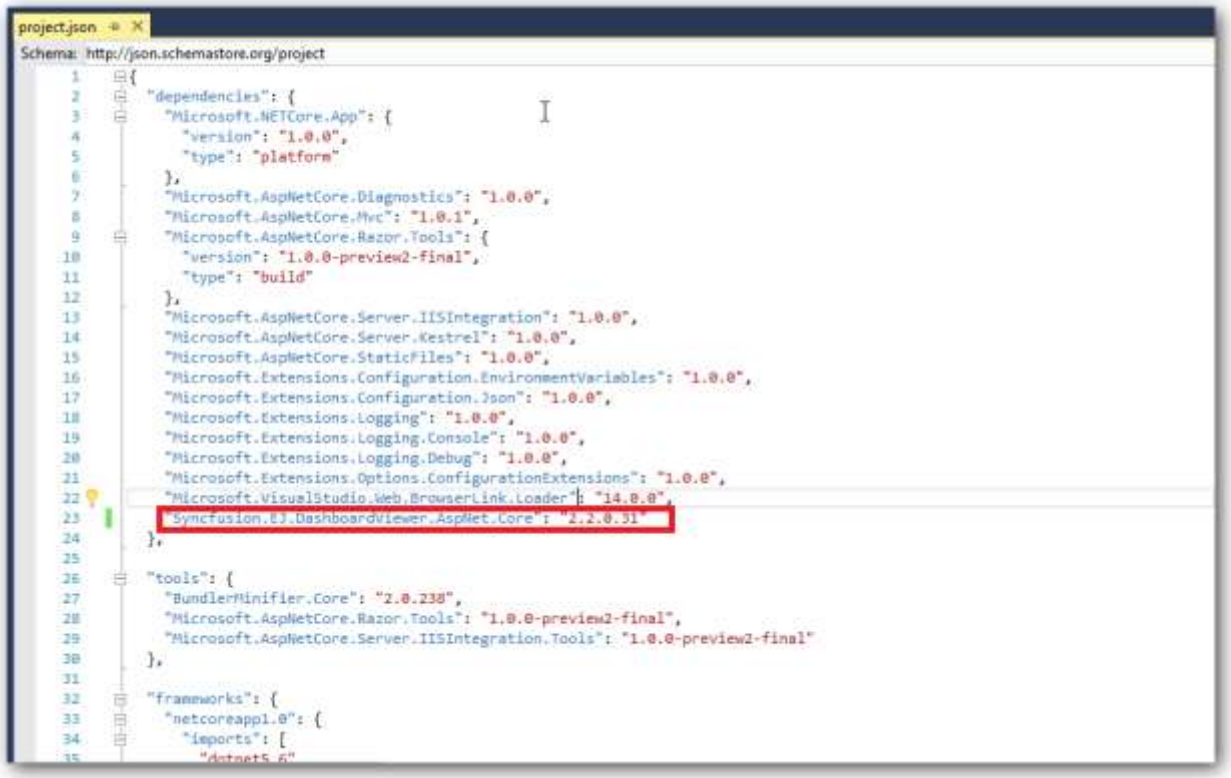
from File | New | Project and save it with a meaningful name as shown below (Select the .NET Core option, which is available by default in the listed Templates on the left side pane in the new project


New Project
Recent
Installed
Templates
Visual C#
Windows
Web
.NET Core
Android
Cloud
Extensibility
iOS
LightSwitch
Office SharePoint
Office/SharePoint
Silverlight
Test
WCF
Workflow
Online
.NET Framework 4.5.2
Sort by: Default
Search Installed Templates (Ctrl+E)
Class Library (.NET Core) Visual C#
Console Application (.NET Core) Visual C#
ASP.NET Core Web Application (.NET Core) Visual C#
Type: Visual C#
Project templates for creating ASP.NET Core applications for Windows, Linux and OS X using .NET Core.
Click here to go online and find templates.
Name: WebApplication1
Location: c:\users\durga.baskar\documents\visual studio 2015\Projects
Solution name: WebApplication1
Browse...
Create directory for solution
Create new Git repository
OK
Cancel



Refer here for more details.

- After project creation, open your project.json file to add our Syncfusion assembly packages (highlighted below).



Adding Scripts and Style References

Create a new folder say, dashboard under wwwroot folder.

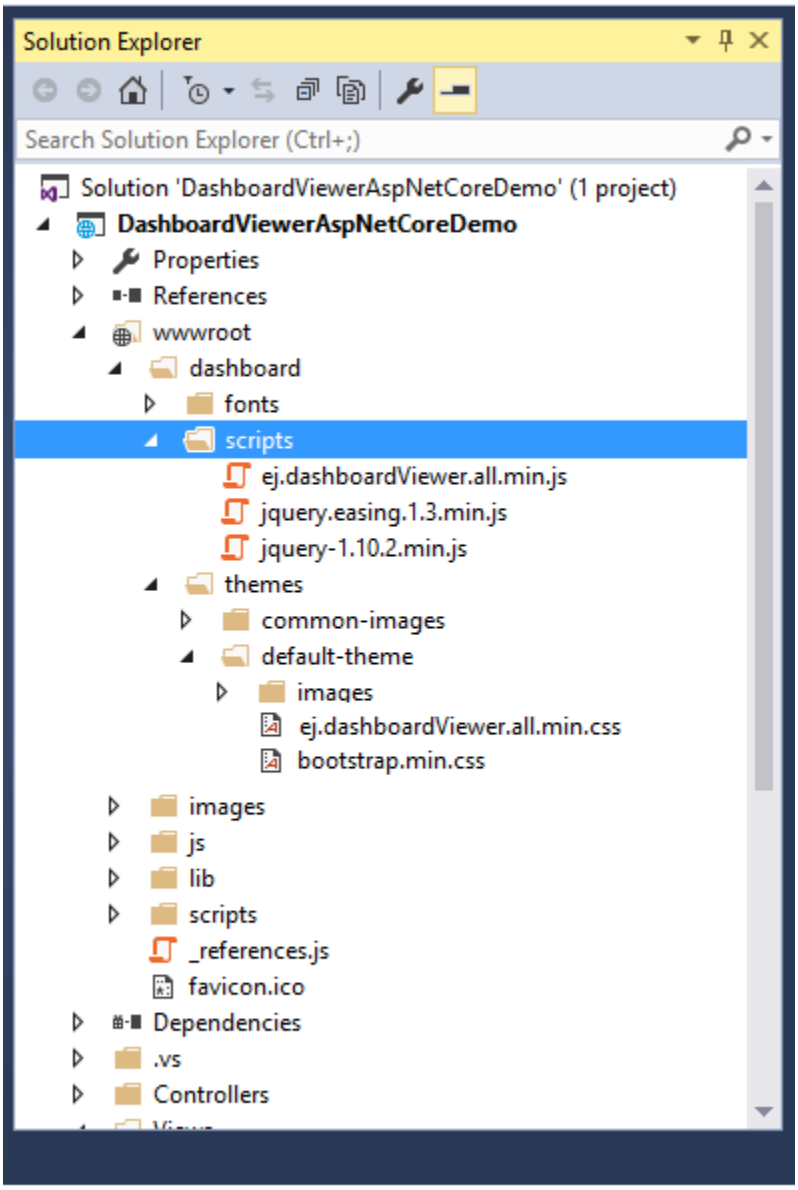
Note: Default project template consists of the wwwroot folder.

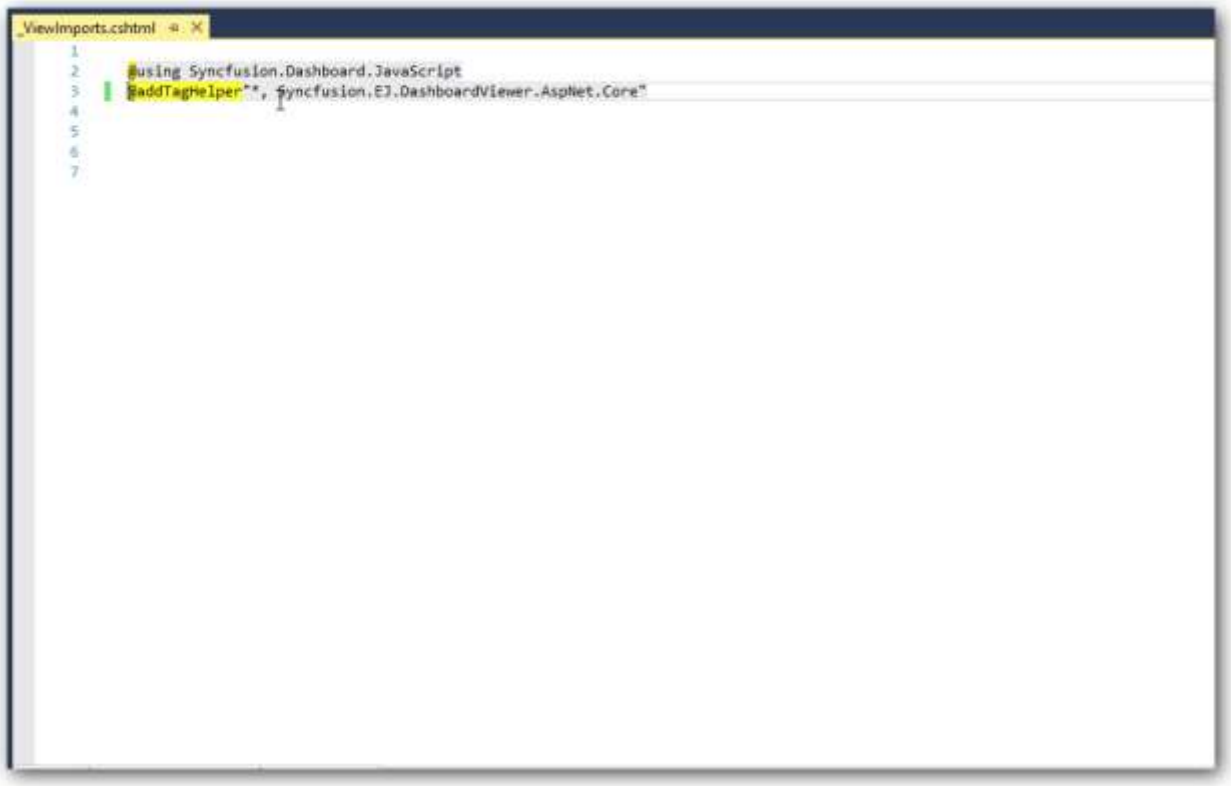
Copy the required scripts, themes and fonts into wwwroot\dashboard folder in your new ASP.NET Core Web application for rendering the ejDashboardViewer.

These can be availed from the Dashboard Platform SDK build installed location mentioned below:

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html

Now open _ViewImports.cshtml file from the Views folder and add the following namespace for components references and Tag Helper support.





Refer the necessary scripts and CSS files (highlighted below) in your layout.cshtml page from wwwroot -> dashboard folder.

```
_Layout.cshtml
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>AspNetCore - DashboardViewer</title>
    <link rel="stylesheet" href="~/dashboard/themes/default-theme/bootstrap.min.css" />
    <link rel="stylesheet" href="~/dashboard/themes/default-theme/ej.dashboardViewer.all.min.css" />
    <script src="~/dashboard/scripts/jquery-1.10.2.min.js"></script>
    <script src="~/dashboard/scripts/jquery.easing.1.3.min.js"></script>
    <script src="~/dashboard/scripts/ej.dashboardViewer.all.min.js"></script>
</head>
<body style="height:100%;width:100%;padding:0;">
    <div class="container body-content" style="height:100%;width:100%;">
        @RenderBody()
        <ej-script-manager></ej-script-manager>
    </div>
    @RenderSection("scripts", required: false)
</body>
</html>
```

Add ScriptManager to the bottom of the layout.cshtml page. The ScriptManager is used to place our control initialization script in the page.

ASPX-CS

```
<ej-script-manager></ej-script-manager>
```


- Now open your view page to render our Syncfusion Dashboard Viewer component in Tag Helper syntax.

ASPX-CS

```
@{
ViewData["Title"] = "DashboardViewer ASP.NET CORE Support";
}
<style>
body,  html,#dashboard {
overflow: hidden !important;
height:100%;
width:100%;
}
</style>
<ej-dashboardviewer id="dashboard" service-
url="https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.s
vc" dashboard-
path="https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDash
board.sydx" ></ej-dashboardviewer>
```

Note: Provided online Dashboard Service URL and Dashboard Path are for demo purpose.

Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

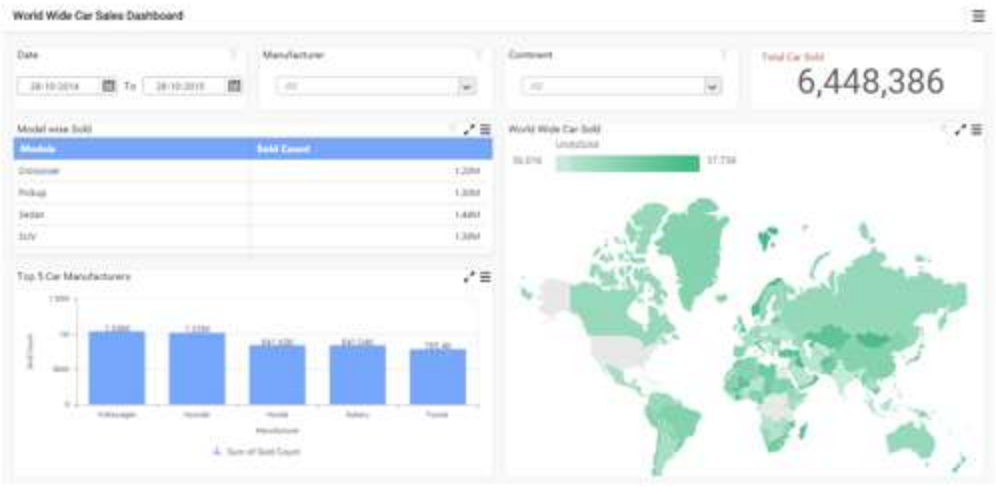
Binding Dashboard Service

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

- Finally compile your project. After successful compilation, press F5 key to deploy your project.

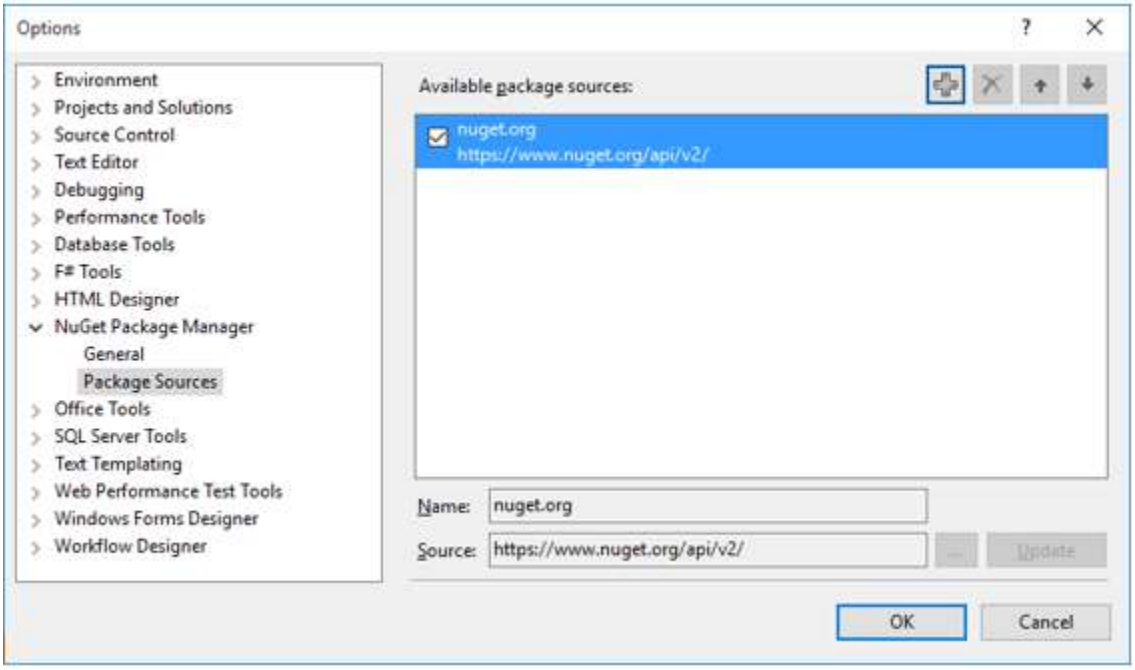


Configuring and Installing Syncfusion NuGet Packages in Visual Studio

NuGet Configuration

The steps to install the Syncfusion Dashboard ASP.NET Core NuGet Packages in Visual Studio are as follows,

1. In Visual Studio, navigate to Tools | NuGet Package Manager | Package Manager Settings, the options dialog will appear on the screen as shows below,

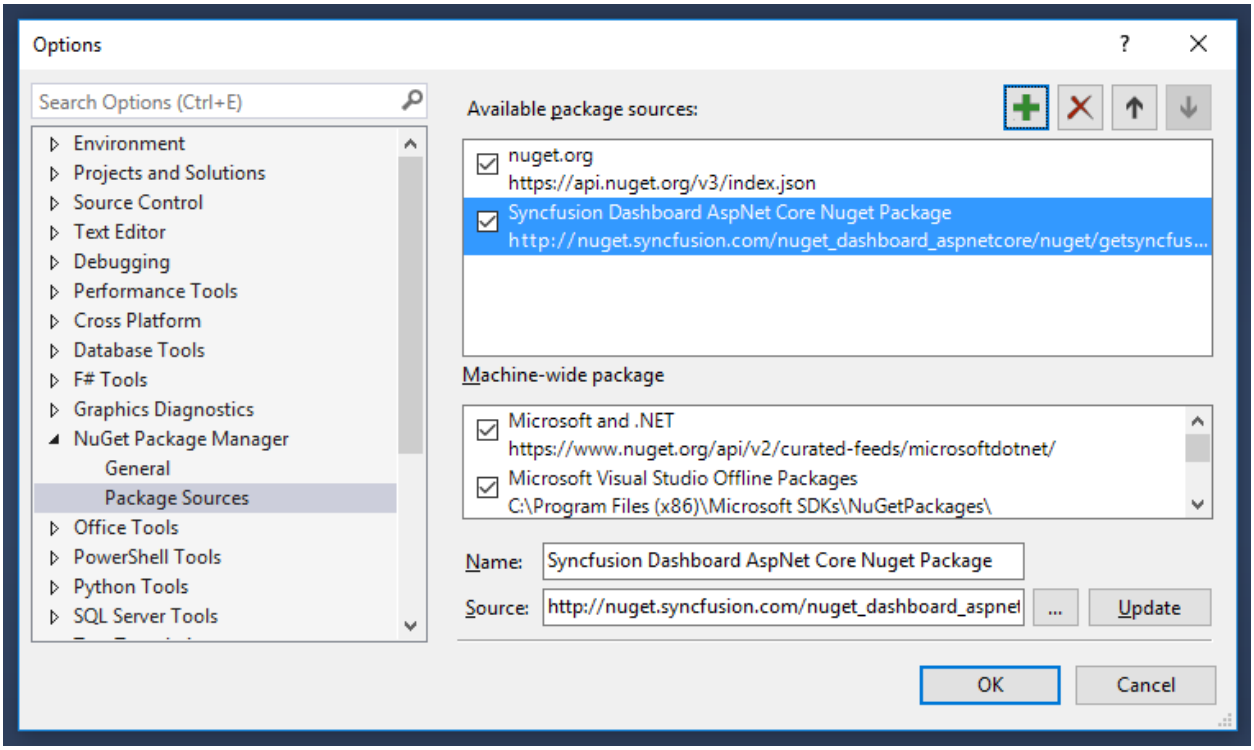


2. Select NuGet Package Manager | Package Sources and click Add button to add the Package Name and Package Source of Syncfusion Dashboard NuGet Packages.

Name: Name of the package that listed in Available package sources

Source: Syncfusion Dashboard ASP.NET Core NuGet Package feed URL

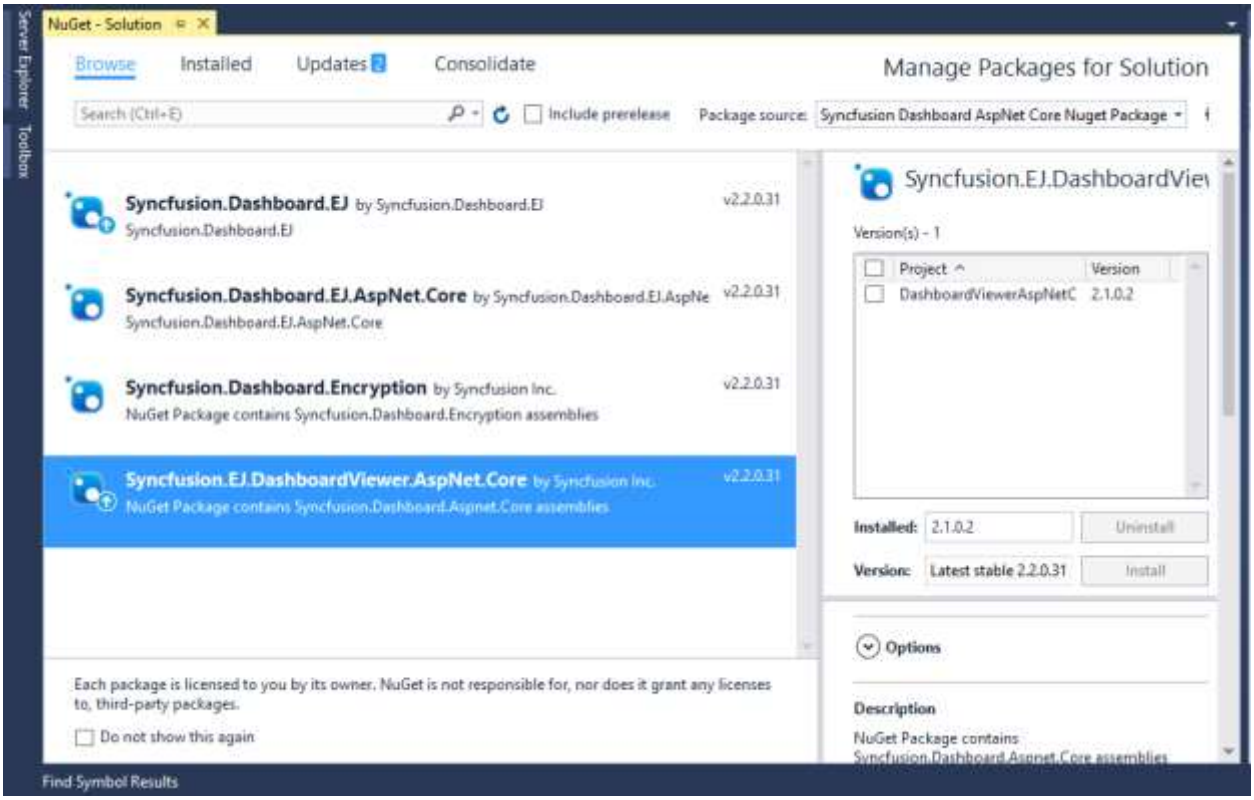
[http://nuget.syncfusion.com/nuget*dashboard*aspnetcore/nuget/getsyncfusionpackages/dashboard/aspnetcore/]



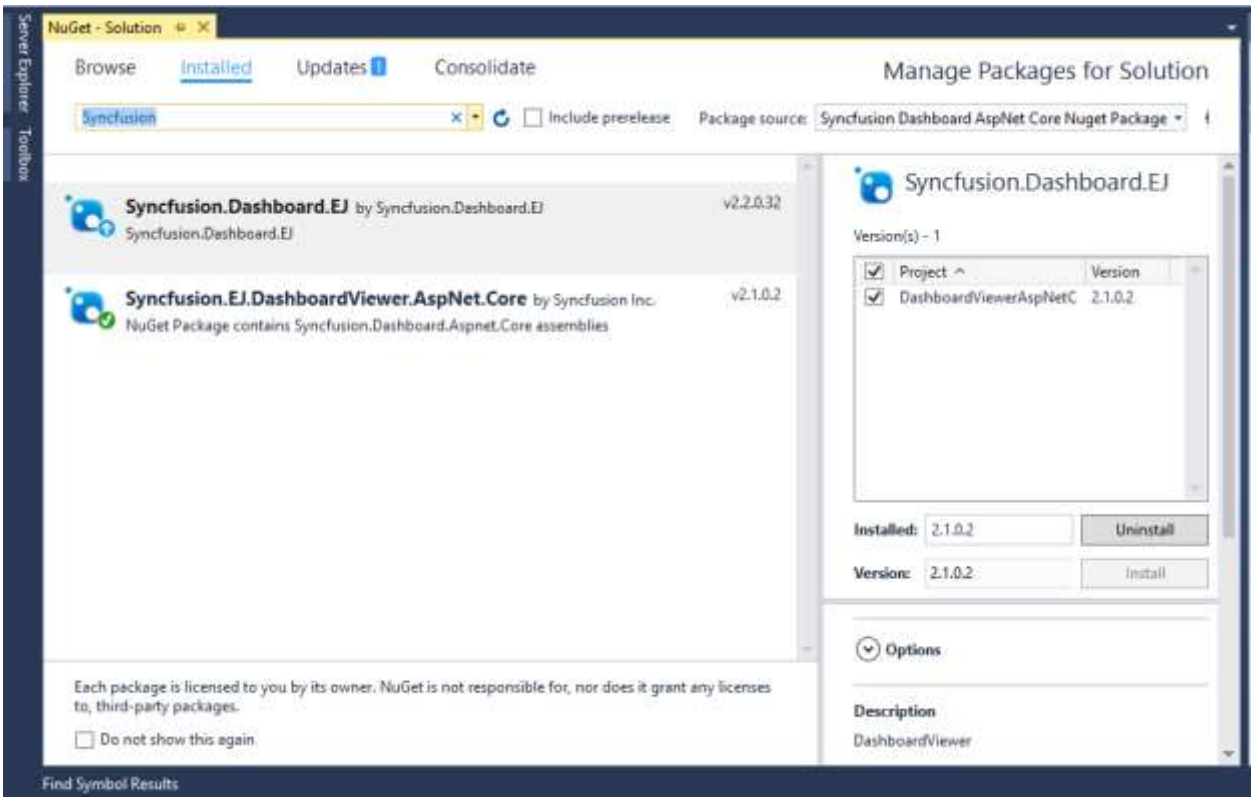
NuGet Installation

Syncfusion ASP.NET Core NuGet can install once configured the package source. The NuGet installation steps as below,

1. Once configured the Package source with Syncfusion NuGet Packages, right click on project and choose Manage NuGet Packages | Browse | .



2. The NuGet Packages are listed which are available in package source location. Install the required packages to your application by clicking Install button. 3. Now you can find the installed NuGet package under the Installed tab list as shown below.



Aurelia

Overview

Syncfusion Dashboard Platform SDK includes Aurelia wrapper for DashboardViewer. The properties of DashboardViewer are defined as attributes for Aurelia component, so you can easily set value to its properties, simply by prefixing property name with e- in component markup.

Getting started

This section guides you to create a web application using Aurelia Framework with DashboardViewer widget wrapper.

Creating an Aurelia Application with Dashboard Viewer Wrapper

Create a folder testapp. Copy the contents of `aurelia app` folder into your newly created `testapp` folder from Dashboard Platform SDK build installed location mentioned below.

%localappdata%\Syncfusion\Dashboard\Samples\JavaScript\aurelia app

* Clear the contents of `src/dashboardviewer` folder (`src` folder is present inside the `testapp` folder).

Binding Dashboard Service

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Initializing Dashboard Viewer

* Create dashboardviewer.html view file inside src/dashboardviewer folder and render ejDashboardViewer Aurelia component using the below code example.

HTML

```
<template>
<require from="./dashboardviewer.css"></require>
<div id="dashboardsample">
<section>
<div class="row">
<div>
<ej-dashboard-viewer  id=dashboard e-service-url.bind="url" e-dashboard-
path.bind="dashboardPath" e-size.bind="size">
</ej-dashboard-viewer>
</div>
</div>
</section>
</div>
</template>
```

* Create dashboardviewer.js file inside src/dashboardviewer folder and populate the properties with respective values as in the below code example.

JAVASCRIPT

```
export class BasicUse {
constructor() {
this.url = '
https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc';
this.dashboardPath =
'https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDashboard
.sydx';
this.size = {width: '100%', height: '100%'};
}
}
```

Note: Provided online Dashboard Service URL and Dashboard Path are for demo purpose.

Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

- Create dashboardviewer.css file inside src/dashboardviewer folder and add the below code example.

CSS

```
body, html, #dashboard, ej-dashboard-viewer {
overflow: hidden !important;
width: 100%! important;
height: 100%!important;
}
```

- Create index.js file inside src/dashboardviewer folder and add the below code example.

JAVASCRIPT

```
import {useView, inject} from 'aurelia-framework';
import {Registry} from 'shared/registry';
@useView('shared/showcase.html')
@inject(Registry)
export class Index {
constructor(registry) {
this.registry = registry;
}
configureRouter(config, router) {
this.router = router;
return this.registry.load(config, 'dashboardviewer');
}
}
```

* Create registry.json file inside src/dashboardviewer folder for route configurations and add the below code example.

JSON

```
{
"title": "DashboardViewer",
"documentation":"https://help.syncfusion.com/dashboard-platform/dashboard-sdk/ejdashboardviewer",
"samples": {
"default":{
"files": ["html", "js","css"],
"default": "true"
}
}
}
```

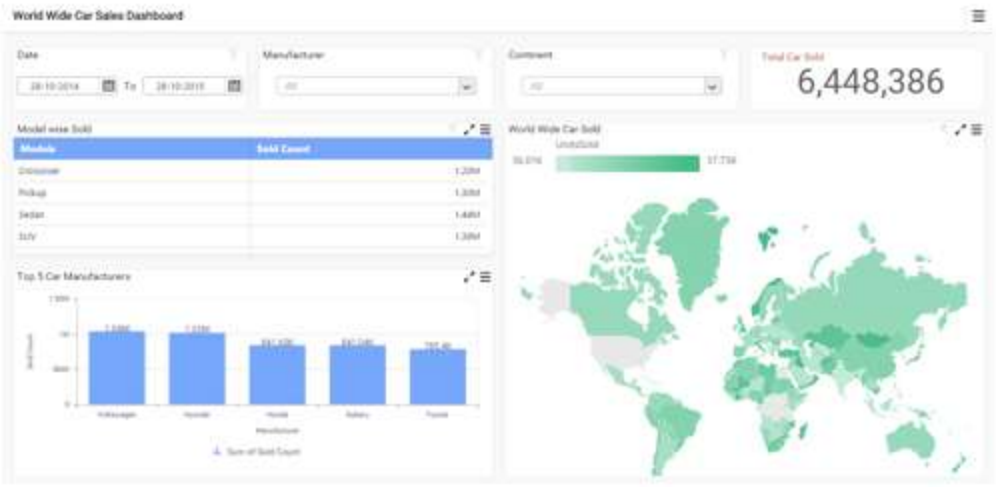
Running the application

Open Command Prompt window and navigate to “testapp” folder and type the below commands to install the required packages and run the Aurelia application.

JAVASCRIPT

```
> npm install
> npm install jspm -global
> jspm install
> "copy.bat"
> gulp watch
```

- Browse to http://localhost:9000 to see the application. The dashboard gets rendered as like in the below screenshot.



Embedding Dashboard Viewer in a HTML page

HTML page creation

Create a new document and save it as .html extension.

Adding required files in HTML page

Note: Follow the link to install Dashboard Platform SDK.

Adding help files

After installing Dashboard SDK setup, scripts, themes, font folders will be found in the below location.

`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html`

Include these scripts, themes, font folders in your application and refer the files of each folder in the HTML page as mentioned below:

ASPX-CS

```
<head>
<script src="scripts/jquery-1.10.2.min.js"></script>
<script src="scripts/jquery.easing.1.3.min.js"></script>
<script src="scripts/ej.dashboardViewer.all.min.js"></script>
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet" />
</head>
```

Hosting Dashboard Service

After installing the Dashboard SDK setup you will find Service and DashboardServiceInstaller folder in the following location and these folders are required to start the Dashboard Service.

`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common`

To host the Dashboard Service in IIS, Please run the `SyncfusionDashboardServiceInstaller-IIS.exe` from the following location and enter the port number in which dashboard service to be hosted.

`%localappdata%\Syncfusion\Dashboard Platform SDK\DashboardServiceInstaller`

For Example:

If you are hosting Dashboard Service at port #607 in localhost, then Dashboard Service URL will be framed as below:

```
http://localhost:607/DashboardService.svc
```

Open this URL in any of the browser and make sure Dashboard Service is running.

Adding Dashboard Service and Dashboard File Location

Add the hosted Dashboard Service URL and the Dashboard file (*.SYDX) file location to the Dashboard Viewer like below.

ASPX-CS

```
<div id="dashboard" style="width: 100%; height: 100%;" ></div>
<script type="text/javascript">
$(document).ready(function () {
$('#dashboard').ejDashboardViewer({
serviceUrl: "http://localhost:607/DashboardService.svc",
dashboardPath:
"D:\\Syncfusion\\Dashboard\\Dashboards\\WorldWideCarSalesDashboard.sydx",
});
});
</script>
```

Note: The Dashboard file path should be accessible from the provided Dashboard Service path (or link).

Note: To initiate the dashboard service instance you can follow anyone of the below methods

Note: 1. Hosting Dashboard Service in IIS

Note: 2. Hosting Dashboard Service in IIS Express

Note: 3. Hosting Dashboard Service as Windows Service Background Process

Finally you get the complete Dashboard Viewer Sample structure like below.

ASPX-CS


```
<html>
<style>
body, html {
height: 100%;
overflow: auto !important;}
</style>
<head>
<script src="scripts/jquery-1.10.2.min.js"></script>
<script src="scripts/jquery.easing.1.3.min.js"></script>
<script src="scripts/ej.dashboardViewer.all.min.js"></script>
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet" />
</head>
<body>
<div id="dashboard" style="width: 100%; height: 100%;" ></div>
<script type="text/javascript">
$(document).ready(function () {
$('#dashboard').ejDashboardViewer({
serviceUrl: "http://localhost:607/DashboardService.svc",
dashboardPath:
"D:\\Syncfusion\\Dashboard\\Dashboards\\WorldWideCarSalesDashboard.sydx",
});
});
</script>
</body>
</html>
```

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Overview

Syncfusion Dashboard Platform SDK includes ionic framework. Ionic is an open source Mobile SDK to develop native and progressive web apps, and it works with newer version of Angular 4 and typescript.

Prerequisites and compatibility

Prerequisites

While running the ionic it requires Node.js (which includes npm) in the system.

Install ionic

Install the latest Cordova and Ionic command-line tools in your terminal. Follow the Windows and Mac guides to install required tools for development.

JS

```
npm install -g cordova ionic
```

Getting started with dashboard viewer in ionic application

This section guides you to start with ionic framework for the dashboard viewer through the step-by-step instructions.

Step 1 : Install the latest version of the ionic CLI. Run the following commands.

JS

```
npm install -g ionic@latest
```


Step 2 : Run the following command to create an Ionic sample project.

JS

```
ionic start Syncfusion-Dashboard blank
```

Note : Syncfusion-Dashboard is the project name and blank is the project template.

Adding JavaScript and CSS reference

Copy the scripts, themes, and fonts folder from the Dashboard Platform SDK build installed location which is mentioned below, and paste them in the src\assets folder under the newly created Syncfusion-Dashboard project.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JavaScript\ionic app\src\assets

Rename the folder, file, and reference name as dashboard from home.

For example : Change home.component.ts to dashboard.component.ts.

In the dashboard folder, create a new component as dashboard.component.ts and import the scripts/themes references in the order mentioned in the following code example.

TS

```
import $ from 'jquery';
import '../../../node_modules/jquery/src/jquery';
import '../../../node_modules/jsrender/jsrender';
import '../../assets/scripts/jquery.easing.min';
import '../../assets/scripts/ej.dashboardViewer.all.min.js';
```

Initializing and configuring the widget

Create a new folder as ej in the src/pages folder.

Copy the angular dashboardviewer source files from the following Dashboard Platform SDK installed location and paste them into the ej folder.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JavaScript\ionic\src\pages\ej

Open the dashboard.component.html file in the dashboard folder and initialize the ejDashboardViewer Angular component using the following code example.

HTML

```
<div>
<ej-dashboardviewer  id="dashboard" [serviceUrl]="url"
[dashboardPath]="dashboardPath"></ej-dashboardviewer>
</div>
```

Import and declare the Syncfusion source component and EJ*DASHBOARDVIEWER*COMPONENTS component in the src\app\app.module.ts file using the following code snippet.

TS


```
import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
import { SplashScreen } from '@ionic-native/splash-screen';
import { StatusBar } from '@ionic-native/status-bar';
import { MyApp } from './app.component';
import { EJ_DASHBOARDVIEWER_COMPONENTS } from
'../pages/ej/dashboardviewer.component';
import { DashboardComponent } from '../pages/dashboard/dashboard.component';
@NgModule({
declarations: [
MyApp,
DashboardComponent,
EJ_DASHBOARDVIEWER_COMPONENTS
],
imports: [
BrowserModule,
IonicModule.forRoot(MyApp),
],
bootstrap: [IonicApp],
entryComponents: [
MyApp,
DashboardComponent
],
providers: [
StatusBar,
SplashScreen,
{provide: ErrorHandler, useClass: IonicErrorHandler}
]
})
export class AppModule {}
```

Copy and paste the following code in the dashboardviewer.component.ts file under the dashboard folder.

TS

```
import { Component } from '@angular/core';
import { NavigationController } from 'ionic-angular';
import $ from 'jquery';
import '../../../node_modules/jquery/src/jquery';
import '../../../node_modules/jsrender/jsrender';
import '../../assets/scripts/jquery.easing.min';
import '../../assets/scripts/ej.dashboardViewer.all.min.js';
declare var $:any;
@Component({
selector: 'ion-app',
templateUrl: 'dashboard.component.html',
})
export class DashboardComponent {
public url; report; dashboardPath;
constructor(public navigationCtrl: NavigationController) {
this.url =
"https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc";
//Service URL
```



```
this.dashboardPath =
"https://dashboardsdk.syncfusion.com//Dashboards//Northwind Traders Sales
Analysis.sydx"; //path of the Dashboard
}
}
```

Note: The online dashboard service URL and dashboard path are provided for the demo purpose.

Make sure whether the given dashboard path should be accessible with the given Dashboard Service URL.

Binding dashboard service

To initiate the dashboard service instance, you can follow anyone of the following methods:

1. Hosting dashboard service in IIS.
2. Hosting dashboard service in IIS express.
3. Hosting dashboard service as Windows service background process.

Information: Hosting dashboard service at IIS is recommended to production environment for object management and other memory management features.

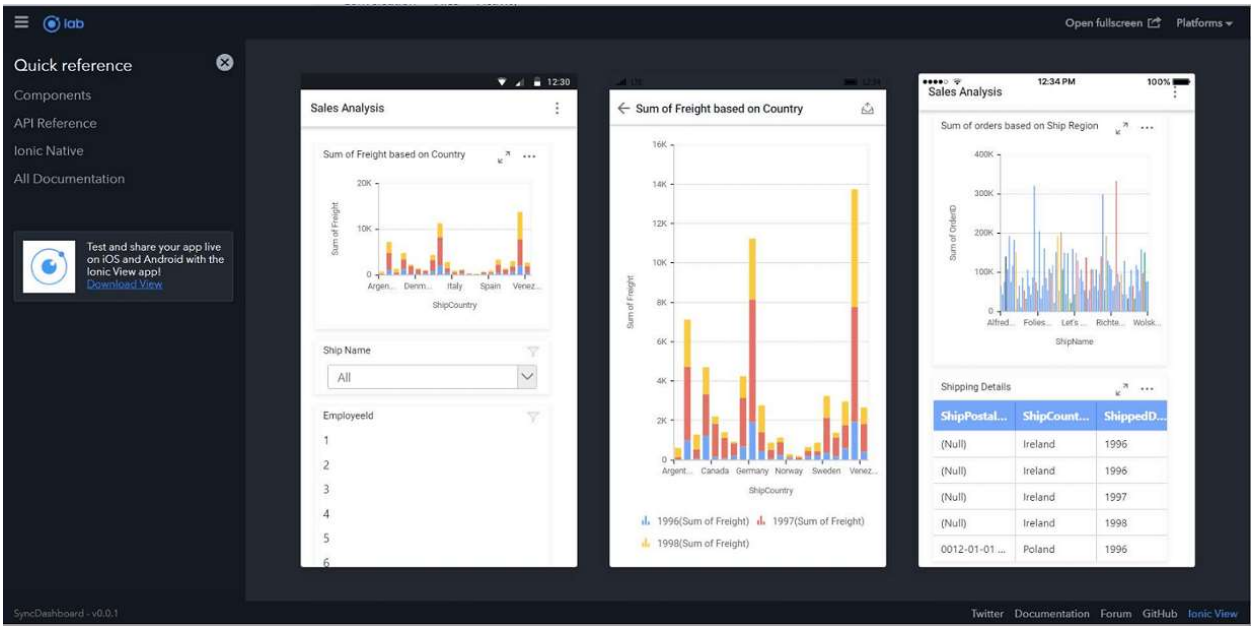
Running the application

- To run the application, execute the following command.

JAVASCRIPT

```
npm install
ionic serve
```

- Browse to http://localhost:8200, to see the application. The component is rendered as shown in the following screenshot. You can make changes in the code found under the src folder, and the browser should auto-refresh itself while saving the files.

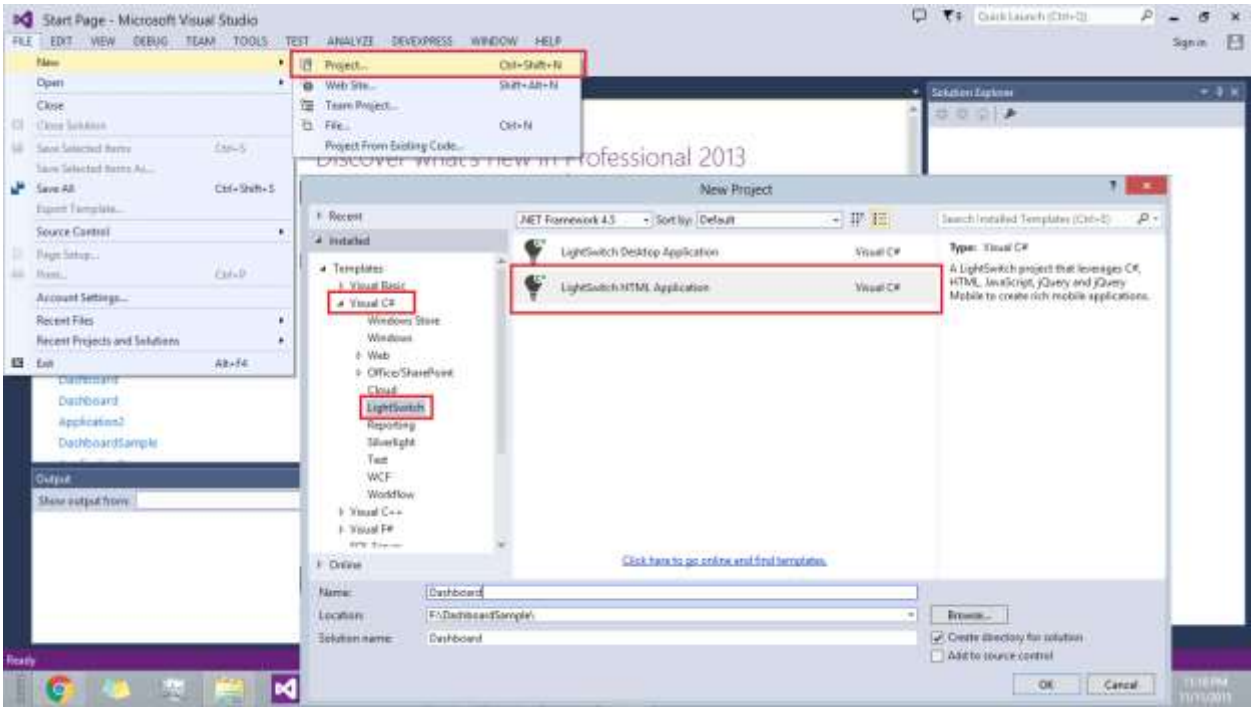


Getting Started with LightSwitch HTML Application

This section describes how to create an LightSwitch HTML application with embedded dashboard viewer.

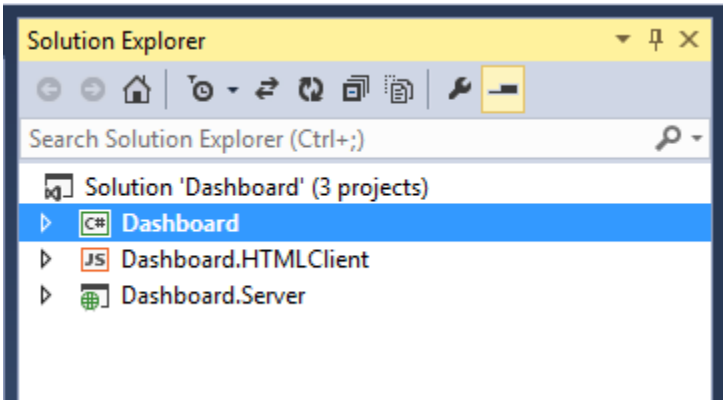
Creating LightSwitch HTML Project

Open Visual Studio IDE and navigate to File->New->Project, click on LightSwitch and select LightSwitch HTML Application and give a name (say Dashboard) and click OK.



Note: You must have the LightSwitch HTML Client extension installed if you are not using Microsoft Visual Studio 2012 Update 2 or later version. You must have Office Developer Tools for Visual Studio 2015 installed, if you are using Microsoft Visual Studio 2015.

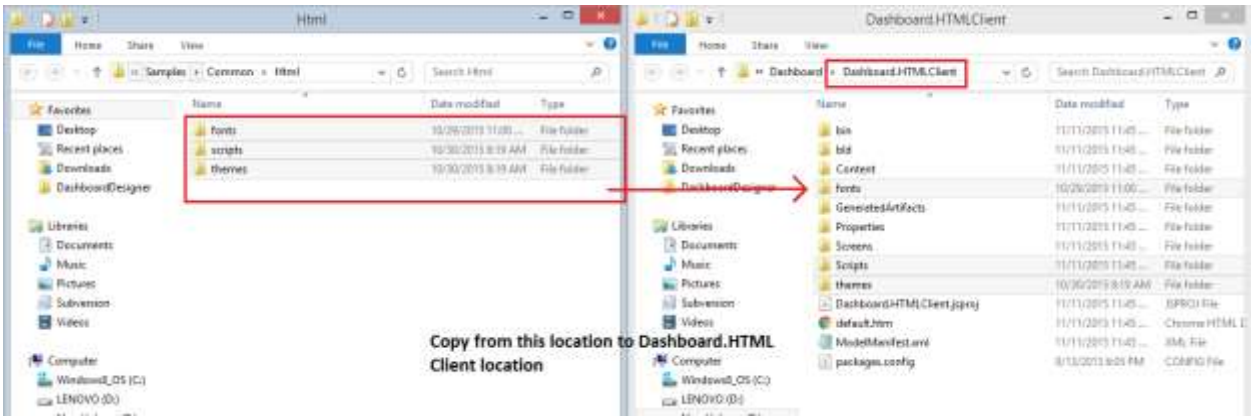
Now you will find a solution with three projects created by default in Solution Explorer View.



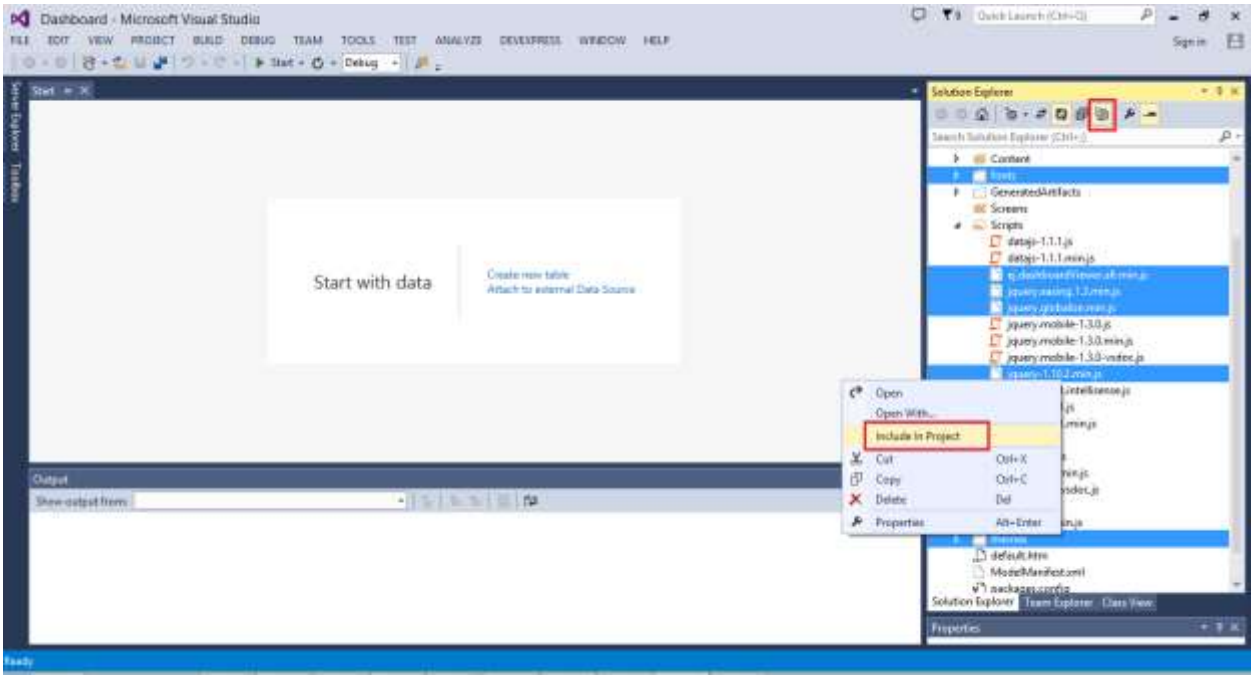
Adding references to HTML Client Project

Right click on Dashboard.HTMLClient project and click on Open Folder in File Explorer, and copy the content from this location into HTMLClient folder.

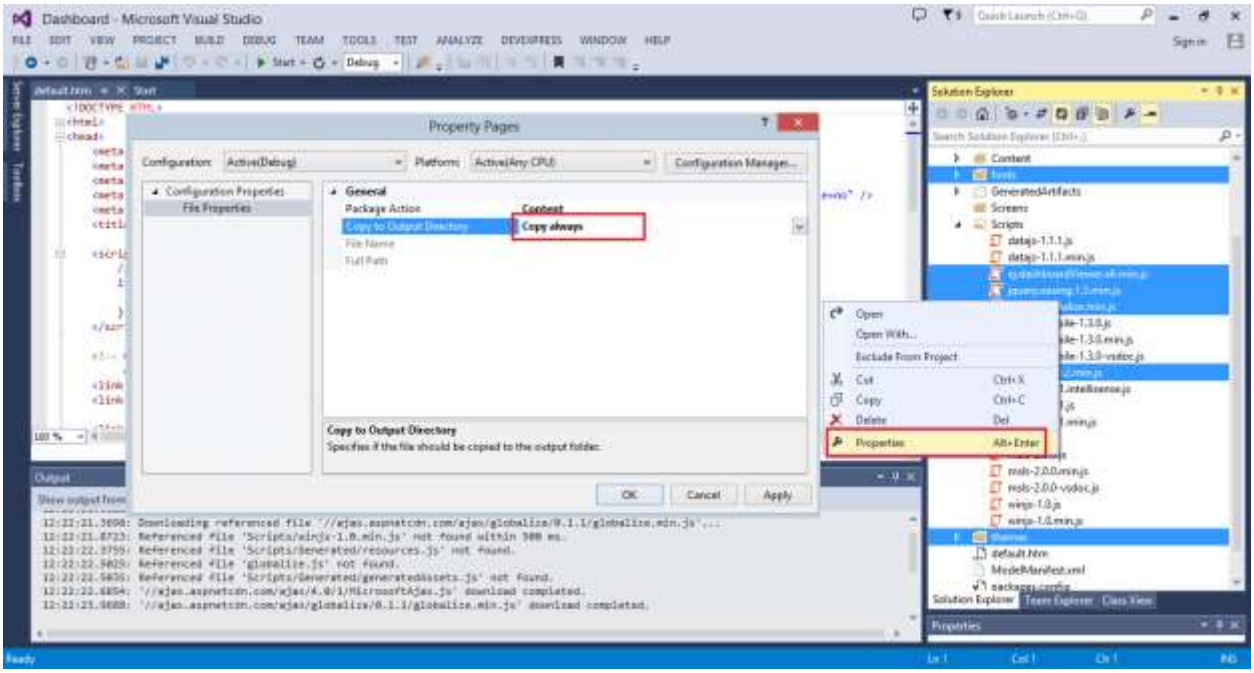
%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html



In Solution Explorer, click Show All Files icon if not checked, select those newly added folder and files, right click and select Include in Project.



Set the Build Action property to Content and the Copy to Output Directory property to Copy always as shown in the following image for all the files added to the project.



Adding scripts and links to HTML Client Project

Open default.htm in Dashboard.HTMLClient project.

Copy the below link references above </head> tag.

ASPX-CS

```
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet" />
```

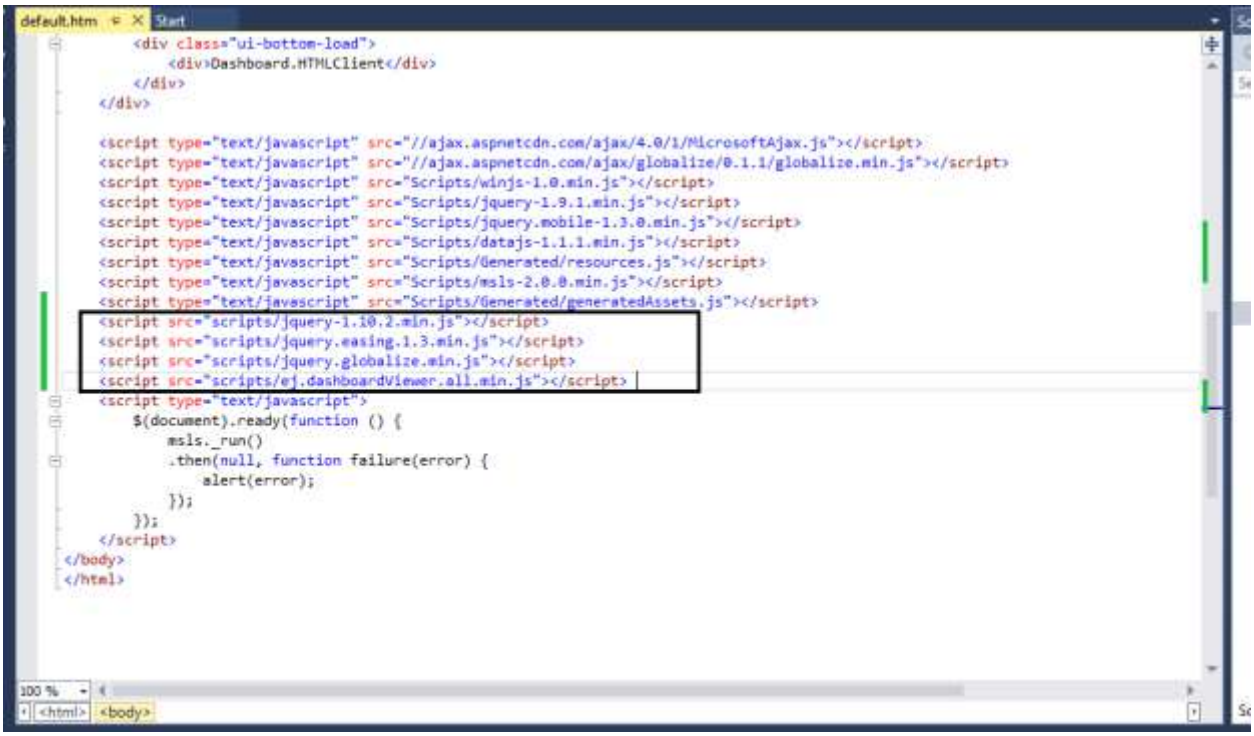

Now, the file will look like below.

Copy the below script references above <script type="text/javascript"> tag.

ASPX-CS

```
<script src="scripts/jquery-1.10.2.min.js"></script>
<script src="scripts/jquery.easing.1.3.min.js"></script>
<script src="scripts/ej.dashboardViewer.all.min.js"></script>
```

Now, the file will look like below.

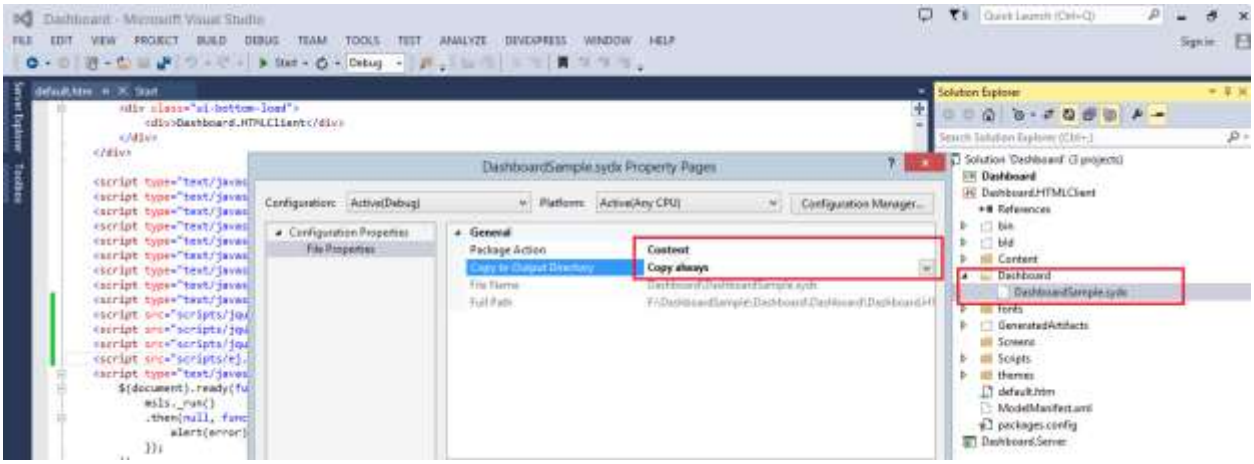


Adding a dashboard to HTML Client Project

Create a new folder under HTMLClient Project, right click and select New Folder (name it as Dashboard) and place the dashboard SYDX file inside this folder (name of dashboard file is DashboardSample.sydx).

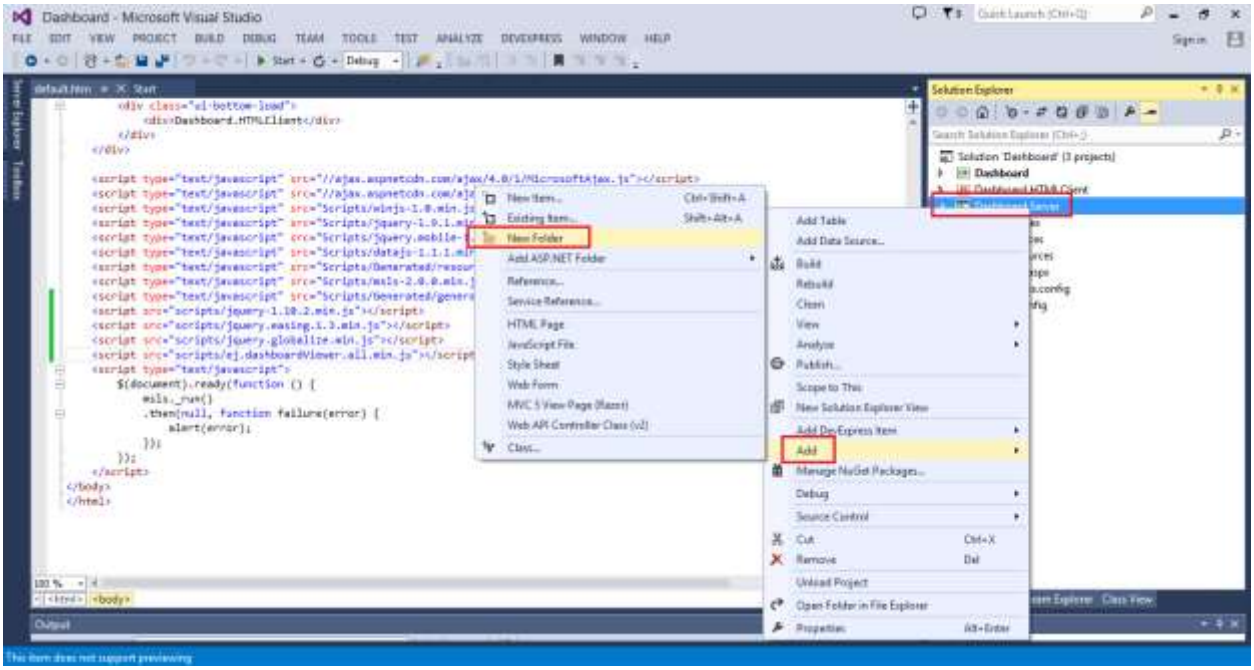
Right click on the SYDX file and select properties,

Set the Package Action to Content and set the Copy to Output Directory to Copy Always.



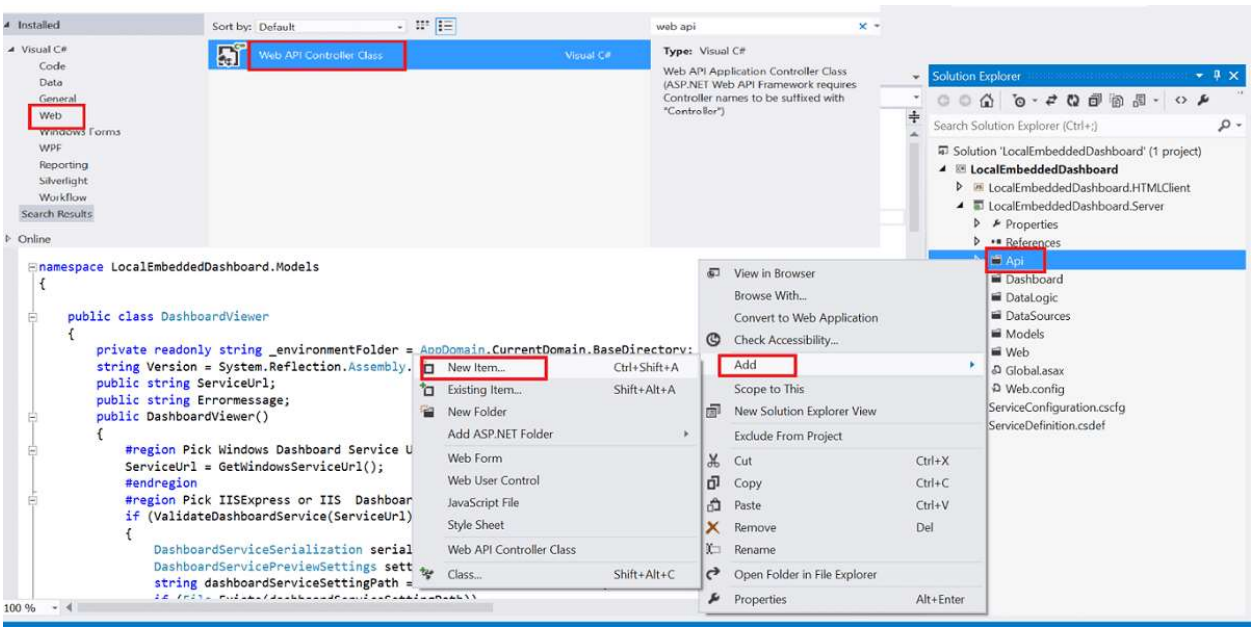
Creating controllers for handling path and service URL under Server project

Create a new folder (named it as API) under Dashboard.Server project.



Create a Web API class inside API folder

Right click on the newly created API folder Add->New Item, navigate to Web ->WebAPI and click on Web API controller class (v2), (name it as DashboardPathController.cs).



Replace all the codes present in this class with the below one.

C#

```
public class DashboardPathController : ApiController
{
// Get API/<controller>
public Dictionary<string, string> Get()
{
Dictionary<string, string> reportDetails = new Dictionary<string, string>();
DashboardWindowsServiceInfo dashboardViewer = new
DashboardWindowsServiceInfo();
string path = AppDomain.CurrentDomain.BaseDirectory.Substring(0,
AppDomain.CurrentDomain.BaseDirectory.LastIndexOf("Bin")) +
"LocalEmbeddedDashboard.Server\\bin\\Debug\\Dashboard\\WorldWideCarSalesDash
board.sydx";
reportDetails.Add("ReportPath", path);
reportDetails.Add("ServiceURL", dashboardViewer.ServiceUrl);
reportDetails.Add("ErrorMessage", dashboardViewer.ErrorMessage);
return reportDetails;
}
}
```

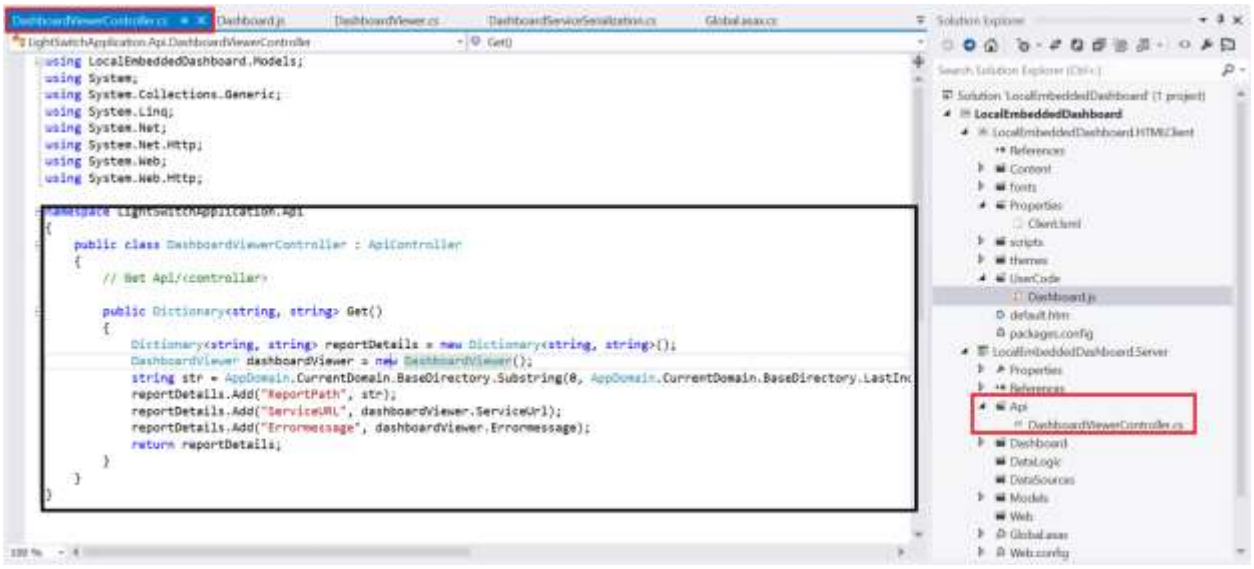
VB.NET

```
Public Class DashboardPathController
Inherits ApiController
' Get api/<controller>
Public Function [Get]() As Dictionary(Of String, String)
Dim reportDetails As New Dictionary(Of String, String)()
Dim dashboardViewer As New DashboardWindowsServiceInfo()
Dim path As String = AppDomain.CurrentDomain.BaseDirectory.Substring(0,
AppDomain.CurrentDomain.BaseDirectory.LastIndexOf("Bin")) &
"LocalEmbeddedDashboard.Server\bin\Debug\Dashboard\WorldWideCarSalesDashboar
d.sydx"
reportDetails.Add("ReportPath", path)
```



```
reportDetails.Add("ServiceURL", dashboardViewer.ServiceUrl)
reportDetails.Add("ErrorMessage", dashboardViewer.ErrorMessage)
Return reportDetails
End Function
End Class
```

Now, the file will look like below.



Creating model

Create a new folder (named it as Models) under Dashboard.Server project.

Create a class DashboardWindowsServiceInfo inside the Models folder with the following code.

C#

```
public class DashboardWindowsServiceInfo
{
private readonly string _environmentFolder =
AppDomain.CurrentDomain.BaseDirectory;
string Version =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
();
public string ServiceUrl;
public string ErrorMessage;
public DashboardWindowsServiceInfo()
{
#region Pick Windows Dashboard Service Url
ServiceUrl = GetWindowsServiceUrl();
#endregion
#region Pick IISExpress or IIS  Dashboard Service URL if Dashboard Windows
Service is not running
if (ValidateDashboardService(ServiceUrl))
{
DashboardServiceSerialization serializer = new
DashboardServiceSerialization();
DashboardServicePreviewSettings settings = new
DashboardServicePreviewSettings();
```



```
string dashboardServiceSettingPath =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) +
@"\Syncfusion\Dashboard Platform SDK\" + Version +
@"\DashboardServiceSetting.xml";
if (File.Exists(dashboardServiceSettingPath))
{
settings = serializer.Deserialize(dashboardServiceSettingPath);
if (!ValidateDashboardService(settings.ServiceURL))
ServiceUrl = settings.ServiceURL;
else
{
ServiceUrl = string.Empty;
ErrorMessage = "Syncfusion Dashboard Service is not running. Please start
the SyncfusionDashboardServiceInstaller-IISExpress.exe file  to start the
service.";
}
}
else
{
ErrorMessage = "Syncfusion Dashboard Service is not running. Please start
the SyncfusionDashboardServiceInstaller-IISExpress.exe file  to start the
service.";
ServiceUrl = string.Empty;
}
}
#endregion
}
/// <summary>
/// Validate whether Dashboard Service is running in the URL
/// </summary>
/// <param name="dashboardServiceUrl">Dashboard Service URL</param>
/// <returns>returns whether valid dashboard service</returns>
private static bool ValidateDashboardService(string dashboardServiceUrl)
{
bool errorOccurred = false;
try
{
if (string.IsNullOrWhiteSpace(dashboardServiceUrl))
{
return true;
}
if (!dashboardServiceUrl.Contains("http://") &&
!dashboardServiceUrl.Contains("https://"))
dashboardServiceUrl = "http://" + dashboardServiceUrl + @"/IsServiceExists";
else
dashboardServiceUrl = dashboardServiceUrl + @"/IsServiceExists";
WebRequest request = WebRequest.Create(new Uri(dashboardServiceUrl,
UriKind.Absolute));
request.Method = "GET";
using (WebResponse response = request.GetResponse())
{
using (StreamReader reader = new StreamReader(response.GetResponseStream()))
{
string text = reader.ReadToEnd();
if
(!text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetB
ytes("DashboardServiceExists"))))
```

```
{
errorOccurred = true;
}
}
}
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
}
catch (Exception e)
{
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
errorOccurred = true;
}
return errorOccurred;
}
/// <summary>
/// Used to pick the Dashboard Windows Service URL
/// </summary>
/// <returns>Service URL of Dashboard Windows Service</returns>
private string GetWindowsServiceUrl()
{
string url = string.Empty;
try
{
RegistryKey key =
Registry.LocalMachine.OpenSubKey(@"Software\SyncfusionDashboard\Syncfusion
Dashboard Service");
if (key == null)
key =
Registry.LocalMachine.OpenSubKey(@"Software\Wow6432Node\SyncfusionDashboard\
Syncfusion Dashboard Service");
if (key != null)
{
url = (string)key.GetValue("ServiceURL");
key.Close();
}
}
catch (Exception)
{
}
return url;
}
}
```

VB.NET

```
Public Class DashboardWindowsServiceInfo
Private ReadOnly _environmentFolder As String =
AppDomain.CurrentDomain.BaseDirectory
Private Version As String =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
()
Public ServiceUrl As String
Public ErrorMessage As String
Public Sub New()
'                       #Region "Pick Windows Dashboard Service Url"
ServiceUrl = GetWindowsServiceUrl()
```

```
'                       #End Region
'                       #Region "Pick IISExpress or IIS  Dashboard Service
URL if Dashboard Windows Service is not running"
If ValidateDashboardService(ServiceUrl) Then
Dim serializer As New DashboardServiceSerialization()
Dim settings As New DashboardServicePreviewSettings()
Dim dashboardServiceSettingPath As String =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) &
"\Syncfusion\Dashboard Platform SDK\" & Version &
"\DashboardServiceSetting.xml"
If File.Exists(dashboardServiceSettingPath) Then
settings = serializer.Deserialize(dashboardServiceSettingPath)
If Not ValidateDashboardService(settings.ServiceURL) Then
ServiceUrl = settings.ServiceURL
Else
ServiceUrl = String.Empty
ErrorMessage = "Syncfusion Dashboard Service is not running. Please start
the SyncfusionDashboardServiceInstaller-IISExpress.exe file  to start the
service."
End If
Else
ErrorMessage = "Syncfusion Dashboard Service is not running. Please start
the SyncfusionDashboardServiceInstaller-IISExpress.exe file  to start the
service."
ServiceUrl = String.Empty
End If
End If
'                       #End Region
End Sub
''' <summary>
''' Validate whether Dashboard Service is running in the URL
''' </summary>
''' <param name="dashboardServiceUrl">Dashboard Service URL</param>
''' <returns>returns whether valid dashboard service</returns>
Private Shared Function ValidateDashboardService(ByVal dashboardServiceUrl
As String) As Boolean
Dim errorOccurred As Boolean = False
Try
If String.IsNullOrWhiteSpace(dashboardServiceUrl) Then
Return True
End If
If Not dashboardServiceUrl.Contains("http://") AndAlso Not
dashboardServiceUrl.Contains("https://") Then
dashboardServiceUrl = "http://" & dashboardServiceUrl & "/IsServiceExists"
Else
dashboardServiceUrl = dashboardServiceUrl & "/IsServiceExists"
End If
Dim request As WebRequest = WebRequest.Create(New Uri(dashboardServiceUrl,
UriKind.Absolute))
request.Method = "GET"
Using response As WebResponse = request.GetResponse()
Using reader As New StreamReader(response.GetResponseStream())
Dim text As String = reader.ReadToEnd()
If Not
text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetByt
es("DashboardServiceExists"))) Then
errorOccurred = True
```

```
End If
End Using
End Using
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
Catch e As Exception
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
errorOccurred = True
End Try
Return errorOccurred
End Function
''' <summary>
''' Used to pick the Dashboard Windows Service URL
''' </summary>
''' <returns>Service URL of Dashboard Windows Service</returns>
Private Function GetWindowsServiceUrl() As String
Dim url As String = String.Empty
Try
Dim key As RegistryKey =
Registry.LocalMachine.OpenSubKey("Software\SyncfusionDashboard\Syncfusion
Dashboard Service")
If key Is Nothing Then
key =
Registry.LocalMachine.OpenSubKey("Software\Wow6432Node\SyncfusionDashboard\S
yncfusion Dashboard Service")
End If
If key IsNot Nothing Then
url = DirectCast(key.GetValue("ServiceURL"), String)
key.Close()
End If
Catch e1 As Exception
End Try
Return url
End Function
End Class
```

Now the file look like below

Configuring Route

Right click Dashboard.Server project Add->New Item and navigate to Web-> General, select Global Application Class.

Add the following code under Application_Start method:

C#

```
RouteTable.Routes.MapHttpRoute(
name: "DefaultApi",
routeTemplate: "Api/{controller}/{id}",
defaults: new { id = System.Web.Http.RouteParameter.Optional }
);
```

VB.NET

```
RouteTable.Routes.MapHttpRoute(
name:="DefaultApi",
routeTemplate:="Api/{controller}/{id}",
defaults:=New With {Key .id = System.Web.Http.RouteParameter.Optional})
```

Add the following namespace declarations.

C#

```
using System.Web.Routing;
using System.Web.Http;
```

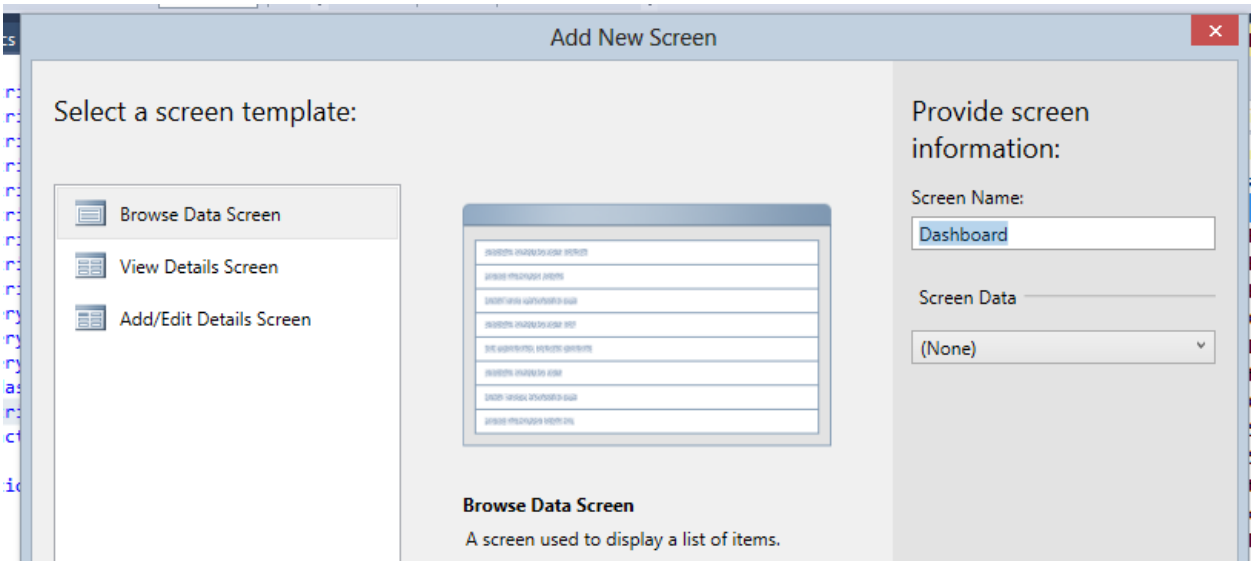
VB.NET

```
Imports System.Web.Routing
Imports System.Web.Http
```

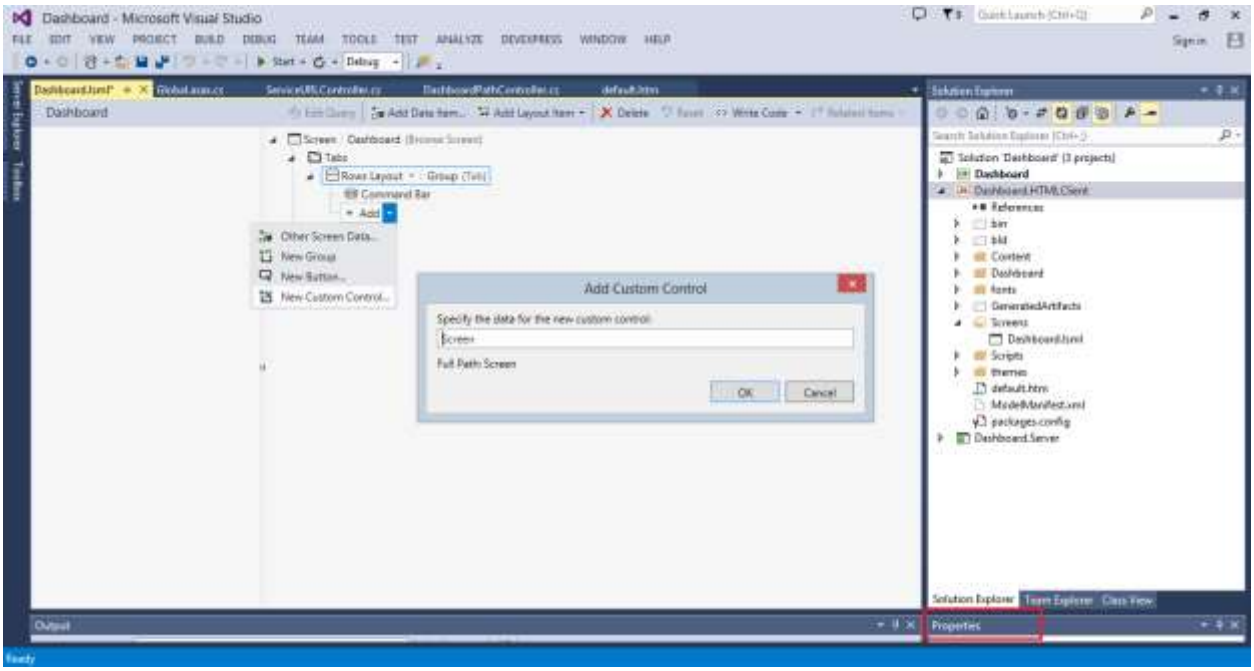
Now, the file will look like below.

Creating a Browser Screen and Custom Control in HTML Client Project

Right click on Dashboard.HTMLClient project and select Add Screen, Select Browser Data Screen, type the screen name as Dashboard and click OK.



Add a custom control through Add -> New Custom Control menu item.



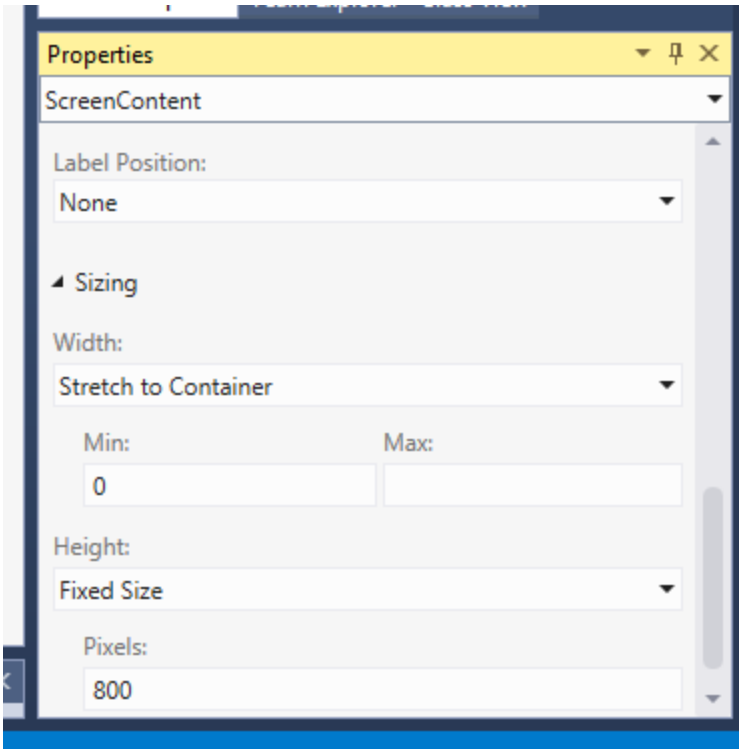
Handle the following settings in Properties window:

Label position: None

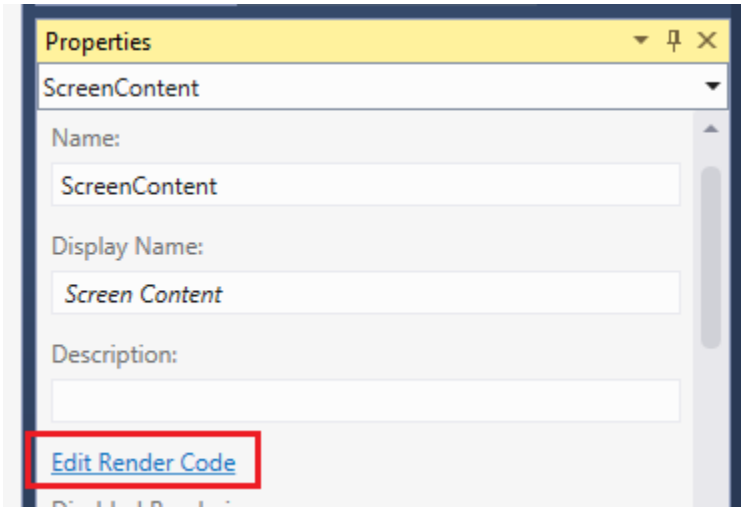
Width: Stretch to container

Height: Fixed Size

Pixels: 800



In Properties window, click on Edit Render Code, to open a JS file named Dashboard.js.



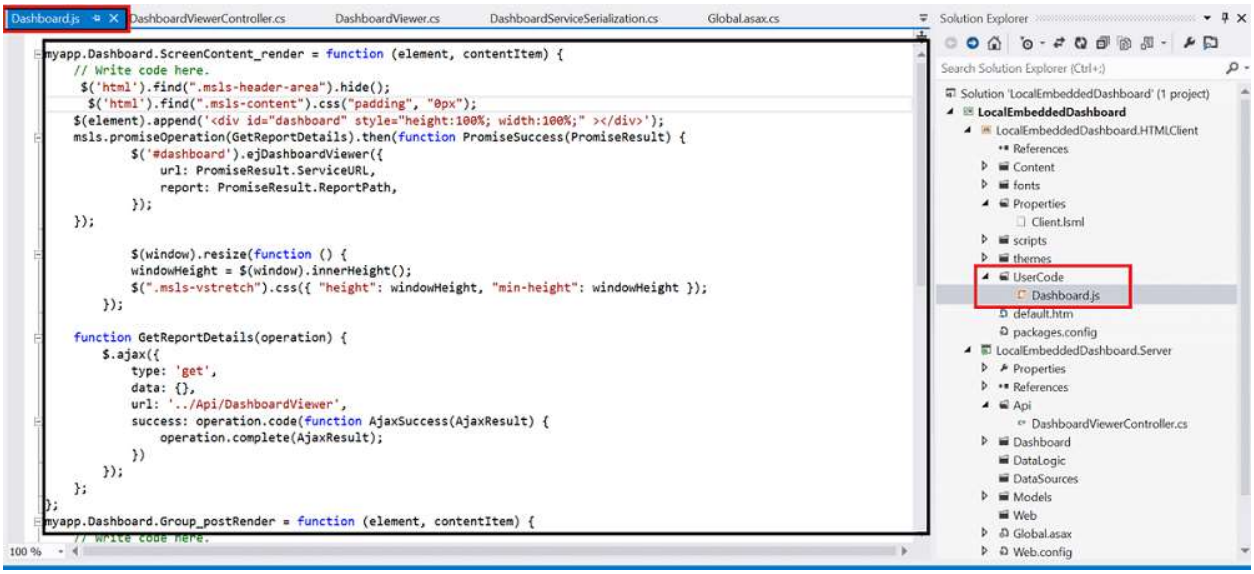
Add the following code lines in the JS file:

ASPX-CS

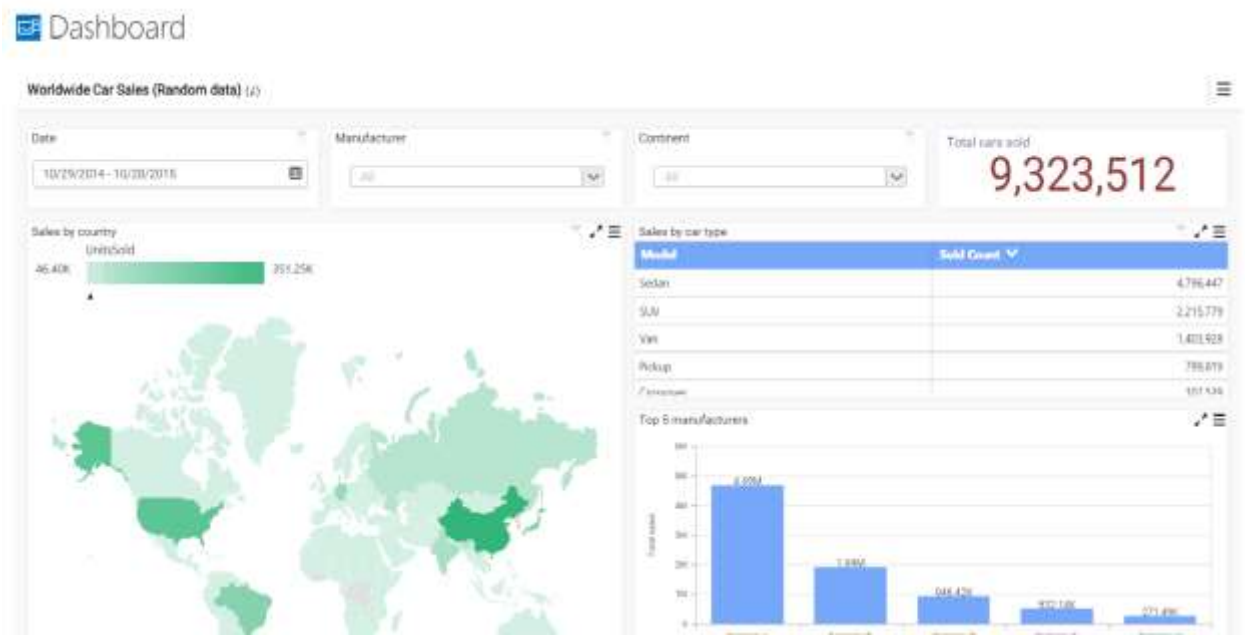
```
myapp.Dashboard.ScreenContent_render = function (element, contentItem) {
// Write code here.
$('html').find(".msls-header-area").hide();
$('html').find(".msls-content").css("padding", "0px");
$(element).append('<div id="dashboard" style="height:100%; width:100%;"
></div>');
msls.promiseOperation(GetReportDetails).then(function
PromiseSuccess(PromiseResult) {
$('#dashboard').ejDashboardViewer({
```



```
url: PromiseResult.ServiceURL,
report: PromiseResult.ReportPath,
});
});
$(window).resize(function () {
windowHeight = $(window).innerHeight();
$(".msls-vstretch").css({ "height": windowHeight, "min-height": windowHeight
});
});
function GetReportDetails(operation) {
$.ajax({
type: 'get',
data: {},
url: '../Api/DashboardViewer',
success: operation.code(function AjaxSuccess(AjaxResult) {
operation.complete(AjaxResult);
})
});
};
};
myapp.Dashboard.Group_postRender = function (element, contentItem) {
// Write code here.
};
```



Click on Start and dashboard will load in default browser.



Getting Started with PHP Application

This section describes how to create a PHP Application with an embedded dashboard viewer through a PHP wrapper.

System Requirements

- Download PHP 5.3.3+ version to your machine.
- Extract the downloaded file content and move to the desired drive.

Creating a Simple PHP Application with PHP Wrapper for Dashboard Viewer

Create a new folder, say PHP.

Copy the fonts, scripts, themes from Dashboard Platform SDK build installed location mentioned below.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html

copy the EJ folder from Dashboard Platform SDK build installed location mentioned below.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\PHP

Paste them under the newly created PHP folder.

Create a new PHP file, say index.php and place it under the PHP folder.

Add the required script and theme references in this file, and refer AutoLoad.php file from the EJ folder as shown below.

HTML

```
<!DOCTYPE html>
<html>
<head>
<!--stylesheet reference-->
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet">
```



```
<!--script reference-->
<script src="scripts/jquery-1.10.2.min.js" ></script>
<script src="scripts/jquery.easing.1.3.min.js" ></script>
<script src="scripts/ej.dashboardViewer.all.min.js" ></script>
</head>
<body>
<!--Refer AutoLoad.php common source to render the DashboardViewer-->
<?php require_once '../EJ/AutoLoad.php'; ?>
<!--DashboardViewer Initialization-->
</body>
</html>
```

DashboardViewer Initialization

Add the following code in the index.php file to initialize the Dashboard Viewer.

HTML

```
<?php
$dashboardviewer= new EJ\DashboardViewer ("dashboard");
$size= new EJ\DashboardViewer\Size ();
$size->width ("100 %")->height ("100 %");
echo $dashboardviewer-
>serviceUrl("https://dashboardsdk.syncfusion.com/DashboardService/DashboardS
ervice.svc")-
>dashboardPath("https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCa
rSalesDashboard.sydx")->size($size)->render();
?>
```

Note: Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

Provided online Dashboard Service URL above, only for demo purpose.

Finally, the index.php page looks like below:

HTML

```
<!DOCTYPE html>
<html>
<head>
<title>PHP - DashboardViewer</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0"
charset="utf-8" />
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet">
<script src="scripts/jquery-1.10.2.min.js" ></script>
<script src="scripts/jquery.easing.1.3.min.js" ></script>
<script src="scripts/ej.dashboardViewer.all.min.js" ></script>
<style>
body,html,#dashboard {
overflow: hidden !important;
height: 100%;
width: 100%;
}
</style>
</head>
<body>
```



```
<?php
$dashboardviewer= new EJ\DashboardViewer ("dashboard");
$size= new EJ\DashboardViewer\Size ();
$size->width ("100 %")->height ("100 %");
echo $dashboardviewer-
>serviceUrl("https://dashboardsdk.syncfusion.com/DashboardService/DashboardS
ervice.svc")-
>dashboardPath("https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCa
rSalesDashboard.sydx")->size($size)->render();
?>
</body>
</html>
```

Note: Provided online Dashboard Service URL and Dashboard Path are for demo purpose.

Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

Binding Dashboard Service

To initiate the dashboard service instance you can follow anyone of the below methods

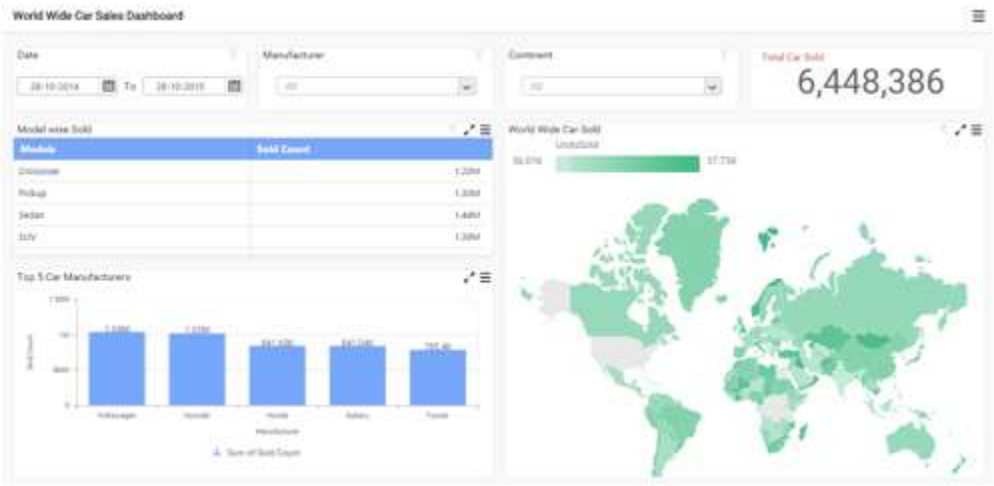
1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Running the Application

To run the application, kindly refer the ReadMe.html file deployed in the following location on Dashboard Platform SDK installation.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\PHP



Getting Started

JSP

System Requirements

To work with JSP, you need to meet the below configuration.

- Java Development Kit (JDK), Java Runtime Environment (JRE) version (1.7 or 1.8) needs to be installed and configured.
- Any web servers (Apache Tomcat 7) that's supports the Java Servlet, JSP specification. Also, ensure that the web servers need to be tested and configured with the IDE.
- Eclipse IDE for Enterprise developers with J2EE support – (Lune, Kepler, Mars, or later) or NetBeans IDE with Java2E support.

Steps to deploy the DashboardViewer JSP demo in Apache Tomcat

1. Download the Apache Tomcat from the official download page. 2. You can find a DashboardViewer JSP sample JSP from [Dashboard Platform SDK] (https://help.syncfusion.com/dashboard-platform/dashboard-sdk/installation-and-deployment) build installed location mentioned below.

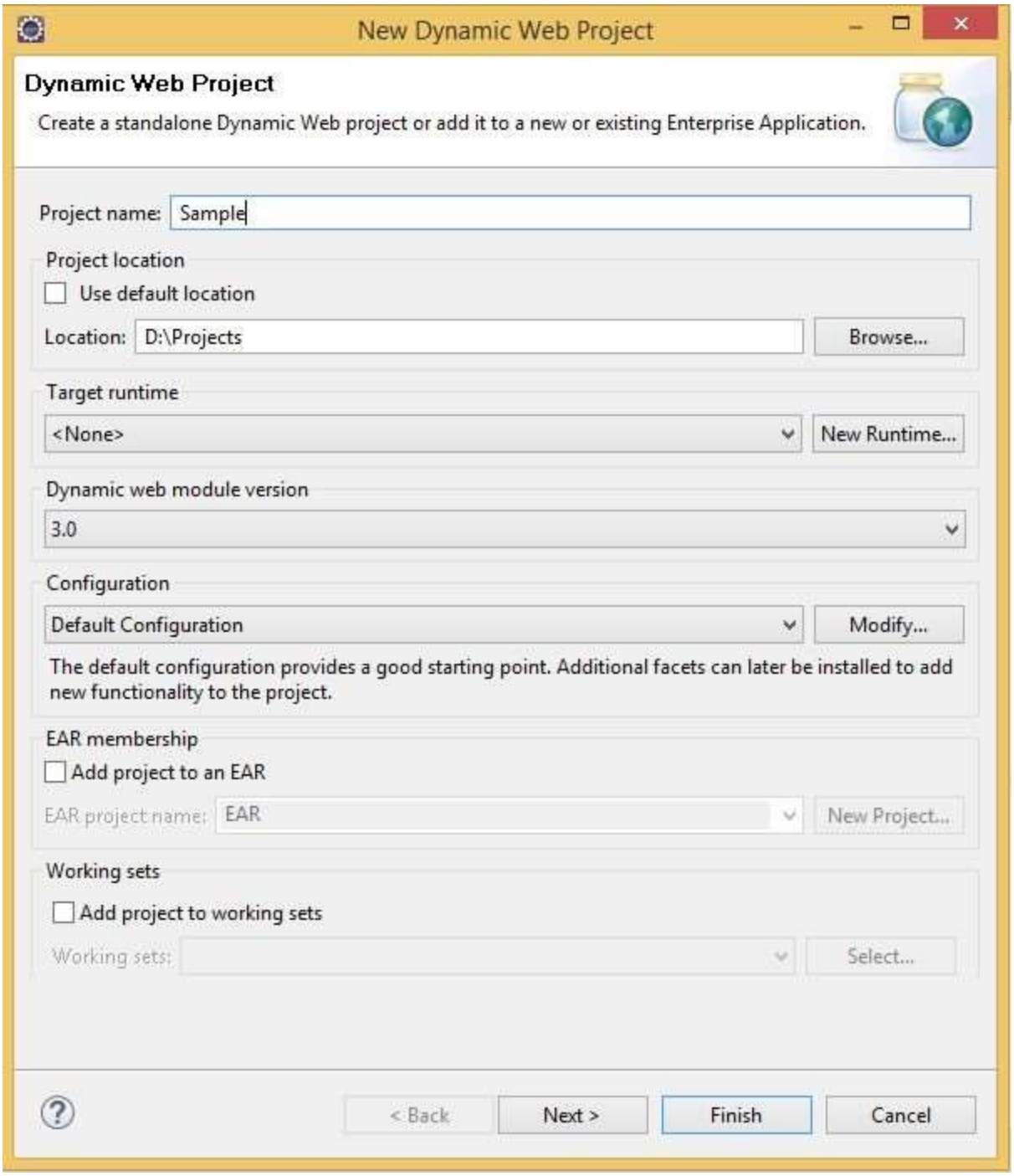
`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JSP\DashboardViewer`

3. Copy the DashboardViewer folder to your Webapps folder of Apache Tomcat. 4. Browse to http://localhost:8080/DashboardViewer to see the Dashboard.

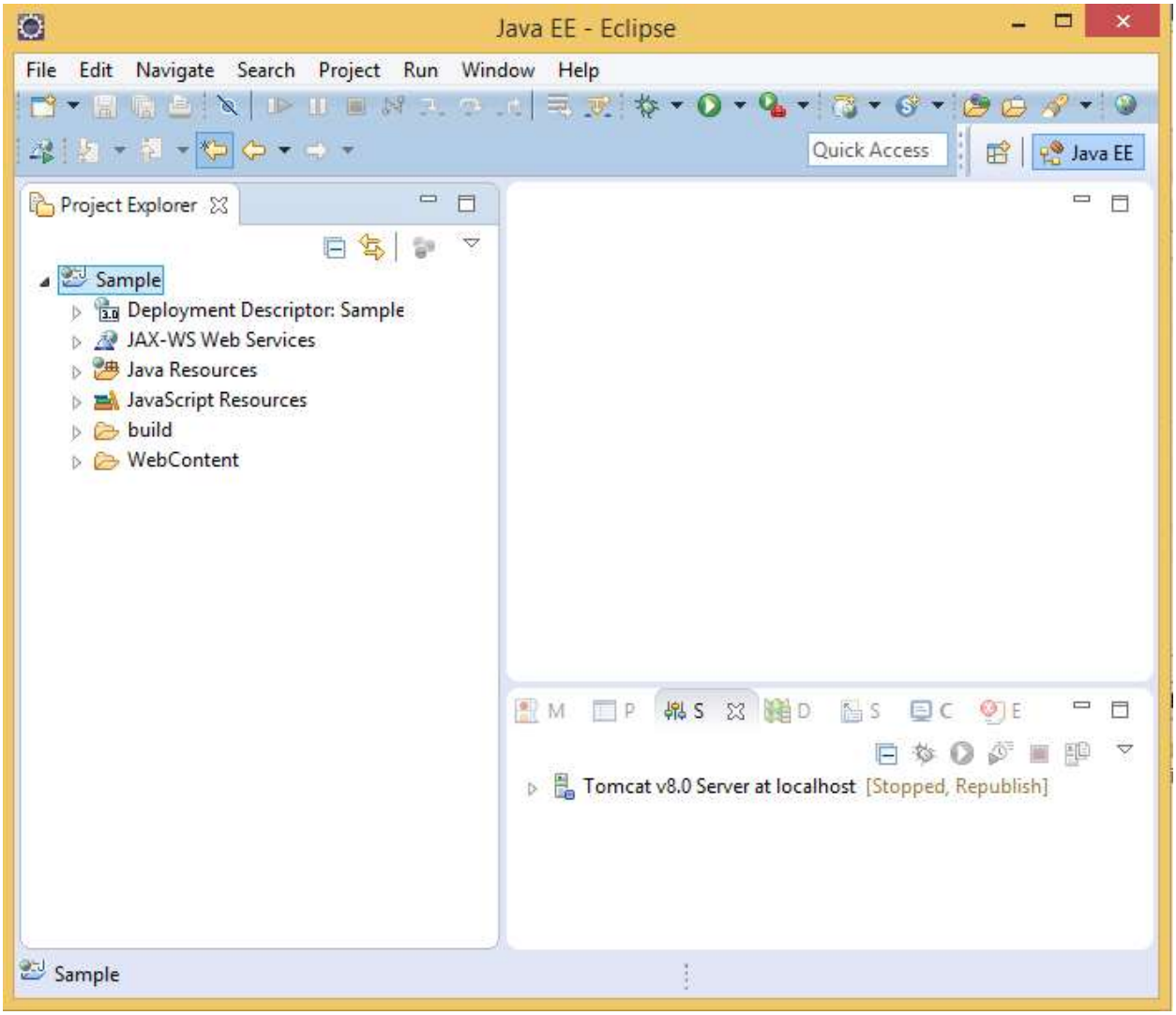
Create a DashboardViewer JSP Dynamic web application with Eclipse environment

To use DashboardViewer JSP in eclipse environment, follow the steps below:

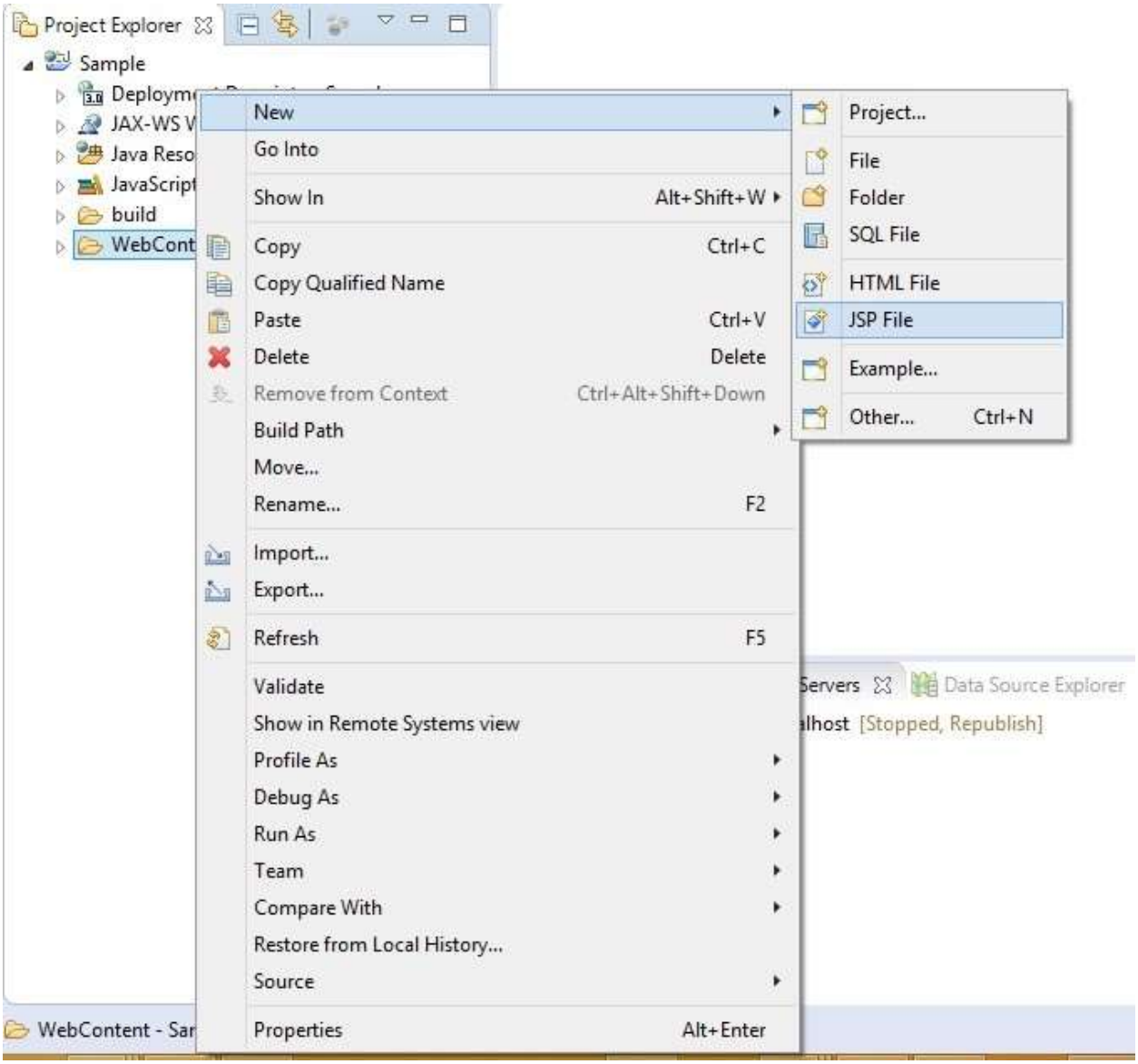
- To create the Dynamic web project from File -> New -> Dynamic web project.
- In this Dynamic Web project window specify the project name and click the Next button to create the project.



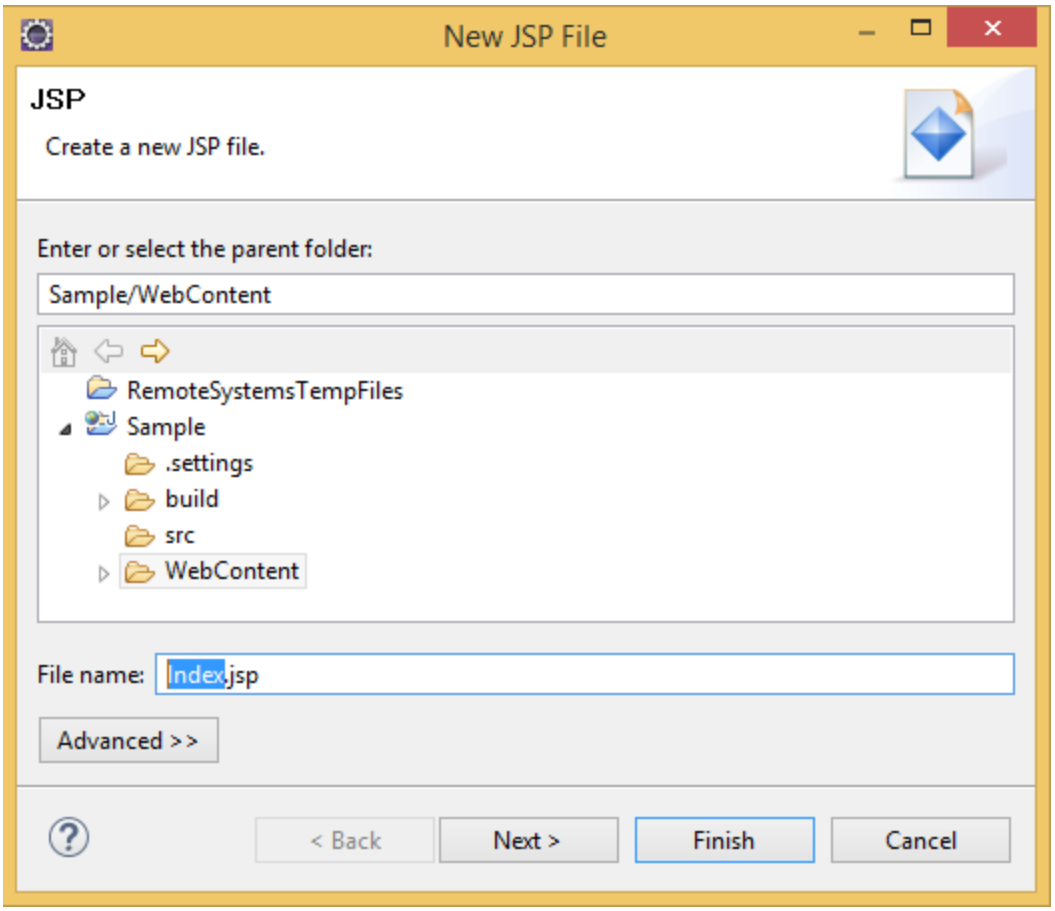
- Once the project has been created successfully, the project file showcased in the left side project Explorer pane in the IDE.



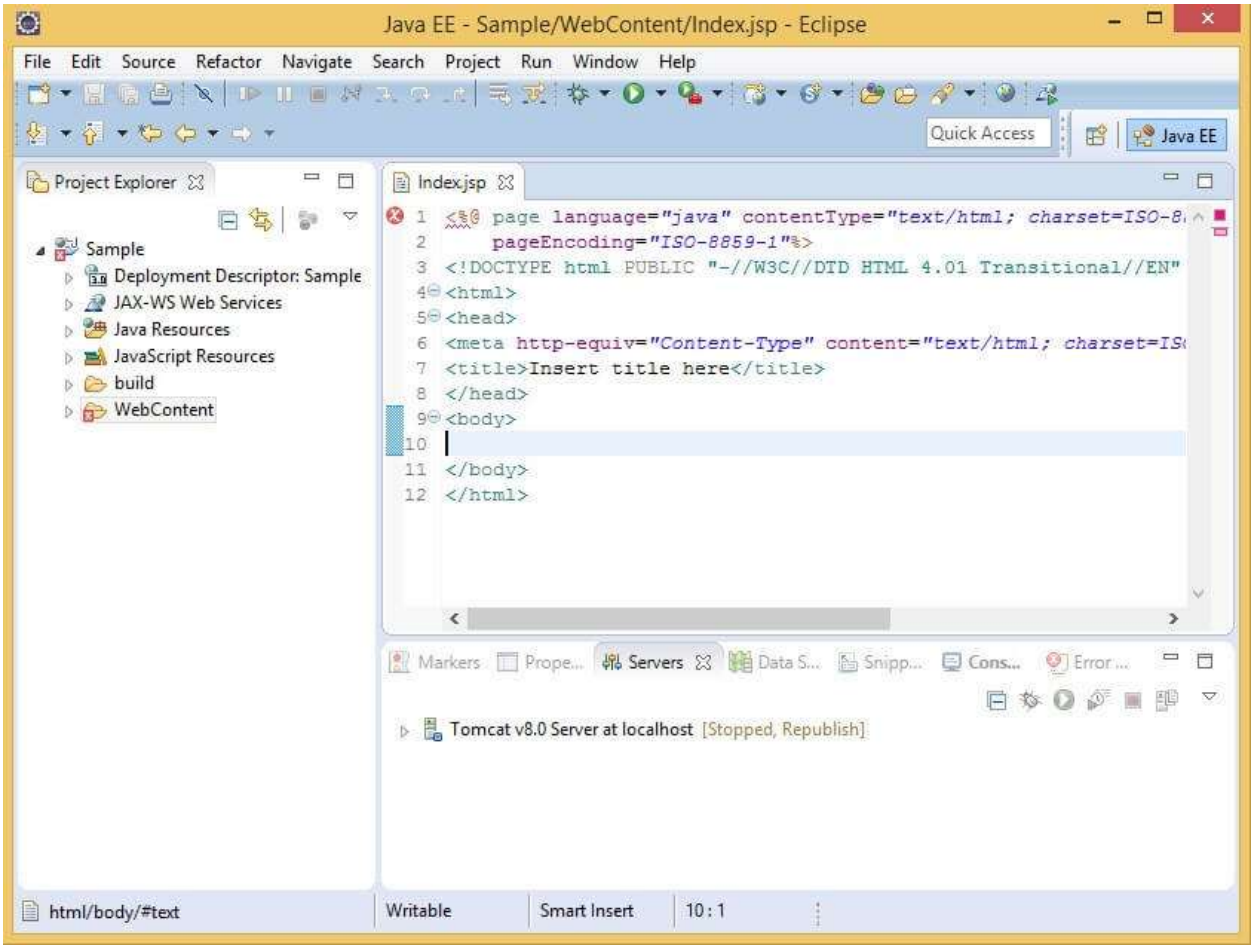
- Then add the new JSP file to the project by right click the WebContent folder in your project and choose New -> JSP.



- Specify the file name for your new JSP file.



- Now the JSP file created with default template content.



- Copy the scripts, themes and fonts folders from Dashboard Platform SDK build installed location

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JSP\DashboardViewer

and paste into your project WebContent folder. Then, add the scripts and CSS references in the order mentioned below.

Example

HTML

```
<html>
<head>
<title> Getting Started DashboardViewer JSP </style>
<link href="Content/themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet" />
<script type="text/javascript" src="scripts/jquery-1.10.2.min.js"></script>
<script type="text/javascript"
src="scripts/jquery.easing.1.3.min.js"></script>
<script type="text/javascript"
src="scripts/ej.dashboardViewer.all.min.js"></script>
<style>
#dashboard, html, body{
```

```
height:100%;
width:100%;
overflow:hidden !important;
}
</style>
</head>
<body>
< !-- DashboardViewer JSP Custom Tag -->
</body>
</html>
```

1. Now add the DashboardViewer JSP source package from \JSP\DashboardViewer\WEB-INF\lib\syncfusion.dashboardviewer.jar to your project's /WebContent/WEB-INF/lib folder.
2. Add the Custom taglib from \JSP\DashboardViewer\WEB-INF\EJ.tld to your project's /WebContent/WEB-INF.
3. Import the DashboardViewer component package into your JSP page.

Example

HTML

```
<%@ page import="com.syncfusion.*"%>
```

4. Add the mapping Tag Library descriptor (tld) file to support Dashboard Platform SDK DashboardViewer JSP custom tag.

Example

HTML

```
<%@ taglib prefix="ej" uri="/WEB-INF/EJ.tld"%>
```

5. Use the Dashboard Platform SDK DashboardViewer JSP custom tag within your JSP file.

Example

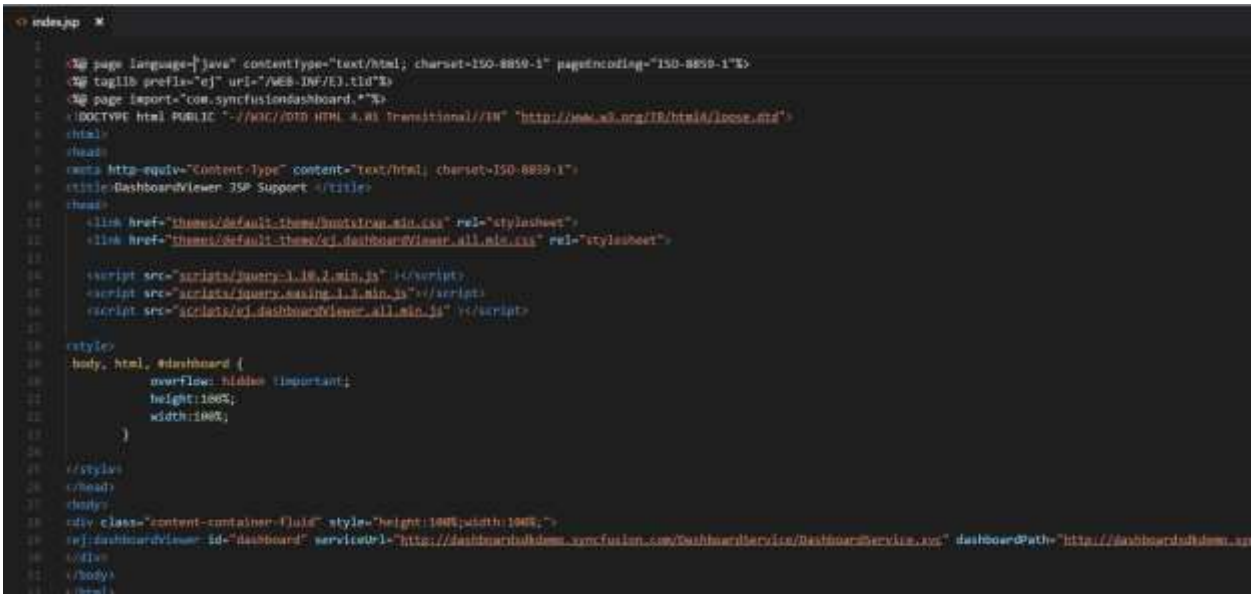
HTML

```
<div class="content-container-fluid" style="height:100%; width:100%;">
<ej:dashboardViewer id="dashboard"
serviceUrl="https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc"
dashboardPath="https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDashboard.sydx"></ej:dashboardViewer>
</div>
```

Note: Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

Provided online Dashboard Service URL above, only for demo purpose.

6. Finally, the index.jsp file looks like below:



Binding Dashboard Service

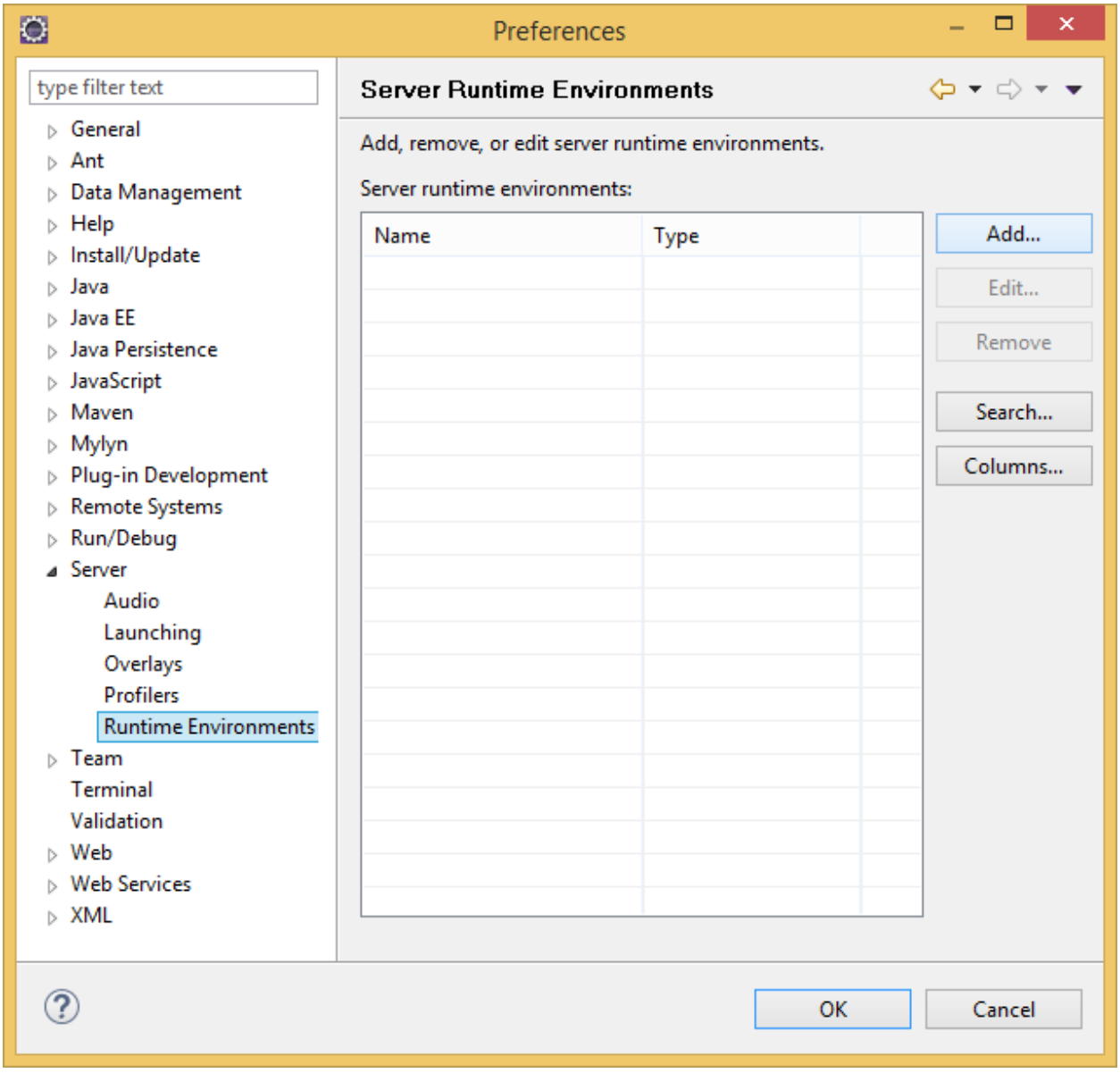
To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

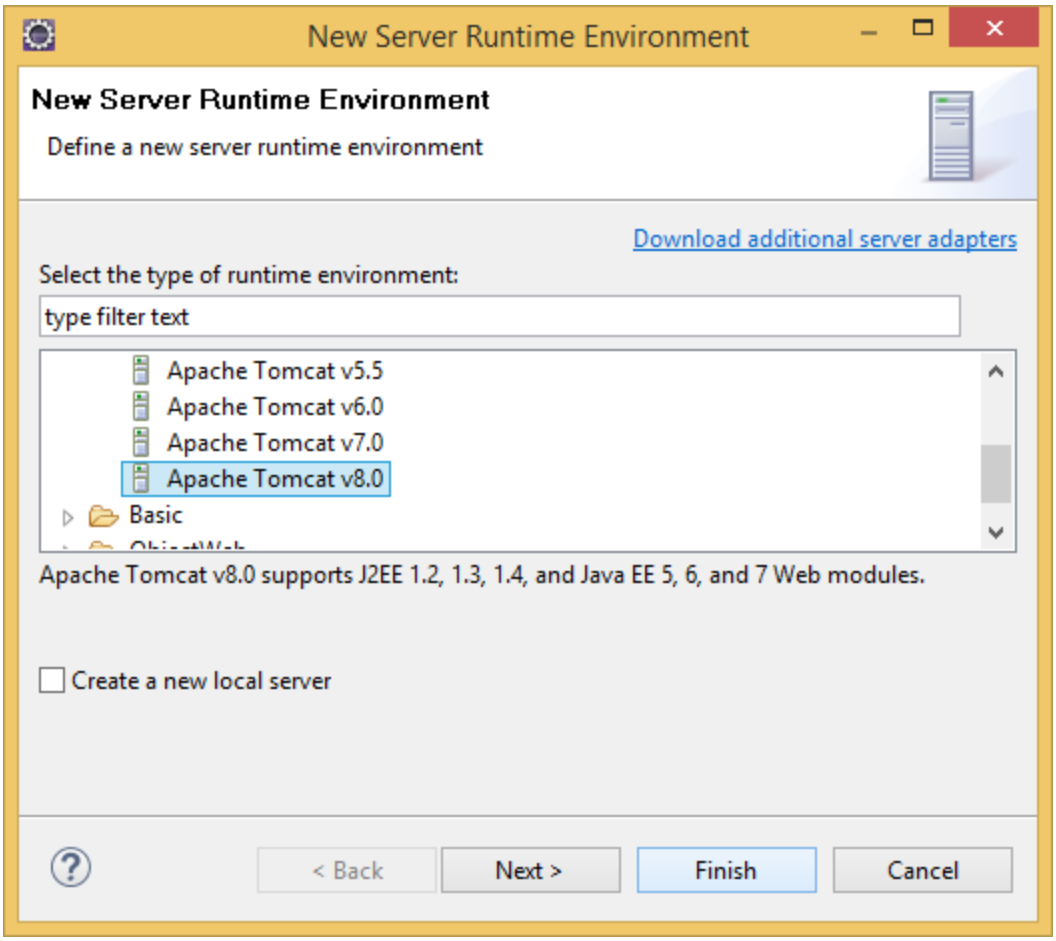
Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Start the server to deploy our project

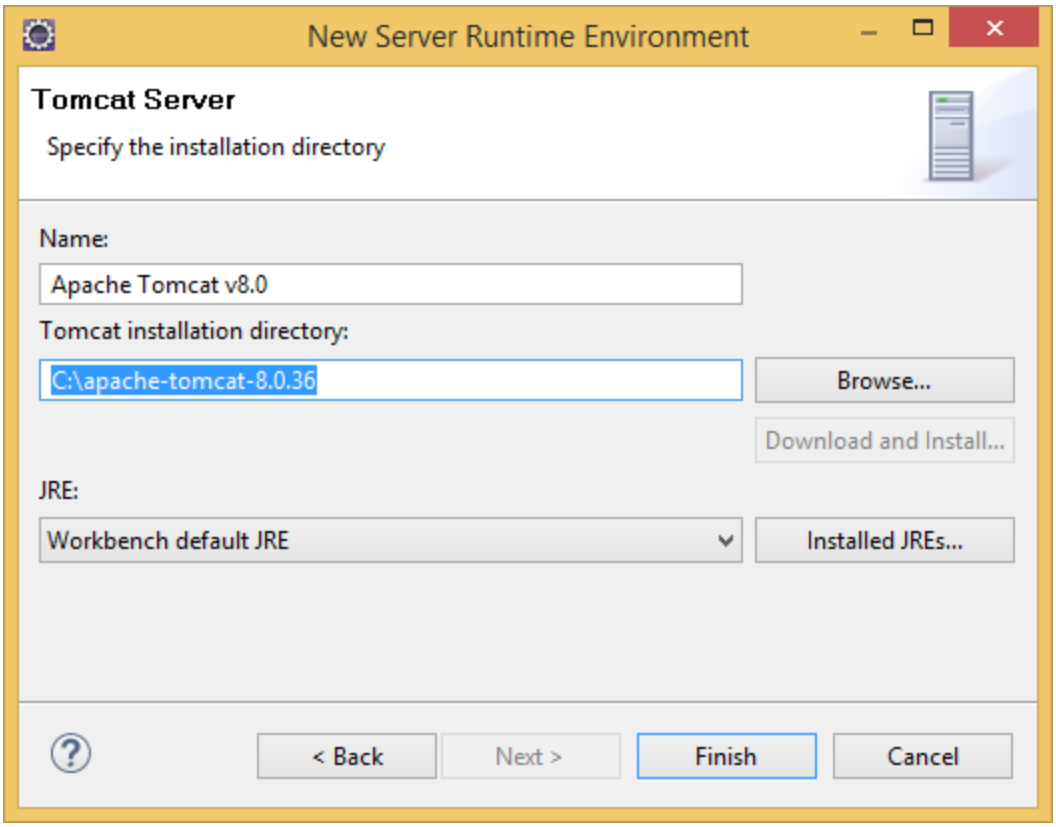
- For initial time server configuration open the Windows -> Preferences options. * In Preferences window, choose the Server -> Runtime Environment.



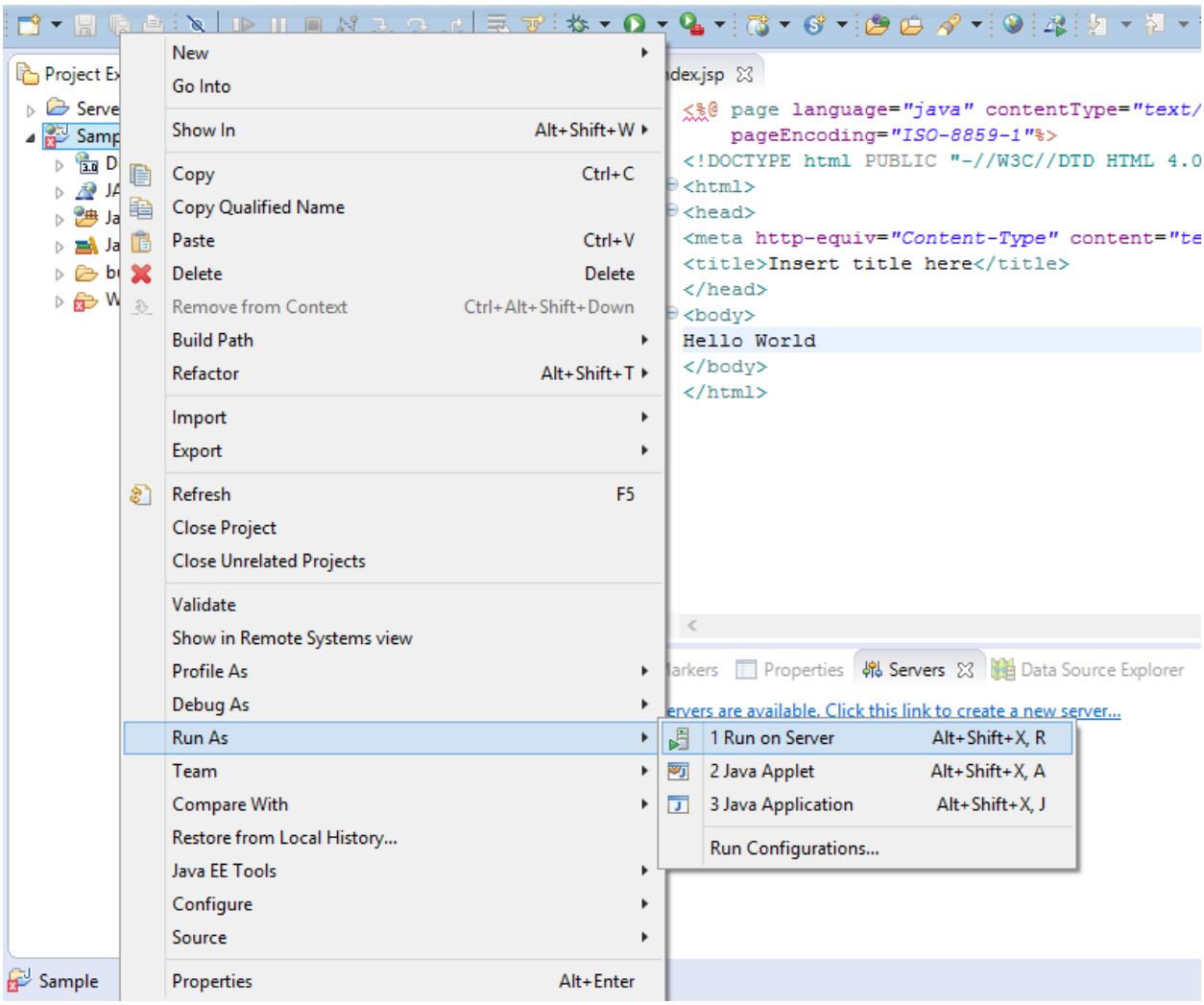
- Now click the Add button to configure the new server for your project deployment.
- In this New Server Runtime Environment window, choose the latest Apache Tomcat version and click the Next button.



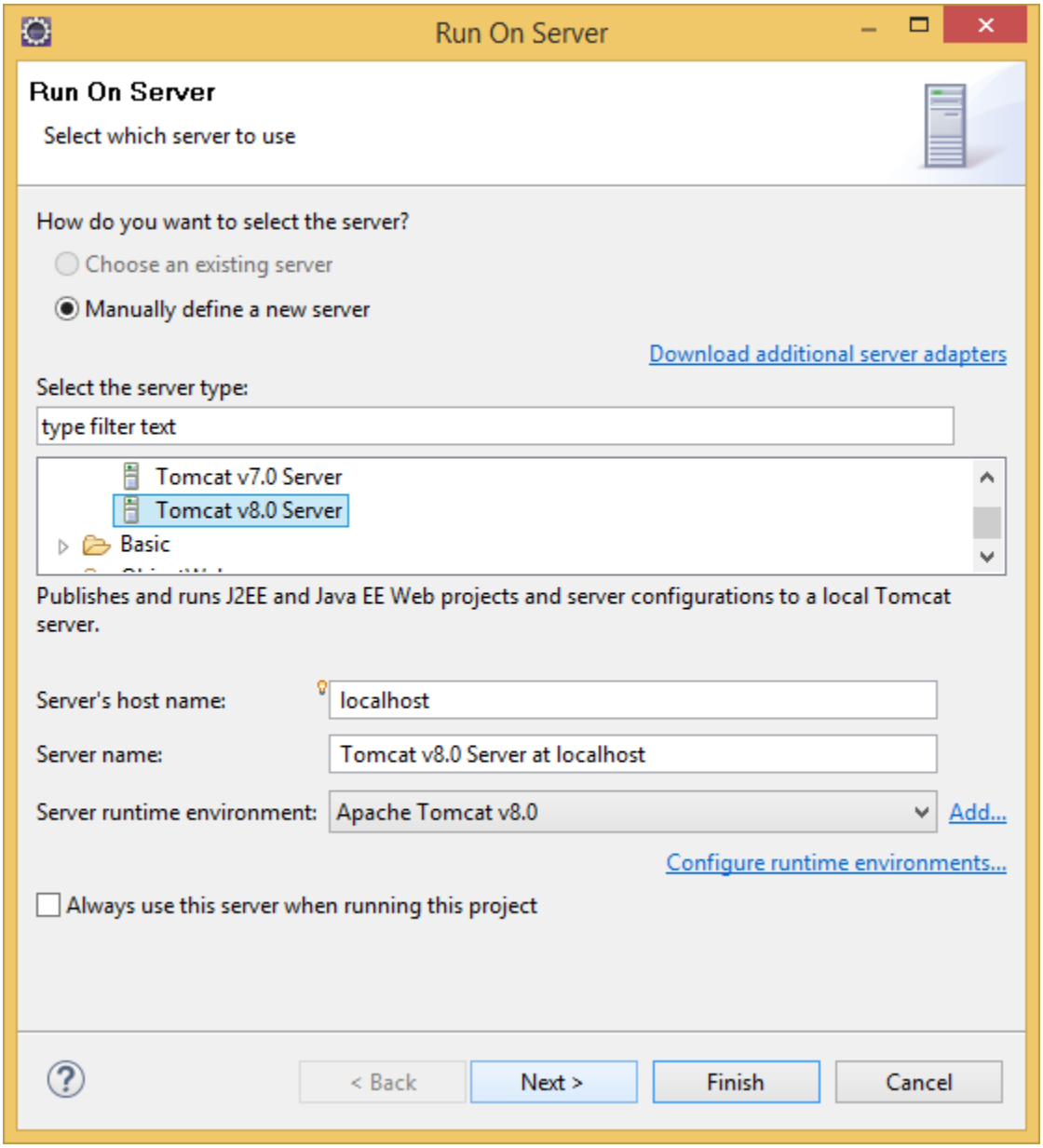
- Now specify your Apache Tomcat installation location by using browse button. Then Click the Finish button to complete server configurations.



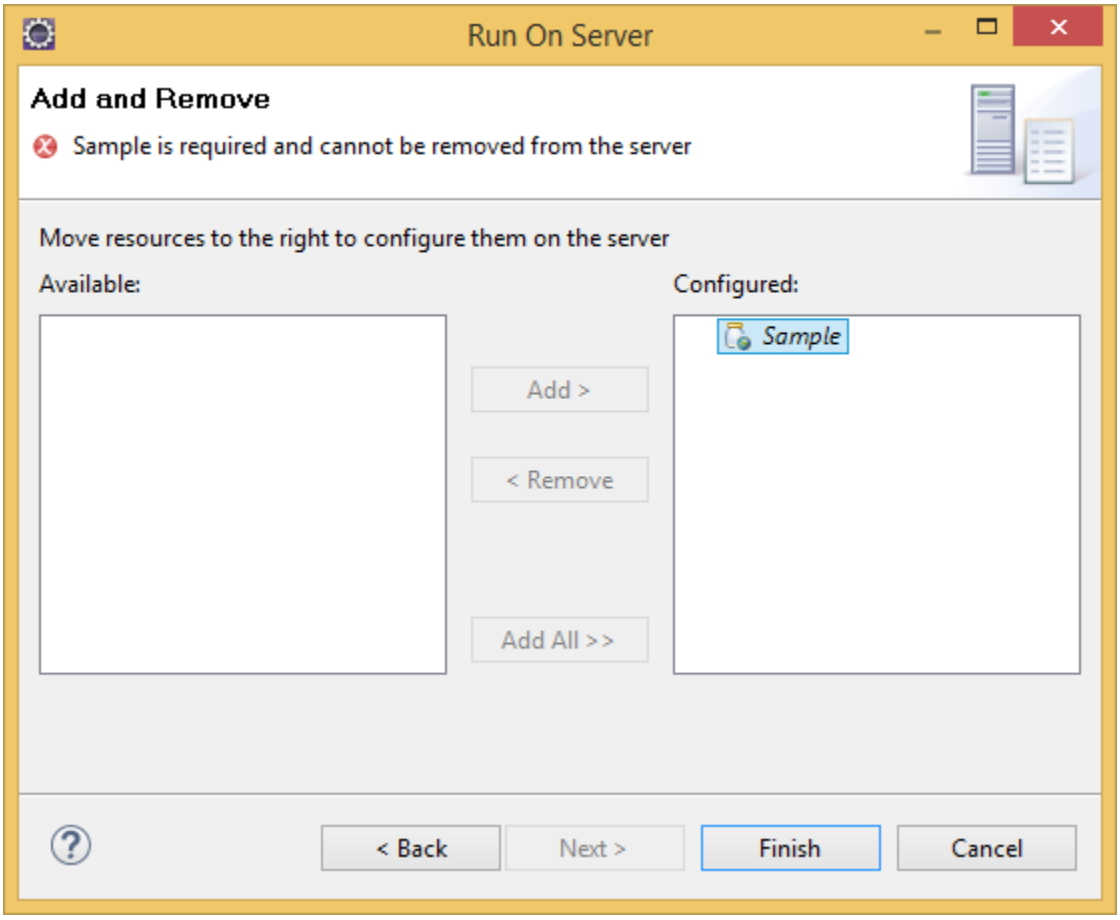
- Now your application is ready for deployment. Right click your project in your Project Explorer window, then choose the Run As - > Run on Server option.



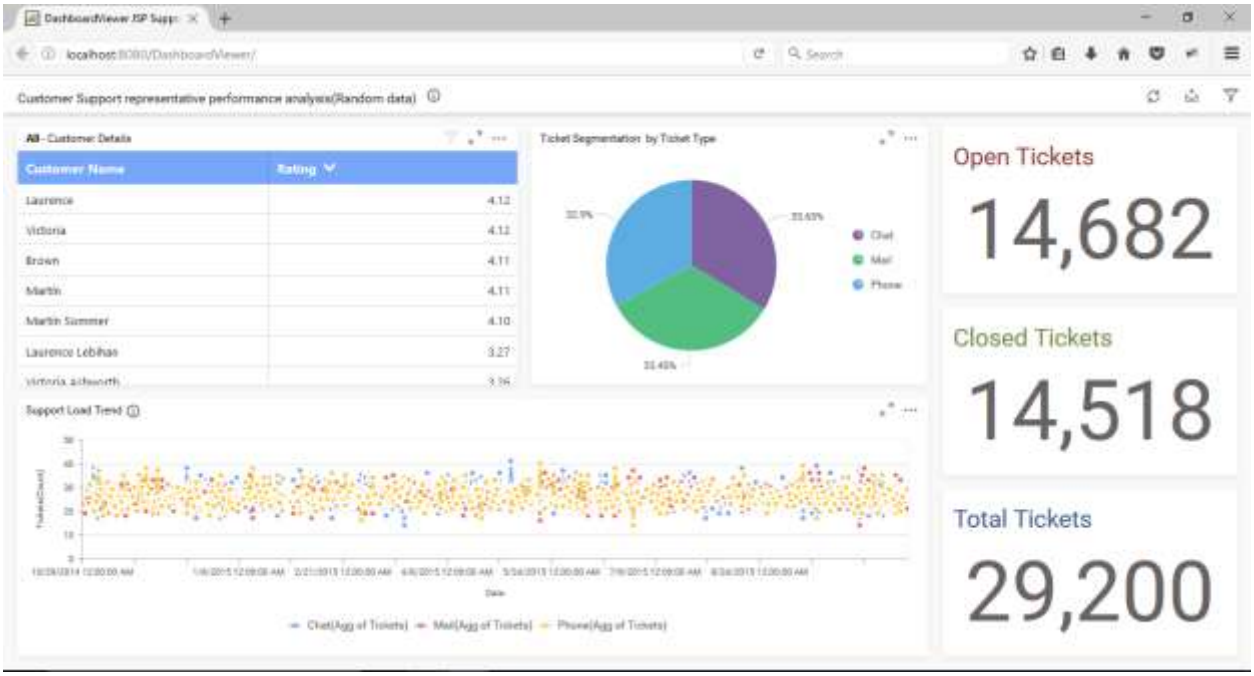
- In the Run on Server Window select our Apache Tomcat version and click the Next button.



- Select your application project in the right side pane and click the Finish button.



- Now your JSP sample will be deployed in Apache sever and the output will be displayed.



Information: Starting with version 4.1.0.x, Dashboard Platform SDK has been temporarily suspended from working with dashboards that are created using web-based Dashboard Designer integrated within Dashboard Server.

TypeScript

Overview

The Syncfusion Dashboard Platform SDK provides the default type definition file `ej.dashboardviewer.all.d.ts` to include the support for type-checking while initializing the DashboardViewer. The important step you need to do is to copy the ej.dashboardviewer.all.d.ts file into your project and then need to refer it in your TypeScript application (`app.ts` file), so that you will get the IntelliSense support and the compile time type-checking.

Apart from `ej.dashboardviewer.all.d.ts` file, it is also necessary to make use of the `jquery.d.ts` file in your TypeScript application.

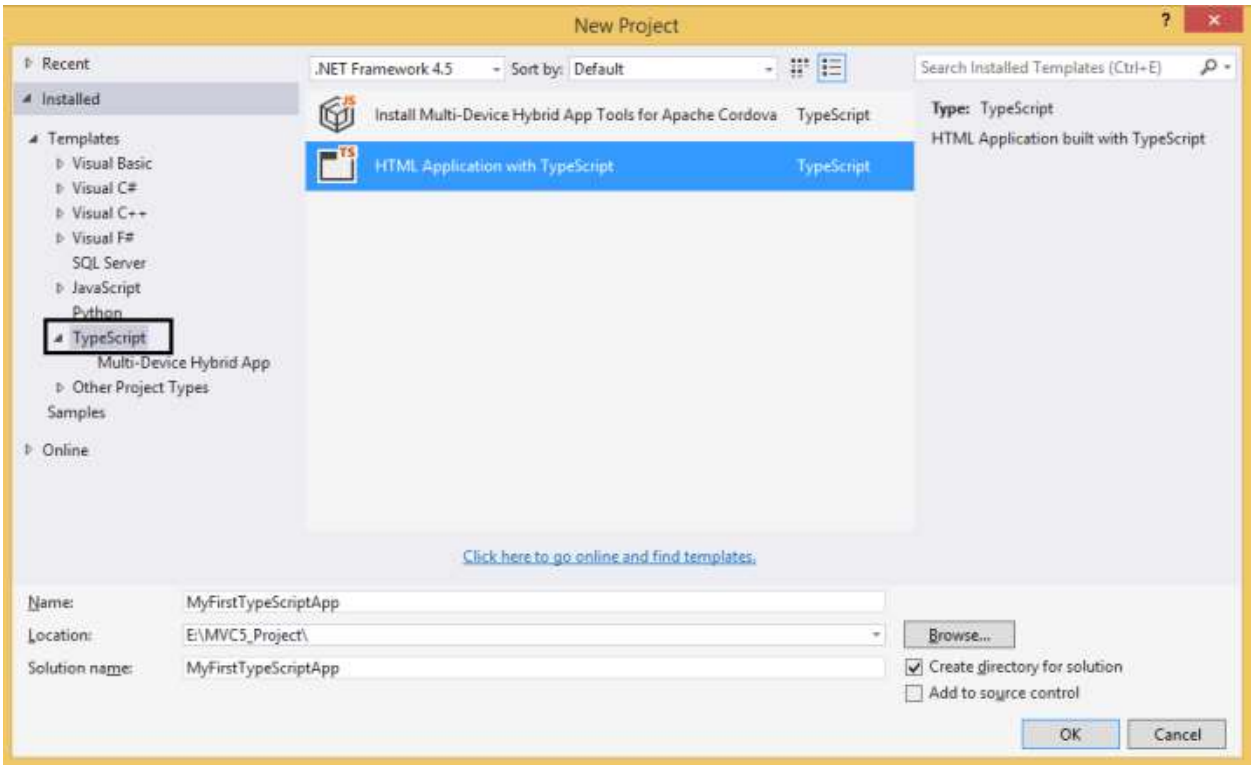
Prerequisites

To work with TypeScript, the below mentioned System requirements are necessary,

- Microsoft Visual Studio 2012 or higher [upto VS2017]
- Install the TypeScript Extension for Visual Studio.

Creating a TypeScript application with DashboardViewer Control Wrapper

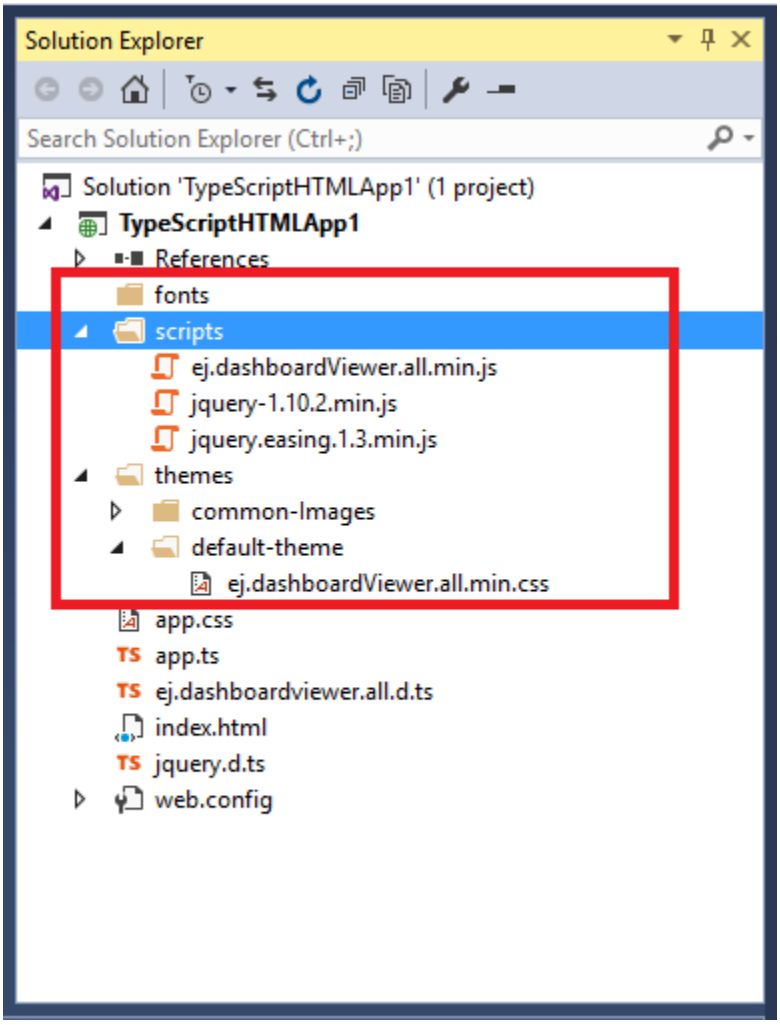
Start Visual Studio 2015 and Create a new TypeScript Application from `File|New|Project` and `save` it with a meaningful name as shown below (Select the **TypeScript** option, which is available by default in the listed Templates on the left side pane in the New Project),



Add the required Scripts and style sheets into your Project, as shown below – Copy the required Scripts ,themes and fonts from the sample installed location on your machine into your new TypeScript application for rendering the DashboardViewer.

Here is the default sample installation location.

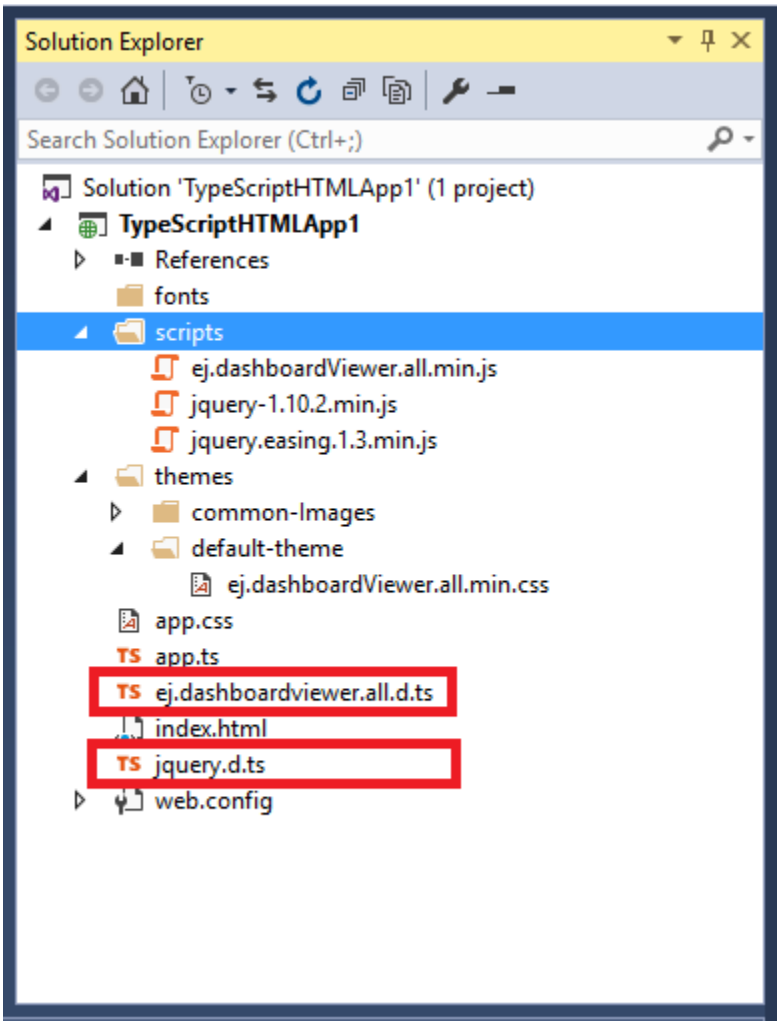
%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JavaScript\typescript app



Scripts & themes and fonts folder copied into current project

Add the ej.dashboardviewer.all.d.ts and jquery.d.ts type-definition files in your project and refer it in the app.ts file of your project as shown below,which is available in sample installed location mentioned below,

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\JavaScript\typescript app\tsFiles



Now, refer these two files within the app.ts file (before referring these files, remove all the unwanted content in that app.ts file) as shown below,

JAVASCRIPT

```
/// <reference path="jquery.d.ts" />
/// <reference path="ej.dashboardviewer.all.d.ts" />
```

Within the **Index.html** page, define the container name for the DashboardViewer control to be used and make the Script and CSS references in this page as shown below,

HTML

```
<html>
<head>
<title>Getting Started TypeScript - DashboardViewer</title>
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet">
<script src="scripts/jquery-1.10.2.min.js"></script>
<script src="scripts/jquery.easing.1.3.min.js"></script>
<script src="scripts/ej.dashboardViewer.all.min.js"></script>
<script src="app.js"></script>
```



```
<style>
body, html, #dashboard {
overflow: hidden !important;
width: 100%;
height: 100%;
}
</style>
</head>
<body style="background-color: #fcf7f7; height: 100%; width: 100 %;">
<!-- Control Initialization -->
</body>
</html>
```

Control Initialization

The DashboardViewer can be created from a HTML 'div' element with the HTML 'id' attribute.

HTML

```
<div class="content-container-fluid" style="height:100%;width:100%;">
<div style="height:100%;width:100%;" id="dashboard">
</div>
</div>
```

open the app.ts file and paste the below code

JAVASCRIPT

```
/// <reference path="jquery.d.ts" />
/// <reference path="ej.dashboardviewer.all.d.ts" />
module DashboardViewerComponent {
$(function () {
var dashboardviewer = new ej.DashboardViewer($("#dashboard"), {
dashboardPath:
"https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDashboard
.sydx",
serviceUrl:
"https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc"
});
});
}
```

Note: Provided online Dashboard Service URL and Dashboard Path are for demo purpose.

Make sure the given dashboard path should be accessible with the given Dashboard Service URL.

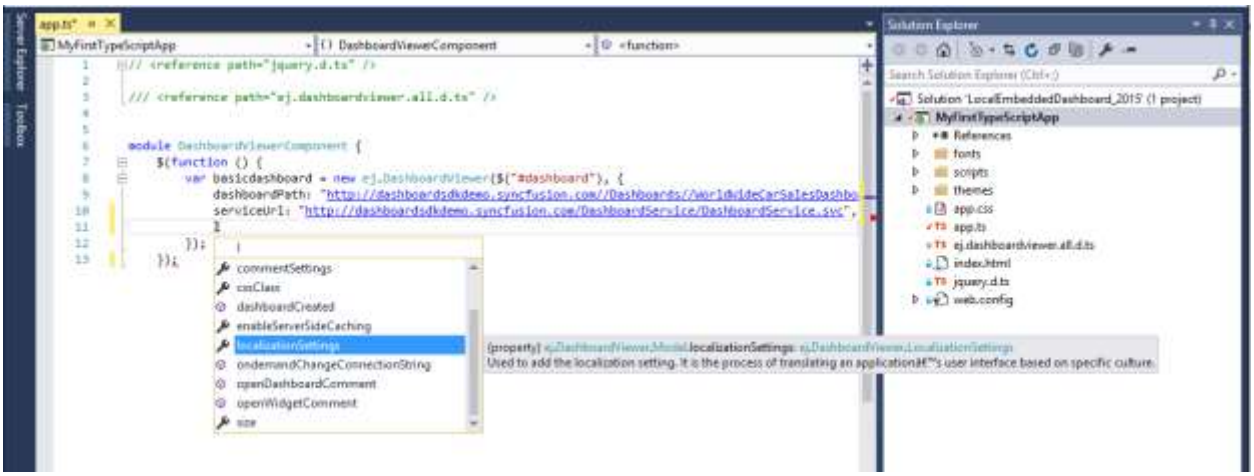
Now build your application, so that the **app.js** file is automatically generated and got added to your project (User have nothing to do with this file). Now, whatever code changes that you make in **app.ts** file will be reflected in app.js file automatically.

Usually, the DashboardViewer widget initialization is done within this **app.ts** file using either of the following two ways.

- Widget Class
- jQuery Interface

Widget Class

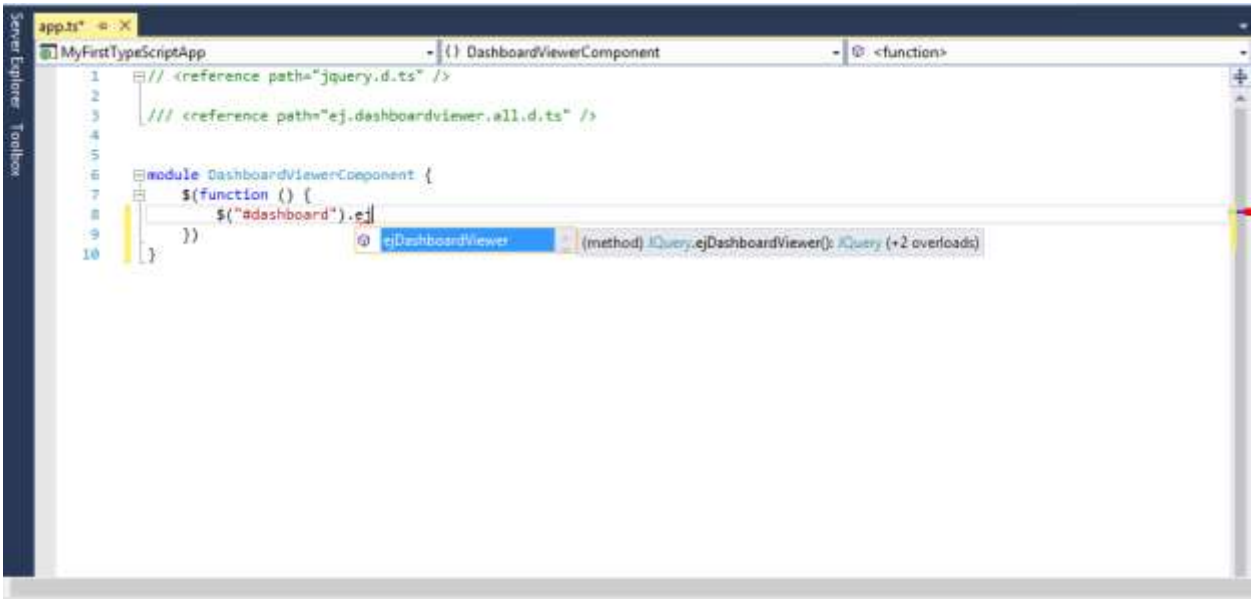
Initialization of DashboardViewer widget can be done through the instance created for its class.



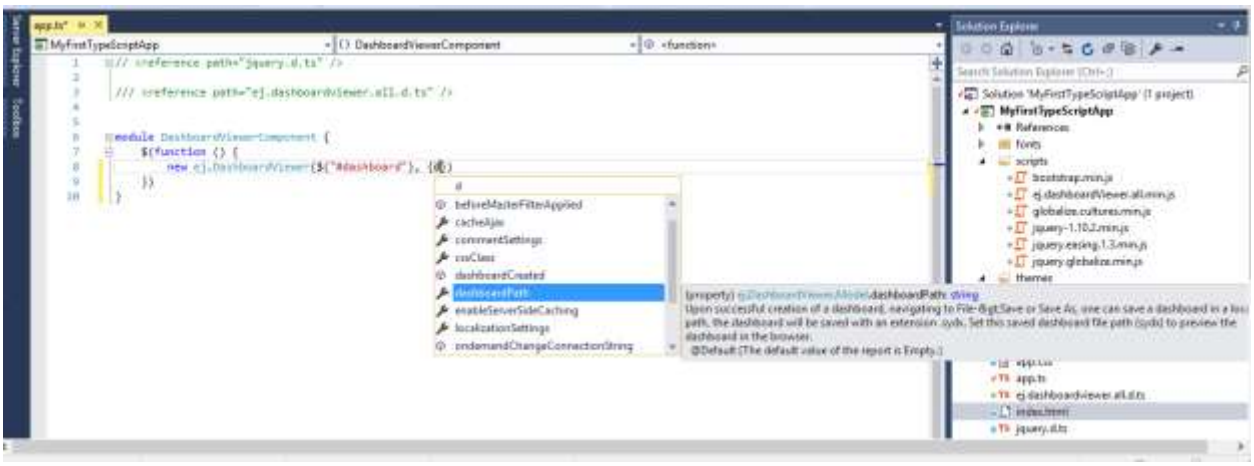
Initializing DashboardViewer widget through ej.DashboardViewer class

jQuery Interface

Here, the widget are initialized using the plug-in name, by passing all the required widget properties to it. The property names can be accessible through intelliSense and while providing input values to those properties – if any wrong data values were assigned to the properties by the user, then it will be automatically notified to the user at the compile time itself with an error message.



Initializing the DashboardViewer widget through intelliSense



Configuring the DashboardViewer properties

Configure the Service URL and the dashboard file to access through respective properties to render in DashboardViewer.

Binding Dashboard Service

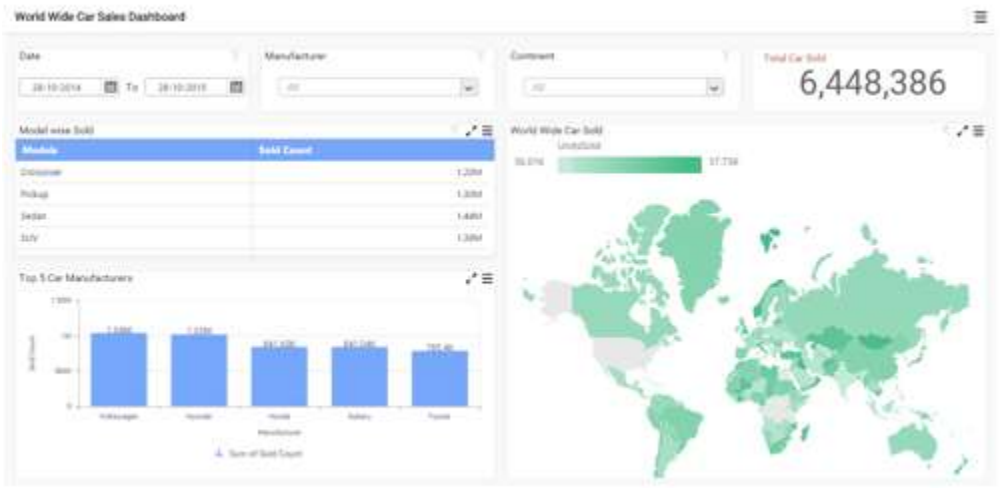
To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Running the application

Build and Run your new TypeScript Project. DashboardViewer will be rendered with a dashboard embedded like below.



Universal Windows Platform (UWP)

Overview

The Syncfusion Dashboard Platform SDK includes a Dashboard Viewer HTML 5 control that can be embedded within your Universal Windows Platform (UWP) application.

System Requirements:

To work with UWP, the following IDE can be used for development that are compatible with Microsoft Windows, both 32bit and 64bit Operating System in Windows 10

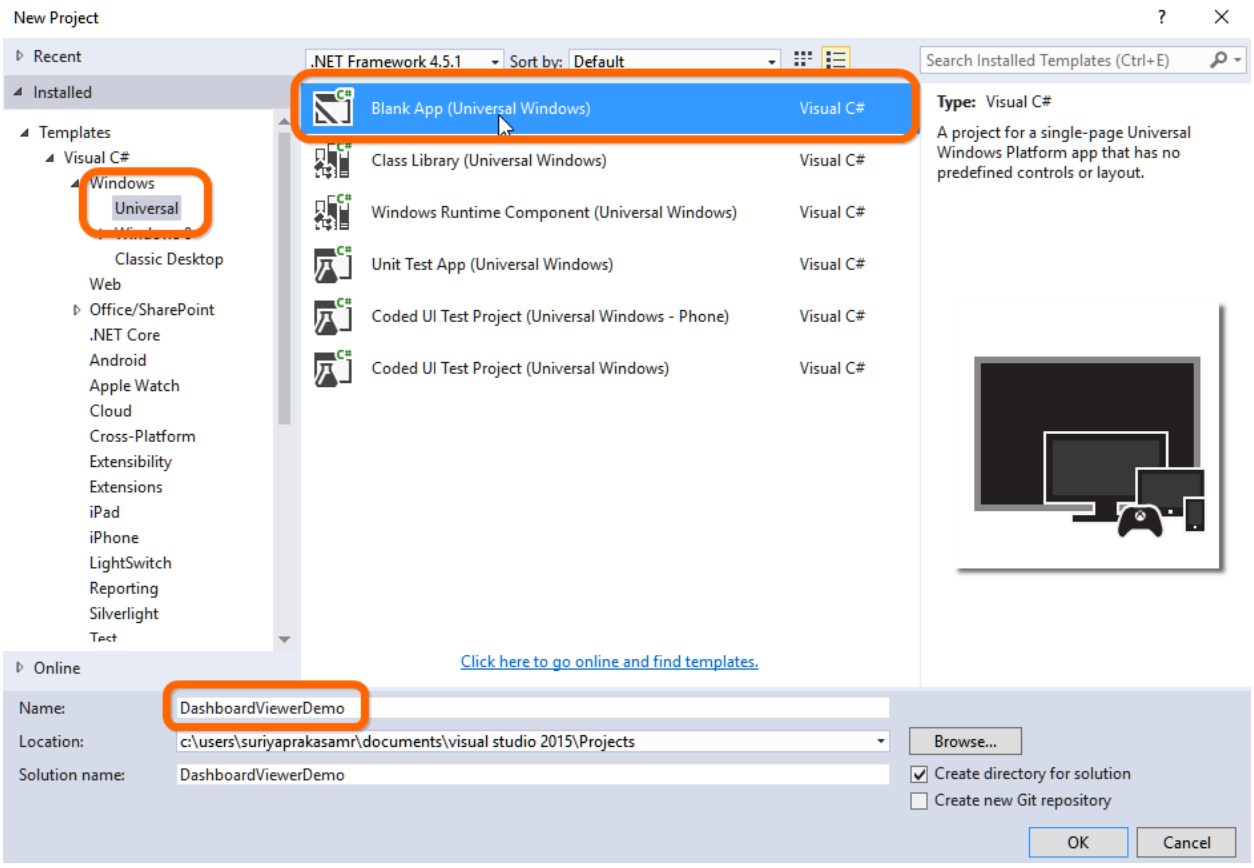
- Microsoft Visual Studio 2015 or higher.

Getting Started

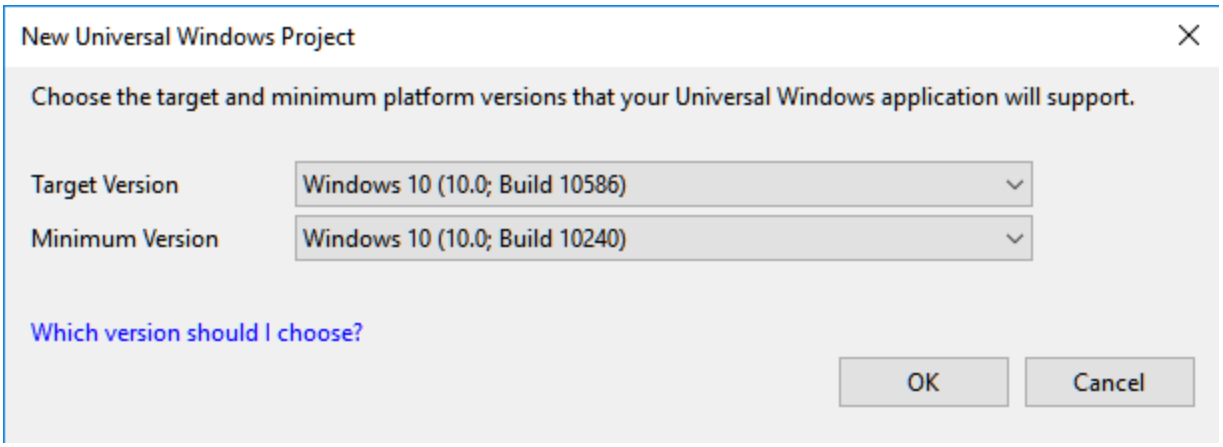
Configure Syncfusion Dashboard Viewer Component in UWP Application:

The following steps helps to create a UWP application by embed the Dashboard Viewer component.

- Open Visual Studio 2015, Create a new project and select Universal Template in Windows group under Visual C# category, type your application name and Click OK.

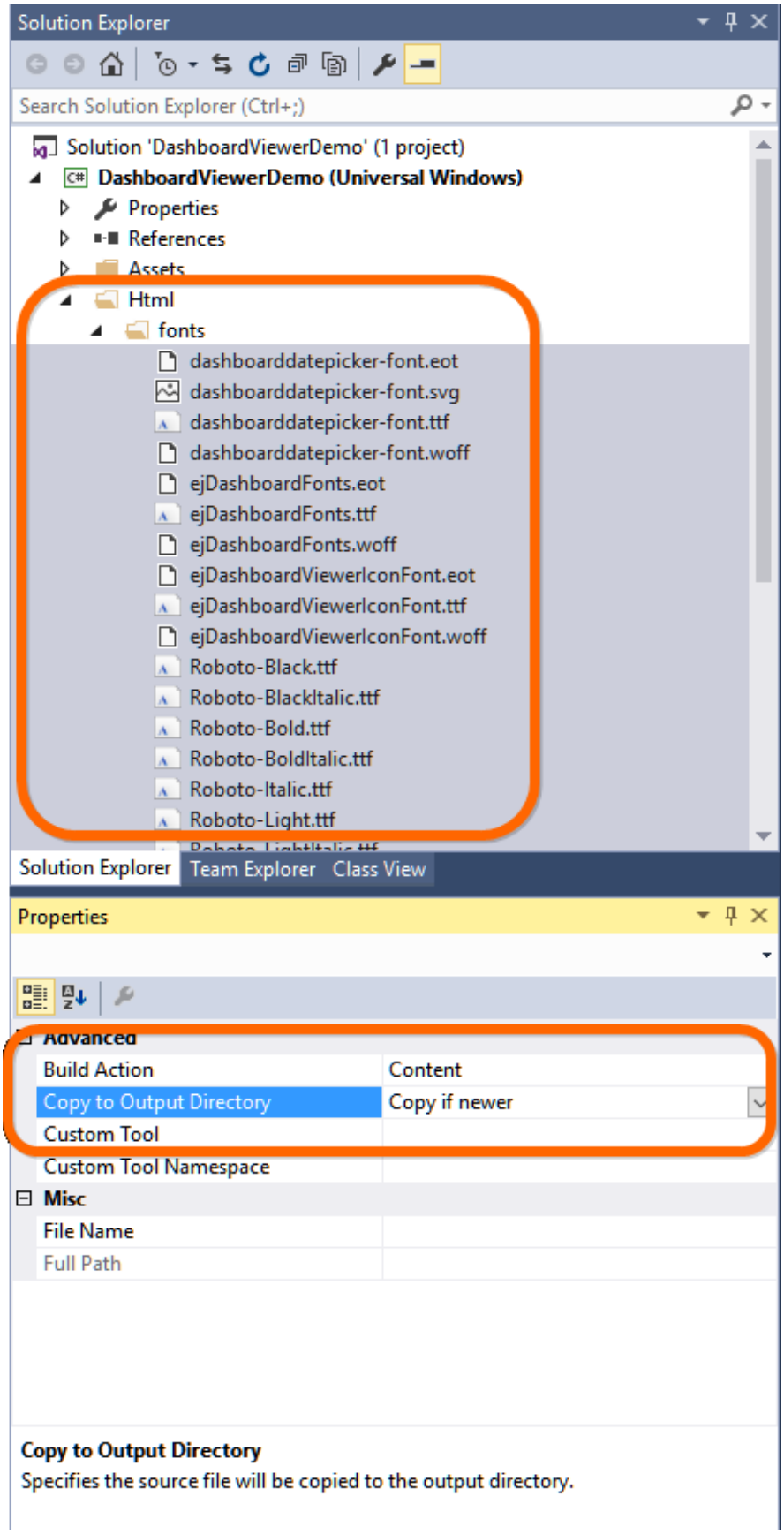


- Select the Target version of your Windows SDK as in below example image and Click OK. Refer here for more detail.



- Once project created, copy HTML folder (To place the font, script and theme files) from the below location and paste into your application and change the property 'Build action' as 'Content' and 'Copy to Output Directory' as 'Copy if newer' for the entire content of HTML folder.

```
%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html
```



Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'DashboardViewerDemo' (1 project)
DashboardViewerDemo (Universal Windows)
Properties
References
Assets
Html
fonts
dashboarddatepicker-font.eot
dashboarddatepicker-font.svg
dashboarddatepicker-font.ttf
dashboarddatepicker-font.woff
ejDashboardFonts.eot
ejDashboardFonts.ttf
ejDashboardFonts.woff
ejDashboardViewerIconFont.eot
ejDashboardViewerIconFont.ttf
ejDashboardViewerIconFont.woff
Roboto-Black.ttf
Roboto-BlackItalic.ttf
Roboto-Bold.ttf
Roboto-BoldItalic.ttf
Roboto-Italic.ttf
Roboto-Light.ttf
Solution Explorer
Team Explorer
Class View
Properties
Advanced
Build Action
Content
Copy to Output Directory
Copy if newer
Custom Tool
Custom Tool Namespace
Misc
File Name
Full Path
Copy to Output Directory
Specifies the source file will be copied to the output directory.

- Go to MainPage.xaml and add a WebView and provide suitable name.

ASPX-CS

```
<Page
x:Class="DashboardViewerDemo.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:local="using:DashboardViewerDemo"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d">
<Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
<WebView x:Name="dashboardViewerWebView" />
</Grid>
</Page>
```

- Go to MainPage.xaml.cs and generate the HTML string to render Dashboard Viewer and save it as an HTML file in the installed location. Do the following steps:

Step 1: Include the following namespaces:

C#

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading.Tasks;
using Windows.Storage;
using Windows.Storage.Streams;
using Windows.UI.Popups;
using Windows.UI.Xaml.Controls;
```

Step 2: Declare the constants inside MainPage class:

C#

```
private const string BeforeHeadHtml = "<!DOCTYPE html>\n"
+ "<html xmlns=\"http://www.w3.org/1999/xhtml\" style=\"width:100%;
height:100%;\">\n"
+ "<head>\n"
+ "<meta charset=\"utf-8\" content=\"width=device-width, initial-
scale=1.0\">"
+ "<script>window.destroyAll = function(){try{ej.widget.destroyAll($('.e-
js').off());}catch(e){} $(document.body).html('-'); CollectGarbage(); };
</script>";
private const string serviceUrlString =
"https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc";
private const string dashboardFilePath =
"https://dashboardsdk.syncfusion.com/Dashboards/WorldWideCarSalesDashboard.s
ydx";
```

Note: *The above two lines has demo links for both Dashboard Service and Dashboard file. Provide your valid URL of Dashboard Service and Dashboard (SYDX) file path in the highlighted lines.*

To know how to host the Dashboard Service Refer Binding Dashboard Service

Step 3: Generate the HTML string to embed the Dashboard Viewer MainPage class:

C#

```
private string GetHtmlString()
{
var jsFiles = new List<string>
{
"jquery-1.10.2.min.js",
"jquery.easing.1.3.min.js",
"ej.dashboardViewer.all.min.js"
};
var cssFiles = new List<string>
{
"default-theme/ej.dashboardViewer.all.min.css"
};
var headSb = new StringBuilder();
foreach (var item in jsFiles)
{
//headSb.Append("<script src=\"ms-appx-web:///Html/scripts/" + item + "\"
type=\"text/javascript\"></script>");
headSb.Append("<script src=\"Html/scripts/" + item + "\"
type=\"text/javascript\"></script>");
}
foreach (var item in cssFiles)
{
//headSb.Append("<link href=\"ms-appx-web:///Html/themes/" + item + "\"
rel=\"stylesheet\"></link>");
headSb.Append("<link href=\"Html/themes/" + item + "\"
rel=\"stylesheet\"></link>");
}
var htmlSb = new StringBuilder();
htmlSb.Append(BeforeHeadHtml);
htmlSb.Append(headSb);
htmlSb.Append("</head>\n");
htmlSb.Append(GetViewerString());
htmlSb.Append("</html>");
return htmlSb.ToString();
}
private string GetViewerString()
{
var viewerString = "$('#dashboard').ejDashboardViewer({serviceUrl: '" +
serviceUrlString +
"', dashboardPath: '" +
dashboardFilePath.Replace(@"\", @"\\") +
"',filterParameters:location.search.substr(1)});";
var stringBuilder = new StringBuilder();
stringBuilder.Append("<body style=\"width:100%;
height:100%;overflow:hidden;\">");
stringBuilder.Append("<div id=\"dashboard\" style=\"width:100%;
height:100%;\" />");
stringBuilder.Append("<script type=\"text/javascript\"
language=\"javascript\">");
stringBuilder.Append(viewerString);
stringBuilder.Append("</script>");
```



```
stringBuilder.Append("</body>");
return stringBuilder.ToString();
}
```

Step 4: Write the generated string as an HTML file and Navigate the URI value of HTML file location inside the MainPage class:

C#

```
StorageFolder installedLocation;
Uri htmlUri;
public MainPage()
{
this.InitializeComponent();
Windows.ApplicationModel.Package package =
Windows.ApplicationModel.Package.Current;
installedLocation = package.InstalledLocation;
FillHtmlUriFromInstalledLocation();
this.dashboardViewerWebView.Navigate(htmlUri);
}
#region Get URL of HTML file
private async void FillHtmlUriFromInstalledLocation()
{
string htmlFileName = string.Empty;
var task = Task.Run(async () => { htmlFileName = await
GetUrlOfHtmlFromInstalledLocation(); });
task.Wait();
htmlUri = new Uri("ms-appx-web:///" + htmlFileName);
}
private async Task<string> GetUrlOfHtmlFromInstalledLocation()
{
try
{
byte[] htmlAsBytes = System.Text.Encoding.UTF8.GetBytes(GetHtmlString());
StorageFile file = await installedLocation.CreateFileAsync("Dashboard.html",
CreationCollisionOption.GenerateUniqueName);
using (IRandomAccessStream fileStream = await
file.OpenAsync(FileAccessMode.ReadWrite))
{
using (IOutputStream outputStream = fileStream.GetOutputStreamAt(0))
{
using (DataWriter dataWriter = new DataWriter(outputStream))
{
dataWriter.WriteBytes(htmlAsBytes);
await dataWriter.StoreAsync();
dataWriter.DetachStream();
}
await outputStream.FlushAsync();
}
}
return file.Name;
}
catch (Exception ex)
{
await new MessageDialog(ex.Message, "Error").ShowAsync();
}
```



```
return null;
}
#endregion
```

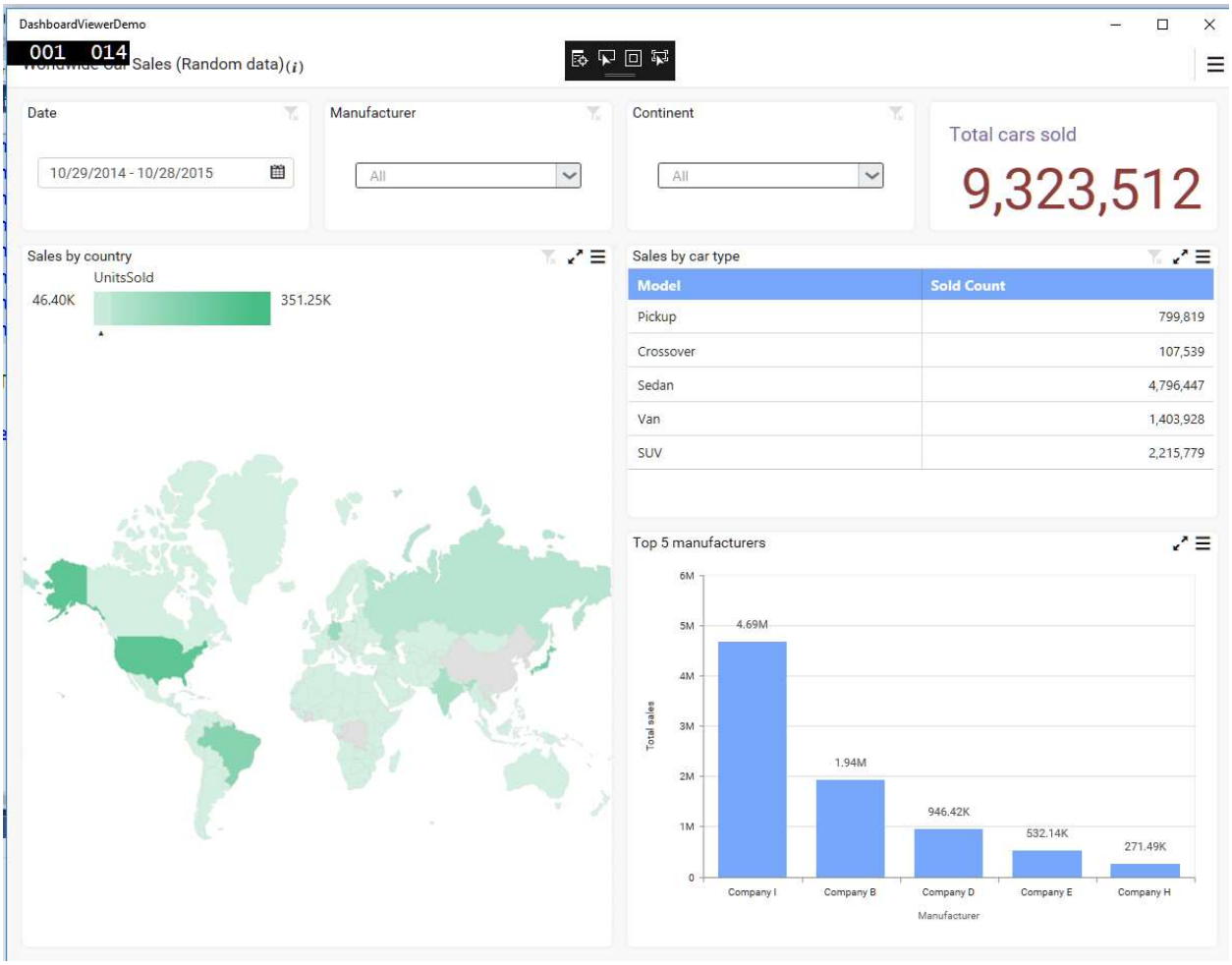
Binding Dashboard Service

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

- Finally compile your project, after successful compilation then press F5 key to deploy and Run your project.



Getting Started with Windows Forms Application

This section describes how to create a Windows Forms application with embedded dashboard viewer.

Project Creation

Create a new Windows Forms Application Project using Microsoft Visual Studio IDE 2012 or higher.

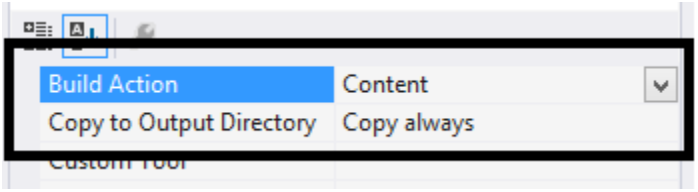
Adding files and references

Add the scripts, styles and refer fonts that are required for the dashboard from the following location to the application project and set the Build Action and Copy to Output Directory properties to Content and Copy always, respectively as shown in below screenshot for the items inside those folders.

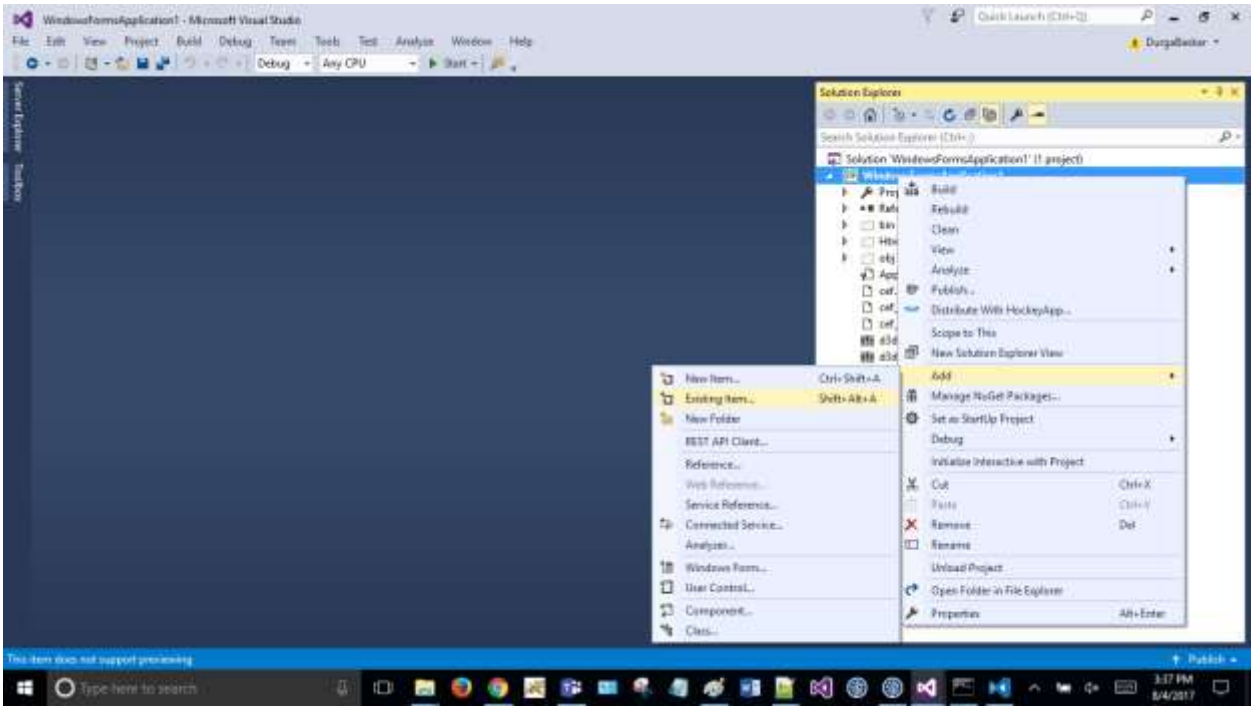
%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html

Include the dashboard file (*.sydx) in the project from the below mentioned location and set the Build Action and Copy to Output Directory properties to Content and Copy always, respectively as shown in below screenshot.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Dashboards



Include the below mentioned files into project from the SDK build installed location as shown in figure below.



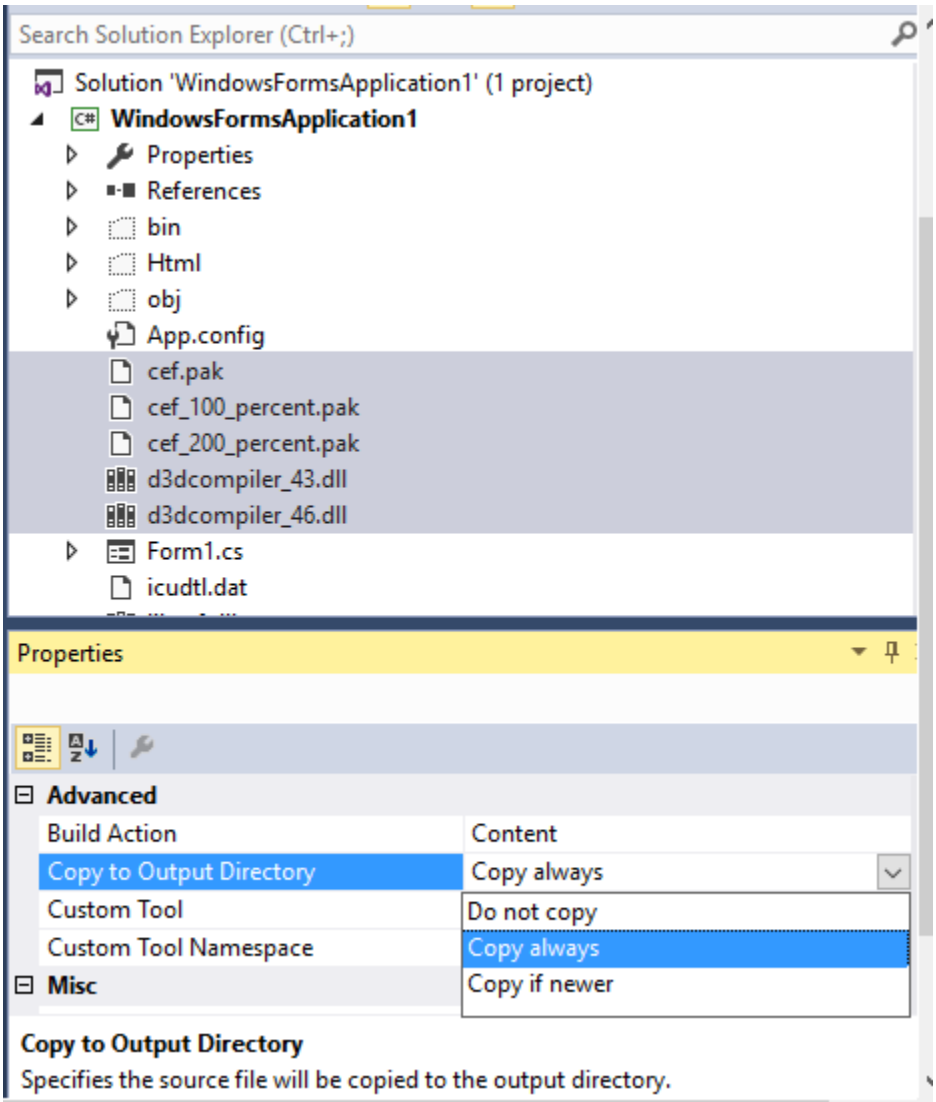
- cef.pak
- cef*100*percent.pak
- cef*200*percent.pak

- d3dcompiler_43.dll
- d3dcompiler_46.dll
- icudtl.dat
- libcef.dll.dll
- libEGL.dll
- libGLESv2.dll

The above mentioned files can be found in the following Dashboard SDK samples location:

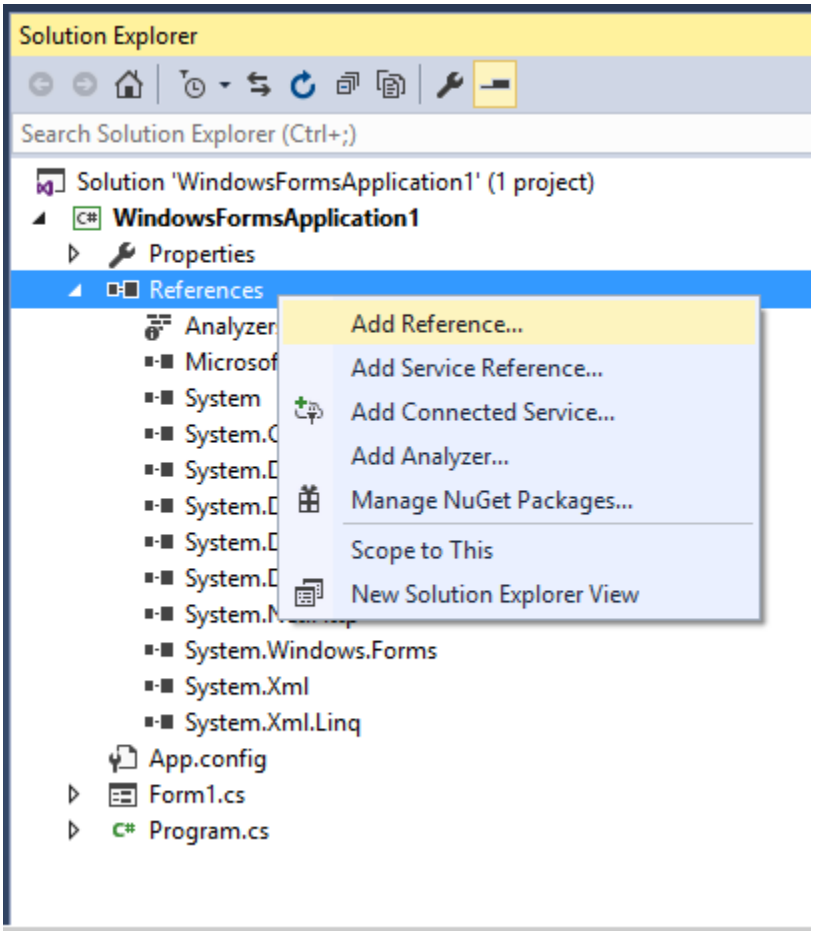
`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Packages`

Set the `Build Action` property to `Content` and the `Copy to Output Directory` property to `Copy always` as shown in the following image for all the files added to the project.



Adding Dashboard Viewer Assembly References

Right-click the project and add the following assembly references through choosing `Add > Reference...` as shown below,

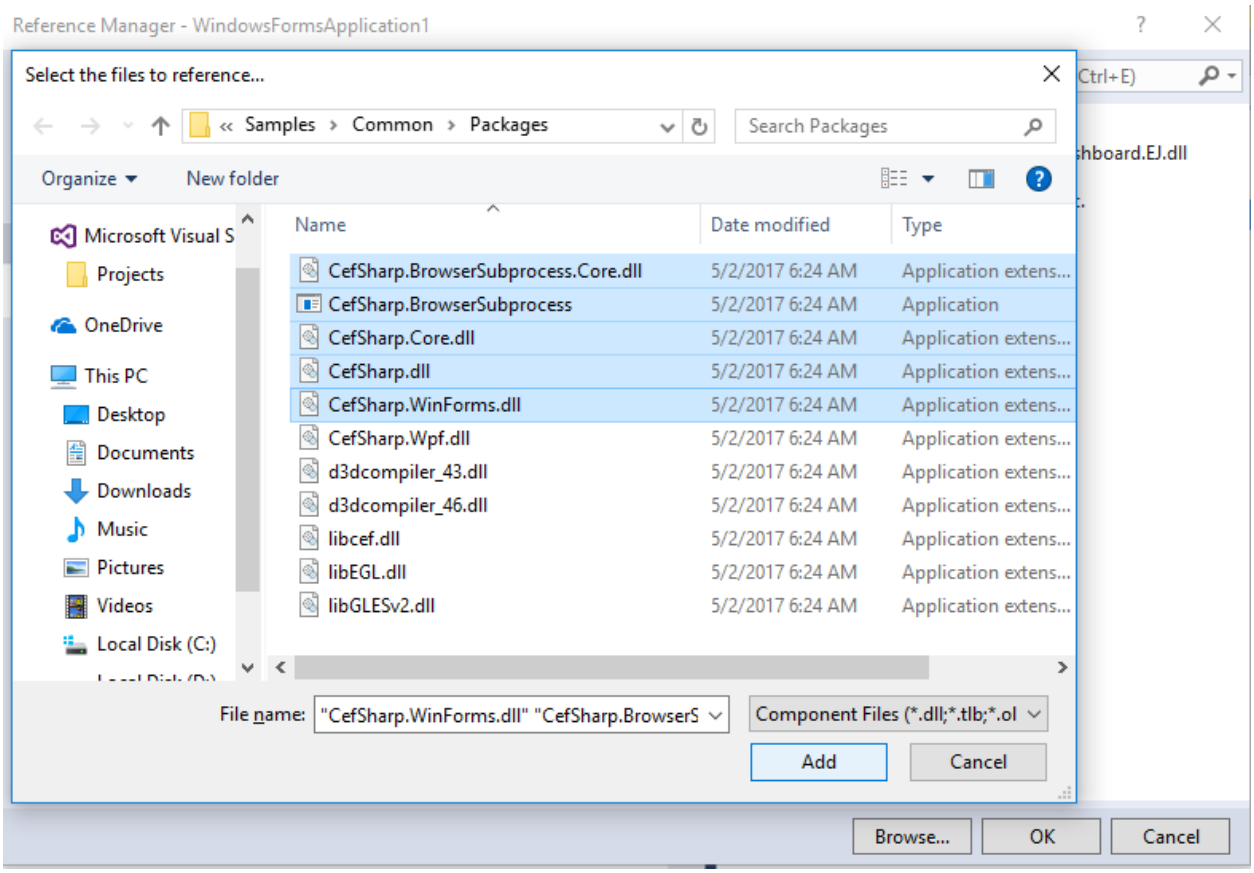


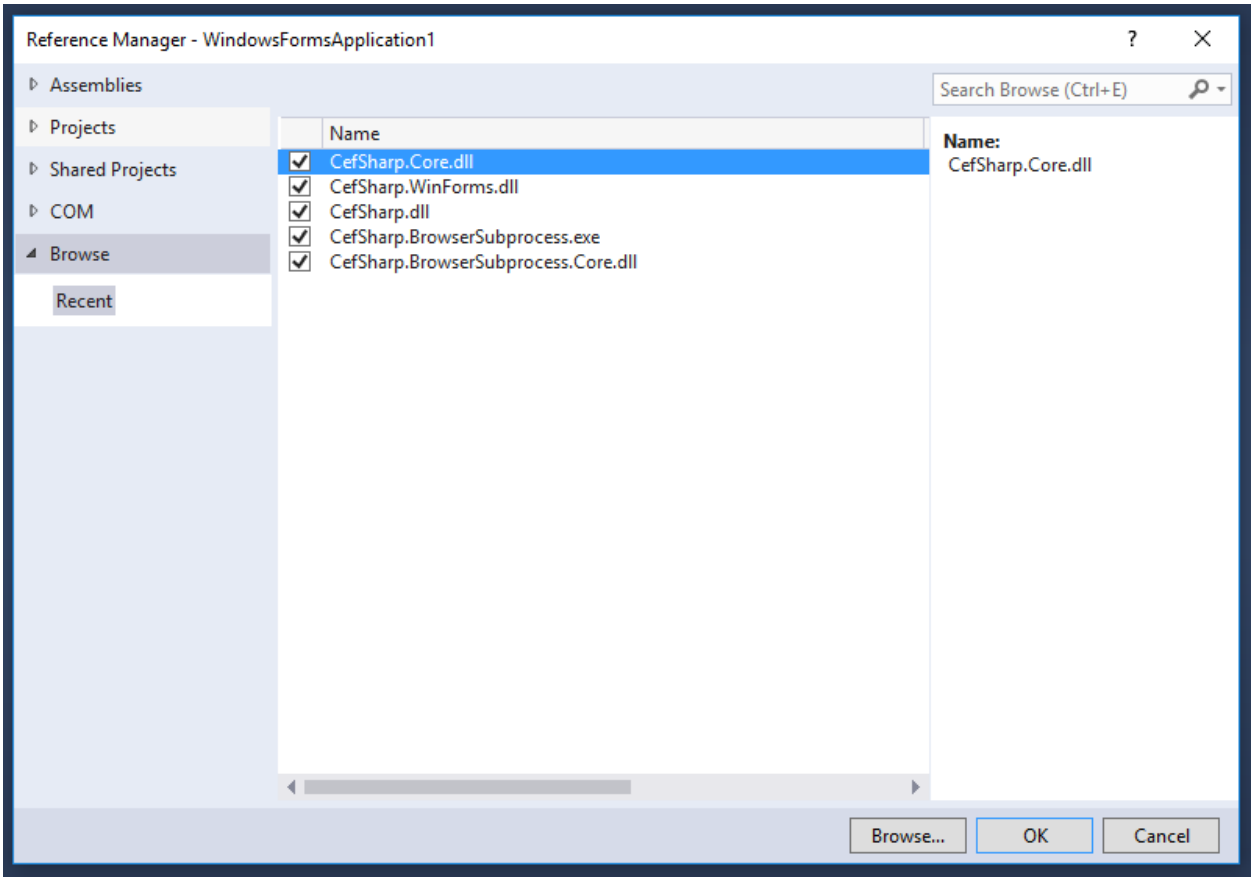
selecting the below mentioned assemblies in dialog shown, which allows you to use any of the Syncfusion Windows Form control within it.

- CefSharp.Core.dll
- CefSharp.WinForms.dll
- CefSharp.dll
- CefSharp.BrowserSubprocess.dll
- CefSharp.BrowserSubprocess.Core.dll

The above mentioned assembly files can be found in the following Dashboard SDK samples location:

`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Packages`


Reference Manager - WindowsFormsApplication1
Select the files to reference...
Samples › Common › Packages
Search Packages
Organize
New folder
Microsoft Visual S
Projects
OneDrive
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Name
Date modified
Type
CefSharp.BrowserSubprocess.Core.dll 5/2/2017 6:24 AM Application extens...
CefSharp.BrowserSubprocess 5/2/2017 6:24 AM Application
CefSharp.Core.dll 5/2/2017 6:24 AM Application extens...
CefSharp.dll 5/2/2017 6:24 AM Application extens...
CefSharp.WinForms.dll 5/2/2017 6:24 AM Application extens...
CefSharp.Wpf.dll 5/2/2017 6:24 AM Application extens...
d3dcompiler_43.dll 5/2/2017 6:24 AM Application extens...
d3dcompiler_46.dll 5/2/2017 6:24 AM Application extens...
libcef.dll 5/2/2017 6:24 AM Application extens...
libEGL.dll 5/2/2017 6:24 AM Application extens...
libGLESv2.dll 5/2/2017 6:24 AM Application extens...
File name: "CefSharp.WinForms.dll" "CefSharp.BrowserS
Component Files (*.dll;*.tlb;*.ol
Add
Cancel
Browse...
OK
Cancel



Generating URL

Add the below code Main method of Program class so to initialize the ChromiumWebBrowserinstance before the code where Form instance is initialized.

C#

```
var settings = new CefSettings();
settings.PackLoadingDisabled = true;
settings.LogSeverity = LogSeverity.Disable;
if(!Cef.Initialize(settings))
{
if(Environment.GetCommandLineArgs().Contains("--type=renderer"))
Environment.Exit(0);
}
```

VB.NET

```
Dim settings = New CefSettings()
settings.PackLoadingDisabled = True
settings.LogSeverity = LogSeverity.Disable
If Not Cef.Initialize(settings) Then
If Environment.GetCommandLineArgs().Contains("--type=renderer") Then
Environment.[Exit](0)
End If
End If
```


Include the following code under the Form1 class that you will launch with dashboard from Main method.

C#

```
using CefSharp;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
public partial class Form1 : Form
{
private string _sydxFileName = "WorldWideCarSalesDashboard.sydx";
public string SydxFileName
{
get { return _sydxFileName; }
set { _sydxFileName = value; }
}
internal static Process dashboardServiceProcess = new Process();
public string Url { get; set; }
private string dashboardsDirectory = string.Empty;
// Get the service from Common Folder
private string serviceDirectory =
Path.GetFullPath(Path.Combine(@"..\..\..\..\")) + "Common\\Service";
/// <summary>
/// Initializing widget on  window and closing IIS Express while closing
widget window.
/// </summary>
public Form1()
{
this.FormClosing += Form1_Closing;
if (GenerateUrl())
InitializeComponent();
else
Environment.Exit(0);
}
/// <summary>
/// Close IIS Express Process while closing Window.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void Form1_Closing(object sender, FormClosingEventArgs e)
{
DashboardWindowsServiceInfo.ClearIISExpressProcess(dashboardServiceProcess);
}
/// <summary>
/// Generate the serviceUrl to render Dashboard
/// </summary>
private bool GenerateUrl()
{
```

```
DashboardWindowsServiceInfo dashboardViewer = new
DashboardWindowsServiceInfo(dashboardServiceProcess, serviceDirectory);
string url = dashboardViewer.GetUrlOfHtmlPage(SydxFileName);
if (!string.IsNullOrWhiteSpace(url))
{
Url = url;
return true;
}
else
return false;
}
}
```

VB.NET

```
Imports CefSharp
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Data
Imports System.Diagnostics
Imports System.Drawing
Imports System.Linq
Imports System.Text
Imports System.Threading.Tasks
Imports System.Windows.Forms
Imports System.IO
Public Partial Class Form1
Inherits Form
Private _sydxFileName As String = "WorldWideCarSalesDashboard.sydx"
Public Property SydxFileName() As String
Get
Return _sydxFileName
End Get
Set
_sydxFileName = value
End Set
End Property
Friend Shared dashboardServiceProcess As New Process()
Public Property Url() As String
Get
Return m_Url
End Get
Set
m_Url = Value
End Set
End Property
Private m_Url As String
Private dashboardsDirectory As String = String.Empty
' Get the service from Common Folder
Private serviceDirectory As String =
Path.GetFullPath(Path.Combine("..\..\..\..\")) + "Common\Service"
''' <summary>
''' Initializing widget on  window and closing IIS Express while closing
widget window.
''' </summary>
Public Sub New()
```

```
AddHandler Me.FormClosing, AddressOf Form1_Closing
If GenerateUrl() Then
InitializeComponent()
Else
Environment.[Exit](0)
End If
End Sub
''' <summary>
''' Close IIS Express Process while closing Window.
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub Form1_Closing(sender As Object, e As FormClosingEventArgs)
DashboardWindowsServiceInfo.ClearIISExpressProcess(dashboardServiceProcess)
End Sub
''' <summary>
''' Generate the serviceUrl to render Dashboard
''' </summary>
Private Function GenerateUrl() As Boolean
Dim dashboardViewer As New
DashboardWindowsServiceInfo(dashboardServiceProcess, serviceDirectory)
Dim url__1 As String = dashboardViewer.GetUrlOfHtmlPage(SydxFileName)
If Not String.IsNullOrWhiteSpace(url__1) Then
Url = url__1
Return True
Else
Return False
End If
End Function
End Class
```

Create a class named DashboardWindowsServiceInfo and add the below code within the class.

C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Security;
using System.Text;
using System.Windows;
using Microsoft.Win32;
using System.Xml.Serialization;
using System.Windows.Forms;
using System.Net.NetworkInformation;
using System.Net;
using System.Linq;
using System.Diagnostics;
using System.Globalization;
/// <summary>
///     Class for Dashboard Viewer
/// </summary>
public class DashboardWindowsServiceInfo
{
#region Private Variables
```

```
private readonly string _environmentFolder =
AppDomain.CurrentDomain.BaseDirectory;
string Version =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
();
private const string HtmlFolder = "Html\\";
public string ServiceUrl;
public string ErrorMessage;
#endregion
/// <summary>
///     Constructor for class DashboardWindowsServiceInfo
/// </summary>
public DashboardWindowsServiceInfo(Process serviceProcess, string
serviceDirectoryPath)
{
DashboardServiceProcess = serviceProcess;
string version =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
();
#region Pick IISExpress Dashboard Service Url
ServiceUrl = GetIISExpressServiceUrl(serviceDirectoryPath);
if (ValidateDashboardService(ServiceUrl))
{
ServiceUrl = string.Empty;
ErrorMessage = "An Error Occurred while hosting Dashboard Service";
MessageBox.Show(ErrorMessage);
Environment.Exit(0);
}
#endregion
}
/// <summary>
/// used to clear other Process while running IIS Express.
/// </summary>
/// <param name="process"></param>
public static void ClearIISExpressProcess(Process process)
{
if (IISExpress.IsRunning(process))
{
process.Kill();
}
}
/// <summary>
/// Used to pick the Hosted IISExpress Dashboard Service URL
/// </summary>
/// <param name="dashboardServicePath"></param>
/// <returns></returns>
private string GetIISExpressServiceUrl(string dashboardServicePath)
{
string availablePort = IISExpress.StartIISExpress(DashboardServiceProcess,
dashboardServicePath);
return "http://localhost:" + availablePort + "/DashboardService.svc";
}
/// <summary>
/// Validate whether Dashboard Service is running in the Url
/// </summary>
/// <param name="dashboardServiceUrl">Dashboard Service Url</param>
/// <returns>returns whether valid dashboard service</returns>
```

```
private static bool ValidateDashboardService(string dashboardServiceUrl)
{
bool errorOccurred = false;
try
{
if (string.IsNullOrWhiteSpace(dashboardServiceUrl))
{
return true;
}
if (!dashboardServiceUrl.Contains("http://") &&
!dashboardServiceUrl.Contains("https://"))
dashboardServiceUrl = "http://" + dashboardServiceUrl + @"/IsServiceExists";
else
dashboardServiceUrl = dashboardServiceUrl + @"/IsServiceExists";
WebRequest request = WebRequest.Create(new Uri(dashboardServiceUrl,
UriKind.Absolute));
request.Method = "GET";
using (WebResponse response = request.GetResponse())
{
using (StreamReader reader = new StreamReader(response.GetResponseStream()))
{
string text = reader.ReadToEnd();
if
(!text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetB
ytes("DashboardServiceExists"))))
{
errorOccurred = true;
}
}
}
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
}
catch (Exception e)
{
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
errorOccurred = true;
}
return errorOccurred;
}
/// <summary>
///     Get Url string for the generated HTML page
/// </summary>
/// <param name="sydxFileName">SYDX file name</param>
/// <param name="serviceUrl">Service URL</param>
/// <returns></returns>
public Process DashboardServiceProcess { get; set; }
public string GetUrlOfHtmlPage(string sydxFileName)
{
if (string.IsNullOrWhiteSpace(sydxFileName) ||
string.IsNullOrWhiteSpace(ServiceUrl))
return string.Empty;
DashboardProperties dashboardProperties = new DashboardProperties
{
SydxFileName = sydxFileName,
ServiceUrl = ServiceUrl,
};
return GetUrlOfHtmlPage(dashboardProperties);
```



```
}
/// <summary>
///     Generate Viewer HTML String
/// </summary>
#region Generate Viewer HTML String
private const string BeforeHeadHtml = "<!-- saved from
url=(0014)about:internet -->\n"
+ "<!DOCTYPE html>\n"
+ "<html xmlns=\"http://www.w3.org/1999/xhtml\" style=\"height :100%
;width:100%;\">\n"
+ "<head>\n"
+ "<meta charset=\"utf-8\" content=\"width=device-width, initial-
scale=1.0\">"
+ "<script>window.destroyAll = function(){try{ej.widget.destroyAll($('.e-
js').off());}catch(e){} $(document.body).html('-'); CollectGarbage(); };
</script>";
/// <summary>
///     Get Url string for the generated html page
/// </summary>
private string GetUrlOfHtmlPage(DashboardProperties dashboardProperties)
{
var jsFiles = new List<string>
{
"jquery-1.10.2.min.js",
"jquery.easing.1.3.min.js",
"ej.dashboardViewer.all.min.js",
};
var cssFiles = new List<string>
{
"default-theme/ej.dashboardViewer.all.min.css",
"chromium.css",
};
try
{
string htmlString = GetHtmlString(GetViewer(dashboardProperties), jsFiles,
cssFiles).ToString();
//Below code block is used to copy the HTML related files from app folder
since app folder doesn't have required permission to write file.
string sourceFolderPath = AppDomain.CurrentDomain.BaseDirectory +
HtmlFolder;
if (!File.Exists(sourceFolderPath))
Directory.CreateDirectory(sourceFolderPath);
string filePath = sourceFolderPath + "Temp.html";
//End
using (FileStream fileStream = new FileStream(filePath, FileMode.Create))
{
using (StreamWriter w = new StreamWriter(fileStream, Encoding.UTF8))
{
w.Write(htmlString);
}
}
return filePath;
}
catch (Exception ex)
{
MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
```

```
}
return string.Empty;
}
/// <summary>
/// Get HTML JavaScript files CSS files.
/// </summary>
private StringBuilder GetHtmlString(string bodyString, IEnumerable<string>
jsFiles, IEnumerable<string> cssFiles)
{
var headSb = new StringBuilder();
foreach (var item in jsFiles)
{
headSb.Append("<script src=\"scripts/" + item + "\"
type=\"text/javascript\"></script>");
}
foreach (var item in cssFiles)
{
headSb.Append("<link href=\"themes/" + item + "\"
rel=\"stylesheet\"></link>");
}
var htmlSb = new StringBuilder();
htmlSb.Append(BeforeHeadHtml);
htmlSb.Append(headSb);
htmlSb.Append("</head>\n");
htmlSb.Append(bodyString);
htmlSb.Append("</html>");
return htmlSb;
}
/// <summary>
///     Generate the Viewer HTML String
/// </summary>
private string GetViewer(DashboardProperties dashboardProperties)
{
if (dashboardProperties == null) return string.Empty;
var sydxPath = _environmentFolder + dashboardProperties.SydxFileName;
var viewerStr = "$('#dashboard').ejDashboardViewer({serviceUrl: '" +
dashboardProperties.ServiceUrl +
"', dashboardPath: '" +
sydxPath.Replace(@"\", @"\\") +
"',filterParameters:location.search.substr(1)});";
var stringBuilder = new StringBuilder();
stringBuilder.Append("<body style=\"width:100%; height:100%;
overflow:hidden;\">");
stringBuilder.Append("<div id=\"dashboard\" style=\"width:100%;
height:100%;\" />");
stringBuilder.Append("<script type=\"text/javascript\"
language=\"javascript\">");
stringBuilder.Append(viewerStr);
stringBuilder.Append("</script>");
stringBuilder.Append("</body>");
return stringBuilder.ToString();
}
#endregion
}
/// <summary>
///     Class for Dashboard Properties
/// </summary>
```



```
public class DashboardProperties
{
/// <summary>
///     Gets or sets the SYDX File Name
/// </summary>
public string SydxFileName { get; set; }
/// <summary>
///     Gets or sets the ServiceUrl
/// </summary>
public string ServiceUrl { get; set; }
}
```

VB.NET

```
Imports System.Collections.Generic
Imports System.IO
Imports System.Security
Imports System.Text
Imports System.Windows
Imports Microsoft.Win32
Imports System.Xml.Serialization
Imports System.Windows.Forms
Imports System.Net.NetworkInformation
Imports System.Net
Imports System.Linq
Imports System.Diagnostics
Imports System.Globalization
''' <summary>
'''     Class for DashboardWindowsServiceInfo
''' </summary>
Public Class DashboardWindowsServiceInfo
#Region "Private Variables"
Private ReadOnly _environmentFolder As String =
AppDomain.CurrentDomain.BaseDirectory
Private Version As String =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
()
Private Const HtmlFolder As String = "Html\"
Public ServiceUrl As String
Public ErrorMessage As String
#End Region
''' <summary>
'''     Constructor for class DashboardWindowsServiceInfo
''' </summary>
Public Sub New(serviceProcess As Process, serviceDirectoryPath As String)
DashboardServiceProcess = serviceProcess
Dim version As String =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
()
'#Region "Pick IISExpress Dashboard Service Url"
ServiceUrl = GetIISExpressServiceUrl(serviceDirectoryPath)
If ValidateDashboardService(ServiceUrl) Then
ServiceUrl = String.Empty
ErrorMessage = "An Error Occurred while hosting Dashboard Service"
MessageBox.Show(ErrorMessage)
Environment.[Exit](0)
```

```
'#End Region
End If
End Sub
''' <summary>
''' used to clear other Process while running IIS Express.
''' </summary>
''' <param name="process"></param>
Public Shared Sub ClearIISExpressProcess(process As Process)
If IISExpress.IsRunning(process) Then
process.Kill()
End If
End Sub
''' <summary>
''' Used to pick the Hosted IISExpress Dashboard Service URL
''' </summary>
''' <param name="dashboardServicePath"></param>
''' <returns></returns>
Private Function GetIISExpressServiceUrl(dashboardServicePath As String) As
String
Dim availablePort As String =
IISExpress.StartIISExpress(DashboardServiceProcess, dashboardServicePath)
Return (Convert.ToString("http://localhost:") & availablePort) +
"/DashboardService.svc"
End Function
''' <summary>
''' Validate whether Dashboard Service is running in the Url
''' </summary>
''' <param name="dashboardServiceUrl">Dashboard Service Url</param>
''' <returns>returns whether valid dashboard service</returns>
Private Shared Function ValidateDashboardService(dashboardServiceUrl As
String) As Boolean
Dim errorOccurred As Boolean = False
Try
If String.IsNullOrWhiteSpace(dashboardServiceUrl) Then
Return True
End If
If Not dashboardServiceUrl.Contains("http://") AndAlso Not
dashboardServiceUrl.Contains("https://") Then
dashboardServiceUrl = (Convert.ToString("http://") & dashboardServiceUrl) +
"/IsServiceExists"
Else
dashboardServiceUrl = dashboardServiceUrl &
Convert.ToString("/IsServiceExists")
End If
Dim request As WebRequest = WebRequest.Create(New Uri(dashboardServiceUrl,
UriKind.Absolute))
request.Method = "GET"
Using response As WebResponse = request.GetResponse()
Using reader As New StreamReader(response.GetResponseStream())
Dim text As String = reader.ReadToEnd()
If Not
text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetByt
es("DashboardServiceExists"))) Then
errorOccurred = True
End If
End Using
End Using
```



```
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
Catch e As Exception
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
errorOccurred = True
End Try
Return errorOccurred
End Function
''' <summary>
'''     Get Url string for the generated HTML page
''' </summary>
''' <param name="sydxFileName">SYDX file name</param>
''' <param name="serviceUrl">Service URL</param>
''' <returns></returns>
Public Property DashboardServiceProcess() As Process
Get
Return m_DashboardServiceProcess
End Get
Set
m_DashboardServiceProcess = Value
End Set
End Property
Private m_DashboardServiceProcess As Process
Public Function GetUrlOfHtmlPage(sydxFileName As String) As String
If String.IsNullOrWhiteSpace(sydxFileName) OrElse
String.IsNullOrWhiteSpace(ServiceUrl) Then
Return String.Empty
End If
Dim dashboardProperties As New DashboardProperties() With {
Key .SydxFileName = sydxFileName,
Key .ServiceUrl = ServiceUrl
}
Return GetUrlOfHtmlPage(dashboardProperties)
End Function
''' <summary>
'''     Generate Viewer HTML String
''' </summary>
#Region "Generate Viewer HTML String"
Private Const BeforeHeadHtml As String = "<!-- saved from
url=(0014)about:internet -->" & vbLf + "<!DOCTYPE html>" & vbLf + "<html
xmlns=""http://www.w3.org/1999/xhtml"" style=""height :100% ;width:100%;"">"
& vbLf + "<head>" & vbLf + "<meta charset=""utf-8"" content=""width=device-
width, initial-scale=1.0"">" + "<script>window.destroyAll =
function(){try{ej.widget.destroyAll($('.e-js').off());}catch(e){}
$(document.body).html('-'); CollectGarbage(); }; </script>"
''' <summary>
'''     Get Url string for the generated HTML page
''' </summary>
Private Function GetUrlOfHtmlPage(dashboardProperties As
DashboardProperties) As String
Dim jsFiles = New List(Of String)() From {
"jquery-1.10.2.min.js",
"jquery.easing.1.3.min.js",
"ej.dashboardViewer.all.min.js"
}
Dim cssFiles = New List(Of String)() From {
"default-theme/ej.dashboardViewer.all.min.css",
"chromium.css"
```



```
}
Try
Dim htmlString As String = GetHtmlString(GetViewer(dashboardProperties),
jsFiles, cssFiles).ToString()
'Below code block is used to copy the HTML related files from app folder
since app folder doesn't have required permission to write file.
Dim sourceFolderPath As String = AppDomain.CurrentDomain.BaseDirectory +
HtmlFolder
If Not File.Exists(sourceFolderPath) Then
Directory.CreateDirectory(sourceFolderPath)
End If
Dim filePath As String = sourceFolderPath & Convert.ToString("Temp.html")
'End
Using fileStream As New FileStream(filePath, FileMode.Create)
Using w As New StreamWriter(fileStream, Encoding.UTF8)
w.Write(htmlString)
End Using
End Using
Return filePath
Catch ex As Exception
MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.[Error])
End Try
Return String.Empty
End Function
''' <summary>
''' Get HTML JavaScript files CSS files.
''' </summary>
Private Function GetHtmlString(bodyString As String, jsFiles As
IEnumerable(Of String), cssFiles As IEnumerable(Of String)) As StringBuilder
Dim headSb = New StringBuilder()
For Each item As var In jsFiles
headSb.Append("<script src=""scripts/" + item + """
type=""text/javascript""></script>")
Next
For Each item As var In cssFiles
headSb.Append("<link href=""themes/" + item + """
rel=""stylesheet""></link>")
Next
Dim htmlSb = New StringBuilder()
htmlSb.Append(BeforeHeadHtml)
htmlSb.Append(headSb)
htmlSb.Append("</head>" & vbLf)
htmlSb.Append(bodyString)
htmlSb.Append("</html>")
Return htmlSb
End Function
''' <summary>
'''     Generate the Viewer HTML String
''' </summary>
Private Function GetViewer(dashboardProperties As DashboardProperties) As
String
If dashboardProperties Is Nothing Then
Return String.Empty
End If
Dim sydxPath = _environmentFolder & dashboardProperties.SydxFileName
```

```
Dim viewerStr =
(Convert.ToString("$('#dashboard').ejDashboardViewer({serviceUrl: '") &
dashboardProperties.ServiceUrl) + "', dashboardPath: '" +
sydxPath.Replace("\", "\\") +
"',filterParameters:location.search.substr(1)});"
Dim stringBuilder = New StringBuilder()
stringBuilder.Append("<body style=""width:100%; height:100%;
overflow:hidden;"">")
stringBuilder.Append("<div id=""dashboard"" style=""width:100%;
height:100%;"" />")
stringBuilder.Append("<script type=""text/javascript""
language=""javascript"">")
stringBuilder.Append(viewerStr)
stringBuilder.Append("</script>")
stringBuilder.Append("</body>")
Return stringBuilder.ToString()
End Function
#End Region
End Class
''' <summary>
'''     Class for Dashboard Properties
''' </summary>
Public Class DashboardProperties
''' <summary>
'''     Gets or sets the SYDX File Name
''' </summary>
Public Property SydxFileName() As String
Get
Return m_SydxFileName
End Get
Set
m_SydxFileName = Value
End Set
End Property
Private m_SydxFileName As String
''' <summary>
'''     Gets or sets the ServiceUrl
''' </summary>
Public Property ServiceUrl() As String
Get
Return m_ServiceUrl
End Get
Set
m_ServiceUrl = Value
End Set
End Property
Private m_ServiceUrl As String
End Class
```

Add a class named DashboardServiceSerialization and add the below code withing the class to serialize and deserialize the DashboardService URL when Dashboard Service is running in IIS Express.

C#

```
using System;
using System.Collections.Generic;
```

```
using System.IO;
using System.Linq;
using System.Web;
using System.Xml.Serialization;
public class DashboardServiceSerialization
{
static readonly XmlSerializer previewSerializer = new
XmlSerializer(typeof(DashboardServicePreviewSettings));
public void Serialize(DashboardServicePreviewSettings settings, string path)
{
try
{
using (StreamWriter writer = new StreamWriter(path))
{
previewSerializer.Serialize(writer, settings);
}
}
catch (Exception)
{
}
}
public DashboardServicePreviewSettings Deserialize(string path)
{
DashboardServicePreviewSettings settings = new
DashboardServicePreviewSettings();
try
{
using (StreamReader reader = new StreamReader(path))
{
settings =
(DashboardServicePreviewSettings)previewSerializer.Deserialize(reader);
}
}
catch (Exception)
{
}
return settings;
}
}
public class DashboardServicePreviewSettings
{
public string ServiceURL { get; set; }
public List<Guid> DashboardServiceInstances { get; set; }
public DashboardServicePreviewSettings()
{
DashboardServiceInstances = new List<Guid>();
}
}
```

VB.NET

```
Imports System.Collections.Generic
Imports System.IO
Imports System.Linq
Imports System.Web
Imports System.Xml.Serialization
```

```
Public Class DashboardServiceSerialization
Shared ReadOnly previewSerializer As New
XmlSerializer(GetType(DashboardServicePreviewSettings))
Public Sub Serialize(settings As DashboardServicePreviewSettings, path As
String)
Try
Using writer As New StreamWriter(path)
previewSerializer.Serialize(writer, settings)
End Using
Catch generatedExceptionName As Exception
End Try
End Sub
Public Function Deserialize(path As String) As
DashboardServicePreviewSettings
Dim settings As New DashboardServicePreviewSettings()
Try
Using reader As New StreamReader(path)
settings = DirectCast(previewSerializer.Deserialize(reader),
DashboardServicePreviewSettings)
End Using
Catch generatedExceptionName As Exception
End Try
Return settings
End Function
End Class
Public Class DashboardServicePreviewSettings
Public Property ServiceURL() As String
Get
Return m_ServiceURL
End Get
Set
m_ServiceURL = Value
End Set
End Property
Private m_ServiceURL As String
Public Property DashboardServiceInstances() As List(Of Guid)
Get
Return m_DashboardServiceInstances
End Get
Set
m_DashboardServiceInstances = Value
End Set
End Property
Private m_DashboardServiceInstances As List(Of Guid)
Public Sub New()
DashboardServiceInstances = New List(Of Guid)()
End Sub
End Class
```

Add a class named IISExpress and add the below code withing the class.

C#

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
```

```
using System.Globalization;
using System.Linq;
using System.Net;
using System.Net.NetworkInformation;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
public static class IISExpress
{
internal static string StartIISExpress(Process iisExpressProcess, string
location)
{
string availablePort = GetAvailablePort();
string iisInstalledPath = GetIISExpressInstalledPath();
try
{
if (!string.IsNullOrEmpty(iisInstalledPath))
{
iisExpressProcess.StartInfo = new ProcessStartInfo { FileName =
iisInstalledPath + "iisexpress.exe", WindowStyle =
ProcessWindowStyle.Hidden, Arguments = @"/path:""" + location + "\" /port:"
+ availablePort, UseShellExecute = false, CreateNoWindow = true };
iisExpressProcess.Start();
return availablePort;
}
}
catch (Exception e)
{
if (e.HResult == -2147467259)
{
MessageBox.Show("IIS Express file is not found", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
else
MessageBox.Show(e.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
return string.Empty;
}
private static string GetAvailablePort()
{
int startIndex = 3002, endIndex = 9000, unusedPort = 0;
IPGlobalProperties properties = IPGlobalProperties.GetIPGlobalProperties();
IPEndPoint[] tcpEndpoints = properties.GetActiveTcpListeners();
List<int> usedPorts = tcpEndpoints.Select(p => p.Port).ToList();
for (int port = startIndex; port < endIndex; port++)
{
if (!usedPorts.Contains(port))
{
unusedPort = port;
break;
}
}
return unusedPort.ToString(CultureInfo.InvariantCulture);
}
private static string GetIISExpressInstalledPath()
{
```

```
return Environment.GetEnvironmentVariable("PROGRAMFILES") + @"\IIS
Express\";
}
internal static bool IsRunning(Process process)
{
try { Process.GetProcessById(process.Id); }
catch (InvalidOperationException) { return false; }
catch (ArgumentException) { return false; }
return true;
}
}
```

VB.NET

```
Imports System.Collections.Generic
Imports System.Diagnostics
Imports System.Globalization
Imports System.Linq
Imports System.Net
Imports System.Net.NetworkInformation
Imports System.Text
Imports System.Threading.Tasks
Imports System.Windows.Forms
Public NotInheritable Class IISExpress
Private Sub New()
End Sub
Friend Shared Function StartIISExpress(iisExpressProcess As Process,
location As String) As String
Dim availablePort As String = GetAvailablePort()
Dim iisInstalledPath As String = GetIISExpressInstalledPath()
Try
If Not String.IsNullOrEmpty(iisInstalledPath) Then
iisExpressProcess.StartInfo = New ProcessStartInfo() With {
Key .FileName = iisInstalledPath & Convert.ToString("iisexpress.exe"),
Key .WindowStyle = ProcessWindowStyle.Hidden,
Key .Arguments = Convert.ToString((Convert.ToString("/path:""") & location)
+ """ /port:") & availablePort,
Key .UseShellExecute = False,
Key .CreateNoWindow = True
}
iisExpressProcess.Start()
Return availablePort
End If
Catch e As Exception
If e.HResult = -2147467259 Then
MessageBox.Show("IIS Express file is not found", "Error",
MessageBoxButtons.OK, MessageBoxIcon.[Error])
Else
MessageBox.Show(e.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.[Error])
End If
End Try
Return String.Empty
End Function
Private Shared Function GetAvailablePort() As String
```



```
Dim startIndex As Integer = 3002, endIndex As Integer = 9000, unusedPort As
Integer = 0
Dim properties As IPGlobalProperties =
IPGlobalProperties.GetIPGlobalProperties()
Dim tcpEndpoints As IPEndPoint() = properties.GetActiveTcpListeners()
Dim usedPorts As List(Of Integer) = tcpEndpoints.[Select](Function(p)
p.Port).ToList()
For port As Integer = startIndex To endIndex - 1
If Not usedPorts.Contains(port) Then
unusedPort = port
Exit For
End If
Next
Return unusedPort.ToString(CultureInfo.InvariantCulture)
End Function
Private Shared Function GetIISExpressInstalledPath() As String
Return Environment.GetEnvironmentVariable("PROGRAMFILES") + "\IIS Express\"
End Function
Friend Shared Function IsRunning(process__1 As Process) As Boolean
Try
Process.GetProcessById(process__1.Id)
Catch generatedExceptionName As InvalidOperationException
Return False
Catch generatedExceptionName As ArgumentException
Return False
End Try
Return True
End Function
End Class
```

Finally set the serviceDirectory property value in Form1.cs file. service which is shipped along with the SDK build. you can find the service directory in below location in your machine.

`%localappdata%\Syncfusion\Dashboard Platform SDK\Service`

for example,replace the serviceDirectory property value as below in your project's Form1.cs file,

C#

```
private string serviceDirectory = "%localappdata%\Syncfusion\Dashboard
Platform SDK\Service";
```

VB.NET

```
Private serviceDirectory As String = "%localappdata%\Syncfusion\Dashboard
Platform SDK\Service"
```

Adding URL to Forms

Include the following code under `Form1` class in `Form1.Designer.cs` file to render the generated URL in Windows Form.

C#

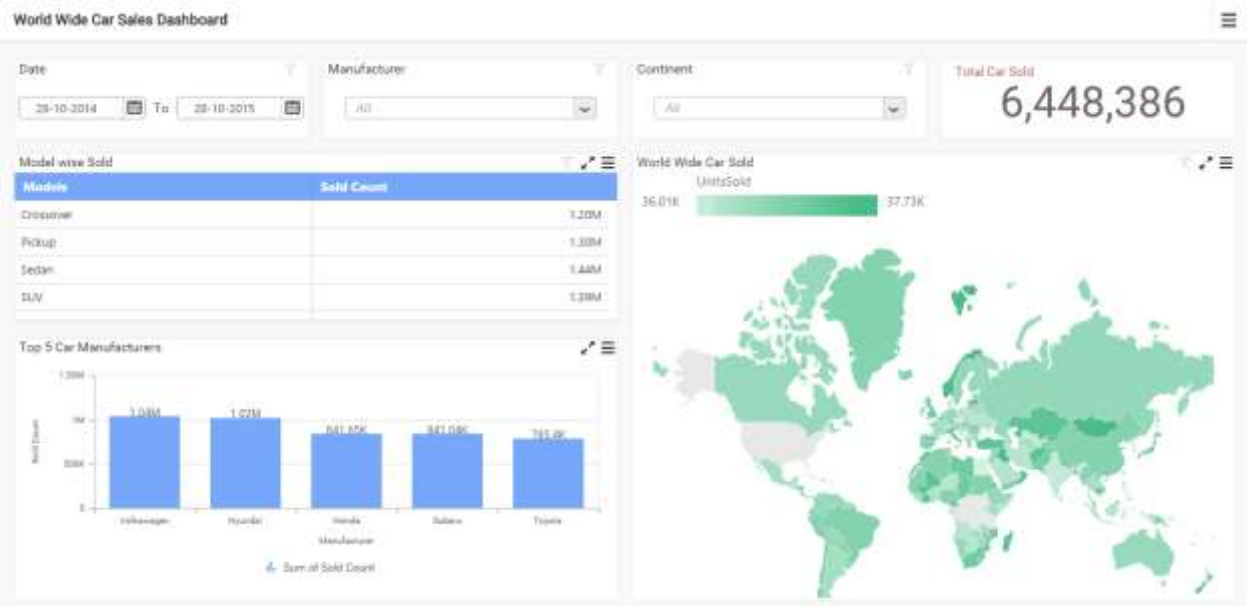
```
private void InitializeComponent()
{
```

```
System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
this.panel1 = new System.Windows.Forms.Panel();
this.WindowBrowser = new CefSharp.WinForms.ChromiumWebBrowser(Url);
this.panel1.SuspendLayout();
this.SuspendLayout();
//
// panel1
//
this.panel1.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.To
p | System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.panel1.BackColor = System.Drawing.Color.White;
this.panel1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
this.panel1.Controls.Add(this.WindowBrowser);
this.panel1.Location = new System.Drawing.Point(4, 12);
this.panel1.Name = "panel1";
this.panel1.Padding = new System.Windows.Forms.Padding(5);
this.panel1.Size = new System.Drawing.Size(966, 468);
this.panel1.TabIndex = 0;
//
// WindowBrowser
//
this.WindowBrowser.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.To
p | System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.WindowBrowser.Location = new System.Drawing.Point(8, 8);
this.WindowBrowser.MinimumSize = new System.Drawing.Size(20, 20);
this.WindowBrowser.Name = "WindowBrowser";
this.WindowBrowser.Size = new System.Drawing.Size(948, 451);
this.WindowBrowser.TabIndex = 0;
//
// DashboardWindow
//
this.AutoScaleDimensions = new System.Drawing.SizeF(9F, 18F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(974, 484);
this.Controls.Add(this.panel1);
this.Font = new System.Drawing.Font("Arial", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(0)));
this.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(96)))),
((int)(((byte)(96)))), ((int)(((byte)(96)))));
this.Margin = new System.Windows.Forms.Padding(4);
this.Name = "Form1";
this.Text = "Dashboard Preview Window";
this.WindowState = System.Windows.Forms.FormWindowState.Maximized;
this.panel1.ResumeLayout(false);
this.ResumeLayout(false);
}
private System.Windows.Forms.Panel panel1;
private CefSharp.WinForms.ChromiumWebBrowser WindowBrowser;
```

VB.NET

```
Private Sub InitializeComponent()
Dim resources As System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(Form1))
Me.panel1 = New System.Windows.Forms.Panel()
Me.WindowBrowser = New CefSharp.WinForms.ChromiumWebBrowser(Url)
Me.panel1.SuspendLayout()
Me.SuspendLayout()
Me.panel1.Anchor = (CType(((((System.Windows.Forms.AnchorStyles.Top Or
System.Windows.Forms.AnchorStyles.Bottom) Or
System.Windows.Forms.AnchorStyles.Left) Or
System.Windows.Forms.AnchorStyles.Right)),
System.Windows.Forms.AnchorStyles))
Me.panel1.BackColor = System.Drawing.Color.White
Me.panel1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle
Me.panel1.Controls.Add(Me.WindowBrowser)
Me.panel1.Location = New System.Drawing.Point(4, 12)
Me.panel1.Name = "panel1"
Me.panel1.Padding = New System.Windows.Forms.Padding(5)
Me.panel1.Size = New System.Drawing.Size(966, 468)
Me.panel1.TabIndex = 0
Me.WindowBrowser.Anchor = (CType(((((System.Windows.Forms.AnchorStyles.Top
Or System.Windows.Forms.AnchorStyles.Bottom) Or
System.Windows.Forms.AnchorStyles.Left) Or
System.Windows.Forms.AnchorStyles.Right)),
System.Windows.Forms.AnchorStyles))
Me.WindowBrowser.Location = New System.Drawing.Point(8, 8)
Me.WindowBrowser.MinimumSize = New System.Drawing.Size(20, 20)
Me.WindowBrowser.Name = "WindowBrowser"
Me.WindowBrowser.Size = New System.Drawing.Size(948, 451)
Me.WindowBrowser.TabIndex = 0
Me.AutoScaleDimensions = New System.Drawing.SizeF(9F, 18F)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.ClientSize = New System.Drawing.Size(974, 484)
Me.Controls.Add(Me.panel1)
Me.Font = New System.Drawing.Font("Arial", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(CByte((0))))
Me.ForeColor = System.Drawing.Color.FromArgb((CInt(((CByte((96)))))),
(CInt(((CByte((96)))))), (CInt(((CByte((96)))))))
Me.Margin = New System.Windows.Forms.Padding(4)
Me.Name = "Form1"
Me.Text = "Dashboard Preview Window"
Me.WindowState = System.Windows.Forms.FormWindowState.Maximized
Me.panel1.ResumeLayout(False)
Me.ResumeLayout(False)
End Sub
Private panel1 As System.Windows.Forms.Panel
Private WindowBrowser As CefSharp.WinForms.ChromiumWebBrowser
```

Run the application to view the dashboard.



Getting Started with WPF Application

This section describes how to create a WPF application with embedded dashboard viewer.

Project Creation

Create a new WPF Application Project using Microsoft Visual Studio IDE 2012 or higher.

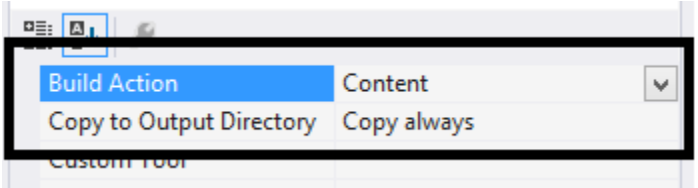
Adding files and references

Add the scripts, styles and refer fonts that are required for the dashboard from the following location to the application project.

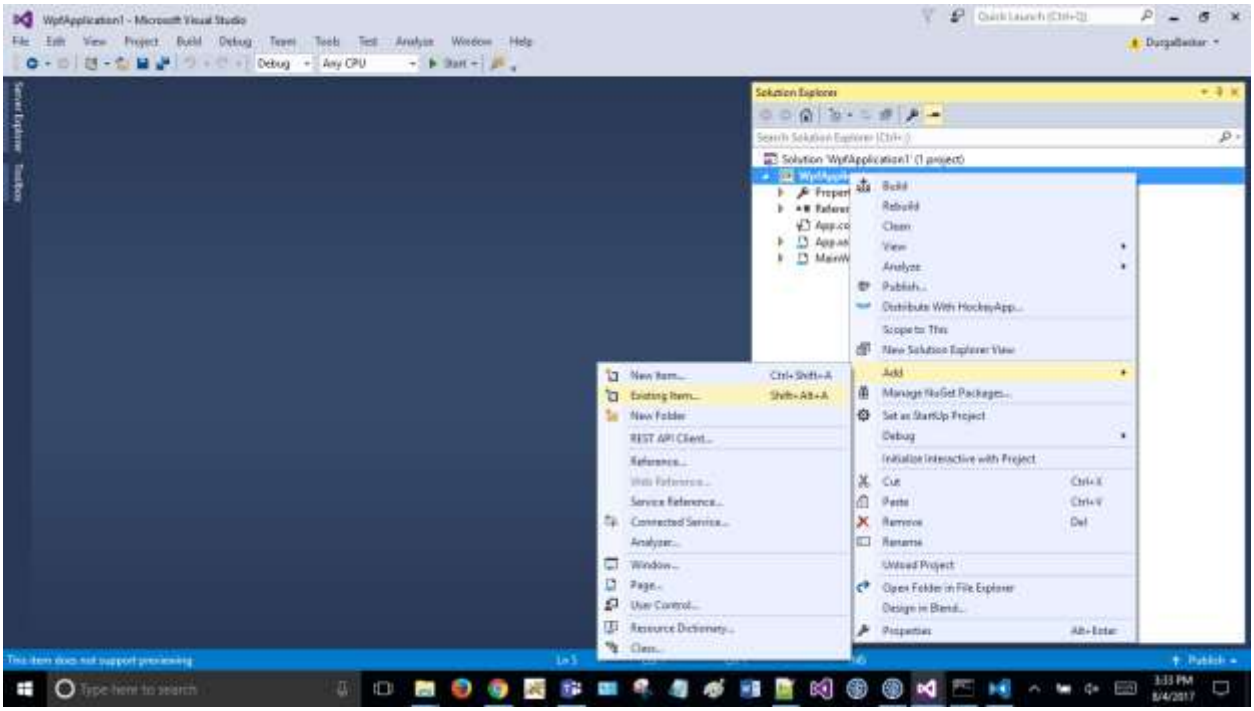
`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html`

Include the dashboard file (*.sydx) in the project from the below mentioned location and set the `Build Action` and `Copy to Output Directory` properties to `Content` and `Copy always`, respectively as shown in below image.

`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Dashboards`



Include the below mentioned files into project from the SDK build installed location as shown in figure below.

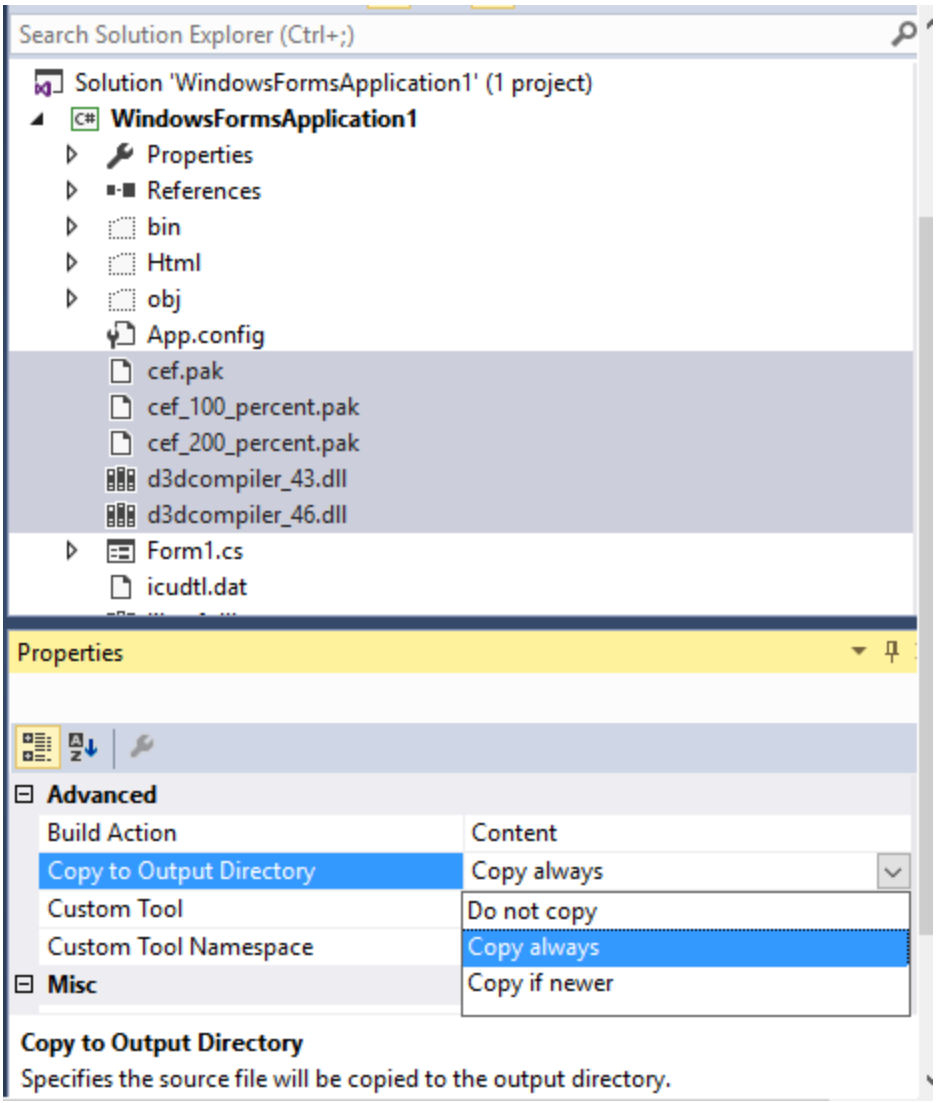


- cef.pak
- cef*100*percent.pak
- cef*200*percent.pak
- d3dcompiler_43.dll
- d3dcompiler_46.dll
- icudtl.dat
- libcef.dll.dll
- libEGL.dll
- libGLESv2.dll

The above mentioned files can be found in the following Dashboard SDK samples location:

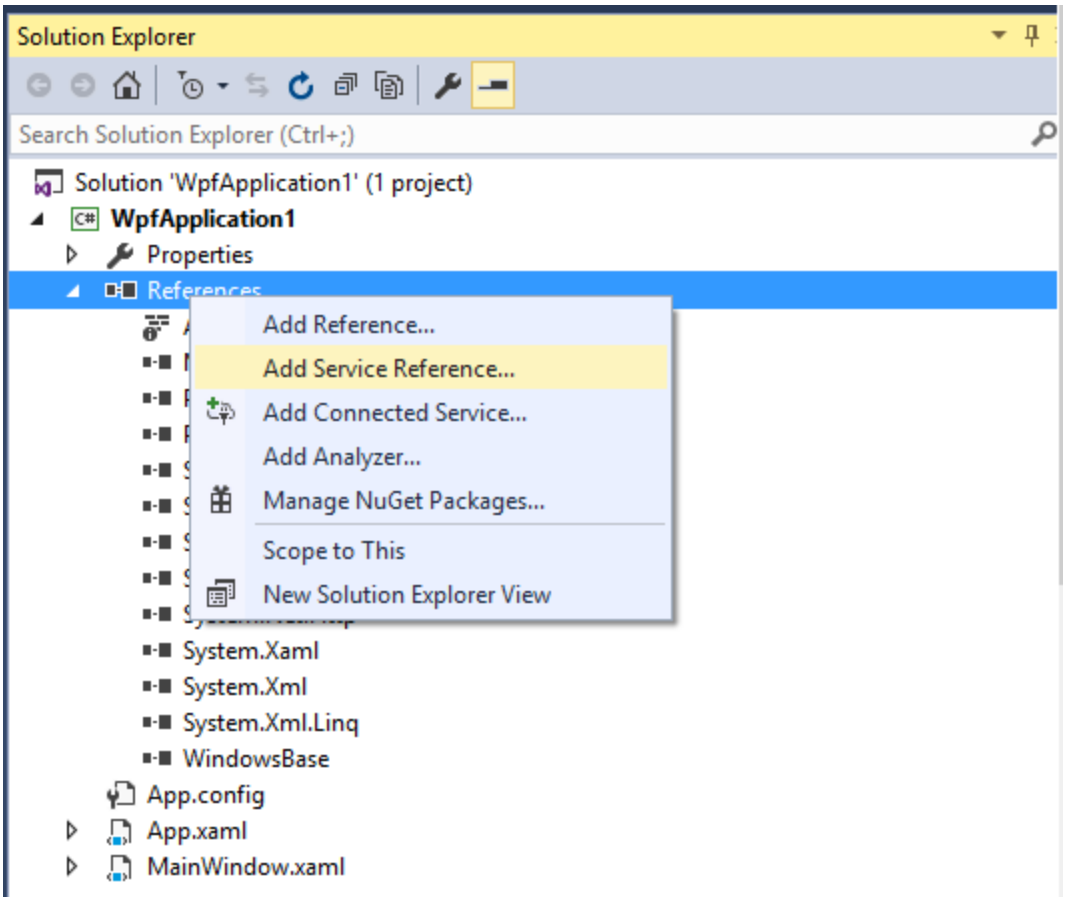
`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Packages`

Set the `Build Action` property to `Content` and the `Copy to Output Directory` property to `Copy always` for all added files as shown in the following image.



Adding Dashboard Viewer Assembly References

Right-click the project and add the following assembly references through choosing Add > Reference... as shown below,

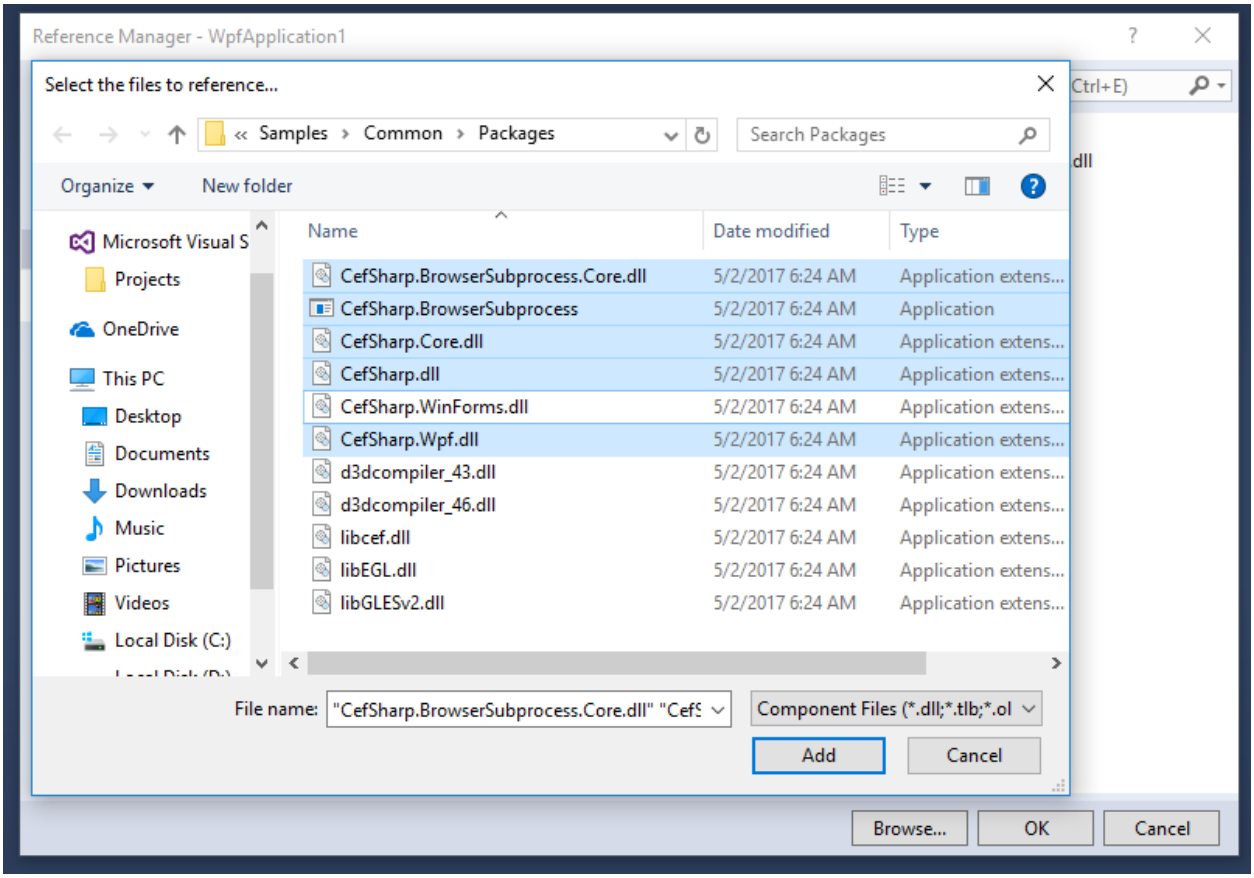


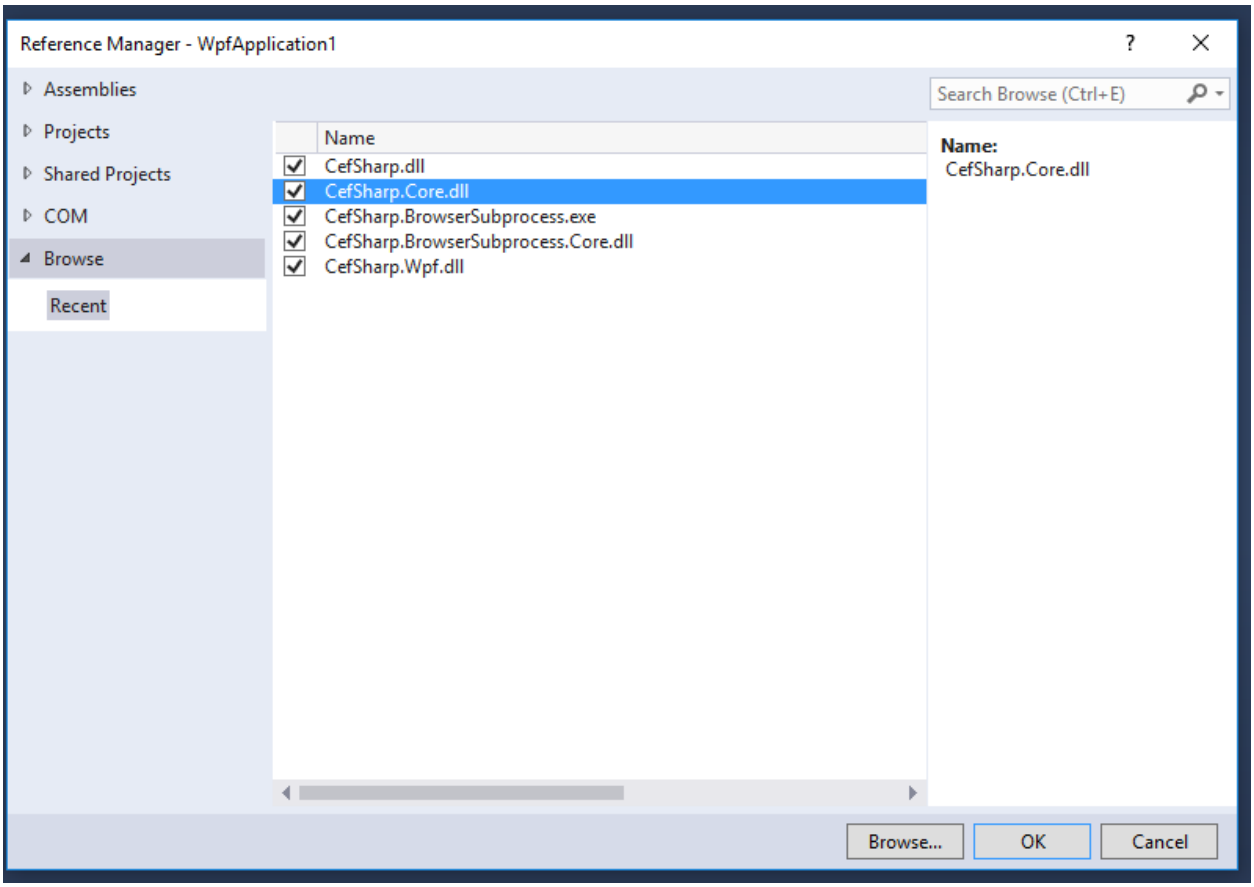
selecting the below mentioned assemblies in dialog shown, which allows you to use any of the Syncfusion Windows Form control within it.

- CefSharp.Core.dll
- CefSharp.Wpf.dll
- CefSharp.dll
- CefSharp.BrowserSubprocess.dll
- CefSharp.BrowserSubprocess.Core.dll

The above mentioned assembly files can be found in the following Dashboard SDK samples location:

`%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Packages`





Control Initialization

Add the below code under constructor method of App.xaml class file so it will execute before initializing the ChromiumWebBrowser control.

C#

```
var settings = new CefSettings();
settings.PackLoadingDisabled = true;
settings.LogSeverity = LogSeverity.Disable;
if (!Cef.Initialize(settings))
{
if (Environment.GetCommandLineArgs().Contains("--type=renderer"))
{
Environment.Exit(0);
}
else
{
return;
}
}
```

VB.NET

```
Dim settings = New CefSettings()
settings.PackLoadingDisabled = True
settings.LogSeverity = LogSeverity.Disable
If Not CefSharp.Cef.Initialize(settings) Then
```



```
If Environment.GetCommandLineArgs().Contains("--type=renderer") Then
Environment.Exit(0)
Else
Return
End If
End If
```

Add the below code under the DashboardWindow.xaml class file.

C#

```
using CefSharp;
using System.Diagnostics;
using System.IO;
using System.Windows;
using System.ComponentModel;
/// <summary>
/// Interaction logic for DashboardWindow.xaml
/// </summary>
public partial class DashboardWindow : Window
{
private string _sydxFileName = "WorldWideCarSalesDashboard.sydx";
private string url = string.Empty;
public string SydxFileName
{
get { return _sydxFileName; }
set { _sydxFileName = value; }
}
public class DownloadHandler : IDownloadHandler
{
public bool OnBeforeDownload(DownloadItem downloadItem, out string
downloadPath, out bool showDialog)
{
downloadPath = downloadItem.SuggestedFileName;
showDialog = true;
return true;
}
public bool OnDownloadUpdated(DownloadItem downloadItem)
{
return false;
}
}
internal static Process dashboardServiceProcess = new Process();
public string Url { get; set; }
// Get the service from Common Folder
private string serviceDirectory =
Path.GetFullPath(Path.Combine(@"..\..\..\..\")) + "Common\\Service";
/// <summary>
/// Initializing Dashboard on  window and closing IIS Express while closing
dashboard window.
/// </summary>
public DashboardWindow()
{
this.Closing += DashboardWindow_closing;
InitializeComponent();
this.Loaded += DashboardWindow_Loaded;
```



```
WindowBrowser.DownloadHandler = new DownloadHandler();
Navigate();
}
/// <summary>
/// Close IIS Express Process while closing Window.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void DashboardWindow_closing(object sender, CancelEventArgs e)
{
DashboardWindowsServiceInfo.ClearIISExpressProcess(dashboardServiceProcess);
}
void DashboardWindow_Loaded(object sender, System.Windows.RoutedEventArgs e)
{
if (string.IsNullOrWhiteSpace(url))
this.Close();
}
private void Navigate()
{
DashboardWindowsServiceInfo dashboardViewer = new
DashboardWindowsServiceInfo(dashboardServiceProcess, serviceDirectory);
url = dashboardViewer.GetUrlOfHtmlPage(SydxFileName);
if (!string.IsNullOrWhiteSpace(url))
WindowBrowser.Address = url;
}
/// <summary>
/// Generate the serviceUrl to render Dashboard
/// </summary>
private bool GenerateUrl()
{
DashboardWindowsServiceInfo dashboardViewer = new
DashboardWindowsServiceInfo(dashboardServiceProcess, serviceDirectory);
string url = dashboardViewer.GetUrlOfHtmlPage(SydxFileName);
if (!string.IsNullOrWhiteSpace(url))
{
Url = url;
return true;
}
else
return false;
}
}
```

VB.NET

```
Imports CefSharp
Imports System.Diagnostics
Imports System.IO
Imports System.Windows
Imports System.ComponentModel
''' <summary>
''' Interaction logic for DashboardWindow.xaml
''' </summary>
Public Partial Class DashboardWindow
Inherits Window
Private _sydxFileName As String = "WorldWideCarSalesDashboard.sydx"
```

```
Private m_url As String = String.Empty
Public Property SydxFileName() As String
Get
Return _sydxFileName
End Get
Set
_sydxFileName = value
End Set
End Property
Public Class DownloadHandler
Implements IDownloadHandler
Public Function OnBeforeDownload(downloadItem As DownloadItem, ByRef
downloadPath As String, ByRef showDialog As Boolean) As Boolean
downloadPath = downloadItem.SuggestedFileName
showDialog = True
Return True
End Function
Public Function OnDownloadUpdated(downloadItem As DownloadItem) As Boolean
Return False
End Function
End Class
Friend Shared dashboardServiceProcess As New Process()
Public Property Url() As String
Get
Return m_Url
End Get
Set
m_Url = Value
End Set
End Property
Private m_Url As String
' Get the service from Common Folder
Private serviceDirectory As String =
Path.GetFullPath(Path.Combine("..\..\..\..\")) + "Common\Service"
''' <summary>
''' Initializing Dashboard on  window and closing IIS Express while closing
dashboard window.
''' </summary>
Public Sub New()
AddHandler Me.Closing, AddressOf DashboardWindow_closing
InitializeComponent()
AddHandler Me.Loaded, AddressOf DashboardWindow_Loaded
WindowBrowser.DownloadHandler = New DownloadHandler()
Navigate()
End Sub
''' <summary>
''' Close IIS Express Process while closing Window.
''' </summary>
''' <param name="sender"></param>
''' <param name="e"></param>
Private Sub DashboardWindow_closing(sender As Object, e As CancelEventArgs)
DashboardWindowsServiceInfo.ClearIISExpressProcess(dashboardServiceProcess)
End Sub
Private Sub DashboardWindow_Loaded(sender As Object, e As
System.Windows.RoutedEventArgs)
If String.IsNullOrWhiteSpace(m_url) Then
Me.Close()
```



```
End If
End Sub
Private Sub Navigate()
Dim dashboardViewer As New
DashboardWindowsServiceInfo(dashboardServiceProcess, serviceDirectory)
m_url = dashboardViewer.GetUrlOfHtmlPage(SydxFileName)
If Not String.IsNullOrWhiteSpace(m_url) Then
WindowBrowser.Address = m_url
End If
End Sub
''' <summary>
''' Generate the serviceUrl to render Dashboard
''' </summary>
Private Function GenerateUrl() As Boolean
Dim dashboardViewer As New
DashboardWindowsServiceInfo(dashboardServiceProcess, serviceDirectory)
Dim url__1 As String = dashboardViewer.GetUrlOfHtmlPage(SydxFileName)
If Not String.IsNullOrWhiteSpace(url__1) Then
Url = url__1
Return True
Else
Return False
End If
End Function
End Class
```

Create a class named DashboardWindowsServiceInfo and add the below code within the class.

C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Security;
using System.Text;
using System.Windows;
using Microsoft.Win32;
using System.Xml.Serialization;
using System.Diagnostics;
using System.Net.NetworkInformation;
using System.Net;
using System.Linq;
using System.Globalization;
/// <summary>
///     Class for Dashboard Viewer
/// </summary>
public class DashboardWindowsServiceInfo
{
#region Private Variables
private readonly string _environmentFolder =
AppDomain.CurrentDomain.BaseDirectory;
string Version =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
();
private const string HtmlFolder = "Html\\";
public string ServiceUrl;
```

```
public string ErrorMessage;
#endregion
/// <summary>
///     Constructor for class DashboardWindowsServiceInfo
/// </summary>
///
public DashboardWindowsServiceInfo(Process serviceProcess, string
serviceDirectoryPath)
{
DashboardServiceProcess = serviceProcess;
string version =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
();
#region Pick IISExpress Dashboard Service Url
ServiceUrl = GetIISExpressServiceUrl(serviceDirectoryPath);
if (ValidateDashboardService(ServiceUrl))
{
ServiceUrl = string.Empty;
ErrorMessage = "An Error Occurred while hosting Dashboard Service";
MessageBox.Show(ErrorMessage);
Environment.Exit(0);
}
#endregion
}
/// <summary>
/// used to clear other Process while running IIS Express.
/// </summary>
/// <param name="process"></param>
public static void ClearIISExpressProcess(Process process)
{
if (IISExpress.IsRunning(process))
{
process.Kill();
}
}
/// <summary>
/// Used to pick the Hosted IISExpress Dashboard Service URL
/// </summary>
/// <param name="dashboardServicePath"></param>
/// <returns></returns>
private string GetIISExpressServiceUrl(string dashboardServicePath)
{
string availablePort = IISExpress.StartIISExpress(DashboardServiceProcess,
dashboardServicePath);
return "http://localhost:" + availablePort + "/DashboardService.svc";
}
/// <summary>
/// <summary>
/// Validate whether Dashboard Service is running in the Url
/// </summary>
/// <param name="dashboardServiceUrl">Dashboard Service Url</param>
/// <returns>returns whether valid dashboard service</returns>
private static bool ValidateDashboardService(string dashboardServiceUrl)
{
bool errorOccurred = false;
try
{
```

```
if (string.IsNullOrWhiteSpace(dashboardServiceUrl))
{
return true;
}
if (!dashboardServiceUrl.Contains("http://") &&
!dashboardServiceUrl.Contains("https://"))
dashboardServiceUrl = "http://" + dashboardServiceUrl + @"/IsServiceExists";
else
dashboardServiceUrl = dashboardServiceUrl + @"/IsServiceExists";
WebRequest request = WebRequest.Create(new Uri(dashboardServiceUrl,
UriKind.Absolute));
request.Method = "GET";
using (WebResponse response = request.GetResponse())
{
using (StreamReader reader = new StreamReader(response.GetResponseStream()))
{
string text = reader.ReadToEnd();
if
(!text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetB
ytes("DashboardServiceExists"))))
{
errorOccurred = true;
}
}
}
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
}
catch (Exception e)
{
dashboardServiceUrl = dashboardServiceUrl.Replace(@"/IsServiceExists", "");
errorOccurred = true;
}
return errorOccurred;
}
public Process DashboardServiceProcess { get; set; }
/// <summary>
///     Get Url string for the generated HTML page
/// </summary>
/// <param name="sydxFileName">SYDX file name</param>
/// <returns></returns>
public string GetUrlOfHtmlPage(string sydxFileName)
{
if (string.IsNullOrWhiteSpace(sydxFileName) ||
string.IsNullOrWhiteSpace(ServiceUrl))
return string.Empty;
DashboardProperties dashboardProperties = new DashboardProperties
{
SydxFileName = sydxFileName,
ServiceUrl = ServiceUrl,
};
return GetUrlOfHtmlPage(dashboardProperties);
}
/// <summary>
///     Generate Viewer HTML String
/// </summary>
#region Generate Viewer HTML String
```

```
private const string BeforeHeadHtml = "<!-- saved from
url=(0014)about:internet -->\n"
+ "<!DOCTYPE html>\n"
+ "<html xmlns=\"http://www.w3.org/1999/xhtml\" style=\"height :100%
;width:100%;\">\n"
+ "<head>\n"
+ "<meta charset=\"utf-8\" content=\"width=device-width, initial-
scale=1.0\">"
+ "<script>window.destroyAll = function(){try{ej.widget.destroyAll($('.e-
js').off());}catch(e){} $(document.body).html('-'); CollectGarbage(); };
</script>";
/// <summary>
///     Get Url string for the generated HTML page
/// </summary>
private string GetUrlOfHtmlPage(DashboardProperties dashboardProperties)
{
var jsFiles = new List<string>
{
"jquery-1.10.2.min.js",
"jquery.easing.1.3.min.js",
"ej.dashboardViewer.all.min.js",
};
var cssFiles = new List<string>
{
"default-theme/ej.dashboardViewer.all.min.css",
"chromium.css",
};
try
{
string htmlString = GetHtmlString(GetViewer(dashboardProperties), jsFiles,
cssFiles).ToString();
//Below code block is used to copy the HTML related files from app folder
since app folder doesn't have required permission to write file.
string sourceFolderPath = AppDomain.CurrentDomain.BaseDirectory +
HtmlFolder;
if (!File.Exists(sourceFolderPath))
Directory.CreateDirectory(sourceFolderPath);
string filePath = sourceFolderPath + "Temp.html";
//End
using (FileStream fileStream = new FileStream(filePath, FileMode.Create))
{
using (StreamWriter w = new StreamWriter(fileStream, Encoding.UTF8))
{
w.Write(htmlString);
}
}
return filePath;
}
catch (Exception ex)
{
MessageBox.Show(ex.Message, "Error", MessageBoxButton.OK,
MessageBoxImage.Error);
}
return string.Empty;
}
/// <summary>
/// Get HTML JavaScript files CSS files.
```

```
/// </summary>
private StringBuilder GetHtmlString(string bodyString, IEnumerable<string>
jsFiles, IEnumerable<string> cssFiles)
{
var headSb = new StringBuilder();
foreach (var item in jsFiles)
{
headSb.Append("<script src=\"scripts/" + item + "\"
type=\"text/javascript\"></script>");
}
foreach (var item in cssFiles)
{
headSb.Append("<link href=\"themes/" + item + "\"
rel=\"stylesheet\"></link>");
}
var htmlSb = new StringBuilder();
htmlSb.Append(BeforeHeadHtml);
htmlSb.Append(headSb);
htmlSb.Append("</head>\n");
htmlSb.Append(bodyString);
htmlSb.Append("</html>");
return htmlSb;
}
/// <summary>
///     Generate Viewer HTML String
/// </summary>
private string GetViewer(DashboardProperties dashboardProperties)
{
if (dashboardProperties == null) return string.Empty;
var sydxPath = _environmentFolder + dashboardProperties.SydxFileName;
var viewerStr = "$('#dashboard').ejDashboardViewer({serviceUrl: '" +
dashboardProperties.ServiceUrl +
"', dashboardPath: '" +
sydxPath.Replace(@"\", @"\\") +
"',filterParameters:location.search.substr(1)});";
var stringBuilder = new StringBuilder();
stringBuilder.Append("<body style=\"width:100%; height:100%;
overflow:hidden;\">");
stringBuilder.Append("<div id=\"dashboard\" style=\"width:100%;
height:100%;\" />");
stringBuilder.Append("<script type=\"text/javascript\"
language=\"javascript\">");
stringBuilder.Append(viewerStr);
stringBuilder.Append("</script>");
stringBuilder.Append("</body>");
return stringBuilder.ToString();
}
#endregion
}
/// <summary>
///     Class for Dashboard Properties
/// </summary>
public class DashboardProperties
{
/// <summary>
///     Gets or sets the SYDX File Name
/// </summary>
```

```
public string SydxFileName { get; set; }
/// <summary>
///     Gets or sets the ServiceUrl
/// </summary>
public string ServiceUrl { get; set; }
}
```

VB.NET

```
Imports System.Collections.Generic
Imports System.IO
Imports System.Security
Imports System.Text
Imports System.Windows
Imports Microsoft.Win32
Imports System.Xml.Serialization
Imports System.Diagnostics
Imports System.Net.NetworkInformation
Imports System.Net
Imports System.Linq
Imports System.Globalization
''' <summary>
'''     Class for Dashboard Viewer
''' </summary>
Public Class DashboardWindowsServiceInfo
#Region "Private Variables"
Private ReadOnly _environmentFolder As String =
AppDomain.CurrentDomain.BaseDirectory
Private Version As String =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
()
Private Const HtmlFolder As String = "Html\"
Public ServiceUrl As String
Public ErrorMessage As String
#End Region
''' <summary>
'''     Constructor for class DashboardWindowsServiceInfo
''' </summary>
'''
Public Sub New(serviceProcess As Process, serviceDirectoryPath As String)
DashboardServiceProcess = serviceProcess
Dim version As String =
System.Reflection.Assembly.GetExecutingAssembly().GetName().Version.ToString
()
'#Region "Pick IISExpress Dashboard Service Url"
ServiceUrl = GetIISExpressServiceUrl(serviceDirectoryPath)
If ValidateDashboardService(ServiceUrl) Then
ServiceUrl = String.Empty
ErrorMessage = "An Error Occurred while hosting Dashboard Service"
MessageBox.Show(ErrorMessage)
Environment.[Exit](0)
'#End Region
End If
End Sub
''' <summary>
''' used to clear other Process while running IIS Express.
```

```
''' </summary>
''' <param name="process"></param>
Public Shared Sub ClearIISExpressProcess(process As Process)
If IISExpress.IsRunning(process) Then
process.Kill()
End If
End Sub
''' <summary>
''' Used to pick the Hosted IISExpress Dashboard Service URL
''' </summary>
''' <param name="dashboardServicePath"></param>
''' <returns></returns>
Private Function GetIISExpressServiceUrl(dashboardServicePath As String) As
String
Dim availablePort As String =
IISExpress.StartIISExpress(DashboardServiceProcess, dashboardServicePath)
Return (Convert.ToString("http://localhost:") & availablePort) +
"/DashboardService.svc"
End Function
''' <summary>
''' <summary>
''' Validate whether Dashboard Service is running in the Url
''' </summary>
''' <param name="dashboardServiceUrl">Dashboard Service Url</param>
''' <returns>returns whether valid dashboard service</returns>
Private Shared Function ValidateDashboardService(dashboardServiceUrl As
String) As Boolean
Dim errorOccurred As Boolean = False
Try
If String.IsNullOrWhiteSpace(dashboardServiceUrl) Then
Return True
End If
If Not dashboardServiceUrl.Contains("http://") AndAlso Not
dashboardServiceUrl.Contains("https://") Then
dashboardServiceUrl = (Convert.ToString("http://") & dashboardServiceUrl) +
"/IsServiceExists"
Else
dashboardServiceUrl = dashboardServiceUrl &
Convert.ToString("/IsServiceExists")
End If
Dim request As WebRequest = WebRequest.Create(New Uri(dashboardServiceUrl,
UriKind.Absolute))
request.Method = "GET"
Using response As WebResponse = request.GetResponse()
Using reader As New StreamReader(response.GetResponseStream())
Dim text As String = reader.ReadToEnd()
If Not
text.Contains(System.Convert.ToBase64String(System.Text.Encoding.UTF8.GetByt
es("DashboardServiceExists"))) Then
errorOccurred = True
End If
End Using
End Using
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
Catch e As Exception
dashboardServiceUrl = dashboardServiceUrl.Replace("/IsServiceExists", "")
errorOccurred = True
```



```
End Try
Return errorOccurred
End Function
Public Property DashboardServiceProcess() As Process
Get
Return m_DashboardServiceProcess
End Get
Set
m_DashboardServiceProcess = Value
End Set
End Property
Private m_DashboardServiceProcess As Process
''' <summary>
'''     Get Url string for the generated HTML page
''' </summary>
''' <param name="sydxFileName">SYDX file name</param>
''' <returns></returns>
Public Function GetUrlOfHtmlPage(sydxFileName As String) As String
If String.IsNullOrWhiteSpace(sydxFileName) OrElse
String.IsNullOrWhiteSpace(ServiceUrl) Then
Return String.Empty
End If
Dim dashboardProperties As New DashboardProperties() With {
Key .SydxFileName = sydxFileName,
Key .ServiceUrl = ServiceUrl
}
Return GetUrlOfHtmlPage(dashboardProperties)
End Function
''' <summary>
'''     Generate Viewer HTML String
''' </summary>
#Region "Generate Viewer HTML String"
Private Const BeforeHeadHtml As String = "<!-- saved from
url=(0014)about:internet -->" & vbLf + "<!DOCTYPE html>" & vbLf + "<html
xmlns=""http://www.w3.org/1999/xhtml"" style=""height :100% ;width:100%;"">"
& vbLf + "<head>" & vbLf + "<meta charset=""utf-8"" content=""width=device-
width, initial-scale=1.0"">" + "<script>window.destroyAll =
function(){try{ej.widget.destroyAll($('.e-js').off());}catch(e){}
$(document.body).html('-'); CollectGarbage(); }; </script>"
''' <summary>
'''     Get Url string for the generated HTML page
''' </summary>
Private Function GetUrlOfHtmlPage(dashboardProperties As
DashboardProperties) As String
Dim jsFiles = New List(Of String)() From {
"jquery-1.10.2.min.js",
"jquery.easing.1.3.min.js",
"ej.dashboardViewer.all.min.js"
}
Dim cssFiles = New List(Of String)() From {
"default-theme/ej.dashboardViewer.all.min.css",
"chromium.css"
}
Try
Dim htmlString As String = GetHtmlString(GetViewer(dashboardProperties),
jsFiles, cssFiles).ToString()
```



```
'Below code block is used to copy the HTML related files from app folder
since app folder doesn't have required permission to write file.
Dim sourceFolderPath As String = AppDomain.CurrentDomain.BaseDirectory +
HtmlFolder
If Not File.Exists(sourceFolderPath) Then
Directory.CreateDirectory(sourceFolderPath)
End If
Dim filePath As String = sourceFolderPath & Convert.ToString("Temp.html")
'End
Using fileStream As New FileStream(filePath, FileMode.Create)
Using w As New StreamWriter(fileStream, Encoding.UTF8)
w.Write(htmlString)
End Using
End Using
Return filePath
Catch ex As Exception
MessageBox.Show(ex.Message, "Error", MessageBoxButton.OK,
MessageBoxImage.[Error])
End Try
Return String.Empty
End Function
''' <summary>
''' Get HTML JavaScript files CSS files.
''' </summary>
Private Function GetHtmlString(bodyString As String, jsFiles As
IEnumerable(Of String), cssFiles As IEnumerable(Of String)) As StringBuilder
Dim headSb = New StringBuilder()
For Each item As var In jsFiles
headSb.Append("<script src=""scripts/" + item + """
type=""text/javascript""></script>")
Next
For Each item As var In cssFiles
headSb.Append("<link href=""themes/" + item + """
rel=""stylesheet""></link>")
Next
Dim htmlSb = New StringBuilder()
htmlSb.Append(BeforeHeadHtml)
htmlSb.Append(headSb)
htmlSb.Append("</head>" & vbLf)
htmlSb.Append(bodyString)
htmlSb.Append("</html>")
Return htmlSb
End Function
''' <summary>
'''     Generate Viewer HTML String
''' </summary>
Private Function GetViewer(dashboardProperties As DashboardProperties) As
String
If dashboardProperties Is Nothing Then
Return String.Empty
End If
Dim sydxPath = _environmentFolder & dashboardProperties.SydxFileName
Dim viewerStr =
(Convert.ToString("$('#dashboard').ejDashboardViewer({serviceUrl: '") &
dashboardProperties.ServiceUrl) + "', dashboardPath: '" +
sydxPath.Replace("\", "\\") +
"',filterParameters:location.search.substr(1)});"
```

```
Dim stringBuilder = New StringBuilder()
stringBuilder.Append("<body style=""width:100%; height:100%;
overflow:hidden;"">")
stringBuilder.Append("<div id=""dashboard"" style=""width:100%;
height:100%;"" />")
stringBuilder.Append("<script type=""text/javascript""
language=""javascript"">")
stringBuilder.Append(viewerStr)
stringBuilder.Append("</script>")
stringBuilder.Append("</body>")
Return stringBuilder.ToString()
End Function
#End Region
End Class
''' <summary>
'''     Class for Dashboard Properties
''' </summary>
Public Class DashboardProperties
''' <summary>
'''     Gets or sets the SYDX File Name
''' </summary>
Public Property SydxFileName() As String
Get
Return m_SydxFileName
End Get
Set
m_SydxFileName = Value
End Set
End Property
Private m_SydxFileName As String
''' <summary>
'''     Gets or sets the ServiceUrl
''' </summary>
Public Property ServiceUrl() As String
Get
Return m_ServiceUrl
End Get
Set
m_ServiceUrl = Value
End Set
End Property
Private m_ServiceUrl As String
End Class
```

Create a class named DashboardServiceSerialization and add the below code within the class.

C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Xml.Serialization;
public class DashboardServiceSerialization
{
```

```
static readonly XmlSerializer previewSerializer = new
XmlSerializer(typeof(DashboardServicePreviewSettings));
public void Serialize(DashboardServicePreviewSettings settings, string path)
{
try
{
using (StreamWriter writer = new StreamWriter(path))
{
previewSerializer.Serialize(writer, settings);
}
}
catch (Exception)
{
}
}
public DashboardServicePreviewSettings Deserialize(string path)
{
DashboardServicePreviewSettings settings = new
DashboardServicePreviewSettings();
try
{
using (StreamReader reader = new StreamReader(path))
{
settings =
(DashboardServicePreviewSettings)previewSerializer.Deserialize(reader);
}
}
catch (Exception)
{
}
return settings;
}
}
public class DashboardServicePreviewSettings
{
public string ServiceURL { get; set; }
public List<Guid> DashboardServiceInstances { get; set; }
public DashboardServicePreviewSettings()
{
DashboardServiceInstances = new List<Guid>();
}
}
```

VB.NET

```
Imports System.Collections.Generic
Imports System.IO
Imports System.Linq
Imports System.Web
Imports System.Xml.Serialization
Public Class DashboardServiceSerialization
Shared ReadOnly previewSerializer As New
XmlSerializer(GetType(DashboardServicePreviewSettings))
Public Sub Serialize(settings As DashboardServicePreviewSettings, path As
String)
Try
```



```
Using writer As New StreamWriter(path)
previewSerializer.Serialize(writer, settings)
End Using
Catch generatedExceptionName As Exception
End Try
End Sub
Public Function Deserialize(path As String) As
DashboardServicePreviewSettings
Dim settings As New DashboardServicePreviewSettings()
Try
Using reader As New StreamReader(path)
settings = DirectCast(previewSerializer.Deserialize(reader),
DashboardServicePreviewSettings)
End Using
Catch generatedExceptionName As Exception
End Try
Return settings
End Function
End Class
Public Class DashboardServicePreviewSettings
Public Property ServiceURL() As String
Get
Return m_ServiceURL
End Get
Set
m_ServiceURL = Value
End Set
End Property
Private m_ServiceURL As String
Public Property DashboardServiceInstances() As List(Of Guid)
Get
Return m_DashboardServiceInstances
End Get
Set
m_DashboardServiceInstances = Value
End Set
End Property
Private m_DashboardServiceInstances As List(Of Guid)
Public Sub New()
DashboardServiceInstances = New List(Of Guid)()
End Sub
End Class
```

Add a class named IISExpress and add the below code withing the class.

C#

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Globalization;
using System.Linq;
using System.Net;
using System.Net.NetworkInformation;
using System.Text;
using System.Threading.Tasks;
```

```
using System.Windows;
public static class IISExpress
{
internal static string StartIISExpress(Process iisExpressProcess, string
location)
{
string availablePort = GetAvailablePort();
string iisInstalledPath = GetIISExpressInstalledPath();
try
{
if (!string.IsNullOrEmpty(iisInstalledPath))
{
iisExpressProcess.StartInfo = new ProcessStartInfo { FileName =
iisInstalledPath + "iisexpress.exe", WindowStyle =
ProcessWindowStyle.Hidden, Arguments = @"/path:""" + location + "\" /port:"
+ availablePort, UseShellExecute = false, CreateNoWindow = true };
iisExpressProcess.Start();
return availablePort;
}
}
catch (Exception e)
{
if (e.HResult == -2147467259)
{
MessageBox.Show("IIS Express file is not found", "Error",
MessageBoxButton.OK, MessageBoxImage.Error);
}
else
MessageBox.Show(e.Message, "Error", MessageBoxButton.OK,
MessageBoxImage.Error);
}
return string.Empty;
}
private static string GetAvailablePort()
{
int startIndex = 3002, endIndex = 9000, unusedPort = 0;
IPGlobalProperties properties = IPGlobalProperties.GetIPGlobalProperties();
IPEndPoint[] tcpEndpoints = properties.GetActiveTcpListeners();
List<int> usedPorts = tcpEndpoints.Select(p => p.Port).ToList();
for (int port = startIndex; port < endIndex; port++)
{
if (!usedPorts.Contains(port))
{
unusedPort = port;
break;
}
}
return unusedPort.ToString(CultureInfo.InvariantCulture);
}
private static string GetIISExpressInstalledPath()
{
return Environment.GetEnvironmentVariable("PROGRAMFILES") + @"\IIS
Express\";
}
internal static bool IsRunning(Process process)
{
try { Process.GetProcessById(process.Id); }
```

```
catch (InvalidOperationException) { return false; }
catch (ArgumentException) { return false; }
return true;
}
}
```

VB.NET

```
Imports System.Collections.Generic
Imports System.Diagnostics
Imports System.Globalization
Imports System.Linq
Imports System.Net
Imports System.Net.NetworkInformation
Imports System.Text
Imports System.Threading.Tasks
Imports System.Windows
Public NotInheritable Class IISExpress
Private Sub New()
End Sub
Friend Shared Function StartIISExpress(iisExpressProcess As Process,
location As String) As String
Dim availablePort As String = GetAvailablePort()
Dim iisInstalledPath As String = GetIISExpressInstalledPath()
Try
If Not String.IsNullOrEmpty(iisInstalledPath) Then
iisExpressProcess.StartInfo = New ProcessStartInfo() With {
Key .FileName = iisInstalledPath & Convert.ToString("iisexpress.exe"),
Key .WindowStyle = ProcessWindowStyle.Hidden,
Key .Arguments = Convert.ToString((Convert.ToString("/path:""") & location)
+ """ /port:") & availablePort,
Key .UseShellExecute = False,
Key .CreateNoWindow = True
}
iisExpressProcess.Start()
Return availablePort
End If
Catch e As Exception
If e.HResult = -2147467259 Then
MessageBox.Show("IIS Express file is not found", "Error",
MessageBoxButton.OK, MessageBoxImage.[Error])
Else
MessageBox.Show(e.Message, "Error", MessageBoxButton.OK,
MessageBoxImage.[Error])
End If
End Try
Return String.Empty
End Function
Private Shared Function GetAvailablePort() As String
Dim startIndex As Integer = 3002, endIndex As Integer = 9000, unusedPort As
Integer = 0
Dim properties As IPGlobalProperties =
IPGlobalProperties.GetIPGlobalProperties()
Dim tcpEndpoints As IPEndPoint() = properties.GetActiveTcpListeners()
Dim usedPorts As List(Of Integer) = tcpEndpoints.[Select](Function(p)
p.Port).ToList()
```



```
For port As Integer = startIndex To endIndex - 1
If Not usedPorts.Contains(port) Then
unusedPort = port
Exit For
End If
Next
Return unusedPort.ToString(CultureInfo.InvariantCulture)
End Function
Private Shared Function GetIISExpressInstalledPath() As String
Return Environment.GetEnvironmentVariable("PROGRAMFILES") + "\IIS Express\"
End Function
Friend Shared Function IsRunning(process__1 As Process) As Boolean
Try
Process.GetProcessById(process__1.Id)
Catch generatedExceptionName As InvalidOperationException
Return False
Catch generatedExceptionName As ArgumentException
Return False
End Try
Return True
End Function
End Class
```

Finally set the serviceDirectory property value in DashboardWindow.xaml file. service which is shipped along with the SDK build. you can find the service directory in below location in your machine.

%localappdata%\Syncfusion\Dashboard Platform SDK\Service

for example,replace the serviceDirectory property value as below in your project's DashboardWindow.xaml file,

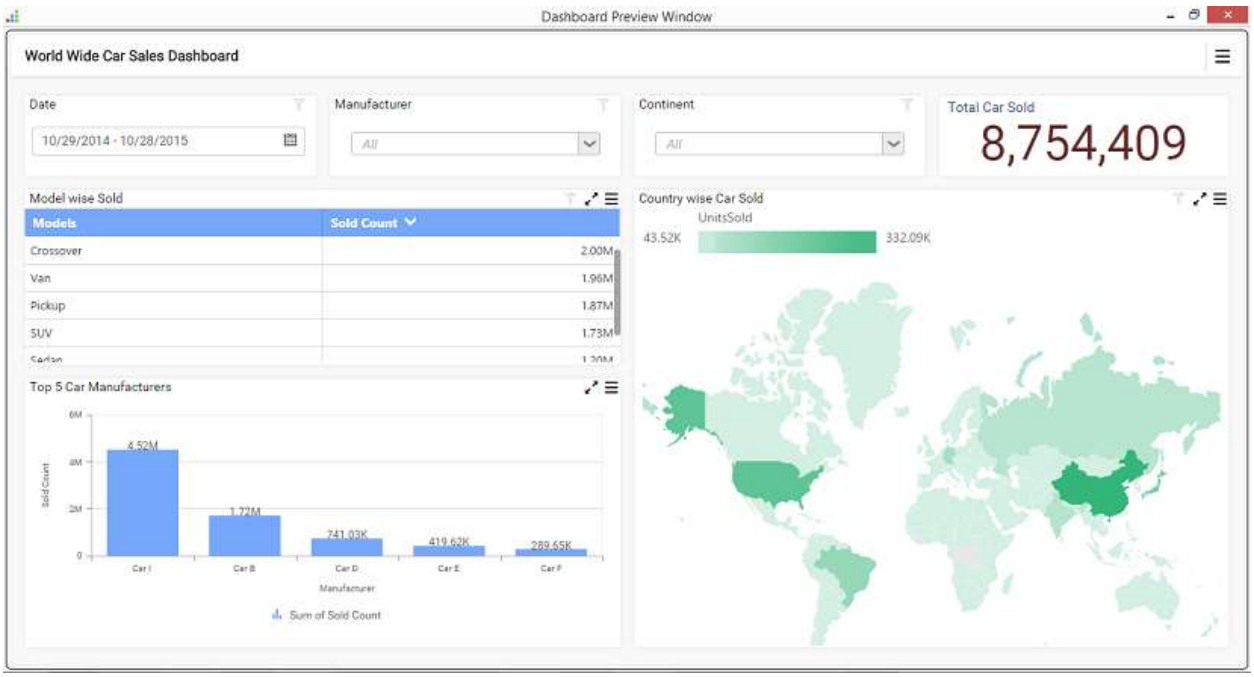
C#

```
private string serviceDirectory = "%localappdata%\Syncfusion\Dashboard
Platform SDK\Service";
```

VB.NET

```
Private serviceDirectory As String = "%localappdata%\Syncfusion\Dashboard
Platform SDK\Service"
```

Build the application and run it to view the dashboard.



Xamarin Application

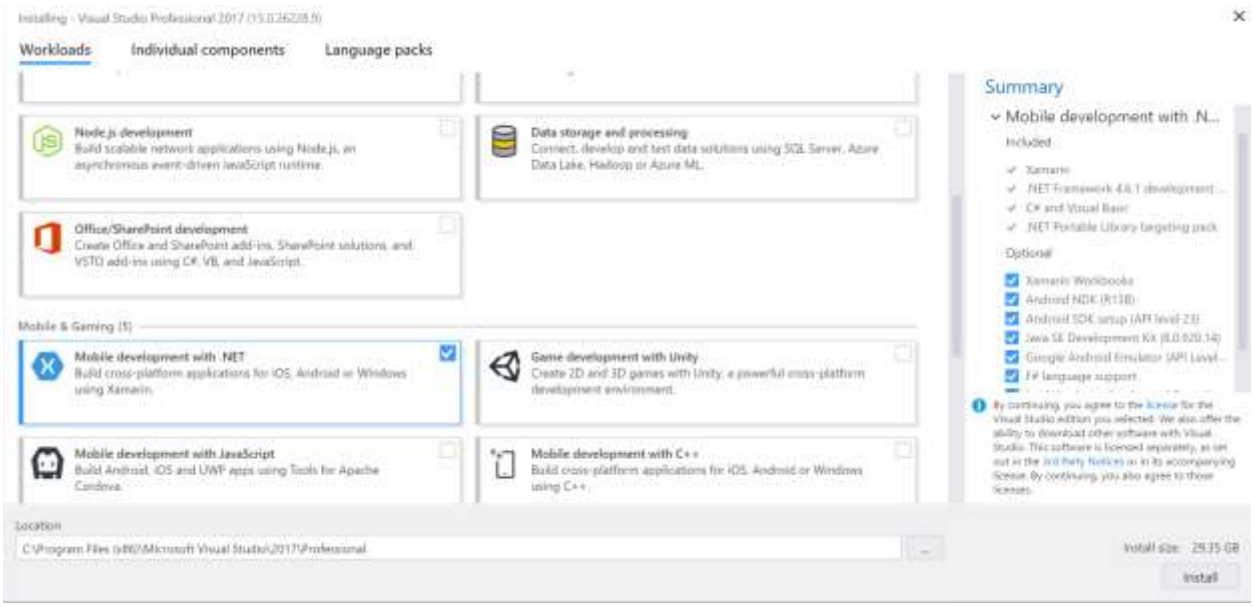
Overview

The Syncfusion Dashboard Platform SDK includes a Dashboard Viewer control that can be embedded within your Xamarin application.

System Requirements:

To work with Xamarin, the following IDE can be used for development that are compatible with Microsoft Windows, both 32bit and 64bit Operating System in Windows 10

- Microsoft Visual Studio 2017 or higher with .NET Mobile development.



Supported Platforms

- Android 4.1 or later.
- iOS 9.0 or later.

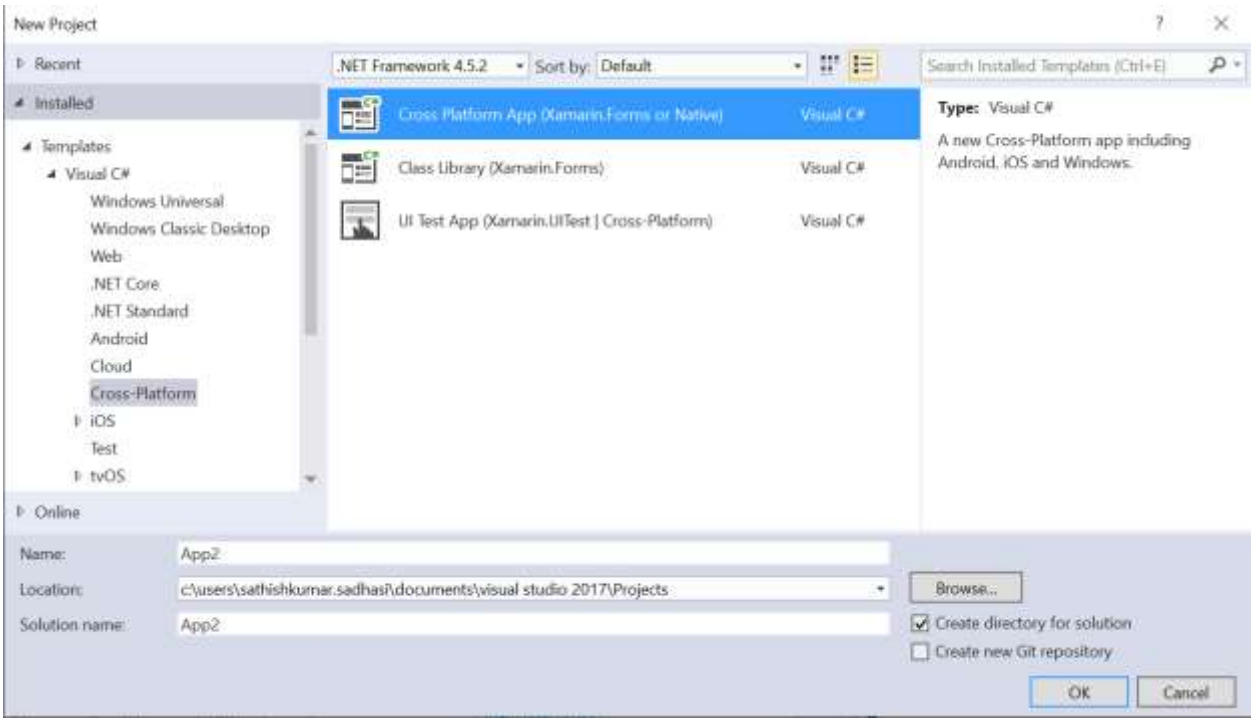
Getting Started

Configure Syncfusion Dashboard Viewer Component in Xamarin Application using Visual Studio:

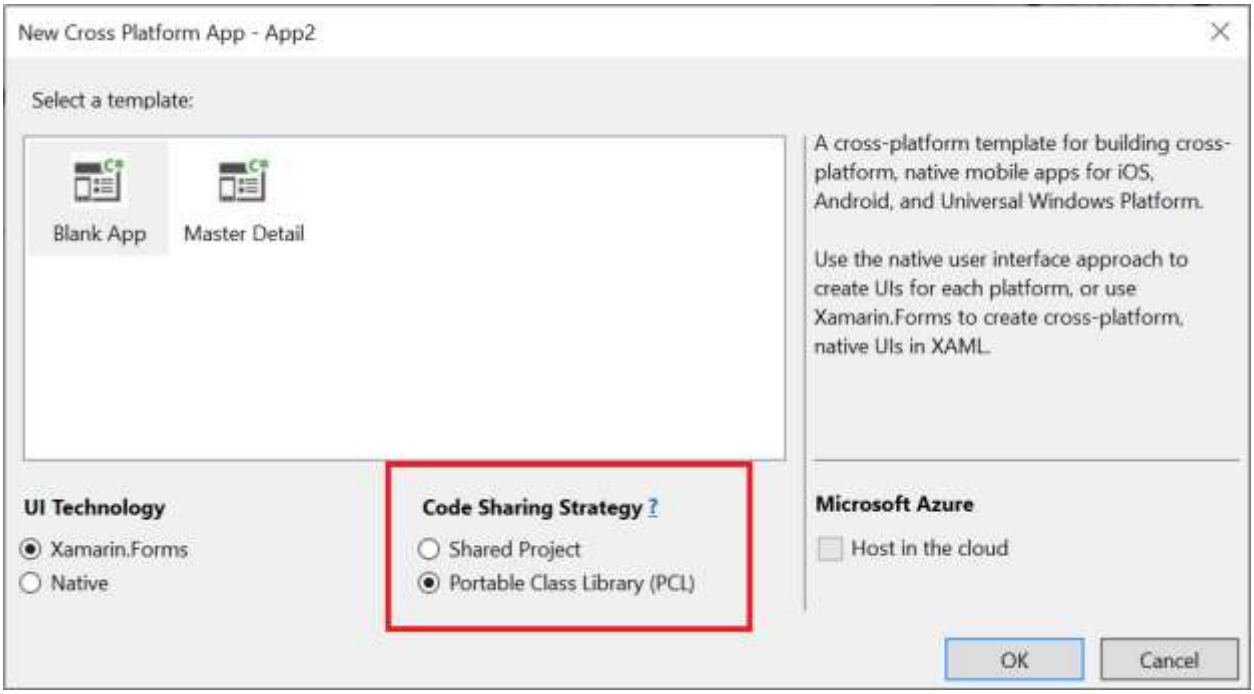
The following steps helps to create a Xamarin application by embed the Dashboard Viewer component.

Visual Studio 2017

- Open Visual Studio 2017, Create a new project and select Cross-Platform Template under Visual C# category, here you need to choose the Cross Platform App (Xamarin Forms or Native), which includes the Android, iOS and Windows
- Type your application name and Click OK

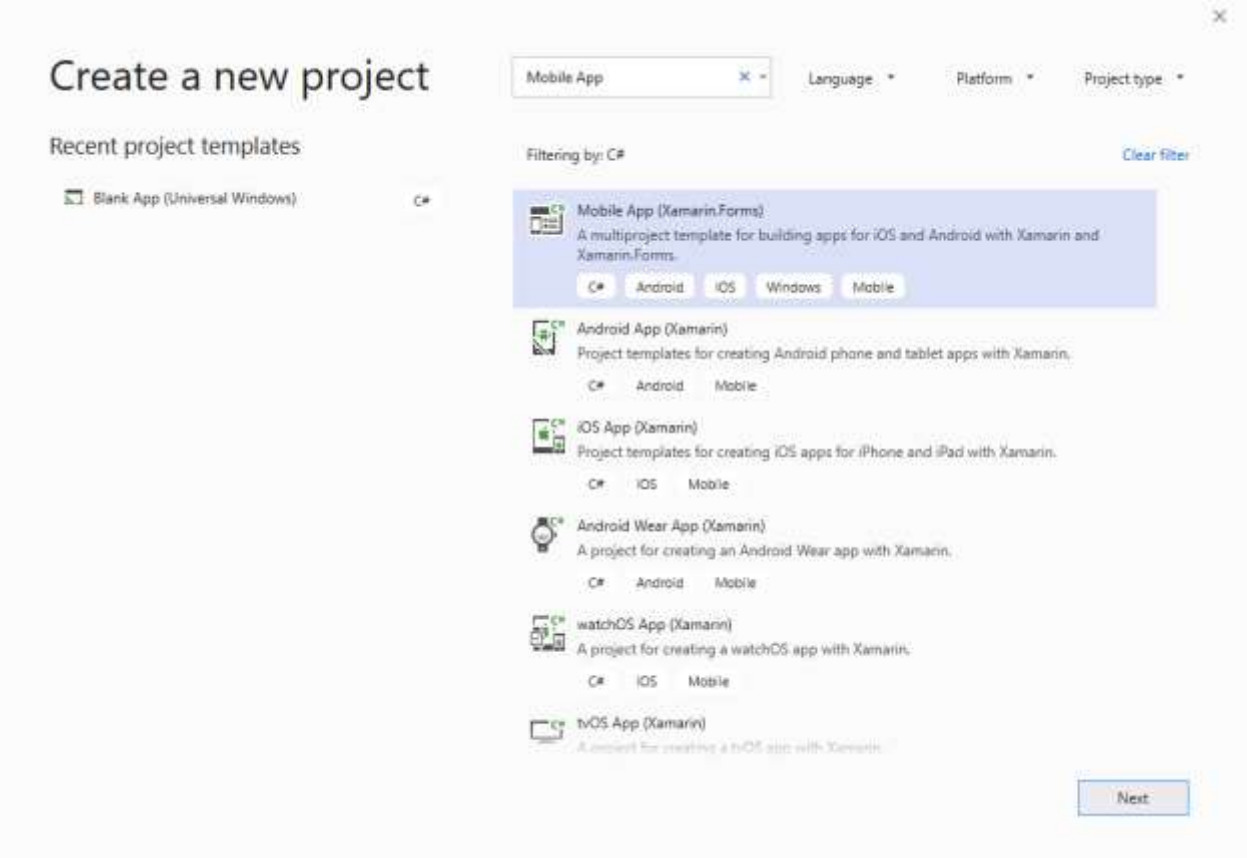


- For a New Cross Platform App, you need to choose the Blank App under the Portable Class Library (PCL) of Code Sharing Strategy

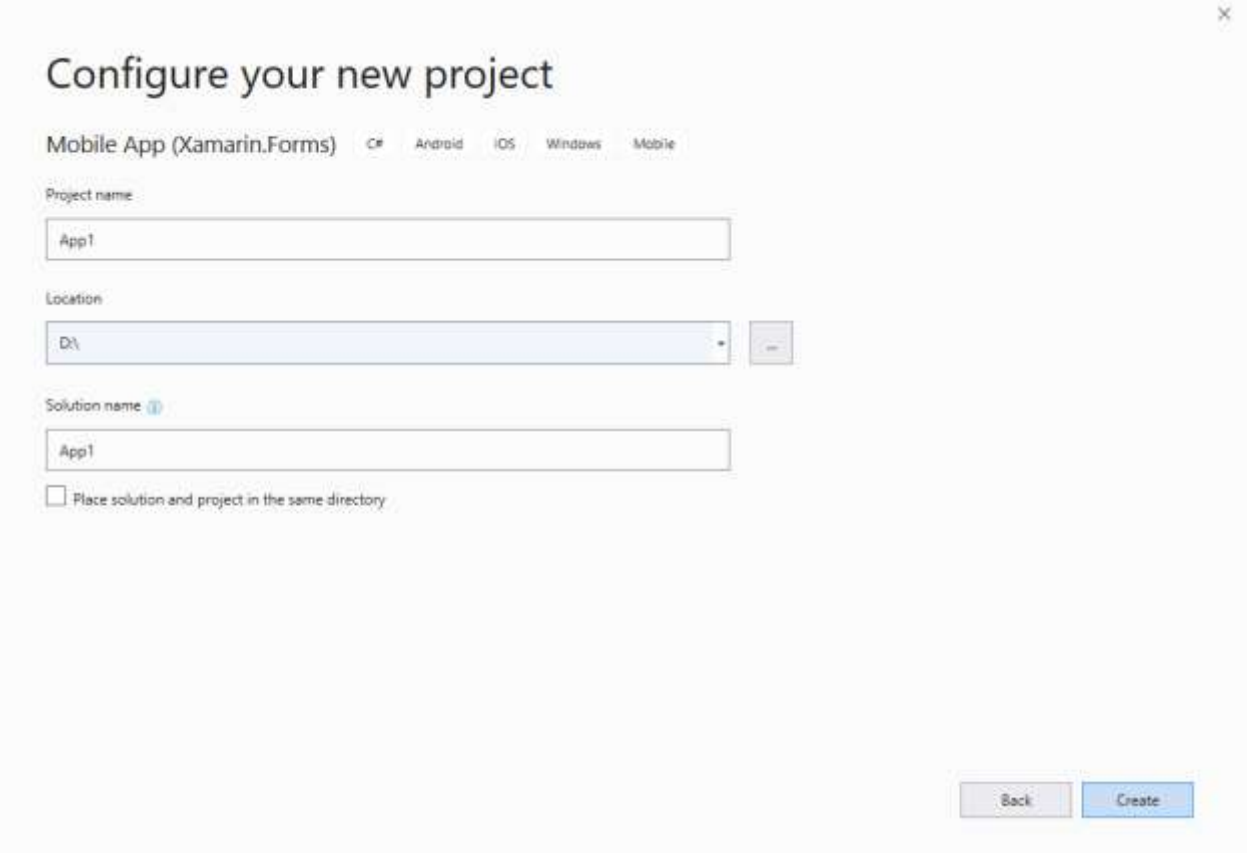


Visual Studio 2019

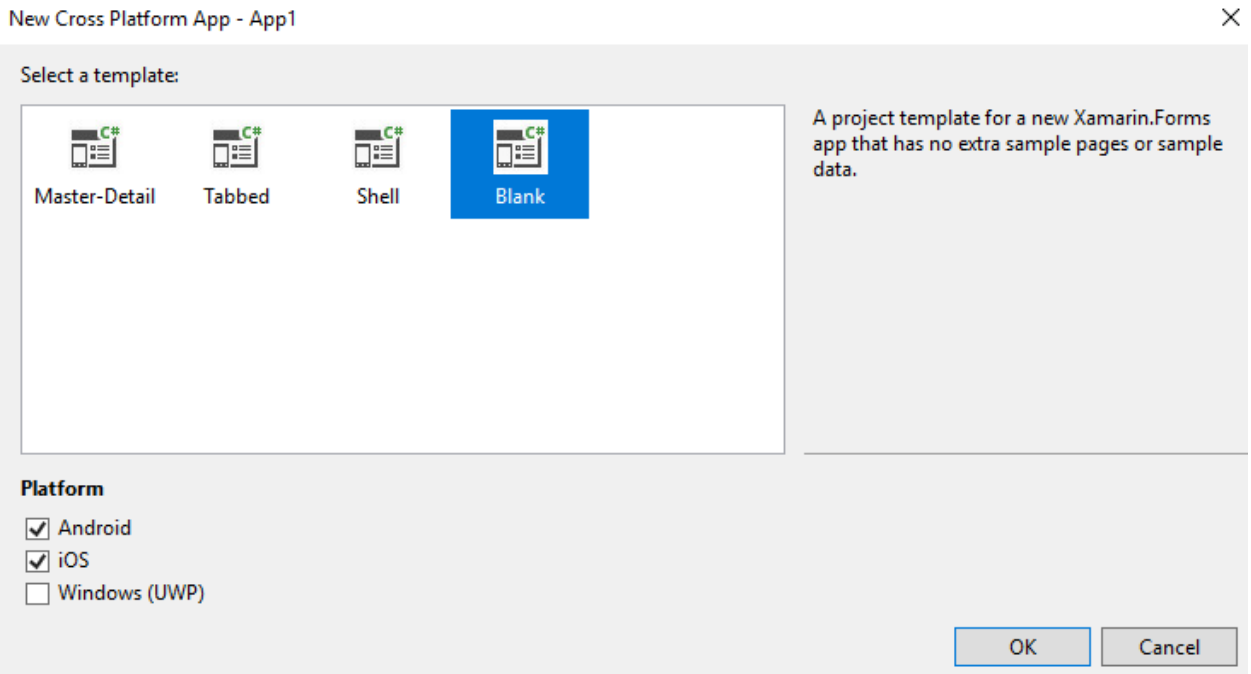
- Open Visual Studio 2019, Create a new project and select C# under the Language category, here you need to choose the Mobile App (Xamarin.Forms), which includes the Android, iOS and Windows
- Select the template then Click Next



- Type your application name and Click Create



- For a New Mobile App(Xamarin.Forms), you need to choose the Blank App



- Create a new folder say, dashboard under %projectname%.Android\Assets and %projectname%.iOS\Resource
- Copy the required scripts, themes and fonts into %projectname%.Android\Assets\dashboard and %projectname%.iOS\Resource\dashboard folder in your new Xamarin application for rendering the ejDashboardViewer.
- These can be availed from the Dashboard Platform SDK build installed location mentioned below:

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html

- Go to MainPage.xaml and add a WebView and provide suitable name.

ASPX-CS

```
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
xmlns:local="clr-namespace:App2"
x:Class="App2.MainPage">
<WebView x:Name="webView" HorizontalOptions="FillAndExpand"
VerticalOptions="FillAndExpand"/>
</ContentPage>
```

- Create an Interface IResourceSettings in the portable project, and add a method to get the base (or common) folder path of the resource files like scripts, fonts and themes reference for Android and iOS projects

C#

```
public interface IResourceSettings
{
string GetResourcePath();
}
```

- Define the above interface on Android and iOS projects by generating new classes for the base (or common) resource path.

For Android, Navigate to Android project and define the below class by using following code snippet.

C#

```
public class ResourceSettingsDroid : IResourceSettings
{
public string GetResourcePath()
{
return "file:///android_asset/";
}
}
```


For iOS, Navigate to iOS project and define the below class by using following code snippet.

C#

```
public class ResourceSettingsiOS : IResourceSettings
{
public string GetResourcePath()
{
return NSBundle.MainBundle.BundlePath;
}
}
```

- Then configure the MainPageXaml.cs with HtmlWebViewSource class by assign the HTML, which contains the dashboard information, to the 'webView' declared in MainPage.Xaml using following code snippet:

C#

```
public partial class MainPage : ContentPage
{
public MainPage()
{
InitializeComponent();
var htmlSource = new HtmlWebViewSource();
htmlSource.BaseUrl = DependencyService.Get< IResourceSettings
>().GetResourcePath();
StringBuilder htmlString = new StringBuilder();
htmlString.Append("<html><head><meta charset = 'utf-8' /><meta name =
'viewport' content = 'width=device-width, initial-scale=1.0'
/><title>Dashboard Viewer Embedding Demo through Xamarin
Application</title>");
htmlString.Append("<link rel = 'stylesheet' href =
'dashboard/themes/default-theme/ej.dashboardViewer.all.min.css' />");
htmlString.Append("<script src = 'dashboard/scripts/jquery-
1.10.2.min.js'></script >");
htmlString.Append("<script src =
'dashboard/scripts/jquery.easing.1.3.min.js'></script>");
htmlString.Append("<script src =
'dashboard/scripts/ej.dashboardViewer.all.min.js'></script>");
htmlString.Append("</head><body style =
'height:100%;width:100%;padding:0;'><div id = 'dashboard' style = 'width:
100%; height: 100%'></div>");
htmlString.Append("<script type = 'text/javascript'>$(function(e) {
$('#dashboard').ejDashboardViewer({serviceUrl:
'https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc',
dashboardPath:
'https://dashboardsdk.syncfusion.com//Dashboards//WorldWideCarSalesDashboard
.sydx'});});</script></body></html>");
htmlSource.Html = htmlString.ToString();
webView.Source = htmlSource; //Bind the HTML Web View source after
configuration to web view.
}
}
```


Note: *The provided serviceUrl and dashboardPath are demo links for Dashboard Service and Dashboard file respectively. Use your valid URL of Dashboard Service, and Dashboard (SYDX) file path for your dashboard configuration.*

- Finally compile your project, after successful compilation run the project via the Android emulator if we deploy the android project or else connecting a mac machine on remote to deploy the iOS project.
- Refer here for connecting a mac machine on remote.

Binding Dashboard Service

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS
2. Hosting Dashboard Service in IIS Express
3. Hosting Dashboard Service as Windows Service Background Process

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Overview

Syncfusion Dashboard Platform SDK includes sample browser which includes features of Syncfusion Dashboard Viewer by using JavaScript API.

Getting Started with SDK sample browser

This is simple walkthrough to get started with Syncfusion Dashboard Platform SDK Sample Browser.

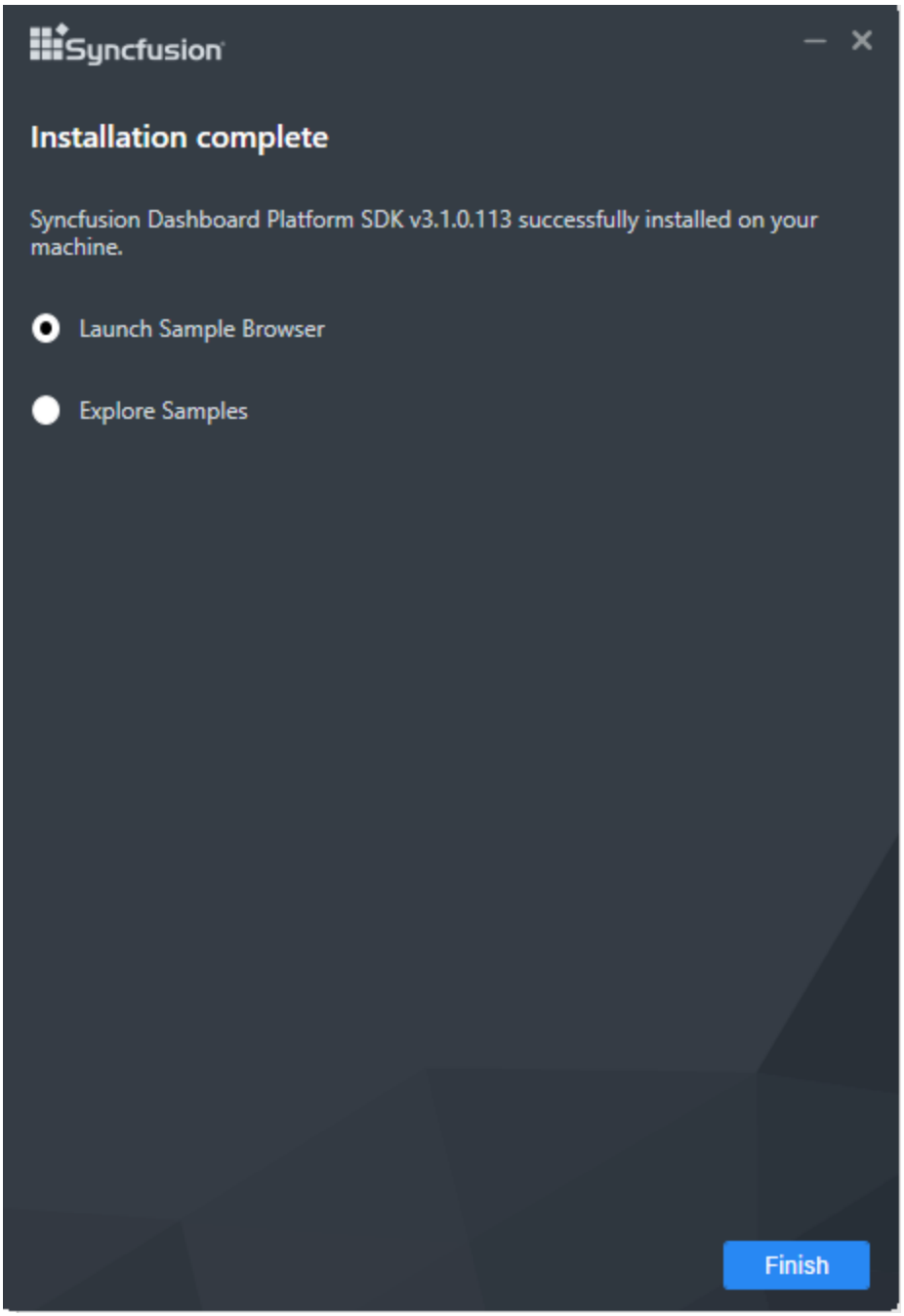
Running SDK sample browser

Click the following link to install the Syncfusion Dashboard Platform SDK

https://help.syncfusion.com/dashboard-platform/dashboard-sdk/installation-and-deployment#installation

After completing the installation, select the Launch Sample Browser radio button, and then click finish. It will be opened in the default browser of your machine.

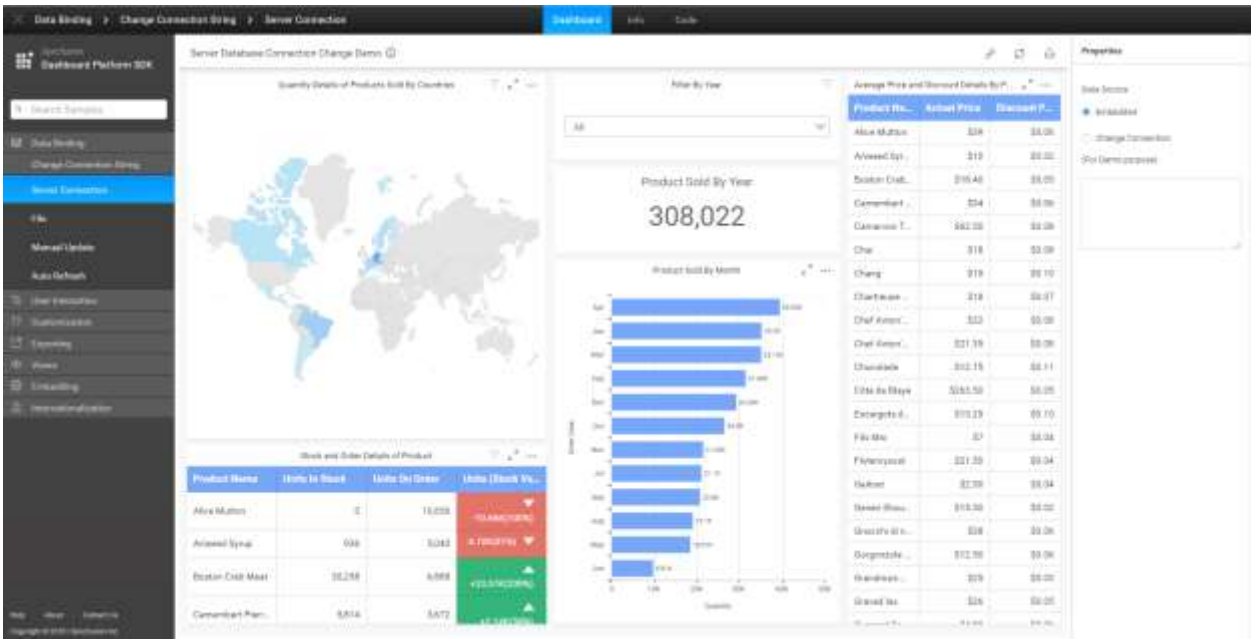
The following screenshot illustrates this process:



Or

Also, you can launch the Dashboard Platform SDK sample browser through double-clicking the shortcut icon

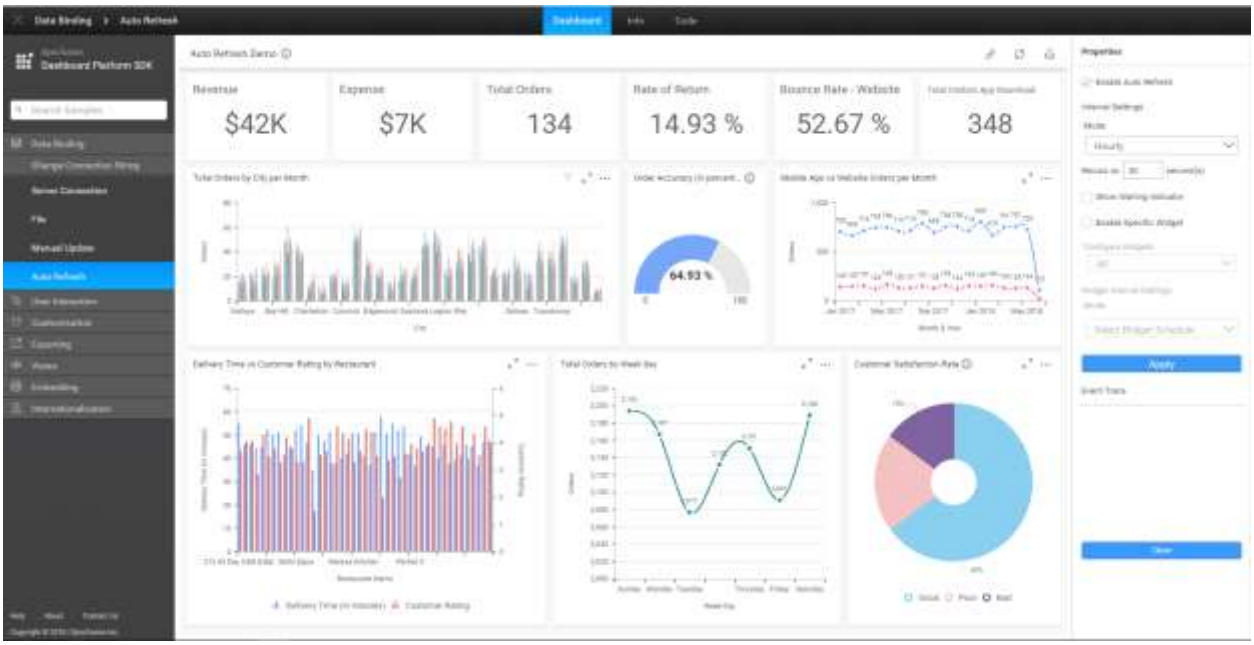
Now, the Dashboard Platform SDK sample browser will open in the default browser of your machine.



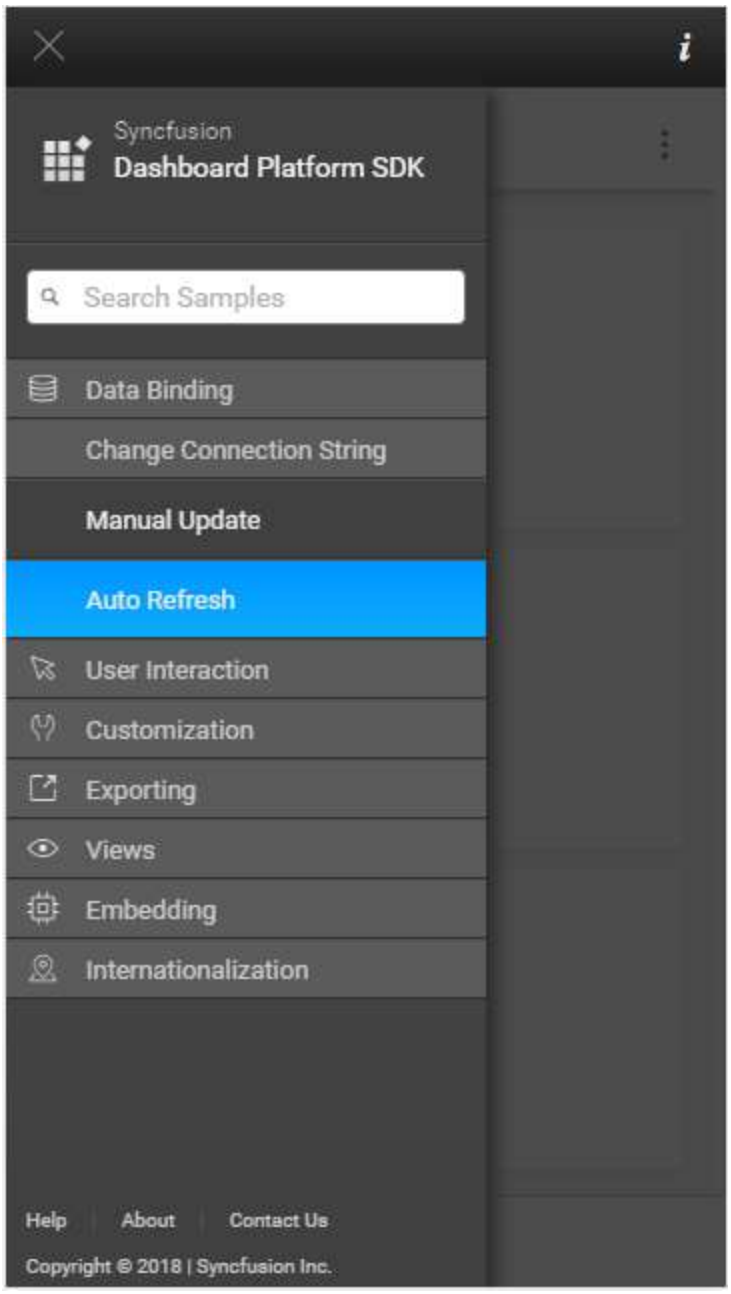
Using Dashboard Platform SDK sample browser

The left pane holds the list of samples of dashboard viewer in the Dashboard Platform SDK. Click any sample, the sample dashboard will be rendered at the center pane and the properties of the corresponding sample will be shown at the right pane.

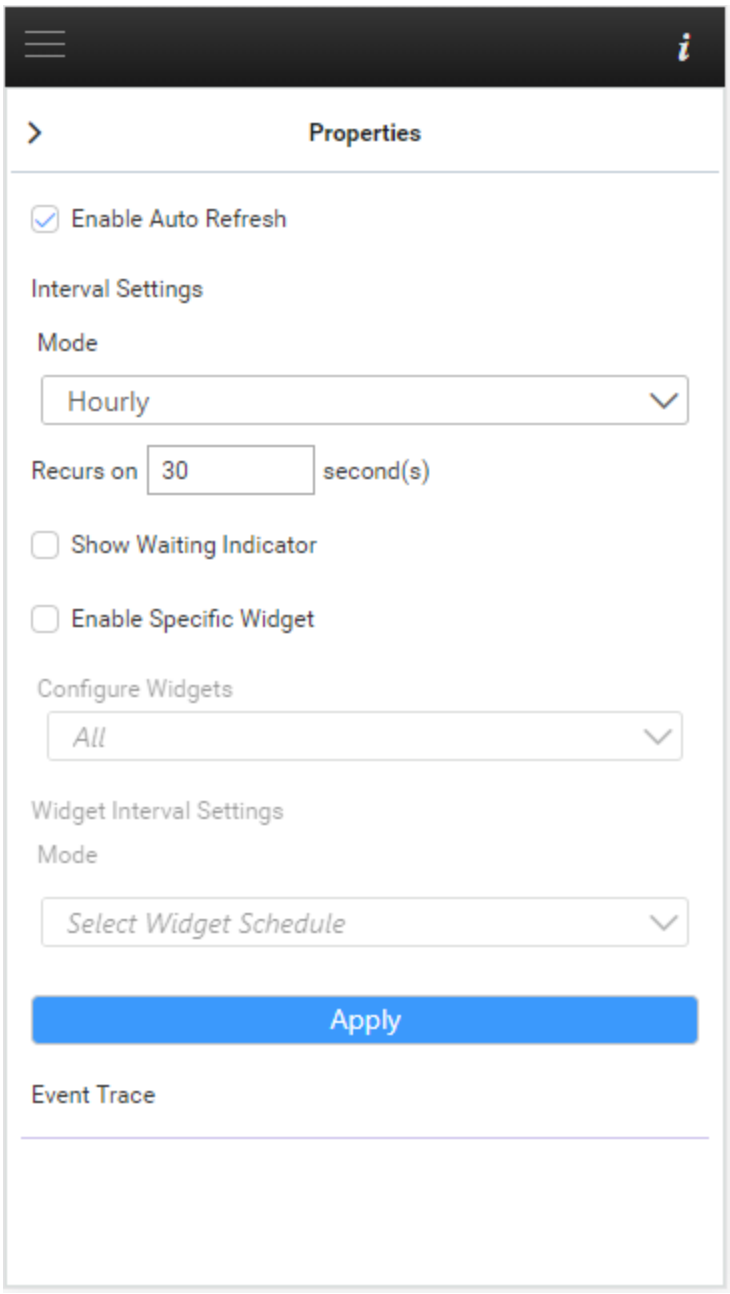
The following screenshot illustrates the dashboard Auto Refresh sample in the desktop view:



In mobile view, the showcase list of samples of the dashboard viewer in the Dashboard Platform SDK is displayed while clicking the header menu like below:



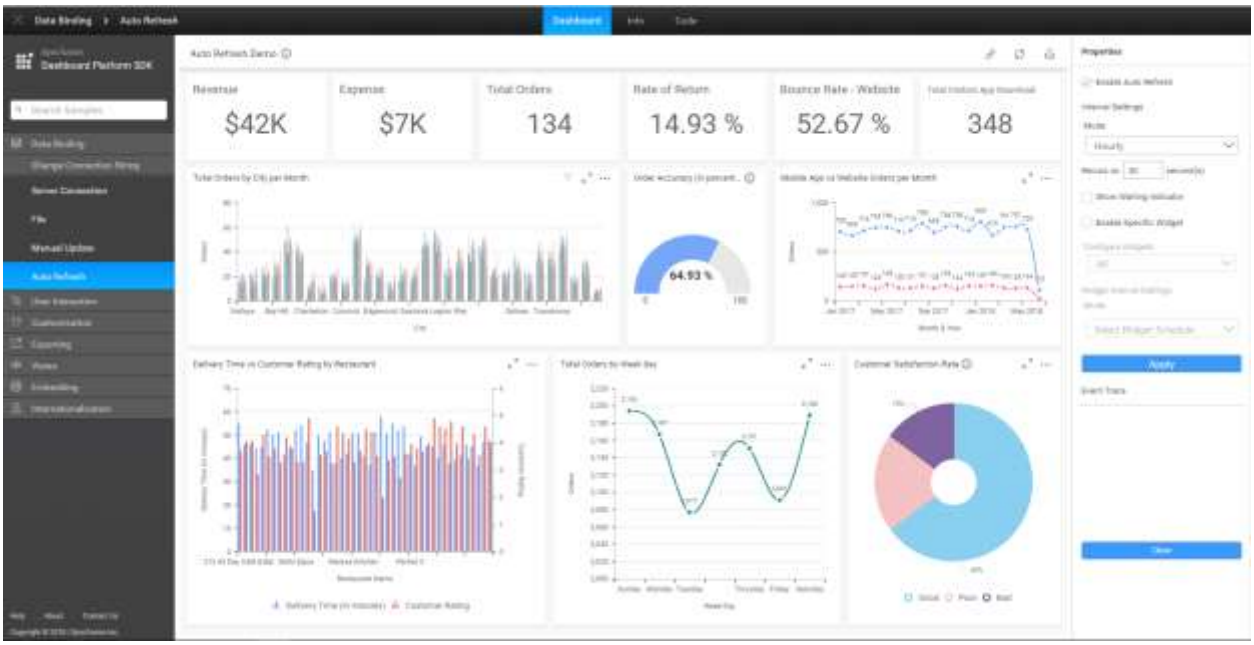
The properties window will be opened while clicking the properties option in the footer like below:



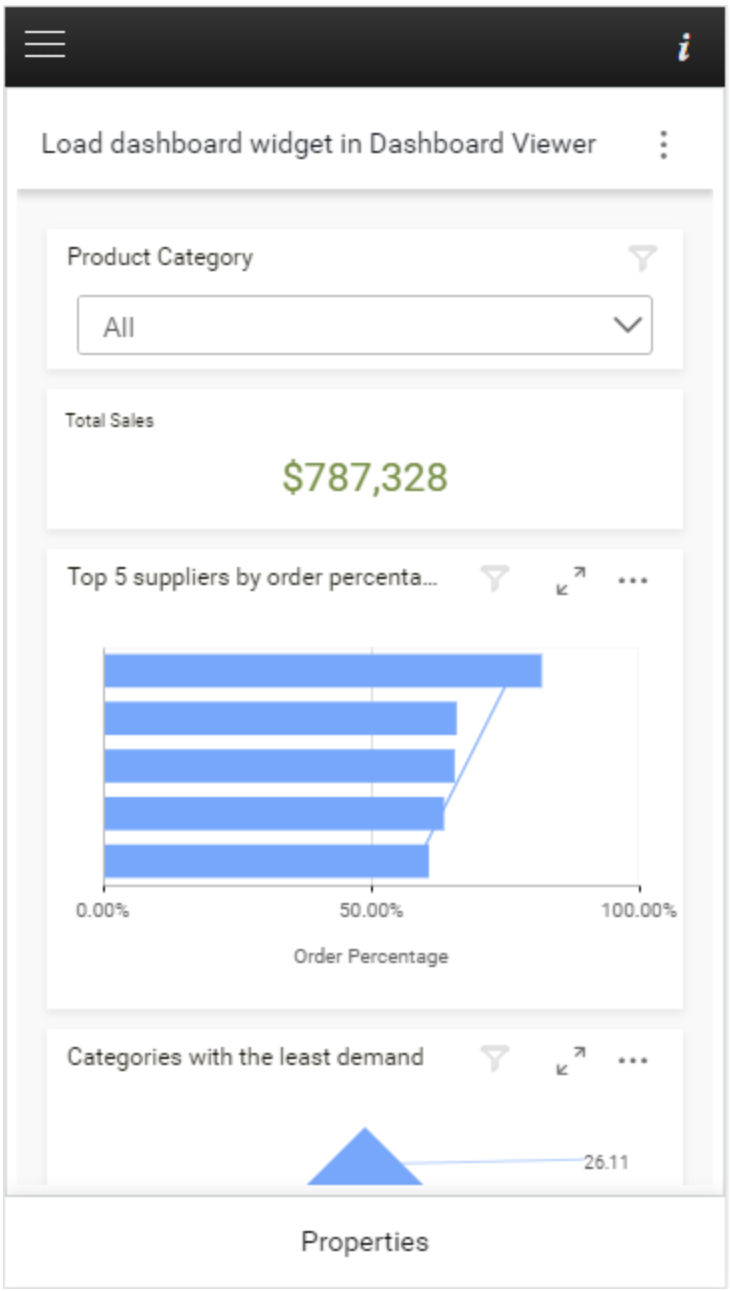
The following are the three tabs provided in the header of the Dashboard Platform SDK sample browser.

- Dashboard
- Info
- Code

The sample dashboard will be rendered at the center pane while clicking the dashboard tab in desktop view like below

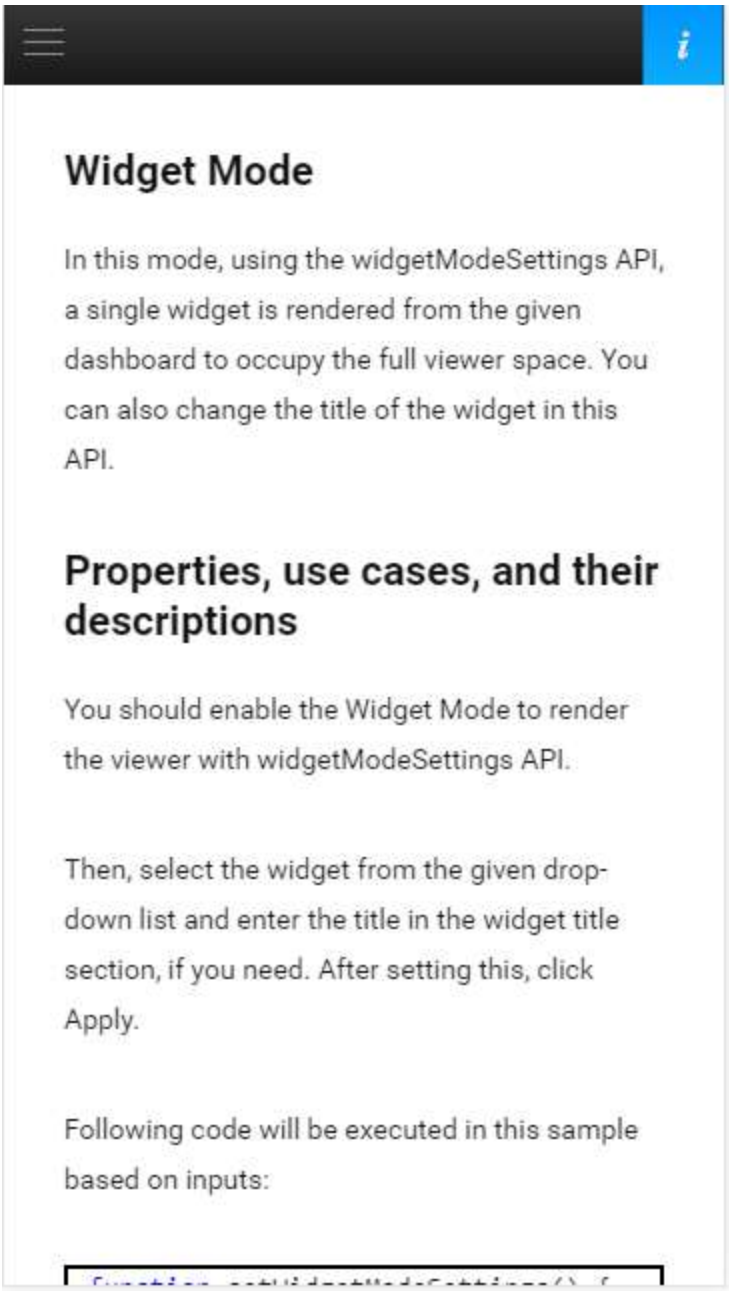


In mobile view:



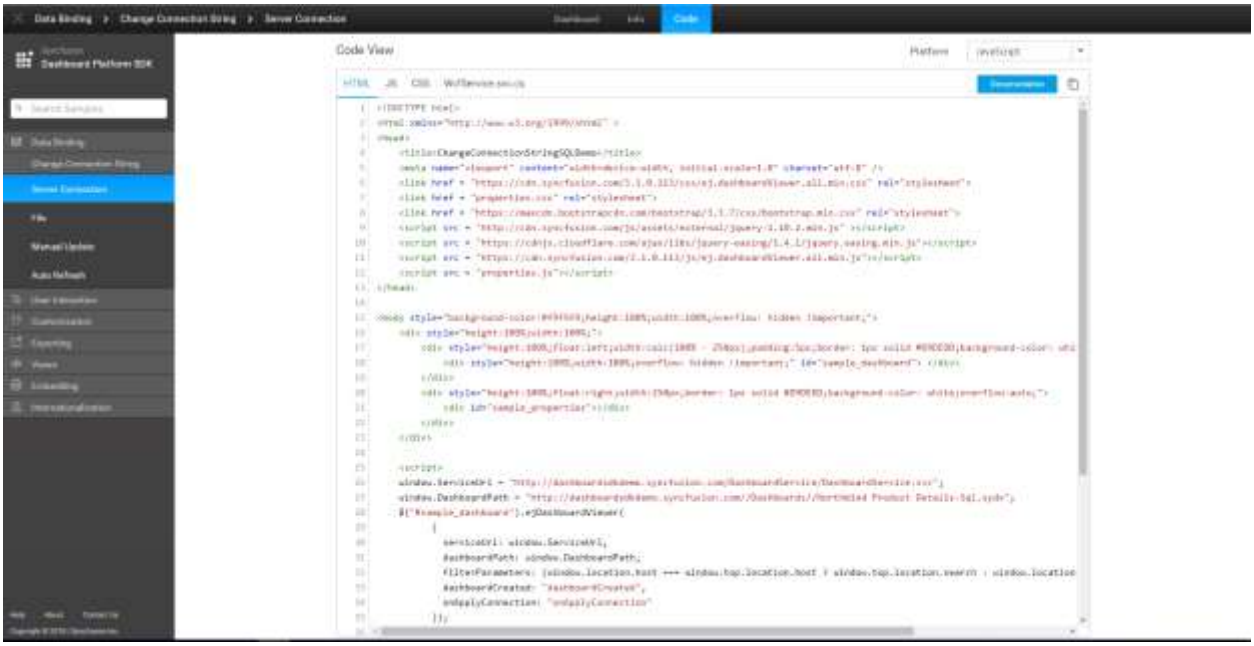
When you switch to Info tab, the selected sample information will be shown as below in the desktop view

In mobile view, information tab will be shown as below



The code view of the selected sample will be showcased based on a platform, when you switch to the Code tab.

The following screenshot illustrates the code view for desktop view



Custom Theme

Custom theme settings allow you to customize the appearance of the dashboard viewer. The user can achieve this theme customization using JSON settings which have separate properties to customize the dashboard viewer. The JSON settings can be given as input for API member “customThemeSettings”.

EnableCustomTheming

EnableCustomTheming property will enable the customization of Dashboard Viewer appearance. After enable this the user can modify the appearance of Dashboard Viewer.

Default Value

False.

Example

JS

```
$("#container").ejDashboardViewer({
enableCustomTheming: true
});
```

customThemeSettings

customThemeSettings contains JSON input which is used to customize the appearance of Dashboard Viewer.

Default Value

“”

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"dashboard": {},
"fontSettings": {},
"banner": {},
```



```
"container": {},
"menuSettings": {},
};
});
```

customThemeSettings.dashboard

Used for customizing the background color, background image, hovering color and menu hovering color of Dashboard Viewer.. User can set background image for Dashboard Viewer also. User can set either background- color or background image for Dashboard Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"dashboard": {
"background-color": "#0c0d0f",
"background-image": "",
"menu-hover-color": "#262A33",
"icon-hover-color": "#F2F2F2",
},
};
});
```

customThemeSettings.fontSettings

Used for customizing the font family, font size used in Dashboard Viewer. The user can use external fonts by specifying the source path of the along with format type. For using the external fonts need to give the source path where the external font is placed and need to give the format type.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"fontSettings": {
"font-family": 'Roboto',
"src": "",
"format": "",
"font-size": "20px"
},
};
});
```

Note: font size should not exceed 26px to view the title without any distortion.

customThemeSettings.banner

Used for customizing the background color, title color, title font size, border color and icons color of dashboard banner. Using this settings user can customize the background color, title color and icons color.

Example

JS

```
$("#container").ejDashboardViewer({
```



```
customThemeSettings: {
"banner": {
"background-color": "linear-gradient(to bottom, rgba(35,36,40,1) 0%,
rgba(24,25,29,1) 100%)",
"title-color": "#aeb2b7",
"icons-color": "#aeb2b7",
"banner-top-border-color": "rgba(35,36,40,1)",
"font-size": "20px"
},
};
});
```

Note: font size should not exceed 26px to view the title without any distortion.

How to hide the banner in the dashboard?

Hide banner in the Dashboard by customizing CSS class ".e-dbrd-banner"

Example

JS

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" style="width:100%; height:100%;">
<head>
<style>
.e-dbrd-banner {
display: none !important;
height: 0px !important;
}
</style>
<title>Dashboard Viewer</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body style="width: 100%;height: 100%">
<script type="text/javascript">
$(function (e) {
$("#container").ejDashboardViewer({
serviceUrl: 'service URL',
dashboardPath: 'Path of the dashboard'
});
});
</script>
<div id="container" style="width: 100%; height: 100%"></div>
</body>
</html>
```

customThemeSettings.Container

Used to customize the widget container. Using this settings user can customize the background color, border, box shadow, border radius, header background color, title color, title font size and icons color.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"container": {
```



```
"background-color": "#232428",
"header-background-color": "#232428",
"title-color": "#aeb2b7",
"icons-color": "#aeb2b7",
"border-radius": "2px",
"box-shadow": "none",
"font-size": "20px"
},
};
});
```

Note: font size should not exceed 26px to view the title without any distortion.

customThemeSettings.menuSettings

Used to customize the widget menu. Using this settings user can customize the background color, border color, text color and font size for menu popup of Dashboard Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"menuSettings": {
"background-color": "#333842",
"border-color": "none",
"font-color": "#f9f9f9",
"font-size": "20px"
},
};
});
```

customThemeSettings.descriptionSettings

Used for customizing the background color, border color , font color and font size of description popup of Dashboard Viewer. Using this settings user can customize the background color, border color and font color of description popup.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"descriptionSettings": {
"background-color": "#ff0000",
"border-color": "rgb(238, 72, 163)",
"font-color": "rgb(255, 255, 255)",
"font-size": "20px"
},
};
});
```

customThemeSettings.popupSettings

Used for customizing the dialogs in the Dashboard Viewer such as export dialog, error detail dialog etc. The user can customize the background color, container background color, title color, font color, icons color, input box background, border and font color, error message color, hyperlink color and font size.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"popupSettings": {
"popup-background-color": "rgba(0, 0, 0, 0.5)",
"popup-container-background-color": "#333842",
"popup-title-color": "rgb(255, 255, 255)",
"popup-font-color": "rgb(255, 255, 255)",
"popup-icons-color": "#ffffff",
"input-box-background-color": "#2a2f38",
"input-box-border-color": "#5a5f66",
"inputbox-font-color": "#f9f9f9",
"hyperlink-color": "rgb(255, 255, 255)",
"error-msg-color": "rgb(255,255,255)",
"font-size": "20px"
},
};
});
```

customThemeSettings.buttonSettings

Used for customizing the buttons in the Dashboard Viewer such as Export, Cancel, Reset, OK etc. The user can customize the background color and font color of buttons. The buttons in Dashboard Viewer are categorized as success button, cancel button, close button, reset button, disable button and font size.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"buttonSettings": {
"success-background-color": "#009aef",
"success-font-color": "#fff",
"close-background-color": "#1f252d",
"close-font-color": "#78a",
"cancel-background-color": "#f4f4f4",
"cancel-font-color": "#161616",
"reset-background-color": "#f4f4f4",
"reset-font-color": "#161616",
"disabled-btn-background-color": "rgb(0, 0, 0)",
"disabled-btn-font-color": "rgb(255, 255, 255)",
"font-size": "20px"
},
};
});
```

customThemeSettings.maximizationSettings

Used for customizing the background color, header background color, container background color, footer background color, title color, icons color, description font color and font size of widgets in maximized view.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"maximizationSettings": {
"max-background-color": "rgba(0, 0, 0, 0.5)",
"max-header-background-color": "#232428",
"max-container-background-color": "#232428",
"max-footer-background-color": "#232428",
"max-title-color": "#ffffff",
"max-icons-color": "#ffffff",
"description-font-color": "#ffffff",
"font-size": "20px"
},
};
});
```

customThemeSettings.filterOverviewSettings

Used for customizing the background color, footer background color, title color, icons color, back icon color, clear all font color and font size of filter overview popup in Dashboard Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"filterOverviewSettings": {
"background-color": "#333842",
"border-color": "#3c3d42",
"footer-background-color": "#009aef",
"font-color": "#f9f9f9",
"icons-color": "#f9f9f9",
"back-icon-color": "#ffffff",
"clear-all-font-color": "#009aef",
"font-size": "20px"
},
};
});
```

customThemeSettings.passiveErrorSettings

Used for customizing the passive error banner that displays the error messages that occur while rendering dashboard. Using this settings user can customize the background color, font color, icons color, link color and font size of passive error.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"passiveErrorSettings": {
"background-color": "#fbcb43",
"font-color": "#1c1c1c",
"icons-color": "#af8423",
"link-color": "#996f07",
```



```
"font-size": "20px"
},
};
});
```

customThemeSettings.erroredContainer

Used for customizing the errors/warnings shown in widget container that prevents the widget from rendering in the viewer. Using this settings user can customize the background color, font color, link color and font size of error container.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"erroredContainer": {
"background-color": "#232428",
"font-color": "#7a7e89",
"link-color": "#3b99fc",
"font-size": "20px"
},
};
});
```

customThemeSettings.tooltipSettings

Used for customizing the dashboard tooltip, that appears upon hovering on dashboard or widget title, maximizing icon, menu icon etc. Using this settings user can customize the background color, font color, border radius and font size of tooltip.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"tooltipSettings": {
"background-color": "#2a2f38",
"font-color": "#ffffff",
"border-radius": "2px",
"font-size": "20px"
},
};
});
```

customThemeSettings.multitabSettings

Used for customizing the background color, font color, tab hovering color, tab hovering font color, selected tab color, font-size etc. Using this settings user can customize the theme in multi tab container.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"multitabSettings": {
"background-color": "#0c0d0f",
```



```
"font-color": "rgb(174, 178, 183)",
"tab-hover-color": "#aeb2b7",
"tab-hover-font-color": "#fff",
"selected-tab-color": "#179bd7",
"selected-font-color": "#fff",
"multitab-export-button-background-color": "#0c0d0f",
"multitab-export-button-font-color": "rgb(174, 178, 183)",
"multitab-mobile-popup-border-color": "rgb(174, 178, 183)",
"arrow-icons-background-color": "#0c0d0f",
"arrow-icons-color": "#aeb2b7",
"font-size": "20px"
},
};
});
```

customThemeSettings.filterPanelSettings

Used for customizing the filter panel banner and container element. User can customize the filter panel background color, banner color, banner background color, banner title font size, container background color, container color, container font size etc.

Example

JS

```
$("#container").ejDashboardViewer({
customThemeSettings: {
"filterPanelSettings": {
"filterPanel": {
"background-color": "#0c0d0f",
"separator-color": "#F2F2F2",
},
"bannerSettings": {
"background-color": "linear-gradient(to bottom, rgba(35,36,40,1) 0%,
rgba(24,25,29,1) 100%)",
"title-color": "#aeb2b7",
"icons-color": "#F2F2F2",
"font-size": "14px"
},
"fontSettings": {
"src": "fonts/GreatVibes-Regular.otf",
"format": "opentype",
"font-size": "14px"
},
"container": {
"background-color": "#232428",
"header-background-color": "#232428",
"title-color": "#aeb2b7",
"icons-color": "#aeb2b7"
"font-size": "17px"
}
},
};
});
```


Customized appearance of Dashboard Viewer

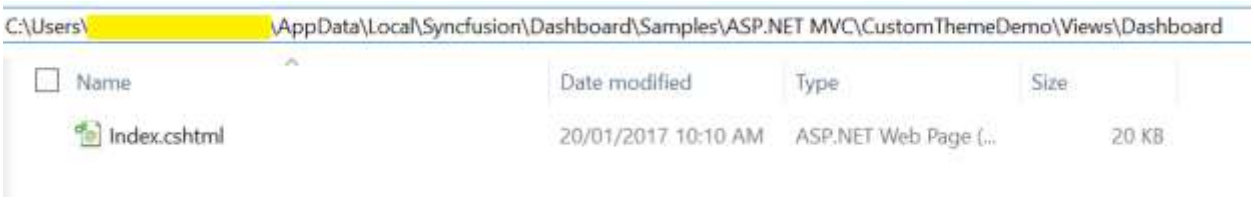
Apply Custom Theme to Widget

Through custom theme support we can apply user defined theme values to widgets in the dashboard.

How to apply custom theme to widget

The following steps illustrates how to define a theme and apply the theme to widgets

- In Custom theme demo sample, open the “index.cshtml” in the following the location.



- We can customize theme of widget under “widgetThemeSettings” section in the file.

CSS properties

We have customized the below CSS properties by using custom theme support.

- background-color
- font-color
- font-family
- border-color
- font-size

Grid

Sections

We can customize the below sections in grid widget.

- Header
- Content
- Hover
- Border
- Selection
- FontSettings

Header

We can customize the below properties in header section.

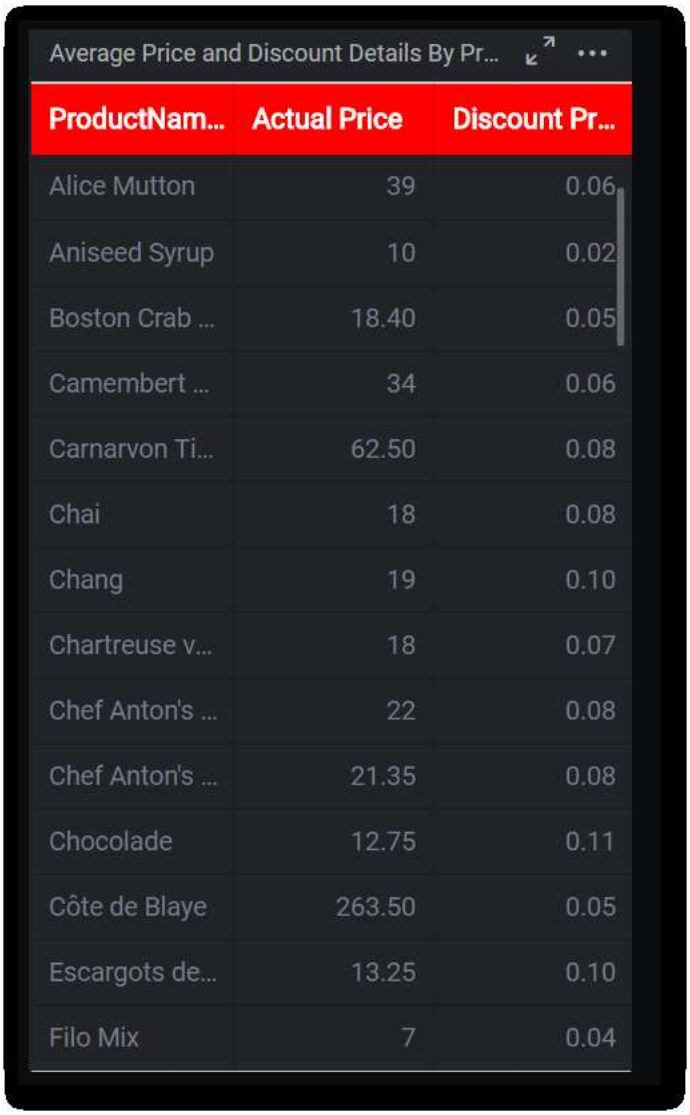
- background-color
- font-color
- font-family

For example, we can change the background color for grid header. Below code snippet represent to change the background color of grid header.

JS

```
"header":
{
"background-color": "red",
"font-color": "#7a7e89",
"font-family": "Roboto",
},
```

The following image illustrates the applied background color for grid header.



Content

We can customize the below properties in content section.

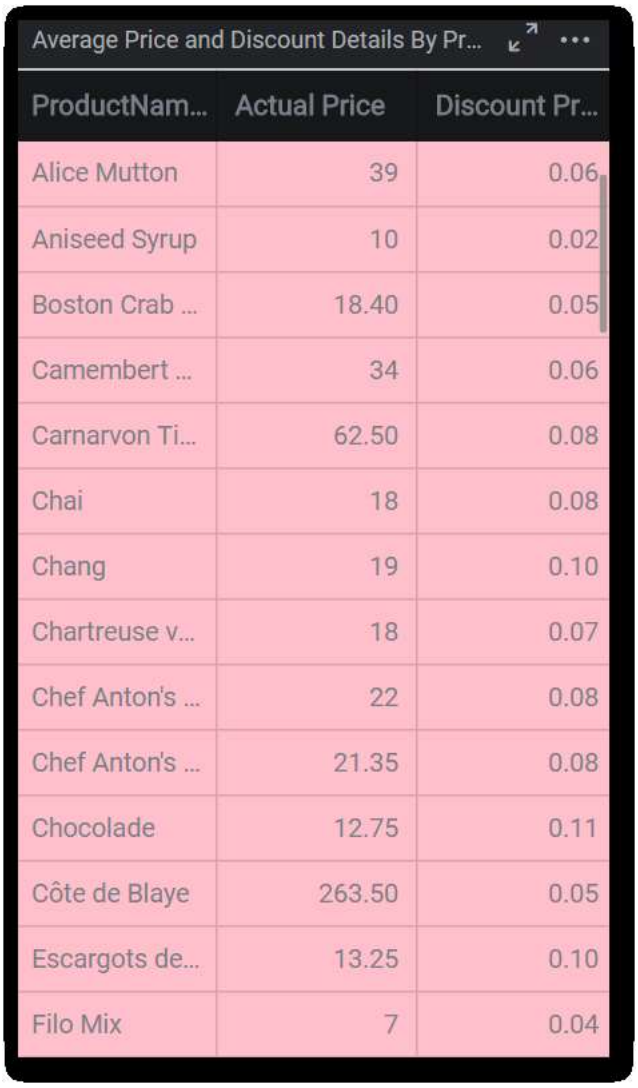
- background-color
- font-family
- font-color

For example, we can change the background color for grid content. Below code snippet represent the to change background color of the grid content.

JS

```
"content":
{
"background-color": "pink",
"font-color": "#7a7e89",
"font-family": "Roboto",
},
```


The following image illustrates the applied background color for grid content.



Hover

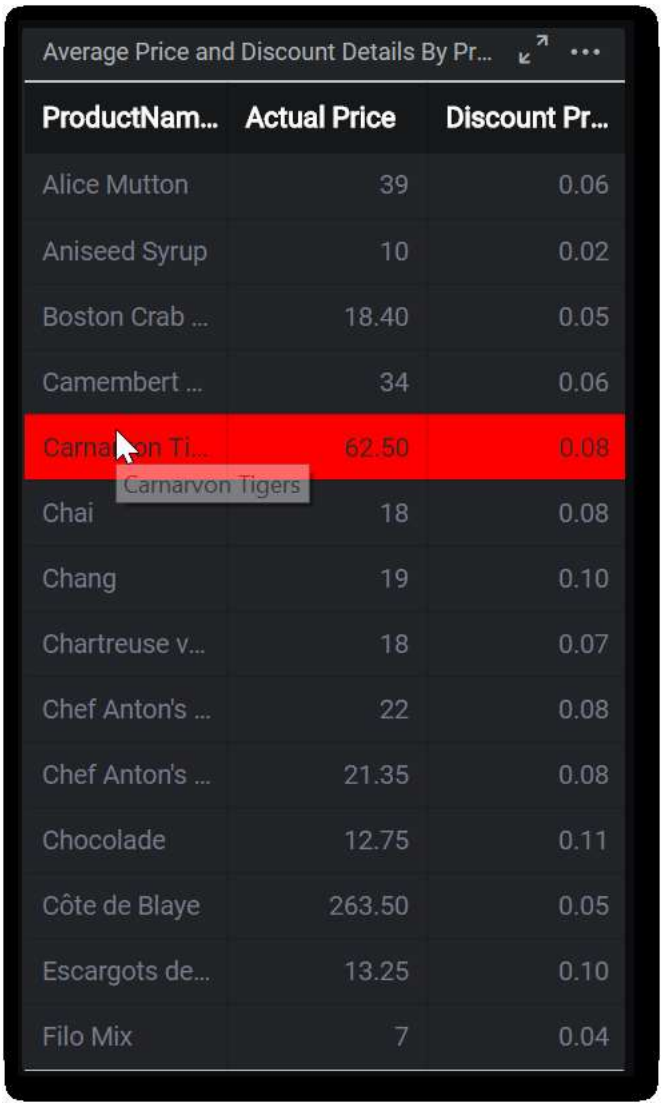
We can customize the 'background-color' for hover section.

For example, while hovering on grid widget background color will be changed as red color. Below code snippet represent the background color for the hover effect.

JS

```
"hover":
{
"background-color": "red",
},
```

The following image illustrates the applied background color for grid hover.



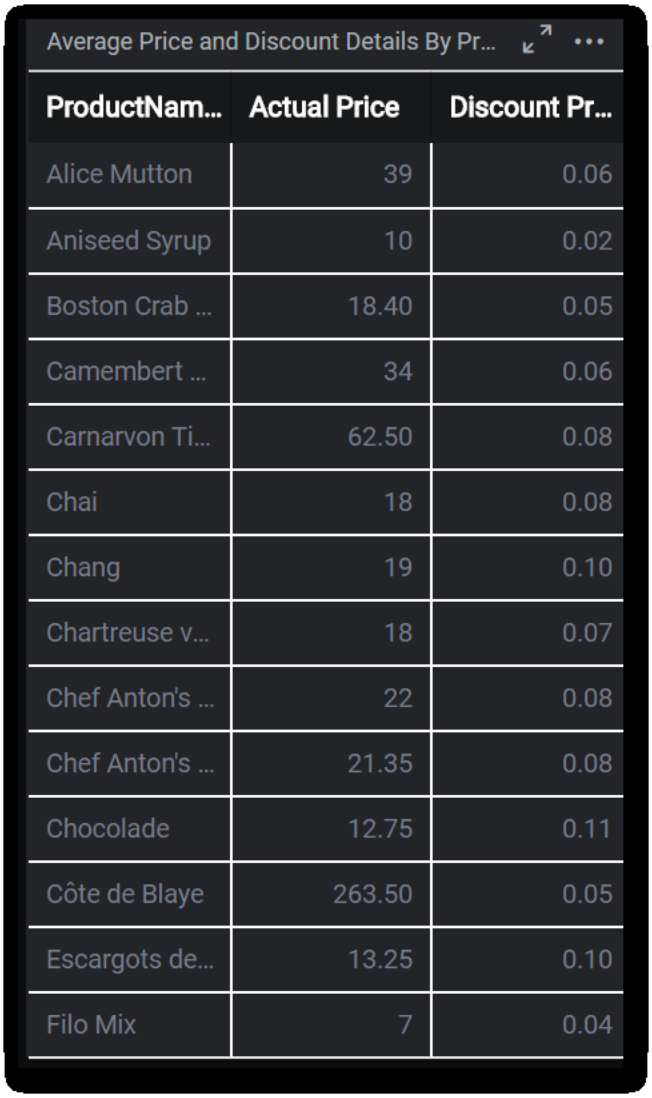
Border

We can customize the border-color for the grid border. Below code snippet represent the border color for grid border.

JS

```
"border":
{
"border-color": "white",
},
```

The following image illustrates the applied border color for grid border.



Selection

We can customize the below properties for selection appearance of grid.

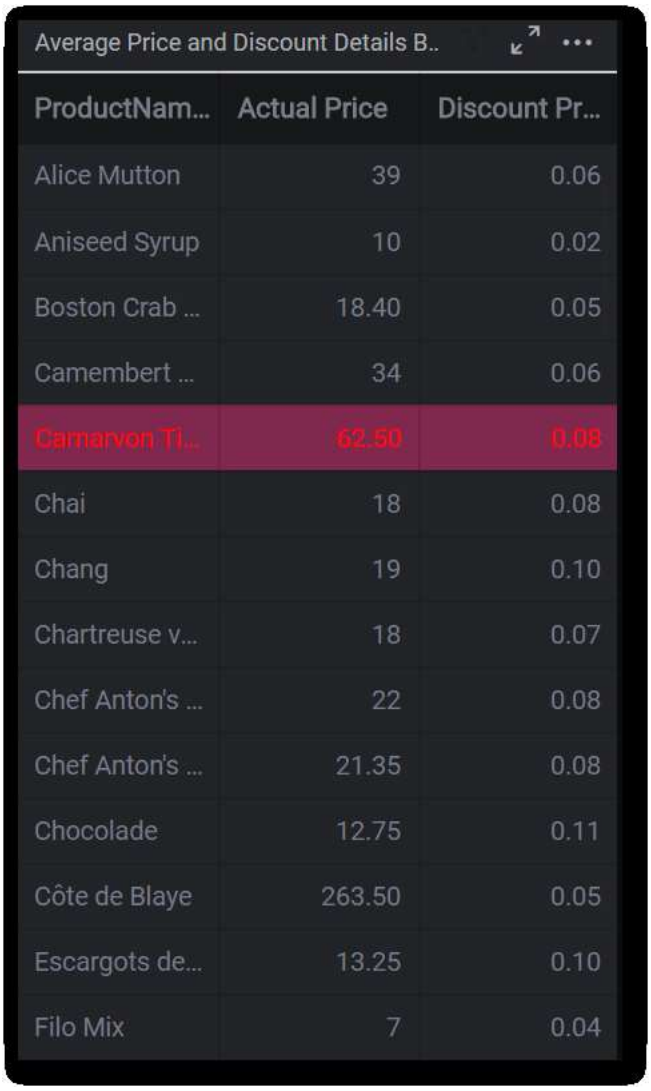
- background-color
- font-color

For example, we can change the font-color for grid selection. Below code snippet represents to change the font color for grid while selection.

JS

```
"selection":
{
"background-color": "#80284d",
"font-color": "red",
},
```

The following image illustrates the applied font color for grid selection.



FontSettings

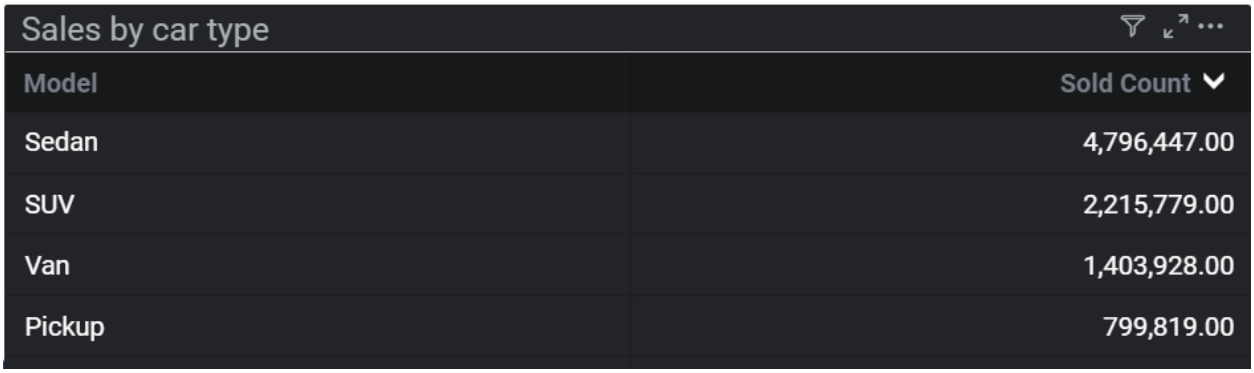
We can customize the 'font-size' property in header, content, paging and summary row section.

For example, we can change the font-size it reflect the grid content, header. Below code snippet represent to change the font size for grid content, header.

JS

```
"fontSettings":
{
"font-size": "16px",
},
```

The following image illustrates the applied font color for grid selection.



PivotGrid

Sections

We can customize the below sections in PivotGrid widget.

- Grouping bar
- Grouping button
- Icon
- Header
- Content
- Summary
- Hover
- FontSettings

Grouping bar

We can customize the below properties in grouping bar section.

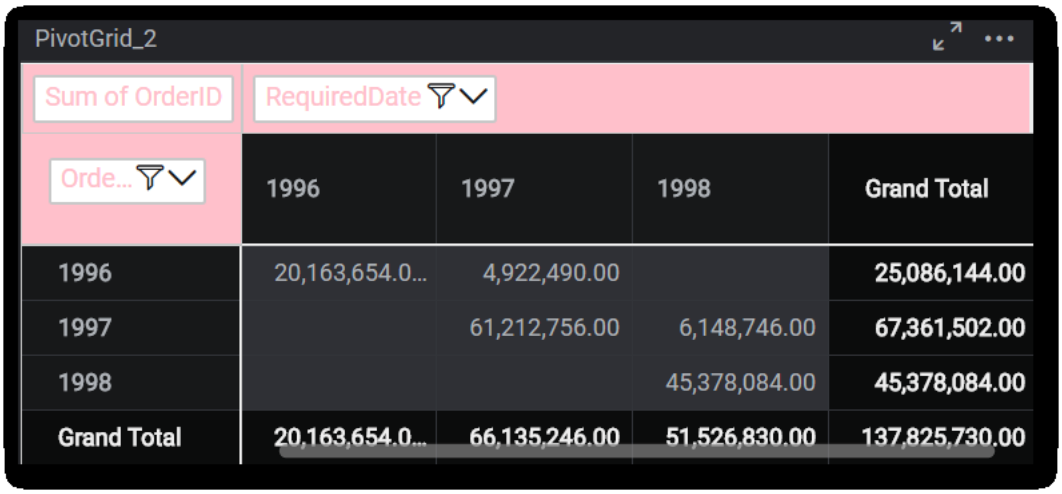
- background-color
- border-color
- font-family
- font-color

For example, we can customize the background color of grouping bar. Below code snippet represent to change the background color of grouping bar.

JS

```
"grouping-bar":
{
"background-color": "pink",
"border-color": "#33353a",
"font-color": "#aeb2b7",
"font-family": "Roboto"
},
```

The following image illustrates the applied background color for PivotGrid grouping bar.



Grouping button

We can customize the below properties in grouping button section.

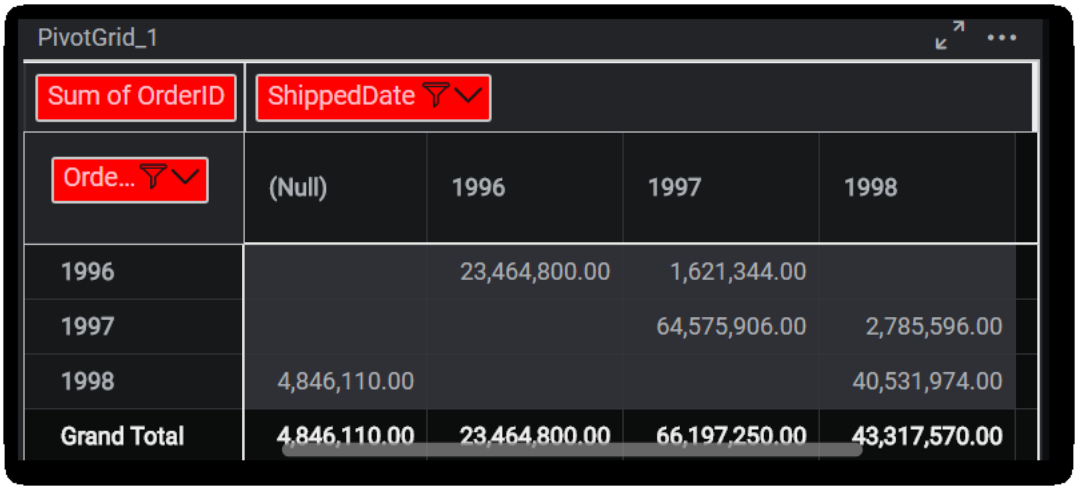
- background-color
- font-color
- font-family

For example, we can customize the background color for grouping button. Below code snippet represents to change the background color of grouping button.

JS

```
"grouping-button":
{
"background-color": "pink",
"font-color": "#fff",
"font-family": "Roboto"
},
```

The following image illustrates the applied background color for PivotGrid grouping button.



Icon

We can customize the below properties in icon section.

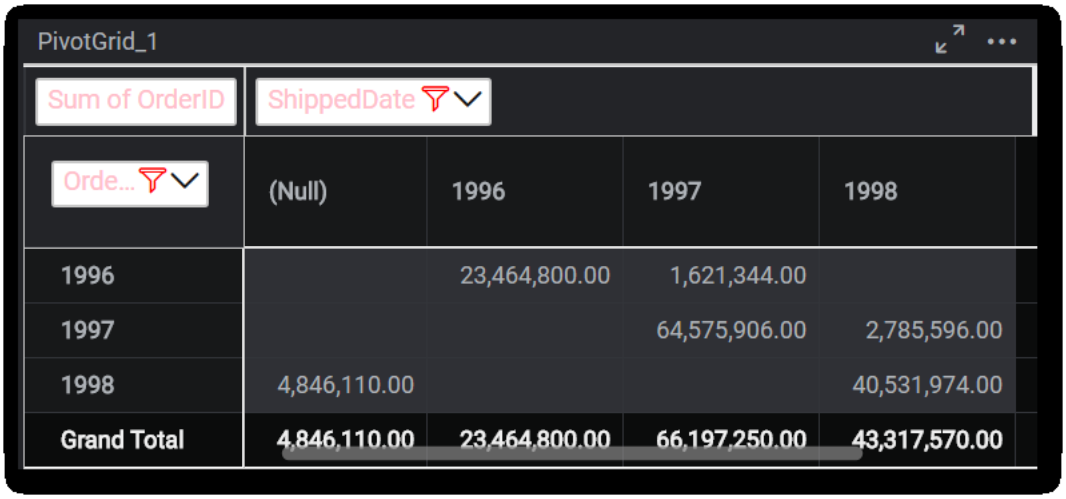
- filter-icon-color
- sort-icon-color
- expand-icon-color
- collapse-icon-color

For example, we can change the filter icon color. Below code snippet represents to change the filter icon color of PivotGrid widget.

JS

```
"icon":
{
"filter-icon-color": "orange",
"sort-icon-color": "#19191c",
"expand-icon-color": "#19191c",
"collapse-icon-color": "#19191c"
},
```

The following image illustrates the applied filter icon color for PivotGrid.



Header

We can customize the below properties in header section

- background-color
- border-color
- font-family
- font-color

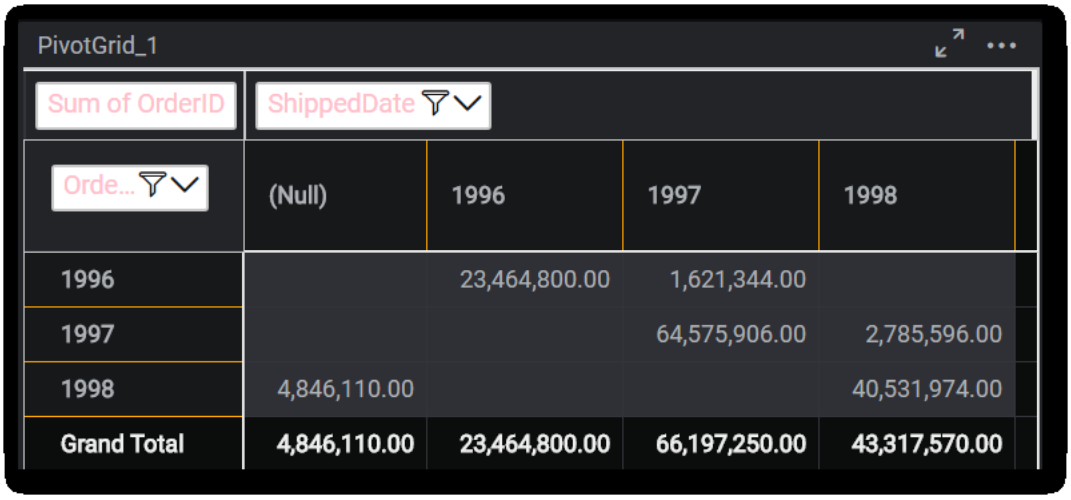
For example, we can customize the border color of header. Below code snippet represents to change the border color of header.

JS

```
"header":
```

```
{
"background-color": "#171819",
"border-color": "orange",
"font-color": "#aeb2b7",
"font-family": "Roboto",
},
```

The following image illustrates the applied border color for PivotGrid header.



Content

We can customize the below properties in content section.

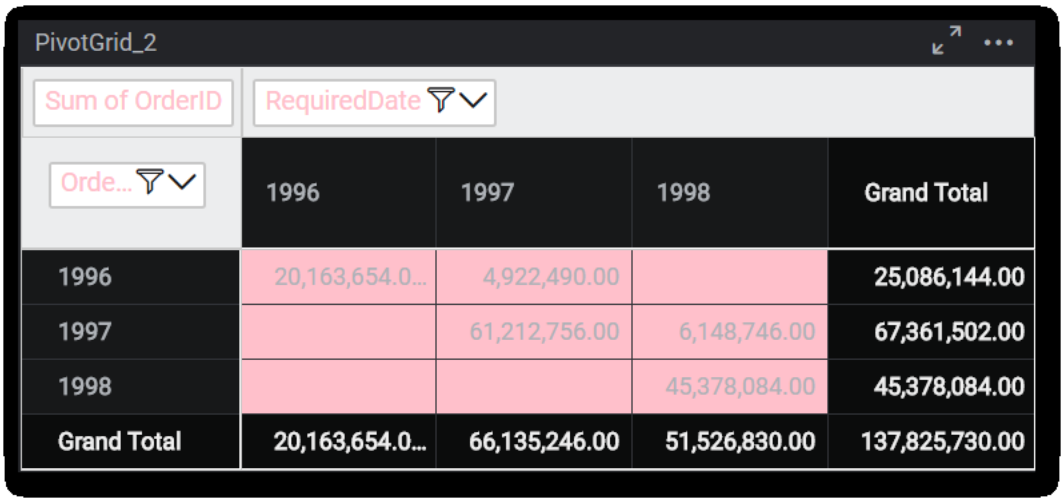
- background-color
- border- color
- font -color
- font-family

For example, we can customize the background color for content. Below code snippet represents to change the background color of content.

JS

```
"content":
{
"background-color": "pink",
"border-color": "#33353a",
"font-color": "#aeb2b7",
"font-family": "Roboto",
},
```

The following image illustrates the applied background color for PivotGrid content.



Summary

We can customize the below properties for summary section.

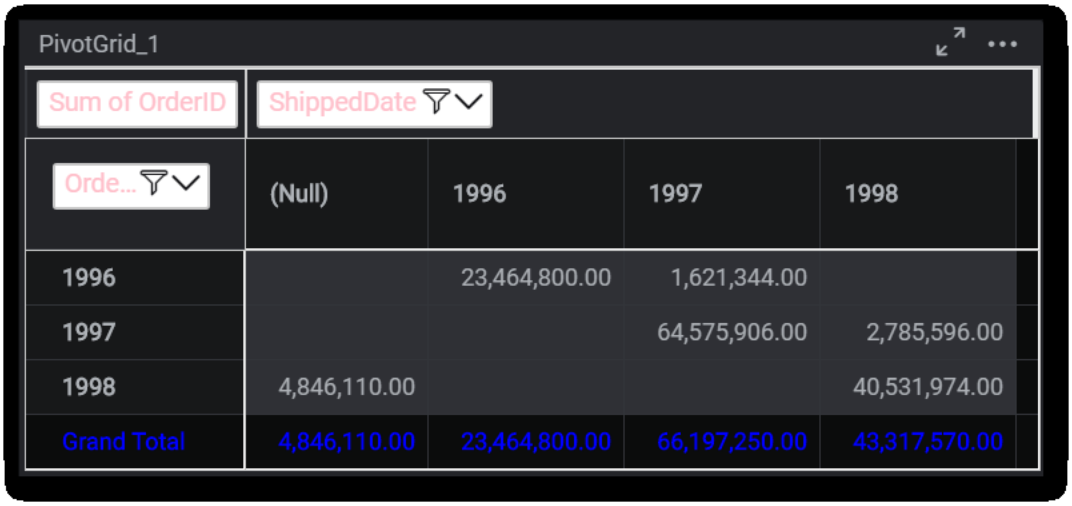
- background-color
- border- color
- font-color
- font-family

For example, we can change the font color of summary. Below code snippet represent to change the font color for summary.

JS

```
"summary":
{
"background-color": "#0b0c0c",
"border-color": "#33353a",
"font-color": "blue",
"font-family": "Roboto",
},
```

The following image illustrates the applied font color for PivotGrid summary.



Hover

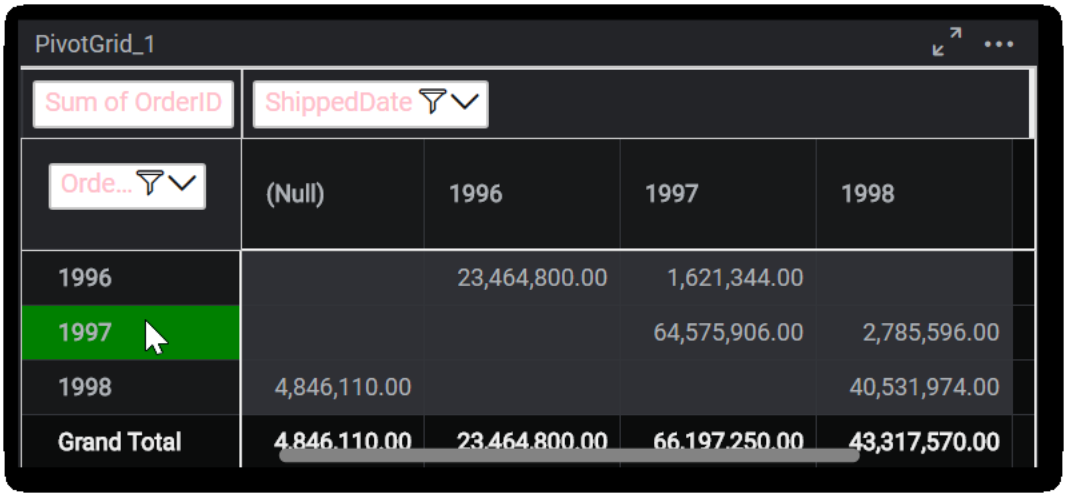
We can customize the 'background-color' of hover for PivotGrid widget.

For example, we can customize the background color of hover. Below code snippet represents to change the background color of hover effect.

JS

```
"hover":
{
"background-color": "green",
},
```

The following image illustrates the applied background color for pivotgrid hover.



FontSettings

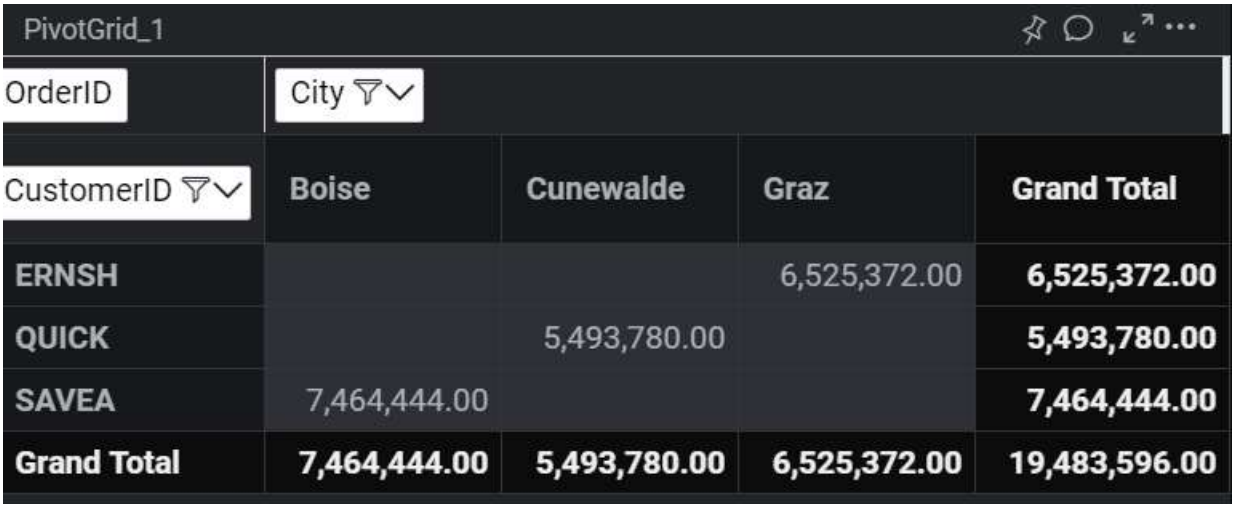
We can customize the 'font-size' for header, content, tooltip, grouping button text and summary.

For example, we can customize the font size of pivot grid. Below code snippet represents to change the font size of header, content, tooltip, grouping button text and summary.

JS


```
"fontSettings":
{
"font-size":"16px"
},
```

The following image illustrates the applied font size for pivotgrid hover.



Card

Sections

We can customize the below sections in card widget.

- Card
- Title
- Value
- Variation
- Selection
- FontSettings

Card

We can customize the below properties in card widget.

- background-color
- border-color

For example, we can customize the background color of card widget. Below code snippet represent to change the background color of card widget.

JS

```
"card":
{
"background-color": "pink",
"border-color": "#3c3d42",
},
```


The following image illustrates the applied background color for card.

Title

We can customize the below properties in title section

- font-color
- font-family

For example, we can customize the font color of title. Below code snippet represent to change the font color of title.

JS

```
"title":
{
"font-color": "blue",
"font-family": "Roboto",
},
```

The following image illustrates the applied font color for card title.

Value

We can customize the below properties in value section

- font-color
- font-family

For example, we can customize the font family for value section. Below code snippet represent to change the font family.

JS

```
"value":
{
"font-color": "#01c4e0",
"font-family": "Times New Roman"
},
```

The following image illustrates the applied font family for card value.

Variation

We can customize the below properties in variation section.

- font-color
- font-family

For example, we can customize the font color for variation section. Below code snippet represent to change the font color.

JS

```
"variation":
{
"font-color": "green",
"font-family": "Roboto",
},
```

The following image illustrates the applied font color for card variation.

Selection

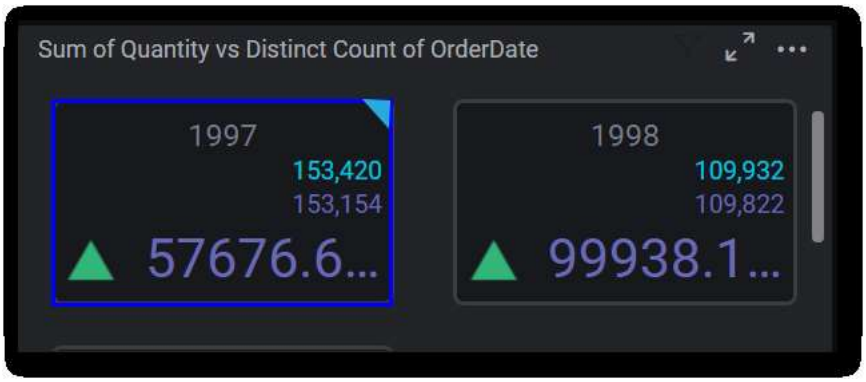
We can customize the highlight-color property in selection section.

For example, we can customize the highlight color for selection. Below code snippet represent to change the highlight color.

JS

```
"selection":
{
"highlight-color": "blue"
},
```

The following image illustrates the applied highlight color for card selection appearance.



FontSettings

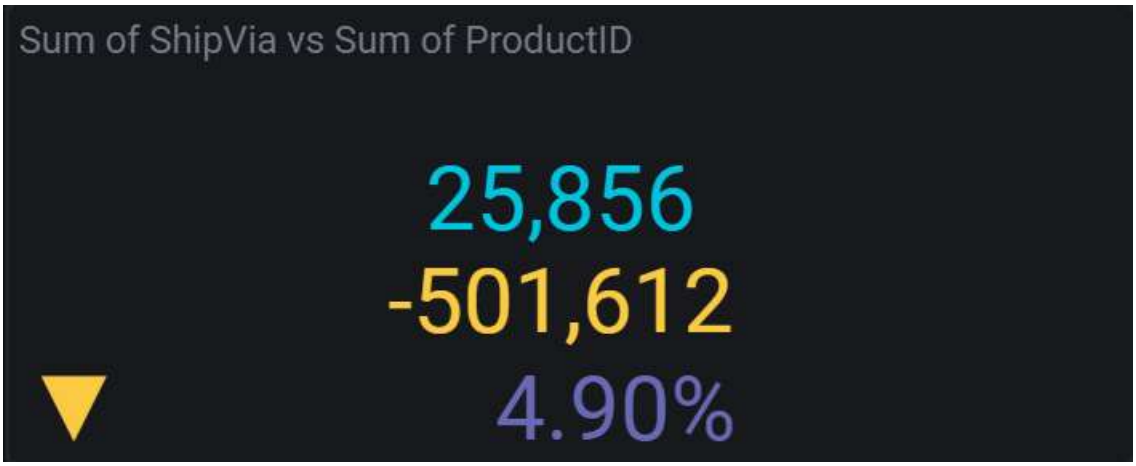
We can customize the font-size property in card widget.

For example, we can customize the font-size of value, variation section of card widget. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size":"45px"
},
```

The following image illustrates the applied font size for card.



FilterComboBox

Sections

We can customize the below sections in FilterComboBox widget.

- Combobox
- Button
- DropDownList
- Icon
- Hover
- SearchBox
- Selection

- FontSettings

Combobox

We can customize the below properties in combobox section.

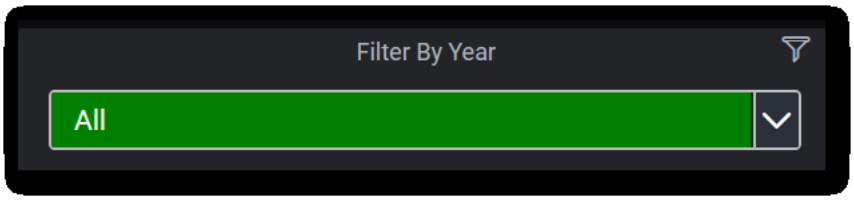
- background-color
- border-color
- font-color
- font-family

For example, we can customize the background color for combobox. Below code snippet represent to change the background color.

JS

```
"combobox":
{
"background-color": "green",
"border-color": "#3c3d42",
"font-color": "#f9f9f9",
"font-family": "Roboto",
},
```

The following image illustrates the applied background color for combobox



Button

We can customize the below properties in button section.

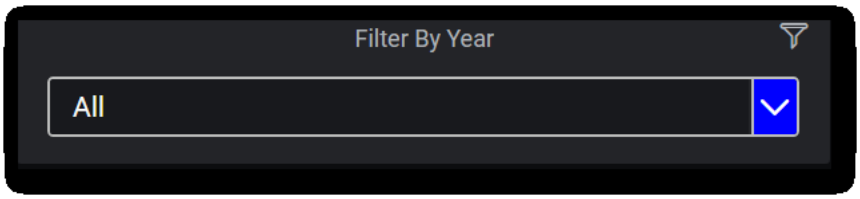
- background-color
- font-color
- border-color

For example, we can customize the background color for combobox. Below code snippet represent to change the background color.

JS

```
"button":
{
"background-color": "blue",
"font-color": "#ffffff",
"border-color": "none",
},
```

The following image illustrates the applied background color for dropdown button.



DropDownList

We can customize the below properties in dropdownlist section.

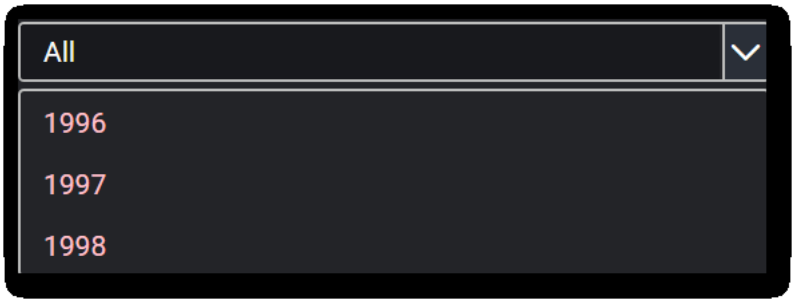
- background-color
- font-color
- font-family

For example, we can customize the font color for DropDownList. Below code snippet represent to change the font color.

JS

```
"dropdownlist":
{
"background-color": "#232428",
"font-color": "pink",
"font-family": "Roboto",
},
```

The following image illustrates the applied font color for DropDownList



Icon

We can customize the below properties in icon section. This section is used for multi selection.

- Background-color
- Border-color
- tick-color
- search-icon-color

For example, we can customize the background color for icon. Below code snippet represent to change the background color.

JS

```
"icon":
{
"background-color": "green",
```



```
"border-color": "#ffffff",
"tick-color": "#ffffff",
"search-icon-color": "white"
},
```

The following image illustrates the applied background color for dropdown icon.

Hover

We can customize the 'background-color' property while hovering.

For example, we can customize the background color while hovering. Below code snippet represent to change the background color.

JS

```
"hover":
{
"background-color": "red"
},
```

The following image illustrates the applied background color for DropDownList hovering effect.

SearchBox

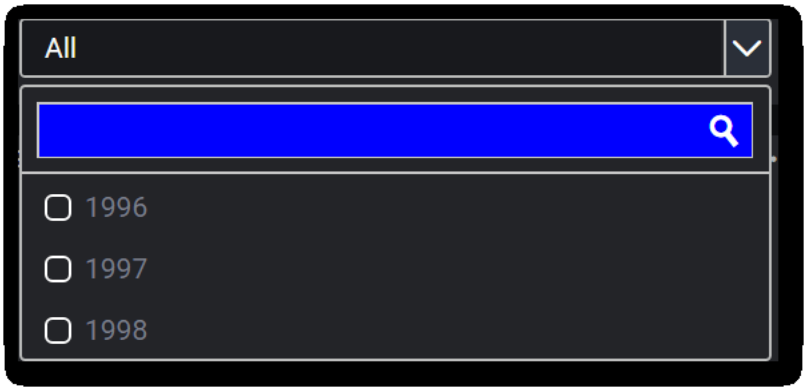
We can customize the 'background-color' property in searchbox section

For example, we can customize the background-color in searchbox section. Below code snippet represent to change the background color.

JS

```
"searchbox":
{
"background-color": "blue"
},
```

The following image illustrates the applied background color DropDownList search box



Selection

We can customize the 'background color' property in selection

For example, we can customize the background color in selection. Below code snippet represent to change the background color.

JS

```
"selection":
{
"background-color": "red"
},
```

The following image illustrates the applied background color for selection appearance in DropDownList.

FontSettings

We can customize the 'font size' property in combobox section.

For example, we can customize the font size of combobox dropdown list and default selected value. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
```



```
},
```

The following image illustrates the applied font size for combobox.

Checkbox

Sections

We can customize the below sections in checkbox widget.

- Label
- Icon
- FontSettings

Label

We can customize the below properties in label section

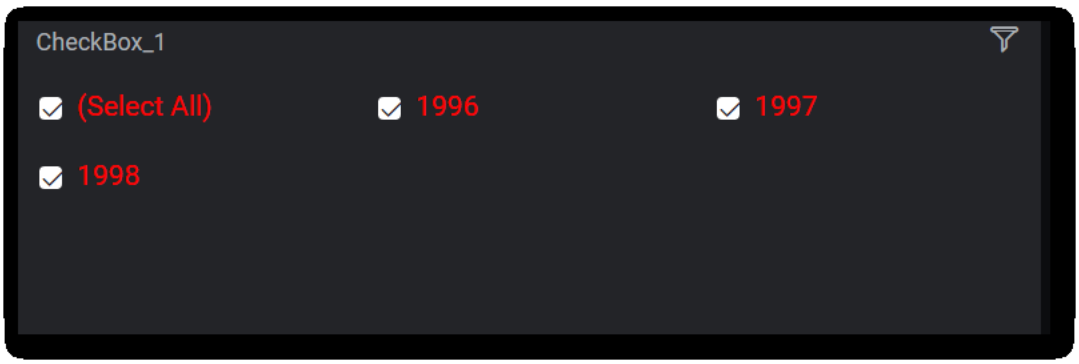
- font-color
- font-family

For example, we can customize the font color in label. Below code snippet represent to change the font color.

JS

```
"label":
{
"font-color": "red",
"font-family": "Roboto",
},
```

The following image illustrates the applied font color for checkbox label.



Icon

We can customize the below properties in icon section.

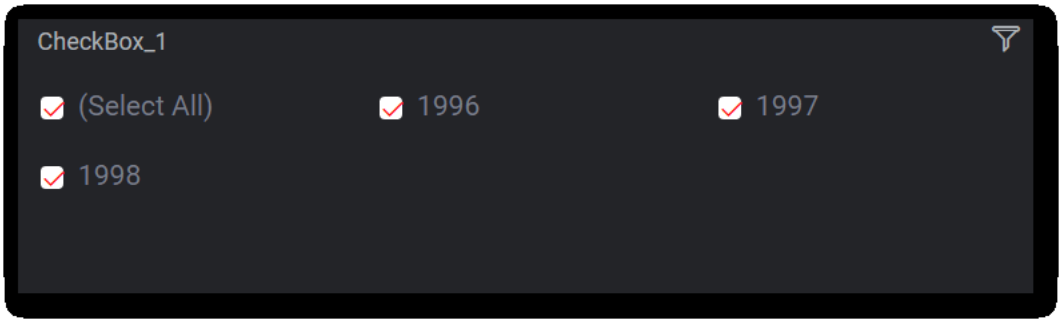
- tick-color
- border-color
- background-color

For example, we can customize the tick color in checkbox icon. Below code snippet represent to change the tick color.

JS

```
"icon":
{
"tick-color": "red",
"background-color": "white",
"border-color": "#232428",
},
```

The following image illustrates the applied tick color for icon.



FontSettings

We can customize the 'font size' property in label.

For example, we can customize the font size in label. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
```

```
},
```

The following image illustrates the applied font size for checkbox label.

Note: Same theme settings was applied for RadioButton also.

RangeSlider

Sections

We can customize the below sections in range slider.

- Container
- Range
- Handle
- Label
- FontSettings

Container

We can customize the background-color property in container section.

For example, we can customize the background color for range slider container. Below code snippet represent to change the background color.

JS

```
"container":
{
"background-color": "red",
},
```

The following image illustrates the applied background color RangeSlider container.

Range

We can customize the below properties in range section.

- Range-color
- Background-color

For example, we can customize the range color for range section. Below code snippet represent to change the range color.

JS

```
"range":
{
"range-color": "blue",
"background-color": "#2f3035"
},
```

The following image illustrates the applied range color for range slider.

Handle

We can customize the below properties in handle section.

- background-color
- border-color

For example, we can customize the background color for handle in range slider. Below code snippet represent to change the background color.

JS

```
"handle":
{
"background-color": "red",
"border-color": "#353535",
},
```

The following image illustrates the applied background color for range slider handle.

Label

We can customize the font-color for the label.

Below code snippet represent to change the font color.

JS

```
"label":
{
"font-color": "green"
},
```

The following image illustrates the applied font color for range slider label.

FontSettings

We can customize the 'font-size' property in range section.

For example, we can customize the font size for range section. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```

The following image illustrates the applied font size for range slider.

Listbox

Sections

We can customize the below sections in listbox widget.

- Listbox

- Icon
- Hover
- FontSettings

Listbox

We can customize the below properties in listbox section

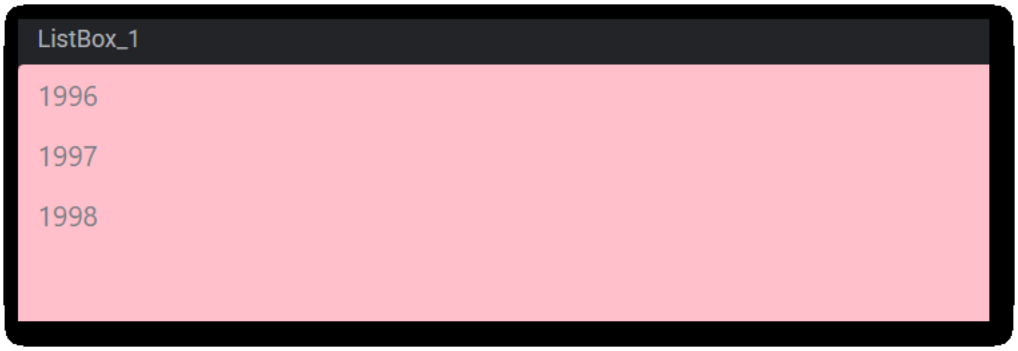
- background-color
- color

For example, we can customize the background color for listbox. Below code snippet represent to change the background color.

JS

```
"listbox":
{
"background-color": "pink",
"color": "#7a7e89",
},
```

The following image illustrates the applied background color for listbox.



Icon

We can customize the below properties in icon section.

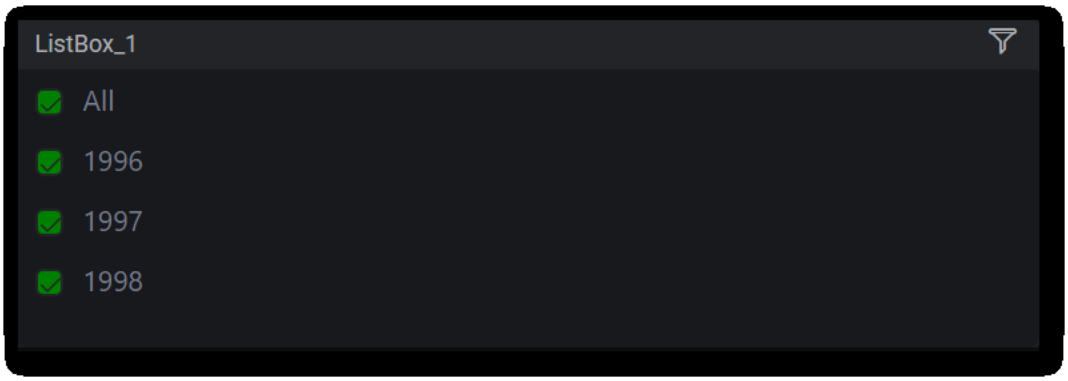
- background-color
- border-color
- tick-color

For example, we can customize the background color for listbox icon while enable multi selection. Below code snippet represent to change the background color.

JS

```
"icon":
{
"background-color": "green",
"border-color": "#232428",
"tick-color": "#232428",
},
```


The following image illustrates the applied background color for listbox multi select icon.



Hover

We can customize the 'background-color' while hovering.

For example, we can customize the background color for listbox while hovering. Below code snippet represent to change the background color.

JS

```
"hover":
{
"background-color": "red",
},
```

The following image illustrates the applied background color for listbox hover.

FontSettings

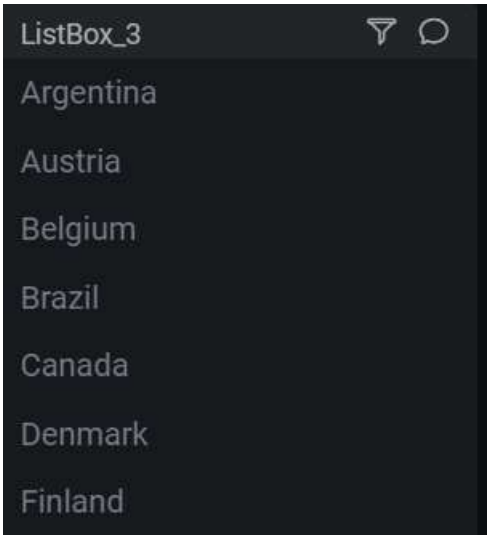
We can customize the 'font-size' property in listbox section

For example, we can customize the font size for listbox. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```


The following image illustrates the applied font size for listbox.



Datepicker

Sections

We can customize the below sections in datepicker widget.

- Textbox
- Calendar
- Selection
- Hover
- Icon
- Footer
- Button
- FontSettings

TextBox

We can customize the below properties in textbox section.

- background-color
- border-color
- font-color

For example, we can customize the border color for textbox in datepicker. Below code snippet represent to change the border color.

JS

```
"textbox":
{
"background-color": "#18191d",
"border-color": "red",
"font-color": "f9f9f9"
},
```

The following image illustrates the applied border color for datepicker textbox.

Calendar

We can customize the below properties in calendar section.

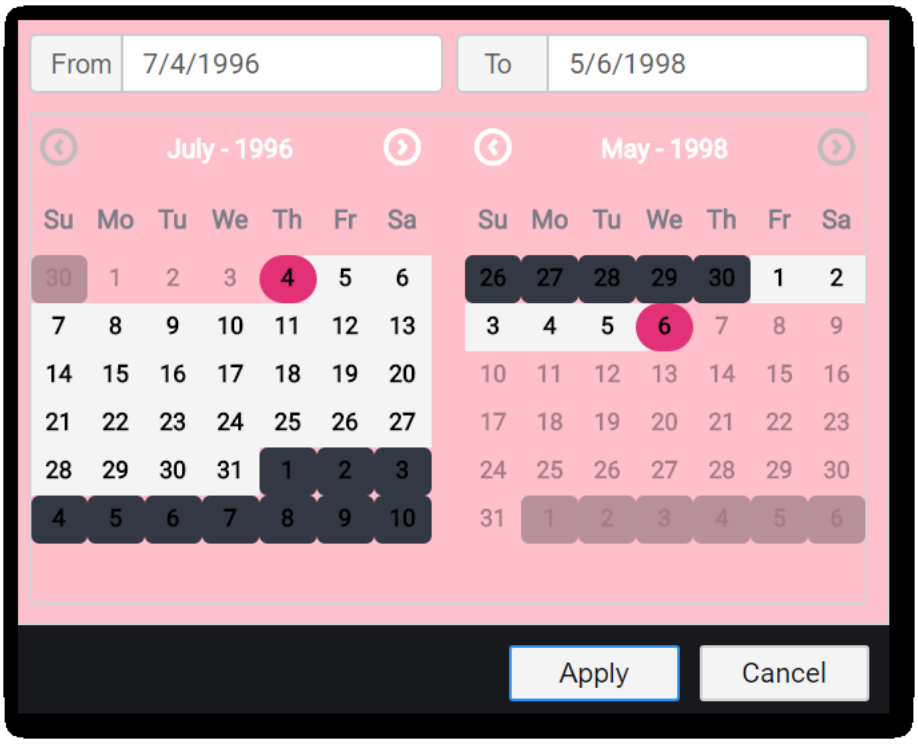
- background-color
- border-color
- currentmonth-font-color
- week-header-color
- days-color
- othermonth-font-color

For example, we can customize the background color for calendar in datepicker. Below code snippet represent to change the background color.

JS

```
"calendar":
{
"background-color": "pink",
"border-color": "#333842",
"currentmonth-font-color": "white",
"week-header-color": "#7a7e89",
"days-color": "white",
"othermonth-font-color": "#333842"
},
```

The following image illustrates the applied background color for calendar.



Selection

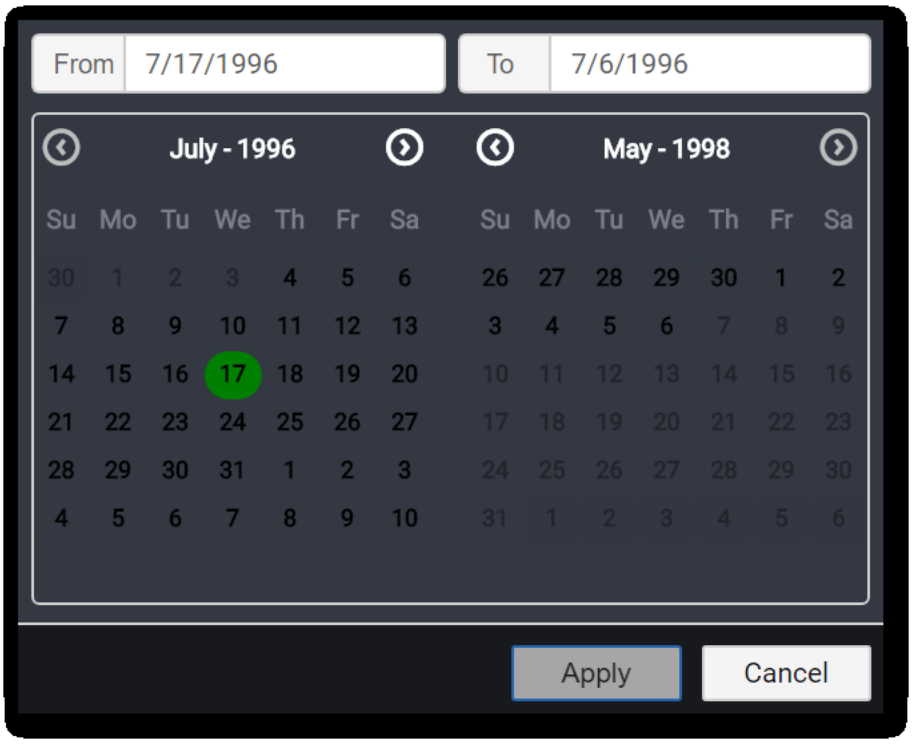
We can customize the 'background-color' for selection

For example, we can customize the background color for selection in datepicker. Below code snippet represent to change the background color.

JS

```
"selection":
{
"background-color": "green"
},
```

The following image illustrates the applied background color for selection appearance .



Hover

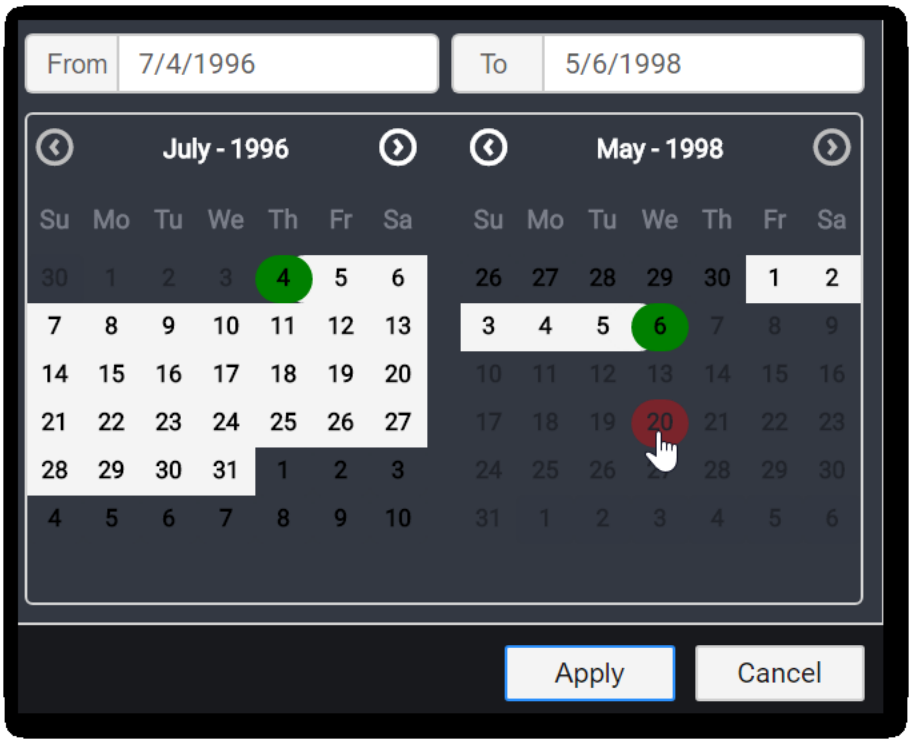
We can customize the 'background-color' for hovering.

For example, we can customize the background color while hovering in datepicker. Below code snippet represent to change the background color.

JS

```
"hover":
{
"background-color": "red"
},
```

The following image illustrates the applied background color for hovering effect.



Icon

We can customize the below properties in icon section

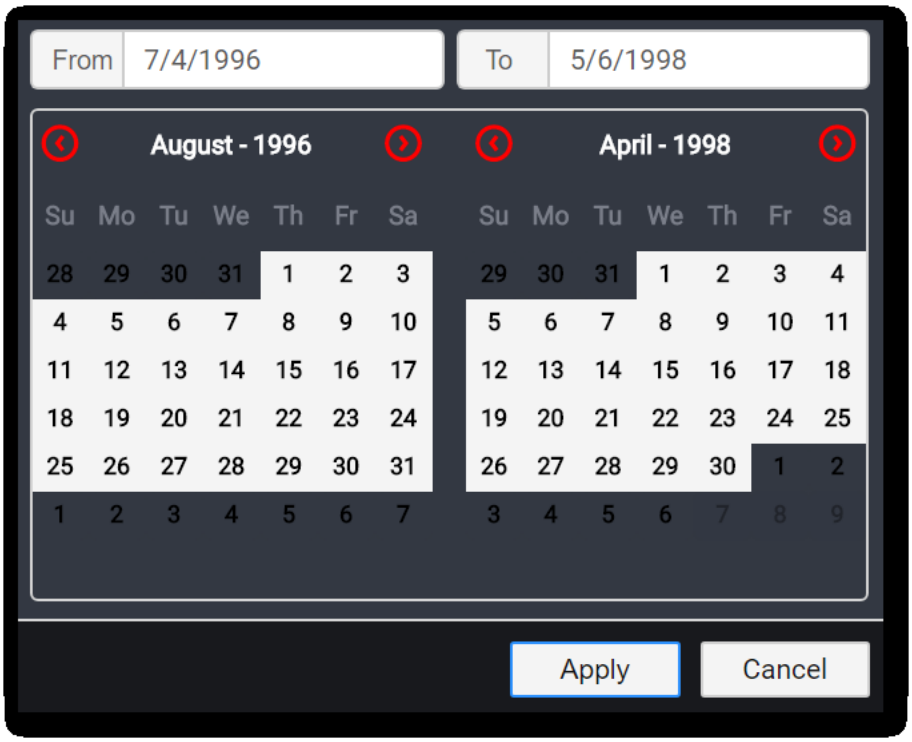
- arrow-color
- background-color

For example, we can customize the arrow color in datepicker. Below code snippet represent to change the arrow color.

JS

```
"icon":
{
"arrow-color": "red",
"calendar-color": "#ffffff",
},
```

The following image illustrates the applied arrow color for datepicker.



Footer

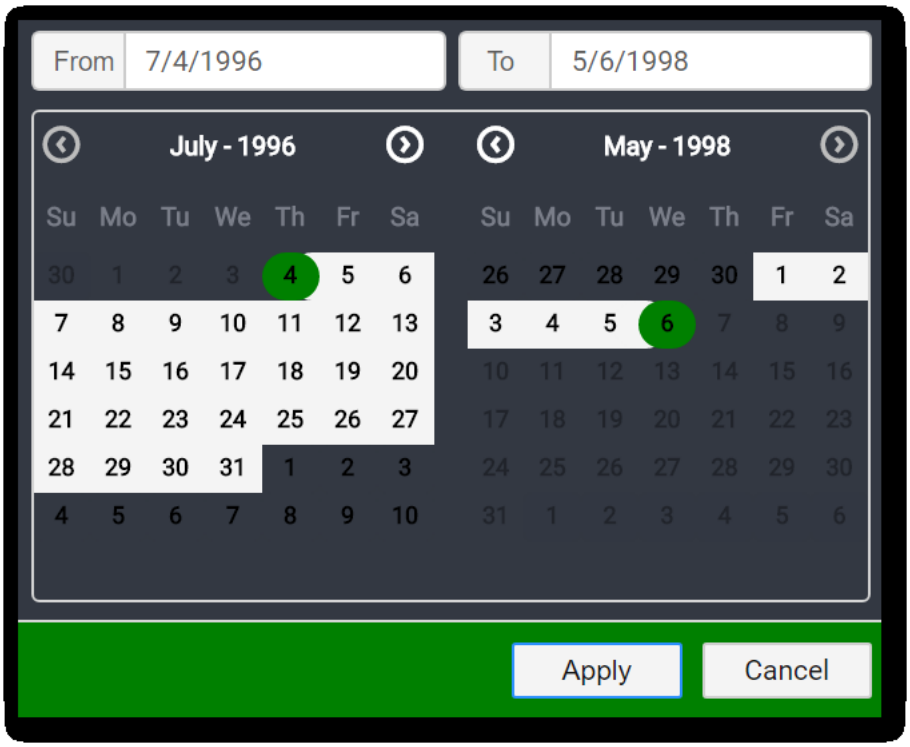
We can customize the 'background-color' for footer.

For example, we can customize the background color for footer in datepicker widget. Below code snippet represent to change the background color.

JS

```
"footer":
{
"background-color": "green",
},
```

The following image illustrates the applied footer color for datepicker.



Button

We can customize the below properties in button section.

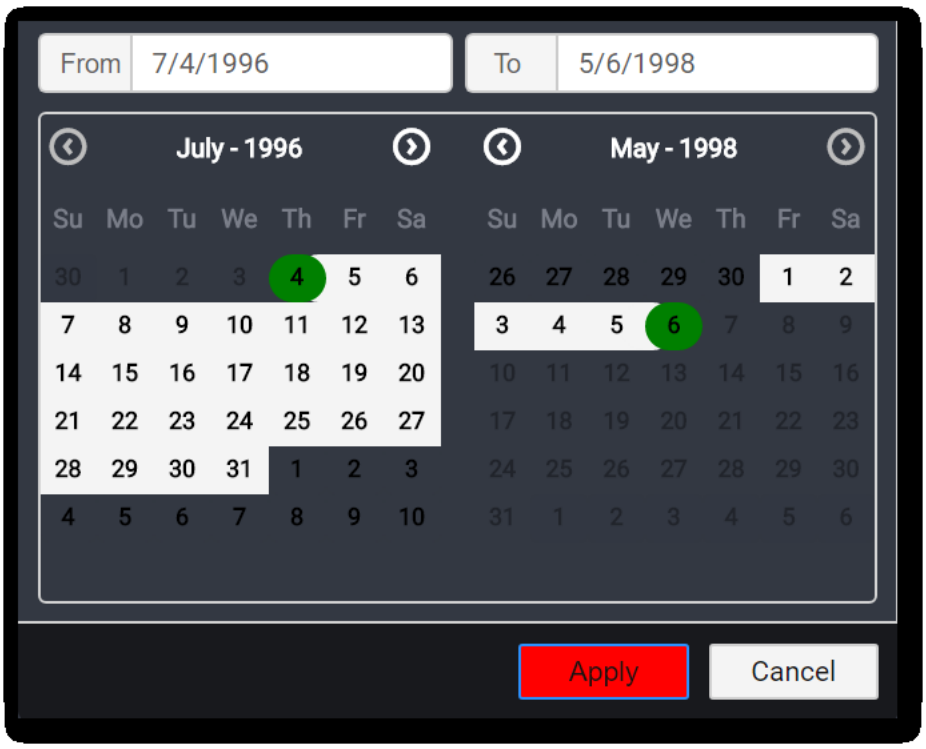
- applybutton-color
- applybutton-font-color
- cancelbutton-color
- cancelbutton-font-color

For example, we can customize the apply button color. Below code snippet represent to change the apply font color.

JS

```
"button":
{
"applybutton-color": "red",
"applybutton-font-color": "white",
"cancelbutton-color": "#f4f4f4",
"cancelbutton-font-color": "#161616"
},
```

The following image illustrates the applied apply button color for datepicker.



FontSettings

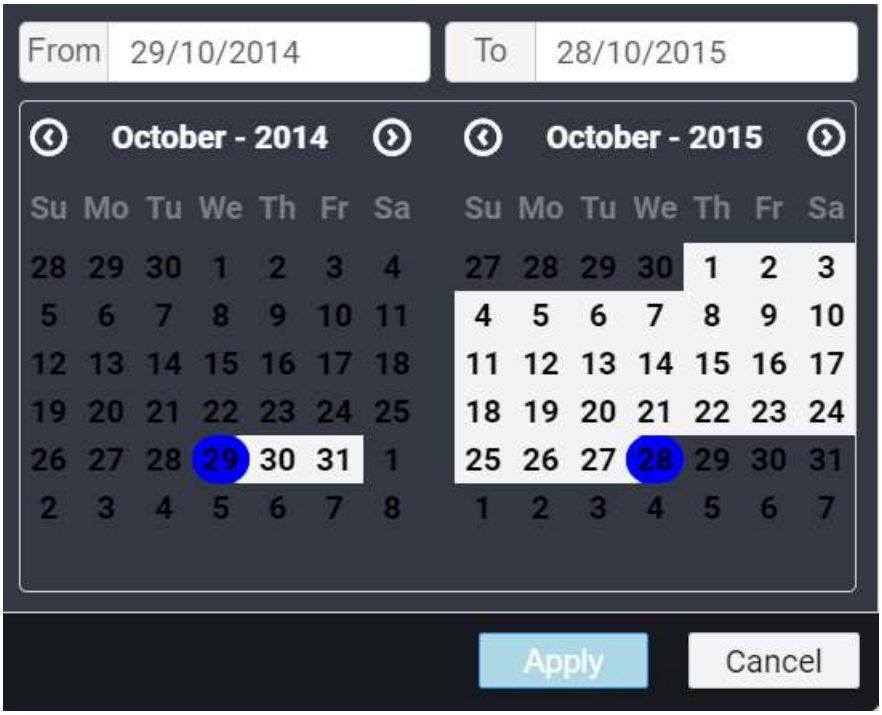
We can customize the 'font-size' property in datepicker section.

For example, we can customize the font size for textbox and calendar in datepicker. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```

The following image illustrates the applied font size for datepicker.



Bubble map

Sections

We can customize the below sections in bubble map.

- Shape
- Hover
- Bubble
- FontSettings

Shape

We can customize the below properties in shape section.

- background-color
- border-color

For example, we can customize the background color for shape. Below code snippet represent to change the background color.

JS

```
"shape":
{
"background-color": "green",
"border-color": "#181a1e",
},
```

The following image illustrates the applied background color for map shape

Hover

We can customize the below properties while hovering

- background-color
- border-color

For example, we can customize the border color while hovering. Below code snippet represent to change the border color.

JS

```
"hover":
{
"background-color": "red",
"border-color": "blue",
},
```

The following image illustrates the applied border color for hovering effect.

Bubble

We can customize the 'background-color' for bubble.

For example, we can customize the background color. Below code snippet represent to change the background color.

JS

```
"bubble":
{
"background-color": "green",
},
```

The following image illustrates the applied background color for map bubbles.

FontSettings

We can customize the 'font-size' property in bubble map section.

For example, we can customize the font size for value label and tooltip. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```

The following image illustrates the applied font size for bubble map

ChoroplethMap

Sections

We can customize the below sections in ChoroplethMap widget.

- Shape
- Hover
- Range
- Legend
- FontSettings

Shape

We can customize the below properties in shape section.

- background-color
- border-color

For example, we can customize the background color for shape. Below code snippet represent to change the background color.

JS

```
"shape":
{
"background-color": "pink",
"border-color": "#181a1e",
},
```

The following image illustrates the applied background color for map shape.

Hover

We can customize the below properties while hovering.

- Background-color
- Border-color

For example, we can customize the border color while hovering. Below code snippet represent to change the border color.

JS

```
"hover":
{
"background-color": "red",
"border-color": "green",
},
```

The following image illustrates the applied border color for hovering.

Range

We can customize the 'background-color' for range. For this we need to enable equal color mapping option.

For example, we can customize the background color for range. Below code snippet represent to change the background color.

JS

```
"range":
{
"background-color": ["red", "blue", "green", "pink"]
},
```

The following image illustrates the applied background color for range.

Legend

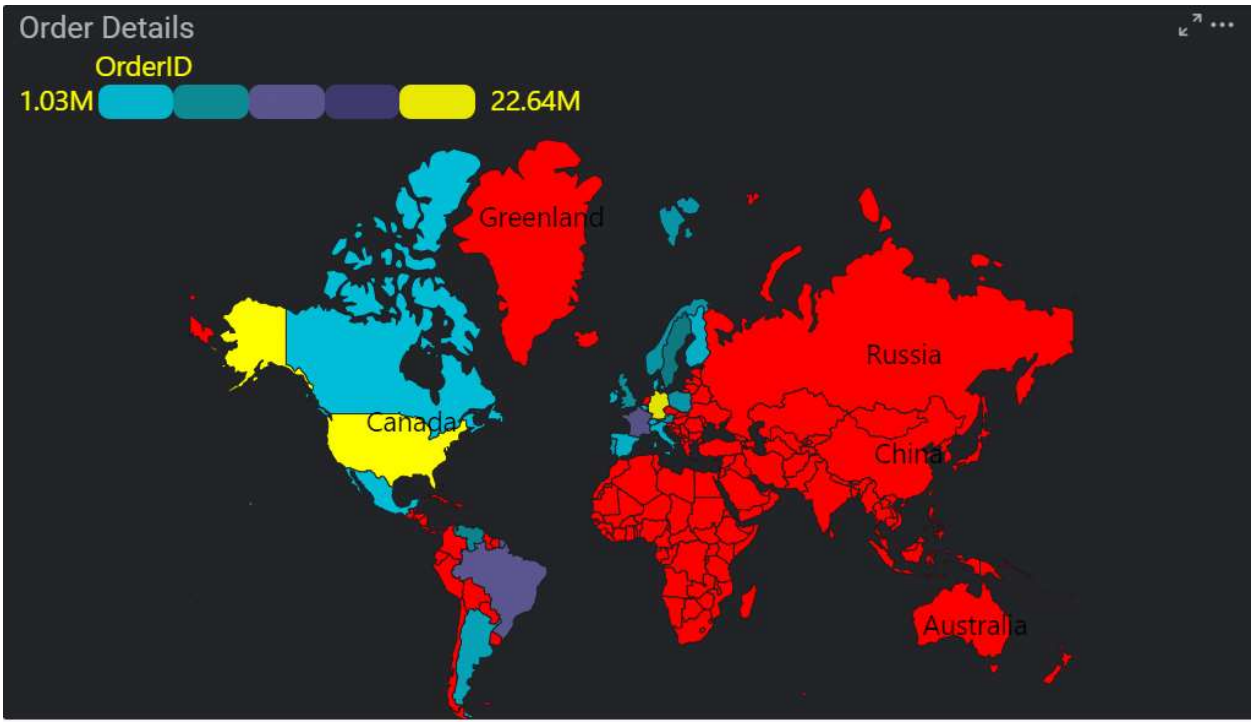
We can customize the font color property for treemap legend.

For example, we can customize the font color for legend. Below code snippet represent to change the font color.

JS

```
"legend":
{
"legendfont-color":"yellow"
},
```

The following image illustrates the applied font color for legend.



FontSettings

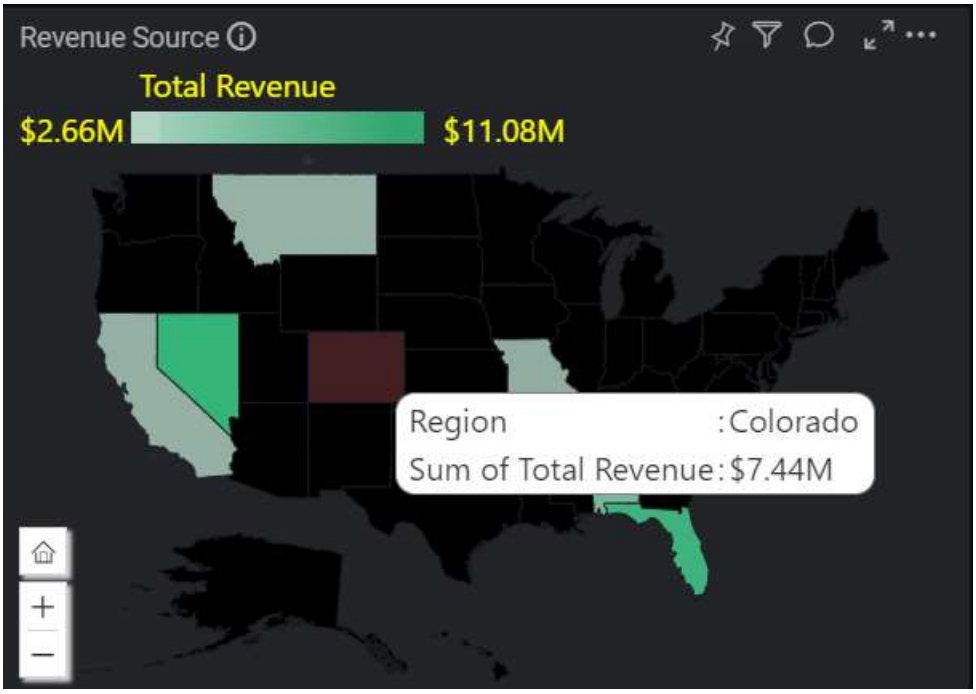
We can customize the 'font-size' property in map section.

For example, we can customize the font size for value label, tooltip and legend settings. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```

The following image illustrates the applied font size for map.



TreeMap

Sections

We can customize the below sections in TreeMap widget.

- DrillDownHover
- Select
- DrillDownHeader
- RangeColorMap
- DesaturateColor
- FontSettings

DrillDownHover

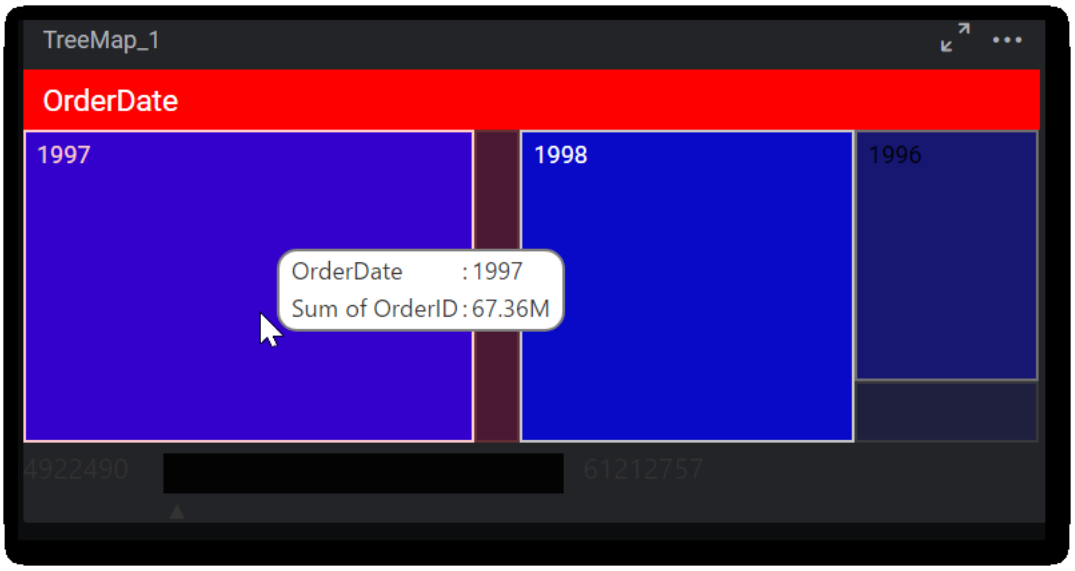
We can customize the 'background color' property in drilldownhover section.

For example, we can customize the background color for drilldown hover. Below code snippet represent to change the background color.

JS

```
"drilldownhover":
{
"background-color": "red",
},
```

The following image illustrates the applied background color for TreeMap drilldown hover.



Select

We can customize the 'border-color' property for selection appearance.

For example, we can customize the border color. Below code snippet represent to change the border color.

JS

```
"select":
{
"border-color": "red",
},
```

The following image illustrates the applied border color for selection appearance.

DrillDownHeader

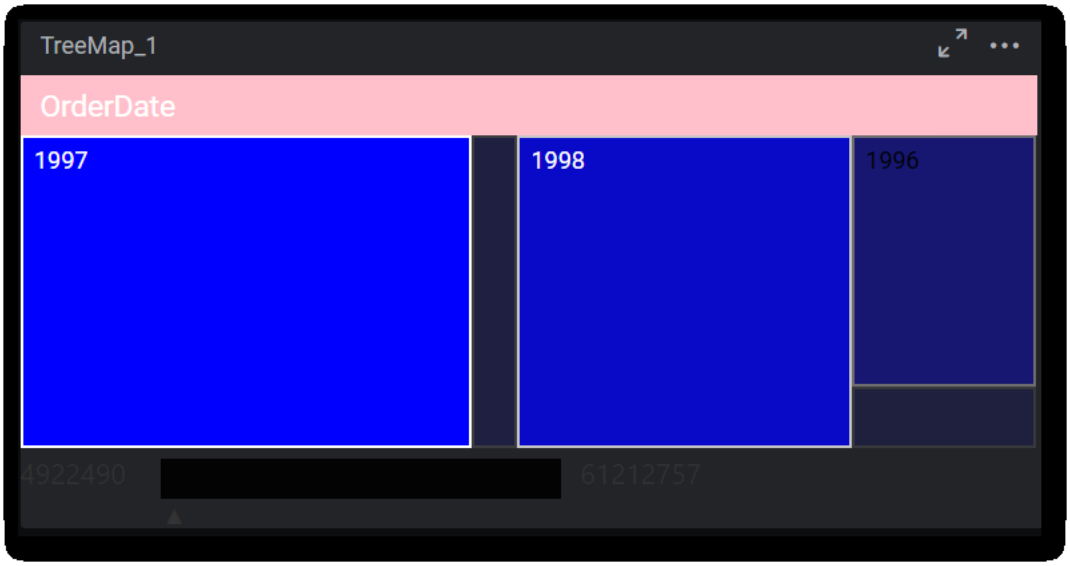
We can customize the 'background-color' property for drilldown header.

For example, we can customize the background color. Below code snippet represent to change the background color.

JS

```
"drilldownheader":
{
"background-color": "pink",
},
```

The following image illustrates the applied background color for drilldown header.



RangeColorMap

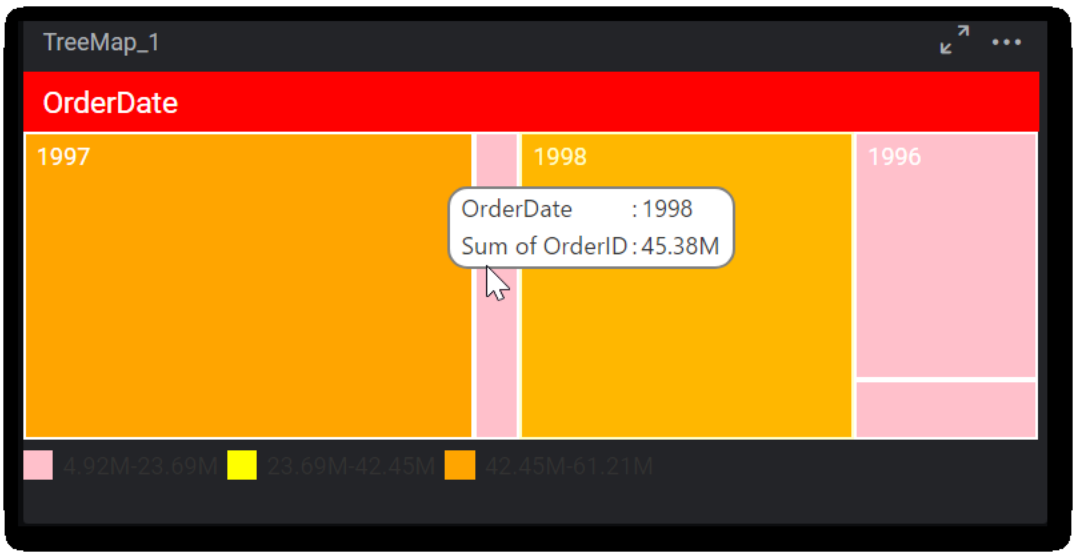
We can customize the 'background-color' for range color map.

For example, we can customize the background color. Below code snippet represent to change the background color.

JS

```
"rangecolormap":
{
"background-color": ["pink", "yellow", "orange"],
},
```

The following image illustrates the applied background color for range color map.



DesaturateColor

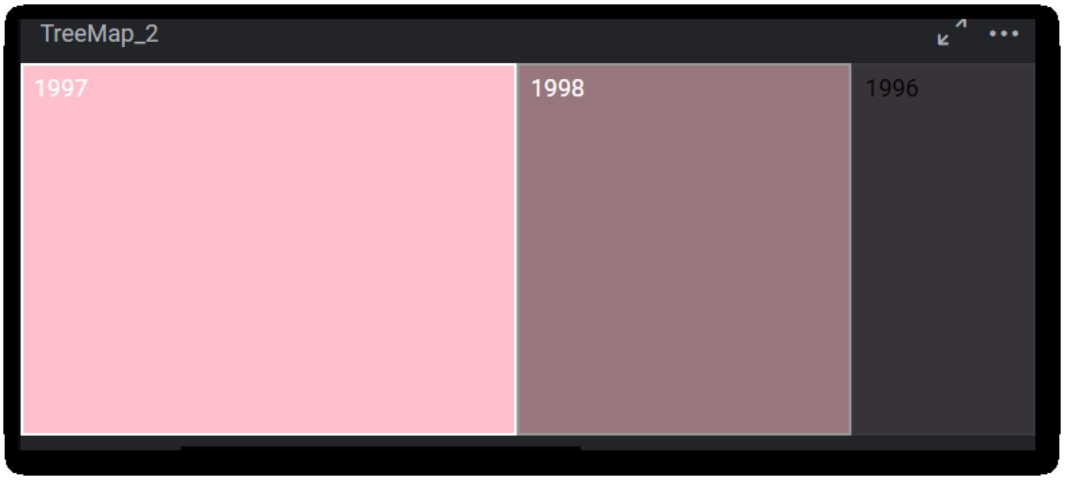
We can customize the 'background-color' for desaturated color in TreeMap widget.

For example, we can customize the background color. Below code snippet represent to change the background color.

JS

```
"desaturatecolor":
{
"background-color": "pink"
},
```

The following image illustrates the applied background color for desaturated color.



FontSettings

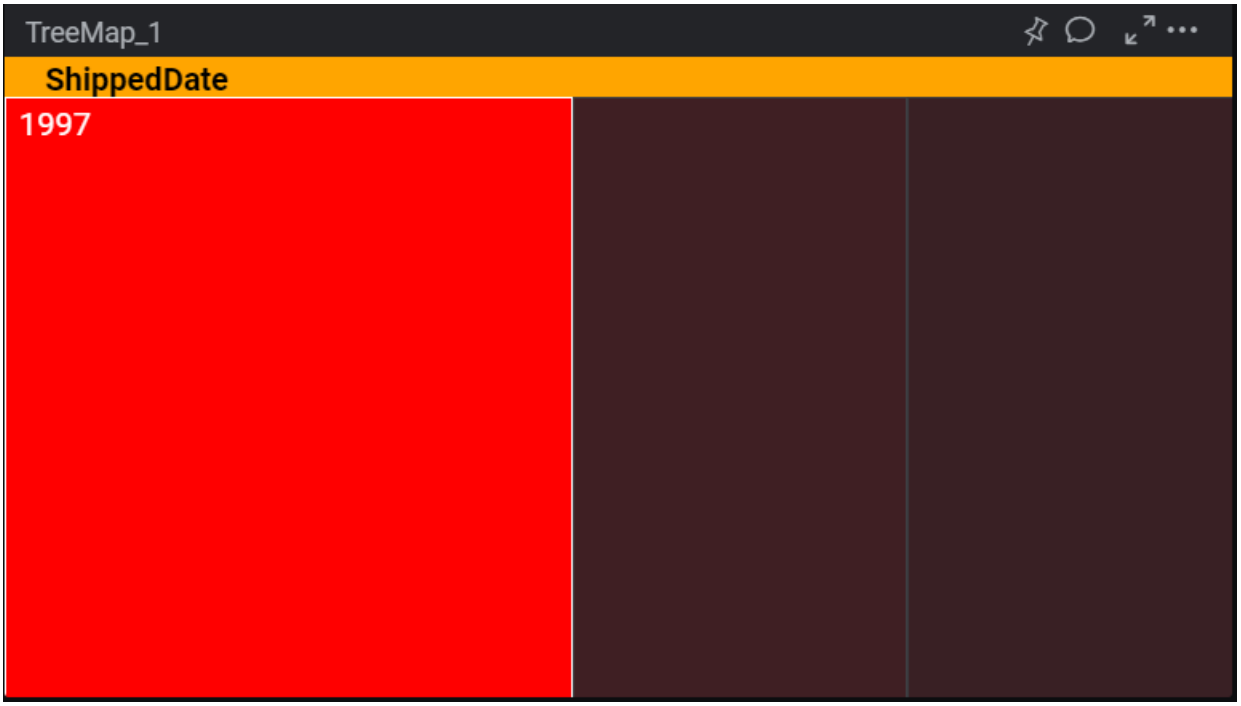
We can customize the 'font-size' property for treemap.

For example, we can customize the font size for drill down header, tooltip, legend and label text. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px",
},
```

The following image illustrates the applied font size for treemap.



RangeNavigator

Sections

We can customize the below sections in RangeNavigator widget.

- Tooltip
- Selected
- Unselected
- Border
- FontSettings

Tooltip

We can customize the below properties in tooltip section.

- Background-color
- font-color

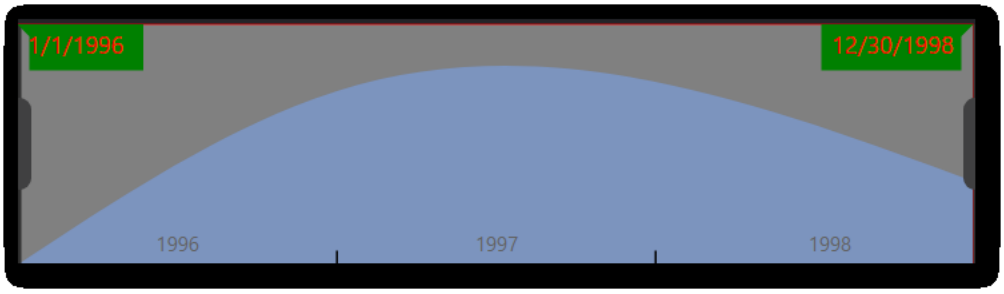
For example, we can customize the background color for tooltip. Below code snippet represent to change the background color.

JS

```
"tooltip":
```

```
{
"background-color": "green",
"font-color": "red"
},
```

The following image illustrates the applied background color for tooltip.



Selected

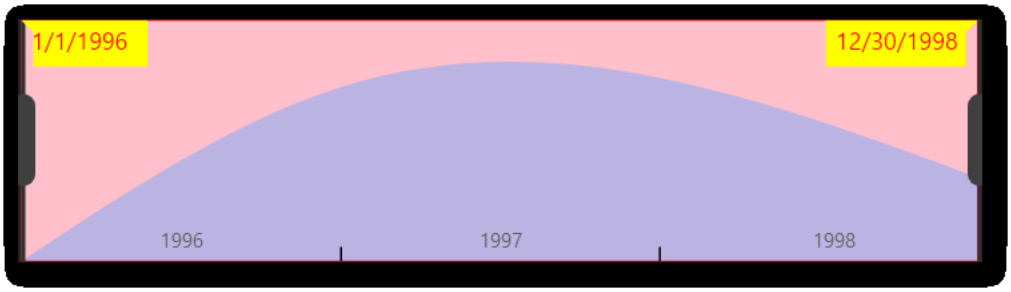
We can customize the 'background-color' for selected portion in RangeNavigator.

For example, we can customize the background color. Below code snippet represent to change the background color.

JS

```
"selected":
{
"background-color": "pink",
},
```

The following image illustrates the applied background color for selected region.



Unselected

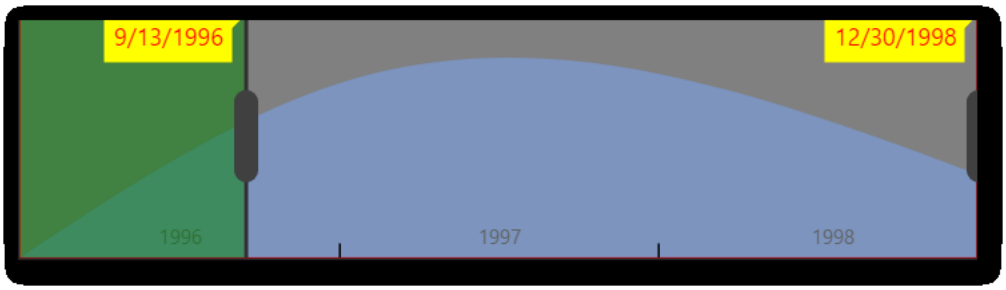
We can customize the background-color for unselected region in RangeNavigator.

For example, we can customize the background color. Below code snippet represent to change the background color.

JS

```
"unselected":
{
"background-color": "green",
},
```

The following image illustrates the applied background color for unselected region.



Border

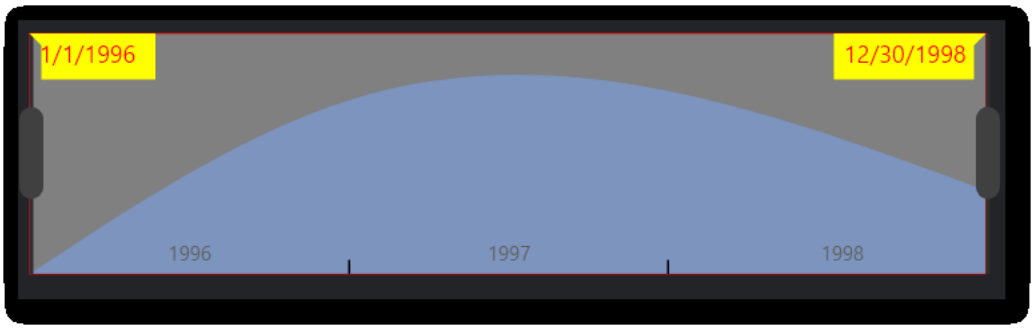
We can customize the border-color for RangeNavigator.

For example, we can customize the border color. Below code snippet represent to change the border color.

JS

```
"border":
{
"border-color": "red",
},
```

The following image illustrates the applied border color for border.



FontSettings

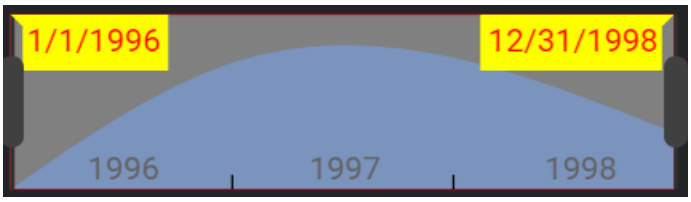
We can customize the 'font-size' property in range navigator.

For example, we can customize the font size for tooltip, range text. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```

The following image illustrates the applied font size for range navigator.



Gauge

Sections

We can customize the below sections in gauge widget.

- Range1
- Range2
- Data
- FontSettings

Range1

We can customize the below properties in range1 section.

- background-color
- border-color

For example, we can customize the background color. Below code snippet represent to change the background color.

JS

```
"range1":
{
"background-color": "green",
"border-color": "red"
},
```

The following image illustrates the applied background color for range1.

Range2

We can customize the below properties in range2 section.

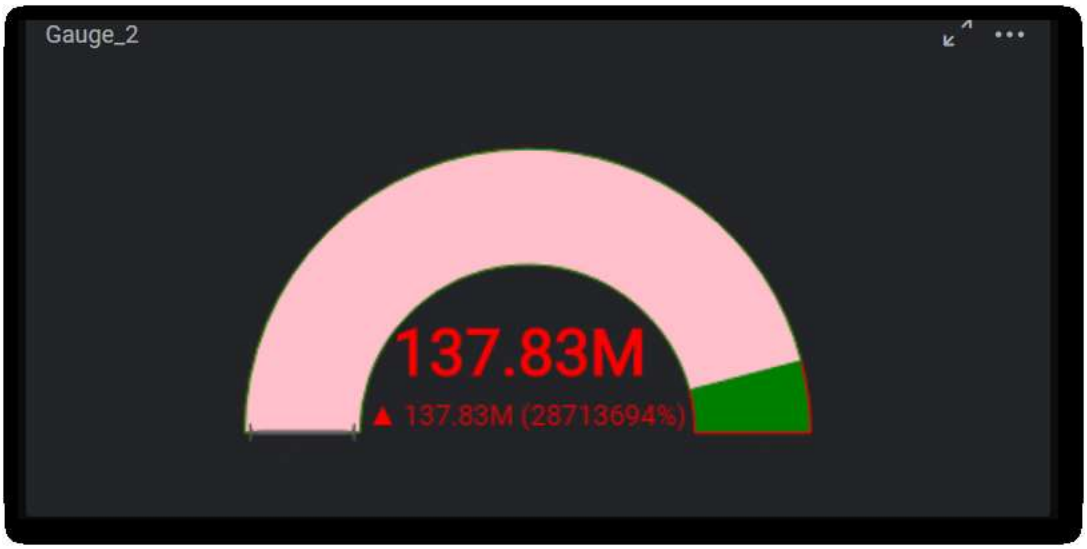
- Background-color
- border-color

For example, we can customize the border color. Below code snippet represent to change the border color.

JS

```
"range2":
{
"background-color": "yellow",
"border-color": "blue"
},
```

The following image illustrates the applied background color for range2.



Data

We can customize the font-color in data section.

Below code snippet represent to change the font color.

JS

```
"data":
{
"font-color": "green"
},
```

The following image illustrates the applied font color for data.

FontSettings

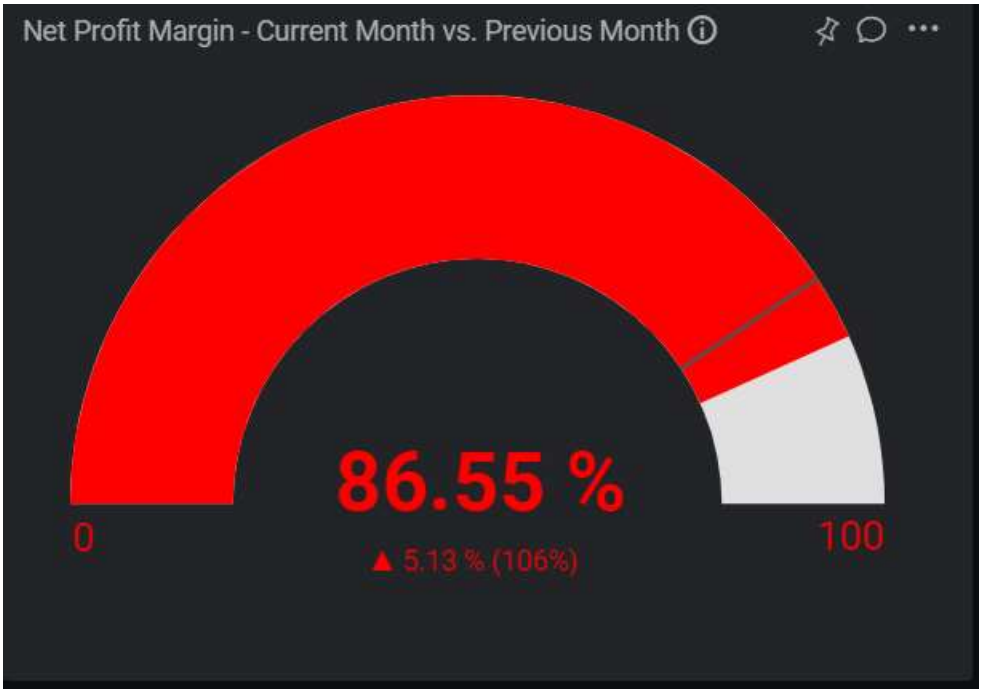
We can customize the 'font-size' property in gauge.

For example, we can customize the font size for data section, tooltip and range value. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```

The following image illustrates the applied font size for gauge.



Chart

Sections

We can customize the below sections in chart widget.

- X-axis
- Y-axis
- Axes
- Series
- Tick
- Legend
- FontSettings

X-axis

We can customize the below properties in x-axis section

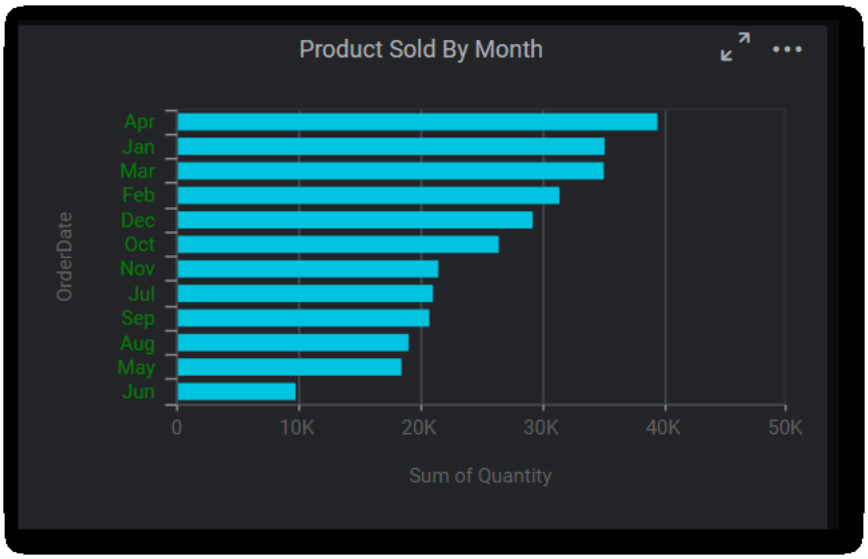
- xaxisline-color
- primaryxaxis-majorgridlines
- primaryxaxis-fontcolor
- primaryxaxis-titlecolor
- primaryxaxisfont-size

For example, we can customize the primary x-axis font color and font size for x-axis. Below code snippet represent to change the font color.

JS

```
"x-axis":
{
"xaxisline-color": "#484c54",
"primaryxaxis-majorgridlines": "yellow",
"primaryxaxis-fontcolor": "green",
"primaryxaxis-titlecolor": "#636363",
"primaryxaxisfont-size": "14px"
},
```

The following image illustrates the primary x-axis font color.



Y-axis

we can customize the below properties in y-axis section.

- yaxisline – color
- primaryyaxis-majorgridlines
- primaryyaxis -fontcolor
- primaryyaxis- titlecolor
- primartyyaxixfont-size

For example, we can customize the primary y-axis font color and font-size for y-axis. Below code snippet represent to change the font color, font size.

JS

```
"y-axis":
{
"yaxisline-color": "#484C54",
"primaryyaxis-majorgridlines": "#484C54",
"primaryyaxis-fontcolor": "blue",
"primaryyaxis-titlecolor": "#636363",
"primartyyaxixfont-size": "14px"
},
```

The following image illustrates the primary y-axis font color.

Axes

we can customize the axis line color for secondary axis.

Below code snippet represent to change the axis line color.

JS

```
"axes":
{
"axisline-color": "red"
},
```

The following image illustrates applied axis line color.

Series

We can customize the below properties in series section.

- Background-color
- font-color

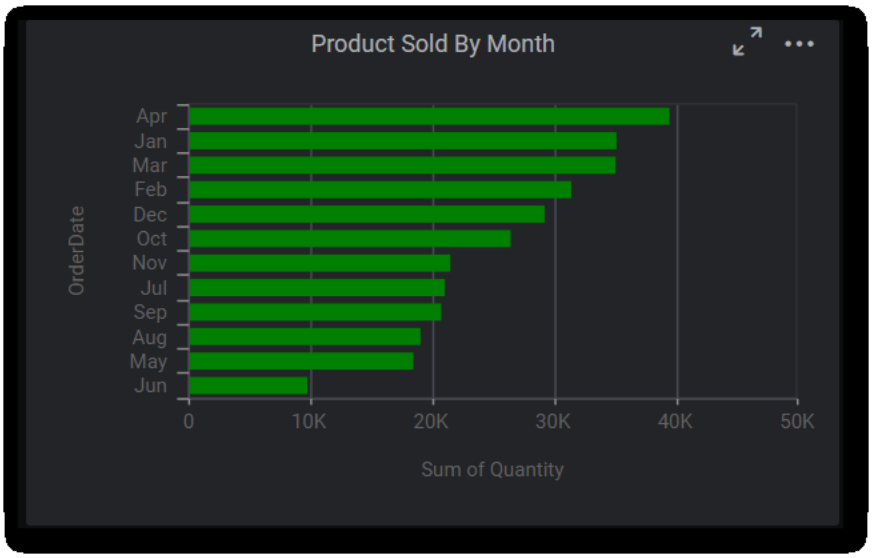
- font-size

For example, we can customize the background color, font color and font size. Below code snippet represent to change the series color and font size.

JS

```
"series":
{
"background-color": ["green"],
"font-color": "red",
"font-size":"15px"
},
```

The following image illustrates the applied series font-color and font-size.



Tick

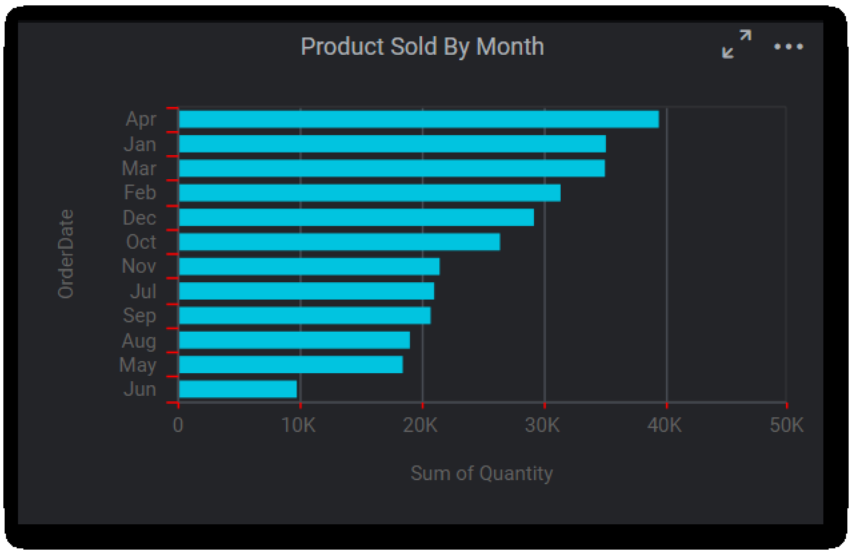
We can customize the major tick color in tick section.

Below code snippet represent to change the tick color.

JS

```
"tick":
{
"majortick-color": "red",
},
```

The following image illustrates the applied tick color.



Legend

We can customize the legend font color property in legend section.

For example, we can customize the legend font color. Below code snippet represent to change the legend font color.

JS

```
"legend":
{
"legendfont-color": "red"
},
```

The following image illustrates the applied legend font color.

FontSettings

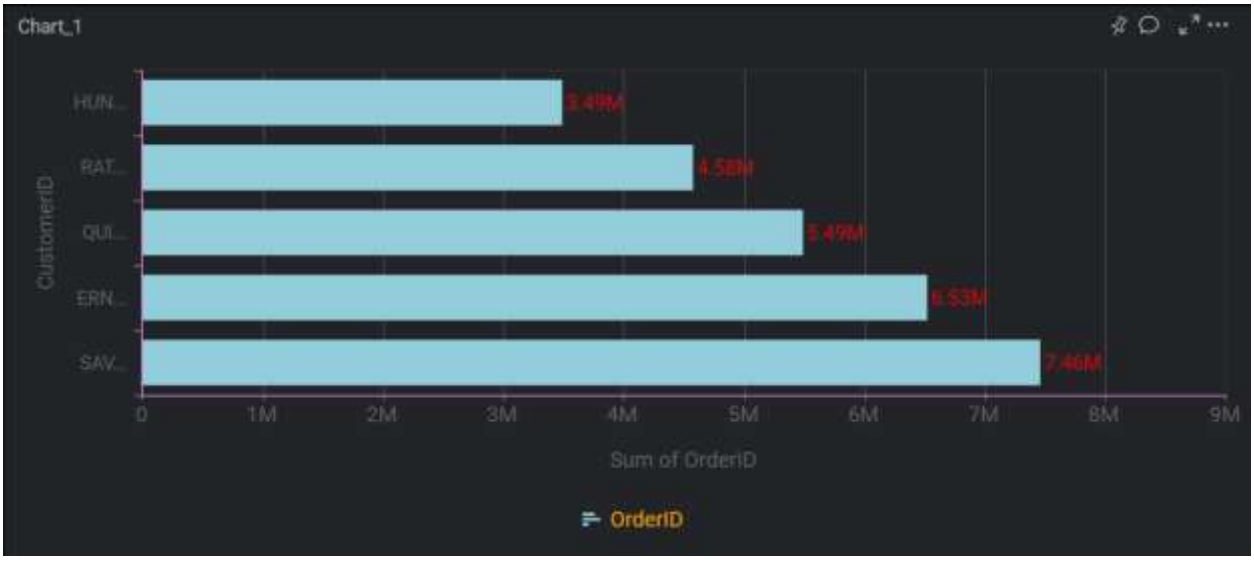
We can customize the 'font-size' property for chart.

For example, we can customize the font size for x-axis label, y-axis label, tooltip, legend, data label. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```

The following image illustrates the font size for chart.



ComboChart

Sections

We can customize the below sections in ComboChart widget.

- X-axis
- y-axis
- Series
- Points
- Legend
- FontSettings

X-axis

We can customize the below properties in x-axis section

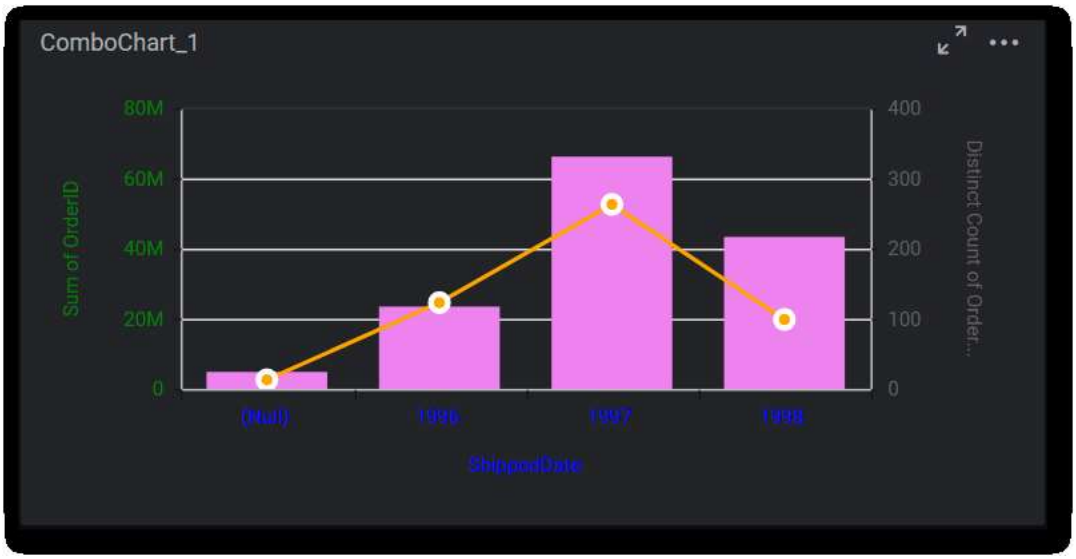
- font-color
- title-color
- primaryxaxisfont-size

For example, we can customize the font color and font size. Below code snippet represent to change the font color and font size.

JS

```
"x-axis":
{
"font-color": "blue",
"title-color": "blue",
"primaryxaxisfont-size": "14px"
},
```

The following image illustrates the applied x-axis title color, font-color and font size.



Y-axis

We can customize the below properties in y-axis section

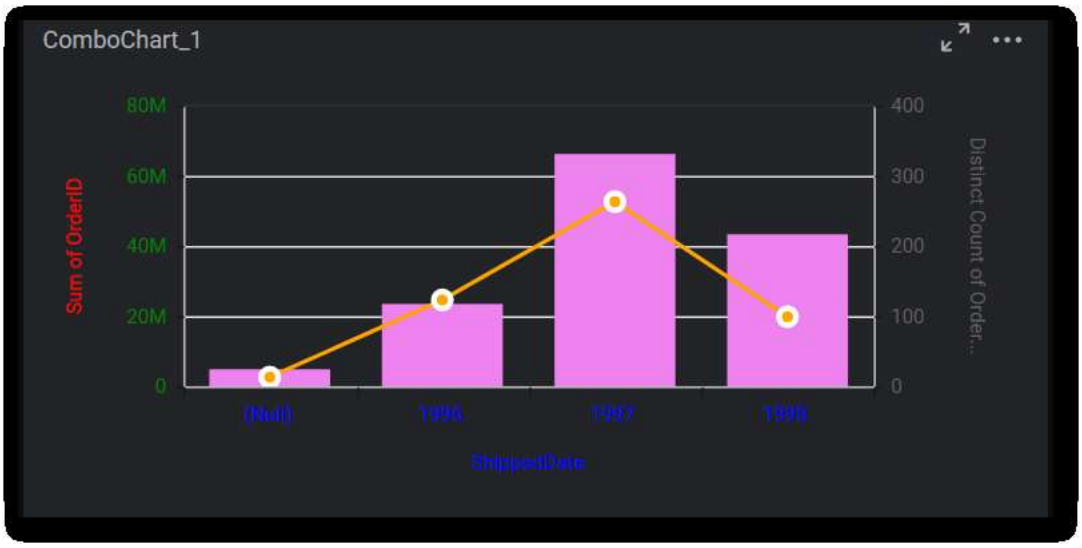
- font-color
- title-color
- primartyyaxixfont-size

For example, we can customize the title color, font color and font size. Below code snippet represent to change the title color, font color and font size.

JS

```
"y-axis":
{
"font-color": "green",
"title-color": "red",
"primartyyaxixfont-size": "15px"
},
```

The following image illustrates the applied y-axis title color and font size.



Series

We can customize the below properties in series section

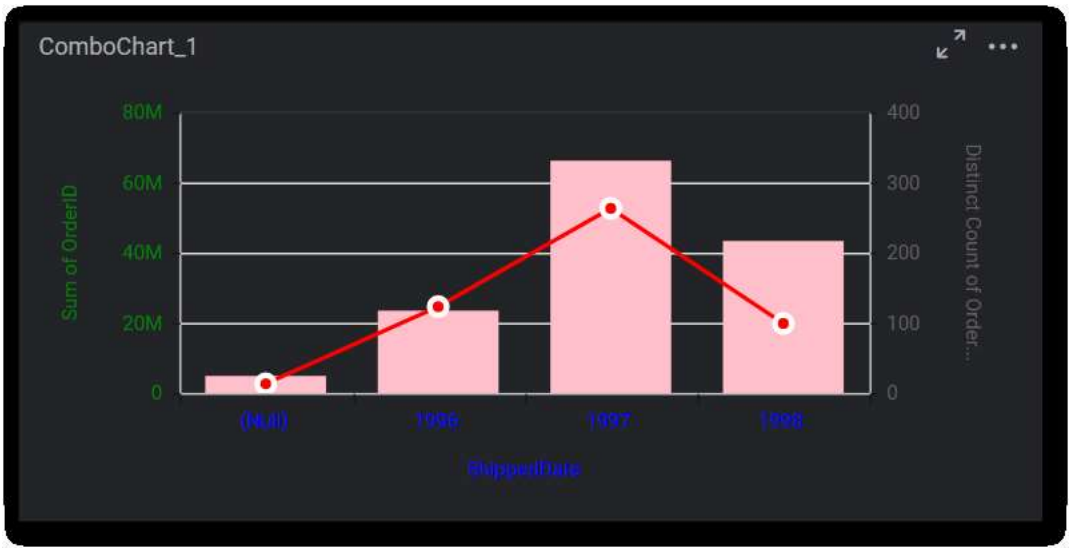
- background-color
- font-color
- font-size

For example, we can customize the background color and font size for series and line. Below code snippet represent to change the background color and font size. First one is bar color and second one is line color.

JS

```
"series":
{
"background-color": ["pink", "red"],
"font-color": "yellow",
"font-size": "15px"
},
```

The following image illustrates the applied series background color and font size.



Legend

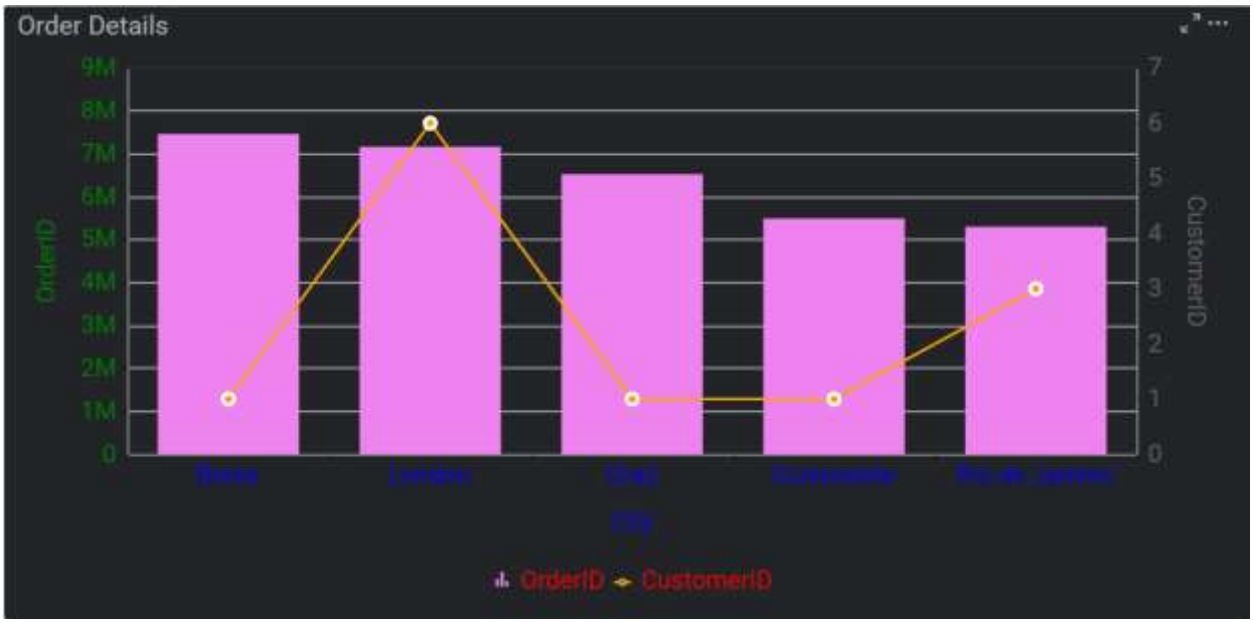
We can customize the legend font color property in legend section.

For example, we can customize the legend font color. Below code snippet represent to change the legend font color.

JS

```
"legend":
{
"legendfont-color": "red"
},
```

The following image illustrates the applied legend font color.



FontSettings

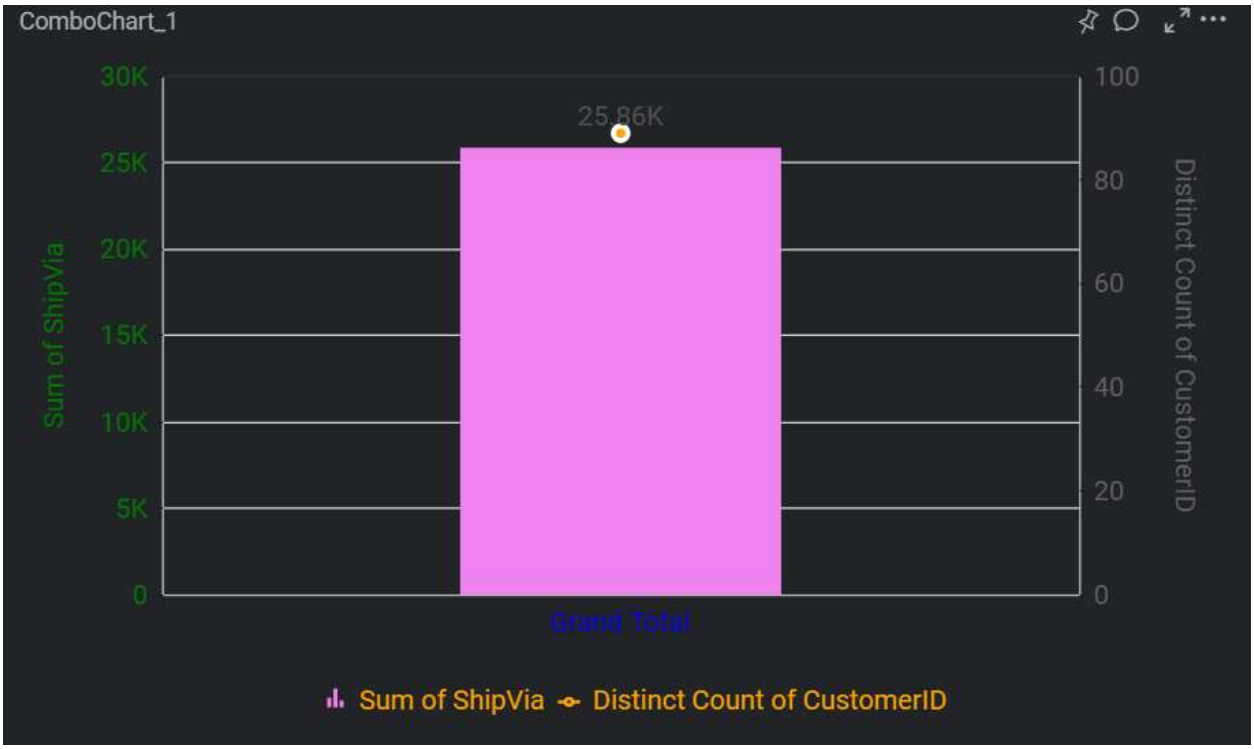
We can customize the 'font-size' property for chart.

For example, we can customize the font size for x-axis label, y-axis label, tooltip, legend, data label. Below code snippet represent to change the font size.

JS

```
"fontSettings":
{
"font-size": "16px"
},
```

The following image illustrates the font size for chart.



Label

Sections

We can customize the below sections in Label widget.

- Content

Content

We can customize the below properties in label text section

- font-color
- font-size

For example, we can customize the font color and font size in label. Below code snippet represent to change the font color and font size.

JS

```
"label":
{
"font-color": "red",
"font-size": "16px",
},
```

The following image illustrates the applied font color and font size for label widget.

Note: We did not provide API for height and width customization due to consideration on view of the layout.

- By using the API “beforeWidgetRender” we can apply customized theme to the widgets.

JS

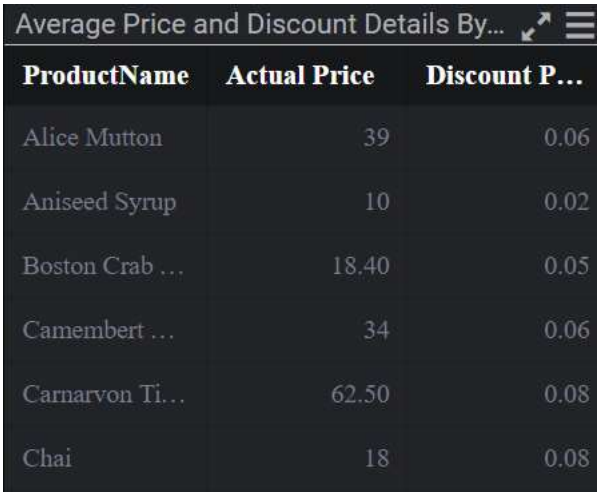
```
$('#dashboard').ejDashboardViewer(
{
serviceUrl: '@ViewBag.ServiceUrl.ToString()',
dashboardPath: '@ViewBag.FirstDashboardPath',
filterParameters: location.search.substr(1),
enableCustomTheming: true,
customThemeSettings: themes,
beforeWidgetRender: ApplyTheme
});
```

- We can apply specific theme which we defined in “widgetThemeSettings” to a widget using the following code snippet.

JS

```
function ApplyTheme(e) {
if (e != null && e.data != null) {
if (e.data.controlType == "Grid") {
this.applyCustomTheme(e.data, "Theme1");
}
}
};
```

- The following image represents “Theme1” settings is applied to Grid widget in dashboard.



How to apply different themes to same widget type

- We can create multiple theme settings for a widget likewise “Theme1” generate under “widgetThemeSettings”.
- Different themes can be applied to same type of widgets by using the following code snippet.

JS

```
function ApplyTheme(e) {
if (e != null && e.data != null) {
if (e.data.controlType == "Grid" && e.data.reportName == "Grid_1") {
this.applyCustomTheme(e.data, "Theme1");
}
if (e.data.controlType == "Grid" && e.data.reportName == "Grid_2") {
this.applyCustomTheme(e.data, "Theme2");
}
}
};
```

- The following image represents “Theme1” and “Theme2” settings is applied to Grid widget. Grid showcased with different font style in the dashboard.

Localization

Localization is the process of translating an application's user interface based on specific cultures.

Localizing Dashboard Viewer

Syncfusion Dashboard Viewer is released with localization support.

The default language is English "en-US".

Read the below documentation on how to add new localizations and how to edit existing localizations in the Syncfusion Dashboard Viewer.

How to add new localization

How to edit existing localizations

In the Dashboard Viewer widget definition added to an application, introduce the localeSettings object with inner objects resourcePath and culture set with respective values like shown below.

Example

JS

```
$('#dashboard').ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath: 'Path of the dashboard',
localeSettings: {
resourcePath: "..//path//", //folder path of the Resources.<Culture
Code>.xml file
culture: "en-US"   //culture of the customized language
},
});
```

If the respective translation file is present in the localization folder of the Dashboard Service, then thelocaleSettings object with inner object culture is enough.

Place the translation file in the below location:

Dashboard Platform SDK :

%localappdata%\Syncfusion\Dashboard Platform SDK\Service\Localization

Dashboard Designer :

C:\ProgramData\Syncfusion\DashboardDesigner\\{{Version}}\IISExpress_DashboardService\Localization

Dashboard Server :

C:\Syncfusion\Dashboard Server\DashboardServer.Web\DashboardService\Localization

Example

JS

```
$('#dashboard').ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath: 'Path of the dashboard',
localeSettings: {
culture: "pt-BR"  //culture for Portuguese (Brazil). Ensure that "pt-BR"
translation file is already present in the
{RootFolderOfDashboardService}\Localization folder.
},
});
```

Note: If any of the values is missing in the XML file, the default culture value will be loaded for that particular name. If the target culture itself is missing in the XML file, the default culture will be loaded.

How to

Bind Dashboard or Widget to the Dashboard Viewer in a HTML page

HTML page creation

Create a new text document and save it as .html extension.(example: Filename.html)

Adding required files in HTML page

Note: Follow the link to install Dashboard Platform SDK.

After installing Dashboard Platform SDK setup, scripts, themes, font folders will be found in the below location.

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Html

Include these scripts, themes, font folders in your application and refer the files of each folder in the HTML page as mentioned below:

ASPX-CS

```
<head>
<script src="scripts/jquery-1.10.2.min.js"></script>
<script src="scripts/jquery.easing.1.3.min.js"></script>
<script src="scripts/ej.dashboardViewer.all.min.js"></script>
<link href="themes/ej.widgets.core.min.css" rel="stylesheet" />
<link href="themes/default-theme/ej.theme.min.css" rel="stylesheet" />
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet" />
</head>
```


Hosting Dashboard Service

After installing the Dashboard Platform SDK setup you will find Service and DashboardServiceInstaller folder in the following location and these folders are required to start the Dashboard Service.

`%localappdata%\Syncfusion\Dashboard\Samples\Common`

To host the Dashboard Service in IIS, Please run the `SyncfusionDashboardServiceInstaller-IIS.exe` from the following location and enter the port number in which dashboard service to be hosted.

`%localappdata%\Syncfusion\Dashboard Platform SDK\DashboardServiceInstaller`

For Example:

If you are hosting Dashboard Service at port #3000 in localhost, then Dashboard Service URL will be framed as below:

`http://localhost:3000/DashboardService.svc`

Open this URL in any of the browser and make sure Dashboard Service is running.

Configuring Dashboard/Widget to Dashboard Viewer

Add the hosted Dashboard Service URL and the Dashboard/Dashboard Widget file (*.SYDX/.SYDW) location to the Dashboard Viewer like below.

For Dashboard:

ASPX-CS

```
<div id="dashboard" style="width: 100%; height: 100%;" ></div>
<script type="text/javascript">
$(document).ready(function () {
$('#dashboard').ejDashboardViewer({
serviceUrl: "http://localhost:3000/DashboardService.svc",
dashboardPath:
"D:\\Syncfusion\\Dashboard\\Dashboards\\WorldWideCarSalesDashboard.sydx",
});
});
```



```
</script>
```

For Dashboard Widget:

ASPX-CS

```
<div id="dashboard_widget" style="width: 100%; height: 100%;" ></div>
<script type="text/javascript">
$(document).ready(function () {
$('#dashboard_widget').ejDashboardViewer({
serviceUrl: "http://localhost:3000/DashboardService.svc",
dashboardPath:
"D:\\Syncfusion\\Dashboard\\Widget\\WorldWideCarSalesWidget.sydw",
});
});
</script>
```

Note: The Dashboard/Dashboard Widget file path should be accessible from the provided Dashboard Service path (or link).

To initiate the dashboard service instance you can follow anyone of the below methods

1. Hosting Dashboard Service in IIS

2. Hosting Dashboard Service in IIS Express

3. Hosting Dashboard Service as Windows Service Background Process

Finally you get the complete Dashboard Viewer Sample structure like below.

ASPX-CS

```
<html>
<style>
body, html {
height: 100%;
overflow: auto !important;}
</style>
<head>
<script src="scripts/jquery-1.10.2.min.js"></script>
<script src="scripts/jquery.easing.1.3.min.js"></script>
<script src="scripts/ej.dashboardViewer.all.min.js"></script>
<link href="themes/ej.widgets.core.min.css" rel="stylesheet" />
<link href="themes/default-theme/ej.theme.min.css" rel="stylesheet" />
<link href="themes/default-theme/ej.dashboardViewer.all.min.css"
rel="stylesheet" />
</head>
<body>
<div id="dashboard" style="width: 100%; height: 100%;" ></div>
<script type="text/javascript">
```



```
$(document).ready(function () {
$('#dashboard').ejDashboardViewer({
serviceUrl: "http://localhost:3000/DashboardService.svc",
dashboardPath:
"D:\\Syncfusion\\Dashboard\\Dashboards\\WorldWideCarSalesDashboard.sydx", //
Replace the file path for Dashboard Widget in Dashboard Viewer like,
D:\\Syncfusion\\Dashboard\\Widgets\\WorldWideCarSalesWidget.sydw
});
});
</script>
</body>
</html>
```

Information: Hosting dashboard service at IIS is recommended for the production environment for object management and other memory management features.

Apply the dashboard parameters using dashboardParameterSettings API

The Dashboard Platform SDK provides support to customize the dashboard parameter settings using following APIs.

- dashboardParameterSettings,
- openParameterDialog()

1.dashboardParameterSettings

This API is used to show or hide the dashboard parameter icon and applying dashboard parameters.

- showIcon
- data

showIcon:

Holds the Boolean value and it is used to show or hide the parameter icon in Dashboard Viewer.

data:

Holds the array of value as below table.

API	Description
parameterName	Name of the parameter
showInPromptDialog	Indicates whether to show the mentioned parameter in prompt dialog, which will appear before applying the value.
showInParameterDialog	Indicates whether to show the mentioned parameter in parameter dialog with its default value.
value	Value of the parameter

Below code snippet is example for passing the value of the parameter:

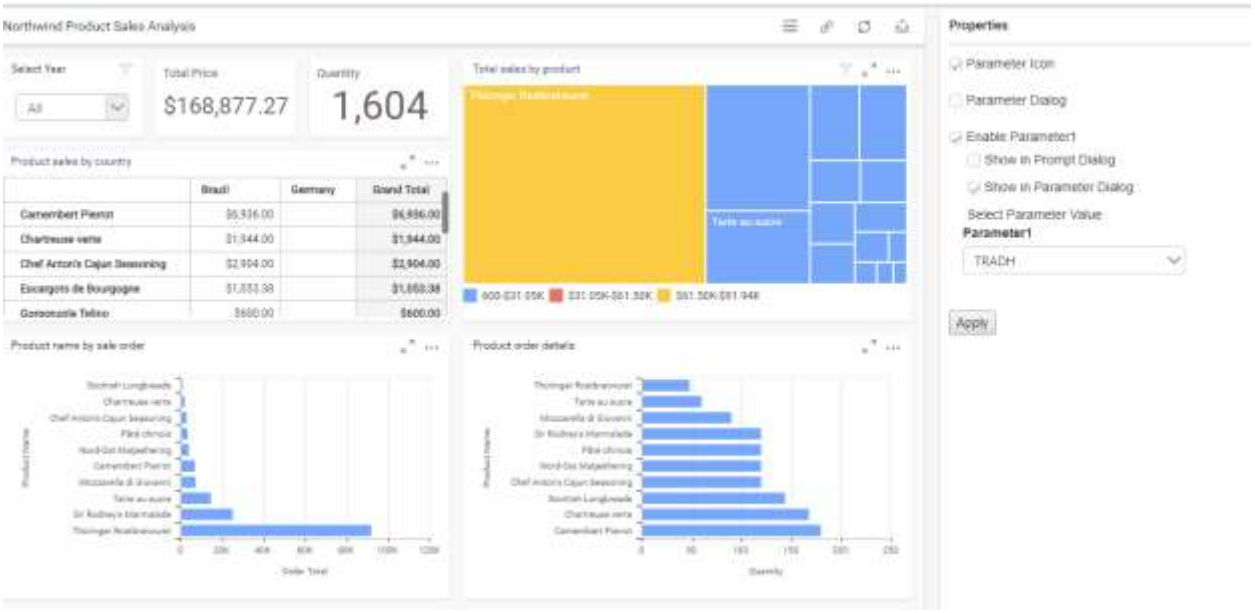
JS

```
$("#dashboard").ejDashboardViewer(
```



```
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
dashboardParameterSettings: {
"showIcon":true, // to show or hide the dashboard  parameter icon
"data":[{
"parameterName":"CustomerId",
"showInParameterDialog":true,
"showInPromptDialog":false,
"value":"TRADH",
}]
},
}
);
```

The below image illustrates the above code snippet,



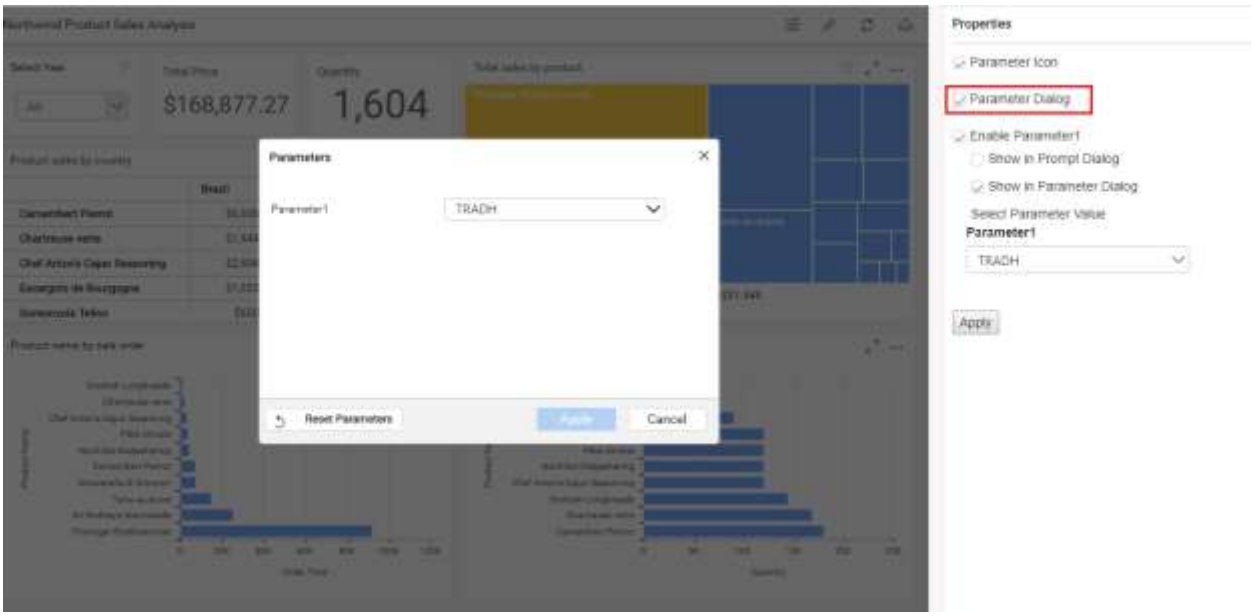
2.openParameterDialog

This method is used to open the dashboard parameter dialog explicitly with dashboard viewer instance.

JS

```
<script type="text/JavaScript">
var dashboardObject;
$(document).ready(function () {
dashboardObject = $('#dashboard').data("ejDashboardViewer");
dashboardObject.openParameterDialog();
});
</script>
```

The below image illustrates the above code snippets,



Apply the auto refresh using autoRefreshSettings API

The Dashboard Platform SDK provides support to customize the auto refresh settings by using the following APIs:

- autoRefreshSettings

autoRefreshSettings

This API is used to apply auto refresh interval and schedule to the whole dashboard or selected widgets in the dashboard.

- tabName
- tabIndex
- isWidgetSpecific
- showWaitingIndicator,
- trackData
- widgets
- intervalSettings

tabName

Holds the String value, and it is used to refresh the data in the given dashboard name of the dashboard viewer.

tabIndex

Holds the Integer value, and it is used to refresh the dashboard data in the given dashboard tab index value of the dashboard viewer.

isWidgetSpecific

Holds the Boolean value. If you want to refresh each widget in the dashboard, you should enable this API. It is used to refresh the collection of widgets alone in the dashboard by denoting the id attribute and interval settings.

showWaitingIndicator

Holds the Boolean value, and it is used to show waiting indicator while refreshing the data in the dashboard.

trackData

Holds the Boolean value, and it is used to refresh the data in the viewer when there is any modifications of data in the data source table.

widgets

Holds the array of widgets collection list, and it contains id with Guid and interval settings. It is used to refresh the listed widget data alone in the viewer.

intervalSettings

Holds the object of values as shown in the following table:

<table>
<tr><th>API</th><th>Description</th><th>Example</th></tr>
<tr><td>mode</td><td>Name of the scheduling type</td><td>mode: "Hourly" or "Daily" or "Weekly" or "Monthly" or "Yearly"</td></tr>
<tr><td>hourlySchedule</td><td>Holds the hourly scheduling order object value. The dashboard or widget will be refreshed in the specified time interval in number of hours, minutes, and seconds. You can set the time interval value to maximum of hours: 99, minutes: 999 and seconds: 9999
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>hours</td><td>Indicates number of hours to refresh the data</td></tr>
<tr><td>minutes</td><td>Indicates number of minutes to refresh the data</td></tr>
<tr><td>seconds</td><td>Indicates number of seconds to refresh the data</td></tr>
</table></td><td>hourlySchedule:{hours: 0,minutes: 0,seconds: 30}</td></tr>
<tr><td>dailySchedule</td><td>Holds the daily scheduling order object value. Th dashboard or widget will be refreshed in specified number of days. If you specify the number of days as 2 to refresh the dashboard or widget, then the data will be refreshed on 12PM at end of the total days
<table>
<tr><th>Name</th><th>Description</th></tr>
<tr><td>days</td><td>Indicates number of days taken to refresh the data</td></tr>
</table></td><td>dailySchedule: { days: 1}</td></tr>
<tr><td>weeklySchedule</td><td>Holds the weekly scheduling order object value. The dashboard or widget will be refreshed in the given week days in the specified number of weeks
<table>
<tr><th>Name</th><th>Description</th></tr>
</table></td><td>weeklySchedule: {weeks: 1, weekDay: [1]}</td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td>weeks</td><td>Indicates number of weeks taken to refresh the data</td></tr><tr><td>weekDay</td><td>Indicates week days value, and it should be arrranged in this specific order [Sunday-Saturday ---> 1-7] to refresh the data</td></tr></table></td><td></td></tr>
<tr><td>monthlySchedule</td><td>Holds the monthly scheduling order object value. The dashboard or widget will be refreshed in the given day of the specified number of months<table><tr><td>Name</td><td>Description</td></tr><tr><td>day</td><td>Indicates a day in a month to refresh the data. If you need to refresh the dashboard or widgets in the last day of a month, then it indicates the day value as 32</td></tr><tr><td>months</td><td>To set the number of months to refresh the data</td></tr></table></td><td>monthlySchedule: {day: 1, months: 1}</td></tr>
<tr><td>yearlySchedule</td><td>Holds the yearly scheduling order object value. The dashboard or widget will be refreshed in the given day of the month in specified number of years<table><tr><td>Name</td><td>Description</td></tr><tr><td>years</td><td>Indicates number of years taken to refresh the data</td></tr><tr><td>day</td><td>Indicates valid range of available days in the given month to refresh the data</td></tr><tr><td>month</td><td>Indicates month value, it should be arranged in this order [January-December ---> 1-12] to refresh the data</td></tr></table></td><td>yearlySchedule: {years: 1, day: 1, month: 1}</td></tr>
</table>

For more details, refer to API autoRefreshSettings

Example for auto refresh interval settings in the without multi-tabbed dashboard:

The following code snippet illustrates passing the value of auto refresh settings to refresh the without multi-tabbed dashboard:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
tabName:"Worldwide Car Sales Random data",  // set dashboard name
tabIndex:1,                                 // set dashboard tab index
value, 1 indicates the second tab dashboard
isWidgetSpecific:false,
showWaitingIndicator:false,
trackData:true,
widgets:[],
intervalSettings: {
mode: "Hourly",                  // set the needed mode type like Hourly,
Daily, Weekly, Monthly, Yearly
```



```
hourlySchedule: {
hours: 0,
minutes: 0,
seconds: 30
}
}
}
});
```

The following code snippet illustrates passing the values to auto refresh settings for widget specific auto refresh in dashboard widgets:

To get the widget's Guid, refer to getCurrentDashboardDetails

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
tabName:"Worldwide Car Sales Random data",  // set dashboard name
tabIndex:1,                                 // set dashboard tab index
value, 1 indicates the second tab dashboard
isWidgetSpecific:true,                      // for widget wise auto refresh
intervalSettings: {},
showWaitingIndicator:false,
trackData:true,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",   // set the guid of the
dashboard widget
intervalSettings:{
mode:"Hourly",                        // set the needed mode type like
Hourly, Daily, Weekly, Monthly, Yearly
hourlySchedule:{
hours: 0,
minutes: 0,
seconds: 30
}
}
}]
}
});
```

Example for auto refresh interval settings in multi-tabbed dashboard:

The following code snippet illustrates passing the value to auto refresh settings for refreshing the multi-tabbed dashboard:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabName:"Worldwide Car Sales Random data",  // set dashboard name
tabIndex:0,                                 // set dashboard tab index
value, 0 indicates the first tab dashboard
isWidgetSpecific:false,
```



```
showWaitingIndicator:false,
trackData:true,
widgets:[],
intervalSettings: {
mode: "Hourly",
hourlySchedule: {
hours: 0,
minutes: 0,
seconds: 30
}
}
},
{
tabName:Worldwide Car Sales Random data 1, // set dashboard name
tabIndex:1,                               // set dashboard tab index value,
1 indicates the second tab dashboard
isWidgetSpecific:false,
showWaitingIndicator:false,
trackData:true,
widgets:[],
intervalSettings: {
mode: "Daily",
dailySchedule: {
days: 1     // Default value is 1
}
}
},
{
tabName:Worldwide Car Sales Random data 2, // set dashboard name
tabIndex:2,                               // set dashboard tab index value,
2 indicates the third tab dashboard
isWidgetSpecific:false,
showWaitingIndicator:false,
trackData:true,
widgets:[],
intervalSettings: {
mode: "Weekly",
weeklySchedule: {
weeks: 1,    // Default value is 1
weekDay: [1] // Default value is [1]
}
}
},
{
tabName:Worldwide Car Sales Random data 3, // set dashboard name
tabIndex:3,                               // set dashboard tab index value,
3 indicates the fourth tab dashboard
isWidgetSpecific:false,
showWaitingIndicator:false,
trackData:true,
widgets:[],
intervalSettings: {
mode: "Monthly",
monthlySchedule: {
day: 1,     // Default value is 1
months: 1   // Default value is 1
}
```



```
}
},
{
tabName:Worldwide Car Sales Random data 4, // set dashboard name
tabIndex:4,                                // set dashboard tab index value,
4 indicates the fifth tab dashboard
isWidgetSpecific:false,
showWaitingIndicator:false,
trackData:true,
widgets:[],
intervalSettings: {
mode: "Yearly",
yearlySchedule: {
years: 1,    // Default value is 1
day: 1,      // Default value is 1
month: 1     // Default value is 1
}
}
}]
});
```

The following code snippet illustrates passing the values to autoRefreshSettings for widget specific auto refresh in dashboard widgets:

To get the widget's Guid, refer to getCurrentDashboardDetails

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabName:"Worldwide Car Sales Random data",  // set dashboard name
tabIndex:1,                                 // set dashboard tab index
value, 1 indicates the second tab dashboard
isWidgetSpecific:true,                      // for widget wise auto refresh
intervalSettings: {},
showWaitingIndicator:false,
trackData:true,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",                    // set the
guid of the dashboard widget
intervalSettings:{
mode:"Hourly",
hourlySchedule: {
hours: 0,
minutes: 0,
seconds: 30
}
}
},
{
id:"232d0fa9-0eb1-4b43-8626-d4b8e0f5e625",                   // set the
guid of the dashboard widget
intervalSettings:{
mode:"Daily",
dailySchedule: {
```

```
days: 1
}
}
}]
},
{
tabName:"Northwind Analysis Dashboard",  // set dashboard name
tabIndex:1,                              // set dashboard tab index value, 1
indicates the second tab dashboard
isWidgetSpecific:true,                   // for widget wise auto refresh
intervalSettings: {},
showWaitingIndicator:false,
trackData:true,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",                    // set the
guid of the dashboard widget
intervalSettings:{
mode:"Weekly",
weeklySchedule: {
weeks: 1,
weekDay: [1]
}
}
},
{
id:"444d0fa9-0eb1-4b43-8626-d4b8e0f5e625",                   // set the
guid of the dashboard widget
intervalSettings:{
mode:"Monthly",
monthlySchedule: {
day: 1,
months: 1
}
}
},
{
id:"887d0fa9-0eb1-4b43-8626-d4b8e0f5e625",                   // set the
guid of the dashboard widget
intervalSettings:{
mode:"Yearly",
yearlySchedule: {
years: 1,
day: 1,
month: 1
}
}
}]
}]
});
```

Apply the widget action using widgetActionSettings API

Widget action settings is used to customize the widget action like master interaction, drilldown, and linking action. The Dashboard Platform SDK provides support to customize the widget action settings by using the following APIs:

- widgetActionSettings

widgetActionSettings

This API is used to apply widget action to the selected widget in the dashboard.

- widgetId
- actionSettings

widgetId

Holds the string value, and it is used to set the guid of the widget.

actionSettings

Holds the array of value as shown in the following table:

API	Description
type	Indicates the mouse actions like ["Click", "DoubleClick", and "RightClick" which acts as a "ContextMenu"] for interaction
actions	Indicates the list of actions such as Filter/Linking/DrillDown. If more than one item is given then instead of triggering the action, set the actionSettings type as the context menu.

For more details, refer to API widgetActionSettings

The following code snippet is example for passing the value to the actionSettings:

To get the Guid, refer to getCurrentDashboardDetails

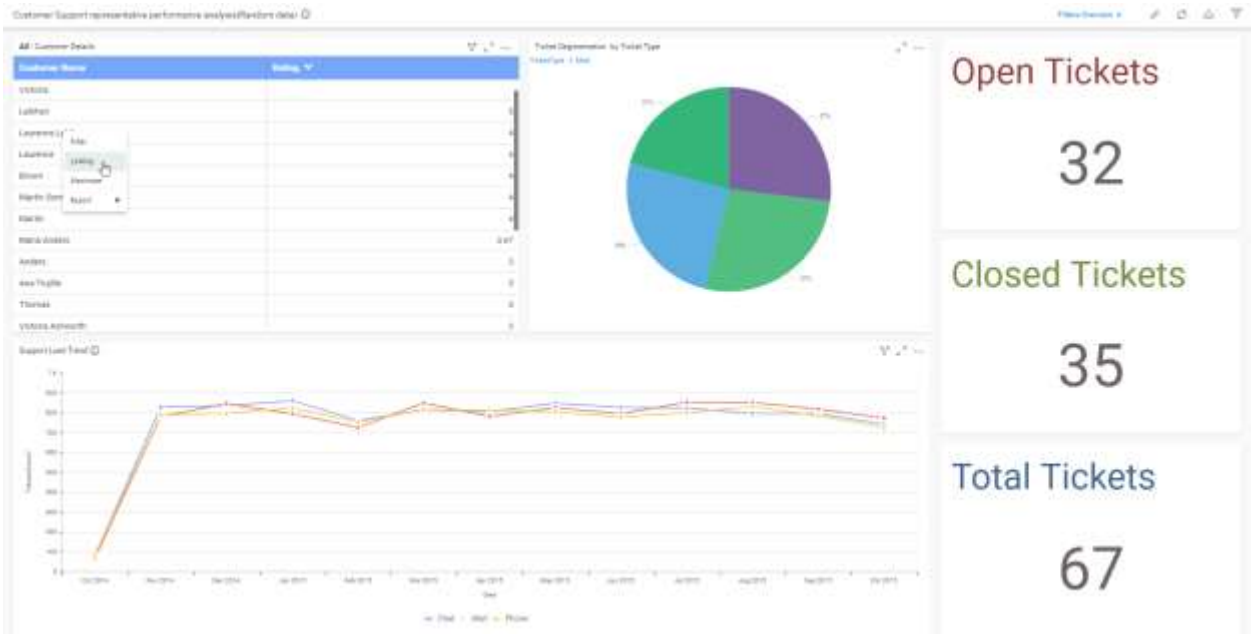
JS

```
$("#container").ejDashboardViewer({
widgetActionSettings:[{
widgetId:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",          // set the guid of
the dashboard widget
actionSettings:[{
type:"Click",
actions:["Filter"]
},
{
type:"ContextMenu",
actions:["Filter, Linking"]
},
{
type:"DoubleClick",
actions:["Linking"]
}]
},
{
widgetId:"897d0fa9-0eb1-4b43-8626-d4b8e0f5e625",         // set the guid of
the dashboard widget
actionSettings:[{
type:"Click",
actions:["Filter"]
},
```



```
{
type:"ContextMenu",
actions:["Filter, Linking, DrillDown"]
},
{
type:"DoubleClick",
actions:["DrillDown"]
}]
}]
});
```

The below image illustrates the above code snippet,



Filtering Views through filterParameters API

Passing Parameter With filterParameters API

You can apply filter to a dashboard using `filterParameters` API. Passing parameters within `filterParameters` API will apply filter in the dashboard on initial load itself.

To set a dashboard parameter with `filterParameters` API, use the following syntax:

`filterParameters: “columnName=filter(value1, value2, value3)”`

where `filter` represents the operator name.

To apply filter to a dashboard, use the following syntax.

`filterParameters: “columnName=value”`

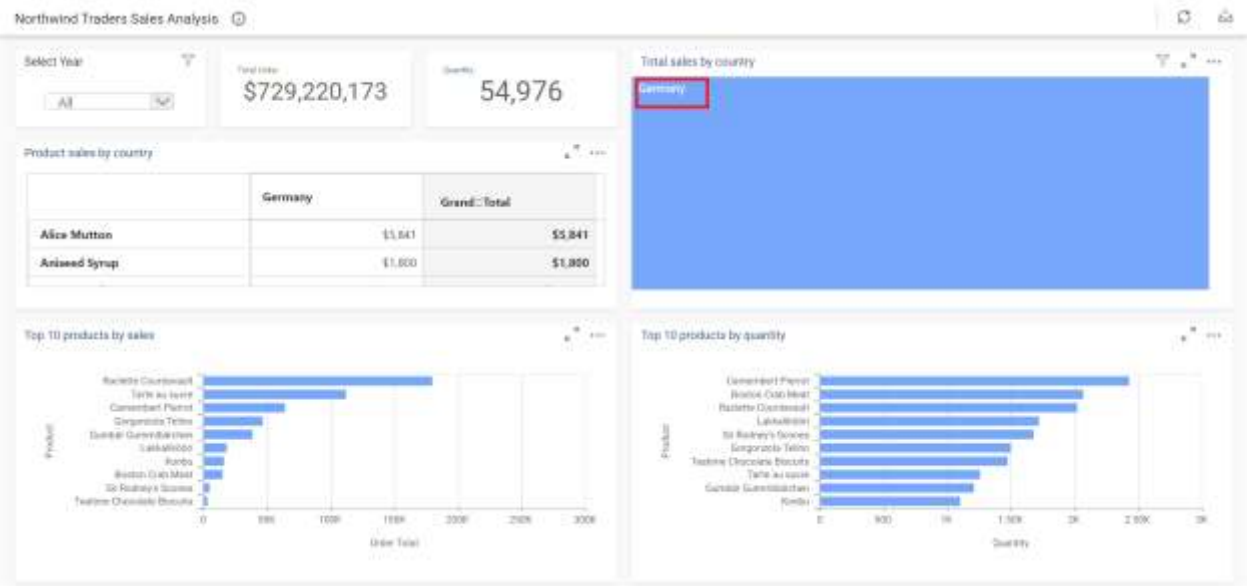
For example, in Sales analysis dashboard that shows the sales trends information for different products. you can show the sales done in Germany alone by passing the Country column filter as Germany in filterParameters API. refer the below code snippet to achieve the same

JS

```
$("#dashboard").ejDashboardViewer(
```

```
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
filterParameters: "Country=Germany",
}
);
```

The below sample illustrates the above applied filter



You can filter the dashboard with multiple values in a column by using the below syntax,

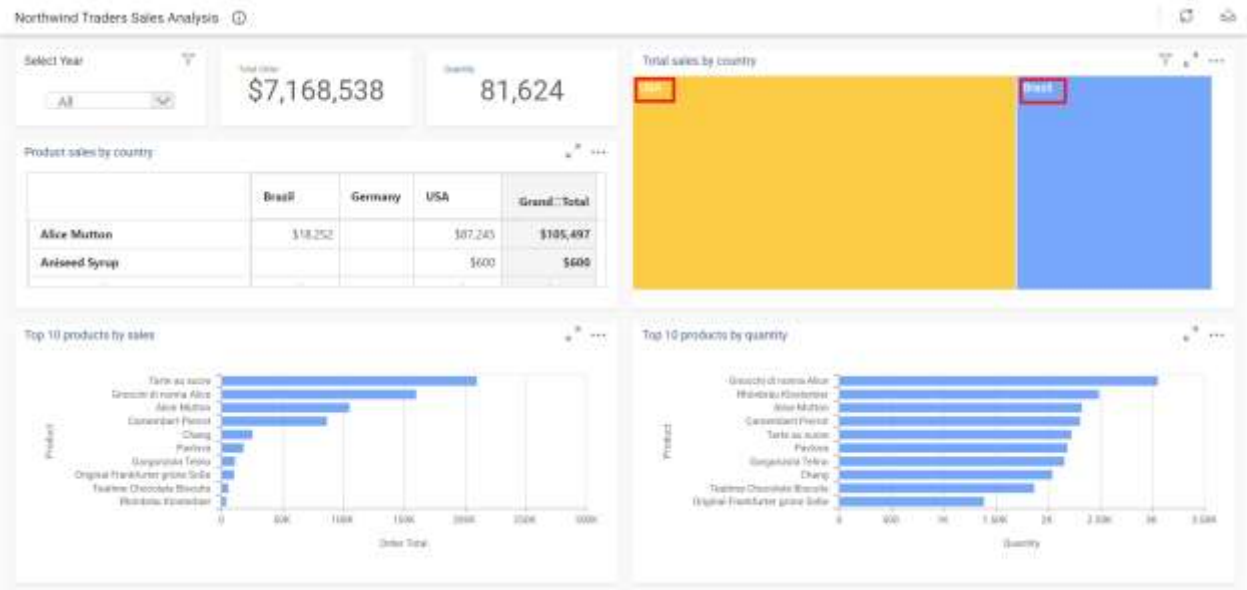
filterParameters: “columnName=value1,value2”

In the below sample code the dashboard will be displaying the data of USA and Brazil alone,

JS

```
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
filterParameters: "Country=USA,Brazil"
}
);
```

The below sample illustrates the above applied filter.



Supported Operators and Date & Time Functions

You can also define parameters with operators to search for one or more values like below.

Operator	Syntax
IN	parameter=IN(value1, value2,..., valueN)
NOTIN	parameter=NOTIN(value1, value2, â€¦, valueN)
BETWEEN	parameter=BETWEEN(value1, value2)
INBETWEEN	parameter=INBETWEEN(value1, value2)
STARTSWITH	parameter=STARTSWITH(value)
ENDSWITH	parameter=ENDSWITH(value)
CONTAINS	parameter=CONTAINS(value1, value2)

You can define parameters (date and time typed columns) with date & time functions applied to filter formatted date values like below.

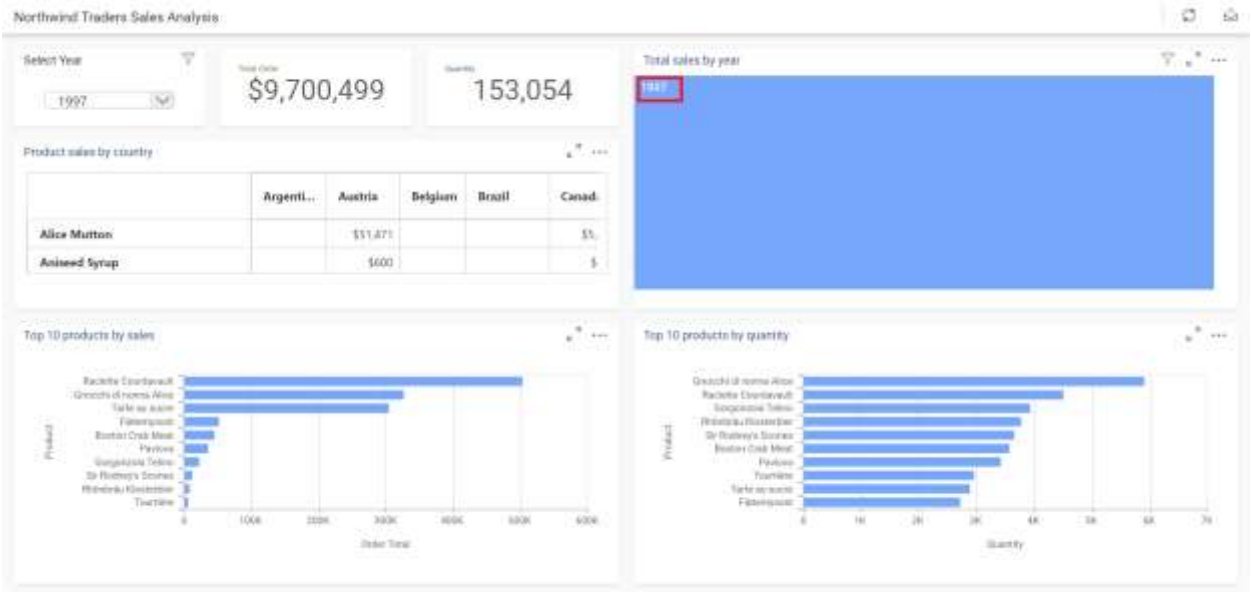
JS

```
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
filterParameters: "Year(OrderDate)=1997"
}
);
```

Function	Syntax

YEAR	YEAR(parameter)=value1, value2, â€¦, valueN
MONTHNAME	MONTHNAME(parameter)=value1, value2, â€¦, valueN
QUARTER	QUARTER(parameter)=value1, value2, â€¦, valueN
QUARTERYEAR	QUARTERYEAR(parameter)=value1, value2, â€¦, valueN
MONTHYEAR	MONTHYEAR(parameter)=value1, value2, â€¦, valueN
DAYMONTHYEAR	DAYMONTHYEAR(parameter)=value1, value2, â€¦, valueN
MONTHDAYYEAR	MONTHDAYYEAR(parameter)=value1, value2, â€¦, valueN
HOURS	HOURS(parameter)=value1, value2, â€¦, valueN
MINUTES	MINUTES(parameter)=value1, value2, â€¦, valueN
DAY	DAY(parameter)=value1, value2, â€¦, valueN
SECONDS	SECONDS(parameter)=value1, value2, â€¦, valueN
DATEHOUR	DATEHOUR(parameter)=value1, value2, â€¦, valueN
DAYOFWEEK	DAYOFWEEK(parameter)=value1, value2, â€¦, valueN
DAYOFYEAR	DAYOFYEAR(parameter)=value1, value2, â€¦, valueN
WEEKOFYEAR	WEEKOFYEAR(parameter)=value1, value2, â€¦, valueN

Here is a dashboard view illustrating the use of parameter with date & time function.



Note: You can also define parameters with operators (except STARTSWITH and ENDSWITH) and date time functions combination.

For example,

filterParameters:"Year(OrderDate)=IN(1996,1998)"

Passing Multiple Parameters With filterParameters API

You can pass more than one parameter within filterParameters API introducing an ampersand (&) symbol in between them to differentiate like below.

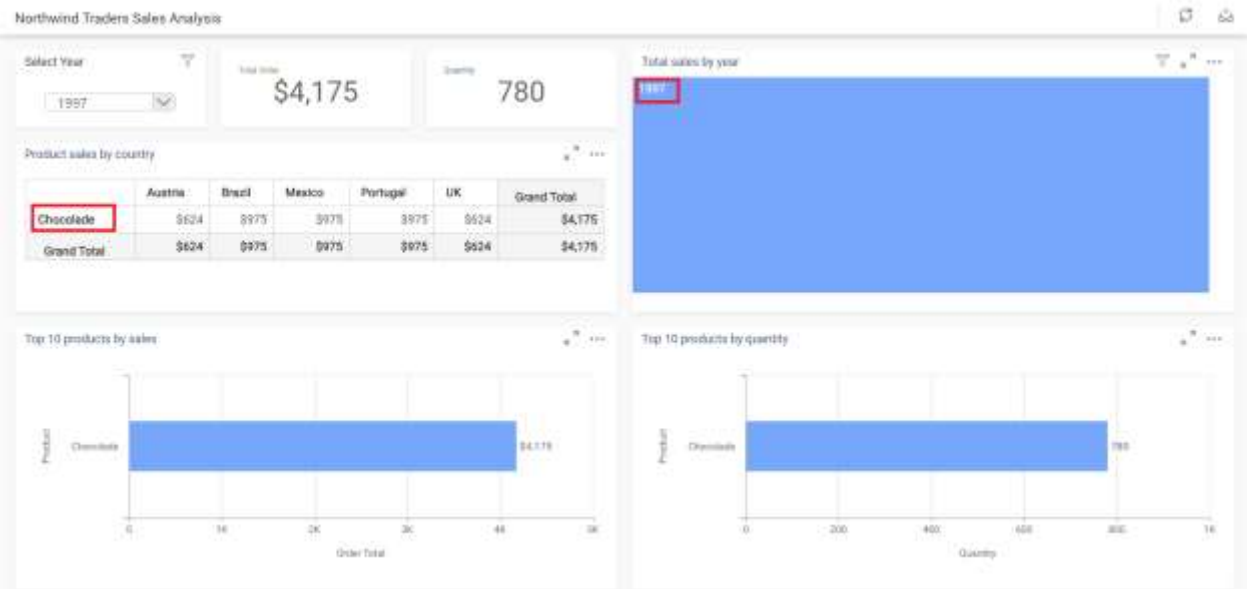
filterParameters: "Year(OrderDate)=1997&ProductName=Chocolate"

To apply filter with multiple parameters to the dashboard, add the following code shown below.

JS

```
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
filterParameters: "Year(OrderDate)=1997&ProductName=Chocolate"
}
);
```

Here is a dashboard view illustrating the same.



Information: The parameter names and values are case-sensitive.

Information:

Information: The operators and date & time function names are case-insensitive.

Information:

Information: Characters like comma (,) and ampersand (&) in value should be prefixed and suffixed with tilde (~) symbol to differentiate itself from syntax elements.For example, CompanyName=Syncfusion Inc~,~

Information:

Information: Invalid parameter name will get ignored from filter consideration.

Information:

Information: Invalid parameter value will result in "No data available to display" in widgets.

How to change data connection string in dashboard at runtime

Note: This walkthrough is based on Getting Started procedures under Dashboard Platform SDK.

Changing connecting string on the fly enables you to switch data connection from one server to another server. This can be achieved through onApplyConnection event API.

Event : onApplyConnection Input type: JSON

We have included a new assembly (Syncfusion.Dashboard.Encryption) for encrypting the connection string. You can get the encryption assembly from the below location

%localappdata%\Syncfusion\Dashboard Platform SDK\Getting Started Samples\Common\Precompiled Assemblies\Syncfusion.Dashboard.Encryption

Refer this assembly in your project as reference to encrypt the connection string.

Also add the NewtonSoft.Json assembly reference in your project to encrypt the connection string. You can download the NewtonSoft.Json package form here

Subscribing to event

JS

```
$('#dashboard').ejDashboardViewer(
{
//Change the connection string in dashboard
onApplyConnection: 'changeConnectionString'
});
```

Handling event

A collection of connection strings can be provided to switch between data servers. Create a model class Connection.cs with the below code.Following code snippets represent the syntax and example.

JSON

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
public class Connection
{
public List<DataSourceConnectionString> DataSources { get; set; }
}
public class DataSourceConnectionString
{
```

```
public string Name { get; set; }
public string ConnectionString { get; set; }
public DataSourceConnectionString()
{
Name = string.Empty;
ConnectionString = string.Empty;
}
}
```

The ConnectionStringBuilder class from Syncfusion.Dashboard.Encryption is used to generate the connection string by providing the respective connection details.

Example

If you created a dashboard with the data source name as 'Northwind Traders Sales Analysis' and Connection type is 'SQL Server' then use the below code snippet to change the data source connection for the respective data source in the dashboard.

JSON

```
{
ConnectionStringBuilder obj = new ConnectionStringBuilder();
ConnectionParameters connectionParameter = new ConnectionParameters
{
AuthenticationType = 0,
ServerName = .,
UserName = **,
Password = **,
Database = "NorthWind"
};
ConnectionProviderType connectionType = ConnectionProviderType.SqlServer; //
Connection type used in the dashboard
OdbcDbmsType odbcType = OdbcDbmsType.None; // ODBC DBMS type, if any
Connection connectionName = new Connection { DataSources = new
List<DataSourceConnectionString>() };
connectionName.DataSources.Add(new DataSourceConnectionString
{
Name = "Northwind Traders Sales Analysis",
ConnectionString = obj.GenerateConnectionString(connectionParameter,
connectionType, odbcType)
});
}
```

The ConnectionParameters should vary based on the connection type used in the dashboard. The following are the parameters required for generating the connecting string with respect to the connection type:

Connection Type	Connection Parameters
SQL Server	AuthenticationType ServerName UserName Password Database

ODBC - MySQL	IsTrustedConnection DriverName UserName Password Database ServerName Database
ODBC - Oracle	DriverName ServiceName PortNumber DriverName ServerName UserName Password
ODBC - Hive	IsSecuredLogin ServerName PortNumber ServerType Database UserName Password
SQLite	FilePath
File Type(Excel, MsAccess, CSV, Text File, XML, JSON)	IsLocal FilePath Password
PostgreSql	ServerName UserName Password Database PortNumber
AzureTableStorage	AccountName AccountKey
SSAS	SourceType AuthenticationType CubePath Database UserName Password
Hive	IsSecuredLogin ServerName PortName ServerType Database

	Username Password
Salesforce	Url UserName Password
Web Data Source	Url UserName Password ClientID ClientSecret OAuthRedirectURL TokenName

Where,

- AuthenticationType holds the value '0' on windows authentication and '1' on Server authentication.
- ServerName points the server where the database exists.
- UserName holds login credential where the server is connected.
- Password holds the secret key to login with the server.
- Database Contains DB name that is used as a data source for the dashboard.
- IsTrustedConnection/IsSecuredLogin holds the Boolean value showing the connection type's security level or visibility.
- DriverName holds the ODBC driver name.
- ServiceName holds the service information for the connection ODBC - Oracle.
- PortNumber contains the port where the server is pointed.
- ServerType holds the information about the type of the server in Hive Connection.
- IsLocal contains the Boolean value to denote whether the file type database is in local or online.
- FilePath holds the location of the file type where the server is connected, either as local path or as online link.
- AccountName is the user credential for the Azure type storage.
- AccountKey holds the secret key to login with the server.
- SourceType has the type of the Cube data for SSAS connection type.
- CubePath holds the path of the DB for the connection SSAS.
- Url has the connection Url for the Salesforce/ Web Data Source connection type.
- ClientID has the client ID for OAuth type on Web data connection.
- ClientSecret has the secret key to login with OAuth type on Web data connection.
- OAuthRedirectURL holds the redirecting URL for OAuth type.
- TokenName holds the name of the Token in OAuth type.

Note: 1. The Name refers to the data source name which is used in Dashboard Designer while designing the dashboard.

2.The changed connection string should be of same connection type and schema like the one you are preferring to change. i.e., if the data source connection type is Microsoft SQL Server, the changed connection string should have the connection type as Microsoft SQL Server and the database should have same schema like the existing one. Missing this, will result in error while previewing the dashboard.

3.If the SYDX contains two or more data sources and you are preferring to change only one of it through specifying the modified connection string like above, the other data sources will assume to represent the existing ones which is used while design the dashboard.

The connection string change can be handled at two different places,

At initial rendering of dashboard After dashboard loading

Change data connection string in dashboard at runtime for ASP.NET/MVC Application

At initial rendering of dashboard

Add the following script in aspx/cshtml file.

JAVASCRIPT

```
<script type="text/javascript">
$('#dashboard').ejDashboardViewer(
{
//Change the connection string in dashboard
onApplyConnection: 'changeConnectionString'
});
// Function calling in script
function changeConnectionString (sender)
{
var changeString = '@ViewBag.EncryptedString';
sender.data.modifiedConnectionStrings = JSON.stringify(changeString);
}
</script>
```

RAZOR

```
<div style ="with:100%;height:100% ">
@(Html.EJDashboard().DashboardViewer("sample_dashboard").
ServiceUrl("https://dashboardsdk.syncfusion.com/DashboardService/DashboardSe
rvice.svc").
DashboardPath("https://dashboardsdk.syncfusion.com/Dashboards/NorthWind
Product Details-Sql.sydx").
FilterParameters("location.search.substring(1)").
ClientSideEvents(event =>
event.DashboardCreated("dashboardCreated").OnApplyConnection("onApplyConnect
ion")))
</div>
<script>
function onApplyConnection(sender) {
var modifiedString = @ViewBag.ModifiedString;
sender.data.modifiedConnectionStrings = JSON.stringify(modifiedString);
}
</script>
```


After dashboard loading

To handle the connection string change after loading the dashboard, you can make use of event (say button click event) as per the below procedure in aspx/cshtml file:

Create a button and register the onclick event with the event caller method.

Add a script holding the changed connection string, event caller and event handler methods.

Following code snippet illustrates the same:

JS

```
<html>
<body>
//Button creation
<div style="padding: 4px 4px 0px 7px;height:4%;background-color: #303030 ">
<input type="button" value="Change Connection String" style="border: none; -
ms-border-radius: 3px; border-radius: 3px;padding: 5px 8px 5px
8px;background-color:  #0080ff; color: #ffffff;font-size: 72%;text-
transform: uppercase;" onclick="changeConnectionString()" />
</div>
<div id="dashboard" style="width:100%; height:94%;"></div>
<script type="text/javascript">
//Button event calling in script
function changeConnectionString() {
var dashboardObj = $('#dashboard').data('ejDashboardViewer');
dashboardObj.model.onApplyConnection = 'onApplyConnection';
dashboardObj.redrawDashboard();
}
function onApplyConnection(sender) {
var changeString = '@ViewBag.EncryptedString';
sender.data.modifiedConnectionStrings = JSON.stringify(changeString);
}
</script>
</body>
</html>
```

Add the following code snippet in DashboardController.cs file.

C#

```
public ActionResult Index() {
DashboardCryptoProvider provider = new
DashboardCryptoProvider(EncryptionTypes.AES);
Connection connectionName = new Connection { DataSources = new
List<DataSourceConnectionString>() };
connectionName.DataSources.Add(new DataSourceConnectionString
{
Name = "Worldwide Car Sales (Random data)",
ConnectionString = "Data Source = .; Integrated Security=True; Initial
Catalog = NORTHWND; MultipleActiveResultSets=True"
});
ViewBag.EncryptedString =
provider.DoEncryption(Converter.SerializeObject(connectionName));
}
```


Change data connection string in dashboard at runtime for Windows Forms/WPF Application

At initial rendering of dashboard

Handle the following procedure in the GetViewer method of DashboardViewer class to change the connection string on load of dashboard.

Include the modified connection string, event caller and event handler methods and frame the HTML string

Return the HTML string to generate a page with it and retrieve its URL path to navigate.

Following code illustrates the same.

C#

```
private string GetViewer(DashboardProperties dashboardProperties)
{
var viewers = "$('#dashboard').ejDashboardViewer({
//Change the connection string in dashboard
onApplyConnection: 'changeConnectionString'
});
var stringBuilder = new StringBuilder();
stringBuilder.Append("<script type=\"text/javascript\"
language=\"javascript\">");
stringBuilder.Append(viewers);
// Function calling in script
stringBuilder.Append("function changeConnectionString (sender)
{sender.data.modifiedConnectionStrings =
JSON.stringify(EncryptedString);}");
stringBuilder.Append("</script>");
return stringBuilder.ToString();
}
```

VB.NET

```
Private Function GetViewer(ByVal dashboardProperties As DashboardProperties)
As String
If dashboardProperties Is Nothing Then Return String.Empty
End If
Dim viewers = "$('#dashboard').ejDashboardViewer({
//Change the connection string in dashboard
onApplyConnection: 'changeConnectionString'
});"
Dim stringBuilder = new StringBuilder()
stringBuilder.Append("<script type=""text/javascript""
language=""javascript"">")
stringBuilder.Append(viewers)
//Function calling in script
stringBuilder.Append("function changeConnectionString(sender)
{sender.data.modifiedConnectionStrings = JSON.stringify(EncryptedString);}")
stringBuilder.Append("</script>")
Return stringBuilder.ToString()
End Function
```

After dashboard loading

Handle the following procedure in the GetViewer method of DashboardViewer class to change the connection string on button click at run time.

Include the modified connection string, event caller and event handler methods and frame the HTML string

Return the HTML string to generate a page with it and retrieve its URL path to navigate.

Following code illustrates the same.

C#

```
private string GetViewer(DashboardProperties dashboardProperties)
{
//Button creation
var stringBuilder = new StringBuilder();
stringBuilder.Append("<body style=\"width:100%; height:100%;\">");
stringBuilder.Append("<div style=\"padding: 4px 4px 0px
7px;height:4%;background-color: #303030 \">"+
"<input type=\"button\" value=\"Change Connection String\" style=\"border:
none; -ms-border-radius: 3px; border-radius: 3px;padding: 5px 8px 5px
8px;background-color:  #0080ff; color: #ffffff;font-size: 72%;text-
transform: uppercase;\" onclick=\"changeConnectionString()\" />"+
"</div>");
stringBuilder.Append("<div id=\"dashboard\" style=\"width:100%;
height:94%;\"></div>");
DashboardCryptoProvider provider = new
DashboardCryptoProvider(EncryptionTypes.AES);
Connection connectionName = new Connection { DataSources = new
List<DataSourceConnectionString>() };
connectionName.DataSources.Add(new DataSourceConnectionString
{
Name = "Worldwide Car Sales (Random data)",
ConnectionString = "Data Source = .; Integrated Security=True; Initial
Catalog = NORTHWND; MultipleActiveResultSets=True"
});
var EncryptedString =
provider.DoEncryption(Converter.SerializeObject(connectionName));
stringBuilder.Append("<script type=""text/javascript""
language=""javascript"">")
//Button event calling in script
stringBuilder.Append("function changeConnectionString() {var dashboardObj =
$('#dashboard').data('ejDashboardViewer');dashboardObj.model.onApplyConnecti
on = 'onApplyConnection';dashboardObj.redrawDashboard();}function
onApplyConnection(sender) {
sender.data.modifiedConnectionStrings = JSON.stringify(EncryptedString);}")
stringBuilder.Append("</script>")
stringBuilder.Append("</body>")
Return stringBuilder.ToString()
}
```

VB.NET

```
Private Function GetViewer(ByVal dashboardProperties As DashboardProperties)
As String
If dashboardProperties Is Nothing Then
Return String.Empty
```



```
//Button creation
Dim stringBuilder = new StringBuilder()
stringBuilder.Append("<body style=""width:100%; height:100%;"">")
stringBuilder.Append("<div style=""padding: 4px 4px 0px
7px;height:4%;background-color: #303030 "">" & "<input type=""button""
value=""Change Connection String"" style=""border: none; -ms-border-radius:
3px; border-radius: 3px;padding: 5px 8px 5px 8px;background-color:
#0080ff; color: #ffffff;font-size: 72%;text-transform: uppercase;""
onclick=""changeConnectionString()"" />" & "</div>")
stringBuilder.Append("<div id=""dashboard"" style=""width:100%;
height:94%;""></div>")
DashboardCryptoProvider provider = new
DashboardCryptoProvider(EncryptionTypes.AES);
Connection connectionName = new Connection { DataSources = new
List<DataSourceConnectionString>() };
connectionName.DataSources.Add(new DataSourceConnectionString
{
Name = "Worldwide Car Sales (Random data)",
ConnectionString = "Data Source = .; Integrated Security=True; Initial
Catalog = NORTHWND; MultipleActiveResultSets=True"
});
Dim EncryptedString =
provider.DoEncryption(Converter.SerializeObject(connectionName));
stringBuilder.Append("<script type=""text/javascript""
language=""javascript"">")
//Button event calling in script
stringBuilder.Append("function changeConnectionString() {var dashboardObj =
$('#dashboard').data('ejDashboardViewer');dashboardObj.model.onApplyConnecti
on = 'onApplyConnection';dashboardObj.redrawDashboard();}function
onApplyConnection(sender) {
sender.data.modifiedConnectionStrings = JSON.stringify(EncryptedString);}")
stringBuilder.Append("</script>")
stringBuilder.Append("</body>")
Return stringBuilder.ToString()
End Function
```

Change data connection string in dashboard at runtime for LightSwitch HTML Application

At initial rendering of dashboard

Register a method to an event `onApplyConnection` in `Dashboard.lsml.js` file like below.

JS

```
myapp.dashboard.ScreenContent_render = function (element, contentItem) {
$('#dashboard').ejDashboardViewer(
{
//Change the connection string in dashboard
onApplyConnection: 'changeConnectionString'
});
});
};
```

Include the modified connection string and the event handler method implementation under `<script>` tag in `Default.html` file like below.

JS

```
<script type="text/javascript">
//Function calling in script
function changeConnectionString(sender) {
sender.data.modifiedConnectionStrings = JSON.stringify(EncryptedString);
}
</script>
```

After dashboard loading

Initialize a button control and register a method to its onclick event `onApplyConnection` in `Dashboard.lsml.js` file like below.

JS

```
myapp.dashboard.ScreenContent_render = function (element, contentItem) {
//Button creation
$(element).append('<div style="padding: 4px 4px 0px
7px;height:4%;background-color: #303030 "><button value="Change Connection
String" style="border: none; -ms-border-radius: 3px; border-radius:
3px;padding: 5px 8px 5px 8px;background-color:       #0080ff; color:
#ffffff;font-size: 72%;text-transform: uppercase;"
onclick="changeConnectionString()" /></div> ');
$(element).append('<div id="dashboard" style="width:100%;
height:94%;"></div>');
//…. …. …
};
```

Include the modified connection string, the event invoker and the event handler method implementation under `<script>` tag in `Default.html` file like below.

JS

```
<script type="text/javascript">
//Button event calling in script
function changeConnectionString()
{
var dashboardObj = $('#dashboard').data('ejDashboardViewer');
dashboardObj.model.onApplyConnection = 'onApplyConnection';
dashboardObj.redrawDashboard();
}
function onApplyConnection(sender)
{
var changeString = PromiseResult.EncryptedString
sender.data.modifiedConnectionStrings = JSON.stringify(changeString);
}
</script>
```

Add the following code snippet in DashboardViewerController.cs file.

C#

```
public Dictionary<string, string> Get() {
Dictionary<string, string> reportDetails = new Dictionary<string, string>();
```



```
DashboardCryptoProvider provider = new
DashboardCryptoProvider(EncryptionTypes.AES);
Connection connectionName = new Connection { DataSources = new
List<DataSourceConnectionString>() };
connectionName.DataSources.Add(new DataSourceConnectionString
{
Name = "Worldwide Car Sales (Random data)",
ConnectionString = "Data Source = .; Integrated Security=True; Initial
Catalog = NORTHWND; MultipleActiveResultSets=True"
});
var EncryptedString =
provider.DoEncryption(Converter.SerializeObject(connectionName));
reportDetails.Add("EncryptedString", EncryptedString);
}
```

How to configure the filter panel programmatically outside of the dashboard viewer

Syncfusion Dashboard Platform SDK provides the support to customize the dedicated filter panel (Refer here to create a filter panel in Dashboard Designer)

Configure a filter panel outside of the Dashboard Viewer by using below APIs.

- filterPanelSettings,
- openFilterPanel,
- closeFilterPanel,
- beforeFilterPanelOpen,
- filterPanelOpen,
- beforeFilterPanelClose,
- filterPanelClose.

To create and bind the filter panel in outside of the Dashboard Viewer, Create a “div” element with “id” attribute to showcase the filter panel and apply styles to the filter panel for filter panel display, refer the below code example.

Example

HTML

```
<div id="filter" style="float: right; width:30%; height:99%; margin-right: -
6px;"></div>
```

1.filterPanelSettings()

filterPanelSettings has the below four inner level properties.

- showIcon
- filterPanelId
- showHeader
- showCloseIcon

showIcon:

It returns the Boolean value. You can show or hide the filter panel icon in the filter panel.

filterPanelId:

It returns the string value. you should pass the filter panel element id as a value to this this filterPanelId API, (i.e.) filterPanelId: "filter".

showHeader:

It returns the Boolean value. You can show or hide the header in filter panel

showCloseIcon:

It returns the Boolean value. You can show or hide the close icon in filter panel.

JS

```
<script>
$(document).ready(function () {
$("#dashboard").ejDashboardViewer({
filterPanelSettings: {
showIcon: false,
filterPanelId: "filter",
showHeader: true,
showCloseIcon: true
},
});
});
</script>
```

2.openFilterPanel ()

To open the filter panel, use this openFilterPanel method API, refer the below example,

JS

```
<script type="text/JavaScript">
var dashboardObject;
$(document).ready(function () {
$("#dashboard").ejDashboardViewer({});
dashboardObject = $('#dashboard').data("ejDashboardViewer");
dashboardObject.openFilterPanel();
});
</script>
```

3.closeFilterPanel ()

To close the filter panel, use this closeFilterPanel method API, refer the below example,

JS

```
<script type="text/JavaScript">
var dashboardObject;
$(document).ready(function () {
$("#dashboard").ejDashboardViewer({});
dashboardObject = $('#dashboard').data("ejDashboardViewer");
dashboardObject.closeFilterPanel();
});
</script>
```

4.beforeFilterPanelOpen

This event fires before the filter panel is open on Dashboard Viewer. Refer the below example,

JS

```
<script type="text/JavaScript">
$(document).ready(function () {
$("#dashboard").ejDashboardViewer({
beforeFilterPanelOpen: "beforeFilterPanelOpenEvent"
});
});
function beforeFilterPanelOpenEvent() {
alert("beforeFilterPanelOpen event is invoked");
}
</script>
```

5.filterPanelOpen

This event fires after the Filter panel is opened. Refer the below example,

JS

```
<script type="text/JavaScript">
$(document).ready(function () {
$("#dashboard").ejDashboardViewer({
filterPanelOpen: "filterPanelOpenEvent"
});
});
function filterPanelOpenEvent() {
alert("filterPanelOpen event is invoked");
}
</script>
```

6.beforeFilterPanelClose

This event fires before the filter panel is close on Dashboard Viewer. Refer the below example,

JS

```
<script type="text/JavaScript">
$(document).ready(function () {
$("#dashboard").ejDashboardViewer({
beforeFilterPanelClose: "beforeFilterPanelCloseEvent"
});
});
function beforeFilterPanelCloseEvent() {
alert("beforeFilterPanelClose event is invoked");
}
</script>
```

7.filterPanelClose

This event fires after Filter panel is closed. Refer the below example,

JS

```
<script type="text/JavaScript">
$(document).ready(function () {
$("#dashboard").ejDashboardViewer({
filterPanelClose: "filterPanelCloseEvent"
});
});
```



```
function filterPanelCloseEvent() {
alert("filterPanelClose event is invoked");
}
</script>
```

Finally, apply the below code example to configure the filter panel in outside of the Dashboard Viewer

JS

```
<div class="content-container-fluid" style="height:100%;">
<div class="row" style="height:100%;">
<div class="cols-sample-area" style="height:100%;width:73.5%;overflow:hidden
!important;">
<div id="dashboard" style="float:left;height:100%;width:100%;"></div>
<div id="filter" style="float:right;width:30%; height:99%;margin-right:-
6px;"></div>
</div>
<div id="sampleProperties" style="margin-bottom:50px">
<div class="prop-grid prop-grid content" style="overflow:hidden
!important;">
<table>
<tr>
<td class="chkrad">
<label for="Checkbox1">Show Filter Panel</label>
<input style="margin:6px;" type="checkbox" id="chkbox" value="showing
dashboard" />
</td>
</tr>
</table>
</div>
<div class="panel-body">
<div id="property-window" class="box wide">
<div>
<div class="heading">
<span>Event Trace</span>
</div>
<div class="prop-grid content">
<div class="eventarea">
<div class="EventLog" id="EventLog">
</div>
</div>
</div>
<div class="evtbtn">
<div class="evtbtn">
<input type="button" style="margin-left:120px;" class="eventclear e-btn"
value="Clear" onclick="onClear()" />
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<script type="text/javascript">
var dashboardObject,checkBoxObj;
```

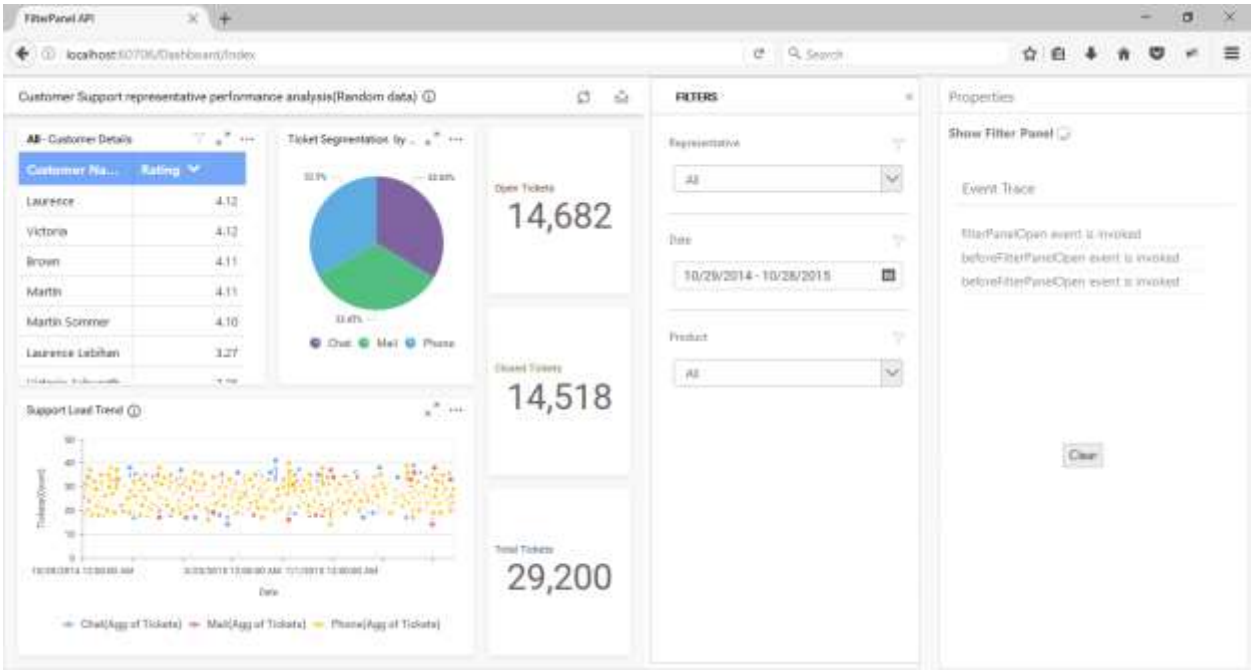


```
$(document).ready(function () {
// declaration
$("#dashboard").ejDashboardViewer({
serviceUrl:
"http://dashboardsdkdemo.syncfusion.com/DashboardService/DashboardService.svc",
dashboardPath:
"http://dashboardsdkdemo.syncfusion.com//Dashboards//WorldWideCarSalesDashboard.sydx",
beforeFilterPanelOpen: "beforeFilterPanelOpenEvent",
filterPanelOpen: "filterPanelOpenEvent",
beforeFilterPanelClose: "beforeFilterPanelCloseEvent",
filterPanelClose: "filterPanelCloseEvent",
filterPanelSettings: {
showIcon: false,
filterPanelId: "filter",
showHeader: true,
showCloseIcon: true
},
});
dashboardObject = $('#dashboard').data("ejDashboardViewer");
checkBoxObj = $("#chkbox").ejCheckBox({
change: "checkBoxChange"
});
});
$("#sampleProperties").ejPropertiesPanel();
function beforeFilterPanelOpenEvent() {
$("#dashboard").css('width', '70%');
this.resizeDashboard();
jQuery.addEventLog("beforeFilterPanelOpen event is invoked");
}
function filterPanelOpenEvent() {
jQuery.addEventLog("filterPanelOpen event is invoked");
}
function beforeFilterPanelCloseEvent() {
$("#dashboard").css('width', '100%');
this.resizeDashboard();
jQuery.addEventLog("beforeFilterPanelClose event is invoked");
}
function filterPanelCloseEvent() {
$("#chkbox").ejCheckBox({checked: false});
jQuery.addEventLog("filterPanelClose event is invoked");
}
function onClear() {
jQuery.clearEventLog();
}
function checkBoxChange(e) {
if (dashboardObject) {
if ($("#chkbox").is(":checked")) {
dashboardObject.openFilterPanel();
$("#filter").show();
}
else {
dashboardObject.closeFilterPanel();
$("#filter").hide();
}
}
```



```
}
</script>
```

The above sample illustrates the below screenshot.



Linking Customization

The Dashboard Platform SDK provides support to customize the Hyperlink before or after navigation through the following JavaScript API's in Dashboard Viewer.

- beforeNavigate
- afterNavigate

beforeNavigate Event

This event is triggered before navigating to the linked URL of the widget and it can be handled to customize like either change the target URL or target location.

Refer the below code snippet, for the Hyperlink action:

JS

```
var targetType = "_blank";
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
beforeNavigate: function(args) {
args.data.target = targetType;
//Get the Source and Target information for your customization
}
}
);
```


Where the args contains the following values:

- url holds linking URL
- target holds the default target type as '_blank'. You can change target types as follows,

Target Type	Behavior
_blank	New tab or page
_self	Same page
_parent	Parent element
_top	Topmost Parent element
frameName	Specific frame

- holds boolean value to handle navigation of link action.
- source holds linked widget information and dashboard information where the navigation action applied.

Note: If target value is null or empty, then default target value will be "_blank" and Hyperlink will navigate into either new tab or page. By default, value for the 'handled' API is false. and we can set 'handled' as true, which restrict the navigation of Hyperlink.

Refer the below code snippet how to restrict the Internal Linking (in between tabs of the multi-tabbed dashboard)

JS

```
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
beforeNavigate: function(args) {
args.data.handled = true;
//Get the Source and Target information for your customization
}
}
);
```

Where args contains the following values:

- tabName holds the target tab name which gets opened
- handled holds boolean value to handle navigation.
- source holds linked widget information and dashboard information where the navigation action applied.

afterNavigate Event

This is triggered once navigation operation is completed. This event just provides information as navigation is done.

Refer the below code snippet, for the Hyperlink action:

JS

```
var targetType = "_blank";
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
afterNavigate: function(args) {
args.data.target = targetType;
//Get the Source and Target information for your customization
}
}
);
```

Where args contains the following values:

- url holds URL, which is opened.
- target holds the target type of the above URL is opened.
- source holds linked widget information and dashboard information where the navigation action applied.

Refer the below code snippet for the Internal Linking (in between tabs of the multi-tabbed dashboard)

JS

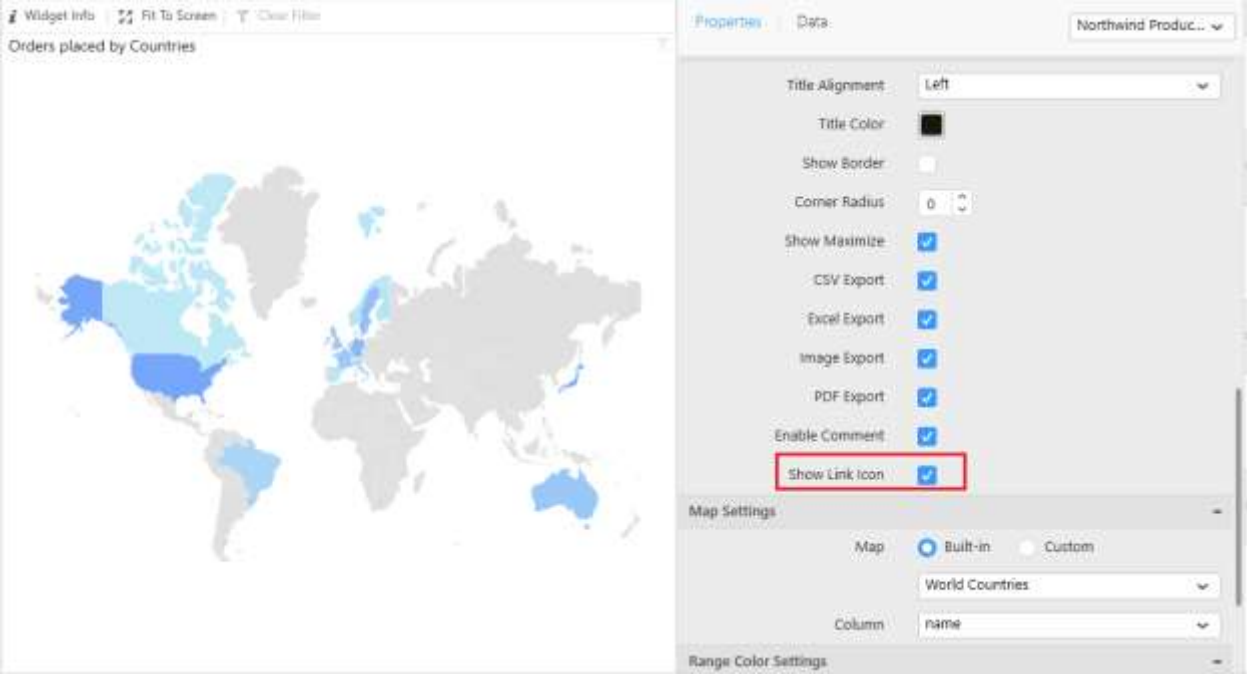
```
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
afterNavigate: function(args) {
//Get the Source and Target information for your customization
}
}
);
```

Where args contains the following values:

- tabName holds the target tab name, which is opened.
- source holds linked widget information and dashboard information where the navigation action applied.

How to show, whether the dashboard widget has navigation in Dashboard Viewer?

The icon for the navigation in dashboard widget can be shown or hide by checking the “Show Link Icon” property in respective widget in Dashboard Designer.



Refer the below image shows the navigation icon in Dashboard Viewer

How to hide the dashboard tabs to render a single tab and also switching between the dashboard tabs

The Dashboard Platform SDK provides the support for hiding the tabs in multi-tabbed dashboard to render a single tab and also supports to switch between the tabs using the following API's programmatically.

- getDashboardTabsInfo()
- selectTabByName(tabName)
- selectTabByIndex(tabIndex)

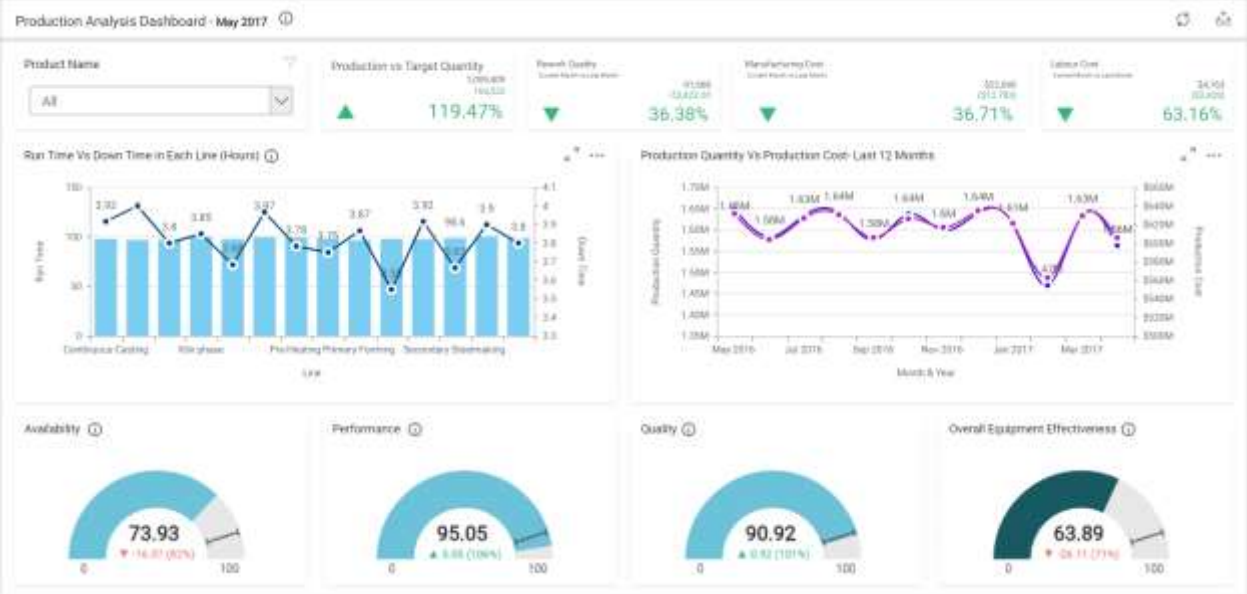
Hiding the tabs

The showTab API to control the show or hide behavior of multi-tabbed dashboard tabs. To hide the dashboard tabs and render a single tab, refer the below example,

JS

```
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
showTab:false
}
);
```

The below sample illustrates the above code snippet,



Note: By default, the tabs will be shown. The **showTab** API value is boolean. You can customize by setting this API value as true or false.

Before switching between the dashboard tabs

You can switch between the dashboard tabs, by following below procedures,

1. To create a dashboard instance, add the below code,

JS

```
function switchTabs()
{
var dashboard = $("#dashboard").data("ejDashboardViewer");
}
```

2. To get the current multi tabbed dashboard tabs information, call the getDashboardTabsInfo() API with dashboard instance.

JS

```
var tabInfo = dashboard.getDashboardTabsInfo();
```

Here **tabInfo** is the collection of objects contains each tab information with tabIndex, tabName and tabId.

Switching between the dashboard tabs using API's

You can achieve tab switching by using following API's

1. selectTabByIndex(tabIndex)
2. selectTabByName(tabName)

Tab switching works even, if we set the **showTab** API as **true**.

1. selectTabByIndex(tabIndex)

To select a particular tab, pass the respective tabIndex value as input to the selectTabByIndex API using dashboardInstance.For example,

JS

```
function switchTabs()
{
var dashboard = $("#dashboard").data("ejDashboardViewer");
var tabInfo = dashboard.getDashboardTabsInfo();
dashboard.selectTabByIndex(tabInfo[1].tabIndex);
}
```

Note: If you pass tab index value as Empty or invalid index value, the current selected tab only in active state. There is no change in tabSelection.

2. selectTabByName(tabName)

To select a tab, pass the tabName as input to the selectTabByName API using dashboardInstance. For example, if you may want to select the same tab with tabName, refer the below example,

JS

```
function switchTabs()
{
var dashboard = $("#dashboard").data("ejDashboardViewer");
var tabInfo = dashboard.getDashboardTabsInfo();
dashboard.selectTabByName(tabInfo[1].tabName);
}
```

The below sample illustrates the above code snippets,

Note: If you pass tabName as Empty or invalid tabName, the current selected tab only in active. There is no change in tabSelection.

How to show or hide the get URL link icon in dashboard viewer

The Dashboard Platform SDK provides the support for hiding get URL icon and also support to get current view URL through following dashboard viewer API programmatically.

- getCurrentViewUrl()

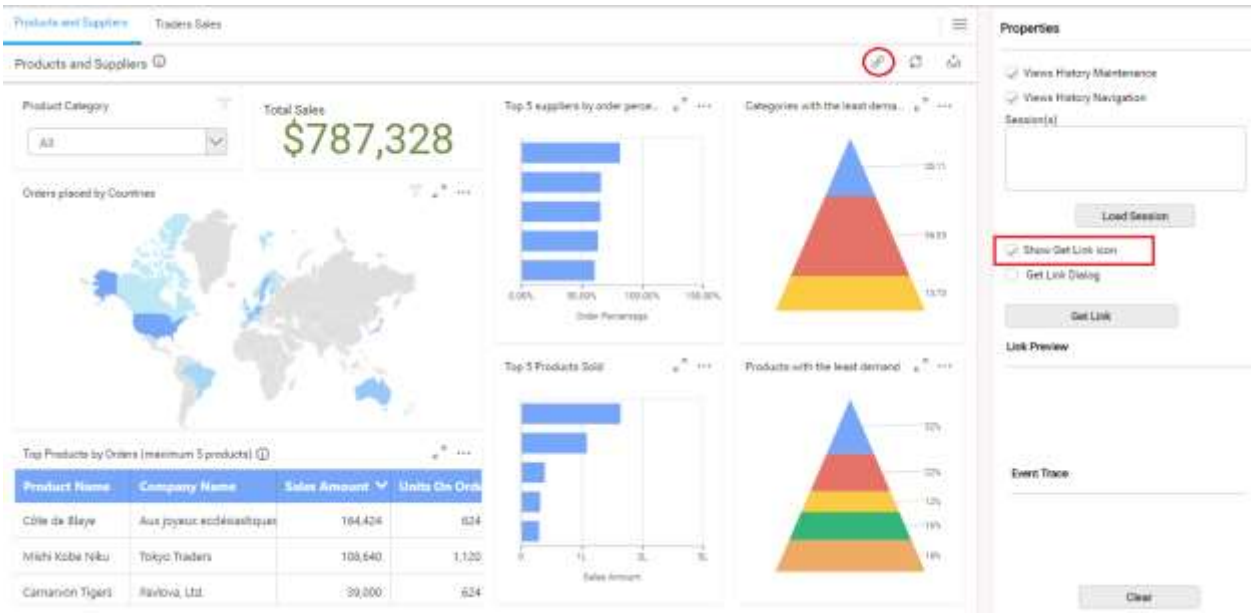
Hiding the get URL icon

The showGetLinkIcon API to show or hide the get link icon, refer the below example,

JS

```
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
dashboardPath:'Path of the dashboard',
showGetLinkIcon: true
}
);
```

The below image illustrates the above code snippet,



Note: By default, the get URL icon will be shown. The **showGetLinkIcon** API value is boolean. You can customize by setting this API value as true or false.

Getting current view URL

You can get the current view URL by following procedures,

1. To create a dashboard instance, add the below code,

JS

```
function getCurrentViewUrl()
{
var dashboard = $("#dashboard").data("ejDashboardViewer");
}
```

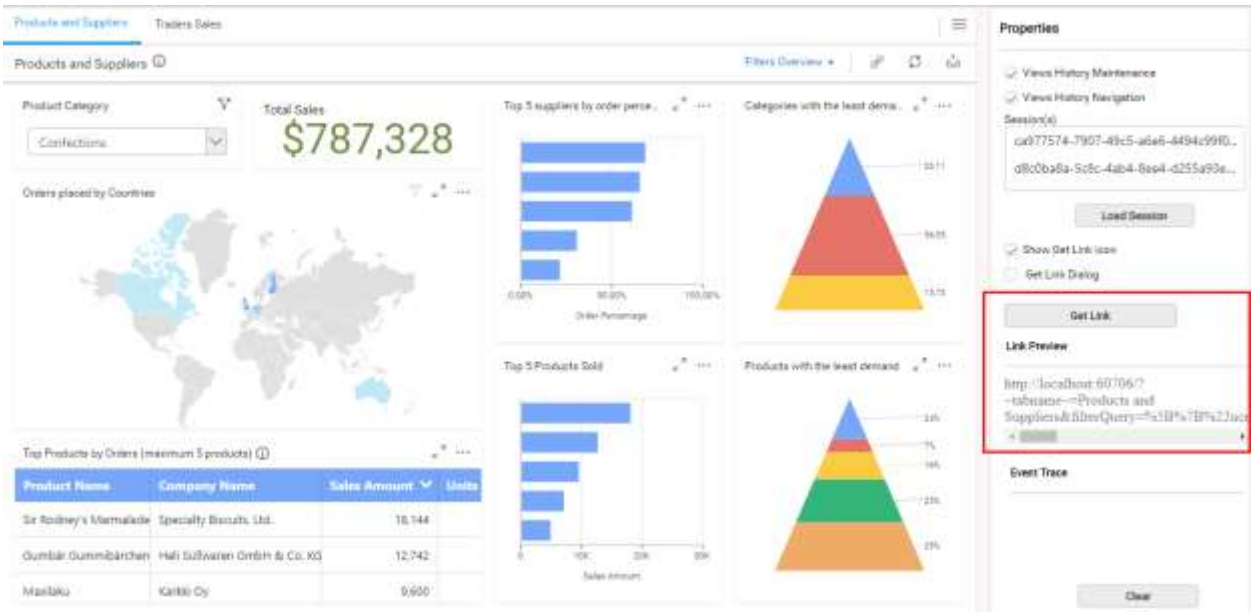
2. To get the current view URL, call the getCurrentViewUrl() API with dashboard instance.

JS

```
var url = dashboard.getCurrentViewUrl();
```

Here **url** is the current dashboard view URL with applied filters

The below image illustrates the above code snippets,



How to host Dashboard Service as Sub Application in IIS

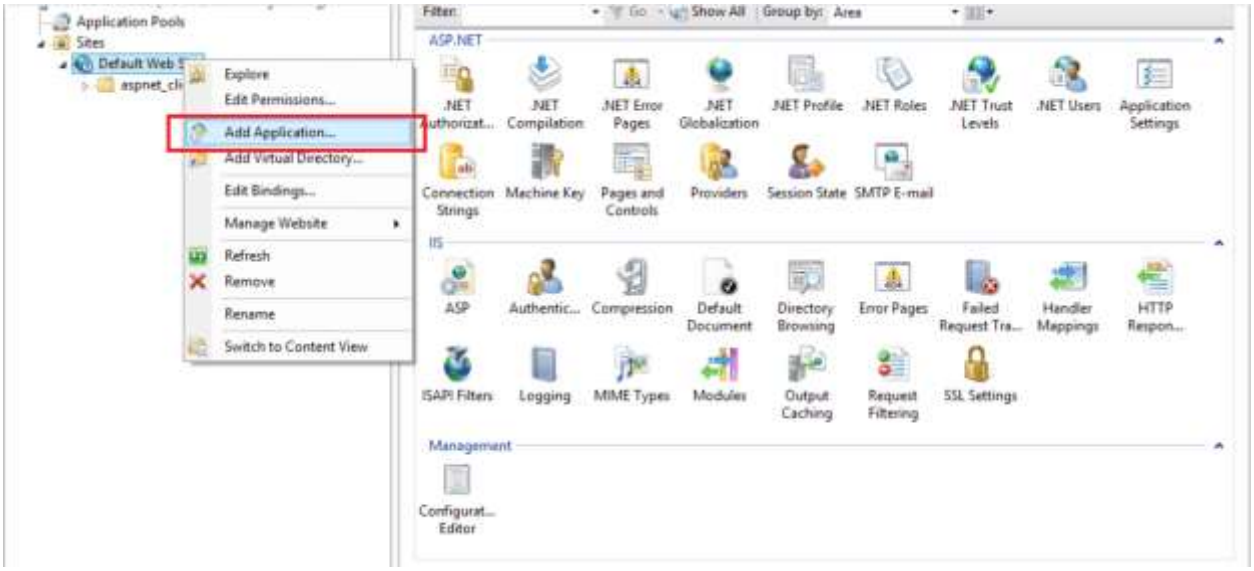
Syncfusion Dashboard Service can be hosted as a sub application in IIS by following the below steps.

Prerequisites

- Refer step by step IIS installation in Windows 7+, Windows Server 2008 and 2008 R2, Windows Server 2012 or Windows Server 2012 R2
- To host the Dashboard Service in IIS requires a prerequisite called WCF should be enabled in IIS. Refer the article how to configure the WCF in IIS.

Adding Syncfusion Dashboard Service as a sub application

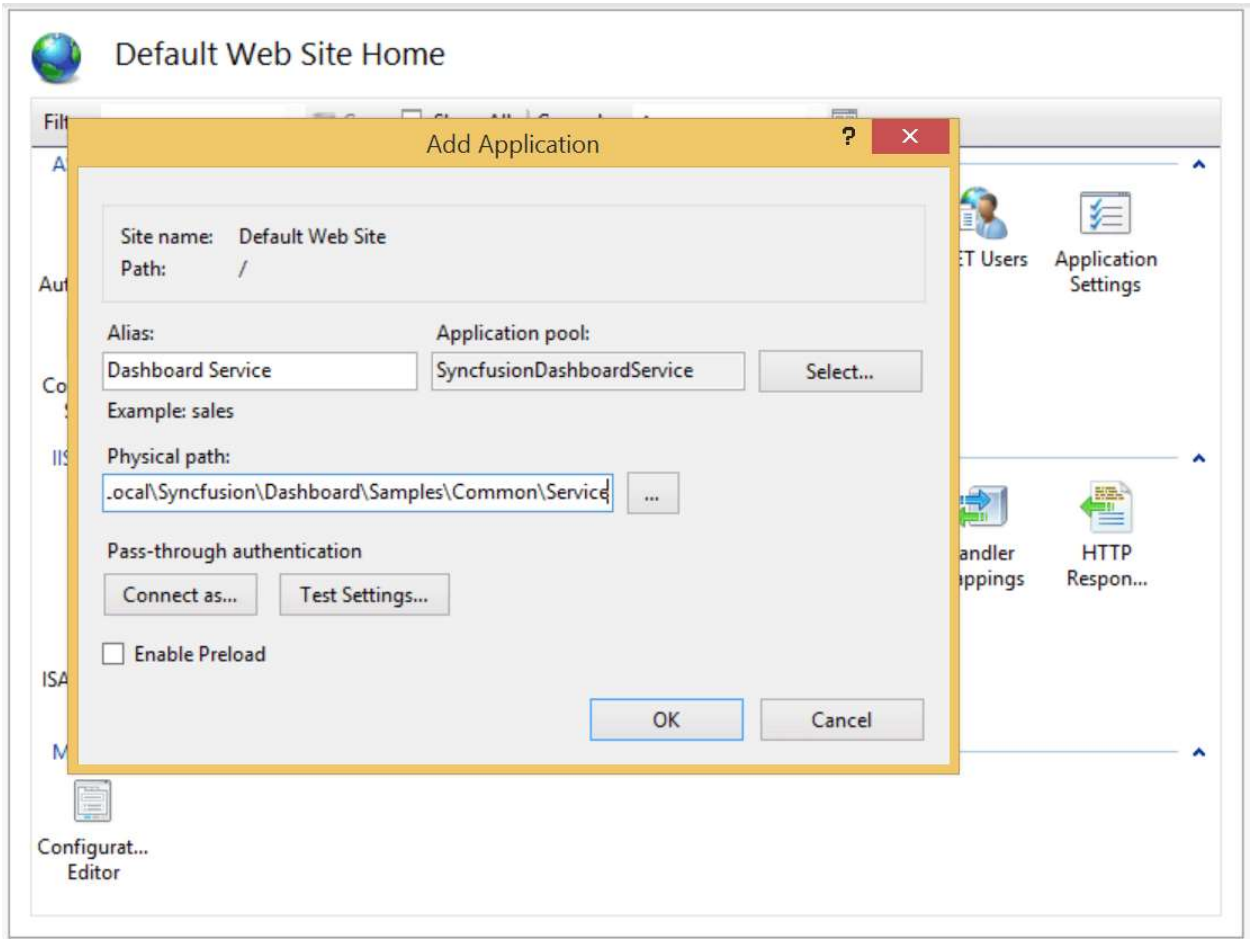
Select the Web Site under which you need to host the Syncfusion Dashboard Service as a sub application. Do a right mouse button click and then select Add Application as shown in the below image.



Fill in the below items in the new application dialog and click on OK

1.**Alias name**- Enter the name that you wish to be an alias for the domain 2.**Application pool**- Application Pools are logical groupings of web applications that will execute in a common process 3.**Physical path**- In the Physical path box, type in the location of the Syncfusion Dashboard Service in the machine, or click the browse button to navigate the file system to find the location.for eg

%localappdata%\Syncfusion\Dashboard Platform SDK\Service



SSL

To enable SSL for the Dashboard Service application, you will need a valid SSL certificate. Please check the below link on how to Obtain an SSL certificate and install it to a website in IIS.

http://www.iis.net/learn/manage/configuring-security/how-to-set-up-ssl-on-iis

Configuring SSL for Syncfusion Dashboard Service

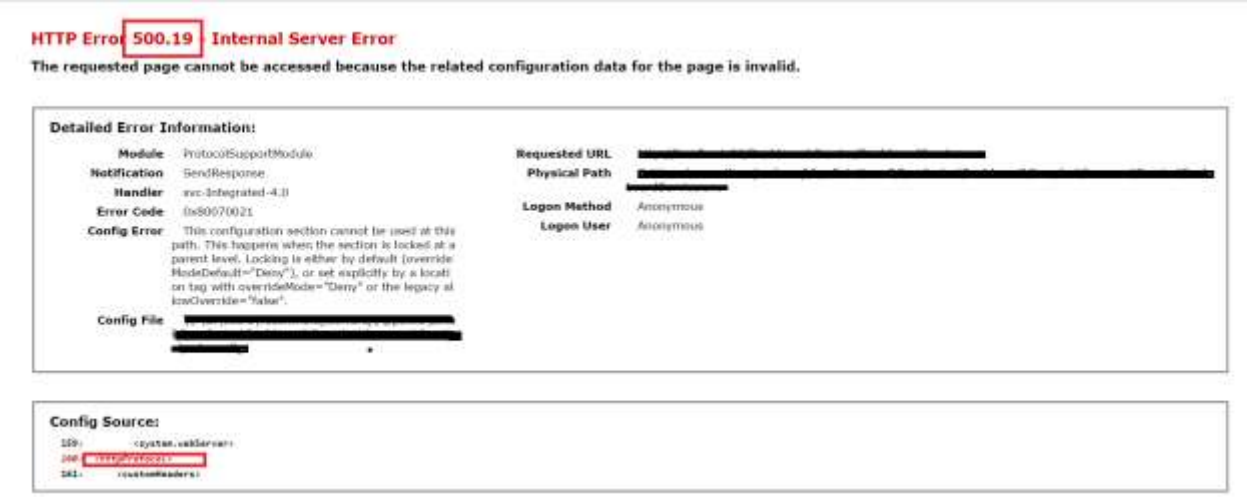
To make the Dashboard Service run under SSL you need to enable support for SSL in the Dashboard Service using the Syncfusion Dashboard Service Configuration Manager utility. Please follow the steps in the below link to configure SSL for Dashboard Service.

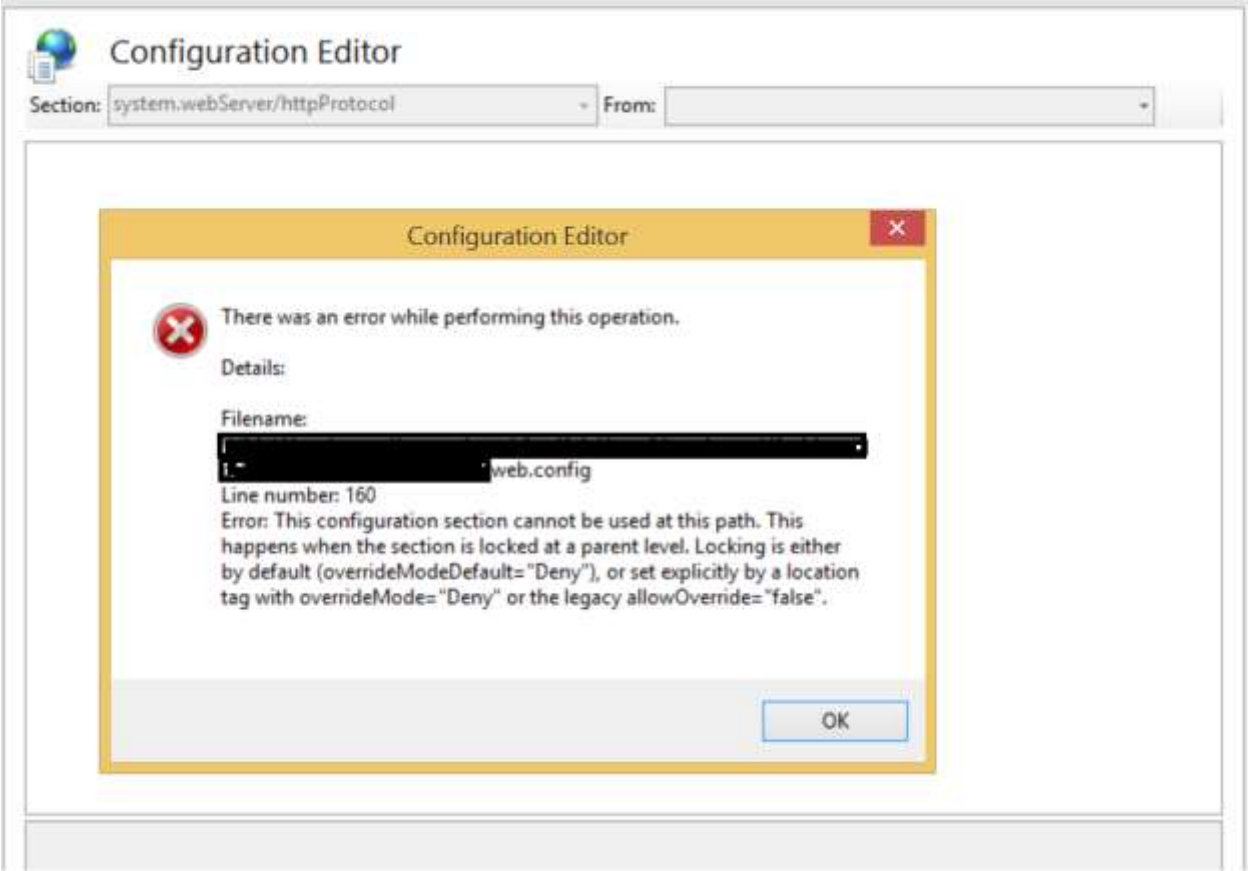
Configuring SSL for Dashboard Service

Troubleshooting

HTTP 500.19

This error will be thrown when there is an error with the Service configuration. And this will commonly occur if a configuration required for the Dashboard Service is locked by the parent web site of the root configuration of the system.

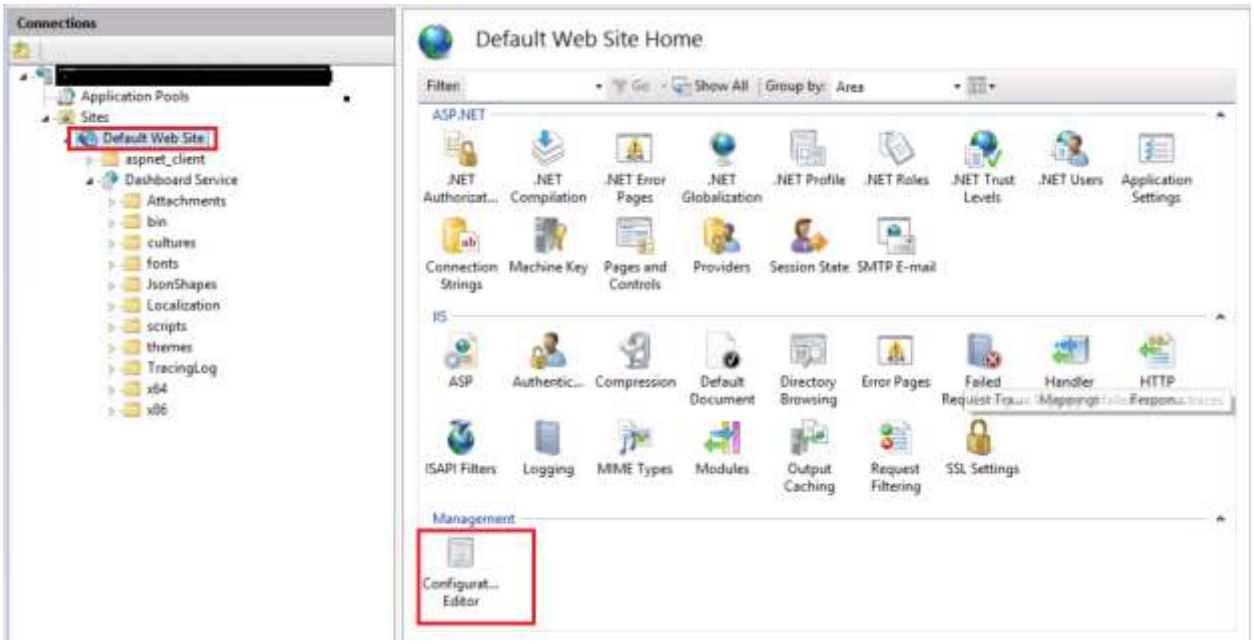




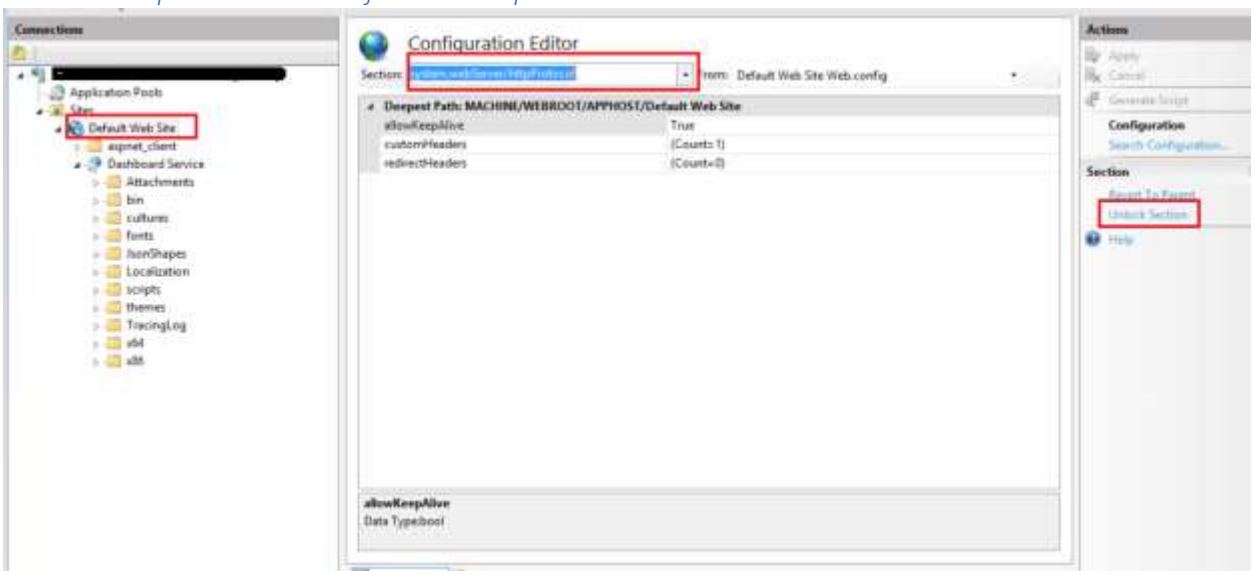
You can clear this error by unlocking the locked configuration section in the parent site by following the below steps.

In the screenshot shown above the service is not able to host since the httpProtocol section is locked at the parent website so the Dashboard Service related changes under this configuration is not applied

Select the Parent Website and then open the Configuration Editor by double clicking icon “Configuration Editor”



Select the httpProtocol section from the drop down and select unlock section



The service will be launched successfully now

Note

If you get above 500.19 error, then You can clear this error by unlocking the locked configuration section in the parent site. If still the problem exists, traverse the hierarchy tree to find where the configuration is locked and then unlock it there and save the configuration.

Dashboard Server's service can be used only to render the Dashboard, which is available in the Syncfusion Dashboard Server. If you want to render the Dashboard file (*.SYDX) that is designed in the Syncfusion Dashboard Designer you need to use the Syncfusion Dashboard SDK's service

How to host Dashboard Service as WebSite in IIS

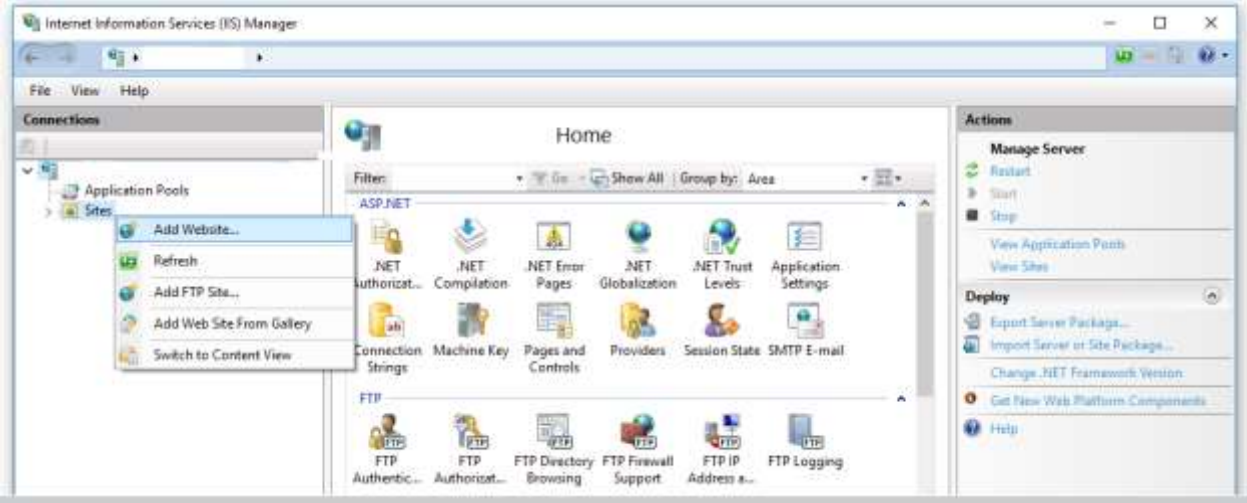
Syncfusion Dashboard Service can be hosted as a Website in IIS manually by following the below steps or by using the tool SyncfusionDashboardServiceInstaller

Prerequisites

- Refer step by step IIS installation in Windows 7+, Windows Server 2008 or 2008 R2, Windows Server 2012 or Windows Server 2012 R2
- To host the Dashboard Service in IIS requires a prerequisite called WCF should be enabled in IIS. Refer the article how to configure the WCF in IIS.

Steps

- Open IIS Manager
- In the Connections pane, right-click the Sites node in the tree, and then click Add website as shown below.



In the Add Web Site dialog box, fill the following details as follows

1.Site name

Type a name for your Web site in the Web site name box. for eg SyncfusionDashboardService

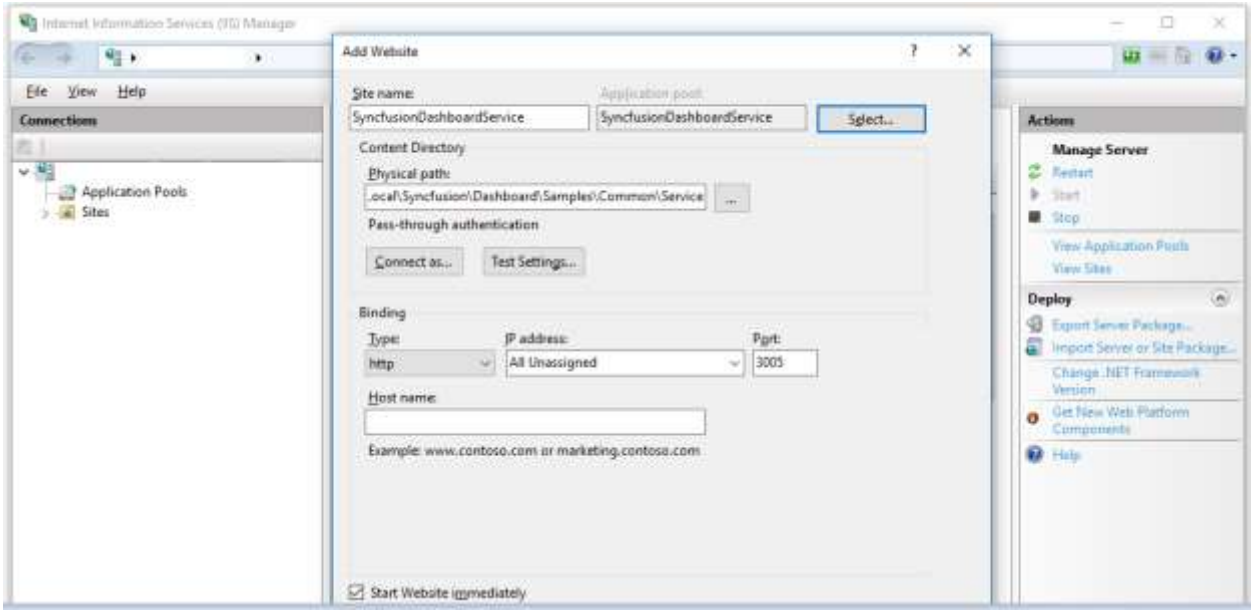
2.Physical path

In the Physical path box, type in the location of the Syncfusion Dashboard Service in the machine, or click the browse button to navigate the file system to find the location.for eg

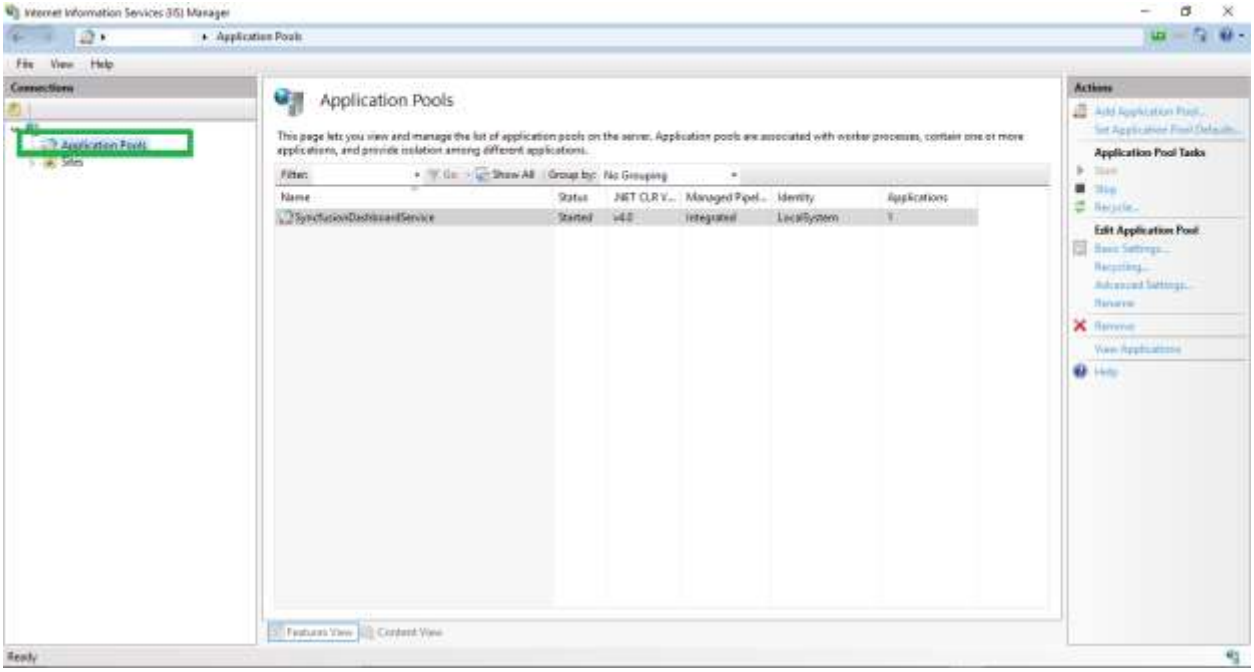
%localappdata%\Syncfusion\Dashboard Platform SDK\Service

3.Port

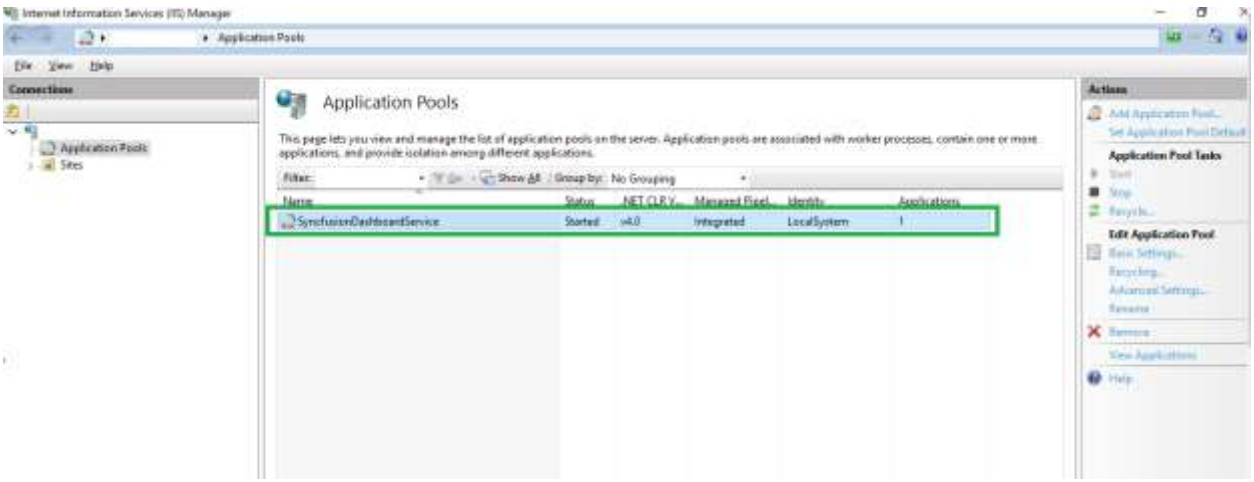
Specify the port at which the SyncfusionDashboardService need to be hosted at.for eg 3005

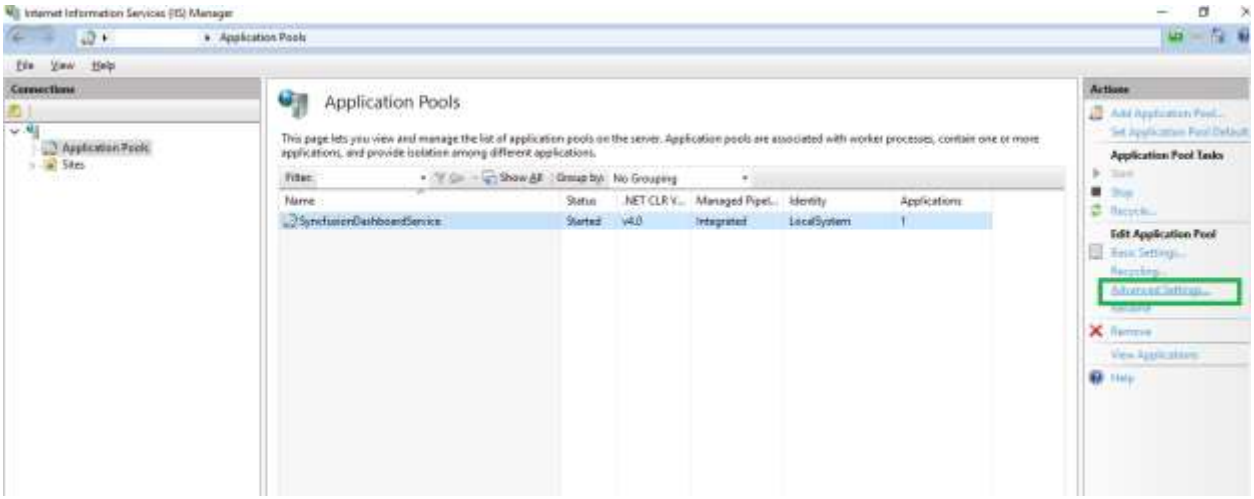


- In the Connections pane, expand the server node and click Application Pools.

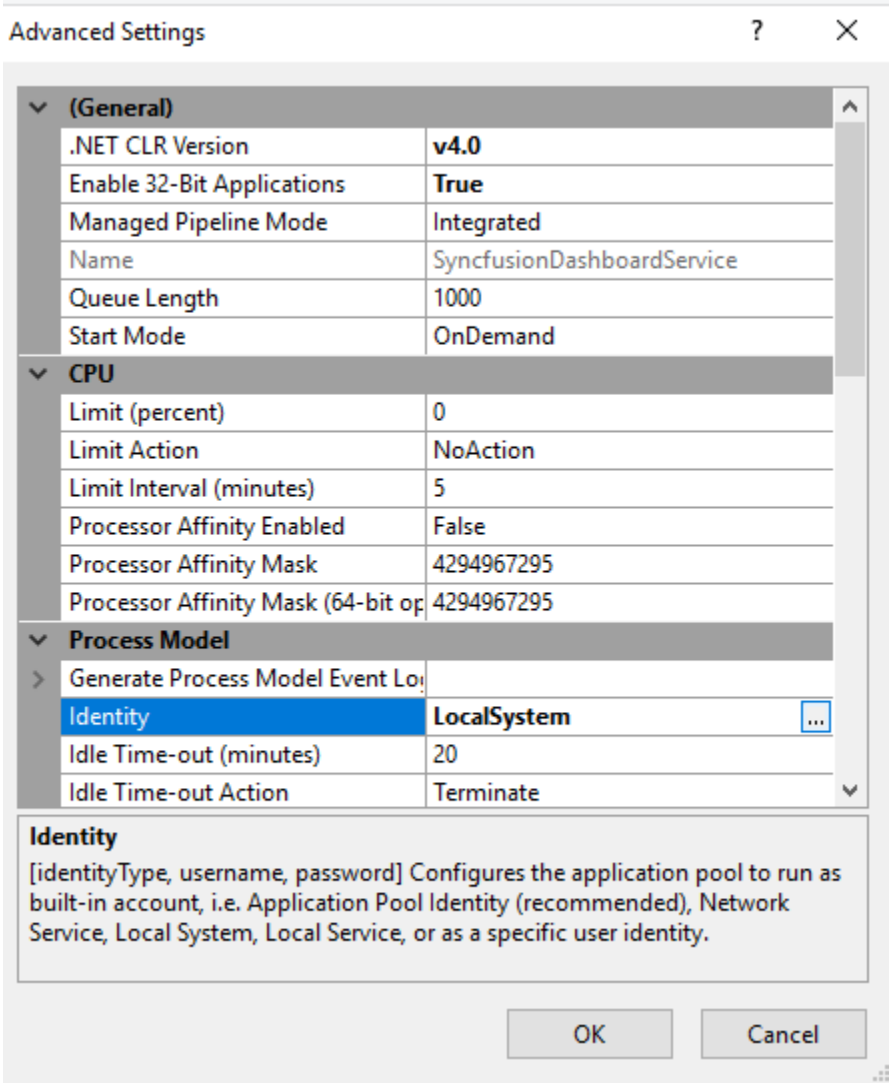


- Specify the name of the application pool in the attached image.

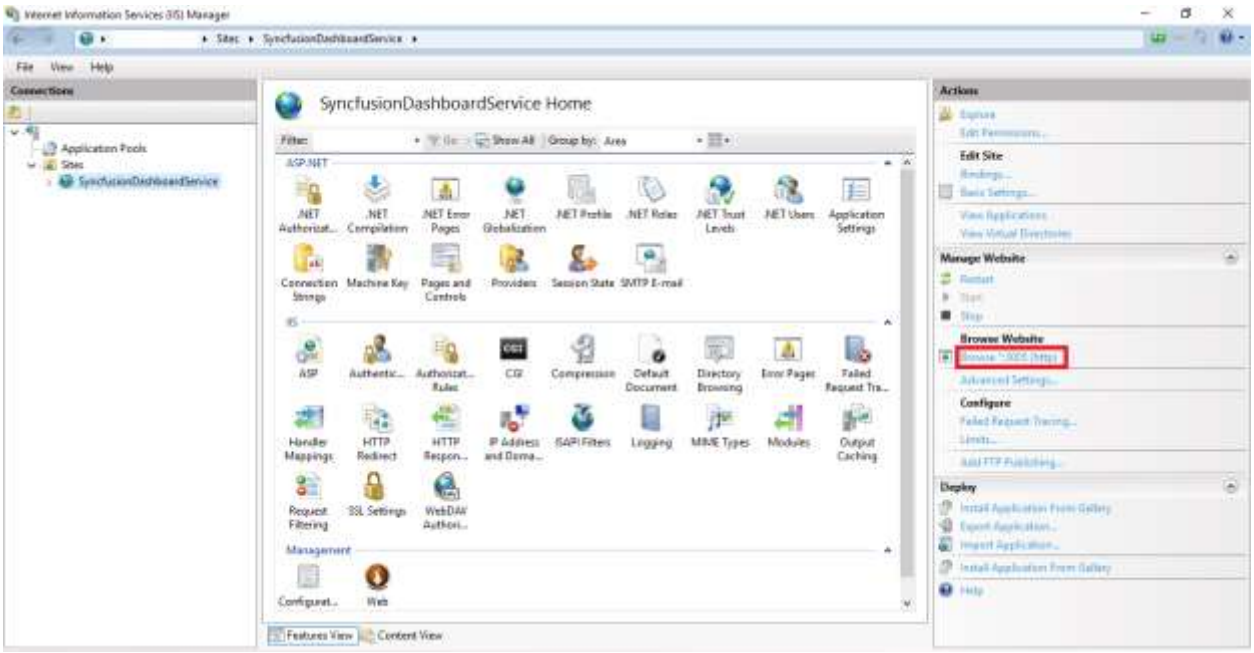




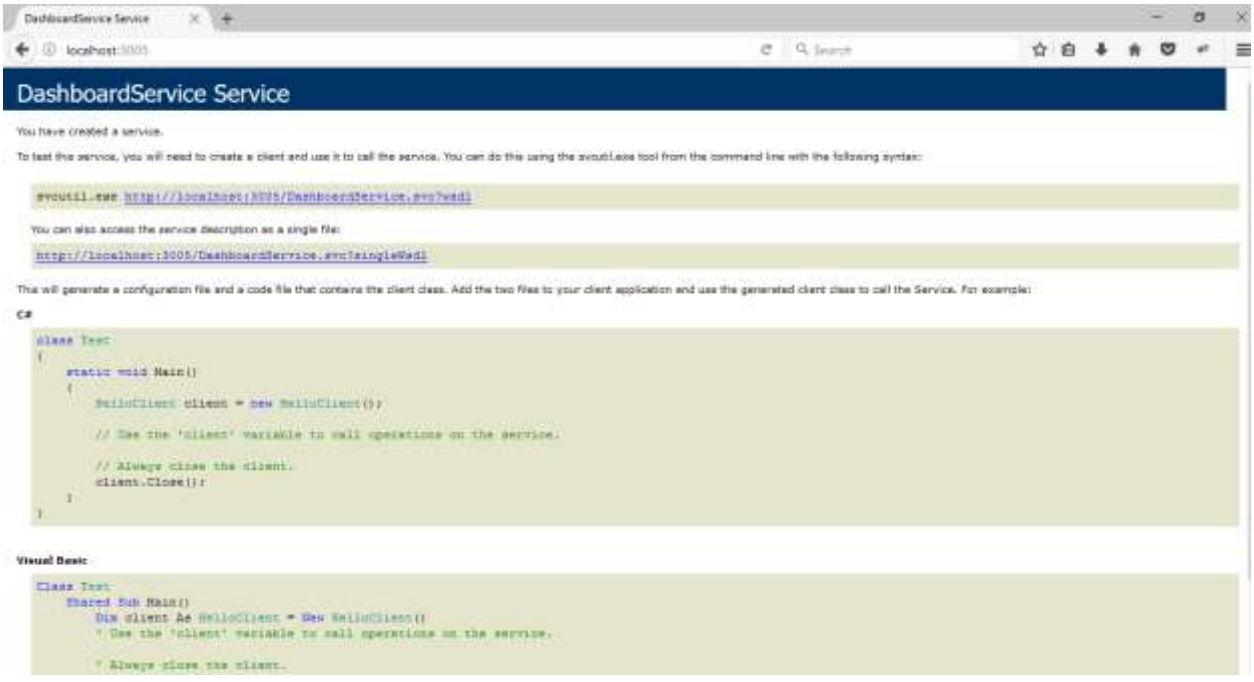
- set the Identity property as LocalSystem under the ProcessModel of Advanced Settings dialog box and then click OK.



- Expand the Sites node in the Connections pane, select the Website name SyncfusionDashboardService then click Browse Link under the Browse Website section of Action pane as below screenshot



- The hosted service will be launched into the default browser as below



Note: Please find the troubleshooting tips here if the "HTTP ERROR 404.17-Not Found" error occurs on hosting the Dashboard Service.

SSL

To enable SSL for the Dashboard Service application, you will need a valid SSL certificate. Please check the below link on how to Obtain an SSL certificate and install it to a website in IIS.

http://www.iis.net/learn/manage/configuring-security/how-to-set-up-ssl-on-iis

Configuring SSL for Syncfusion Dashboard Service

To make the Dashboard Service run under SSL you need to enable support for SSL in the Dashboard Service using the Syncfusion Dashboard Service Configuration Manager utility. Please follow the steps in the below link to configure SSL for Dashboard Service.

Configuring SSL for Dashboard Service

How to host Dashboard Service in the Server where Dashboard Platform is not installed

Syncfusion Dashboard Service can be hosted in the server where dashboard platform is not installed by following the below steps.

- Copy the Service folder form the machine where the Dashboard Platform SDK is installed from the location %localappdata%\Syncfusion\Dashboard Platform SDK\Service.
- Paste the copied files into the server where the dashboard platform is not installed.
- Now you can host the Dashboard Service by following any of the below methods:
    1. Hosting Dashboard Service in IIS
    2. Hosting Dashboard Service in IIS Express
    3. Hosting Dashboard Service as Windows Service Background Process
    4. Hosting Dashboard Service Manually
    5. Hosting Dashboard Service as sub application

Note: While host the Dashboard Service manually use the Service folder location where it is pasted in the Server as physical path.

How to pass a custom header to fetch the dashboard

To pass a custom header with an authorization token, use the following Dashboard Viewer event:

- beforeDashboardFetch

This event will be triggered when the dashboard is loaded for the first time, and it will be invoked before making the request to service to fetch the layout information.

Refer here to know more about this event.

The following SYDX path is passed to the dashboardPath API of the Dashboard Viewer. To access this, the URL authentication information is needed.

SYDX path:
http://dashboardsdkdemo.syncfusion.com//Dashboards//WorldWideCarSalesDashboard

Refer the following code snippet to pass the custom header for fetching the dashboard or rendering the dashboard layout.

JS

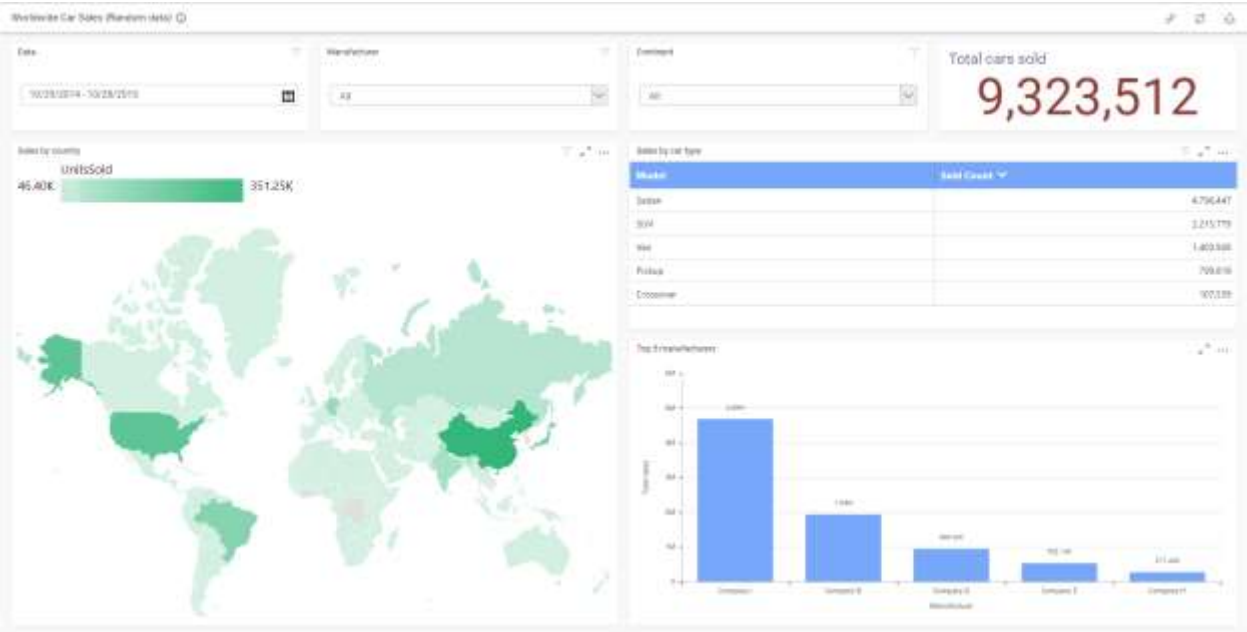
```
$("#dashboard").ejDashboardViewer(
{
serviceUrl: 'service URL',
```



```
dashboardPath:'Path of the dashboard',
beforeDashboardFetch: function (args) {
args.setRequestHeader("Syncfusion", "eRC3HKL3dnaiNOB829HtdyTr5462Hgys"); //
pass the key and value
},
}
);
```

The passed custom header will be added to the setRequestHeader for beforeDashboardFetch event when the call to dashboard path link http://dashboardsdkdemo.syncfusion.com//Dashboards//WorldWideCarSalesDashboard is made to get the dashboard file.

The following screenshot illustrate the dashboard preview:



Filtering Views through URL Parameters

Passing parameter with URL

You can pass parameters to a dashboard by including them in a dashboard URL. Passing parameter values within URL will apply filter in the dashboard on initial load itself.

To set a dashboard parameter within a URL, use the following syntax.

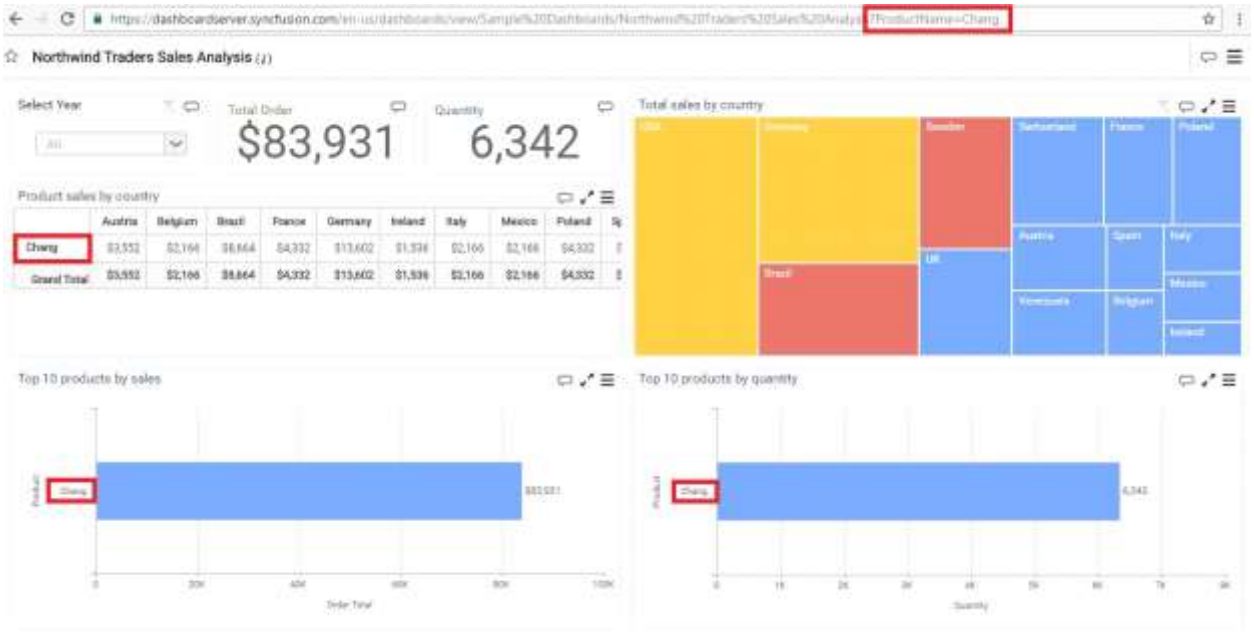
parameter=value1, value2,…, valueN

where parameter represents the column name.

To append your query string made with parameters and values to URL, add a prefix (?) to the query string for single-valued parameter as follows.

http://<servername>/<culturename>/dashboards/view/<category>/<dashboardname>?ProductName=Chang

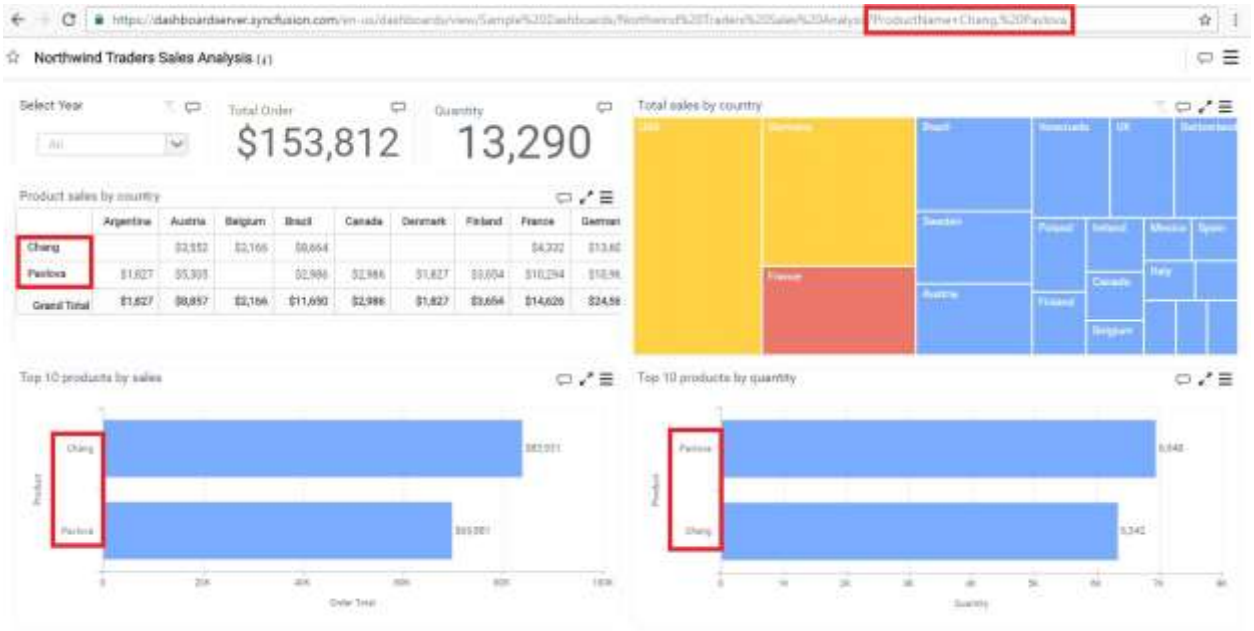
Here is a dashboard view illustrating the same with single-valued parameter.



To append your query string made with parameters and values to URL, add a prefix (?) to the query string for multi-valued parameter as follows.

http://<servername>/<culturename>/dashboards/view/<category>/<dashboardname>?ProductName=Chang,Pavlova

Here is a dashboard view illustrating the same with multi-valued parameter.



Supported operators and date and time functions

You can also define parameters with operators to search one or more values as follows.

Operator	Syntax

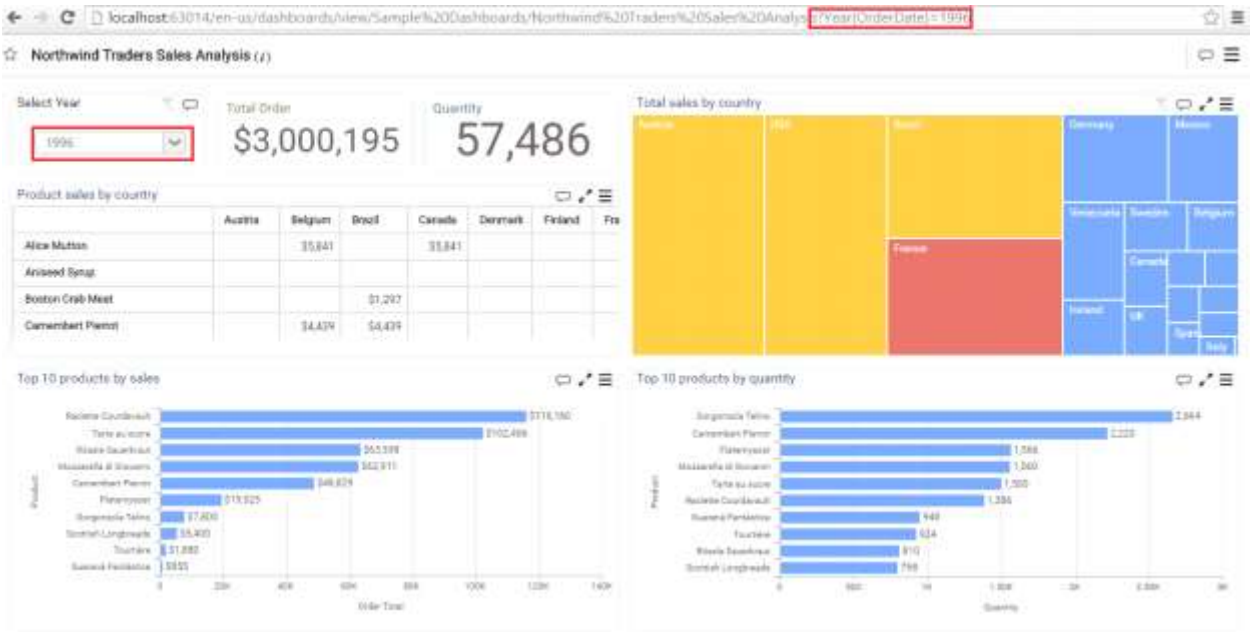
IN	parameter=IN(value1, value2,..., valueN)
NOTIN	parameter=NOTIN(value1, value2, â€¦, valueN)
BETWEEN	parameter=BETWEEN(value1, value2)
INBETWEEN	parameter=INBETWEEN(value1, value2)
STARTSWITH	parameter=STARTSWITH(value)
ENDSWITH	parameter=ENDSWITH(value)
CONTAINS	parameter=CONTAINS(value1, value2)

You can define parameters (date and time typed columns) with date and time functions that are applied to search the formatted date values as follows.

http://<servername>/<culturename>/dashboards/view/<category>/<dashboardname>?Year(OrderDate)=1996

Function	Syntax
YEAR	YEAR(parameter)=value1, value2, â€¦, valueN
MONTHNAME	MONTHNAME(parameter)=value1, value2, â€¦, valueN
QUARTER	QUARTER(parameter)=value1, value2, â€¦, valueN
QUARTERYEAR	QUARTERYEAR(parameter)=value1, value2, â€¦, valueN
MONTHYEAR	MONTHYEAR(parameter)=value1, value2, â€¦, valueN
DAYMONTHYEAR	DAYMONTHYEAR(parameter)=value1, value2, â€¦, valueN
MONTHDAYYEAR	MONTHDAYYEAR(parameter)=value1, value2, â€¦, valueN
HOURS	HOURS(parameter)=value1, value2, â€¦, valueN
MINUTES	MINUTES(parameter)=value1, value2, â€¦, valueN
DAY	DAY(parameter)=value1, value2, â€¦, valueN
SECONDS	SECONDS(parameter)=value1, value2, â€¦, valueN
DATEHOUR	DATEHOUR(parameter)=value1, value2, â€¦, valueN
DAYOFWEEK	DAYOFWEEK(parameter)=value1, value2, â€¦, valueN
DAYOFYEAR	DAYOFYEAR(parameter)=value1, value2, â€¦, valueN
WEEKOFYEAR	WEEKOFYEAR(parameter)=value1, value2, â€¦, valueN

Here is a dashboard view illustrating the use of parameter with date and time function.



Note: You can also define parameters with operators (except STARTSWITH and ENDSWITH) and date time functions combination.

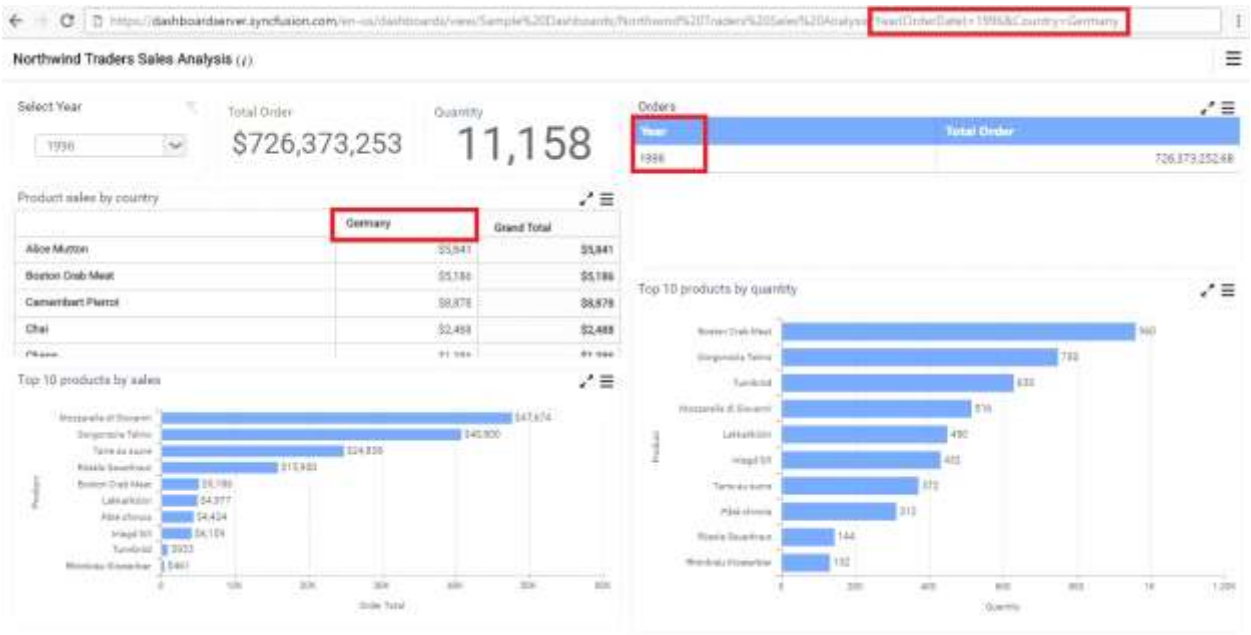
For example, YEAR(OrderDate)=IN(1996,1997)

Passing multiple parameters with URL

You can pass more than one parameter within a URL introducing an ampersand (&) symbol in between them to differentiate like below.

```
http://<servername>/<culturename>/dashboards/view/<category>/<dashboardname>?Year(OrderDate)=1996&Country=Germany
```

Here is a dashboard view illustrating the same.



Information: The parameter names and values are case-sensitive.

Information:

Information: The operators and date and time function names are case-insensitive.

Information:

Information: Characters like comma (,) and ampersand (&) in value should be prefixed and suffixed with tilde (~) symbol to differentiate itself from syntax elements. For example, CompanyName=Syncfusion Inc~,~

Information:

Information: Invalid parameter name will be ignored from the filter consideration.

Information:

Information: Invalid parameter value will result in "No data available to display" in widgets.

How to enable the security for the Dashboard Service at the SDK level

To enable the SDK security for the Dashboard Service, set the `enableSecurity` key as true in Dashboard Service's web.config which is available in the below location

%localappdata%\Syncfusion\Dashboard Platform SDK\Service

Steps to generate the Access Token and pass it to the Dashboard Service

Once the security is enabled, access token must be generated and passed it to the specified API while calling the ejDashboardViewer. If the access token is not passed after the enableSecurity key is set as true the dashboard won't render and it will throw access denied error. Access Token can be generated by calling the GetAccessToken method using AJAX as in the below code snippet.

JS

```
var getToken = function () {
var token = "";
var info = JSON.stringify({
"validationKey": "",  // Pass the validationkey available in the web.config file
"decryptionKey": "",  // Pass the decryptionkey available in the web.config file
"domainName": "" // Pass domain name where your application is hosted.
});
var result = "";
$.ajax({
type: "POST",
url:  "http://localhost/DashboardService.svc/GetAccessToken",
datatype: "json",
contentType: "application/json;charset=utf-8",
async: false,
data: info,
success: function (result) {
token = result.d;
}
});
return token;
}
```

If the Domain Name is passed while generating the access token the Dashboard Service will be accessible within your domain. If an user tries to access the hosted Dashboard Service with the different domain name then the access will be denied to that user/try.

Now pass the generated Access Token to the ejDashboardViewer by specifying accessToken API as per the below code snippet

JS

```
$(function (e) {
```



```
$("#dashboard").ejDashboardViewer({
serviceUrl:
"https://dashboardsdk.syncfusion.com/DashboardService/DashboardService.svc",
// Pass the Dashboard Service URL you have hosted
dashboardPath: "http://dashboardsdk.syncfusion.com//Dashboards//NorthWind
Product Details-Sql.sydx", // Pass the path of the Dashboard File you have
designed.
accessToken : "Generated Access Token"
});
});
```

Note: The dashboardPath and serviceUrl used in the above code snippet are only for demo purpose.

Validity of the Access Token

We have provided the support to set the validity of the access token. The validity of the access token can be set in TokenValidityTime Key which is available in web.config file

Once the validity of the access token is expired the dashboard won't render. To render the dashboard properly you need to generate and pass the new access Token

Note: The Validity of the accessToken can be set in the format of 5d 7h 30m

Troubleshooting errors

The below troubleshooting steps will helps to resolve the respective issues.

DV001

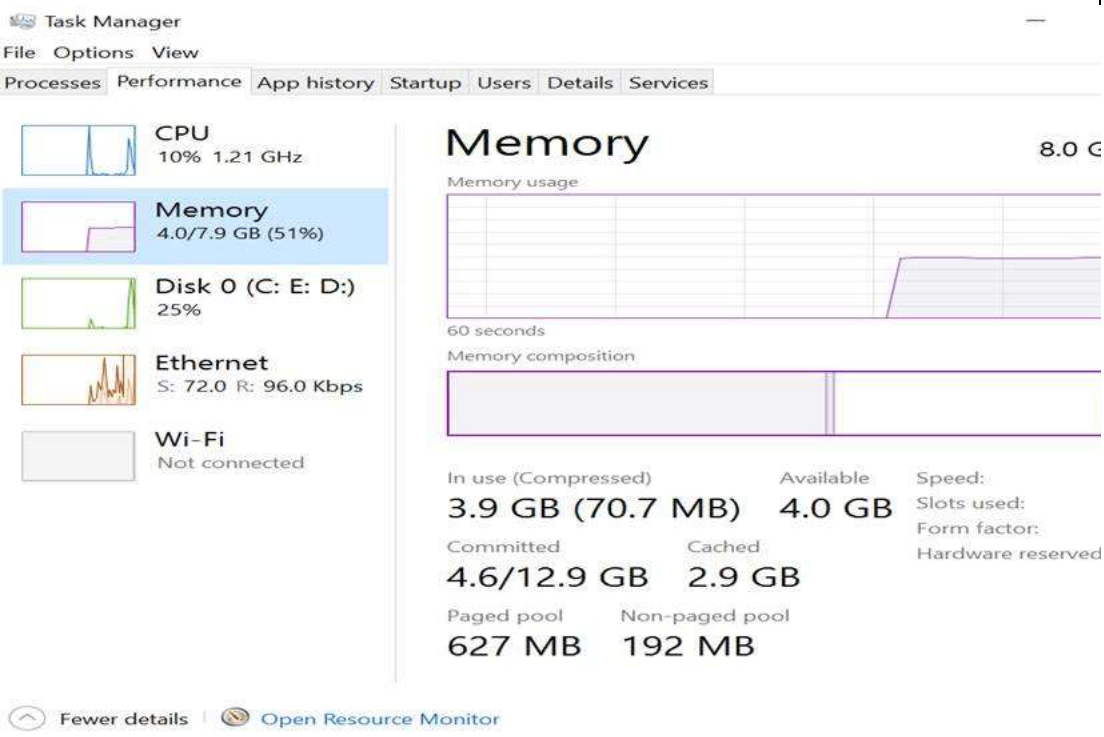
CODE	DV001
TEXT	Unable to load the Dashboard since it was created in a higher version of Dashboard Designer (Version: X.X.X.X)
DESCRIPTION	The dashboard should have been created in a higher version of Dashboard Designer (For example, v2.1.0.250) than that of dashboard service through which it is currently running (For example, v2.1.0.232).
SOLUTION	• Ensure the version of dashboard designer and that of the deployed dashboard service are same. • If the dashboard is going to be embedded in an application, the dashboard service version will be the version of the Dashboard Platform SDK.

	• If the dashboard is going to be hosted in a Dashboard Server, the dashboard service version will be the version of the Dashboard Server. • If the dashboard file belongs to the version higher than the targeted deployment environment, recreate the dashboard through the respective versioned Dashboard Designer.

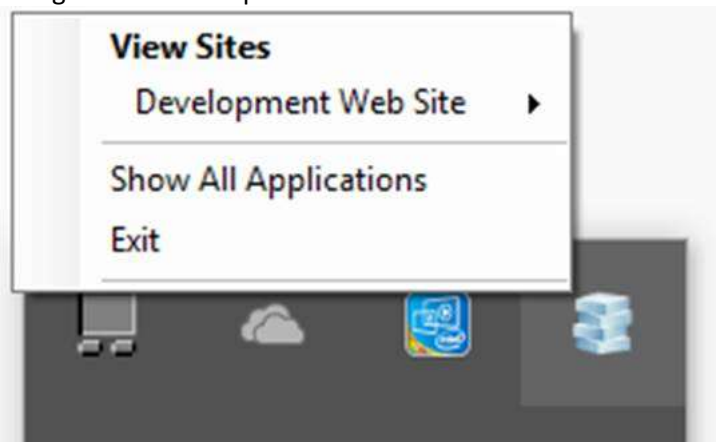
DV002

CODE	DV002
TEXT	The Dashboard Service could not be contacted. There may be a temporary glitch or the server may be down.
DESCRIPTION	Dashboard Service is not reachable either due to connection failure or long awaited response or it is being used by some other process.
SOLUTION	• Make sure that the Dashboard Service is running in the machine where the dashboard exists. You can test the service URL in the browser of the respective machine whose format will look like below. http://domain:port/DashboardService.svc (or) https://domain:port/DashboardService.svc 

- Check System Memory usage and make sure, it has at least a minimum of 5% free space for use.



- Restart the IIS or IIS Express where the dashboard service is hosted.
- In case of previewing from Dashboard Designer, stop the IIS Express and click Preview in Designer Toolbar to preview the dashboard.



- If you hosted the Dashboard Service as a sub application. Refer to the user guide and make sure that Dashboard Service is running properly.
- If you have installed Dashboard Platform SDK setup in remote machine and referred the Dashboard Service of that remote machine, then, the Dashboard Service should be accessible in your application, where the Dashboard Viewer is embedded.
- If the Dashboard Service is running as a Windows Service Background Process, then this Dashboard Service can be accessible only at local machine. To access the Dashboard Service

	in remote machine, host the Dashboard Service in IIS. Refer here how to host the Dashboard Service in IIS. • Follow the link to troubleshoot this issue for the dashboard in the Dashboard Server. • Make sure you are using the Syncfusion Dashboard Platform SDK's service available in the location %localappdata%\Syncfusion\Dashboard Platform SDK\Service not the Dashboard Server's Service

DV003

CODE	DV003
TEXT	Dashboard could not be rendered since the service URL is not set.
DESCRIPTION	Dashboard will not render when either the service URL is not set or the given URL is incorrect.
SOLUTION	• Make sure that the service URL was specified. For example, {% highlight js %} {% endhighlight %} • Ensure the given URL is in appropriate format. For example,http://domain:port/DashboardService.svc (or) https://domain:port/DashboardService.svc

DV004

CODE	DV004
TEXT	Dashboard could not be rendered since the dashboard path is not set.
DESCRIPTION	Dashboard will not render when either the Dashboard path is not set or the given path is incorrect.
SOLUTION	• Make sure that the Dashboard path was specified. • Ensure the given path is correct. Also, the dashboard path (either local or remote file location) should be accessible from Dashboard Service.For Example, {% highlight js %} {% endhighlight %}

DV005

CODE	DV005
TEXT	Access to the service is denied to the current user.
DESCRIPTION	Current user is not authorized to access the respective dashboard.
SOLUTION	• Ensure the user has permission to view the dashboard. If not, get view permission from the server administrator. For more information, refer to the below link,https://help.syncfusion.com/dashboard-platform/dashboard-server/administration/manage-permissions

DV006

CODE	DV006
TEXT	The requested dashboard file was not found. It may be either moved or renamed.
DESCRIPTION	The loaded dashboard file may either be not available in the specified path or the file name is incorrect or the loaded file itself is invalid.
SOLUTION	• Ensure the dashboard file is available in the specified dashboard path. • Ensure the name of the dashboard file specified is appropriate. • Ensure the loaded file has valid file extension (*.SYDX). • If the dashboard path is the hosted URL, ensure that the URL is downloading the dashboard file by pasting it in the browser • If the dashboard file is hosted in IIS, make sure to enable the MIME type to make the file downloadable. To know how to enable the MIME type, follow the below link:https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc725608(v=ws.10)

DV007

CODE	DV007
TEXT	The requested file or assembly was not found. It may be either moved or renamed.
DESCRIPTION	Any of the assembly related to Dashboard Service is missing at installation location.
SOLUTION	• If you are previewing dashboard from Dashboard Designer, check whether any assembly is missing in the installed location (%ProgramData%\Syncfusion\DashboardDesigner\DashboardDesigner*Version**IISExpress*DashboardService\bin) or reinstall the Dashboard Designer application. • If you are loading dashboard from SDK sample, check whether any assembly is missing in the installed location (%localappdata%\Syncfusion\Dashboard Platform SDK\Service\bin) or reinstall the Dashboard Platform SDK application. • If you are loading dashboard from Dashboard Server, check whether any assembly is missing in the installed location (C:\Syncfusion\Dashboard Server\DashboardServer.Web\DashboardService\bin) or reinstall the Dashboard Server application. List of assemblies in the installed location • Kent.Boogaart.HelperTrinity.dll • Kent.Boogaart.KBCsv.dll • Kent.Boogaart.KBCsv.Extensions.Data.dll

	• Kent.Boogaart.KBCsv.Extensions.dll • Microsoft.AnalysisServices.AdomdClient.dll • Microsoft.AnalysisServices.dll • Microsoft.Data.Edm.dll • Microsoft.Data.OData.dll • Microsoft.WindowsAzure.Storage.dll • Microsoft.WindowsAzure.StorageClient.dll • Newtonsoft.Json.dll • Npgsql.dll • Syncfusion.Compression.Base.dll • Syncfusion.Dashboard.Base.dll • Syncfusion.Dashboard.Encryption.dll • Syncfusion.Dashboard.ExportWrapper.dll • Syncfusion.DashboardService.dll • Syncfusion.Server.Base.Encryption.dll • Syncfusion.ThriftHive.Base.dll • Syncfusion.XlsIO.Base.dll • System.Data.SQLite.dll • System.Data.SQLite.Linq.dll • System.Spatial.dll • Thrift.dll

DV008

CODE	DV008
TEXT	Dashboard could not be viewed due to an error.
DESCRIPTION	An internal error occurred while processing the loaded dashboard file.
SOLUTION	Check the error log file(s) generated in the below mentioned location: • Loading Dashboard from Designer: %ProgramData%\Syncfusion\DashboardDesigner\DashboardDesigner*Version**IISExpress*DashboardService\ErrorLog • Loading Dashboard from Server:C:\Syncfusion\Dashboard Server\DashboardServer.Web\API\ErrorLog

	• Loading Dashboard from SDK:%localappdata%\Syncfusion\Dashboard Platform SDK\Service\ErrorLog • Please contact the Syncfusion Support Team with the error log. To create the support incidents please follow the link below:https://www.syncfusion.com/support/directtrac/incidents

DV009

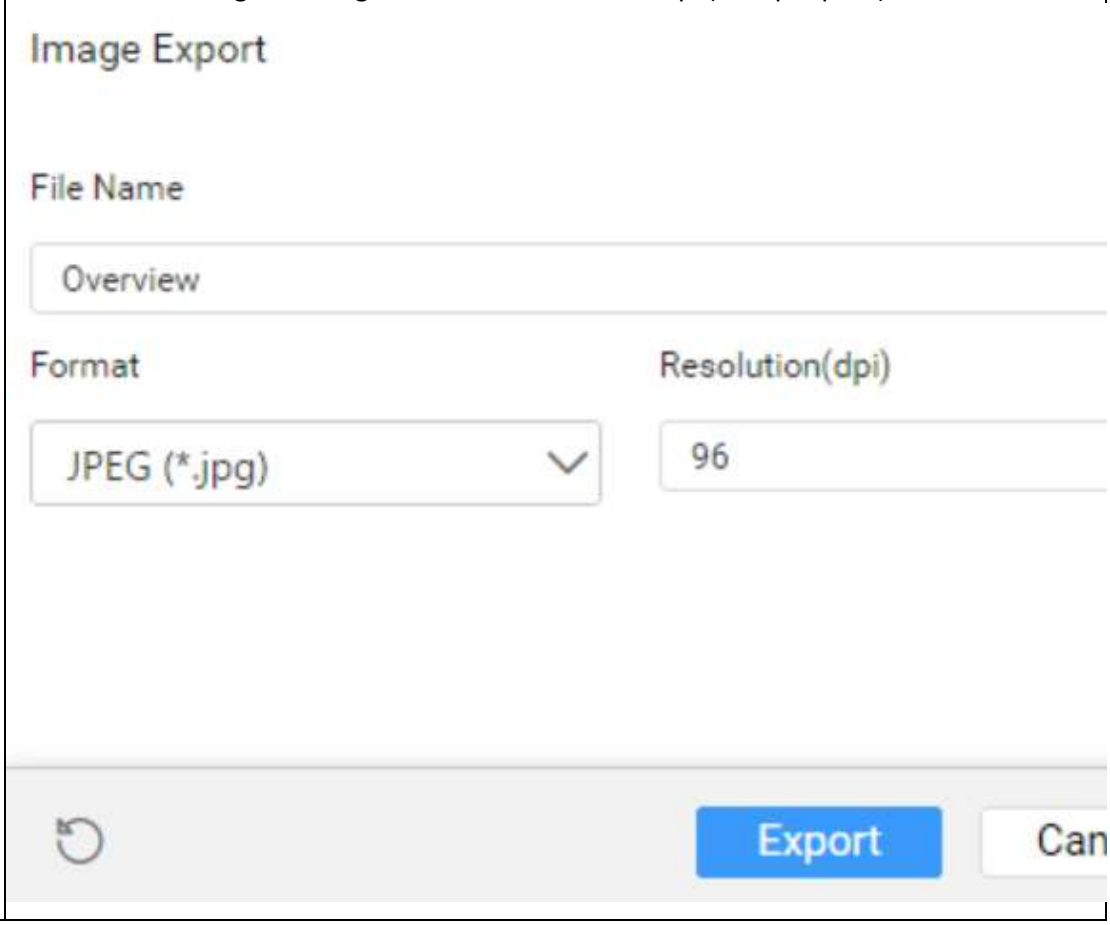
CODE	DV009
TEXT	The loaded file is not a valid dashboard file.
DESCRIPTION	The loaded dashboard file may either be corrupted or invalid.
SOLUTION	• Ensure the file specified in the dashboard path is valid and has valid extension (*.SYDX) • Make sure the presence of the dashboard file on the location/folder you have specified. • Make sure that the dashboard path you have specified is under the location/domain where the hosted server pointing to.

DV010

CODE	DV010
TEXT	The dashboard could not be viewed since we were unable to validate the current user authorization.
DESCRIPTION	The user is trying to access the dashboard without logging into the Dashboard Server.
SOLUTION	Ensure the user logged into the Dashboard Server.

DV011

CODE	DV011
TEXT	The exported image file could not be downloaded due to large file size.
DESCRIPTION	The image resolution is higher than the system capacity. Large resolution leads to larger file size.

SOLUTION	Recommended range for Image resolution is 96 to 300 dpi (dots per pixel). 

Card Widget Rendering Issue

If the card widget is not rendered properly in the SDK application after updating the Dashboard Platform SDK to version 3.1, follow the given steps to resolve the issue.

- Remove the old CSS file (ej.widgets.core.min.css) referred in the application
- Add DOCTYPE html html tag in the main HTML page of the SDK application

ejDashboardViewer

Dashboard Viewer allows you to visualize the dashboard that was published in Dashboard Server or deployed in local physical path, opened through Dashboard Designer and run, navigating to specific URL in web browser. Certain options to widgets that have been enabled during design time like Maximize button, CSV, Excel & Image export options can be availed at run time through Dashboard Viewer. Also, you can export entire dashboard itself to image.

Syntax

JS

```
$(element).ejDashboardViewer()
```


Example

JS

```
<div id="container"></div>
<script>
// Create Dashboard
$('#container').ejDashboardViewer();
</script>
```

Requires

- module:jquery-1.10.2.min.js
- module:jquery.easing.1.3.min.js
- module:ej.dashboardViewer.all.min.js

Members

accessToken `string`

It holds the user token generated at Dashboard Server and it is used for embedding the dashboard which is available in Dashboard Server.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
accessToken:""
});
```

allowComment `boolean`

Adds a comment icon to Dashboard and also the Widgets in the Dashboard.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
allowComment:true
});
```

autoRefreshSettings `object`

The auto refresh allows you to configure the scheduled refreshing process of the dashboard. It is used to refresh the data, based on the specified time. Either the whole dashboard or specific widgets in the dashboard can be refreshed automatically.

autoRefreshSettings.tabName `string`

Specifies to refresh the given name of the dashboard.

Default value

null

Example

The following code snippet illustrates auto refreshing of tab name in multi-tabbed dashboard:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabName:"Worldwide Car Sales Random data"  // set dashboard name
}]
});
```

autoRefreshSettings.tabIndex `number`

Specifies to refresh the given tab index of the dashboard, and the default value 0 indicates the first tab of the dashboard.

Default value

0

Example

The following code snippet illustrates auto refreshing of tab index item in multi-tabbed dashboard:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabIndex:1  // set dashboard tab index value, 1 indicates the second tab
dashboard
}]
});
```

autoRefreshSettings.intervalSettings `object`

Specifies the time interval based on the mode type for auto refreshing of data. You can set the time interval settings by using the mode type("Hourly, Daily, Weekly, Monthly, and Yearly") and set the corresponding schedule settings depends on the mode type.

For hourlySchedule API : The dashboard or widget will be refreshed in the specified time interval in number of hours, minutes, and seconds. You can set the time interval value maximum of hours-99, minutes-999 and seconds-9999.

For DailySchedule API : The dashboard or widget will be refreshed in the specified number of days. If you specify 2 days once to refresh the dashboard or widget, then the data will be refreshed on 12PM at end of the total days.

For weeklySchedule API : The dashboard or widget will be refreshed in the given week days in the specified number of weeks. Week days value is arranged in this specific order [Sunday-Saturday ---> 1-7] to refresh the data.

For monthlySchedule API : The dashboard or widget will be refreshed in the given day of the specified number of months. If you need to refresh the dashboard or widgets in the last day of [any] month, then set the day value as 32. The month value is arranged in this specific order [January-December ---> 1-12] to refresh the data.

For yearlySchedule API: The dashboard or widget will be refreshed in the given day of the month in the specified number of years.

Default value

{}

Example

This example illustrates auto refresh interval settings for without multi tabbed dashboard:

Code snippet for hourlySchedule API interval settings:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
intervalSettings: {
mode: "Hourly",
hourlySchedule: {
hours: 0,    // Default value is 0
minutes: 0, // Default value is 0
seconds: 30 // Default value is 30
}
}
}
});
```

Code snippet for dailySchedule API interval settings:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
intervalSettings: {
mode: "Daily",
dailySchedule: {
days: 1      // Default value is 1
}
}
}
});
```

Code snippet for weeklySchedule API interval settings:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
intervalSettings: {
mode: "Weekly",
weeklySchedule: {
weeks: 1,     // Default value is 1
weekDay: [1] // Default value is [1]
}
}
```



```
}
});
```

Code snippet for monthlySchedule API interval settings:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
intervalSettings: {
mode: "Monthly",
monthlySchedule: {
day: 1,      // Default value is 1
months: 1    // Default value is 1
}
}
}
});
```

Code snippet for yearlySchedule API interval settings:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
intervalSettings: {
mode: "Yearly",
yearlySchedule: {
years: 1,    // Default value is 1
day: 1,      // Default value is 1
month: 1     // Default value is 1
}
}
}
});
```

This example illustrates auto refresh interval settings for multi-tabbed dashboard:

Code snippet for hourlySchedule, dailySchedule, weeklySchedule, monthlySchedule, and yearlySchedule API interval settings:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabName:Worldwide Car Sales Random data,
tabIndex:0,
intervalSettings: {
mode: "Hourly",
hourlySchedule: {
hours: 0,    // Default value is 0
minutes: 0, // Default value is 0
seconds: 30 // Default value is 30
```



```
}
}
},
{
tabName:Worldwide Car Sales Random data 1,
tabIndex:1,
intervalSettings: {
mode: "Daily",
dailySchedule: {
days: 1      // Default value is 1
}
}
},
{
tabName:Worldwide Car Sales Random data 2,
tabIndex:2,
intervalSettings: {
mode: "Weekly",
weeklySchedule: {
weeks: 1,     // Default value is 1
weekDay: [1] // Default value is [1]
}
}
},
{
tabName:Worldwide Car Sales Random data 3,
tabIndex:3,
intervalSettings: {
mode: "Monthly",
monthlySchedule: {
day: 1,      // Default value is 1
months: 1    // Default value is 1
}
}
},
{
tabName:Worldwide Car Sales Random data 4,
tabIndex:4,
intervalSettings: {
mode: "Yearly",
yearlySchedule: {
years: 1,    // Default value is 1
day: 1,      // Default value is 1
month: 1     // Default value is 1
}
}
}]
});
```

autoRefreshSettings.isWidgetSpecific `boolean`

Specifies to refresh the data of specific widgets alone in the viewer.

Default value

false

Example

Code snippet for auto refresh widget specific item in the without multi-tabbed dashboard.

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{isWidgetSpecific:true}
});
```

Code snippet for auto refresh widget specific item in the multi-tabbed dashboard.

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabName:Worldwide Car Sales Random data,
tabIndex:0,       // 0 indicates the first tab dashboard
isWidgetSpecific:true
}]
});
```

autoRefreshSettings.showWaitingIndicator `boolean`

Specifies to show the waiting indicator when the data is refreshed.

Default value

false

Example

The following code snippet for auto refresh shows waiting indicator item in the without multi-tabbed dashboard:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{showWaitingIndicator:true}
});
```

The following code snippet for auto refresh shows waiting indicator item in the multi-tabbed dashboard:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabName:Worldwide Car Sales Random data,
tabIndex:0,       // 0 indicates the first tab dashboard
showWaitingIndicator:true
}]
});
```


autoRefreshSettings.trackData `boolean`

Specifies whether to trigger refreshing process of data in the dashboard viewer when there is any modifications of data in the data source table.

Default value

false

Example

The following code snippet illustrates auto refreshing of track data item in without multi-tabbed dashboard:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{trackData:true}
});
```

The following code snippet illustrates auto refreshing of track data item in multi-tabbed dashboard:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabName:Worldwide Car Sales Random data,
tabIndex:0,       // 0 indicates the first tab dashboard
trackData:true
}]
});
```

autoRefreshSettings.widgets `array`

Specifies the widget collection that will refresh the data automatically. Set the id as dashboard widget's Guid.

For hourlySchedule API : The dashboard or widget will be refreshed in the specified time interval in number of hours, minutes, and seconds. You can set the time interval value maximum of hours-99, minutes-999, and seconds-9999.

For DailySchedule API : The dashboard or widget will be refreshed in the specified number of days. If you set number of days as 2 to refresh the dashboard or widget, then the data will be refreshed on 12PM at end of the total days.

For weeklySchedule API : The dashboard or widget will be refreshed in the given week days in the specified number of weeks. Week days value is arranged in this specific order [Sunday-Saturday ---> 1-7] to refresh the data.

For monthlySchedule API : The dashboard or widget will be refreshed in the given day of the specified number of months. If you need to refresh the dashboard or widgets in the last day of [any]month, then set the day value as 32. The month value is arranged in this specific order [January-December ---> 1-12] to refresh the data.

For yearlySchedule API: The dashboard or widget will be refreshed in the given day of the month in the specified number of years.

Default value

[]

Example

Example for auto refresh interval settings in the without multi tabbed dashboard:

To get the widget's Guid, refer to [getCurrentDashboardDetails]

Code snippet for hourlySchedule API interval settings for widget specific data refresh.

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
isWidgetSpecific:true,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",       // set the Guid of the
dashboard widget
intervalSettings:{
mode: "Hourly",
hourlySchedule: {
hours: 0,   // Default value is 0
minutes: 0, // Default value is 0
seconds: 30 // Default value is 30
}
}
}]
}
});
```

Code snippet for dailySchedule API interval settings for widget specific data refresh.

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
isWidgetSpecific:true,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",       // set the Guid of the
dashboard widget
intervalSettings:{
mode:"Daily",
dailySchedule: {
days: 1 // Default value is 1
}
}
}]
}
});
```

Code snippet for weeklySchedule API interval settings for widget specific data refresh.

JS

```
$("#container").ejDashboardViewer({
```



```
enableAutoRefresh:true,
autoRefreshSettings:{
isWidgetSpecific:true,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",          // set the Guid of the
dashboard widget
intervalSettings:{
mode:"Weekly",
weeklySchedule: {
weeks: 1,        // Default value is 1
weekDay: [1]     // Default value is [1]
}
}
}]
}
});
```

Code snippet for monthlySchedule API interval settings for widget specific data refresh.

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
isWidgetSpecific:true,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",            // set the Guid of the
dashboard widget
intervalSettings:{
mode:"Monthly",
monthlySchedule: {
day: 1,      // Default value is 1
months: 1    // Default value is 1
}
}
}]
}
});
```

Code snippet for yearlySchedule API interval settings for widget specific data refresh.

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:{
isWidgetSpecific:true,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",          // set the Guid of the
dashboard widget
intervalSettings:{
mode: "Yearly",
yearlySchedule: {
years: 1,    // Default value is 1
day: 1,      // Default value is 1
month: 1     // Default value is 1
```



```
}
}
}]
}
});
```

Example for auto refresh interval settings in multi tabbed dashboard:

The following code snippet illustrates hourlySchedule, dailySchedule, weeklySchedule, monthlySchedule, and yearlySchedule API interval settings:

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true,
autoRefreshSettings:[{
tabName:Worldwide Car Sales Random data,
tabIndex:0,
widgets:[{
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",       // set the Guid of the
dashboard widget
intervalSettings:{
mode: "Hourly",
hourlySchedule: {
hours: 0,    // Default value is 0
minutes: 0, // Default value is 0
seconds: 30 // Default value is 30
}
}
},
{
id:"937d0fa9-0eb1-4b43-8626-d4b8e0f5e625",      // set the Guid of the
dashboard widget
intervalSettings:{
mode: "Daily",
dailySchedule: {
days: 1     // Default value is 1
}
}
},
{
id:"878d0fa9-0eb1-4b43-8626-d4b8e0f5e625",       // set the Guid of the
dashboard widget
intervalSettings:{
mode: "Weekly",
weeklySchedule: {
weeks: 1,     // Default value is 1
weekDay: [1] // Default value is [1]
}
}
}]
},
{
tabName:Worldwide Car Sales Random data 1,
tabIndex:1,
widgets:[{
```

```
id:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625",       // set the Guid of the
dashboard widget
intervalSettings:{
mode: "Monthly",
monthlySchedule: {
day: 1,      // Default value is 1
months: 1    // Default value is 1
}
}
},
{
id:"932d0fa9-0eb1-4b43-8626-d4b8e0f5e625",       // set the Guid of the
dashboard widget
intervalSettings:{
mode: "Yearly",
yearlySchedule: {
years: 1,    // Default value is 1
day: 1,      // Default value is 1
month: 1     // Default value is 1
}
}
}]
}]
});
```

cacheSettings `object`

Contains the settings for customization of client side caching.

cacheSettings.capacity `number`

Specifies the number of requests to be cached. If the request count exceeds the capacity old cached items will be replaced with new one's.

Default value

10

Example

JS

```
$("#container").ejDashboardViewer({
enableClientSideCache:true,
cacheSettings:{capacity:50}
});
```

cacheSettings.cleanupInterval `number`

Specifies the interval in minutes

Default value

5

Example

JS

```
$("#container").ejDashboardViewer({
enableClientSideCache:true,
cacheSettings:{cleanupInterval:20}
});
```


cacheSettings.enableAutoCleanup `boolean`

Specifies whether to automatically clear the client side cached data in specific intervals.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
enableClientSideCache:true
cacheSettings:{enableAutoCleanup:false}
});
```

canLoadDataOffline `boolean`

Specifies whether to load the offline data.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
canLoadDataOffline:false
});
```

commentSettings `object`

Comment settings allows you to enable commenting for the Dashboard and/or its individual widgets. You can add the comment for dashboard and each widgets which is available in the dashboard. The user can add more information about the dashboard and the widgets.

commentSettings.isDashboardCommented `boolean`

Used to set the comment icon for the dashboard. If it is commented, then the information will be available by clicking the comment icon in the Dashboard Viewer.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
commentSettings:{isDashboardCommented:true}
});
```

commentSettings.isWidgetCommented `boolean`

Used to set the comment icon for the widgets in the dashboard. If it is commented,then the information will be shown by clicking the comment icon.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
commentSettings:{isWidgetCommented:true}
});
```

commentSettings.widgets `array`

List of widgets collection will be available in the array (widget id). Commented widget list will be available in the array with widget id.

Default value

[]

Example

JS

```
$("#container").ejDashboardViewer({
enableComment:true,
commentSettings:{ isWidgetCommented:true, widgets:["Chart_1", "Grid_1"]}
});
```

cssClass `string`

Sets the root CSS class for Dashboard Viewer theme, which is used for customizing the styling of Dashboard Viewer.

Default value

””

Example

JS

```
$("#container").ejDashboardViewer({
cssClass: "e-CustomTheme"
});
```

customThemeSettings `string`

Used to customize the colors and fonts of dashboard based on given settings.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
enableCustomTheming:true,
customThemeSettings:ej.dashboardViewer.customTheme
});
```

dashboardData `string`

Sets the layout information and all the widget data present in the current dashboard in the form of string.

To get dashboard data value, refer to getDashboardData

Default value

null

Example

JS

```
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
$("#container").ejDashboardViewer({
dashboardData: dashboardObj.getDashboardData() // Get the complete
information of the currently rendered dashboard
});
```

dashboardParameterSettings `object`

dashboardParametersSettings API is used to update the dashboard parameter collection details in viewer. Using this we can modify the design time dashboard parameter settings in viewer.

dashboardParameterSettings.data `array`

List of dashboard parameter data collection will be available in the array (data). Dashboard Parameter list will be available in the array with data.

Default value

[]

Example

JS

```
$("#dashboard").ejDashboardViewer(
{
dashboardParameterSettings: {
"showIcon":true, // to show or hide the dashboard parameter icon
"data":[{
"parameterName":"CustomerId",
"showInParameterDialog":true,
"showInPromptDialog":false,
"value":"TRADH",
}]
},
}
);
```

dashboardParameterSettings.data.value

array

The value holds single or collection of parameter values, which replaces the default value collection in the parameter.

Default value

[]

Example

JS

```
$("#dashboard").ejDashboardViewer(
{
```



```
dashboardParameterSettings: {
"showIcon":true, // to show or hide the dashboard parameter icon
"data":[{
"parameterName":"CustomerId",
"showInParameterDialog":true,
"showInPromptDialog":false,
"value":["TRADH", "HANAR", "FALKO"],
}]
},
}
);
```

string

The value contains the relative date values as input to the parameter to filter the data with date range.

Default value

""

Example

JS

```
$("#dashboard").ejDashboardViewer(
{
dashboardParameterSettings: {
"showIcon":true, // to show or hide the dashboard parameter icon
"data":[{
"parameterName":"OrderDate",
"showInParameterDialog":true,
"showInPromptDialog":false,
"value":"Last 1 Week(s)",
}]
},
}
);
```

For example, if you configure Last 1 Week as filter, you will get the last 1 week data in that widget. After a week, it will show you the previous 1 week data from the date you are viewing. Through this, you will no longer required to set the filter statically and change week by week to see the previous week results.

The following are the different possibilities and use case on relative date values in the dashboard parameter value:

Range	Type	Examples
Current Previous Succeeding Last Next Past Upcoming	Year(s) Quarter(s) Month(s) Week(s) Day(s) Hour(s)	Current Year(s), Previous Month(s), Succeeding Week(s), Last 3 Quarter(s), Next 1 Year(s), Past 2 Week(s), Upcoming 1 Week, Tomorrow, Today, Yesterday

Year to date	-	Year to date
Quarter to date	-	Quarter to date
Month to date	-	Month to date
Week to date	-	Week to date

object

The value also holds the date values as input by representing the start and end dates.

Default value

{}

dashboardParameterSettings.data.value.start `string`

Start date holds the date value in the yyyy/MM/dd format from which the parameter can filter the data with respect to the date-time column assigned to, via the stored procedure data.

Default value

""

dashboardParameterSettings.data.value.end `string`

End date holds the date value in the yyyy/MM/dd format from which the parameter can filter the data with respect to the date-time column assigned to, via stored procedure data.

Default value

""

Example

JS

```
$("#dashboard").ejDashboardViewer(
{
dashboardParameterSettings: {
"showIcon":true, // to show or hide the dashboard parameter icon
"data":[{
"parameterName":"OrderDate",
"showInParameterDialog":true,
"showInPromptDialog":false,
"value": {"start" : "1996/09/21", "end" : "2018/02/13"},  // DateTime string
should be in the format yyyy/MM/dd
}]
},
}
);
```

dashboardParameterSettings.showIcon `boolean`

Used to set the dashboard parameter icon for the dashboard. If icon set in visible state, the dashboard parameter data information will be available in dashboard parameter dialog by clicking the dashboard parameter icon in the Dashboard Viewer.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
dashboardParameterSettings:{showIcon:false}
});
```

dashboardPath `string`

Sets the Dashboard file path (SYDX). This API can hold absolute path of SYDX file in disk or the URL of SYDX file hosted in a web server (This link should return the SYDX file as stream)

Default value

””

Example

JS

```
$("#container").ejDashboardViewer({
dashboardPath: "C:\\Documents\\Sample Dashboard.sydx"
});
```

enableAutoRefresh `boolean`

Enables the refreshing of dashboard data on specific intervals automatically.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
enableAutoRefresh:true
});
```

enableClientSideCache `boolean`

Sets the caching of interaction data of Dashboard Viewer in browser.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
enableClientSideCache:true
});
```

enableCustomTheming `boolean`

Specifies whether to apply the custom theme to be applied to the dashboard.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
enableCustomTheming:true
});
```

enableDashboardScaling `boolean`

Specifies whether to render dashboard based on the scaling factor (resolution).

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
enableDashboardScaling:true
});
```

enableFilterOverview `boolean`

Specifies whether to show the filter overview in Dashboard Viewer

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
enableFilterOverview:false
});
```

enableServerSideCache `boolean`

Sets the caching of data in Dashboard Service.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
enableServerSideCache:true
});
```

enableViewData `boolean`

Specifies enable or disable the view data support in Dashboard Viewer.

Default value

 true

Example

JS


```
$("#container").ejDashboardViewer({
enableViewData: false
});
```

enableWidgetMode `boolean`

Specifies enable or disable the widget mode settings in Dashboard viewer.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
enableWidgetMode:false
});
```

favoriteSettings `object`

Specifies the favorite settings of the dashboard viewer.

favoriteSettings.enabled `boolean`

Specifies whether to enable or disable the favorite icon in favorite settings.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
favoriteSettings:{enabled:false}
});
```

favoriteSettings.isFavorite `boolean`

Specifies whether to enable or disable the favorite icon in favorite settings.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
favoriteSettings:{isFavorite:false}
});
```

filterOverviewSettings `object`

The filter overview shows the filter's applied in the current view. using the filter over view you can clear the applied filters one after the other, or you may clear all the filters.

filterOverviewSettings.showSaveAsIcon `boolean`

Specifies whether to Show or hide the save as icon for saving the applied filter as view.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
filterOverviewSettings:{showSaveAsIcon:false}
});
```

filterOverviewSettings.showSaveIcon `boolean`

Specifies whether to Show or hide the save icon for saving the applied filter as view.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
filterOverviewSettings:{showSaveIcon:false}
});
```

filterOverviewSettings.showViewSavedFilterIcon `boolean`

Enables or disables the filter overview View Saved filter icon in Dashboard Viewer.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
filterOverviewSettings:{showViewSavedFilterIcon:false}
});
```

filterOverviewSettings.viewId `string`

Contains Id attribute of the saved filter view, used to maintain this as a unique key for working in back end.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
filterOverviewSettings:{viewId:""}
});
```

filterOverviewSettings.viewName `string`

Name of the saved filter view, this name will be displayed as title for the Applied filters list.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
filterOverviewSettings:{viewName:""}
});
```

filterPanelSettings `object`

Filter panel settings allows you to apply filters in the dashboard viewer.

filterPanelSettings.state `string`

Specifies to show or hide the filter panel when the state is expanded or collapsed. The size of the dashboard will adjust on its own when the filter panel gets opened/hidden in the viewer.

Default value

collapsed

Example

The following code snippet illustrates expanded state of filter panel while rendering the dashboard:

JS

```
$("#container").ejDashboardViewer({
filterPanelSettings:{state:expanded}    // expanded - show the filter panel
while rendering the dashboard
});
```

filterPanelSettings.filterPanelId `string`

Contains the id attribute of the external div, where the filter panel needs to be rendered.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
filterPanelSettings:{filterPanelId:""}
});
```

filterPanelSettings.showCloseIcon `boolean`

Enables or disables the filter panel close icon in the dashboard viewer.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
filterPanelSettings:{showCloseIcon:true}
});
```


filterPanelSettings.showHeader `boolean`

Enables or disables the filter panel header icon in the dashboard viewer.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
filterPanelSettings:{showHeader:true}
});
```

filterPanelSettings.showIcon `boolean`

Specifies whether to show or hide the filter panel toggle icon in the dashboard viewer.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
filterPanelSettings:{showIcon:true}
});
```

filterParameters `string`

Specifies the filter that has to be applied to the dashboard.

Default value

null

Example

JS

```
$("#container").ejDashboardViewer({
filterParameters: "Continent=Asia,Africa,Europe"
});
```

interactionSettings `object`

Interaction settings contain history maintenance property to enable or disable history maintenance in the dashboard viewer.

interactionSettings.multiSelectionInterval `number`

The "multiSelectionInterval" property is used to set the time interval for performing multi selection in widgets filter of the dashboard viewer. The default value of this property is 500 milliseconds.

Default value

500

Example

JS


```
$("#container").ejDashboardViewer({
interactionSettings:{multiSelectionInterval:500} // Default values is 500
});
```

interactionSettings.allowHistoryMaintenance `boolean`

This property Used to store the current state of the dashboard in browser session storage.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
interactionSettings:{allowHistoryMaintenance:false}
});
```

interactionSettings.handleHistoryEvent `boolean`

This property Used to restrict the browser back and forth button action.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
interactionSettings:{handleHistoryEvent:false}
});
```

localeSettings `object`

Specifies the locale of the dashboard viewer.

localeSettings.culture `string`

Specifies the language to the applied to the dashboard viewer.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
localeSettings:{culture:"en-US"}
});
```

localeSettings.dateFormat `string`

It is used to convert the date value to given date format in dashboard viewer.

Default value

""

Example

JS


```
$("#container").ejDashboardViewer({
localeSettings:{dateFormat:"yyyy/MM/DD"}
});
```

localeSettings.resourcePath `string`

Specifies the path of the resource file present in the server or else URL path of the xml resource file.

Default value

""

Example

The following code snippet illustrates to specifies the localization server path:

JS

```
$("#container").ejDashboardViewer({
localeSettings:{resourcePath:"D:\Dashboard\Trunk\Samples\Common\Resources\"}
});
```

The following code snippet illustrates to specifies the localization xml resource URL path:

JS

```
$("#container").ejDashboardViewer({
localeSettings:{resourcePath:"https://dashboardsdk.syncfusion.com//Resources
//Resource.en-US.xml"}
});
```

localeSettings.timeFormat `string`

It is used to convert the date value to given time format in dashboard viewer.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
localeSettings:{timeFormat:"HH:MM:SS"}
});
```

responsiveSettings `object`

Specifies the responsiveness of the dashboard viewer.

responsiveSettings.showExport `boolean`

Specifies whether to show or hide the Export icon in responsive settings.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
responsiveSettings:{showExport:false}
```



```
});
```

responsiveSettings.showMaximize `boolean`

Specifies whether to show or hide the maximize icon in responsive settings.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
responsiveSettings:{showMaximize:false}
});
```

scalingFactor `nullable<int>`

Sets the scaling factor, which helps to render dashboard based on the resolution given, only if enableDashboardScaling sets to true.

Default Value: null

Example

JS

```
$("#container").ejDashboardViewer({
scalingFactor: null
});
```

selectedTabGuid `string`

Sets the guid that indicates the selected Guid dashboard while rendering multiple dashboards in viewer.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
selectedTabGuid:""
});
```

selectedTabIndex `number`

Sets a index value that indicates the selected index dashboard with the given index position while rendering multiple dashboards.

Default Value: 0

Example

JS

```
$("#container").ejDashboardViewer({
selectedTabIndex:0
});
```


selectedTabName `string`

Sets the dashboard name that indicates the selected tab with the given selected dashboard while rendering multiple dashboards in viewer.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
selectedTabName:""
});
```

serviceUrl `string`

Sets the URL of the Dashboard Service that will be used for processing the viewed Dashboard file.

Default value

The default value of the URL is Empty.

””

Example

JS

```
$("#container").ejDashboardViewer({
serviceUrl:"http://localhost:[port]/service.svc"
});
```

showGetLinkIcon `boolean`

Specifies whether to show or hide the Get Link icon in viewer.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
showGetLinkIcon:true
});
```

showTab `boolean`

Specifies whether to show or hide the tabs when using multiple dashboards in viewer.

Default value

true

Example

JS

```
$("#container").ejDashboardViewer({
showTab:true
});
```


size `object`

Sets the height and width of the Dashboard Viewer.

size.height `string`

sets the Height of the Dashboard Viewer in % or in pixels.

Default Value

””

Example

HTML

```
$("#dashboard").ejDashboardViewer({size{height:'800'}});
```

size.width `string`

sets the Width of the Dashboard Viewer in % or in pixels.

Default Value

””

Example

HTML

```
$("#dashboard").ejDashboardViewer({size{width:'800'}});
```

tabDetails `array`

Sets the tab names to be updated in the list of tab details(tabId, displayName) of a dashboard in an array.

Default value

[]

Example

JS

```
$("#container").ejDashboardViewer({
tabDetails:[ { tabId: "", displayName: "" } ]
});
```

updatePageTitle `boolean`

This enables the automatic updation of page title based on the filter applied in dashboard.

Default value

false

Example

JS

```
$("#container").ejDashboardViewer({
updatePageTitle:true
});
```


widgetActionSettings `array`

Widget Action Settings contains the widget guid and collection of action settings value like type and array of action data to perform the navigation, master interaction and drilldown action in the dashboard viewer.

widgetActionSettings.widgetId `string`

Specifies the widget Guid.

Default value

""

Example

To get the widget's Guid, refer to getCurrentDashboardDetails

JS

```
$("#container").ejDashboardViewer({
widgetActionSettings:[{
widgetId:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625"        // set the Guid of
the dashboard widget
}]
});
```

widgetActionSettings.actionSettings `array`

Specifies the collection of action type and actions list to do customize the widget action for the given widgets.

widgetActionSettings.actionSettings.type `string`

Specifies the type of mouse action type like "Click", "DoubleClick", RightClick act as "ContextMenu".

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
widgetActionSettings:[{
actionSettings:[{
[type:"Click"]
}]
}]
});
```

widgetActionSettings.actionSettings.actions `array`

Specifies the action / list of actions to be shown i.e. Filter / Linking / DrillDown. If more than one item is given then instead of triggering the action, to set the actionSettings type as context menu.

Default value

[]

Example

To get the widget's Guid, refer to getCurrentDashboardDetails

JS


```
$("#container").ejDashboardViewer({
widgetActionSettings:[{
widgetId:"957d0fa9-0eb1-4b43-8626-d4b8e0f5e625"            // set the Guid
of the dashboard widget
actionSettings:[{
type:"Click",
actions:["Filter"]   // When click the widget data it perform master
interaction
}]
}
{
widgetName:"923f0fa9-0eb1-4b43-8626-d4b8e0f5e625"          // set the Guid
of the dashboard widget
actionSettings:[{
type:"DoubleClick",
actions:["Linking"]  // When double click the widget data it perform linking
}]
}
{
widgetName:"322d0fa9-0eb1-4b43-8626-d4b8e0f5e625"          // set the Guid
of the dashboard widget
actionSettings:[{
type:"ContextMenu",
actions:["DrillDown, Filter"]   // When right click the widget data context
menu shown with drill down and filter options in the menu
}]
}]
});
```

widgetModeSettings `object`

Specifies the widget Mode of the dashboard viewer.

widgetModeSettings.id `string`

Get the widget id attribute which is loaded in Dashboard Viewer.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
widgetModeSettings:{id:"dashboardviewer"}
});
```

widgetModeSettings.name `string`

Gets name of the widget which is loaded in Dashboard Viewer.

Default value

""

Example

JS

```
$("#container").ejDashboardViewer({
widgetModeSettings:{name:""}
```



```
});
```

Methods

applyComments()

This method will mark the widgets that are commented or not.

Following are the parameter that you can pass to this method,

Parameters	Type	Description
commentSettings	object	Returns the comment settings values are null or not.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Add the comments to dashboard.
$("#dashboard").ejDashboardViewer({
allowComment: true,
commentSettings: {
isDashboardCommented: true,
isWidgetCommented: true,
widgets: [1]
}
});
var dashboard = $("#dashboard").ejDashboardViewer();
dashboard.applyComments(true);
</script>
```

applyCustomTheme()

This method will apply custom theme to dashboard or widget.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer(enableCustomTheming: true);
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.applyCustomTheme(); // add the custom theme to widget.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// add the custom theme to widget.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("applyCustomTheme");
</script>
```


clearAllFilters()

This method will reset all master filters applied to the dashboard.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.clearAllFilters(); // clear the dashboard filter.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// clear the dashboard filter.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("clearAllFilters");
</script>
```

clearWidgetFilter()

This method will reset the filter applied by a master widget.

Following are the parameters that you can pass to this method,

Parameters	Type	Description
id	string	Specifies the id of the master widget whose filter need to be removed from the dashboard.
widgetType	string	Specifies the master widget type such as combo box, chart etc.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.clearWidgetFilter("Chart_1","Chart"); // clear the widget
filter.
</script>
```

closeAllWindows()

This method is used to close all the popups/windows currently opened.Which includes maximize, export, filter overview, description and filter panel windows

Example

HTML


```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.closeAllWindows();
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
$("#dashboard").ejDashboardViewer("closeAllWindows");
</script>
```

closeFilterPanel()

This method will collapse the filter panel.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.closeFilterPanel(); // collapse the filter panel.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// collapse the filter panel.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("closeFilterPanel");
</script>
```

closeViewDataDialog()

This method will close the opened view data dialog in the dashboard view.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.closeViewDataDialog(); // close the View Data dialog.
</script>
```

exportToExcel()

This method will export widget data into excel file.

Following are the parameters that you can pass to this method:

Parameters	Type	Description
Widget/Dashboard Name	string	Denotes the name of the Dashboard/Widget.
filename	string	Denotes the export Excel Sheet title name.
fileType	string	Denotes the Excel file type such as XLSX or XLS.
hideProgress	bool	The bool value specifies to Show/hide the loading indicator.
isWholeDashboard	bool	The bool value(true/false) that specifies whether to export the whole dashboard or not.
dashboardCollection	array	Contain the collection of Multi tab in an array, that specifies which are the tab to be exported from the dashboard.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.exportToExcel("Chart_1", "Worldwide Car Sales Random data",
"xlsx",true,false,[]); // export the widget data.
</script>
```

exportToImage()

This method will export the dashboard or widget in an image.

Following are the parameters that you can pass to this method:

Parameters	Type	Description
Widget/Dashboard name	string	Denotes the name of the Dashboard/Widget.
imageType	string	Denotes the export image type such as jpg, png etc..
resolution	string	Denotes the resolution of the image.
fileName	string	Denotes the file name of the exported image.
title	string	Denotes the exported image title name.
hideProgress	bool	The bool value specifies to Show/hide the loading indicator.
isWholeDashboard	bool	The bool value(true/false) that specifies whether to export the whole dashboard or not.
dashboardCollection	array	Contain the collection of Multi tab in an array, that specifies which are the tab to be exported from the dashboard.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboard = $("#dashboard").data("ejDashboardViewer");
dashboard.exportToImage("Chart_1", "jpg", "96", "Worldwide Car Sales Random
data", "Worldwide Car Sales (Random data)",false, true,[]); // export the
dashboard or widget.
</script>
```

exportToPdf()

This method will export the dashboard widget into a PDF file.

Following are the parameters that you can pass to this method:

Parameters	Type	Description
Widget/Dashboard name	string	Denotes the name of the Dashboard/Widget.
fileName	string	Denotes the name of the PDF file.
pageSize	string	Denotes the exported PDF page size such as A4,A3 etc.
title	string	Denotes the title of the dashboard or widget to be shown in the exported PDF document.
orientation	string	Denotes the orientation of the document such as Portrait or landscape.
hideProgress	bool	The bool value specifies to Show/hide the loading indicator.
isWholeDashboard	bool	The bool value(true/false) that specifies whether to export the whole dashboard or not.
dashboardCollection	array	Contain the collection of Multi tab in an array, that specifies which are the tab to be exported from the dashboard.
widgetCollection	array	Contain the collection of widgets in an array, that specifies which are the Widget to be exported from the dashboard.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboard = $("#dashboard").data("ejDashboardViewer");
dashboard.exportToPdf("dashboard","Worldwide Car Sales Random
data","A4","Worldwide Car Sales (Random
```



```
data)","portrait",true,true,[],[Sales by country,Sales by car type]); //
export the widget data
</script>
```

getCurrentDashboardDetails()

This method will returns the widgets information like Widget Title, Widget Subtitle, Widget Description, Widget Id, Widget Guid and Widget Type of the currently active dashboard.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getCurrentDashboardDetails(); // get the widget info of the
current dashboard
</script>
```

getCurrentFilters()

This method will return the collection of filter applied in the current dashboard view.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getCurrentFilters(); // get the current filter.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// get the current filter.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("getCurrentFilters");
</script>
```

getCurrentTab()

This method will return the current tab element.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
```



```
dashboardObj.getCurrentTab(); // select the current tab.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// select the current tab.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("getCurrentTab");
</script>
```

getCurrentViewUrl()

This method will return the current view (with applied filters) of the dashboard as a URL.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getCurrentViewUrl();
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
$("#dashboard").ejDashboardViewer("getCurrentViewUrl");
</script>
```

getDashboardData()

This method will return the layout information and widgets data of the currently rendered dashboard.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getDashboardData(); // get the detailed information of the
rendered dashboard.
</script>
```

getDashboardTabsInfo()

This method will return the currently active tab data and its meta data.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
```



```
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getDashboardTabsInfo(); // get the dashboard tab info.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// get the dashboard tab info.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("getDashboardTabsInfo");
</script>
```

getFilterPanel()

This method will return the filter panel instance. If no filter panel is rendered, it will return null.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getFilterPanel(); // get the filter panel instance.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// get the filter panel instance.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("getFilterPanel");
</script>
```

getFilterPanelParent()

This method will return the element where the filter panel instance is bind. If no filter panel is rendered, it will return null.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getFilterPanelParent(); // get the filter panel parent.
</script>
```


HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// get the filter panel parent.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("getFilterPanelParent");
</script>
```

getFilterPanelState()

This method will return the state of filter panel. Whether the filter panel is in open or close state.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getFilterPanelState(); // get the filter panel state.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
//  get the filter panel state.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("getFilterPanelState");
</script>
```

getMasterControlsBySlave()

This method will get the master controls by slave controls.

Following are the parameter that you can pass to this method:

Parameters	Type	Description
element	string	Returns the master widget list for the given widget id.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getMasterControlsBySlave("Chart_1"); // get the master control.
</script>
```


getWidgetDataByReportName()

This method will return the data of the given widget based on its ID/name.

Following are the parameter that you can pass to this method:

Parameters	Type	Description
id	string	Returns the the given widget data based on its ID.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getWidgetDataByReportName("ComboBox_2"); // get the widget data
by report name.
</script>
```

getWidgetTitle()

This method will refund the heading/title of the given widget.

Following are the parameter that you can pass to this method:

Parameters	Type	Description
id	string	The id of the widgets whose title need to be returned.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.getWidgetTitle("DatePicker_1"); // get the widget title.
</script>
```

loadDashboard()

This method loads the layout information and dashboard information and renders the dashboard.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.loadDashboard(); // Load the dashboard.
```



```
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Load the dashboard.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("loadDashboard");
</script>
```

loadHistoryData()

This method get sessionKey as an argument and render the dashboard from the corresponding session data of the passed session key. It may returns the true value for successful rendering of the dashboard.

Name	Type	Description
args	string	Which holds the browser state key(GUID).

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.loadHistoryData();
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
$("#dashboard").ejDashboardViewer("loadHistoryData");
</script>
```

loadLayout()

This method will load the layout information and render the layout alone.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.loadLayout(); // Load the dashboard layout.
</script>
```

HTML


```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Load the dashboard layout.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("loadLayout");
</script>
```

openFilterPanel()

This method is used to open the filter panel.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.openFilterPanel(); // show the filter panel.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// show the filter panel.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("openFilterPanel");
</script>
```

openGetLinkDialog()

This method is used to open the GetLink dialog.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.openGetLinkDialog();
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
$("#dashboard").ejDashboardViewer("openGetLinkDialog");
</script>
```

openParameterDialog()

This method is used to open the dashboard parameter dialog.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.openParameterDialog();
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
$("#dashboard").ejDashboardViewer("openParameterDialog");
</script>
```

redrawDashboard()

This method will re-render the complete dashboard and will reset the event hooks.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.redrawDashboard(); // Re-render the dashboard.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Re-render the dashboard.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("redrawDashboard");
</script>
```

refreshDashboard()

This method will refresh the whole dashboard data or reload a specific widget data.

Following are the parameters that you can pass to this method:

Parameters	Type	Description
controlId	string	Returns the name of the Control ID.
canAutoRefresh	boolean	Return the bool value that specifies whether to autorefresh the whole dashboard.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.refreshDashboard("Grid_1",true); // Refresh the dashboard.
</script>
```

refreshData()

This method will refresh the dashboard in viewer.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.refreshData(); // refresh the dashboard data.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// refresh the dashboard data.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("refreshData");
</script>
```

refreshWidgetData()

This method will refresh the widget data.

Following are the parameter that you can pass to this method:

Parameters	Type	Description
WidgetList	string	Returns the widget list for the given widget id.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.refreshWidgetData("Chart_1"); // refresh the widget data.
</script>
```


removeCommentToggleState()

This method will remove the comment toggle state.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.removeCommentToggleState(); // remove the comment toggle state.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// remove the comment toggle state.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("removeCommentToggleState");
</script>
```

resizeDashboard()

This method will resize the dashboard viewer to fit the parent container size.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.resizeDashboard(); // resize the dashboard.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// resize the dashboard.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("resizeDashboard");
</script>
```

selectTabByIndex()

This method will make the tab with the given tab index as active. If an invalid index is given there will not be any action done.

Following are the parameter that you can pass to this method:

Parameters	Type	Description
tabIndex	Integer	Returns the tab with given tab index.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.selectTabByIndex(1); // load the tab by tab index.
</script>
```

selectTabByName()

This method will make the tab with the given tab name as active. If an invalid tab name is given no action will be done.

Following are the parameter that you can pass to this method,

Parameters	Type	Description
tabName	string	Returns the tab name with the given tab name as active.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.selectTabByName("Northwind Analysis Dashboard"); // load the
tab by tab name.
</script>
```

showViewData()

This method will show the data view of a particular widget based on its Guid/ReportName. Enable the dashboard parameter enableViewData to preview the data.

Following are the parameter that you can pass to this method,

Parameters	Type	Description
Guid	guid	Returns the given widget data based on its current dashboard widget's guid.

Example

To get the widget's Guid, refer to getCurrentDashboardDetails

HTML


```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.showViewData("957d0fa9-0eb1-4b43-8626-d4b8e0f5e625"); // set
the Guid of the dashboard widget.
</script>
```

toggleDashboardComment()

This method will toggle the dashboard comment.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.toggleDashboardComment(); // toggle the dashboard comment.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// toggle the dashboard comment.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("toggleDashboardComment");
</script>
```

toggleDashboardParameterIcon()

This method will return the current view (with applied filters) of the dashboard as a URL.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.toggleDashboardParameterIcon();
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
$("#dashboard").ejDashboardViewer("toggleDashboardParameterIcon");
</script>
```


toggleFavoriteIcon()

This method will toggle the favorite icon i.e. mark and unmarked the favorite icon.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.toggleFavoriteIcon(); // toggle the favorite icon.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// toggle the favorite icon.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("toggleFavoriteIcon");
</script>
```

toggleWidgetComment()

This method will toggle the widget comment.

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.toggleWidgetComment(); // toggle the widget comment.
</script>
```

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// toggle the widget comment.
$("#dashboard").ejDashboardViewer();
$("#dashboard").ejDashboardViewer("toggleWidgetComment");
</script>
```

updateFilterOverview()

This method will update the filter overview with the applied filters data.

Following are the parameters that you can pass to this method:

Parameters	Type	Description
viewName	string	Name of the Filter View

Parameters	Type	Description
viewName	string	Unique ID of the Filter View

Example

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.updateFilterOverview("NorthWind","94a48f0e-2898-4ccc-ad8f-
b421ff87d25f");
</script>
</script>
```

updateTabName()

This method will update the tab name in the multi-tab dashboards.

Following are the parameters that you can pass to this method:

Parameters	Type	Description
tabId	string	Id of the tab to be updated
displayName	string	Updated name of the tab

Example

To get the tab details of the existing dashboard, refer to getDashboardTabsInfo

HTML

```
<div id="dashboard" style="height:100%;width:100%"></div>
<script>
// Create DashboardViewer
$("#dashboard").ejDashboardViewer();
var dashboardObj = $("#dashboard").data("ejDashboardViewer");
dashboardObj.dashboardObj.updateTabName([{ tabId:
'Dashboard_adeaf53d_03d9_4fe3_b9ba_e70d12d0e3eb', displayName: 'Worldwide
Sales (Random data)' }]);
</script>
</script>
```

Events

actionBegin

This event will be triggered at the beginning of every viewer actions while applying connection, layout/ dashboard rendering, popup opening and close, refreshing, filtering, exporting etc.

Name	Type	Description
eventType	Enum	Will holds the current event type that triggered the event
source	object	This will contain the data related to the current event
Source.element	Object	Holds the current event site UI element. If there involves no UI element will be returned as null
Source.data	object	Holds the current event data.

Example

JS

```
$("#container").ejDashboardViewer({
actionBegin: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

actionComplete

This event will be triggered on completion of the every viewer action which completes.

Name	Type	Description
eventType	Enum	Will holds the current event type that triggered the event
source	object	This will contain the data related to the current event
Source.element	Object	Holds the current event site UI element. If there involves no UI element will be returned as null
Source.data	object	Holds the current event data.

Example

JS

```
$("#container").ejDashboardViewer({
actionComplete: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

afterLayoutRender

Triggered after the dashboard layout is rendered.

Example

JS

```
$("#container").ejDashboardViewer({
afterLayoutRender: function (args) {
// Write a code block to perform an operation after layout rendered }
```



```
});
```

afterNavigate

This event is triggered after linking operation is completed.

Url / Dashboard Linking

Name	Type	Description
url	string	Will holds the URL to be navigated.
target	string	Will holds the target type.
source	Object	Holds the source object where the linking triggered.

Internal Linking

Name	Type	Description
tabName	string	Will hold the tab name.
source	Object	Holds the source object where the linking triggered.

Example

JS

```
$("#container").ejDashboardViewer({
afterNavigate: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

afterWidgetRender

Triggered after each widget gets rendered in viewer.

Example

JS

```
$("#container").ejDashboardViewer({
afterWidgetRender: function (args) {
// Write a code block to perform an operation on widget after render }
});
```

beforeContextMenuOpen

Triggered before the context menu is opened for each widget.

Example

JS

```
$("#container").ejDashboardViewer({
beforeContextMenuOpen: function (args) {
// Write a code block to perform an operation if before context menu open }
});
```


beforeControlMenuOpen

This event gets triggered before the menu is opened upon clicking the more icon for each widget.

Example

JS

```
$("#container").ejDashboardViewer({
beforeControlMenuOpen: function (args) {
// Write a code block to perform an operation if before control menu open }
});
```

beforeDashboardFetch

This event will be triggered while fetching the dashboard data or rendering the dashboard layout. Following table illustrates this event's arguments:

Name	Type	Description
model	Object	Holds the Dashboard Viewer model set by the user.
setRequestHeader	void	This method is used for setting the custom header and its value. User can call this method n number of types to set the custom headers.If the given header key is already present in the custom header, then the value alone will be updated.
isHeaderPresent	bool	This method checks whether the given header key is already present in the custom header collection or not.

Example

JS

```
$("#container").ejDashboardViewer({
beforeDashboardFetch: function (args) {
// Write a code block to perform an operation to pass custom header when
fetching dashboard data
args.setRequestHeader(key, value);
}
})
```

beforeDashboardIconRendered

This event get Triggered before the Dashboard icons are rendered in Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
beforeDashboardIconRendered: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```


beforeFilterPanelClose

This event Fires before the filter panel is closed on the Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
beforeFilterPanelClose: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

beforeFilterPanelOpen

This event fires before the Filter panel is opened on the Viewer.

Name	Type	Description
model	object	returns the Dashboard Viewer model
datasource	string	returns the data source of widgets rendered
element	object	returns the HTML element of DashboardViewer where the filter panel is rendered

JS

```
$("#container").ejDashboardViewer({
beforeFilterPanelOpen: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
);
```

beforeLayoutRender

Triggered before the dashboard layout is rendered.

Example

JS

```
$("#container").ejDashboardViewer({
beforeLayoutRender: function (args) {
// Write a code block to perform an operation before Layout rendered }
});
```

beforeMasterFilterApplied

Event gets triggered before the master widget filter is applied to the slave widgets.

Example

JS

```
$("#container").ejDashboardViewer({
beforeMasterFilterApplied: function (args) {
// Write a code block to perform an operation before MasterFilter applied }
});
```


<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>{% highlight html %}argument{% endhighlight %}</td><td>object</td><td>Event parameters from Dashboard Viewer
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>{% highlight html %}event{% endhighlight %}</td><td>object</td><td>Handle the filter values applied on dynamic interaction(s) with master widget and supply the modified values to its listener(s) conditionally.</td></tr>
</table></td></tr>
</table>

beforeNavigate

This event is triggered before the linked URL is navigated.

Url / Dashboard Linking

In beforeNavigate Event arguments.data will have the following information

Name	Type	Description
url	string	Will holds the URL to be navigated.
target	string	Will holds the target type.
handled	bool	If this is set as true, the navigation code in viewer will not be triggered.
source	Object	Holds the source object where the linking triggered.

Internal Linking

In beforeNavigate Event arguments.data will have the following information

Name	Type	Description
tabName	string	Will hold the tab name.
handled	bool	If this is set as true, the navigation code in viewer will not be triggered.
source	Object	Holds the source object where the linking triggered.

Example

JS

```
$("#container").ejDashboardViewer({
beforeNavigate: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

beforePassiveErrorRender

This event get triggered before the Passive Error Dialog show on the Dashboard Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
beforePassiveErrorRender: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

beforeWidgetIconRendered

This event gets triggered before the title icons is rendered for each widget.

Example

JS

```
$("#container").ejDashboardViewer({
beforeWidgetIconRendered: function (args) {
// Write a code block to perform an operation before the widget icon before
rendered }
});
```

beforeWidgetRender

Triggered before each widget gets rendered in viewer.

Example

JS

```
$("#container").ejDashboardViewer({
beforeWidgetRender: function (args) {
// Write a code block to perform an operation before widget rendered }
});
```

dashboardCreated

This event will be triggered when the dashboard is created.

Example

JS

```
$("#container").ejDashboardViewer({
dashboardCreated: function (args) {
// Write a code block to perform an operation after dashboard created }
});
```

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>{% highlight html %}argument{% endhighlight %}</td><td>object</td><td>Event parameters from Dashboard Viewer
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>{% highlight html %}model{% endhighlight %}</td><td>object</td><td>Returns the Dashboard Viewer model</td></tr>
</table>
</td></tr>
</table>

Name	Type	Description		
		{% highlight html %}data{% endhighlight %}	object	Hold the Dashboard Viewer data and layout information

destroy

Event gets triggered when Widget and Dashboards are destroyed

Example

JS

```
$("#container").ejDashboardViewer({
destroy: function (args) {
// Write a code block to perform an operation Dashboard destroy }
});
```

filterPanelClose

This Event Fires after Filter panel is closed.

Example

JS

```
$("#container").ejDashboardViewer({
filterPanelClose: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

filterPanelOpen

This Event Fires after Filter panel is opened.

Example

JS

```
$("#container").ejDashboardViewer({
filterPanelOpen: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onApplyConnection

Triggered before the first AJAX request is send to service. You can modify the connection string using this event.

Example

JS

```
$("#container").ejDashboardViewer({
onApplyConnection: function (args) {
// Write a code block to perform an operation if connection string is
changed }
});
```


onBannerIconClick

This event Fires, when banner get clicked in Dashboard Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
onBannerIconClick: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onCommentDialogClosing

Triggered when the comment window is closing.

Example

JS

```
$("#container").ejDashboardViewer({
onCommentDialogClosing: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onDashboardClick

Triggered when ever a click is made over dashboard area.

Example

JS

```
$("#container").ejDashboardViewer({
onDashboardClick: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onDashboardCommented

This event triggers, when the dashboard gets commented in Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
onDashboardCommented: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onFavoriteStateChange

This event will be triggered when the user clicks on the favorite icon in viewer.

Example

JS


```
$("#container").ejDashboardViewer({
onFavoriteStateChange: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onFilterOverviewUpdated

This event will be triggered before the filter overview gets updated with the new filter data.

Example

JS

```
$("#container").ejDashboardViewer({
onFilterOverviewUpdated: function (args) {
// Write a code block to perform an operation before filter over view is
updated }
});
});
```

onHistoryPopStateChange

This event triggers, while click on the browser back and forth buttons.

Example

JS

```
$("#container").ejDashboardViewer({
onHistoryPopStateChange: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onMenuIconClick

Triggered when the menu icon is clicked from the dashboard widget container.

Example

JS

```
$("#container").ejDashboardViewer({
onMenuIconClick: function (args) {
// Write a code block to perform an operation on Dashboard menu icon click
}
});
```

onMenuItemClick

Triggered when a menu item is selected from the drop-down menu of each widget or the dashboard.

Example

JS

```
$("#container").ejDashboardViewer({
onMenuItemClick: function (args) {
// Write a code block to perform an operation on Dashboard menu item click
}
});
```


onSaveAsFilter

This event gets Triggered when you click the save As icon while filtering on Viewer.

Name	Type	Description
model	object	returns the Dashboard Viewer model
data	object	Contains the data of the Filtered Information
dashboard id	string	Guid of the Dashboard
tab info	object	contains the current Tab information in case of multiple tab dashboards

Example

JS

```
$("#container").ejDashboardViewer({
onSaveAsFilter: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onSaveFilter

This event gets Triggered when you click the save icon while filtering.

Name	Type	Description
model	object	returns the Dashboard Viewer model
data	object	Contains the data of the Filtered Information
dashboard id	string	Guid of the Dashboard
tab info	object	contains the current Tab information in case of multiple tab dashboards
viewName	string	Name of the Filter View
view id	string	Id of the Filter View

Example

JS

```
$("#container").ejDashboardViewer({
onSaveFilter: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

onWidgetCommented

This event triggers, when widget gets commented in Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
onWidgetCommented: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

tabActive

This event get Triggered after a tab activated in Viewer.

Example

JS

```
$("#container").ejDashboardViewer({
tabActive: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

tabSelect

This event get Triggered after a tab selected in Viewer.

Example

JS

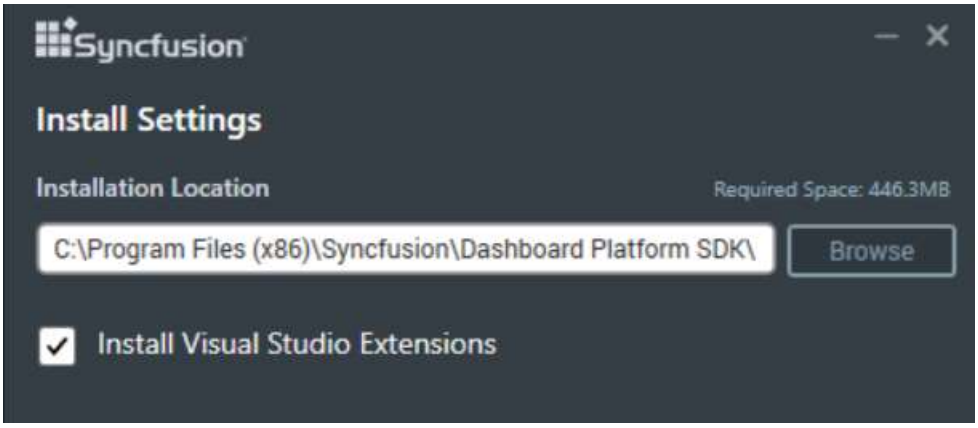
```
$("#container").ejDashboardViewer({
tabSelect: function (args) {
// Write a code block to perform an operation on click anywhere in Dashboard
}
});
```

Overview

The Syncfusion Dashboard Viewer ASP.NET Extensions provide quick access to create or configure the Syncfusion ASP.NET projects.

Configure Syncfusion Dashboard Platform SDK Extension

1. Refer here to install Syncfusion Dashboard Platform SDK.
2. Enable the Install Visual Studio Extensions check box, while installing the Syncfusion Dashboard Platform SDK.



The Syncfusion Dashboard Platform SDK Extension provides the following project templates for the Dashboard Viewer.

1. Syncfusion Dashboard Viewer Project Templates for ASP.NET Core Application.
2. Syncfusion Dashboard Viewer Project Templates for ASP.NET MVC Application.
3. Syncfusion Dashboard Viewer Project Templates for ASP.NET Web Forms Application.

Dashboard Designer (Desktop)

Overview

The Syncfusion Dashboard Designer application lets you connect to data and visualize them as dashboards using data visualization and filter widgets. The composed dashboard can then be published to the dashboard server for sharing from within the designer application itself.

Key features

Wide variety of data sources: Connects file based data sources such as, Microsoft Excel, CSV, and JSON or server based data sources such as SQL Server, Microsoft SQL Server Analysis Services (SSAS), Azure Analysis Services, PostgreSQL, SQLite, ODBC Connection (MySQL, Oracle and Access), Microsoft Azure Table Storage, Salesforce, Hive and Spark SQL, or RESTful web services through the Web Data Source.

Common ODBC (ANSI SQL): Connects to wide range of databases that inherit the ANSI SQL 92 standard through their ODBC drivers.

Built-in Simple ETL: You can combine multiple tables in a structured database or an excel workbook to generate a de-normalized virtual data table, which suits for visualization as dashboards.

Rich selection of widgets: All the widgets commonly used in business dashboards including a variety of charts, gauges, maps, treemaps, and grids have been included. You may also include your own HTML5 and JavaScript based custom widgets to dashboard using a few steps.

Tabbed view of dashboards: Showcases multiple dashboards in one page through the tabbed view.

Scrollable canvas with half-sized widgets: Add widgets with various sizes beyond the last row to accommodate more widgets in a scrollable view.

Seamless integration with the Dashboard Server: You can create, publish, and import dashboards, data sources, and widgets to the Dashboard Server within the designer application.

SQL Server Impersonation within domain: Under Windows authentication mode, user accounts imported from Active Directory to Dashboard Server will impersonate as Dashboard Server user to communicate with SQL Server instance.

Dynamic Parameters: Add parameters for custom queries or stored procedures and dynamically change the parameter values in the Dashboard Viewer. Parameters can also be used in expressions and can be mapped for user-based filters.

User based filtering: Shows different data based on users permissions.

Dedicated filter panel: Allows you to save space for visualization and show filter elements in a dedicated shelf that can be collapsed. Wide range of filter elements such as combo box, date picker, range slider, list box, checkbox, and radio button can be used here.

Integration with Syncfusion Data Integration Platform: Creates a dashboard data source from Data Integration Platform data flows using the server explorer in the Dashboard Designer.

Localization: Localizes the Dashboard Designer's user interface to any desired language.

Custom branding: The dashboard designer has built-in customization capabilities allowing you to add your organization’s name, logo, and more.

Create a support incident

If you are still not able to find the information that you are looking for self-help resources mentioned above, then please contact us by creating a support ticket.

System Requirements

This section explains the system requirements to run Syncfusion Dashboard Desktop Designer.

Dashboard Designer

Hardware Requirements

The following minimum hardware requirements are necessary to run the Syncfusion Dashboard Designer:

- Processor - Dual Core 32-bit CPU (x86)
- Hard disk - 2 GB
- RAM - 2 GB

Software Requirements

The following minimum software requirements are necessary to run the Syncfusion Dashboard Designer:

- Operating System - Windows 7, 8+, Windows Server 2012+
- Microsoft .NET Framework 4.5
- IIS Express
- Browser - Internet Explorer 9+, Microsoft Edge, Mozilla Firefox 22+, Chrome 17+, Opera 12+, Safari 5+

Note: In Internet Explorer 11, if you enabled the Enterprise mode, make sure that you have disabled it. For more details, follow the link.

Note: In the Internet Explorer, make sure that you have turned off the Compatibility Settings for the intranet sites. To know how to turn off the Compatibility Settings, follow the link.

The following minimum software requirements are necessary in a data server (can be local or remote) for respective server connection types:

- Microsoft SQL Server - Microsoft SQL Server 2005+
- Microsoft SQL Server Analysis Services - Microsoft SQL Server 2012+
- PostgreSQL – PostgreSQL Server 9.x+
- Spark SQL - Syncfusion Big Data Cluster Manager (Spark Server should be started through Service Manager after installation. It will be running on port 10001 by default.)
- Hive - Syncfusion Big Data Cluster Manager (Hive Server should be started through Service Manager after installation. It will be running on port 10000 by default.)
- SQL through ODBC Connection - SQL Native Client or SQL Server Native Client 10.0 (Any one of the these drivers need to be installed and database need to be setup as discussed here).
- MySQL through ODBC Connection - MySQL ODBC 5.3 Unicode Driver (The driver need to be installed and database need to be setup as discussed here).
- Oracle through ODBC Connection - Microsoft ODBC for Oracle or Oracle in OraClient 11g_home1 (Any one of the these drivers need to be installed and database need to be setup as discussed here).
- Access through ODBC Connection - 2007 Office System Driver: Data Connectivity Components (The driver need to be installed and database need to be setup as discussed here).

Dashboard Data Agent

Hardware Requirements

The following minimum hardware requirements are necessary to run the Syncfusion Dashboard Data Agent:

- Processor - Dual Core 32-bit CPU (x86) (depends upon the data)
- Hard disk - 2 GB (depends upon the data)
- RAM - 4 GB (minimum)

Software Requirements

The following minimum software requirement is necessary to run the Syncfusion Dashboard Data Agent:

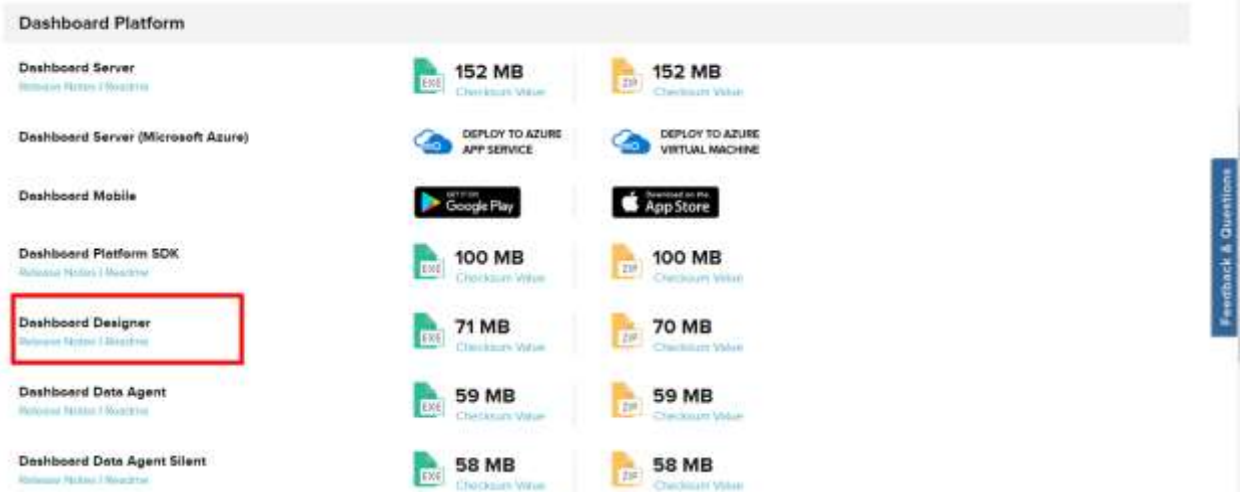
- Microsoft SQL Server 2005+

Installation

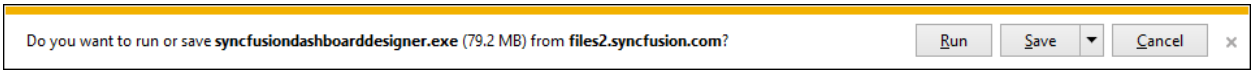
This section discusses about the installation/uninstallation, upgrading, service hosting procedures of Syncfusion Dashboard Designer.

Downloading Dashboard Designer

Download the Syncfusion Dashboard Designer from here.Licensed customers can also download the install from the downloads section. You may download either as EXE file or as ZIP file and extract the EXE.



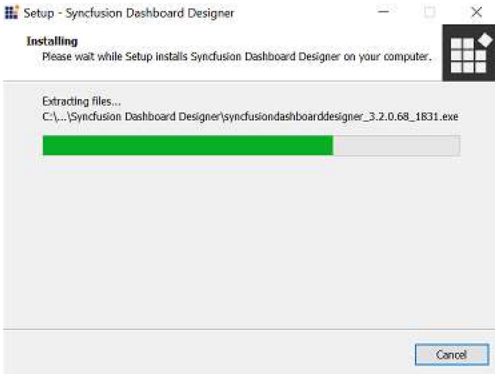
Save it to your preferred location in your machine.

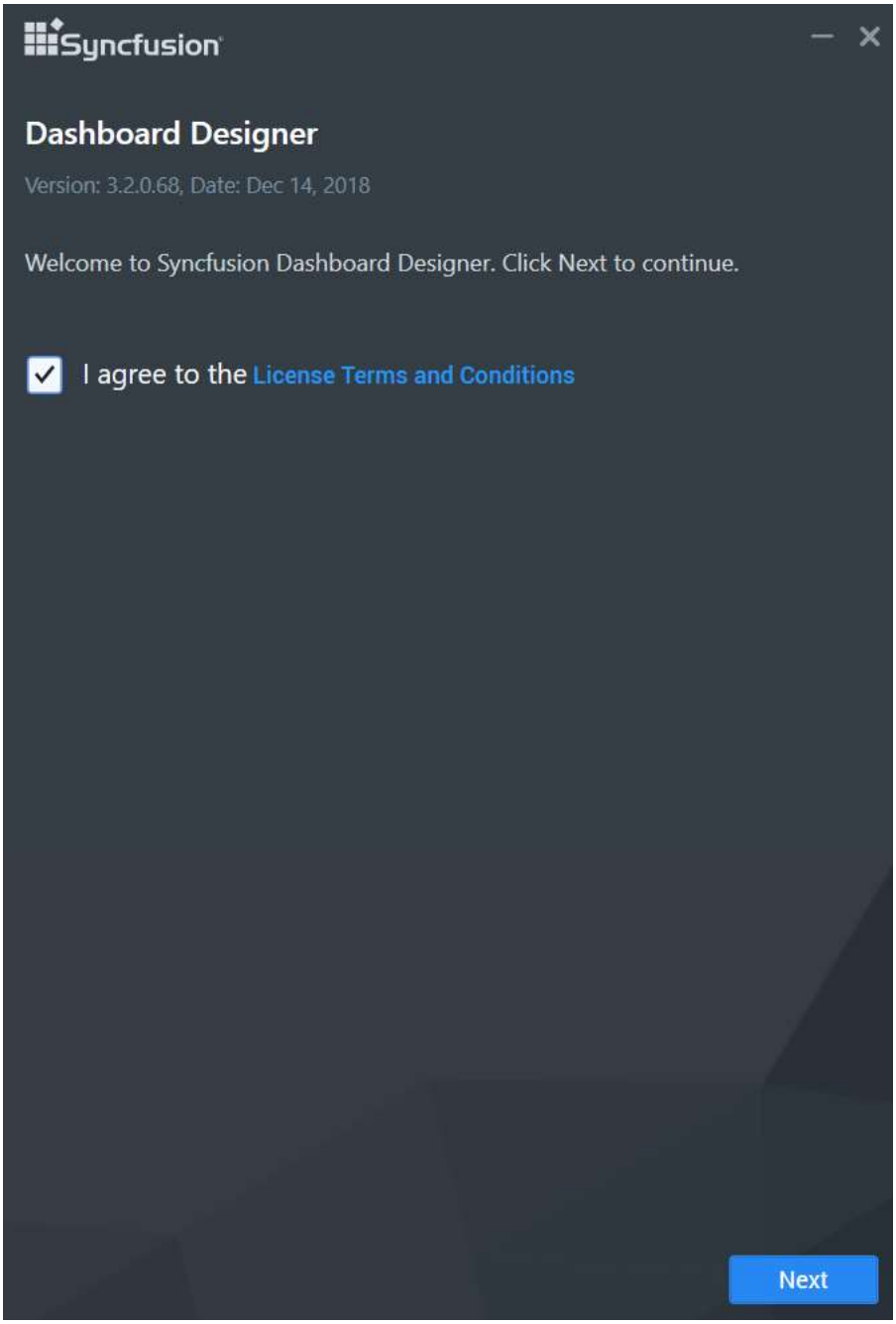


Installing Dashboard Designer

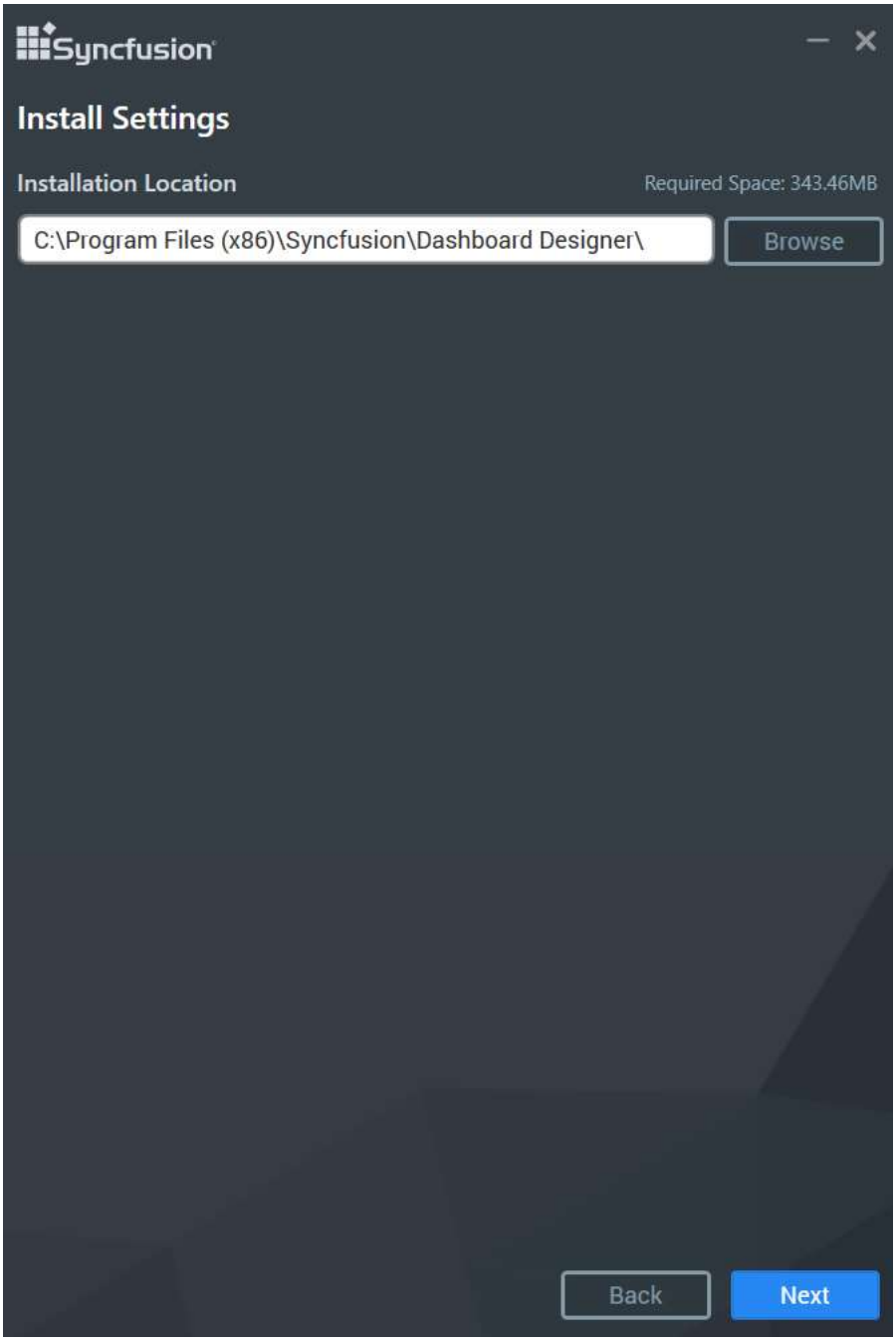
To learn about the system requirements needed to install the Syncfusion Dashboard Designer in your machine, see System Requirements section.

Run the saved installer either through clicking the Run button or by double-clicking the EXE file from the saved location.



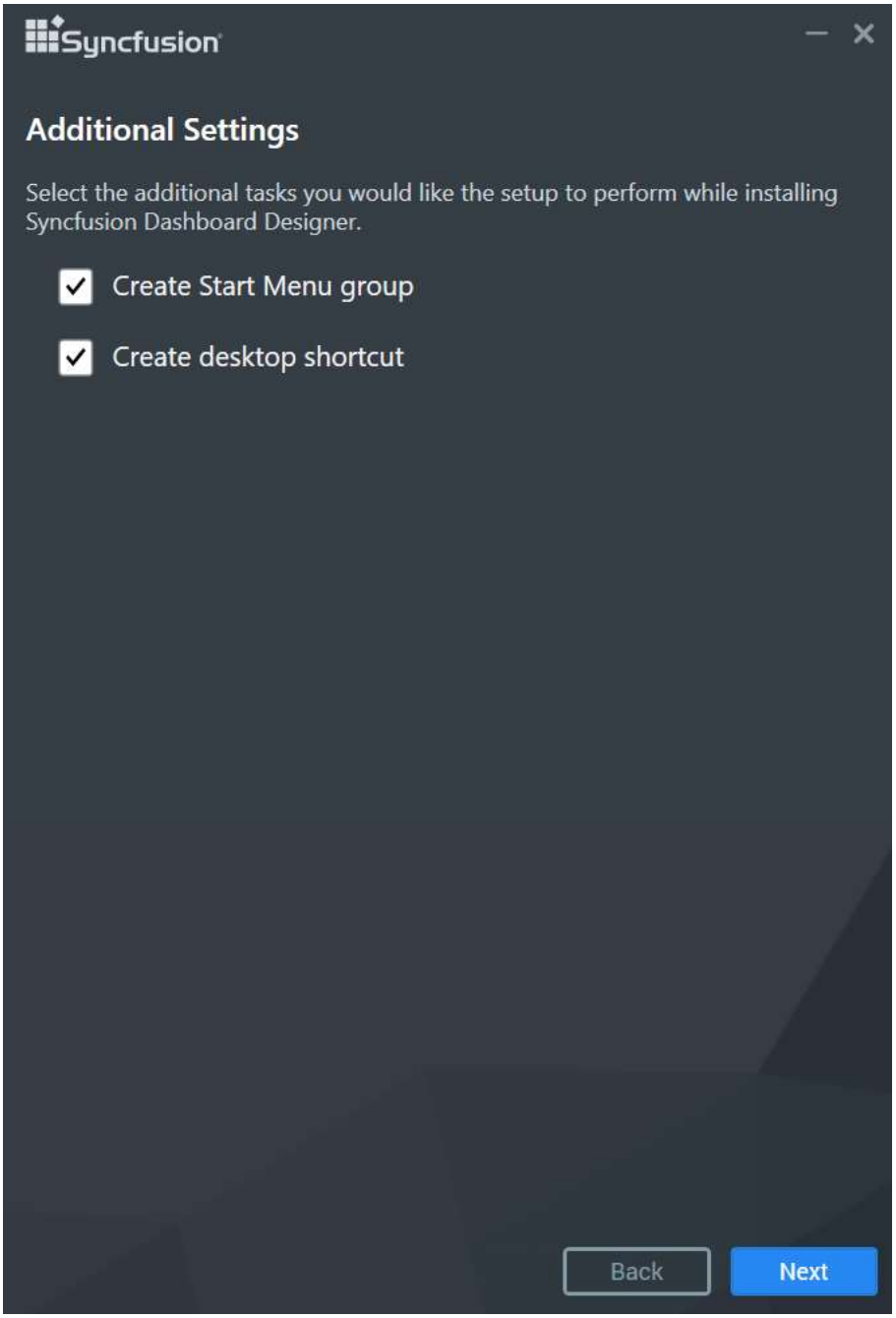


Read and accept the license terms and conditions through checking the option I accept the terms and conditions and click NEXT.



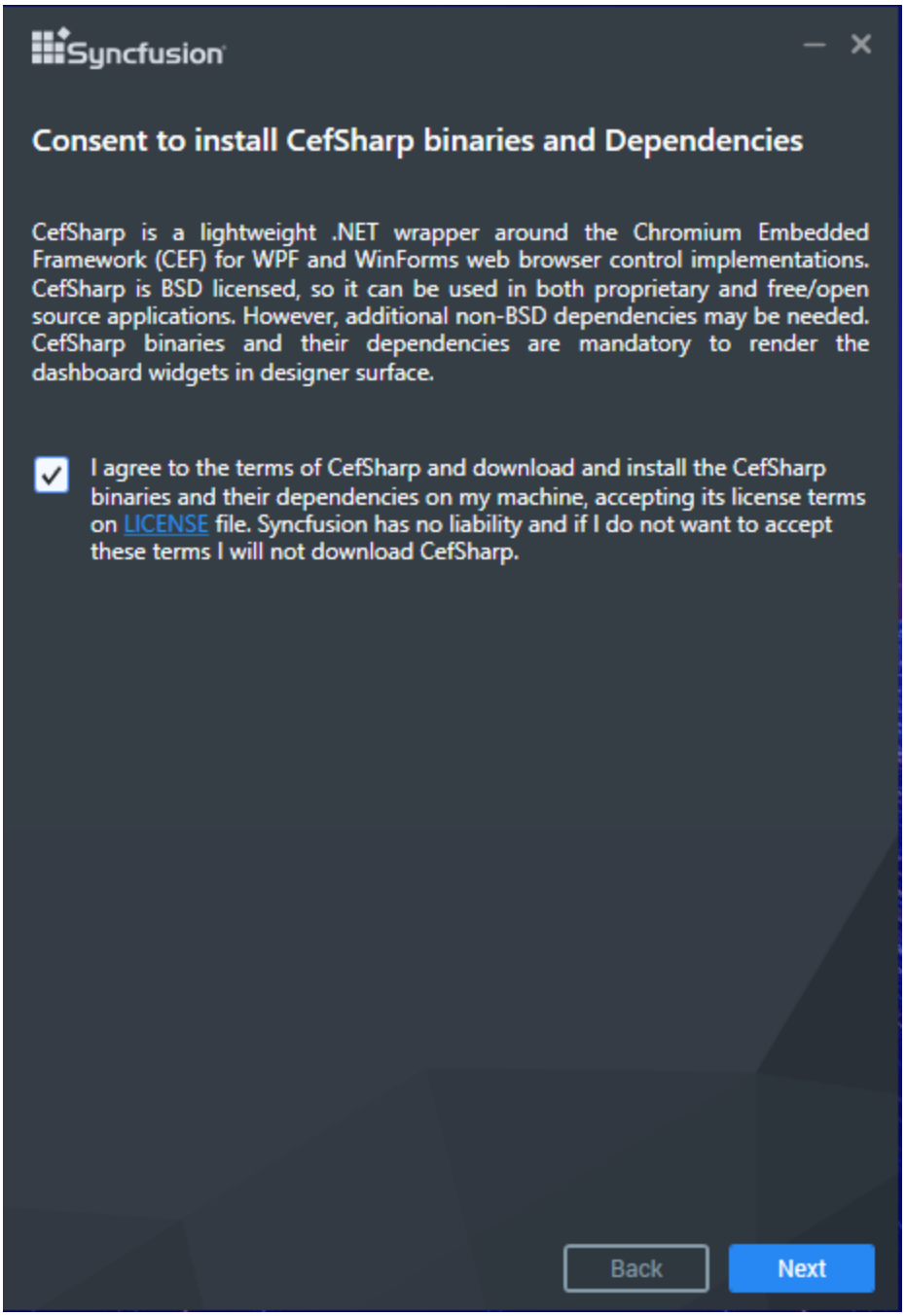
Browse to the location where you want to install the Dashboard Designer application, and then click Next.

Perform the additional tasks desktop shortcuts creation and start menu shortcut creations. If you want to perform the additional tasks, you can check the options. Otherwise, you can uncheck and click NEXT`.



Read and accept the Consent the install CefSharp binaries and Dependencies and click NEXT.

Information: cefSharp is a lightweight .NET wrapper around the Chromium Embedded Framework (CEF) used for WPF and WinForms web browser control implementations. CefSharp is BSD licensed, so it can be used in both proprietary and free/open source applications. However, additional non-BSD dependencies may be needed. CefSharp binaries and their dependencies are mandatory to render dashboard widgets in the designer surface.

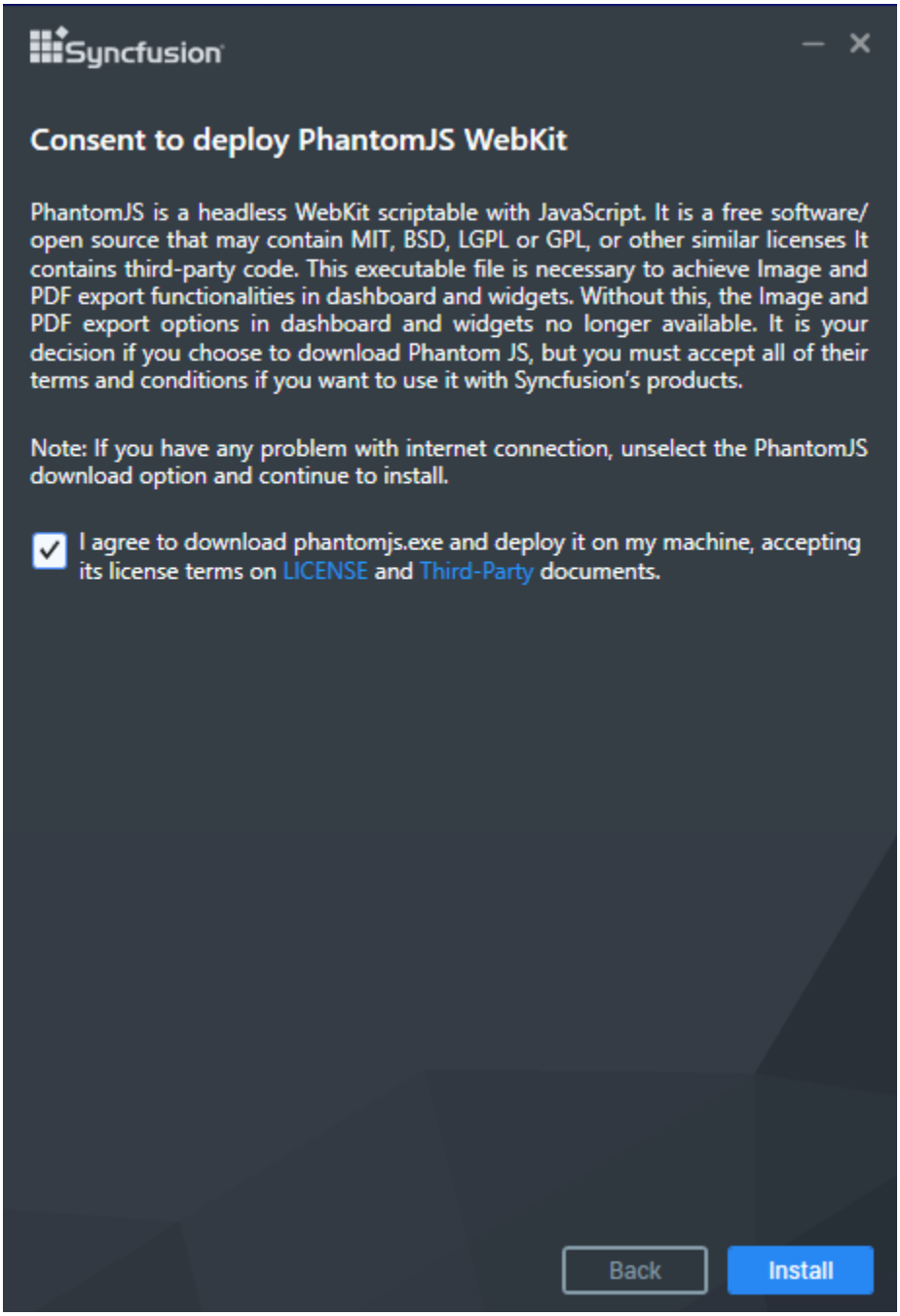


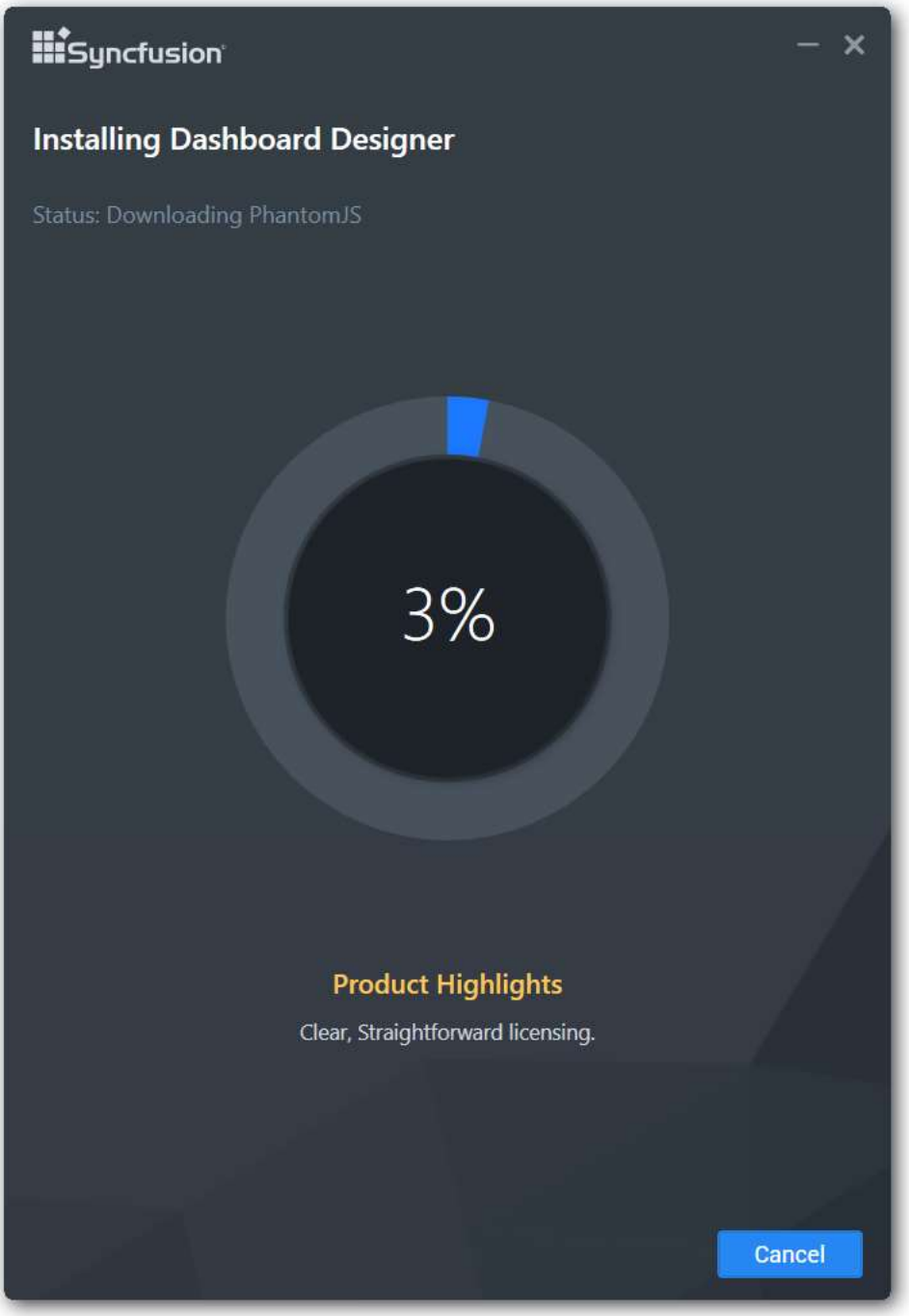
Read and accept the Consent to deploy PhantomJS Webkit and click Install.

Information: PhantomJS is a headless WebKit scriptable with JavaScript. This is a free software/open source, and it may contain MIT, BSD, LGPL, or GPL, or other similar licenses that contain third-party code. This executable file is necessary to achieve Image and PDF export functionalities in the Dashboard and widgets. Without this file, the image and PDF export options in the Dashboard and widgets will no longer be available. If you choose to download PhantomJS, must accept all terms and conditions to use it with Syncfusion's products.

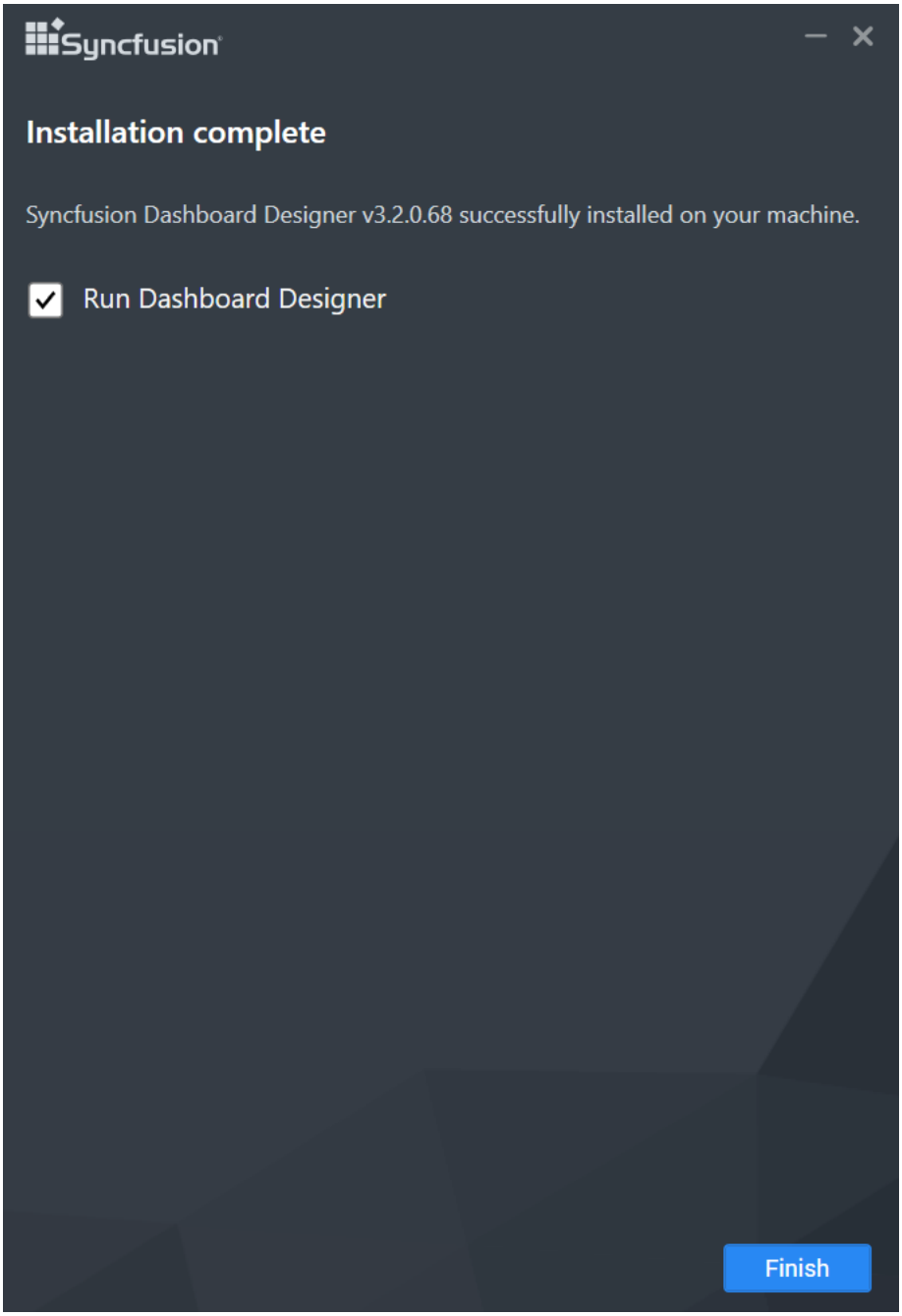
Note: If you have any problem with internet connection or do not have internet connection, unselect the PhantomJS download option and continue to install. To manually install the PhantomJS, please refer

this.





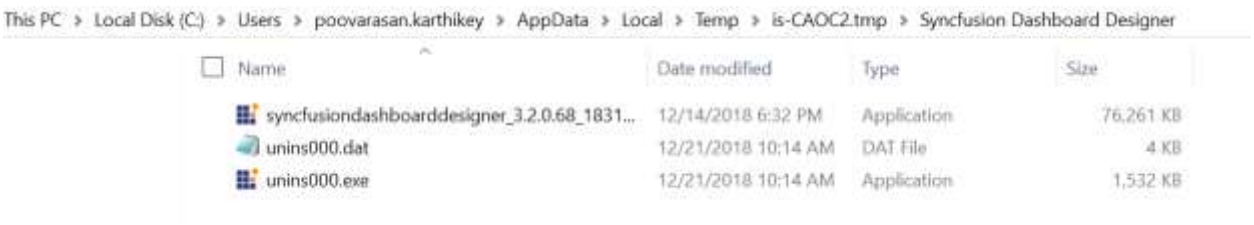
Now, the installation begins. You can cancel the installation at anytime by pressing CANCEL, if you prefer.



On successful installation, the above screen appears. Click Finish to close the installation wizard and run the newly installed Dashboard Designer. You may also run the application later through unchecking the option Run Dashboard Designer.

Silent Installation

1.Double click the Syncfusion Dashboard Designer setup. 2.Syncfusion Dashboard Designer setup will be extracted in Temp location (%temp%).



3.Copy the extracted Dashboard Designer setup to some other location and cancel the installation.
4.Open the command prompt with administrative privileges and run the extracted Dashboard Designer setup with the following arguments.

Arguments:

Dashboard Designer V3.1:

/Install silent /InstallPath:{InstallationPath} /Log "{LogFilePath\LogFile.log}"

Example:

/Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard /Log "C:\temp\Install.log"

```
D:\>syncfusiondashboarddesigner_3.1.0.113_0056.exe /Install silent /InstallPath:C:\Program Files (x86)\New\Dashboard
/log "C:\Program Files (x86)\New\Install.log"
```

Dashboard Designer V3.2:

/Install silent /InstallPath:{InstallationPath} /isdesktopshortcut:{TRUE or FALSE} /isstartmenushortcut:{TRUE or FALSE} /Log "{LogFilePath\LogFile.log}"

Example:

/Install silent /InstallPath:C:\New\Dashboard Designer /isdesktopshortcut:TRUE /isstartmenushortcut:FALSE /Log "C:\temp\Install.log"

```
C:\>C:\Users\labuser\Desktop\SilentSetup\syncfusiondashboarddesigner_3.2.0.68_18
31.exe /Install silent /InstallPath:C:\New\Dashboard Designer /isdesktopshortcut
:TRUE /isstartmenushortcut:FALSE /Log "C:\temp\Install.log"
```

Now, Syncfusion Dashboard Designer will be installed in silent mode.

Consent to deploy PhantomJS WebKit

PhantomJS is a headless WebKit scriptable with JavaScript. It is a free software/open source that may contain MIT, BSD, LGPL or GPL, or other similar licenses It contains third-party code. This executable file is necessary to achieve Image and PDF export functionalities in dashboard, widgets and schedules. Without this, the Image and PDF export options in dashboard, widgets and schedules no longer available. It is your decision if you choose to download Phantom JS, but you must accept all of their terms and conditions if you want to use it with Syncfusion's products.

To download PhantomJS application and deploy it on your machine, you should accept it's license terms on LICENSE and Third-Party document. Then, you can download PhantomJS by clicking here.

Once download completed, extract the zip file and then copy the PhantomJS application from the zip extracted location and paste it in the below mentioned install locations.

Install Location:

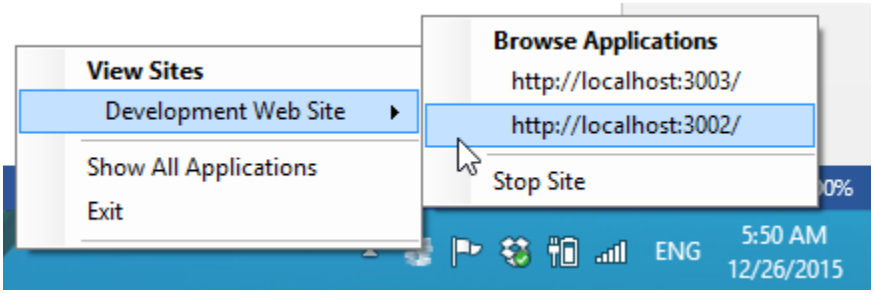
%ProgramData%\Syncfusion\DashboardDesigner\{Version}\IISExpress_DashboardService

Example:

C:\ProgramData\Syncfusion\DashboardDesigner\3.2.0.68\IISExpress_DashboardService

Hosting Dashboard Service in IIS Express

Dashboard Service is a web service through which data requests will be processed by Dashboard Designer application. This service will get hosted in IIS Express and run automatically on previewing a dashboard through Dashboard Designer.



Two sites, one representing the dashboard service and the other representing the dashboard preview will be running in IIS Express on clicking the Preview in the Dashboard Designer application. On closing the Dashboard Designer application, these started sites will get stopped automatically.

Upgrading Dashboard Designer

Syncfusion releases several major versions in a year. Each version includes new features, bug fixes and other improvements.

Syncfusion Dashboard Designer can be upgraded to latest version at any time manually, and there are no automatic updates for Syncfusion Dashboard Designer. Before upgrading, you can refer the features and enhancements from the Release Notes.

Upgrading Guidelines

- On any new release of Dashboard Designer, it can be downloaded and installed on top without uninstalling the previous version. The installation of Dashboard Designer will replace the previous installation and related files.
- You can always download the latest Syncfusion Dashboard Designer from here and follow the installation steps from the above section Installing Dashboard Designer.

Information: The latest versions of the Dashboard designer are backward compatible. You can refer the KB article for more details on Syncfusion Dashboard Designer application version compatibility.

Uninstalling Dashboard Designer

You can uninstall the dashboard designer application, if required, through uninstalling the Syncfusion Dashboard Designer entry from Control Panel.

Troubleshooting the Dashboard Designer installation errors

If you find any difficulties while installing the Dashboard Designer application in your machine, follow the given steps:

1. Ensure whether your machine meets the minimum system requirements for installing the Dashboard Designer application.

2. Check whether any other software installation or anti-virus blocks the Dashboard Designer application installation.
3. Check whether you have enough permission to install applications in the machine.
4. Collect the log files by following the steps mentioned in the KB article.
5. Please contact us by creating a support ticket and attach the log files mentioned in the step 4 with your basic machine information.

Self-help Resources

The Syncfusion Dashboard team provides different self-help resources for making the dashboard creation process simpler and quicker. You can get quick answers for your basic queries about the features from existing resources such as KB articles and forums.

Videos

The following page lists all available videos about getting started and using the Syncfusion Dashboard Designer application related features.

Videos Link: https://www.syncfusion.com/products/dashboard/videos

Frequently asked questions

You can instantly find answers for the most frequently asked questions about the Syncfusion Dashboard platform through the KB articles published by our Syncfusion team and public forums.

Syncfusion Knowledge Base Articles: https://www.syncfusion.com/kb/dashboard

Syncfusion Forums: https://www.syncfusion.com/forums/dashboard

Syncfusion technical blogs

Refer to the technical blog website to get the detailed technical blogs about Syncfusion Dashboard features.

https://blog.syncfusion.com/

Release history

The release information of all the public versions of the Dashboard platform can be referred in the following link.

https://www.syncfusion.com/products/release-history/dashboard

Create a support incident

If you are still not able to find the information that you are looking for self-help resources mentioned above, please contact us by creating a support ticket.

While creating the support incidents, try to provide detailed explanation of your requirement with necessary screenshots or videos. For reporting any defect, consider providing the exact replication procedure and necessary log files. It will help our support team to reach you earlier with a better solution.

Getting Started with Dashboard Designer

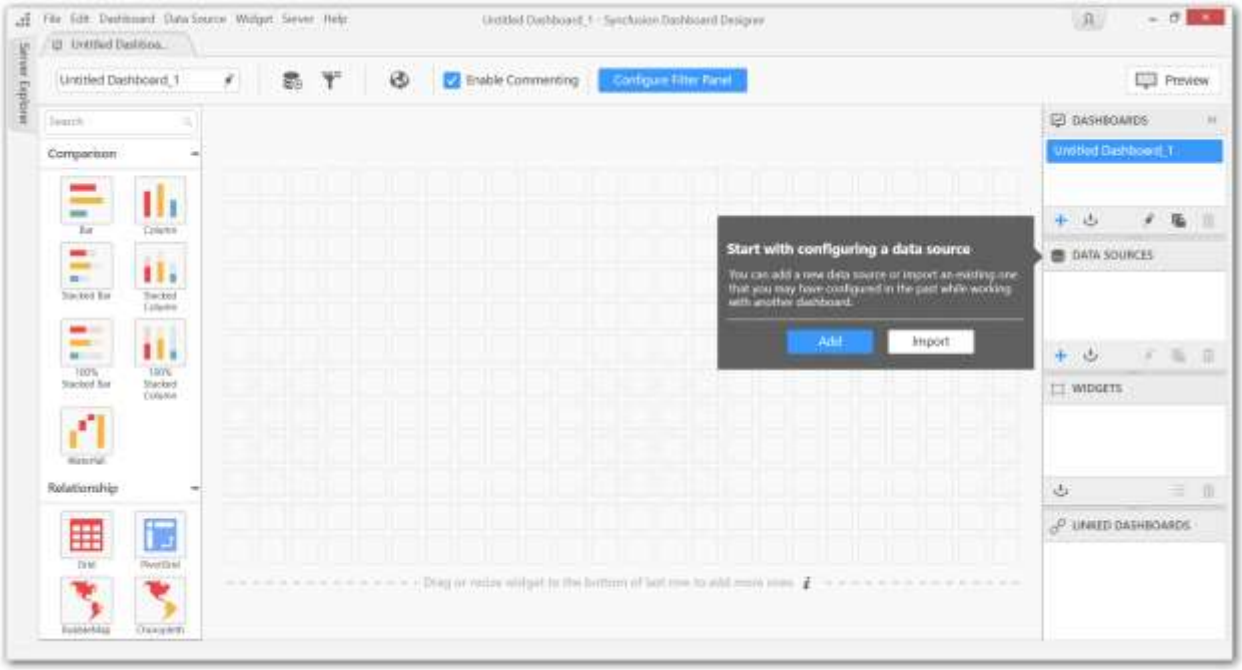
This is a simple walkthrough to get you started with Dashboard Designer application. Throughout this walkthrough, the Northwind database is used to demonstrate each feature of the Dashboard Designer.

Running Dashboard Designer

Launch the Dashboard Designer application by double-clicking the shortcut icon in the desktop.

Alternatively, you can launch the Dashboard Designer application from the Apps list.

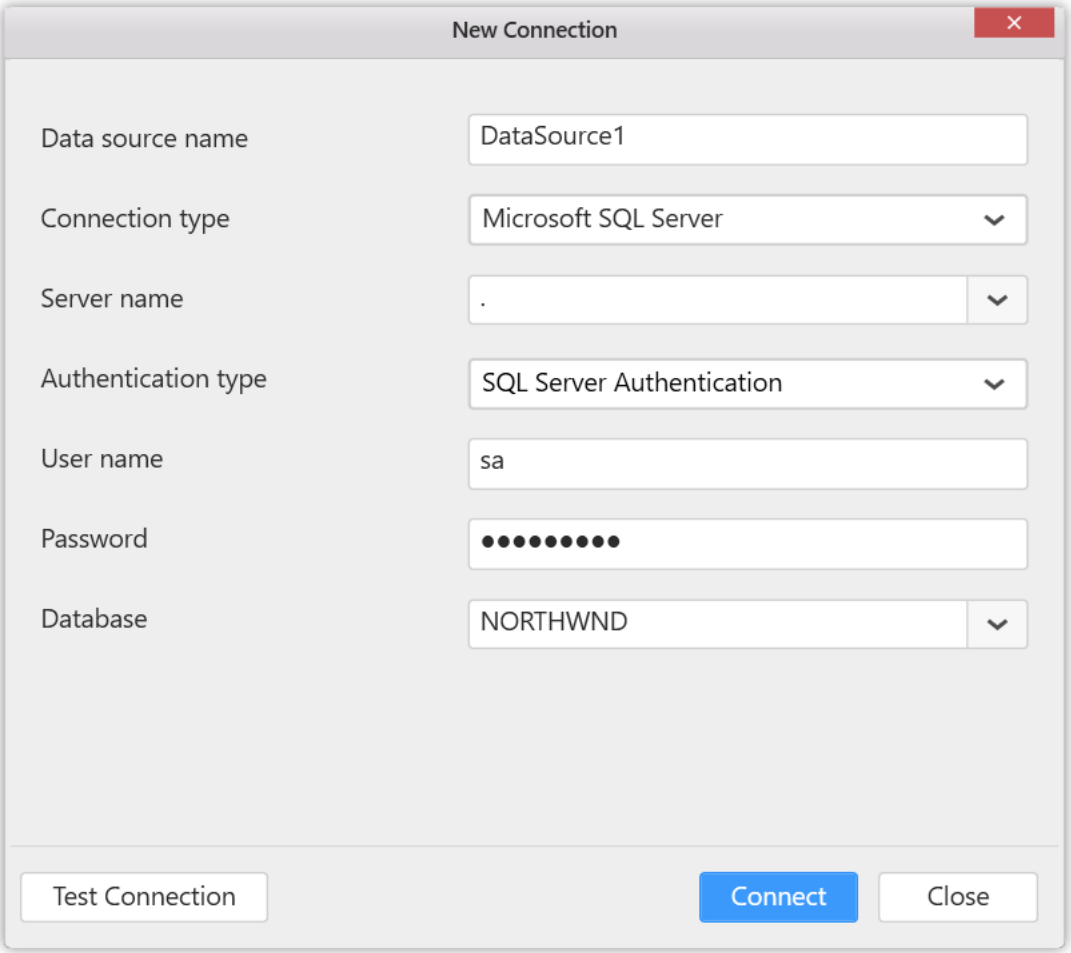
After launching, the Dashboard Designer application opens with the following view.



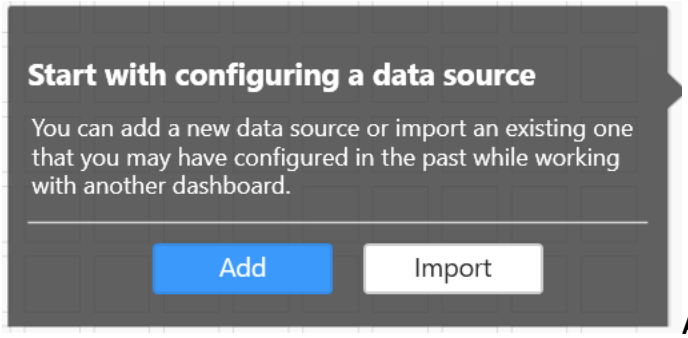
Connecting to data

Now, add a new data source after establishing a data connection with one of the supported data connection types as follows.

Setting up connection



Click



Add in the data source configuration smart screen window to launch a New Connection configuration dialog.

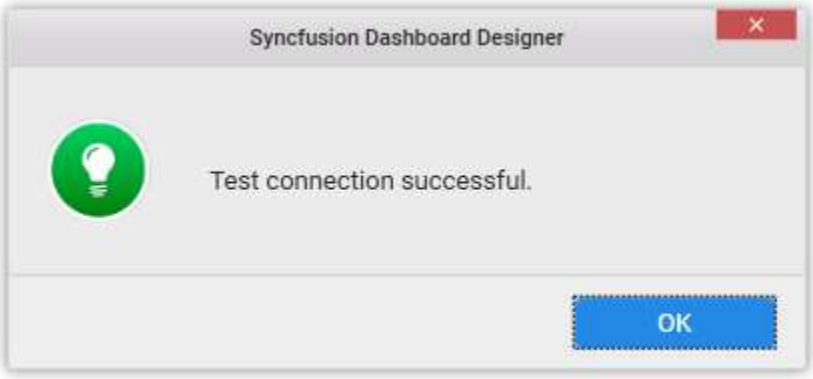
![Add new data source](images/newdatasource.png)

Configuring tables and views

In the New Connection configuration dialog, fill the connection type and required details.

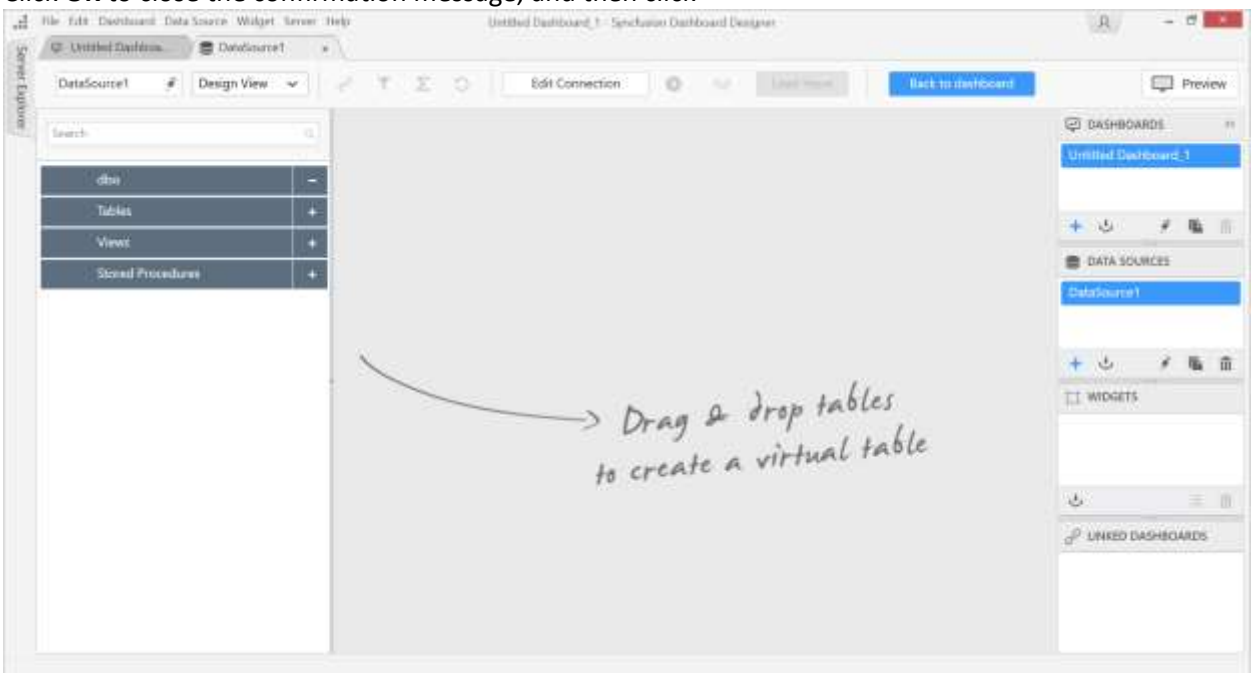
![New connection](images/newconnections.png)

Test the connection for its validity by clicking the



Test Connection button. The following confirmation message will be displayed.

Click **OK** to close the confirmation message, and then click



Connect in the New Connection dialog. The following view will be displayed.

The left pane holds the tables and views associated with the connected database. Drag your preferred

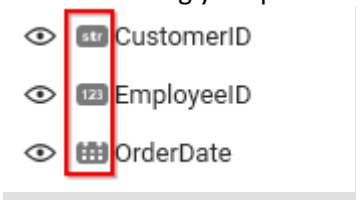
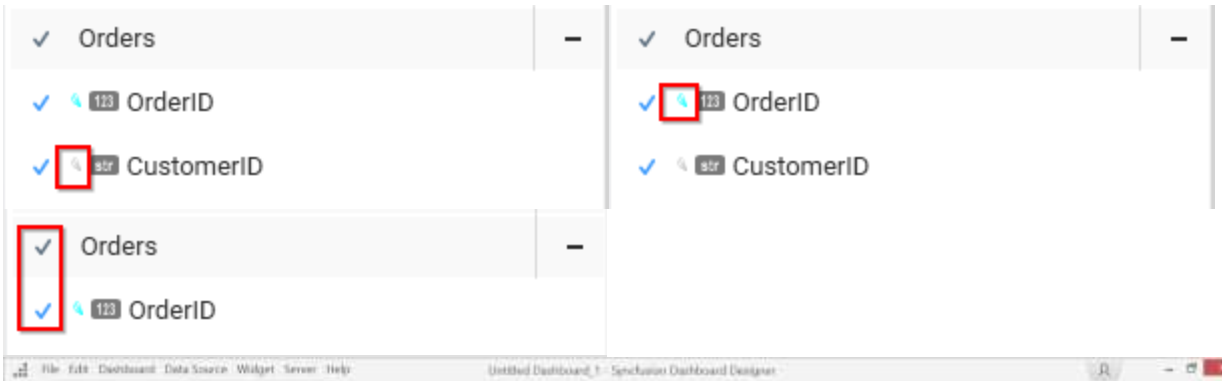
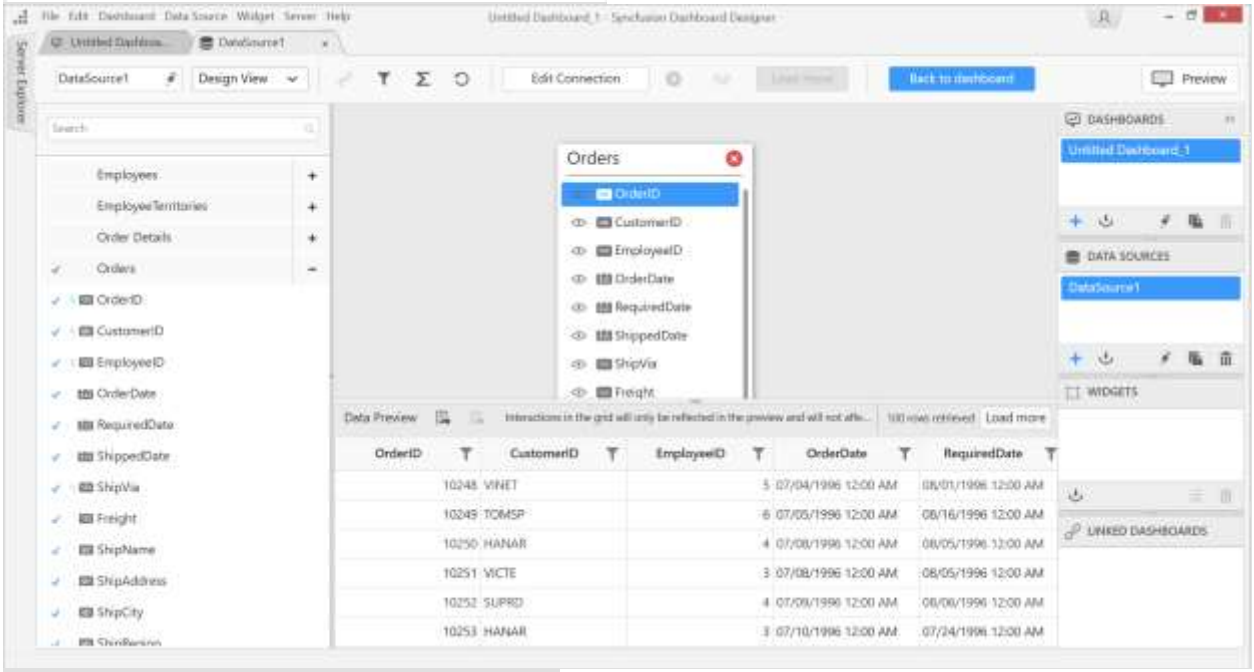


table or view from the left pane and drop into the center pane labeled





Drag & drop tables to create a virtual table as follows.

![Virtual table](images/virtualtable.png)

The dropped tables will be remarked by a tick mark before the name of both table and its columns in the left pane as follows.

![Tickmark of table](images/tickmarkoftable.png)

The primary key defined in the table in the connected database will also be remarked.

![Primary key](images/primarykeymarked.png)

Similarly, the foreign key will be checked as follows.

![Foreign key](images/foreignkey.png)

The data type of each column will be represented nearby as follows.

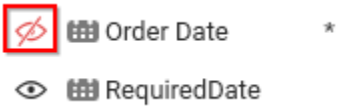
![Data type icon](images/datatypeicon.png)

Add more than one table, if you prefer, through the following same drag n drop operation. This is subjected to joining of tables.

Transforming data

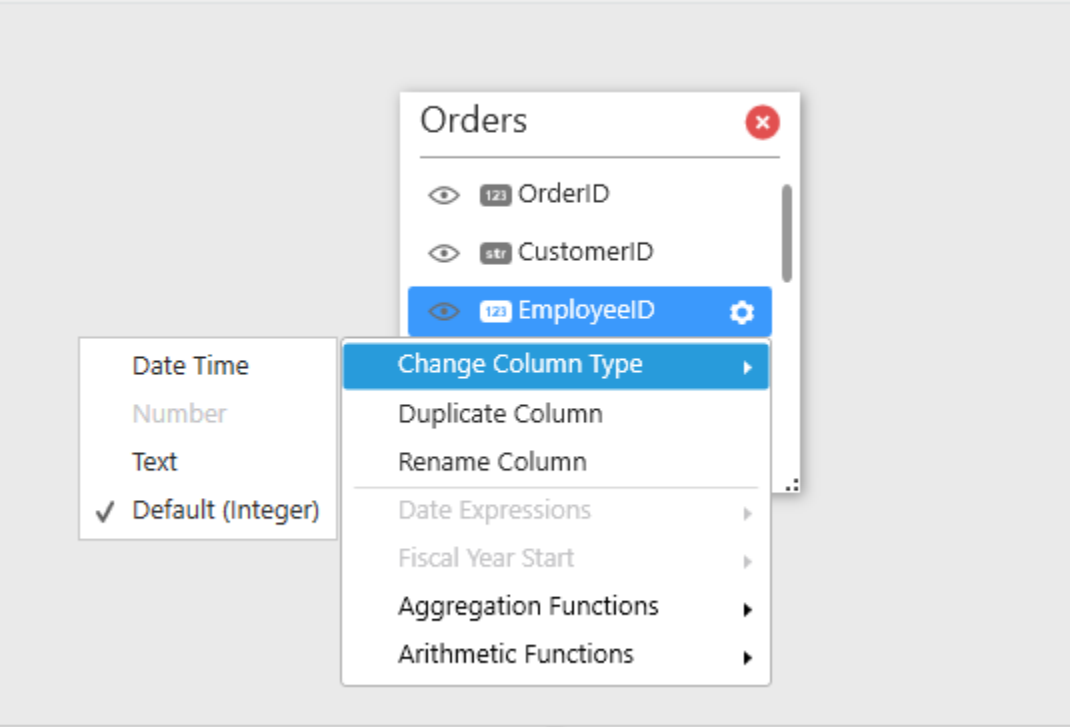
Rename the column as required, either by selecting the options in the Settings drop-down menu or double-clicking the column to enable the edit mode.

Remove the unwanted column by clicking the icon at the left of the respective column.



The removed column will be represented with the red colored icon as above. Clicking the same, will re-include the removed column.

Handle column type conversion, if required, by clicking the **Settings** icon of the respective column and navigate to **Change Column Type** option. Select the preferred type to convert as shown in the following.



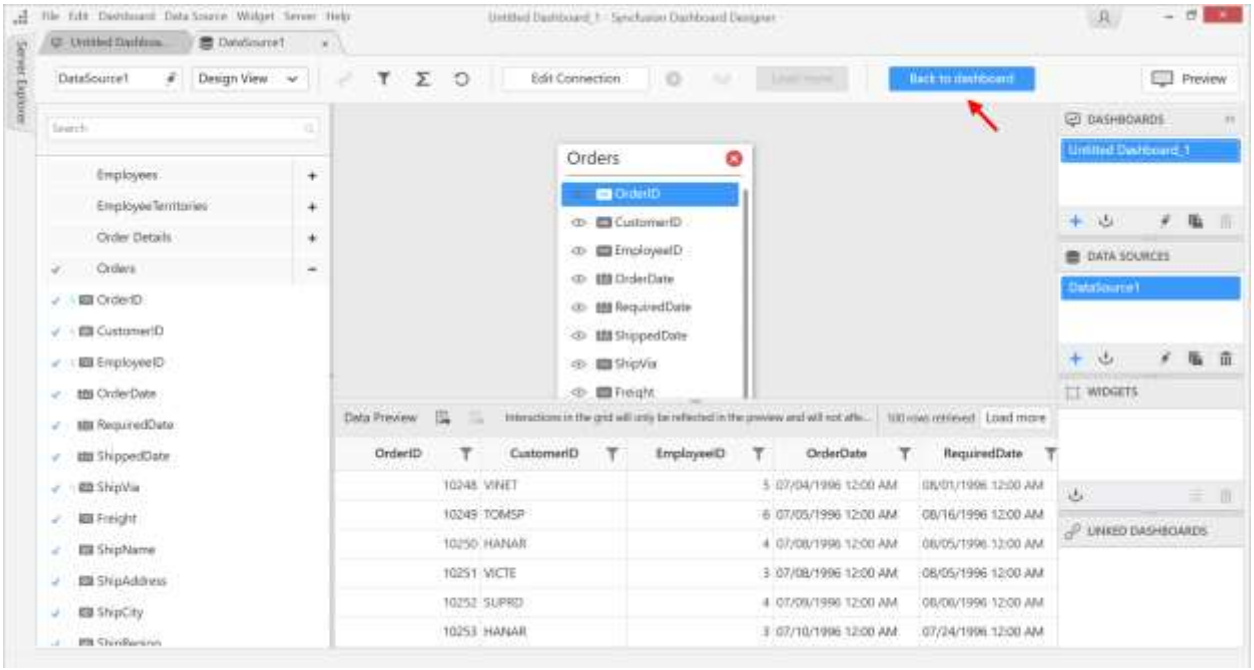
For supported column types and their equivalent convertible types, refer to Formatting Columns.

Add the desired expression columns by creating it using the built-in functions and existing columns.

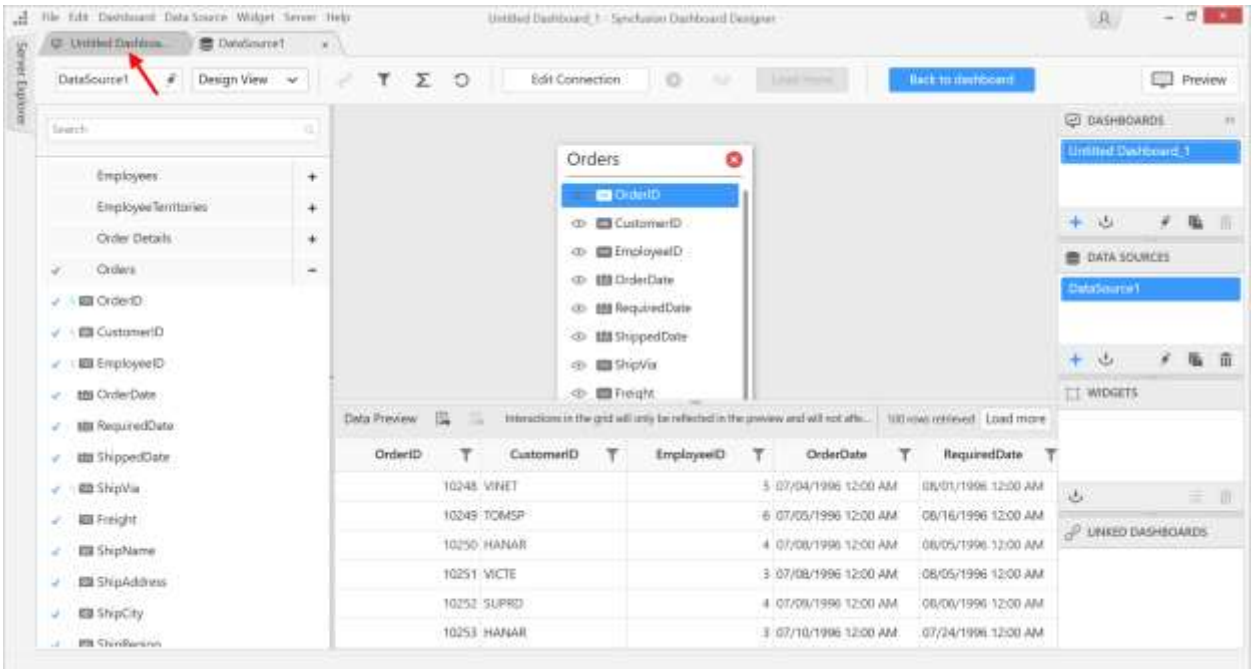
Filter the data that is not required in the dashboard using the data filters option in the tools pane in data design view as highlighted below.

Creating Dashboard

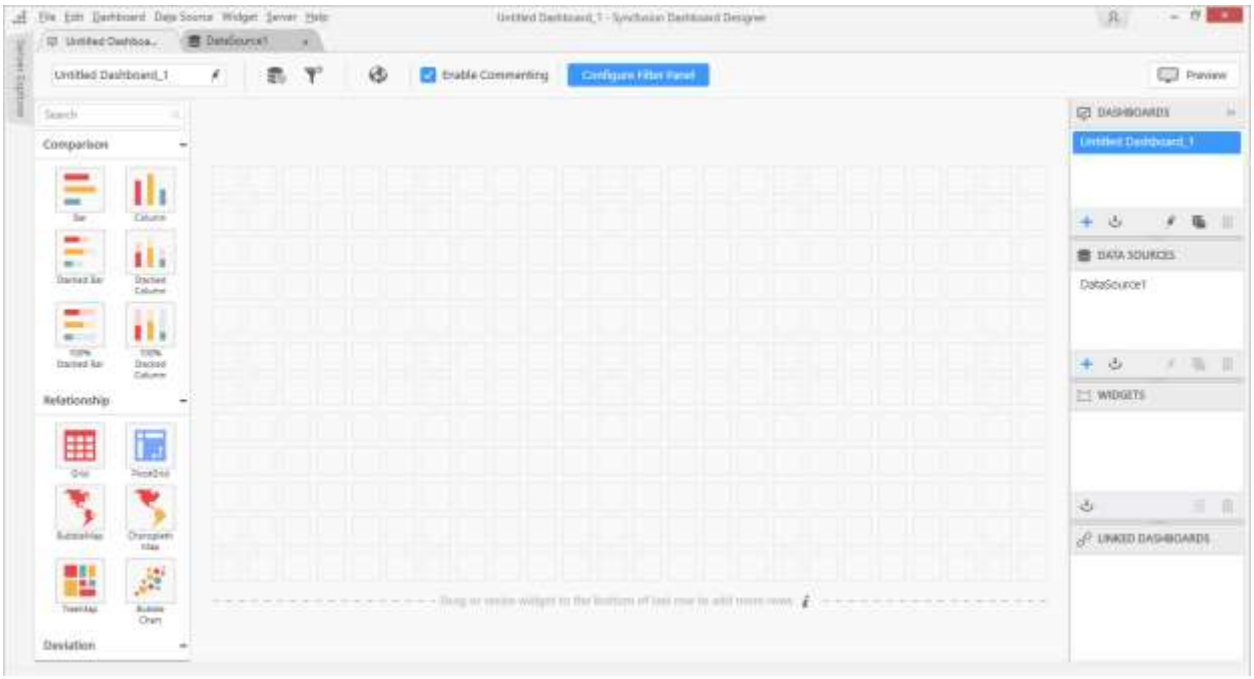
To create a Dashboard, click **Back to dashboard** and navigate to the dashboard design view.



You can also click the respective dashboard tab directly to navigate to its design view.

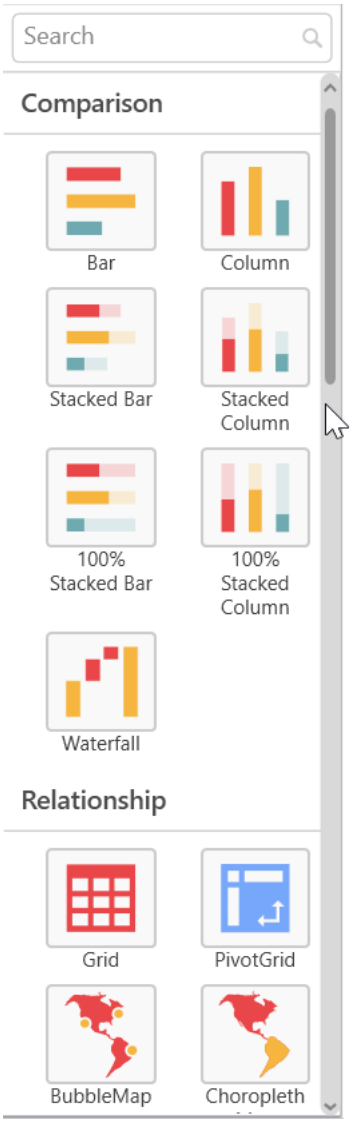


Now, the dashboard designer tab will open as follows.

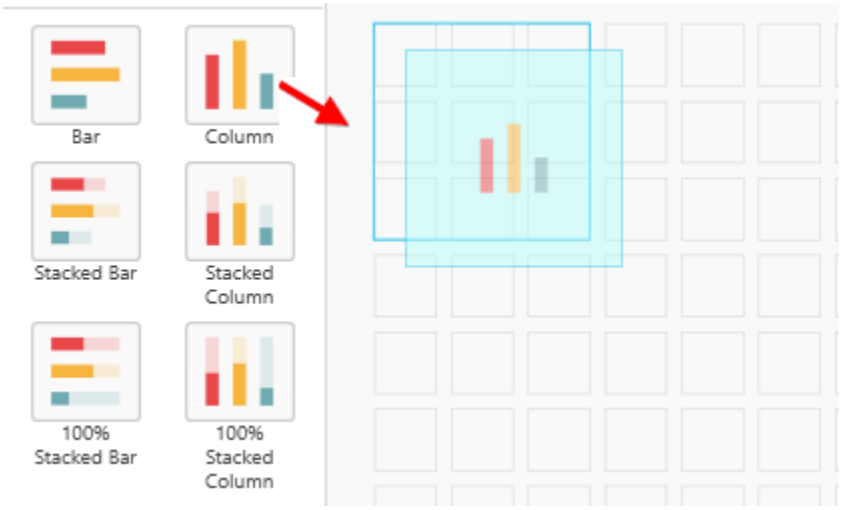


Adding a widget to design view

The left toolbox pane consists of data visualization, filter, and miscellaneous widgets to design an interactive dashboard.

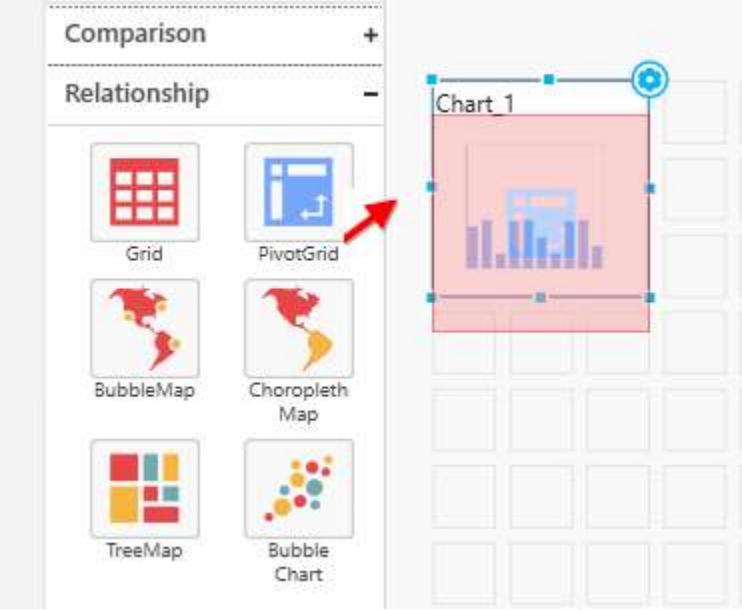


Click and drag the preferred widget from toolbox and drop it in the available space of the design area.

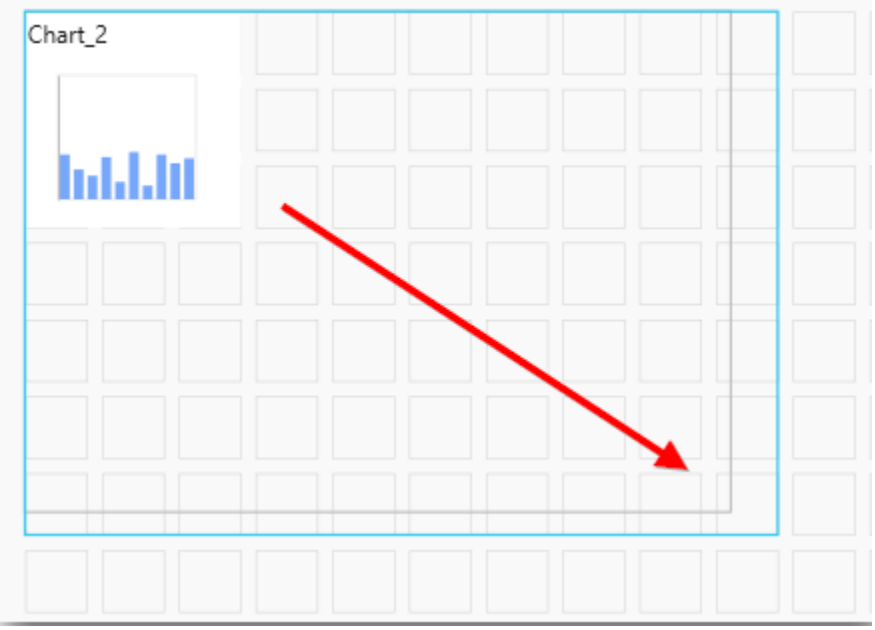


The widget drop will happen only when you drop it in the appropriate region. In the above image, the blue border of the cell indicates that the targeted region is valid to drop.

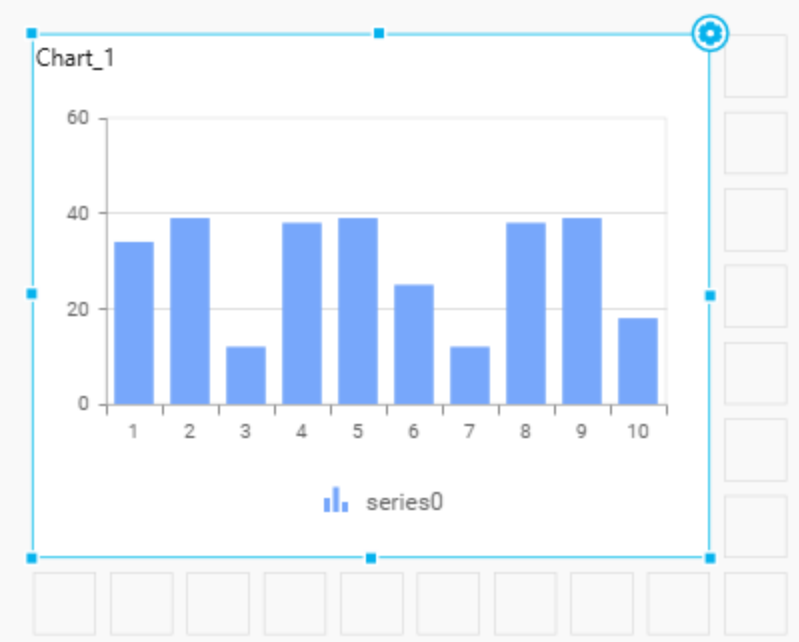
The invalid region will be represented by a red border. This happens when you drop over a region where a widget or its part already exists.



You can resize the dropped widget by placing the focus over the widget and dragging the widget corner as follows.



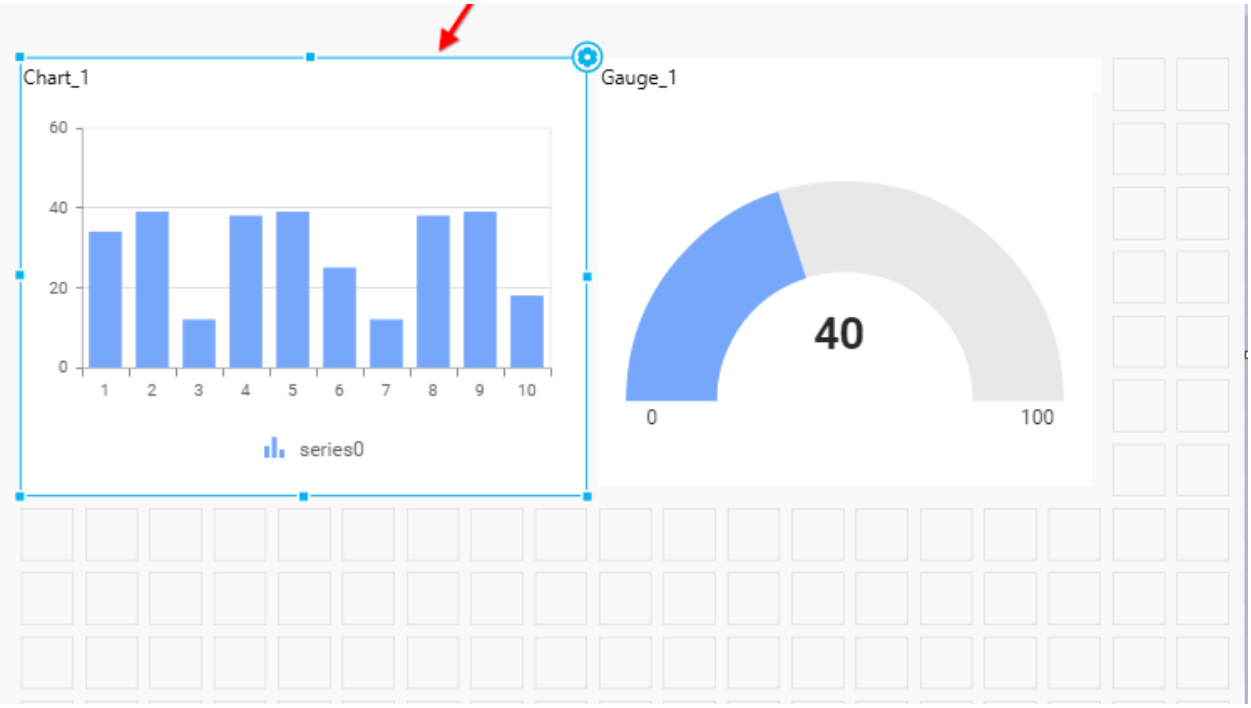
Doing so will render the widget to the size you dragged. Here, the blue border indicates the occupied cell range of widget after resizing.



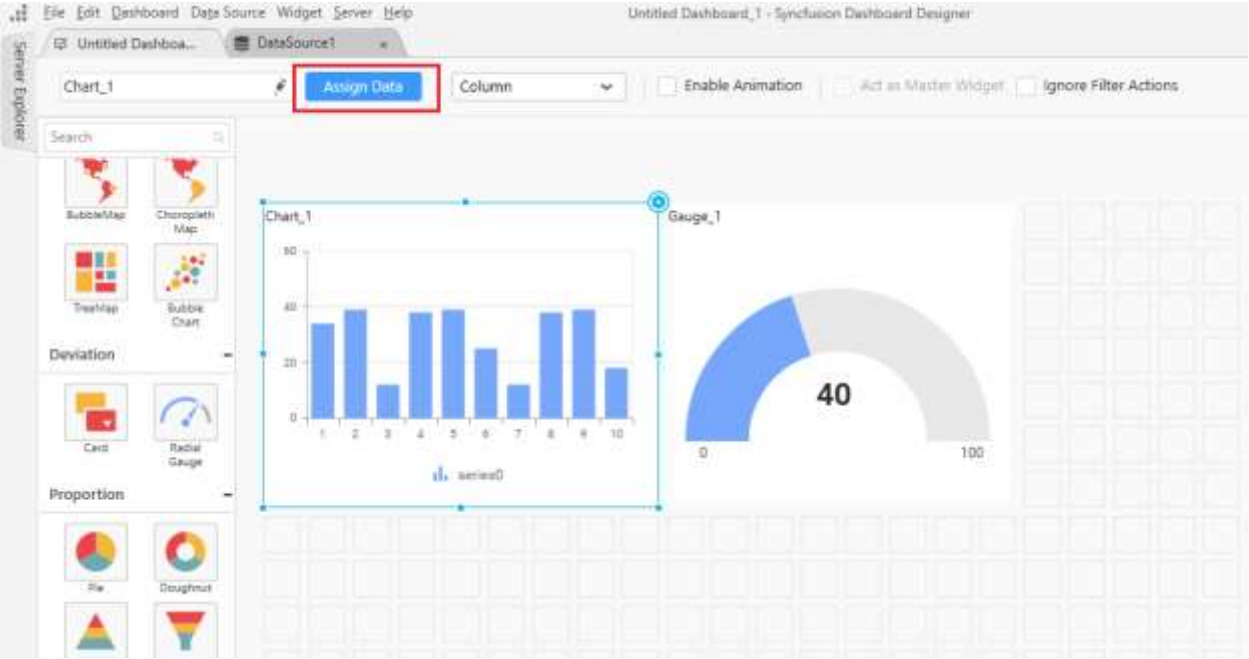
Assigning data to widget

Note: This step is applicable only for widgets that do not belong to miscellaneous category.

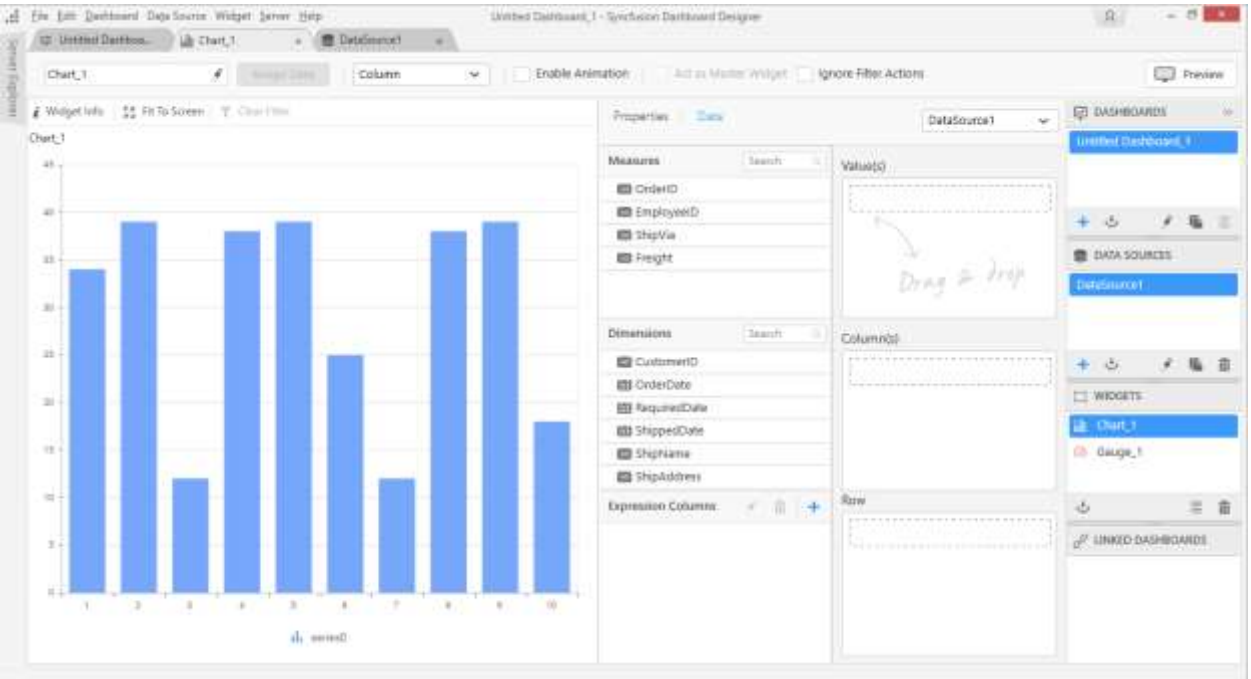
To bind data to a widget placed in the design area, focus that widget.



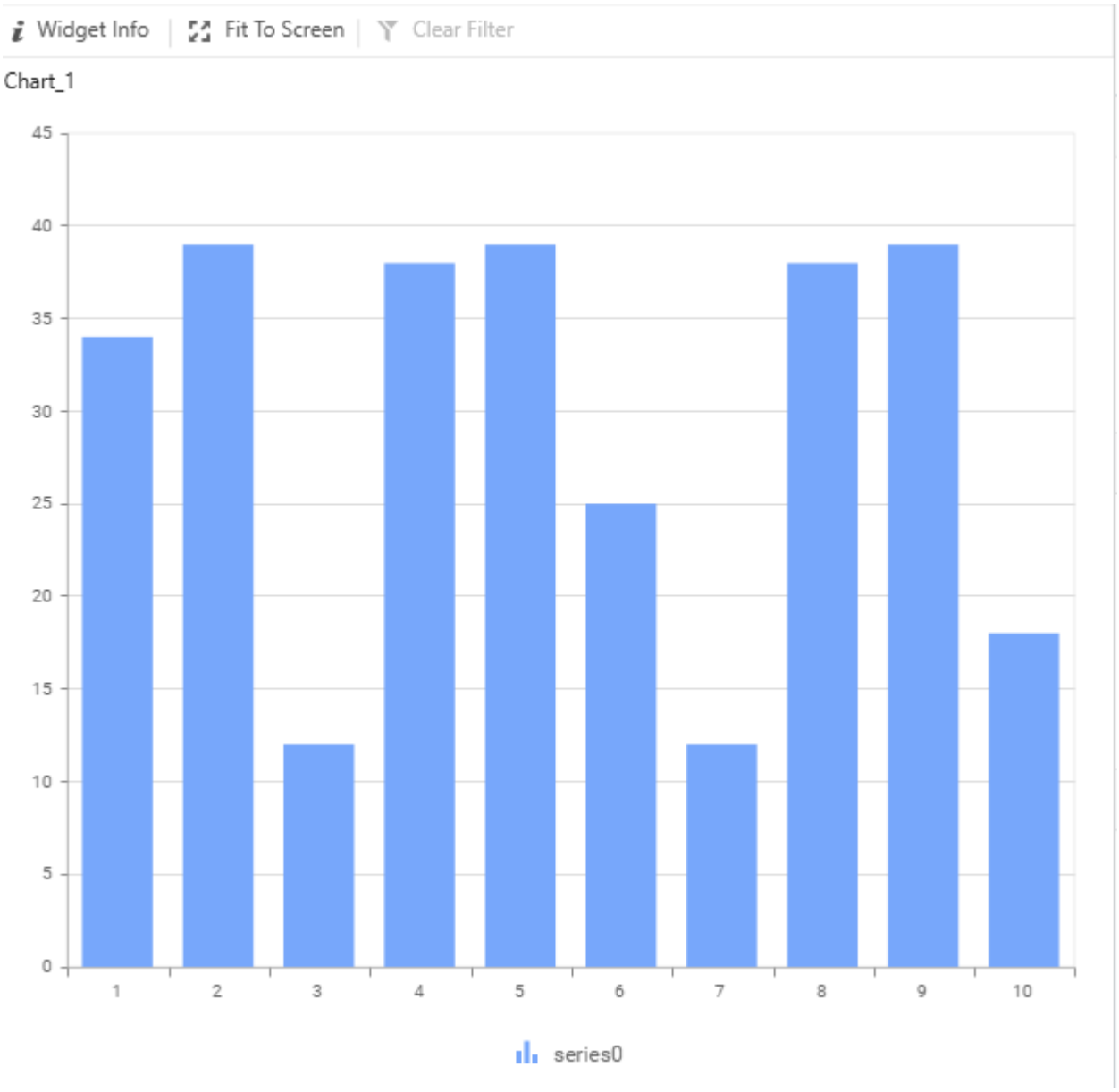
Click the **Assign Data** button in the design tools pane.



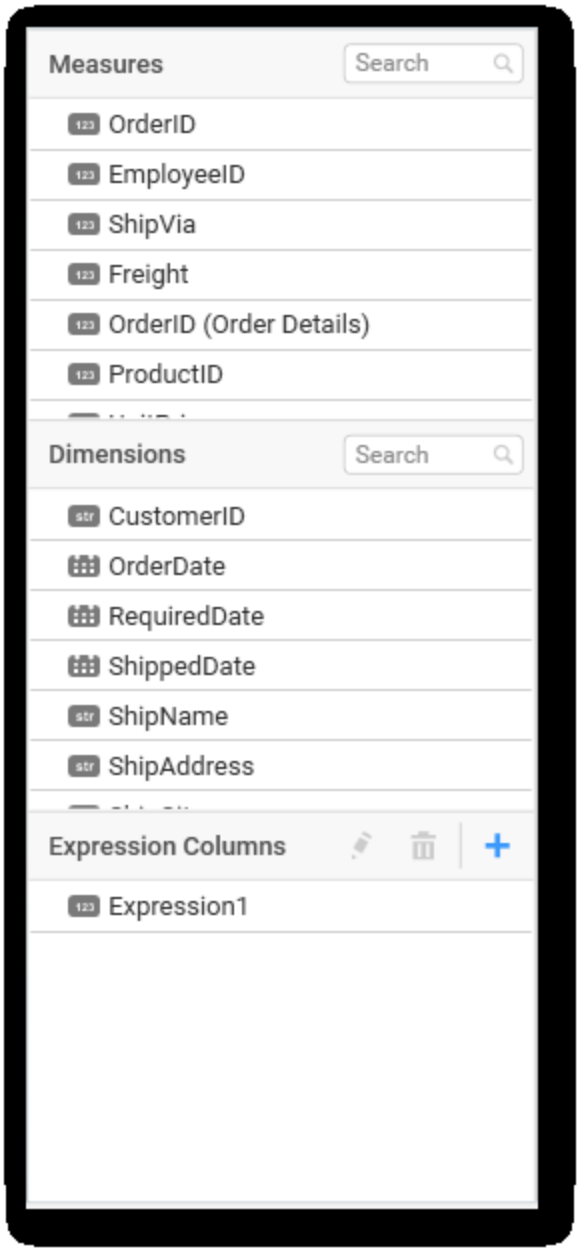
Now, the widget view opens in a new tab.



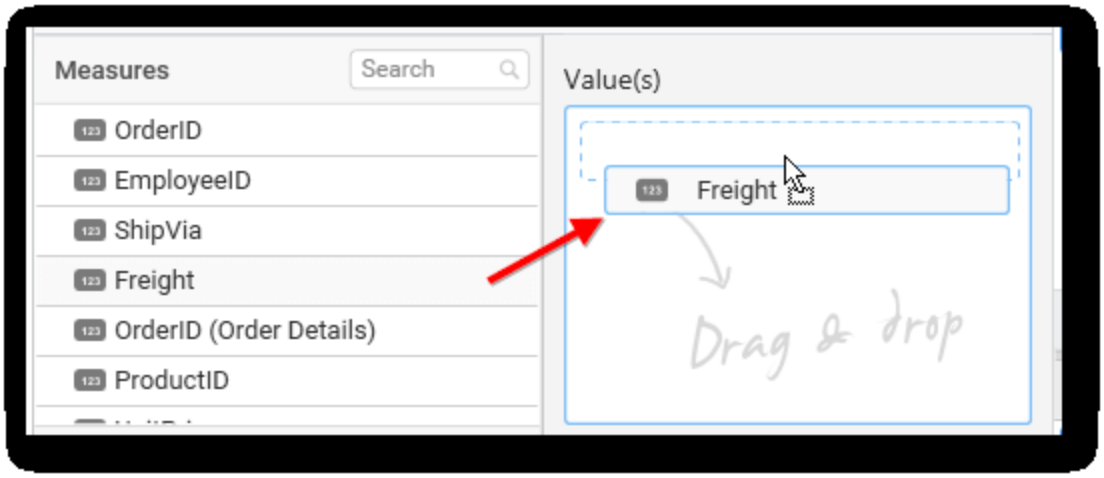
Here, the left pane holds the widget preview that reflects the configuration changes made to the widget.



Right pane holds data configuration view. The numeric columns get listed under the Measures section; other type columns are listed under the Dimensions section; the expression columns are created dynamically and listed under Expression Columns section.

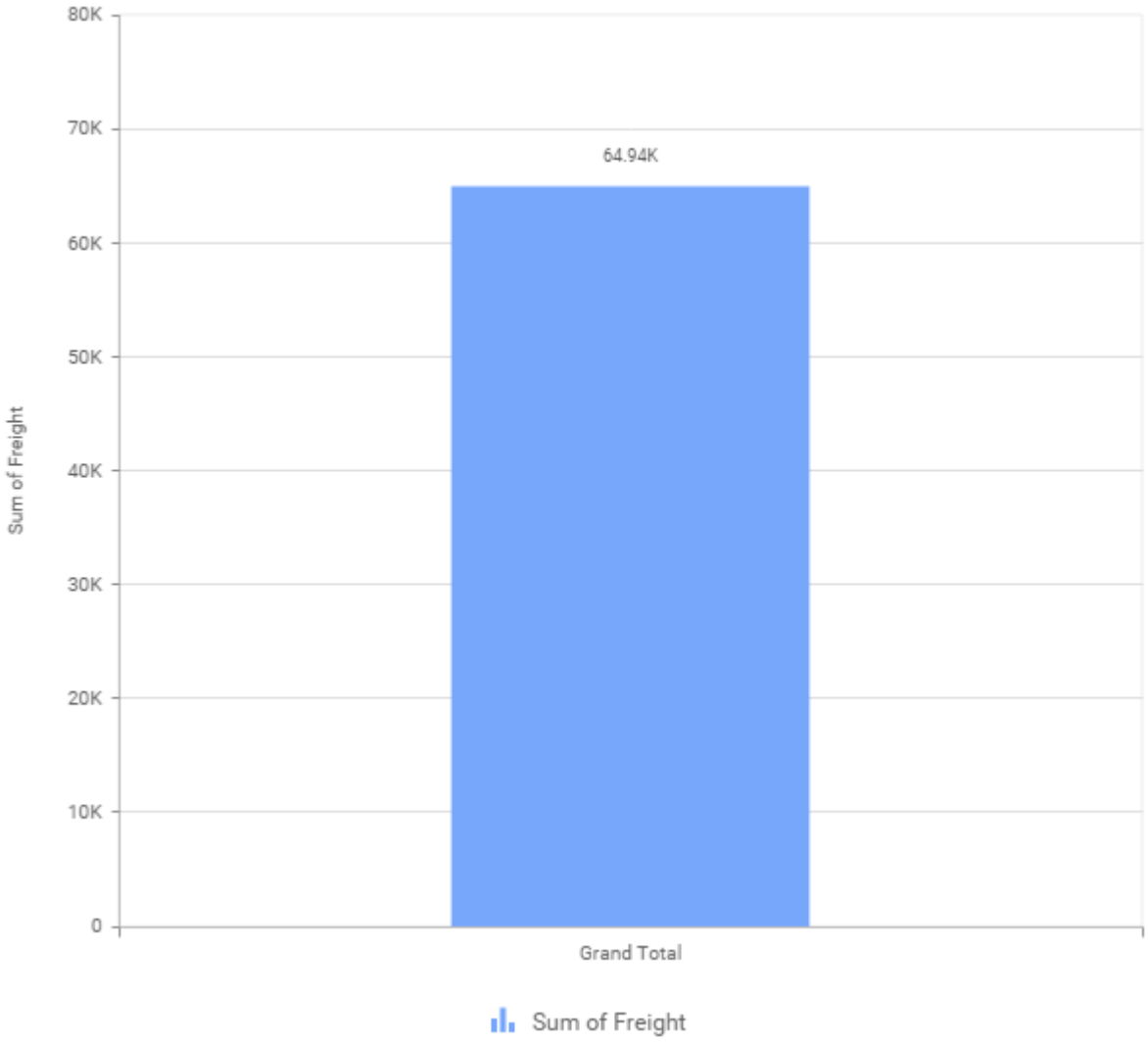


Select and drag the numeric column (measure element) from the measures section or the expression column (from Expression Columns section) to be measured and drop it to the Value(s) section.

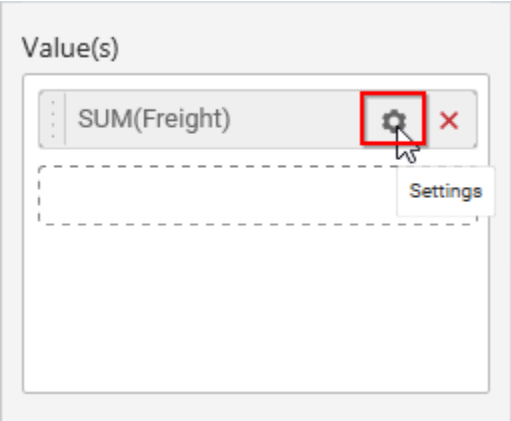


Now, the widget preview will be as follows.

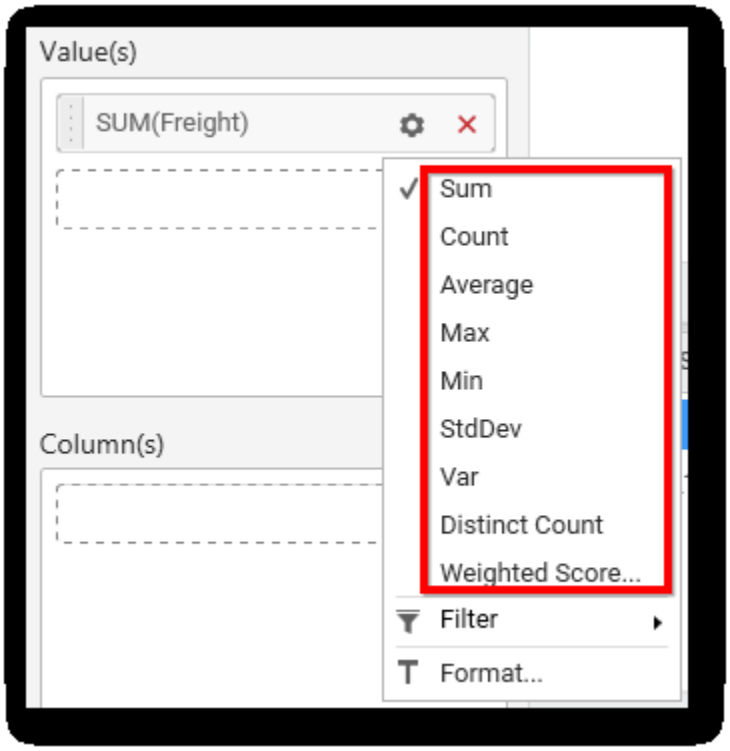
Chart_1



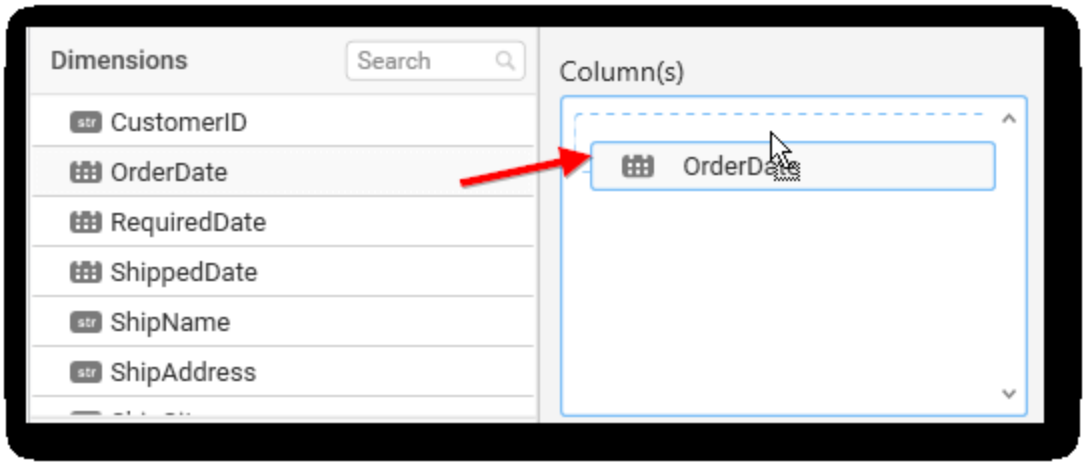
Click the **Settings** icon (highlighted) to open the aggregation type drop-down list.



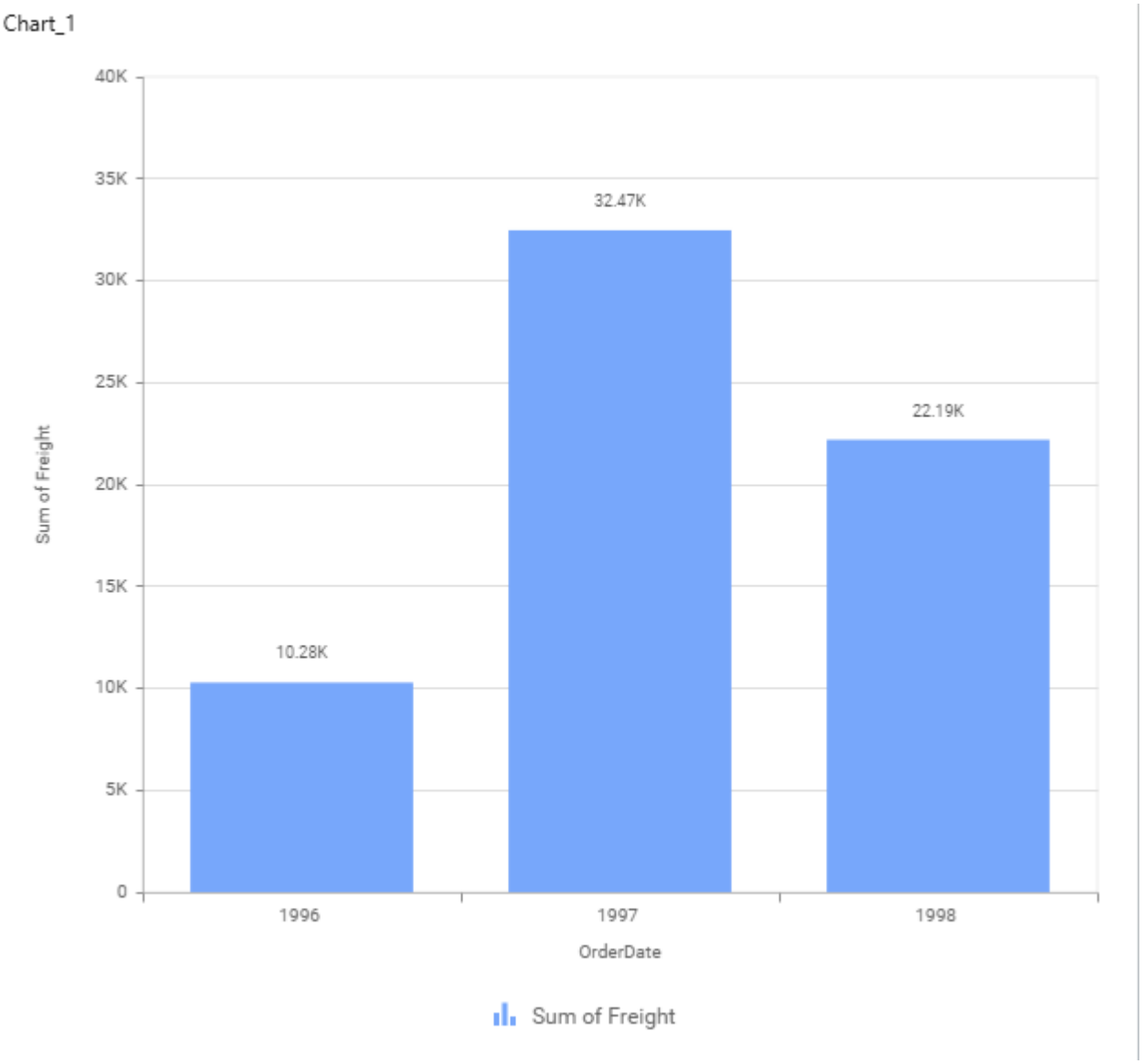
Set the preferred aggregation type to compute the dropped measure column.



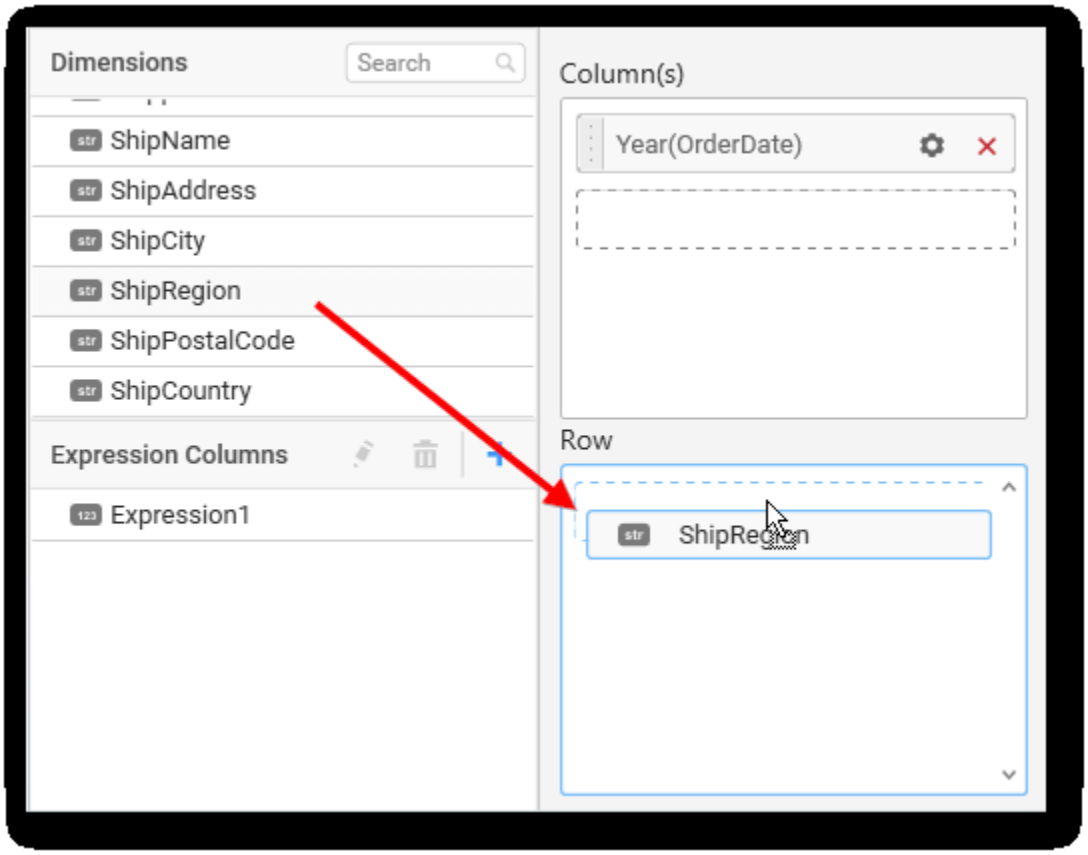
Select and drag the non-numeric column (dimension element) from the Dimensions section against which you need to measure the numeric column(s) dropped, and drop it to the Column(s) section.



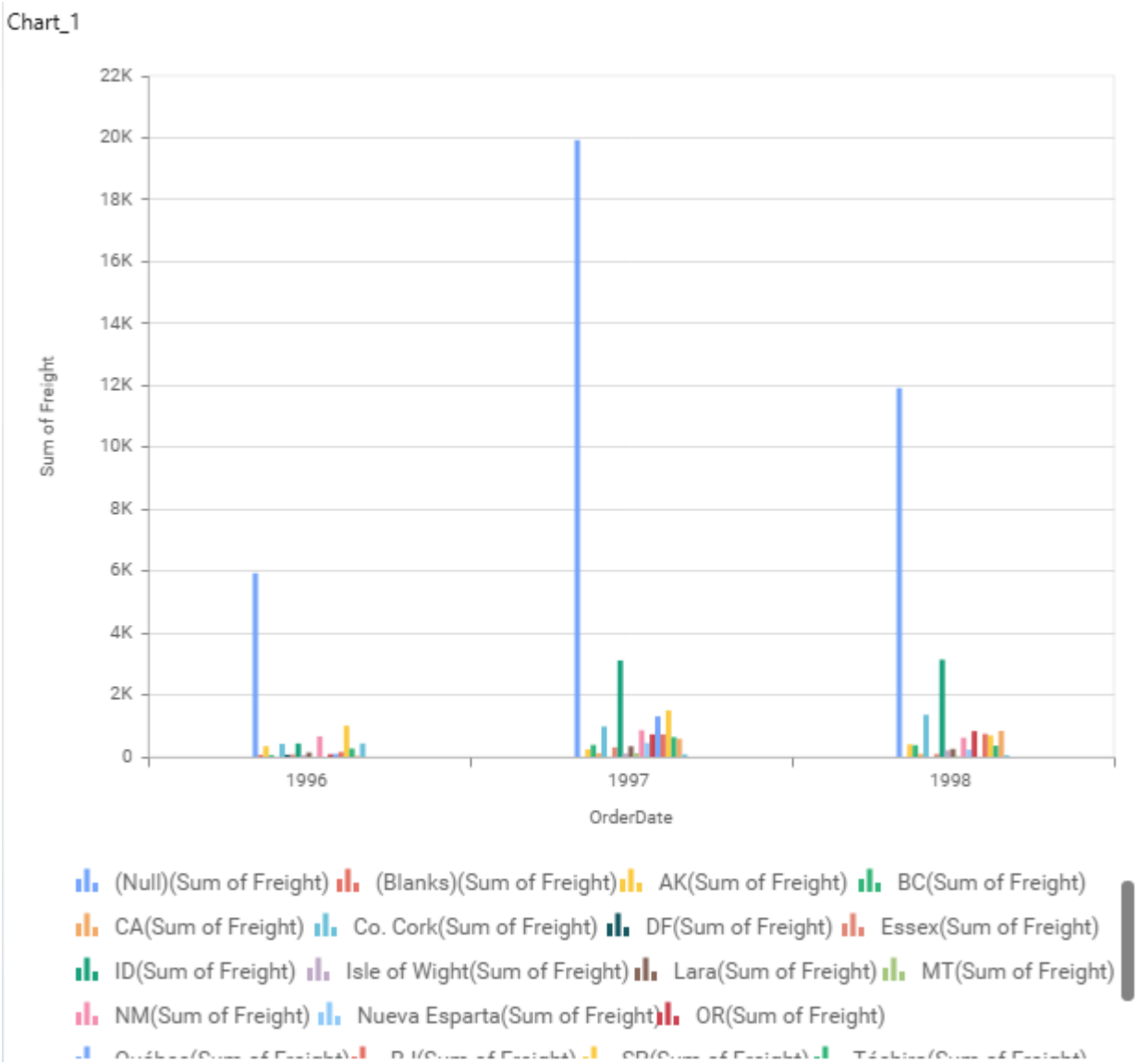
Now, the widget preview will be as follows.



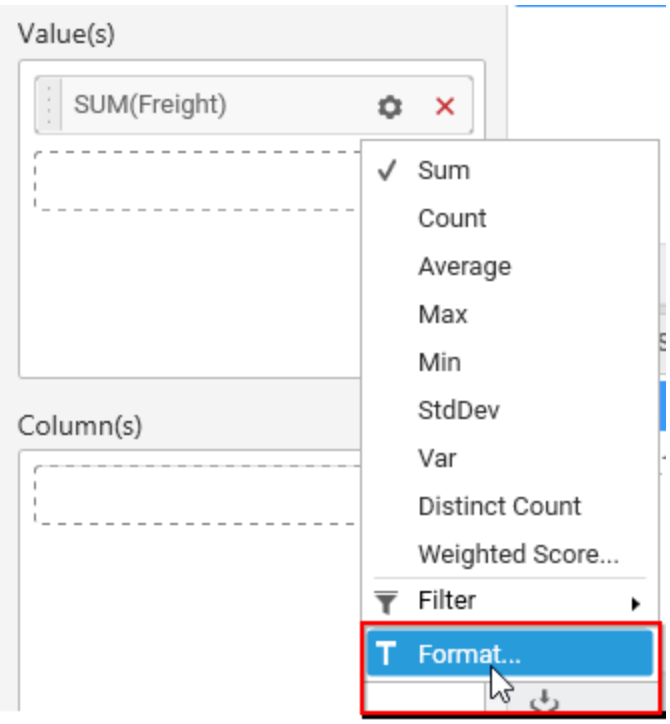
To group the added column elements by a column, add the respective non-numeric column (dimension element) into Row(s) section.



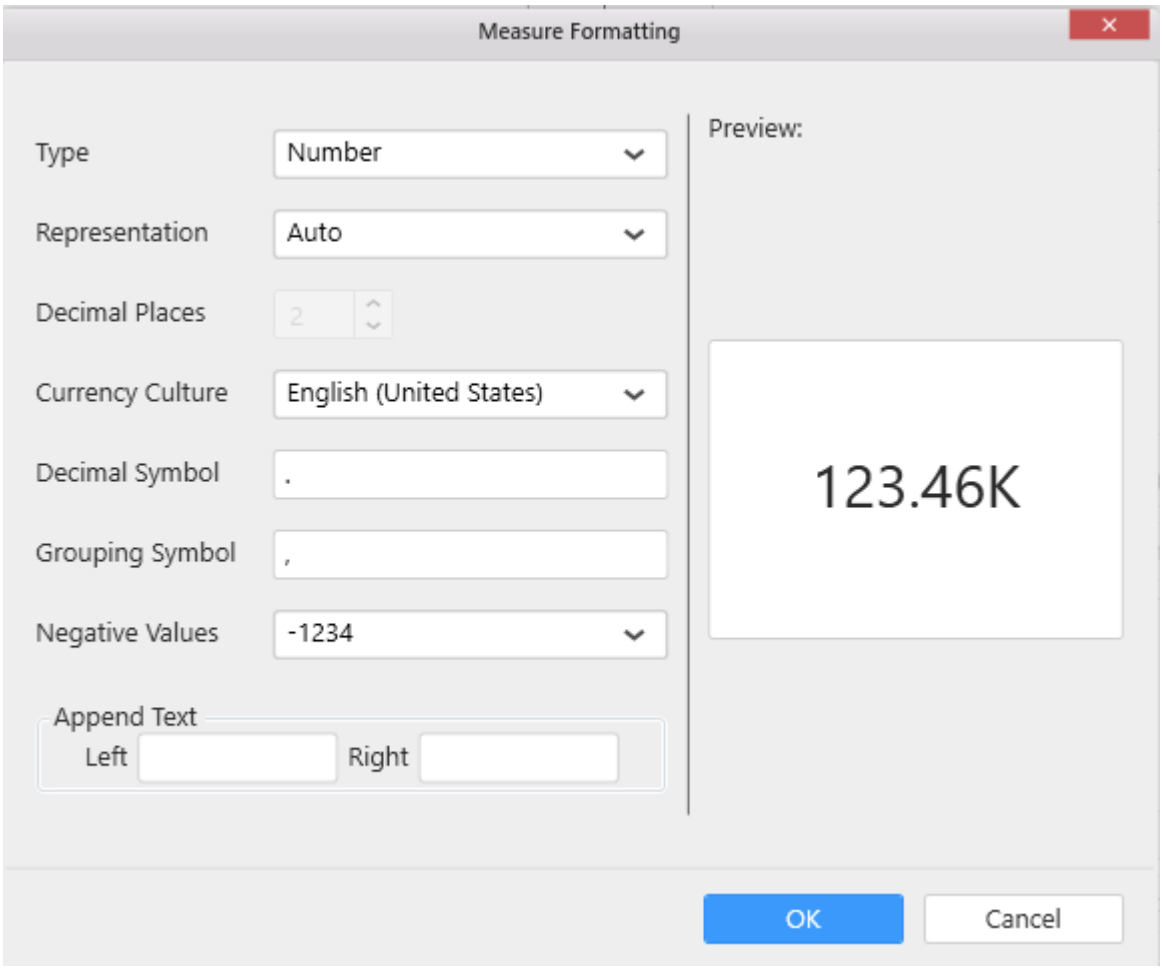
After the above, the widget preview will be as follows.



To format the measure column values, click the **Settings** icon to open the drop-down list and select the **Filter** option to open the Measure Formatting dialog.



Measure Formatting dialog:



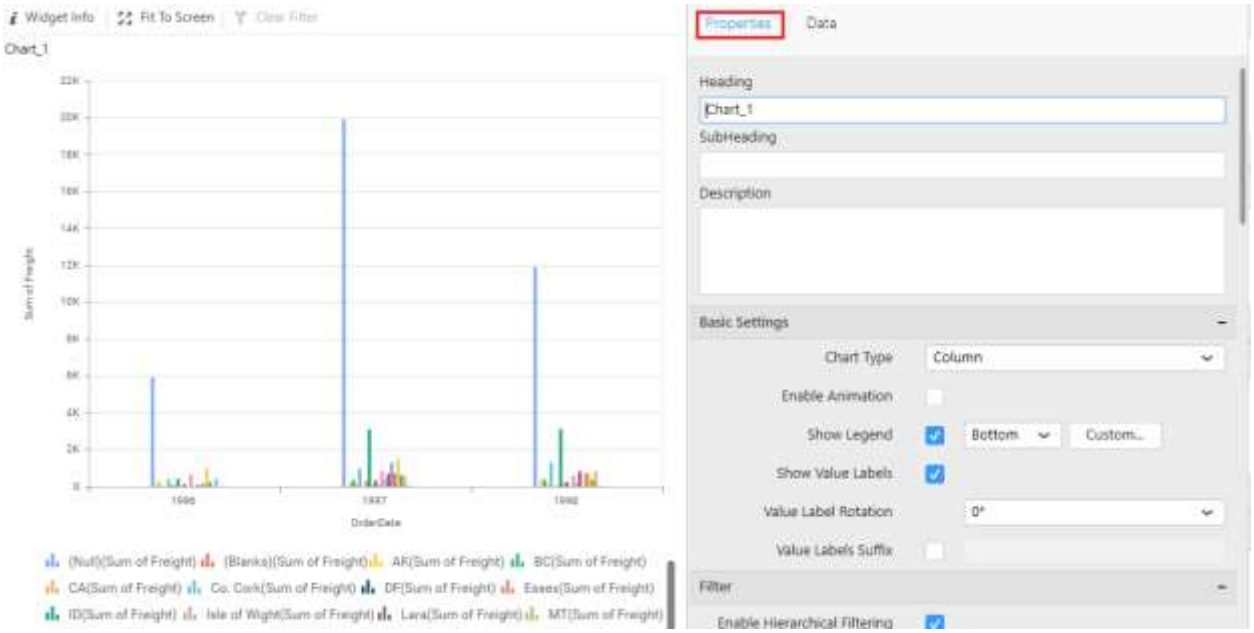
Make necessary changes and click **OK** to save pending changes. Click **Cancel** to cancel the unsaved changes and close the dialog.

If required, apply filter to the dropped measure type or dimension type column(s) through Measure Filter and Filters dialogs respectively. This filters widget data (from getting bound to widget) that does not meet filter criteria.

Configuring properties to widget

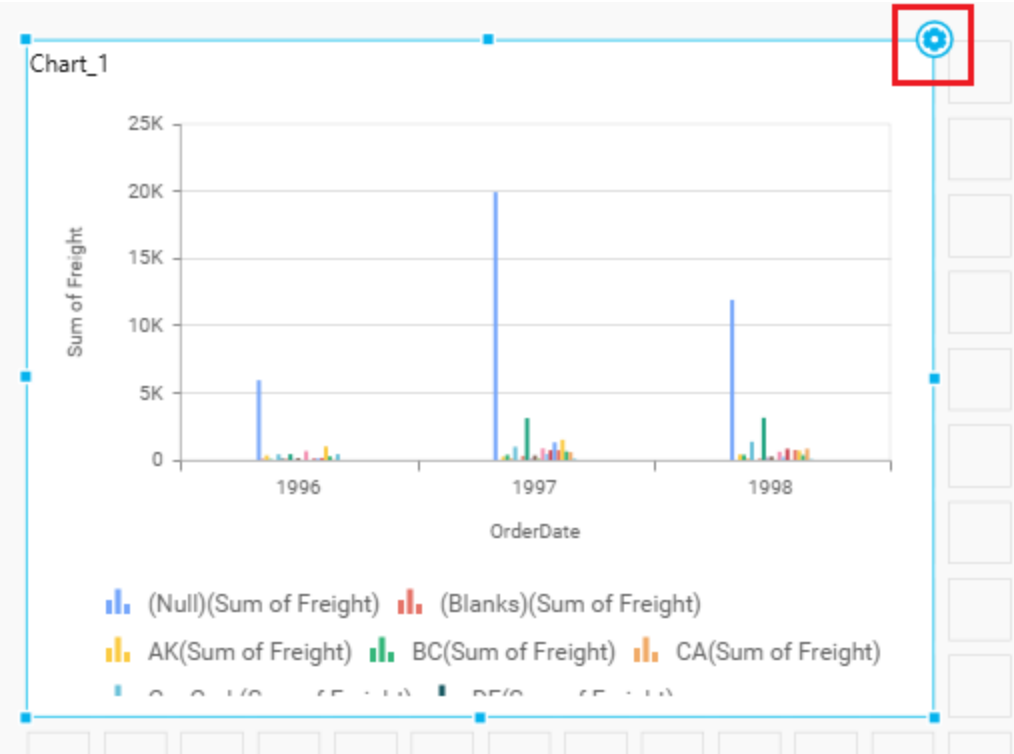
Note: This step is applicable only for widgets that do not belong to miscellaneous category.

Navigate to the Properties pane in the data design view tab.



From the dashboard design tab, you can navigate to the Properties pane by placing the focus in the widget as follows.

Click the **Settings** icon at the top-right corner of the widget. Focus moves to the data design view tab and opens the Properties pane.



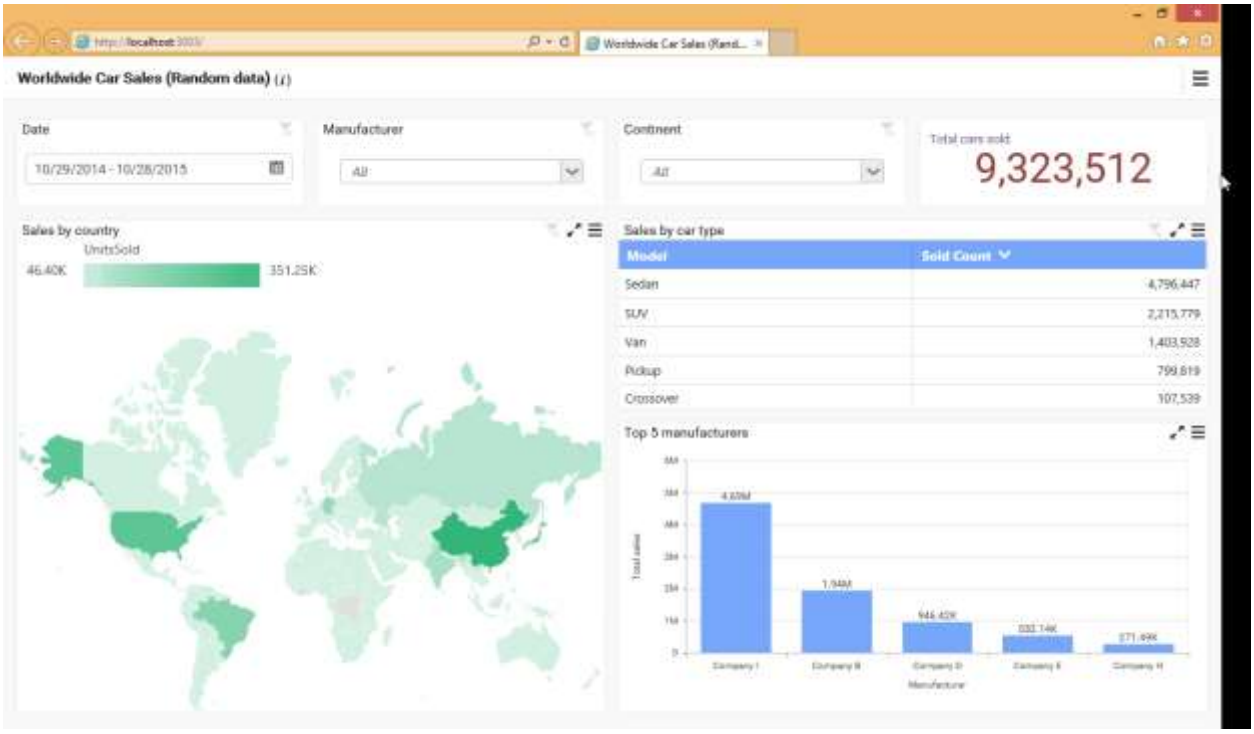
This pane holds some general settings and some specific to the widget. You can configure the desired settings. Refer to the Properties Configuration widget-wise for more details.

You can add more widgets by following the same procedure.

Once you are done with the dashboard, save the dashboard in local as SYDX formatted file.

Click **Preview** at the top right of tools pane to see the dashboard preview launched in the web browser page.

Now, the dashboard preview can be visualized through the built-in dashboard viewer in web browser page.



Sharing Dashboard

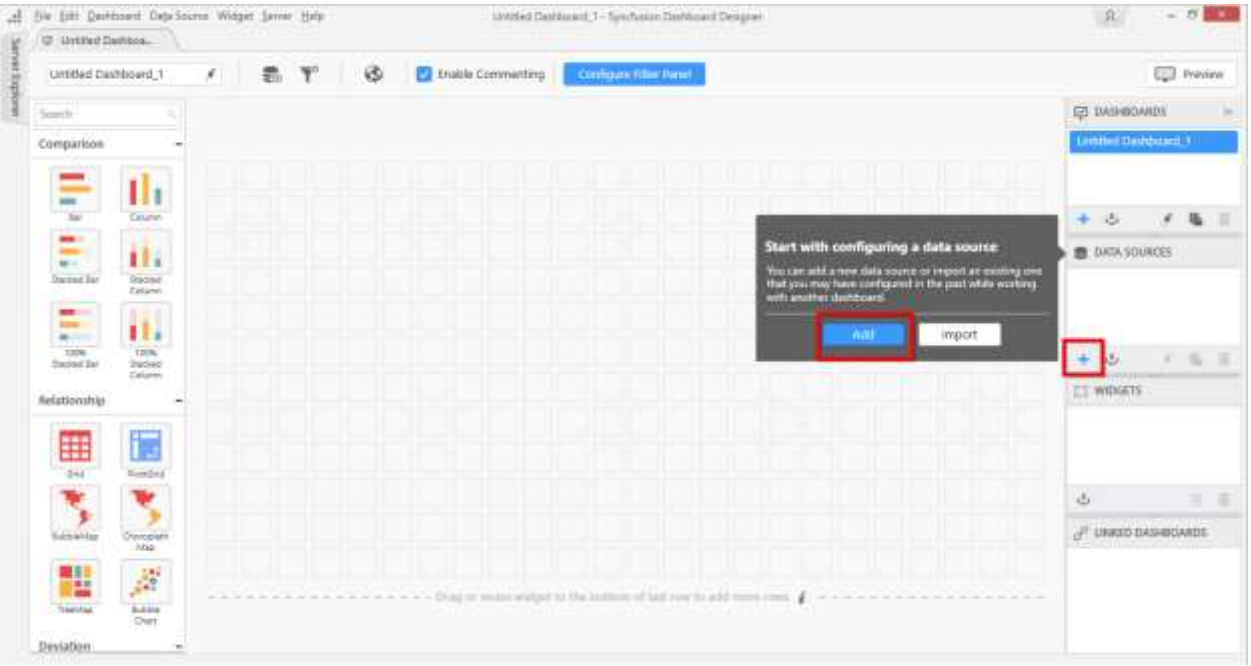
Publish the created dashboard to the Dashboard Server through authorized publish access from the Dashboard Designer.

Connecting to Data

Creating a new Data source

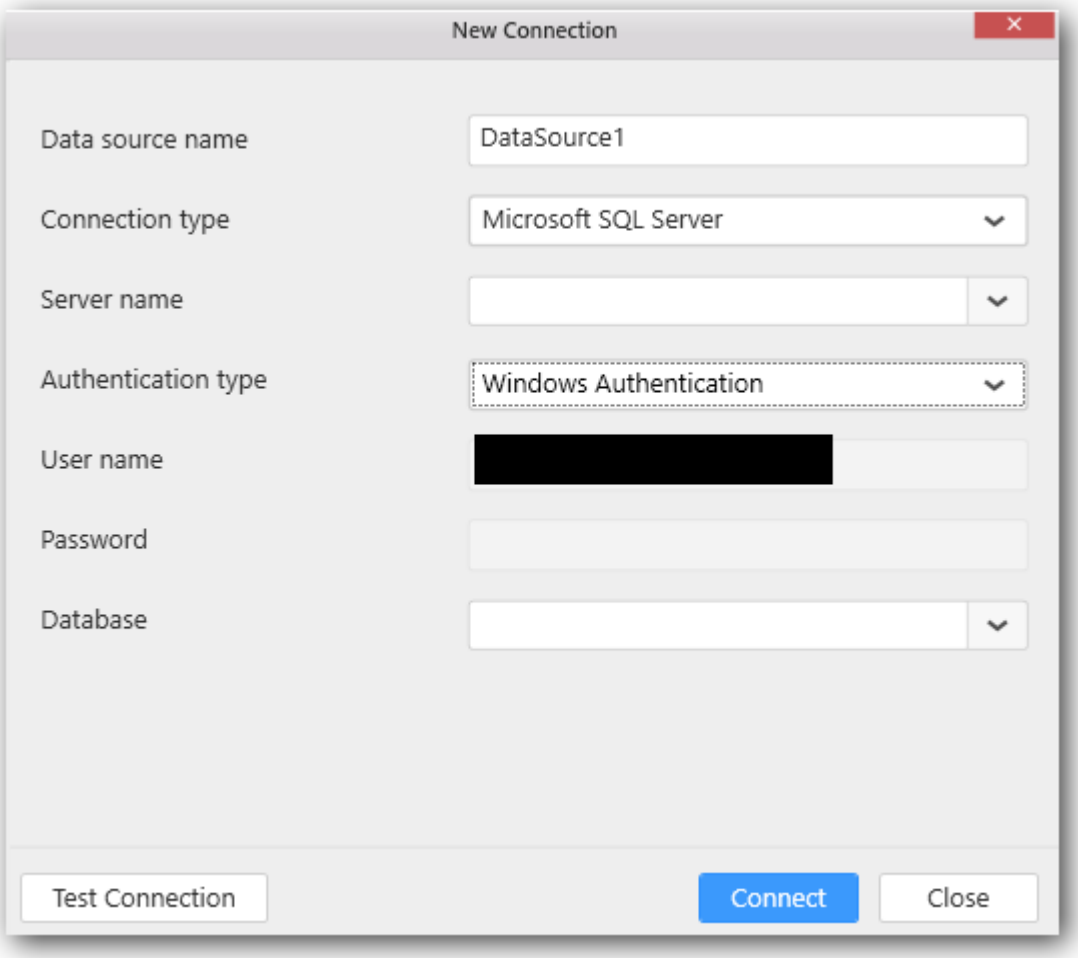
To bind data to a widget, a minimum of one data source is required. A data source can be created through the following procedure:

Go to the Dashboard Design View and click the + button at the bottom of DATA SOURCES container as indicated in the below image.



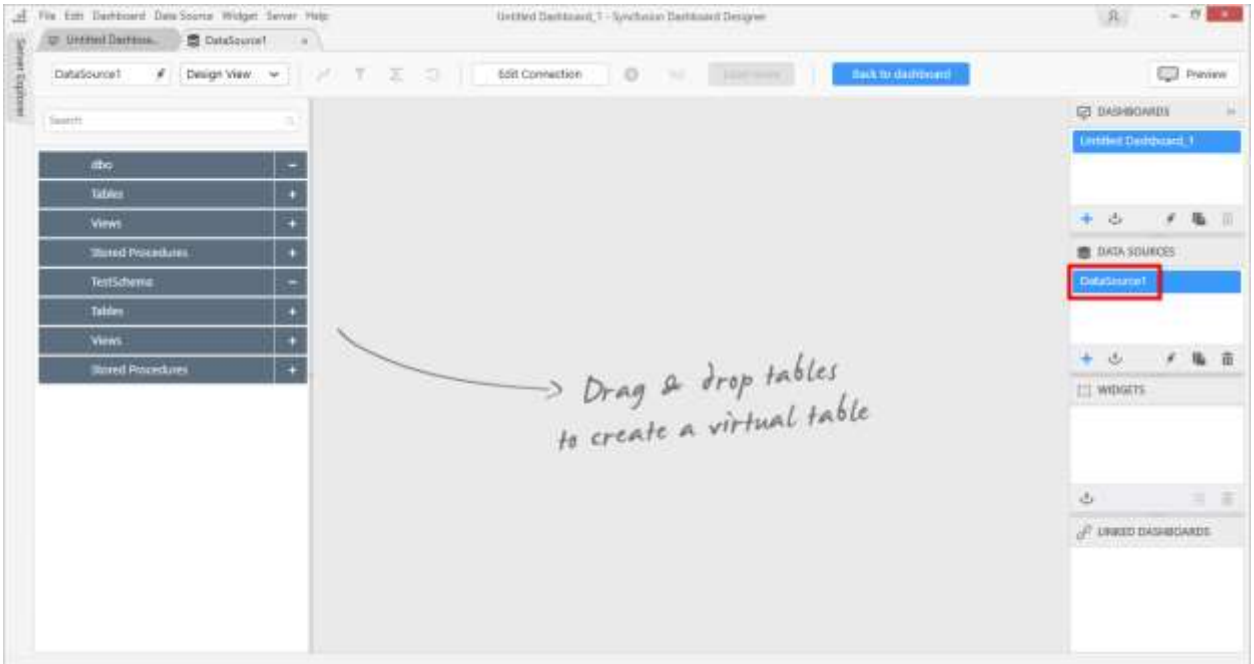
If it is the empty dashboard that was newly created, on launch, it will display a smart screen window through which too, the data source can be created as indicated in the above image.

The New Connection dialog will launch like below.

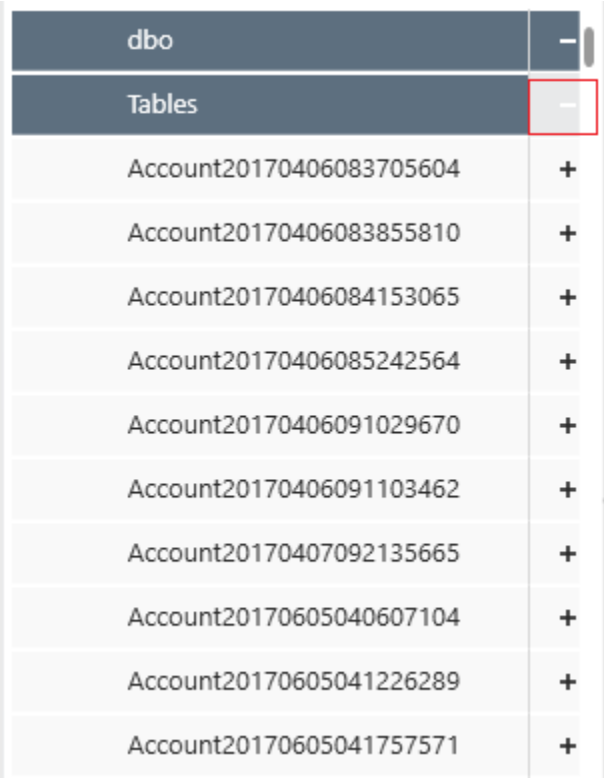


Enter a suitable name for data source at your convenience, fill the other details as requested and establish connection to the respective data source. Test Connection option will be available for server based data connection types to validate the connection with the details that you filled in.

Click Connect to navigate to the data design view. It holds the tables, views, etc. available under the connected data source. Now, a new data source was created.



You can drag and drop tables or views in data design view by expanding the tree view.



Connecting to Data

Before designing a dashboard, it is necessary to decide the data that you are going to visualize, to pick the right widgets for visualization.

Data Connection Types

Dashboard Designer supports various data connection types such that the structured data can be in any specific format to consider for processing.

- Microsoft Excel
- CSV
- JSON
- Text Document
- Microsoft SQL Server
- Microsoft SQL Server Analysis Services
- ODBC

      1. Microsoft SQL Server

      2. MySQL

      3. Oracle

      4. Microsoft Access

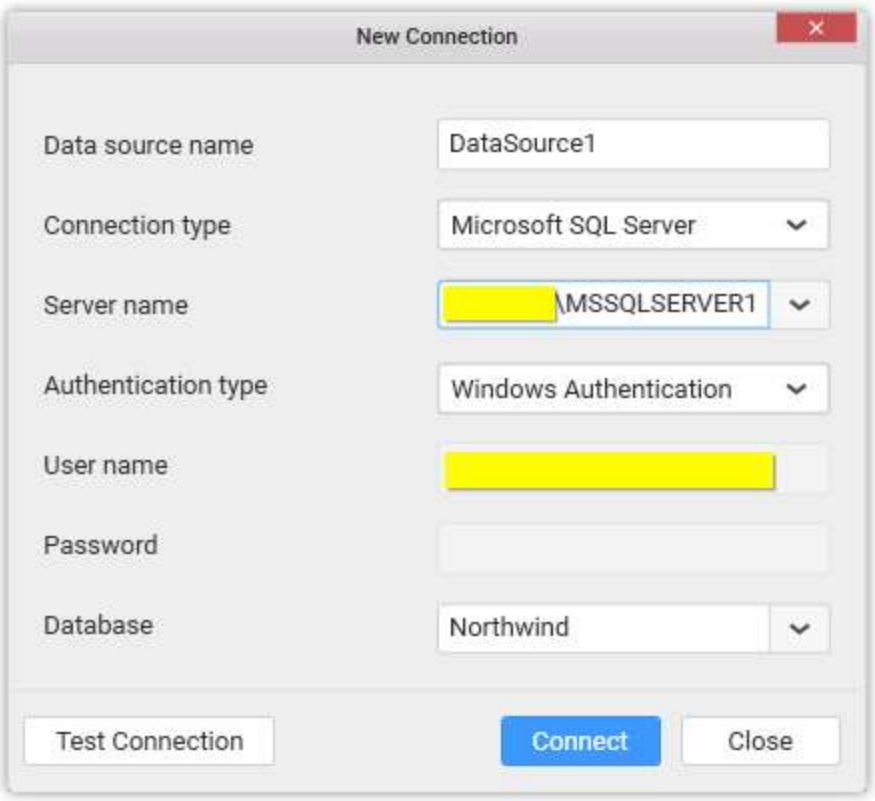
      5. Hive

      6. Others (ANSI SQL)

- PostgreSQL
- SQLite
- Microsoft Azure Table Storage
- Salesforce
- Spark SQL
- Web Data Source
- Hive

Connecting to SQL Server Database

With Microsoft SQL Server data connection type, you can connect to a database hosted in Microsoft SQL Server whose version should be 2005 and above. Also you can connect to SQL Azure hosted in Azure cloud.



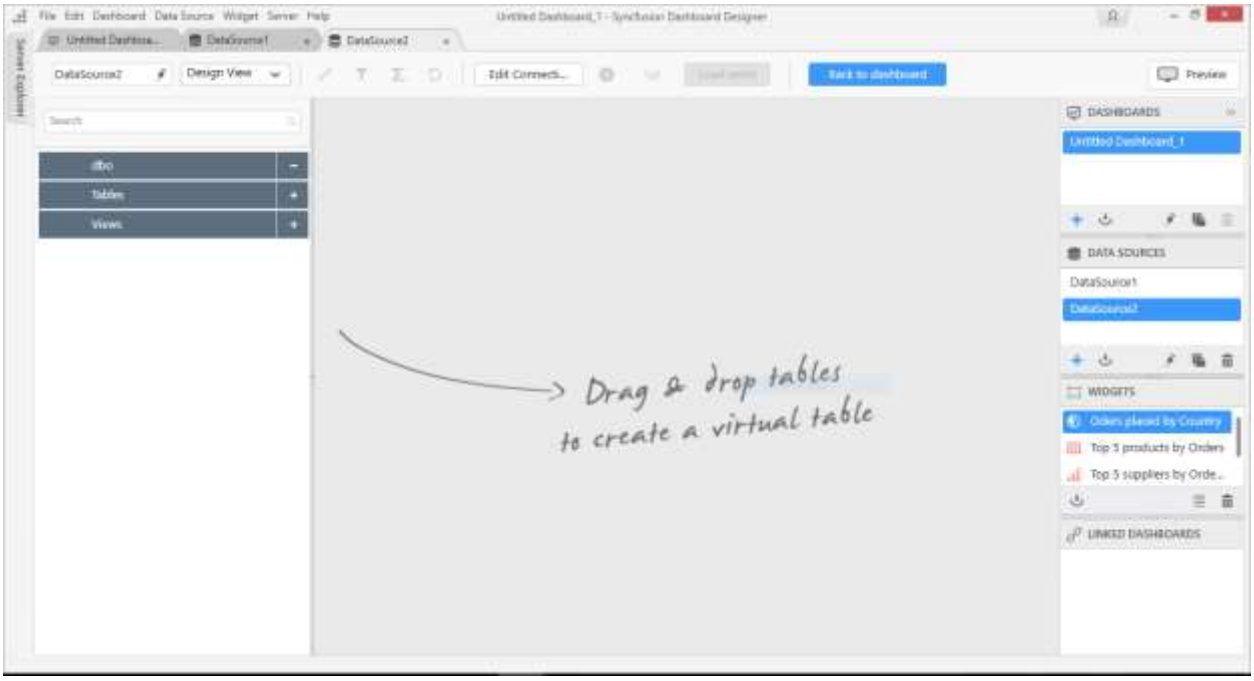
Set Connection type as Microsoft SQL Server.

Set the Server name. You may either select the server existing in the local network from the dropdown list or specifying the specific remote server name like **myserver.domain.com**, **myserver.domain.com,port**, **ipaddress** and **ipaddress,port**.

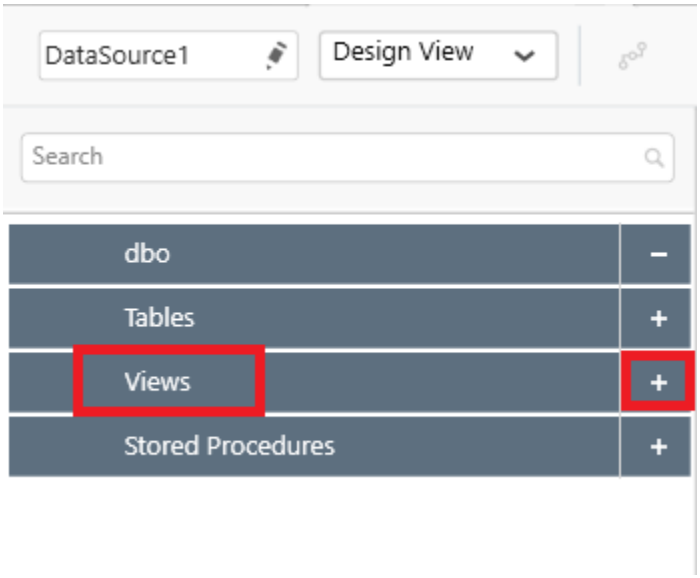
Set the Authentication type. For SQL Server Authentication, credentials are required.

Set the Database that you need to connect and test its connection. If it succeeds, proceed to connect it through clicking the Connect button.

Now you will get into the data design view window.



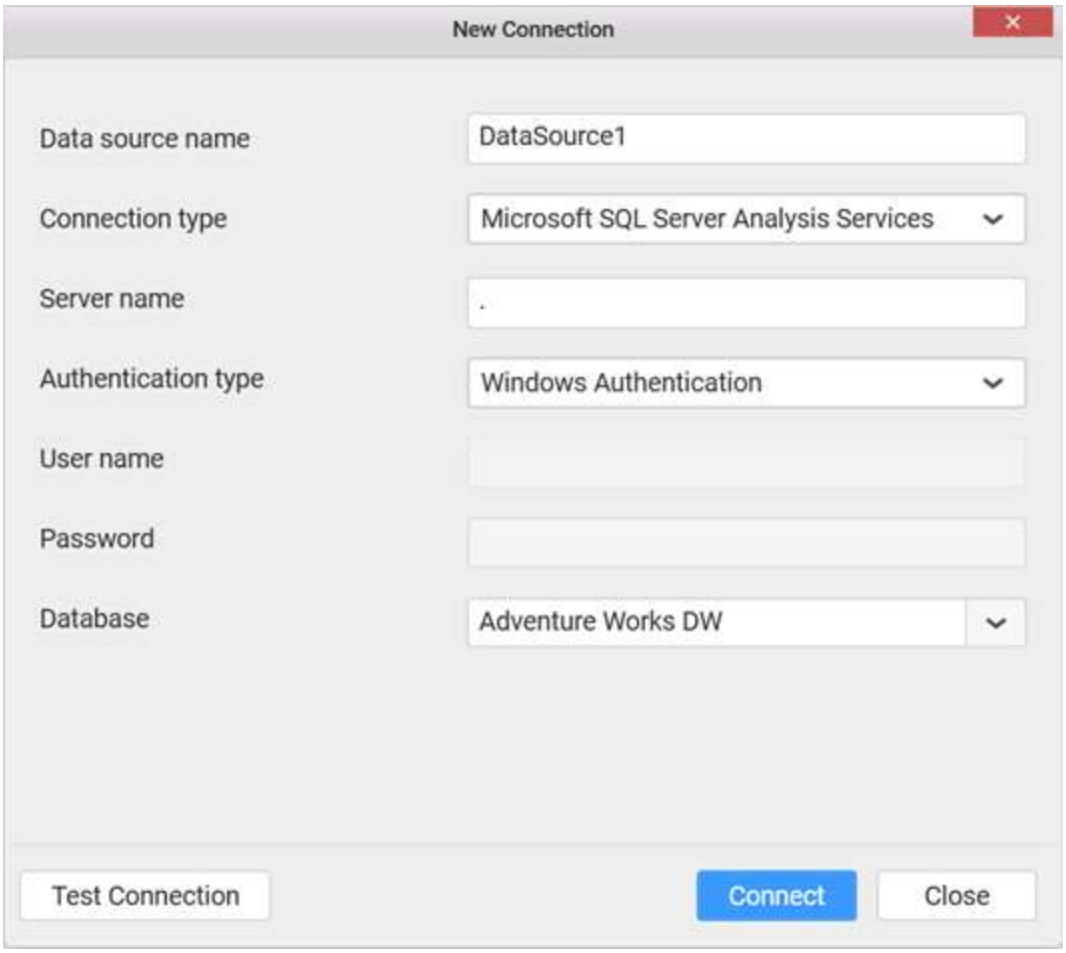
You can also connect SQL Views in design view just like tables. Simply drag and drop any SQL View under Views section in tree-view to design view.



Note: If you have only one table in the bounded database, the data design view will have that one table added by default.

Connecting to SQL Server Analysis Services Cube

With Microsoft SQL Server Analysis Services data connection type, you can connect to a cube hosted in Microsoft SQL Server Analysis Services whose version should be 2012 and above. Also you can connect to a cube hosted in Azure Analysis Services using Server Authentication type alone.



In the New Connection dialog, enter a data source name and select connection type as Microsoft SQL Server Analysis Services.

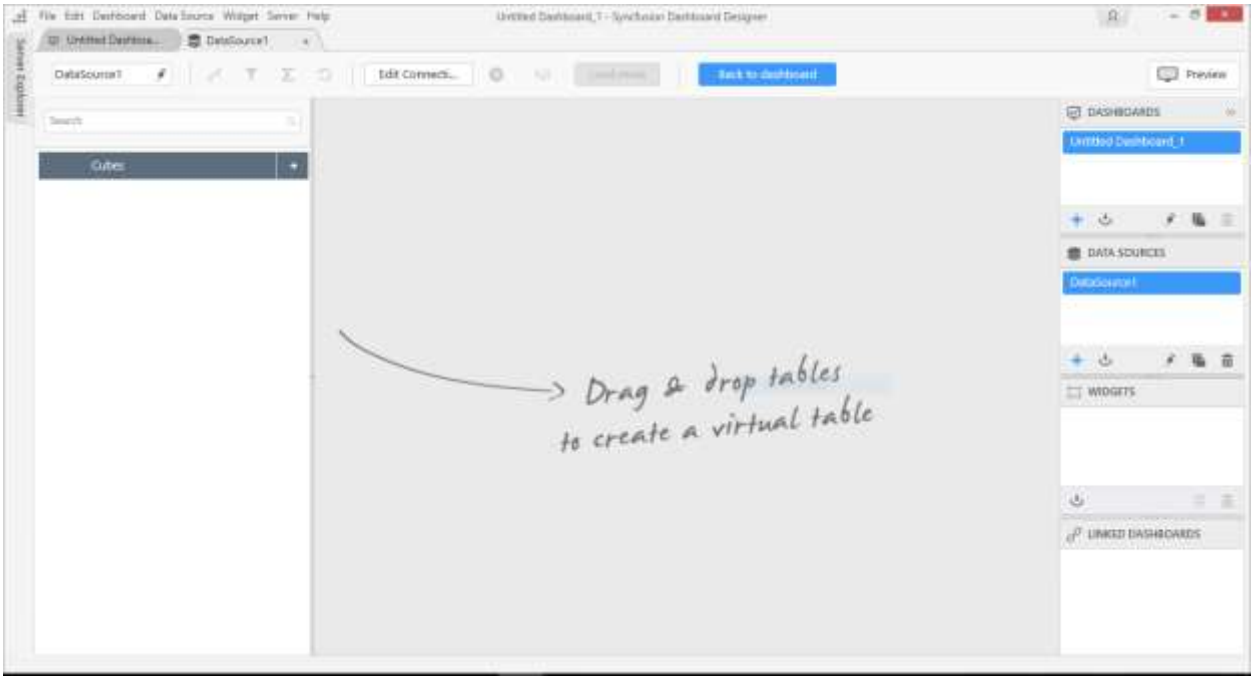
Set the name of the server. You may either select the server existing in the local network from the dropdown list or specify the remote server name like **myserver.domain.com**, **myserver.domain.com:port**, **ipaddress** and **ipaddress:port**.

Note: For more SQL Server name formats, please refer here.

Set the authentication type. For Server Authentication, user credentials are required.

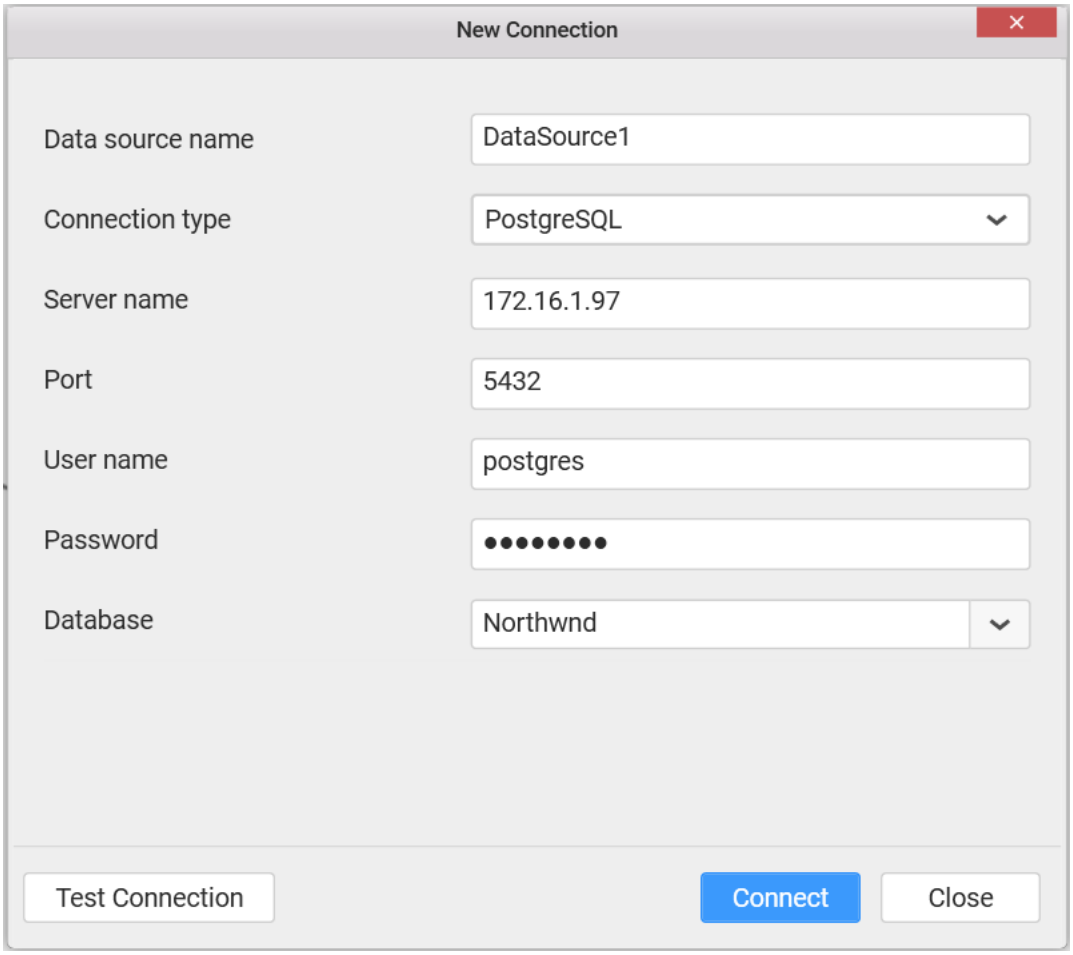
Select the Database that you need to connect and test its connection. If it succeeds, proceed to connect it through clicking the Connect button.

Now you will get into the data design view window



Connecting to PostgreSQL Database

With PostgreSQL connection type, you can connect to a PostgreSQL database hosted in local or remote machine whose version should be 9.x and above.



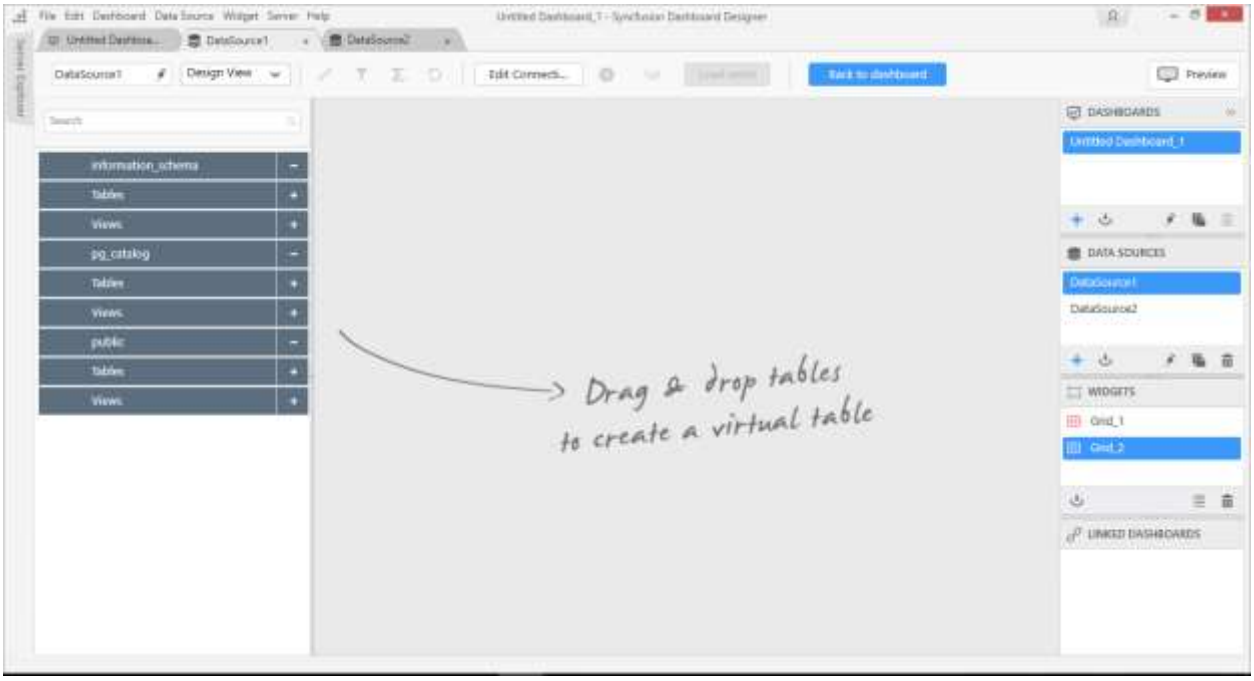
In the New Connection dialog, enter a data source name and select connection type as PostgreSQL from the drop down list.

Set the server name and port number where the PostgreSQL service is running.

Enter the user name and password to connect to the PostgreSQL server.

Select the database that you need to connect and test its connection. If it succeeds, proceed to connect it through clicking the Connect button.

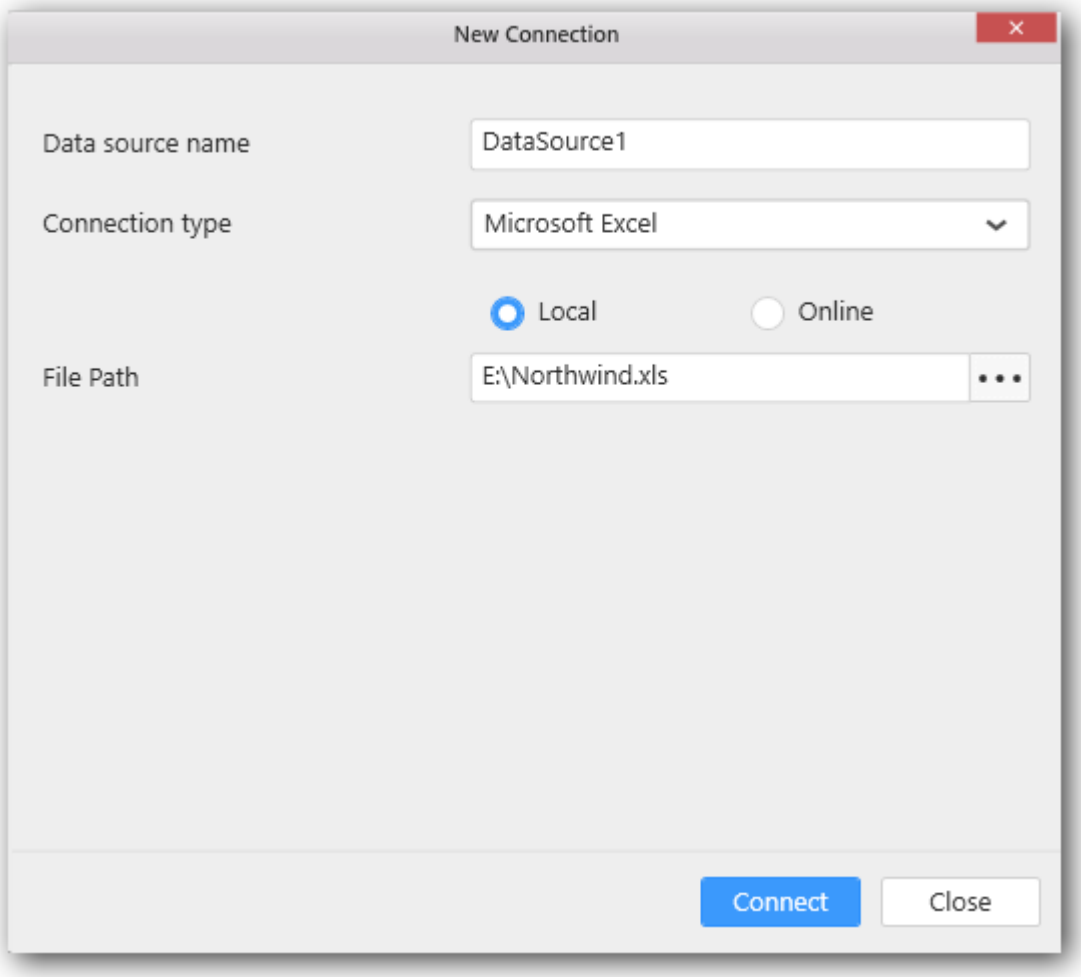
Now you will get into the data design view window.



Note: If you have only one table in the bounded database, the data design view will have that one table added by default.

Connecting to an Excel workbook

With Microsoft Excel data connection type, you can connect to an Excel workbook either from Local Machine or Cloud File Storage, whose format can be XLS or XLSX.



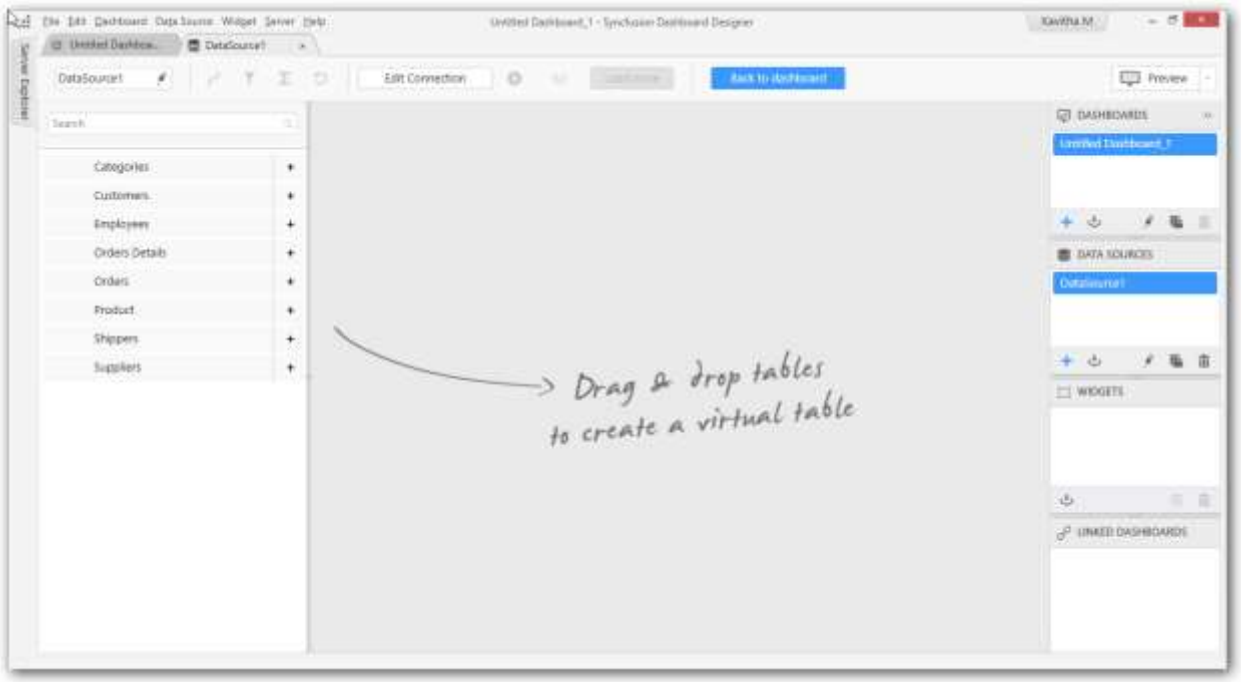
Set Connection type as Microsoft Excel.

Local

Through Local storage you can connect the Excel workbook which is existing in your local machine.

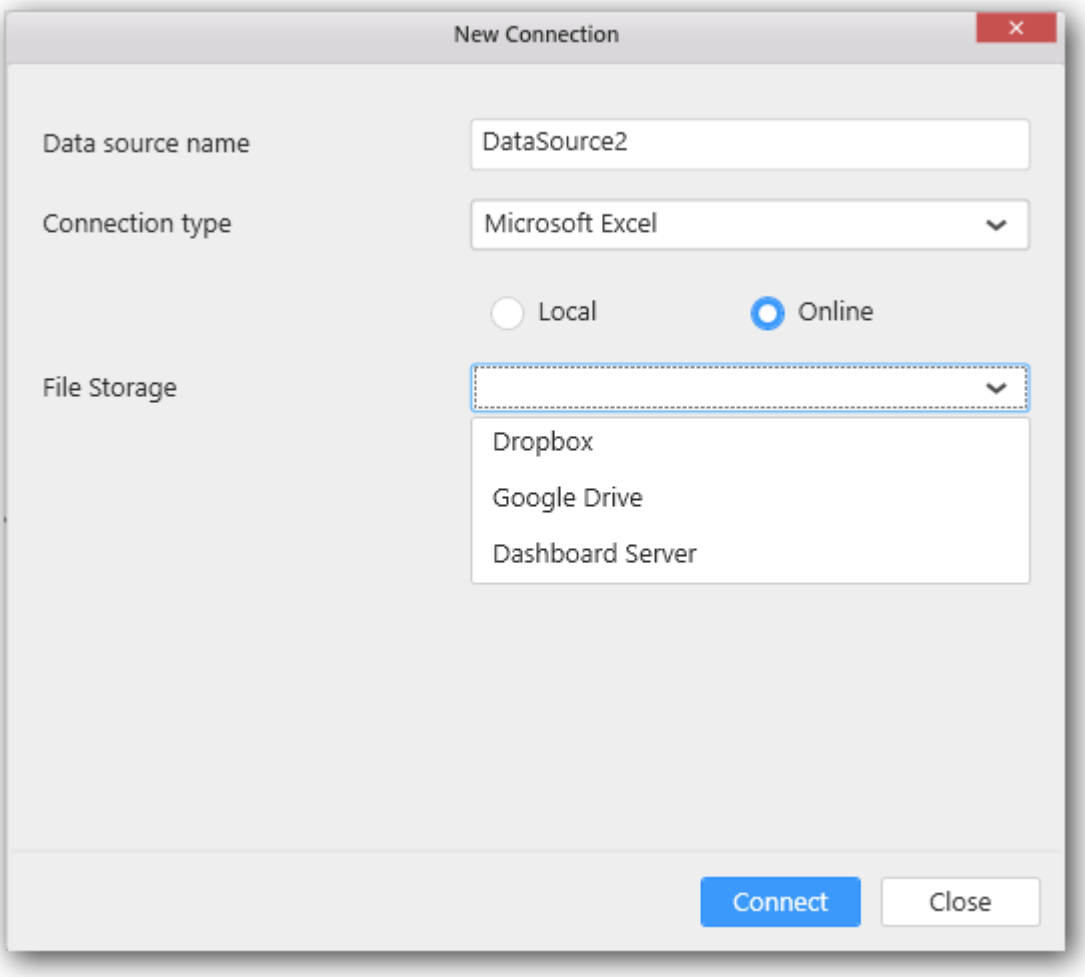
Set the File Path through locating the Excel workbook accessible location and click *Connect*.

Now you will get into data design view window.



File Storage

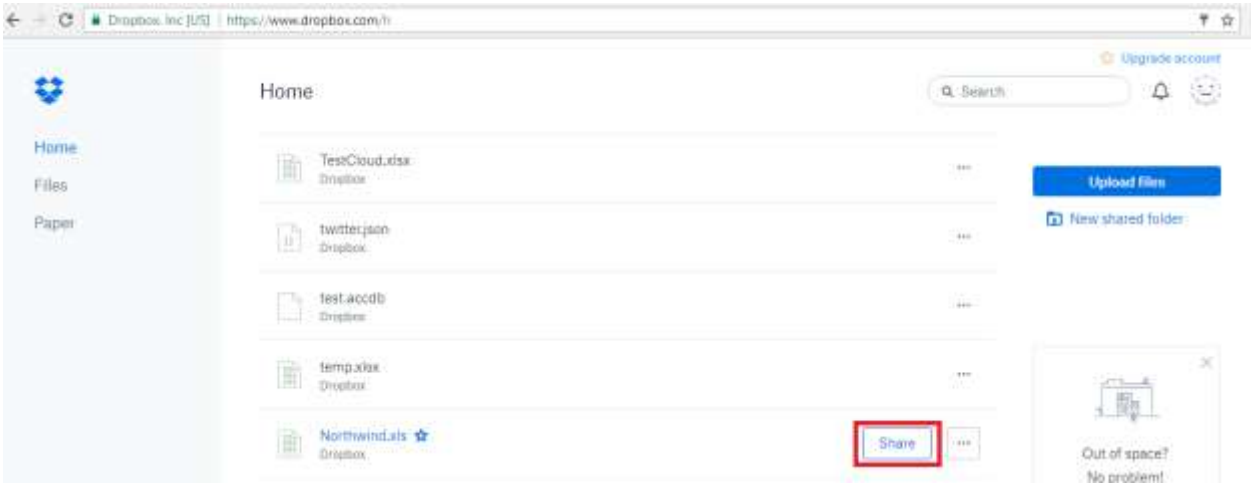
Through File Storage connection you can connect Excel workbook from cloud service using either Dropbox or Google Drive account.



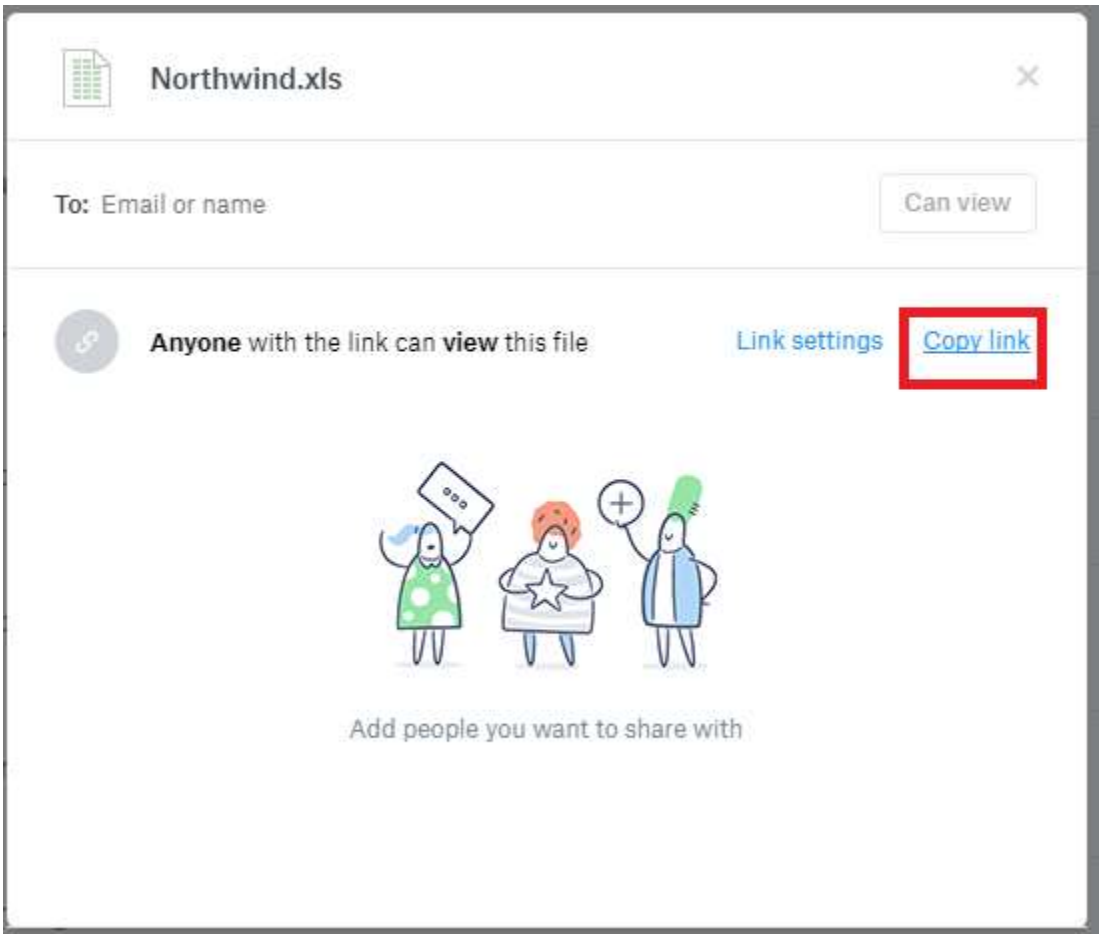
Procedure to get file URL from cloud service:

Dropbox

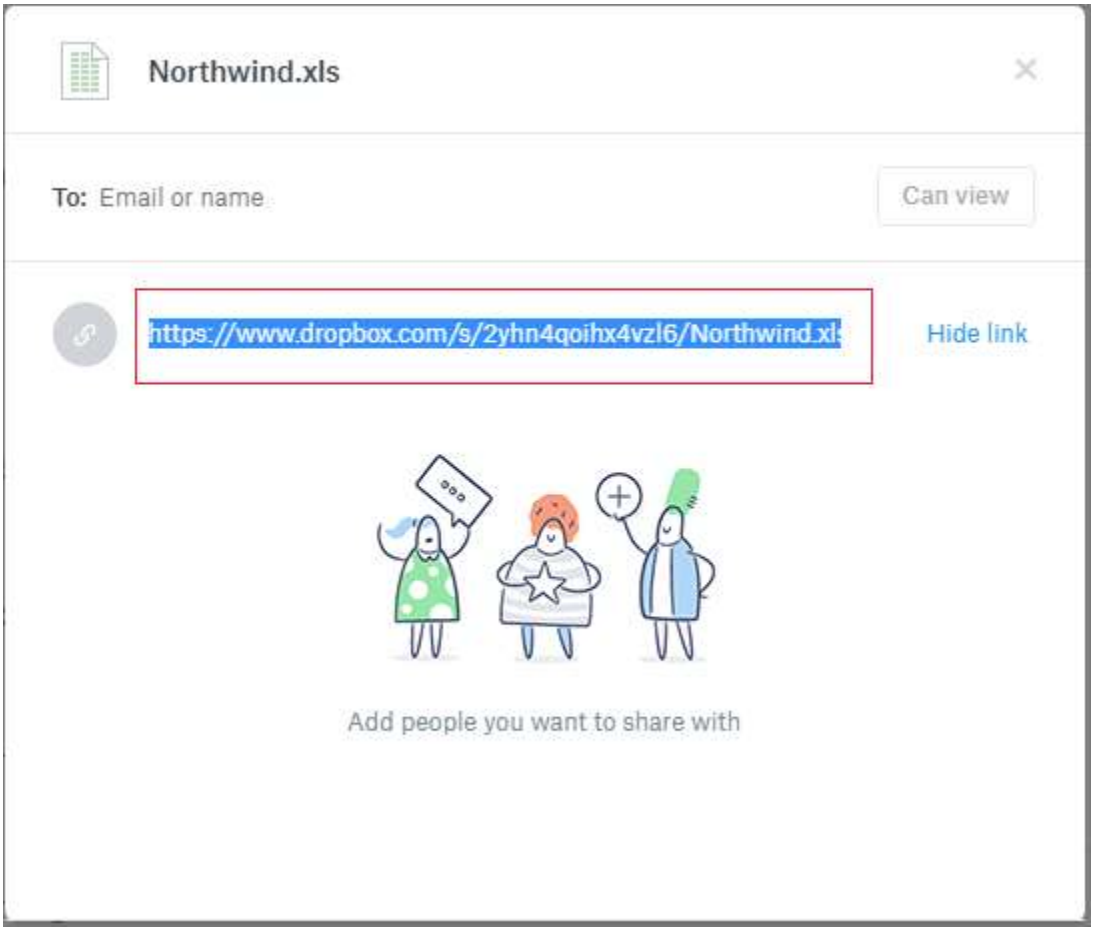
Sign in to your Dropbox account and choose the file to upload. Once the file is uploaded, click the Share option available next to it.



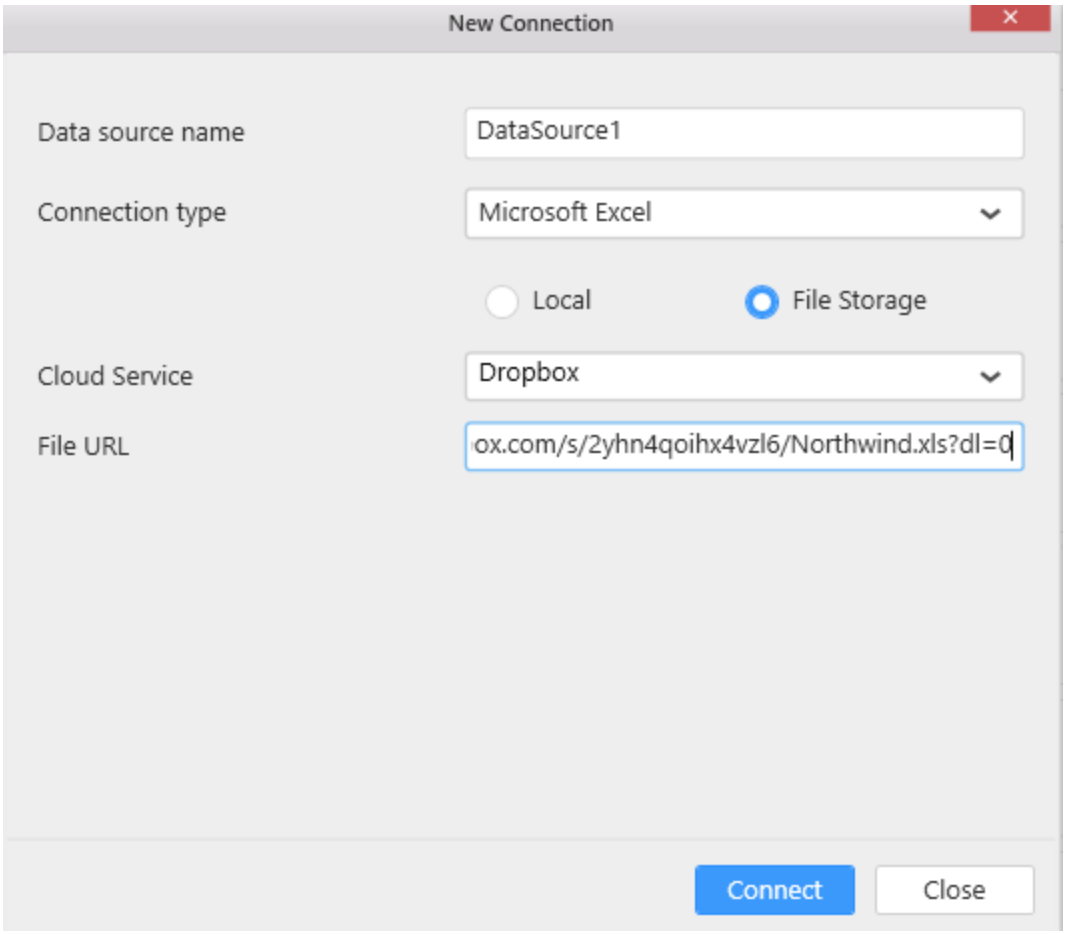
Then choose Copy Link option to copy the URL link.



Now you will get the File URL to connect the selected file in dashboard designer.

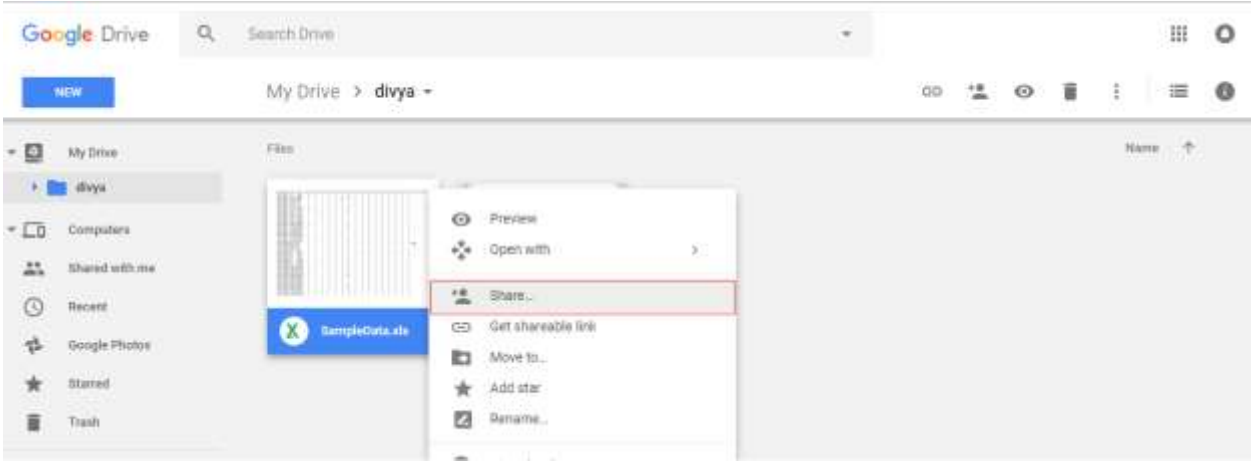


Paste the link in File URL box and click Connect button. Now you will get data design view window.

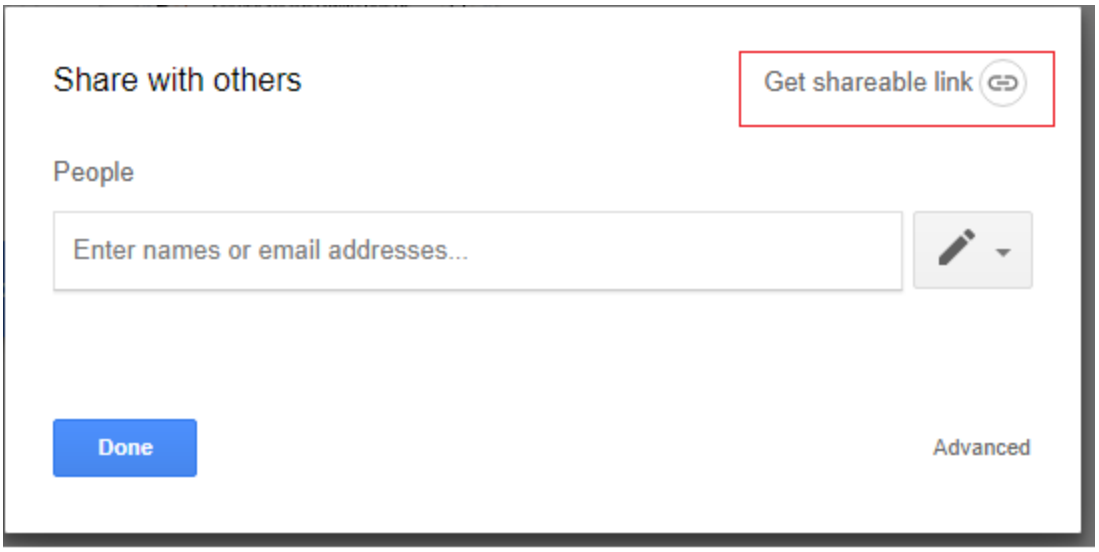


Google Drive

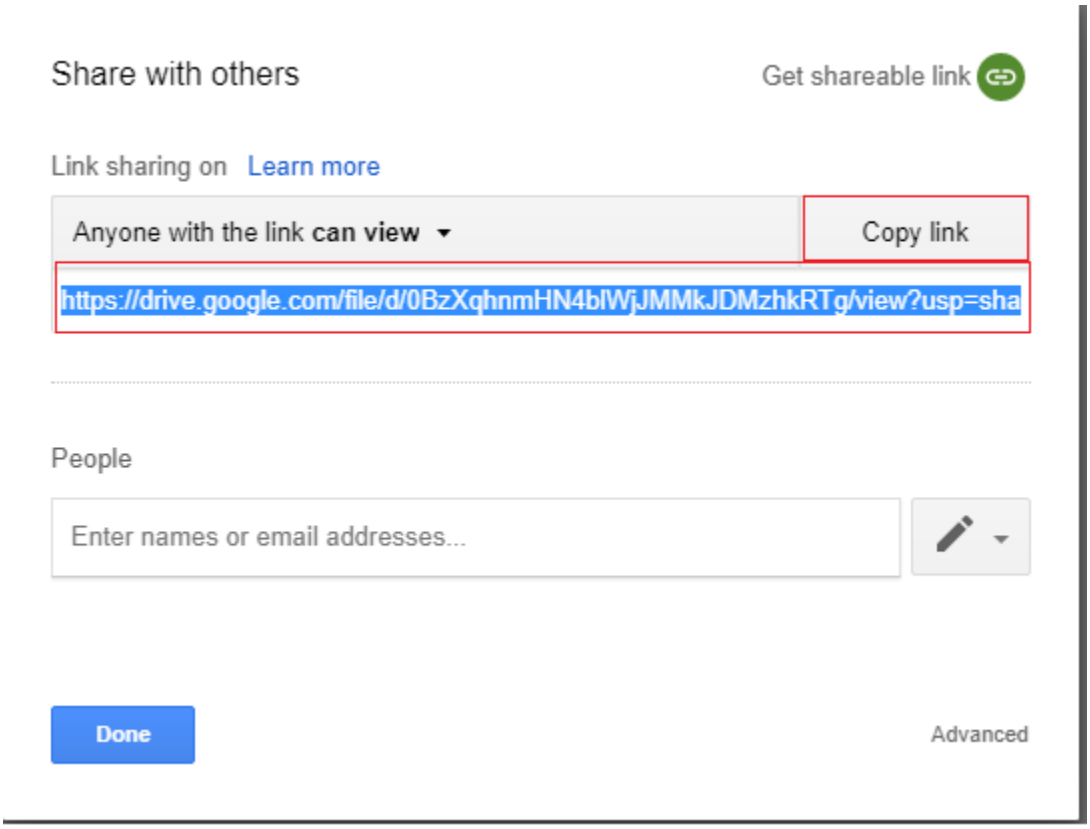
Sign in to your Google Drive account and choose the file to upload. Once the file is uploaded, right-click on it.



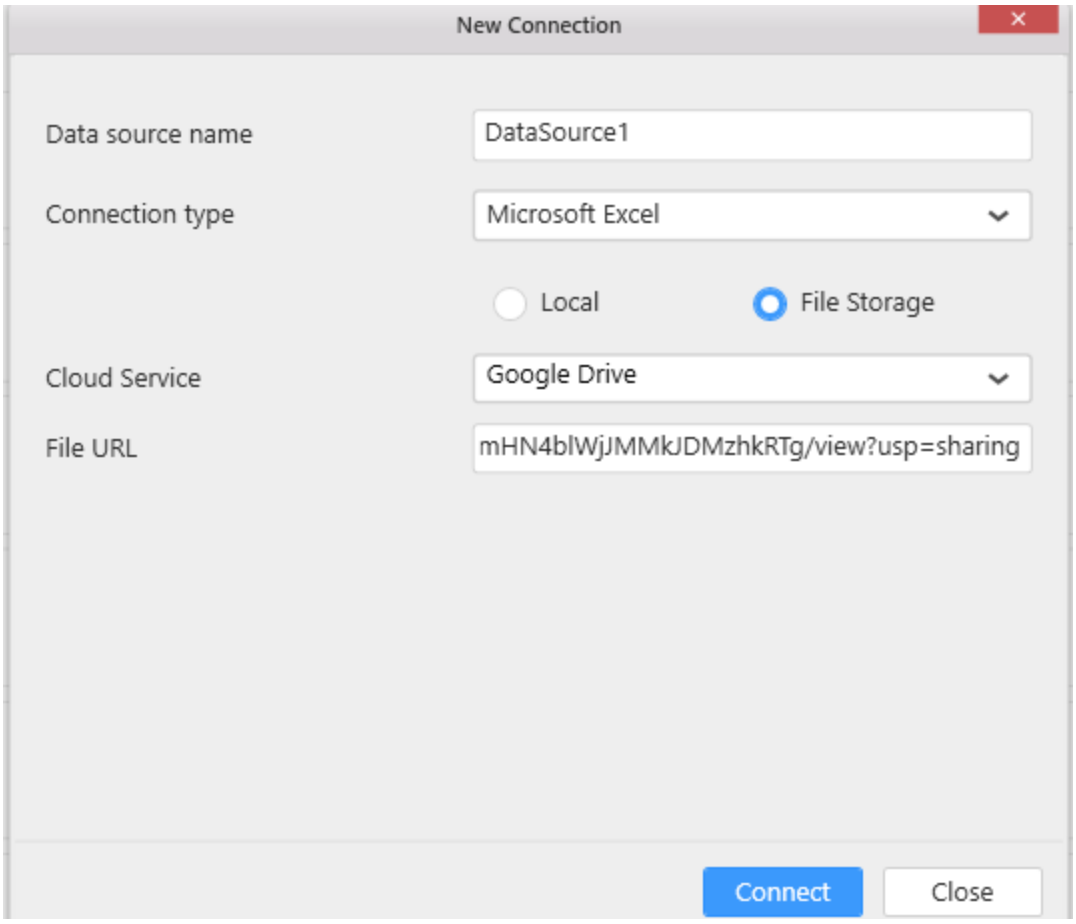
Now choose Get Shareable Link option to get the file URL.



Now you will get the link and use Copy Link option to copy the URL link.



Paste the respective link in File URL box and click Connect button. Now you will get data design view window.



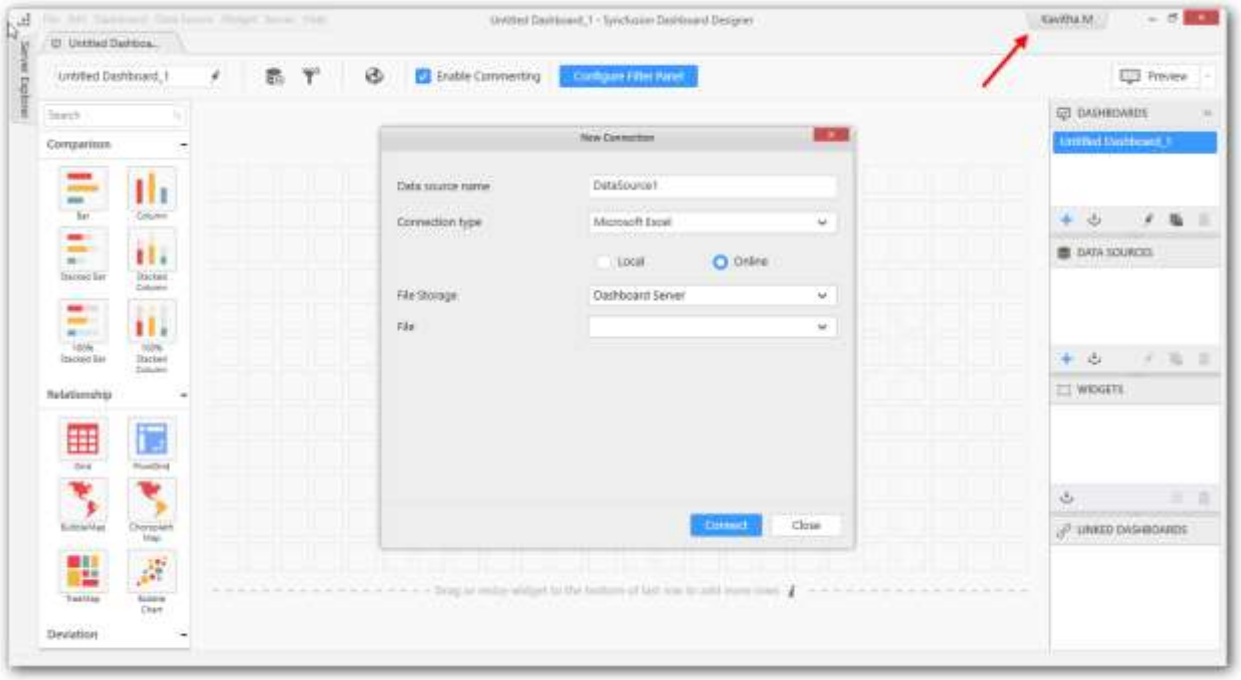
Dashboard Server

Before connecting a file from the Dashboard Server, you must upload a file in the Dashboard Server.

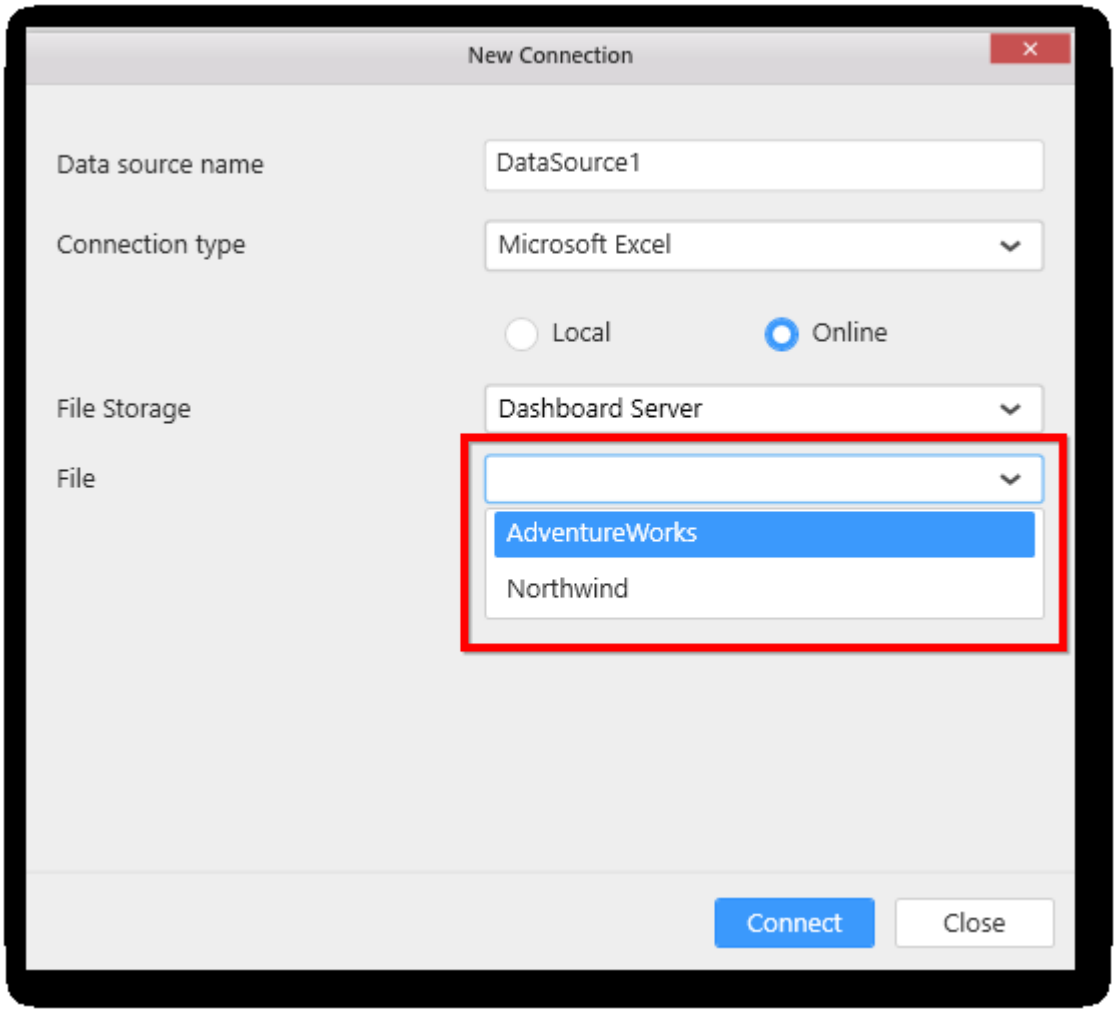
Refer here for file uploading in the Dashboard Server.

When the file uploading is completed, you can connect those files from the Dashboard Designer.

Log on your Dashboard Server account from the Dashboard Designer.



Select the respective file from the drop-down list and click the Connect button. Now, you will get the data design view window.



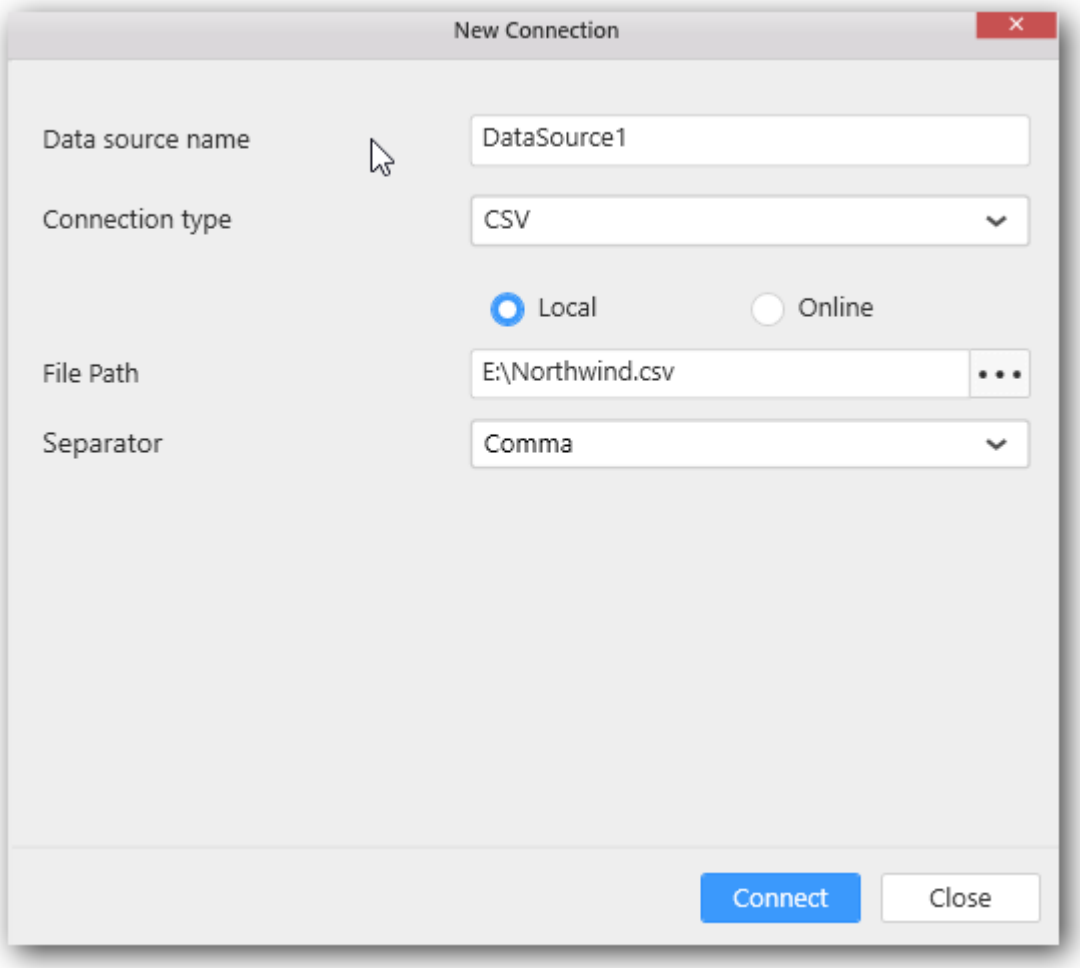
Note: You can also connect CSV, JSON, Text document, SQLite and Microsoft Access connection via cloud file storage using the above mentioned procedures.

Note: If you have only one worksheet in the excel workbook, the data design view will have that one table added by default.

Information: The Excel workbook that need to be connected, should have column names at first row of sheet followed by data rows.

Connecting to a CSV file

With CSV data connection type, you can connect to any comma separated value formatted file (CSV). Apart from Comma Separator, Semicolon, Space and Tab (*.tsv files) Separated files are also supported.

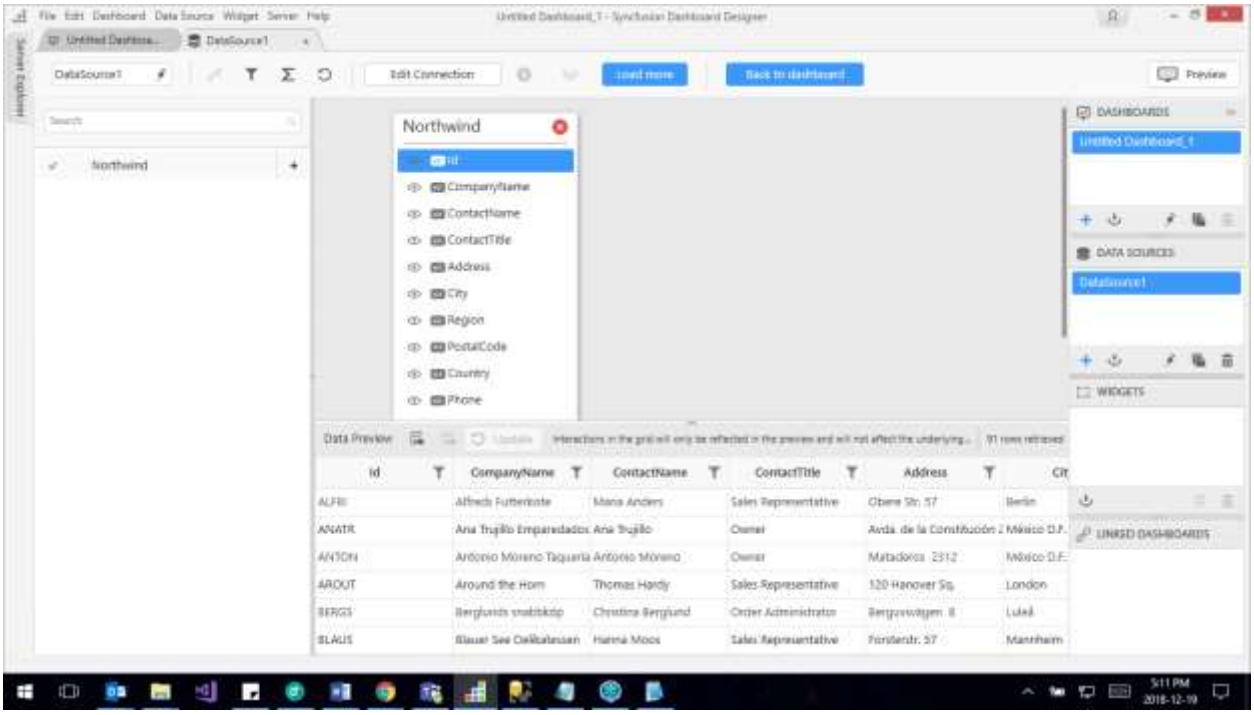


Set Connection type as CSV.

Set the File Path through locating the CSV file existing in your machine accessible location and click Connect.

Select a Separator value by which the respective file values are separated.

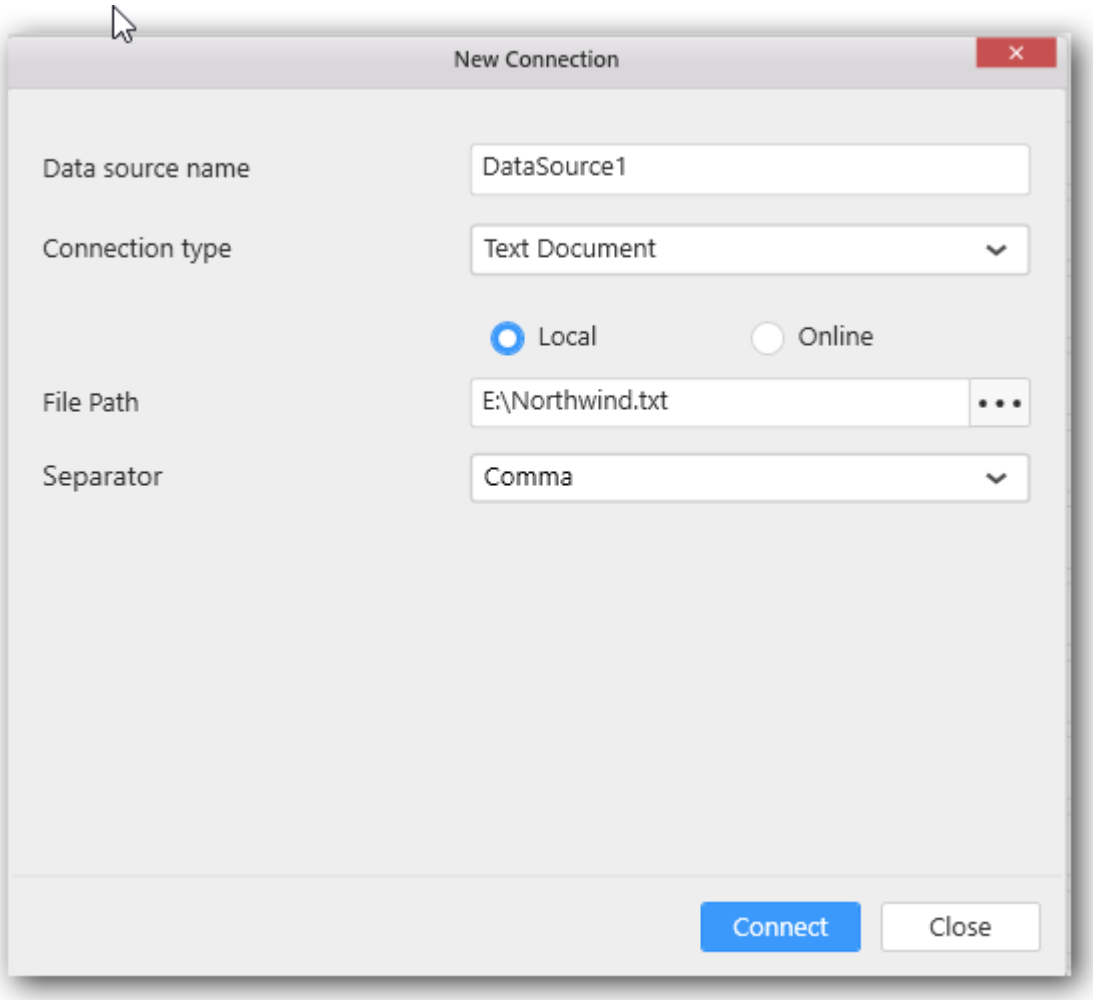
Now you will get into data design view window.



Note: The data design view will have that only table added by default.

Connecting to a Text Document

With Text Document data connection type, you can connect to any text document (*.txt) whose values are separated by any one of the separators like Comma, Semicolon, Space and Tab.

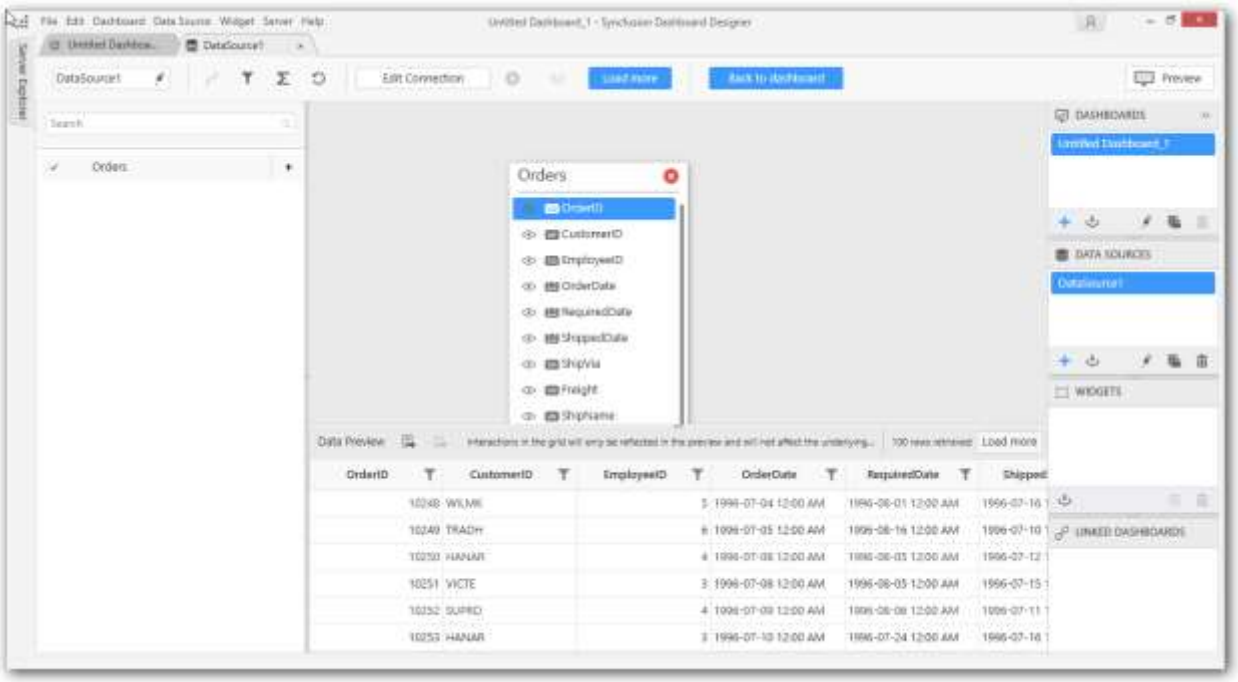


Set Connection type as Text Document.

Set the File Path through locating the Text Document file existing in your machine accessible location and click Connect.

Select a Separator value by which the respective file values are separated.

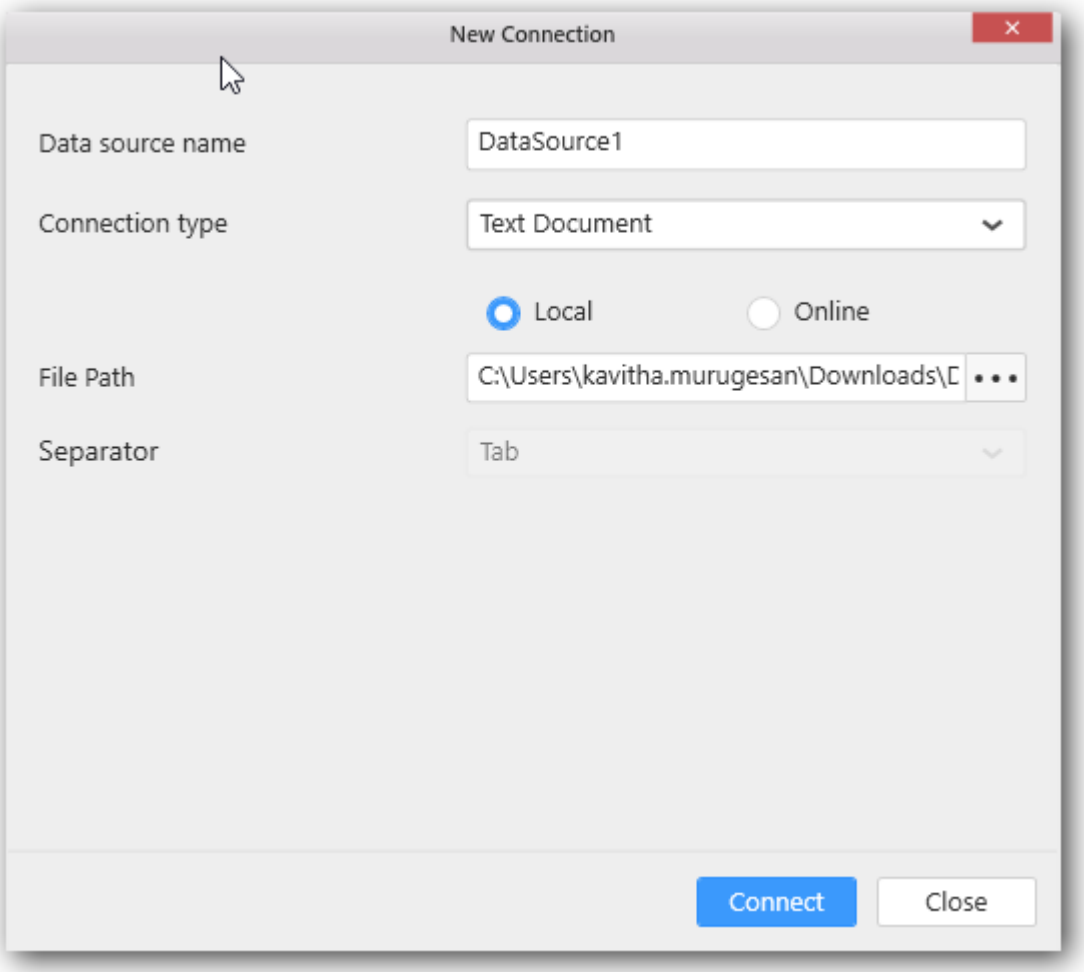
Now you will get into data design view window.



Note: The data design view will have that only table added by default.

Connecting to a XER File

Text Document option now establishes connection with *.xer extension file types which comes with multiple tables along with tab-separated values.

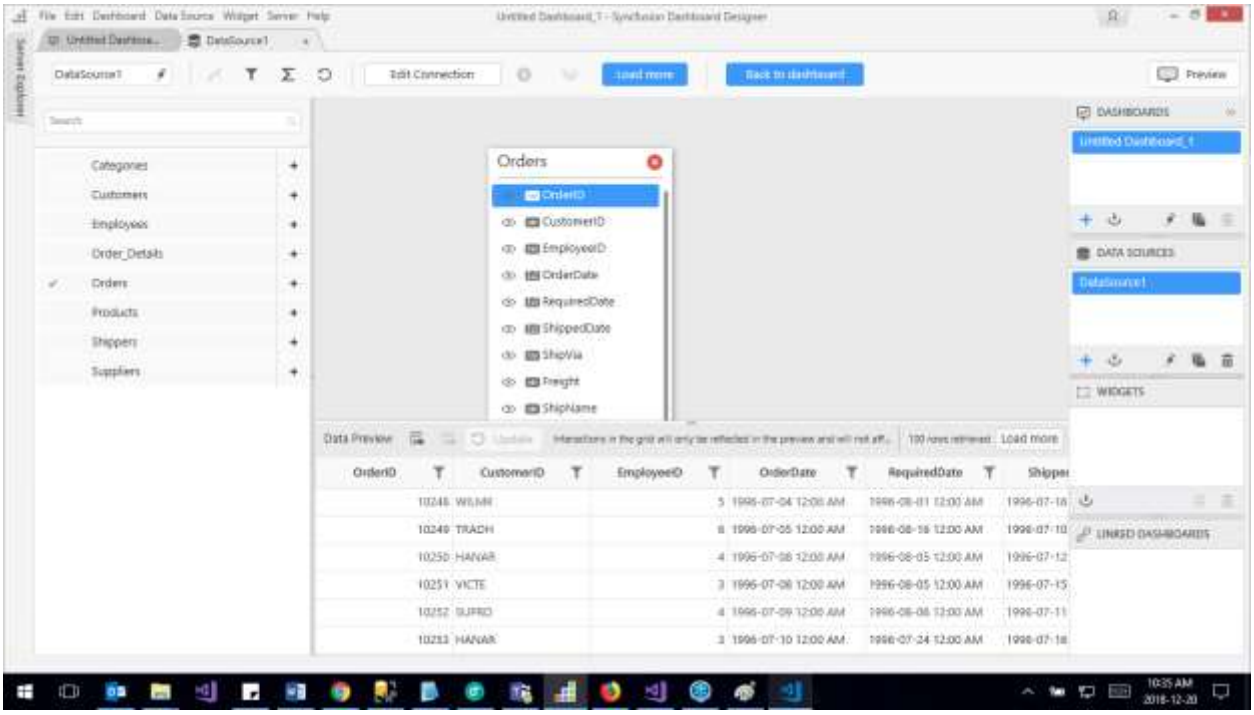


Set Connection type as Text Document.

Set the File Path through locating the XER file existing in your machine accessible location.

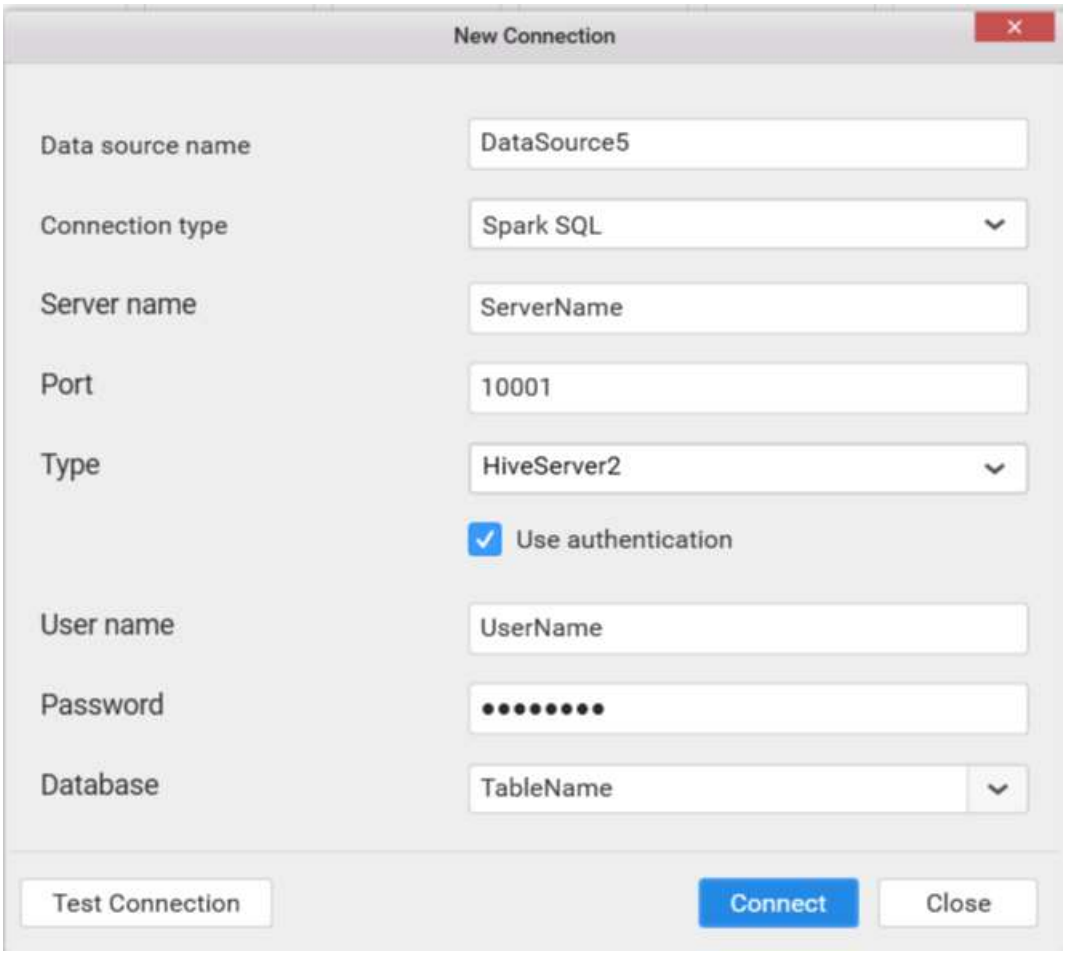
By default Tab Separator will set automatically and finally click Connect.

Now you will get into data design view window.



Connecting to Spark SQL Data

With Spark SQL data connection type, you can connect to data placed in HDFS processed by Spark on Hive Server.

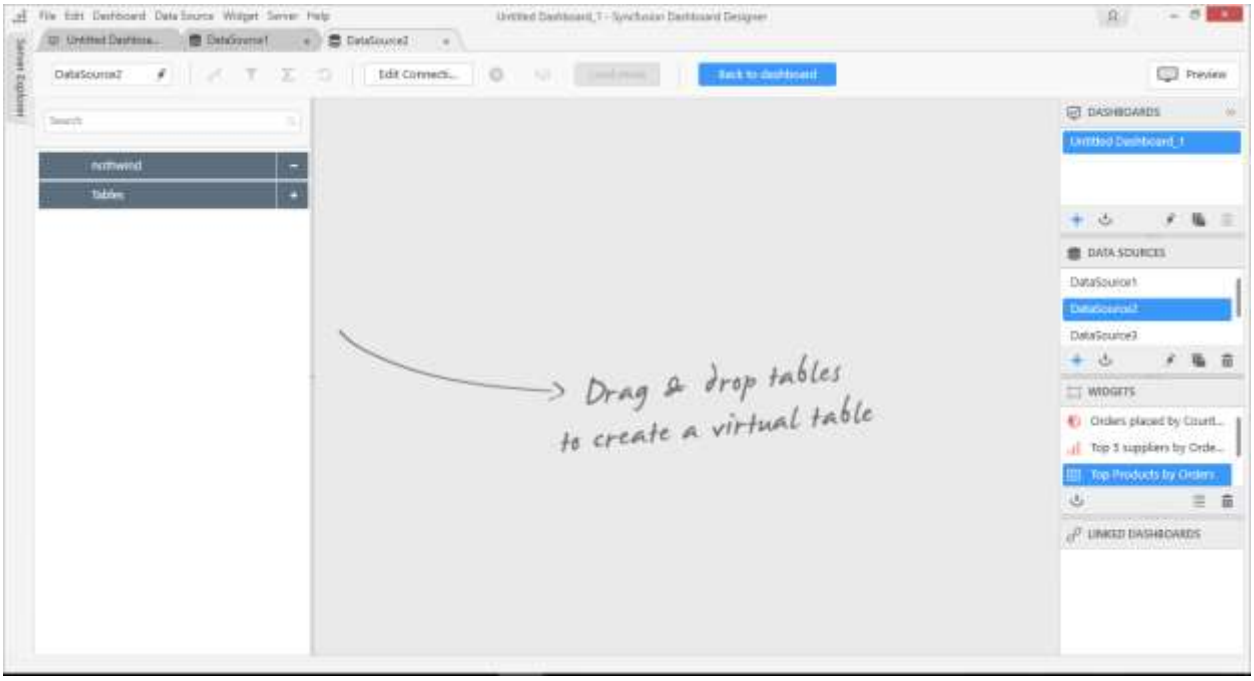


Set Connection type as Spark SQL.

Set the Server name which can be either IP address or the host name of the server where the data resides in HDFS and Spark is running.

The Port number and the Type will be filled by default as 10001 and HiveServer2 respectively.

Select the Database to connect to and test the connection. If it succeeds, you will get into the data design view like below.

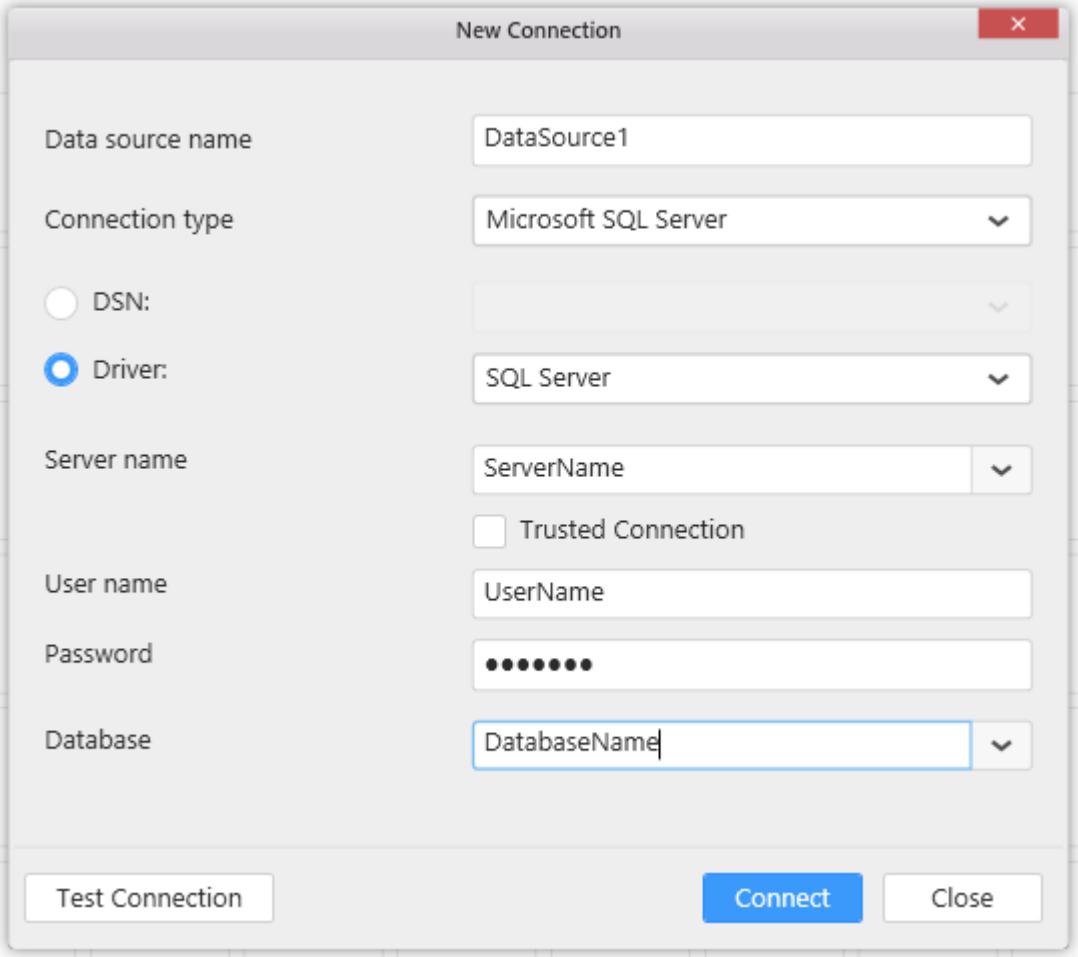


Note: If you have only one table in the bounded database, the data design view will have that one table added by default.

Connecting to Microsoft SQL Server Database through ODBC

Through Microsoft SQL Server (in ODBC category) data connection type, you can connect to ODBC enabled SQL database hosted in Microsoft SQL Server 2005 & above through the following ODBC driver.

- SQL Server



Setting up an ODBC-enabled SQL Server database

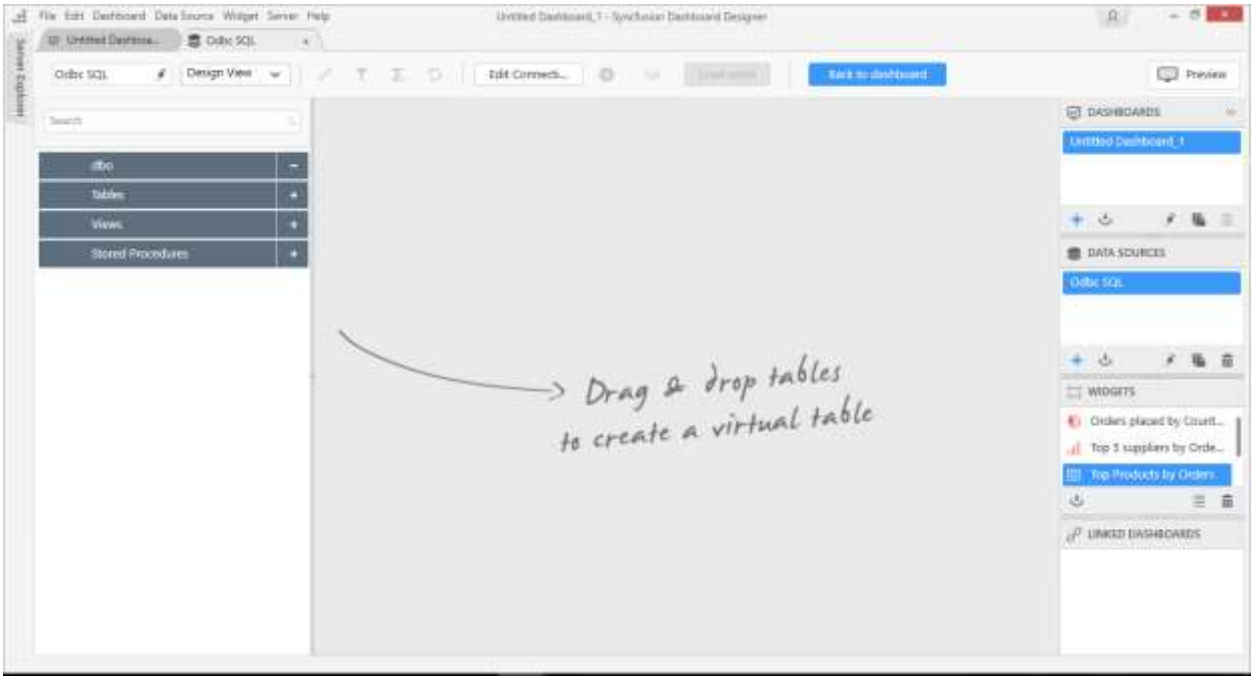
To setup a new data source with SQL database, add the installed driver through the following procedure.

1. Navigate to the Control Panel -> System and Security -> Administrative Tools folder location.
2. Double click the ODBC Data Sources (32-bit) tool to open the ODBC Data Source Administrator (32-bit) dialog.
3. Select the System DSN or User DSN tab where the data sources already available get listed in the list. Click Add to select the specific driver and bind the preferred database to create a new ODBC data source.

Connecting to ODBC SQL Server data source

1. Select Microsoft SQL Server in ODBC category as Connection type in the New Connection wizard.
2. Select Data Source Name (DSN) or the SQL ODBC driver that you added, and bind the SQL Server database path and connect, to create a new data source for dashboard.

Now you will get into the data design view window.

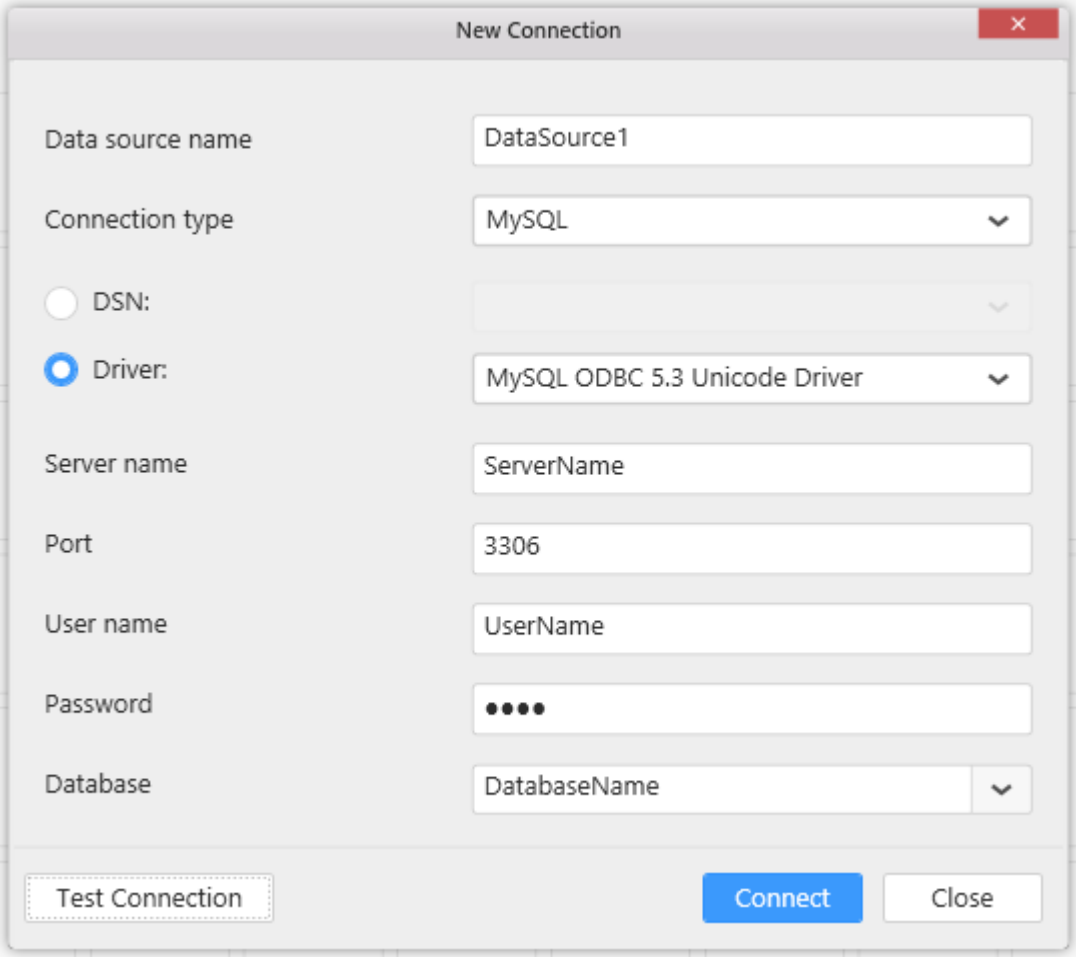


Note: For connecting SQL Server Database through ODBC, ensure both SQL driver and Syncfusion Dashboard Designer are installed on the same machine.

Information: We **recommend** to use **direct connection** of Microsoft SQL Server in database category instead of going to connect a Microsoft SQL Database through ODBC.

Connecting to MySQL Database through ODBC

Through MySQL data connection type, you can connect to ODBC enabled MySQL database.



Setting up an ODBC-enabled MySQL database

1. Download & Install the MySQL ODBC 5.3 Unicode driver from here 2. Add the installed driver through the following procedure to setup a new data source with MySQL database:

a. Navigate to the Control Panel -> System and Security -> Administrative Tools folder location.

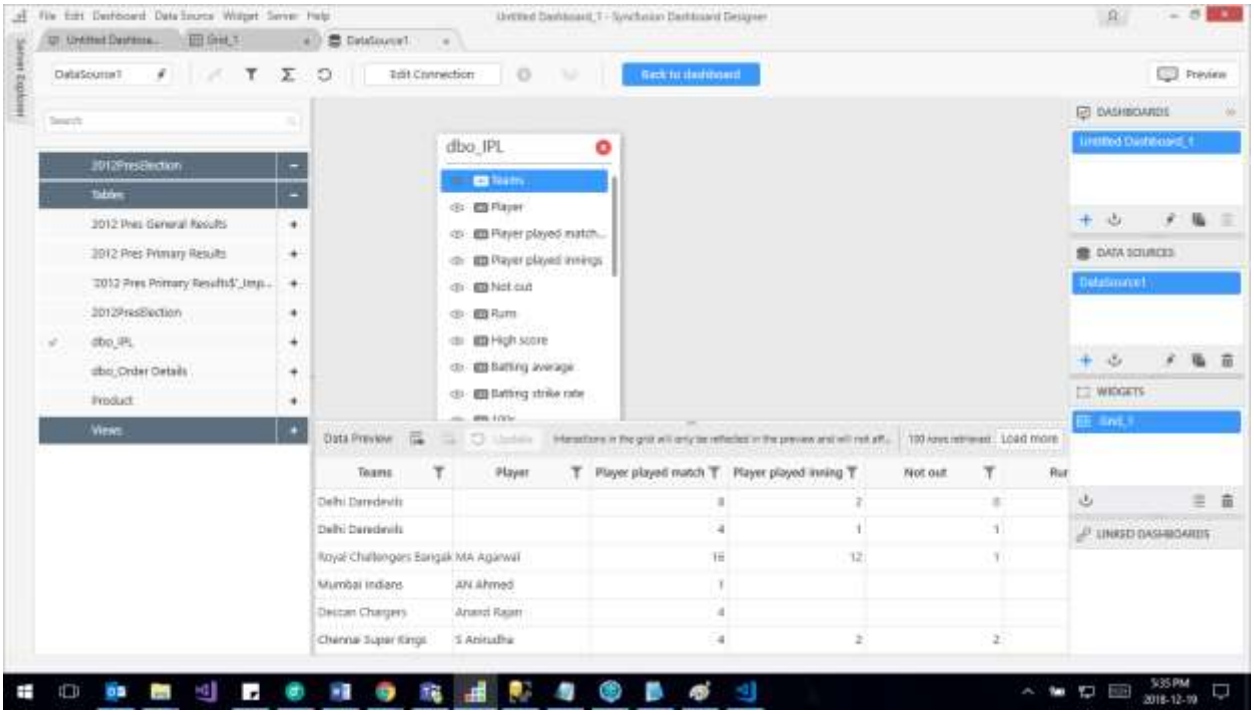
b. Double click the ODBC Data Sources (32-bit) tool to open the ODBC Data Source Administrator (32-bit) dialog.

c. Select the User DSN tab. In this tab, the user data sources already available get listed in the list. Click Add to select the specific driver and bind the preferred database to create a new ODBC data source.

Connecting to ODBC MySQL data source

1. Select MySQL as Connection type in the New Connection wizard. 2. Select Data Source Name (DSN) or the MySQL driver that you added, and select the MySQL database by providing the Server Name,Port,User Name,Password and connect, to create a new data source for dashboard.

Now you will get into data design view window.



Note: For connecting MySQL Database through ODBC, ensure both MySQL driver and Syncfusion Dashboard Designer are installed on the same machine.

Information: We can follow the same connecting procedure to connect **MariaDB** database as well.

Connecting to Oracle Database through ODBC

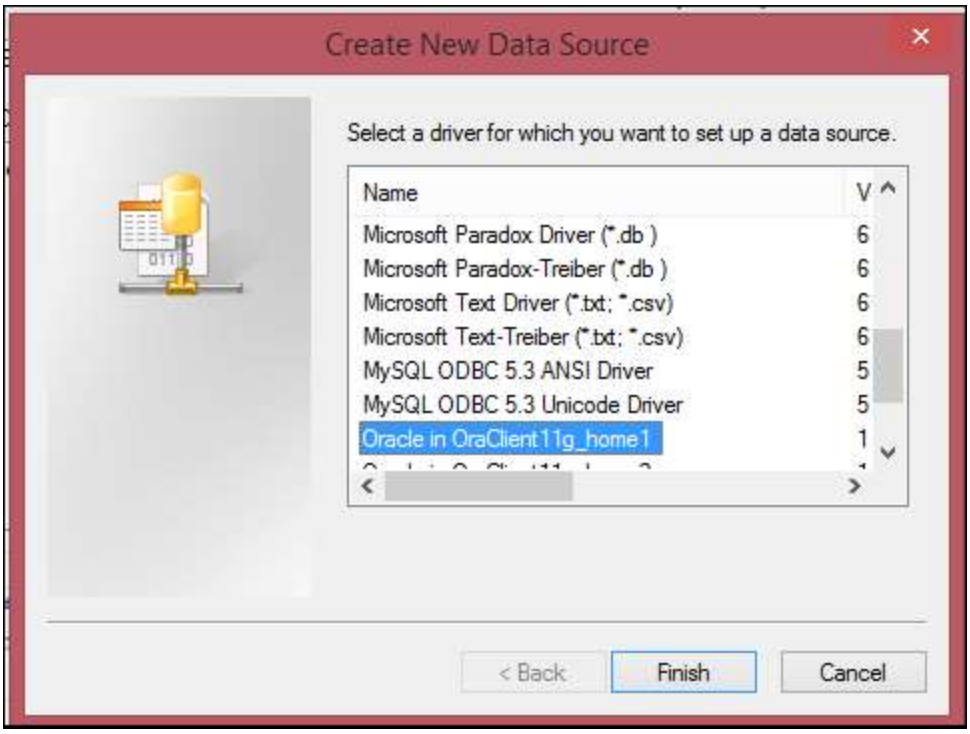
Through Oracle data connection type, you can connect to ODBC enabled Oracle database through any of the following ODBC drivers.

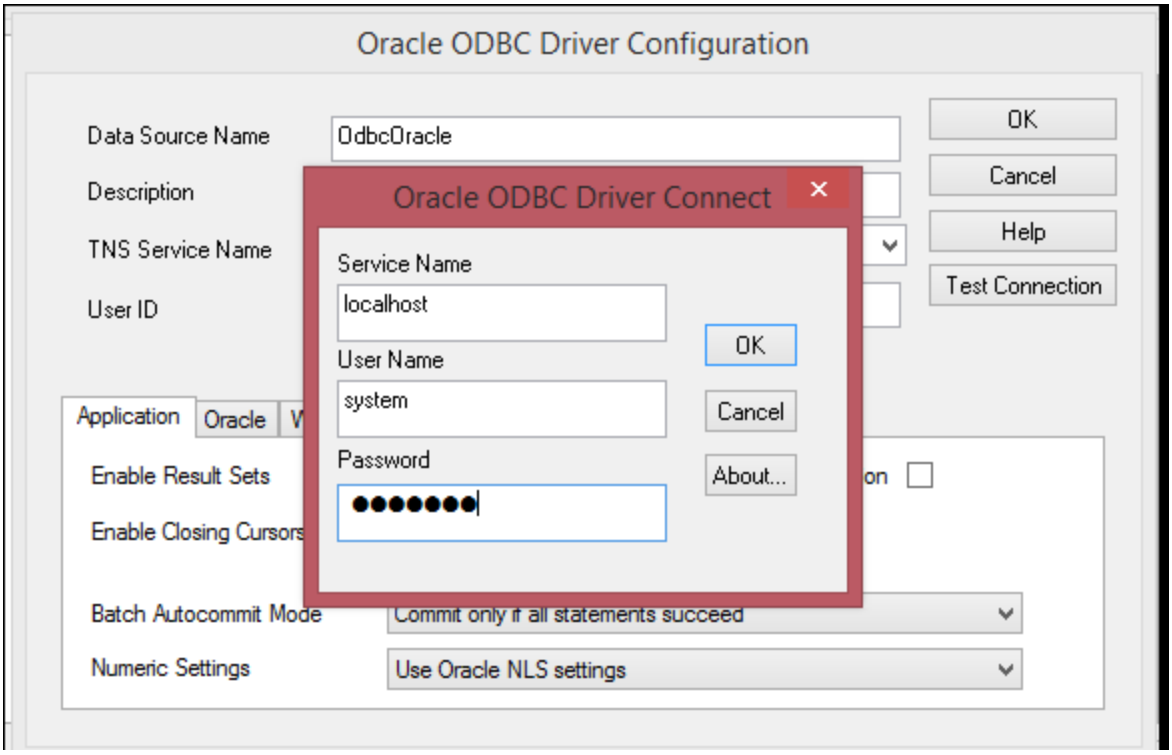
- Microsoft ODBC for Oracle - This comes with Office installation. Need to install Oracle Client software separately to use this driver.
- Oracle in OraClient 11g_home1 – This can be downloaded from here – v11.2.0.2.1

Setting up an ODBC-enabled Oracle database

To setup a new data source with Oracle database, add the installed driver through the following procedure.

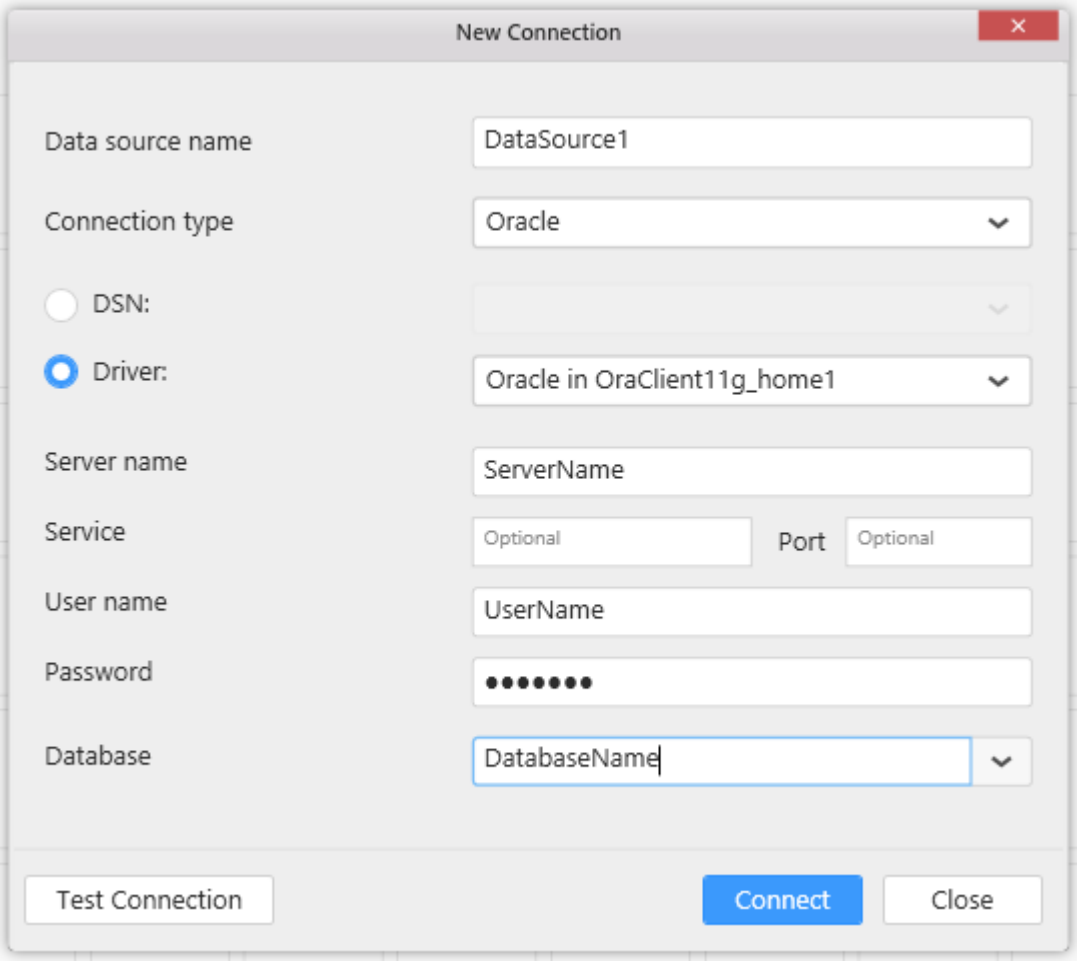
1. Navigate to the Control Panel -> System and Security -> Administrative Tools folder location.
2. Double click the ODBC Data Sources (32-bit) tool to open the ODBC Data Source Administrator (32-bit) dialog.
3. Select the System DSN or User DSN tab where the data sources already available get listed in the list. Click Add to select the specific driver and bind the preferred database to create a new ODBC data source.





Connecting to ODBC Oracle data source

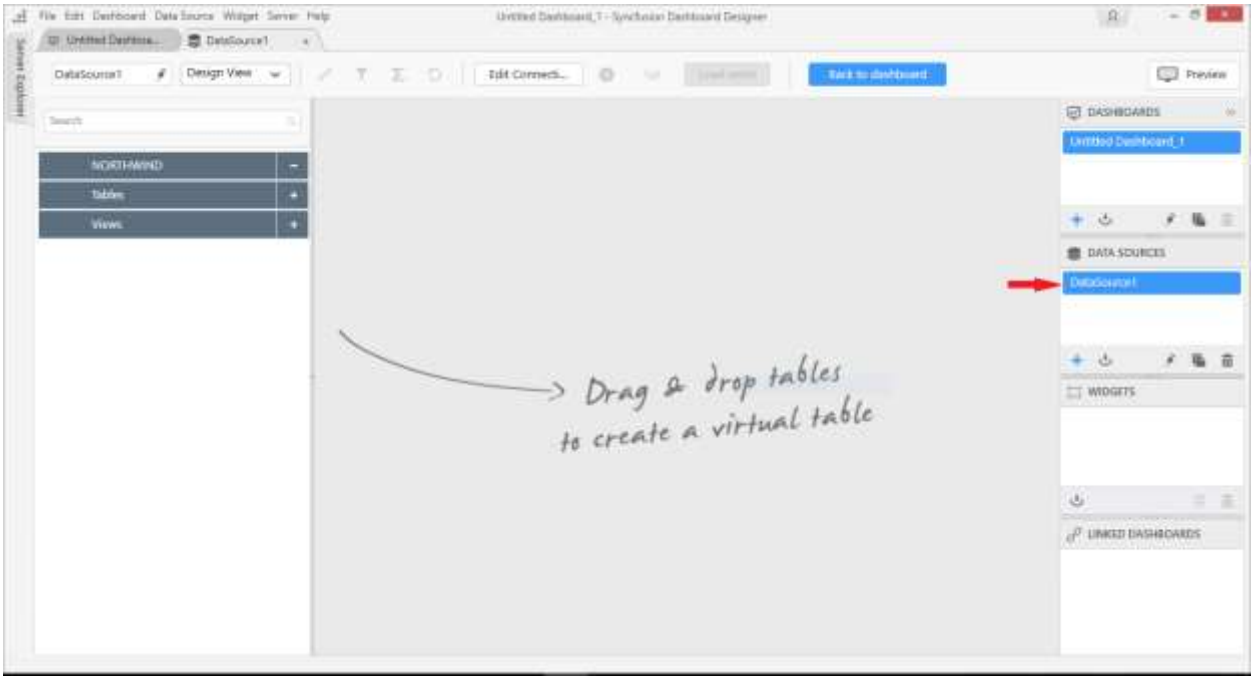
To establish Oracle connection, click on add data source which displays New Connection window as shown below.



1. Select Oracle as Connection type in the New Connection wizard.
2. Select Data Source Name (DSN) or the Oracle ODBC driver that you added, and bind the Oracle database path and Connect, to create a new data source for dashboard.

We can also test the connection by clicking on Test Connection button.

Click Connect to navigate to the data design view. It holds the tables, views, etc. available under the connected data source. Now, a new oracle data source was created.

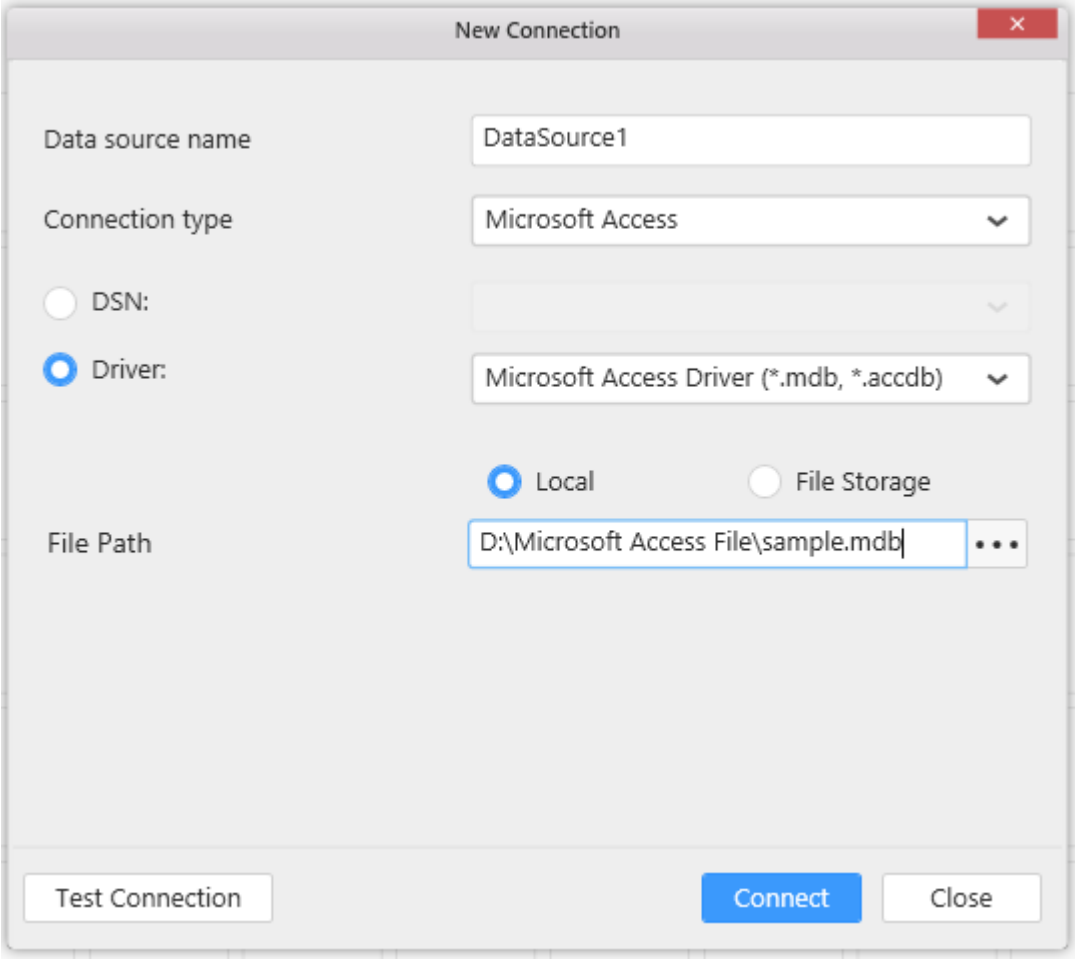


Note: For connecting Oracle Database through ODBC, ensure both Oracle driver and Syncfusion Dashboard Designer are installed on the same machine.

Using Oracle data connection type, you can connect Oracle 11g or Oracle 12c database.

Connecting to Microsoft Access Database through ODBC

Through Microsoft Access data connection type, you can connect to ODBC enabled Microsoft Access database file whose format can be either MDB or ACCDB.



Setting up an ODBC-enabled Access database

1. Download & Install the redistributable driver from here 2. Add the installed driver through the following procedure to setup a new data source with Access database:

a. Navigate to the Control Panel -> System and Security -> Administrative Tools folder location.

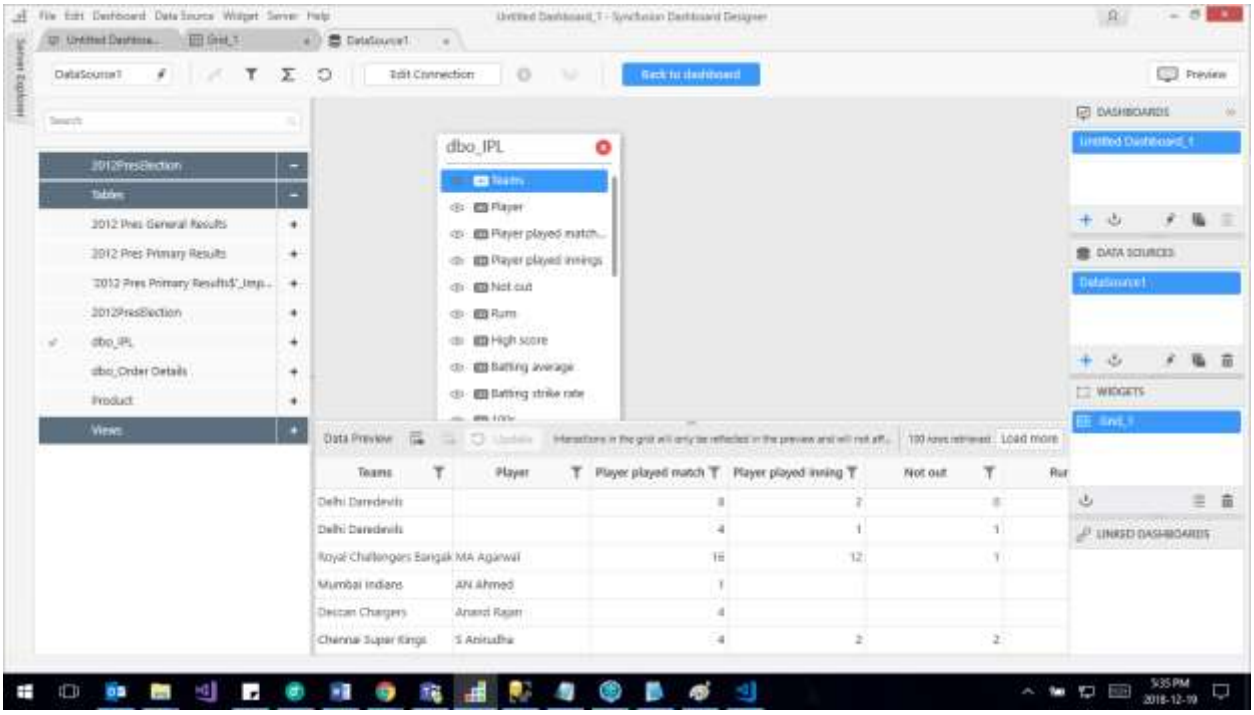
b. Double click the ODBC Data Sources (32-bit) tool to open the ODBC Data Source Administrator (32-bit) dialog.

c. Select the User DSN tab. In this tab, the user data sources already available get listed in the list. Click Add to select the specific driver and bind the preferred database to create a new ODBC data source.

Connecting to ODBC Access data source

1. Select Microsoft Access as Connection type in the New Connection wizard. 2. Select Data Source Name (DSN) or the Access driver that you added, and bind the Access database file path and connect, to create a new data source for dashboard.

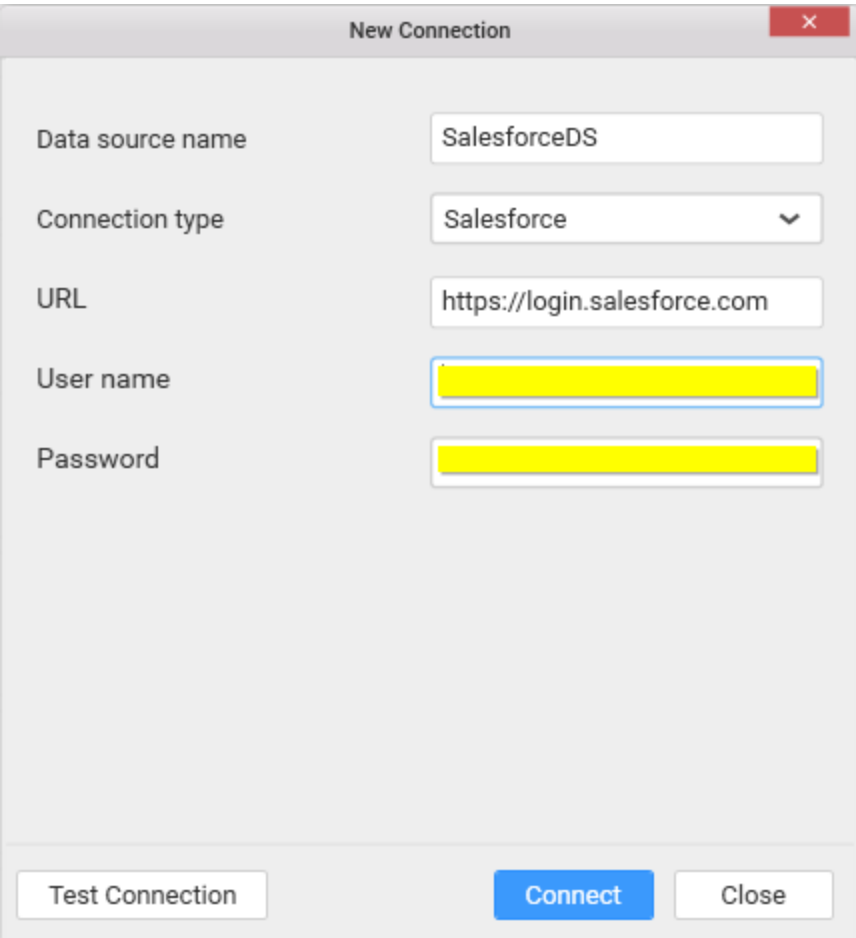
Now you will get into data design view window.



Note: For connecting Microsoft Access Database through ODBC, ensure both Access driver and Syncfusion Dashboard Designer are installed on the same machine.

Connecting to Salesforce

With Salesforce data connection type, you can connect the objects from your Salesforce account.



Set Connection type as Salesforce.

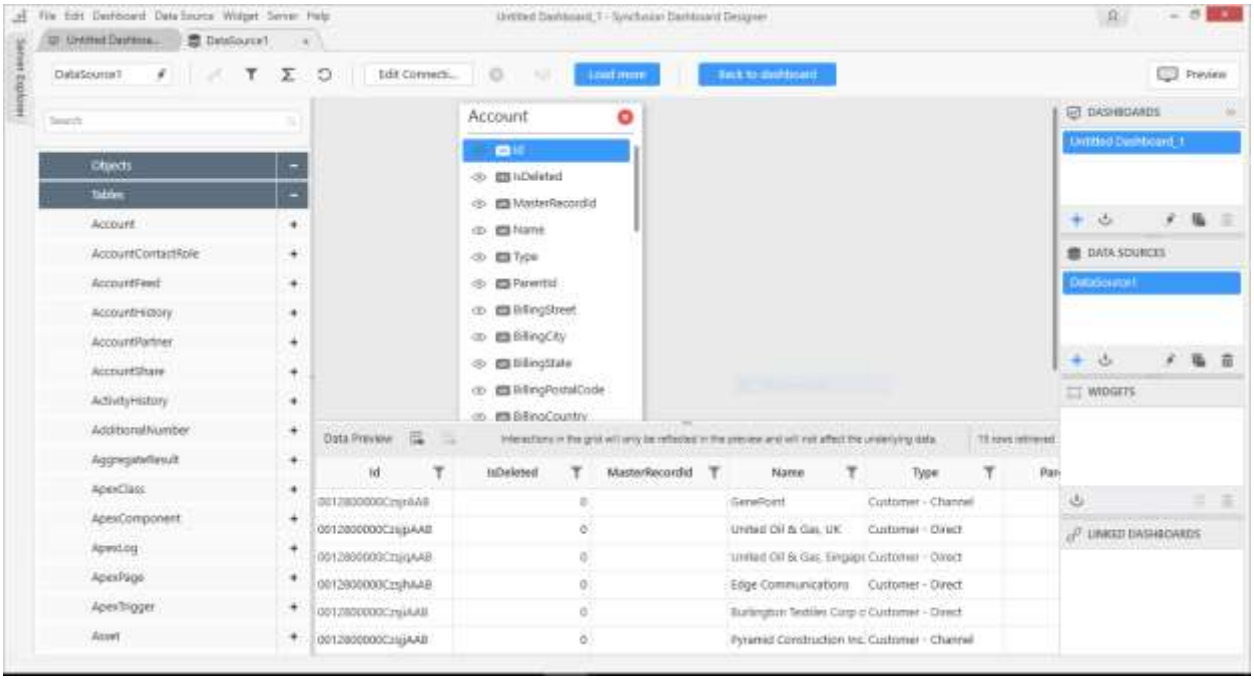
Set the URL for the Salesforce server you are connecting to. By default, the server is here

Fill your user name and password for Salesforce.com.

In order to Salesforce login, you may need to append a special security token to the end of your password.

To get the security token, refer How to obtain my security token.

Click Connect, now you will get into data design view window.



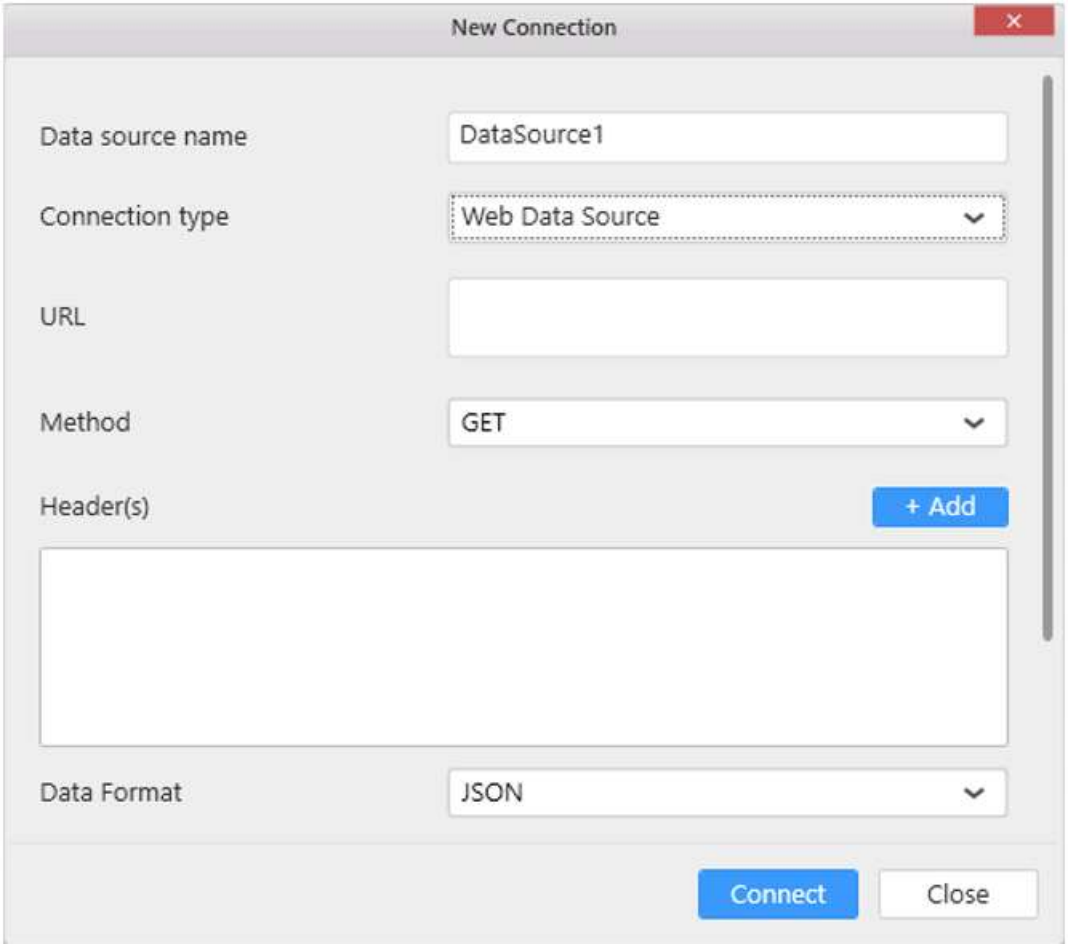
Salesforce Account Configuration

In order to connect Salesforce data, all of the following must be enabled on your Salesforce account.

- SOAP API – To sign in and get Salesforce objects information.
- BULK API – To retrieve data.
- REST API – To retrieve data that BULK API doesn't support.

Connecting to RESTful Web Services

With RESTful Web Services connection type, you can connected to data from web which is accessible through RESTful APIs.



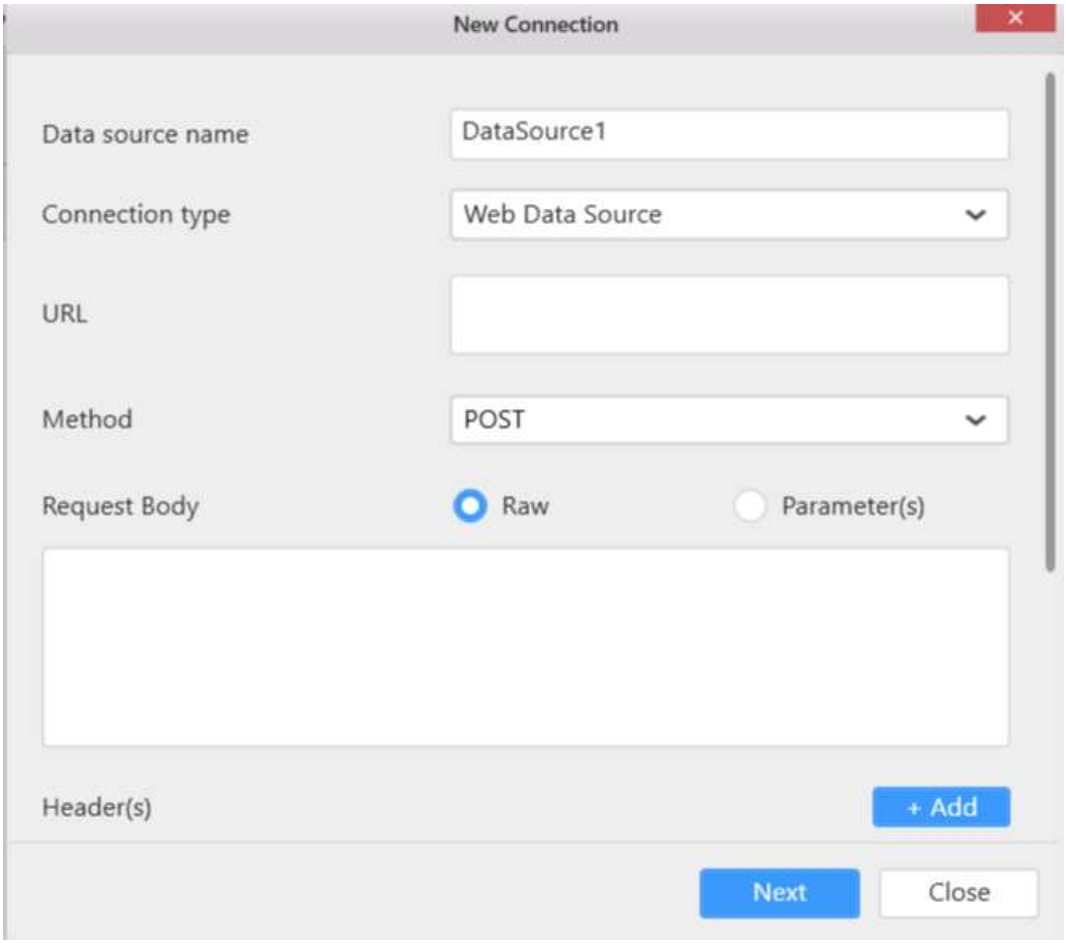
Set Connection type as Web Data Source.

Enter the API in URL text box which must be a valid REST API.

Choose an appropriate Method for the REST API; it can be either GET or POST in the type combo box.

When you choose POST method, Request Body section will be displayed.

Request Body allows to send data request to the server to update it: as often the case with POST requests. It can be chosen either Raw or Parameter(s).



For Example, Consider the table below,

Raw	Parameter(s)
{ “status”: “Hello Everyone,” }	Status: Hello Everyone,

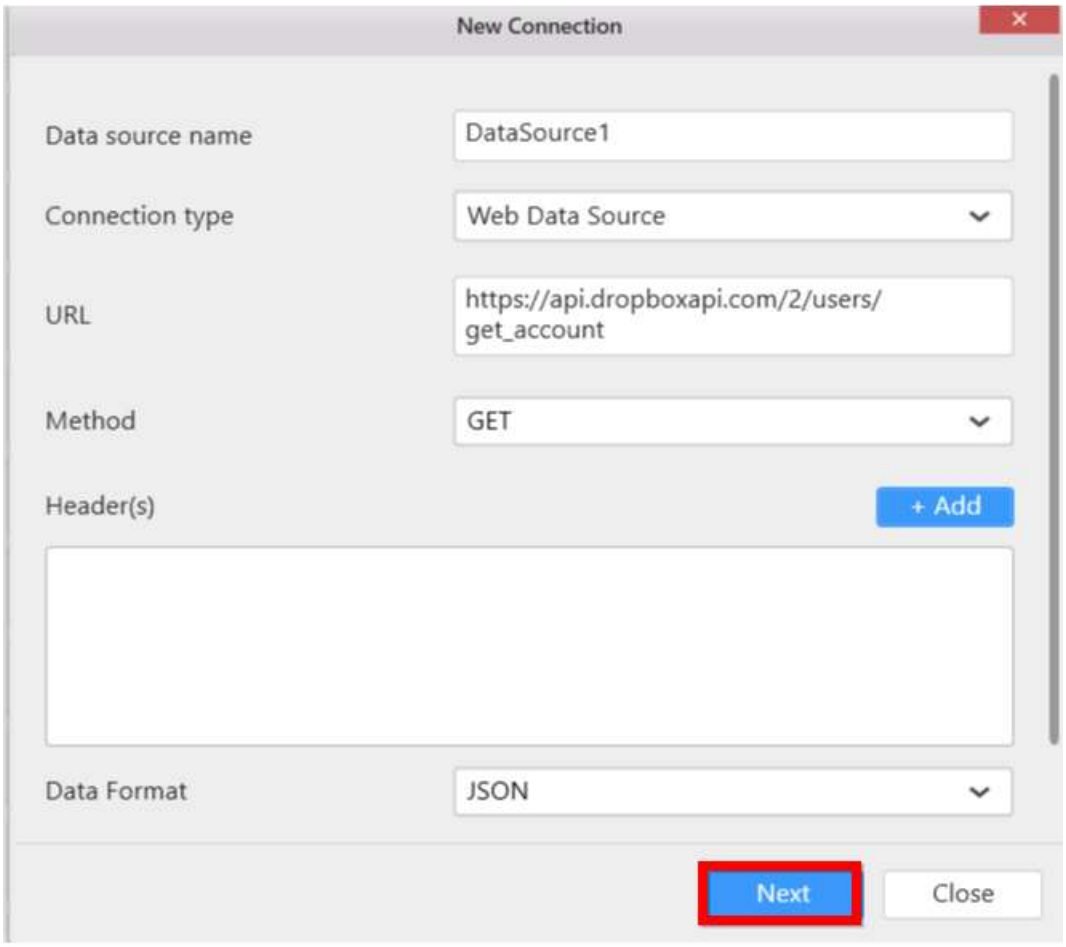
Header(s) allows the client and the server to pass additional information with the request or the response.

Example: Content-Type: application/json

You can choose your response Data Format. We have supported the JSON and CSV data format.

You can choose your Authentication type for the REST API from drop down list,it can be either Basic HTTP Authentication or No Authentication or OAuth.

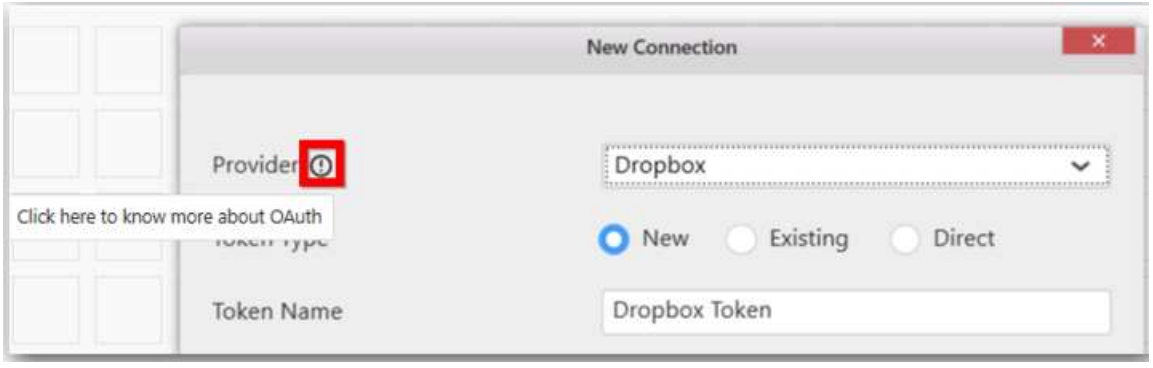
After filling required details, click Next button in the New Connection window as shown below.



OAuth

Through OAuth authentication type, you can get Web API from some providers like Facebook, Twitter, LinkedIn, Google, Yahoo, Instagram, GitHub and Custom.

Note: You can click information icon shown below to know more details about OAuth.



Here, we have taken Dropbox as example to explain OAuth Authentication type.

You can choose Provider from drop down list.

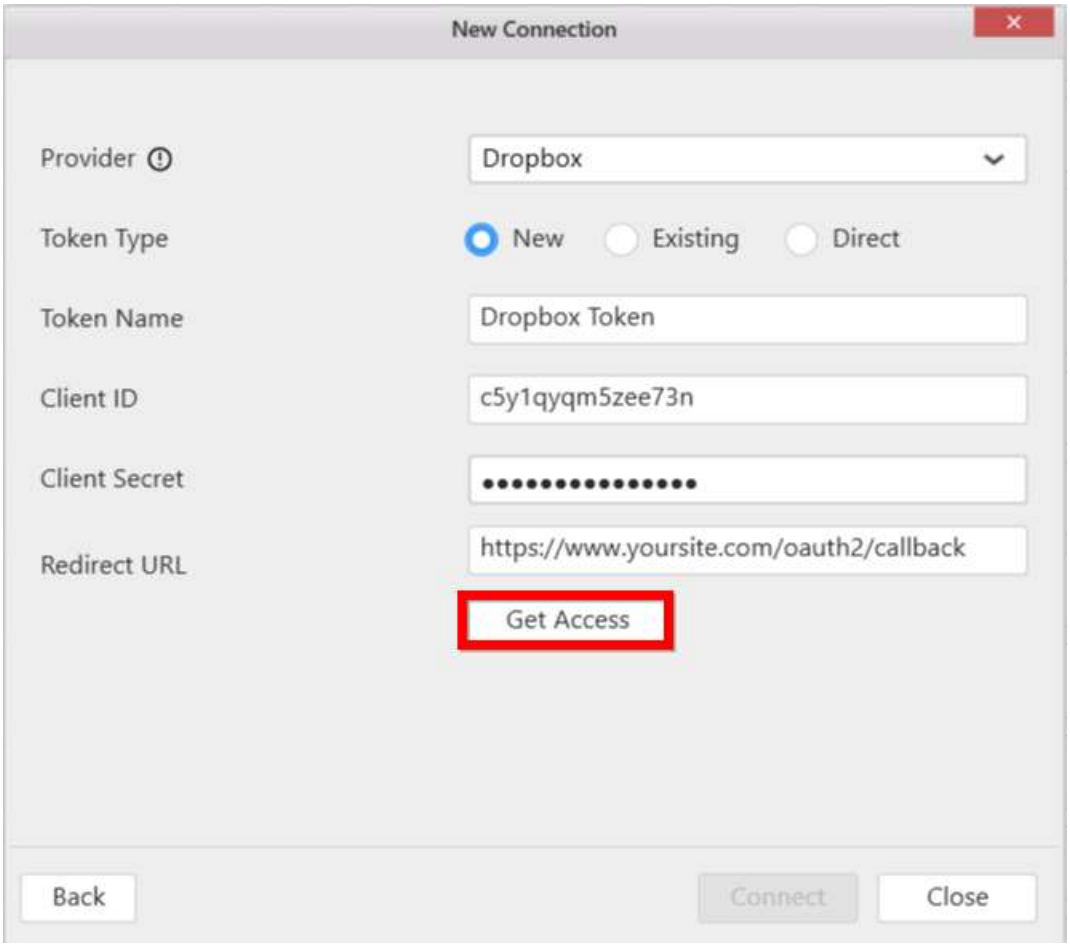
You can choose Token Type, we have provided the three kinds of Token type listed in below,

1. New Access Token. 2. Existing Access Token. 3. Direct Access Token.

New Access Token

Enter Token Name, Client ID, Client secret and Redirect URL for selected provider type.

Click Get Access button to get access token.



Note: After getting the Access Token, the alert message shown in bottom and the connect button will enable.

Existing Access Token

You can reuse the token by choosing Existing Access Token, which is already created by using the New Token type.

You can choose created token from Select Token drop down list.

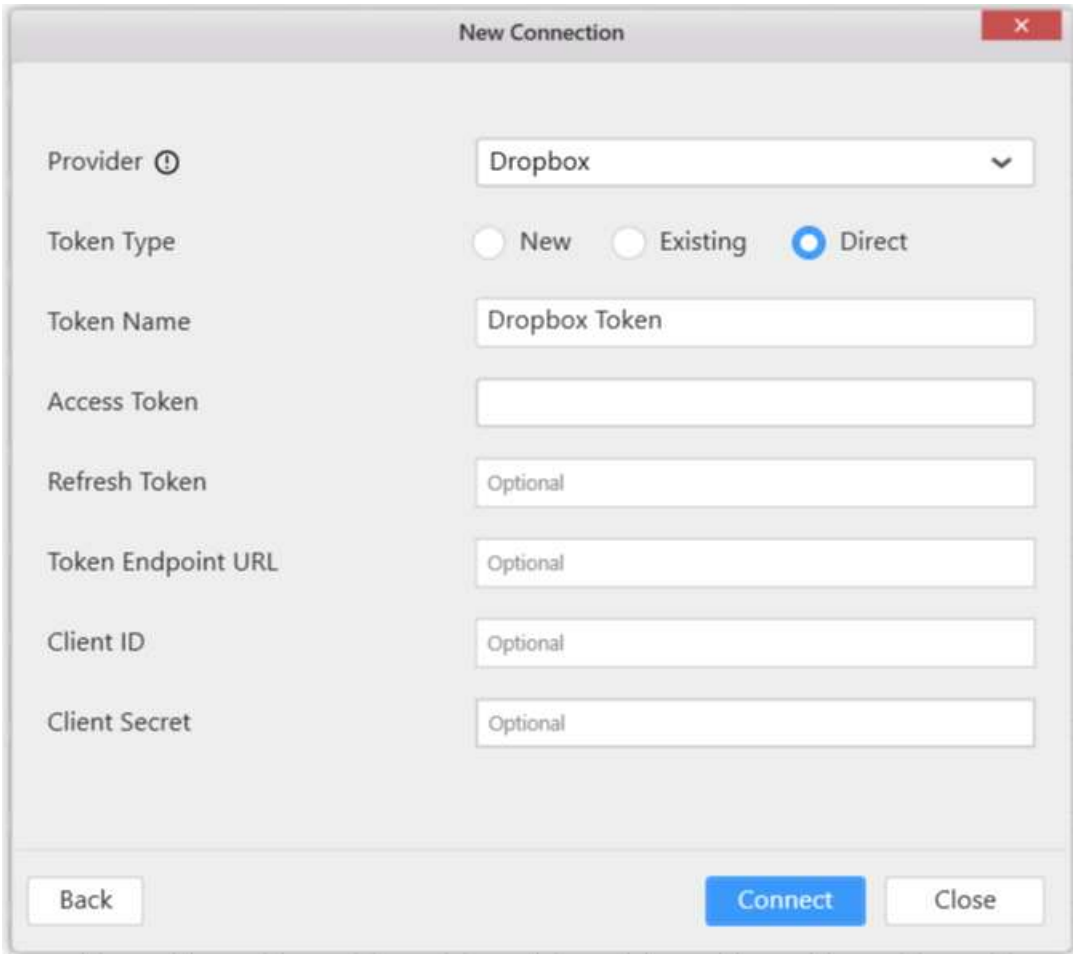
Click Connect button to get the data.

Direct Access Token

We have provided the support to get data from your own access token. If you have direct access token for your registered application, you can avoid the process for getting the access token with our application. You can connect directly through choosing the Direct Access Token option which is available in Token Type selection radio button.

Enter valid Access Token for the provider type.

Here, the Refresh Token, Client ID, Client secret and Token Endpoint URL are optional parameters which used only for refreshing the access token.

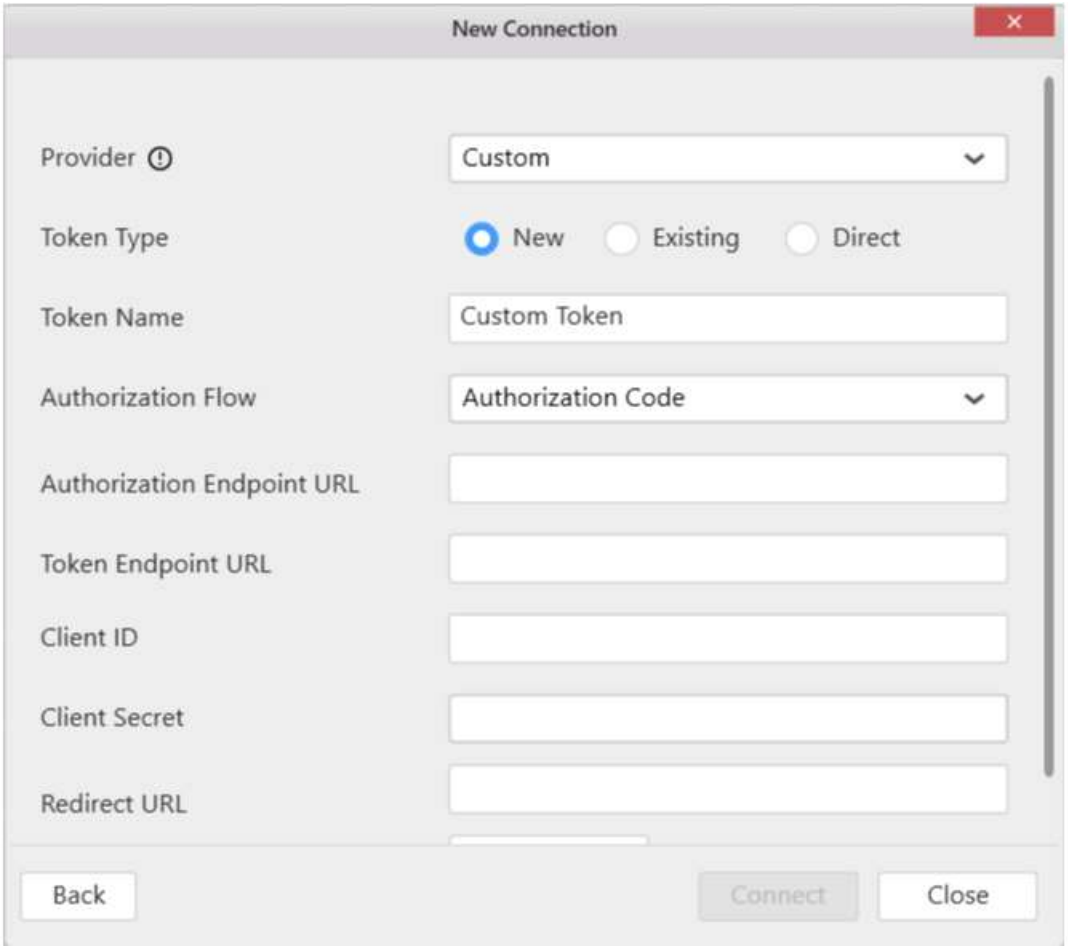


Click the Connect button.

If your URL returned data in JSON format, then JSON Schema Designer window will open.

Custom OAuth

Custom option available in our web data source OAuth provider selection drop down list shown below.



Here, we have provided four kinds of Authorization Flow (Grant Type) for OAuth Custom provider type.

1. Authorization Code. 2. Implicit. 3. Password. 4. Client Credential.

Authorization Endpoint URL is the endpoint for authorization server, which retrieves the authorization.

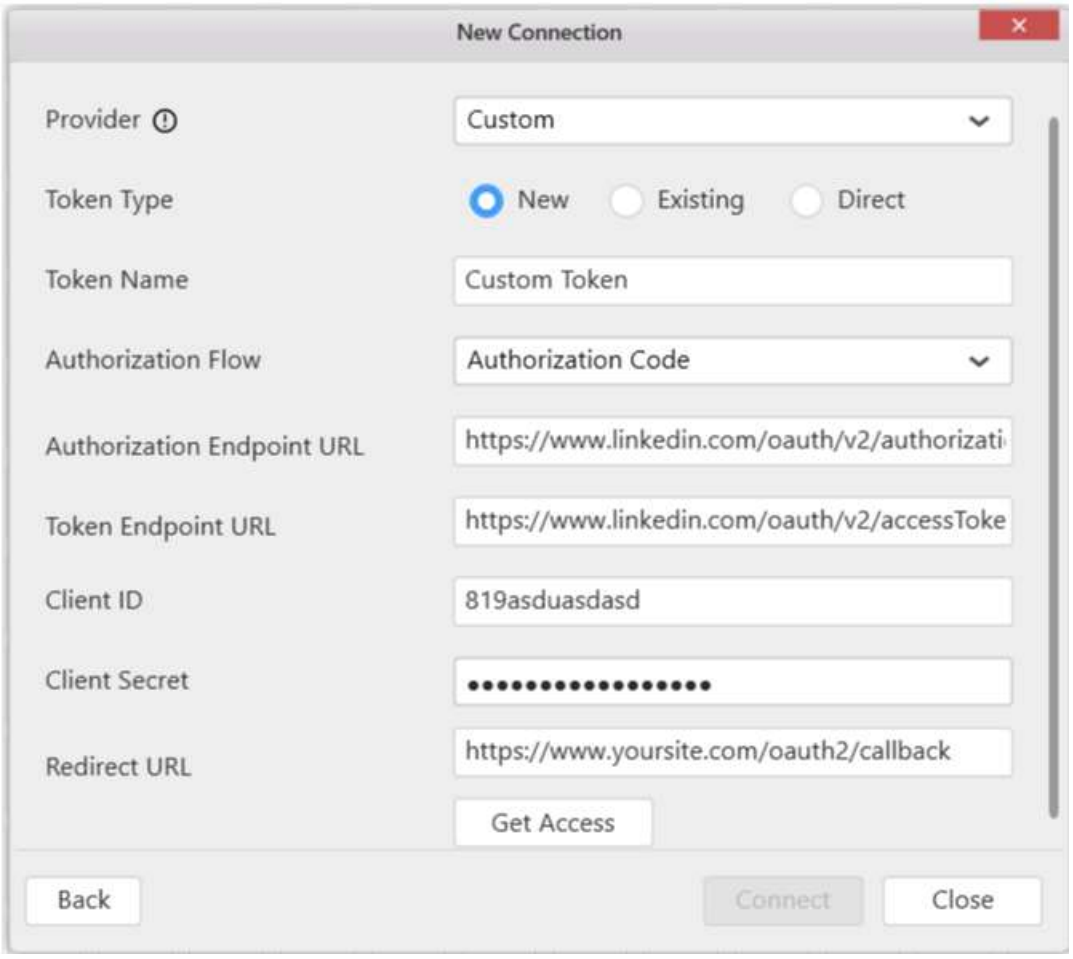
Token Endpoint URL is the endpoint for the resource server, which exchanges the authorization code for an access token.

Enter your registered ClientID and Client Secret.

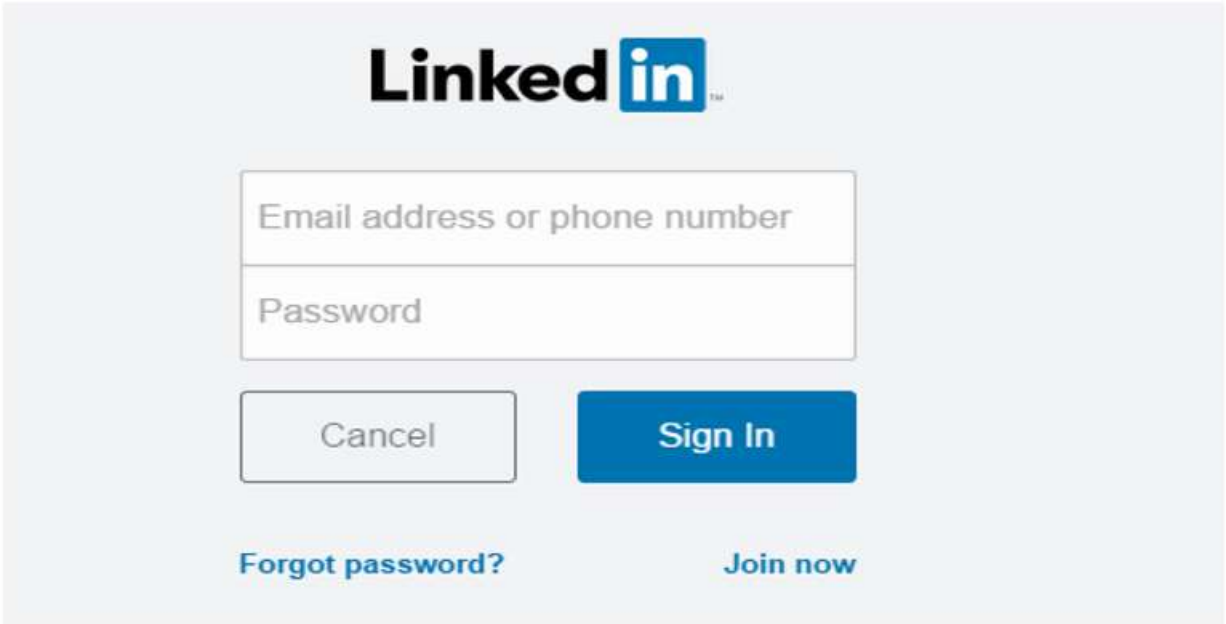
Redirect URL allows to redirect which you provided in your application registration process.

Enter User Name and Password for your web service.

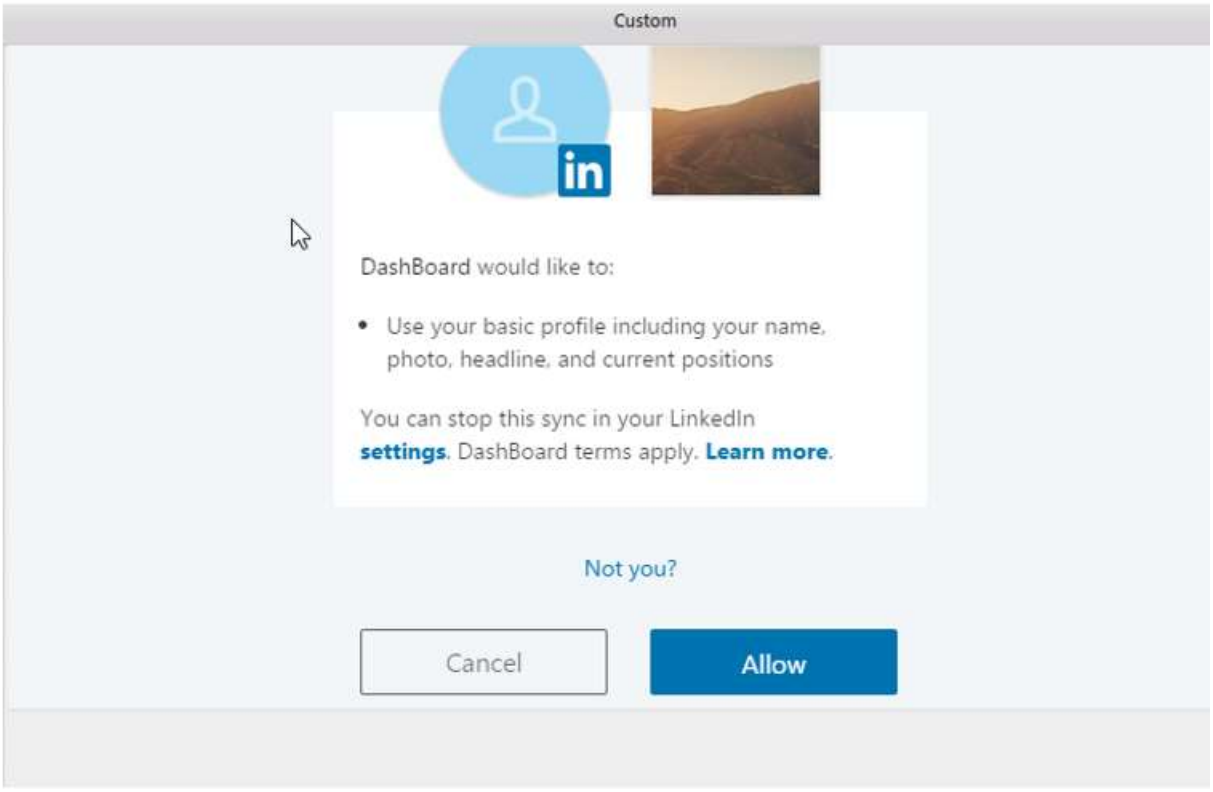
After filling the all required fields, you can click the Get Access button



It will navigate to your service login window. Login your account with credentials.



The authorization server will present them with a prompt asking if they would like to authorize this application's request.

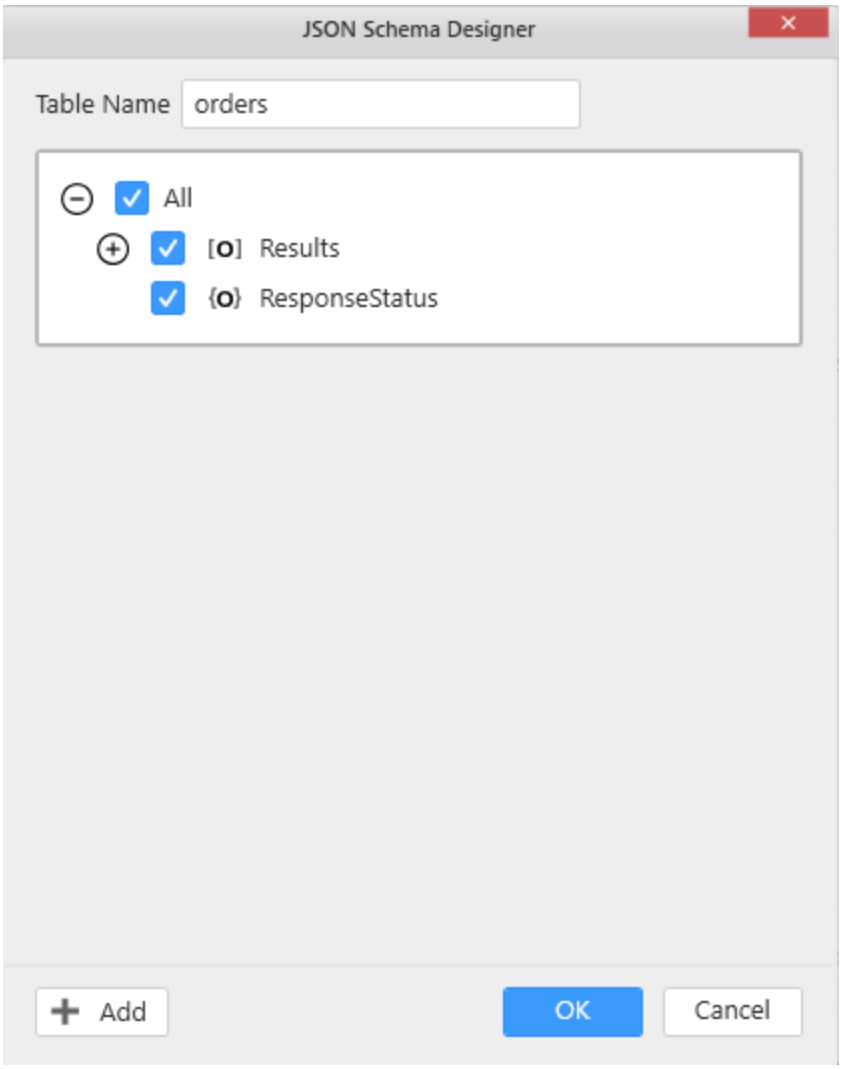


Here, the application required the permission to get your basic profile information. It can be defined through the Authorization URL's Scope section.

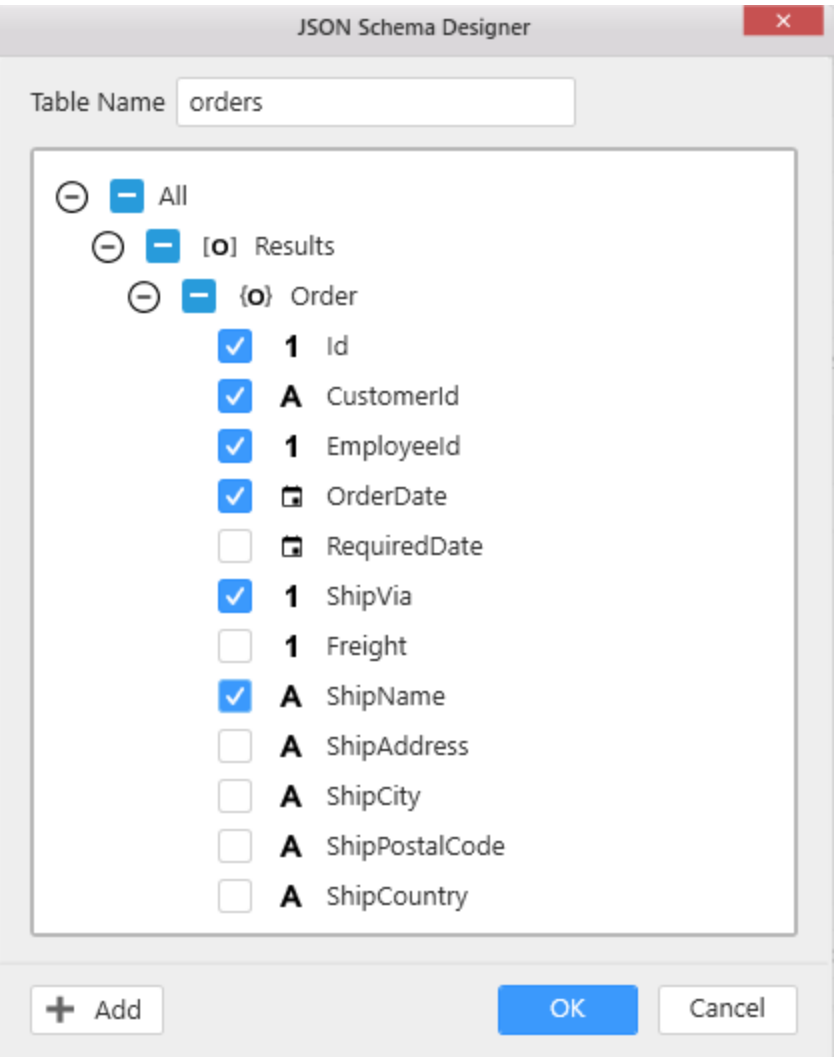
After getting access token , Connect button will be enabled and you can get data from your provided Rest API.

JSON Schema Designer

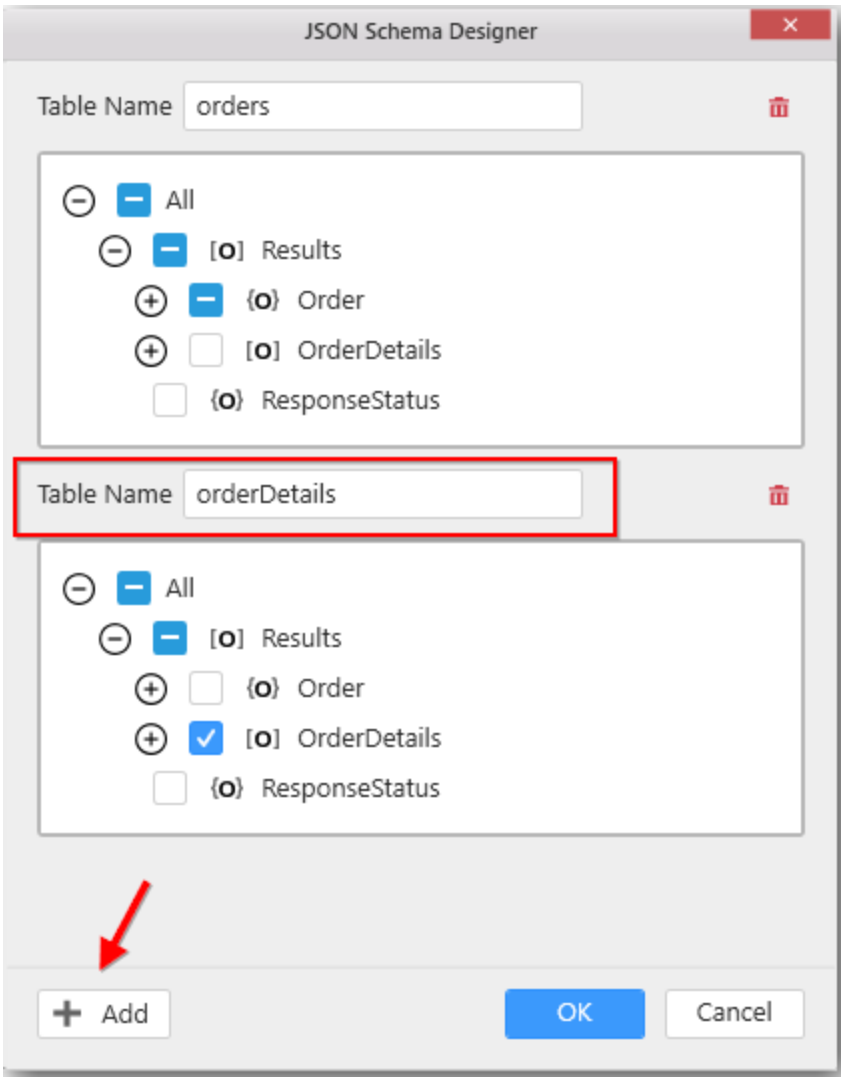
The JSON Schema Designer is used to design one or more than one table based on JSON schemas in your JSON string.



In the JSON schema designer, all JSON schema information are fetched from the JSON string and all JSON schema are listed in hierarchical order to design one or more than one table. To design a table, you can select any schema which you want to use in that table.

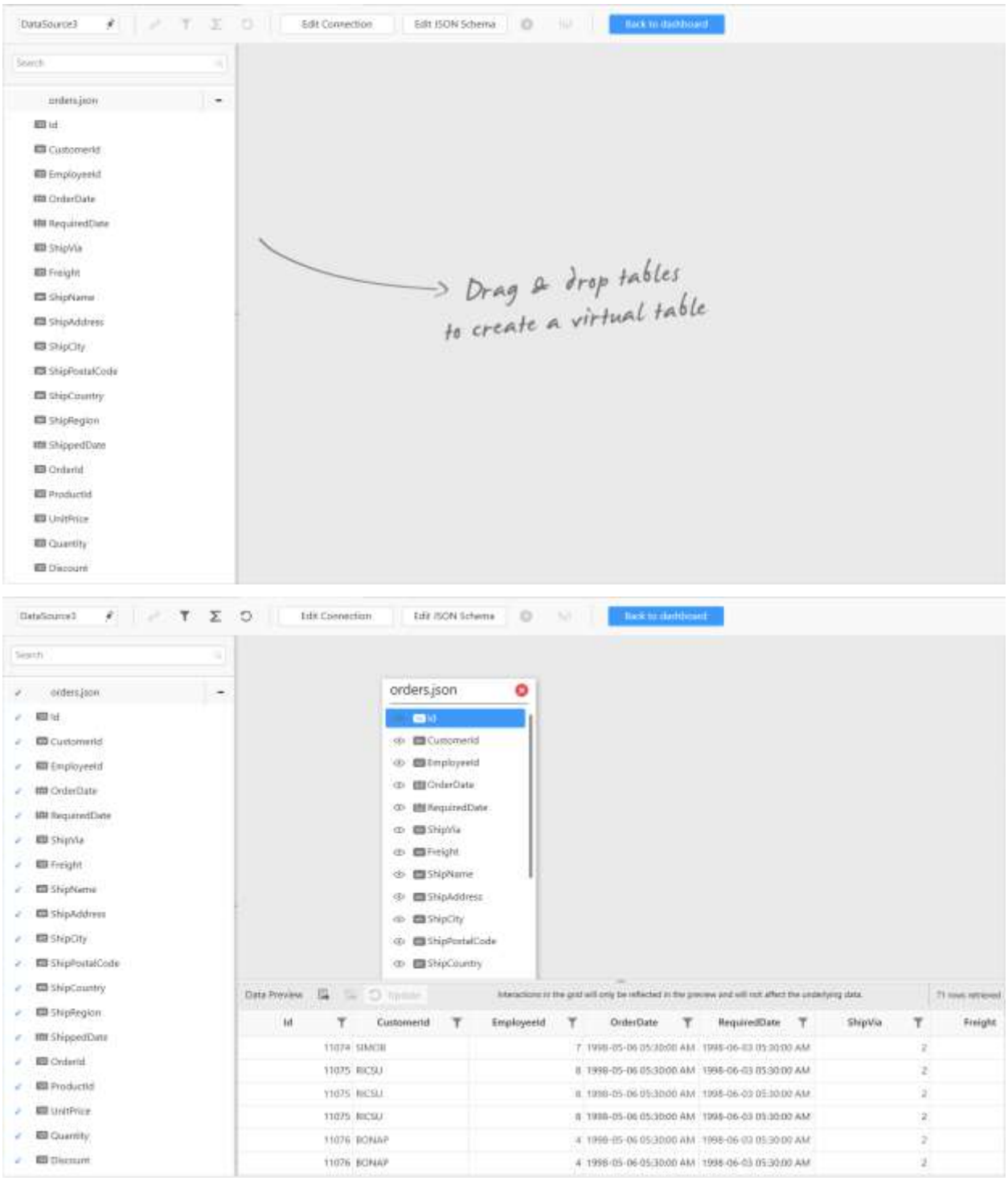


If you have multiple tables in DBMS (Database Management System) structure in the JSON string, then you can use Add option to design the multiple tables as follows.



After designing the table(s) using the JSON schema, click OK.

Now the designed table(s) in JSON Schema Designer will generate table(s) which will have columns and rows as follows.



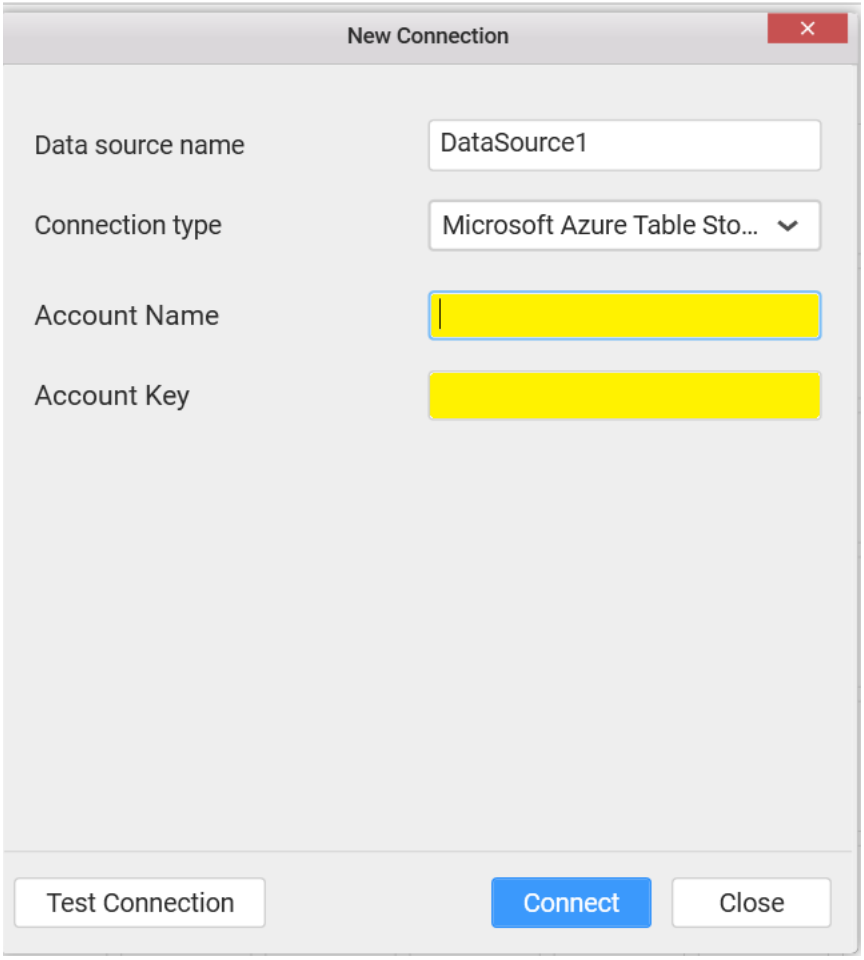
Note: 1. With our Web Data Source connection type, you can also interact with Entity Framework data layer through RESTful HTTP services that were built using ASP.NET Web API.

2. Currently, our dashboard designer supports HTTP Get requests and does not have support for HTTP Post requests since it requests additional data from client to server in the message body like JSON, XML, TEXT etc.

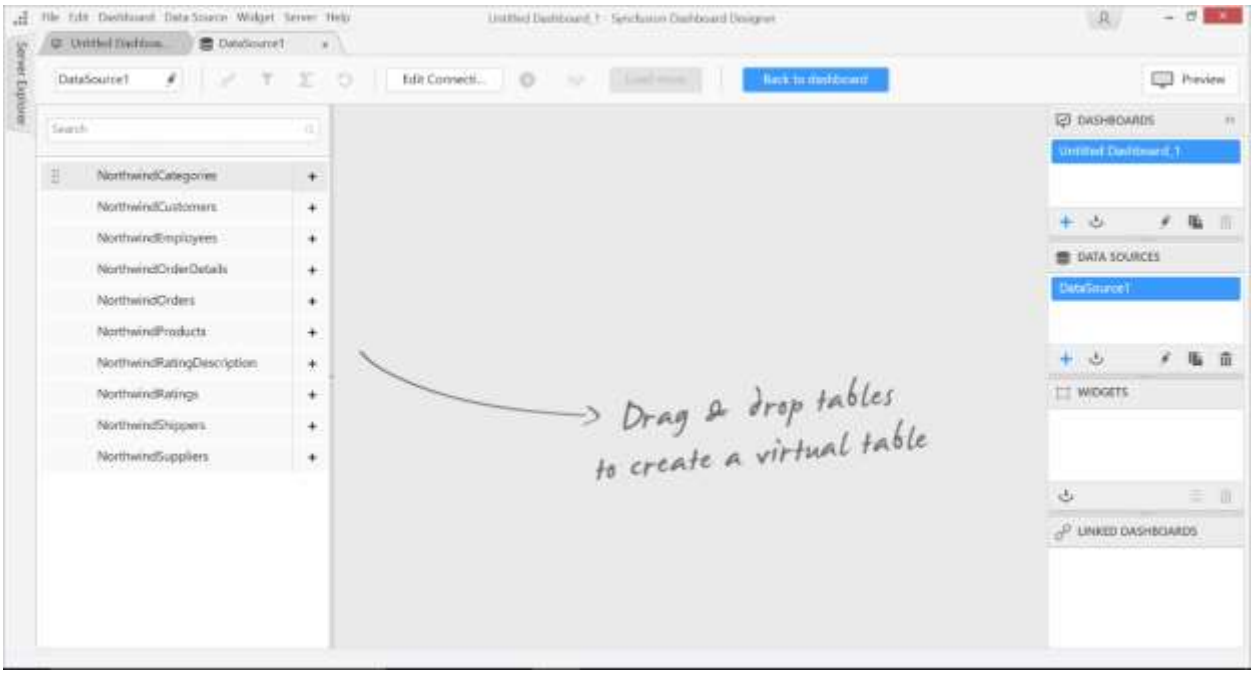
3. The Column names in the REST API response must be unique.

Connecting to Microsoft Azure Table Storage

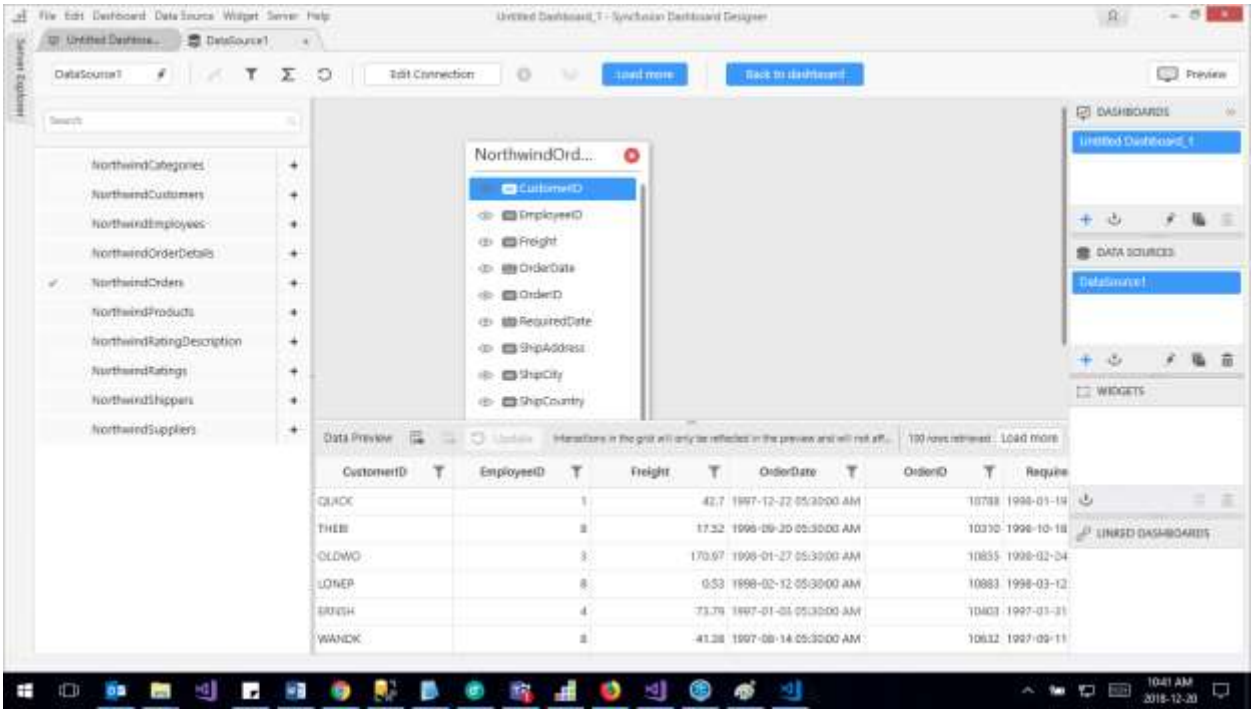
To create a new data source select the Microsoft Azure Table Storage option from the connection type.



Provide the Account Name and Account Key about the data source and then proceed to test the connection. Finally, on clicking the Connect button a new Data source with the given name will be created.

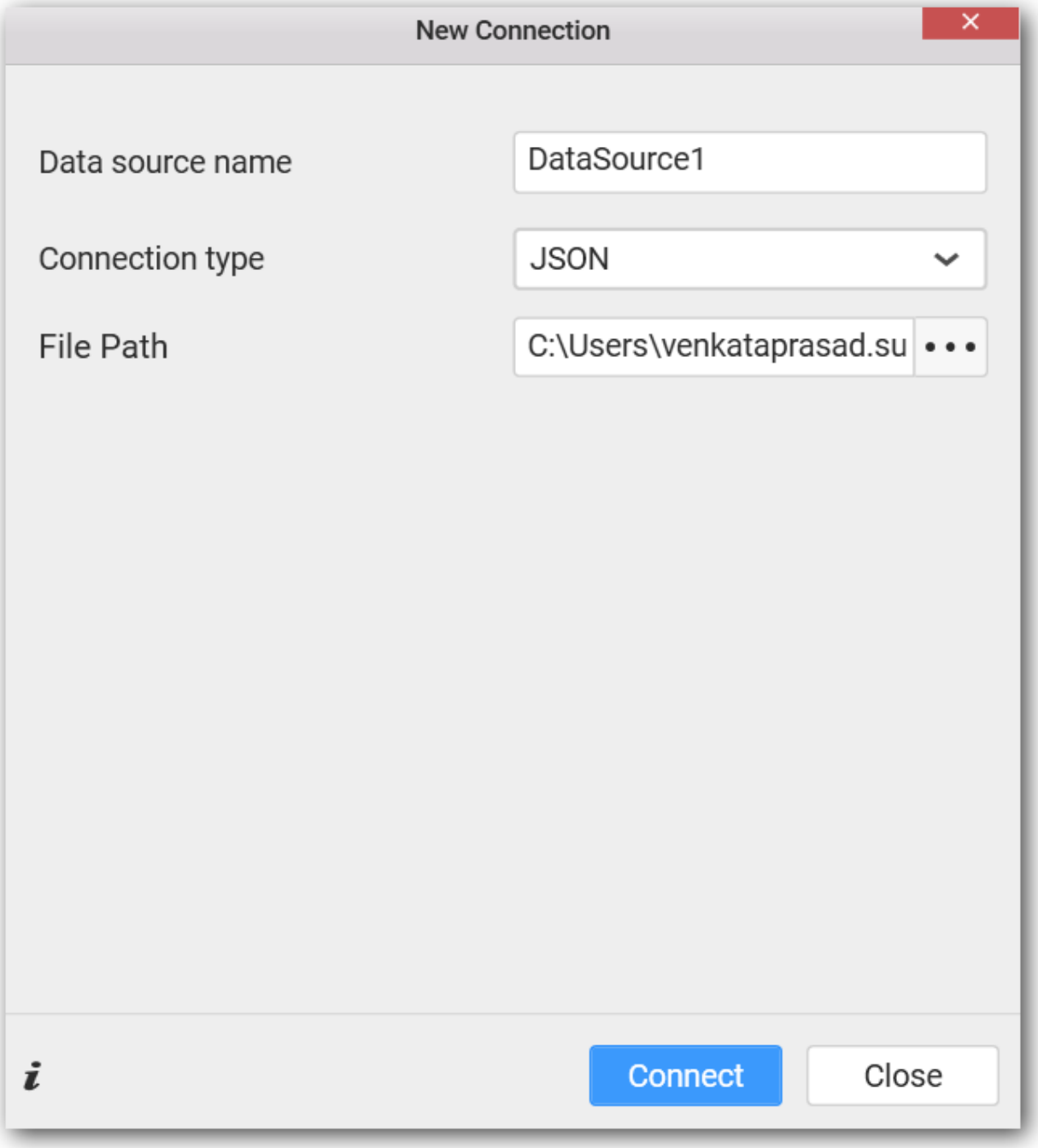


Dragging and dropping tables listed on the left pane to the center pane creates a de-normalized virtual table that will be used for creating your dashboard. The dashboard designer application joins multiple tables by automatically detecting the related fields but it can be manually defined. A preview of the transformed data is shown in the data grid.



Connecting to JSON

With JSON data connection type, you can connect to JSON file whose format will be JSON.



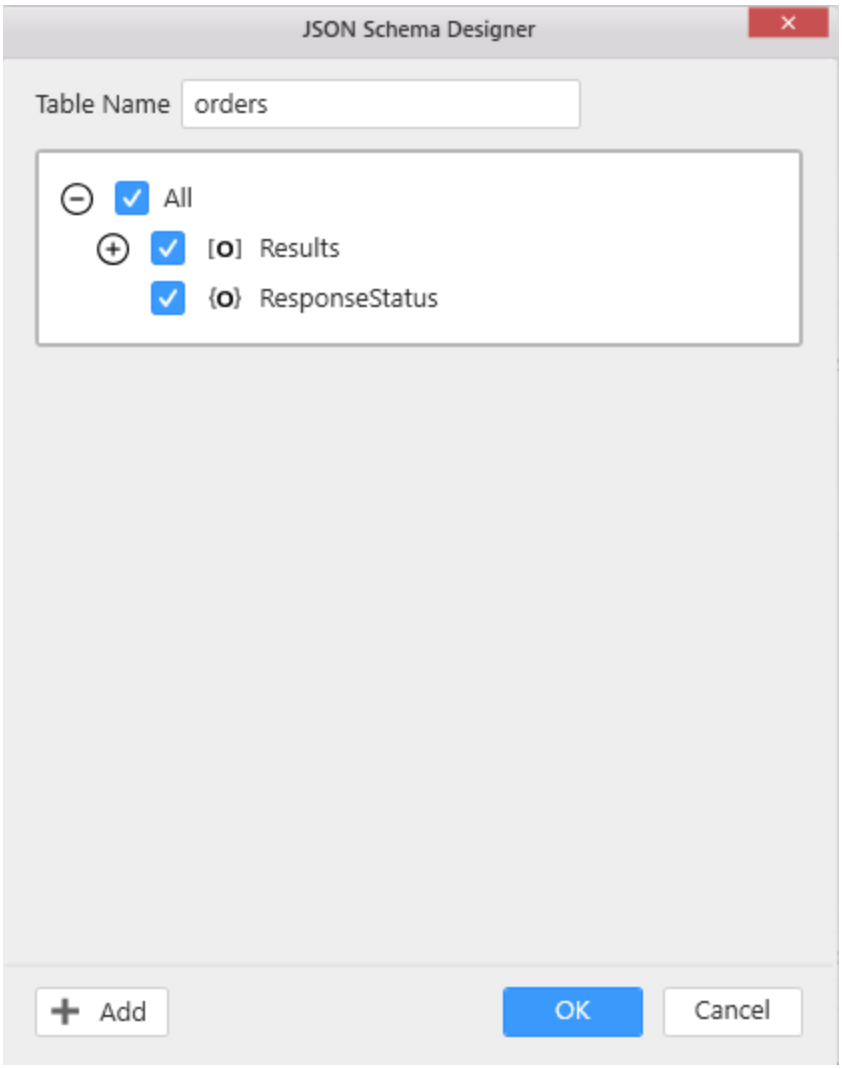
Set Connection type as JSON.

Set the Database Path by locating the JSON file existing in your machine accessible location.

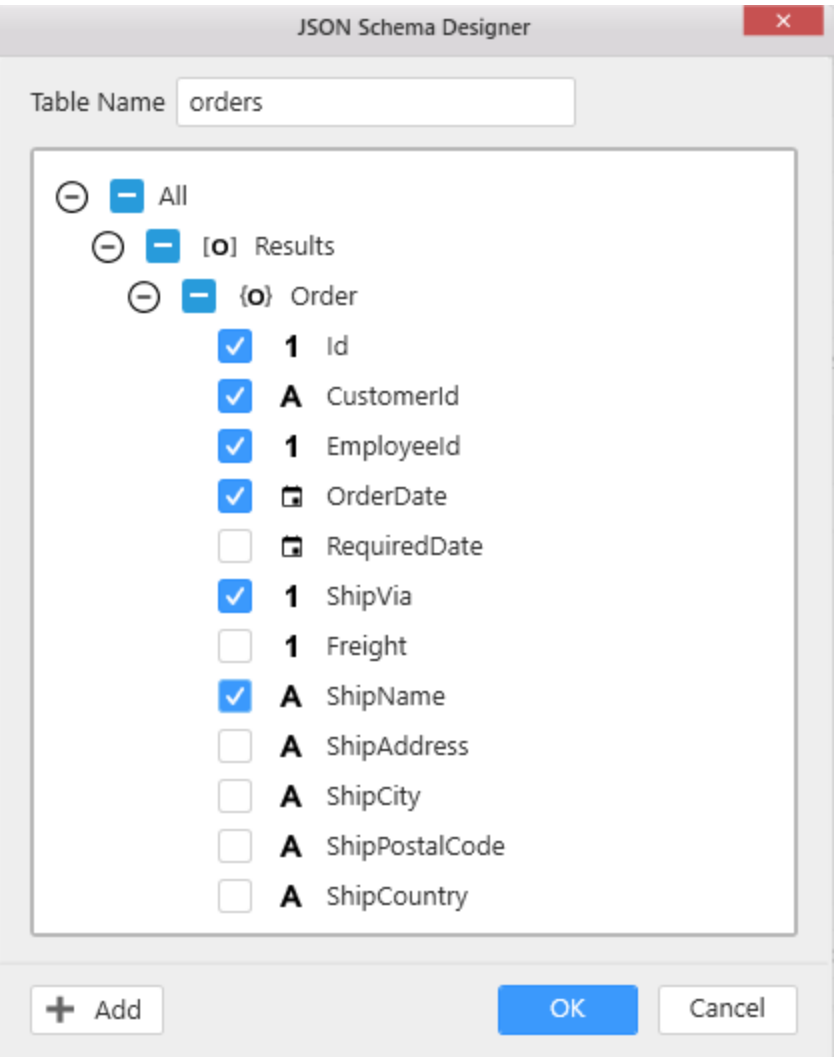
Click the Connect button, the JSON Schema Designer window will open.

JSON Schema Designer

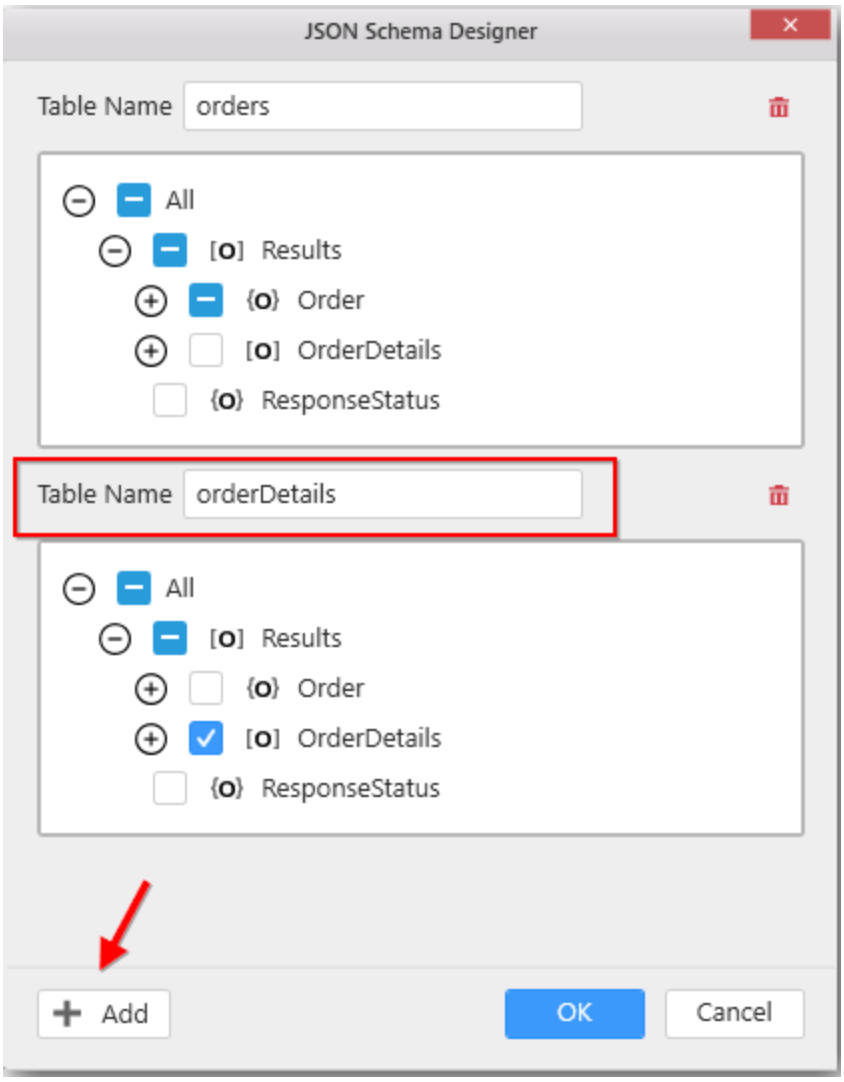
The JSON Schema Designer is used to design one or more than one table based on JSON schemas in your JSON string.



In the JSON schema designer, all JSON schema information are fetched from the JSON string and all JSON schema are listed in hierarchical order to design one or more than one table. To design a table, you can select any schema which you want to use in that table.

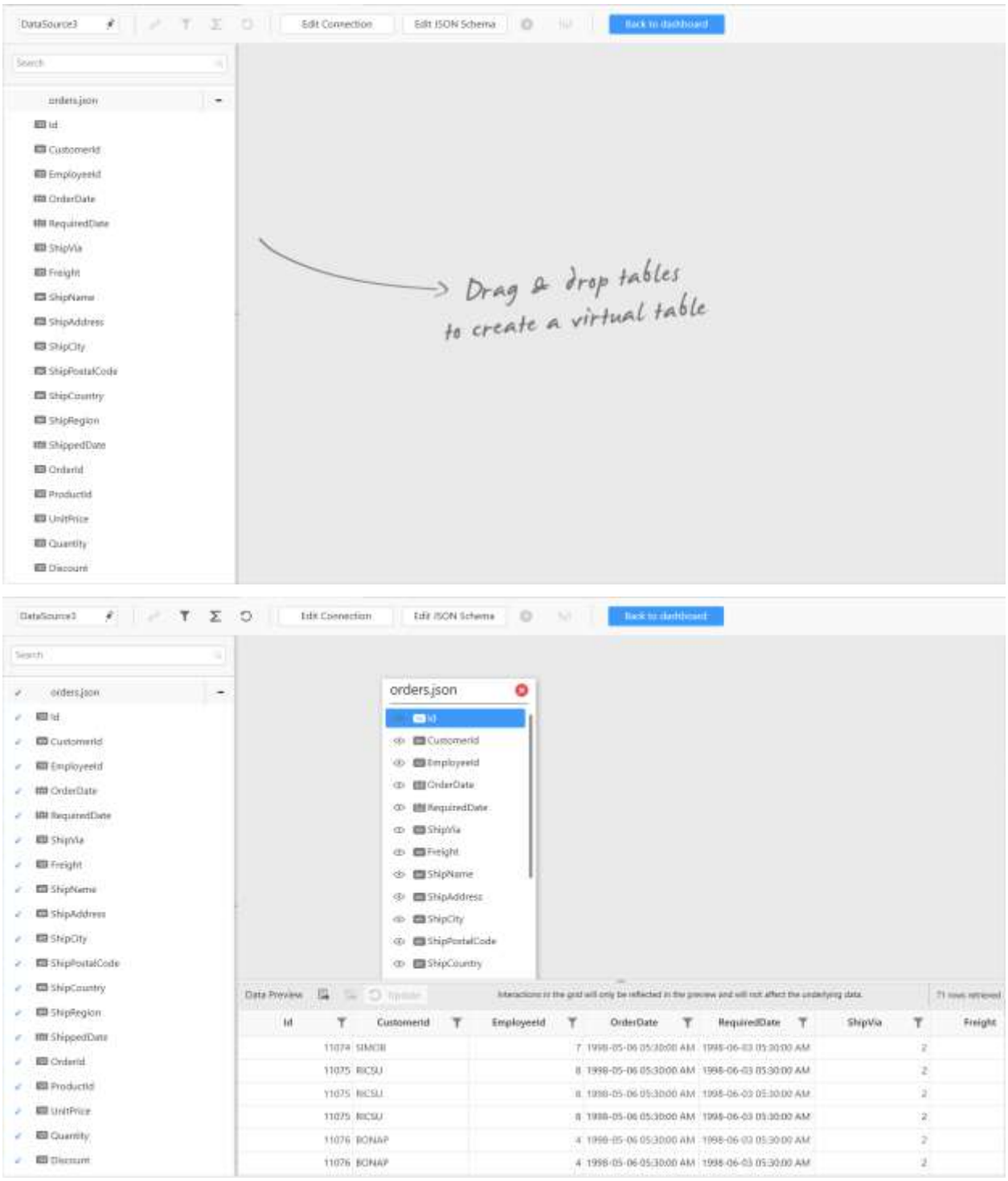


If you have multiple tables in DBMS (Database Management System) structure in your JSON string, then you can use Add option to design the multiple tables as follows.



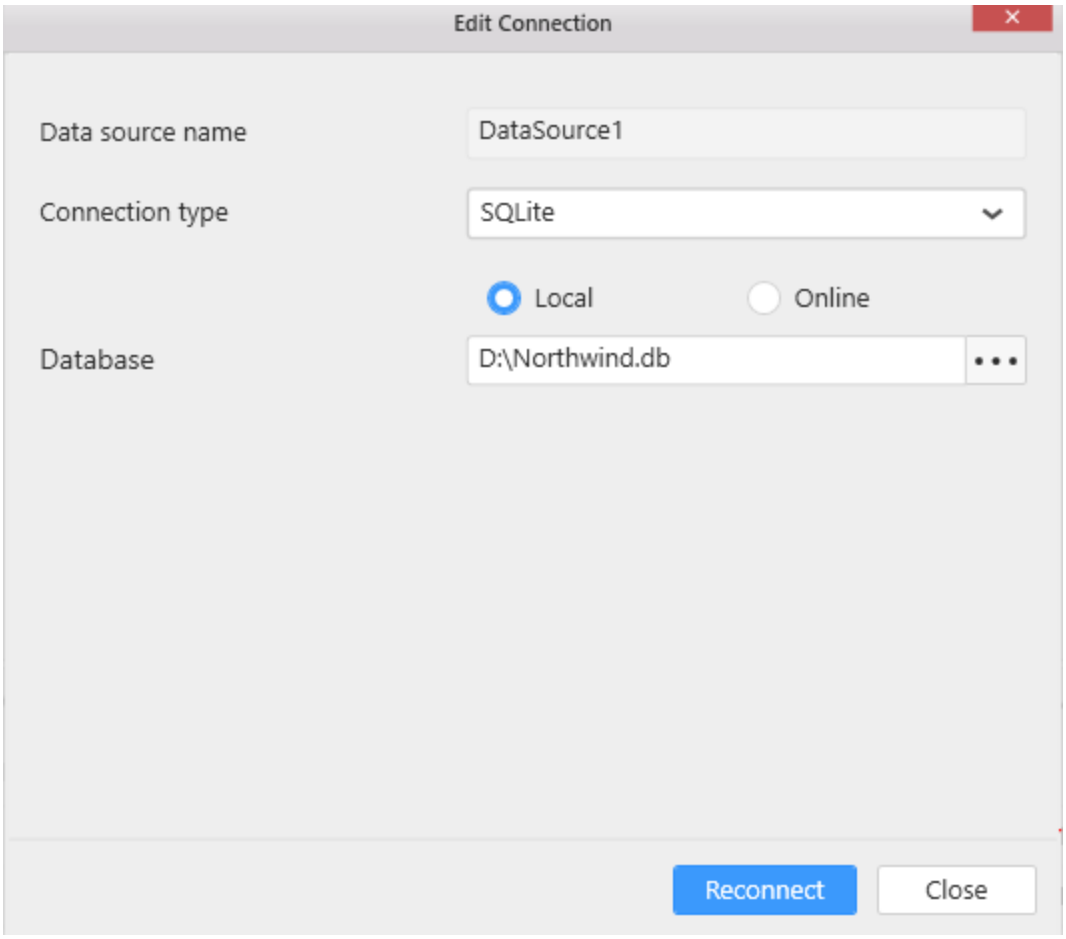
After designing the table(s) using the JSON schema, click OK.

Now the designed table(s) in the JSON Schema Designer will generate table(s) which will have columns and rows as follows.



Connecting to SQLite Database

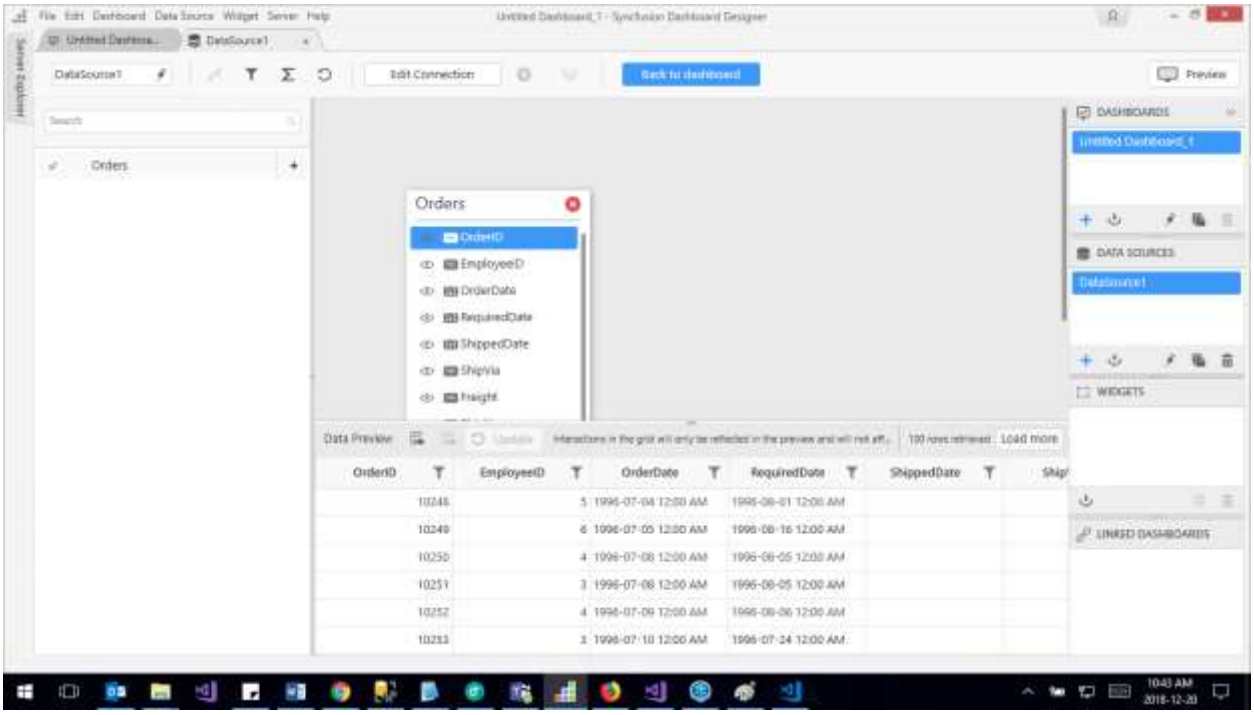
With SQLite data connection type, you can connect to SQLite database whose format will be in DB.



Set Connection type as SQLite.

Set the Database through locating the SQLite Database file existing in your machine accessible location and click Connect.

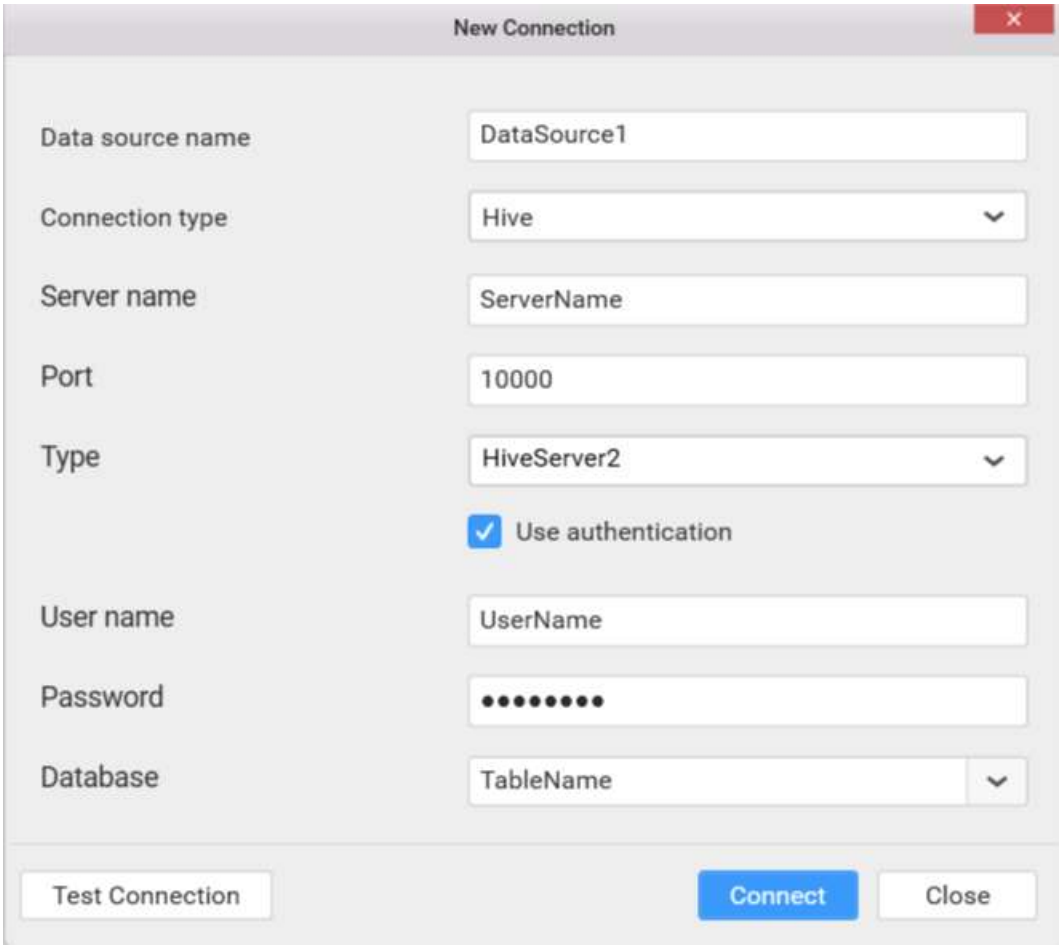
Now you will get into data design view window.



Note: If you have only one table in the bounded database, the data design view will have that one table added by default.

Connecting to Hive Data

With Hive data connection type, you can connect to data placed in Hive Server.

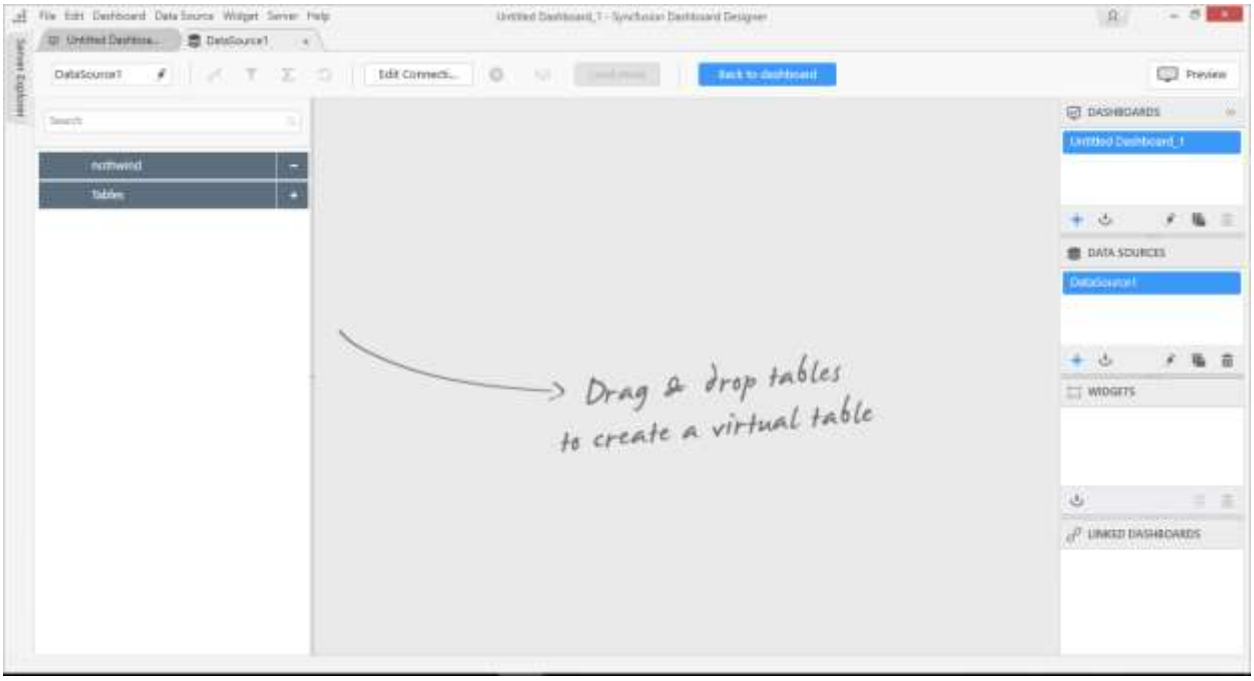


Set Connection type as Hive.

Set the Server name which can be either IP address or the host name of the hive server.

The Port number and the Type will be filled by default as 10000 and HiveServer2 respectively.

Select the Database to connect to and test the connection. If it succeeds, you will get into the data design view like below.



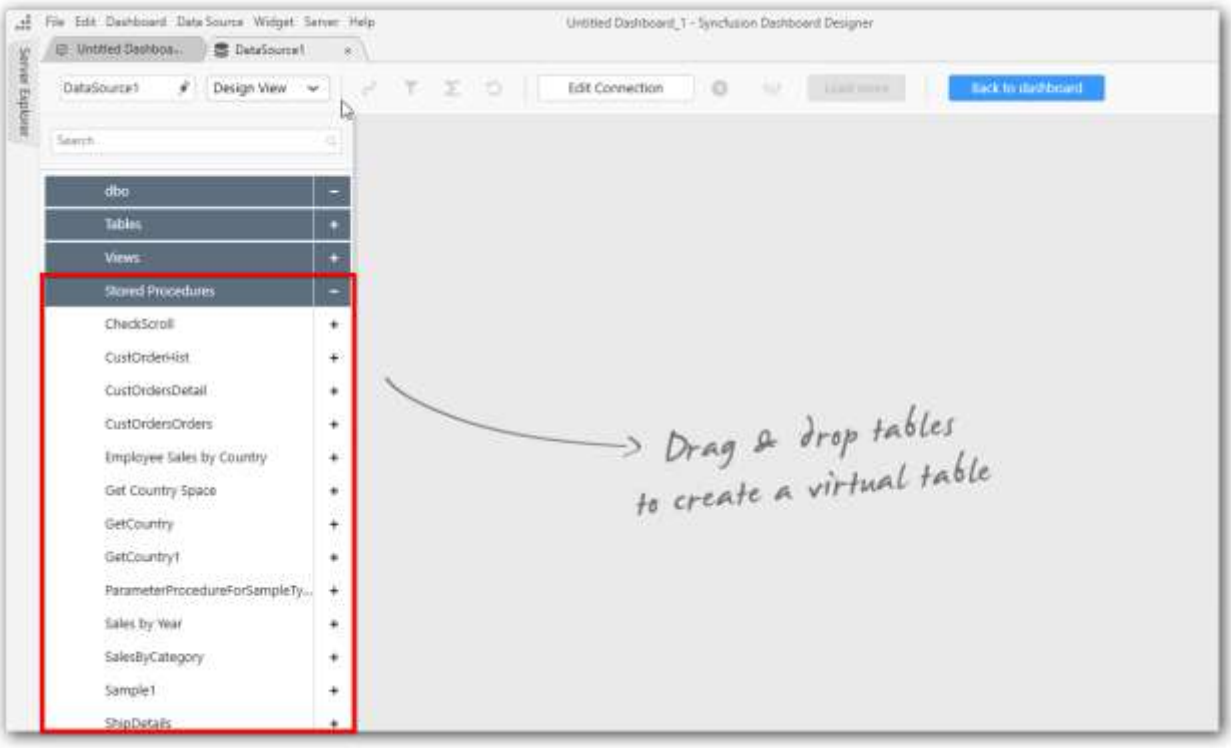
Connecting to Stored procedures in SQL Server Database

Syncfusion Dashboard Designer allows you to use stored procedures defined in SQL Server database either through direct SQL Server connection or through ODBC connection.

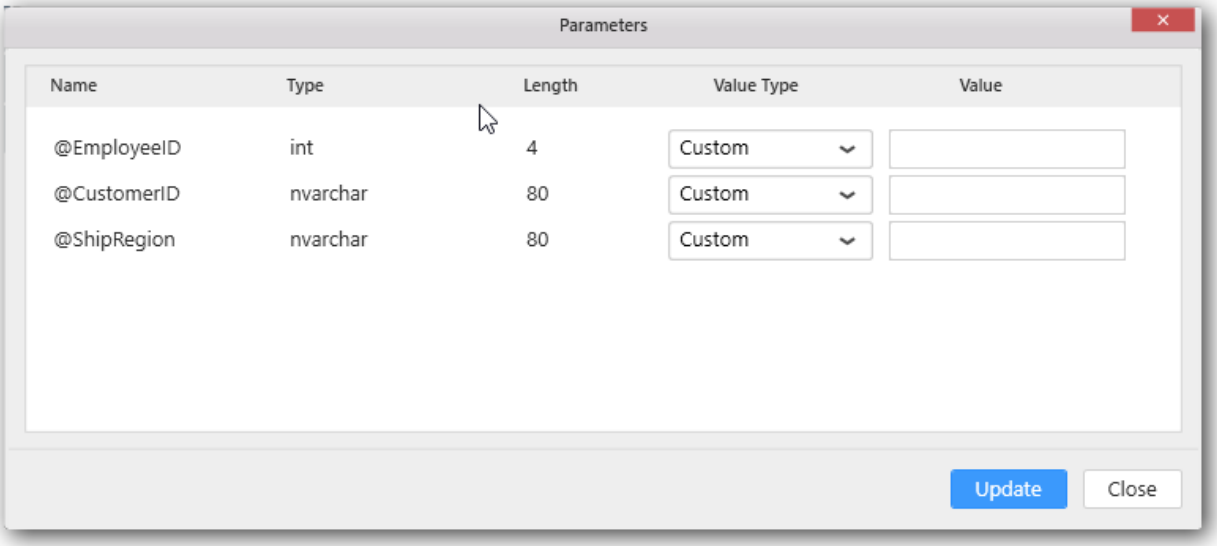
Note: You can connect to a database hosted in Microsoft SQL Server whose version should be 2012 or above.

Connecting to SQL Server Database

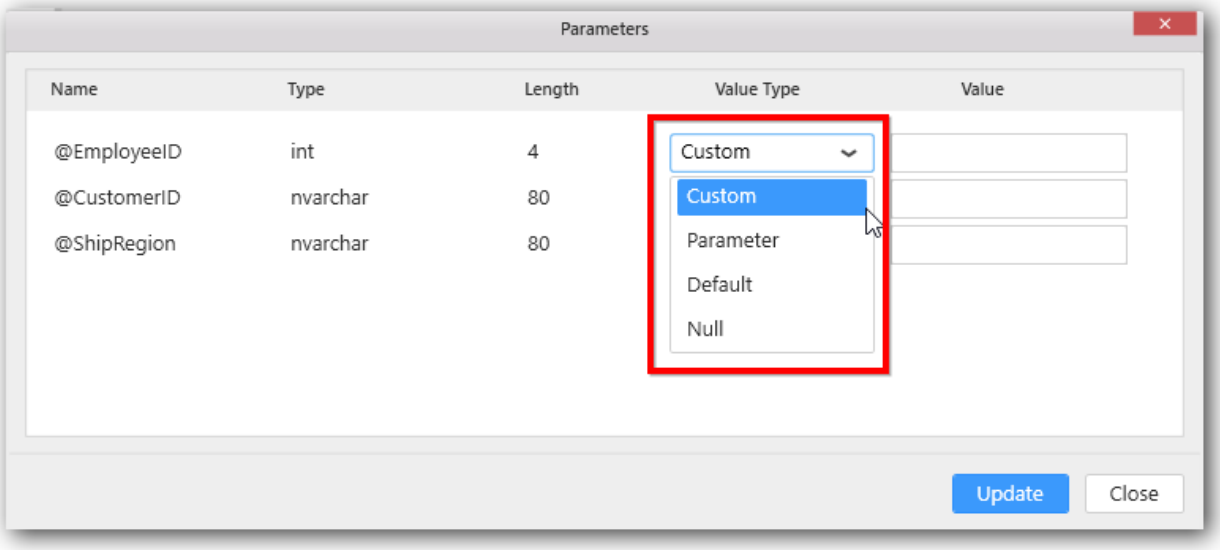
If it succeeds, you will get into the data design view like below with the available stored procedure will be displayed on the left pane of tree view. And the available parameters also can be displayed while expand the procedure.



You can drag the desired stored procedure into the canvas area to create a table view with the supplied parameters to this procedure.

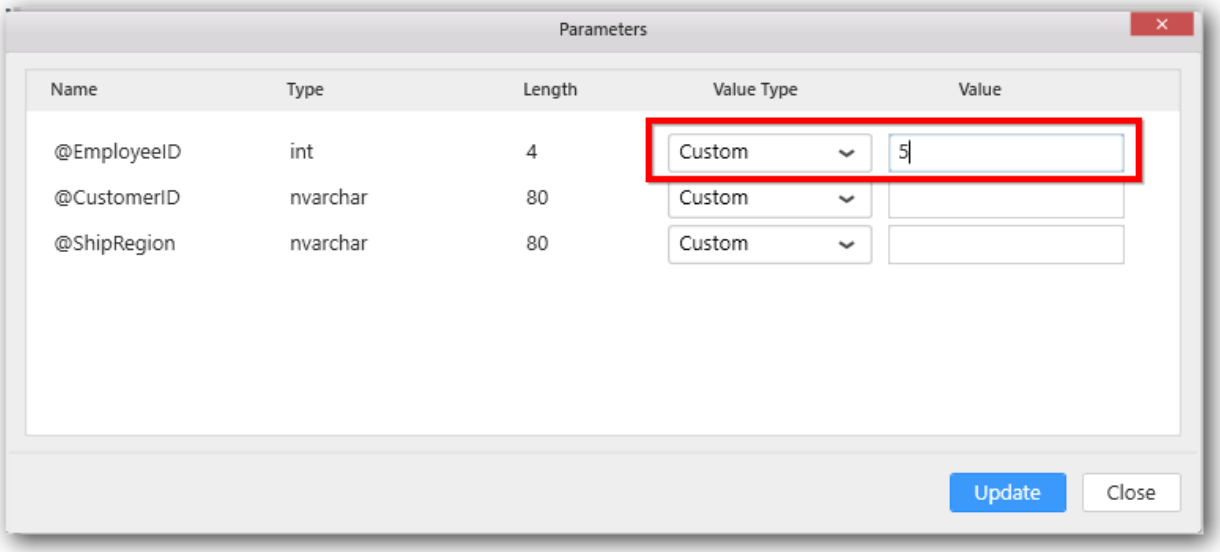


You can pass the values to stored procedure parameters in four ways as shown in the following screenshot.



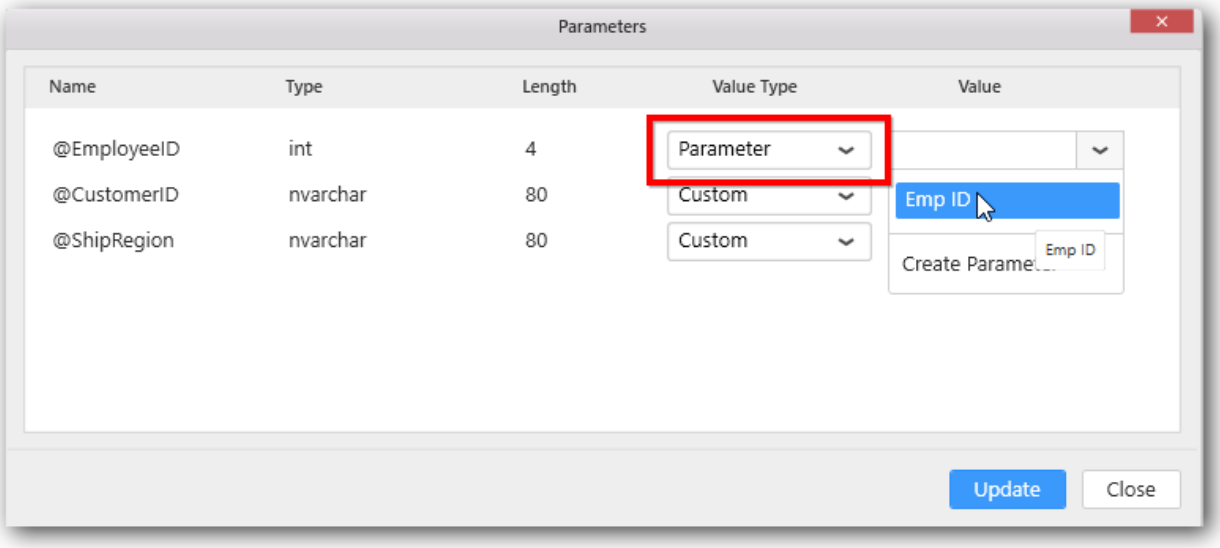
Custom

Through Custom option, you can manually enter the value to pass parameter.



Parameter

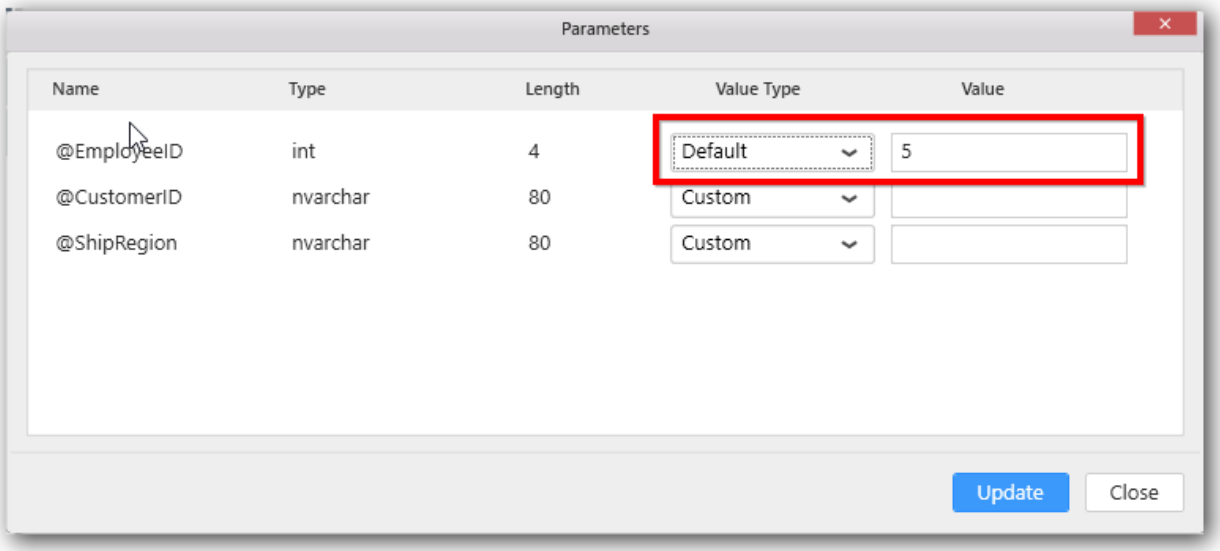
Through Parameter option, you can create a dashboard parameter and use this value to stored procedure parameters.



Default

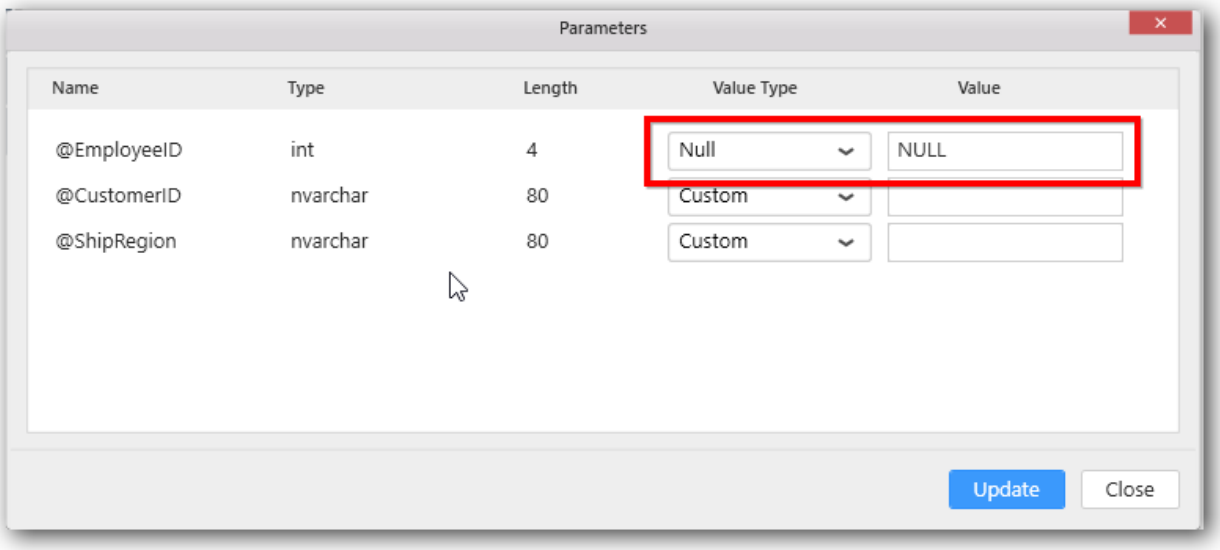
Through Default option, you can pass default value of stored procedure. Corresponding parameter value (default value) will display in the **Value** text box.

Note: If the stored procedure is defined with default value for their parameters, the Default option will display in drop-down box.



Null

You can choose Null option if you want to pass null value to parameter.



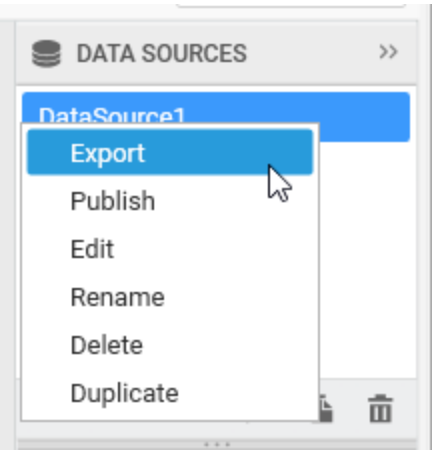
The Syncfusion Dashboard Designer allows to edit the supplied parameters by using the edit parameters button which is available in the created table.

Saving a Data source

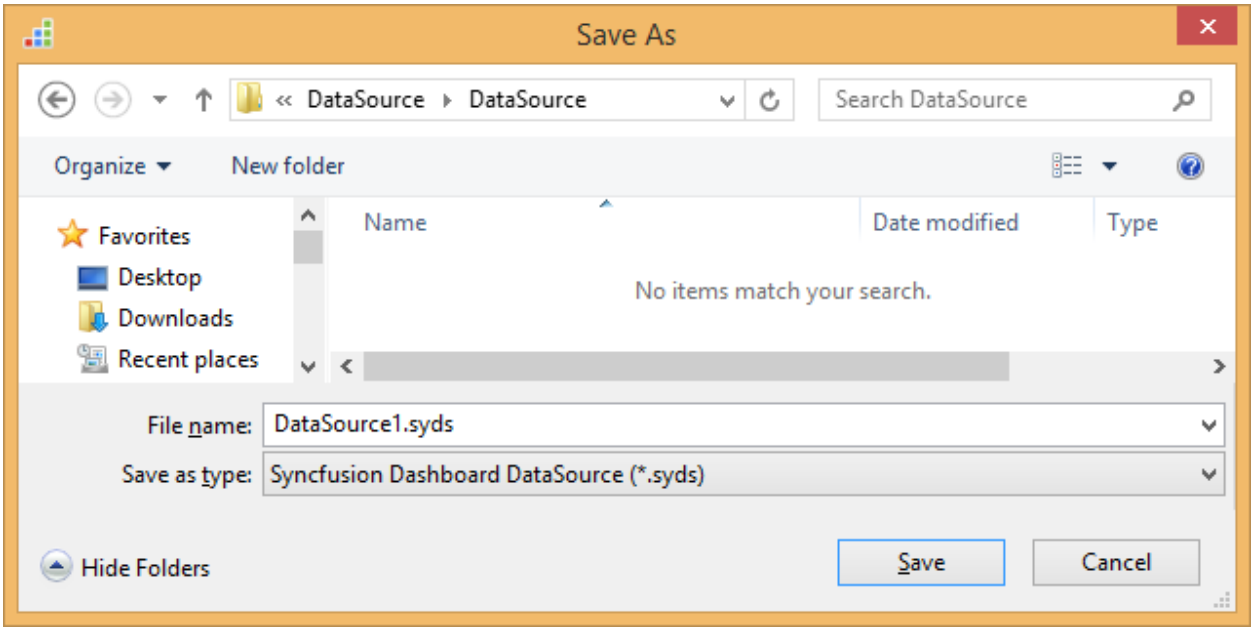
A data source can be saved to any accessible location in local machine in SYDS format. Here is the procedure to save a data source.

Select the respective data source name in the DATA SOURCES container.

Right click the selected item to pop out the context menu and select Export option.



Now, the Save As dialog opens to set the location and file name under which this data source need to be saved.



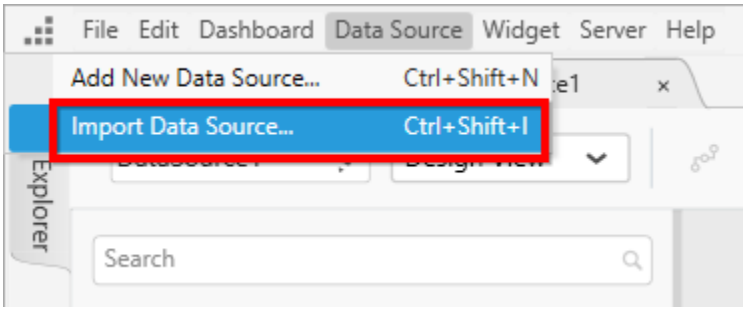
Once it was saved, the saved file will get placed in the mentioned location like below.

Import Data Source

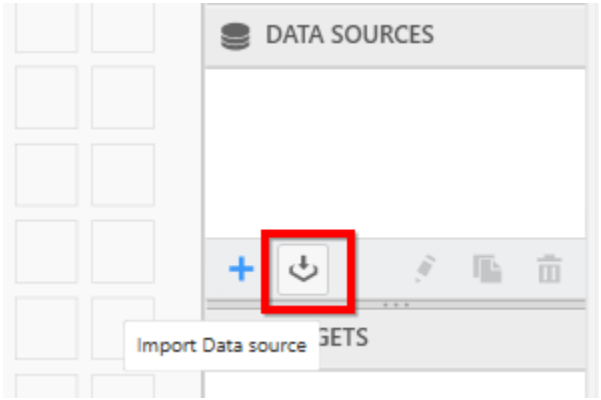
You can use the existing data source in your current dashboard using the Import Data Source option in the Dashboard Designer.

You can open the import Data Source window using anyone of the following ways.

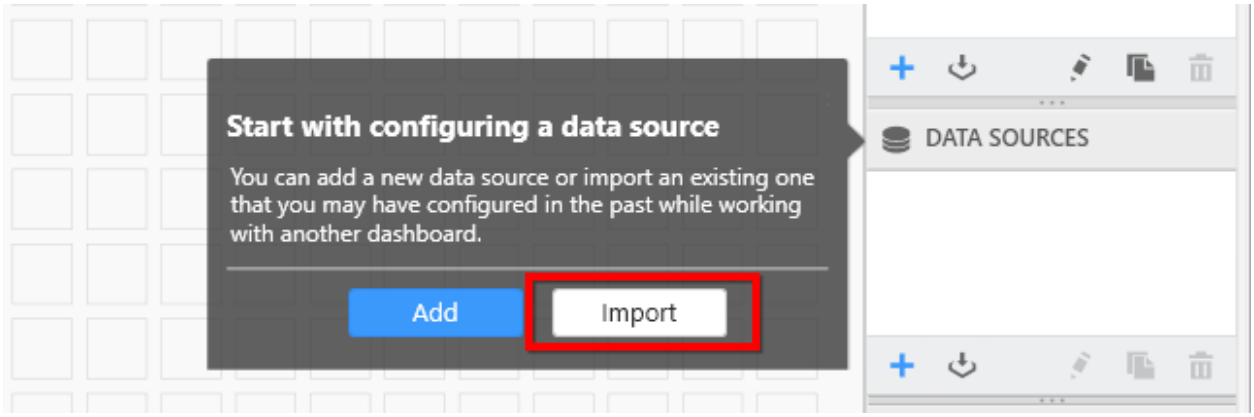
Click the Data Source menu from the top of the Dashboard Designer and select the Import Data Source... item.



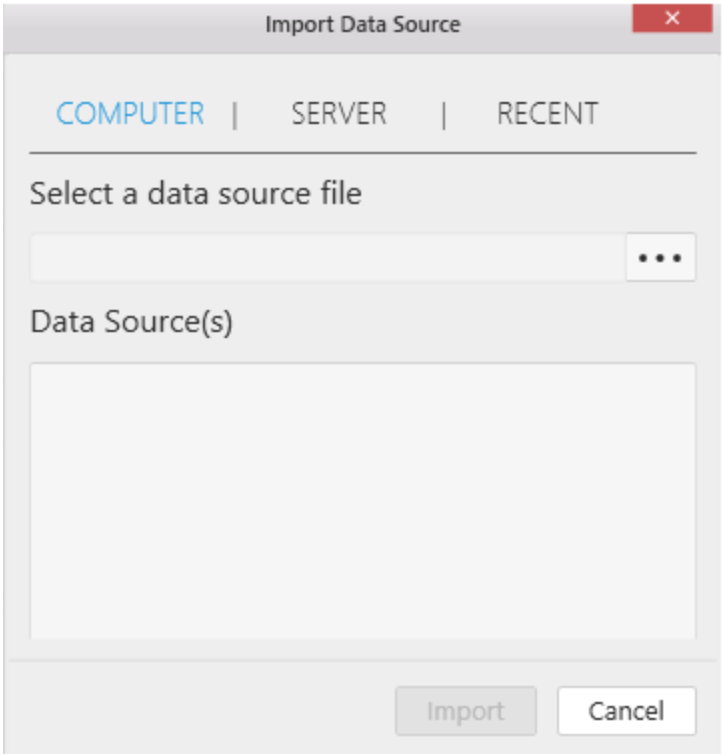
Also you can open the Import Data Source window through keyboard shortcut (Ctrl + Shift + I) or through the Import Data Source icon from the Data Sources container.



When you open the Dashboard Designer for the first time or add a new Dashboard, the following option will show. You can choose Import to open the Import Data Source window.

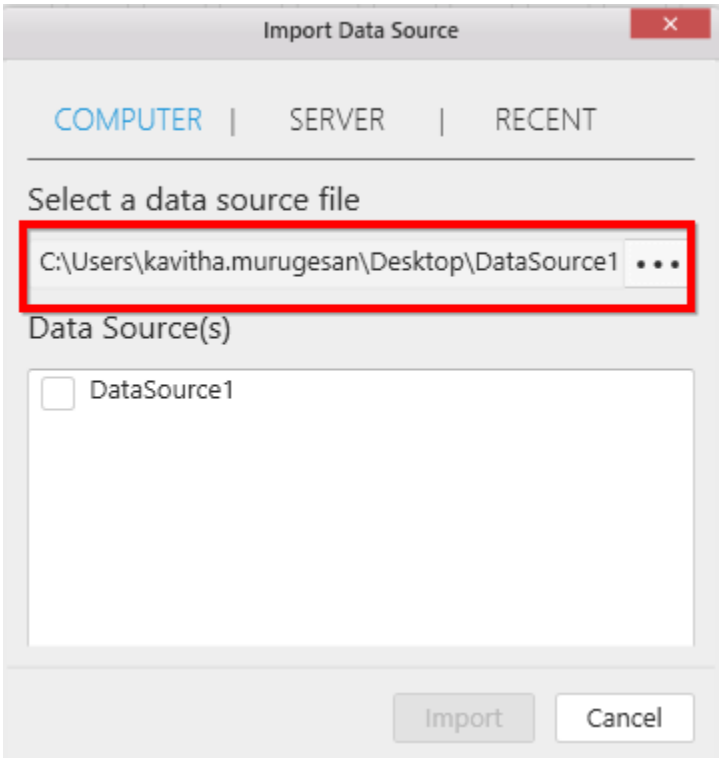


The Import Data Source window opens as follows.

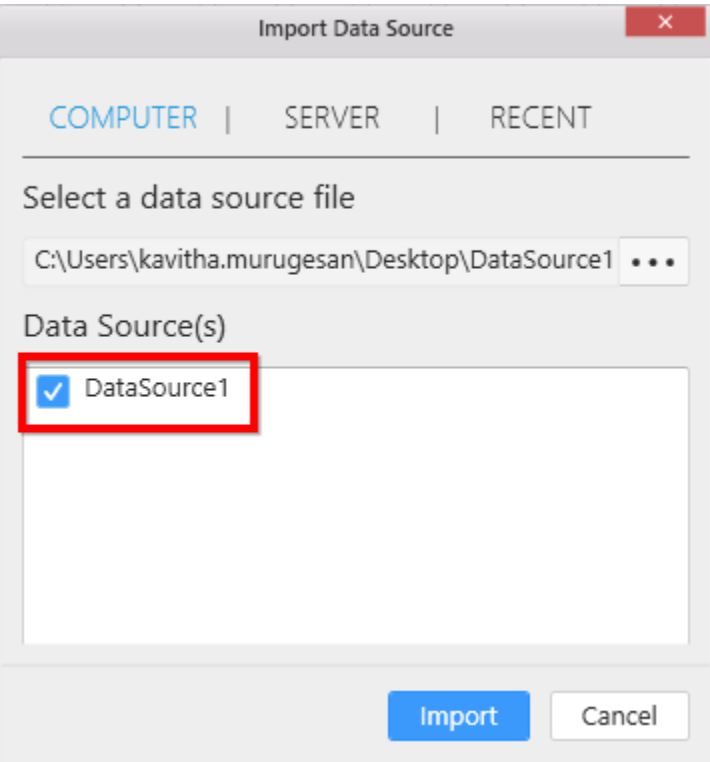


COMPUTER

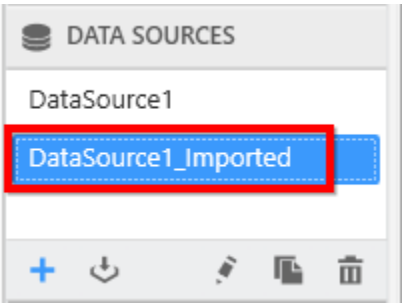
Using this option, you can browse the data source file from the local machine. Here, you can import data sources from SYDS/SYDX/SYDW formatted files. When selecting a file, you can get a list of available data sources displayed as in the following.



Select the required data sources to import into your dashboard and click Import.



Now the respective data source(s) will be imported into the DATA SOURCES container as follows.

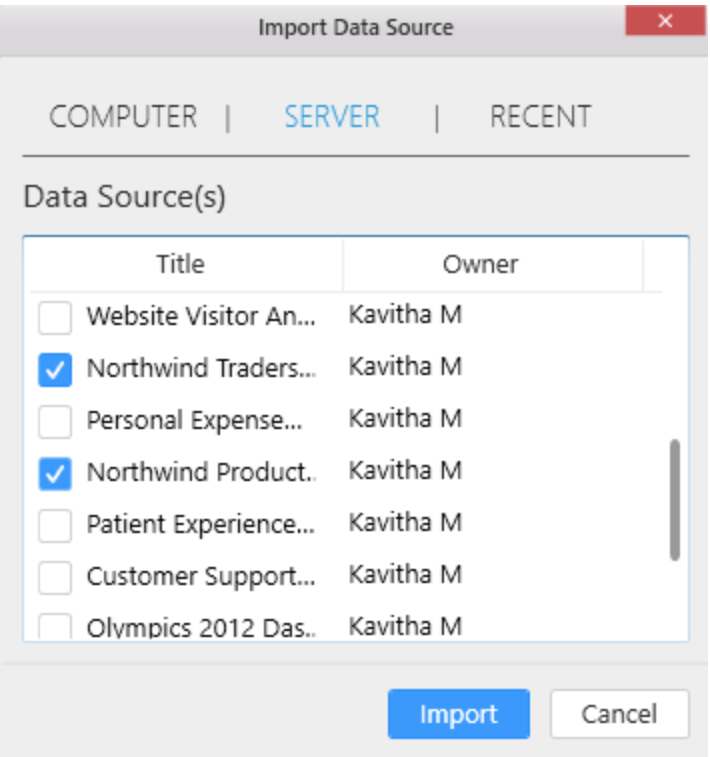


Note: The imported data source will have its original name displayed here if it doesn't match the data source in the current dashboard. If it matches, the datasources in the current dashboard will be renamed on import.

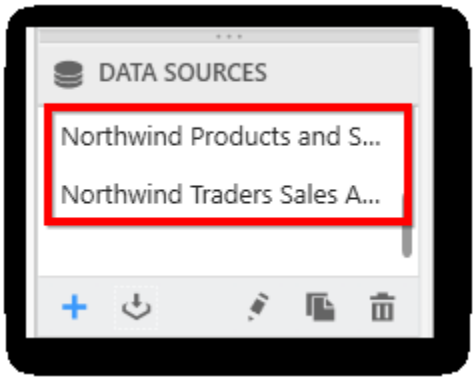
SERVER

You can import data sources from the Dashboard Server using the following ways. To display the data sources that are present in the server, you should sign in the Dashboard Server account in the Dashboard Designer.

Select the SERVER tab. You will get list of available Data Sources that are present in the Dashboard Server. Choose the data sources and click Import.



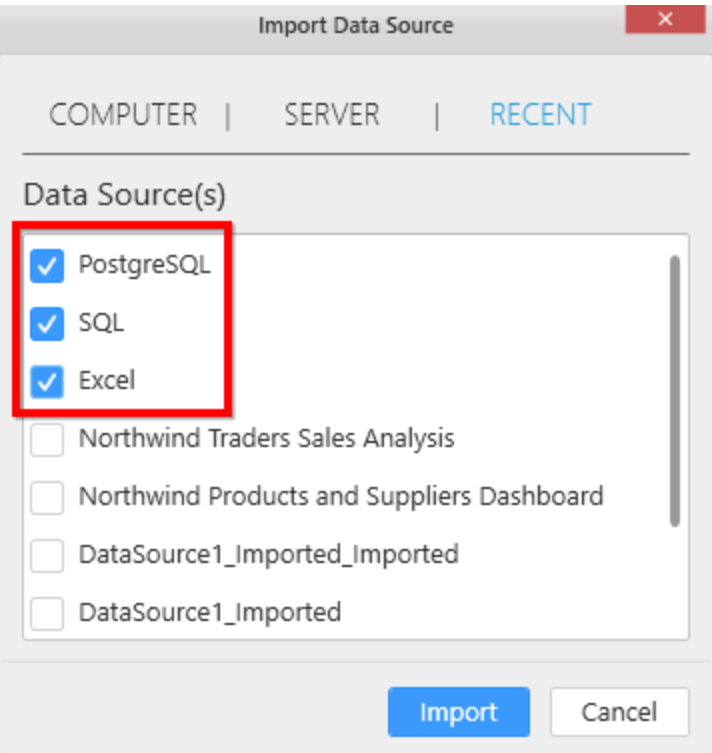
Now the respective data source(s) will be imported into the DATA SOURCES container as in the following.



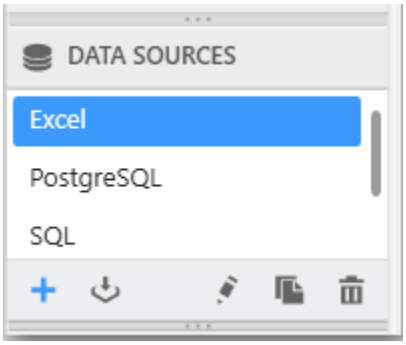
RECENT

You can import recently used data sources in the current dashboard using the following ways.

Select the RECENT tab and choose the data sources you prefer to import and click Import.



Now the respective data source(s) will be imported into the DATA SOURCES container as follows.



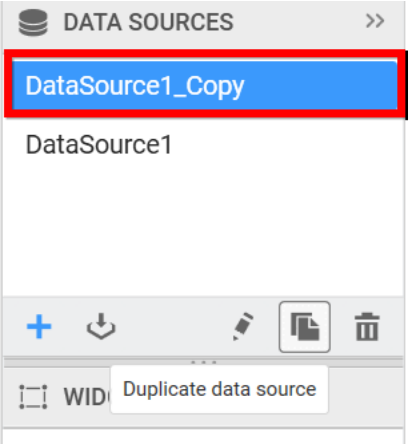
Duplicating a data source

You can create a duplicate copy of an existing data source through the following procedure:

Select the data source that you need to duplicate in the DATA SOURCES container.

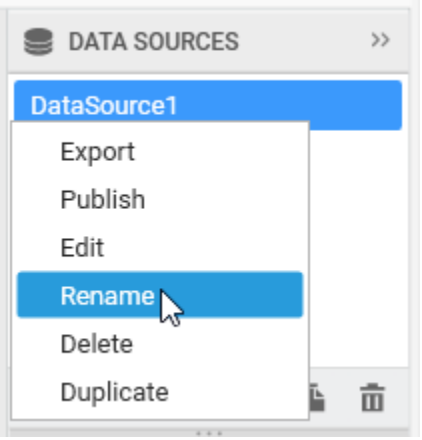
Click the highlighted icon to duplicate the selected data source.

Now a duplicated data source will be created like below.

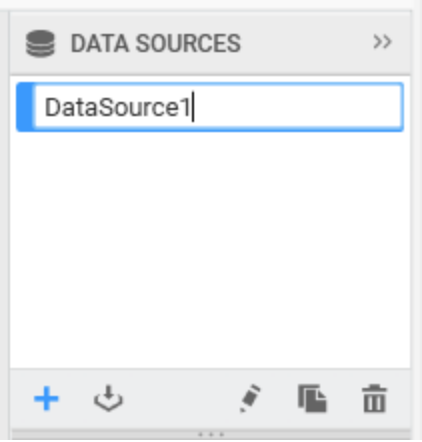


Renaming a data source

You can rename a data source through the Rename option in context menu shown on right click over the respective data source in the **DATA SOURCES** container like below.

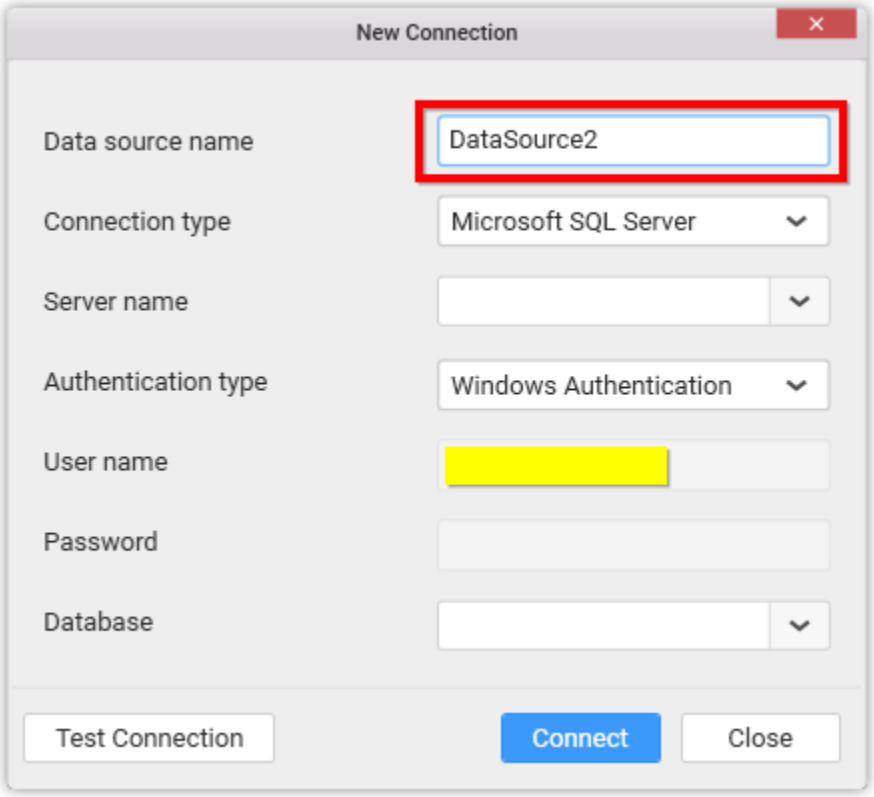


On clicking the Rename option, you will get the selected data source name in edit mode like below to enter the modified name.

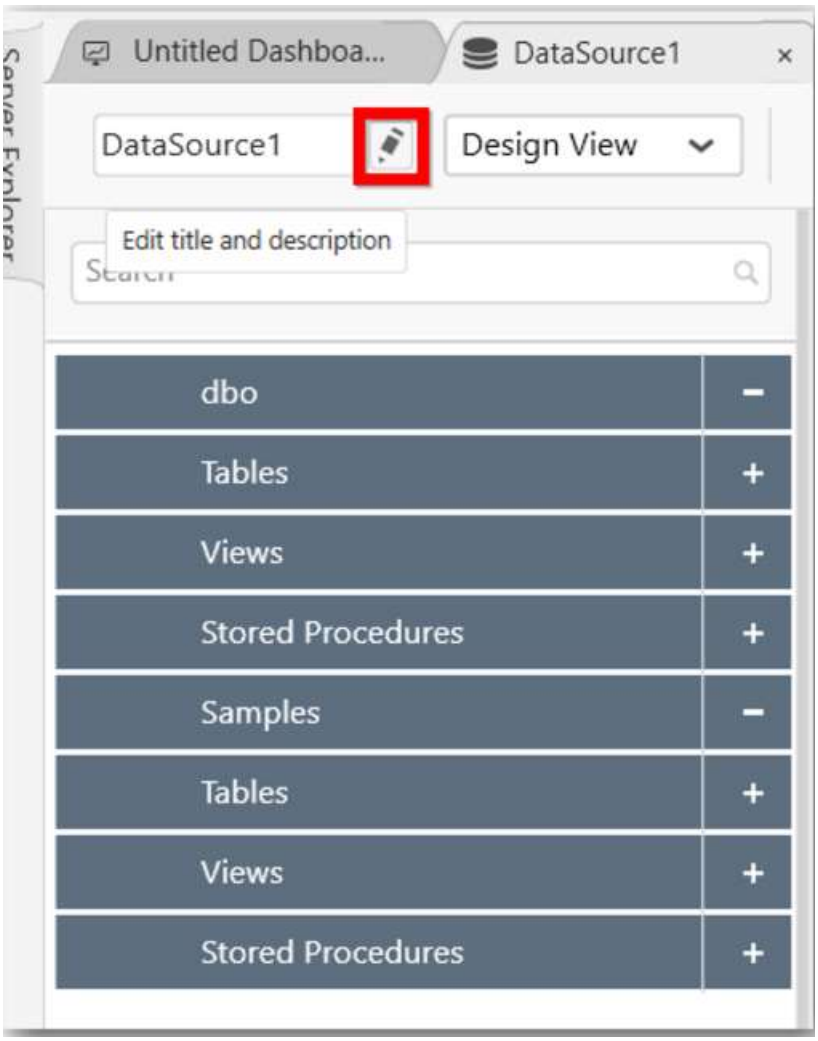


Once modified, clicking out of that edit region will save the pending changes.

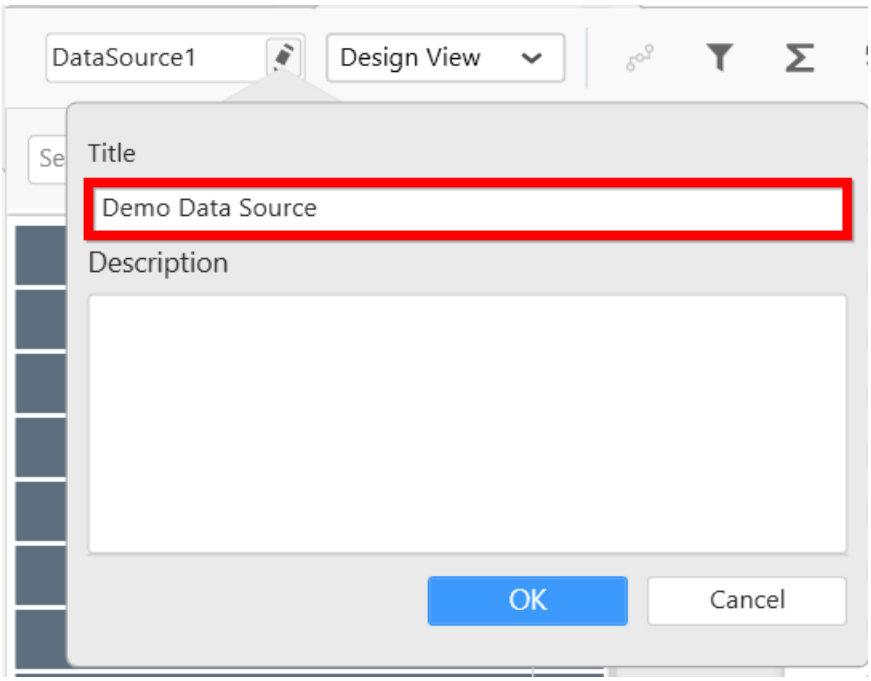
You can also rename a data source while creating it by changing the Data source name field.



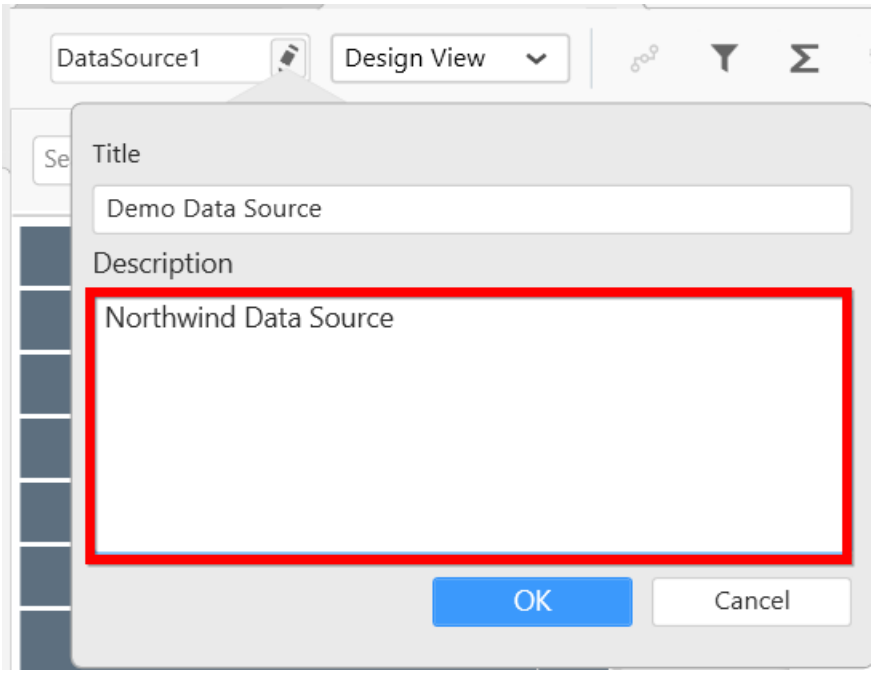
You can also rename a data source from data design view tab by clicking the Edit icon in the data source name text box as shown below.



Rename the data source in the Title text box.



You can also add description for the data source in Description text box.



Click Ok to proceed.

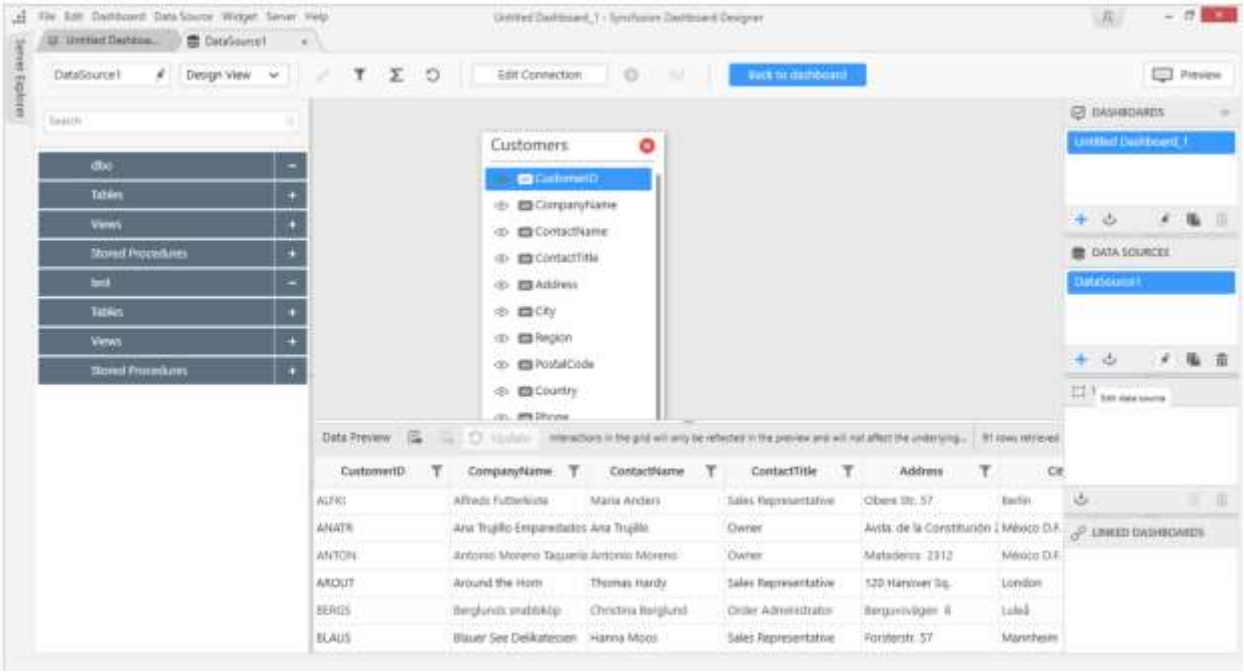
Editing a data source

You can edit a data source through the following procedure:

Select a data source in the DATA SOURCES container that you need to edit.

Click the highlighted icon to edit the selected data source.

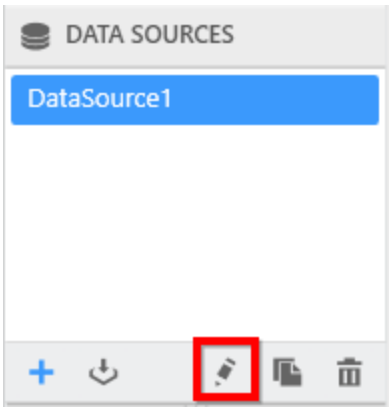
Now the respective data source will get opened in a separate tab to handle your modification.



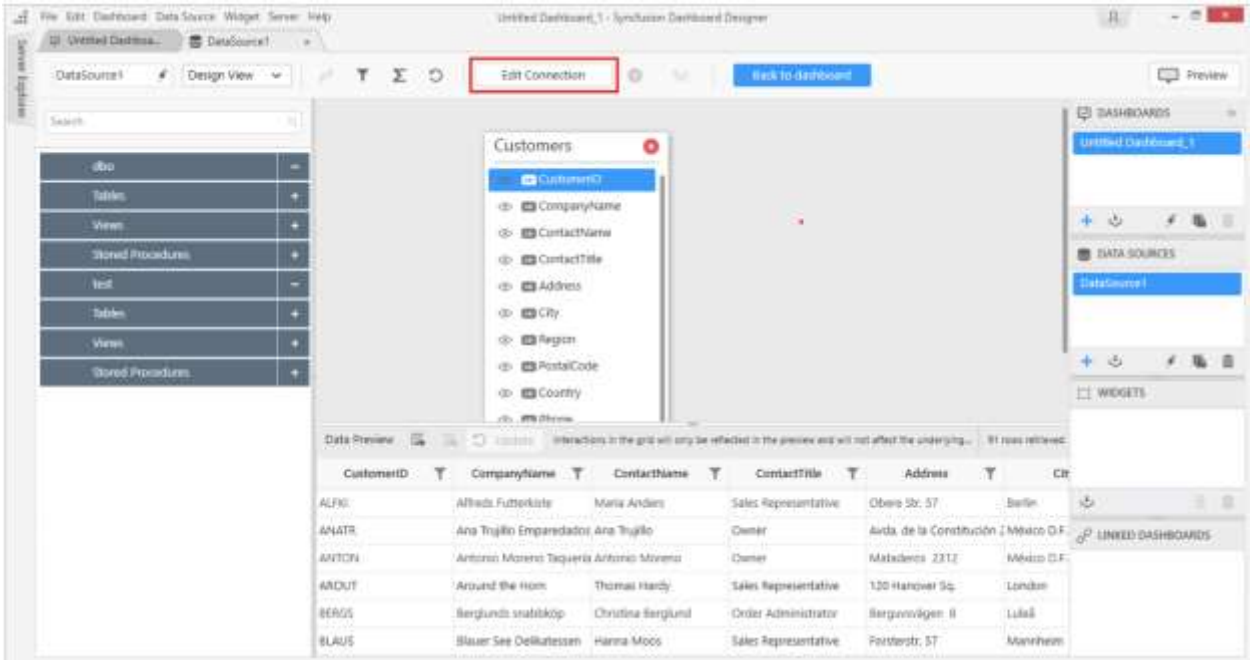
After modification, click Back to dashboard to get back to the dashboard view.

Editing a Data Connection#

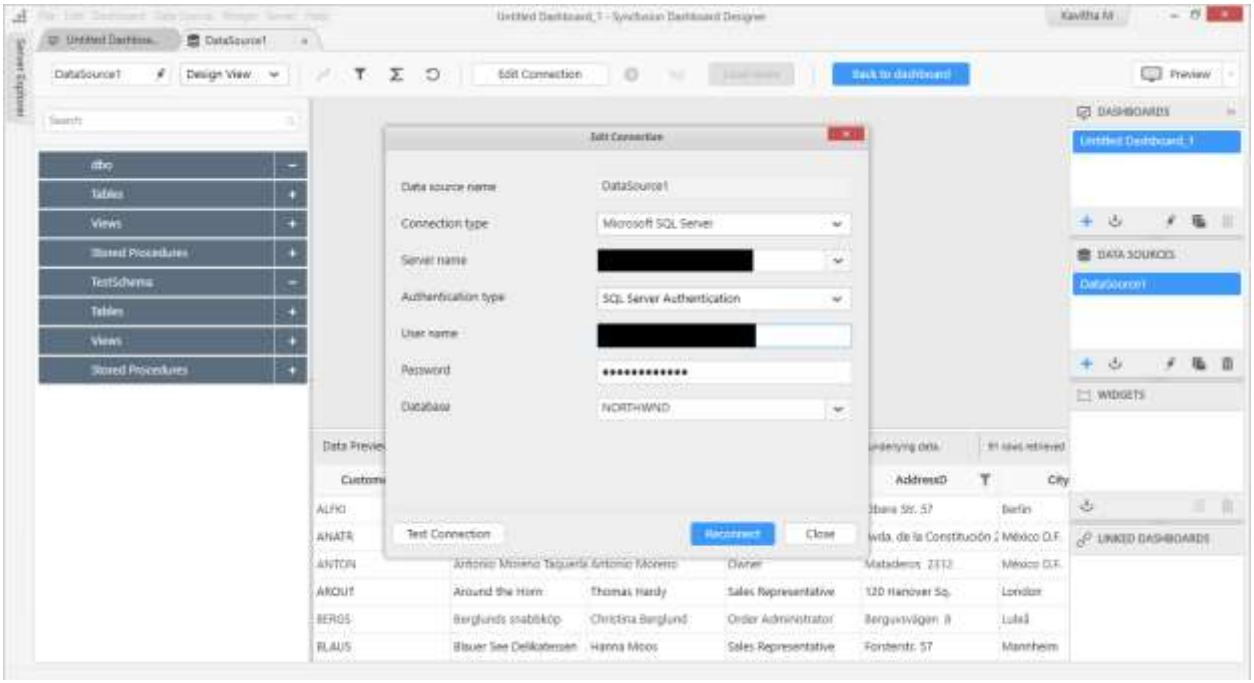
Select the data source associated with the connection that you need to edit in the DATA SOURCES container window.



To open the data source window in separate tab, either click the Edit icon (highlighted above) at bottom of this container, or right click the selected data source and select the Edit option in the context menu displayed.



Click the Edit Connection… button in the data design window toolbar. Now, Edit Connection dialog opens.



Make the preferred changes and click Reconnect button.

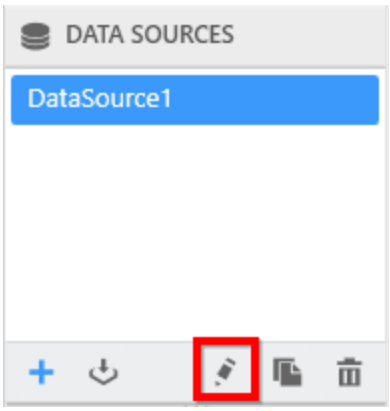
Information: Reconnection will persist the dropped table(s), table relationships, data filters, data configuration to widgets, unless the schema is different from the previous data connection. i.e., the reconnected database should have similar schema like the previously connected database, which may exist in same or different location. If the reconnected one don't have a column that is available in previous one, reconnection will just ignore that column and its related settings alone and persist others. Beyond that level, reconnection will drop previous settings entirely.

Note: Cross-data source filter configuration handled in Dashboard Filters Configuration window will not be maintained on reconnection even the schema is similar.

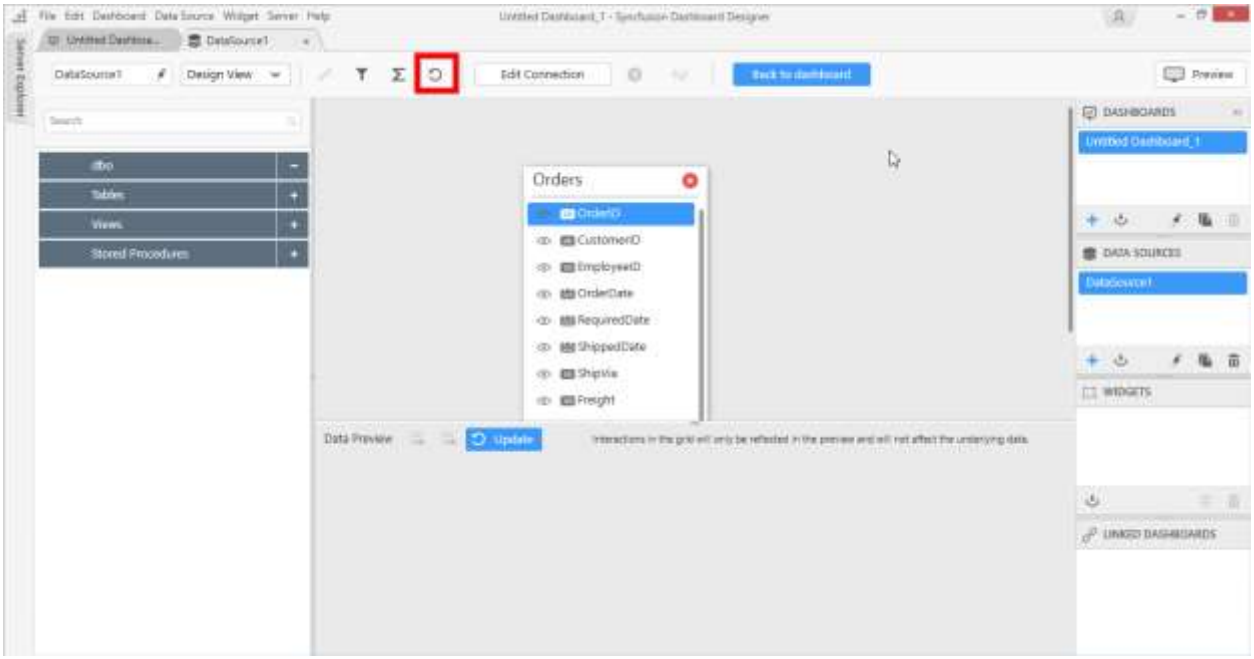
Refreshing a Data Connection

Select the data source associated with the connection that you need to edit in the DATA SOURCES container window.

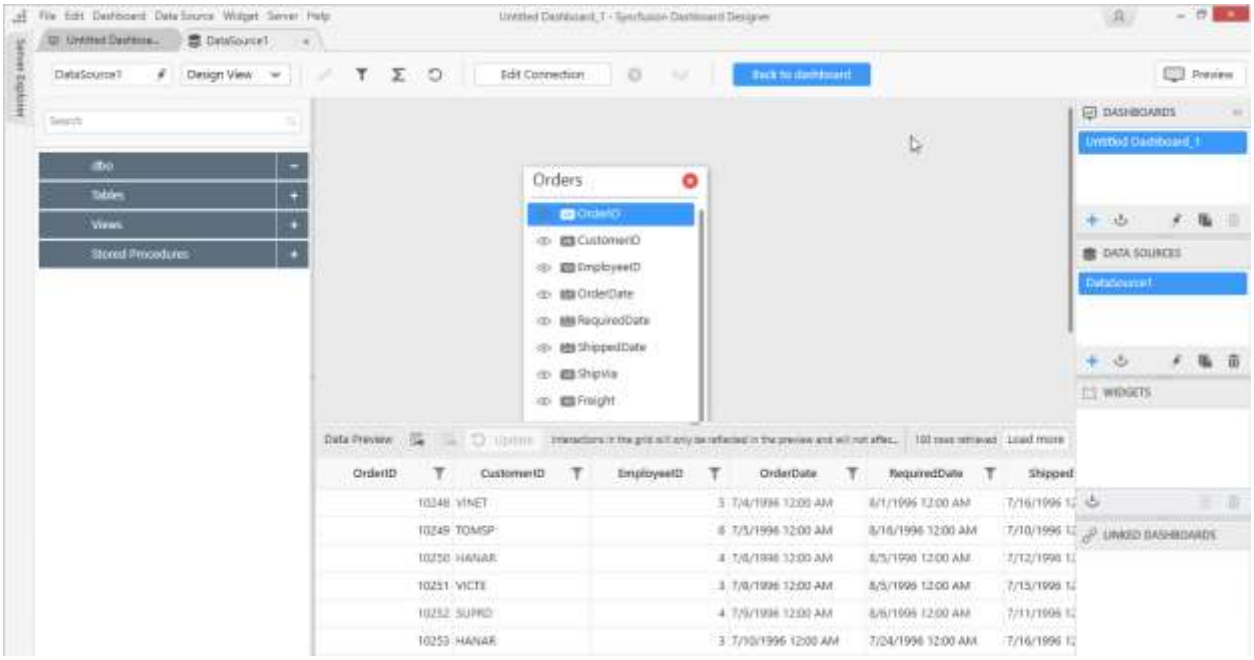
To open the data source window in separate tab, either click the Edit icon (highlighted below) at bottom of data source container, or right click the selected data source and select the Edit option in the context menu displayed.



Click the Refresh icon in the data design window toolbar. Now data connection gets refresh.



By clicking on Update button the data will be updated in preview grid for the configured tables. Configured widgets also get refreshed with new data.

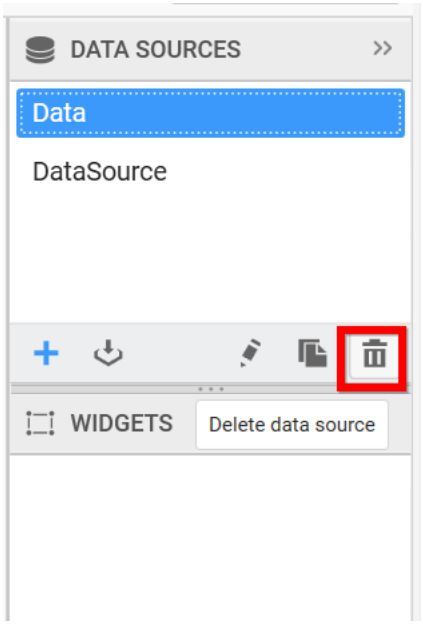


Deleting a data source

You can delete an existing data source through the following procedure:

Select a data source in the DATA SOURCES container that you need to delete from dashboard.

Click the highlighted icon to delete the selected data source.



Now the selected data source will get removed from the DATA SOURCES container.

After deleting a data source, widget(s) will get reset which are configured using deleted datasource.

Connecting through Custom SQL Query

You can also create the data source through custom SQL query. This feature allows you to define the data source with the manually written queries instead of manual drag and drop of tables for data

connection types like Microsoft SQL Server, PostgreSQL, ODBC Connections (SQL, MySQL and Oracle) and ODBC ANSI SQL.

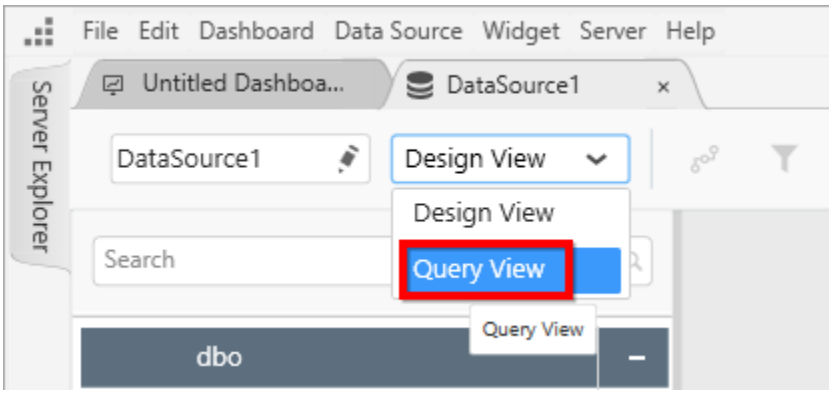
[Establish Data Connection](https://help.syncfusion.com/dashboard-platform/dashboard-designer/connecting-to-data/connecting-to-data)

In New Connection wizard, set the respective Connection type and configure other details based on the chosen connection.

Click Connect to establish the connection to data source.

Switch to Code View

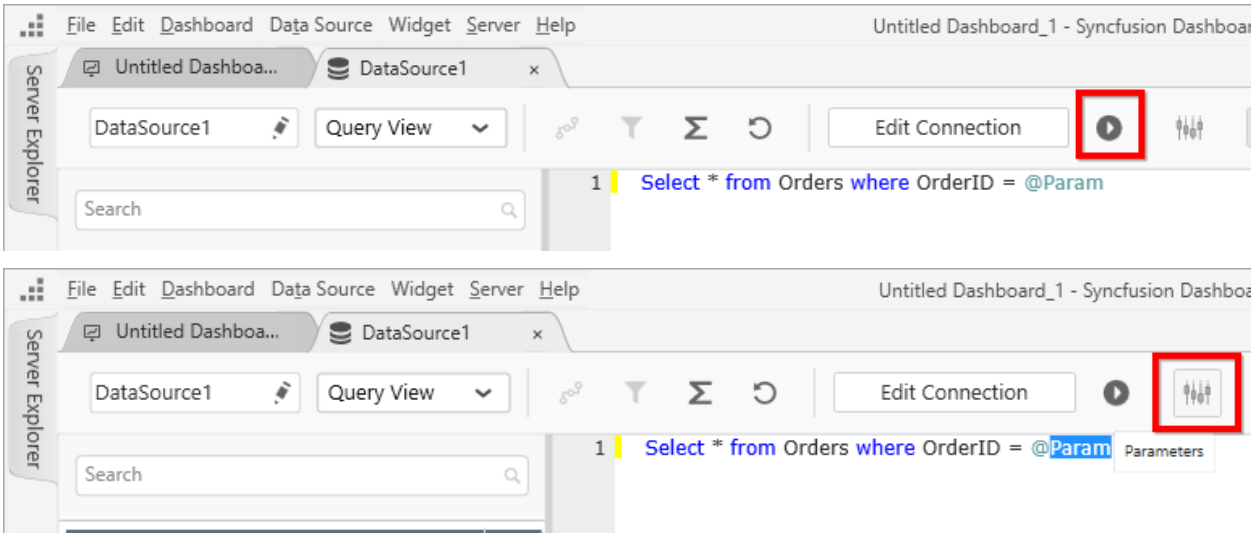
Be default, data source tab will get opened with design view. Switch to the code view through the drop down menu in the tool bar to access the Query editor pane.



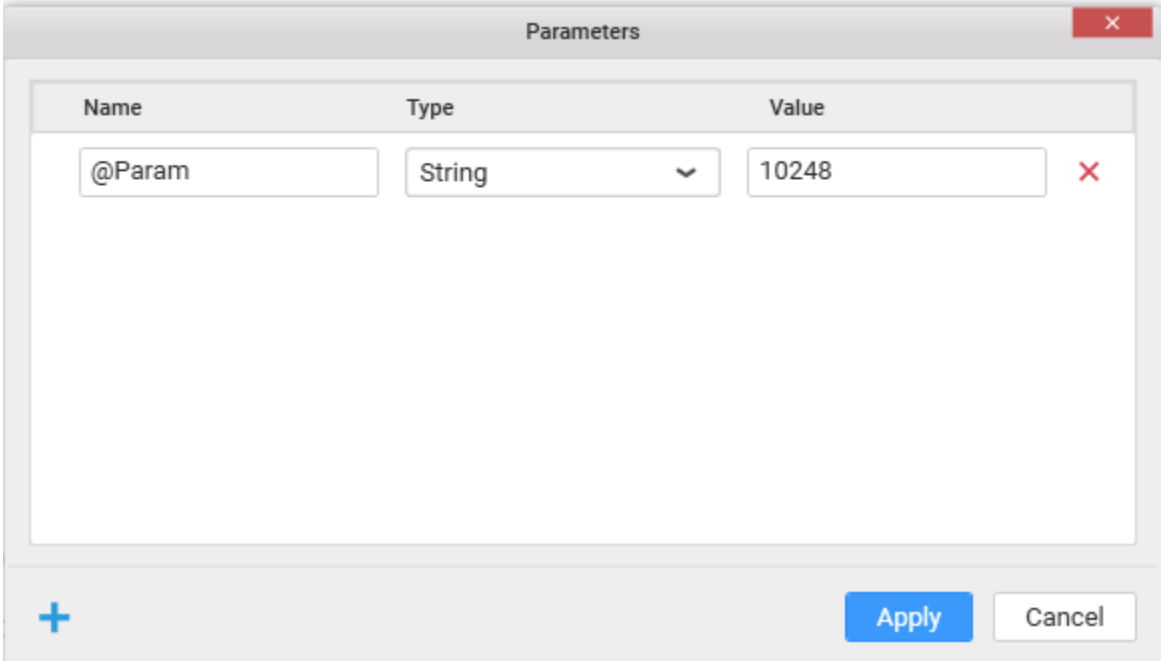
Write Query and Execute

Write your custom SQL query in the Query editor and click Execute button in the toolbar to process that query. This will get you the table schema information and create a new data source which can be bounded to dashboard widgets. The data preview can be seen through the preview grid at bottom pane.

Query editor supports to query with parameters too. Click Parameters button in toolbar to launch Query Parameters window and add one or more parameters that you require to define.

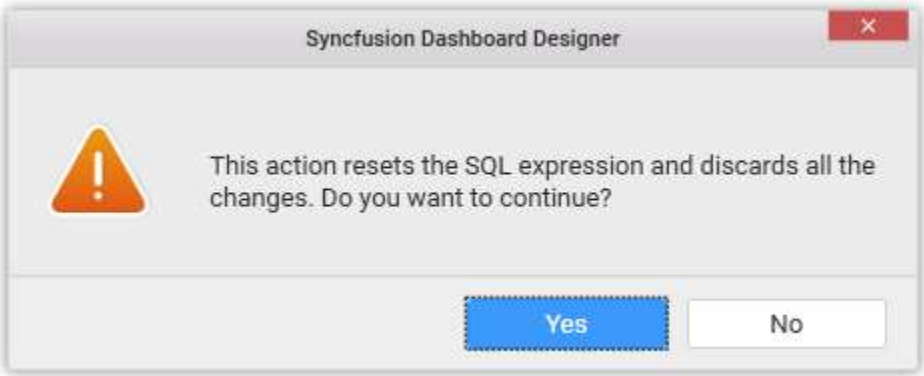


You may also edit the added parameters through the same window.



Click Apply to save the parameter changes and click Execute to generate data based on the defined parameters.

If you tried to switch back to design view, you will be prompted with an alert message for confirmation as proceeding with this action, will reset the code view expression.



Note: There are some limitations on query usage in query view. You can use **Select** and **Exec** statements only in query editor. It does not supports other statements like, Create, update, Alter, Insert Into etc.,

Classification of data sources queried directly and in-memory

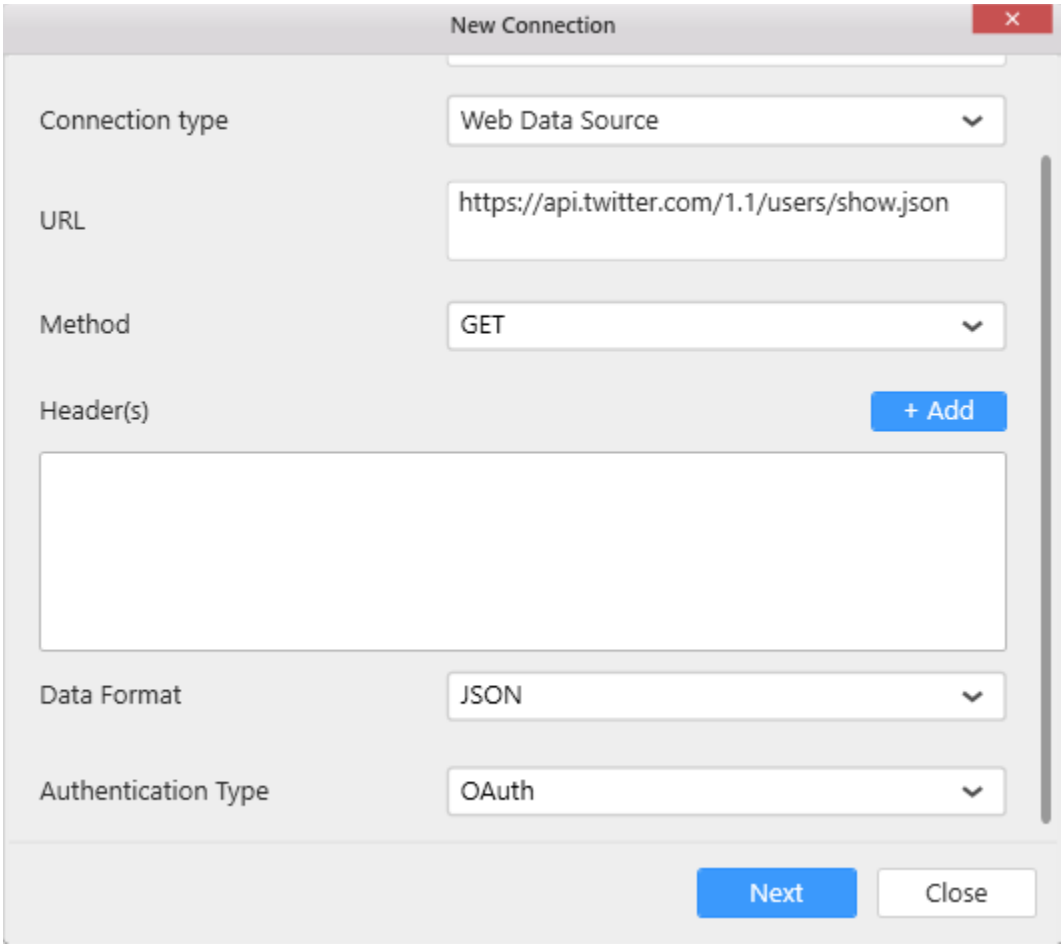
Our Dashboard application supports more than 15 data sources which includes both files type and server type connections. Out of these, server type connections issue query directly to the database for results, on each and every operation done in the dashboard. Whereas for file type connections, querying is not applicable directly, so data are moved to in-memory initially and further processing are carried out there.

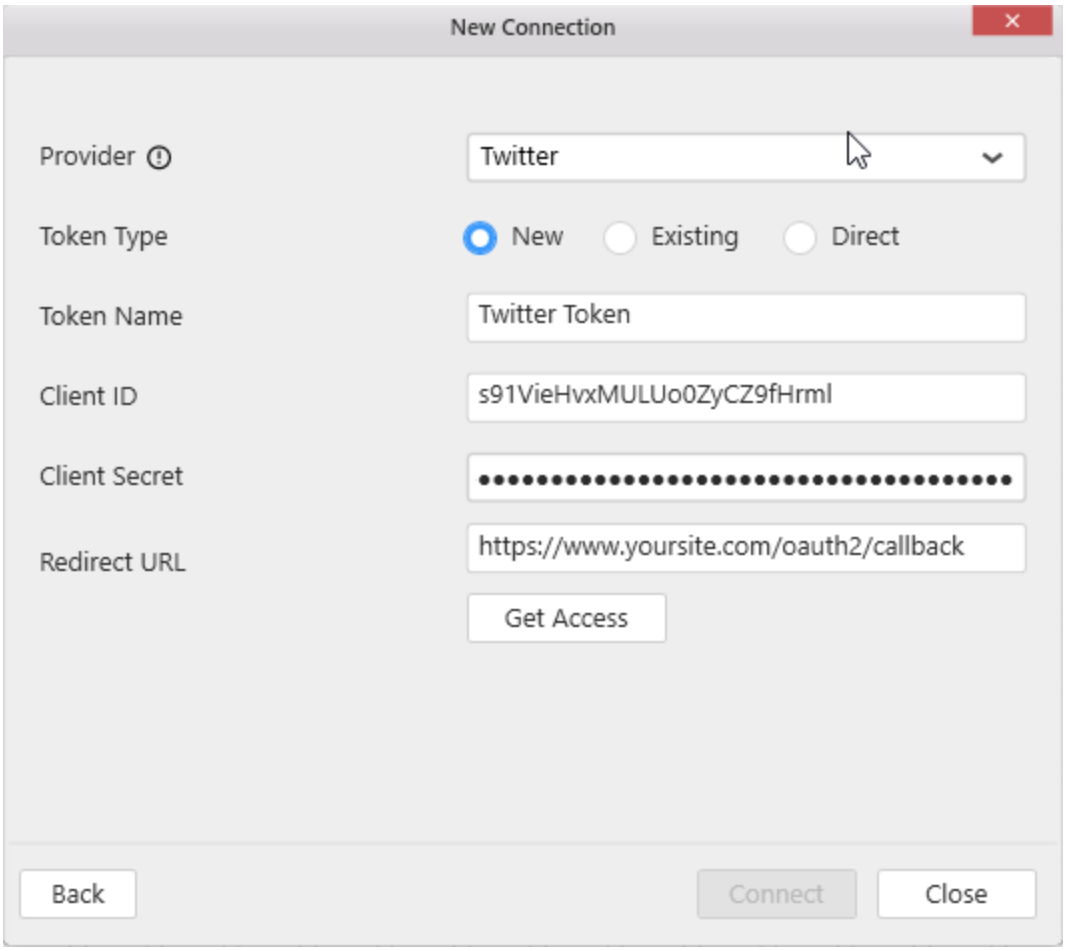
List of data sources whose data are moved to in-memory for processing are - Microsoft Excel, CSV, JSON, Text Document, Microsoft Access, Microsoft Azure Table Storage, Salesforce, ODBC ANSI SQL and Web Data Source.

List of data sources which are directly communicated are - Microsoft SQL Server, Microsoft SQL Server Analysis Services, PostgreSQL, SQLite, MySQL, Oracle, ODBC ANSI SQL, Spark SQL and Hive.

Connecting data through OAuth

You can connect to the web services of popular sites like, Twitter, Facebook, LinkedIn, Google, Yahoo, Instagram, Dropbox, GitHub and your own or any other OAuth services using OAuth through Web Data Source connection type. To enable this connection, you need to register Dashboard Designer application to the respective service provider with your own account details. During the registration process, you will get the Client ID, Client Secret and Redirect URL utilized for data fetch.





Fill the required details in "New Connection" window and click "Get Access" button to provide login information to access your account.

Click "Connect" button to get the data from the corresponding providers and now you will be redirected to data design view window.

Note: The Connect button is enabled only after getting the access successfully.

How to get Client ID, Client Secret and Redirect URL

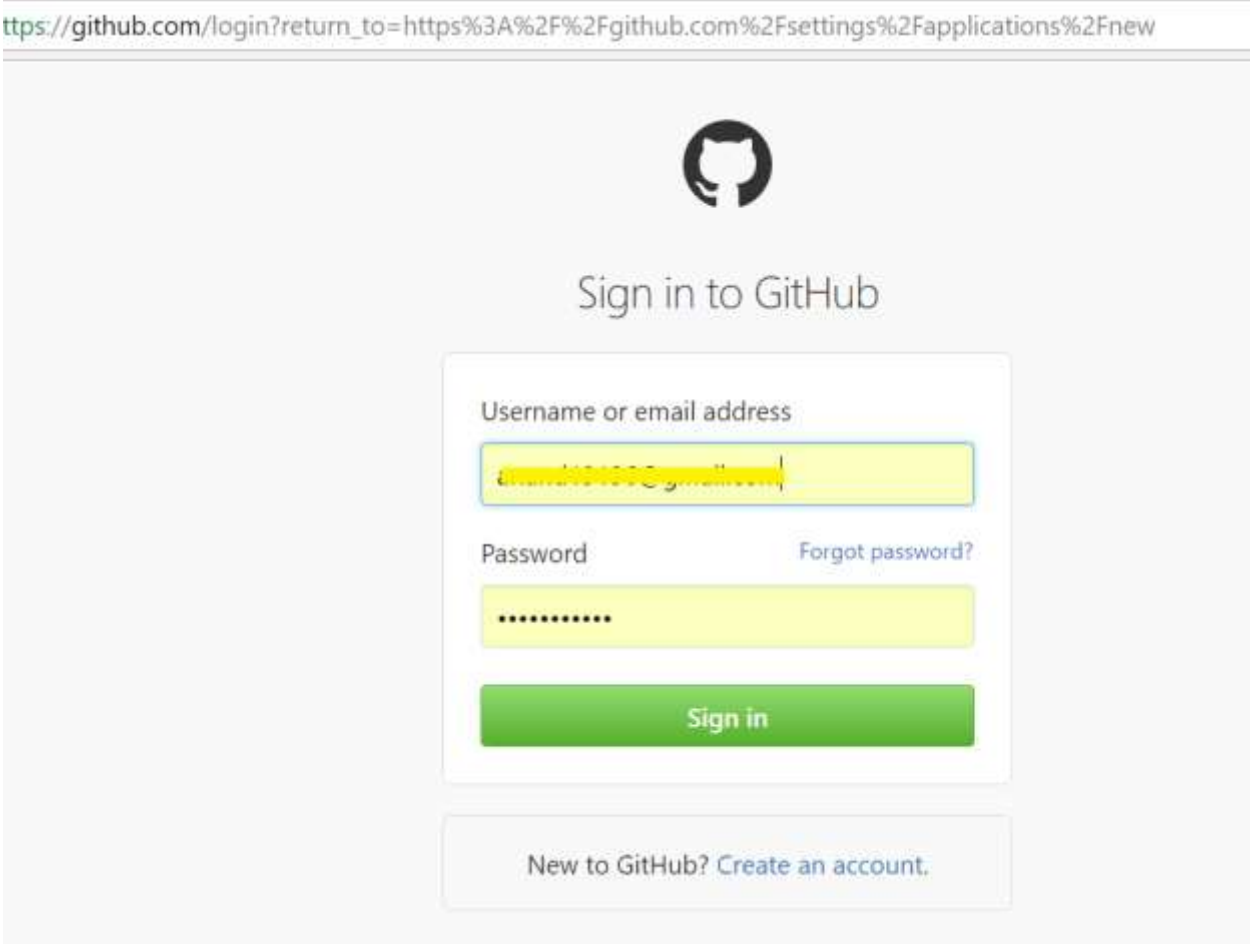
1. Login with your selection provider's development site. 2. Create new application 3. Register your new application with fill the all mandatory fields they required to register the application. 4. The Client ID and Client Secret was generated once the registration was completed.

For Example, to get Client ID, Client Secret and Redirect URL for GitHub data access

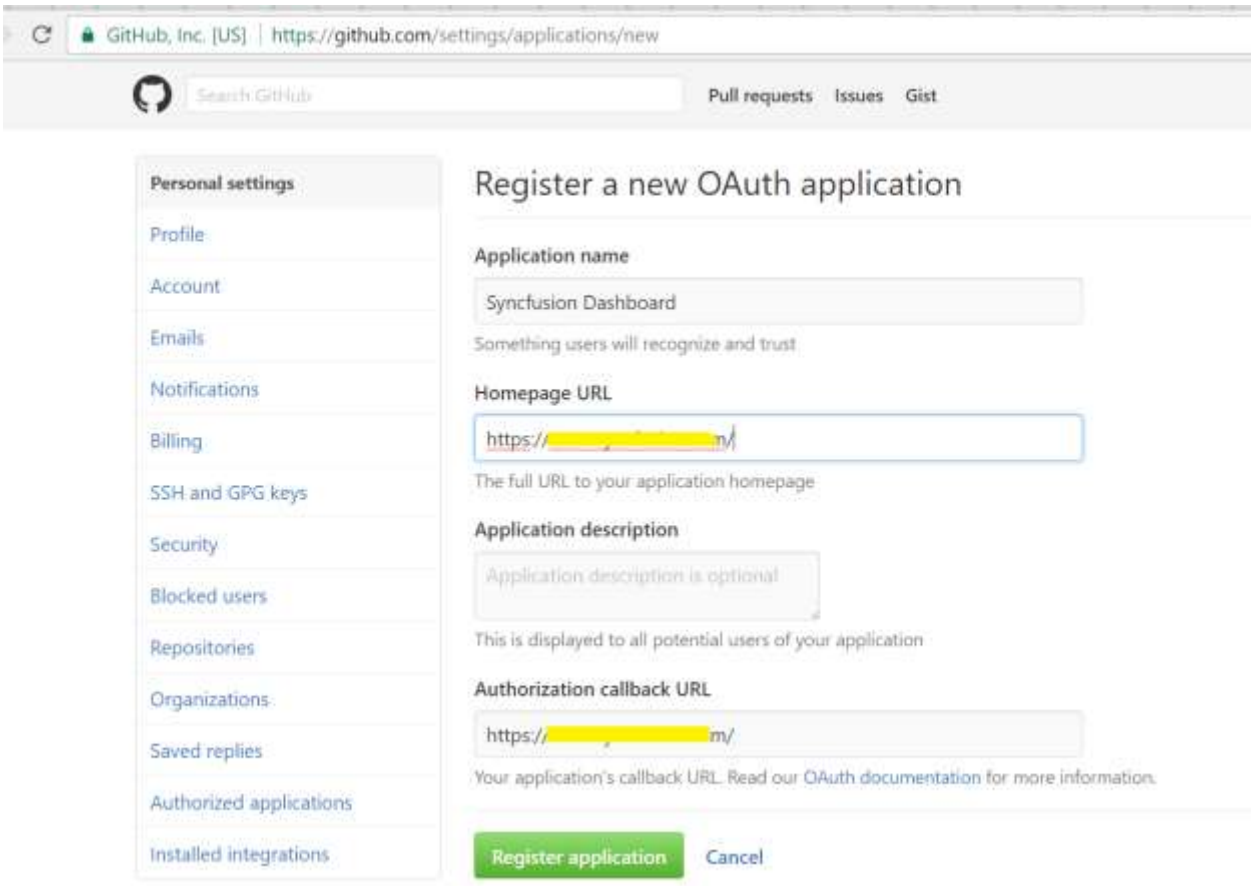
1. Click here to register your application in GitHub development site.

https://github.com/settings/applications/new

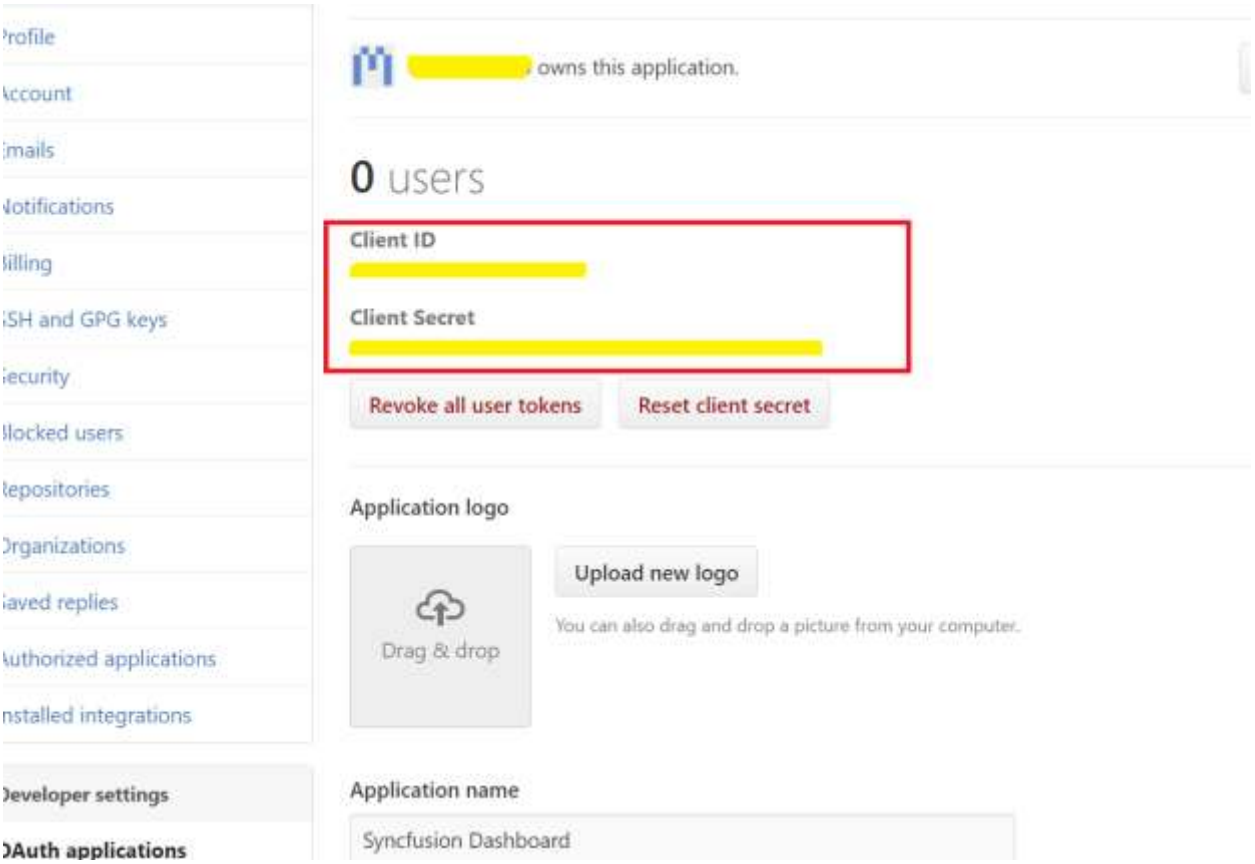
2. Login with your GitHub credential.



3. Fill all the required fields in the registration page.



4. Click Register application. Once the registration completes, you will get Client ID and Client Secret.



Note: Here, the Callback URL is nothing but the Redirect URL. Redirect URL (or Callback URL) holds the token information which received from the respective Web Data Source. So you need to give a valid Redirect URL (or Callback URL) for registration.

Links for registering your application to respective OAuth service providers

Twitter : https://apps.twitter.com/app/new

Facebook : https://developers.facebook.com/apps/

LinkedIn : https://www.linkedin.com/developer/apps/new

Google : https://console.developers.google.com

Yahoo : https://developer.yahoo.com/apps/create/

Dropbox : https://www.dropbox.com/developers/apps/create

GitHub : https://github.com/settings/applications/new

Instagram : https://www.instagram.com/developer/clients/register/

Common ODBC connector based on ANSI SQL

We have provided support to connect to different databases through their ODBC drivers. More specifically, database solutions providing ODBC drivers based on ANSI SQL standard can establish connection with our application.

When it comes to ANSI SQL supportive ODBC, not all functions are supported by vendors but still most of the functions and keywords used would be common and based on the standards. As long as the query

that we issue from our Syncfusion Dashboard gets executed in the respective database, we may stay in the live query mode. If any function or keyword is not supported, we would recommend the user to switch to extracted mode and continue the dashboard design as usual further. This is a generic way to approach this scenario.

Live Query Mode : Issuing a query directly to the database via ODBC and retrieving the data as a resultant. This happens when the functions and keywords in the query are accepted by the database via its ODBC driver.

Extracted Mode : When the functions or keywords in a query is not accepted by the database via its ODBC driver, user is provided with an option to extract and move their data to an embedded (internal) data source. Further, we continue issuing queries to the embedded (internal) data source and retrieve the data as a resultant. Following are few of the options which does not have direct or alternate functions under ANSI SQL standards and they remain as a limitation with our Syncfusion Dashboard. But if you still prefer to work on them, then we recommend you switching to the extracted mode.

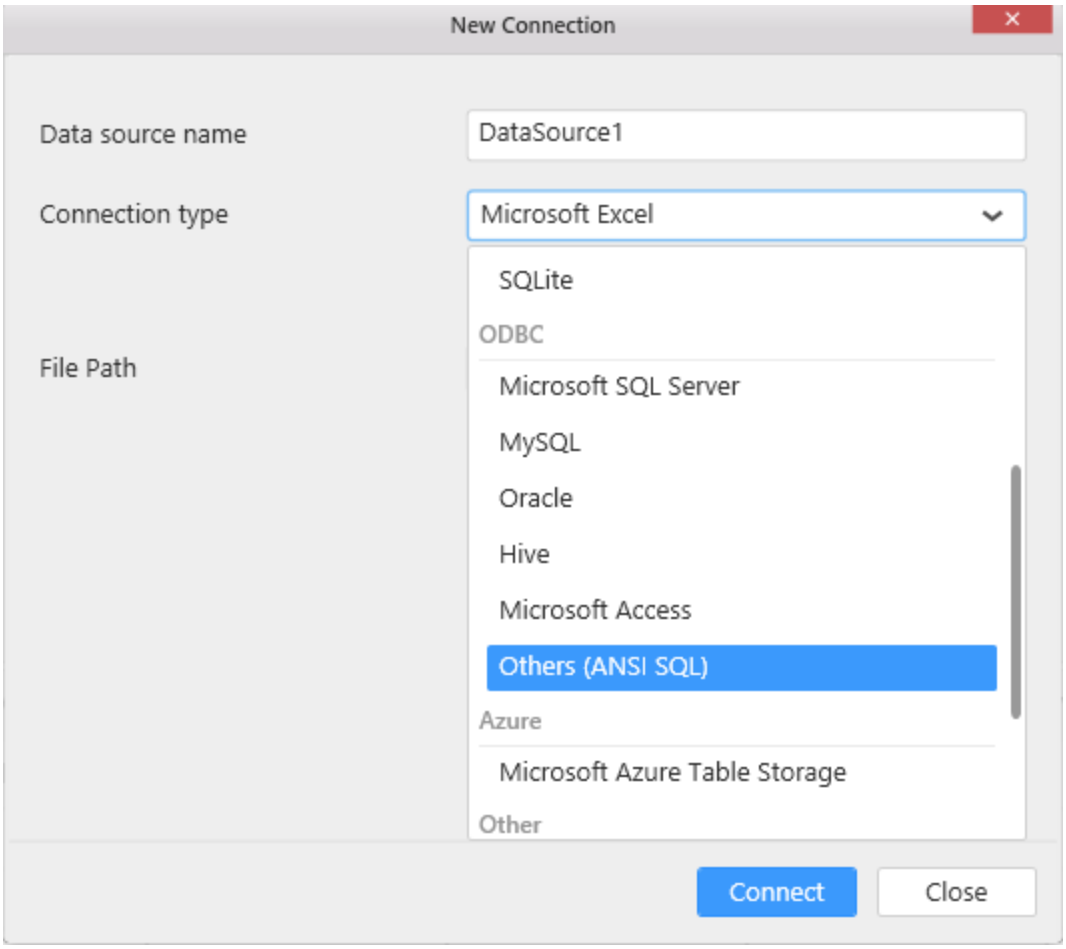
      •   Standard Deviation and Variance under Aggregation functions

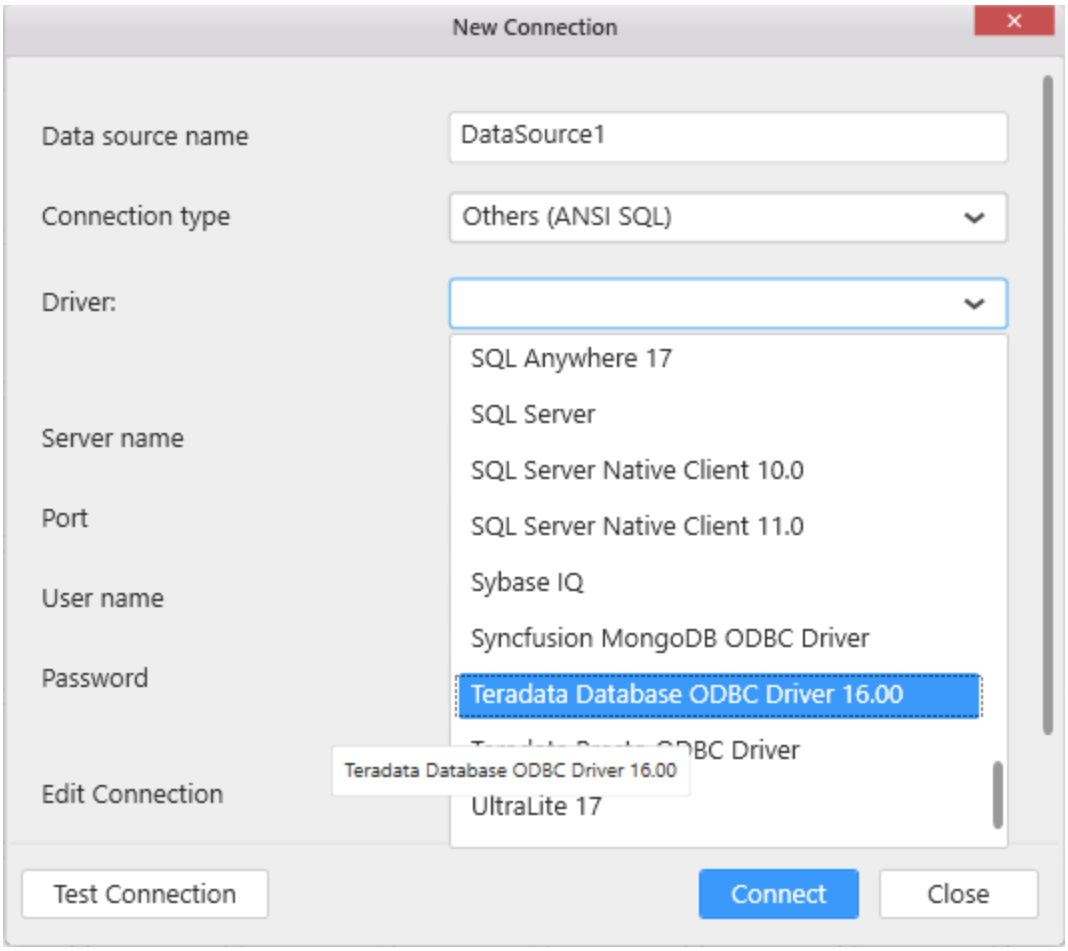
      •   Relative Date Filters

      •   Changing data types dynamically

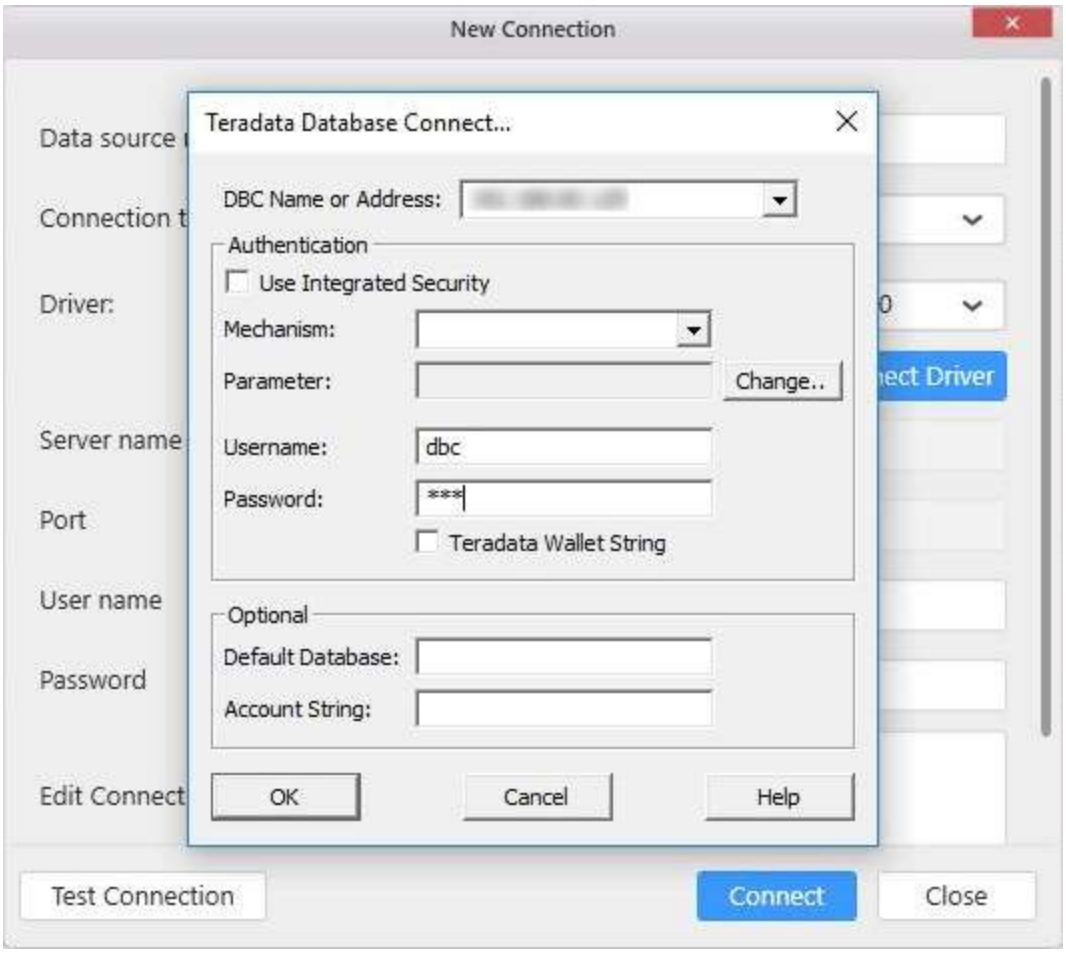
To start with, first download and install the ODBC driver provided by the respective database vendor. In this example, we have downloaded and configured Teradata ODBC driver available for Windows OS. Once the installation is over, launch our dashboard designer application.

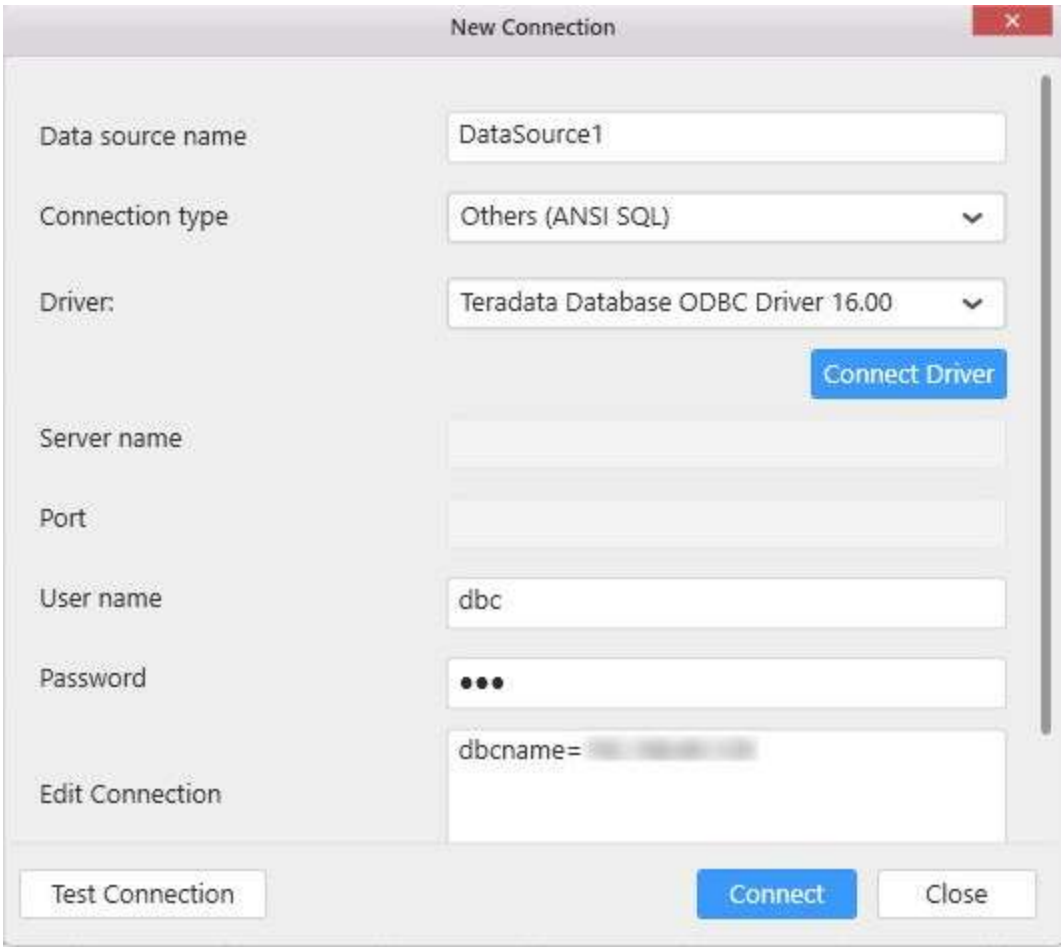
Navigate to the Add Data Source section under Data Sources and you may now find a new connection wizard". Change the connection type as Others (ANSI SQL) and select the driver which you have installed.



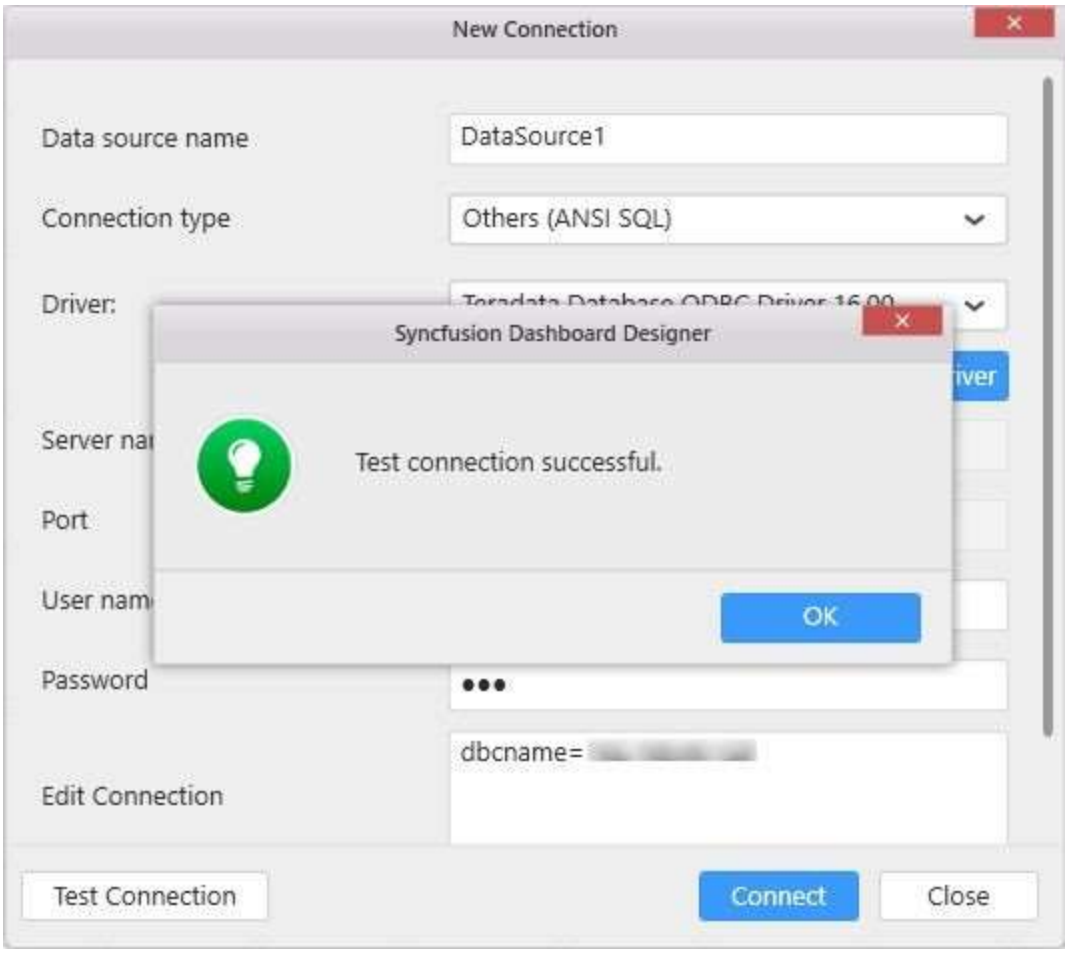


Once the driver is selected, go to Connect Driver option where database server information needs to be entered in-order to generate an appropriate connection string and establish a connection.

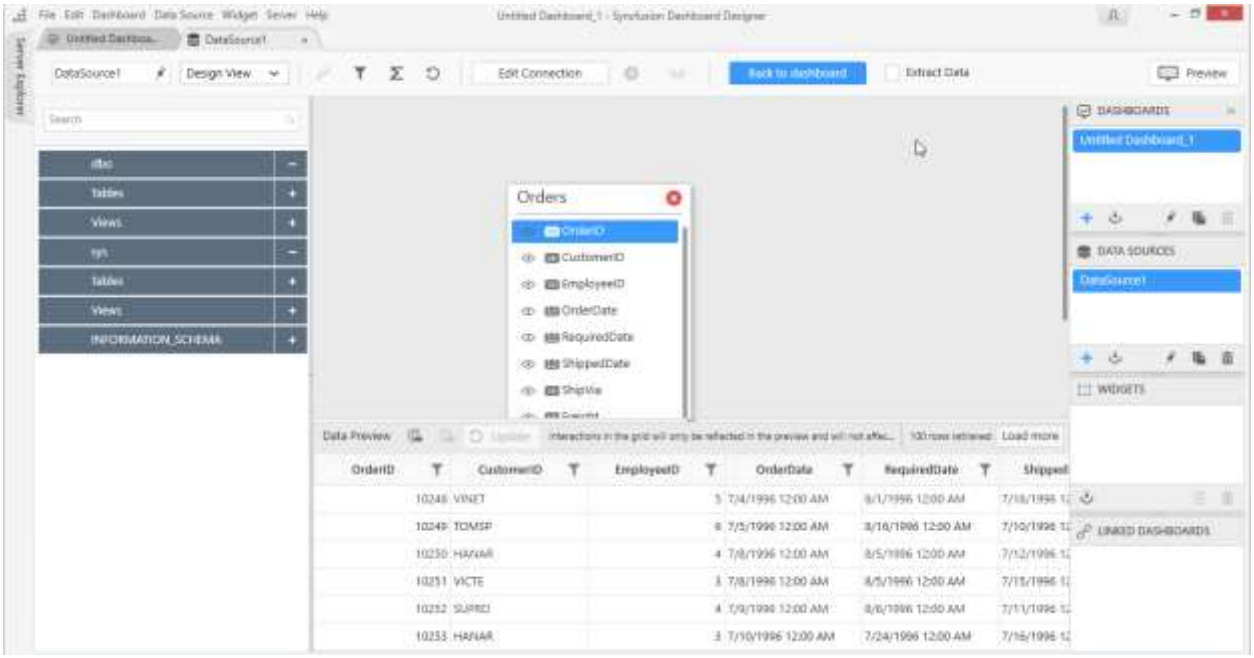




After entering the database server information, you can check your connection using Test Connection option.



On successful connection, now you are allowed to access the respective database. Further, database schema is obtained from which table information are extracted and populated in the designer canvas. Finally you are now ready to design a dashboard with tables or views available. If you need more knowledge about designing our dashboard, just click here.



Data Connection Drivers

The compliant databases whose connectivity through Syncfusion Dashboard Designer was officially confirmed with ANSI SQL standard compliant ODBC drivers are listed below.

Database	Driver Name	Driver Details
IBM DB2	IBM DB2 ODBC DRIVER – DB2 COPY1	https://www-01.ibm.com/marketing/iwm/iwm/web/pick.do?source=swg-db2expressc&S_CMP=db2teamblog (DB2 ODBC Driver install during the DB2 server installation)
Advantage Database Server (ADS)	Advantage StreamlineSQL ODBC	https://www.sap.com/cmp/dg/crm-xu15-int-ads12ddm/index.html (ADS ODBC Driver install during the ADS server installation)
Firebird	Firebird/InterBase[r] driver	https://firebirdsql.org/en/odbc-driver
Google BigQuery	Simba ODBC Driver for Google BigQuery	http://www.simba.com/product/google-bigquery-driver-with-sql-connector
Sybase ASE	Adaptive Server Enterprise	https://www.sap.com/cmp/syb/crm-xu15-int-asewindm/index.html (Sybase ASE ODBC Driver install during the Sybase ASE server installation)
Sybase IQ	Sybase IQ ODBC Driver	https://www.sap.com/cmp/syb/crm-xm13-dtb-dbtch-tr20/crm-xm13-dtb-dbtch-tr20/index.html (Sybase IQ ODBC Driver install during the Sybase IQ server installation)

| Teradata | Teradata Database ODBC Driver | http://downloads.teradata.com/download/connectivity/odbc-driver/windows |

| EXASolution | EXASolution Driver | https://www.exasol.com/portal/display/DOWNLOAD/6.0 |

| Splunk | Splunk ODBC Driver | https://splunkbase.splunk.com/app/1606/ |

| Aster Database | Aster ODBC Driver | http://downloads.teradata.com/download/aster/aster-client-tools-for-windows |

| Actian Vector | Ingres ODBC Driver | http://esd.actian.com/ |

| IBM PDA (Netezza) | NetezzaSQL | https://www-01.ibm.com/marketing/iwm/iwm/web/reg/download.do?source=swg-im-ibmndn&lang=en*US&S*PKG=d12&cp=UTF-8&dlmethod=http |

| SAP HANA | HDBODBC | https://www.sap.com/cmp/ft/crm-xu16-dat-hddedft/typ.html |

| Sap SQL Anywhere | SQL Anywhere | https://www.sap.com/india/products/sql-anywhere.licensing-purchasing.html (SAP SQL Anywhere ODBC Driver install during the SQL Anywhere server installation)|

| MarkLogic | MarkLogic SQL ODBC Driver | https://developer.marklogic.com/products/odbc|

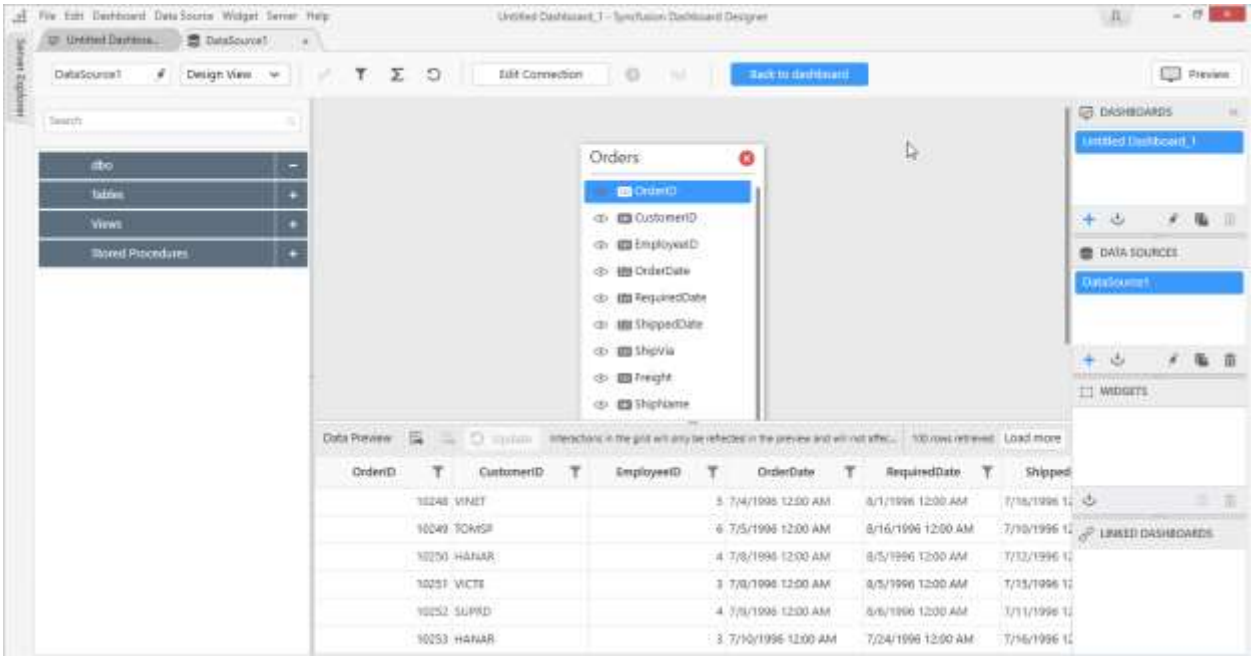
| MongoDB | Simba MongoDB ODBC Driver | https://www.simba.com/product/mongo-driver-with-sql-connector/|

| Apache Cassandra | Simba Cassandra ODBC Driver | https://www.simba.com/product/cassandra-driver-with-sql-connector/|

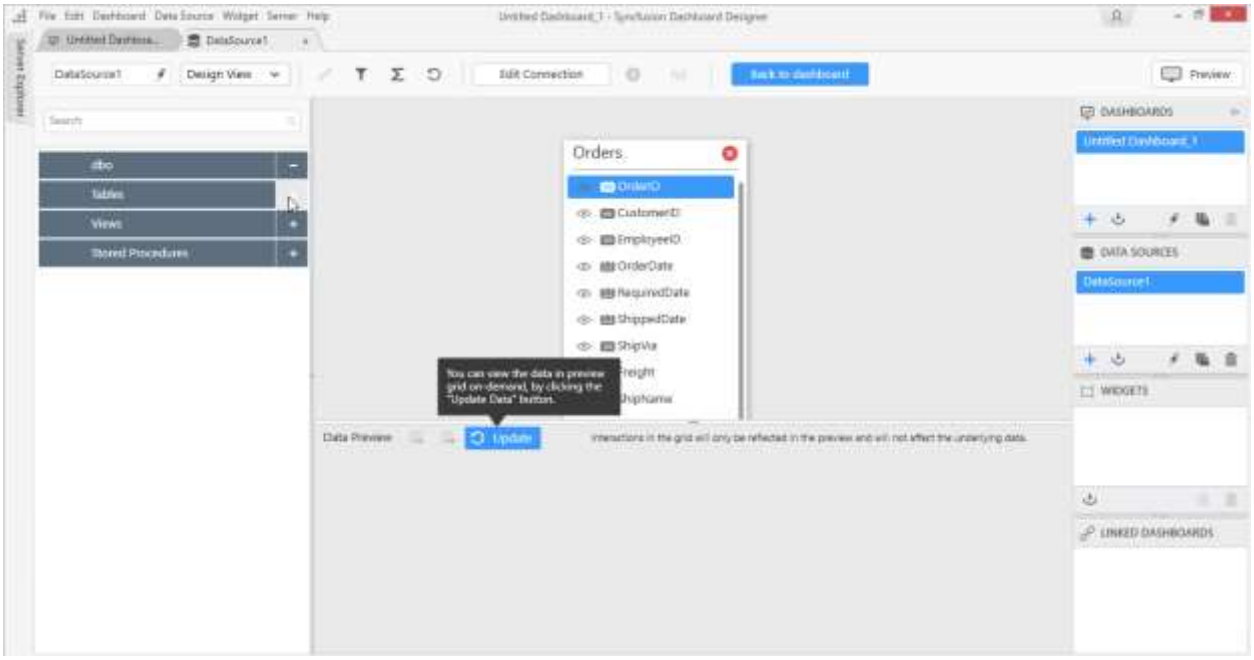
| Amazon DynamoDB | CData DynamoDB ODBC Driver | https://www.cdata.com/drivers/dynamodb/odbc/

Data Preview Grid

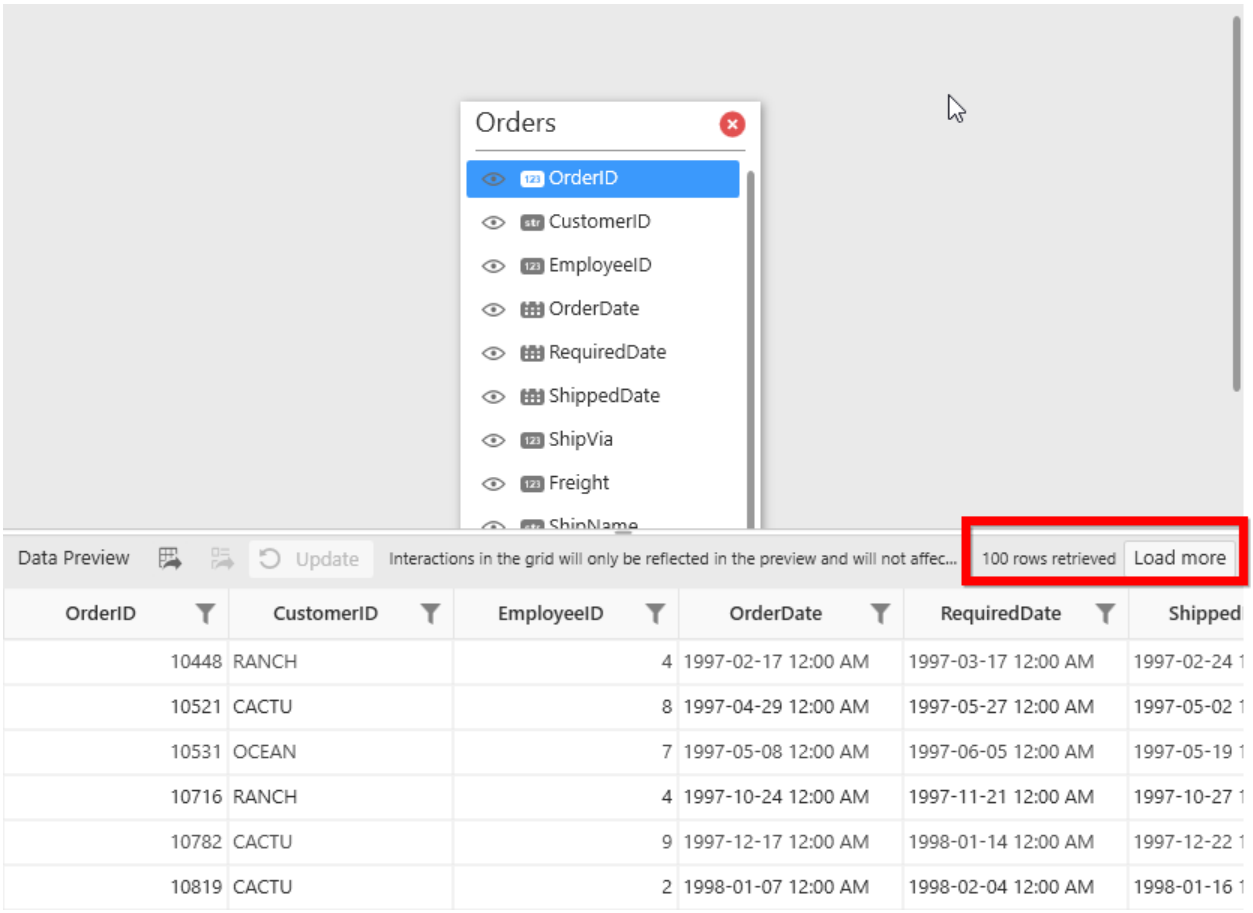
Data Preview grid is used to show the raw data of the selected table(s) in table canvas.



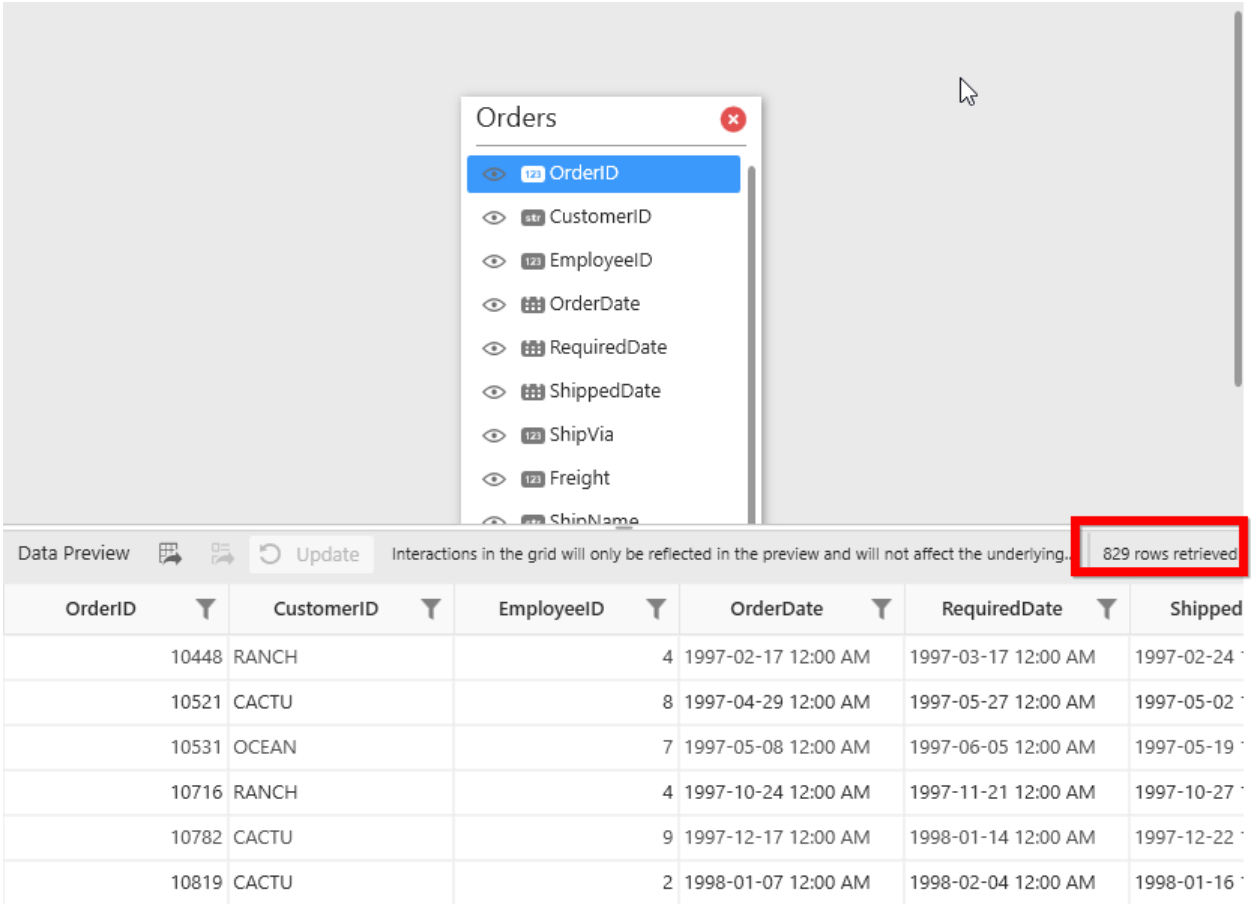
By default Data Preview grid shows like below.



By clicking Update button, the data preview grid loaded and initially it shows only 100 records.

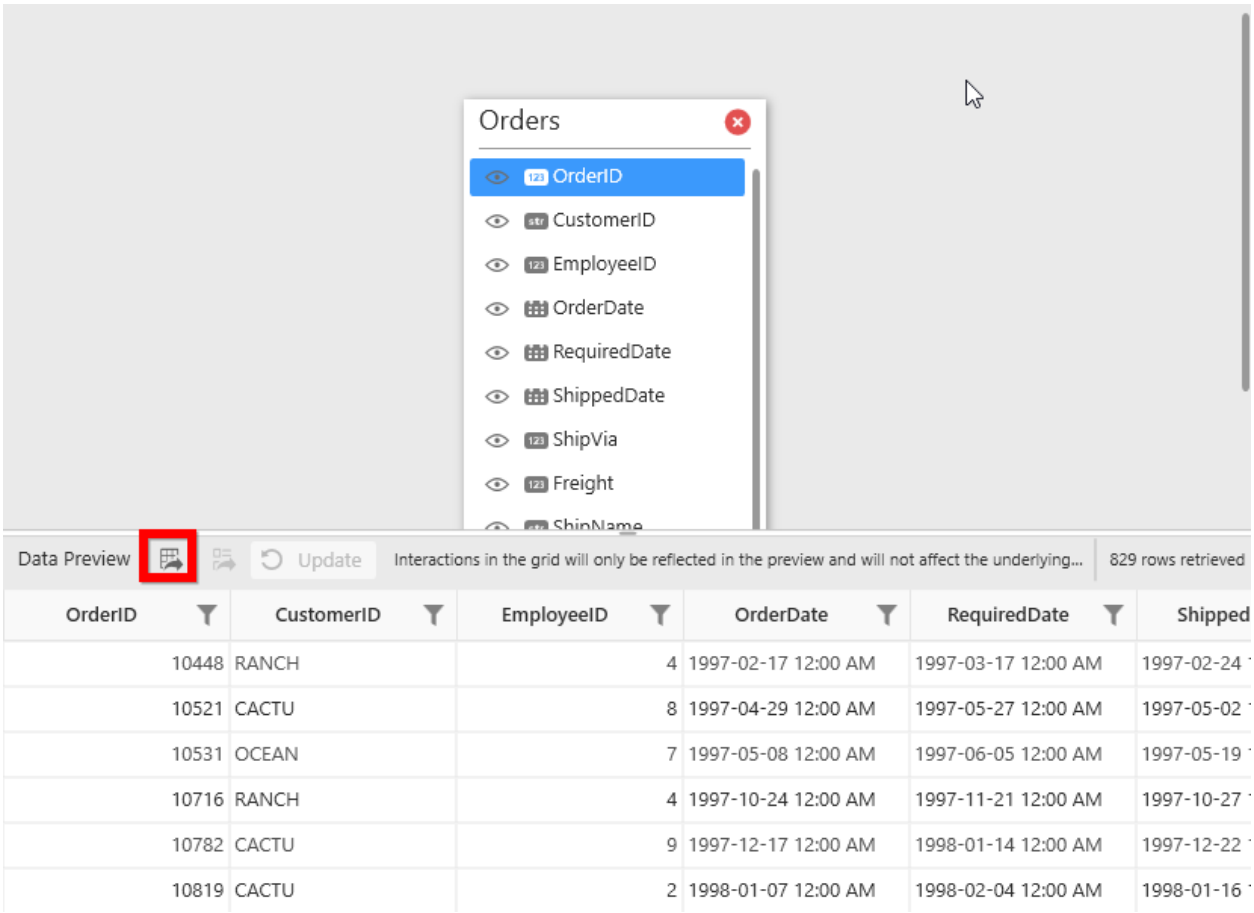


By clicking Load more option, you can get all data in preview grid.



By following below procedure, you can export the data to excel file which are displayed in data preview grid.

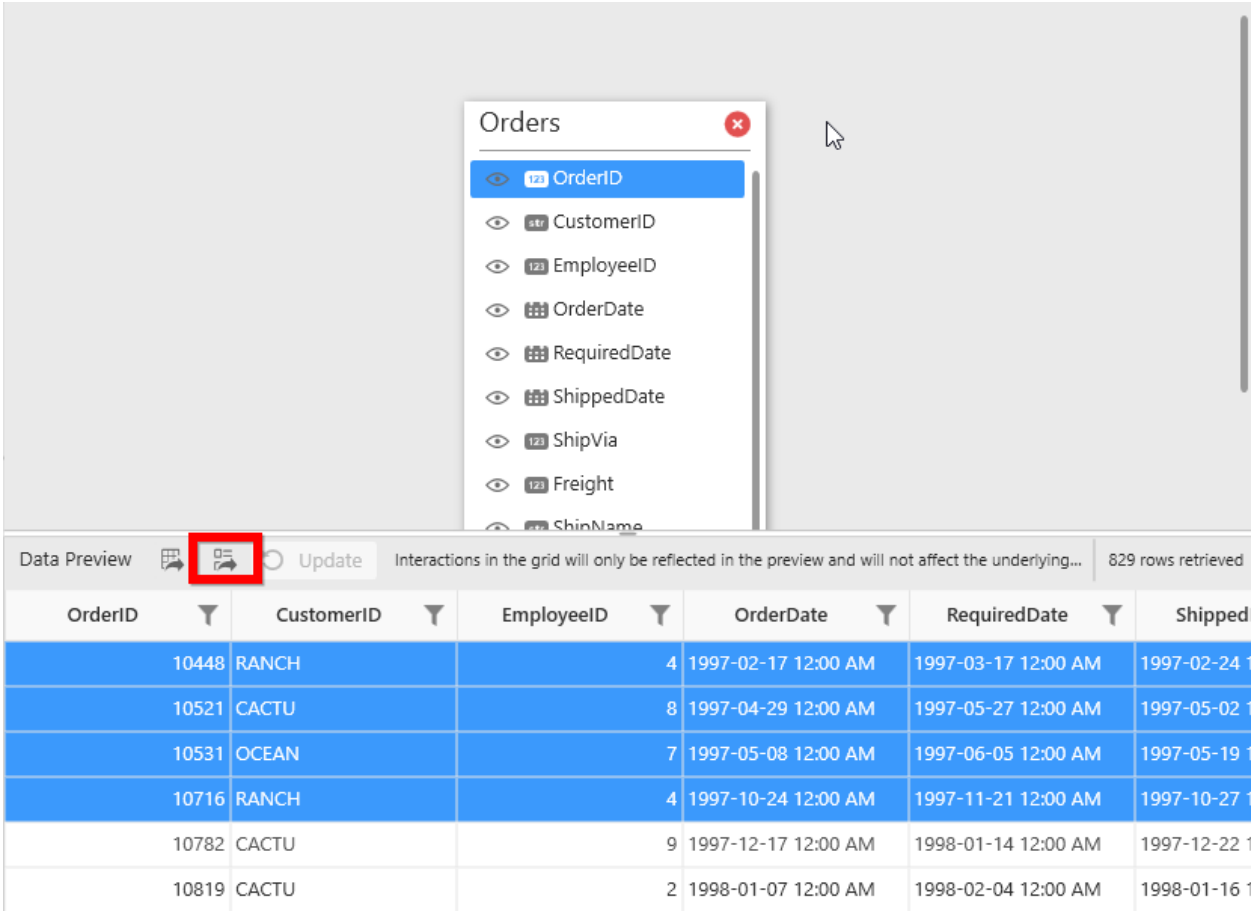
Click Export All icon in data preview grid.



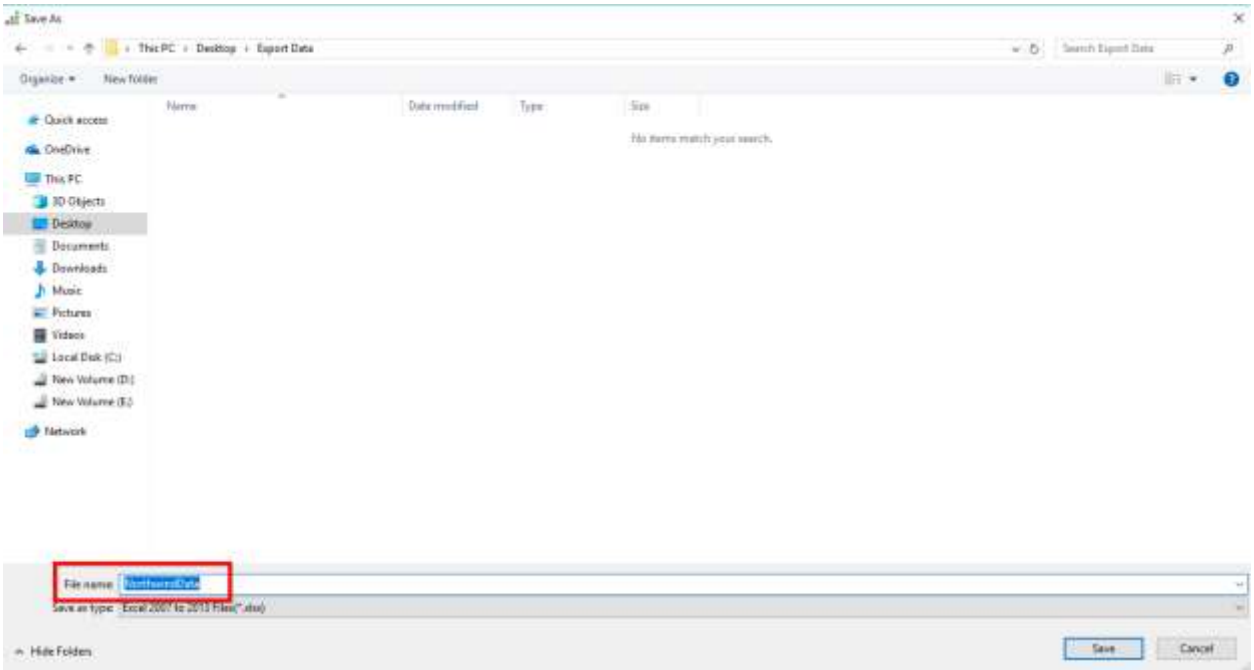
OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	Shipped
10448	RANCH	4	1997-02-17 12:00 AM	1997-03-17 12:00 AM	1997-02-24
10521	CACTU	8	1997-04-29 12:00 AM	1997-05-27 12:00 AM	1997-05-02
10531	OCEAN	7	1997-05-08 12:00 AM	1997-06-05 12:00 AM	1997-05-19
10716	RANCH	4	1997-10-24 12:00 AM	1997-11-21 12:00 AM	1997-10-27
10782	CACTU	9	1997-12-17 12:00 AM	1998-01-14 12:00 AM	1997-12-22
10819	CACTU	2	1998-01-07 12:00 AM	1998-02-04 12:00 AM	1998-01-16

Export Selected Rows option will be enabled by selecting the rows in the data preview grid.

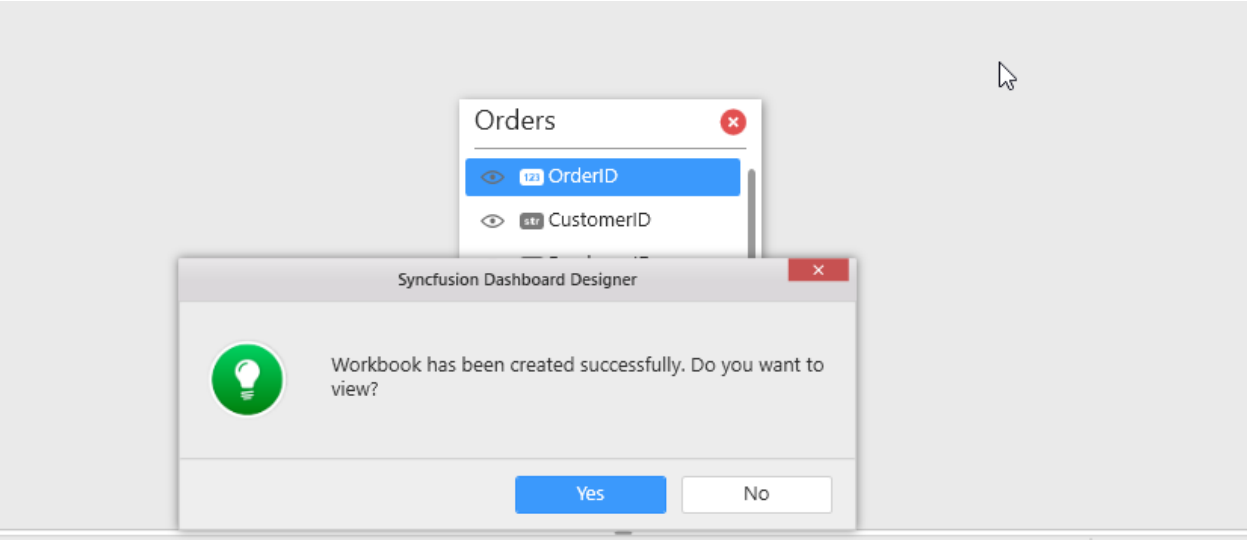
Click Export Selected Rows icon in data preview grid to export selected row(s) of the data.



Now Save as window will open. Enter suitable name and choose the specified location. By clicking Save button, you can save the data in excel format.



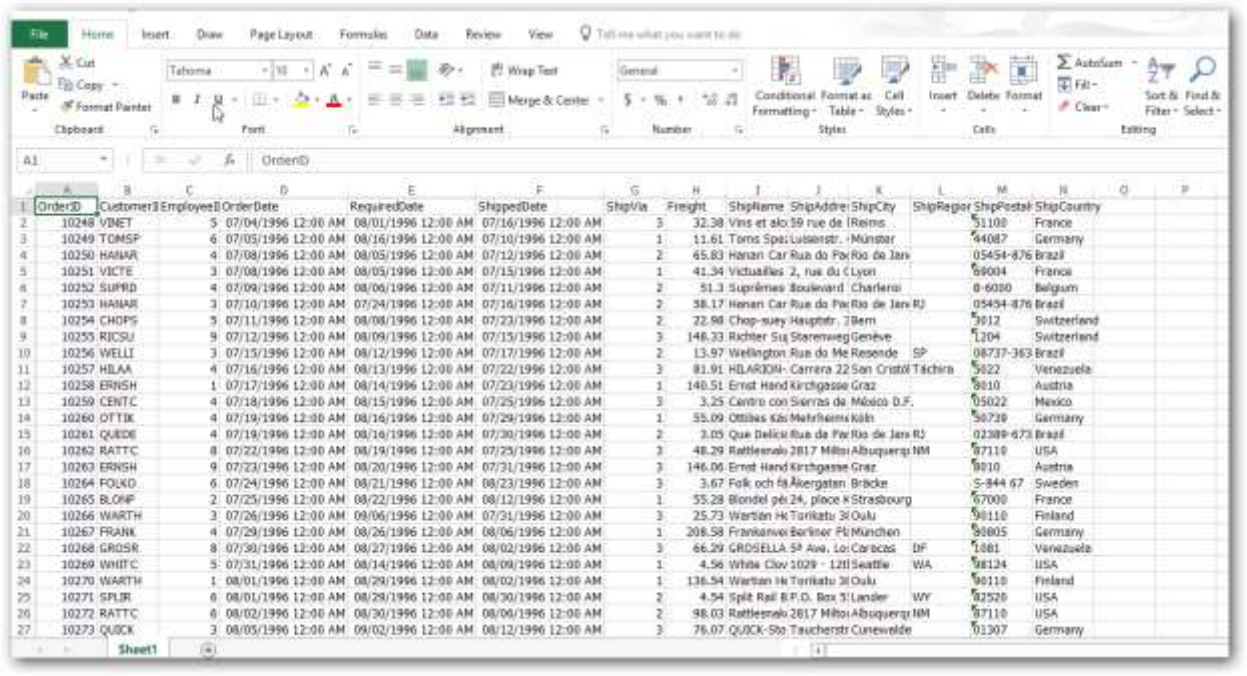
Once data saved successfully, you will get message as follows.



Data Preview | Update | Interactions in the grid will only be reflected in the preview and will not affect the underlying... | 829 rows retrieved

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	Shipped
10494	COMMI	4	1997-04-02 12:00 AM	1997-04-30 12:00 AM	1997-04-09 1
10969	COMMI	1	1998-03-23 12:00 AM	1998-04-20 12:00 AM	1998-03-30 1
11042	COMMI	2	1998-04-22 12:00 AM	1998-05-06 12:00 AM	1998-05-01 1
10254	CHOPS	5	1996-07-11 12:00 AM	1996-08-08 12:00 AM	1996-07-23 1
10370	CHOPS	6	1996-12-03 12:00 AM	1996-12-31 12:00 AM	1996-12-27 1
10519	CHOPS	6	1997-04-28 12:00 AM	1997-05-26 12:00 AM	1997-05-01 1

Click Yes to view the saved data in excel file as follows.

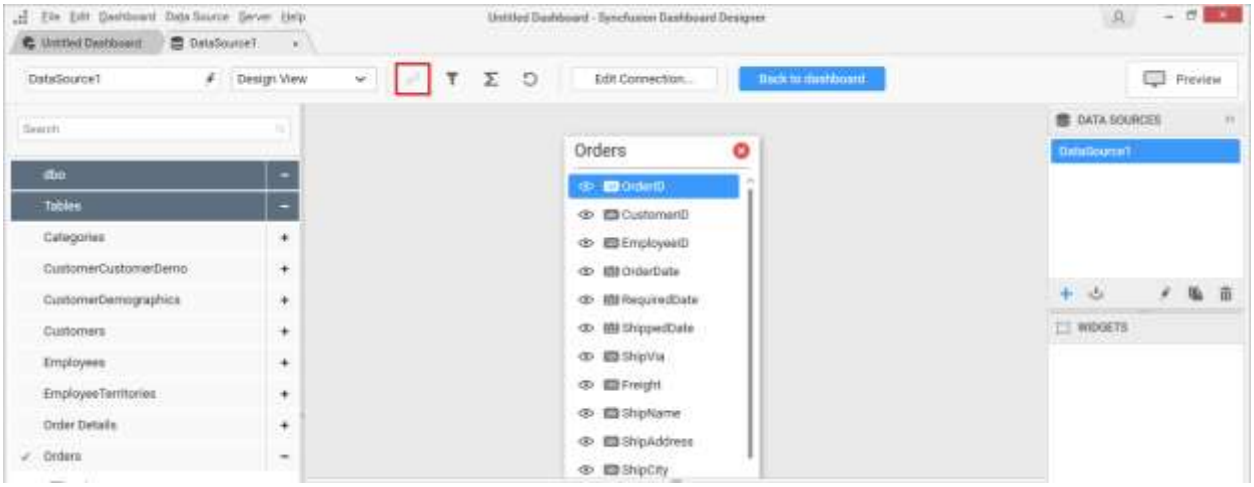


Transforming Data

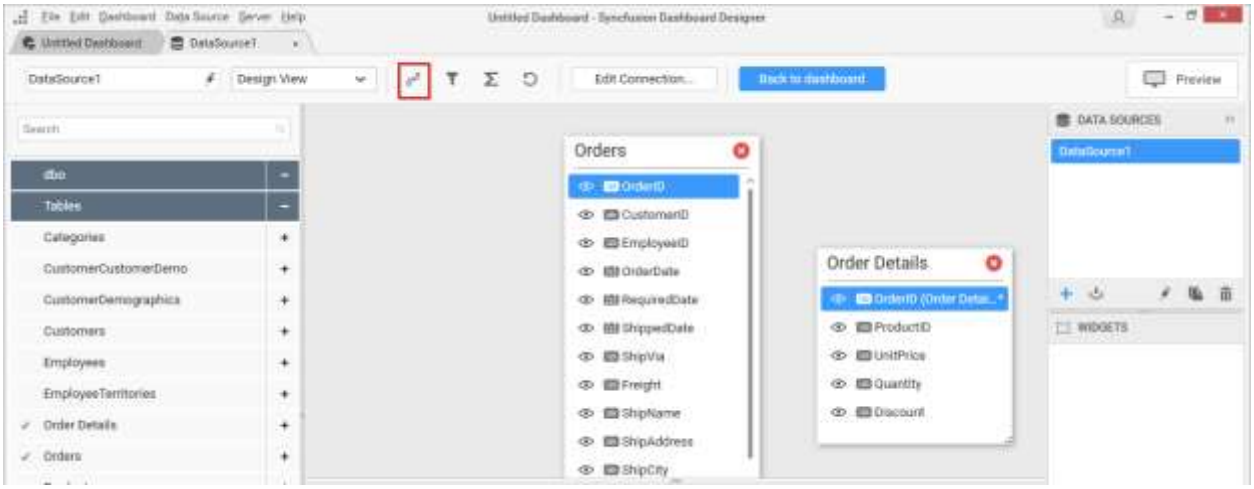
Joining Tables

Note: For Microsoft SQL Server Analysis Services connection type, table joining is not applicable.

Joining of tables is required when you are going to use more than one table in your data source design. The join icon (highlighted below) in the tools pane at data design view will be in disabled state, if there was only one table found dropped in table design view like below:

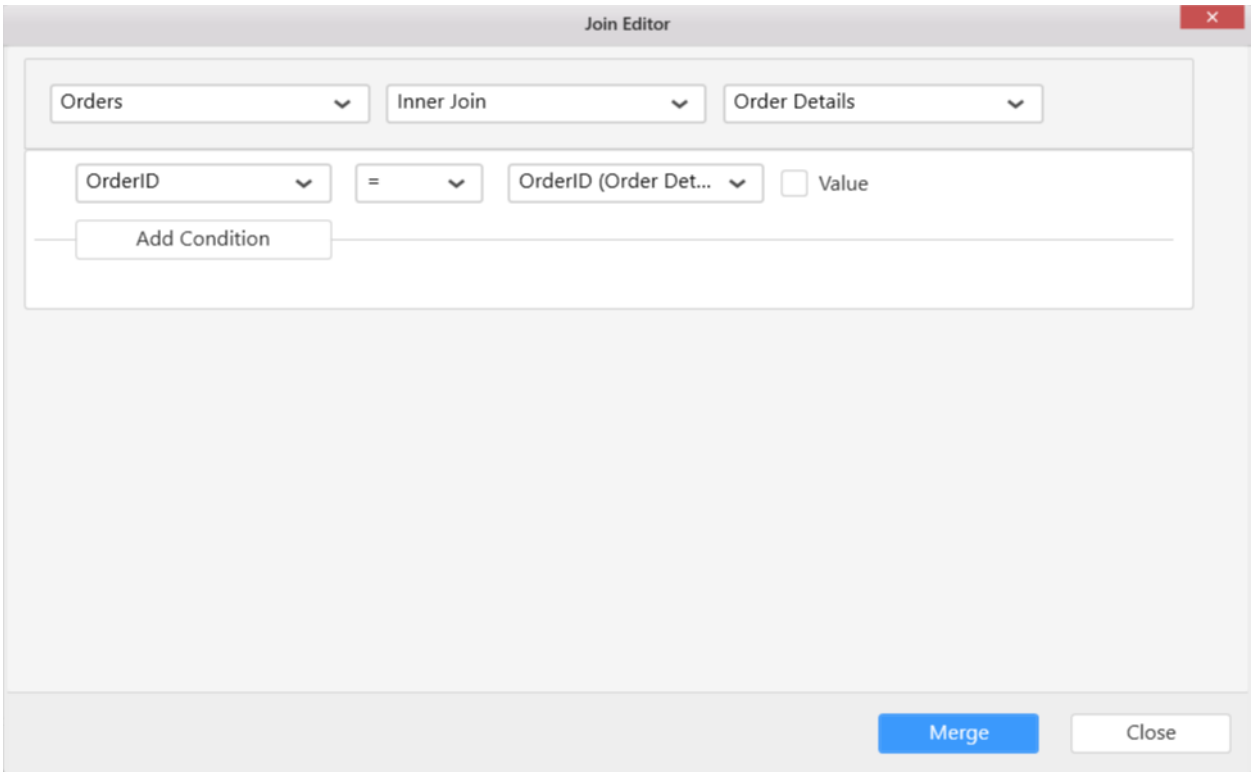


It will get enabled once you drop the 2nd table like below:



Adding a join condition

If the subsequent table being dropped, has any of its column as foreign key in any of the already dropped tables, the joining will take place automatically. Else, it will prompt the join editor like below to let you define the keys (columns) to join between this table and any one of the already dropped tables.

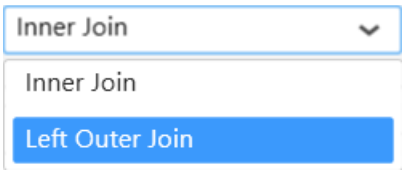


In the above screen shot, the LeftTable illustrate the list of table dropped already. The RightTable illustrate the table which you have dropped recently and that requires to set up a relation with any of the previously dropped tables. The drop-down list below to that, represent the join condition. You can add multiple join condition for single table relation just by click Add Condition button.

The join type, compare operator and relational operator to make relationship between the two tables, can be defined through the options available in join editor.

Join Types

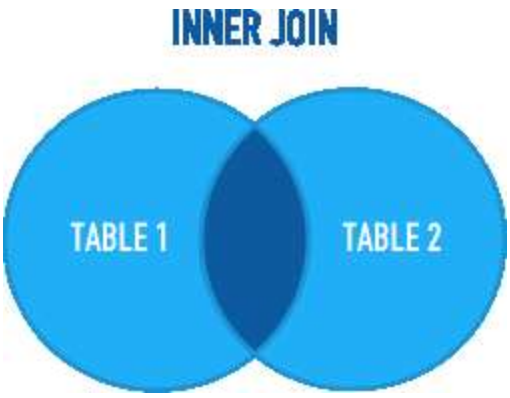
Two types of joins can be made between tables in join editor. They are Left Outer Join and Inner Join.



Inner Join

INNER JOIN will return the records from two or more tables, while records are matching in both the tables.

An inner join of Table1 and Table2 gives the result of Table1 intersect Table2, i.e. the inner part of a Venn diagram intersection.



For example, consider the below two tables.

Table1

Supplier*Id*	*Supplier*Name
100	James
101	John
102	Robert
103	Michael

Table2

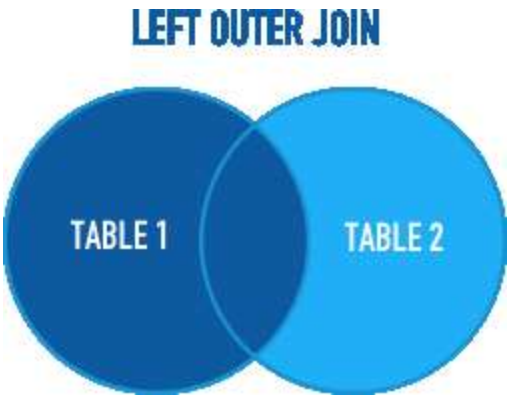
Order*Id*	*Supplier*Id	Order_Date
20125	100	09/21/2017
20126	101	09/22/2017
20127	104	09/23/2017

If we join (INNER JOIN) Table1 and Table2 based on Supplier_Id column and equals (=) as comparison operator, then Syncfusion Dashboard will return the result like below.

Supplier*Id*	*Supplier*Name	Order*Id*	*Supplier*Id(Table2)	Order_Date
100	James	20125	100	09/21/2017
101	John	20126	101	09/22/2017

Left outer join

LEFT OUTER JOIN will return all record from the left table and the matched records from the right table. The result is NULL from the right table, if there is no match.



For example, consider the below two tables.

Table1

Supplier*Id*	*Supplier*Name
100	James
101	John
102	Robert
103	Michael

Table2

Order*Id*	*Supplier*Id	Order_Date
20125	100	09/21/2017
20126	101	09/22/2017
20127	104	09/23/2017

If we join (LEFT OUTER JOIN) Table1 and Table2 based on Supplier_Id column and equals (=) as comparison operator, then Syncfusion Dashboard will return the result like below.

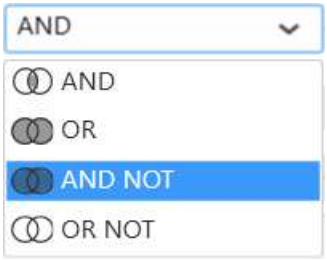
Supplier*Id*	*Supplier*Name	Order*Id*	*Supplier*Id(Table2)	Order_Date
100	James	20125	100	09/21/2017
101	John	20126	101	09/22/2017
102	Robert			
103	Michael			

Join Condition

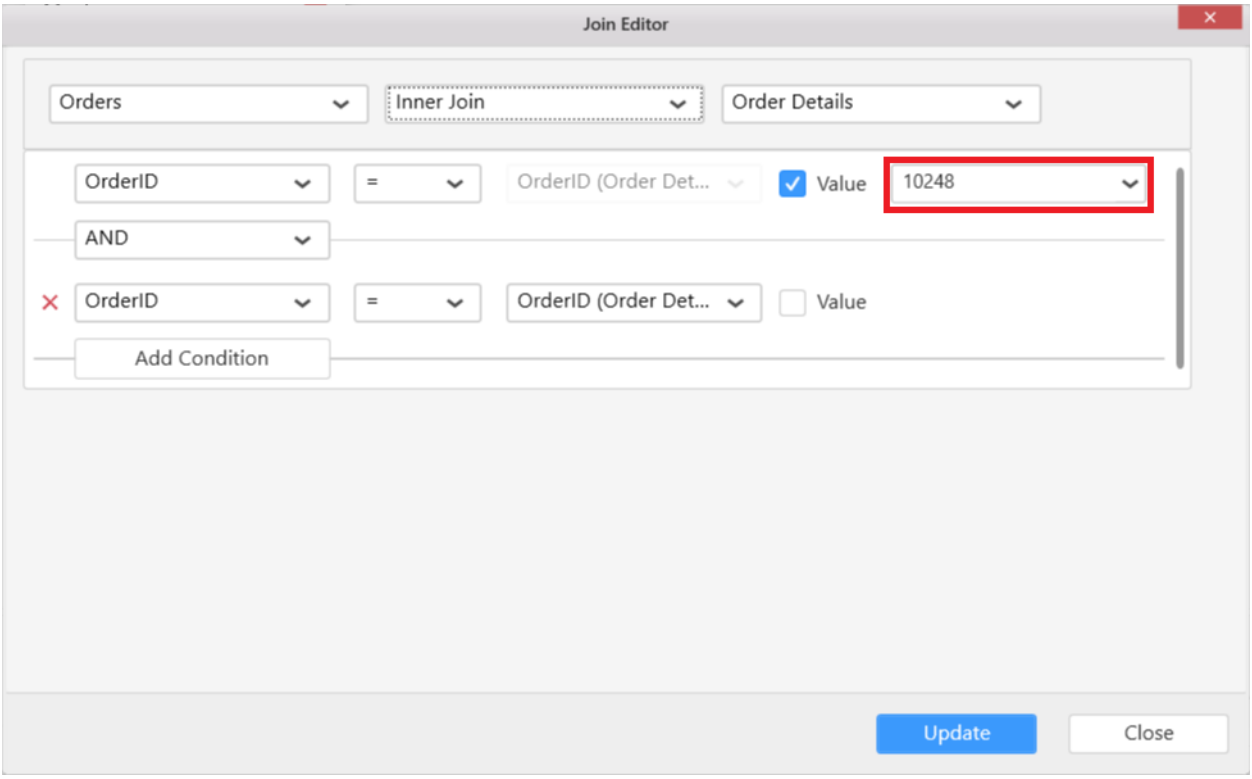
You can define a condition for joining two tables through any of the compare operator for comparing the values of the two columns (one from each table) by which relation between tables need to be made.

Join Condition Relationship

You can define the relationship for joining, with multiple condition, in the condition selection block.



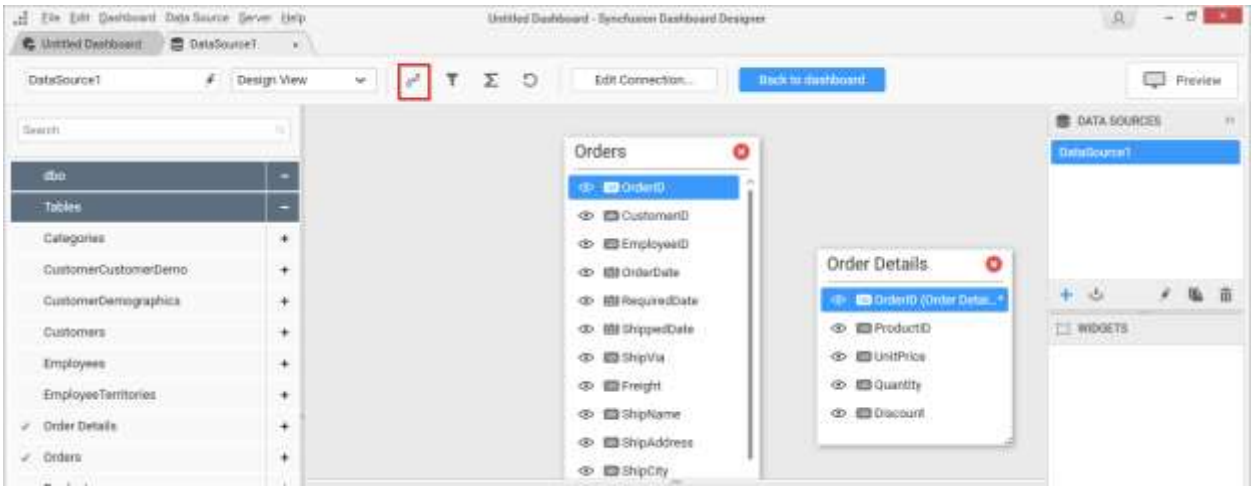
You can also create condition using a constant value instead of choosing column as right operand to join tables.



Updating a join condition

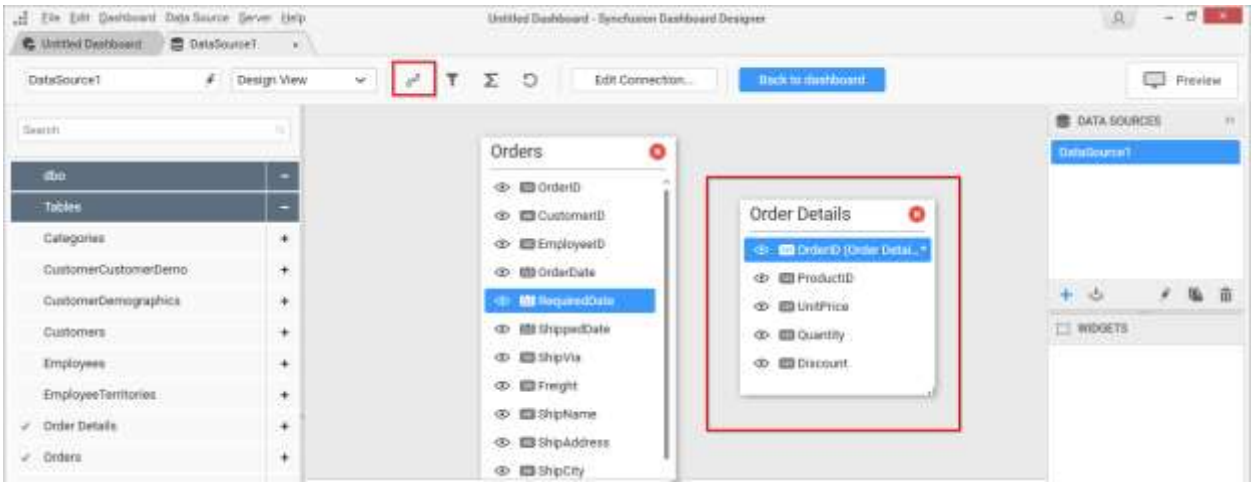
Update an existing join condition through selecting that in the top table and then edit the mapping between columns through interacting with columns list, join type and compare operator.

If you are not at the join editor, it can be invoked through clicking the highlighted icon below in the data design view.



Note: Updating an existing join condition will allow you to edit the column mapping only between those two tables.

Click Update to save the changes that you made.



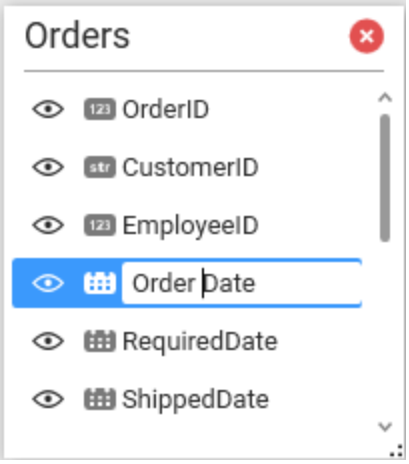
Click Close to close the join editor.

Formatting Columns

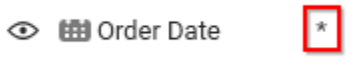
Renaming Column

Note: For Microsoft SQL Server Analysis Services connection type, column renaming is not supported.

Rename the column, if required, through double clicking the respective column to enable the edit mode and typing the modified name. Press <Enter> key to commit the modification done.

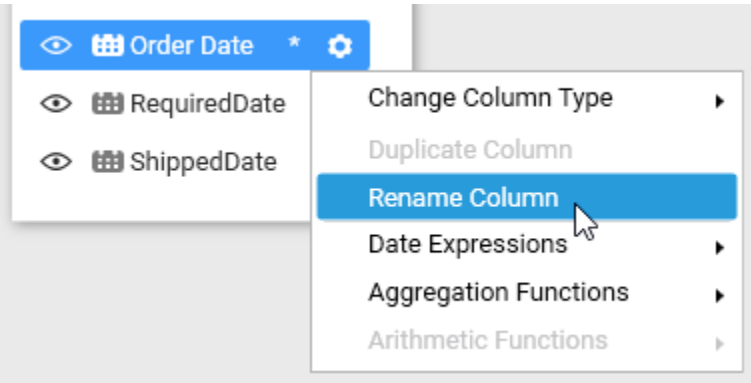


The renamed column will be represented by (*) mark.



You can also rename the column through the Settings icon which will be displayed like below while hovering the respective column.

Click this icon to drop down the menu with Rename Column, clicking which the edit mode will be enabled in that column.



Handling Column Type Conversion

Note: For Microsoft SQL Server Analysis Services Connection type, type conversions can be handled only on string typed columns. The possible conversions are Text to Date Time and vice-versa.

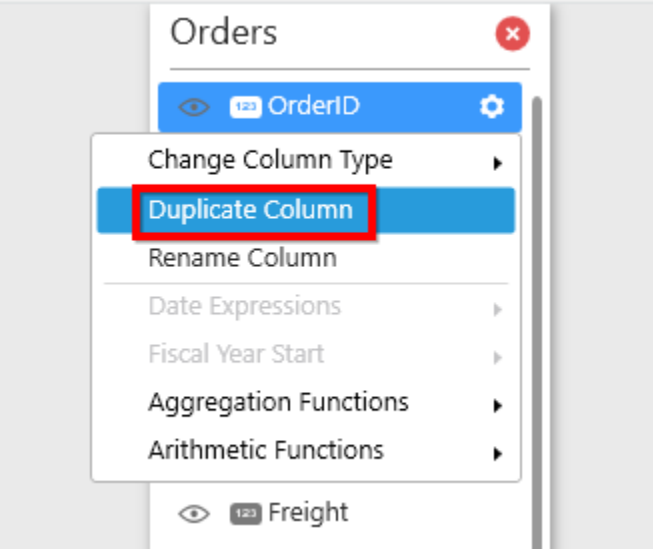
The following table represents the column types and their equivalent convertible types that are supported in Dashboard Designer.

Column Data Type	Equivalent Convertible Types
Decimal	DateTime, Text

Integer	Decimal, DateTime, Text
Date Time	Decimal, Text
Text	DateTime, Decimal

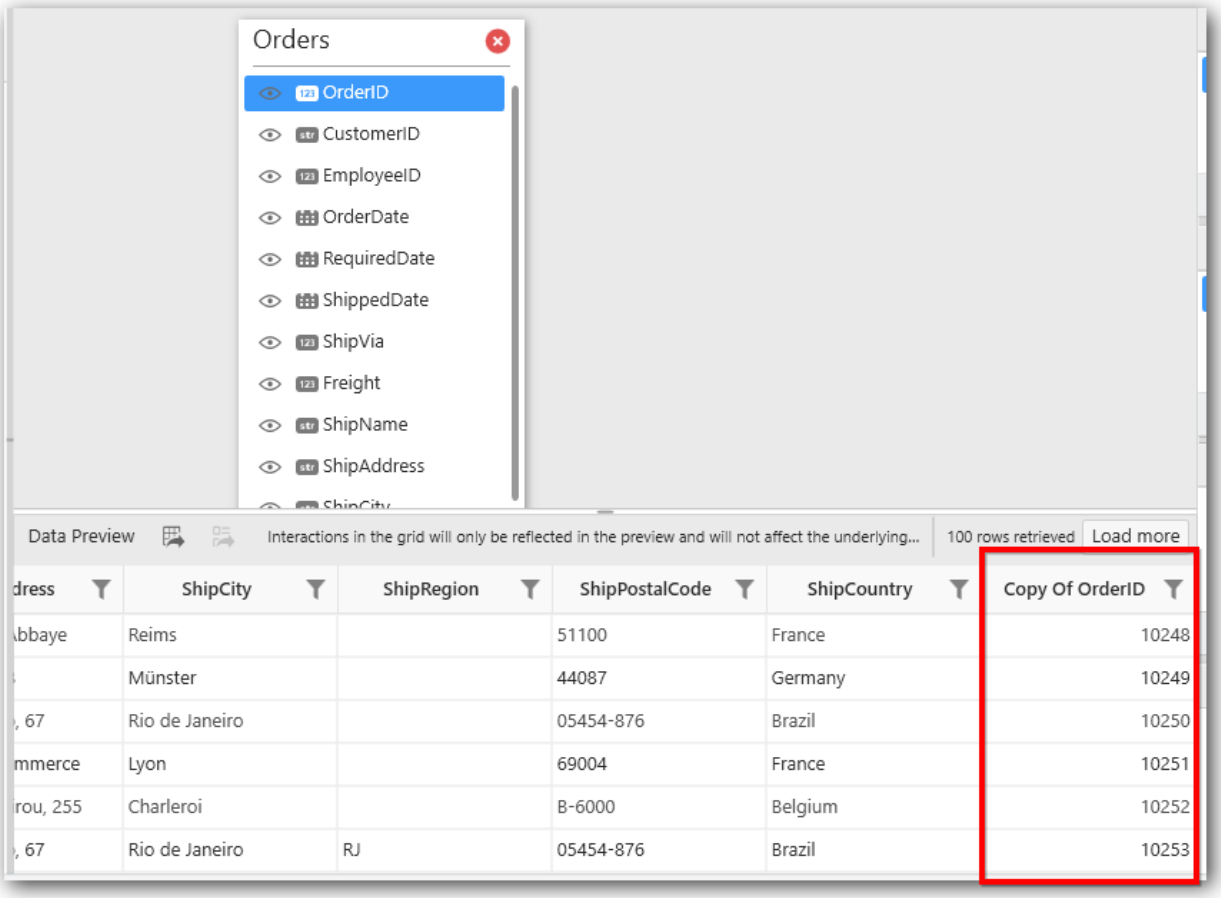
Duplicate Column

You can get the copy of a column through the Settings icon which will be displayed like below while hovering the respective column.



Click Duplicate Column option. It will be consider as a expression column. So this column automatically added in Expression Designer.

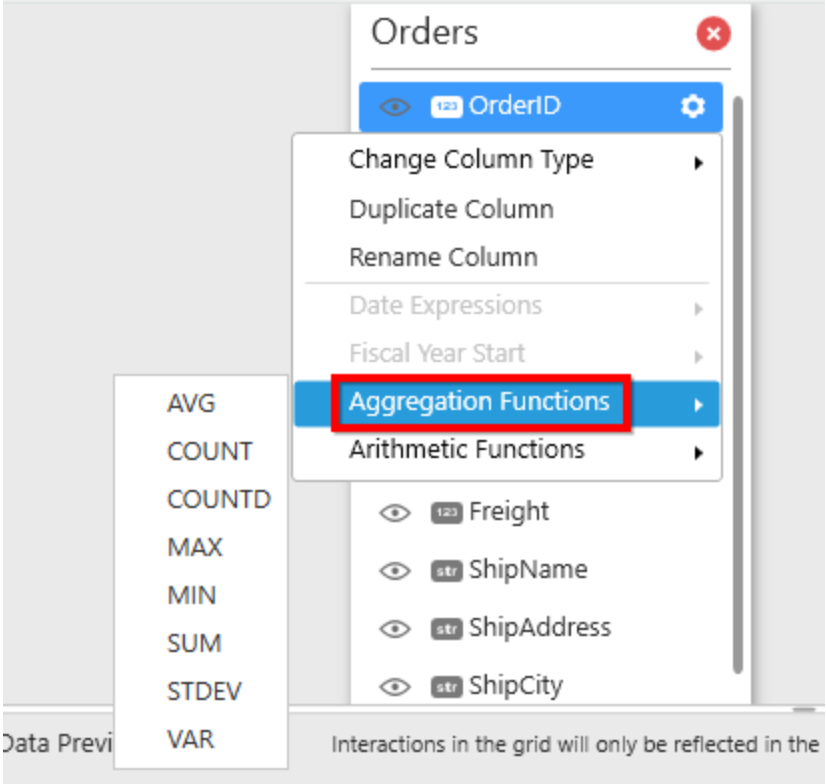
Now duplicate column will display in preview grid as follows.



Note: Duplicate Column option will be enabled only for measure column.

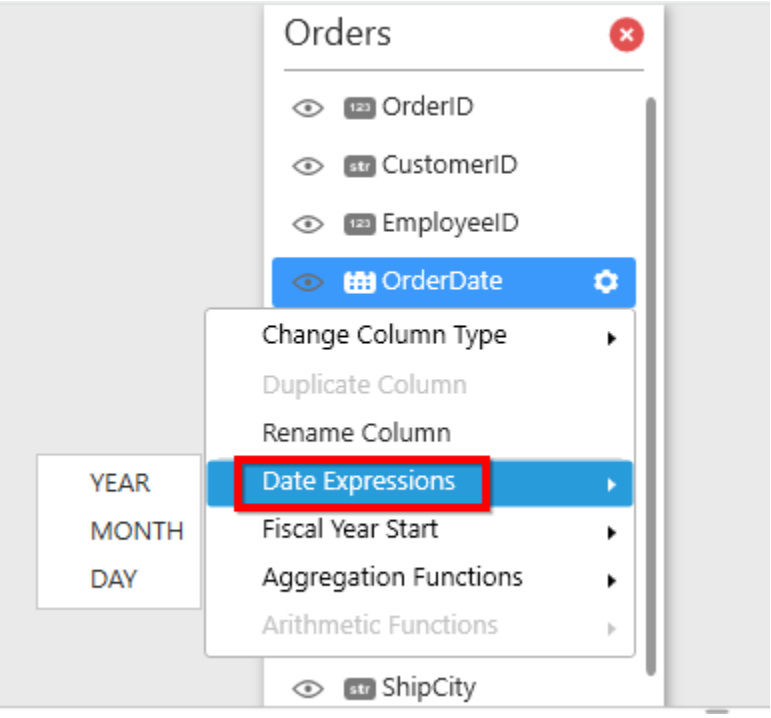
Aggregation Functions

Using Aggregation Functions option, you can get aggregated result for a column in data design view. You can apply this option through the Settings icon which will be displayed like below while hovering the respective column. Click the Aggregation Functions, which displays the list of functions to choose.



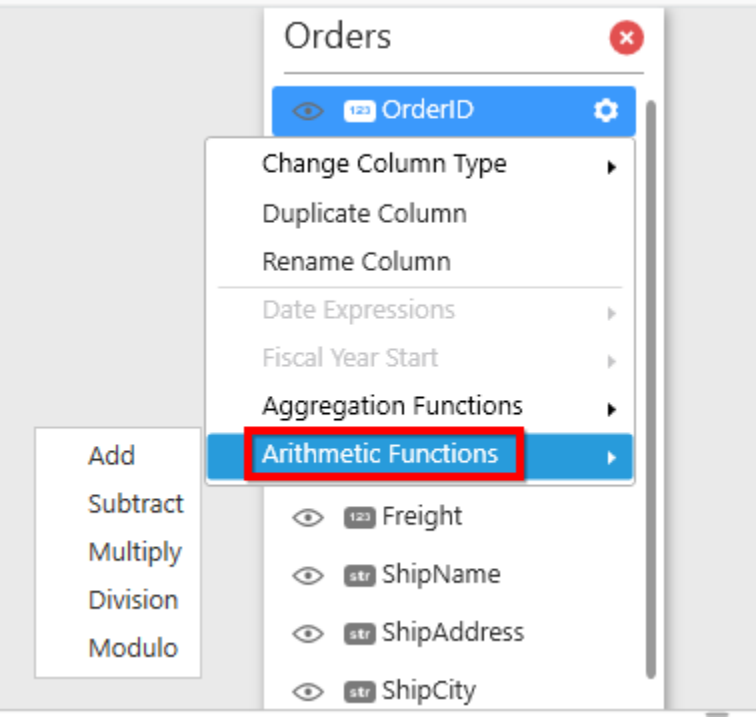
Date Expressions

This option will enable only for Date type column(s). You can apply date expression for a column through Settings icon. Click Date Expressions, which displays date functions like Year, Month, Day.

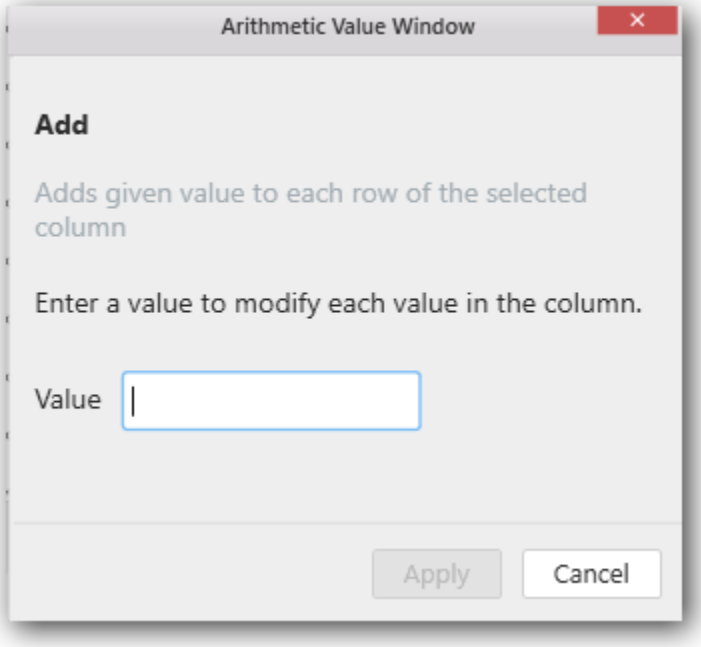


Arithmetic Functions

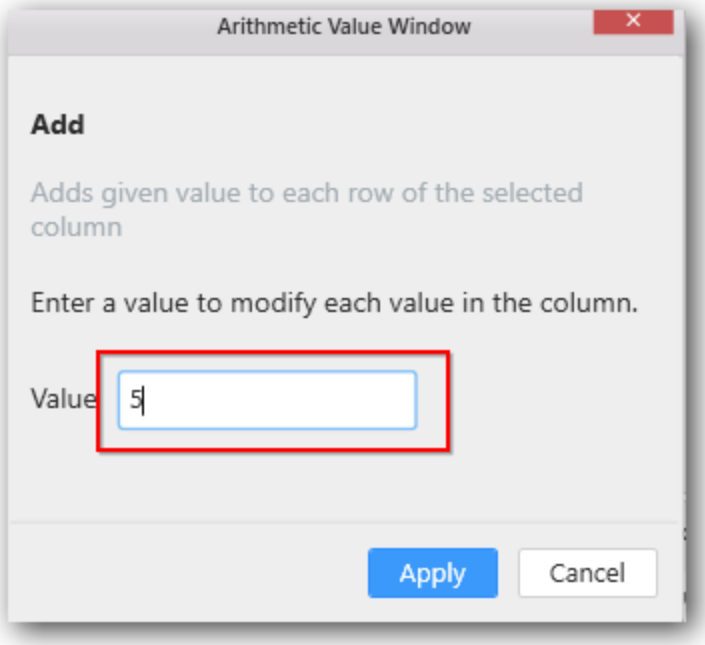
Using Arithmetic Functions option, you can add/subtract/multiply/divide a specific value from a measure column. You can select this option as follows.



Now select the function to pass a specific value, Arithmetic Value Window will open as follows.



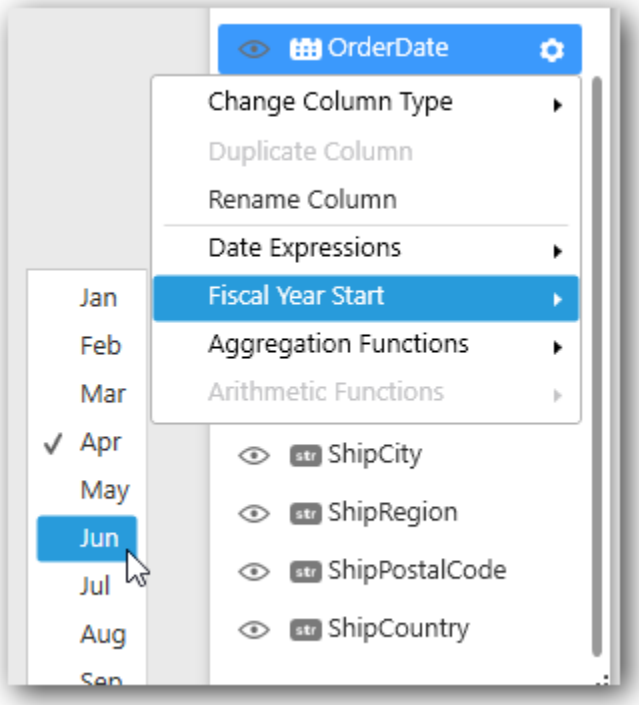
Enter the value in value text box and click Apply.



Note: When you use options like Duplicate Column, Aggregation Functions, Date Expressions, Arithmetic Functions, it will be consider as expression column. You can find these columns in Expression Designer.

Applying the Fiscal Year Start in data Source level

You can apply fiscal format to the date column through the Settings icon which will be displayed like below while hovering the respective date column. Click the Fiscal Year Star, which displays list of months to choose.

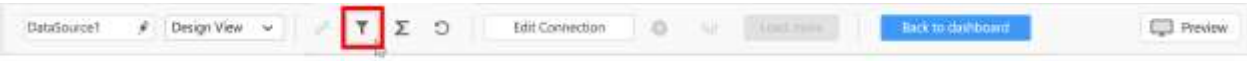


Data filters

Configuring Data Filters

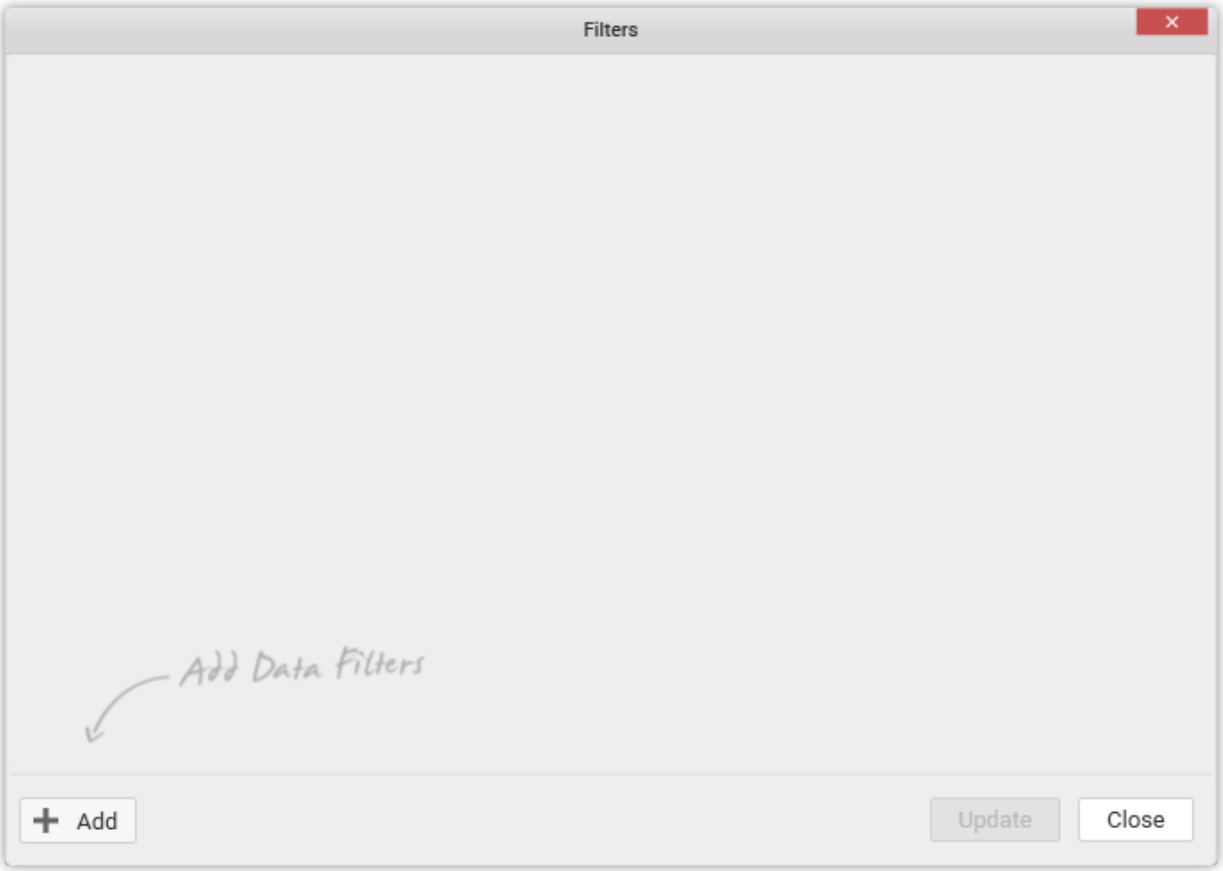
Data Filters can be configured to restrict records visibility based on defined criteria. The configuration can be done by adding and deleting a filter condition.

The Initial filters window can be opened by clicking on the Add filters button provided in the data source tab toolbar. You can also use the keyboard shortcut Ctrl + I to open the initial filters window.

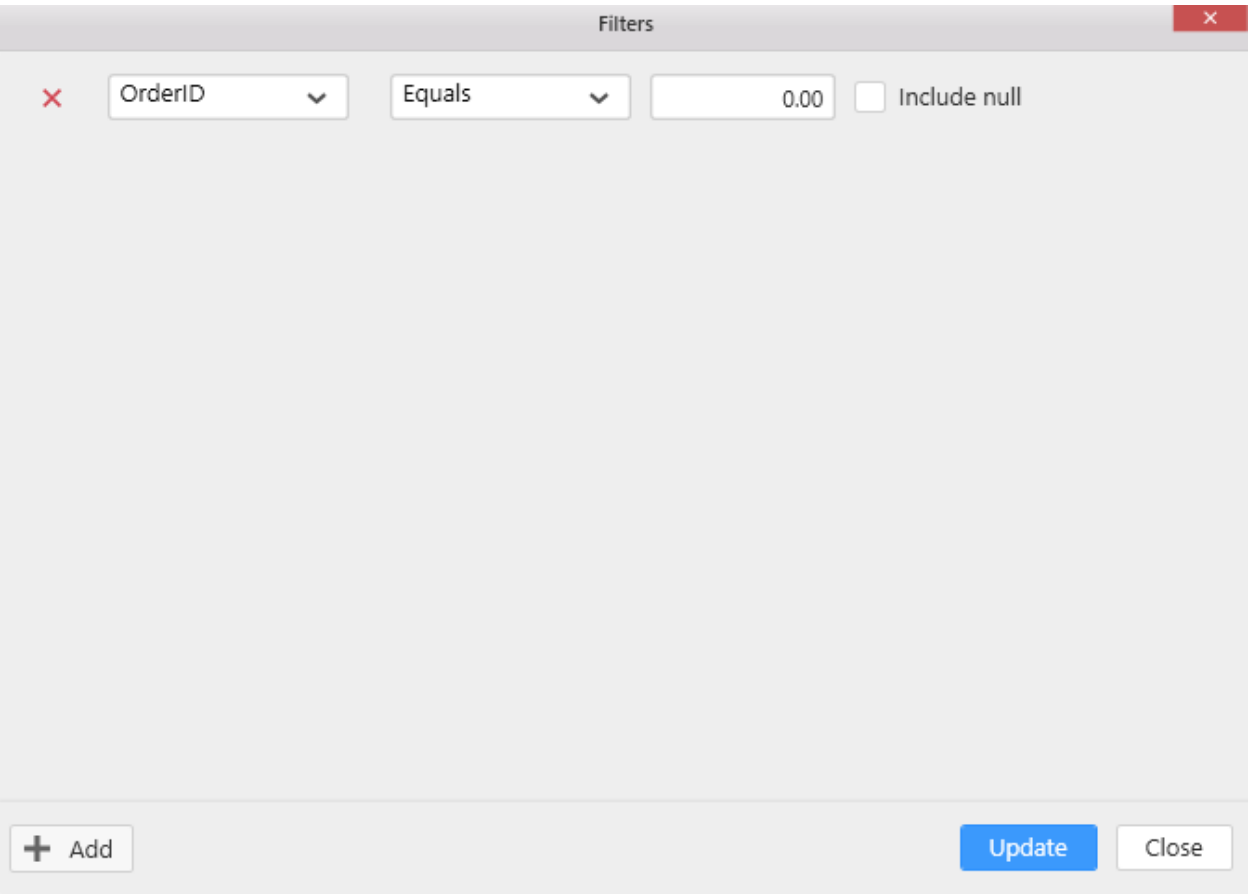


Adding a filter condition

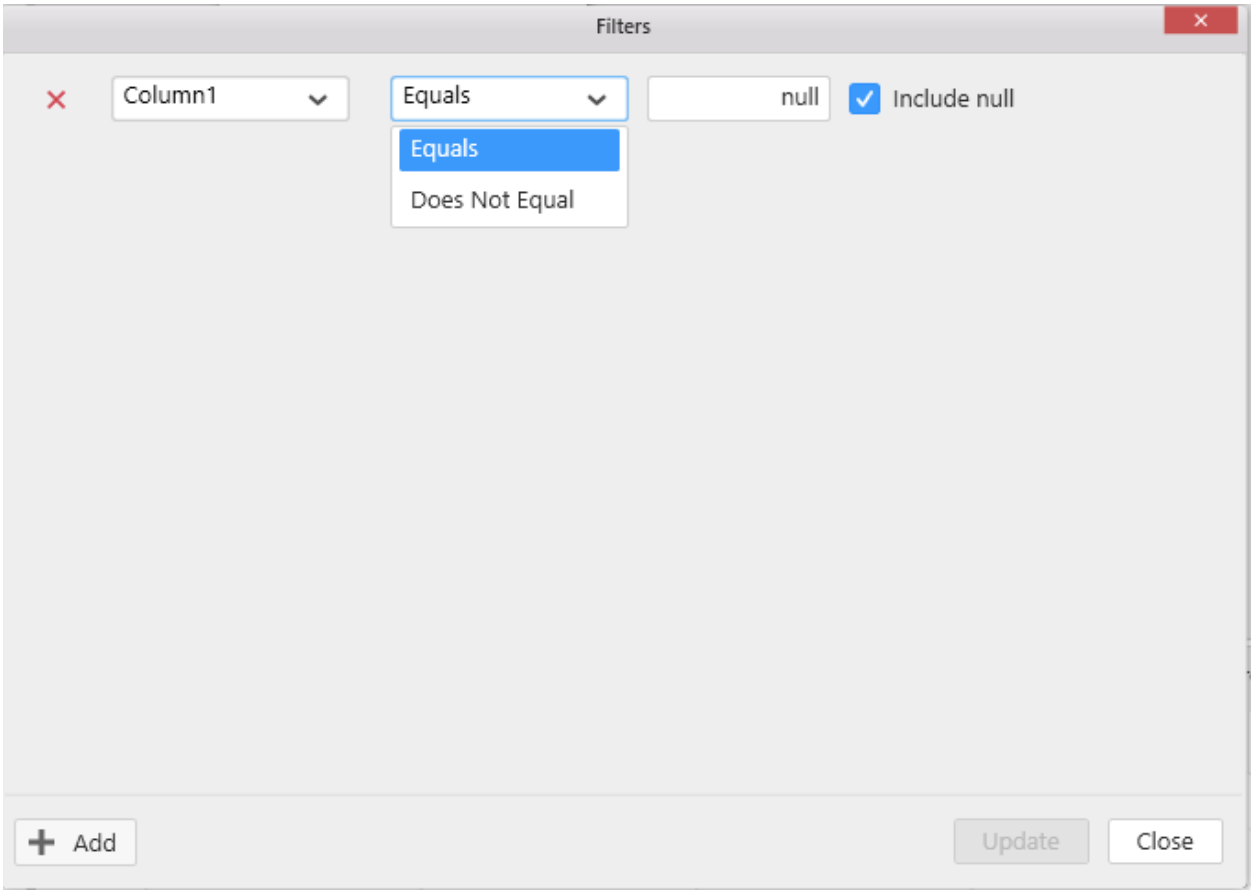
Add a filter condition through clicking the Add button in the Filters window.



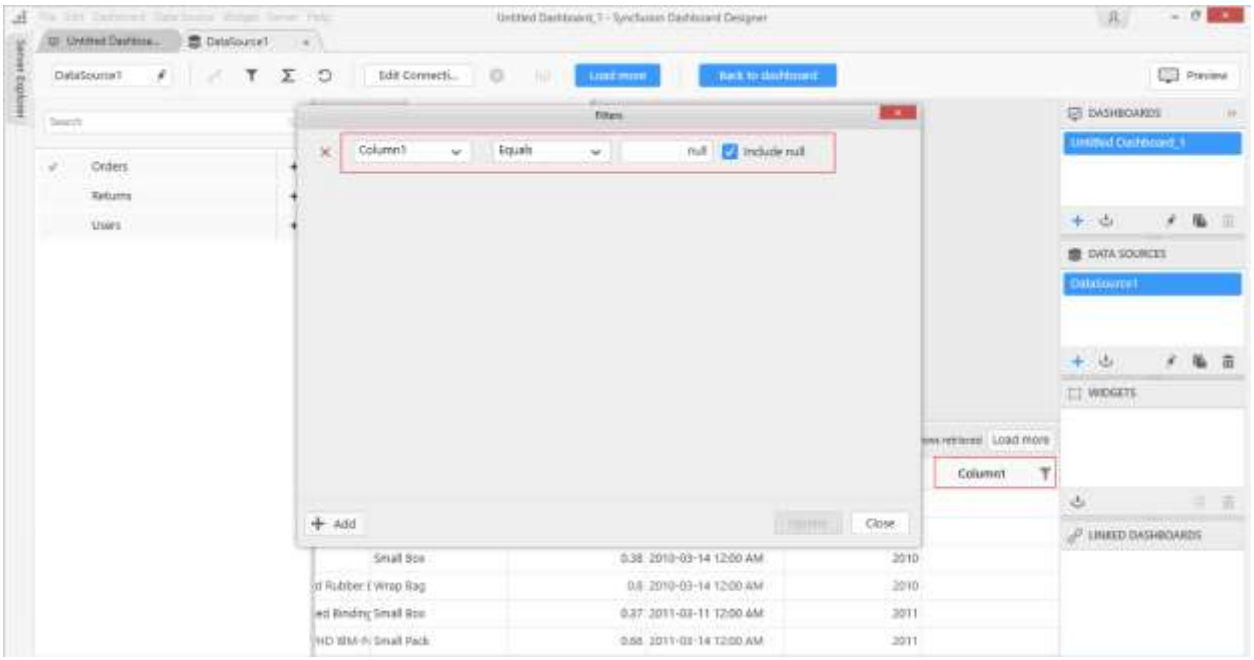
Now, a filter condition will get added by default like below:



You can filter values with or without Null by checking or unchecking the Include null option. If you check this option, only Equals and Does Not Equal condition will be shown like below. Based on these conditions, you may filter values with or without Null.



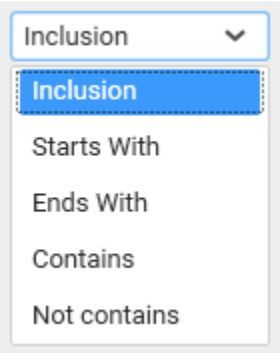
Below example shows column with Null values.



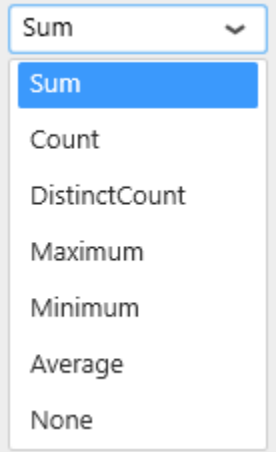
Modify the condition as you require and define criteria. The condition can be defined based on the column. Based on its data type, the parameters to define will differ.

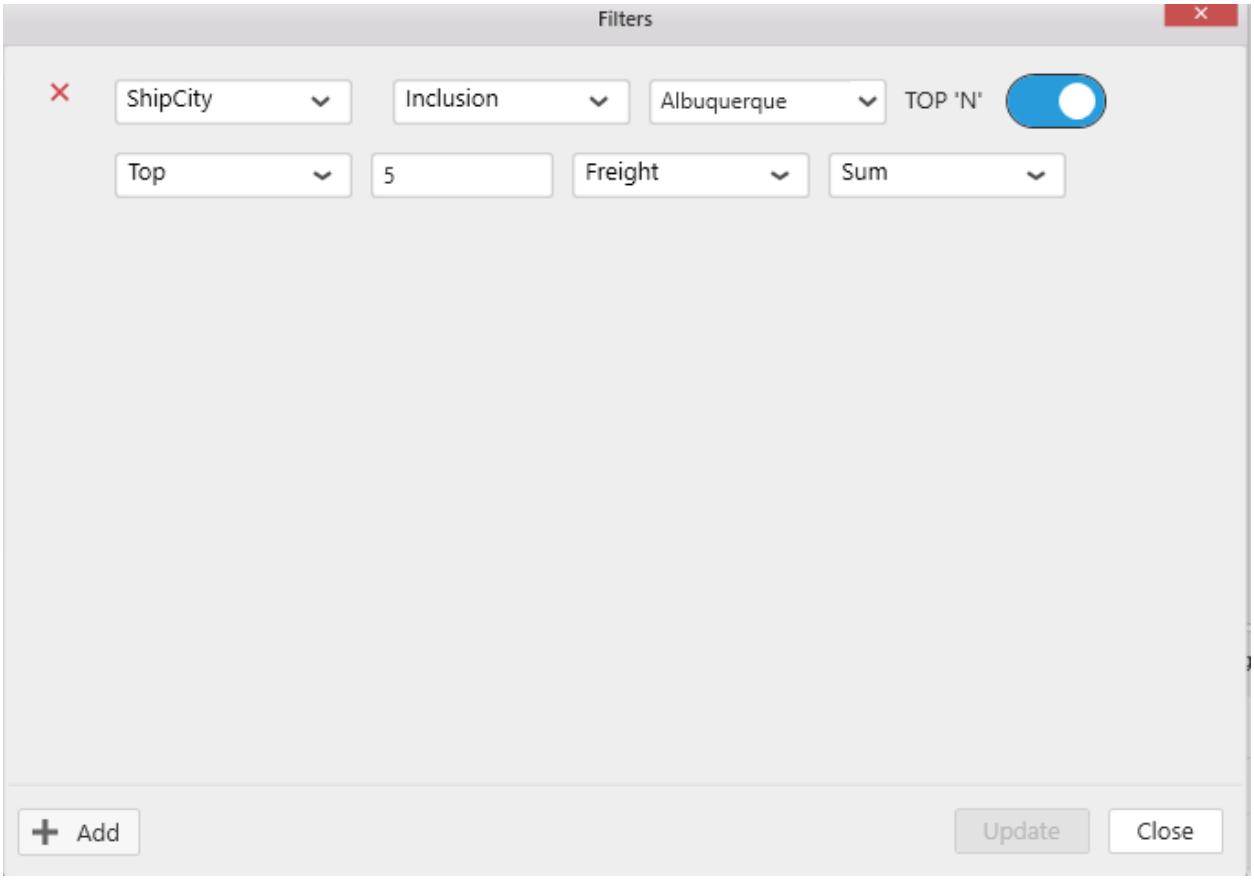
With the date time type or text type column like above, you may get a toggle button TOP ‘N’ at right, to enable the Top N filter to configure the field and the condition by which it has to be applied.

The options below are displayed when selecting text type columns.

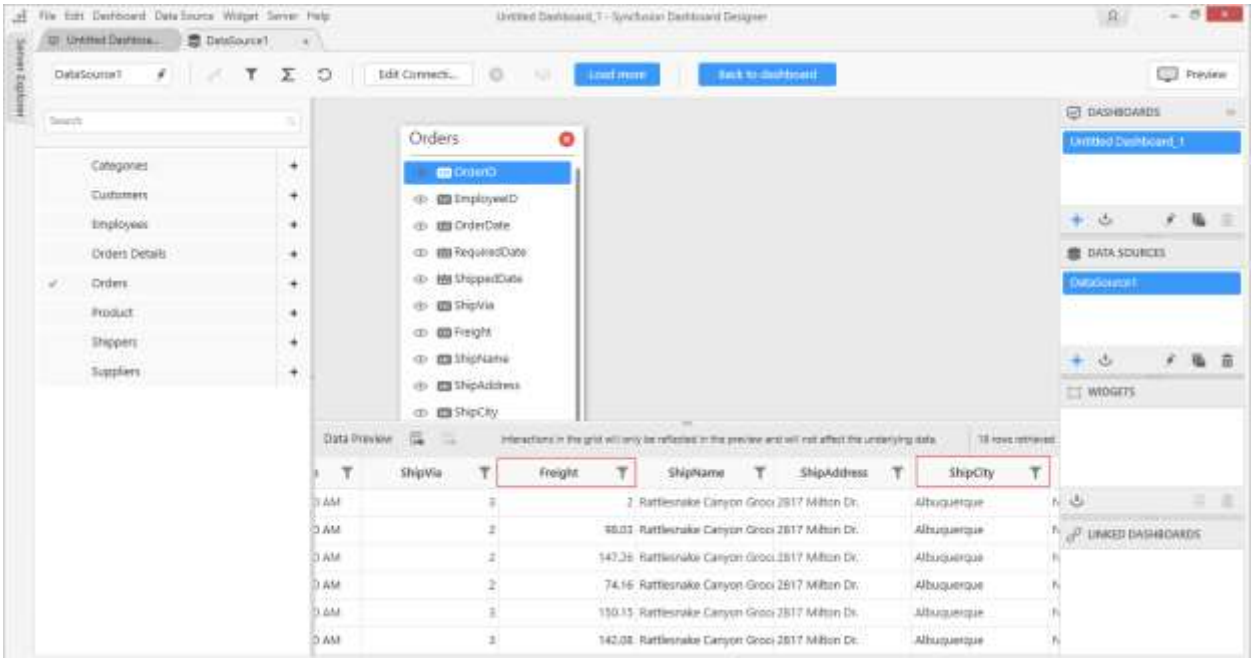


The options below will be displayed when applying the Top 'N' filter based on the numeric column.

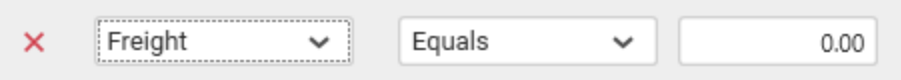


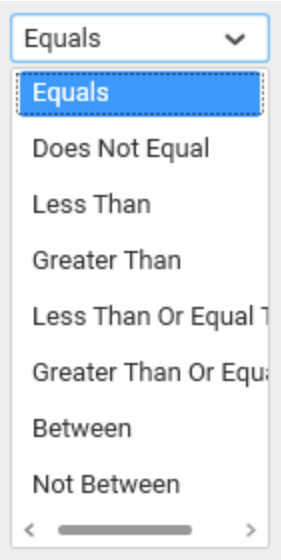


Below example shows top 5 freight data of a city.

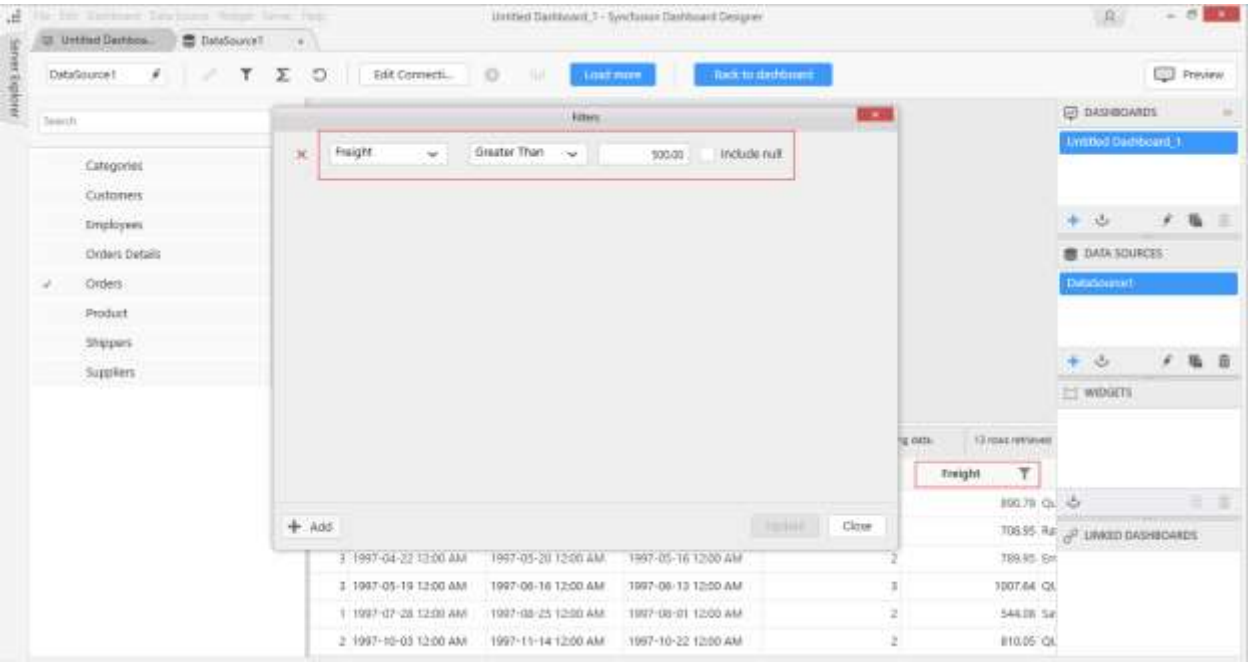


For numeric type column, the parameters will be like below:

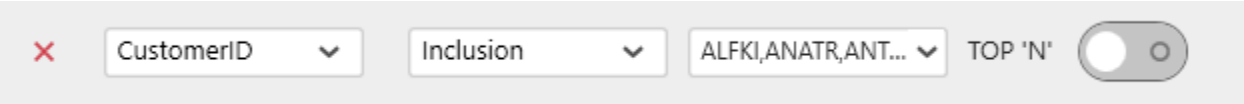




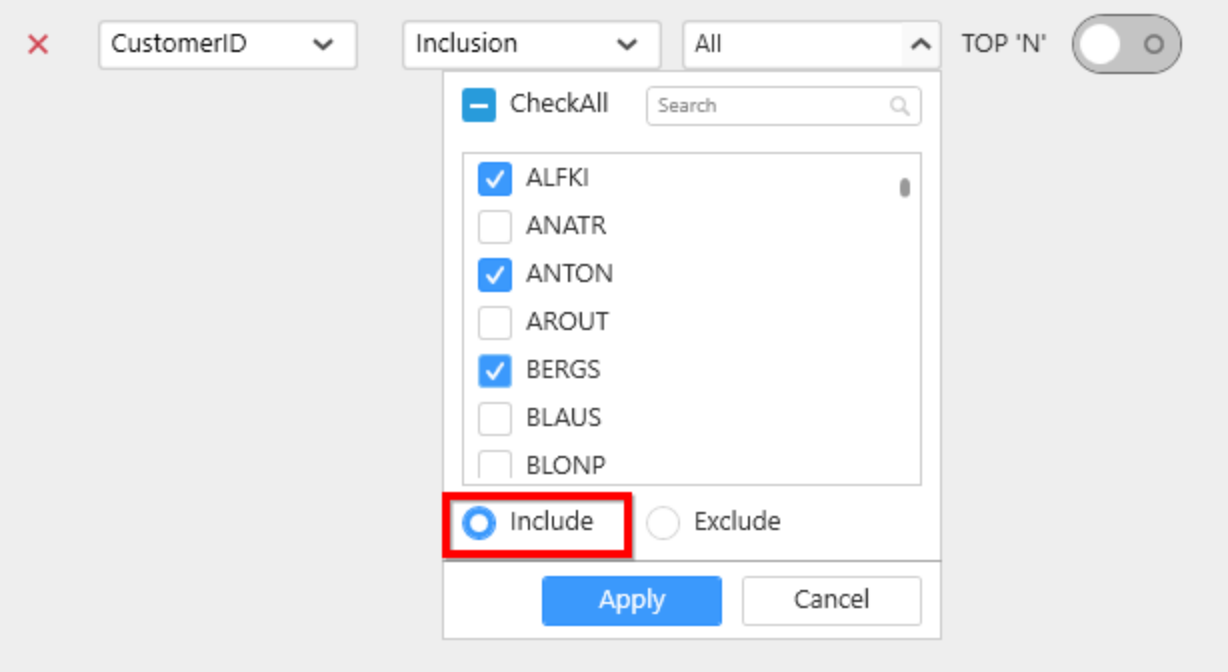
Below example shows data of freight whose values are greater than 500.



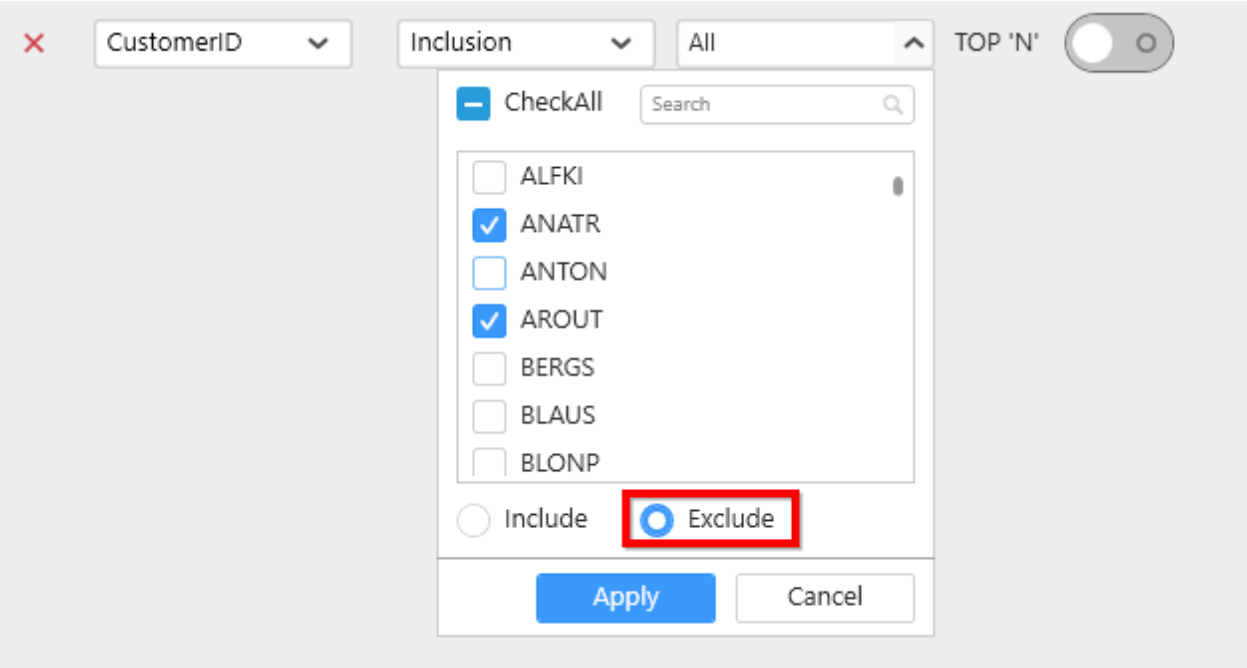
For text type column, the parameters will be like below.



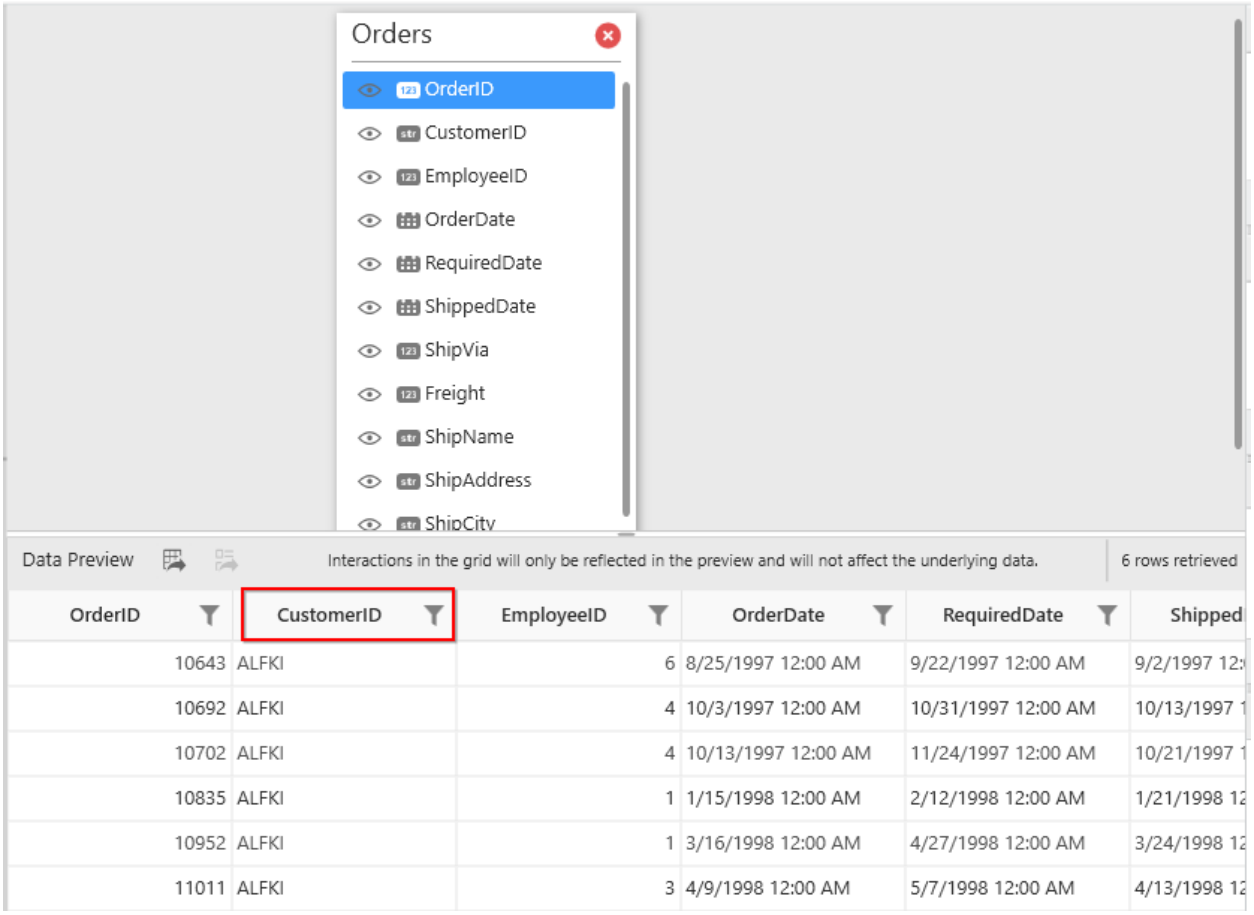
Using the Include option, you can filter particular value(s) from a text column. To do this, uncheck all items, select the required value(s) and choose Include option. By default, Include option will be selected.



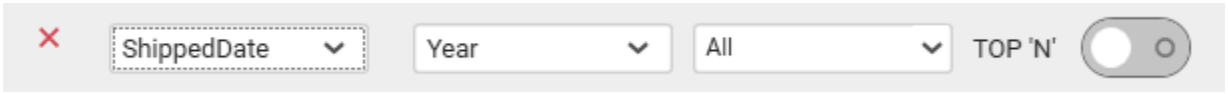
Using the Exclude option, you can filter your text column without specific value(s). To do this, uncheck all items, select your required value(s) and choose Exclude option.

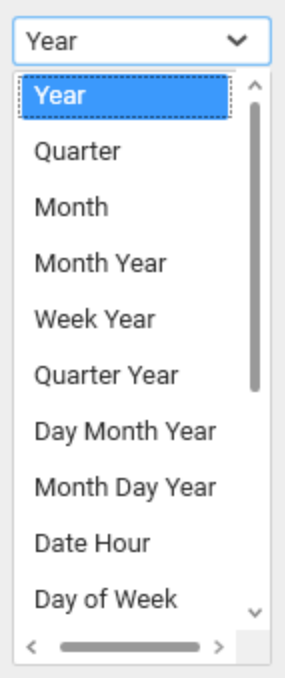


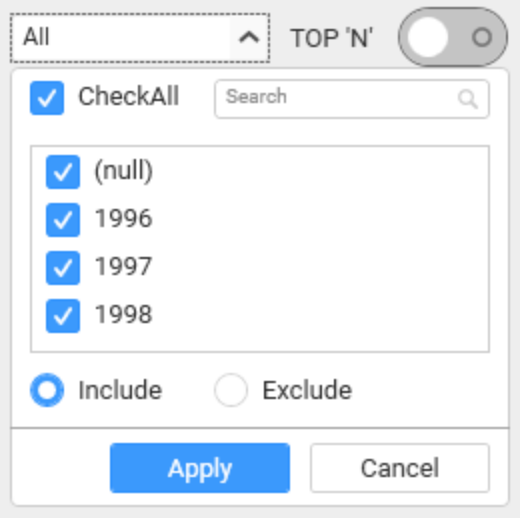
Below example shows data of CustomerID whose values is ALFKI.

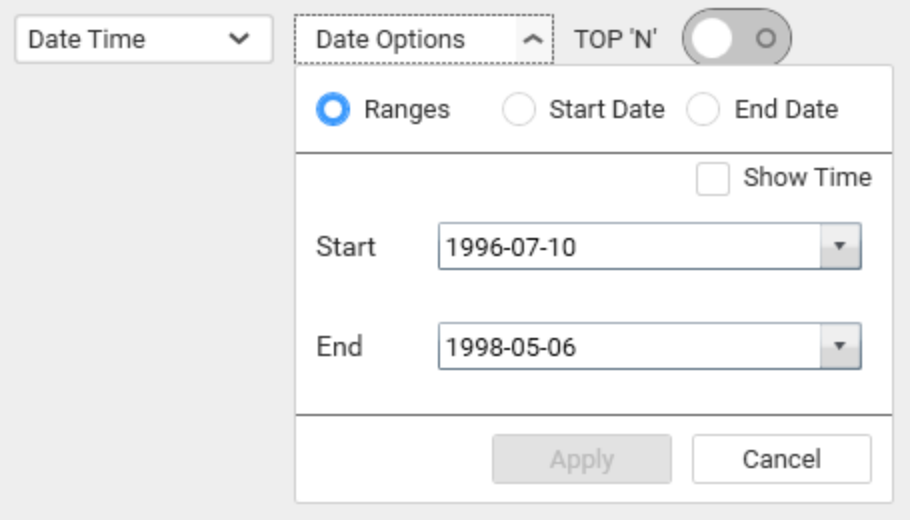


For date time type column, the parameters will be like below:

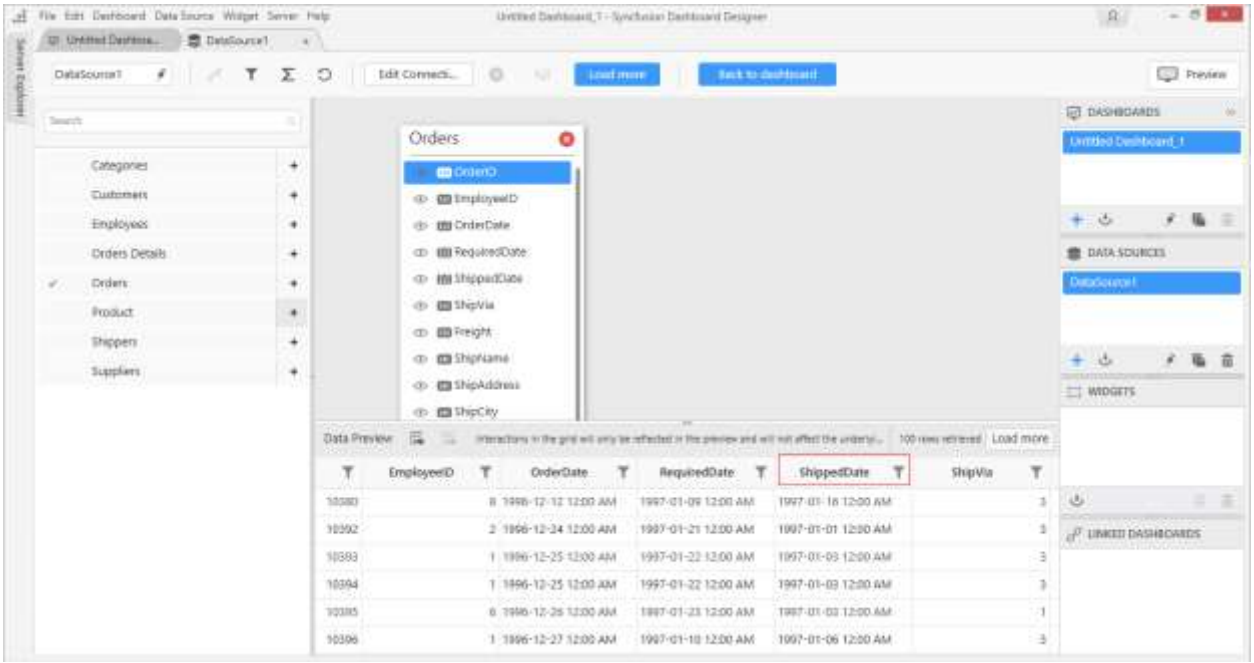




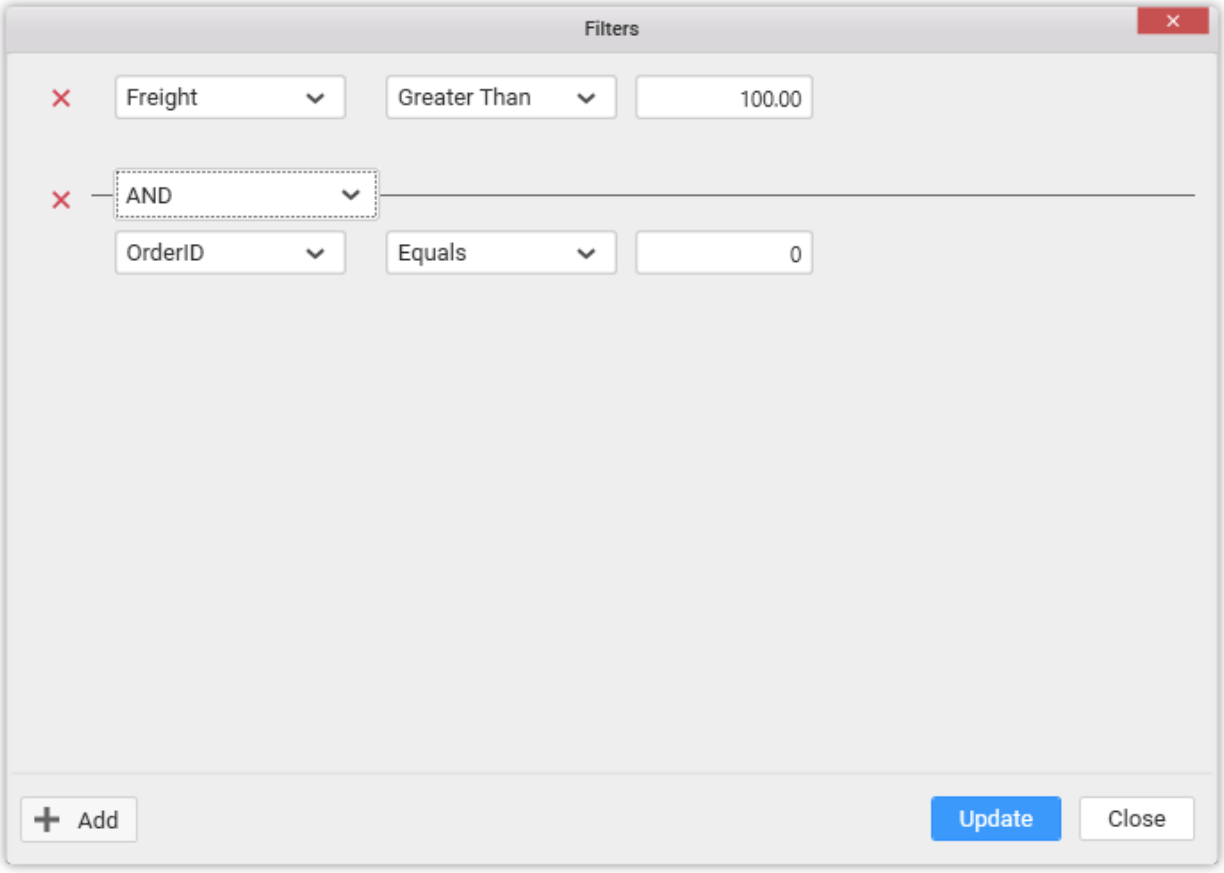




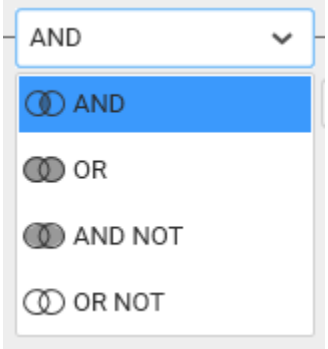
Below example shows data within the applied date range.



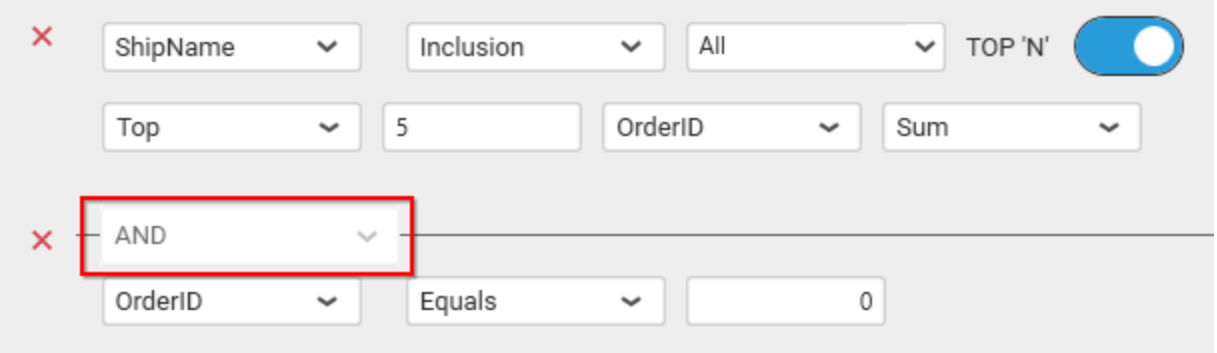
Add more than one condition if you prefer, through clicking the Add button.



By default, AND operation will be handled in between two conditions. However, you may change it to any of the below:



However, this operation cannot be changed, when you have the TOP N option enabled like below:

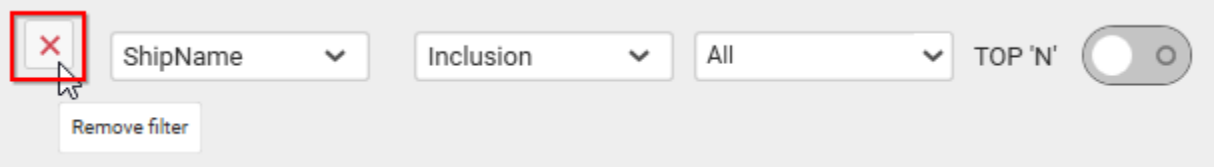


Click Update to save the defined data filter conditions.

Click Close button or the Close icon at top right corner of the window to close the Filters window.

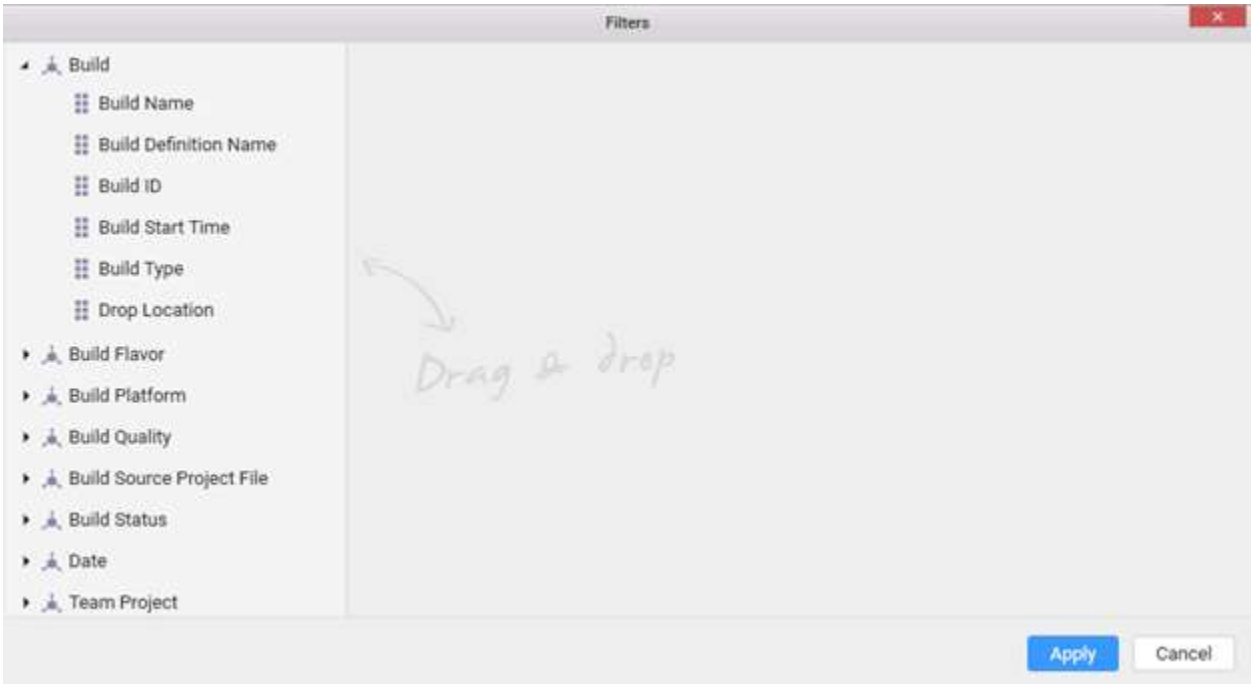
Deleting a filter condition

You can remove a filter condition by clicking the close icon (highlighted below) at left of the respective filter condition.



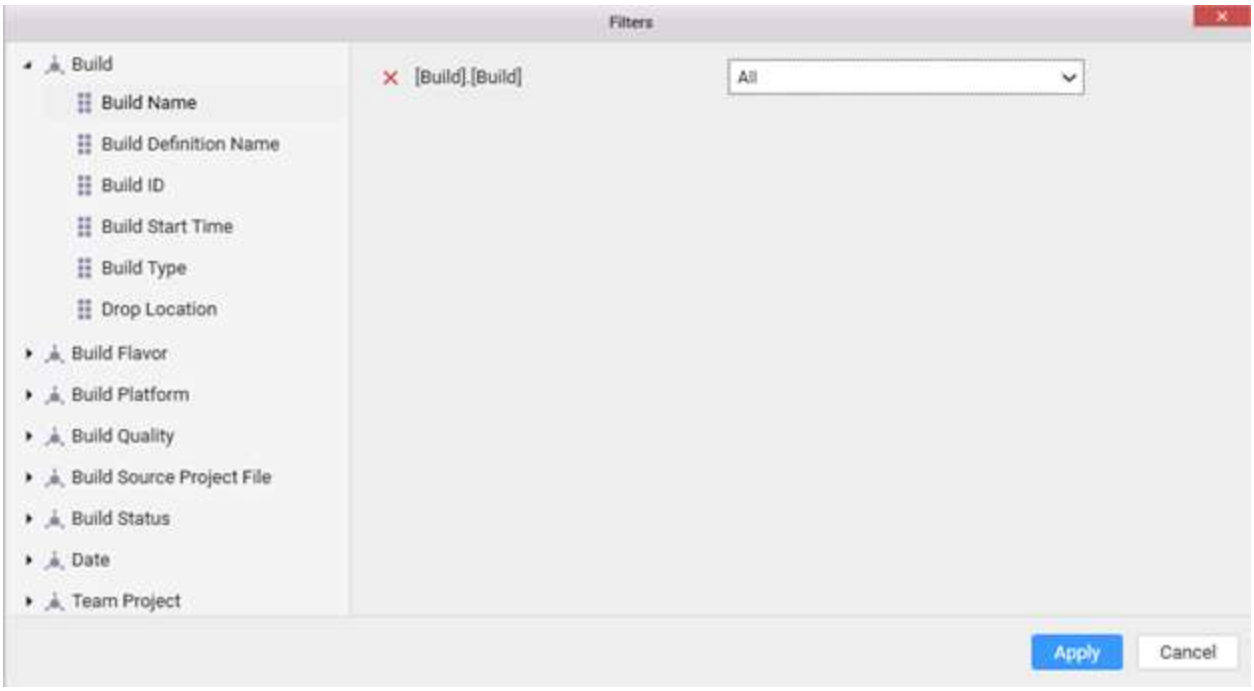
Configuring Data Filters for SSAS Data source

Note: Currently, Data filters for SSAS data source can be defined only through string type columns.

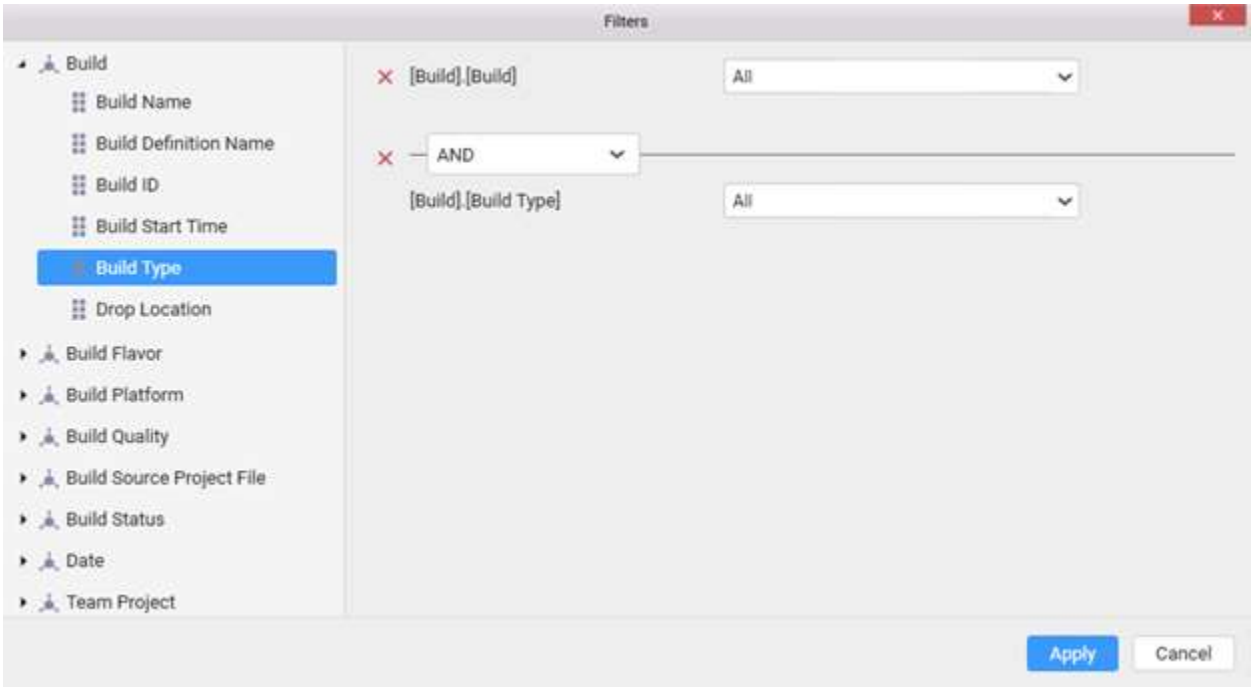


Adding a filter condition

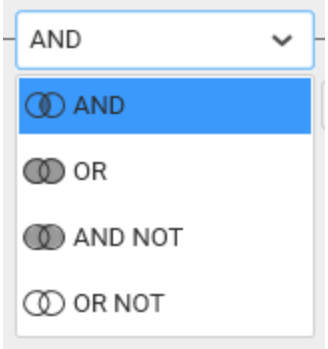
Expand the dimension and drop and drop a hierarchy or level to the right area to add a filter condition.



Add more than one filter condition if you prefer, through drag and drop.

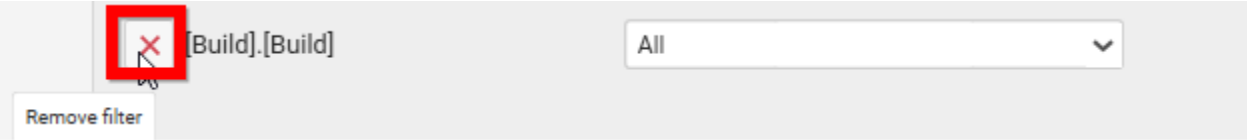


By default, AND operation will be handled in between two conditions as other connection. However, you may change it to any of the below:



Removing a filter condition

You can remove a filter condition by clicking the close icon at left of the respective filter condition.



Configuring Expression Columns

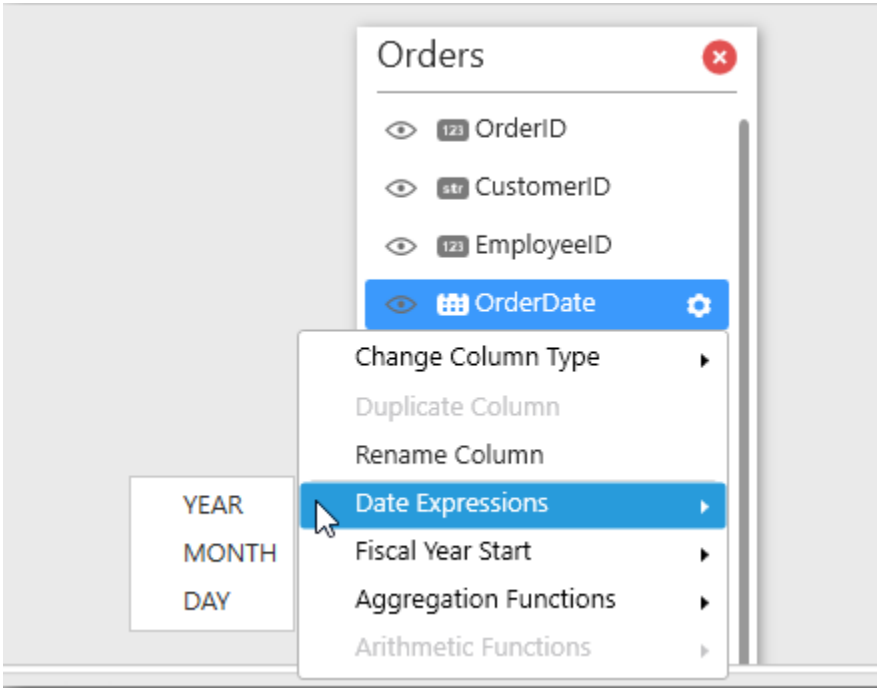
An expression column is used to create an expression which is a combination of data columns, operators and built-in functions. This expression column will act as a calculated measure that can be configured to widget like other normal numeric columns as a quantitative measure.

Adding an Expression Column

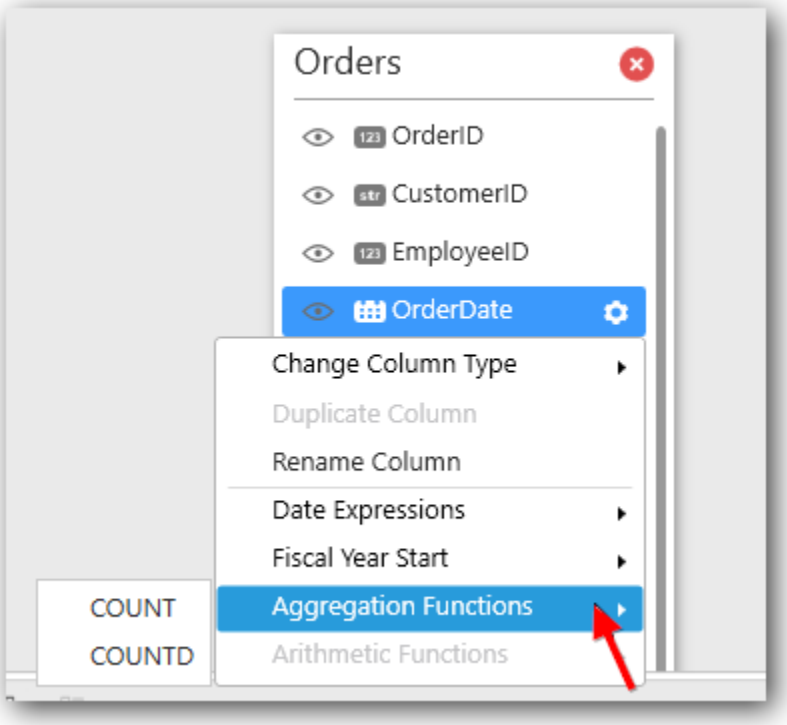
An expression field can be added either through the Settings menu with certain built-in functions based on the respective column or through the expression designer.

Note: Adding expression column through the Settings menu in table design view is not supported for SSAS data source currently.

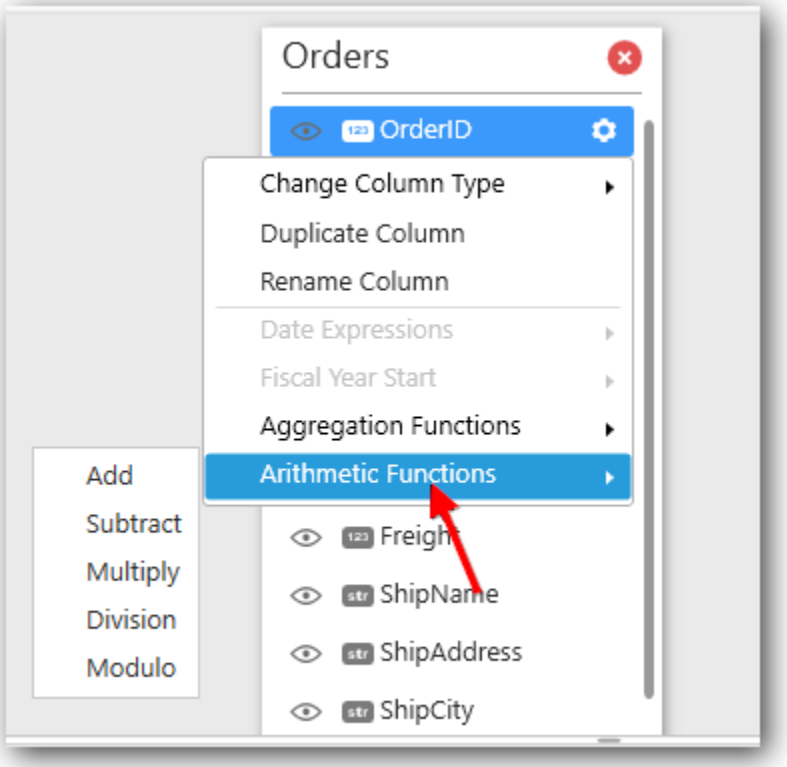
Using Date Expressions



Using Aggregation Functions



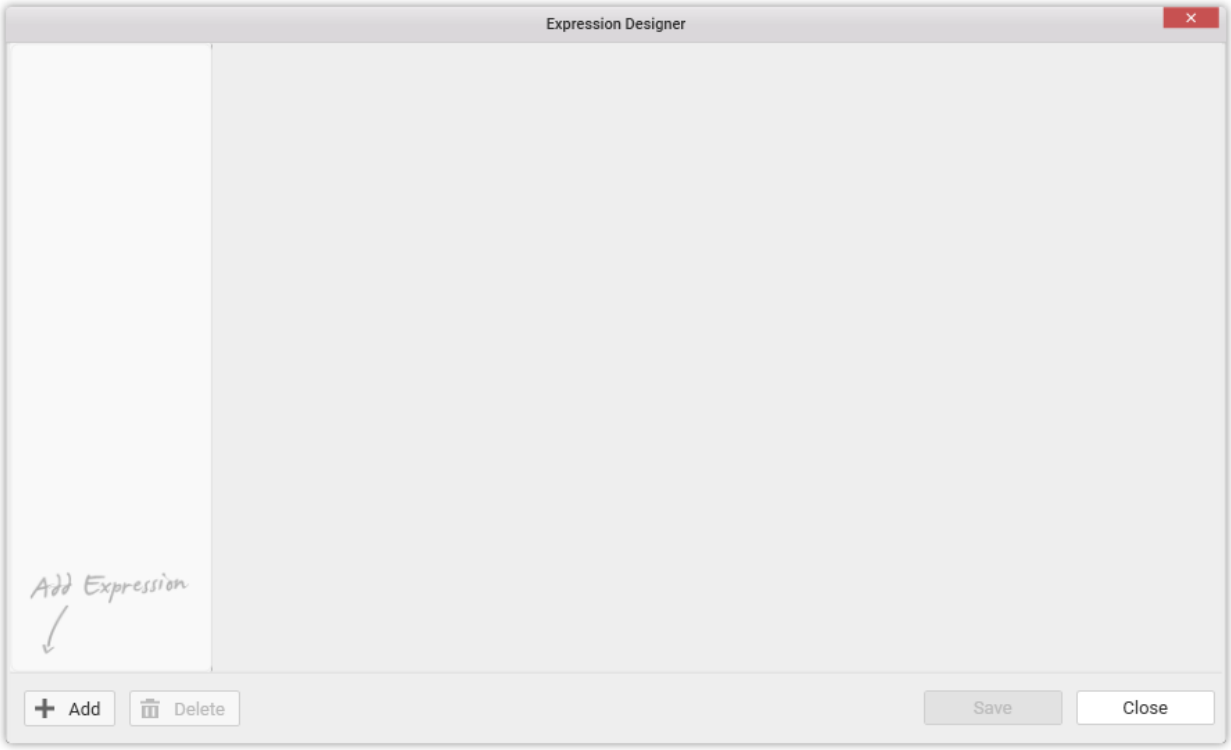
Using Arithmetic Function



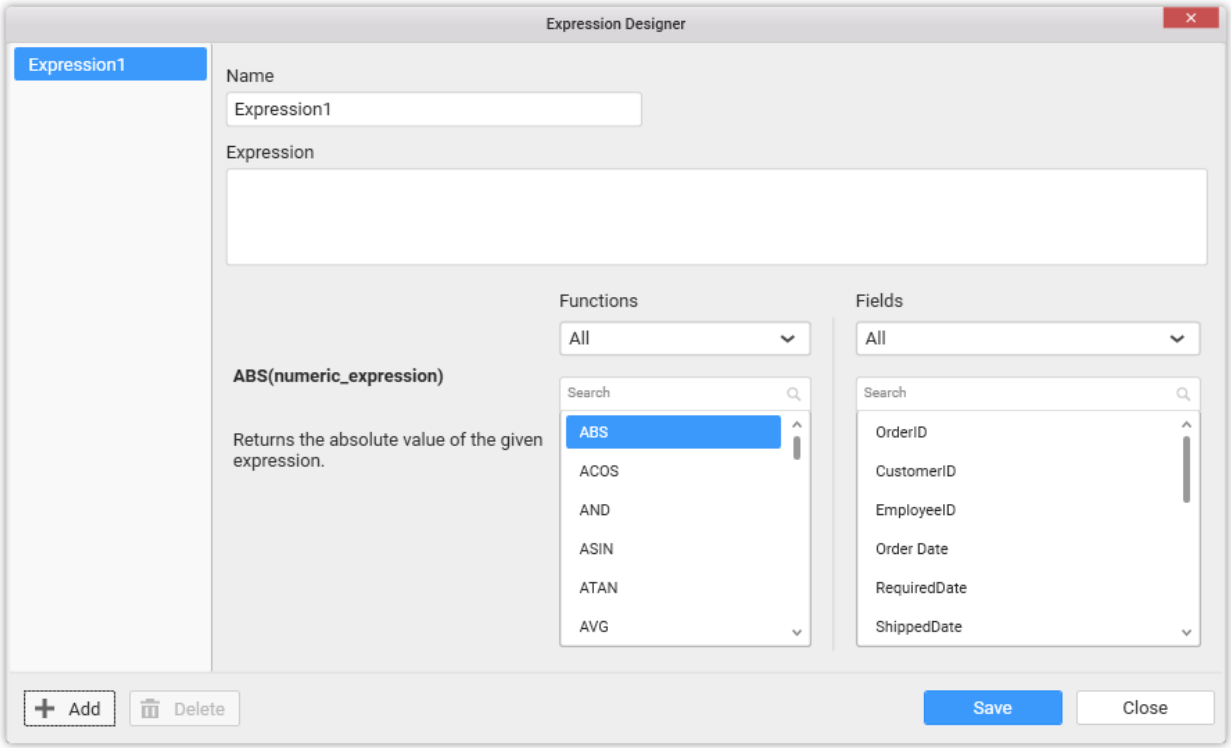
Expression Designer can be launched either through the tools pane in the data design view highlighted below:

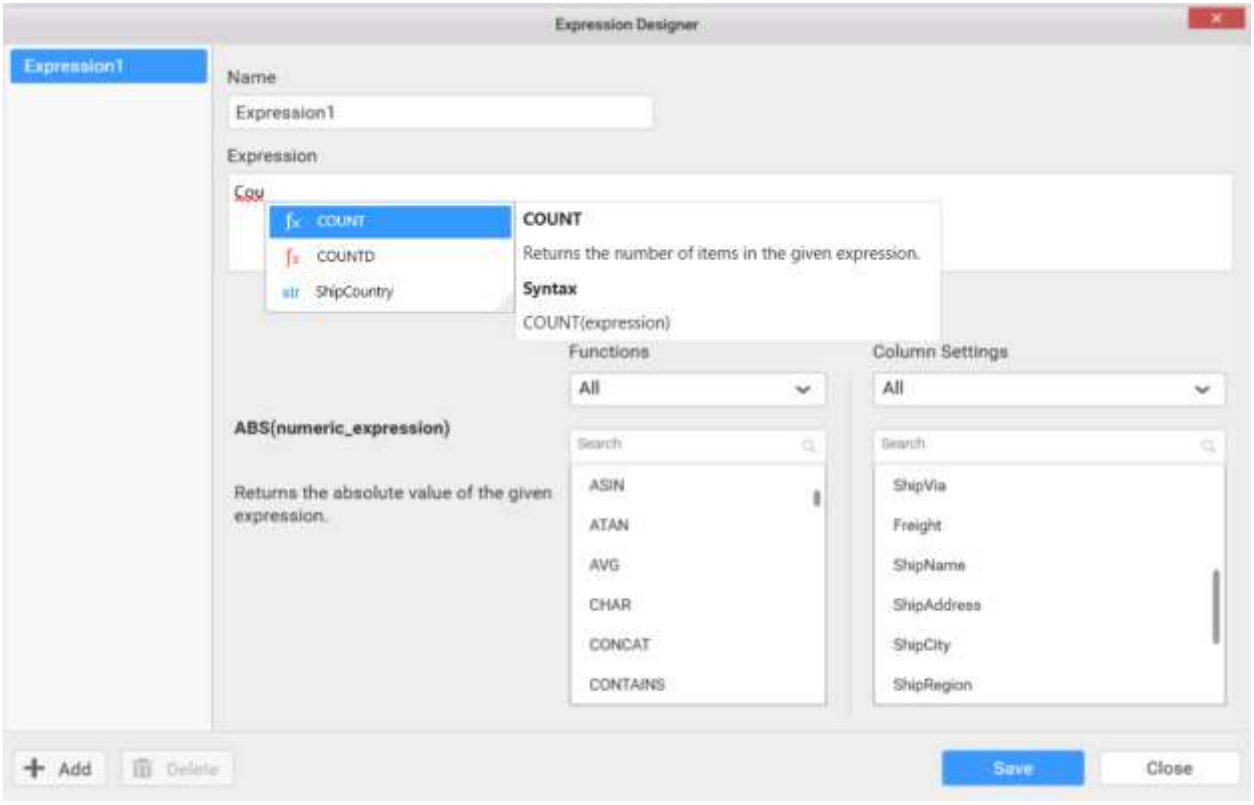
Or through the Expression Columns container in Widget View highlighted below:

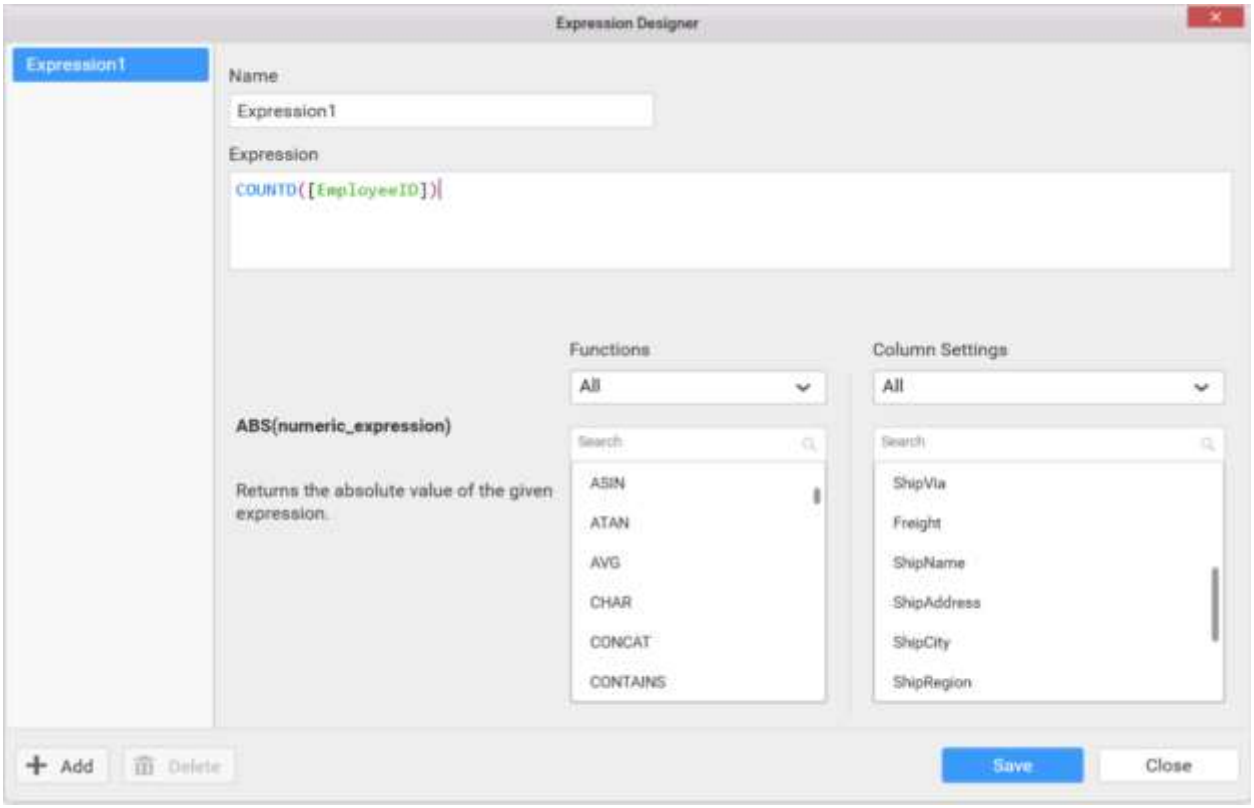
Click Add in the Expression Designer window to add a new expression column.



Enter a suitable name for the expression in the Name text area. By default, it will be Expression1.


Expression Designer
Expression1
Name
Expression1
Expression
Functions
All
Fields
All
ABS(numeric_expression)
Returns the absolute value of the given expression.
Search
ABS
ACOS
AND
ASIN
ATAN
AVG
OrderID
CustomerID
EmployeeID
Order Date
RequiredDate
ShippedDate
Add
Delete
Save
Close


Expression Designer
Expression1
Name
Expression1
Expression
Cou
COUNT
COUNTD
ShipCountry
COUNT
Returns the number of items in the given expression.
Syntax
COUNT(expression)
Functions
All
Column Settings
All
ABS(numeric_expression)
Returns the absolute value of the given expression.
ASIN
ATAN
AVG
CHAR
CONCAT
CONTAINS
ShipVia
Freight
ShipName
ShipAddress
ShipCity
ShipRegion
Add
Delete
Save
Close



Enter the expression that you like to define in the Expression text area. The syntax for defining a simple expression is,

{function name(}[columnname]{operator[columnname]…}

Where, content within curly braces is optional.

Some expressions for reference:

1) YEAR([Order Date]) – To compute year of order date. 2) COUNTD([EmployeeID]) – To compute distinct count of employees. 3) [Freight]+100 – To compute the total with 100 added to Freight.

Note: Currently, SSAS data source supports only functions existing under Numbers category.

Following built-in functions are supported in Expression Designer.

Category	Functions	Syntax & Descriptions
Numbers	ABS	Syntax: ABS(numeric_expression) Description: Returns the absolute value of the given expression. Example: ABS([Freight])
	ACOS	Syntax: ACOS(numeric_expression) Description:

		Returns the inverse cosine (also known as arccosine) of the given numeric expression. Example: ACOS(0.25)
	ASIN	Syntax: ASIN(numeric_expression) Description: Returns the inverse sine (also known as arcsine) of the given numeric expression. Example: ASIN(0.25)
	ATAN	Syntax: ATAN(numeric_expression) Description: Returns the inverse tangent (also known as arctangent) of the given numeric expression. Example: ATAN(0.25)
	COS	Syntax: COS(numeric_expression) Description: Returns the cosine of the angle specified in radians, in the given expression. Example: COS(0.25)
	DEGREES	Syntax: DEGREES(numeric_expression) Description: Returns the angle in degrees for the one specified in radians, in the given numeric expression. Example: DEGREES(1.5708)
	EXP	Syntax: EXP(numeric_expression) Description: Returns the exponential value of the given expression. Example: EXP([UnitsInStock])
	LOG	Syntax: LOG(numeric_expression) Description: Returns the logarithm of the given expression to the specified base.

		Example: LOG(DEGREES(PI()))
	PI	Syntax: PI() Description: Returns the constant value of PI. Example: EXP(PI())
	POWER	Syntax: POWER(expression1, expression2) Description: Returns the value of the given expression (expression1) to the specified power (expression2). Example: POWER(EXP(1),SIN(90))
	ROUND	Syntax: ROUND(numeric_expression) Description: Returns a rounded value. Example: ROUND(PI())
	RADIAN	Syntax: RADIAN(numeric_expression) Description: Returns the angle in radians for the one specified in degrees, in the given numeric expression. Example: RADIANS(90)
	SIGN	Syntax: SIGN(numeric_expression) Description: Returns a value representing the positive (+1), zero (0), or negative (-1) sign of the given numeric expression. Example: SIGN([UnitsOnOrder])
	SIN	Syntax: SIN(numeric_expression) Description: Returns the sine of the angle specified in radians, in the given expression.

		Example: SIN(0.25)
	SQRT	Syntax: SQRT(numeric_expression) Description: Returns the square root of the given numeric expression. Example: SQRT([UnitsInStock])
	TAN	Syntax: TAN(numeric_expression) Description: Returns the tangent of the given numeric expression. Example: TAN(0.25)
Aggregation	AVG	Syntax: AVG(numeric_expression) Description: Returns the average of the values in the given expression. Example: AVG([UnitPrice])
	COUNT	Syntax: COUNT(numeric_expression) Description: Returns the number of items in the given expression. Example: COUNT([OrderID])
	COUNTD	Syntax: COUNTD(numeric_expression) Description: Returns the distinct number of items in the given expression. Example: COUNTD([OrderID])
	MAX	Syntax: MAX(numeric_expression) Description: Returns the maximum value in the given expression.

		Example: MAX([UnitPrice])
	MIN	Syntax: MIN(numeric_expression) Description: Returns the minimum value in the given expression. Example: MIN([UnitPrice])
	STDEV	Syntax: STDEV(numeric_expression) Description: Returns the standard deviation of values in the given expression. Example: STDEV[OrderID]
	SUM	Syntax: SUM(numeric_expression) Description: Returns the sum of values in the given expression. Example: SUM([UnitPrice])
	VAR	Syntax: VAR(numeric_expression) Description: Returns the variance of values in the given expression. Example: VAR([OrderID])
	TOTAL	Syntax: TOTAL(aggregated_expression) Description: Returns the summarized total of values across each table row resulted from the aggregated expression. Example: TOTAL(SUM([UnitPrice]))
Conditional	IF	Syntax: IF(expression, true*part, false*part) Description: Returns either true part or false part, depending on the evaluation of the expression. Example:

		IF([CustomerID]='VINET' AND [OrderID]=10248, [Discount], [Discount]+0.1)
	IFNULL	Syntax: IFNULL(expression1, expression2) Description: Returns expression1 if the expression1 evaluates to be not null. Example: IFNULL([ShipRegion],'Region not specified')
	ISNULL	Syntax: ISNULL(expression) Description: Returns true if the given expression evaluates to null. Example: ISNULL([ShipRegion])
	ISNOTNULL	Syntax: ISNOTNULL(expression) Description: Returns true if the given expression evaluates to be not null. Example: ISNOTNULL([ShipRegion])
Logical	AND	Syntax: (expression1) AND (expression2) Description: Returns true if both the expressions evaluates to true. Example: IF([CustomerID]='VINET' AND [OrderID]=10248, [Discount], [Discount]+0.1)
	NOT	Syntax: NOT(expression) Description: Returns the reversed logical value of the expression being evaluated. Example: IF(NOT [CustomerID]='VINET', [Freight], [Freight] - 100)
	OR	Syntax: (expression1) OR (expression2) Description: Returns true if any of the expressions evaluates to true.

		Example: IF([CustomerID]='VINET' OR [OrderID]=10248, [Discount], [Discount]+0.1)
Date	DATEADD	Syntax: DATEADD(numeric*expression, date*expression) Description: Adds the number of days to the specified date. Example: DATEADD(7,[OrderDate])
	DATESUB	Syntax: DATESUB(numeric*expression, date*expression) Description: Returns the date subtracted from the specified date. Example: DATESUB(7,[OrderDate])
	DAY	Syntax: DAY(date_expression) Description: Returns a numeric value representing the day part of the specified date. Example: DAY([OrderDate])
	DAYDIFF	Syntax: DAYDIFF(start*date*expression, end*date*expression) Description: Returns a numeric value representing the difference between two specified dates. Example: DAYDIFF(MAX([OrderDate]),'1998-08-08')
	HOUR	Syntax: HOUR(date_expression) Description: Returns the hour of the given date as an integer. Example: HOUR([OrderDate])
	MINUTE	Syntax: MINUTE(date_expression) Description: Returns a numeric value representing the minute part of the date resulted from specified date expression.

		Example: MINUTE([OrderDate])
	MONTH	Syntax: MONTH(date_expression) Description: Returns a numeric value representing the month part of the date resulted from specified date expression. Example: MONTH([OrderDate])
	NOW	Syntax: NOW() Description: Returns the current date and time. Example: DATEPART("year",NOW())
	TODAY	Syntax: TODAY() Description: Returns the current date. Example: DATEPART("month",TODAY())
	YEAR	Syntax: YEAR(date_expression) Description: Returns a numeric value representing the year part of the date resulting from the specified date expression. Example: YEAR([OrderDate])
	DATENAME	Syntax: DATENAME(date*part, date*expression) Description: Returns a string representing the specified date_part of the given date expression. Example: DATENAME("day",[OrderDate])
	DATEPART	Syntax: DATEPART(date*part, date*expression) Description: Returns an integer value representing the specified date_part of the given date expression.

		Example: DATEPART("year",MAX([OrderDate]))
	MAX	Syntax: MAX(expression) Description: Returns the maximum value in the given expression. Example: MAX([OrderDate])
	MIN	Syntax: MIN(expression) Description: Returns the minimum value in the given expression. Example: MIN([OrderDate])
String	LEN	Syntax: LEN(string_expression) Description: Returns the number of characters in the given string expression. Example: LEN([ShipPostalCode])
	CHAR	Syntax: CHAR(integer_expression) Description: Converts the given integer ASCII code into a character. Example: CHAR(70)
	CONCAT	Syntax: CONCAT(string*expression1, string*expression2,â€¦, string_expressionN) Description: Returns a string value resulting from the concatenation of two or more string values. Example: CONCAT('http://en.wikipedia.org/wiki/',[ShipCity])
	CONTAINS	Syntax: CONTAINS(string*expression, substring*expression) Description: Returns true if the given string expression contains the specified substring expression.

		Example: CONTAINS([Shipping Address], [ShipCountry])
	ENDSWITH	Syntax: ENDSWITH(string*expression substring*expression) Description: Returns true if the given string expression ends with the specified substring expression. Example: ENDSWITH([CustomerID],'A')
	LEFT	Syntax: LEFT(string*expression, numeric*expression) Description: Returns the specified number of characters from start of the given string expression. Example: LEFT([ShipAddress],6)
	LOWER	Syntax: LOWER(string_expression) Description: Returns a lower case converted string value from a given string expression. Example: LOWER([ShipCountry])
	LTRIM	Syntax: LTRIM(string_expression) Description: Returns the string value with any leading blanks removed from string expression. Example: LTRIM(' Removes trailing spaces.')
	MAX	Syntax: MAX(expression) Description: Returns the maximum value in the given expression. Example: MAX([ProductName])
	MIN	Syntax: MIN(expression) Description: Returns the minimum value in the given expression.

		Example: MIN([ProductName])
	RIGHT	Syntax: RIGHT(string*expression, numeric*expression) Description: Returns the specified number of characters from end of the given string expression. Example: RIGHT([ProductName],6)
	RTRIM	Syntax: RTRIM(string_expression) Description: Returns the string value with any trailing blanks removed from string expression. Example: RTRIM('Removes trailing spaces. ')
	STARTSWITH	Syntax: STARTSWITH(string*expression, substring*expression) Description: Returns true if the given string expression starts with the specified substring expression. Example: STARTSWITH([CustomerID],'A')
	SUBSTR	Syntax: SUBSTR(string*expression, starting*index, length*of*the_string) Description: Returns a specific length of string starting from specific index from the given string expression. Example: SUBSTR([CustomerID],1,3)
	UPPER	Syntax: UPPER(string_expression) Description: Returns an upper case converted string value from a given string expression. Example: UPPER([ShipCountry])
Filter	CURRENTUSER	Syntax: CURRENTUSER()=[column_name] Description: Returns the current user name. This function uses Dashboard Server user name when the user is

		signed in, else it returns null. Example: CURRENTUSER()=[CustomerID]
	FULLNAME	Syntax: FULLNAME()=[column_name] Description: Returns the current user's Full name. This function uses Dashboard Server user's full name when the user is signed in, else it returns null. Example: FULLNAME()=[CustomerID]
	EMAIL	Syntax: EMAIL()=[column_name] Description: Returns the current user's email. This function uses Dashboard Server user's Email id when the user is signed in, else it returns null. Example: EMAIL()=[CustomerID]
Row	INDEX	Syntax: INDEX() Description: Returns index value of each row. Example: INDEX()
	RUNNINGAVG	Syntax: RUNNINGAVG(aggregated_expression) Description: Returns Running Average of each row. Example: RUNNINGAVG(MAX([UnitsInStock]))
	RUNNINGCOUNT	Syntax: RUNNINGCOUNT(aggregated_expression) Description: Returns Running Count of each row. Example: RUNNINGCOUNT(MAX([OrderID]))
	RUNNINGMAX	Syntax: RUNNINGMAX(aggregated_expression) Description: Returns Running Maximum value of each row. Example: RUNNINGMAX(SUM([UnitsInStock]))

	RUNNINGMIN	Syntax: RUNNINGMIN(aggregated_expression) Description: Returns Running Minimum value of each row. Example: RUNNINGMIN(SUM([UnitsInStock]))
	RUNNINGSTDEV	Syntax: RUNNINGSTDEV(aggregated_expression) Description: Returns Running Stdev value of each row. Example: RUNNINGSTDEV(MAX([UnitsInStock]))
	RUNNINGSUM	Syntax: RUNNINGSUM(aggregated_expression) Description: Returns Running Sum of each row. Example: RUNNINGSUM(MAX([UnitsInStock]))
	RUNNINGVAR	Syntax: RUNNINGVAR(aggregated_expression) Description: Returns Running Variance value of each row. Example: RUNNINGVAR(MAX([UnitsInStock]))

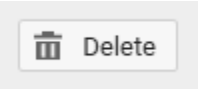
You may also include the function names and the column names just by placing the cursor in respective position in the Expression text area and double-clicking the specific name in respective lists.

Once framing an expression, click Save in Expression Designer window.

Deleting an Expression Column

Select an expression column in left pane.

Click Delete to remove the selected expression column.



Updating an Expression Column

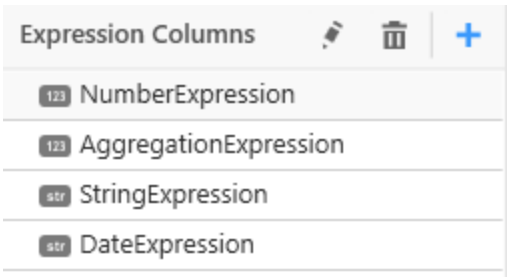
Select an expression column in left pane that you need to update.

Edit the Name and Expression text areas, if required.

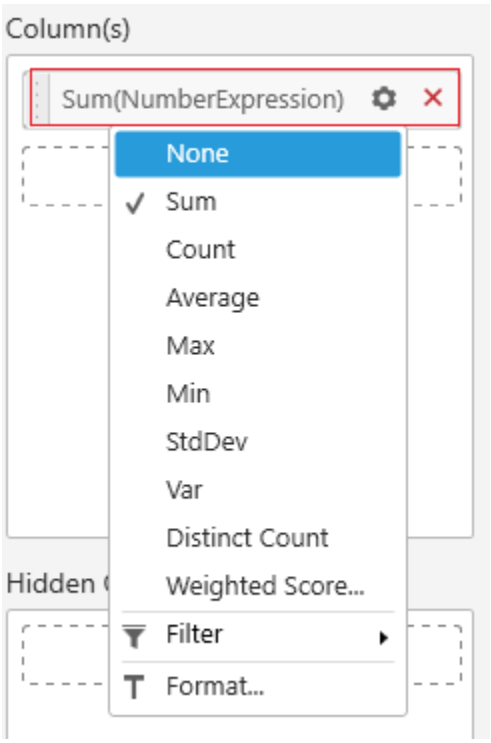
Click Save in Expression Designer window to save the modifications handled.

Configuring Expression Column in Widgets

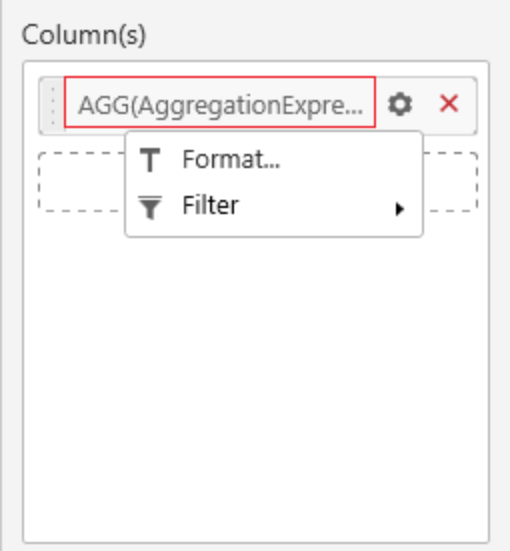
Saved expression will be shown in Expression Columns section of Data Configuration pane like below.



You can also drag and drop expression column into widgets just like measure and dimension fields. For numeric expressions, you can apply any aggregation function like below.

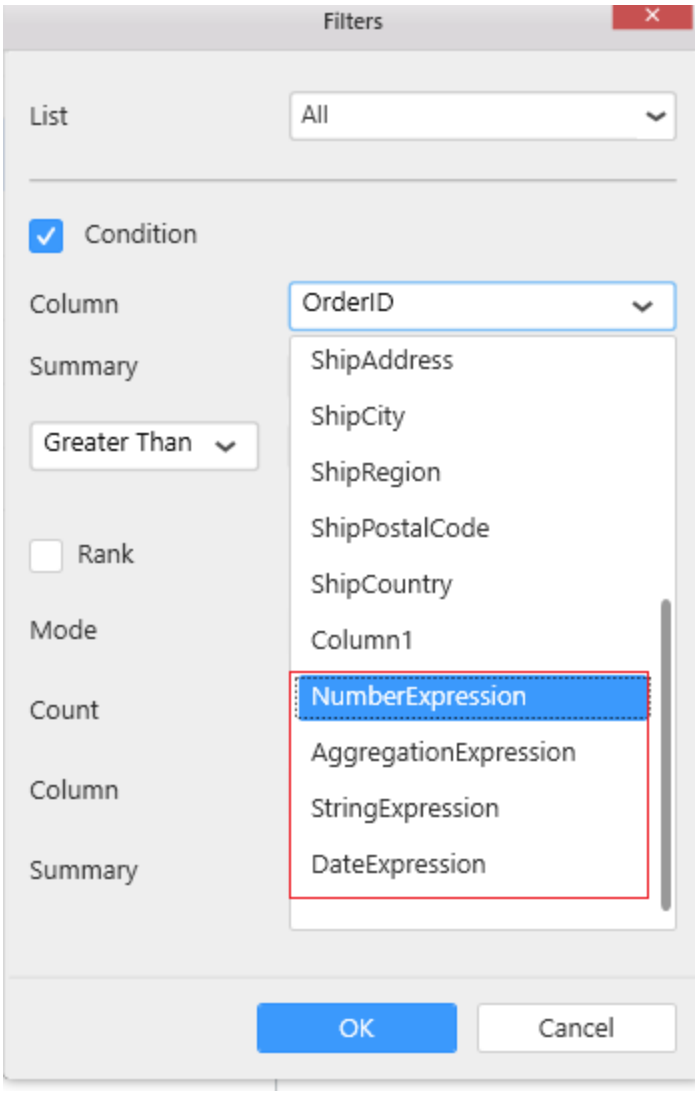


For aggregation expressions alone, AGG keyword will be automatically added in front of the expression name to represent that the created expression is of aggregation type. You can't change the aggregation type in this case.



You can also apply filters for expression column which is used in widget. For aggregation and numeric expressions, you can apply filter just like a measure filter. For string and date expressions, you can apply filter just like a dimension filter.

Expression column will also be shown in Columns section of condition and rank filter.



Note: Expression Designer allows you to create an expression using the functions which are supported by the corresponding database server. Since few functions will have compatibility with specific database server, it is not listed inExpression Designer. For example, CONVERT function is supported in SQL server and MySQL server but not in SQLite. Hence you can create an expression using CONVERT function with appropriate syntax supported by SQL and MySQL server.

What is Bin

Creating Bin is required when you are going to use discrete measure values as dimension. Using **Create Bin** Support, you can convert measure column as dimension column in the Syncfusion Dashboard Designer.

Consider an example table, Employee Table contains Employee*ID, Employee*Name, Age, and Salary columns. You can create a view by binding the age in the column and salary in the value. You can create a numeric dimension (discrete values) for age measure column using the following table.

| Employee*ID* | *Employee*Name | Age | Salary |

|---|---|---|---|

E1	James	23	15000
E2	John	26	17000.55
E3	Robert	30	18500
E4	Michael	33	20000
E5	William	37	250000
E6	David	41	240000

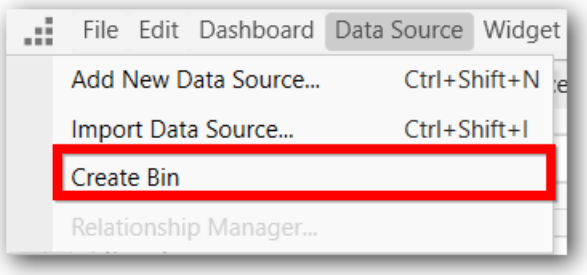
Consider the age field as set of bins i.e, you can group the measure values based on defined bin size. You can get the total salary between the defined age range.

How to create a bin

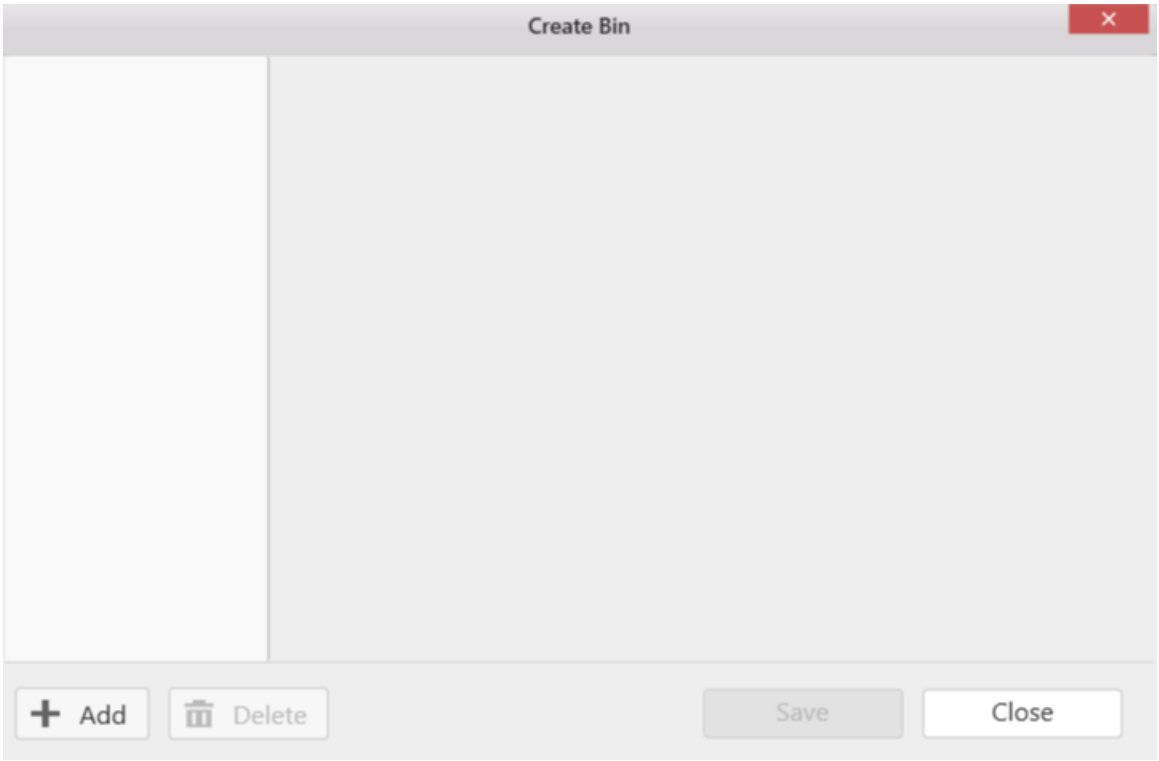
You can create a numeric dimension using the **Create Bin option** in the **Data Source** menu.

Note: The **Create Bin** option will be enabled from the data source tab.

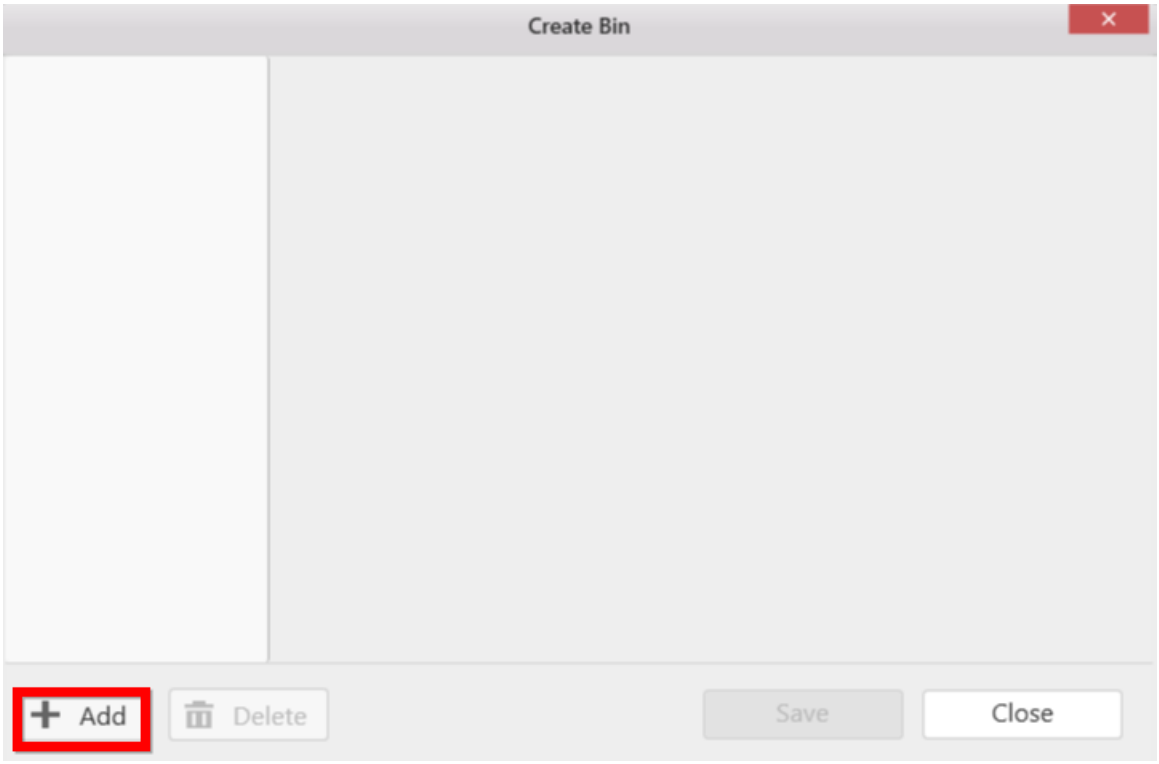
To create bin, click **Create Bin** option in the **Data Source** menu.



The **Create Bin** window will be opened as follows.

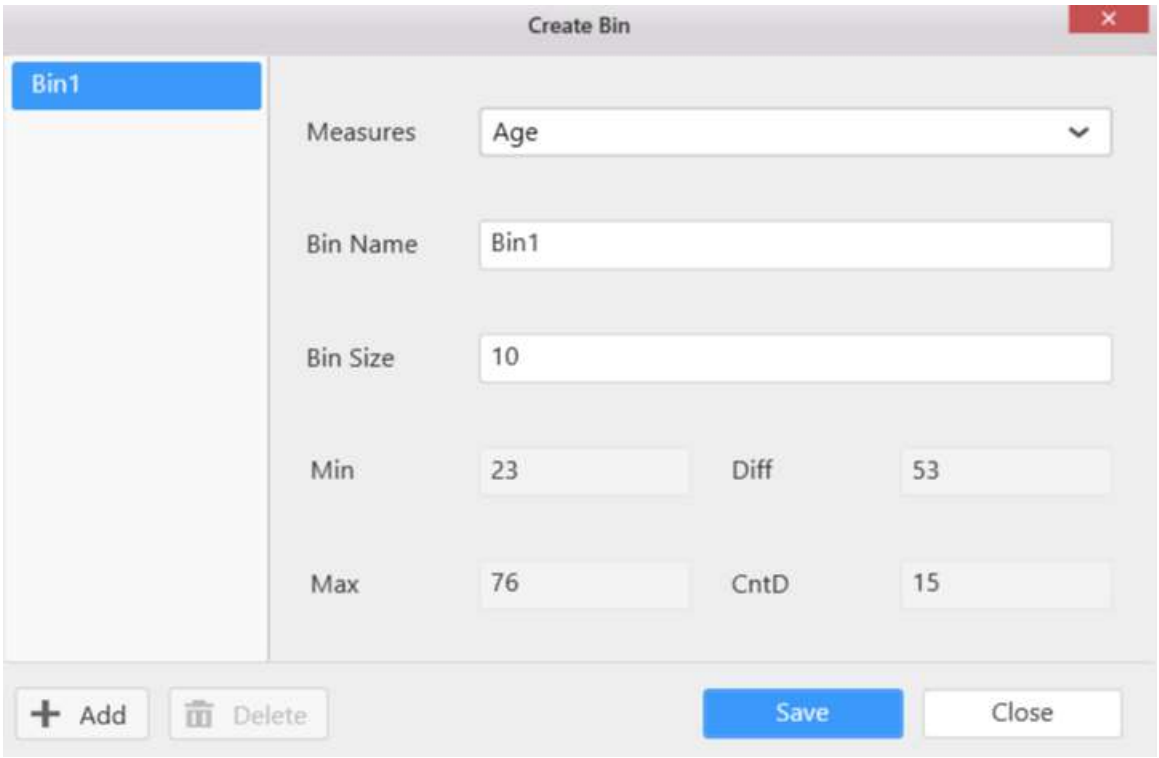


Click **Add** to create a bin.



You can select the measures column listed in the **Measures** drop-down list.

You can edit the Bin Name and Bin Size as follows.



Click **Save** to create a bin.

Click the **Close** button or Close icon to close the **Create Bin** window.

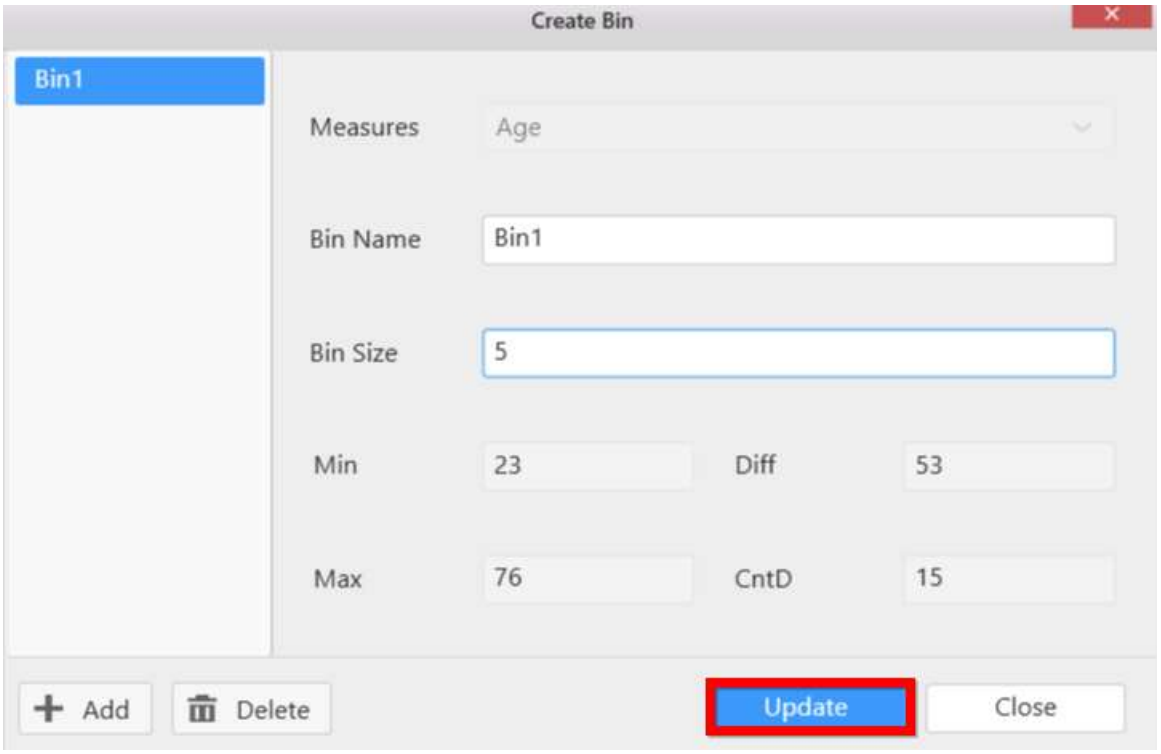
The bin column will be added to the data preview grid as follows.

Employee_Id	Employee_Name	Age	Salary	Bin1
E1	James	23	15000	20
E2	John	26	17000.55	20
E3	Robert	30	18500	30
E4	Michael	33	20000	30
E5	William	37	25000.55	30
E6	David	41	24000	40

Editing a created bin

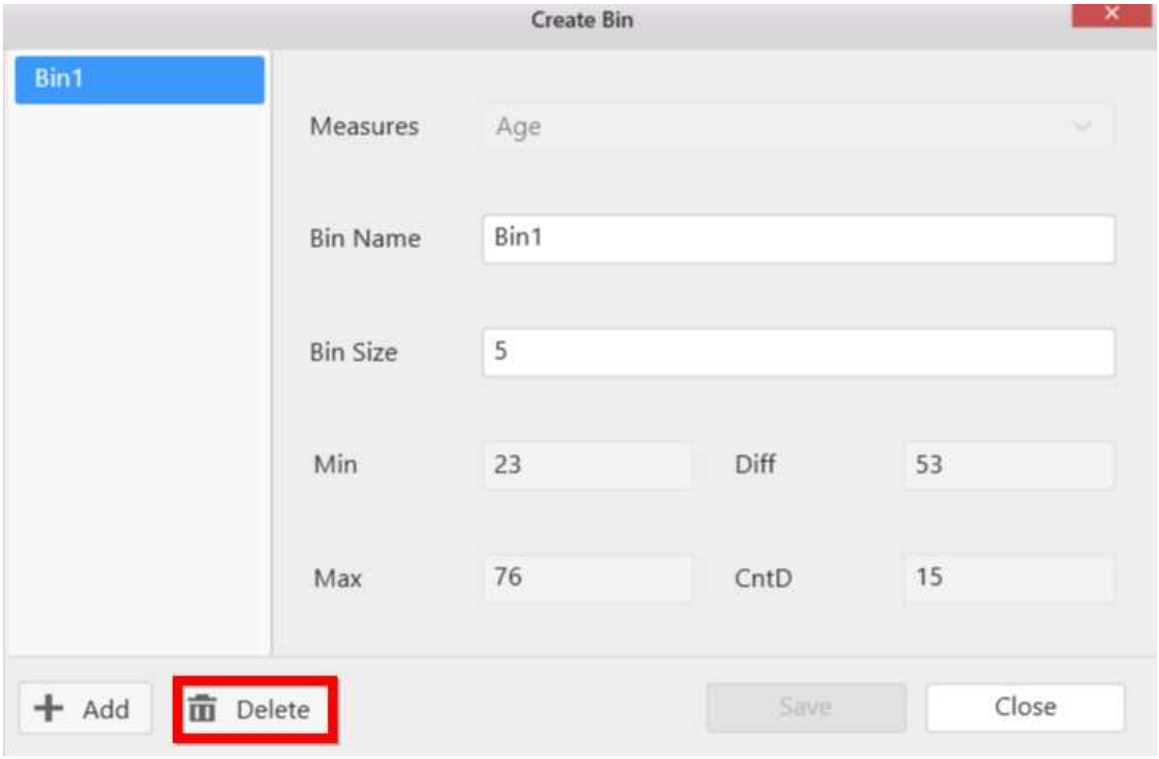
You can edit the Bin Name and Bin Size for the bin.

Click **Update** to save the changes.



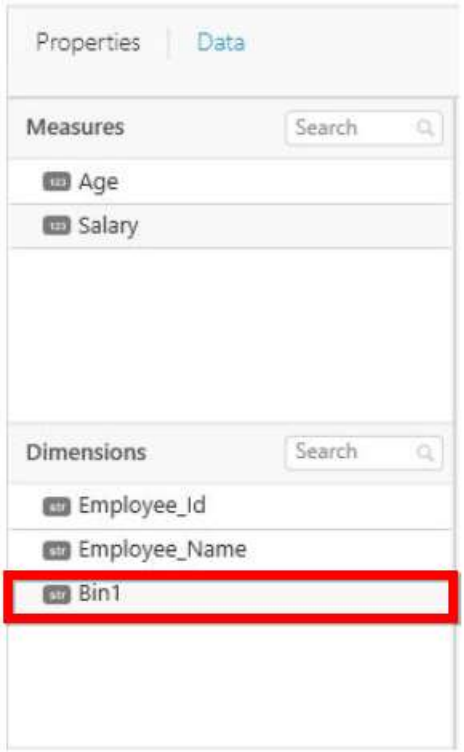
Deleting a created bin

You can delete a bin by clicking **Delete**.

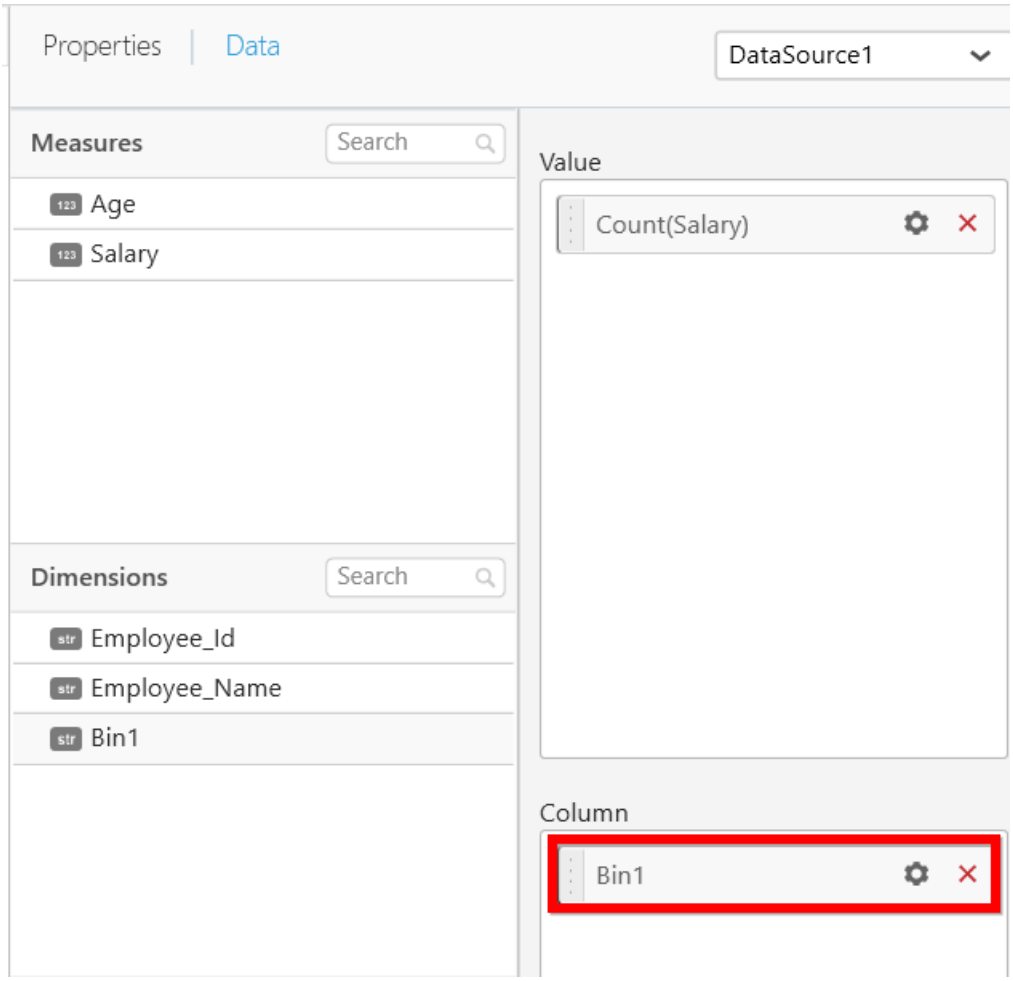


Configuring bin column in widgets

The bin will be shown in the **Dimension** section of the **Data** tab of the widget as follows.

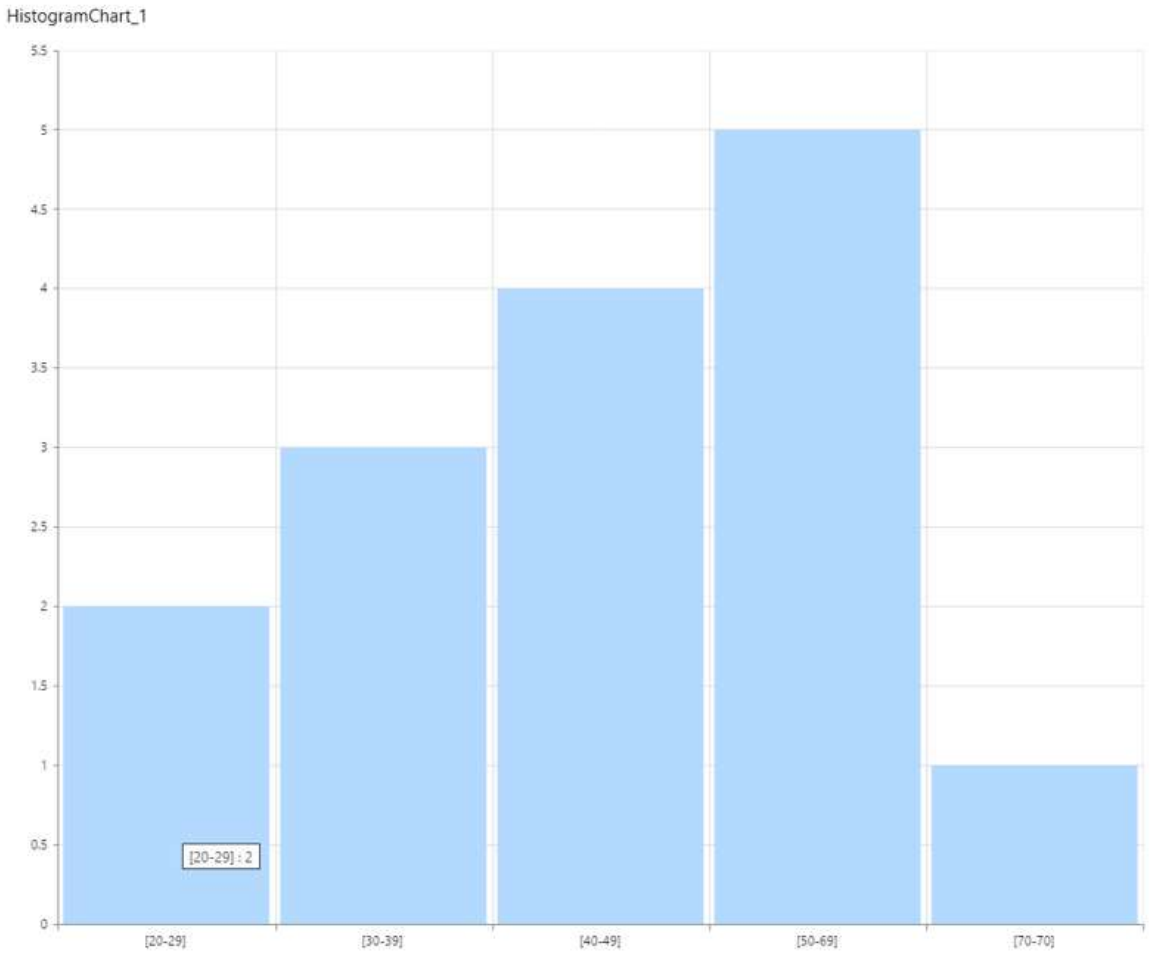


You can drag and drop the bin column into widgets like other fields.



Here, Bin1 is bound in the column like dimension field and the Salary column is bound in the value.

The output will be obtained as follows.



Note: You can use the Histogram Chart to understand the **Create Bin** support as shown above.

Data Blending

Data blending is required when you are going to use more than one data source column in a widget. Data blending can be performed on the common columns present in the table. So, it has the capability to put accurate actionable data.

For example, consider the following two data sources.

Data source1 contains Table A

Employee*Id*	*Employee*Name	Lead*Id*	*Stories*Completed
E1	James	L1	4
E2	John	L2	4
E3	Robert	L3	3
E4	Michael	L2	5
E5	Smith	L1	5
E6	Williams	L1	2

Data source2 contains Table B

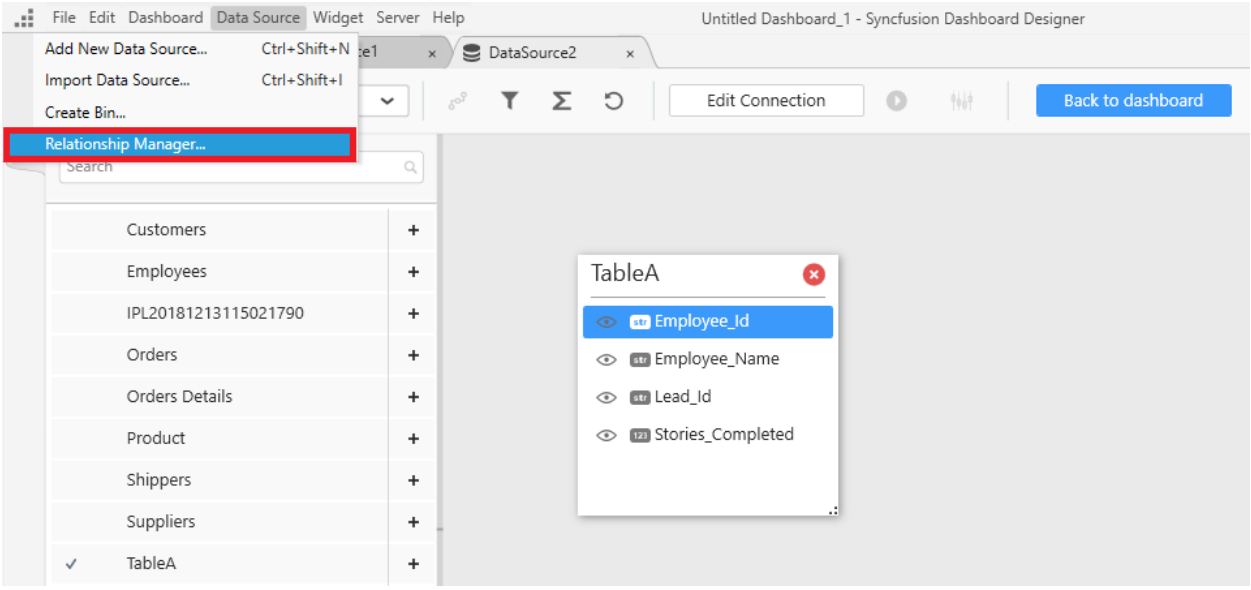
Lead*Id*	*Lead*Name	Month	LeaveDays_Count
L1	Charles	Jan	3
L2	David	Jan	1
L3	Joseph	Jan	3
L1	Charles	Feb	2
L2	David	Feb	4
L3	Joseph	Feb	2

If you blend the two data sources based on common column Lead_Id, the Syncfusion Dashboard Designer will return the output in the widget as follows.

Lead*Id*	*Lead*Name	SUM(LeaveDays*Count)*	*SUM(Stories*Completed)
L1	Charles	5	11
L2	David	5	9
L3	Joseph	5	3

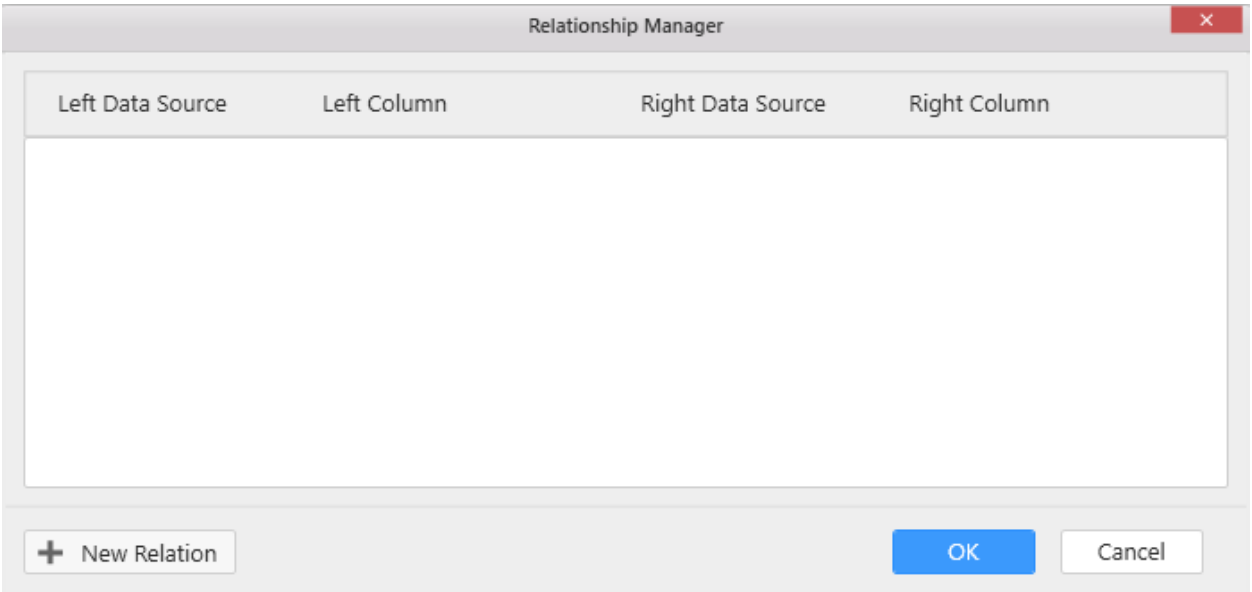
Adding a relationship

To get the blending result between two data sources, you should set relationship between the two data sources using the Relationship Manager in the Data Source menu.

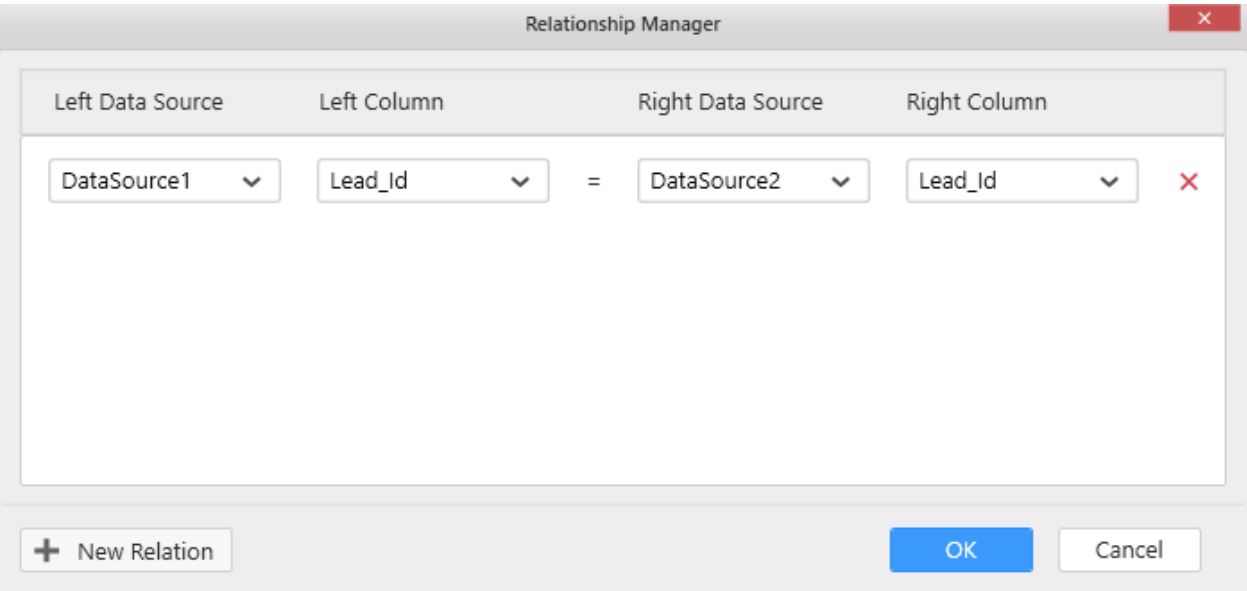


Note: The Relationship Manager menu is enabled only if you have two data sources with same connection string in SQL connection type.

After you click the Relationship Manager in the Data Source menu, the Relationship Manager window will be opened as follows.

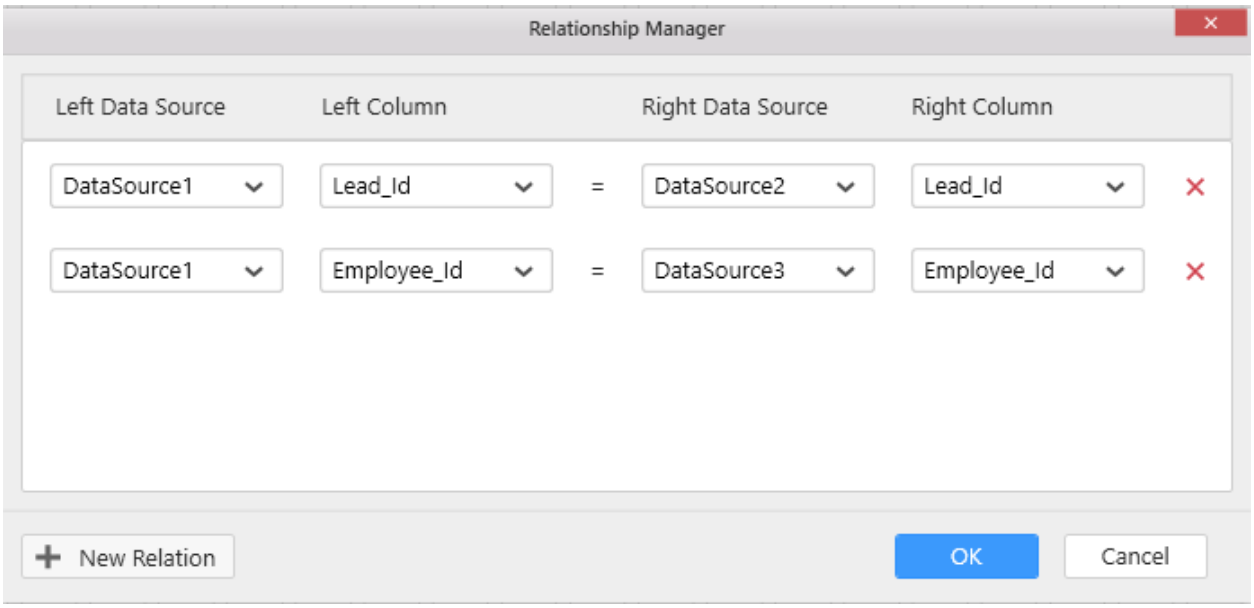


Click New Relation to define your relationship based on the columns present in the data source tables.



In the above screenshot, the data source lists are shown in the Left Data Source and Right Data Source combo boxes. You can choose the data sources for which the relationship is needed. The Left Column shows the columns present in the Left Data Source tables and the Right Column shows the columns present in the Right Data Source tables.

You can add more than one relation if you prefer by clicking the New Relation button.



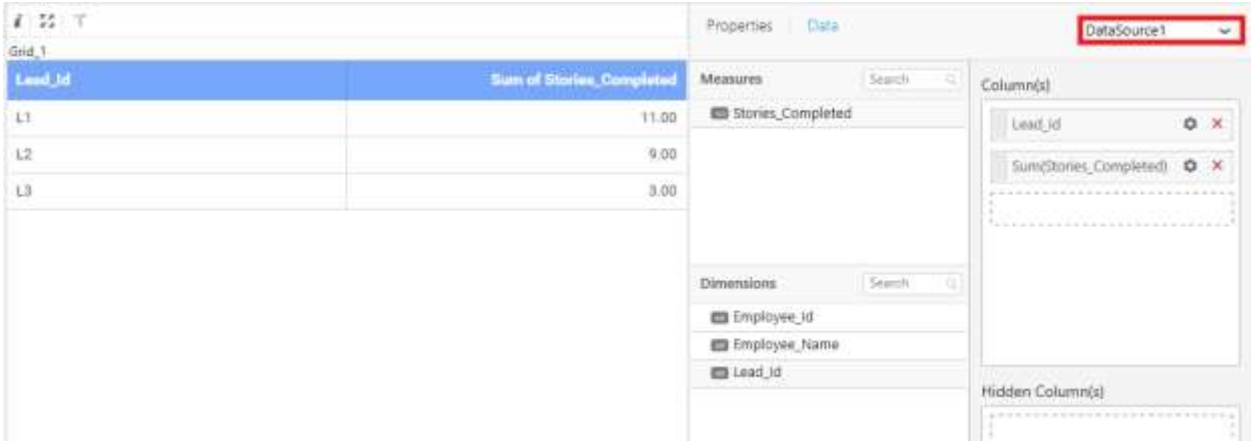
Click OK to update the Relationship Manager window.

Configuring a widget with blending data

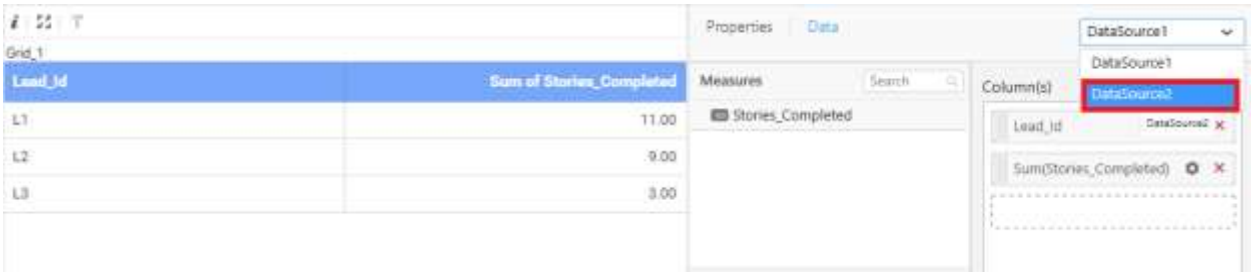
You can configure your widget using the data sources for which blending relationship is applied.

For example, consider the grid widget with two data sources.

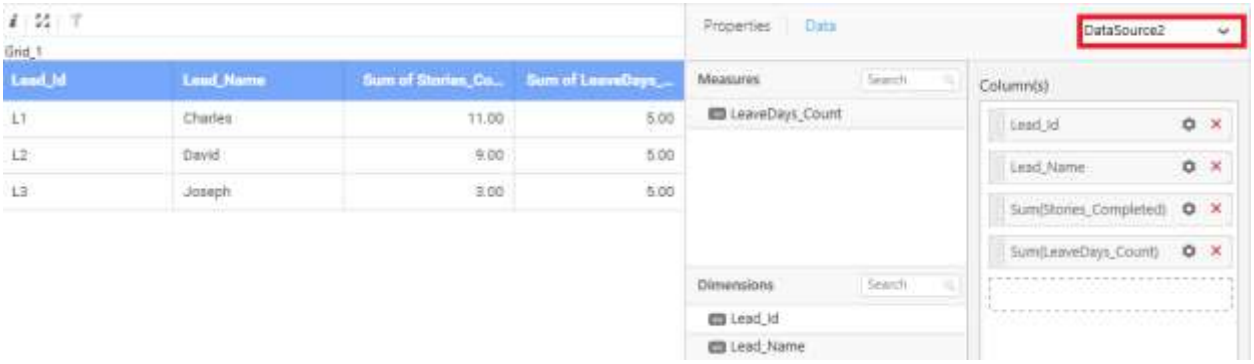
Bind the column named Lead*Id, Stories*Completed in the grid widget from Data source1.



To bind the columns from Data source2 in the grid widget, switch the data source from Data source1 to Data source2.



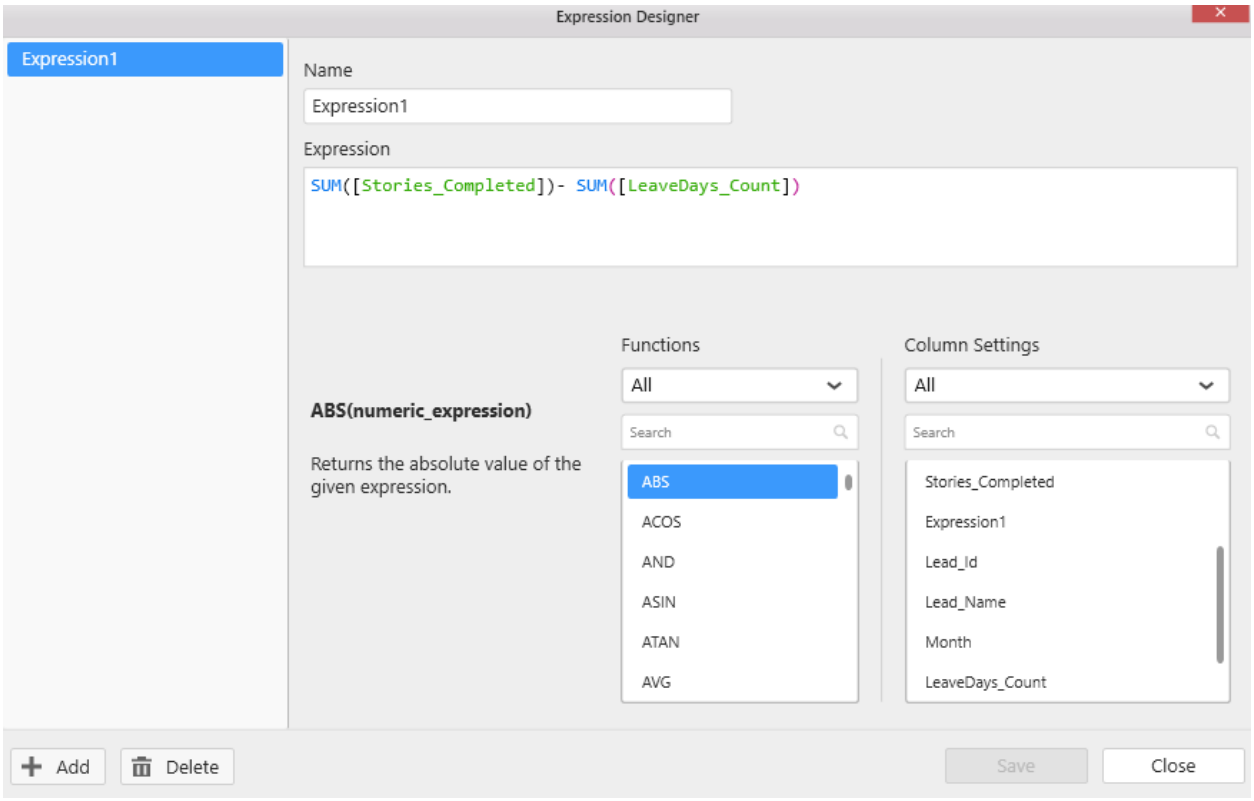
Now, you can bind the columns from both data sources as follows



Creating multi data source column expression

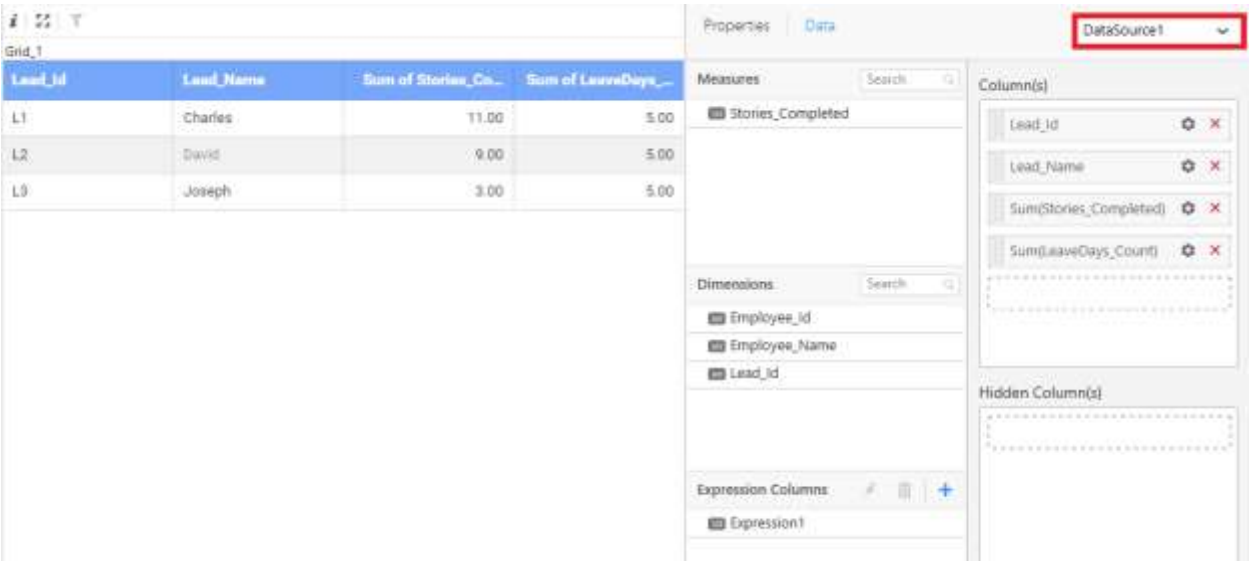
You can create an expression using two data source columns that have relationship between them.

For example, if you want to create an expression for Data source1 using the columns present in both Data source1 and Data source2, you can create it as shown in the following screenshot.



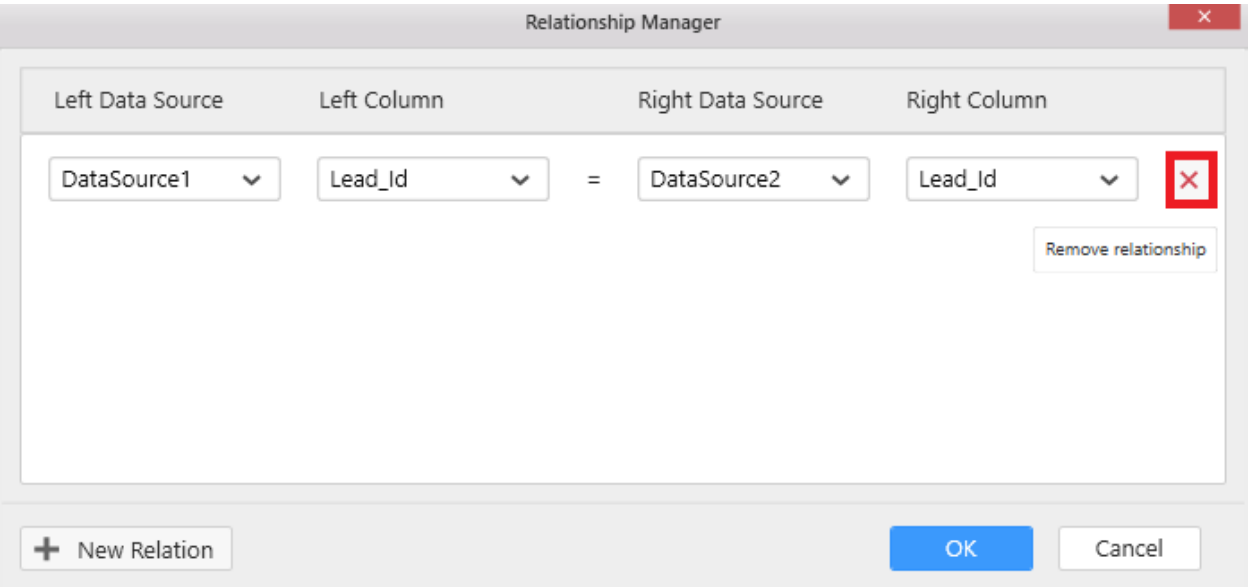
Note: The created expression column will be maintained only in the data source, in which the expression is created.

For example, if the Expression1 is created in the Data source1, then the Expression1 will be shown only for Data source1 in the Control Designer page.



Removing a relationship

You can remove the relationship condition by clicking the Remove relationship button as follows.



Click OK to update the Relationship Manager window.

Note:

Removing a relationship from the Relationship Manager will reset your widget in which the blending relationship is used.

Note:

Removing a relationship from the Relationship Manager will also delete your expression created using the blending relation.

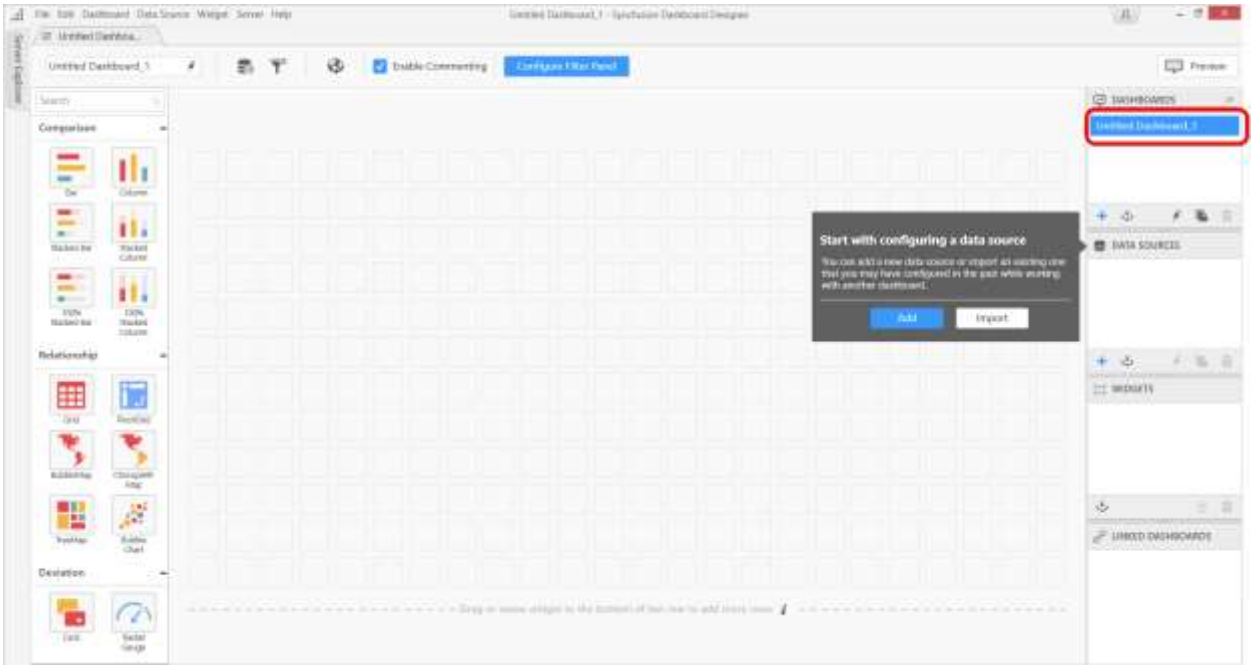
Compose Dashboard

Compose Dashboard

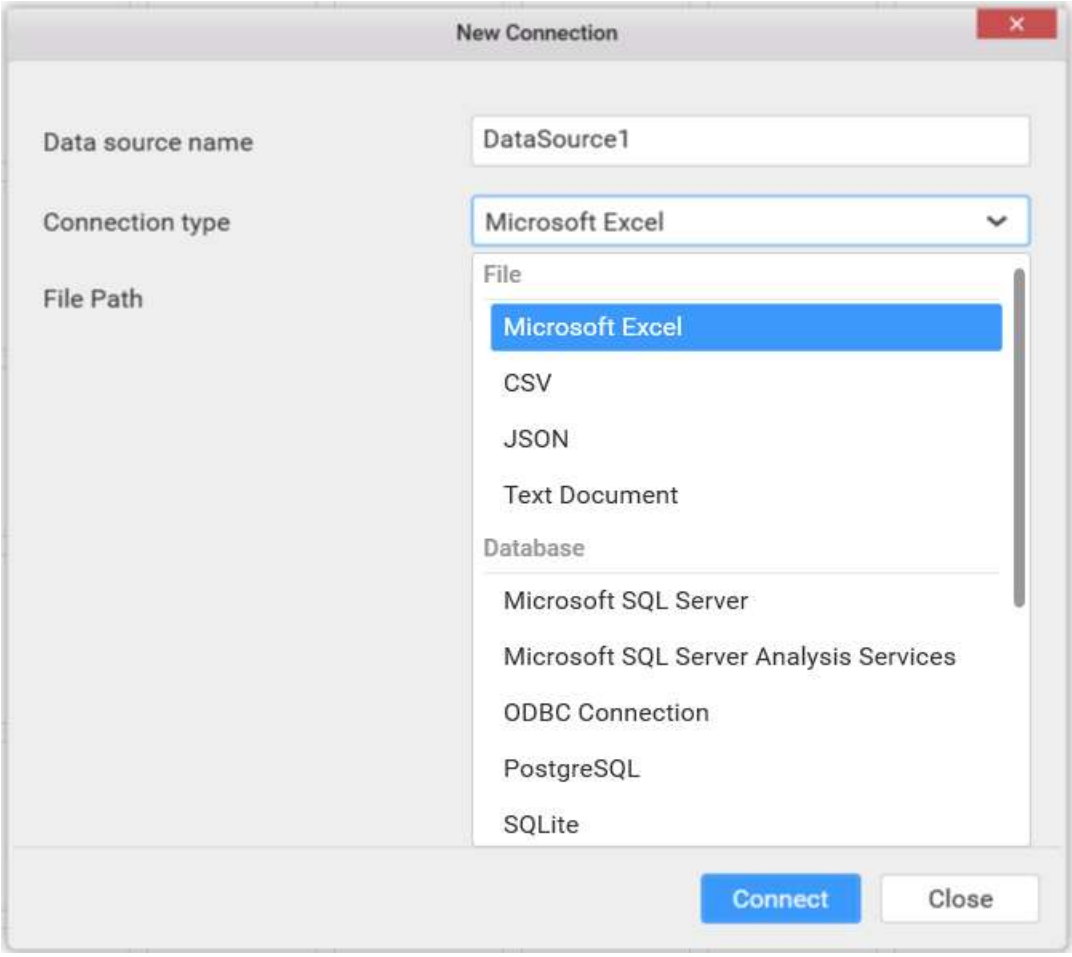
Once you are ready with the formatted data source, you can start to compose a new dashboard with required filter, data visualization and/or miscellaneous widgets configured with data.

Creating a Dashboard

Run the Syncfusion Dashboard Designer application through the shortcut icon that has been placed on your desktop. After the application is launched, you can see a prompt to configure a data source first.



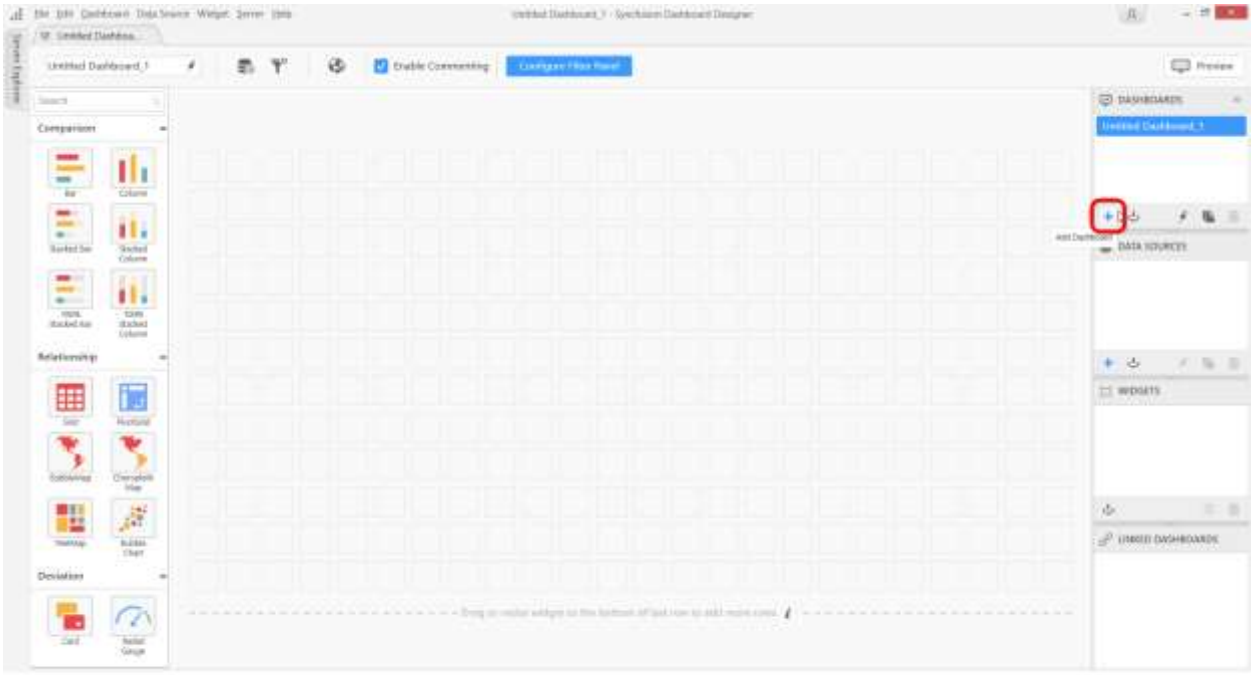
You can click the Add option to select the required Connection type to add the data source into the dashboard.



Drag and drop the Dashboard widgets based on your requirements.

Creating multi tabbed Dashboard

You can start adding multiple dashboard reports in a single dashboard report by clicking the Add Dashboard icon or clicking the Add Dashboard option in the Dashboard file menu.



Now the second dashboard will be created and the tab will be opened.

You can find the name of the active dashboard in the dashboard container.

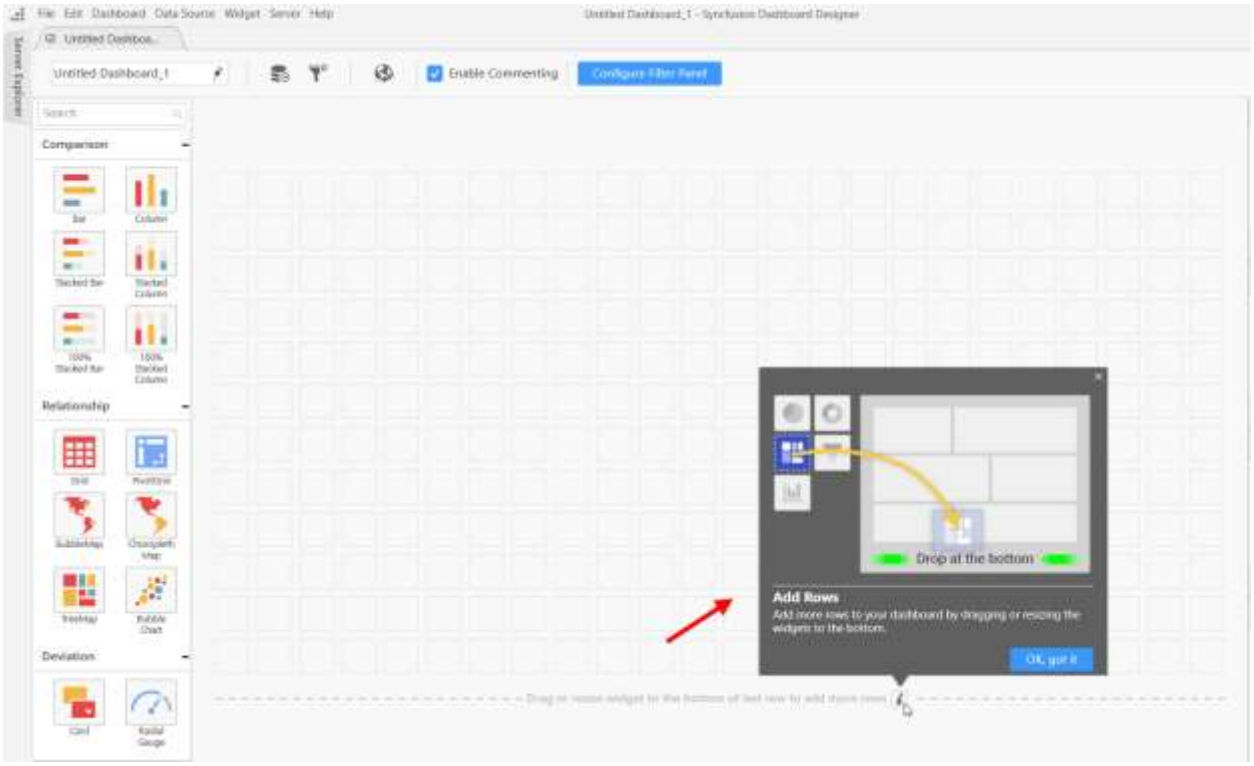
You can add any number of dashboards to the dashboard report.

Note: You can add any number of dashboard tabs in a single dashboard report(***.sydx file**).

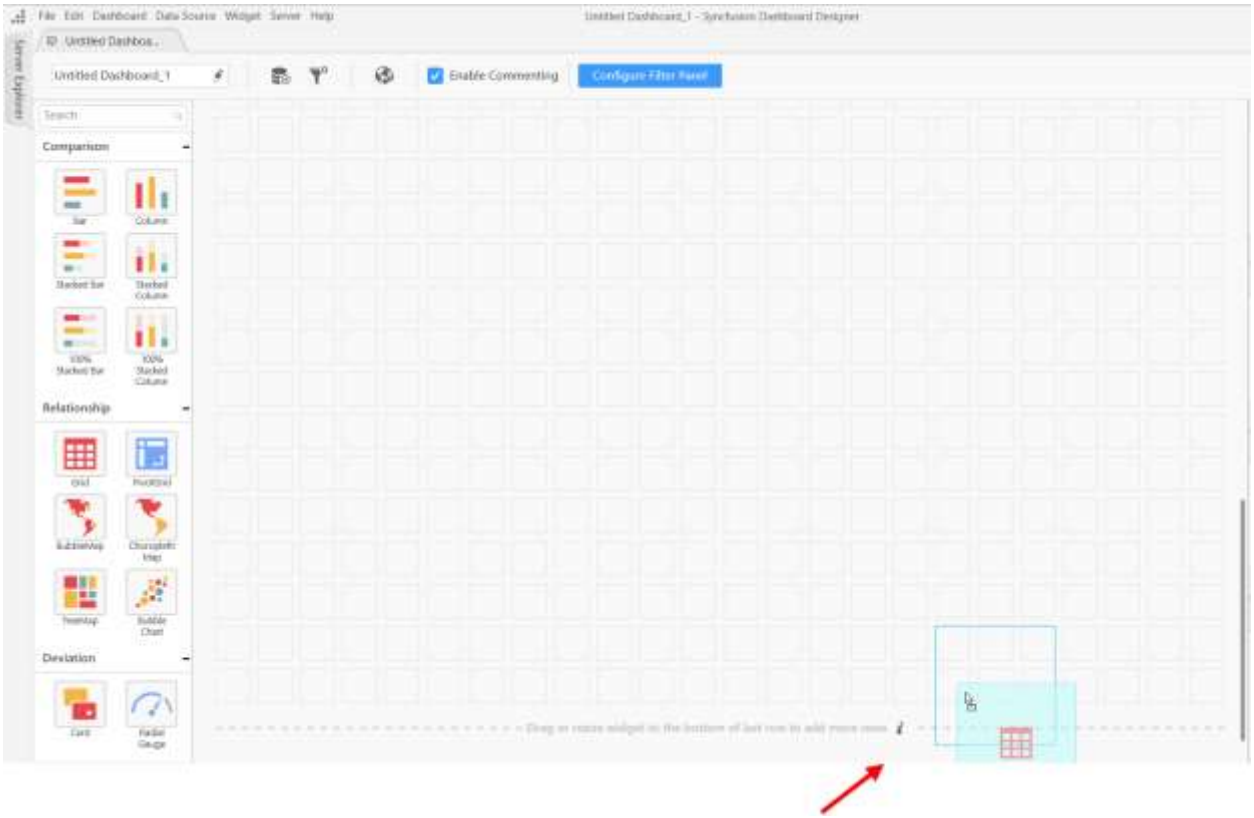
Note: You can also use the import existing dashboards option to create a multi-tabbed dashboards.

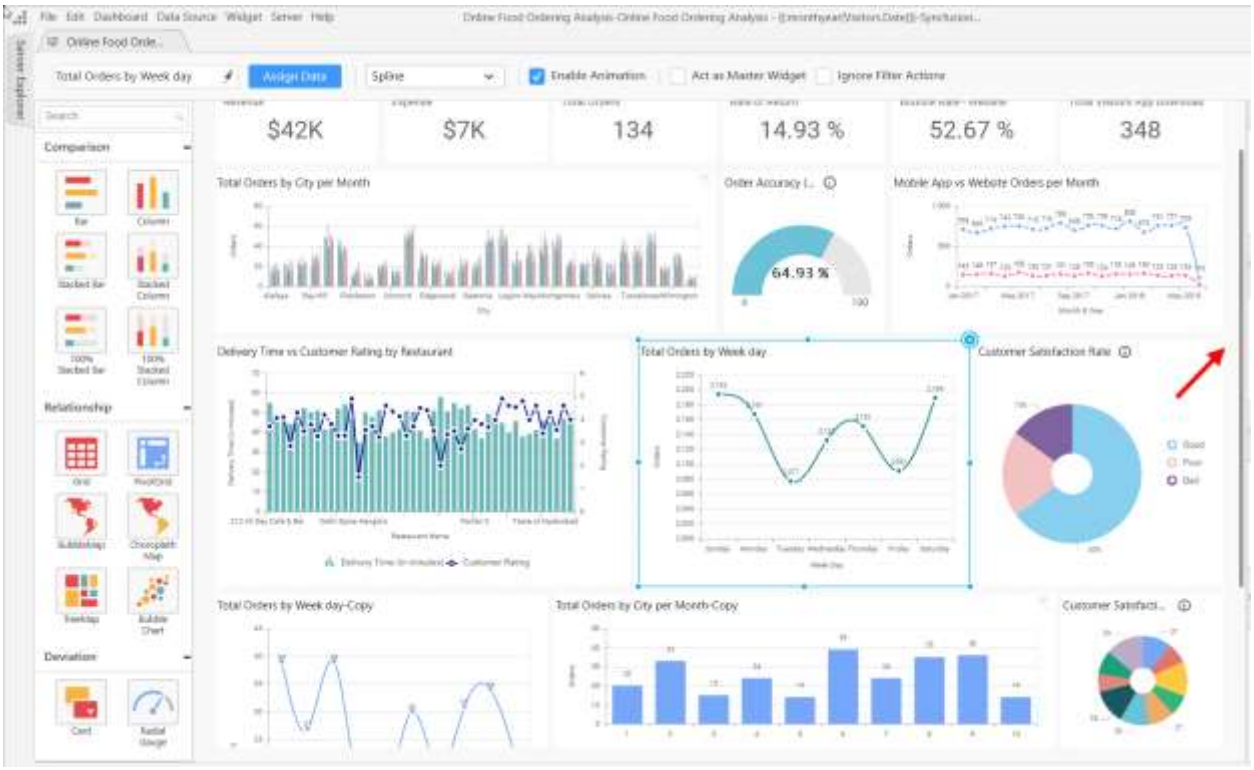
Creating scrollable Dashboard

You can add more number of widgets in same page by using scrollable canvas support.



To enable scrollable, canvas drag or resize the widgets beyond the last row of canvas.

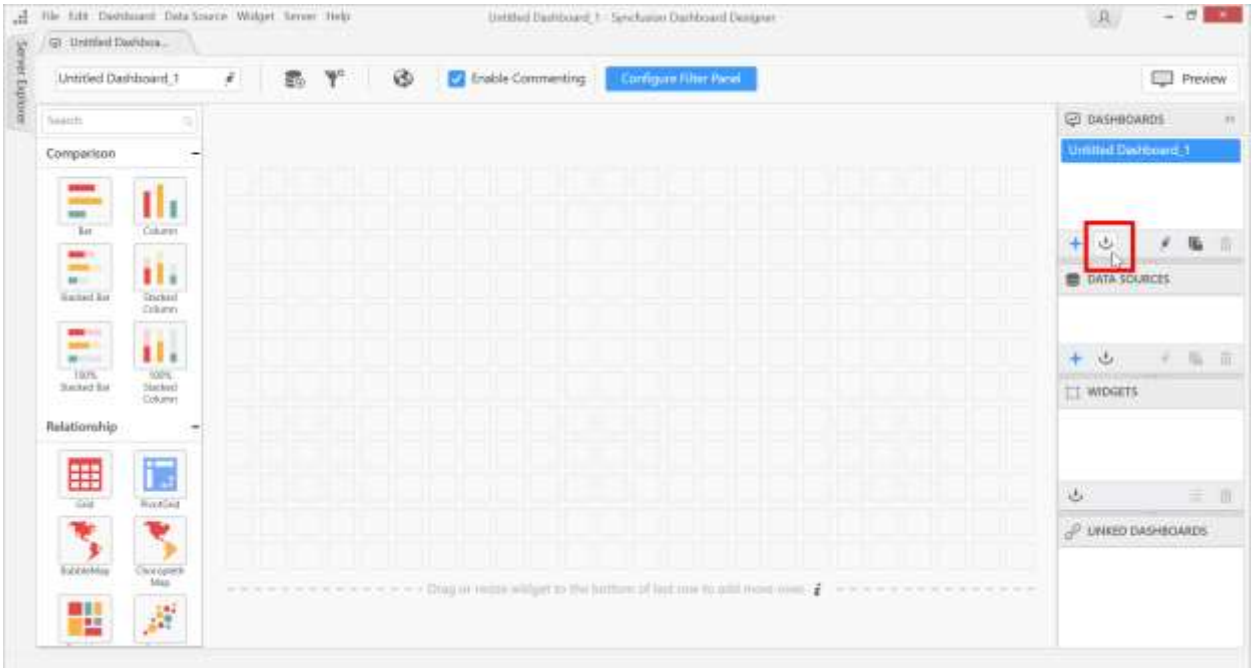




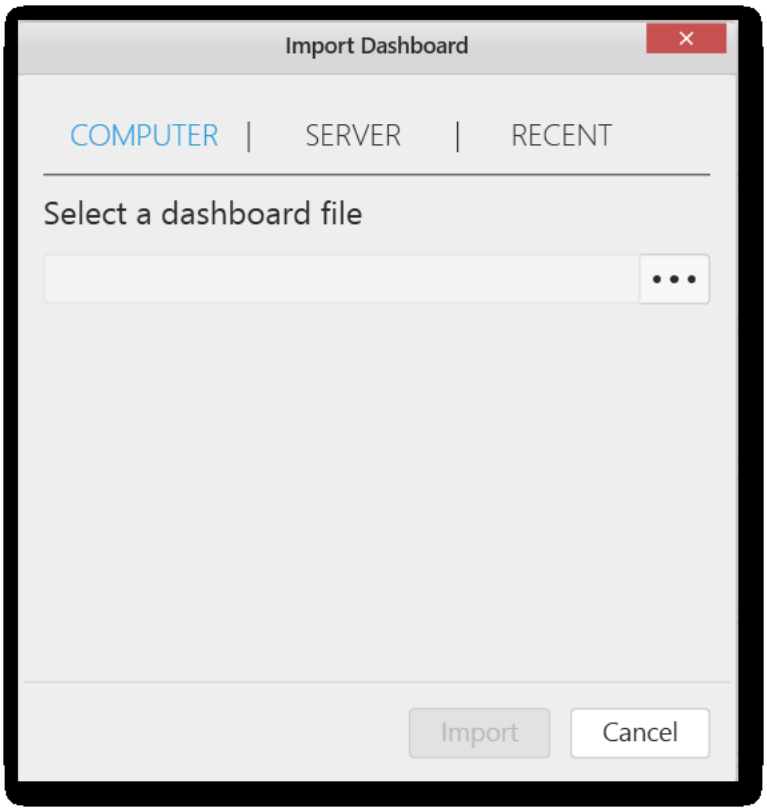
Note: You can add infinity number of rows to single page.

Importing Existing Dashboards

You can import the already saved dashboards by clicking on the Import Dashboard option provided in the Dashboard container.

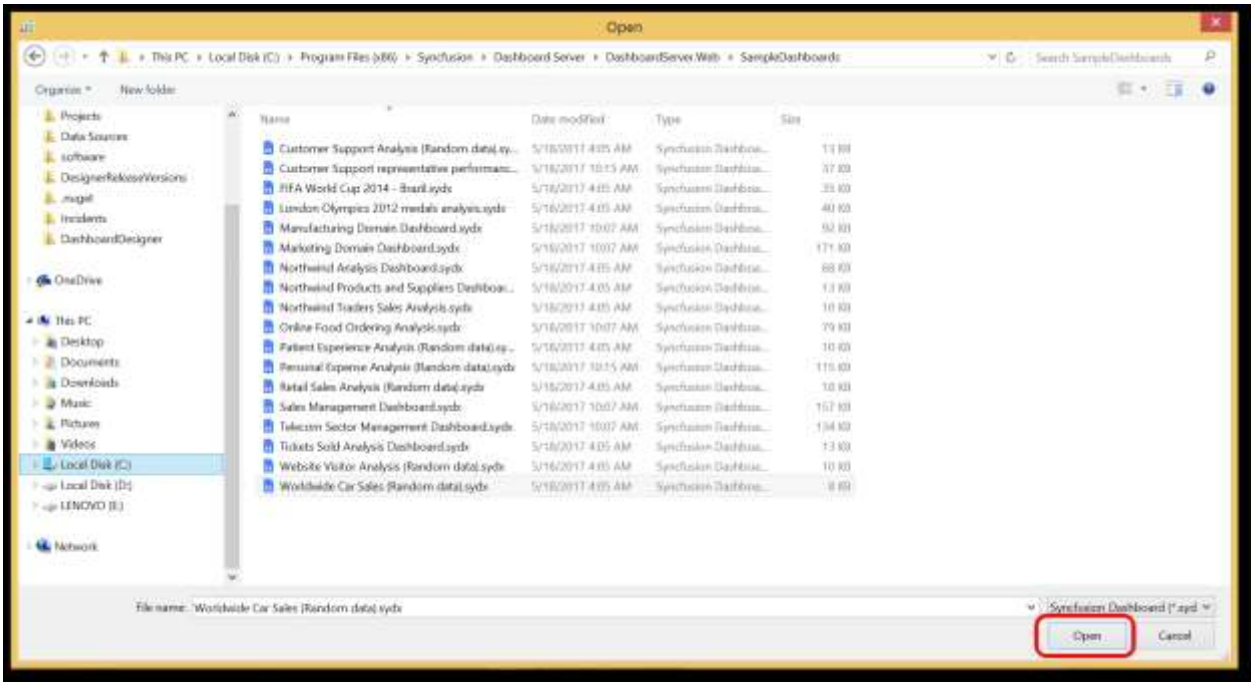


Now the Import Dashboard window will be opened.



Importing from computer

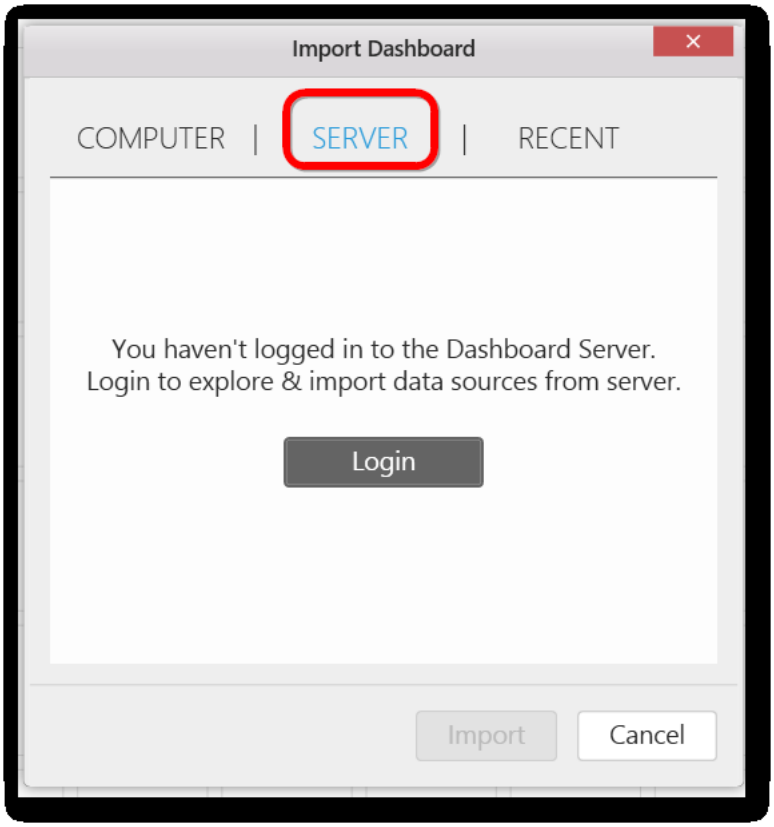
Using the file browser you can select the Dashboard file (.SYDX) already saved in your machine.



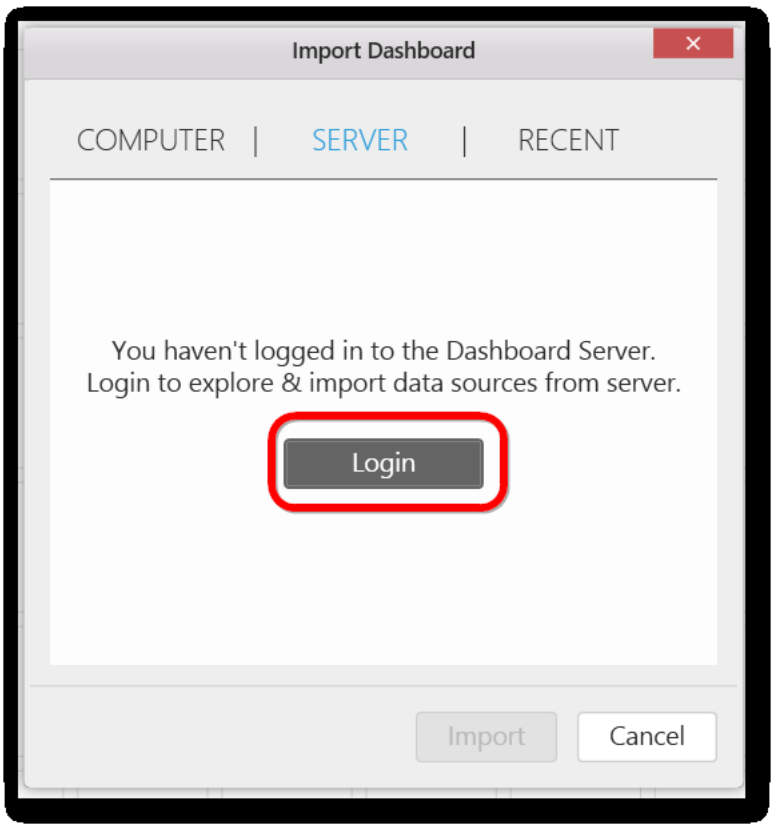
You can import dashboards from your local drive , Recent list or from dashboard server.

Importing from Server

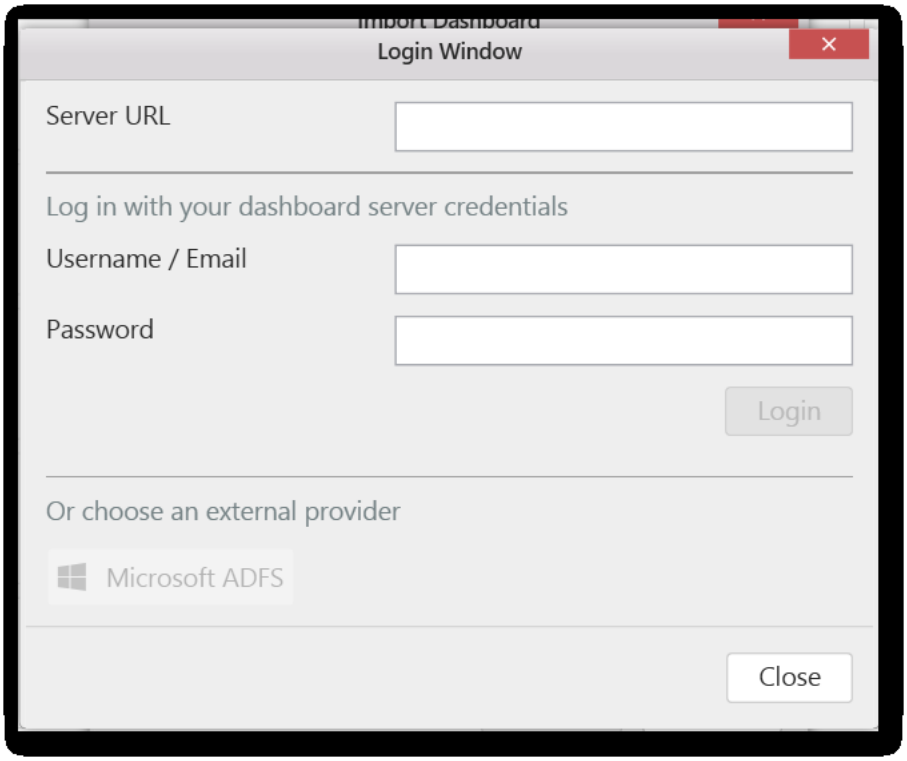
To import the dashboards published in your Dashboard server select the Server option in the Import dashboard window.



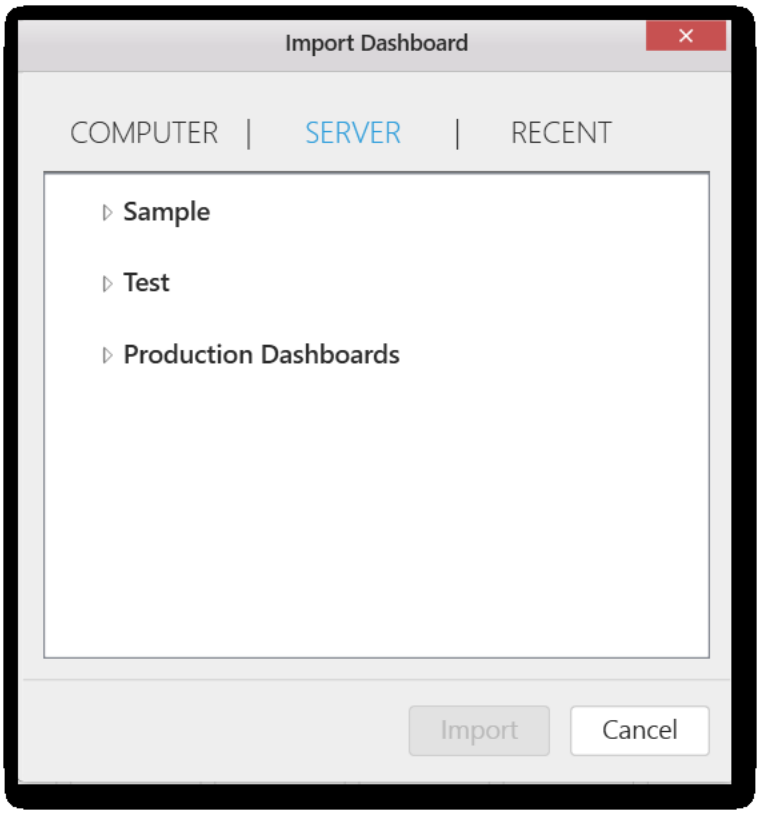
If you are not logged in your Dashboard server account, first login to the Server using the login option shown below.



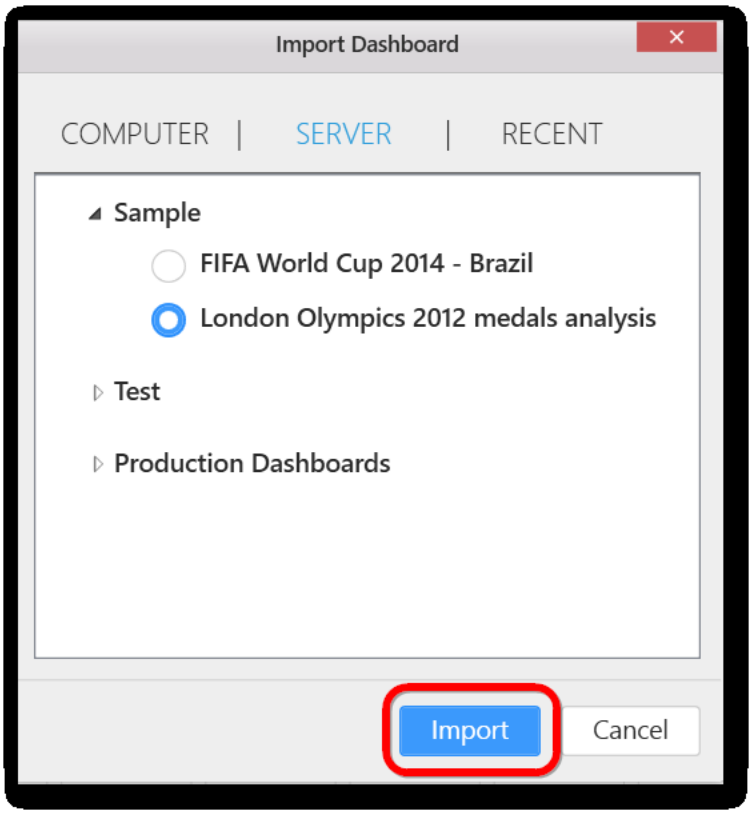
Fill the following details and log in to the Server.



Now the dashboard files available in the list will be shown here.

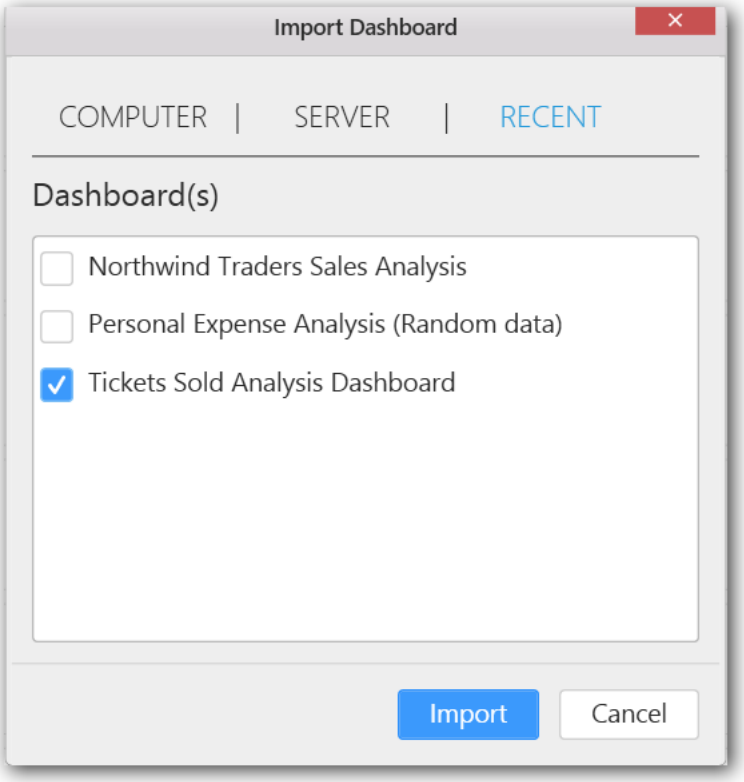


Expand the category you want and select the dashboard. Now click on Import button as shown in the below image:

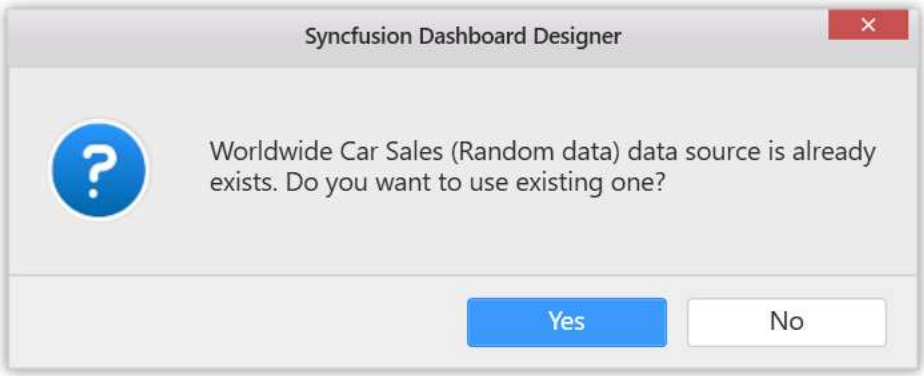


Importing from Recent dashboard list

Recently used dashboard will be listed in the list and you can select the dashboards you want to import by clicking on the Import button.



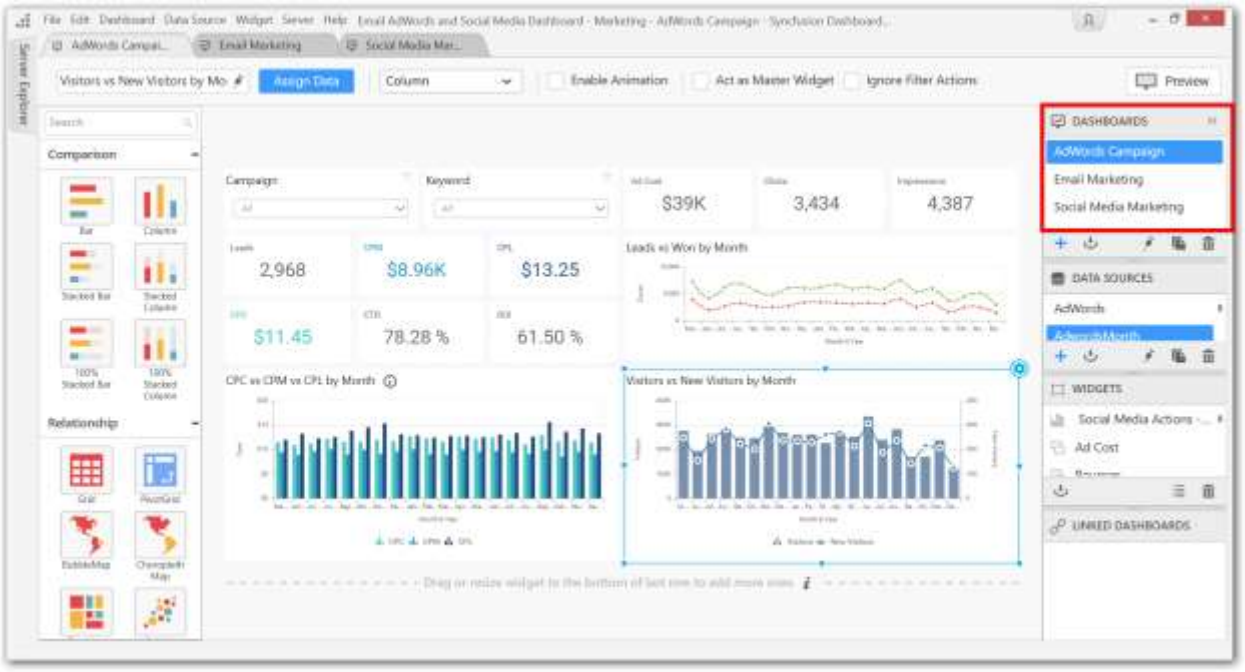
Information: If the imported dashboard has been configured using the data source that is available in the current dashboard report, the following alert message will be shown when using the existing data source without importing the redundant data source file again. If you click yes, the existing data source added in the current dashboard report will be used in the imported dashboard. Otherwise, the data source used in the imported dashboard file will be imported again.



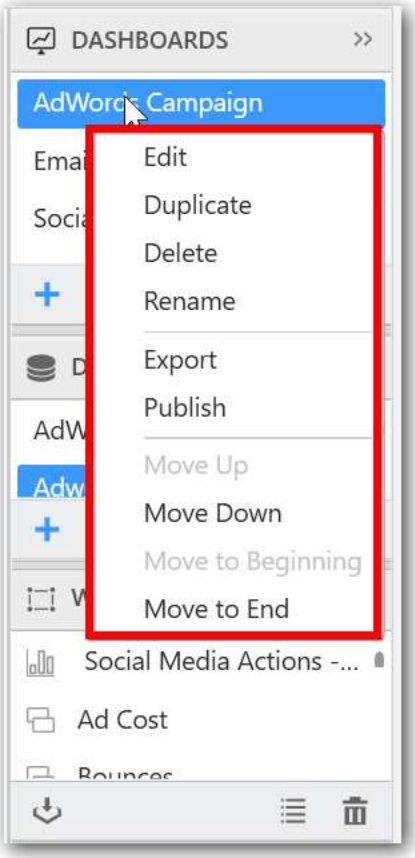
Exporting a dashboard

You can save a single dashboard tab as SYDX file from multiple dashboards using the Export option by following the below steps.

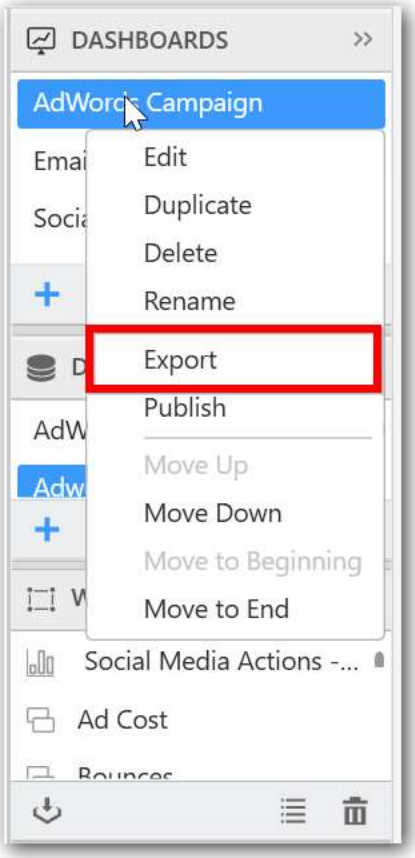
1. Select the dashboard tab you want to export in the Dashboards container.



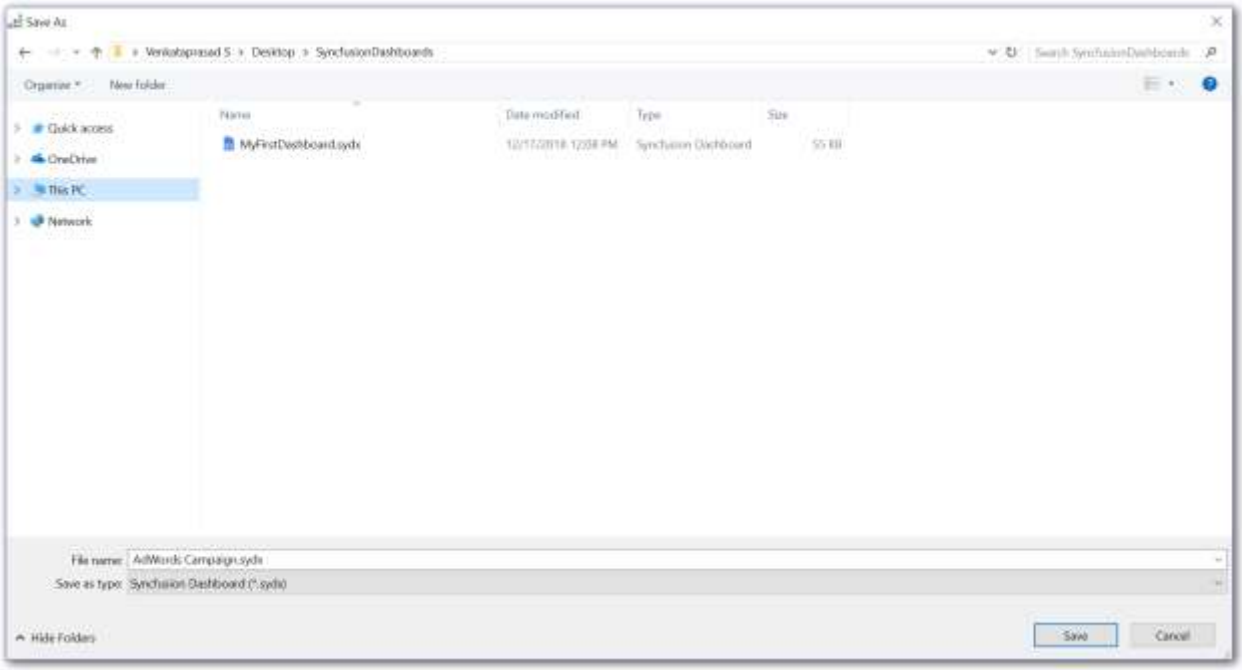
2. Right click on the widget name to open the context menu.



3. Select the Export option in the menu.



4. Select the file name and the file path in the save dialog.

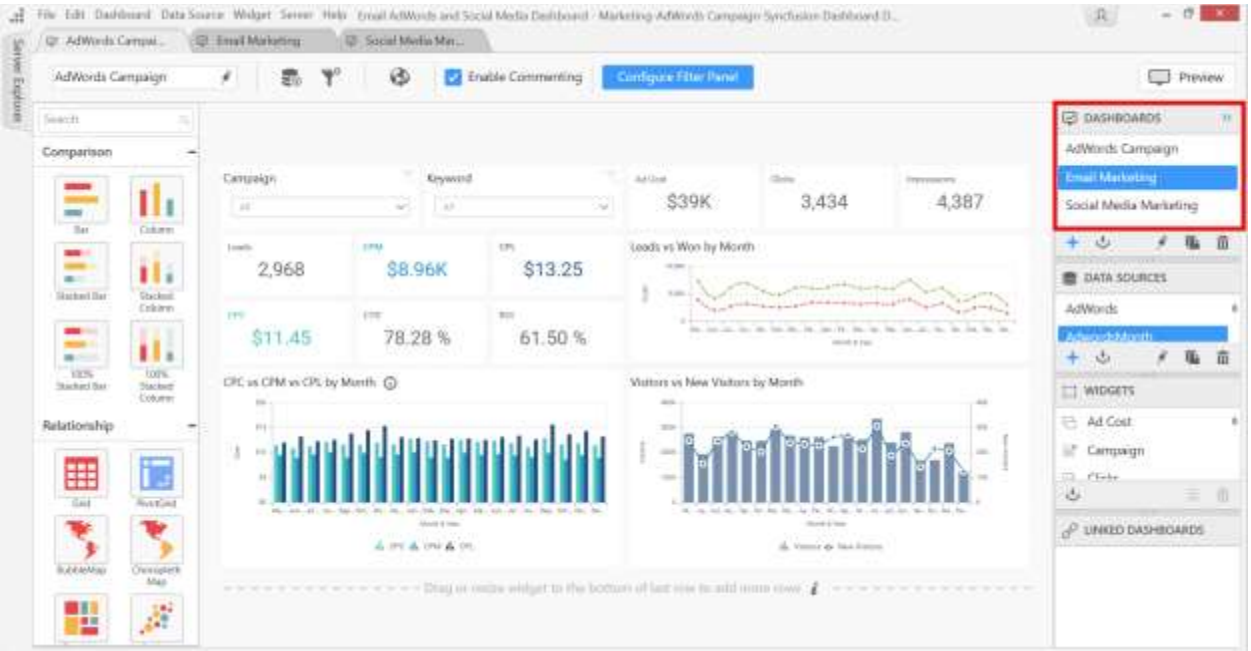


Note: The Dashboard report and its corresponding data sources and widgets will be exported in the Syncfusion Dashboard(.sydx) file format. You can import the sydx files in Dashboard Designer application any time or you can also add the file to your Dashboard Server application.

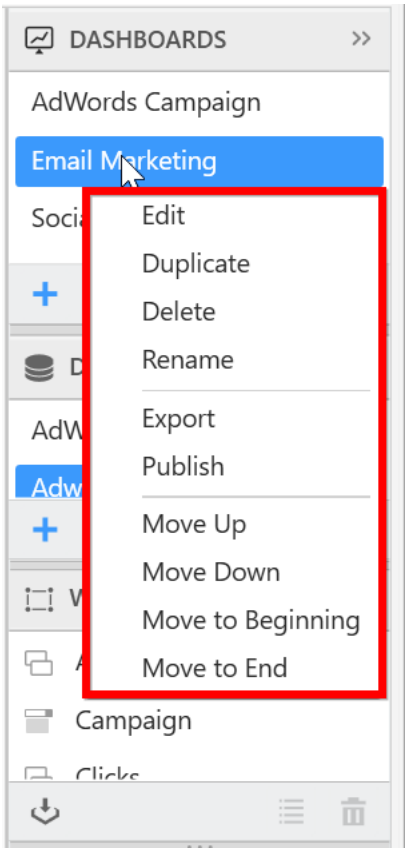
Renaming a dashboard

You can rename the Dashboard to a meaningful name as you require by following the below steps:

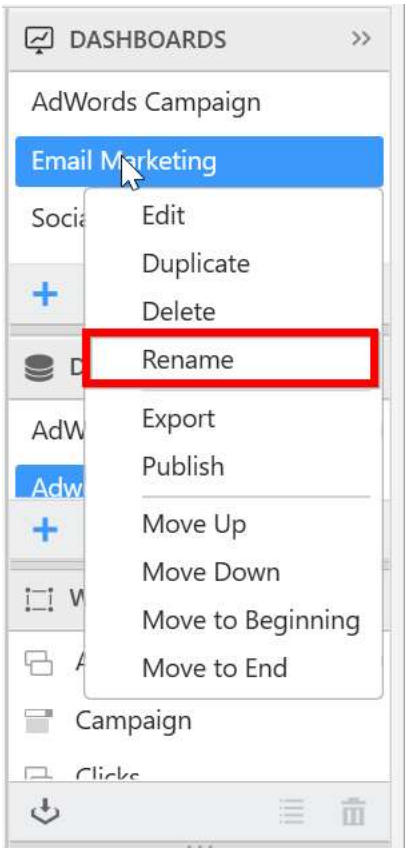
1. Select the dashboard tab you need to rename in the DASHBOARDS container.



2. Right click on the dashboard name to open the context menu.

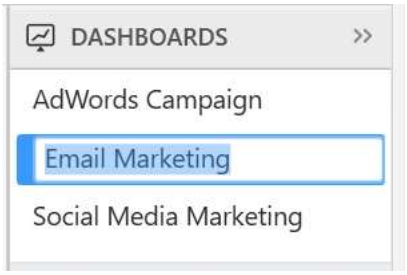


3. Select the Rename option and the text box will be displayed now.



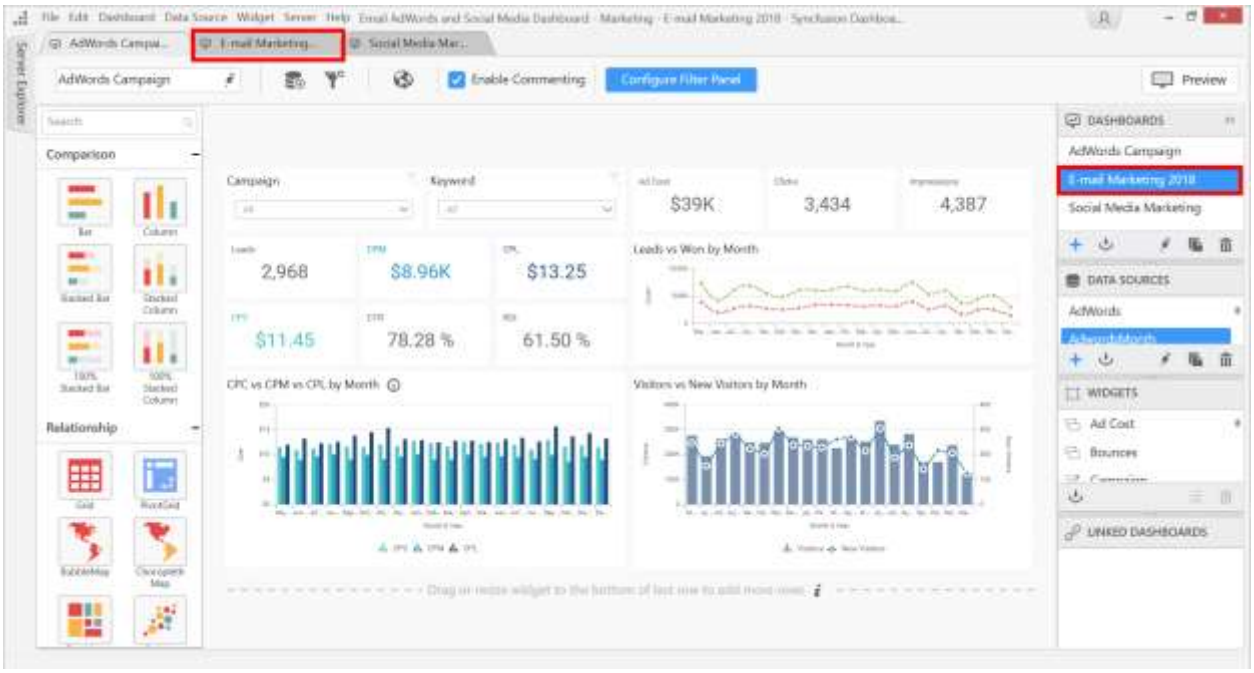
Note: You can also you can use the Keyboard shortcut F2.

4. Type the name you want to set for the Dashboard.



Note: The Dashboard names must be unique and non-empty.

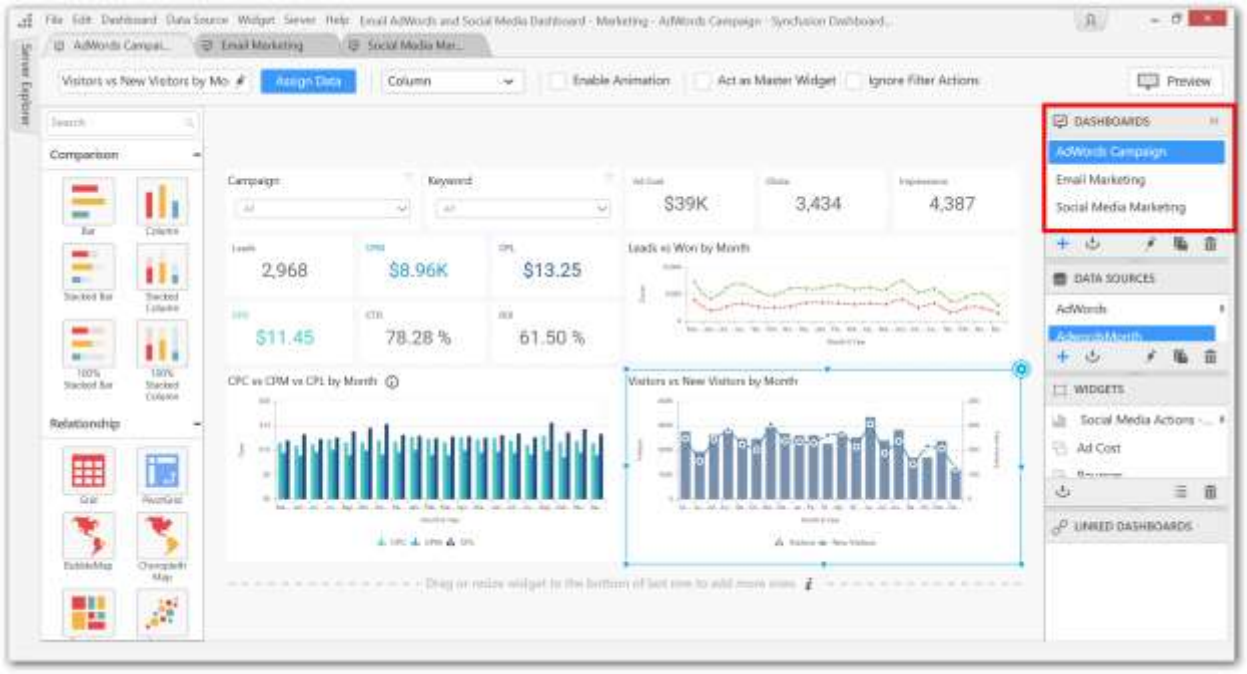
Now the dashboard name will be changed in the respective places.



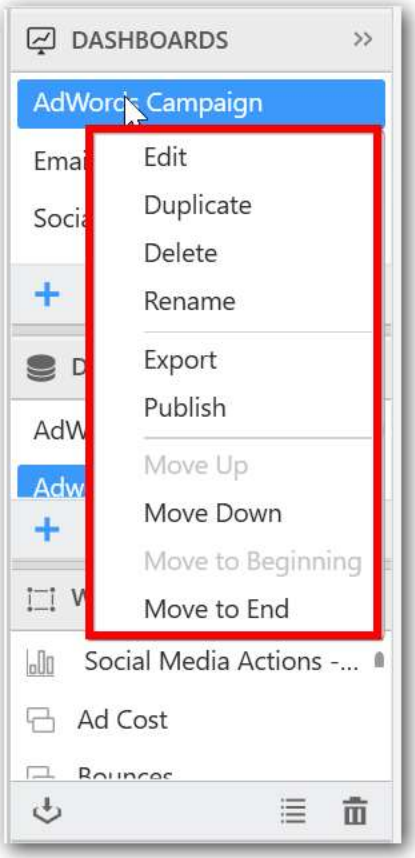
Deleting a specific Dashboard

You can delete the specific dashboard tab from multiple tabs by following the below steps.

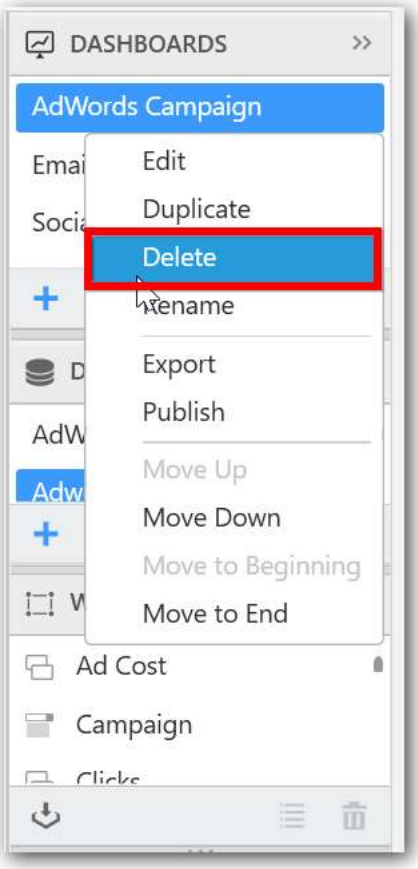
1. Select the dashboard tab you want to delete in the Dashboards container.



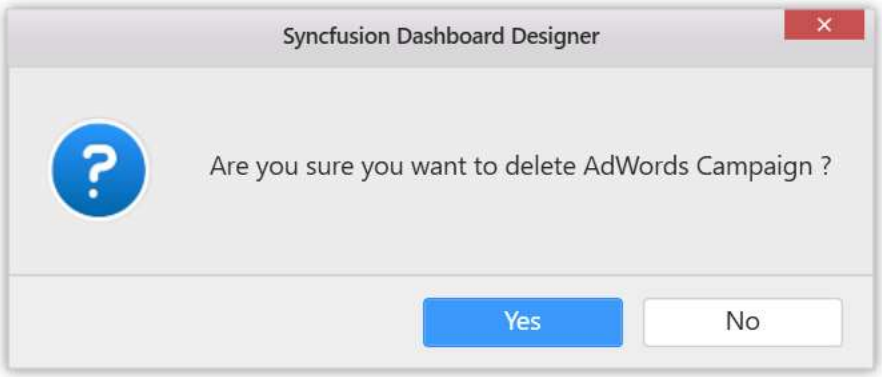
2. Right click on the widget name to open the context menu.



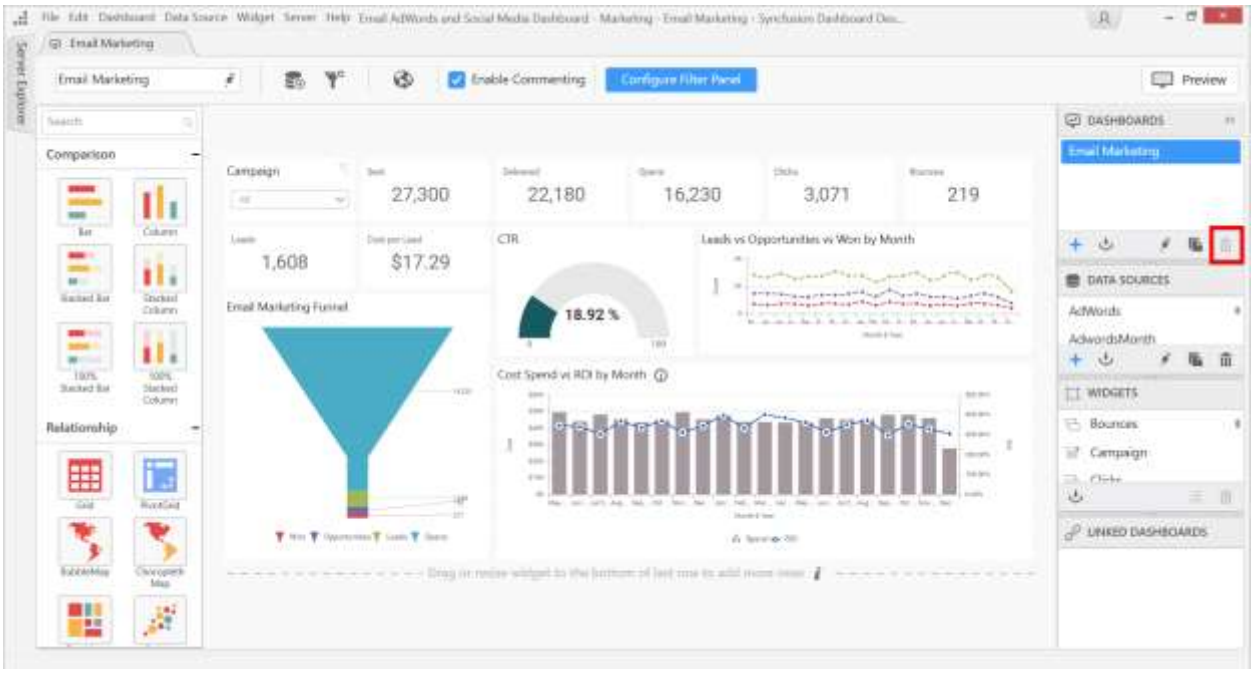
3. Select the Delete option in the menu.



4. Confirmation alert will be shown before deleting the dashboard tab, so click Yes to delete. Otherwise click on No option.



Note: If you have only one dashboard in your current report then delete option will be disabled by default.

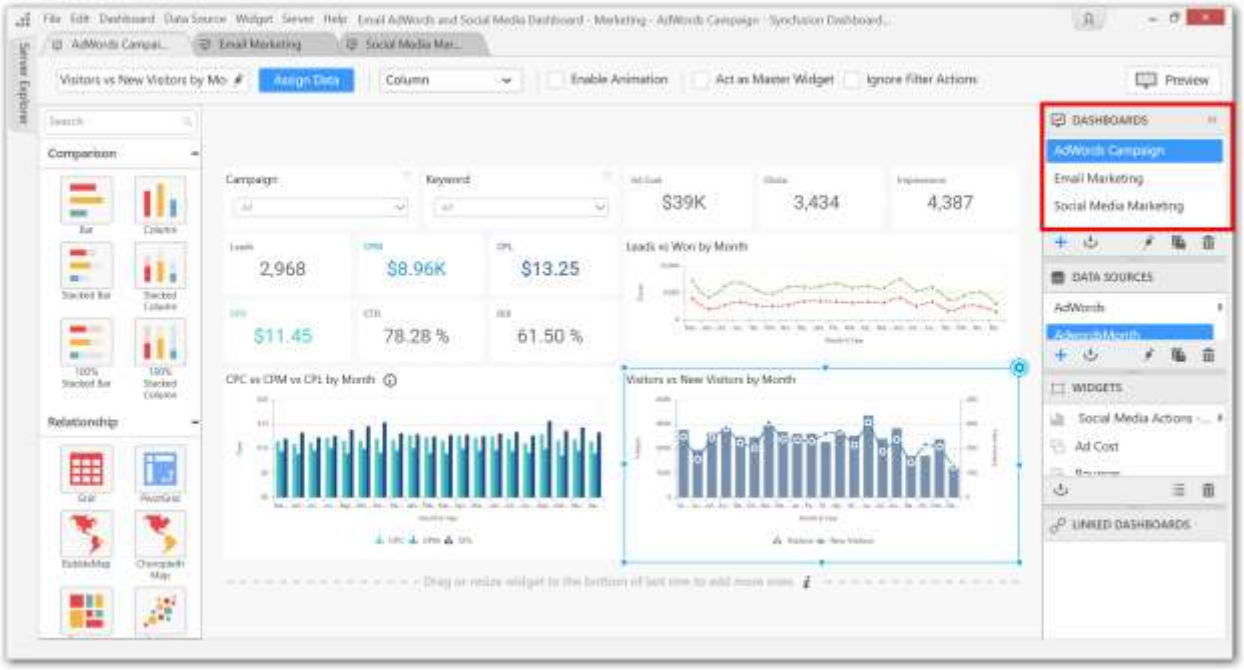


Note: Now, the dashboard and widgets that are used in the dashboard report will be deleted. The shared widgets and the data sources will not be deleted.

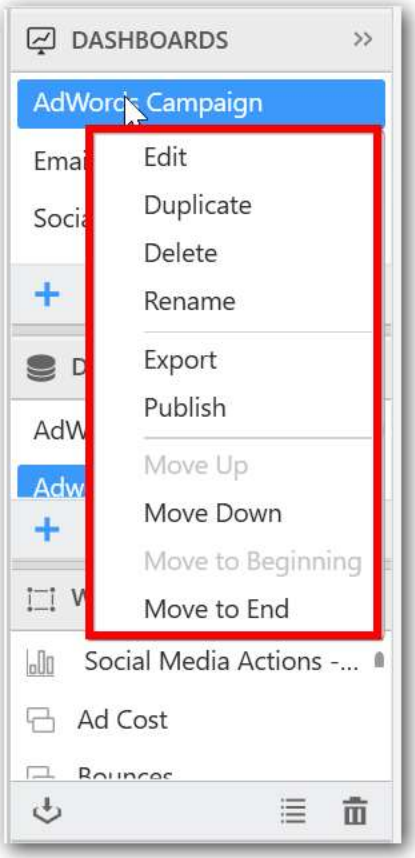
Editing a dashboard

You can open and edit a particular dashboard tab by using the Edit Dashboard option.

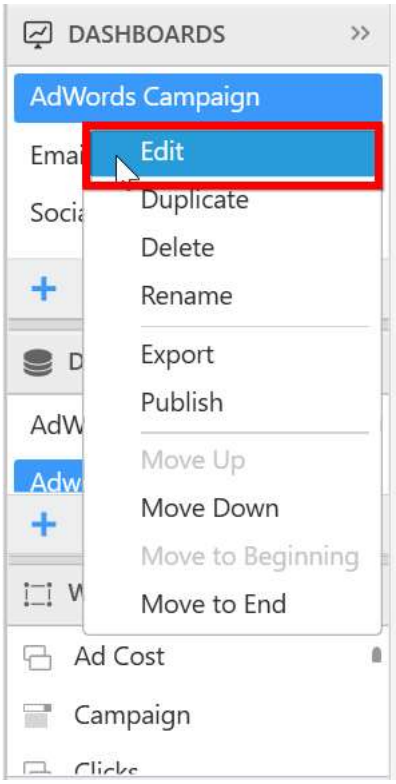
1. Select the dashboard tab you want to open or edit in the Dashboards container.



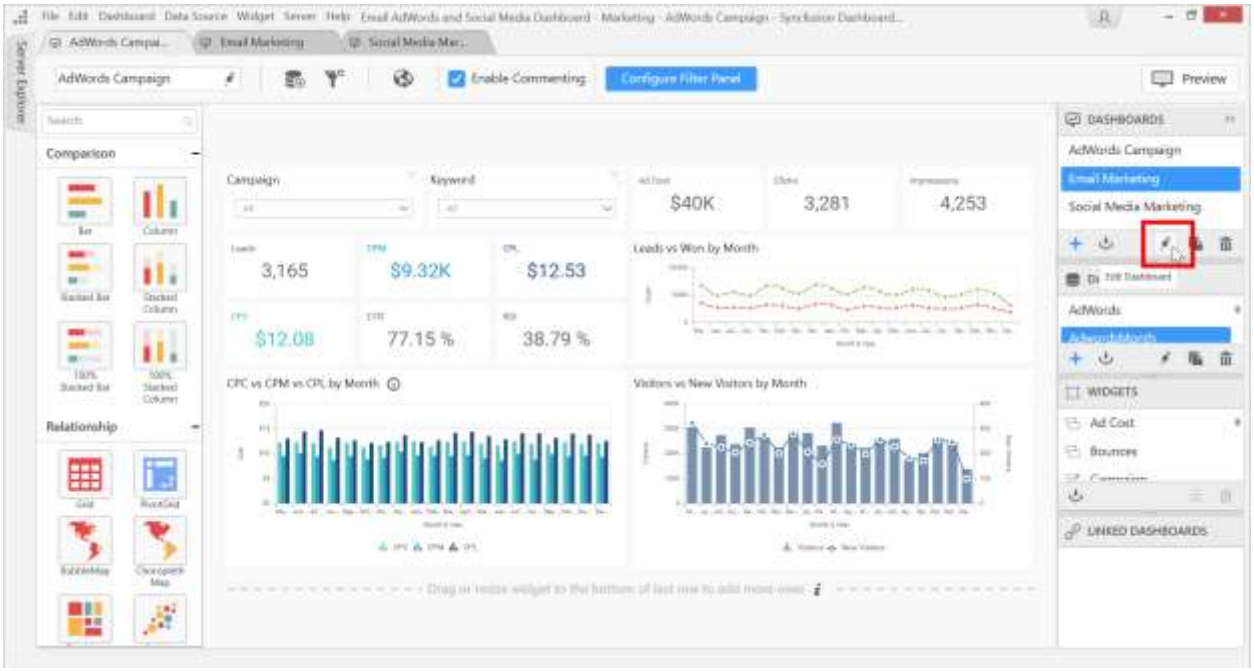
2. Right click on the widget name to Edit the context menu.



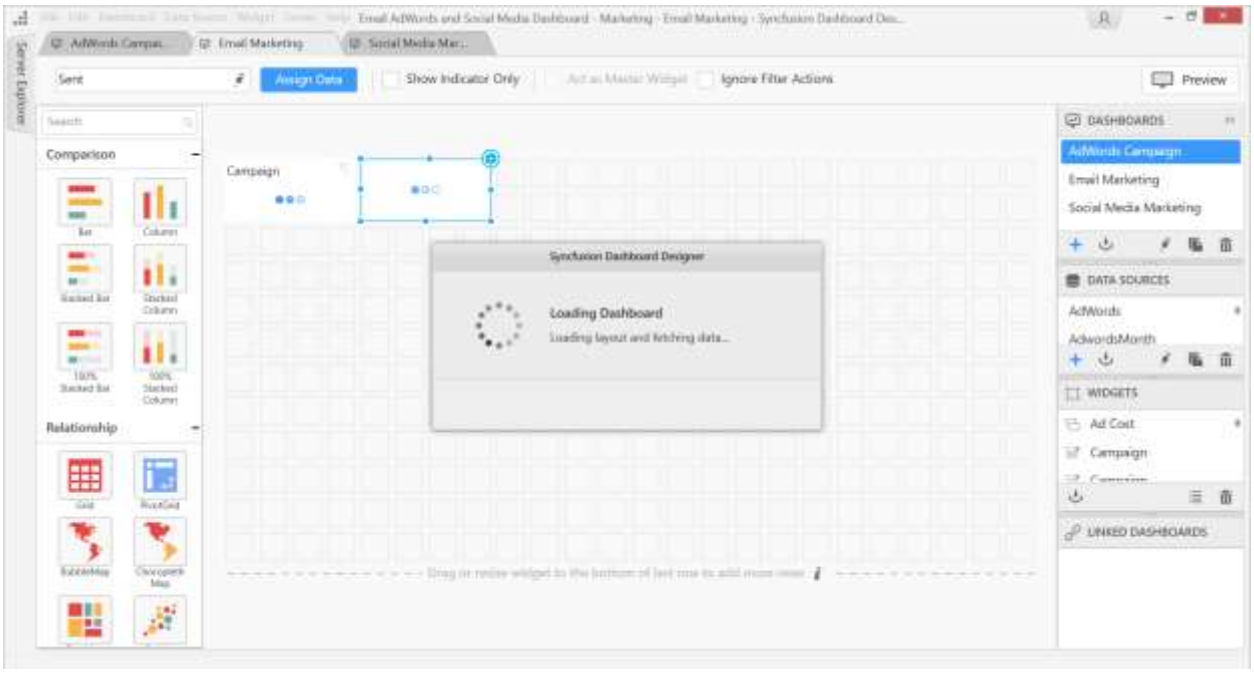
3. Select the Edit option in the menu.



You can also use the Edit dashboard icon for editing a particular dashboard tab.



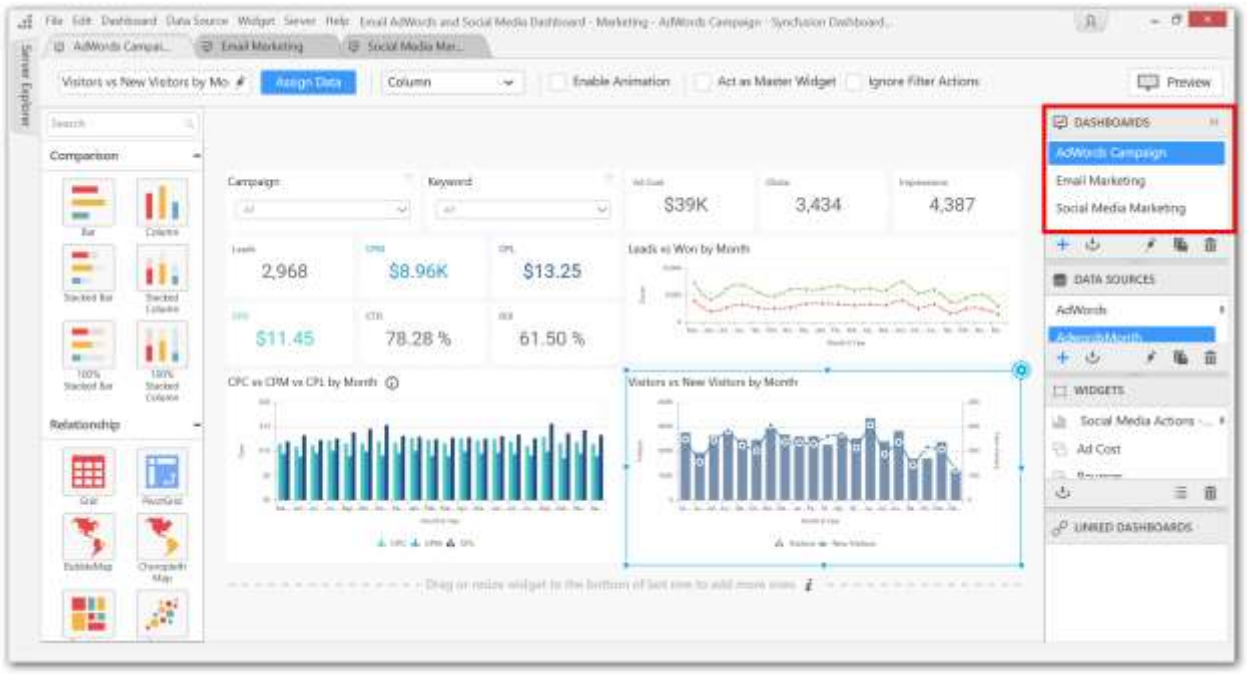
Now, the selected dashboard tab will be switched if it is already in loaded state. If it is in unloaded state the widgets will be loaded like shown in the following screenshot.



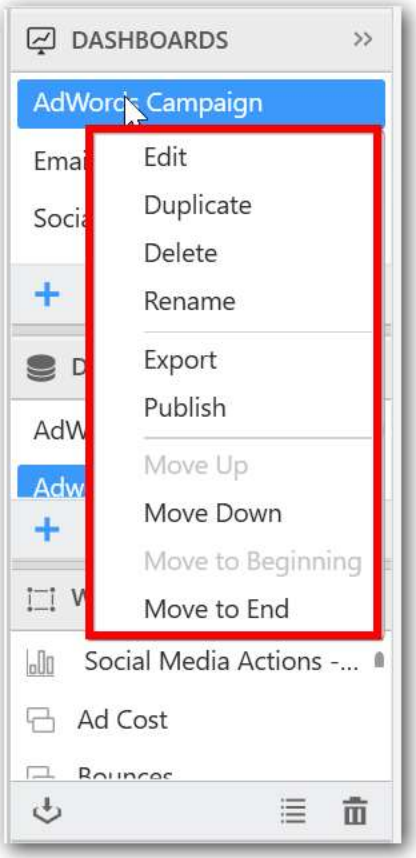
Duplicating a Dashboard

You can copy or duplicate a particular dashboard tab using the Duplicate option.

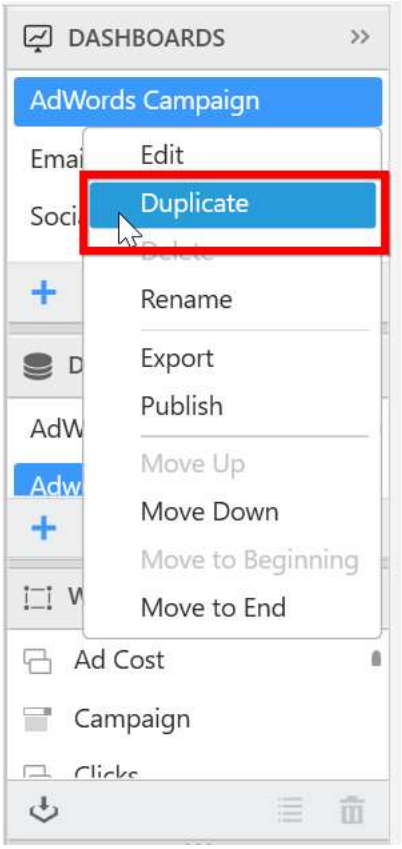
1. Select the dashboard tab you want to copy or duplicate in the Dashboards container.



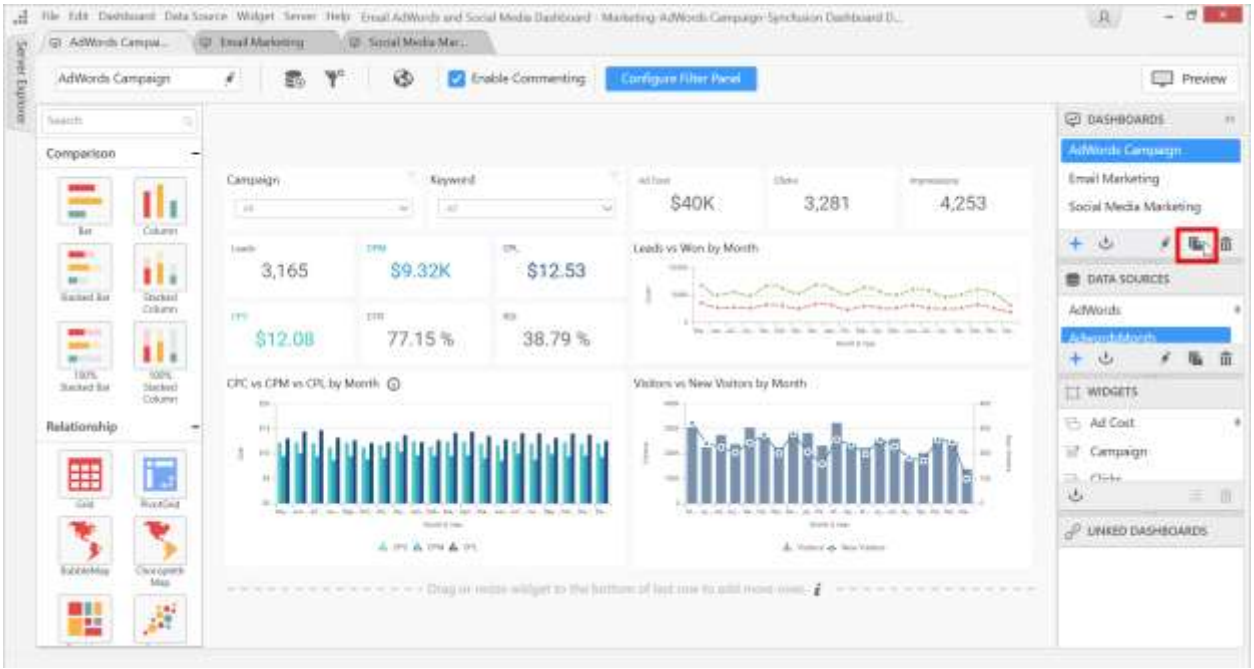
2. Right click on the widget name to open the context menu.



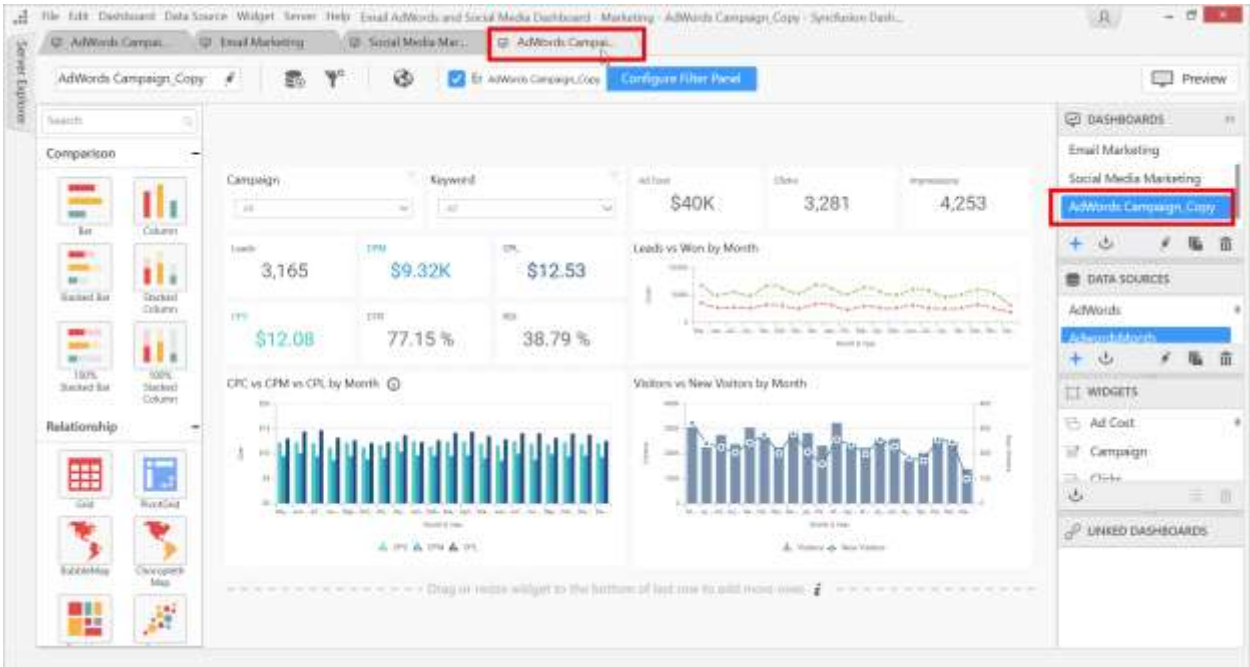
3. Select the Duplicate option in the menu.



You can also use the Duplicate dashboard icon for editing a particular dashboard tab.



This will create a copy of the existing dashboard tab that will be added as a new tab to the current dashboard report.



Note: The widgets in the copy dashboard will act as shared widgets.

Publishing Dashboard to Server

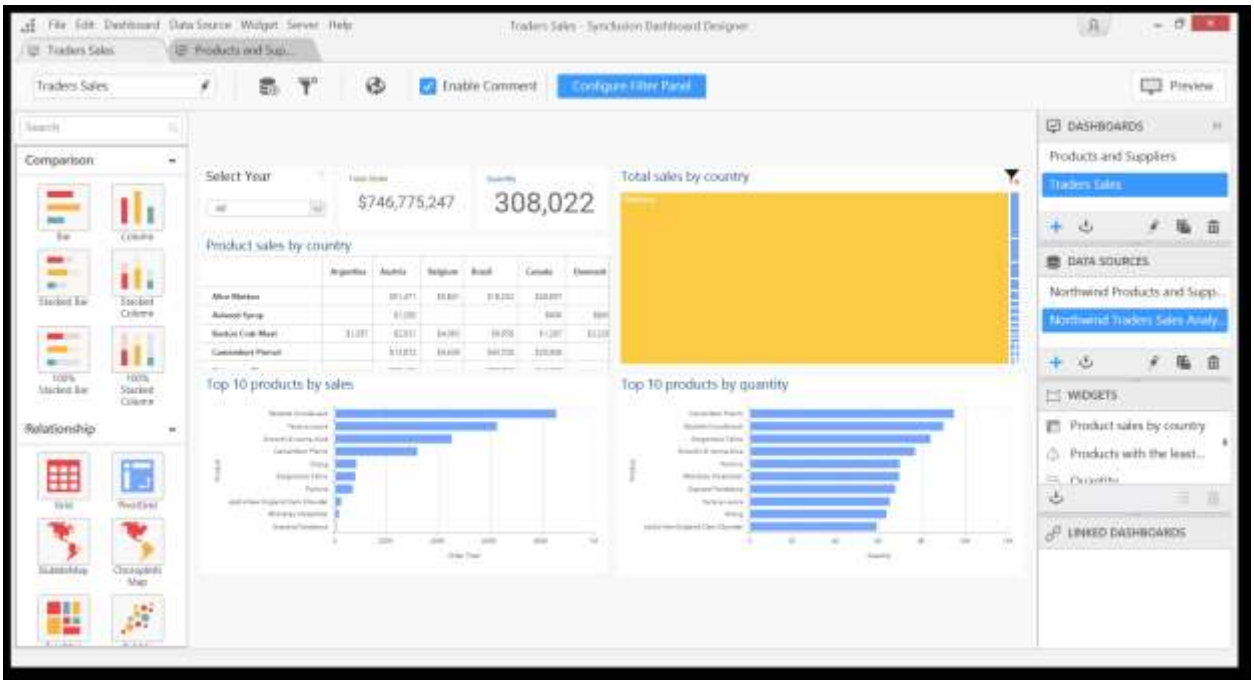
You can publish or update the created dashboard reports to the Dashboard Server. Refer to more details on the publishing from this section.

Sharing Widgets and Data sources in multi-tabbed Dashboards

Now you can use a same Data source or widget in multi-tabbed dashboards without duplicating or creating multiple instances of the data source or widgets.

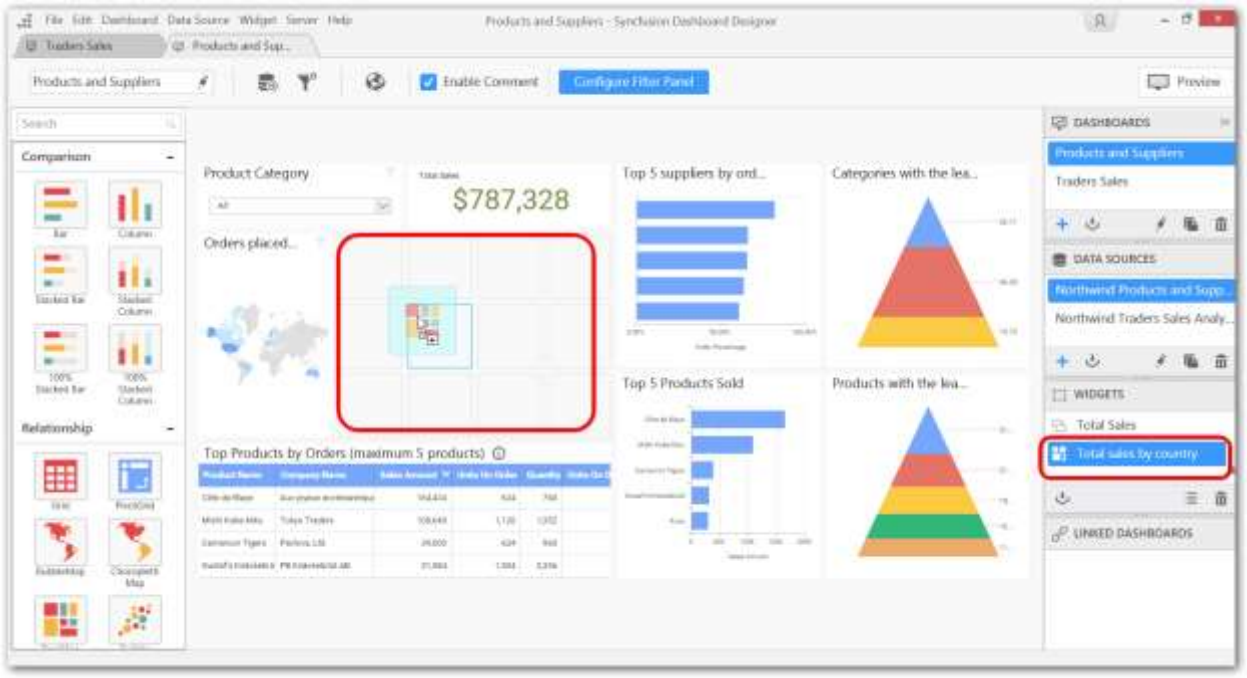
For example,

You have two dashboards in your Dashboard report.

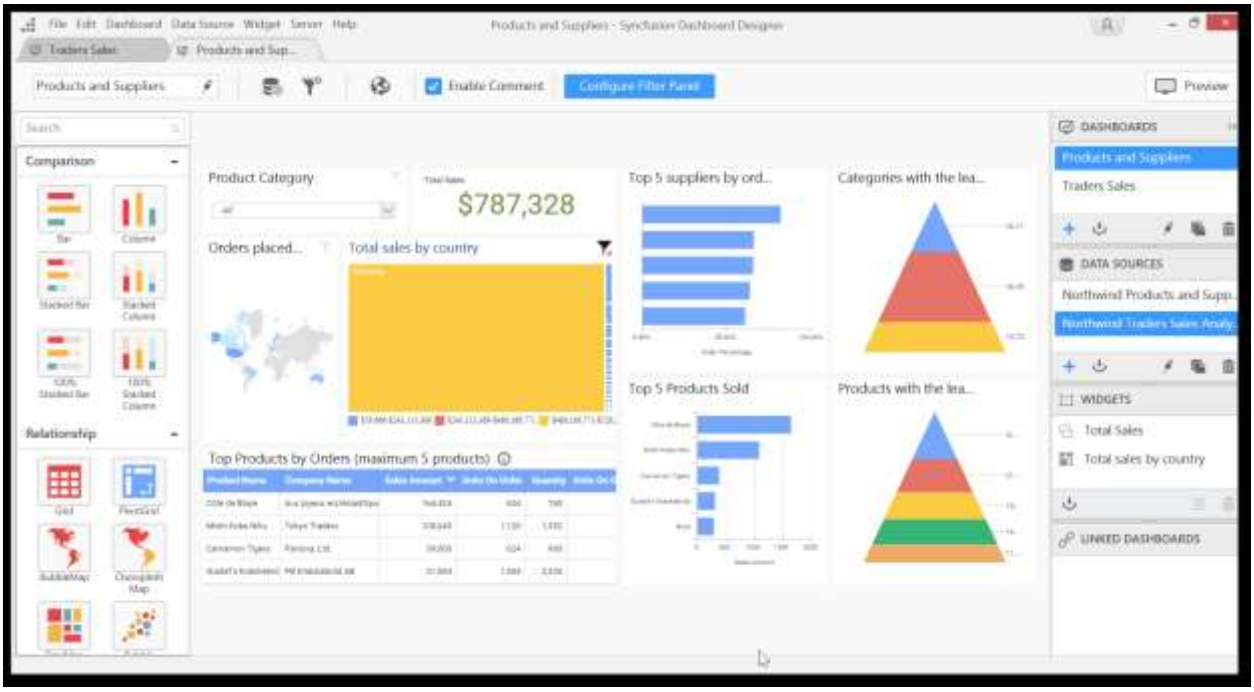


Total Sales by country widget is present in the **Traders Suppliers** dashboard and you want to use this widget in the **Products and suppliers'** dashboard.

Switch to the **Products and supplier's** dashboard drag the **Totals Sales by Country** widget from the WIDGETS container as shown in the below screenshot:

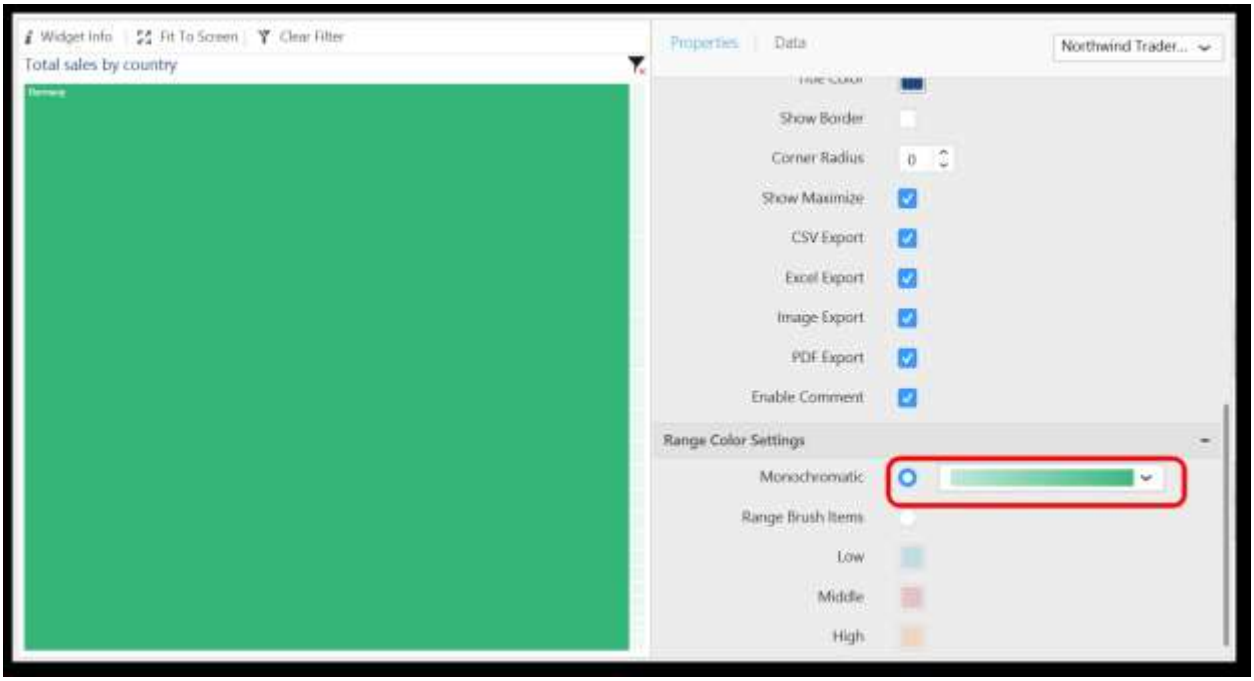


Now you shared the widgets between multiple dashboards without duplicating the widget.



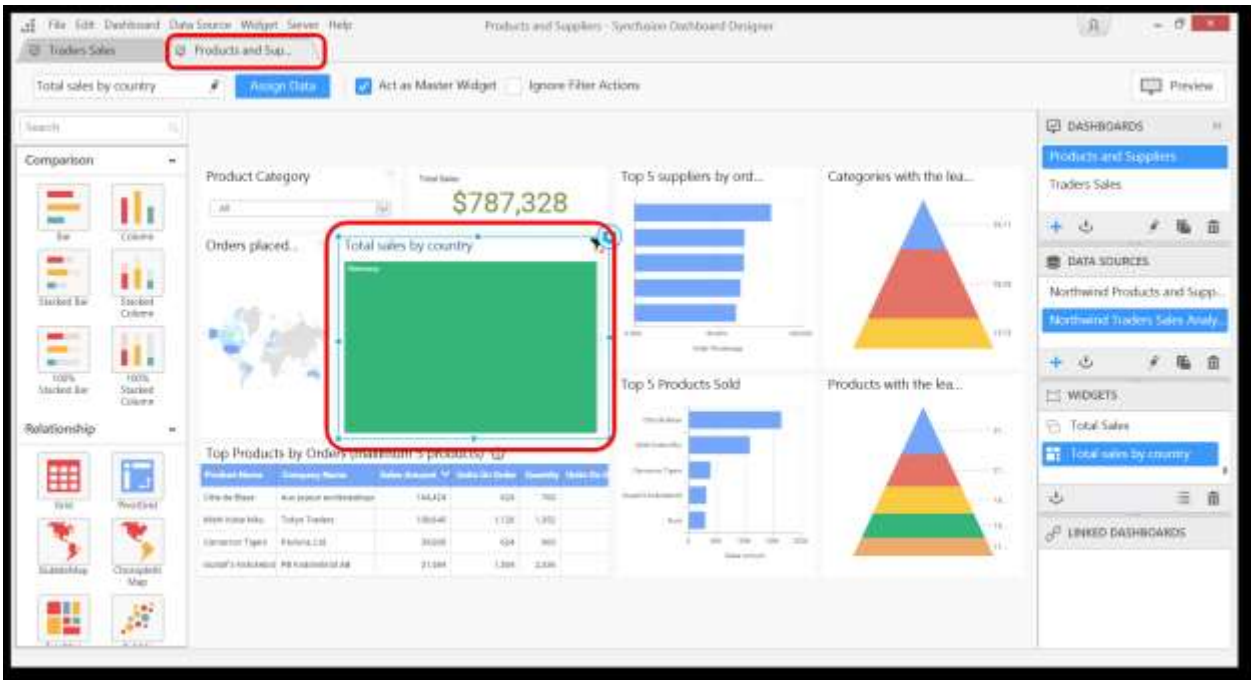
Changes in the widget will be reflected in the all dashboards if we change the properties of the shared widget.

For example, we are going to change the Range color settings of the widget.

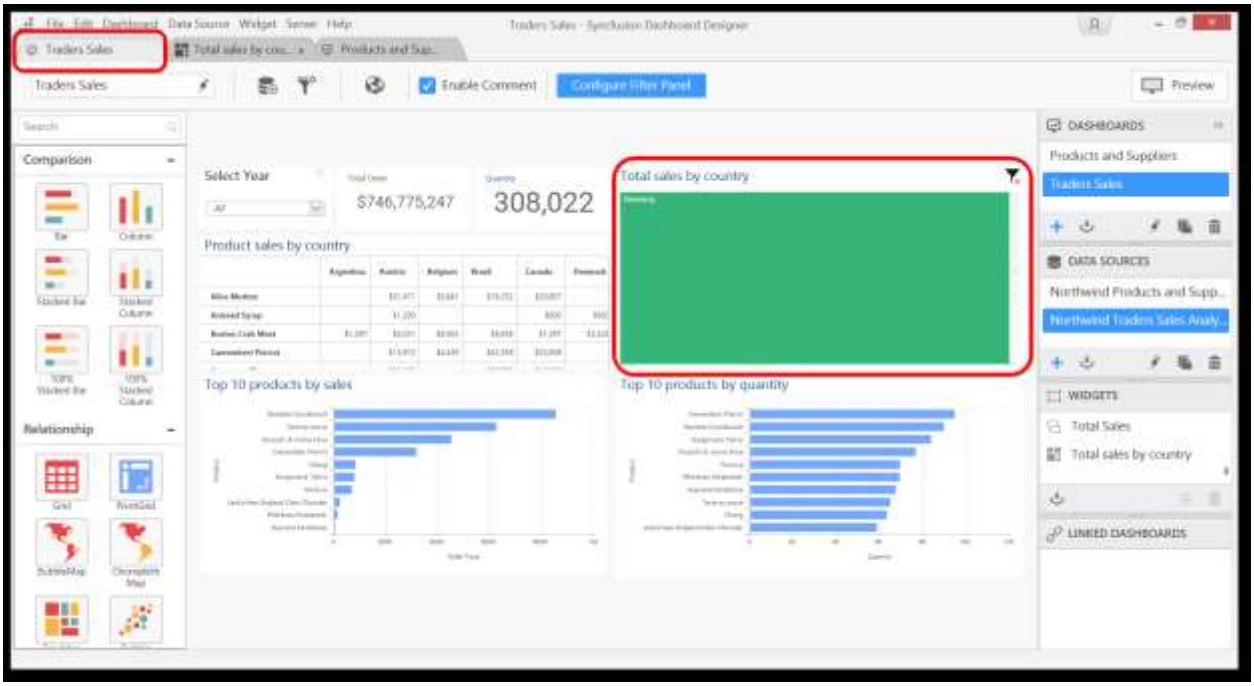


It will be reflected in both dashboards.

Products and supplier's dashboard:



Traders Suppliers dashboard:



The same shared behavior is applicable for data sources; You can use a single data source across multiple dashboard tabs.

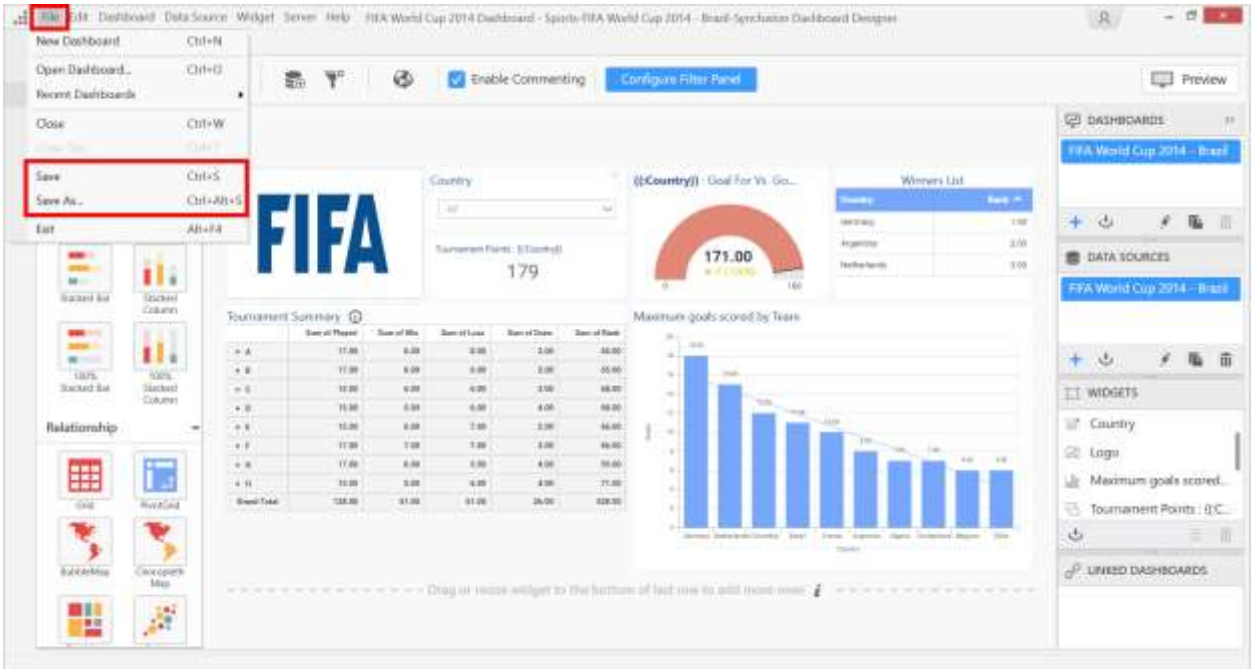
Saving a Dashboard

Syncfusion Dashboard File Format

Save the newly created dashboard in local or existing dashboard with same or different name in same or different location as `Syncfusion Dashboard(*.sydx)` formatted file.

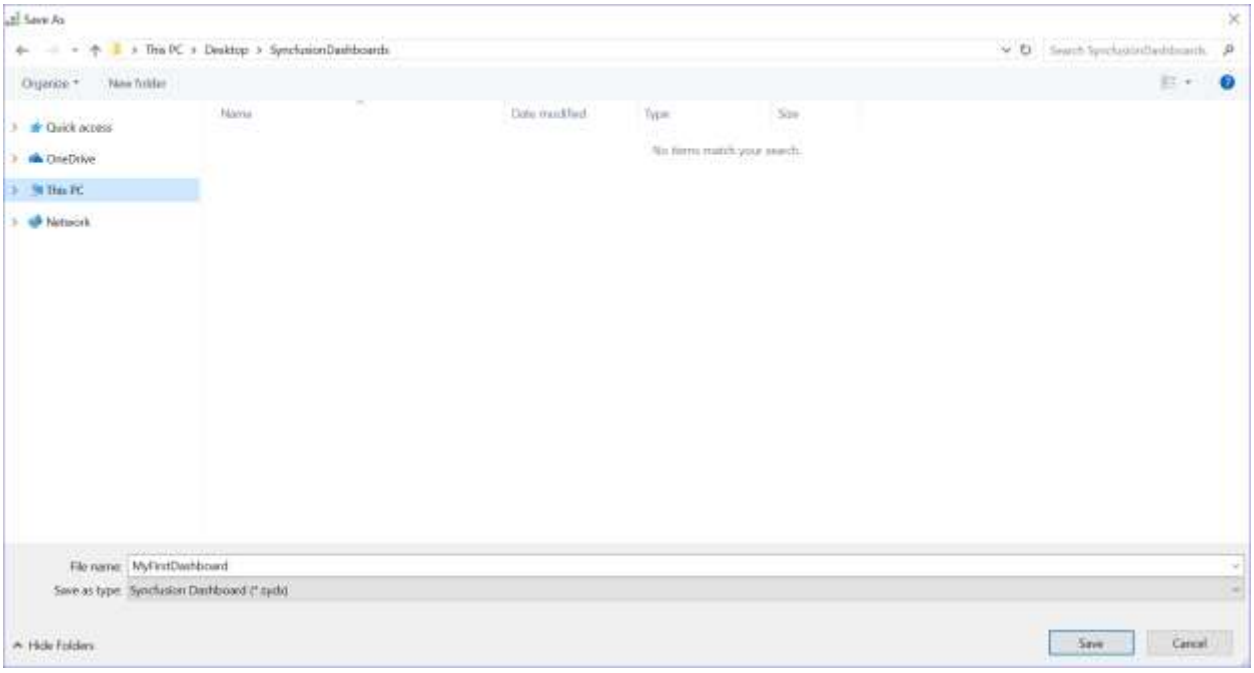
The Syncfusion Dashboard(*.sydx) file holds details about dashboard reports, widget and data source configurations, images, if any bounded to image control from local, map JSON files, if any bounded to map control from local, CSV, Excel or JSON file for file based connection types like CSV, Microsoft Excel and JSON respectively.

Saving operation can be handled through the File Menu items such as, Save and Save As…



Save as option

When Save as option is used, the save file dialog will be displayed. The keyboard shortcut Ctrl + Alt + S can be used for save as option.



The dashboard will be saved as SYDX file in the selected path with the mentioned name in the File save dialog.

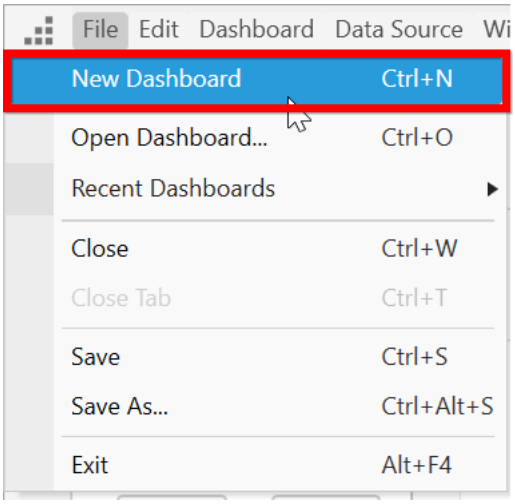
Save option

When Save option is used, the existing dashboard will be updated. If saving the dashboard for the first time, this option will prompt the Save As… dialog even on clicking the Save menu item.

The keyboard shortcut Ctrl + S can be used for save as option.

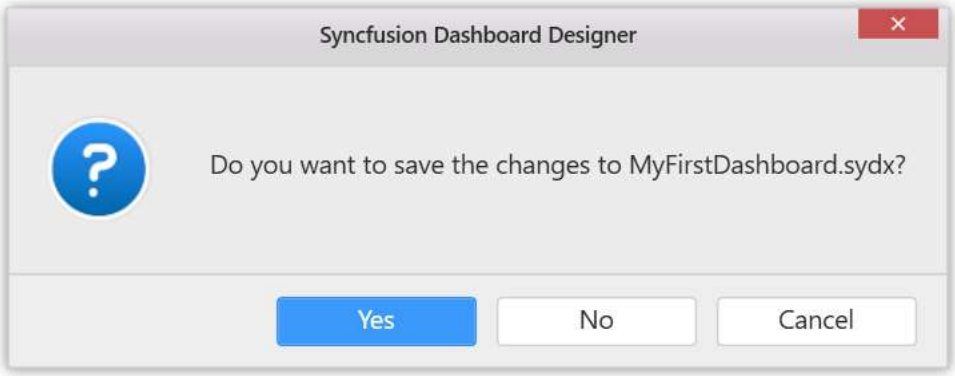
Ignore the file modification

If you want to ignore the changes made to the dashboard report you can click on the New Dashboard option from the File menu and the following message will be shown.

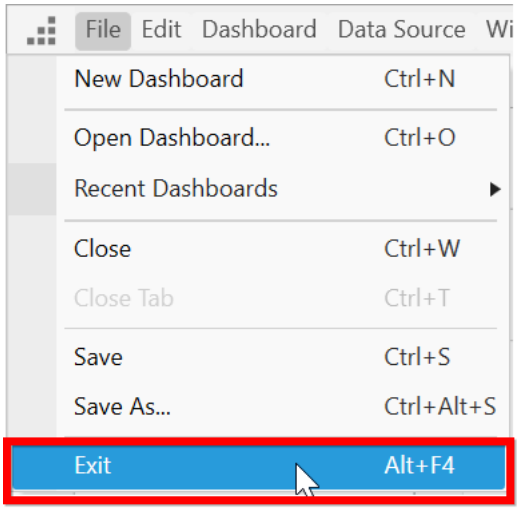


Note: You can also use the Keyboard shortcut Ctrl + N for new dashboard.

You can click on No option in the message dialog and the changes made to the existing dashboard will be ignored.



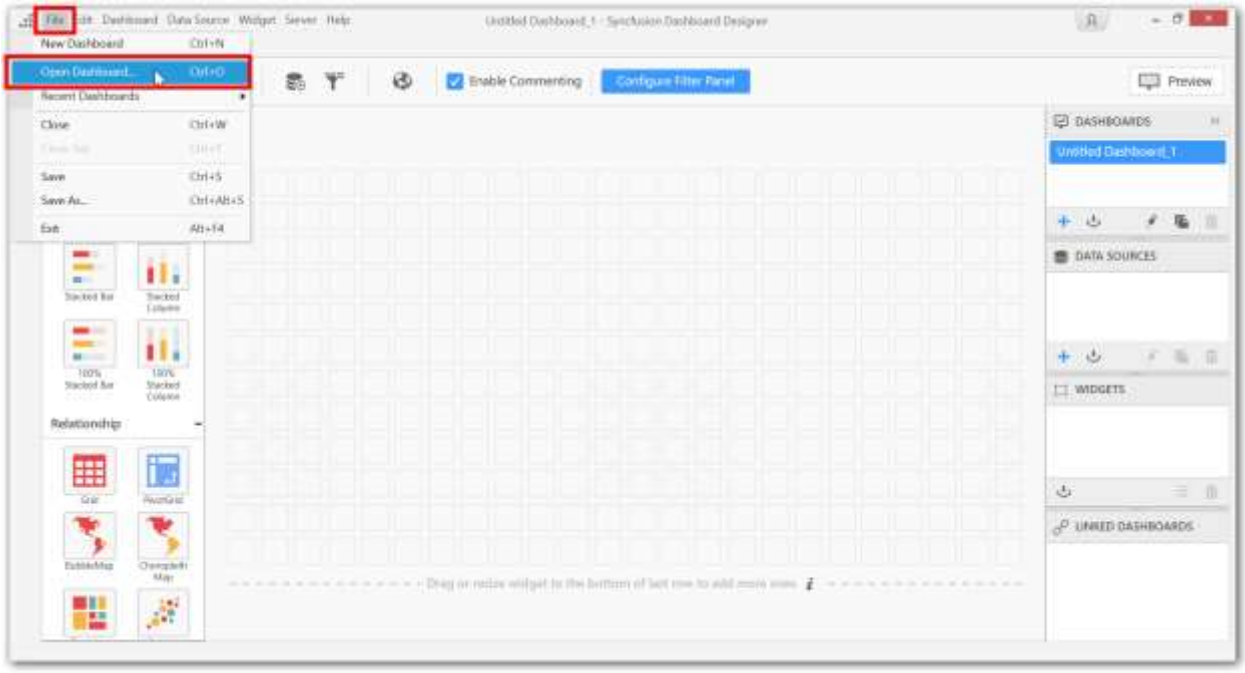
If you need to quit the Dashboard designer application you can click on the Exit option from the File menu. The keyboard shortcut is Alt + F4.



Opening a Saved Dashboard Report

Opening from local

You can use the existing dashboard by clicking the Open Dashboard through the File menu and select the required existing dashboard.The keyboard shortcut for open dashboard option is Ctrl + O.



You can also open the SYDX files saved in your local by double clicking the file, it will launch the dashboard designer application and the selected file will be loaded.

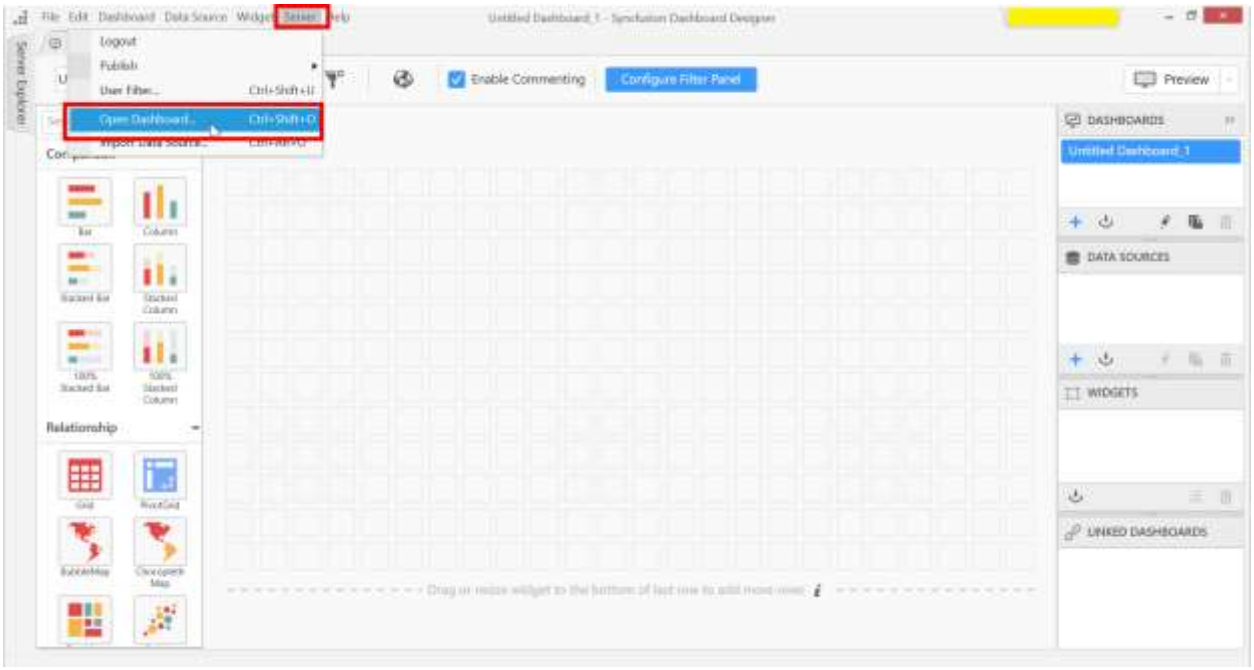
Opening from recent list

You can also open a dashboard from the recent dashboards list. Please refer this section for more details.

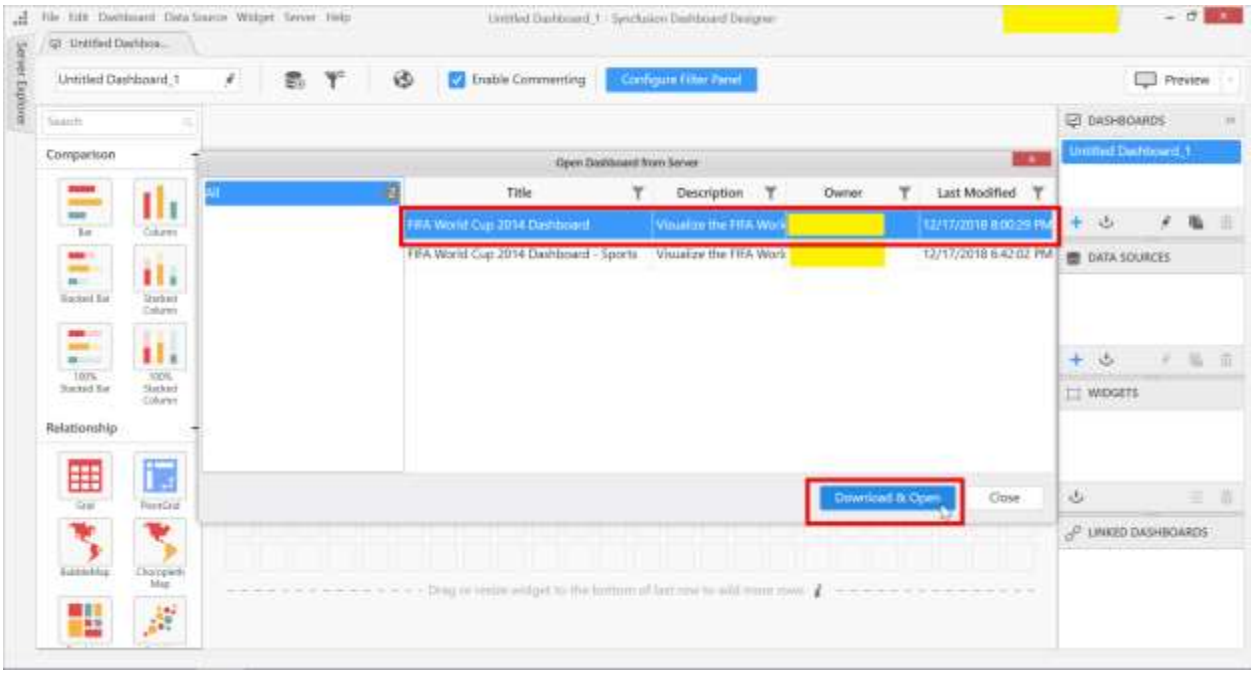
Opening from Dashboard Server

You can open a published dashboard server from the dashboard designer by using this steps. Also make sure you have logged into your dashboard server account from designer application.

1. Click on the Open Dashboard menu item under the Server menu.



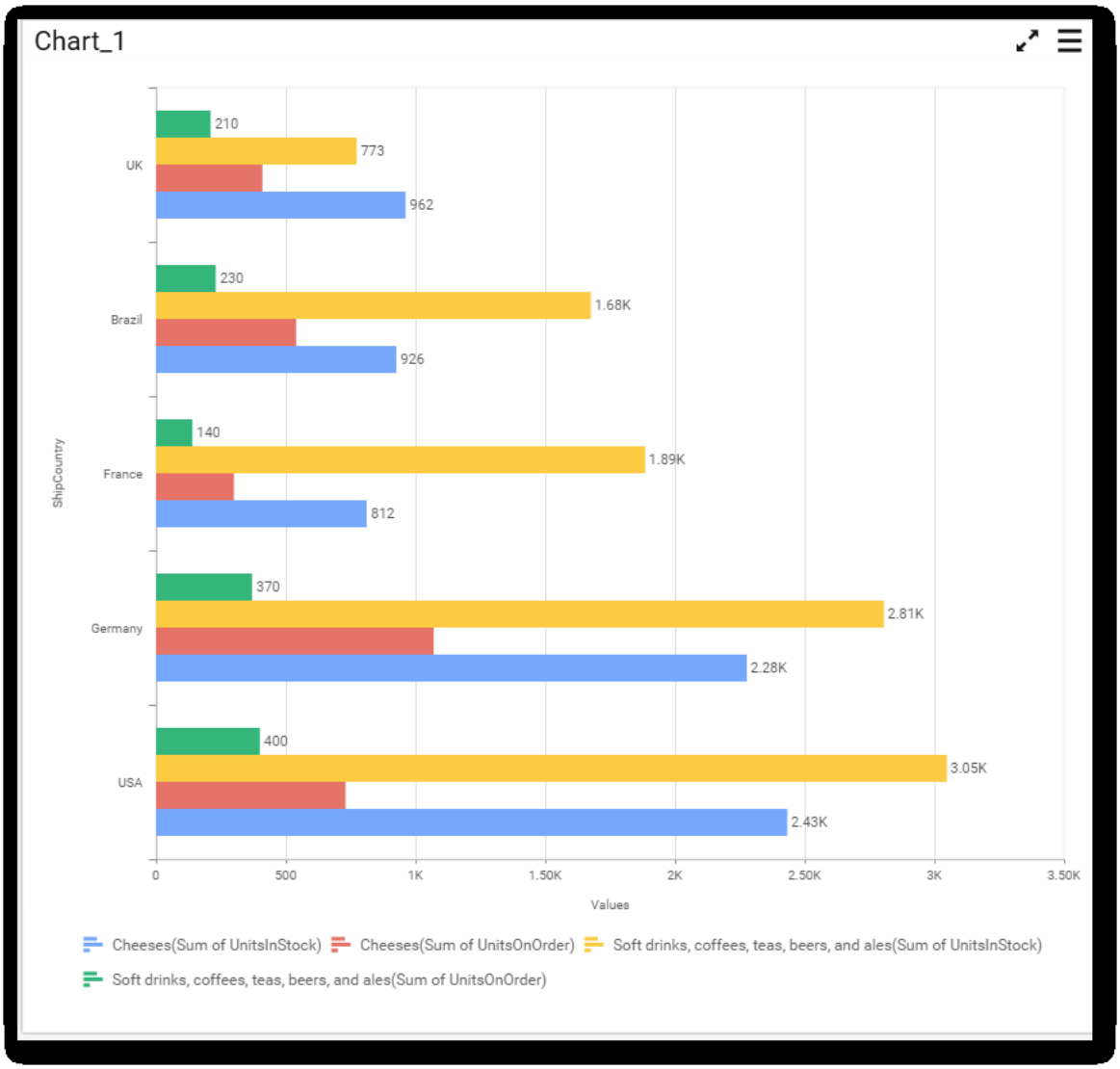
2. The list of published dashboards will be shown and select a dashboard you want to open in designer and click on Download & Open option.



Configuring and Formatting Dashboard Widgets

Bar Chart

Bar Chart allows you to compare values for a set of unordered items across categories through horizontal bars ordered vertically.

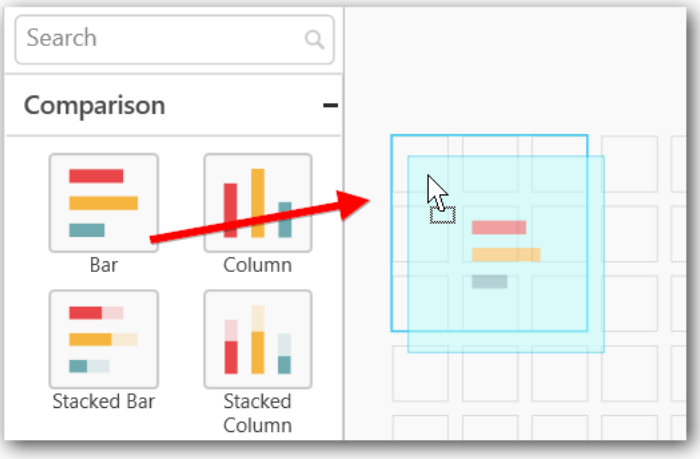


How to configure flat table data to Bar Chart?

Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure column or expression column that you would like to analyze can be dropped into Values(s) block. The dimension column that you would like to categorize the measure column, can be dropped onto Column(s) block. If you would like to categorize based on a series column, then the respective dimension column can be dropped onto Row(s) block in addition. These blocks are composed into Data pane.

Follow the steps to configure data into bar chart widget.

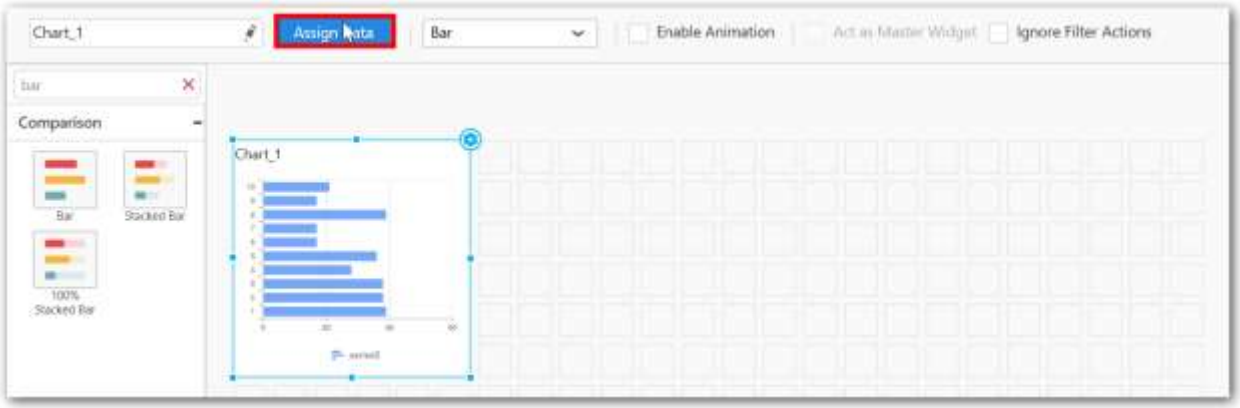
Drag and drop the bar chart widget into canvas and resize it to your required size.



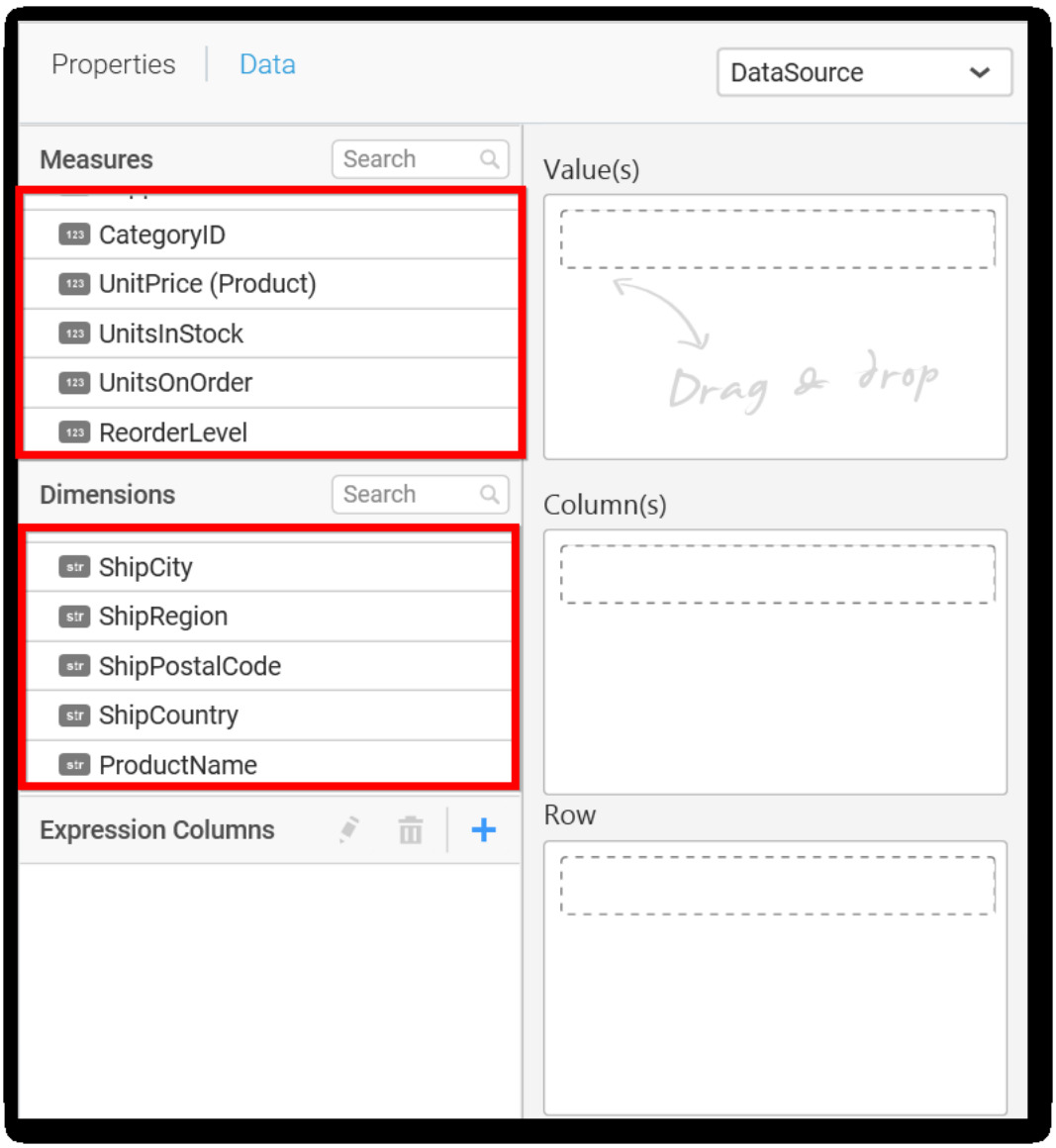
Connect to data source.

Focus on the bar chart widget.

Click on Assign Data button.

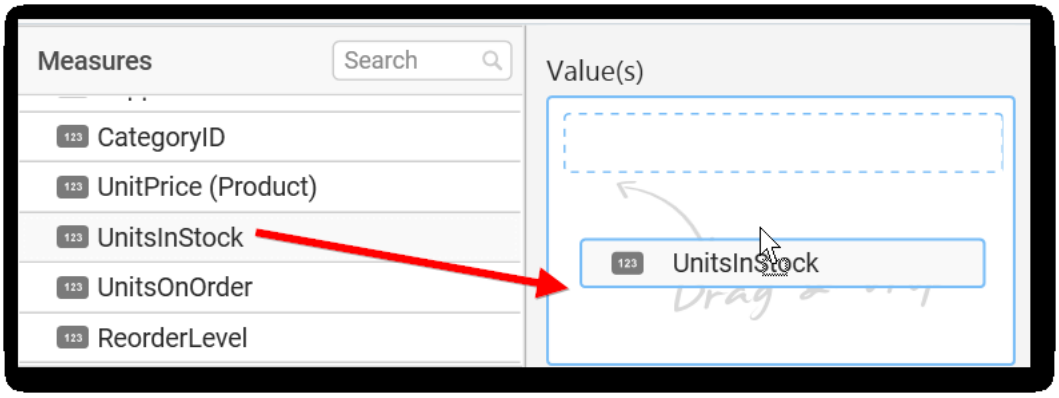


The data pane will be opened with available measures and dimensions from the connected data source

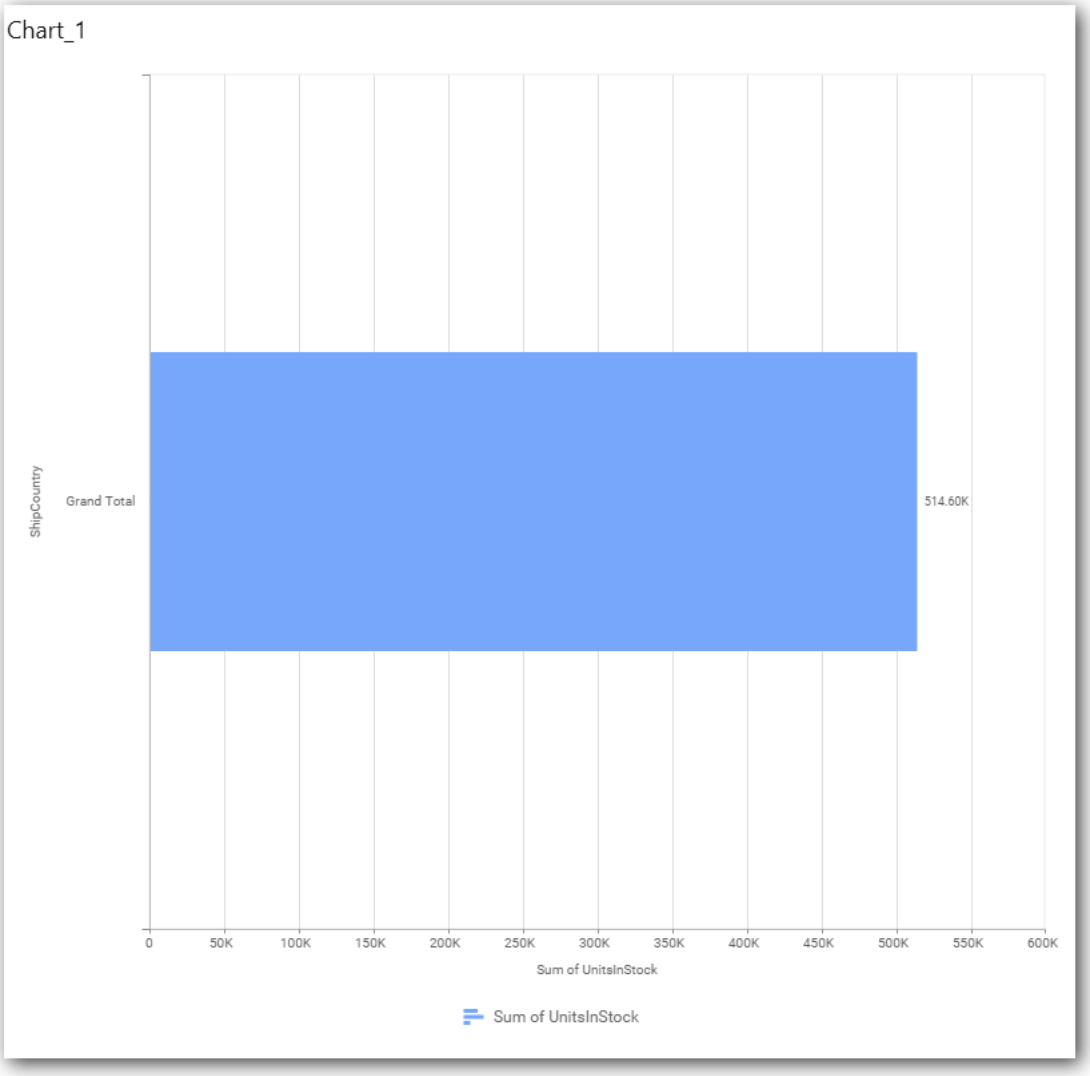


Assigning Value(s)

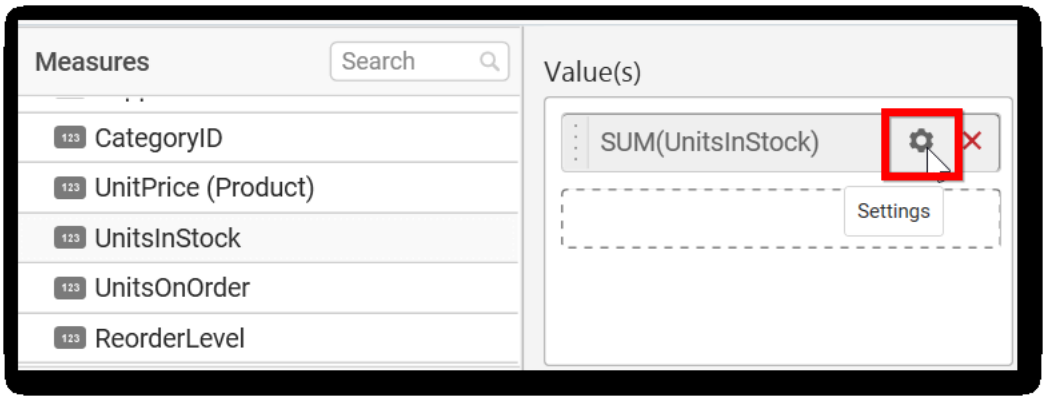
Drag and drop the Measure into Value(s)



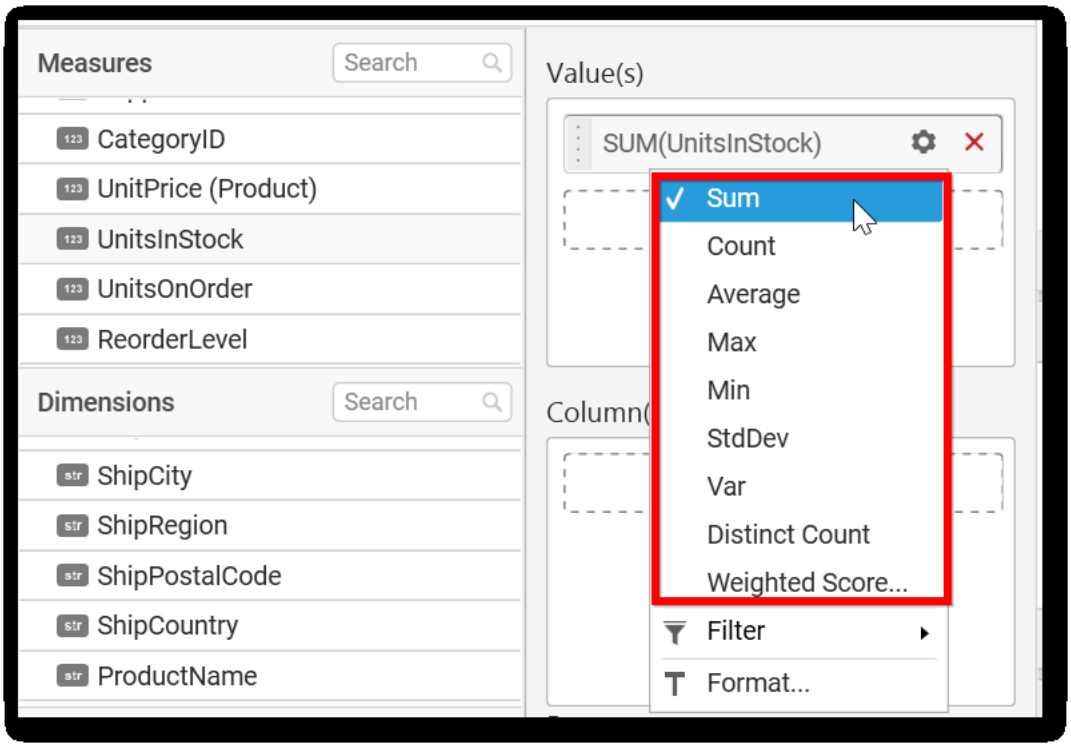
Now the chart will be rendered like this.



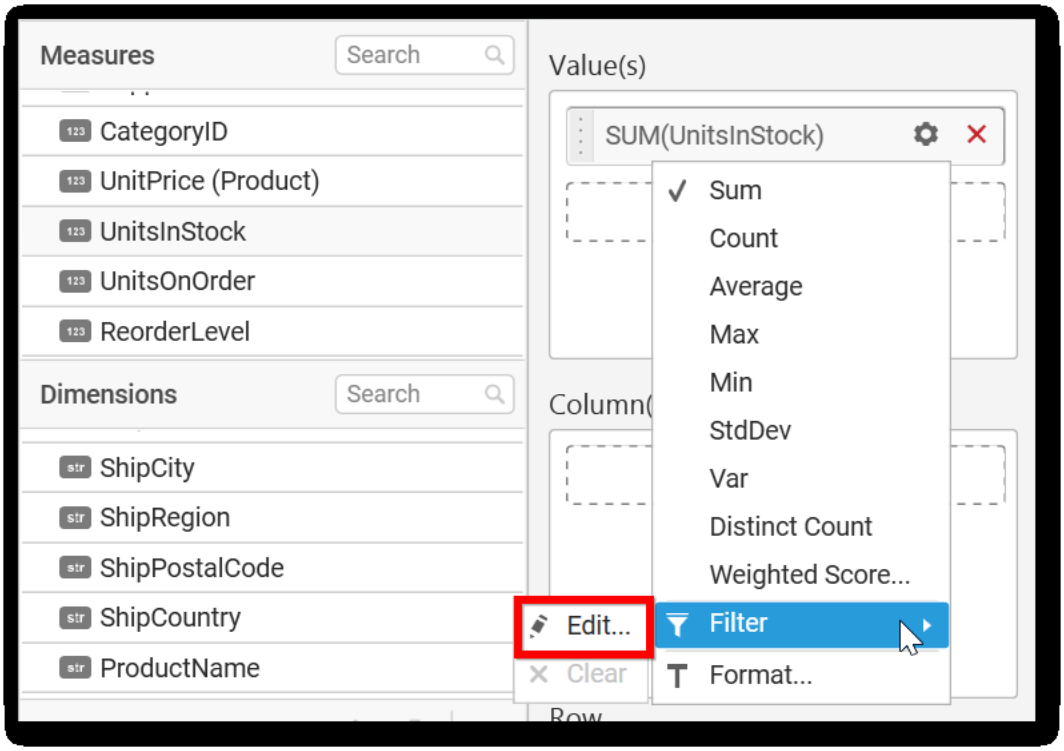
You can change the summary type of the value by clicking on settings option



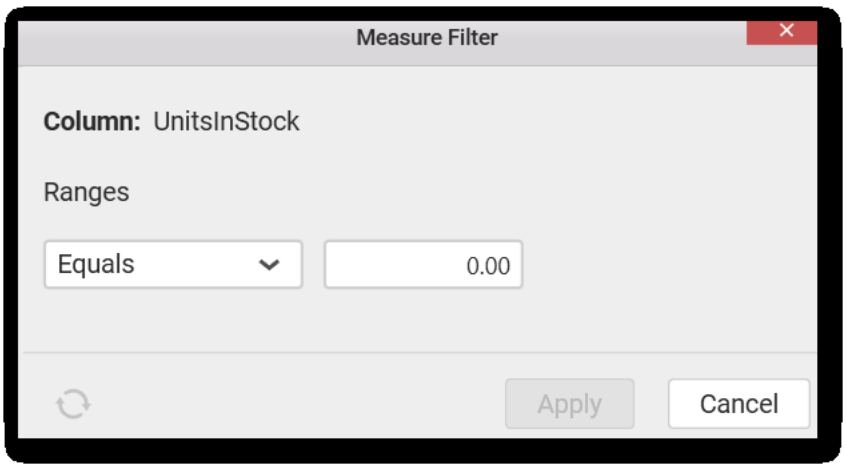
Select the required summary type from list.



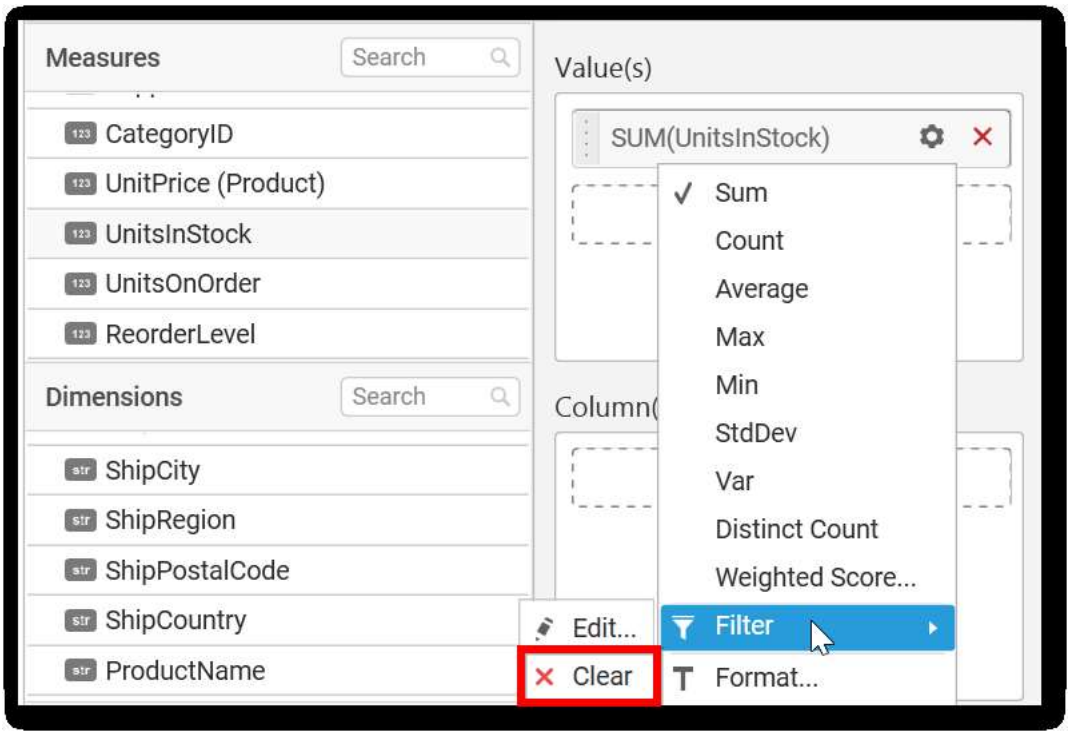
You can select what data to be displayed by choosing filter option



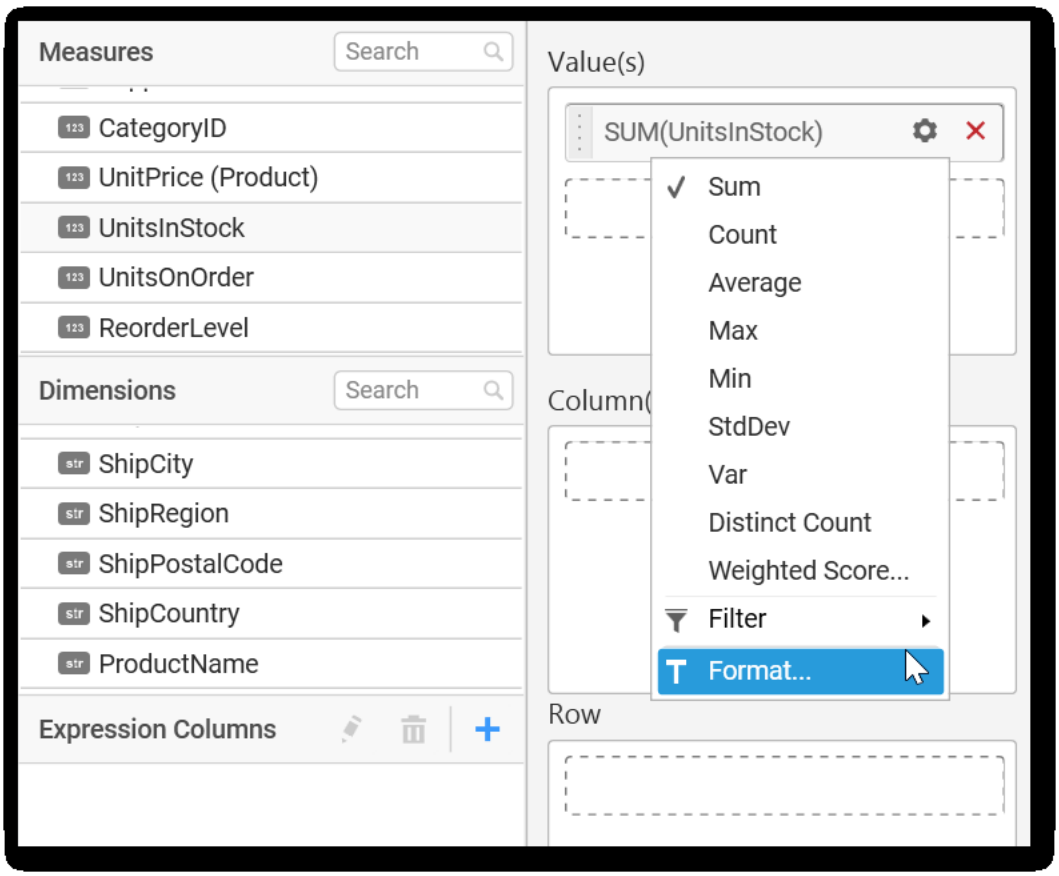
The Measure filter option will be shown and you can choose the filter condition and apply the condition value.



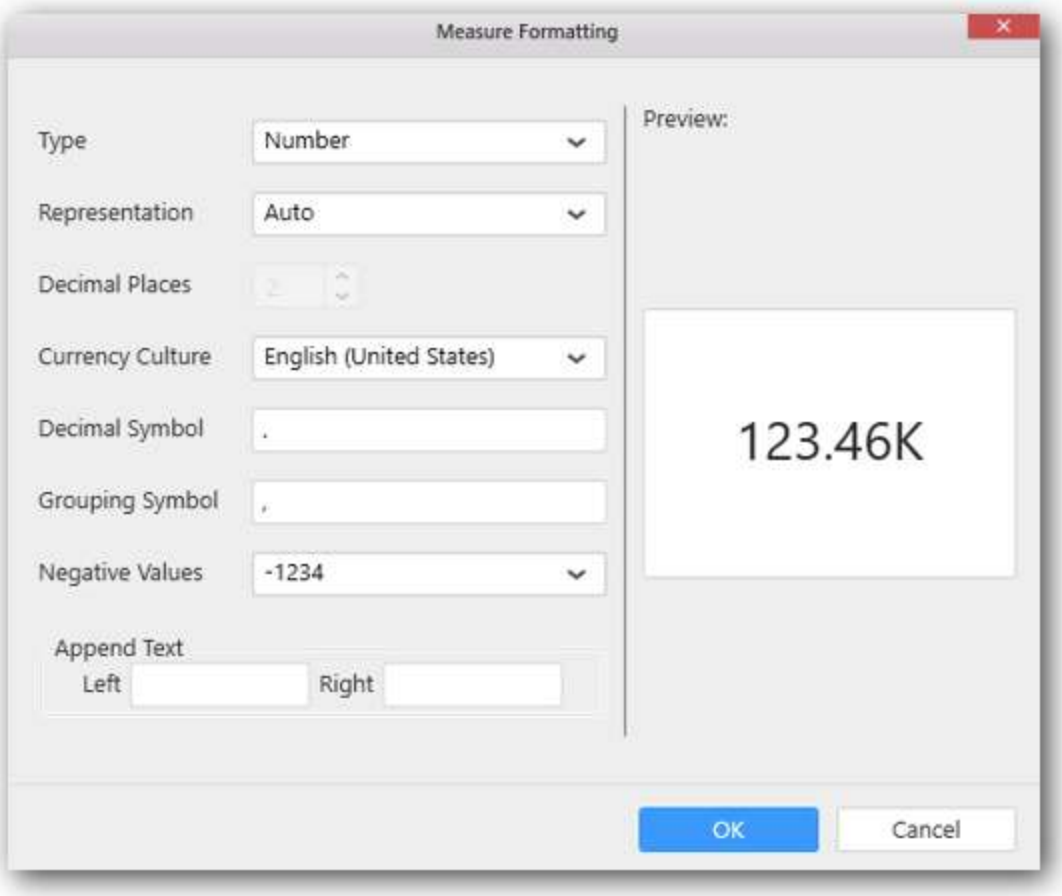
You can clear the filter.



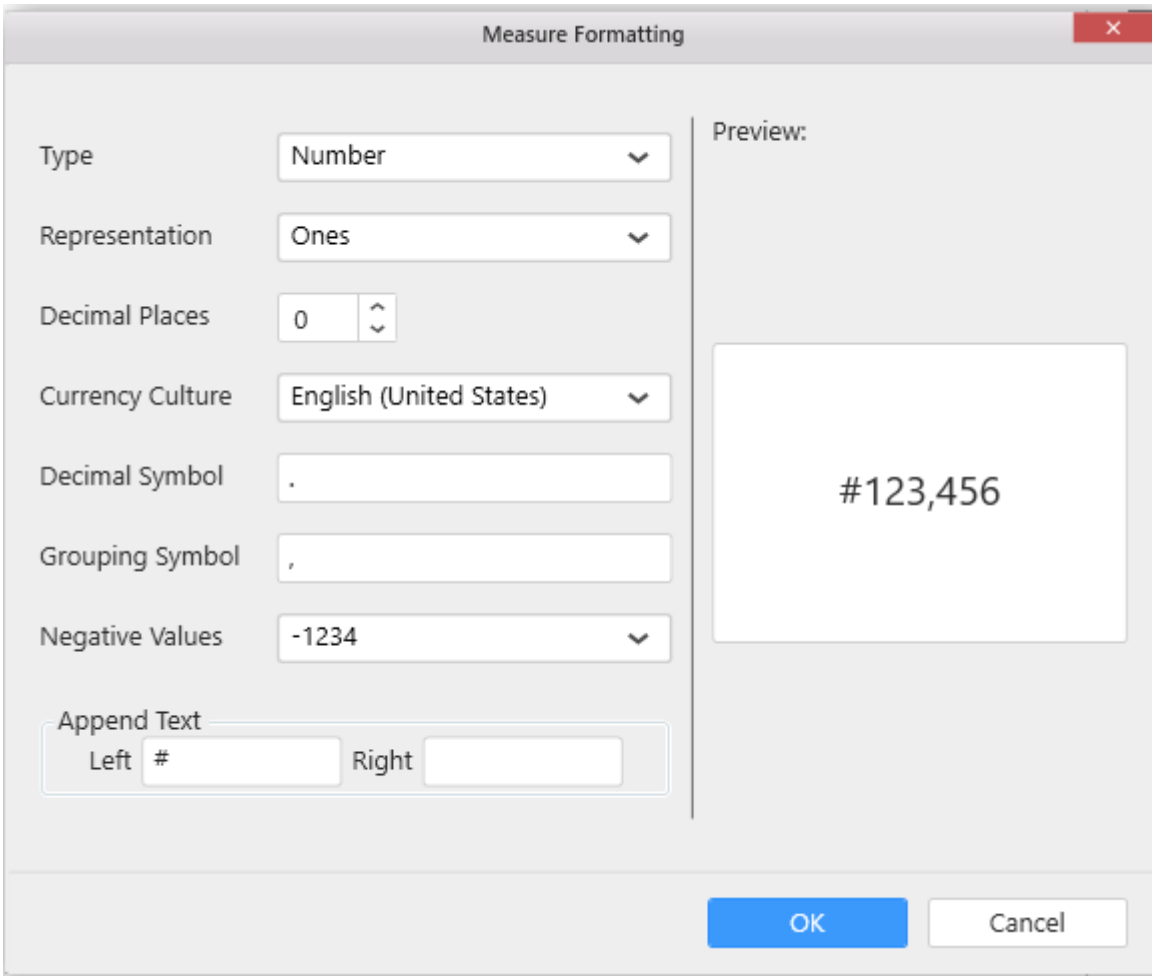
You can Format the value.



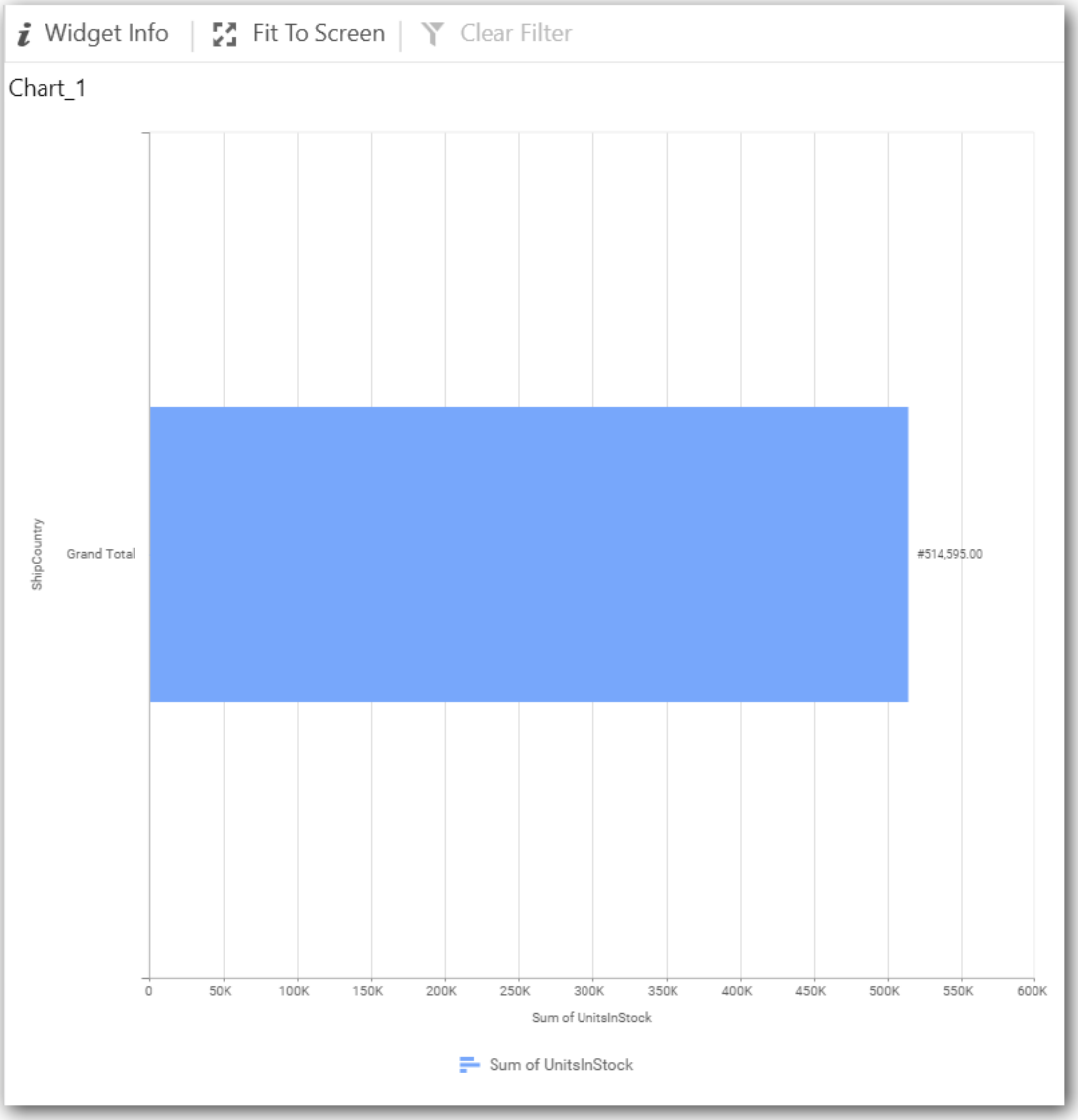
The format options will be shown.



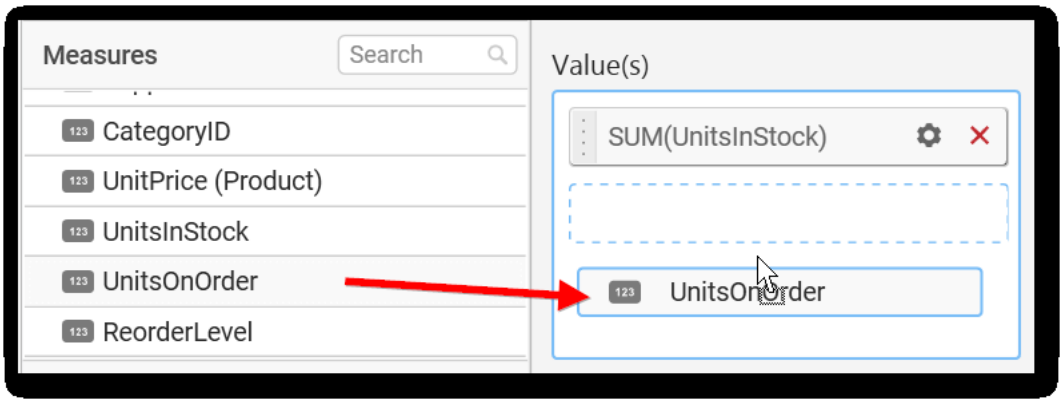
Choose the options you need and click OK.

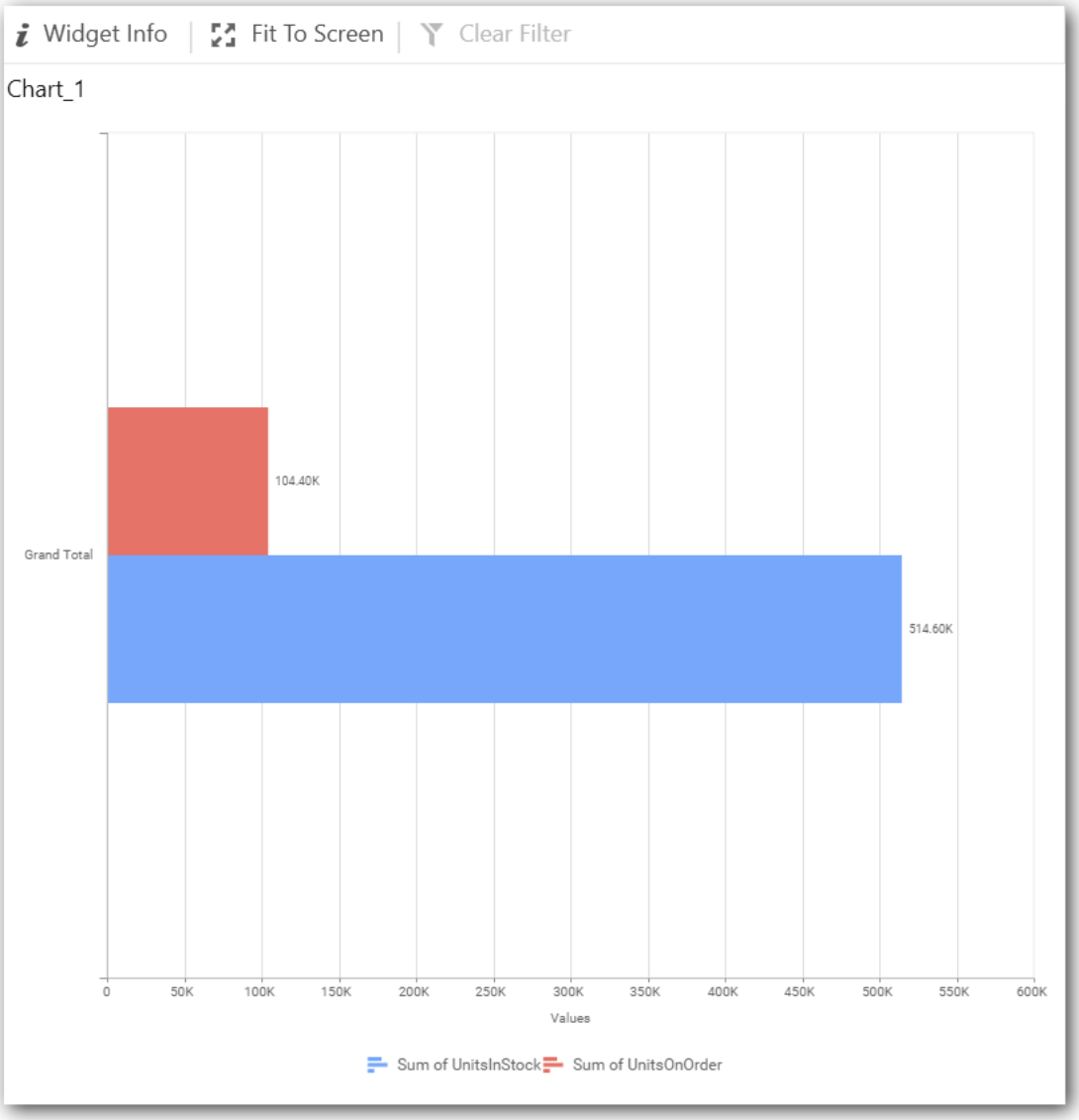


Now the Chart will be rendered like this.



You can add more number values by drag drop the Measures into Value(s) field.

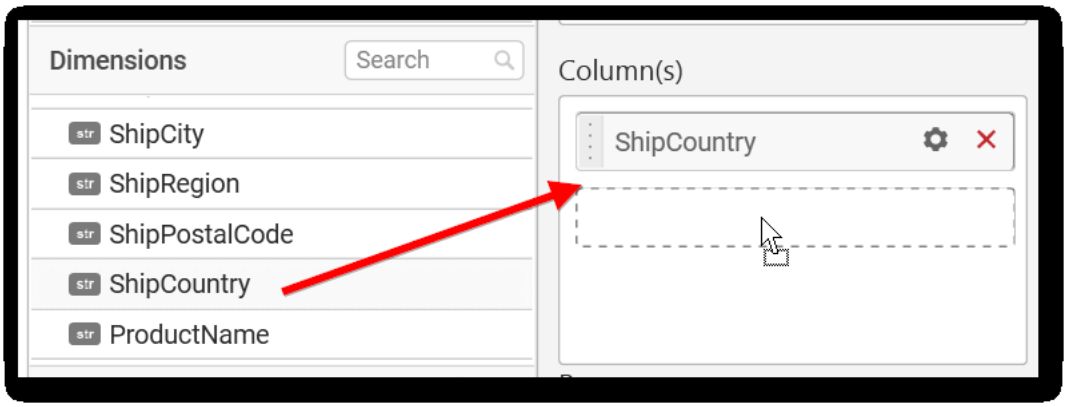


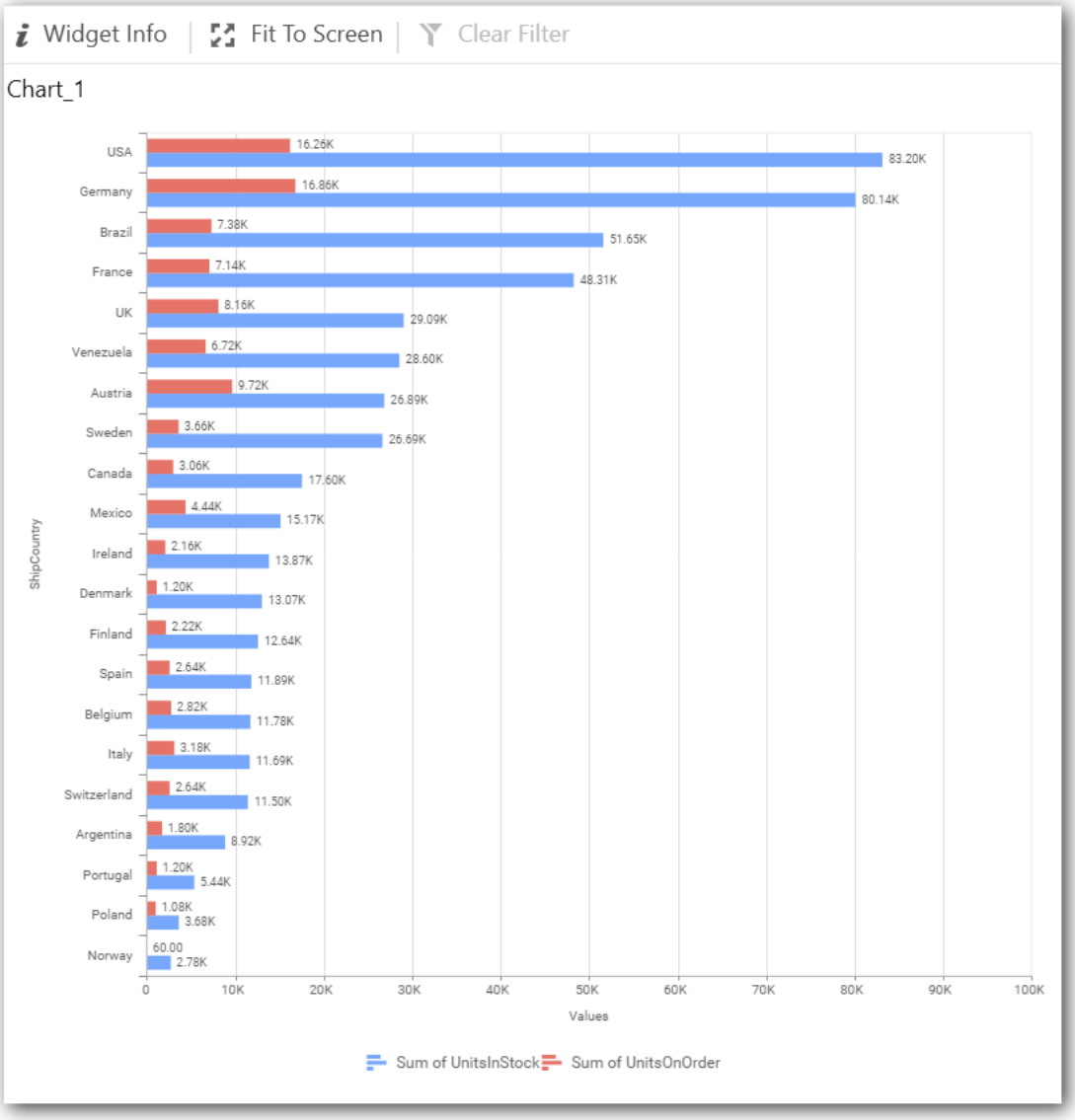


You can also add Dimensions and Columns to Value(s).

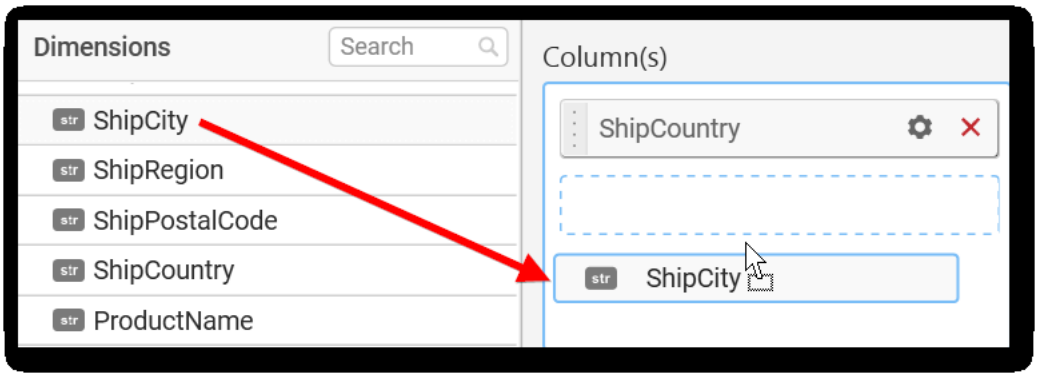
Assigning Column(s)

You can add the Dimension into Column field by drag drop.

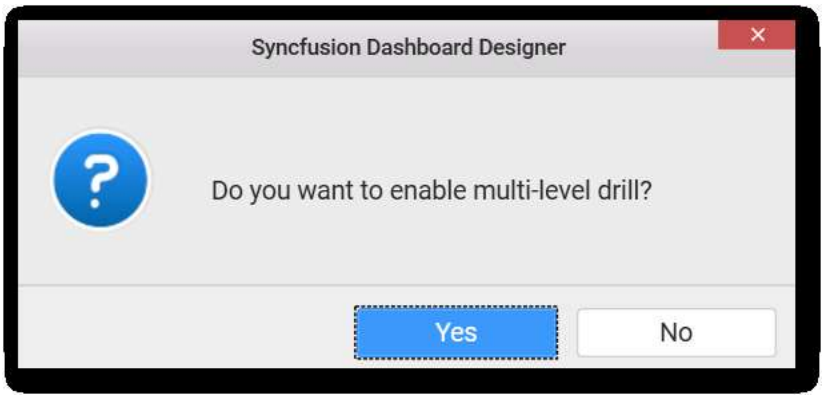




You have option to add more than one Column Value.

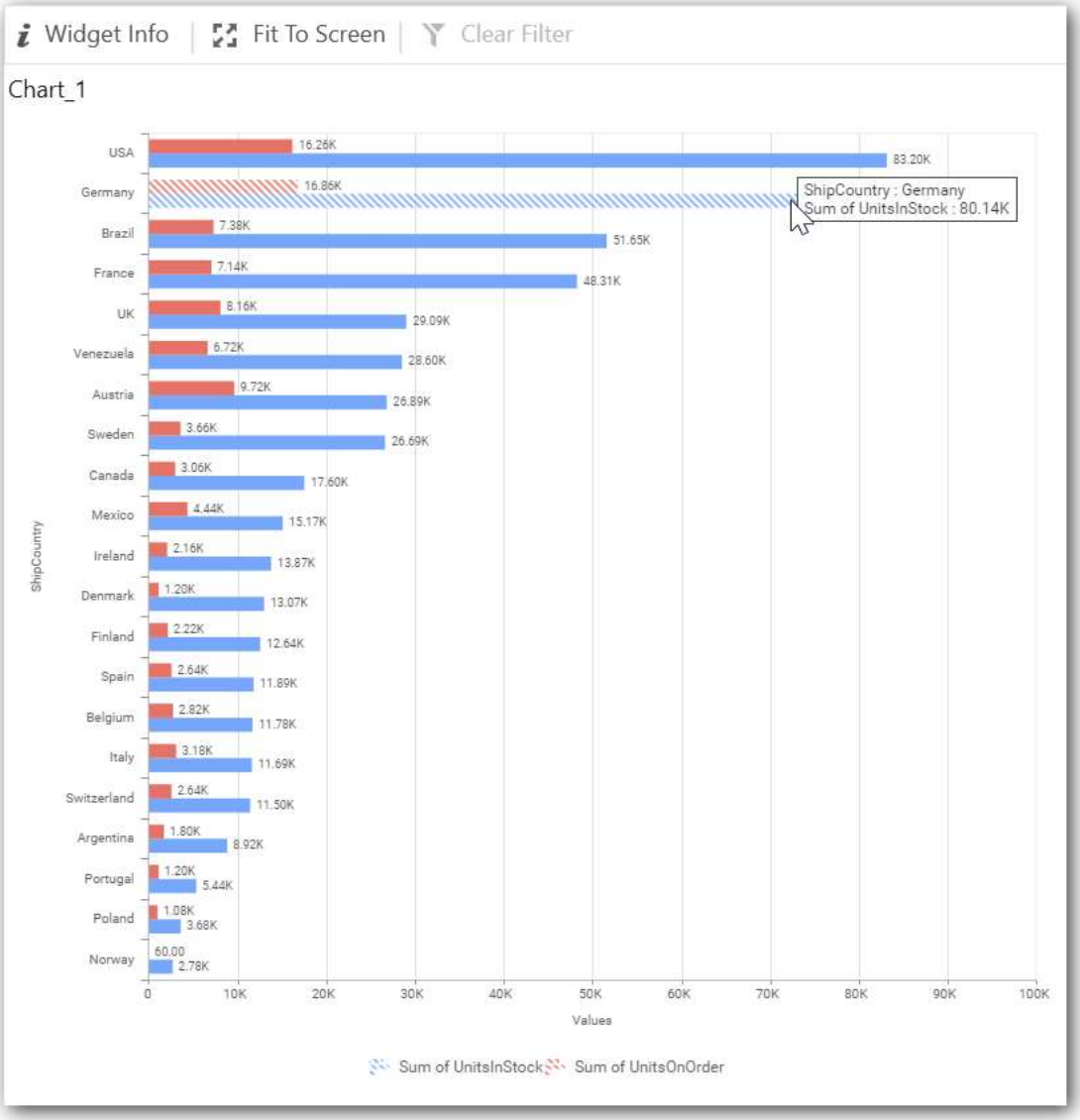


The following alert message will be shown.

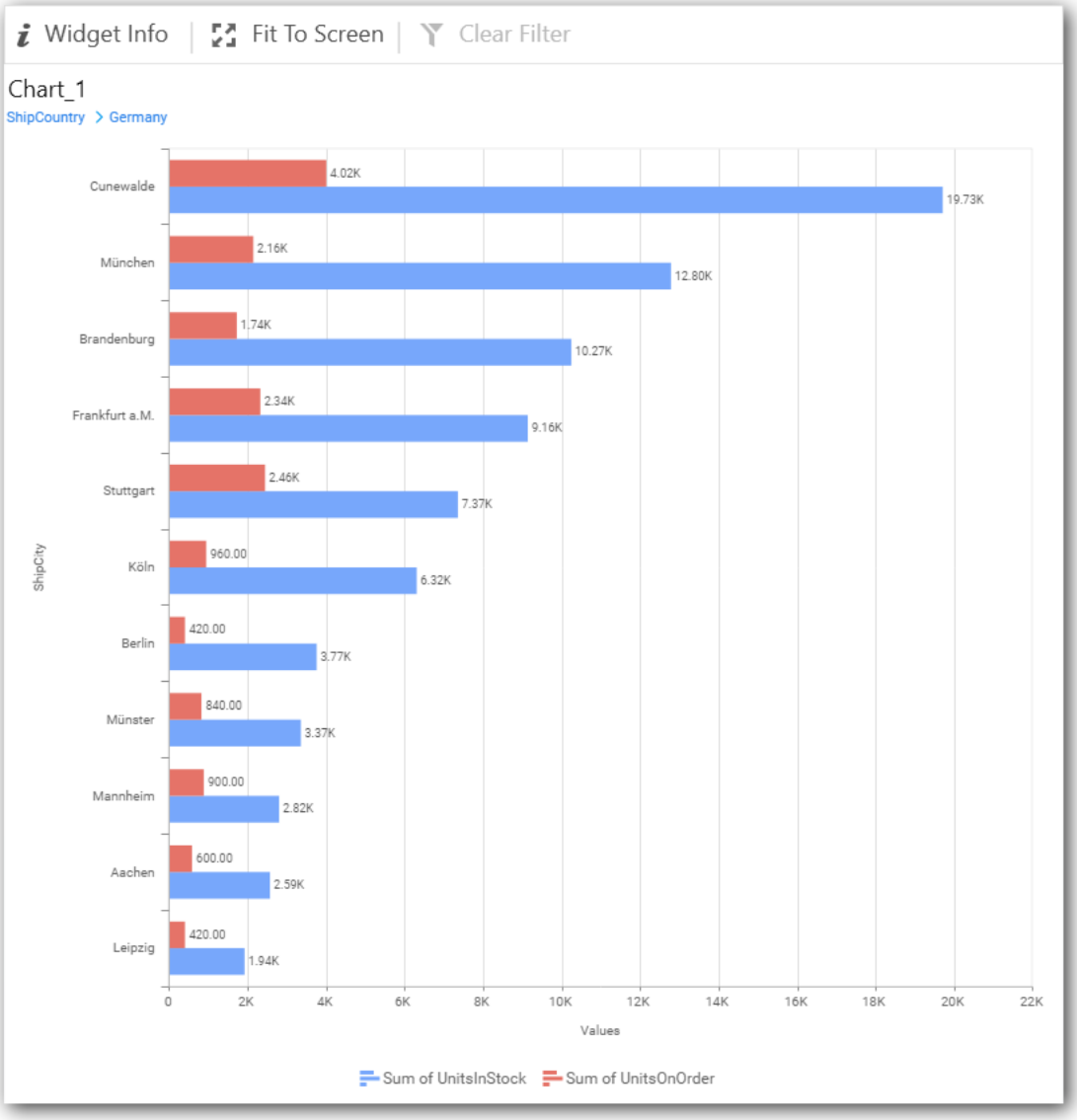


If you choose Yes Drill down option will be enabled.

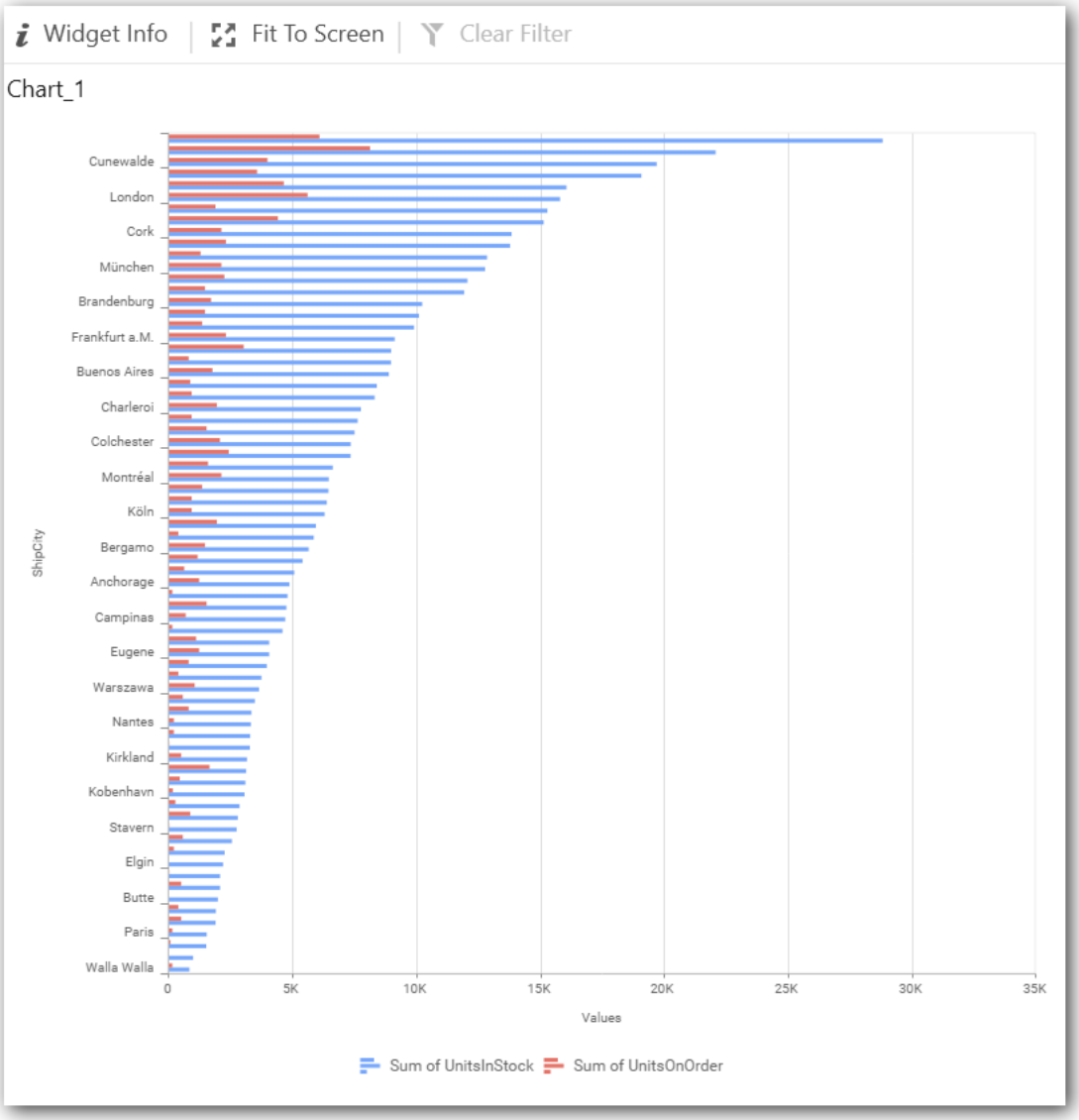
You can drill down the chart by clicking on the chart.



The drilled view of the chart is follows.

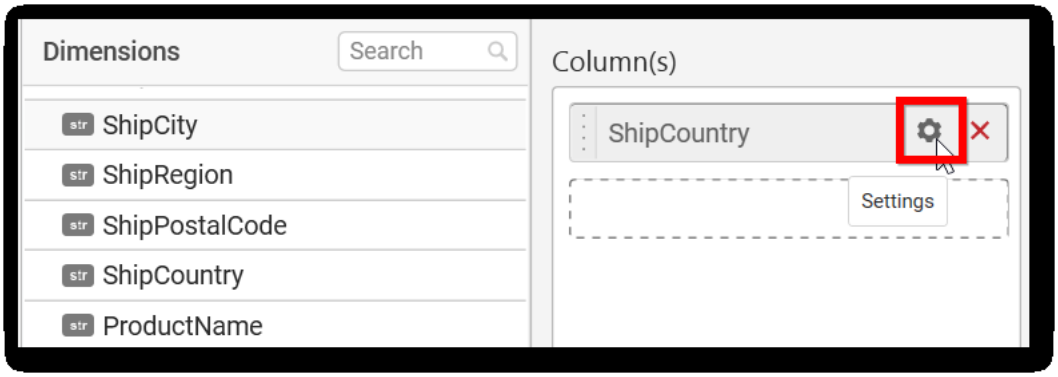


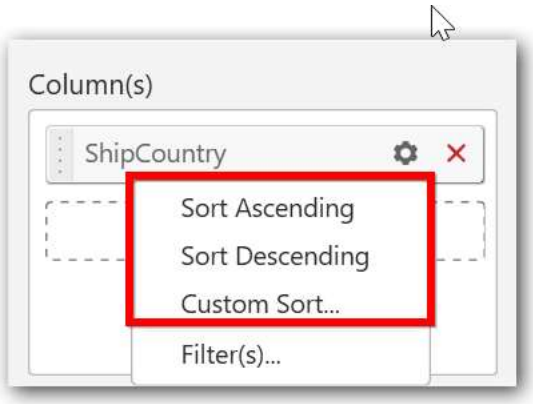
If you click No the new Dimension value will replace old value.



You can also add Measures and Expression columns into Column(s) field.

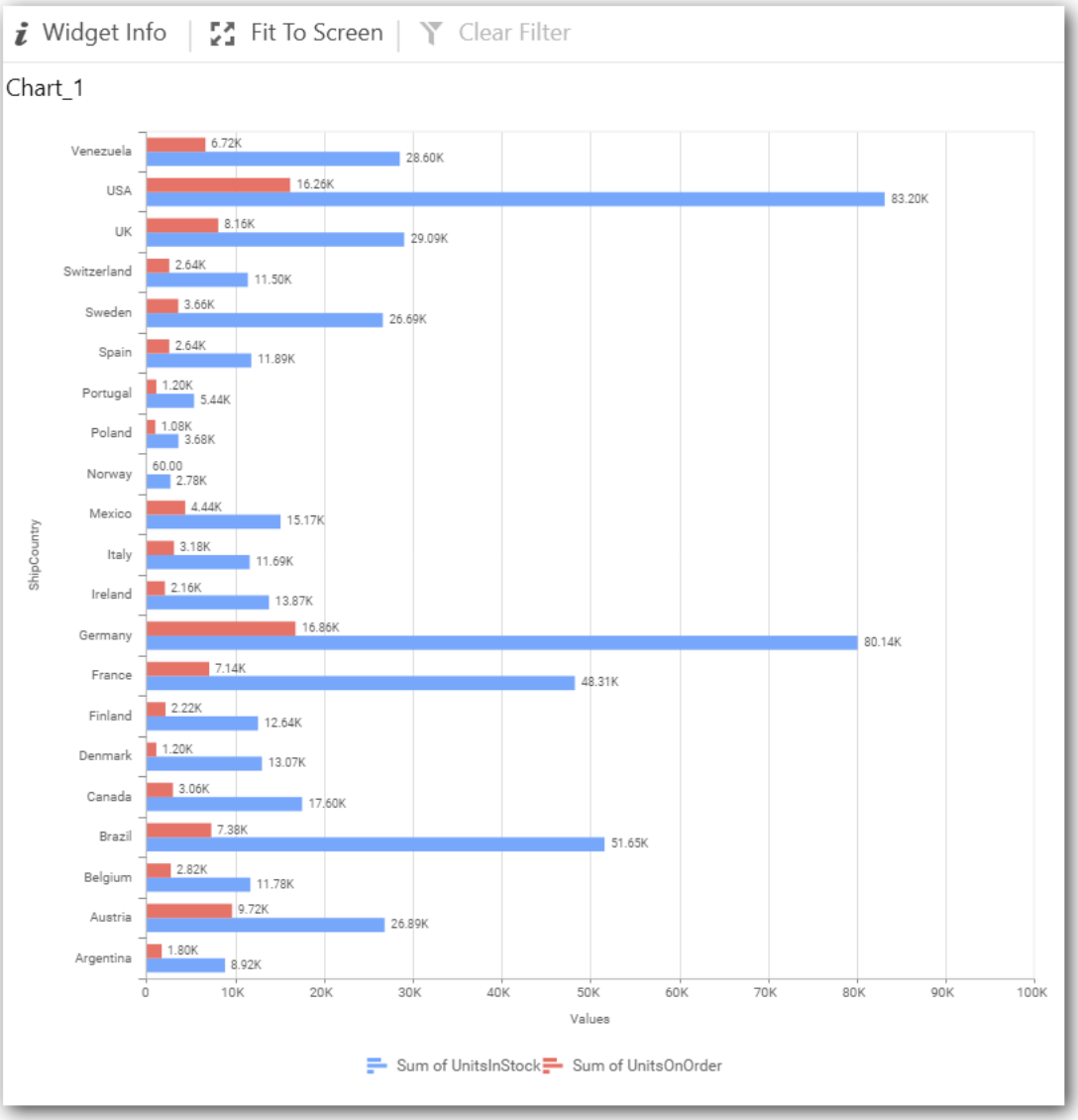
You have options to change the settings.



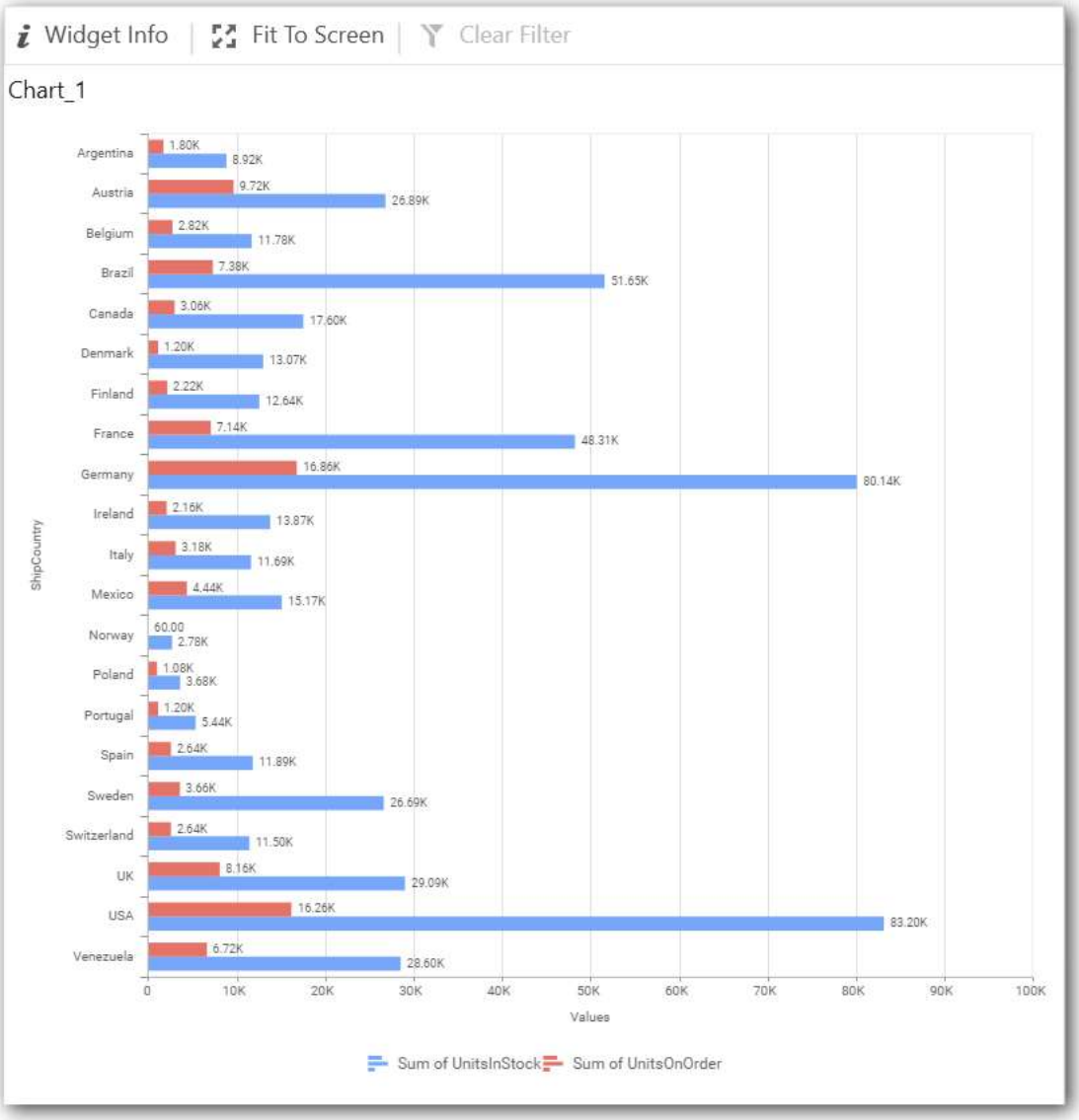


You can sort the chart either in Ascending or Descending series.

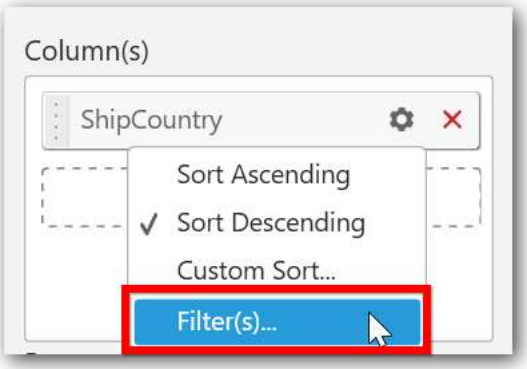
Ascending Order

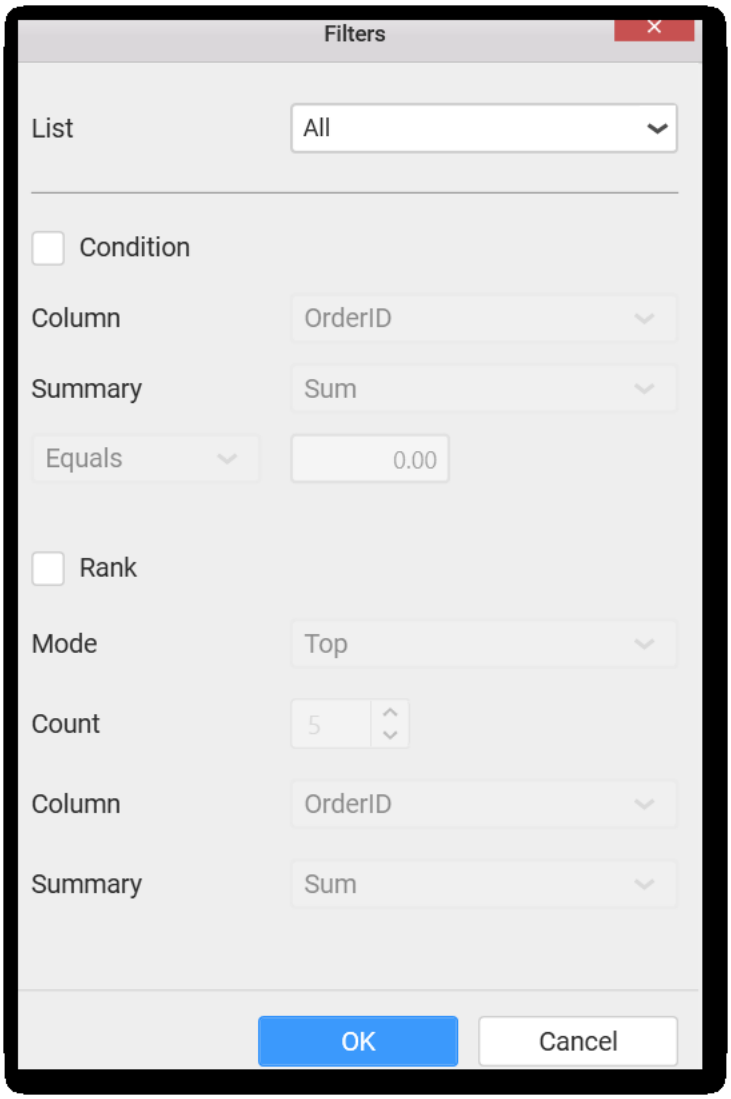


Descending order

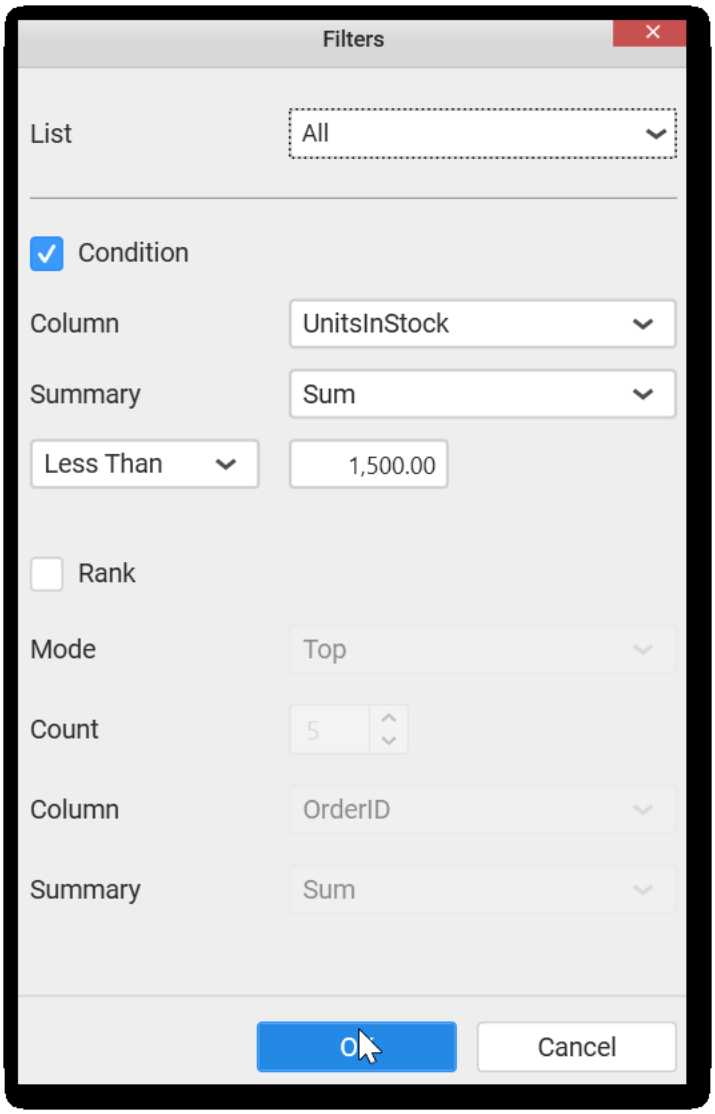


You can apply a filter.

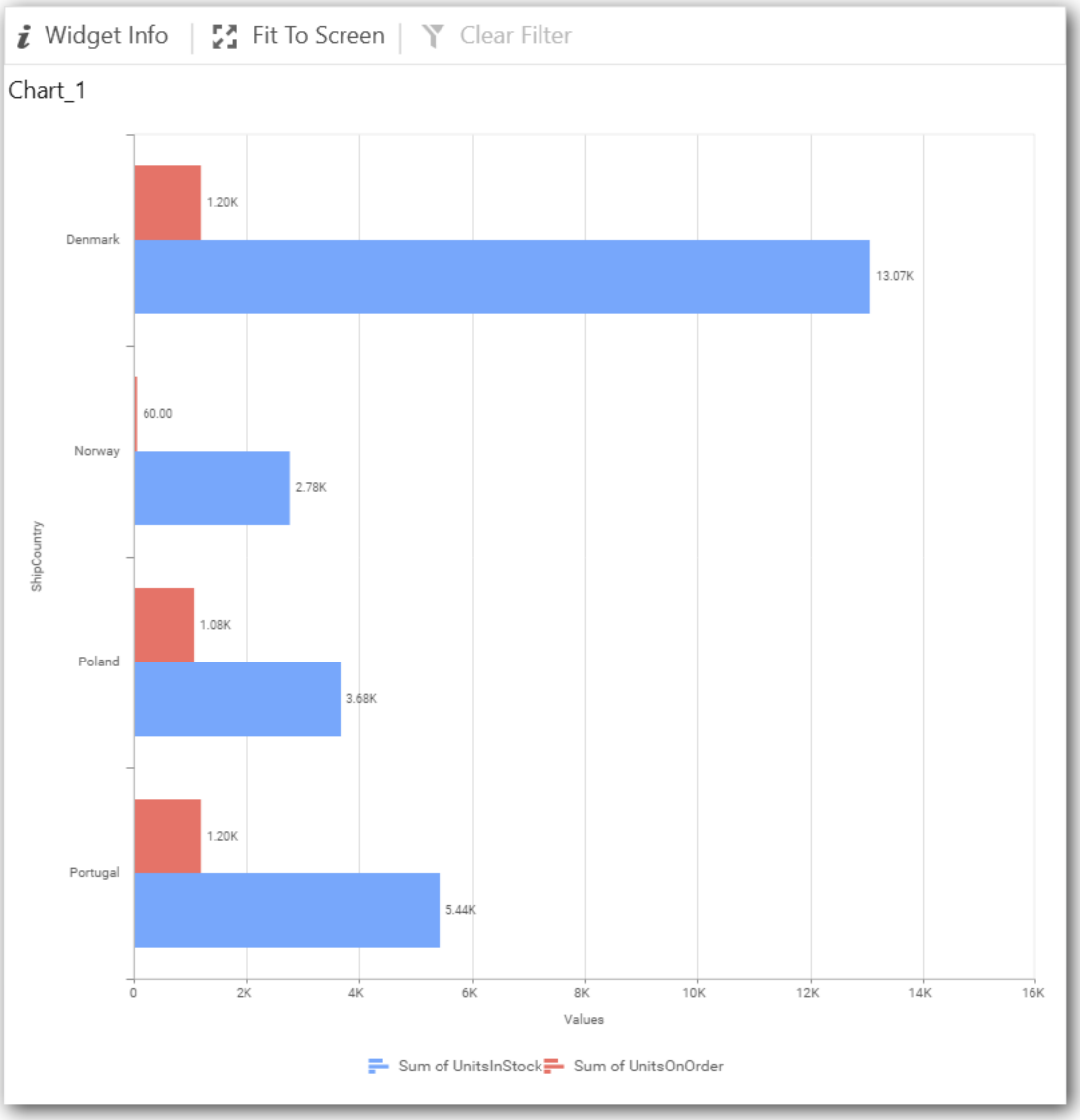




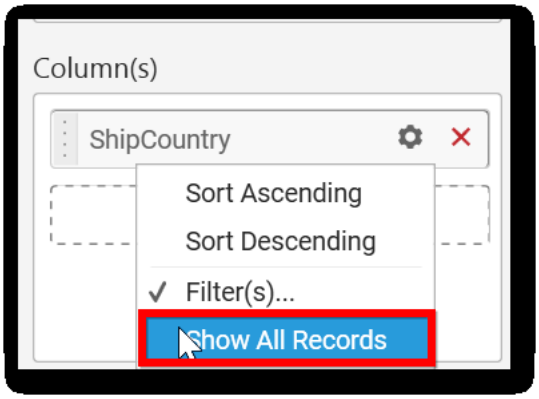
Select the Conditions and Rank you need.



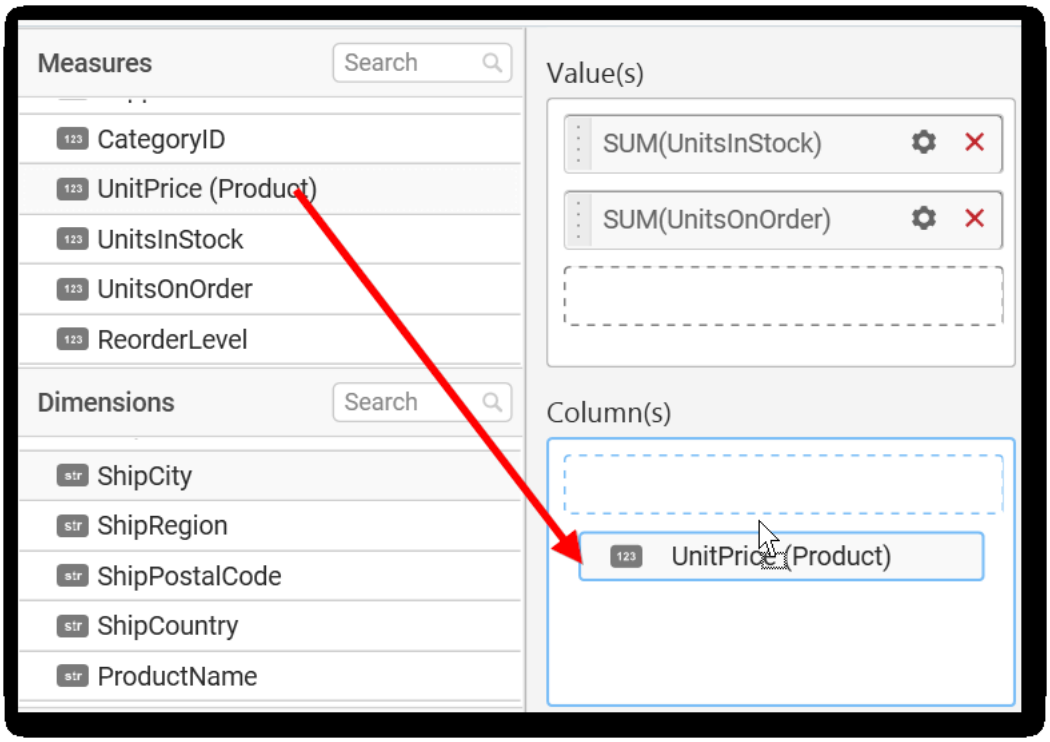
Now the chart will be rendered like this

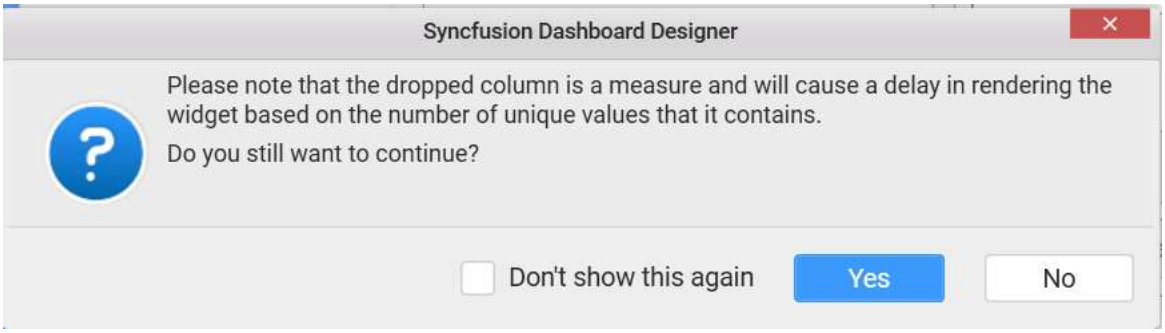


To show all records again click on Show All Records.



On adding Measures into Column(s) will show the following alert.

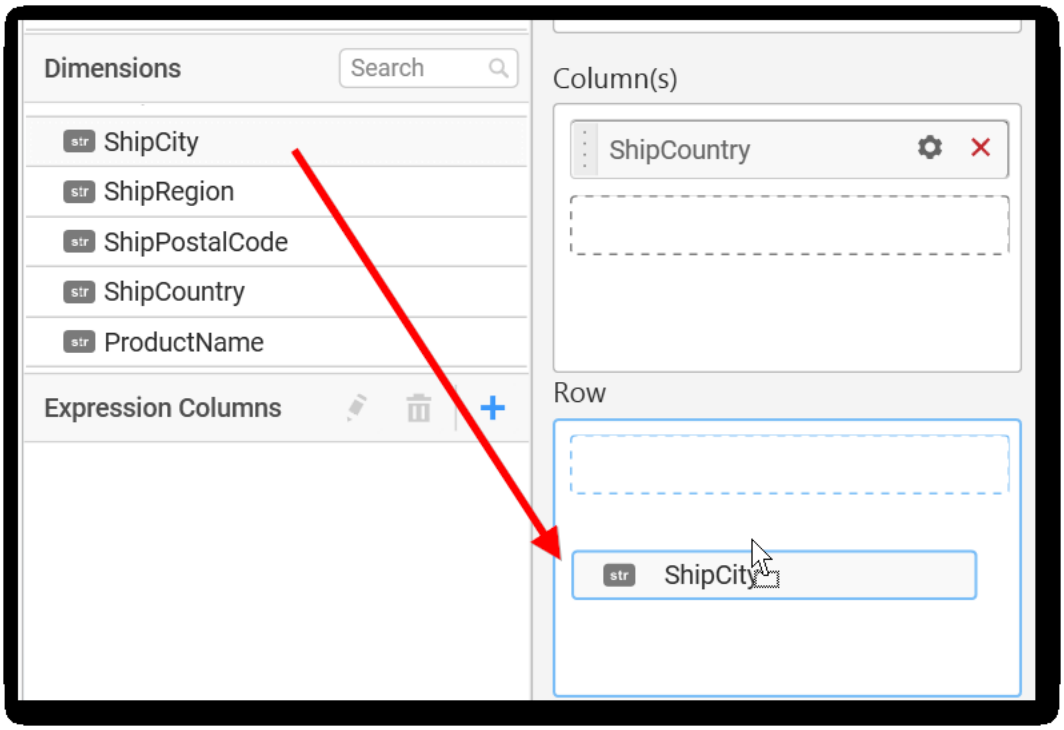




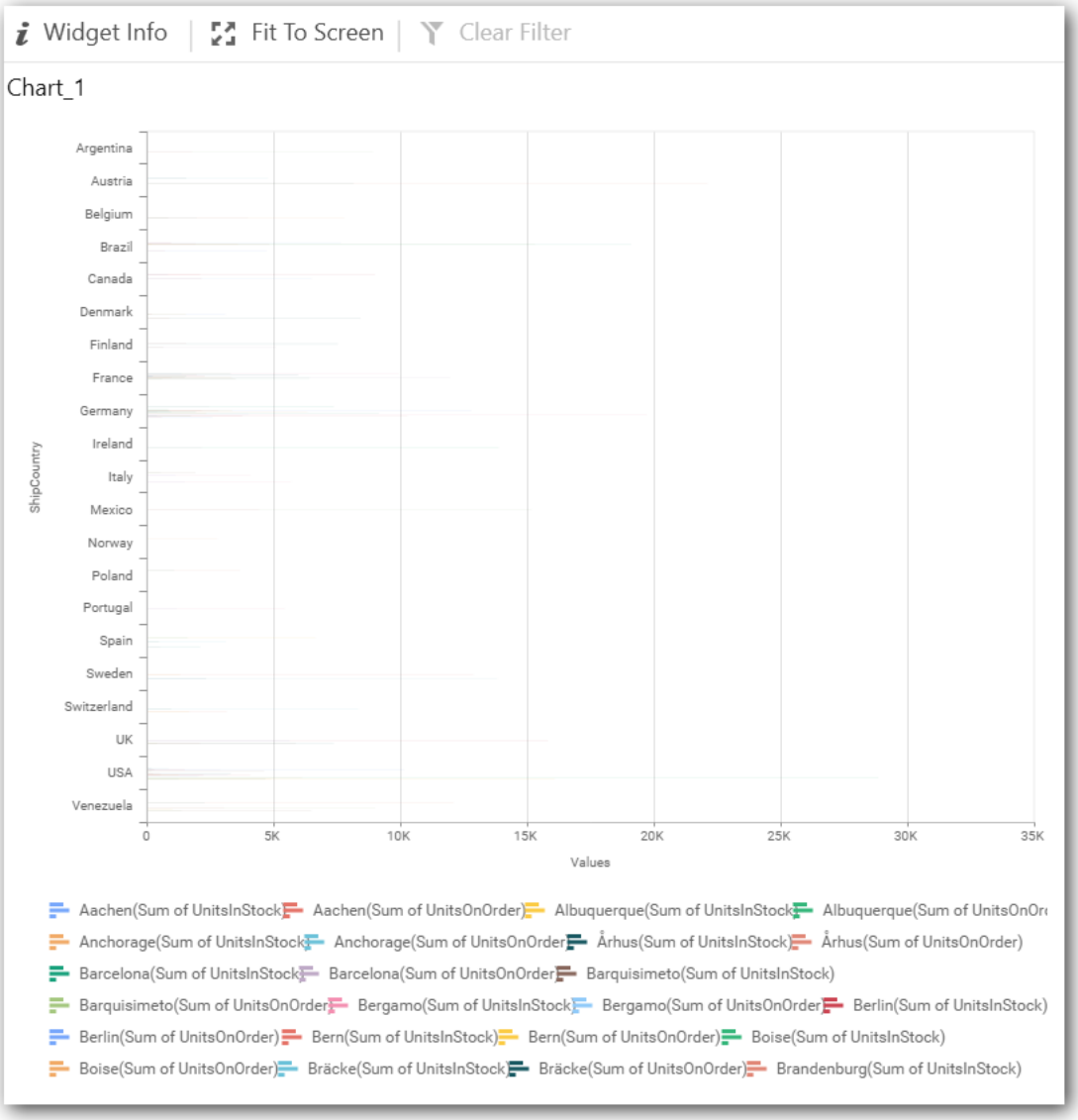
To continue select Yes, otherwise select No.

Assigning Row

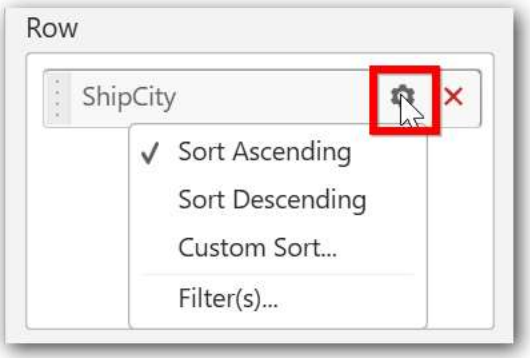
You can add Dimension into the Row field for series chart



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s).



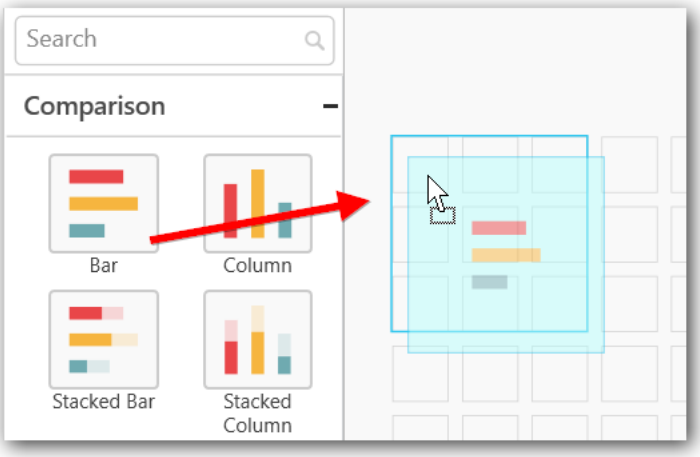
How to configure the SSAS data to Bar Chart?

Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure column or expression column that you would like to analyze can be dropped into Values(s) block. The dimension

column that you would like to categorize the measure column, can be dropped onto Column(s) block. If you would like to categorize based on a series column, then the respective dimension column can be dropped onto Row(s) block in addition. These blocks are composed into Data pane.

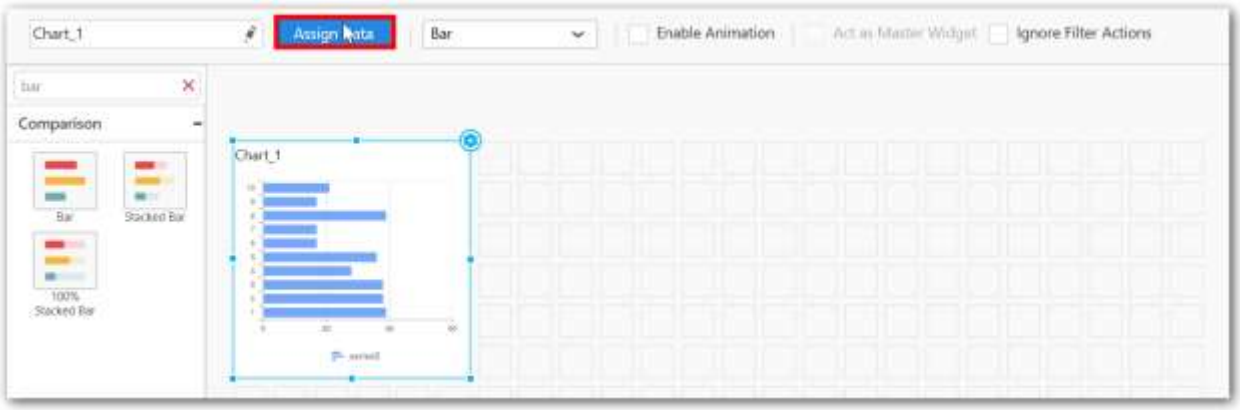
Following steps illustrates configuring SSAS data to bar chart.

Drag and drop the Bar chart into canvas and resize it to your required size.

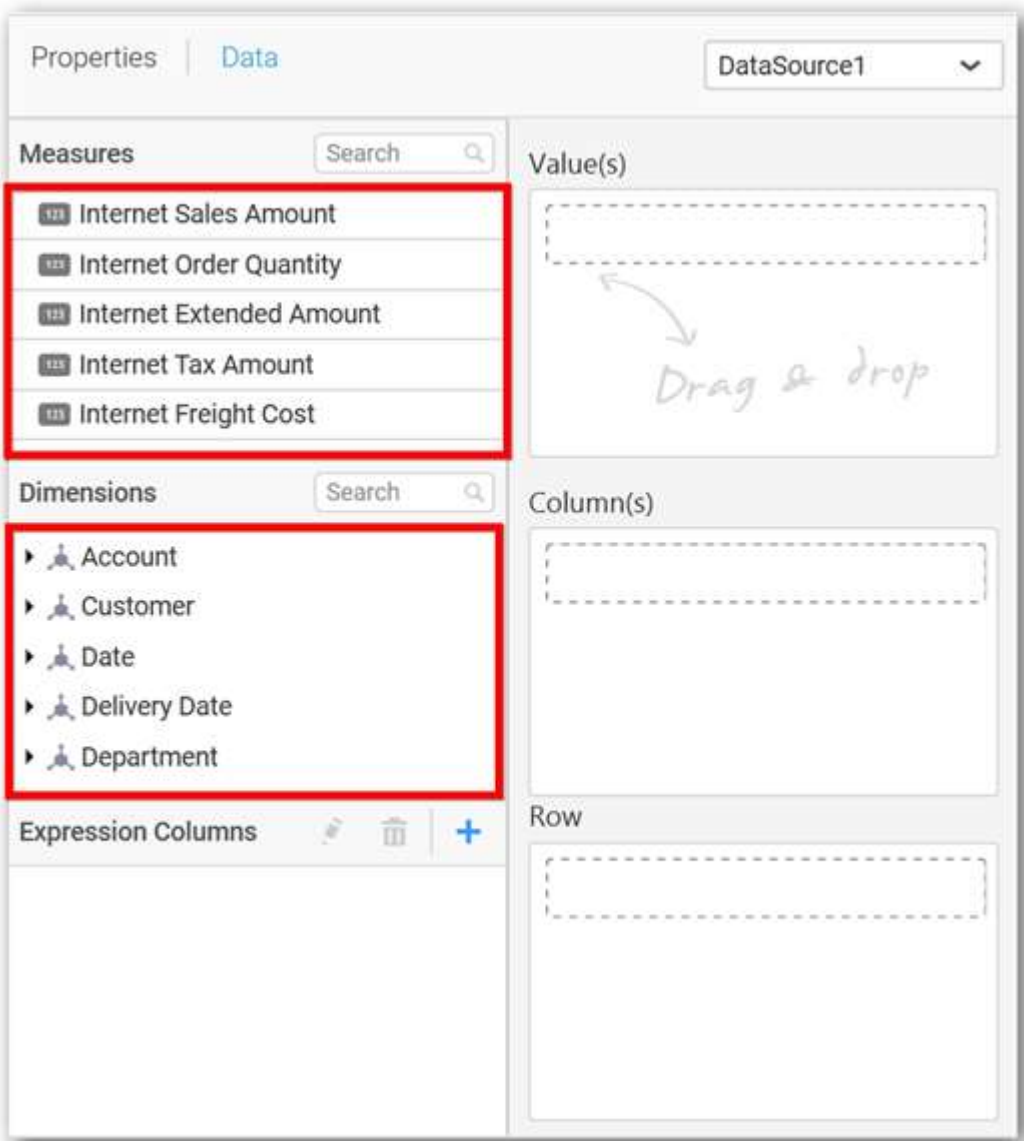


Select the dropped widget using mouse.

Click on Assign Data button.

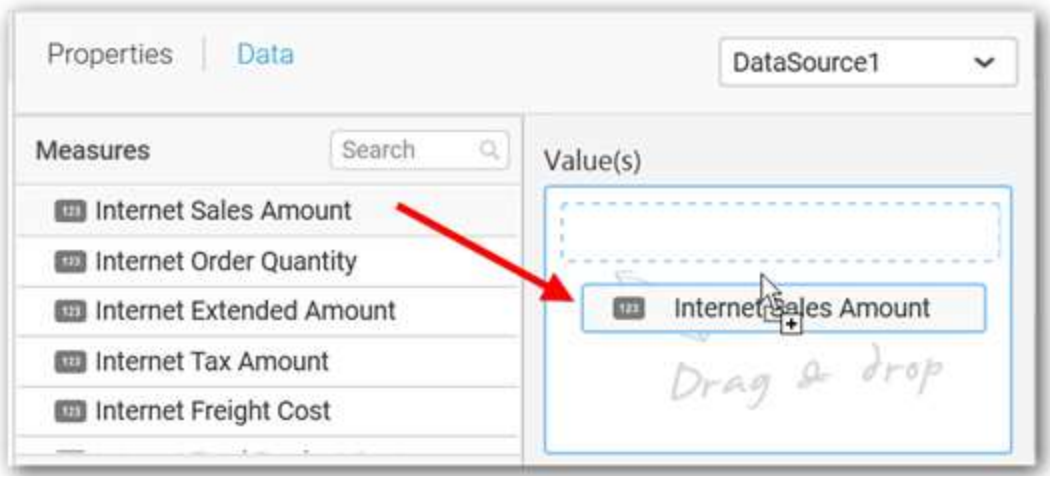


A Data pane will be opened with available Measures and Dimensions

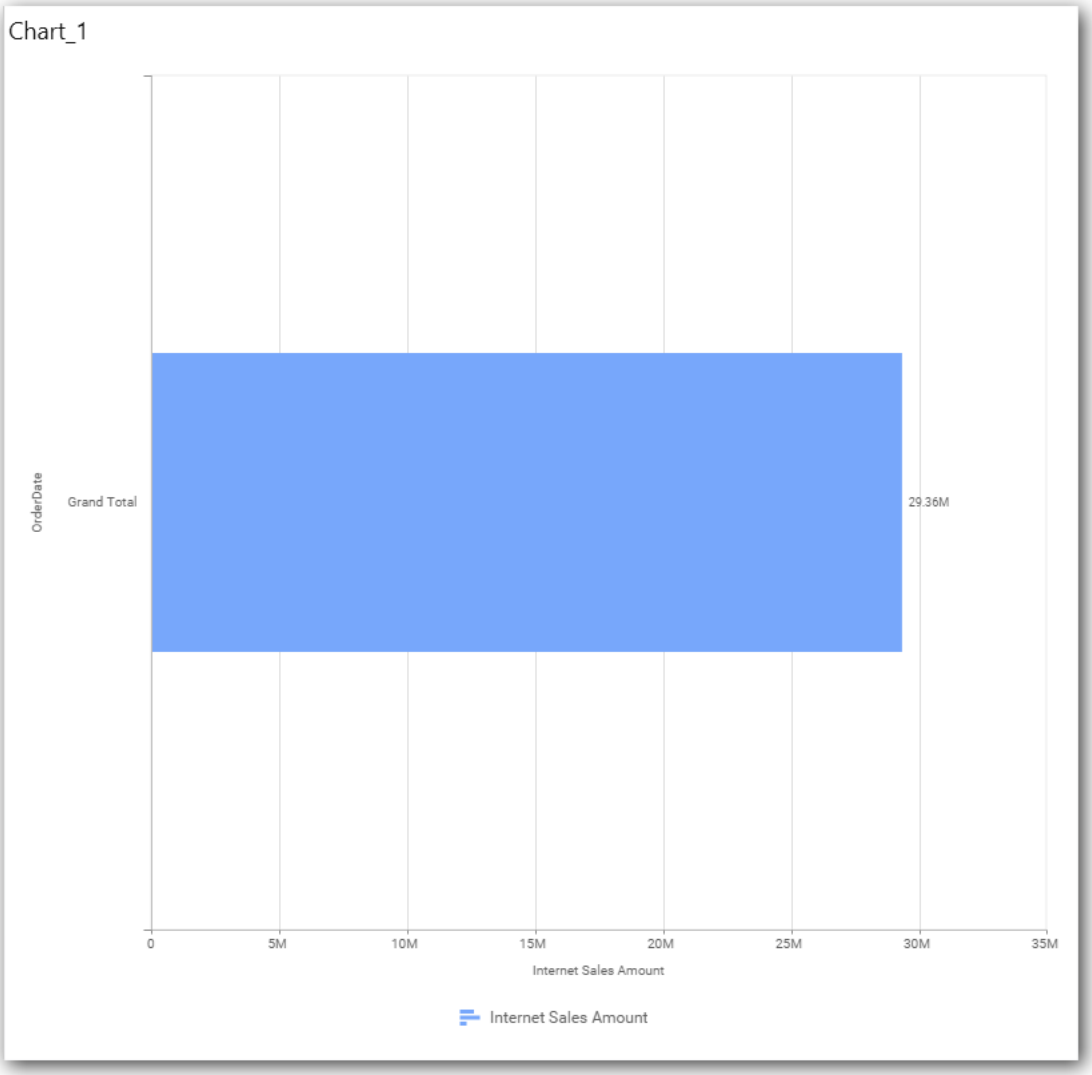


Assigning Value(s)

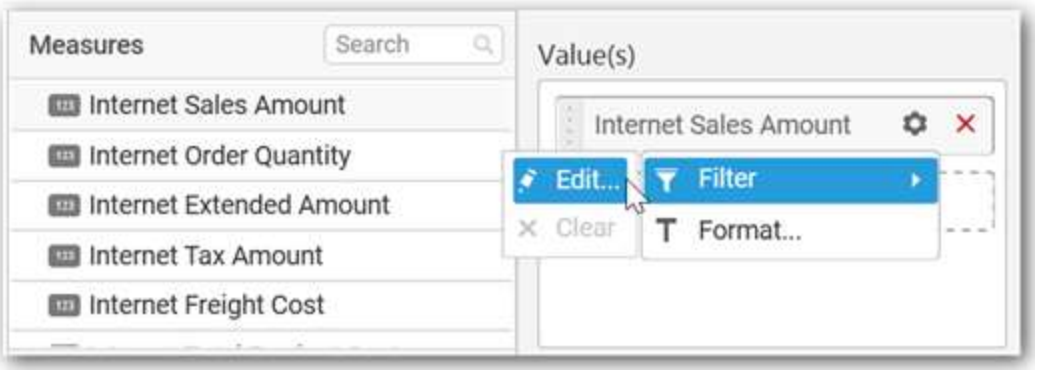
Drag and drop a column under Measures category into Value(s) section.



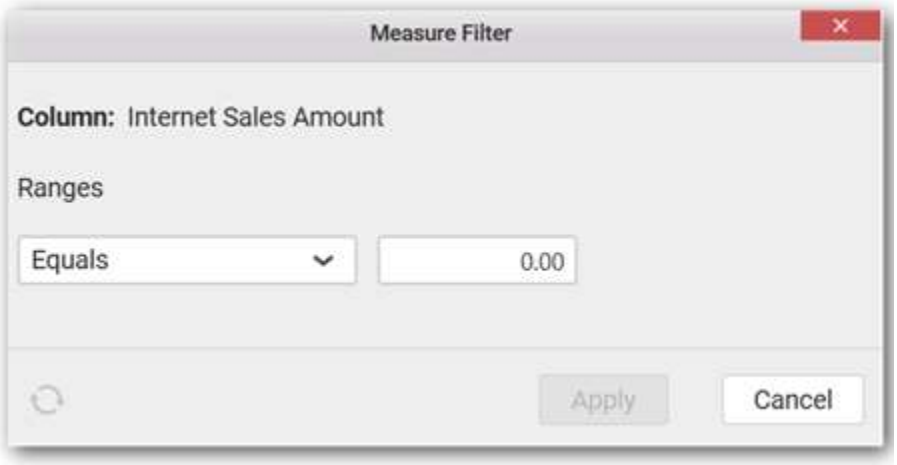
Now the chart will be rendered like this.



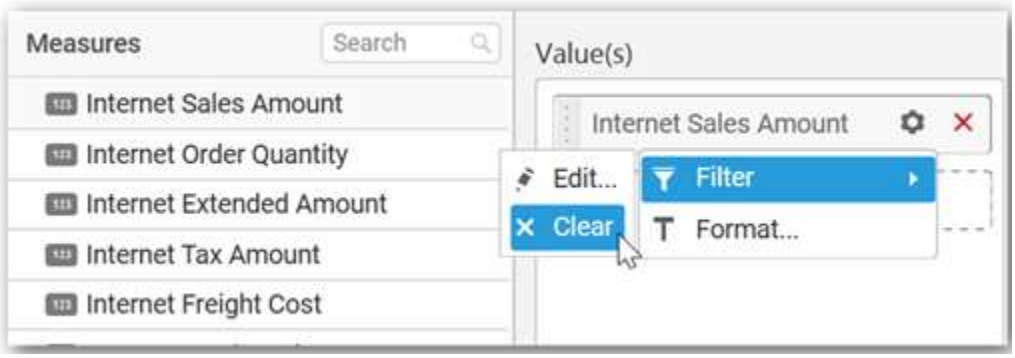
You can select what data to be displayed by choosing filter option



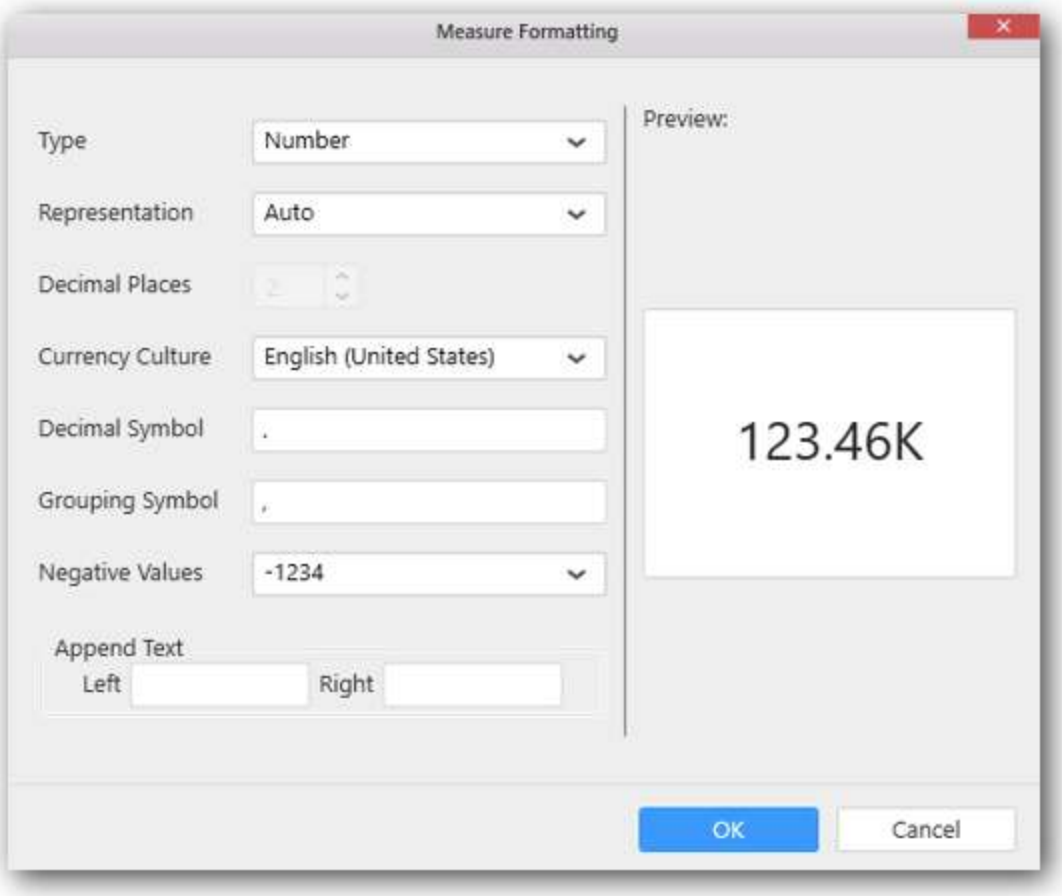
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



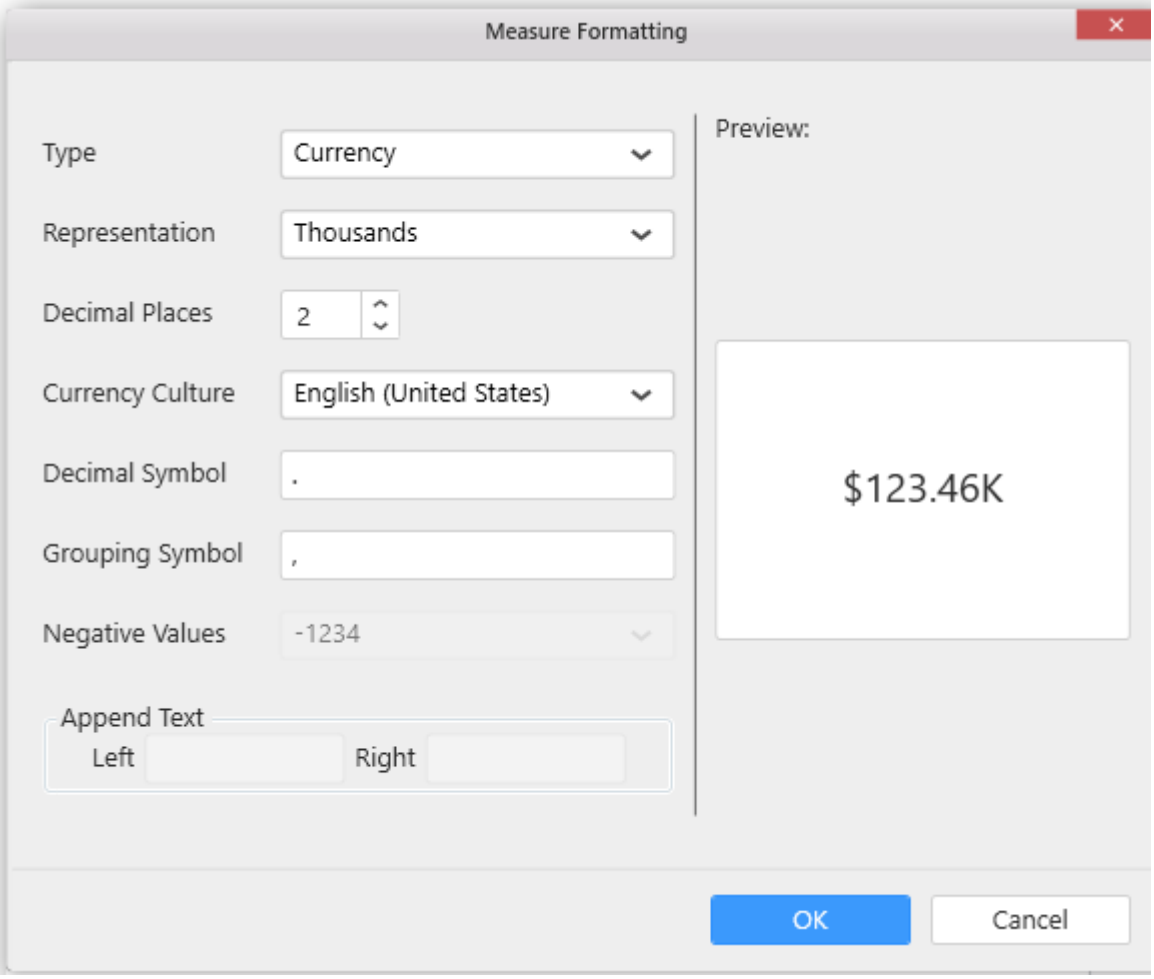
Select Clear option to clear the defined filter.



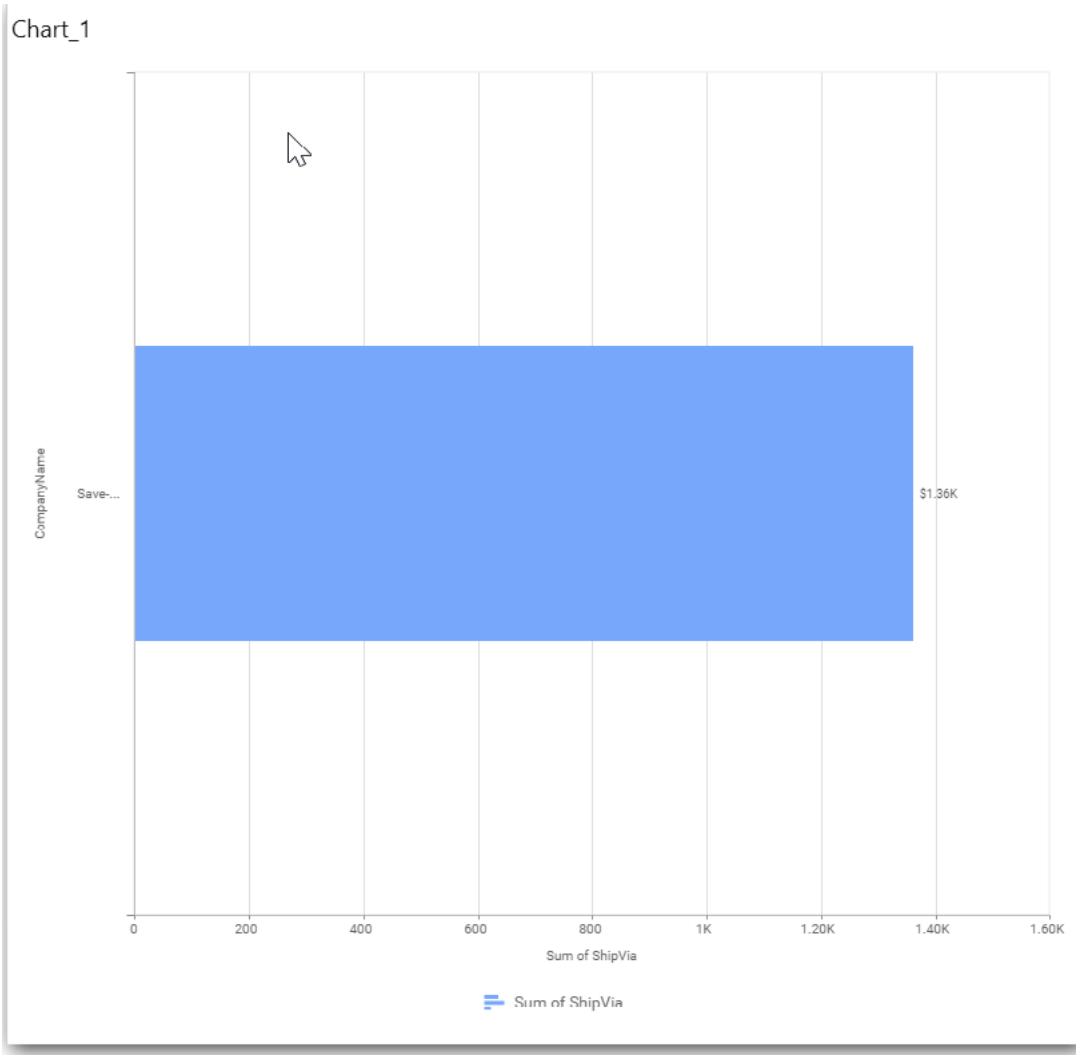
Select Format option to define the display format to the values in the column through Measure Formatting window.



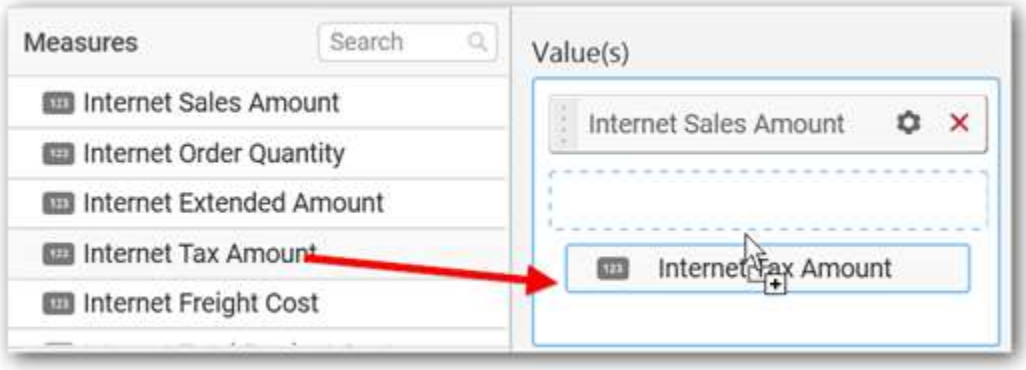
Choose the options you need and click OK.

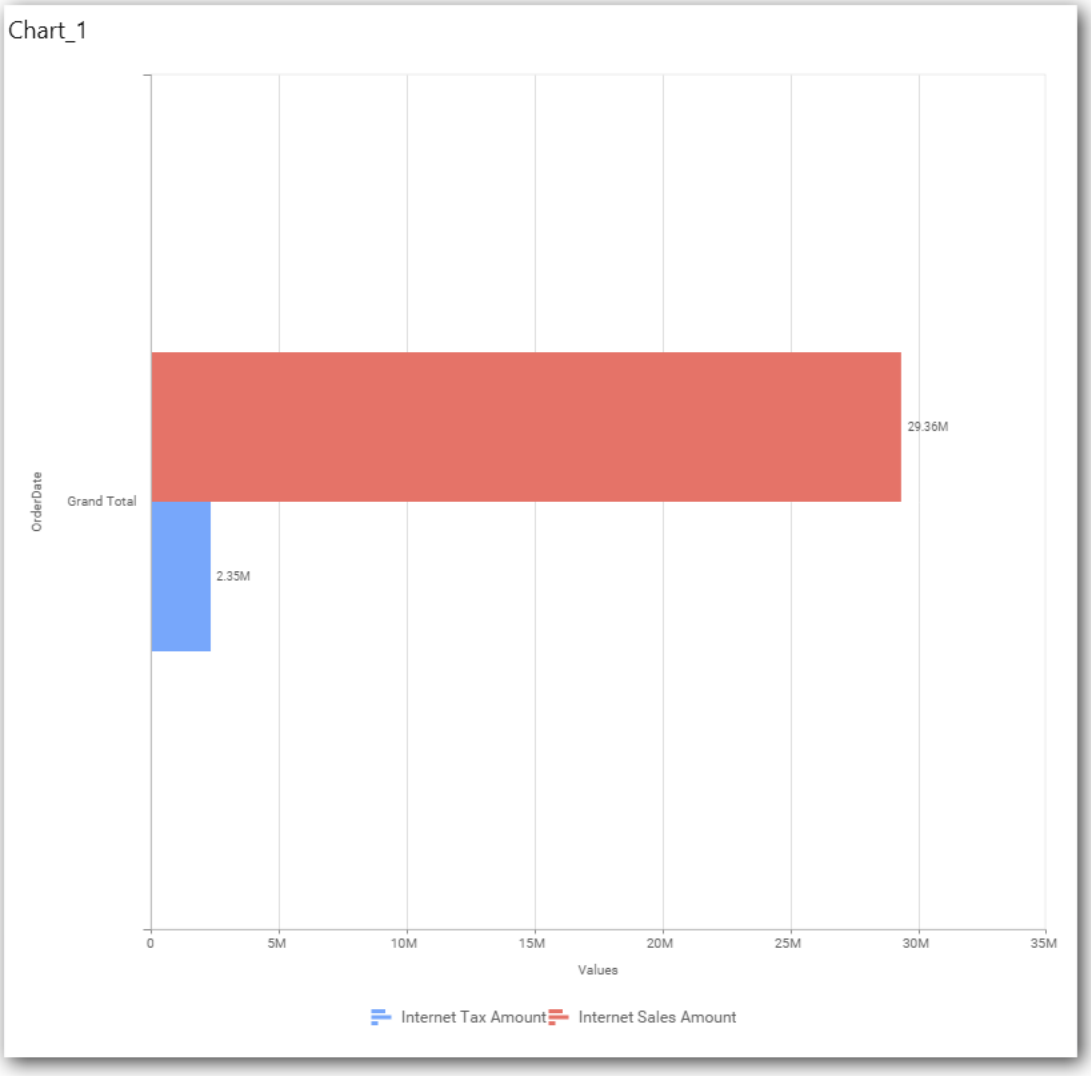


Now the Chart will be rendered like this.



You can add more number values by drag drop the Measures into Value(s) field.

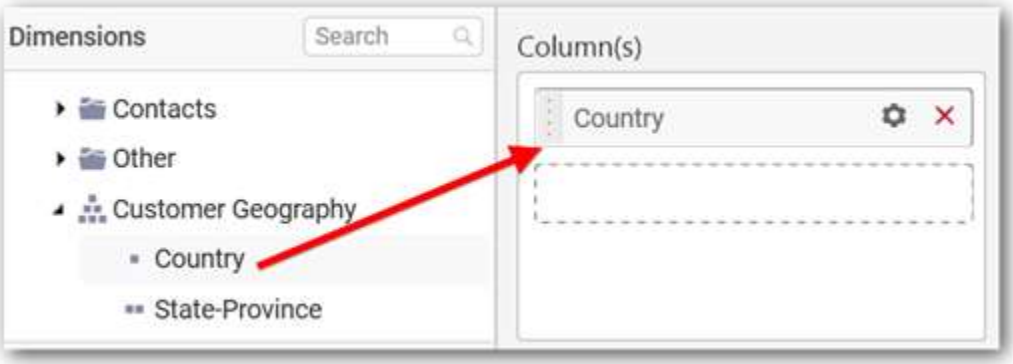


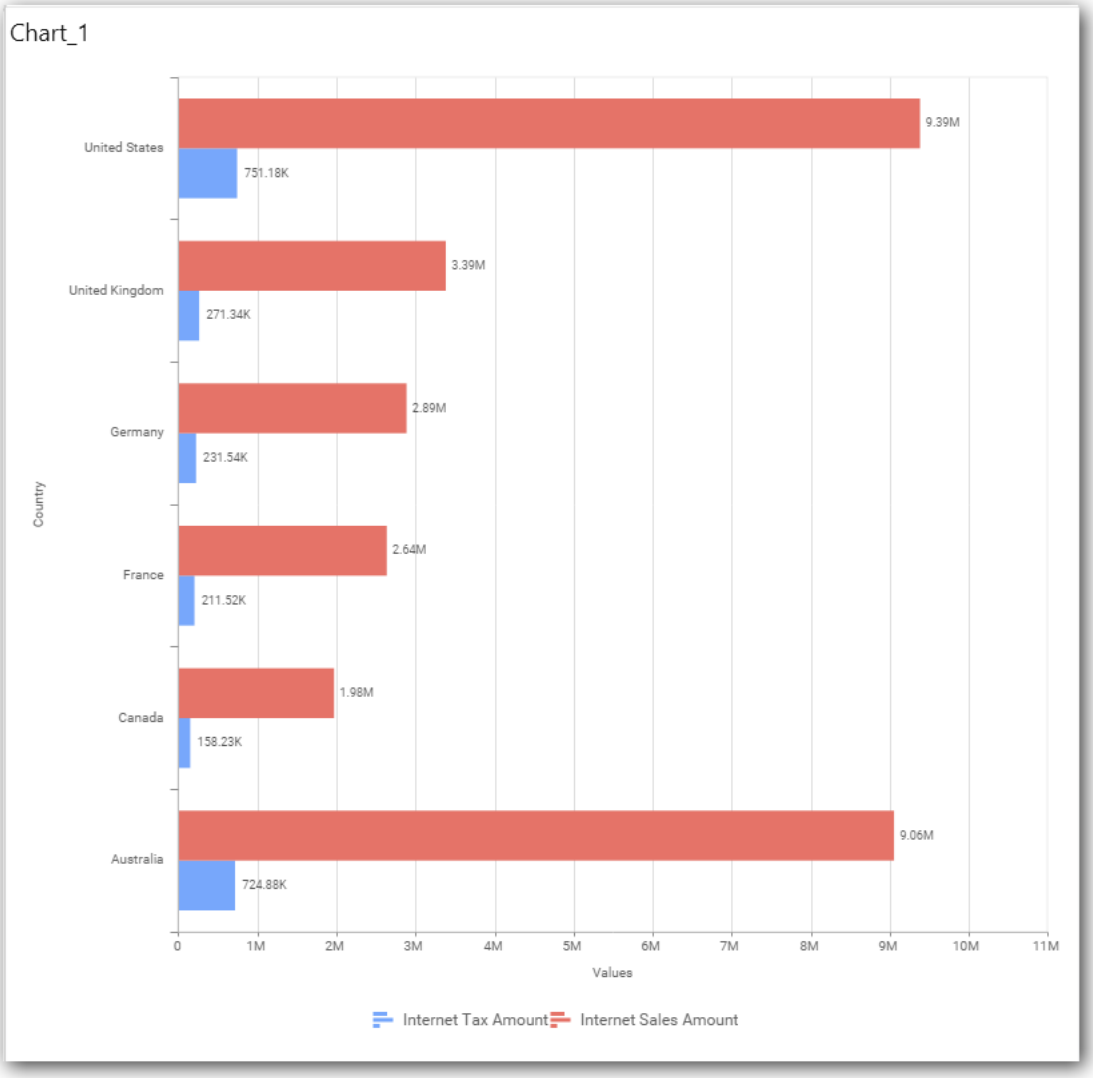


You can also add Dimensions and Columns to Value(s).

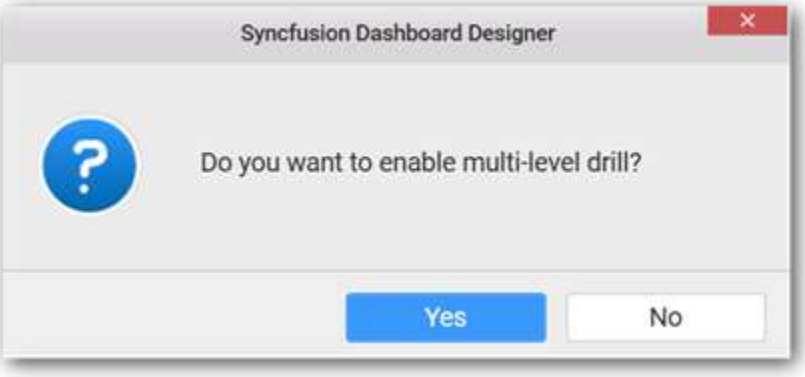
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.



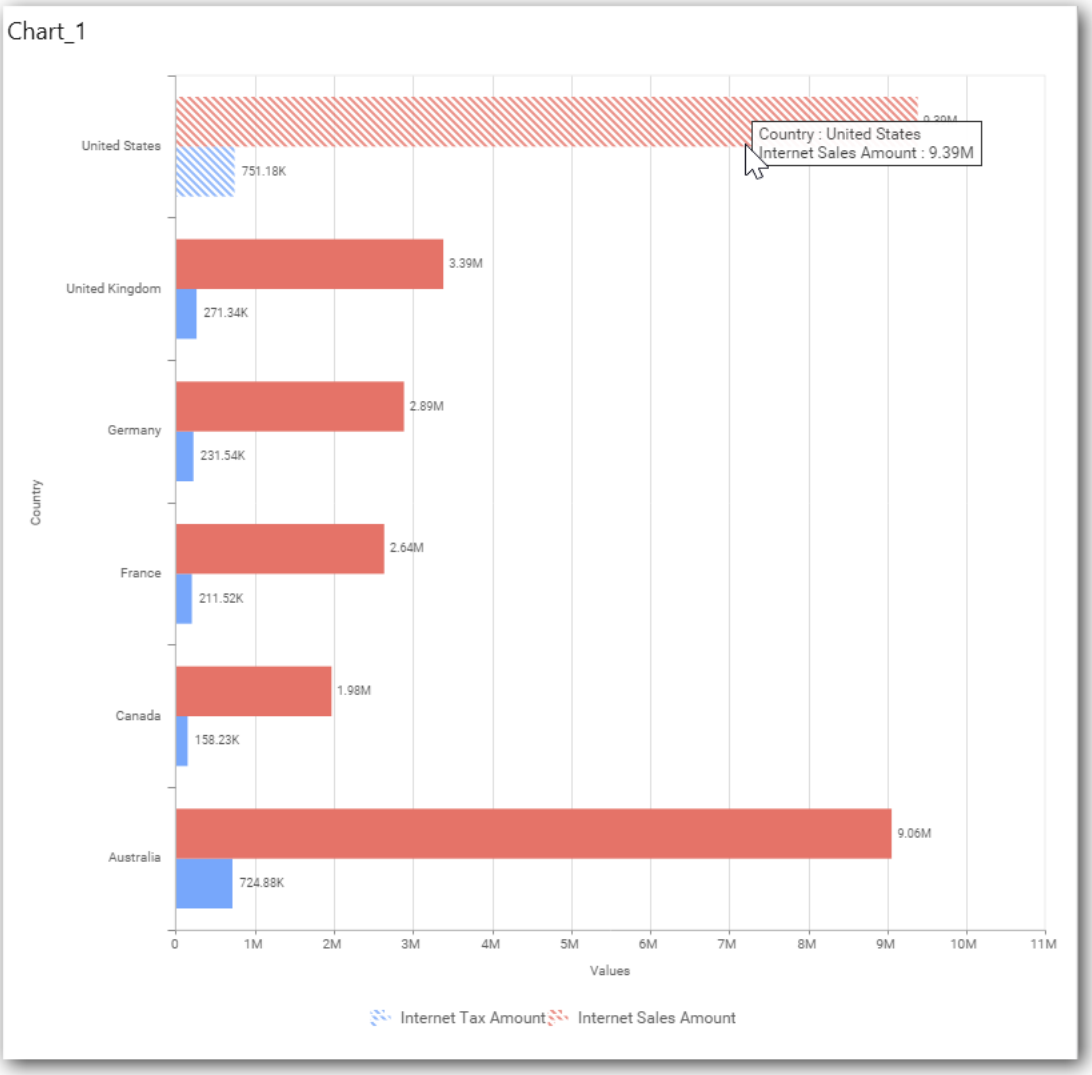


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

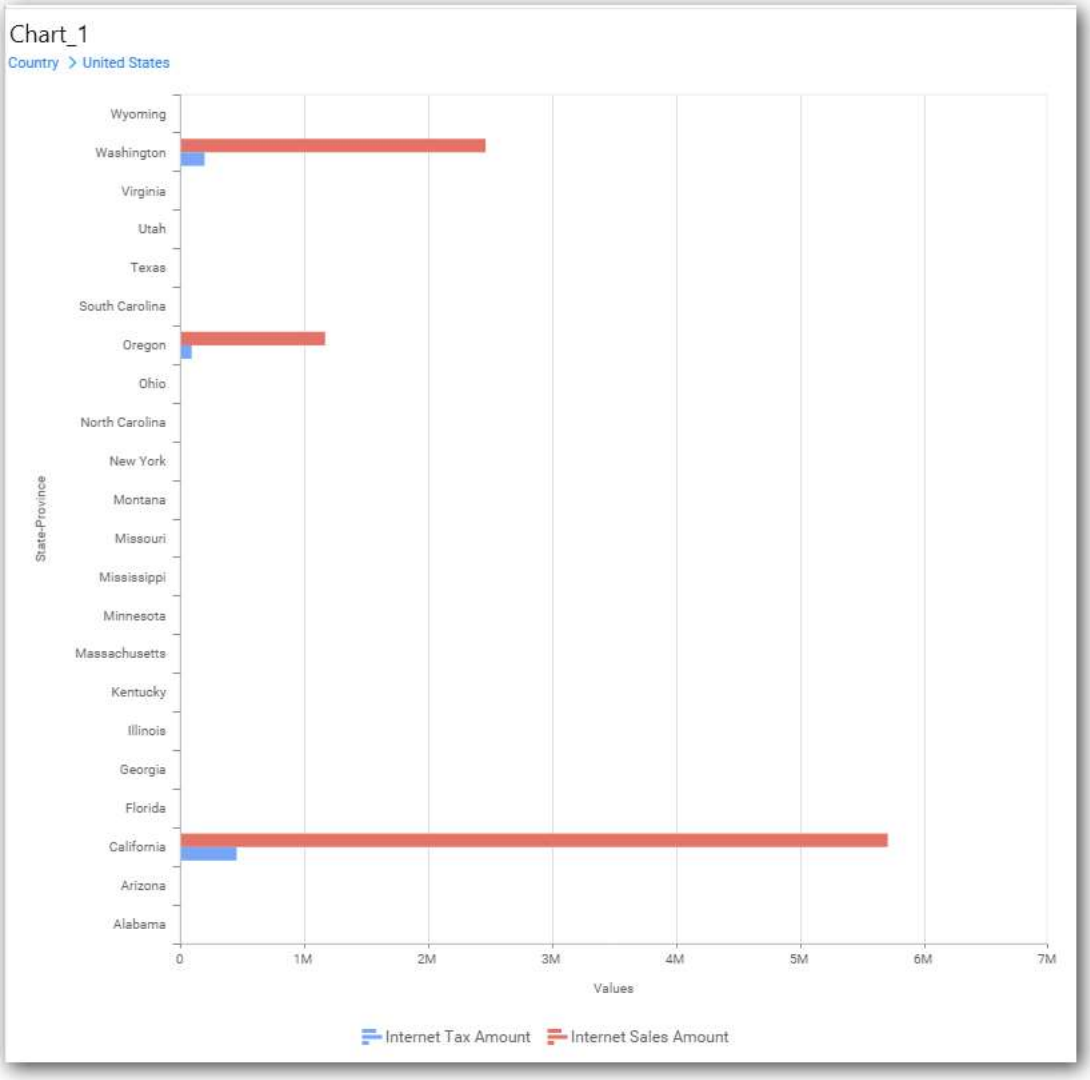


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

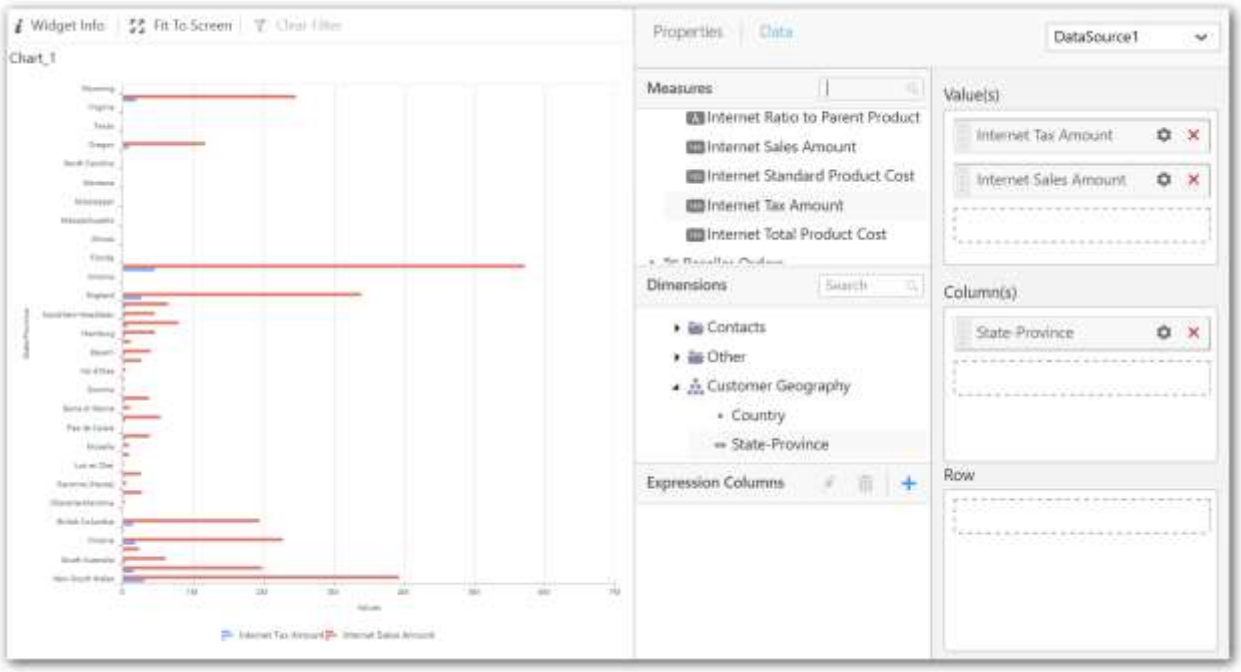
Click the respective data value marker in chart to drill into its inner level.



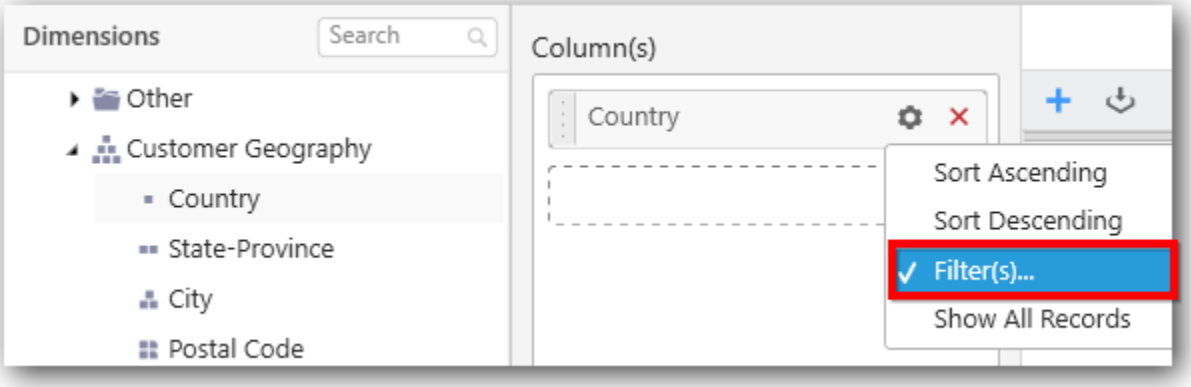
The drilled view of the chart is follows.



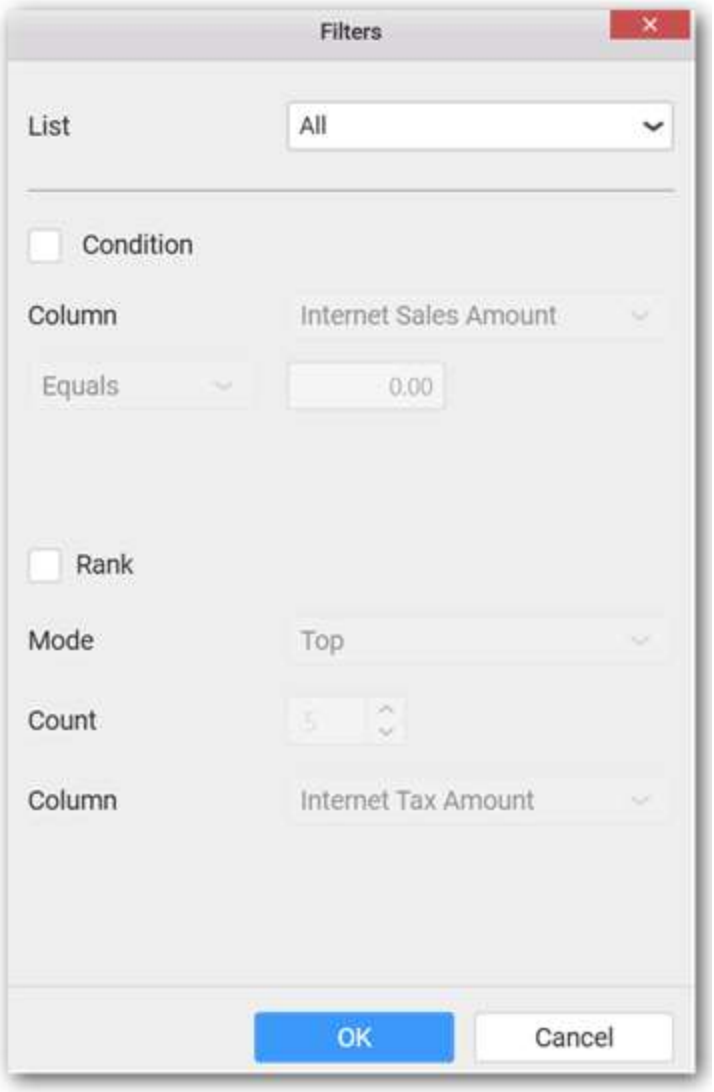
Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level



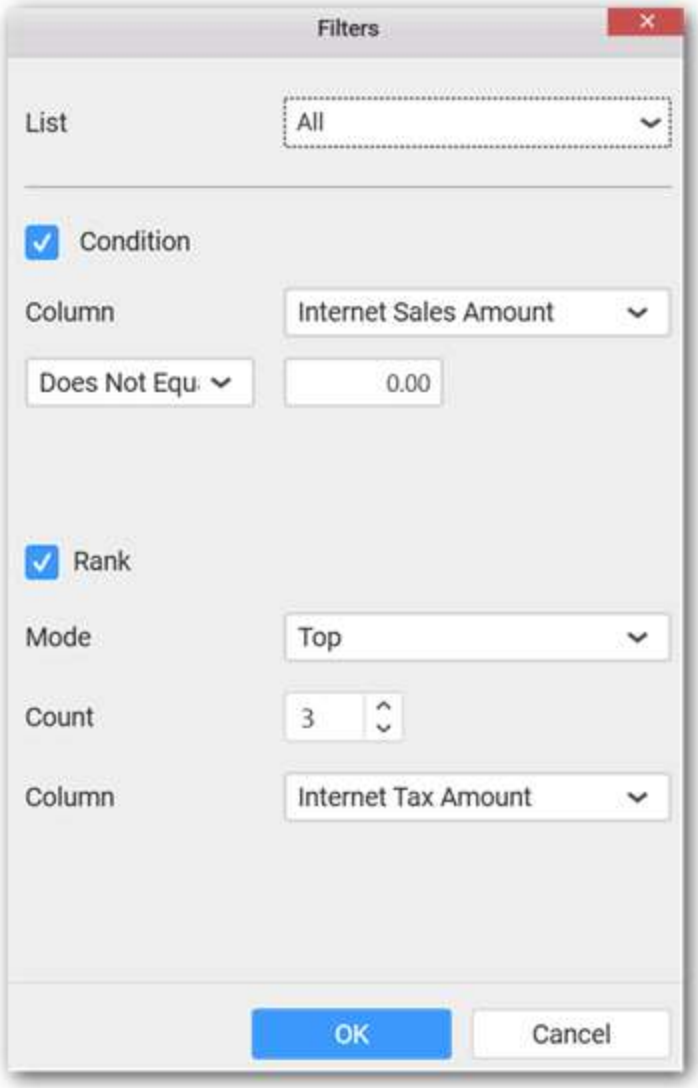
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



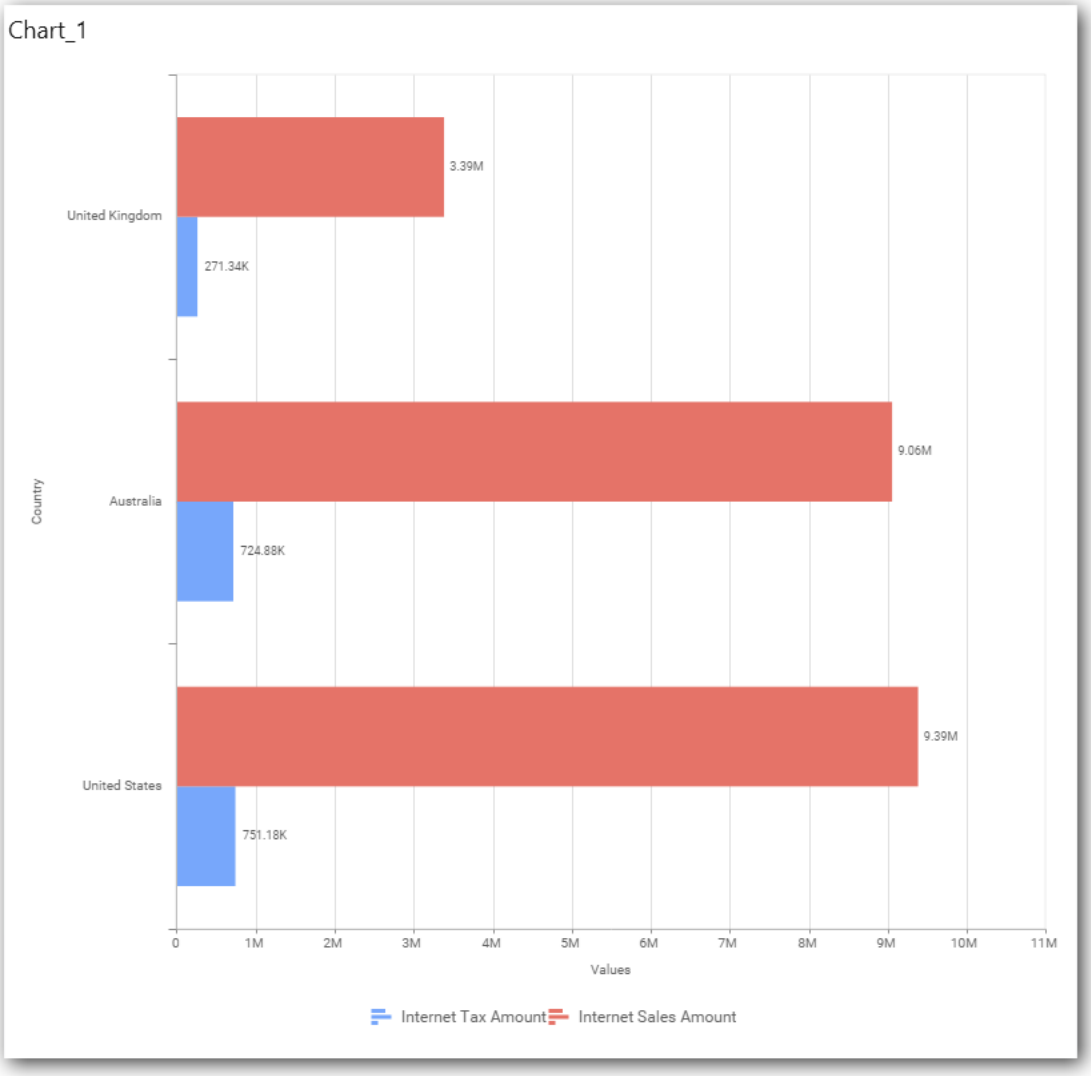
Select Filter(s)... to launch the Filters window.



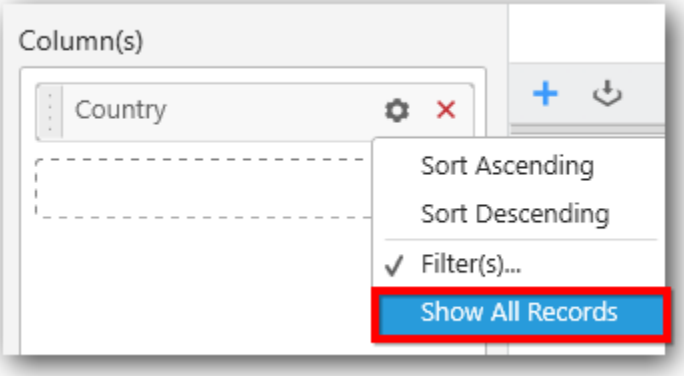
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

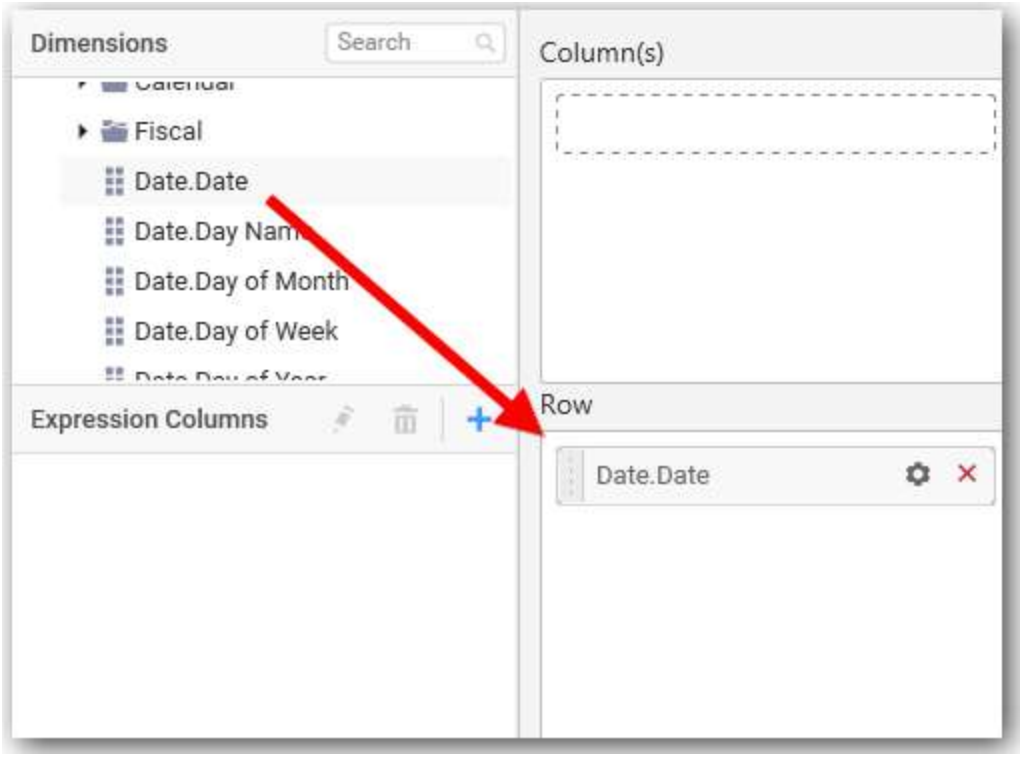


To show all records again click on Show All Records.

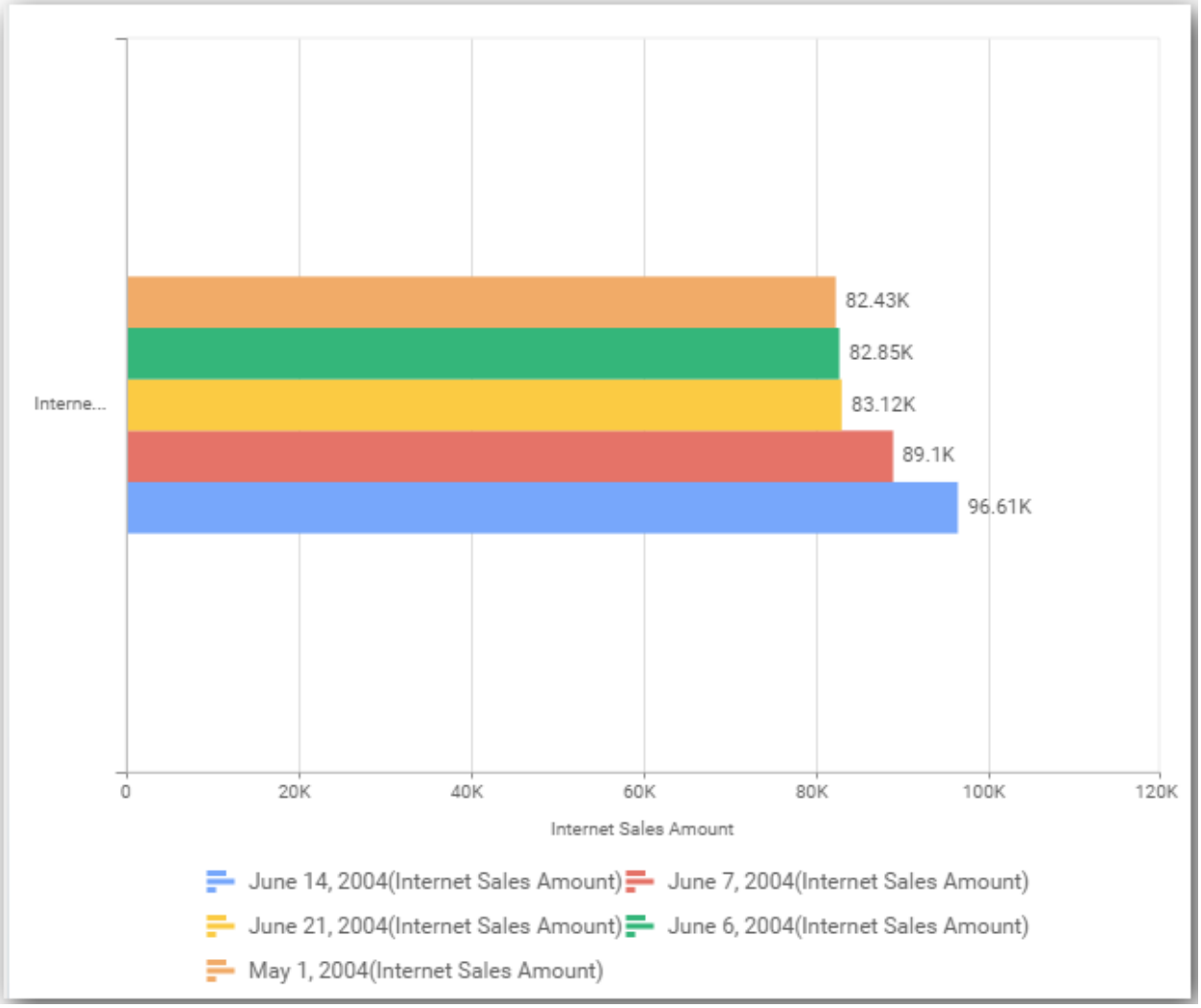


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.

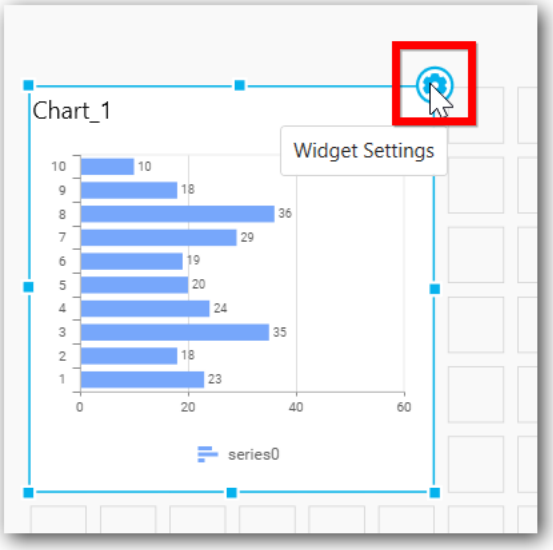


How to format Bar Chart?

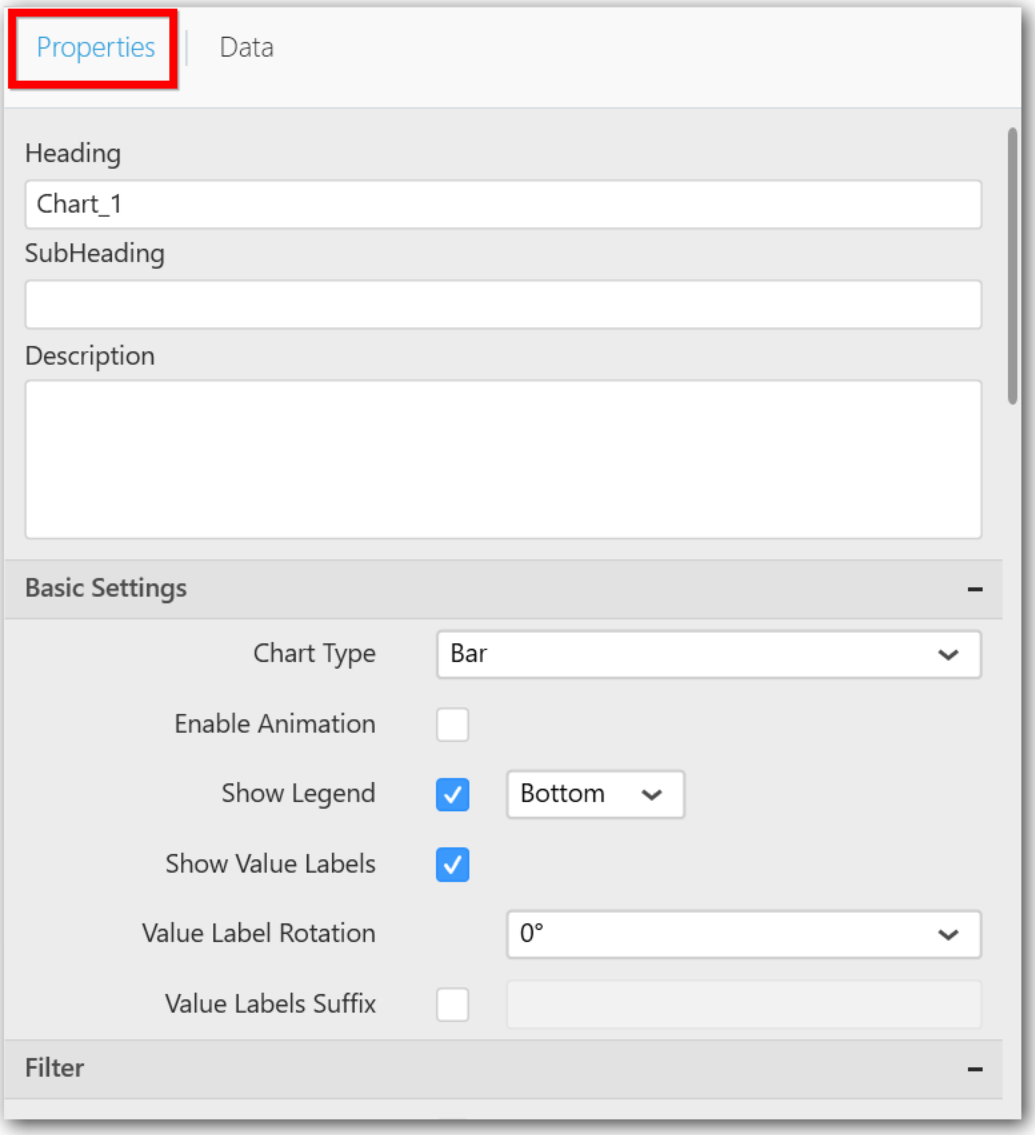
You can format the bar chart for better illustration of the view that you require, through the settings available in Properties pane.

To format bar chart follow the steps

1. Drag and drop the bar chart into canvas and resize it to your required size. 2. Configure the data into bar chart. 3. Focus on the bar chart and Click on Widget Settings.

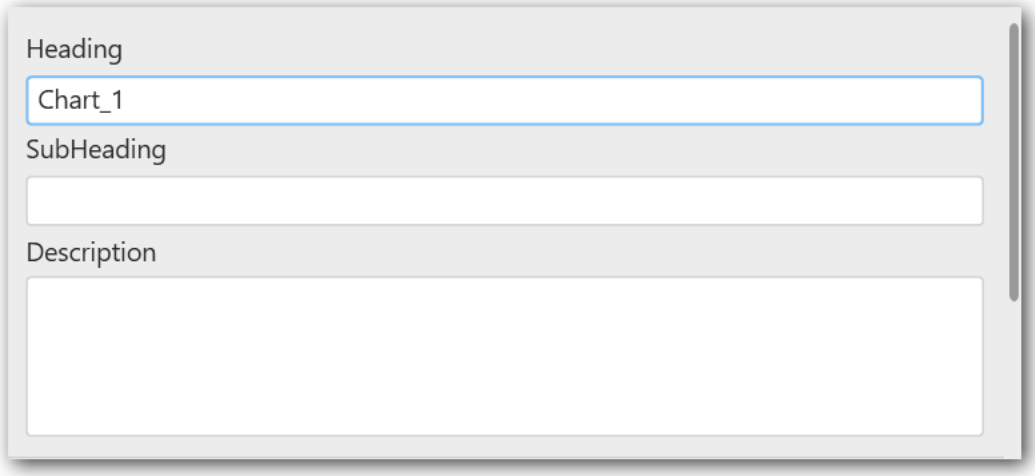


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this bar chart widget.

SubHeading

This allows you to set sub-title for this bar chart widget.

Description

This allows you to set description for this bar chart widget, whose visibility will be denoted by I icon, hovering which will display this description in tooltip.

Basic Settings

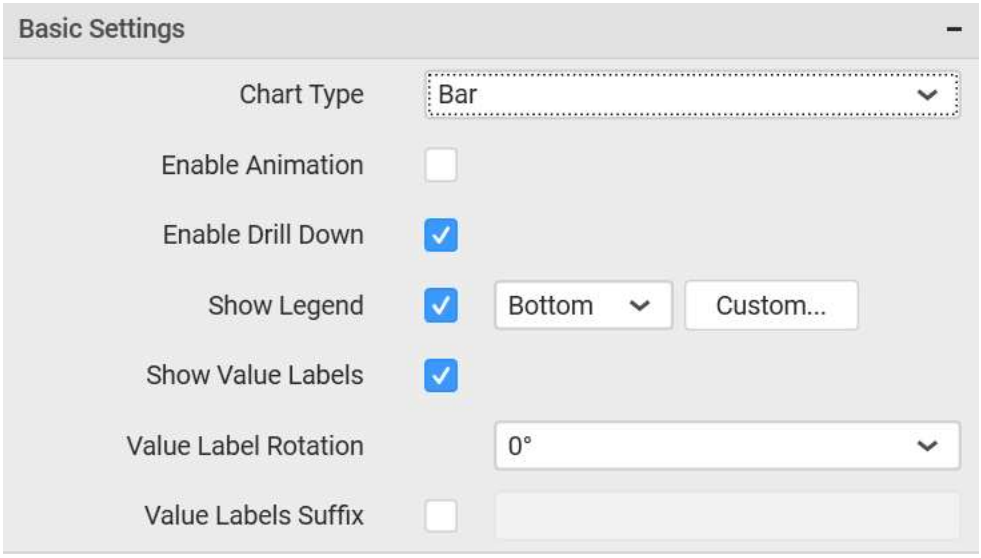


Chart Type

This allows you to switch the widget view from current chart type to another convertible chart type.

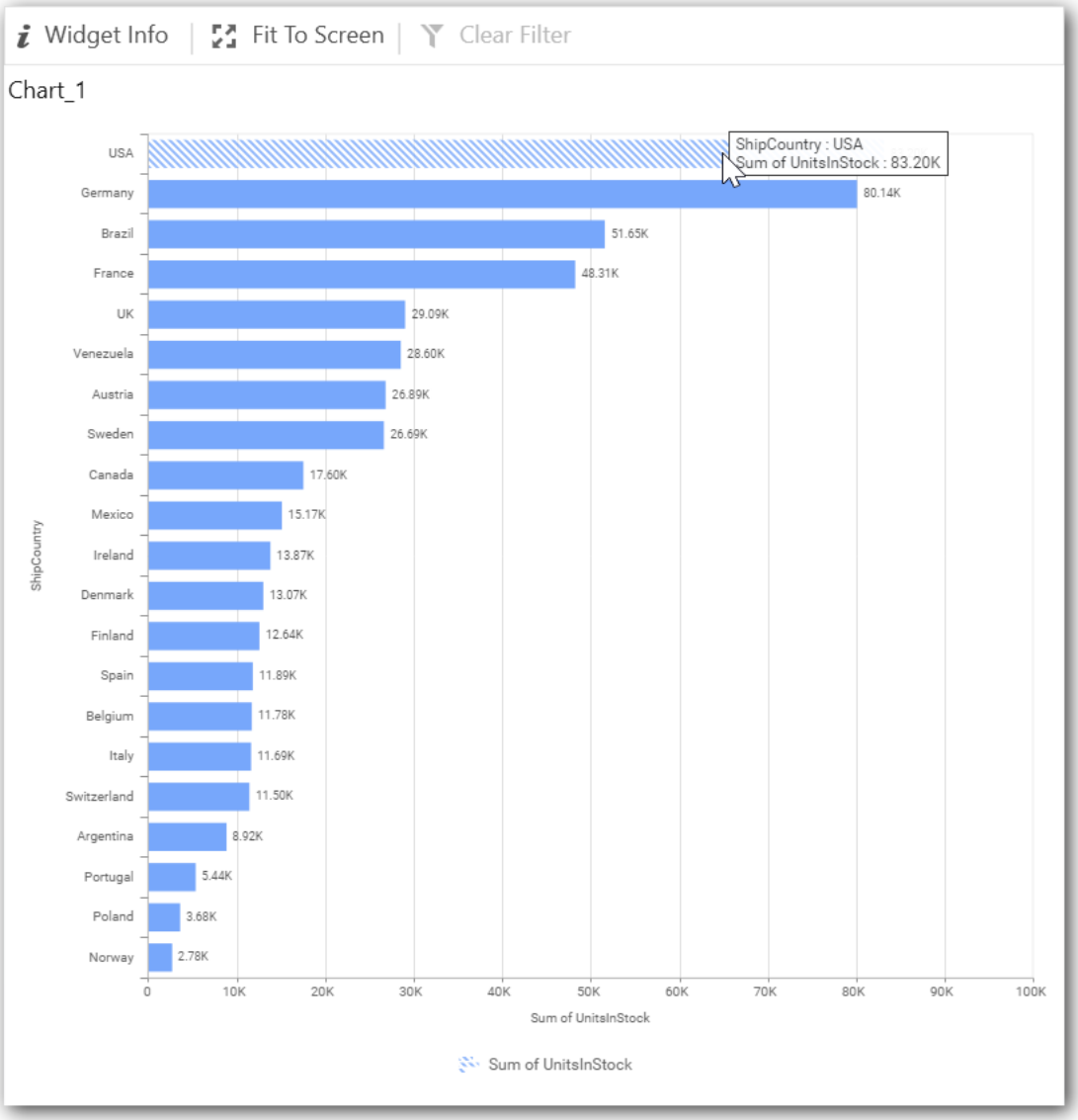
Enable Animation

This allows you to enable the series rendering in animated mode.

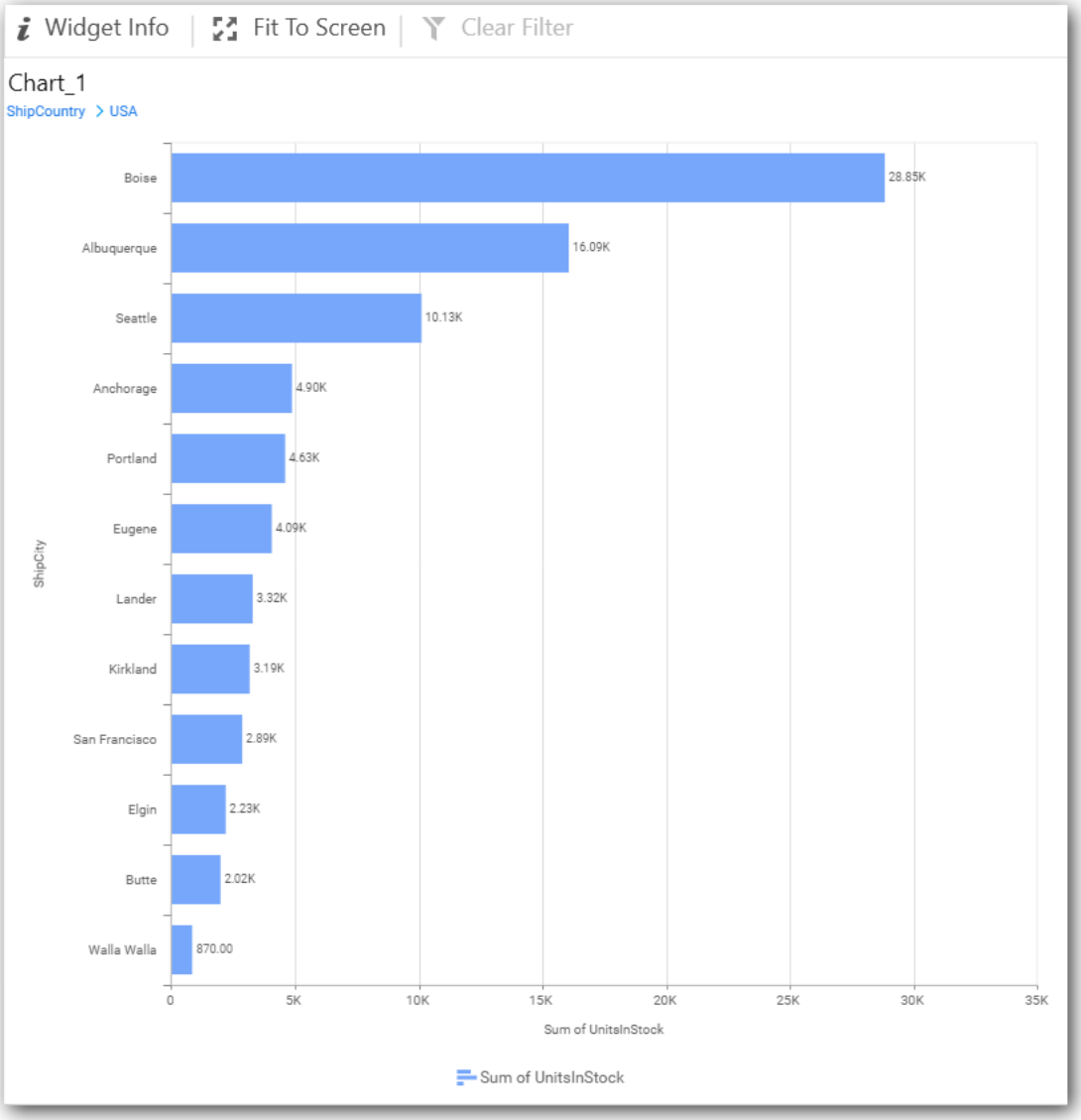
Enable Drill Down

This allows you to add more than one dimension column to the Column(s) block in Data Pane such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. With this setting disabled, trying to add more than one dimension column will replace the existing one.

Initial View

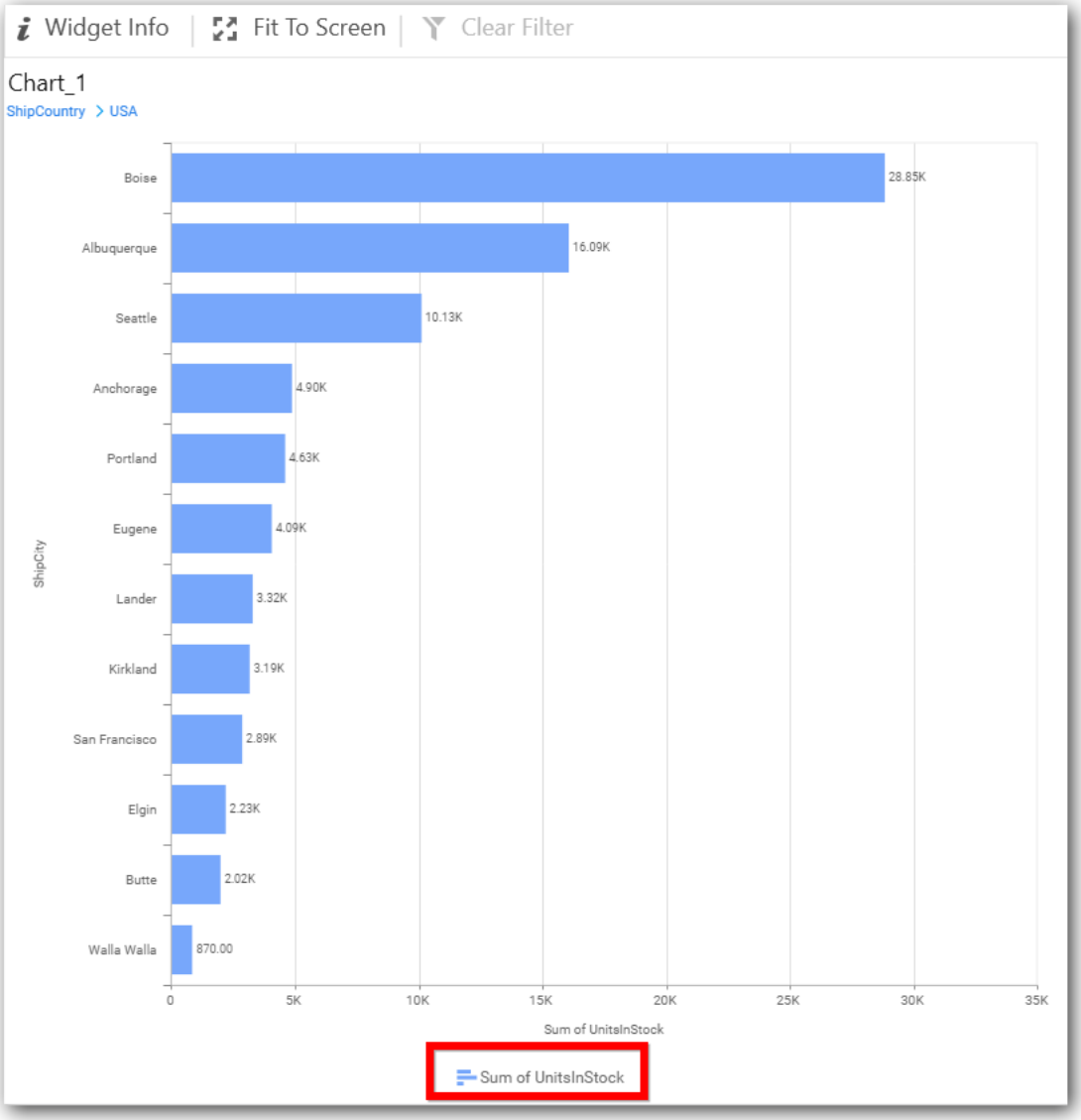


Drilled View



Show Legend

This allows you to toggle the visibility of legend. You can also set custom legend text (through the text area) for each legend series (selecting through the combo box).



Custom Legend Settings

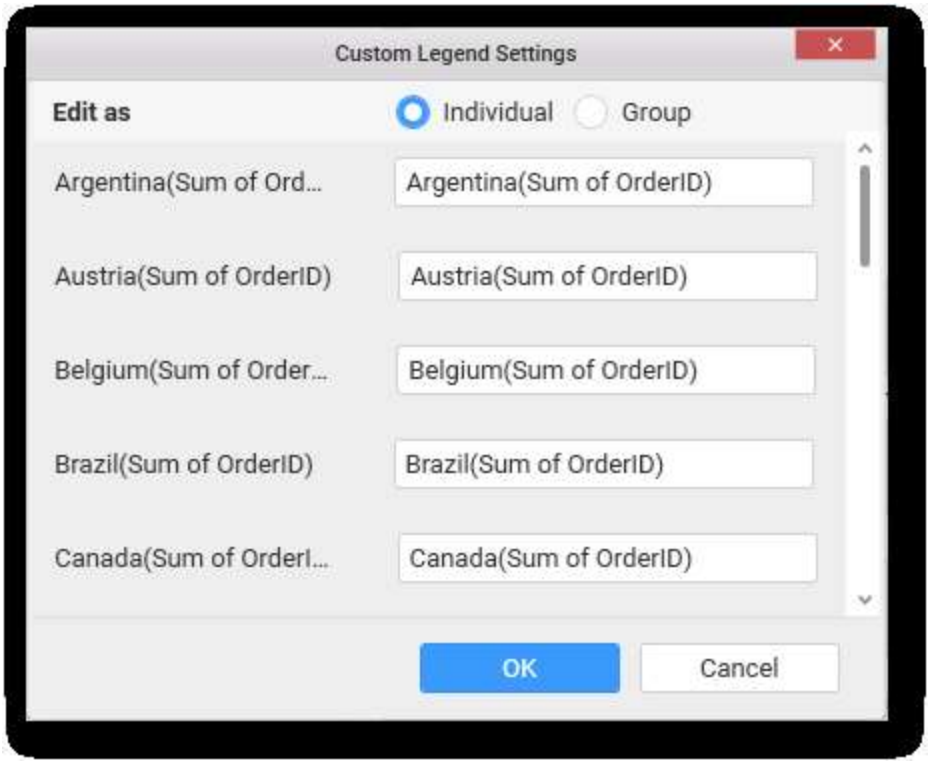
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

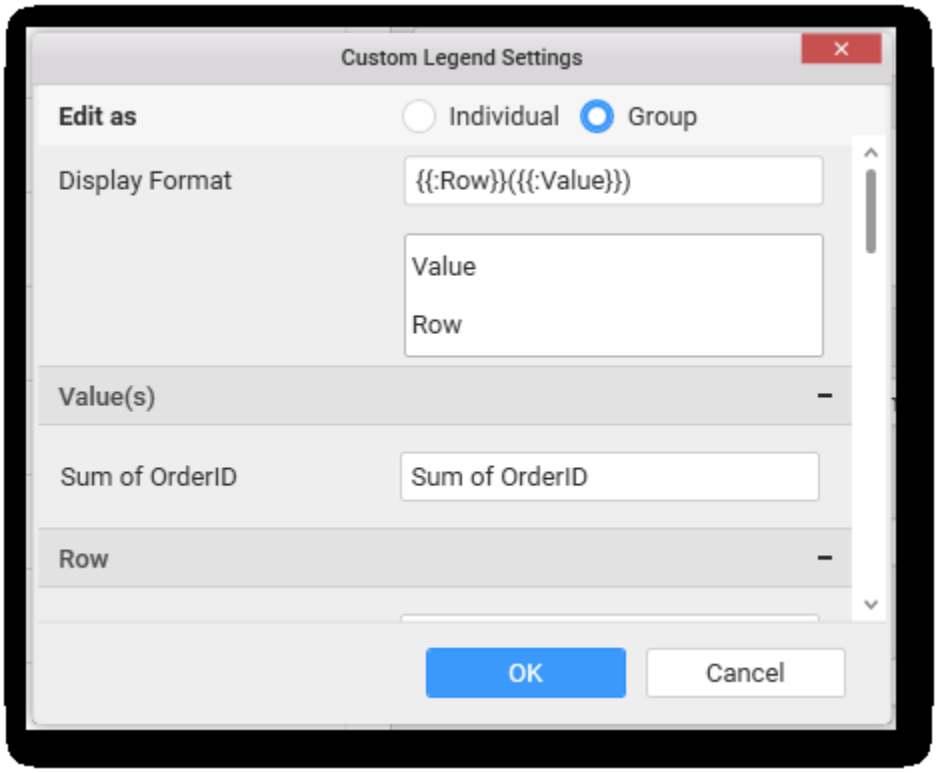
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.



Group

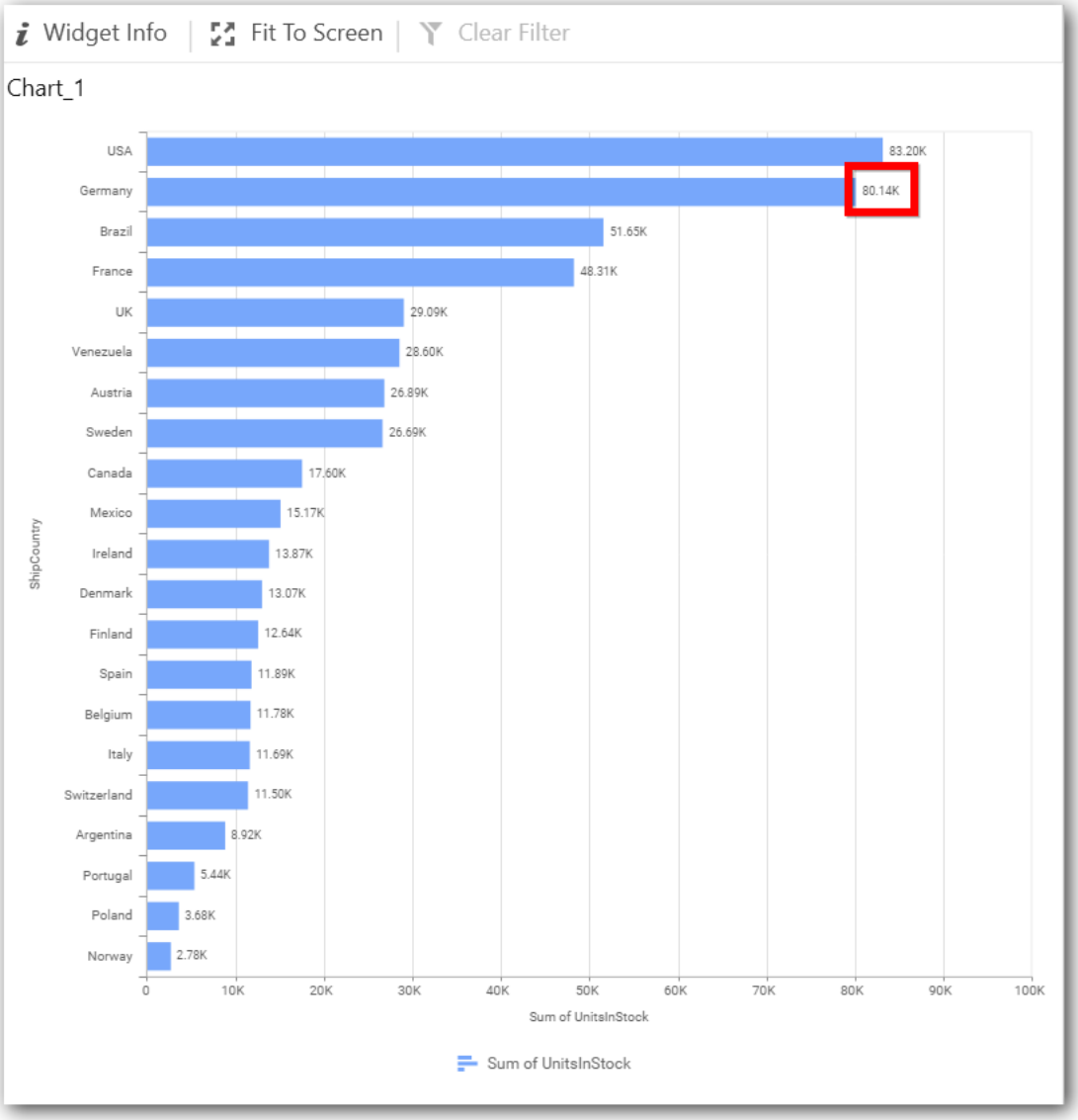
Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)

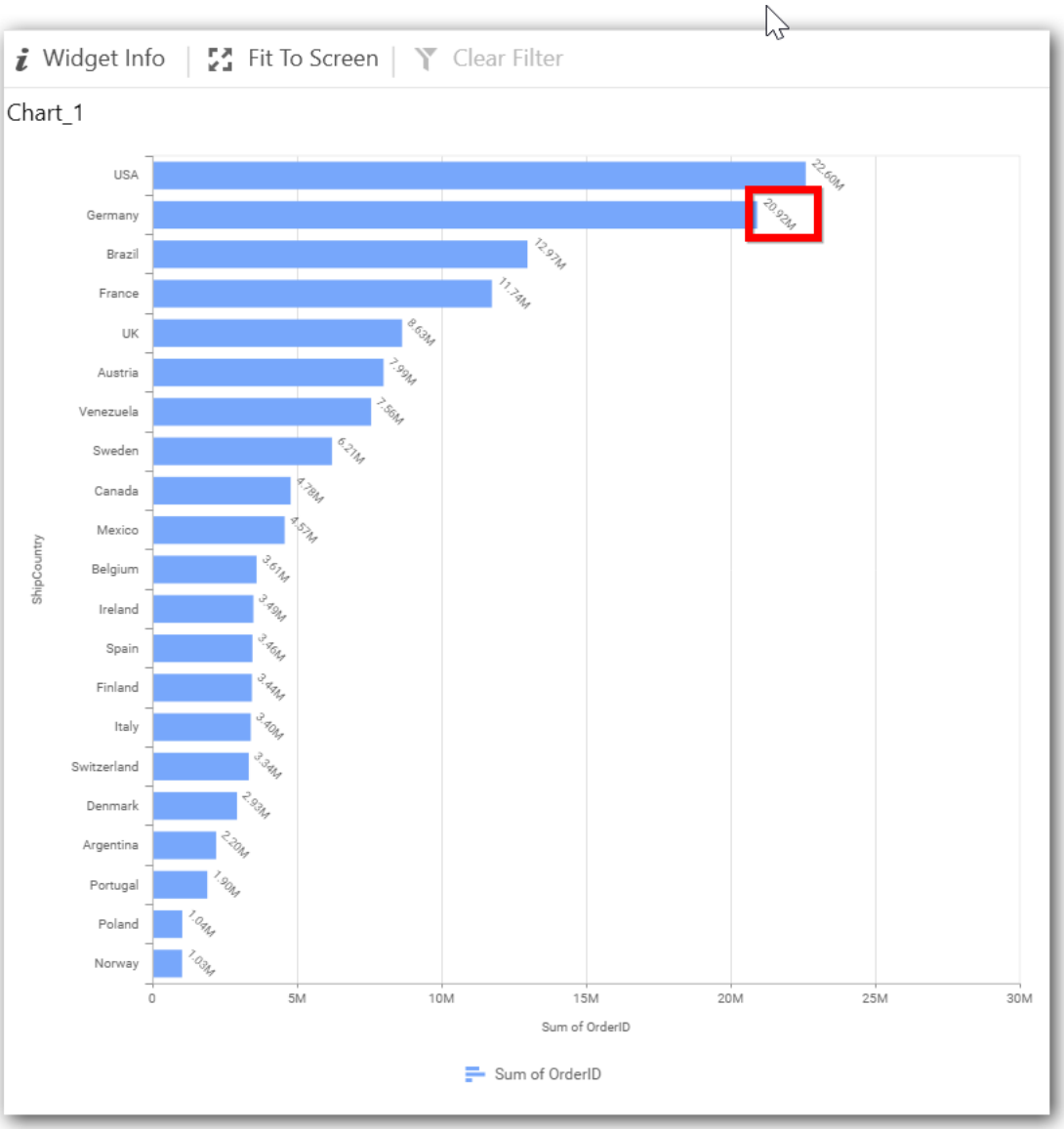
Show Value labels

This allows you to toggle the visibility of value labels. You can also set suffix to the value labels.



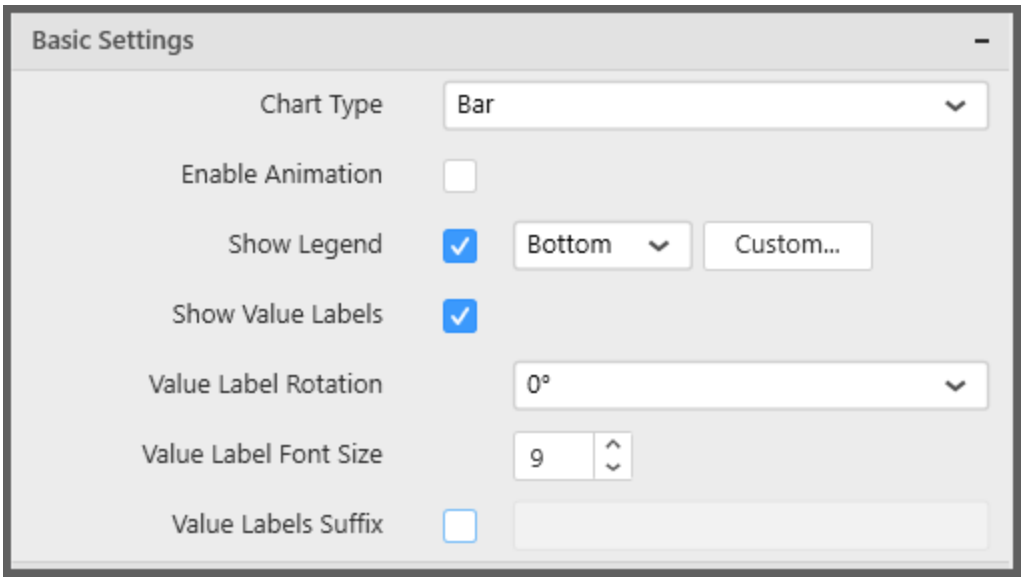
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.

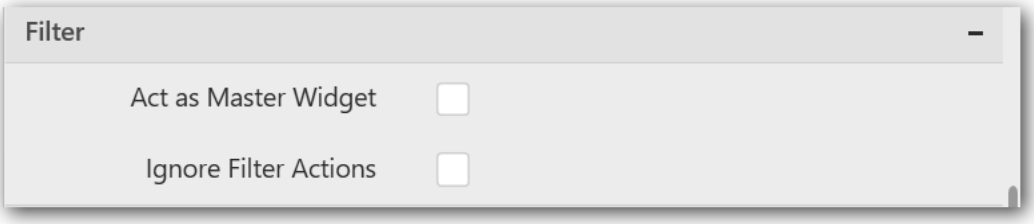


Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top ‘n’ filtering which can be flat or hierarchical, in this bar chart widget.

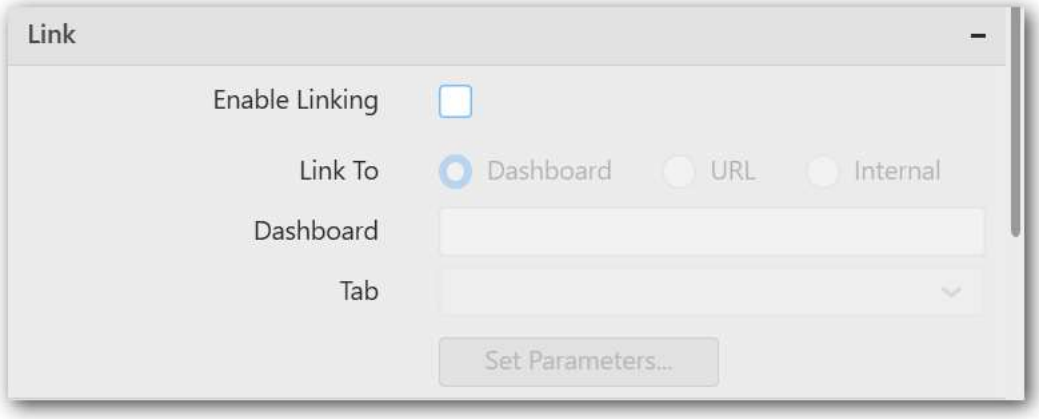
Act as Master Widget

This allows you to define this bar chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

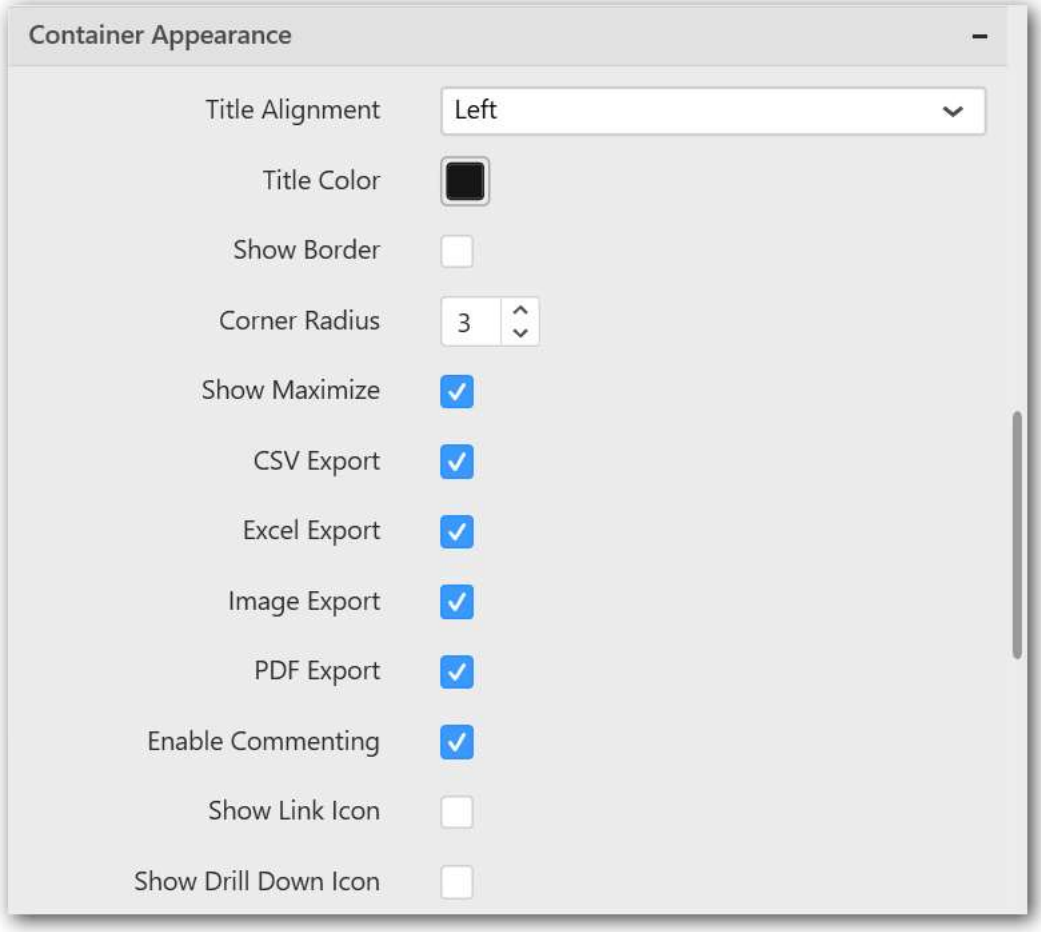
This allows you to define this bar chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Settings



Title Alignment

This allows you the handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this bar chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this bar chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this bar chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

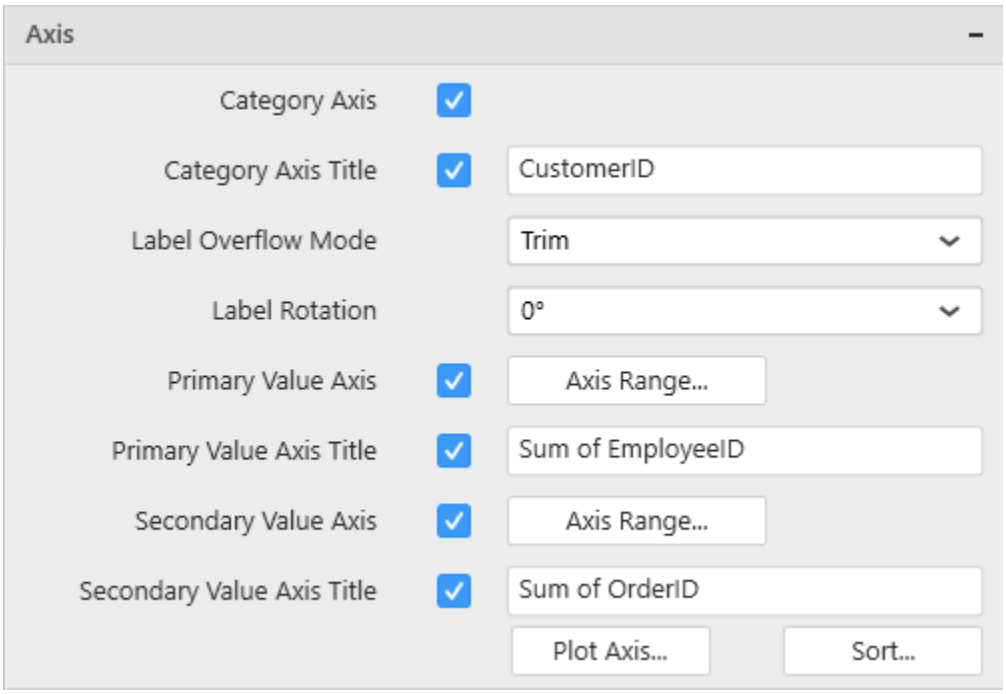
Image Export

This allows you to enable/disable the image export option for this bar chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Axis Settings

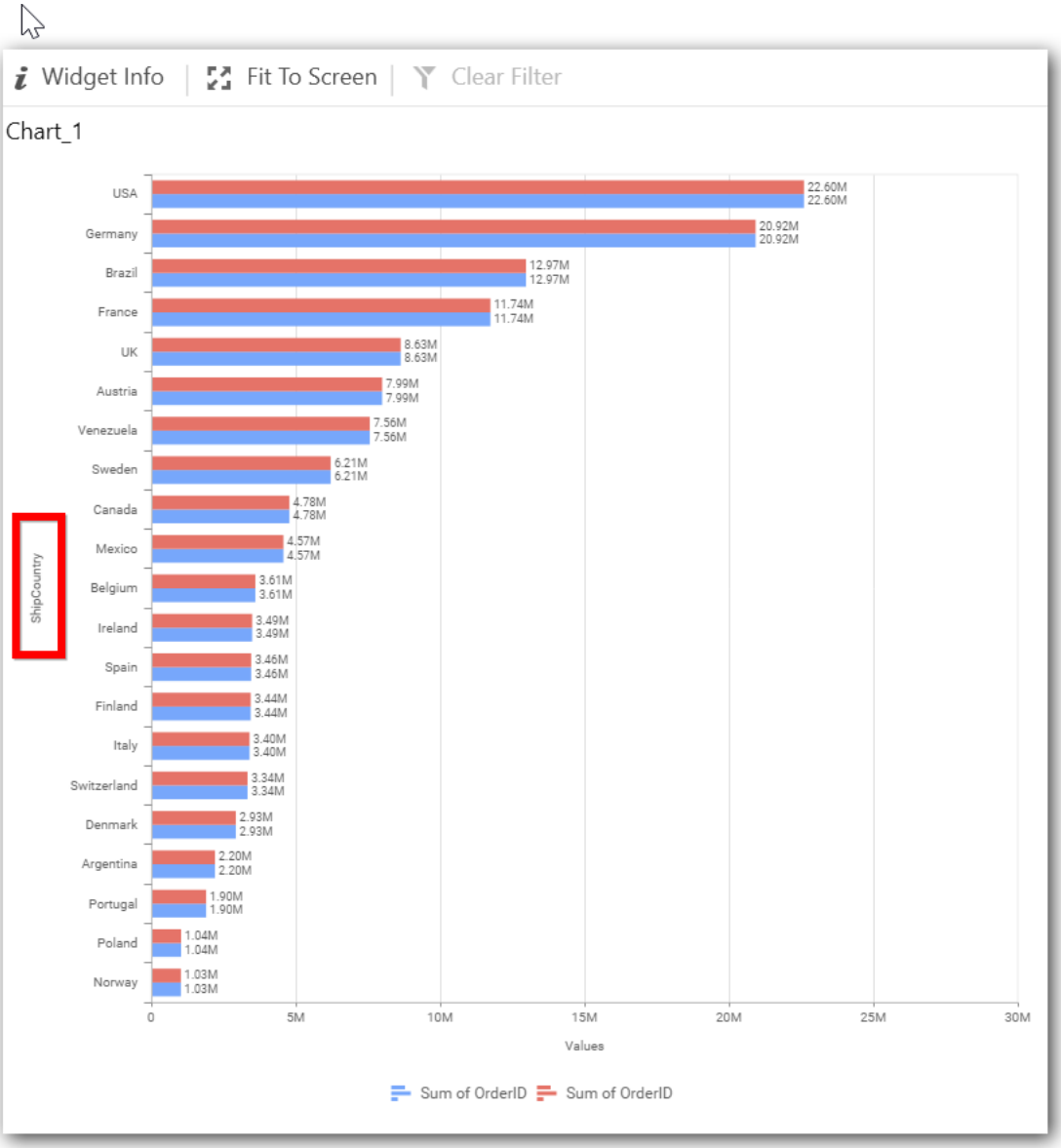


Category Axis

This allows to enable or edit the option of Category Axis. It will reflect in chart area x-axis name.

Category Axis Title

This allows you to toggle the visibility of Category axis title.

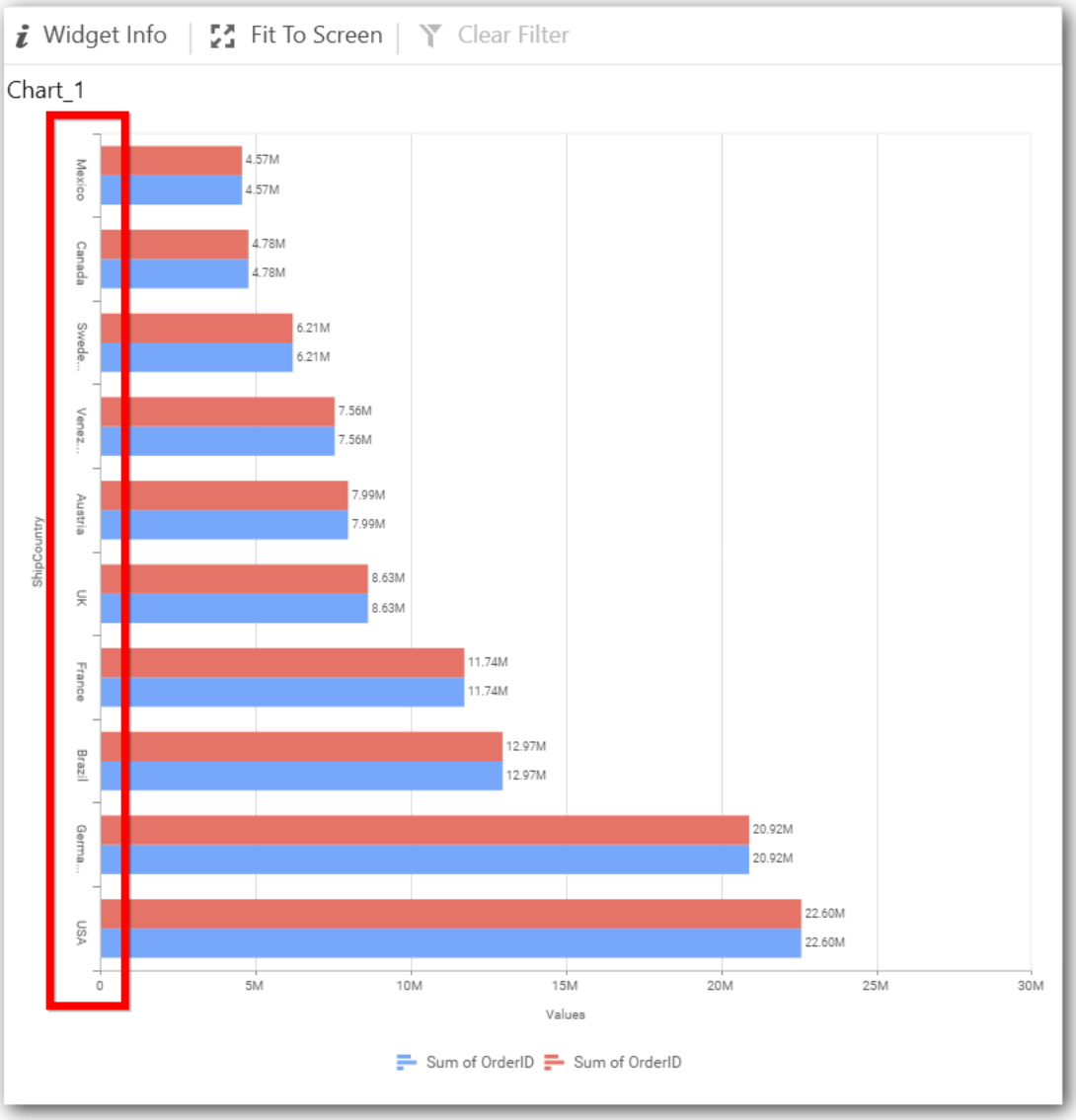


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

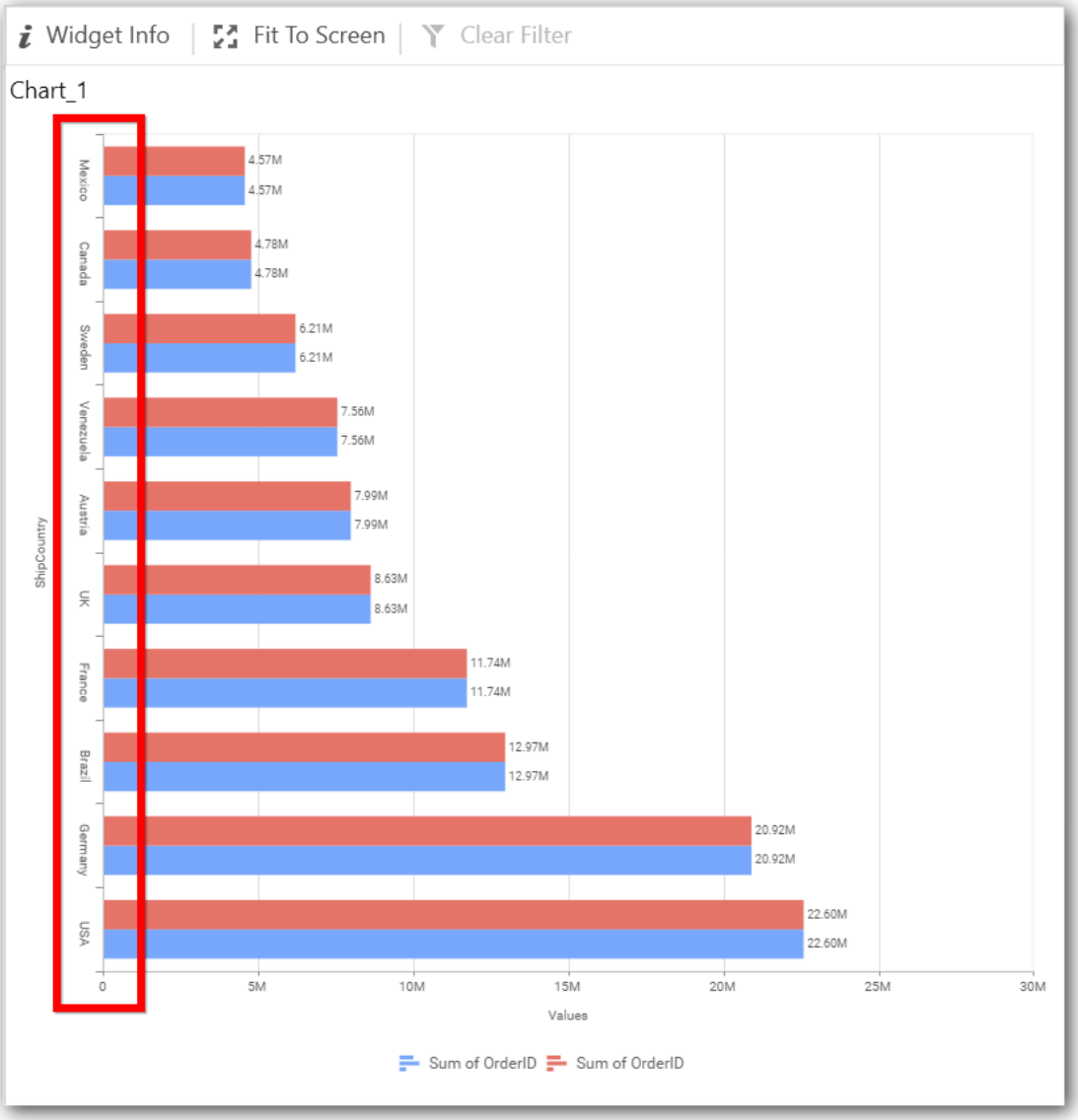
Trim

This option trims the end of overlapping label in the axis.



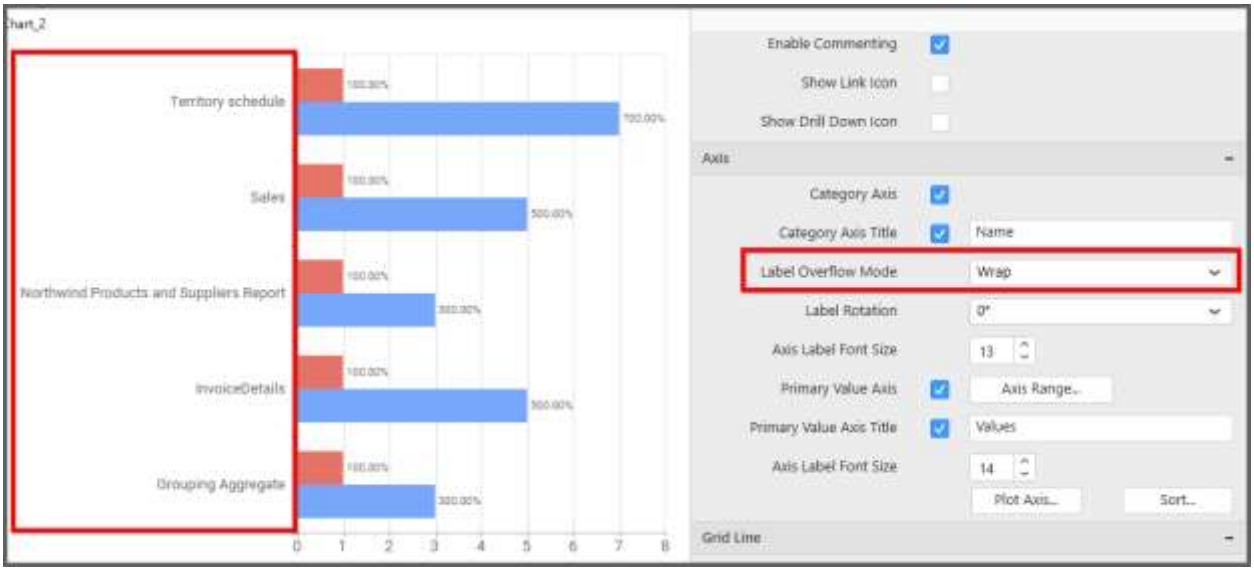
Hide

This option hides the overlapping label in the axis.



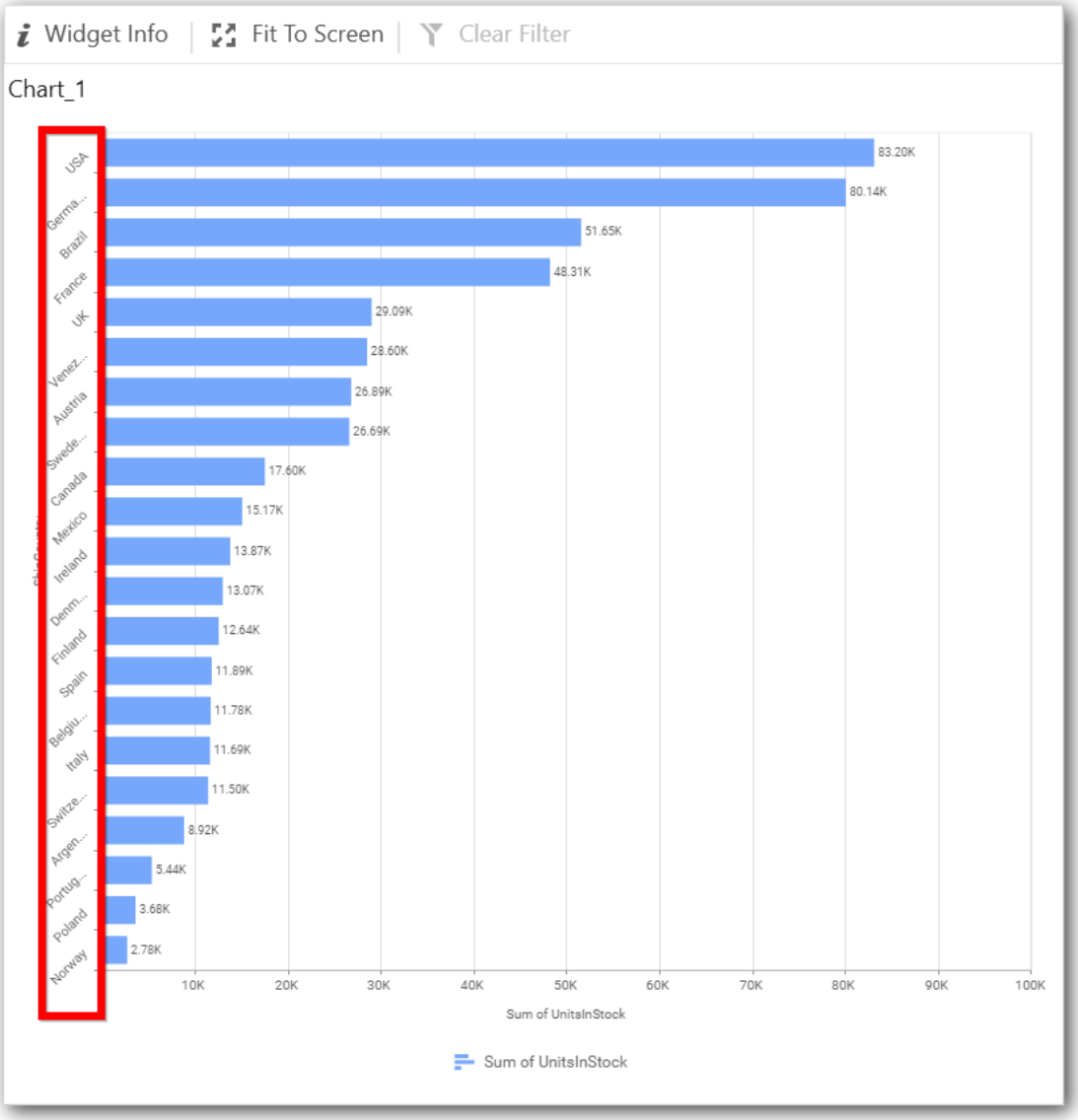
Wrap

This option wraps the lengthy label text in the axis.



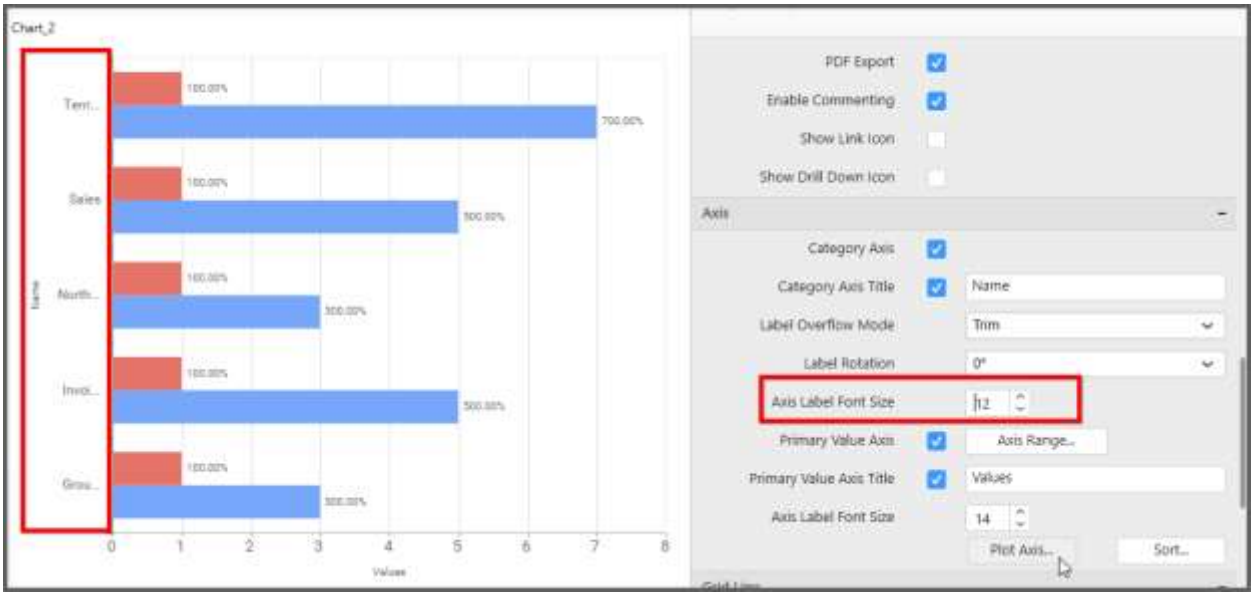
Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



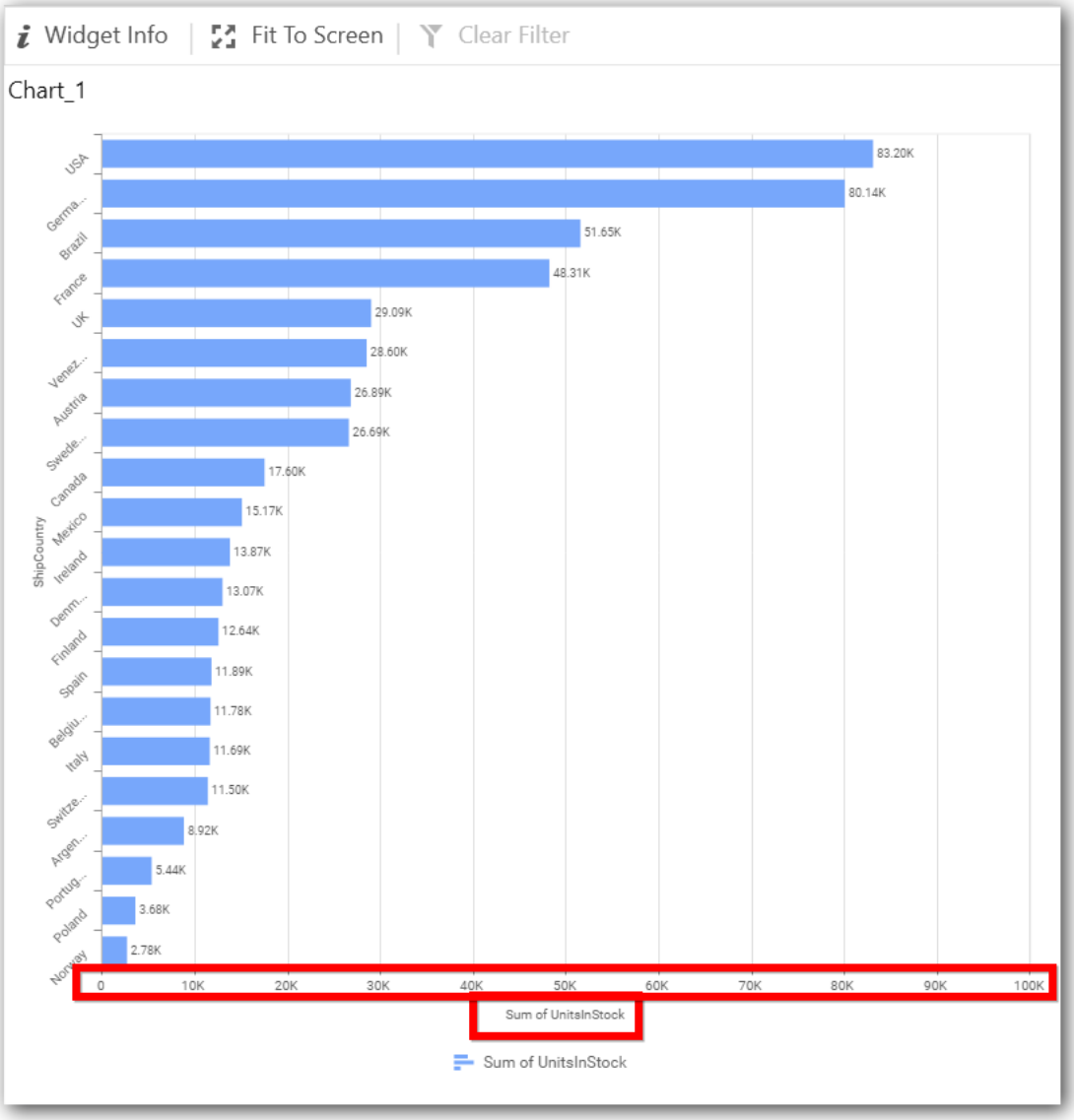
Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.



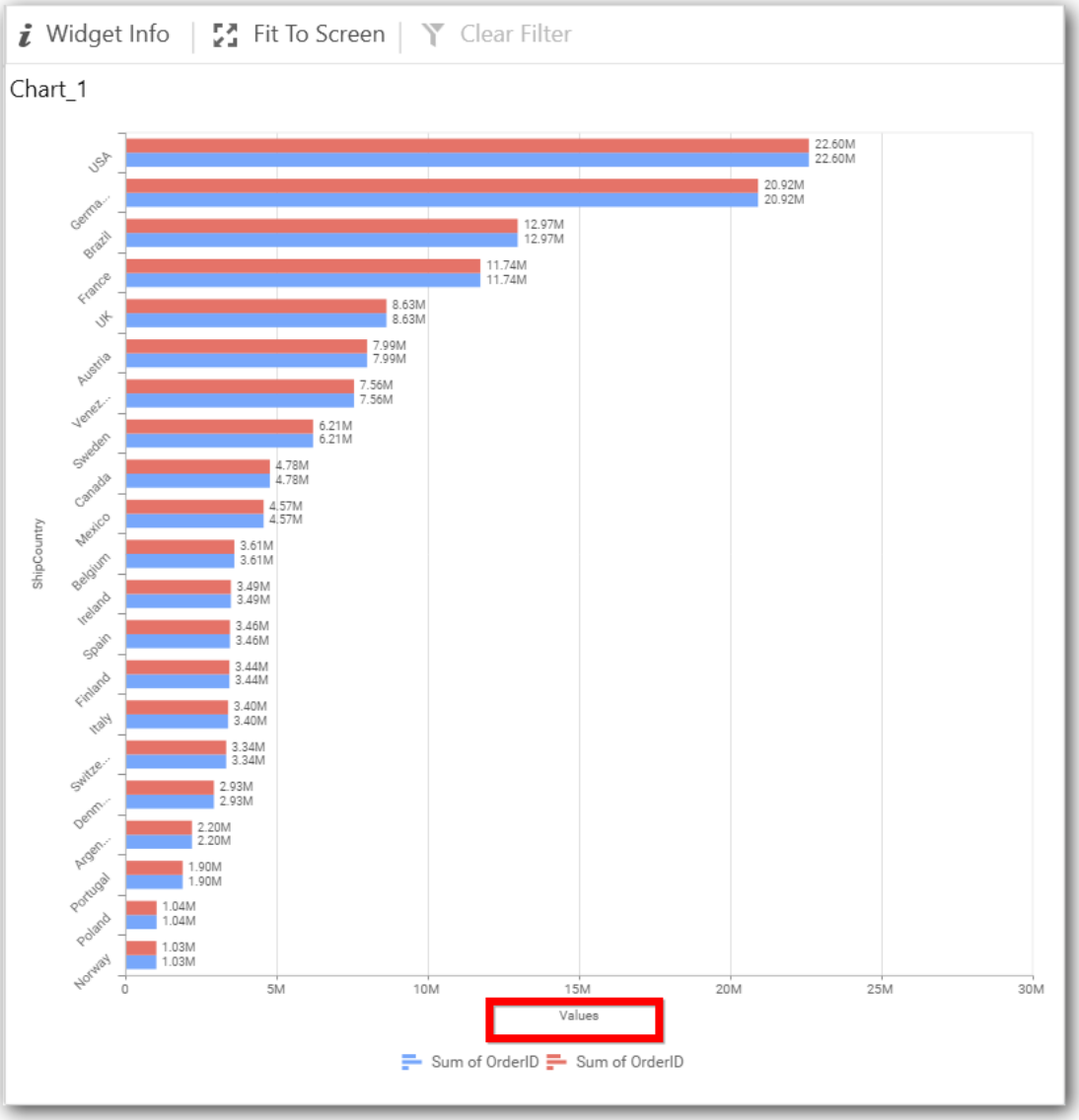
Primary Value Axis

This allows you to toggle the visibility of primary value axis title and labels.



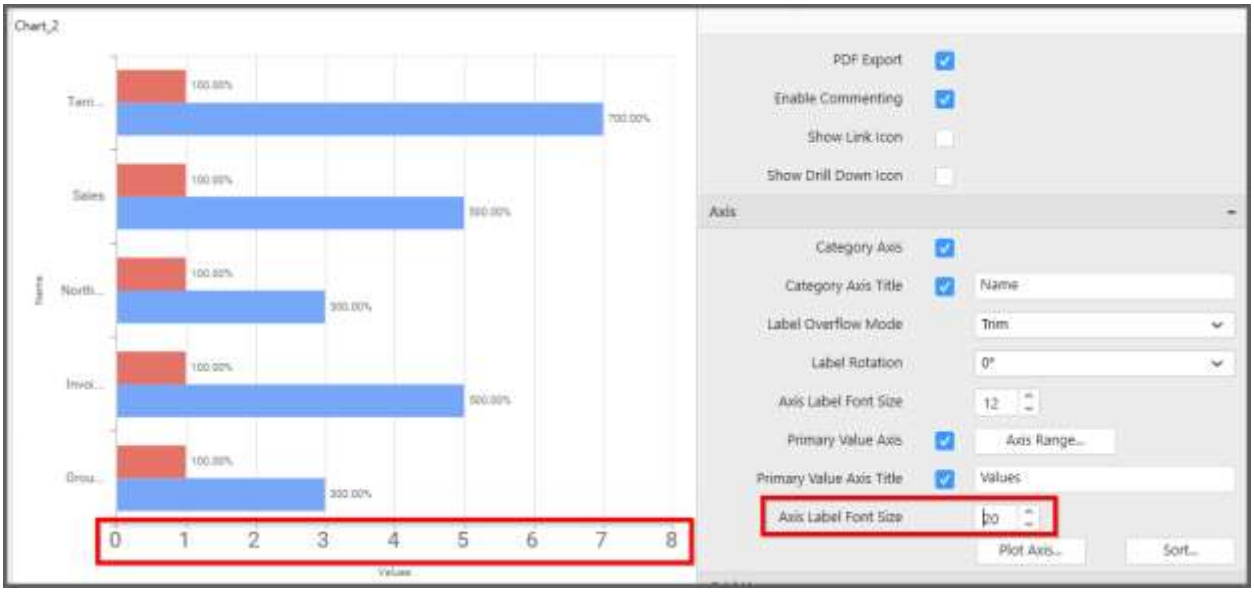
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

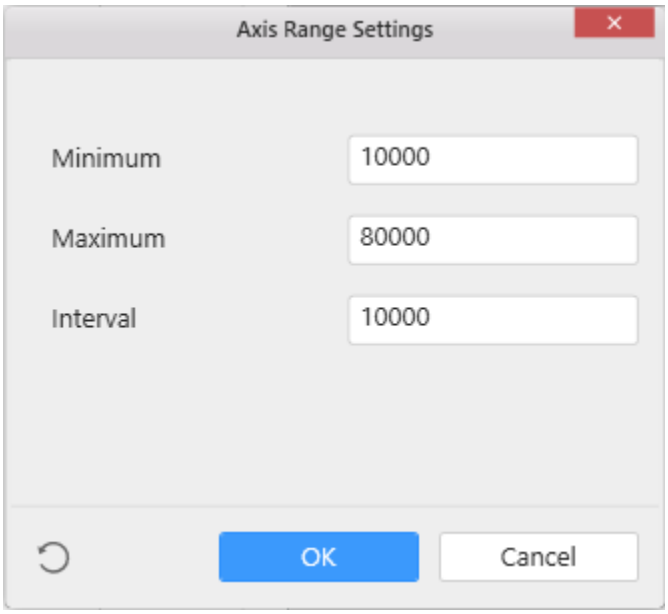


Primary Value Axis Range

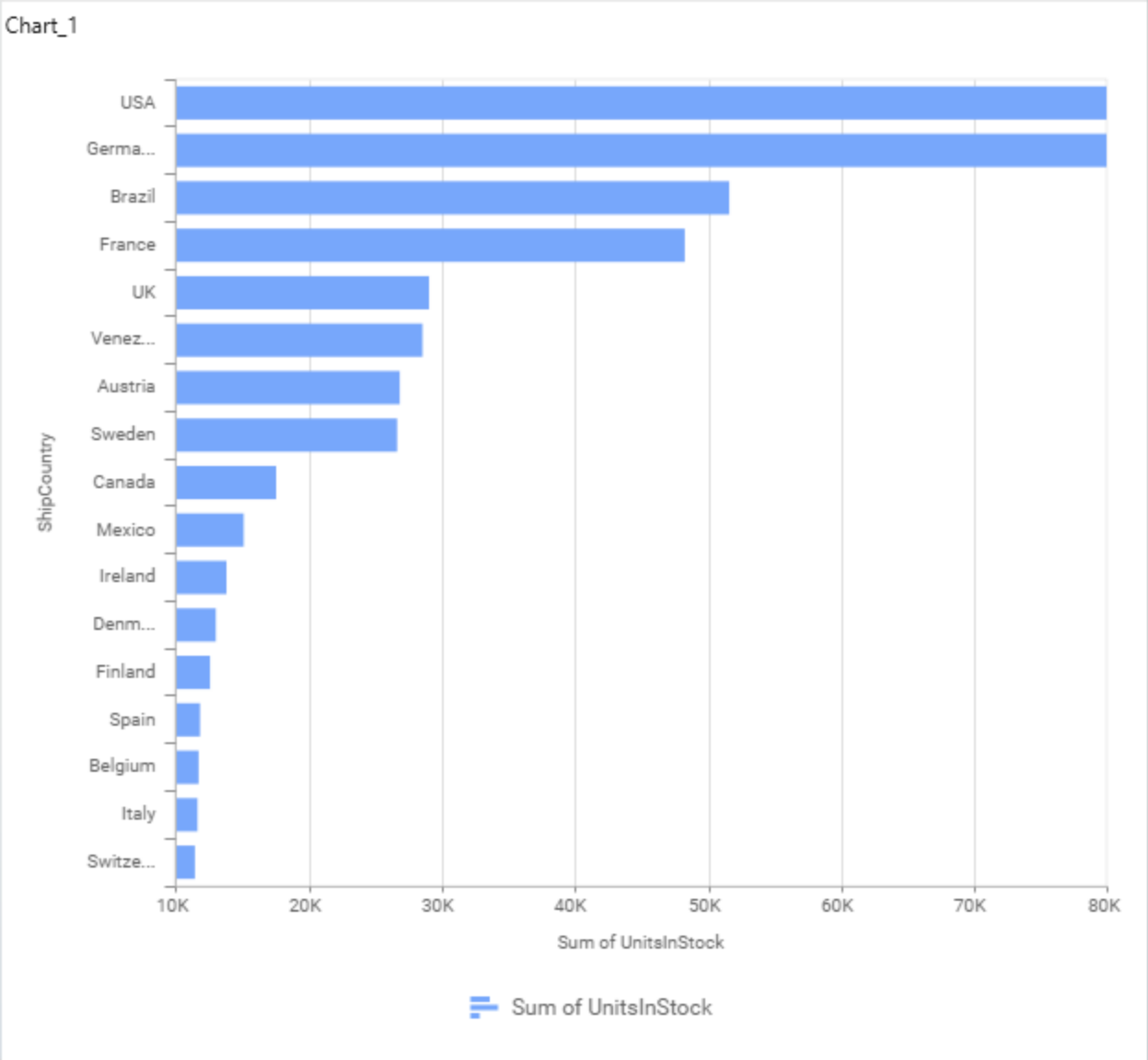
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.

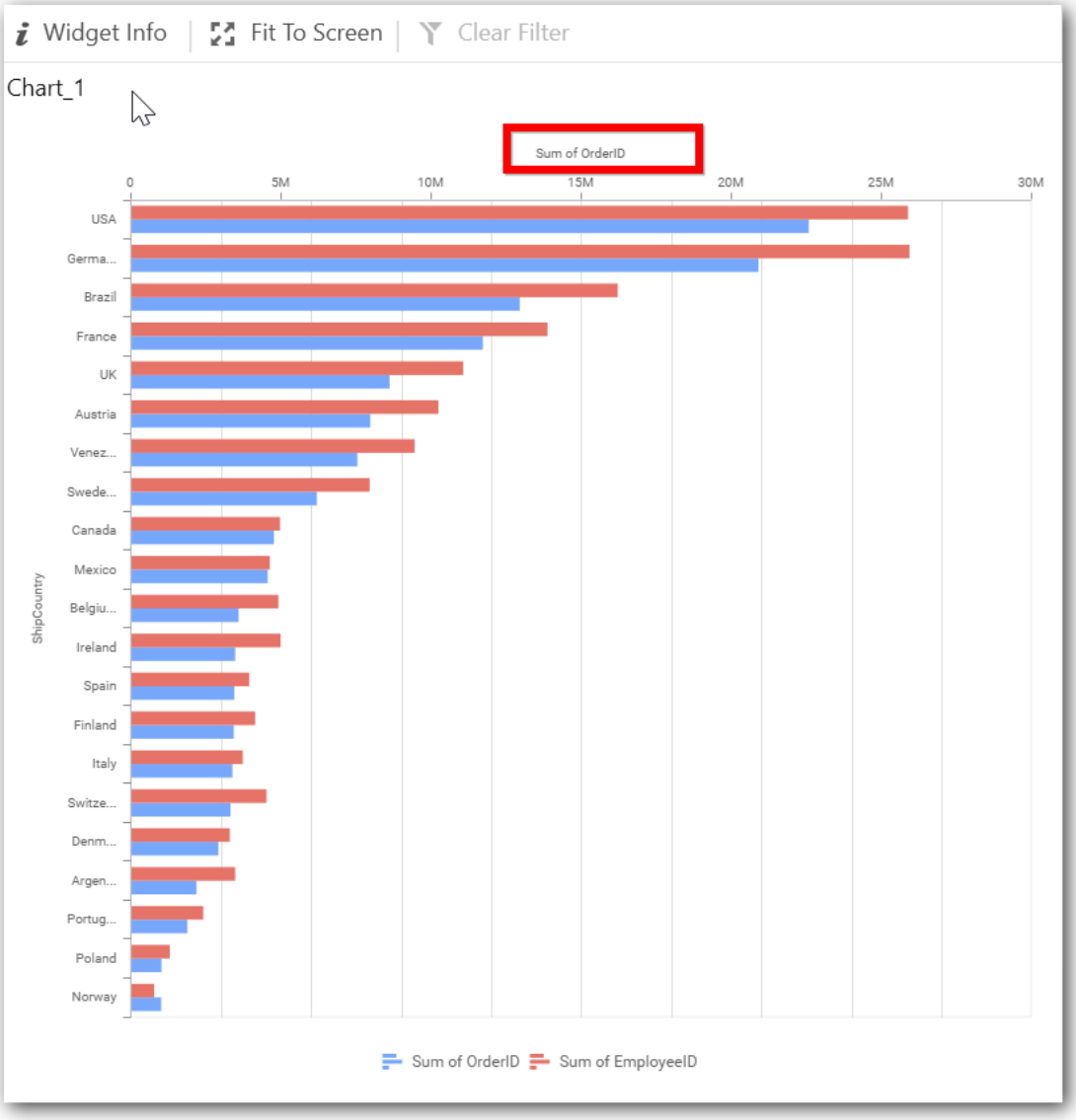


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



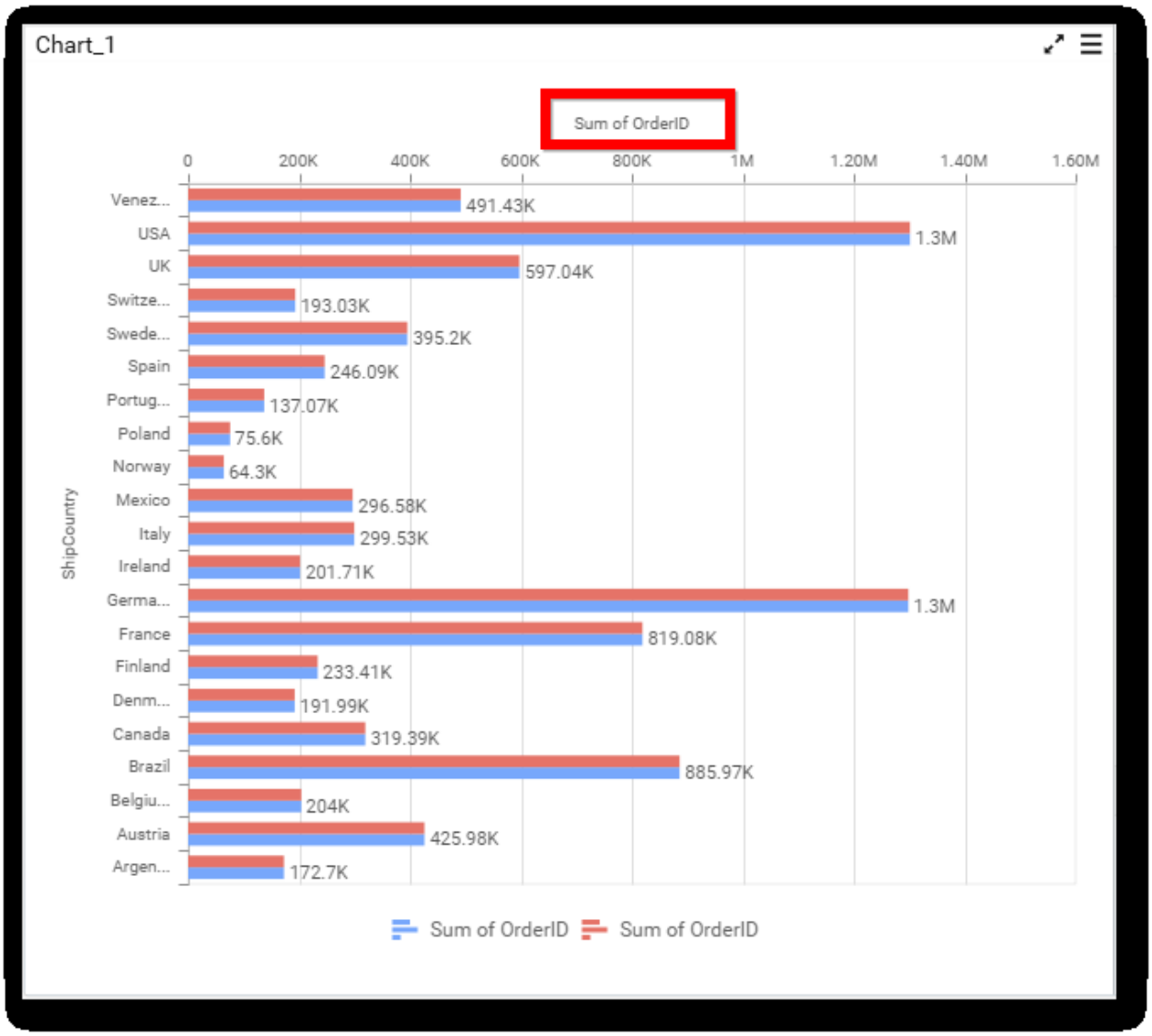
Secondary Value Axis

This allows you to toggle the visibility of secondary value axis title and labels.



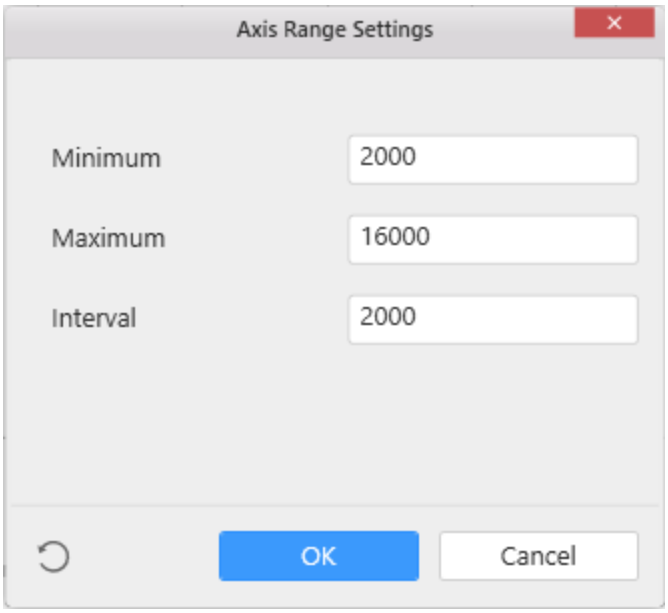
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

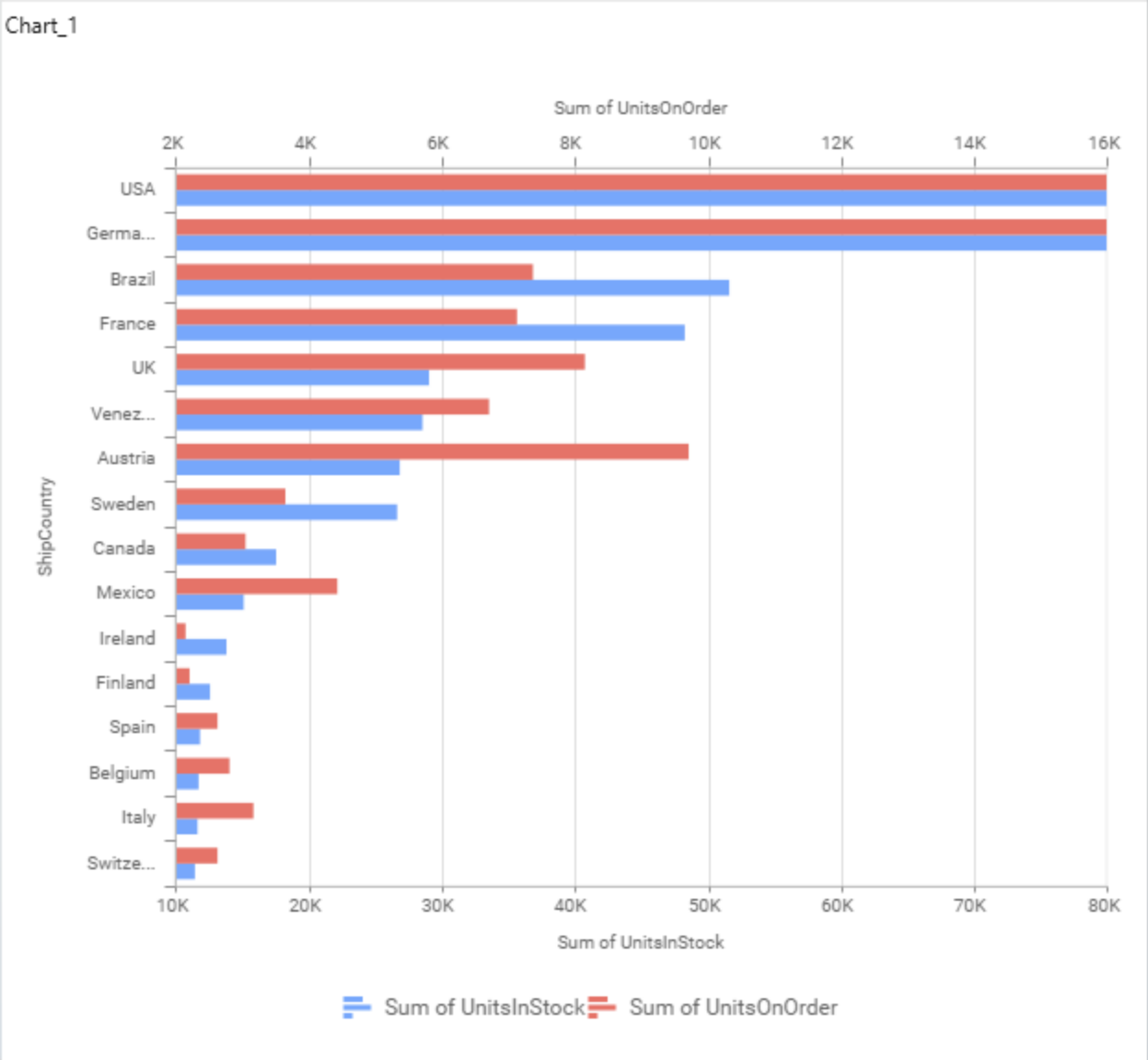


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

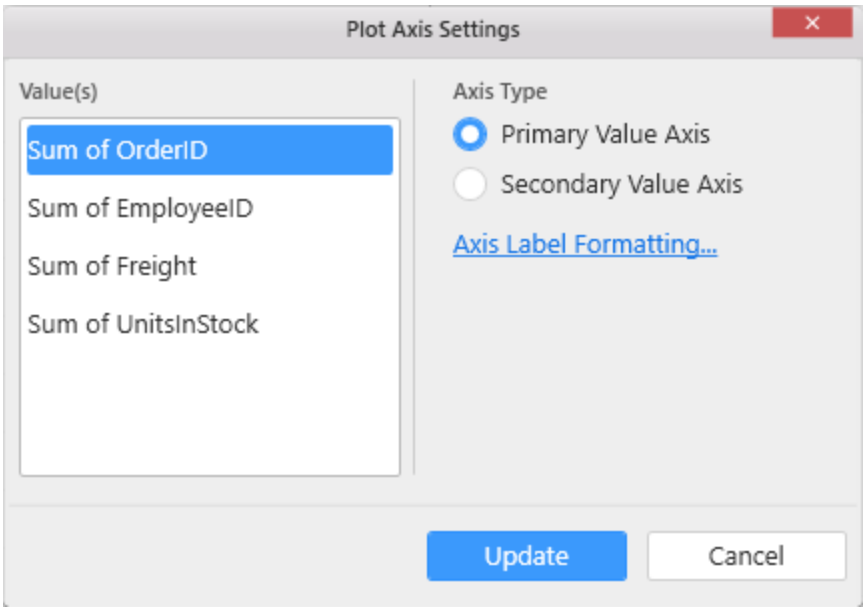


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



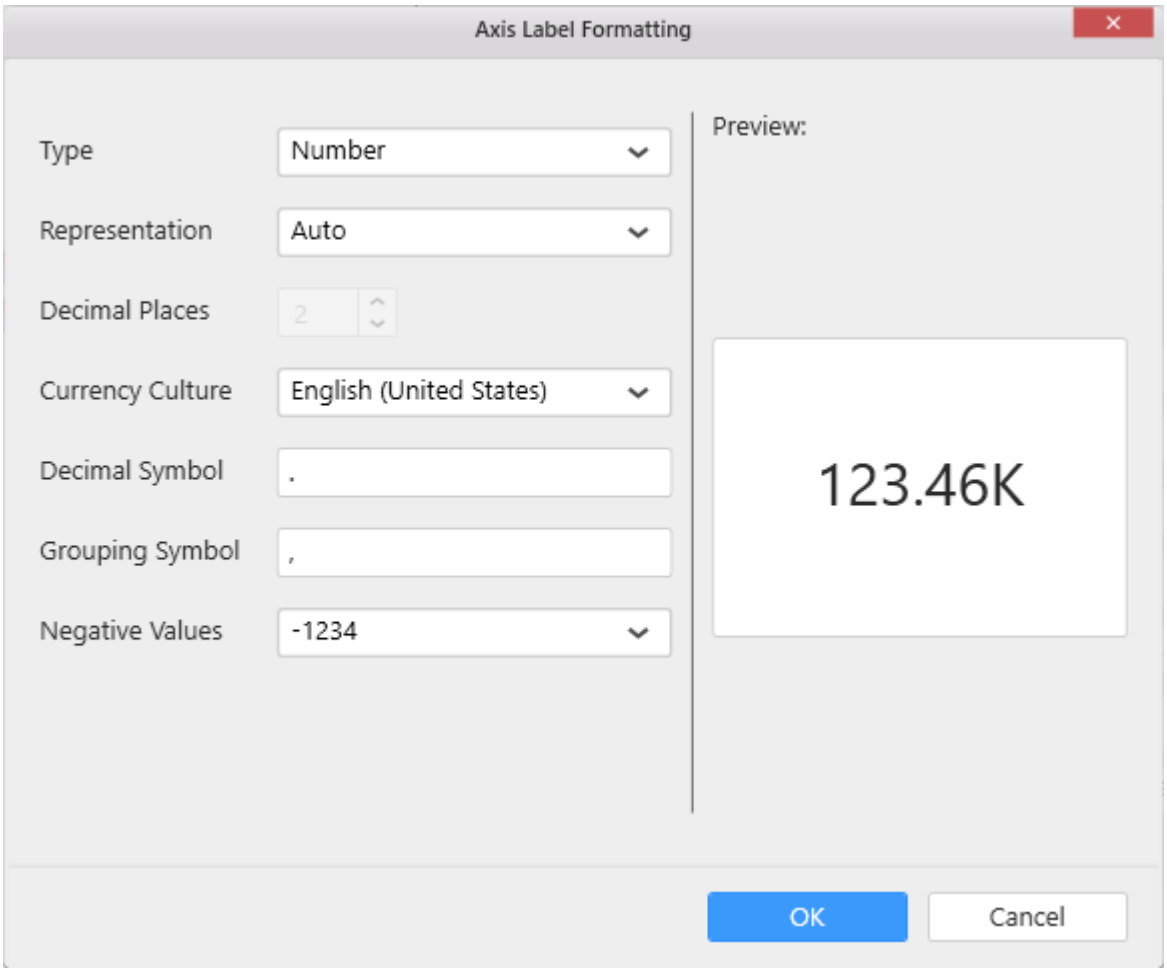
Plot Axis Settings

This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



Axis Label Formatting

This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.



Sort Order

This allows you to define the sort order for each measure column added.

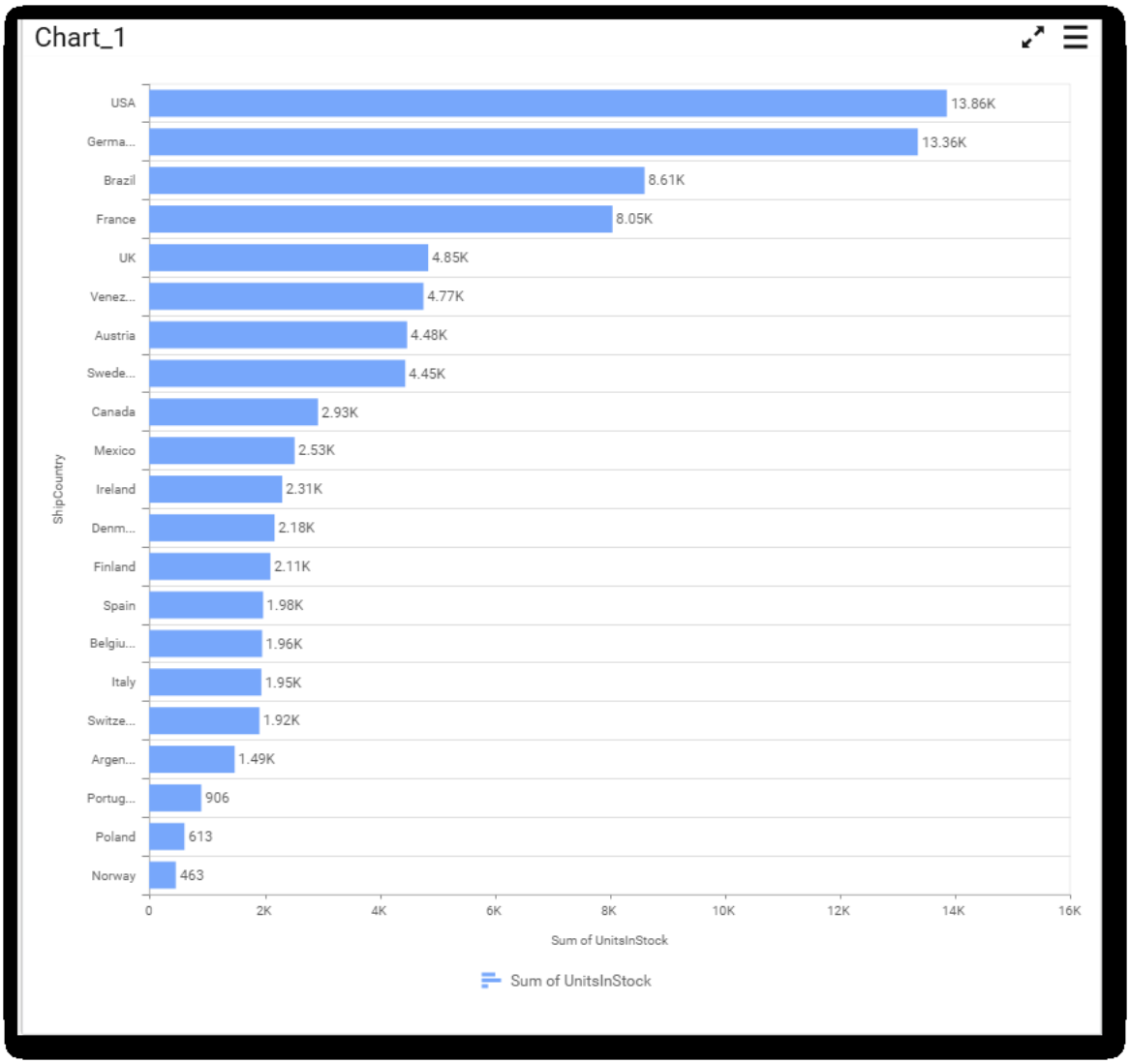
Grid Line Settings

Primary value Axis

This allows you to toggle the visibility of Primary Value Axis gridlines.

Category Axis

This allows you to toggle the visibility of Category Axis gridlines.



Secondary Value Axis

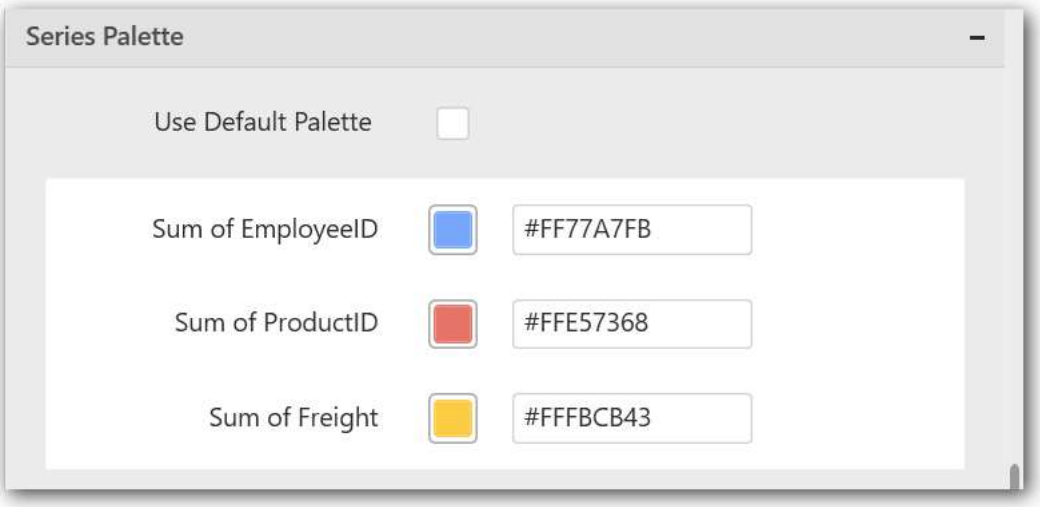
This allows you to toggle the visibility of secondary value axis gridlines.

Series Palette

This allows you to customize the chart series color through Series Palette section.

Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



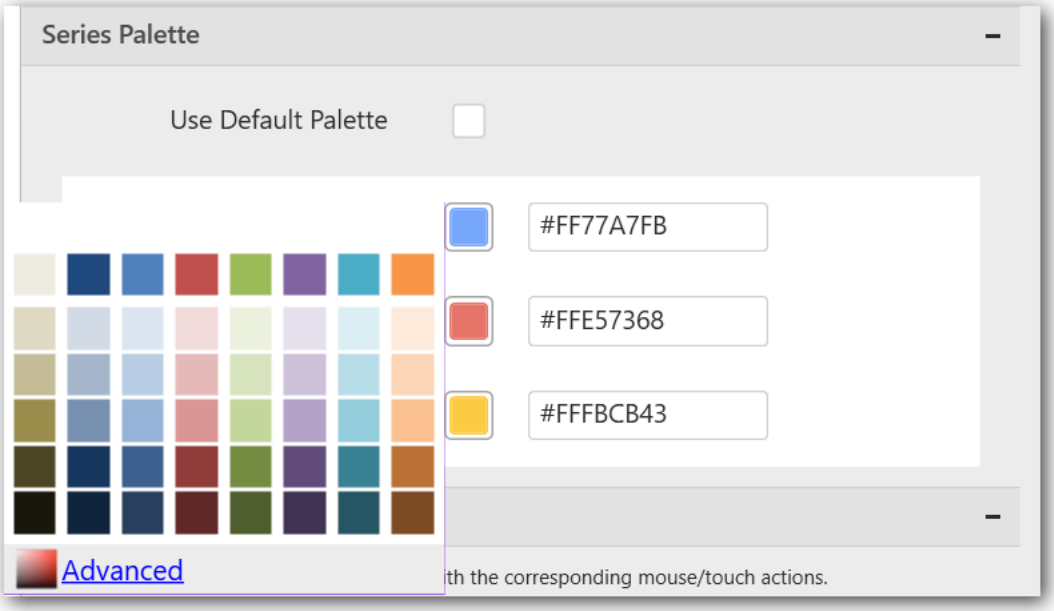
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.

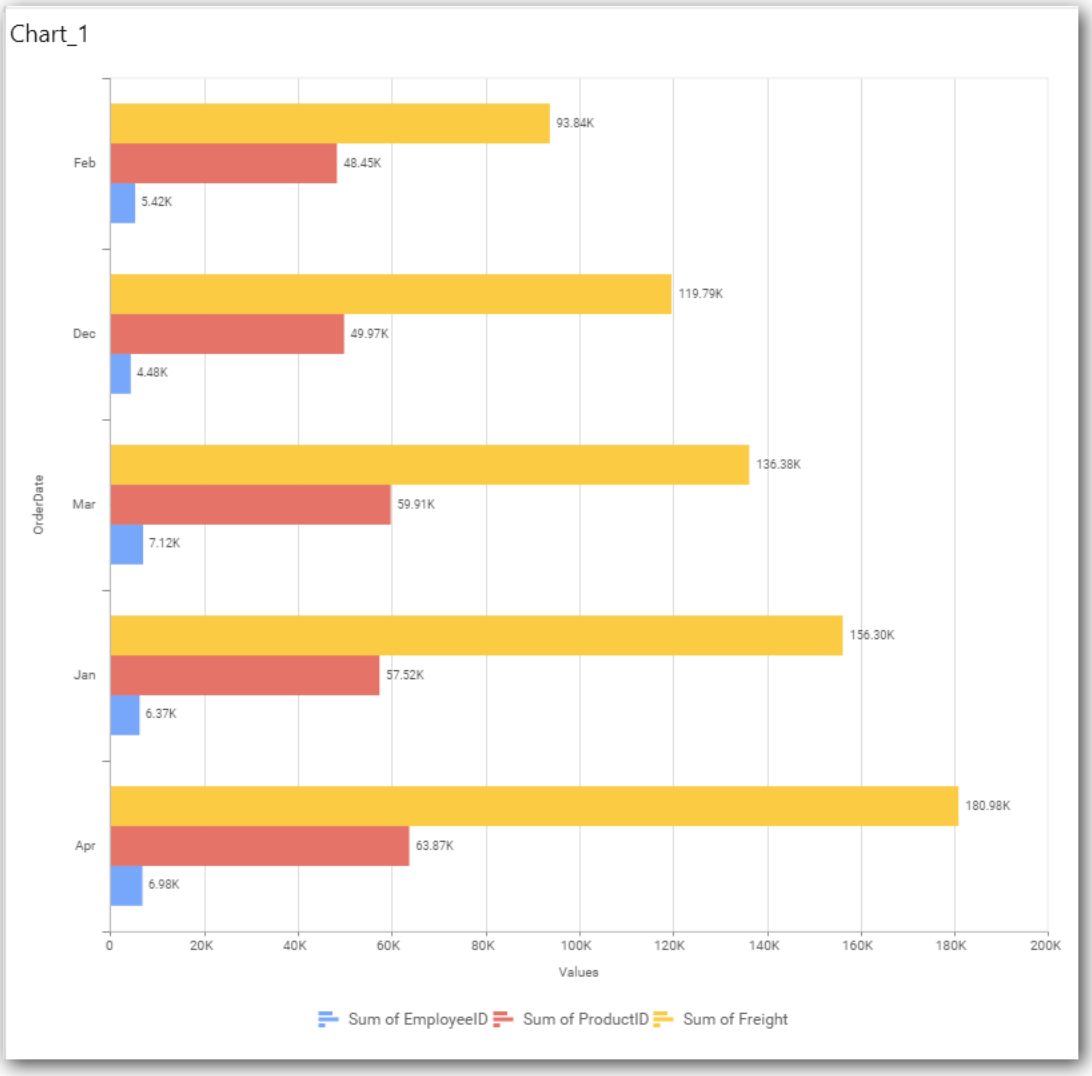
Information: In Syncfusion Dashboard Dashboard Application, The HEX color codes are used for color palette. So, we have to enter **8 characters(Except #) for color code** and we have to append "FF" as a prefix of 6 character HEX color code.

Information: **Note:** FF is a transparency value (Range from 00 to FF.

Information: ***For Example,***

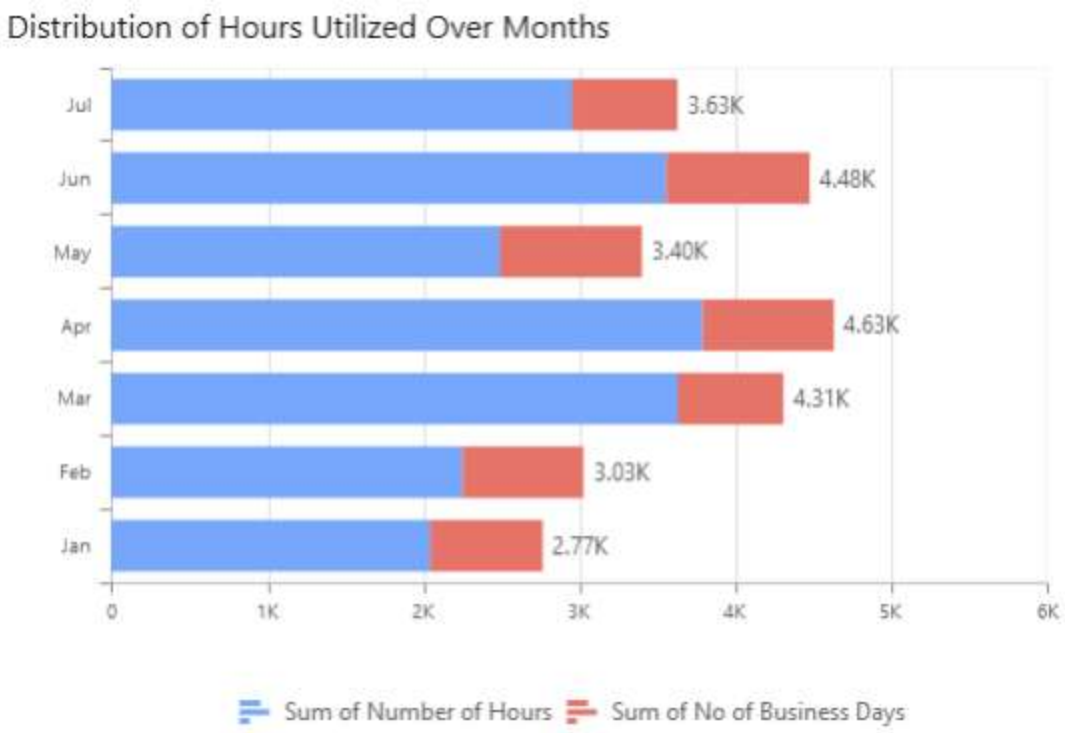
Information: You have to enter FF0066FF instead of 0066FF.





Stacked Bar Chart

Stacked Bar Chart allows you to compare multiple measures through bars stacked one after the other horizontally.

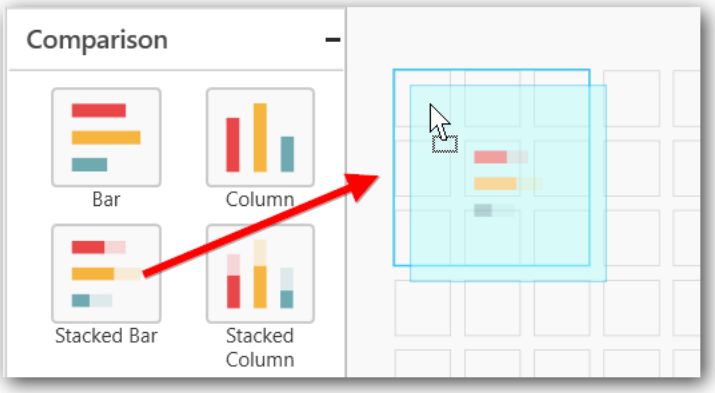


How to configure the flat table data to Stacked Bar Chart?

Stacked Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

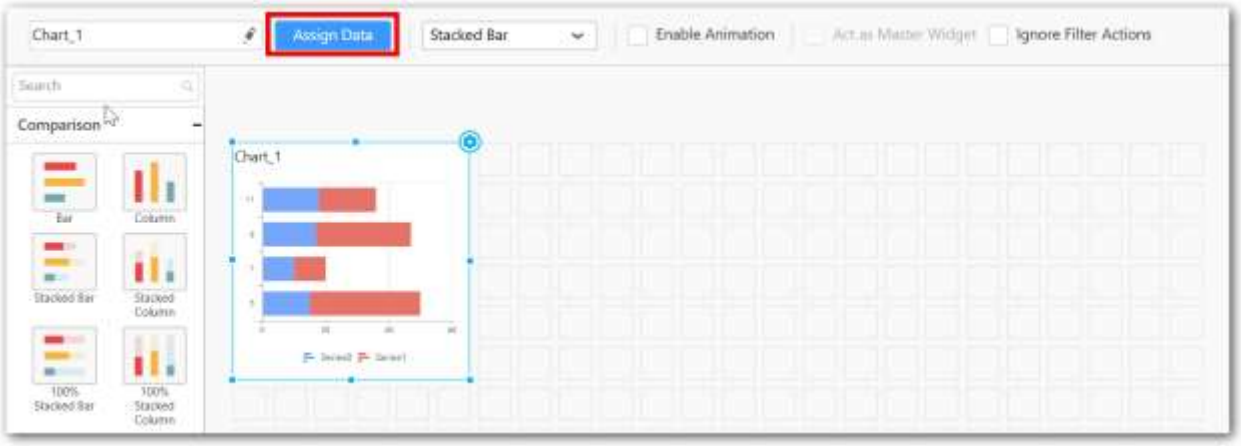
Follow the steps to configure data to stacked bar chart.

Drag and drop the stacked bar chart into canvas and resize it your required size.

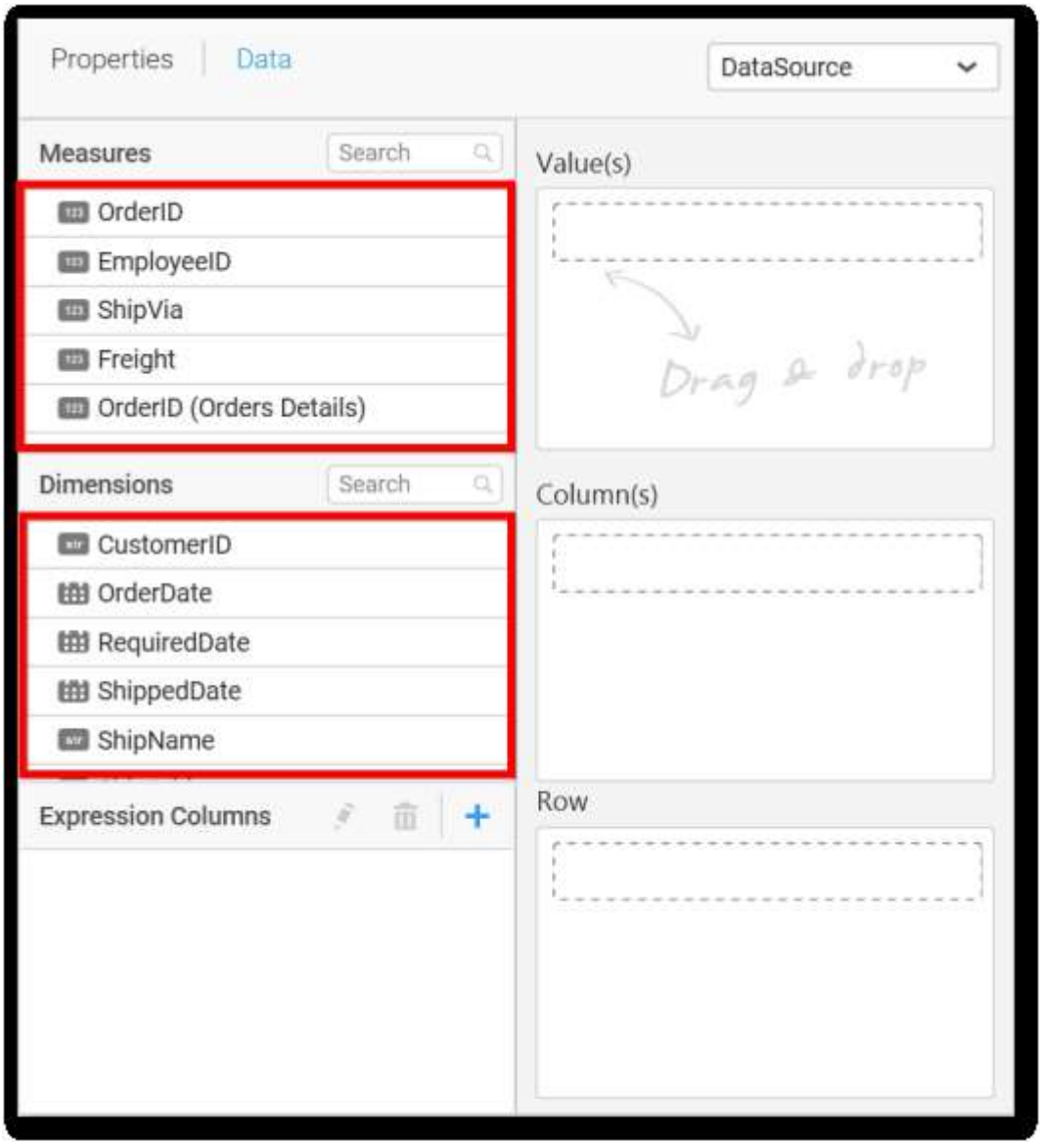


Connect to data source.

Focus on the stacked bar chart and click on Assign Data.



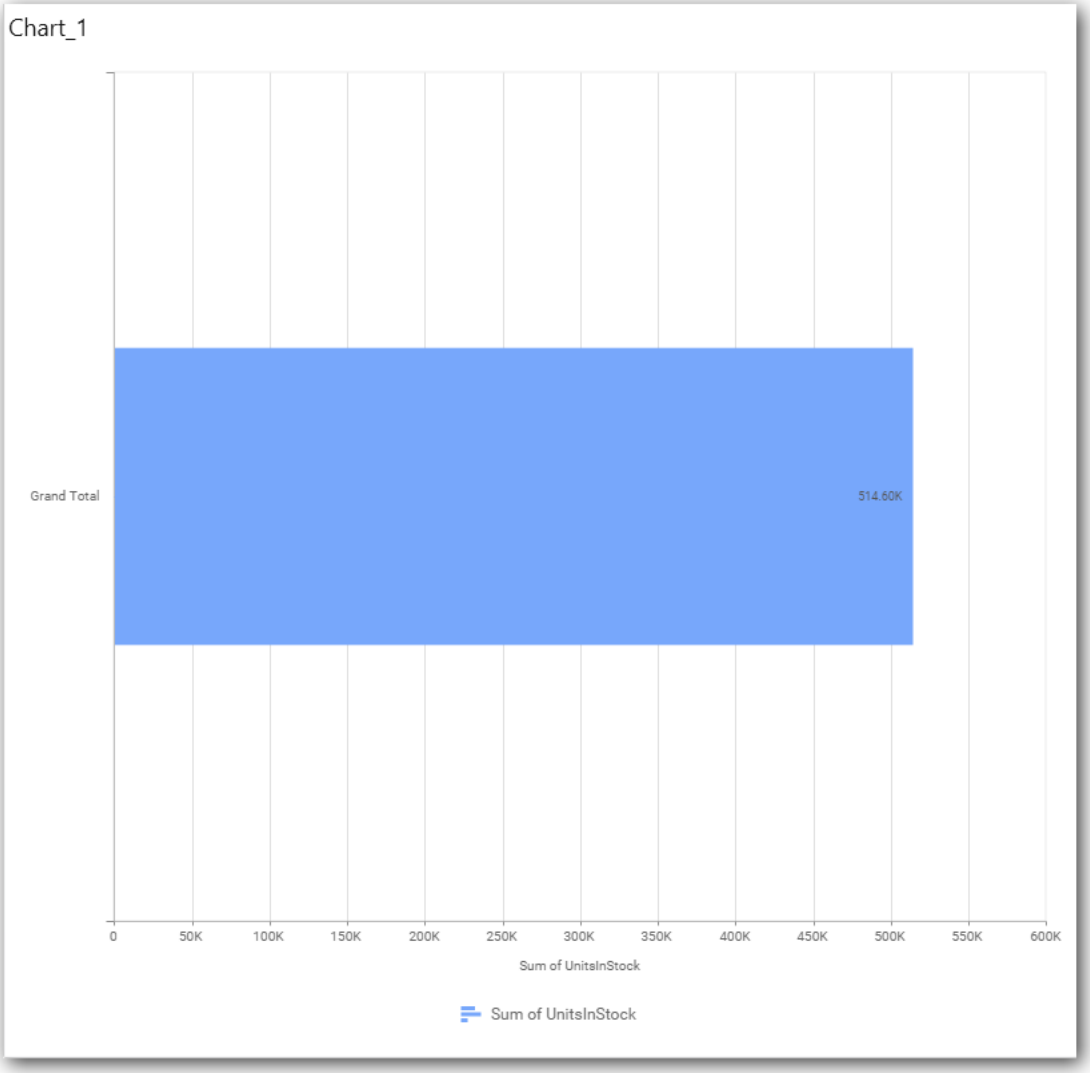
The data pane will be opened with available Measures and Dimensions from the connected data source.



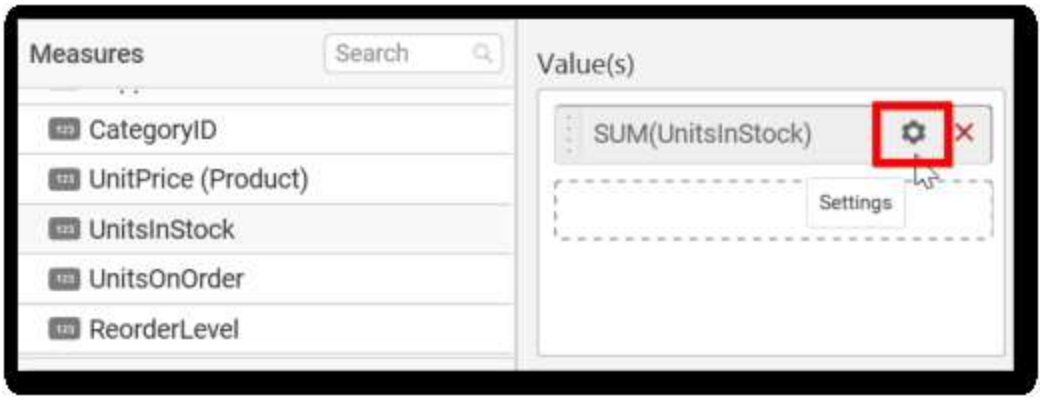
Assigning Value(s)

Drag and drop the Measure into Value.

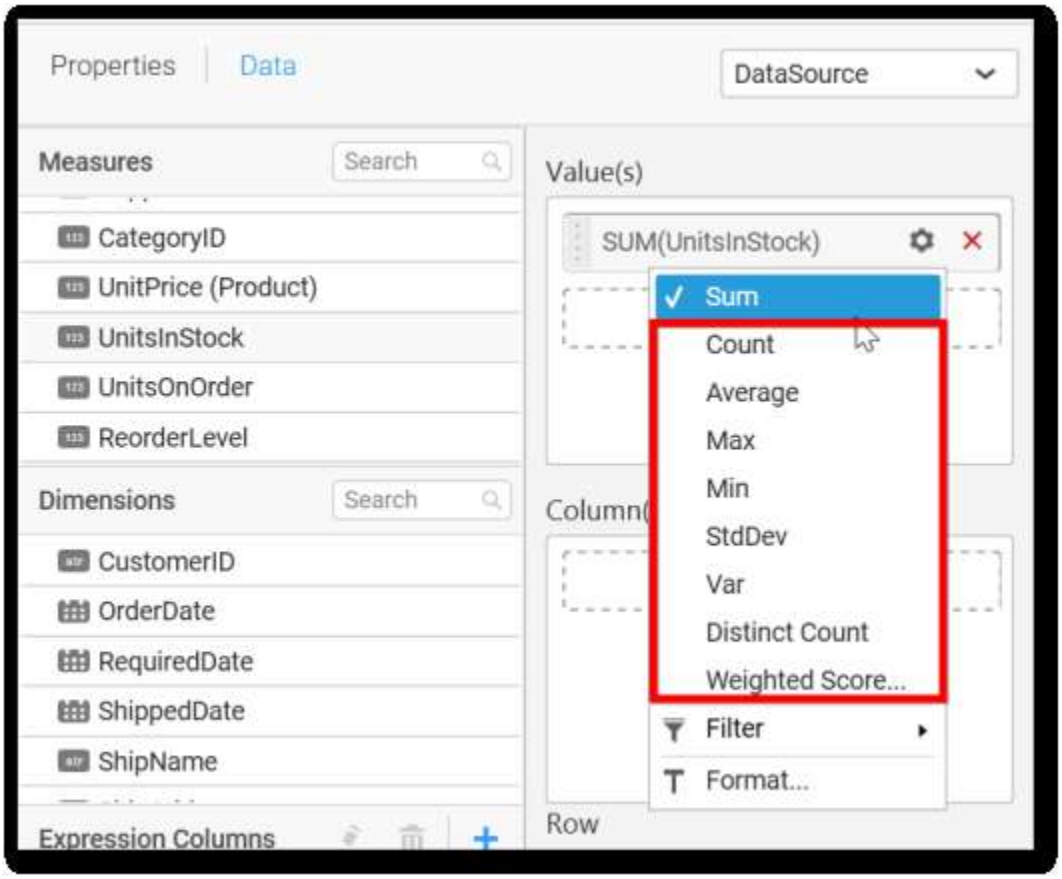
Now the chart will be rendered like this.



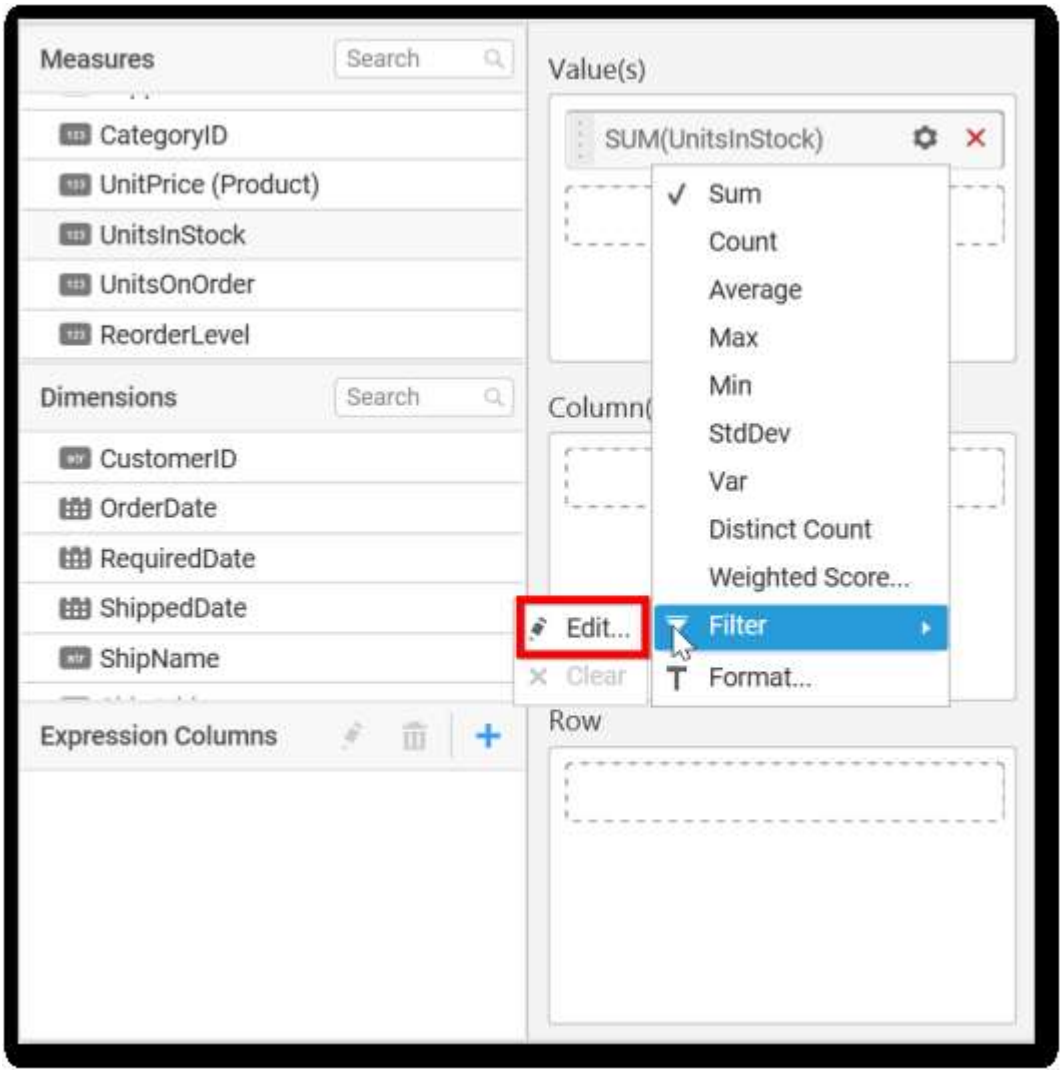
You can change the summary type of the value by clicking on Settings option.



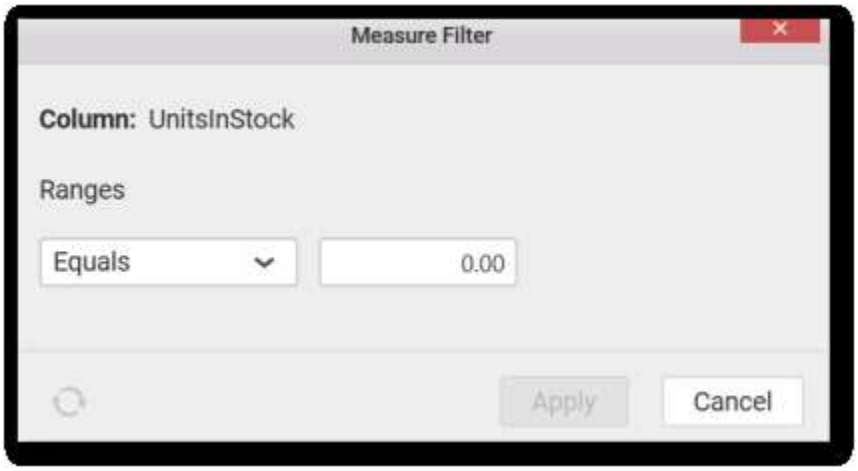
Select the required summary type from list.

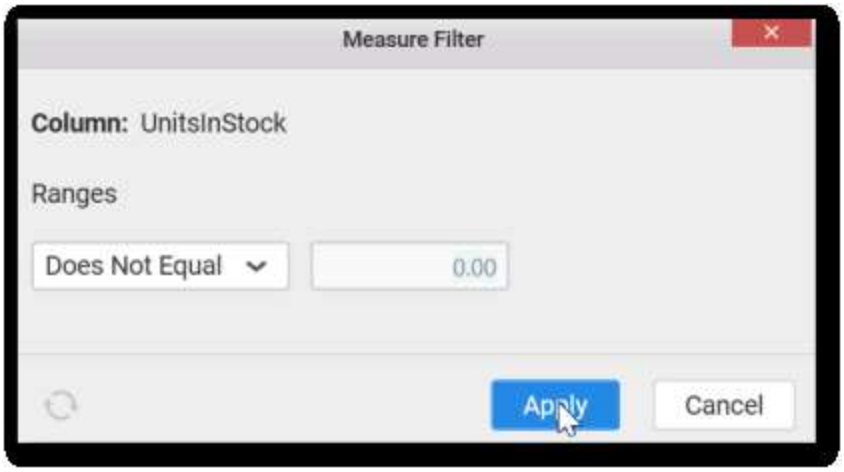


You can select what data to be displayed by choosing filter option.

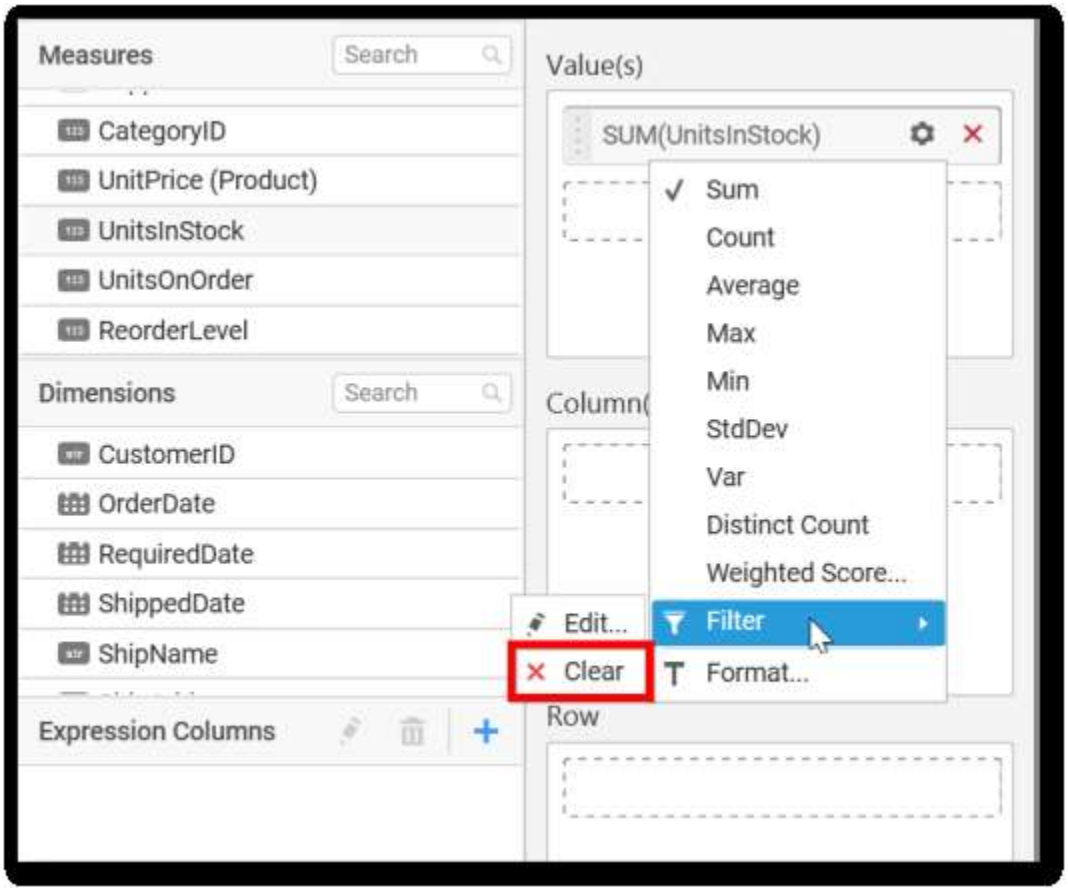


The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.

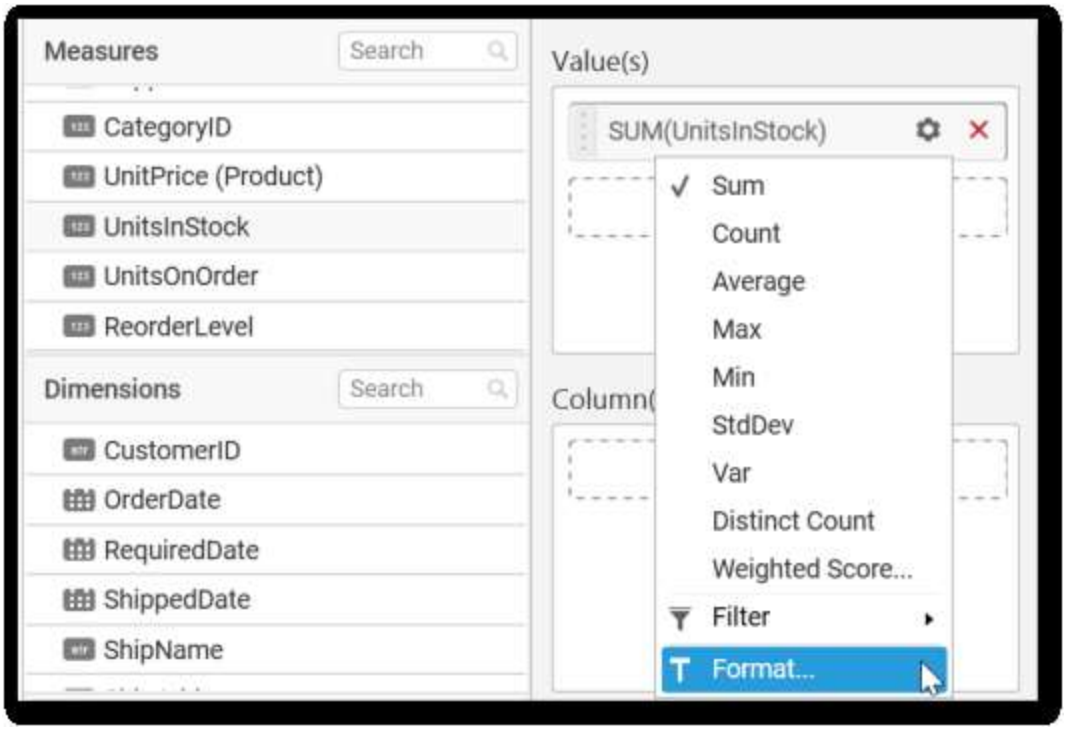




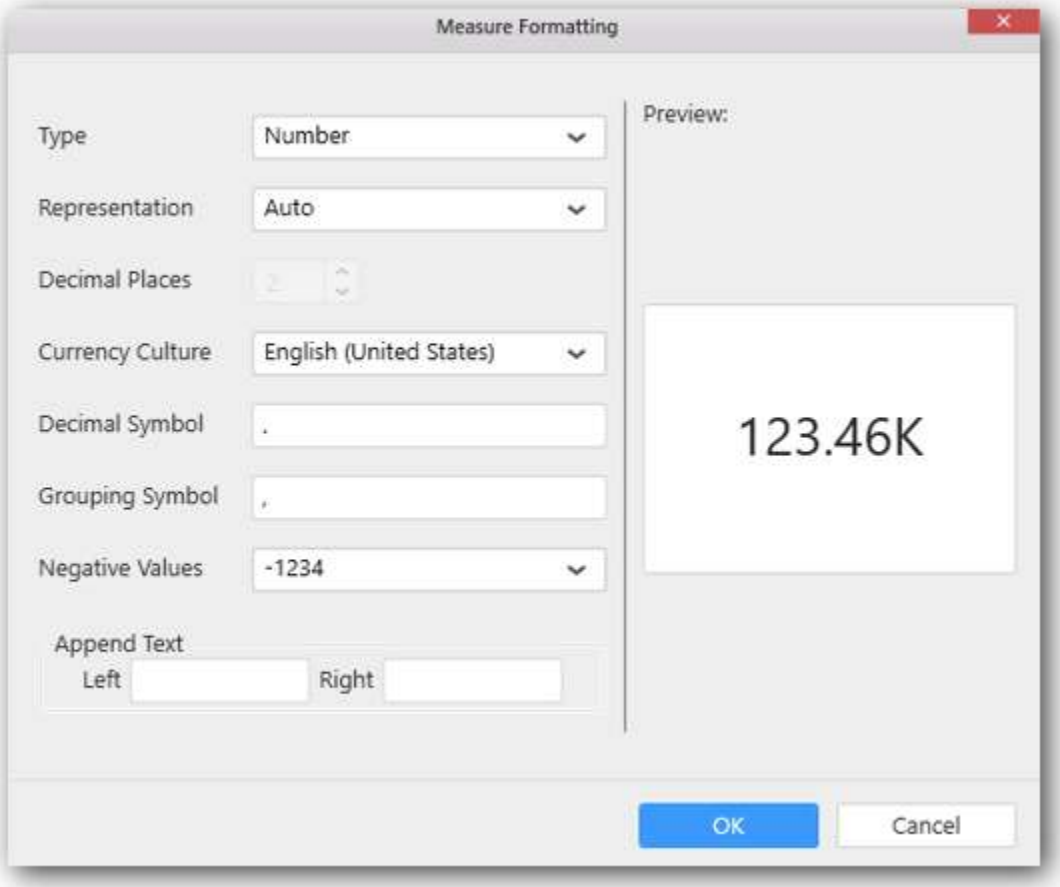
You can clear the filter.



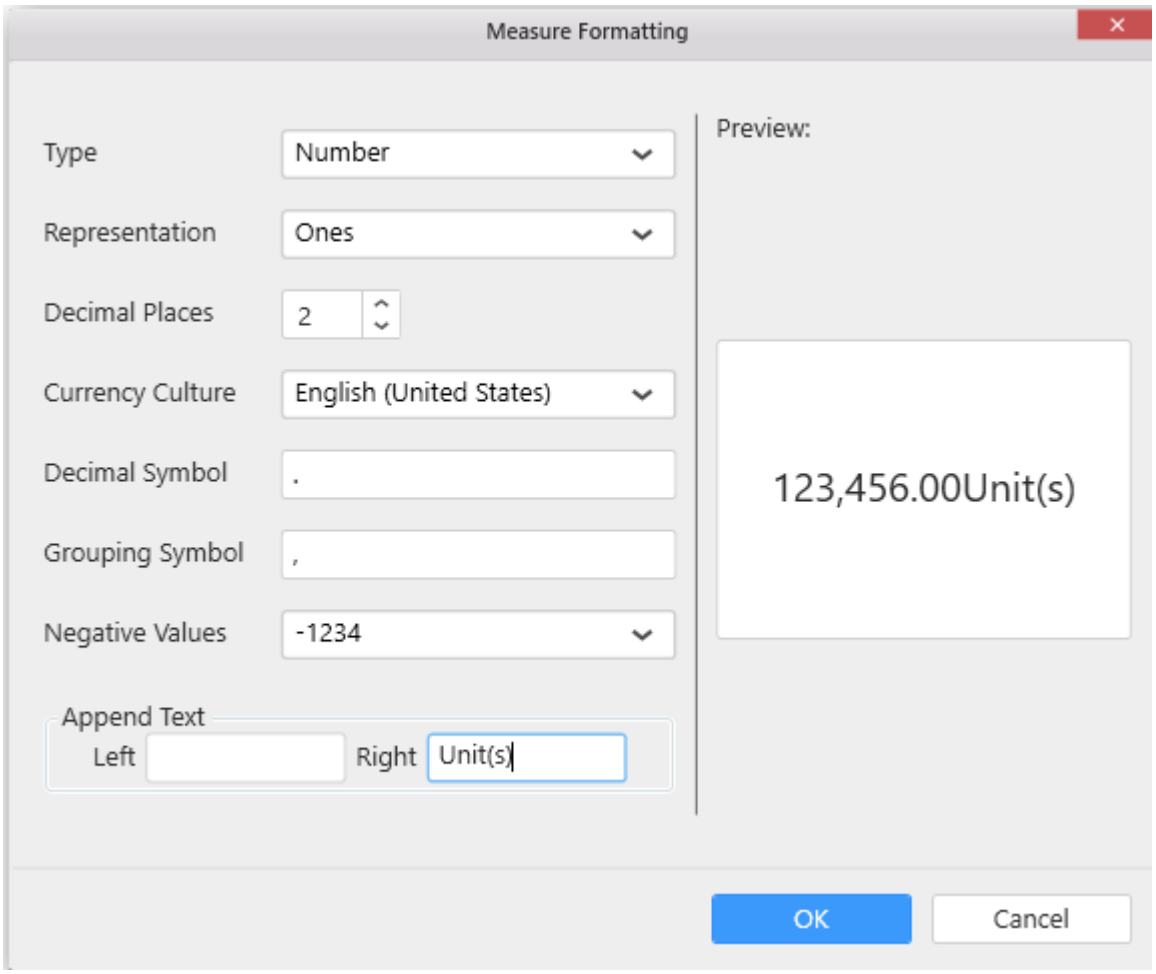
You can Format the value.



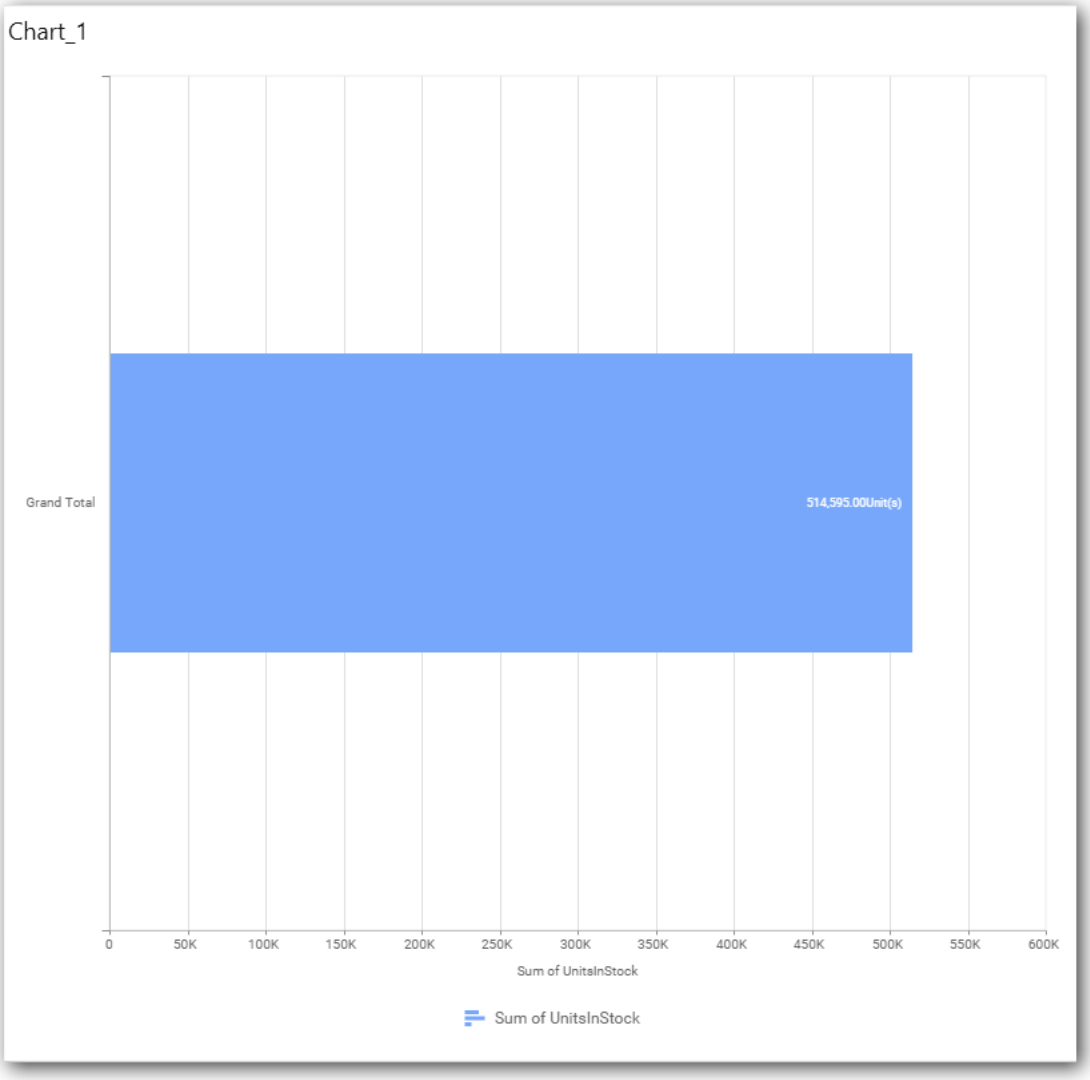
The format options will be shown.



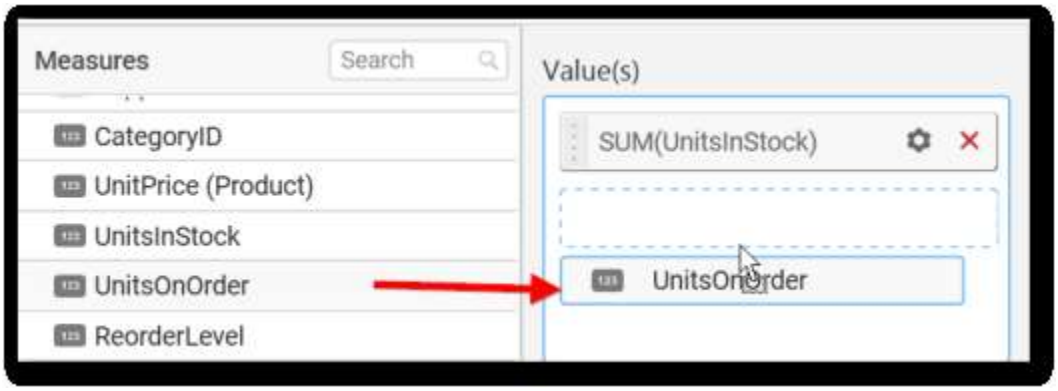
Choose the options you need and click OK.

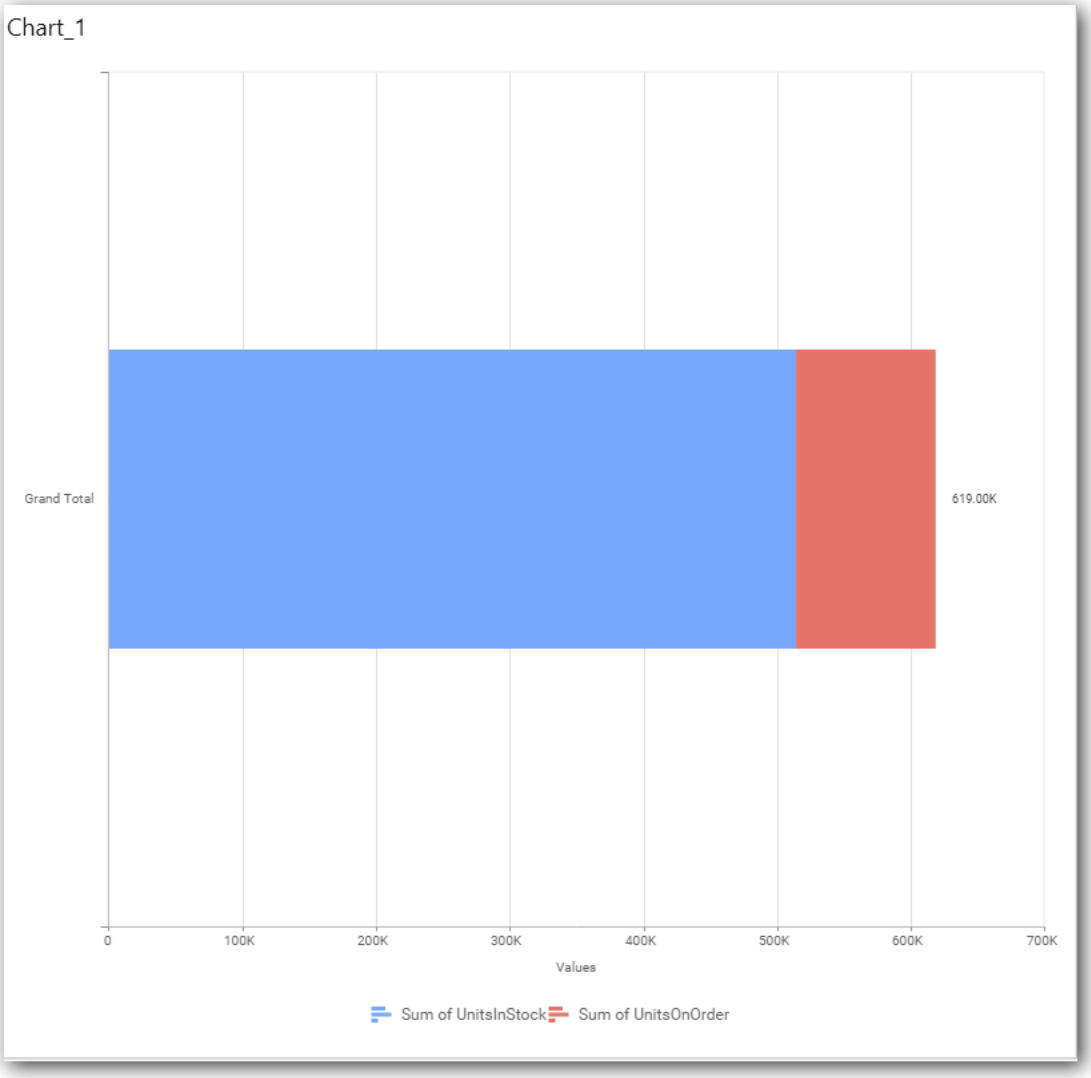


Now the Chart will be rendered like this.



You can add more number values by drag drop the Measures into Value field.

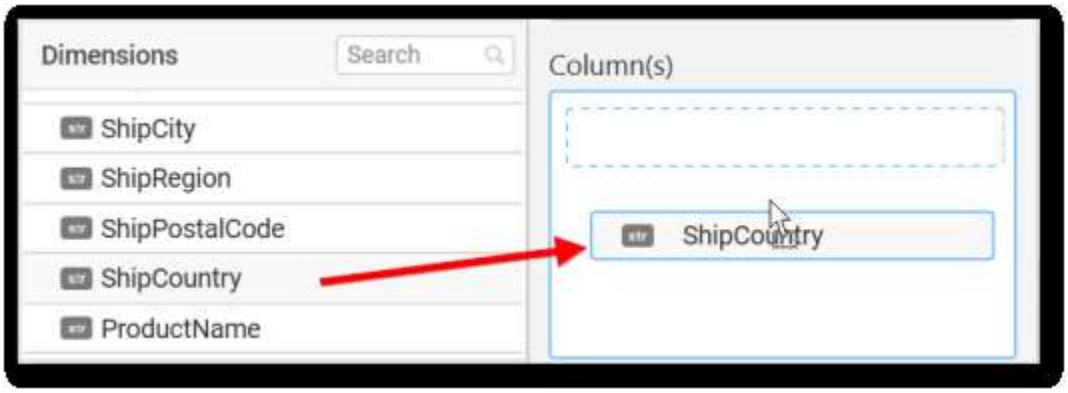


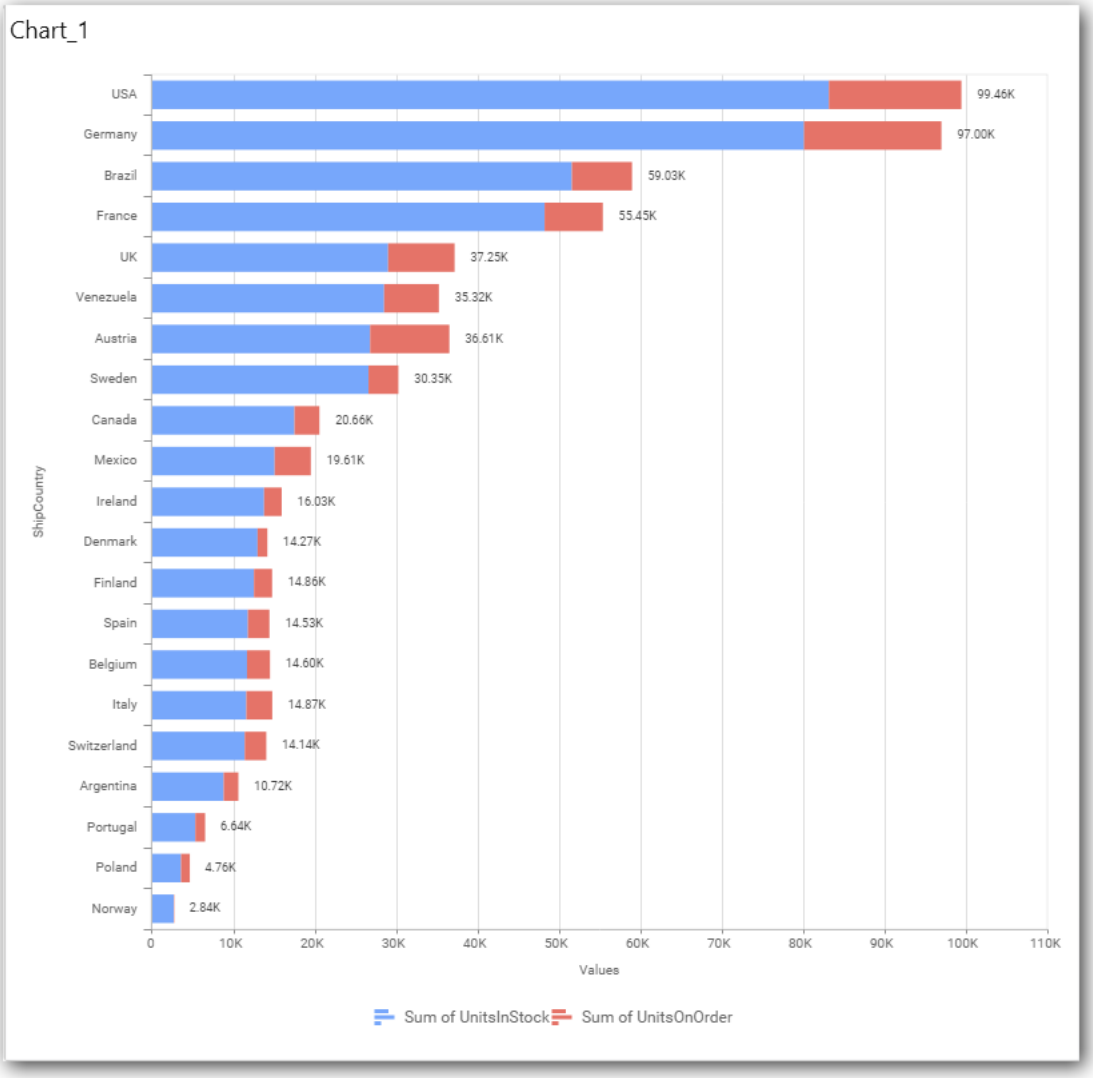


You can also add Dimensions and Columns to Value(s).

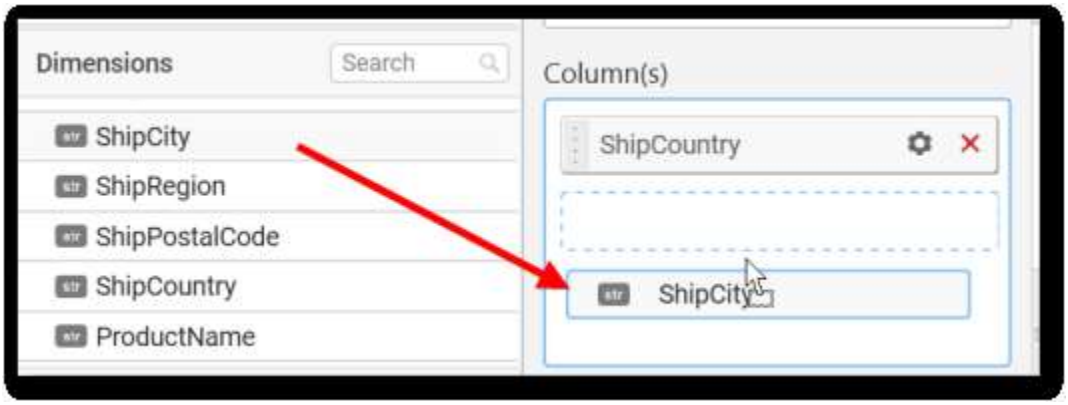
Assigning Column(s)

You can add the Dimension into Columns section by drag drop.

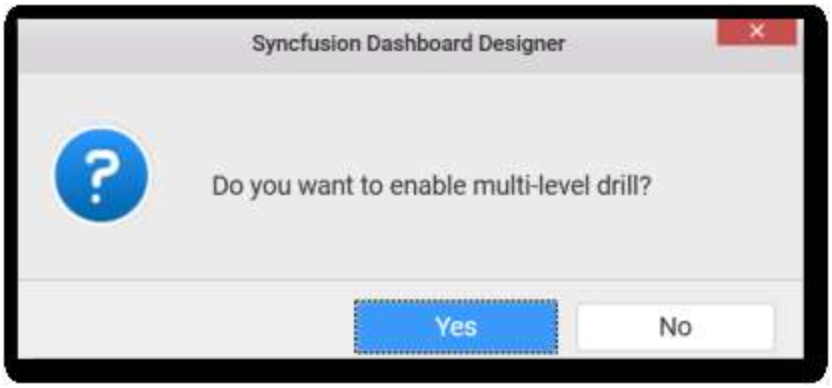




You have option to add more than one Column Value.

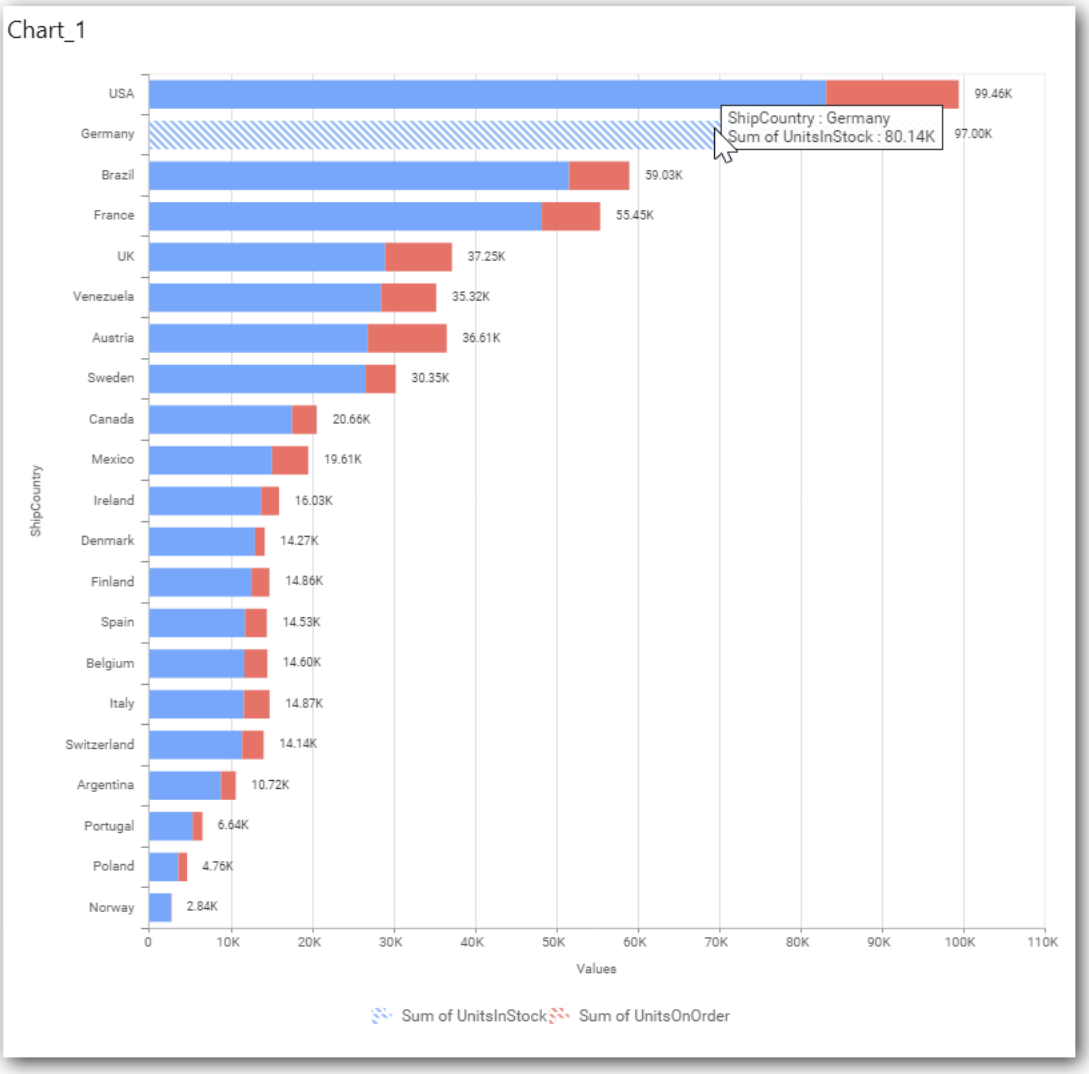


The following alert message will be shown.

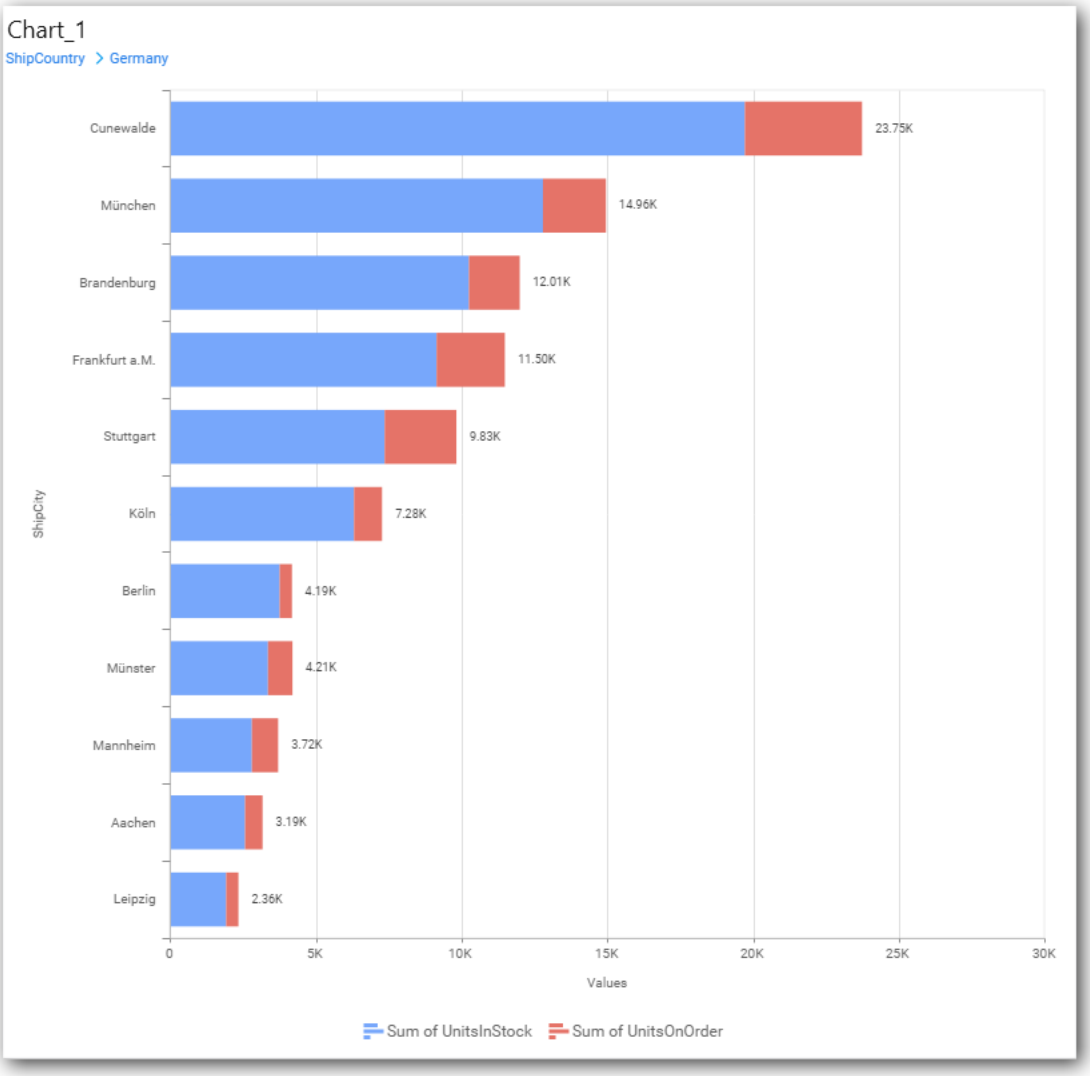


- If you choose Yes Drill down option will be enabled.

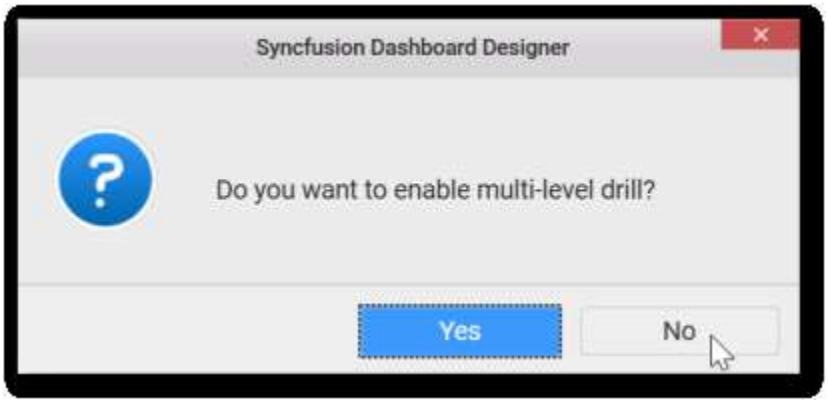
You can drill down the chart by clicking on the chart.

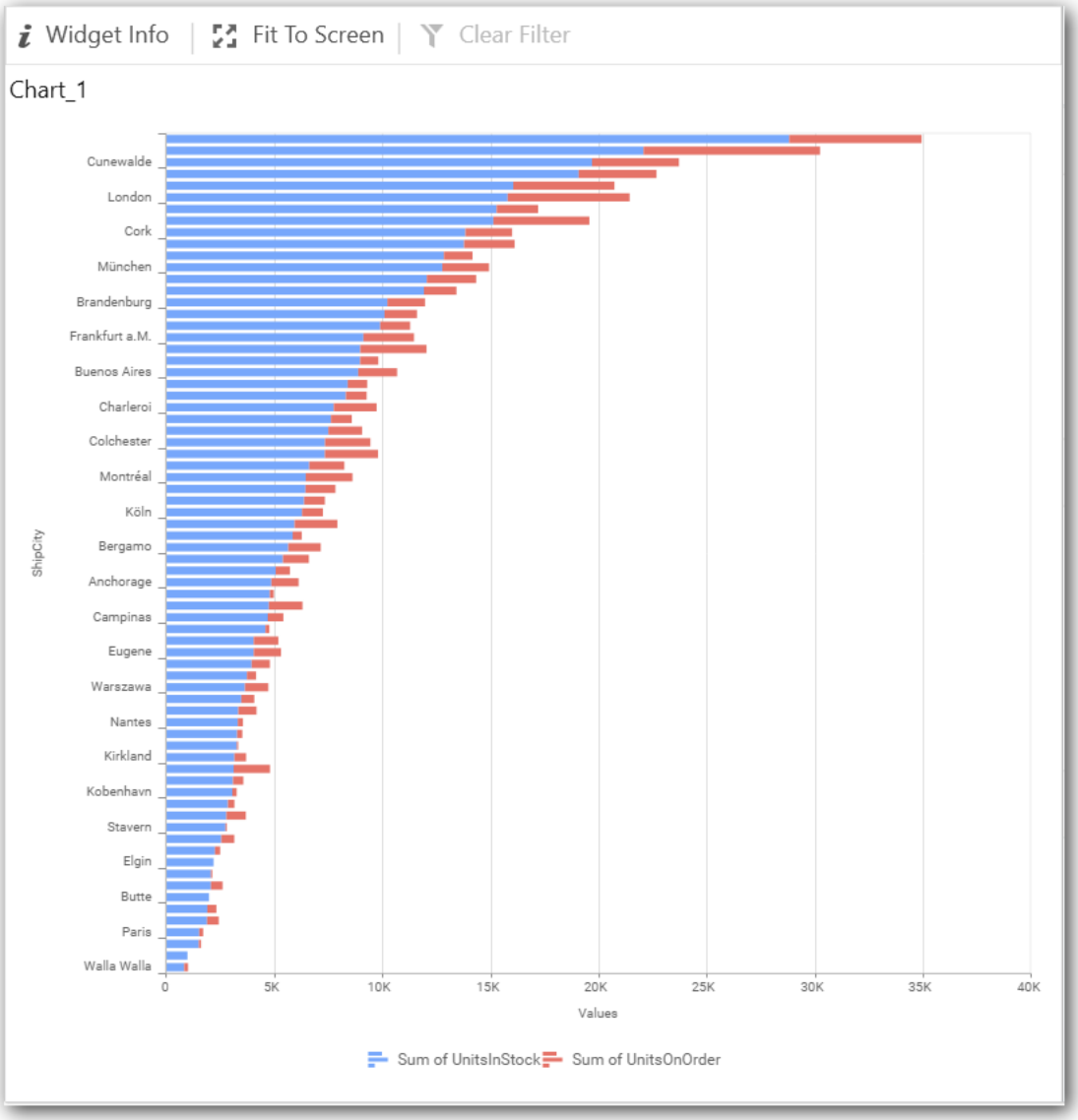


The drilled view of the chart is follows.



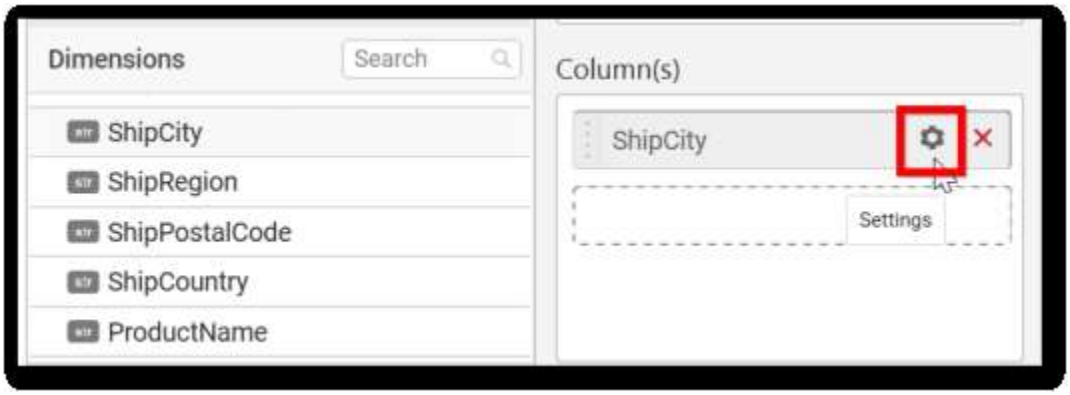
- If you click **No** the new Dimension value will replace old value.

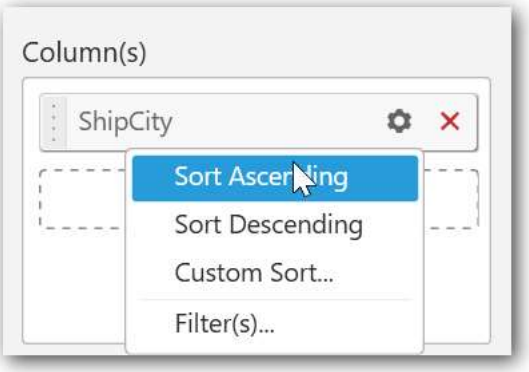




You can also add Measures and Expression columns into Column(s) field.

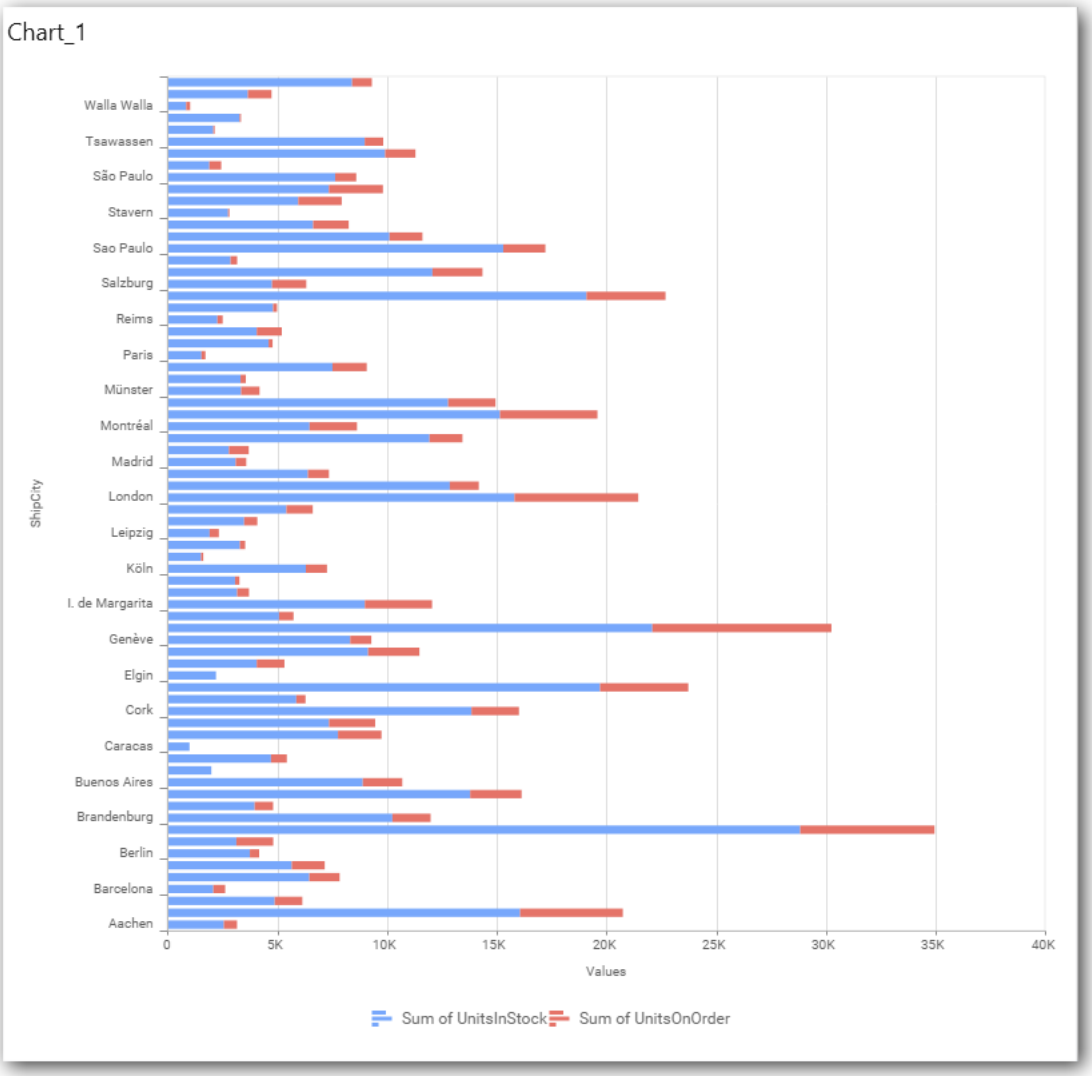
You have options to change the settings.



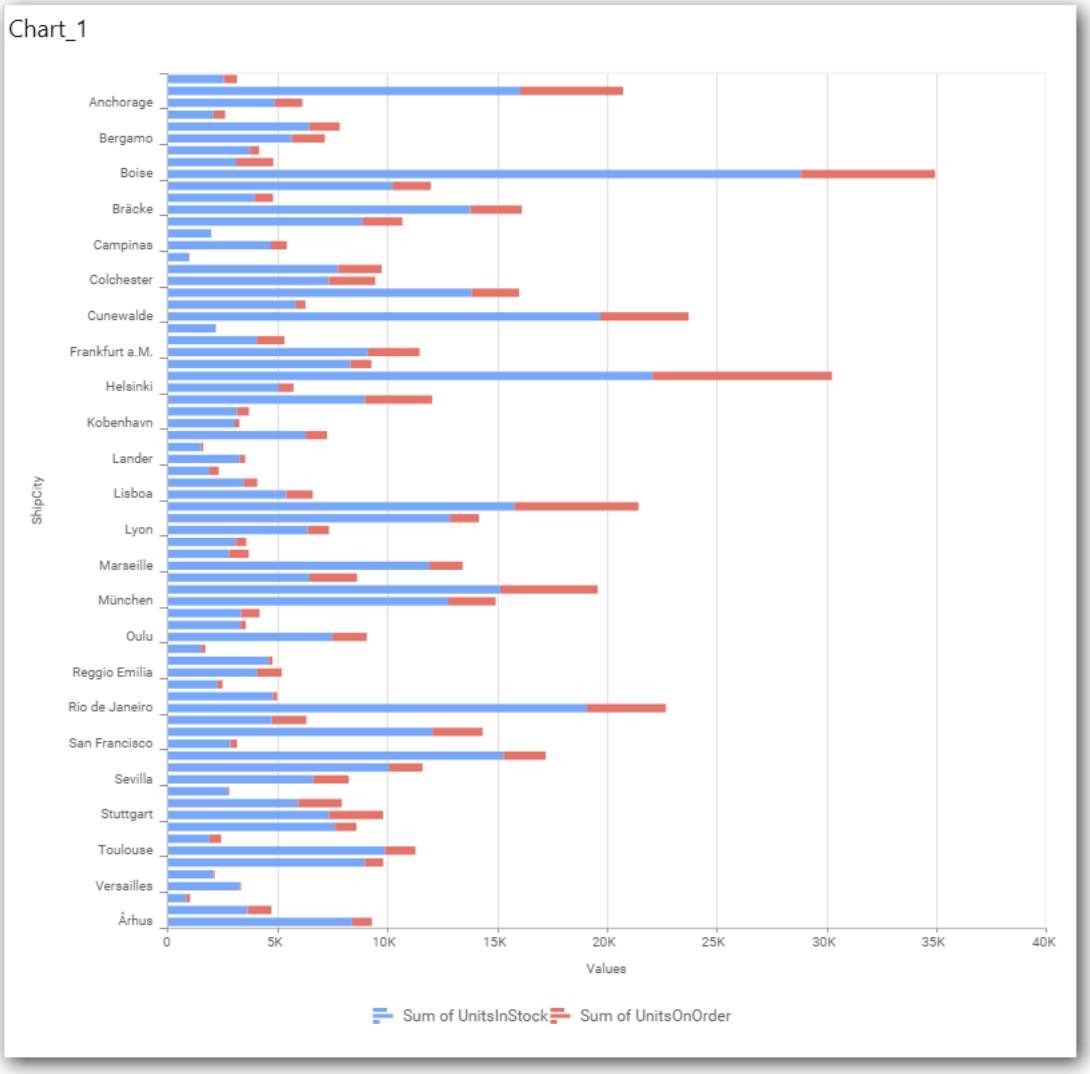


You can sort the chart either in Ascending or Descending series.

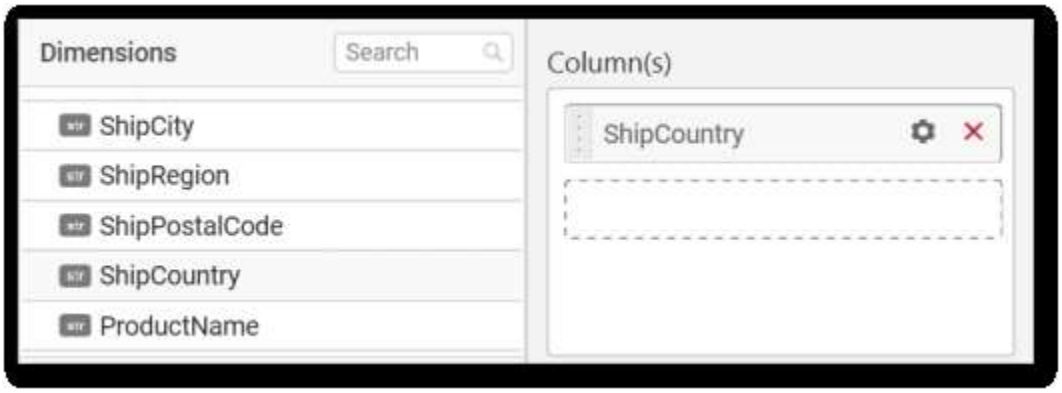
Ascending Order



Descending order

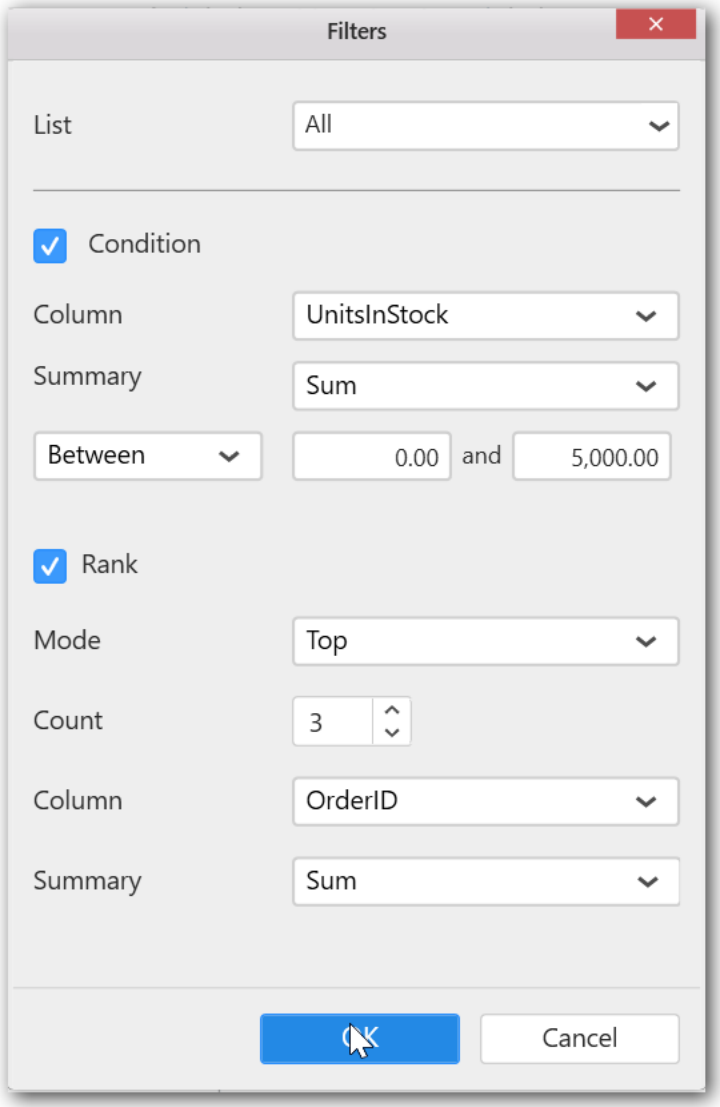


You can apply a filter.

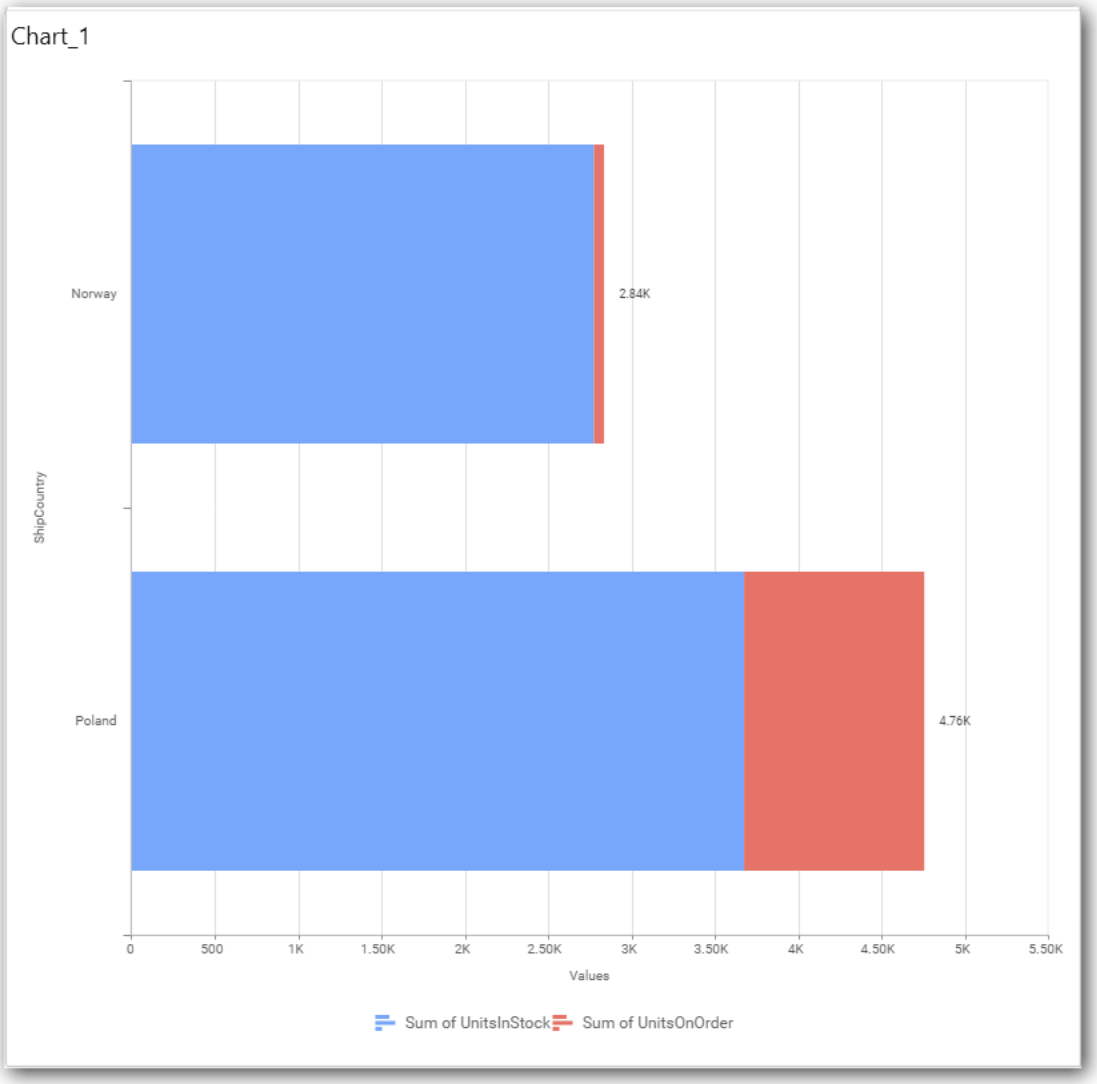


!Chart illustration[](images/stackedbarchart_img32.png)

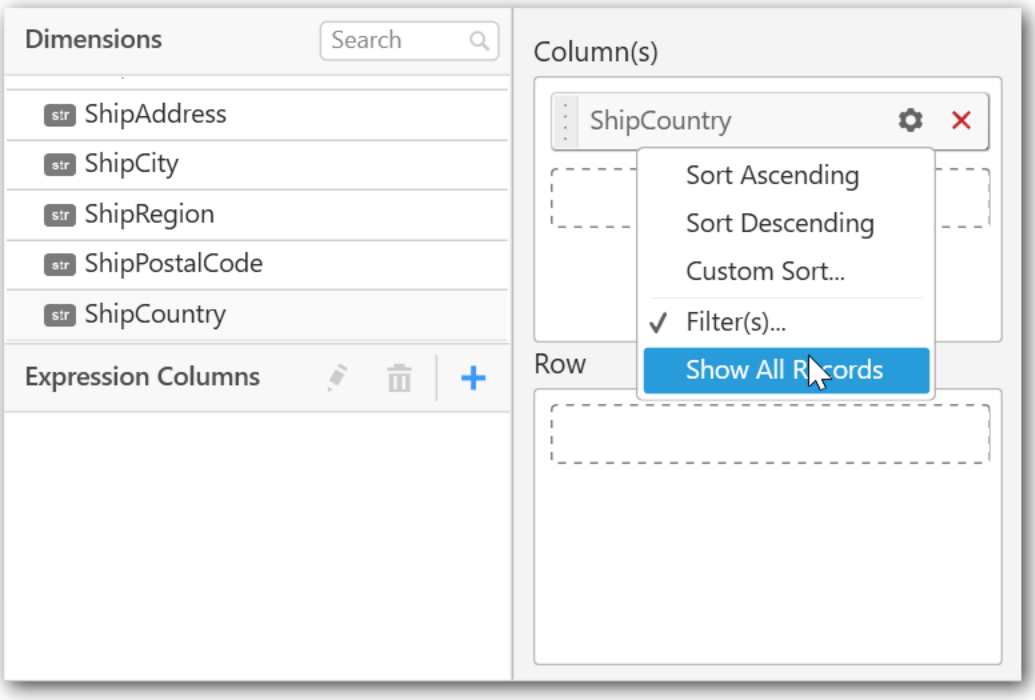
Select the Conditions and Rank you need.



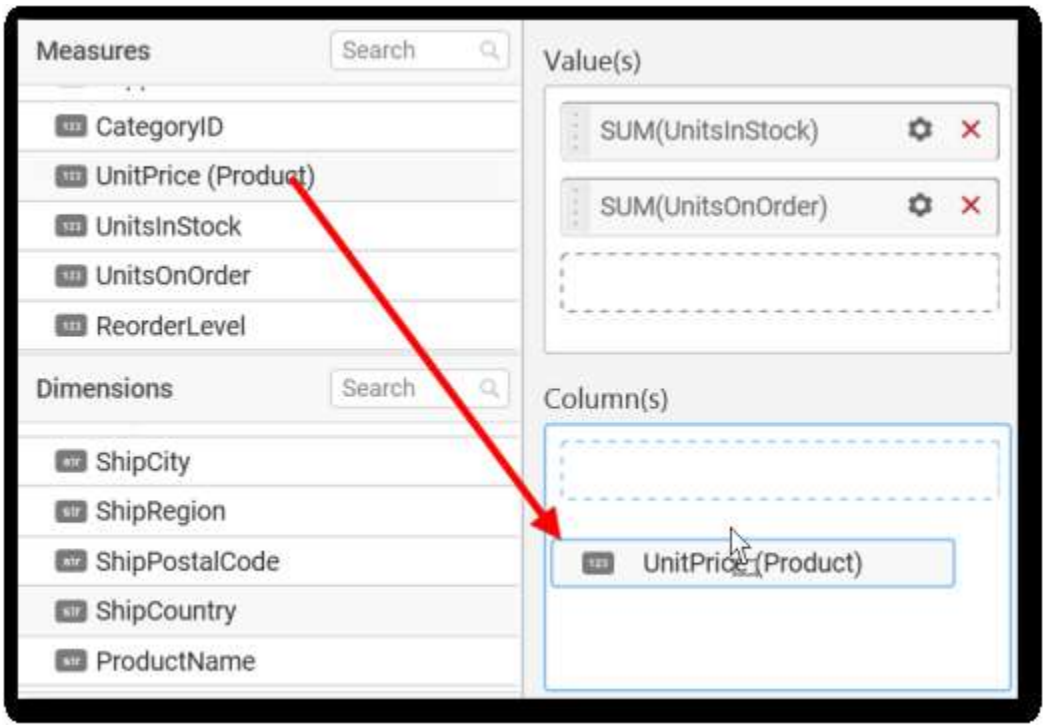
Now the chart will be rendered like this.

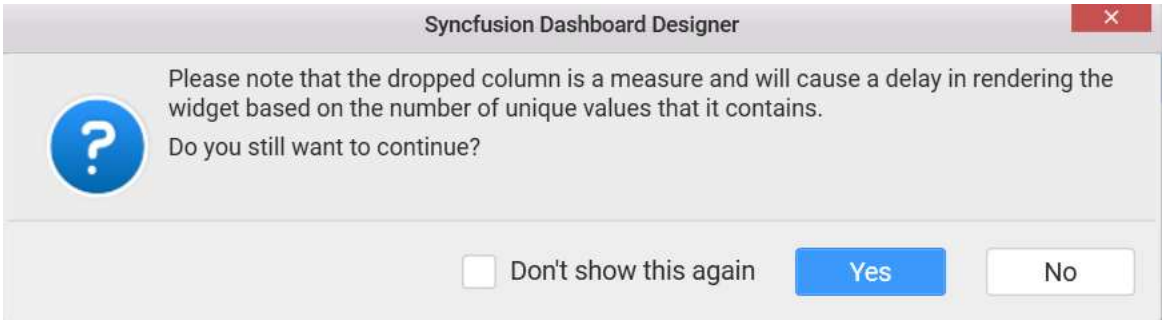


To show all records again click on Show All Records.



On adding Measures into Column(s) will show the following alert.

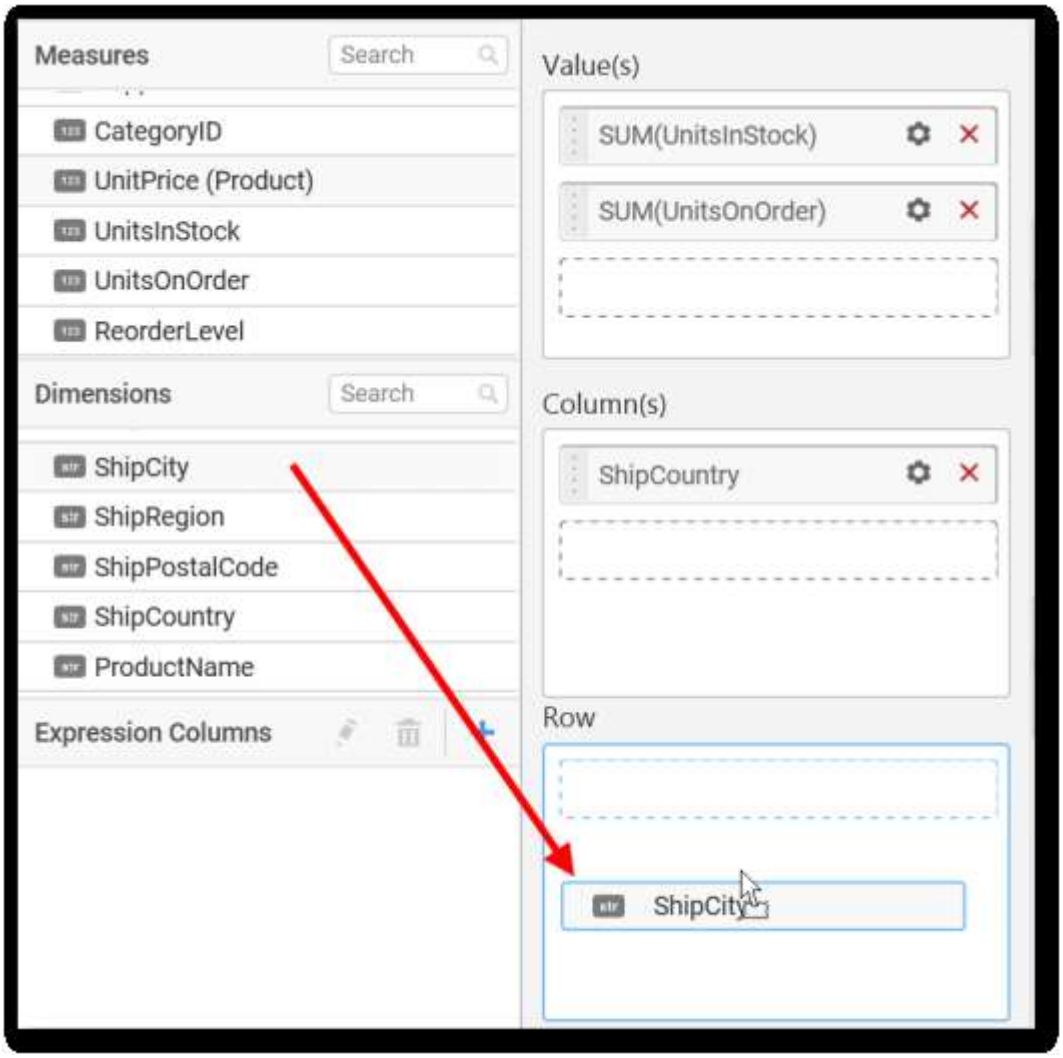




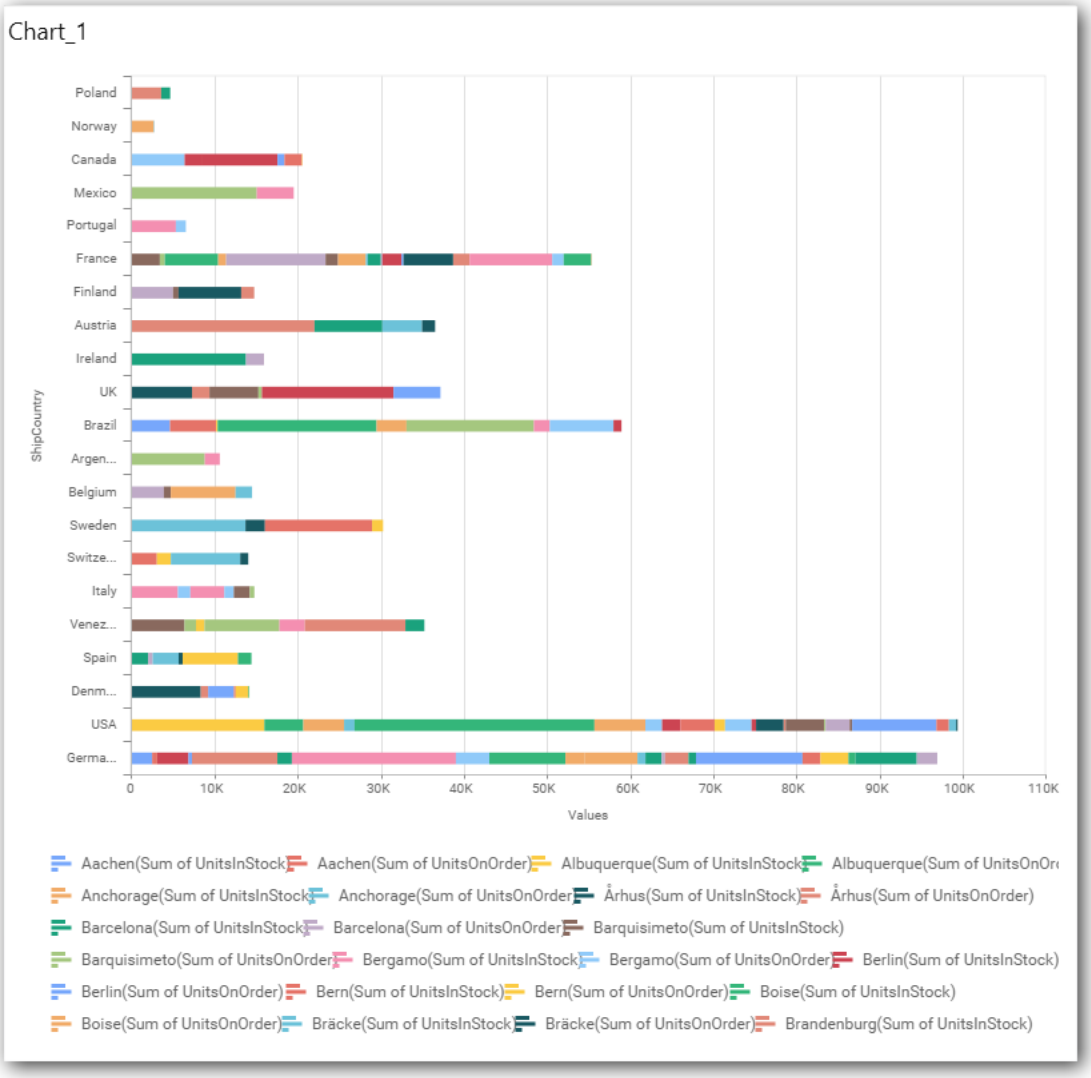
To continue select Yes, otherwise select No.

Assigning Row

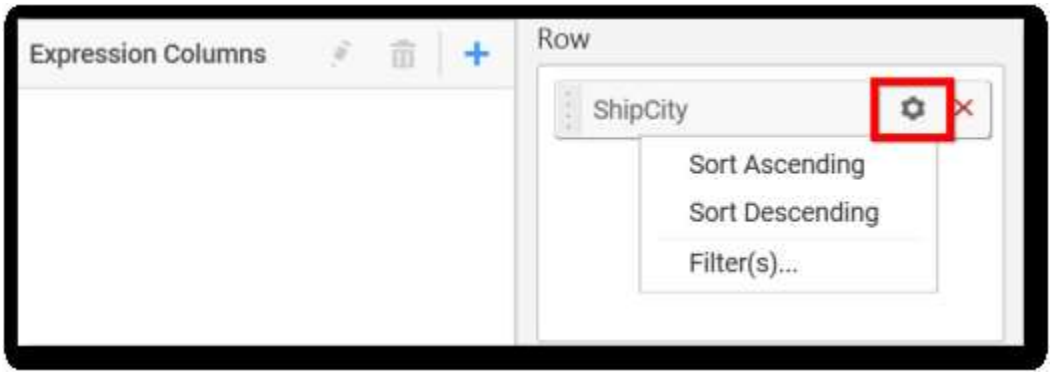
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image



You have settings options similar to Column(s).



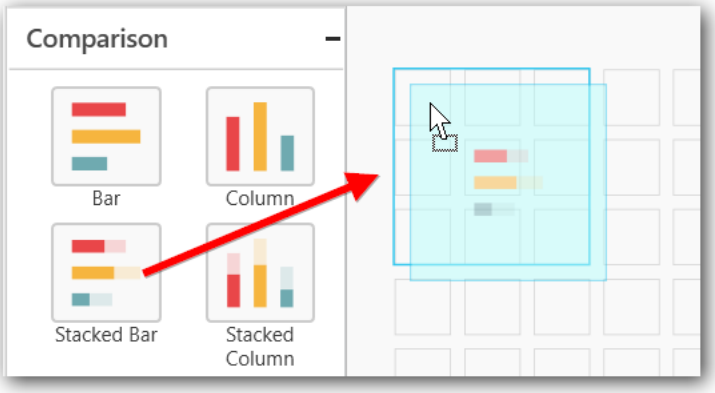
How to configure SSAS data to stacked bar Chart?

Stacked Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension

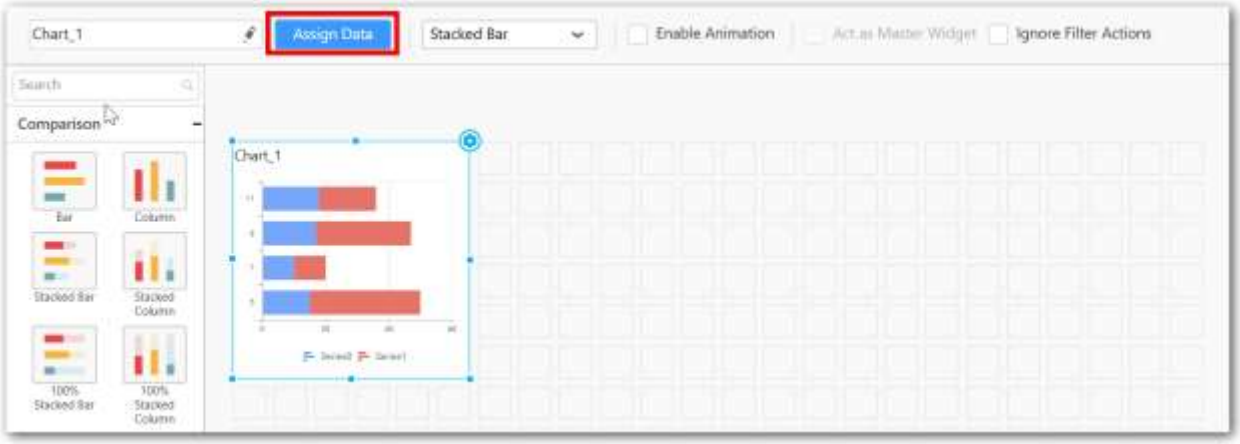
that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuration of SSAS data to Stacked Bar Chart

Drag and drop the Stacked Bar chart widget into canvas and resize into your required size.

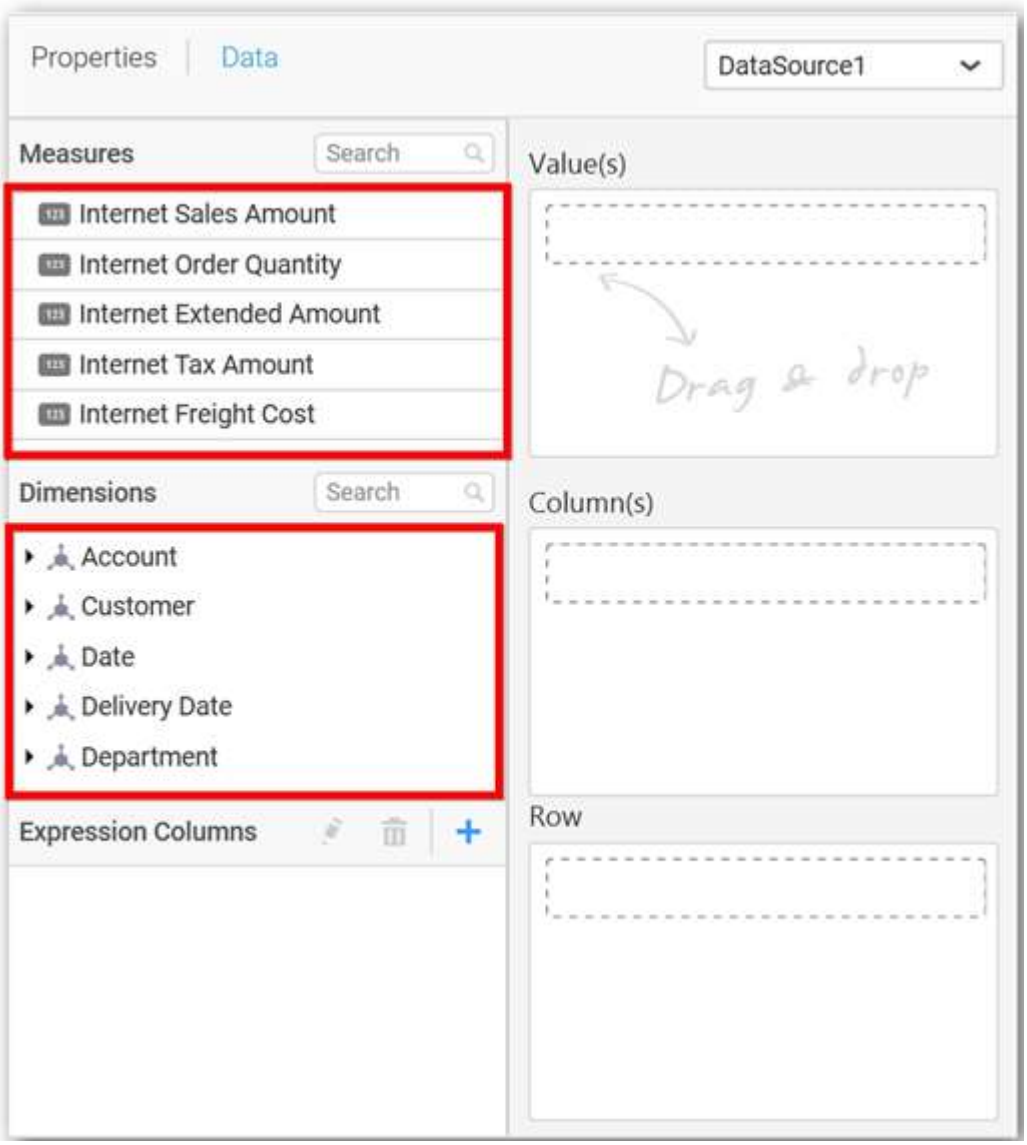


Select the dropped widget using mouse.



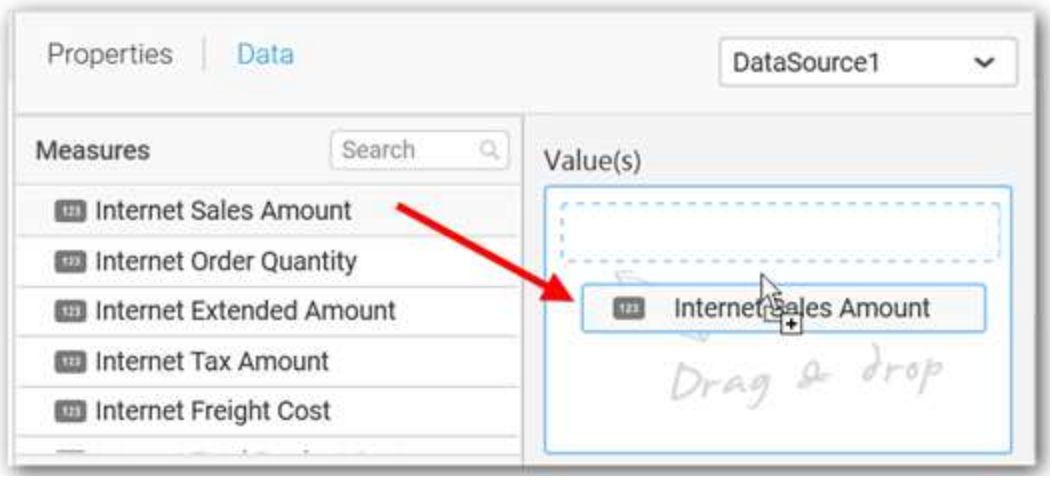
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions.

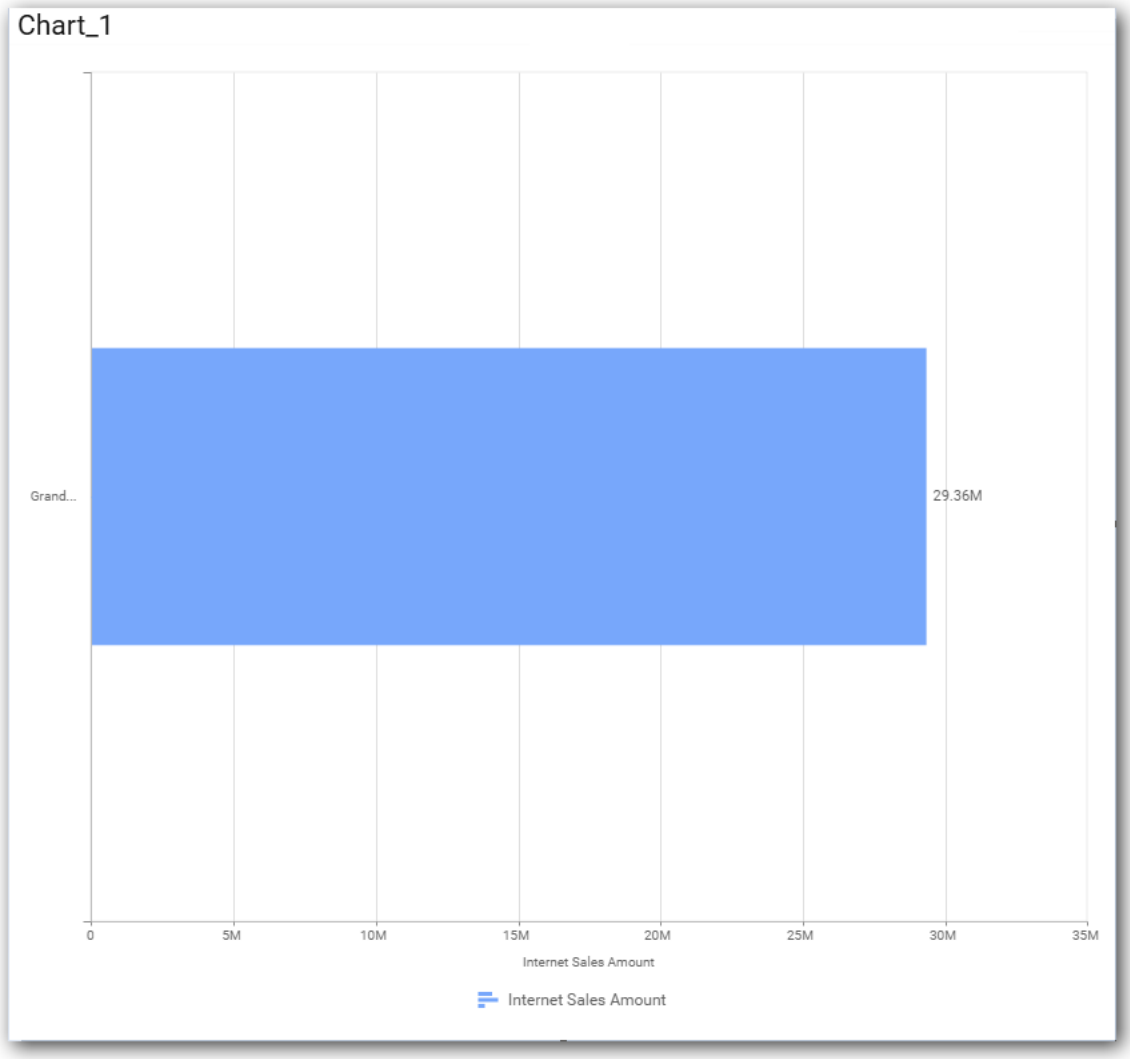


Assigning Value(s)

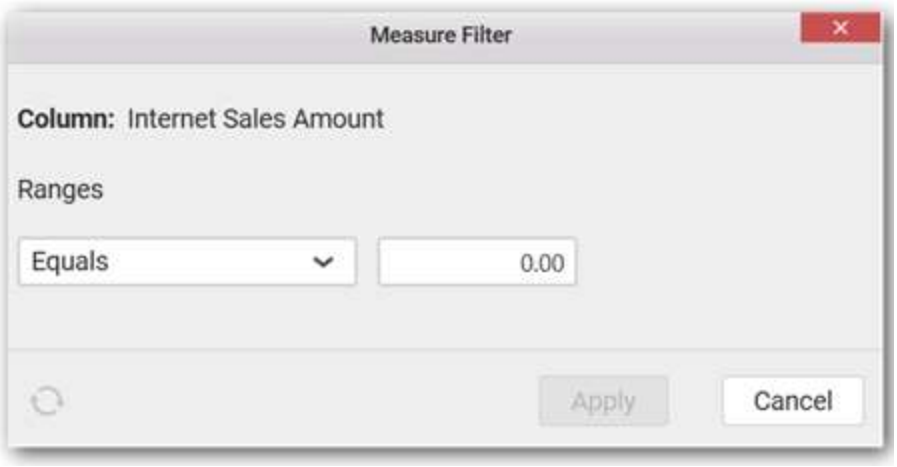
Drag and drop a column under Measures category into Value(s) section.



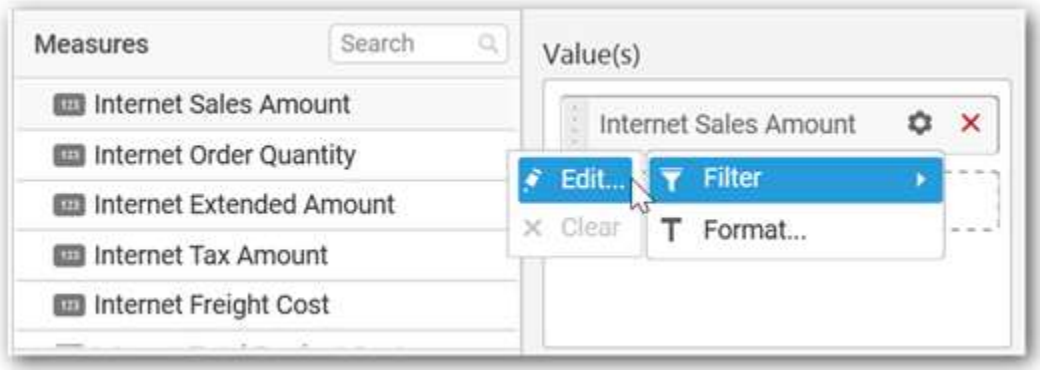
Now the chart will be rendered like this.



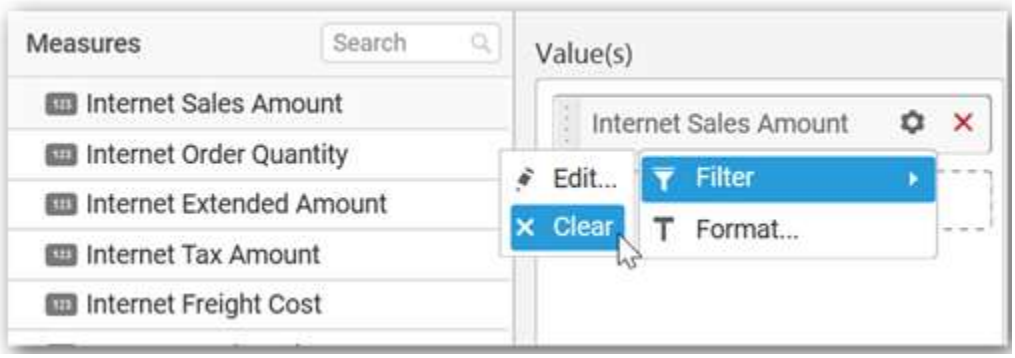
Define the filter criteria to match through choosing Edit option in Filter menu item.



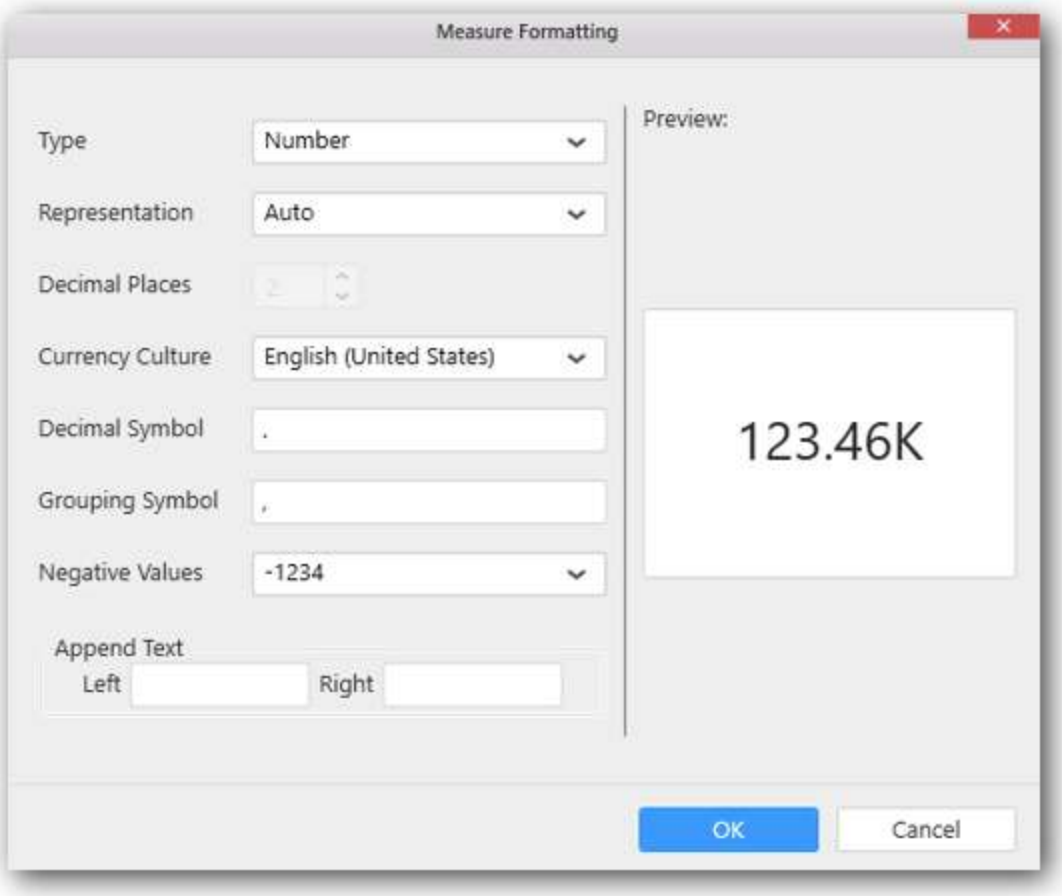
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



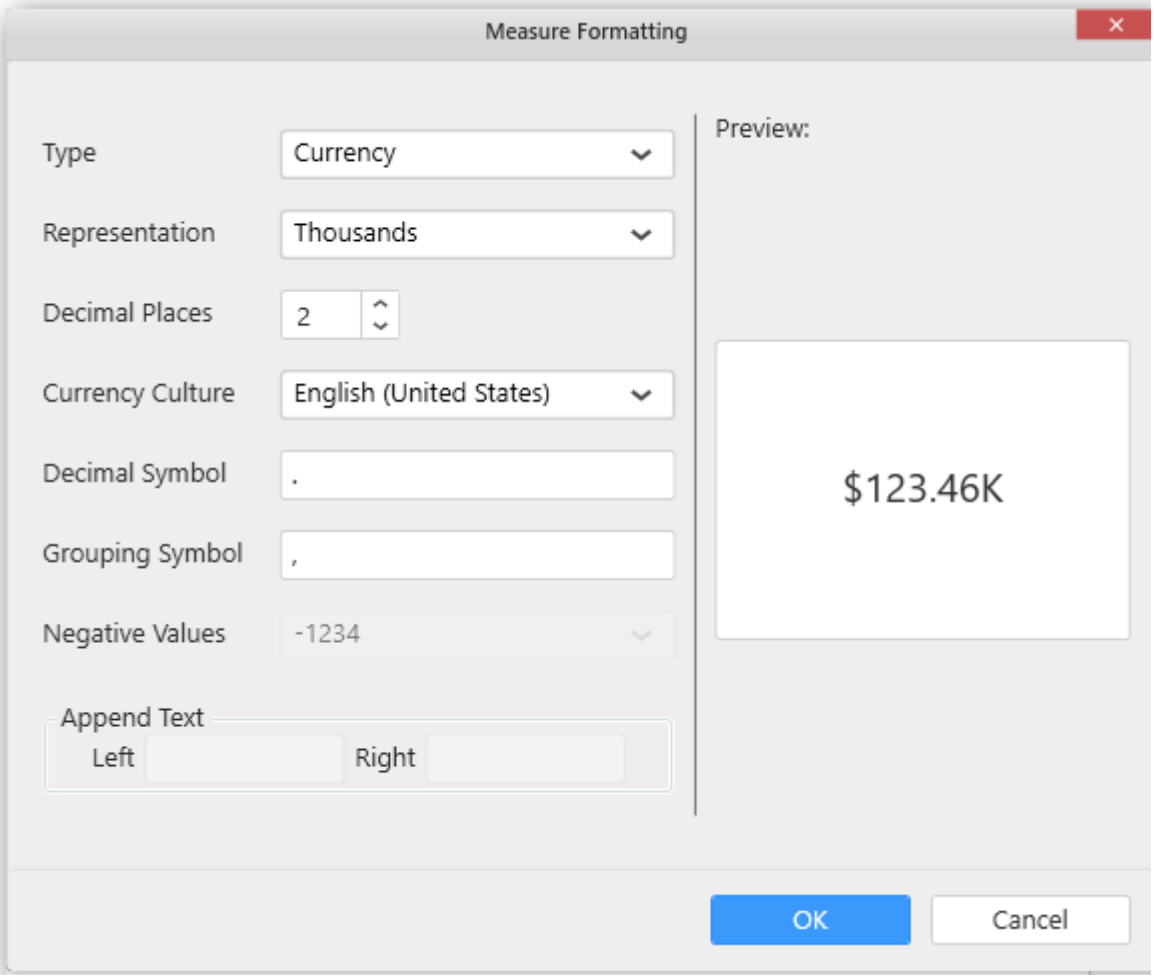
Select Clear option to clear the defined filter.



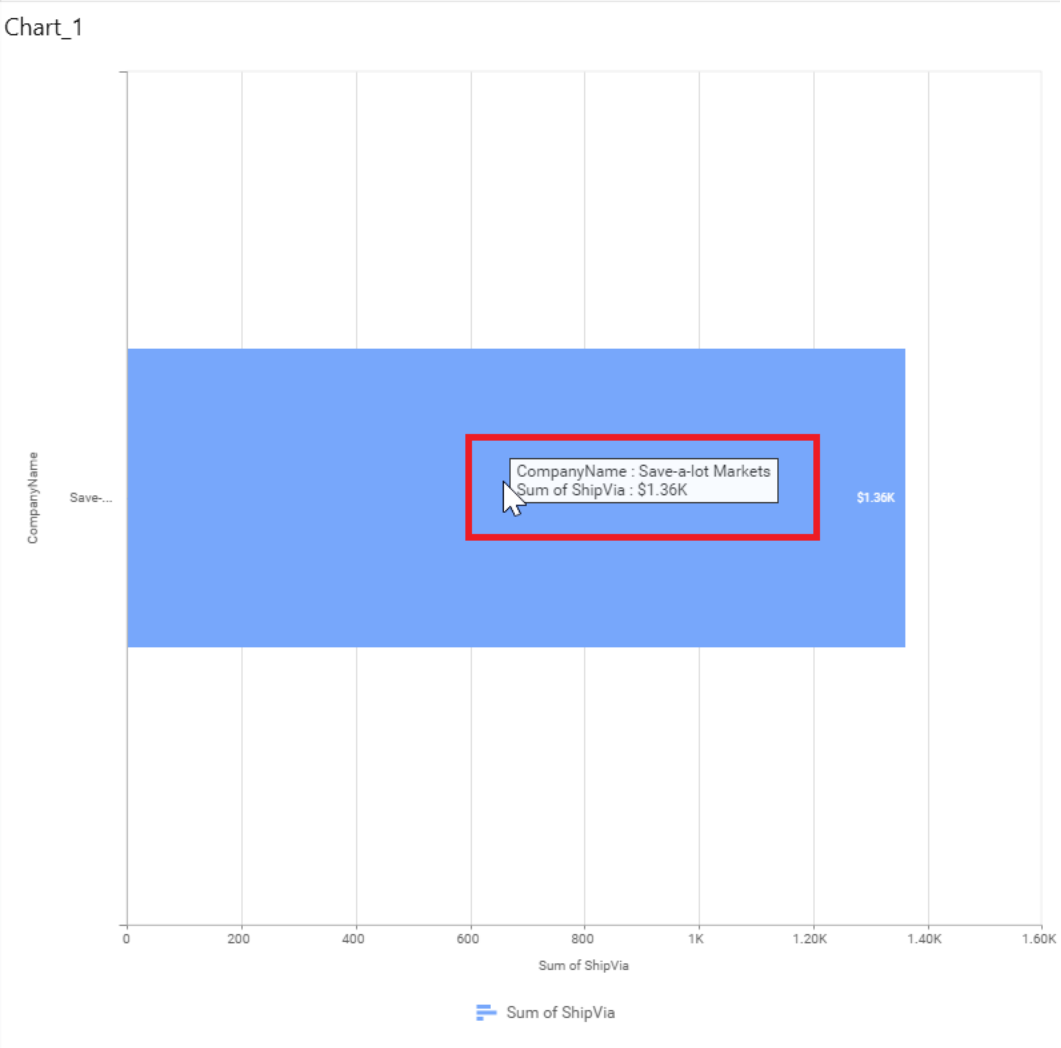
Select Format option to define the display format to the values in the column through Measure Formatting window.



Choose the options you need and click OK

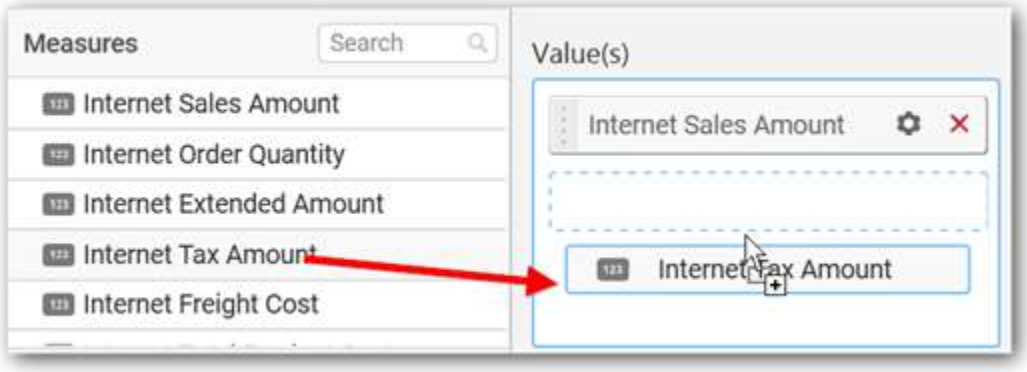


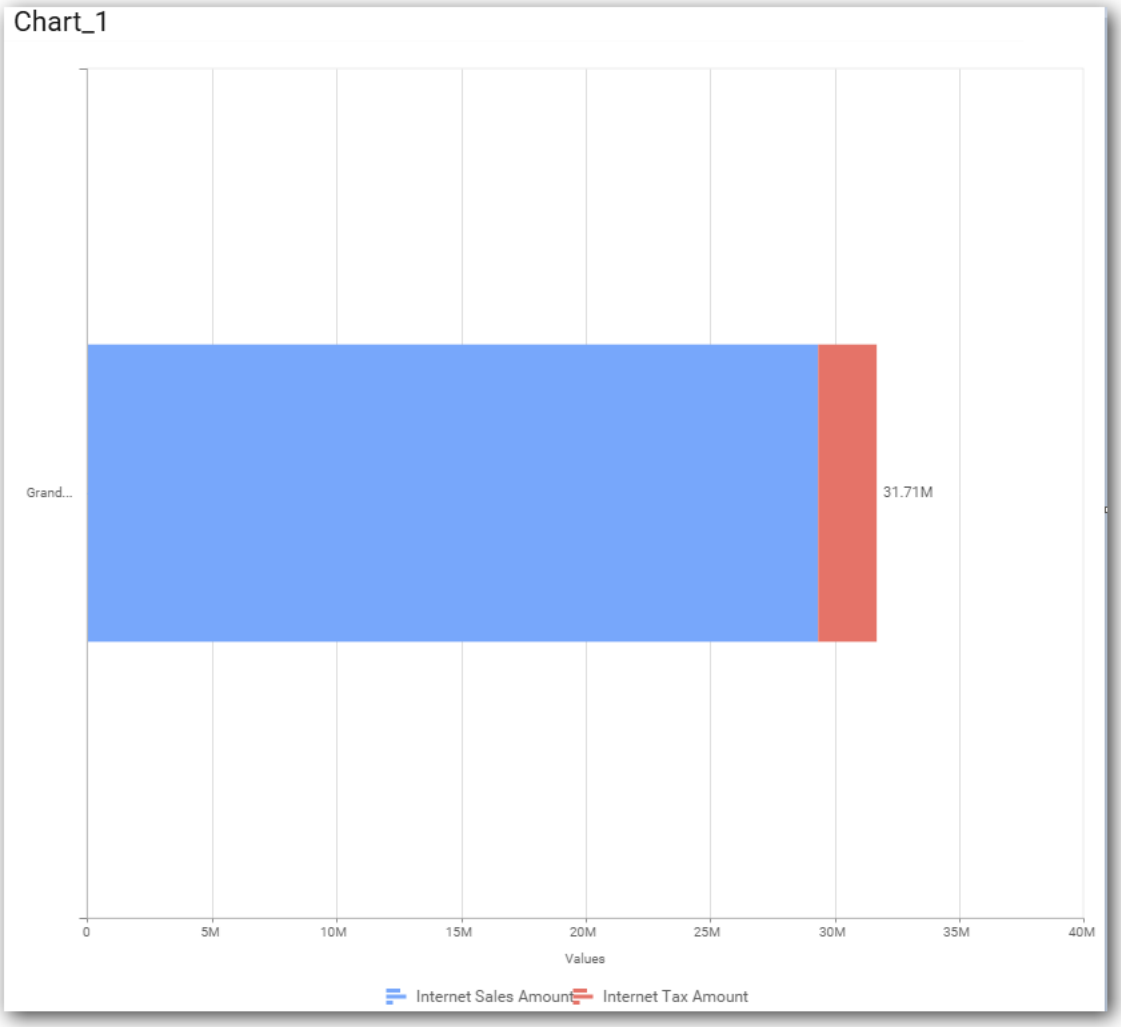
Now the Chart will be rendered like this.



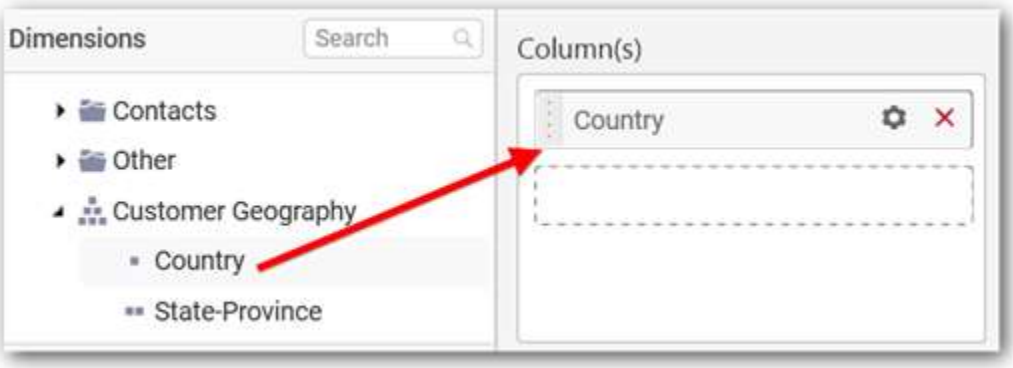
Assigning Column(s)

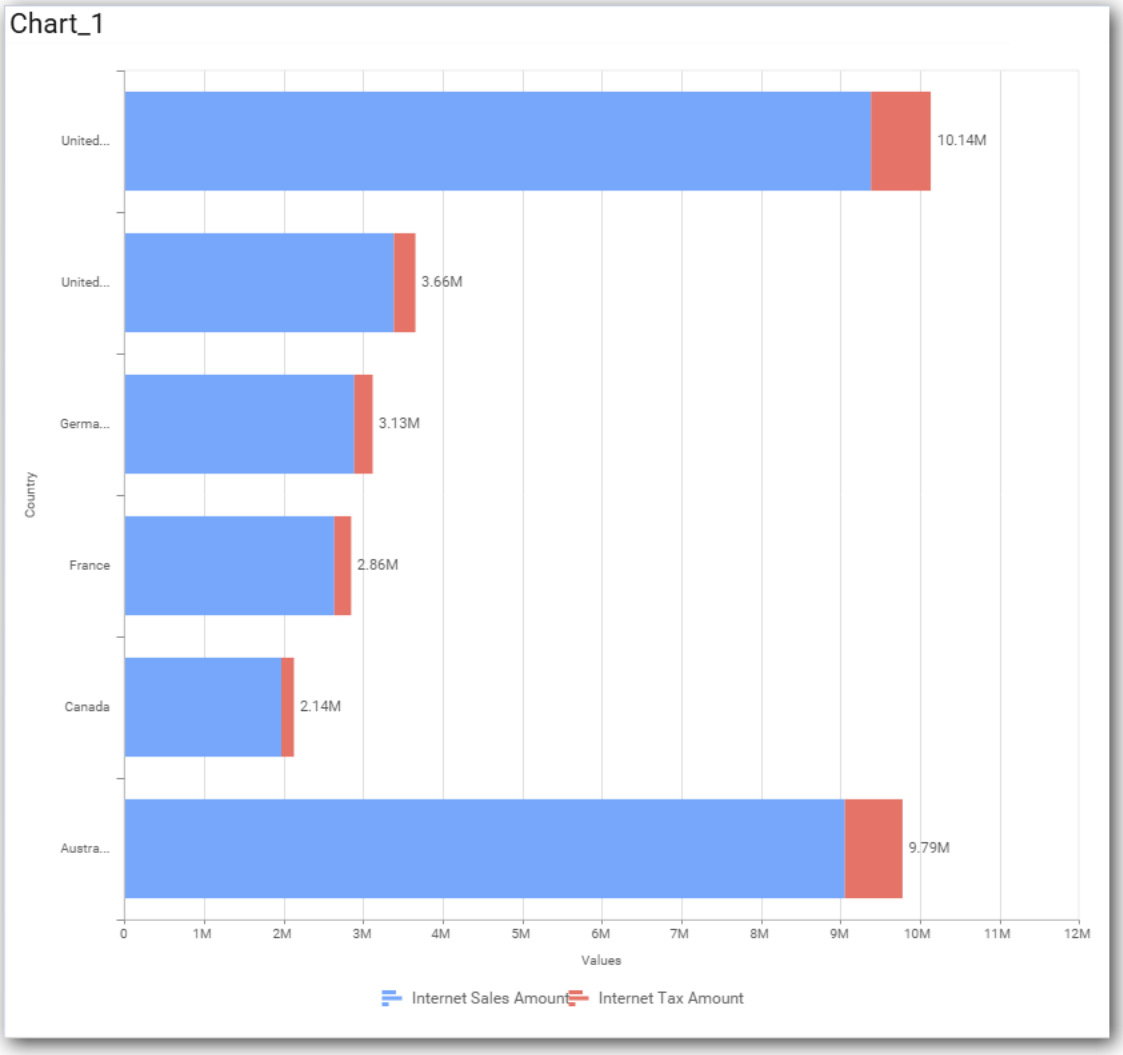
You can also add more than one column to the Value(s) section.



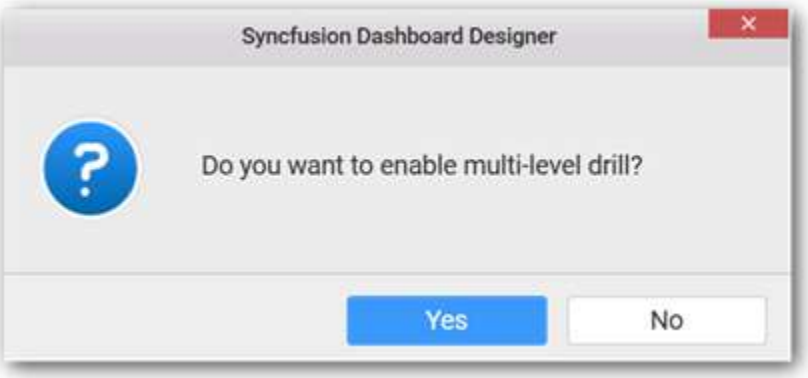


Add a dimension level or hierarchy into Column(s) section through drag and drop.



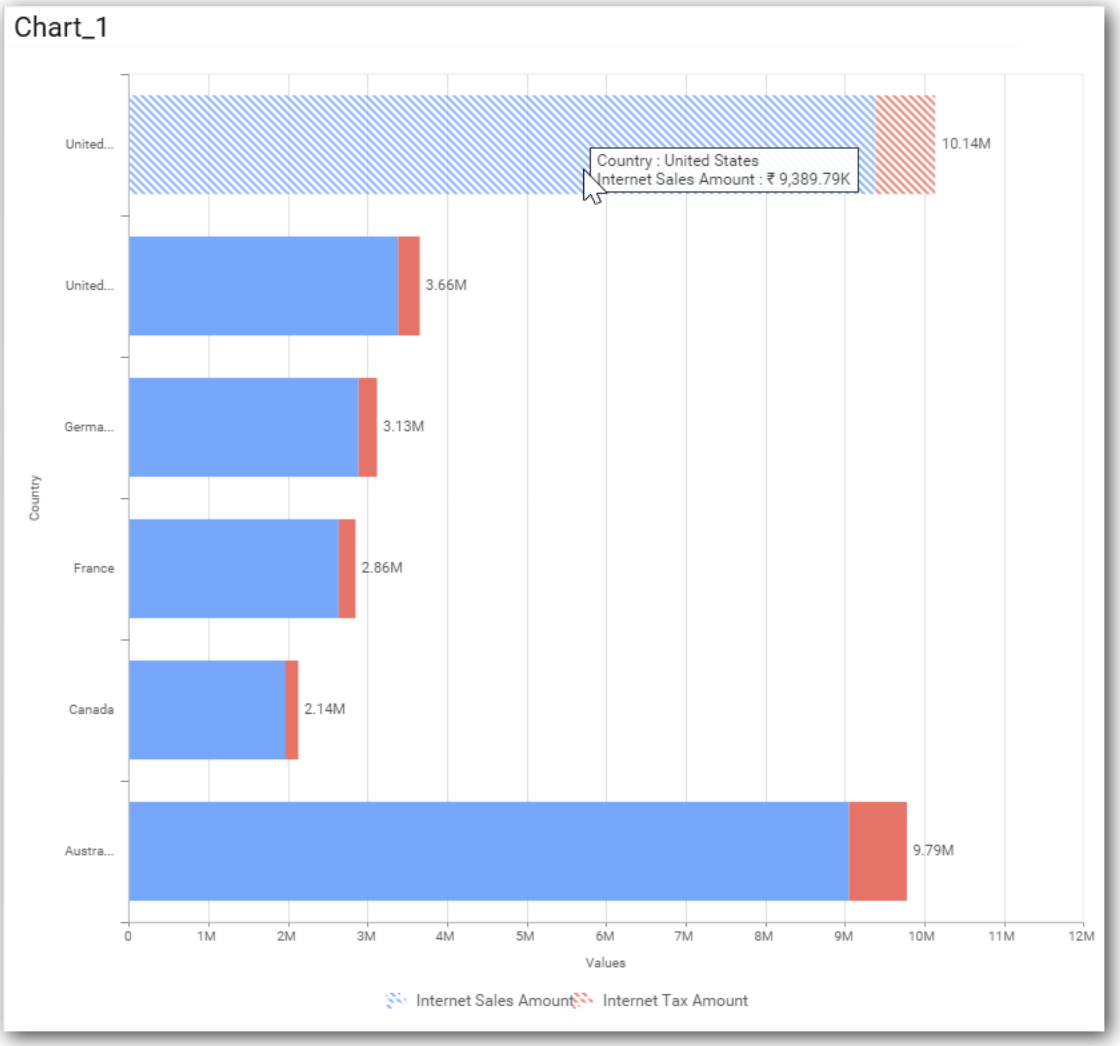


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

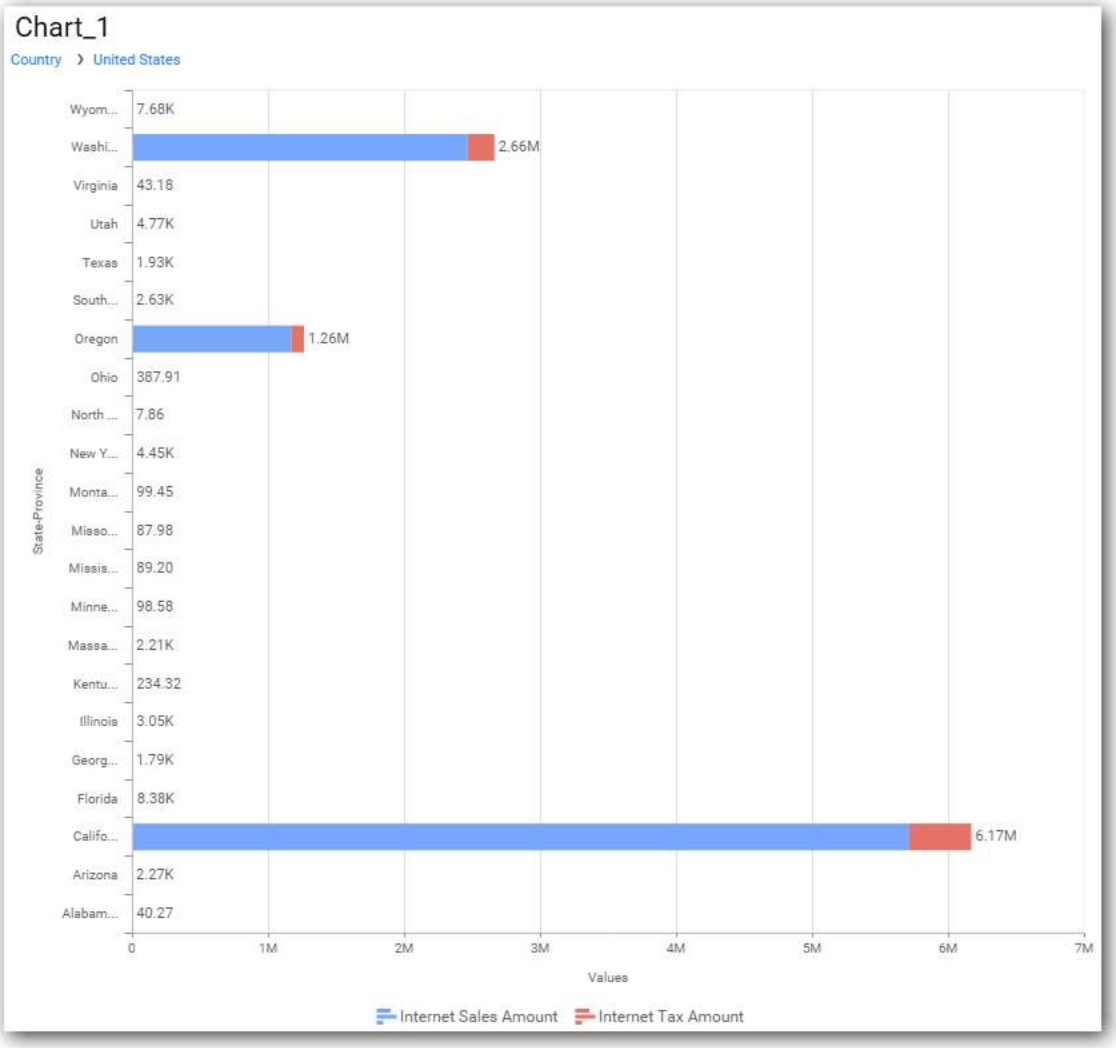


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

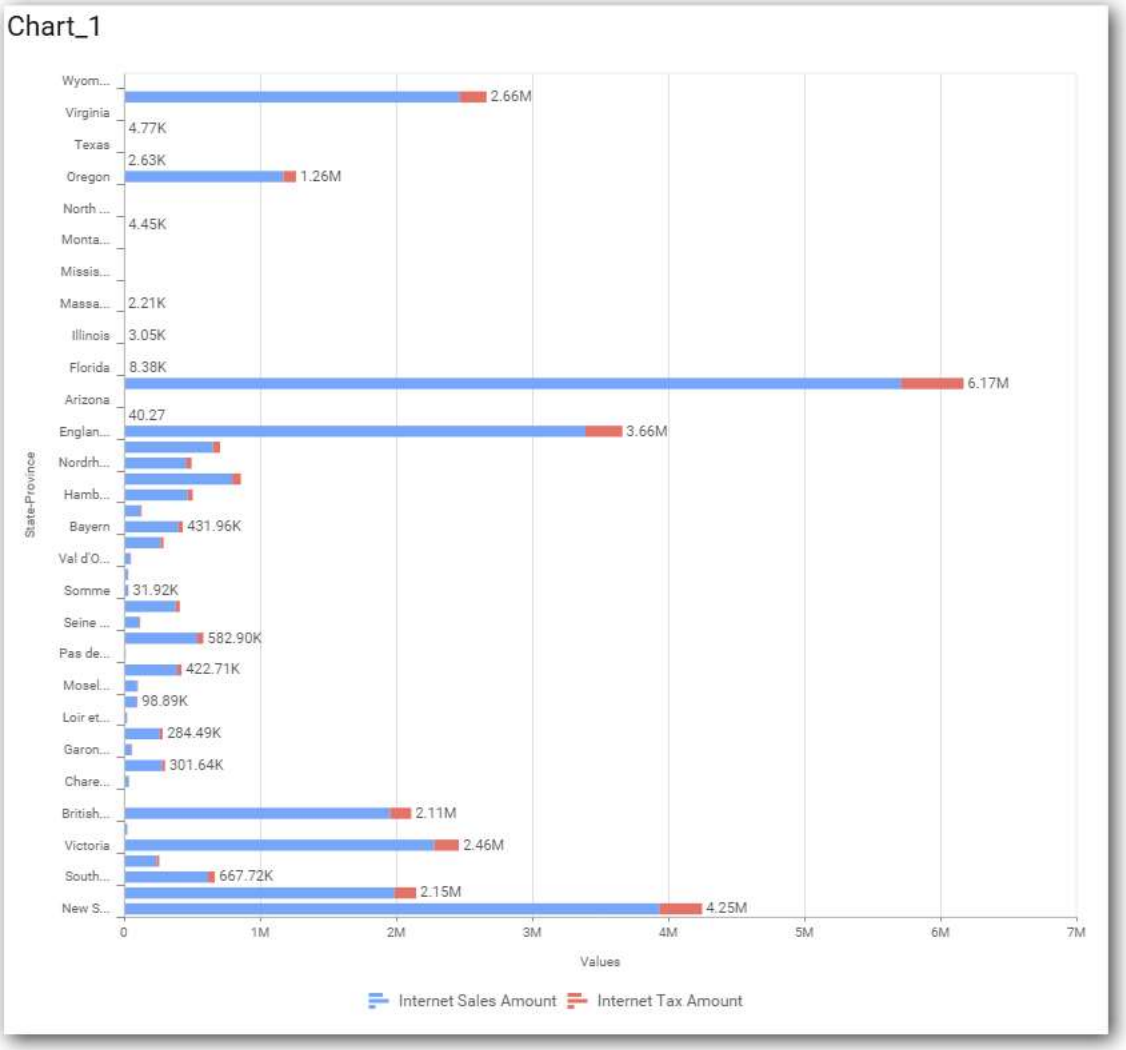
Click the respective data value marker in chart to drill into its inner level.



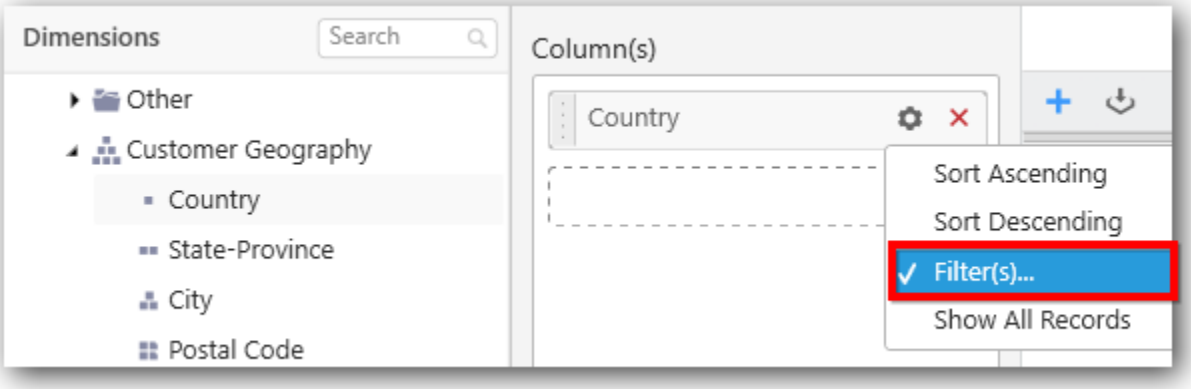
The drilled view of the chart is follows.



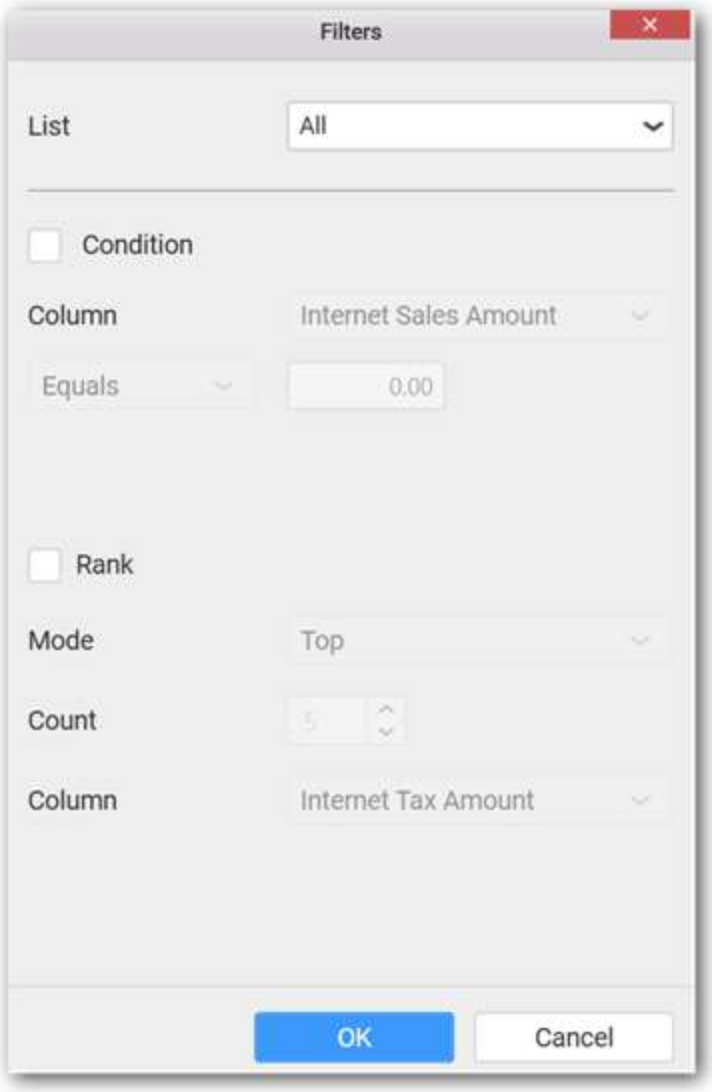
Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level.



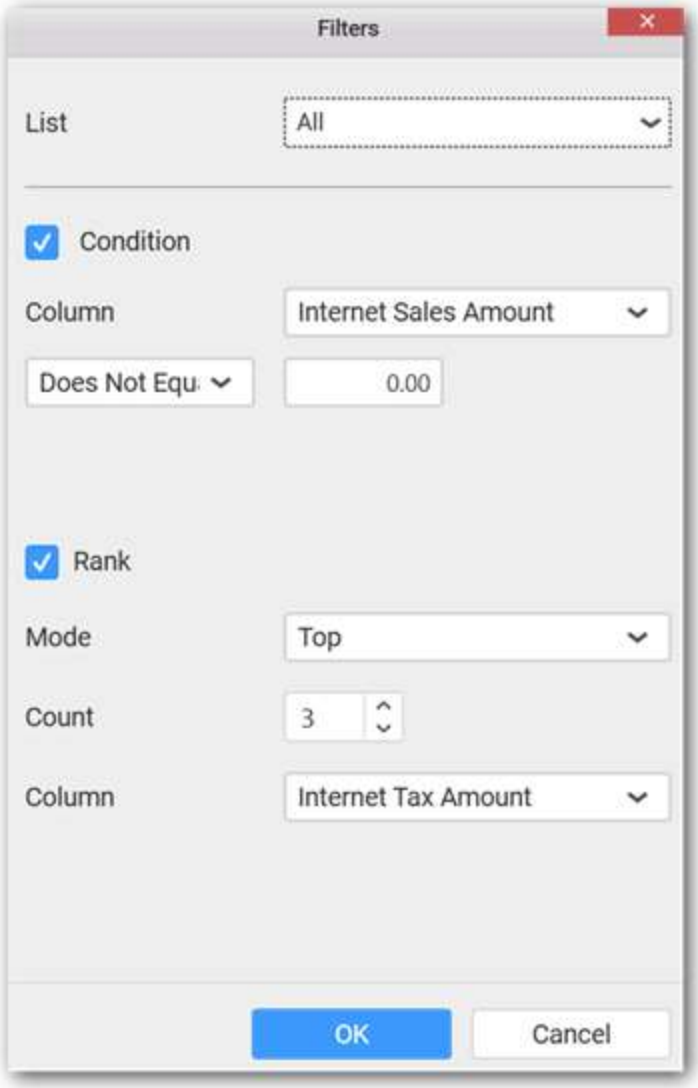
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



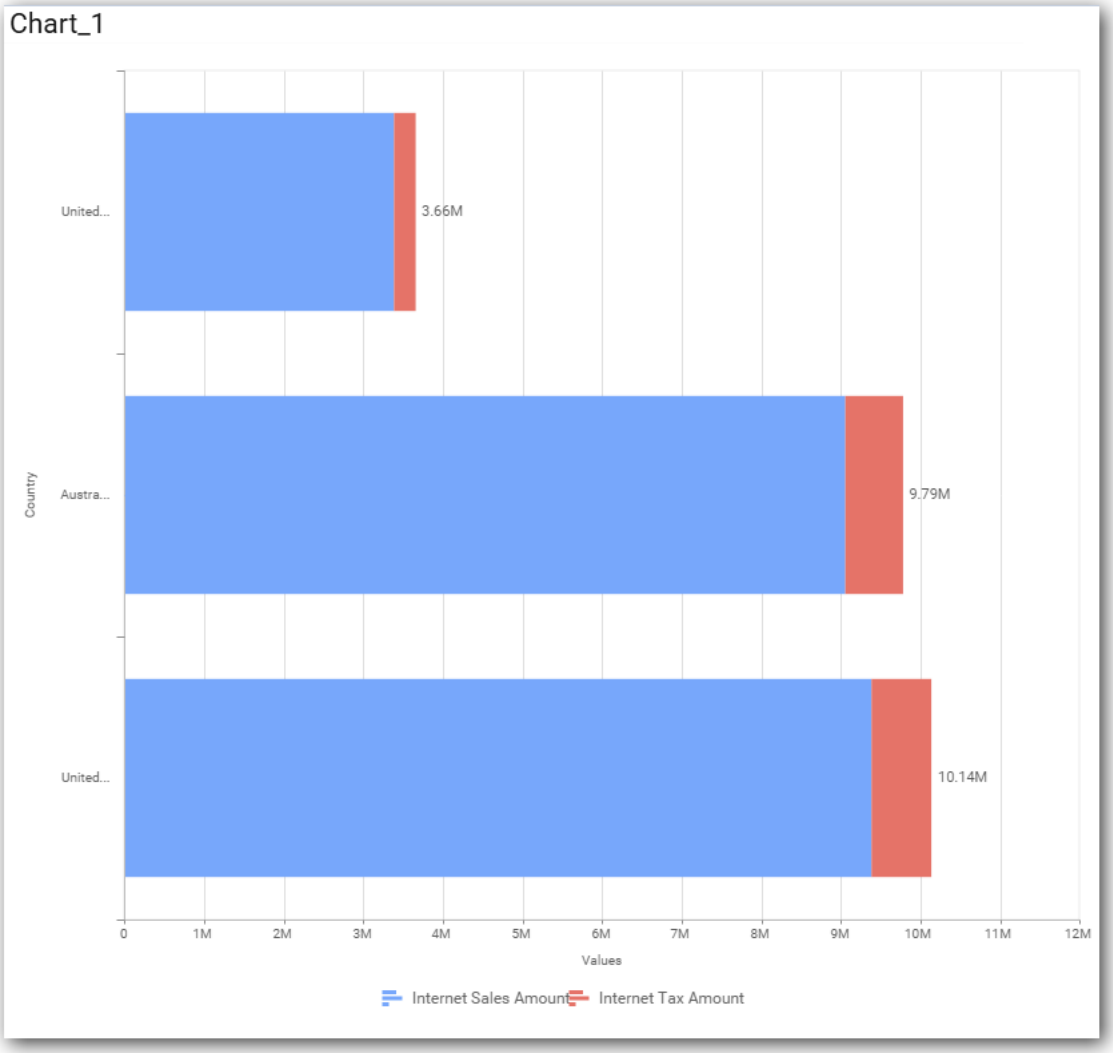
Select Filter(s)... to launch the Filters window.



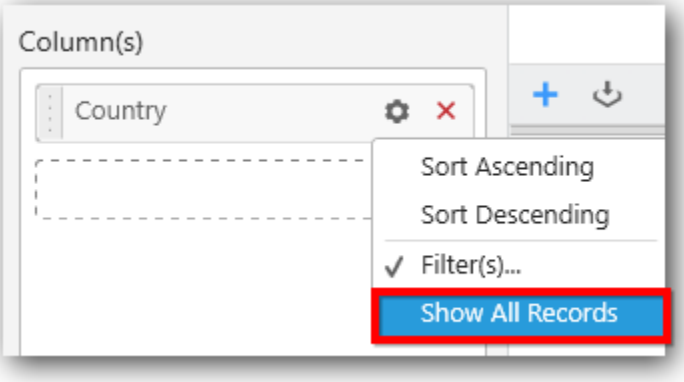
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

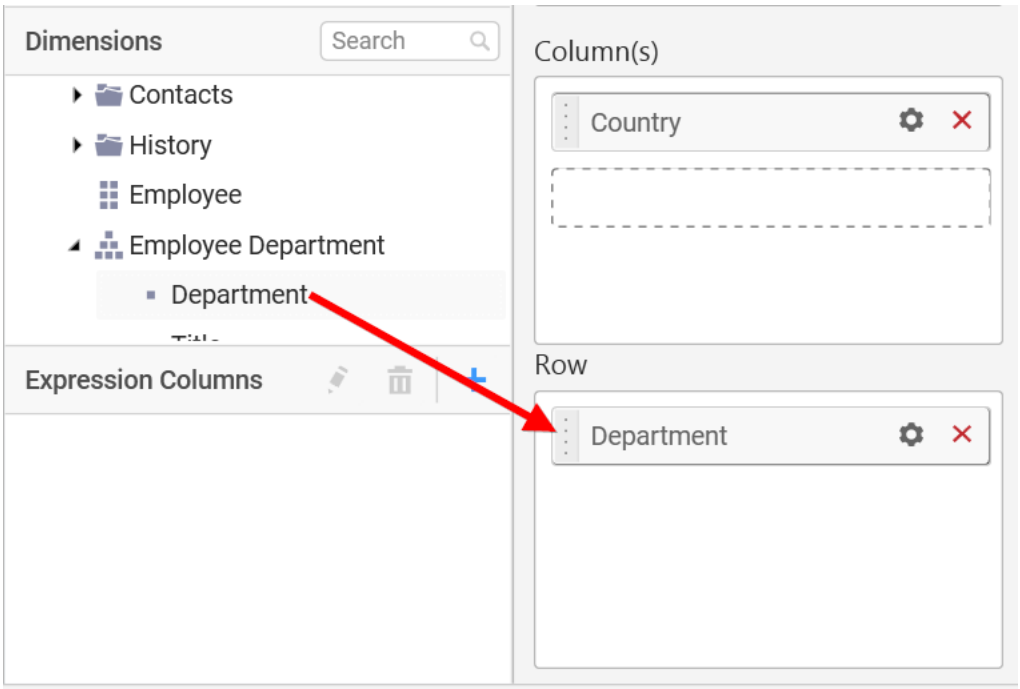


To show all records again click on Show All Records.

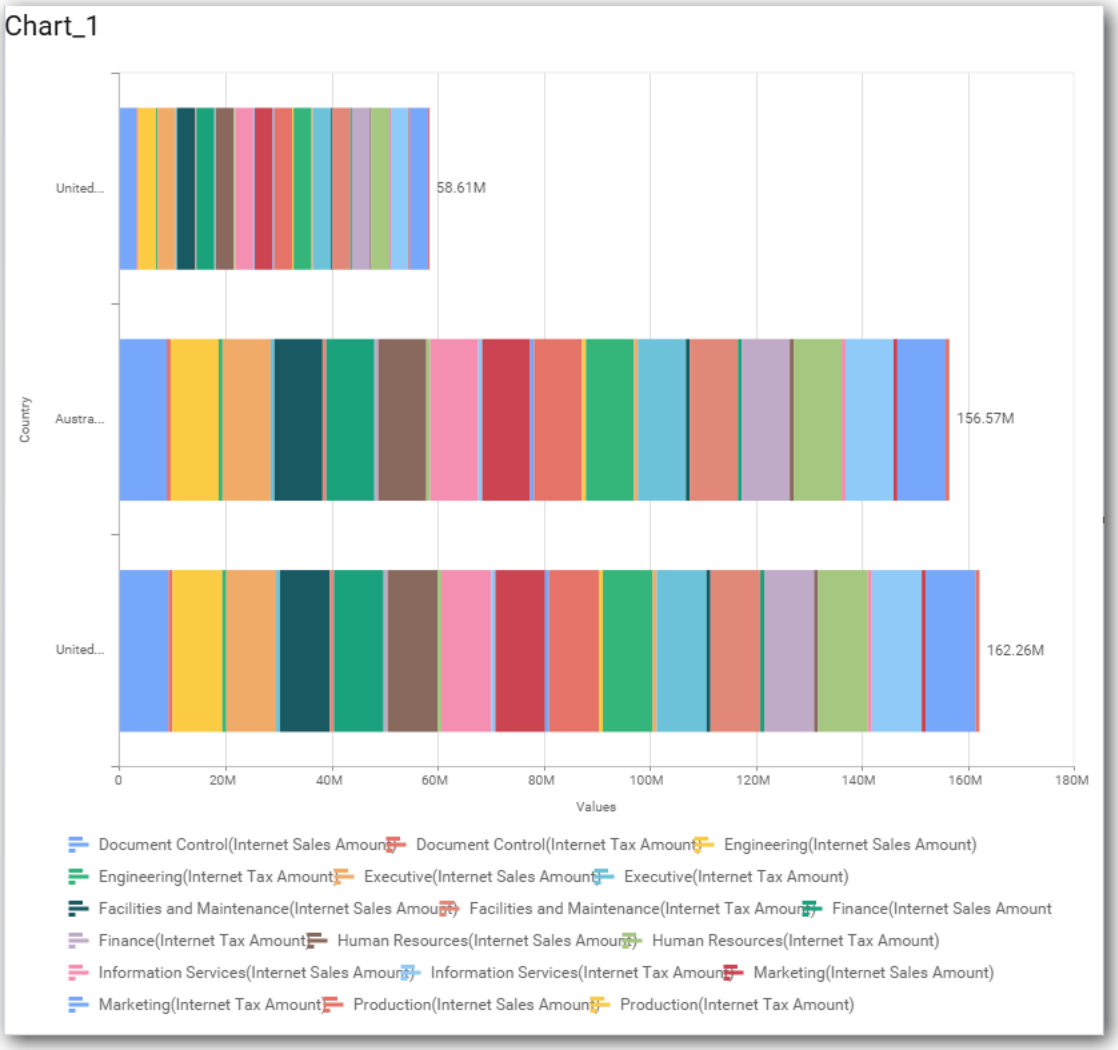


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.



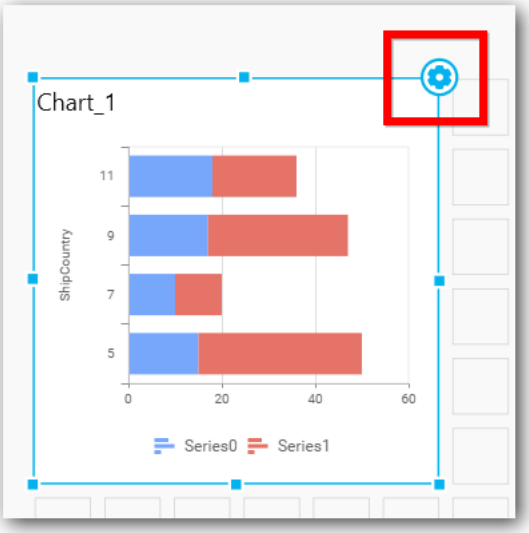
How to format Stacked Bar Chart?

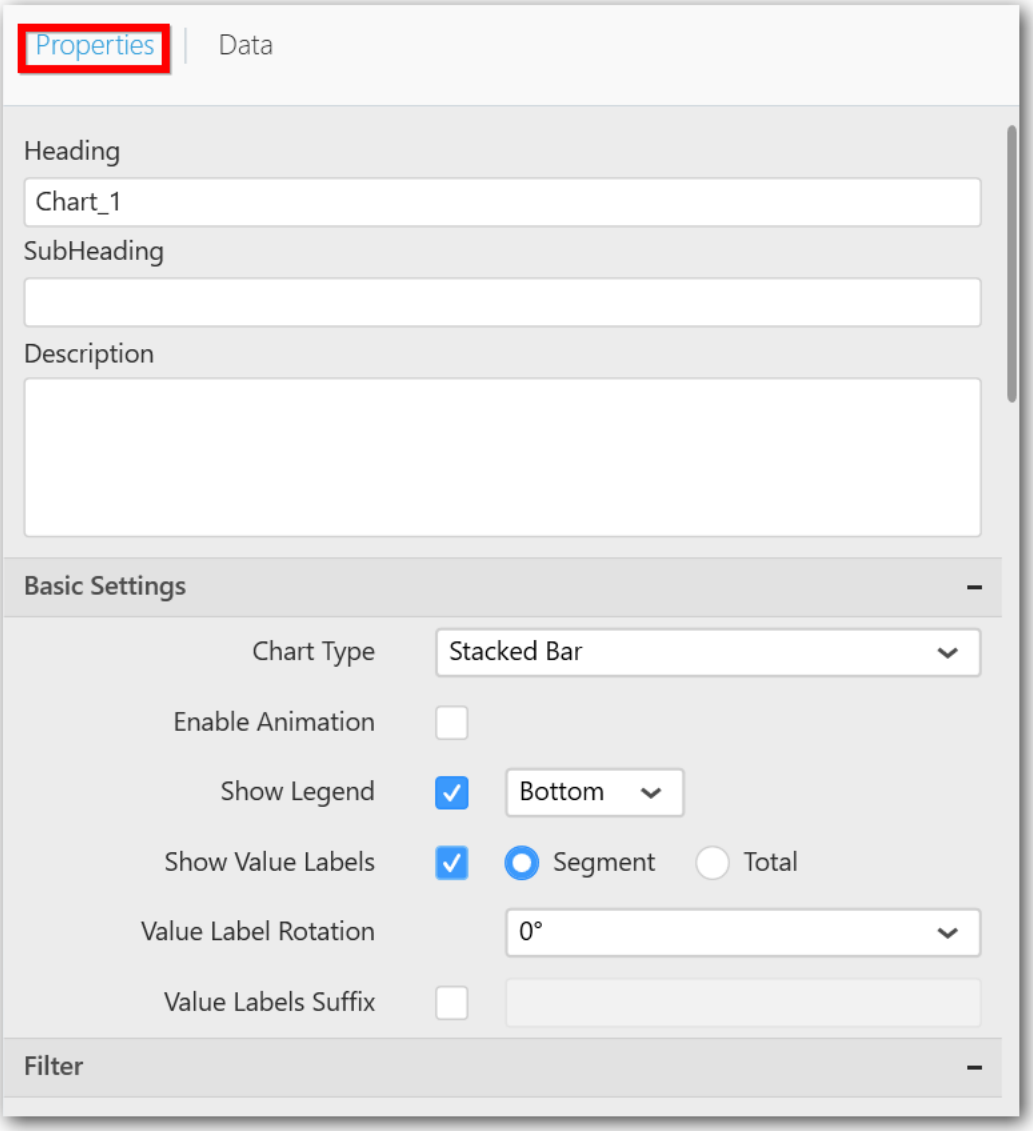
You can format the stacked bar chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into stacked bar chart follow the steps

1. Drag and drop the stacked bar chart into canvas and resize it to your required size.
2. Configure the data into stacked bar chart.
3. Focus on the stacked chart and Click on Widget Settings.

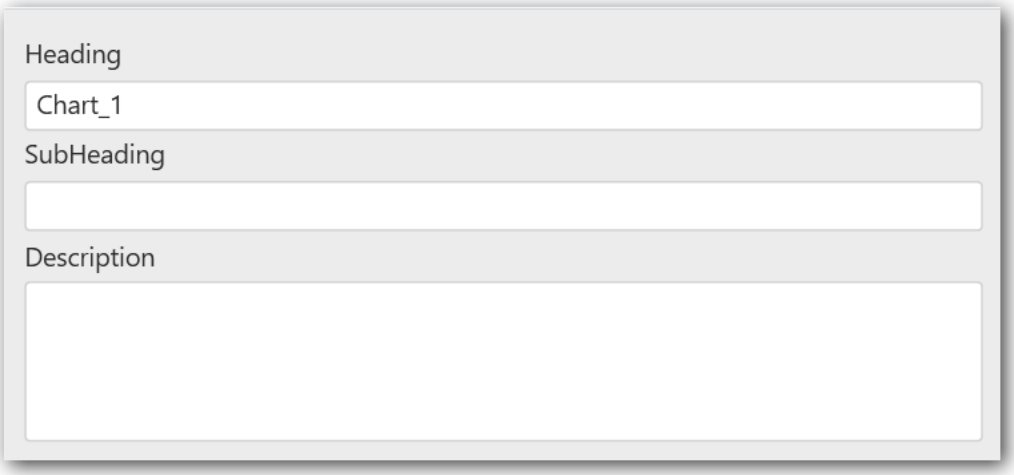
The property window will be opened.





You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this stacked bar chart widget.

SubHeading

This allows you to set sub-title for this stacked bar chart widget.

Description

This allows you to set description for this stacked bar chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

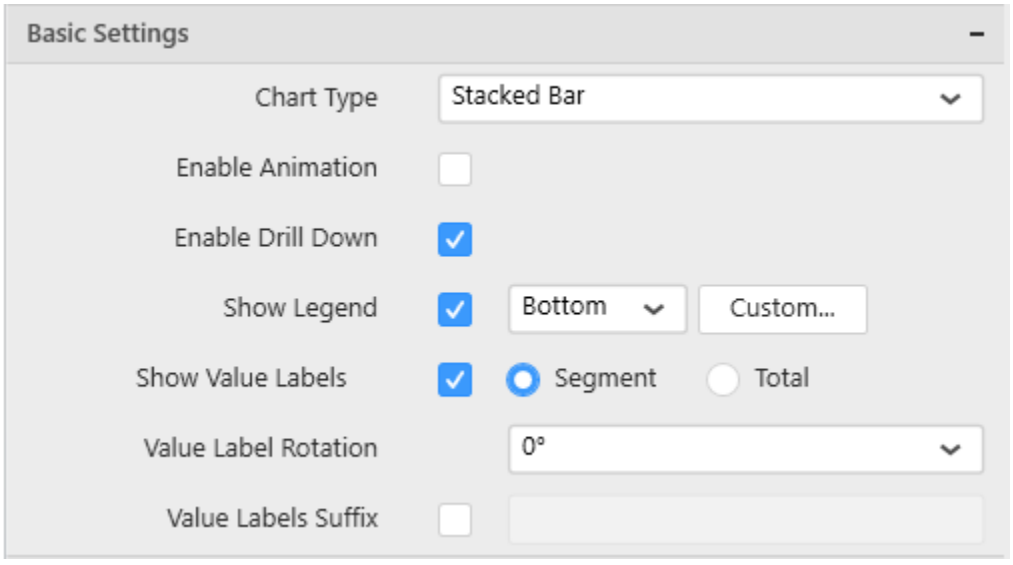


Chart Type

This allows you to switch the widget view from current chart type to another convertible chart type.

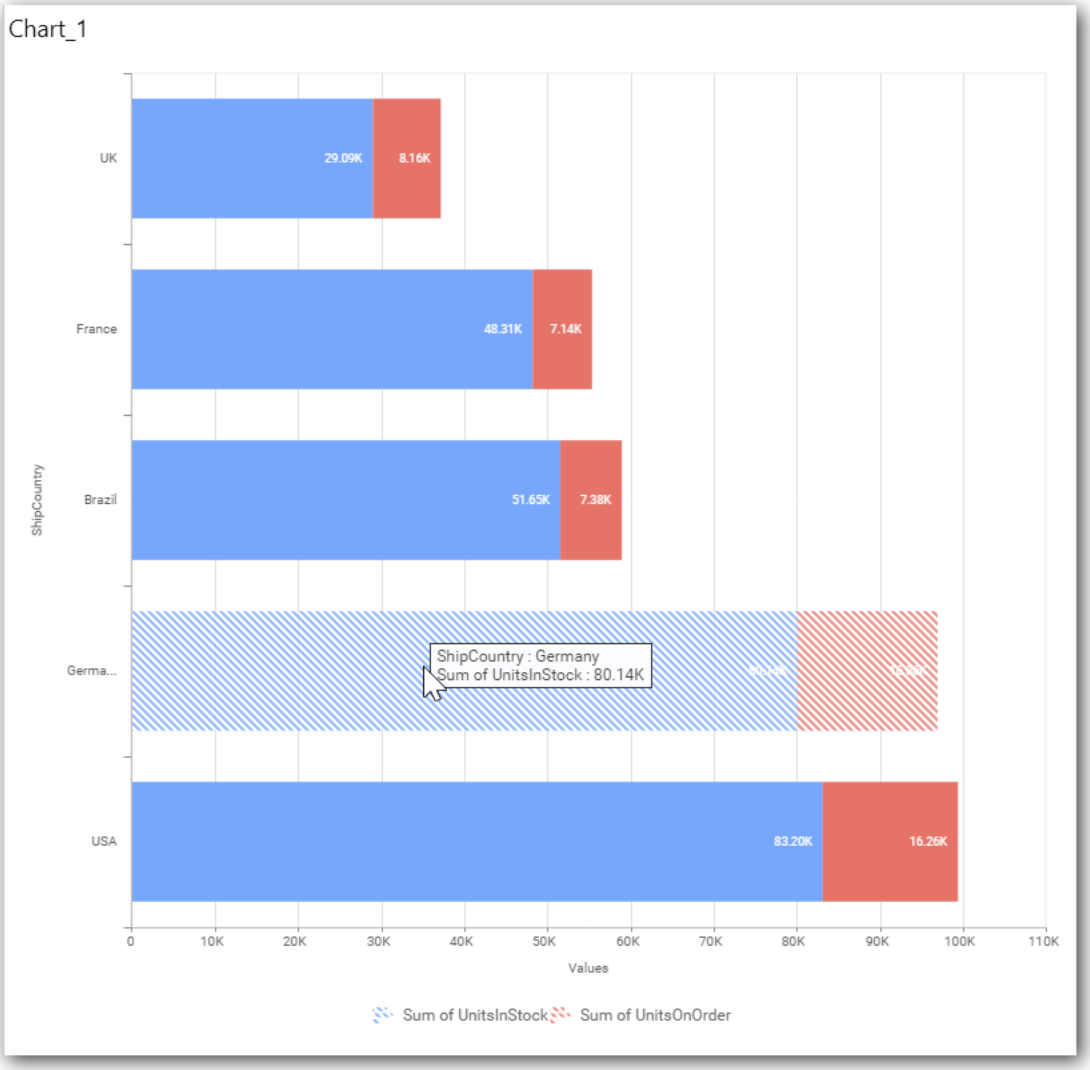
Enable Animation

This allows you to enable the series rendering in animated mode.

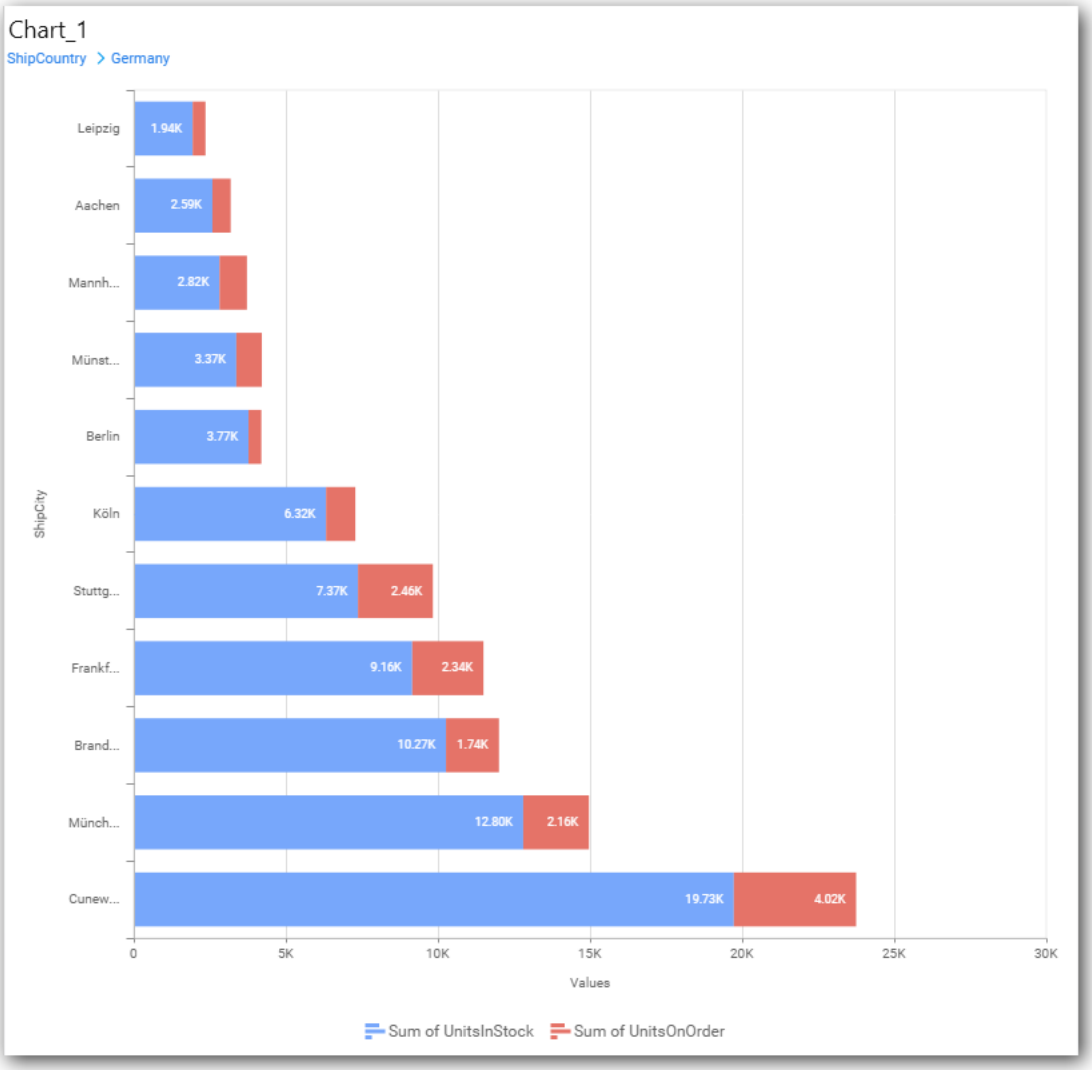
Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View

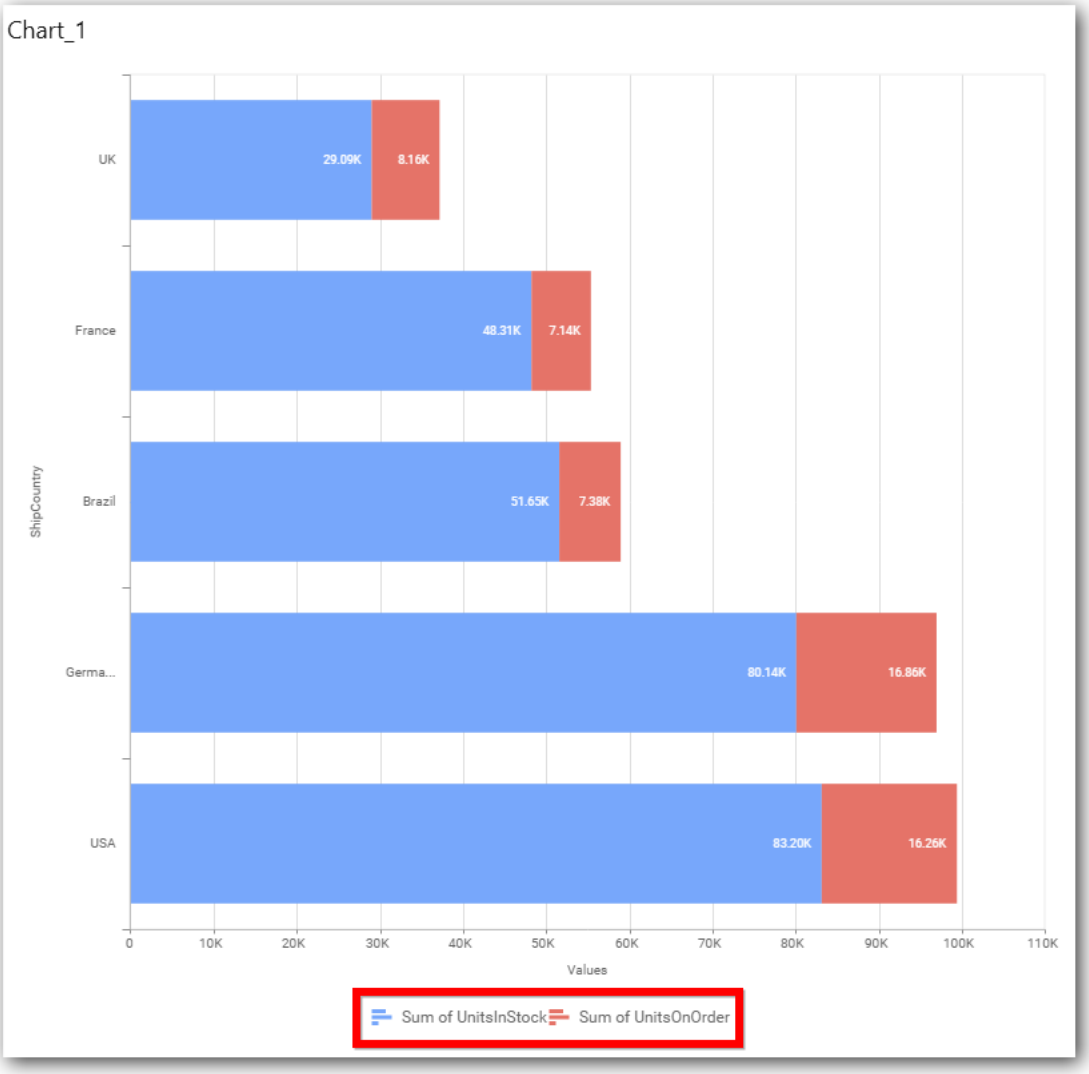


Drilled View



Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling the option Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

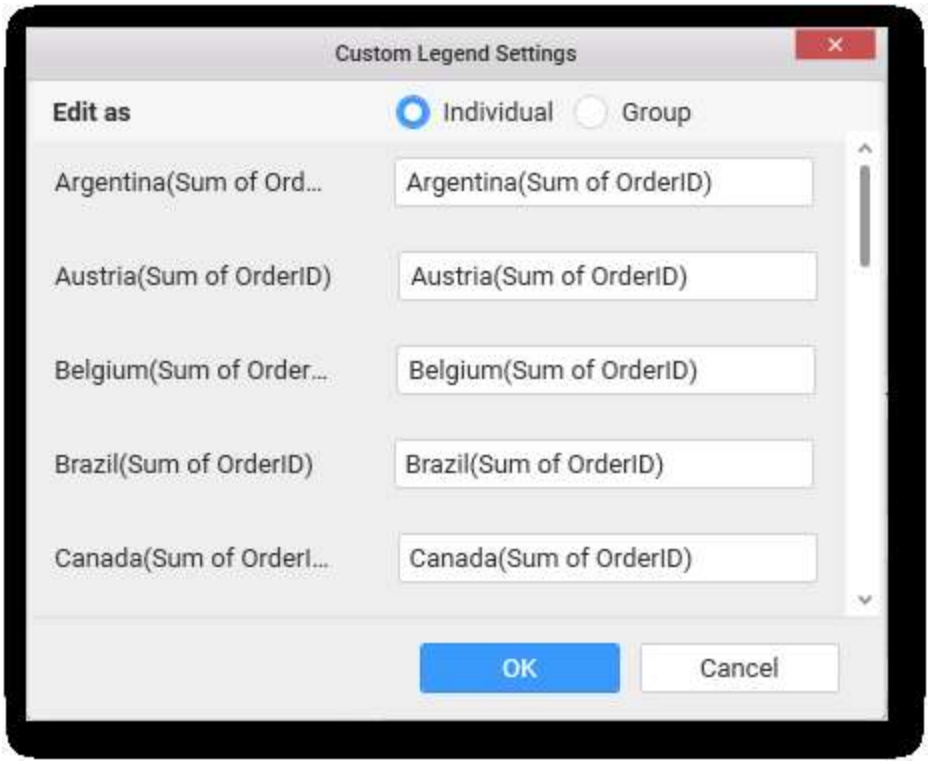
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

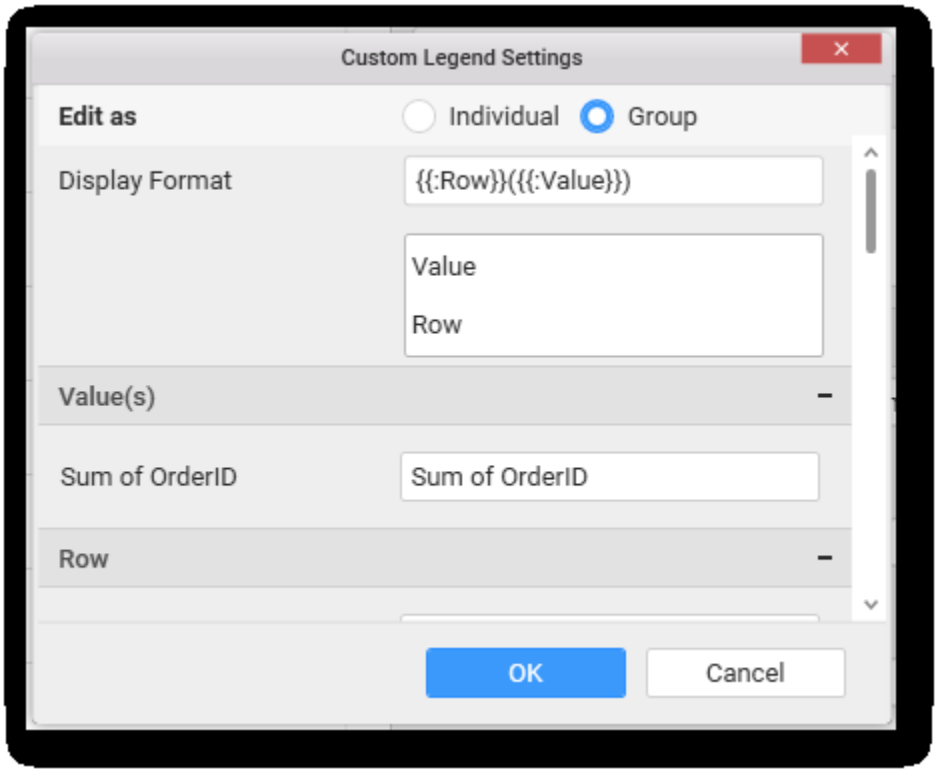
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.



Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)

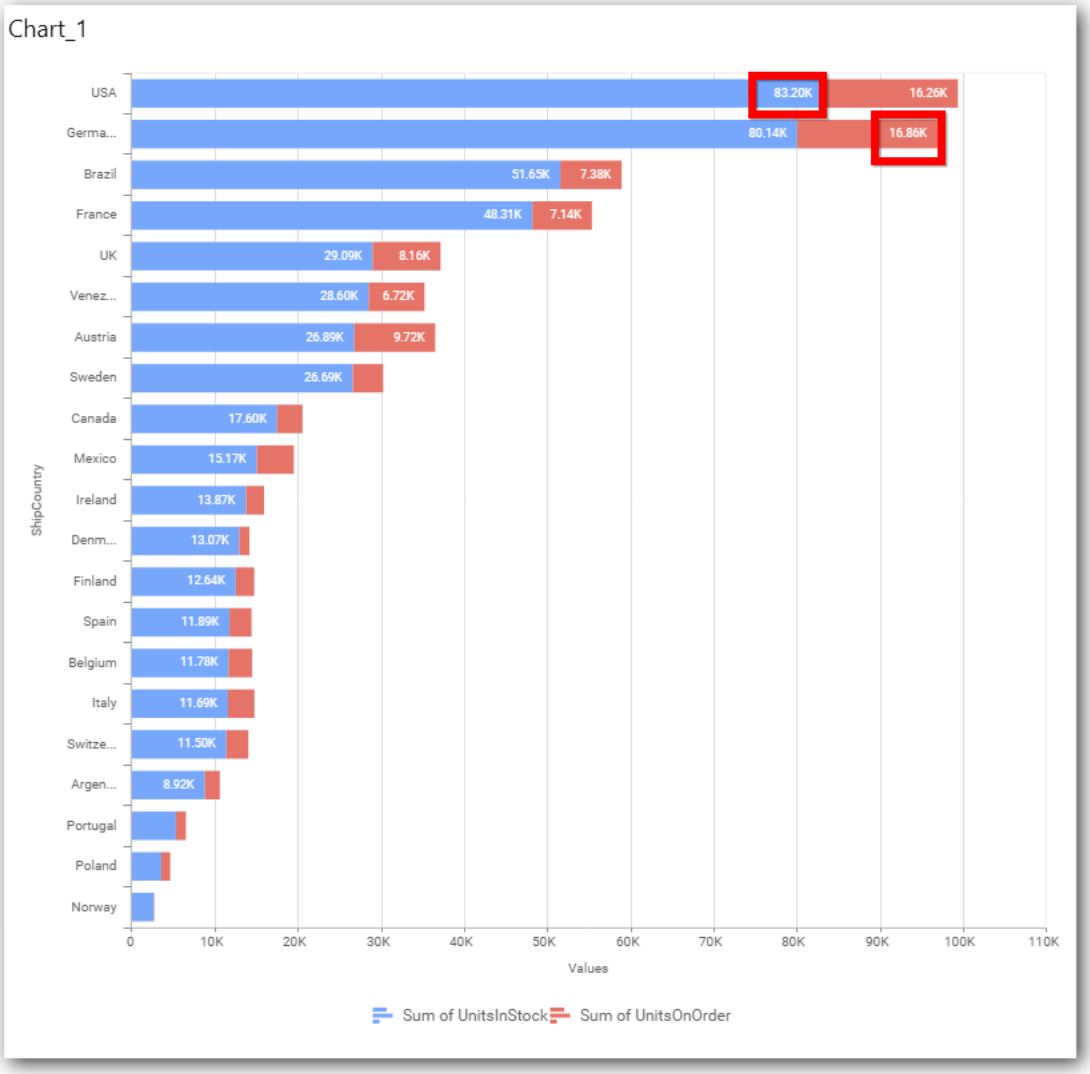
Show Value Labels

This allows you to toggle the visibility of value labels. When you toggle on, two options will be provided to change the display mode of the labels.

1. Segment
2. Total

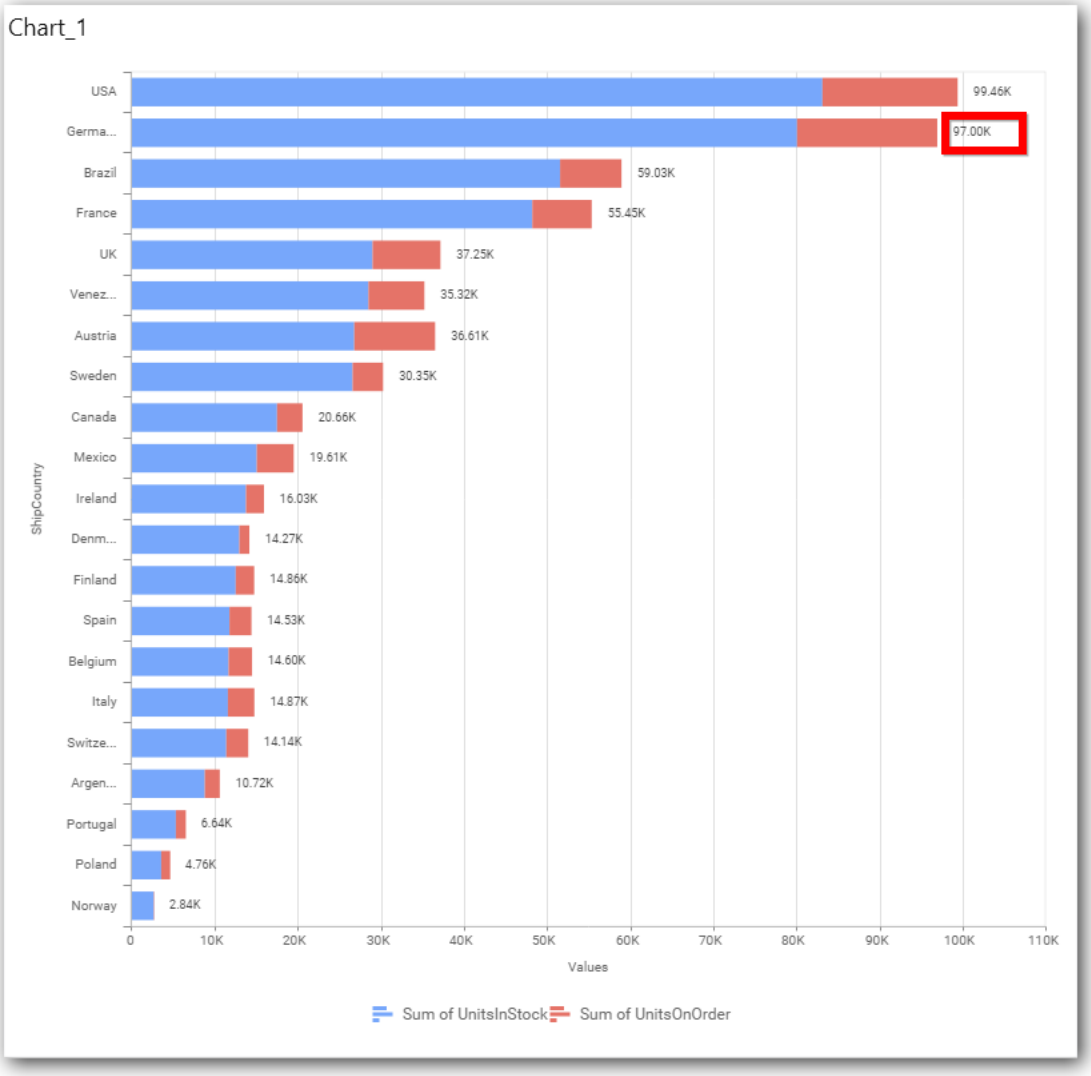
Segment

Displays value label for each series segment.



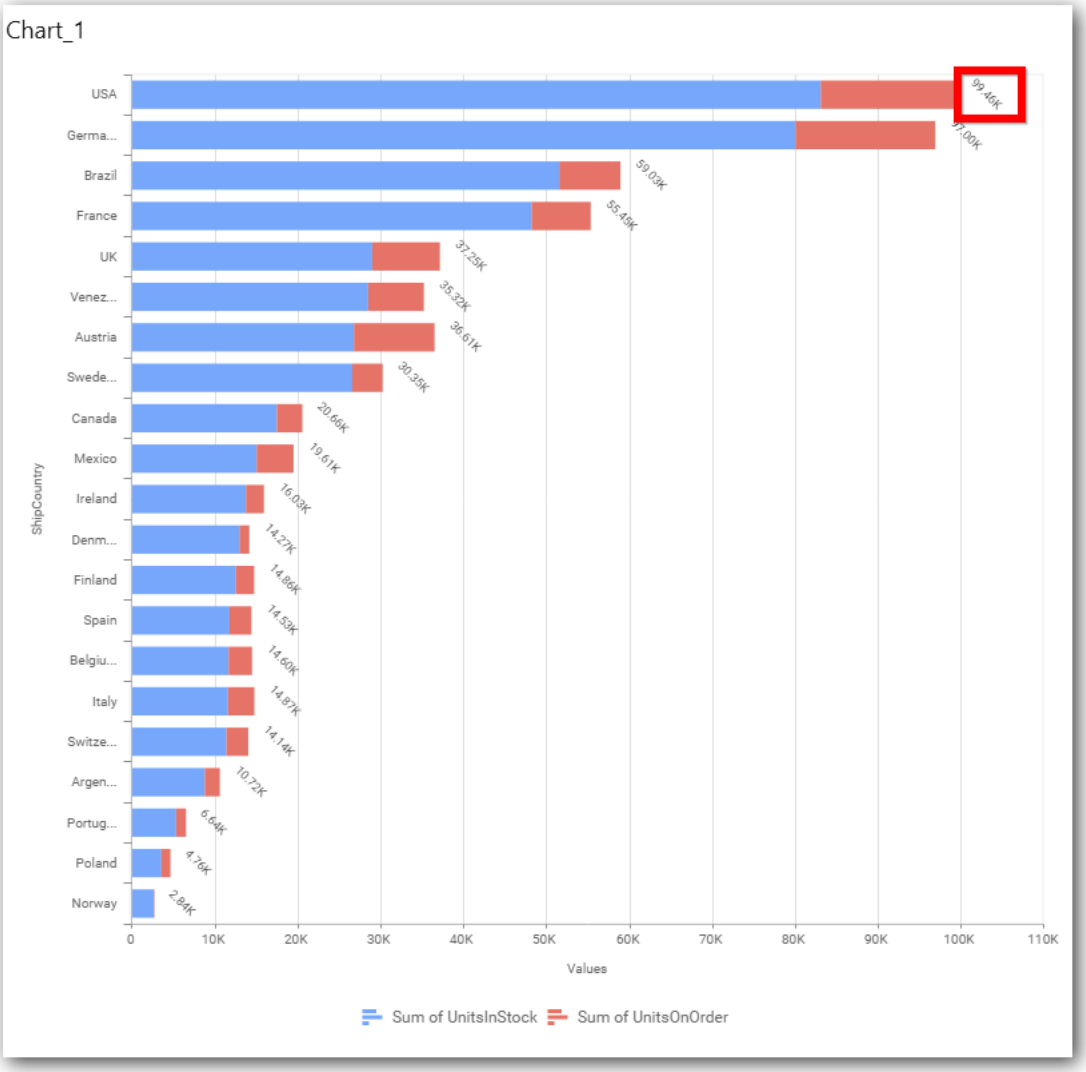
Total

Displays sum value label of all the stacked segment on the top most segment.



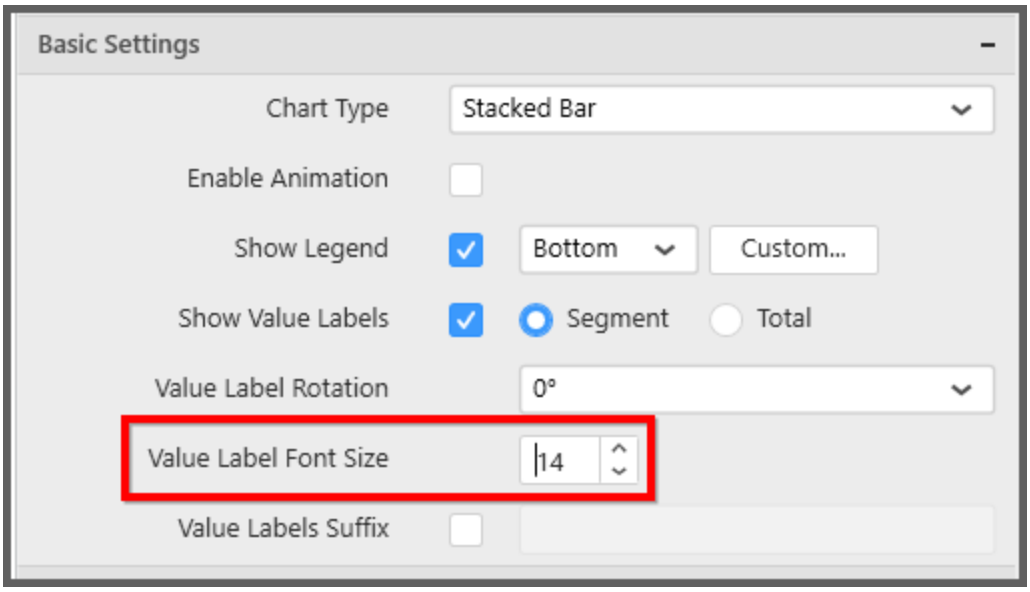
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



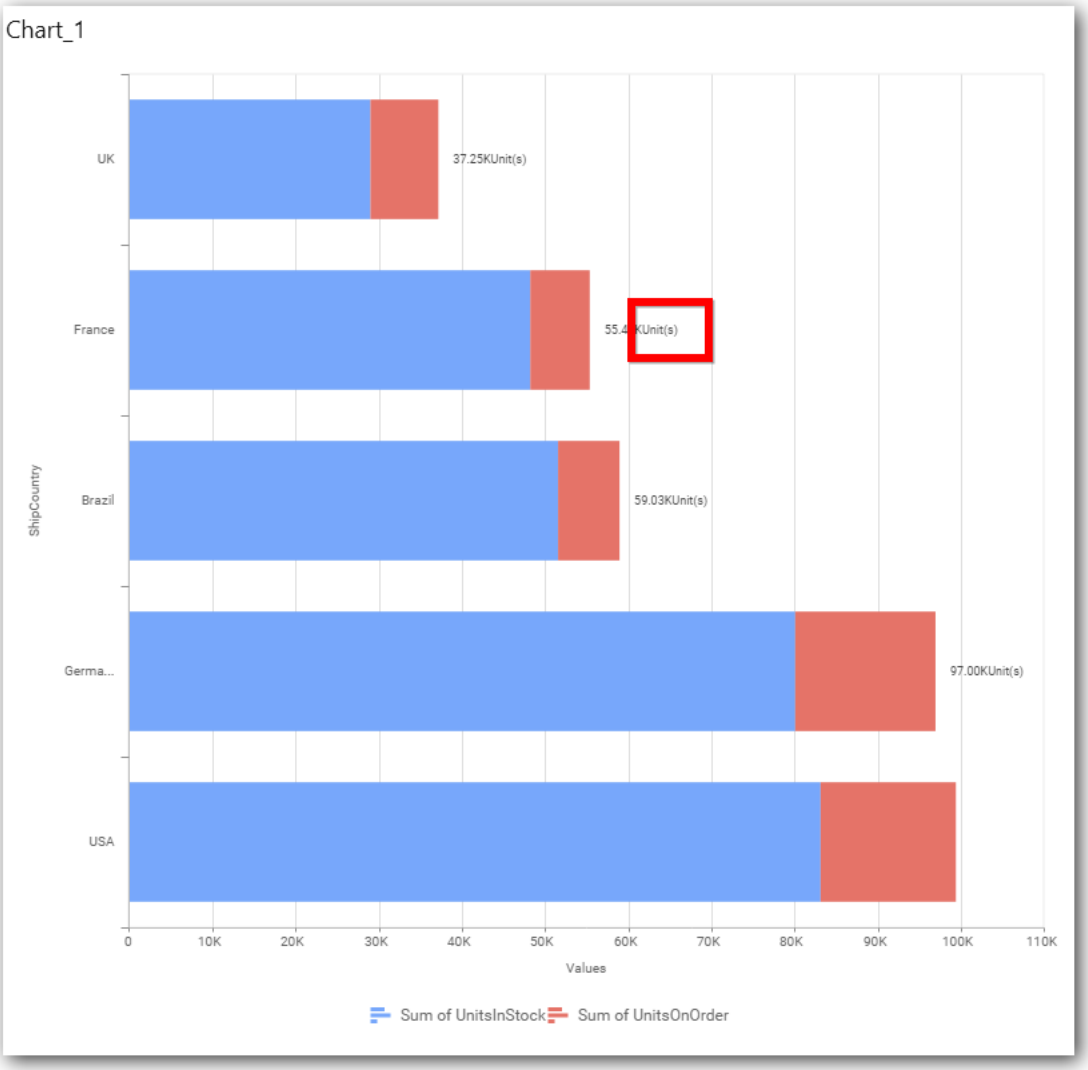
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

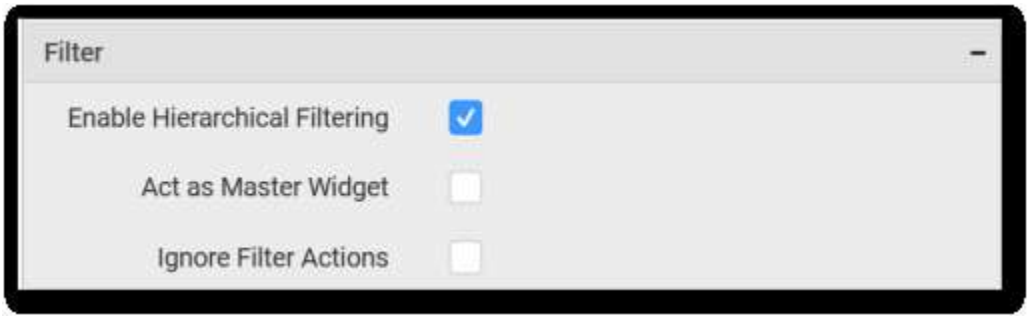


Value Labels Suffix

Allows you to set suffix to the value/data labels.



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

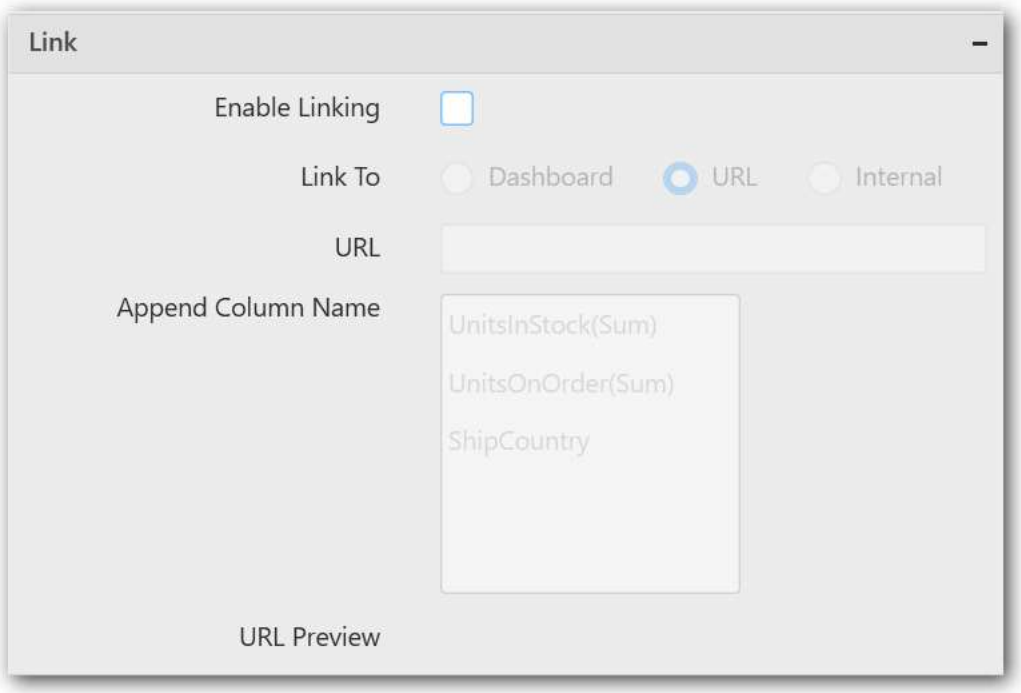
Act as Master Widget

This allows you to define this stacked bar chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

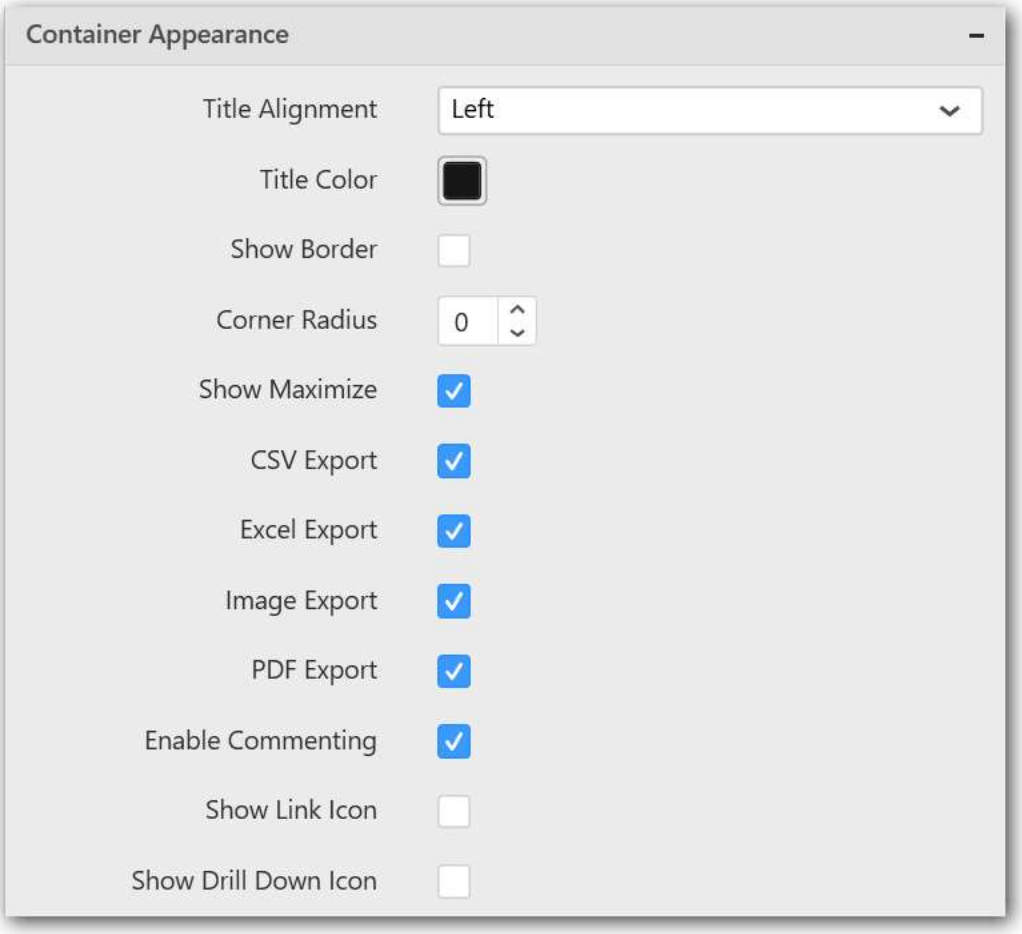
This allows you to define this stacked bar chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this stacked bar chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this stacked bar chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this stacked bar chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

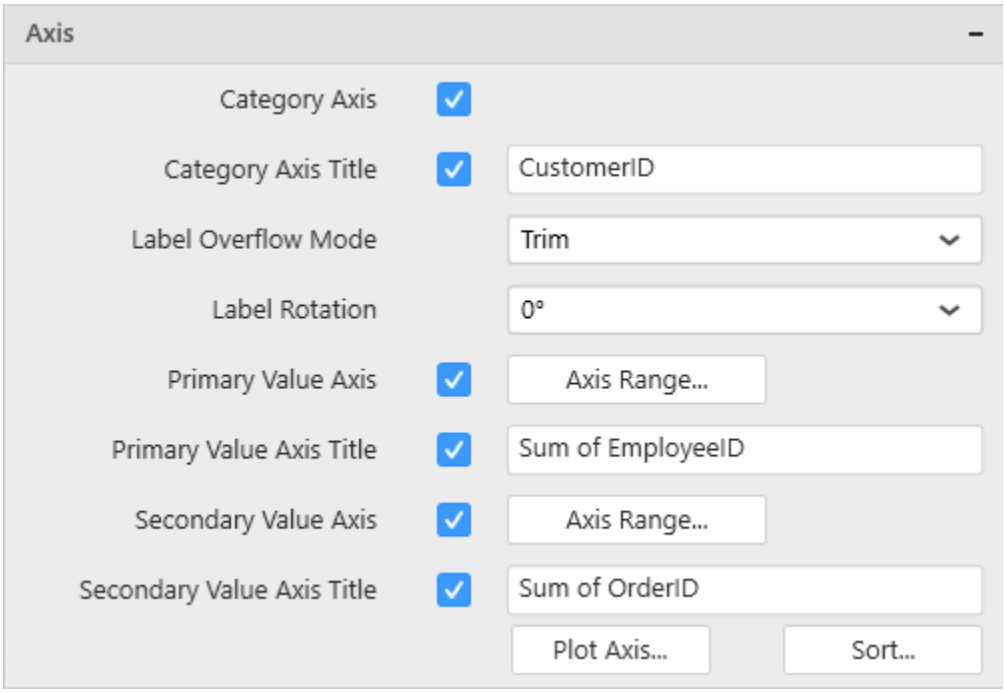
Image Export

This allows you to enable/disable the image export option for this stacked bar chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

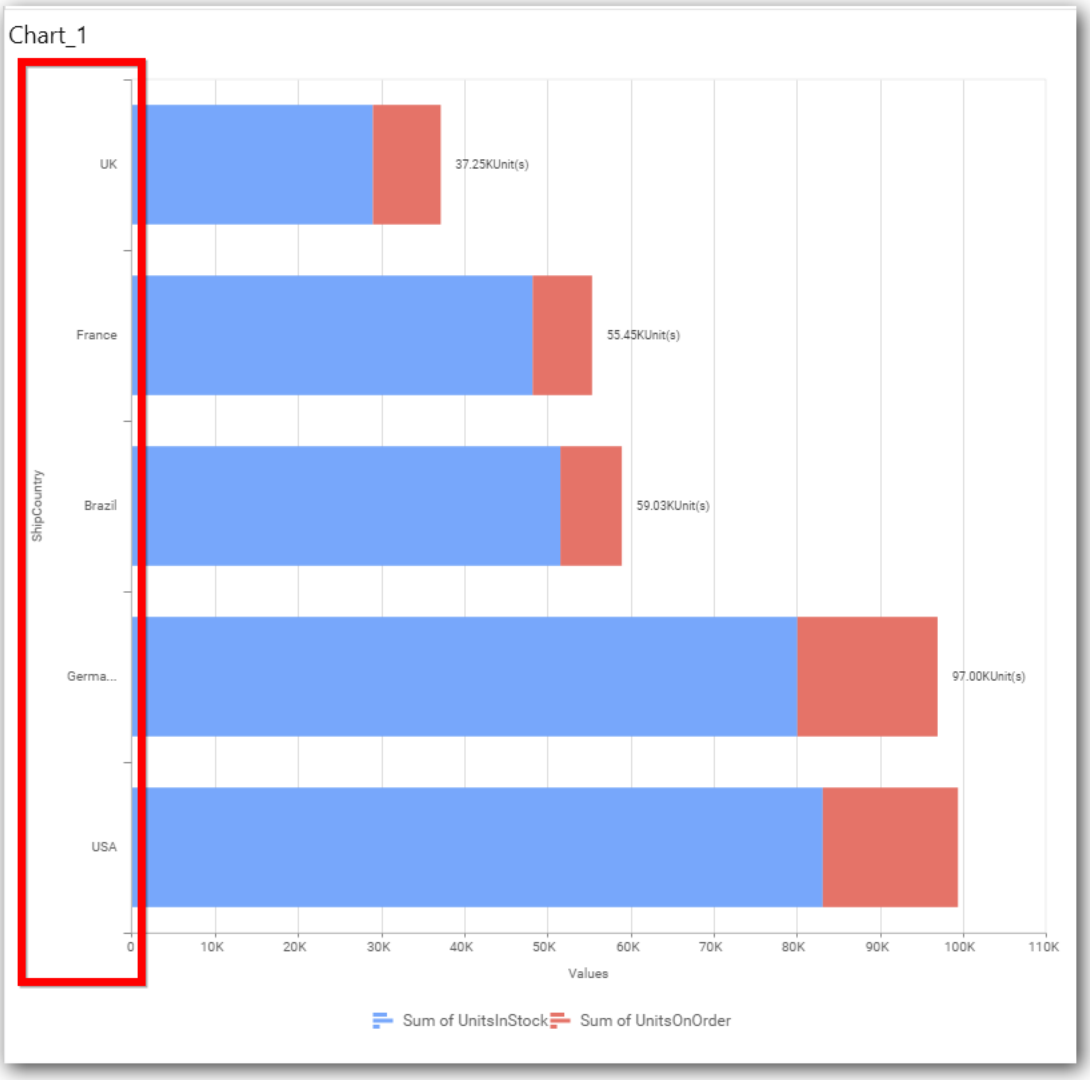
Axis Settings



This section allows you to customize the axis settings in chart.

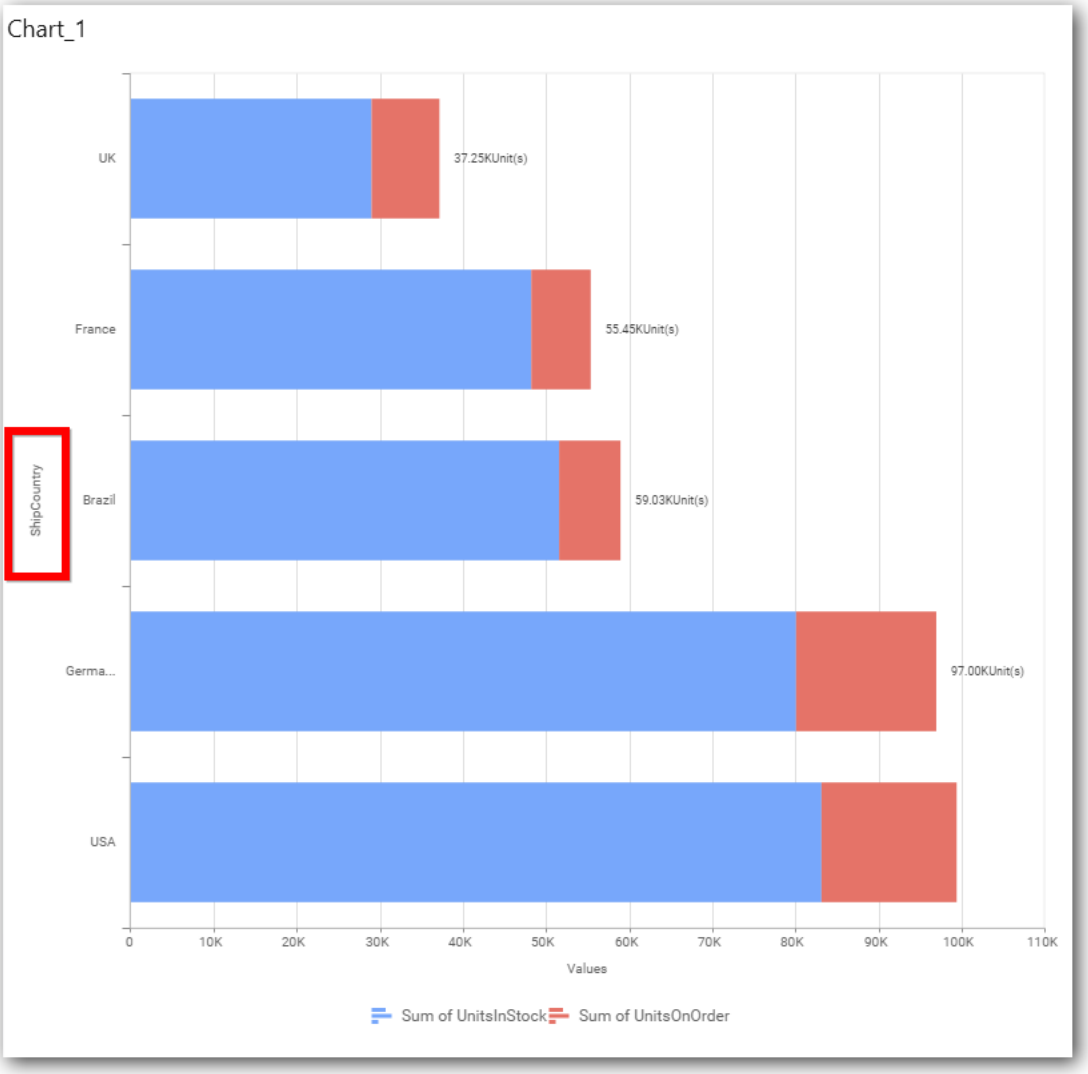
Category Axis

This allows you to enable/edit the Category Axis title. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

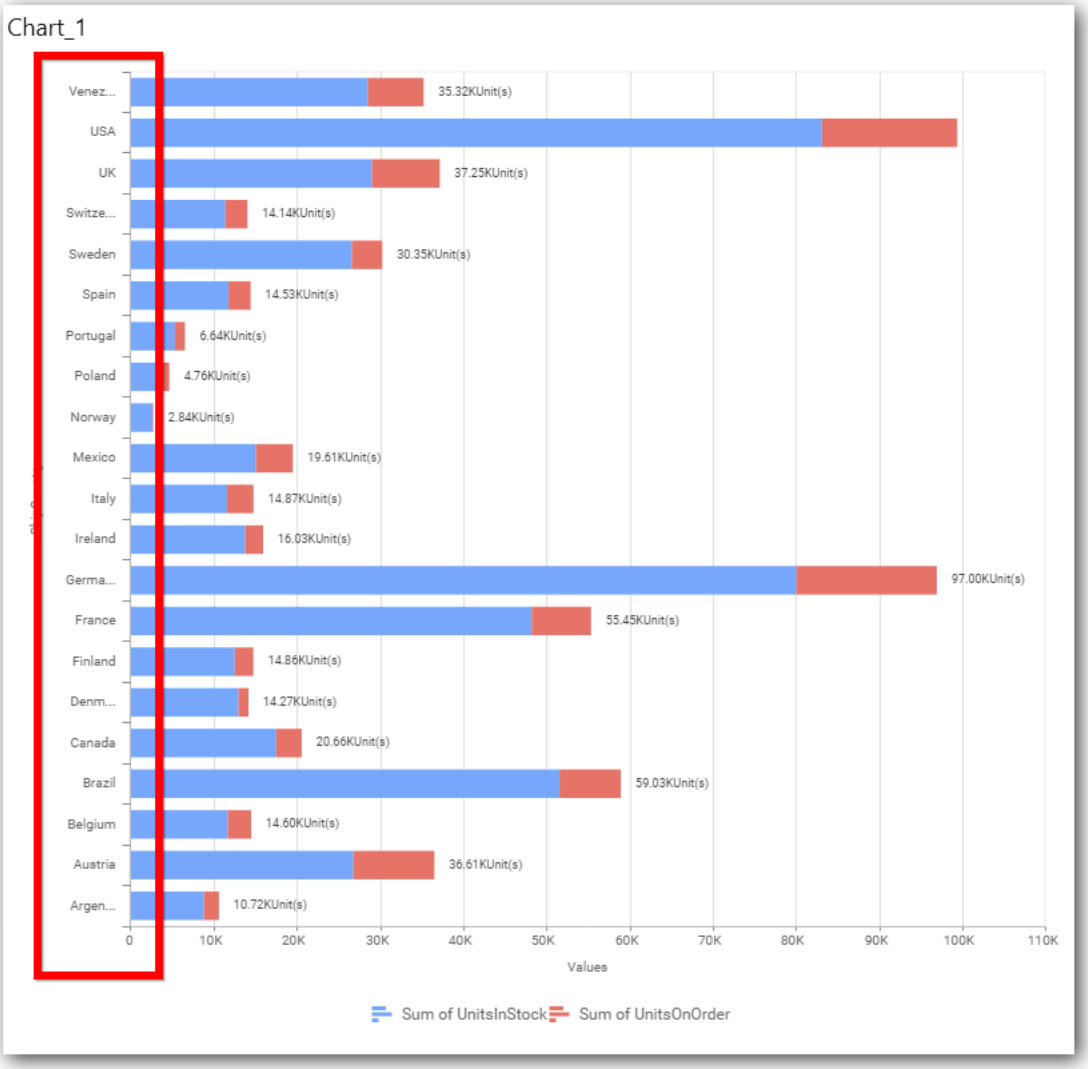


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

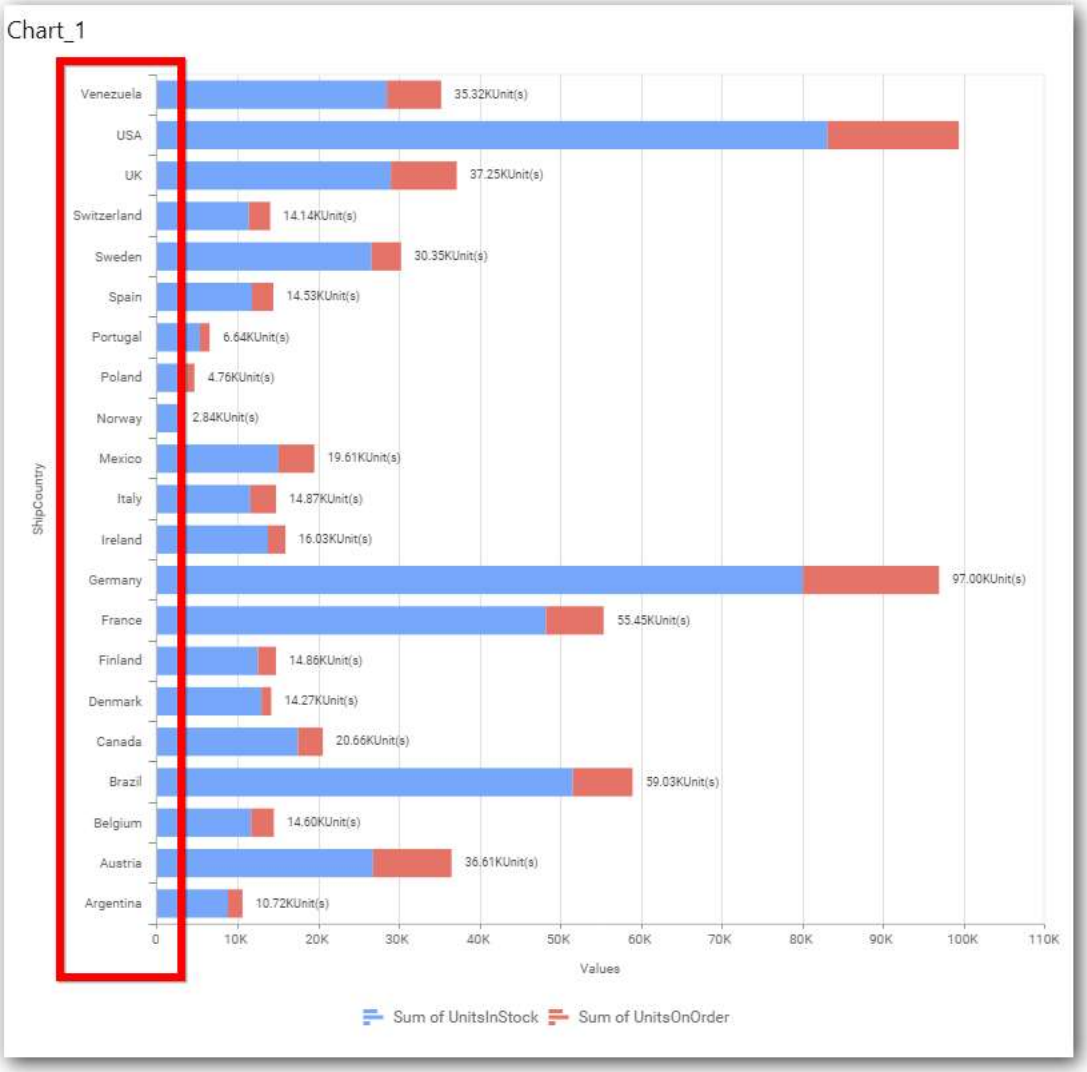
Trim

This option trims the end of overlapping label in the axis.



Hide

This option hides the overlapping label in the axis.

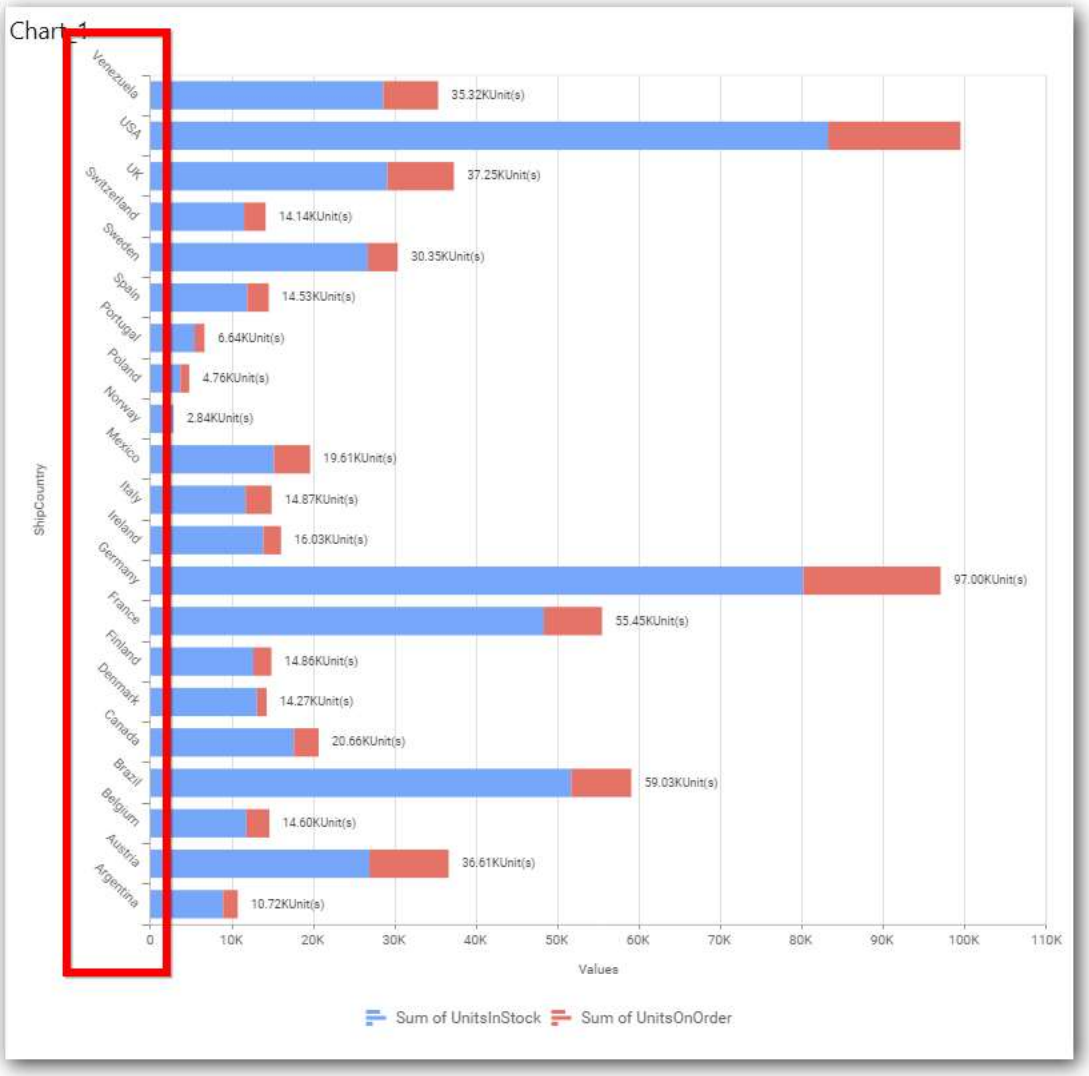


Wrap

This option wraps the lengthy label text in the axis.

Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

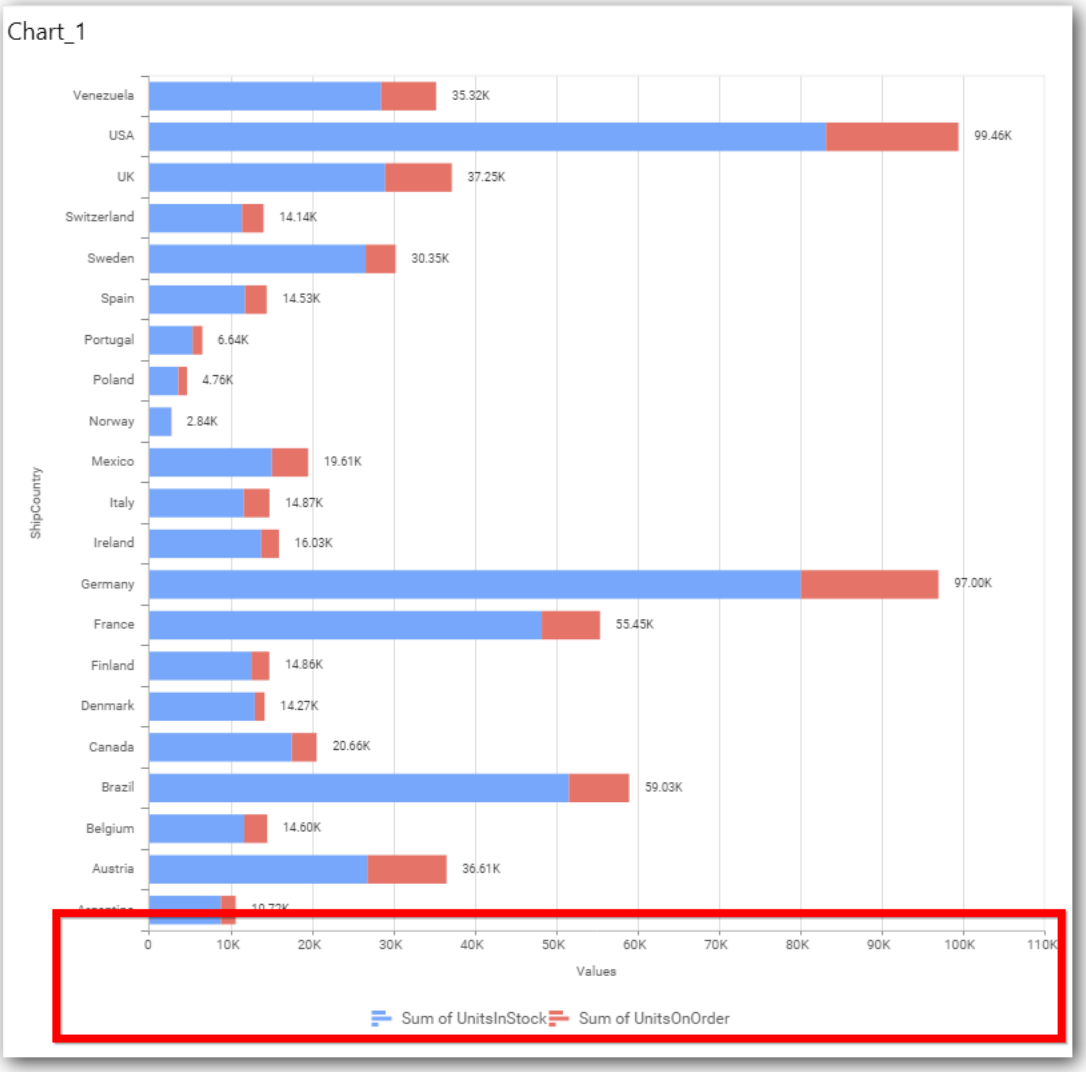


Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.

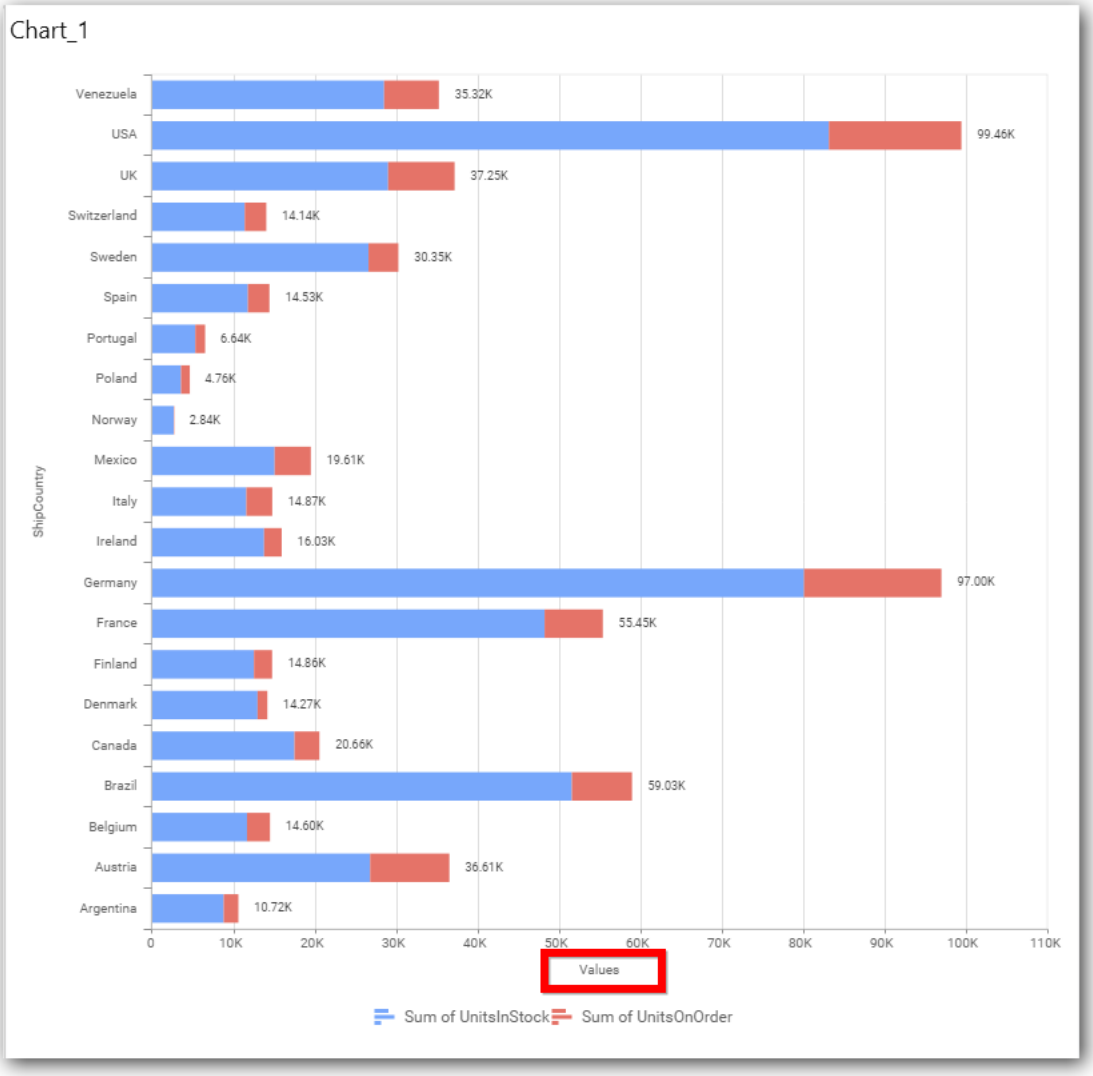
Primary Value Axis

This allows you to enable/edit the Primary Value Axis title. It will reflect in chart area axis.



Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

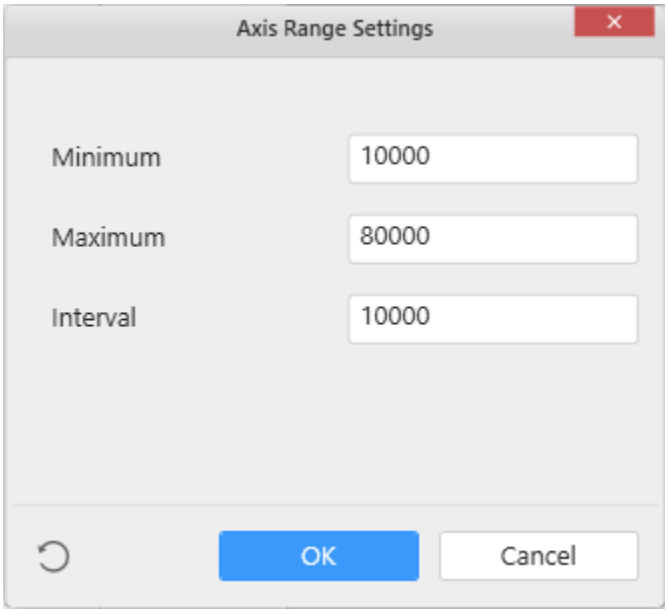
This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

Primary Value Axis Range

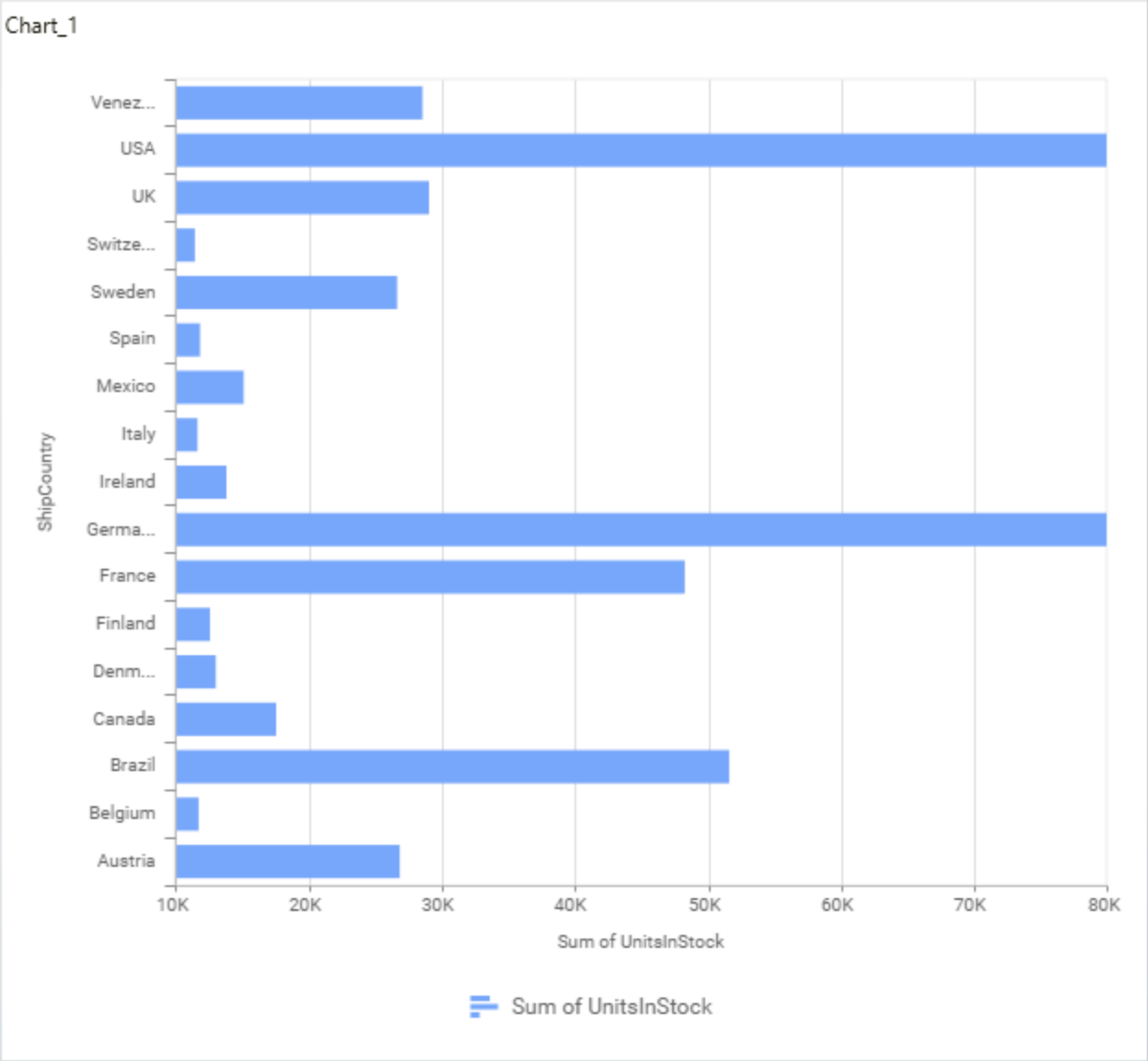
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

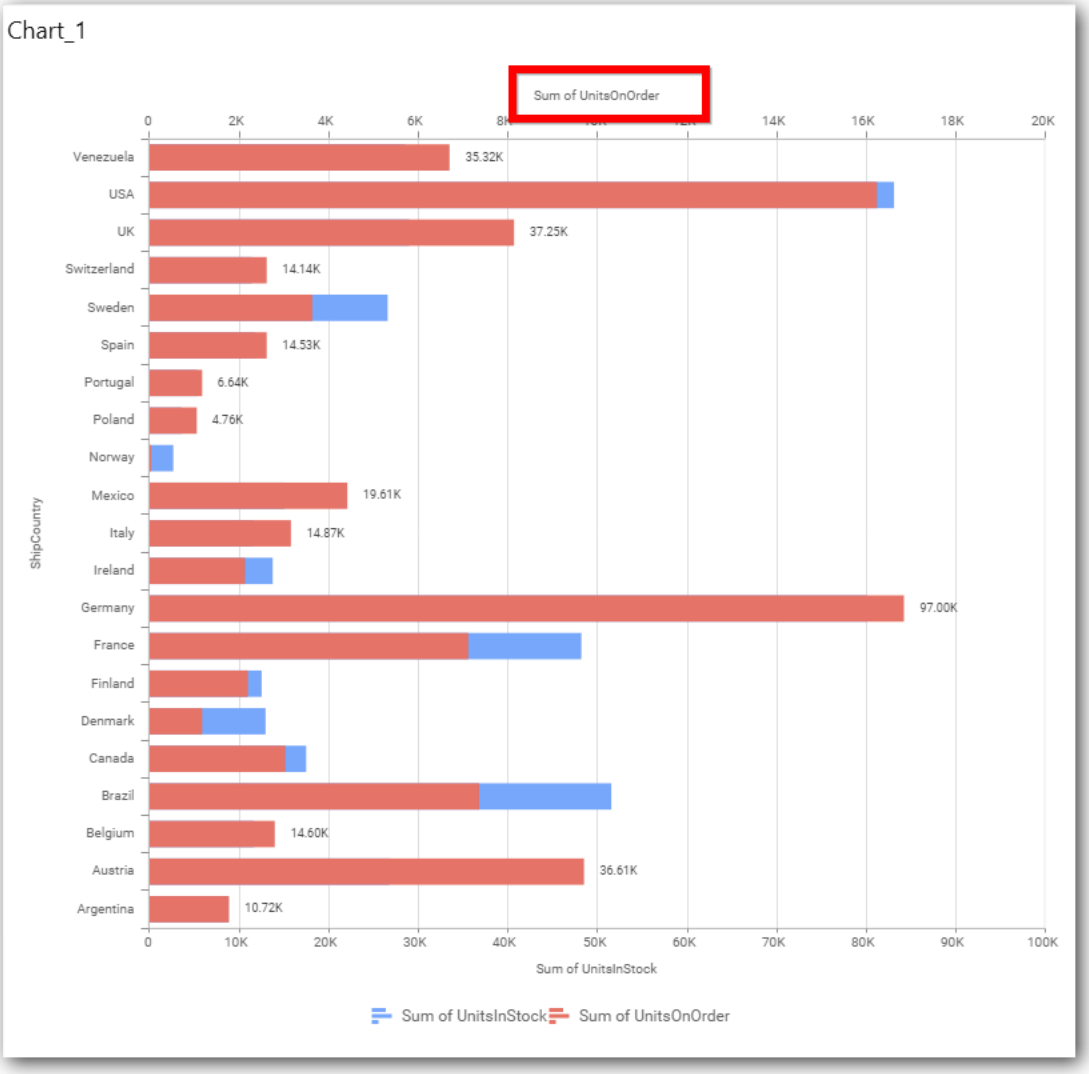


Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.

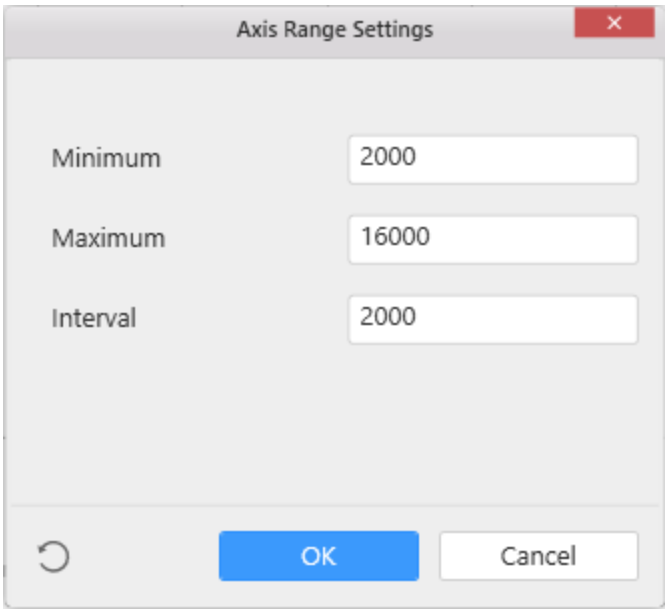
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

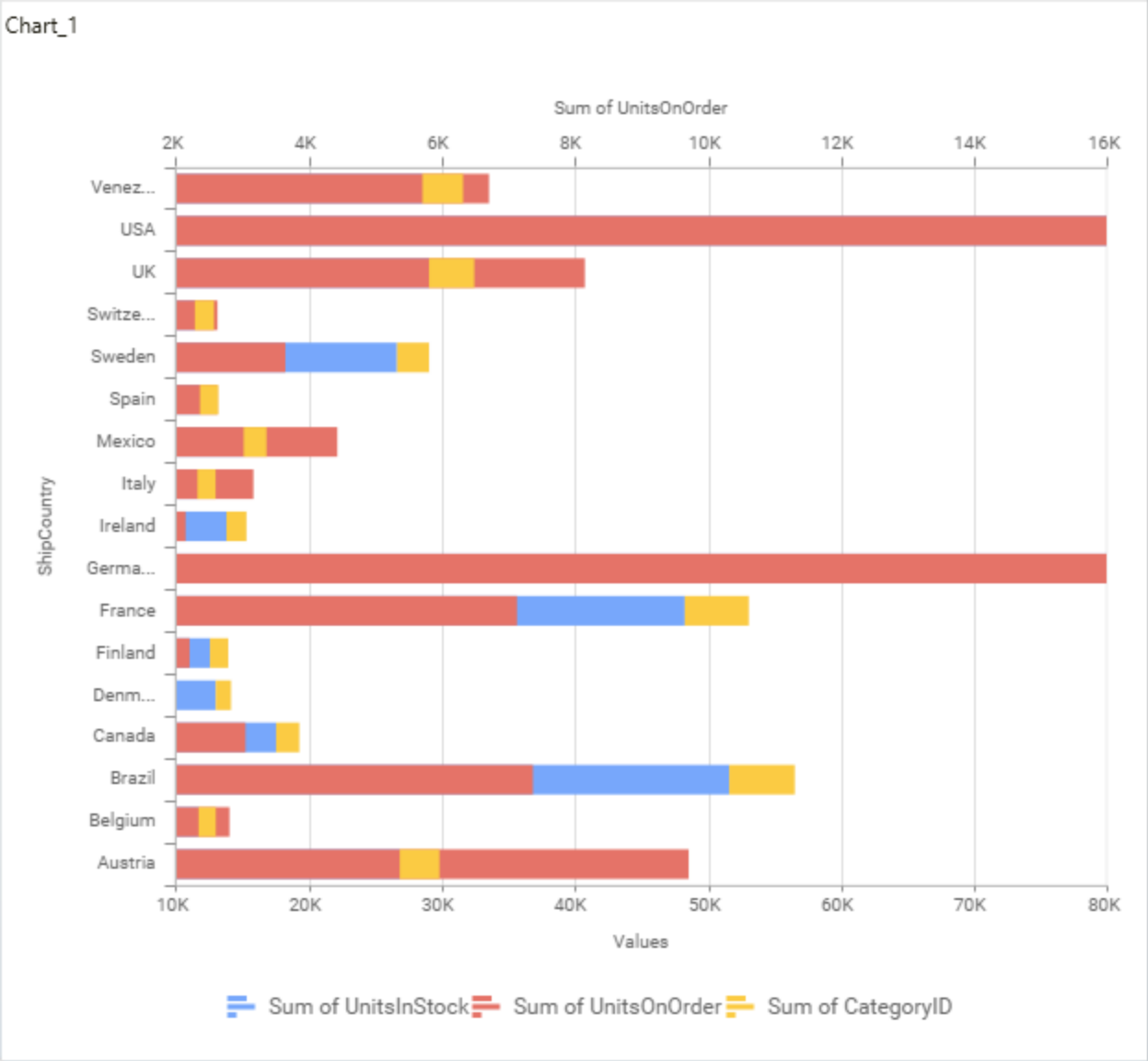


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

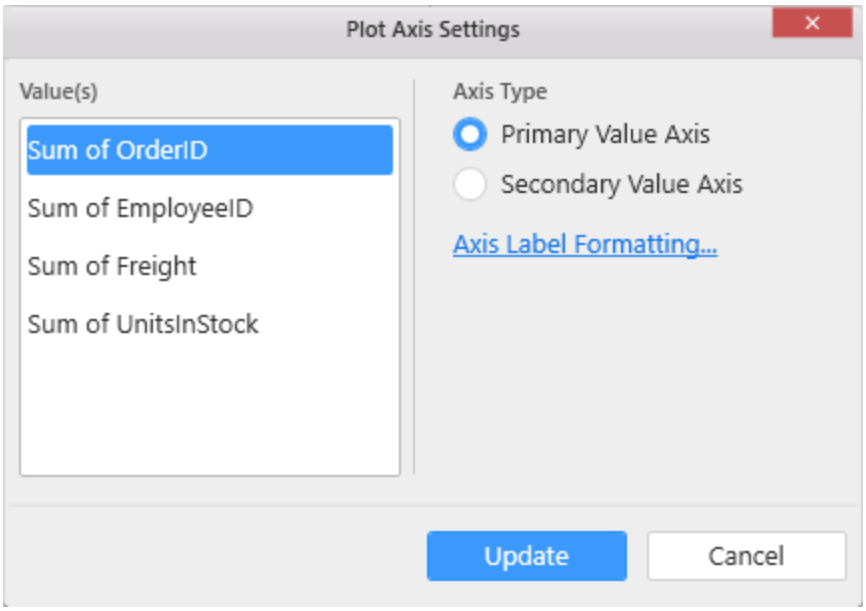


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



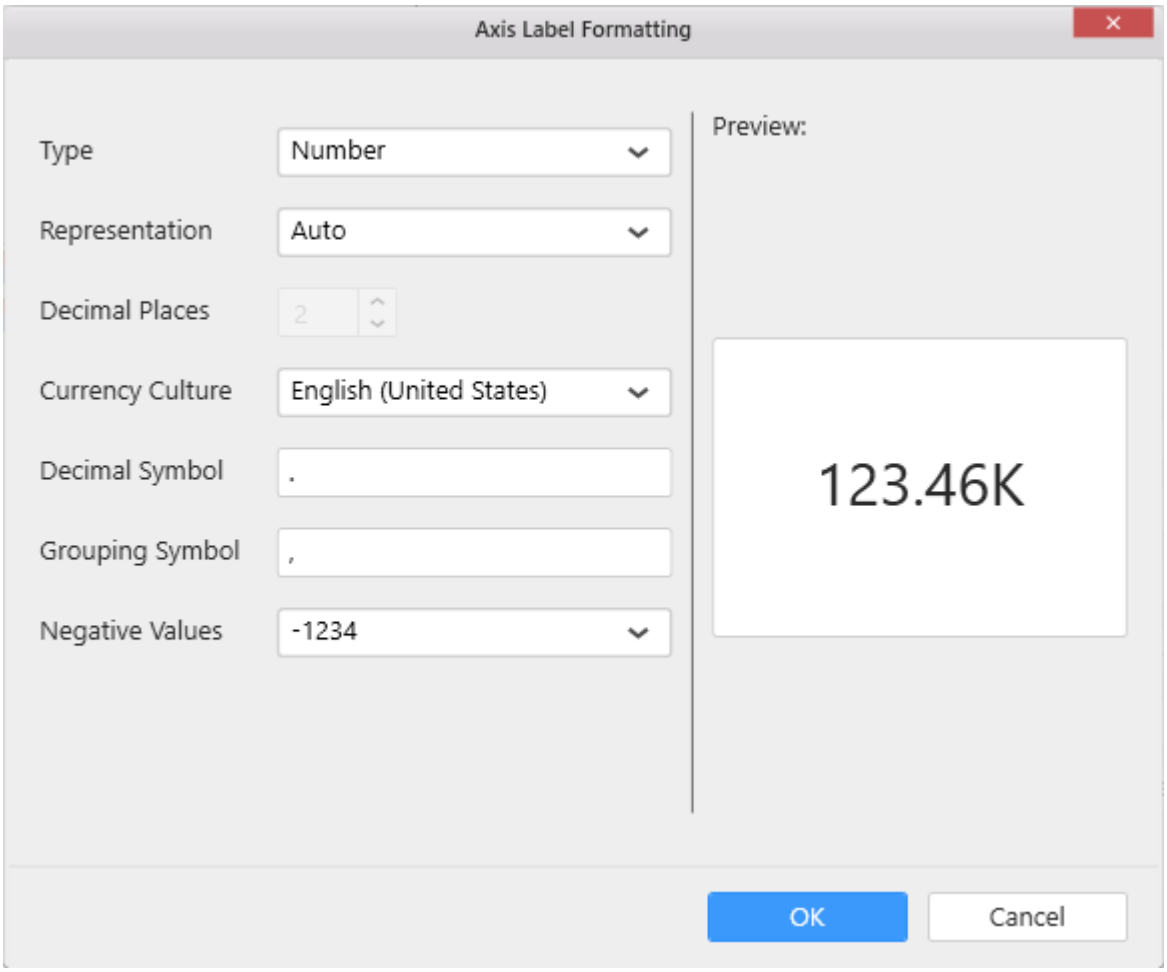
Plot Axis Settings

This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



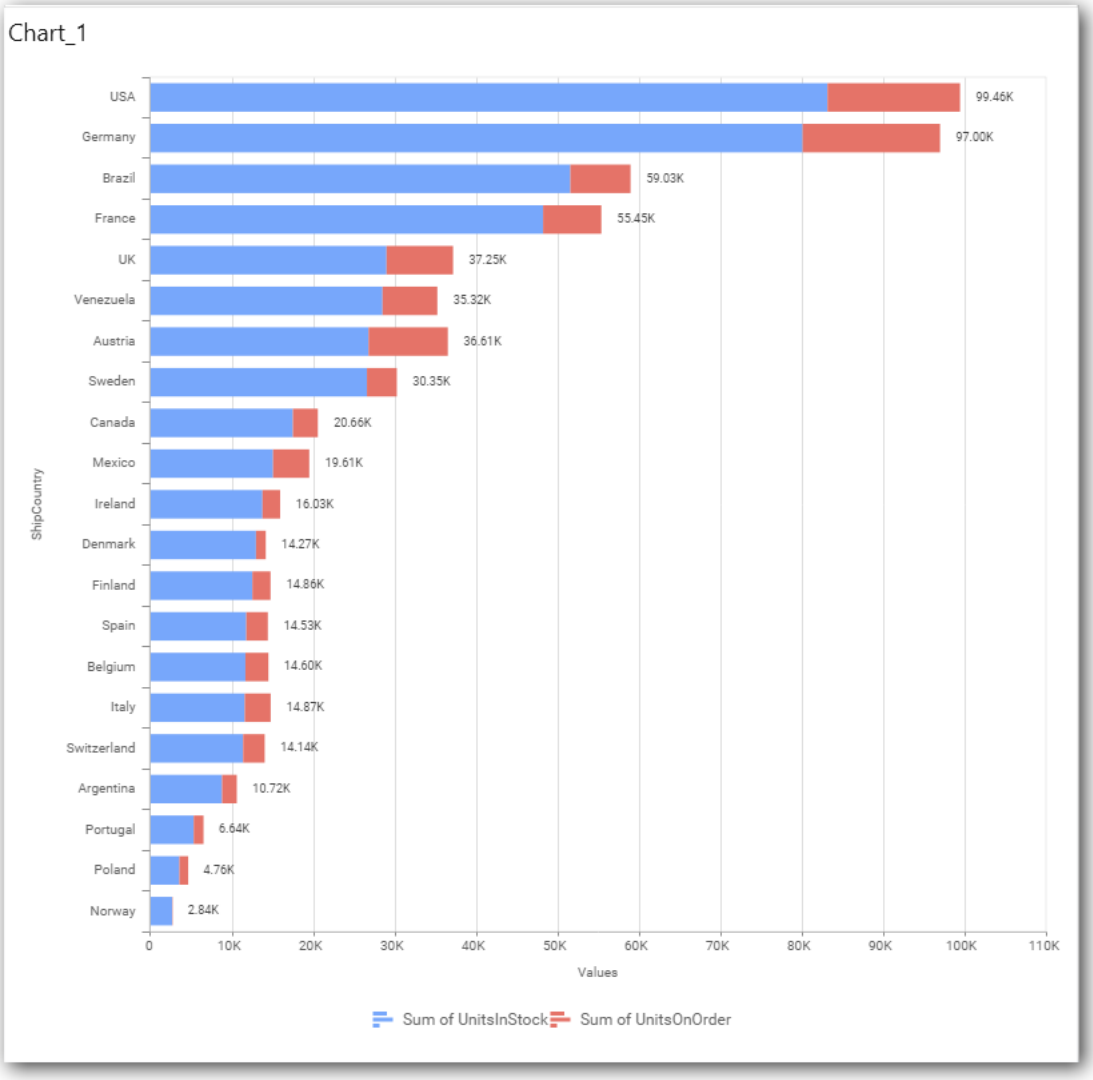
Axis Label Formatting

This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.

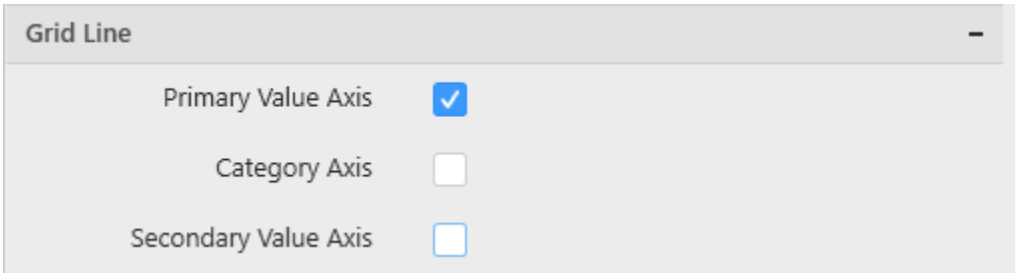


Sort Order

This allows you to define the sort order for each measure column added.



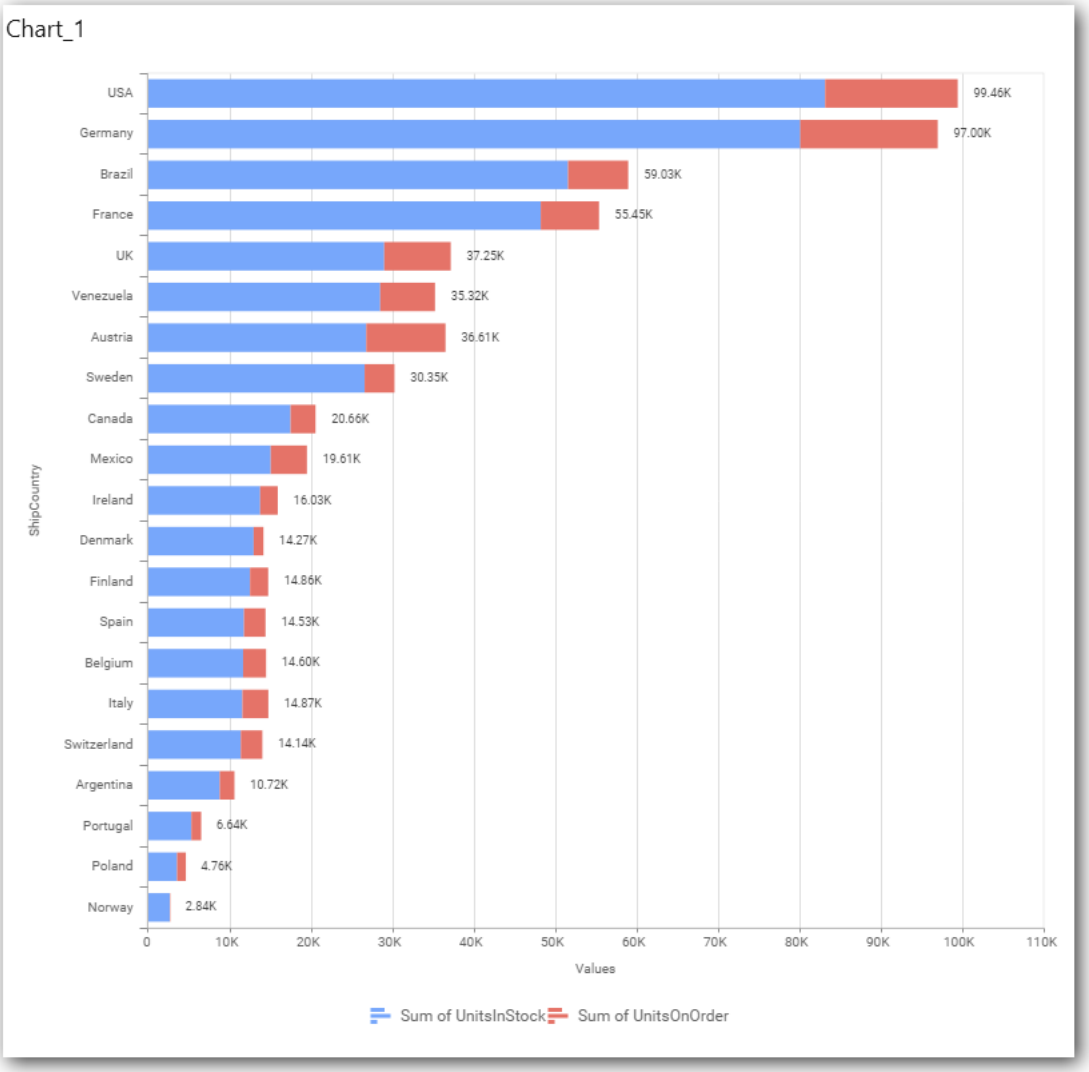
Grid Line Settings



Primary value Axis

This allows you to enable the primary value axis' gridlines for the stacked bar chart.

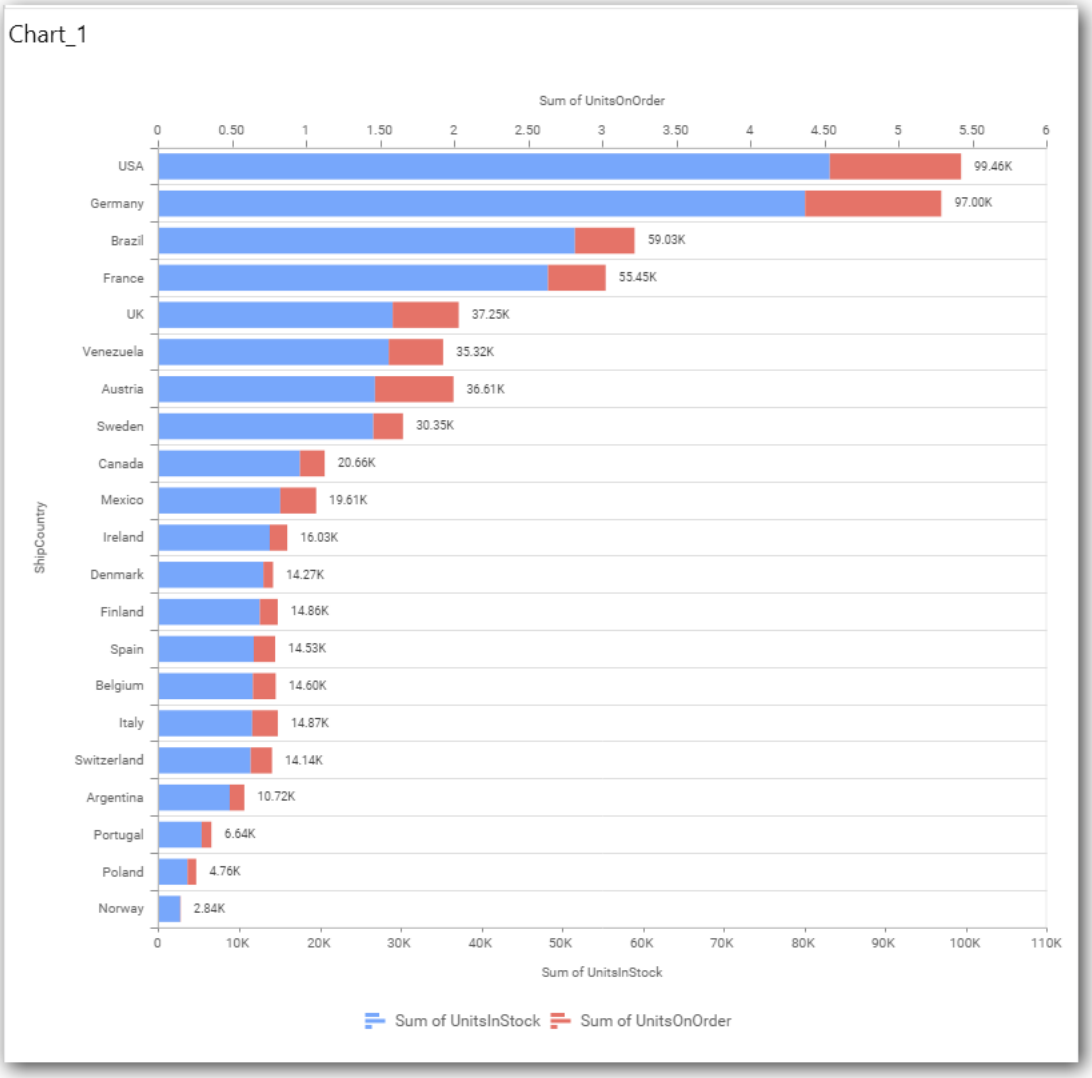
Primary Value Line



Category Axis

This allows you to define the visibility of Category axis' gridlines.

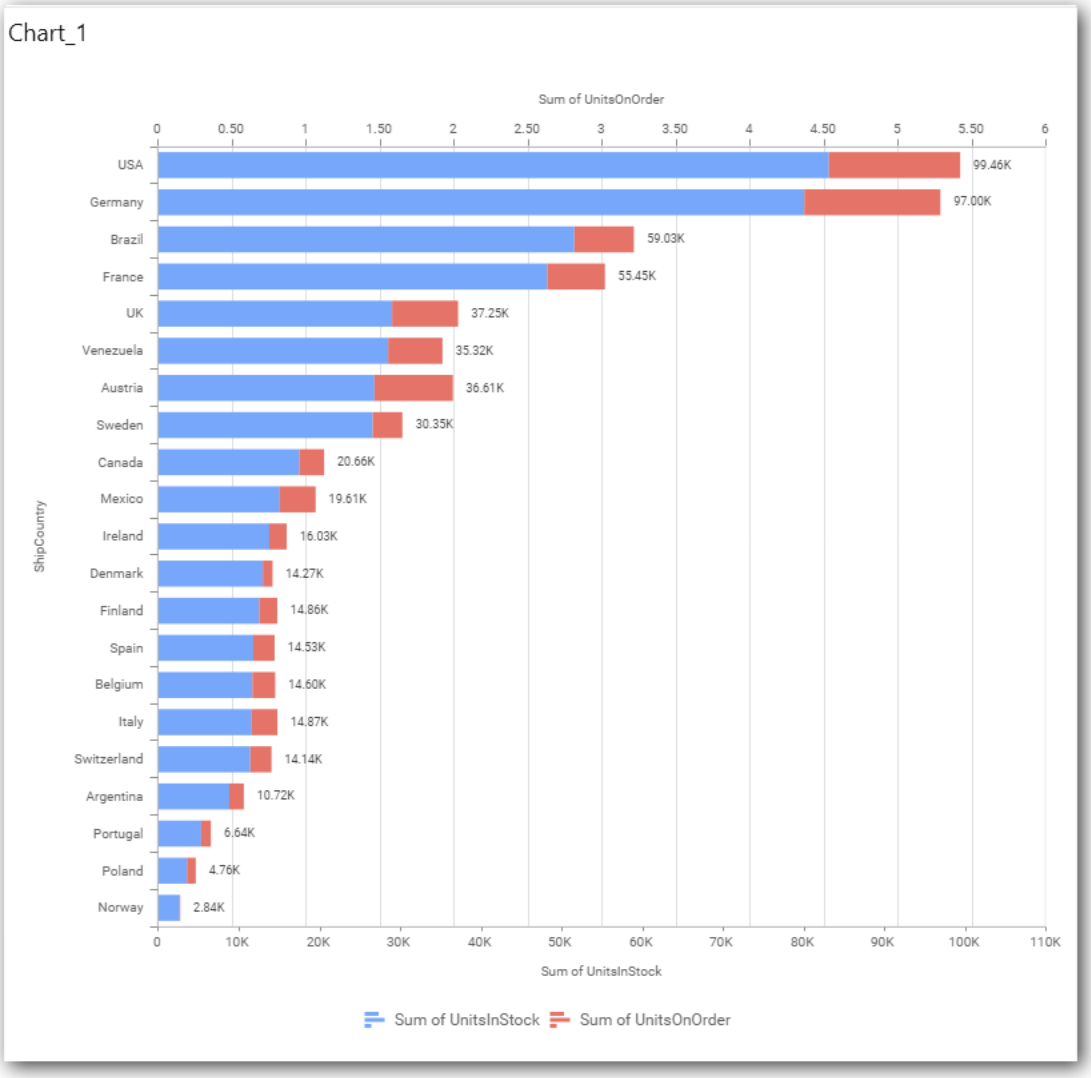
Category Value Line



Secondary Value Axis

This allows you to toggle the visibility of secondary value axis' gridlines.

Secondary Value Line

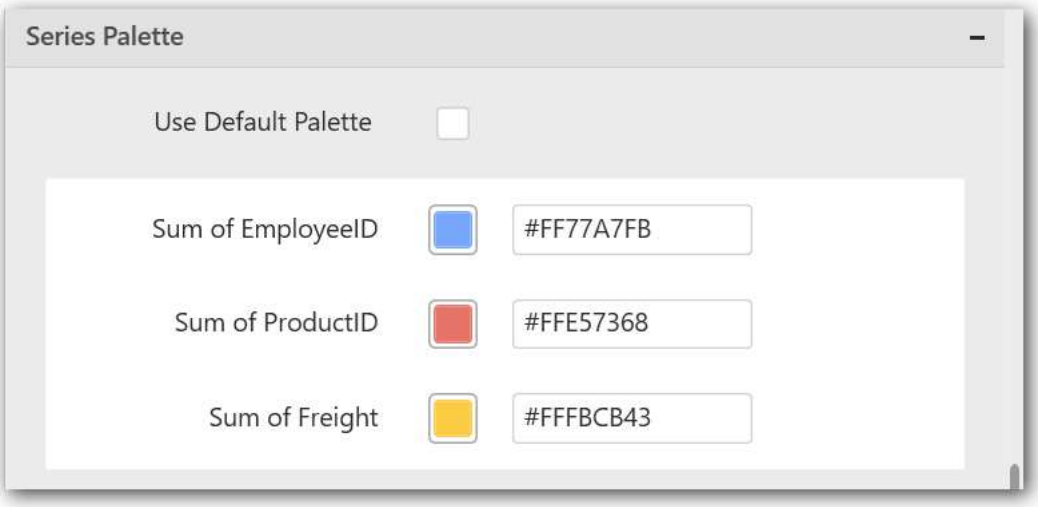


Series Palette

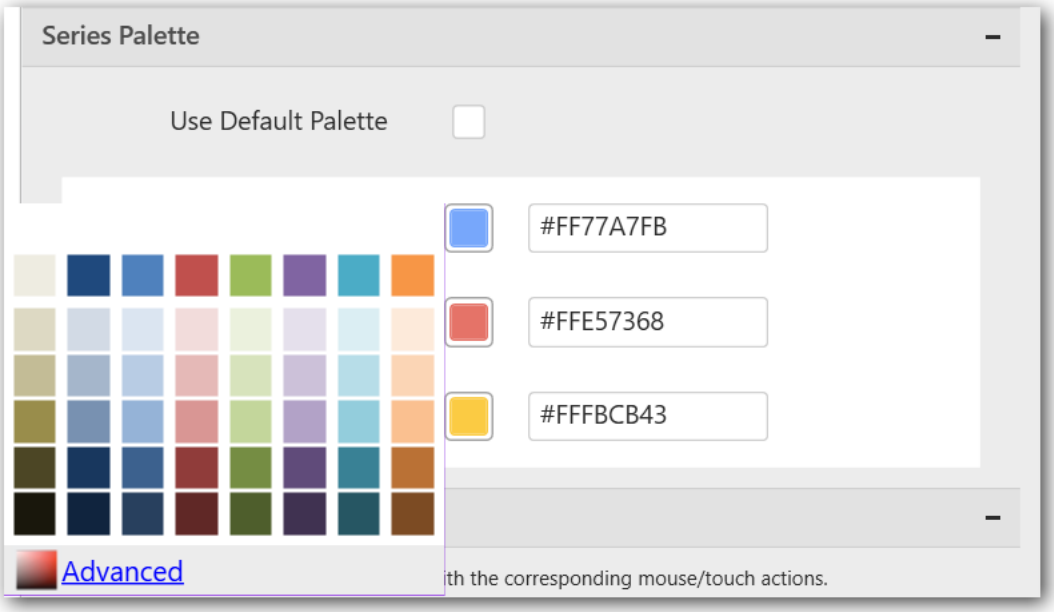
This allows you to customize the chart series color through Series Palette section.

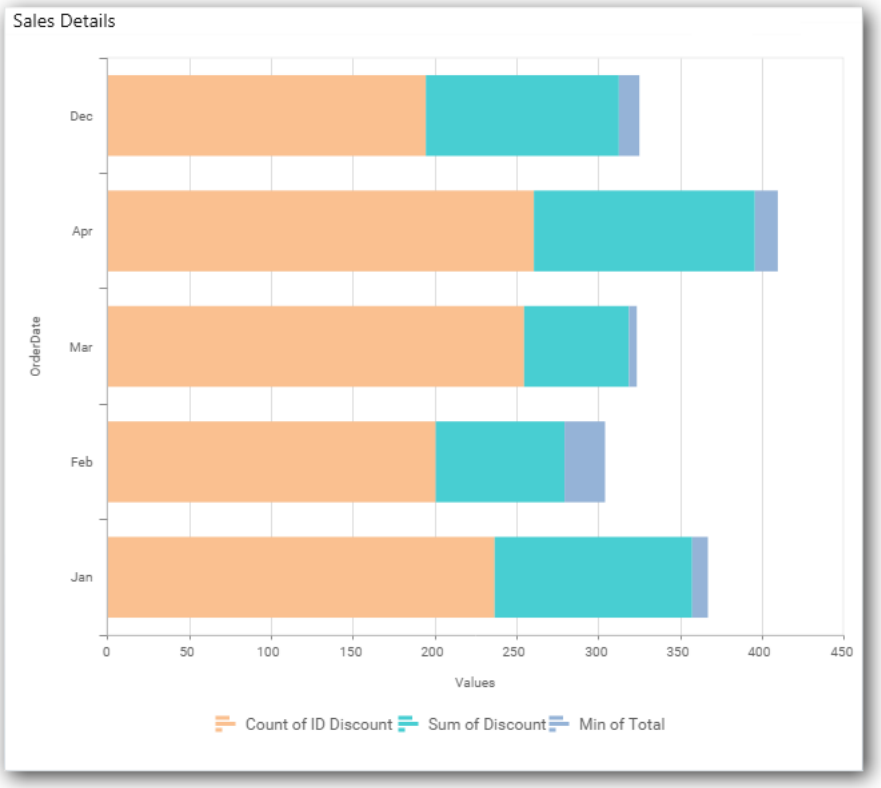
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



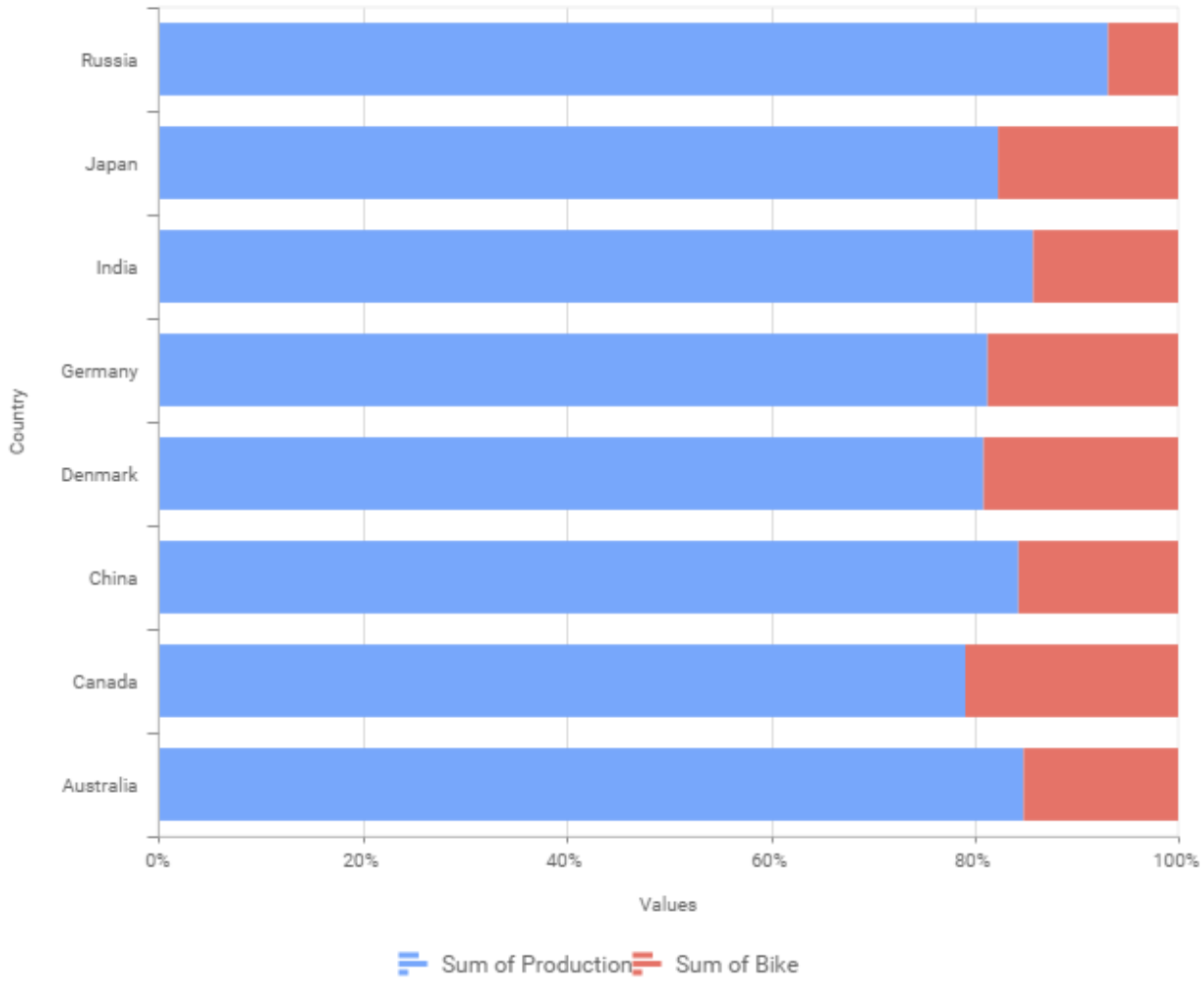
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





100% Stacked Bar Chart

100% Stacked Bar Chart allows you to compare multiple measures through bars stacked one after the other horizontally.



How to configure the flat table data to 100% Stacked Bar Chart?

100% Stacked Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

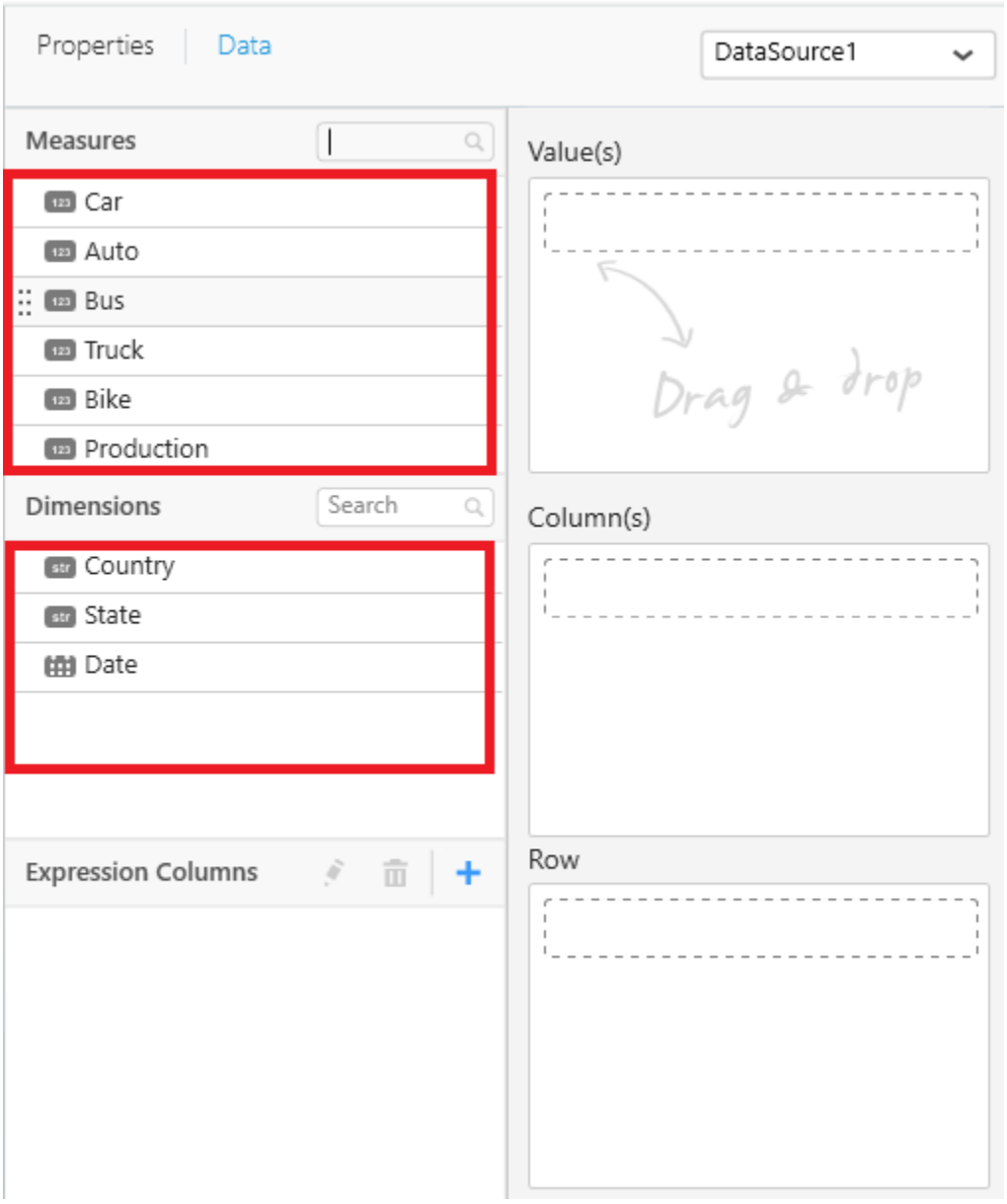
Follow the steps to configure data to 100% stacked bar chart.

Drag and drop the 100% stacked bar chart into canvas and resize it your required size.

Connect to data source.

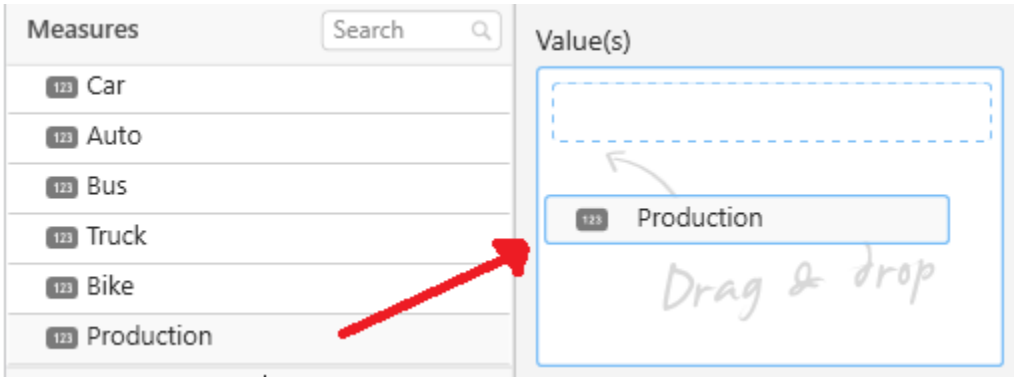
Focus on the 100% stacked bar chart and click on Assign Data.

The data pane will be opened with available Measures and Dimensions from the connected data source.

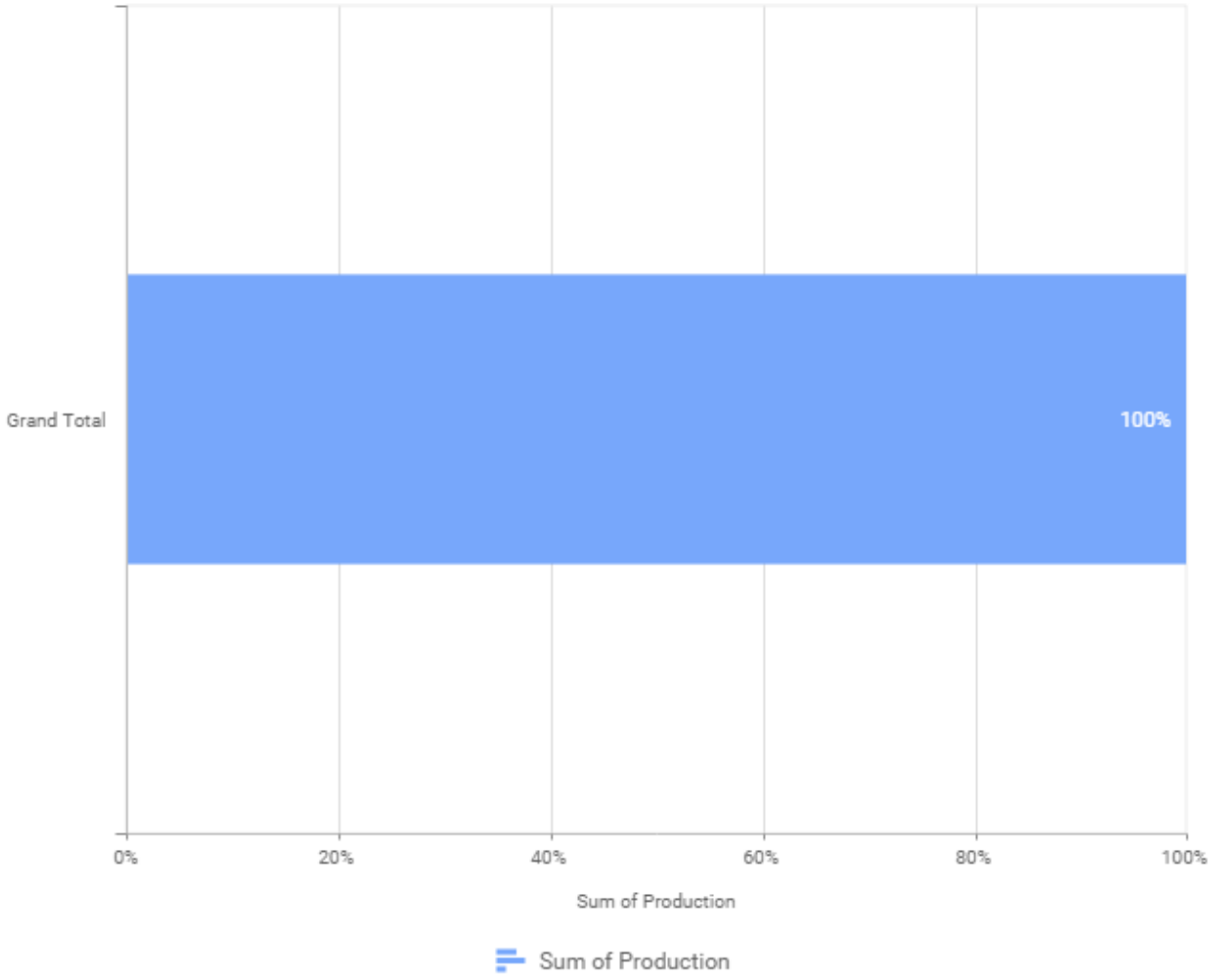


Assigning Value(s)

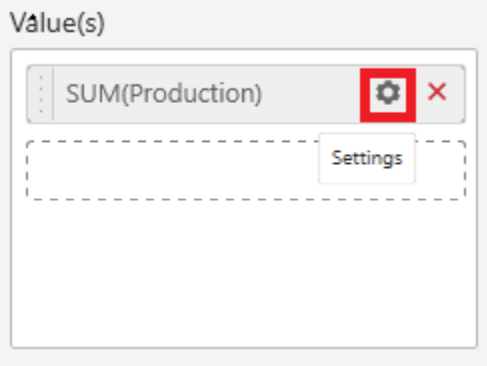
Drag and drop the Measure into Value.



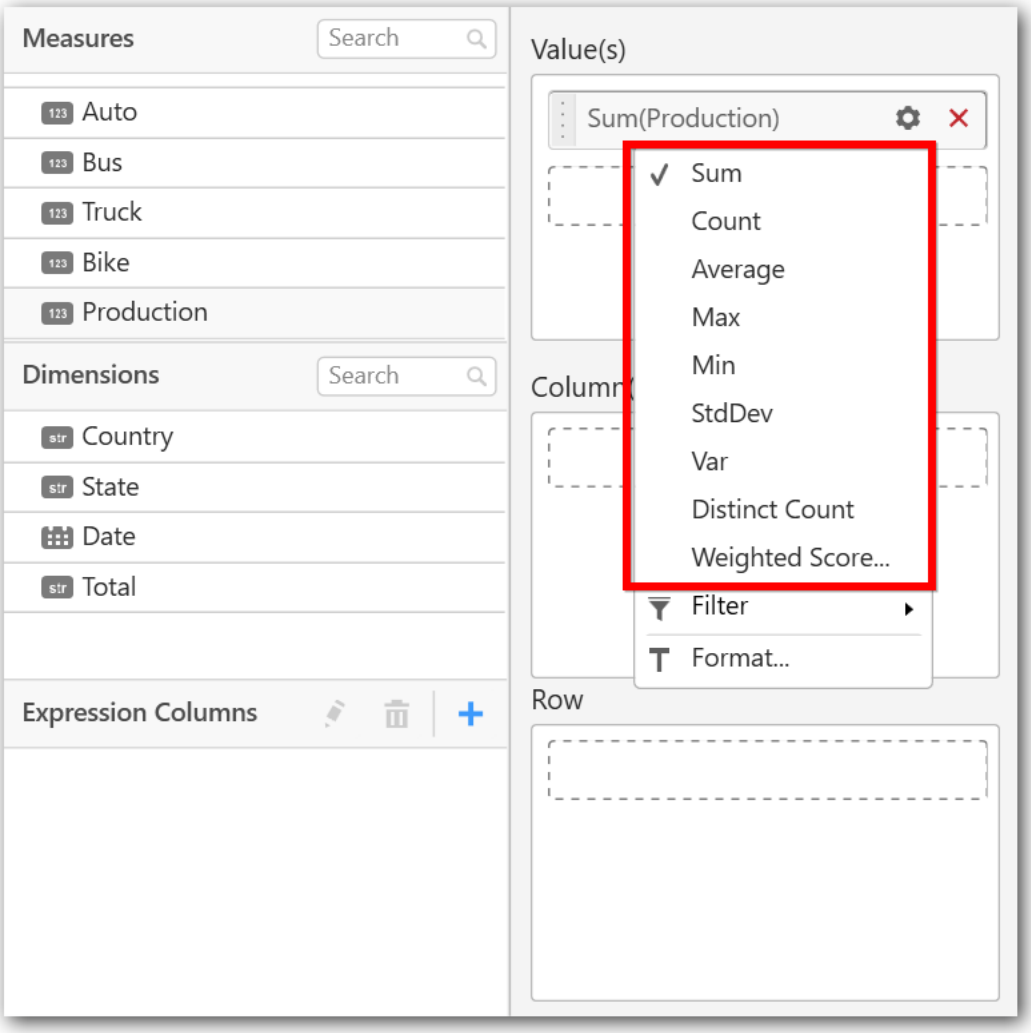
Now the chart will be rendered like this.



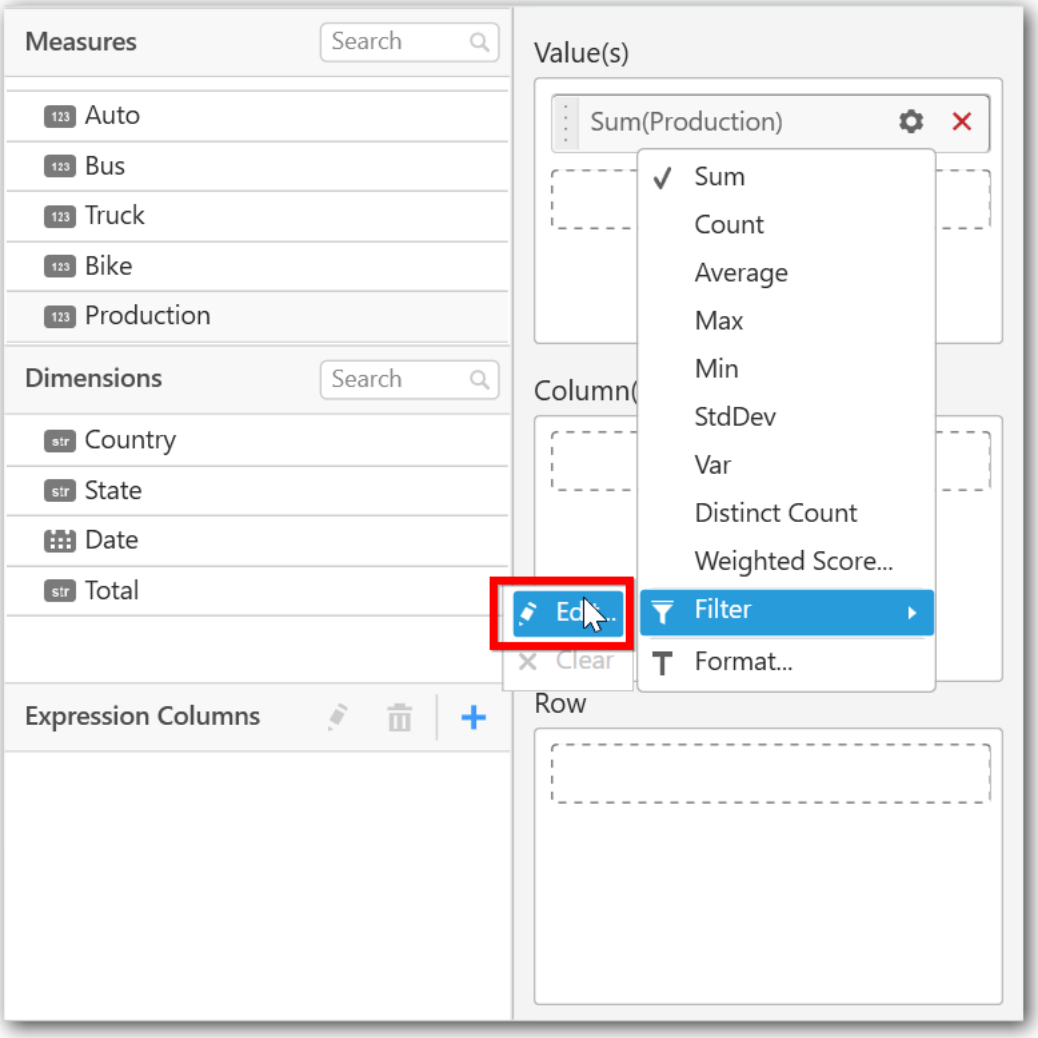
You can change the summary type of the value by clicking on Settings option.



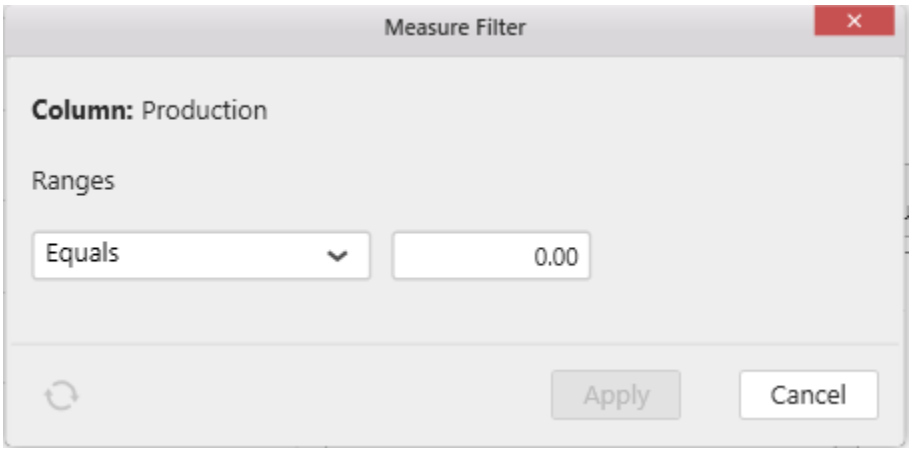
Select the required summary type from list.

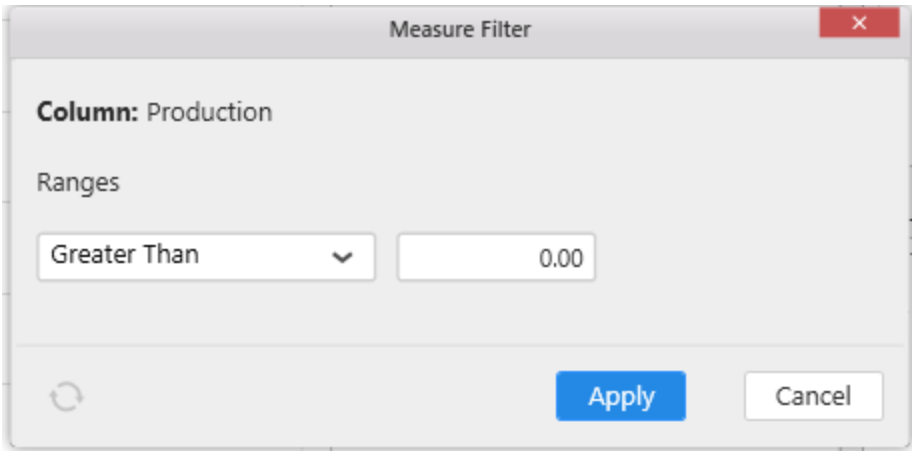


You can select what data to be displayed by choosing filter option.

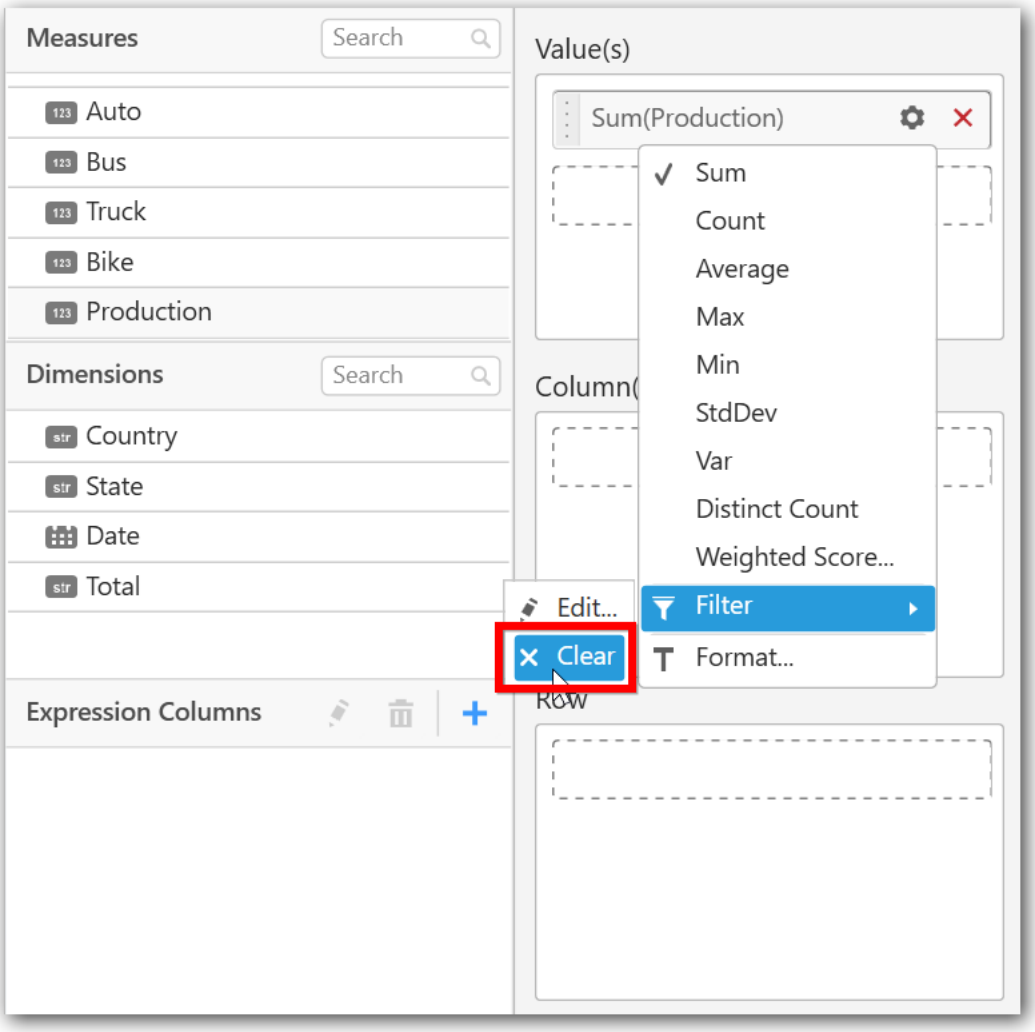


The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.

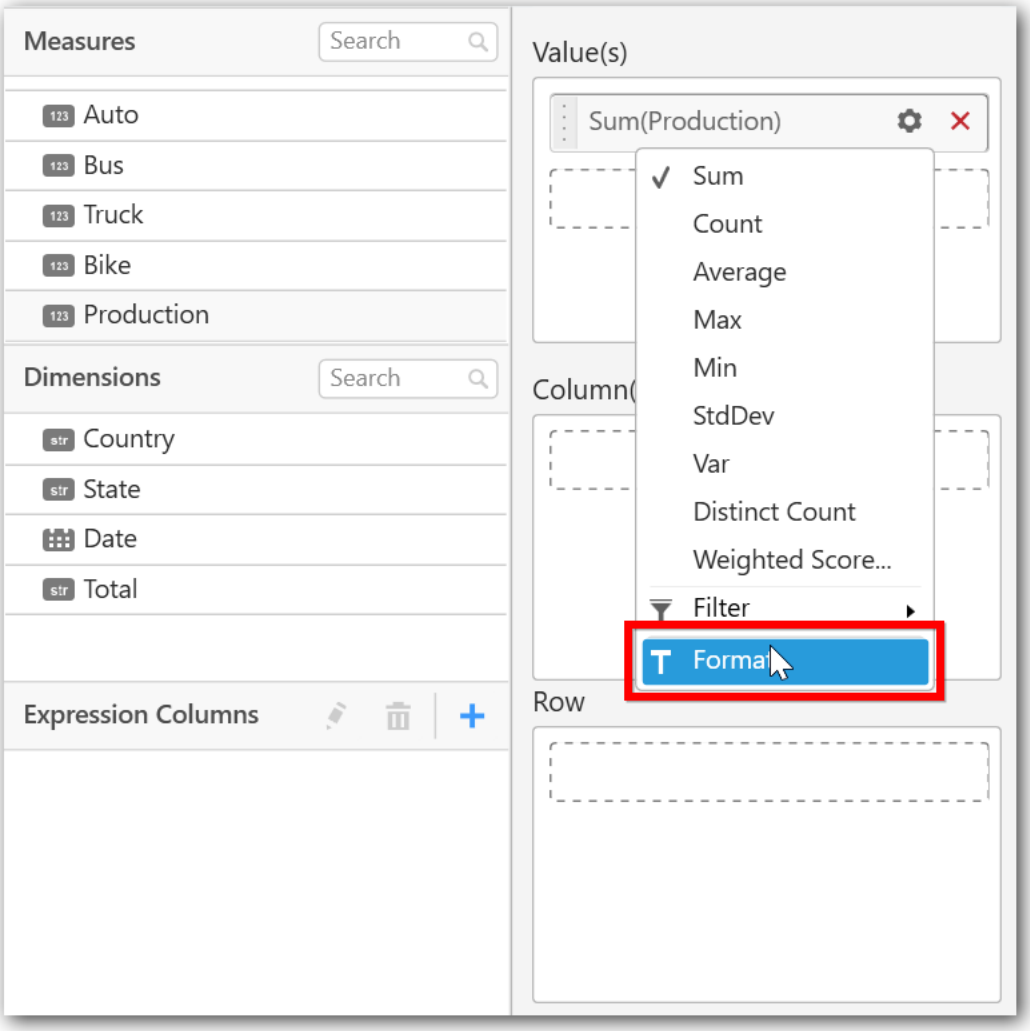




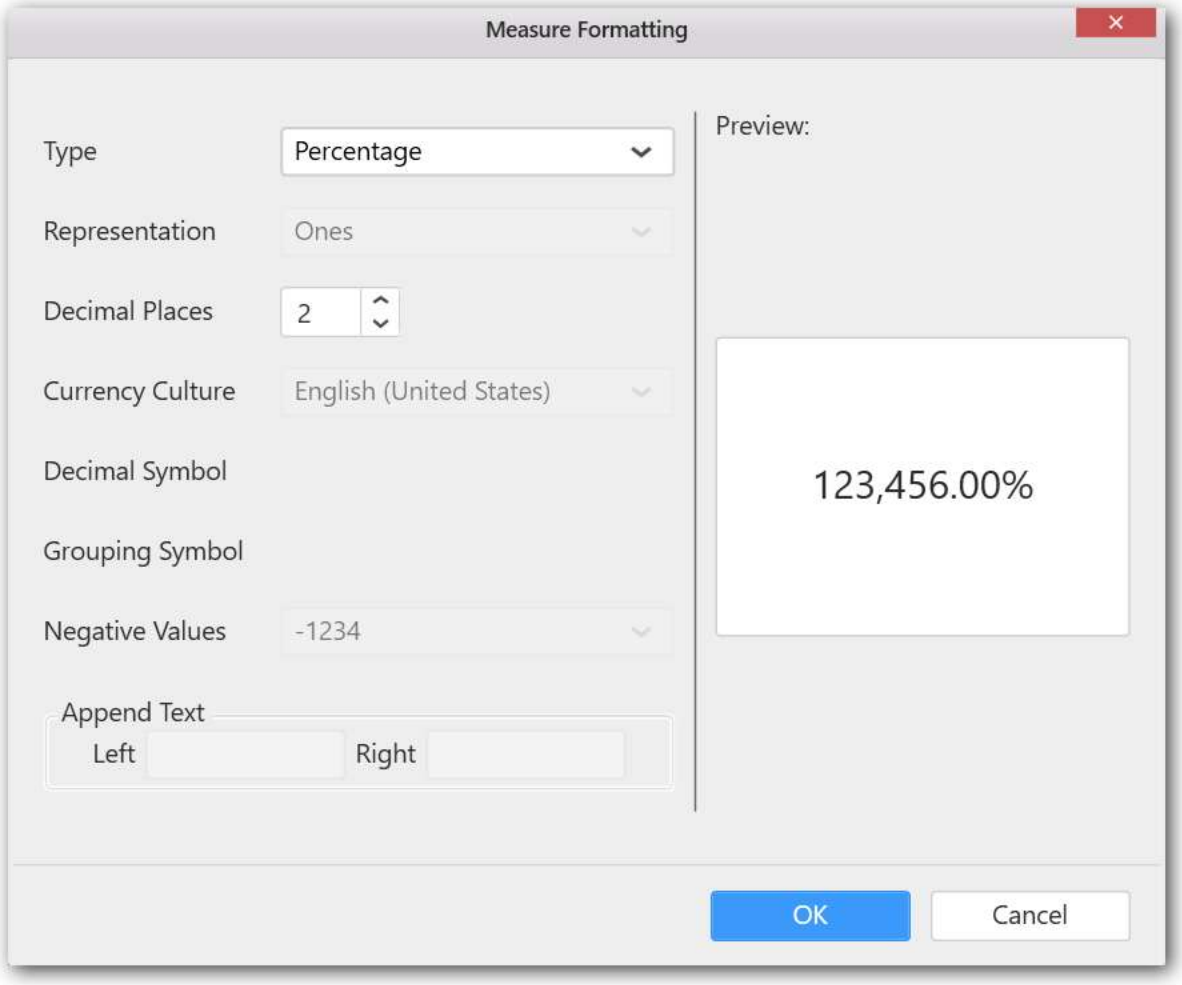
You can clear the filter.



You can Format the value.

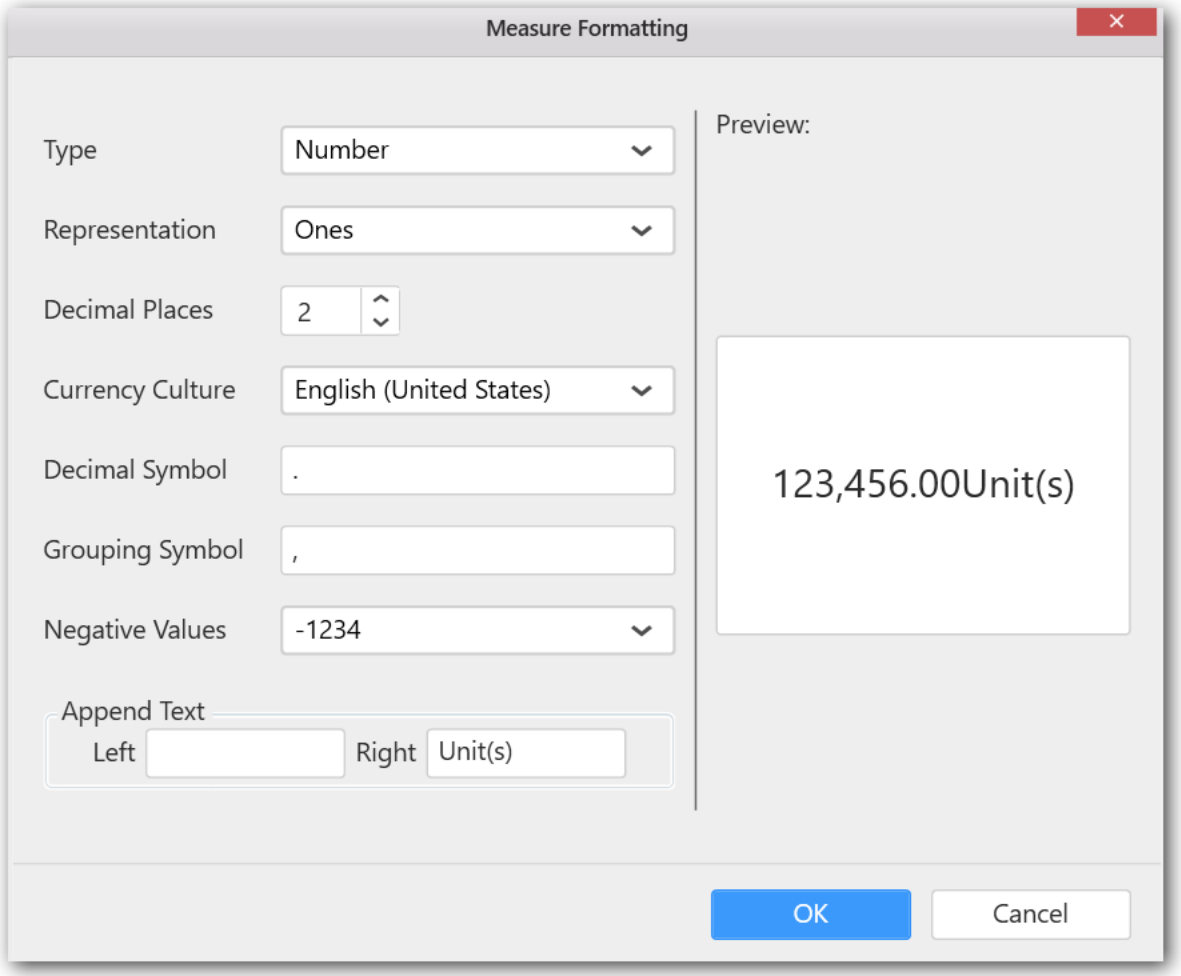


The format options will be shown.

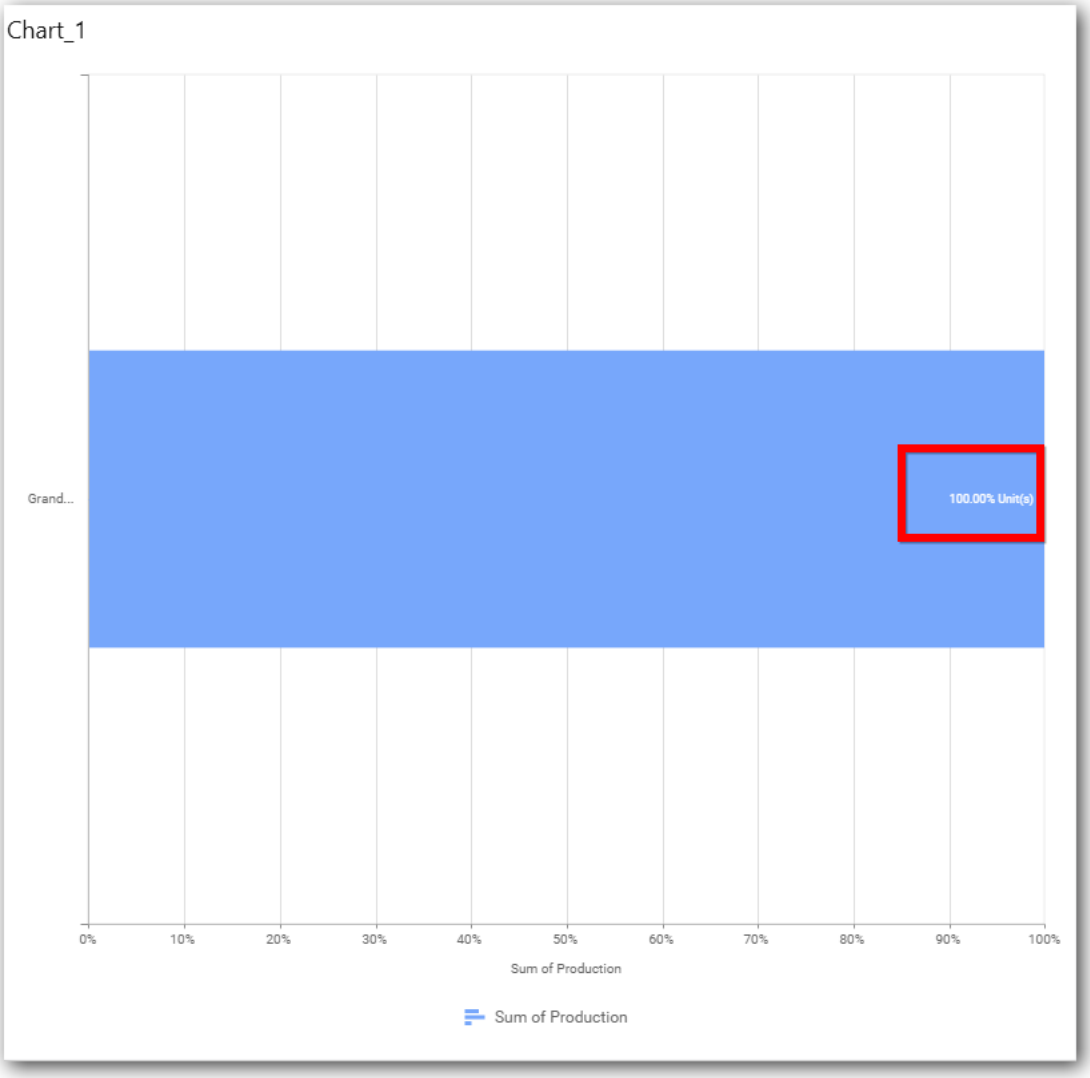


Note: For 100% Stacked Charts, default format type is Percentage.

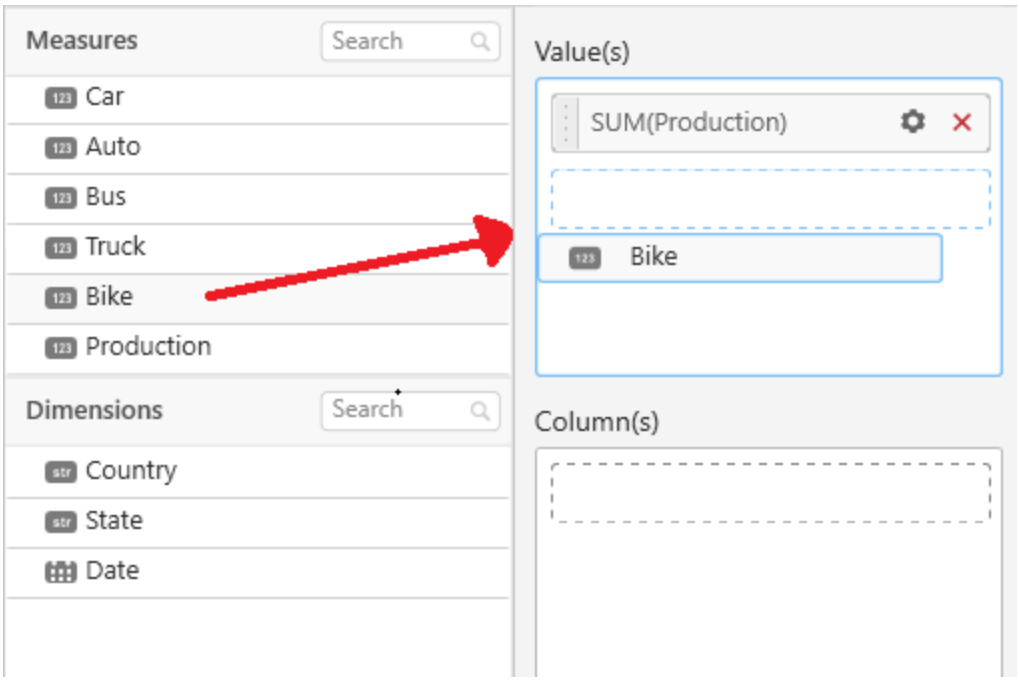
Choose the options you need and click OK.

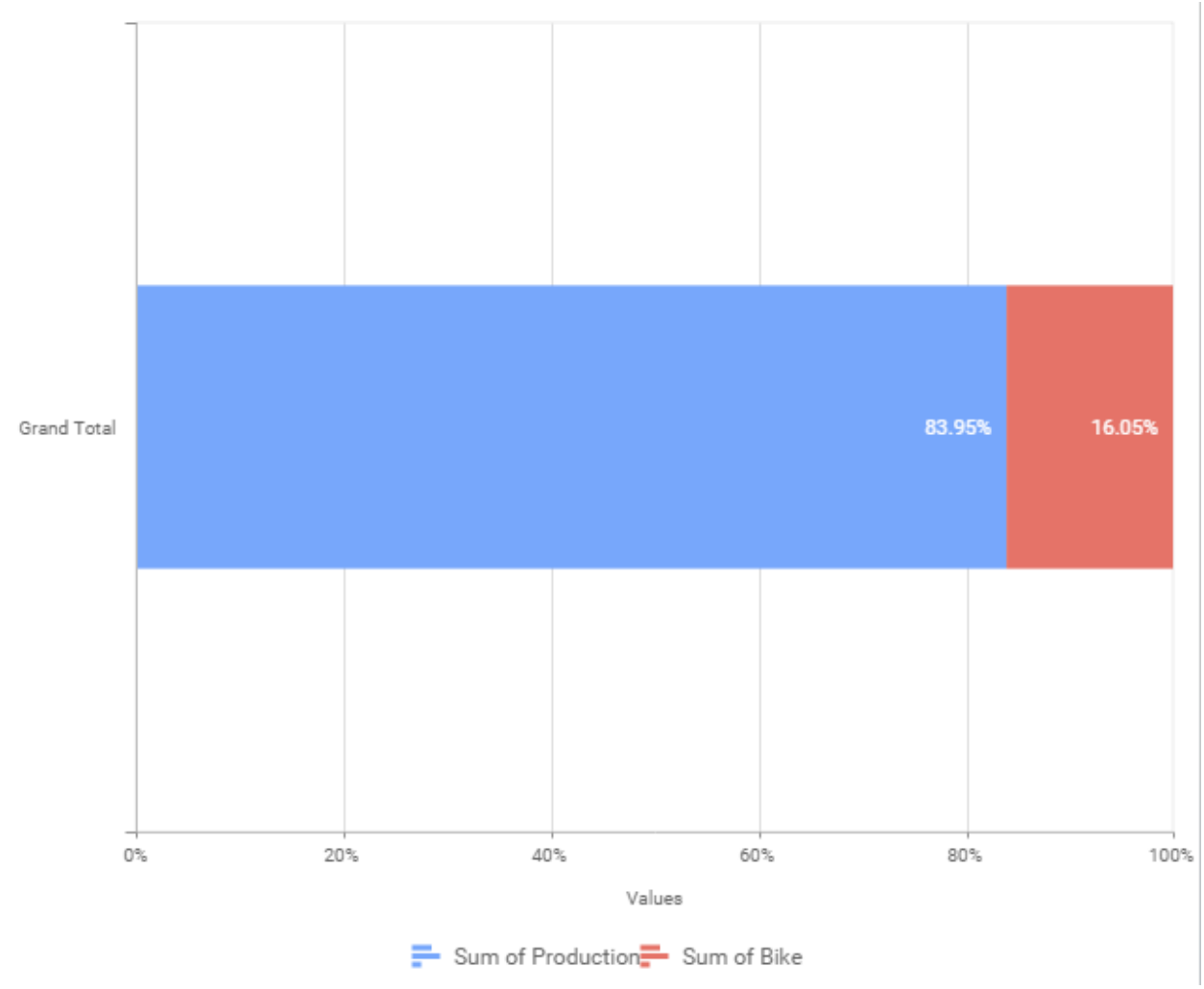


Now the Chart will be rendered like this.



You can add more number values by drag drop the Measures into Value field.

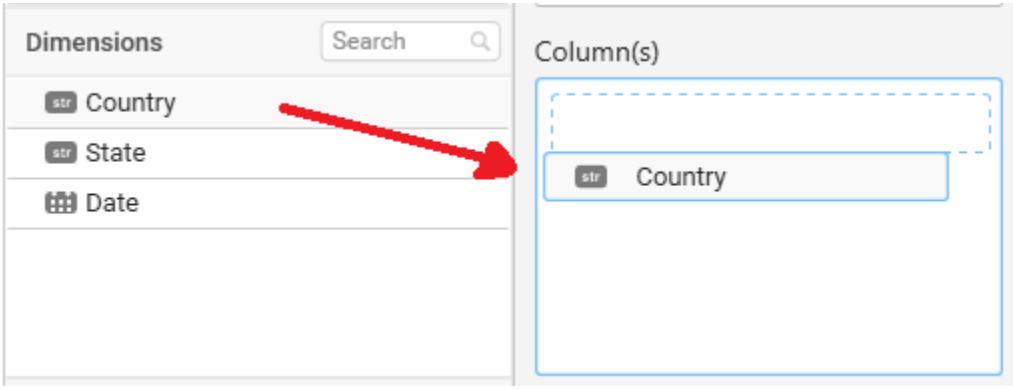


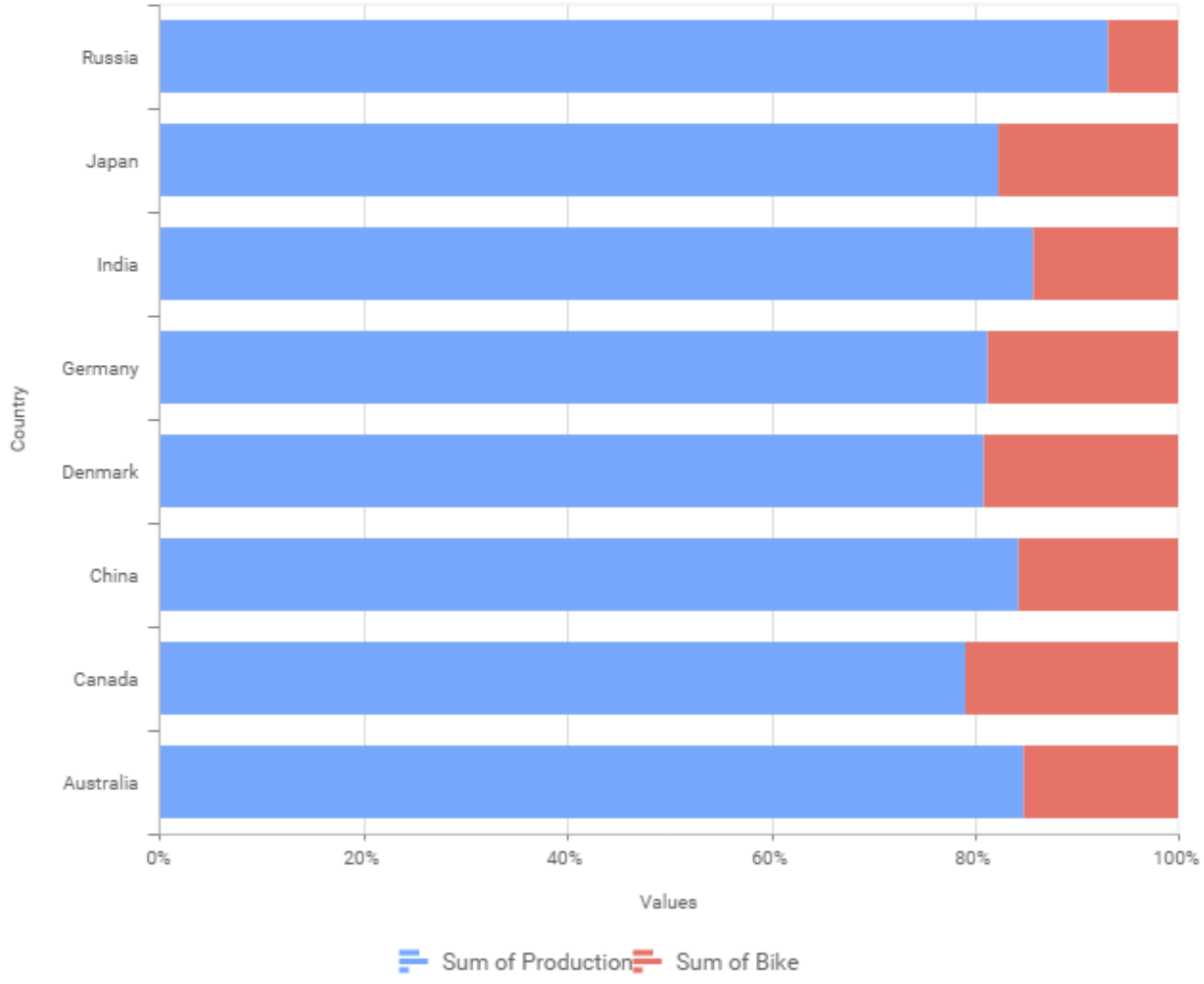


You can also add Dimensions and Columns to Value(s).

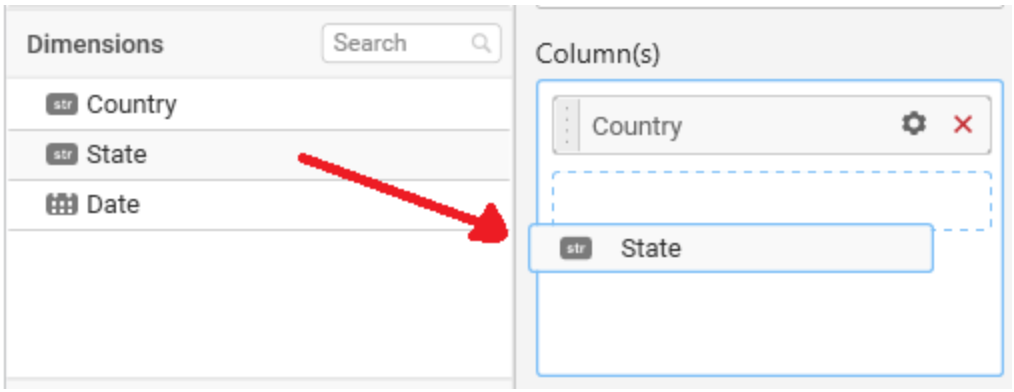
Assigning Column(s)

You can add the Dimension into Columns section by drag drop.

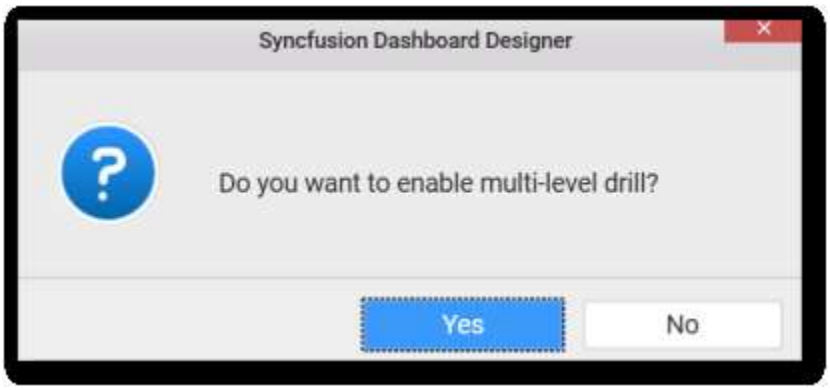




You have option to add more than one Column Value.

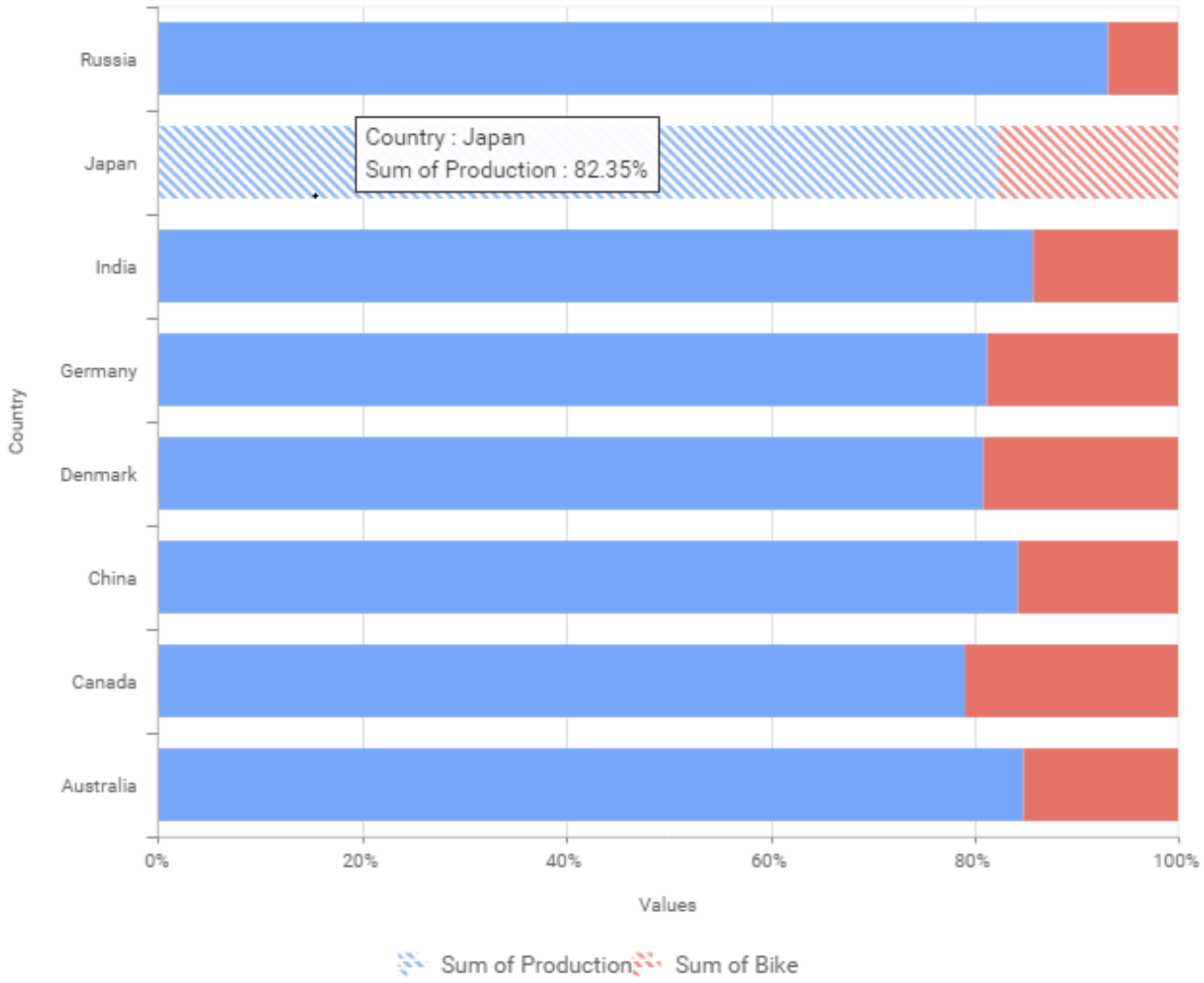


The following alert message will be shown.

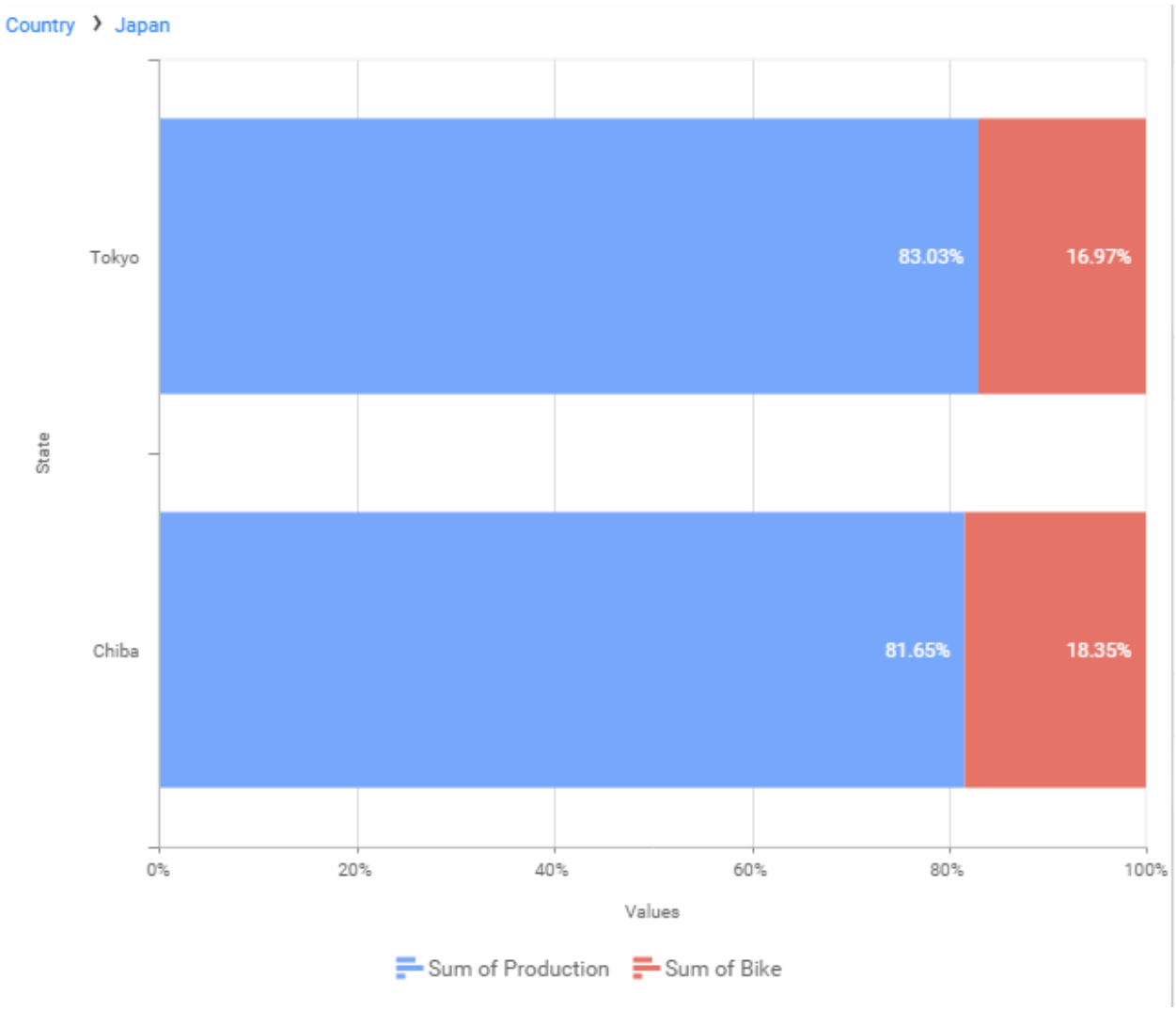


- If you choose Yes Drill down option will be enabled.

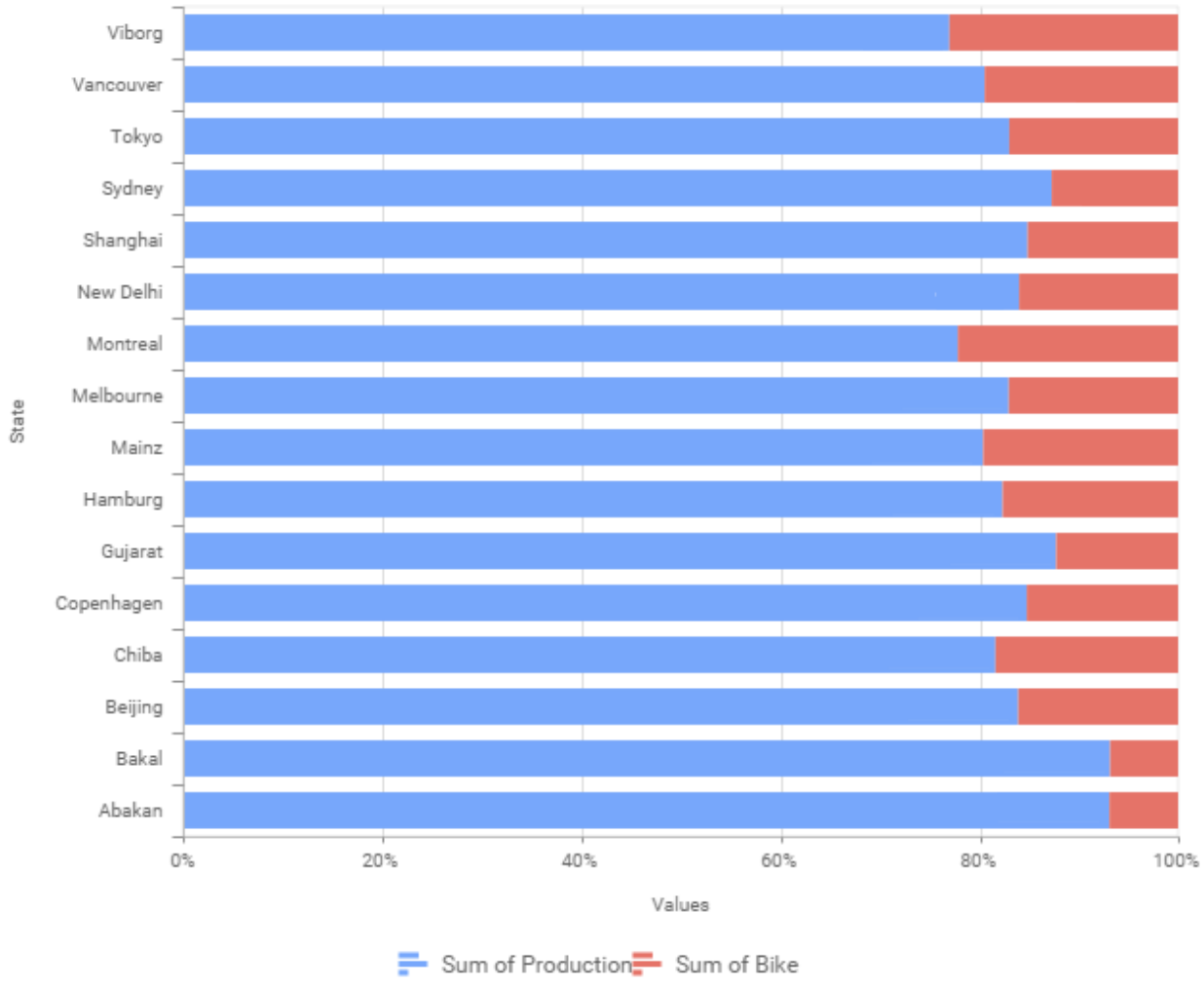
You can drill down the chart by clicking on the chart.



The drilled view of the chart is follows.

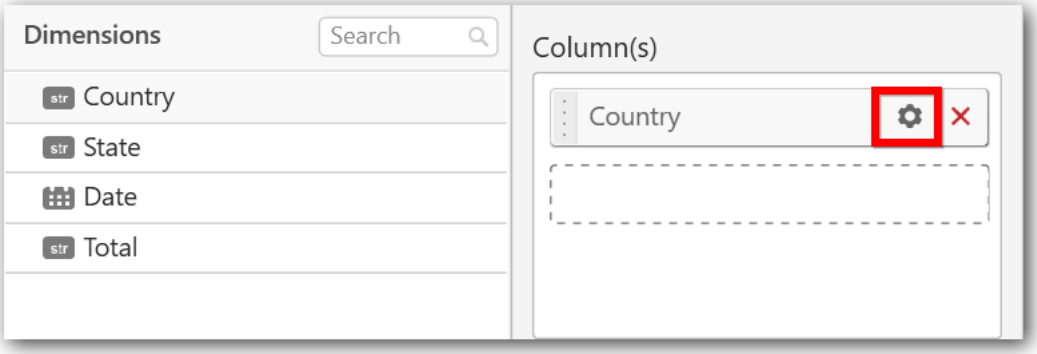


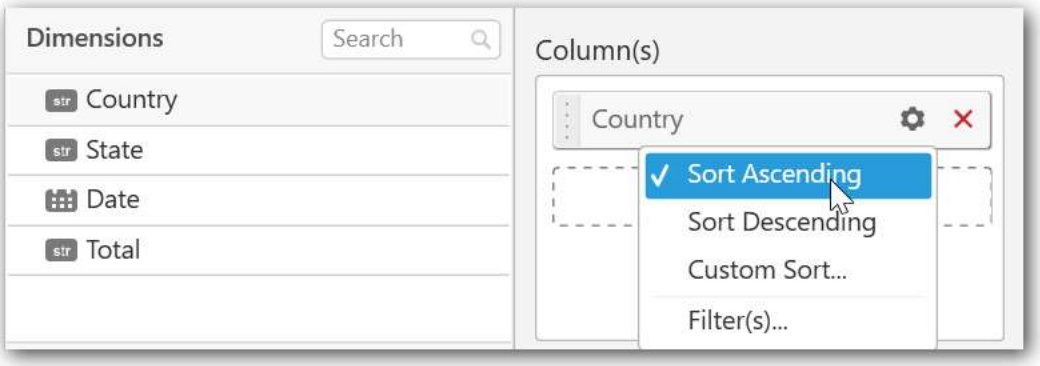
- If you click **No** the new Dimension value will replace old value.



You can also add Measures and Expression columns into Column(s) field.

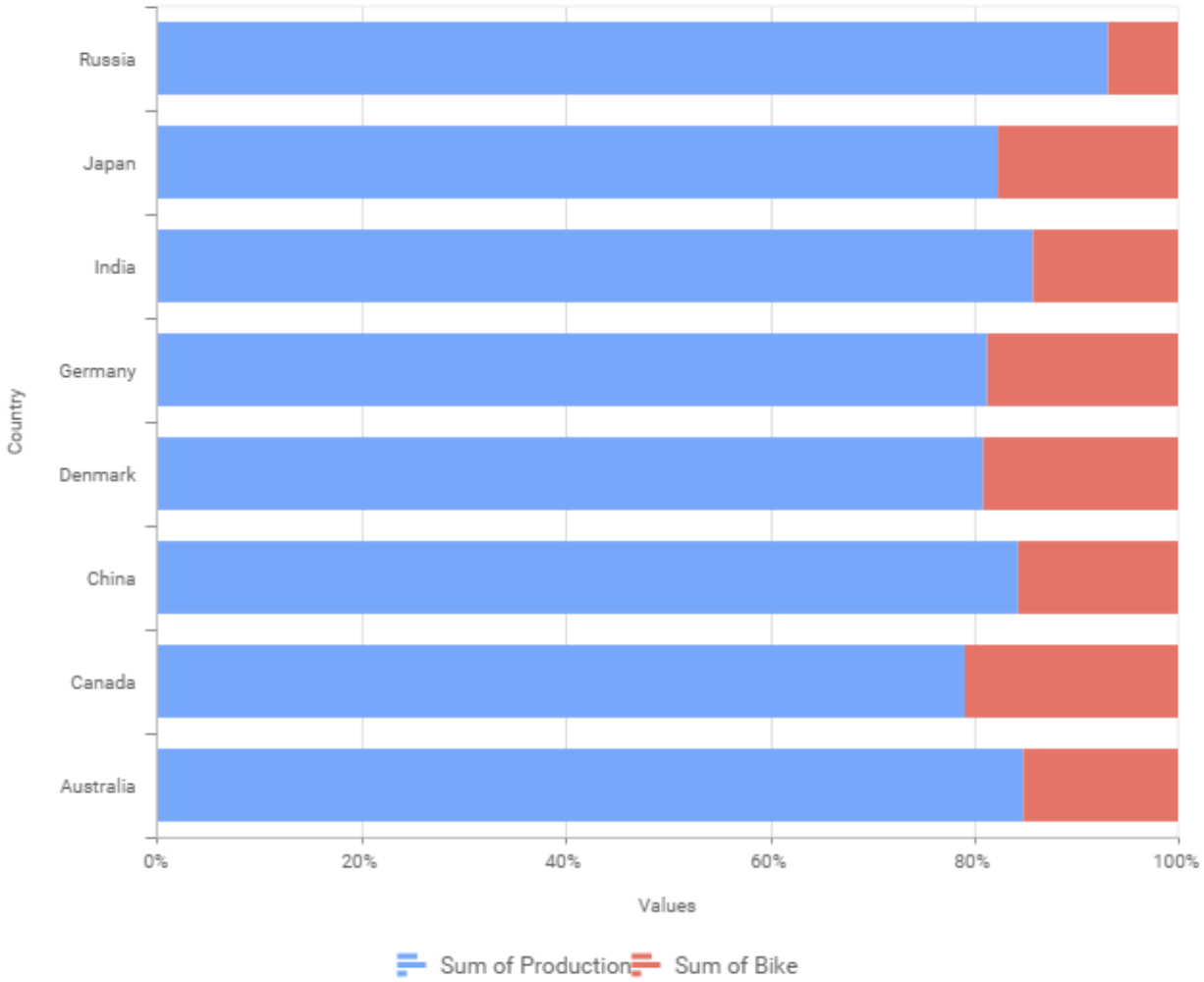
You have options to change the settings.



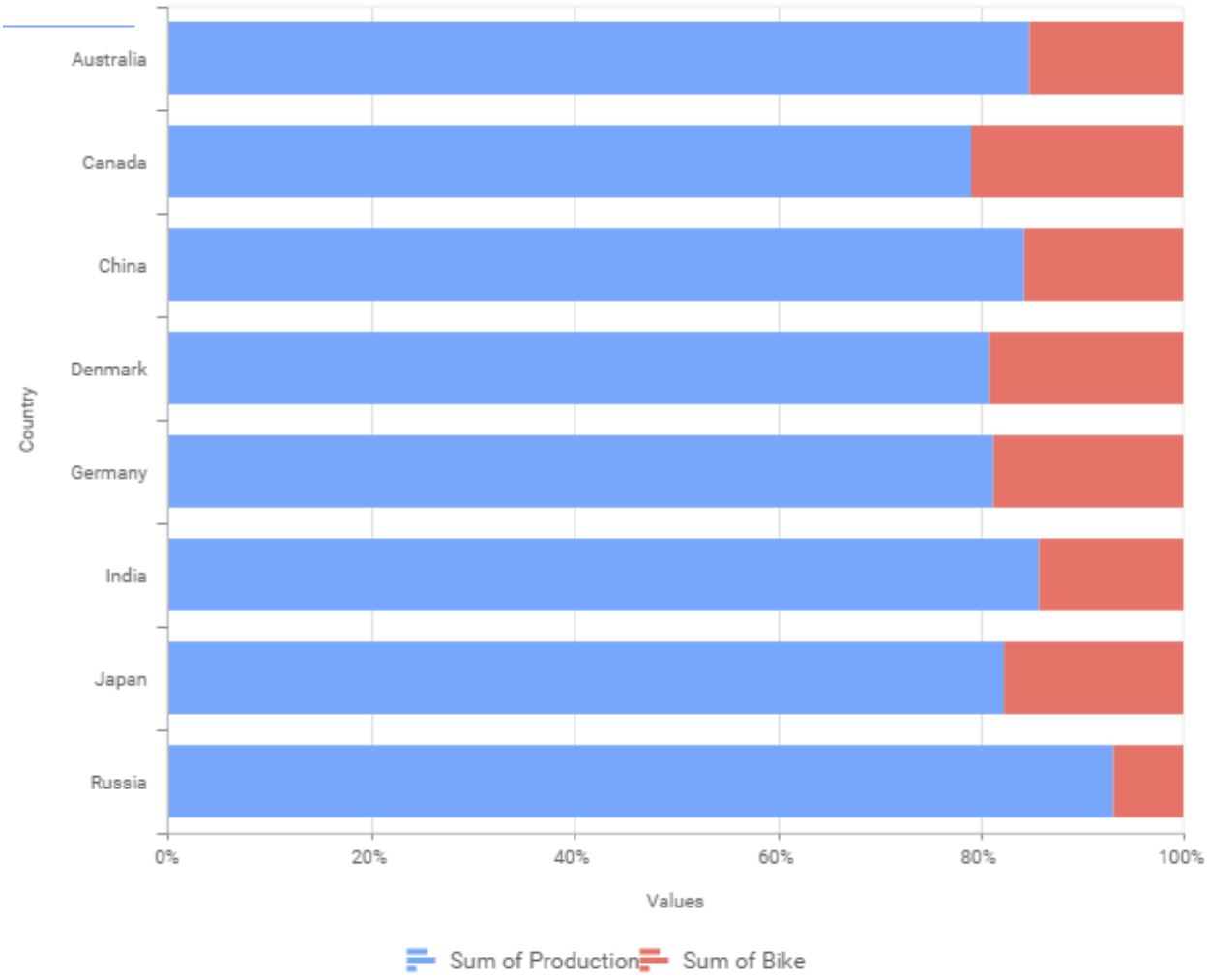


You can sort the chart either in Ascending or Descending series.

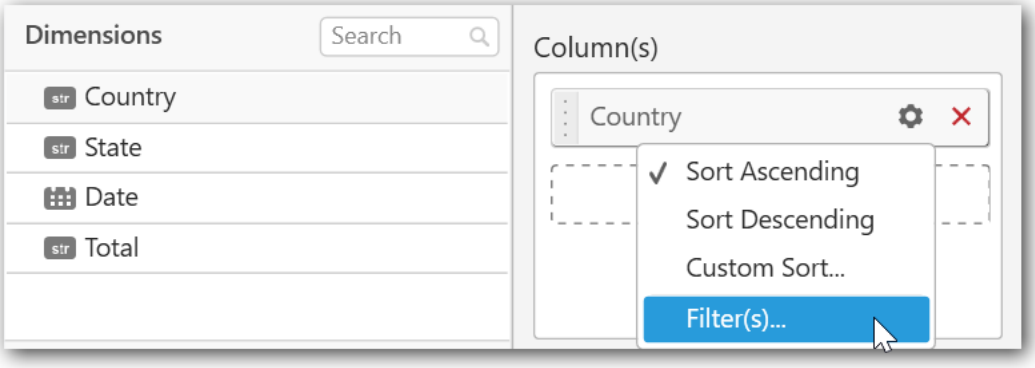
Ascending Order

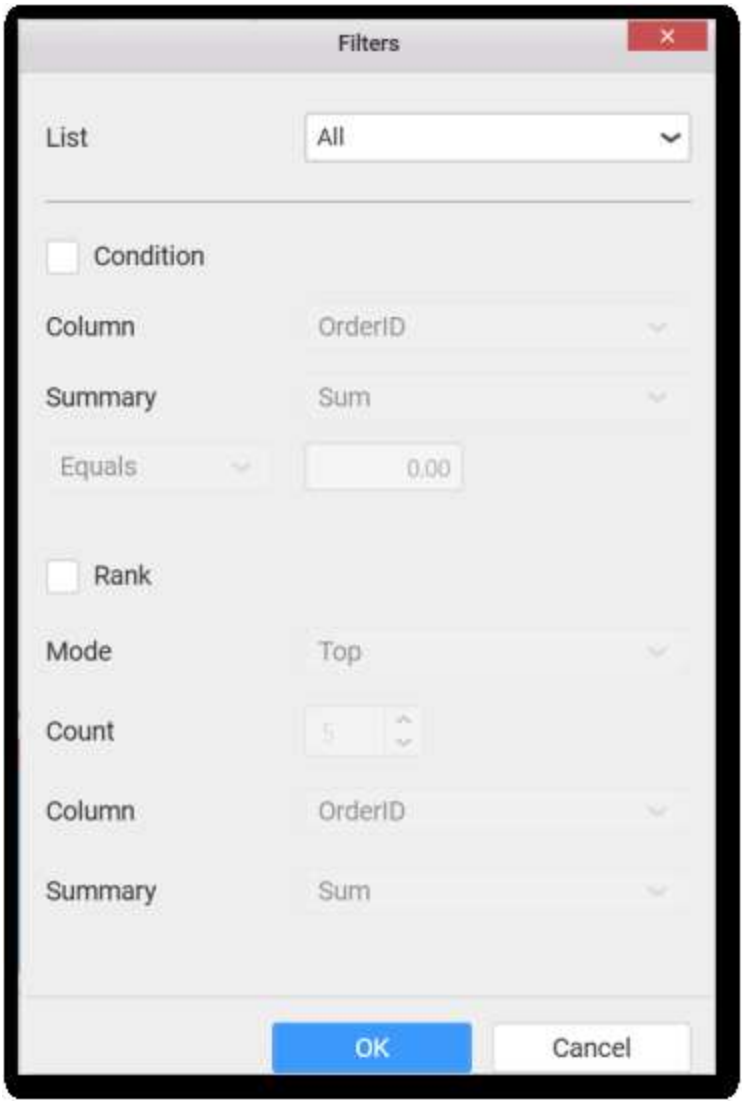


Descending order

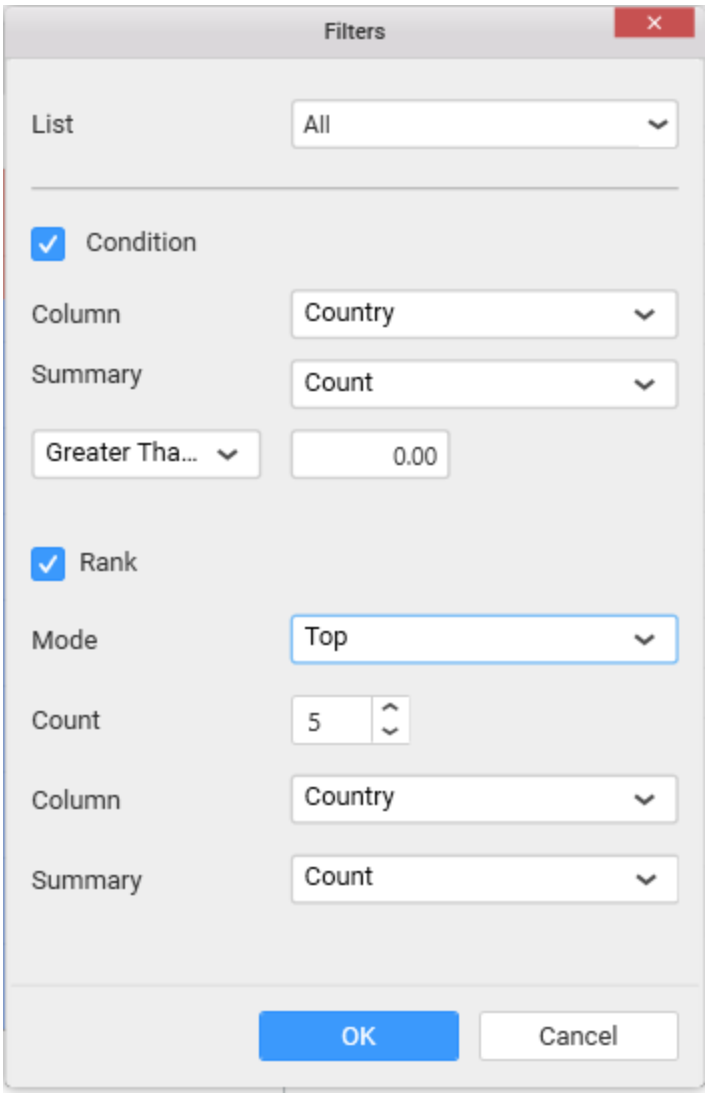


You can apply a filter.

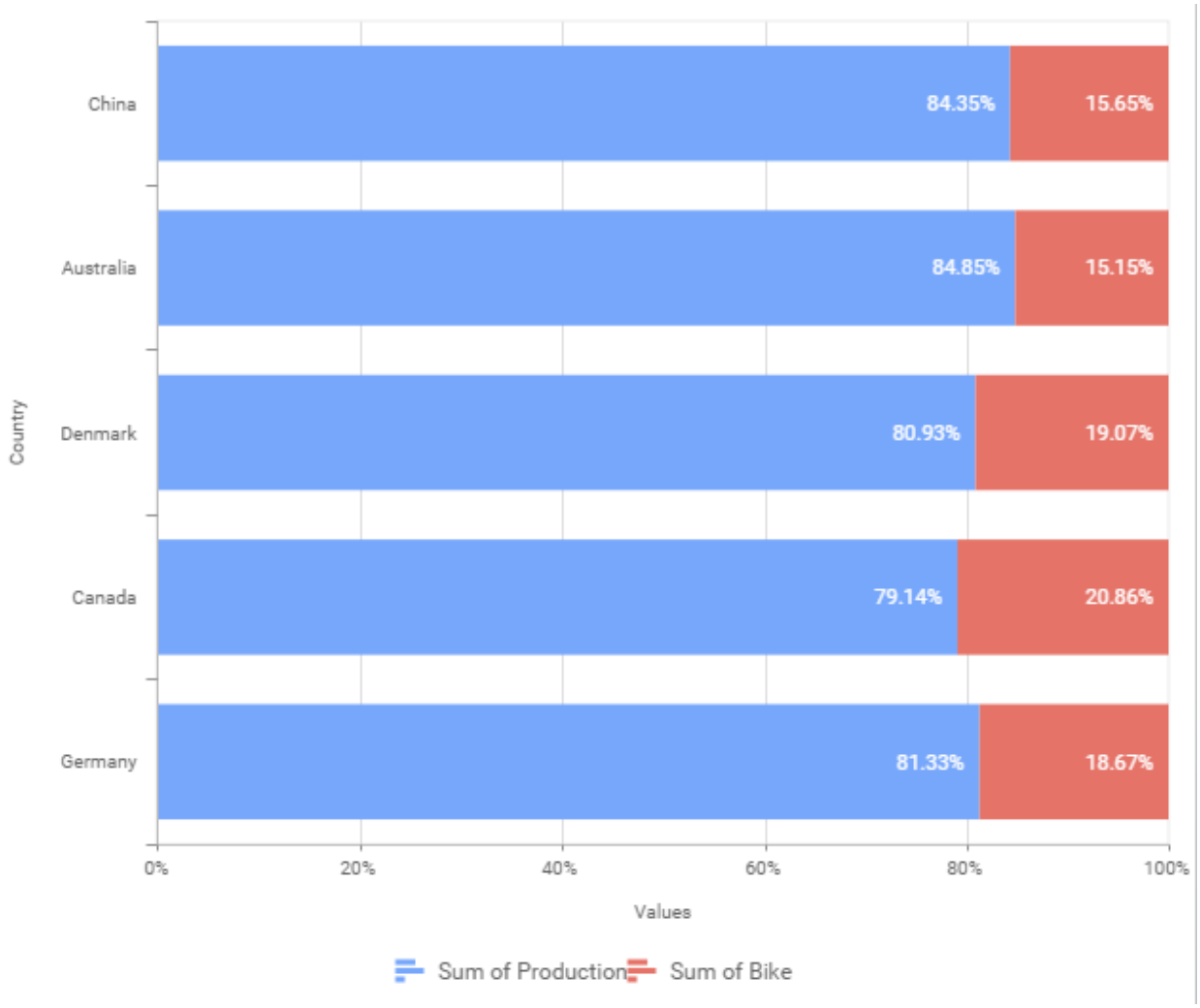




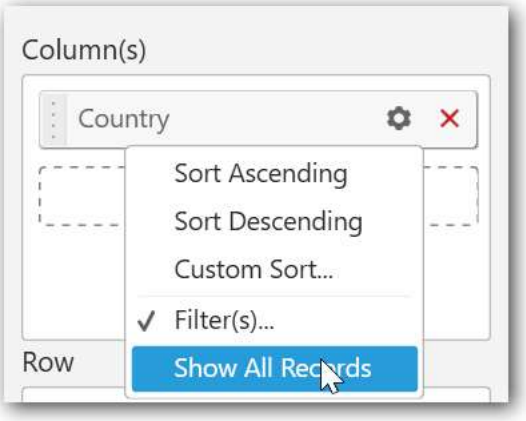
Select the Conditions and Rank you need.



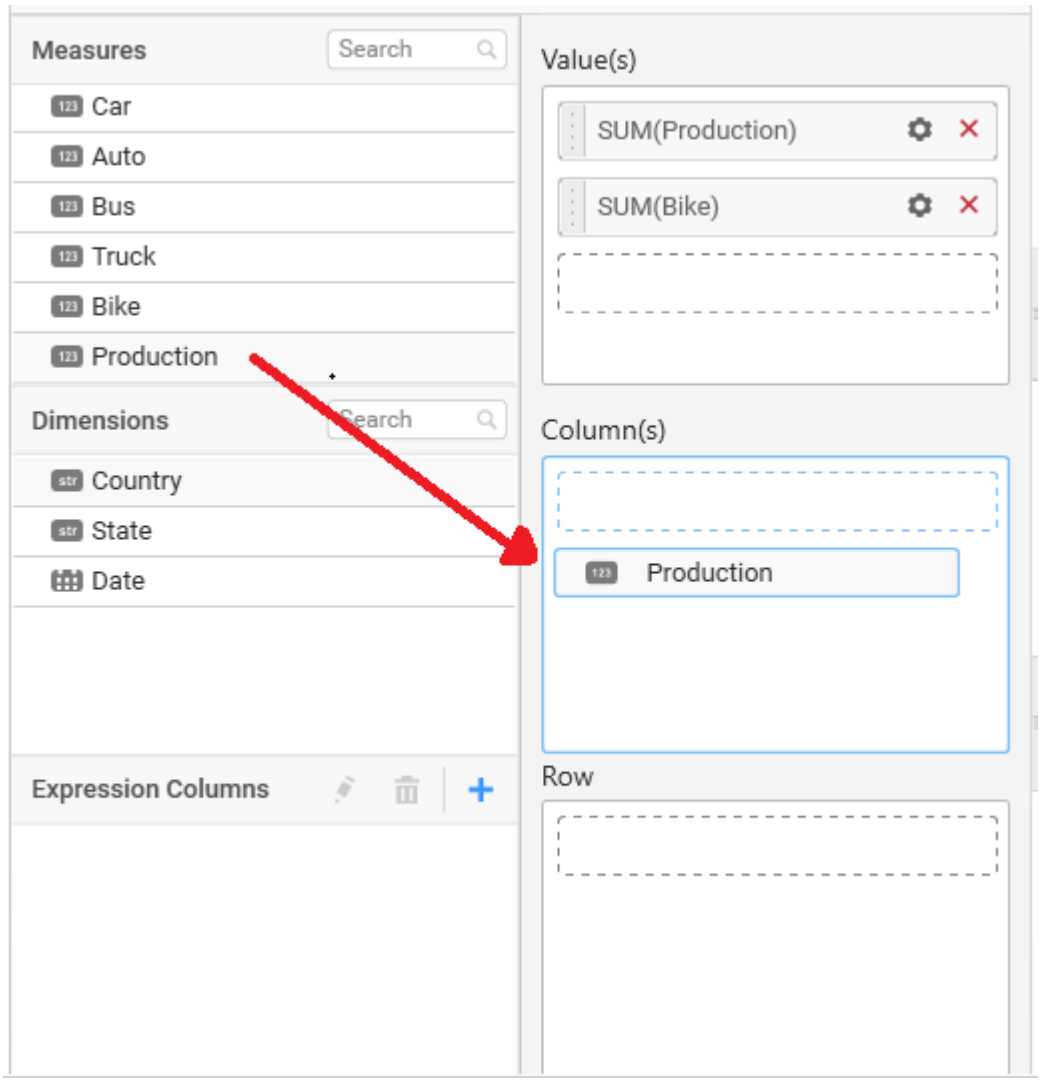
Now the chart will be rendered like this.

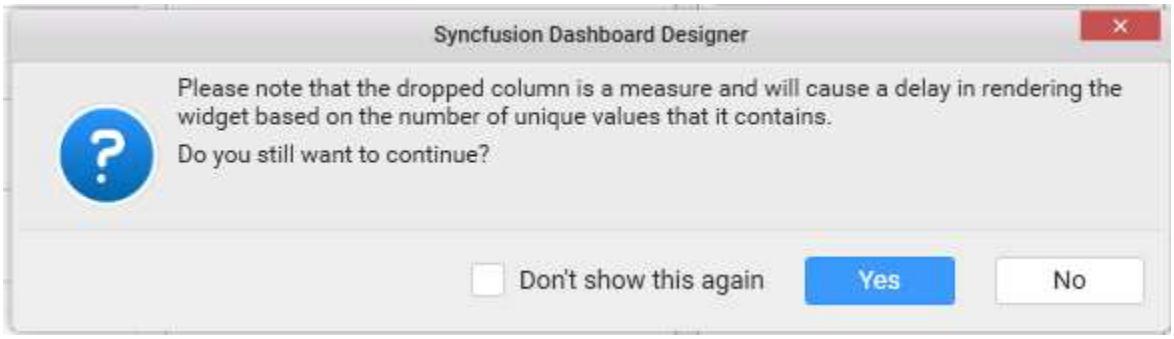


To show all records again click on Show All Records.



On adding Measures into Column(s) will show the following alert.

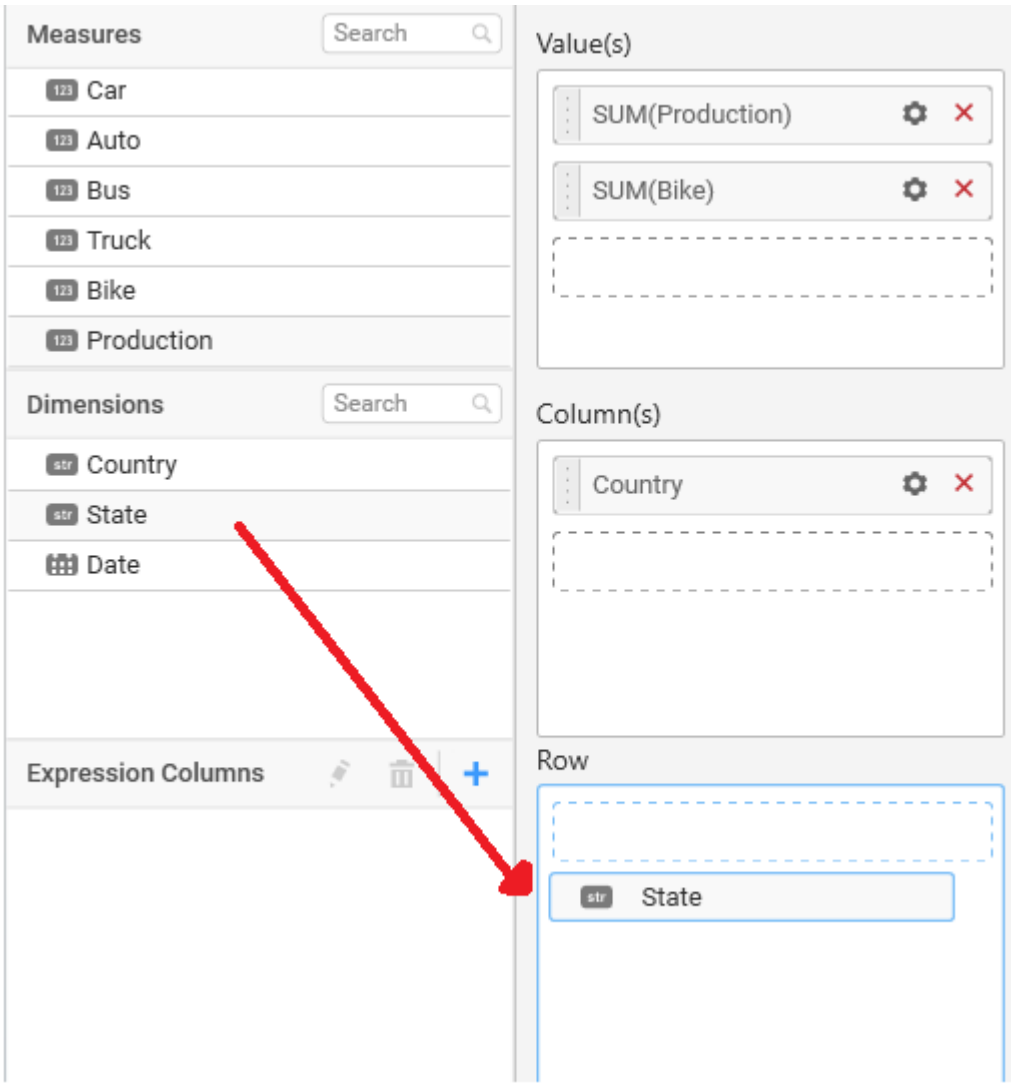




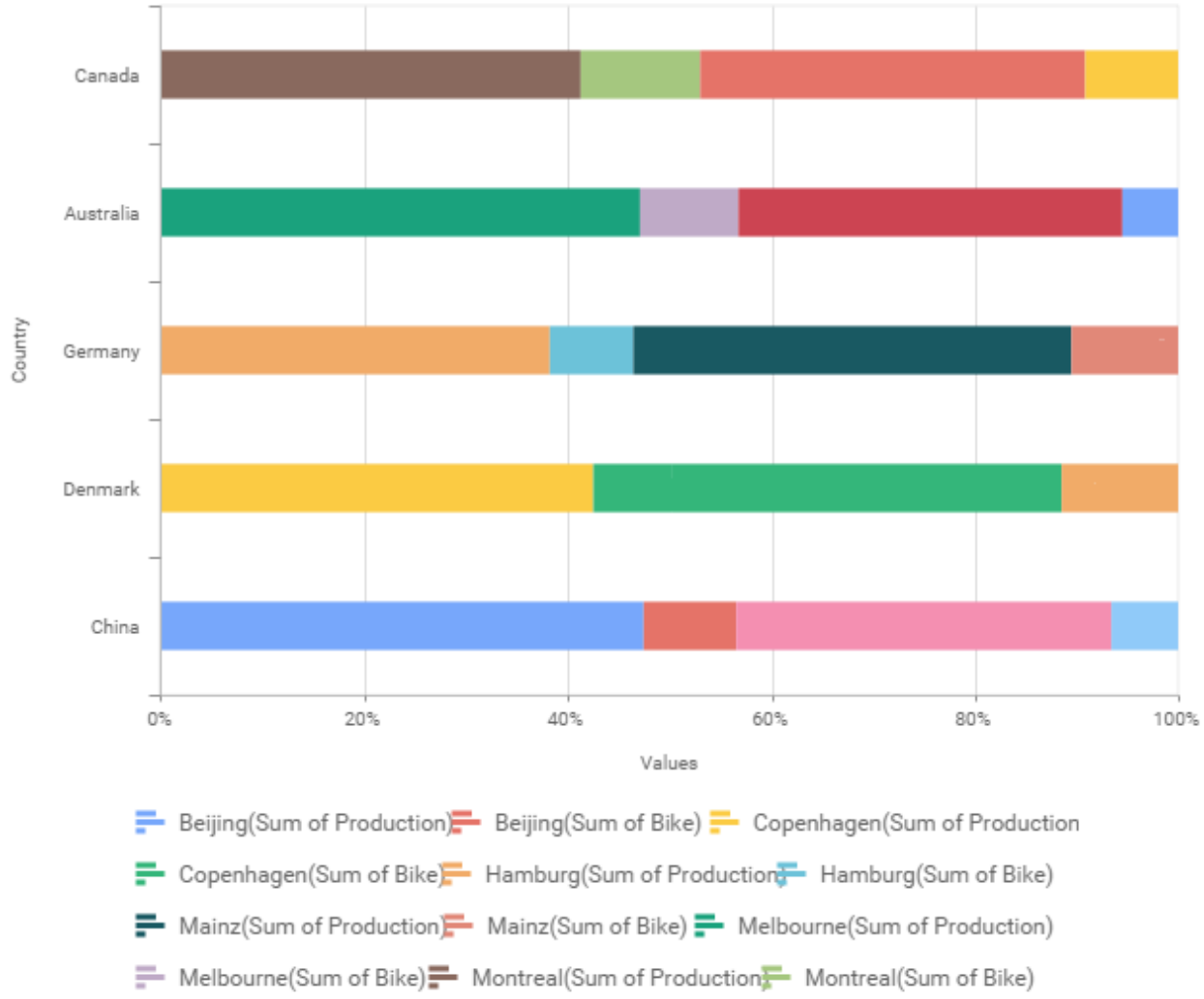
To continue select Yes, otherwise select No.

Assigning Row

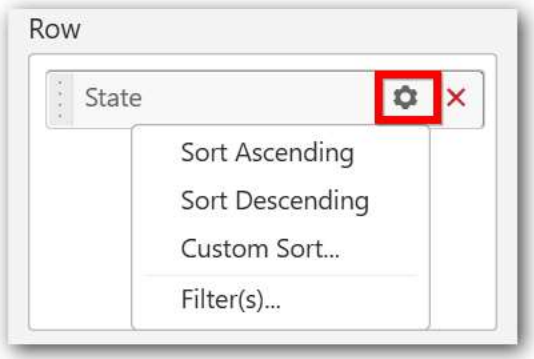
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image



You have settings options similar to Column(s).



How to configure SSAS data to 100% stacked bar Chart?

100% Stacked Bar Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you

would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

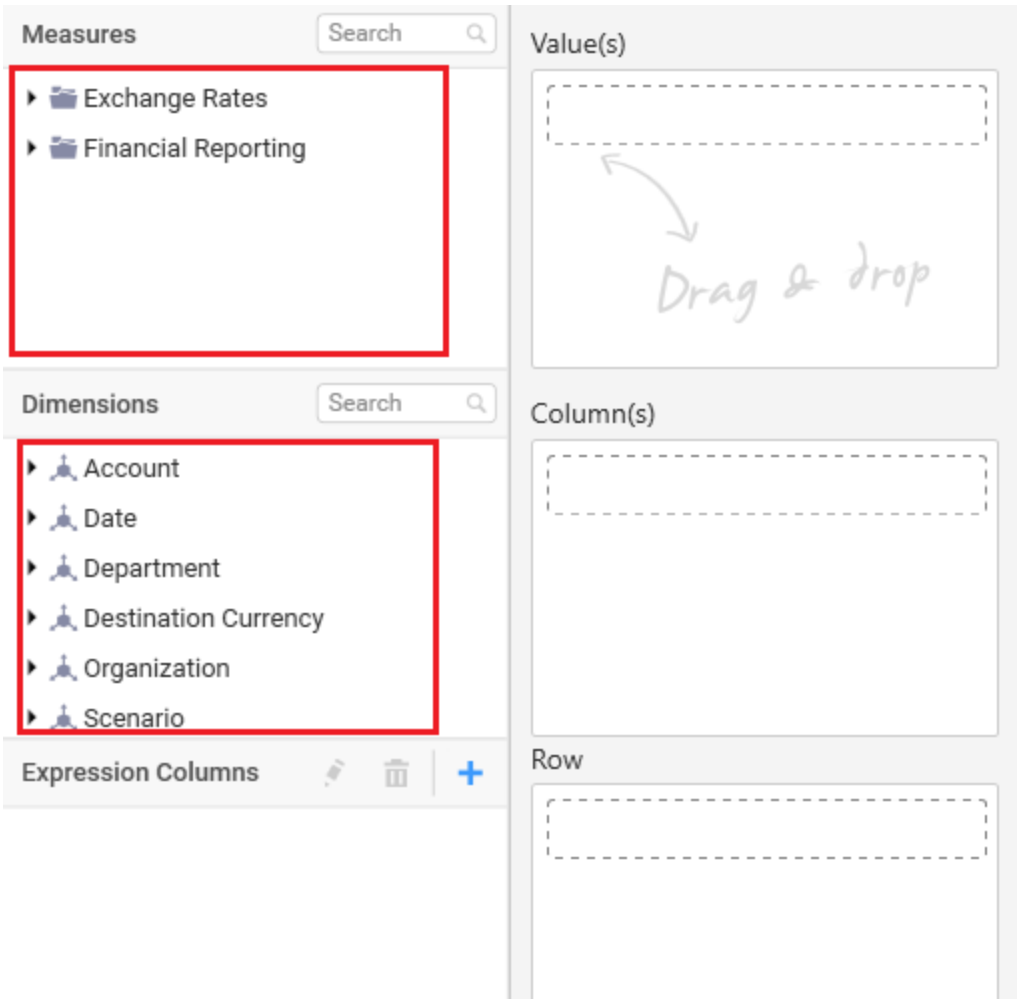
Following steps illustrates configuration of SSAS data to 100% Stacked Bar Chart

Drag and drop the 100% Stacked Bar chart widget into canvas and resize into your required size.

Select the dropped widget using mouse.

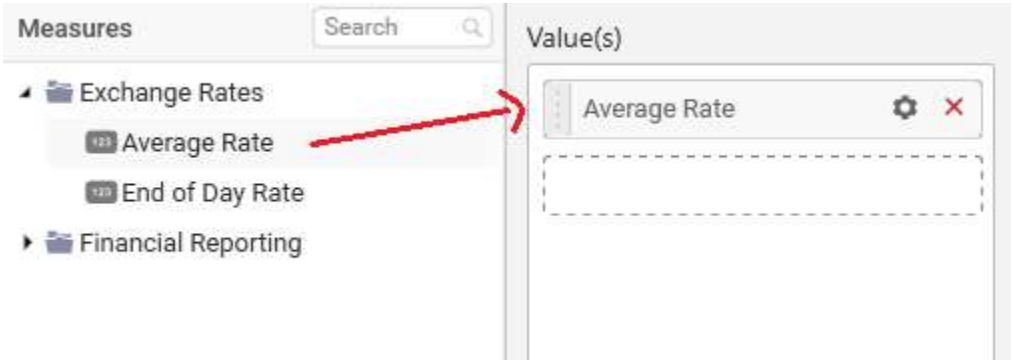
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions.

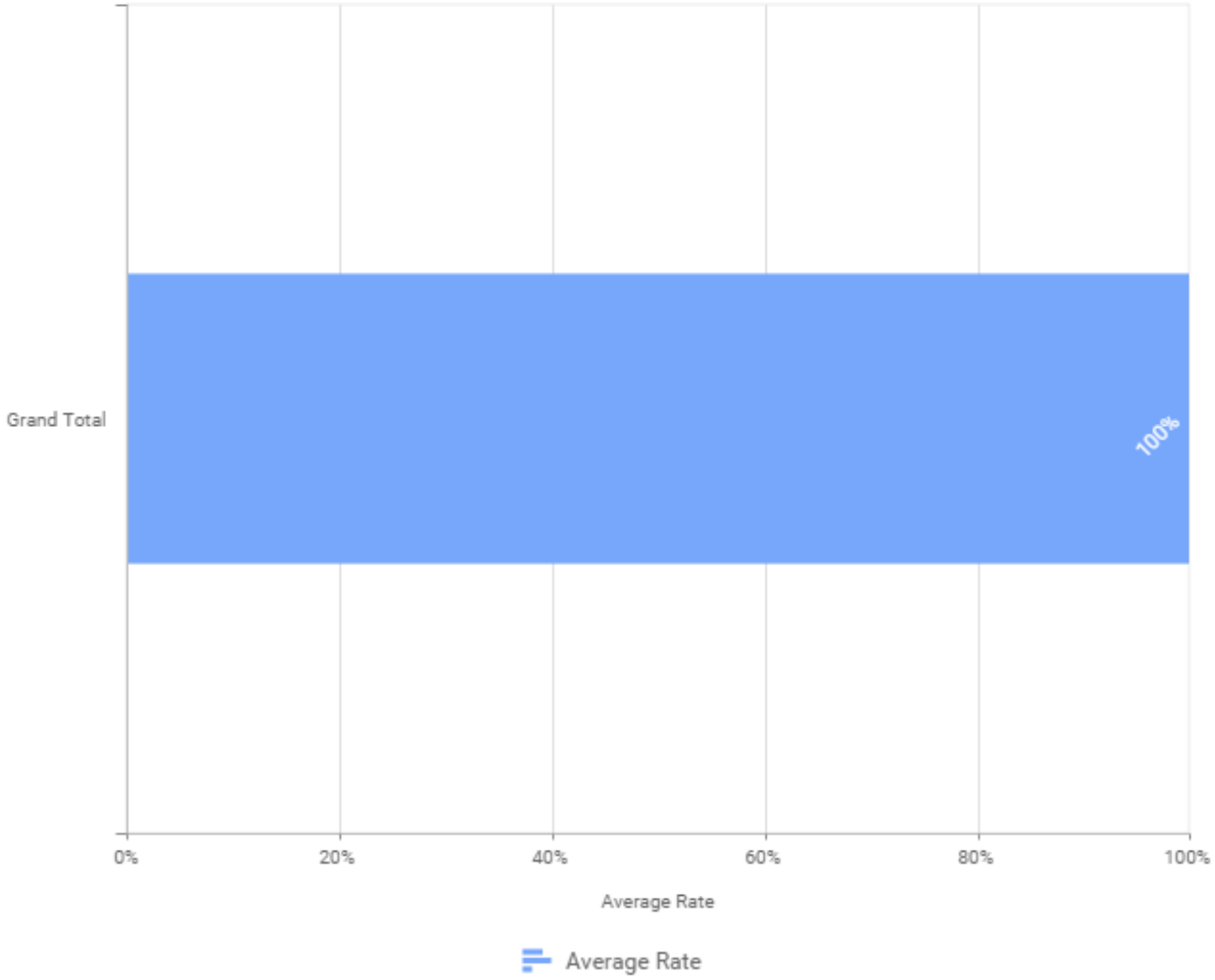


Assigning Value(s)

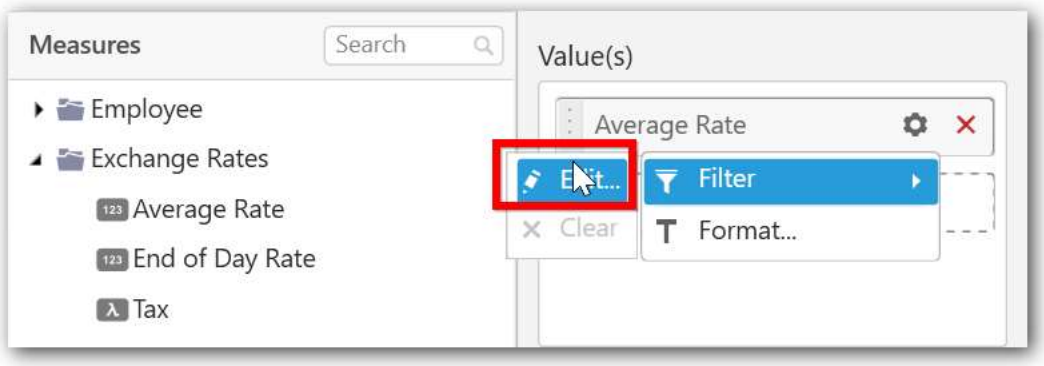
Drag and drop a column under Measures category into Value(s) section.



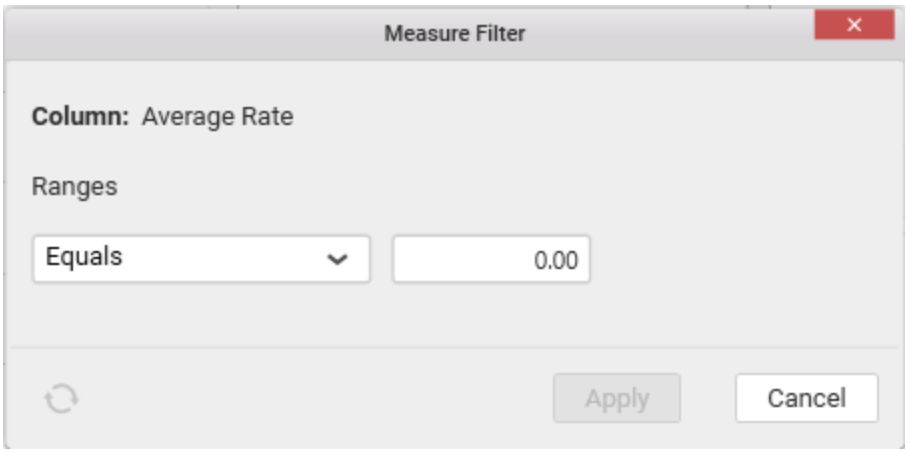
Now the chart will be rendered like this.



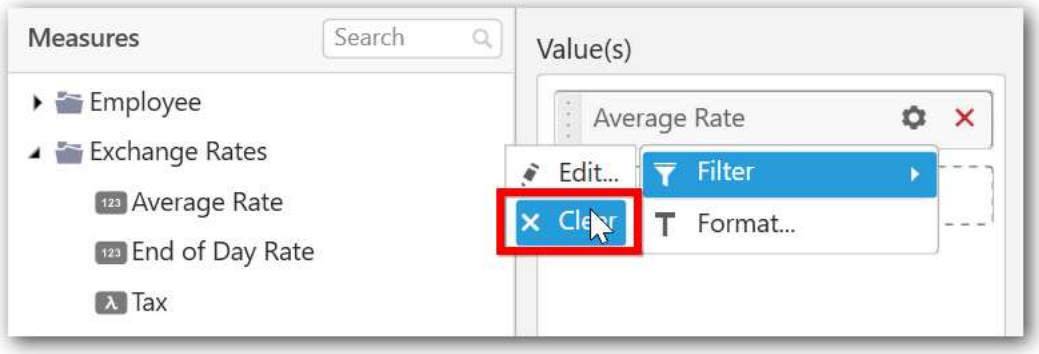
Define the filter criteria to match through choosing Edit option in Filter menu item.



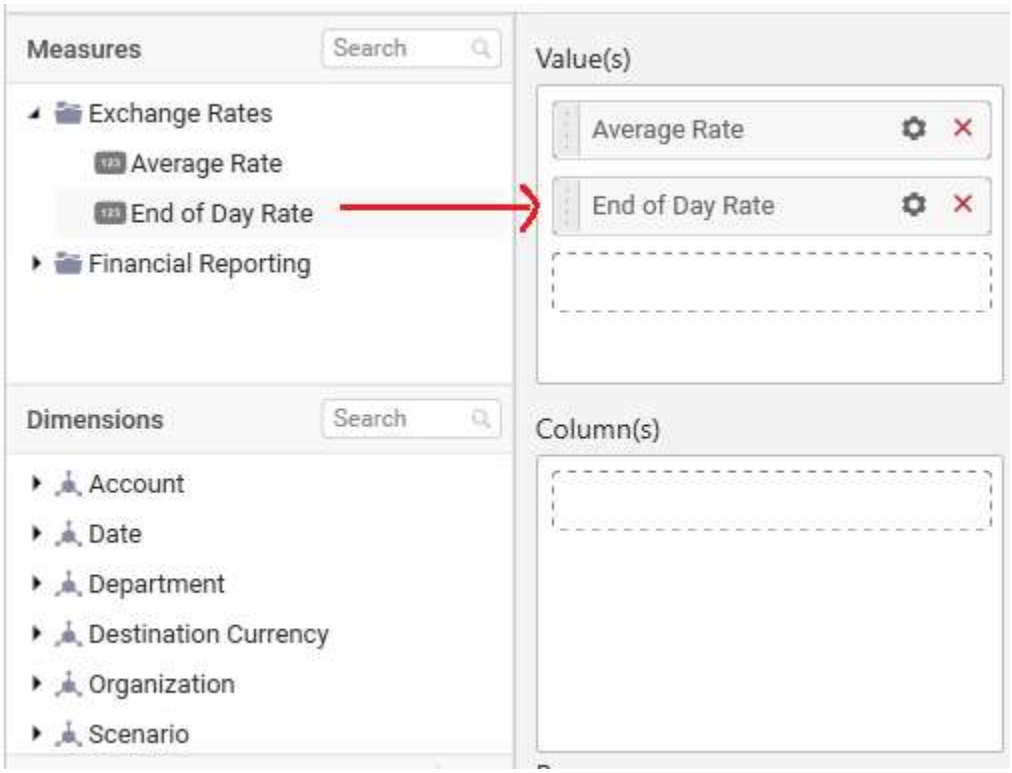
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.

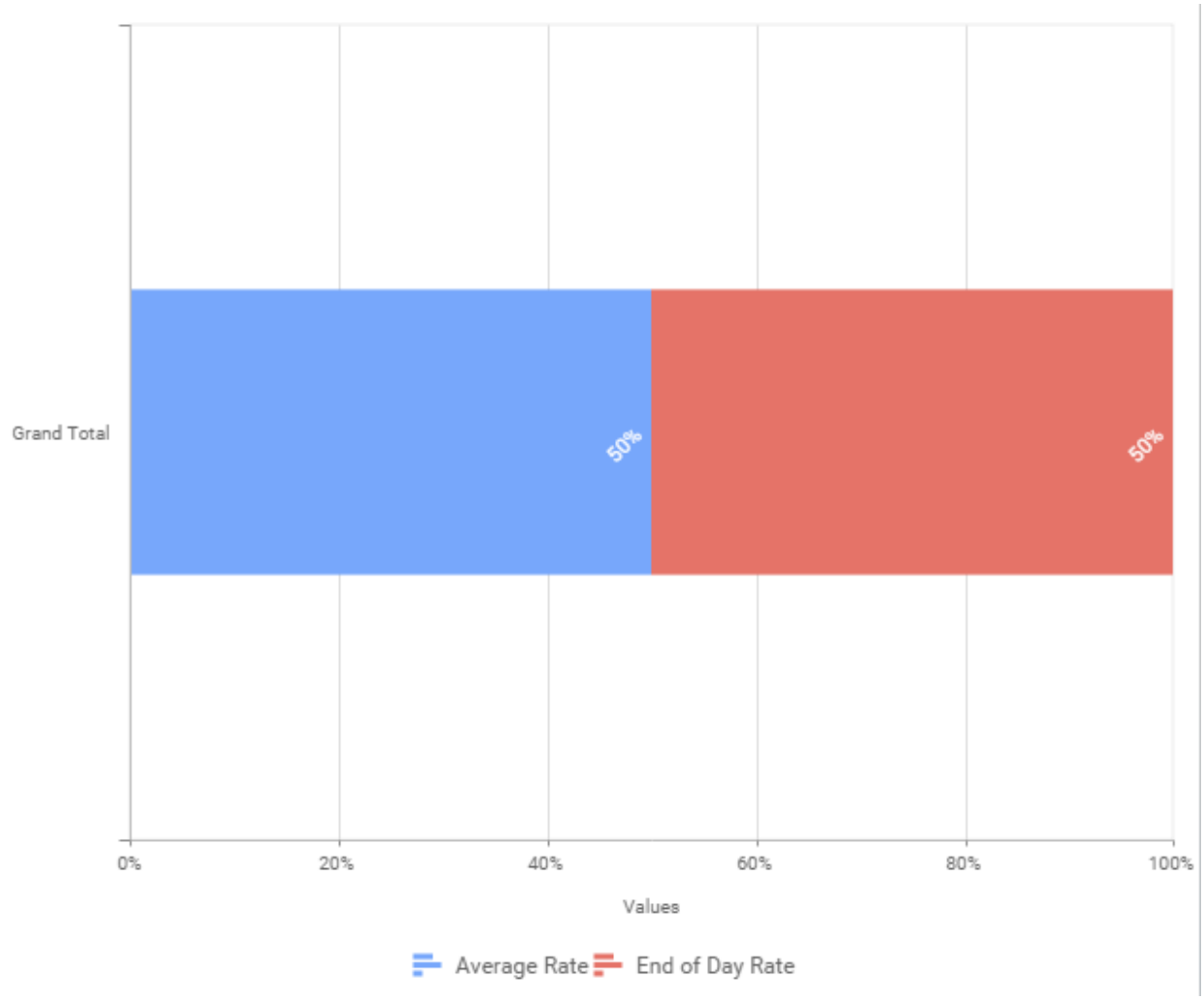


Select Clear option to clear the defined filter.



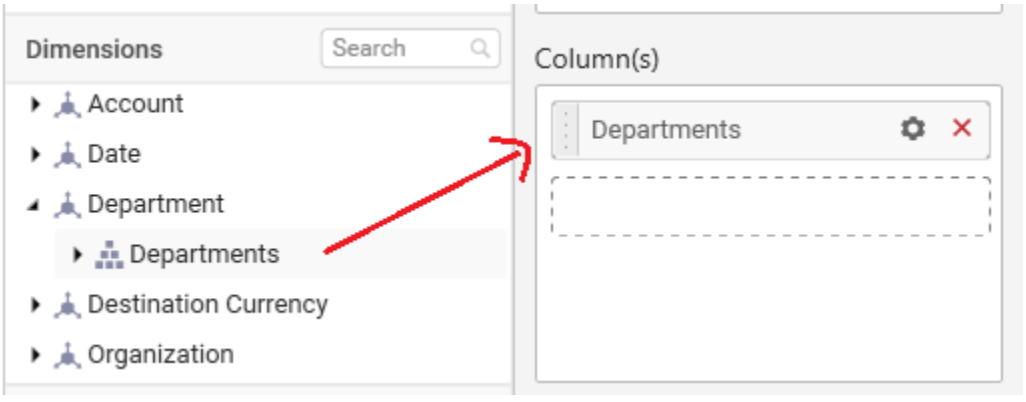
You can also add more than one column to the Value(s) section

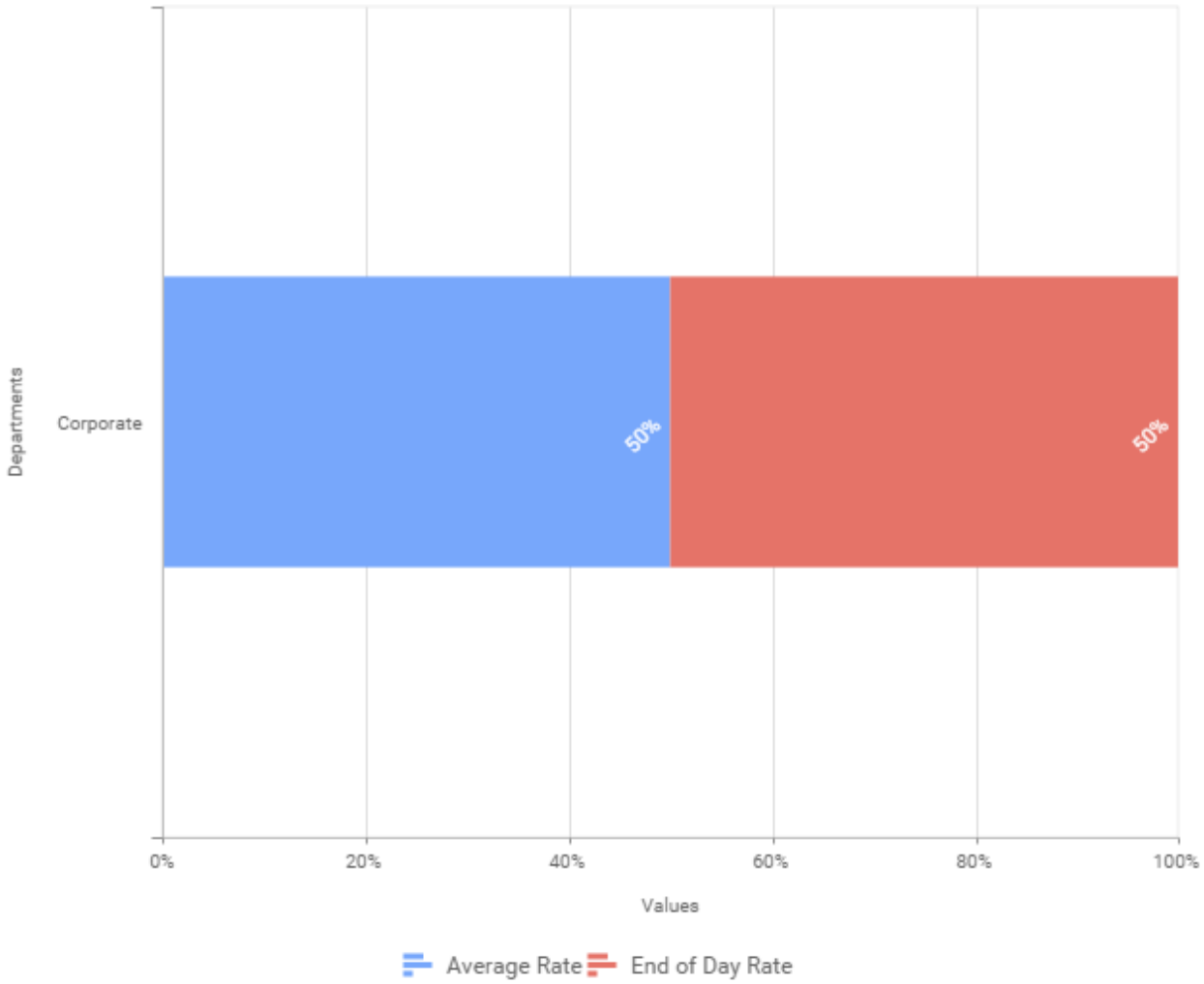




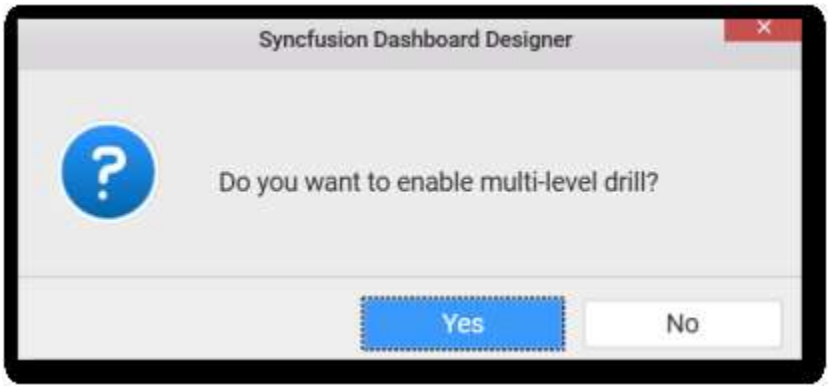
Assigning Column(s)

You can also add more than one column to the Value(s) section.



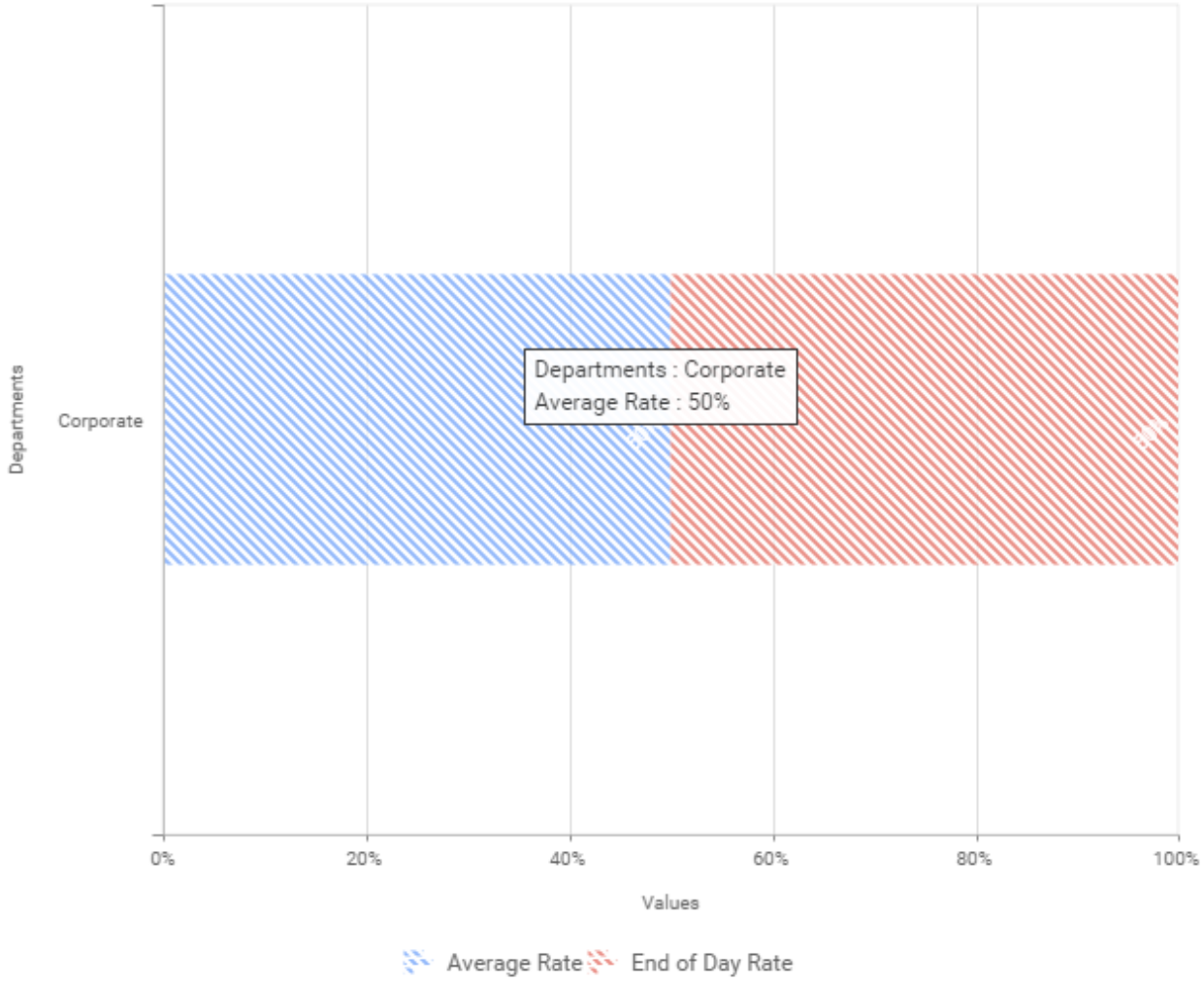


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.



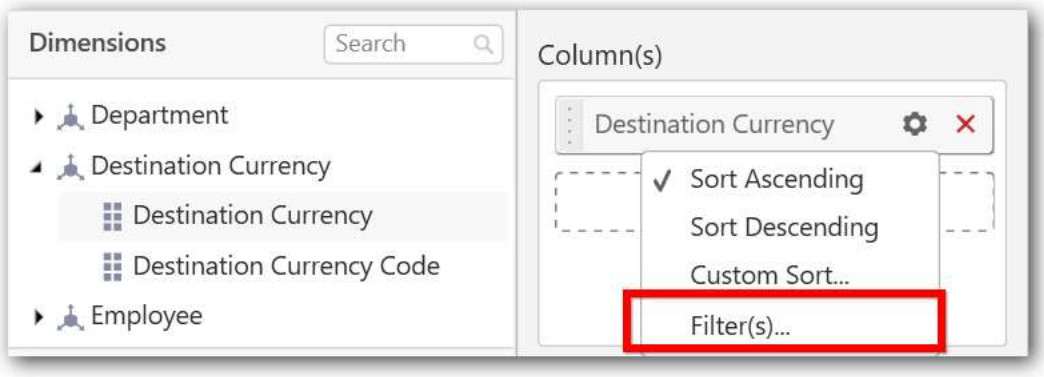
Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

Click the respective data value marker in chart to drill into its inner level.

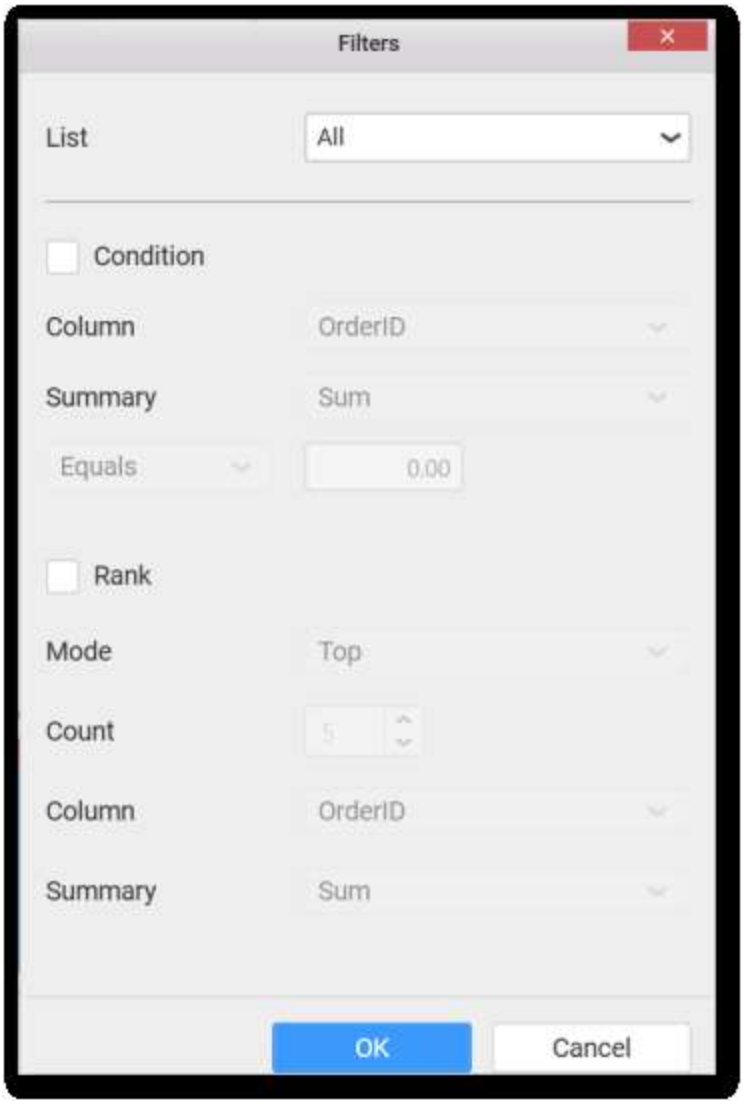


Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level.

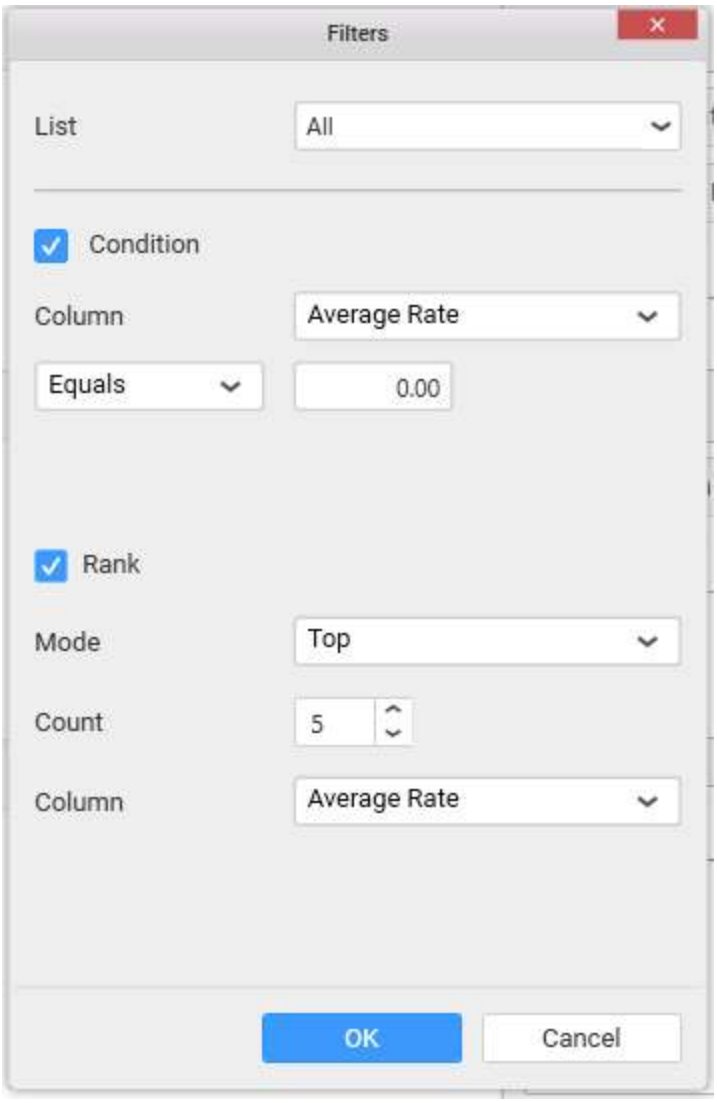
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



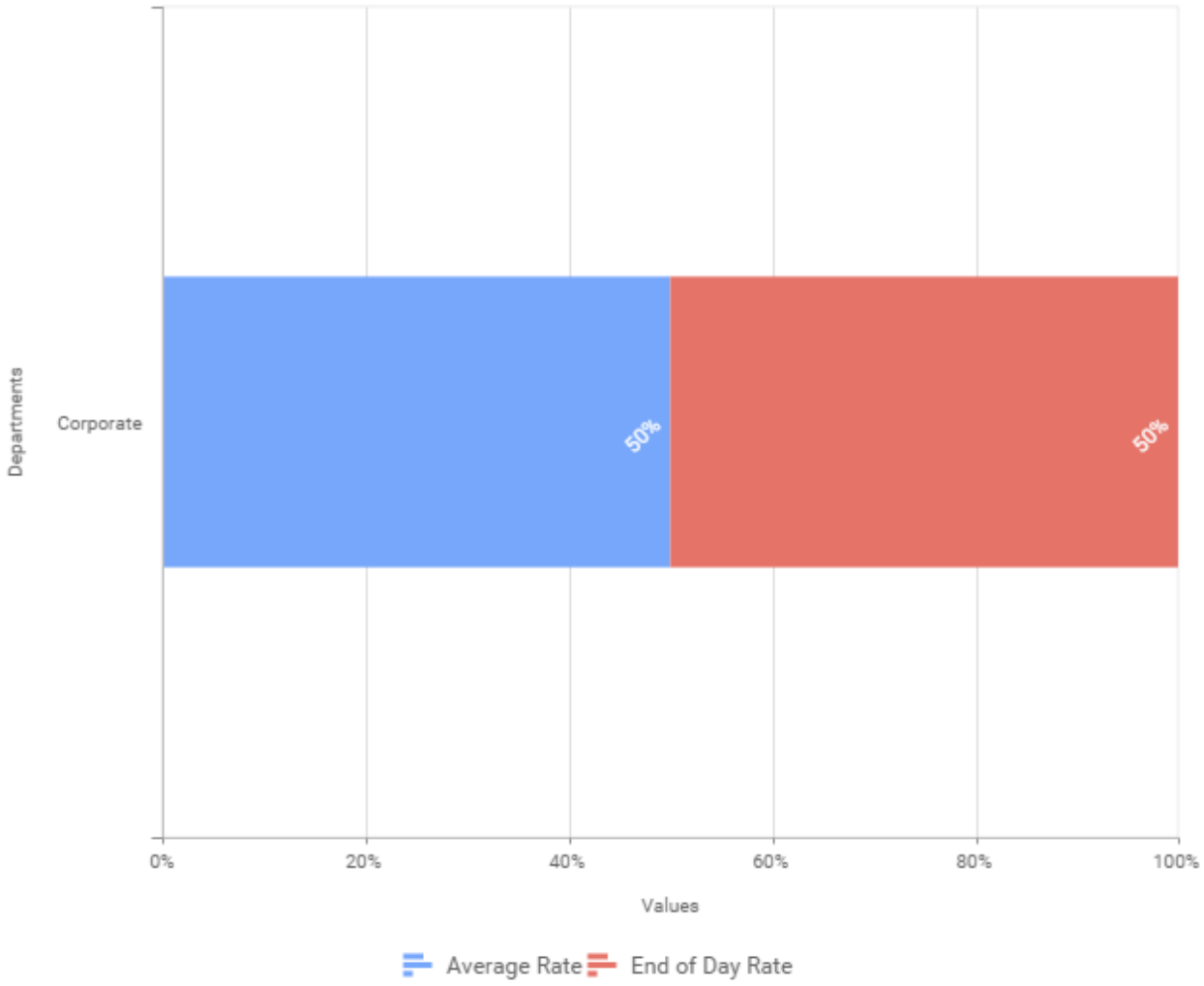
Select Filter(s)… to launch the Filters window.



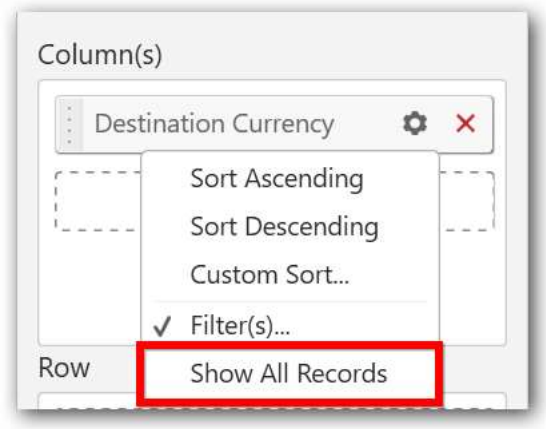
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

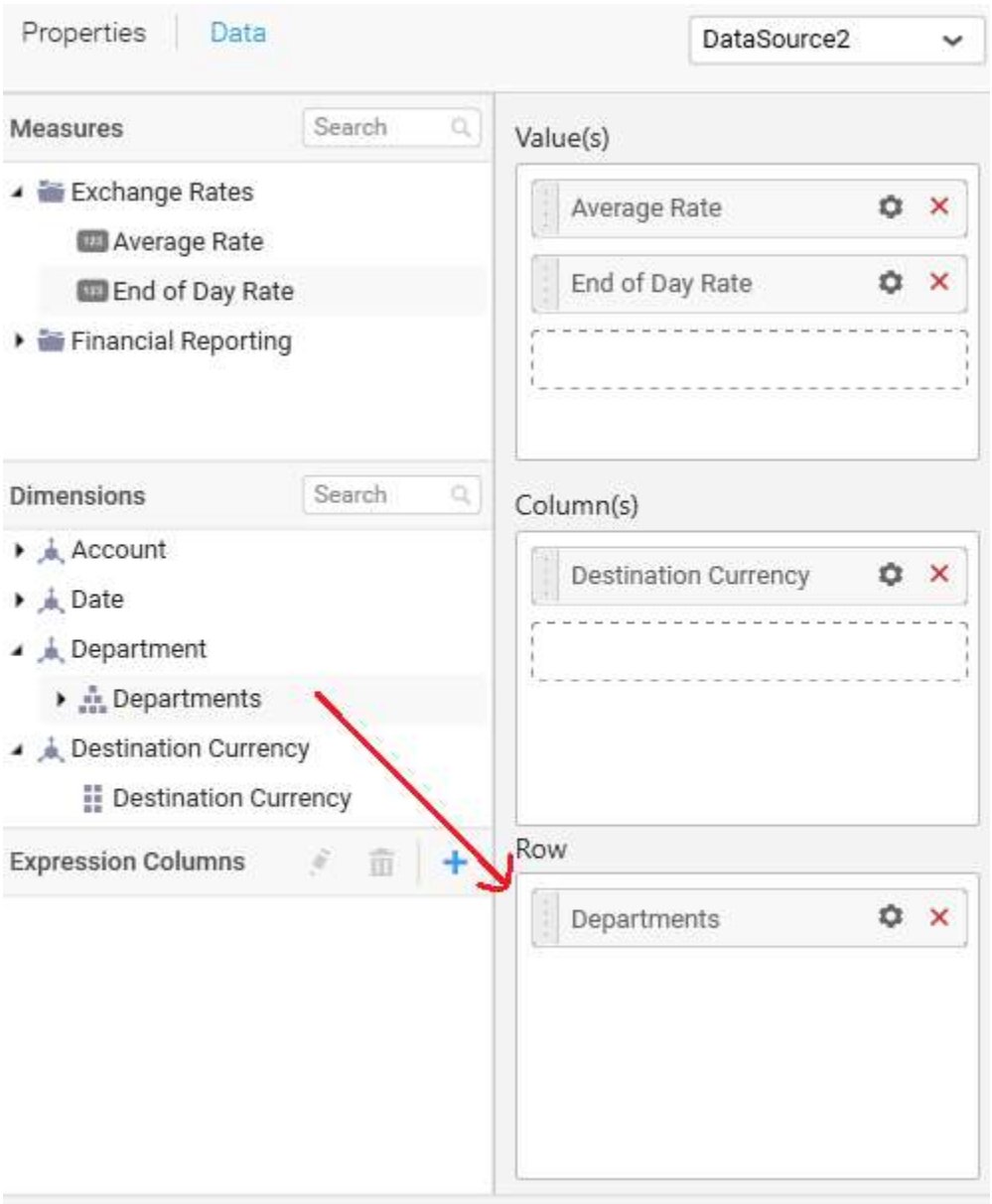


To show all records again click on Show All Records.

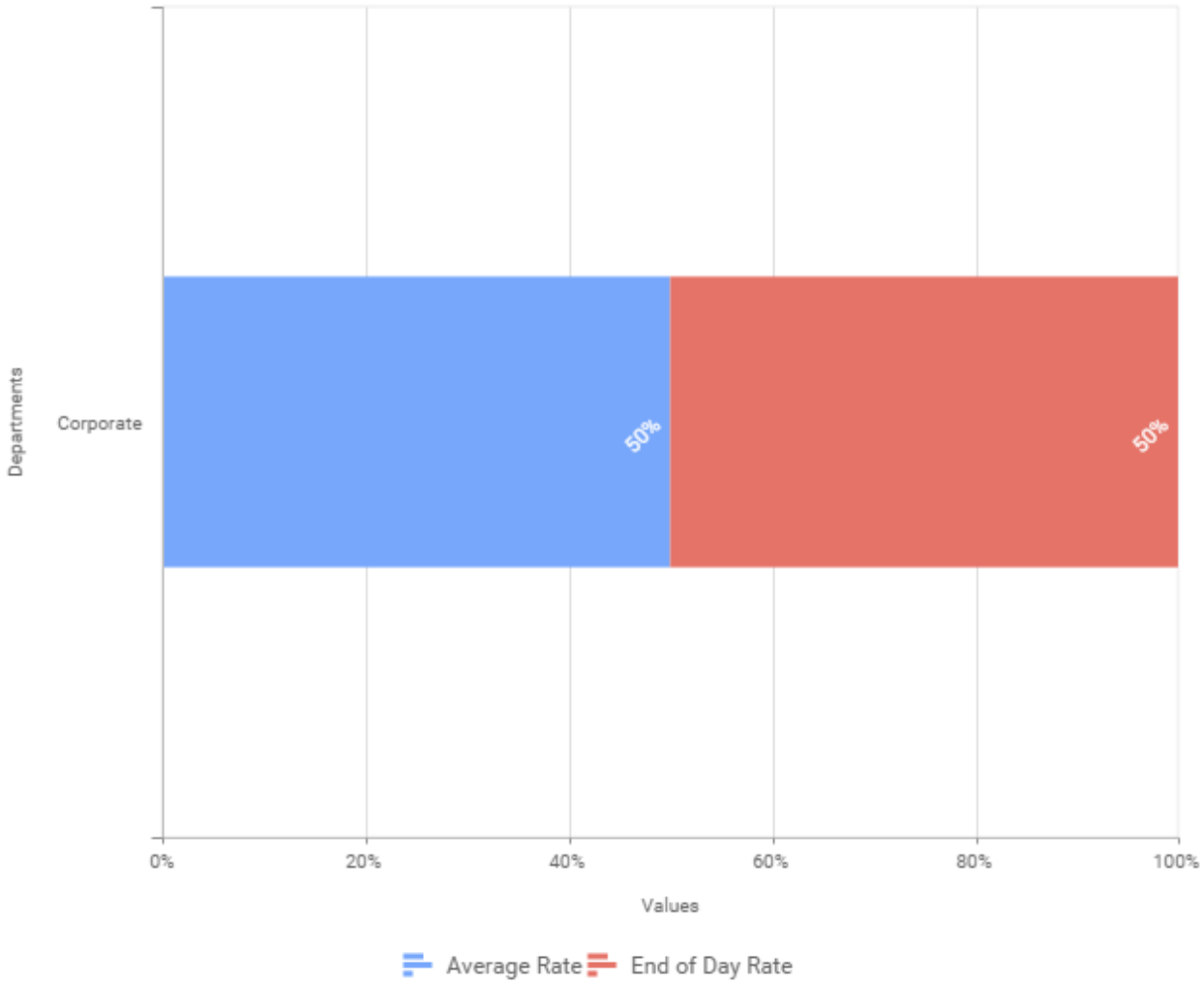


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.

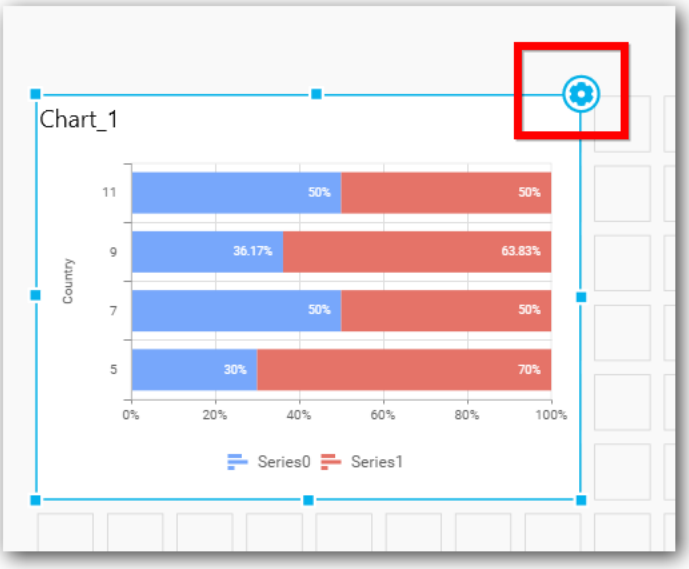


How to format 100% Stacked Bar Chart?

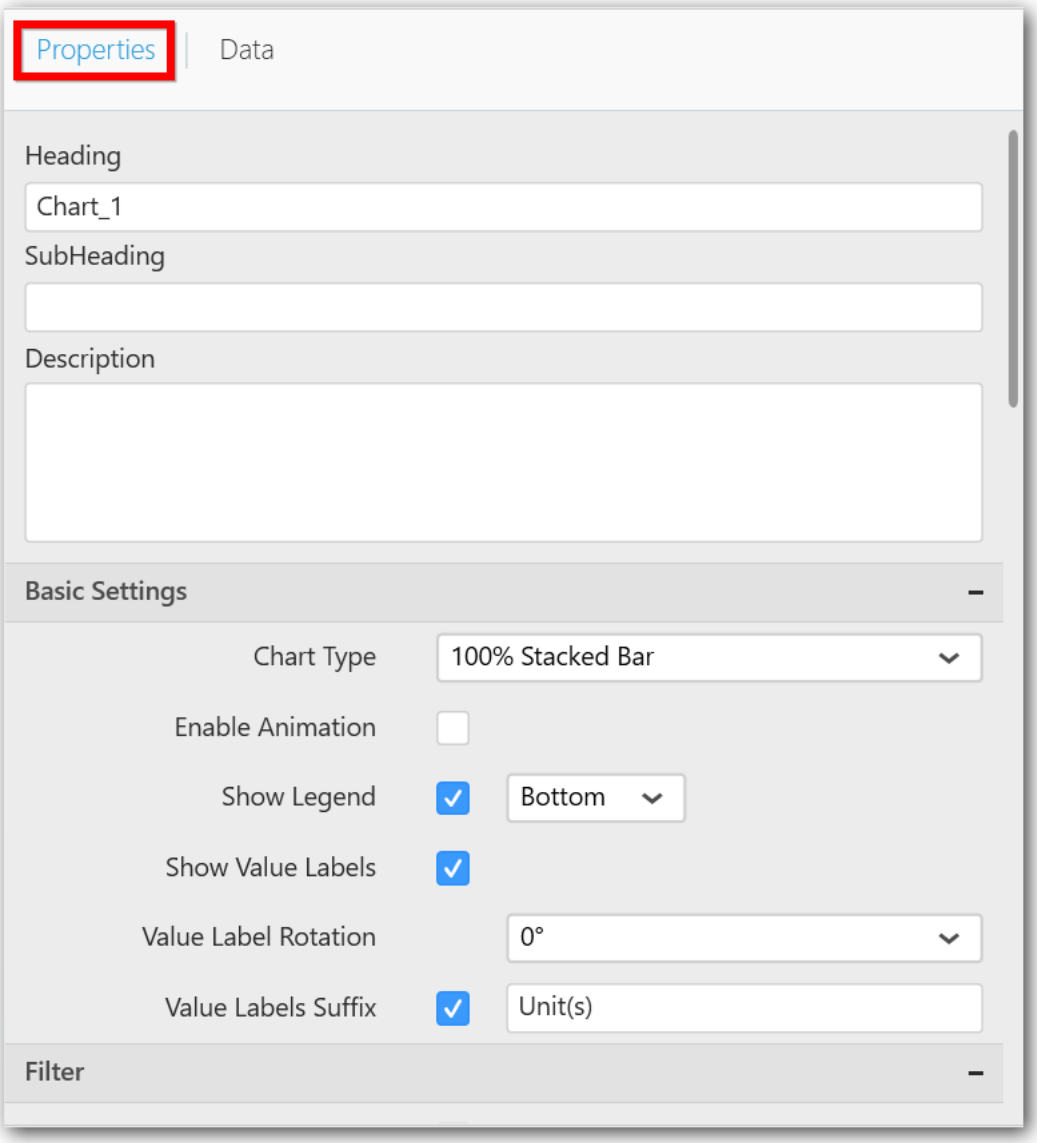
You can format the 100% stacked bar chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into 100% stacked bar chart follow the steps

1. Drag and drop the 100% stacked bar chart into canvas and resize it to your required size.
2. Configure the data into 100% stacked bar chart.
3. Focus on the 100% stacked bar chart and Click on Widget Settings.

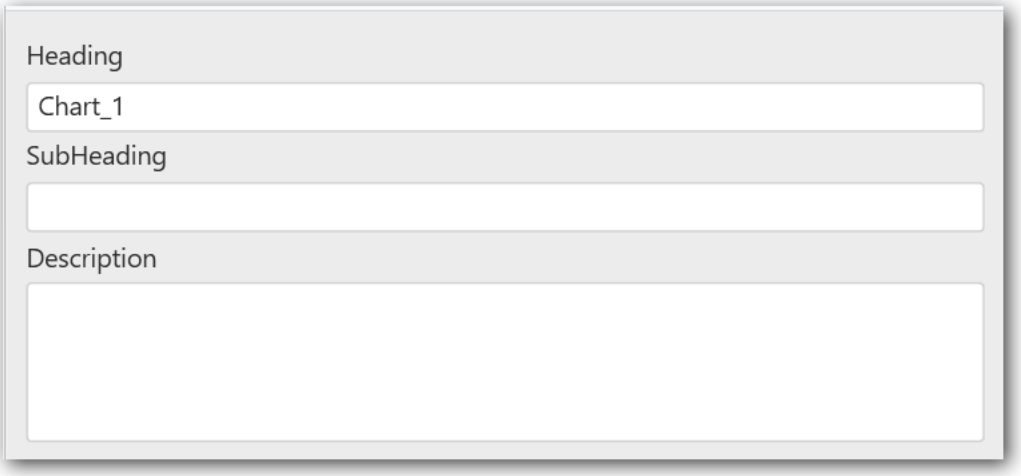


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this 100% stacked bar chart widget.

SubHeading

This allows you to set sub-title for this 100% stacked bar chart widget.

Description

This allows you to set description for this 100% stacked bar chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

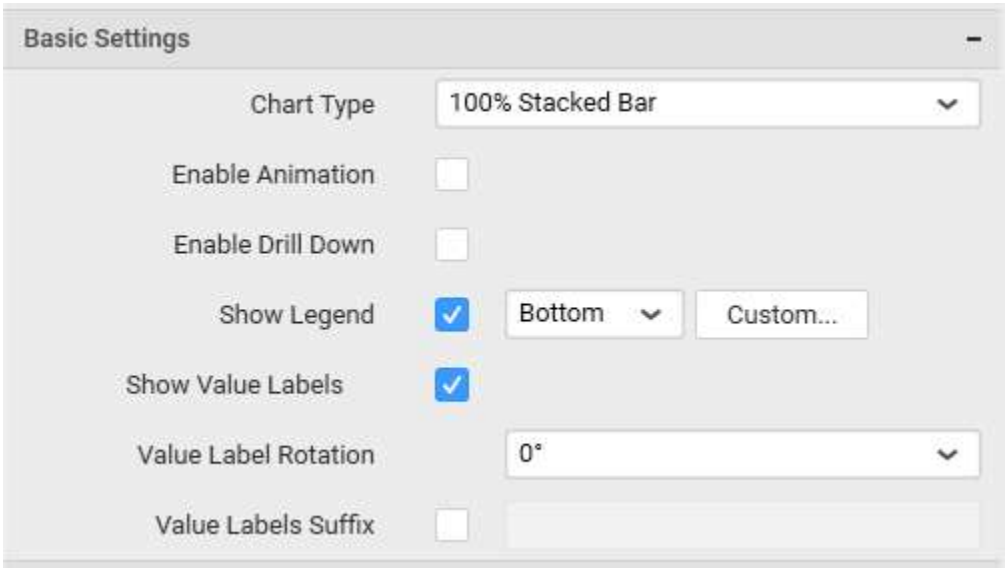


Chart Type

This allows you to switch the widget view from current chart type to another convertible chart type.

Enable Animation

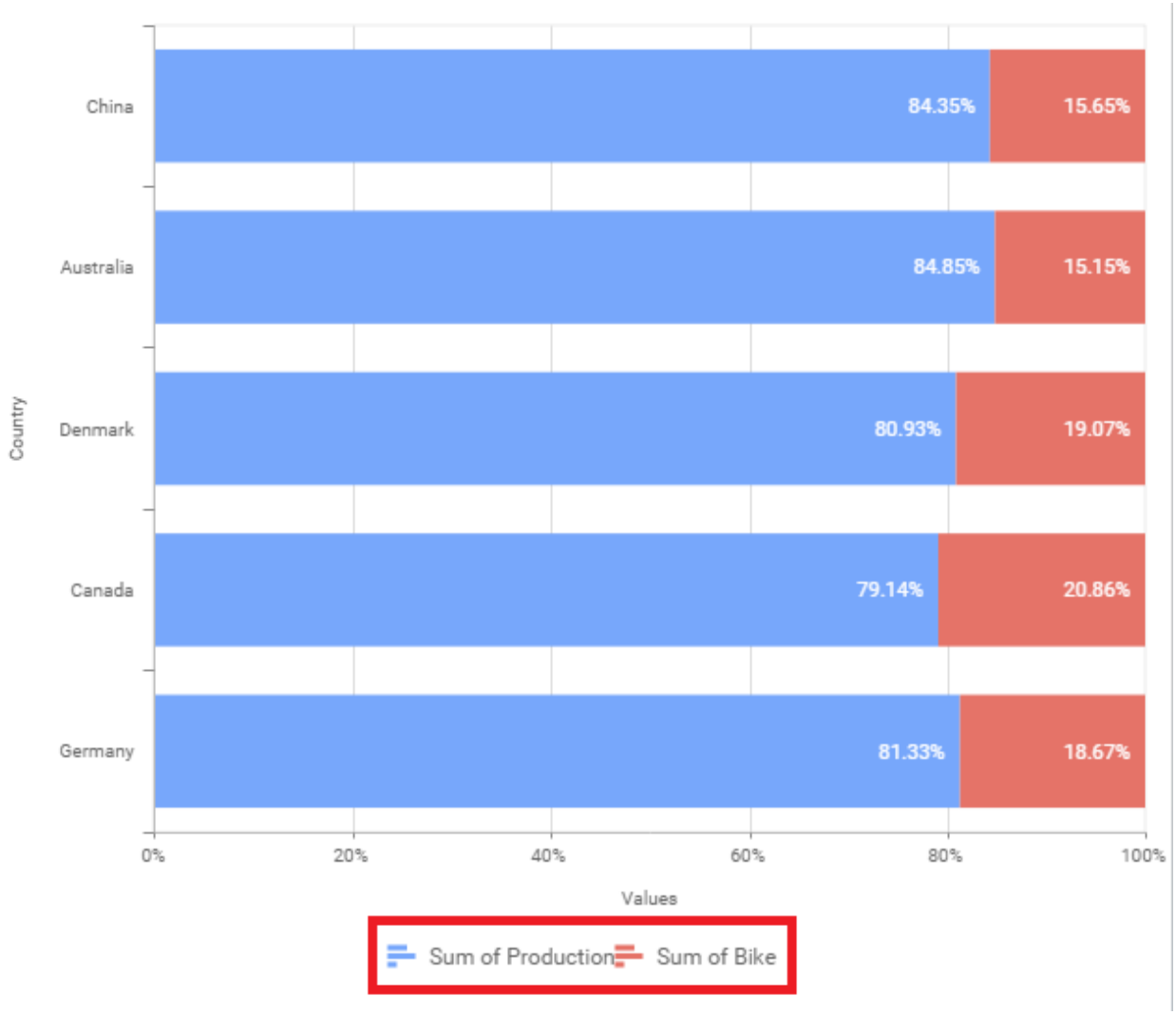
This allows you to enable the series rendering in animated mode.

Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling the option Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

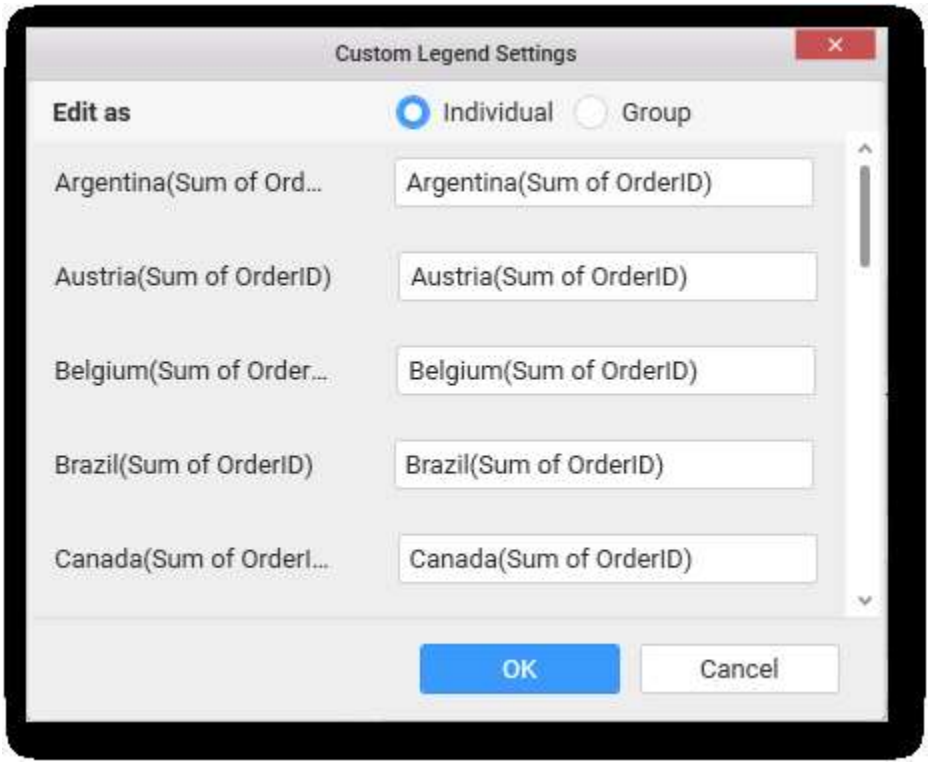
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

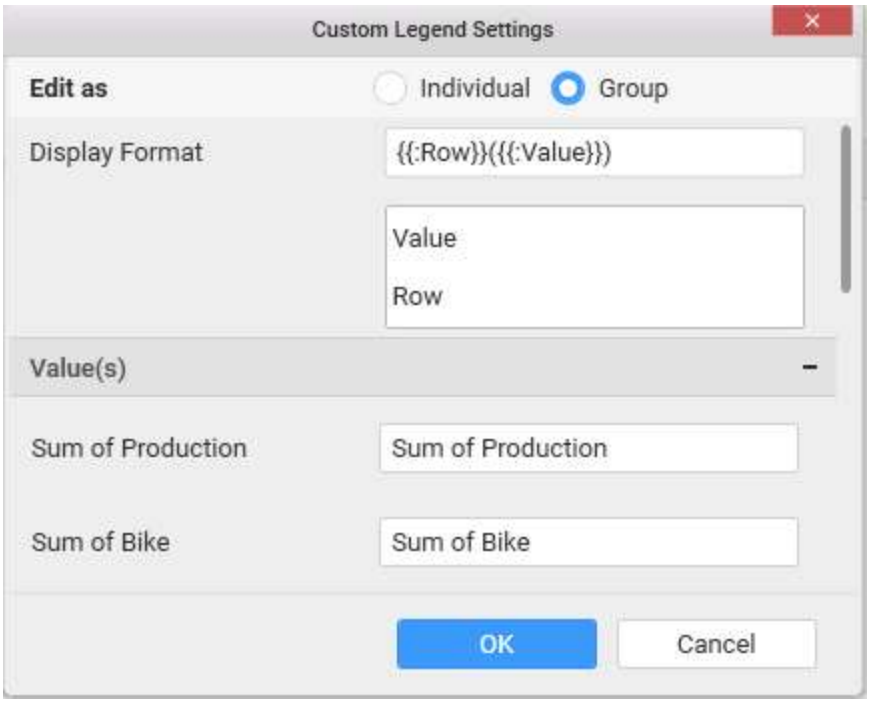
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

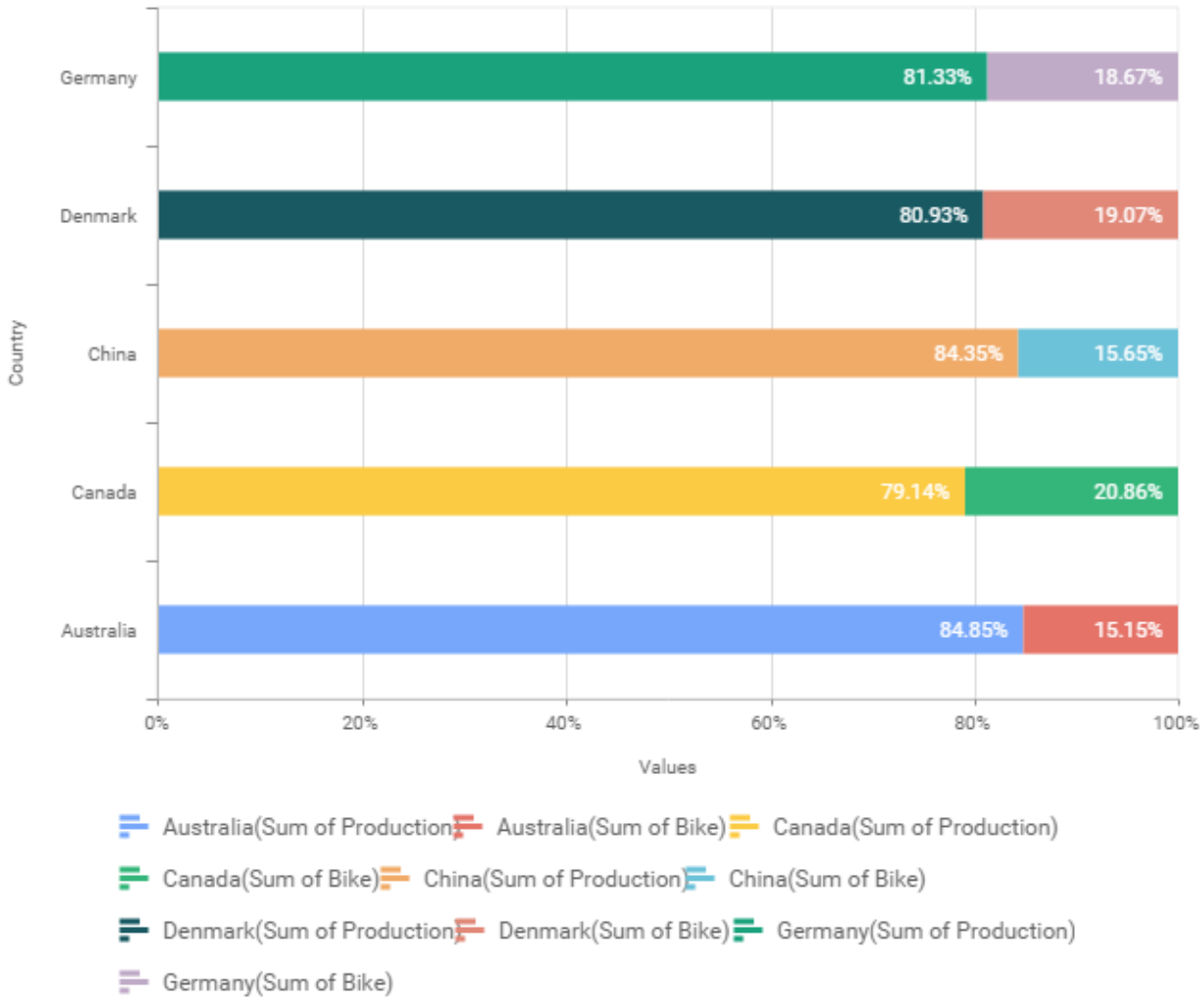


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

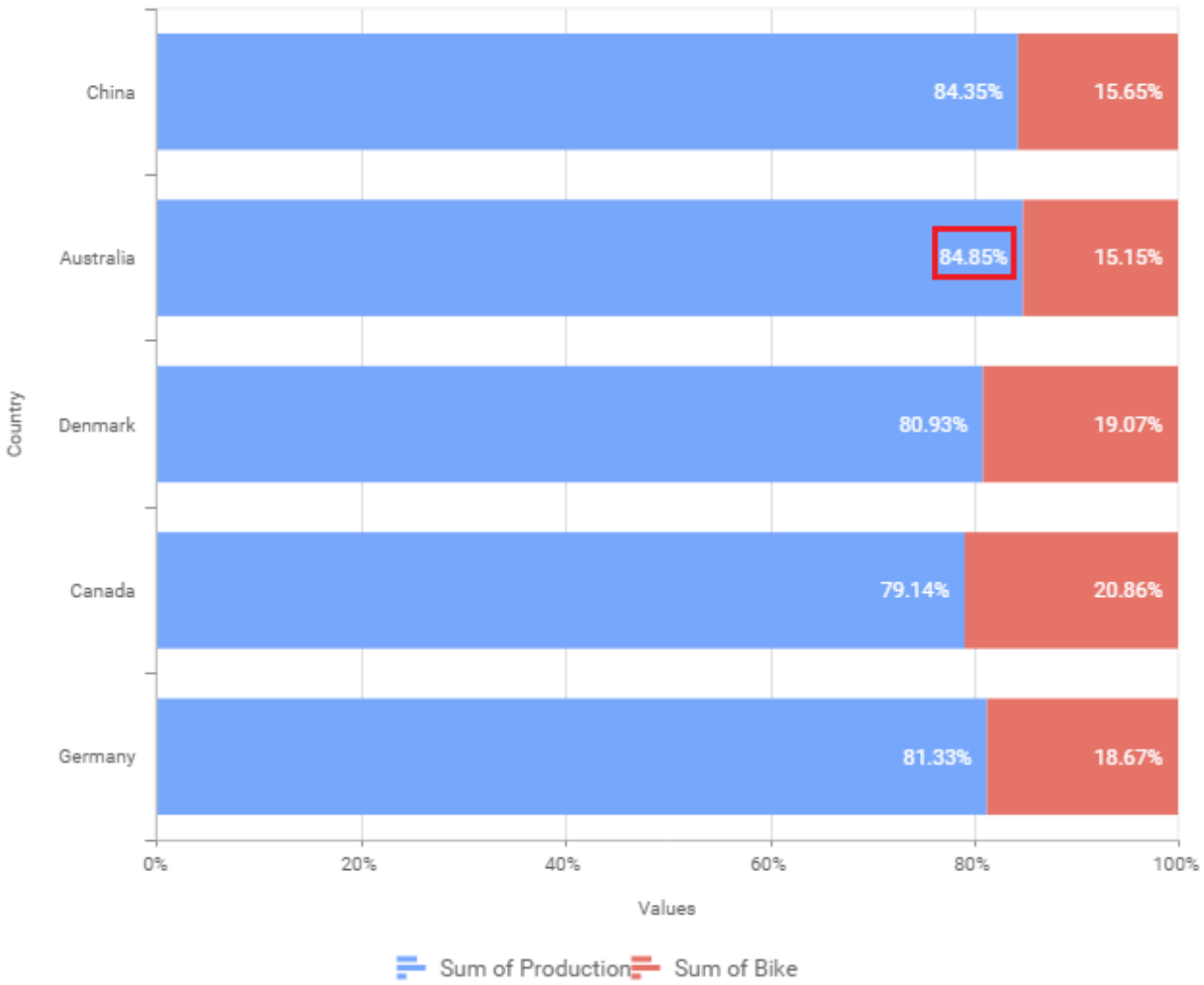


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



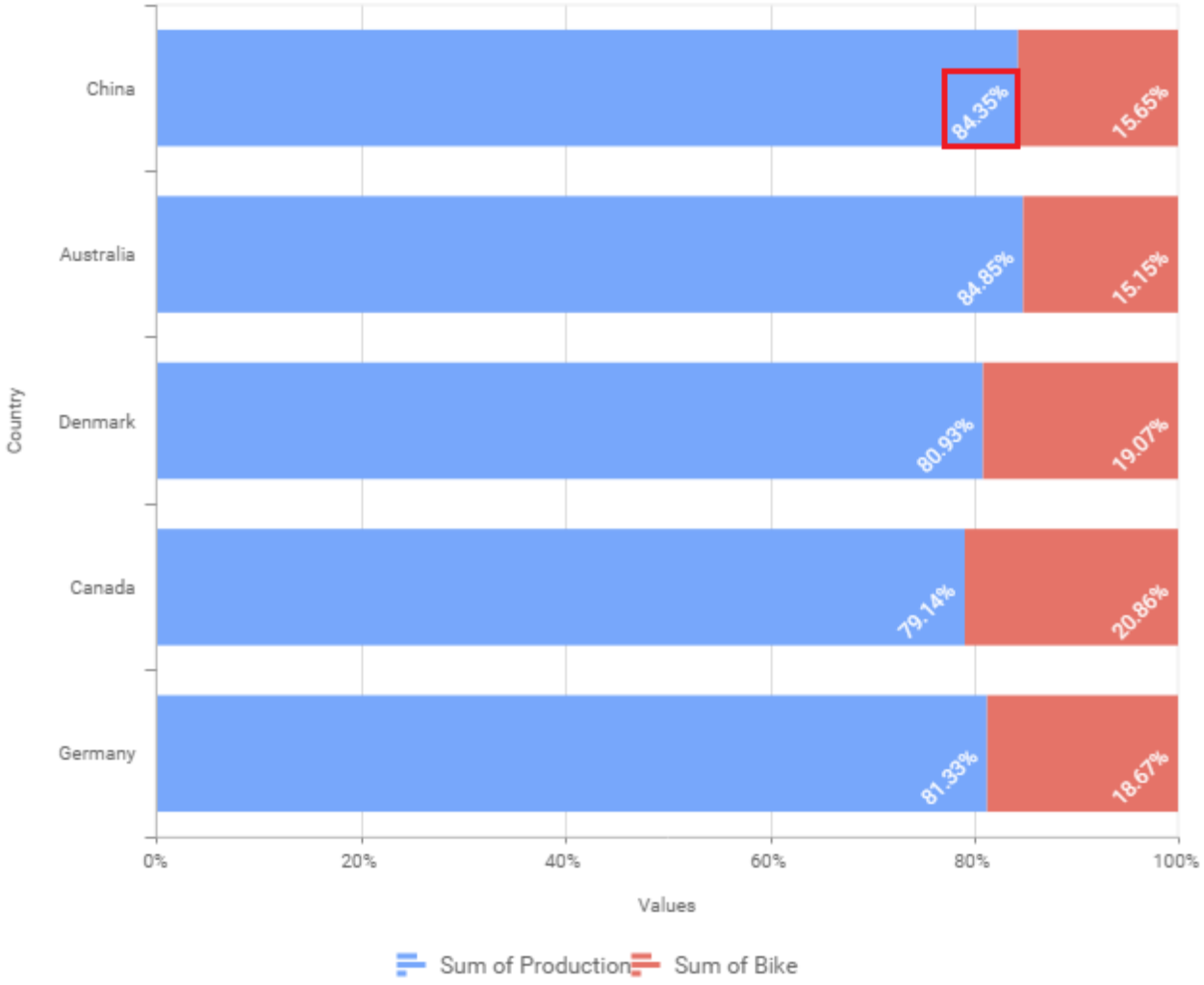
Show Value Labels

This allows you to toggle the visibility of value or data labels.



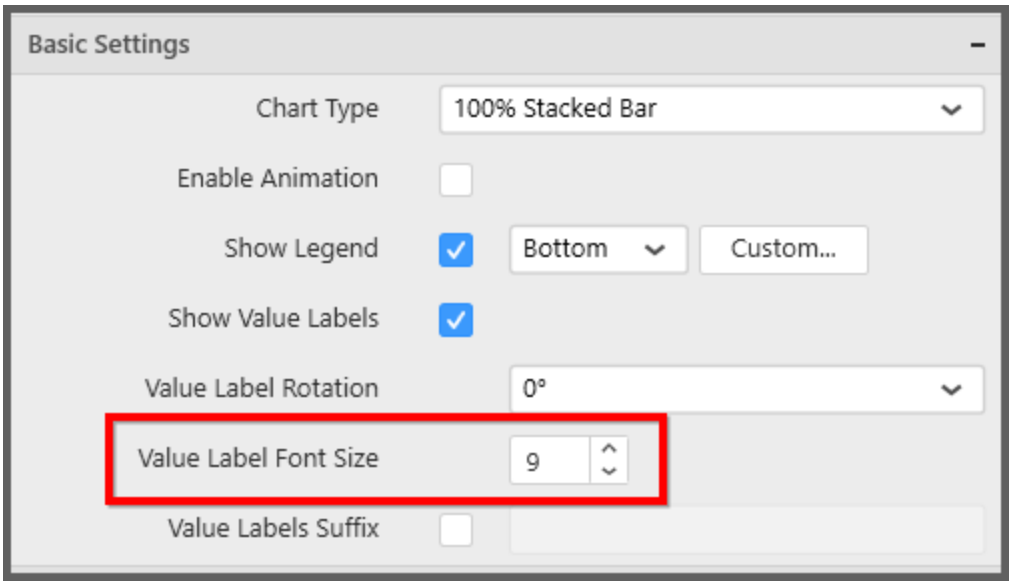
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



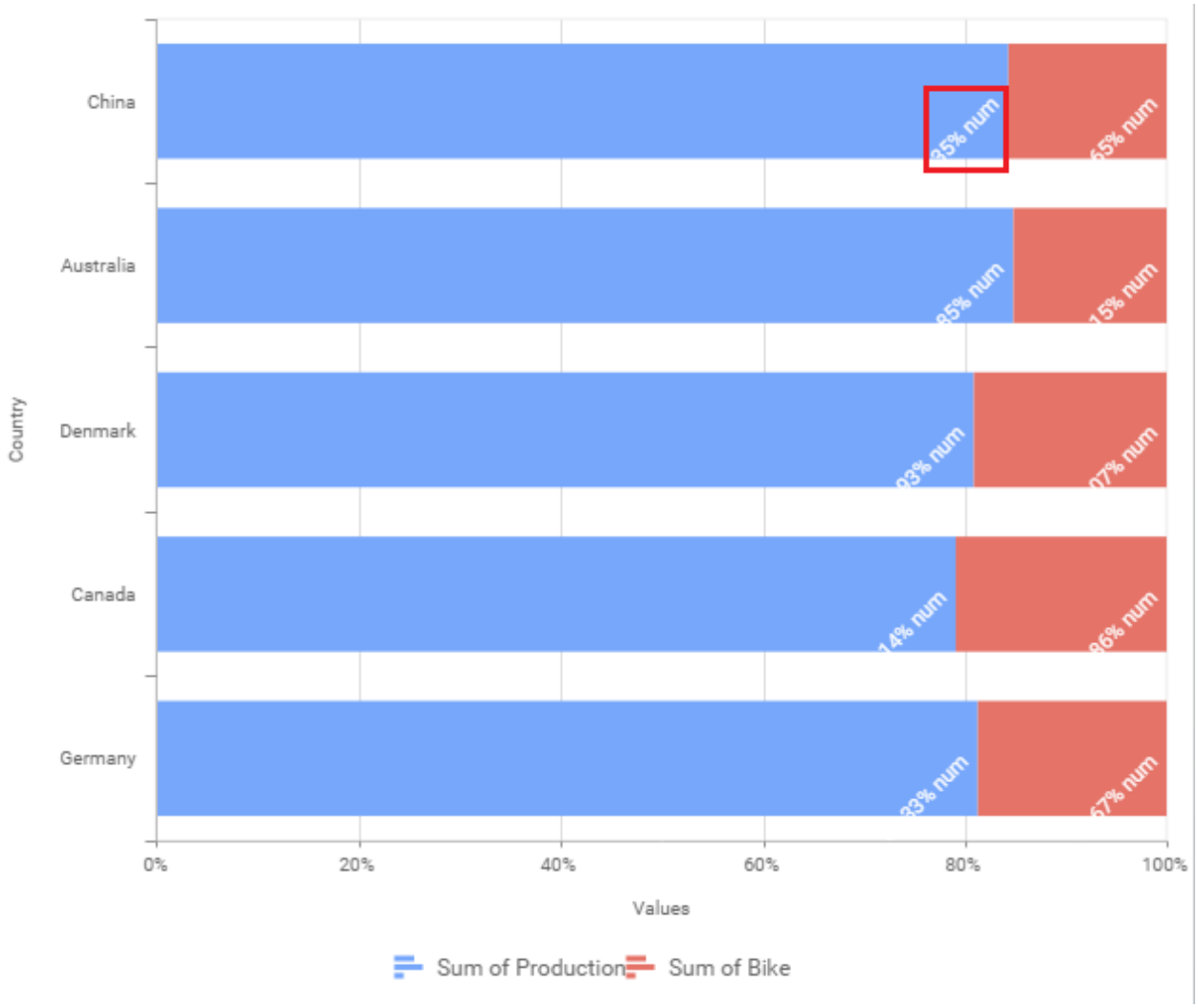
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

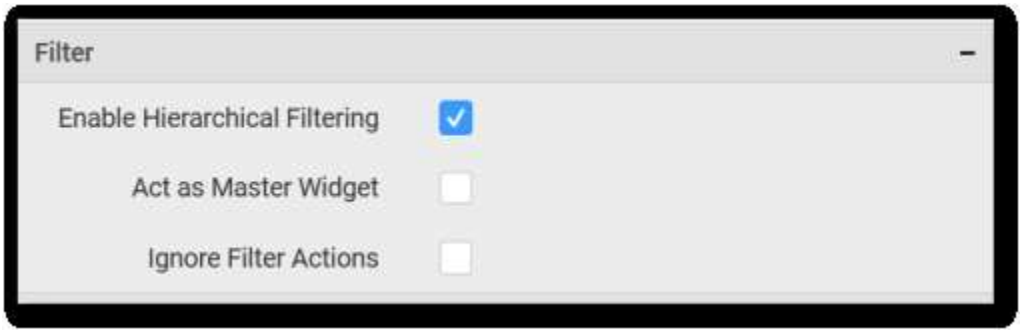


Value Labels Suffix

Allows you to set suffix to the value/data labels.



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

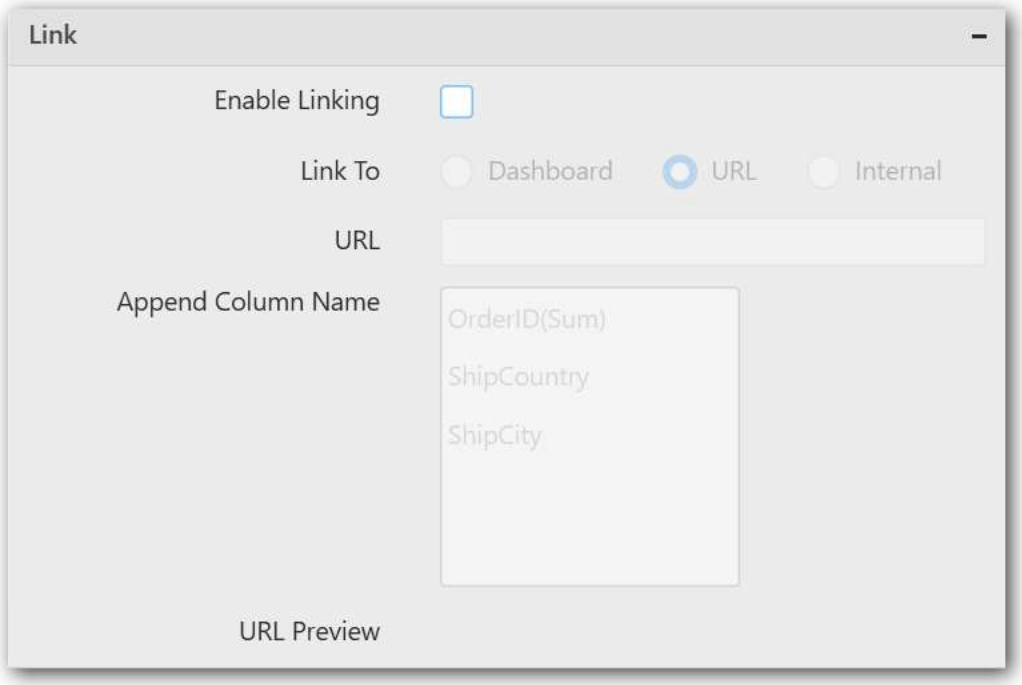
Act as Master Widget

This allows you to define this 100% stacked bar chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

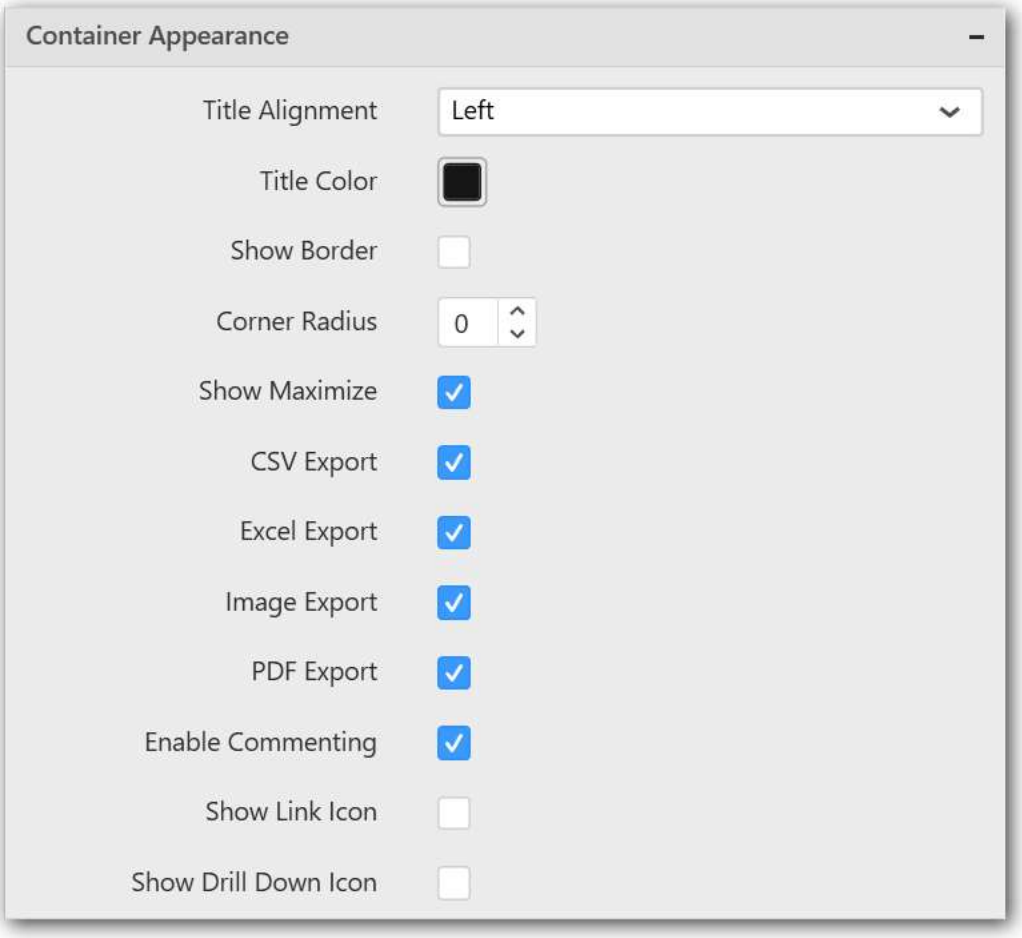
This allows you to define this 100% stacked bar chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this 100% stacked bar chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this 100% stacked bar chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this 100% stacked bar chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

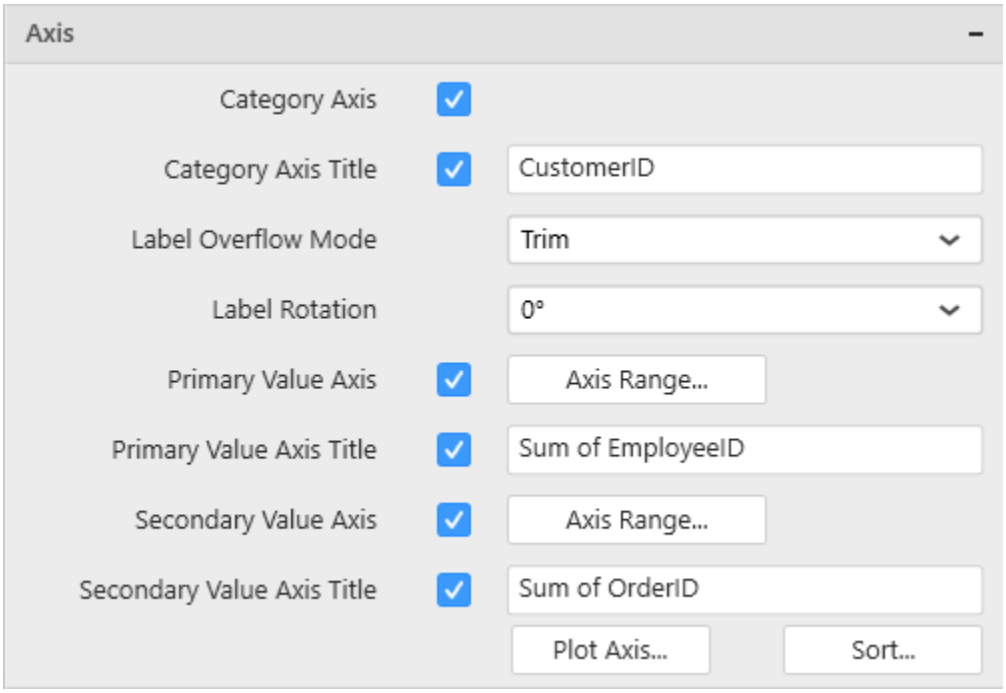
Image Export

This allows you to enable/disable the image export option for this 100% stacked bar chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

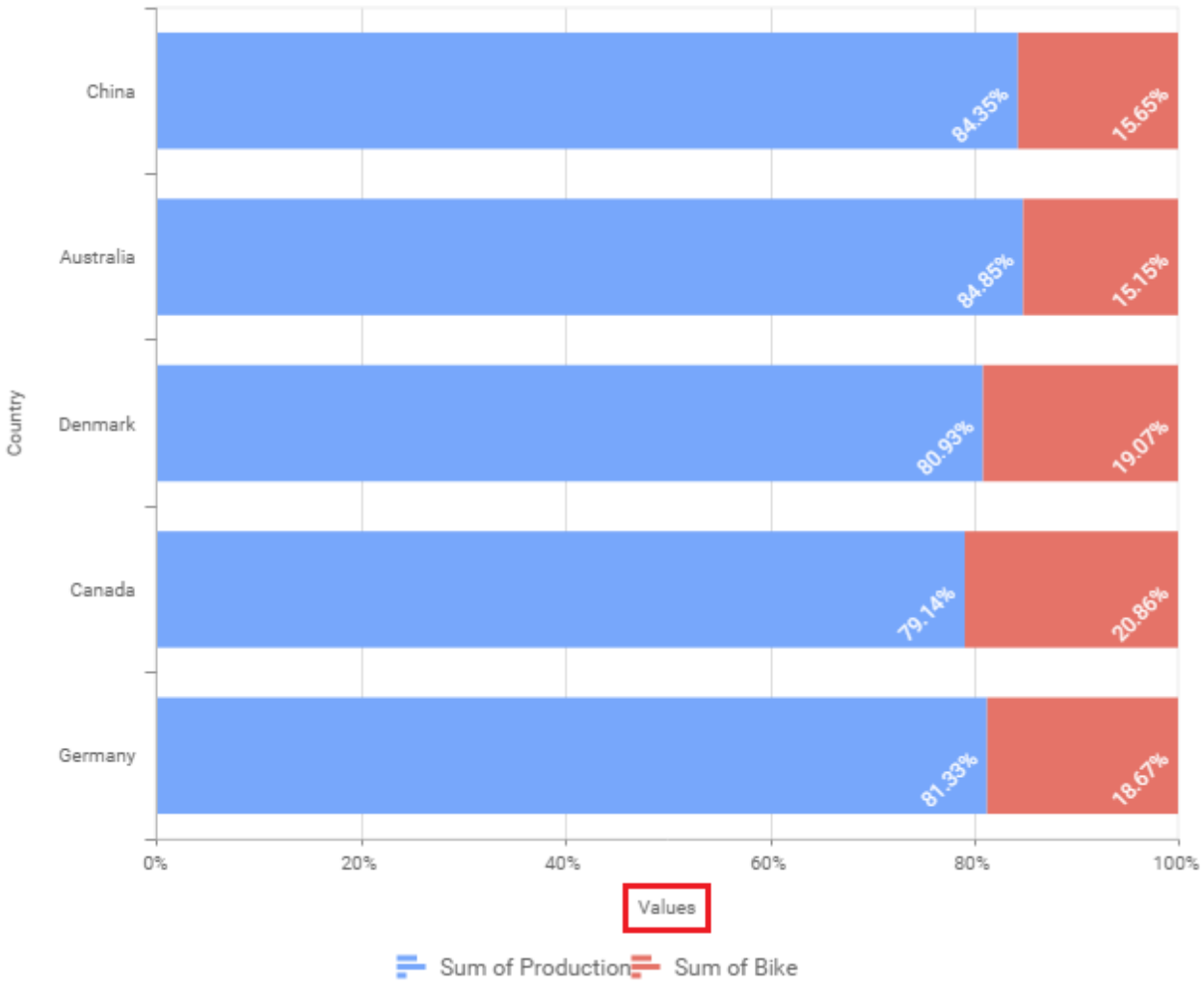
Axis Settings



This section allows you to customize the axis settings in chart.

Category Axis

This allows you to enable/edit the Category Axis title. It will reflect in chart area x-axis name.



Category Axis Title

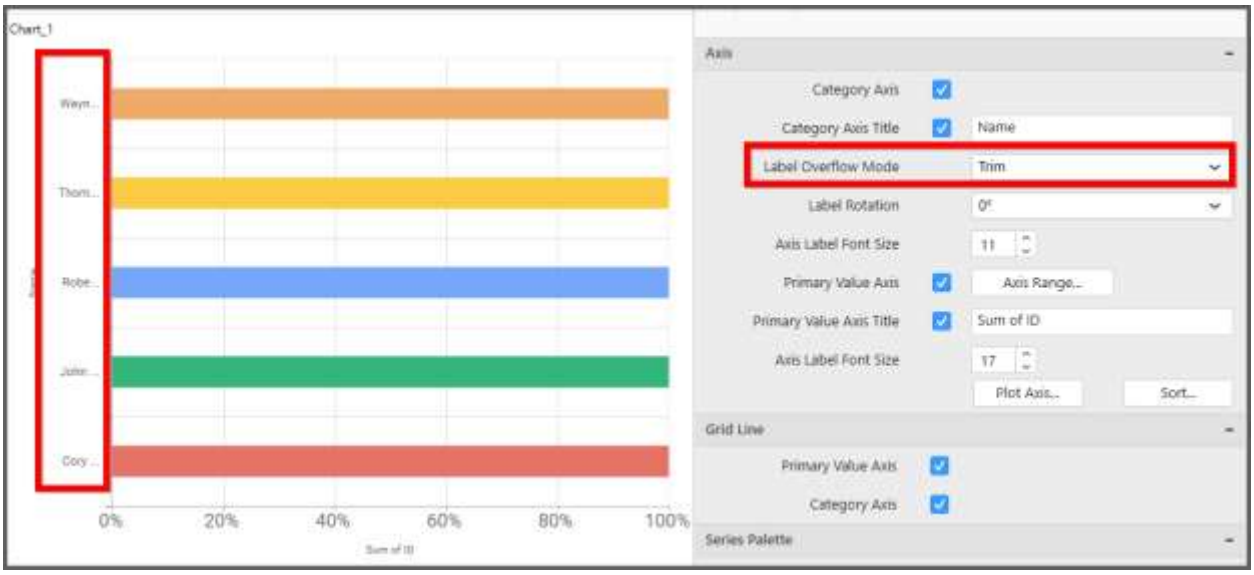
This allows you to toggle the visibility of Category axis title.

Label overflow mode

This allows you to handle the display mode of the overlapping labels.

Trim

This option trims the end of overlapping label in the axis.

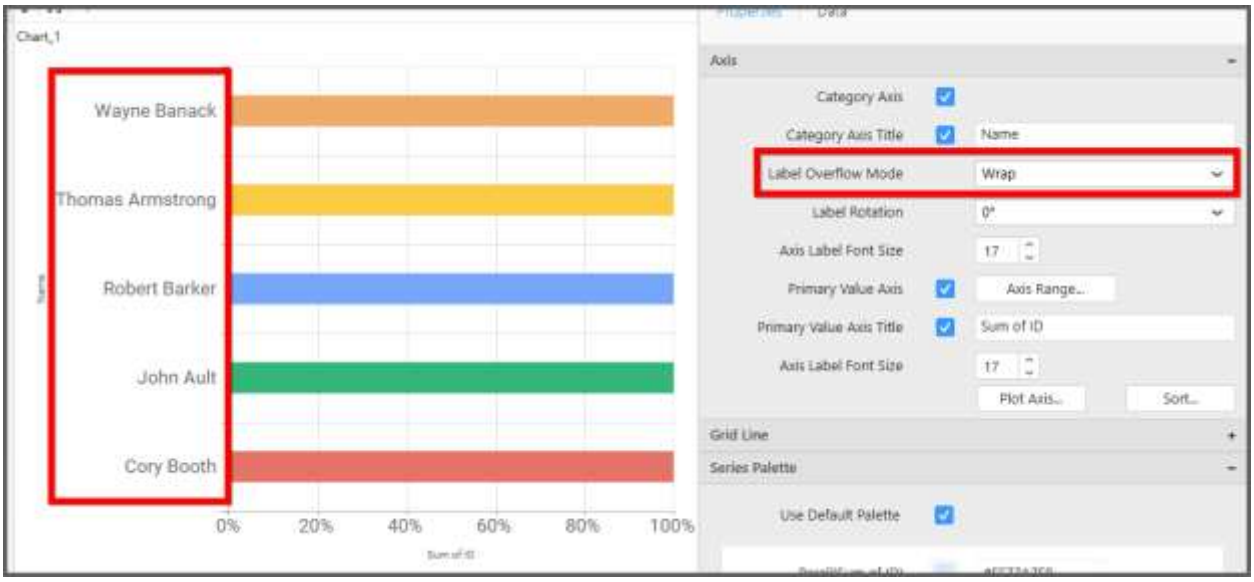


Hide

This option hides the overlapping label in the axis.

Wrap

This option wraps the lengthy label text in the axis.

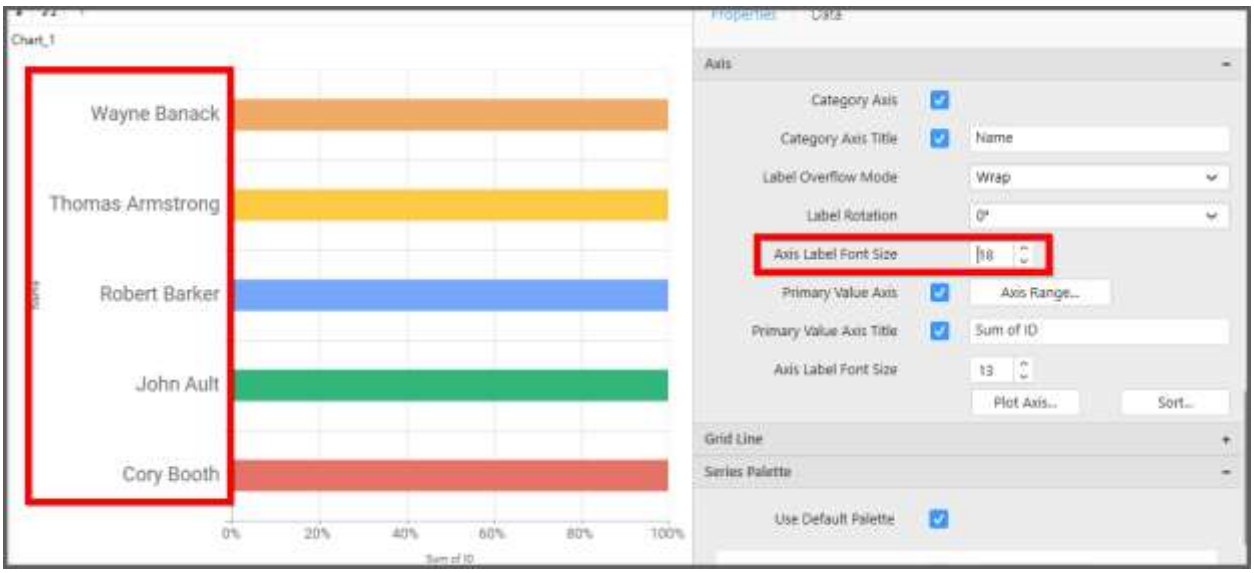


Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

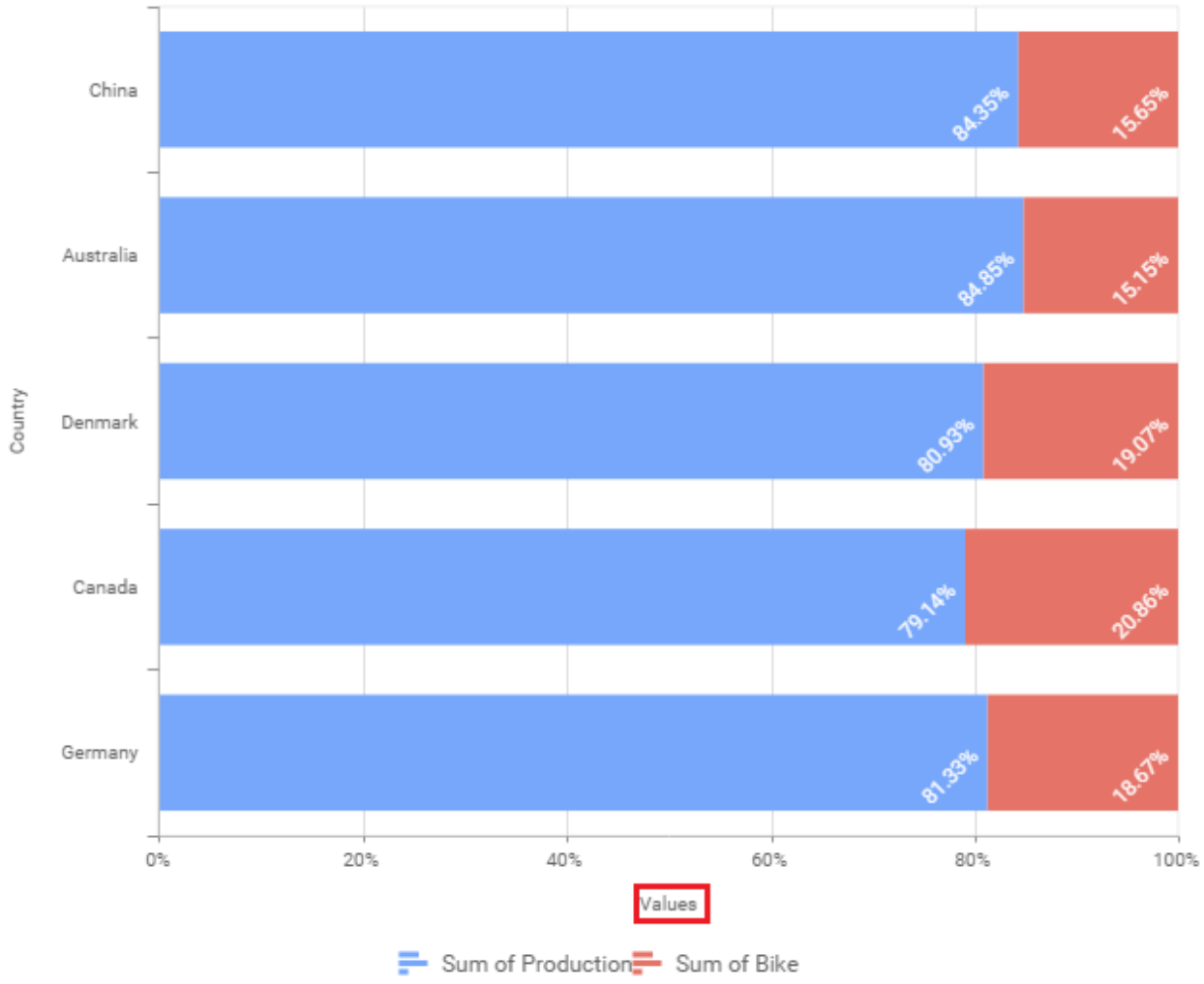
Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.



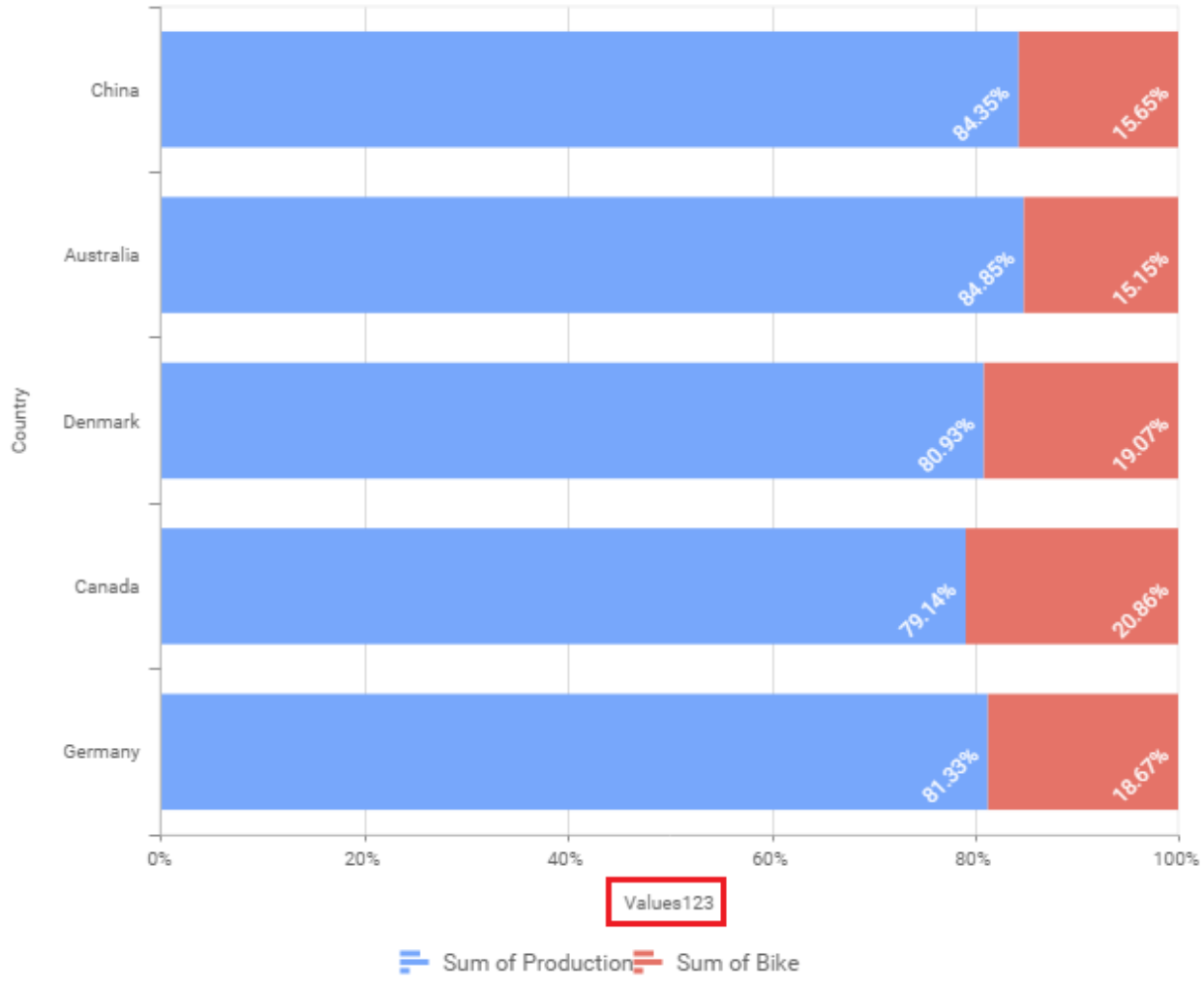
Primary Value Axis

This allows you to enable/edit the Primary Value Axis title. It will reflect in chart area axis.



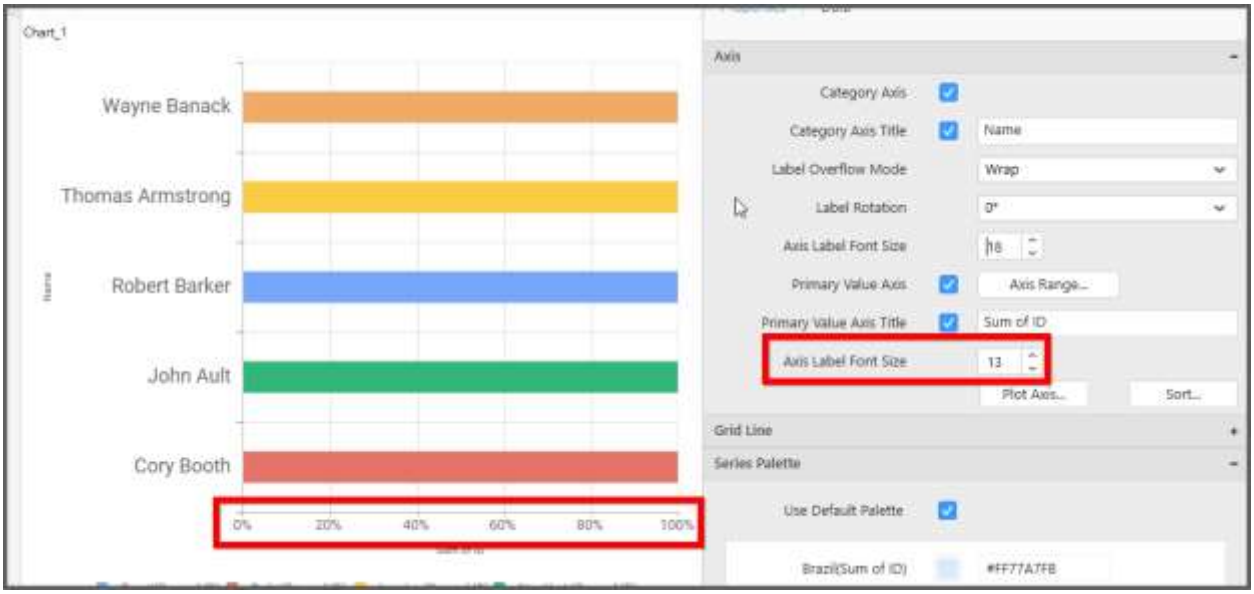
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

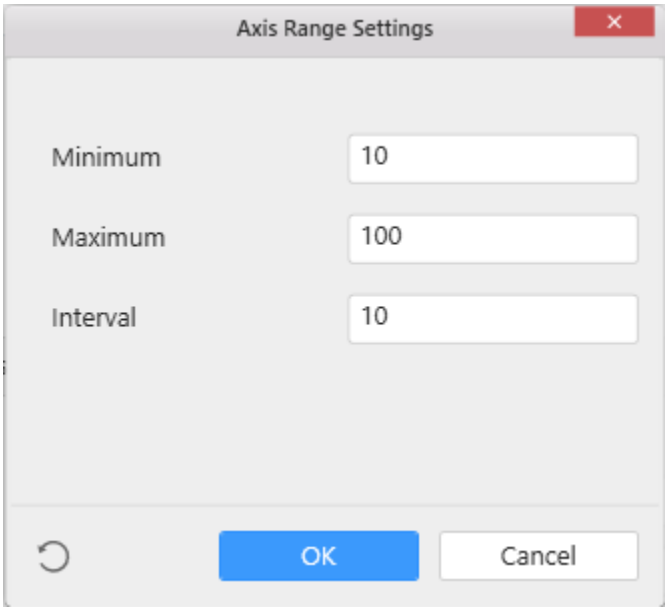


Primary Value Axis Range

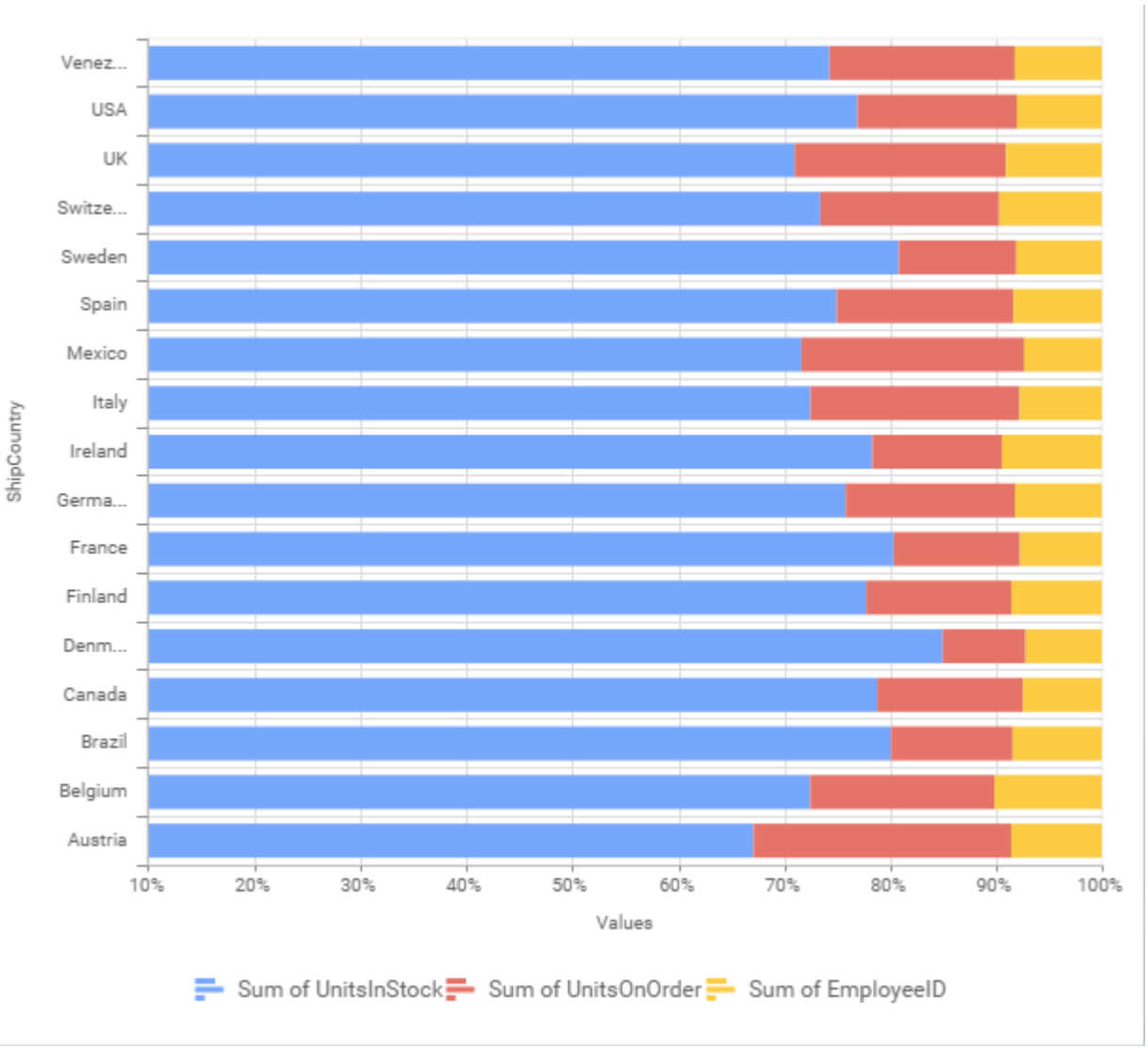
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.

Secondary Value Axis Title

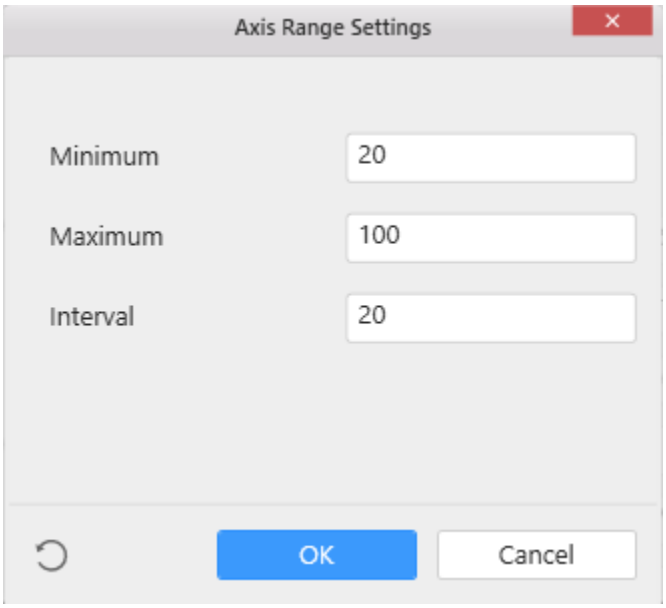
This allows you to toggle the visibility of secondary value axis title.

Axis Label Size

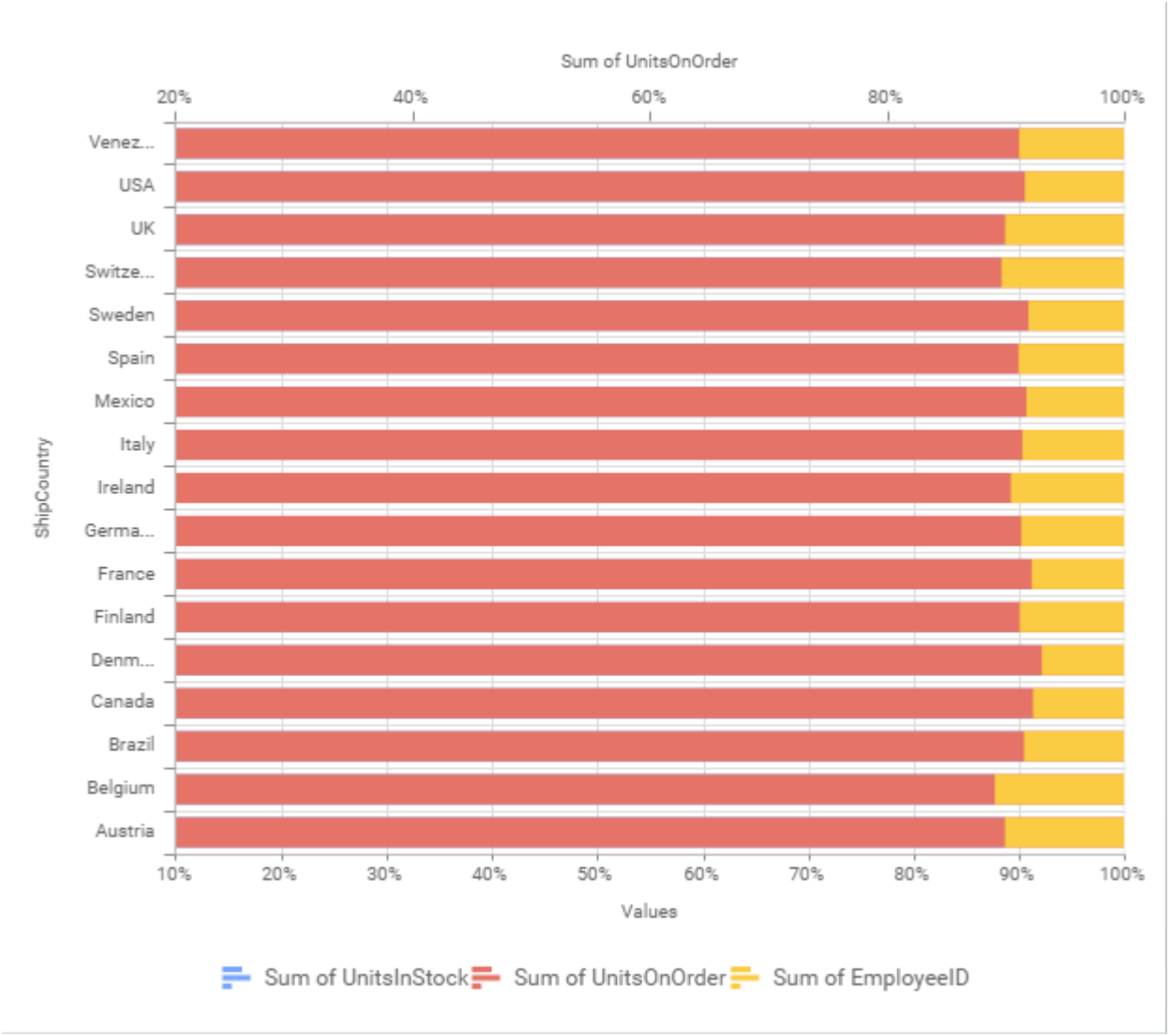
This allows you to increase or decrease the font size of the secondary axis label. Default font size for the secondary axis label was 10 pixels.

Secondary Value Axis Range

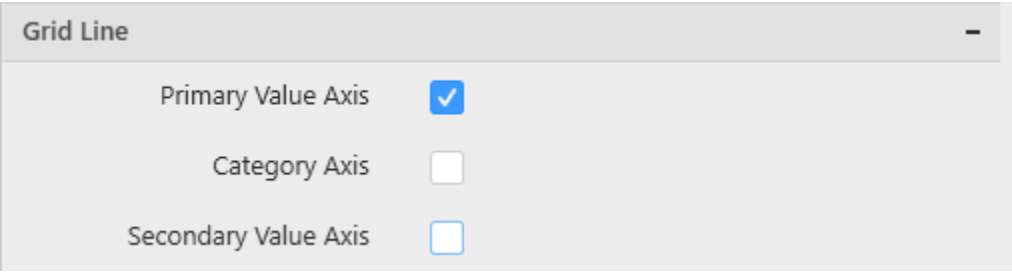
This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

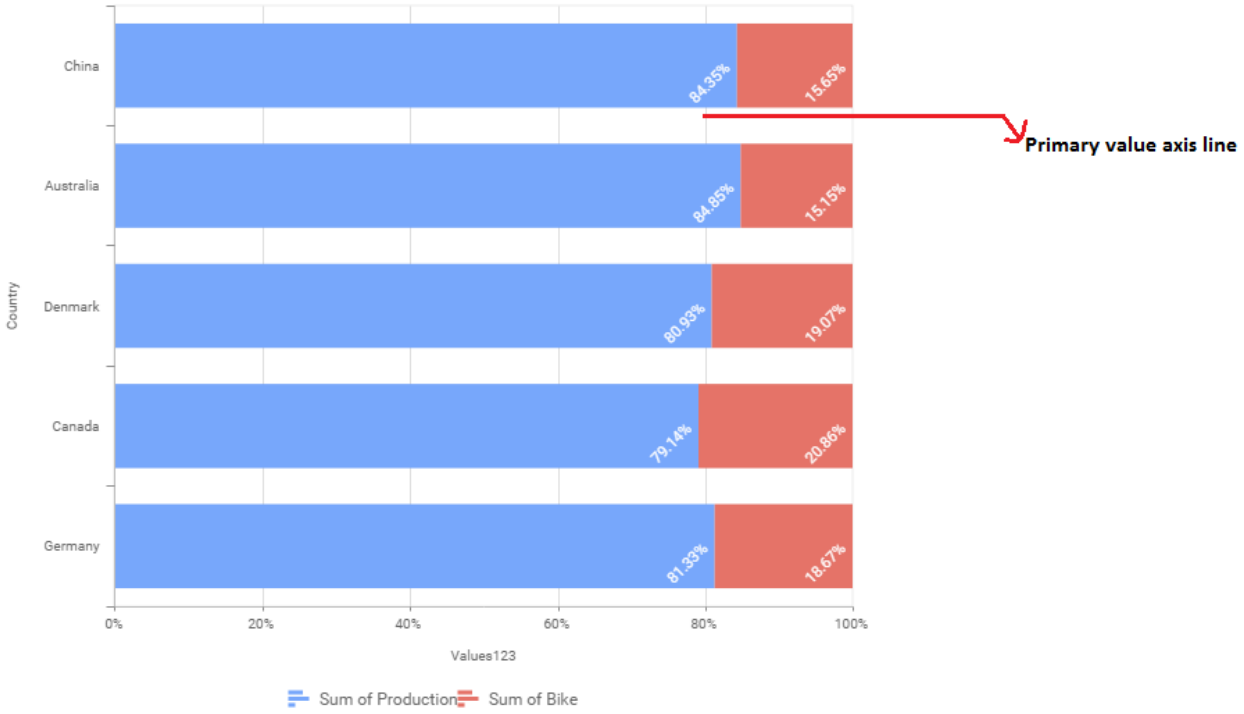


Grid Line Settings



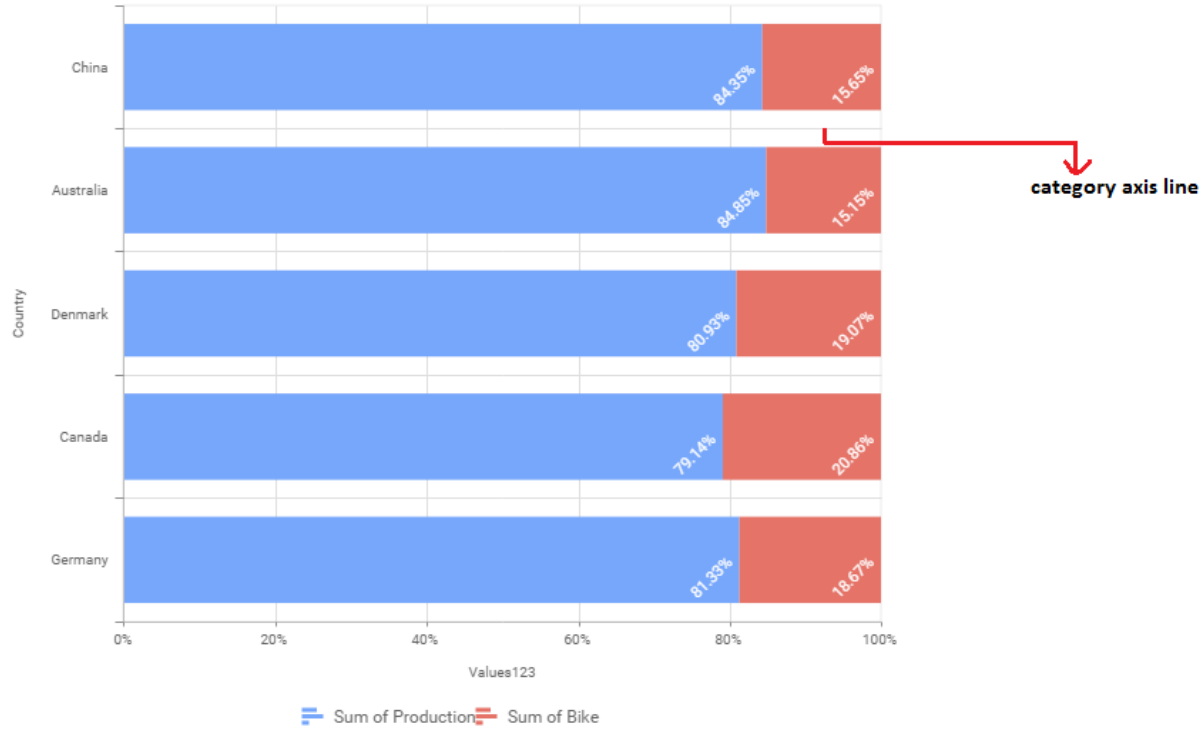
Primary value Axis

This allows you to enable the primary value axis' gridlines for the 100% stacked bar chart.



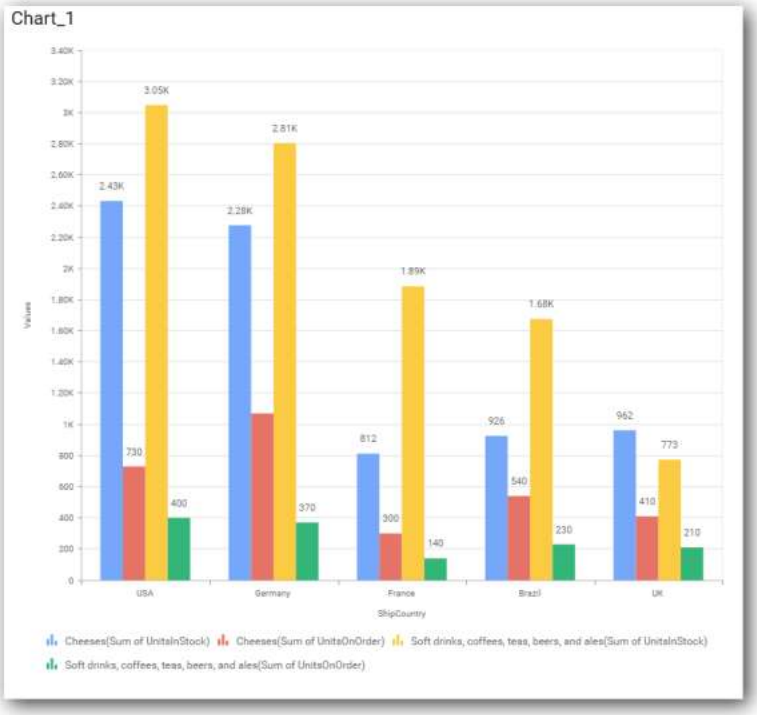
Category Axis

This allows you to define the visibility of Category axis' gridlines.



Column Chart

Column Chart allows you to compare values for a set of unordered items across categories through vertical bars ordered horizontally.

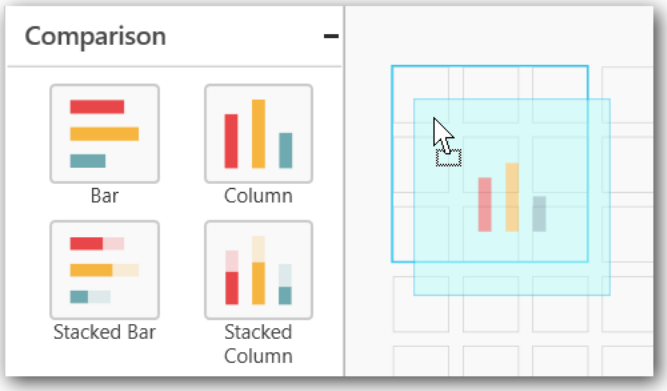


How to configure flat table data to Column Chart?

Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Follow the steps to configure data to column chart

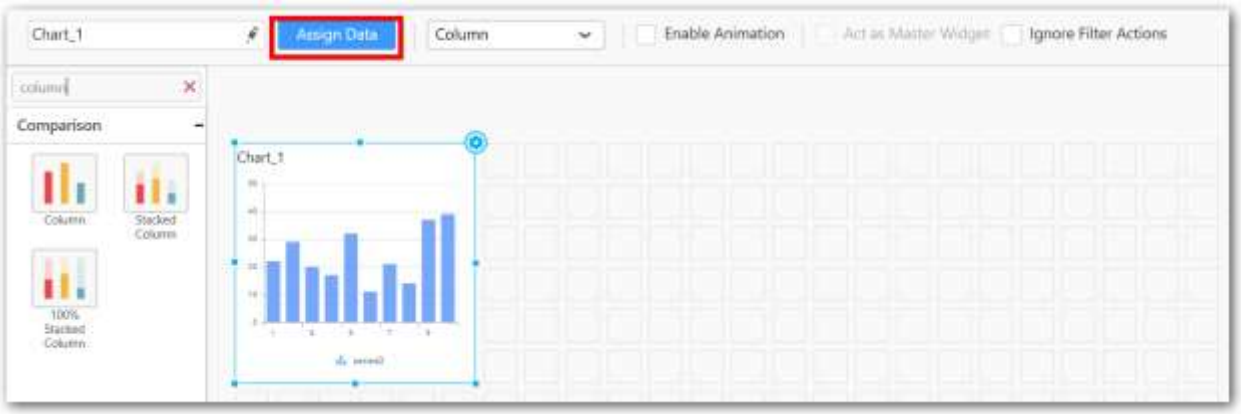
Drag and drop the column chart widget into canvas and resize it to your required size.



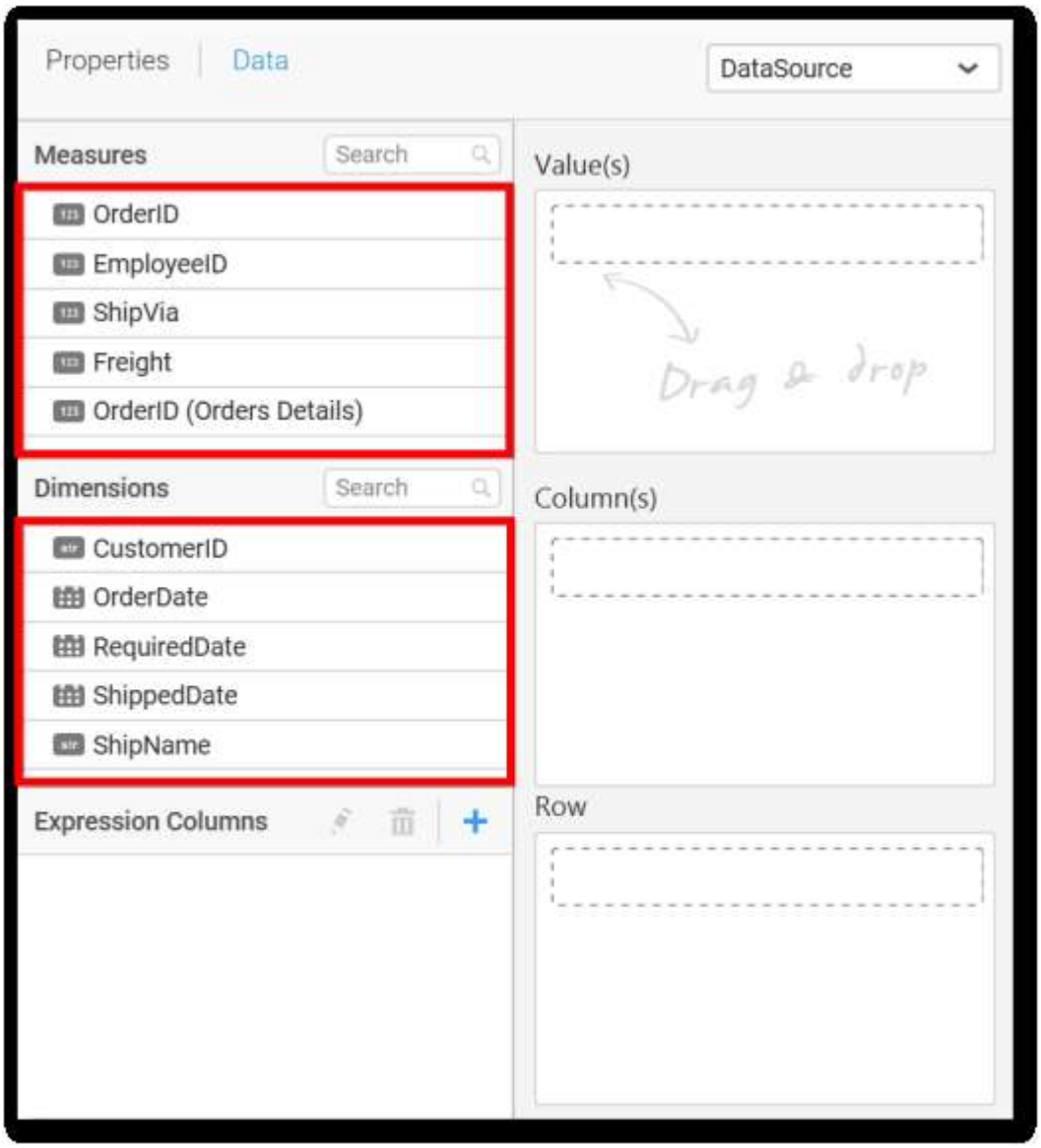
Connect to the data source.

Focus on the Column chart.

Click on Assign Data.



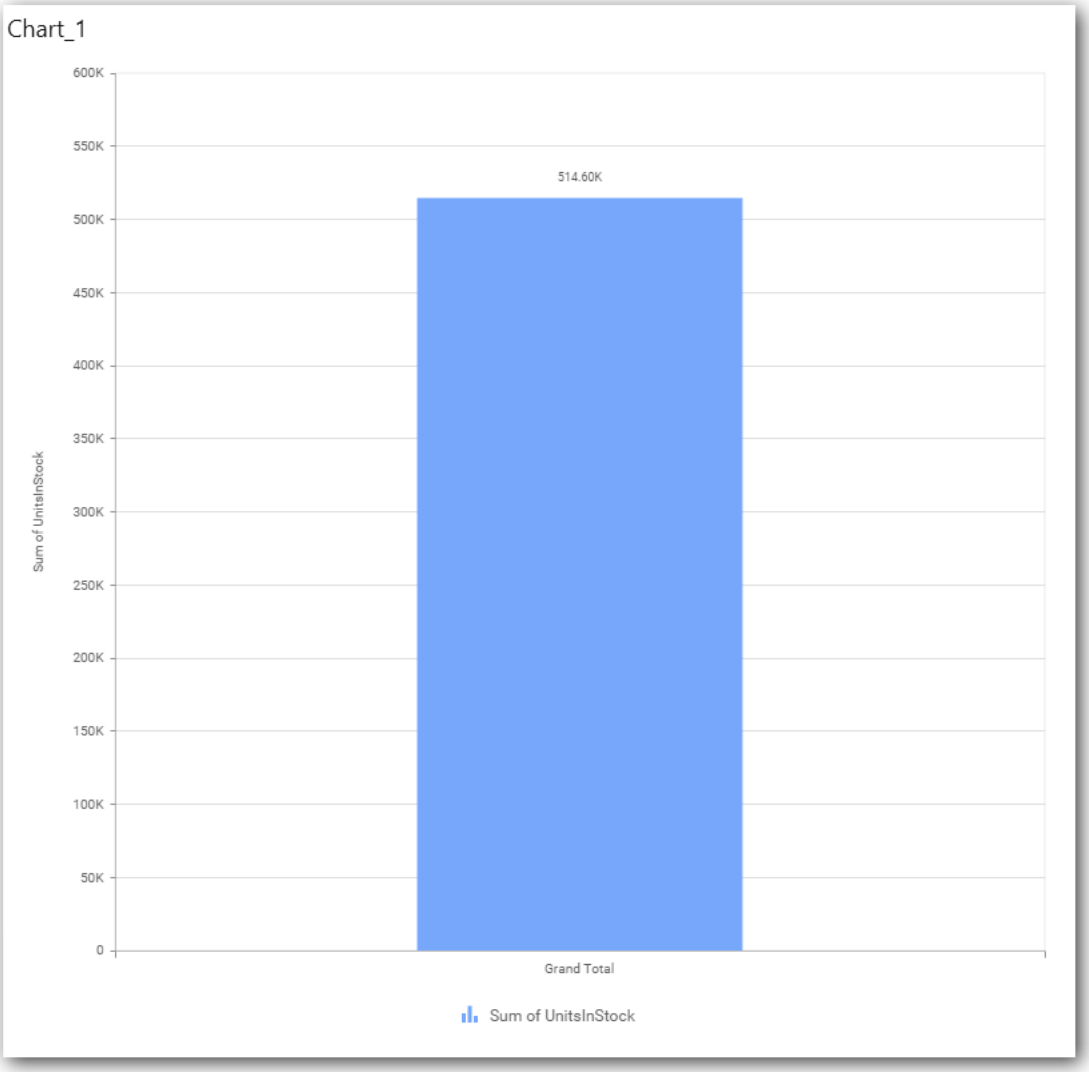
The data pane will be opened with available measures and dimensions in the connected data source.



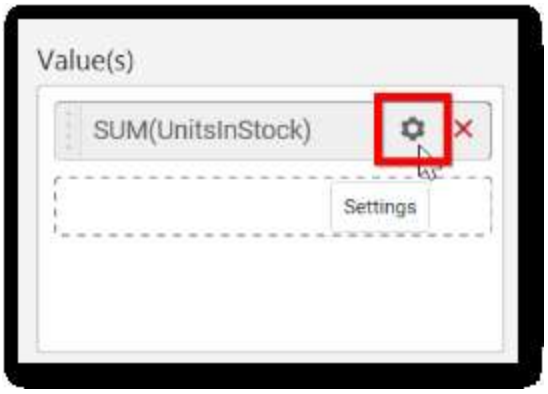
Assigning Value(s)

Drag and drop the Measure into Value.

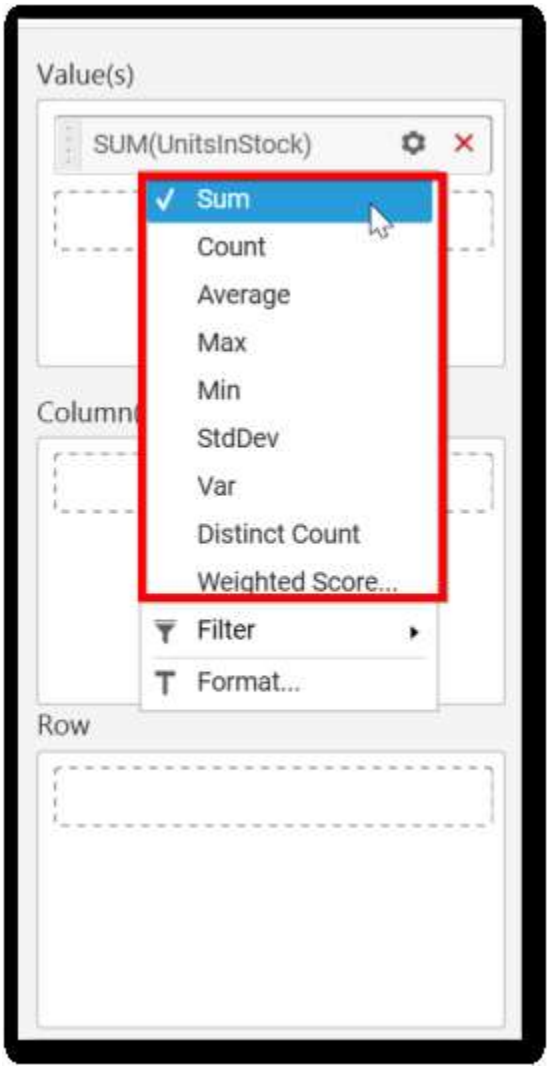
Now the chart will be rendered like this.



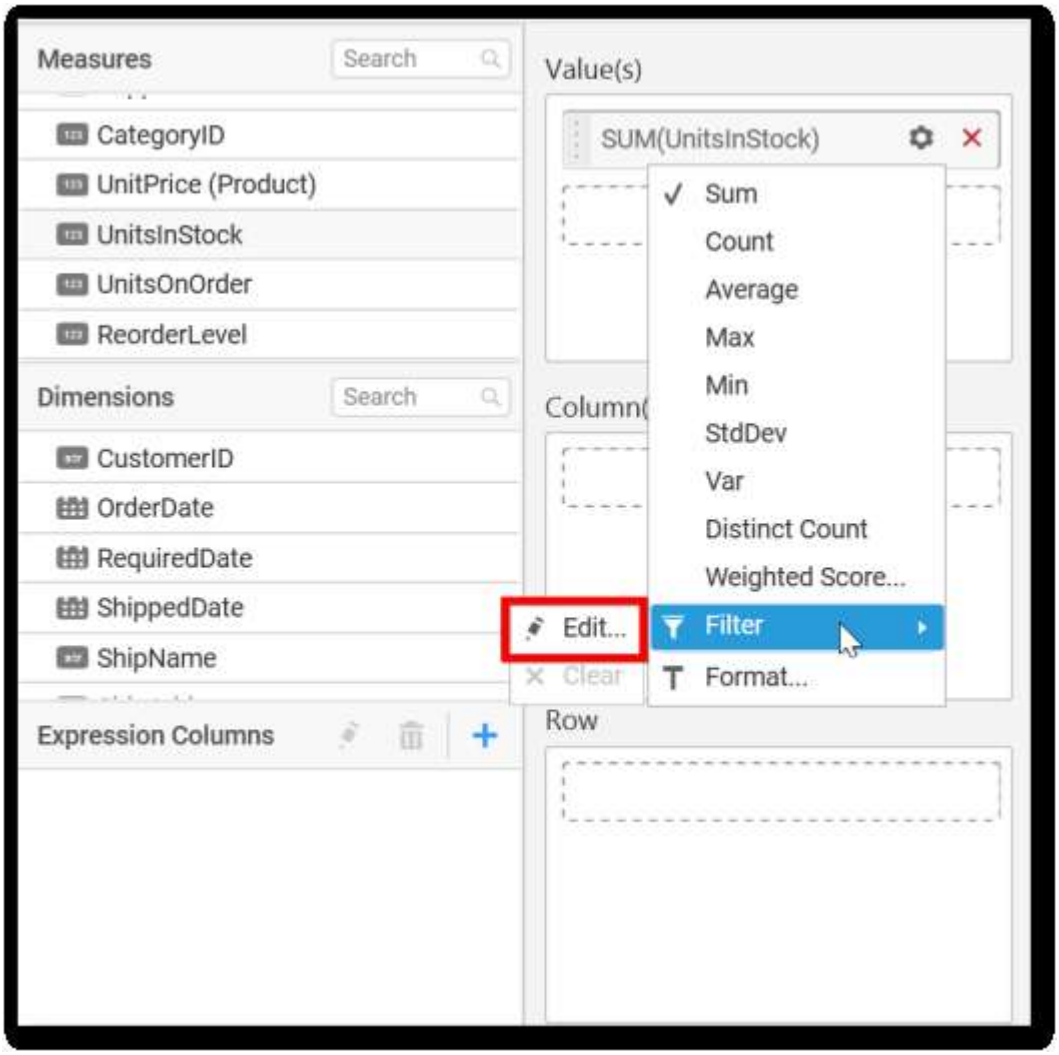
You can change the summary type of the value by clicking on Settings option.



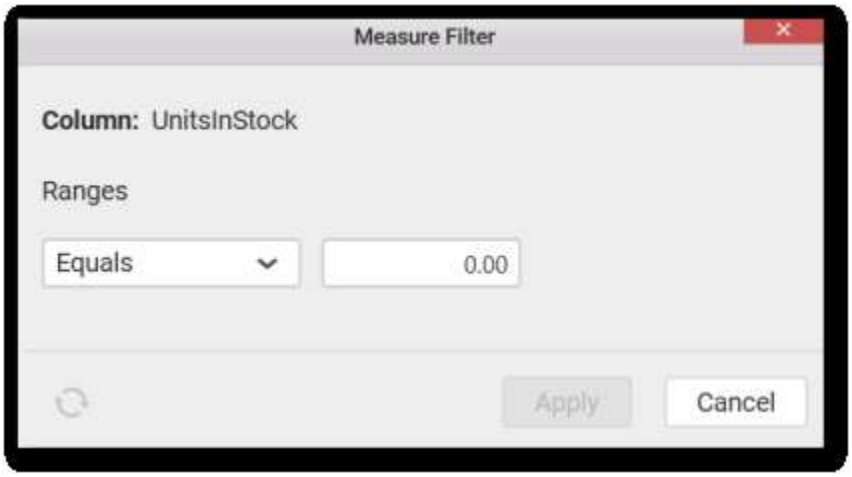
Select the required summary type from list.

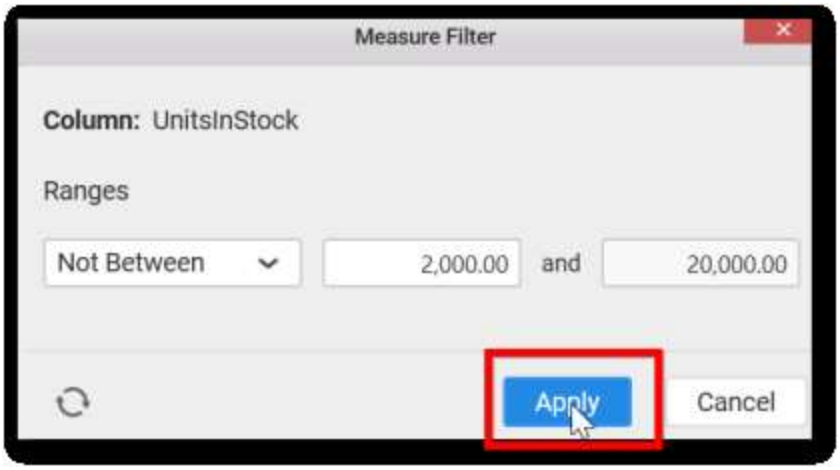


You can select what data to be displayed by choosing filter option.



The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.

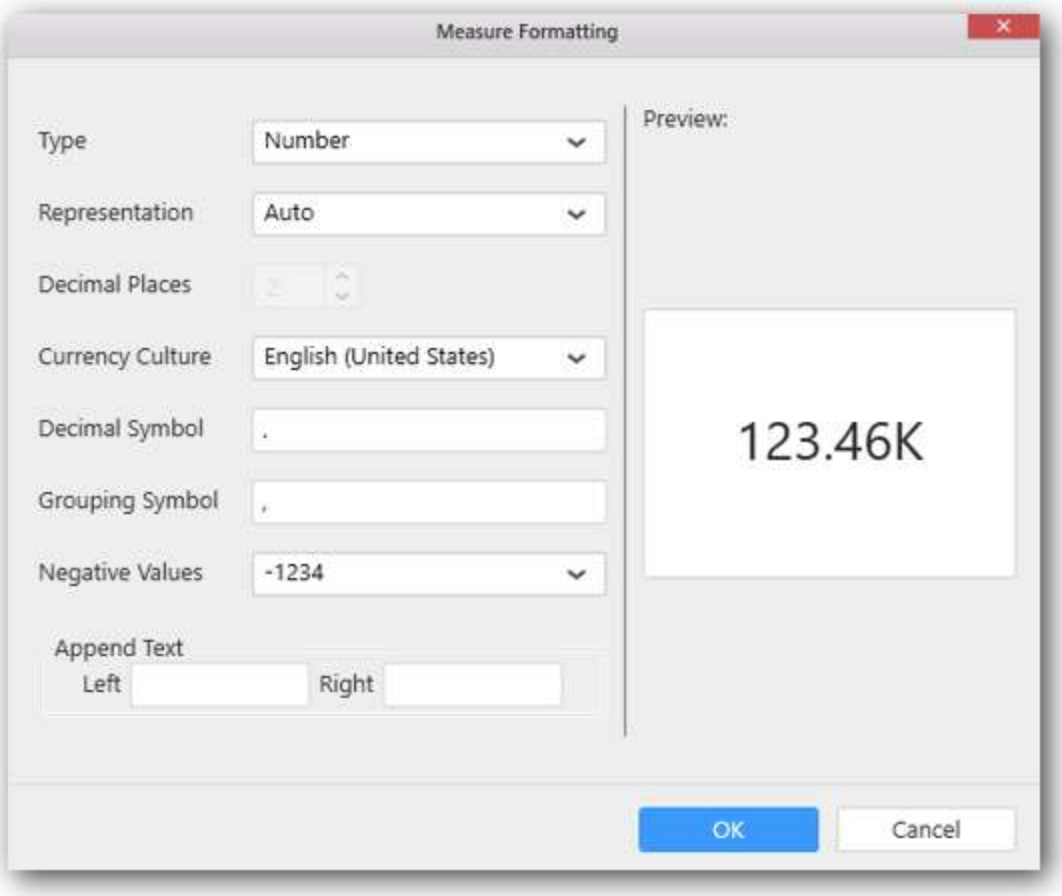




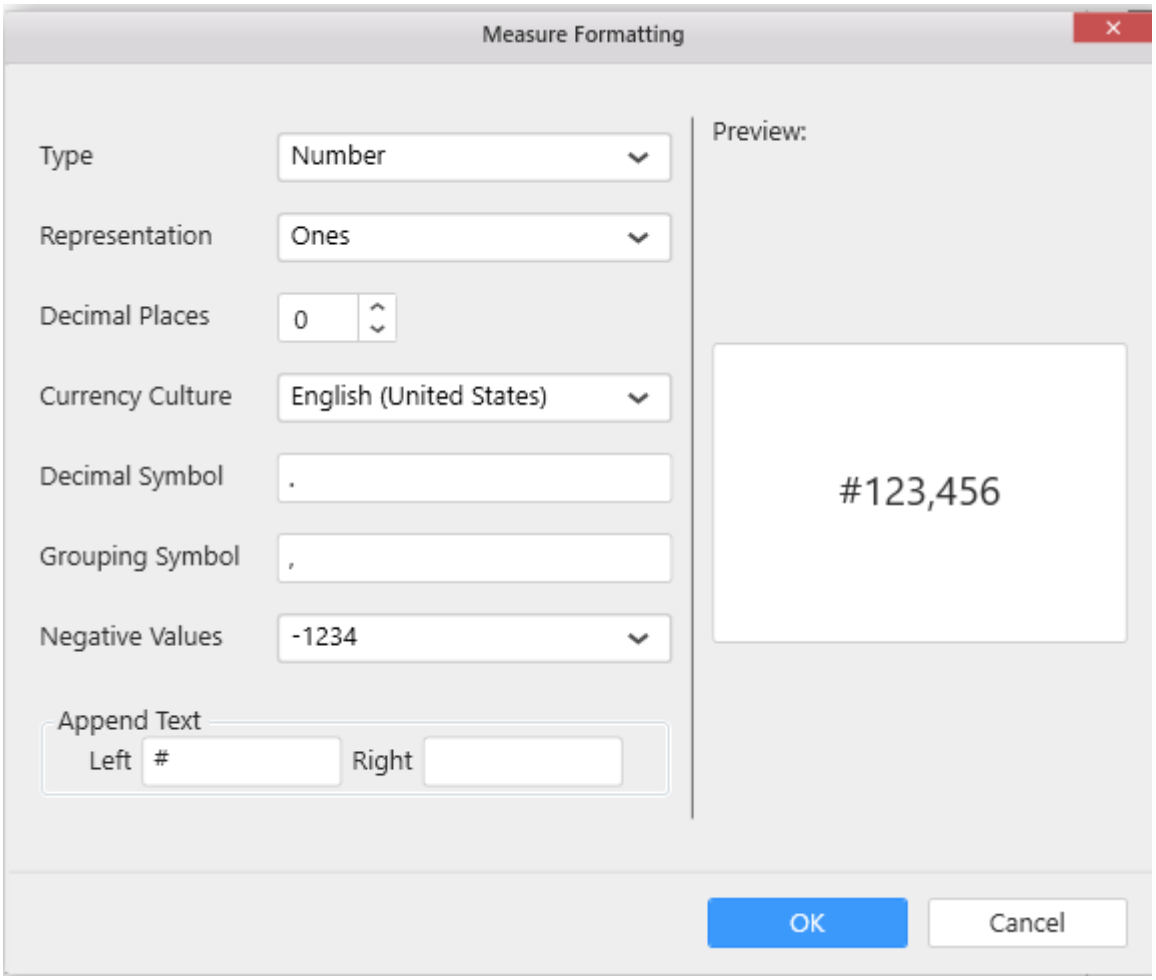
You can clear the filter.

You can Format the value

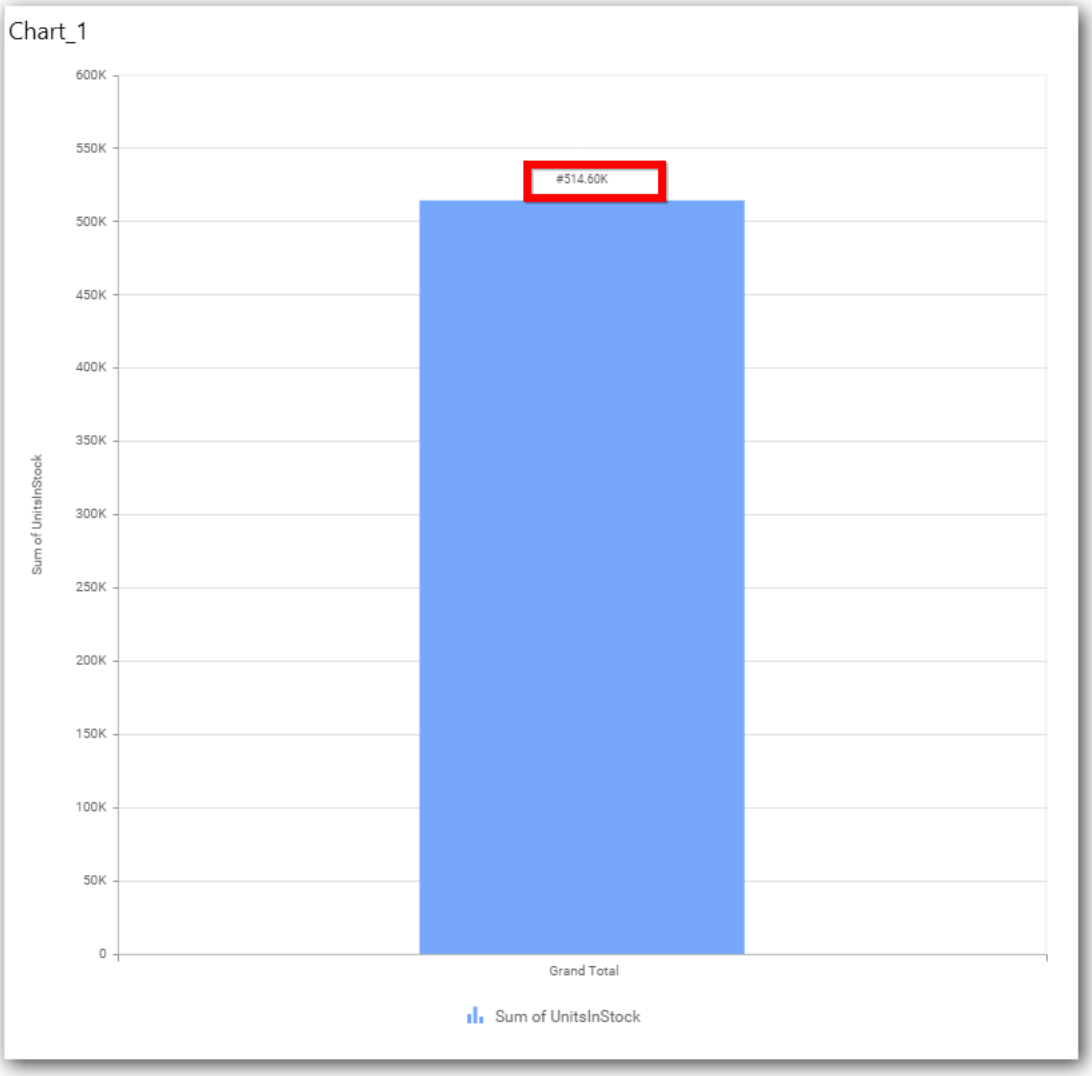
The format options will be shown.



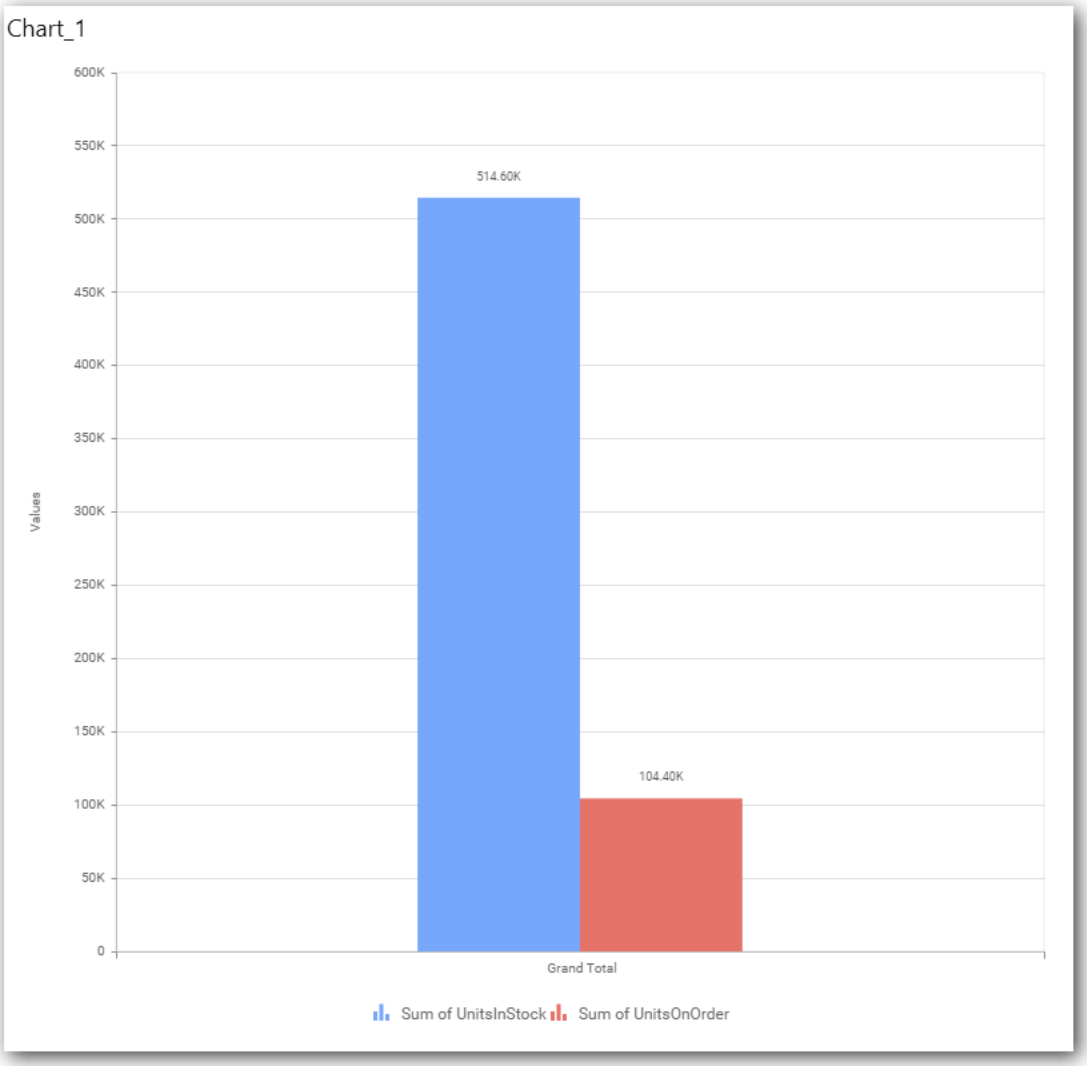
Choose the options you need and click OK.



Now the Chart will be rendered like this.



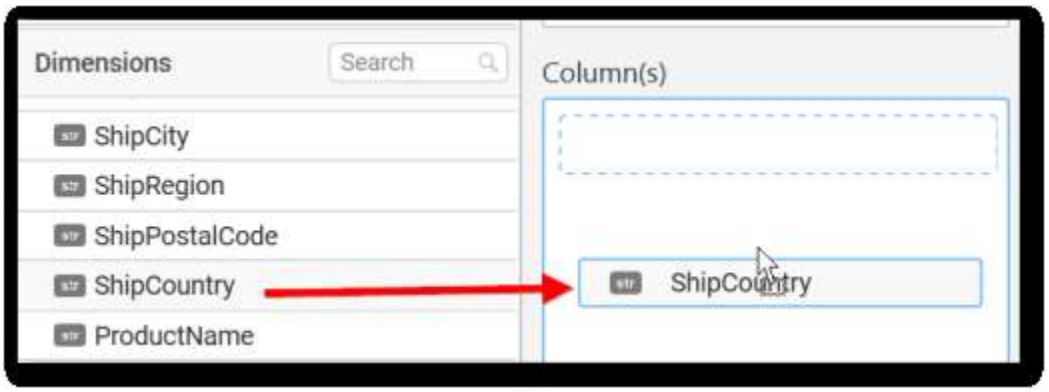
You can add more number values by drag drop the Measures into Value field.

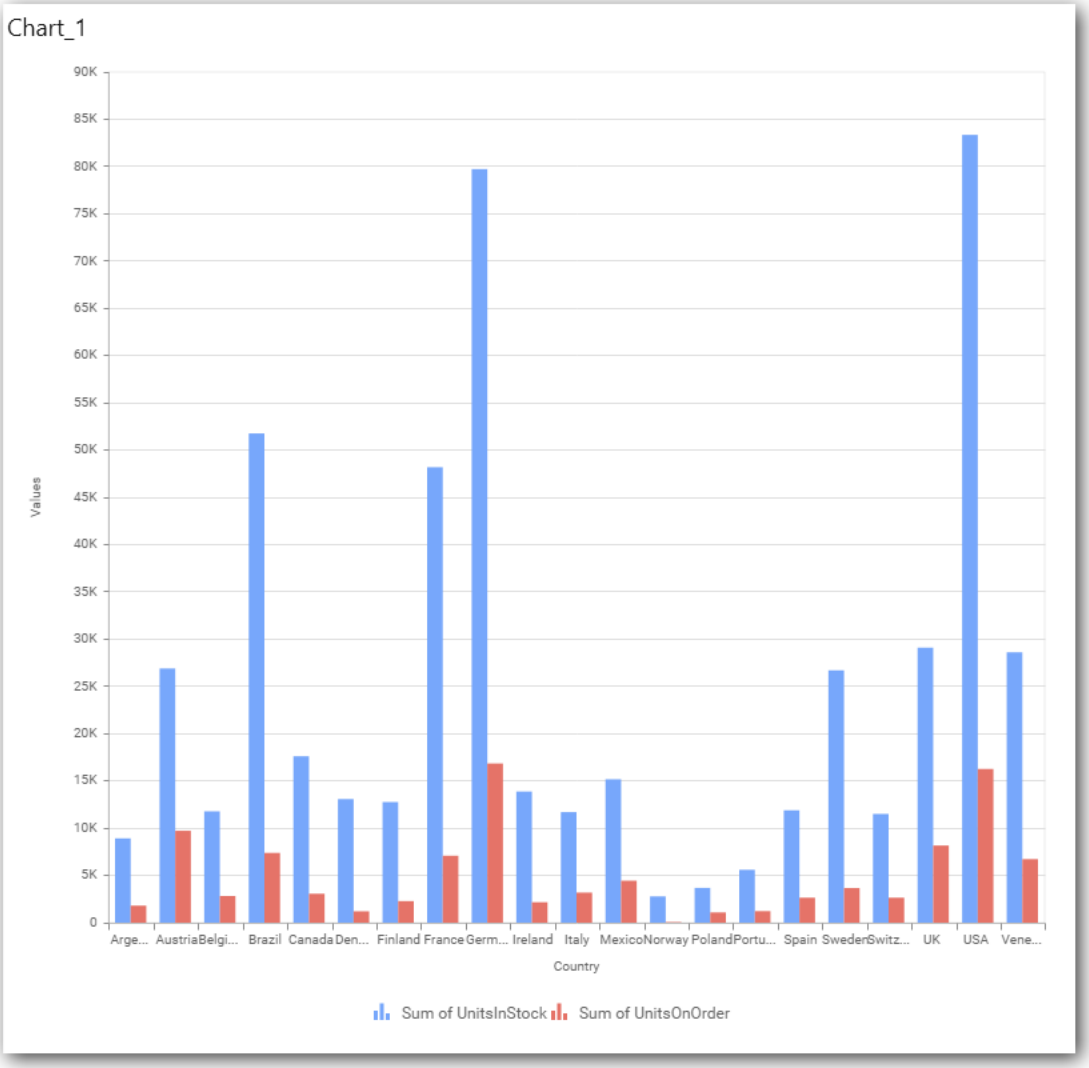


You can also add Dimensions and Columns to Value(s).

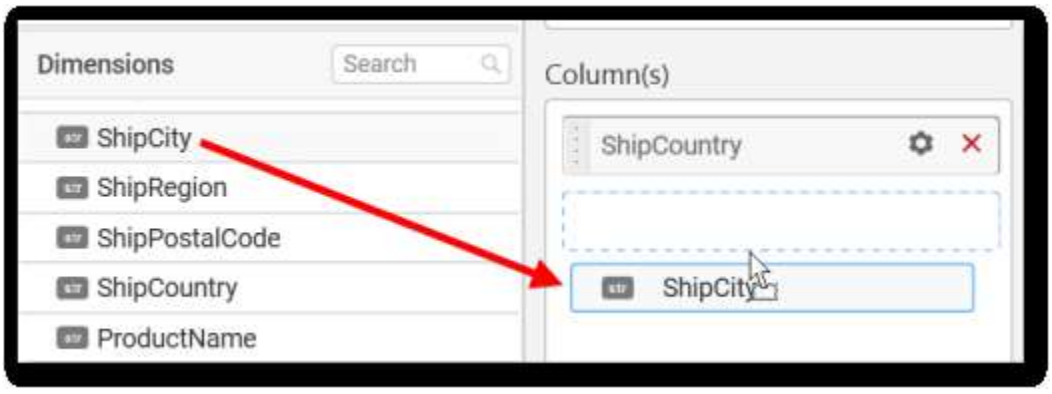
Assigning Column(s)

You can add the Dimension into Column field by drag and drop.

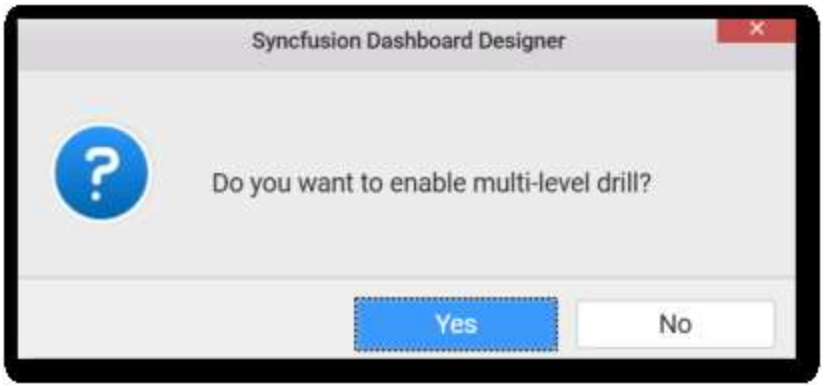




You have option to add more than one Column Value.

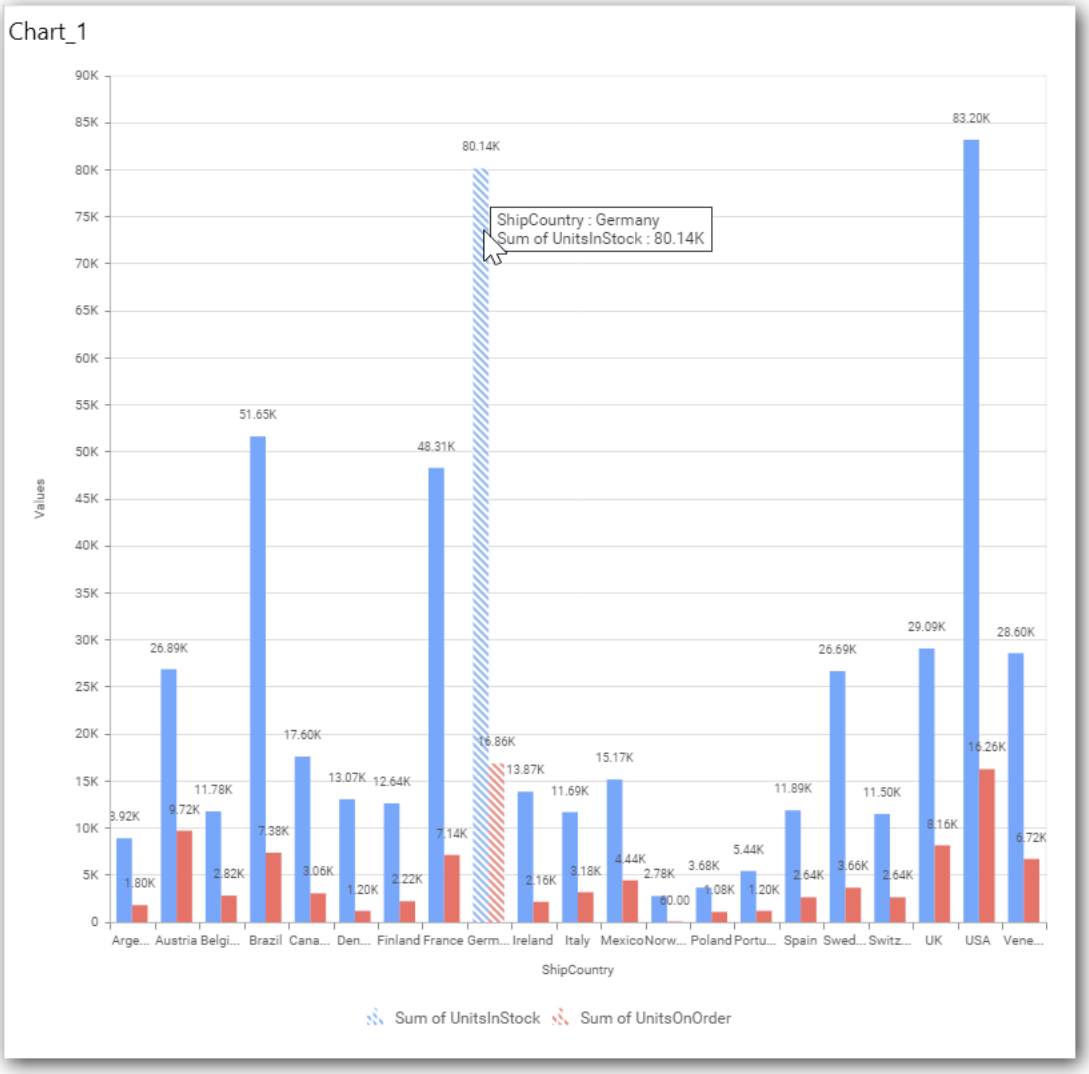


The following alert message will be shown.

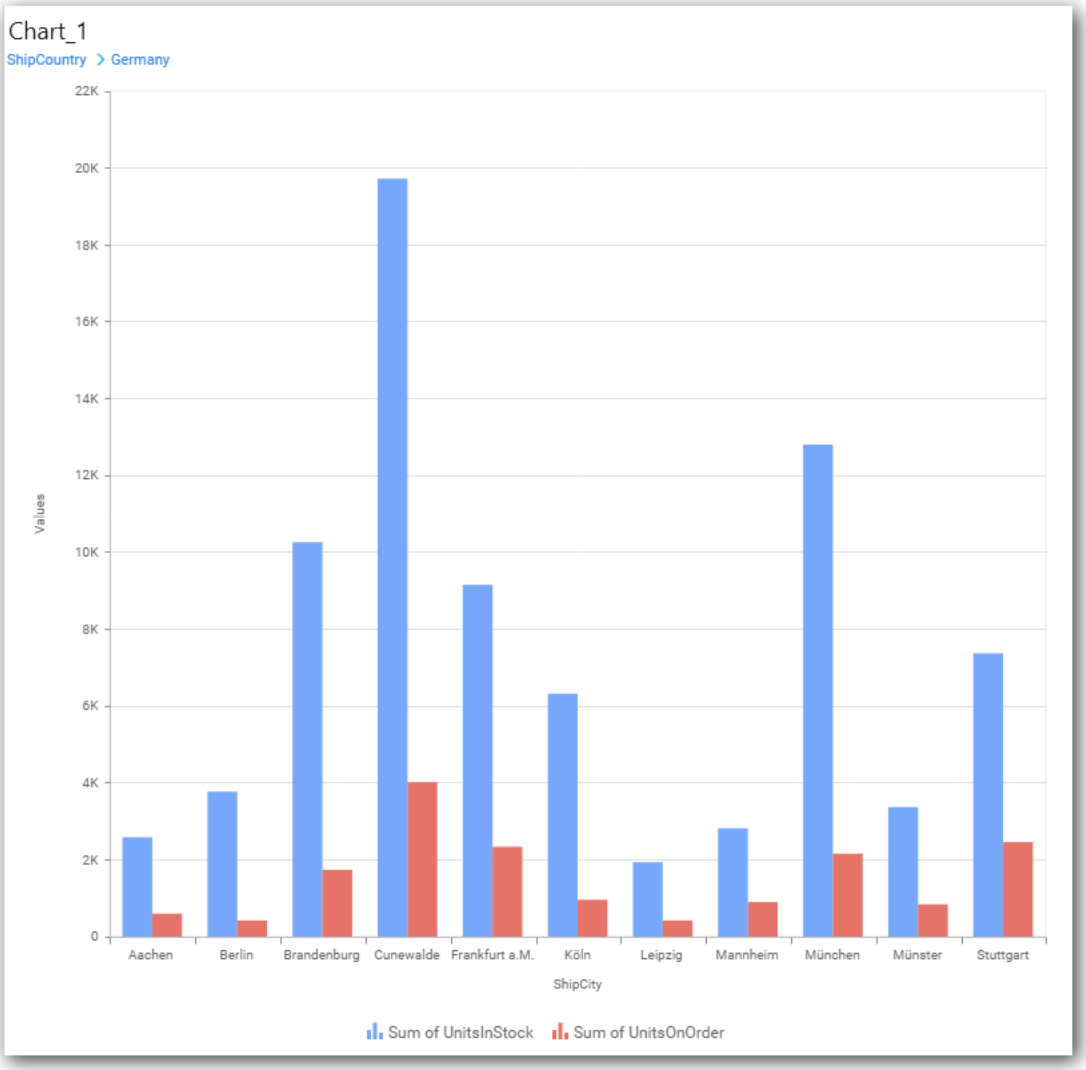


If you choose Yes Drill down option will be enabled.

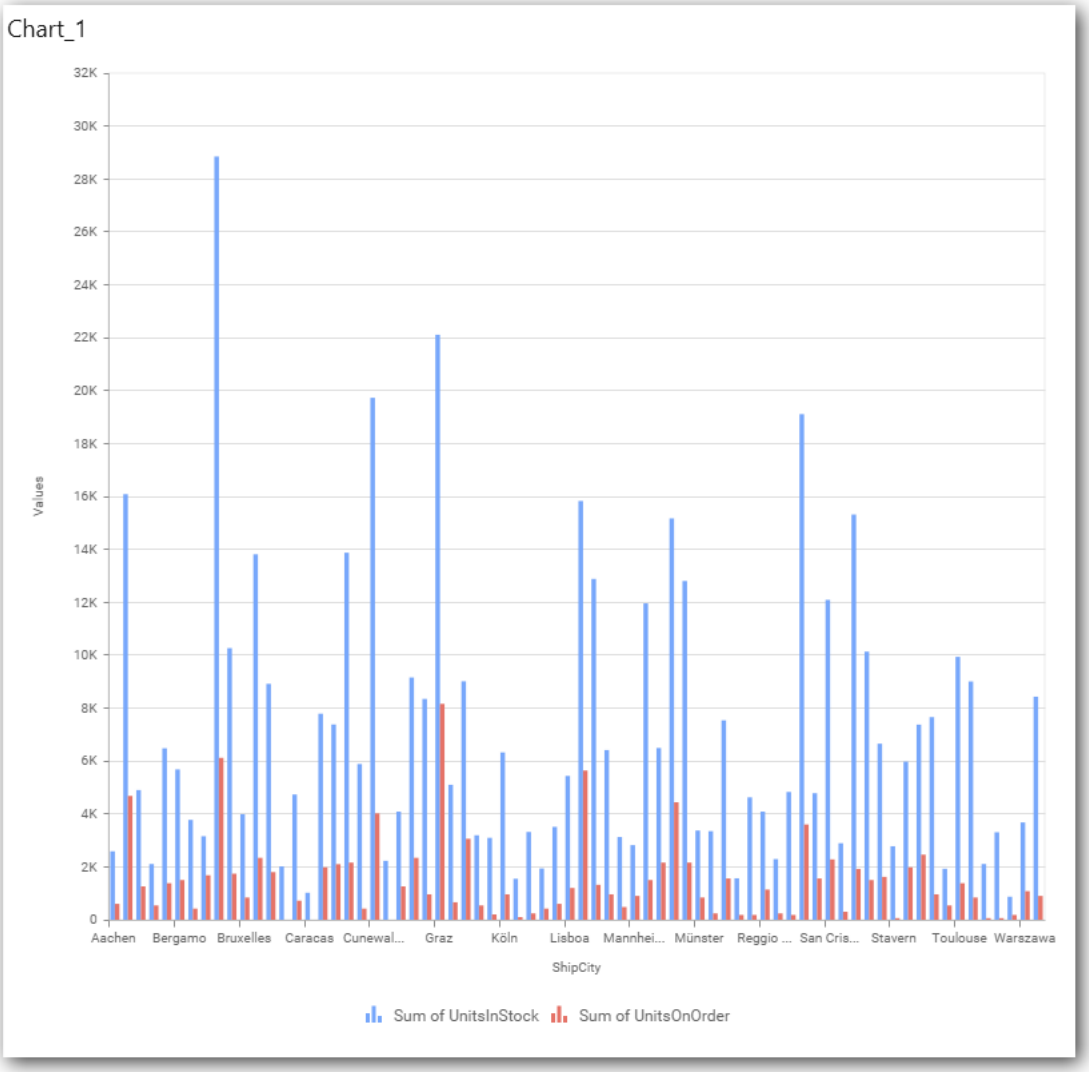
You can drill down the chart by clicking on the chart.



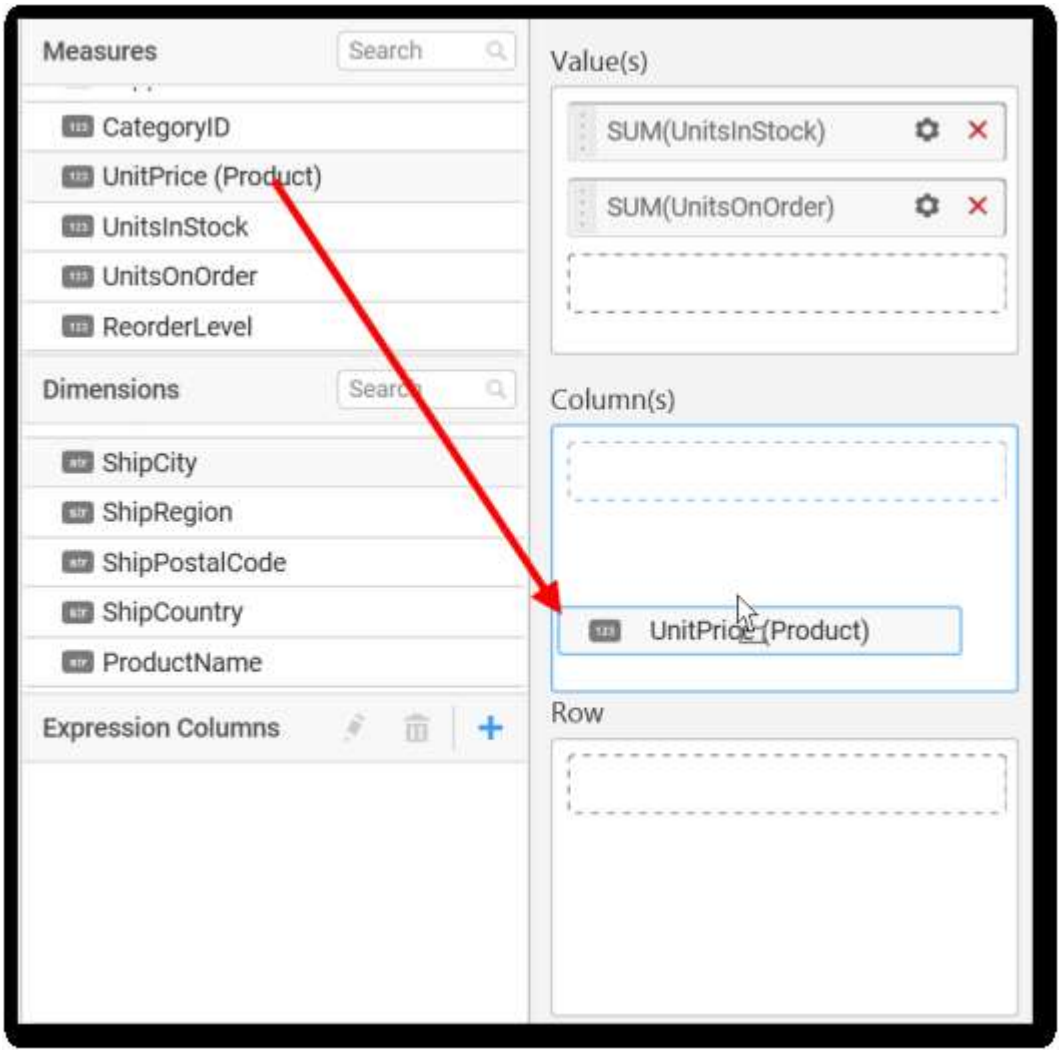
The drilled view of the chart is follows.

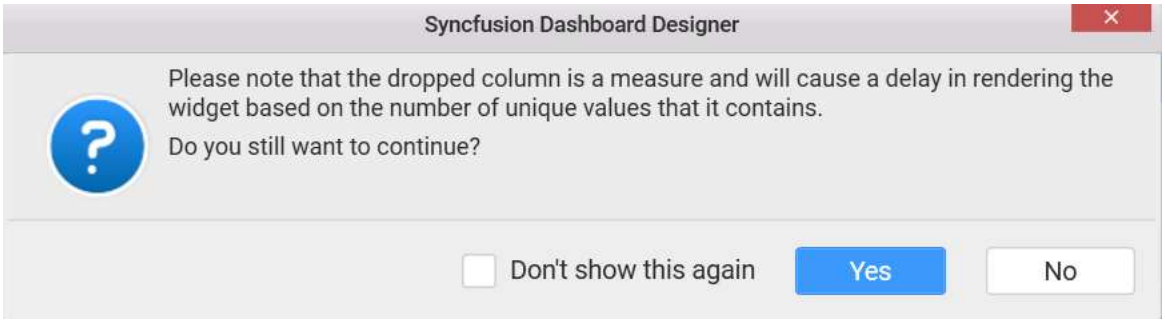


If you click No the new Dimension value will replace old value.

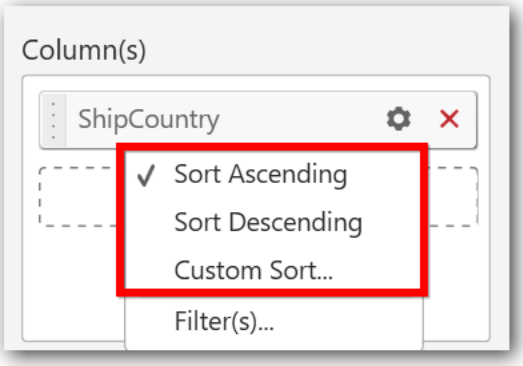


You can also add Measures and Expression columns into Column(s) field.



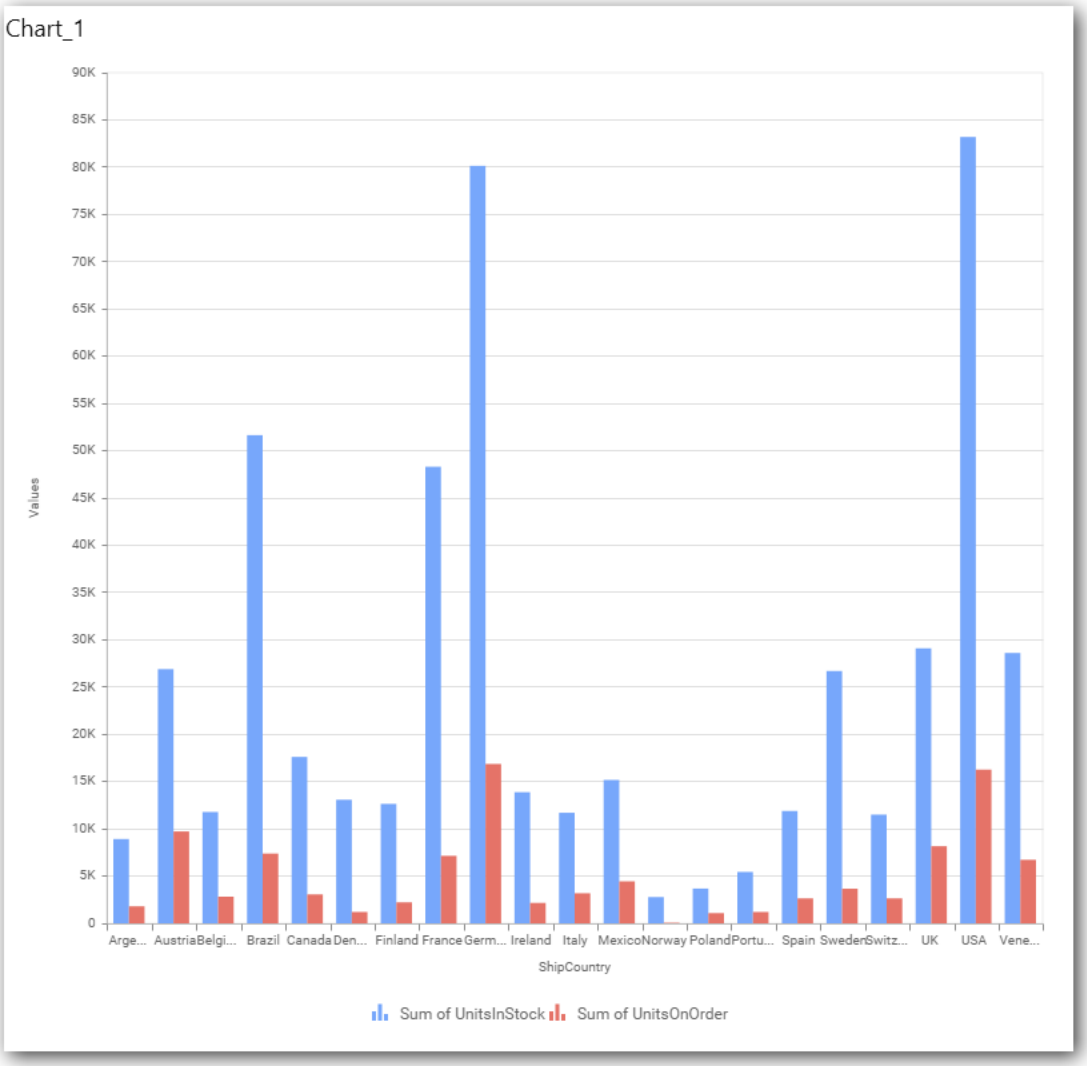


You have options to change the settings.

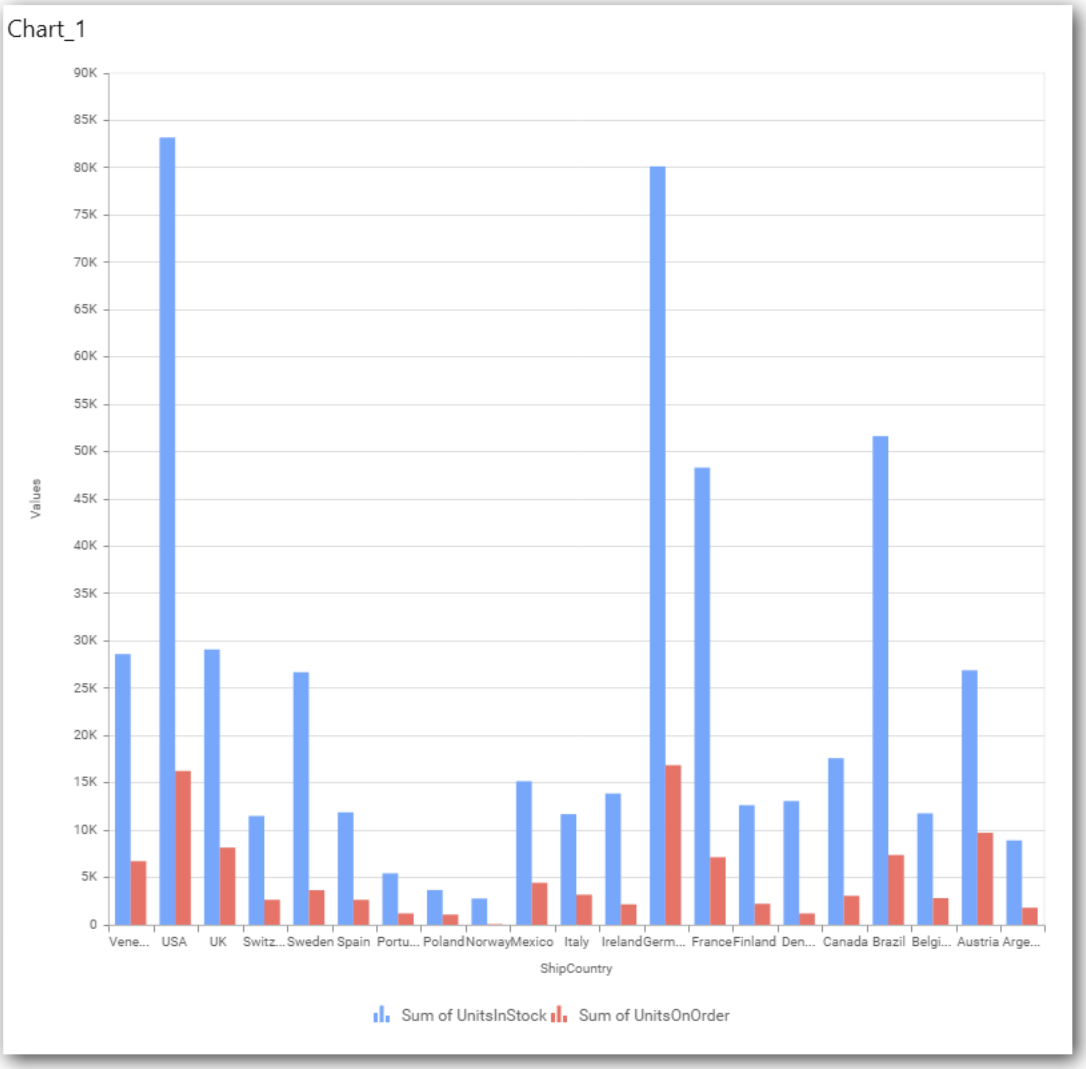


You can sort the chart either in Ascending or Descending series.

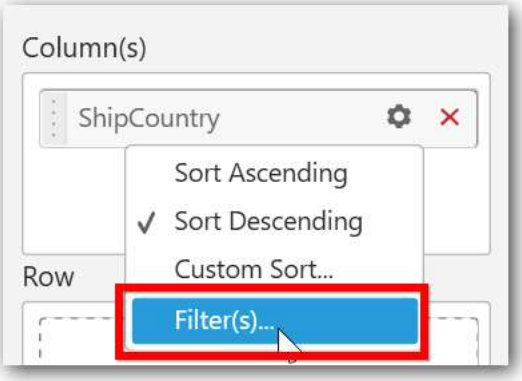
Ascending Order:

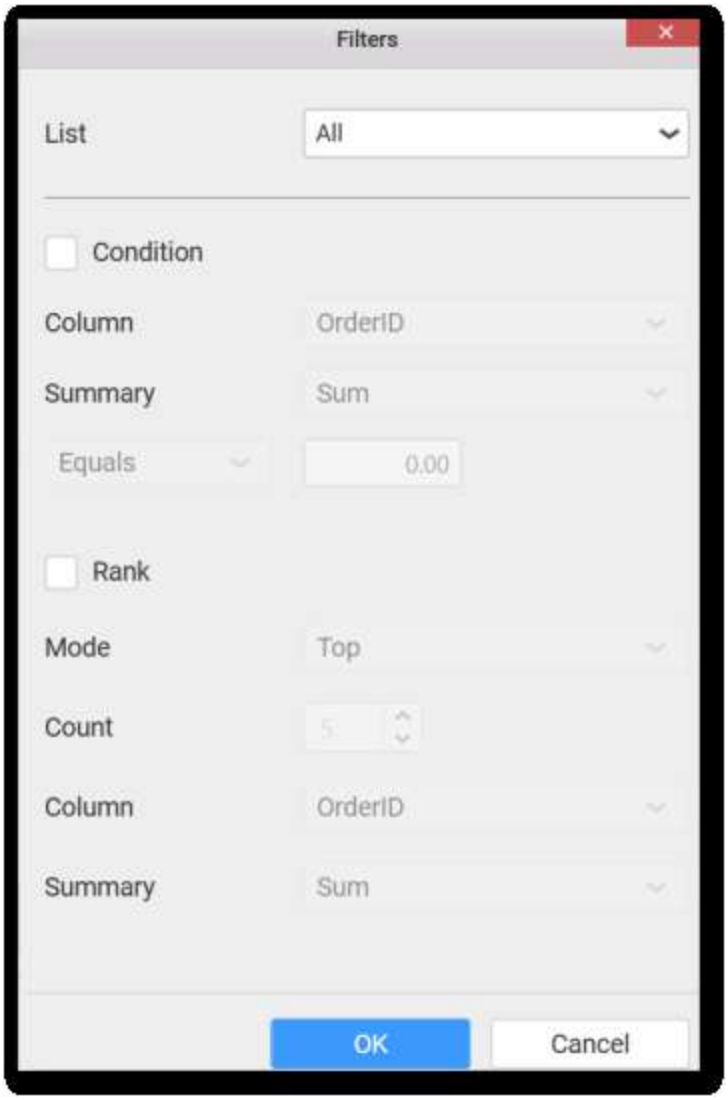


Descending order:

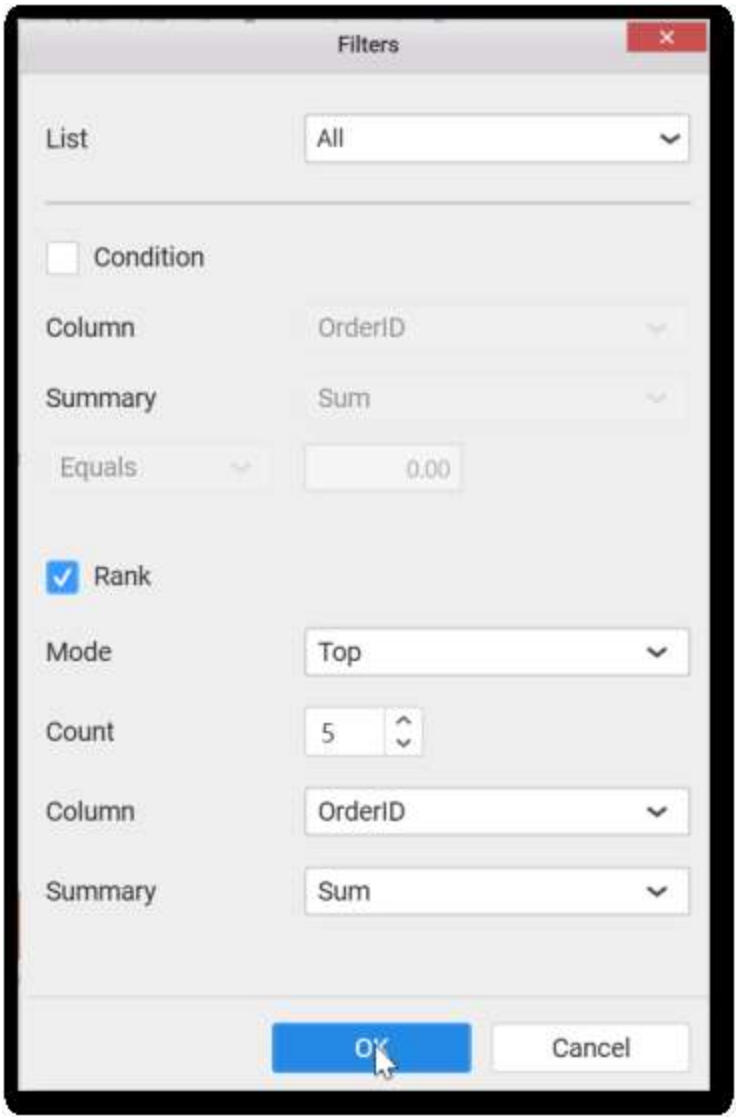


You can apply a filter.

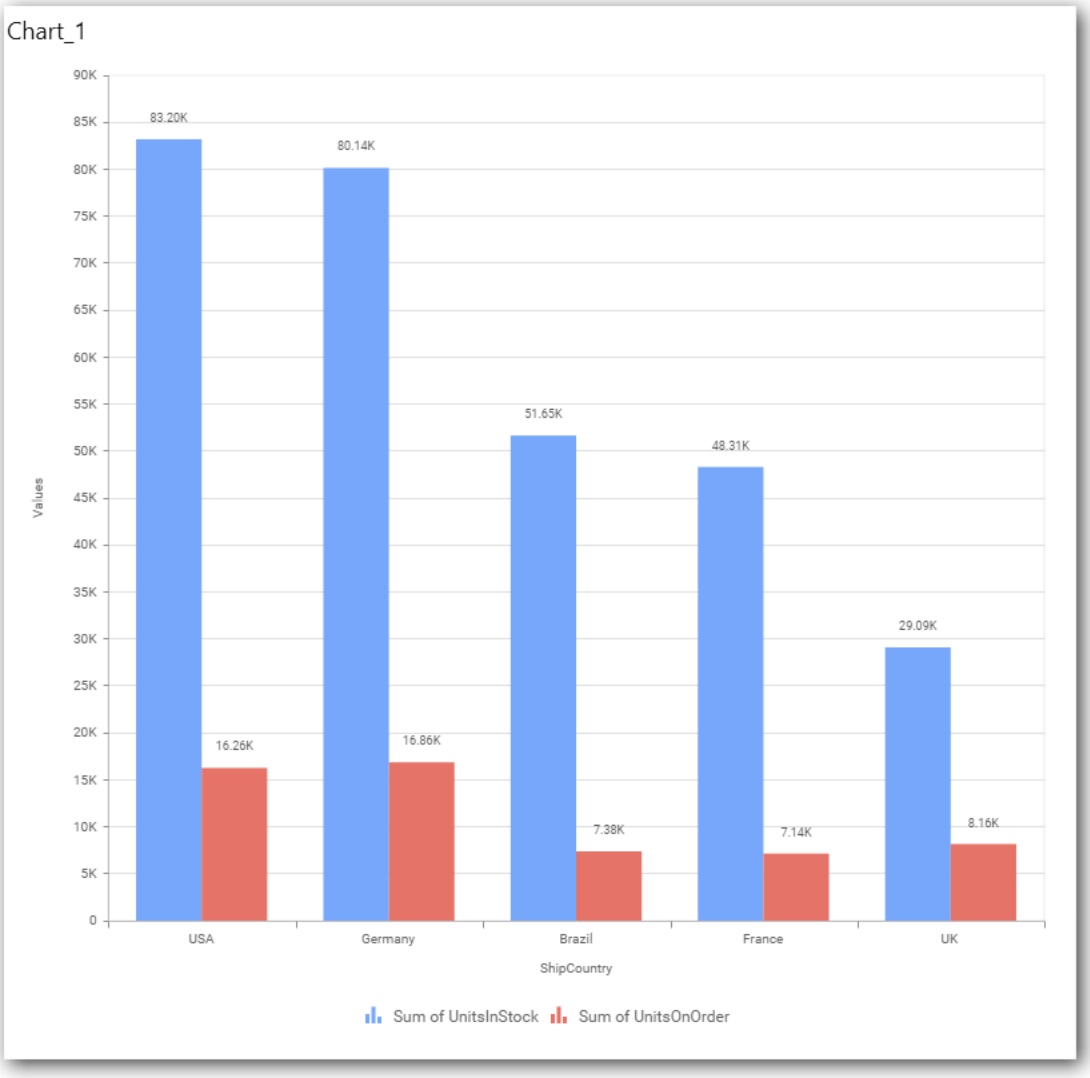




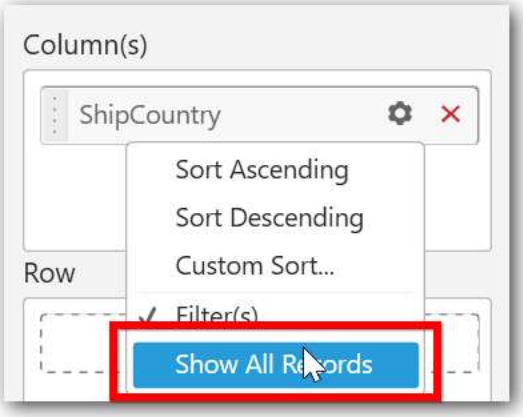
Select the Conditions and Rank you need.



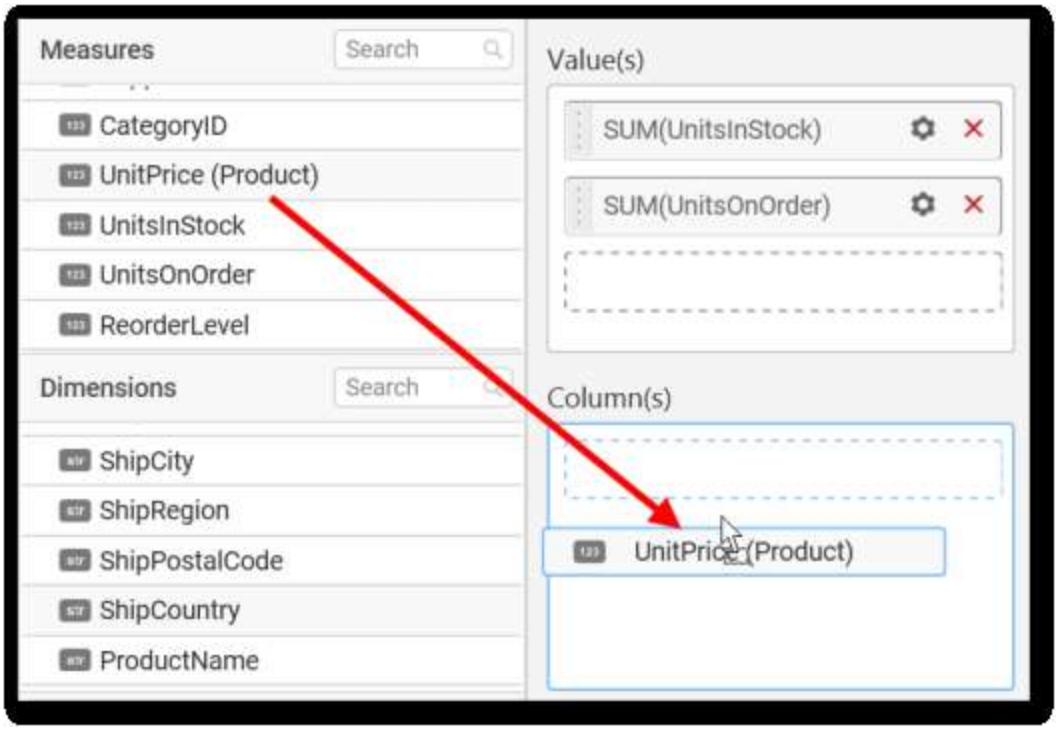
Now the chart will be rendered like this

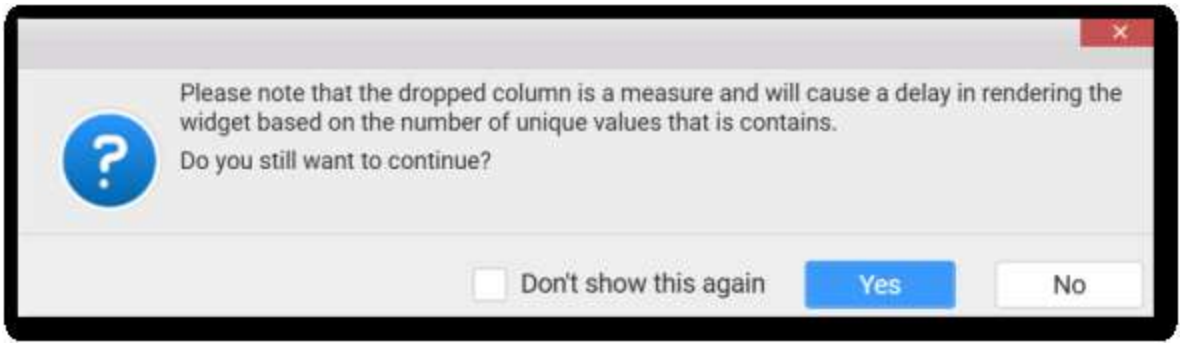


To show all records again click on Show All Records.



On adding Measures into Column(s) will show the following alert

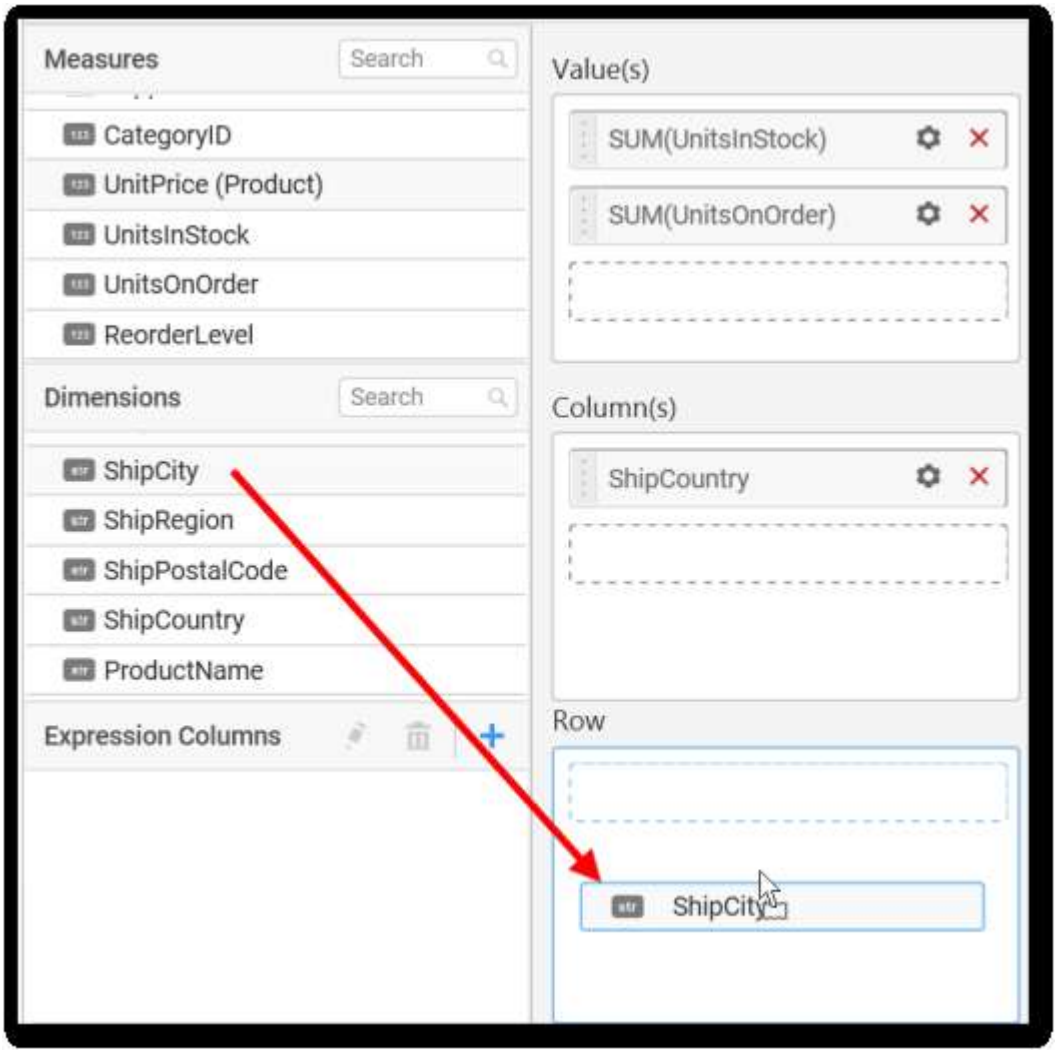




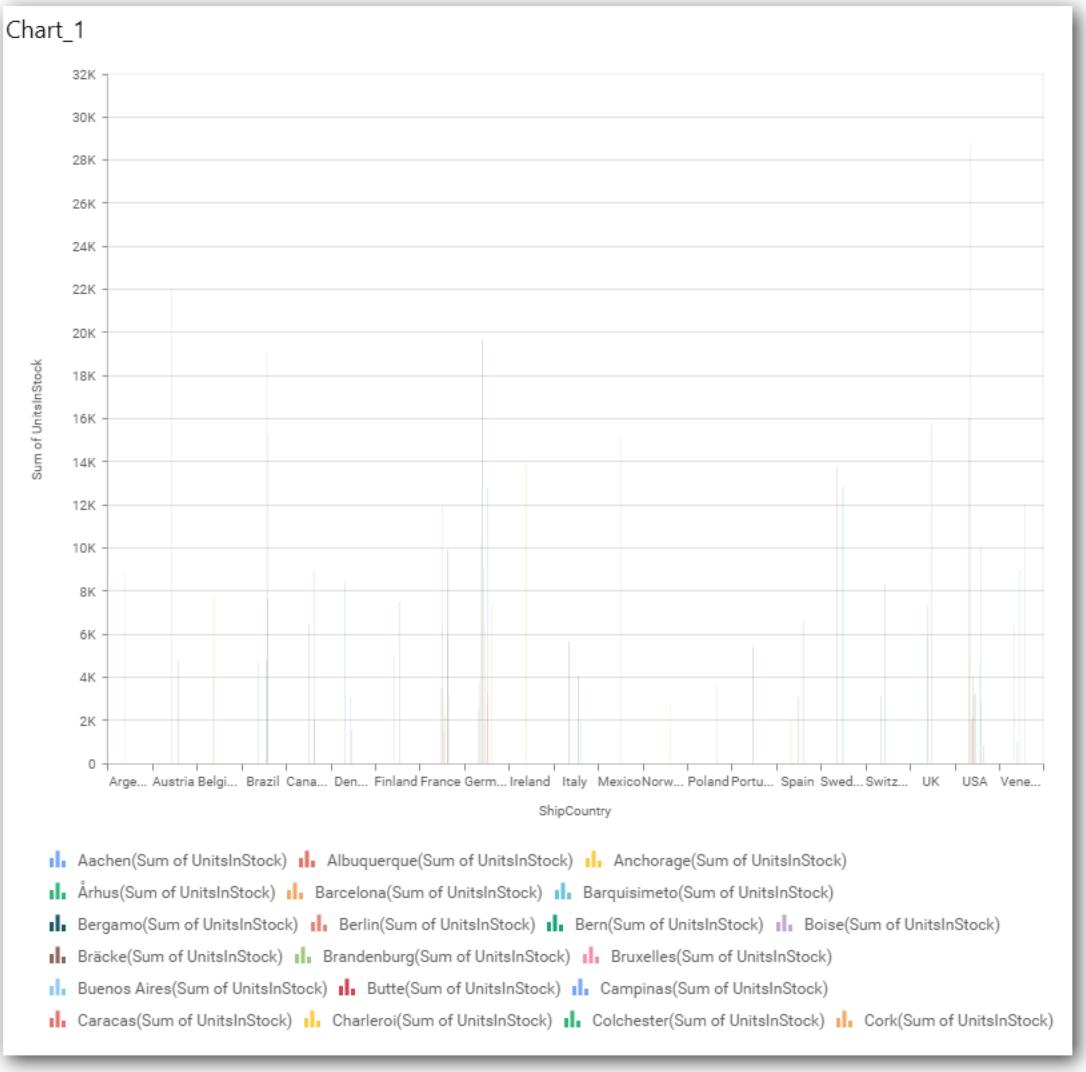
To continue select Yes, otherwise select No.

Assigning Row

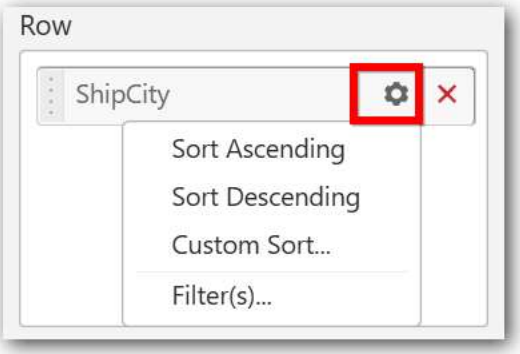
You can add Dimension into the row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s).



How to configure the SSAS data to Column Chart?

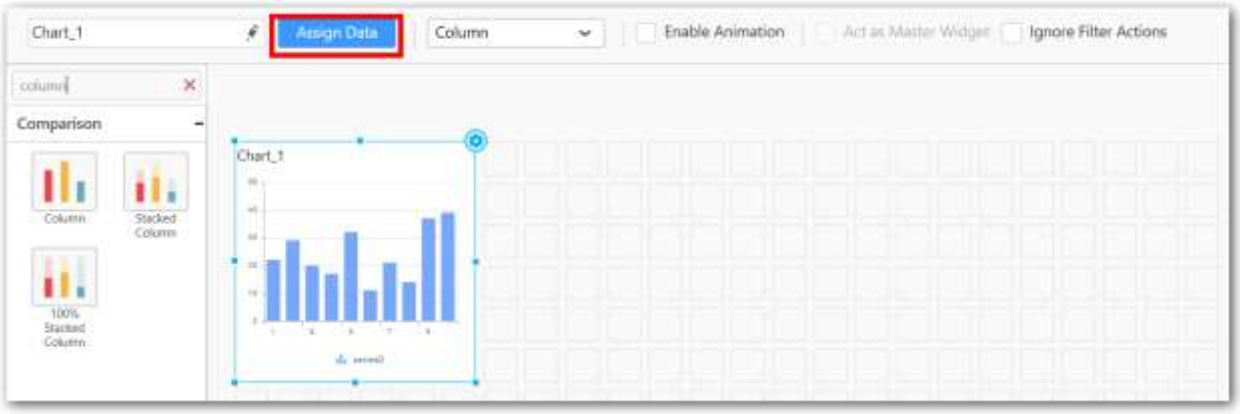
Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that

you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuration of SSAS data to Column chart

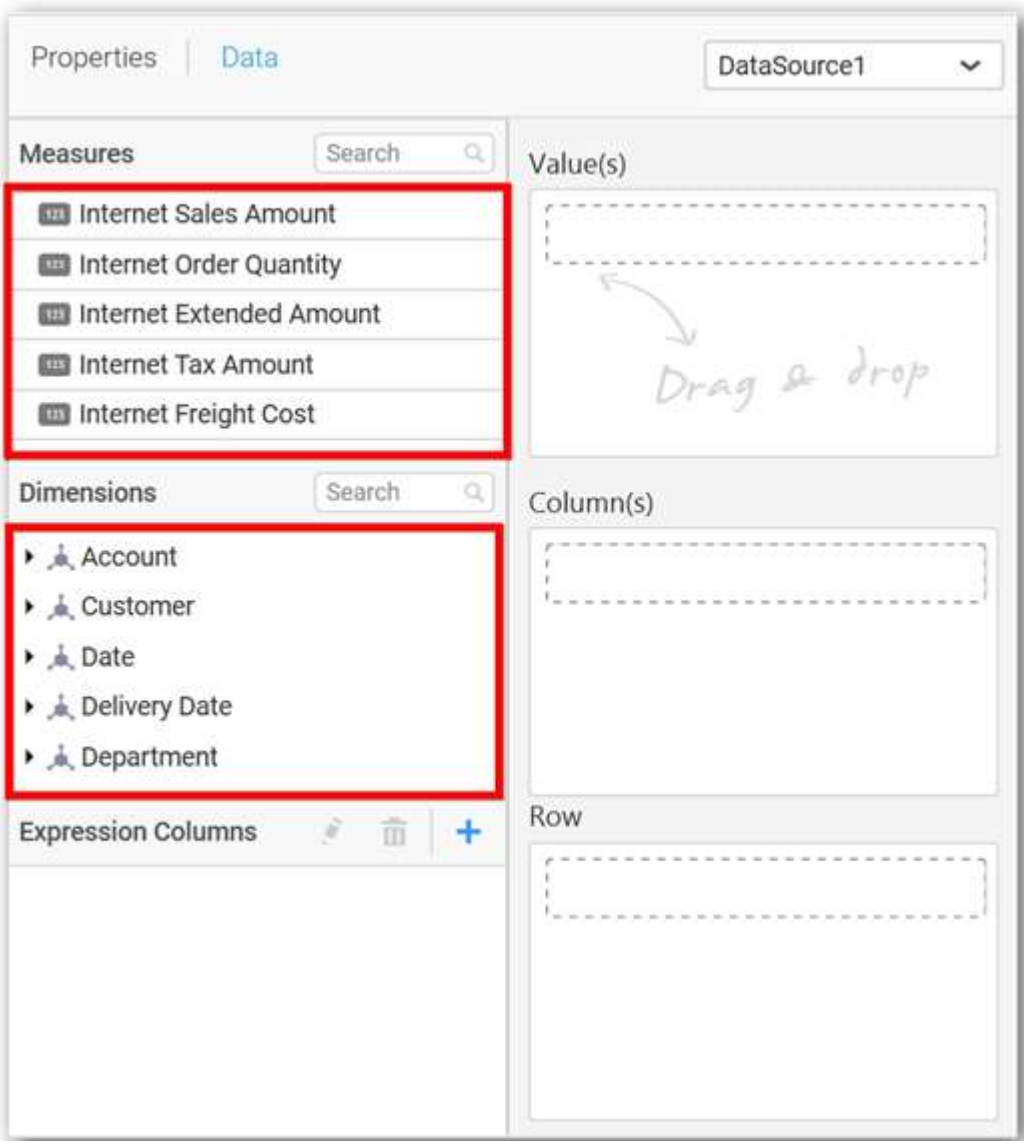
Drag and drop the Column chart widget into canvas and resize it to your required size.

Select the dropped widget using mouse.



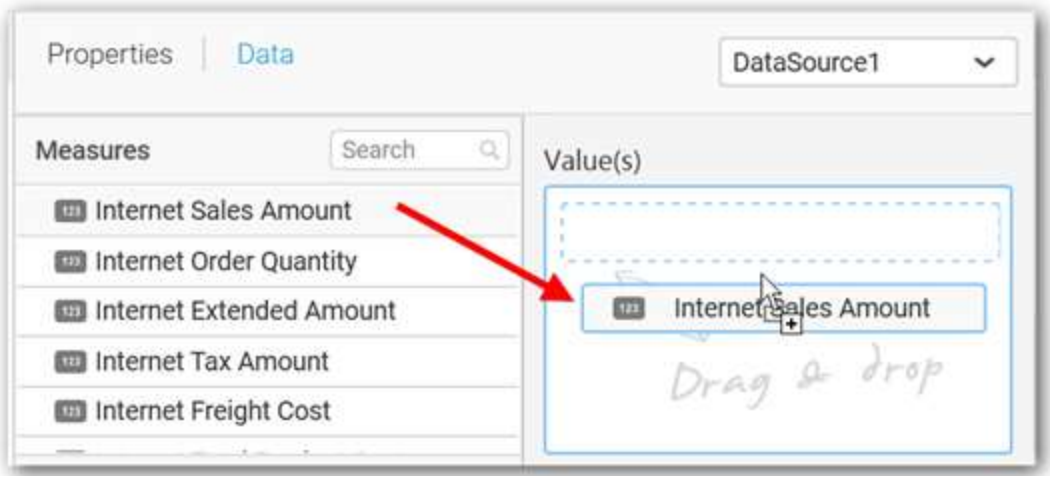
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions

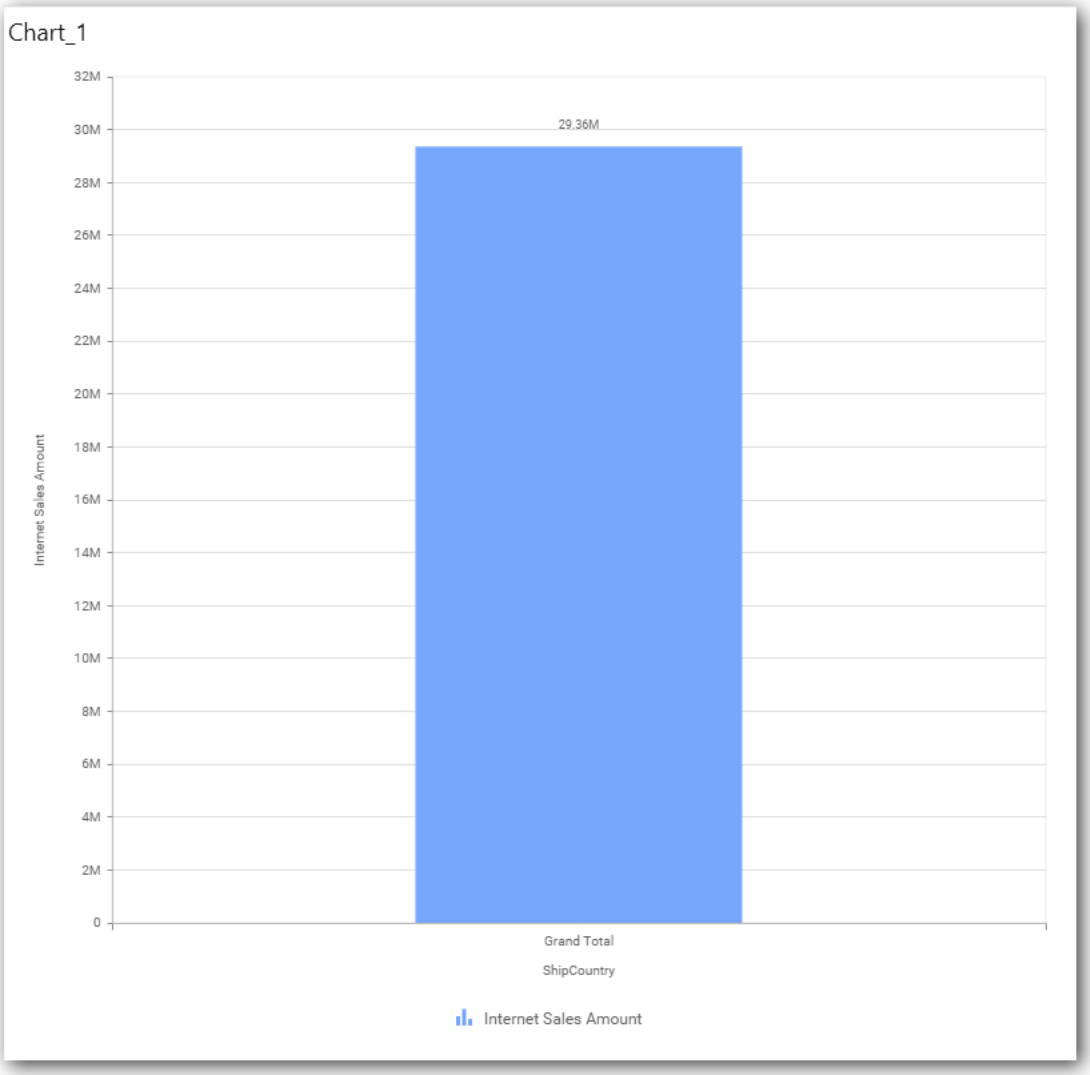


Assigning Value(s)

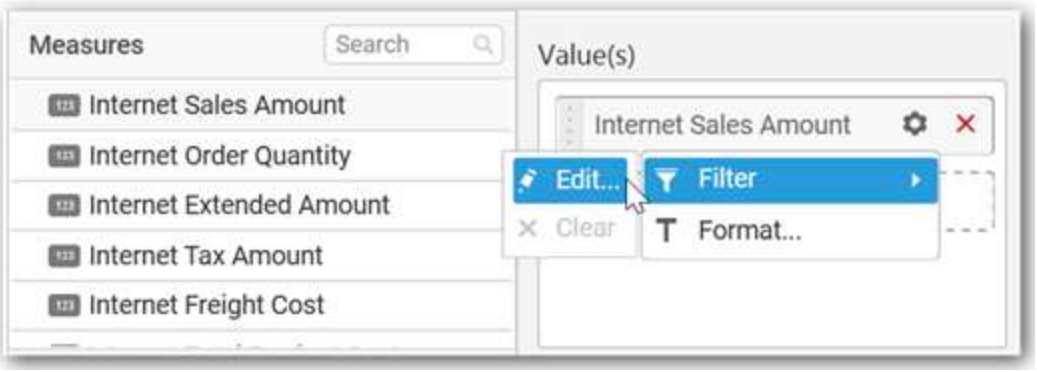
Drag and drop a column under Measures category into Value(s).



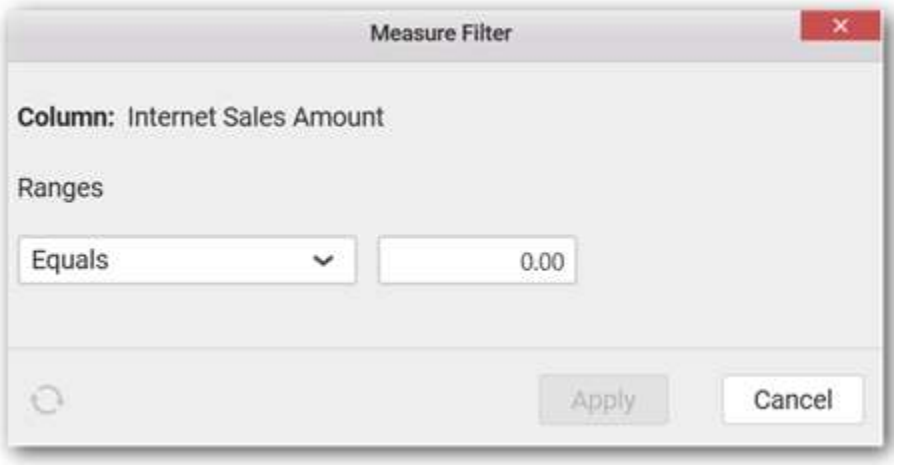
Now the chart will be rendered like this.



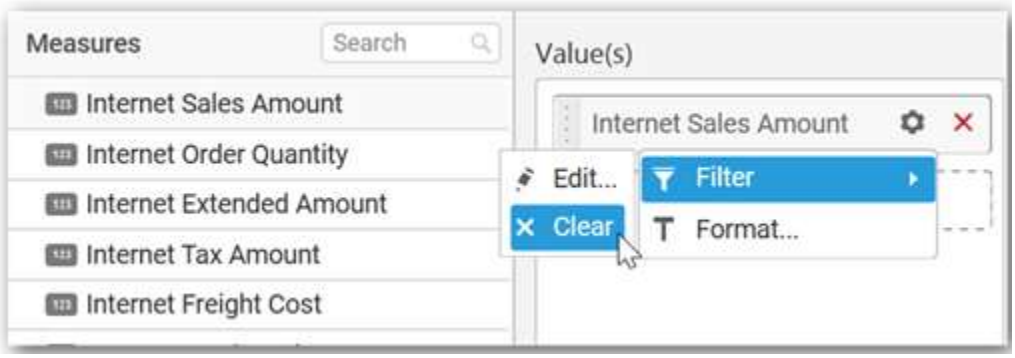
Define the filter criteria to match through choosing Edit option in Filter menu item.



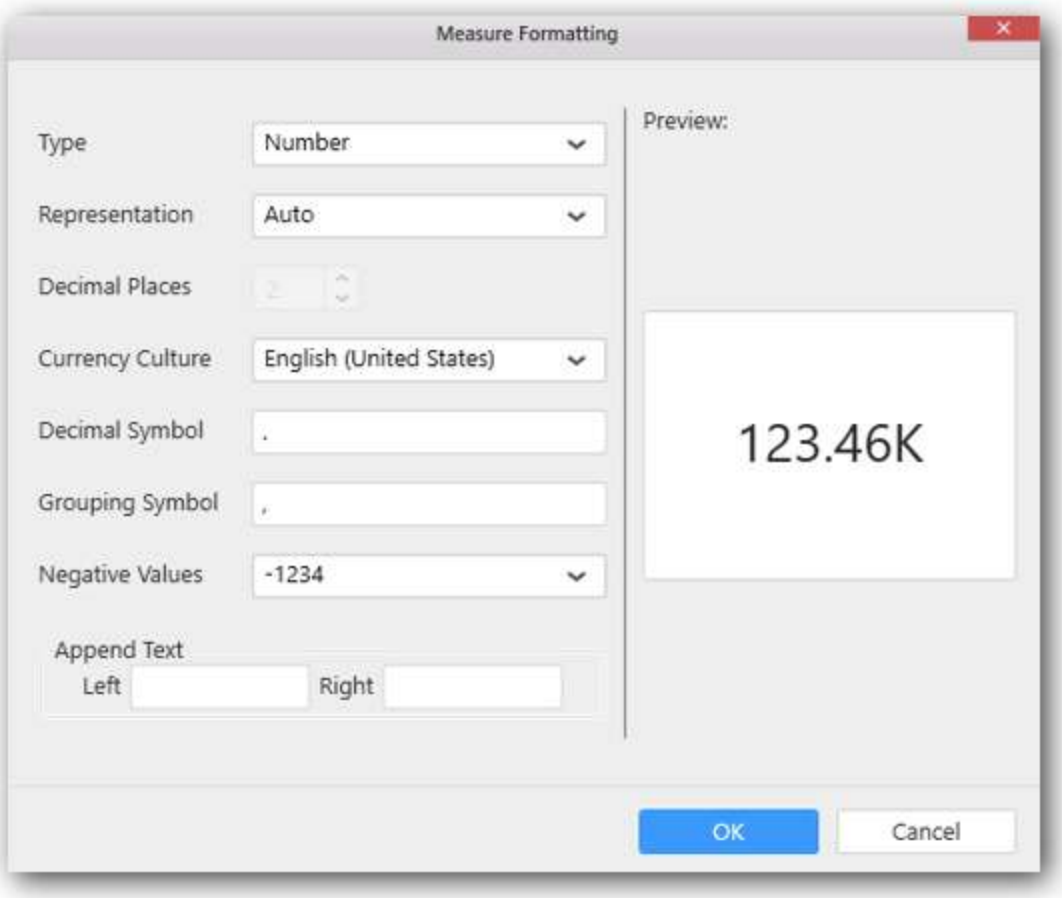
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



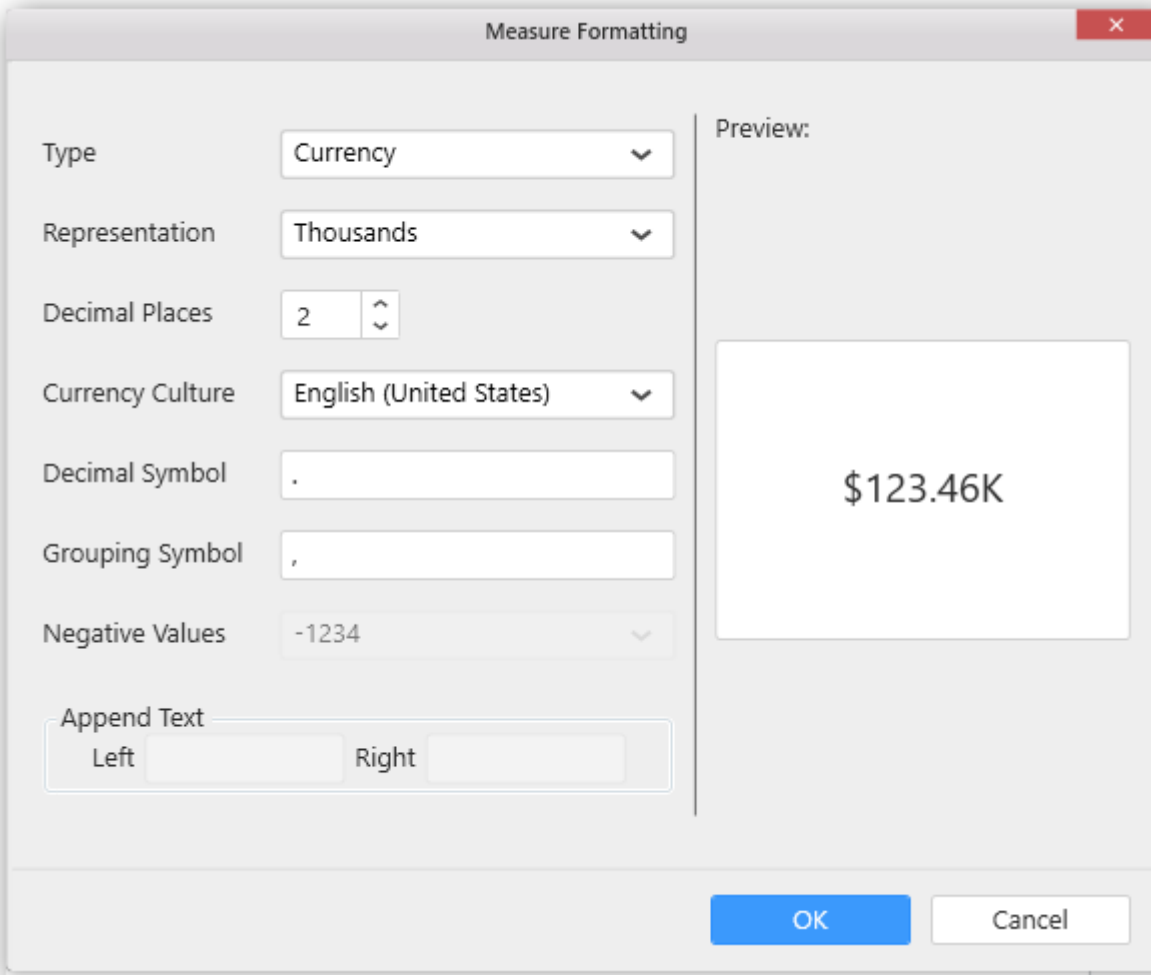
Select Clear option to clear the defined filter.



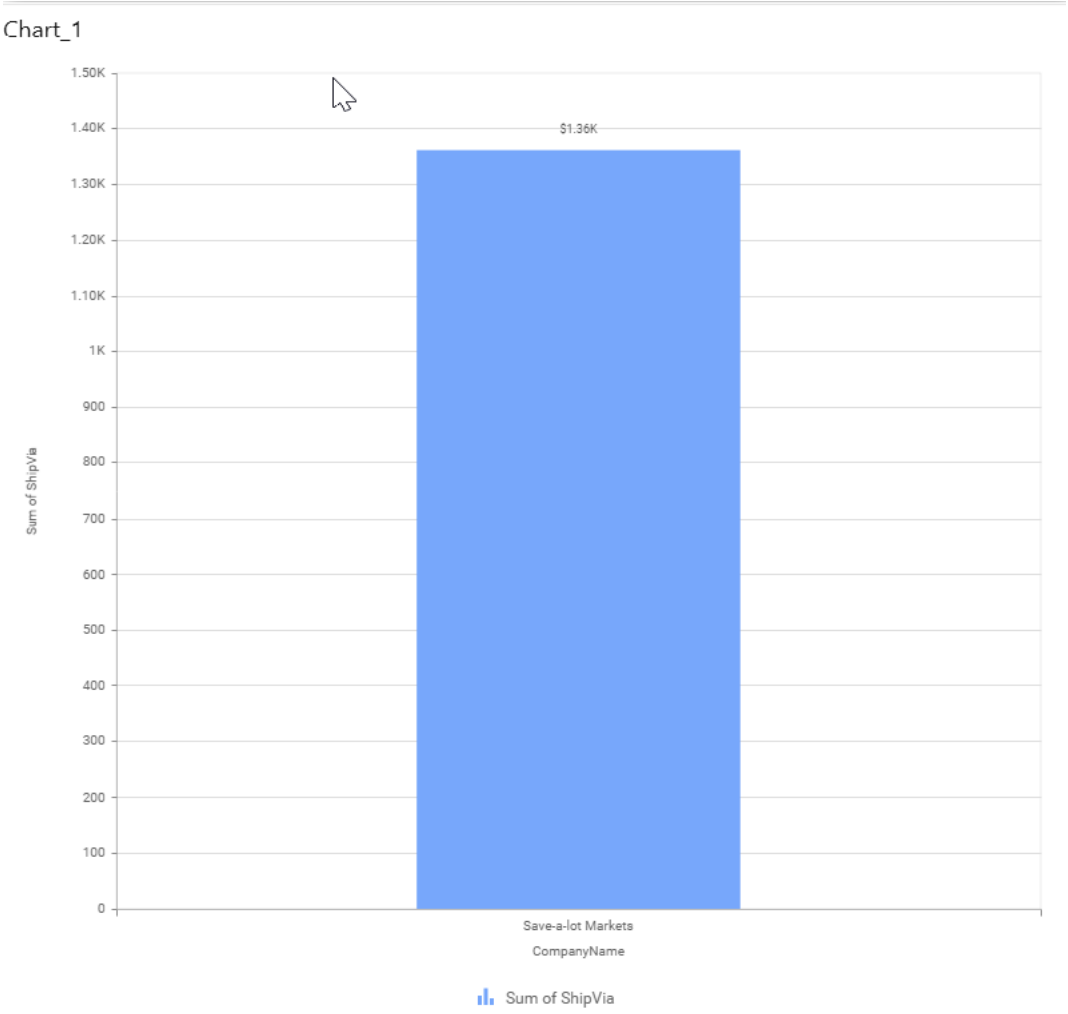
Select Format option to define the display format to the values in the column through Measure Formatting window.



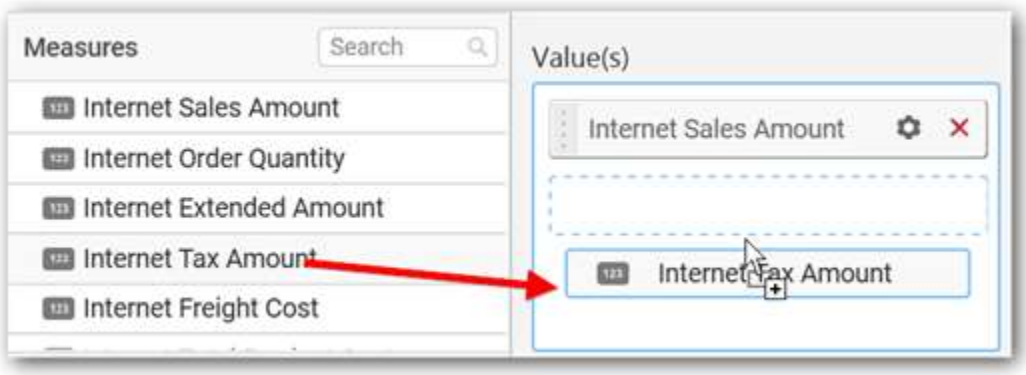
Choose the options you need and click OK.

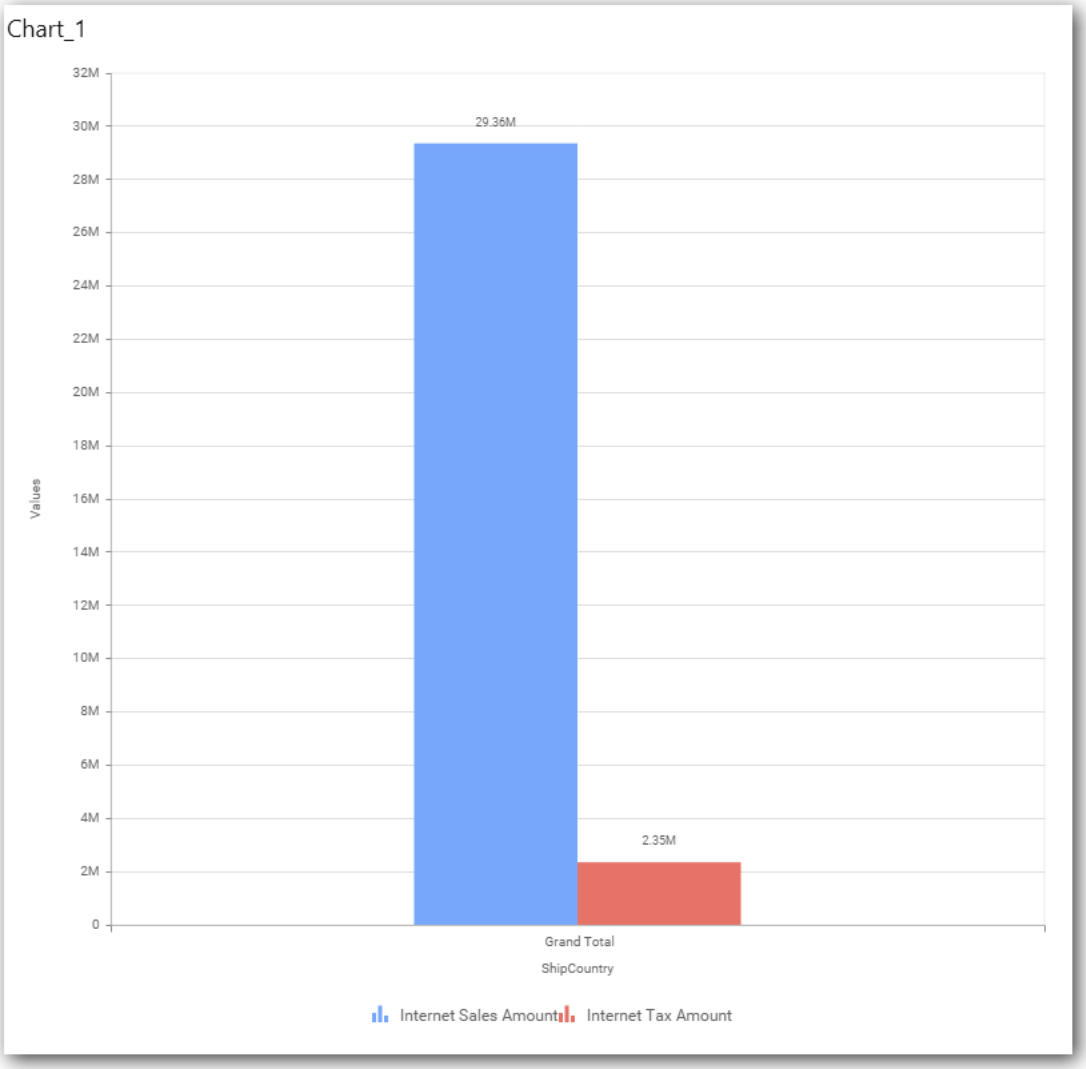


Now the Chart will be rendered like this.



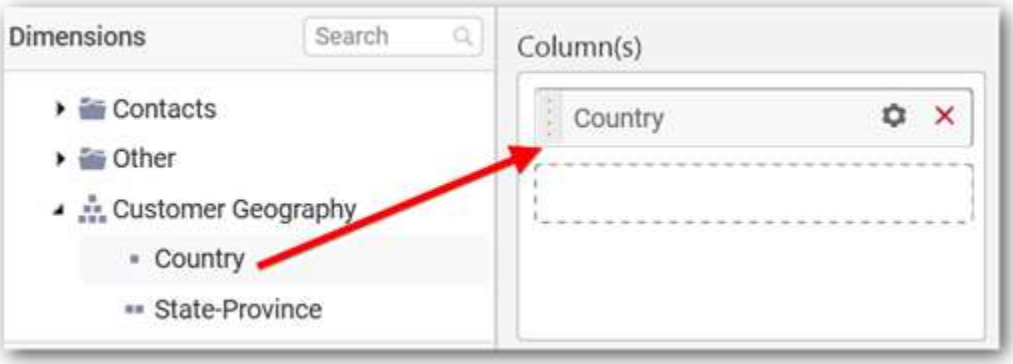
You can also add more than one column to the Value(s) section.

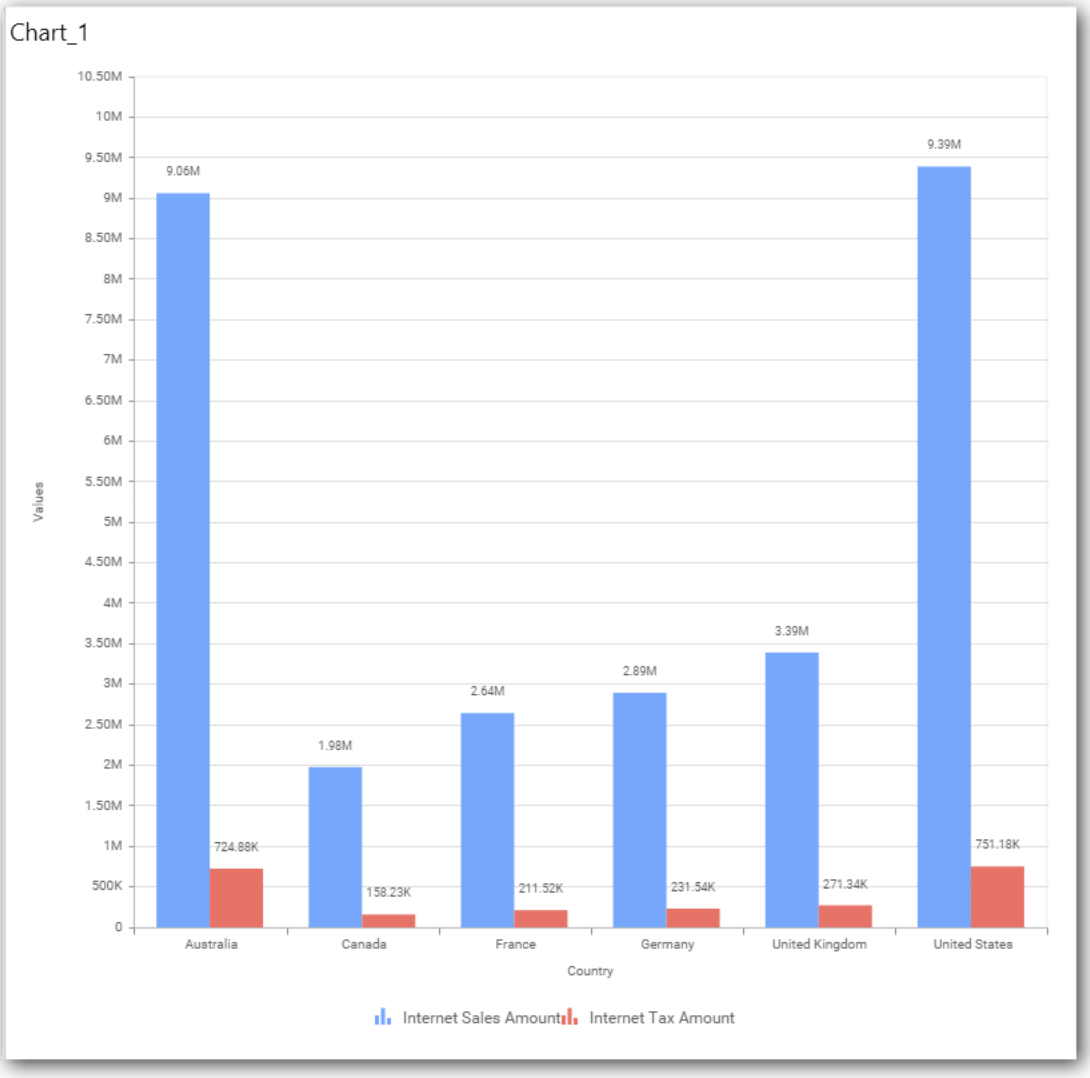




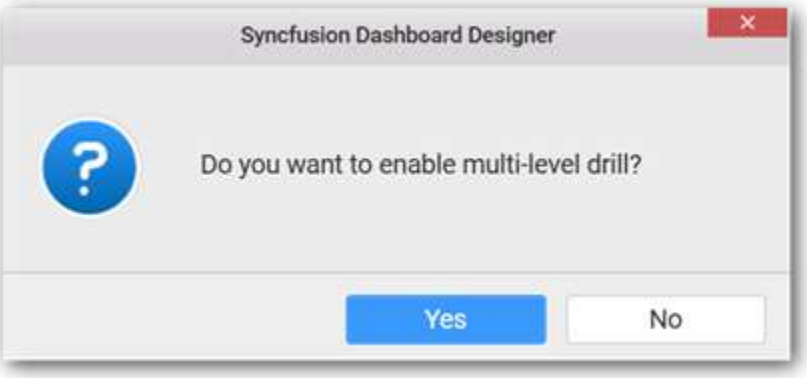
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.



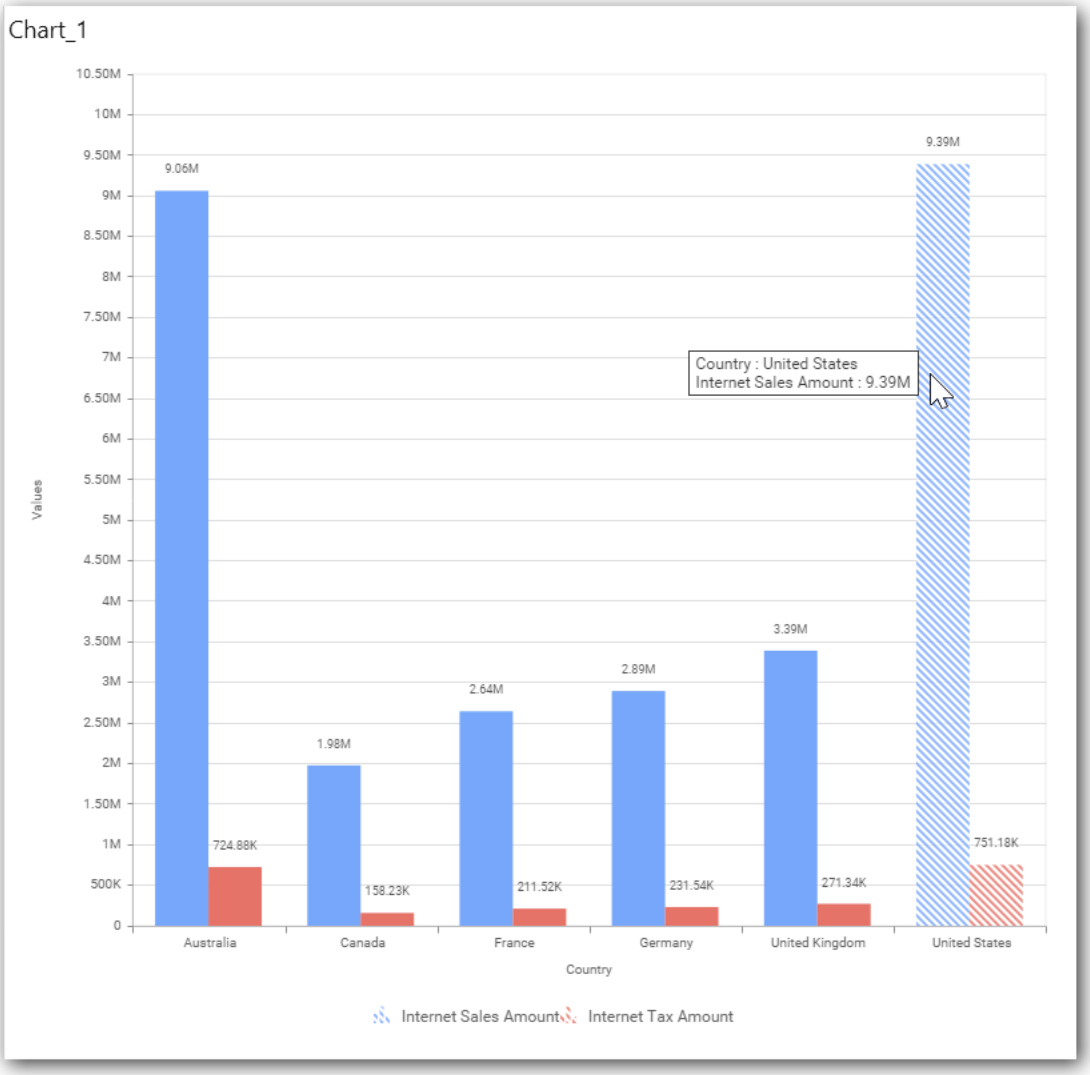


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

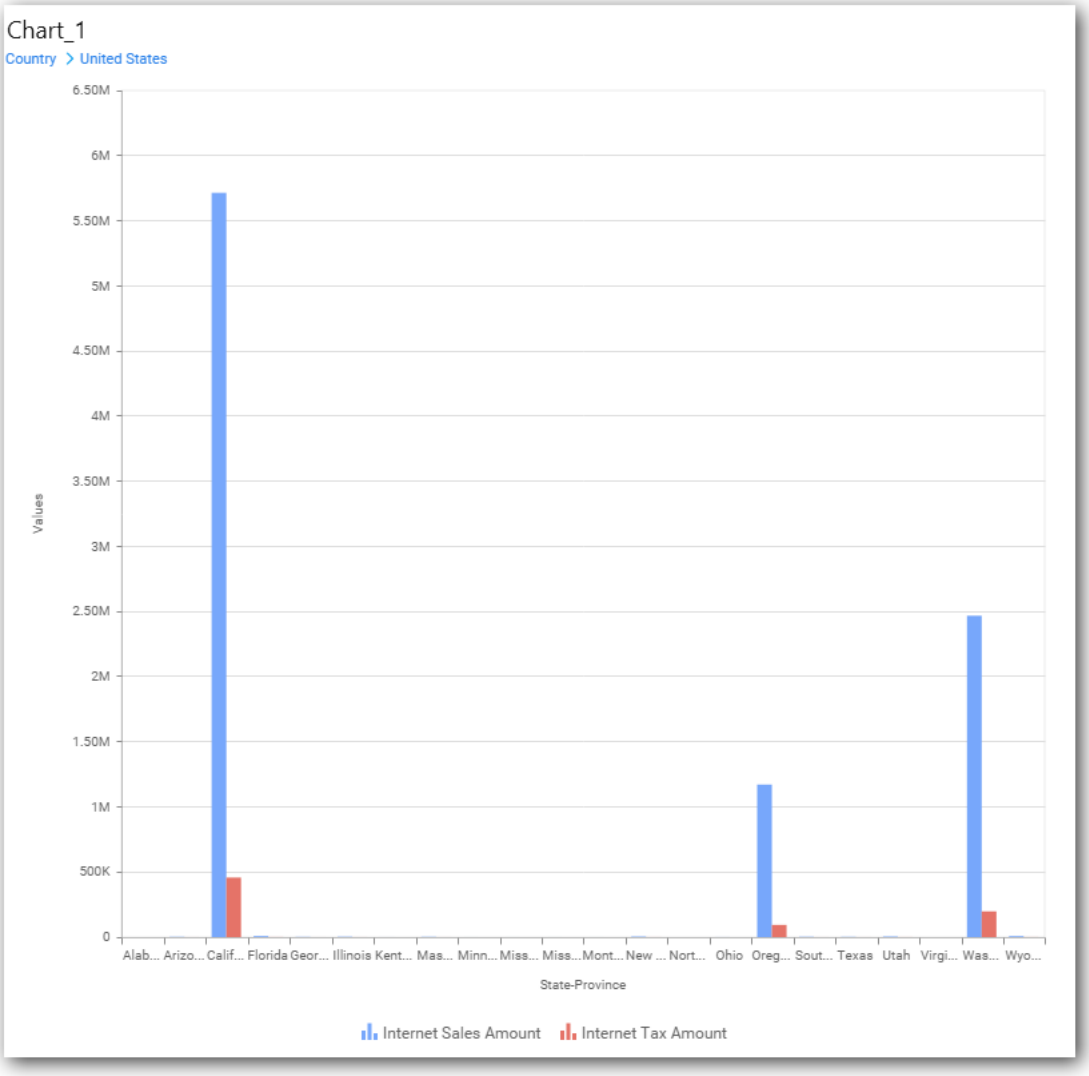


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

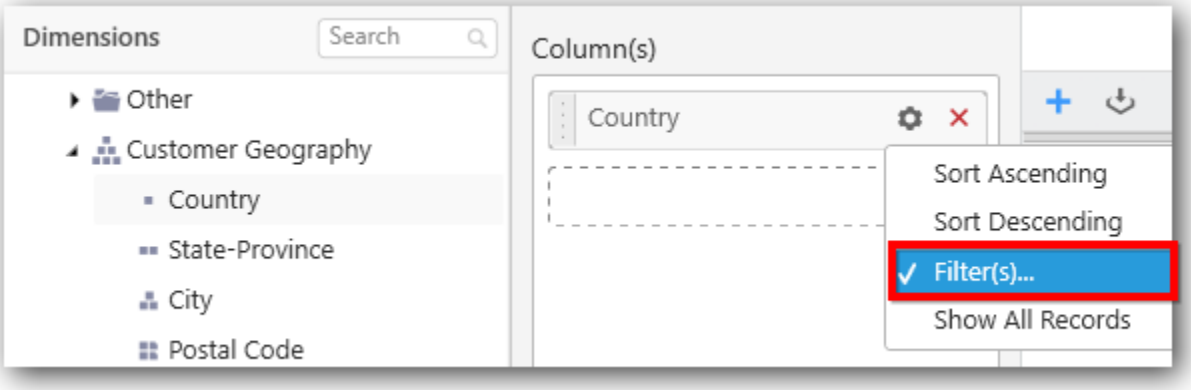
Click the respective data value marker in chart to drill into its inner level.



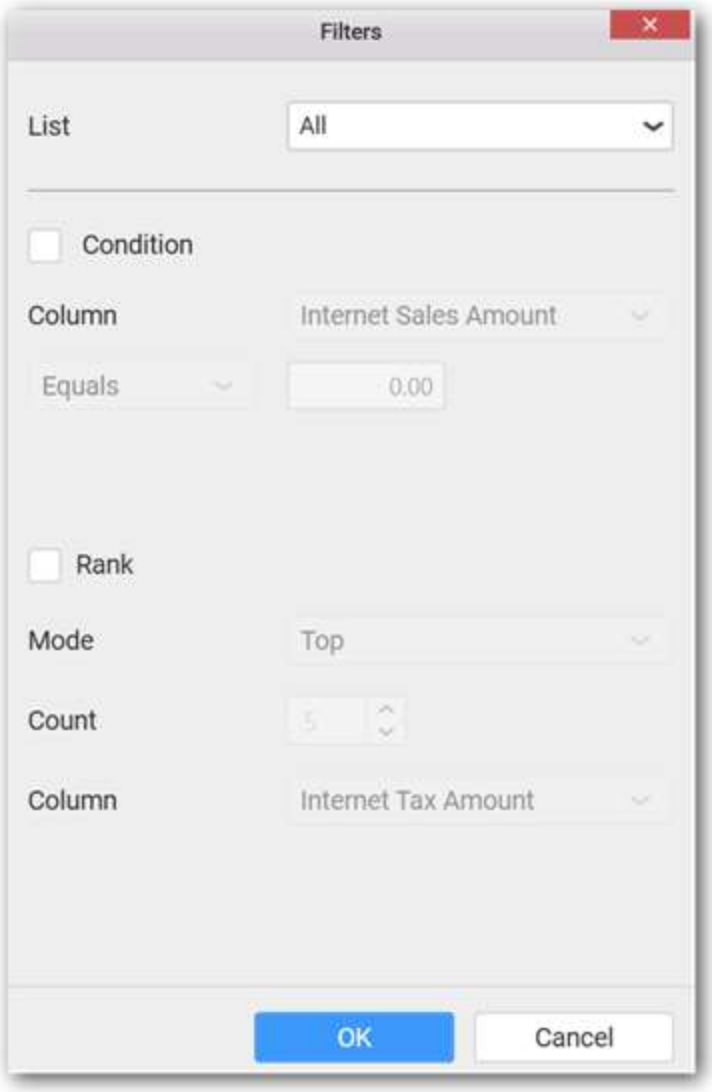
The drilled view of the chart is follows.



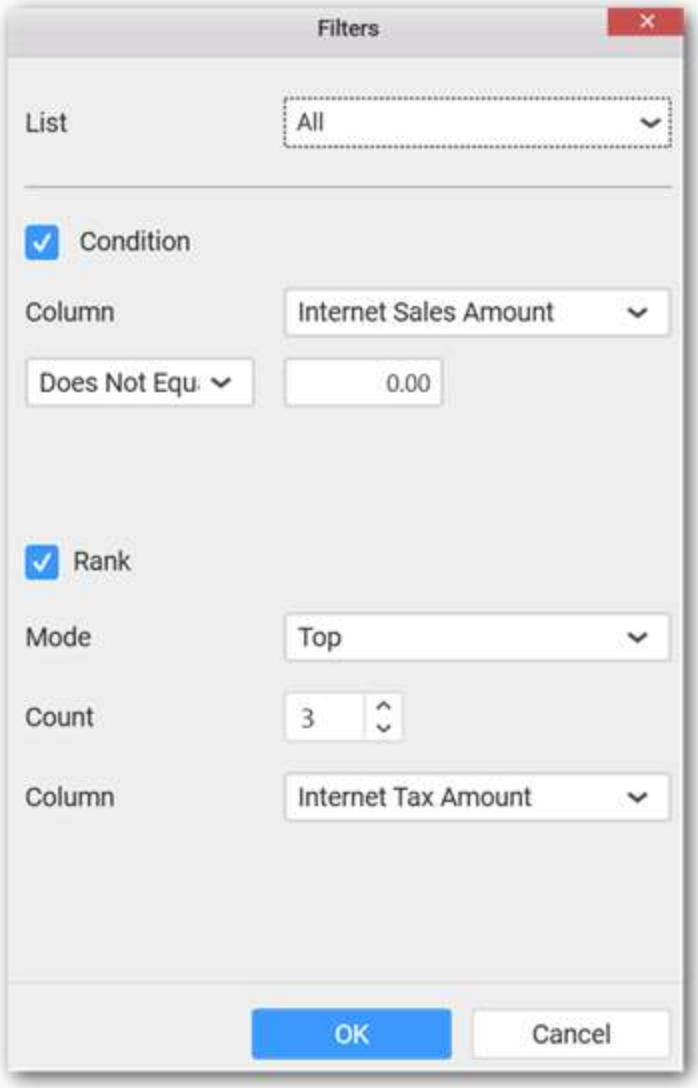
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



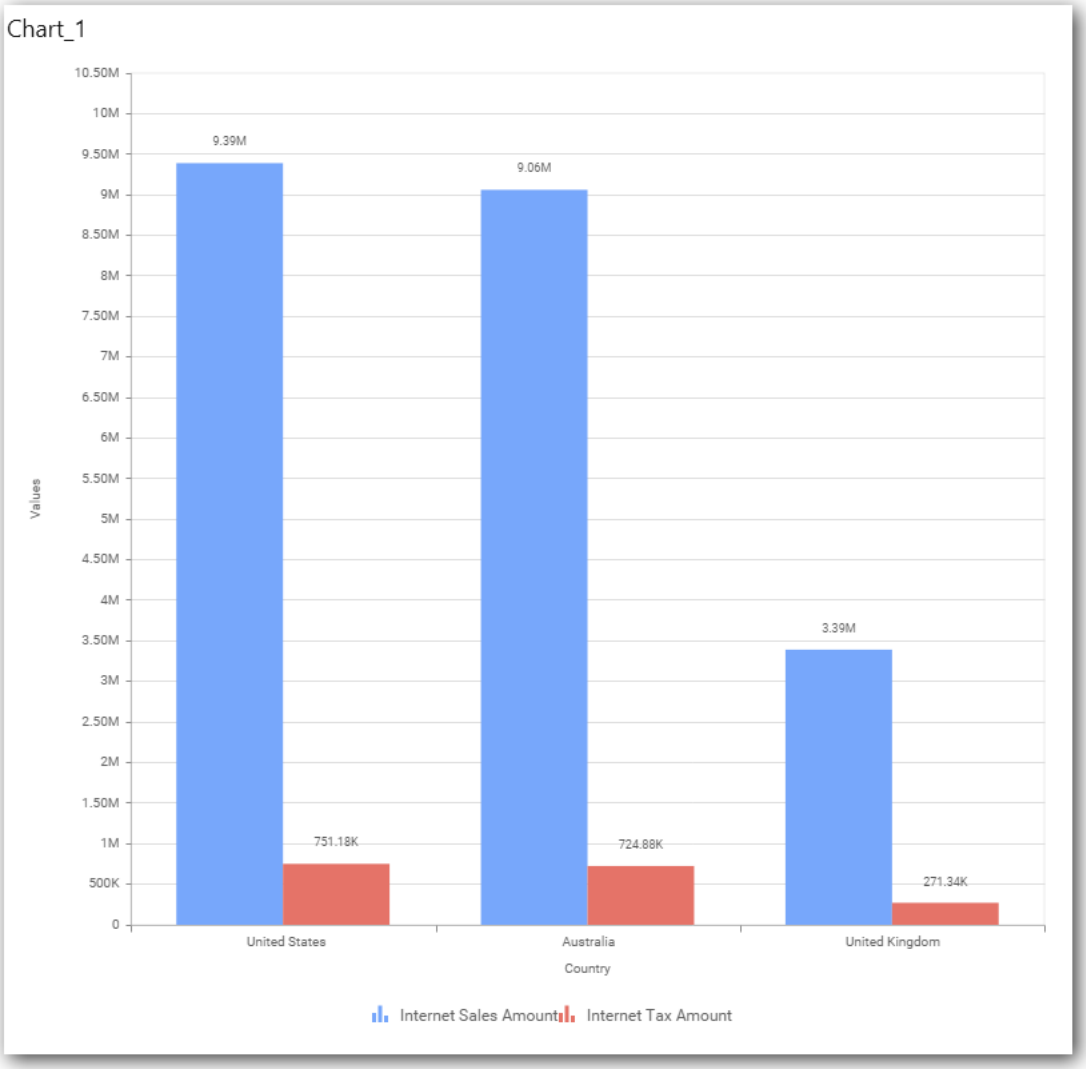
Select Filter(s)… to launch the Filters window.



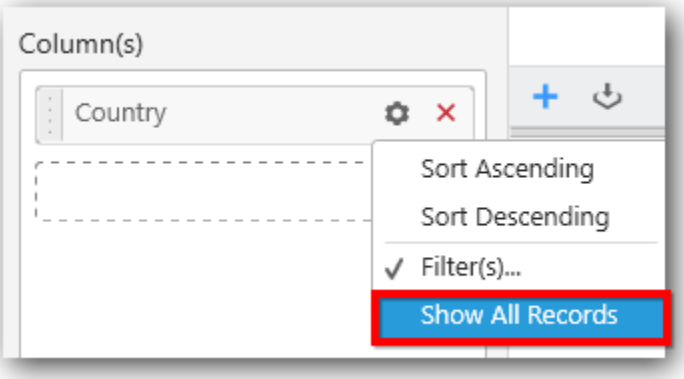
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

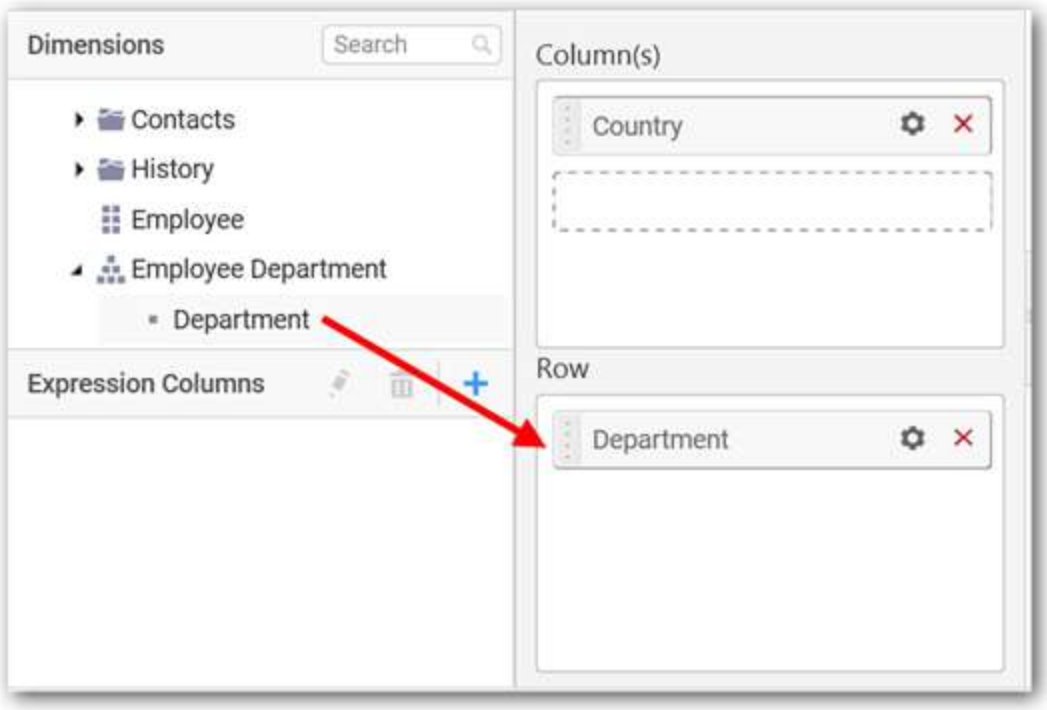


To show all records again click on Show All Records.

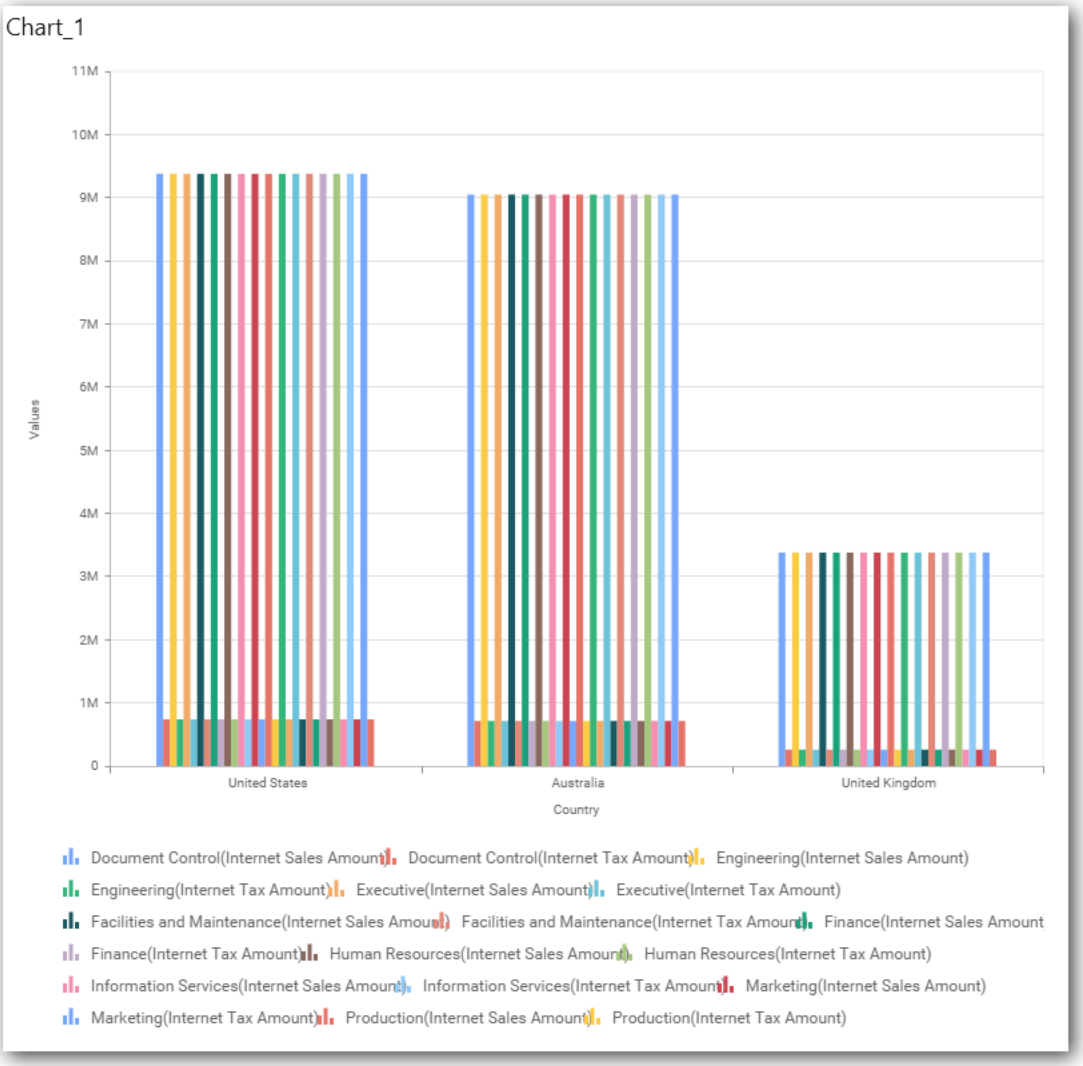


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.

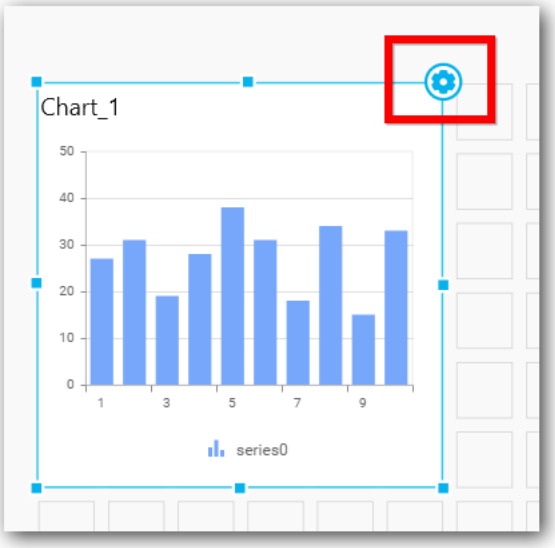


How to format Column Chart?

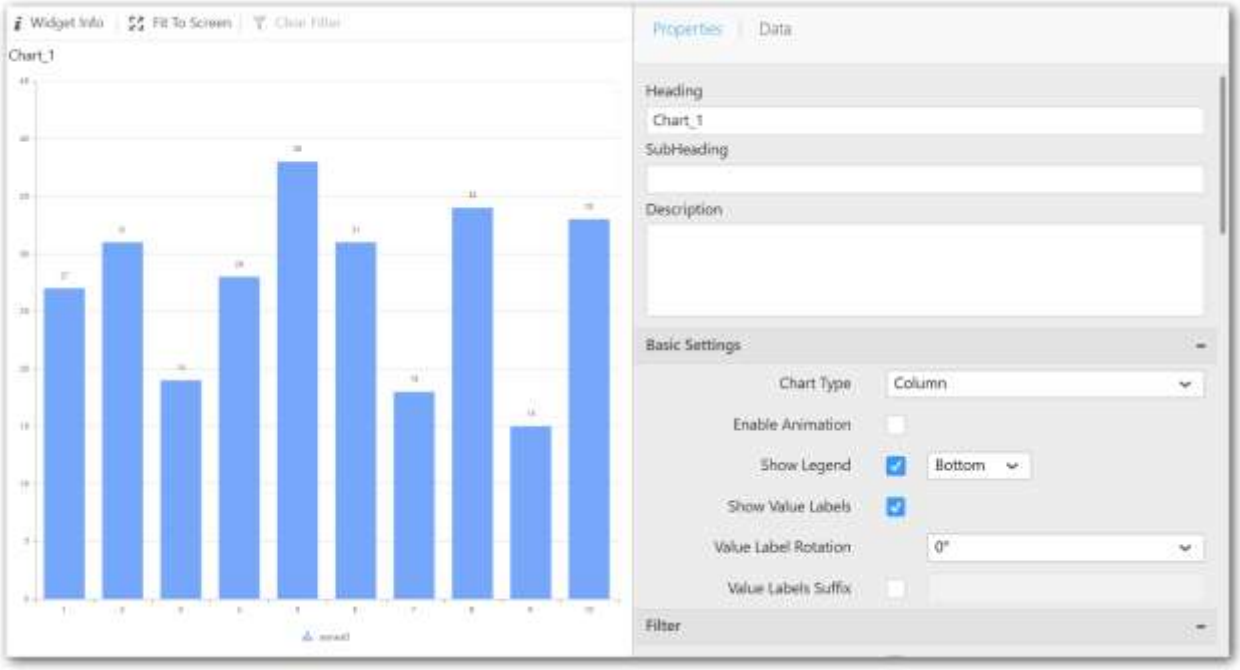
You can format the column chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into Column chart follow the steps

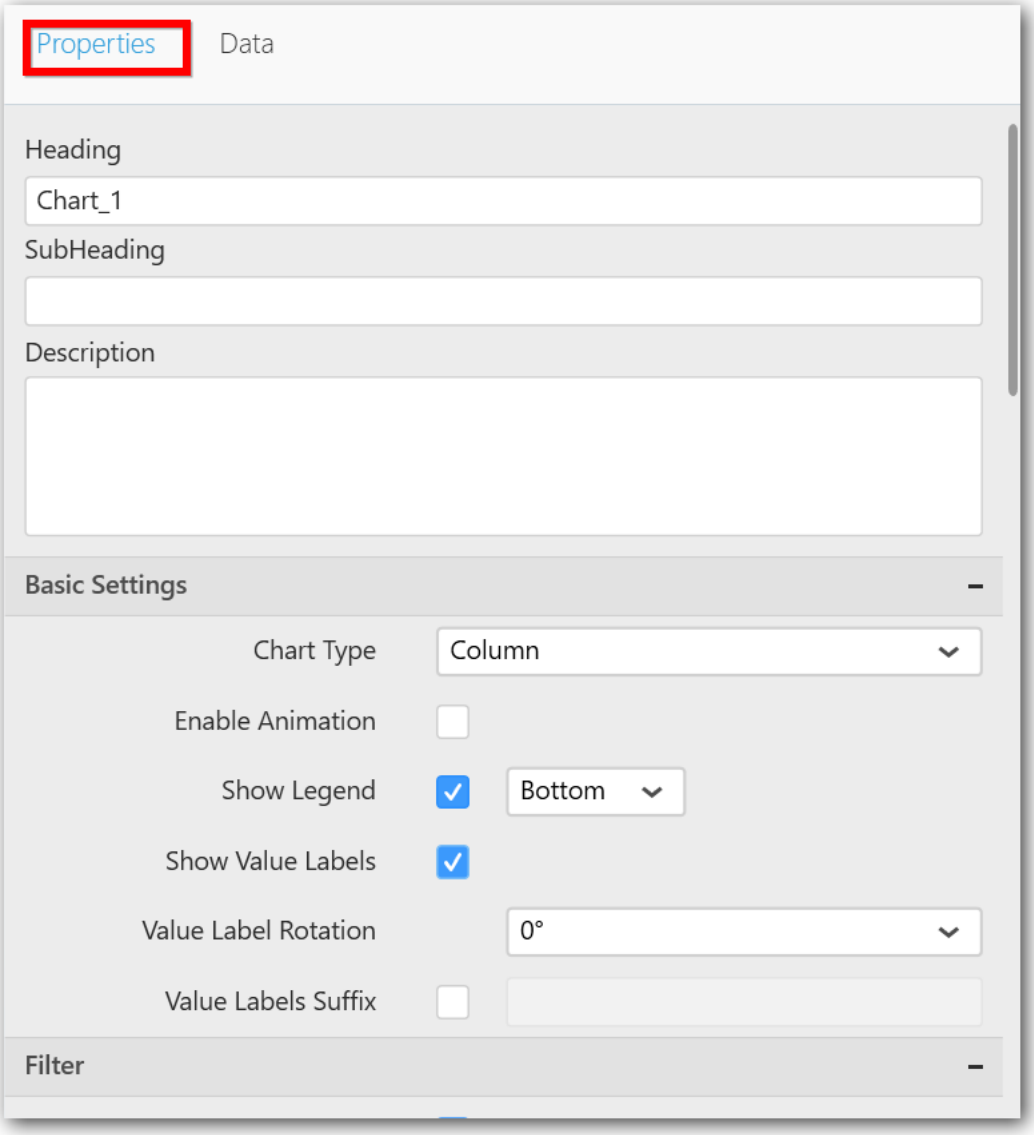
1. Drag and drop the column chart into canvas and resize it to your required size.
2. Configure the data into column chart.
3. Focus on the column chart and Click on Widget Settings.



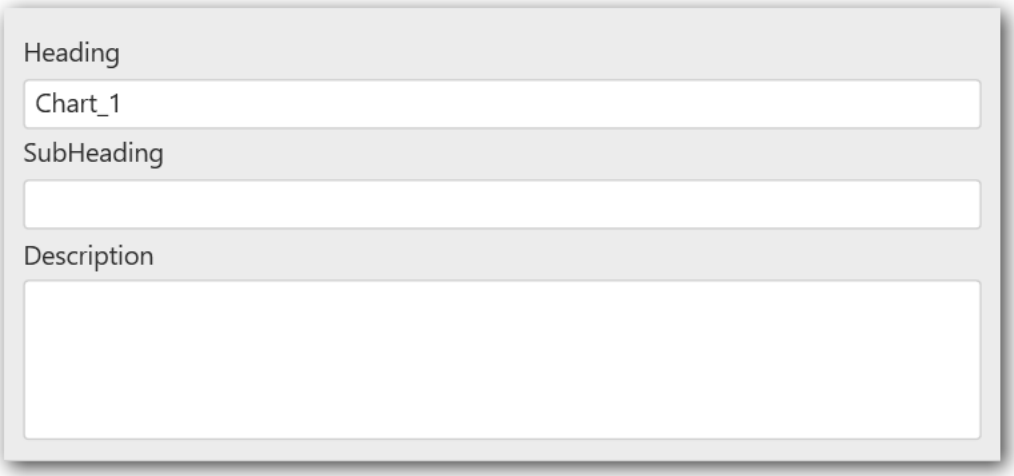
The property window will be opened.



You can see the list of properties available for the widget with default value.



General Settings



Header

This allows you to set title for this column chart widget.

Description

This allows you to set description for this column chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

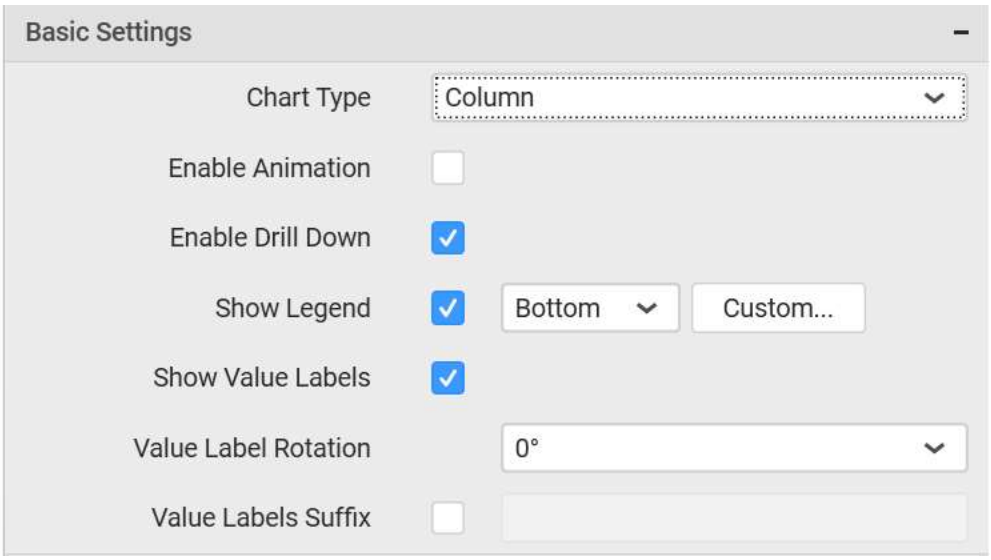


Chart Type

This allows you to switch the widget view from current chart type to another convertible chart type.

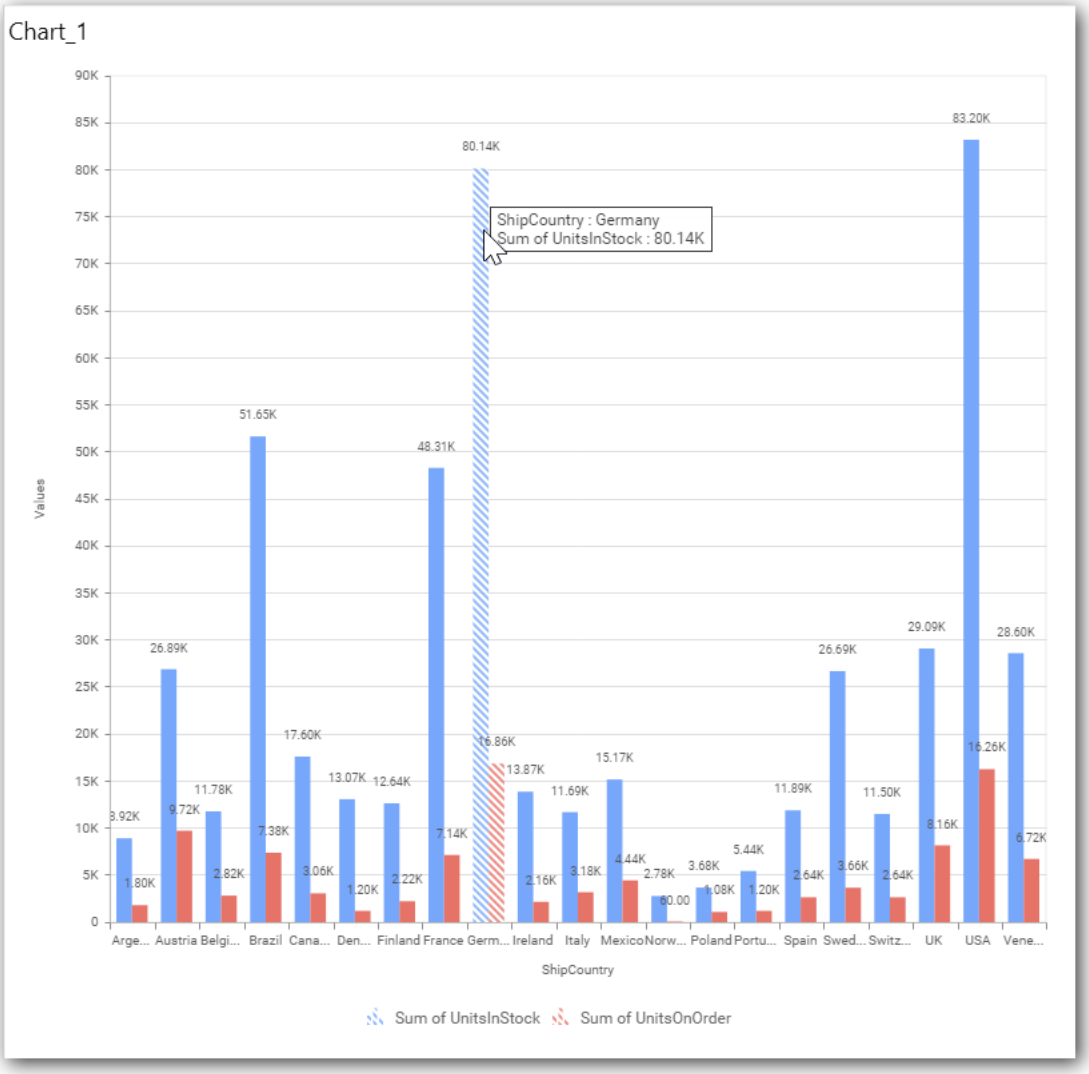
Enable Animation

This allows you to enable the series rendering in animated mode.

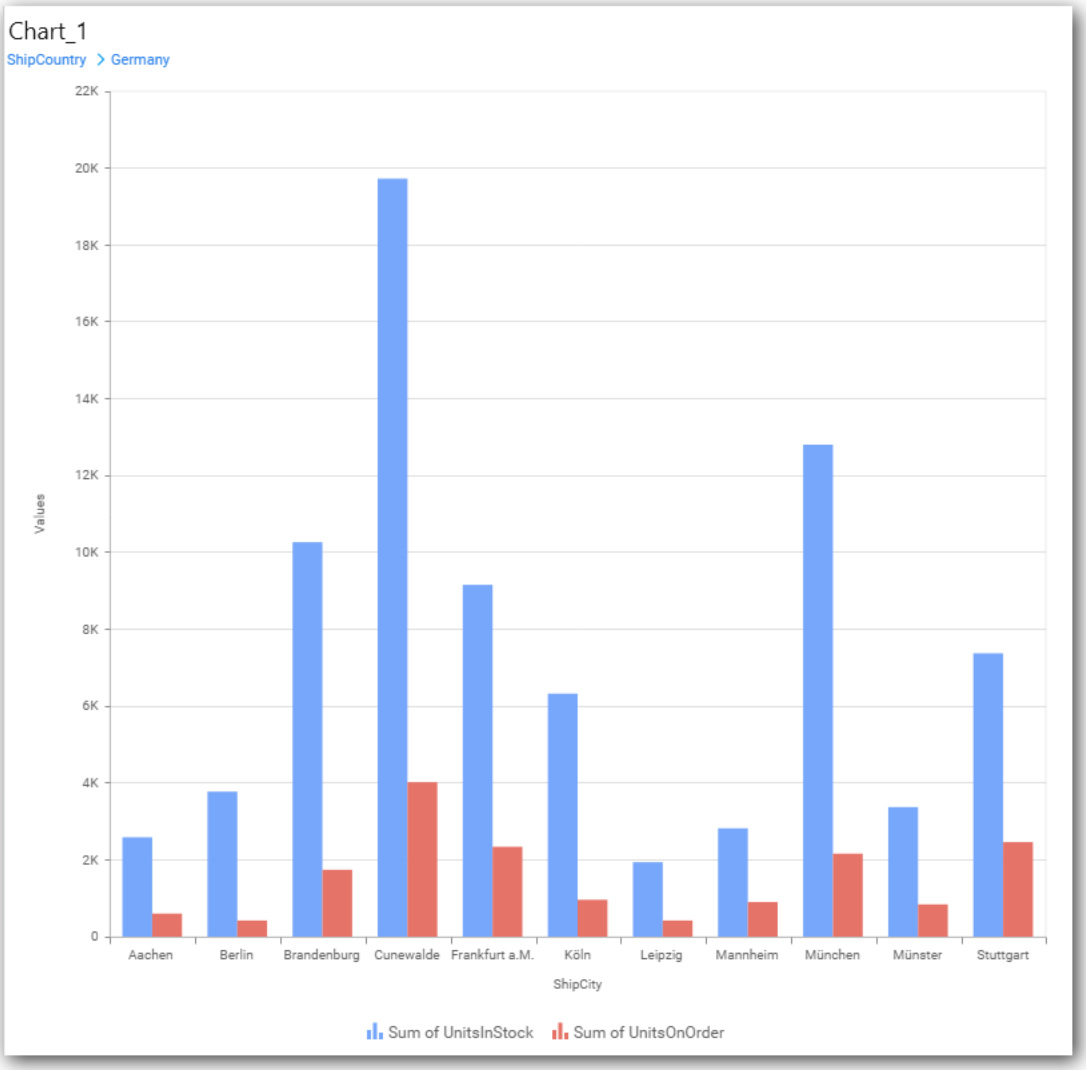
Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View

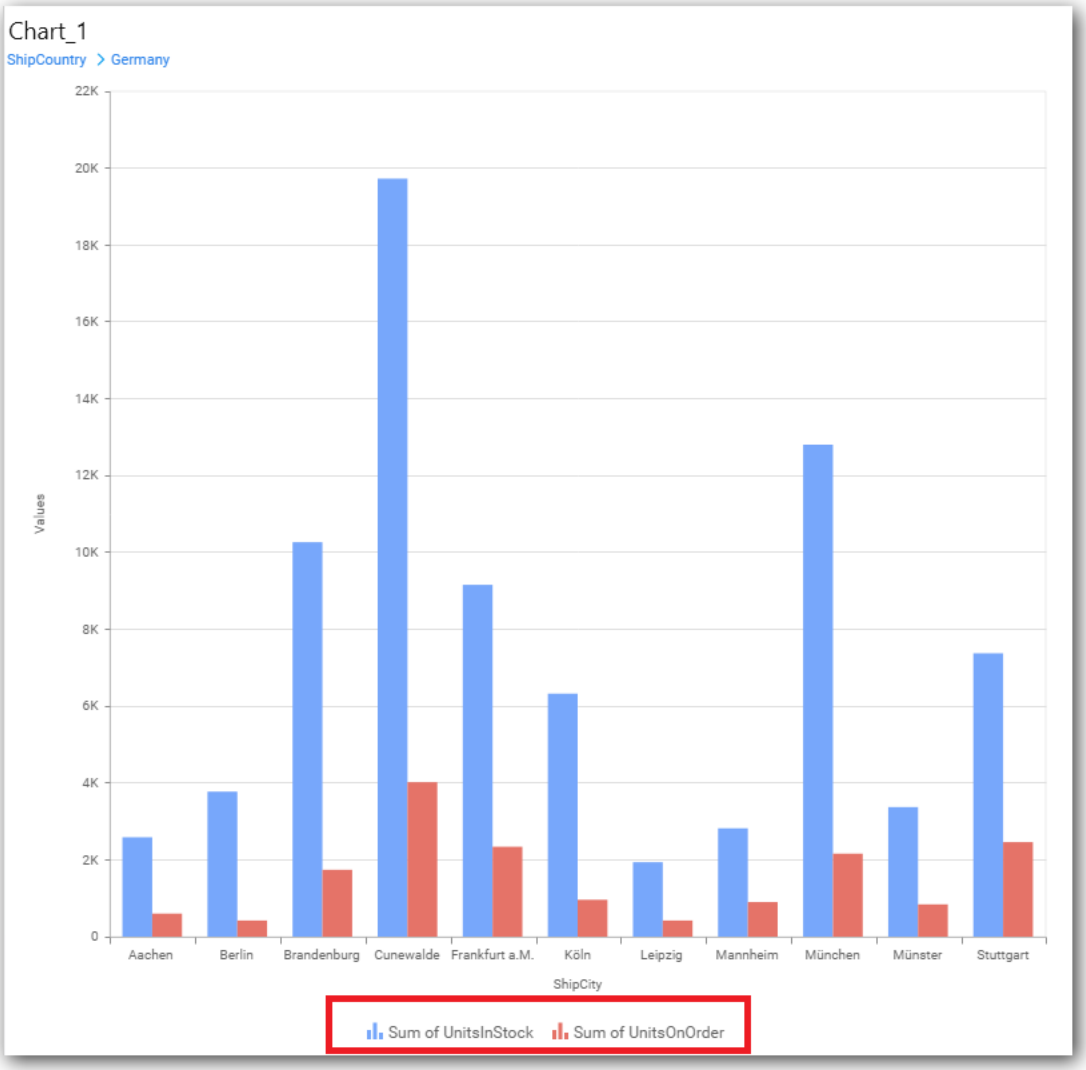


Drilled View



Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

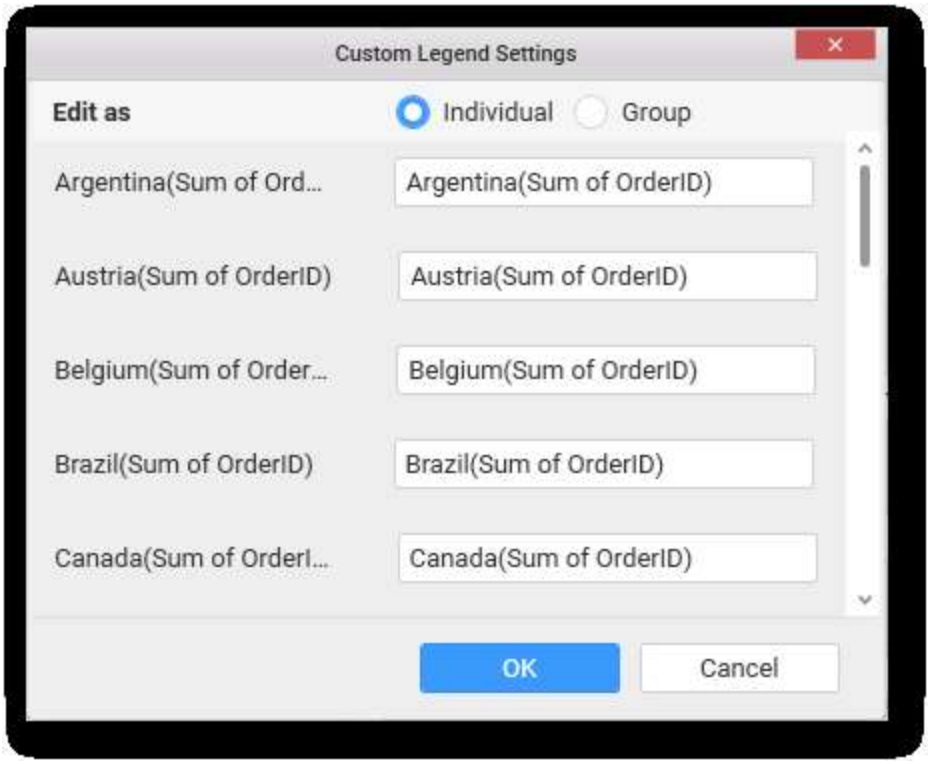
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

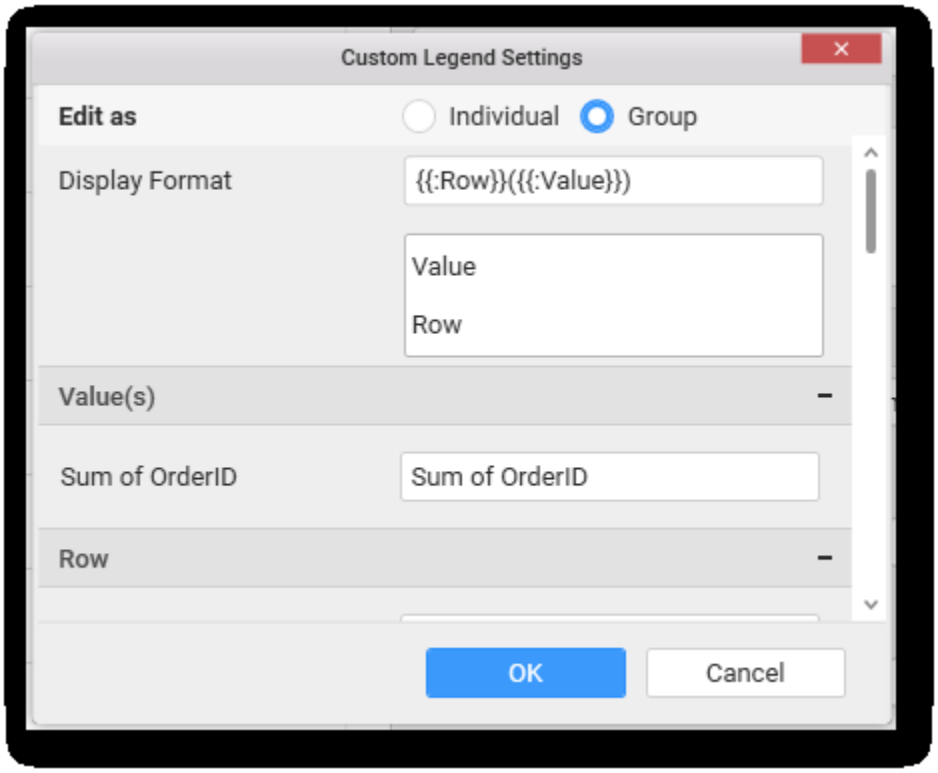
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.



Group

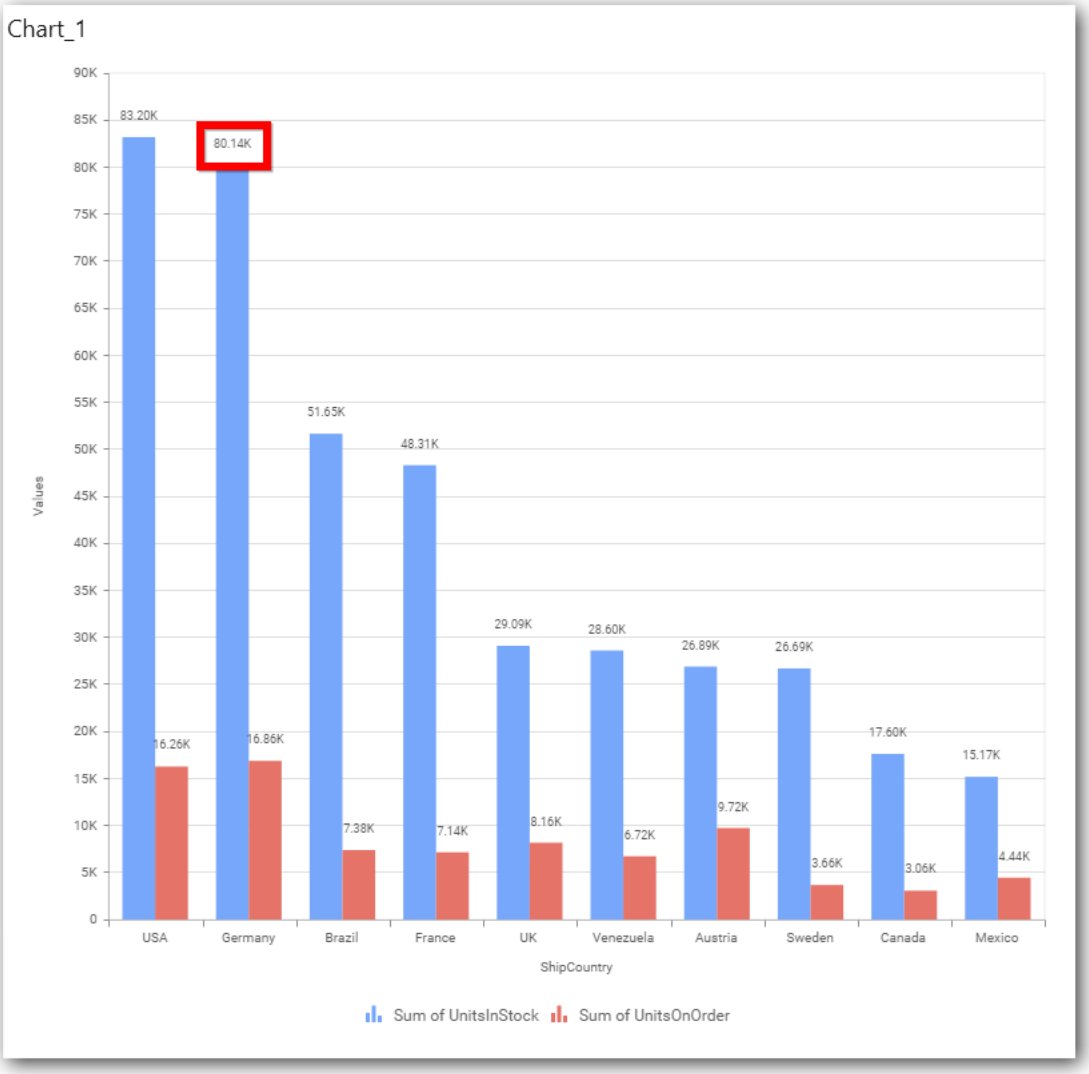
Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)

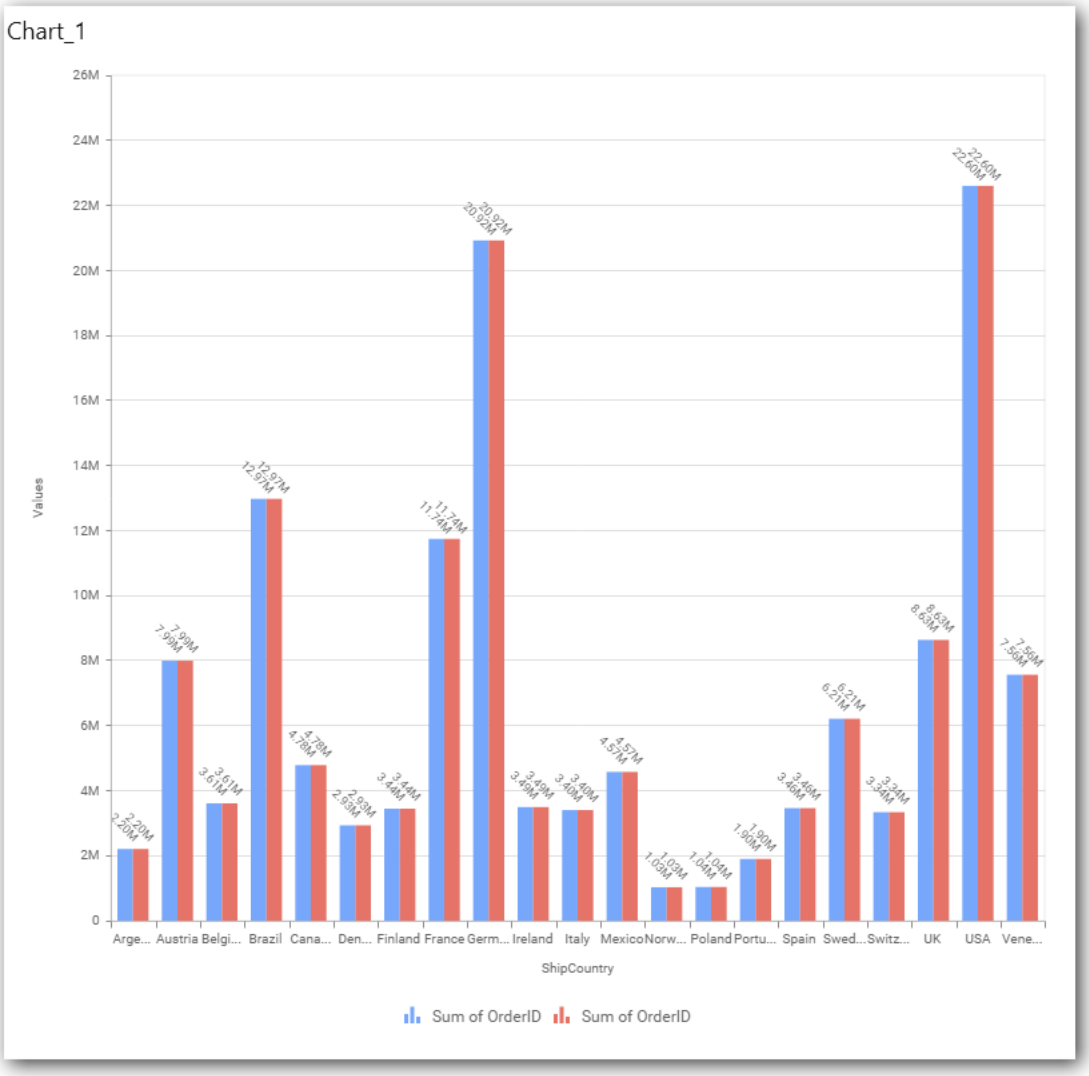
Show Value Labels

This allows you to toggle the visibility of value labels.



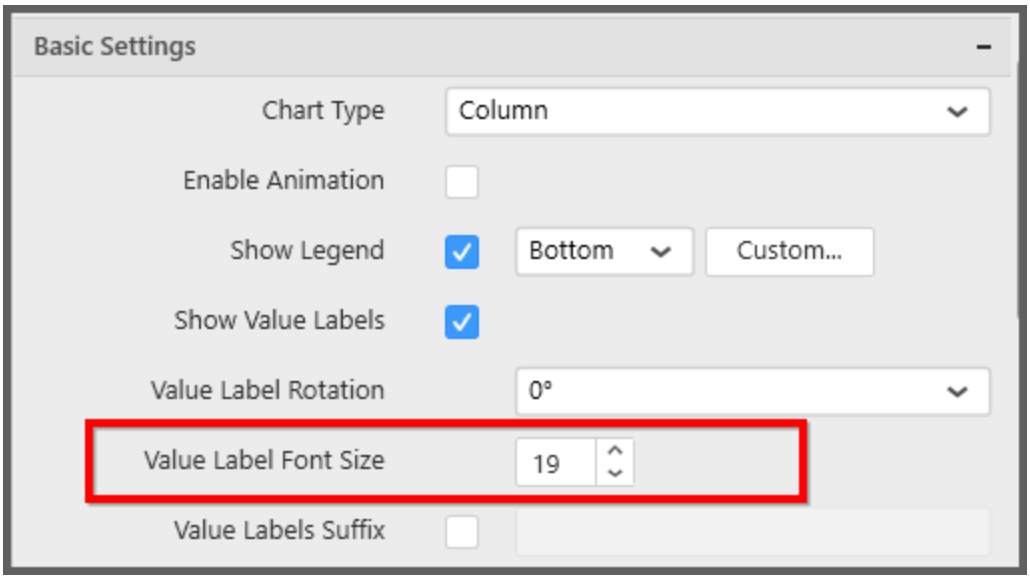
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



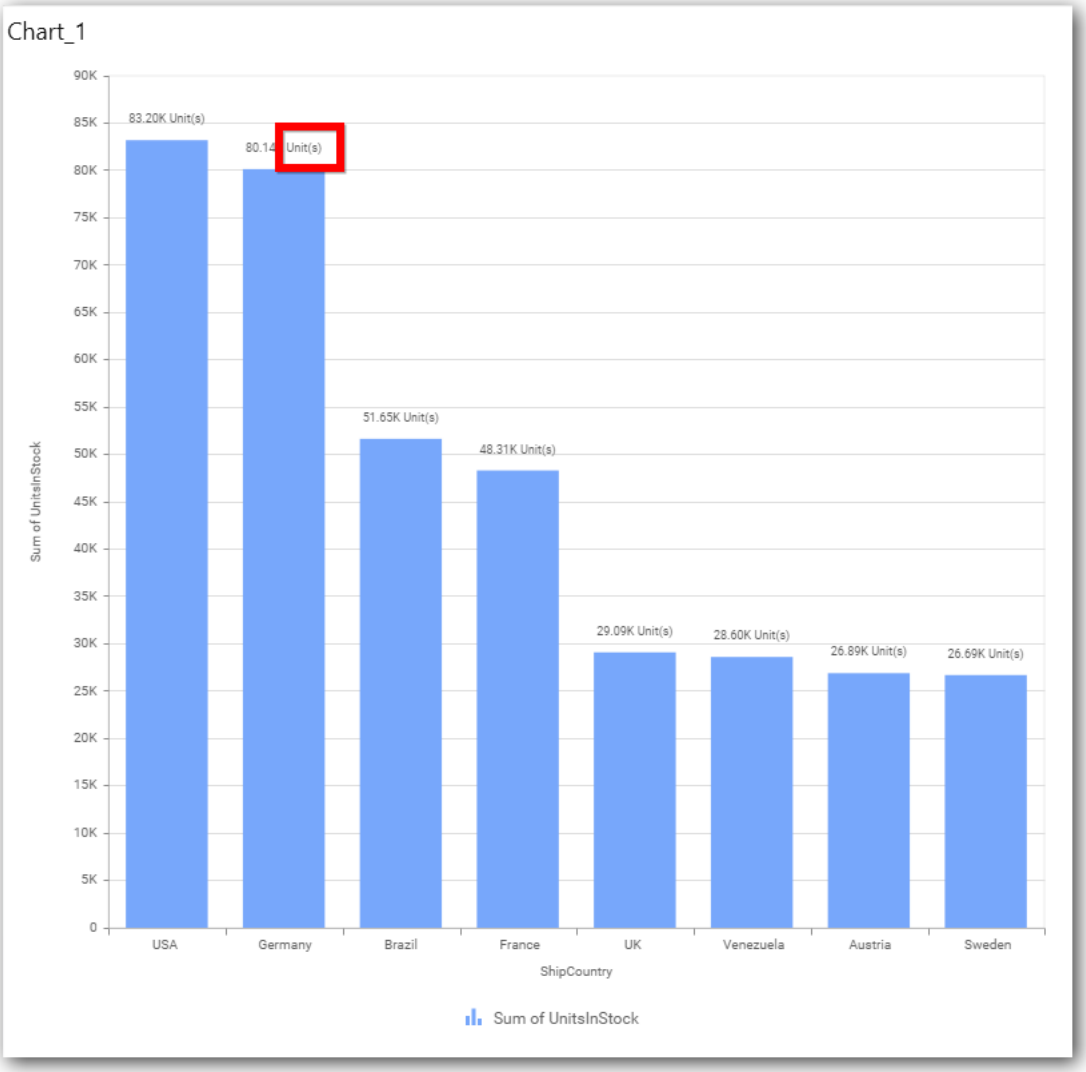
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

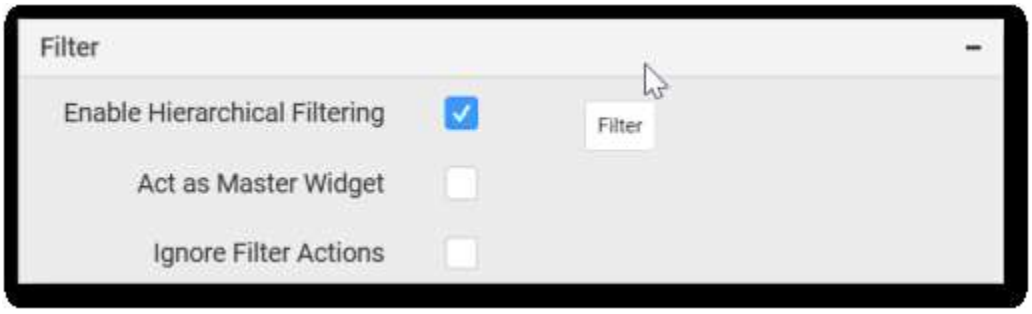


Value Labels Suffix

This allows you to set suffix to the value labels.



Filter Settings



Hierarchical Filtering

This allows you to define the behavior of top **n** filtering which can be flat or hierarchical, in this bar chart widget.

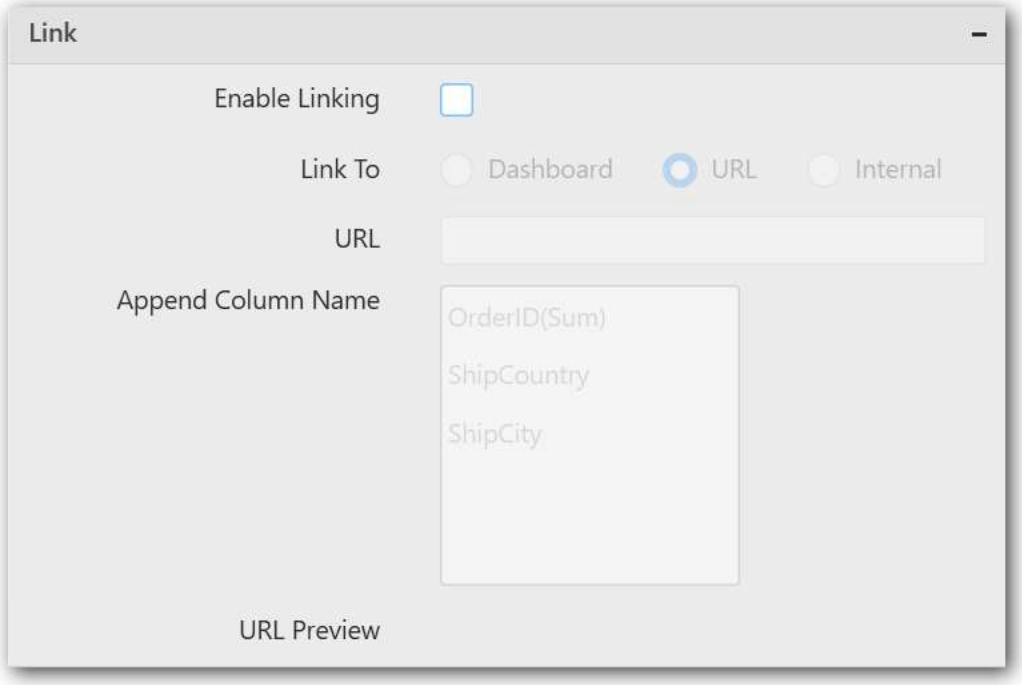
Act as Master Widget

This allows you to define this column chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

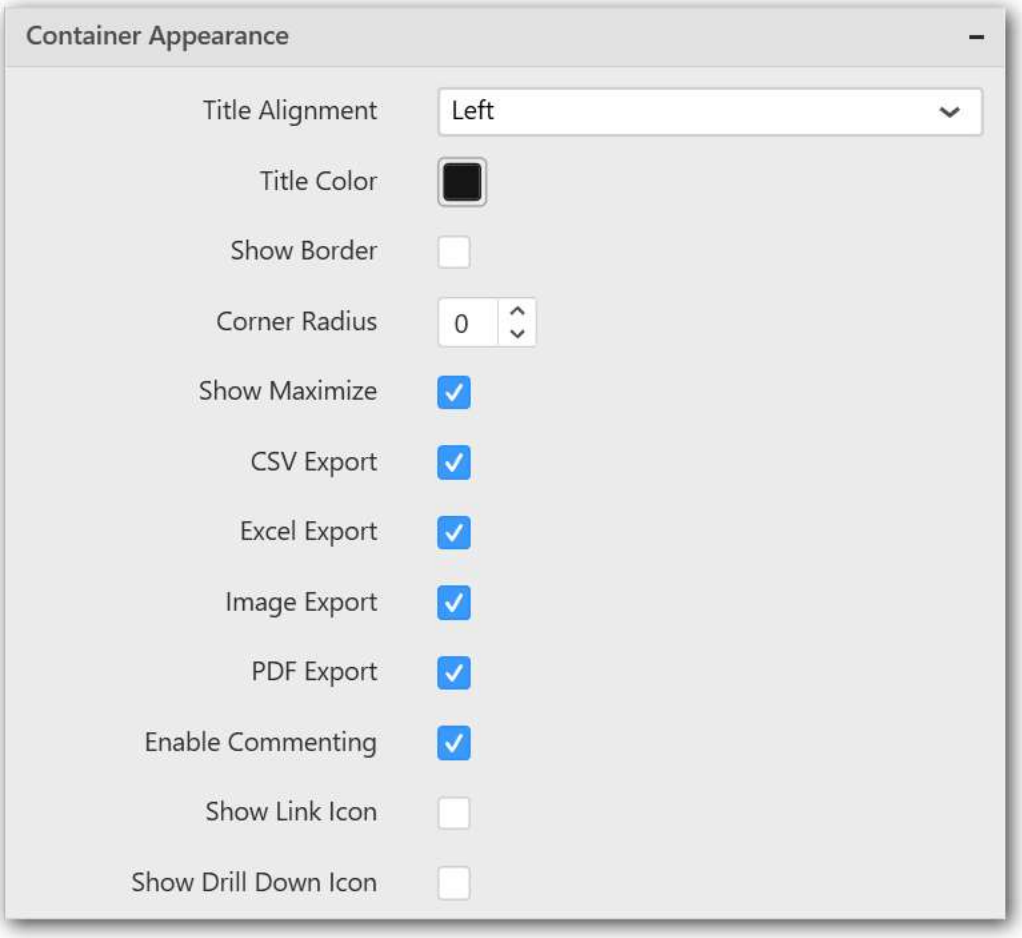
This allows you to define this column chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximized

This allows you to enable/disable the maximized mode of this column chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this column chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this column chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

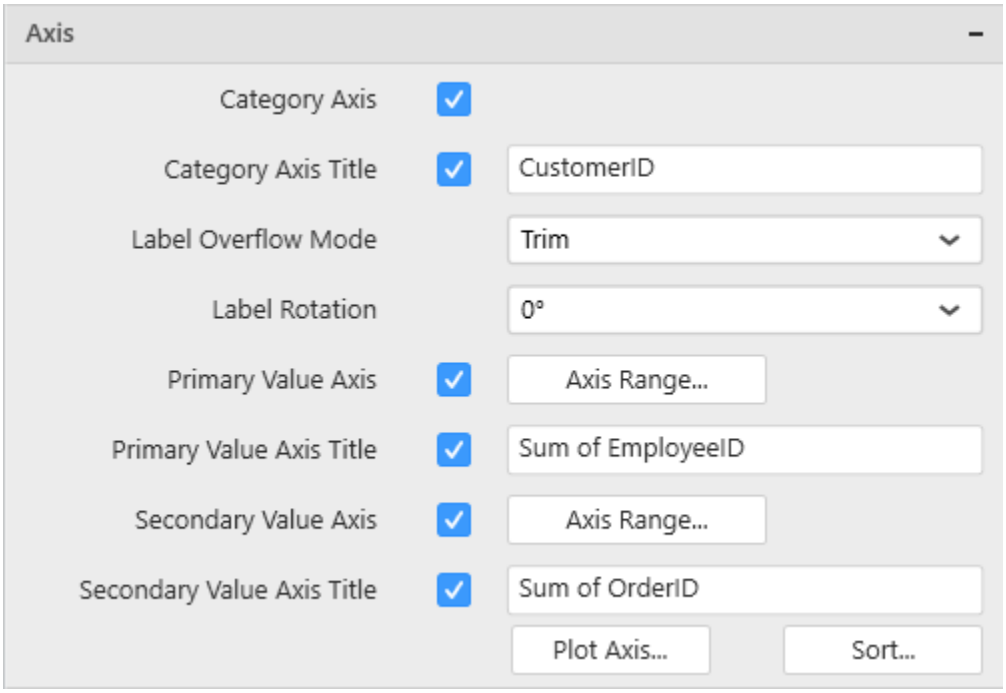
Image Export

This allows you to enable/disable the image export option for this column chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

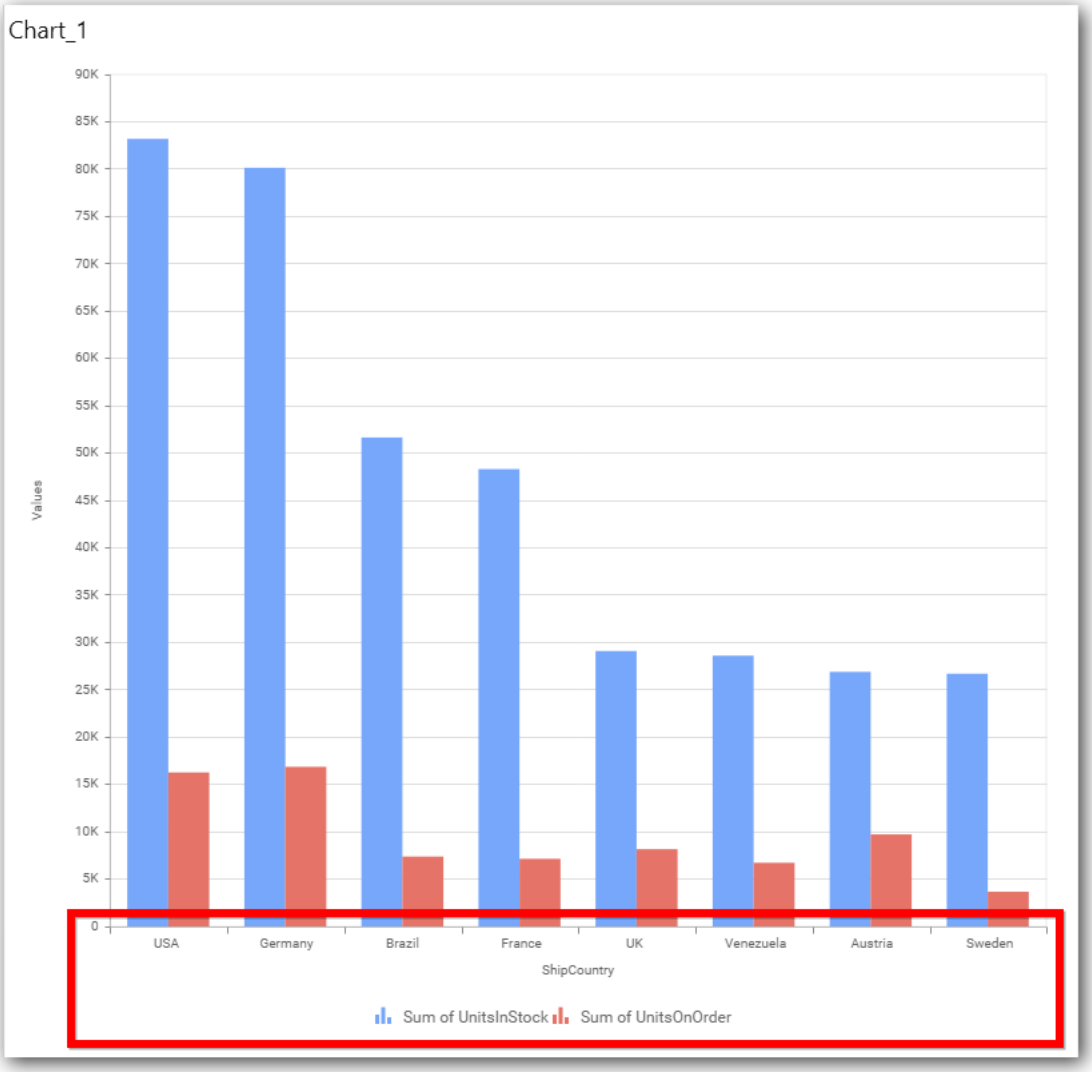
This allows you to enable/disable comment for dashboard widget. For more details refer here

Axis Settings



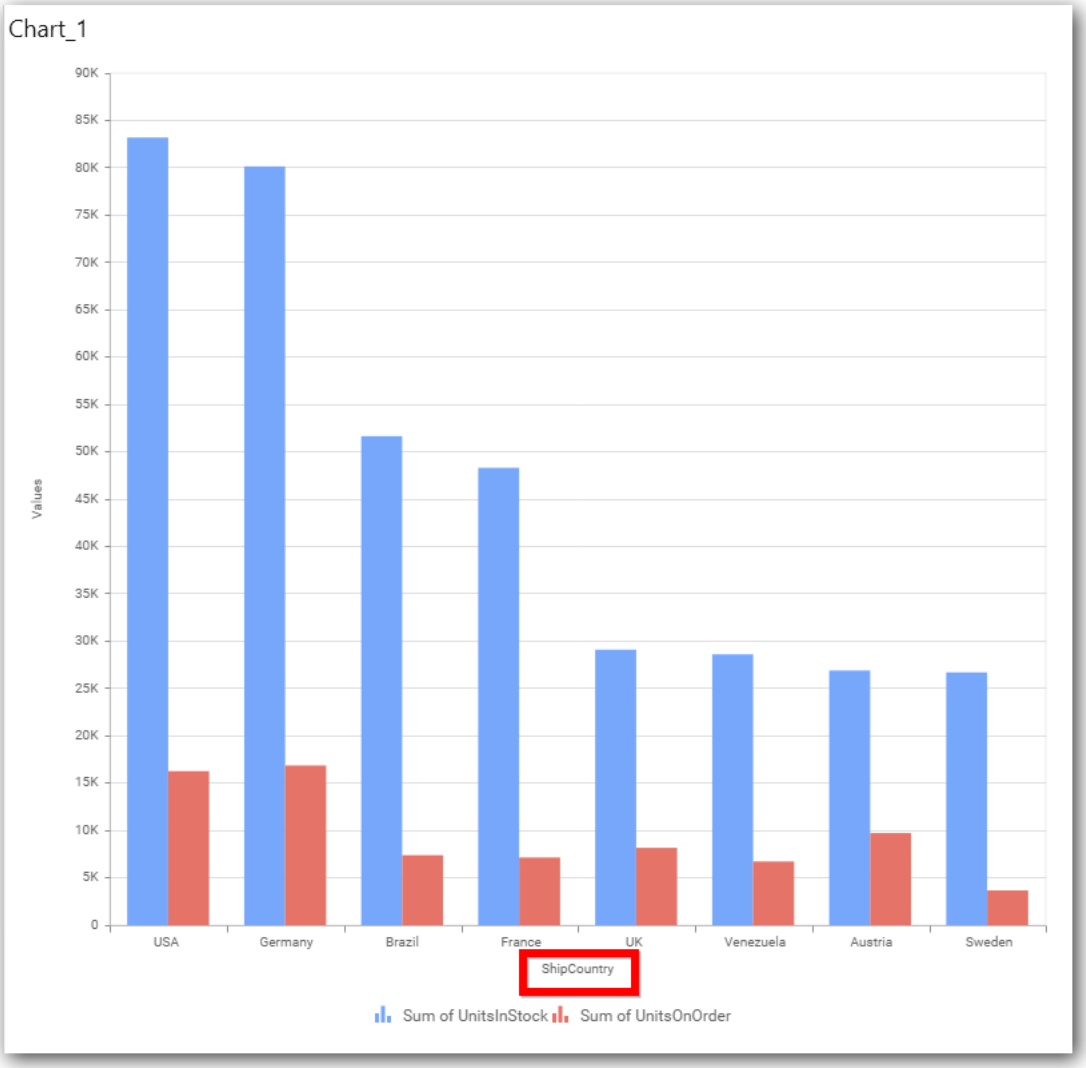
Category Axis

This allows you to enable/edit the option of Category Axis. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

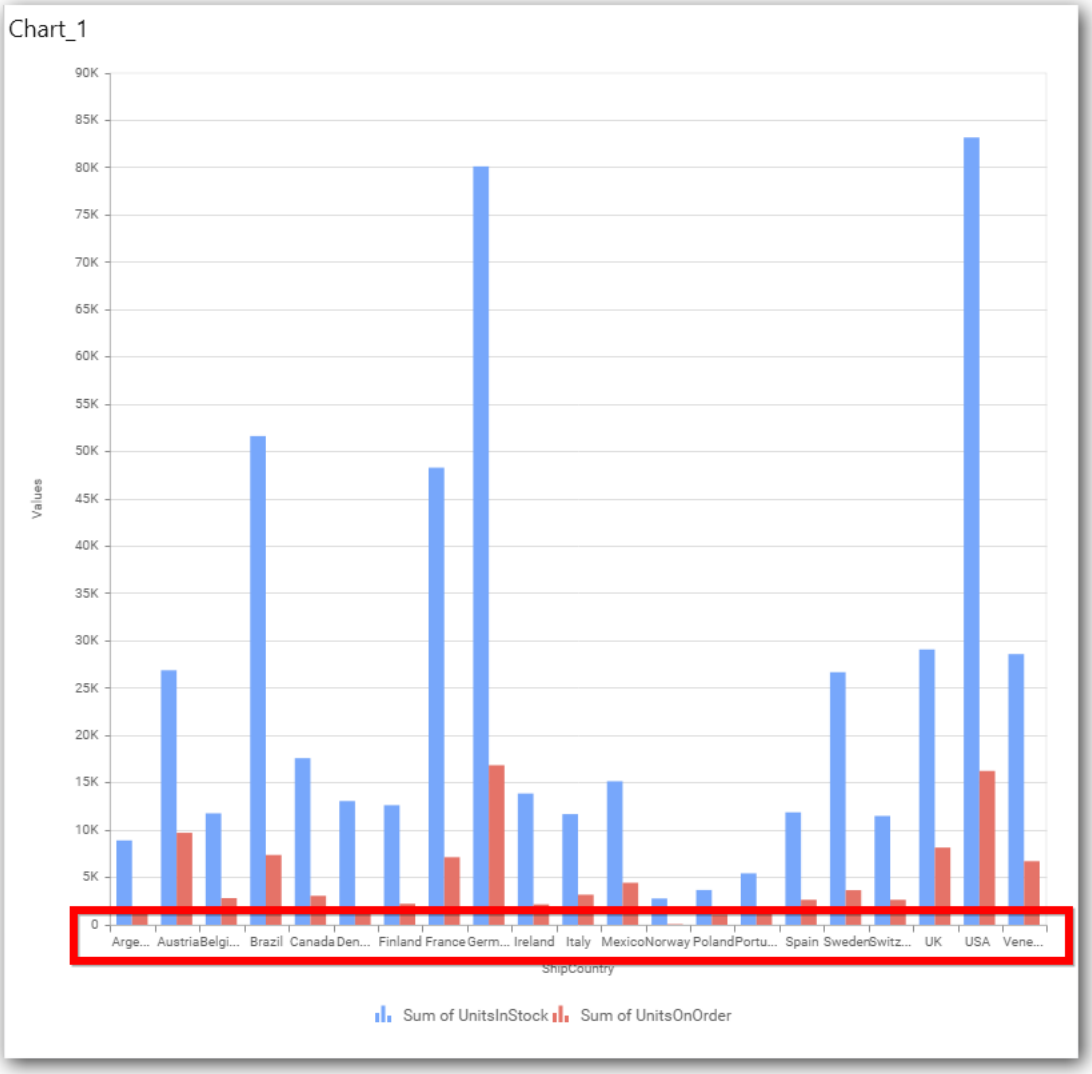


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

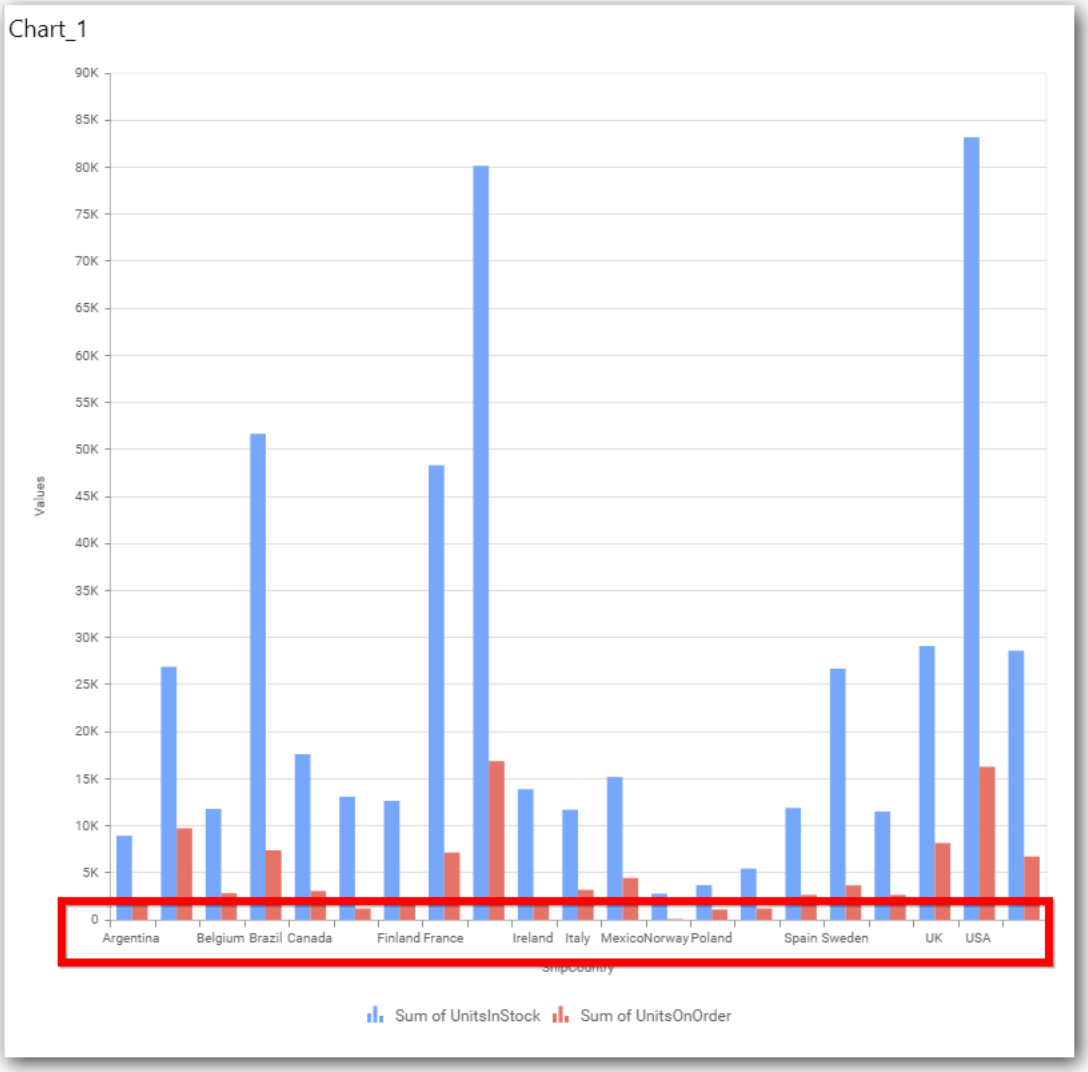
Trim

This option trims the end of overlapping label in the axis.



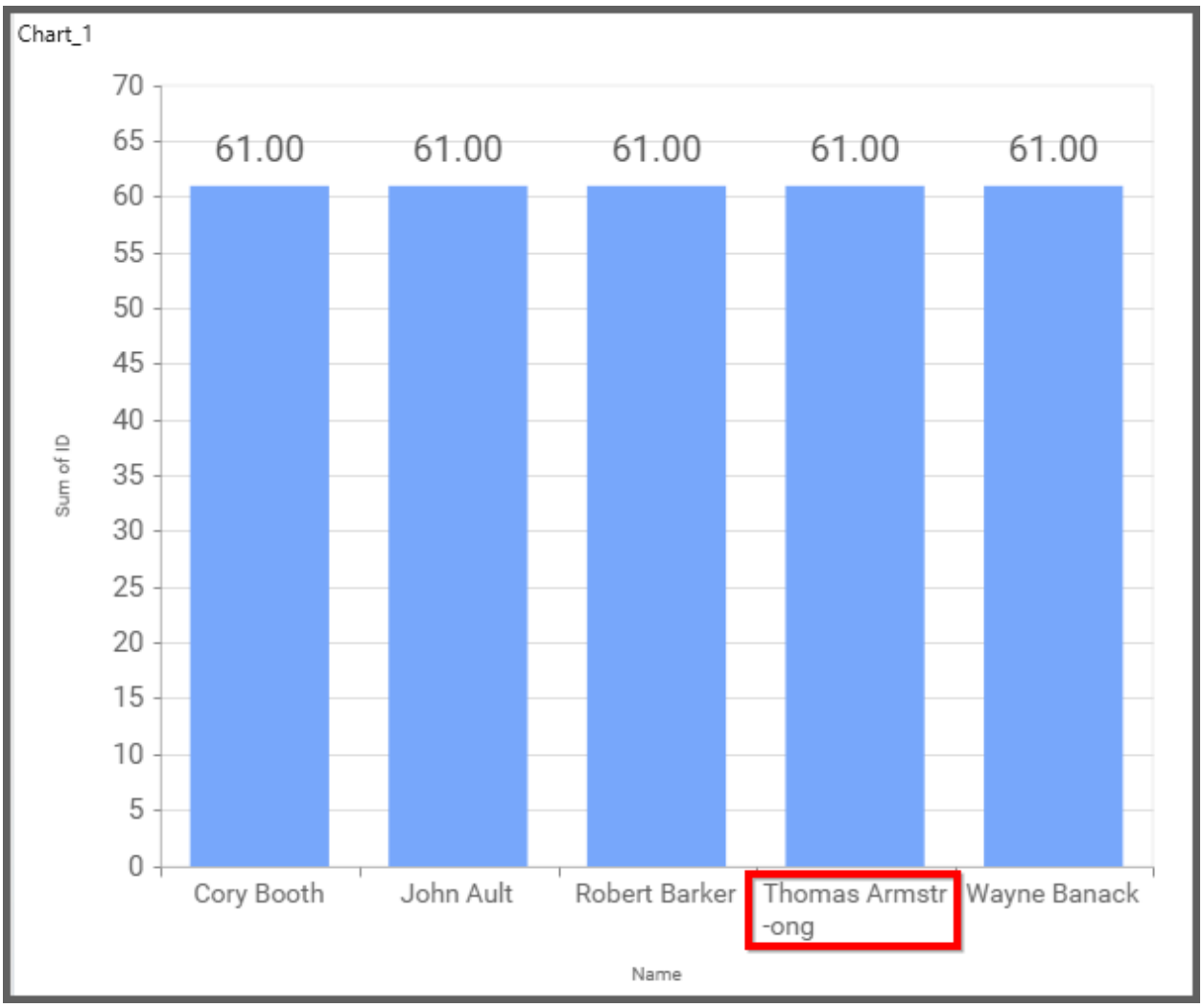
Hide

This option hides the overlapping label in the axis.



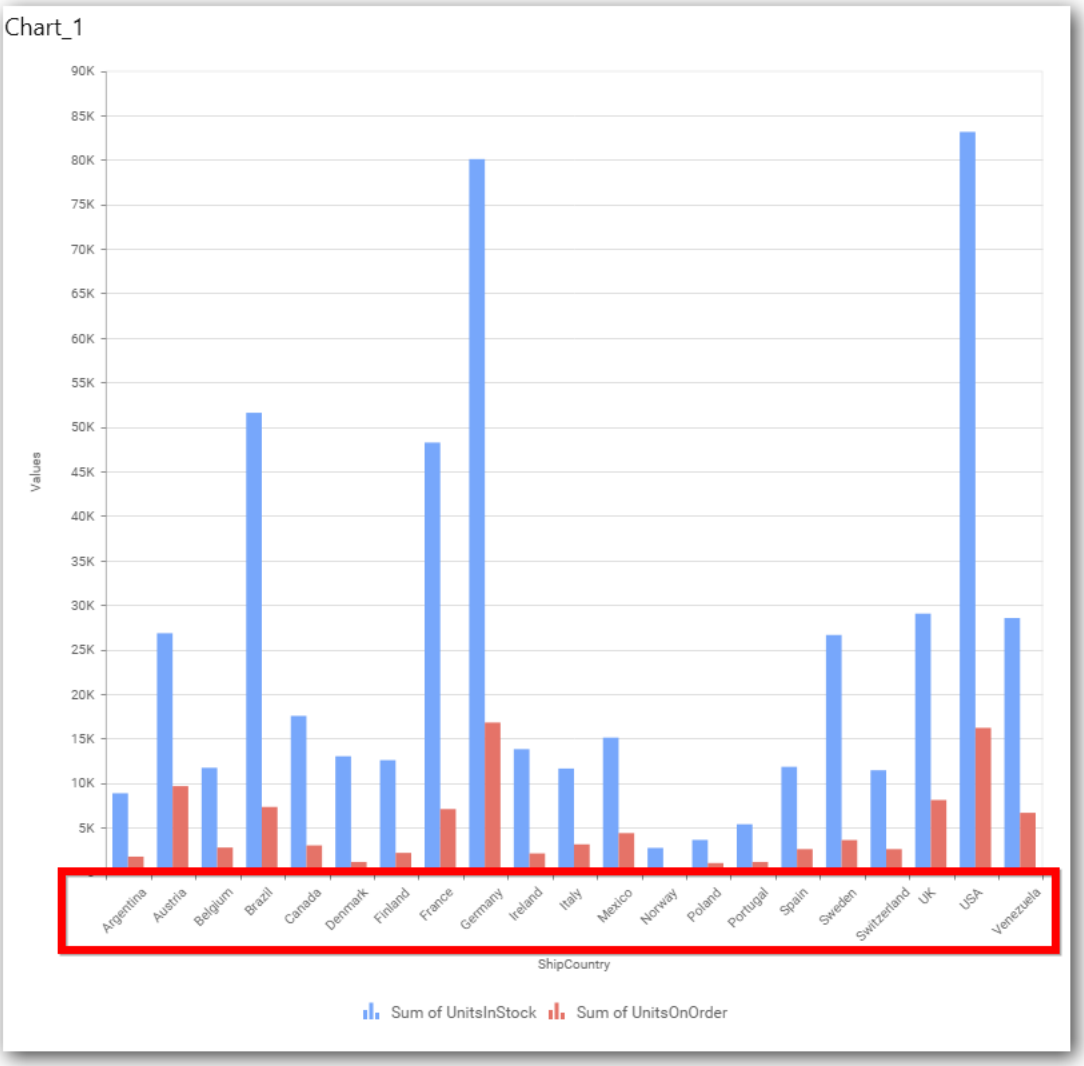
Wrap

This option wraps the lengthy label text in the axis.



Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

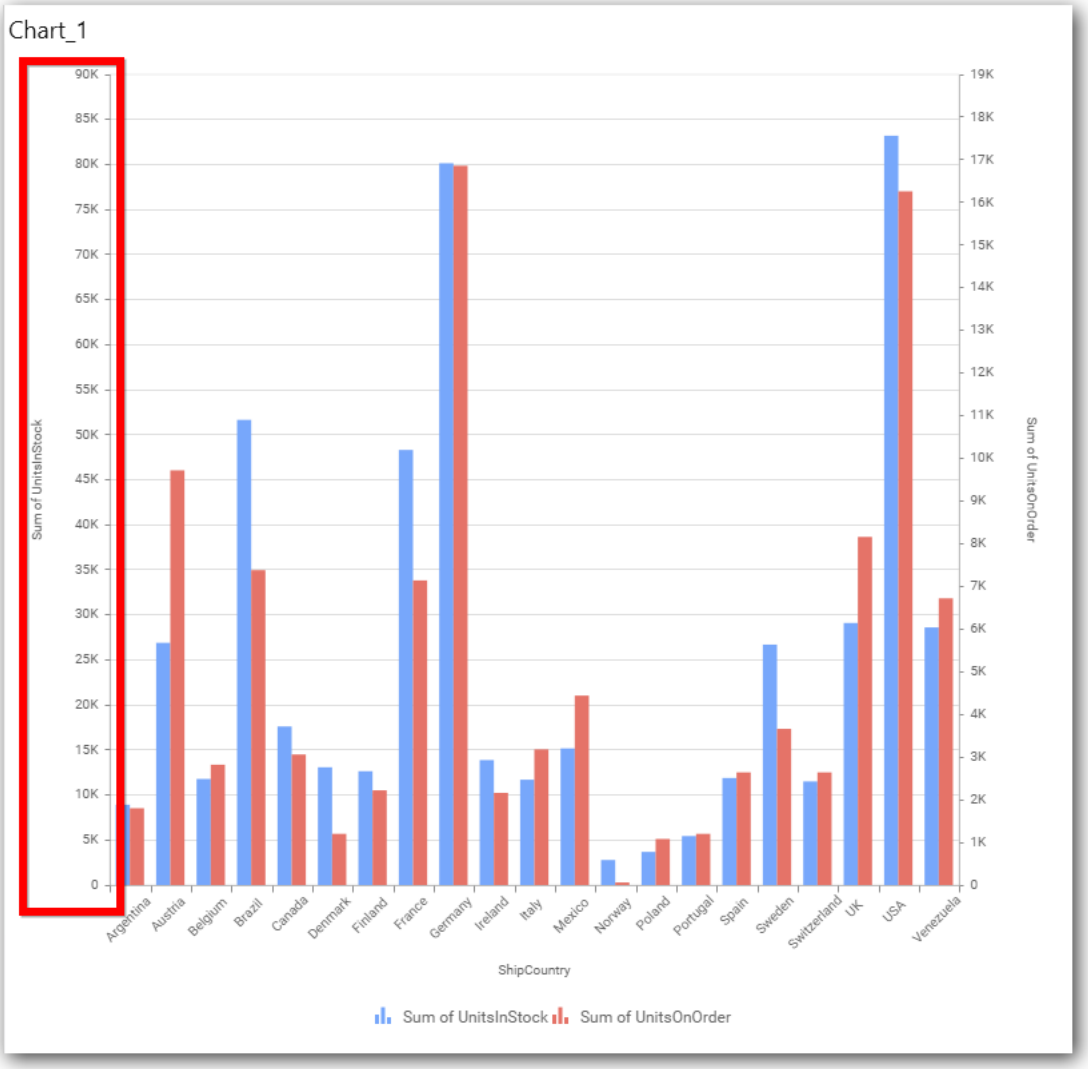


Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.

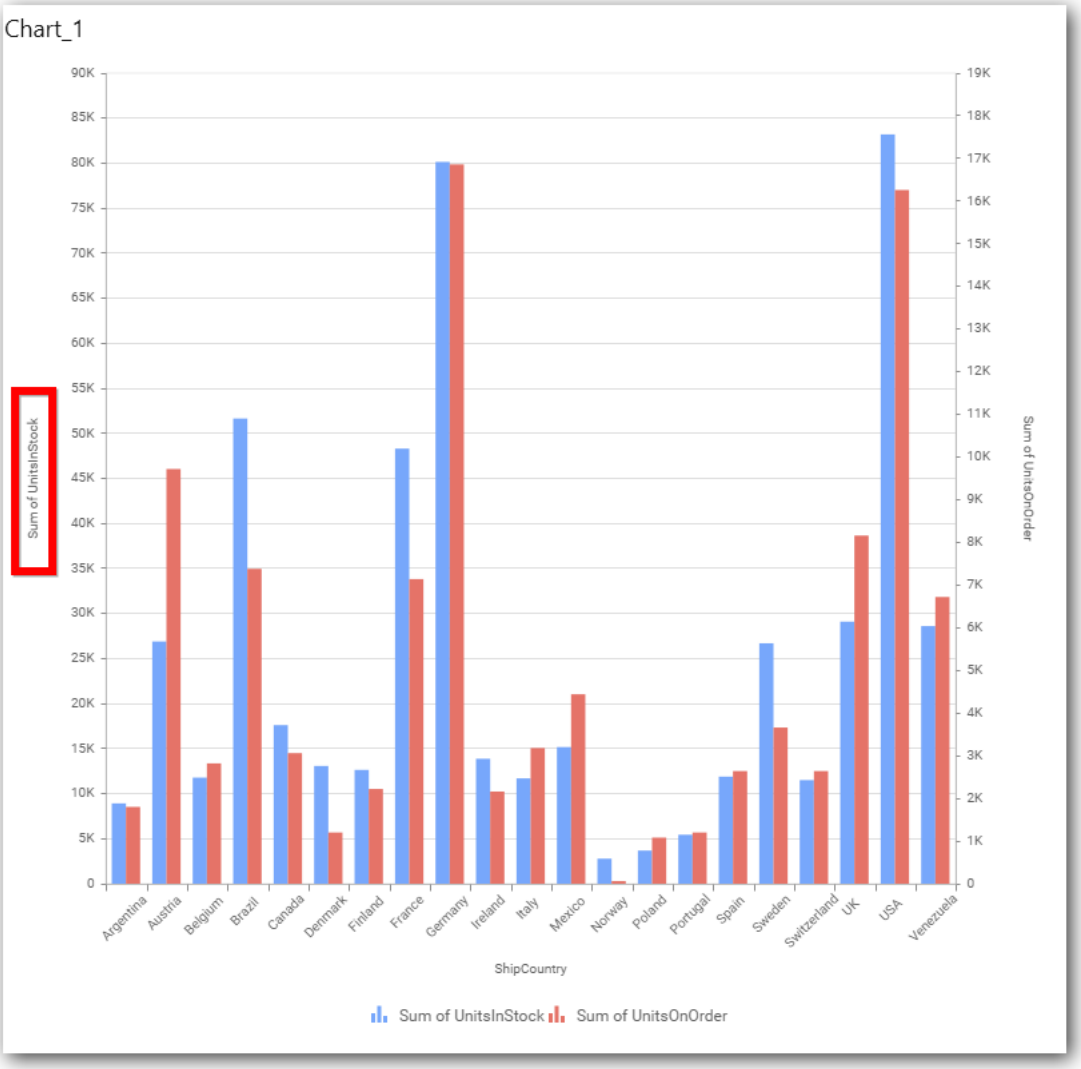
Primary Value Axis

This allows you to enable/edit the option of Primary Value Axis. It will reflect in chart area y-axis name.



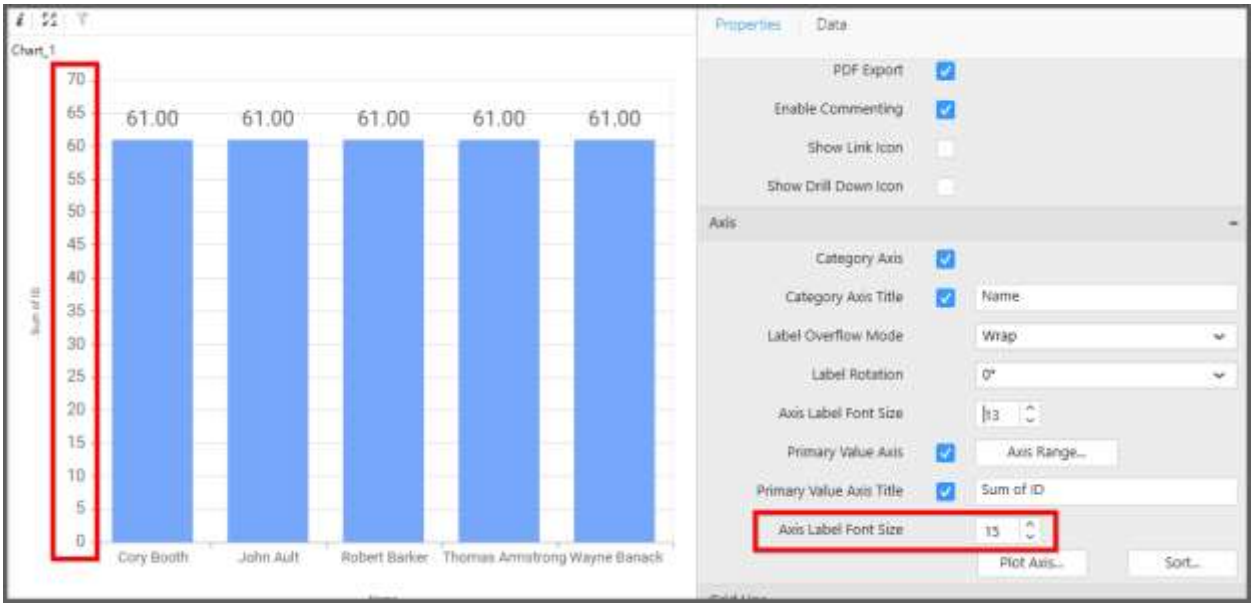
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

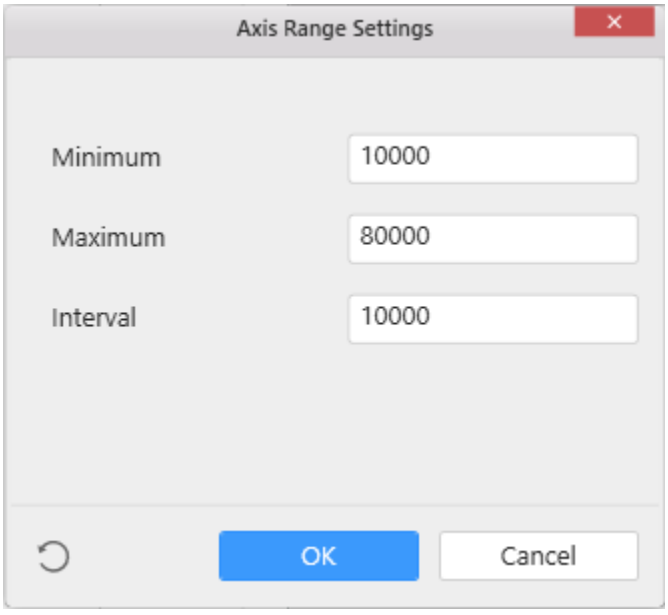


Primary Value Axis Range

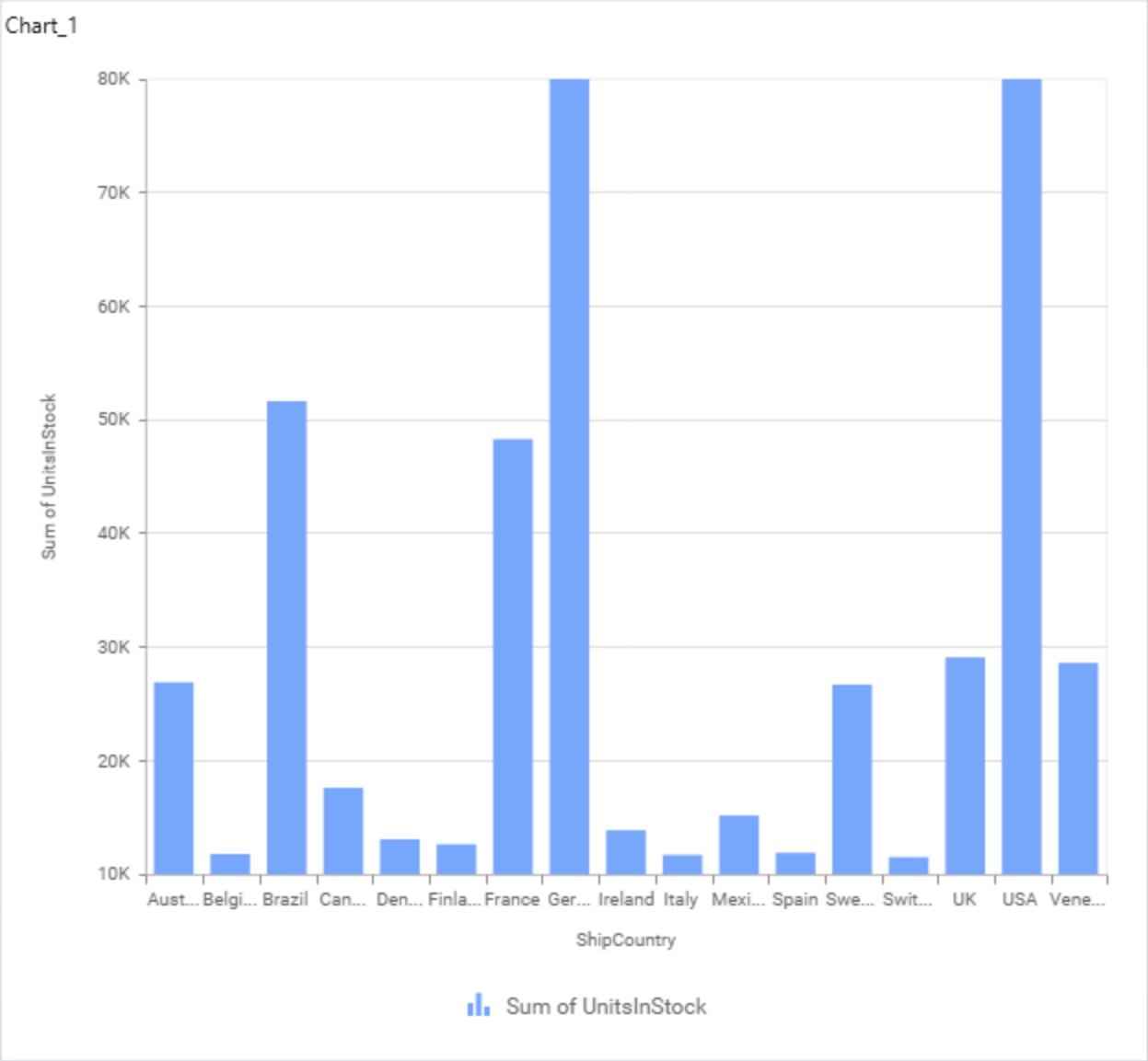
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.

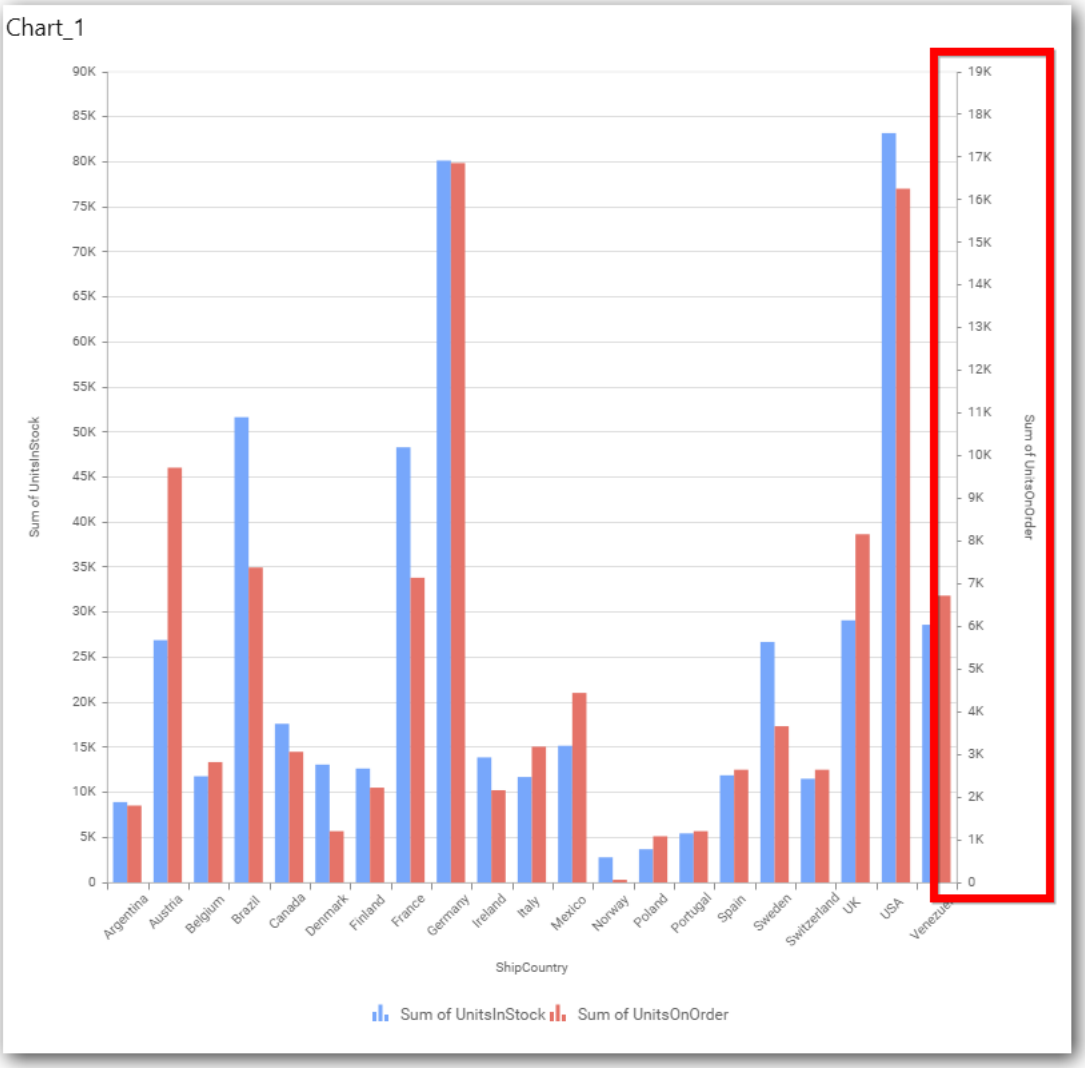


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



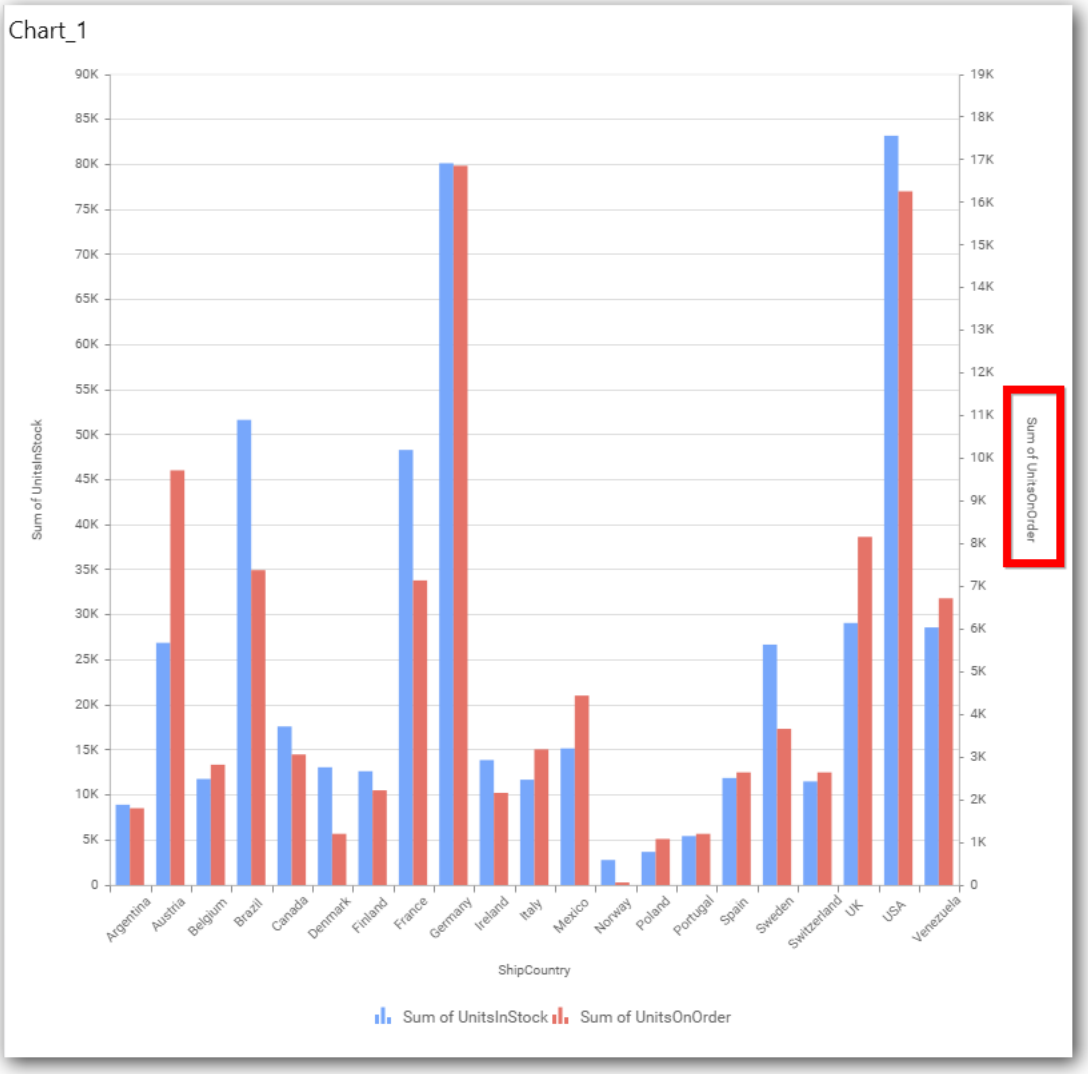
Secondary Value Axis

This allows you to enable/edit the option of Secondary Value Axis. It will reflect in chart area secondary y-axis name.



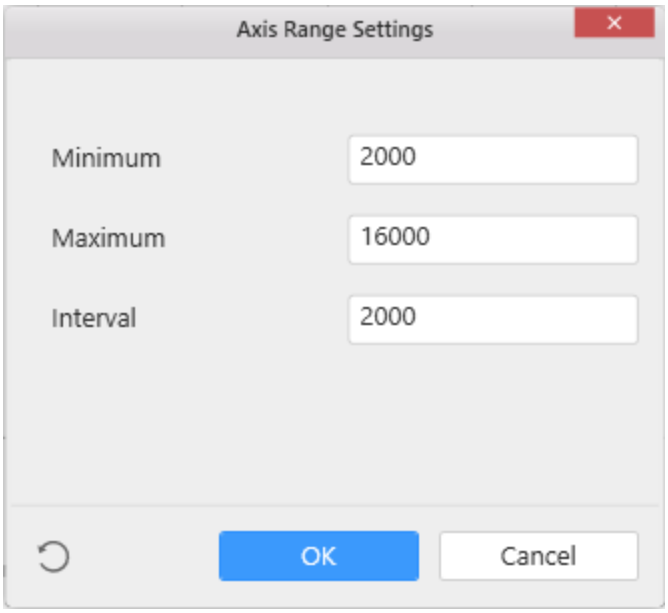
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

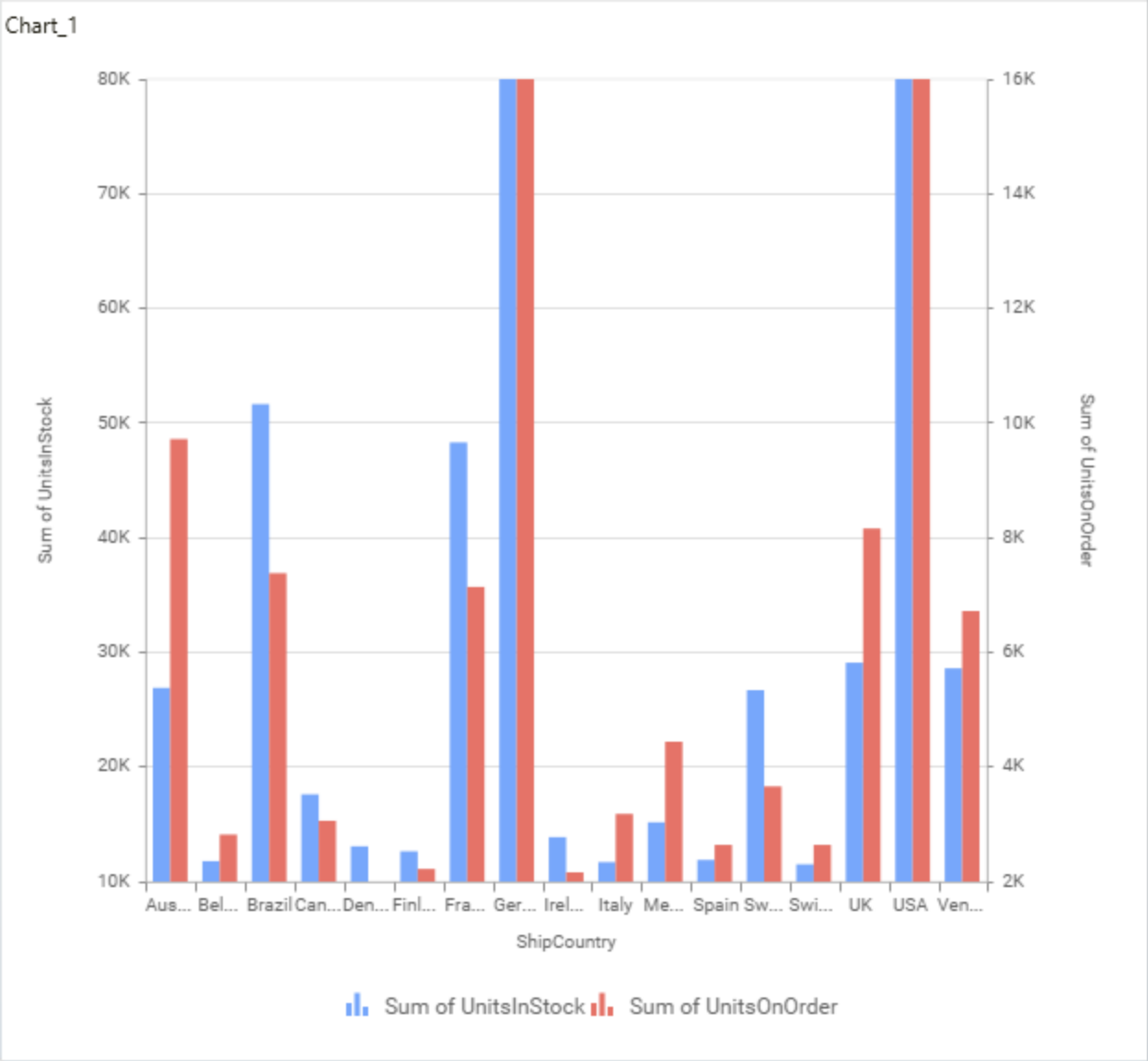


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

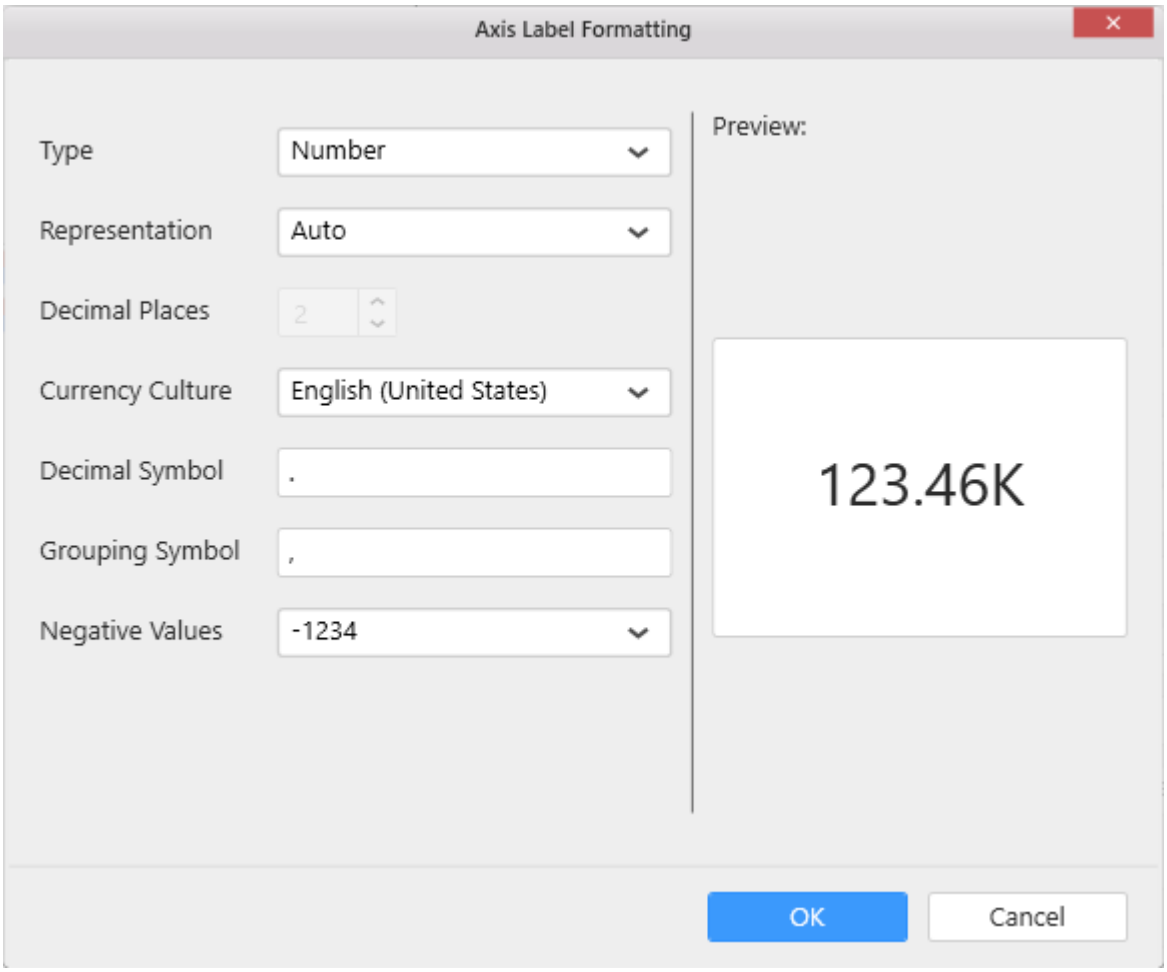


Plot Axis Settings

This allows you to define which measure column need to be plotted against which value axis (primary or secondary).

Axis Label Formatting

This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.



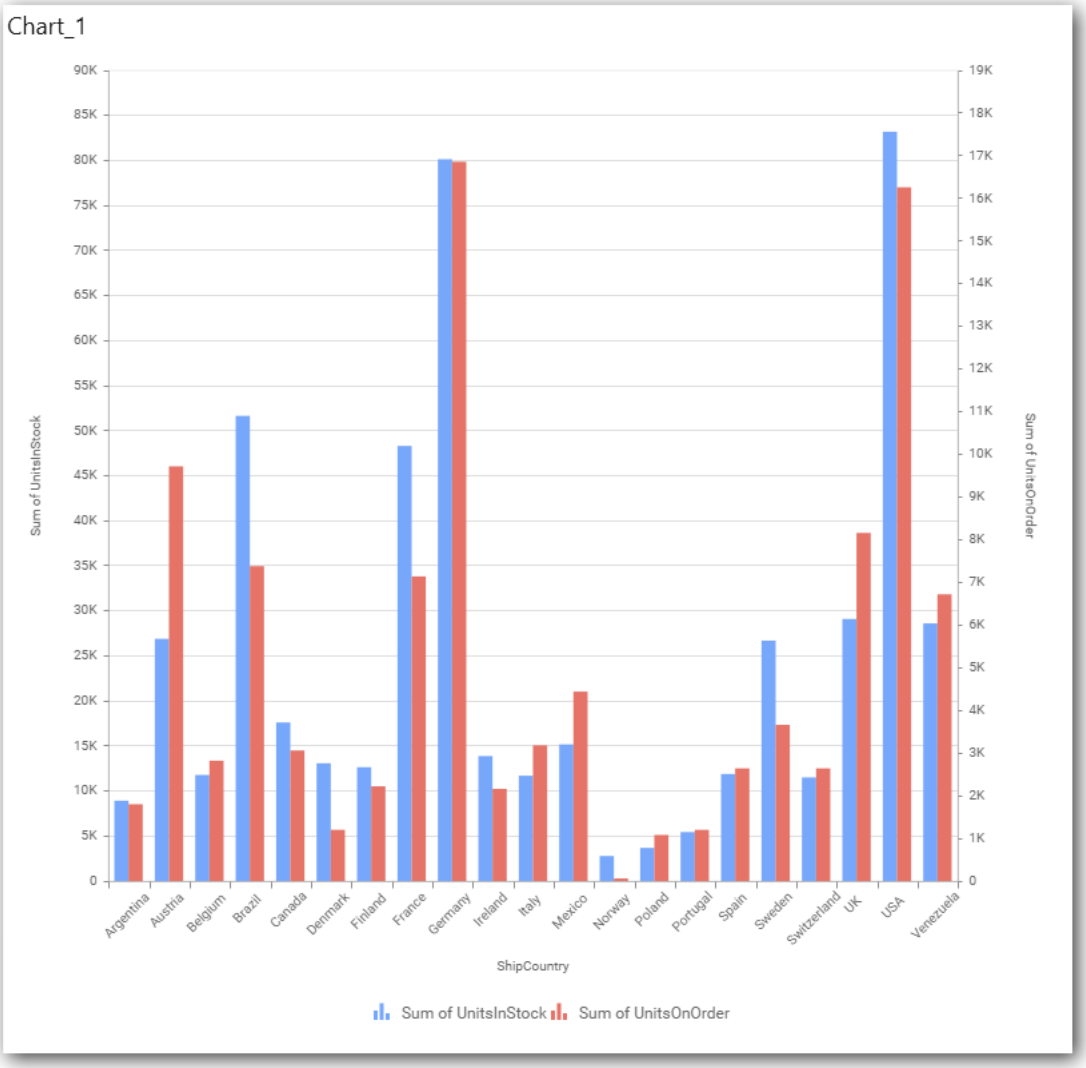
Sort Order

This allows you to define the sort order for each measure column added.

Grid Line Settings

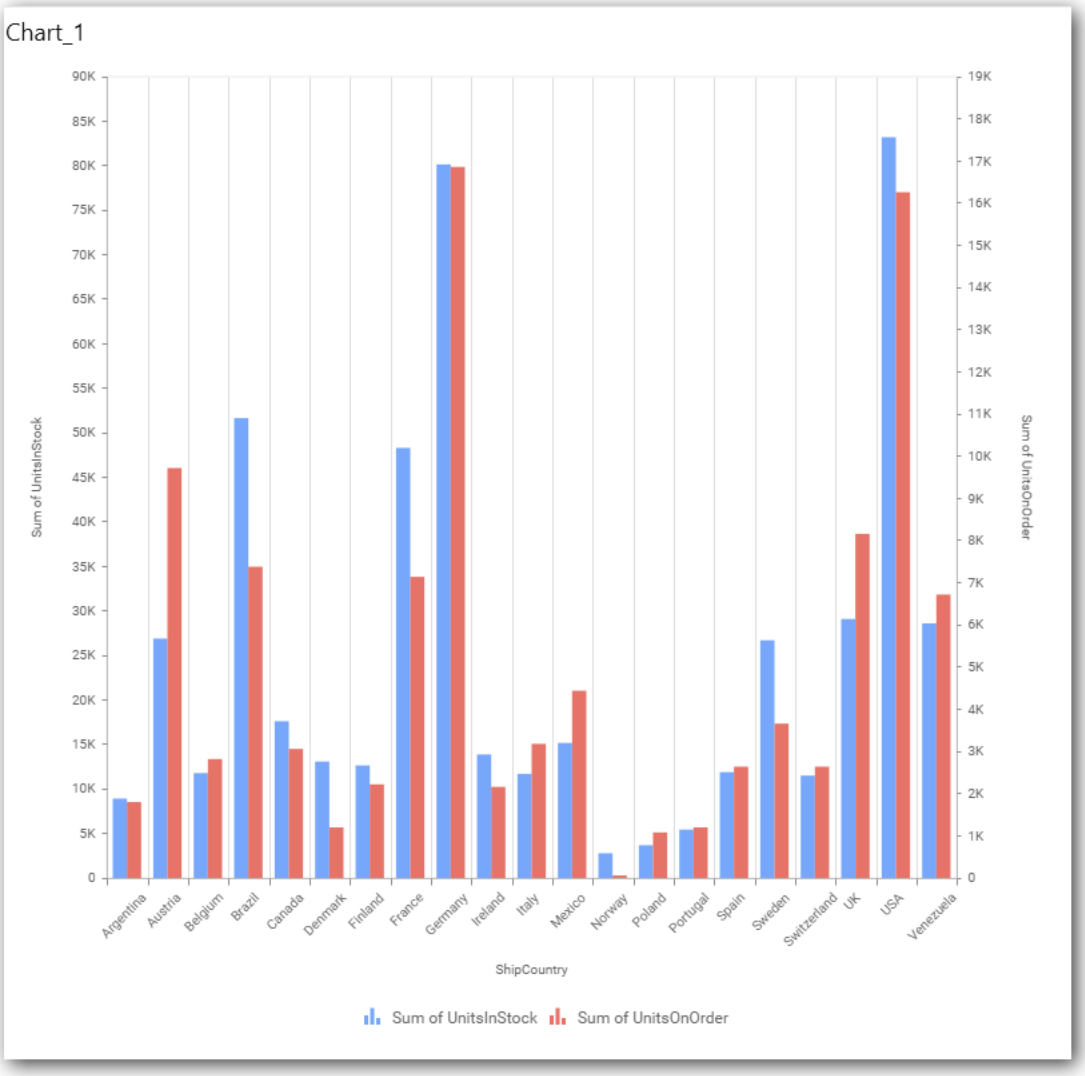
Primary Value Axis

This allows you to toggle the visibility of primary value axis' gridlines.



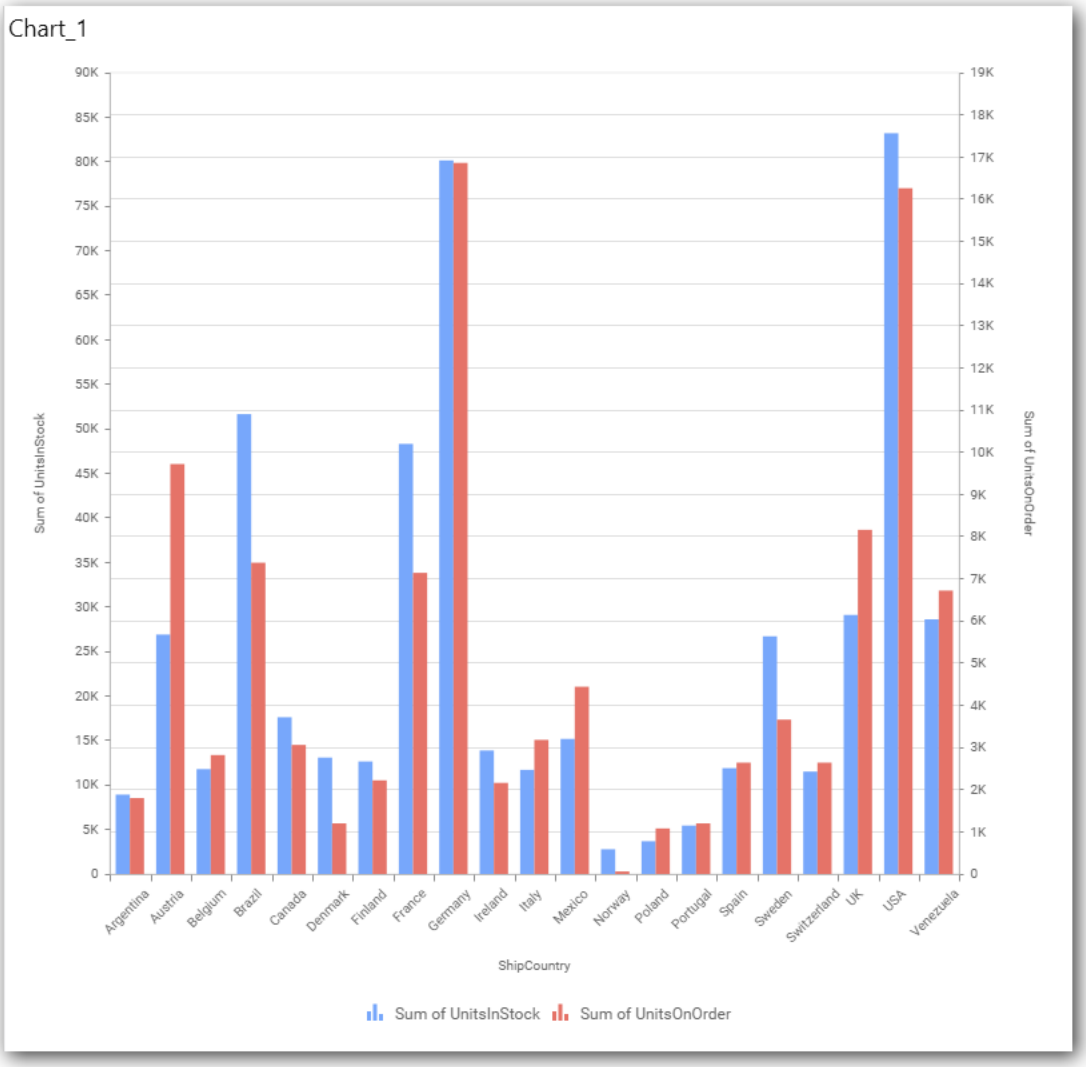
Category Axis

This allows you to toggle the visibility of category axis' gridlines.

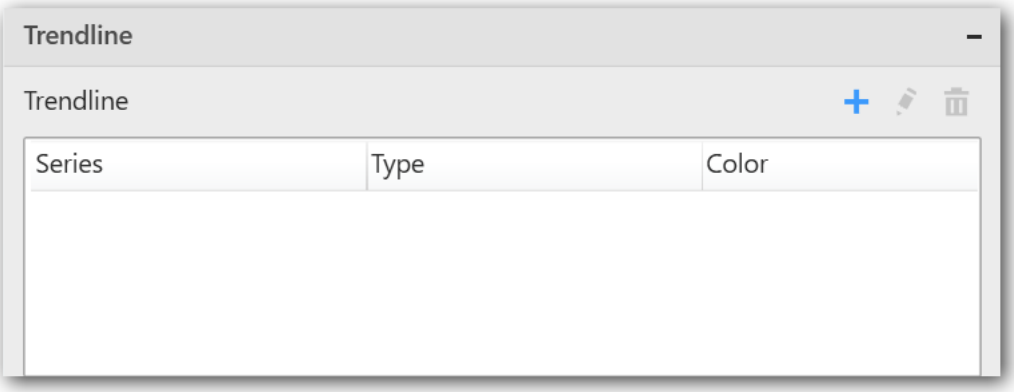


Secondary Value Axis

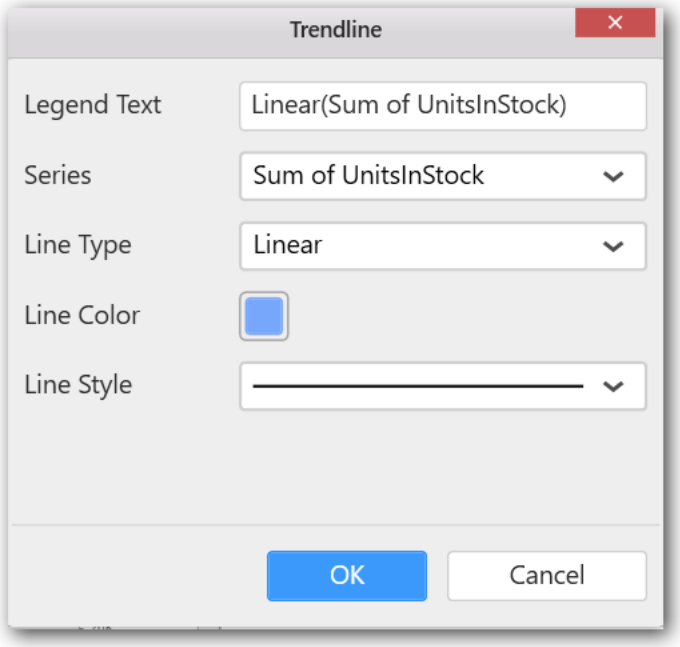
This allows you to toggle the visibility of secondary value axis' gridlines.



Trend line Settings



You can add trend line to chart based on dropped measure that you select. You can also customize its legend text, line type and line color. Trend line is not visible, by default.



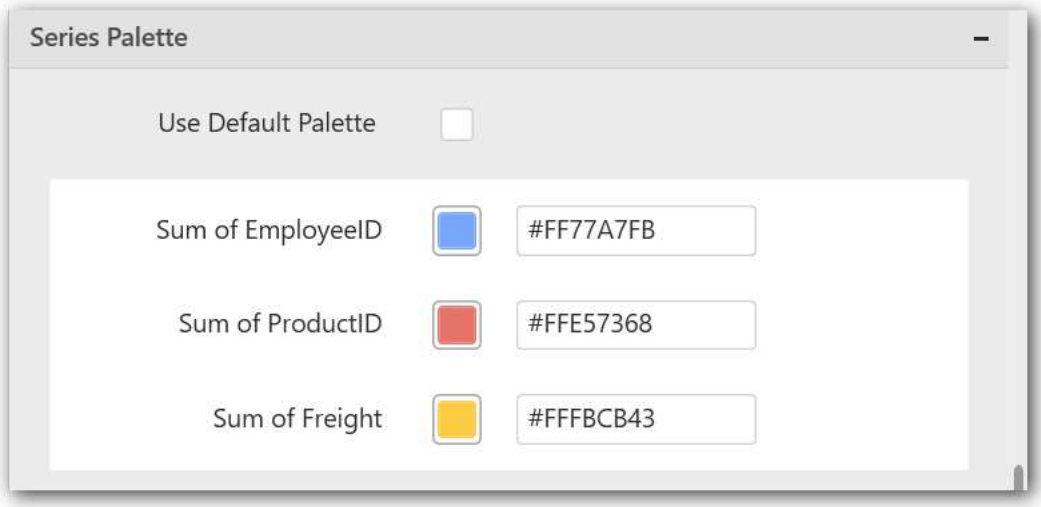
You have options to edit or delete the added trend lines.

Series Palette

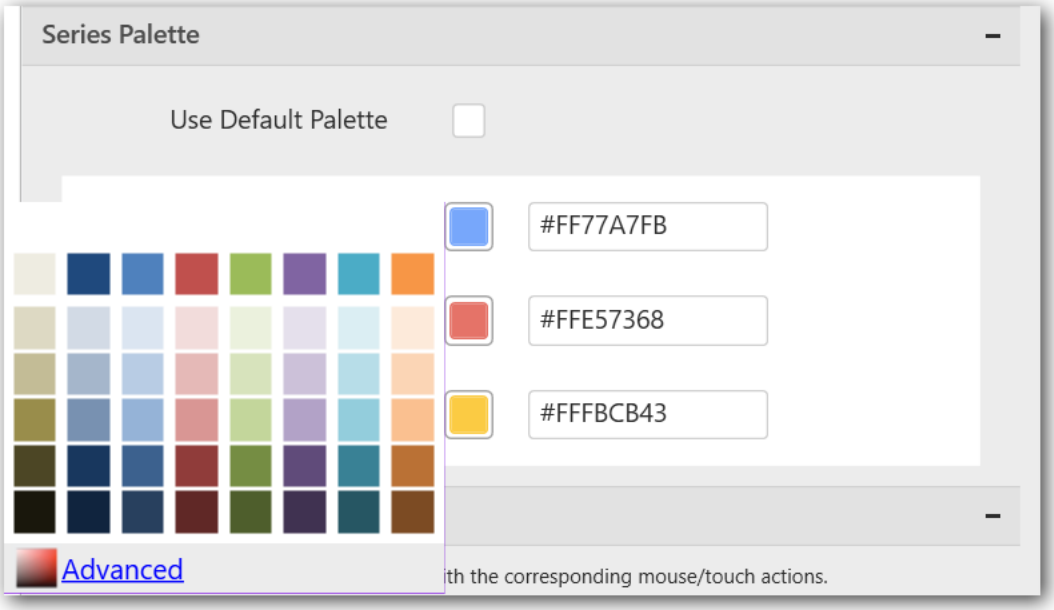
This allows you to customize the chart series color through Series Palette section.

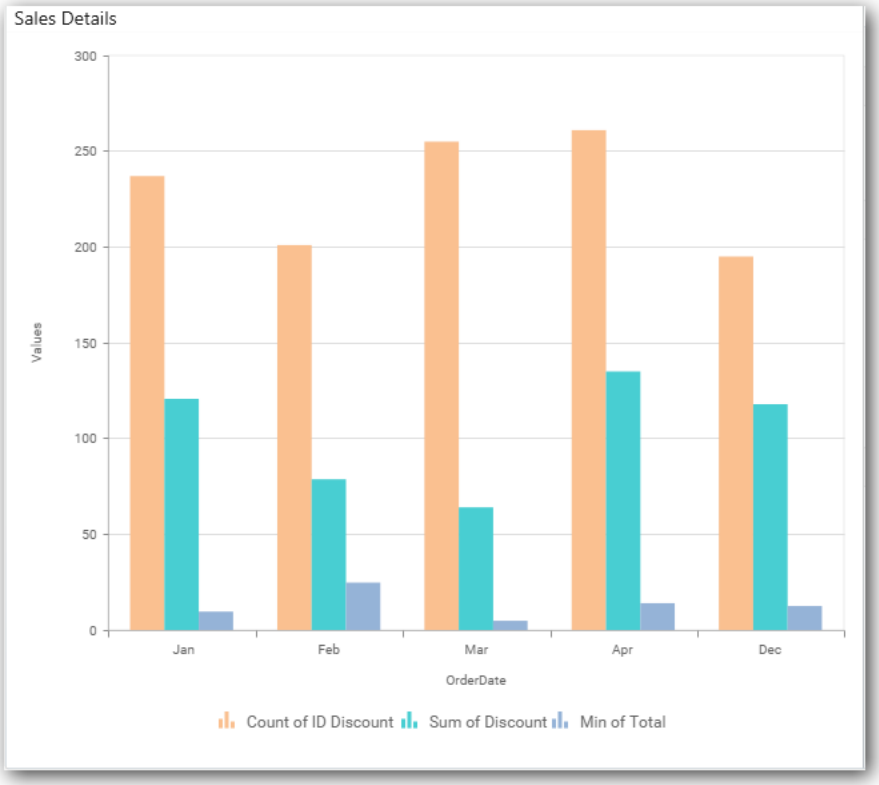
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



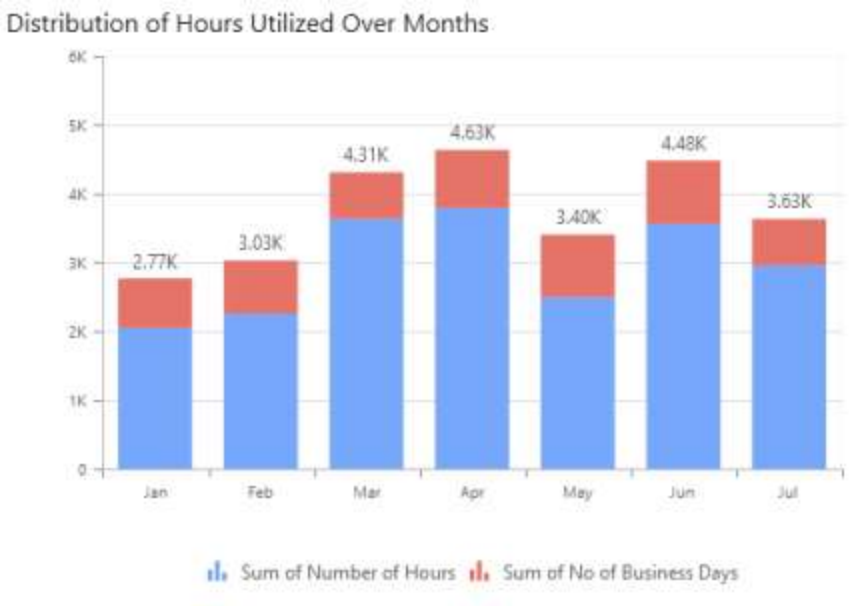
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Stacked Column Chart

Stacked Column Chart allows you to compare multiple measures through bars stacked one after the other vertically.

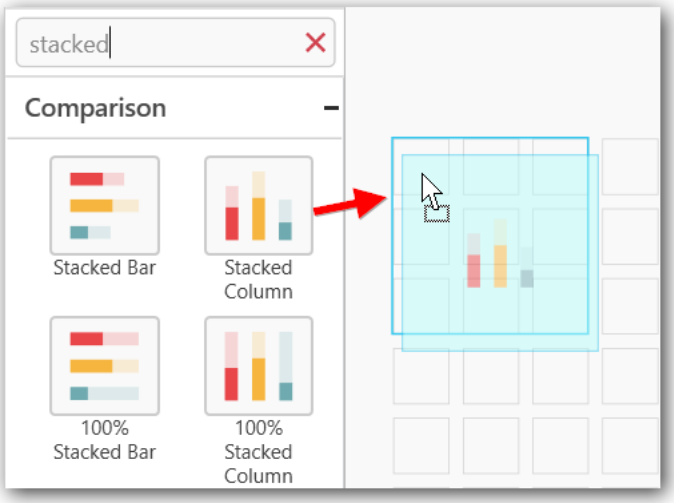


How to configure the flat table data to Stacked Column Chart?

Stacked Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

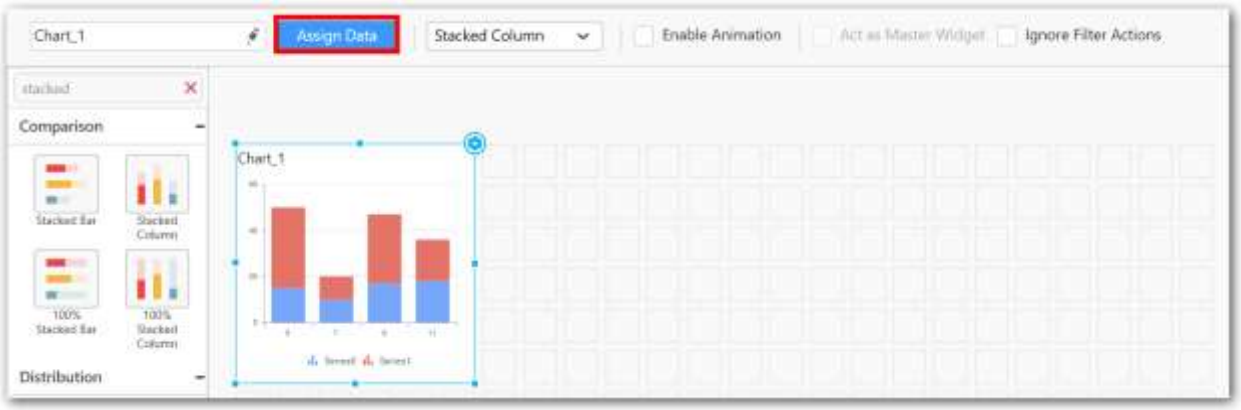
Follow the steps configure data to stacked column chart

Drag and drop the stacked column chart to canvas and resize it to your required size.

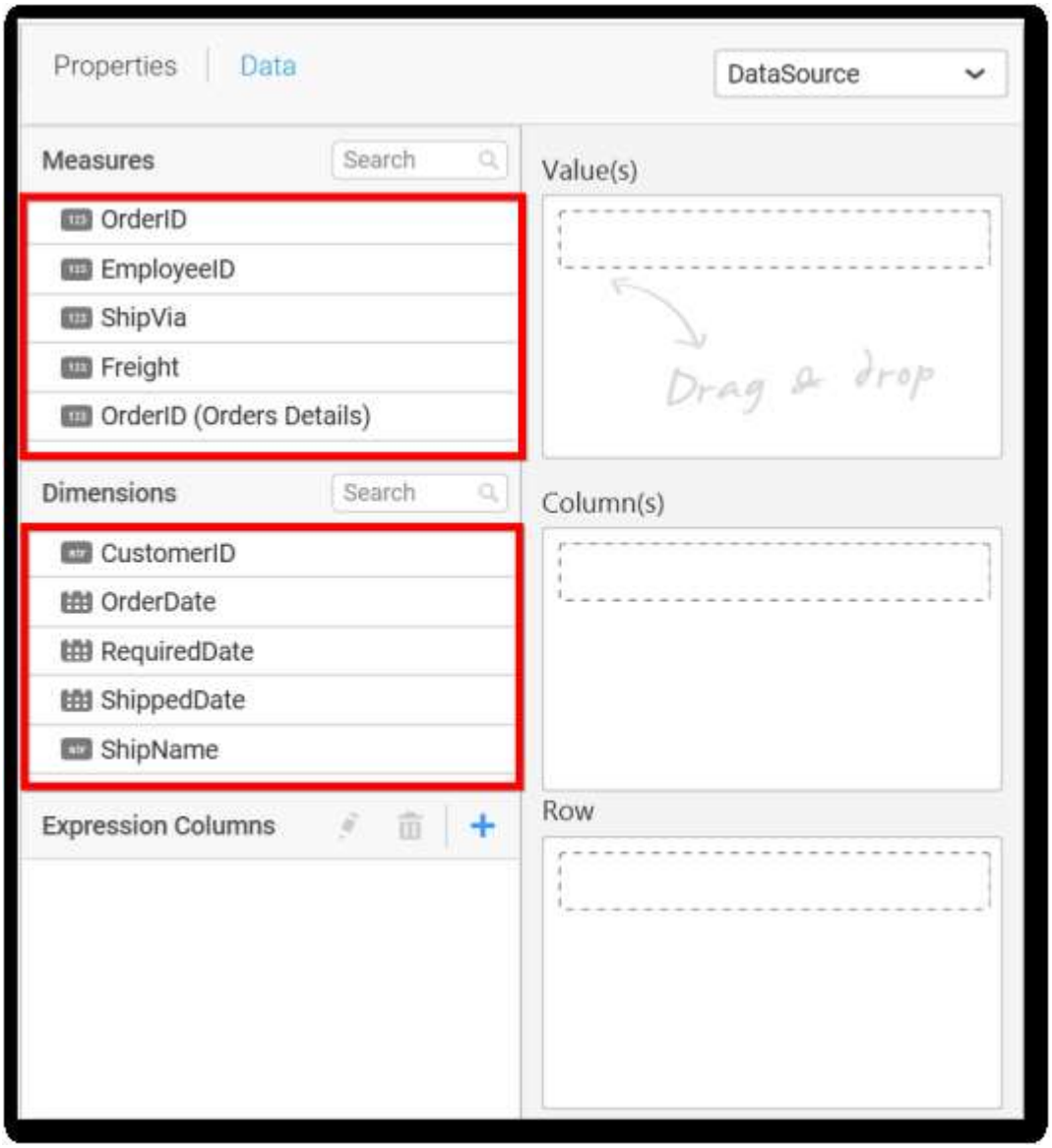


Connect to the data source.

Focus on the stacked column chart and click on Assign Data button

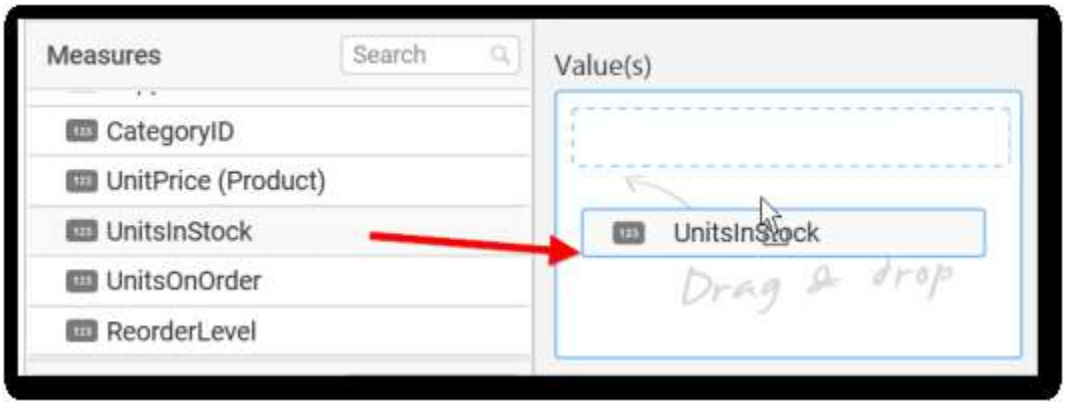


The data pane will be opened with available Measures and Columns from the connected data source.

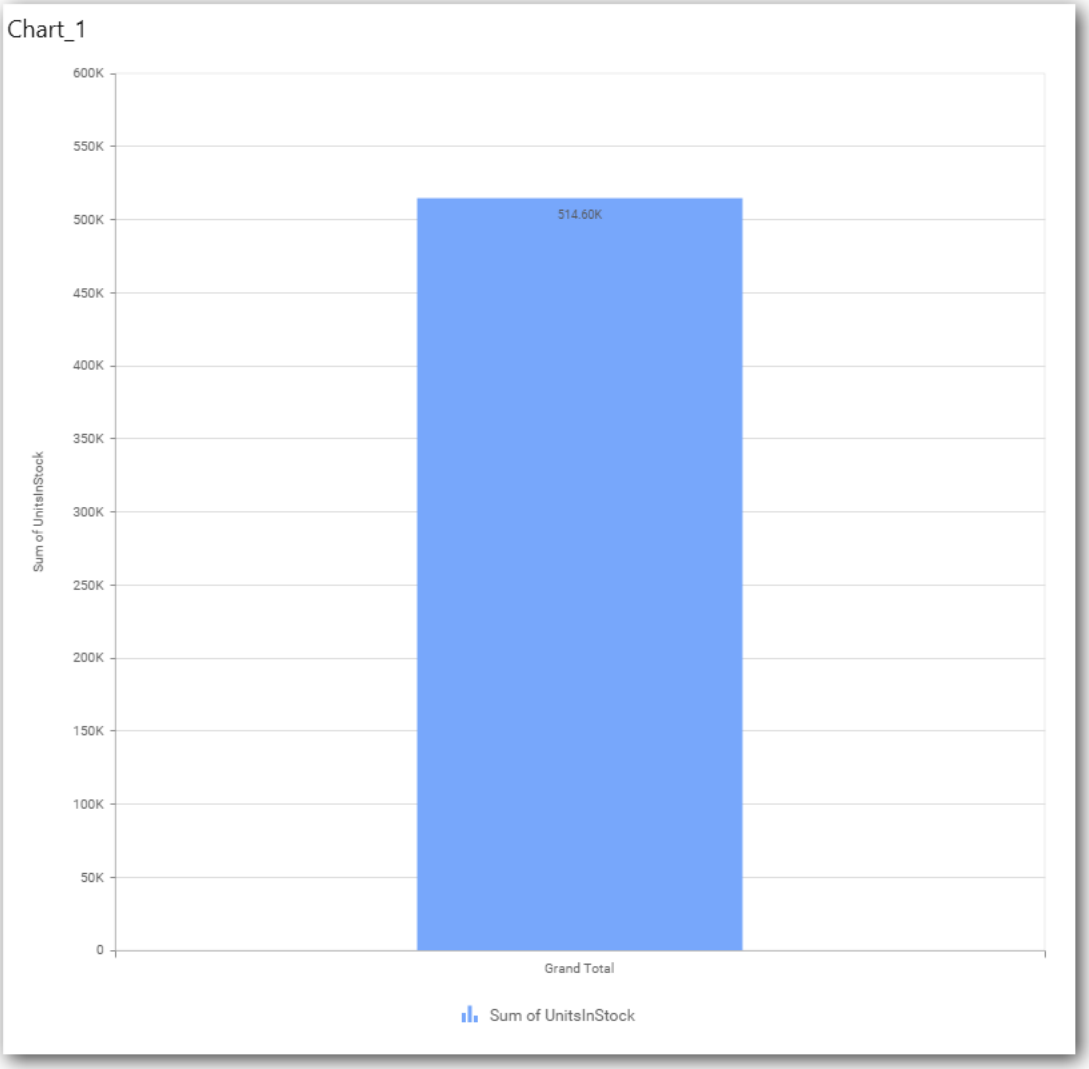


Assigning Value(s)

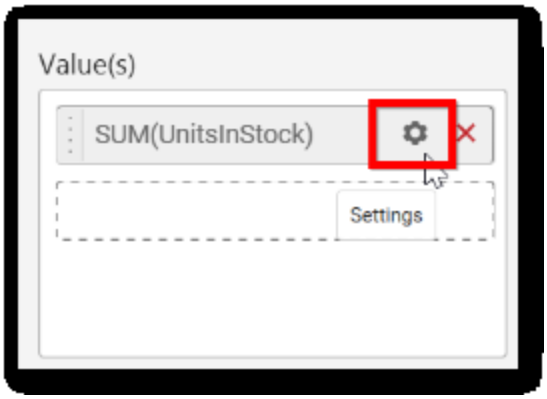
Drag and drop the Measure into Value.



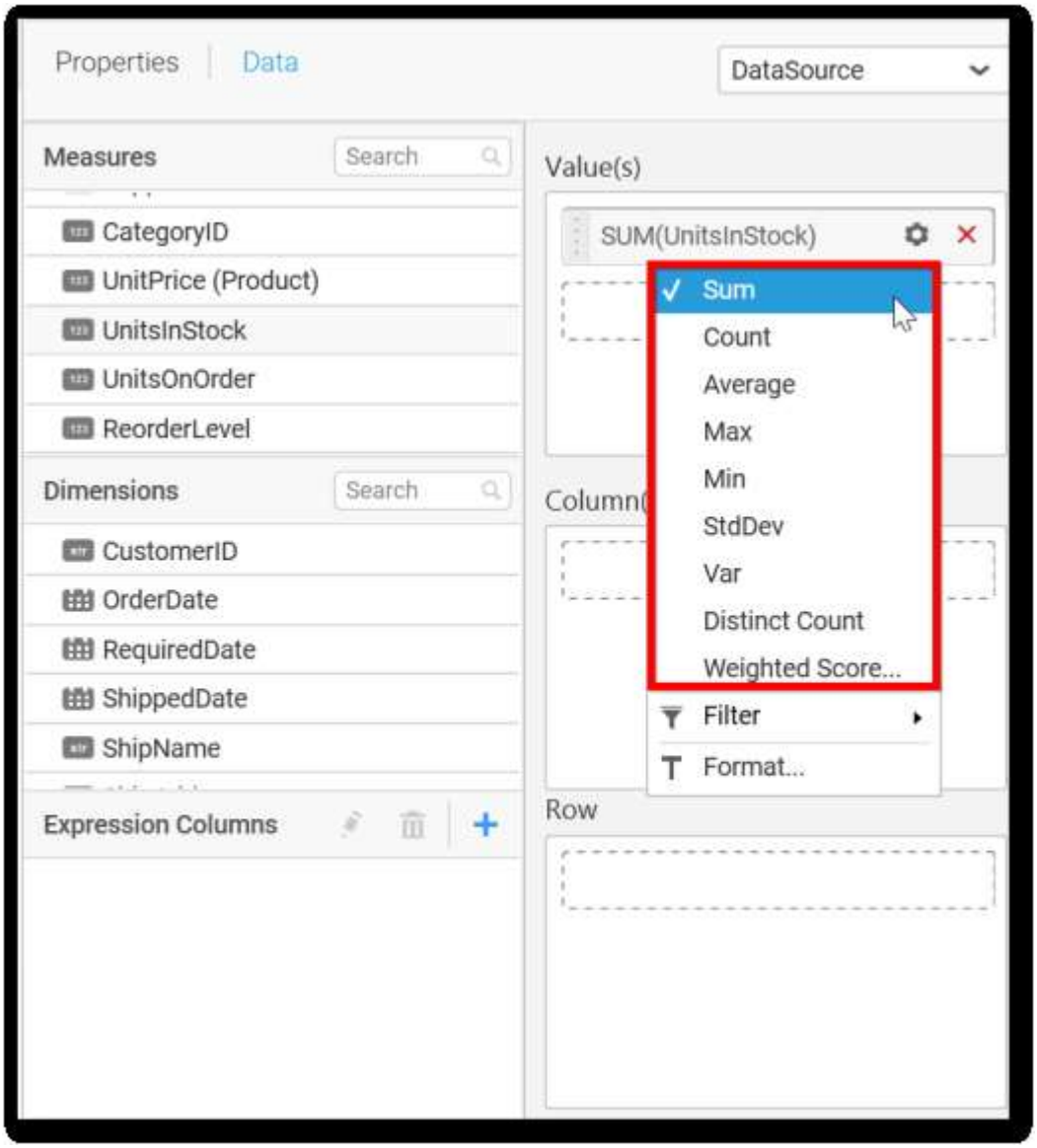
Now the chart will be rendered like this.



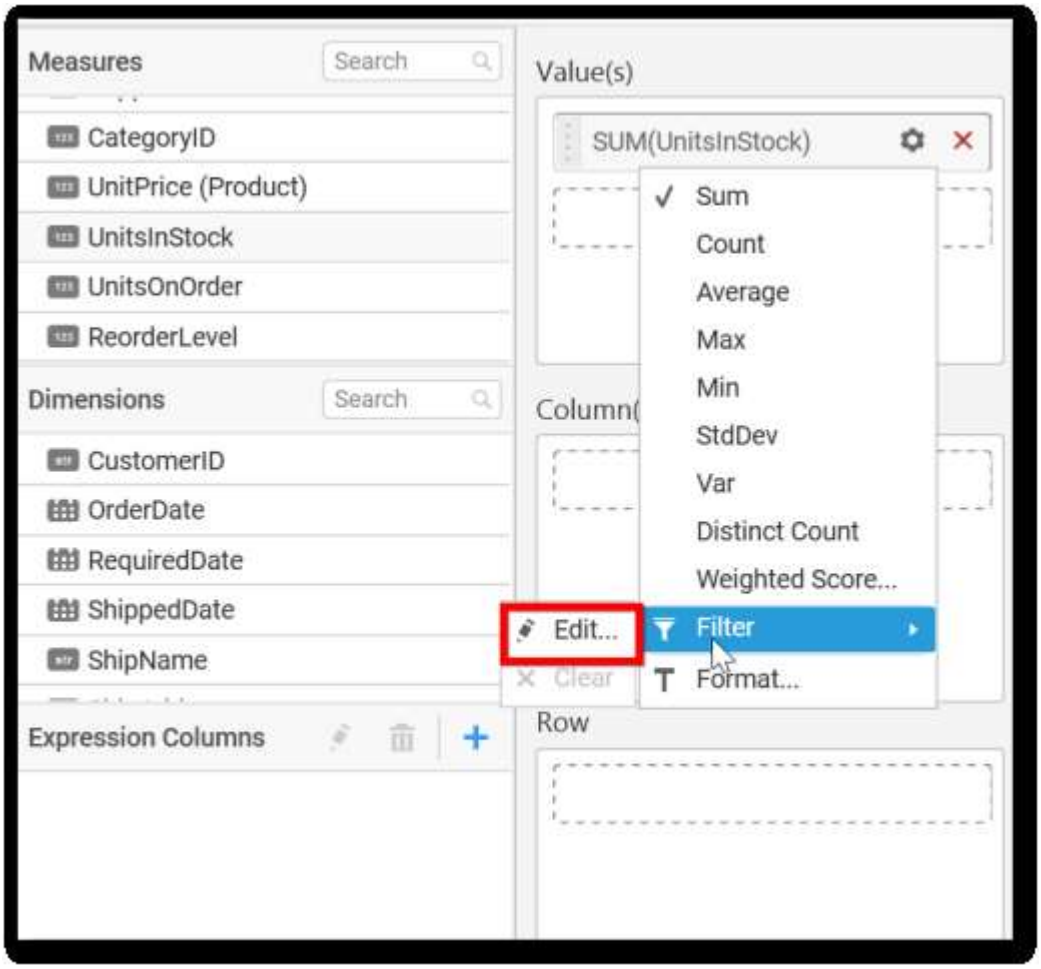
You can change the summary type of the value by clicking on Settings option.



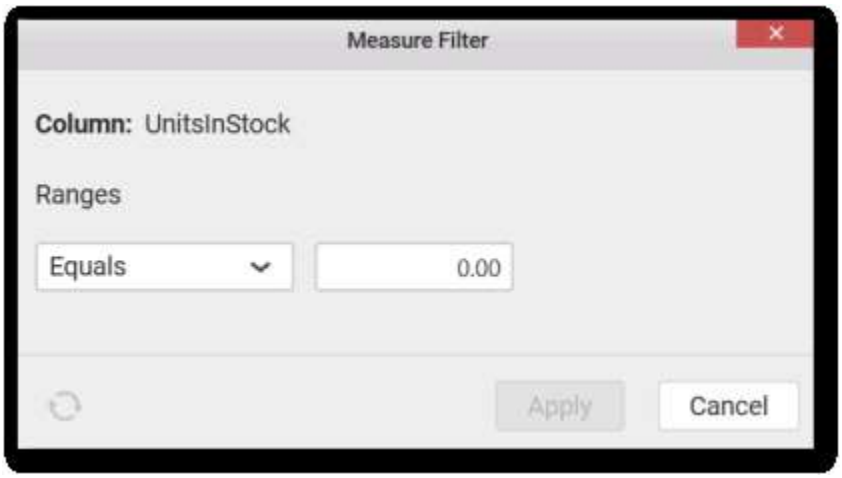
Select the required summary type from list.

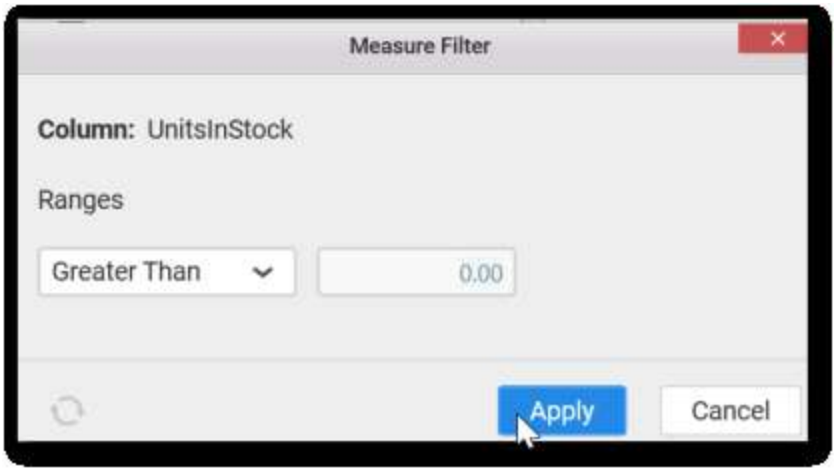


You can select what data to be displayed by choosing filter option.

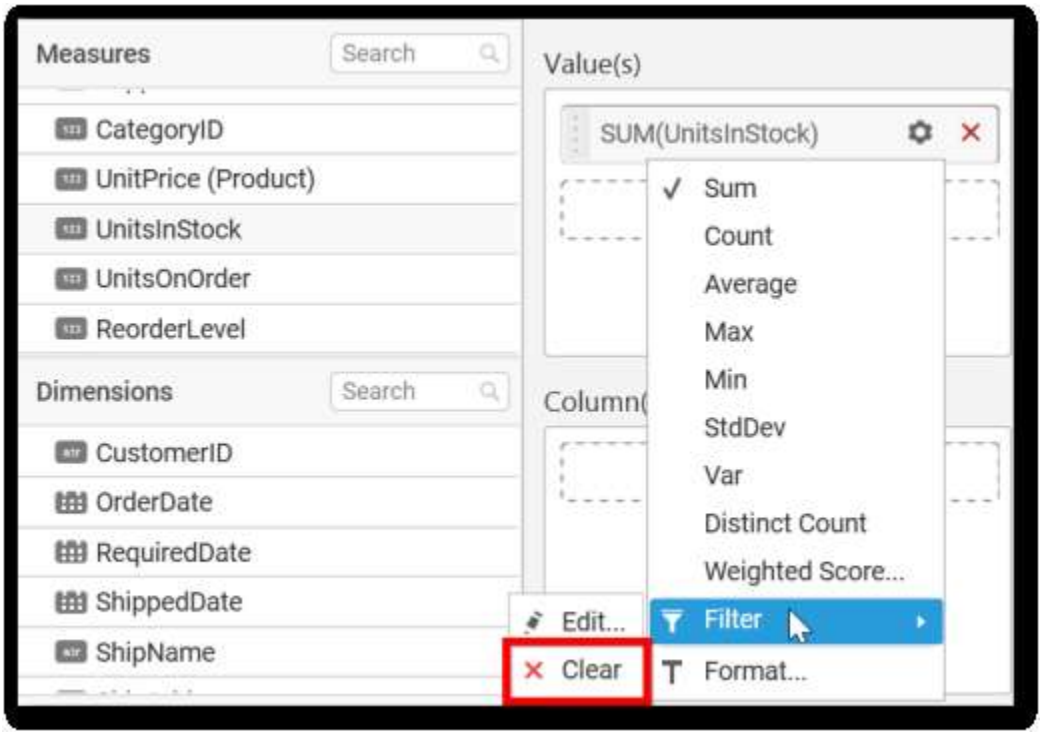


The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.

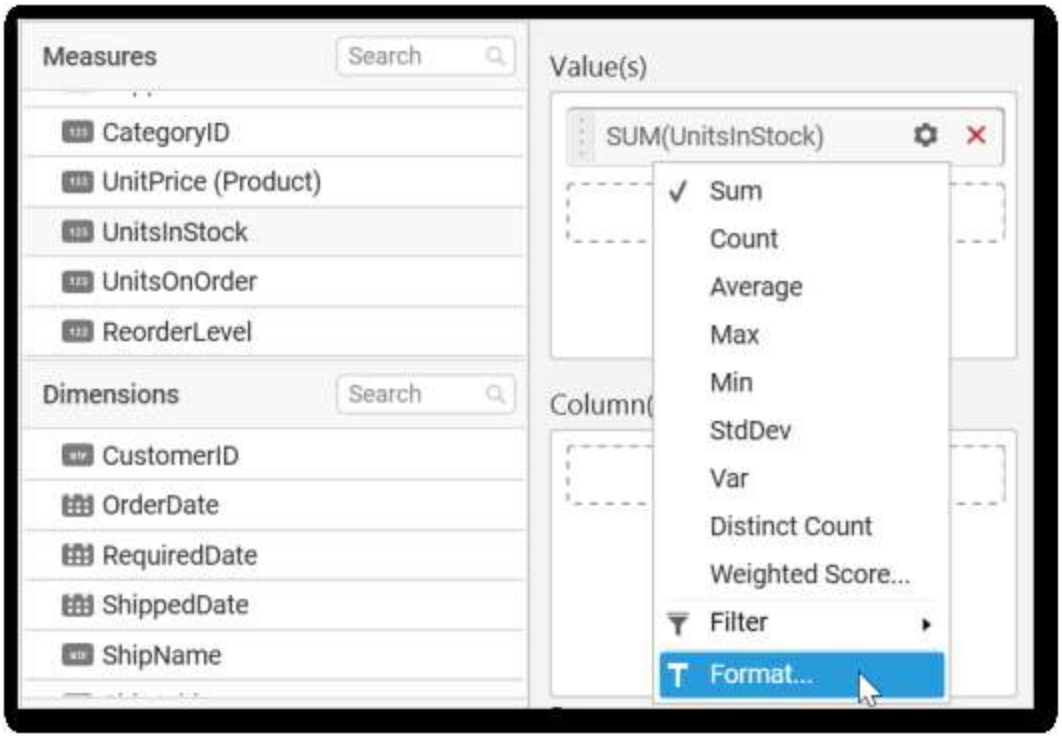




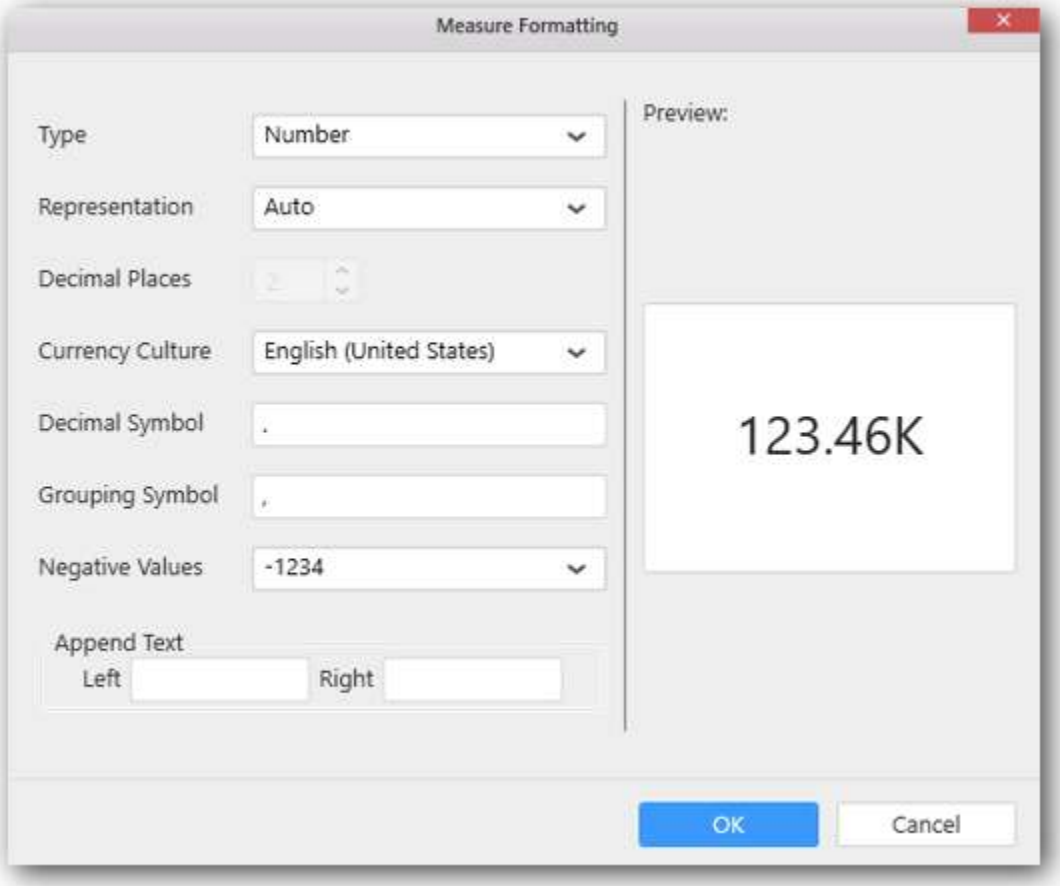
You can clear the filter.



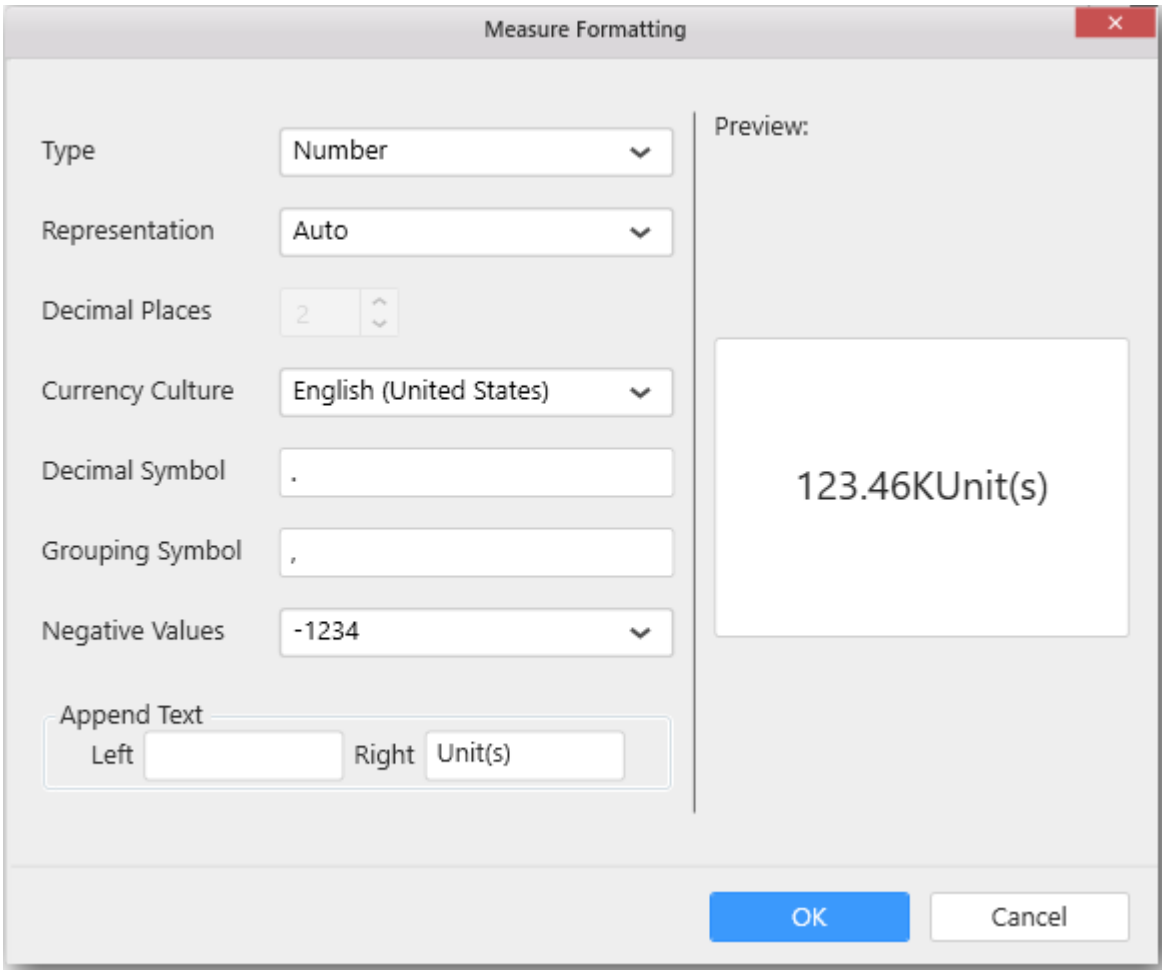
You can Format the value.



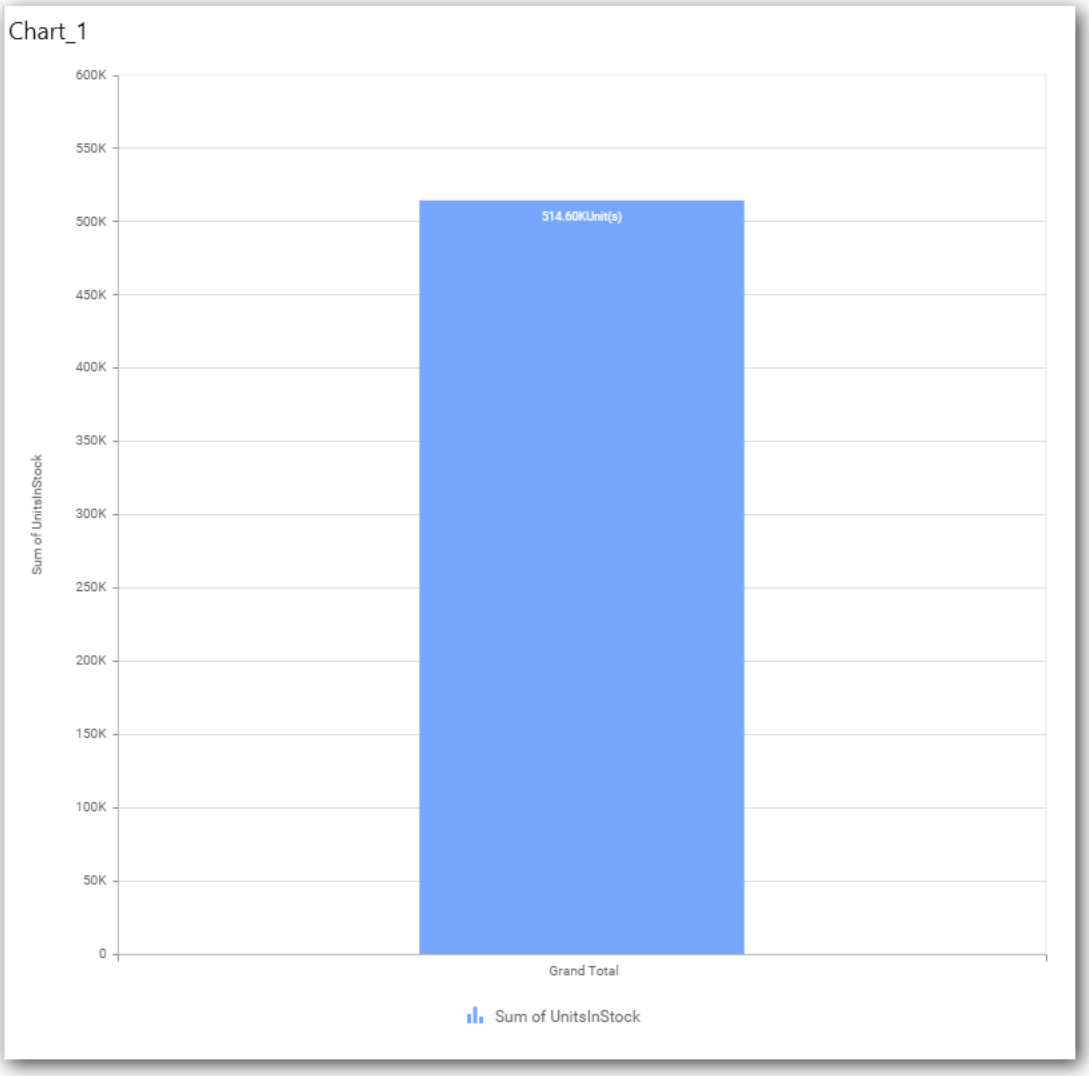
The format options will be shown.



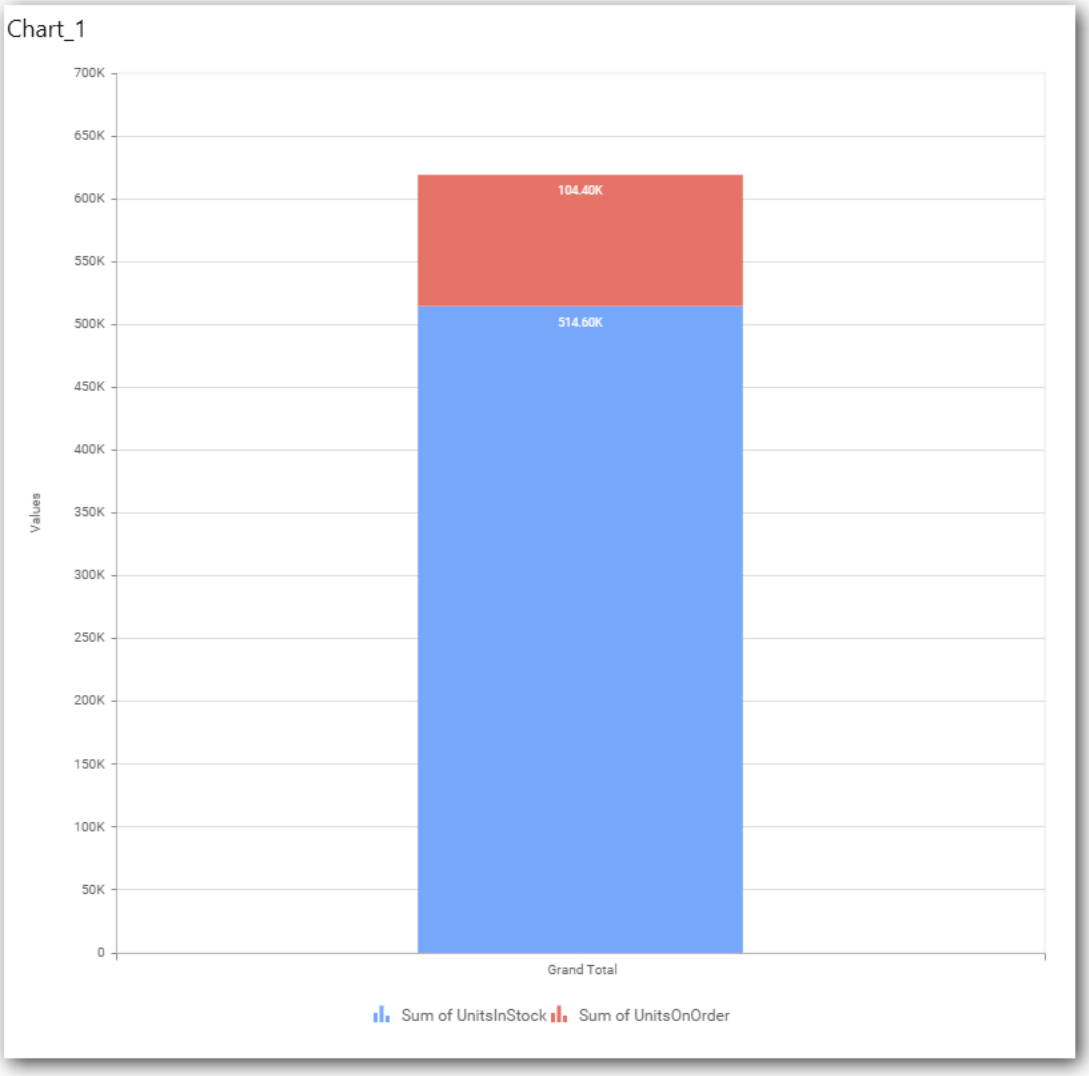
Choose the options you need and click OK.



Now the Chart will be rendered like this.



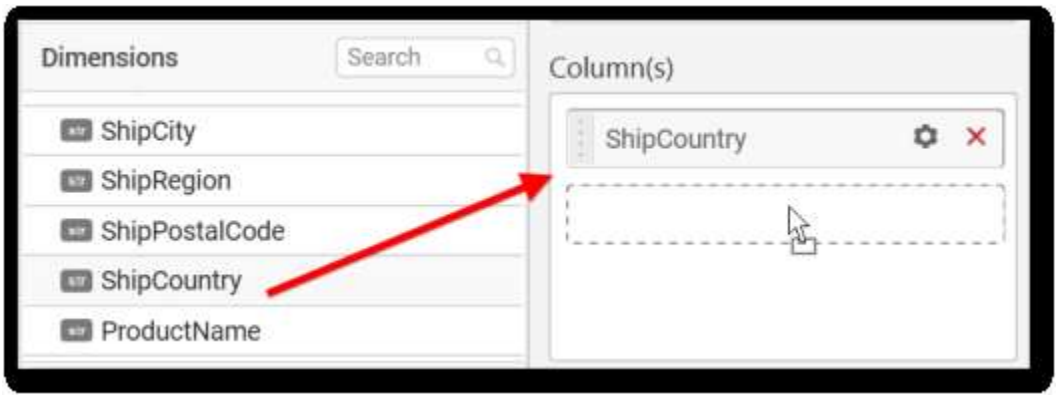
You can add more number values by drag drop the Measure into Value field.

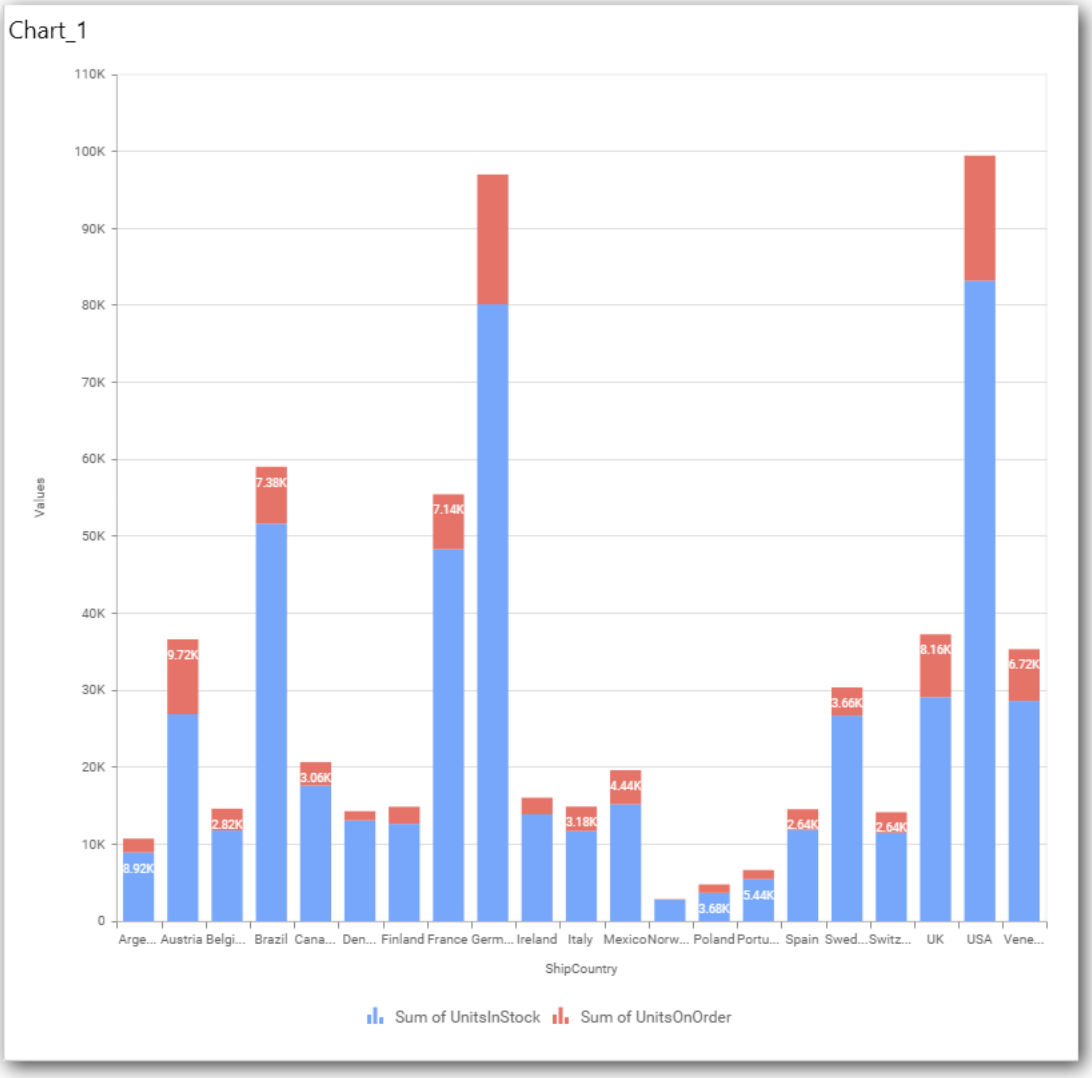


You can also add Dimensions and Columns to Value(s).

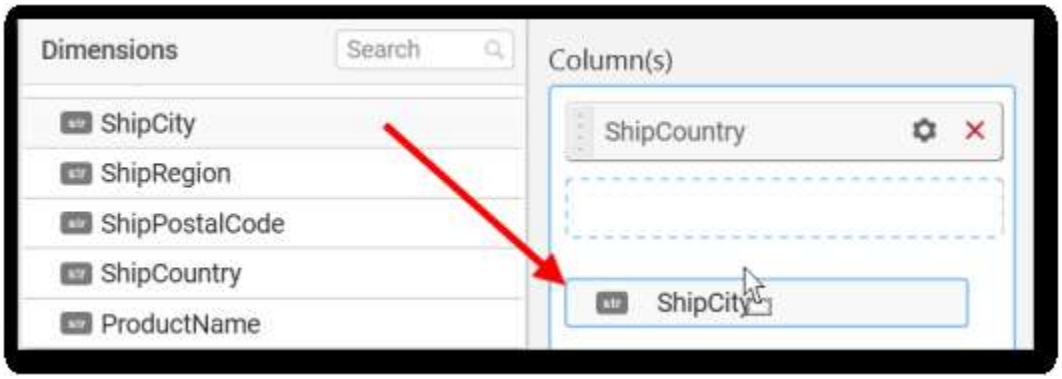
Assigning Column(s)

You can add the Dimension into Column field by drag and drop.

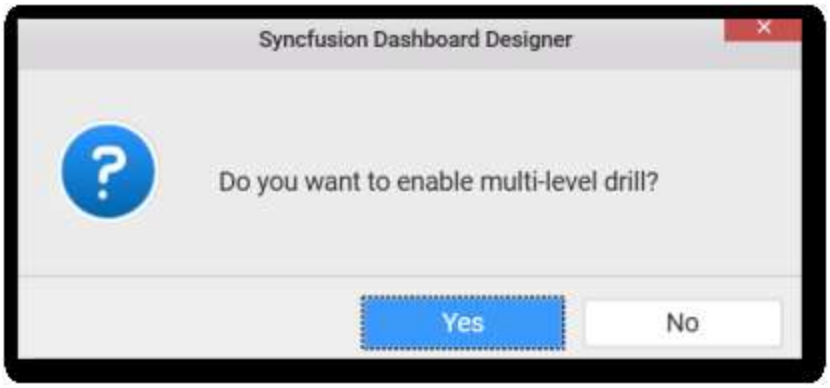




You have option to add more than one Column to Value.

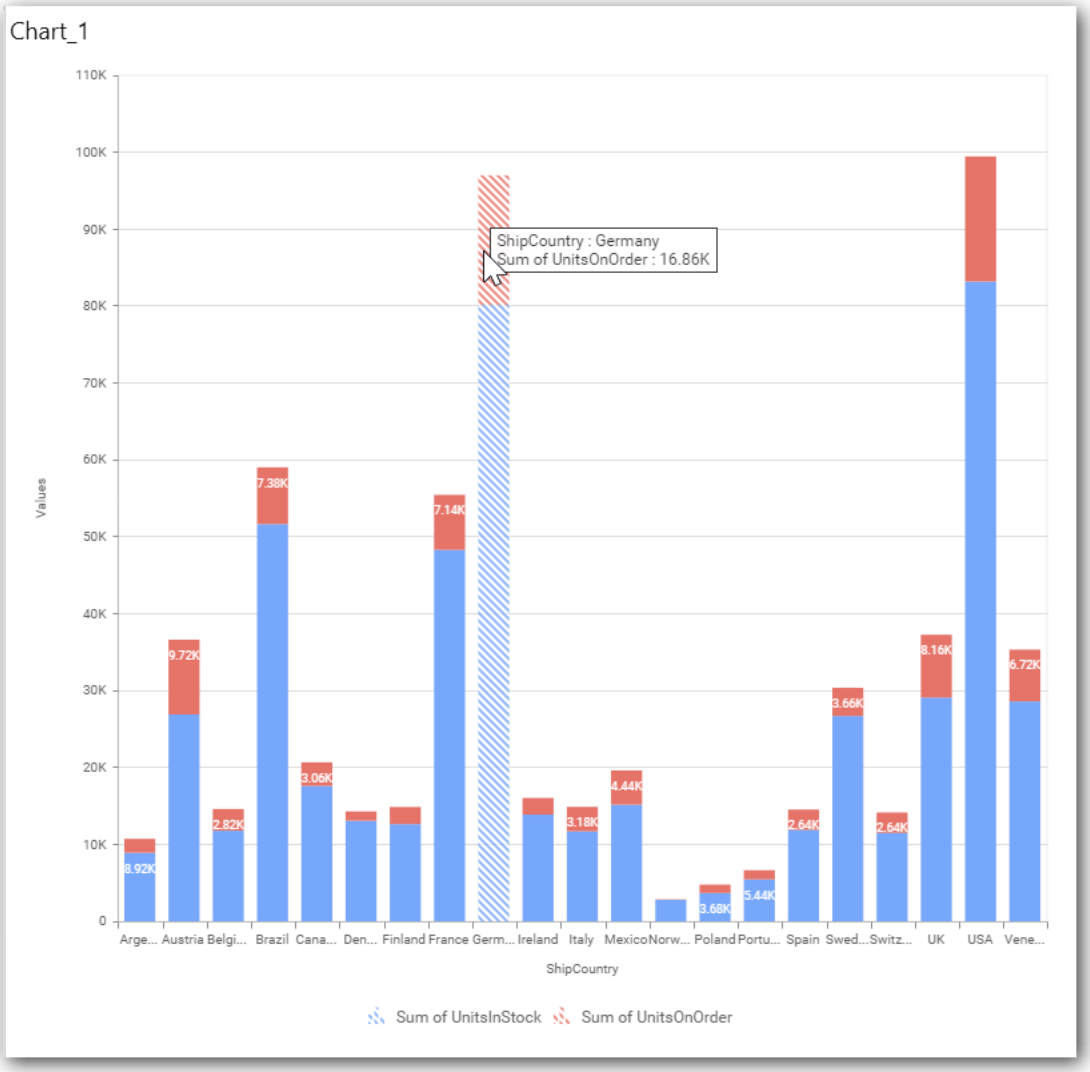


The following alert message will be shown.

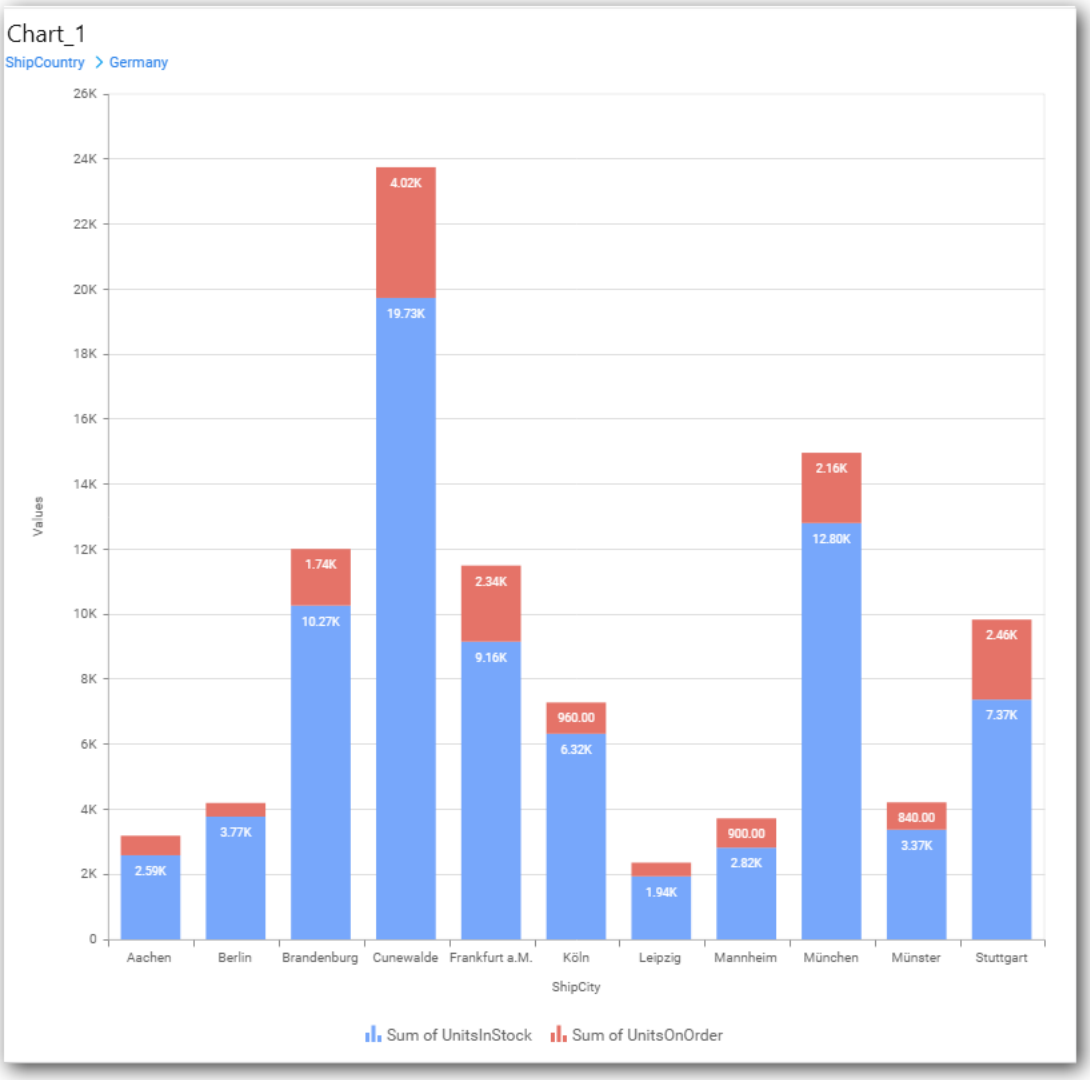


If you choose **Yes** Drill down option will be enabled.

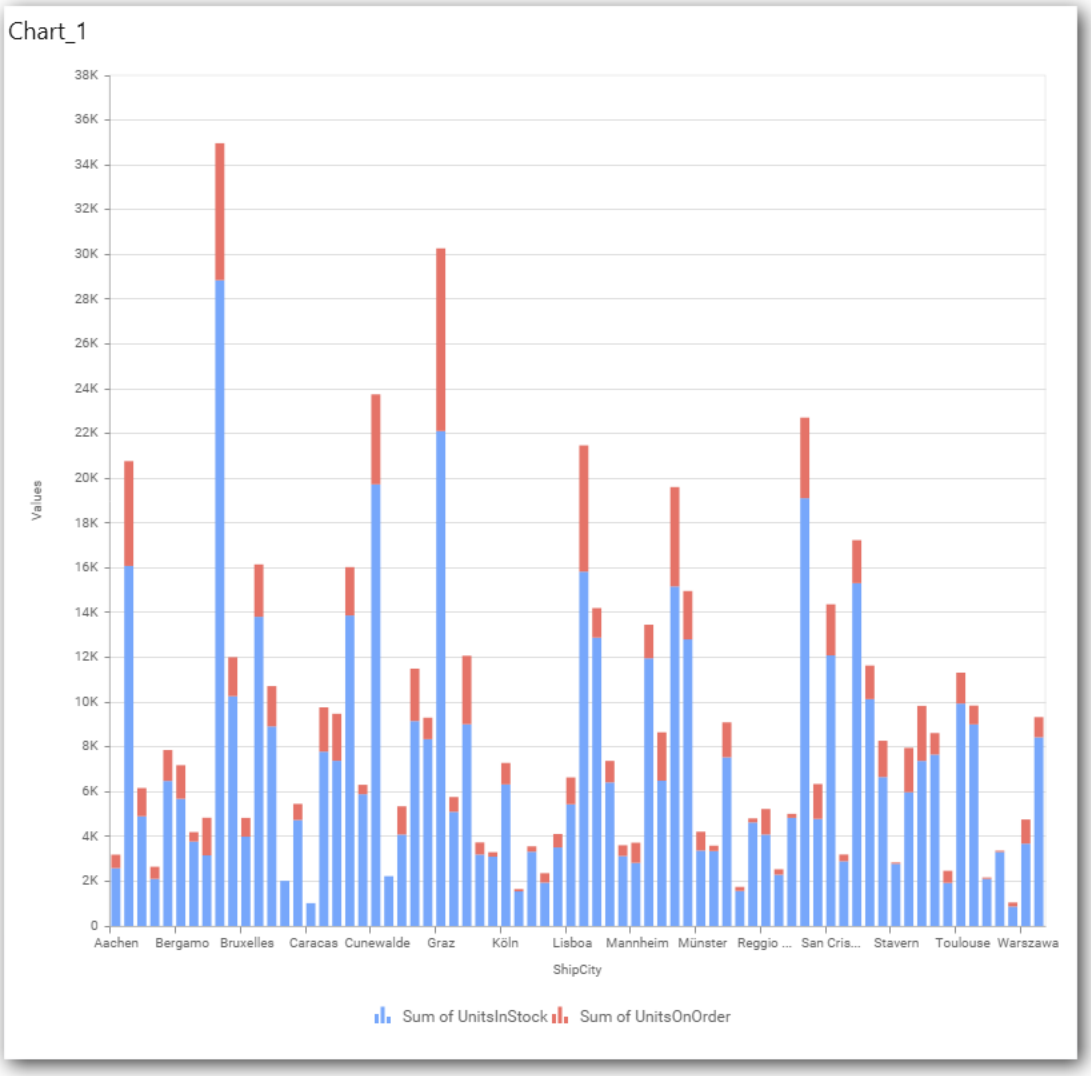
You can drill down the chart by clicking on the chart.



The drilled view of the chart is follows

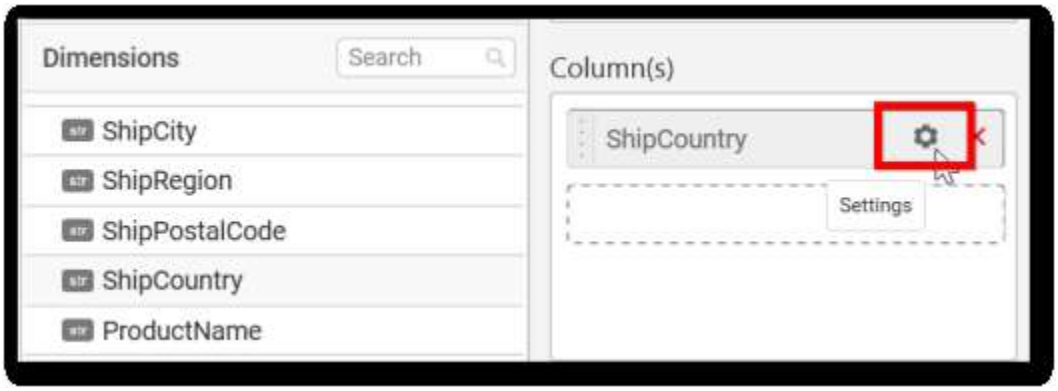


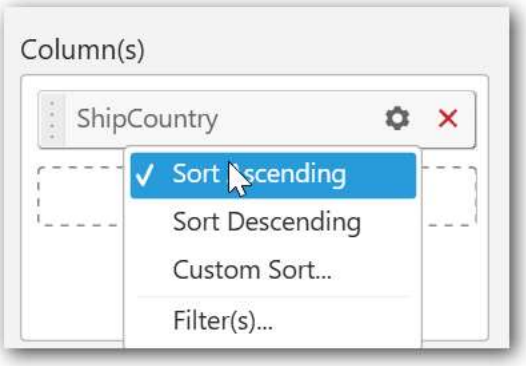
If you click No the new Dimension value will replace old value.



You can also add Measures and Expression columns into Column(s) field.

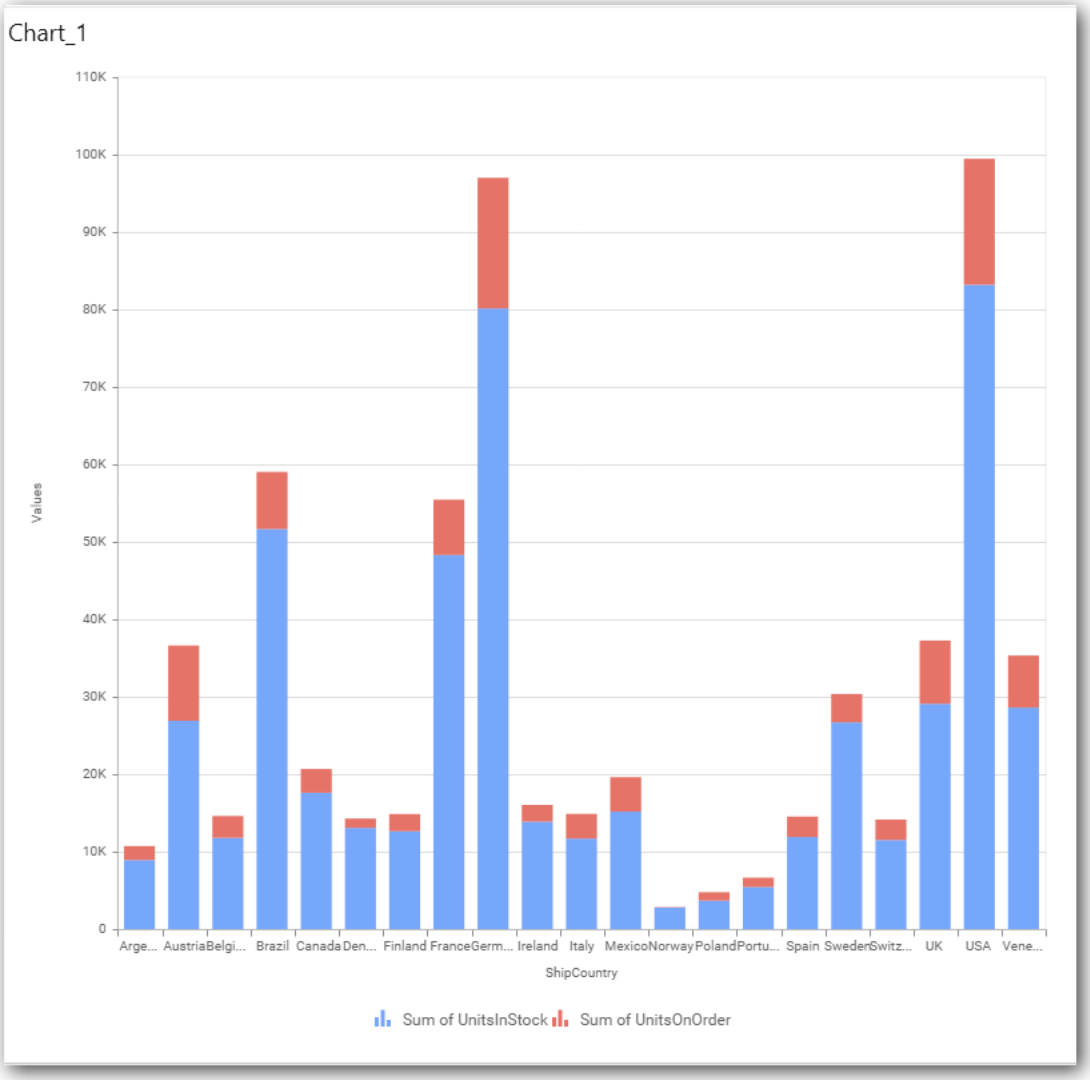
You have options to change the settings.



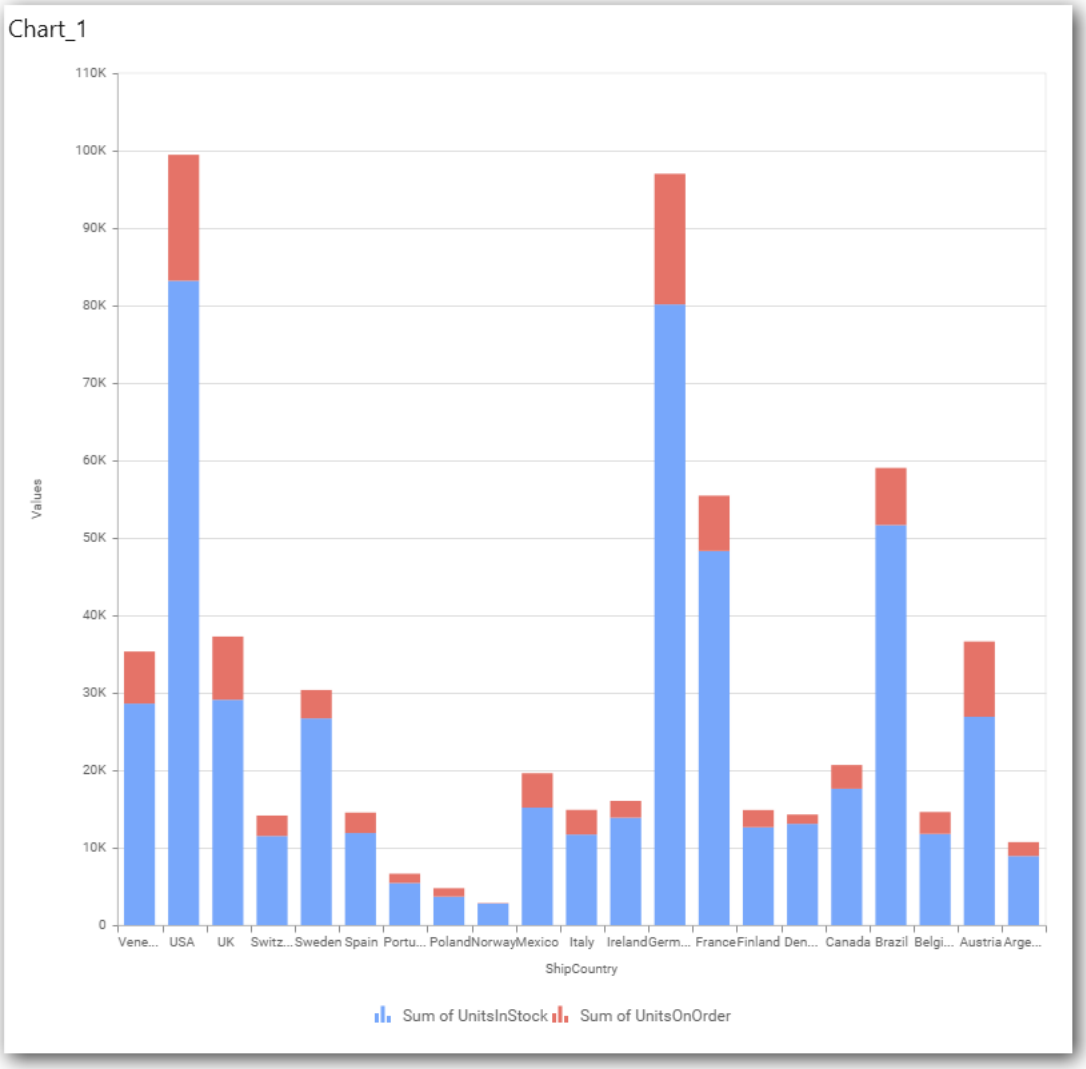


You can sort the chart either in Ascending or Descending series.

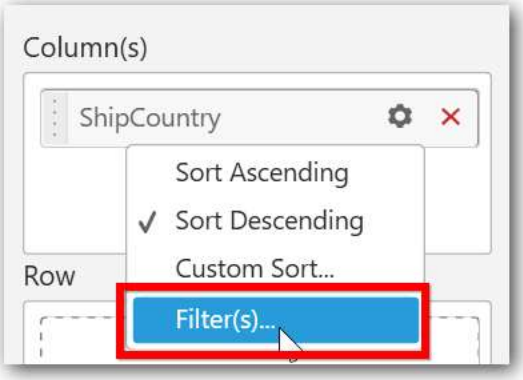
Ascending Order:

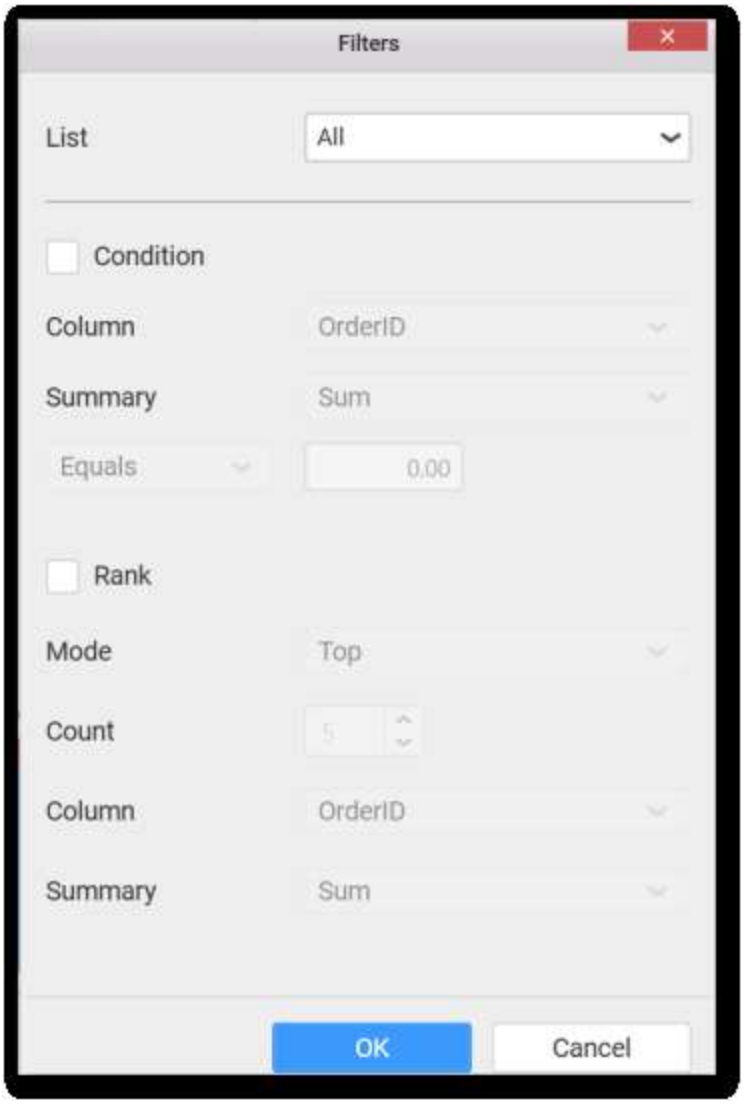


Descending order:

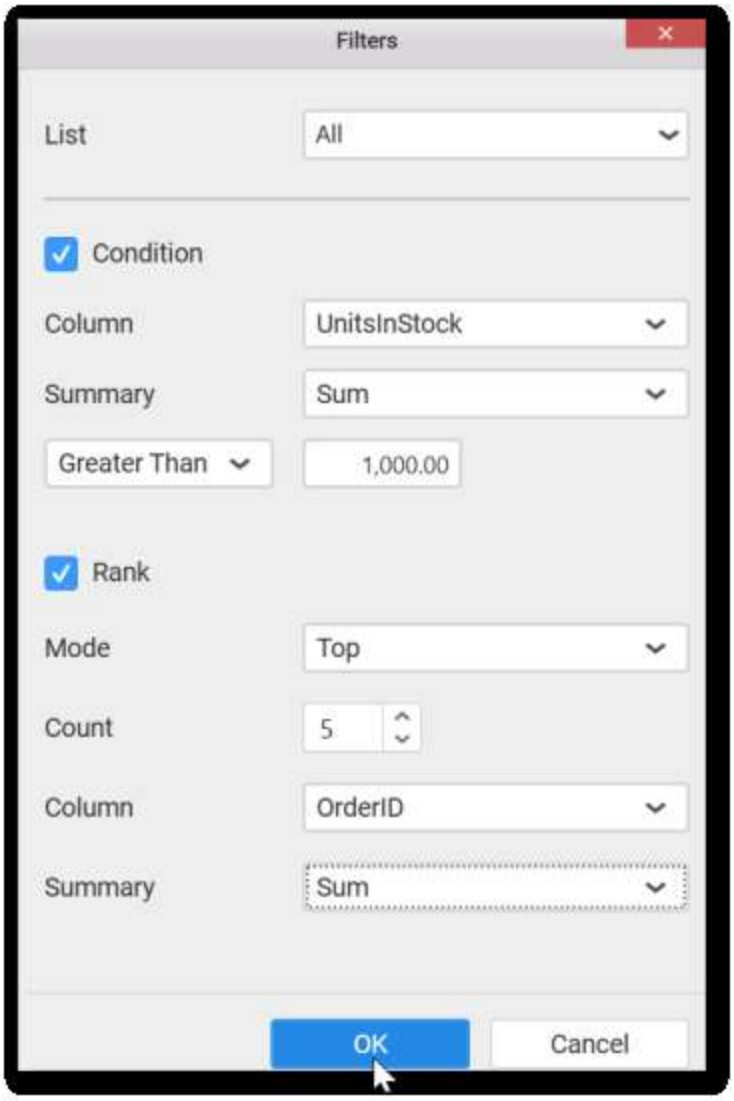


You can apply a filter

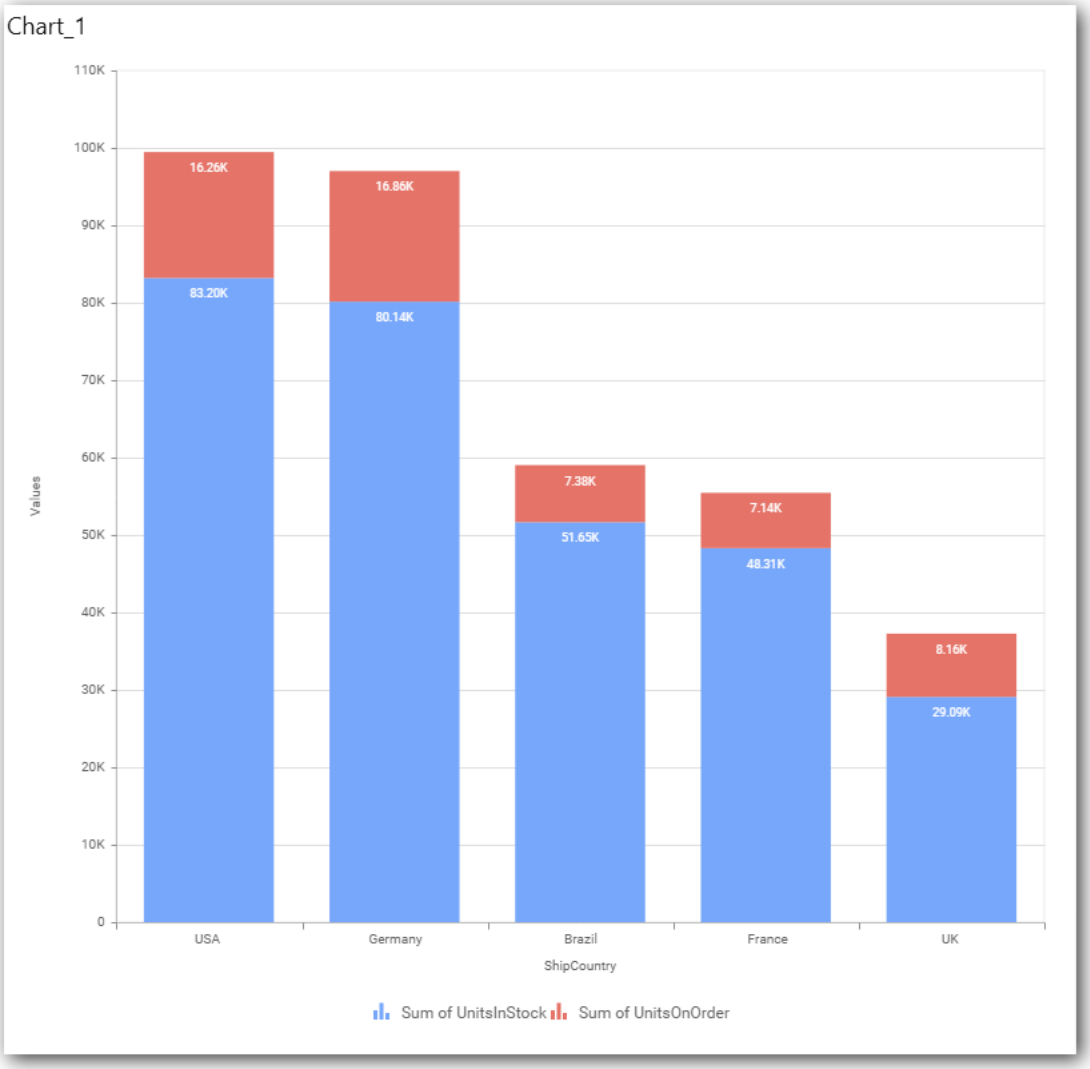




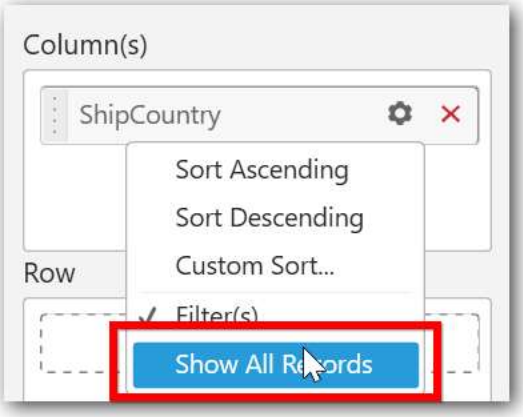
Select the Conditions and Rank you need.



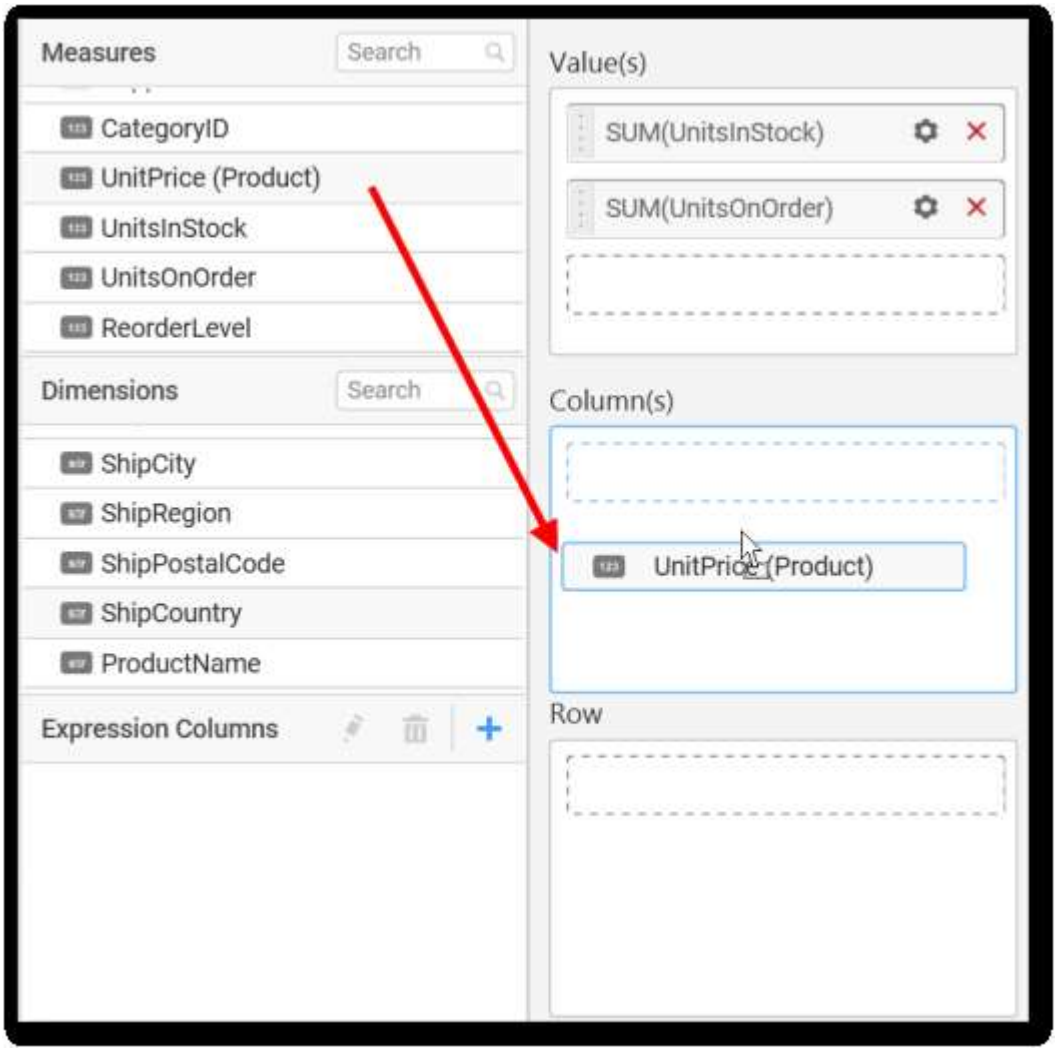
Now the chart will be rendered like this.

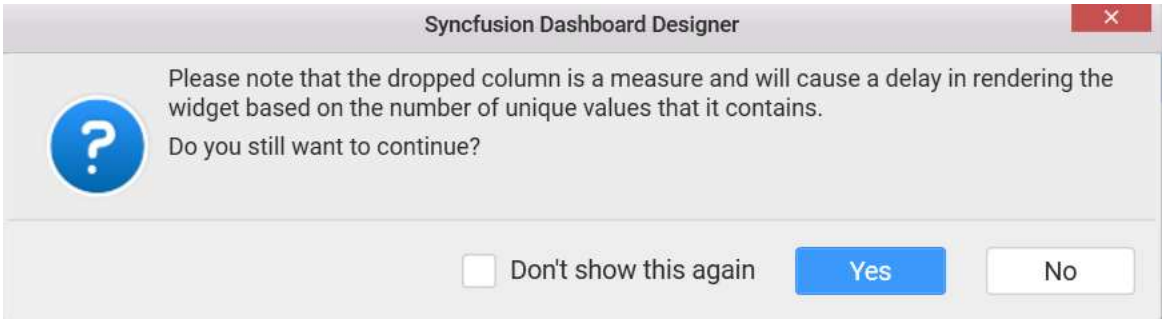


To show all records again click on Show All Records.



On adding Measures into Column(s) will show the following alert.

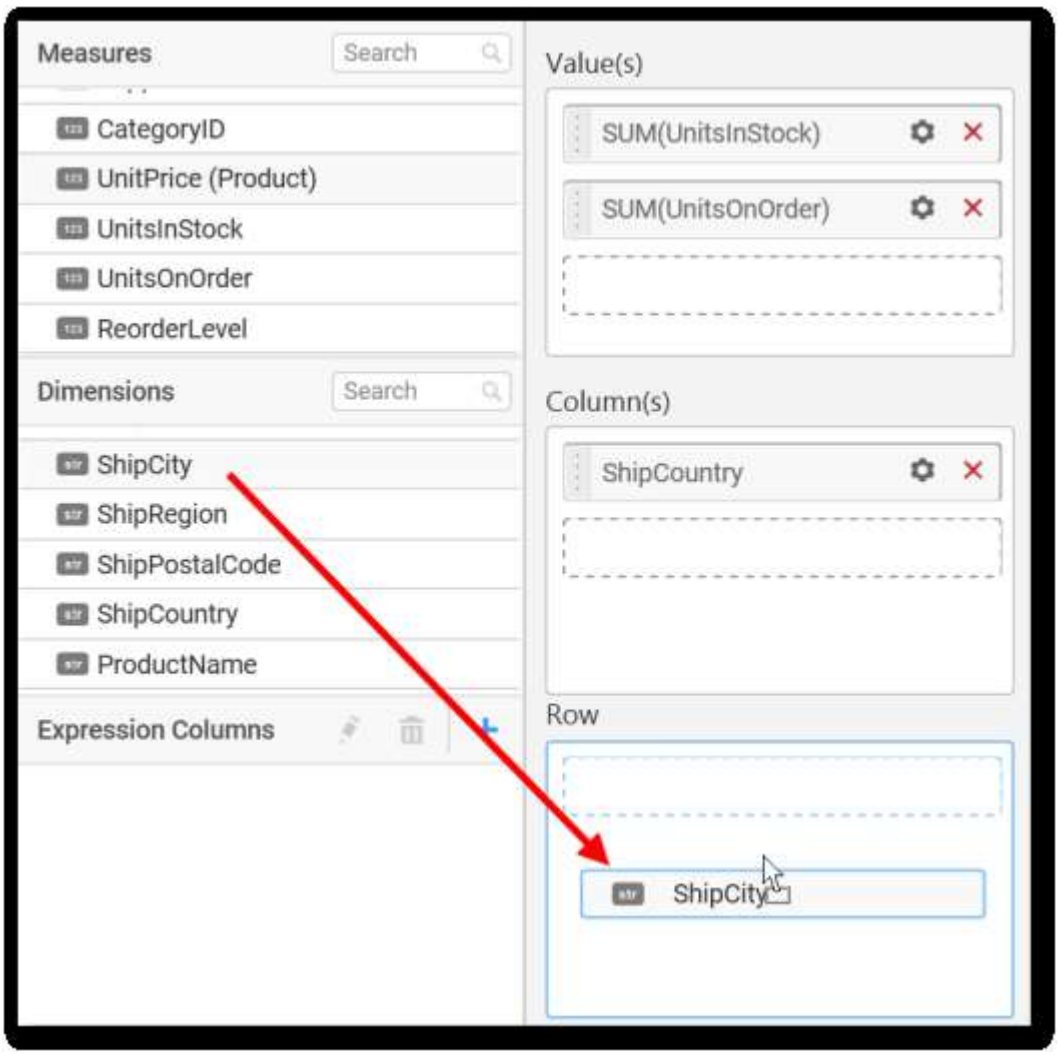




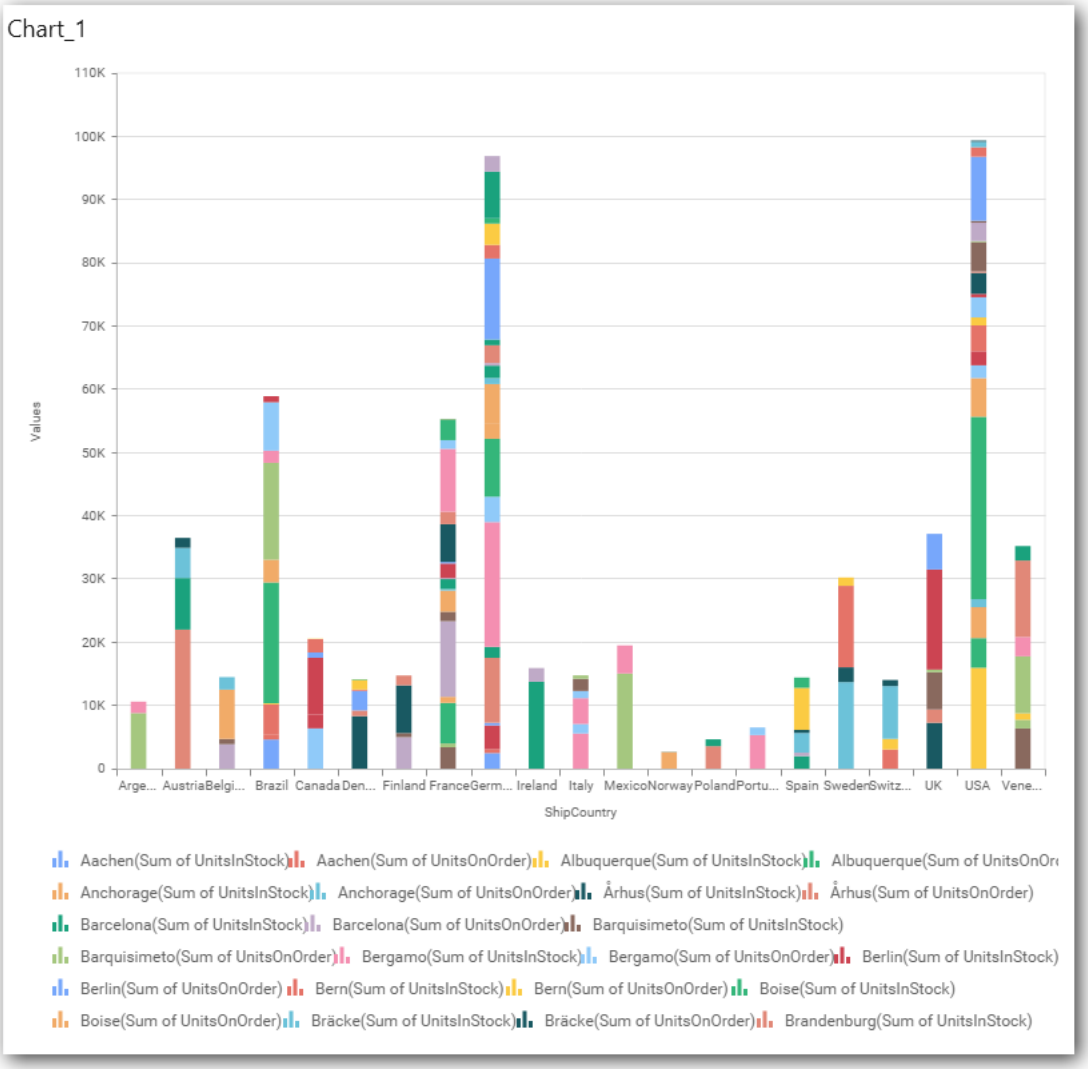
To continue select Yes, otherwise select No.

Assigning Row

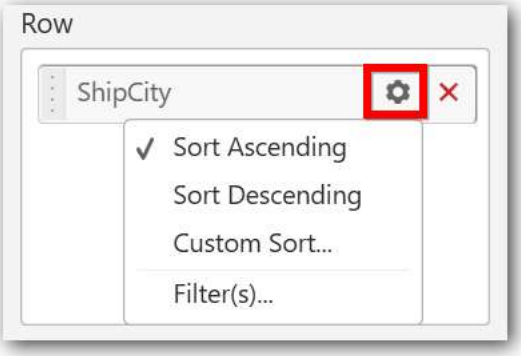
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s)



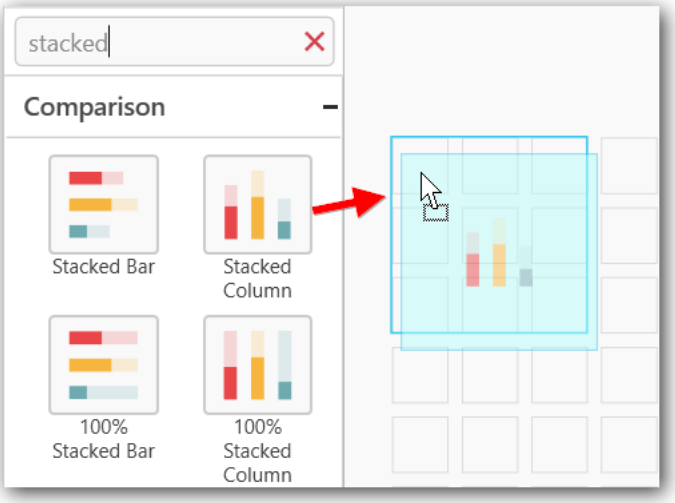
How to configure SSAS to Stacked Column Chart?

Stacked Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you

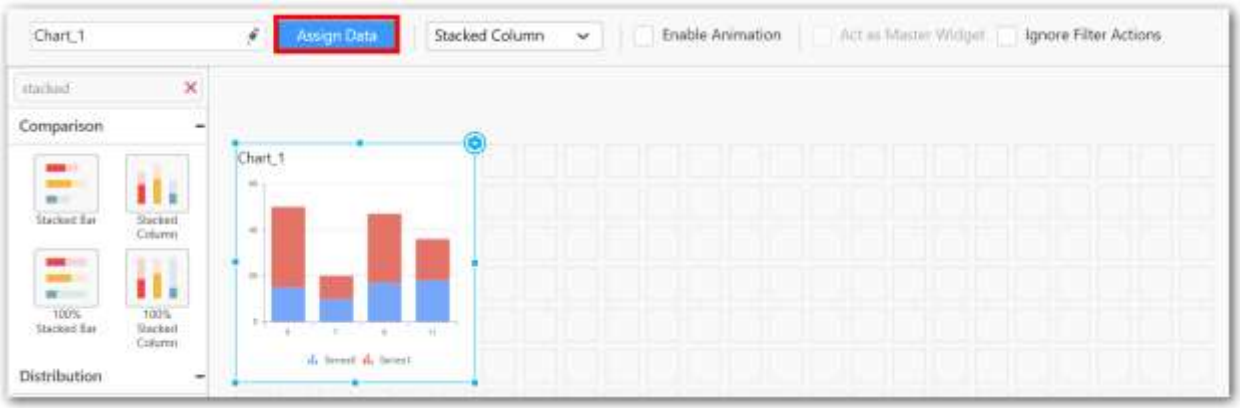
would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuration of SSAS data to Stacked Column chart

Drag and drop the Stacked Column chart widget into canvas and resize into your required size.

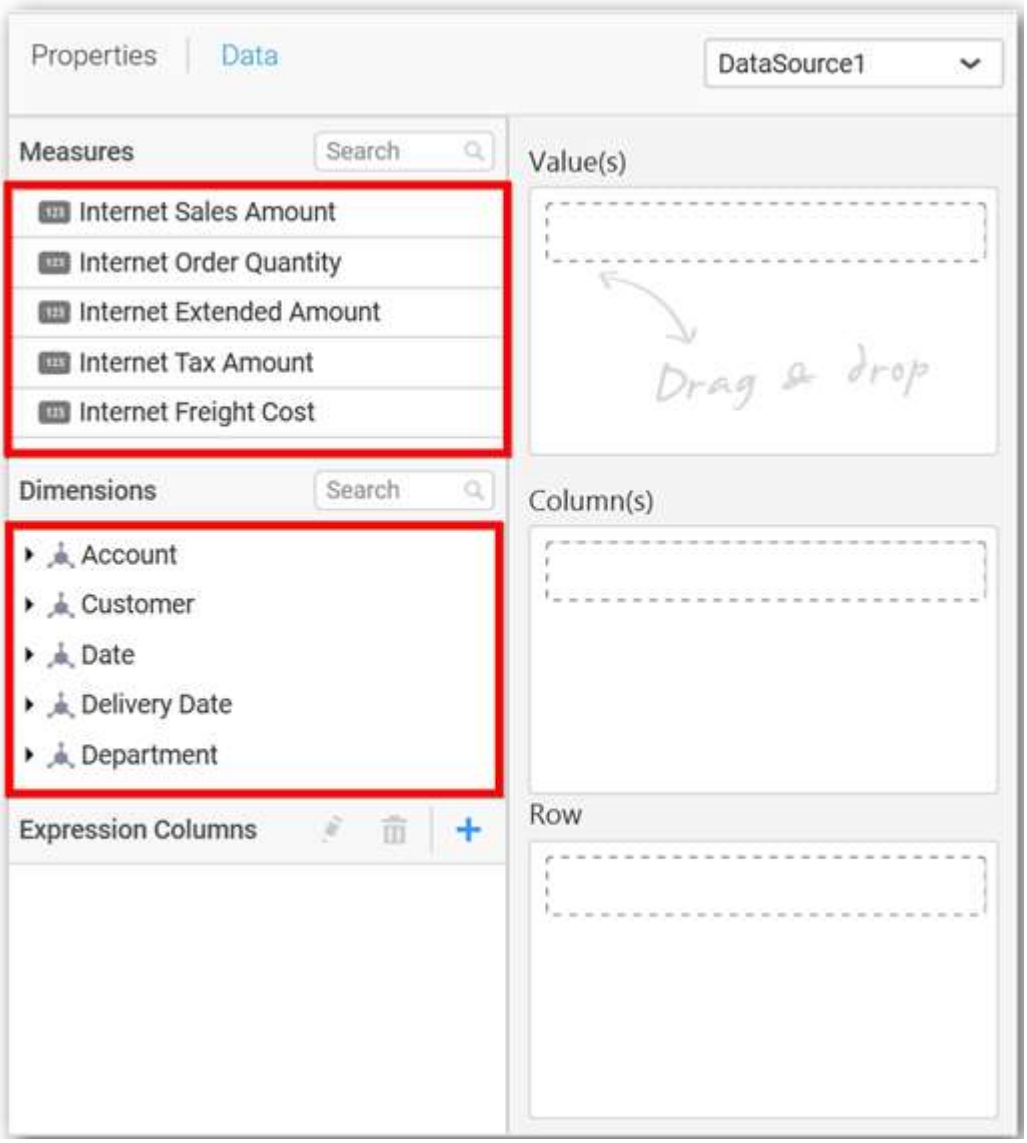


Select the dropped widget using mouse.



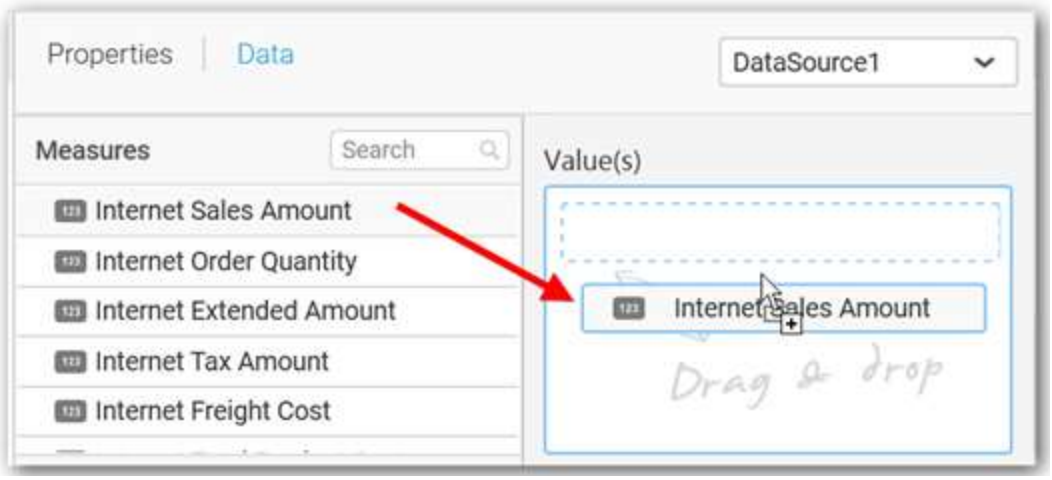
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions.

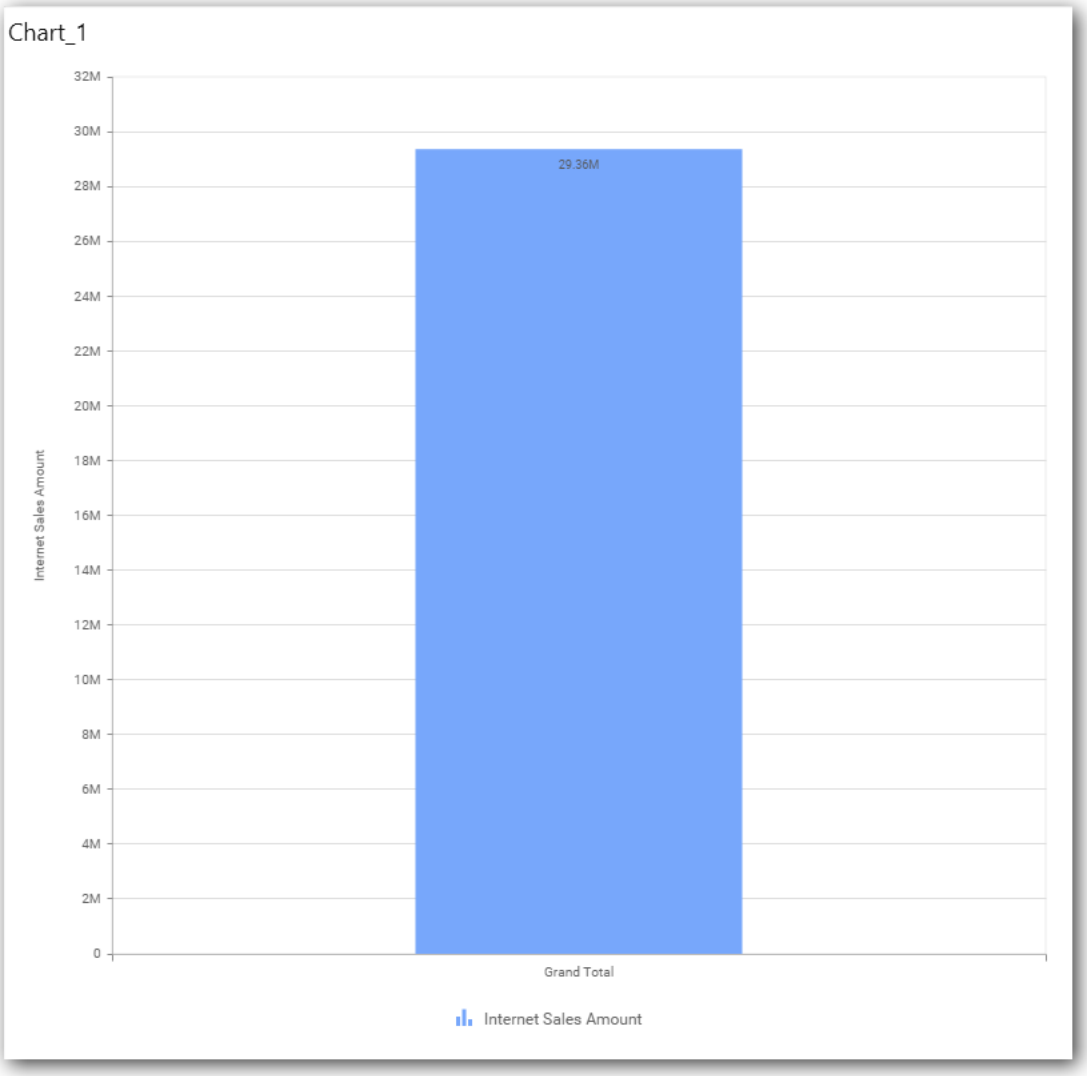


Assigning Value(s)

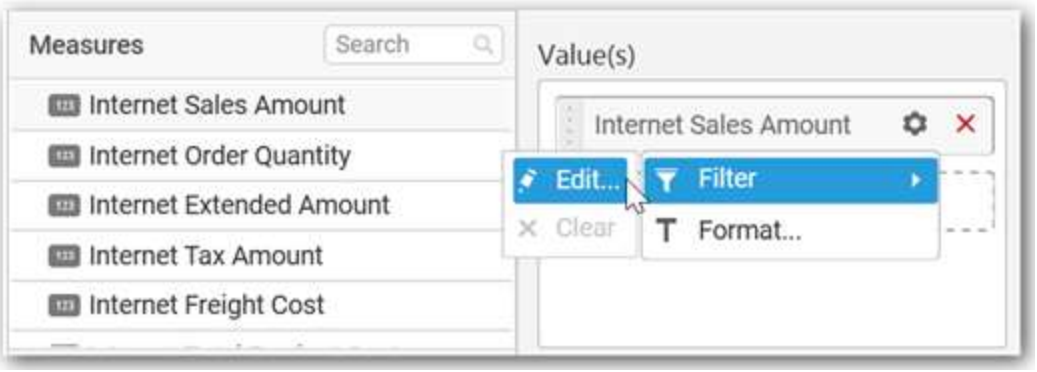
Drag and drop a column under Measures category into Value(s) section.



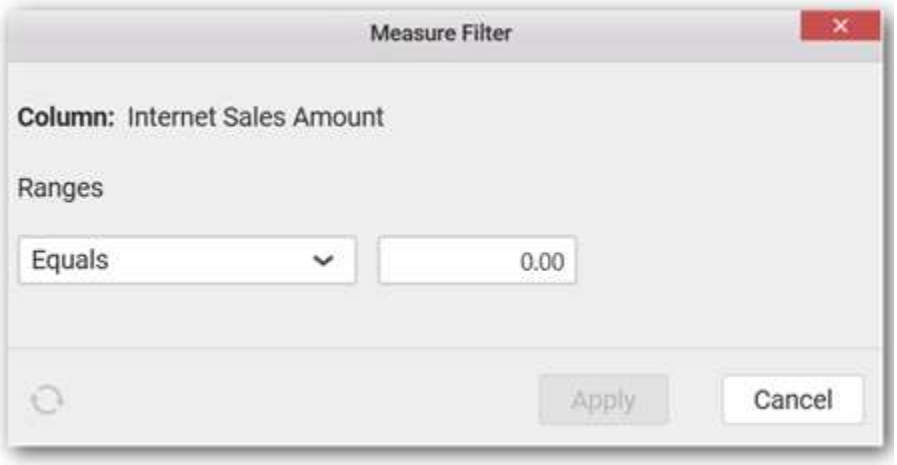
Now the chart will be rendered like this.



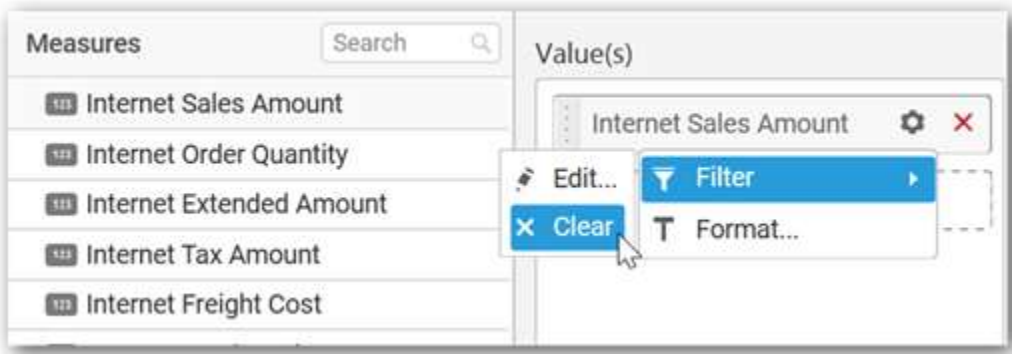
Define the filter criteria to match through choosing Edit option in Filter menu item.



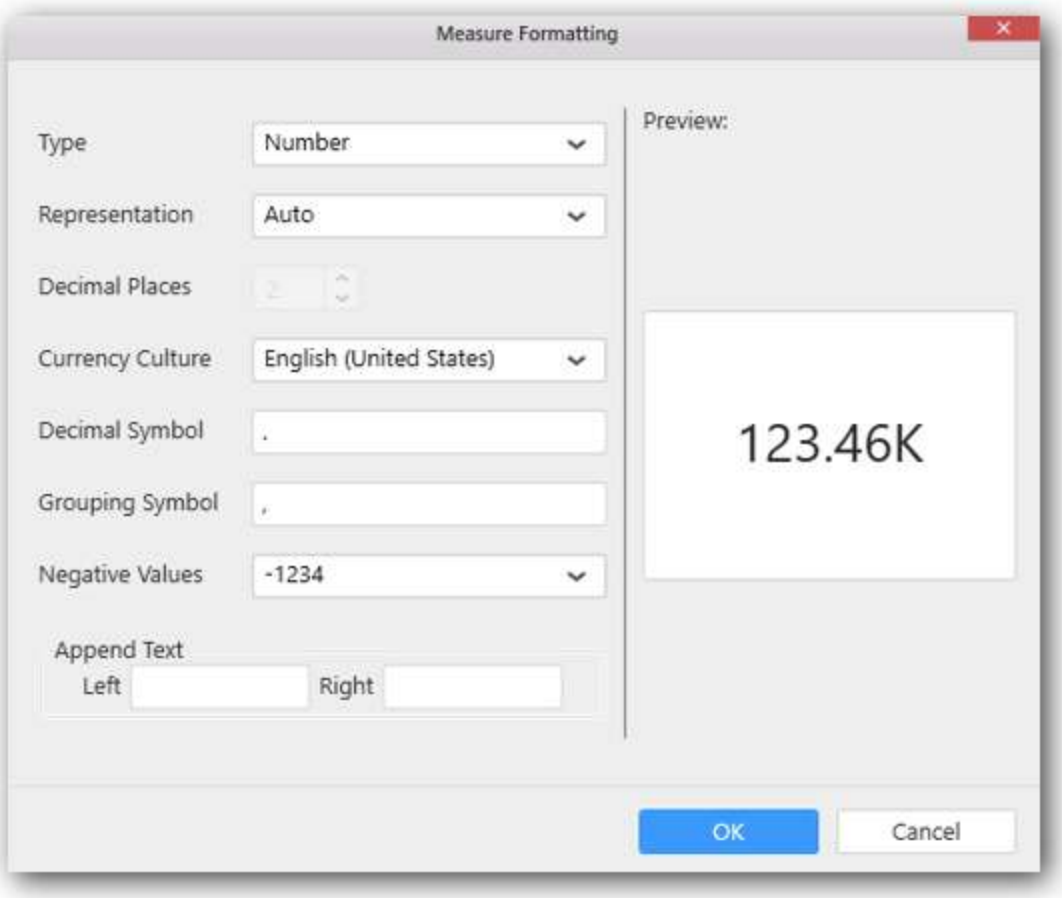
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



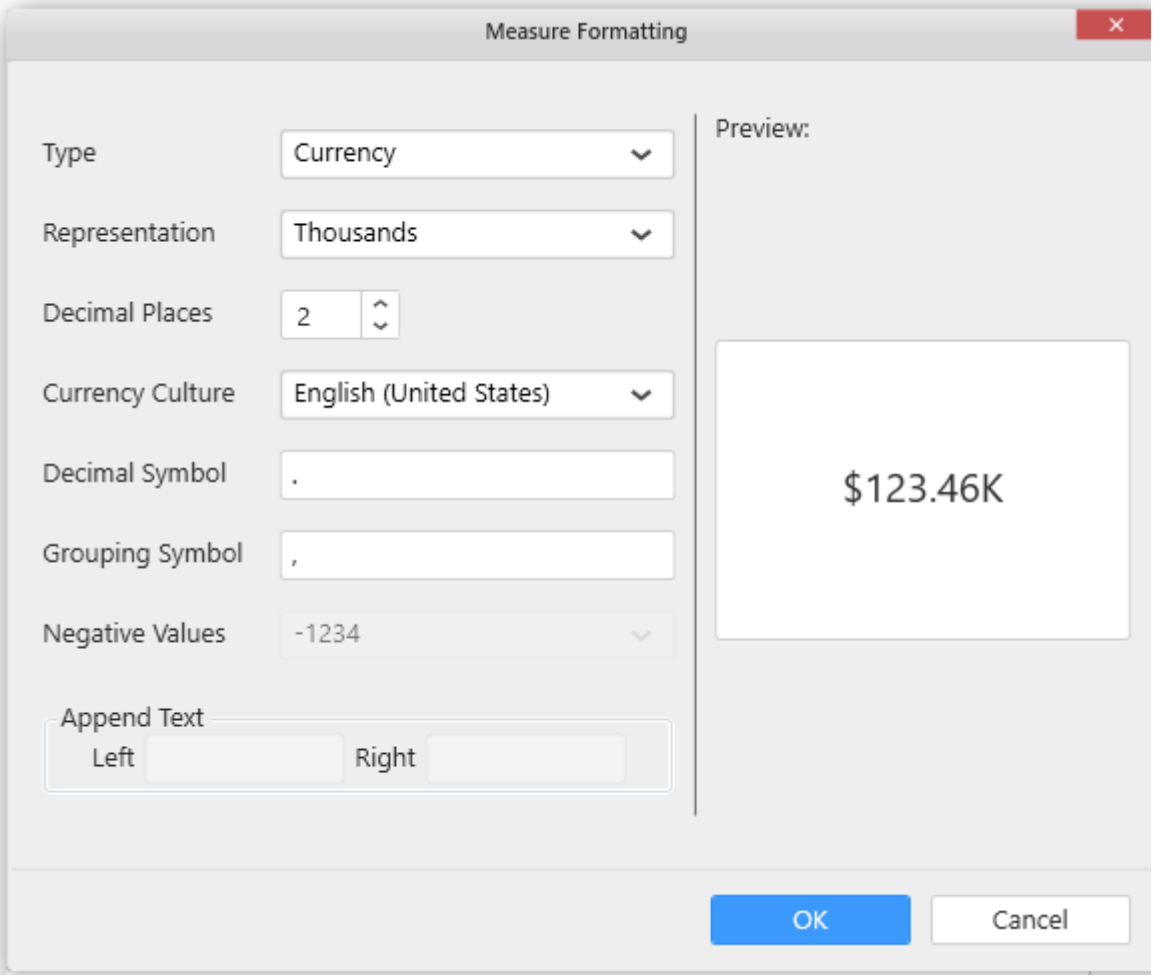
Select Clear option to clear the defined filter.



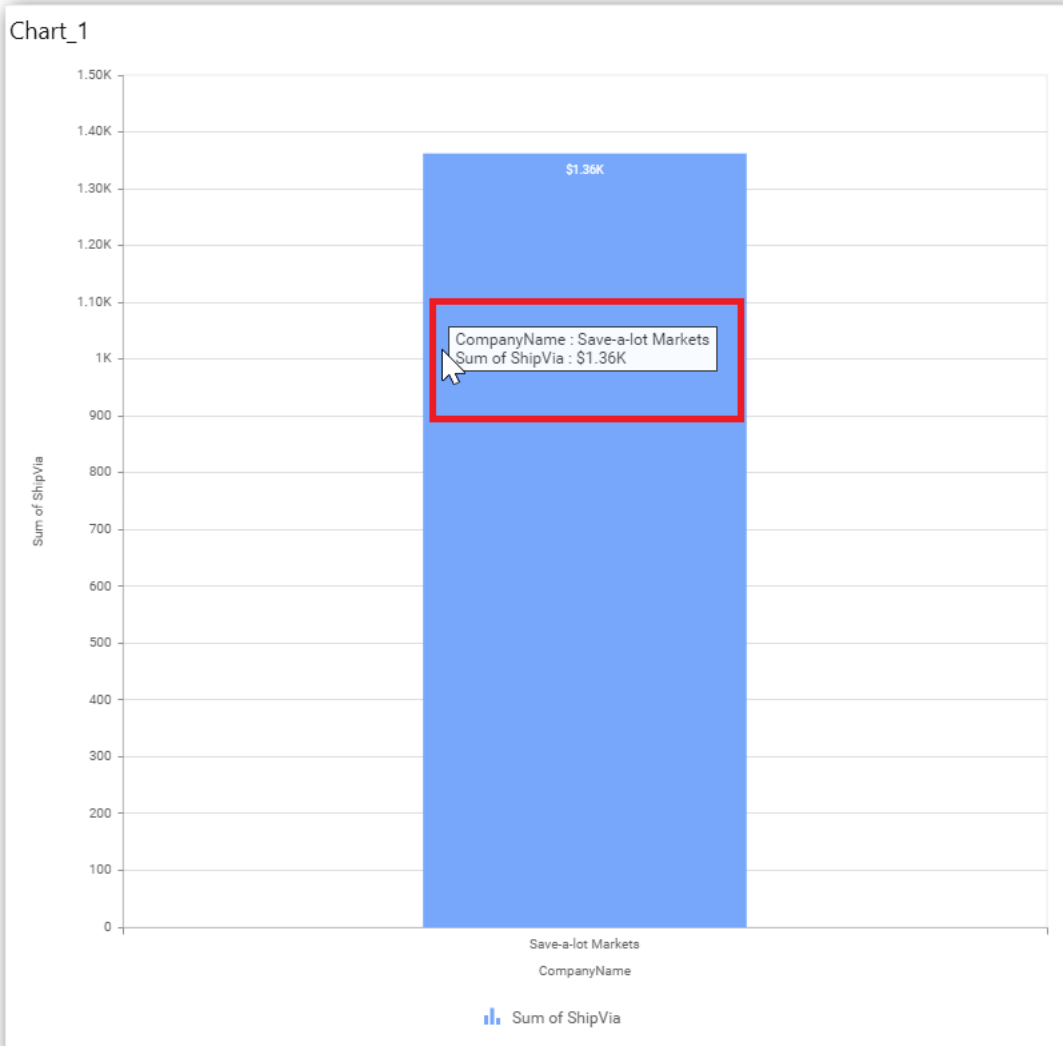
Select Format option to define the display format to the values in the column through Measure Formatting window.



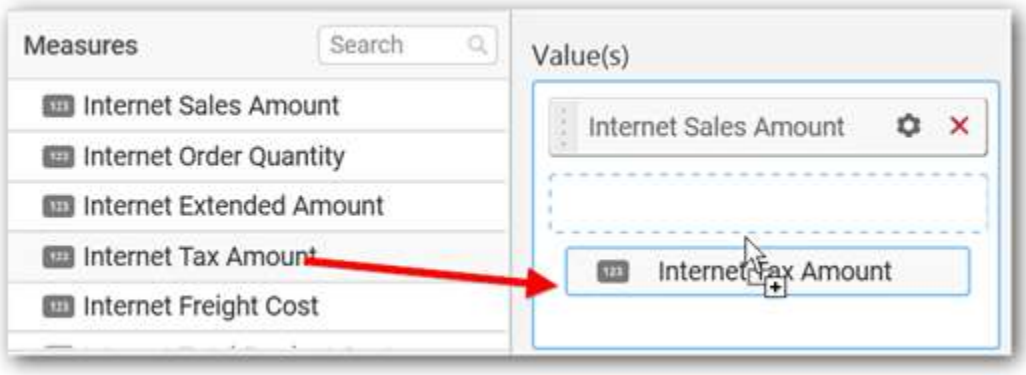
Choose the options you need and click OK.

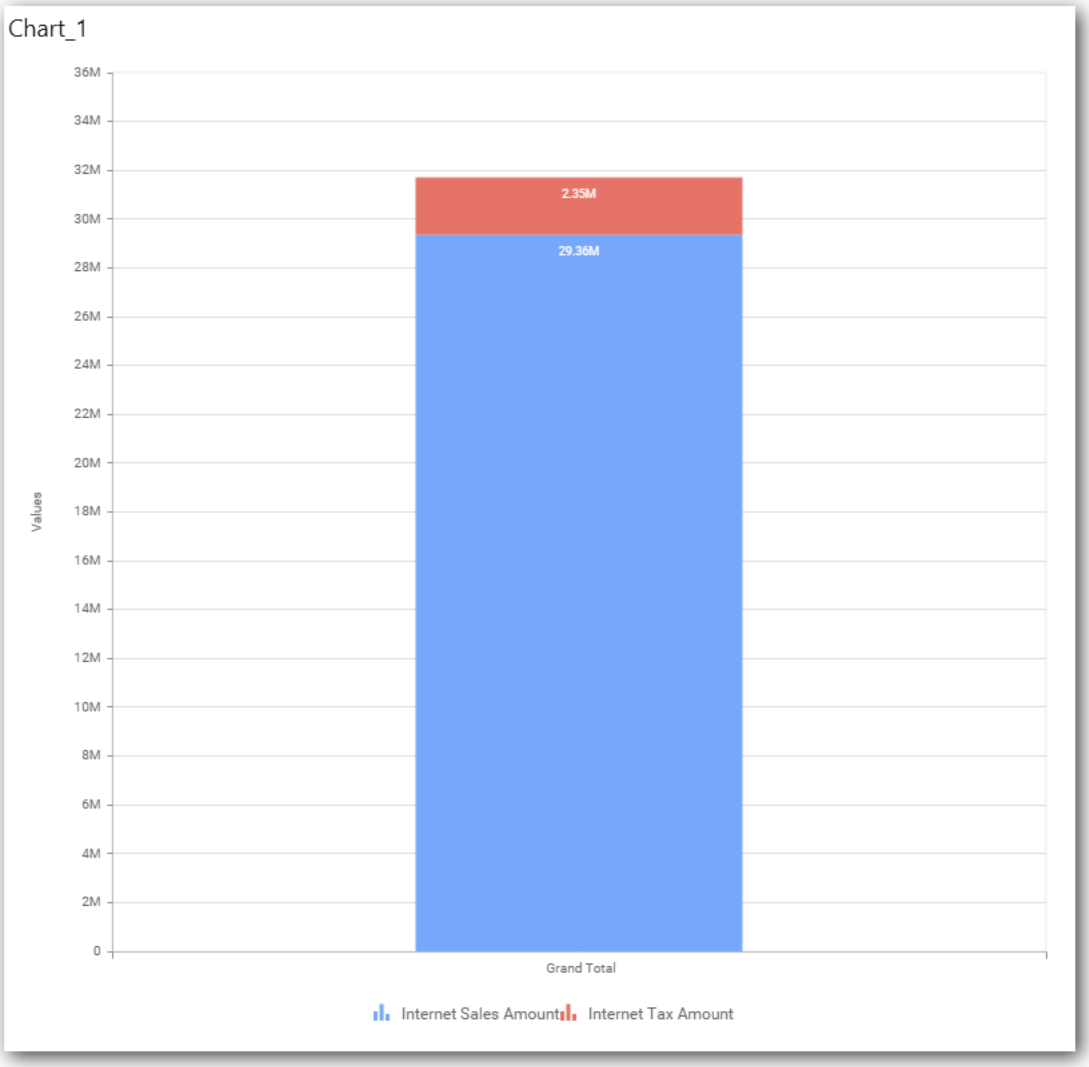


Now the Chart will be rendered like this.



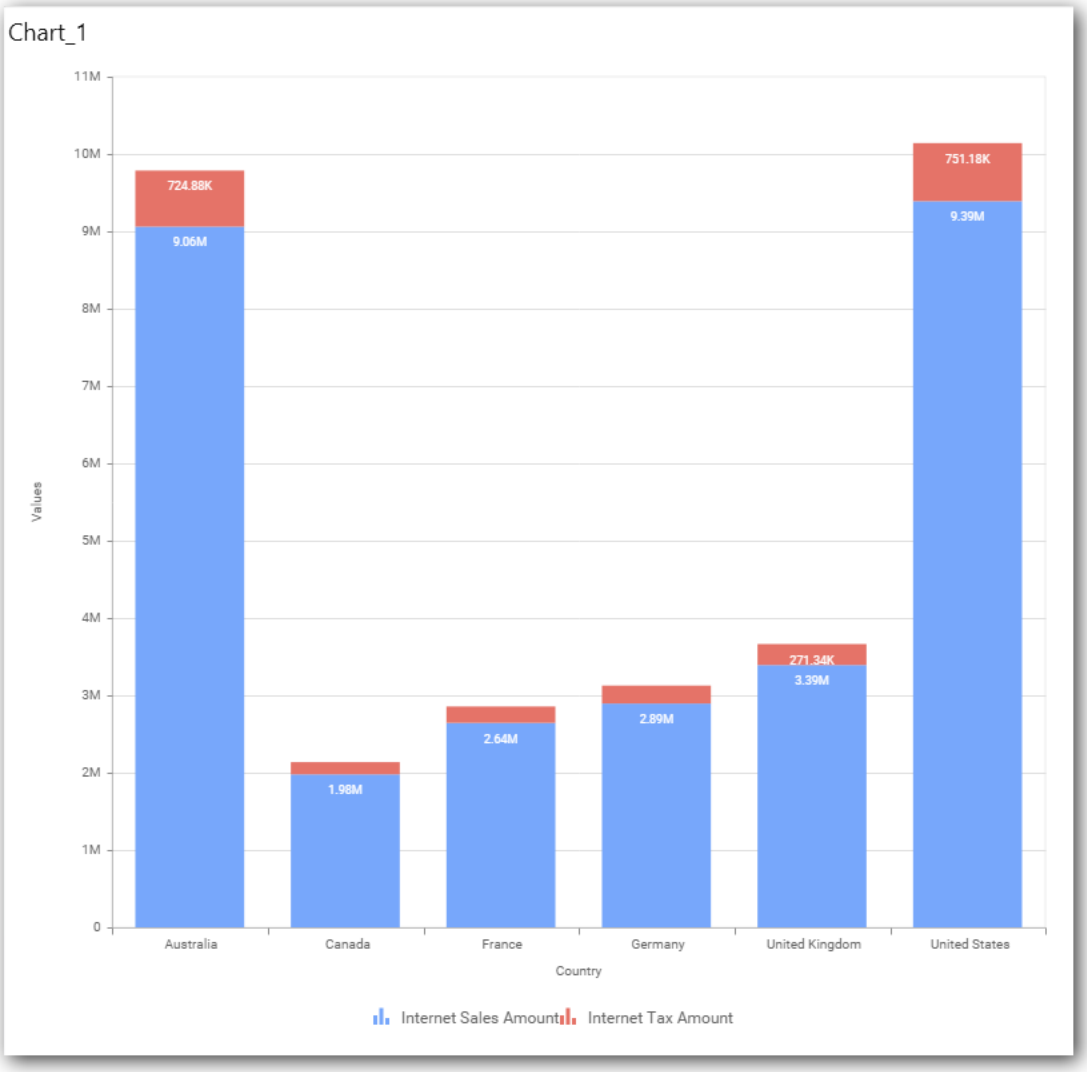
You can also add more than one column to the Value(s) section.



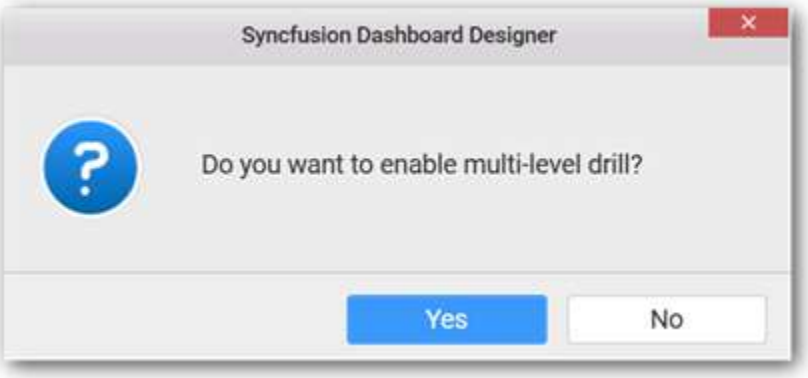


Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.

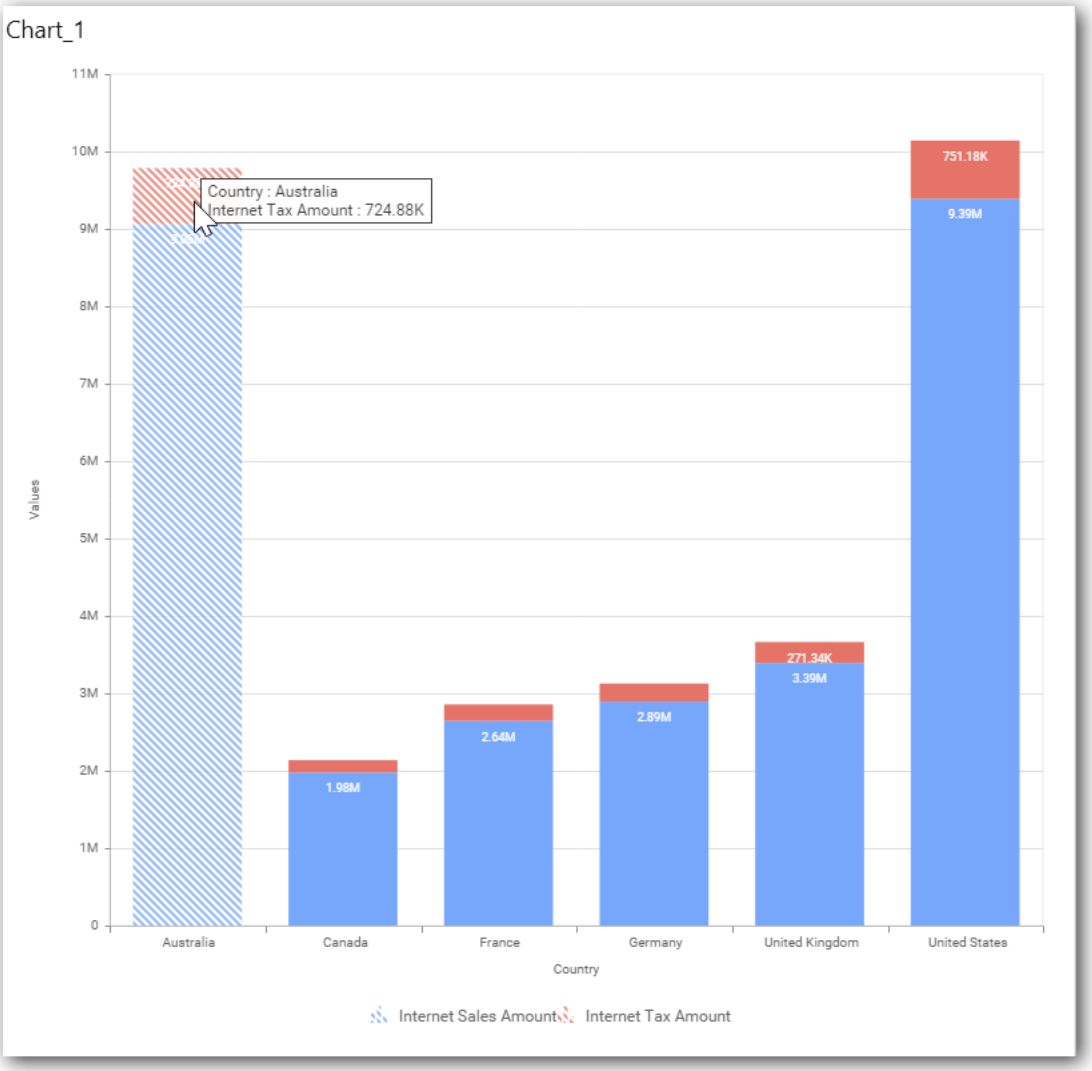


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

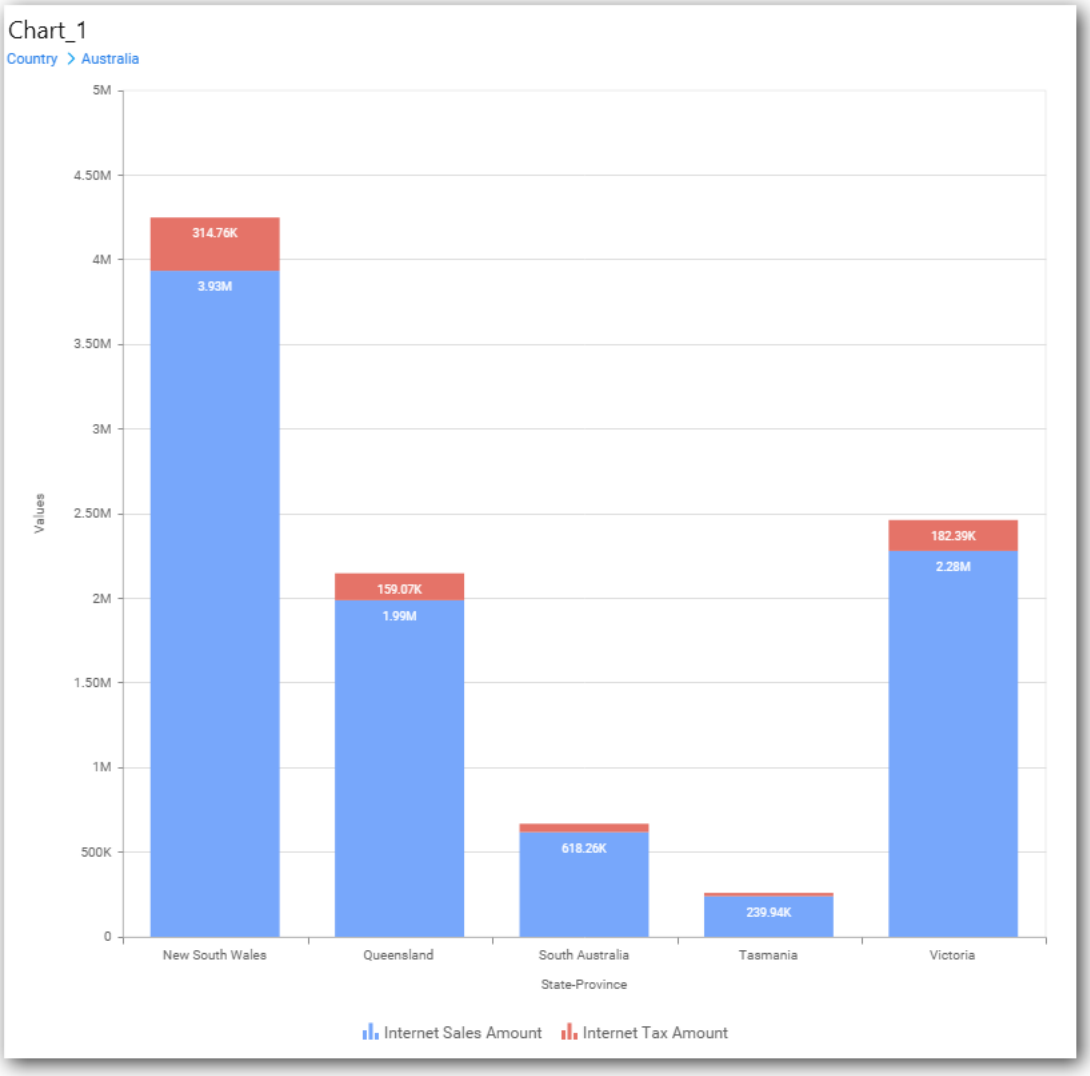


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

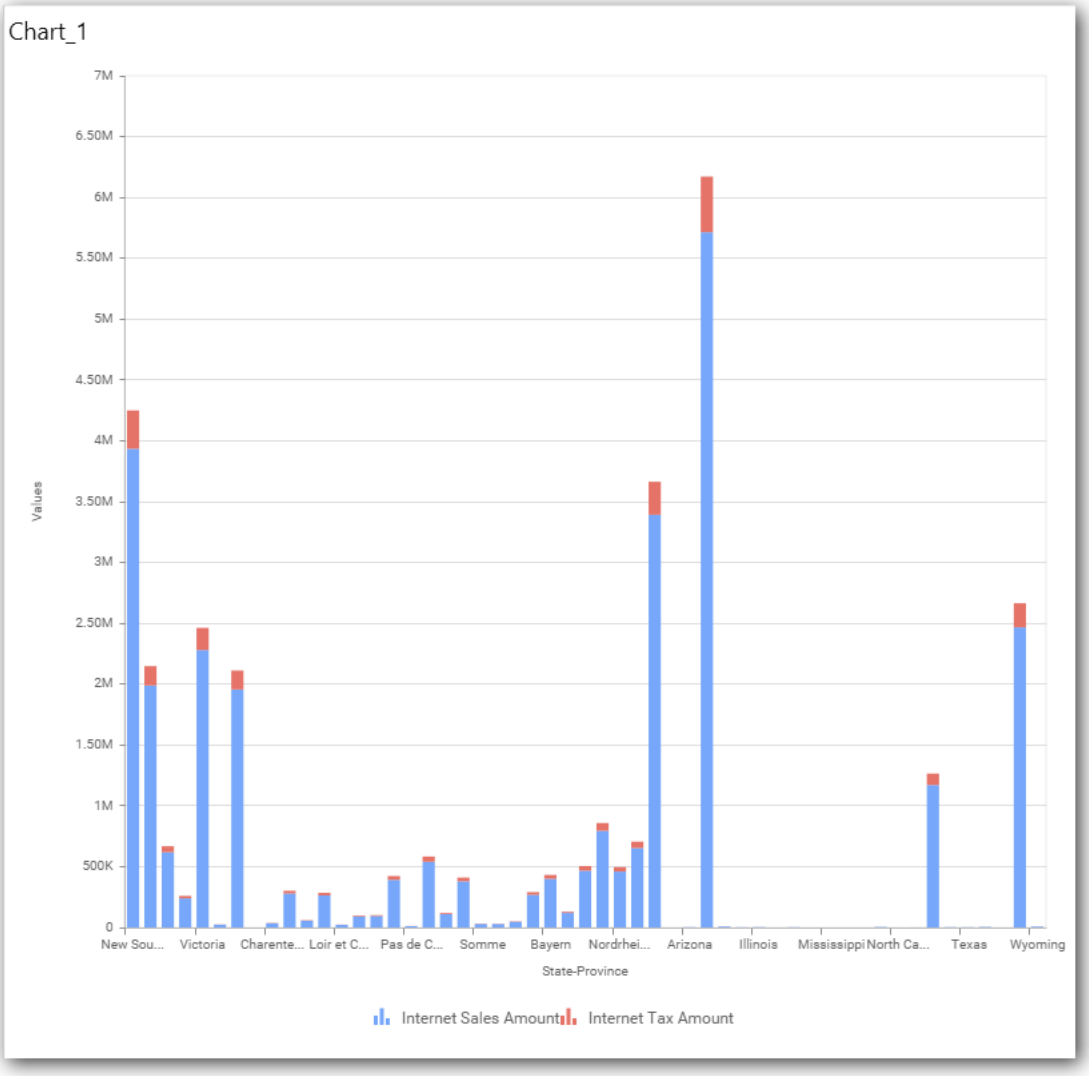
Click the respective data value marker in chart to drill into its inner level.



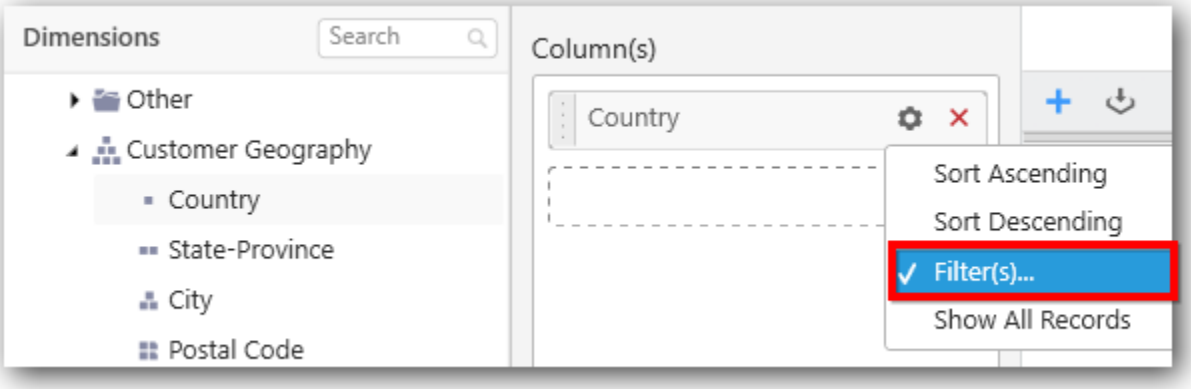
The drilled view of the chart is follows.



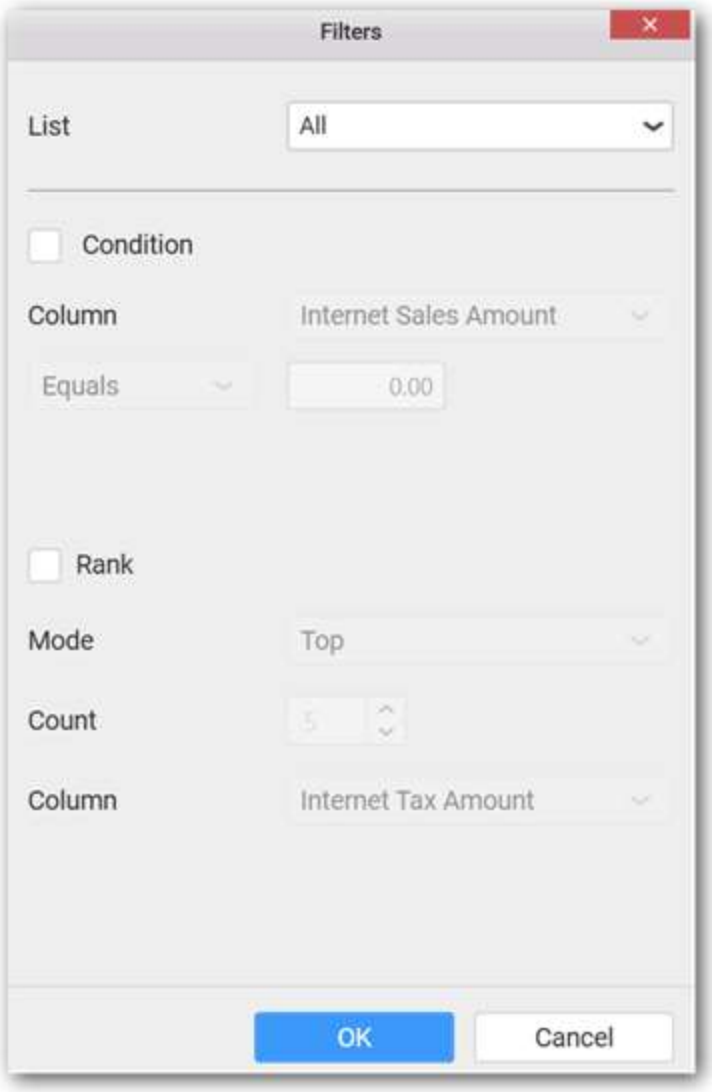
Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level.



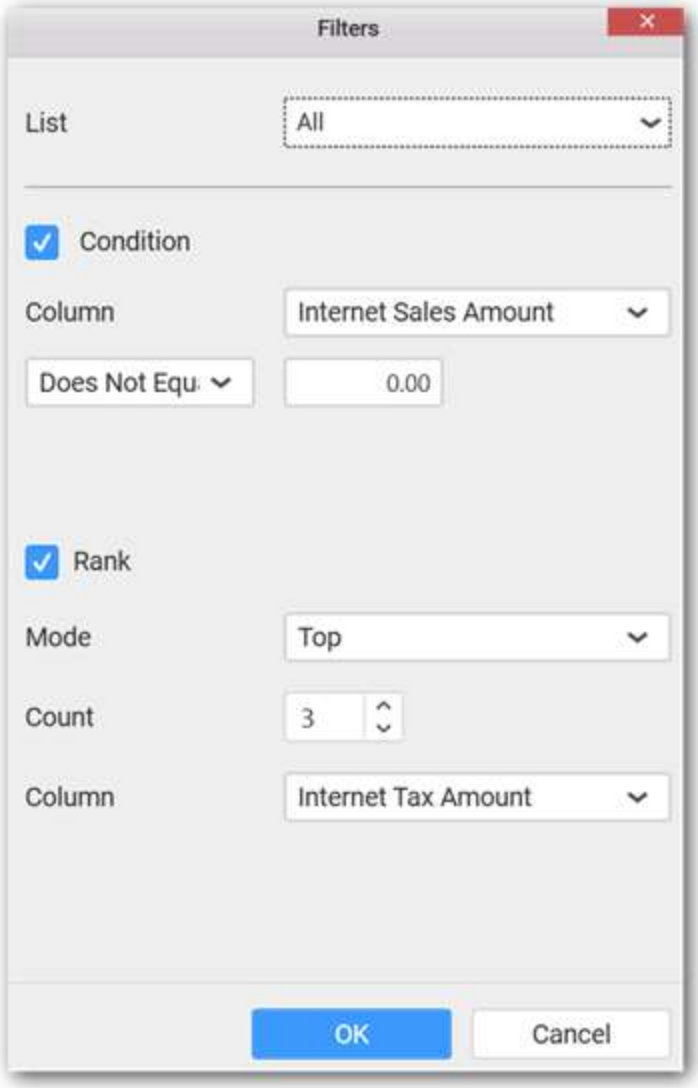
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



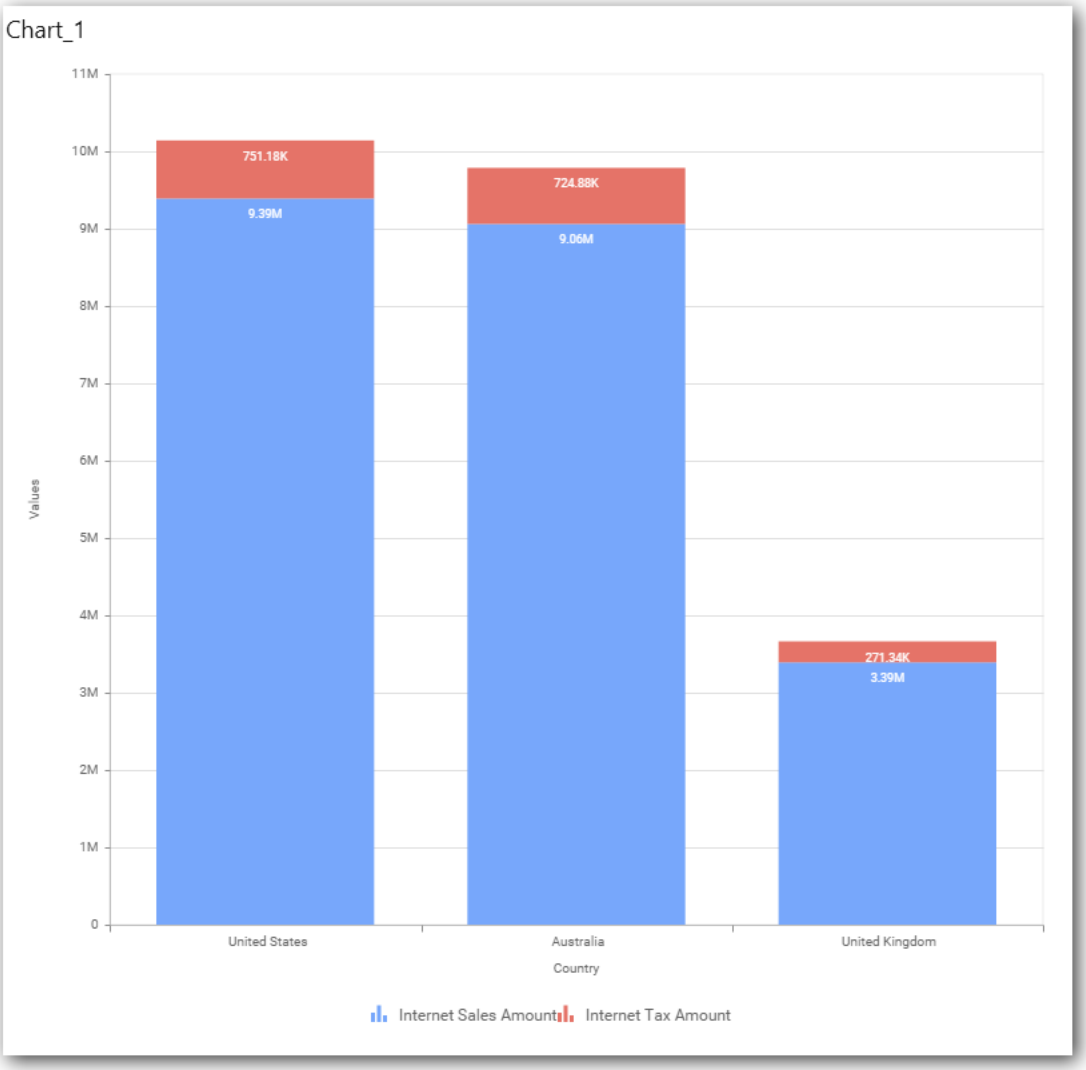
Select Filter(s)... to launch the Filters window.



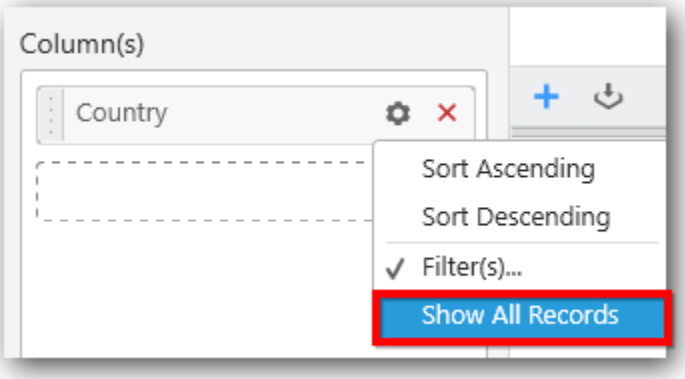
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

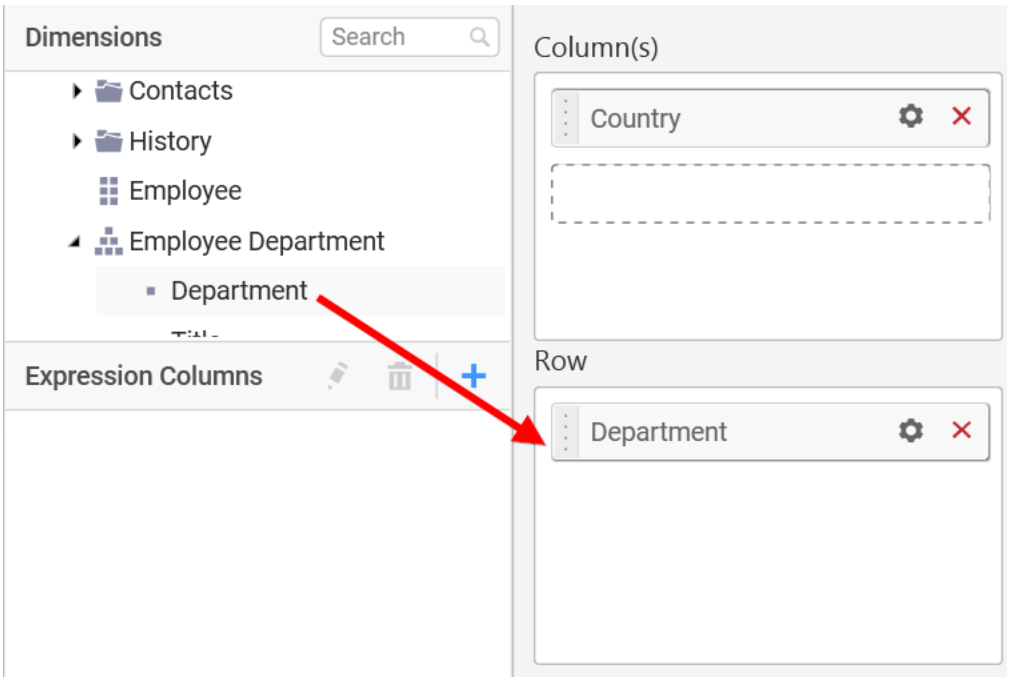


To show all records again click on Show All Records.



Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



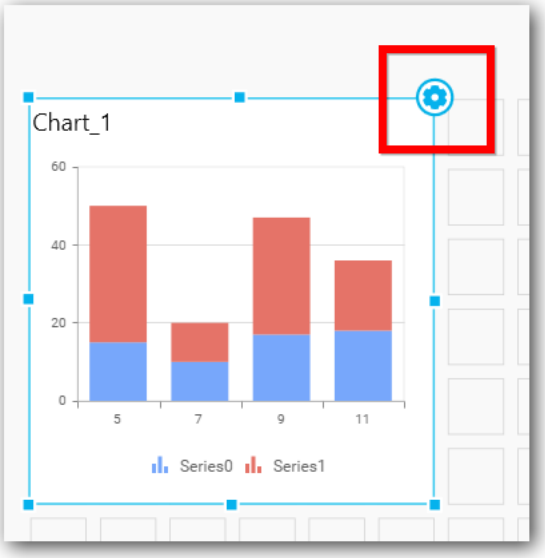
The chart will be rendered in series as shown in the image below.

How to format Stacked Column Chart?

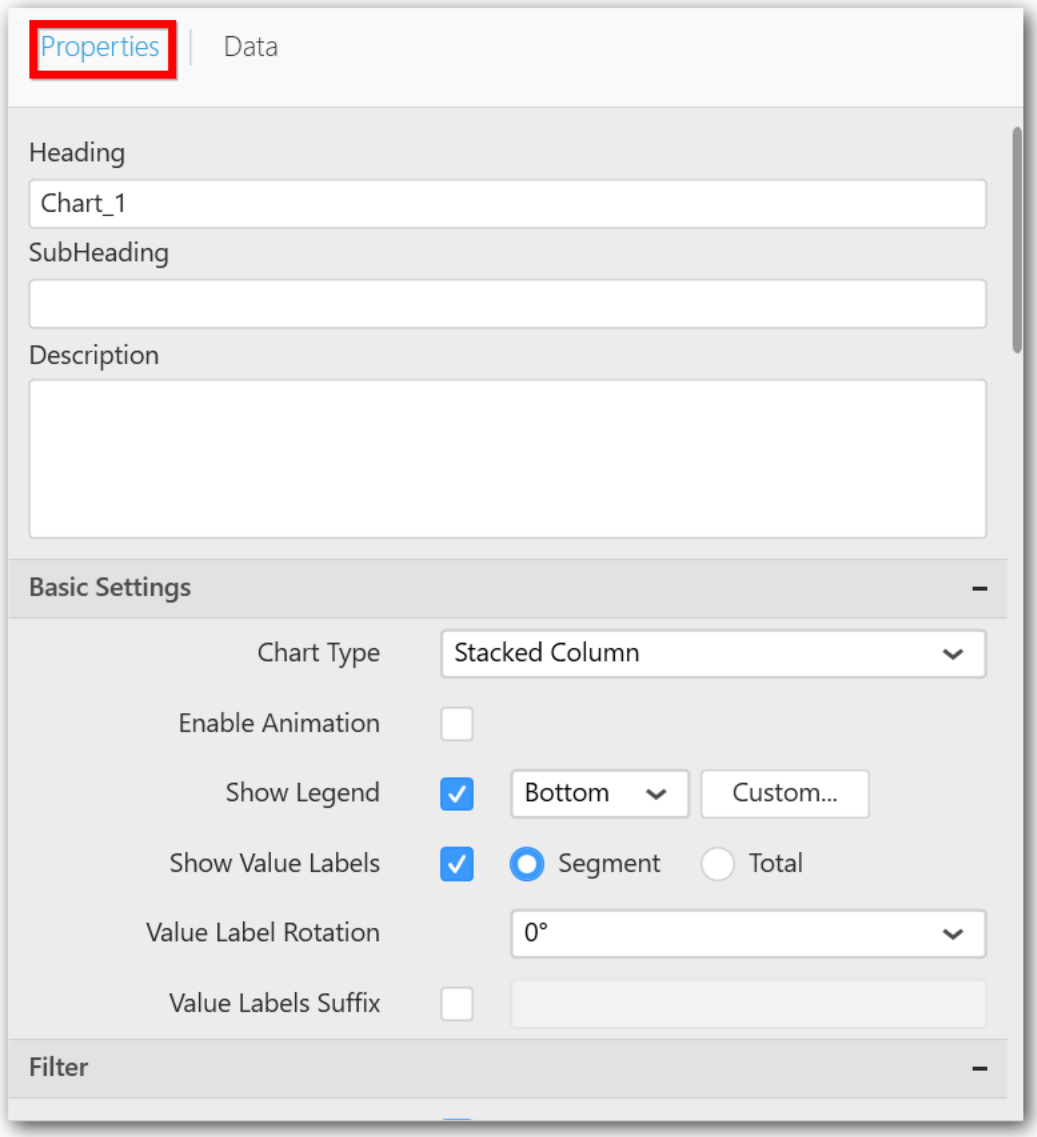
You can format the stacked column chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into stacked column chart follow the steps

1. Drag and drop the stacked column chart into canvas and resize it to your required size.
2. Configure the data into stacked column chart.
3. Focus on the stacked column chart and Click on Widget Settings.

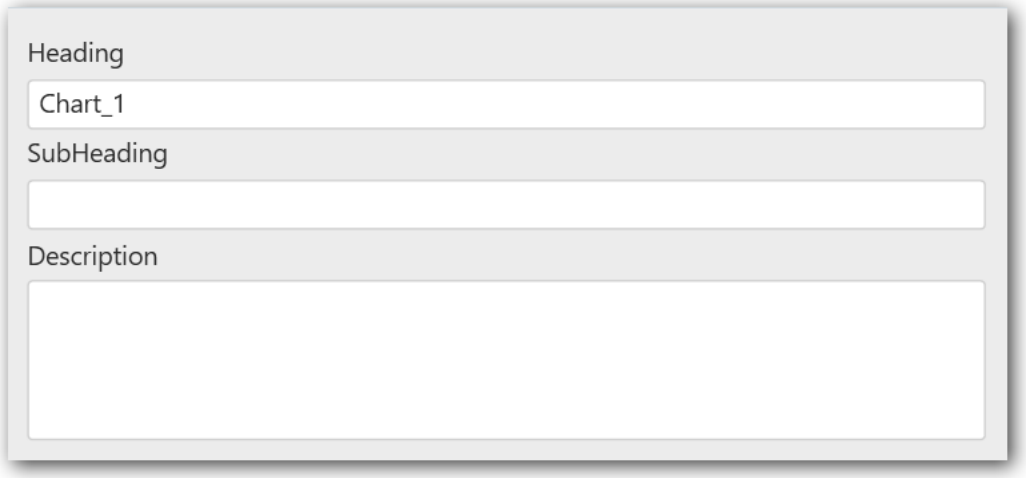


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this stacked column chart widget.

SubHeading

This allows you to set sub-title for this stacked column chart widget.

Description

This allows you to set description for this stacked column chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

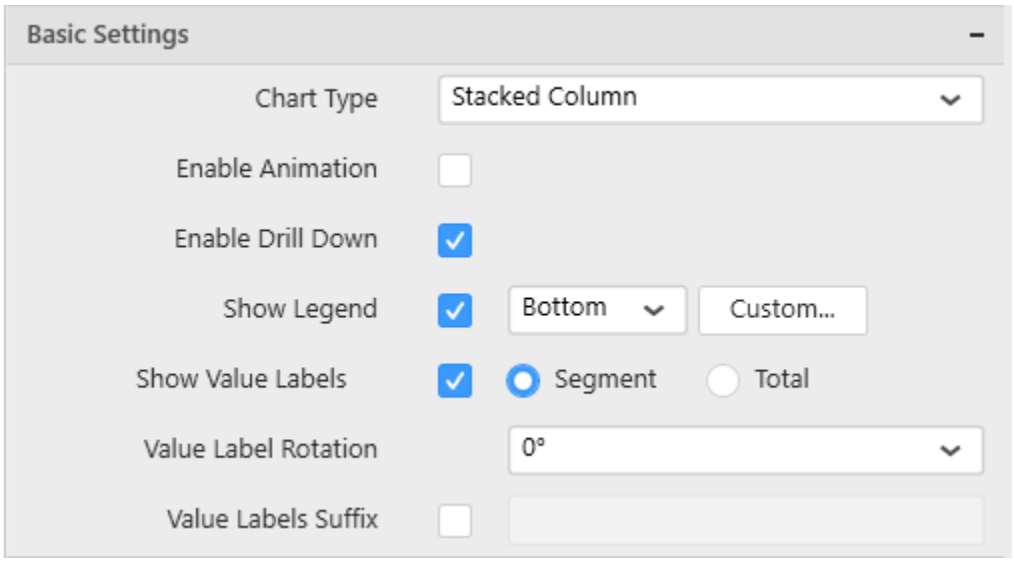


Chart Type

This allows you to switch the widget view from current chart type to another convertible chart type.

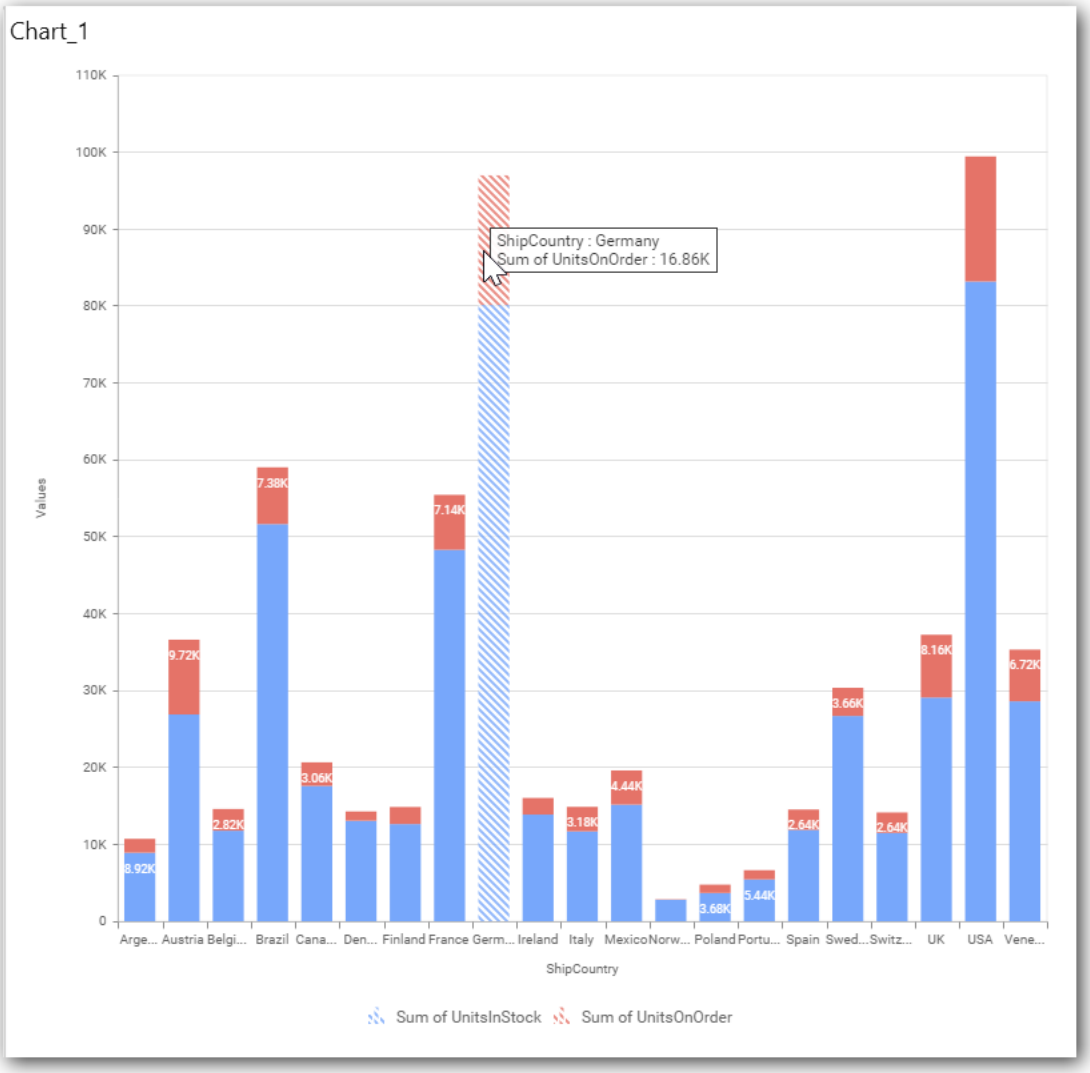
Enable Animation

This allows you to enable the series rendering in animated mode.

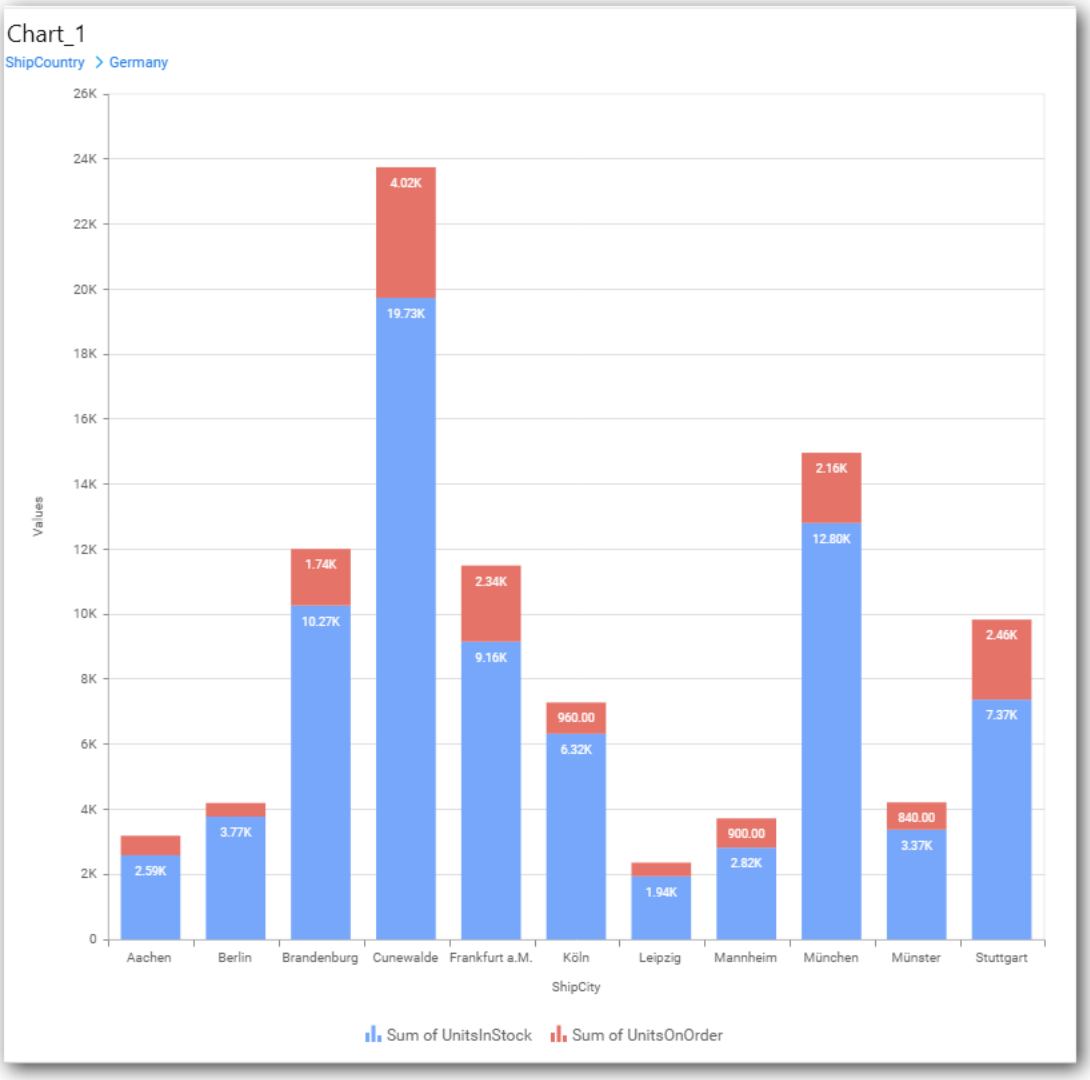
Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View

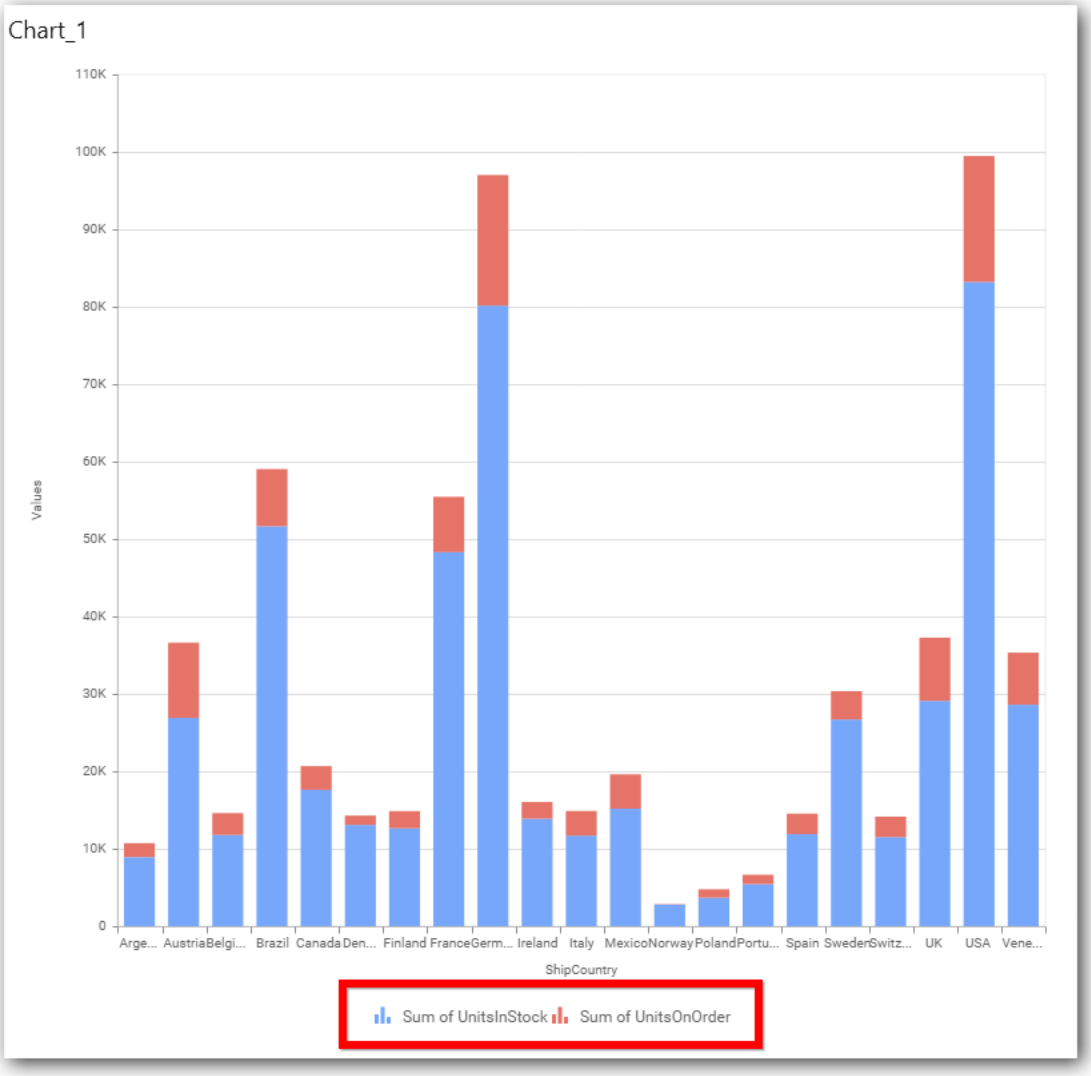


Drilled View



Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of Custom Legend Text will allow us to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

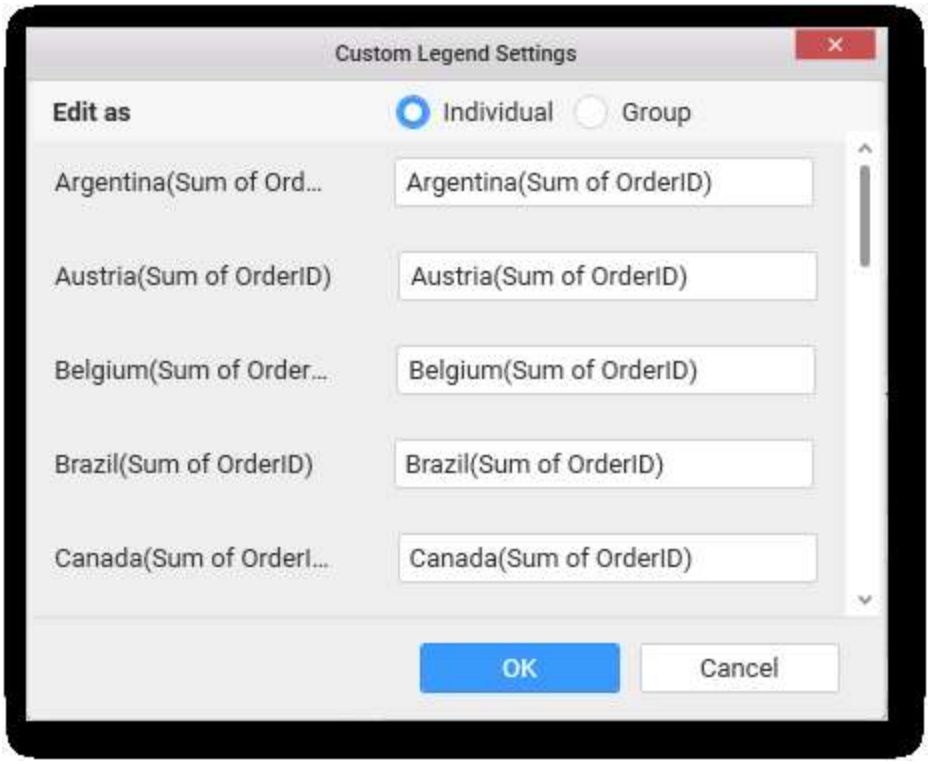
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

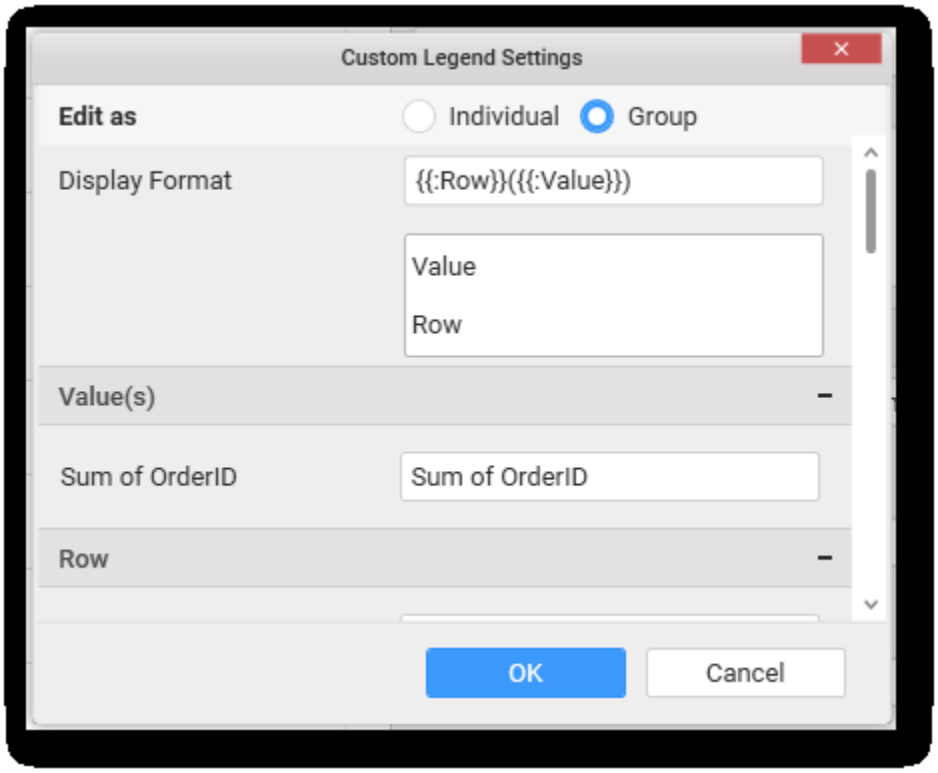
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

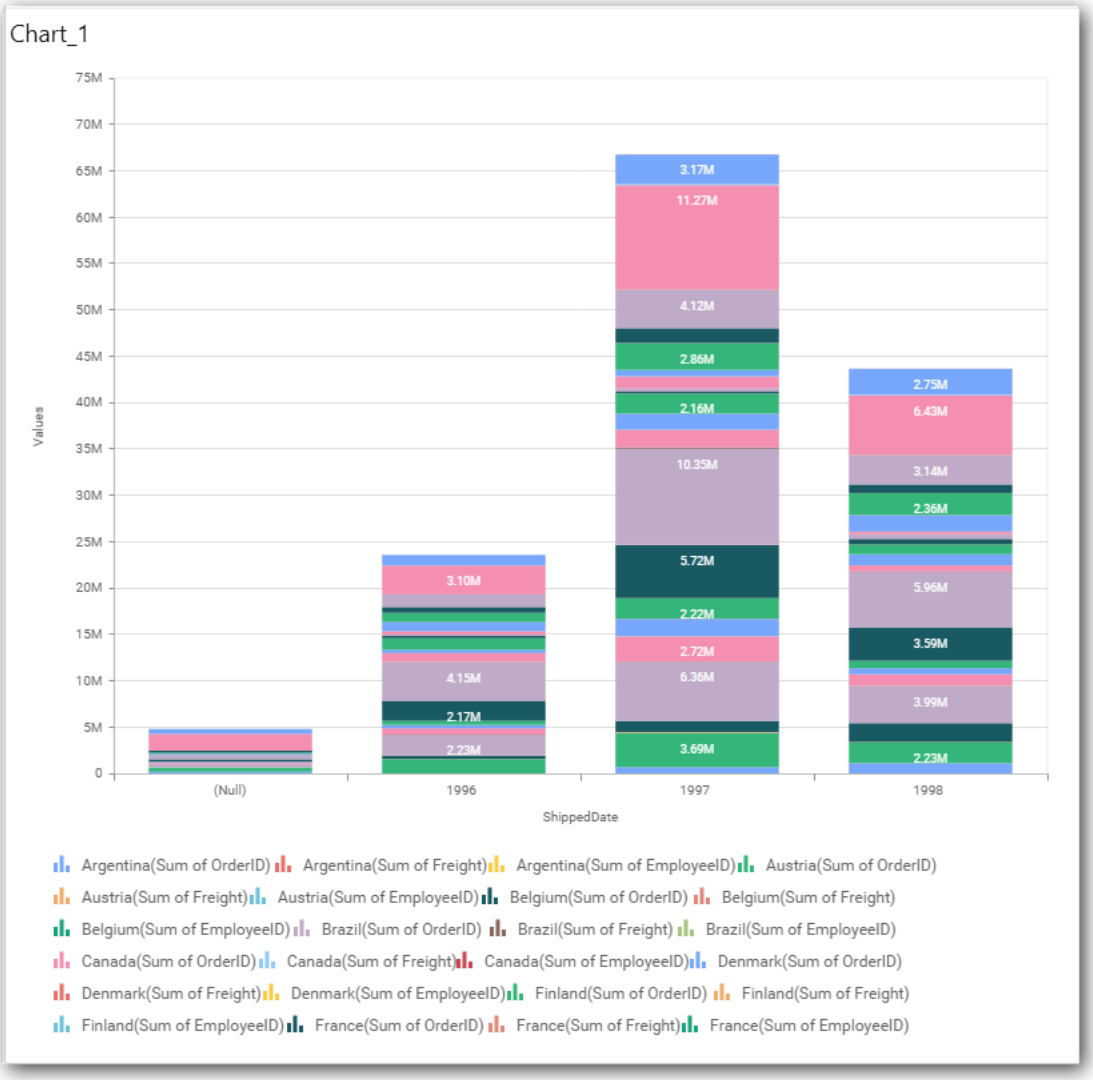


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}} ({{"{{"}} : Value {{}}}), then Legend series will display like Argentina (Sum of Order ID)



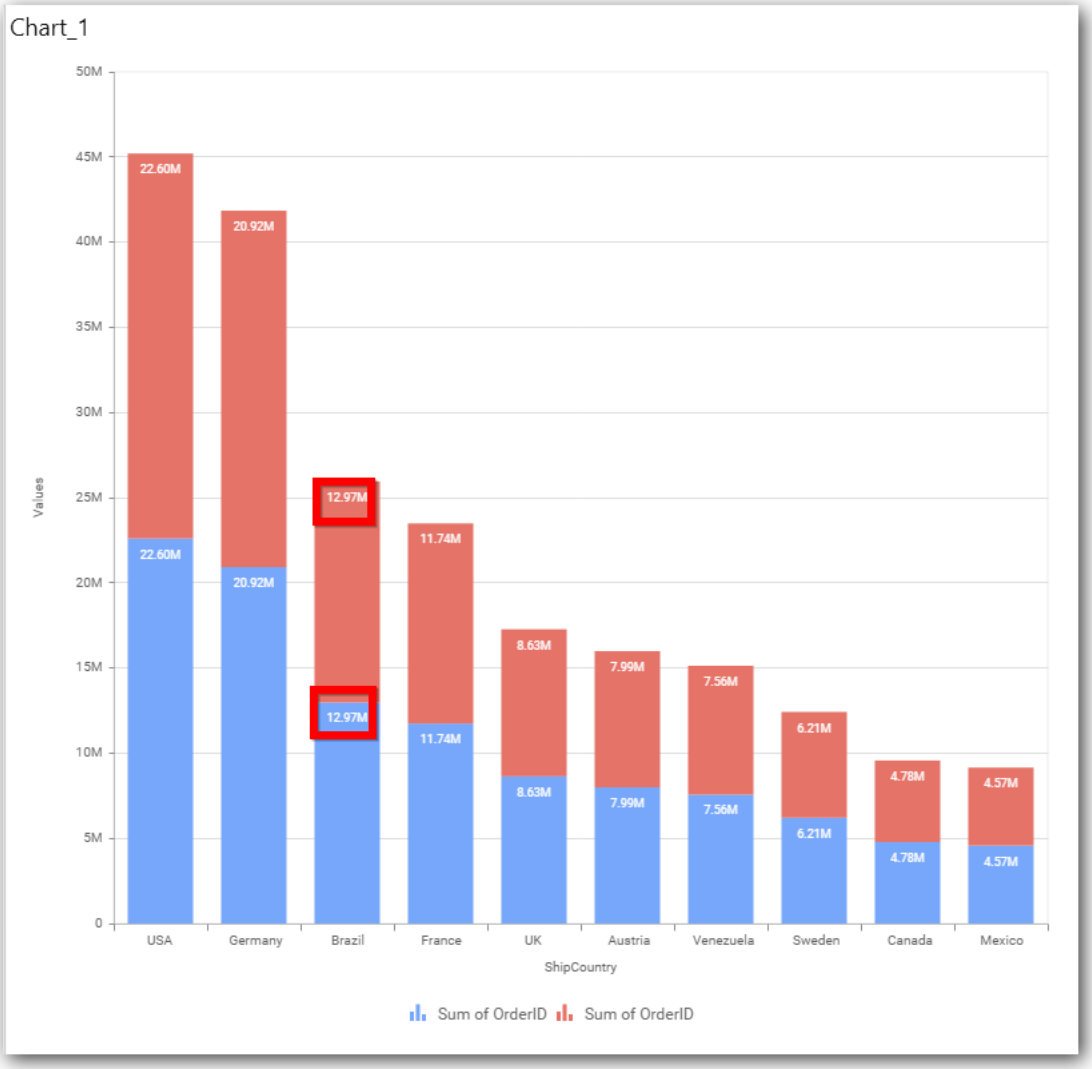
Show Value Labels

This allows you to toggle the visibility of value labels. When you toggle on, two options will be provided to change the display mode of the labels.

1. Segment
2. Total

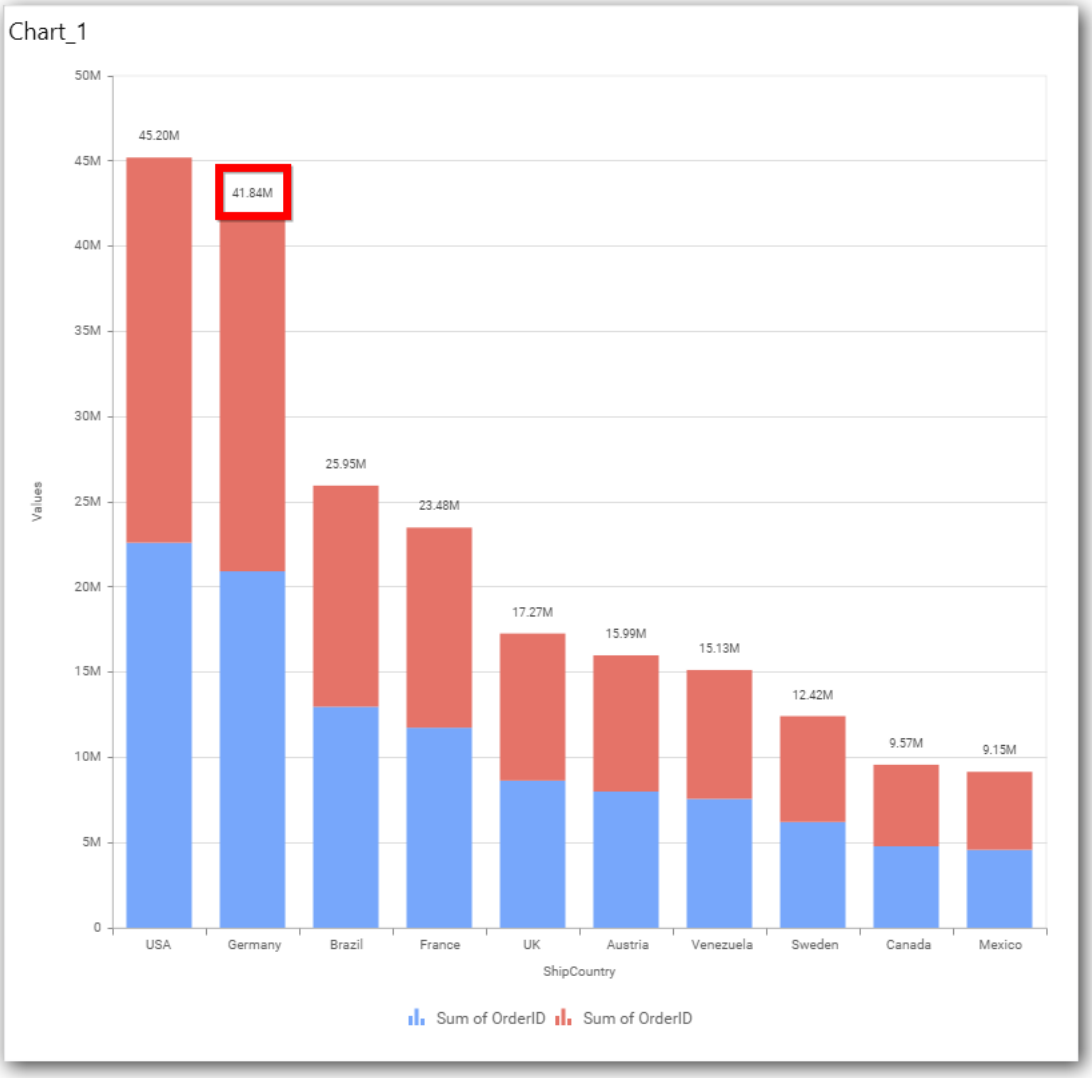
Segment

Displays value label for each series segment.



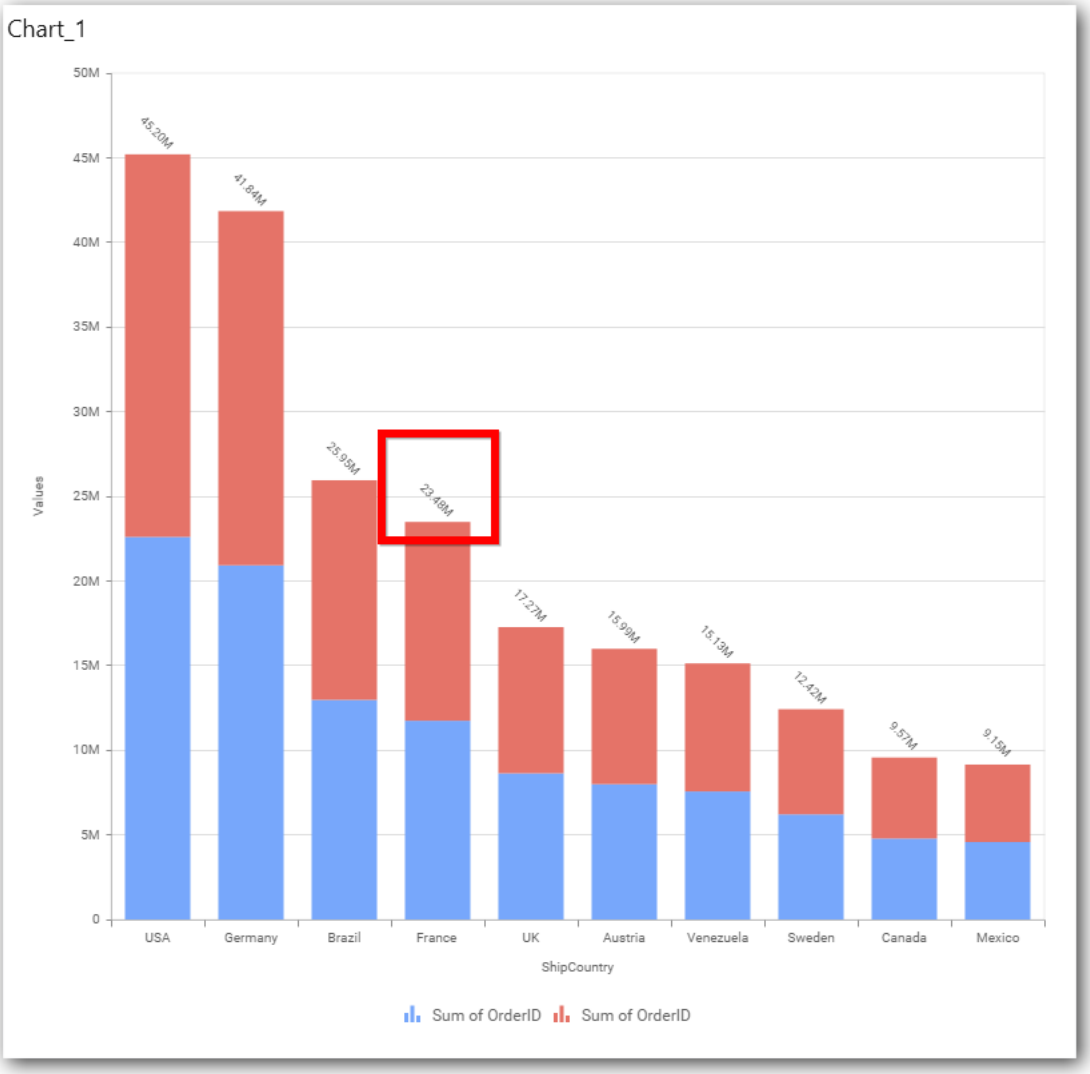
Total

Displays sum value label of all the stacked segment on the top most segment.



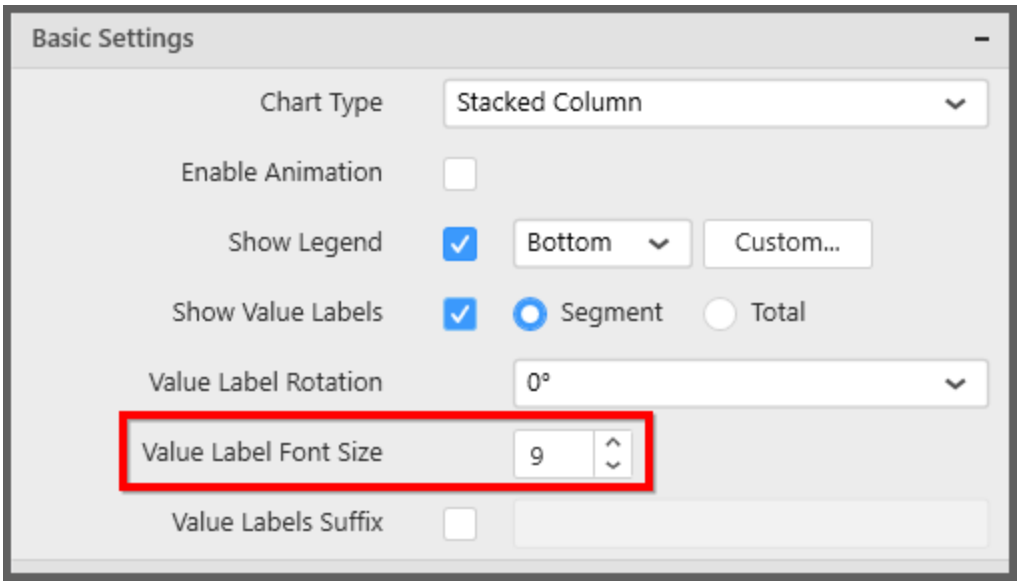
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



Value Label Font Size

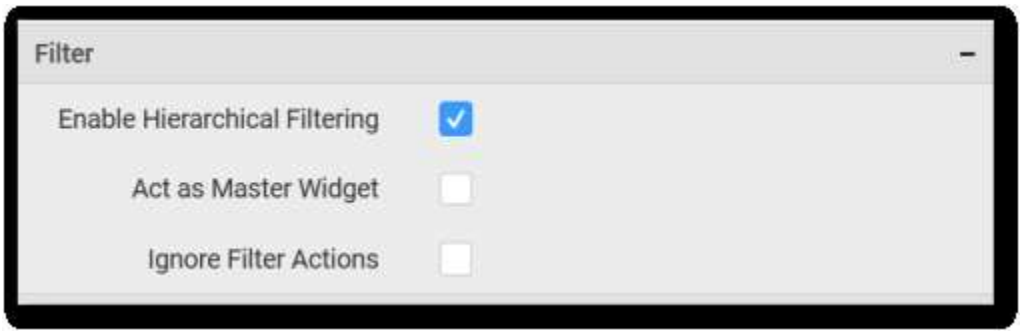
This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.



Value Labels Suffix

Allows you to set suffix to the value labels.

Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

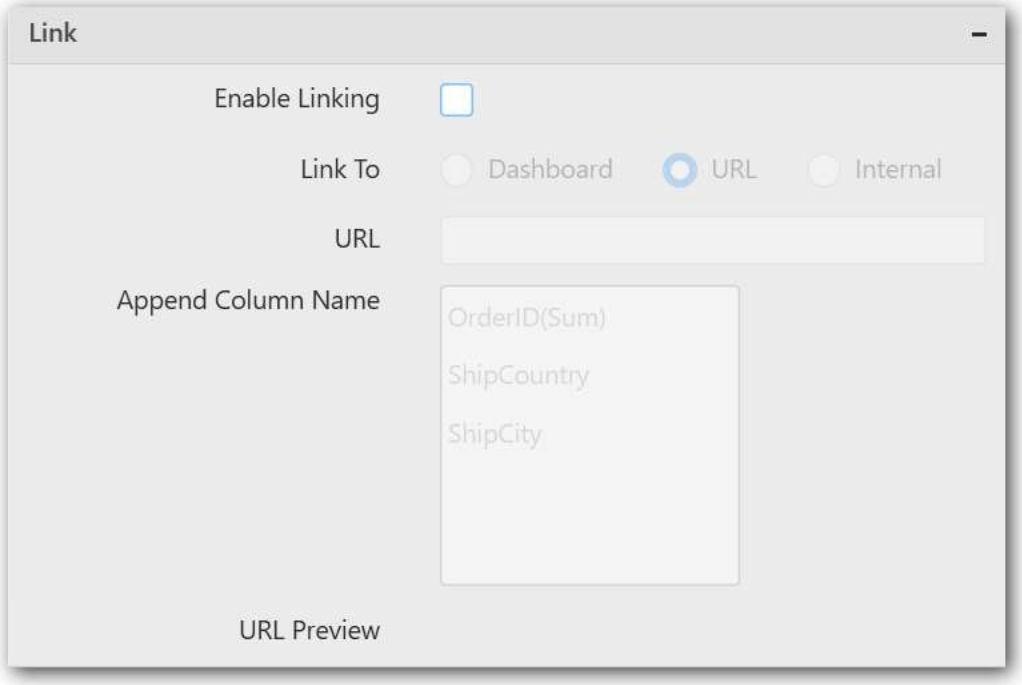
Act as Master Widget

This allows you to define this stacked column chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

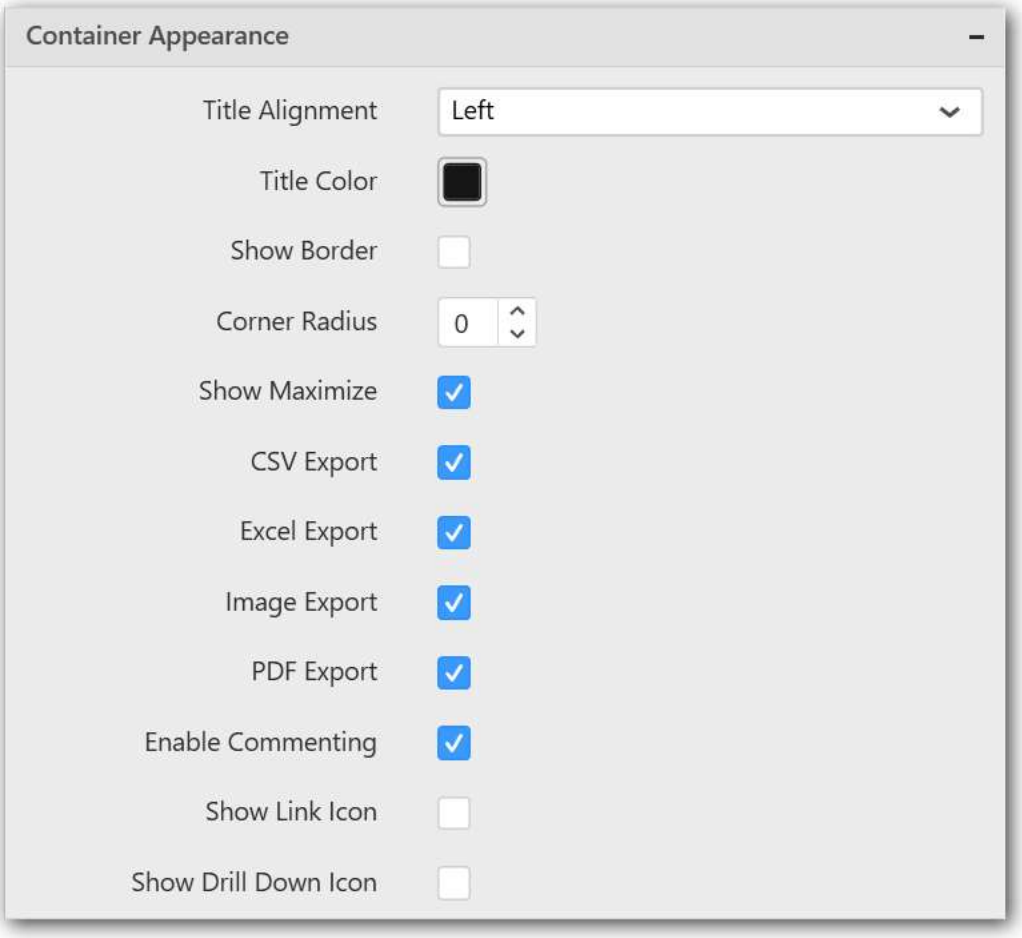
This allows you to define this stacked column chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this stacked column chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this stacked column widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this stacked column widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

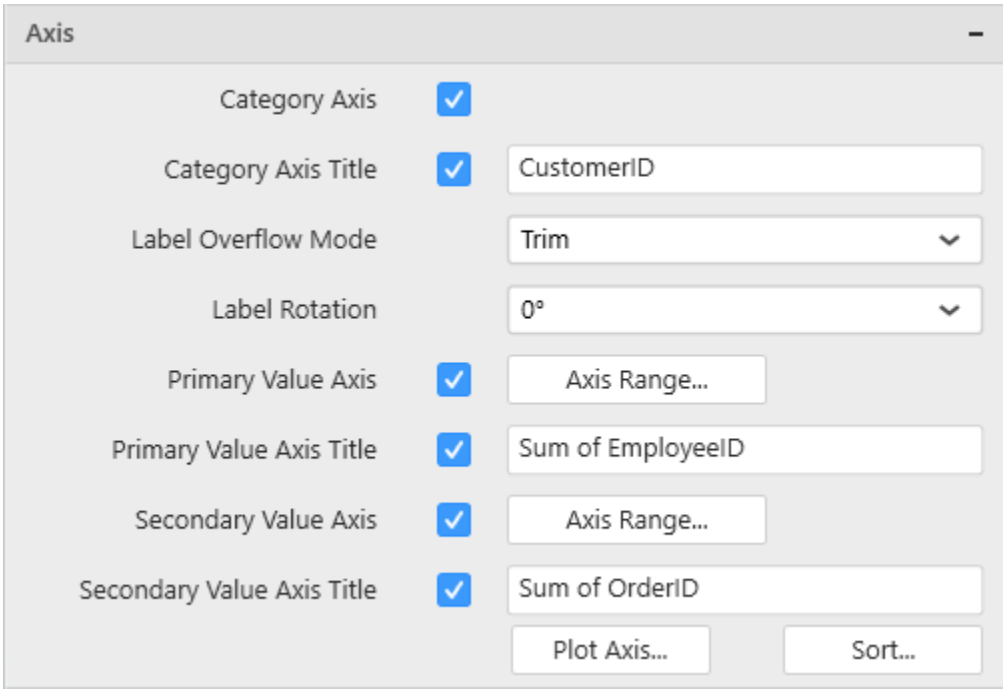
Image Export

This allows you to enable/disable the image export option for this stacked column chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

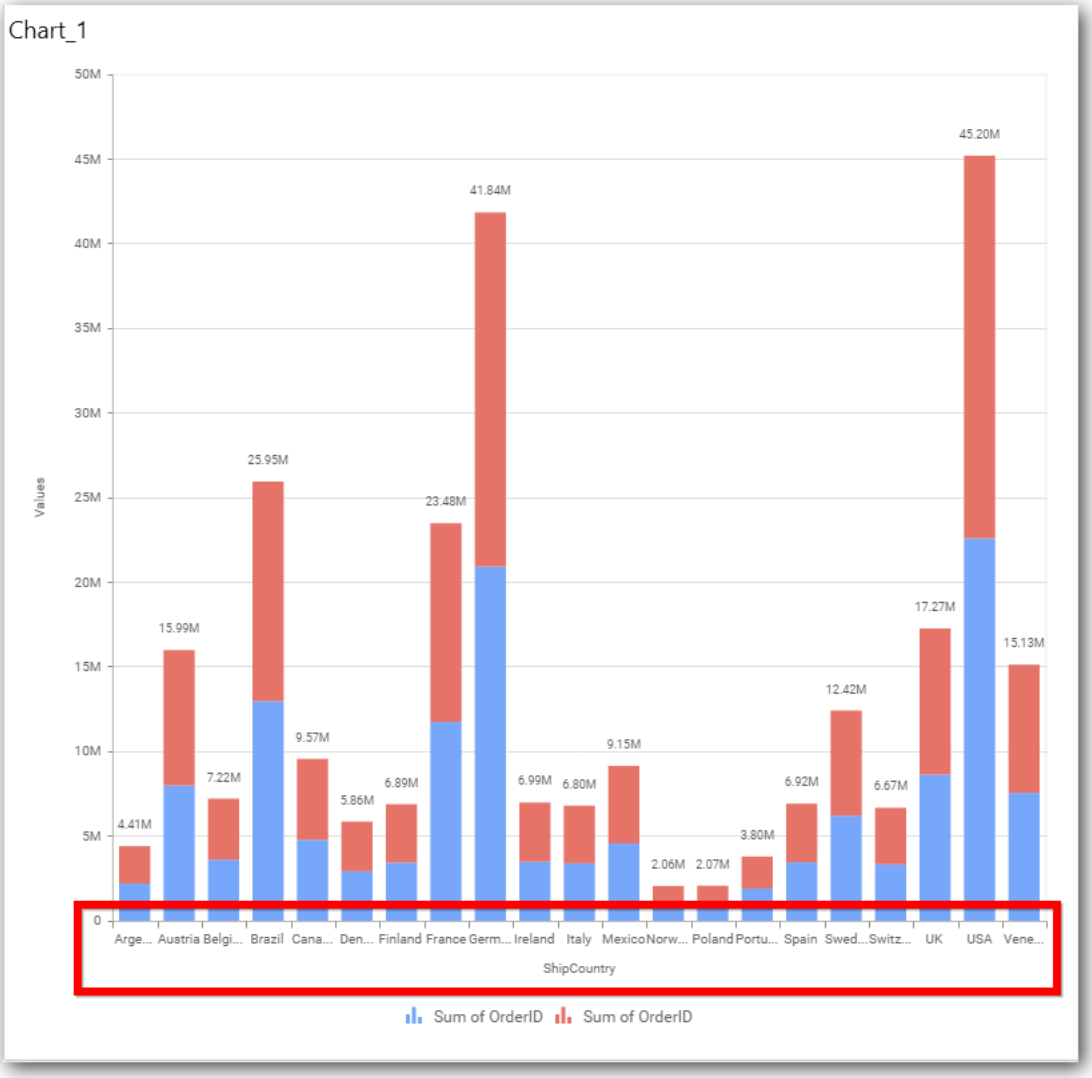
Axis Settings



This section allows you to customize the axis settings in chart.

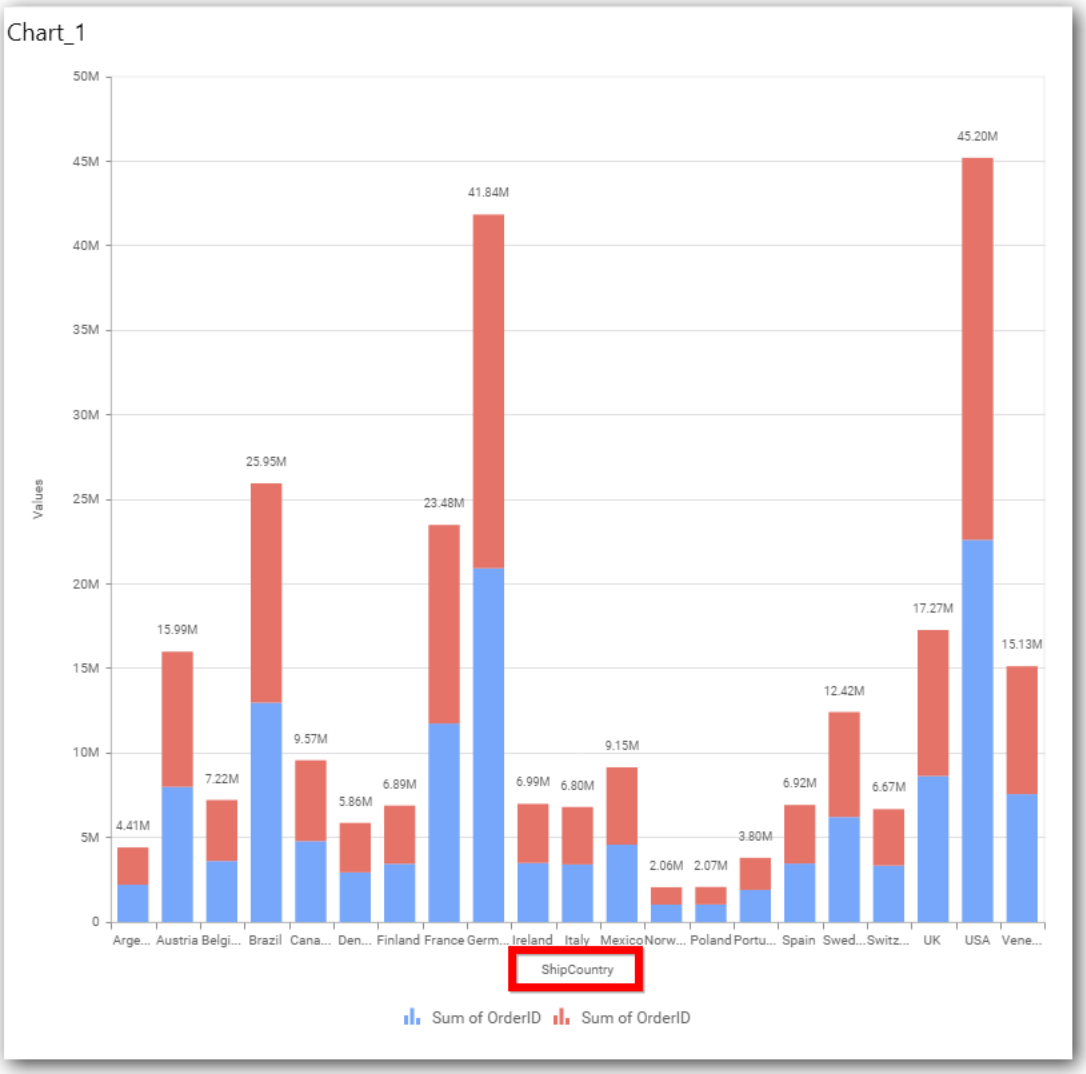
Category Axis

This allows you to enable/edit the option of Category Axis. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

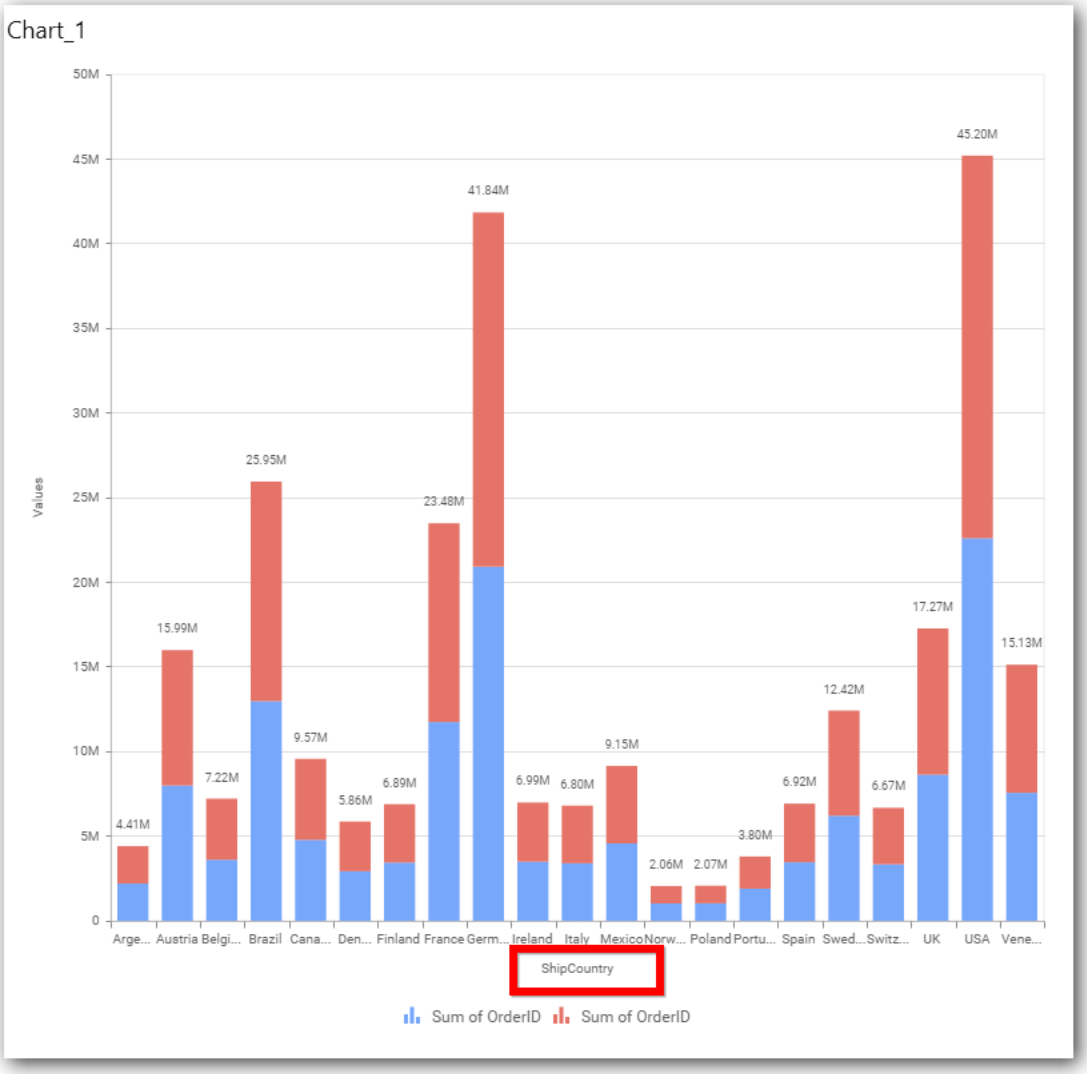


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

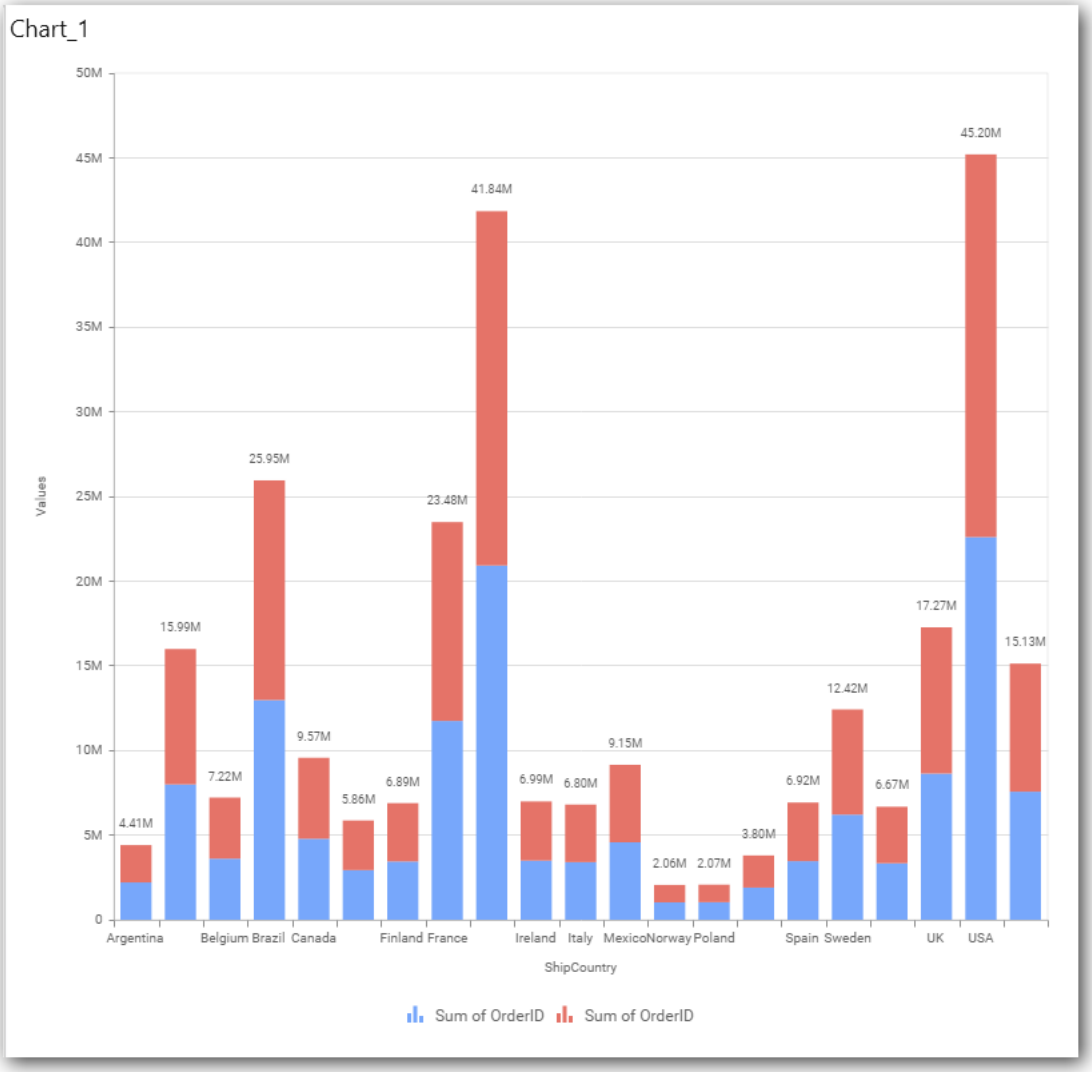
Trim

This option trims the end of overlapping label in the axis.



Hide

This option hides the overlapping label in the axis.

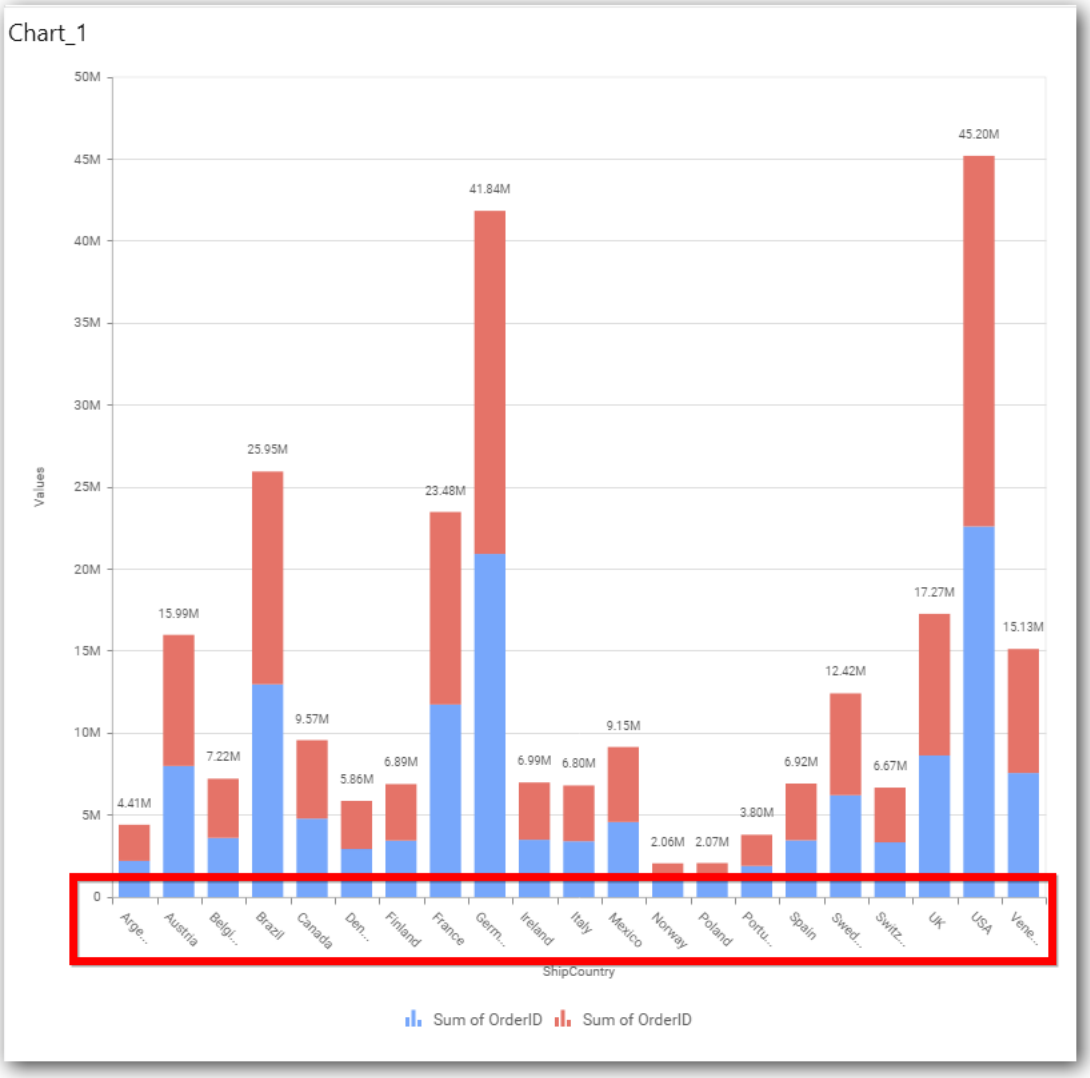


Wrap

This option wraps the lengthy label text in the axis.

Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

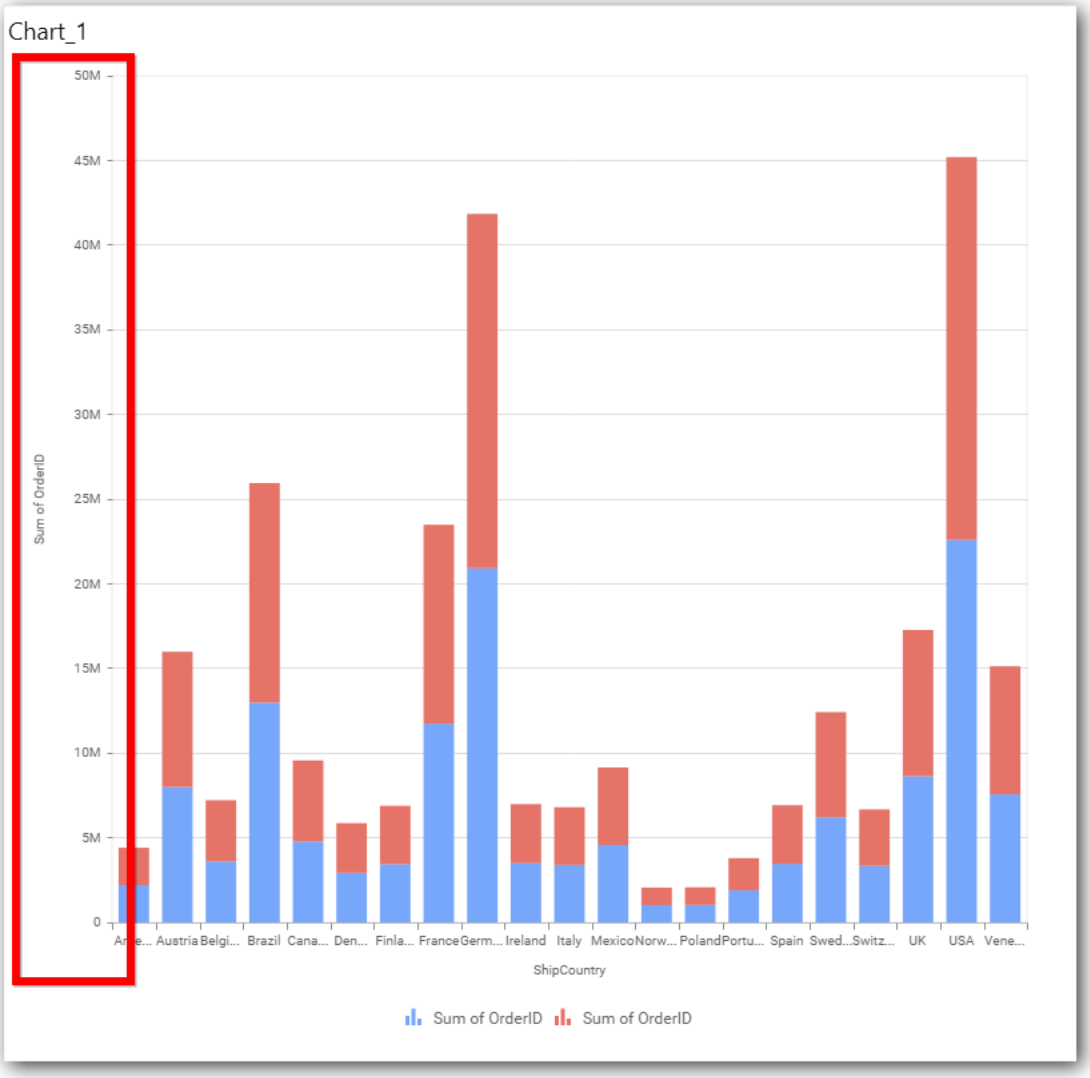


Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.

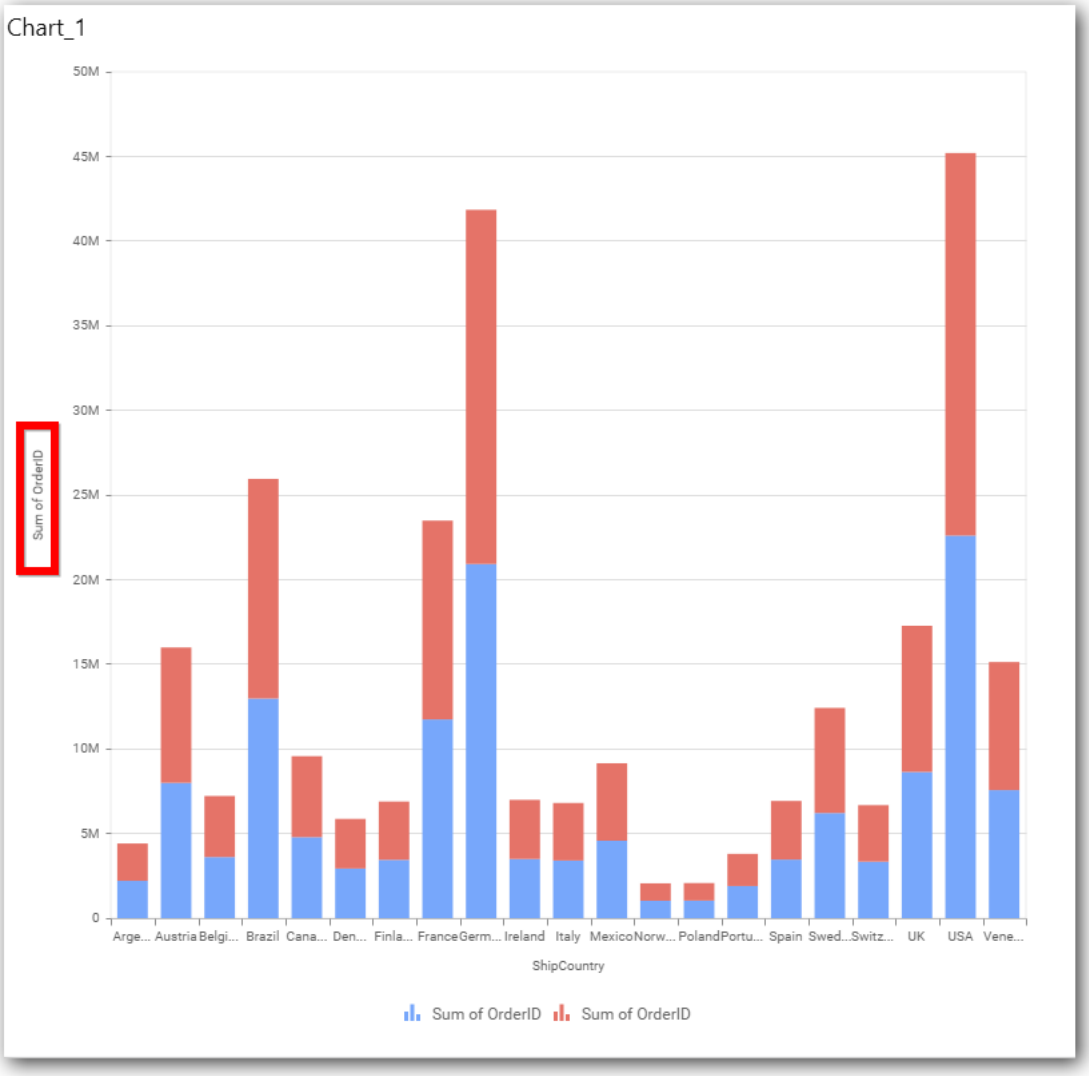
Primary Value Axis

This allows you to enable/edit the Primary Value Axis title. It will reflect in chart area y-axis name.



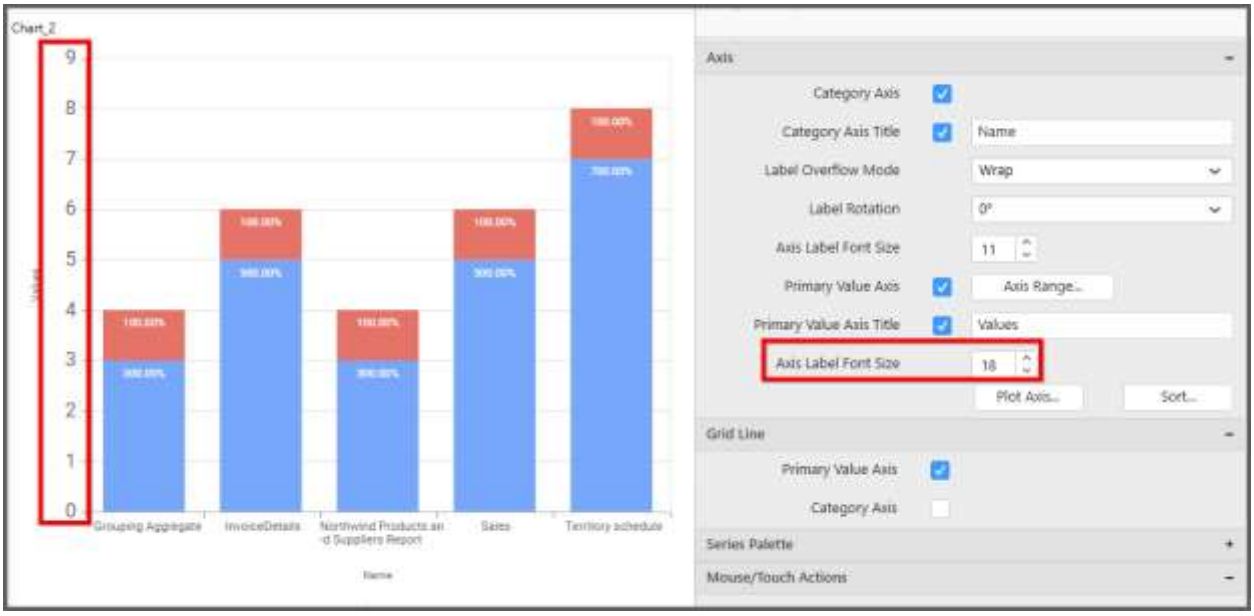
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

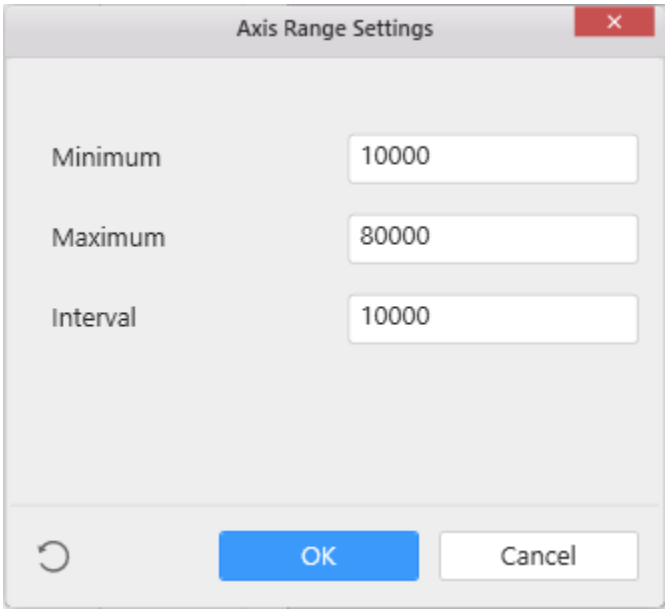


Primary Value Axis Range

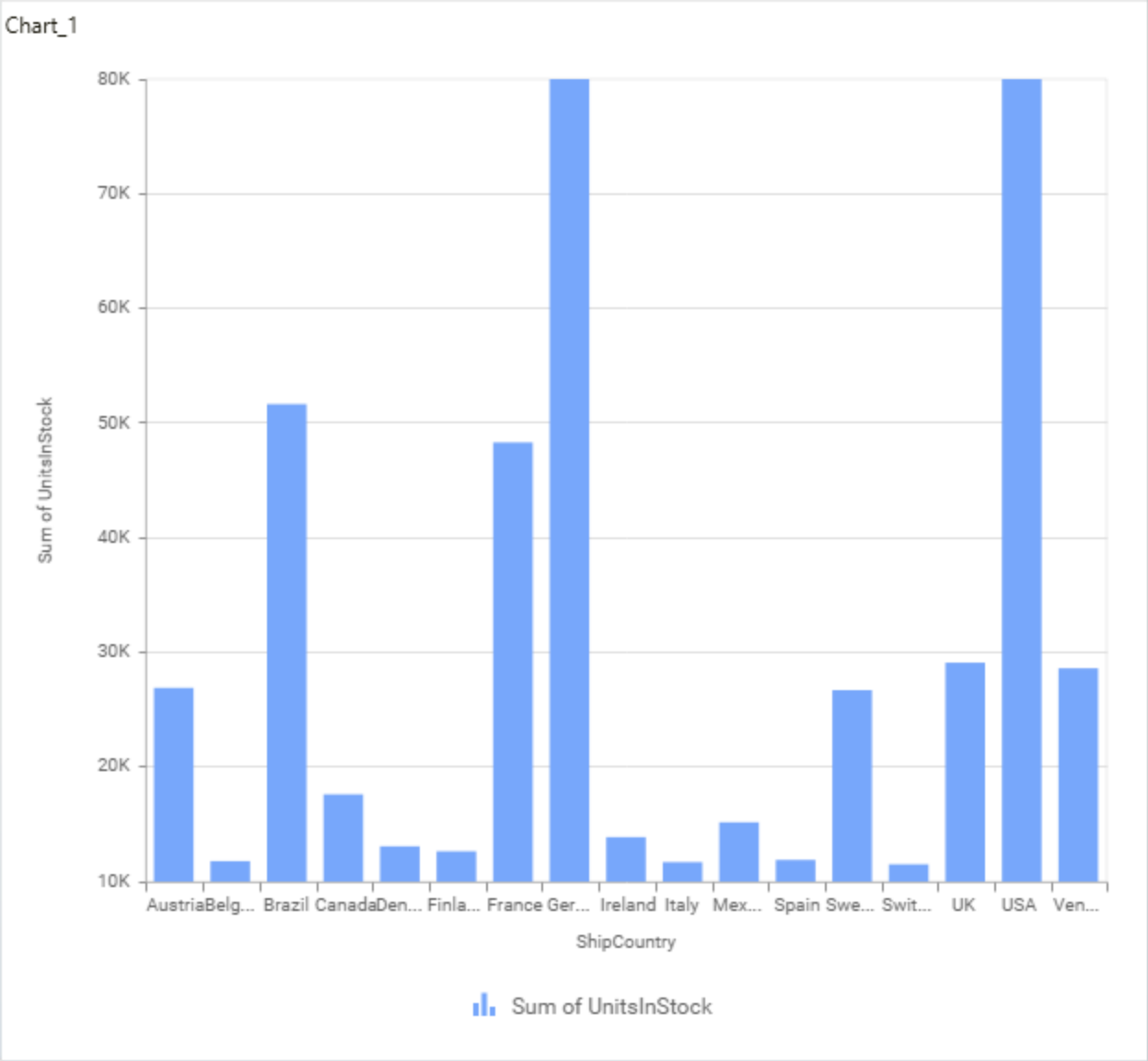
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

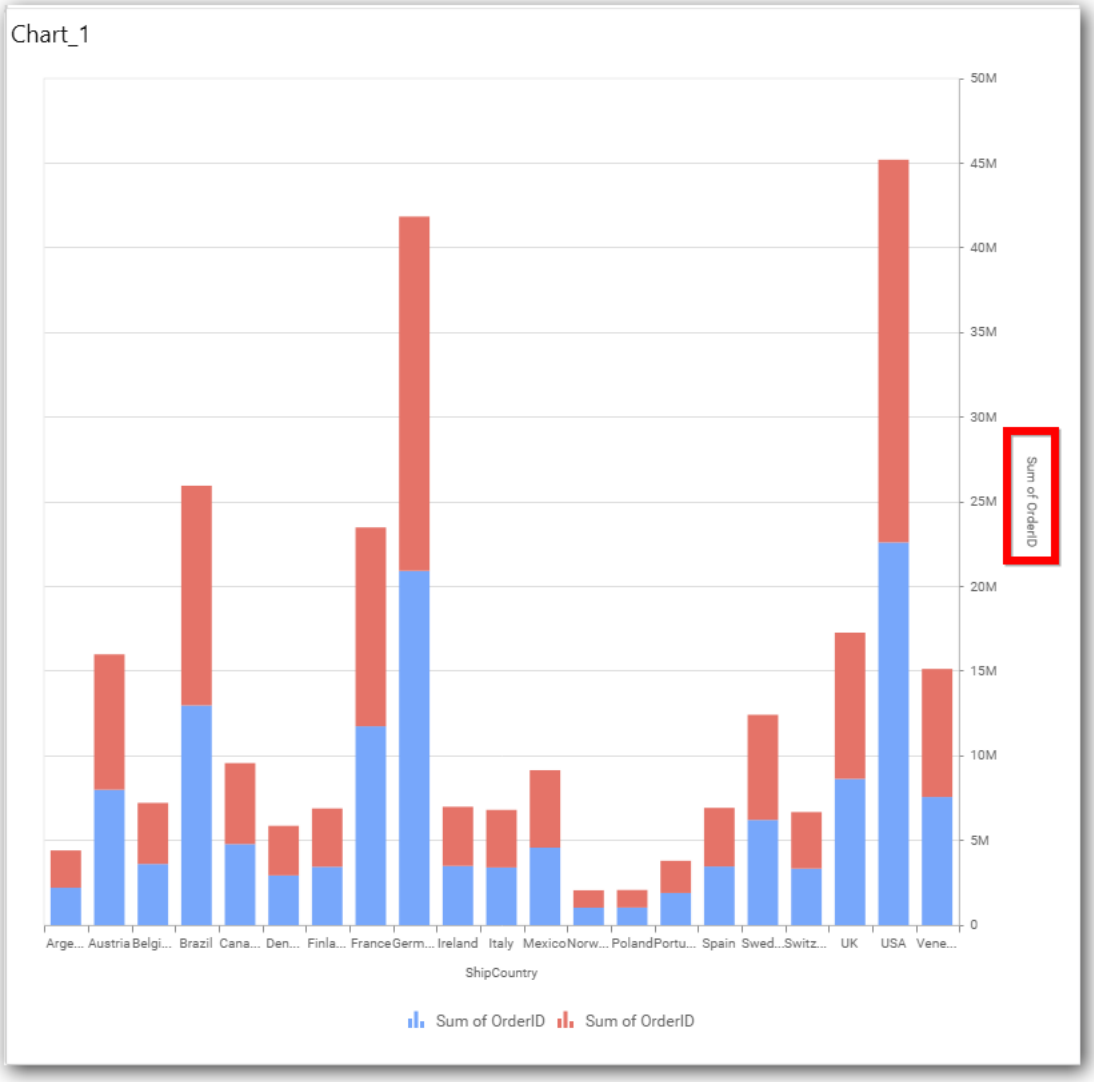


Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.

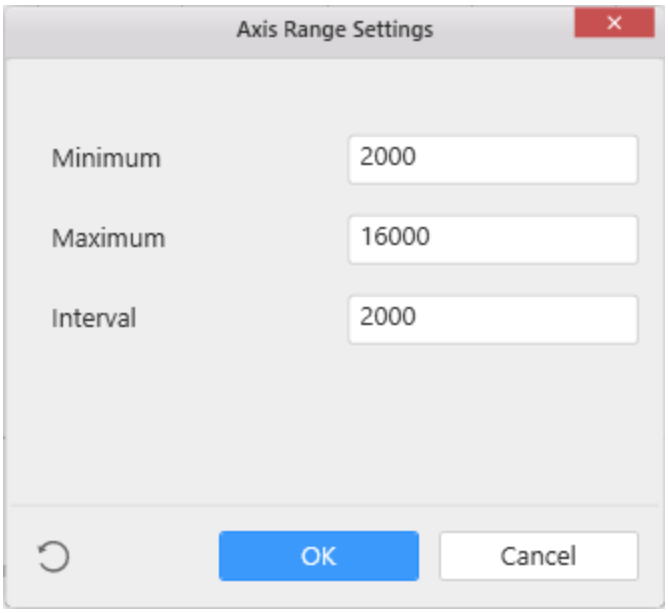
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

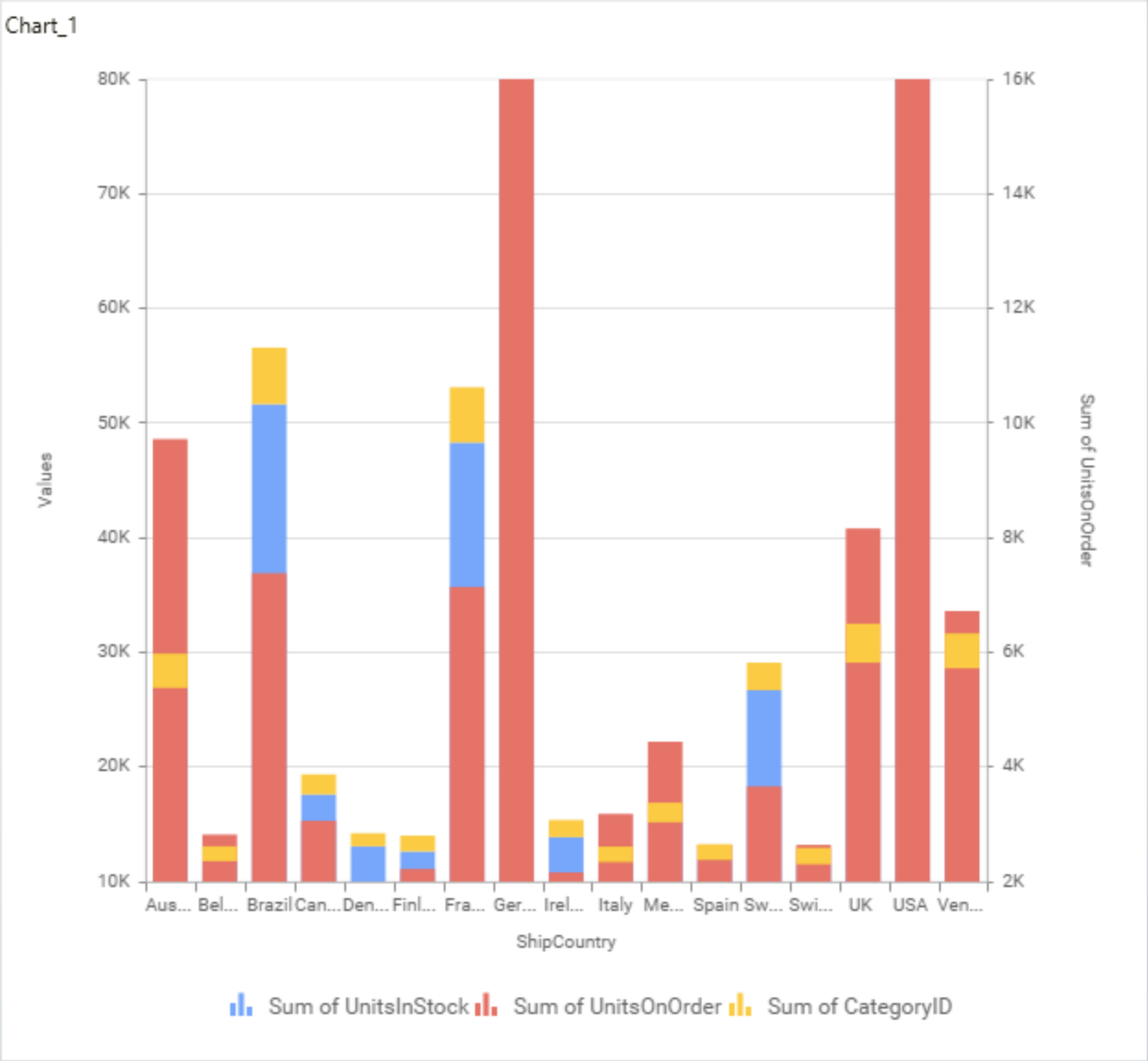


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

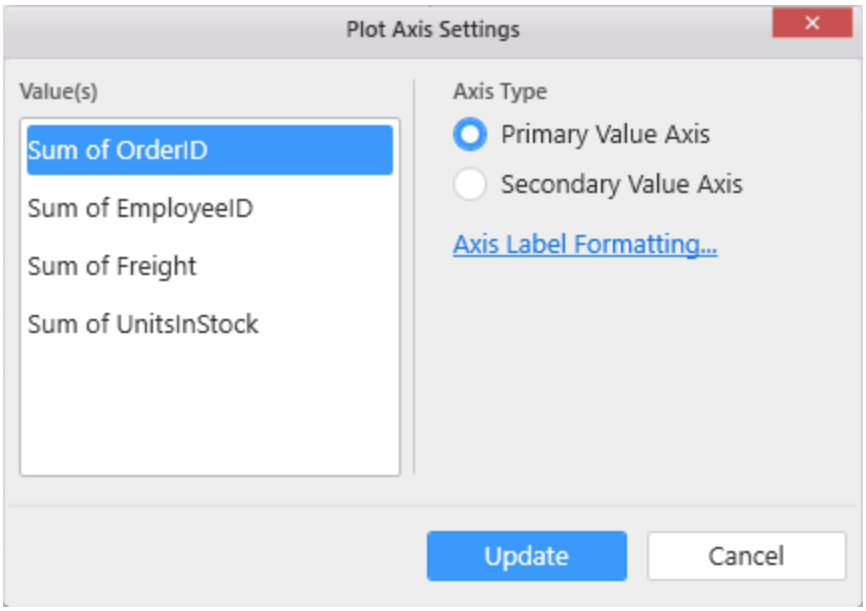


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



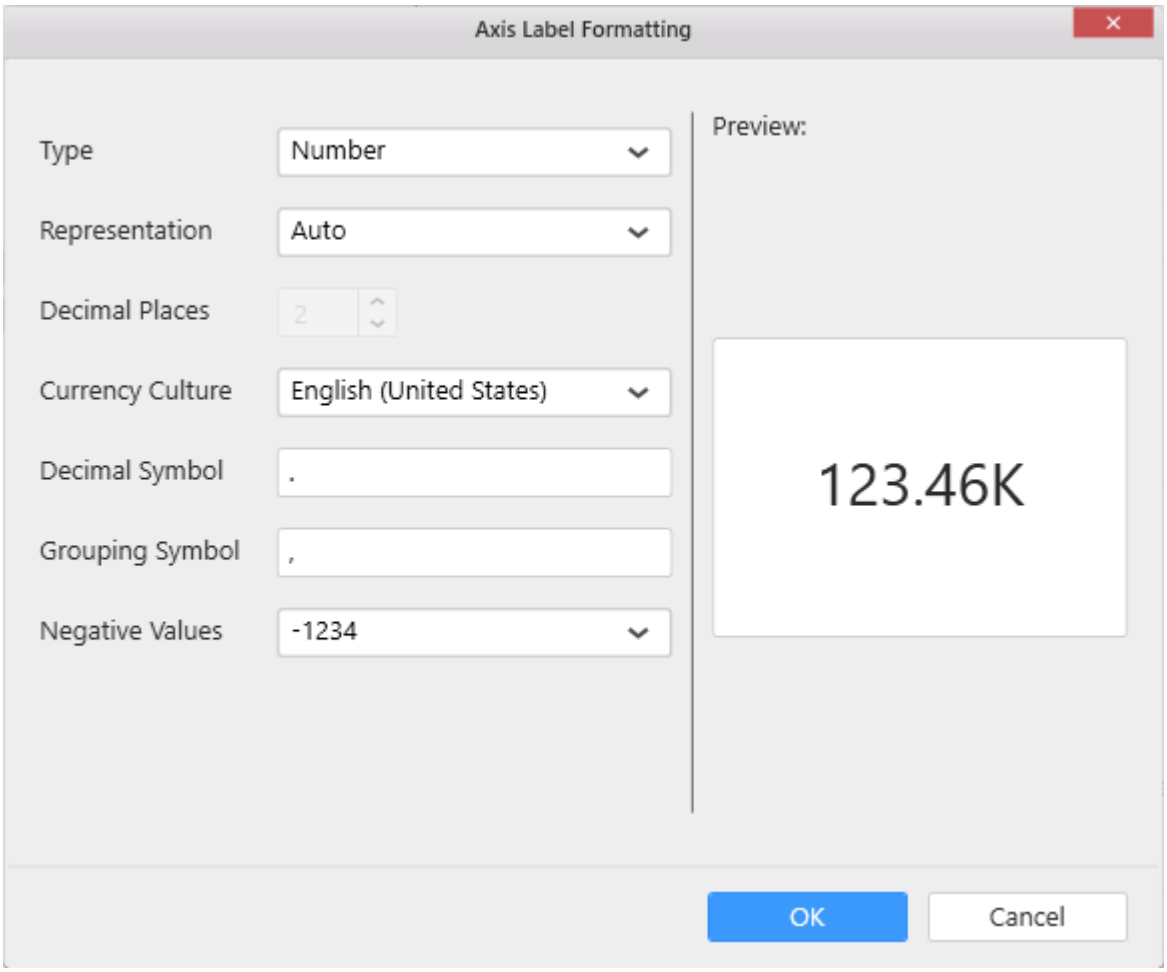
Plot Axis Settings

This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



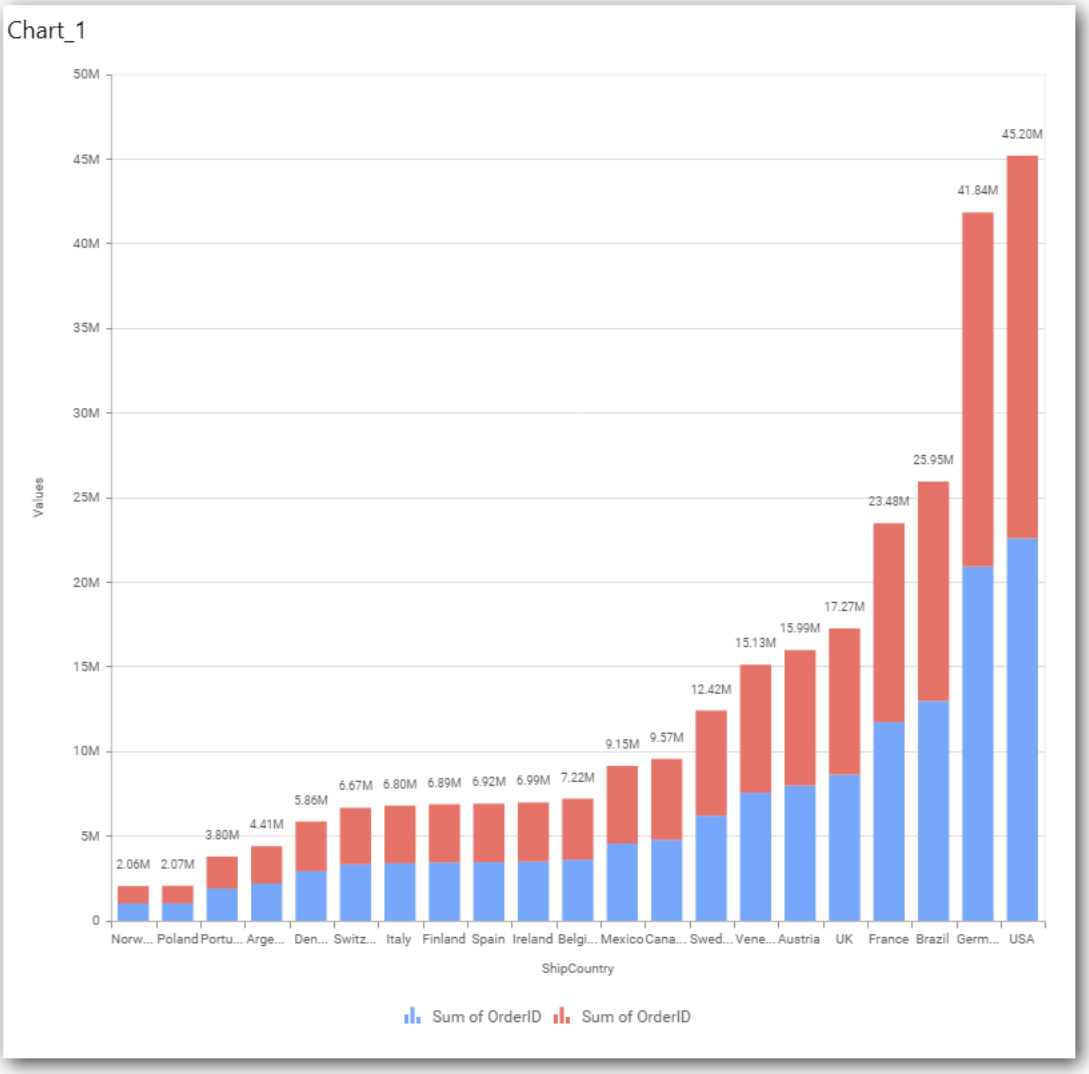
Axis Label Formatting

This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.



Sort Order

To define the sorting of chart based on any of the measures that you dropped and its order through this option. Following screenshot illustrates the Ascending sort order.



Grid Line Settings

Primary Value Axis

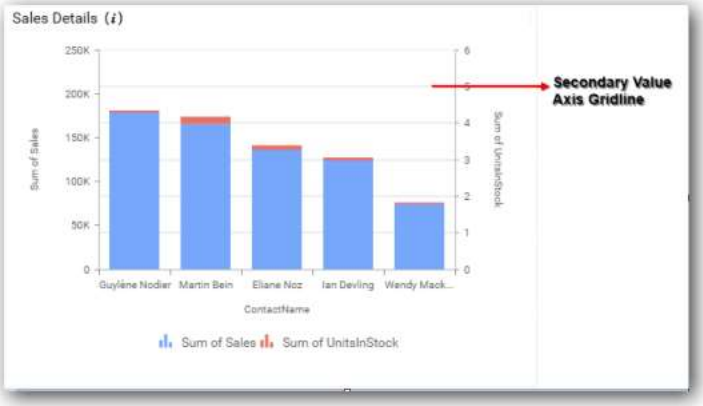
This allow to enable the primary value axis' gridlines for the stacked column chart.

Category Axis

This allows you to define the visibility of Category axis' gridlines.

Secondary Value Axis

This allows you to toggle the visibility of secondary value axis' gridlines.

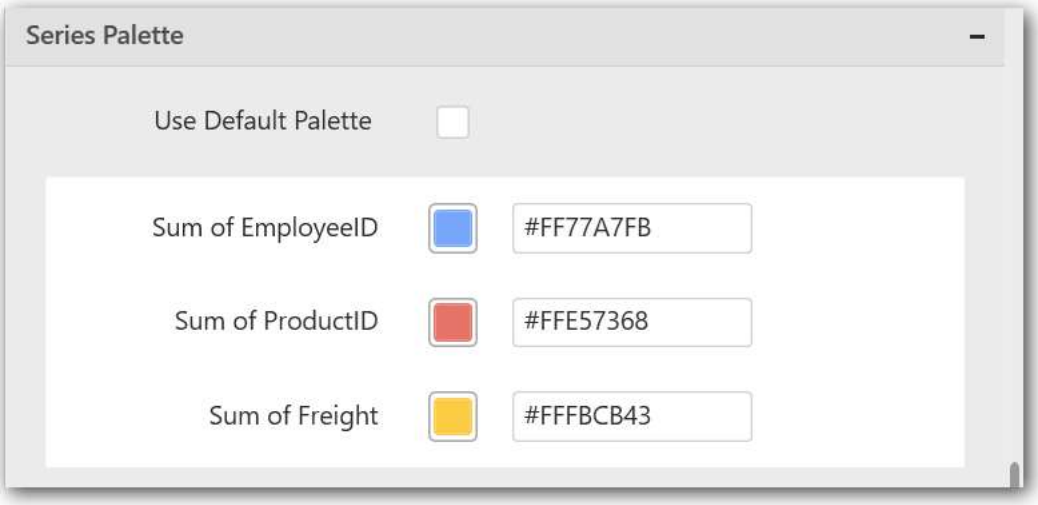


Series Palette

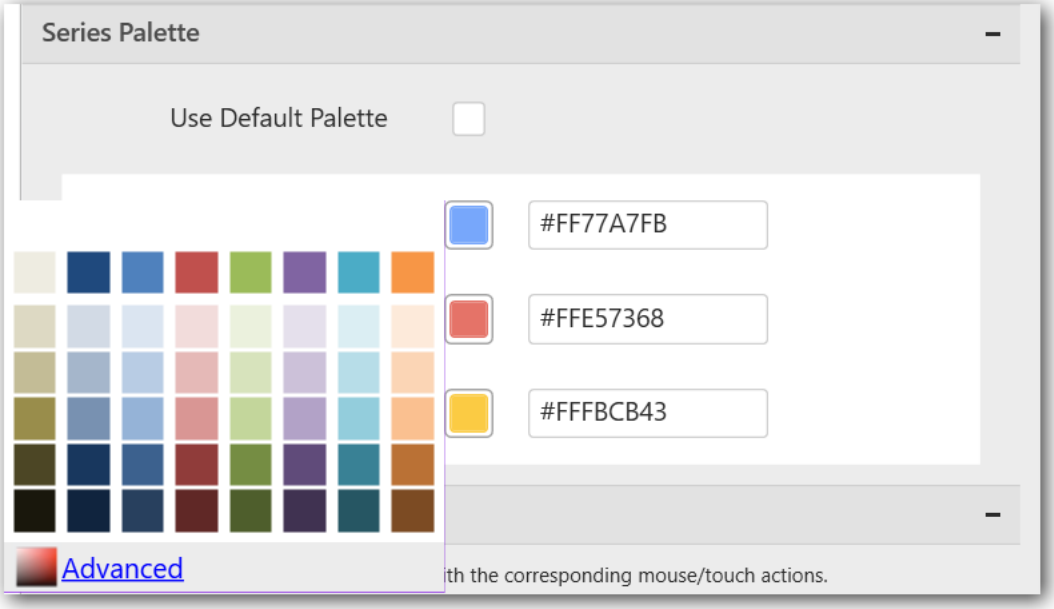
This allows you to customize the chart series color through Series Palette section.

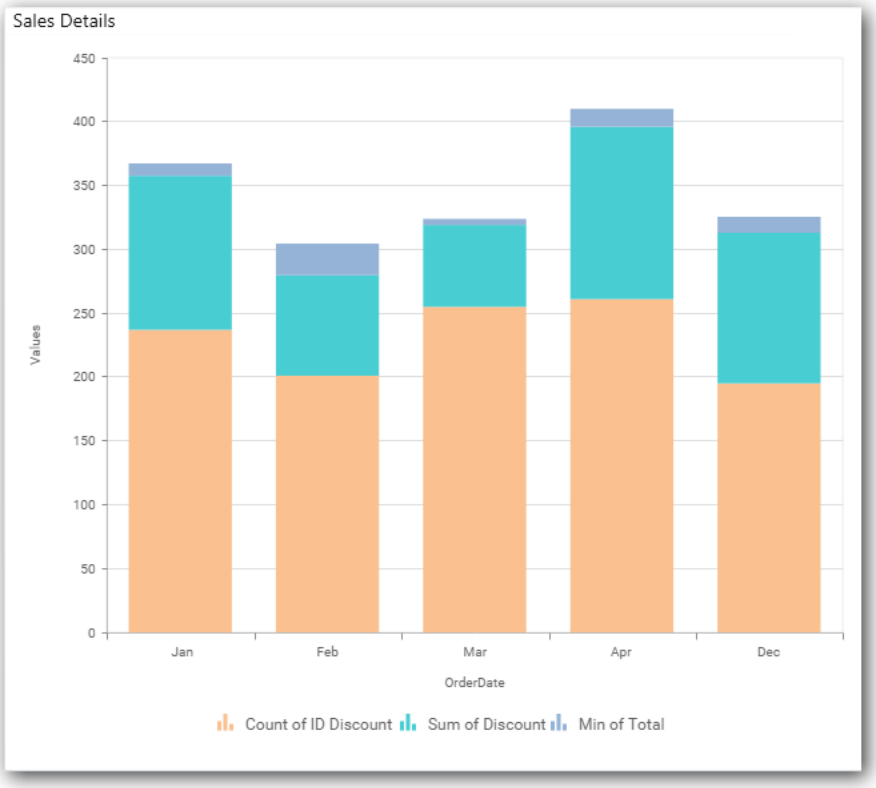
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



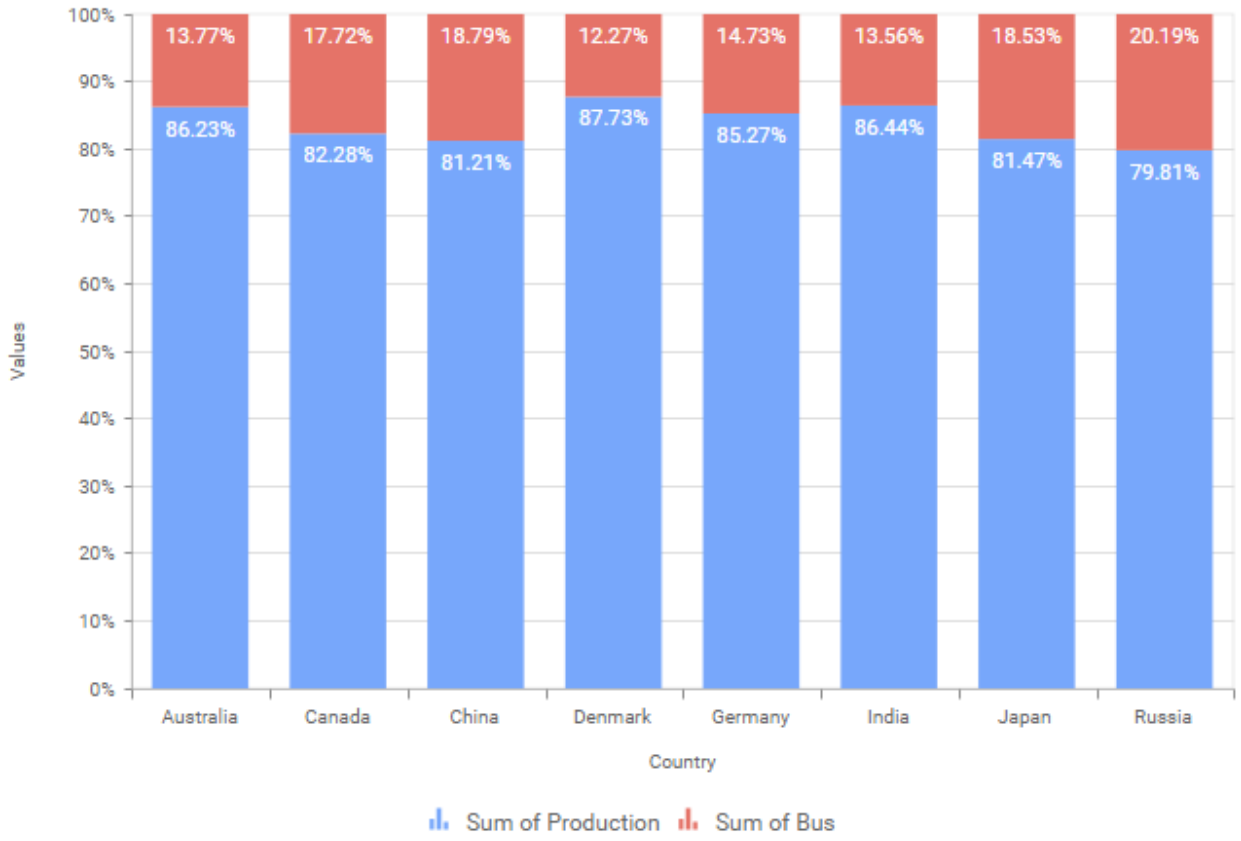
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





100% Stacked Column Chart

100% Stacked Column Chart allows you to compare multiple measures through bars stacked one after the other vertically.

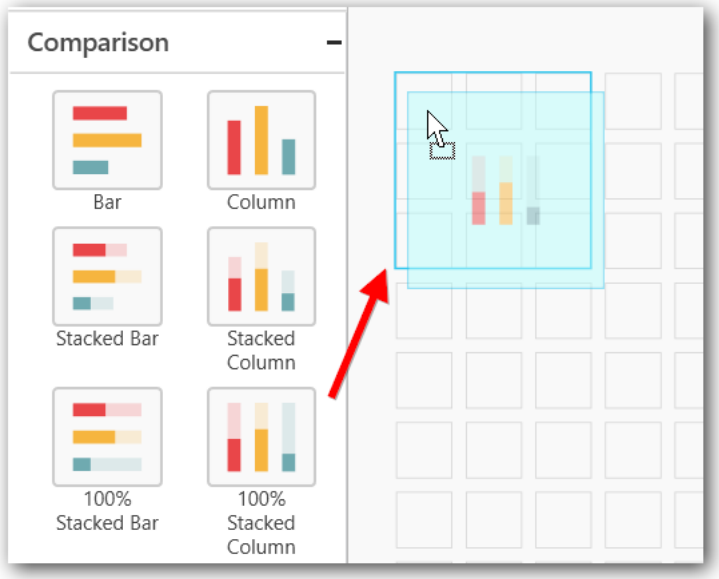


How to configure the flat table data to 100% Stacked Column Chart?

100% Stacked Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Follow the steps configure data to 100% stacked column chart

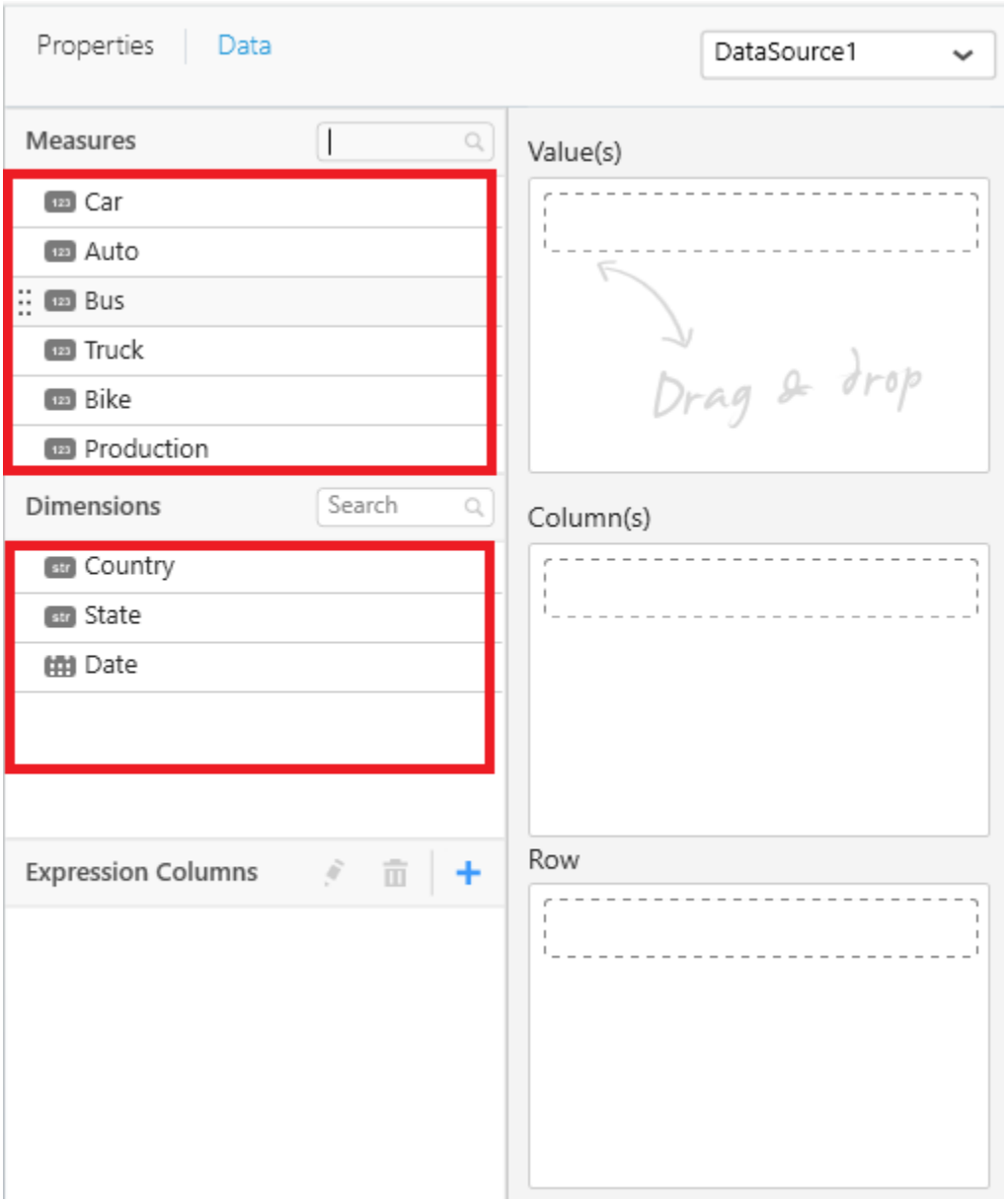
Drag and drop the 100% stacked column chart to canvas and resize it to your required size.



Connect to the data source.

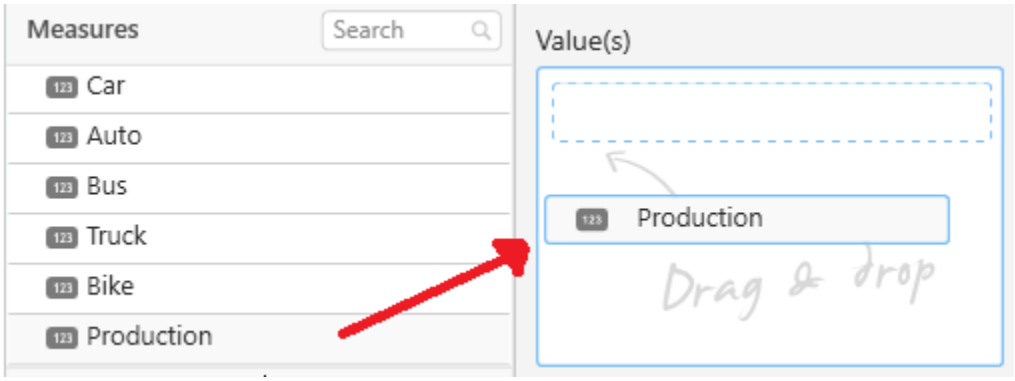
Focus on the 100% stacked column chart and click on Assign Data button

The data pane will be opened with available Measures and Columns from the connected data source.

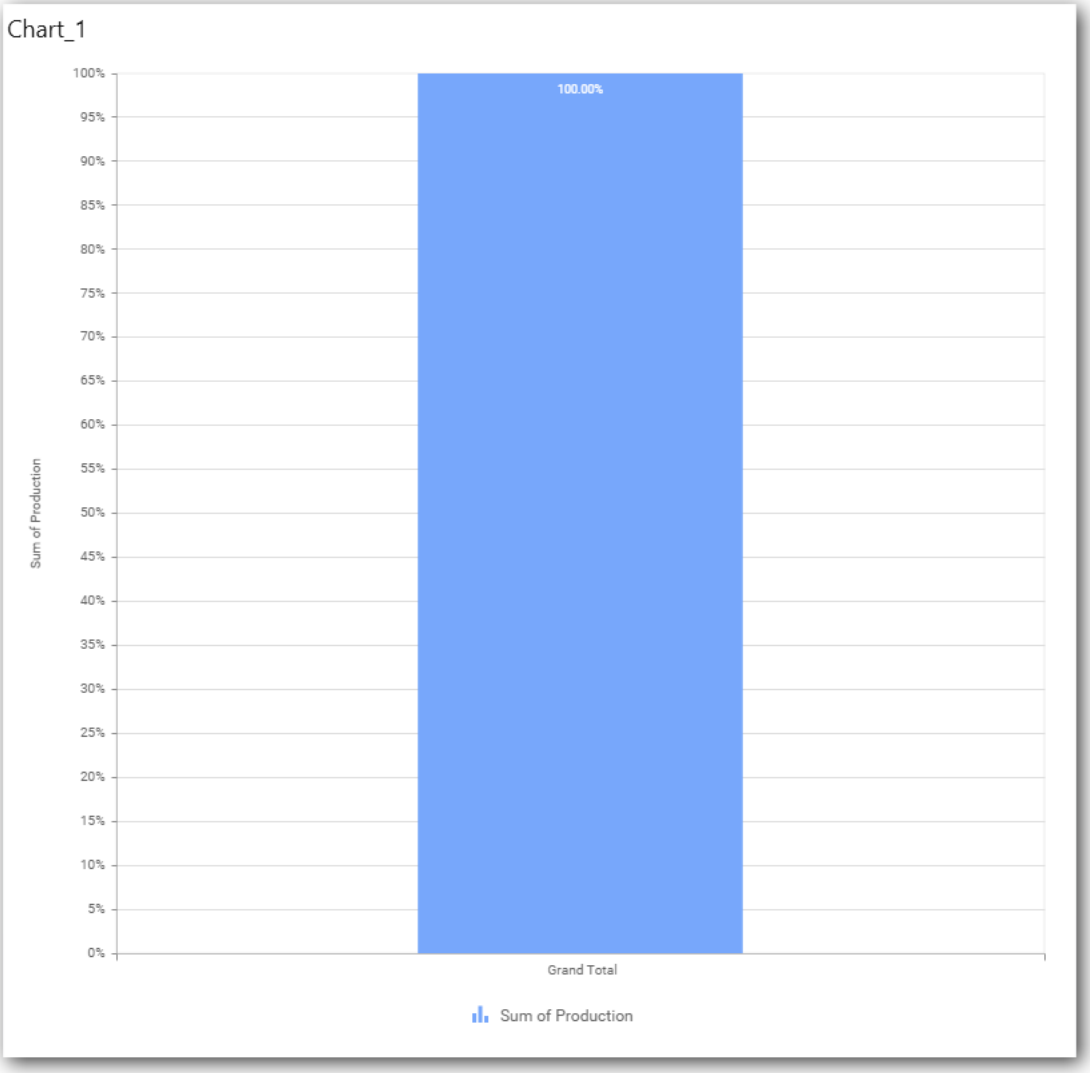


Assigning Value(s)

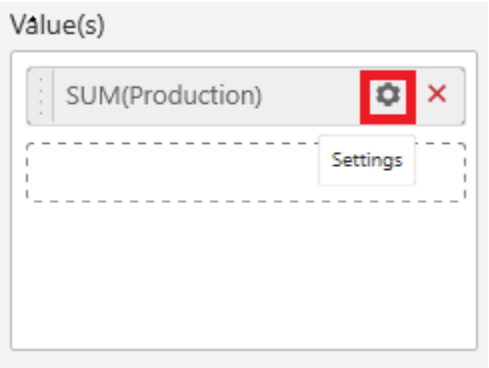
Drag and drop the Measure into Value.



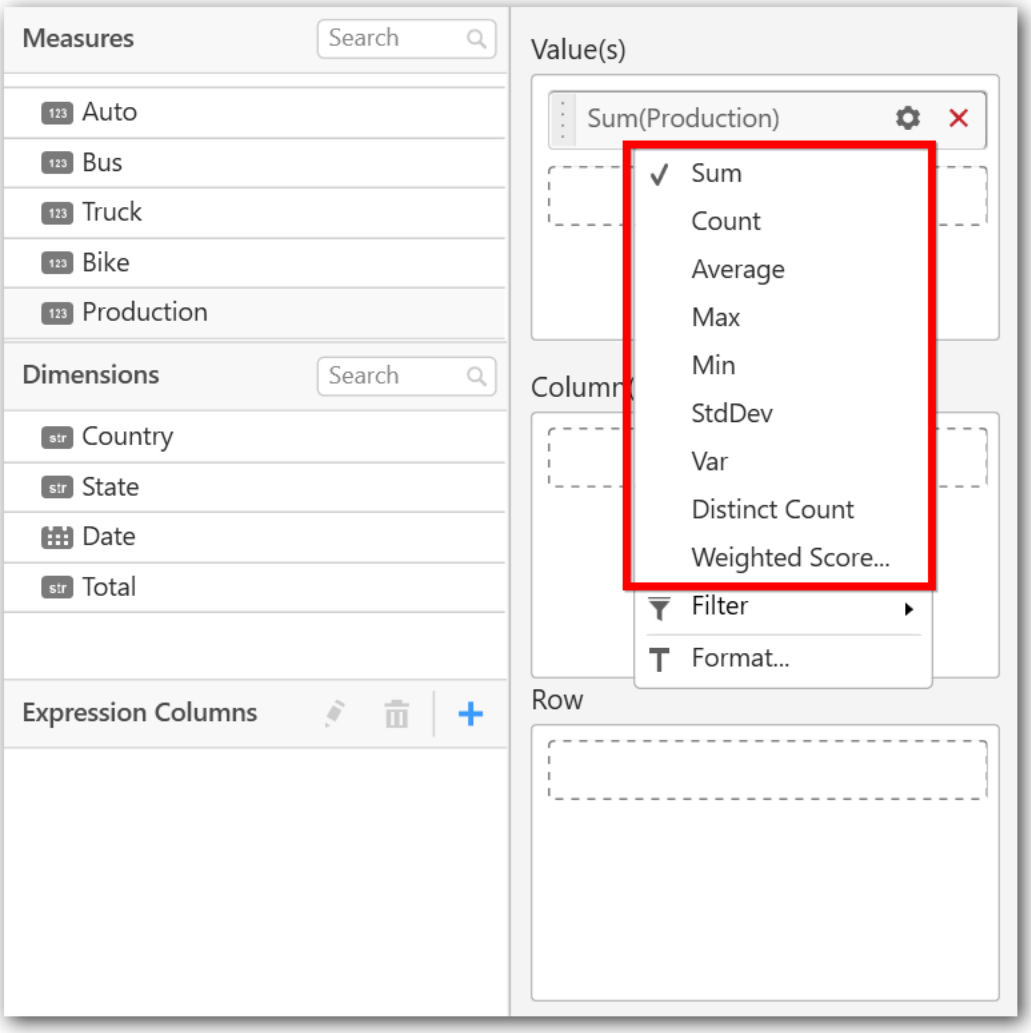
Now the chart will be rendered like this.



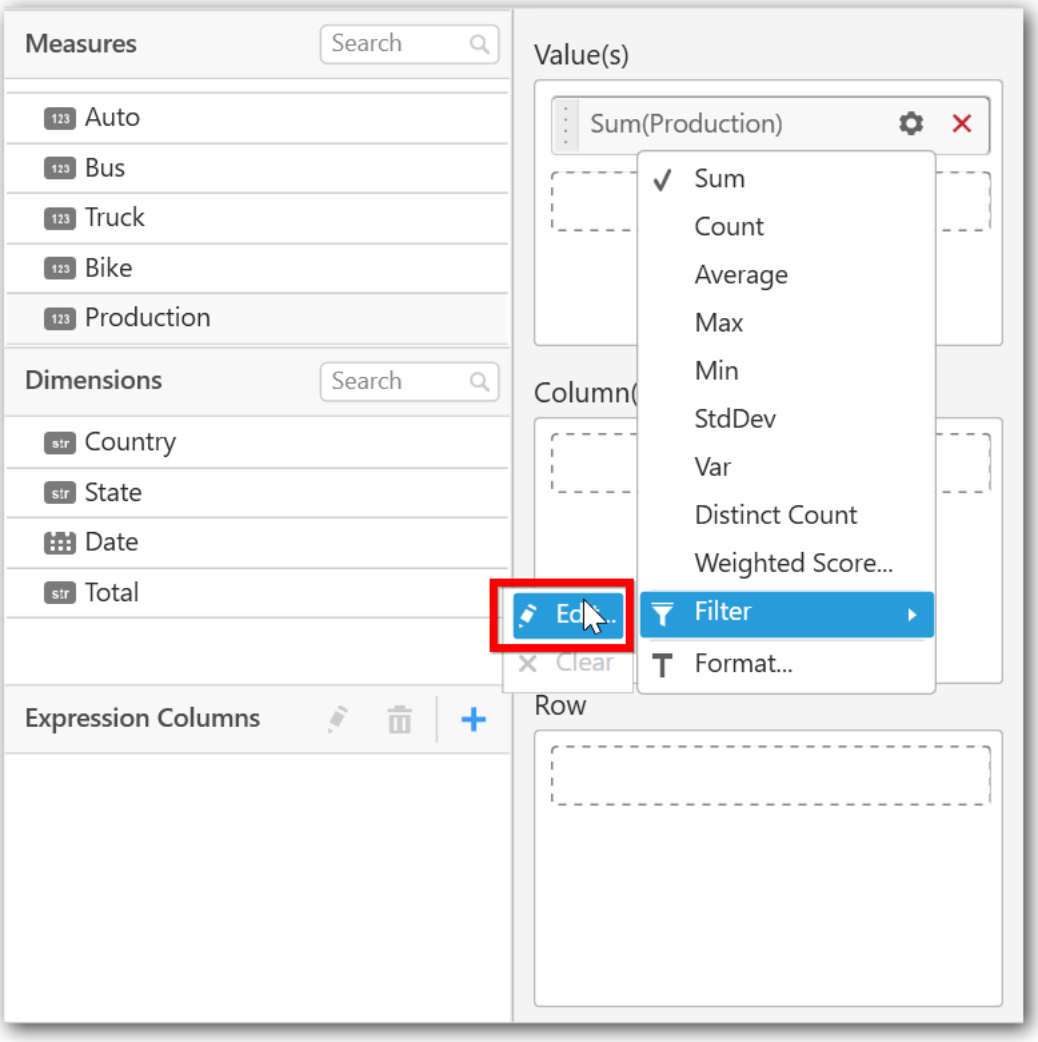
You can change the summary type of the value by clicking on Settings option.



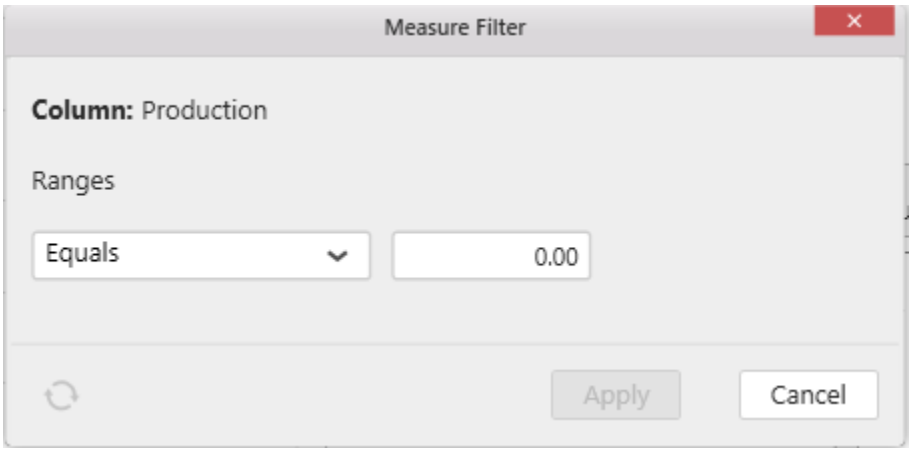
Select the required summary type from list.

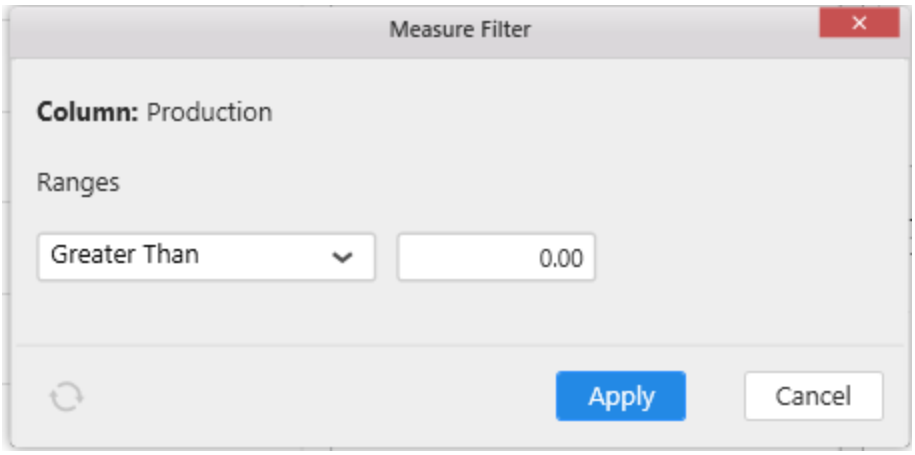


You can select what data to be displayed by choosing filter option.

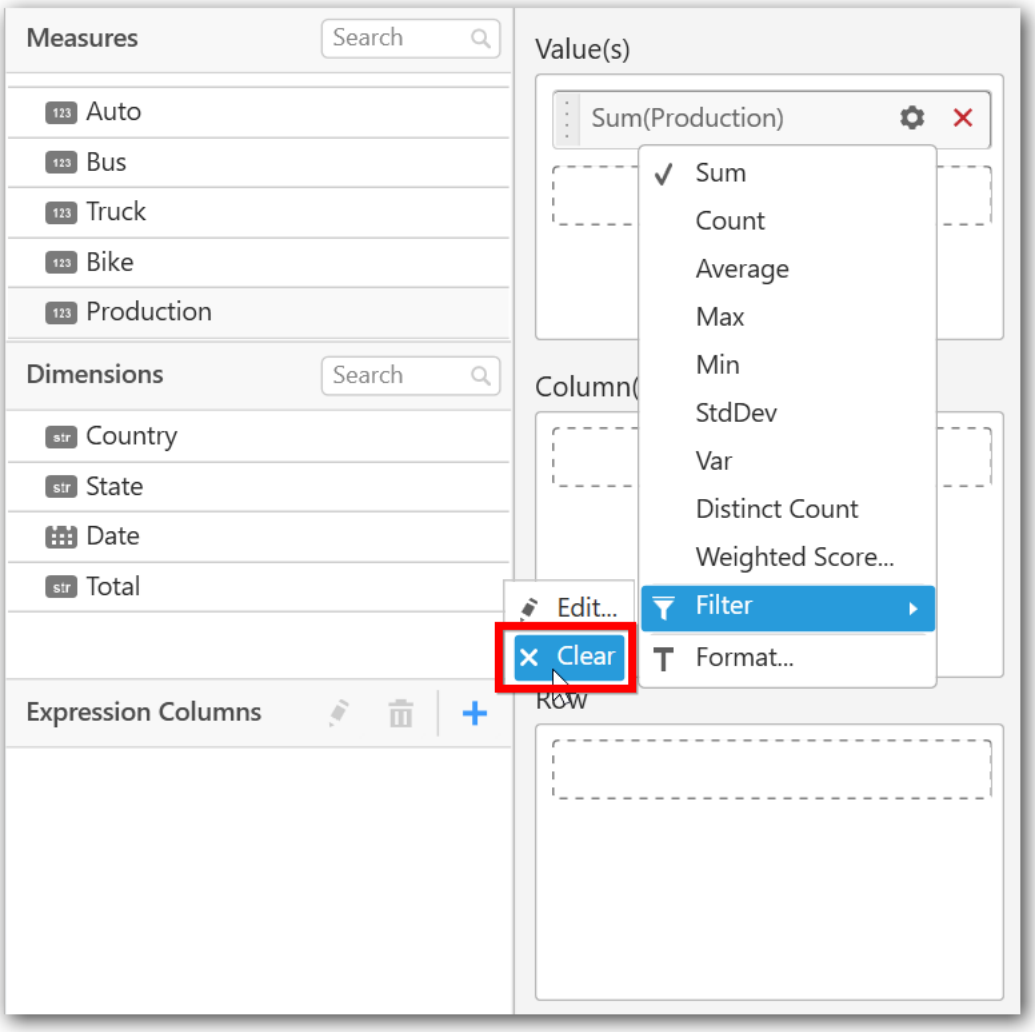


The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.

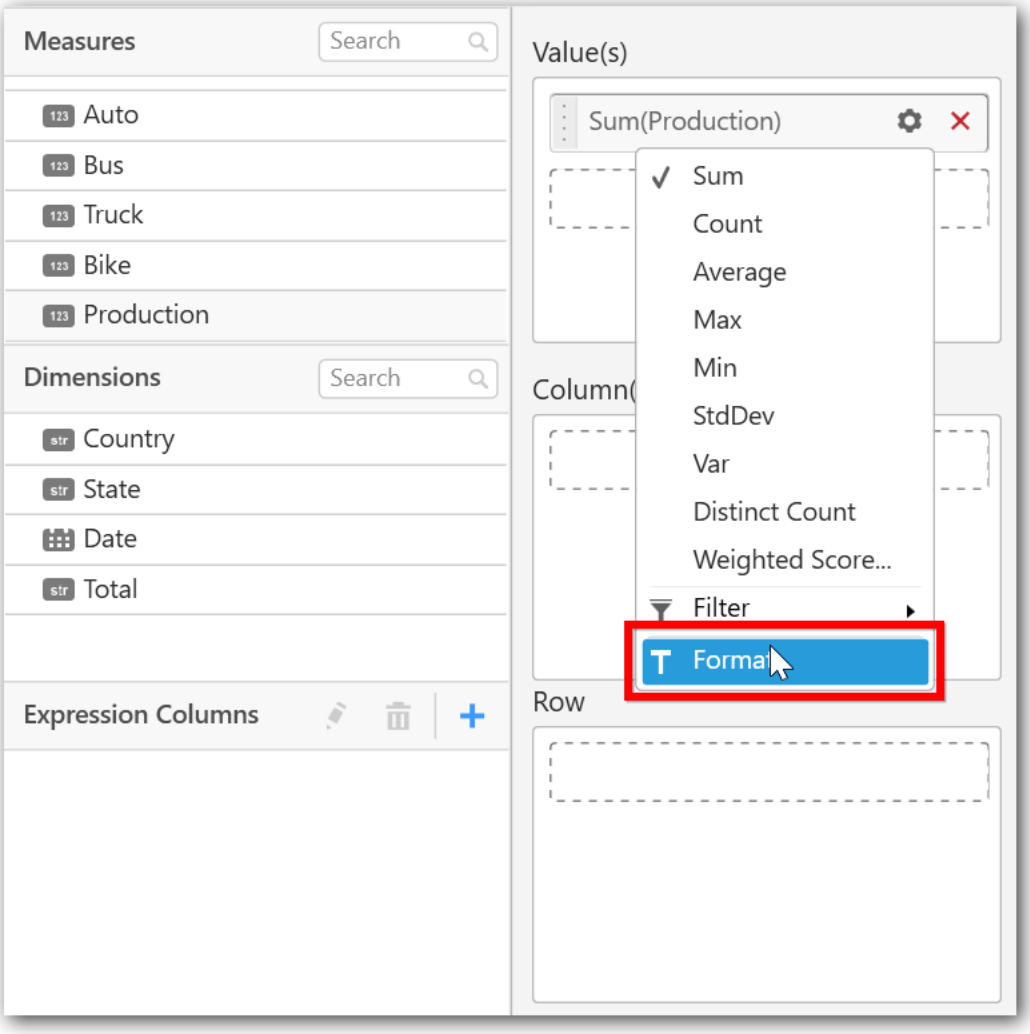




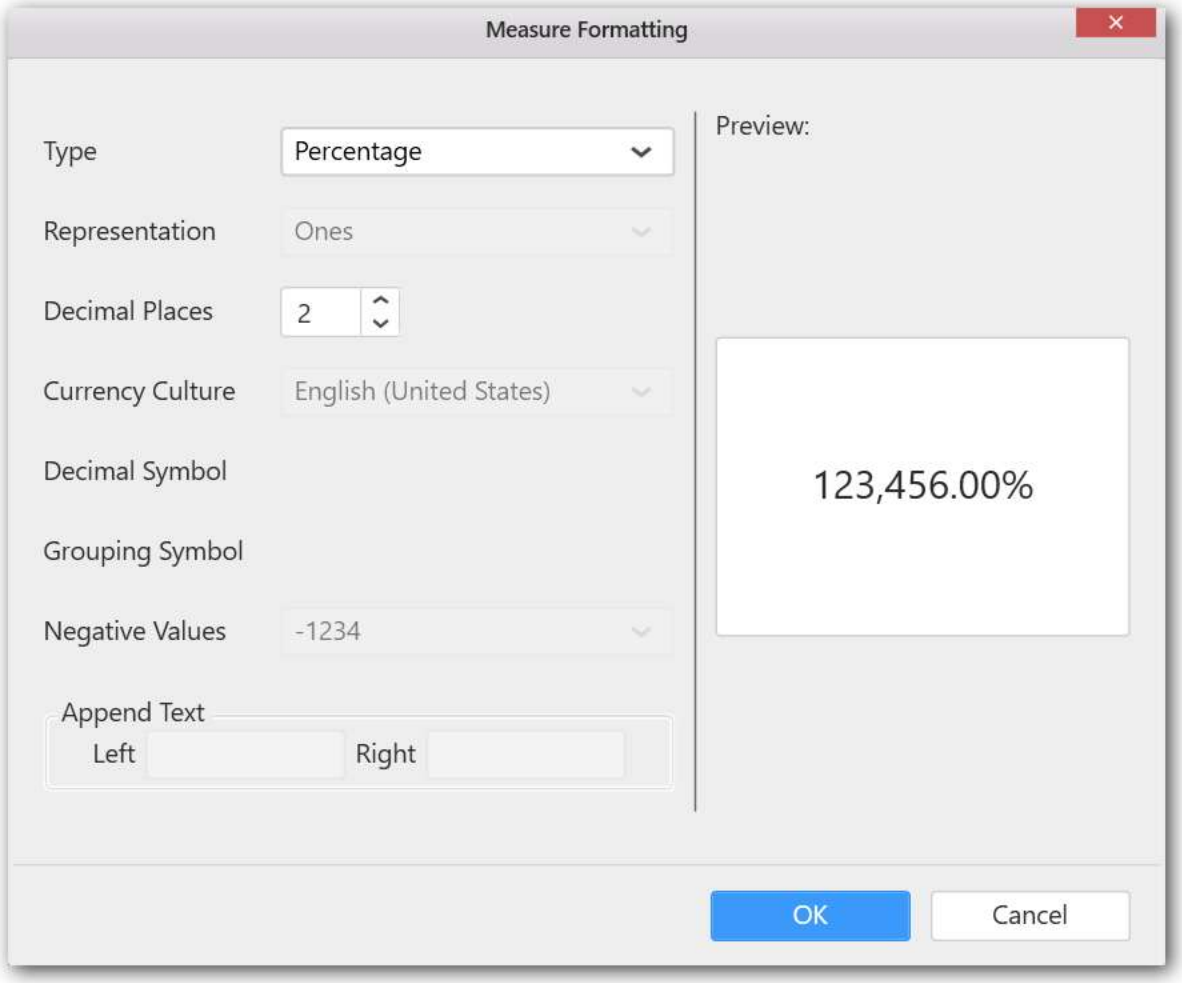
You can clear the filter.



You can Format the value.

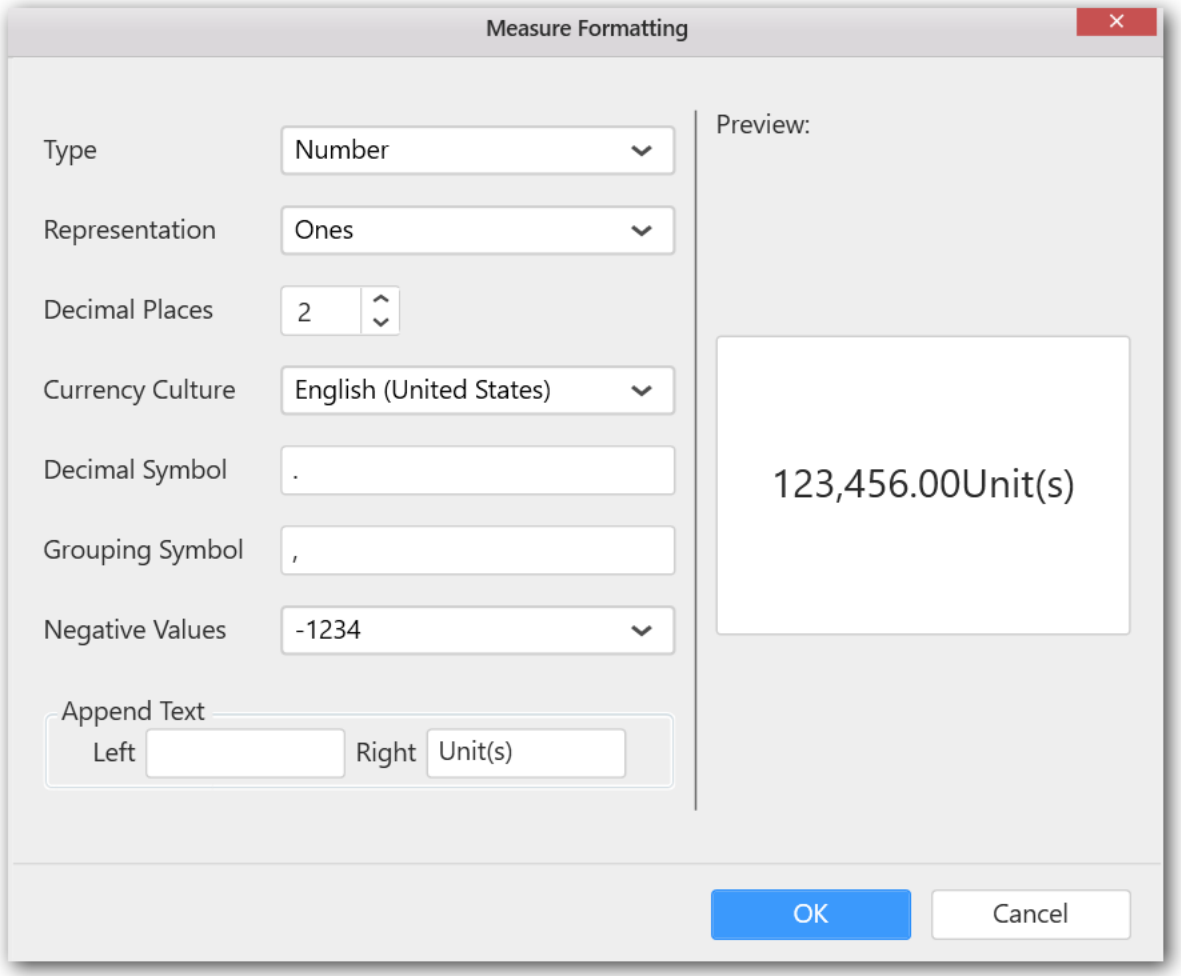


The format options will be shown.

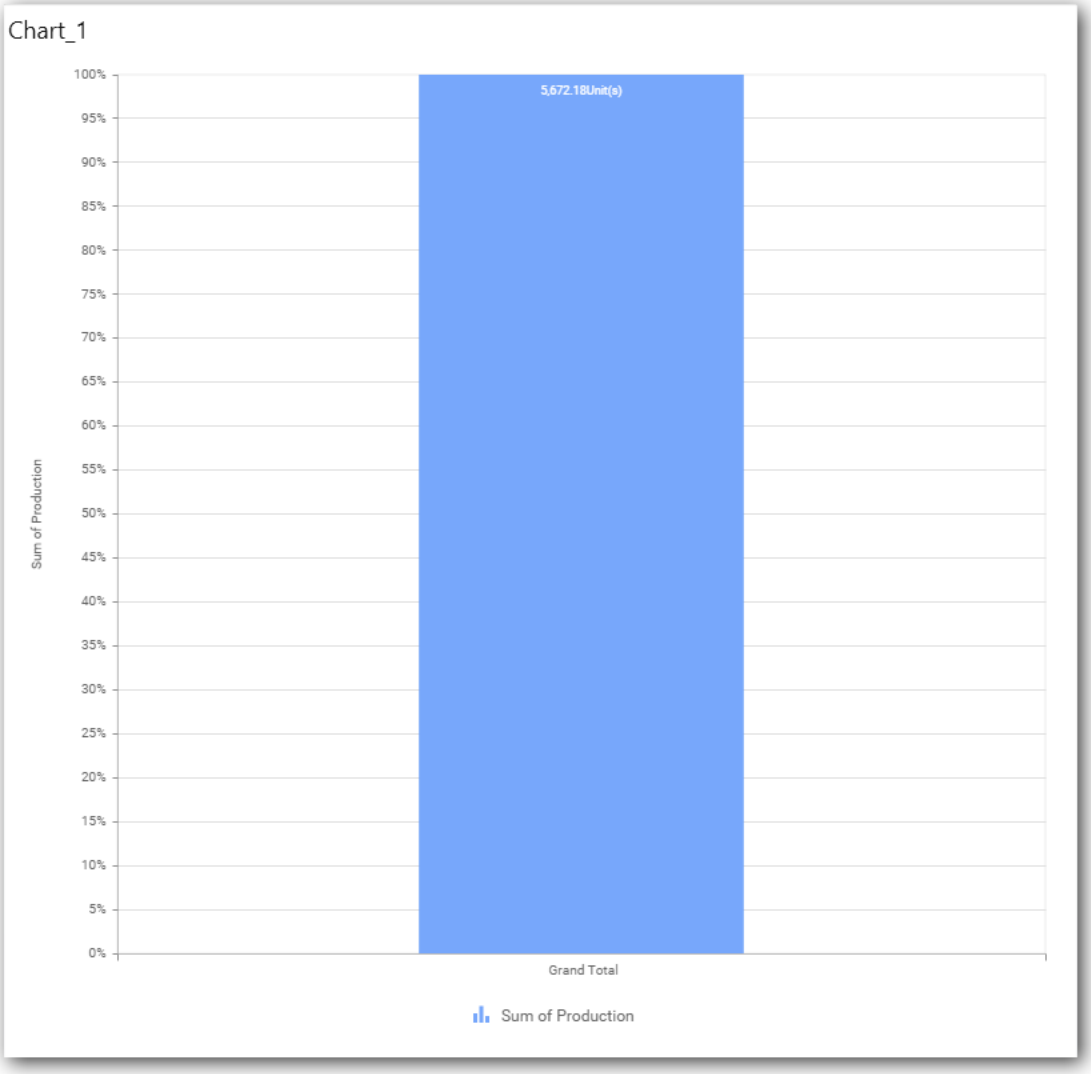


Note: For 100% Stacked Charts, default format type is Percentage.

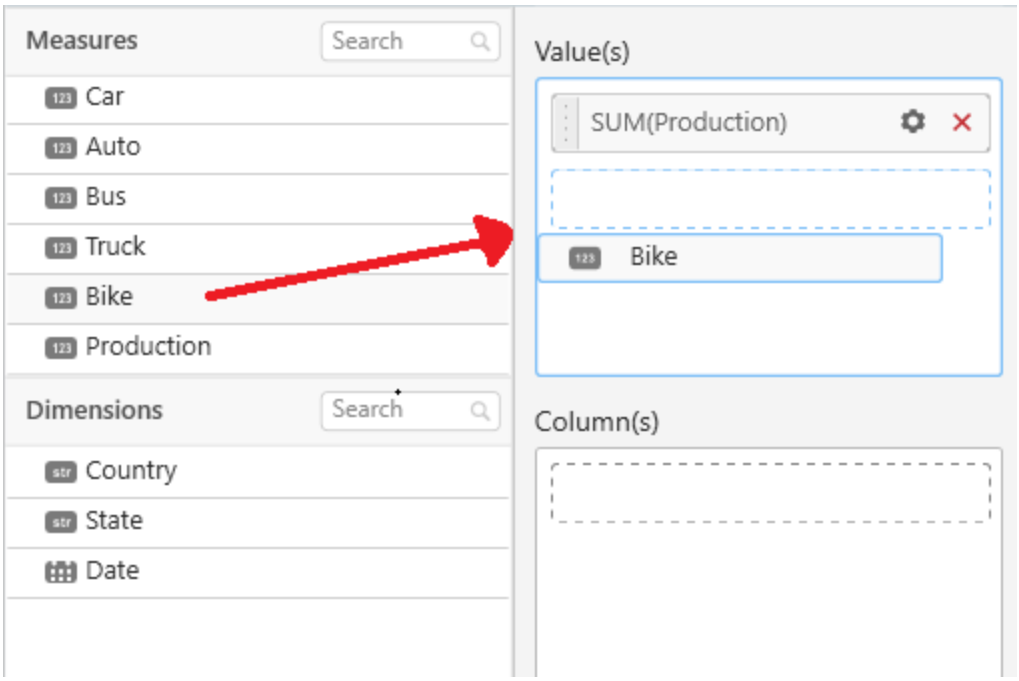
Choose the options you need and click OK.

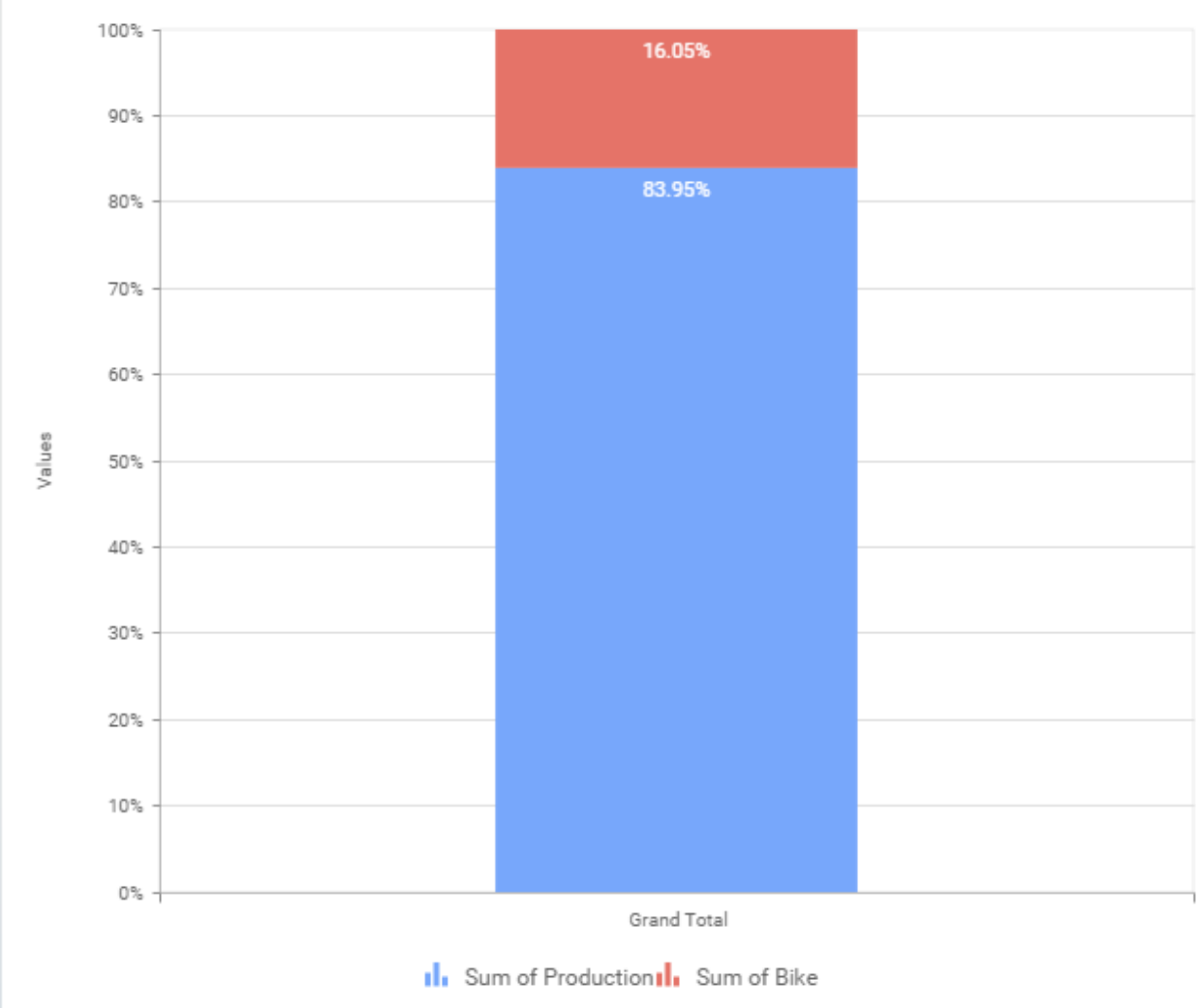


Now the Chart will be rendered like this.



You can add more number values by drag drop the Measure into Value field.

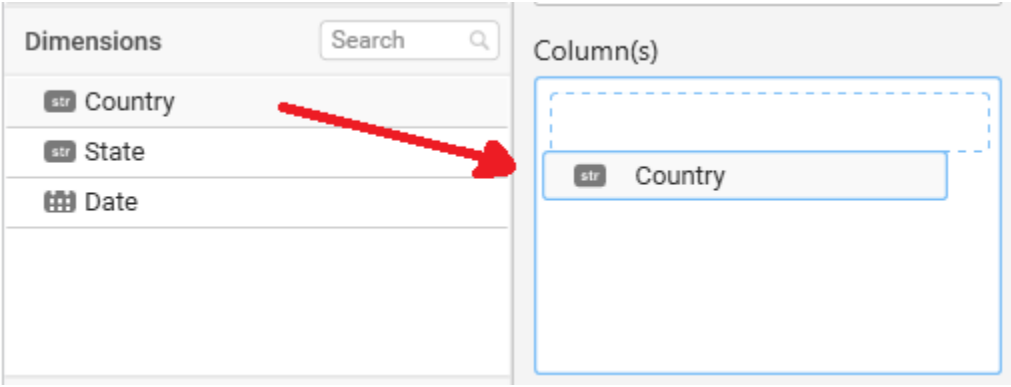


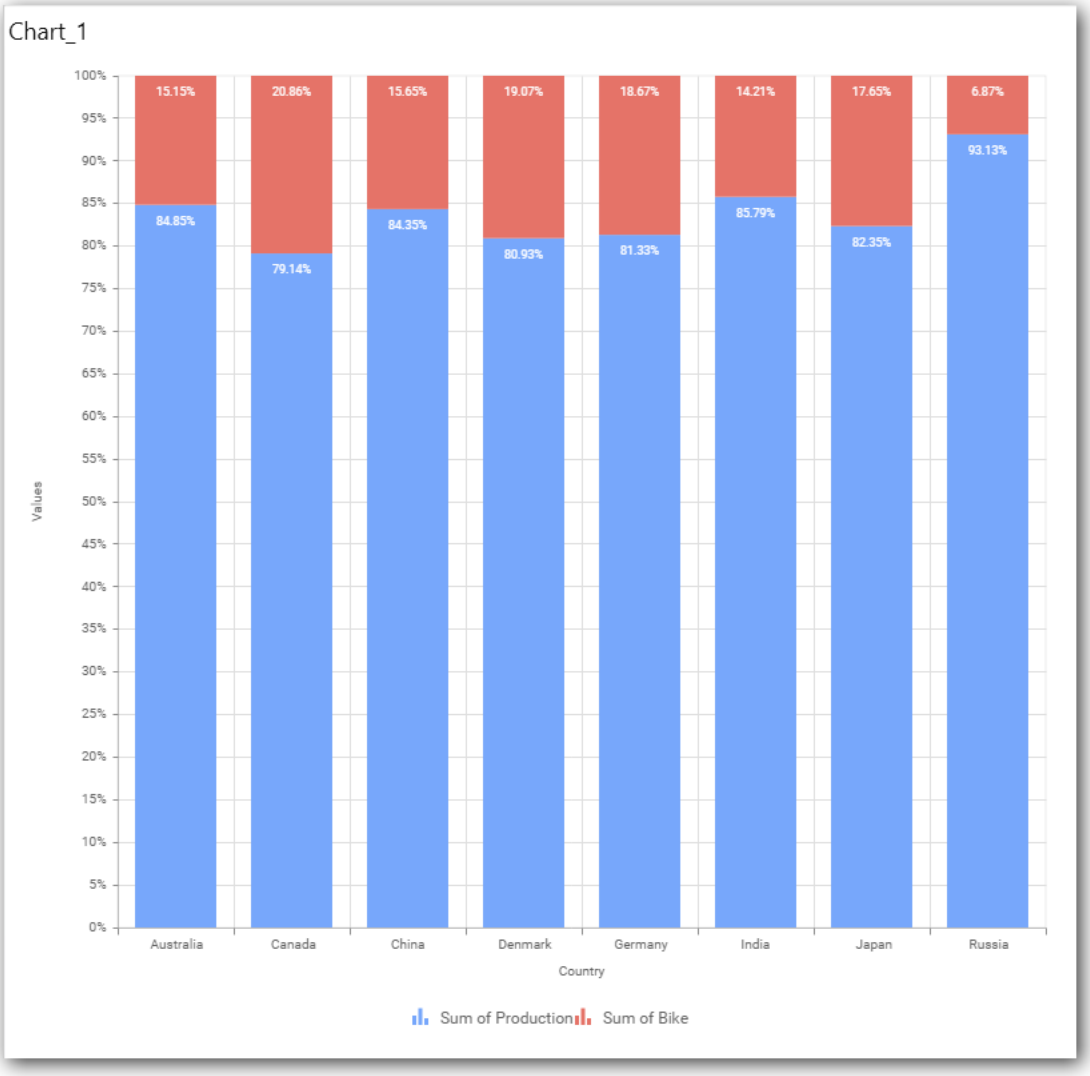


You can also add Dimensions and Columns to Value(s).

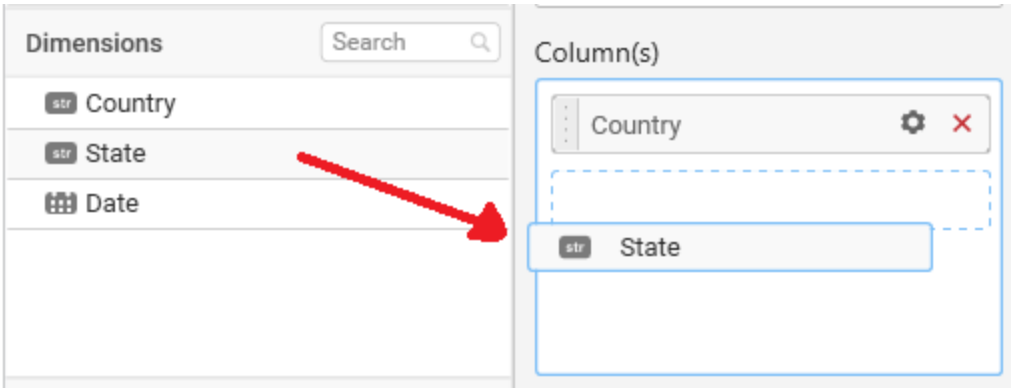
Assigning Column(s)

You can add the Dimension into Column field by drag and drop.

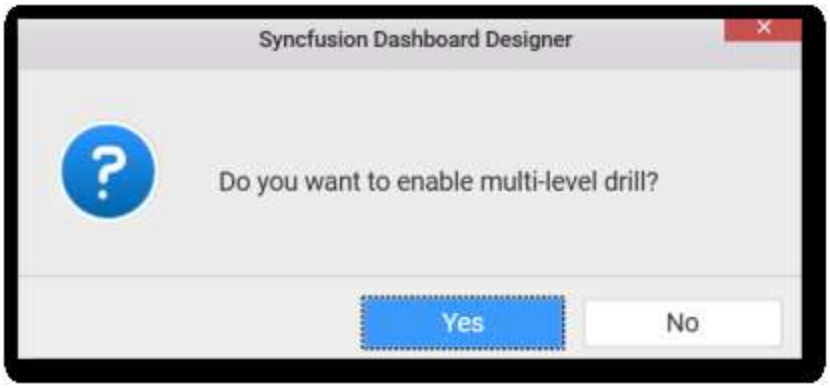




You have option to add more than one Column to Value.

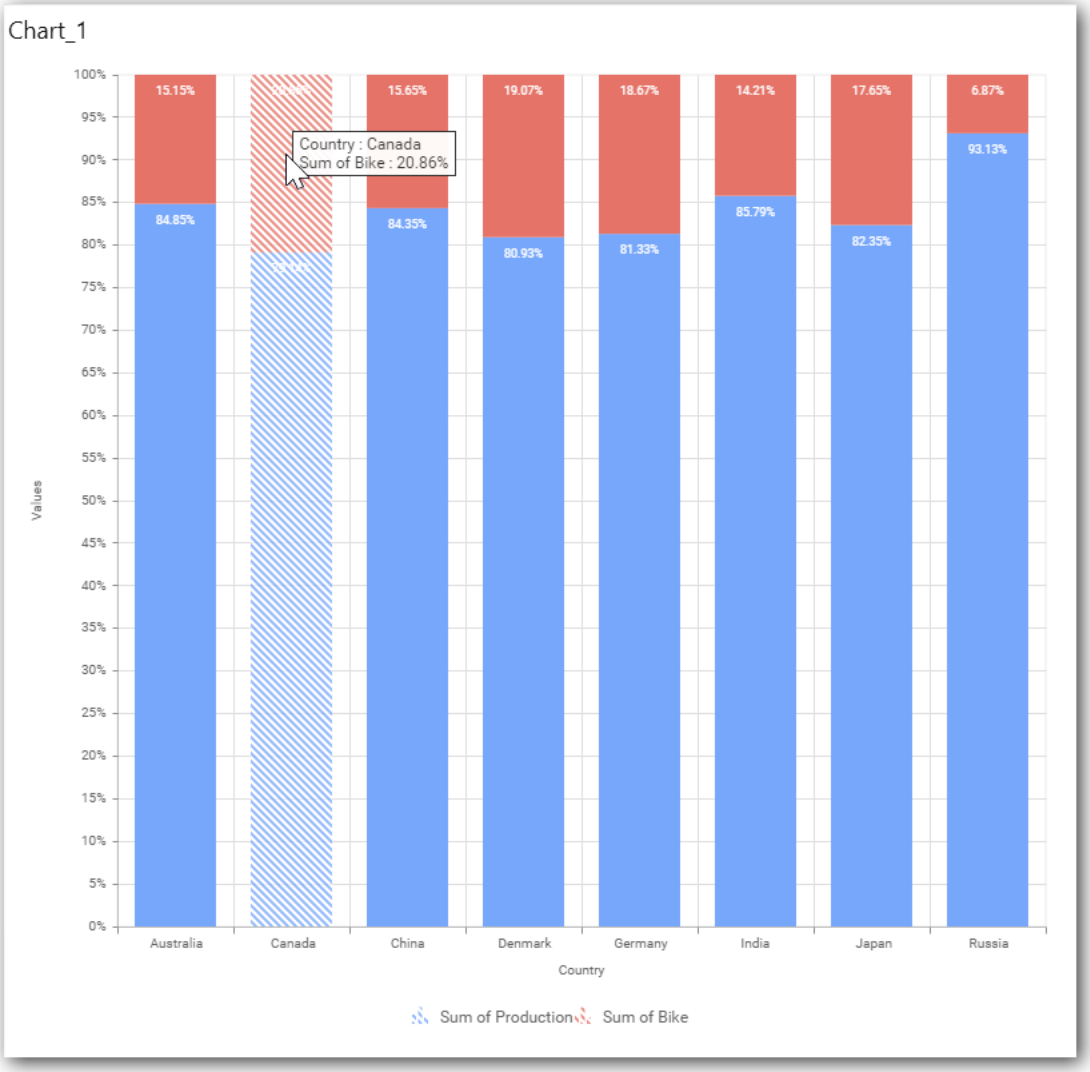


The following alert message will be shown.

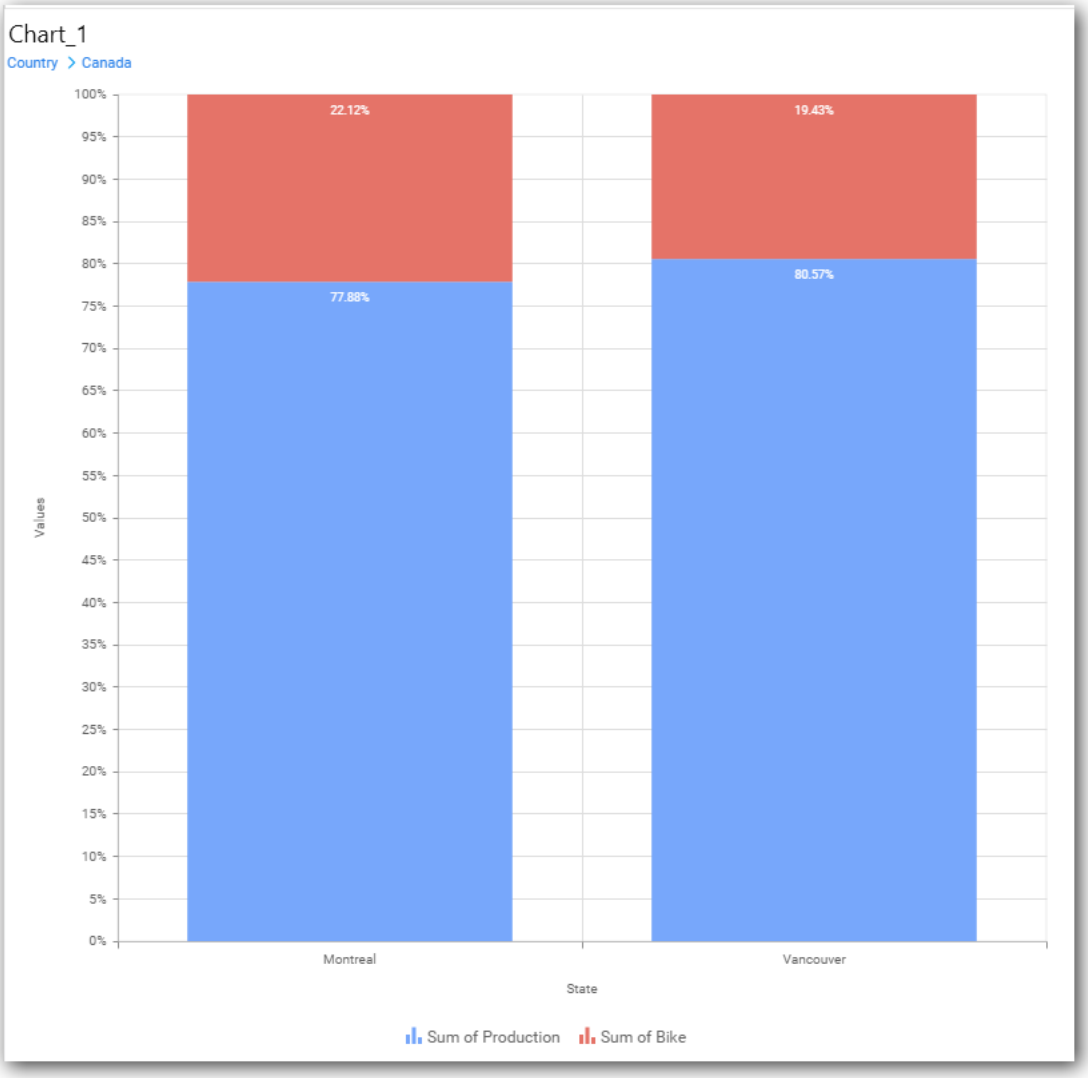


If you choose **Yes** Drill down option will be enabled.

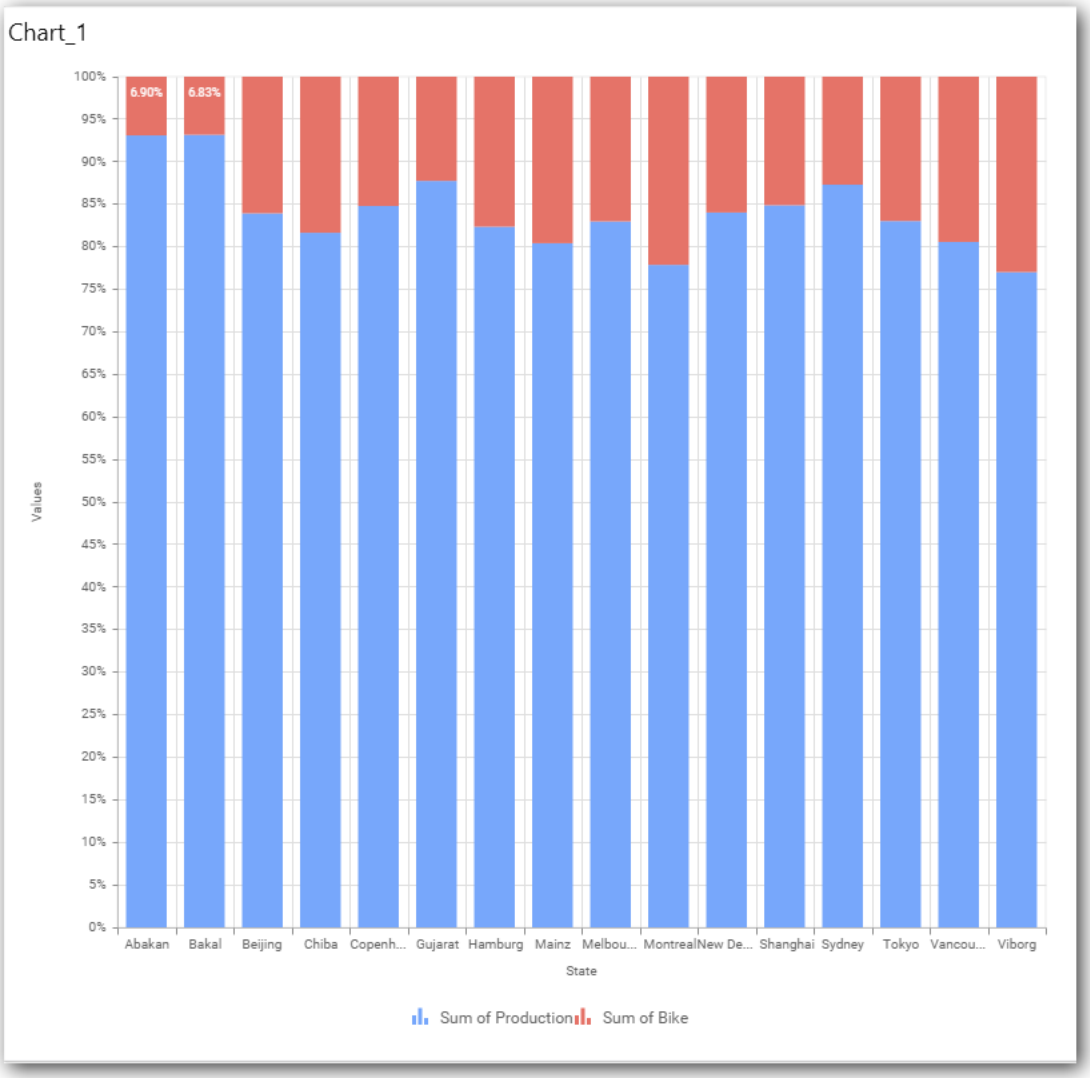
You can drill down the chart by clicking on the chart.



The drilled view of the chart is follows

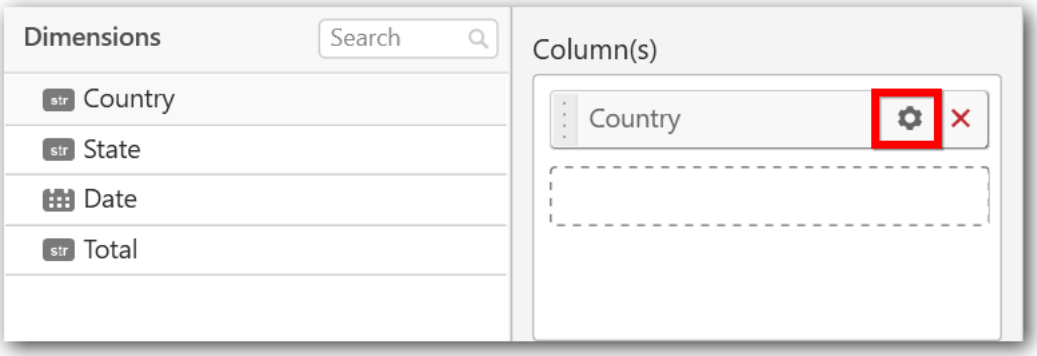


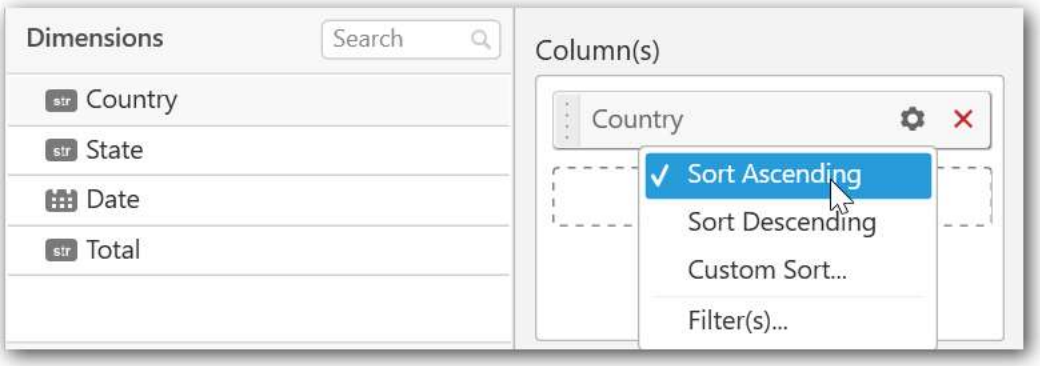
If you click No the new Dimension value will replace old value.



You can also add Measures and Expression columns into Column(s) field.

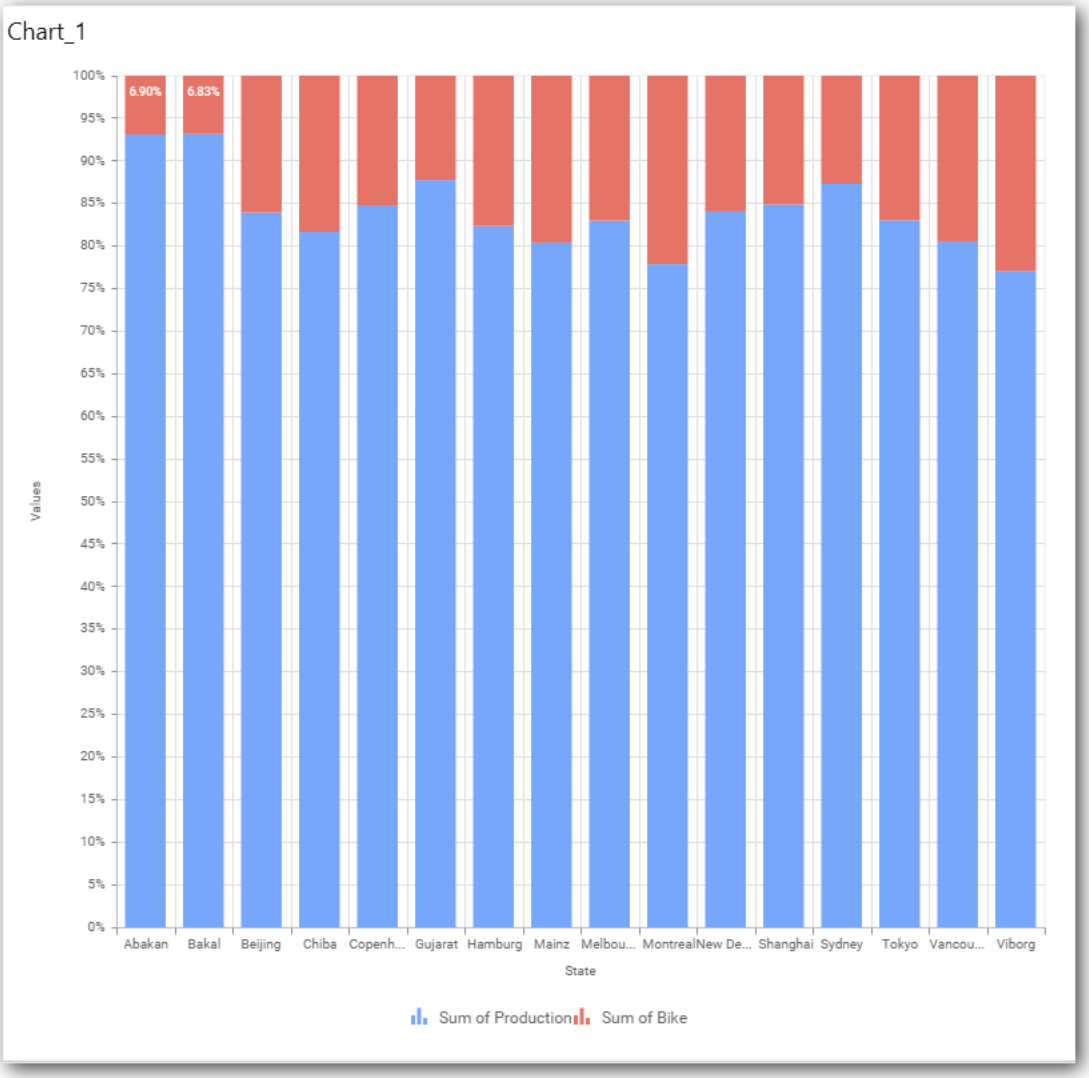
You have options to change the settings.



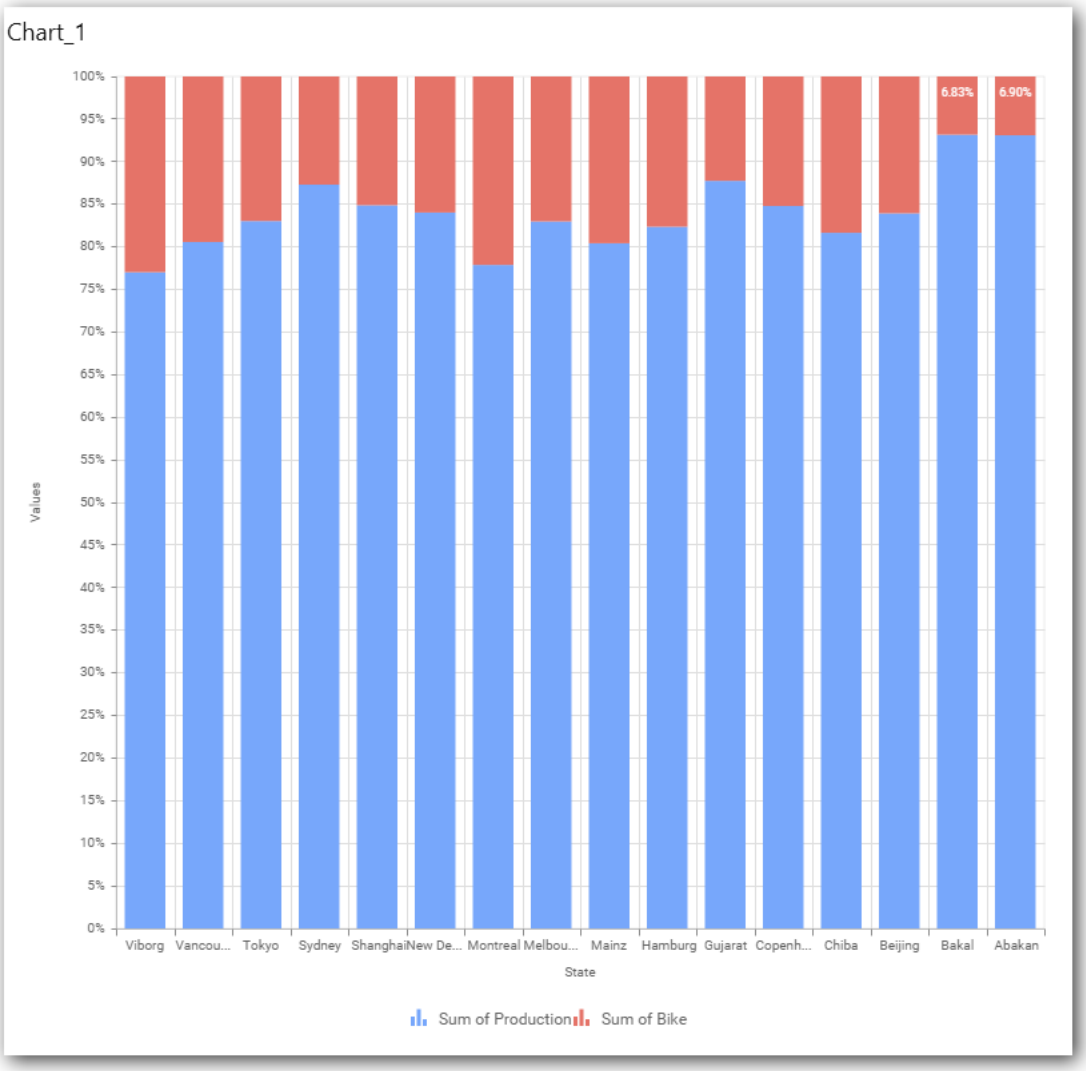


You can sort the chart either in Ascending or Descending series.

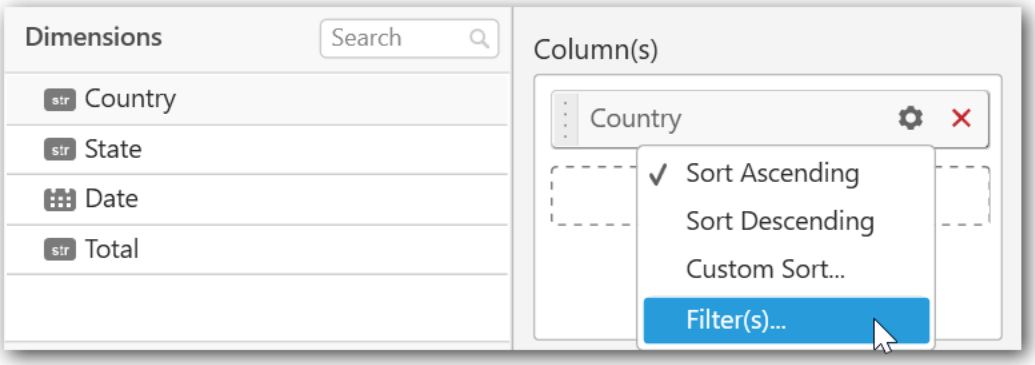
Ascending Order:

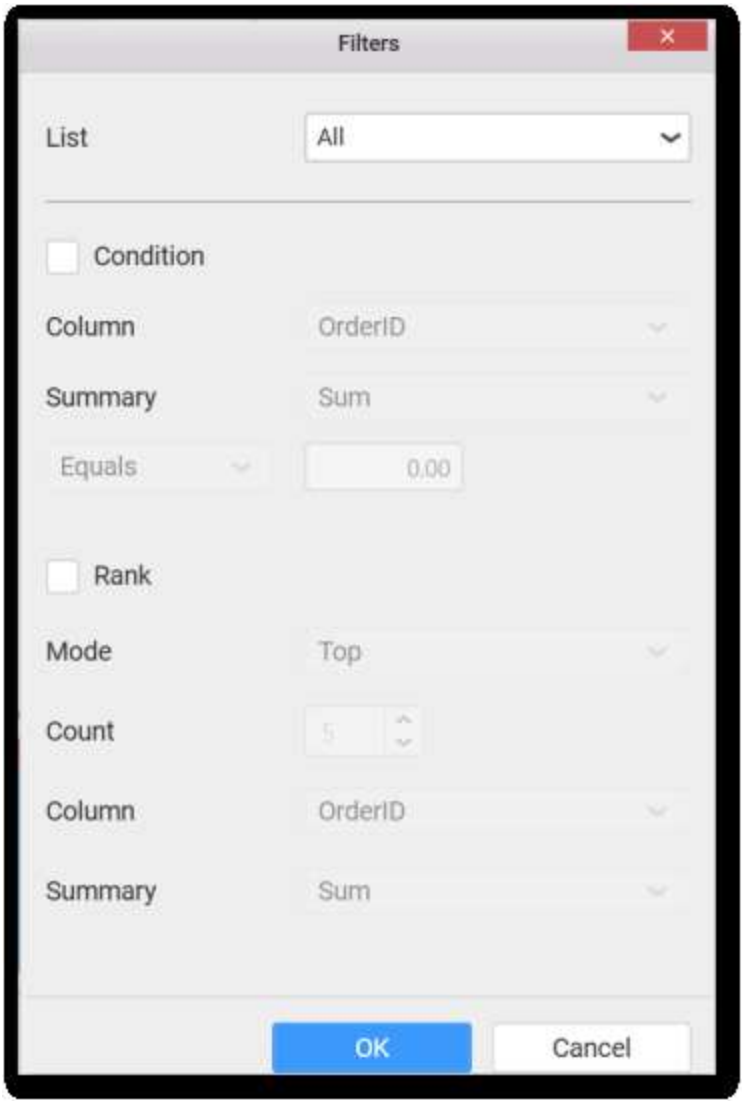


Descending order:

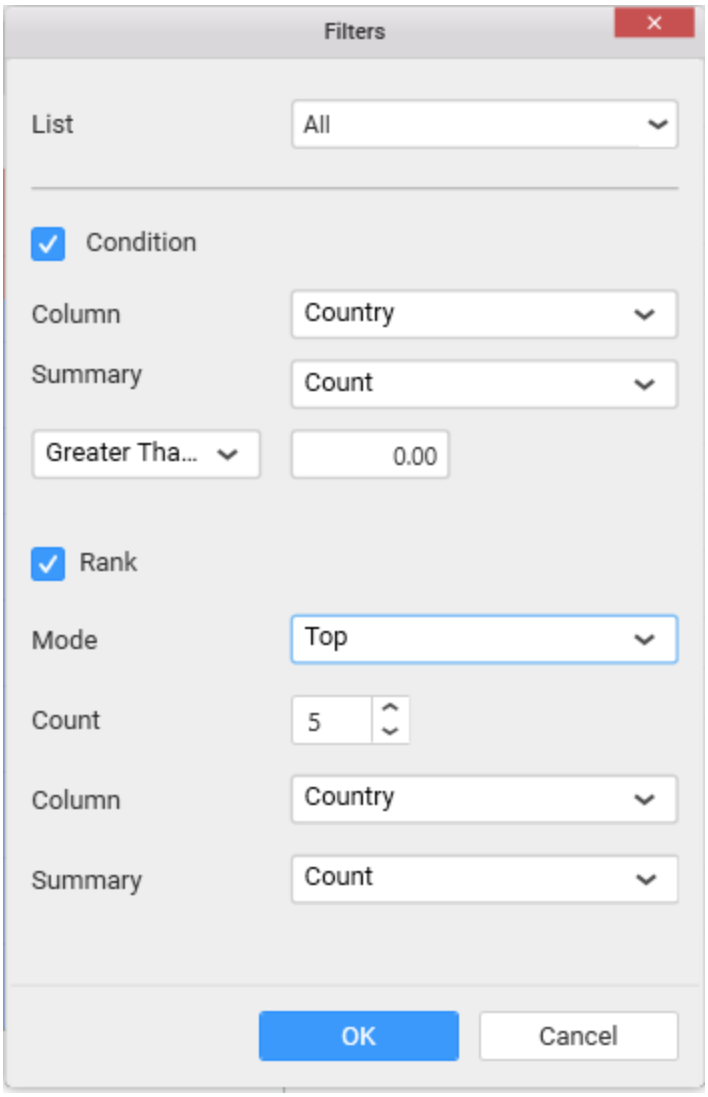


You can apply a filter

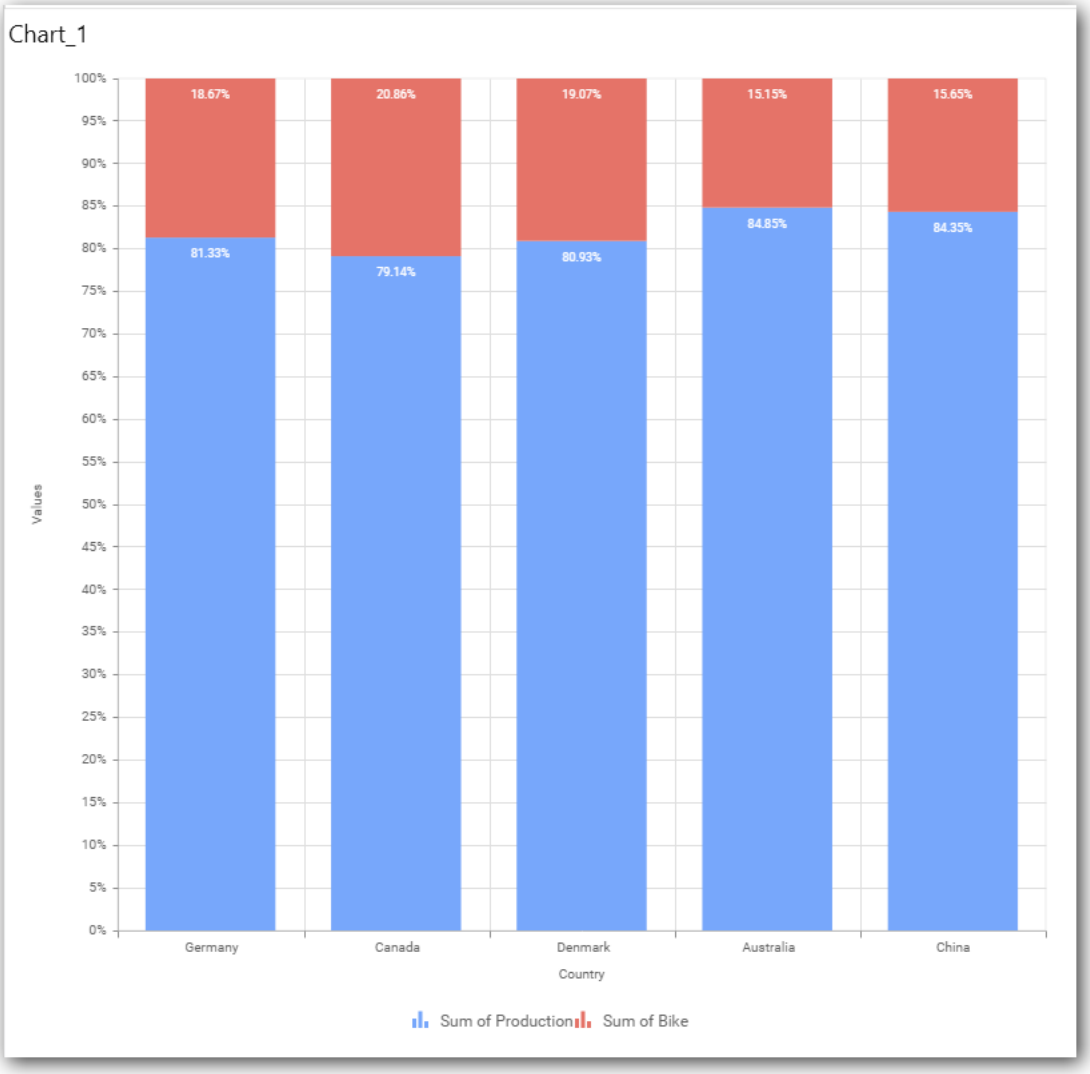




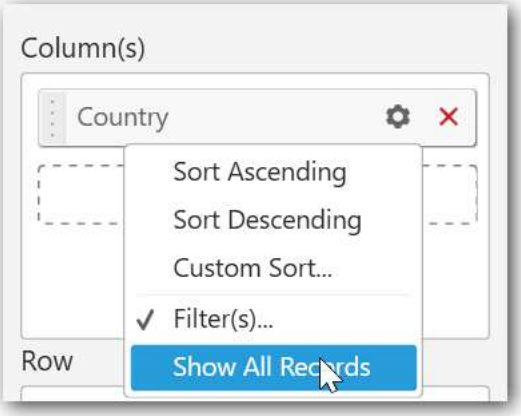
Select the Conditions and Rank you need.



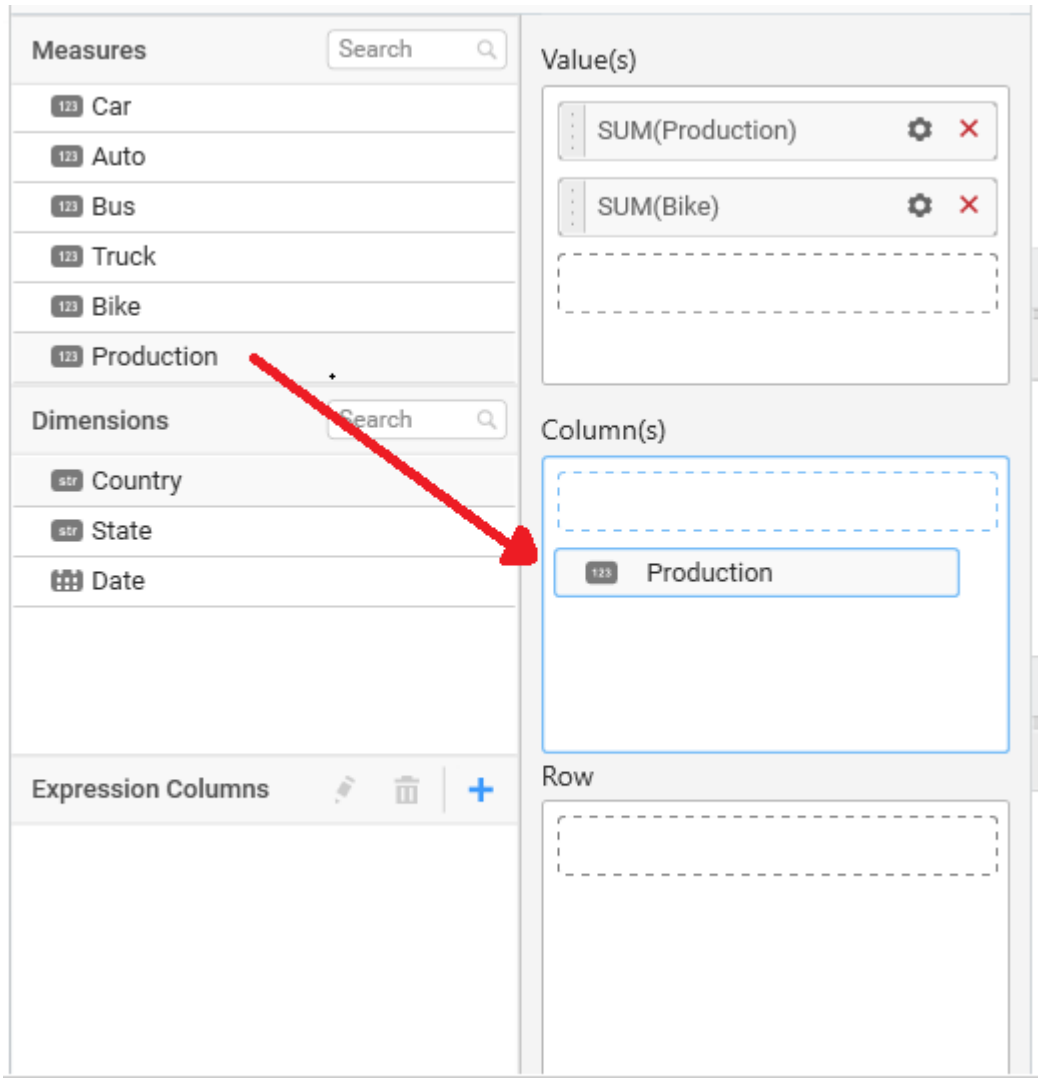
Now the chart will be rendered like this.

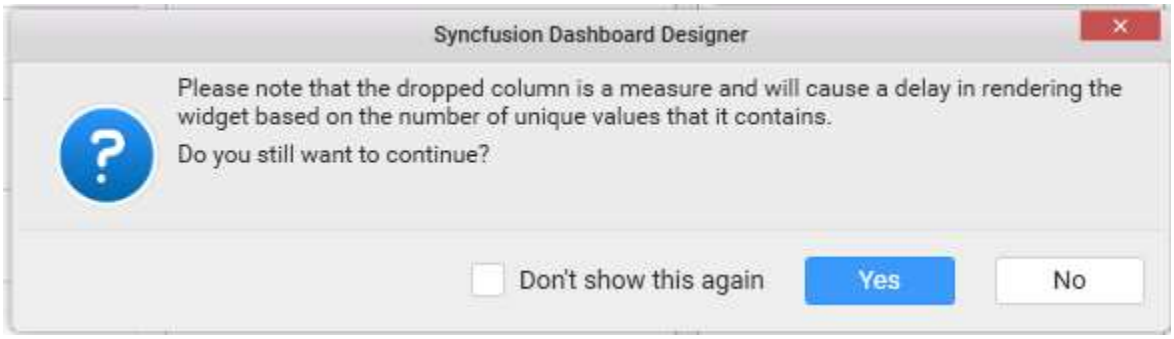


To show all records again click on Show All Records.



On adding Measures into Column(s) will show the following alert.

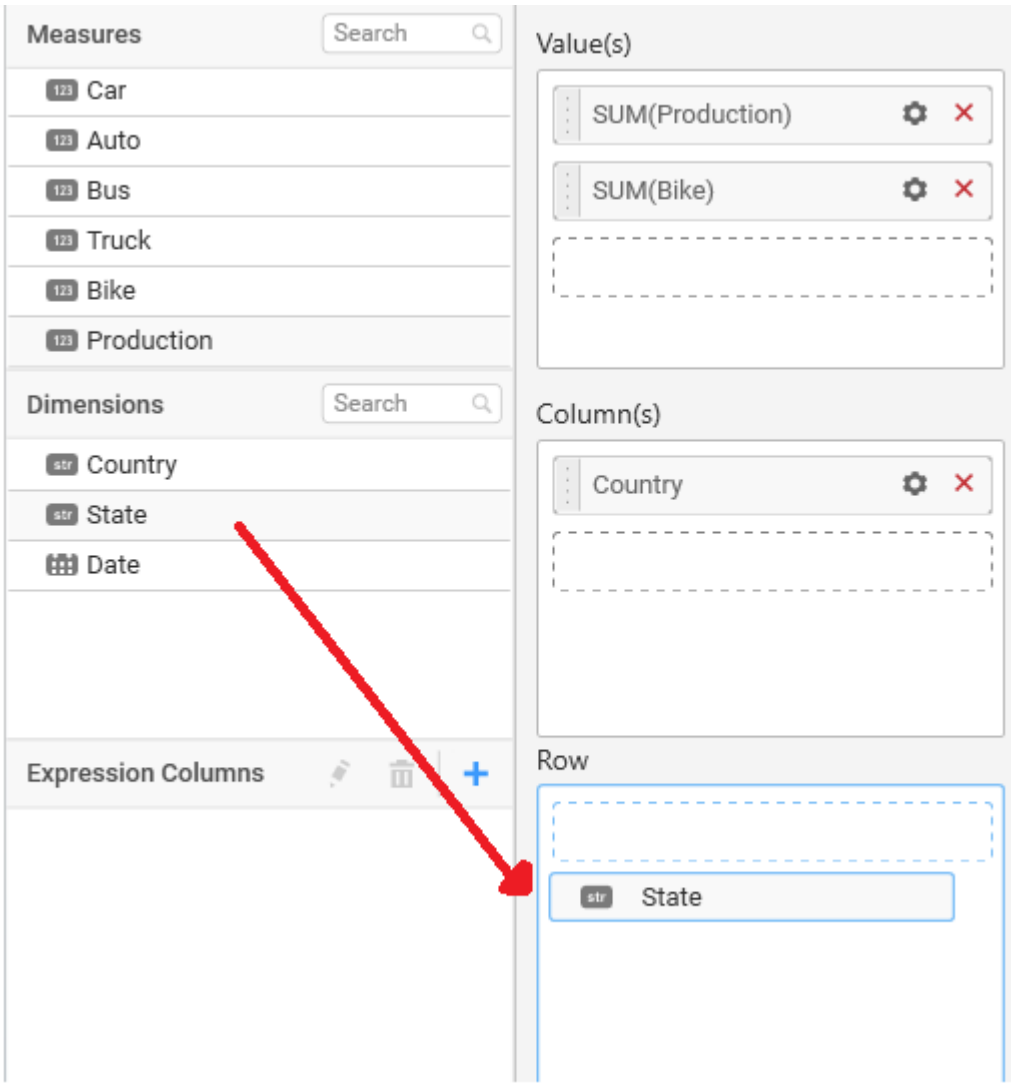




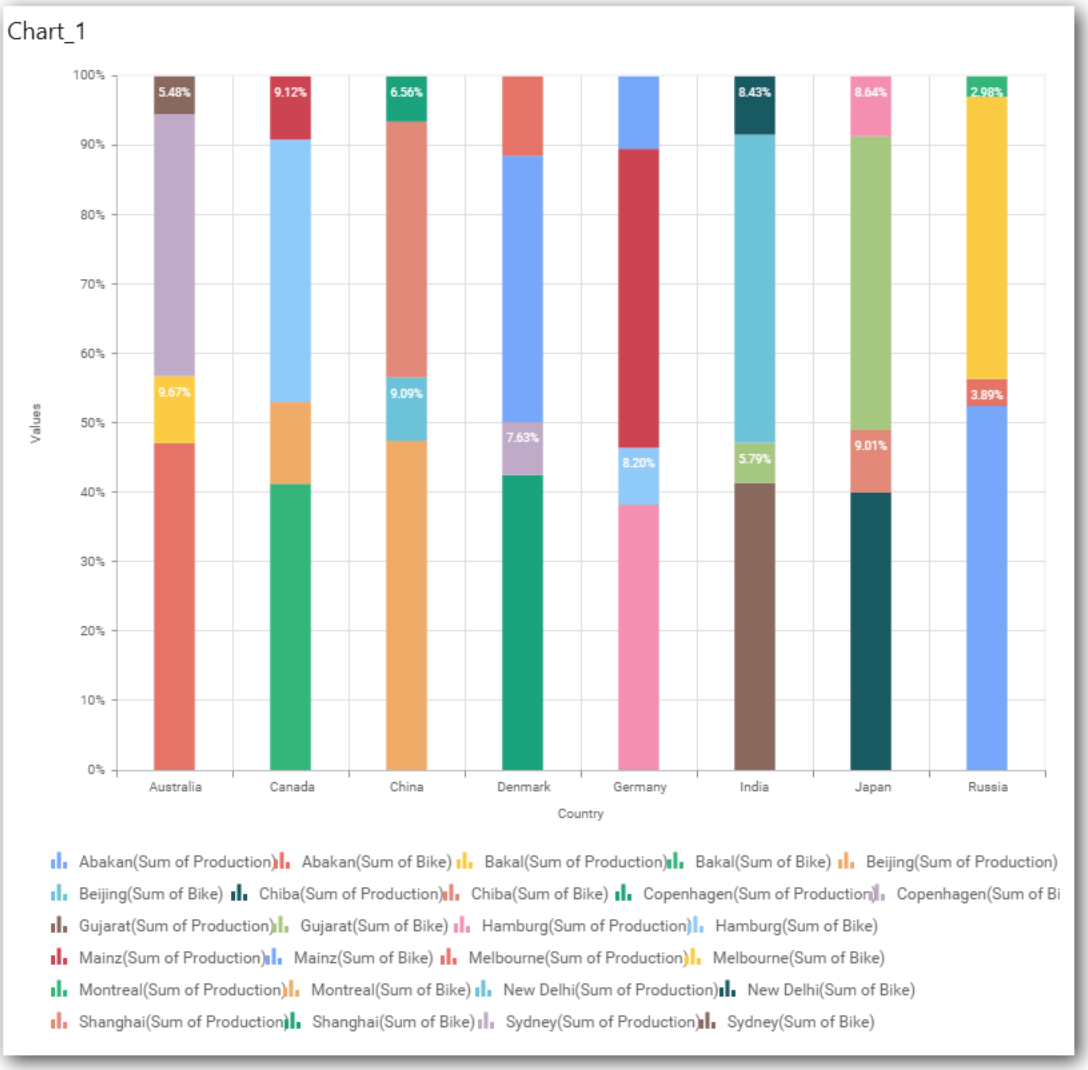
To continue select Yes, otherwise select No.

Assigning Row

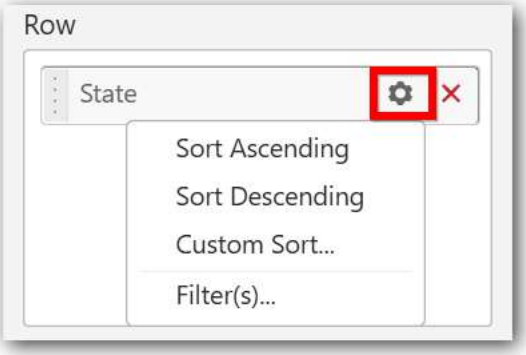
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s)



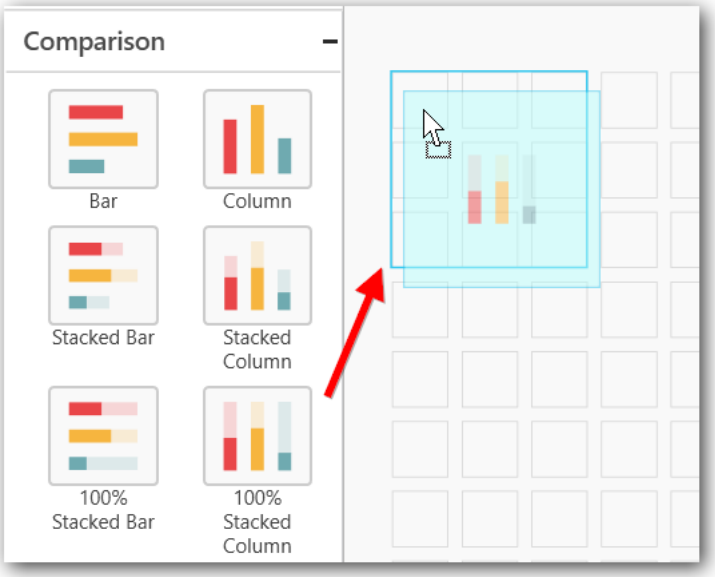
How to configure SSAS to 100% Stacked Column Chart?

100% Stacked Column Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you

would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuration of SSAS data to 100% Stacked Column chart

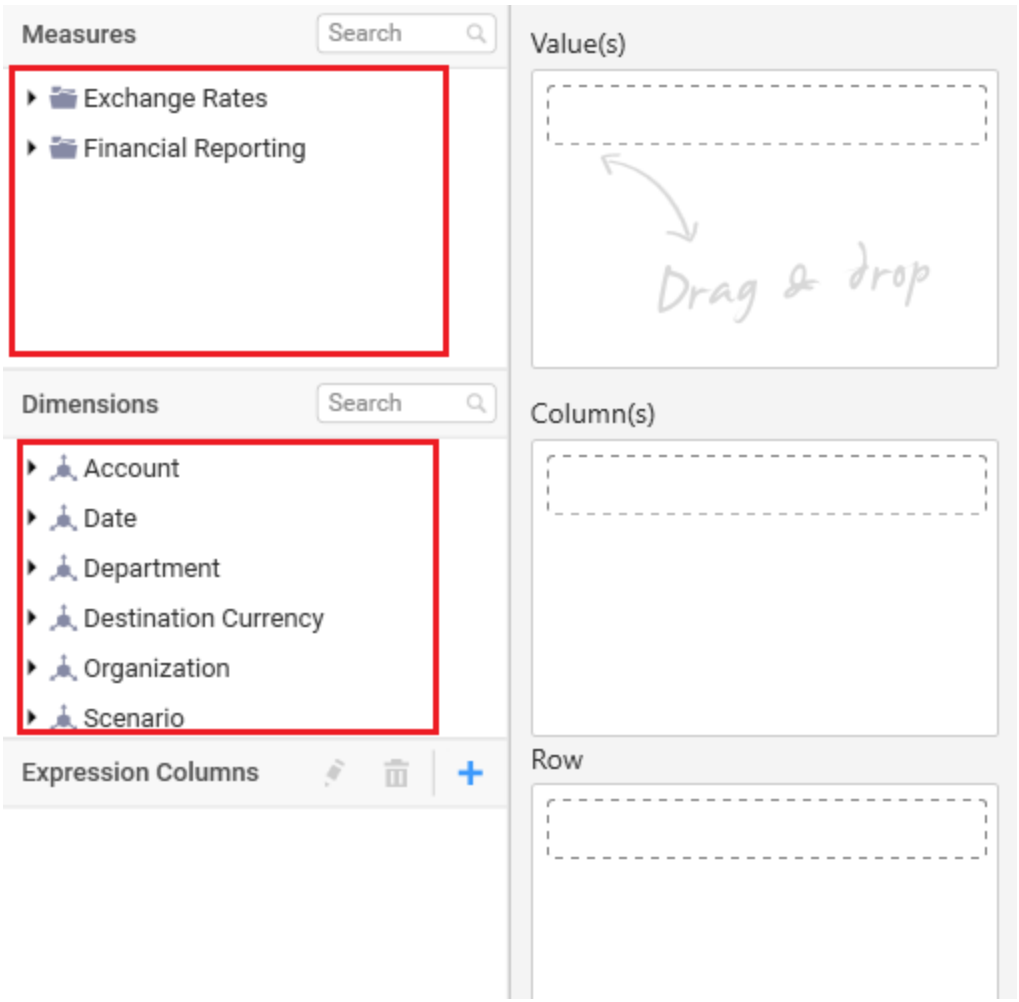
Drag and drop the 100% Stacked Column chart widget into canvas and resize into your required size.



Select the dropped widget using mouse.

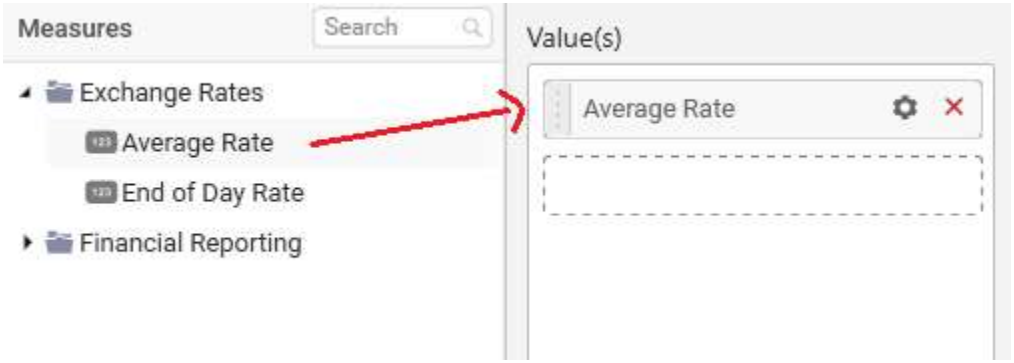
Click the Assign Data button in the toolbar.

A data pane will be opened with available Measures and Dimensions.

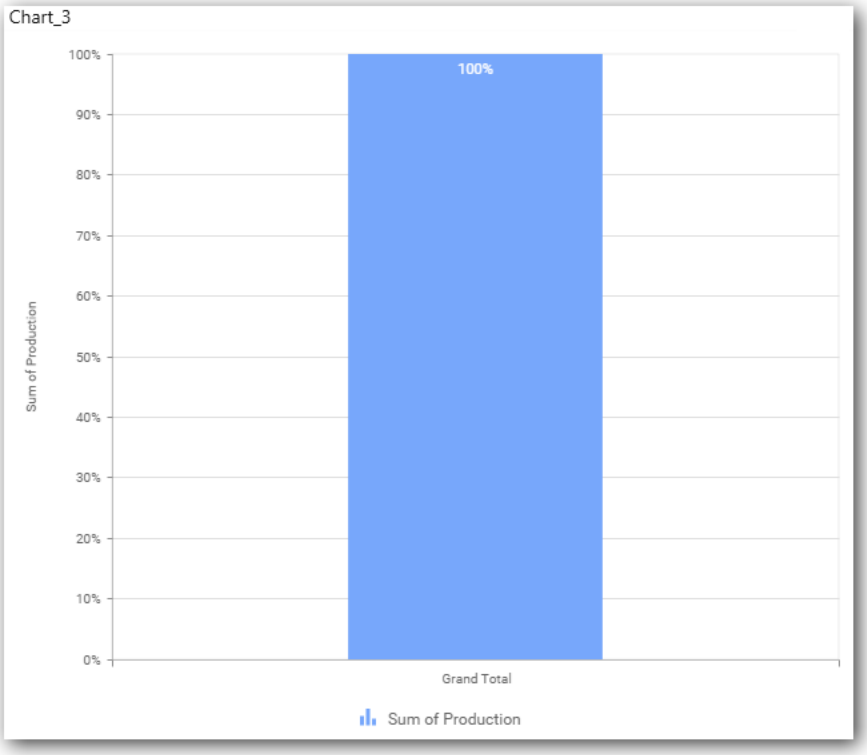


Assigning Value(s)

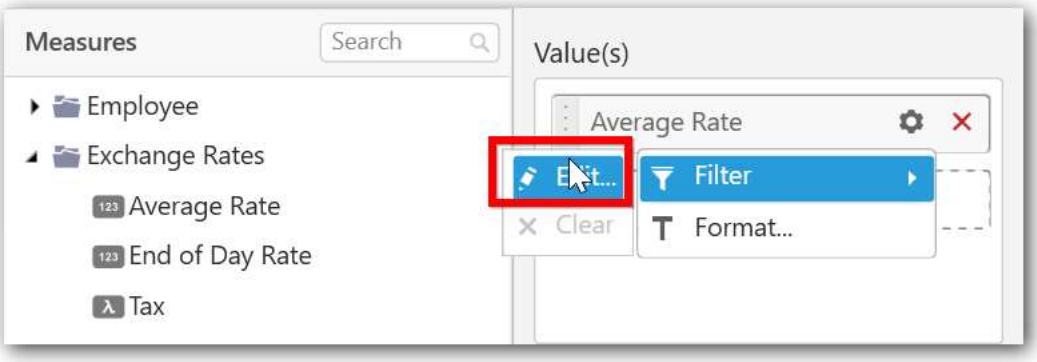
Drag and drop a column under Measures category into Value(s) section.



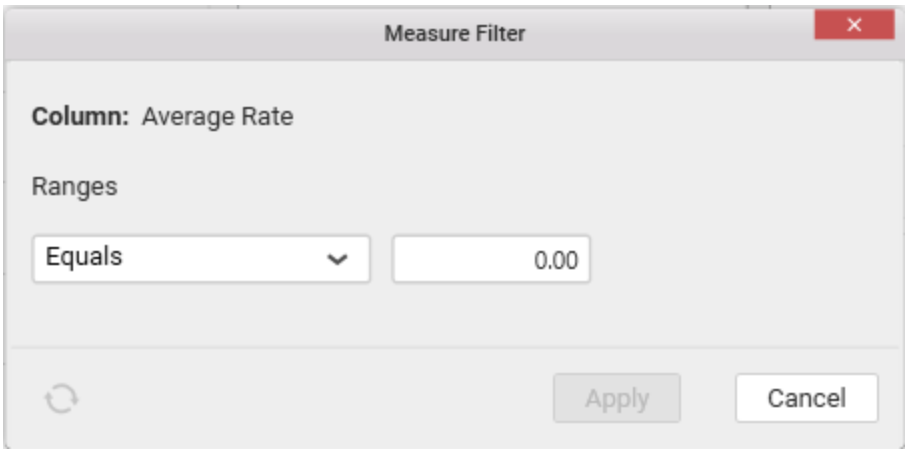
Now the chart will be rendered like this.



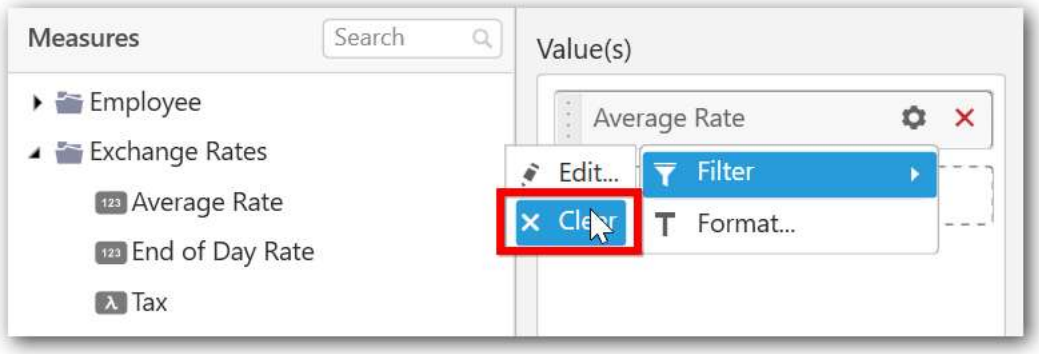
Define the filter criteria to match through choosing Edit option in Filter menu item.



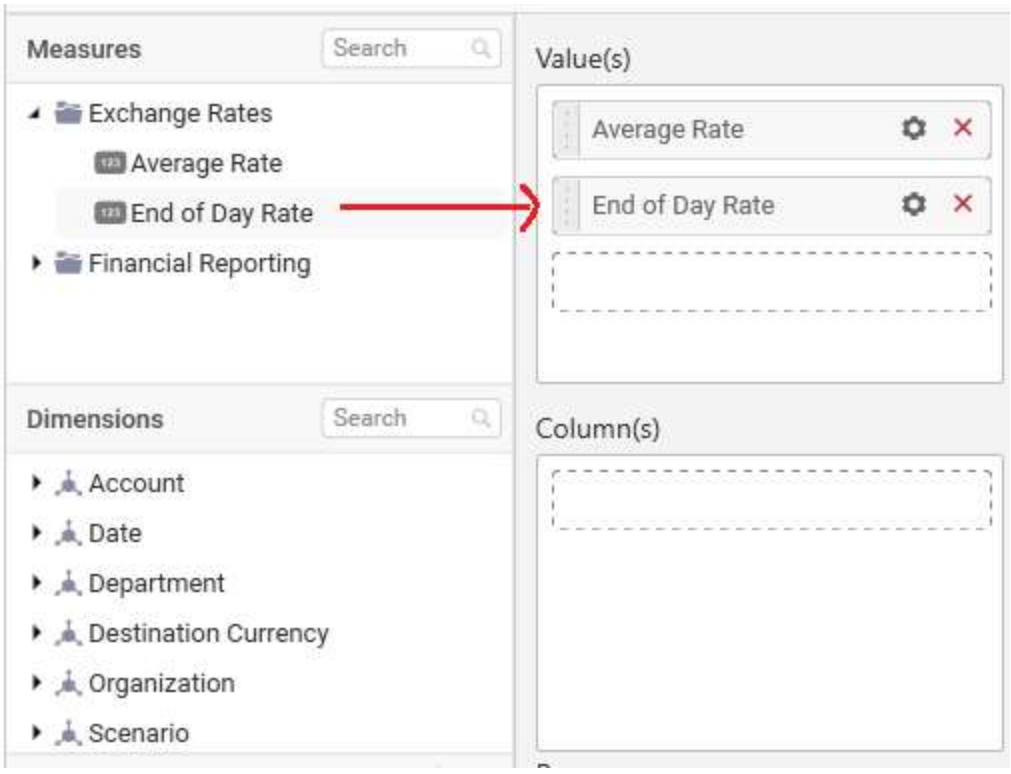
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.

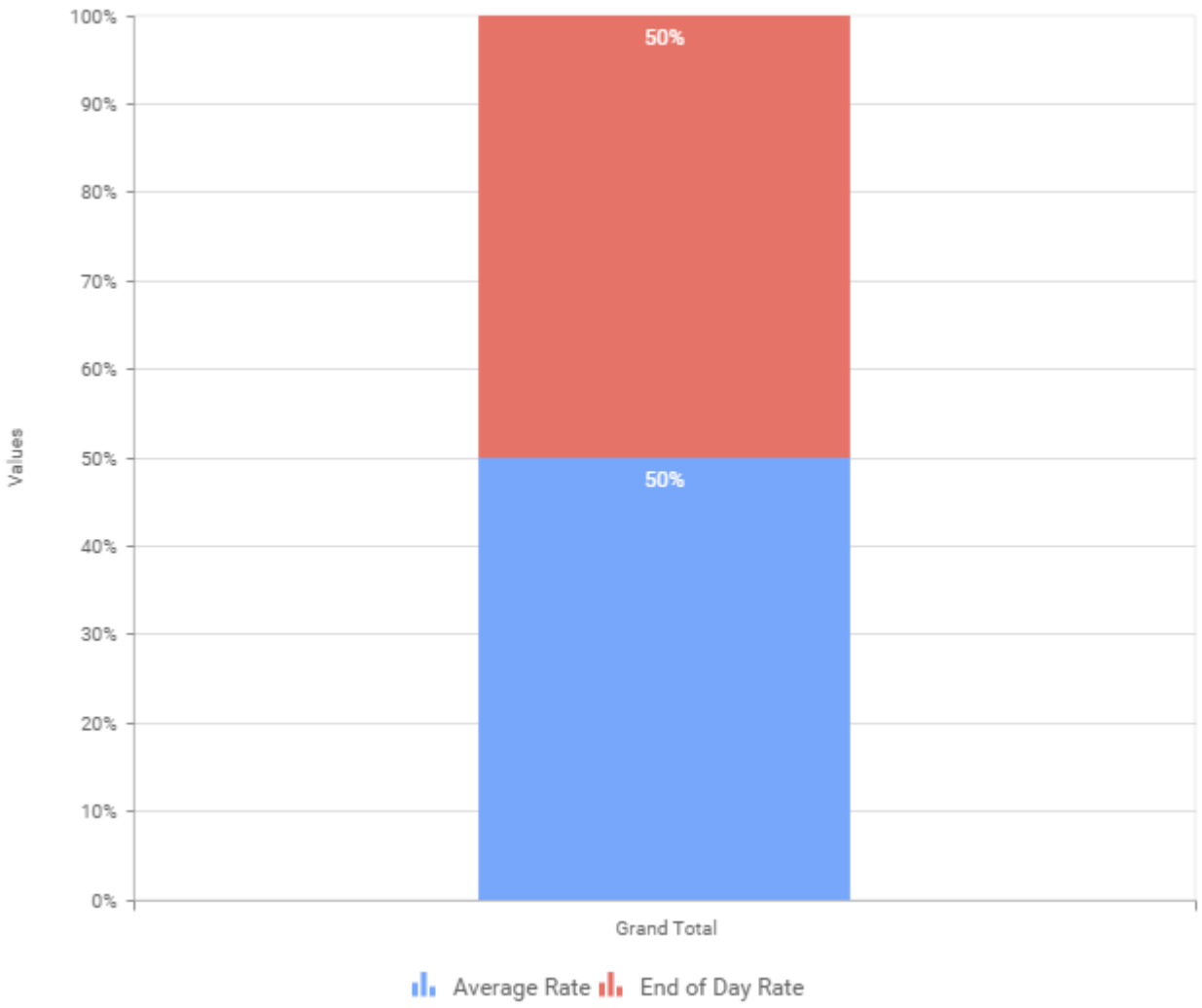


Select Clear option to clear the defined filter.



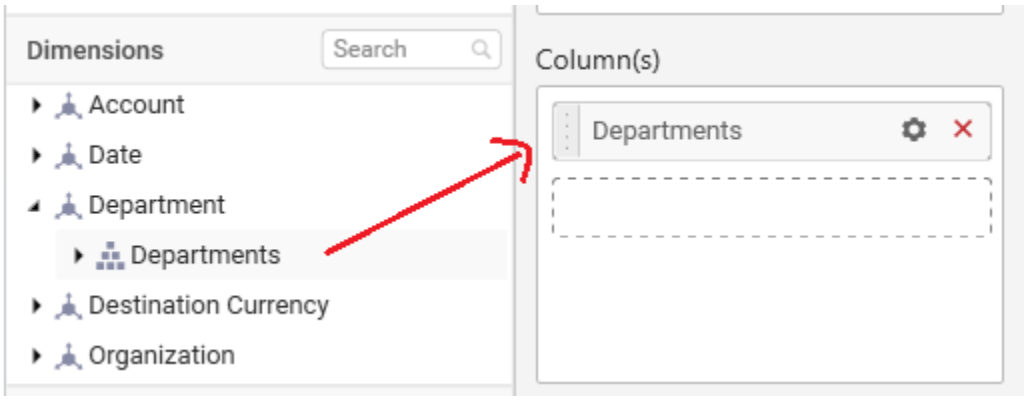
You can also add more than one column to the Value(s) section.

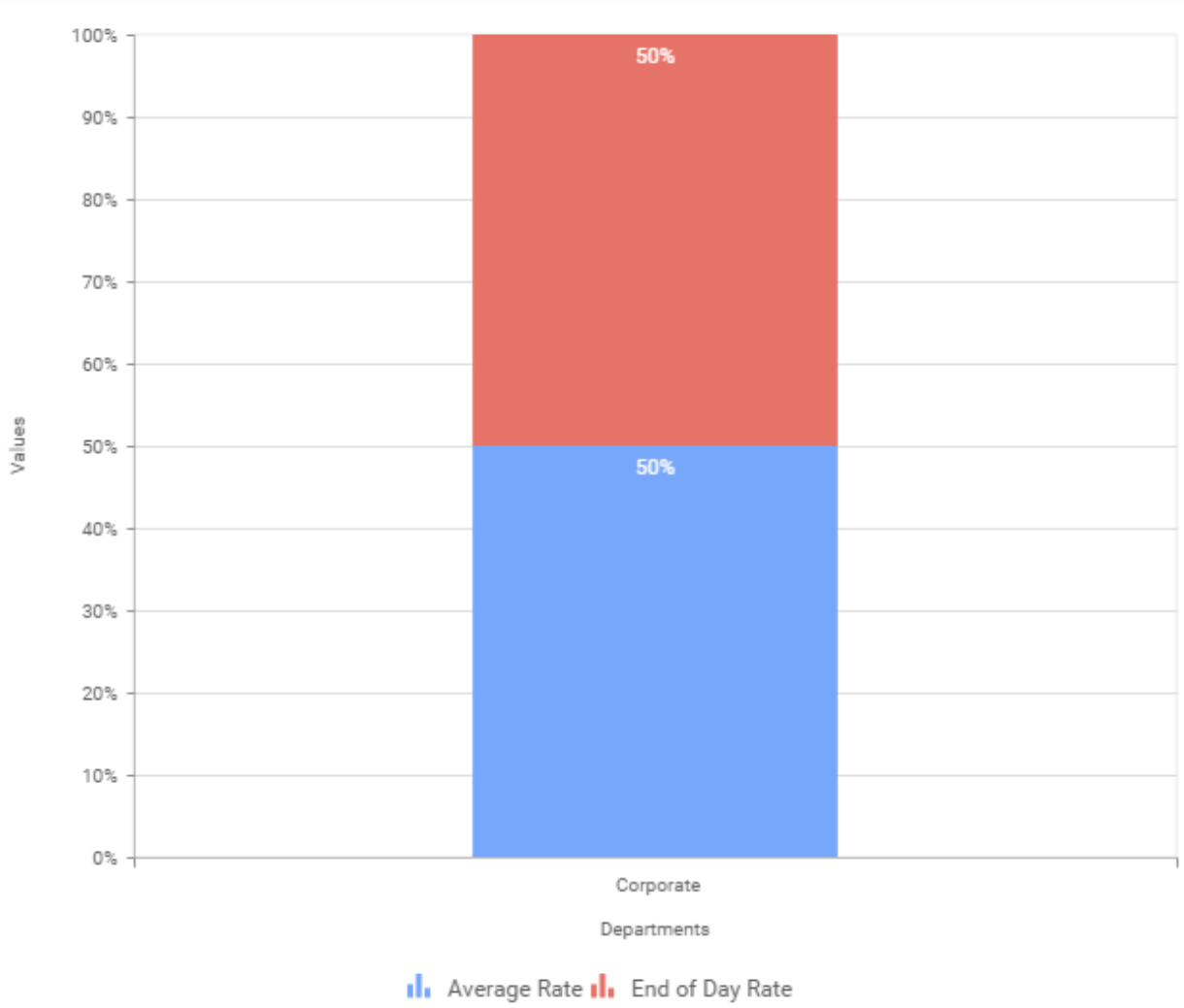




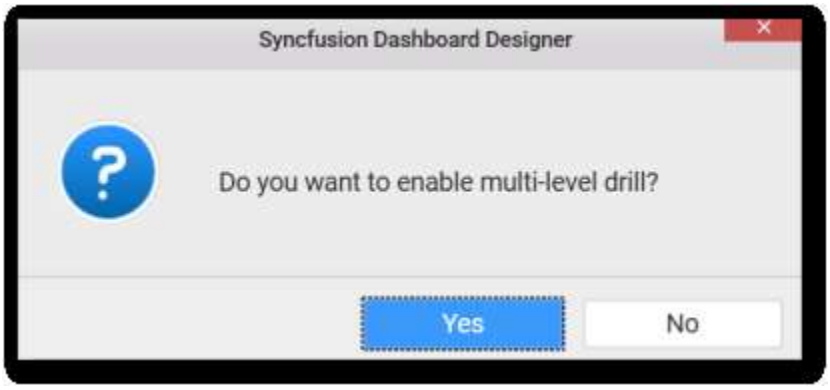
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.



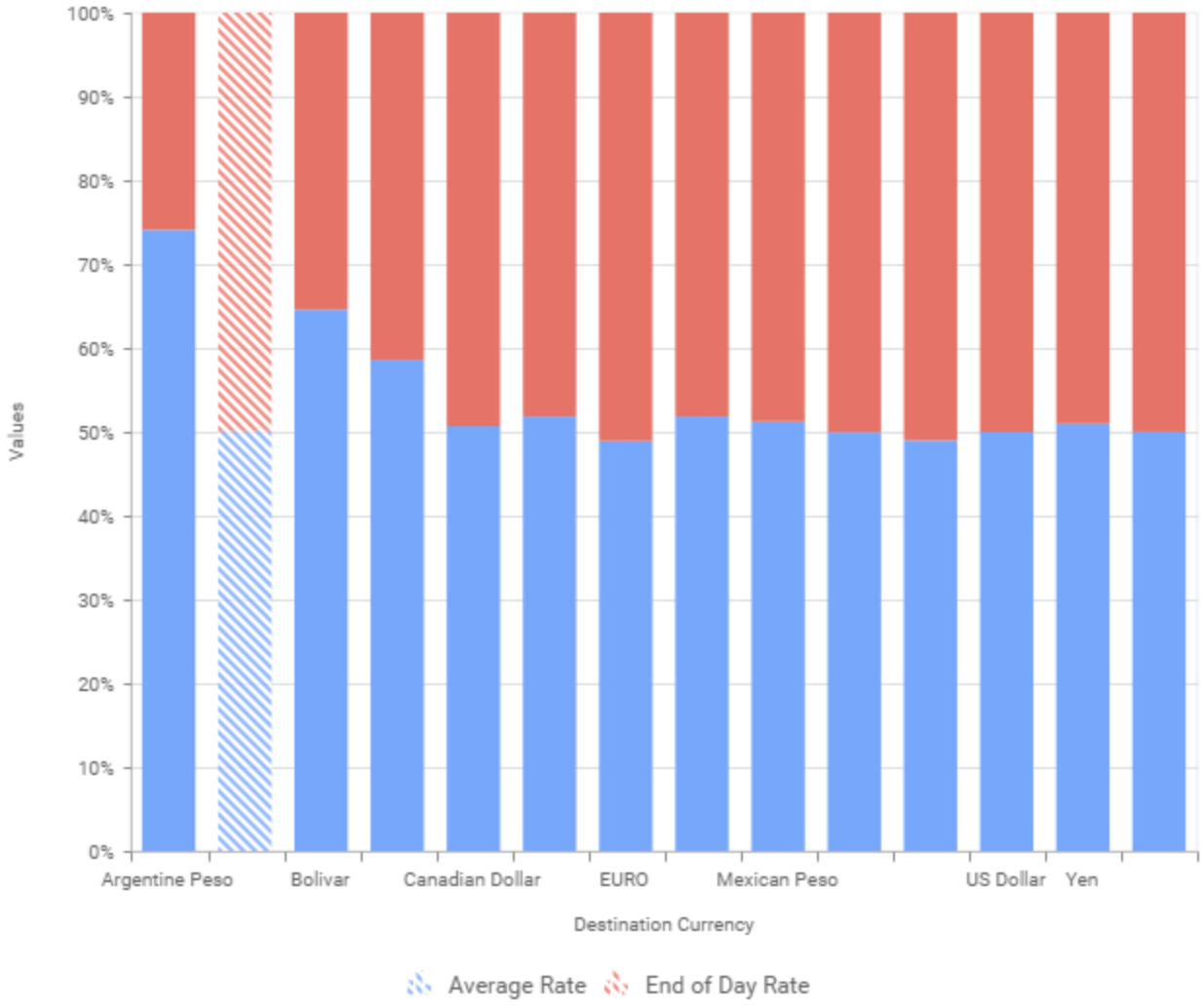


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

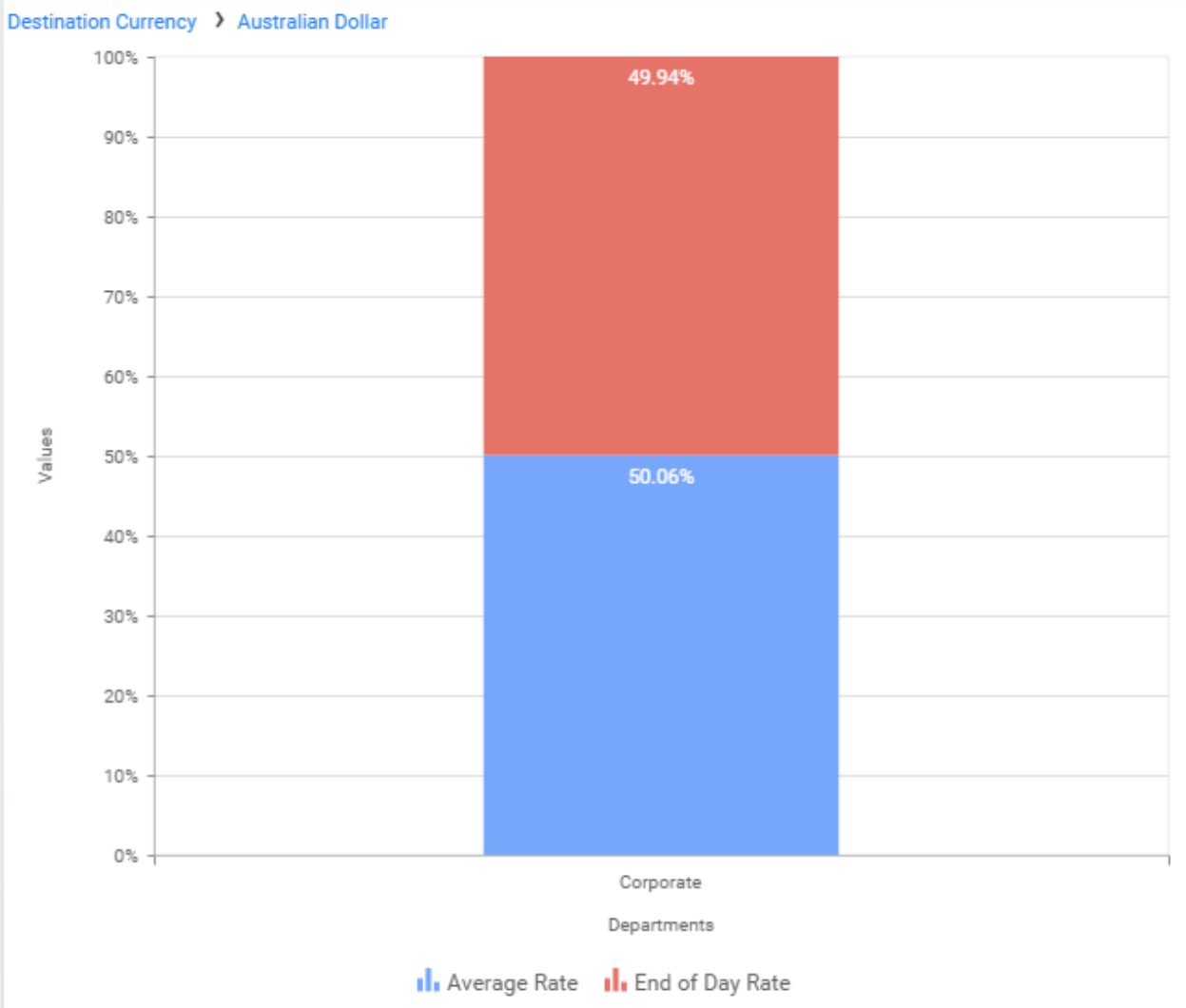


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

Click the respective data value marker in chart to drill into its inner level.

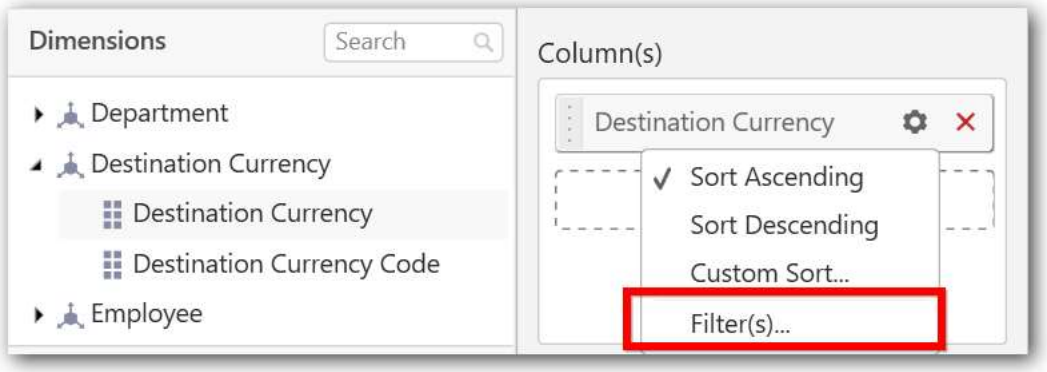


The drilled view of the chart is follows.

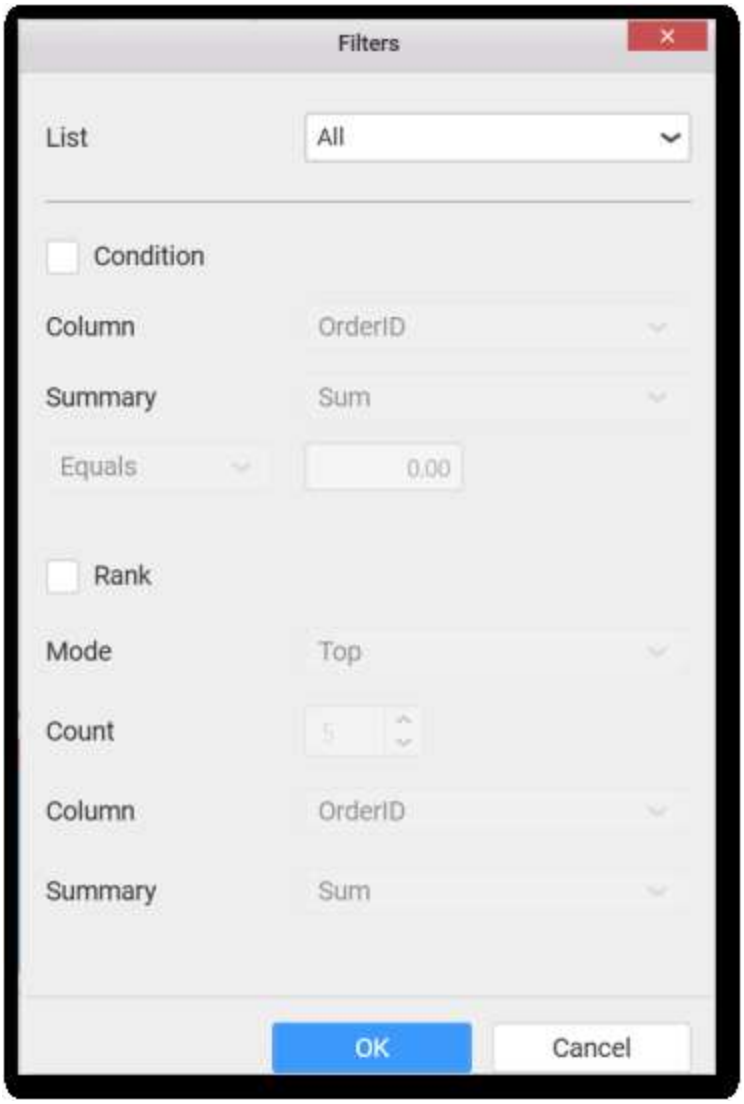


Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level.

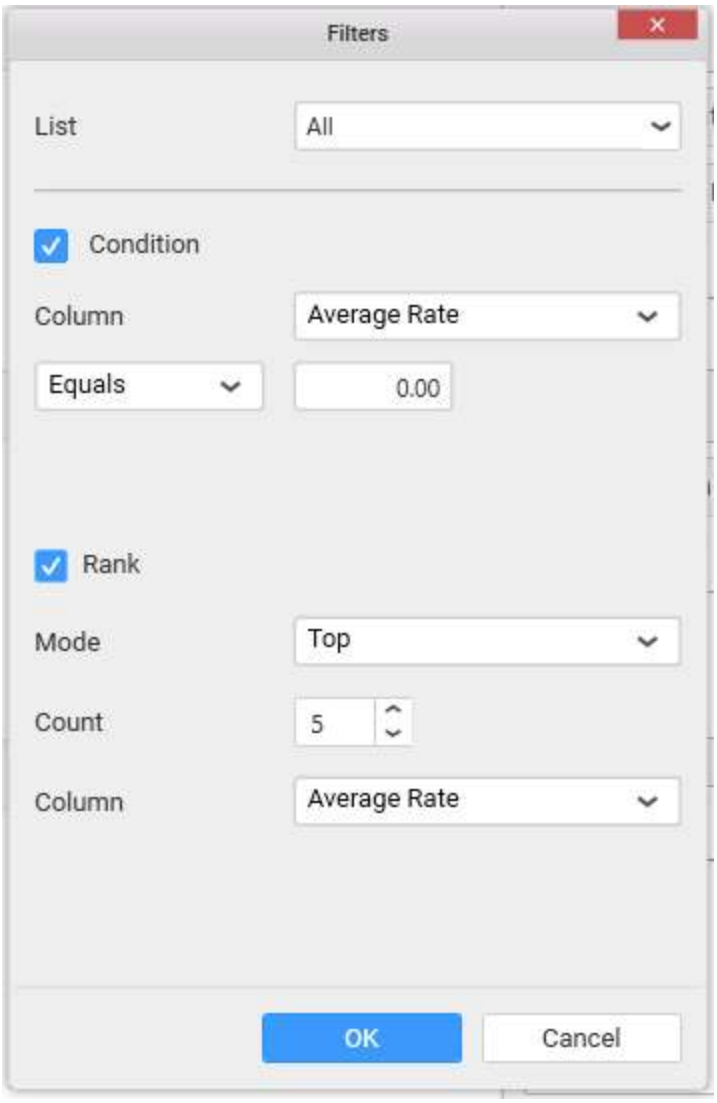
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



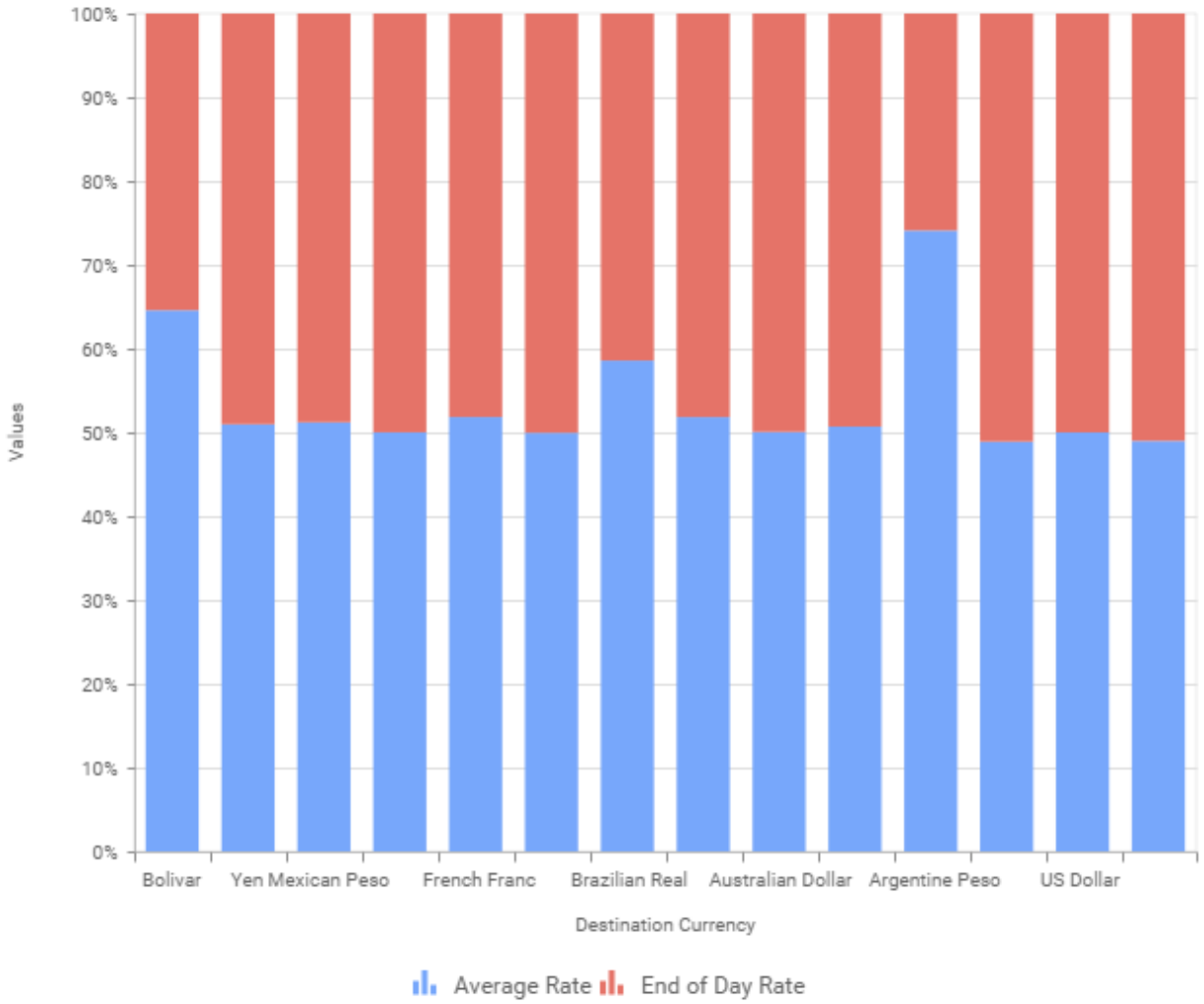
Select Filter(s)... to launch the Filters window.



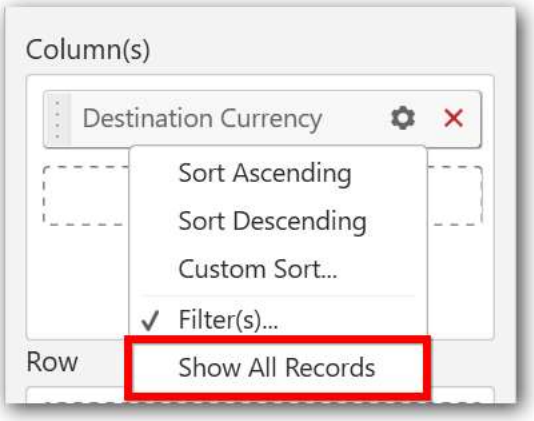
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

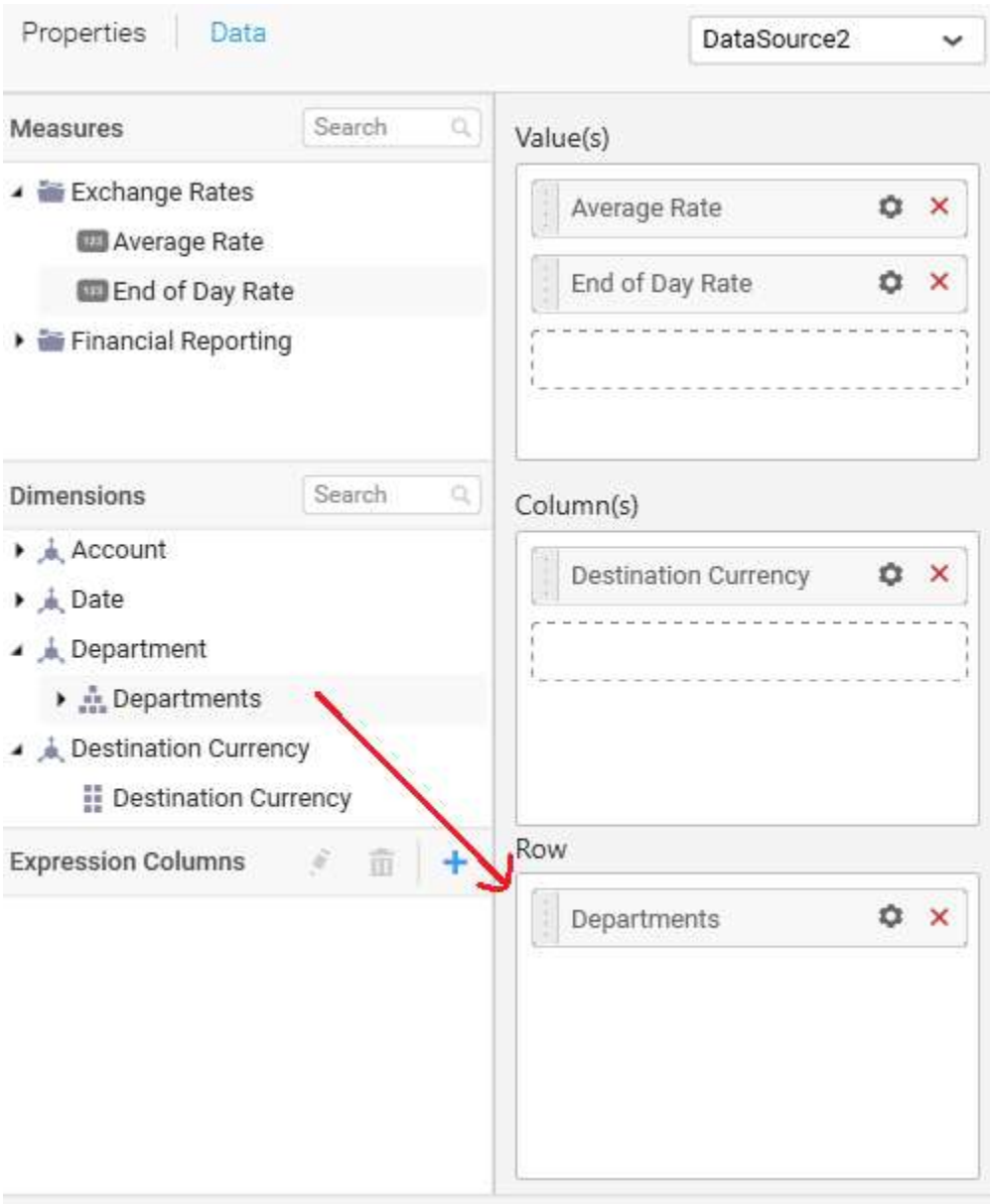


To show all records again click on Show All Records.

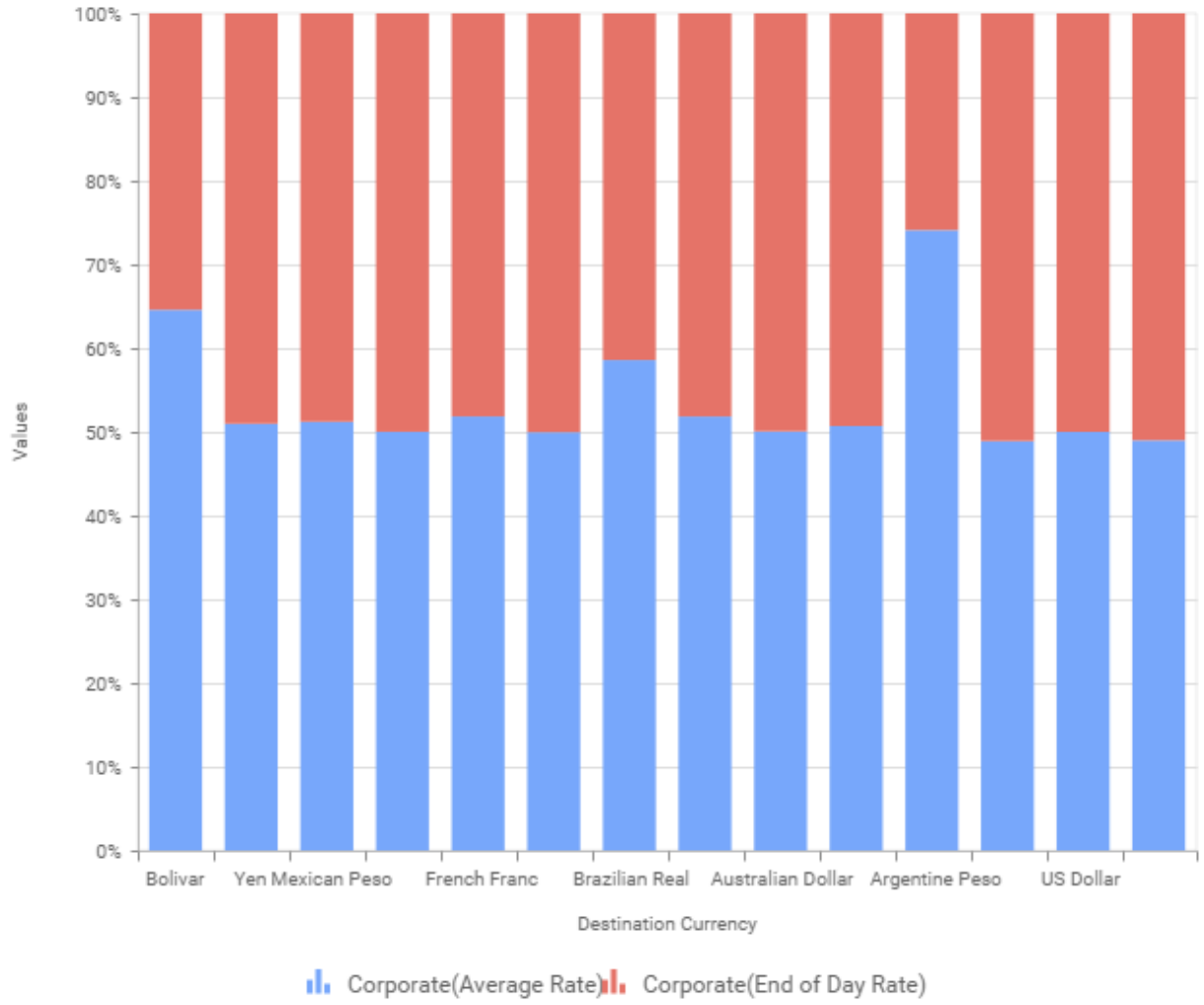


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.



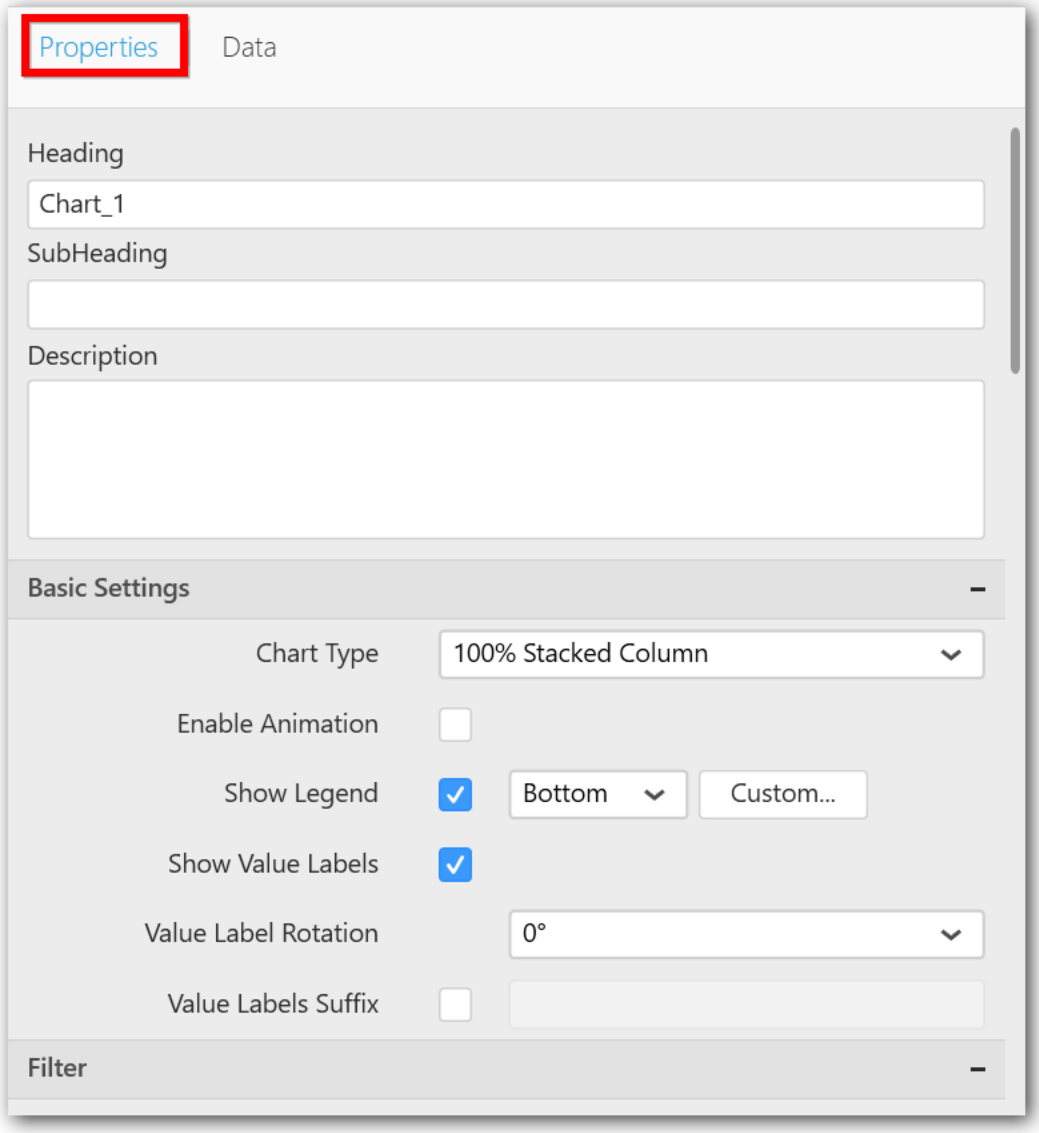
How to format 100% Stacked Column Chart?

You can format the 100% stacked column chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into 100% stacked column chart follow the steps

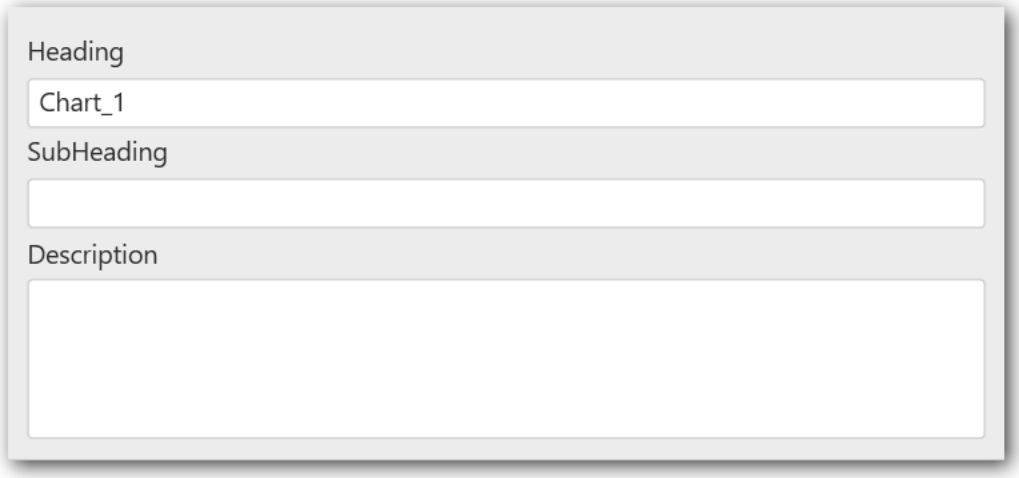
1. Drag and drop the 100% stacked column chart into canvas and resize it to your required size.
2. Configure the data into 100% stacked column chart.
3. Focus on the 100% stacked column chart and Click on Widget Settings.

The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this 100% stacked column chart widget.

SubHeading

This allows you to set sub-title for this 100% stacked column chart widget.

Description

This allows you to set description for this 100% stacked column chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

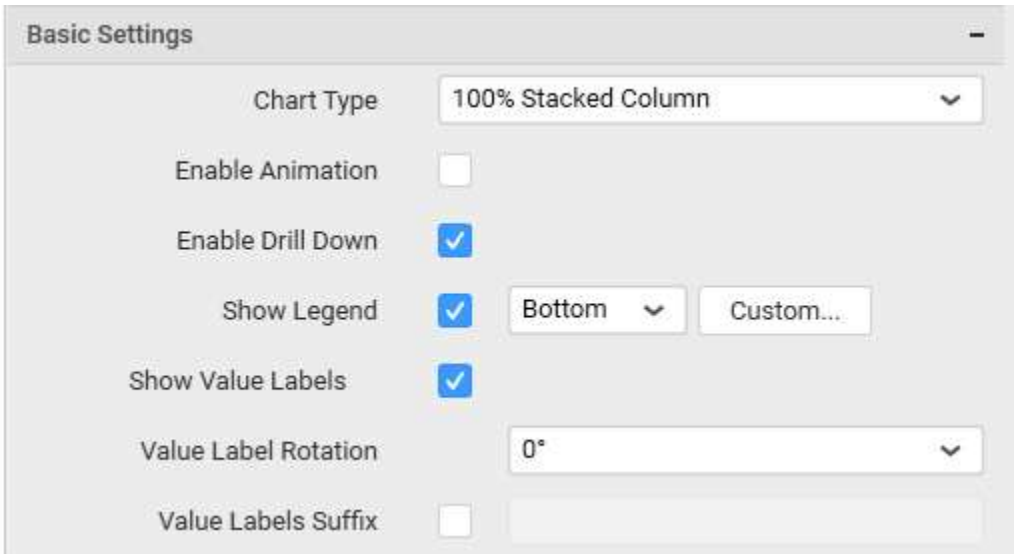


Chart Type

This allows you to switch the widget view from current chart type to another convertible chart type.

Enable Animation

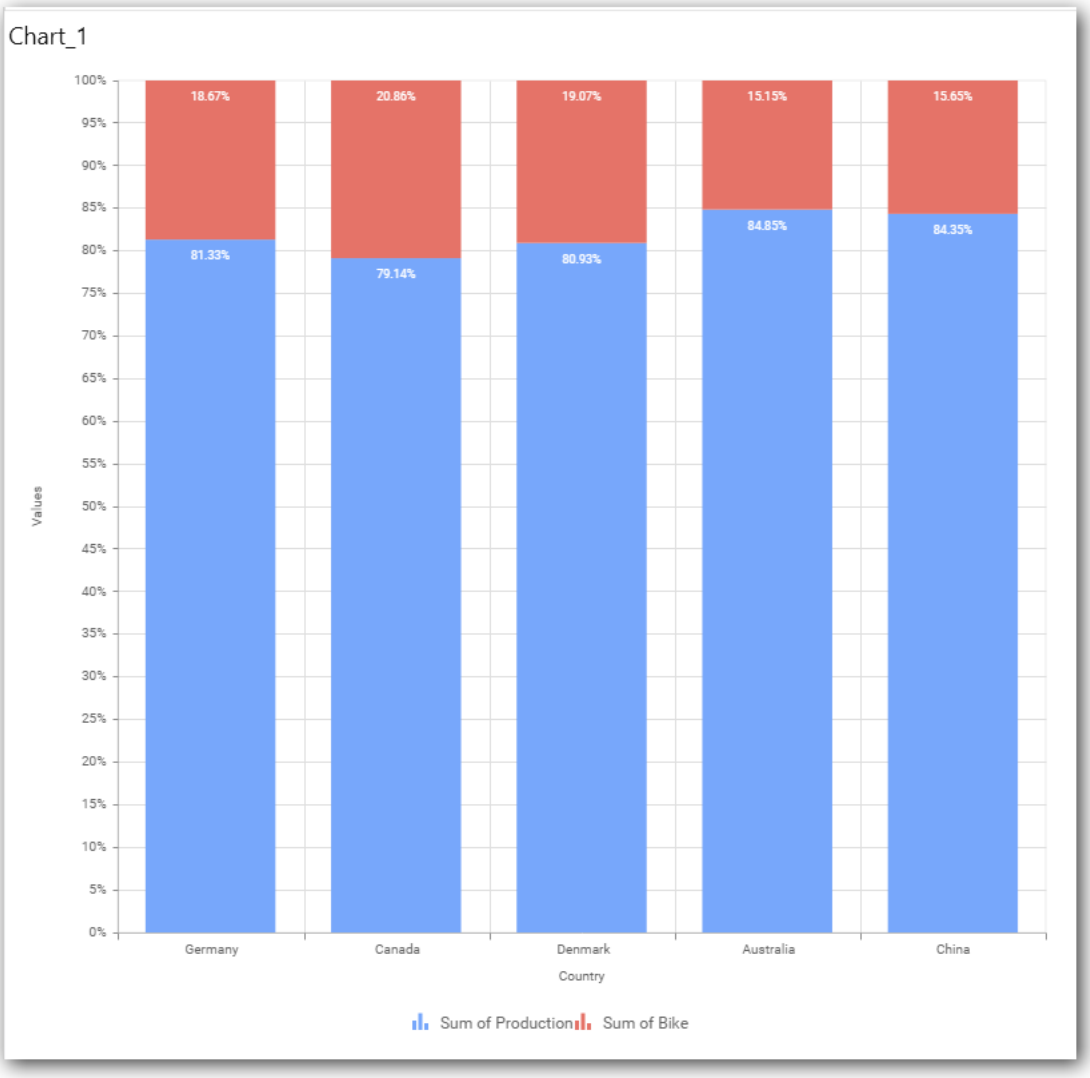
This allows you to enable the series rendering in animated mode.

Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of Custom Legend Text will allow us to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

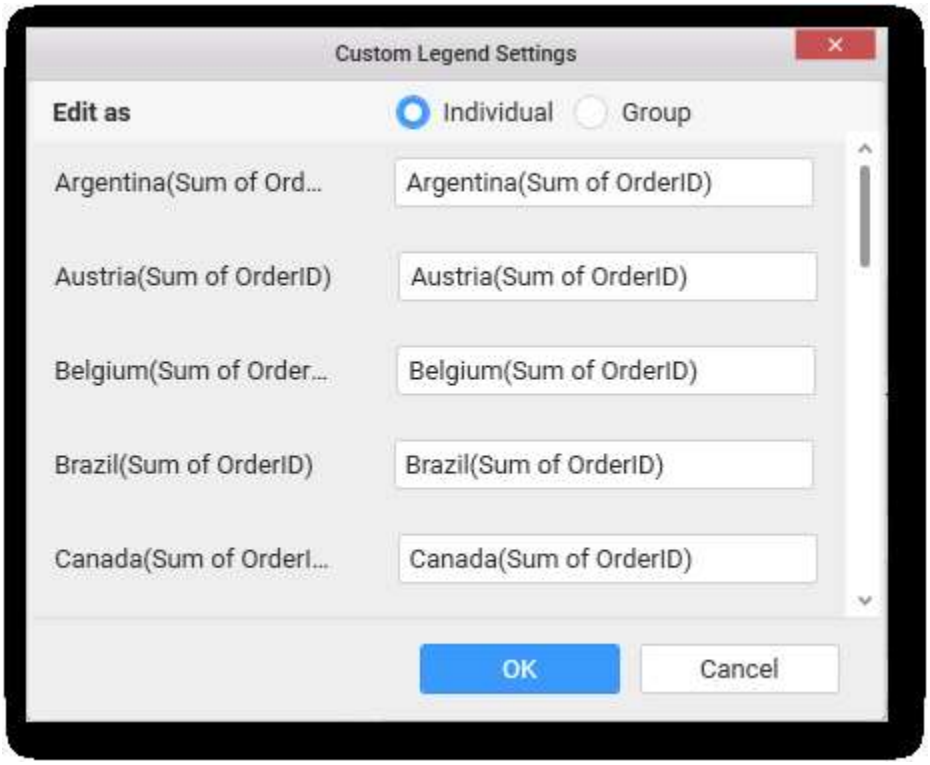
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

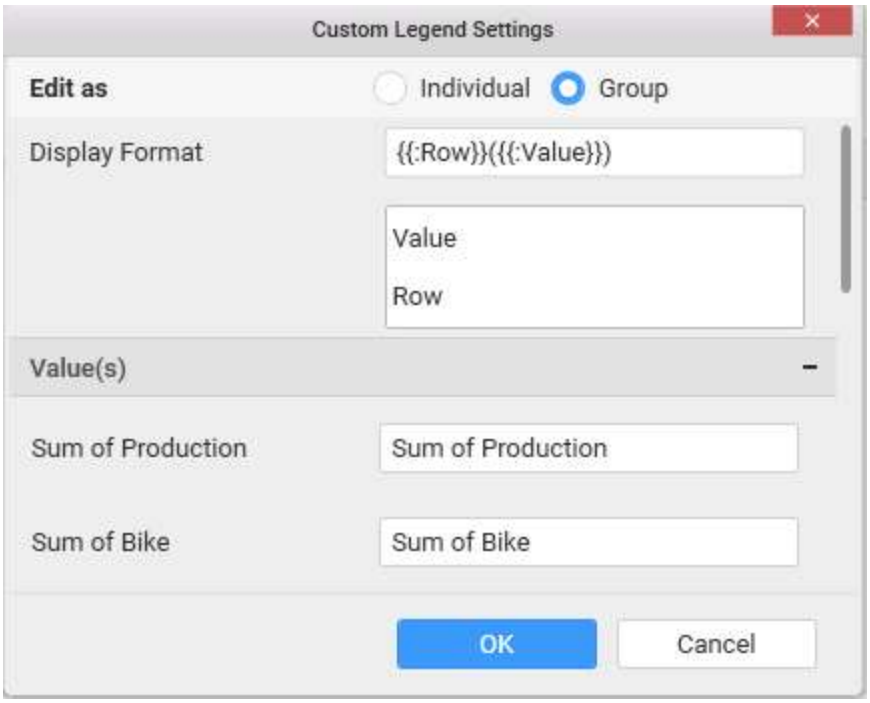
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

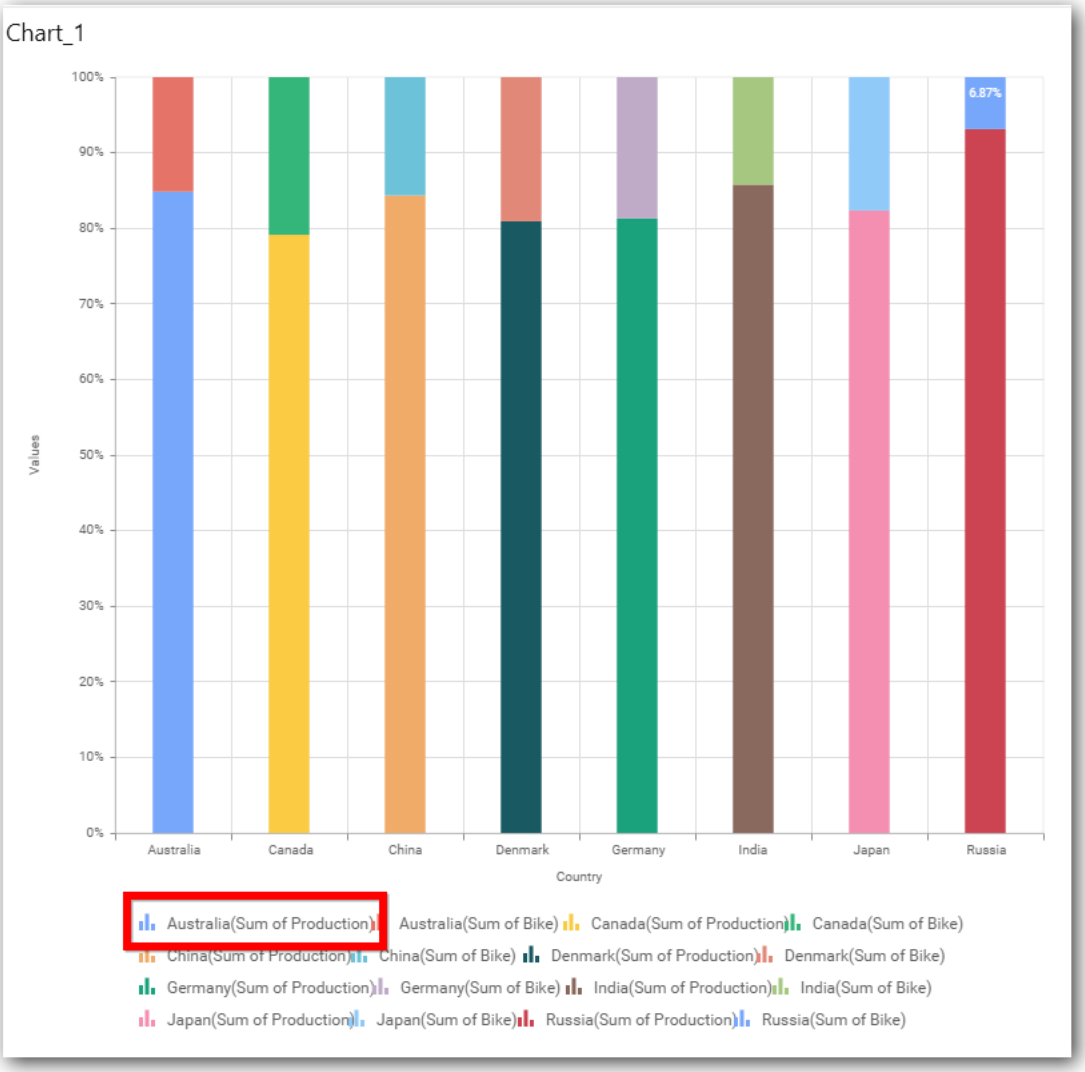


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

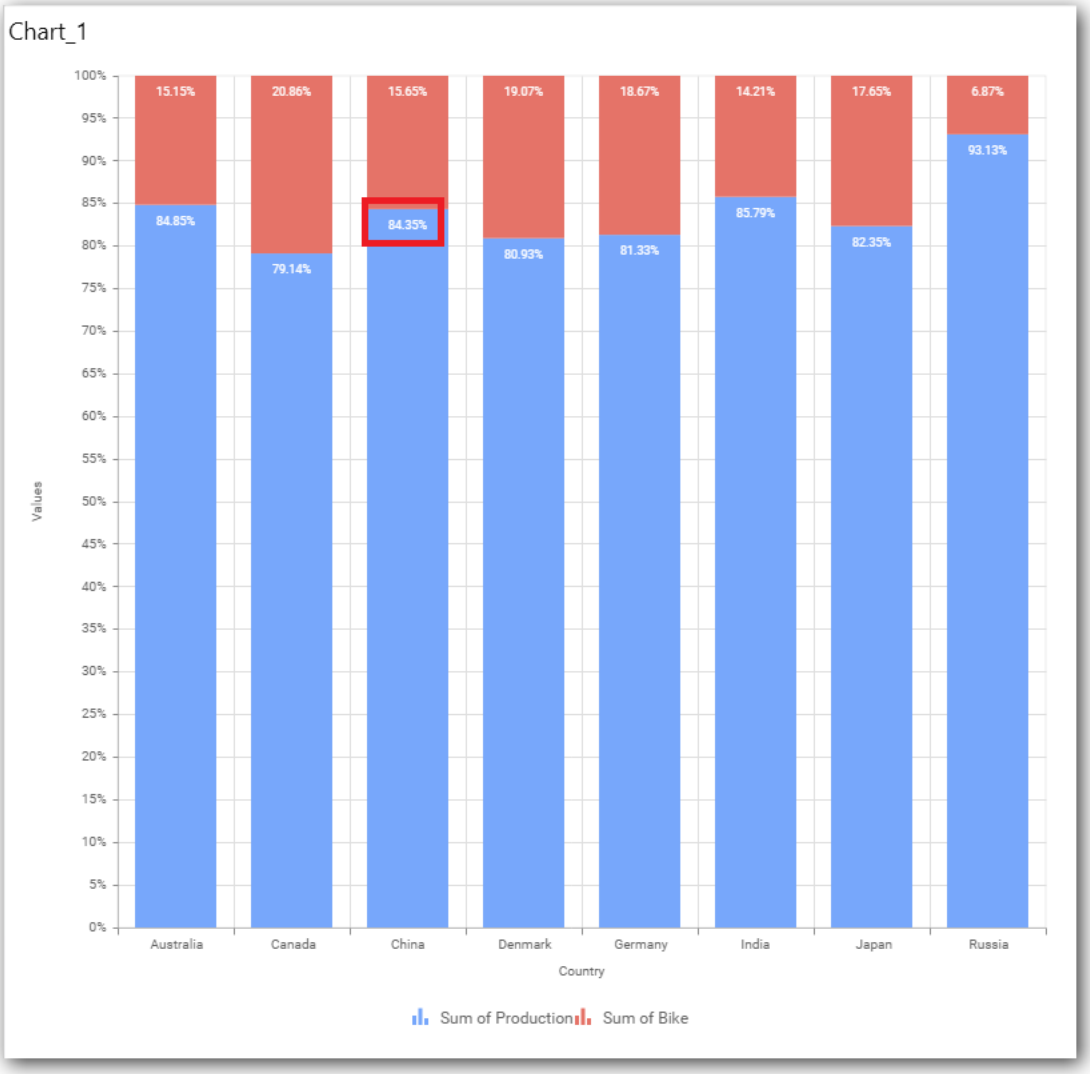


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



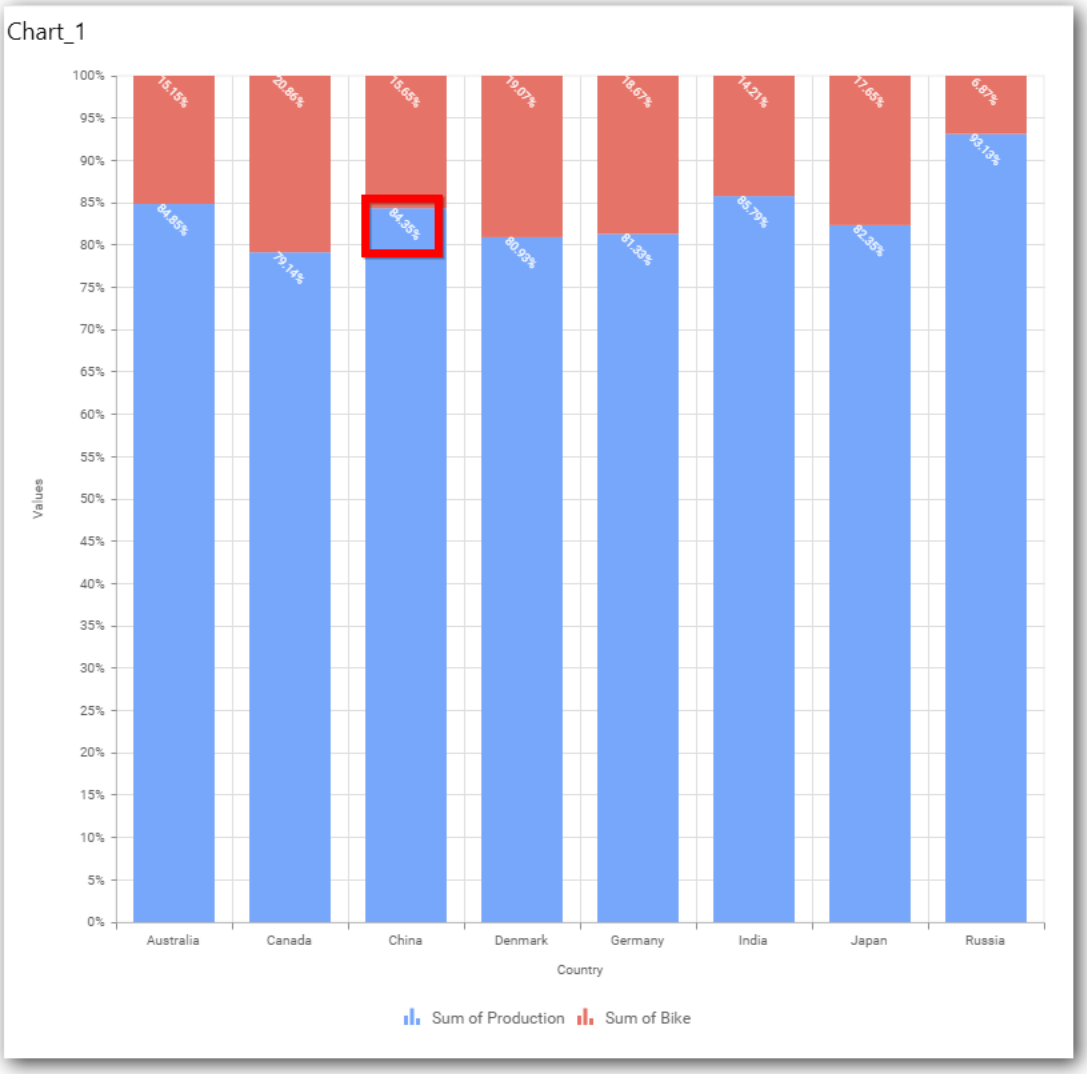
Show Value Labels

This allows you to toggle the visibility of value labels.



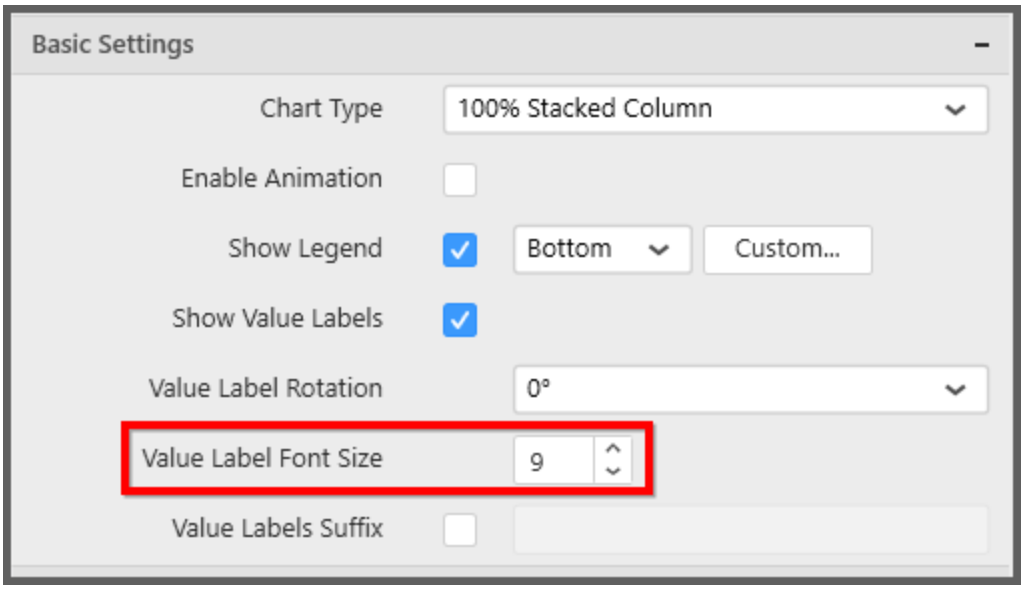
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



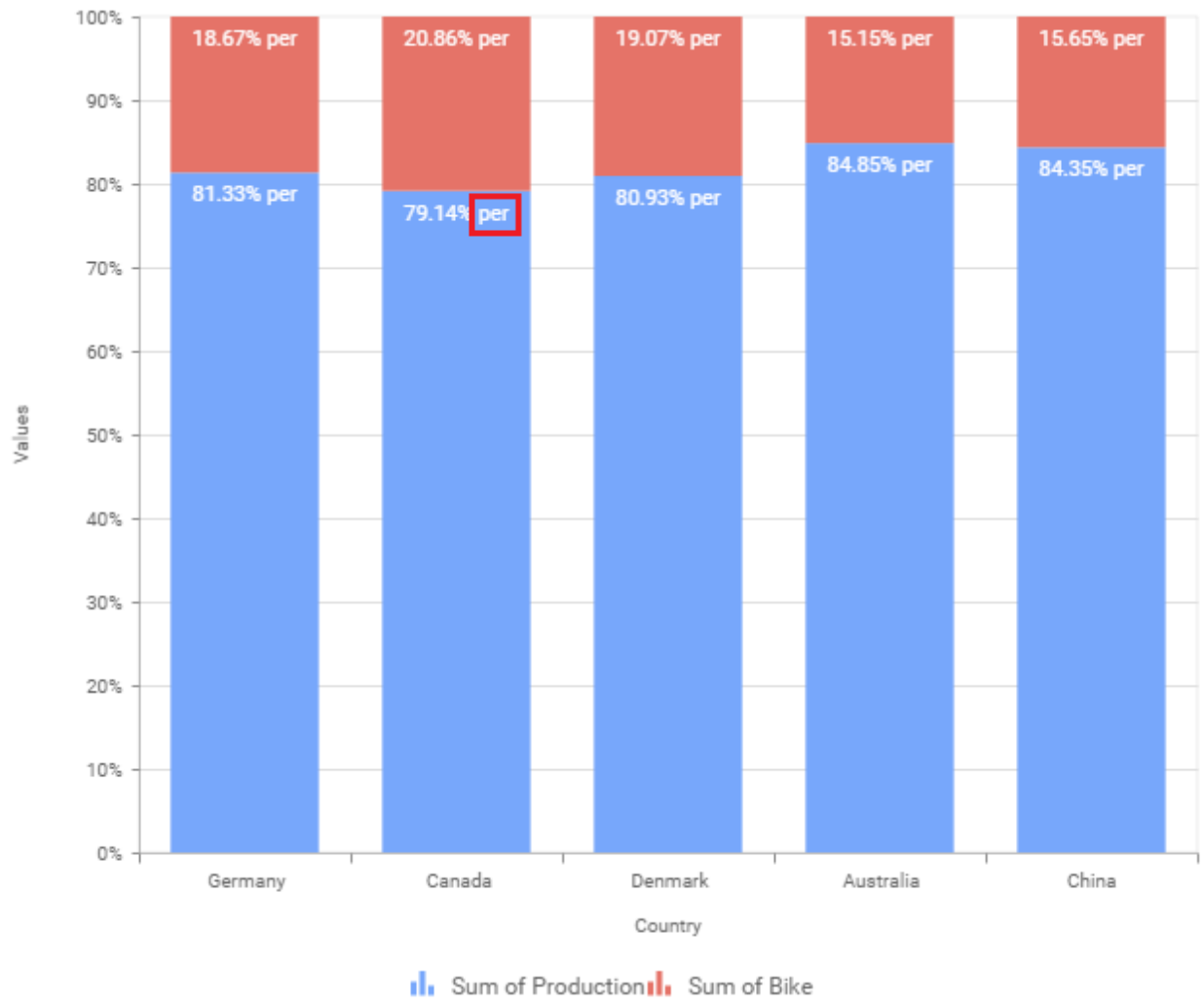
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

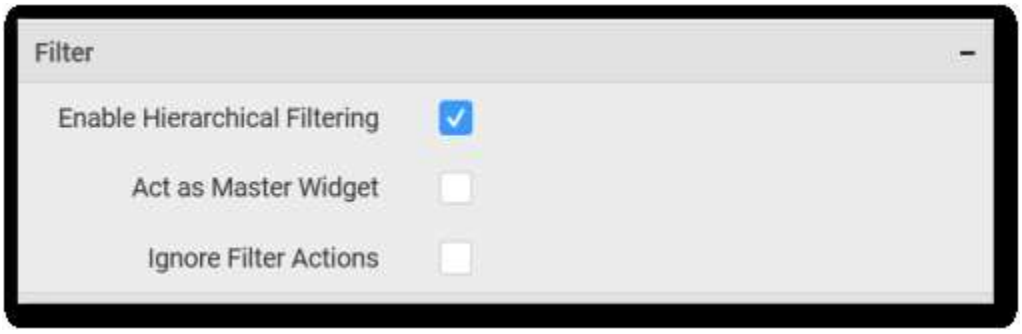


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

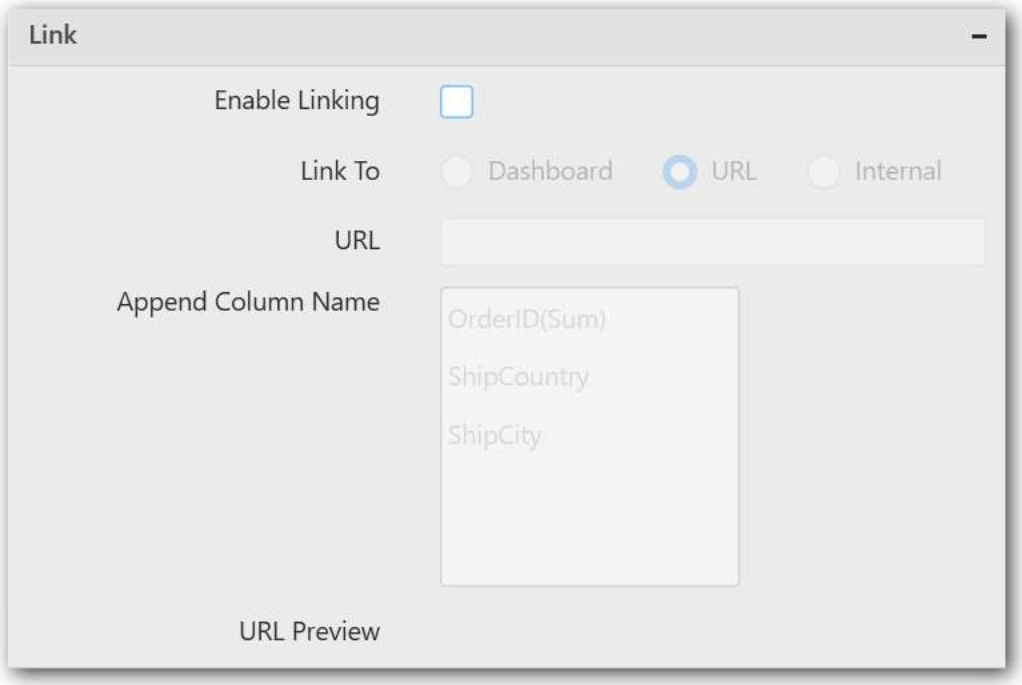
Act as Master Widget

This allows you to define this 100% stacked column chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

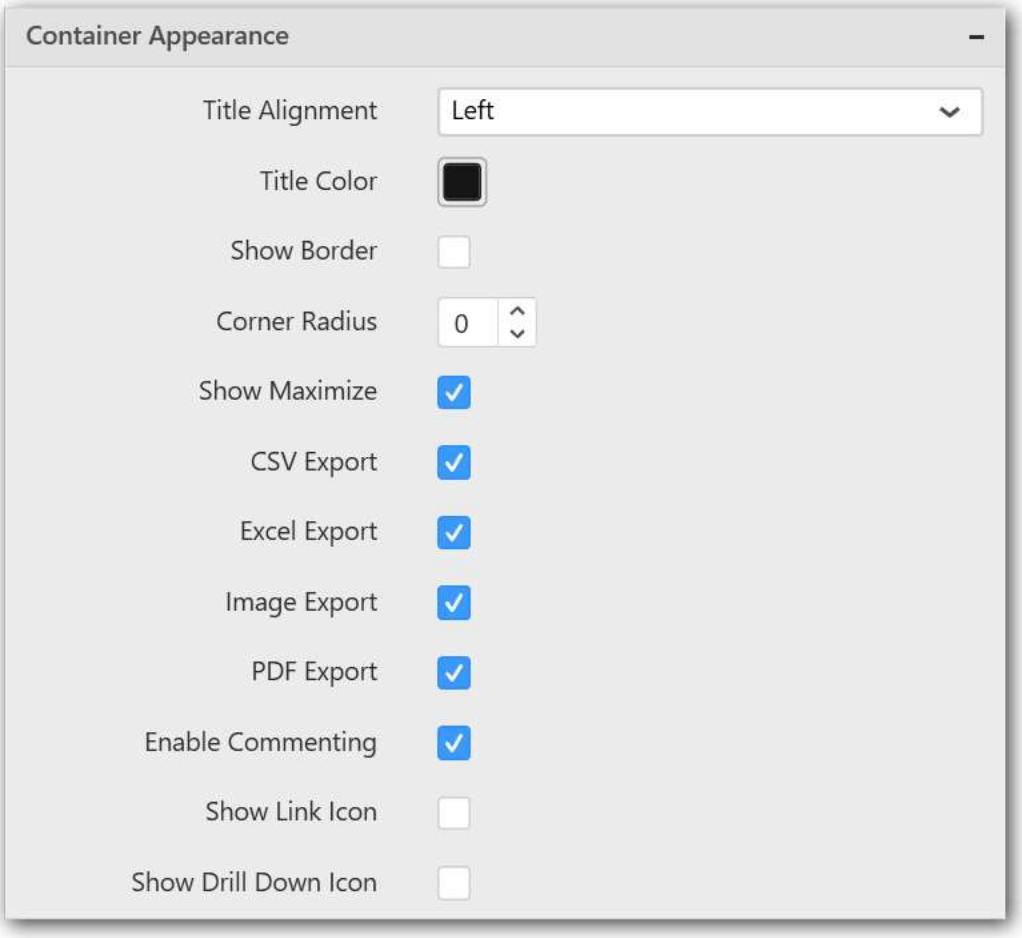
This allows you to define this 100% stacked column chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this 100% stacked column chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this 100% stacked column widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this 100% stacked column widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

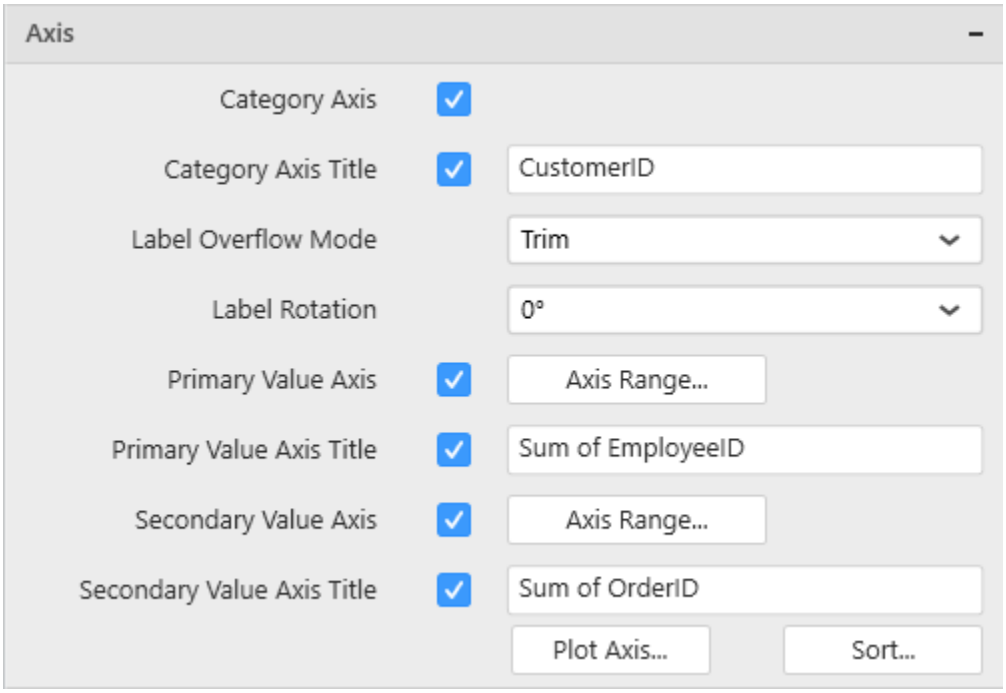
Image Export

This allows you to enable/disable the image export option for this 100% stacked column chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

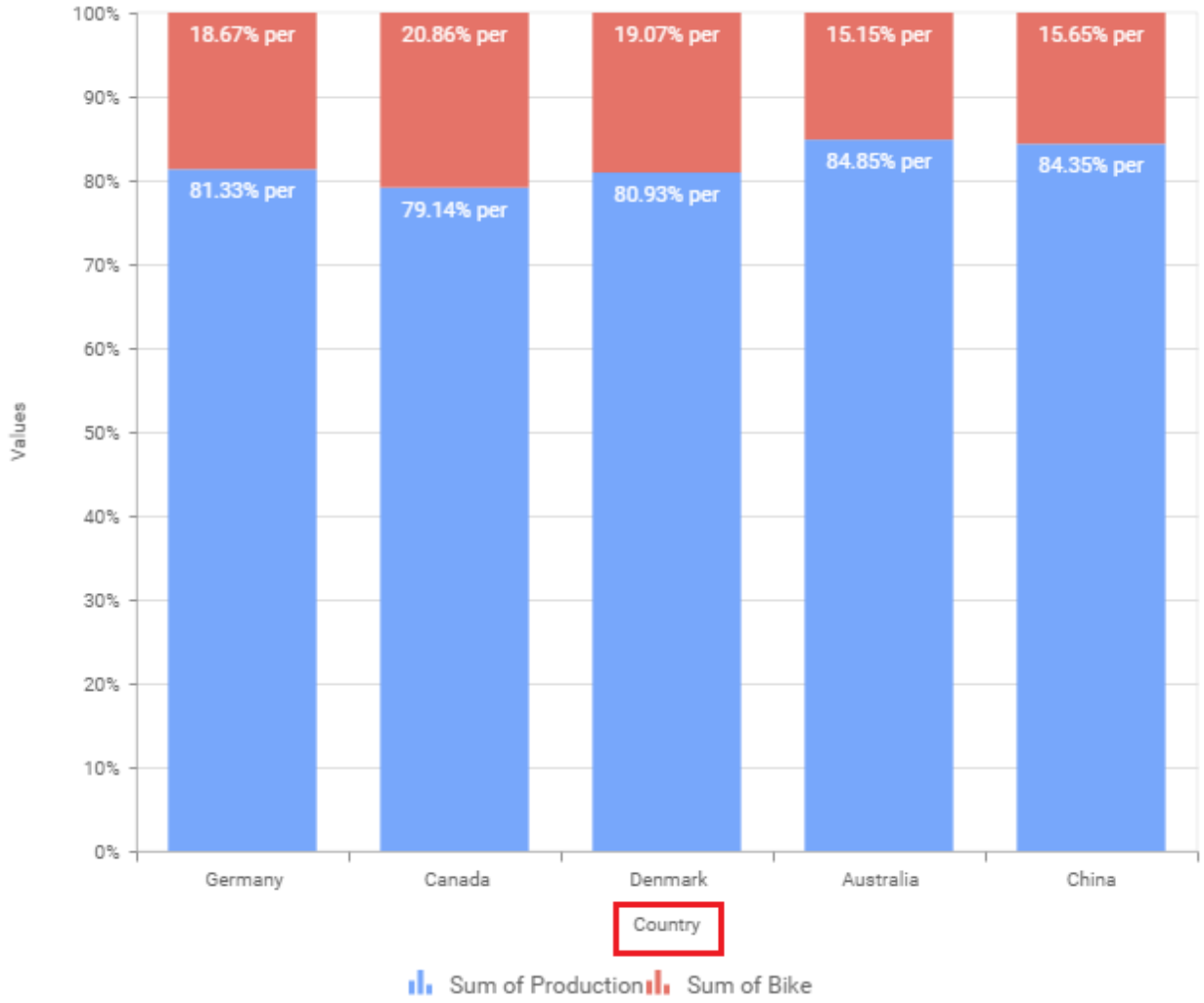
Axis Settings



This section allows you to customize the axis settings in chart.

Category Axis

This allows you to enable/edit the option of Category Axis. It will reflect in chart area x-axis name.



Category Axis Title

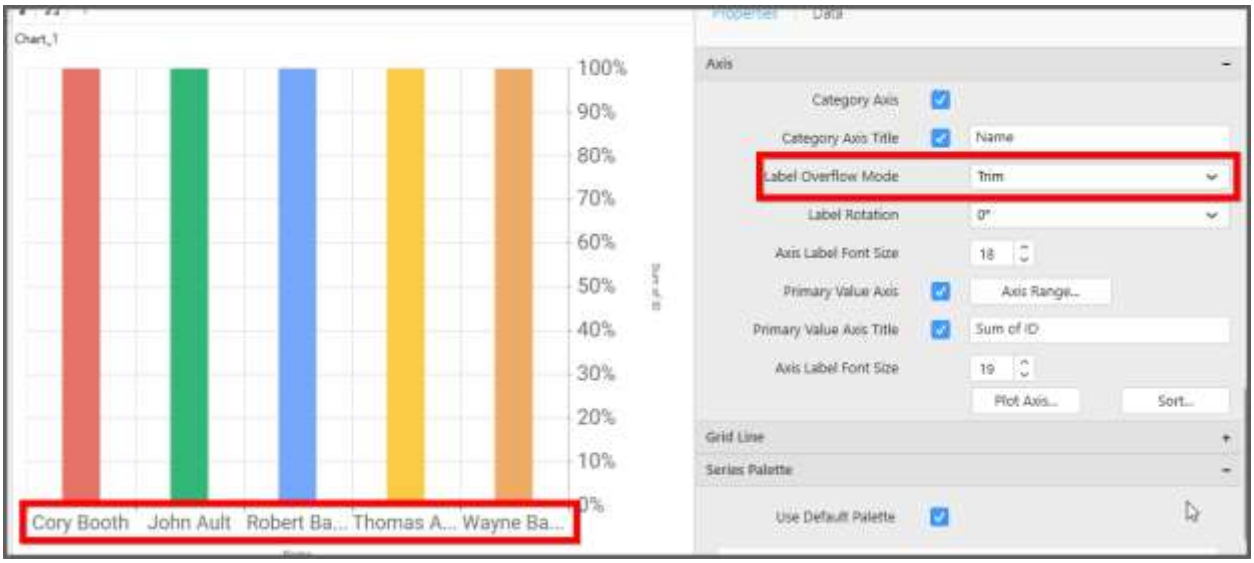
This allows you to toggle the visibility of Category axis title.

Label overflow mode

This allows you to handle the display mode of the overlapping labels.

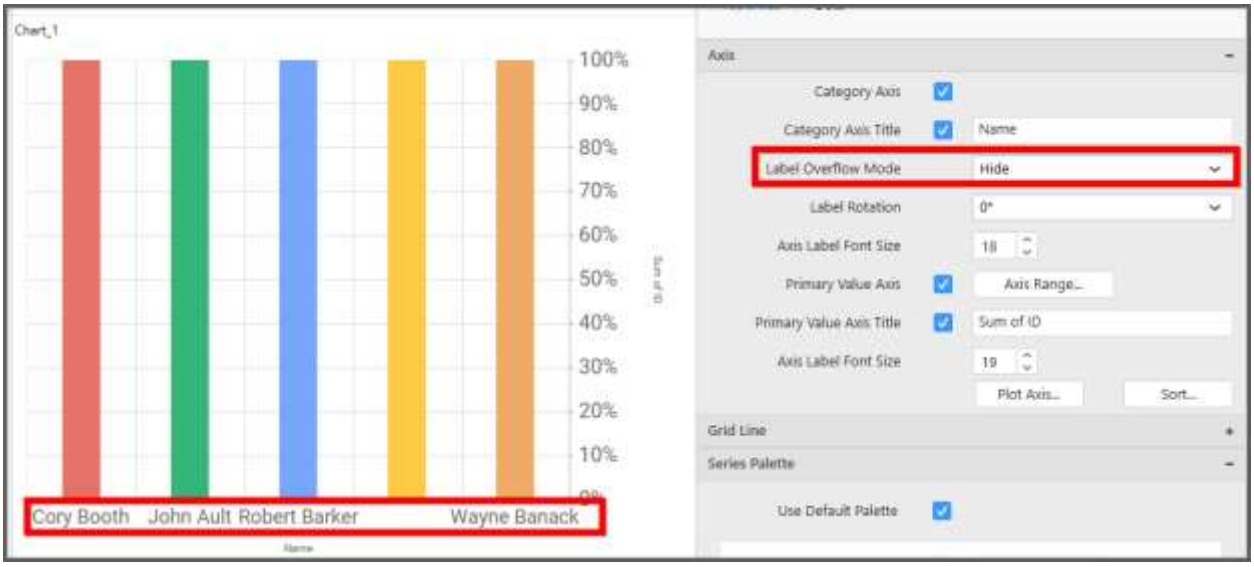
Trim

This option trims the end of overlapping label in the axis.



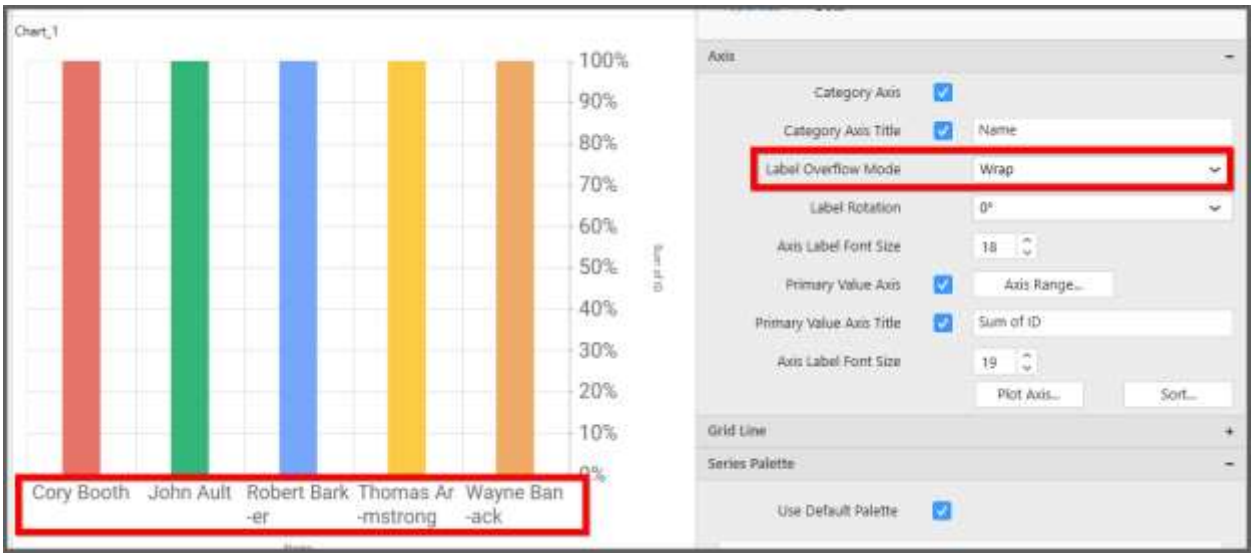
Hide

This option hides the overlapping label in the axis.



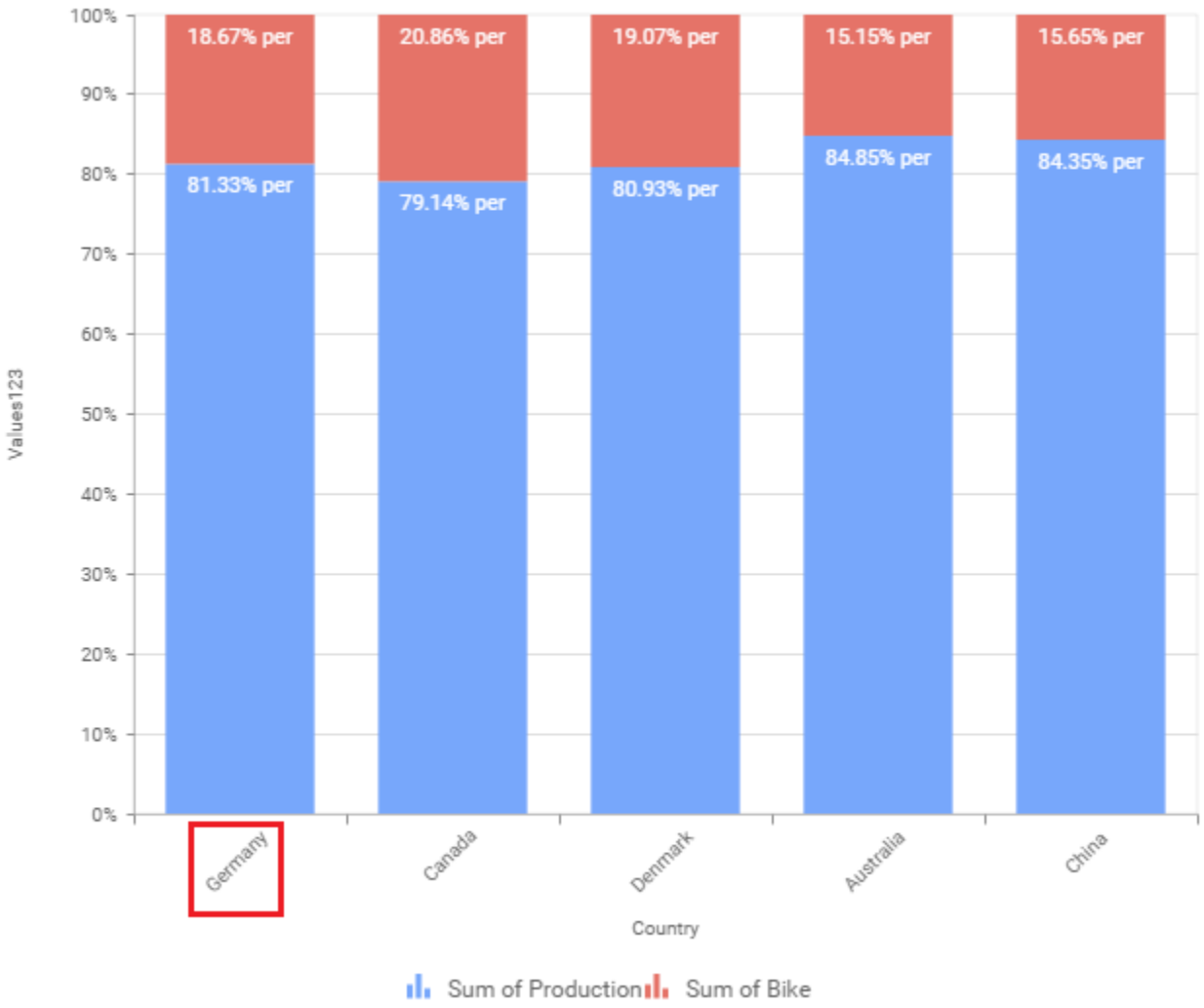
Wrap

This option wraps the lengthy label text in the axis.



Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

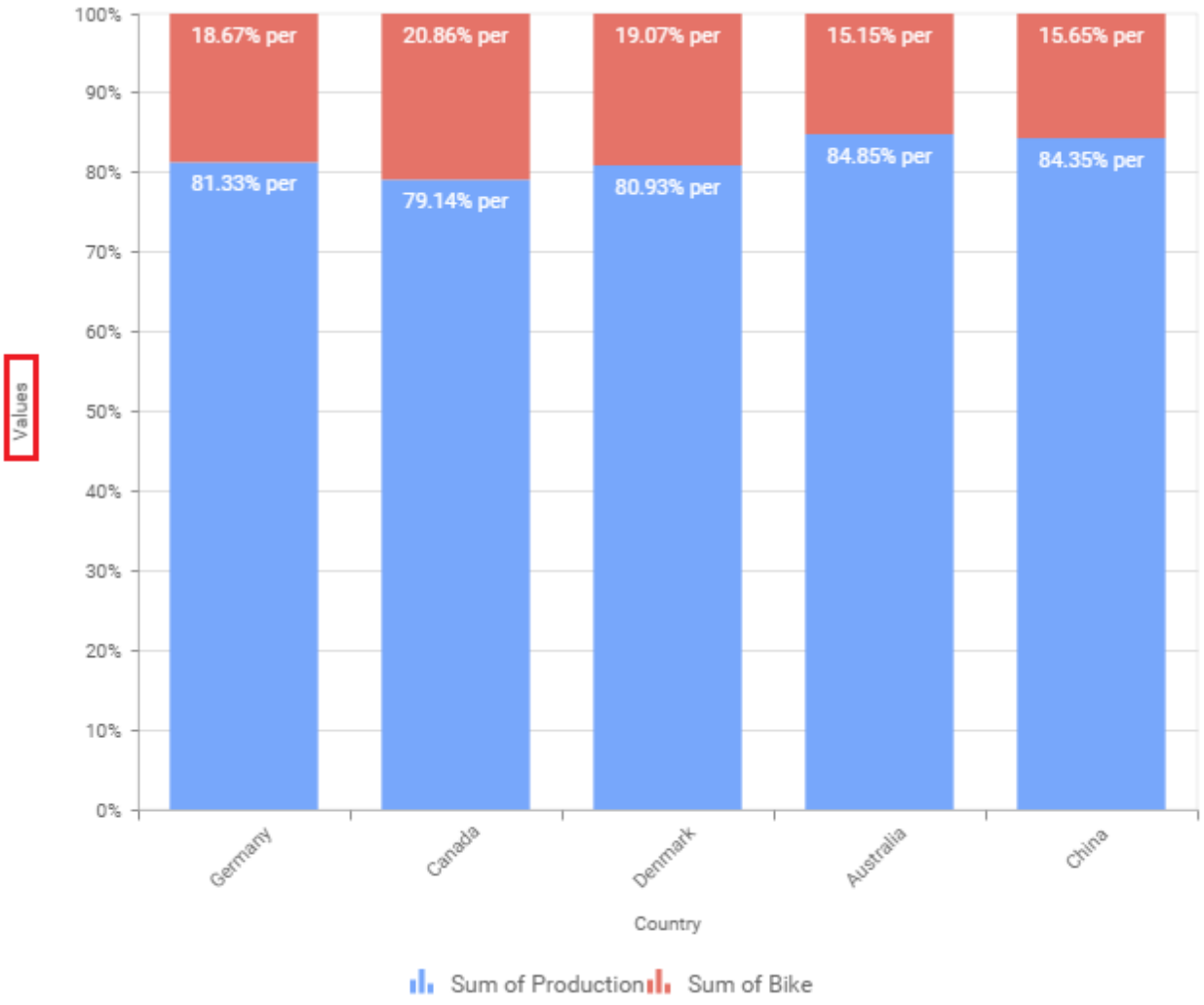


Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.

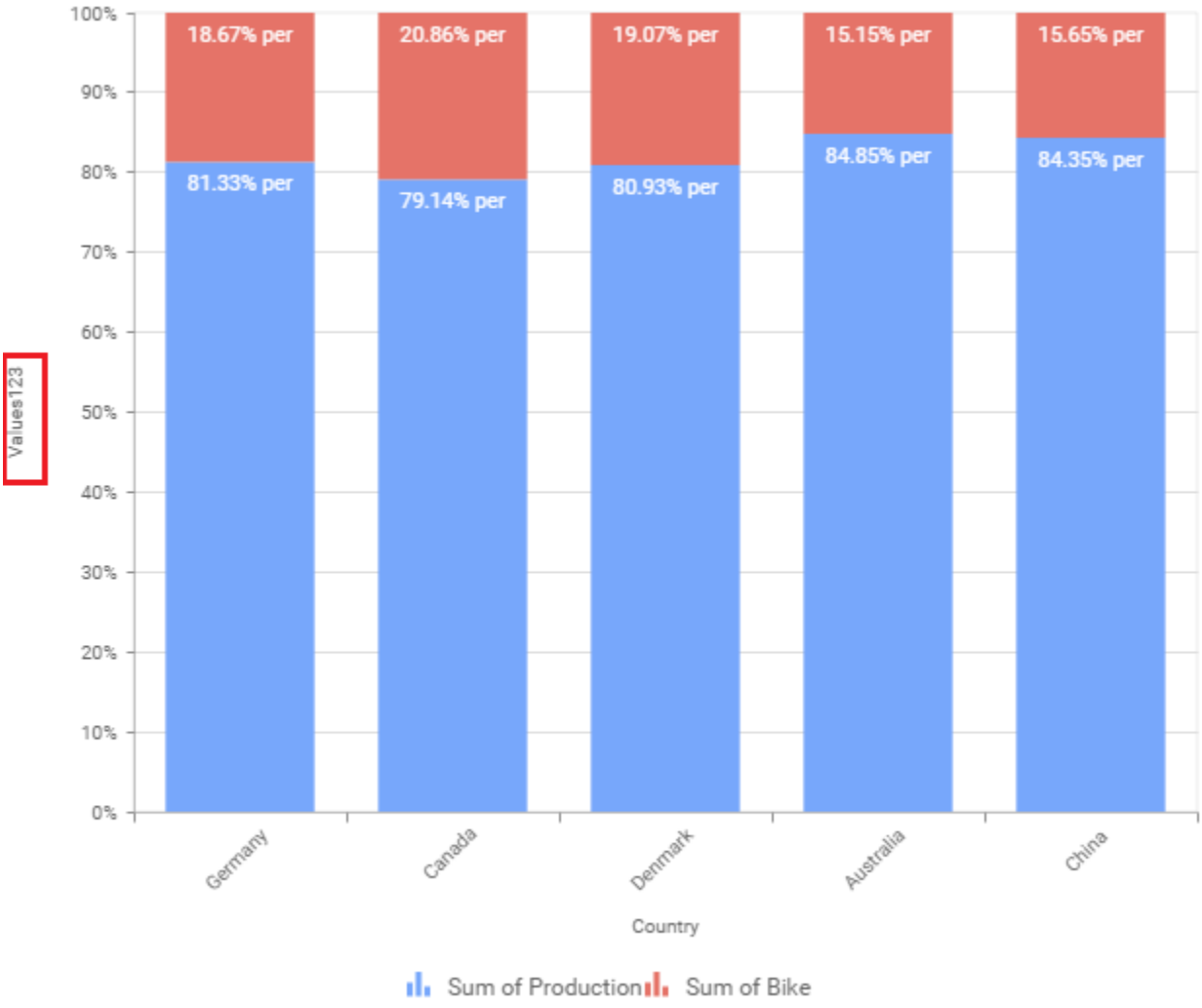
Primary Value Axis

This allows you to enable/edit the Primary Value Axis title. It will reflect in chart area y-axis name.



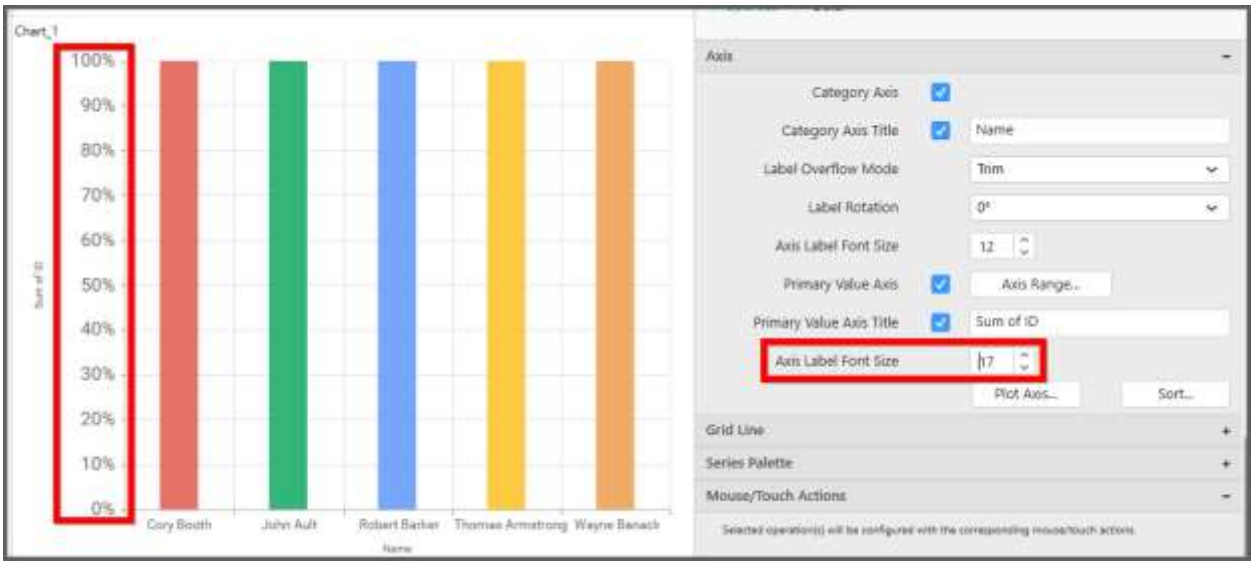
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

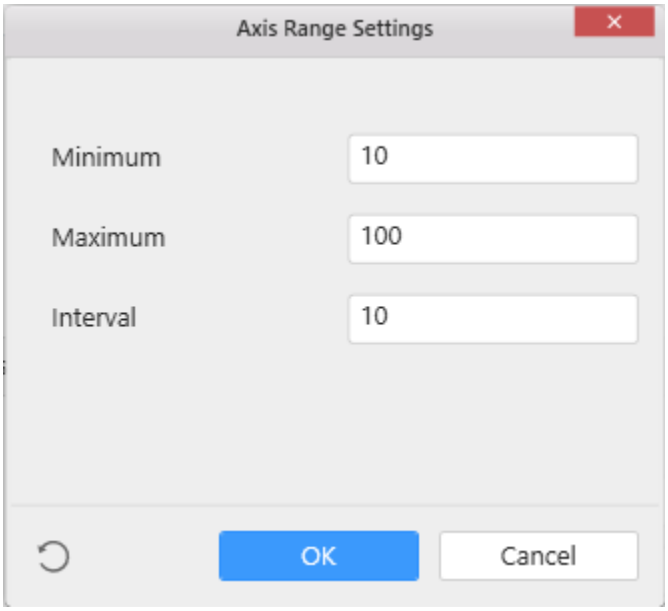


Primary Value Axis Range

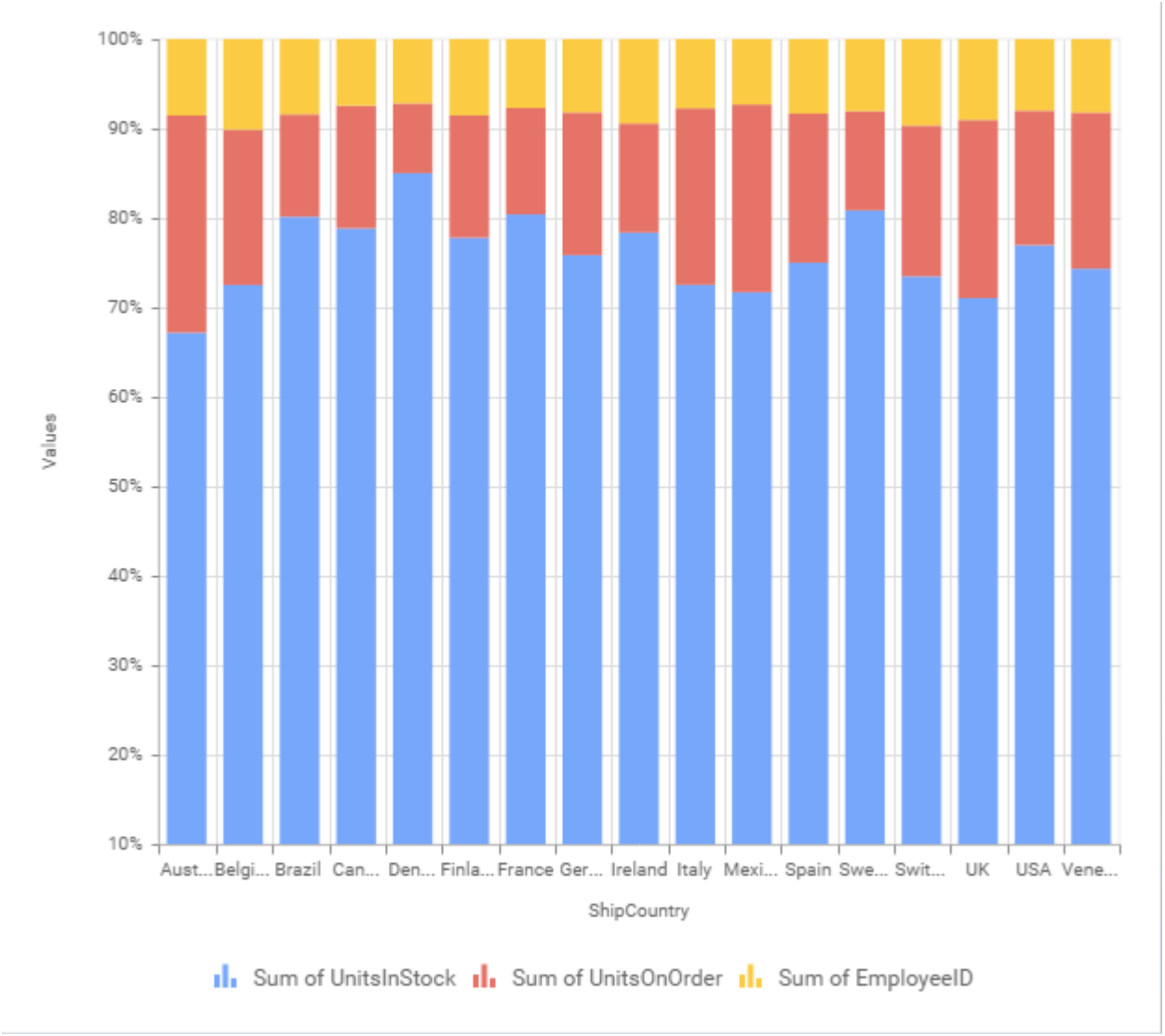
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.

Secondary Value Axis Title

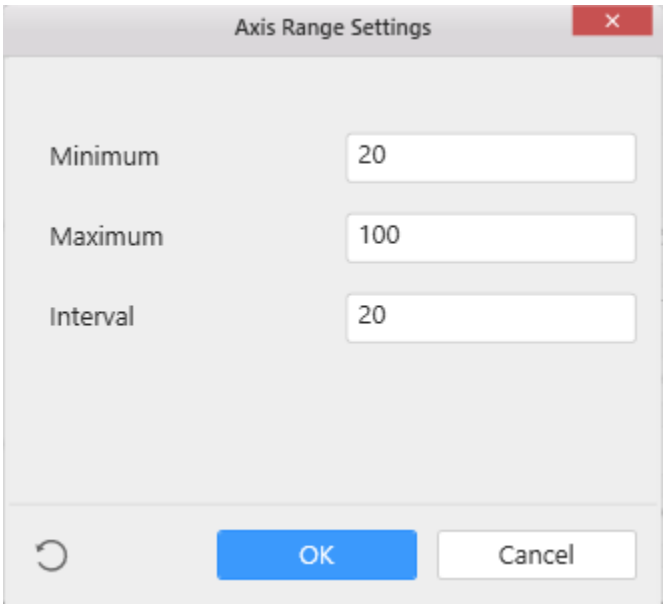
This allows you to toggle the visibility of secondary value axis title.

Axis Label Size

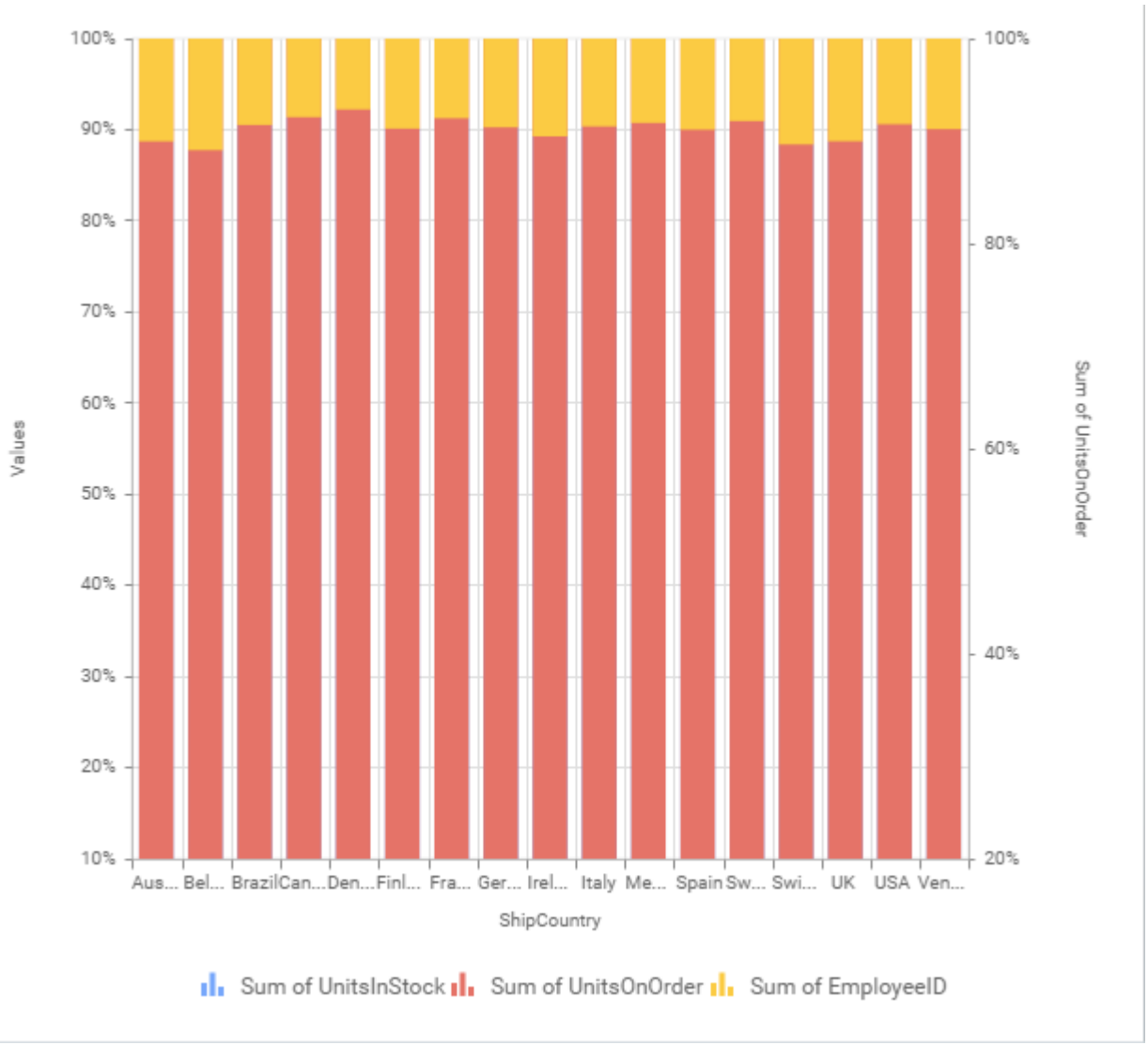
This allows you to increase or decrease the font size of the secondary axis label. Default font size for the secondary axis label was 10 pixels.

Secondary Value Axis Range

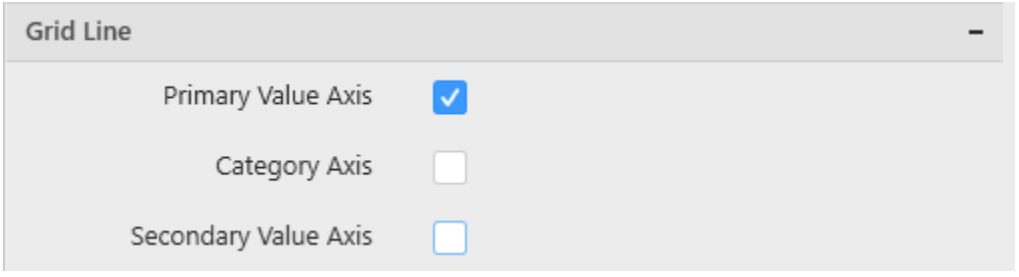
This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

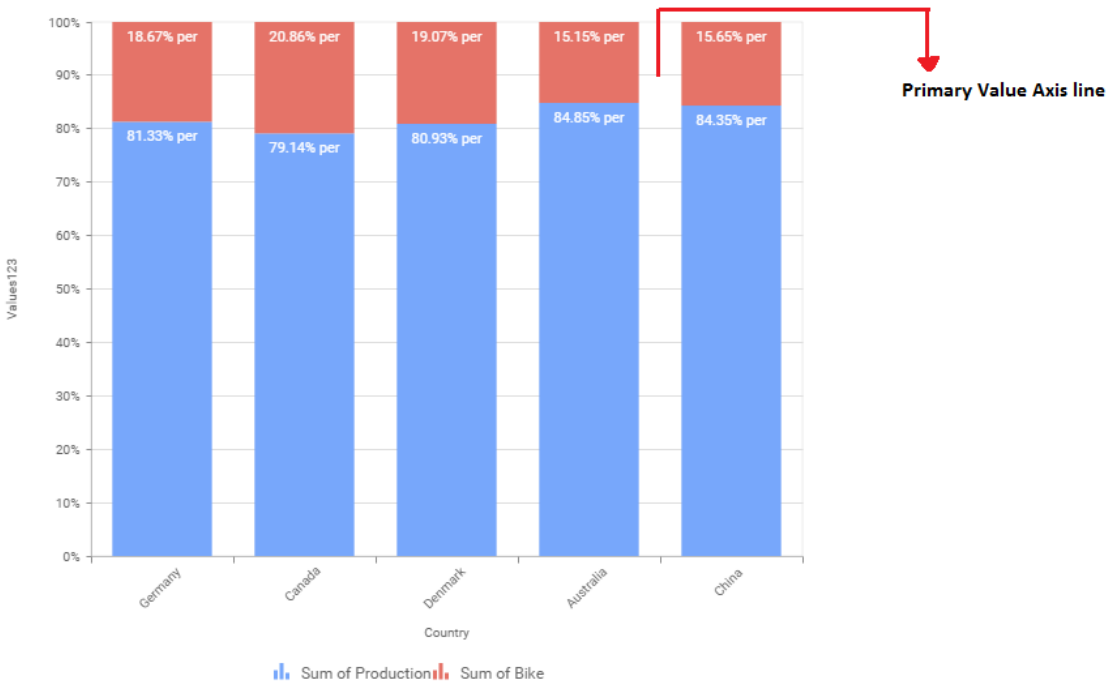


Grid Line Settings



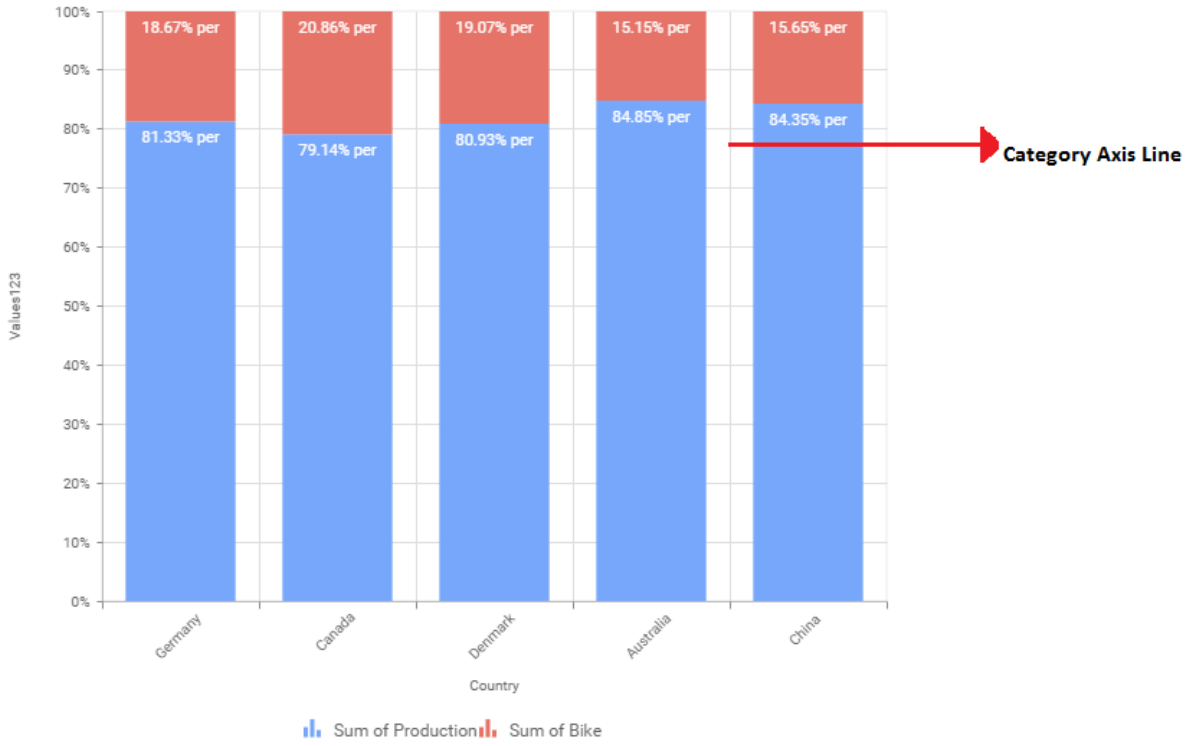
Primary value Axis

This allow to enable the primary value axis' gridlines for the 100% stacked column chart.



Category Axis

This allows you to define the visibility of Category axis' gridlines.



Area Chart

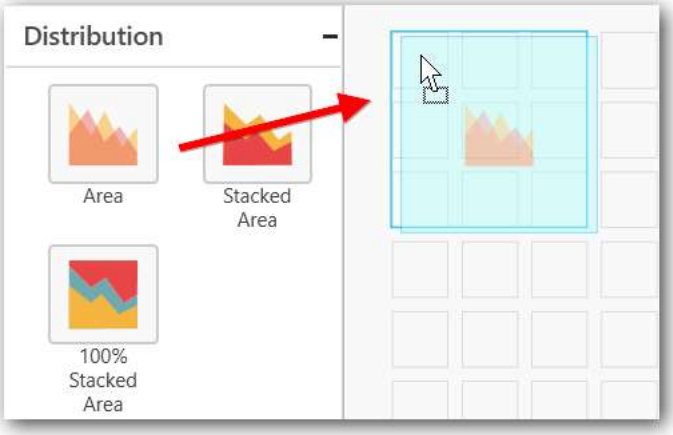
Area Chart allows you to compare values for a set of unordered items across categories through filled curves ordered vertically.

How to configure flat table data to Area Chart?

Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following chart illustrates about how to configure data to area chart

Drag and drop the **Area** chart widget into canvas and resize into your required size.

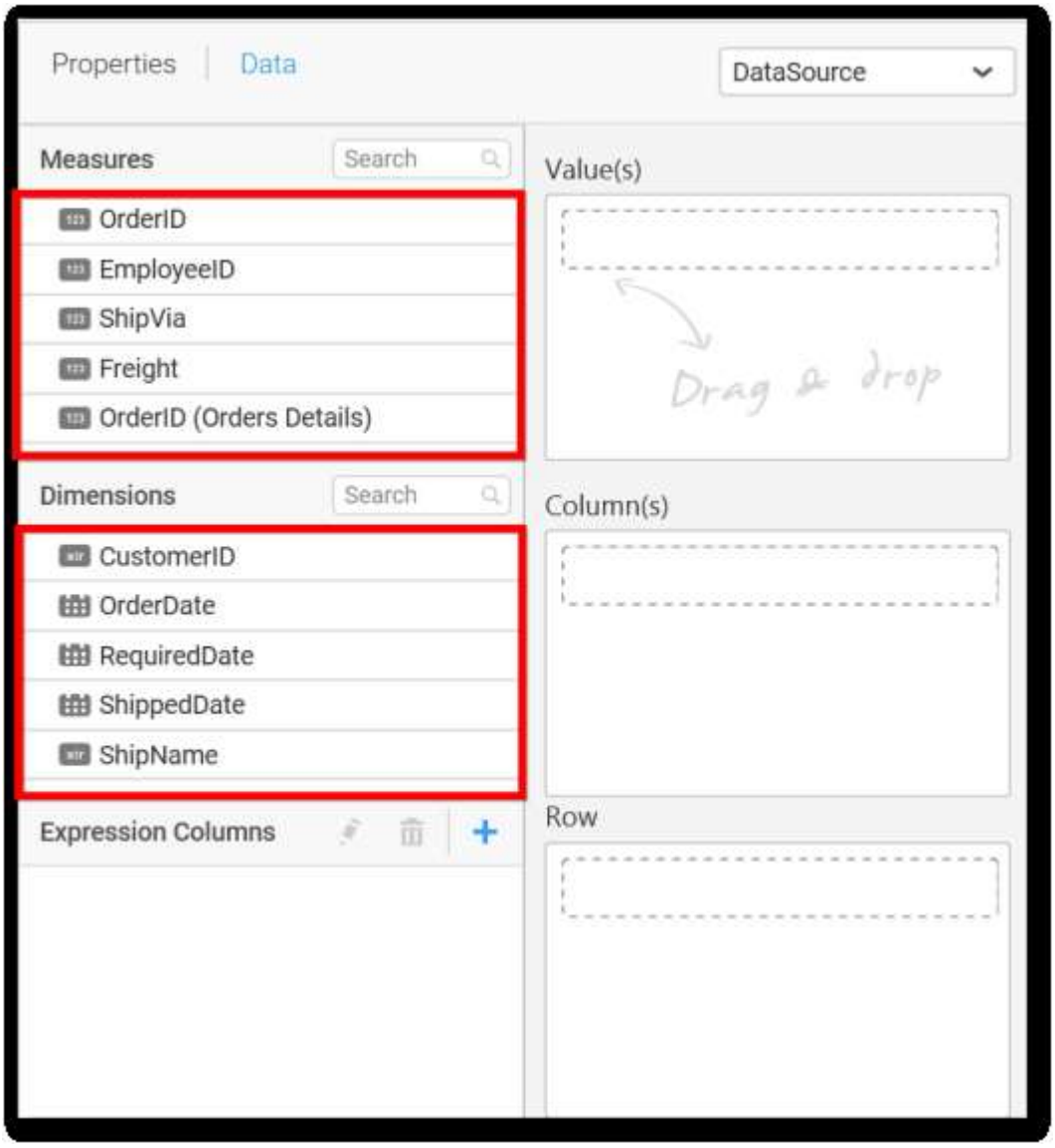


Connect to the data source.

Focus on the Chart widget.

Click on the Assign Data Button.

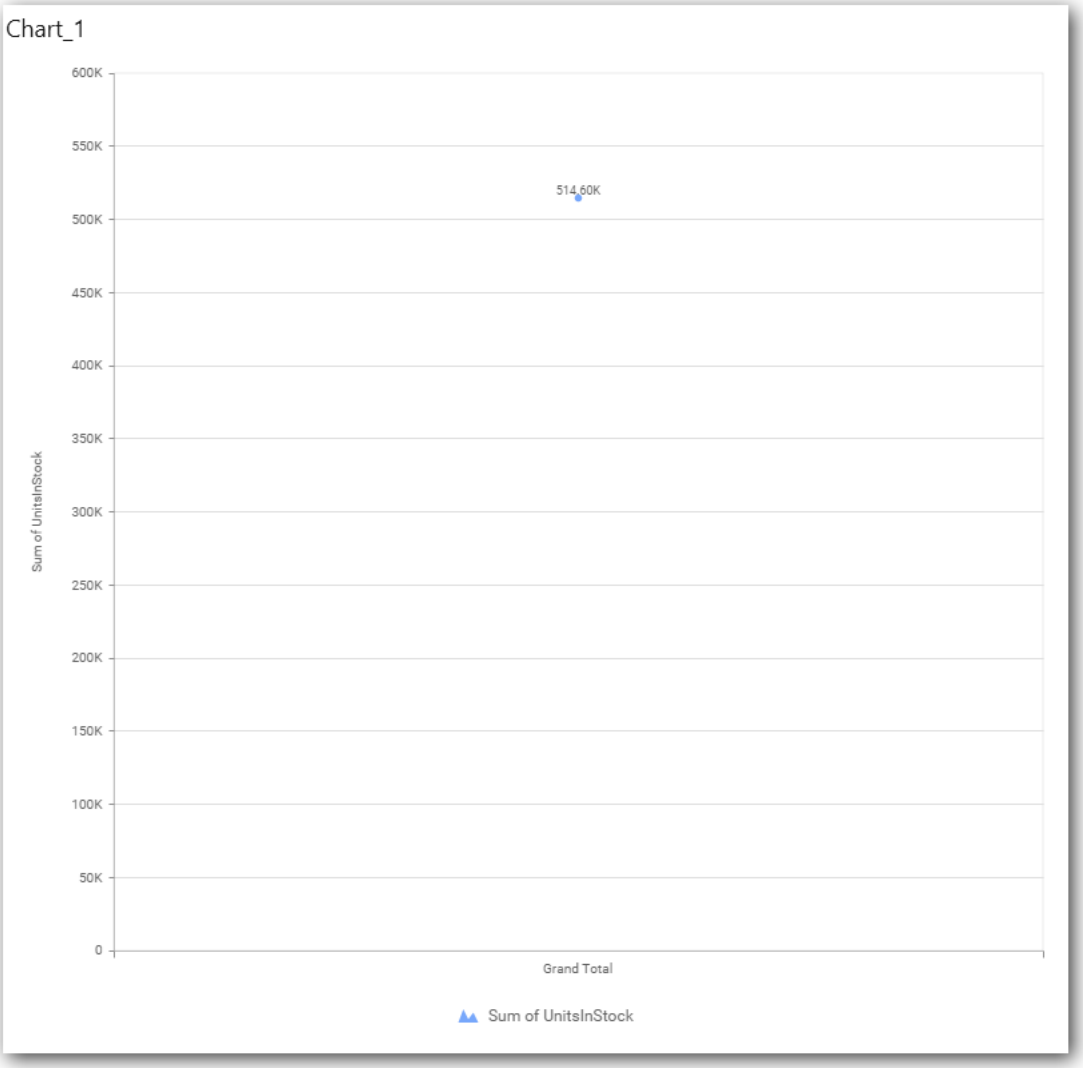
A Data pane will be opened with available Measures and Dimensions.



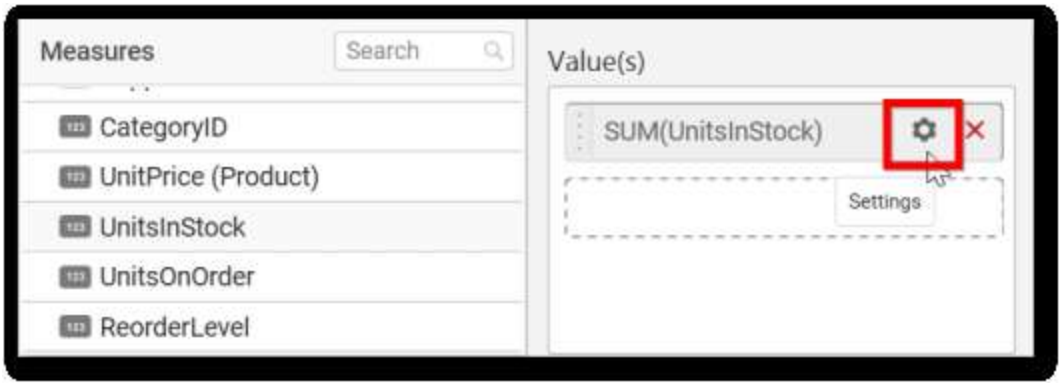
Assigning Value(s)

Drag and drop the Measure into Value.

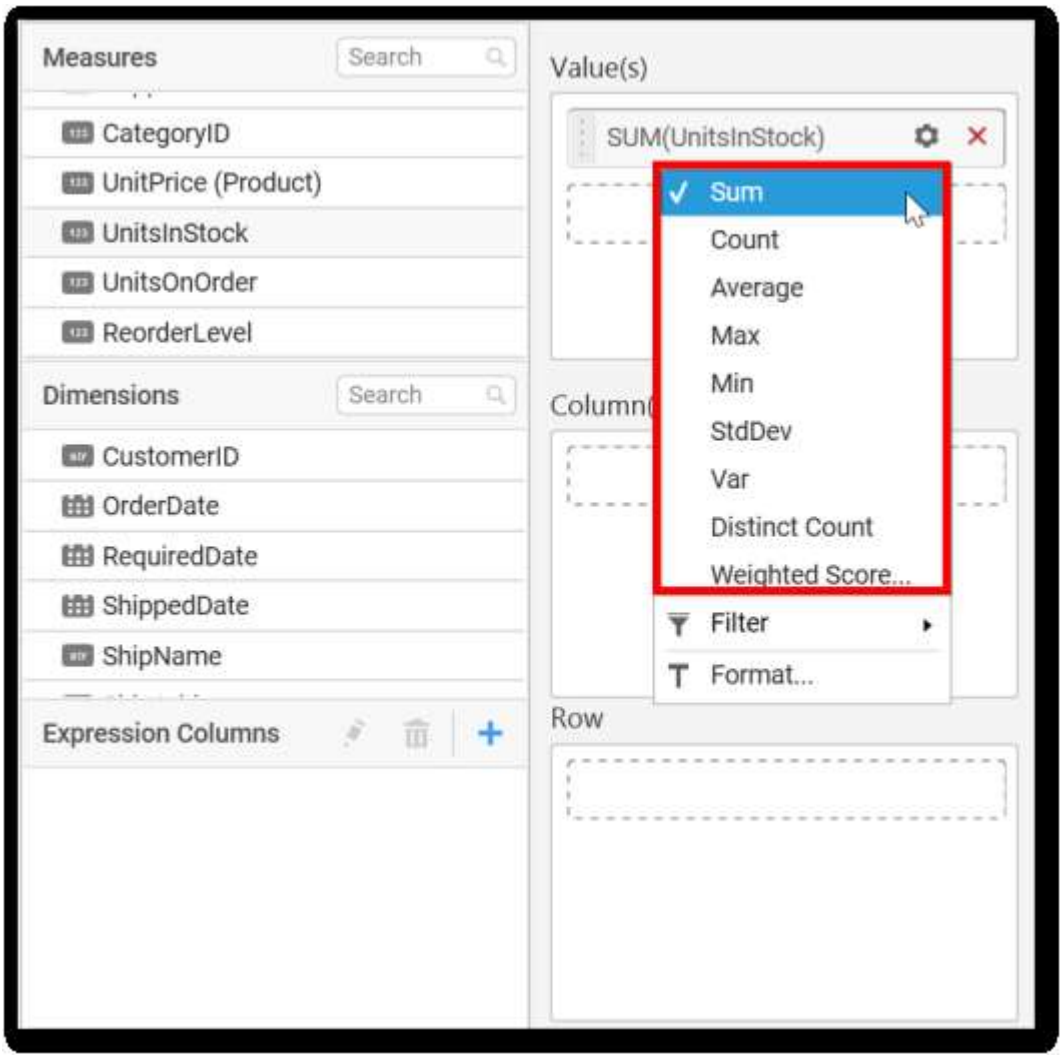
Now the chart will be rendered like this.



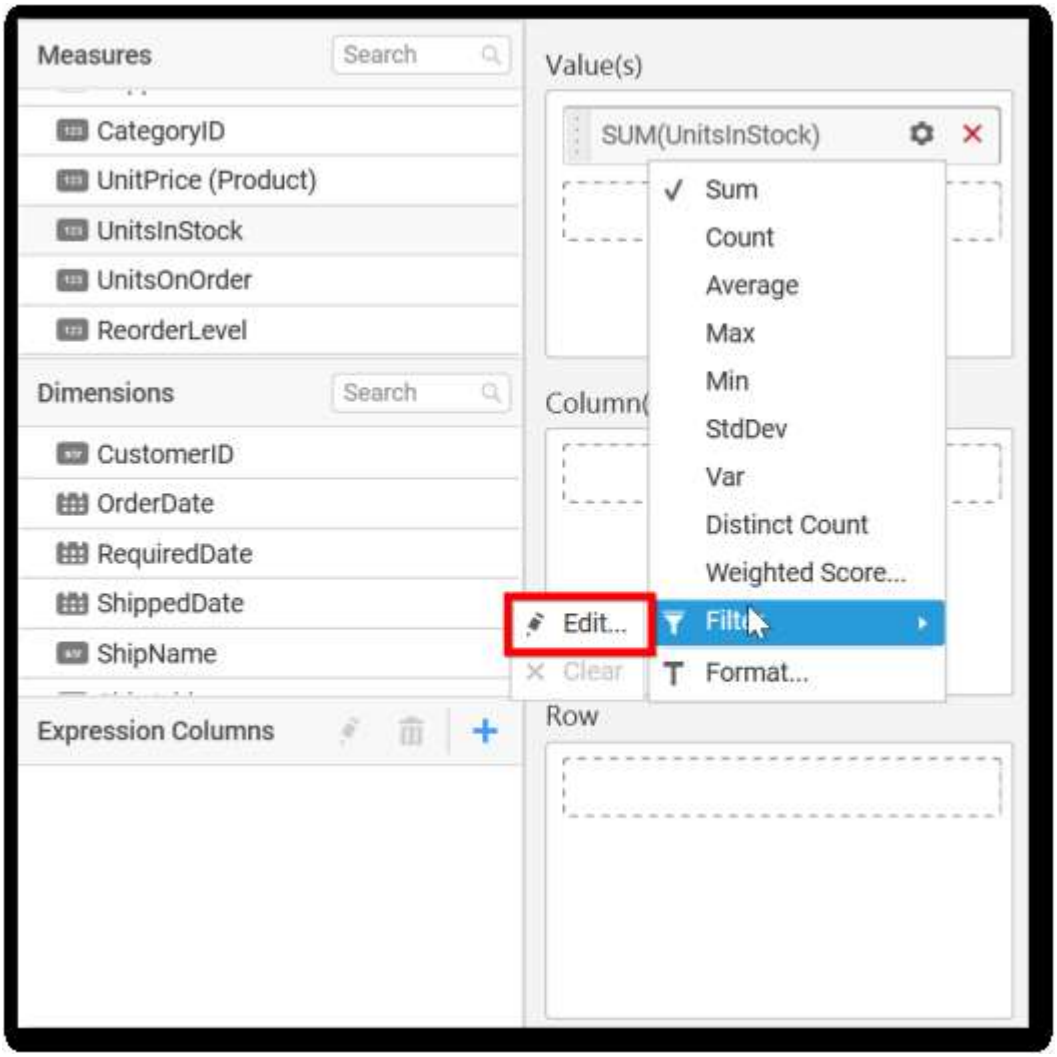
You can change the summary type of the value by clicking on Settings option.



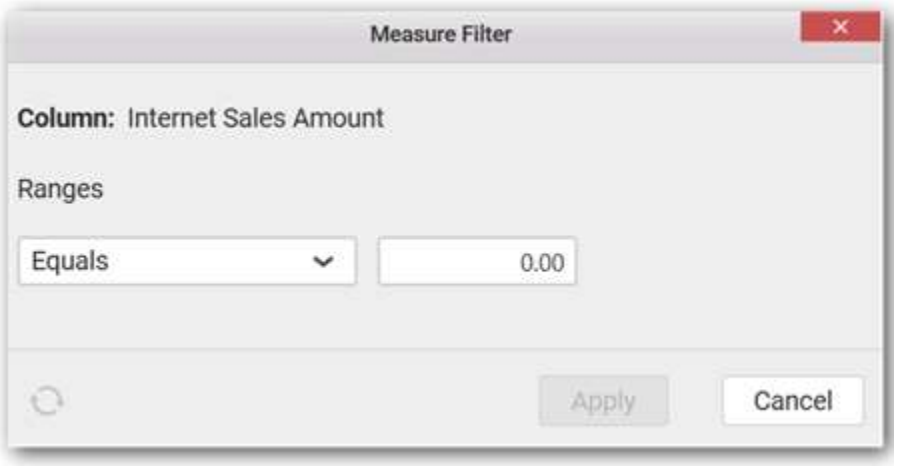
Select the required summary type from list.



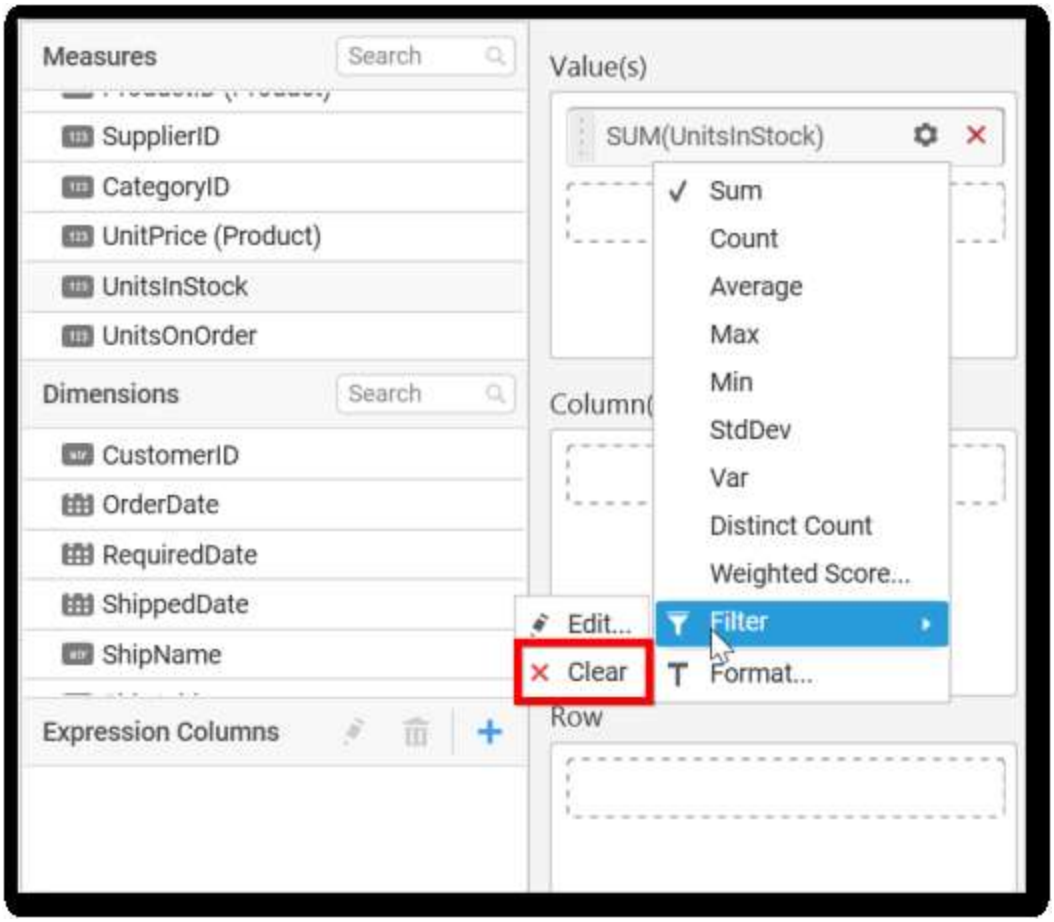
You can select what data to be displayed by choosing filter option.



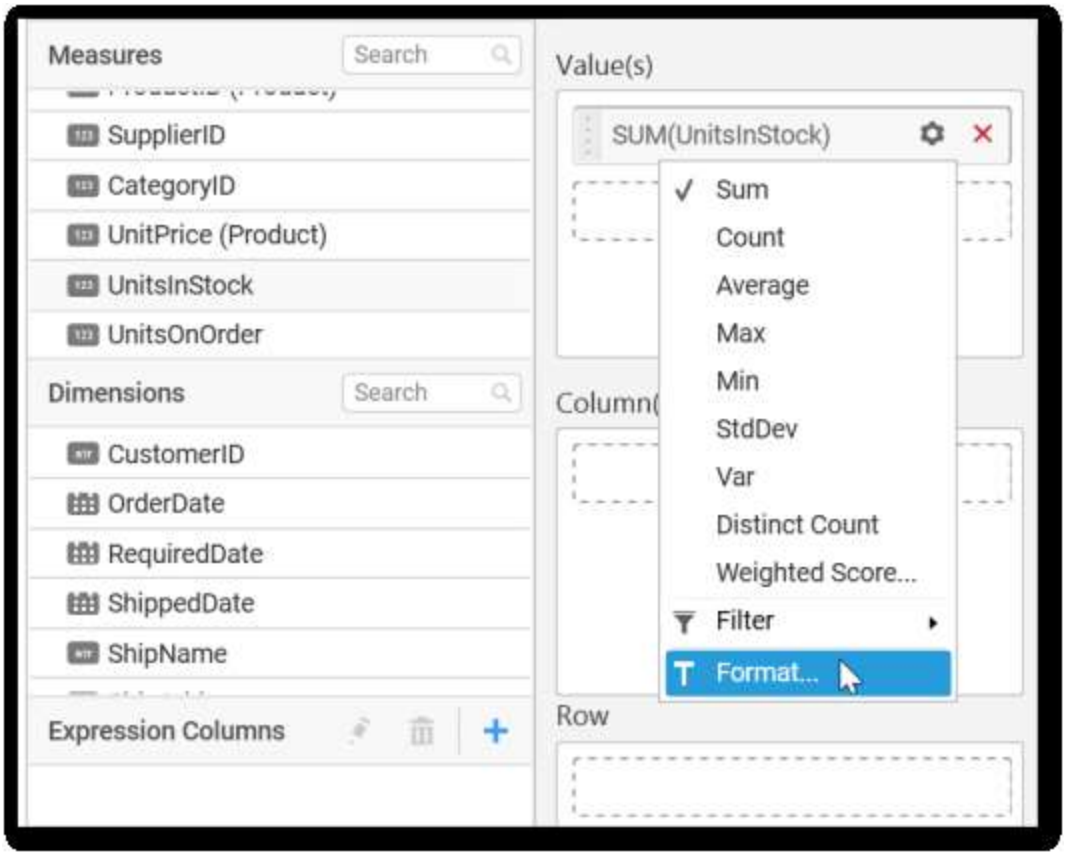
The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.



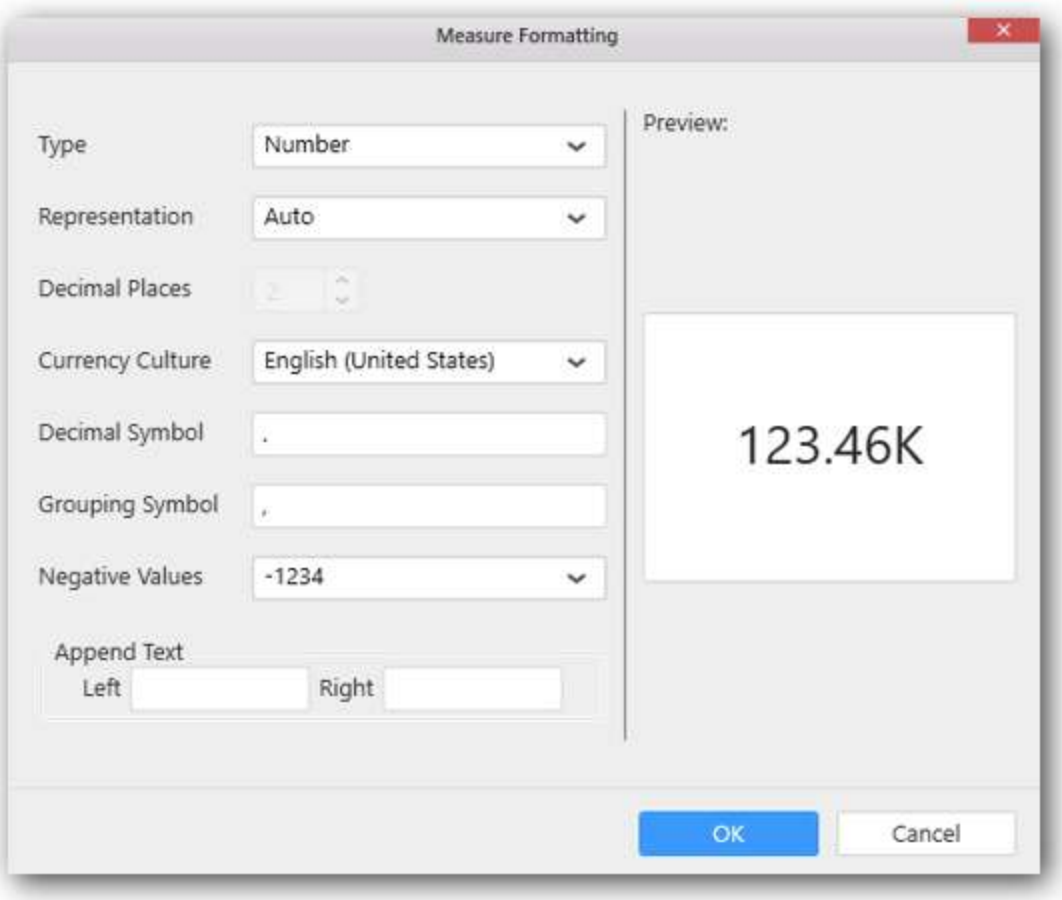
You can clear the filter.



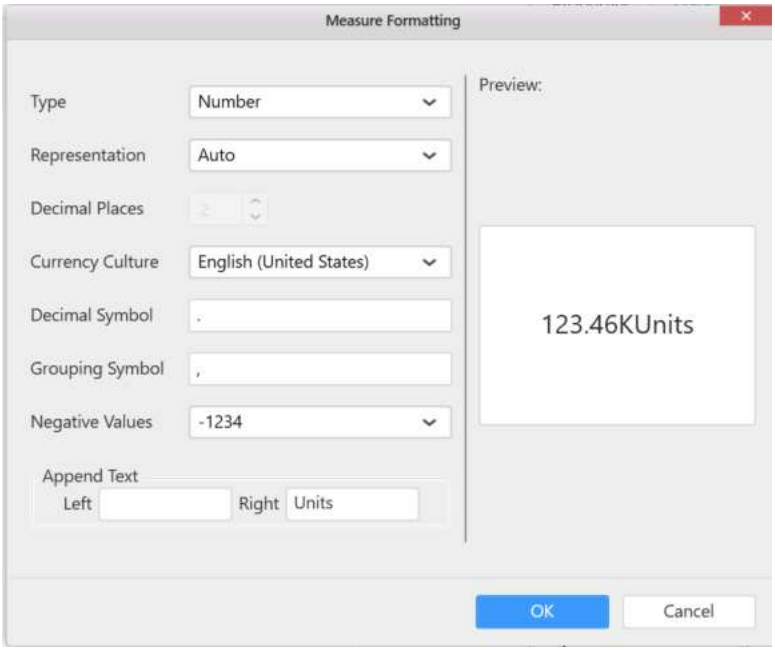
You can Format the value.



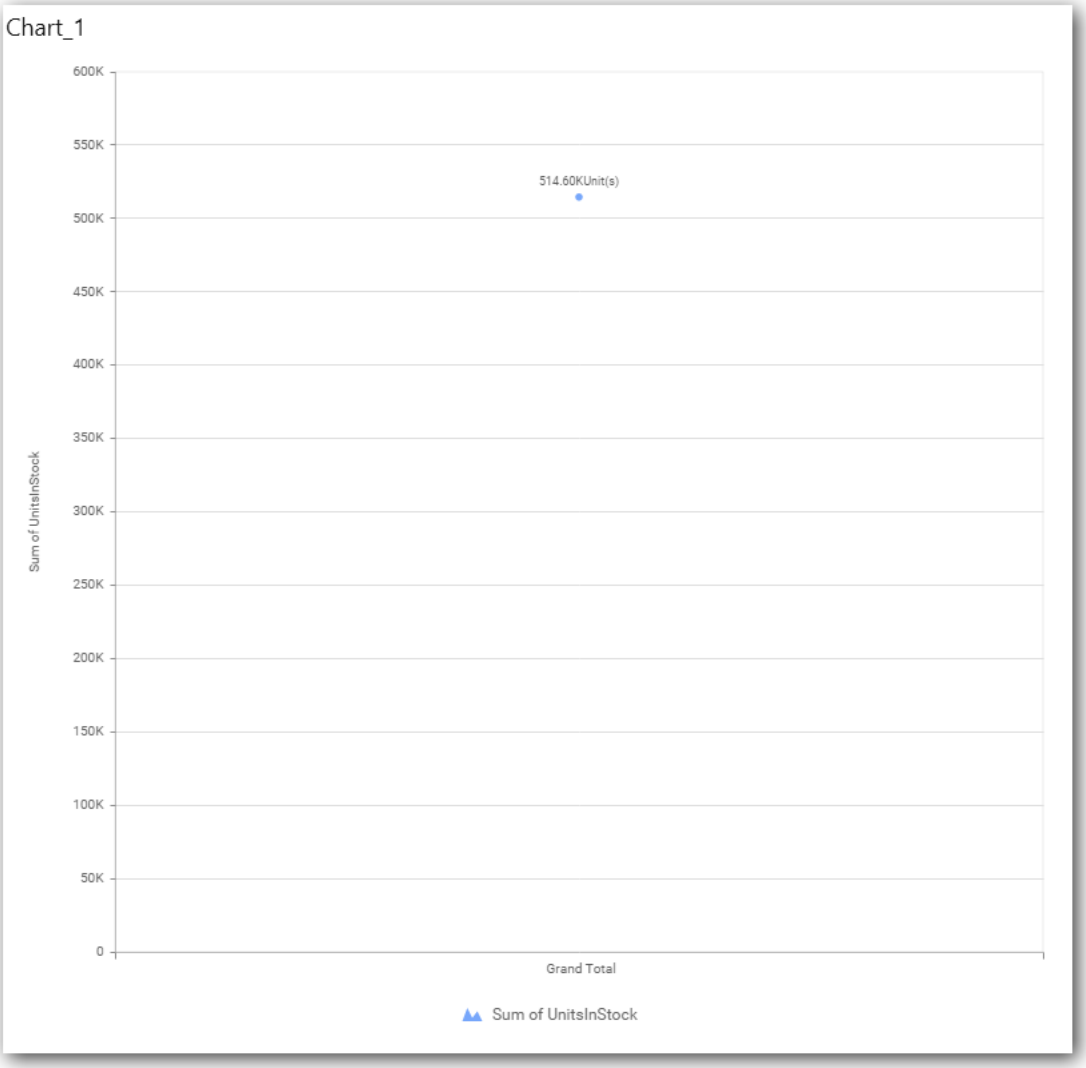
The format options will be shown.



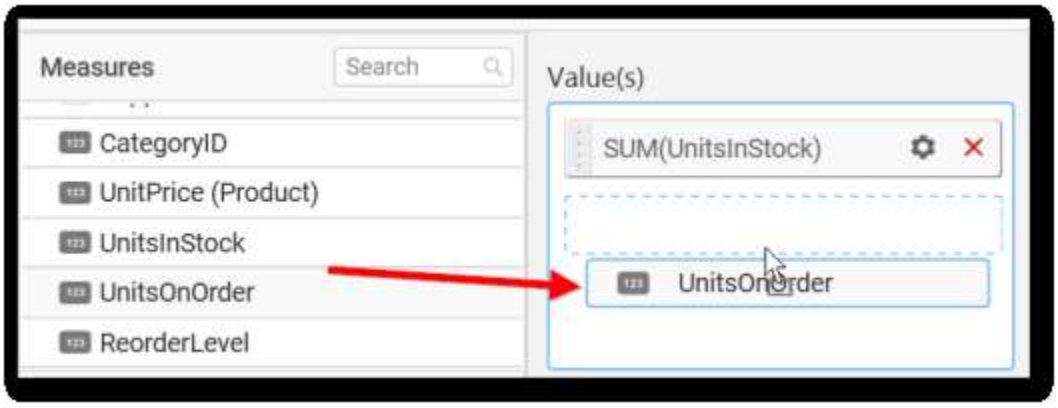
Choose the options you need and click OK.

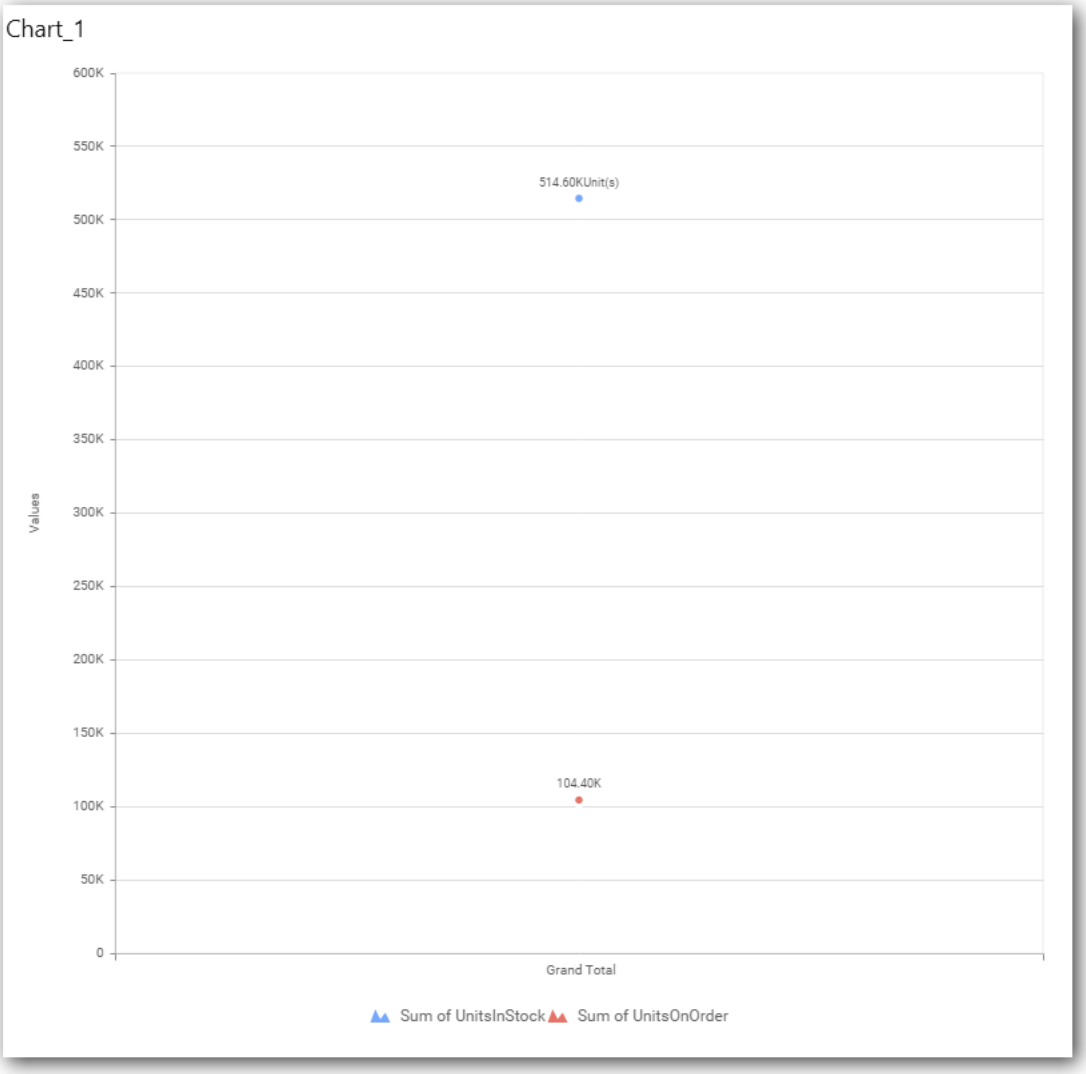


Now the Chart will be rendered like this.



You can add more number of values by drag drop the Measures into Value field.

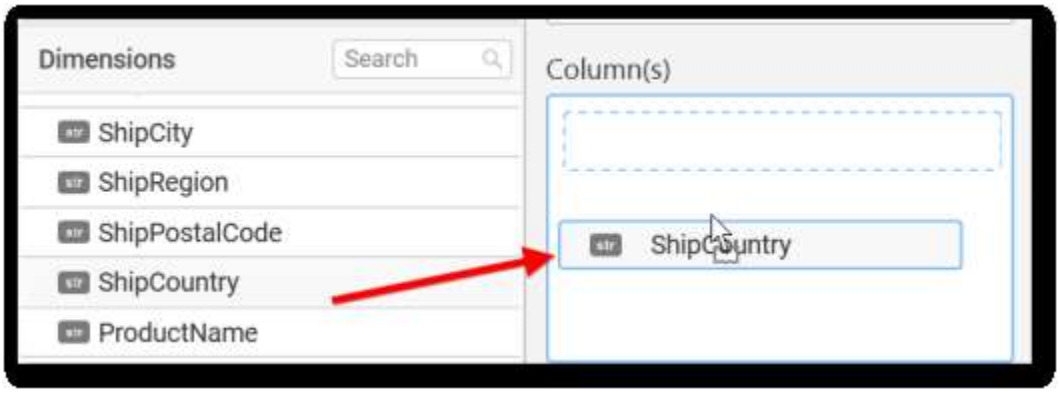


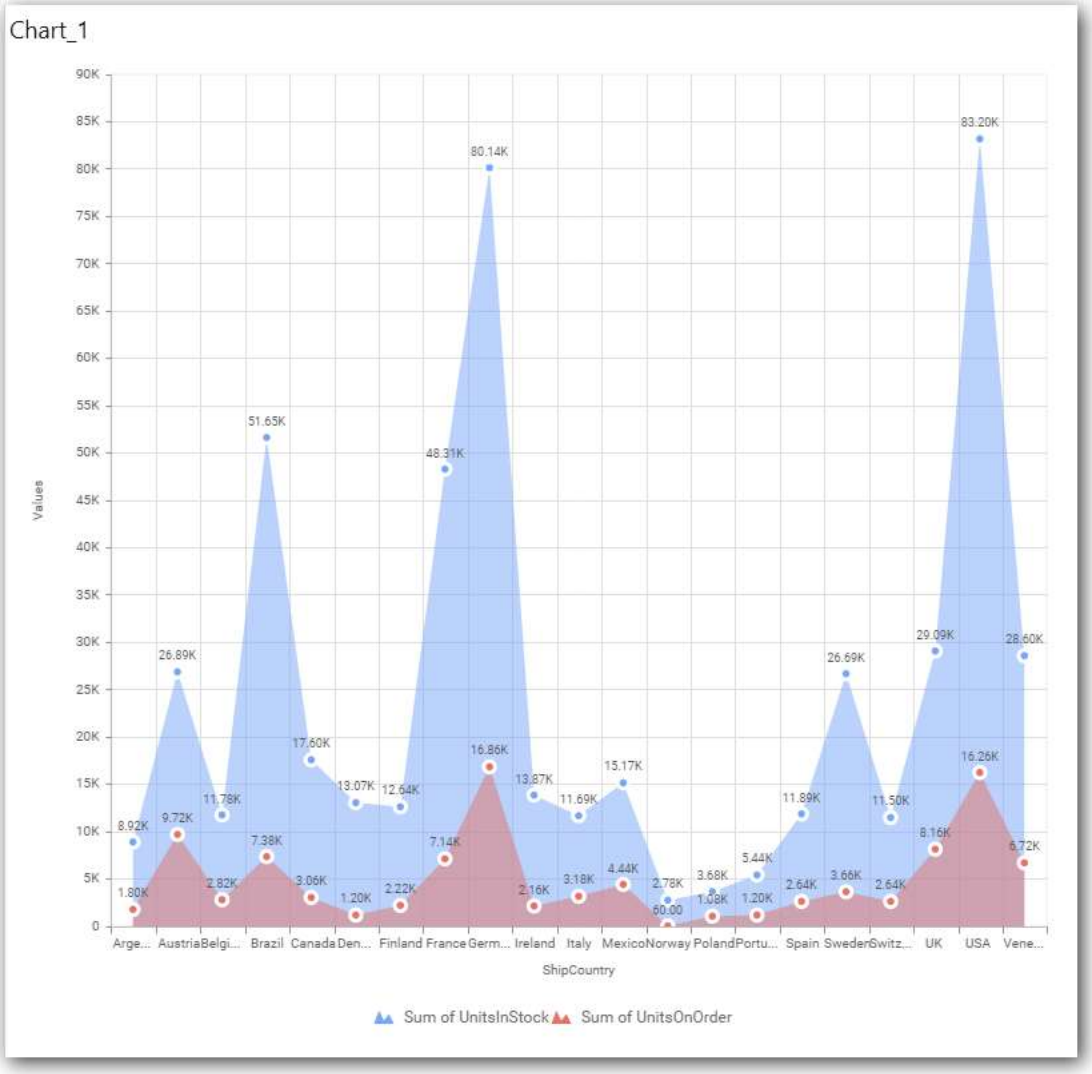


You can also add Dimensions and Columns to Value(s).

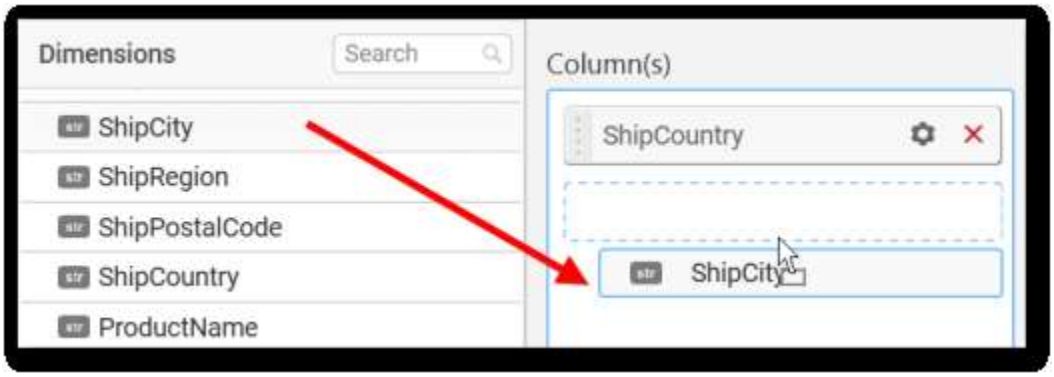
Assigning Column(s)

You can add the Dimension into Column field by drag and drop.

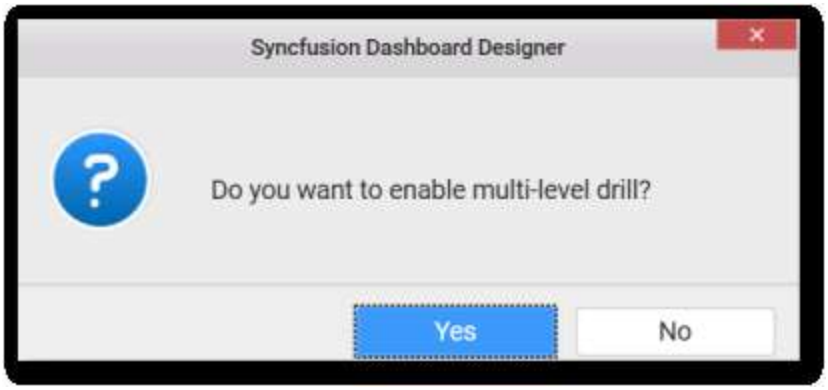




You have option to add more than one Column Value.

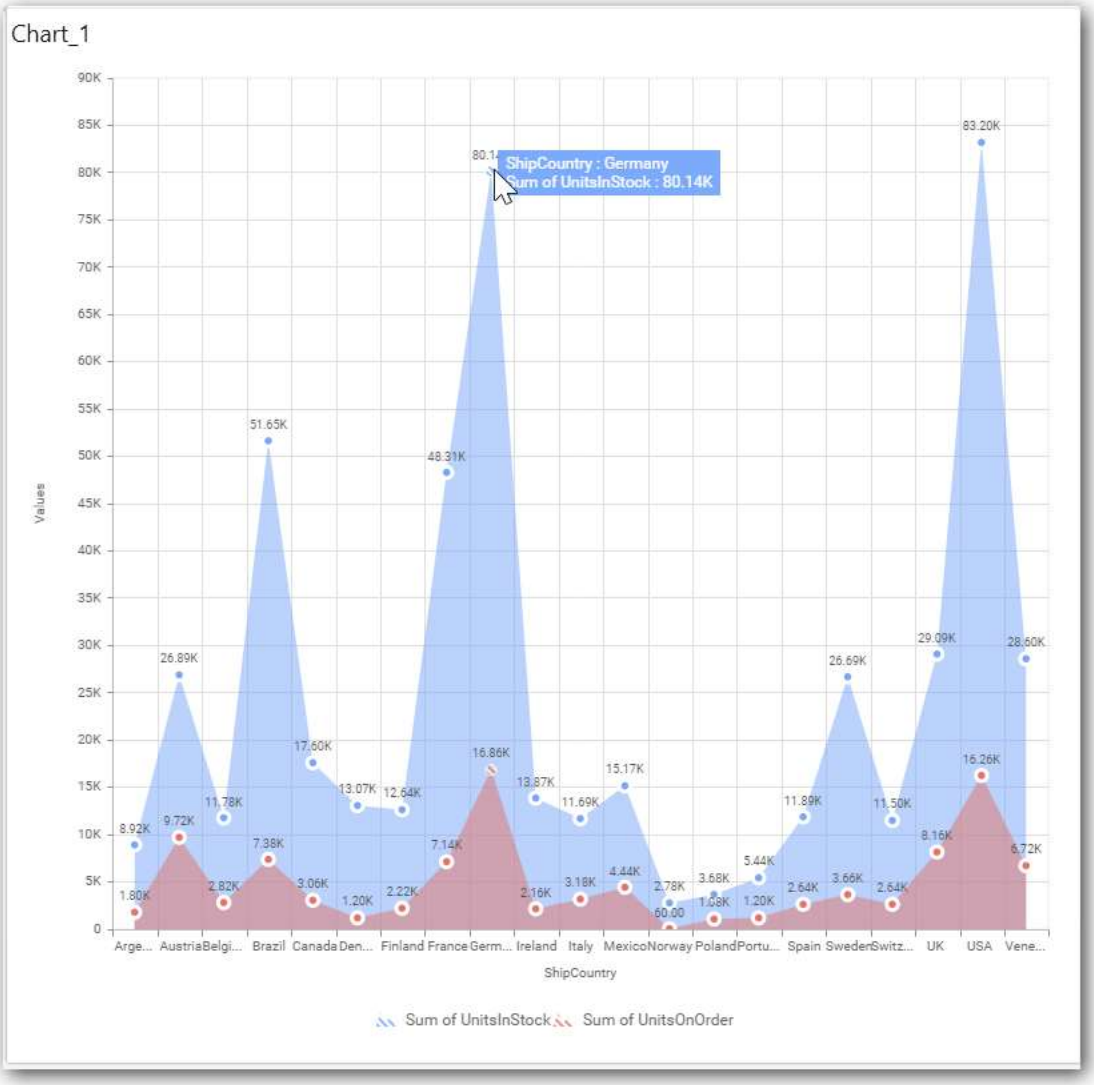


The following alert message will be shown.

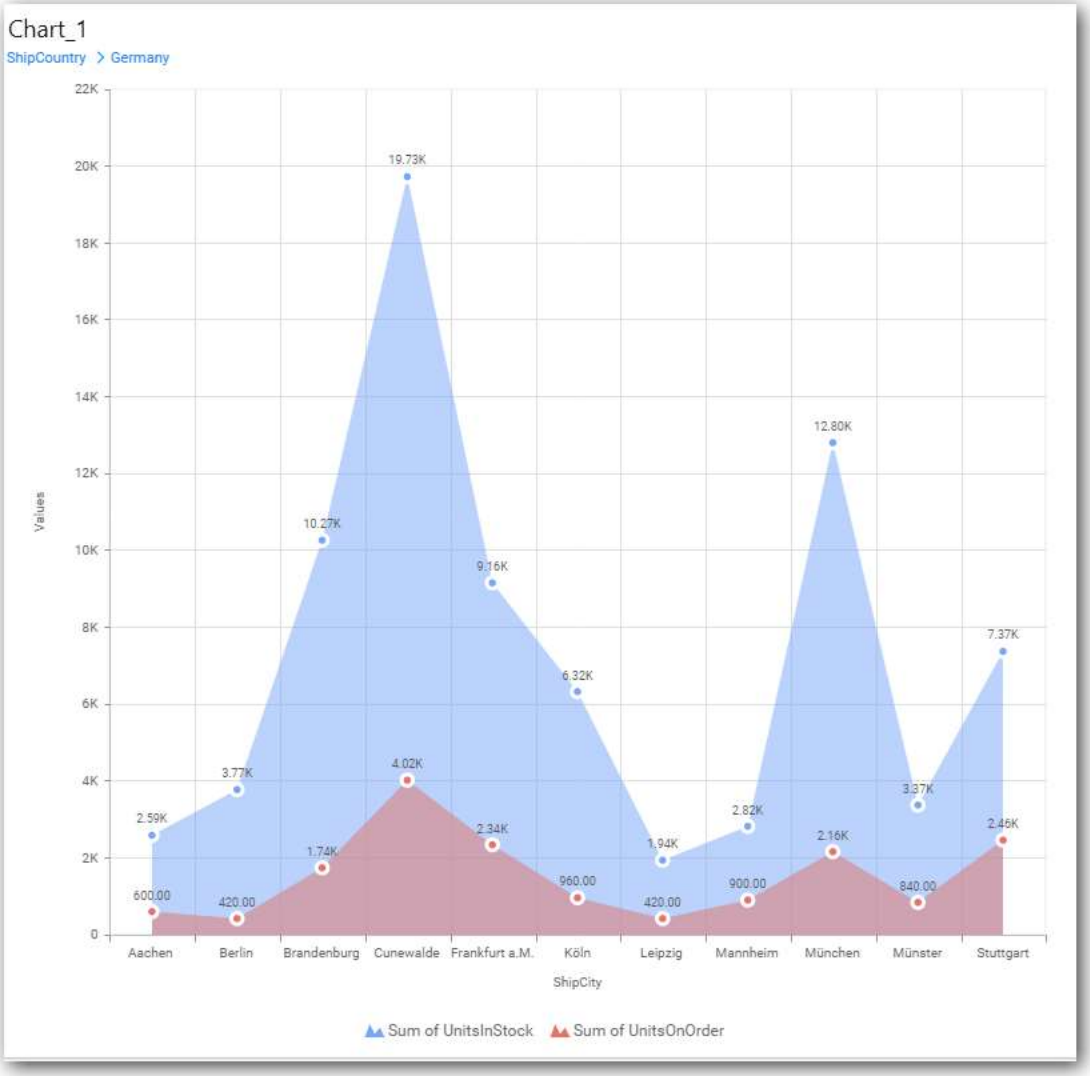


- If you choose Yes Drill down option will be enabled.

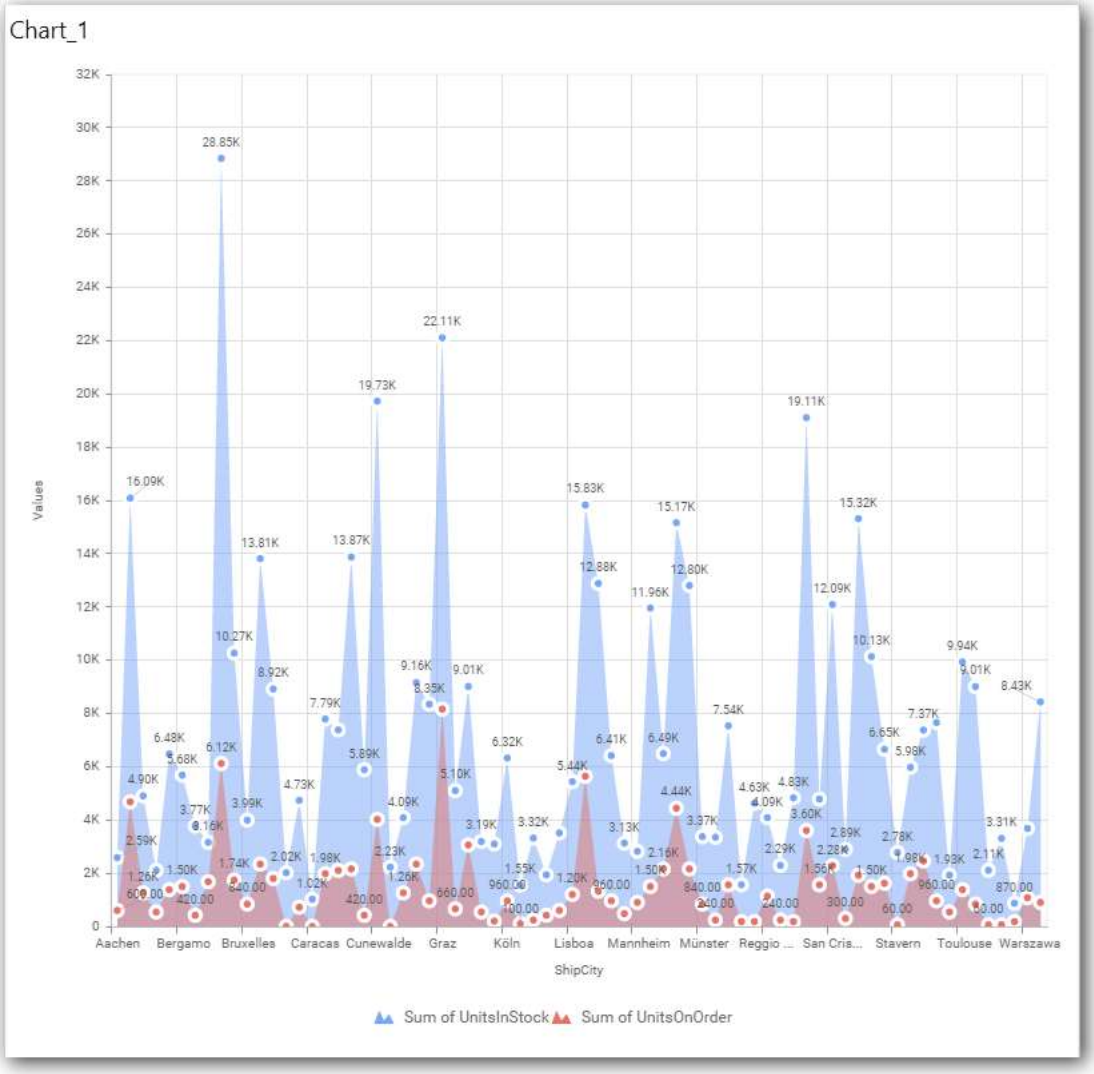
You can drill down the chart by clicking on the chart.



The drilled view of the chart is follows.

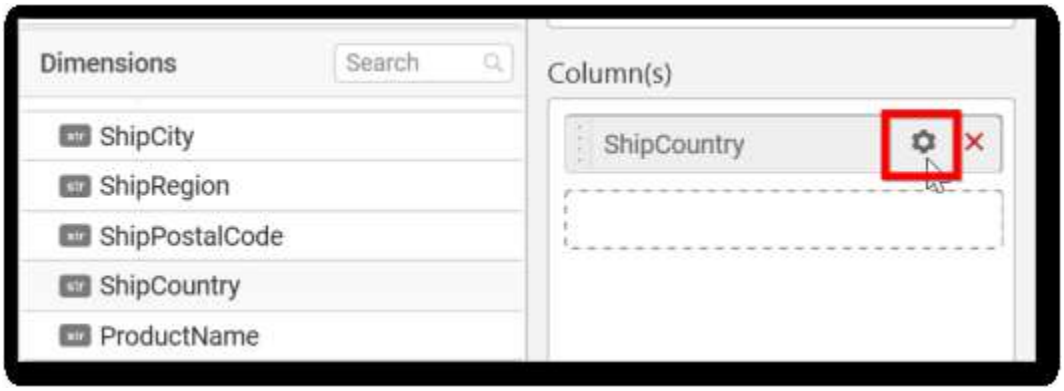


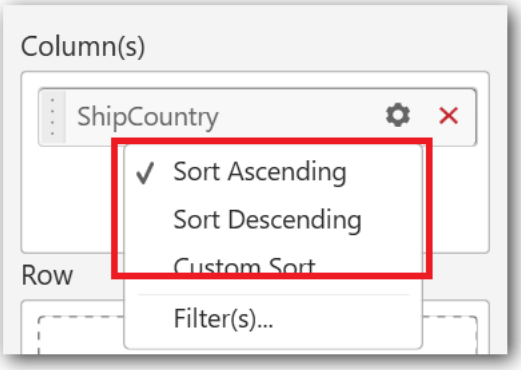
- If you click No the new Dimension value will replace old value.



You can also add Measure and Expression columns into Column(s) field.

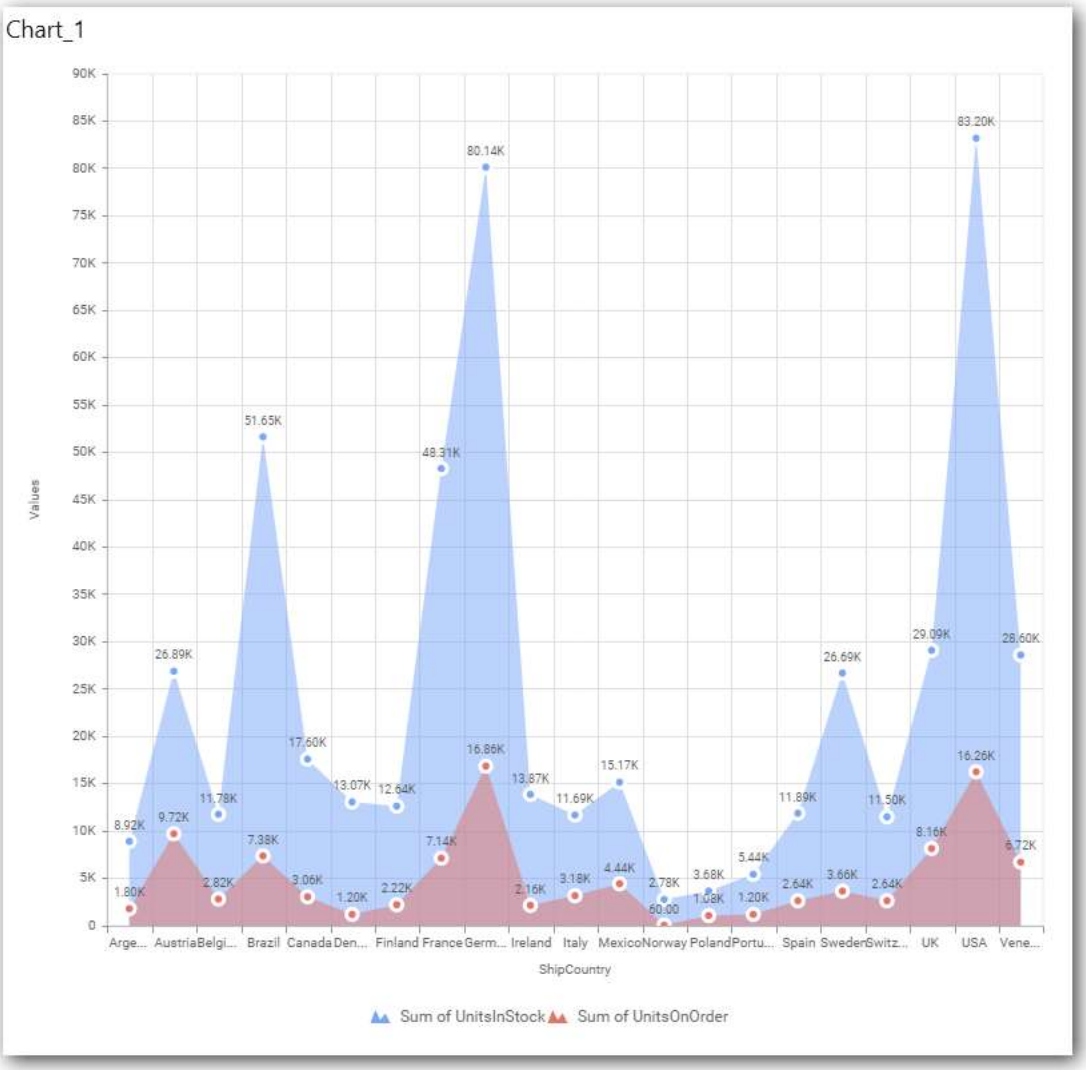
You have options to change the settings.



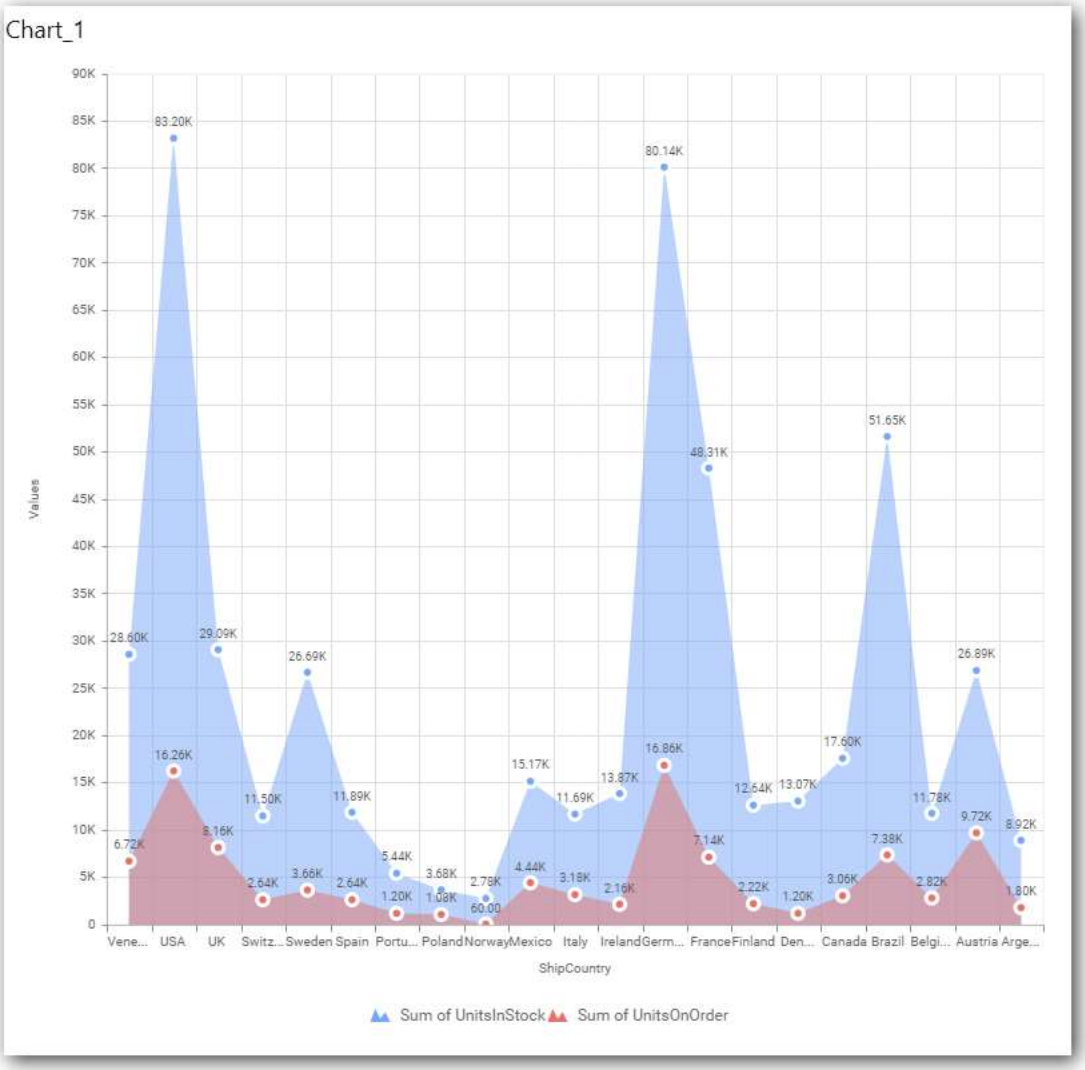


You can sort the chart either in Ascending or Descending series.

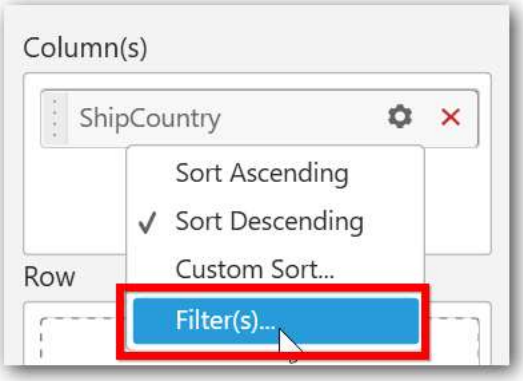
Ascending Order:

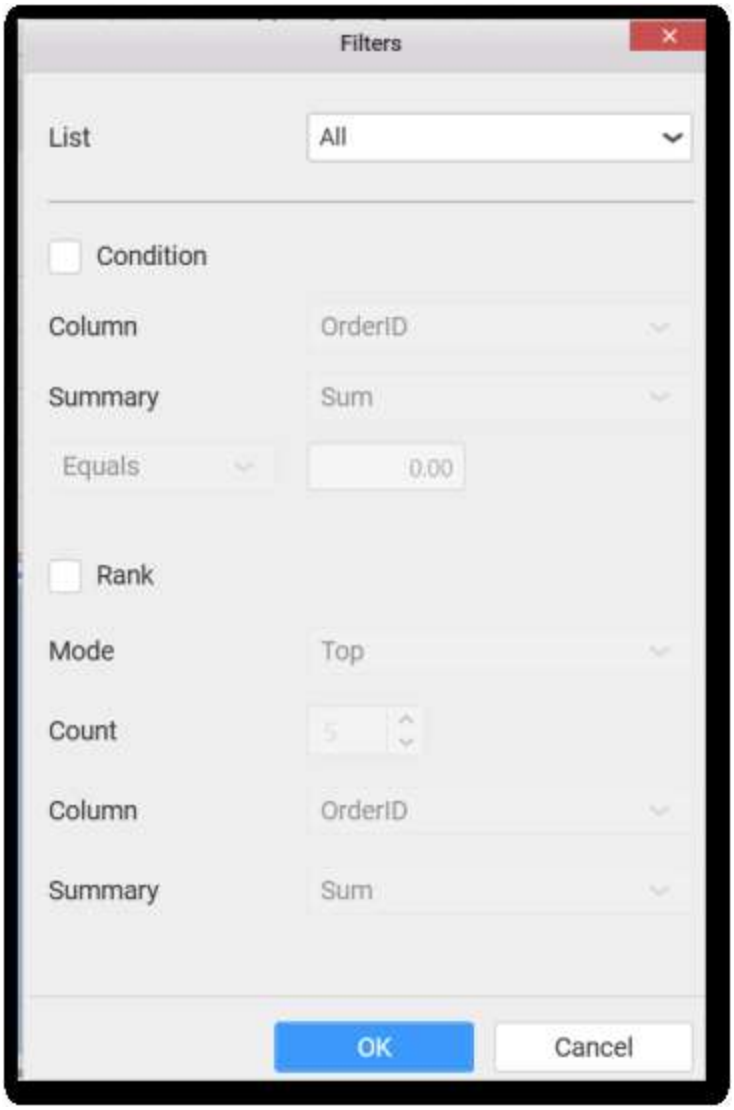


Descending order:

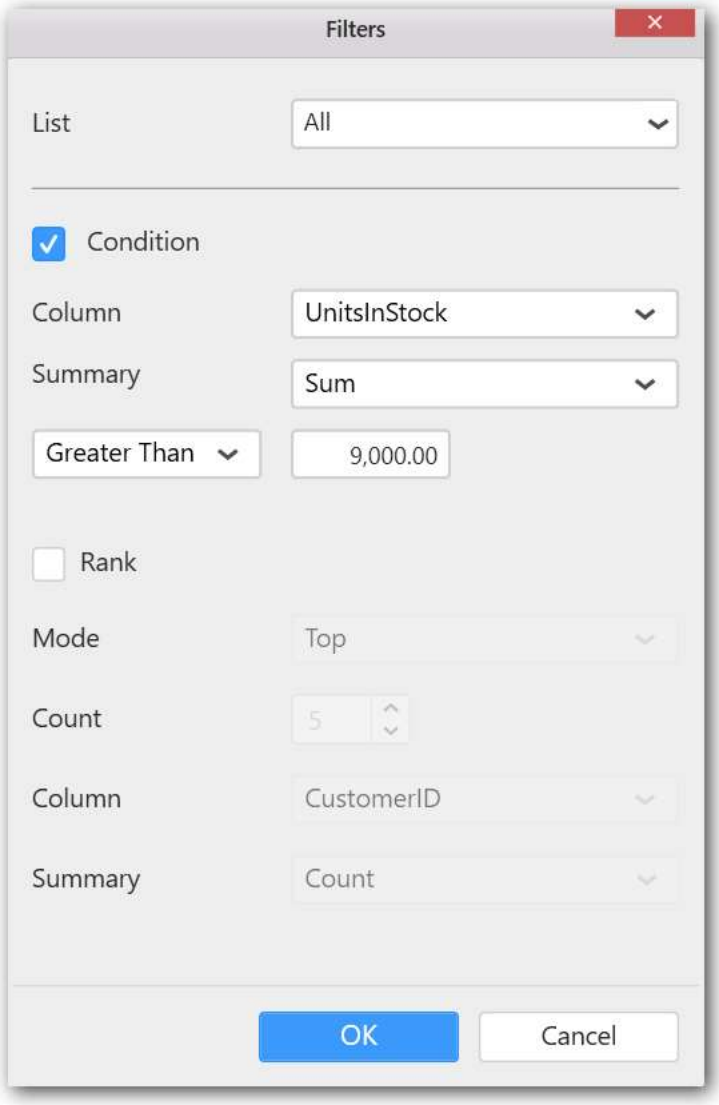


You can apply a filter.

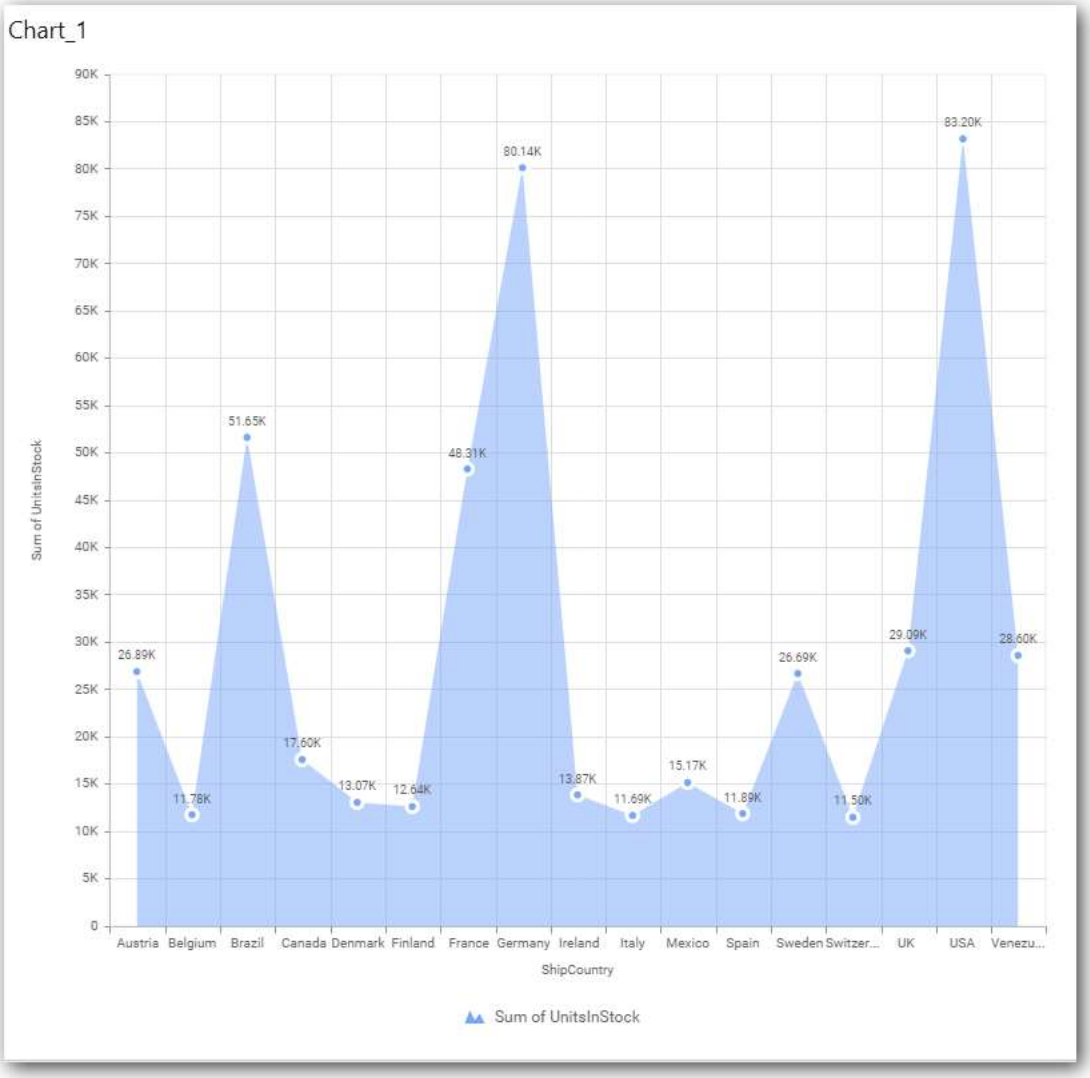




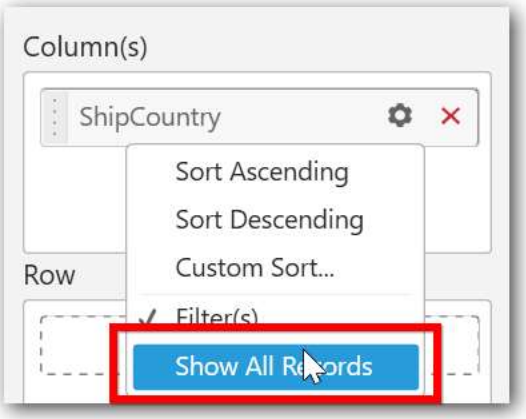
Select the Conditions and Rank you need.



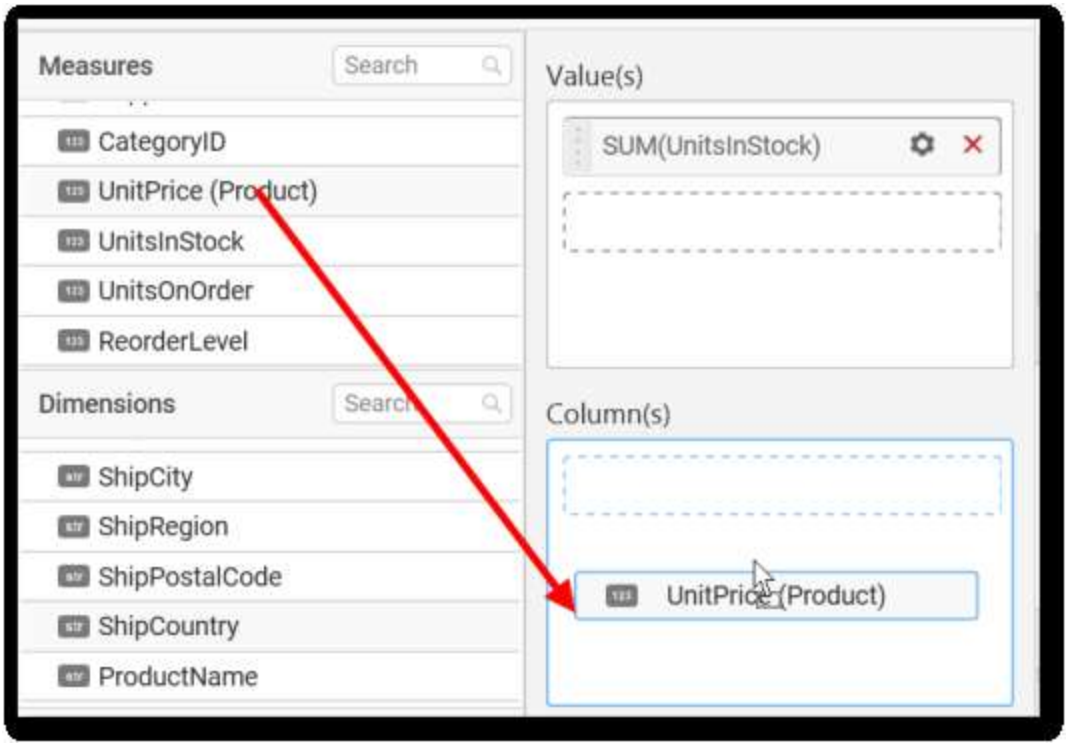
Now the chart will be rendered like this.

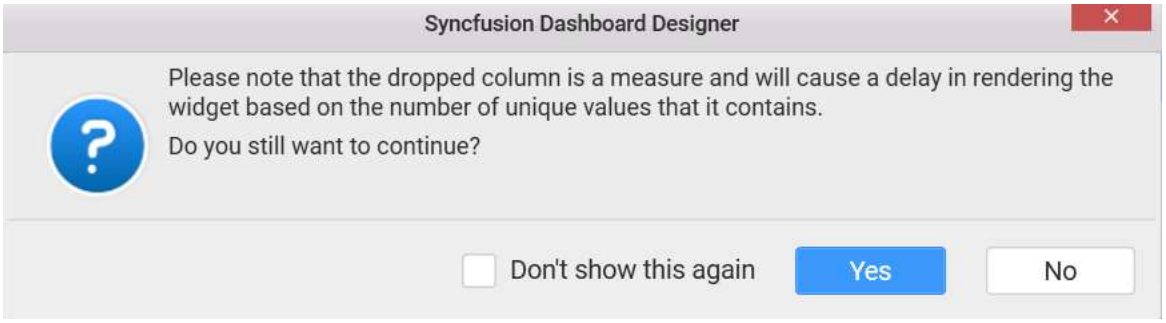


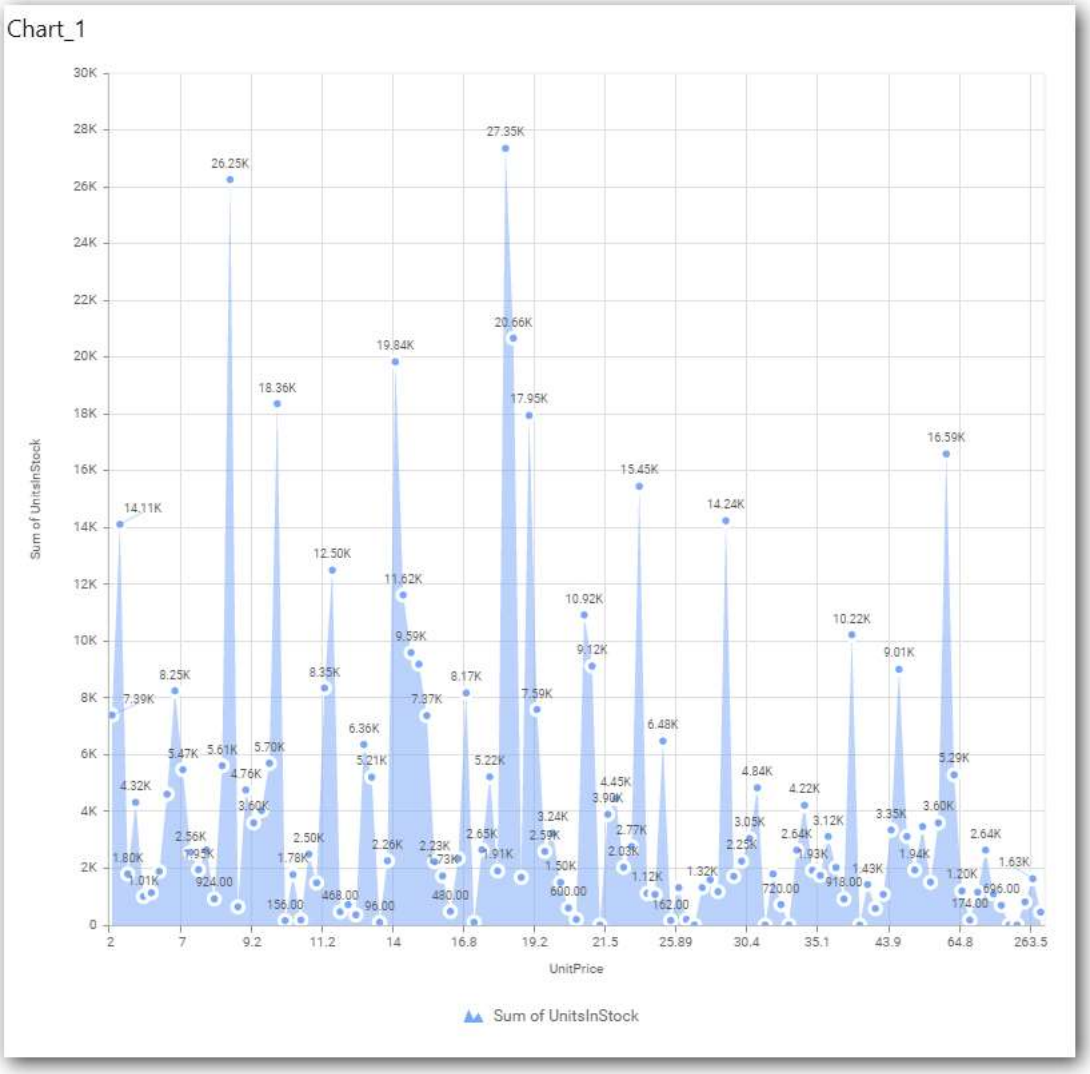
To show all records again click on Show All Records.



You can add Measures into Column(s).

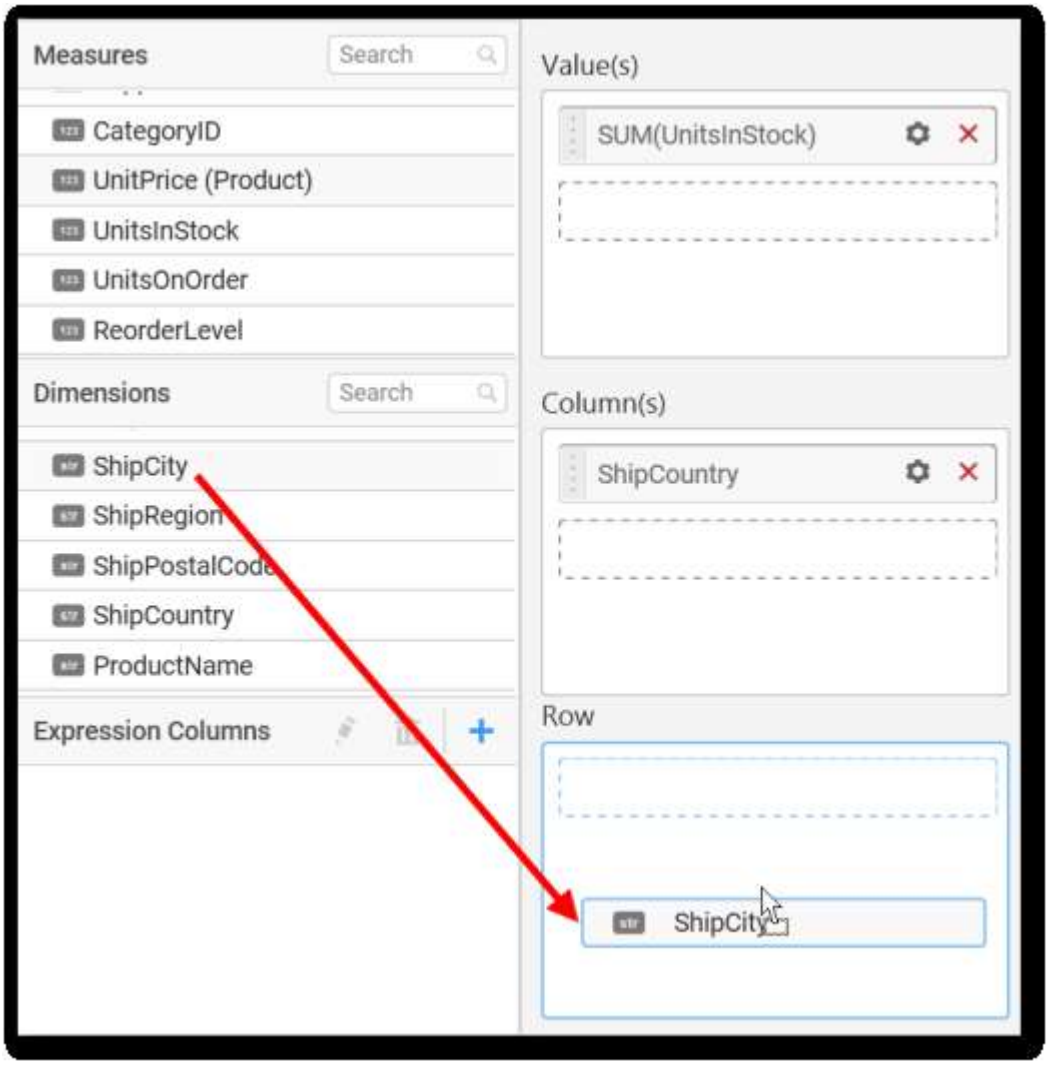




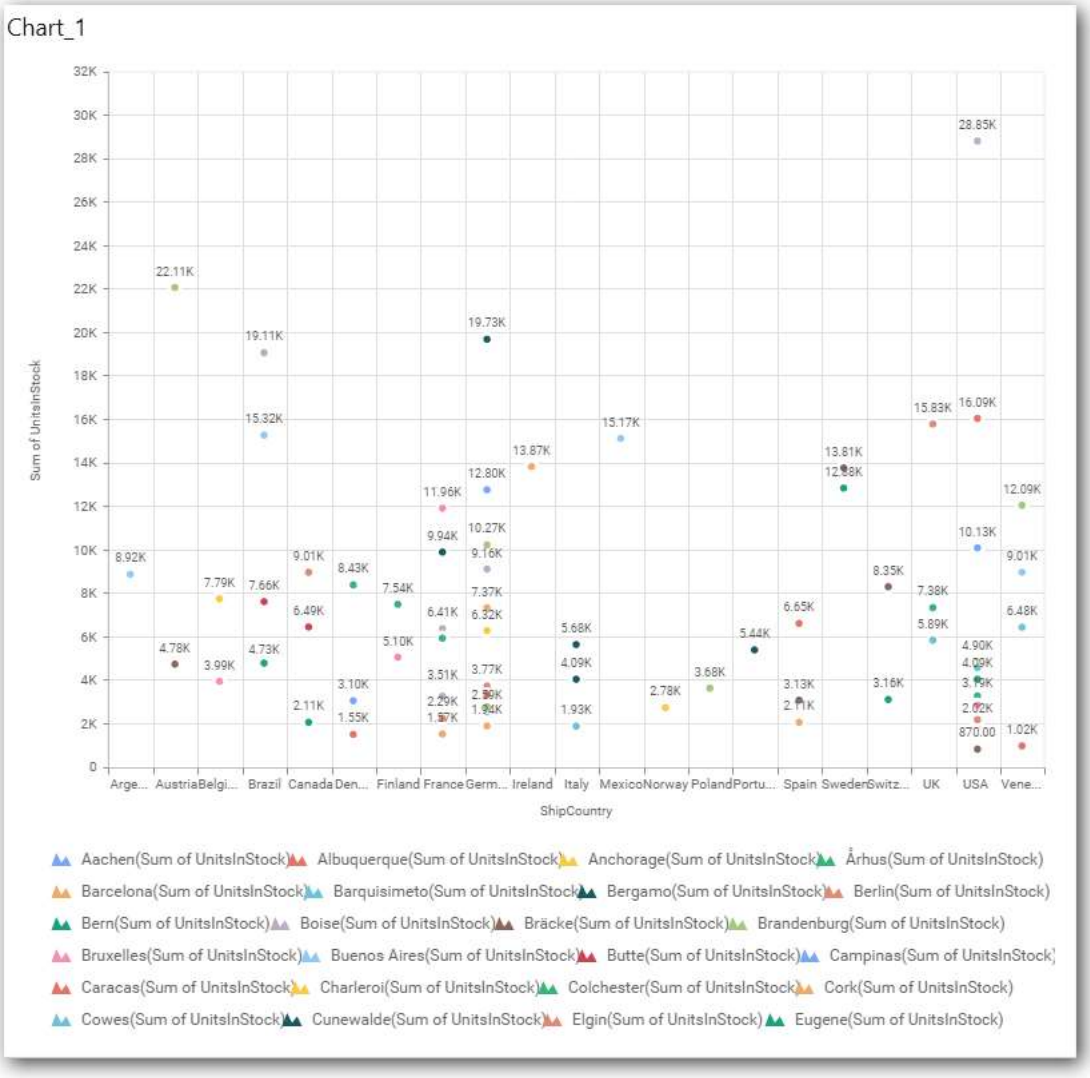


Assigning Row

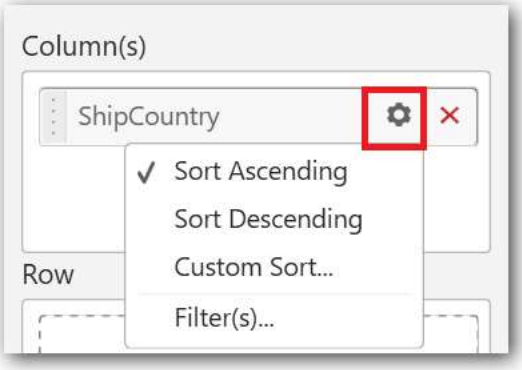
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to column(s).



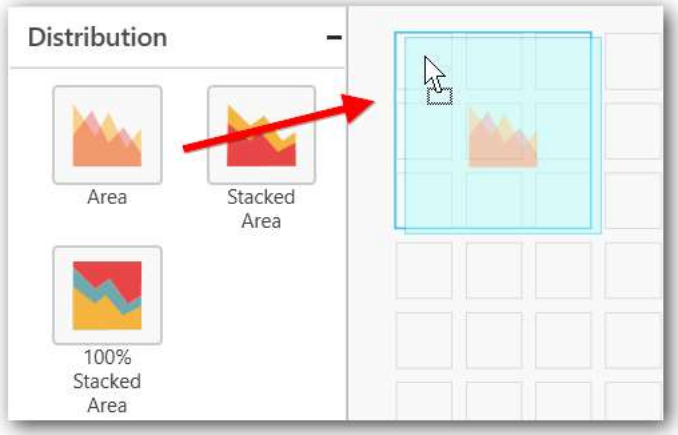
How to configure the SSAS data to Area Chart?

Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that

you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuring SSAS data to Area chart

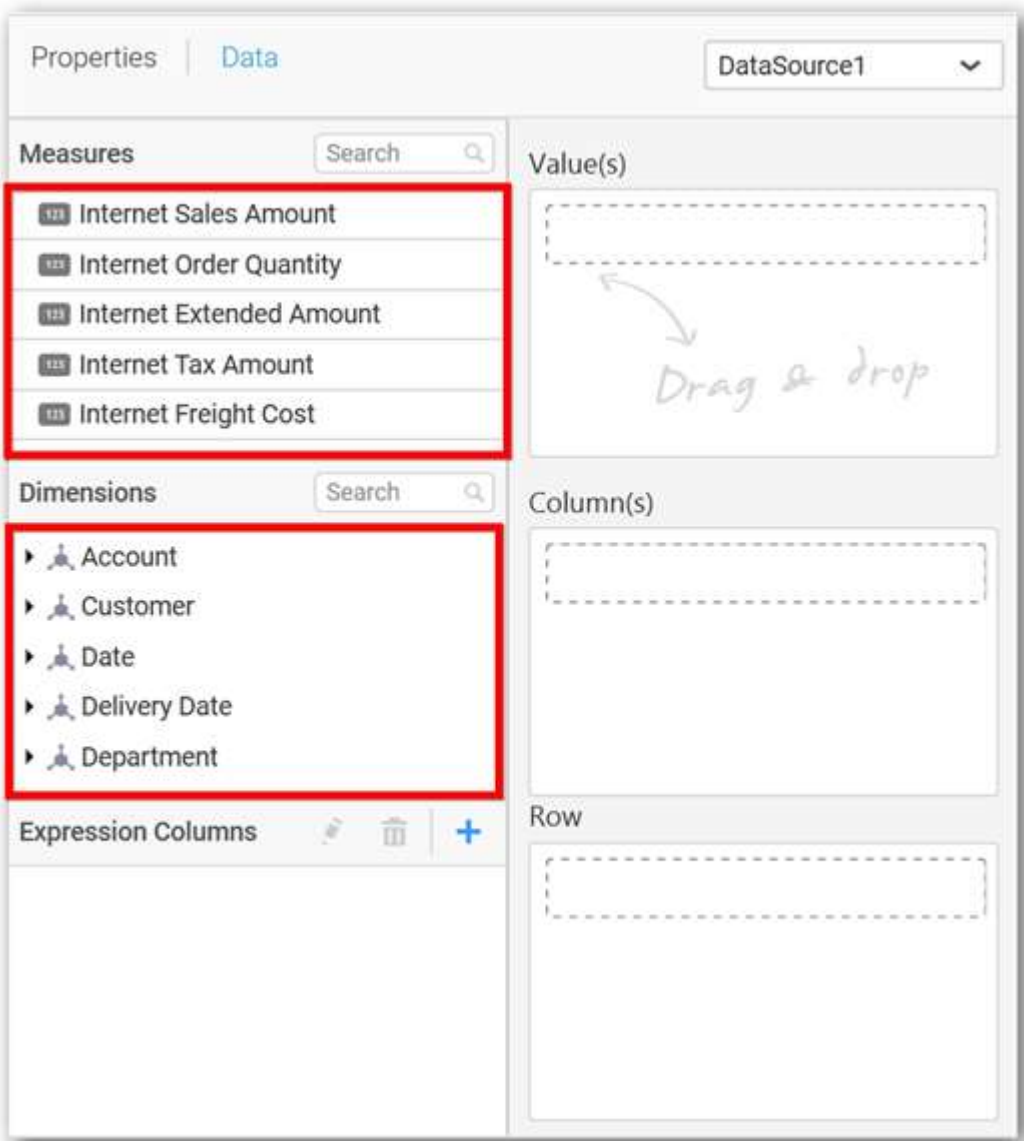
Drag and drop the Area chart widget into canvas and resize into your required size.



Select the dropped widget using mouse.

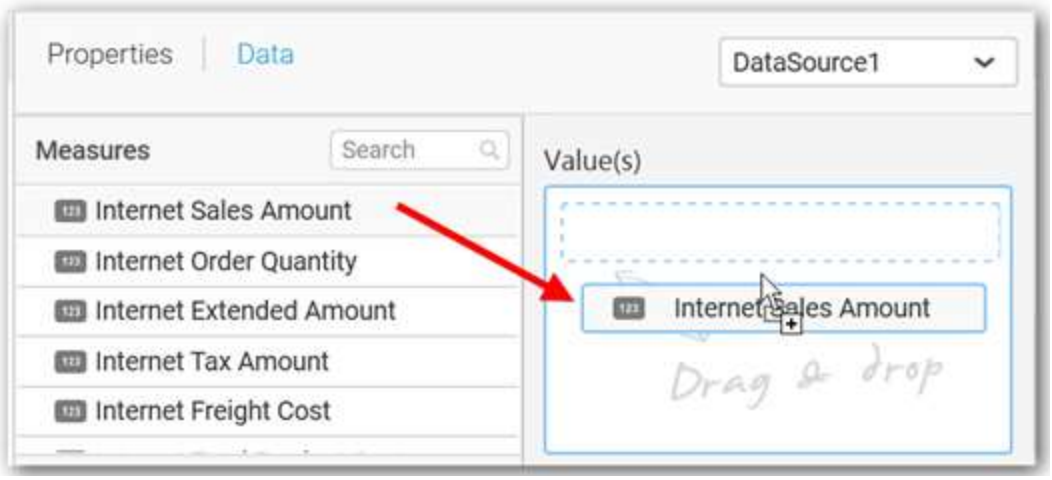
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions.

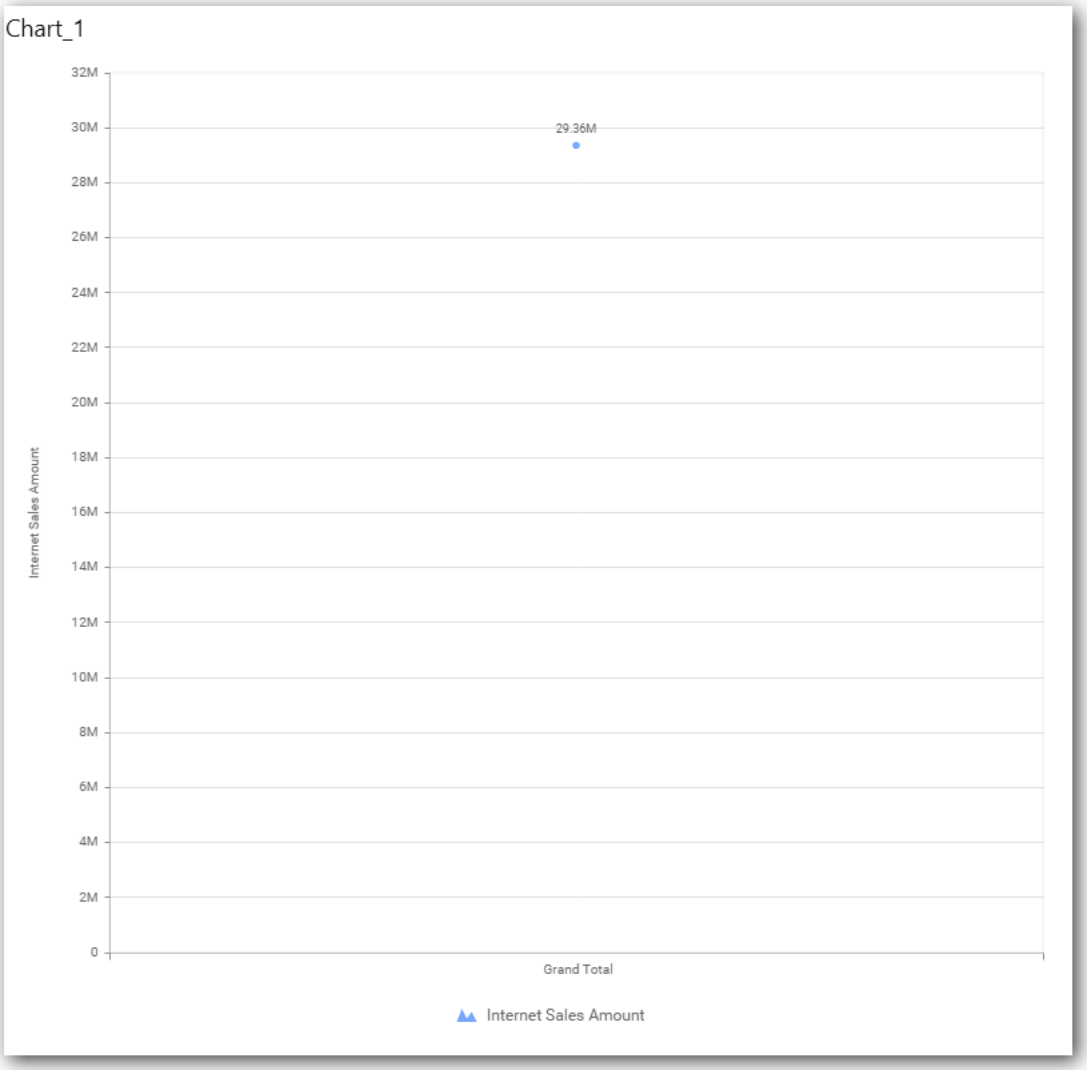


Assigning Value(s)

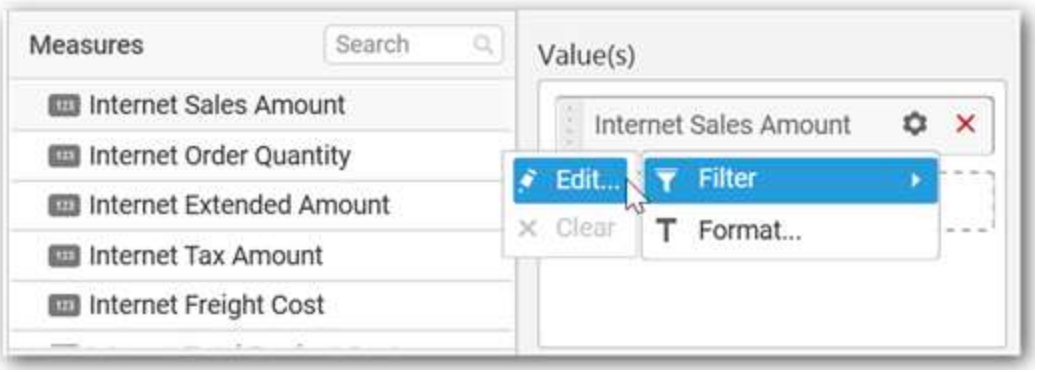
Drag and drop a column under Measures category into Value(s) section.



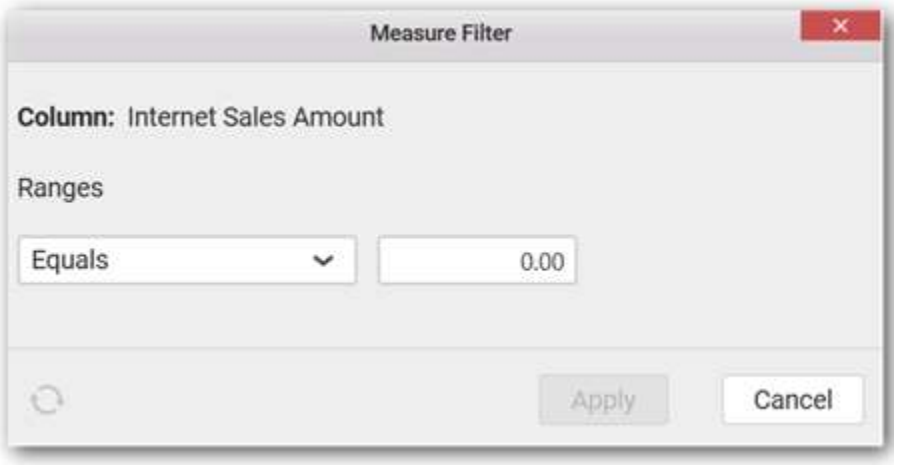
Now the chart will be rendered like this.



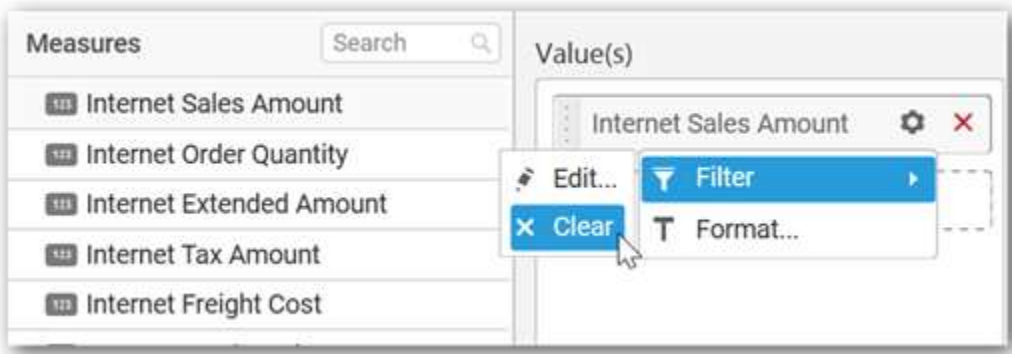
Define the filter criteria to match through choosing Edit option in Filter menu item.



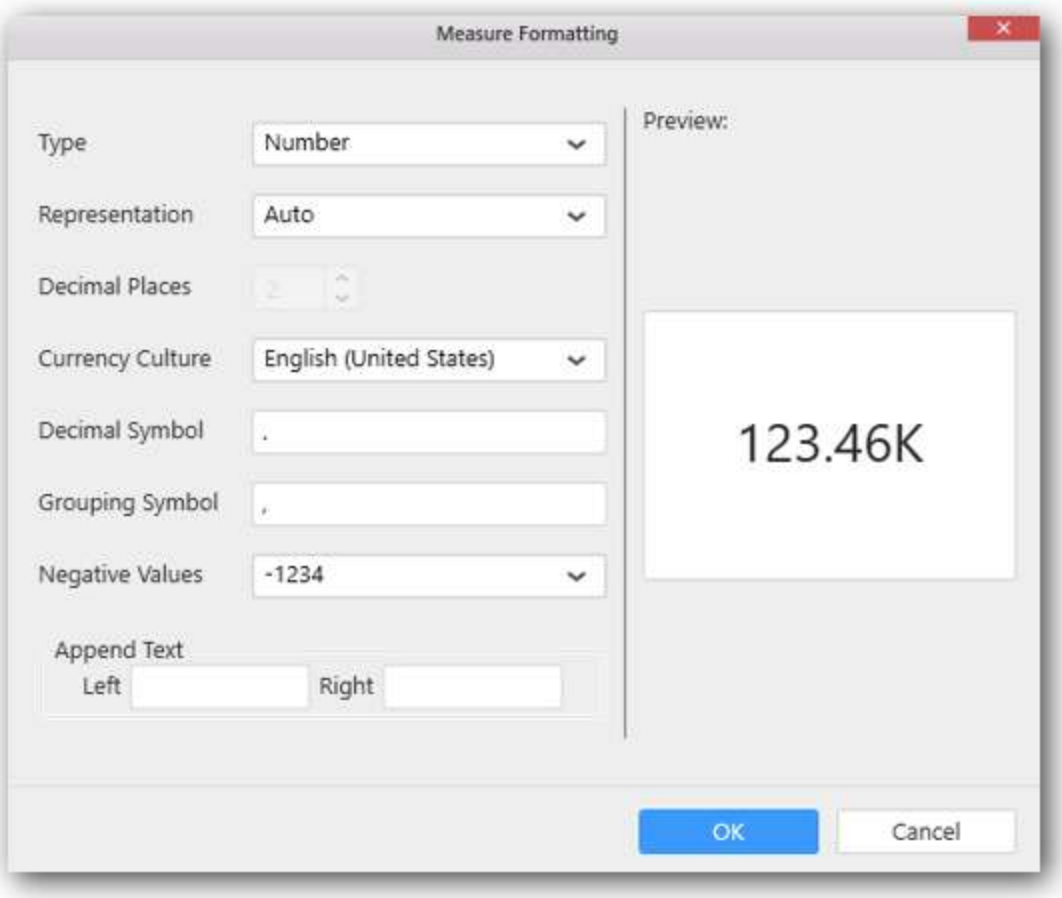
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



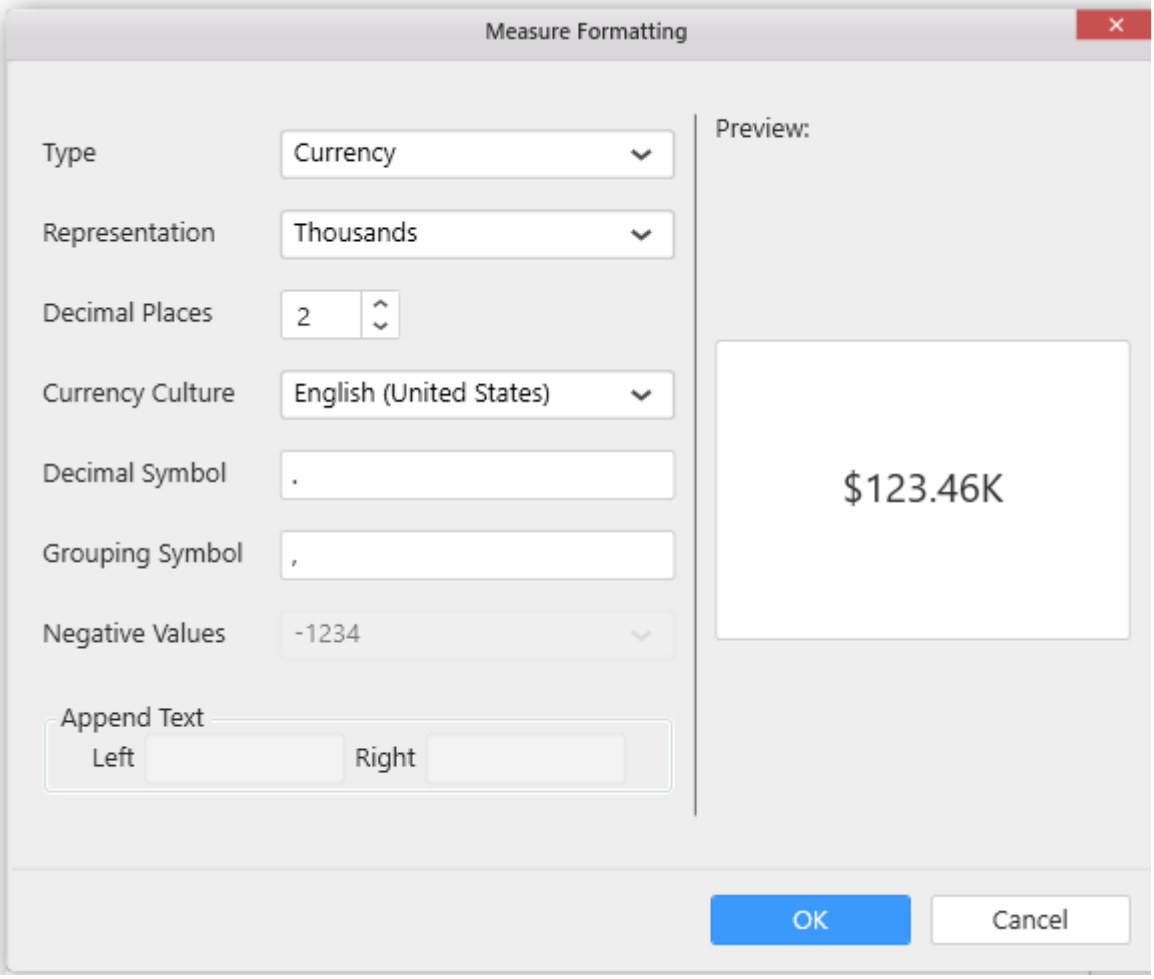
Select Clear option to clear the defined filter.



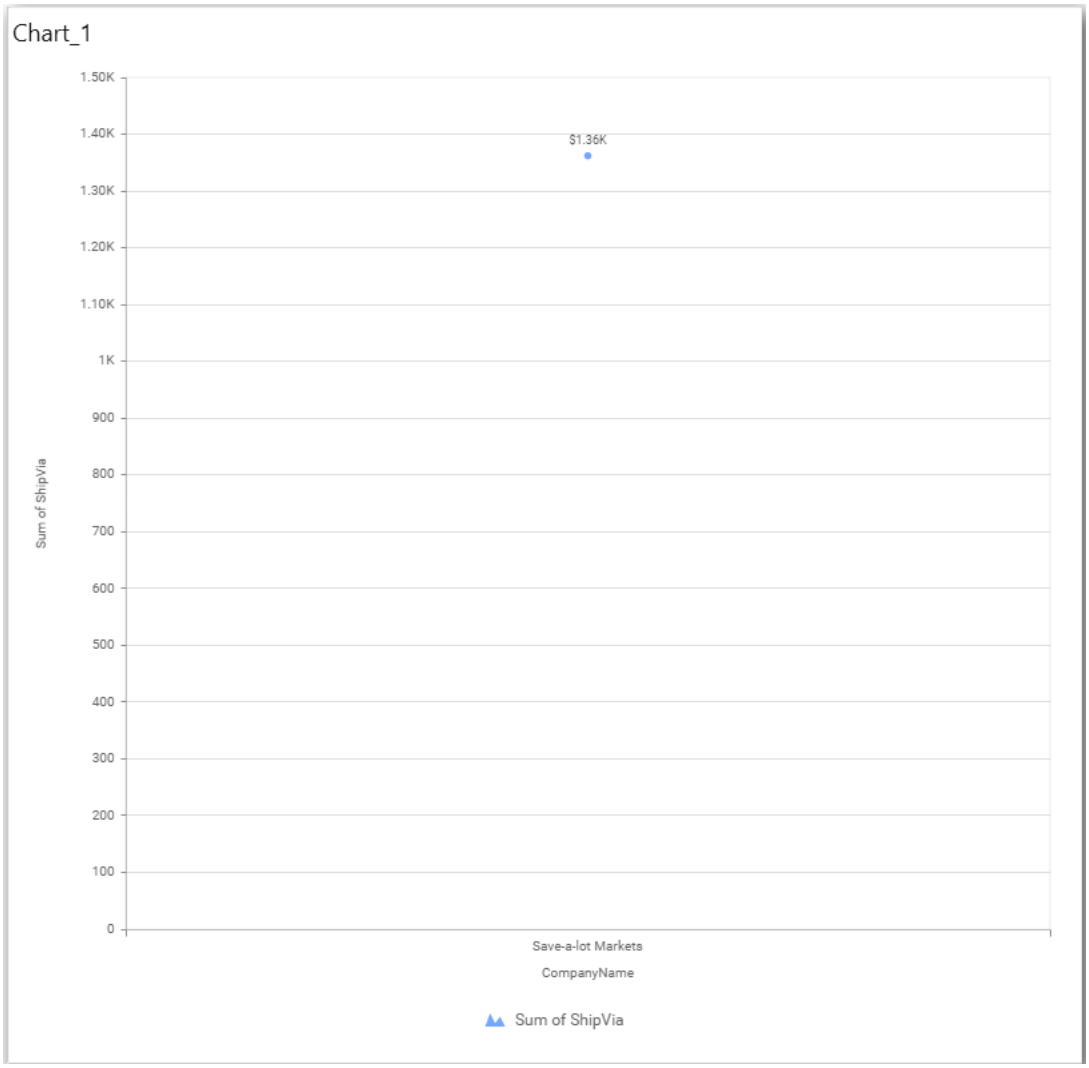
Select Format option to define the display format to the values in the column through Measure Formatting window.



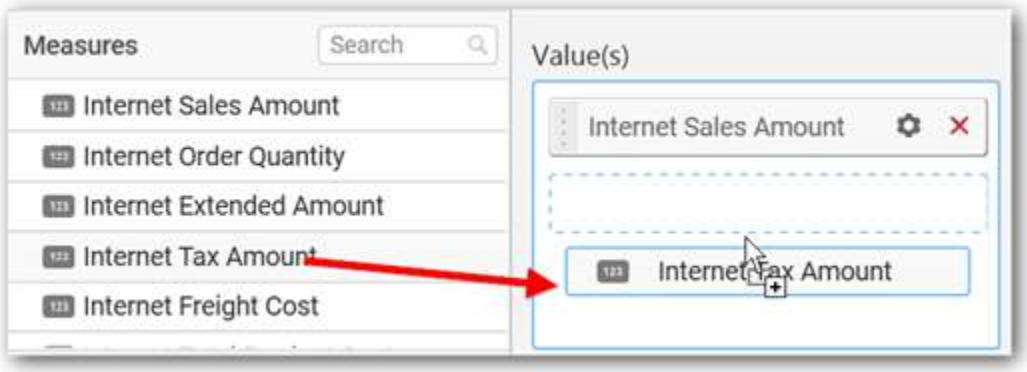
Choose the options you need and click OK.

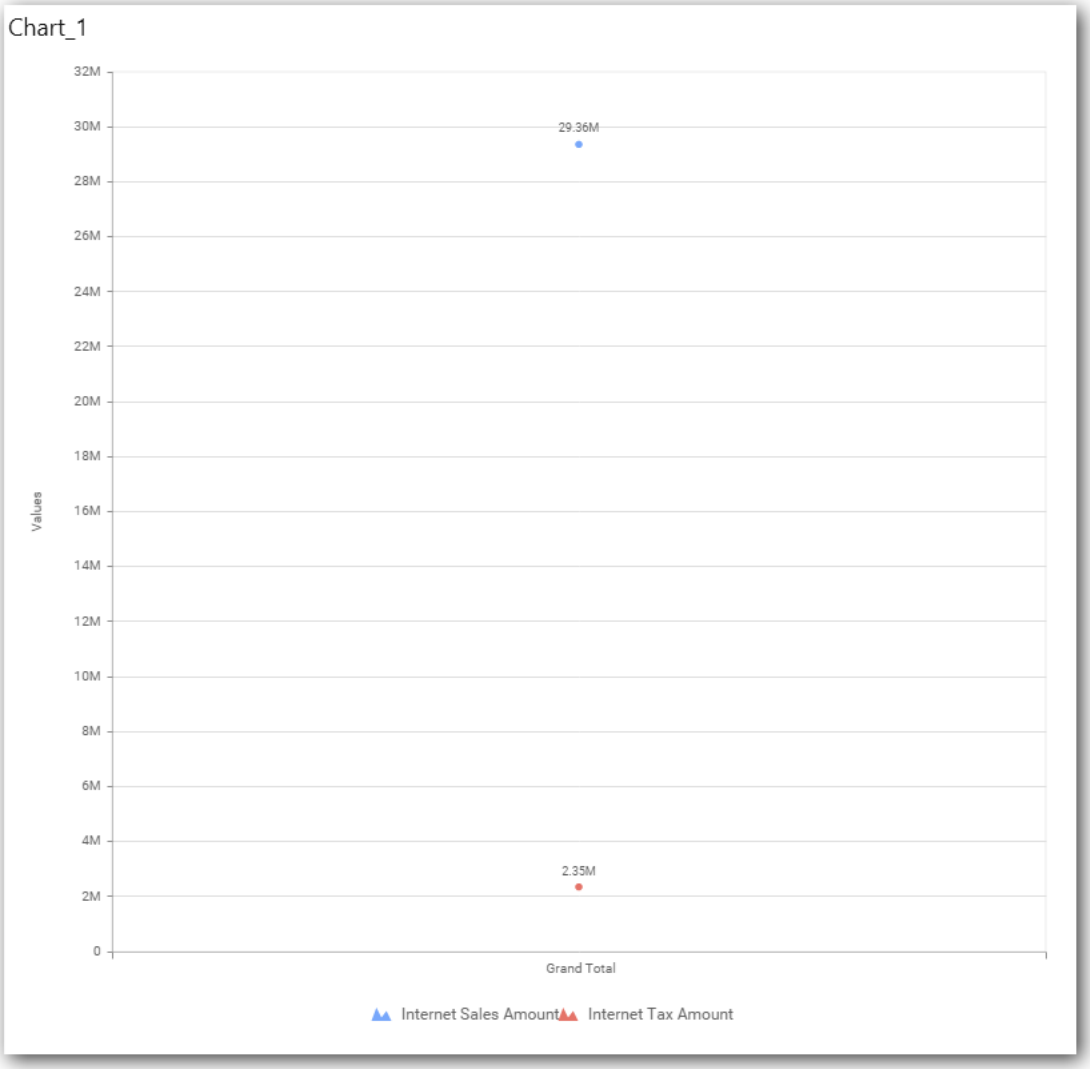


Now the Chart will be rendered like this.



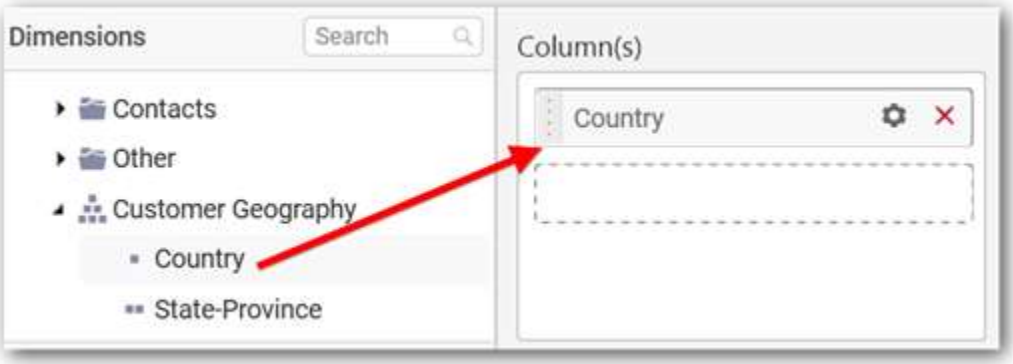
You can also add more than one column to the Value(s) section.

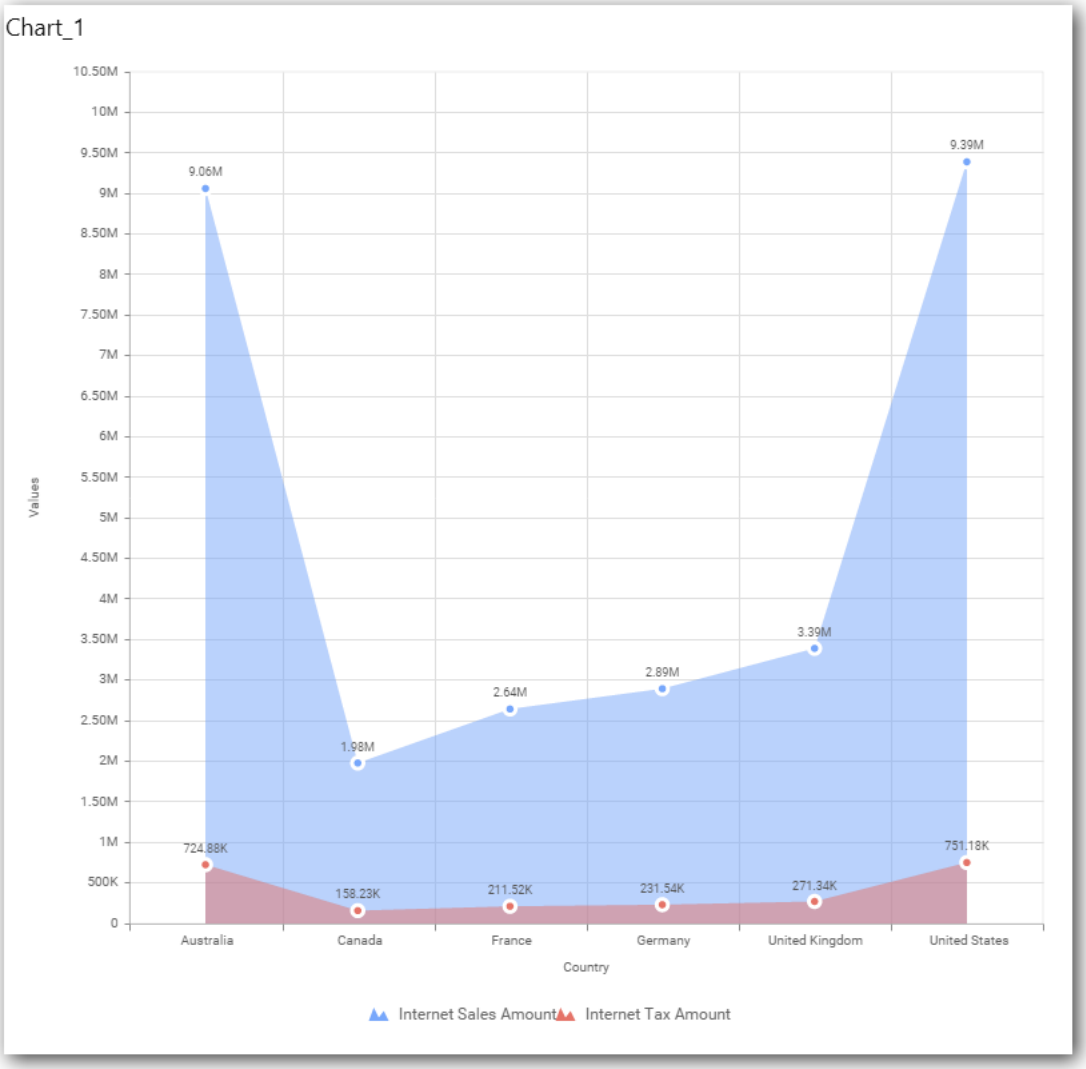




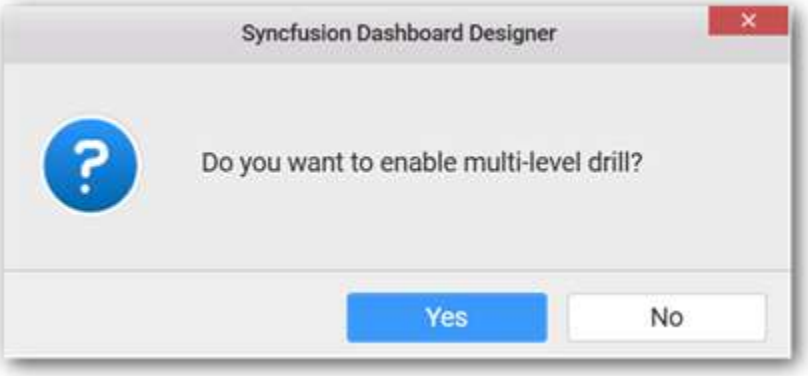
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.



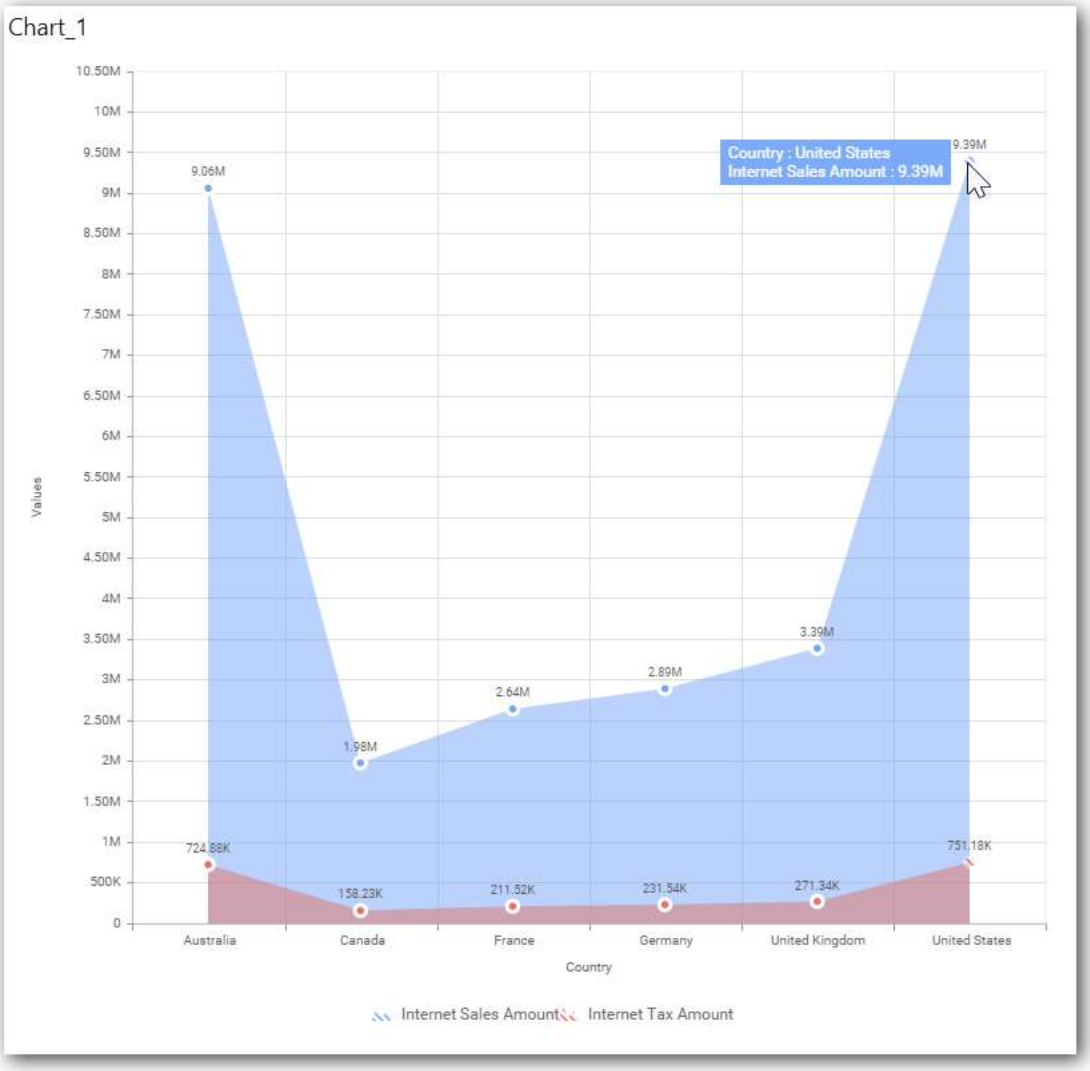


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

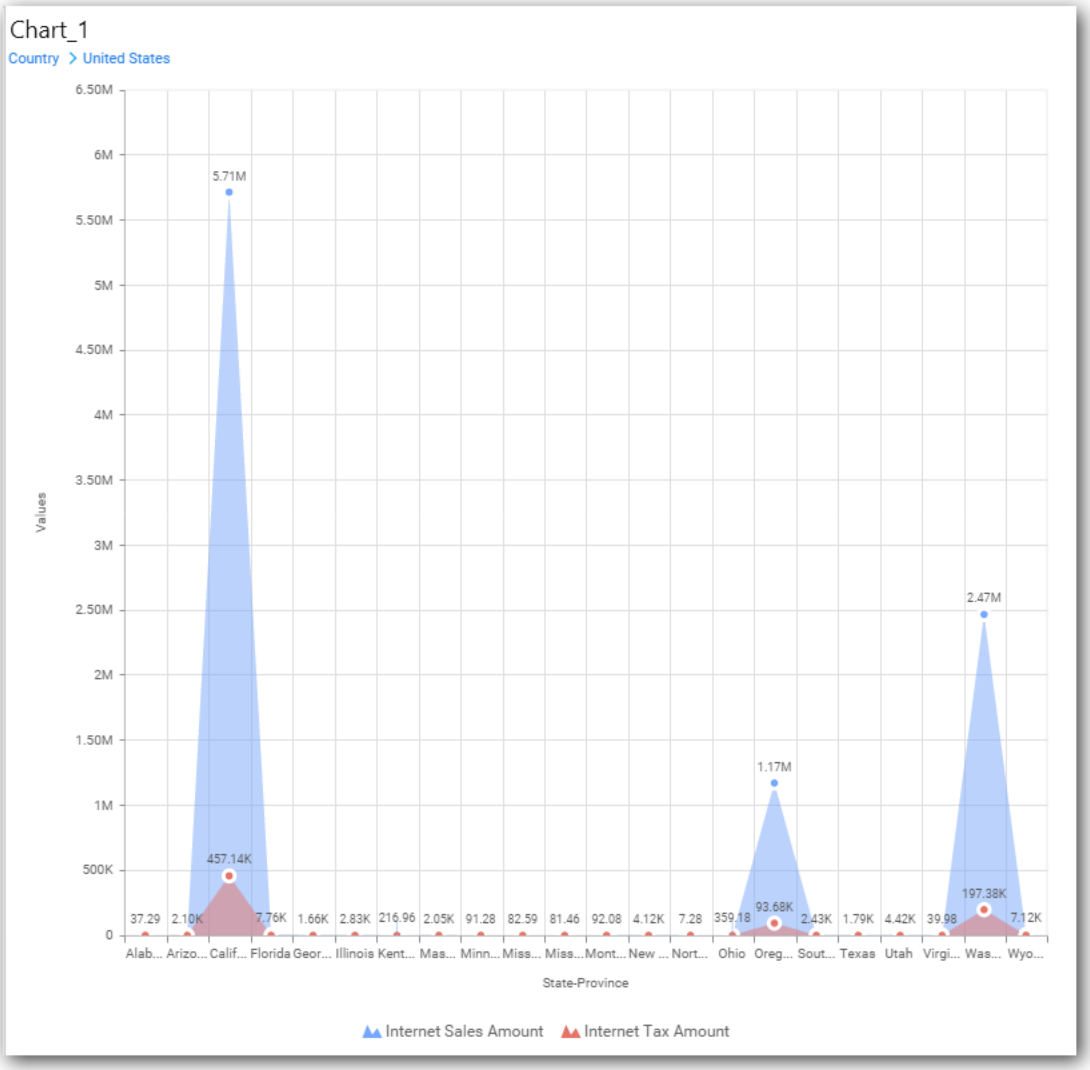


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

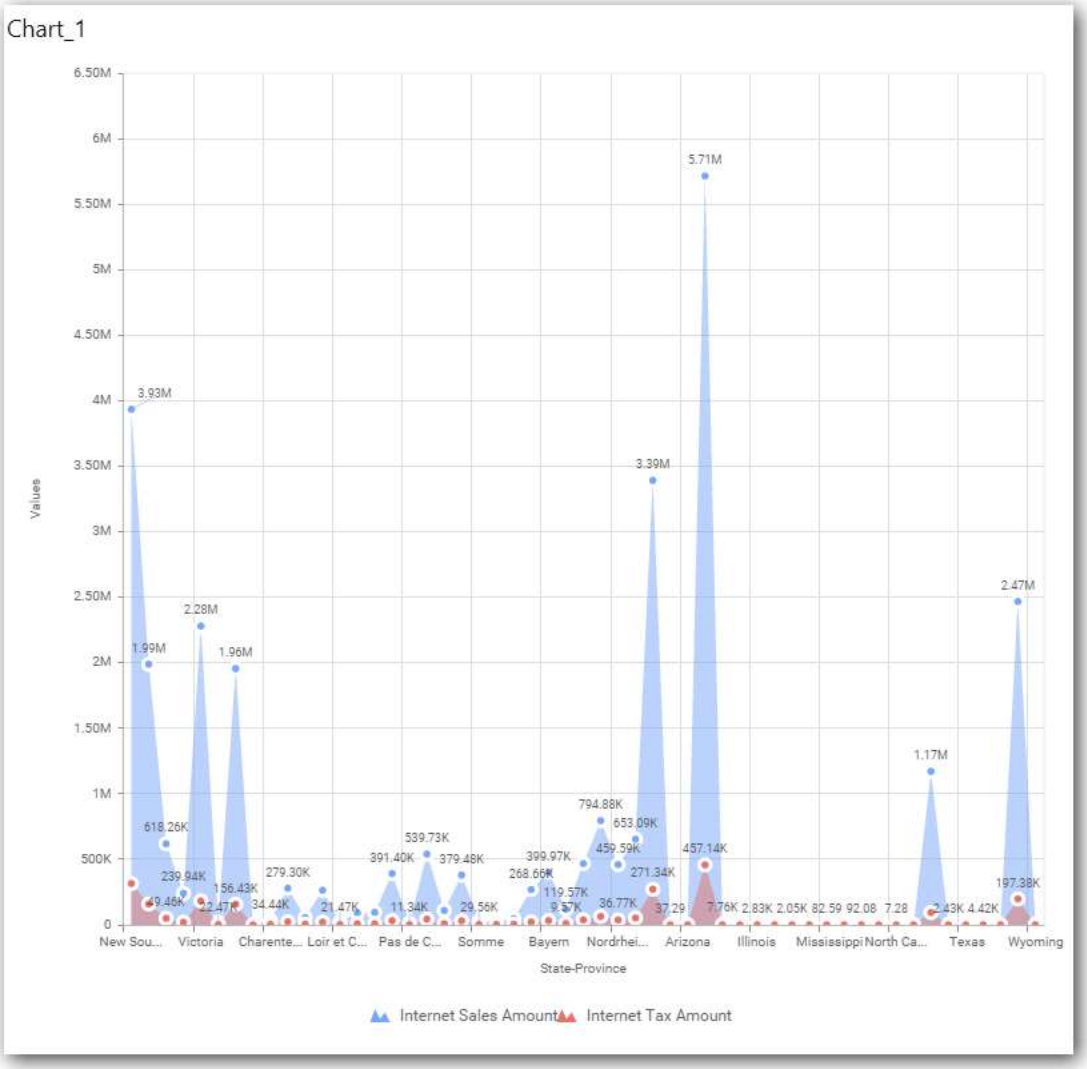
Click the respective data value marker in chart to drill into its inner level.



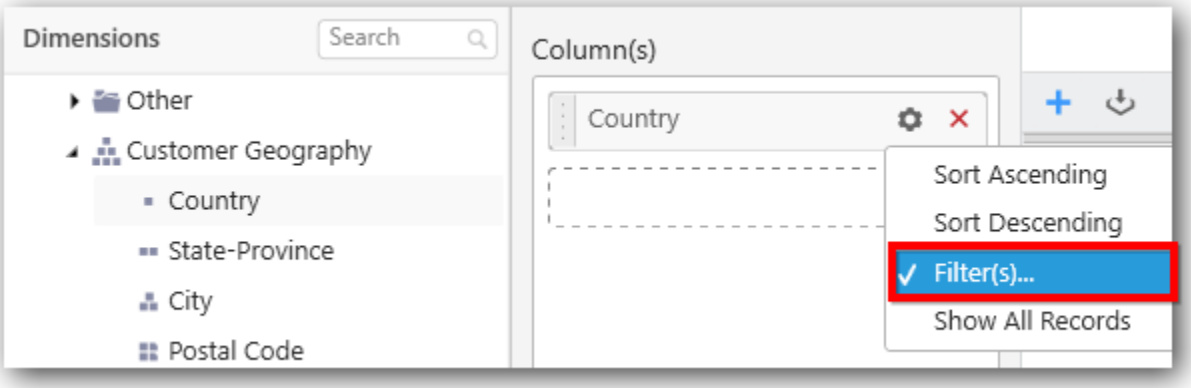
The drilled view of the chart is follows.



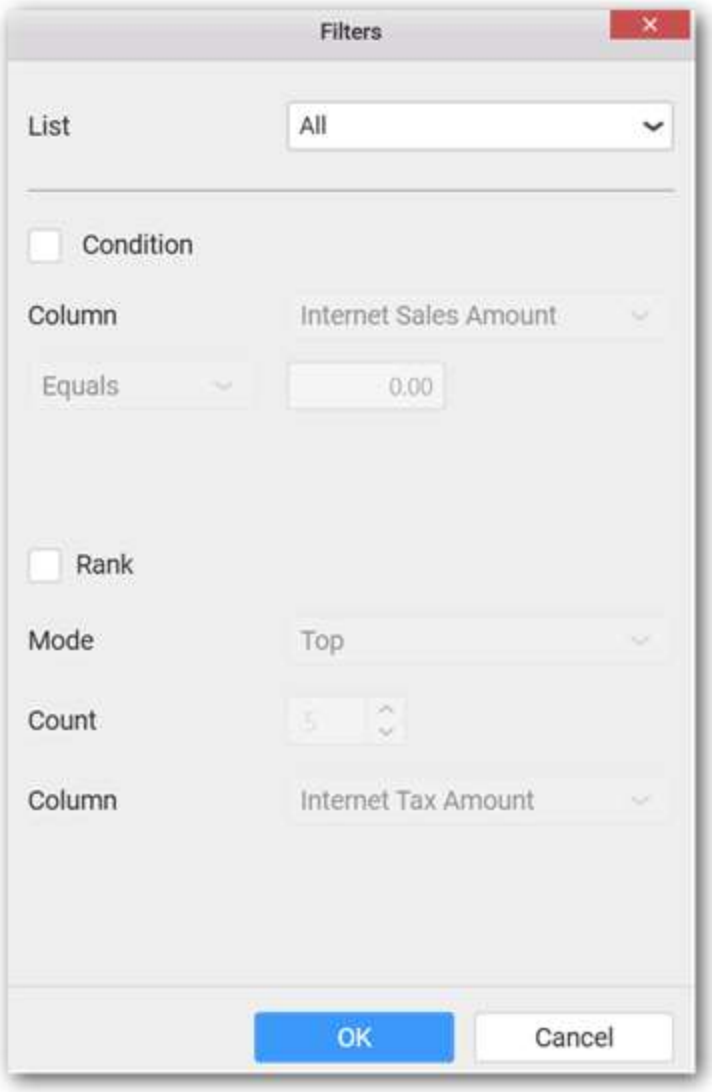
Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level.



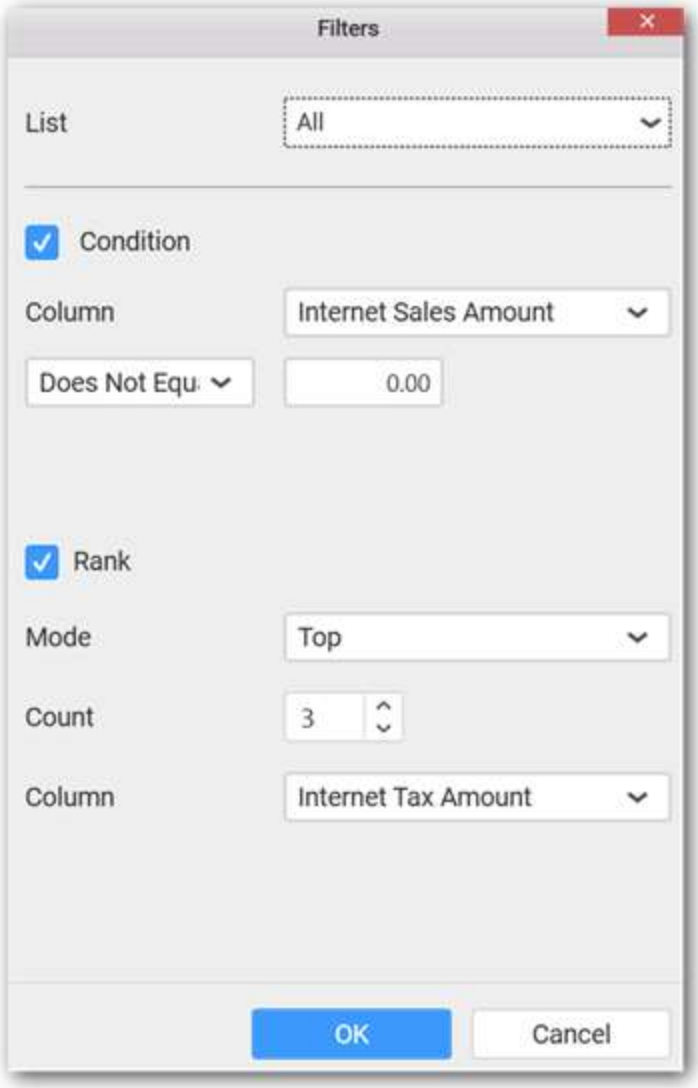
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



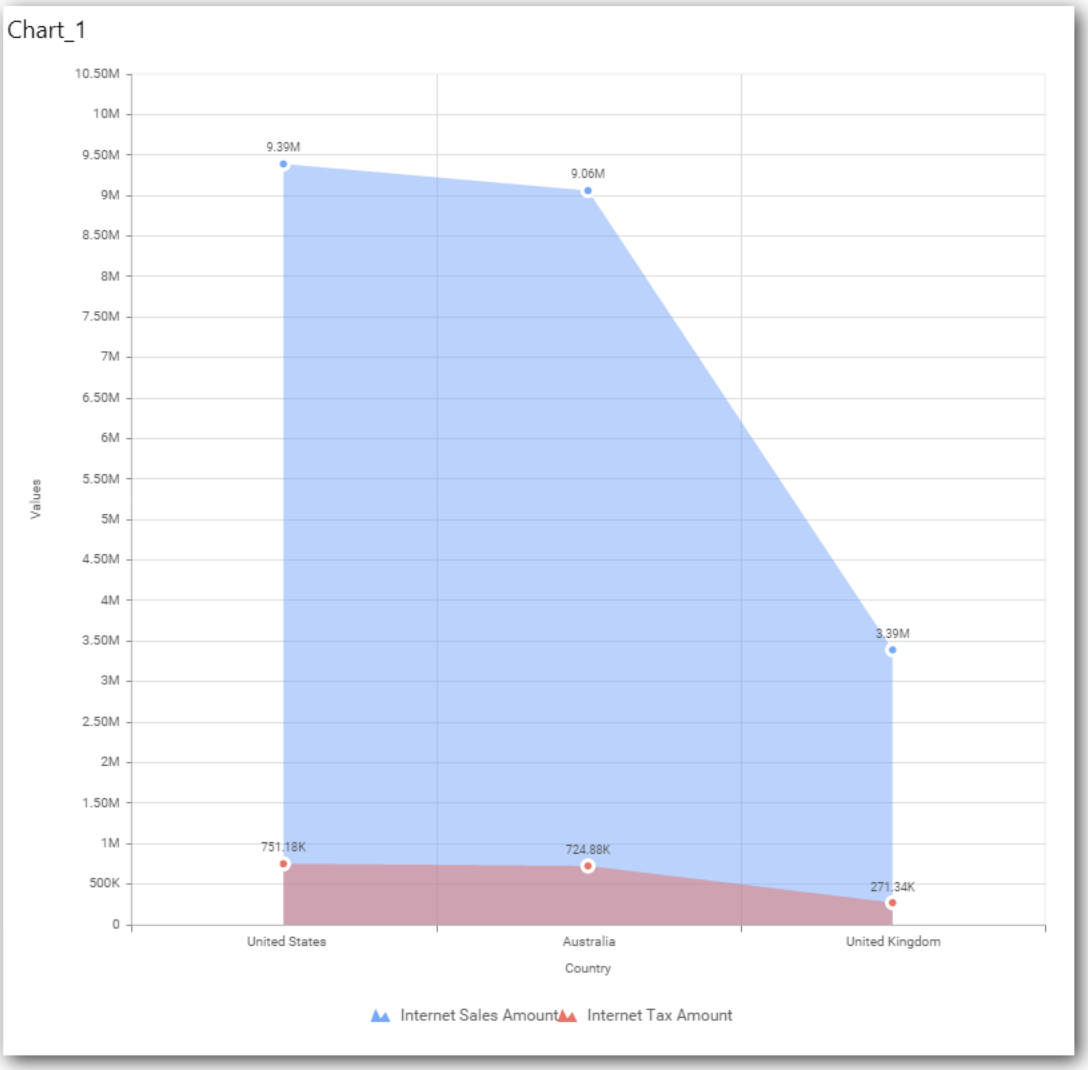
Select Filter(s)… to launch the Filters window.



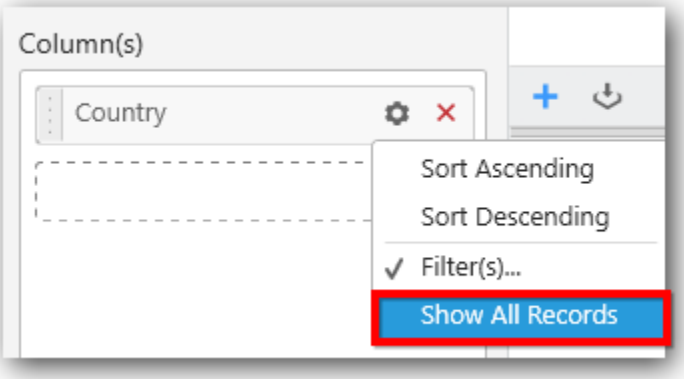
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

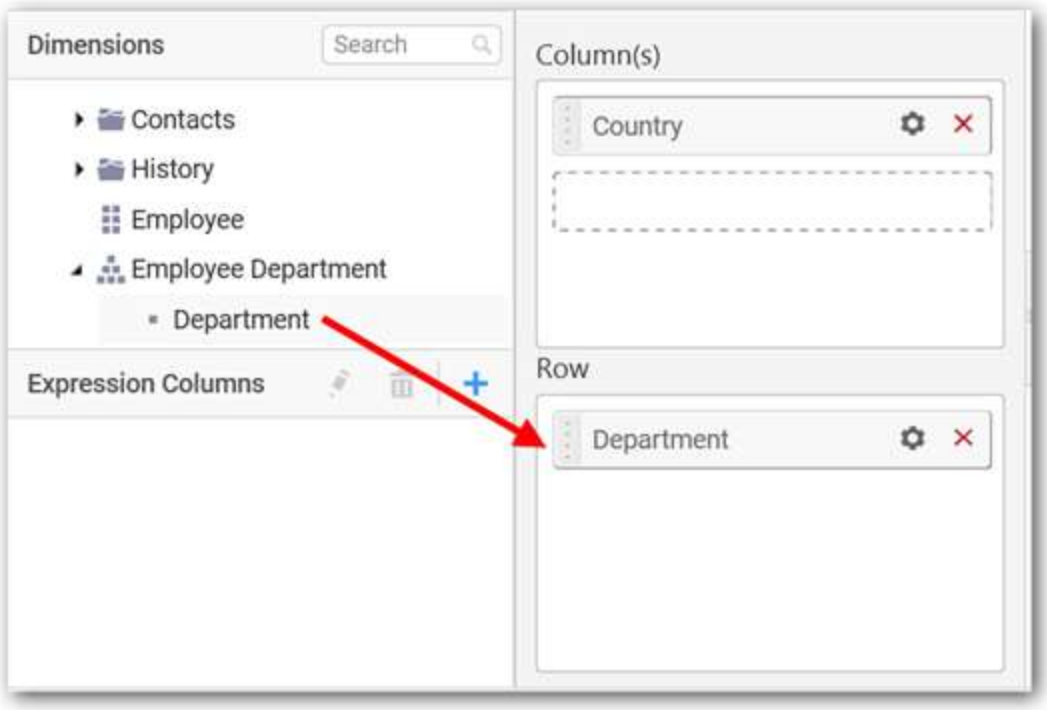


To show all records again click on Show All Records.

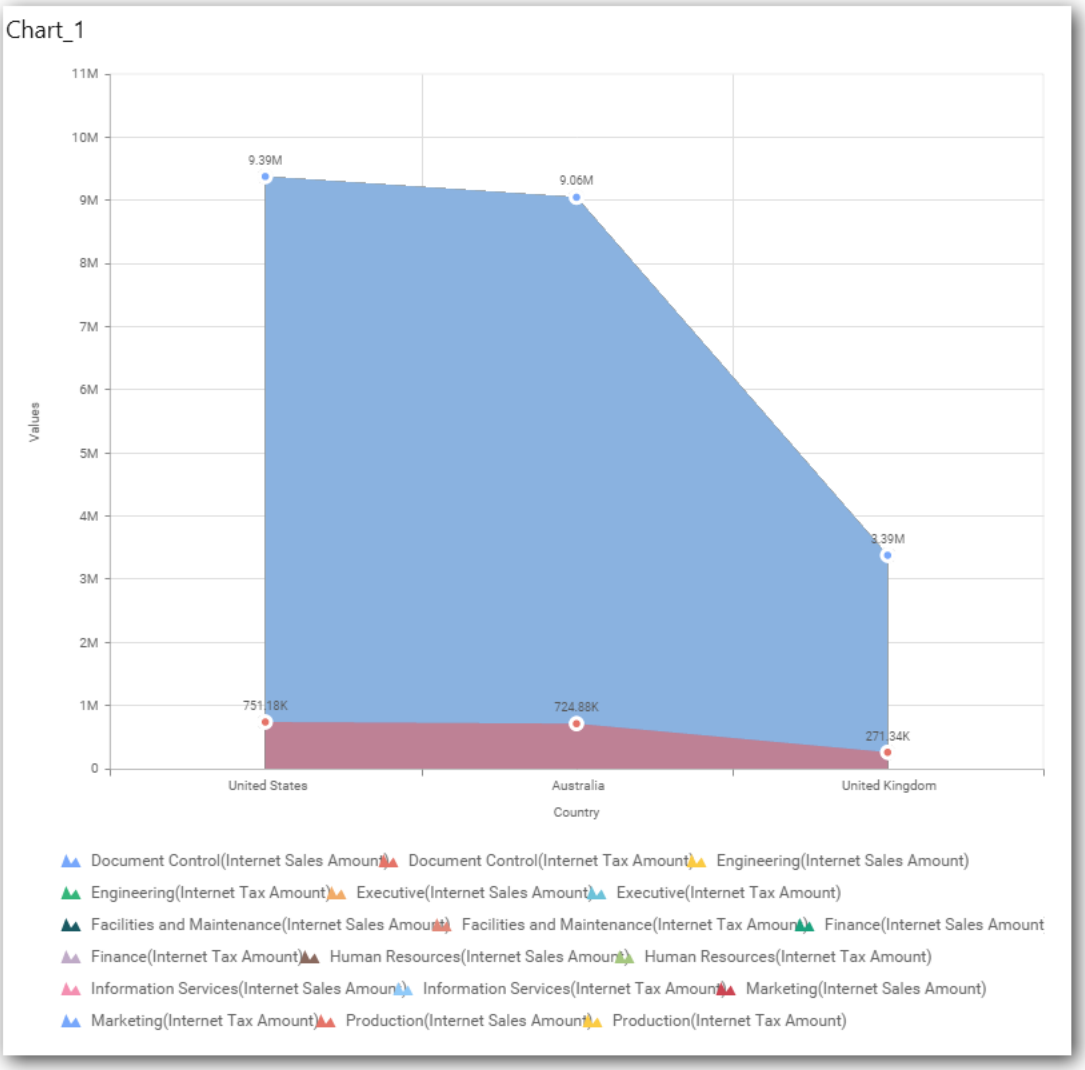


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.



How to format Area Chart?

You can format the Area chart for better illustration of the view that you require, through the settings available in Properties pane.

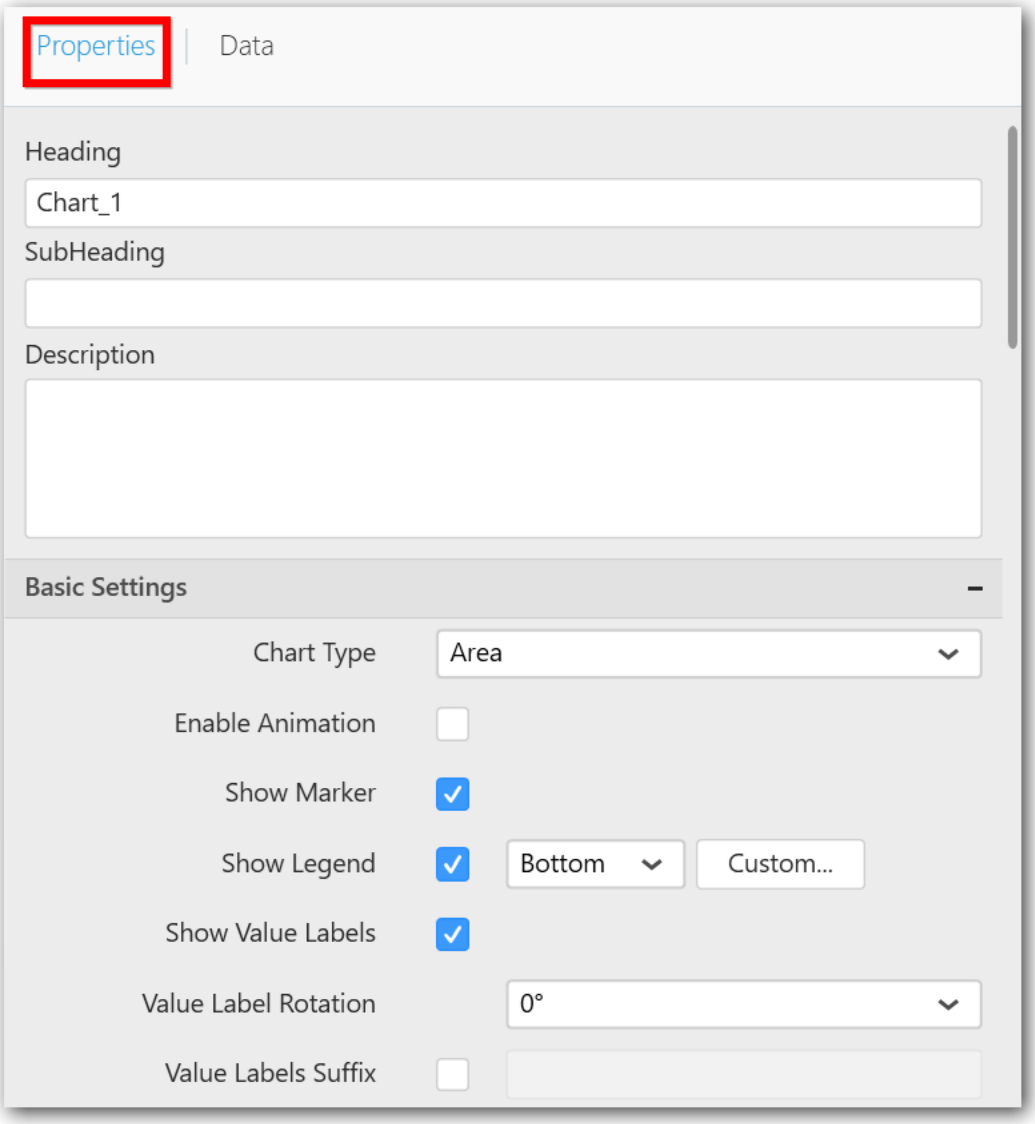
To format area chart follow the steps

Drag and drop the area chart into canvas and resize it to your required size.

Configure the data into bar chart.

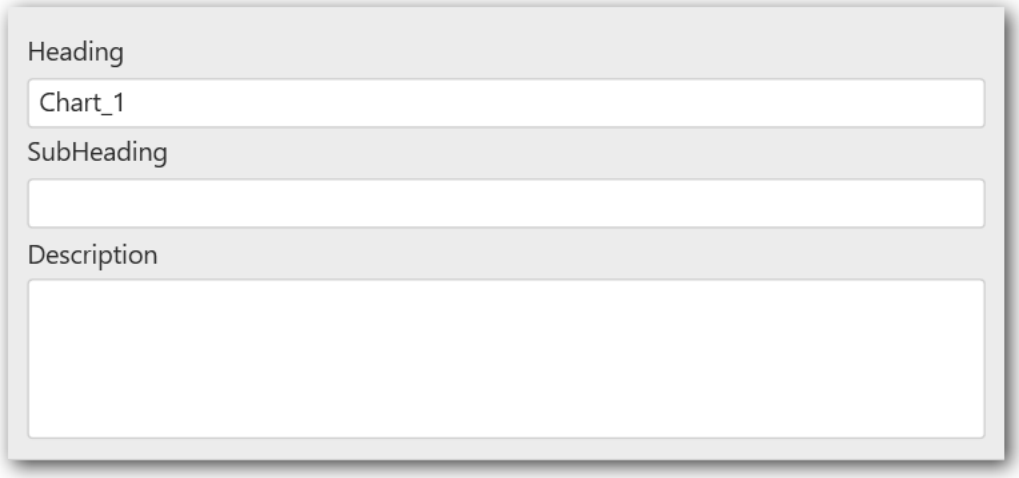
Focus on the area chart and Click on Widget Settings.

The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this area chart widget.

SubHeading

This allows you to set sub-title for this area chart widget.

Description

This allows you to set description for this area chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

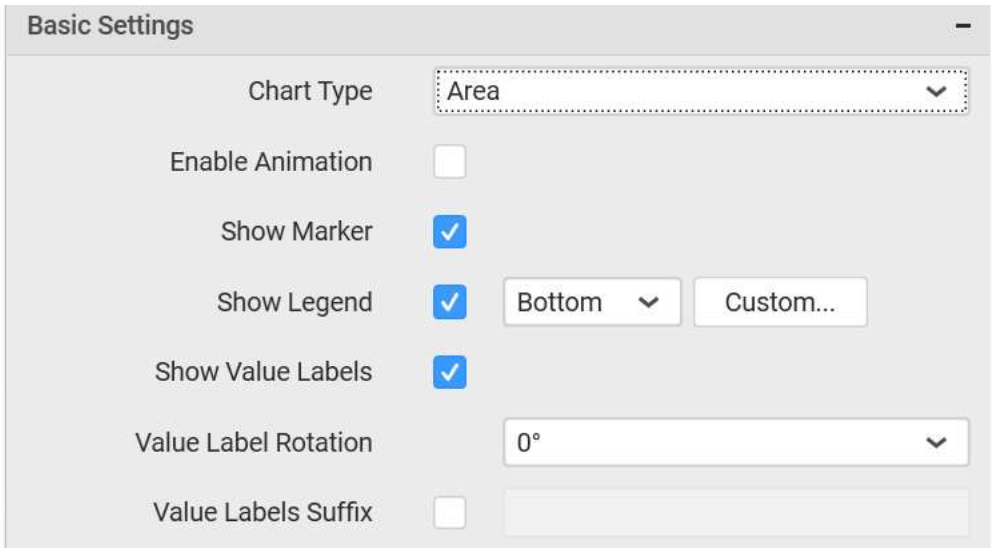


Chart Type

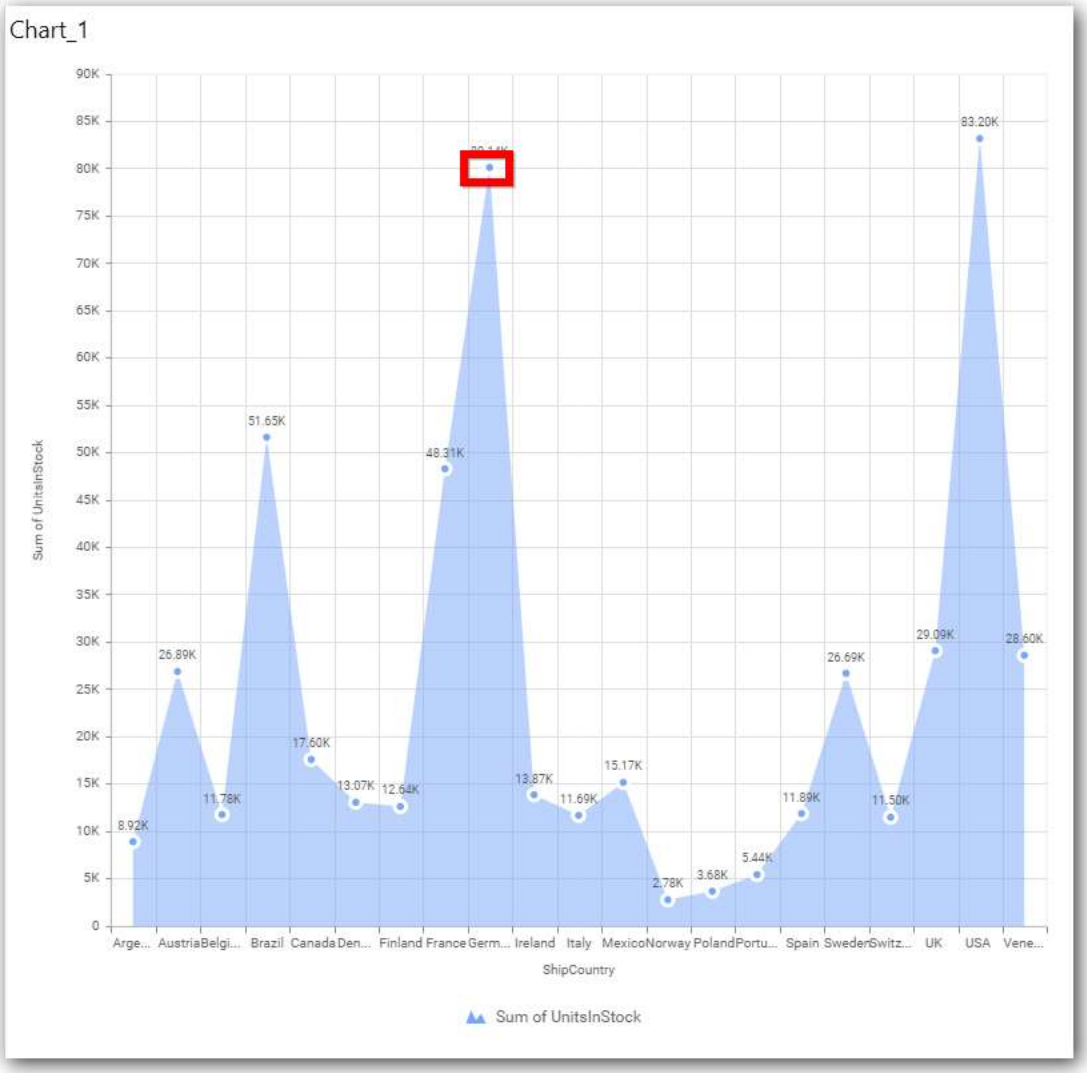
This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Marker

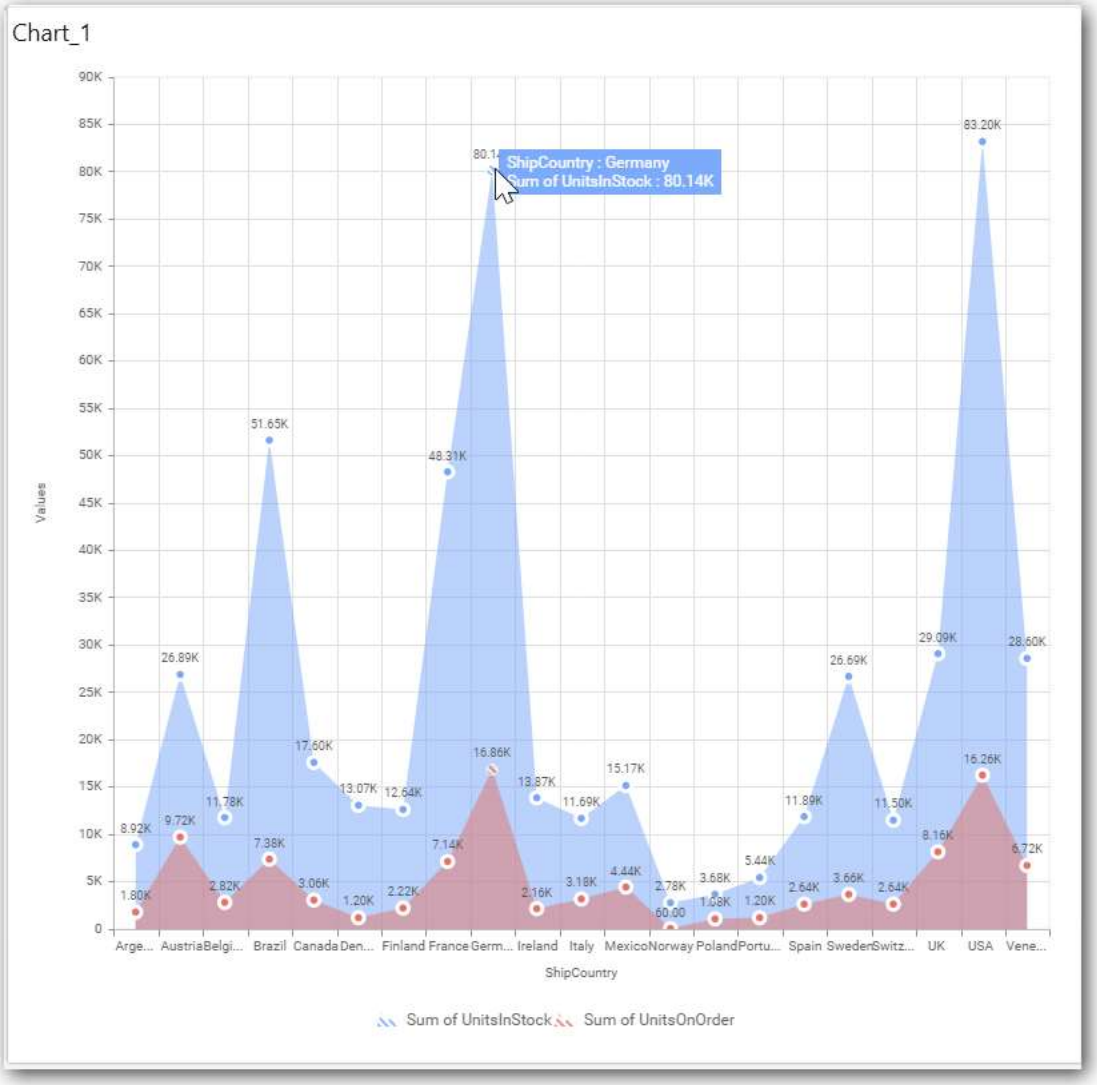
This allows you to toggle the visibility of marker from label to adorn each data point in chart series.



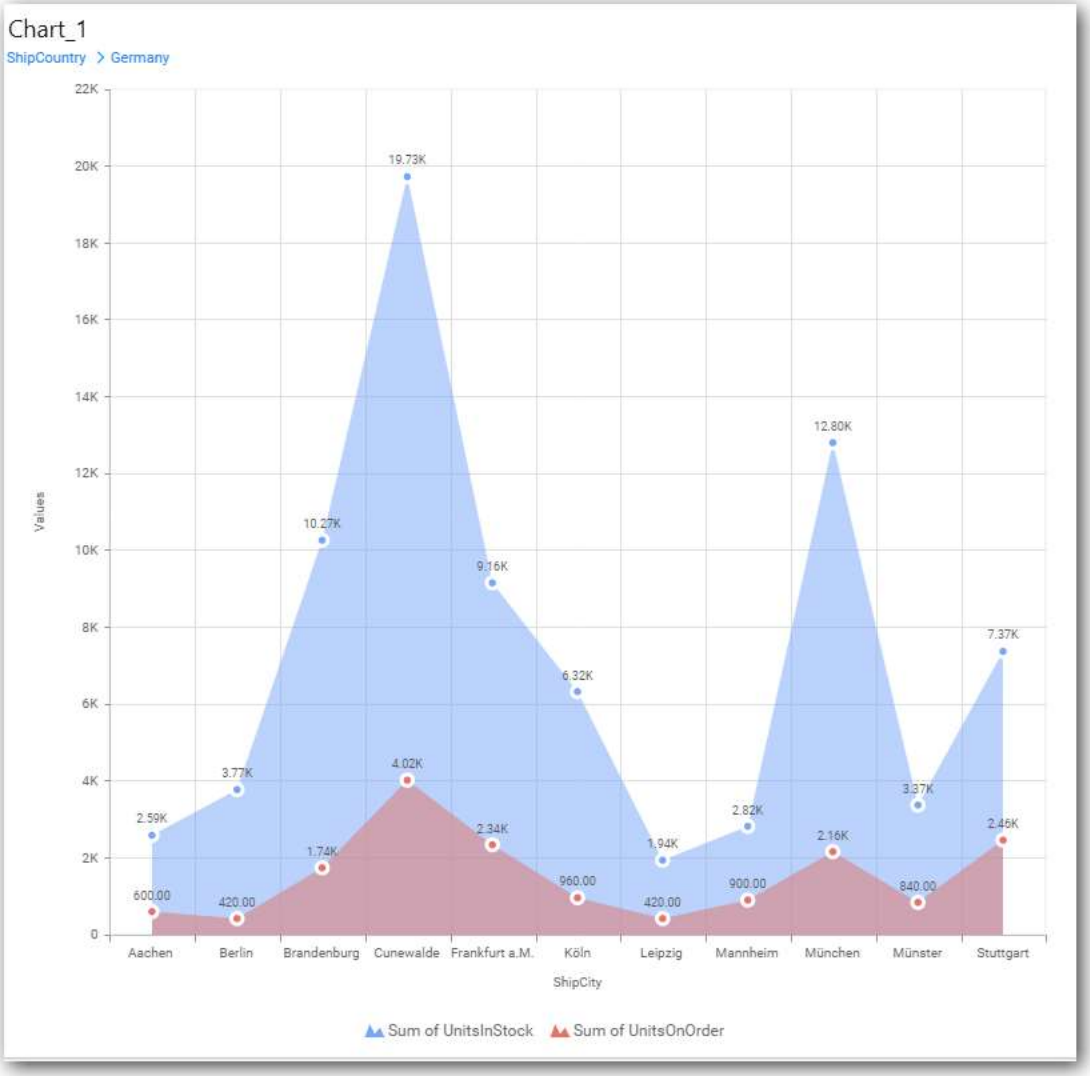
Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View

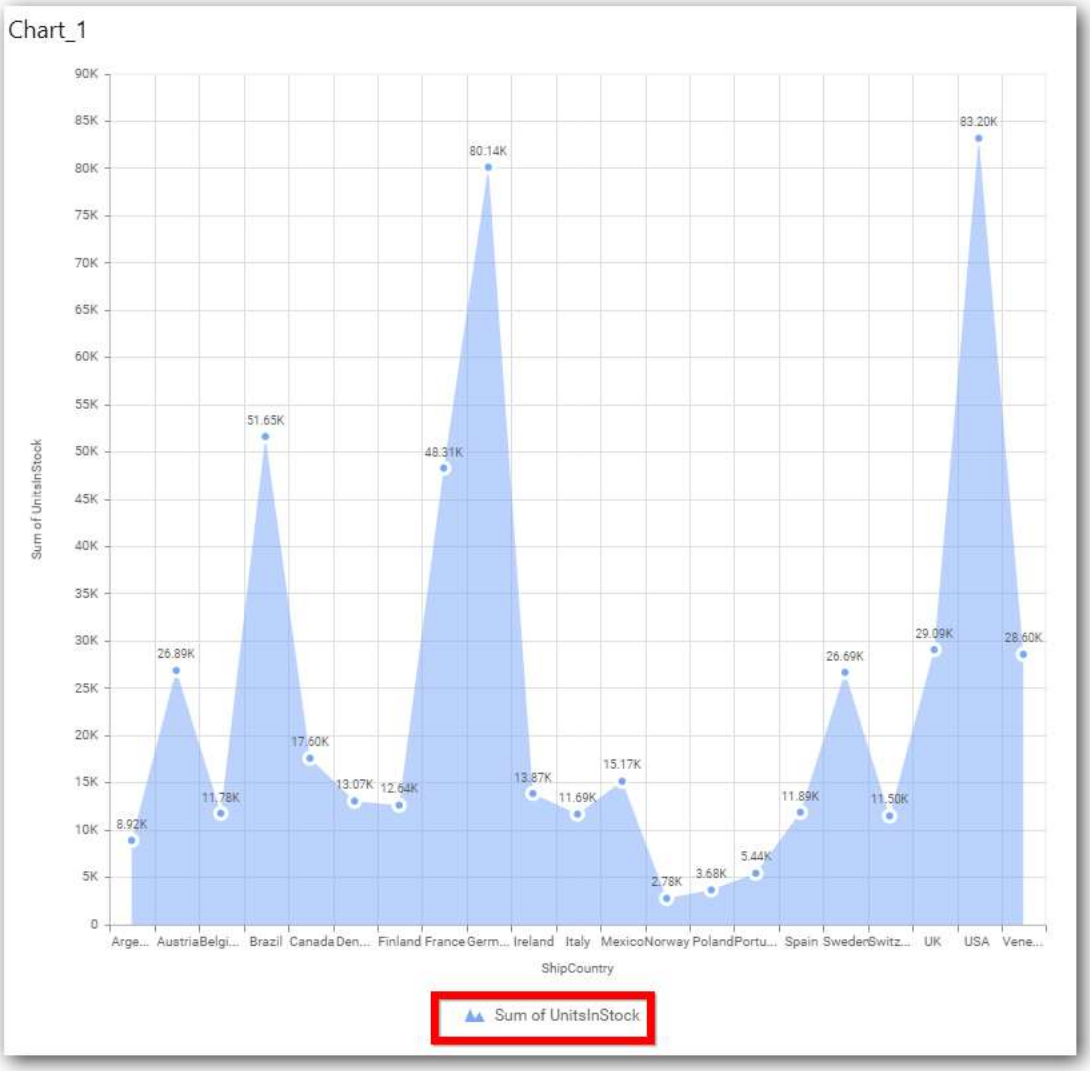


Drilled View



Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

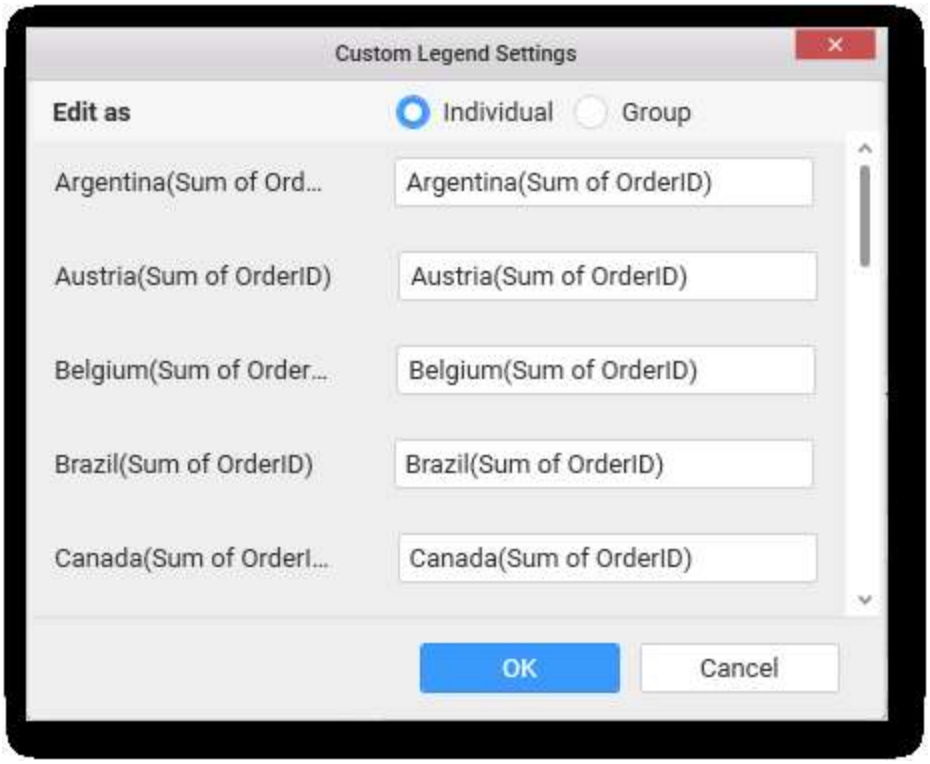
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

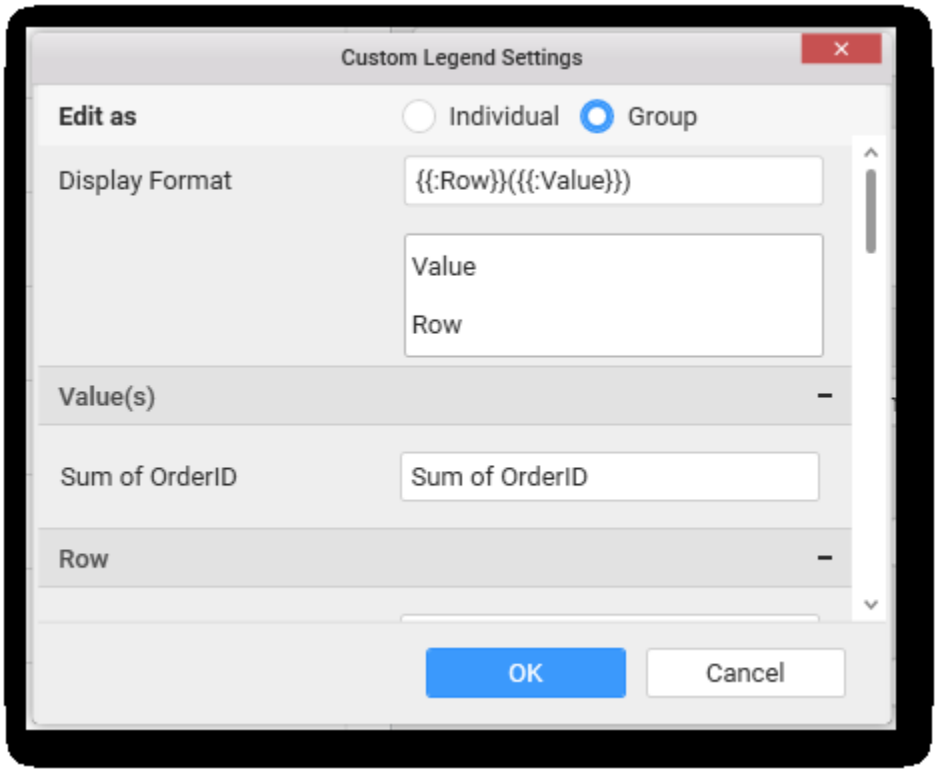
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

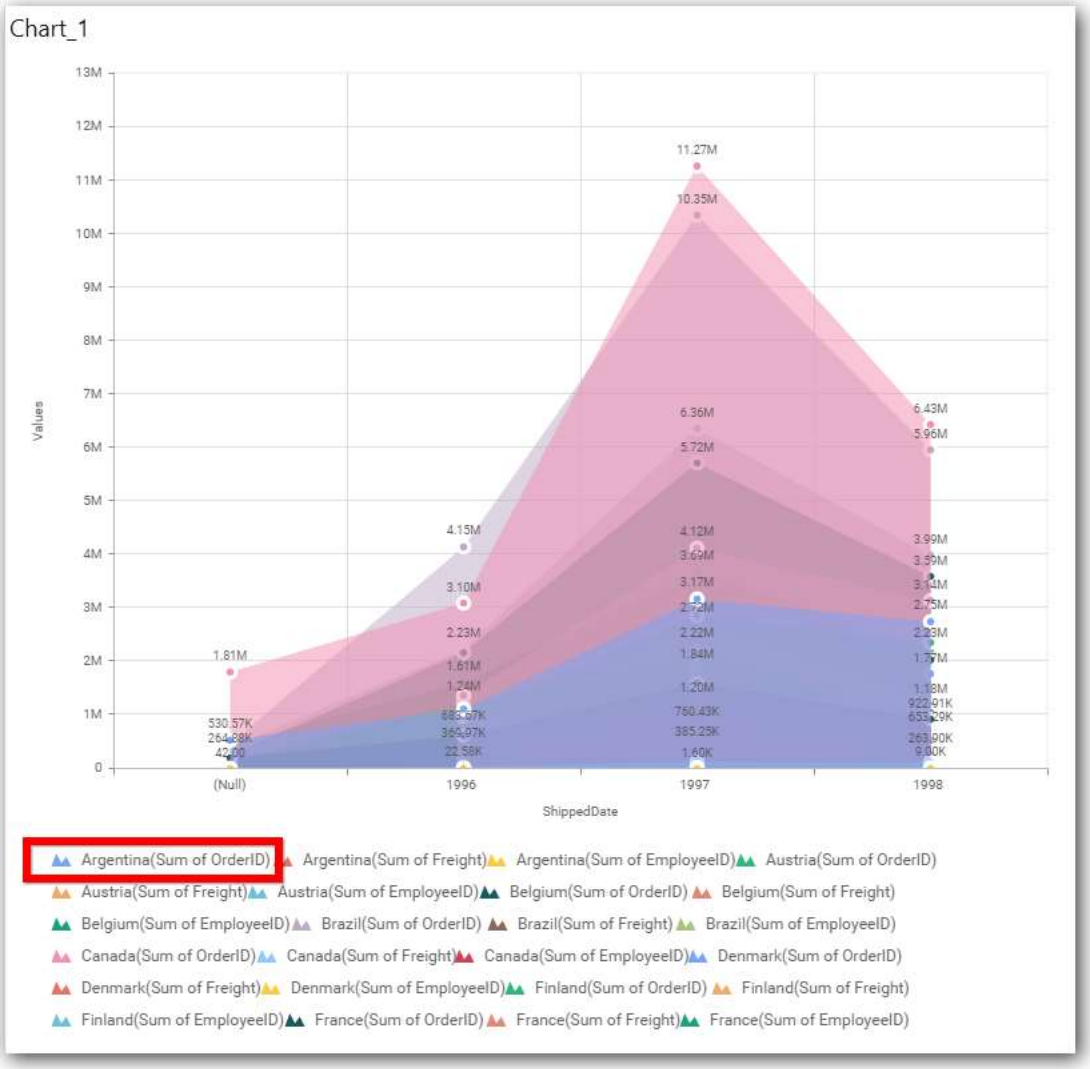


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

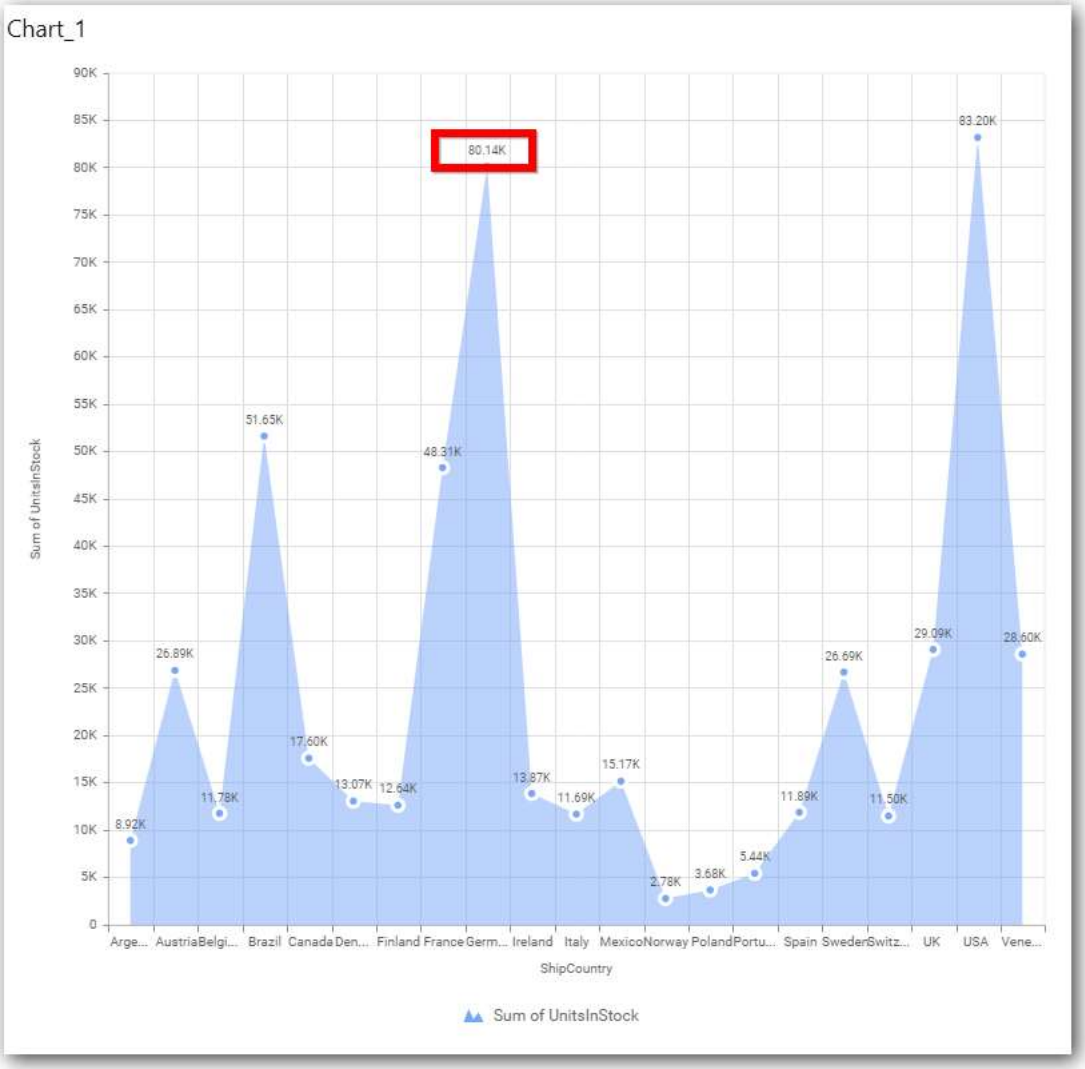


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



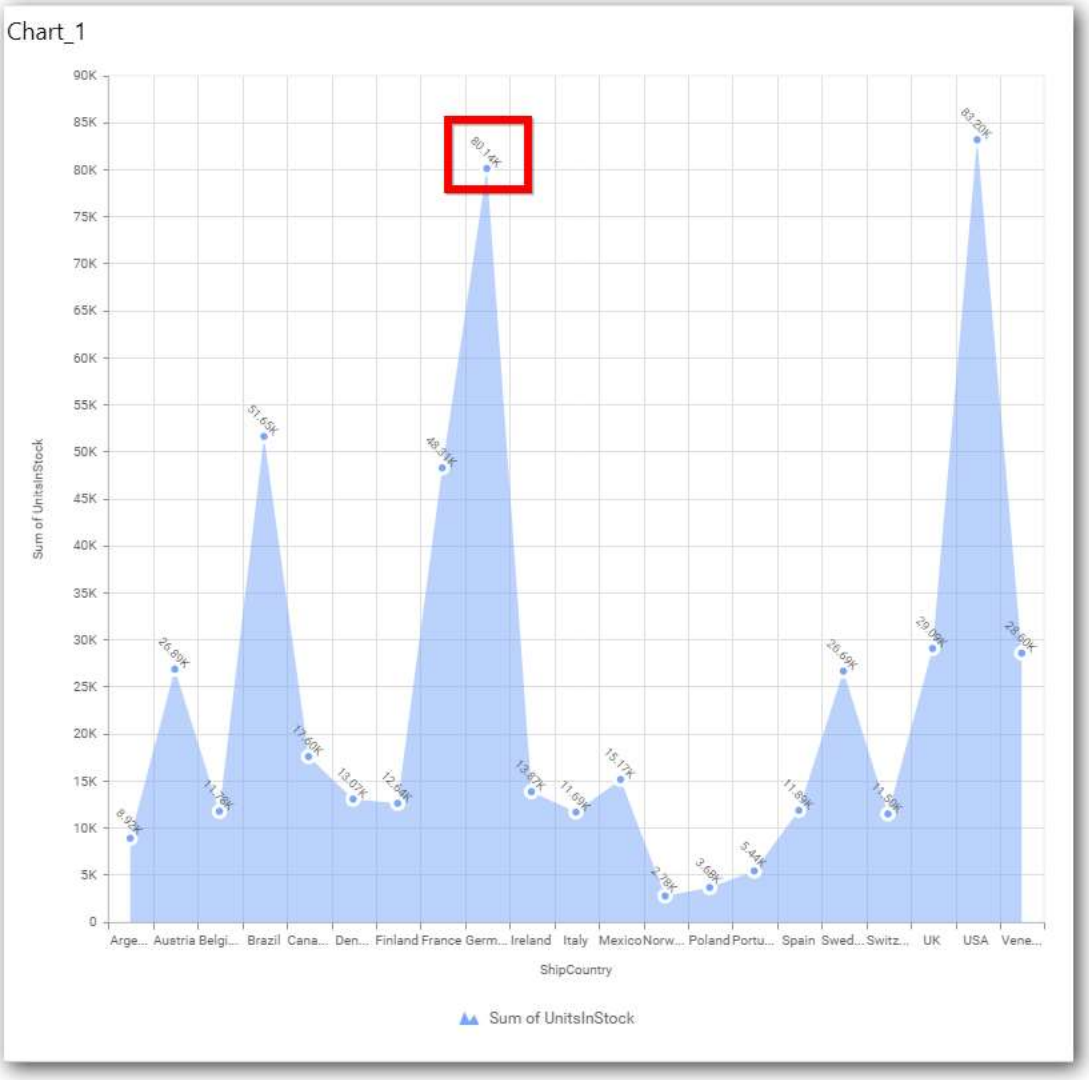
Show Value Labels

This allows you to toggle the visibility of value labels.



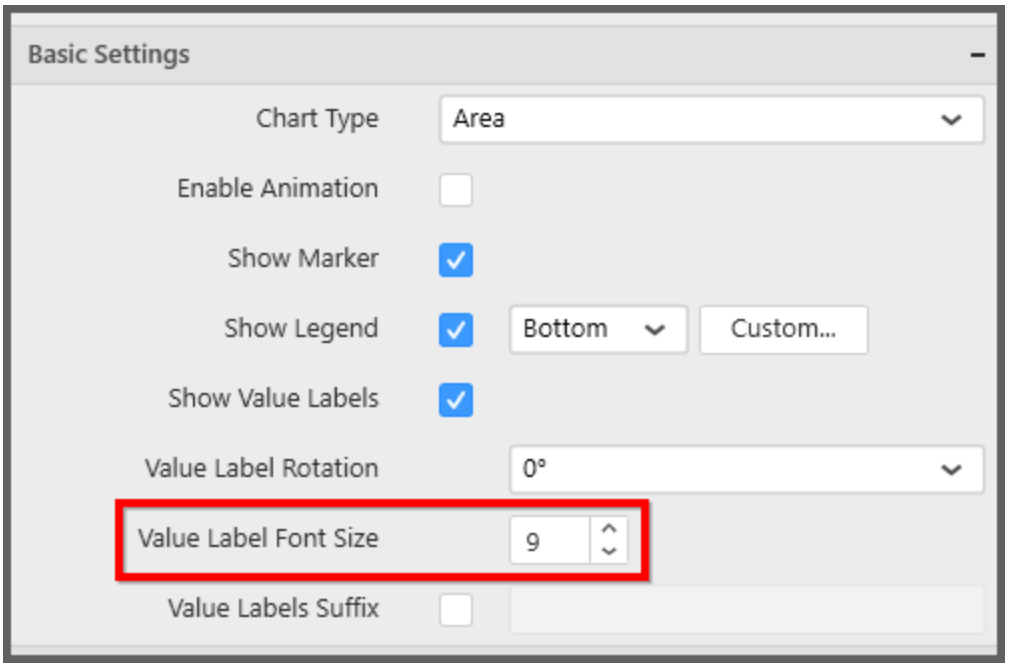
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



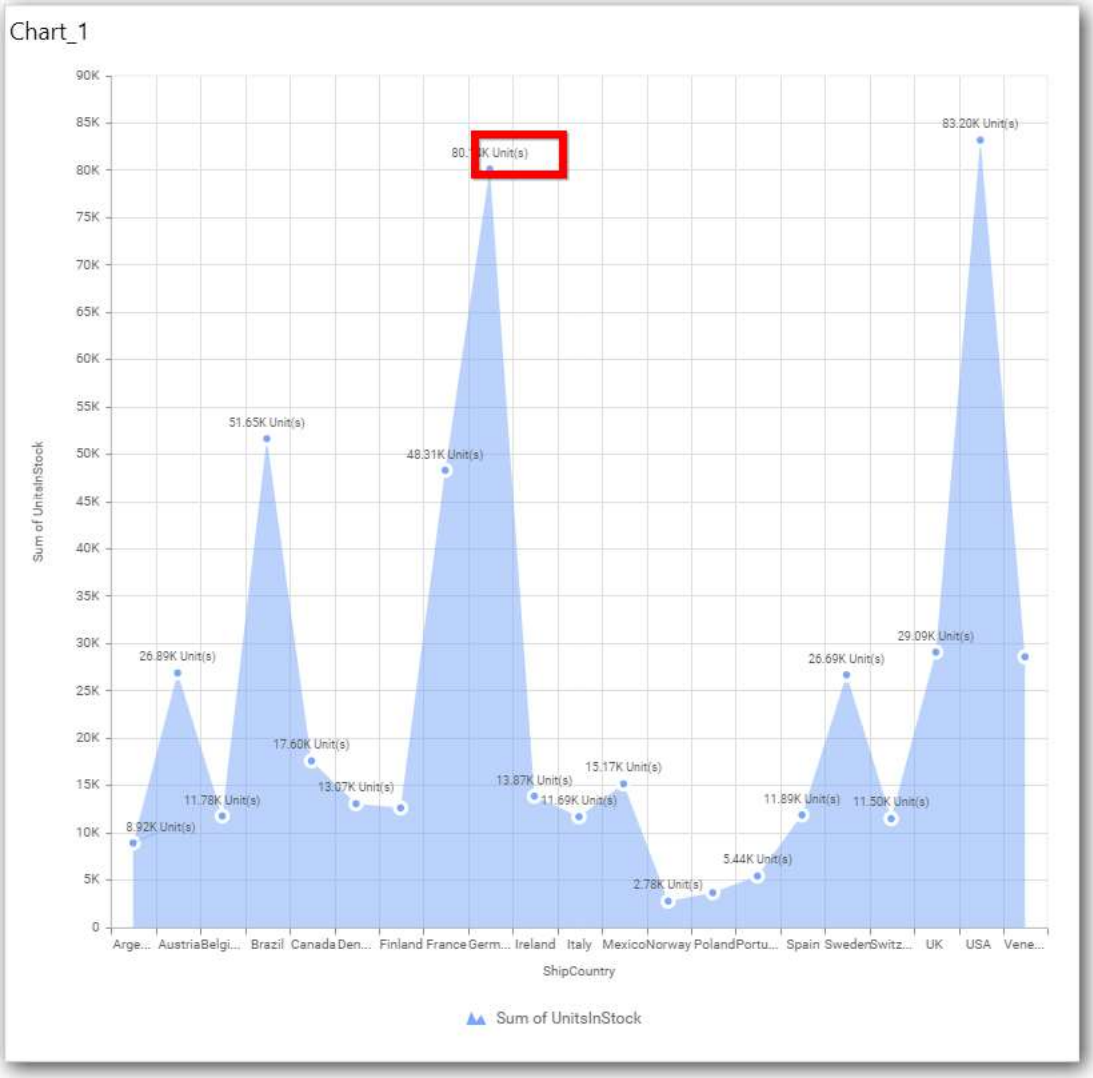
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

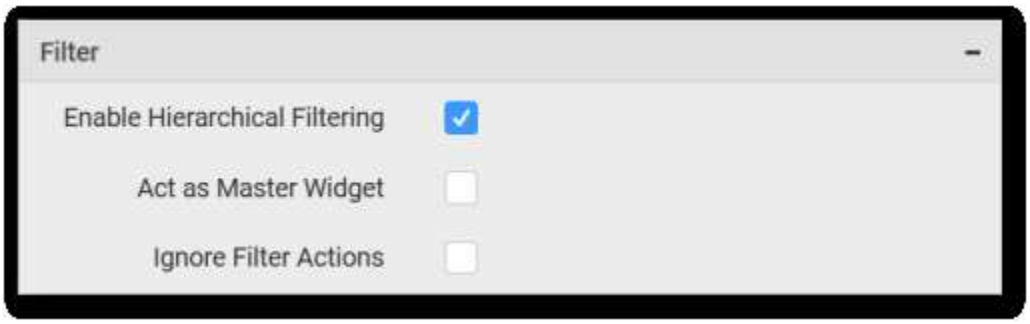


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter Settings



Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

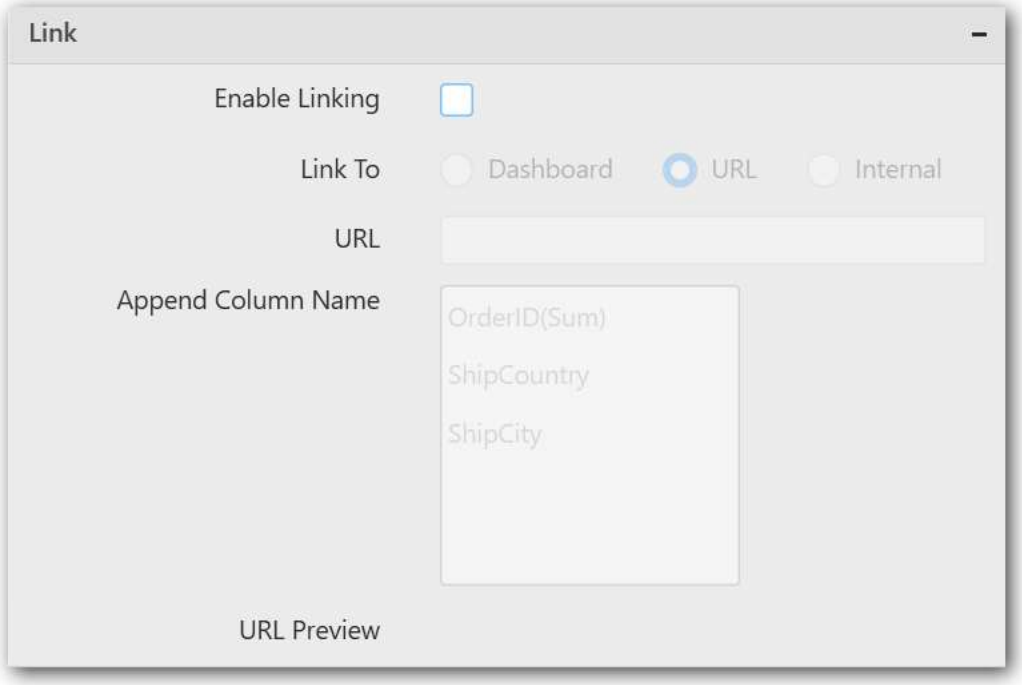
Act as Master Widget

This allows you to define this area chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

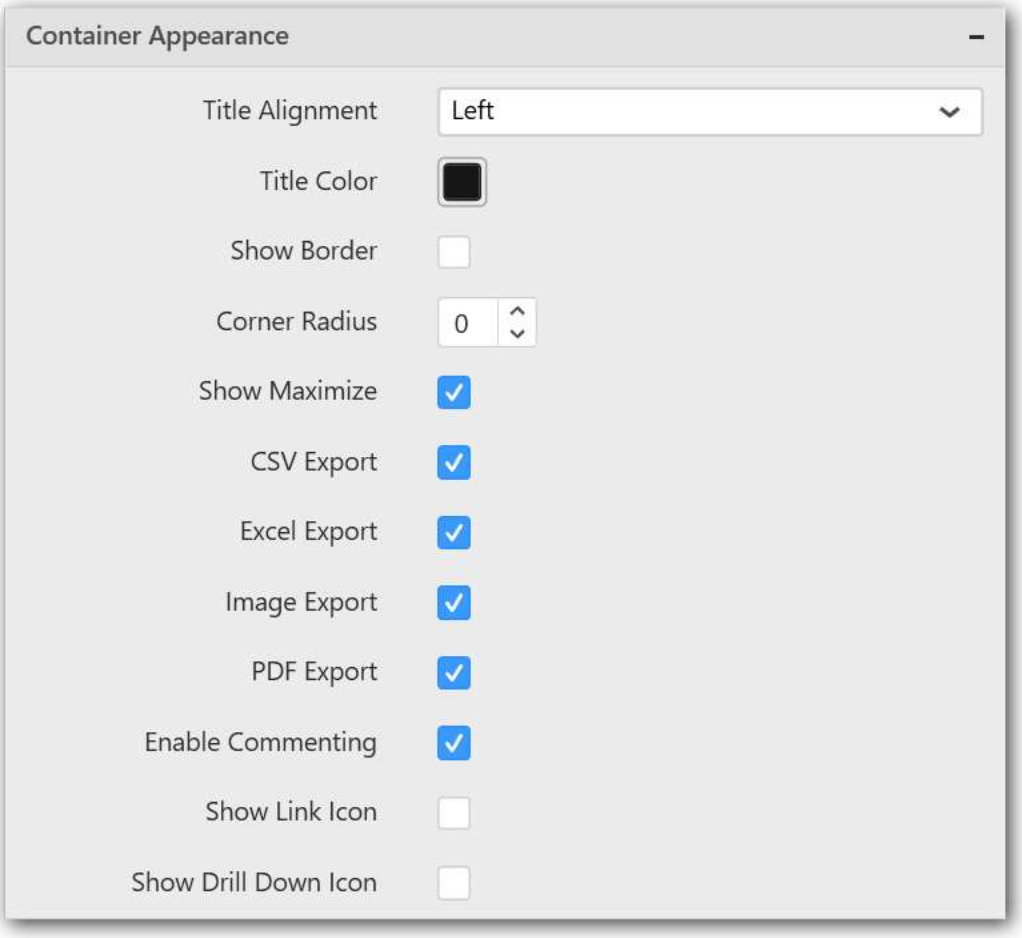
This allows you to define this area chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Border Visibility

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Maximized View

This allows you to enable/disable the maximized mode of this area chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this area chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this area chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

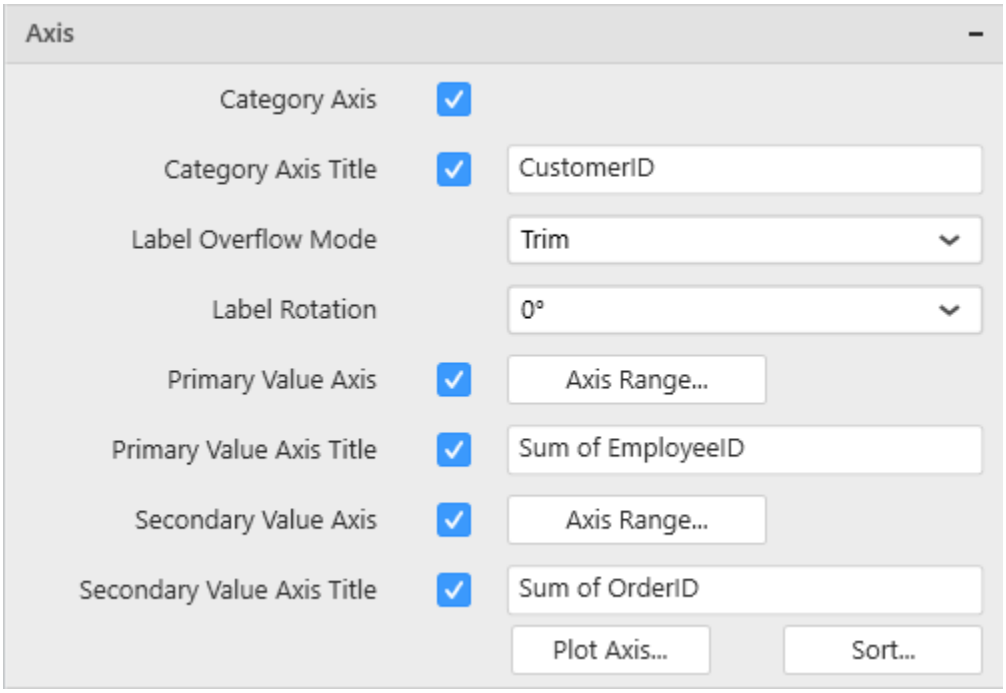
Image Export

This allows you to enable/disable the image export option for this area chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

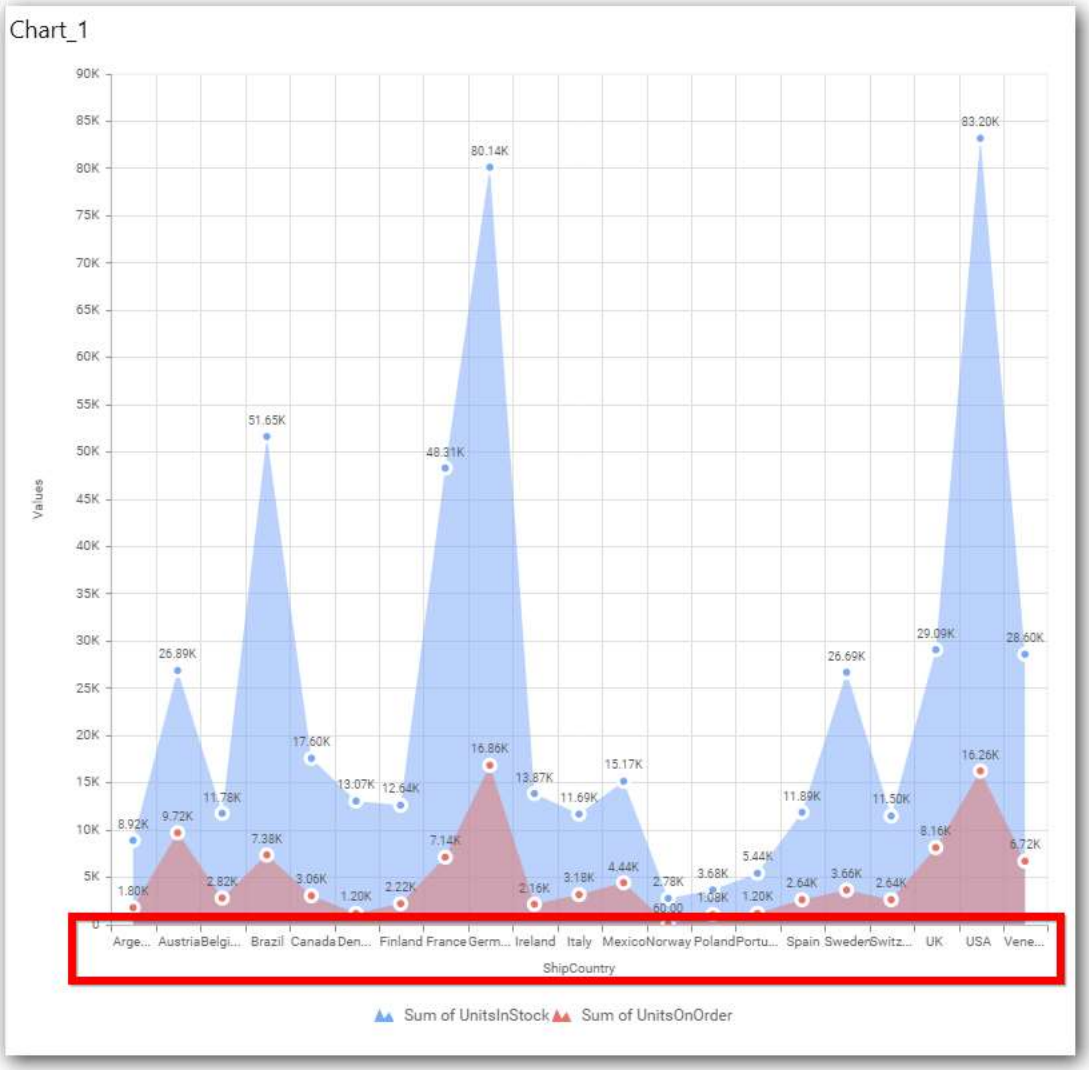
Axis Settings



This section allows you to customize the axis settings in chart.

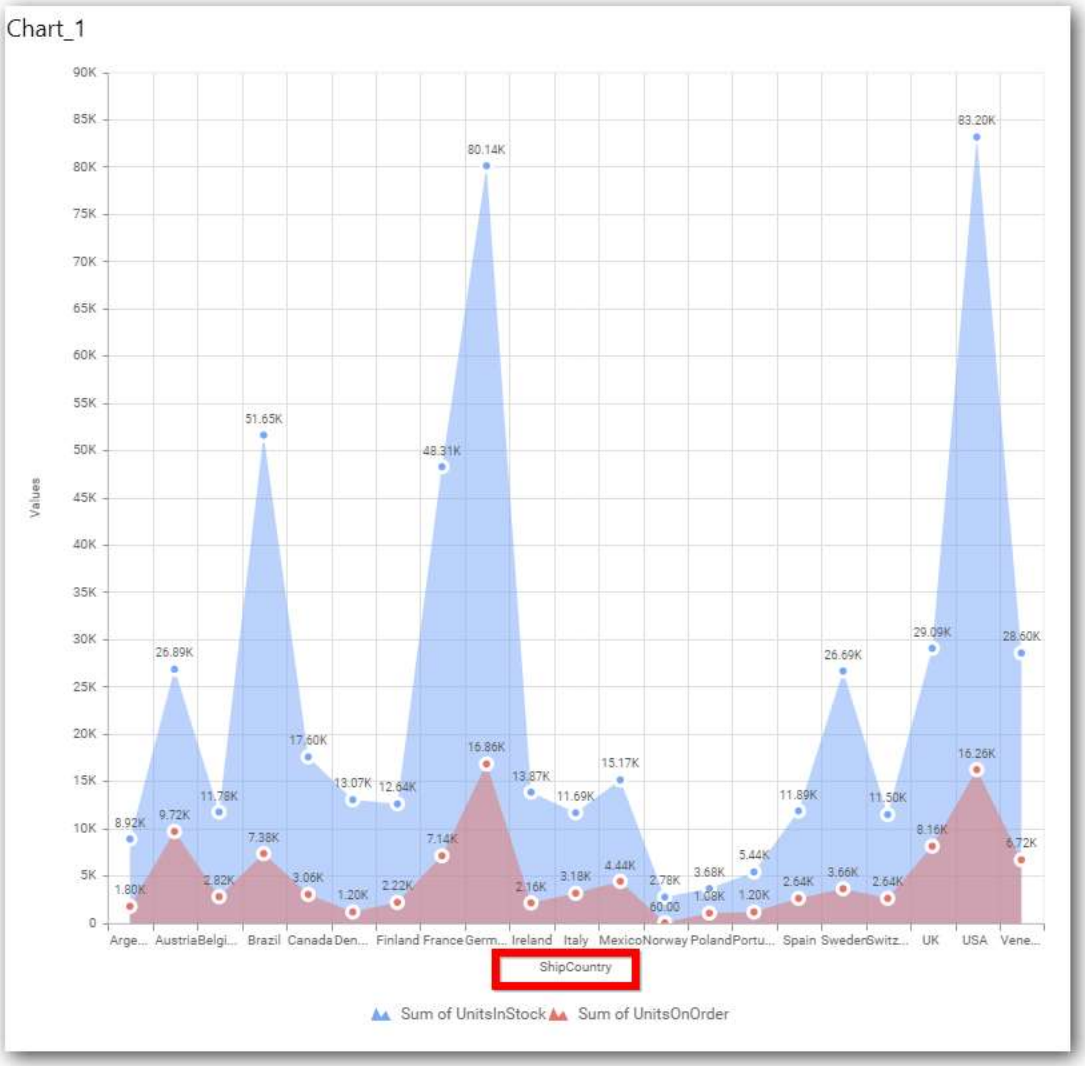
Category Axis

This allows to enable or edit the option of Category Axis. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

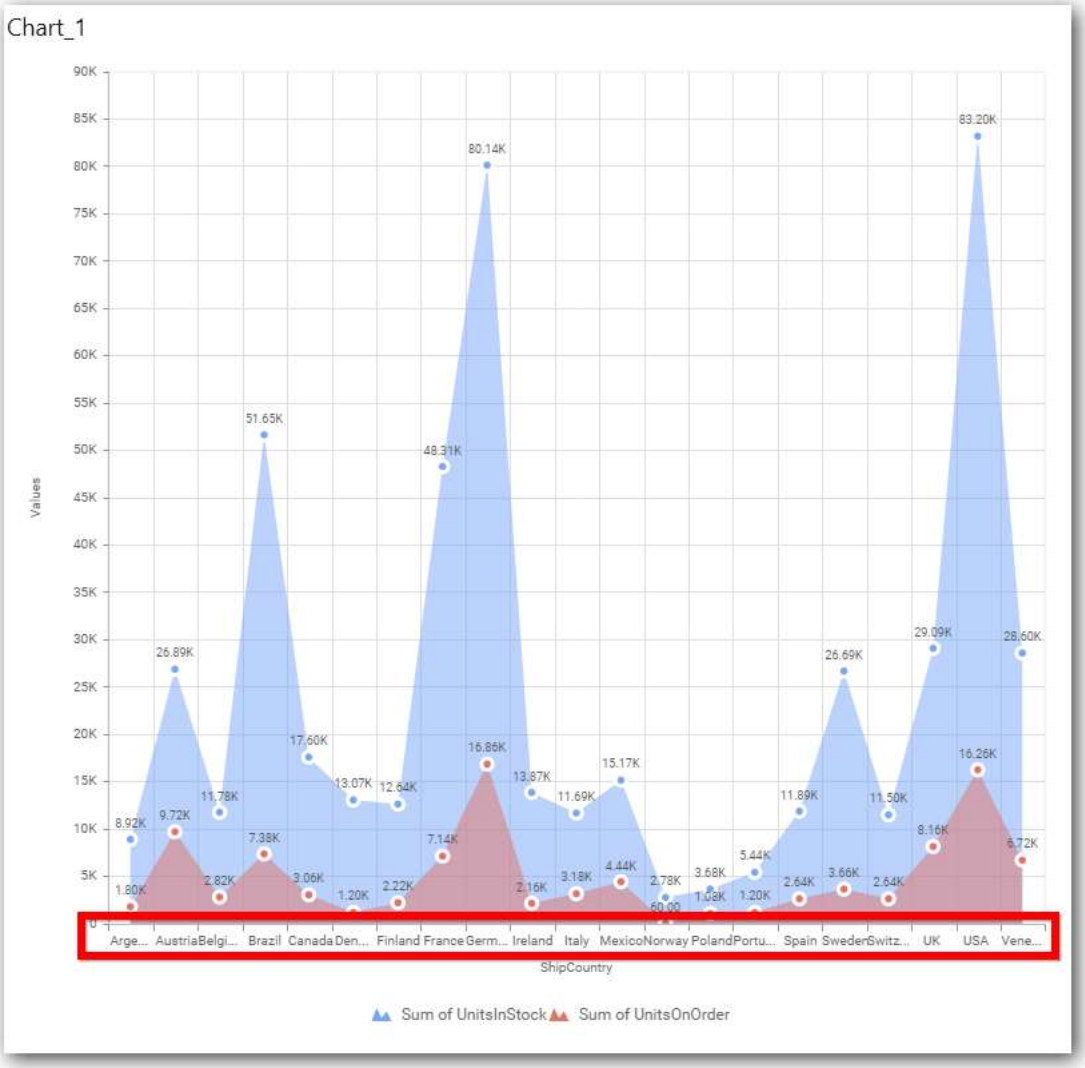


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

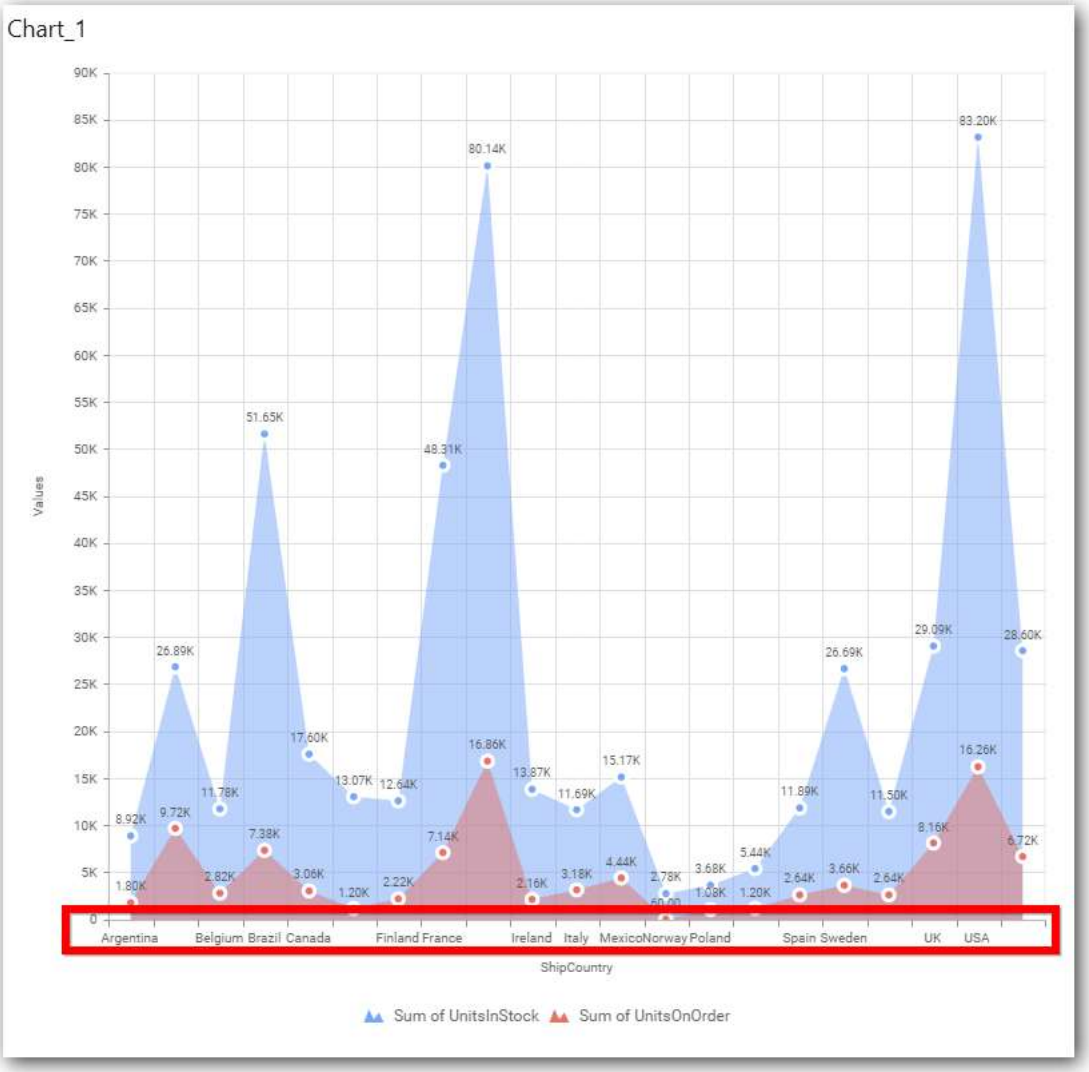
Trim

This option trims the end of overlapping label in the axis.



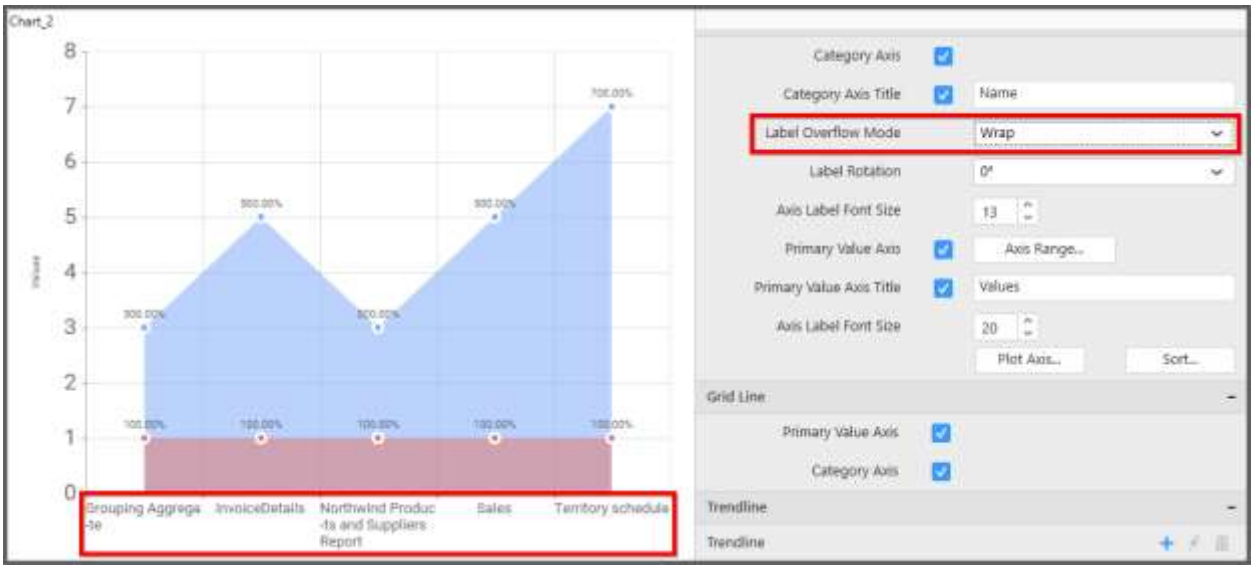
Hide

This option hides the overlapping label in the axis.



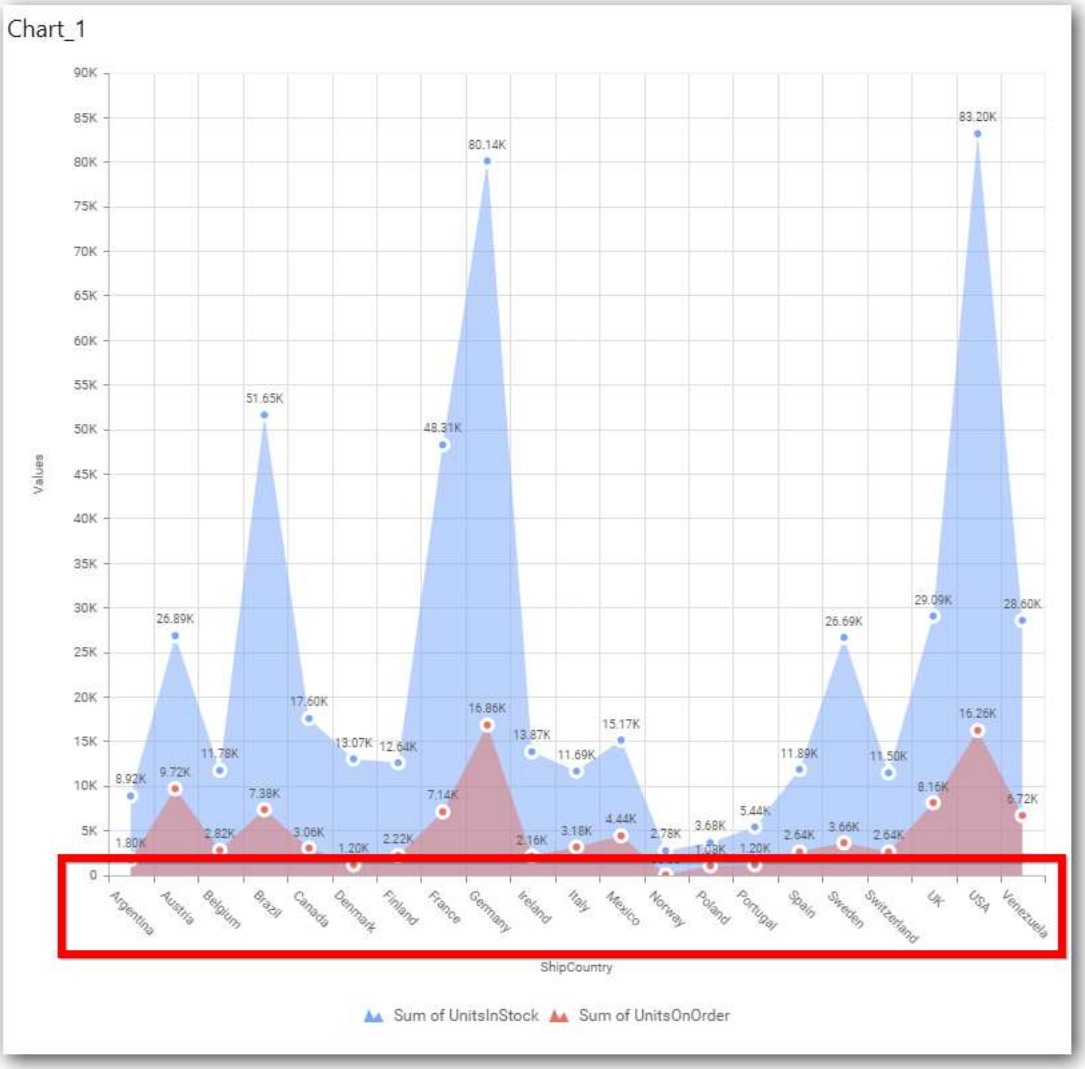
Wrap

This option wraps the lengthy label text in the axis.



Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

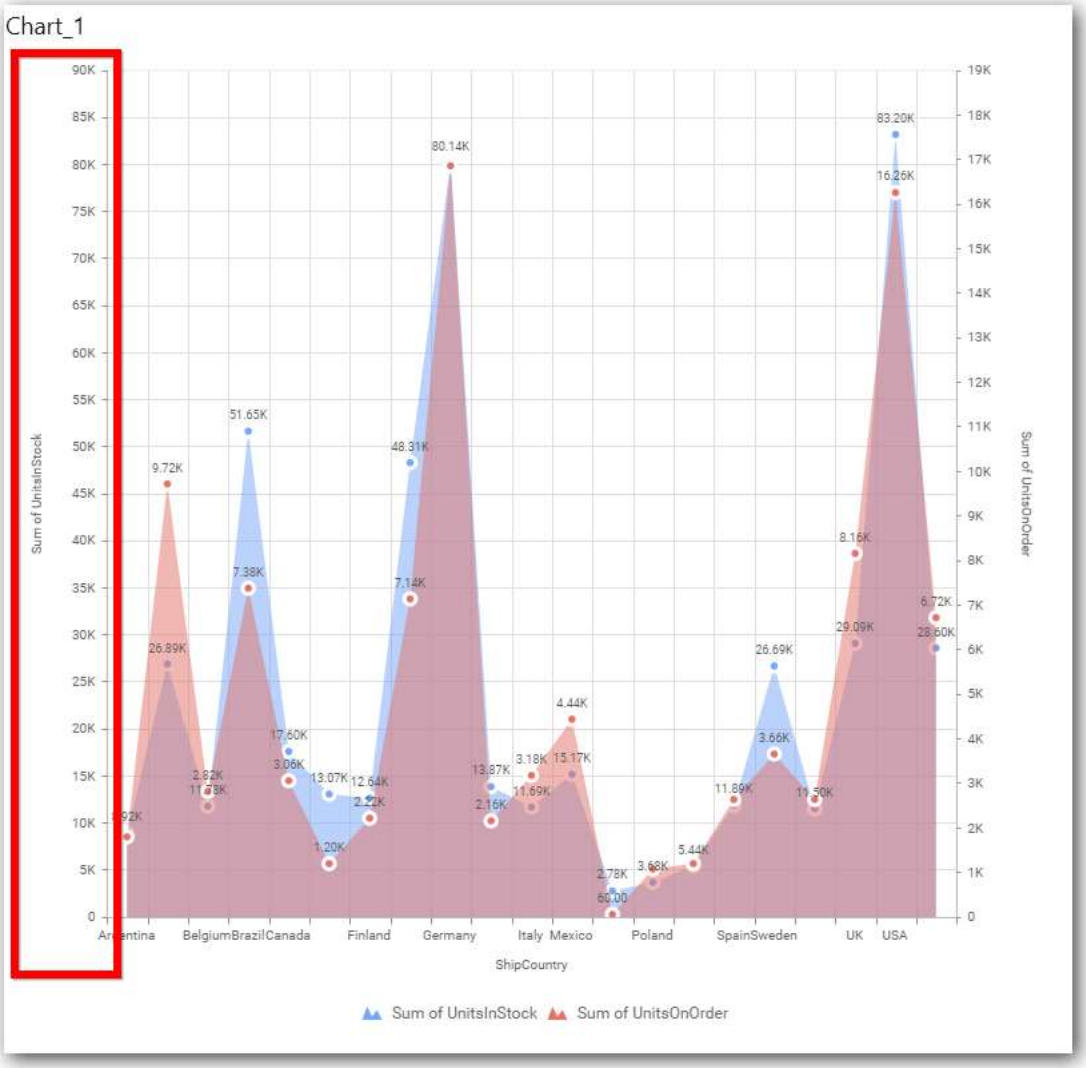


Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.

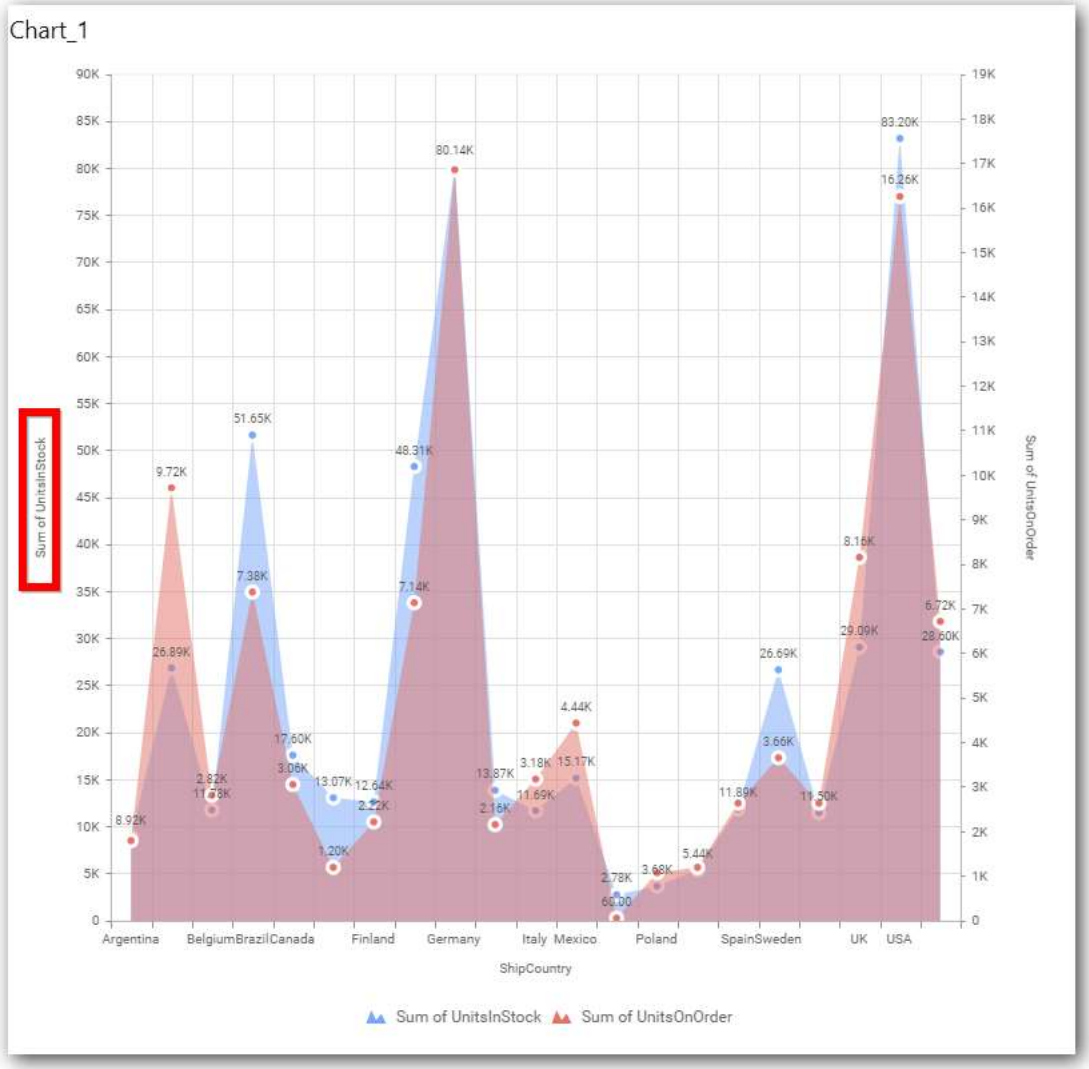
Primary Value Axis

This allows you to enable and edit the Primary Value Axis title. It will reflect in chart area y-axis name.



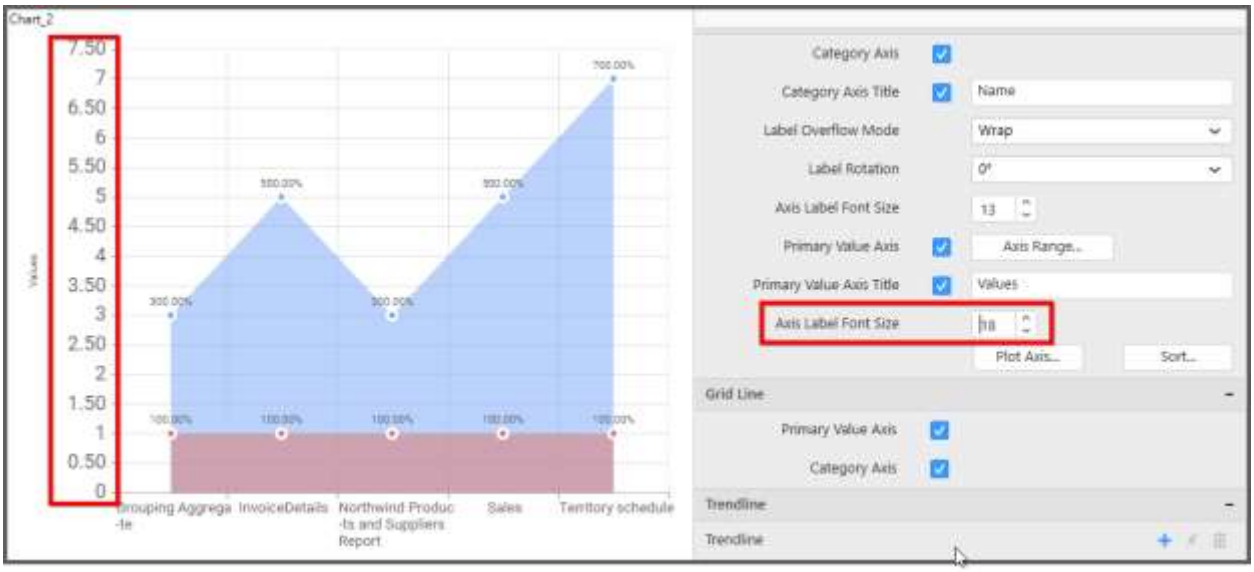
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

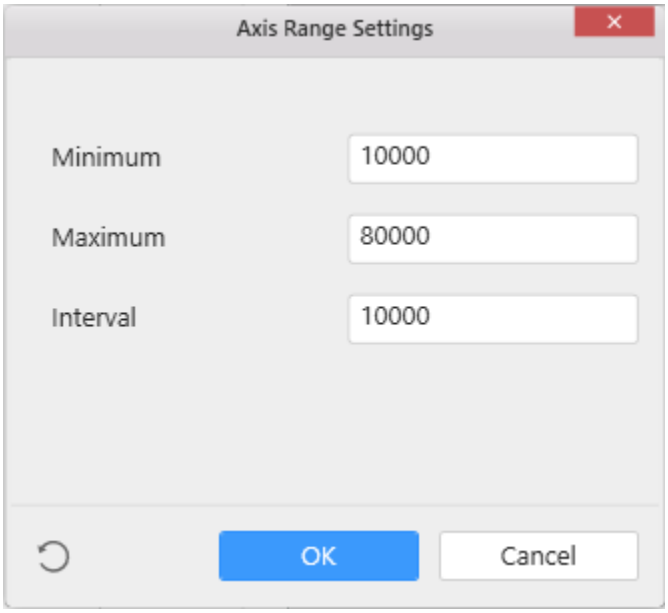


Primary Value Axis Range

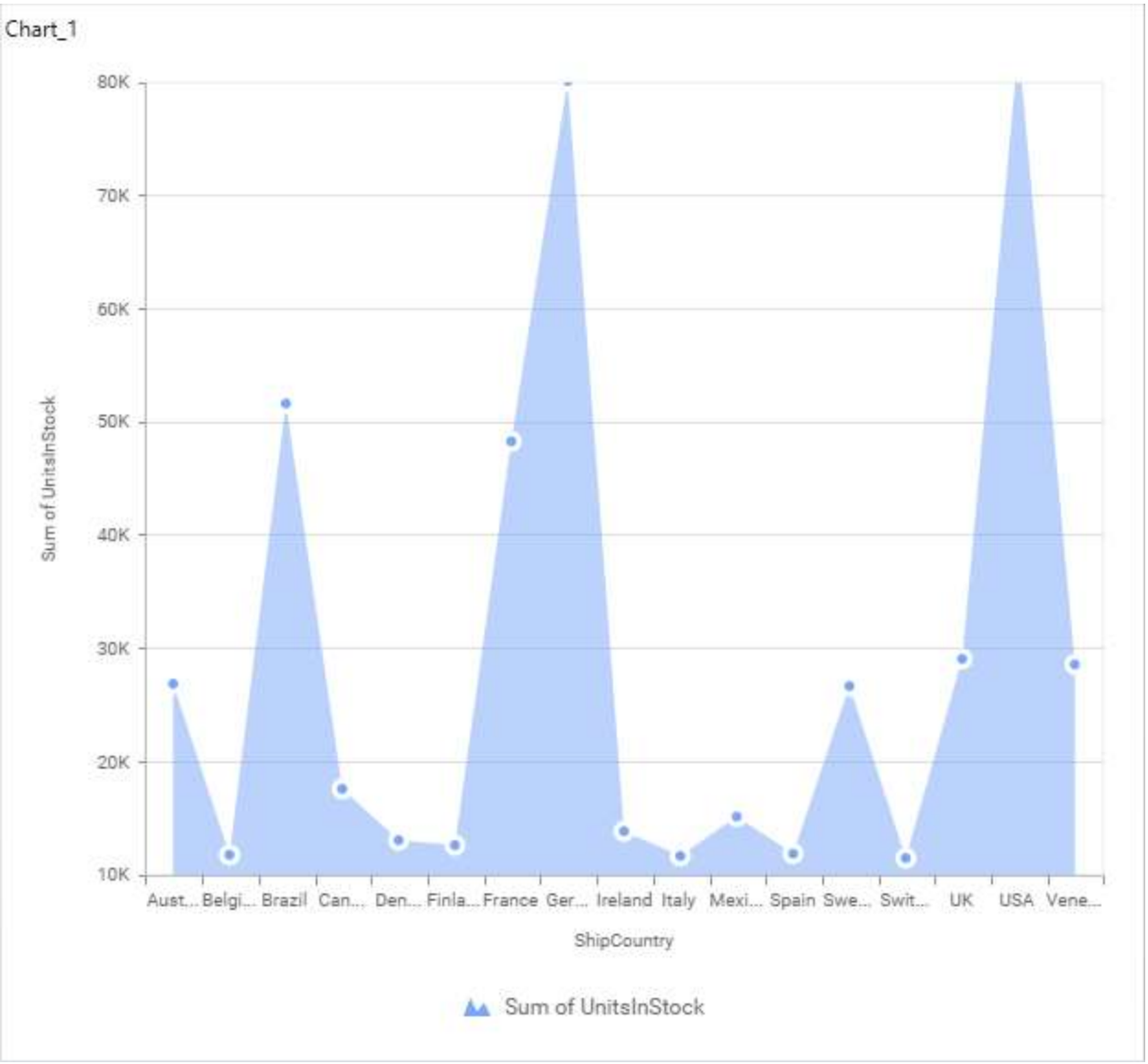
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.

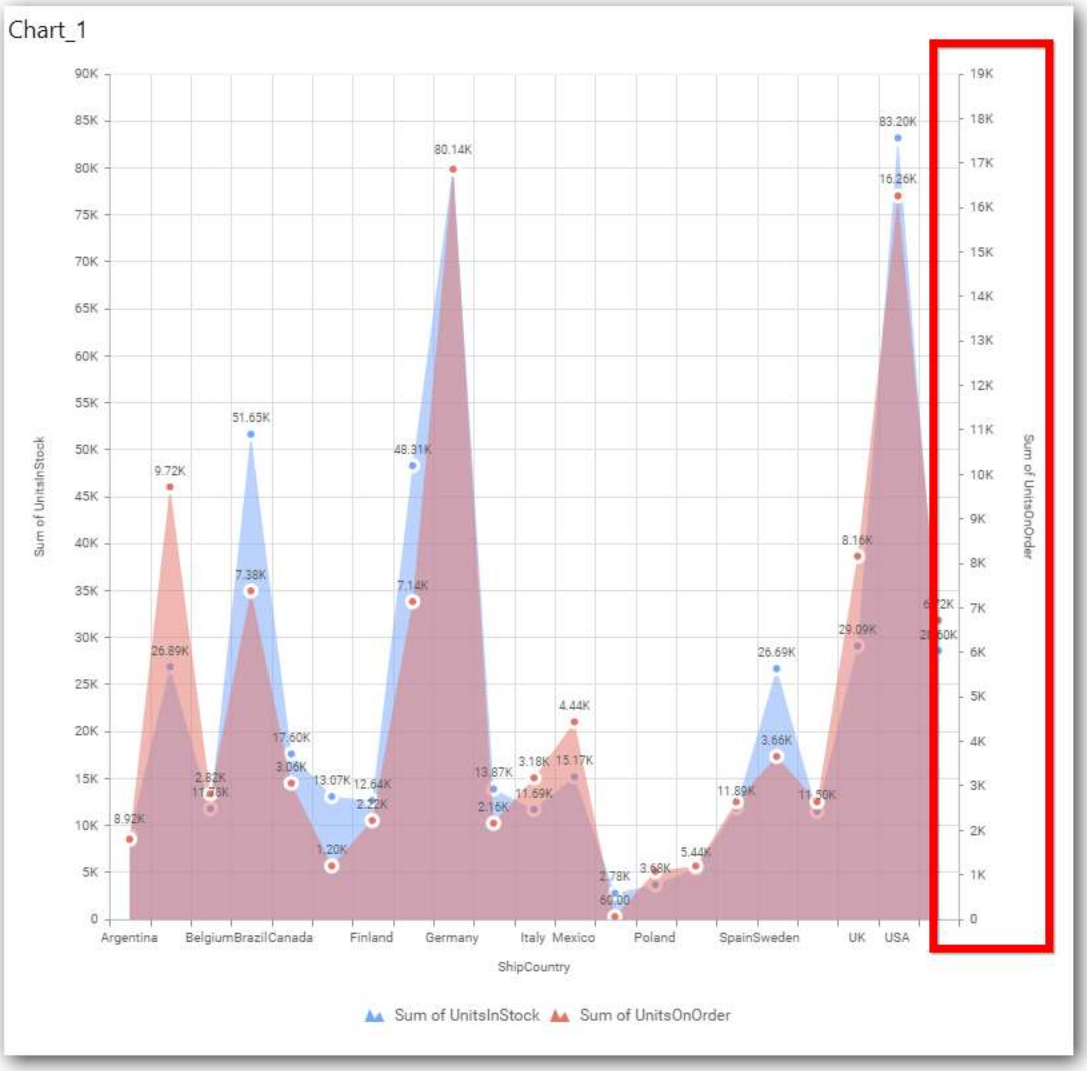


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



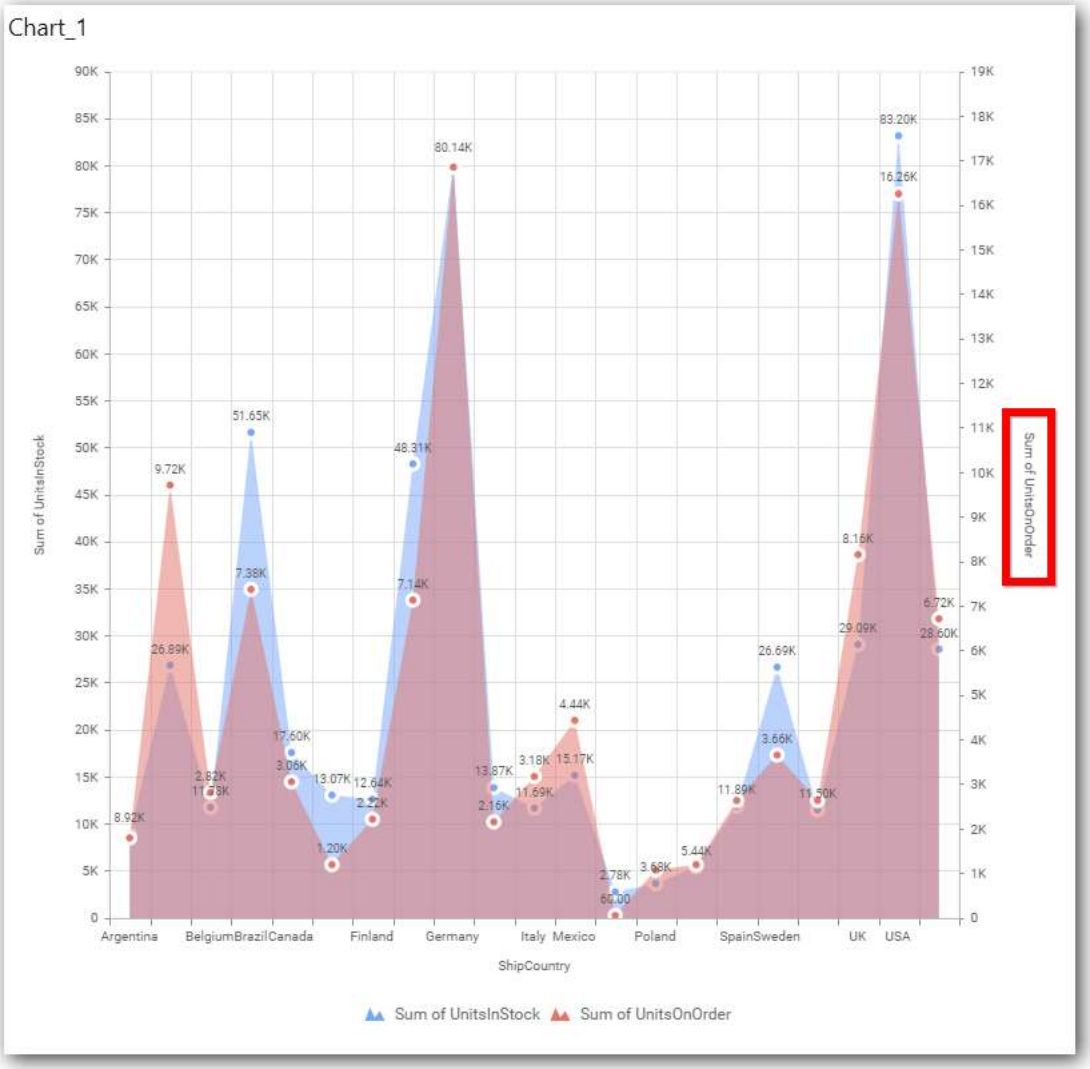
Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.



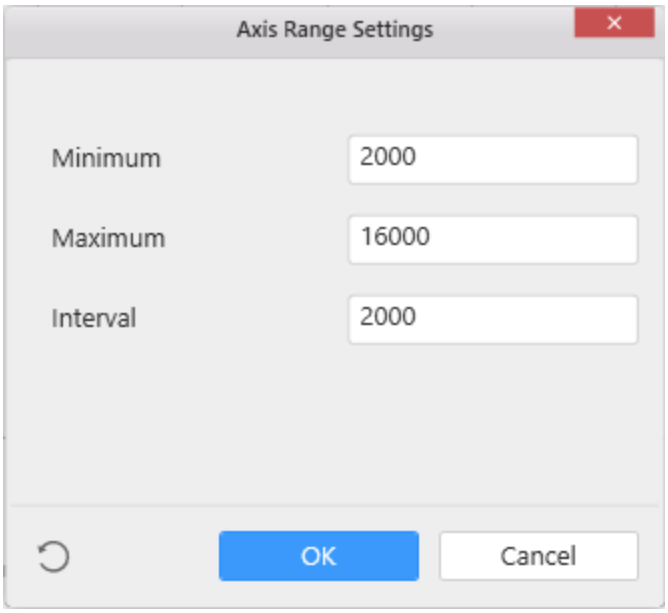
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

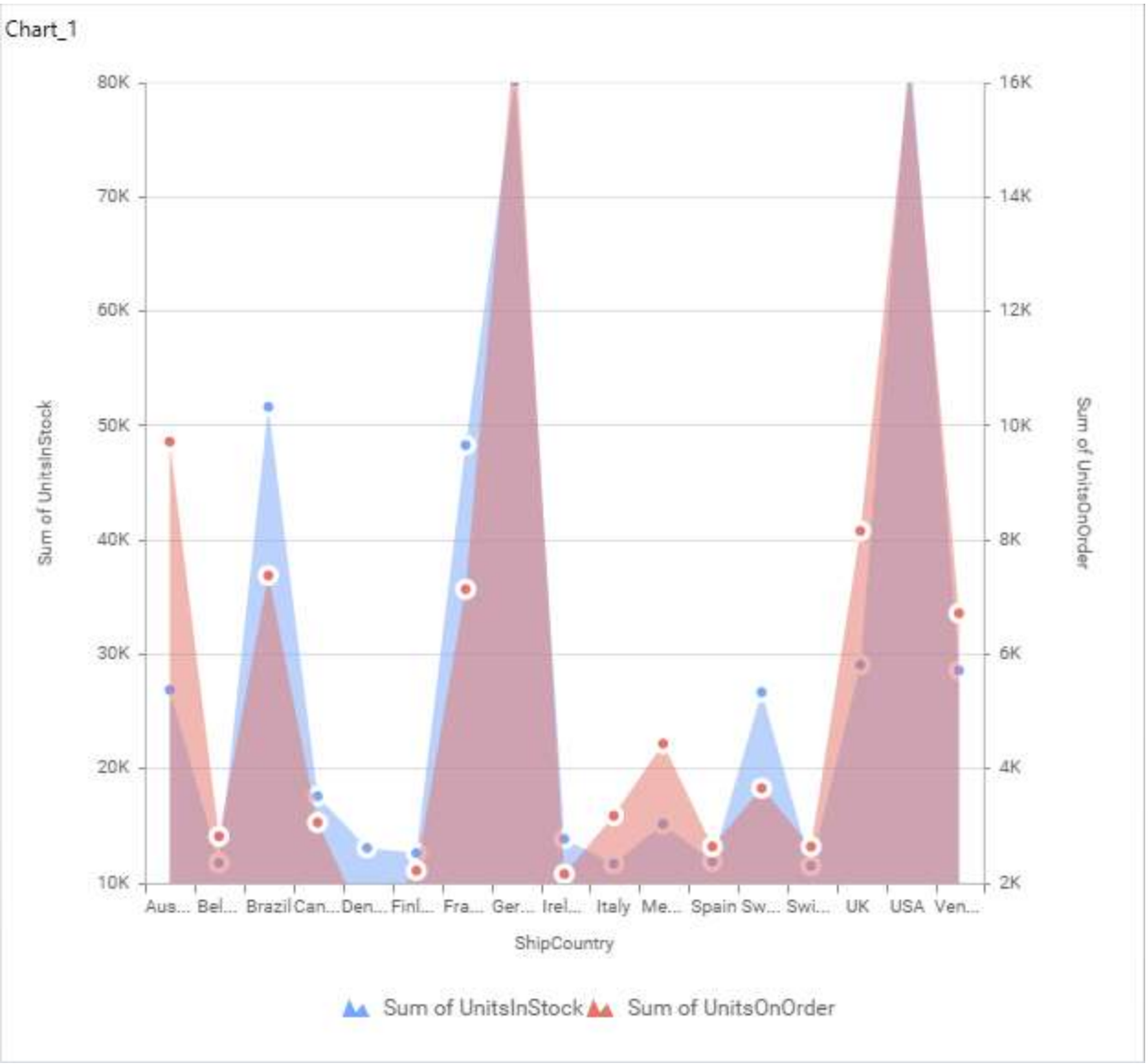


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

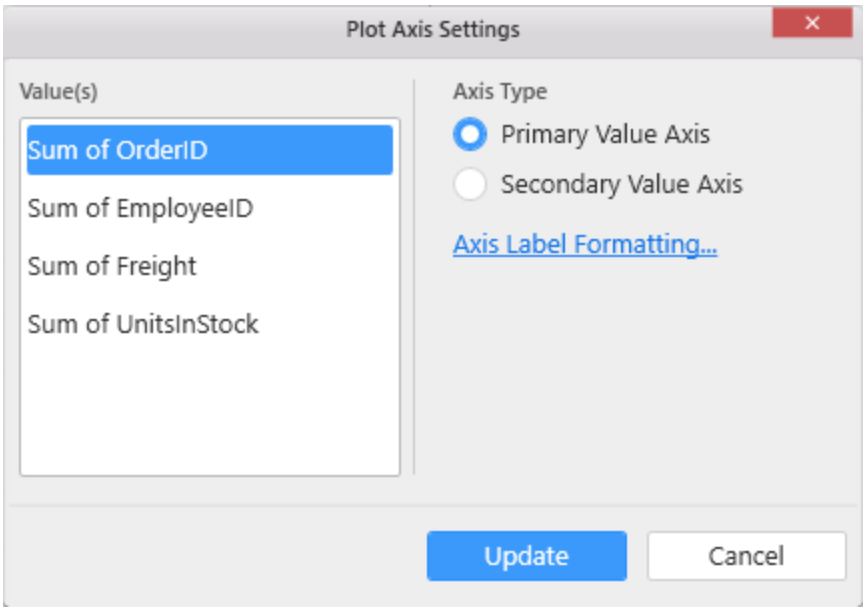


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

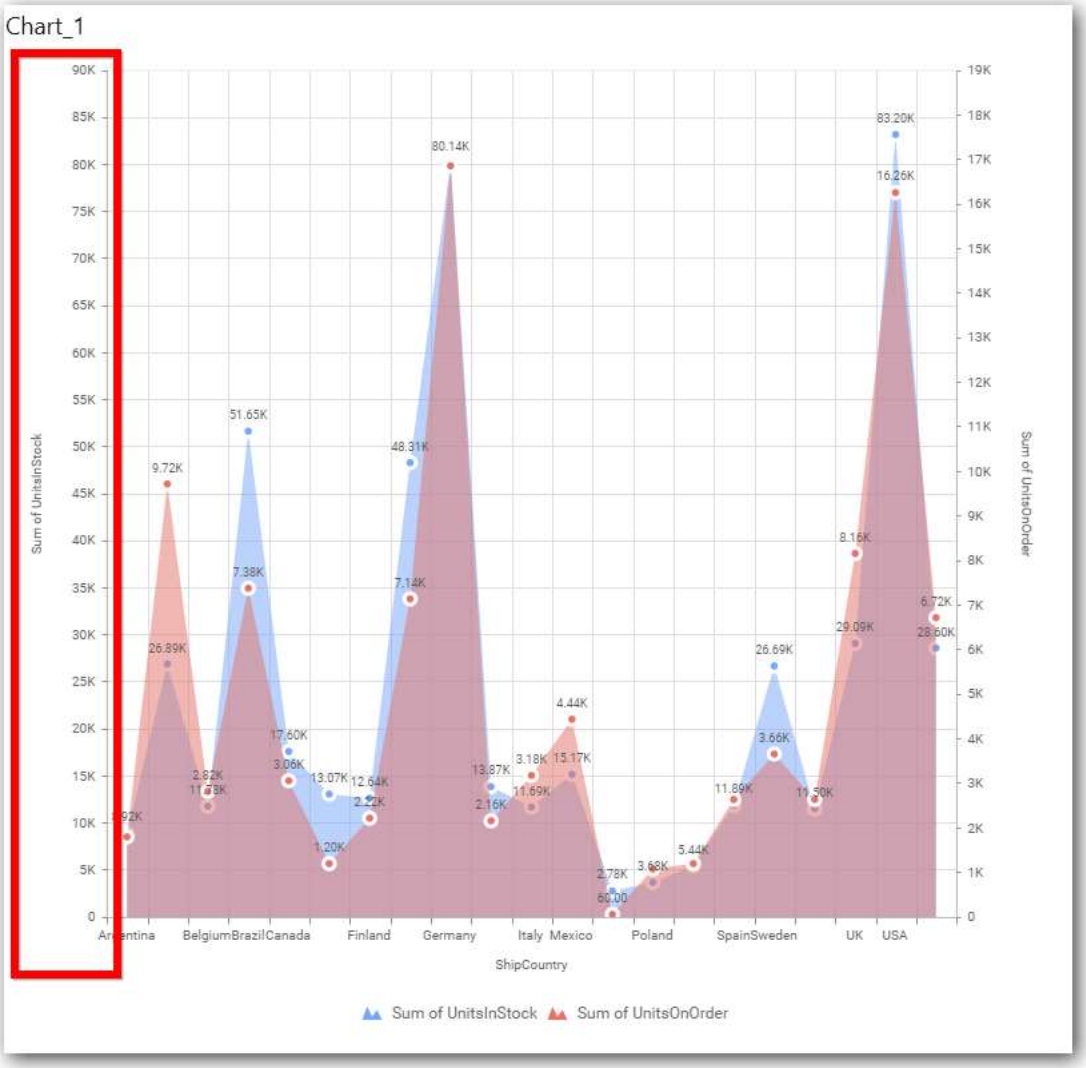


Plot Axis Settings

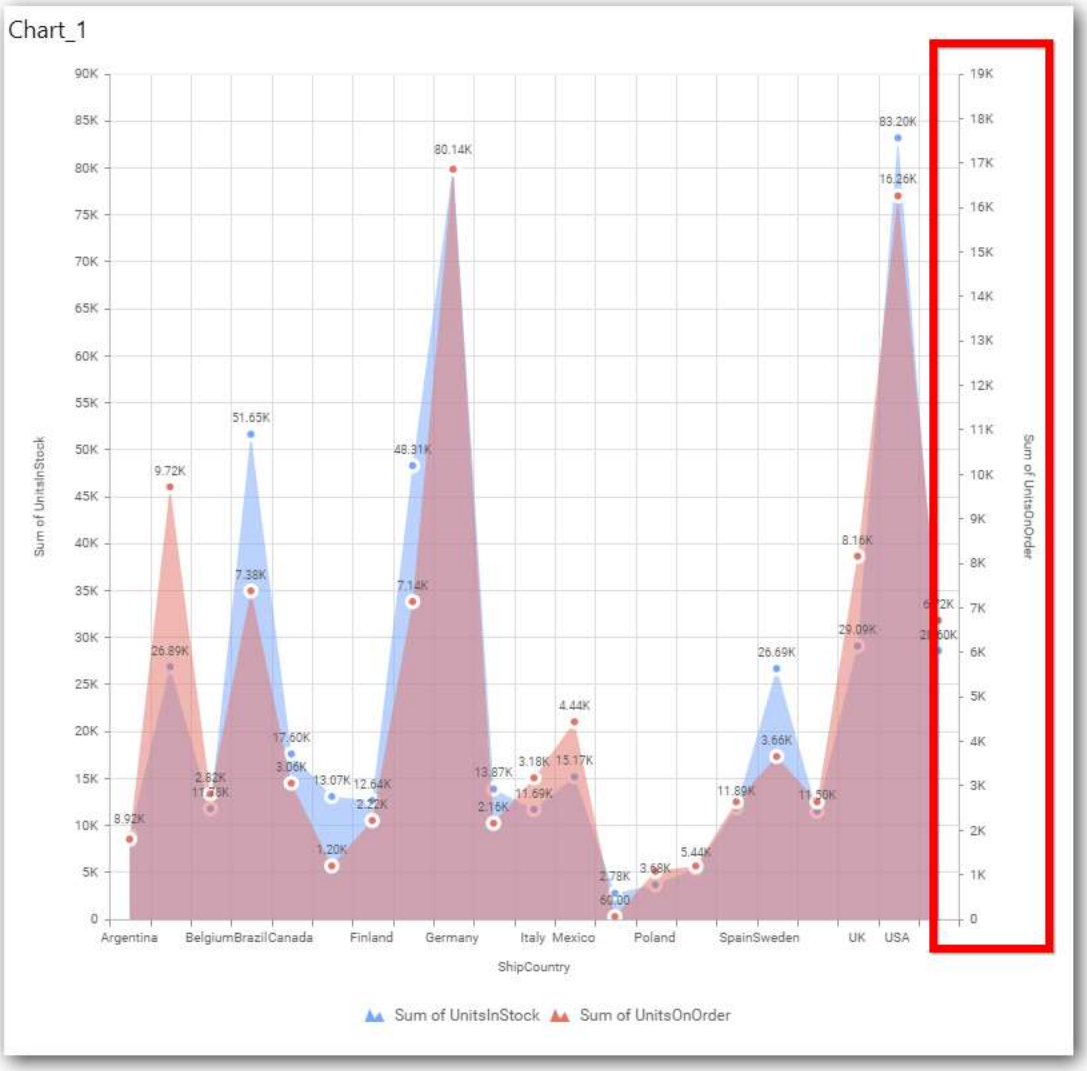
This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



Primary Value Axis

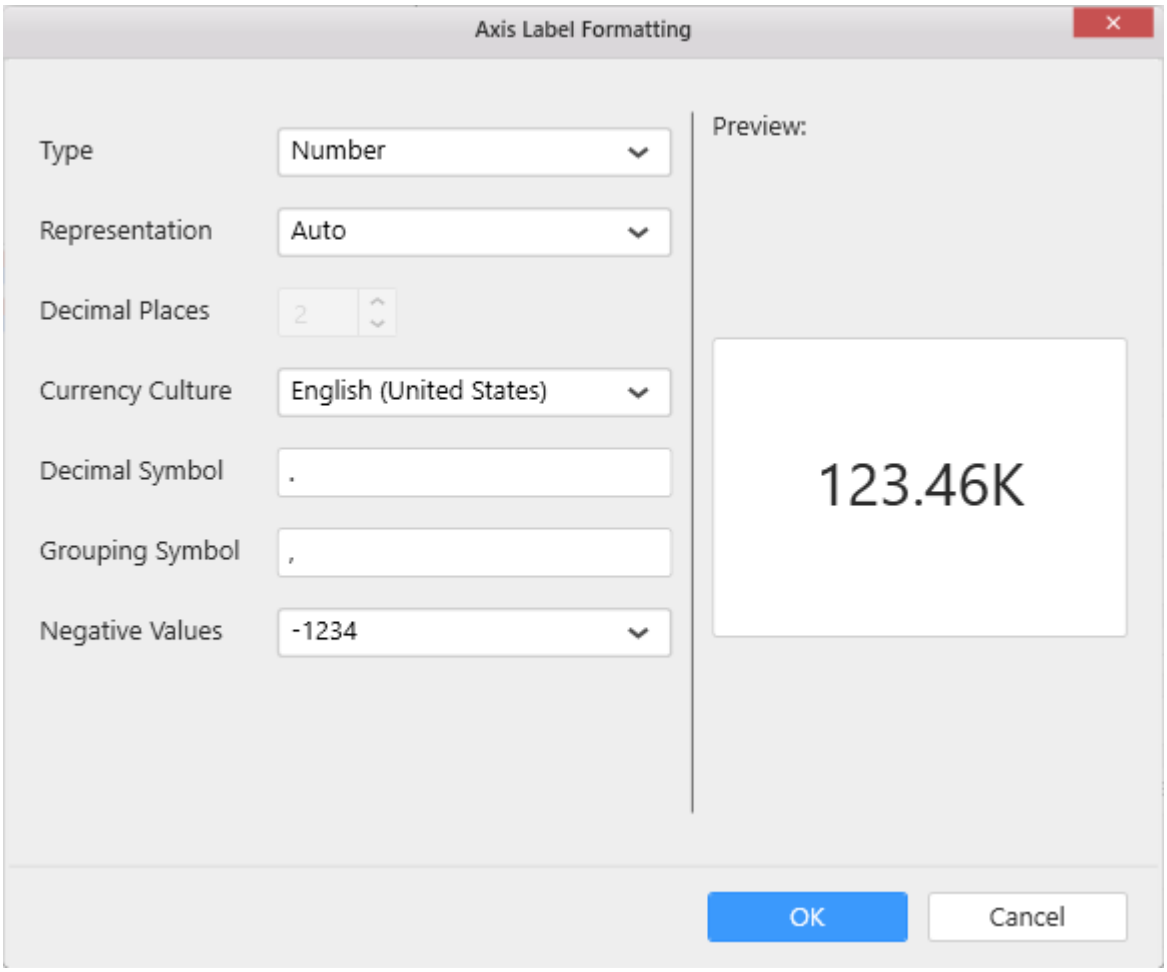


Secondary Value Axis



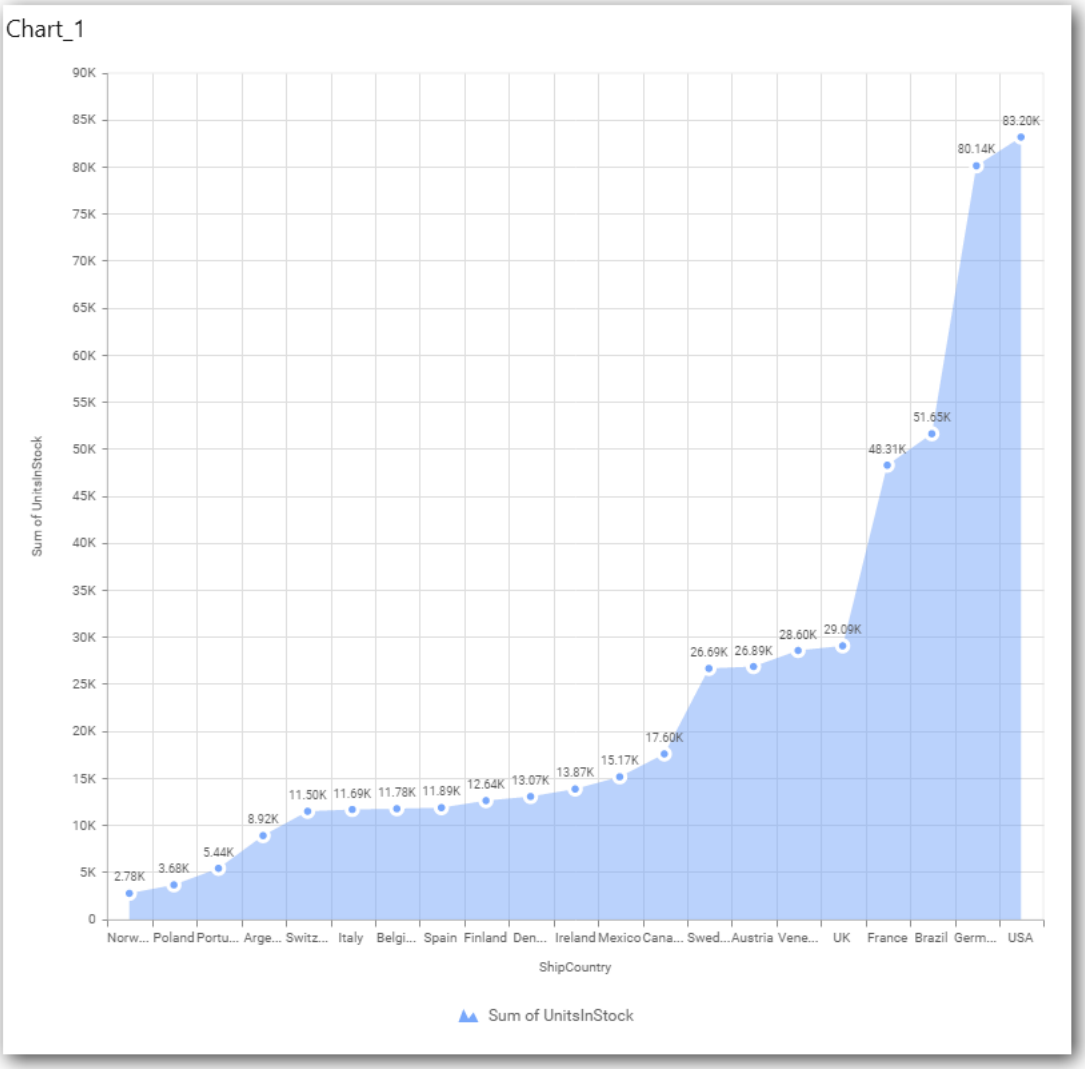
Axis Label Formatting

This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.

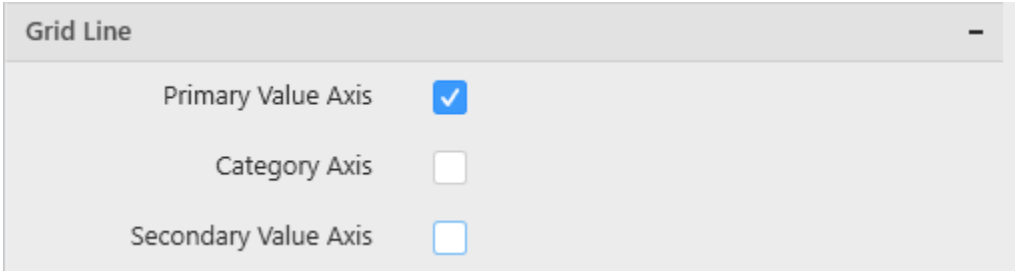


Sort Order

This allows you to define the sorting of chart based on any of the measures that you dropped and its order through this option. Following screenshot illustrates the Ascending sort order.

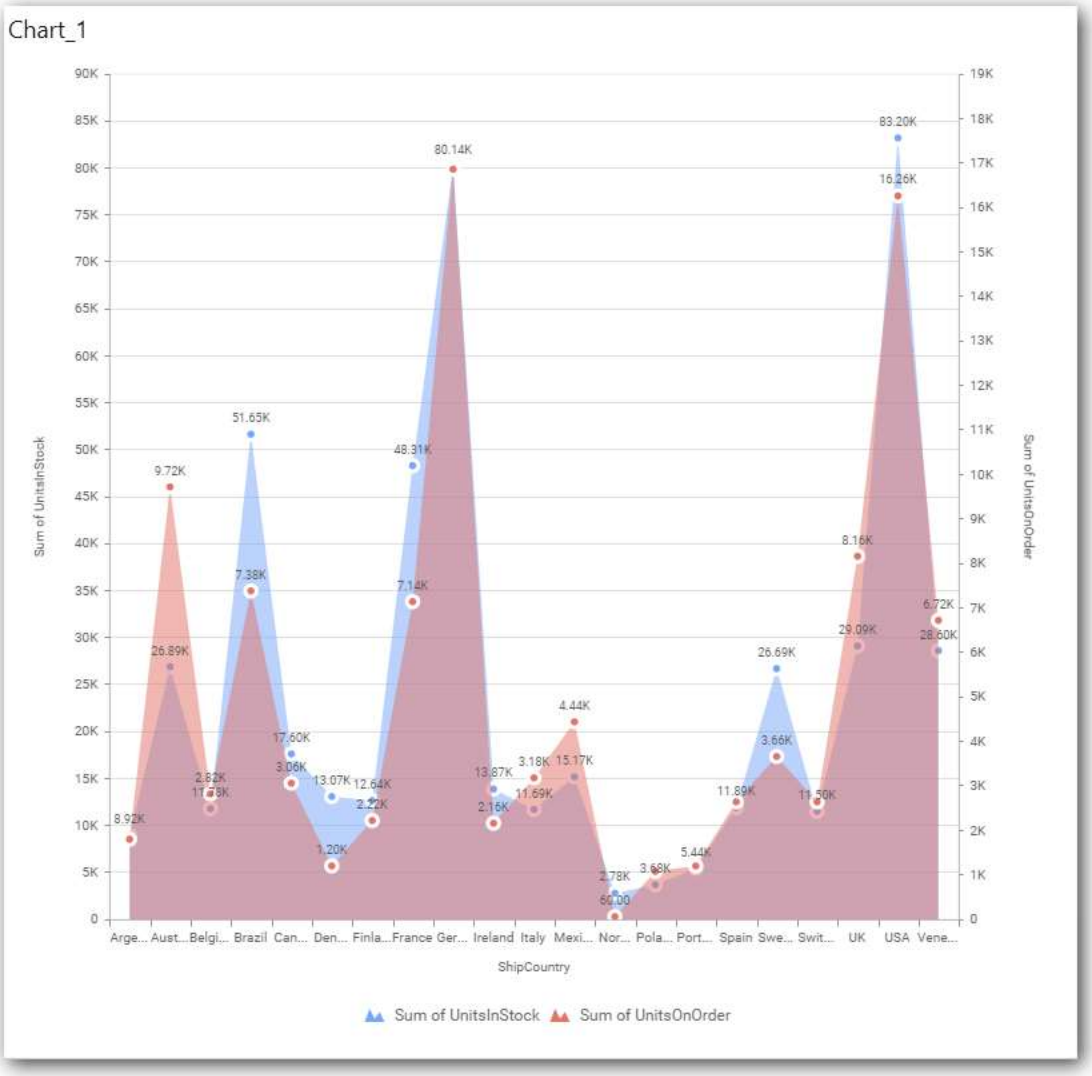


Grid Line Settings



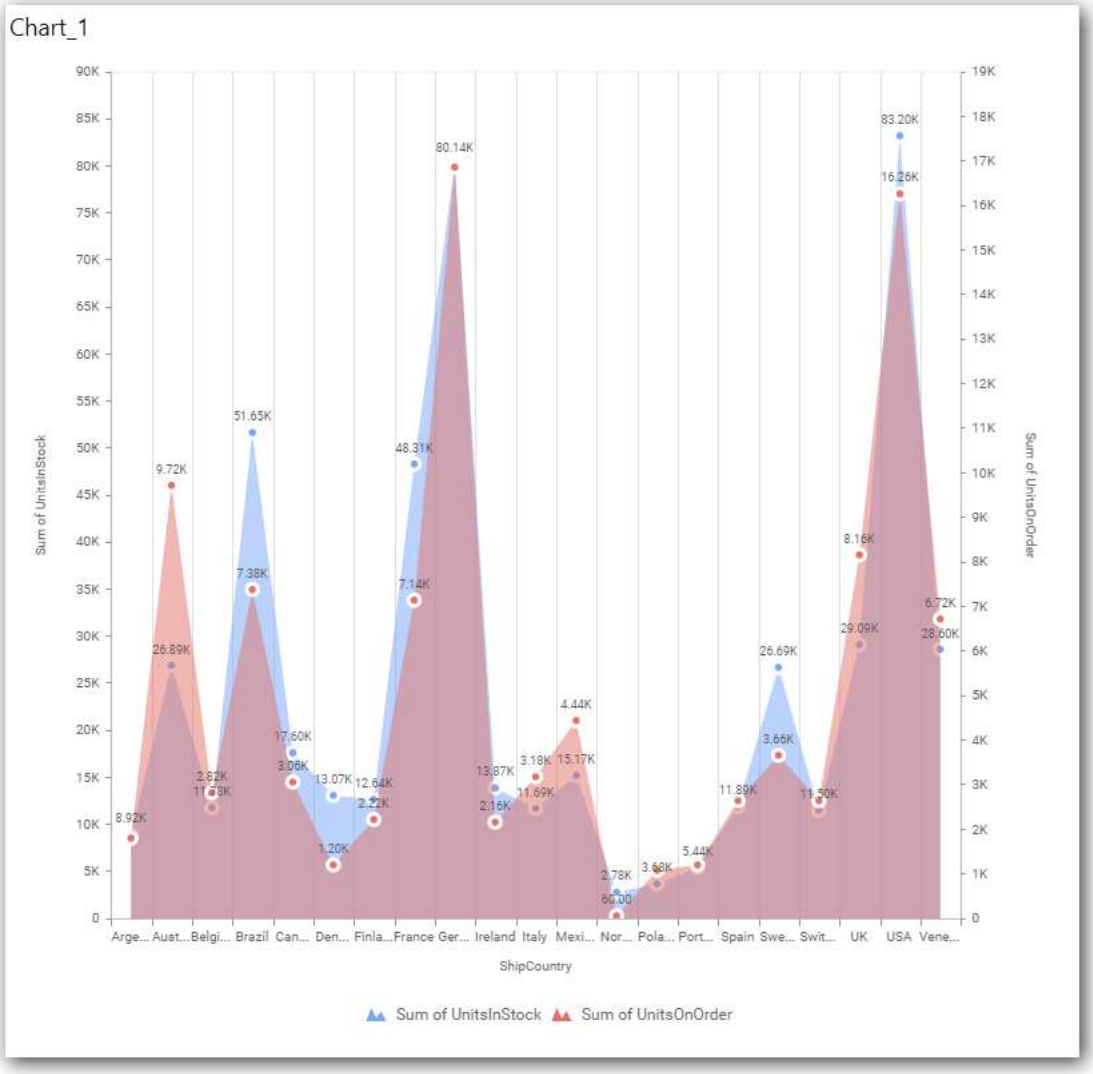
Primary Value Axis

This allows you to enable the primary value axis gridlines.



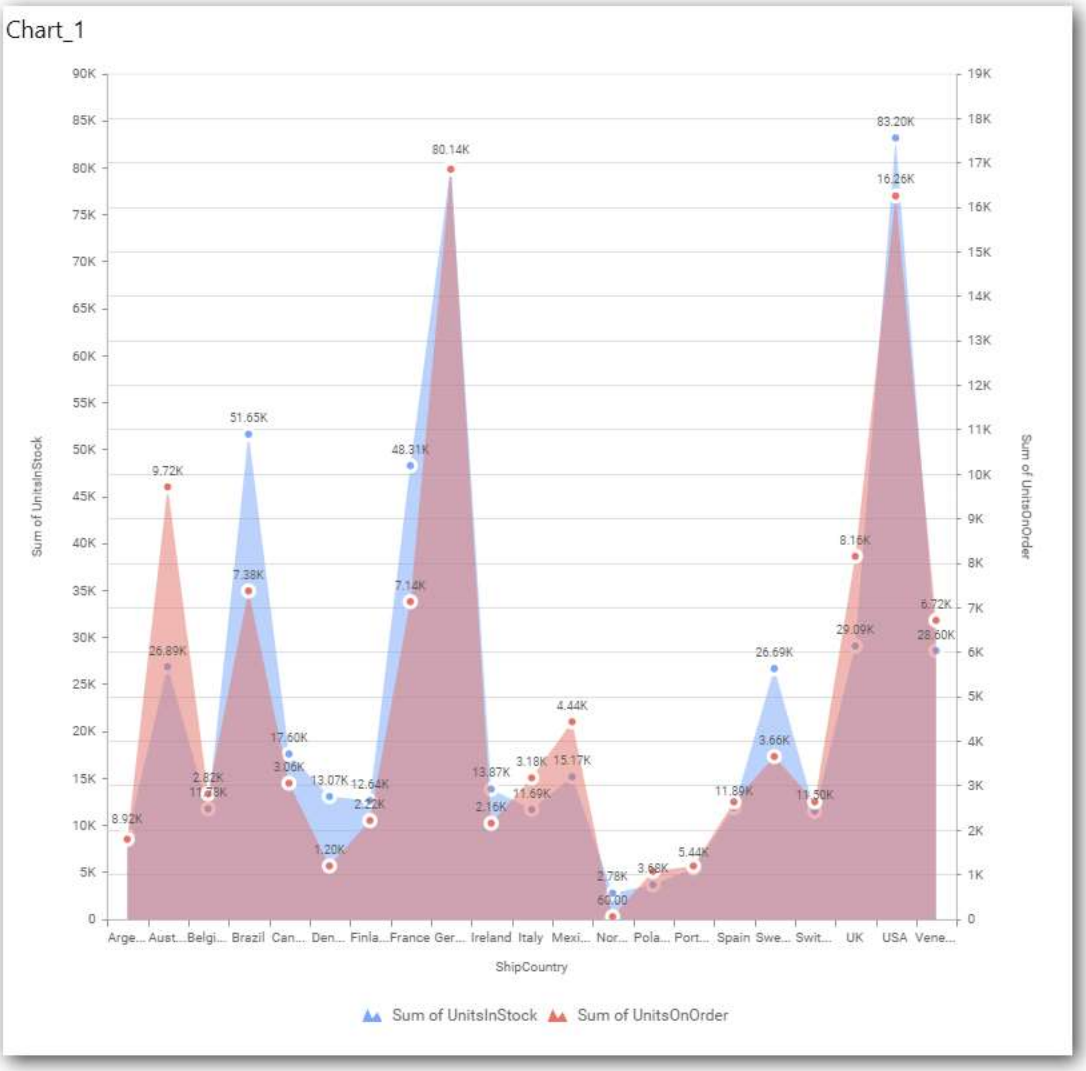
Category Axis

This allows you to toggle the visibility of category axis gridlines.

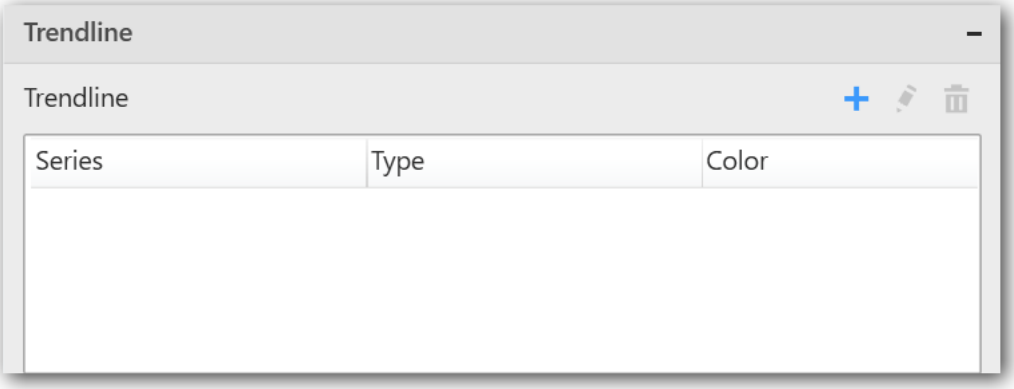


Secondary Value Axis

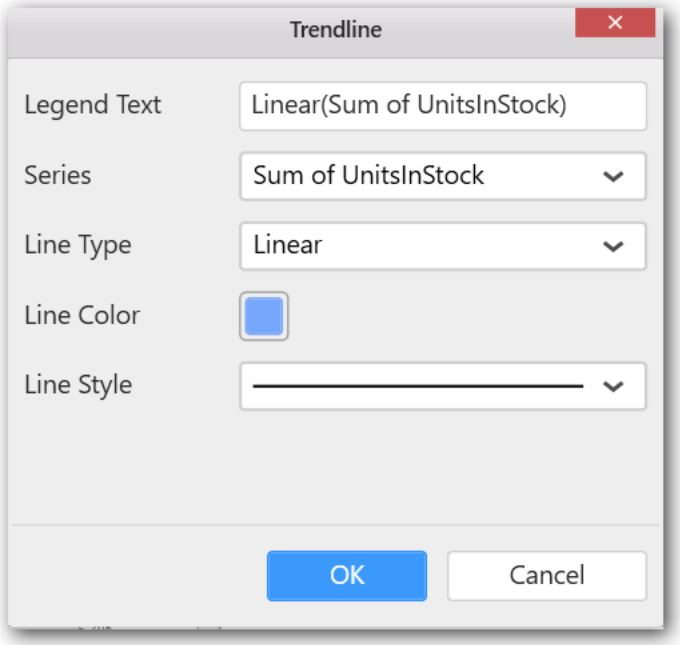
This allows you to toggle the visibility of secondary value axis gridlines.



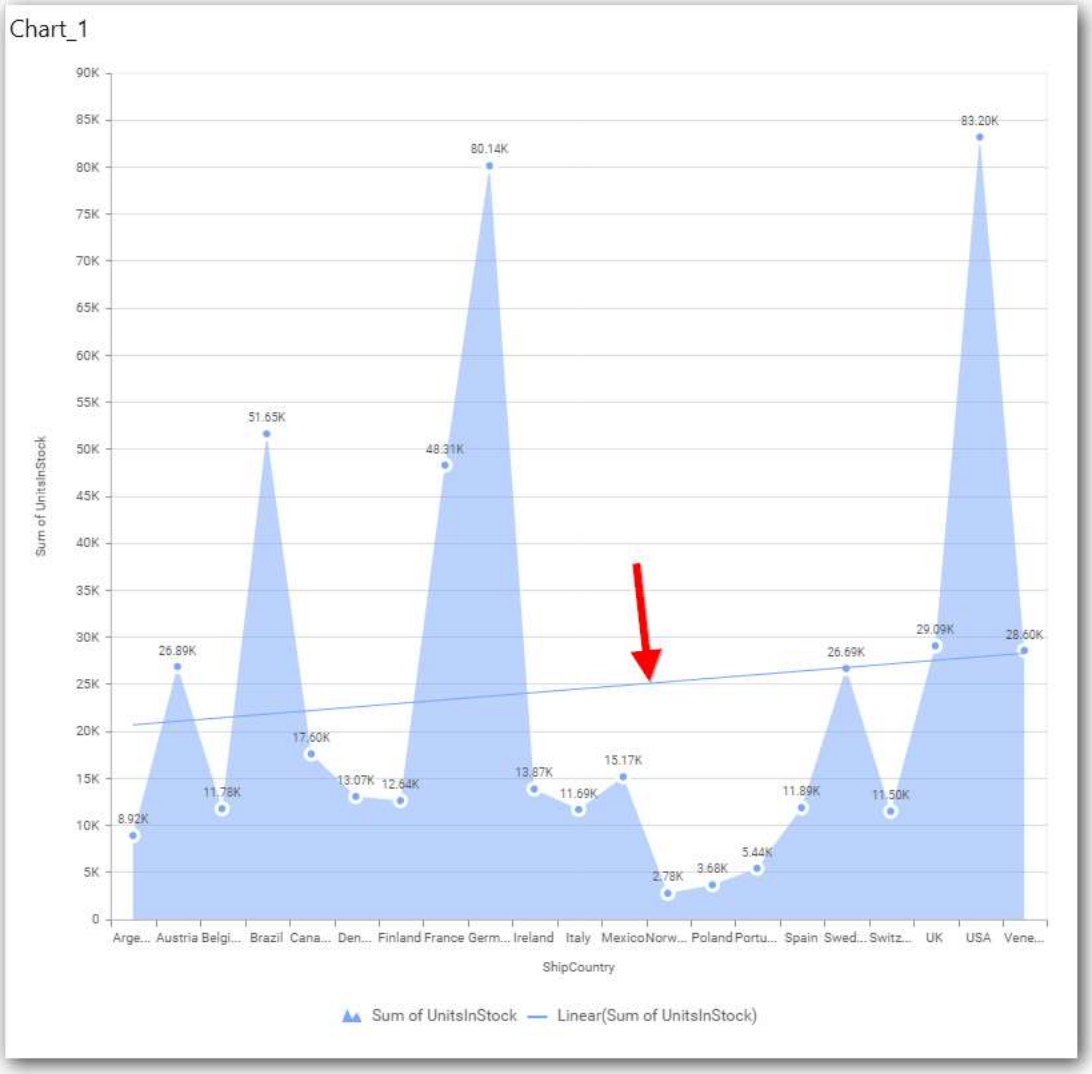
Trend Line Settings



You can add trend line to chart based on dropped measure that you select. You can also customize its legend text, line type and line color. Trend line is not visible, by default.



After applying these settings, it will reflect in chart like below.



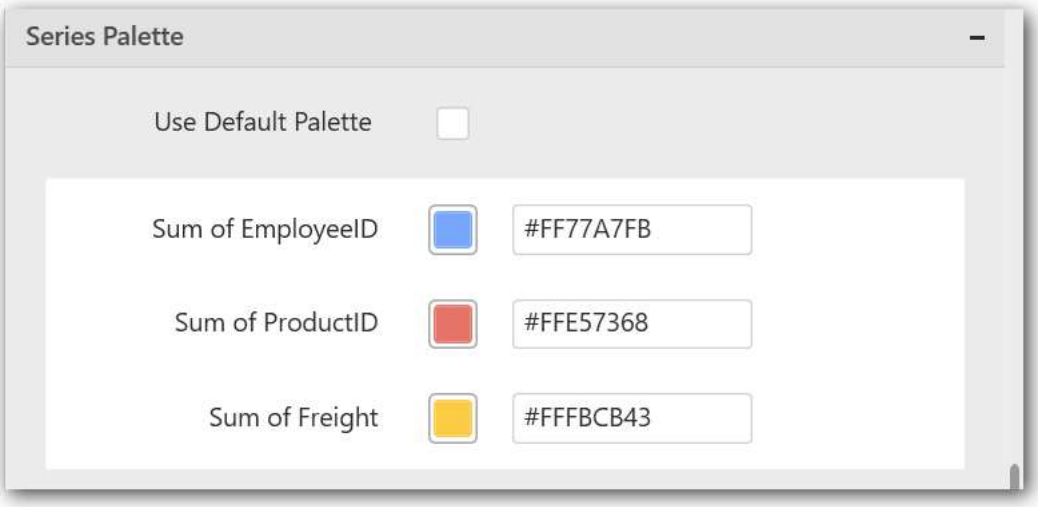
You have options to edit or delete the added trend lines.

Series Palette

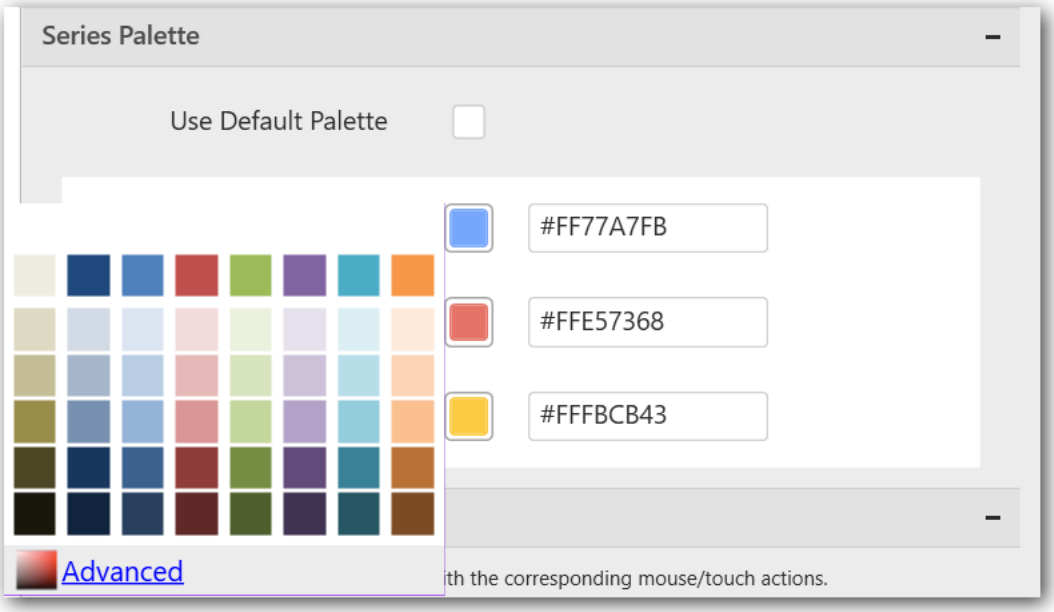
This allows you to customize the chart series color through Series Palette section.

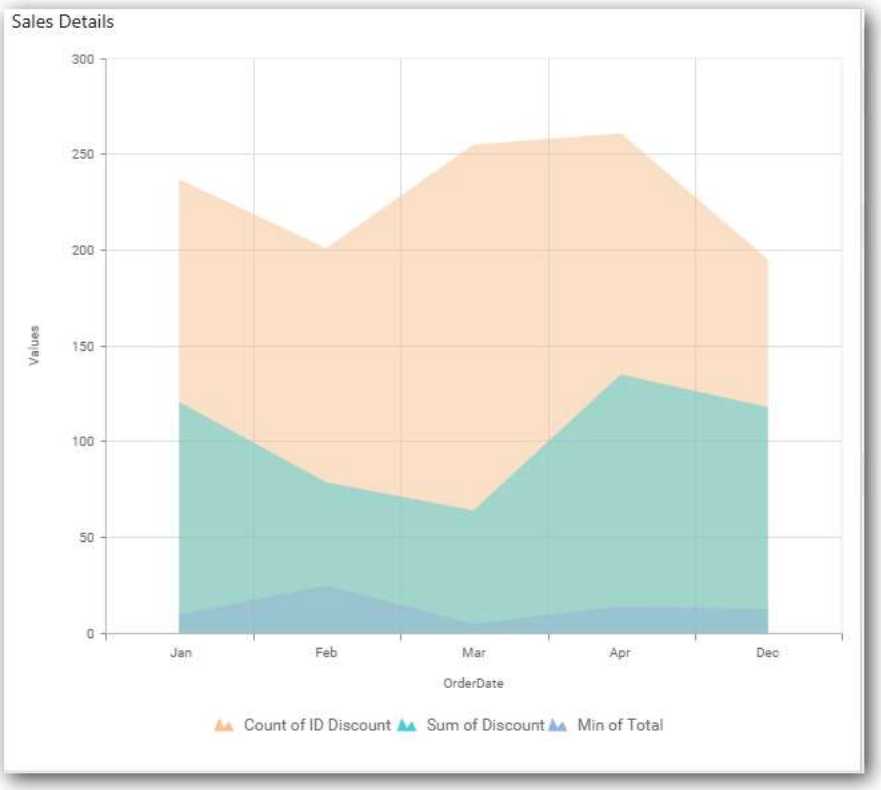
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



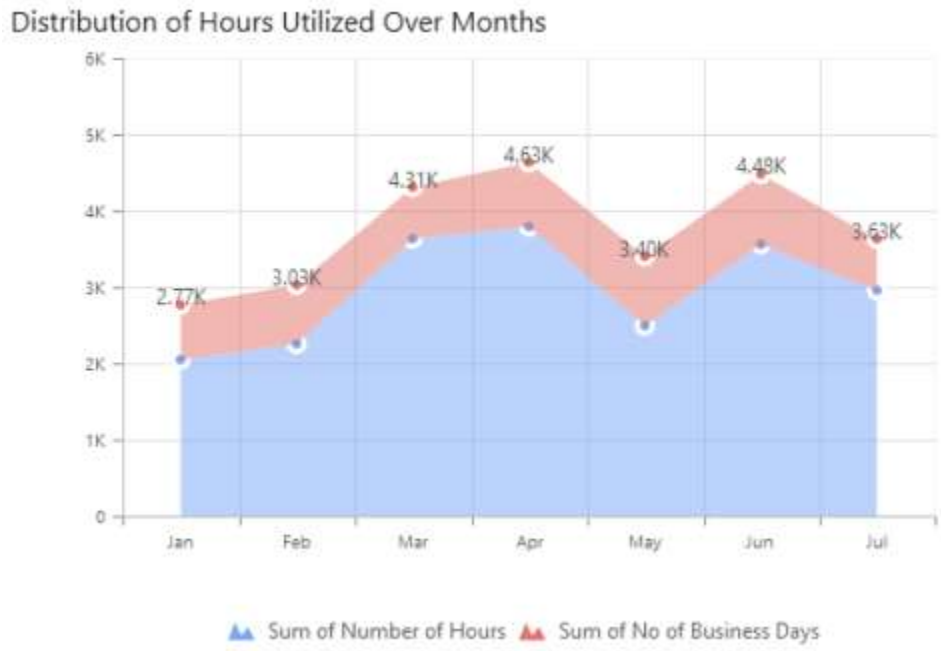
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Stacked Area Chart

Stacked Area Chart allows you to compare multiple measures through filled curves stacked one after the other vertically.

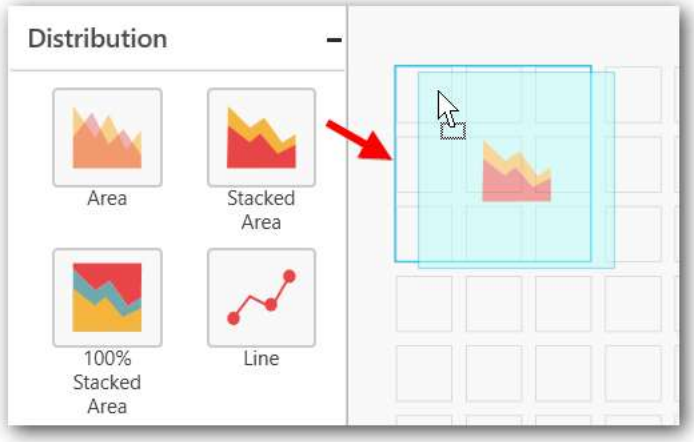


How to configure the flat table data to Stacked Area Chart?

Stacked Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

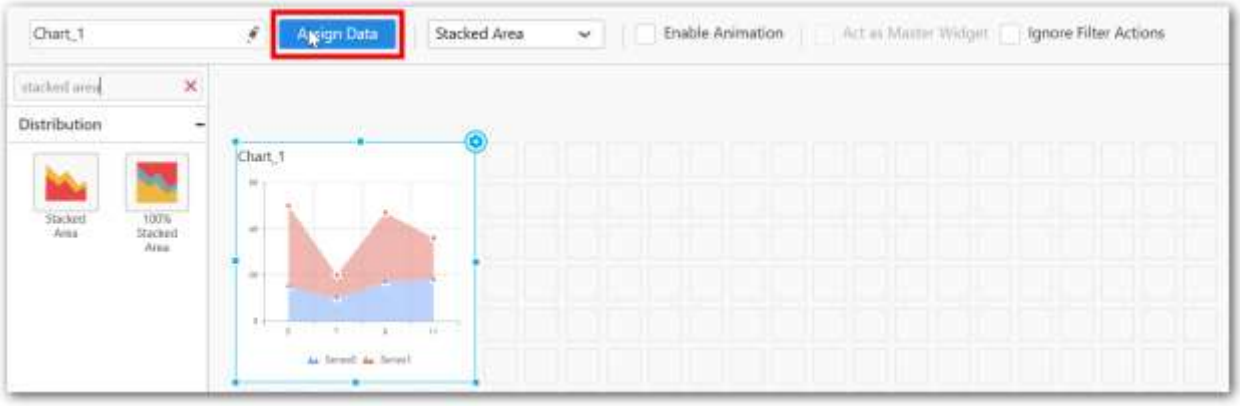
Follow the steps to configure data to stacked area chart

Drag and drop the stacked area chart into canvas and resize it to your required size.

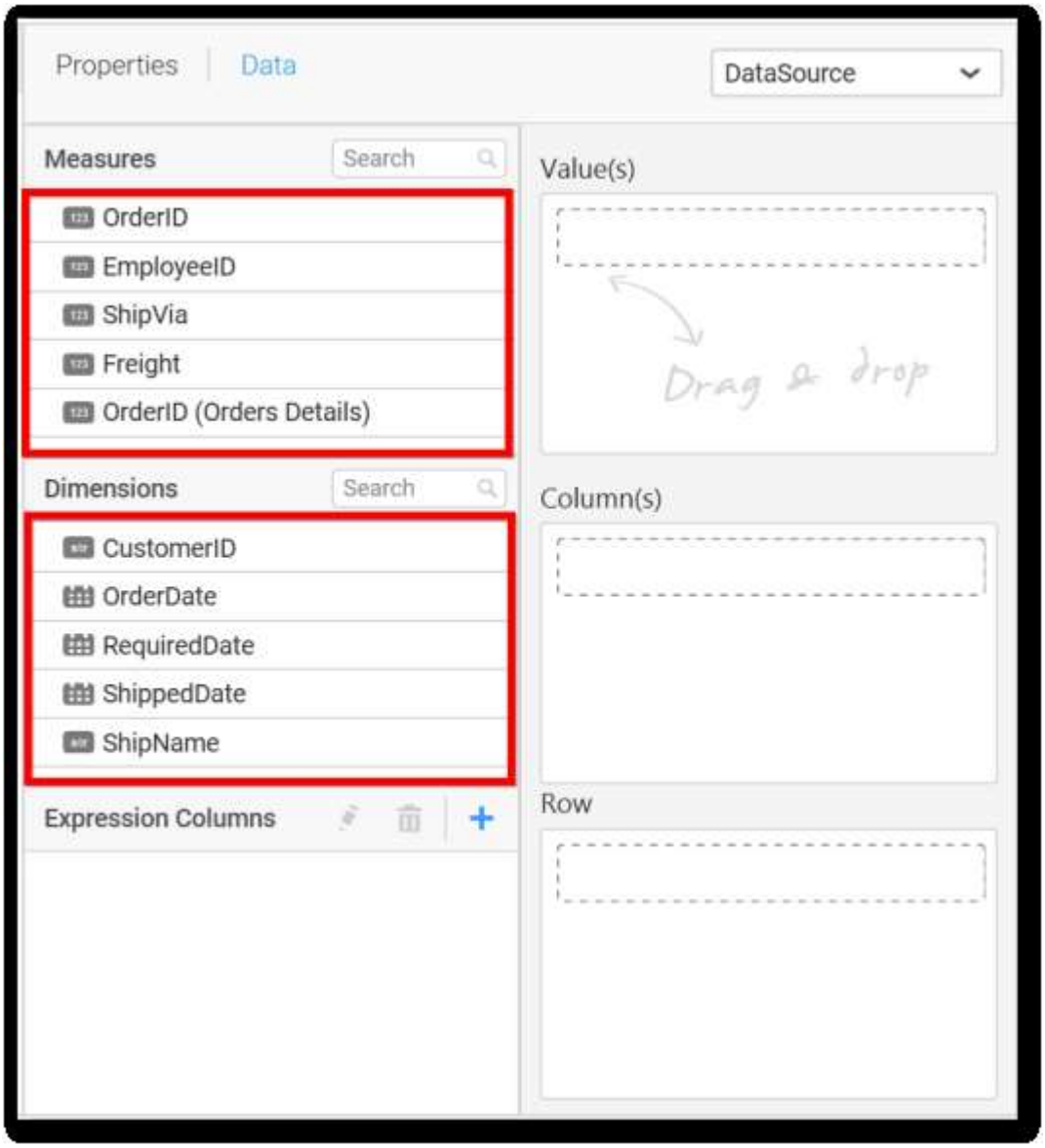


Connect to the data source.

Focus on the stacked area chart and click on Assign Data.



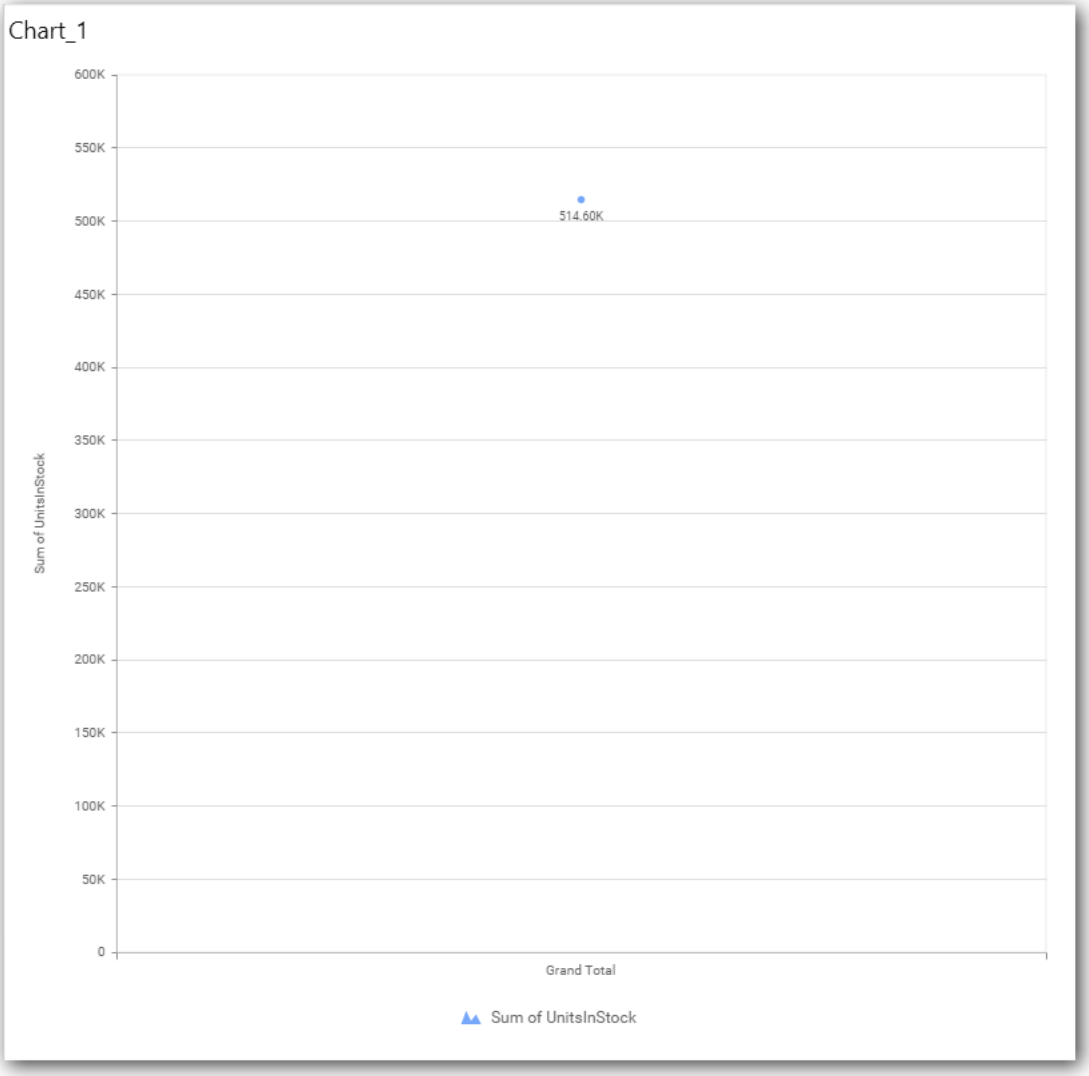
The data pane will be opened with available measures and dimensions from the connected data source.



Assigning Value(s)

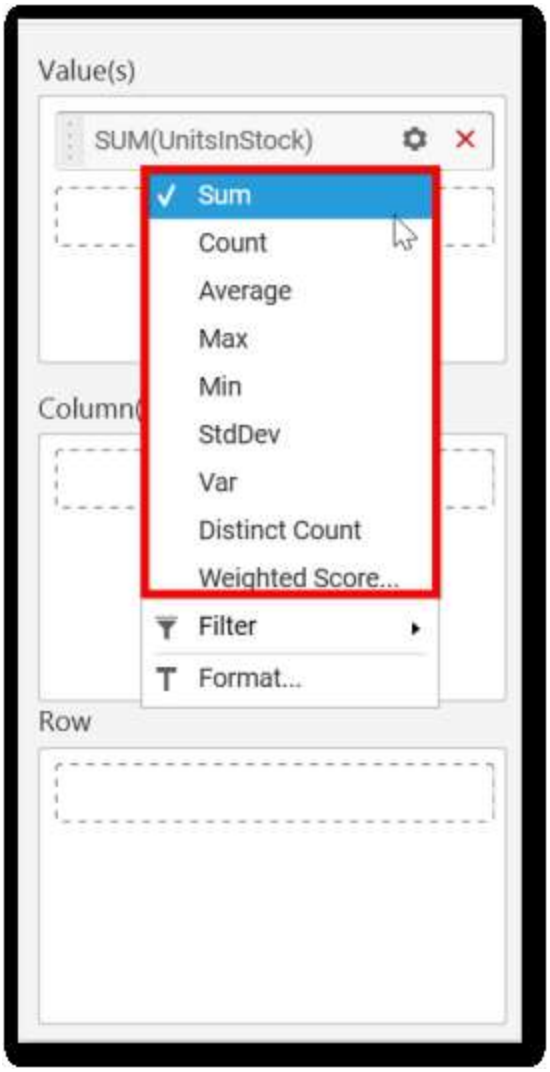
Drag and drop the Measure into Value.

Now the chart will be rendered like this

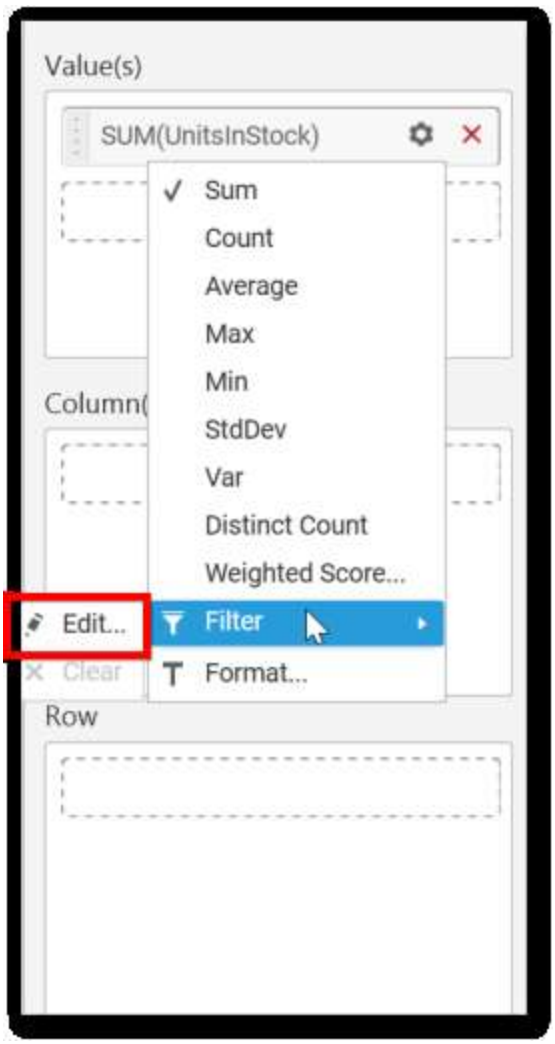


You can change the summary type of the value by clicking on Settings option.

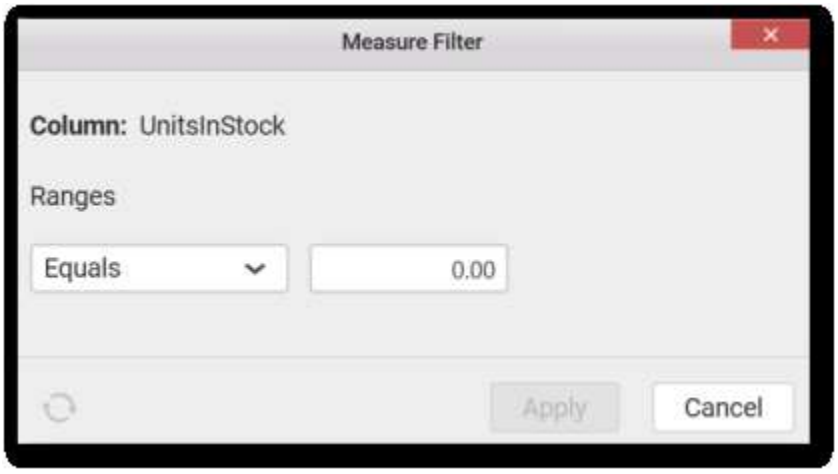
Select the required summary type from list.



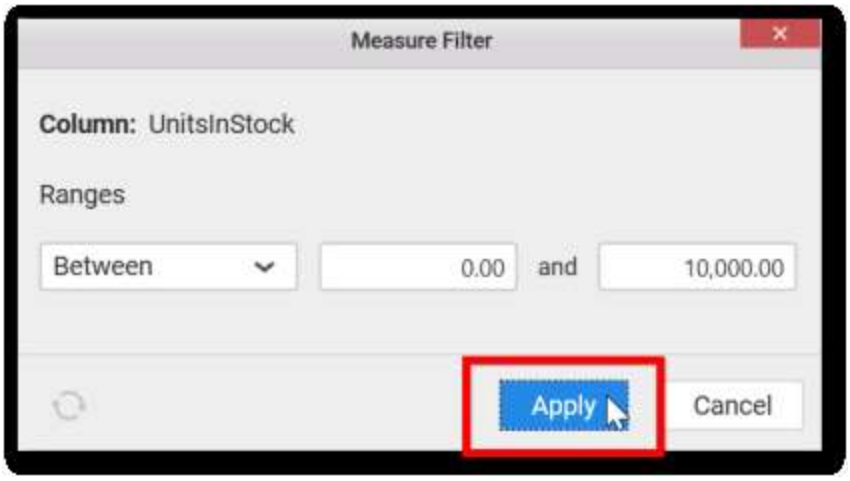
You can select what data to be displayed by choosing filter option.



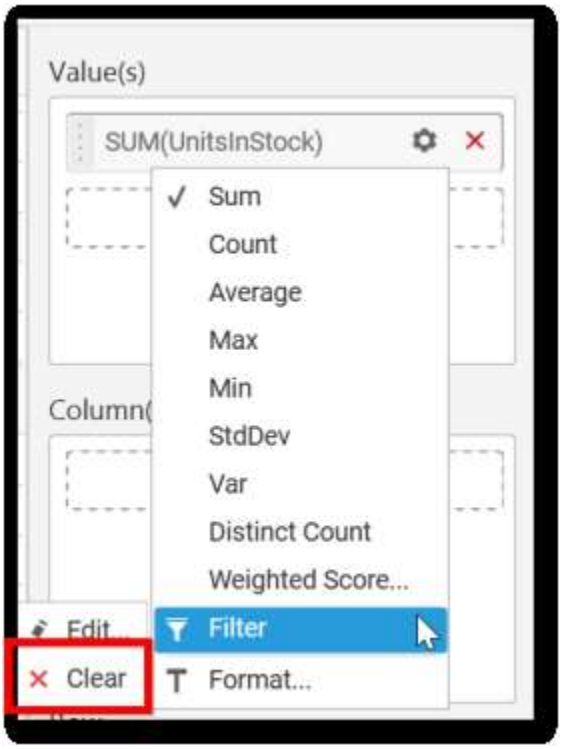
The Measure Filter option will be shown.



You can choose the filter condition and apply the condition value.

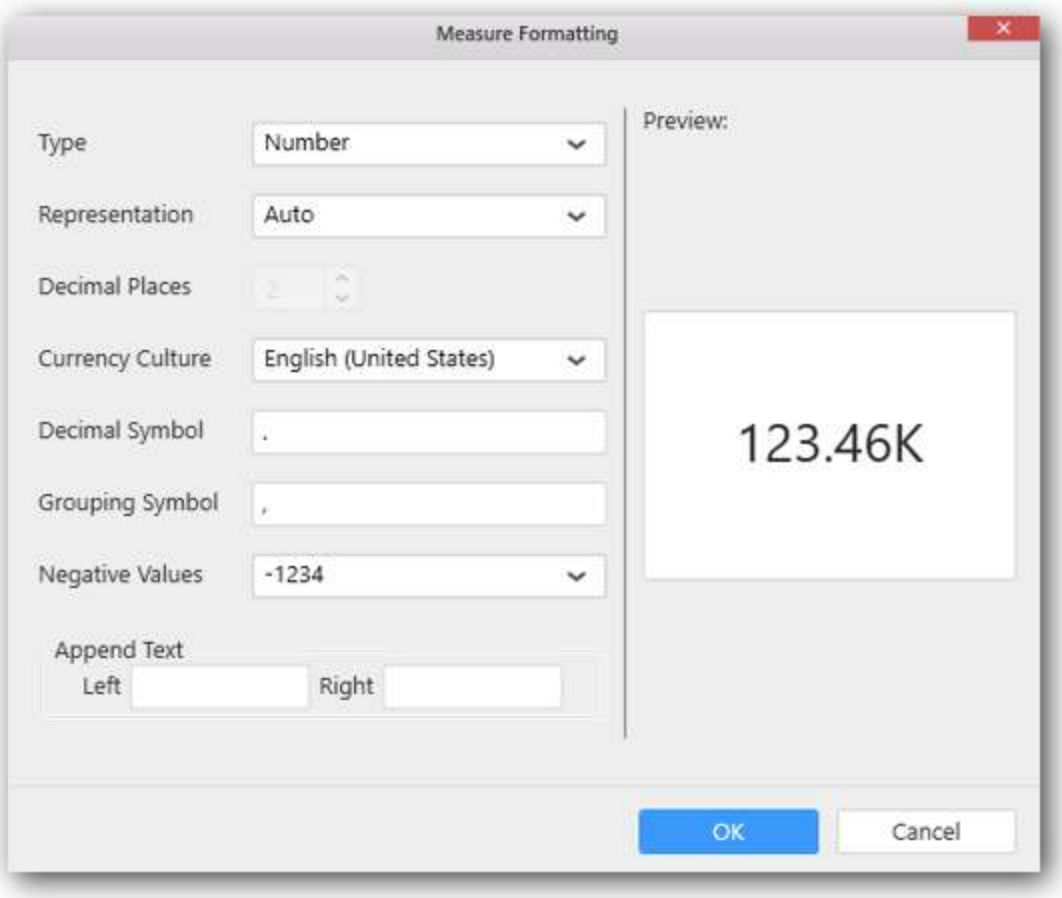


You can clear the filter.

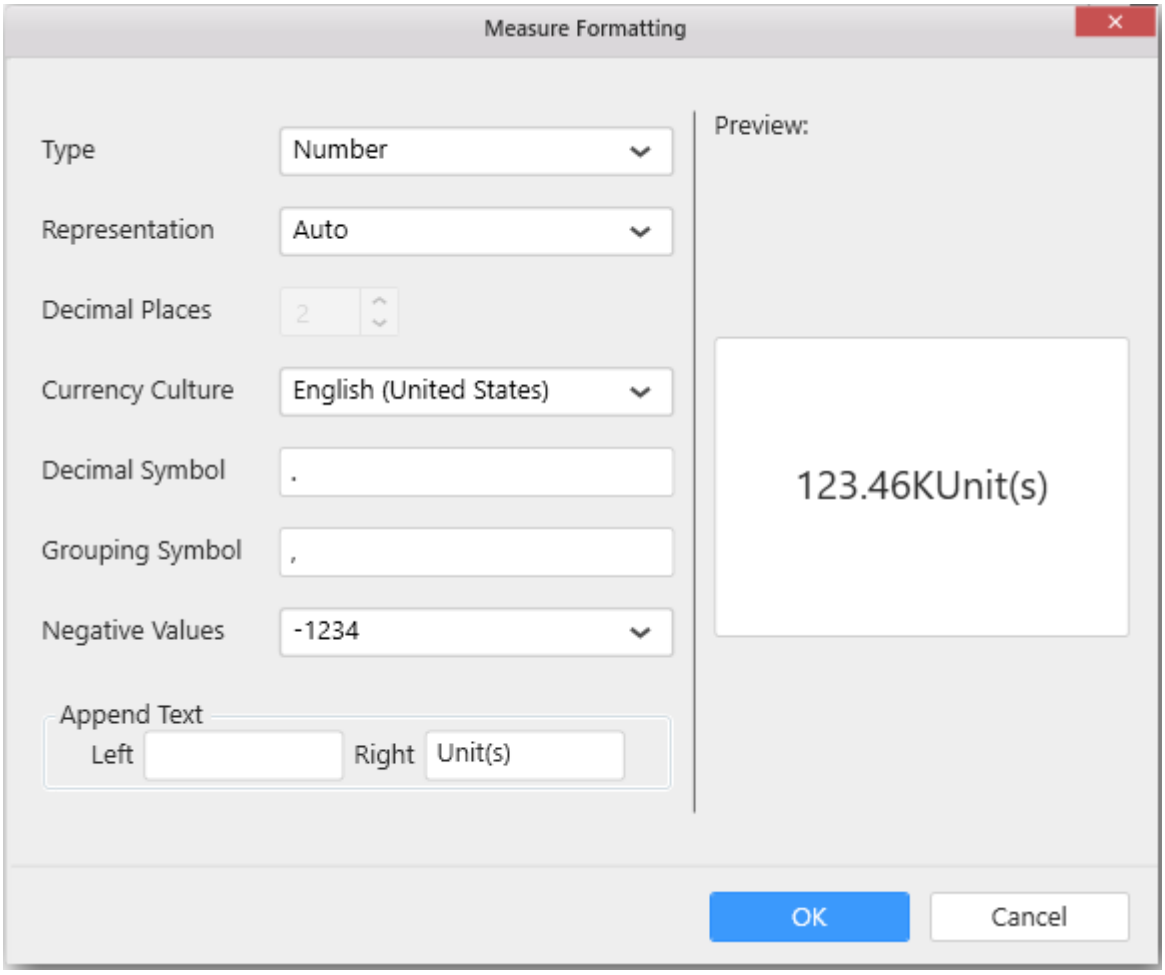


You can Format the value.

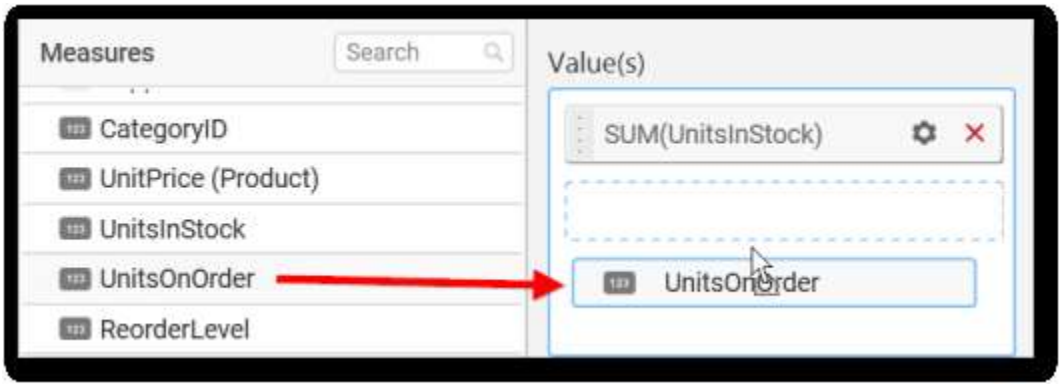
The format options will be shown.



Choose the options you need and click OK.



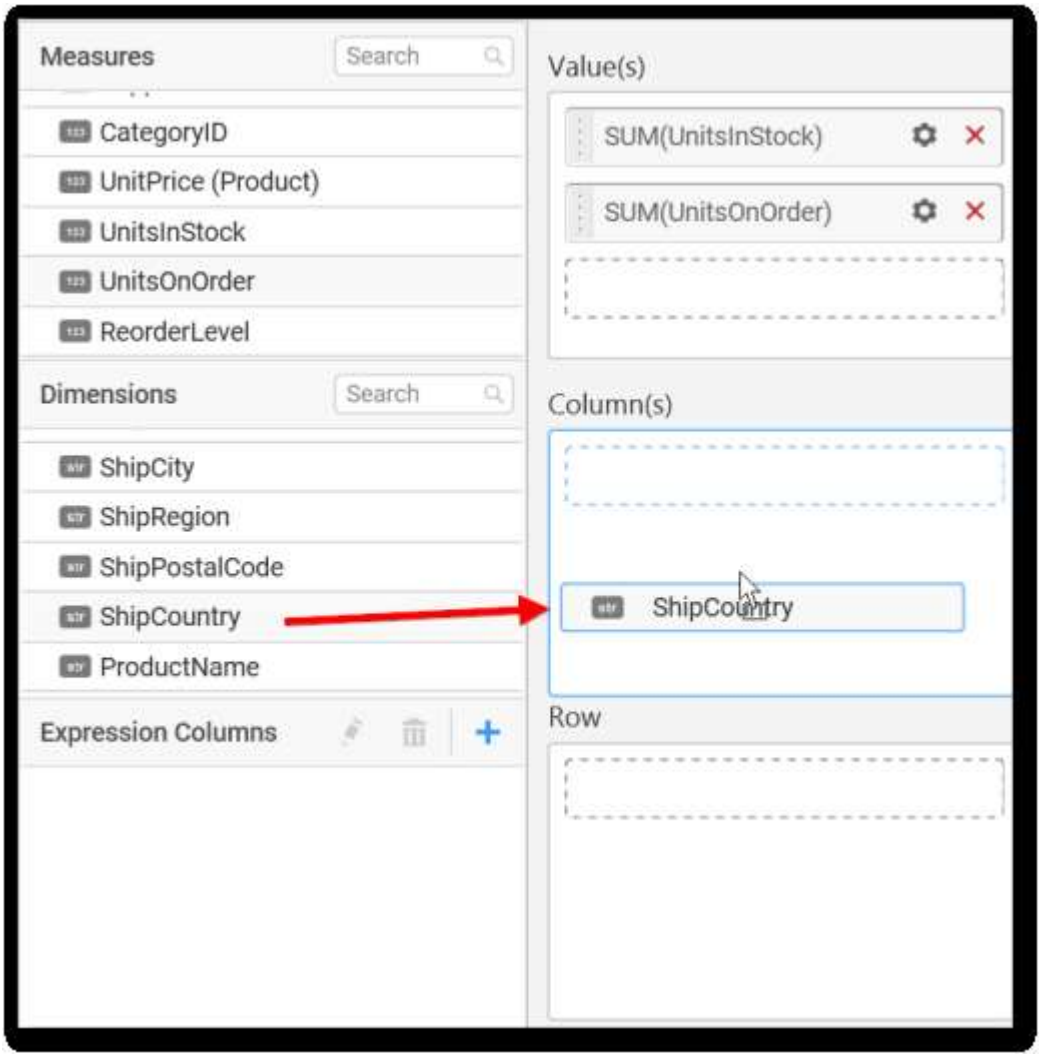
You can add more number values by drag drop the Measures into Value field.

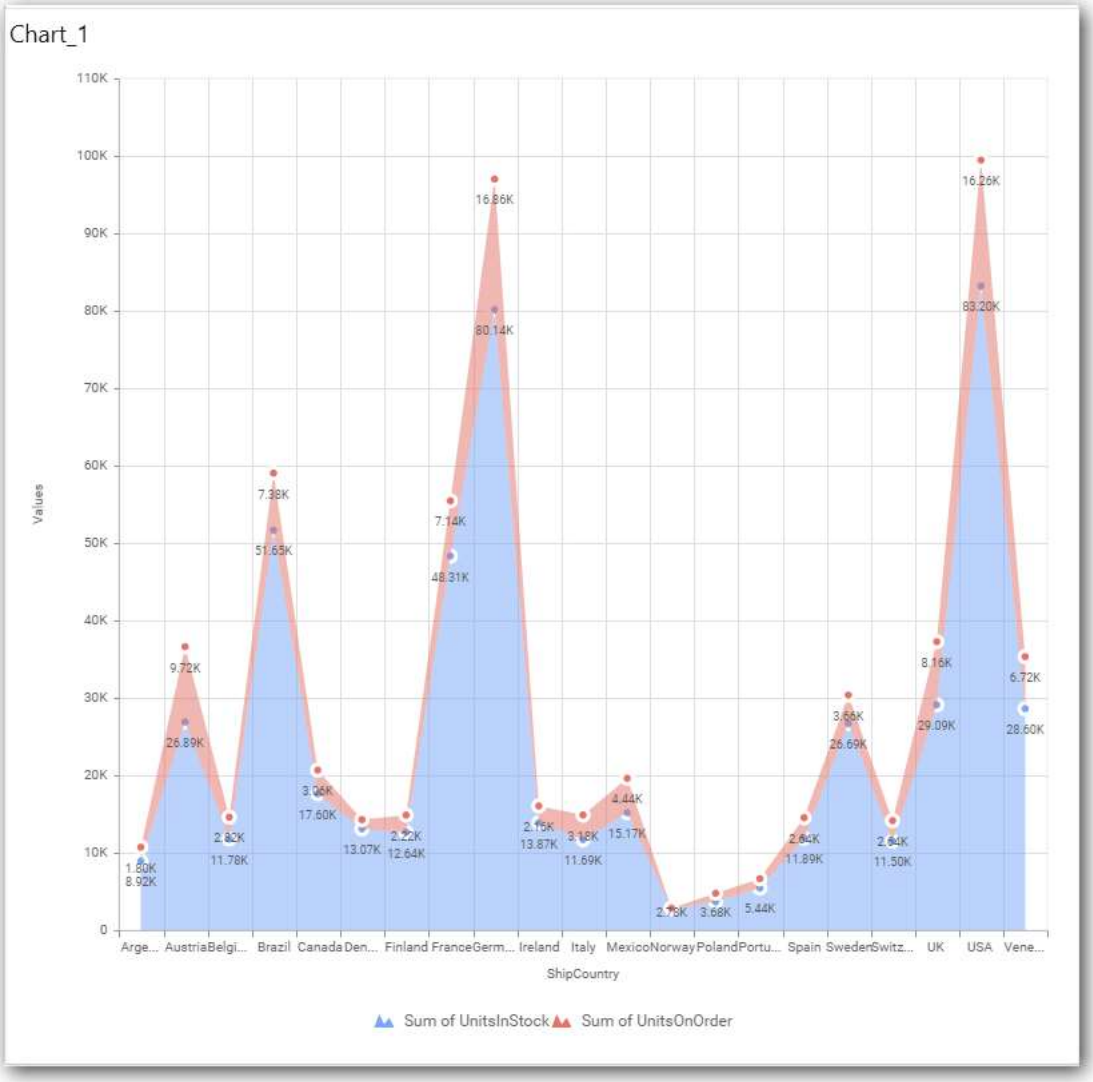


You can also add Dimensions and Columns to Value(s).

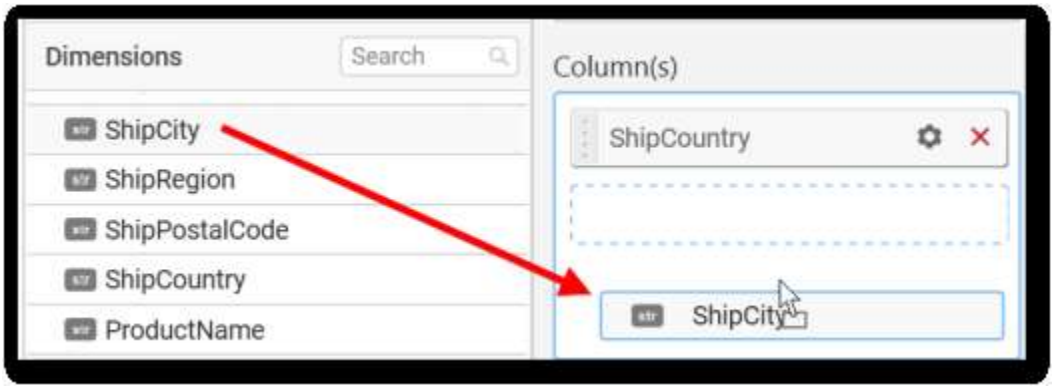
Assigning Column(s)

You can add the Dimension into Column field by drag and drop.

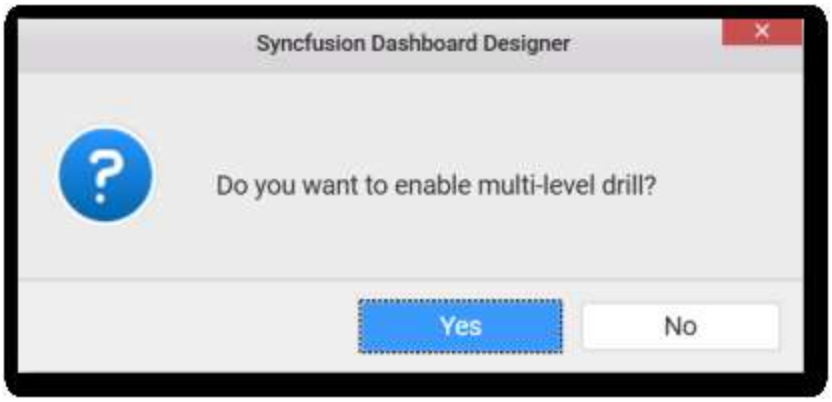




You have option to add more than one Column Value.

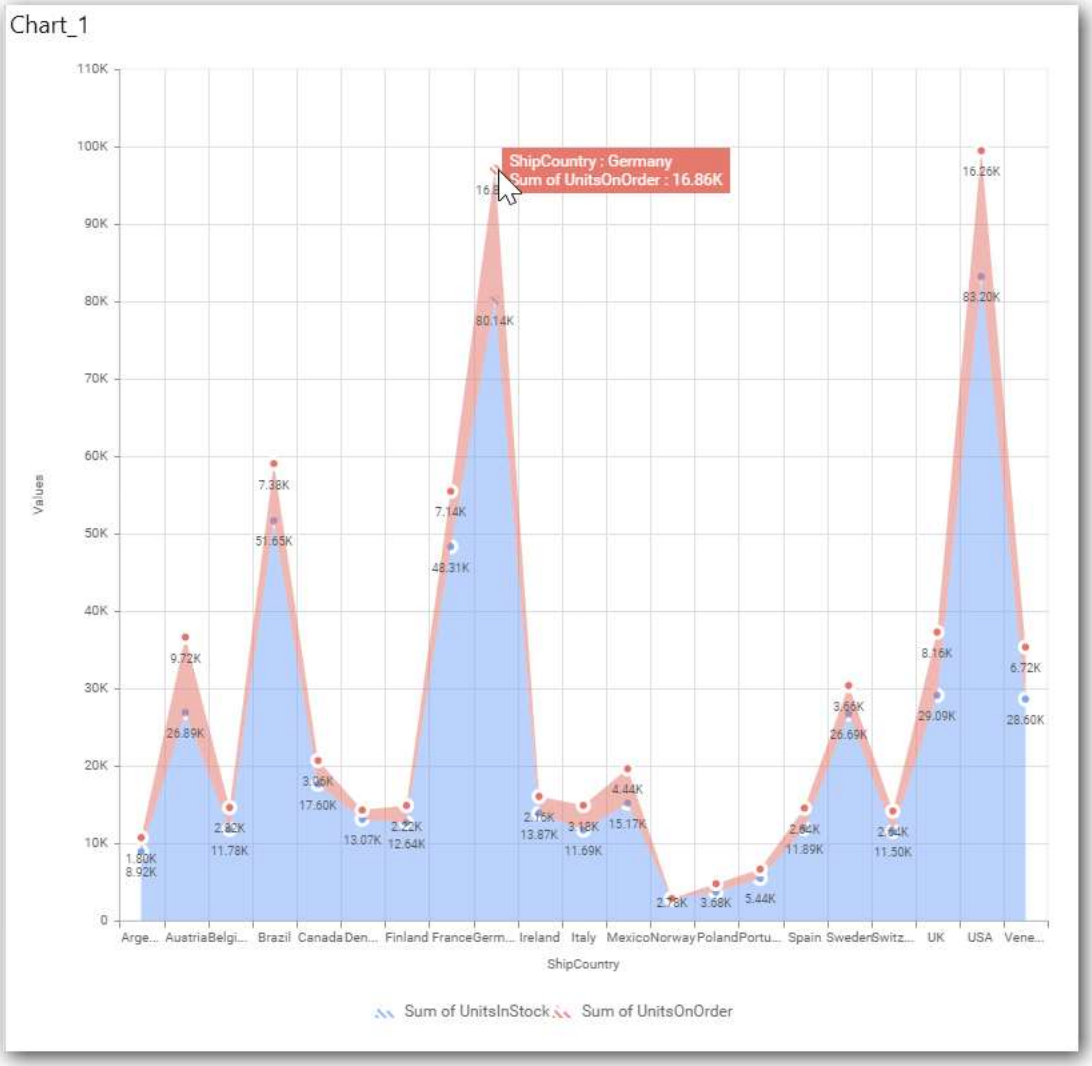


The following alert message will be shown.

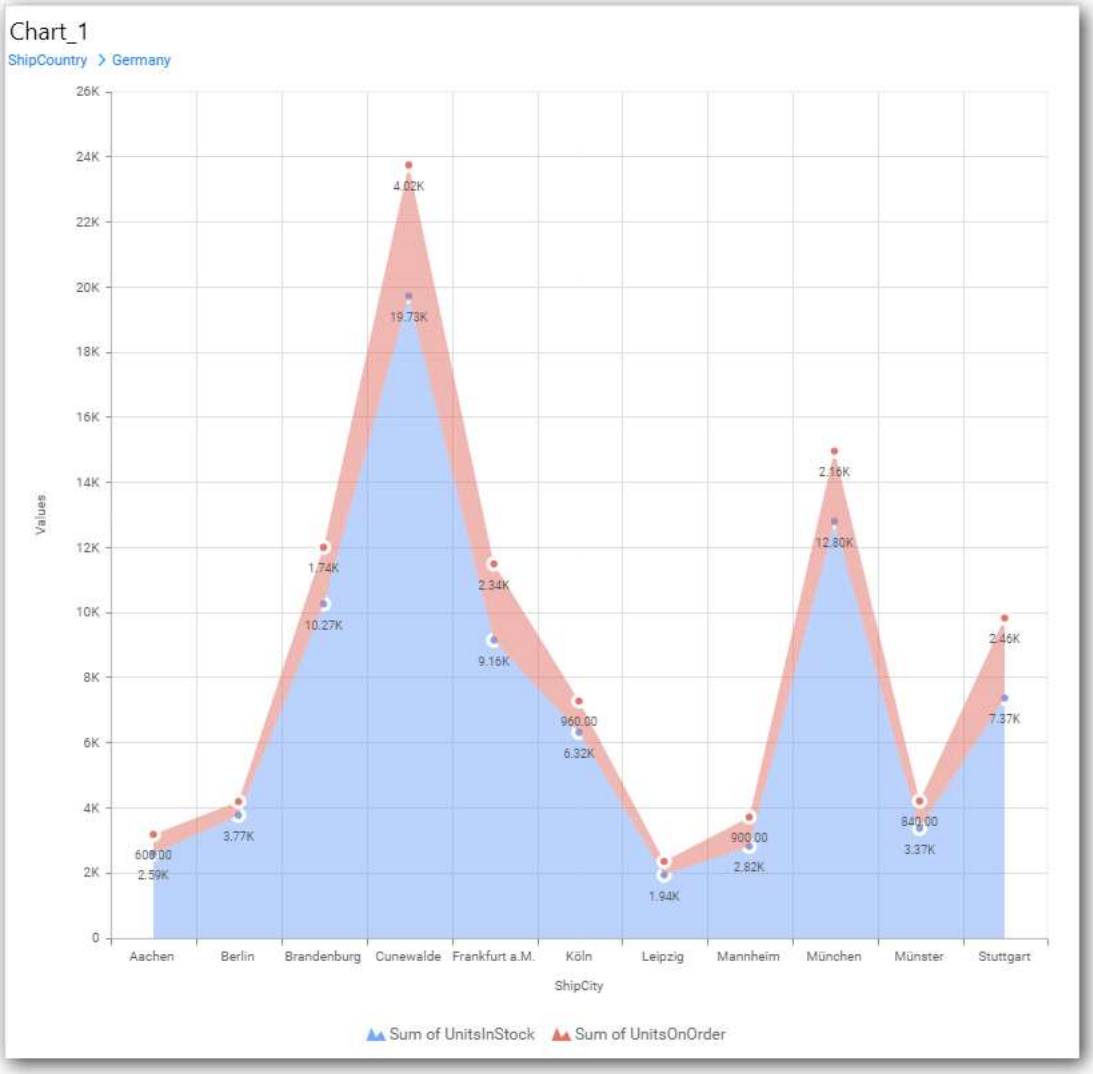


- If you choose Yes Drill down option will be enabled.

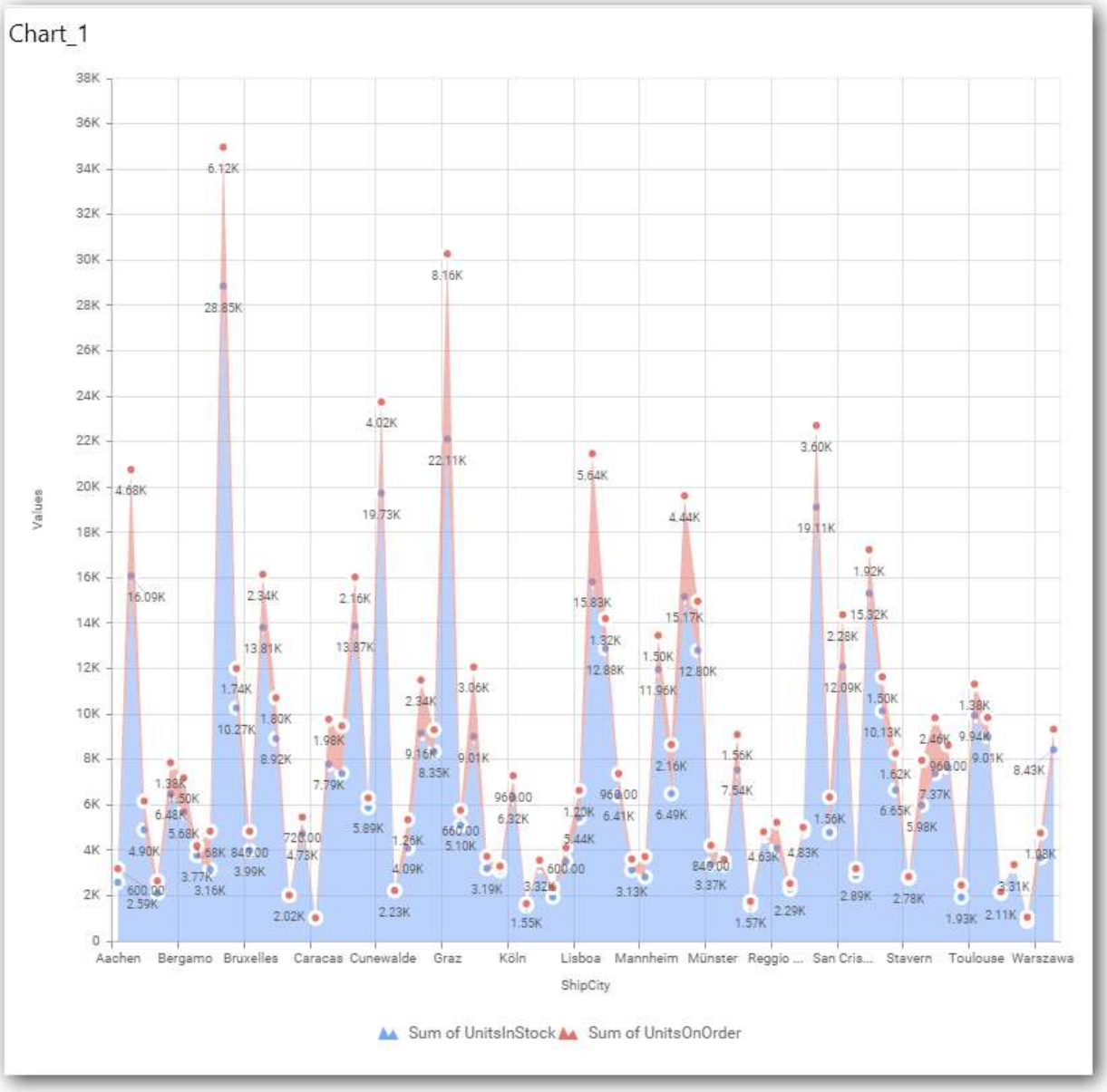
You can drill down the chart by clicking on the chart.



The drilled view of the chart is follows.

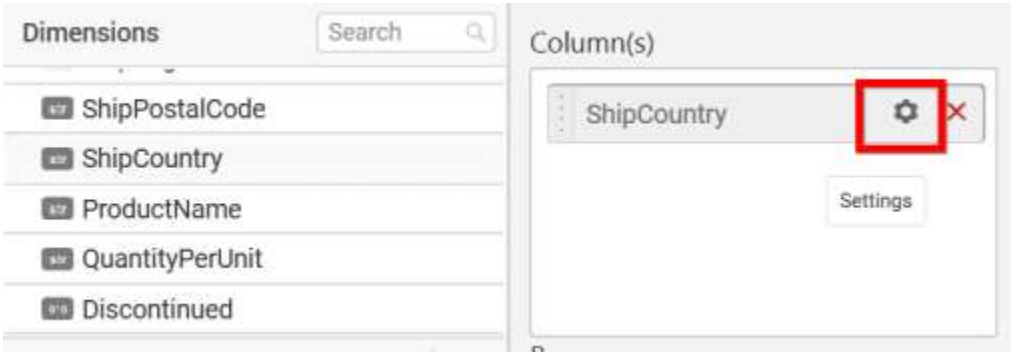


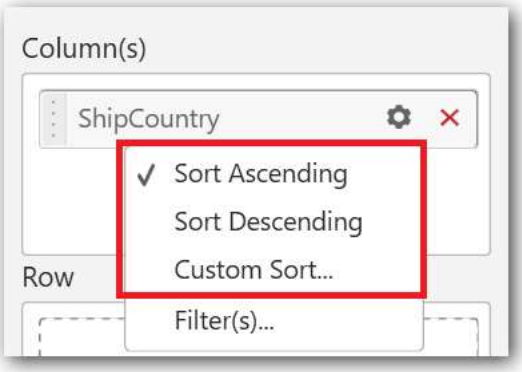
- If you click No the new Dimension value will replace old value.



You can also add Measures and Expression columns into Column(s) field.

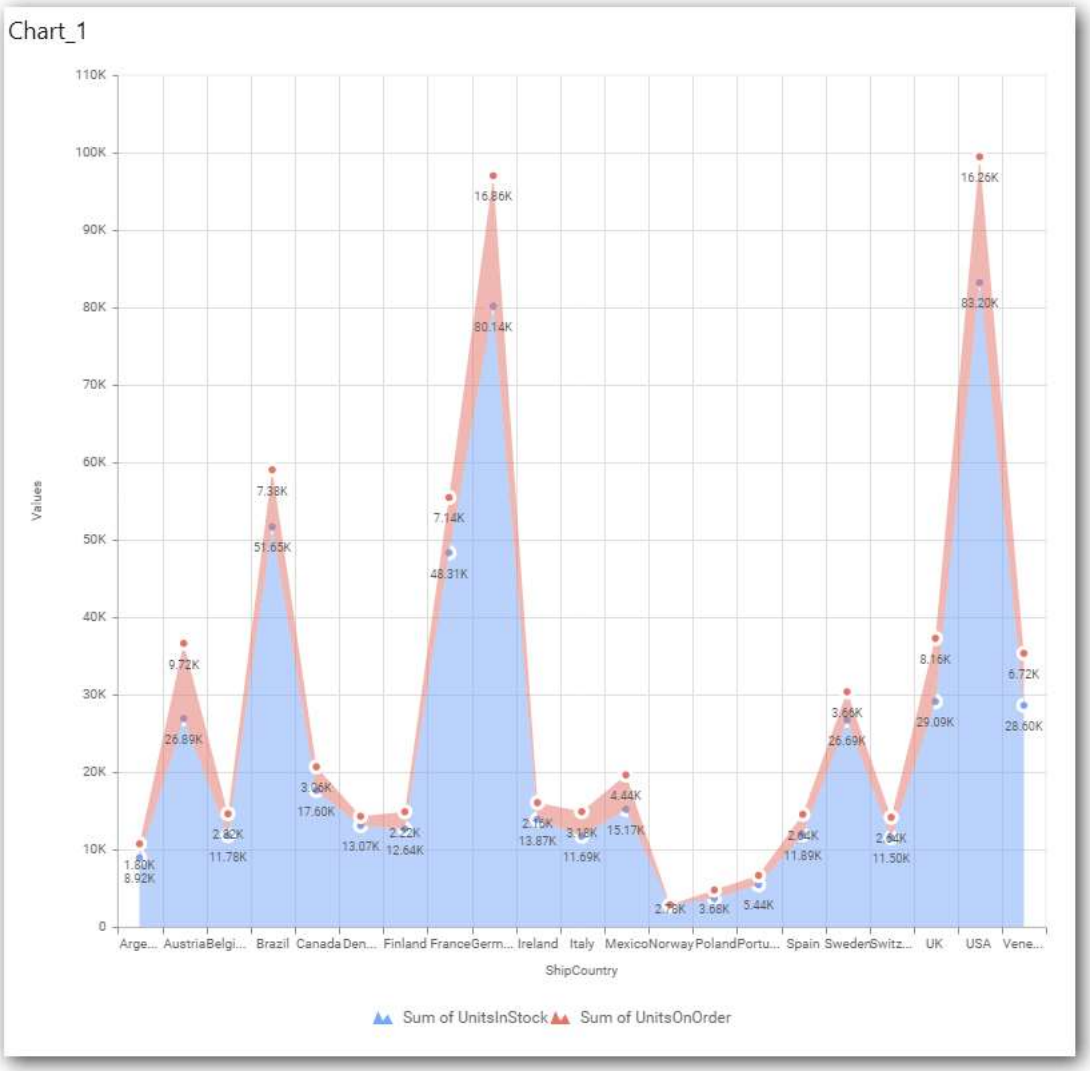
You have options to change the settings.



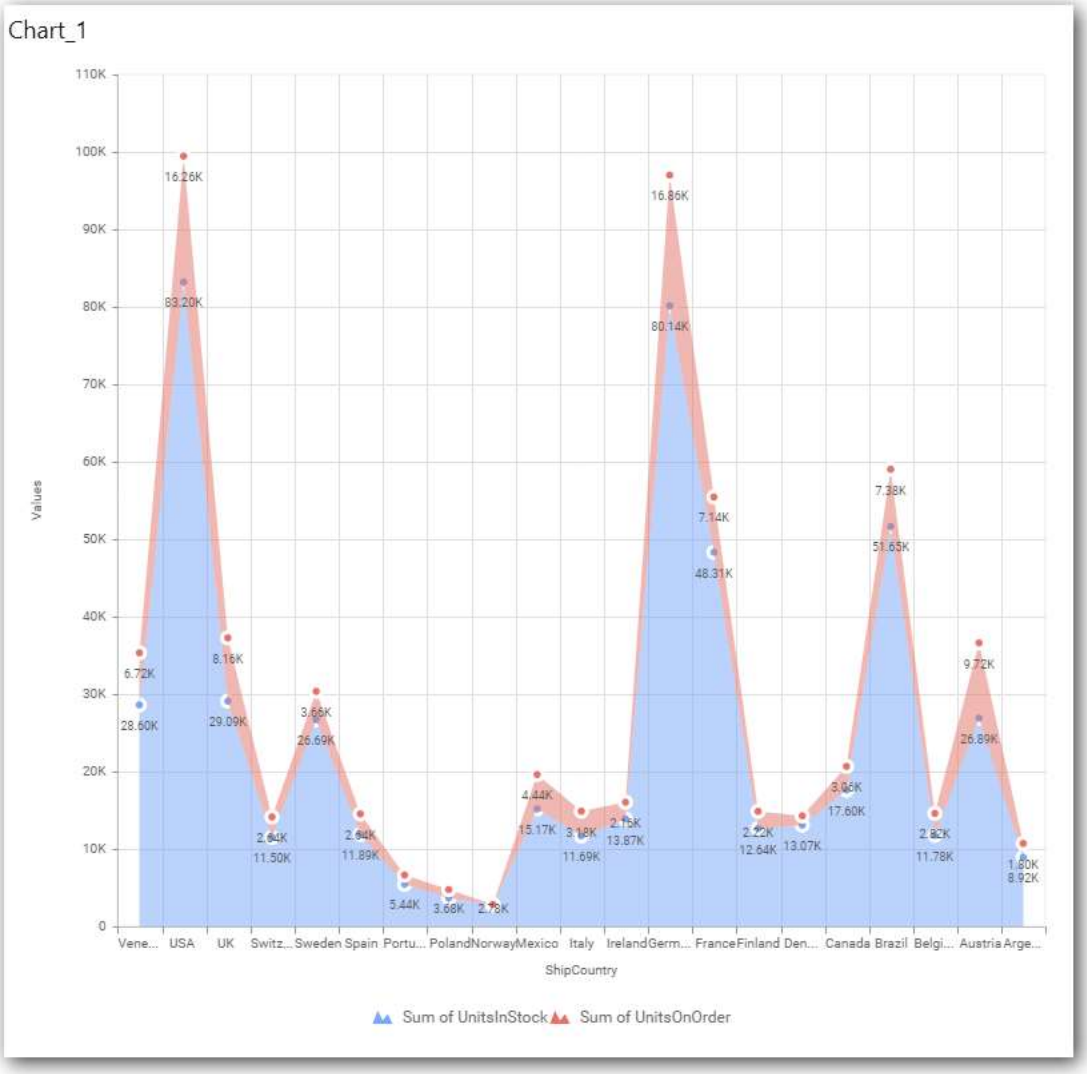


You can sort the chart either in Ascending or Descending series.

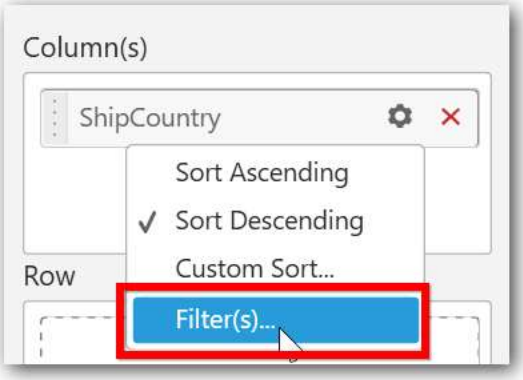
Ascending Order

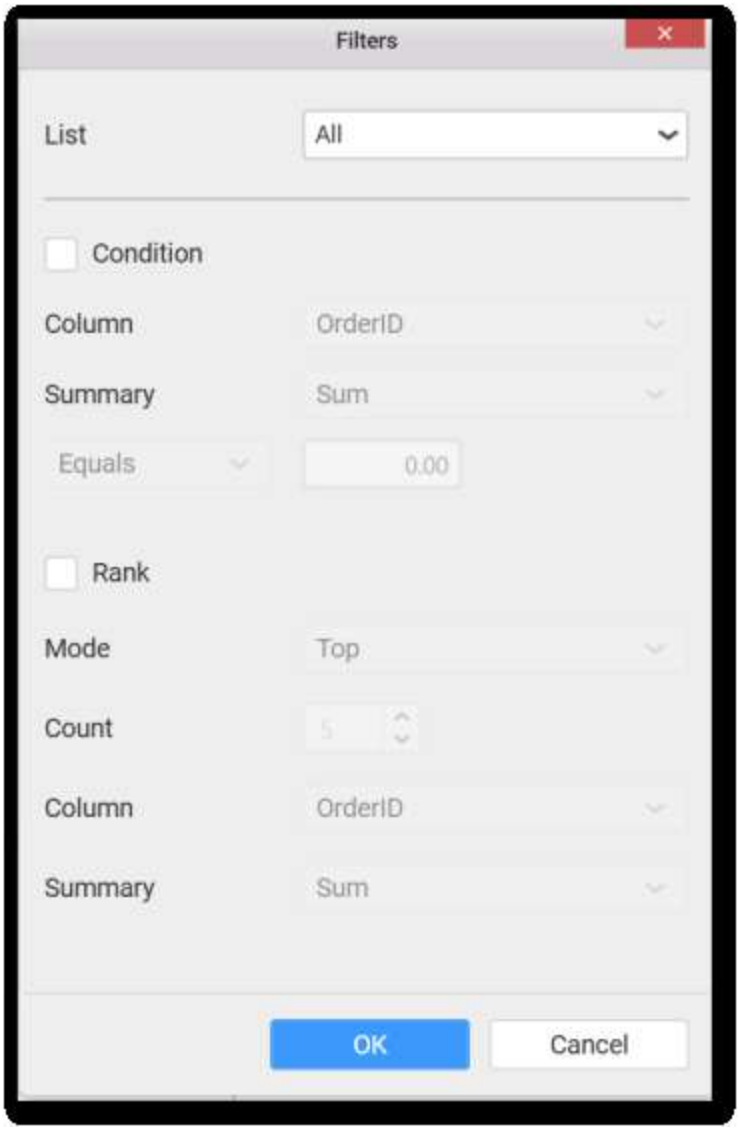


Descending order

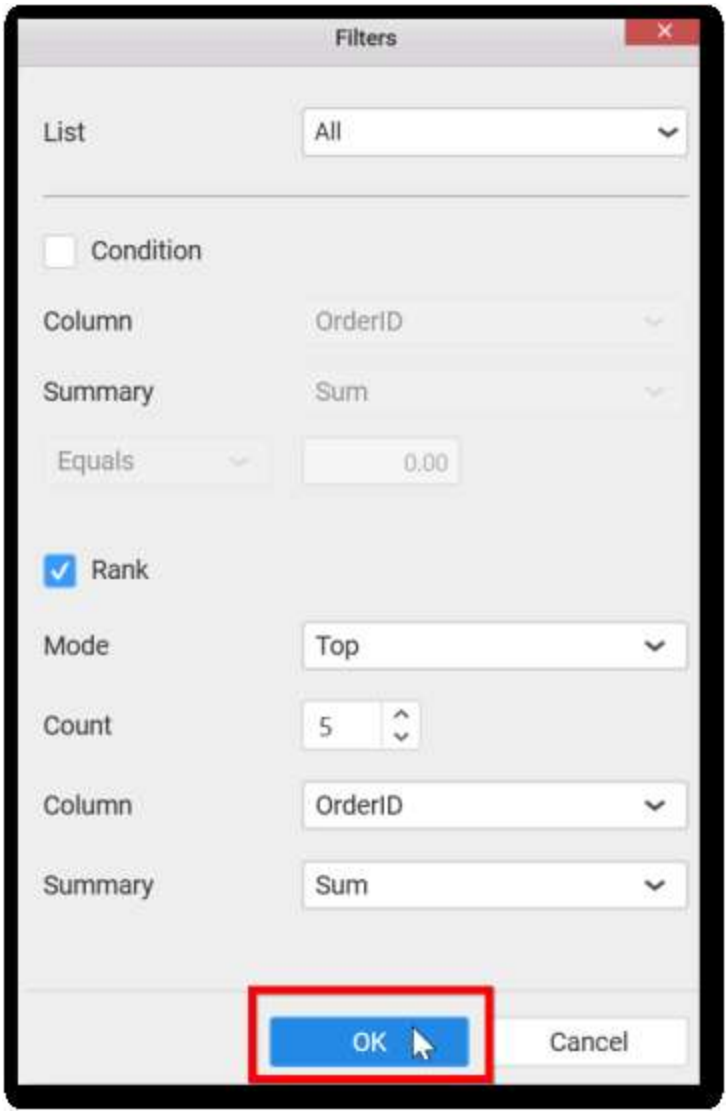


You can apply a filter.

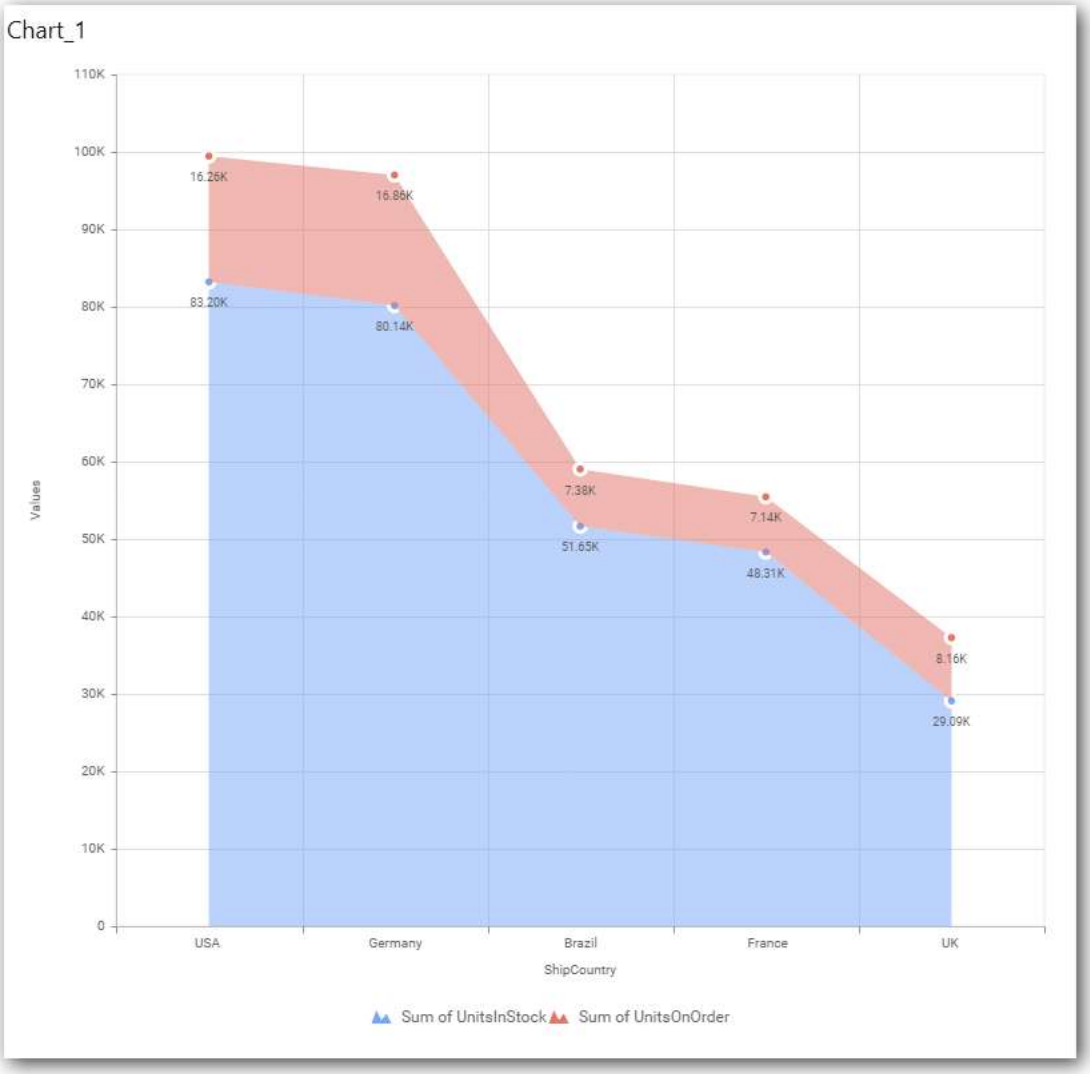




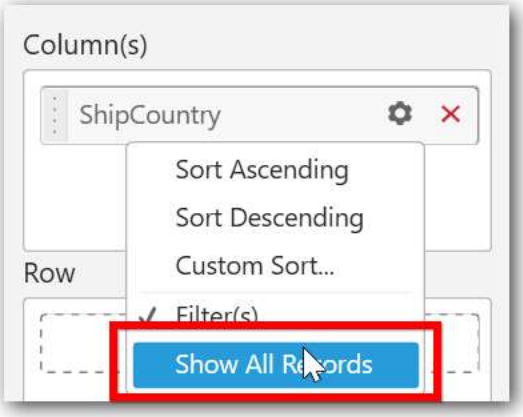
Select the Conditions and Rank you need.



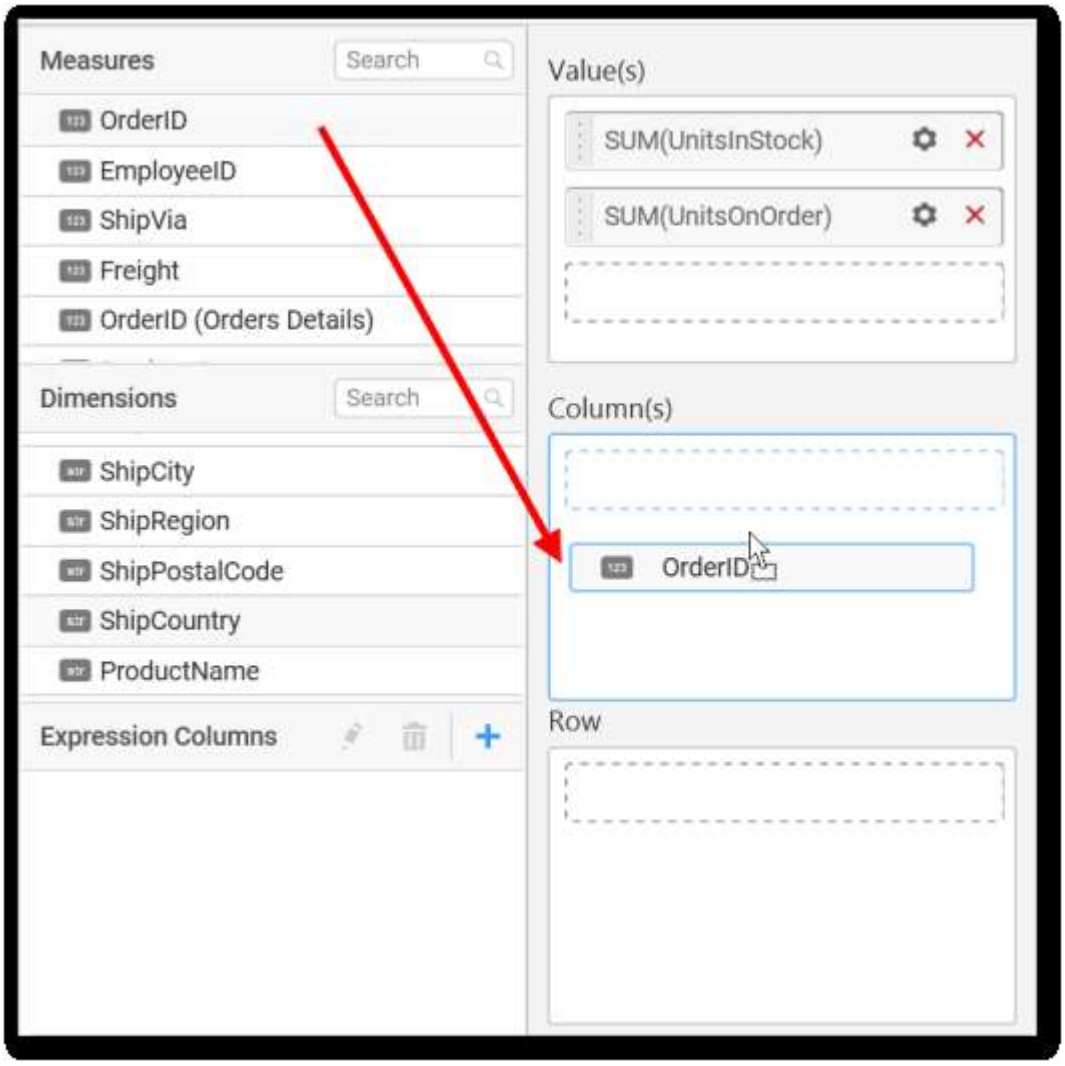
Now the chart will be rendered like this.

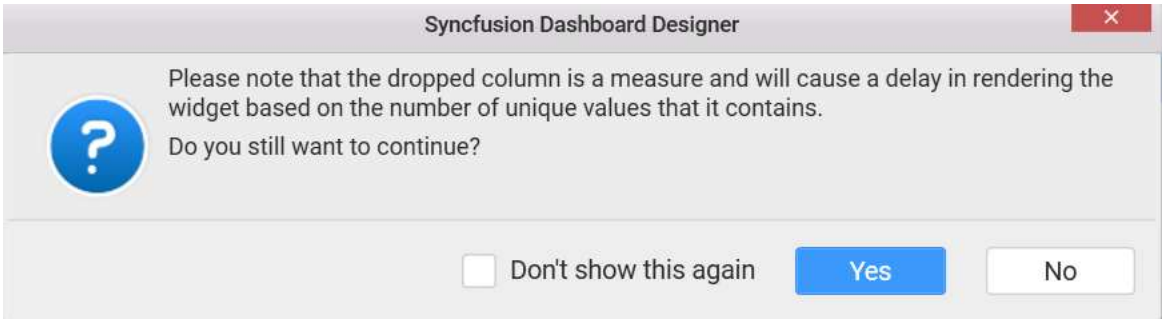


To show all records again click on Show All Records.



On adding measures into Column(s) will show the following alert.

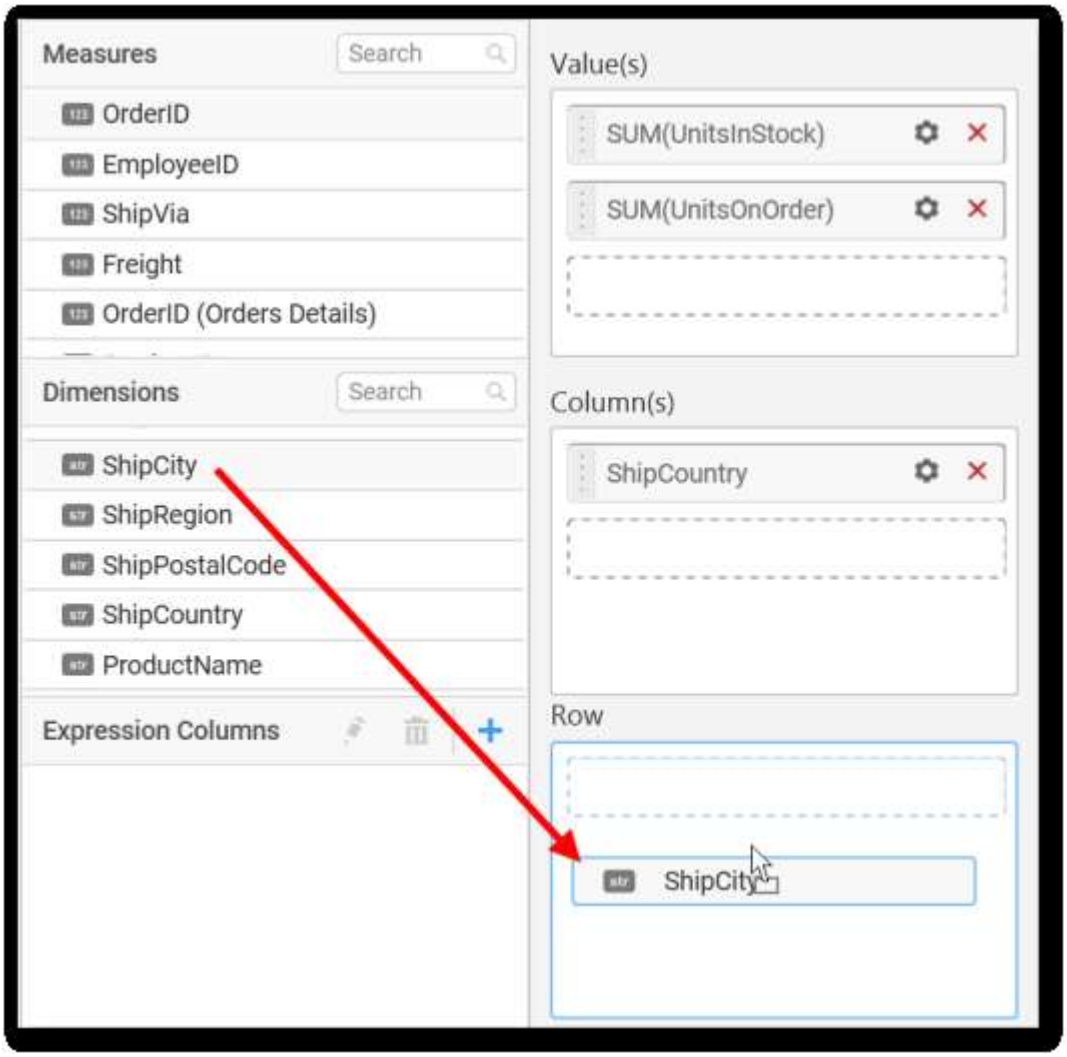




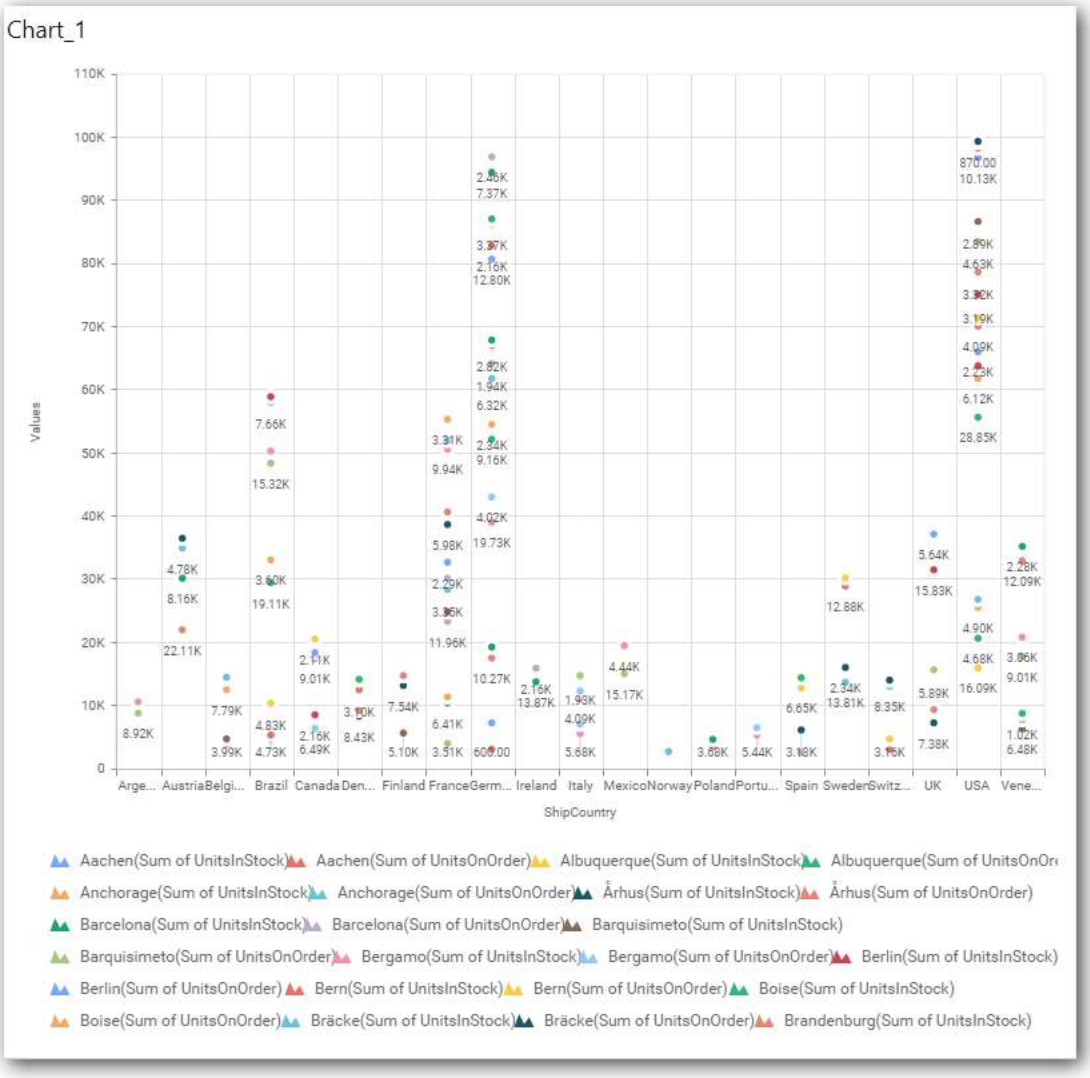
To continue select Yes, otherwise select No.

Assigning Row

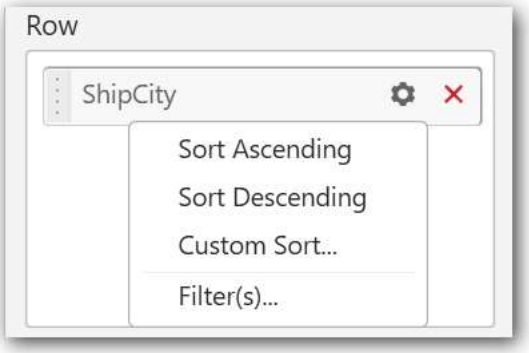
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s).



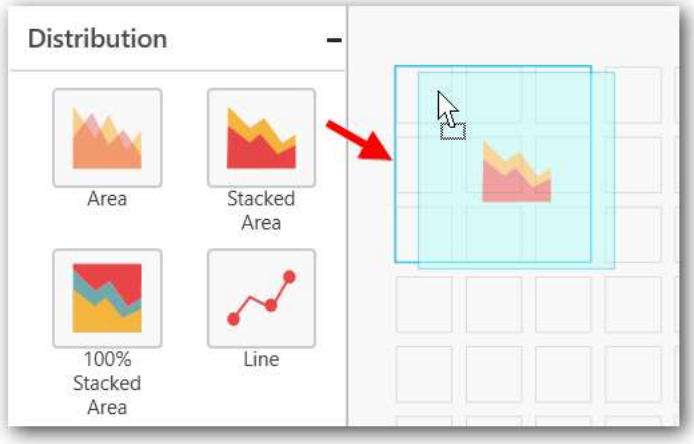
How to configure SSAS data to stacked Area Chart?

Stacked Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you

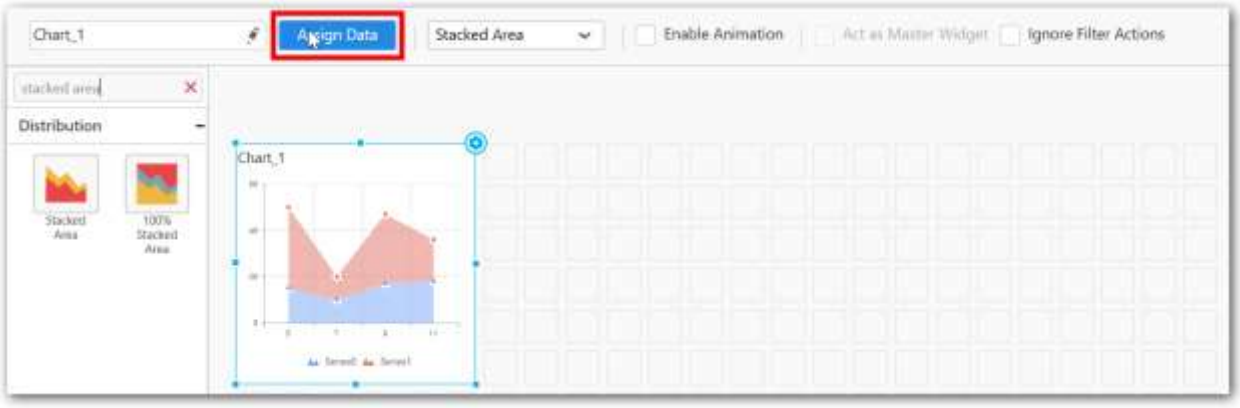
would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuration of SSAS data to Stacked Area chart

Drag and drop the Stacked Area chart widget into canvas and resize into your required size.

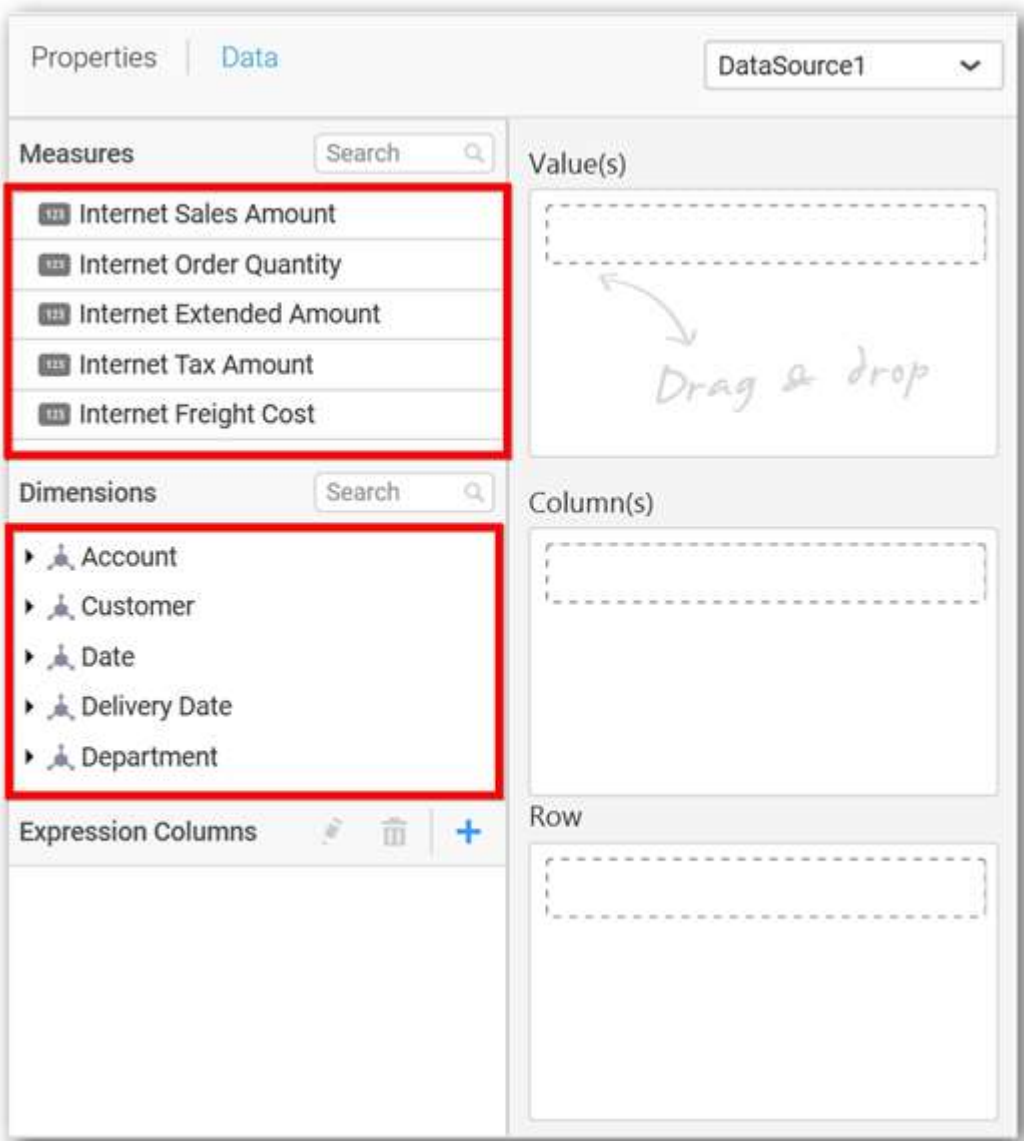


Select the dropped widget using mouse.



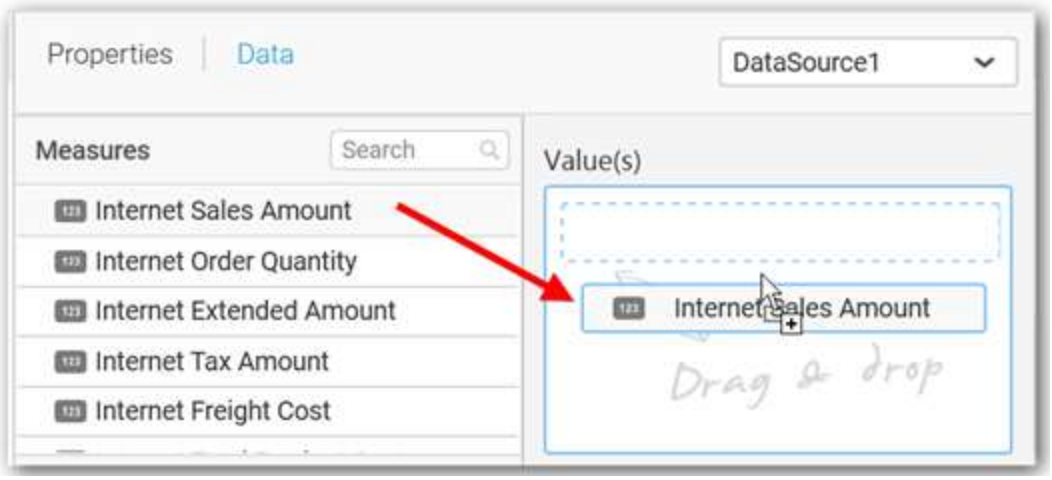
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions.

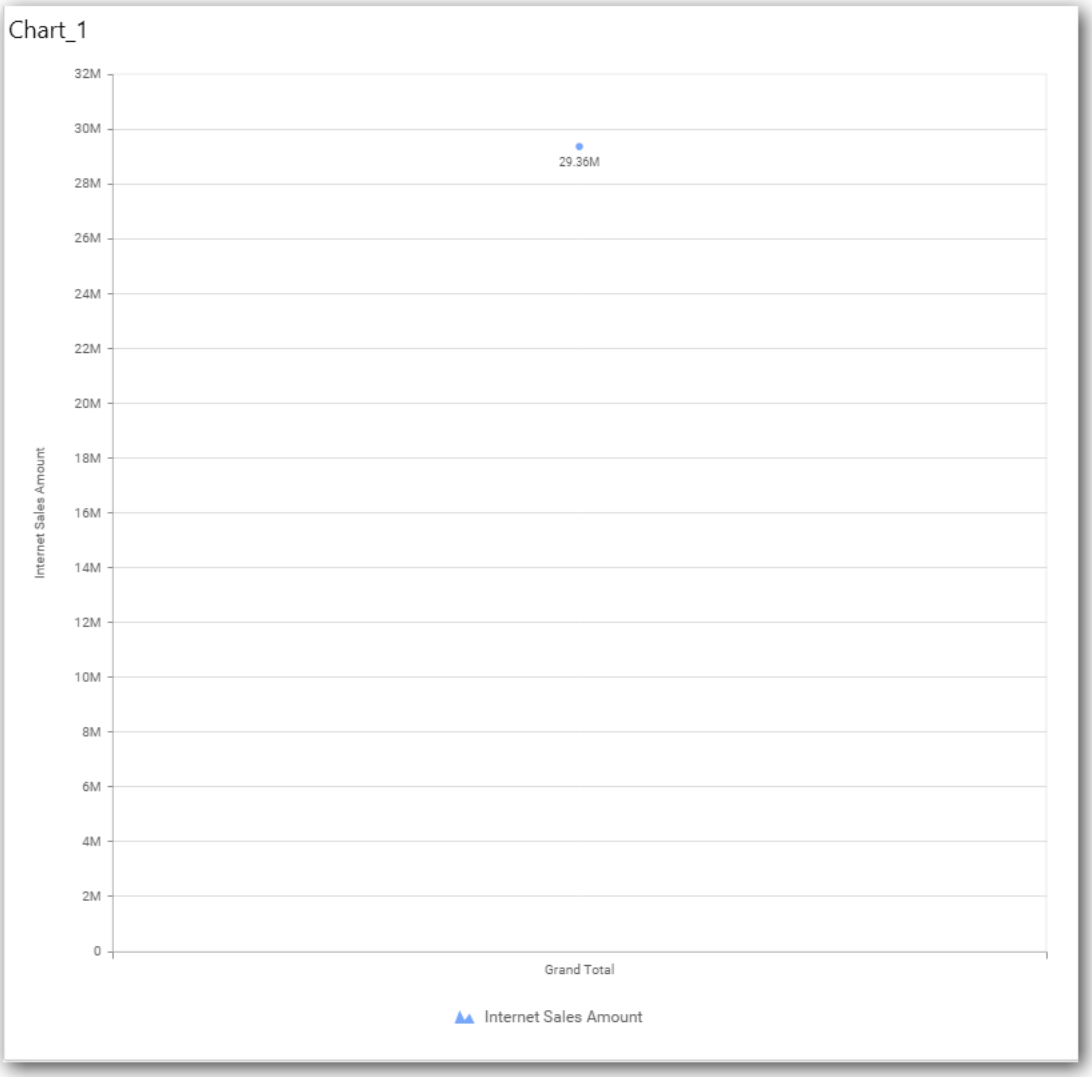


Assigning Value(s)

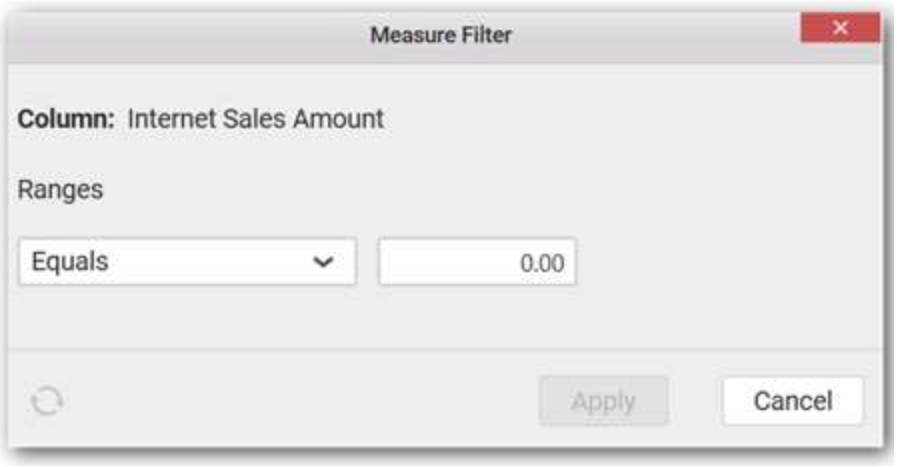
Drag and drop a column under Measures category into Value(s) section.



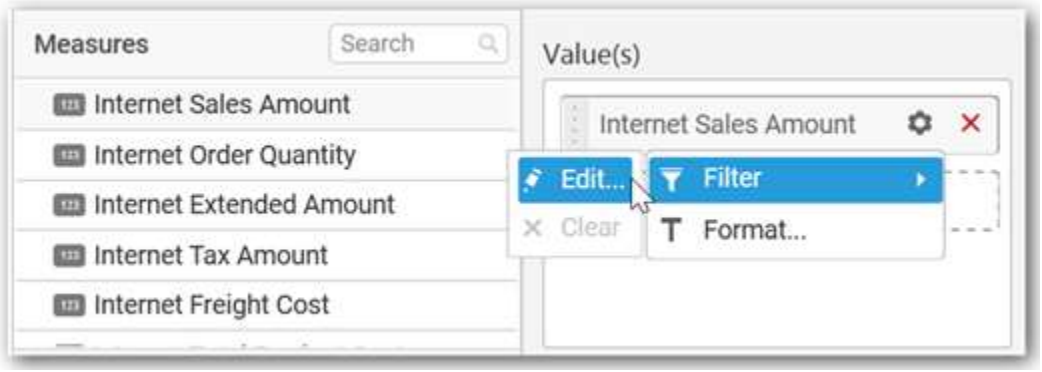
Now the chart will be rendered like this.



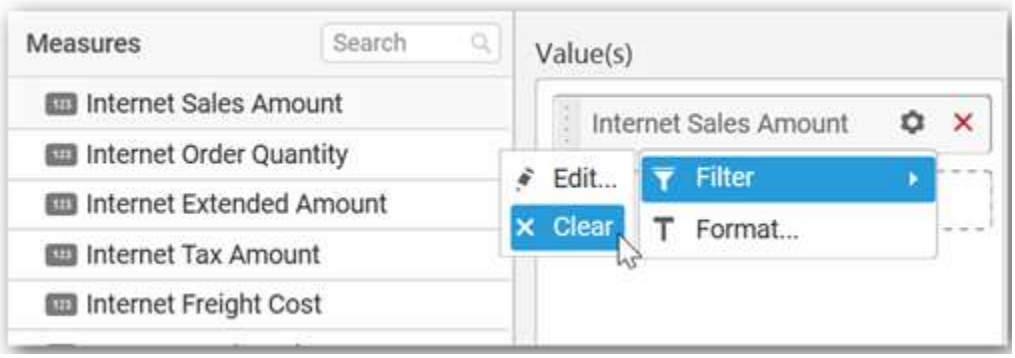
Define the filter criteria to match through choosing Edit option in Filter menu item.



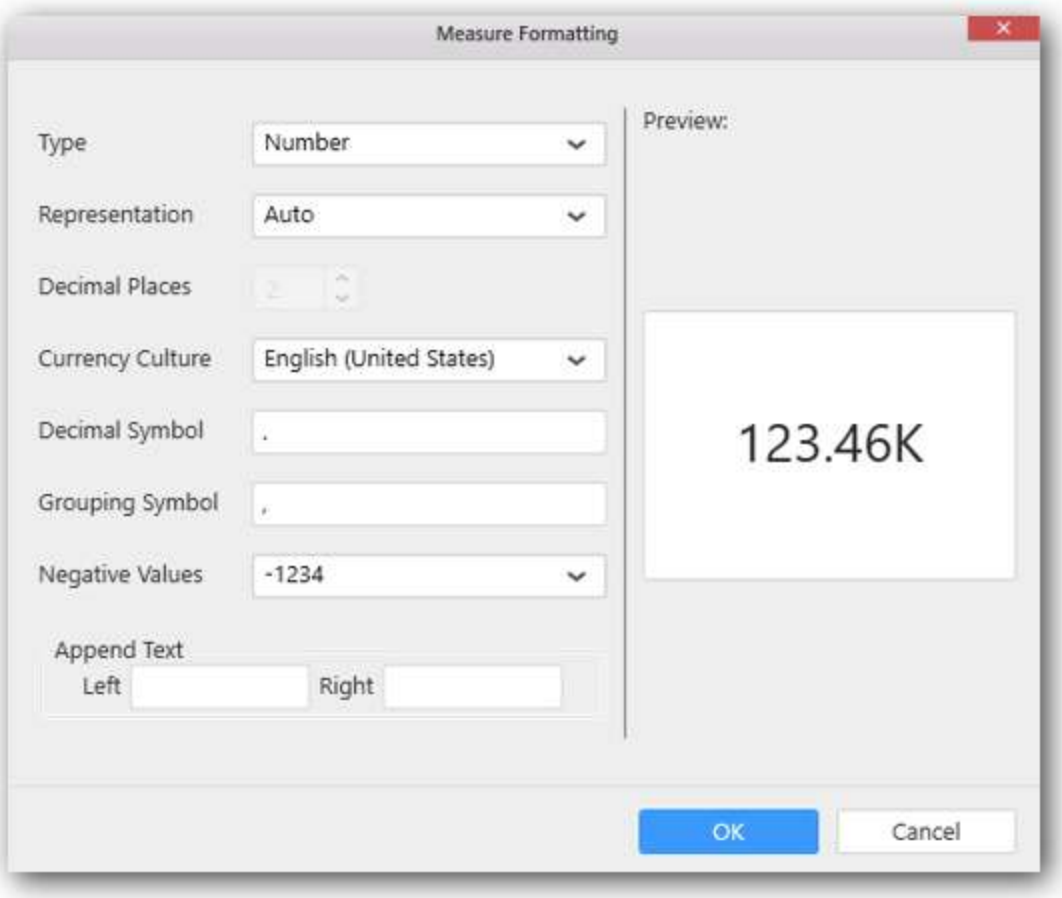
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



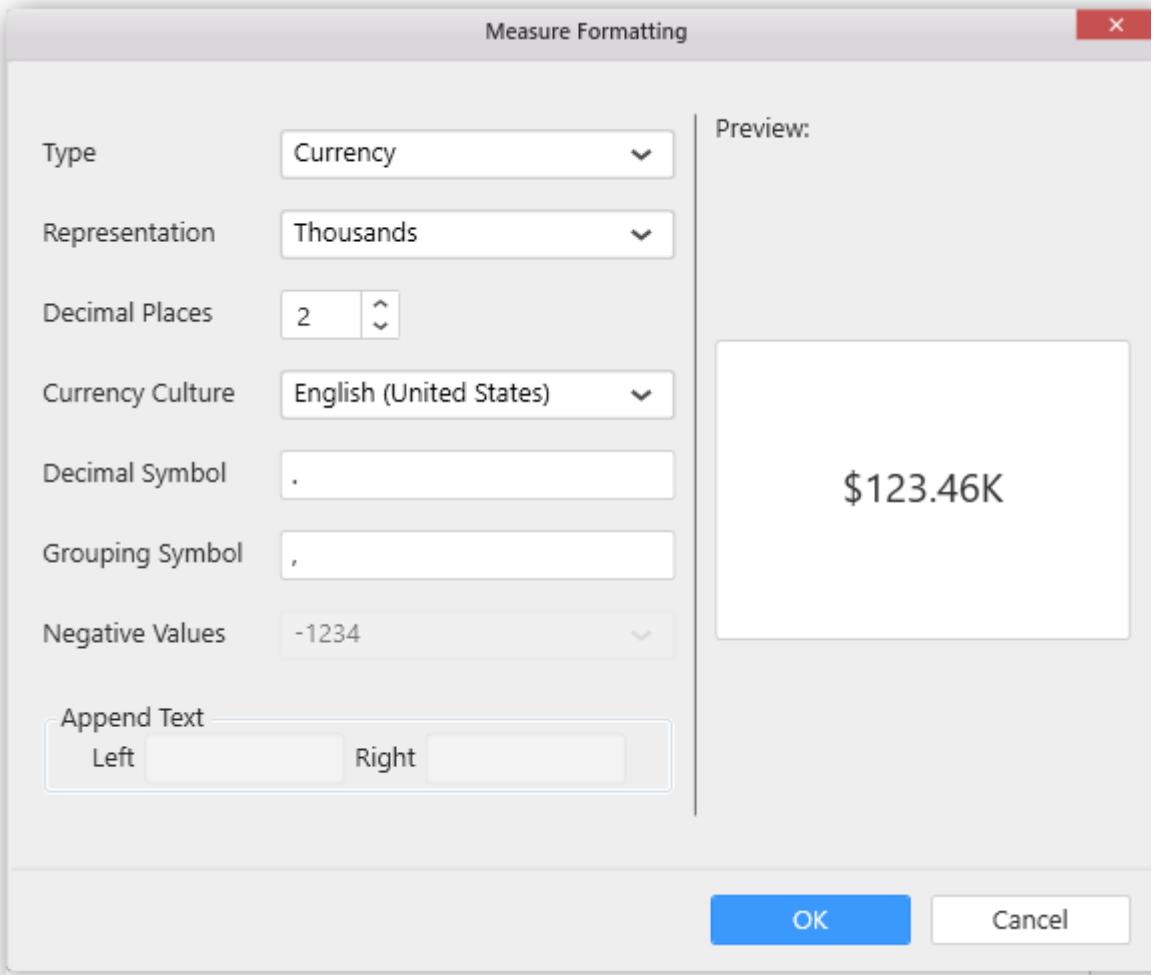
Select Clear option to clear the defined filter.



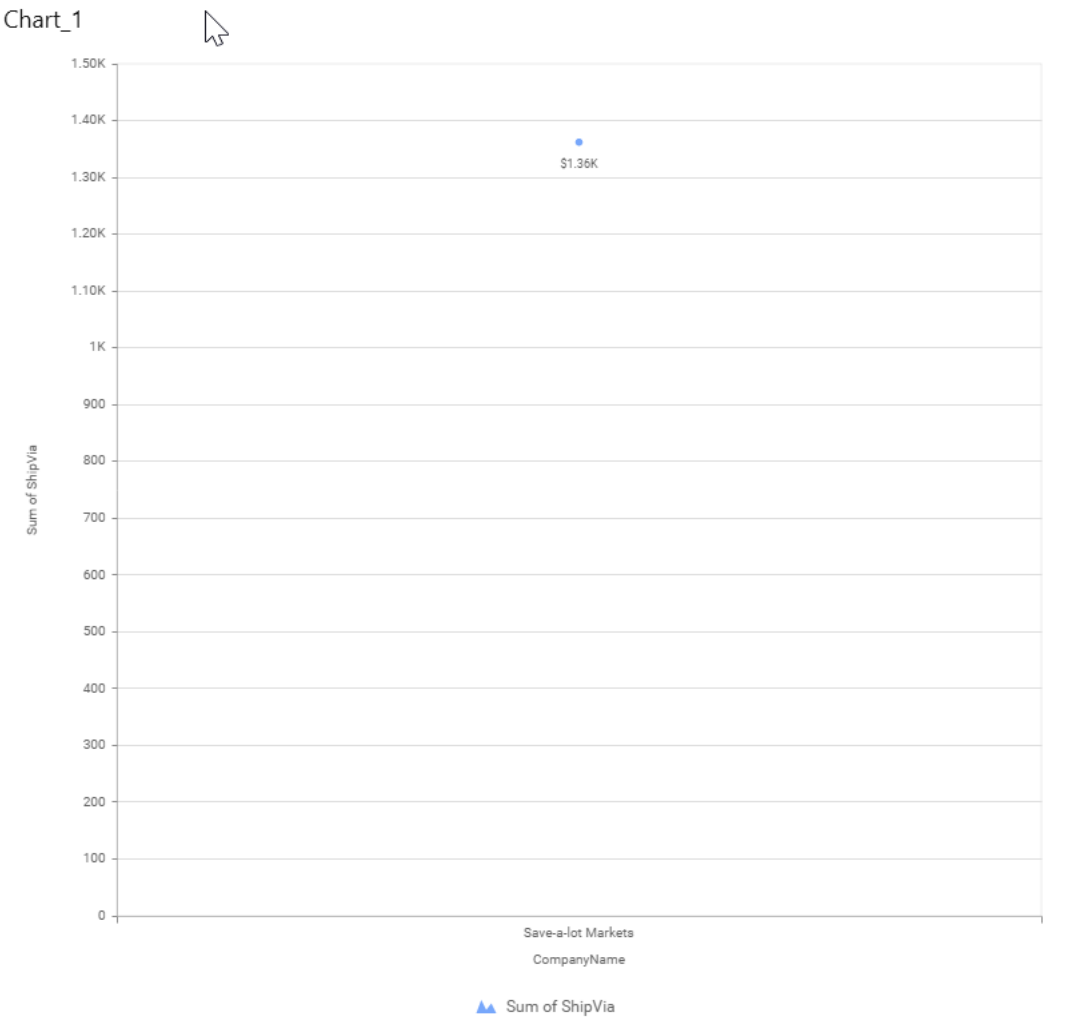
Select Format option to define the display format to the values in the column through Measure Formatting window.



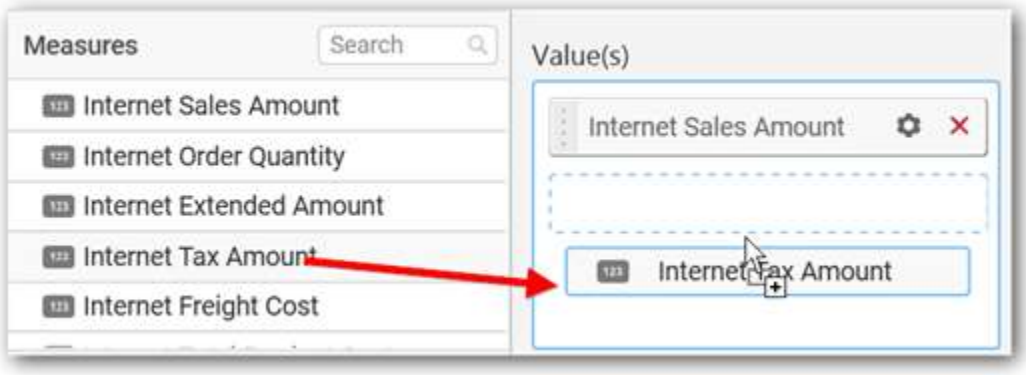
Choose the options you need and click OK

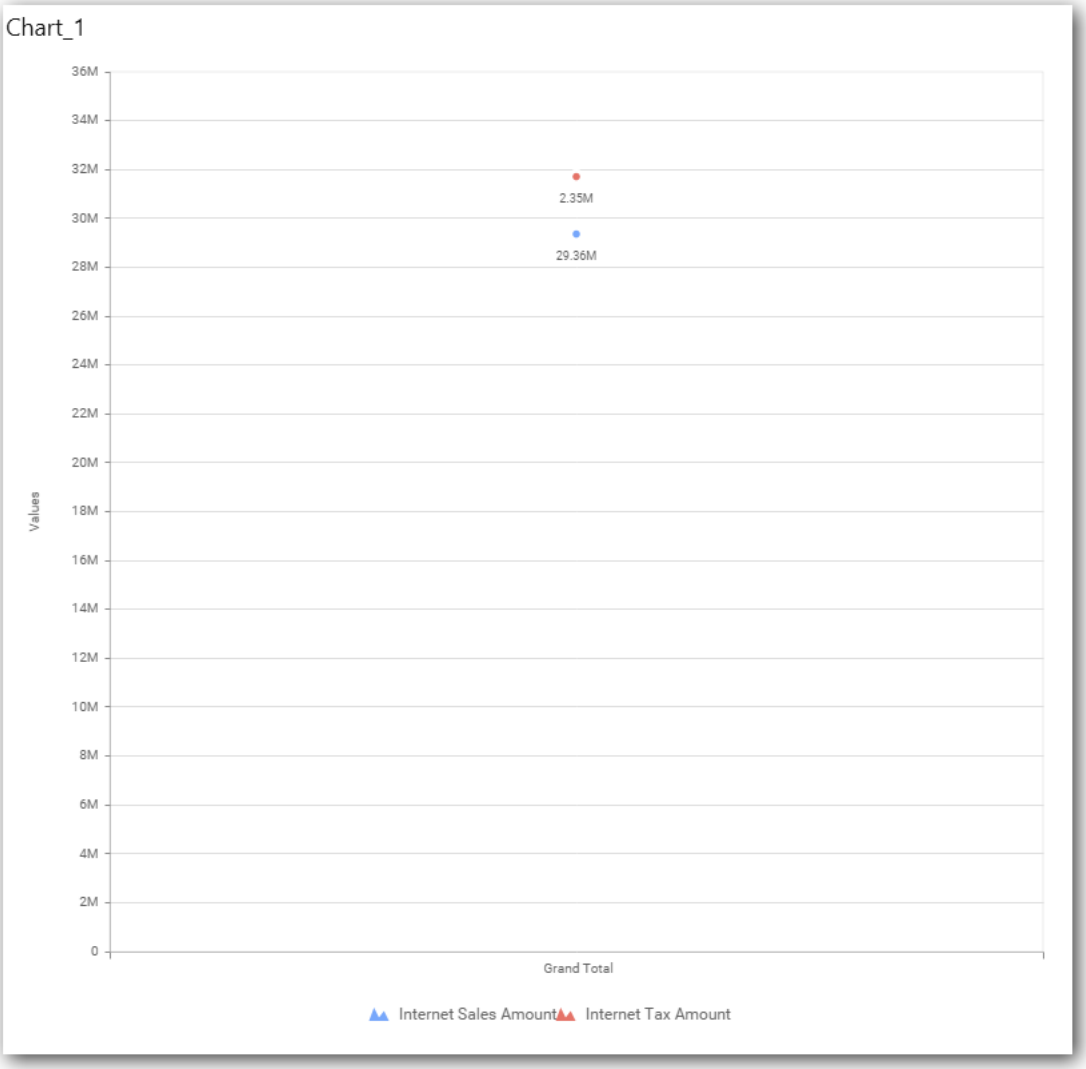


Now the Chart will be rendered like this.



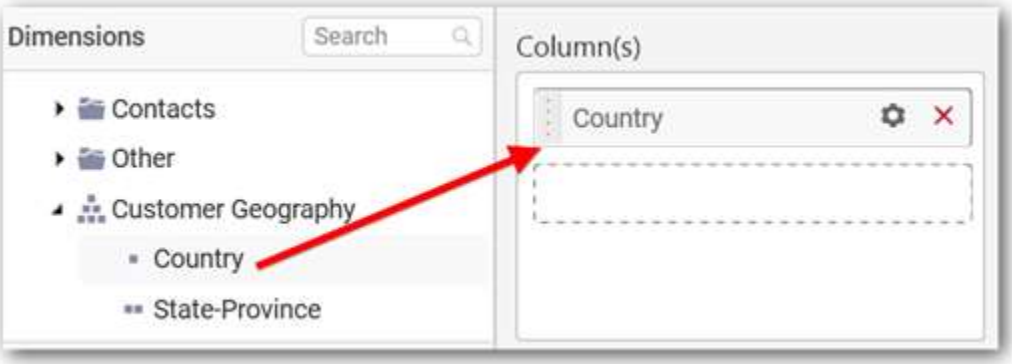
You can also add more than one column to the Value(s) section.

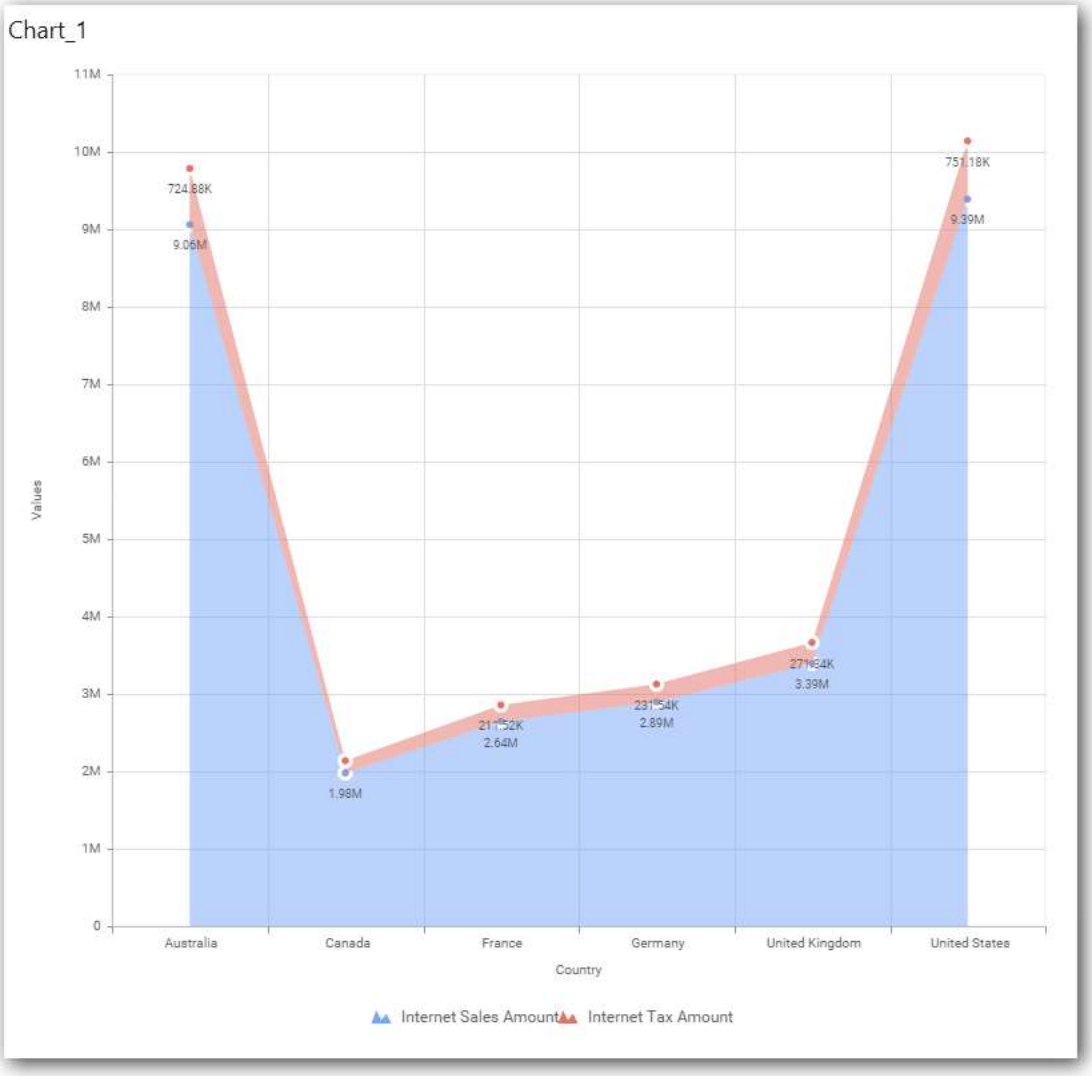




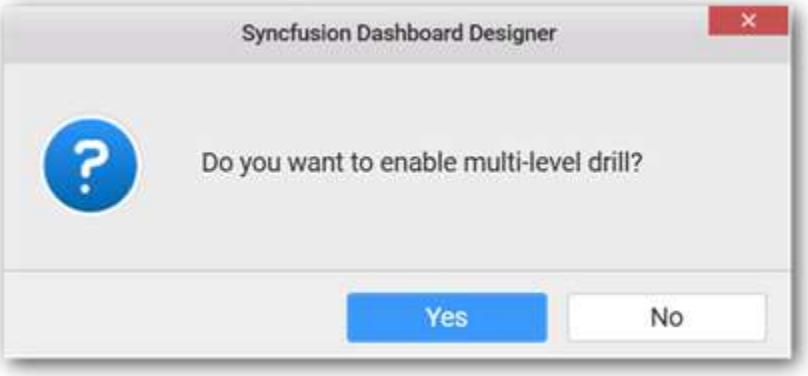
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.



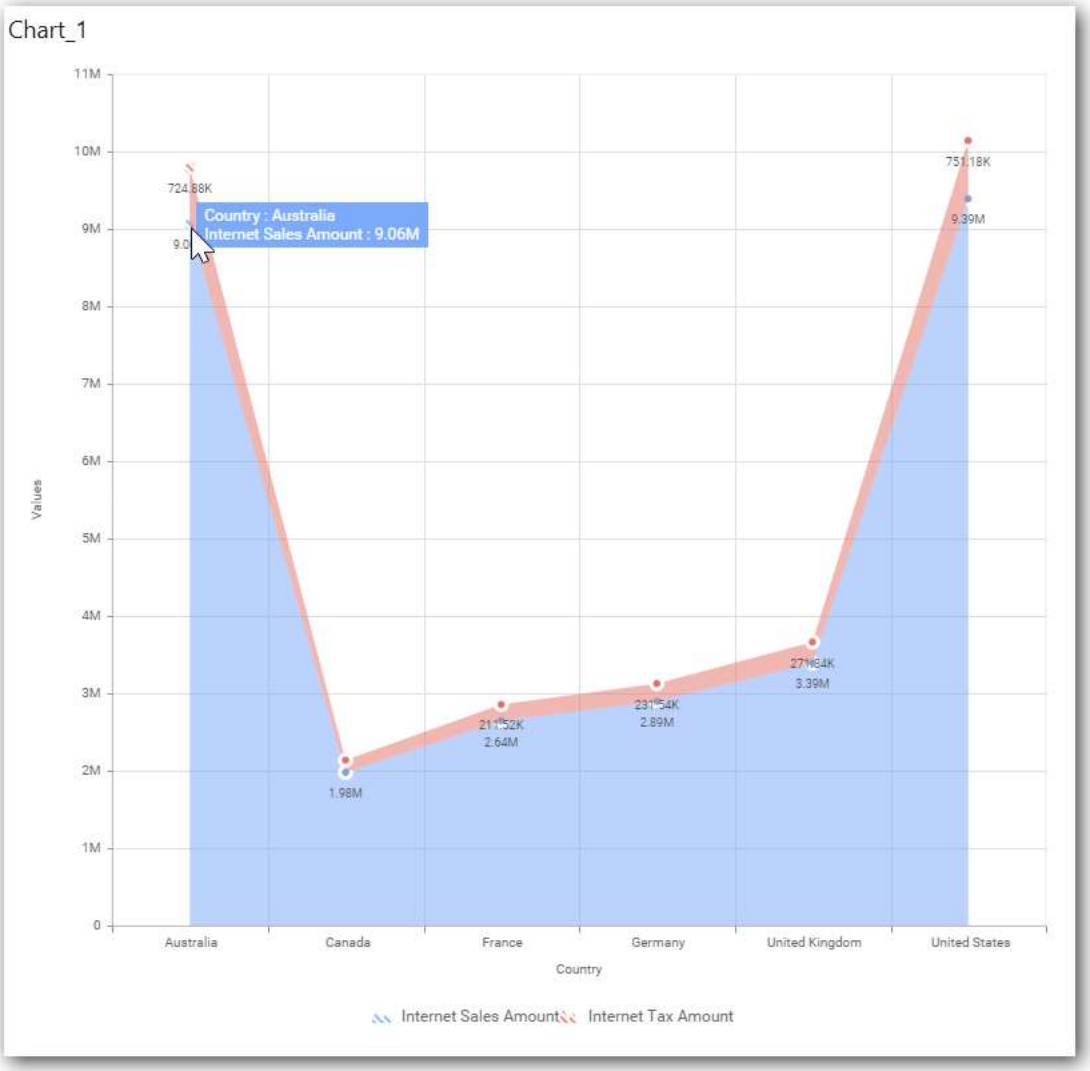


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

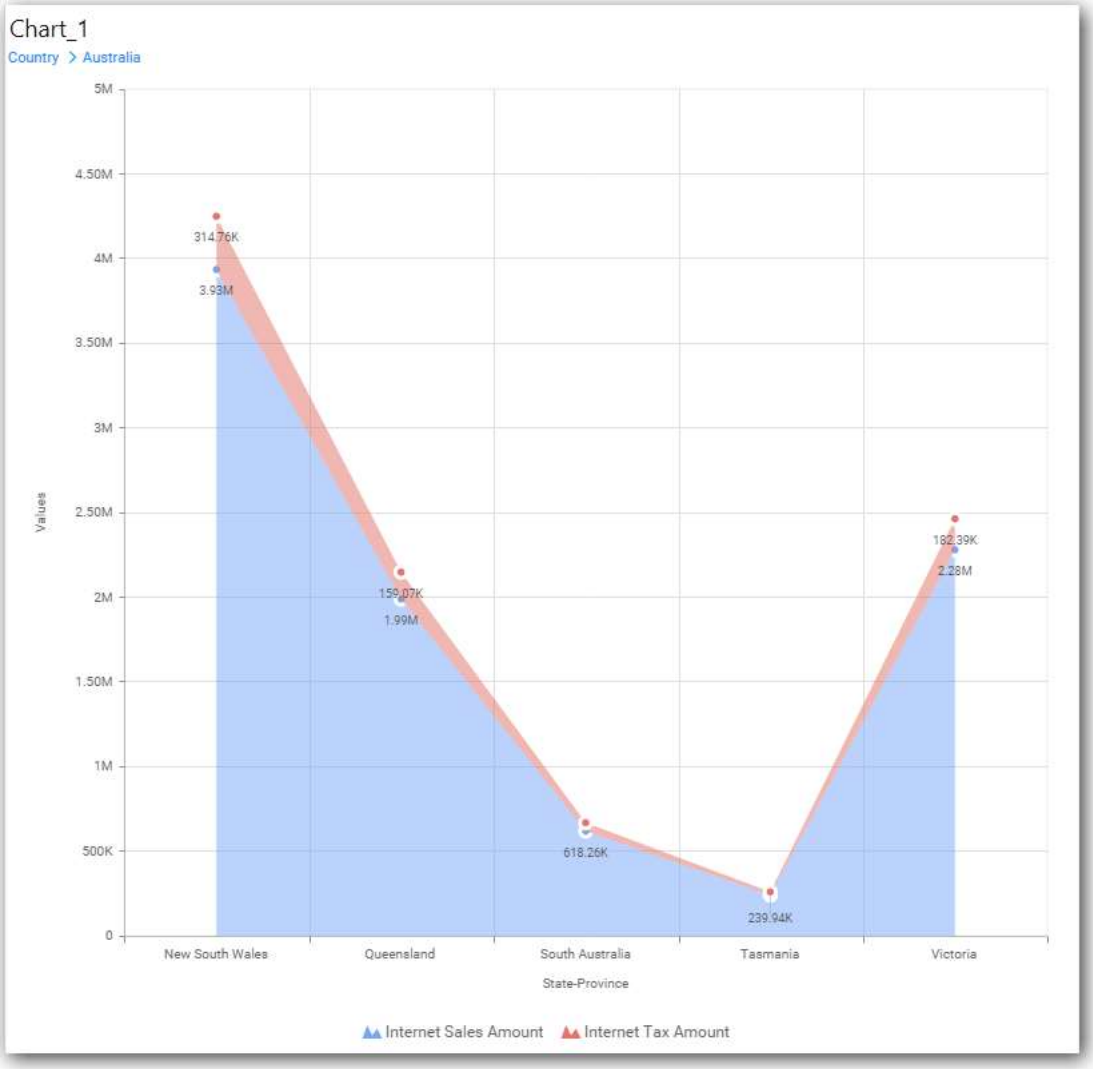


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

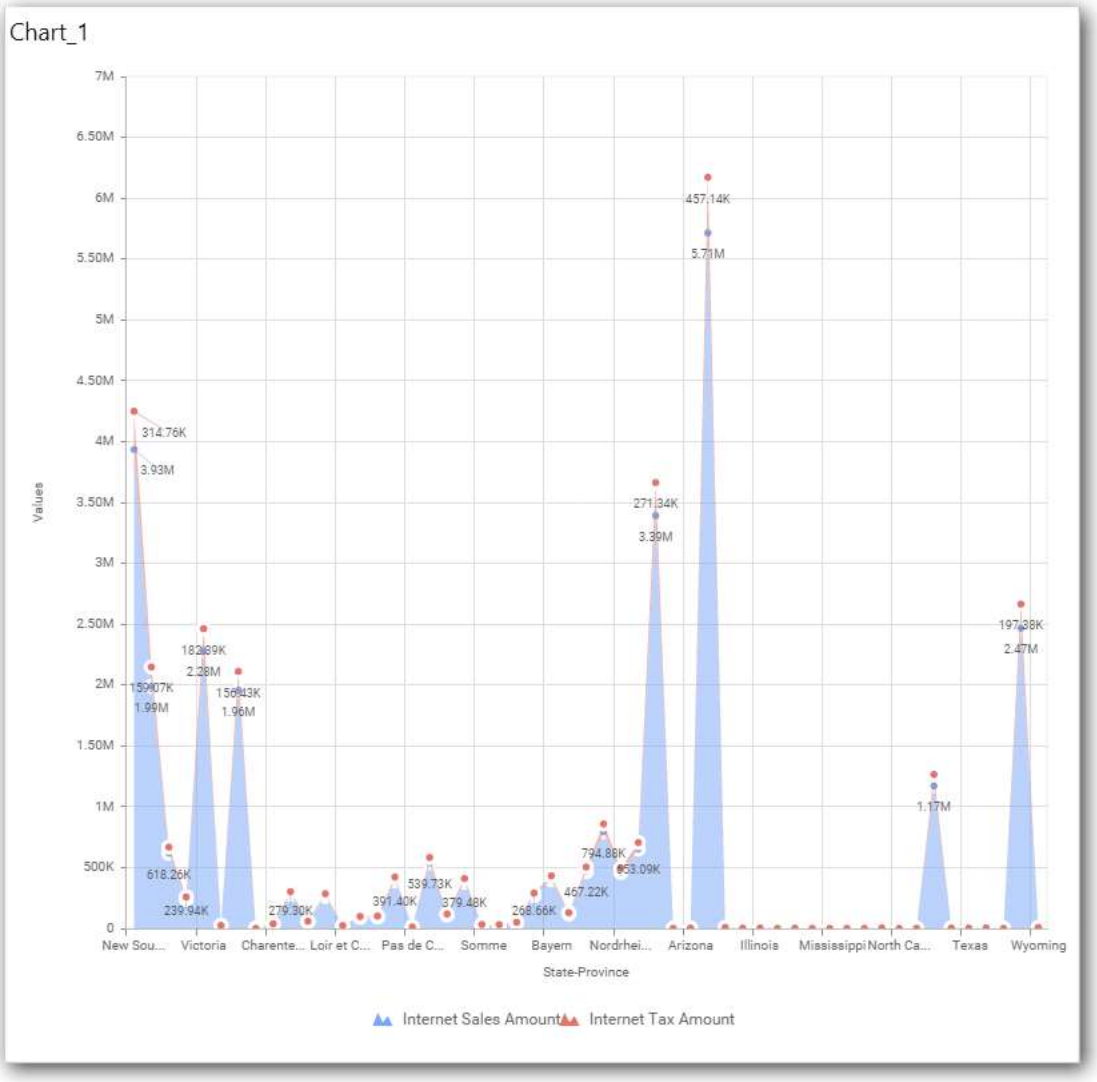
Click the respective data value marker in chart to drill into its inner level.



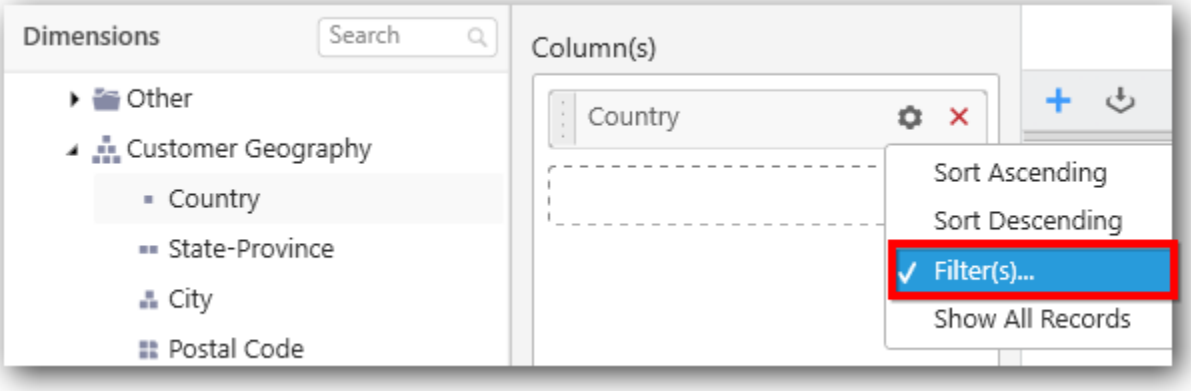
The drilled view of the chart is follows.



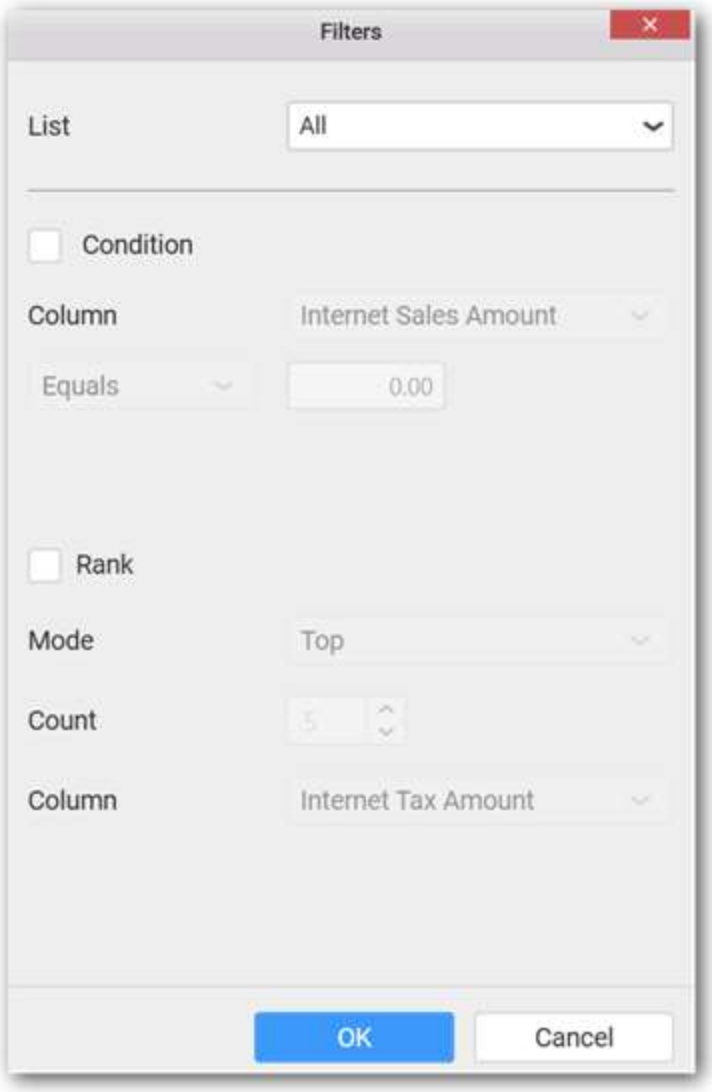
Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level.



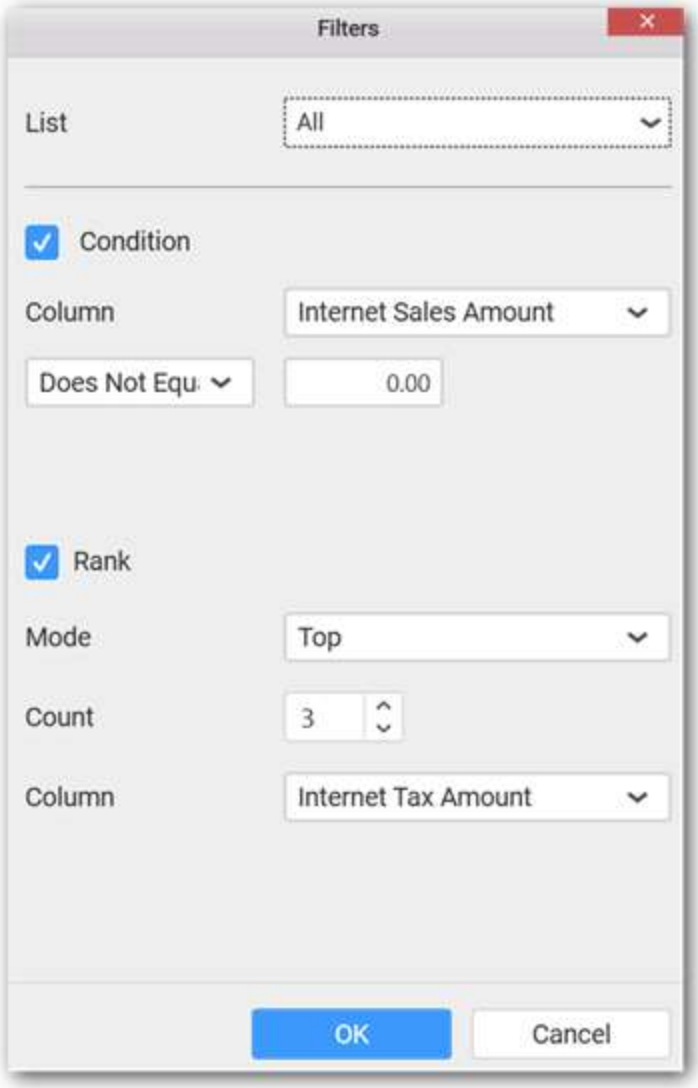
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



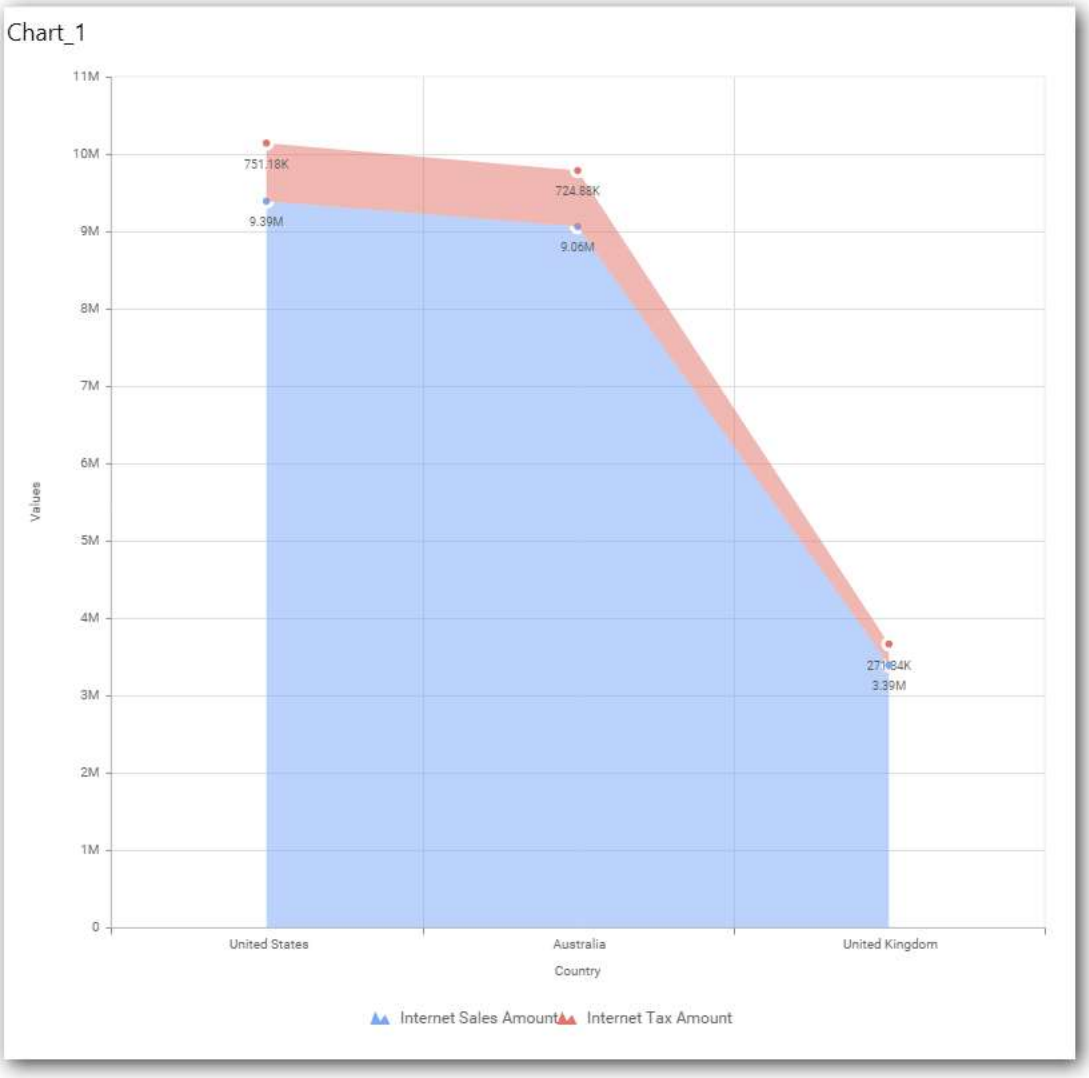
Select Filter(s)… to launch the Filters window.



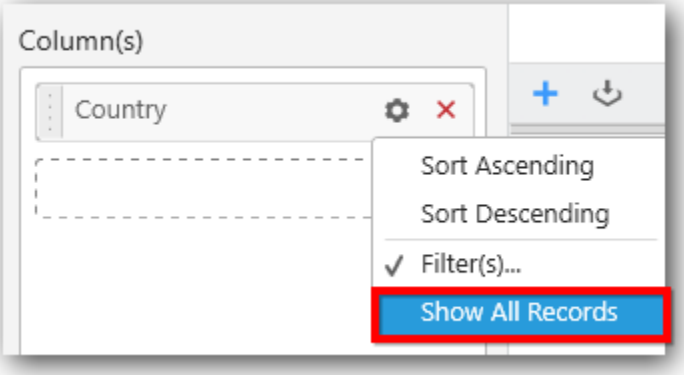
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

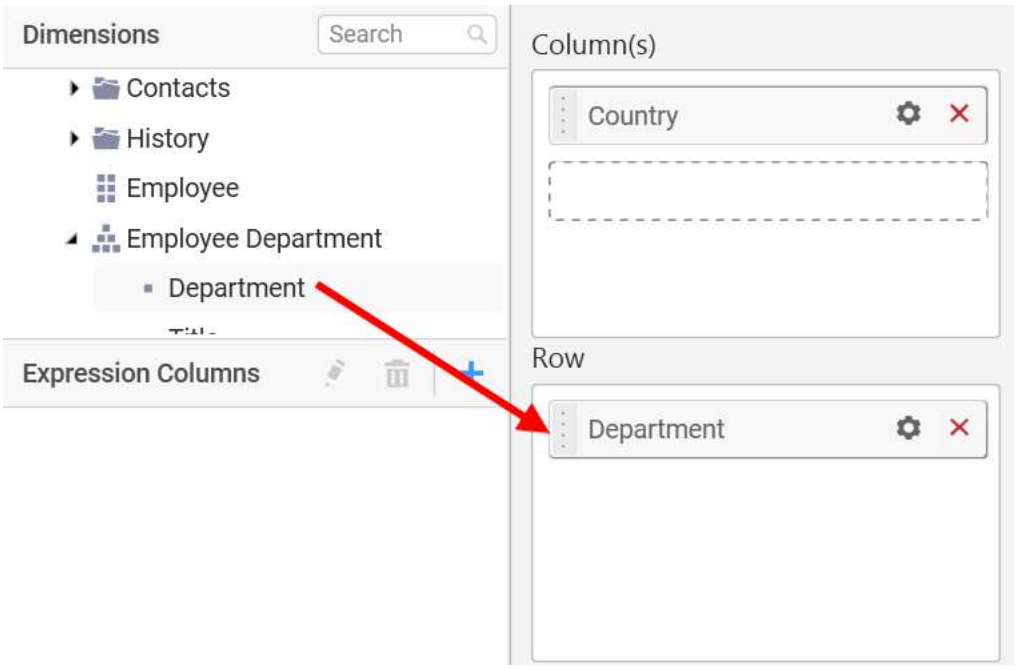


To show all records again click on Show All Records.

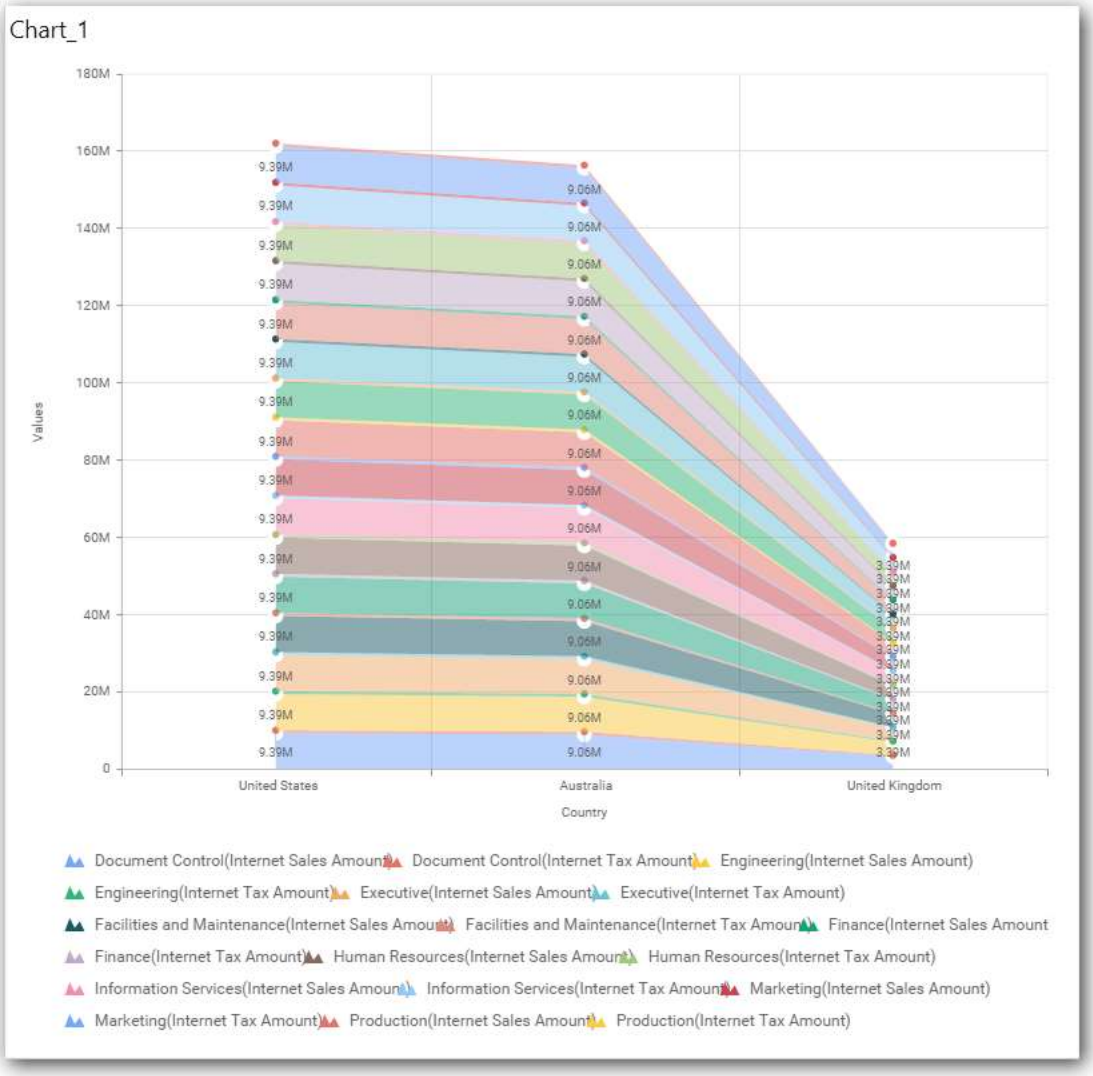


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.

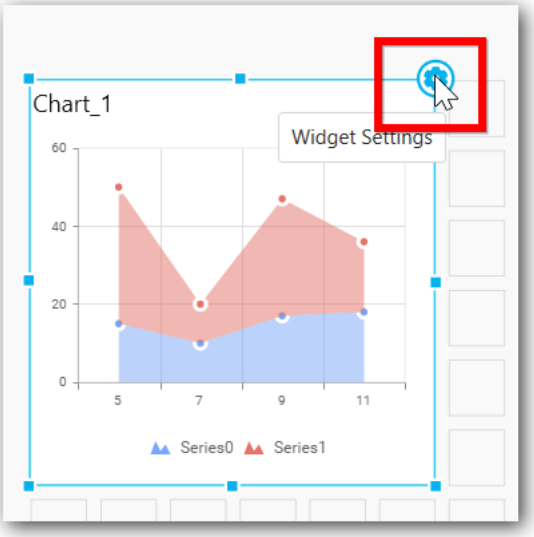


How to format Stacked Area Chart?

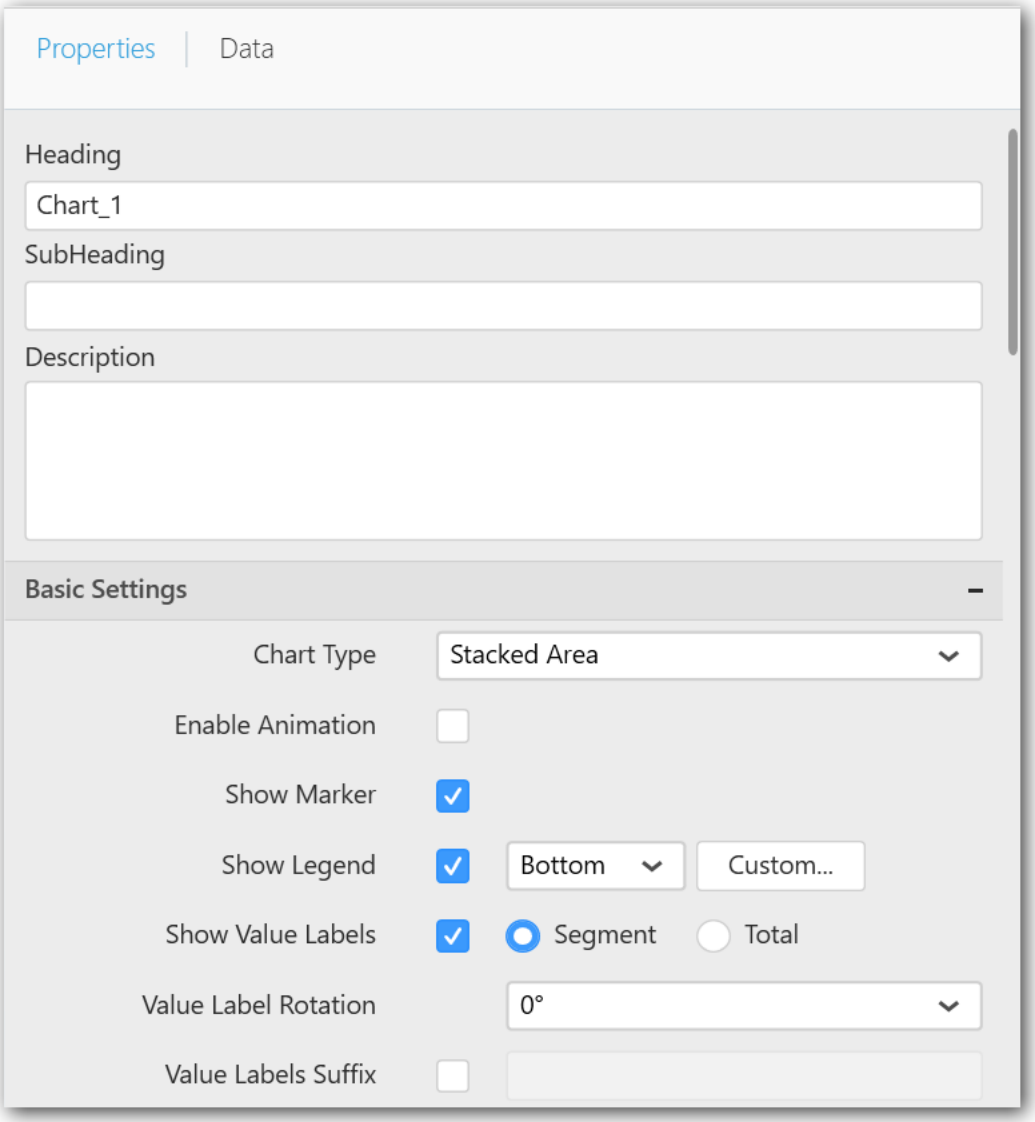
You can format the stacked area chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into bar chart follow the steps

1. Drag and drop the Pie chart into canvas and resize it to your required size.
2. Configure the data into bar chart.
3. Focus on the Pie chart and Click on Widget Settings.

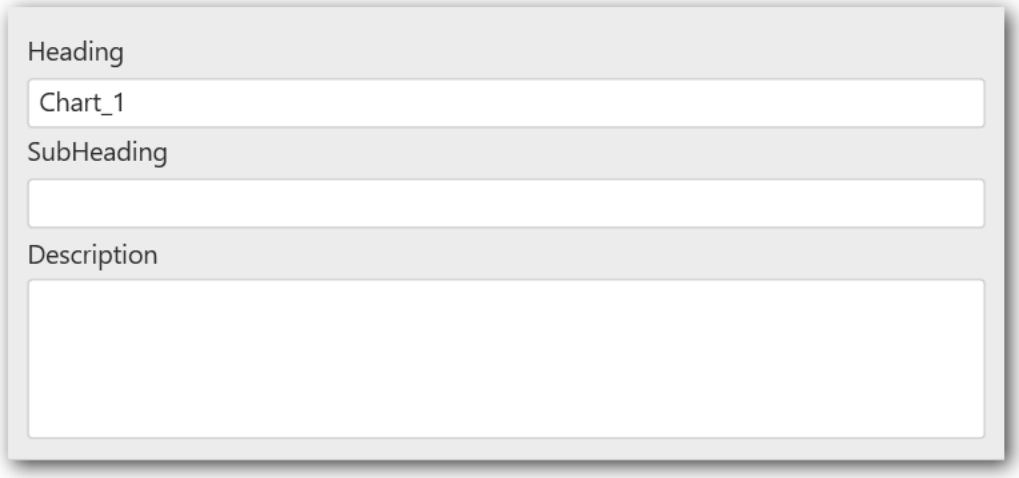


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this stacked area chart widget.

SubHeading

This allows you to set sub-title for this line chart widget.

Description

This allows you to set description for this stacked area chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

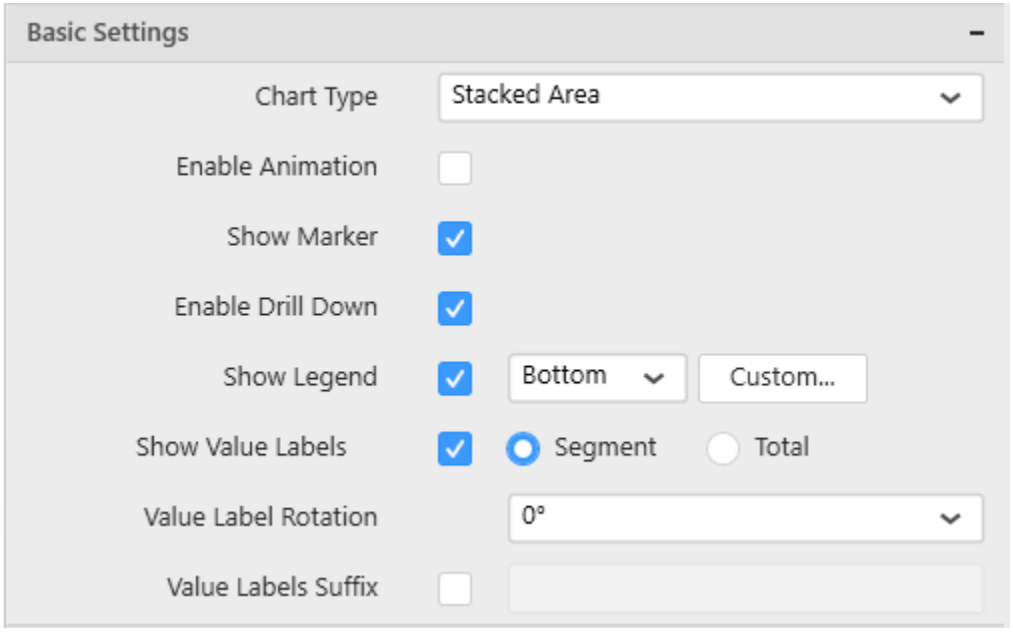


Chart Type

This allows you to switch the widget view from current chart type to another chart type. To selecting chart type through combo box.

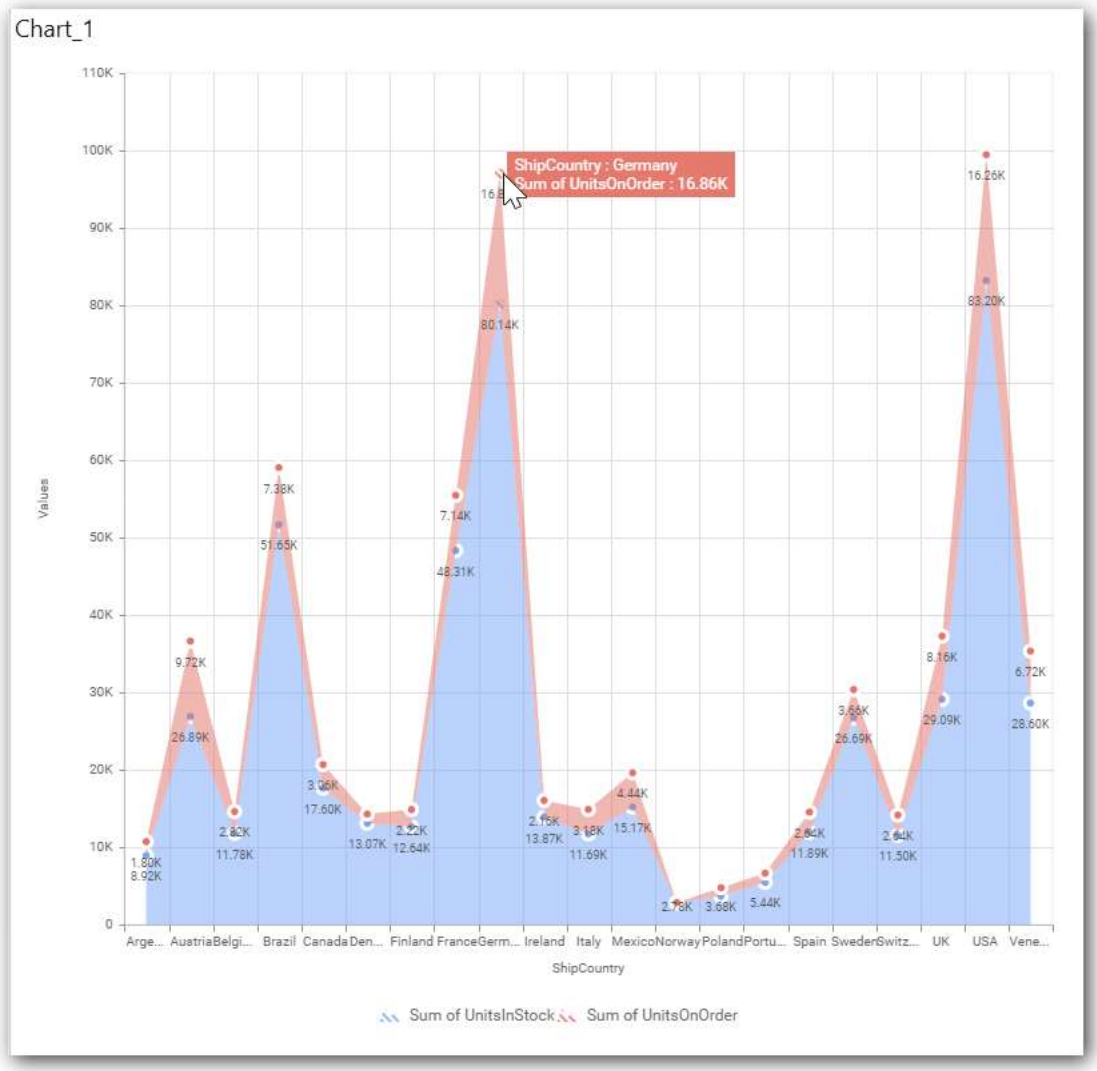
Enable Animation

This allows you to enable the series rendering in animated mode.

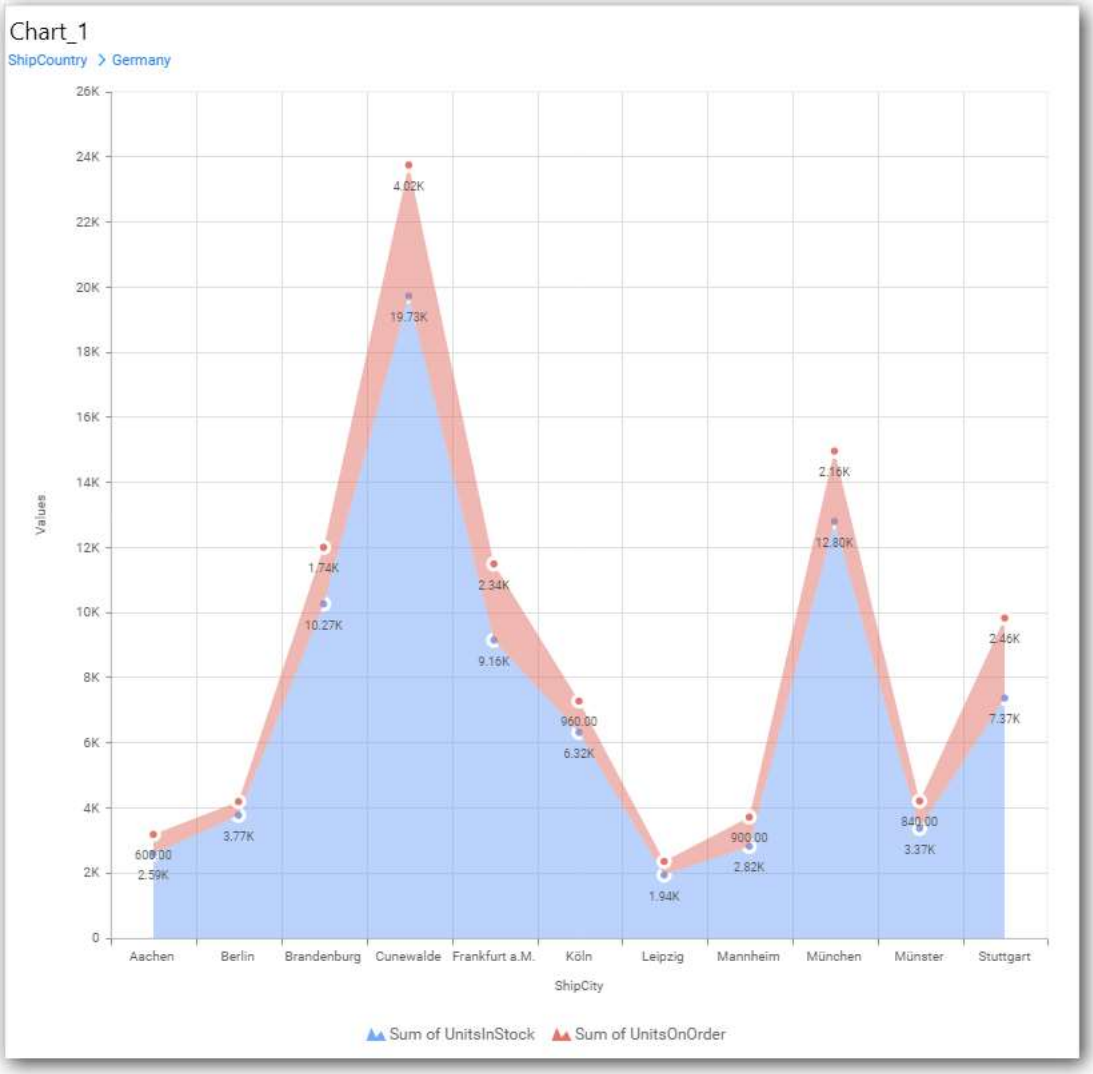
Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View

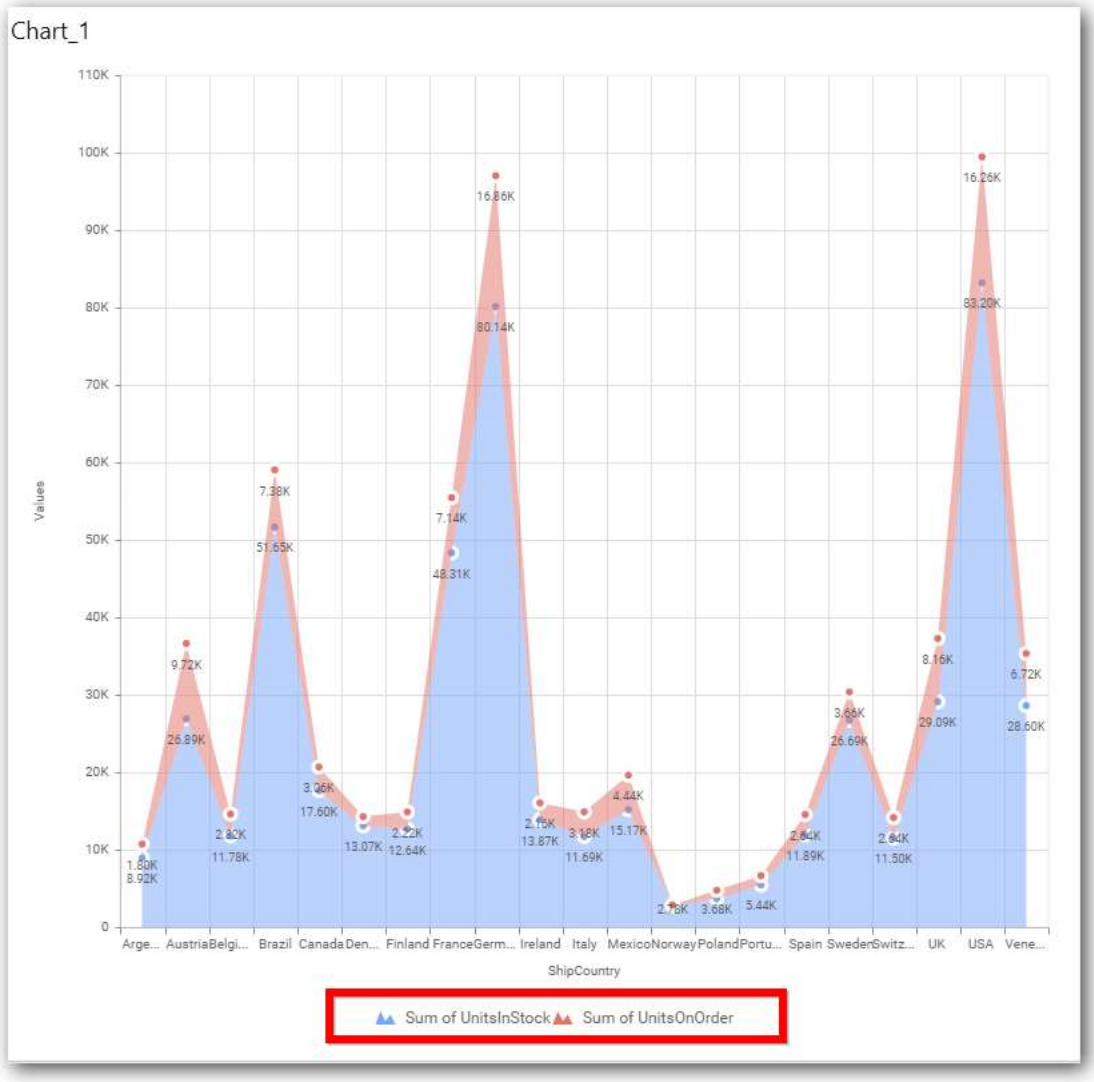


Drilled View



Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling the option Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

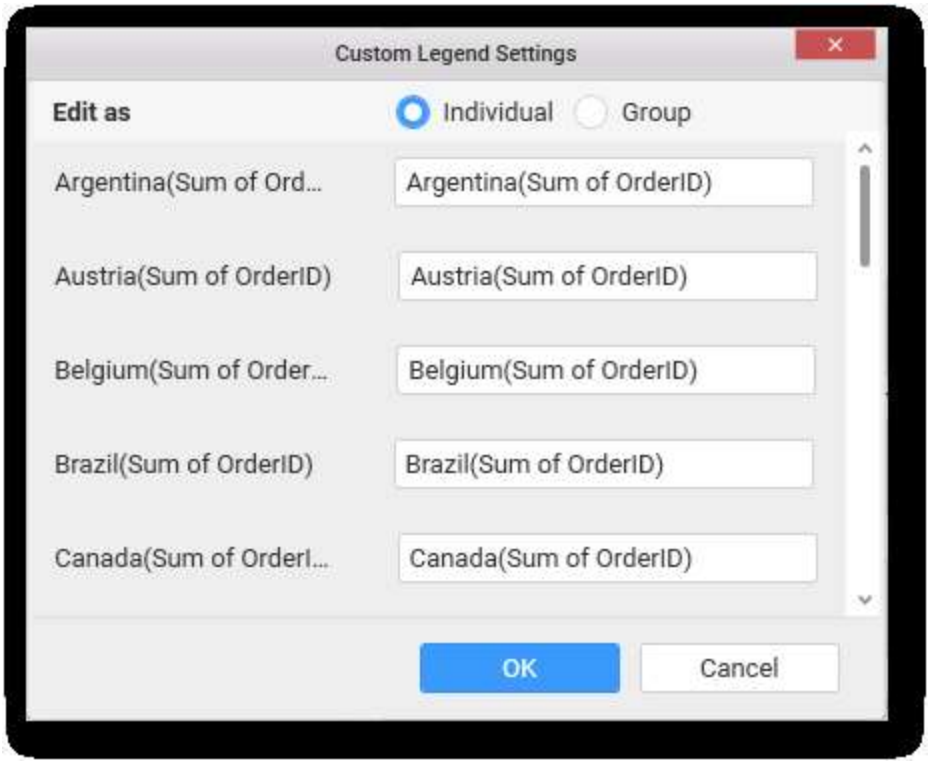
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

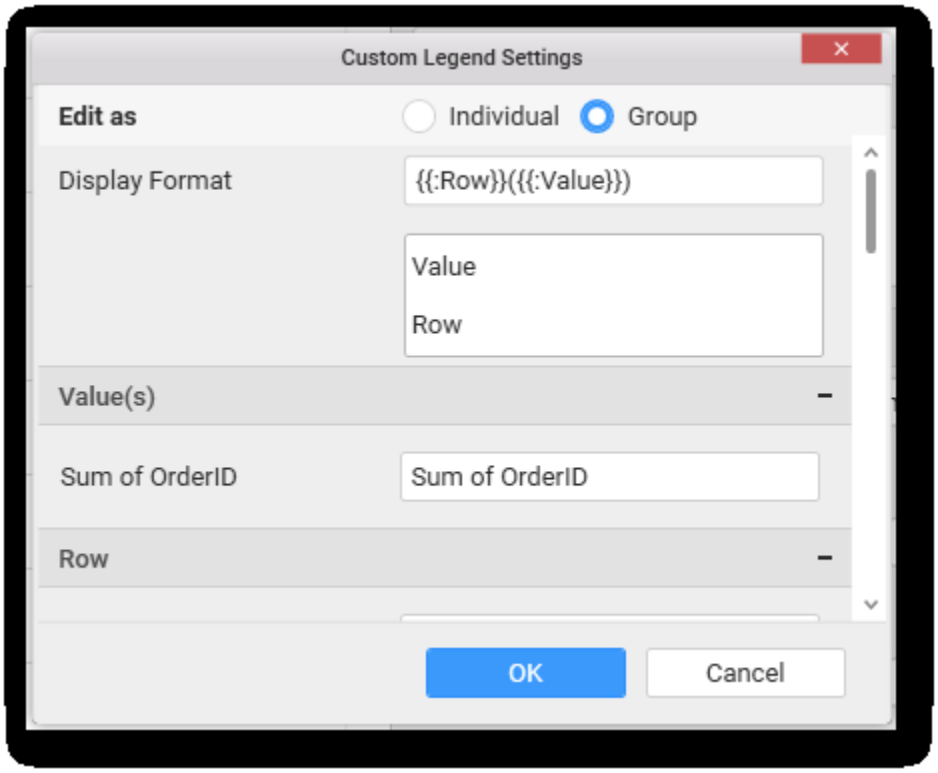
{{"{"}} : Row {{}}}} ({{"{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

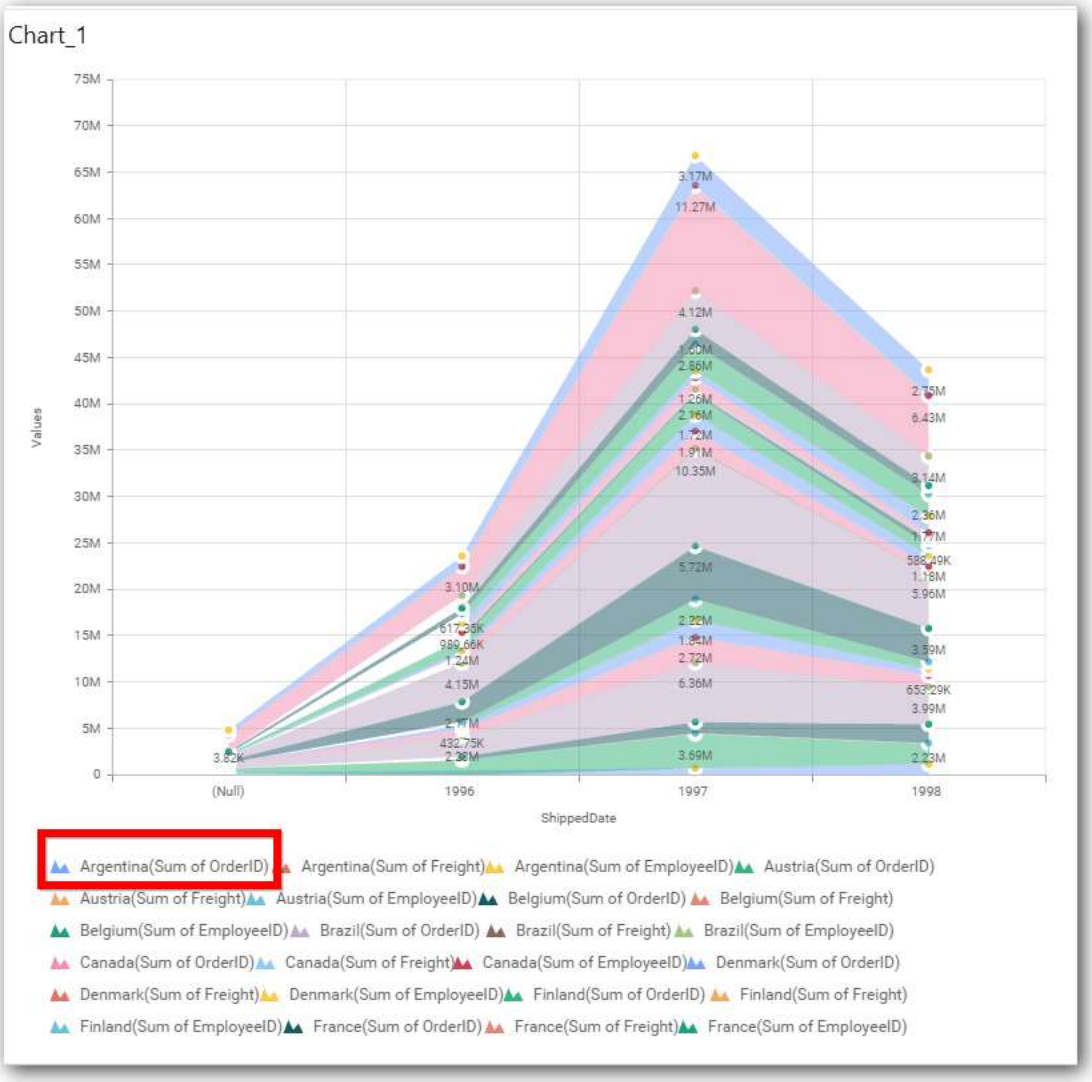


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.



For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



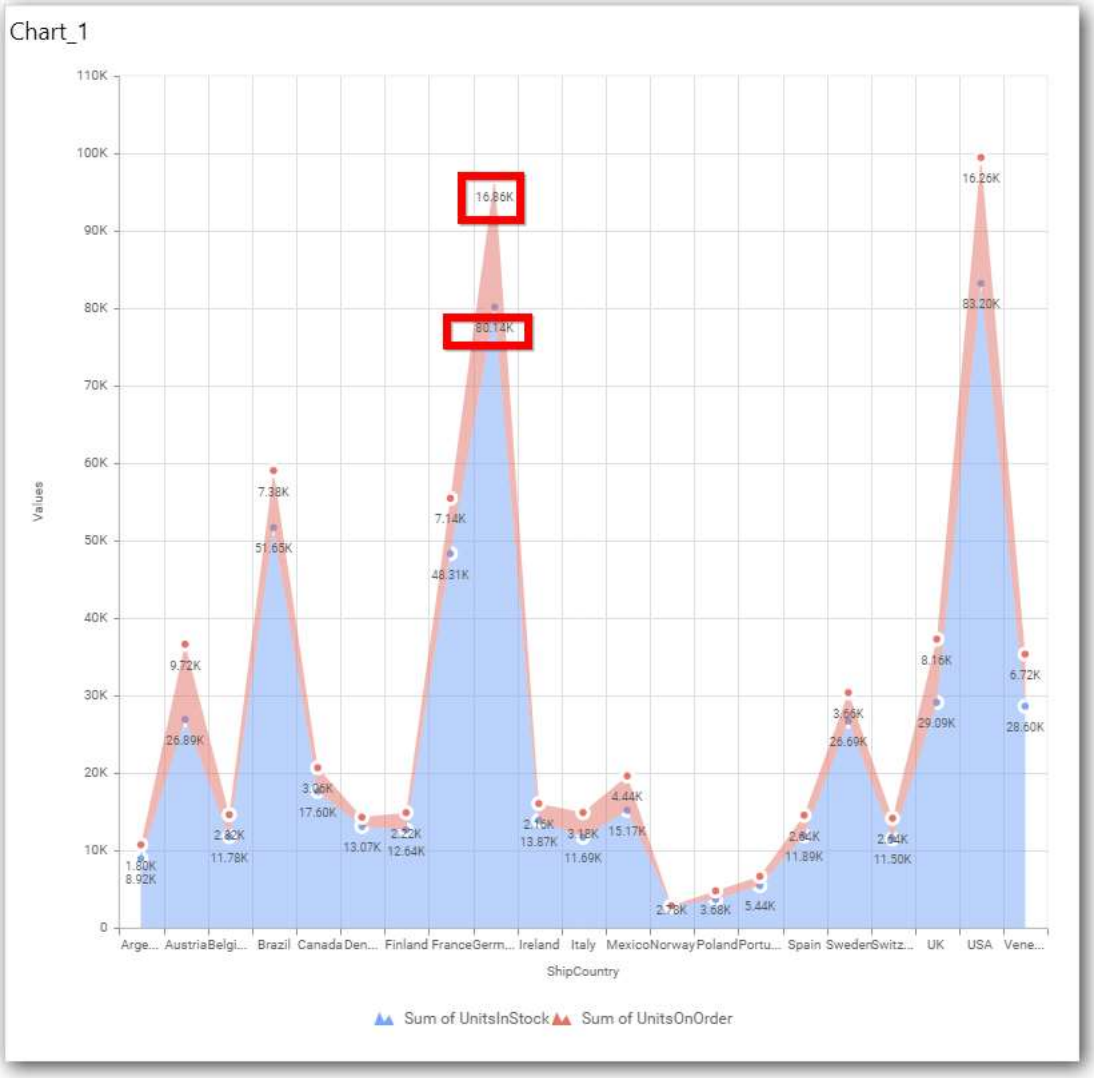
Show Value Labels

This allows you to toggle the visibility of value labels. When you toggle on, two options will be provided to change the display mode of the labels.

1. Segment
2. Total

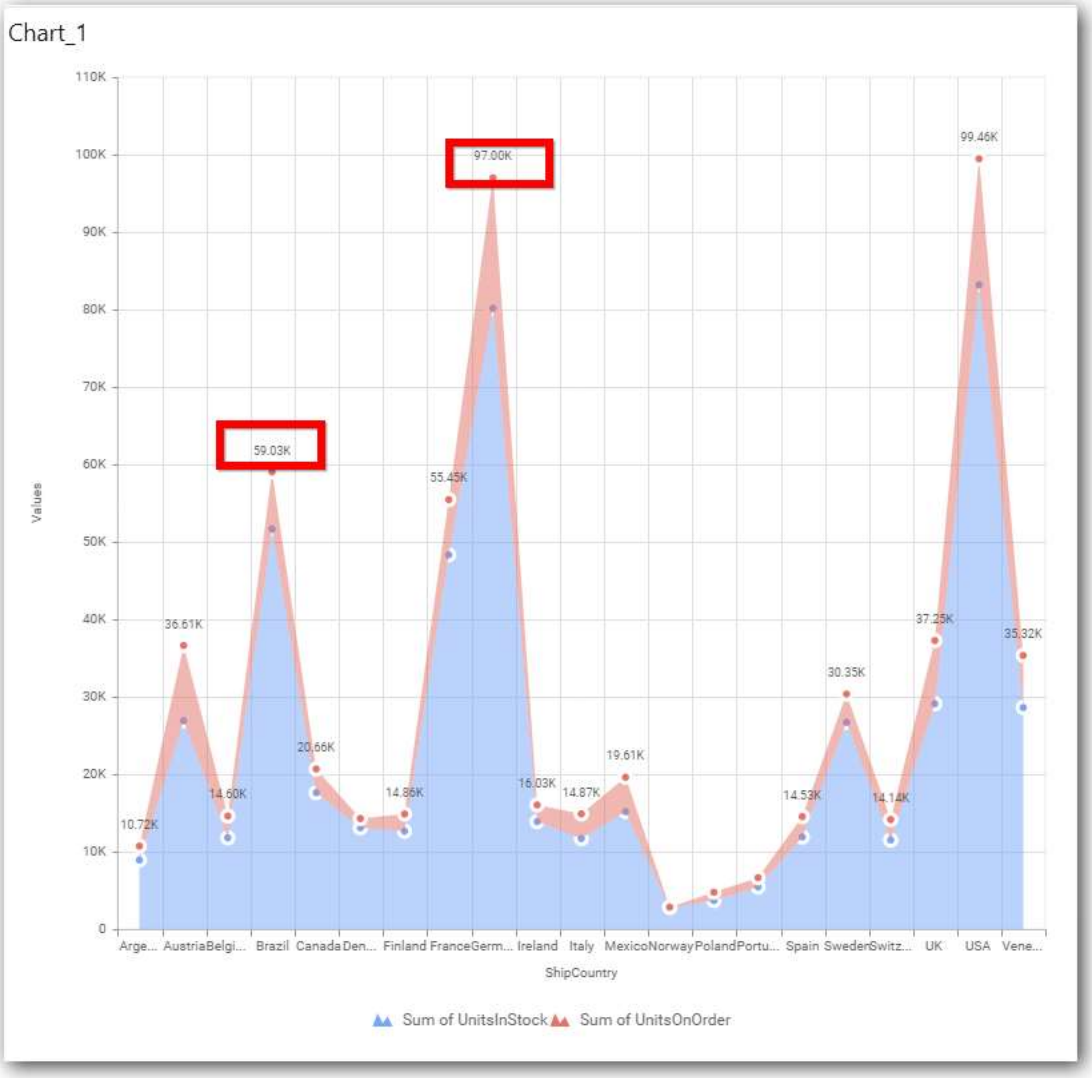
Segment

Displays value label for each series segment.



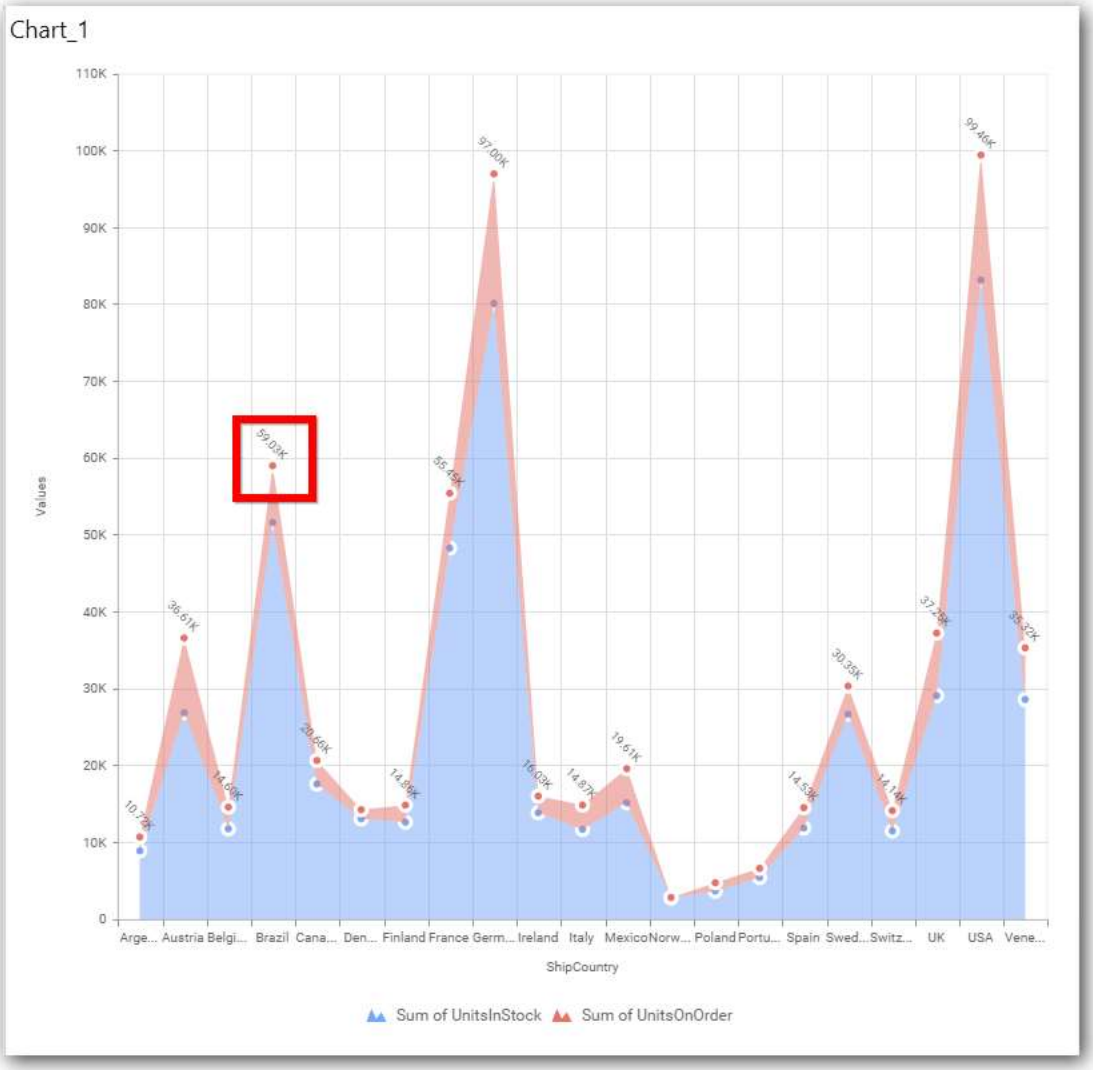
Total

Displays sum value label of all the stacked segment on the top most segment.



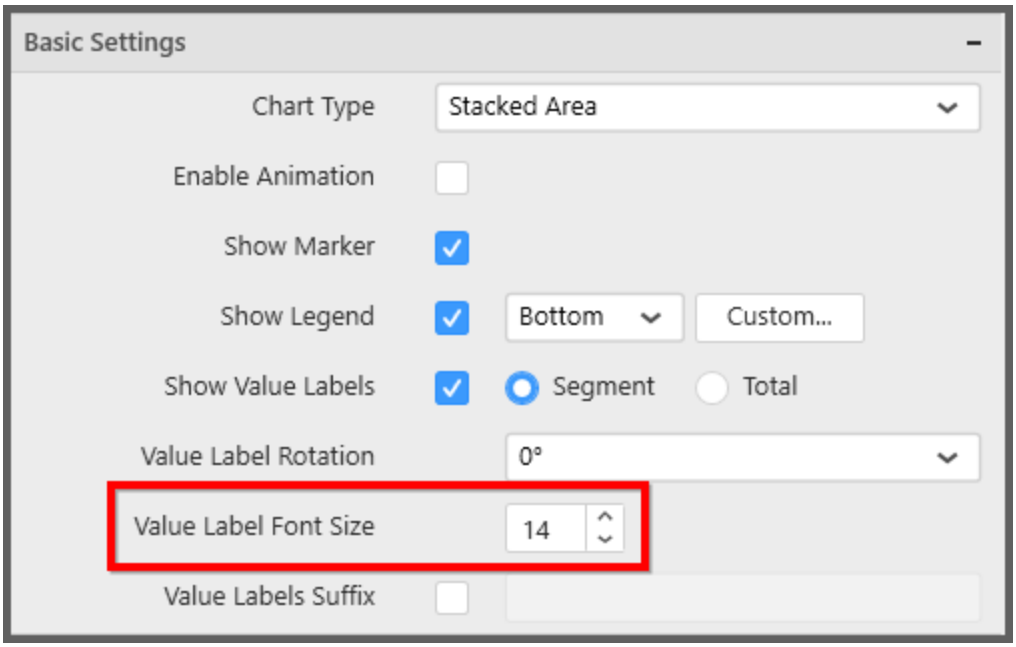
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



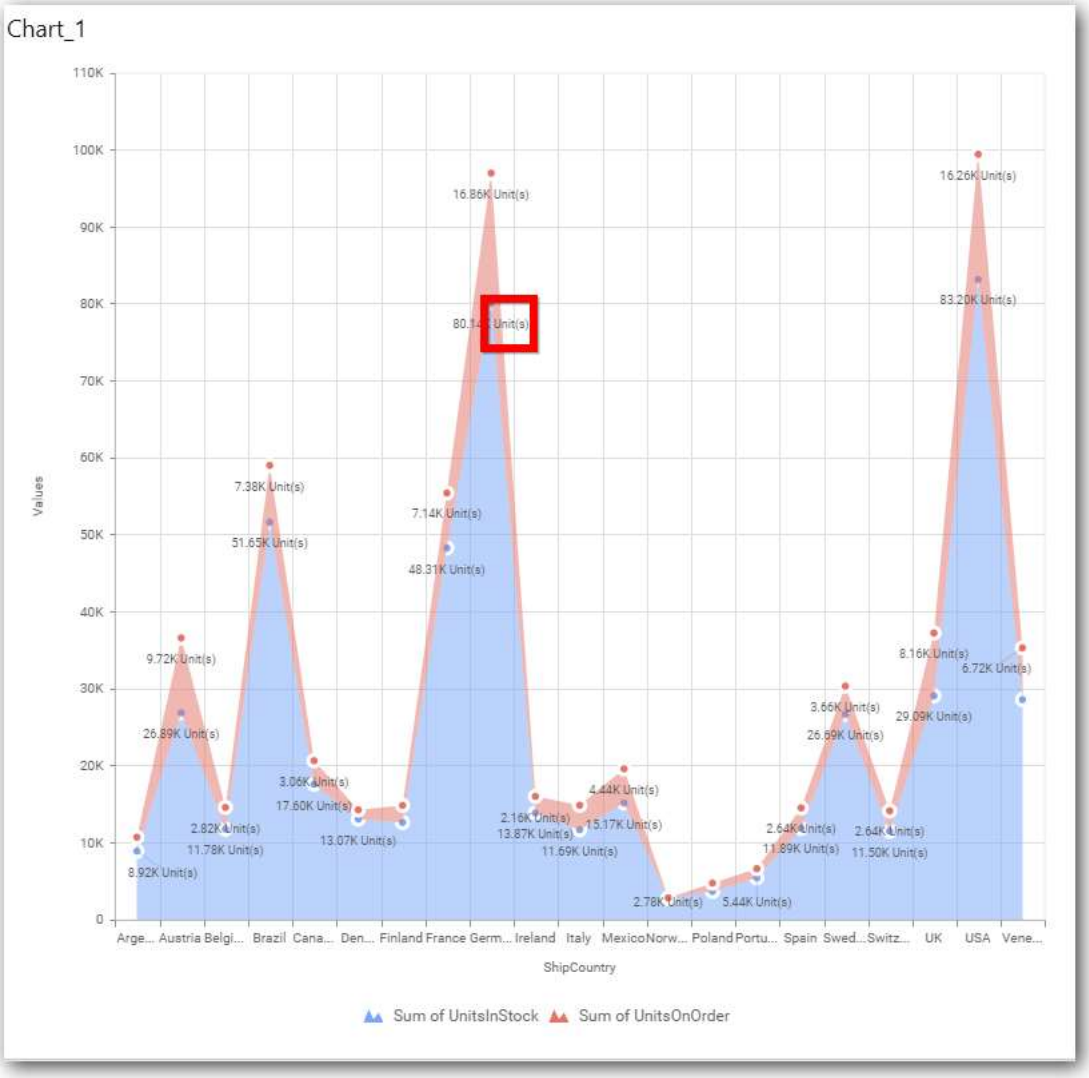
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.



Value Labels Suffix

Allows you to set suffix to the value labels.



Filter Settings

Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

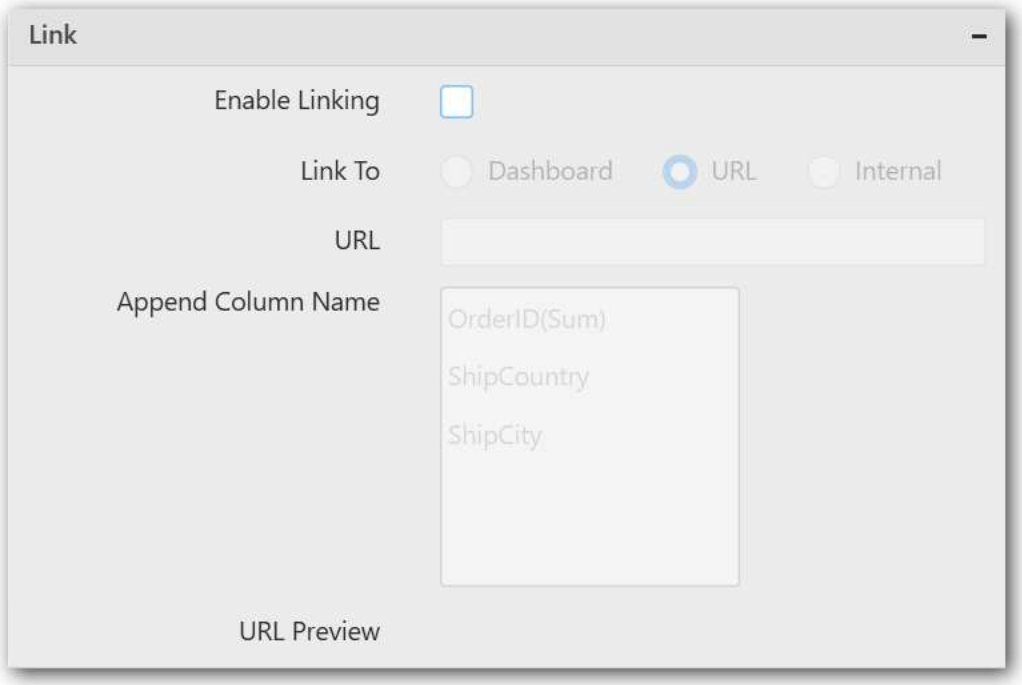
Act as Master Widget

This allows you to define this stacked area chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

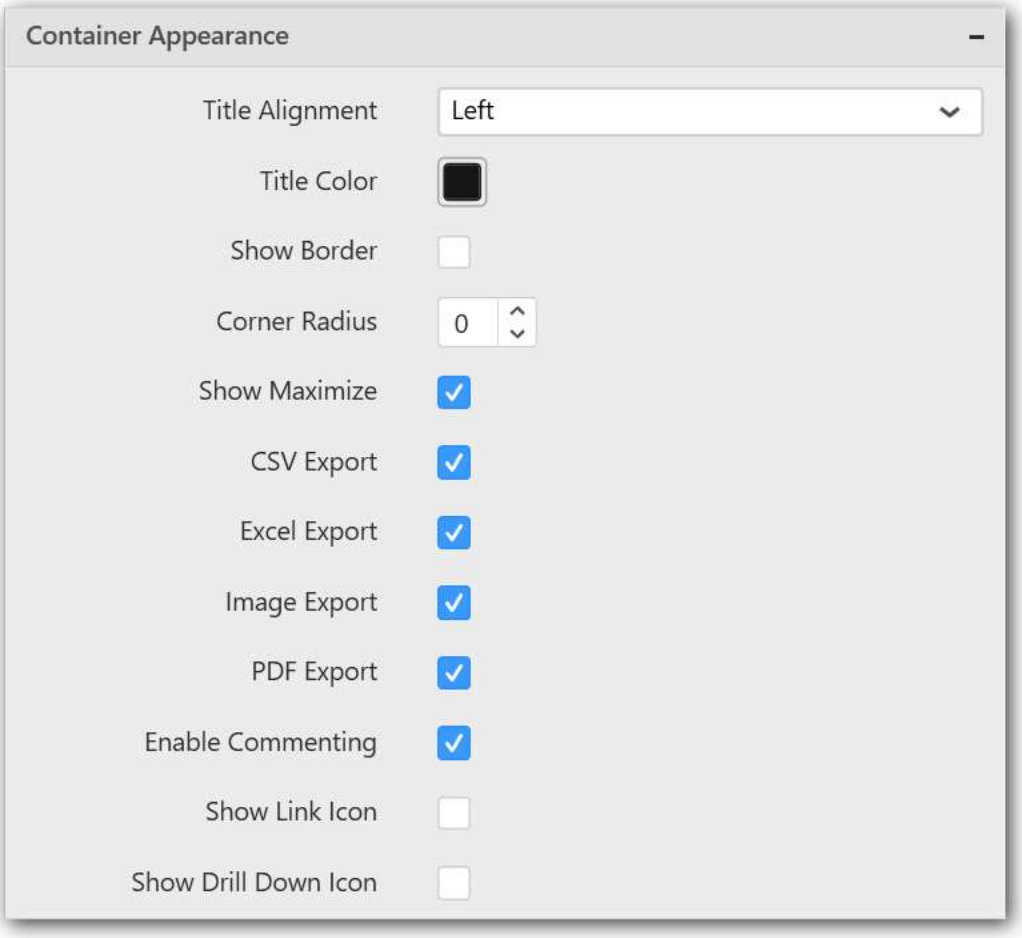
This allows you to define this stacked area chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this stacked area chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this stacked area chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this stacked area chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

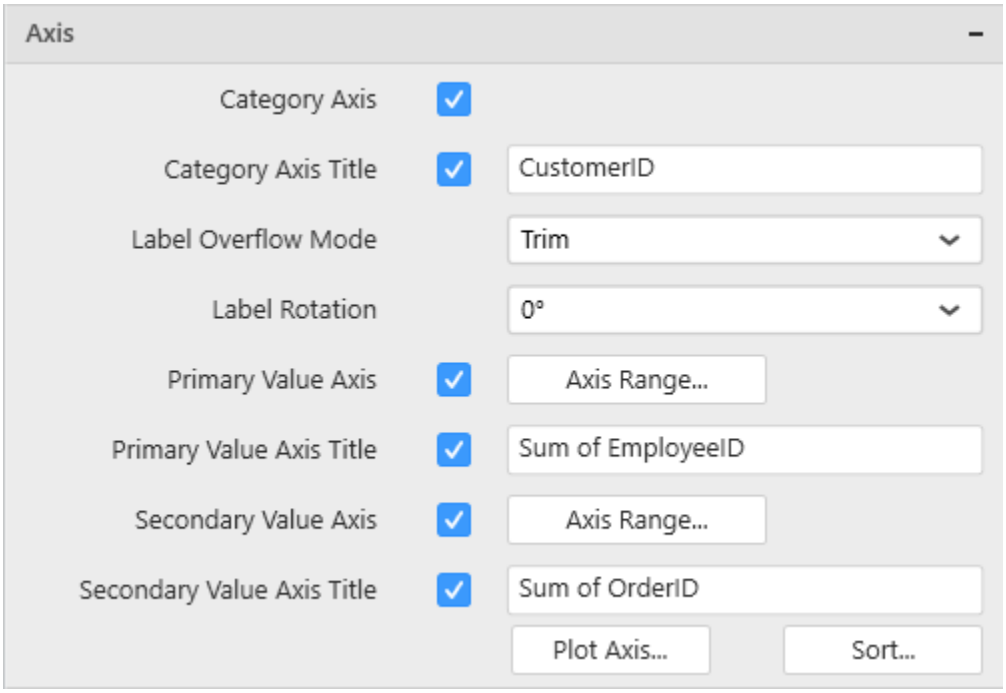
Image Export

This allows you to enable/disable the image export option for this stacked area chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

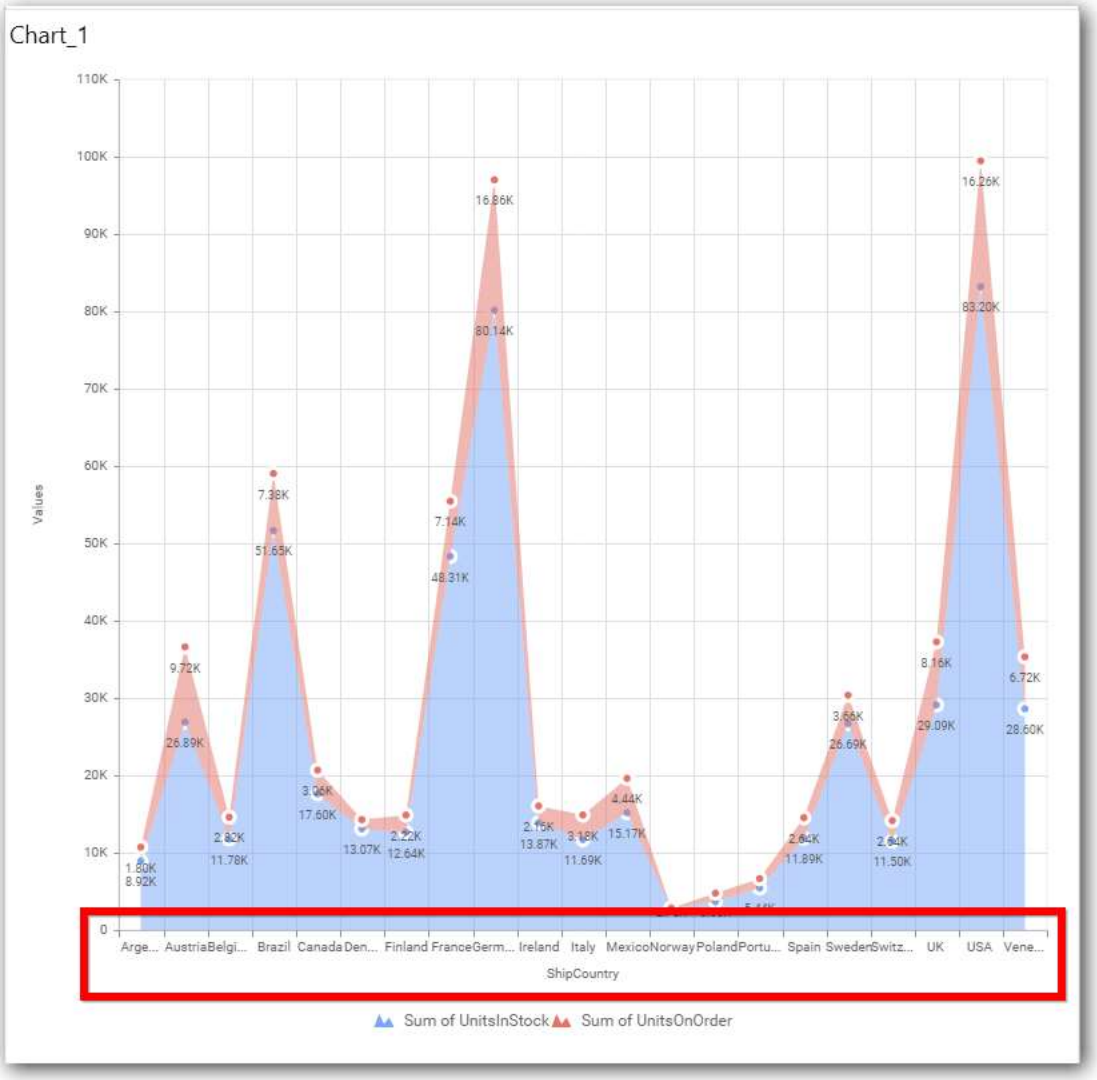
Axis Settings



This section allows you to customize the axis settings in chart.

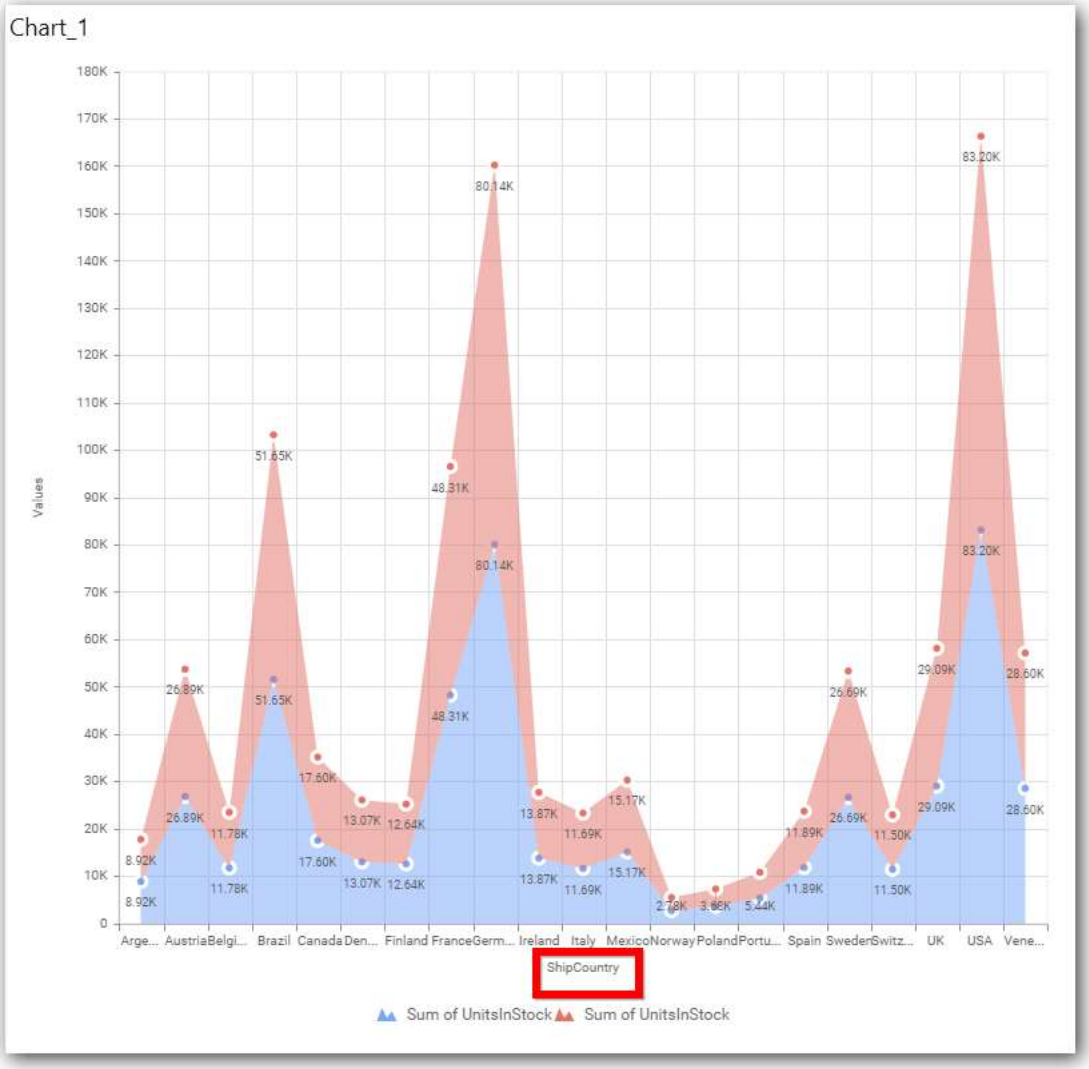
Category Axis

This allows you to enable/edit the Category Axis title. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

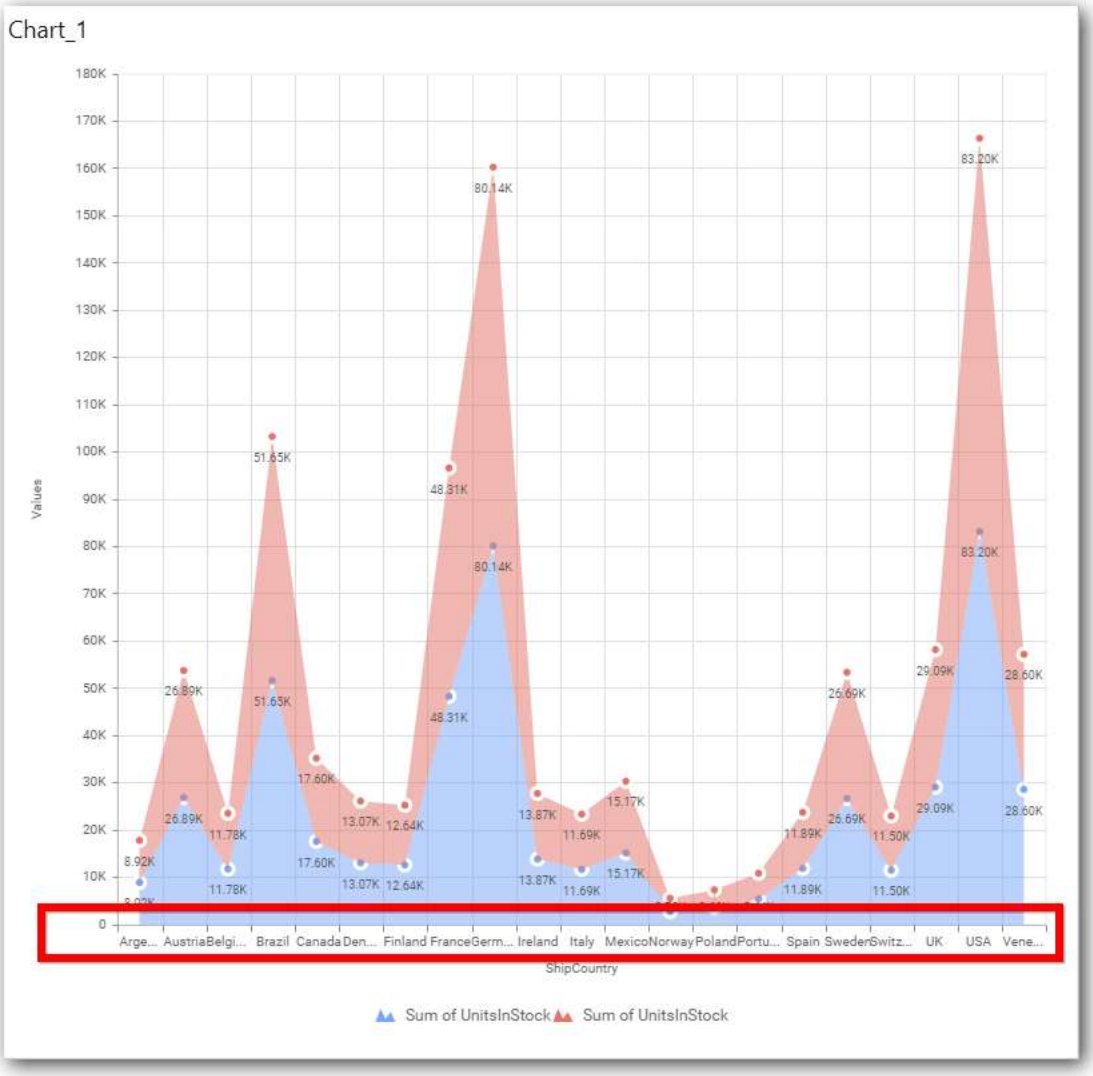


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

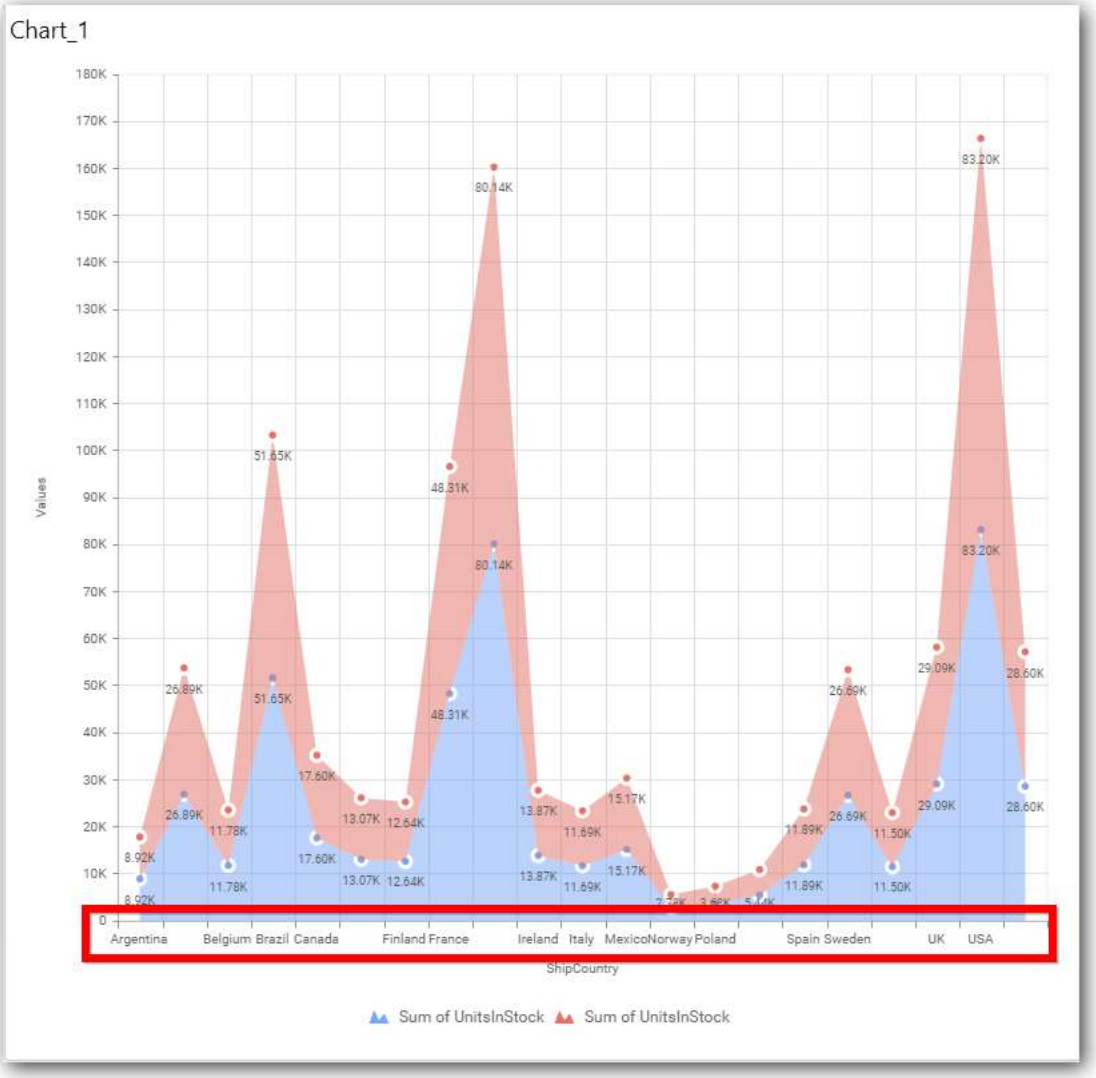
Trim

This option trims the end of overlapping label in the axis.



Hide

This option hides the overlapping label in the axis.

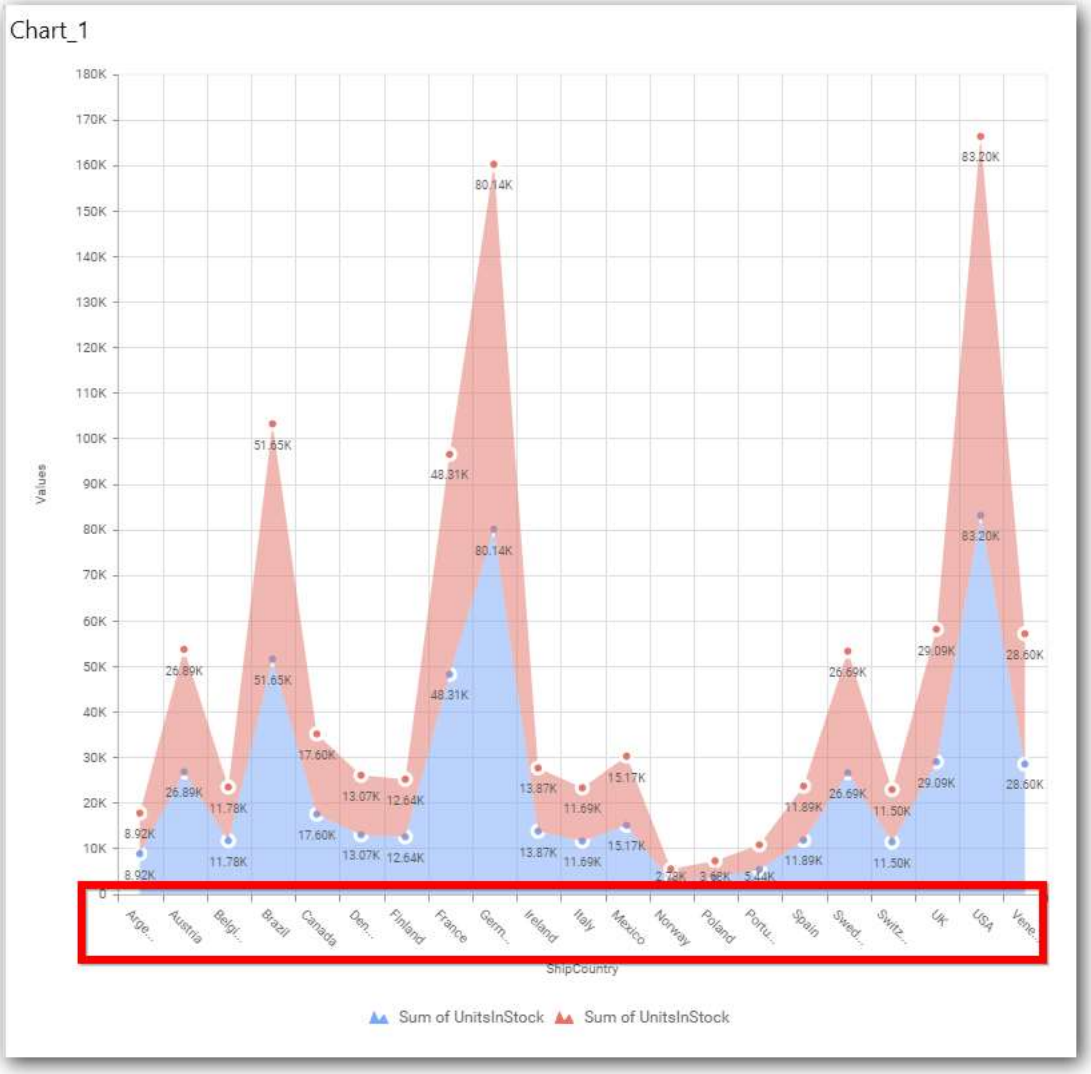


Wrap

This option wraps the lengthy label text in the axis.

Label Rotation

This allows you to define the rotation angle for the category axis labels to display.

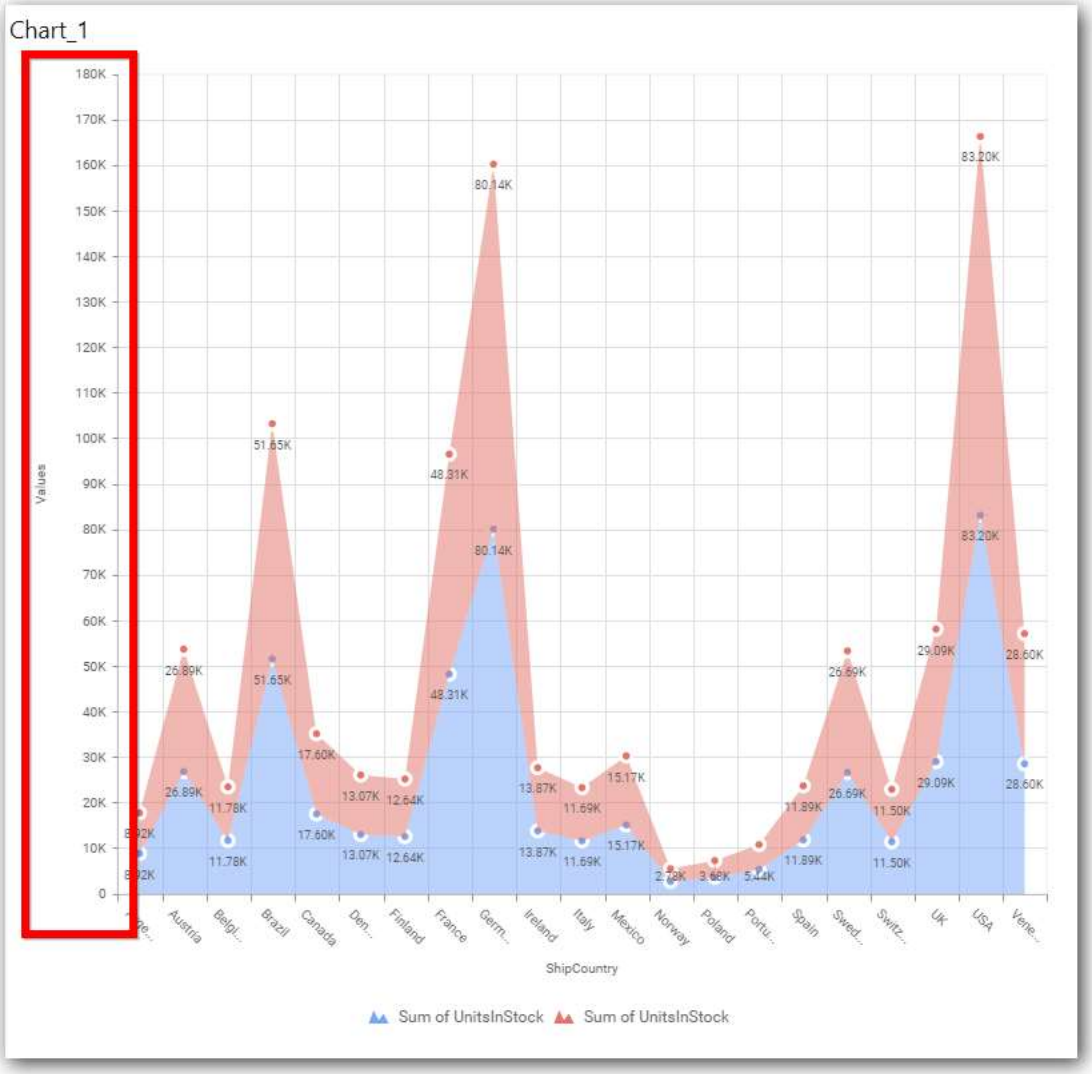


Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.

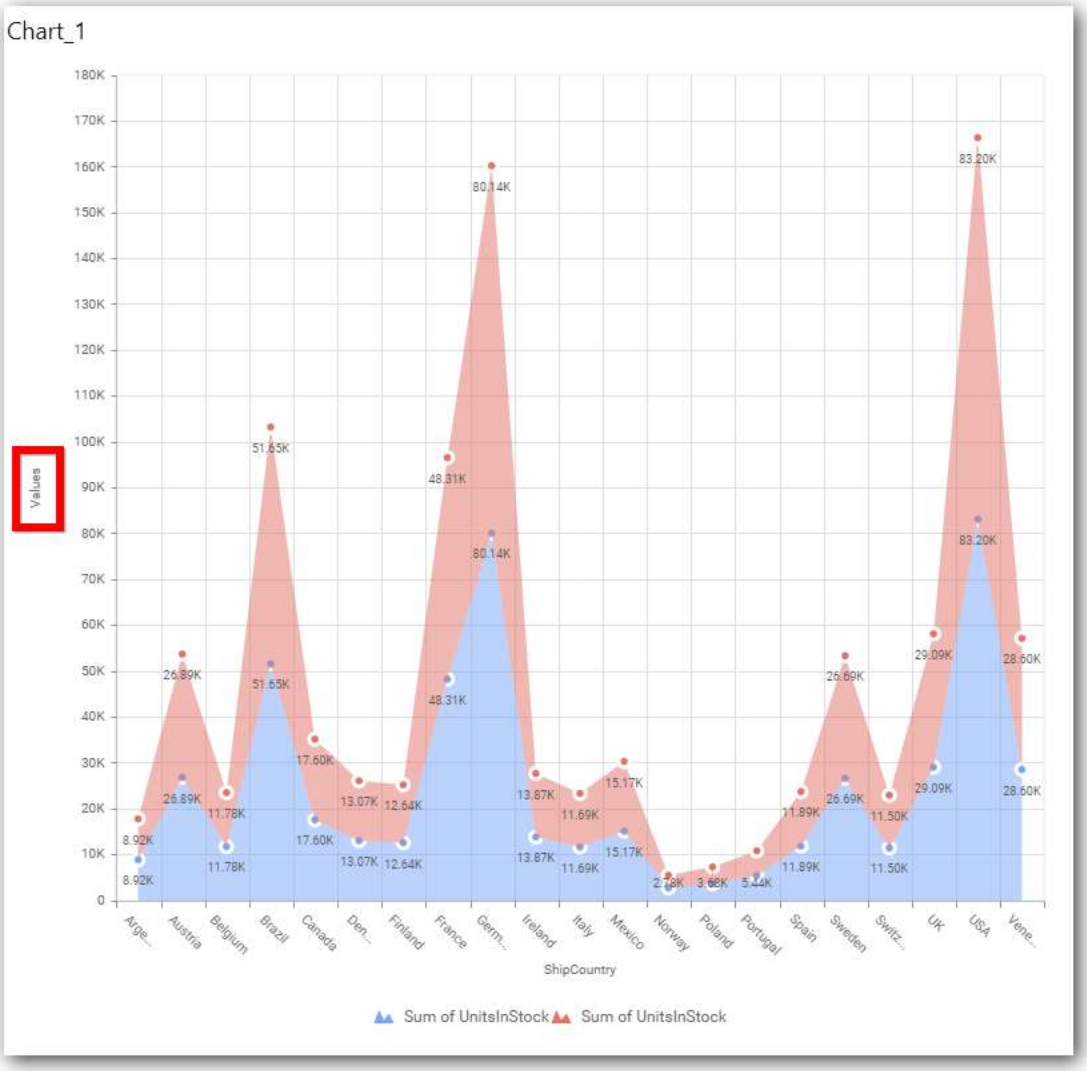
Primary Value Axis

This allows you to enable/edit the Primary Value Axis title. It will reflect in chart area y-axis name.



Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

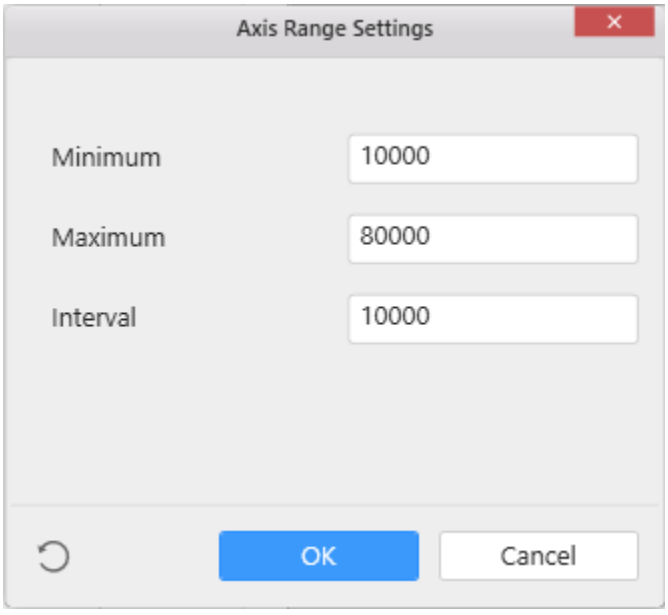
This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

Primary Value Axis Range

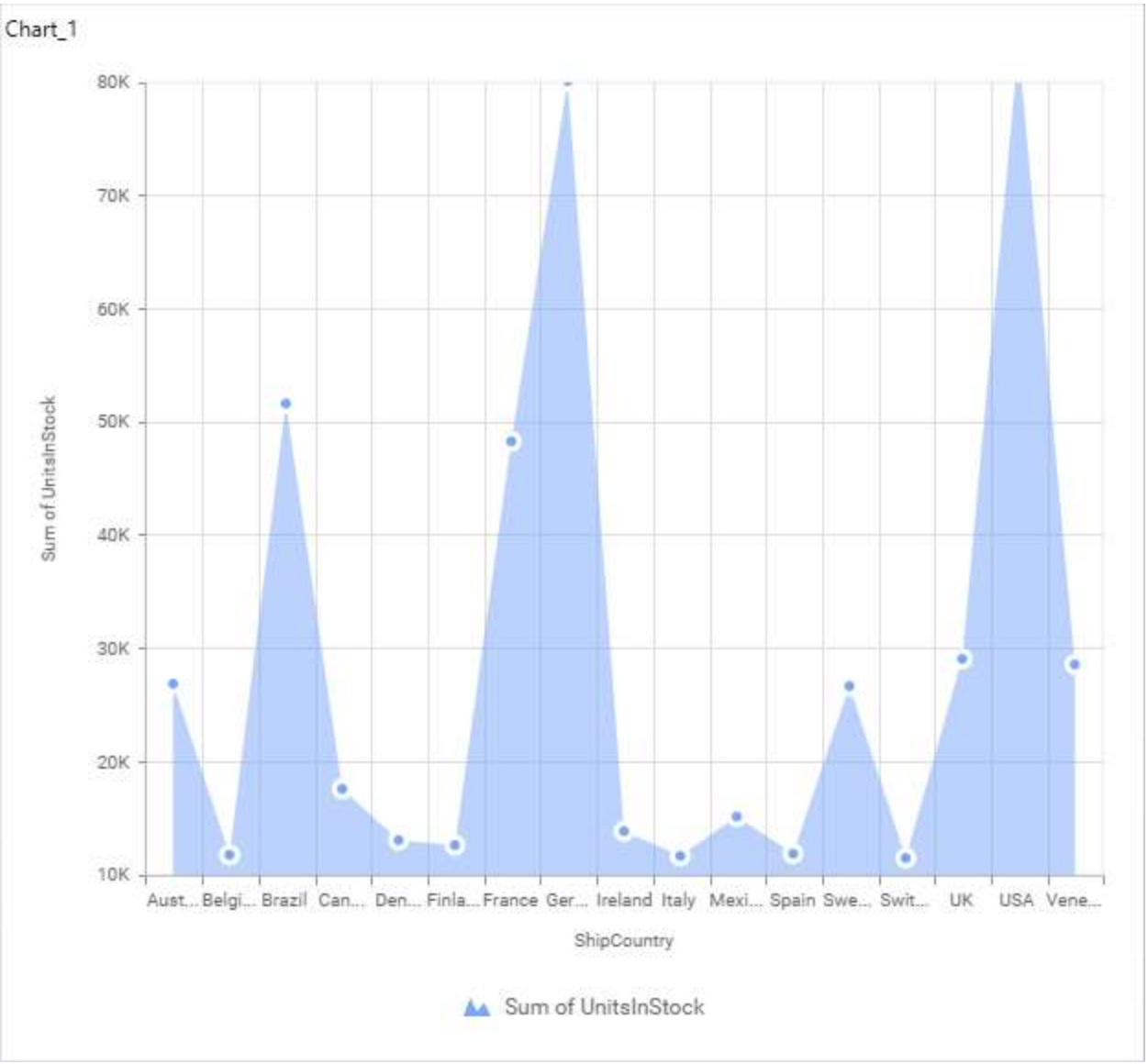
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

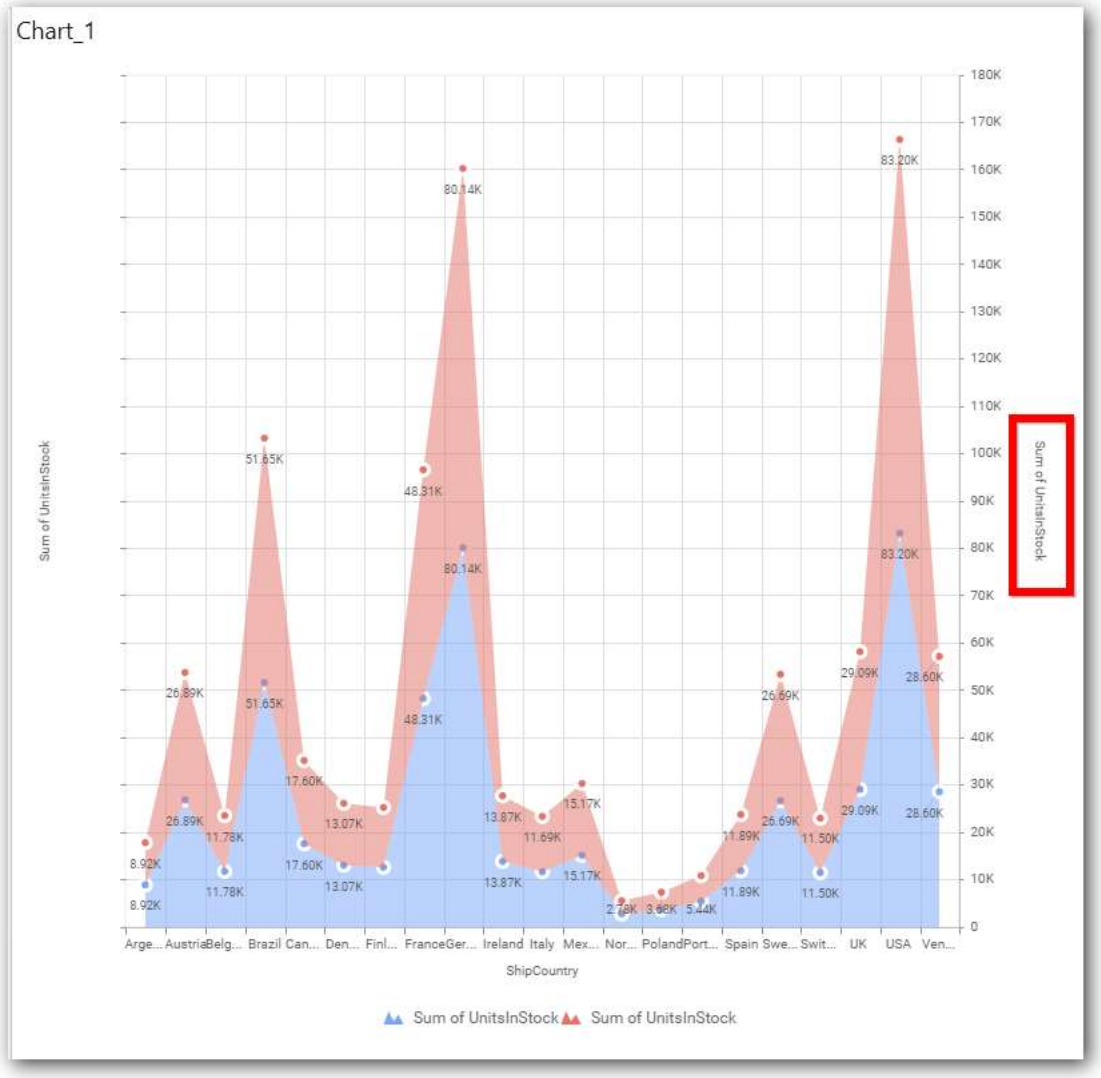


Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.

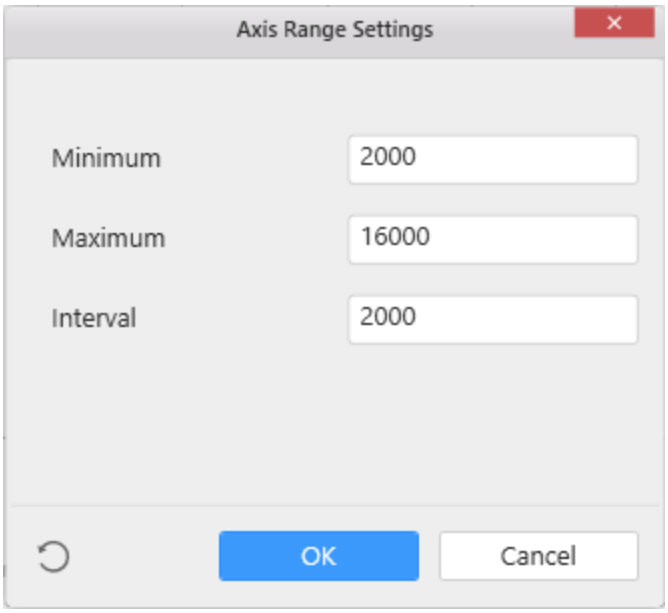
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

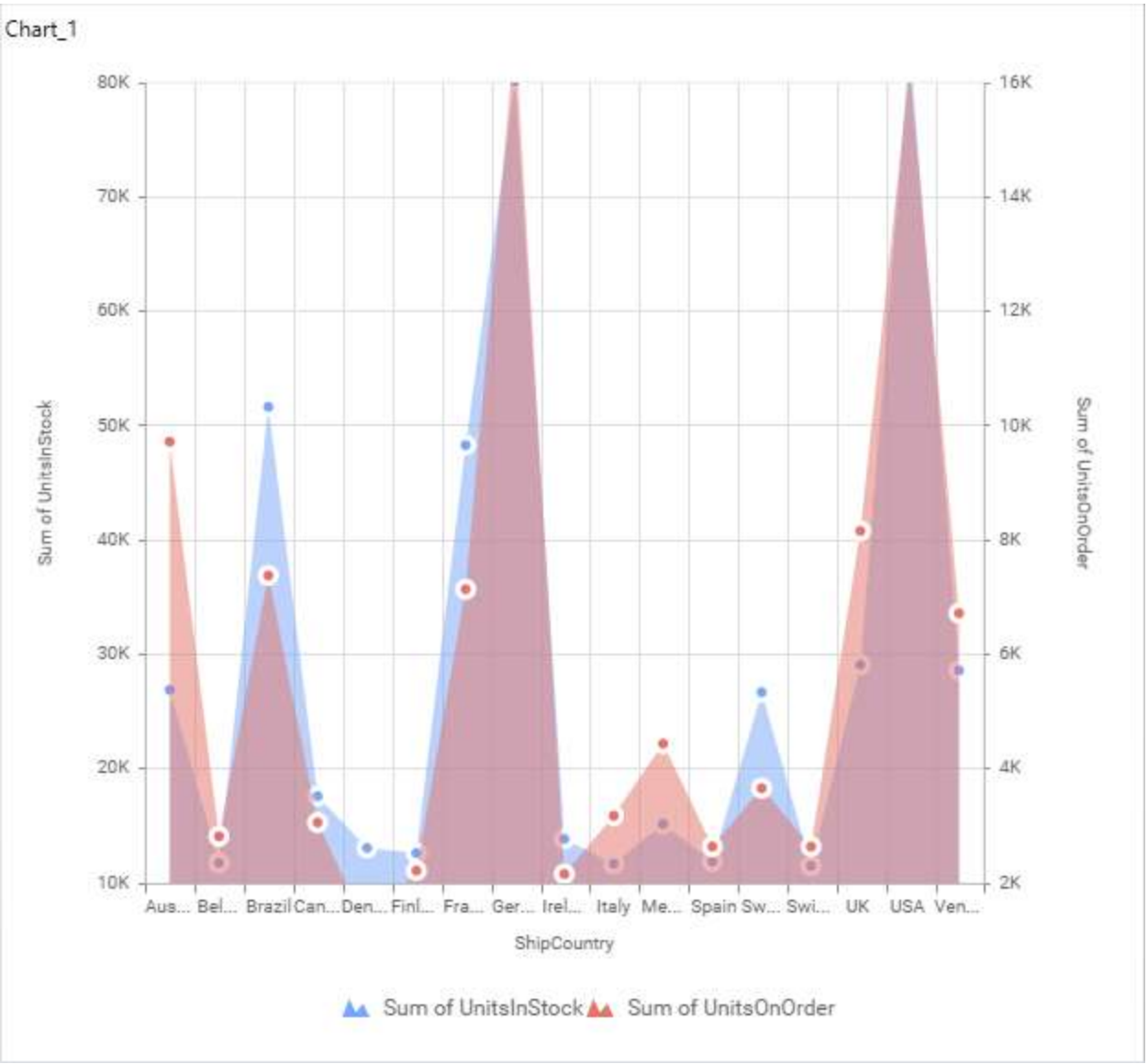


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

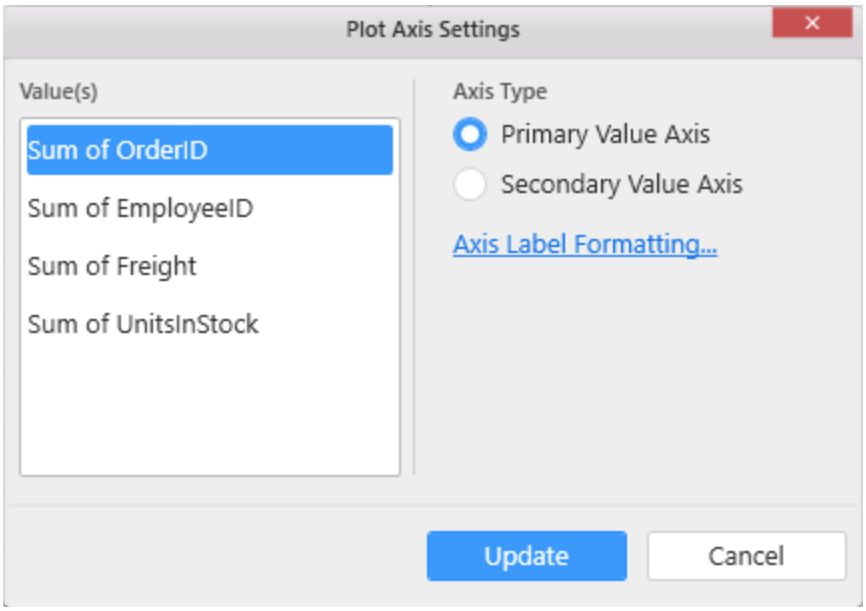


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



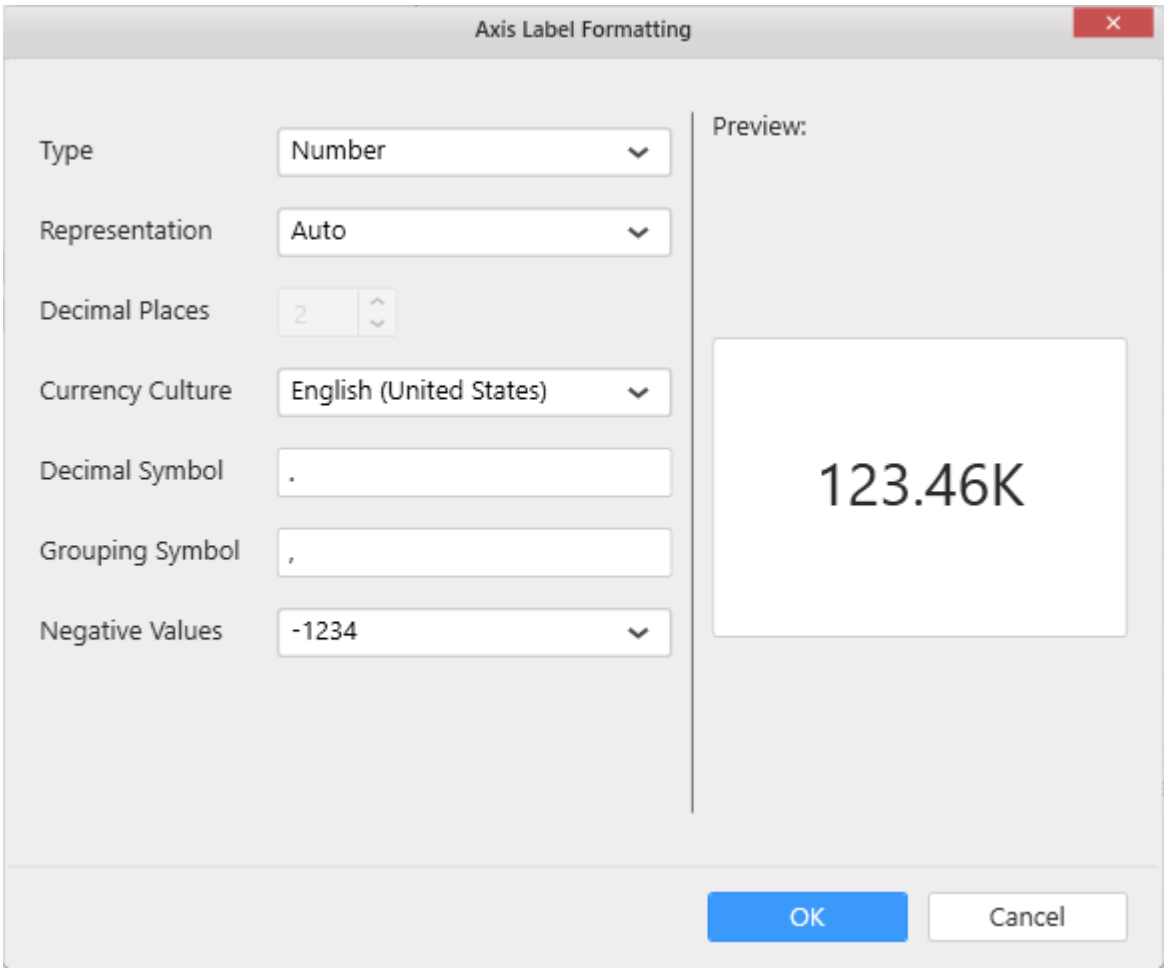
Plot Axis Settings

This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



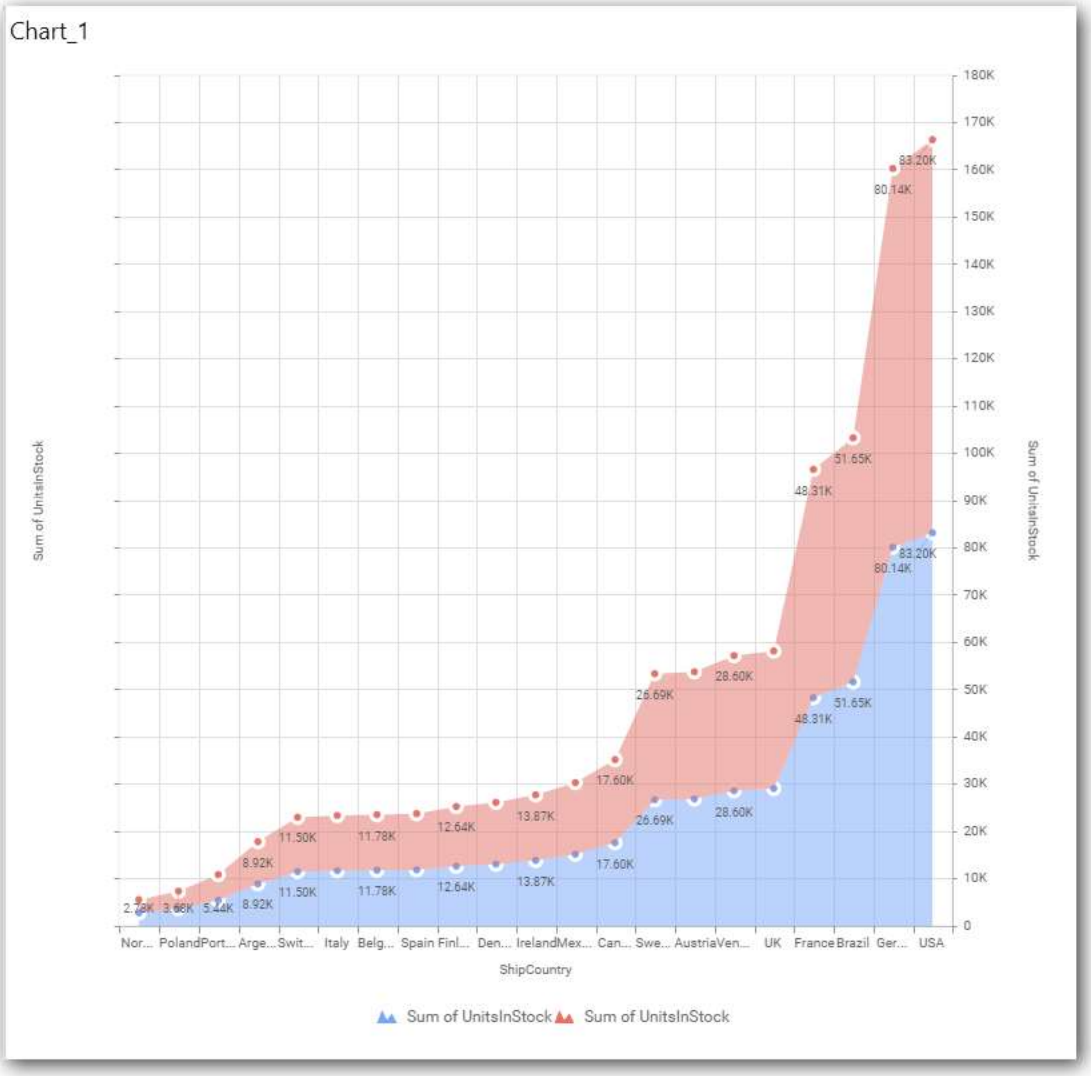
Axis Label Formatting

This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.



Sort Order

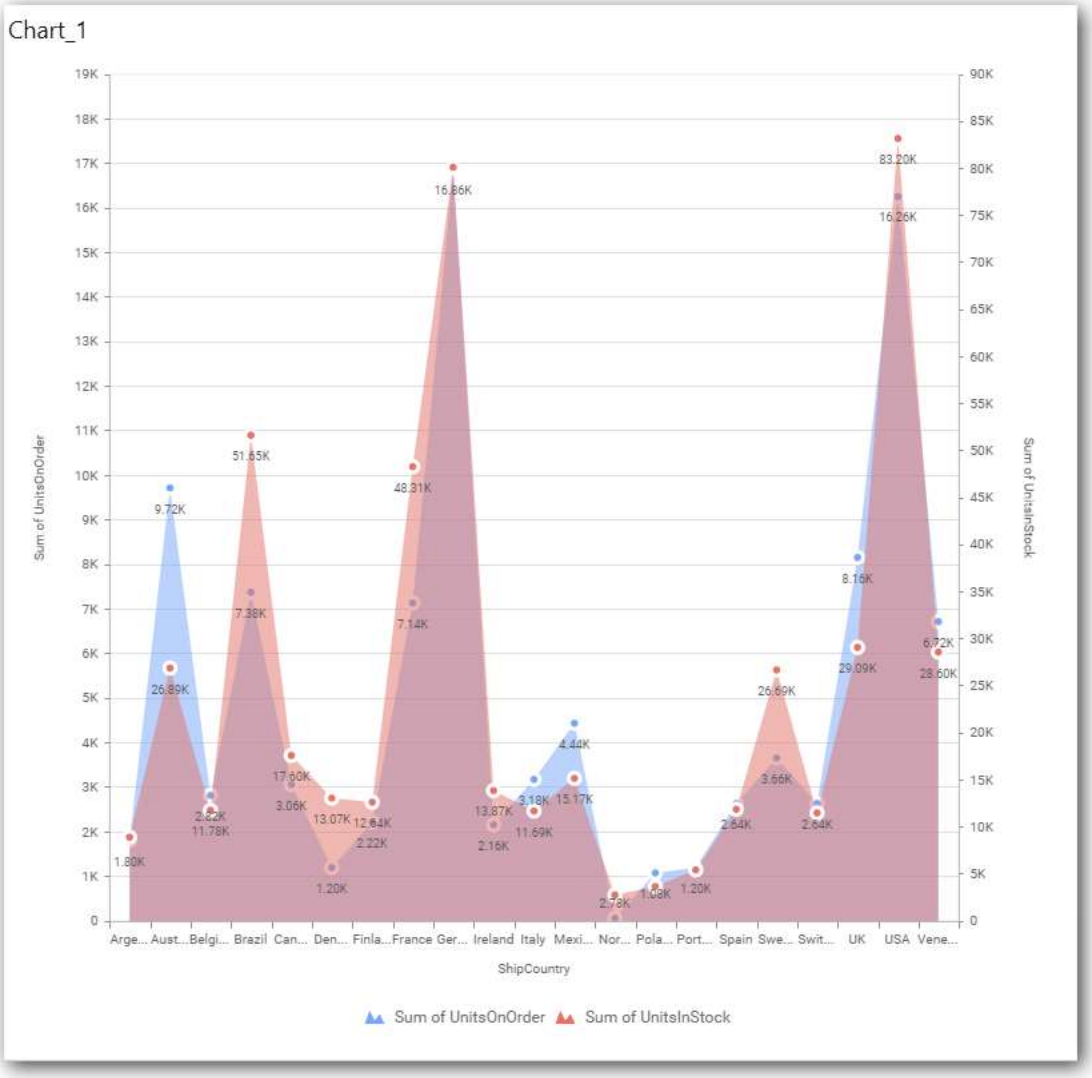
This allows you to define the sort order for each measure column added.



Grid Line Settings

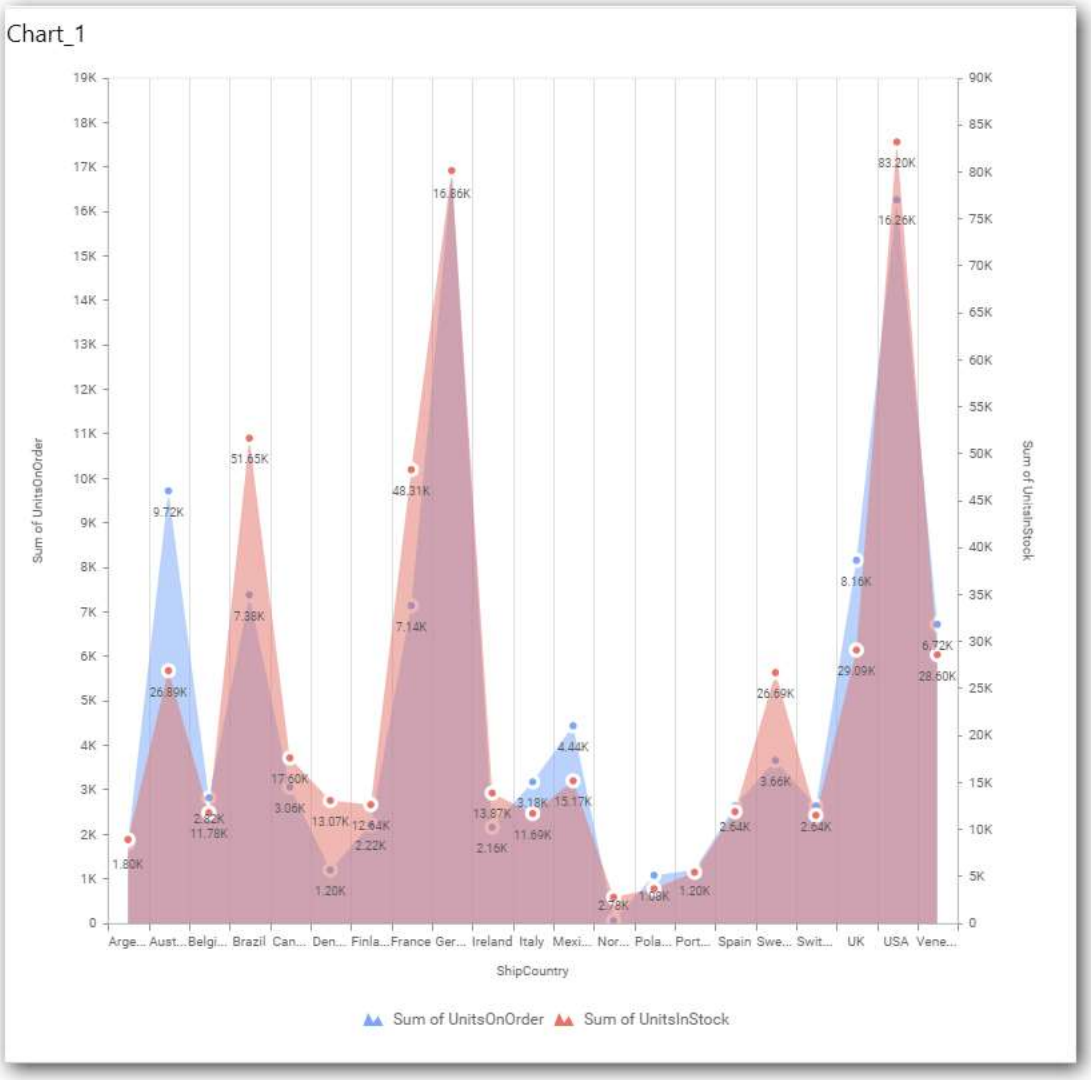
Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the stacked area chart.



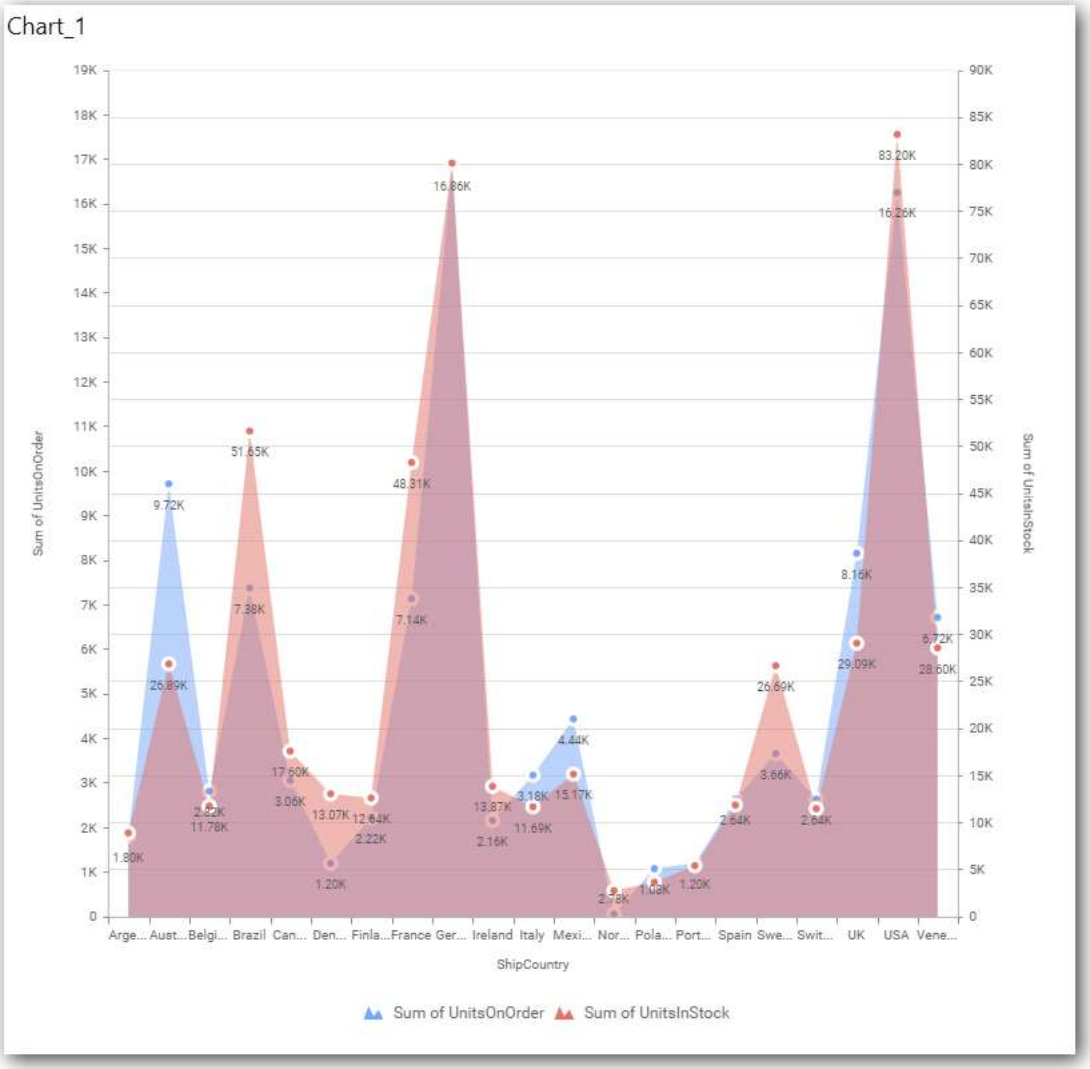
Category Axis

This allows you to define the visibility of Category axis gridlines.



Secondary Value Axis

This allows you to toggle the visibility of Secondary Value Axis gridlines.

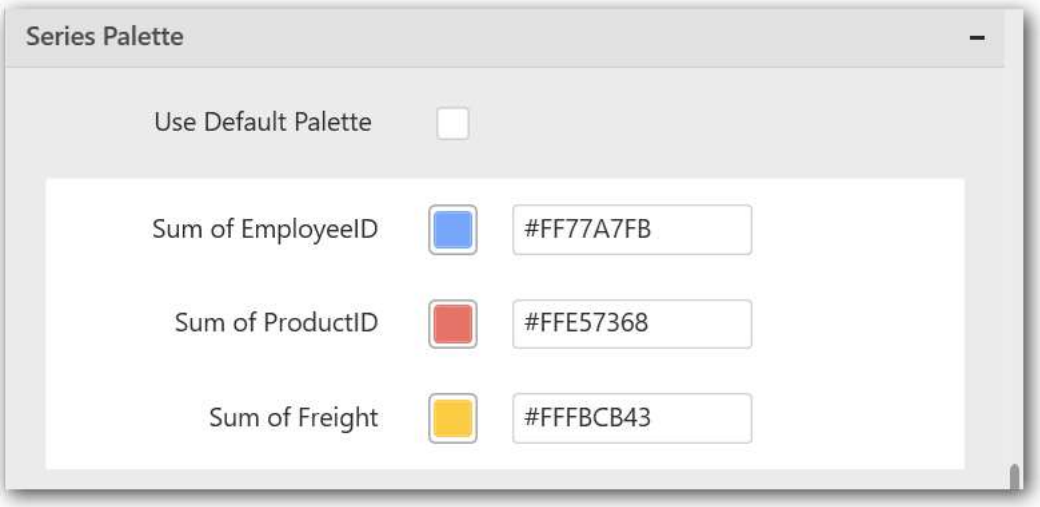


Series Palette

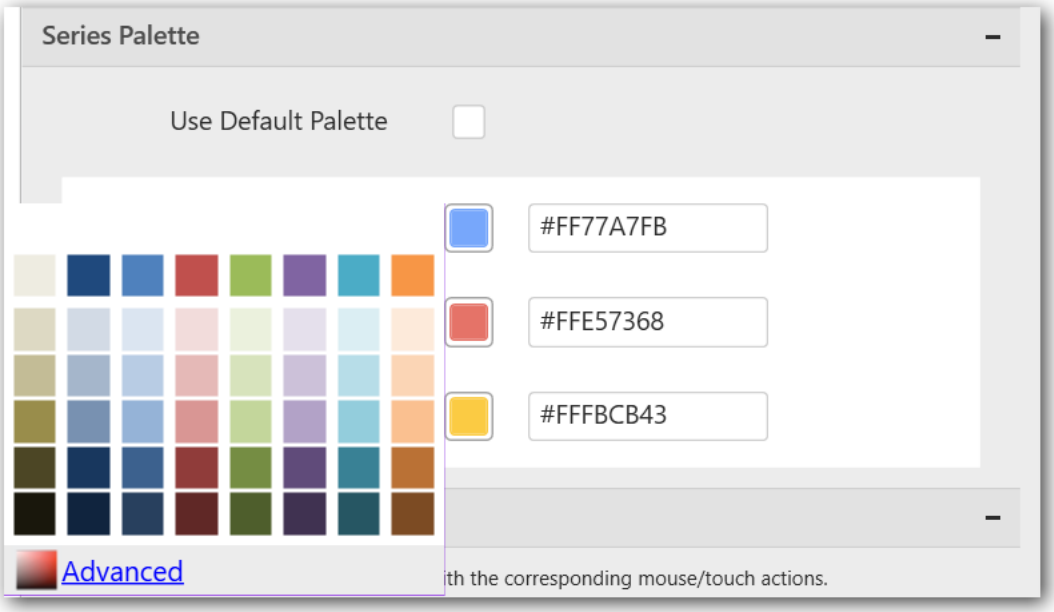
This allows you to customize the chart series color through Series Palette section.

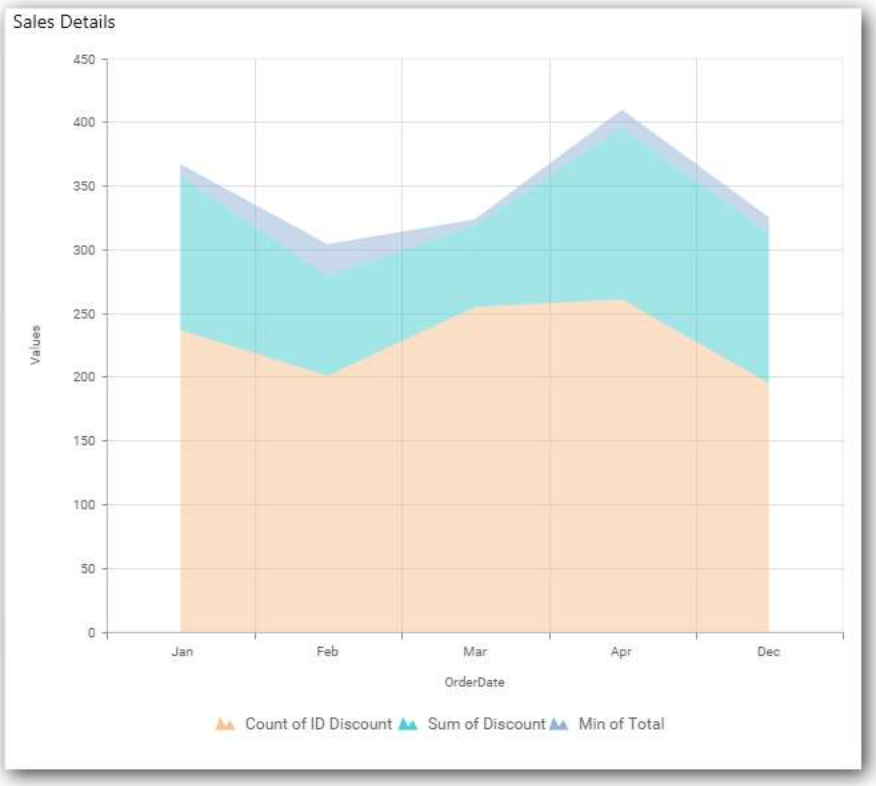
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





100% Stacked Area Chart

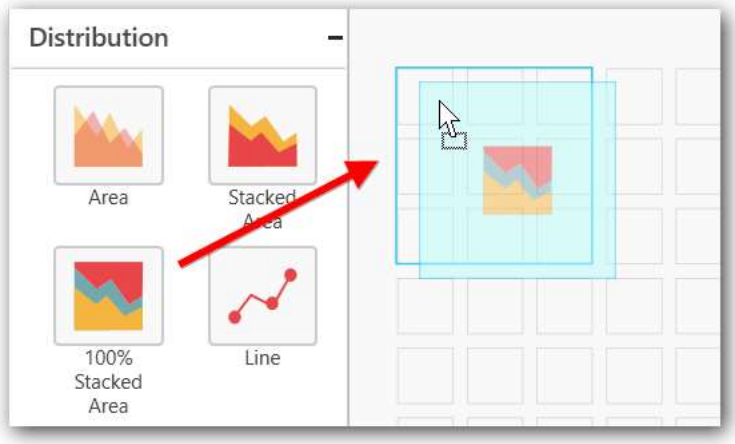
100% Stacked Area Chart allows you to compare multiple measures through filled curves stacked one after the other vertically.

How to configure the flat table data to 100% Stacked Area Chart?

100% Stacked Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Follow the steps to configure data to 100% stacked area chart

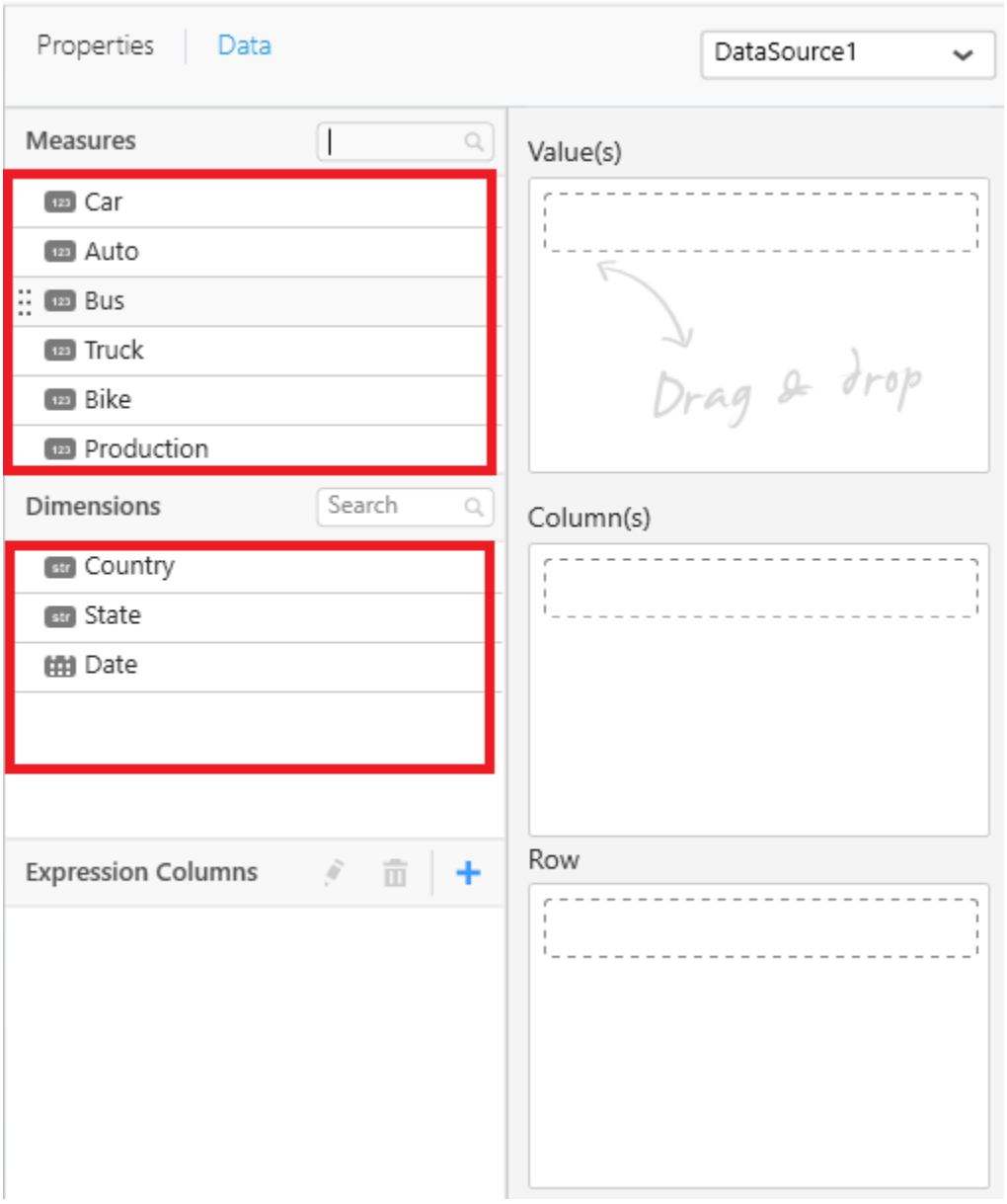
Drag and drop the 100% stacked area chart into canvas and resize it to your required size.



Connect to the data source.

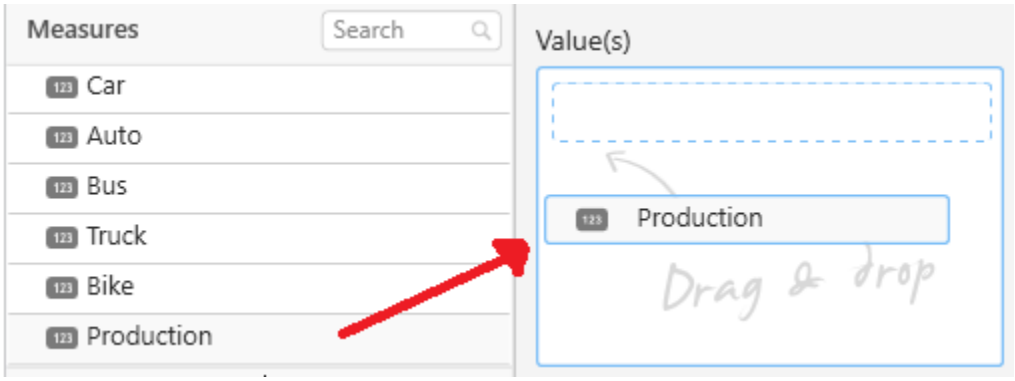
Focus on the 100% stacked area chart and click on Assign Data.

The data pane will be opened with available measures and dimensions from the connected data source.

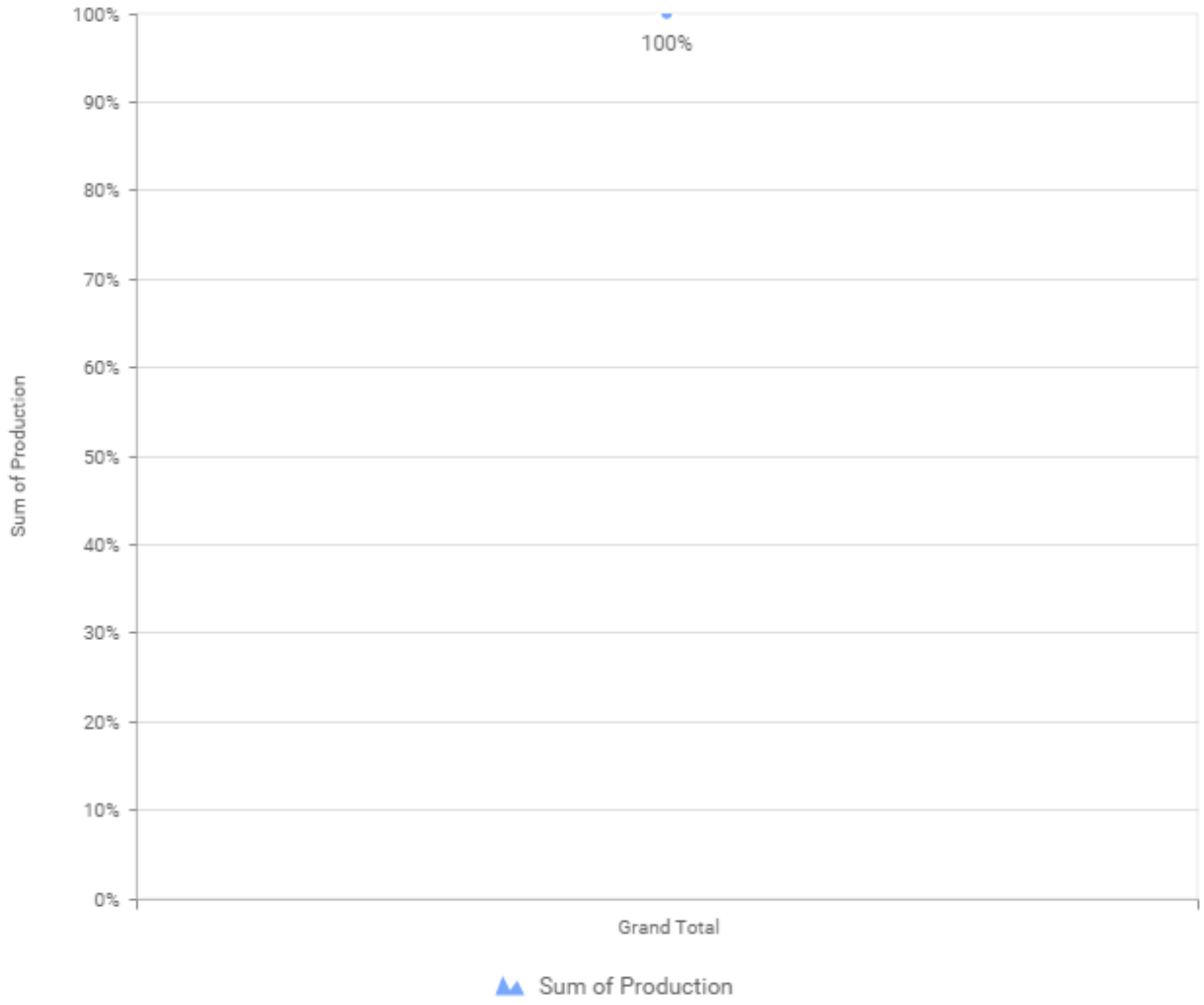


Assigning Value(s)

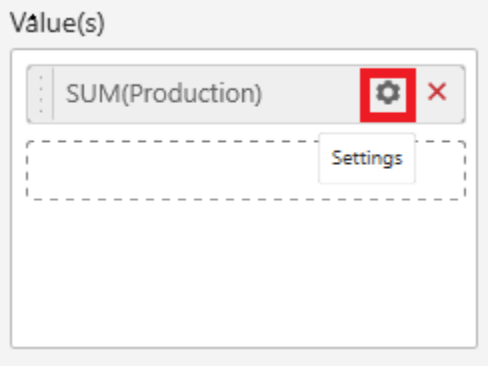
Drag and drop the Measure into Value.



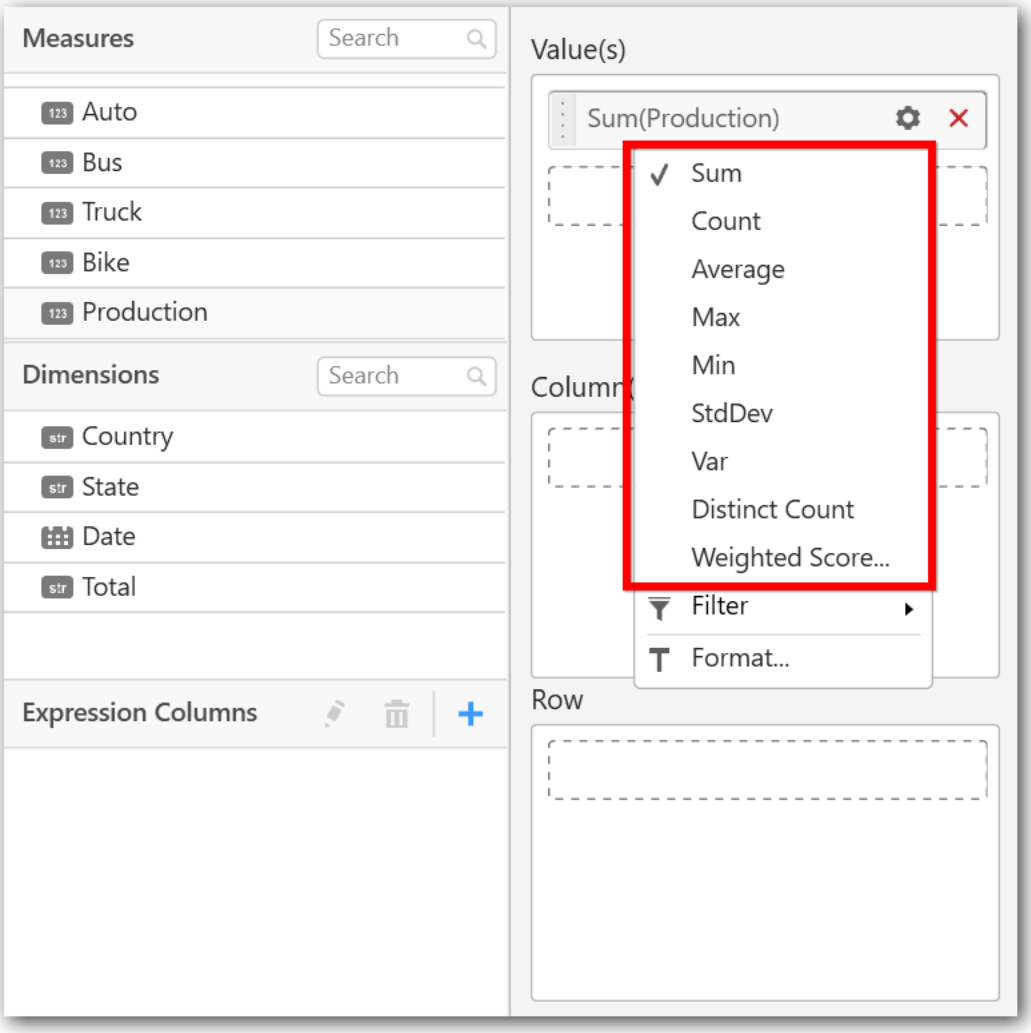
Now the chart will be rendered like this



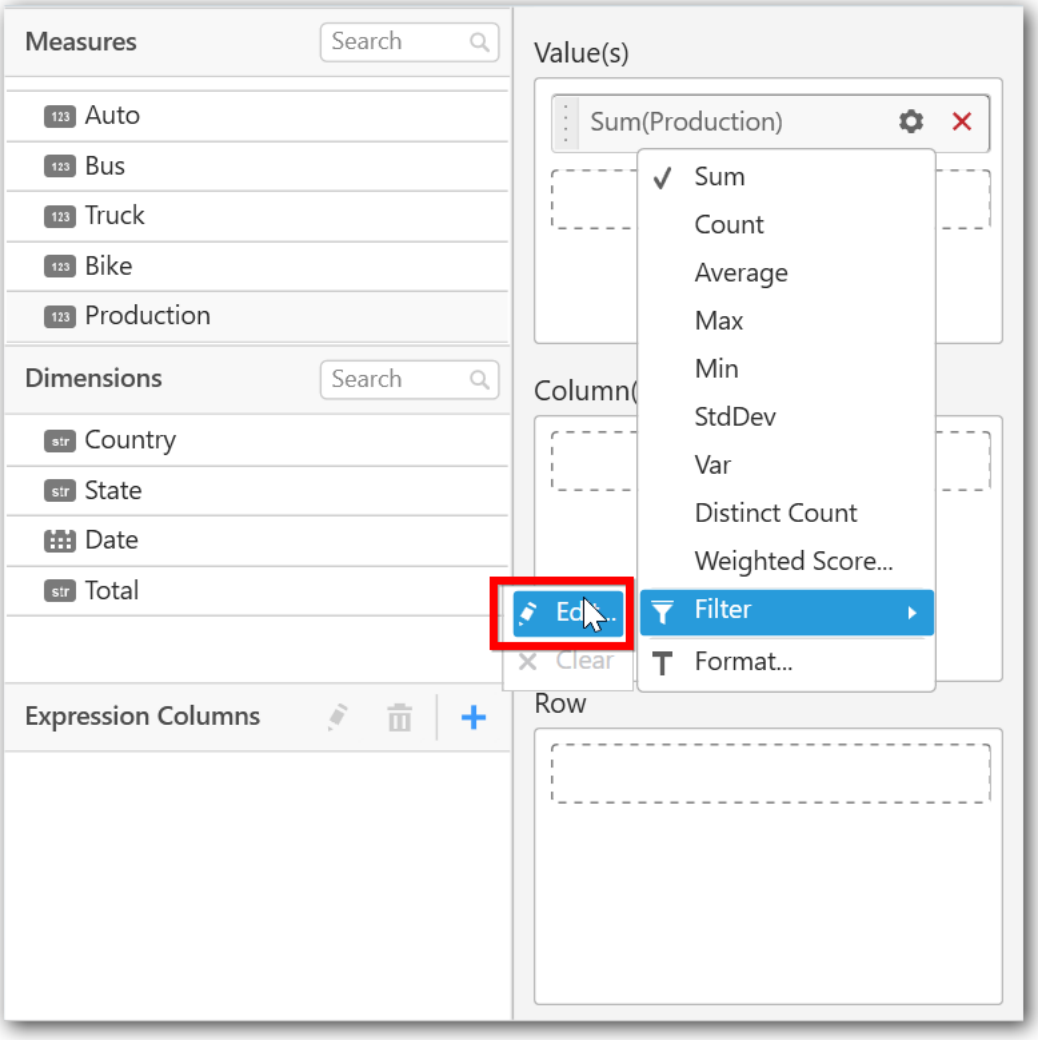
You can change the summary type of the value by clicking on Settings option.



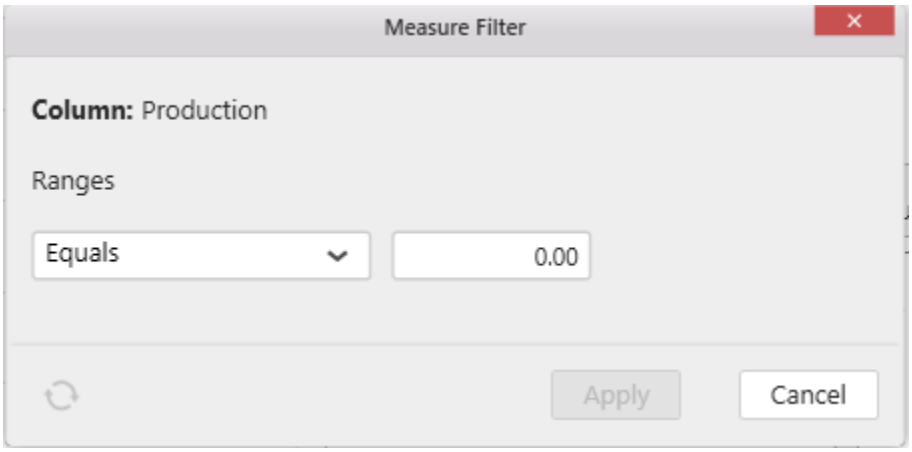
Select the required summary type from list.

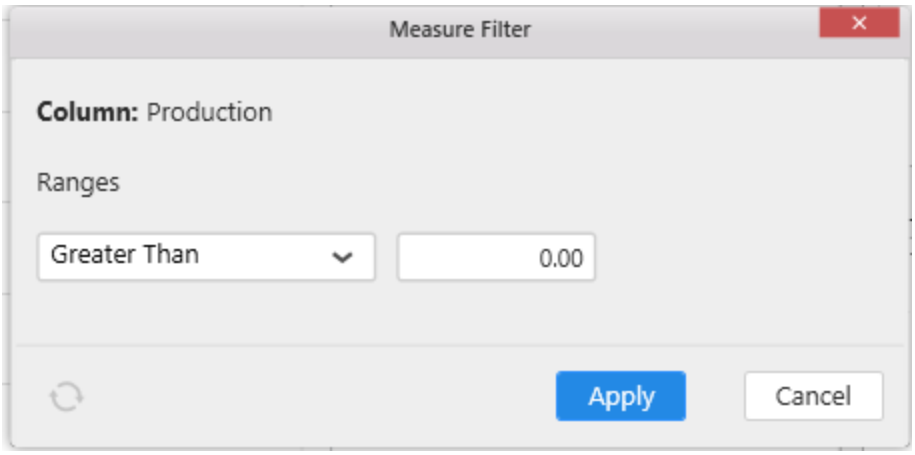


You can select what data to be displayed by choosing filter option.

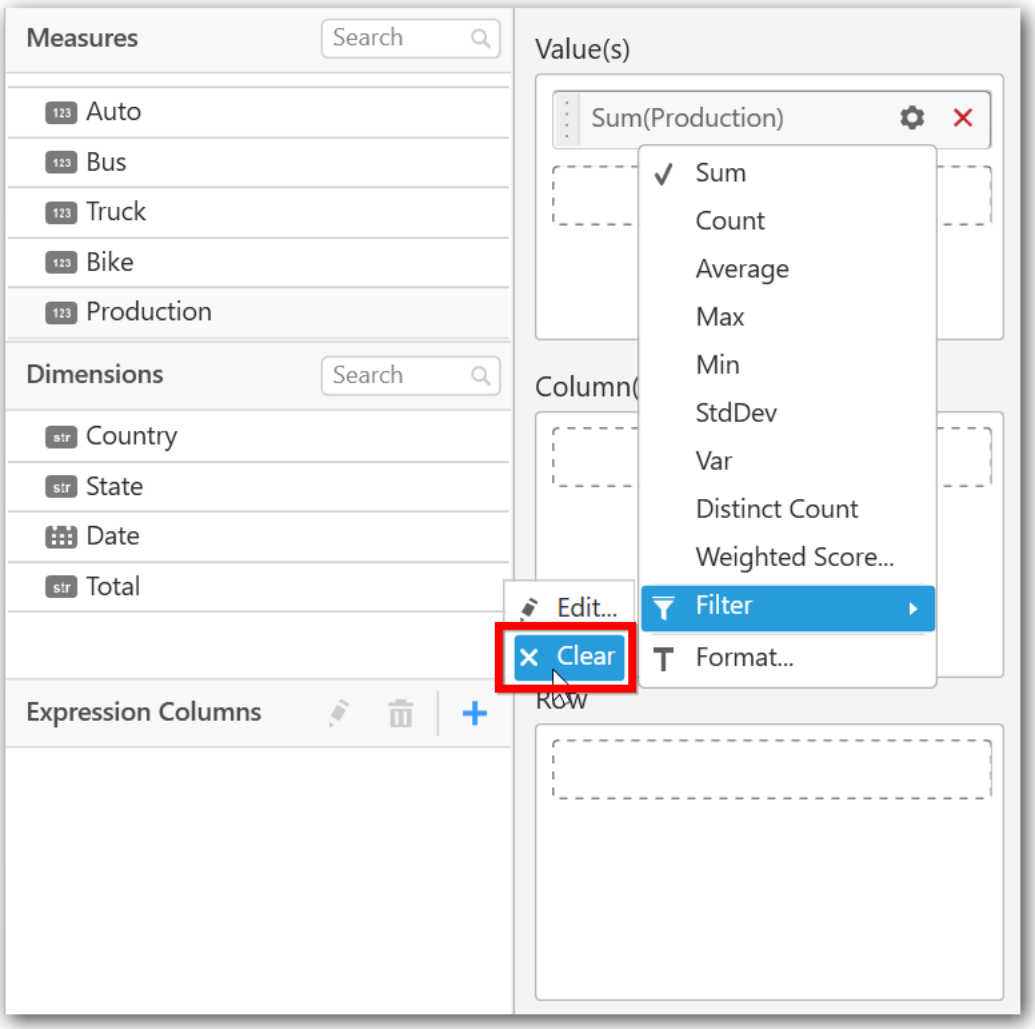


The Measure Filter option will be shown.

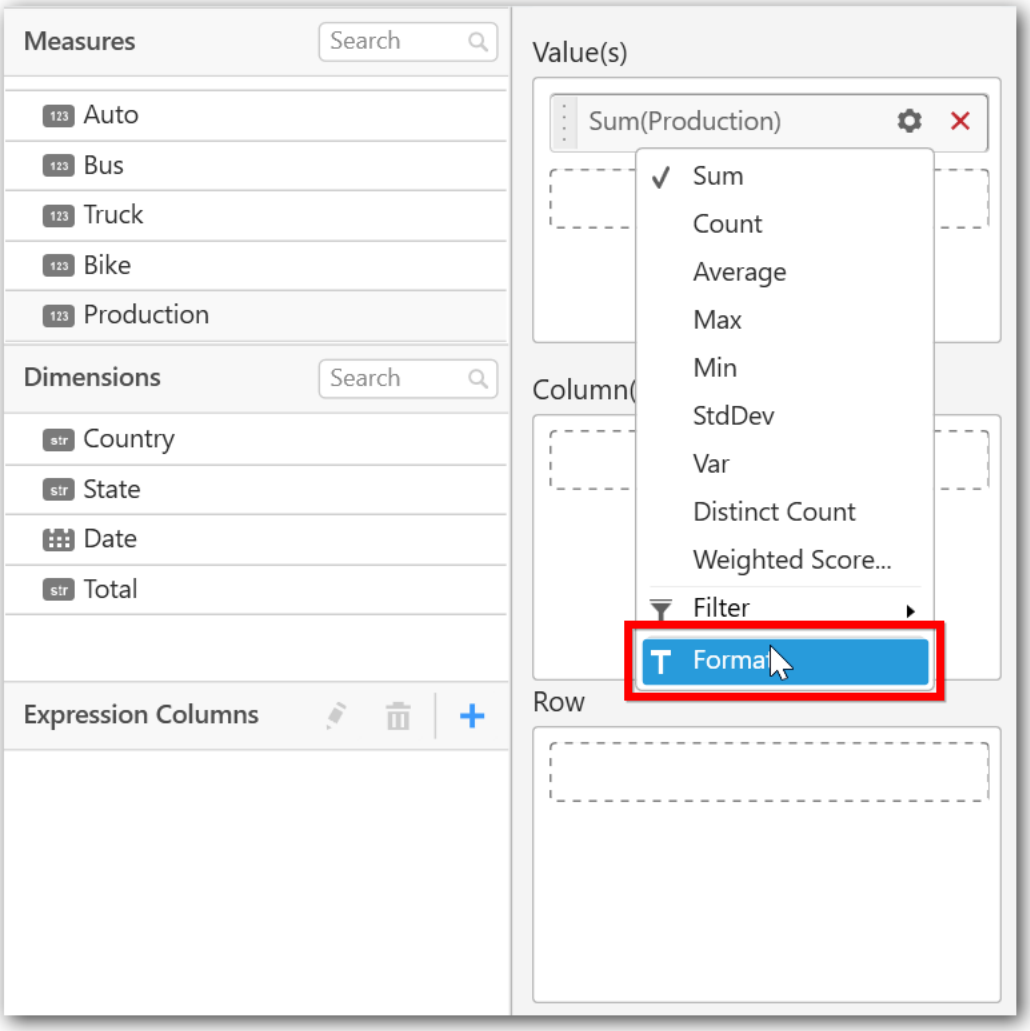




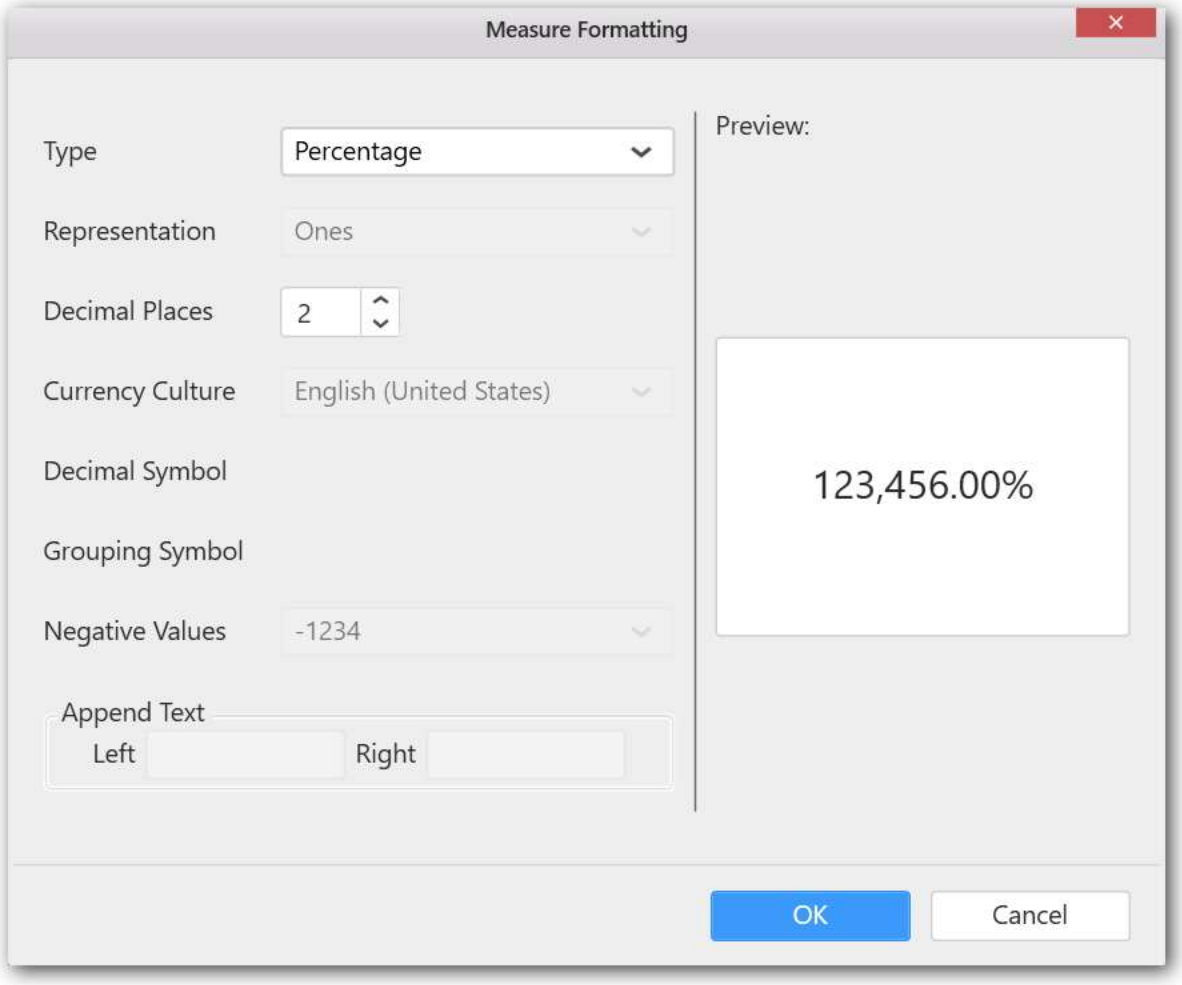
You can clear the filter.



You can Format the value.

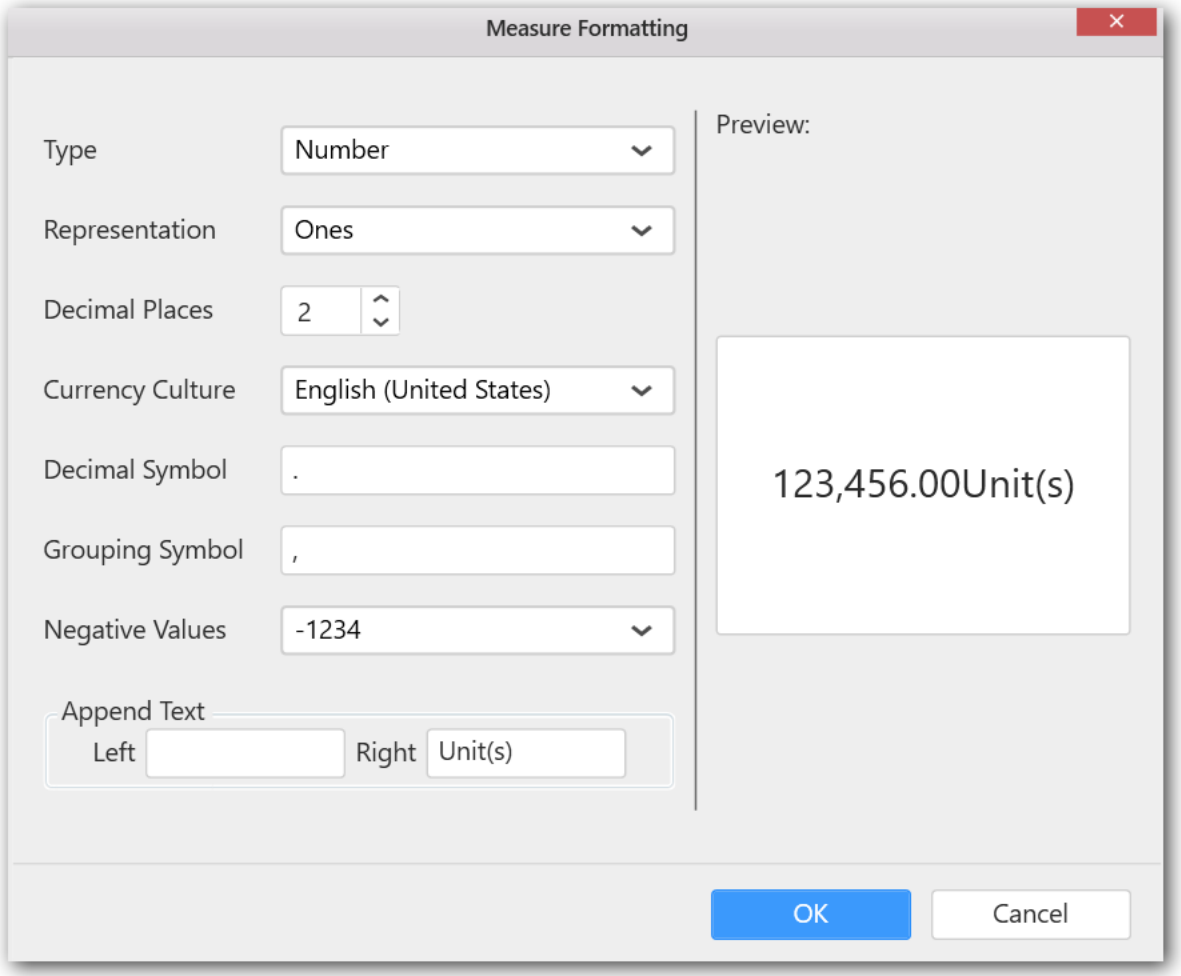


The format options will be shown.

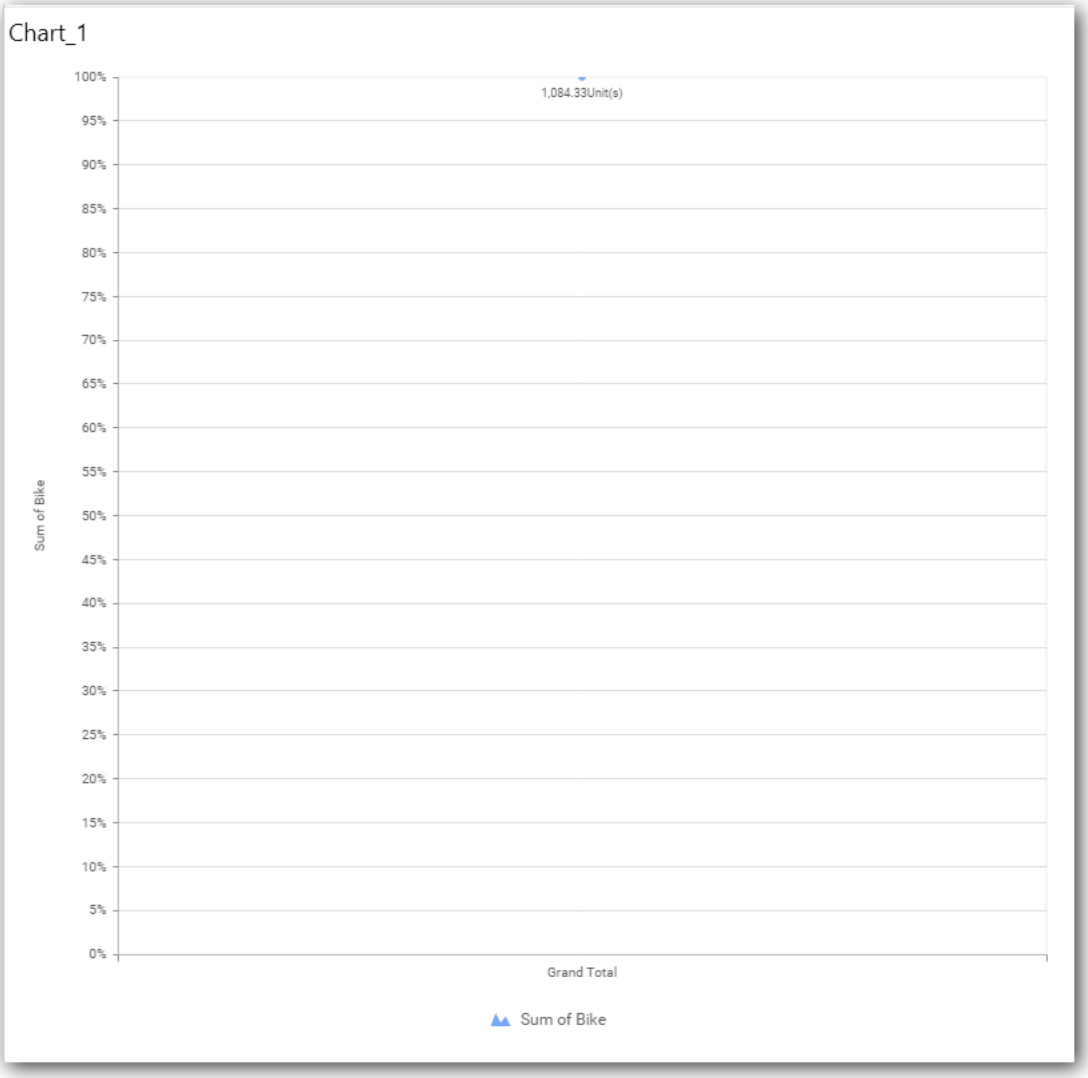


Note: For 100% Stacked Charts, default format type is Percentage.

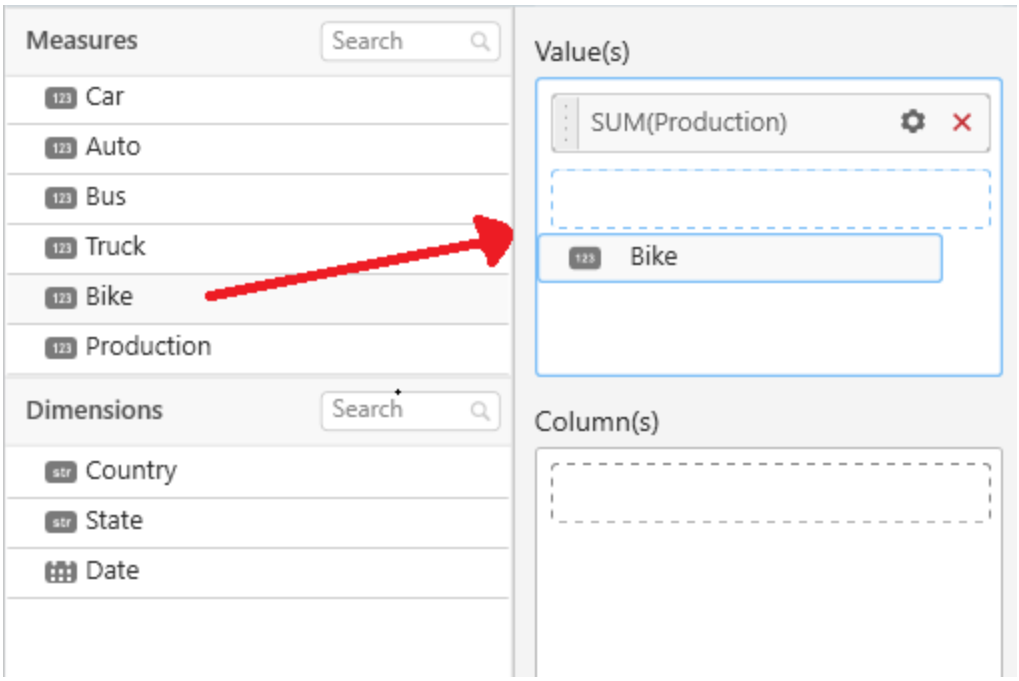
Choose the options you need and click OK.

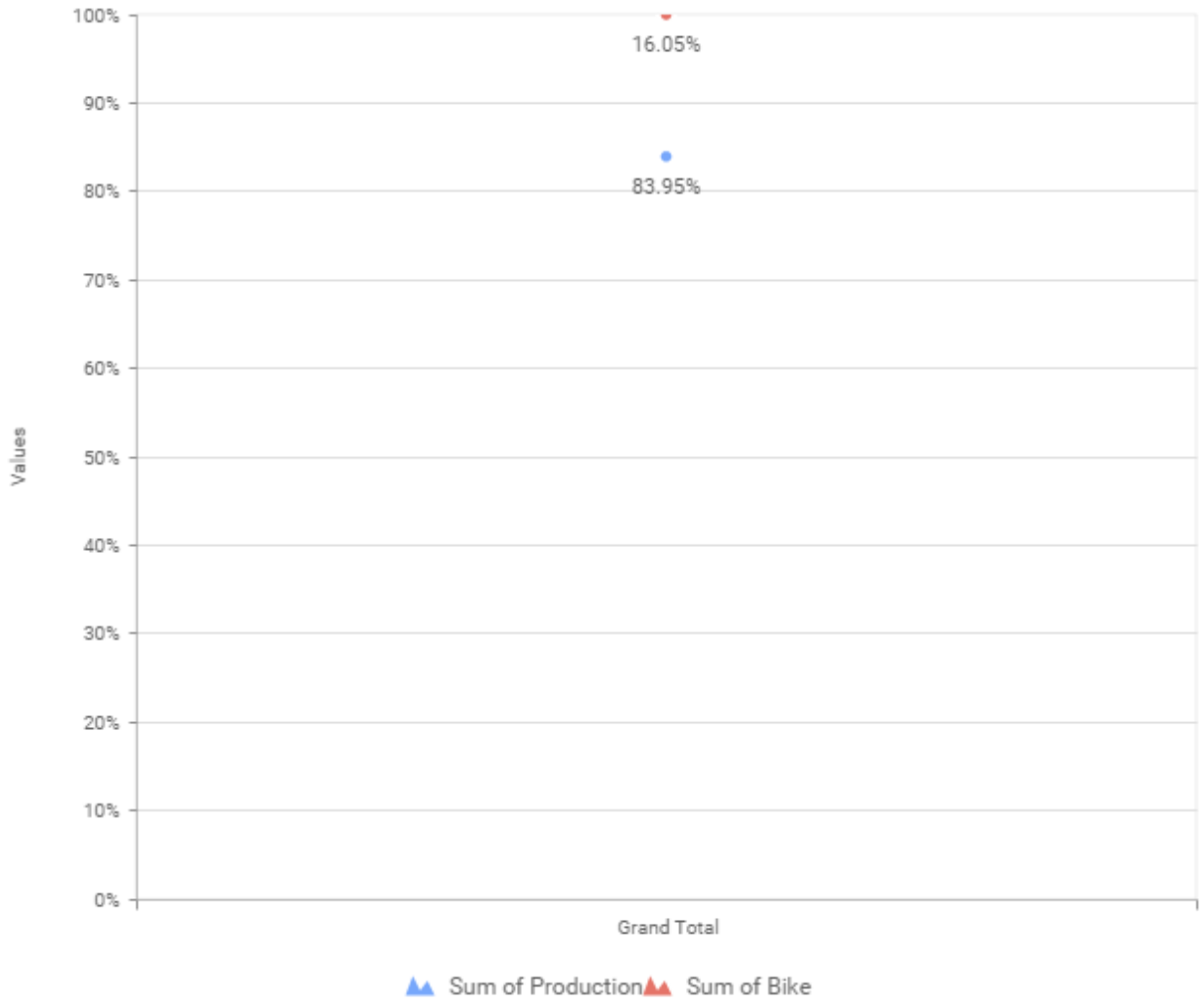


Now the Chart will be rendered like this.



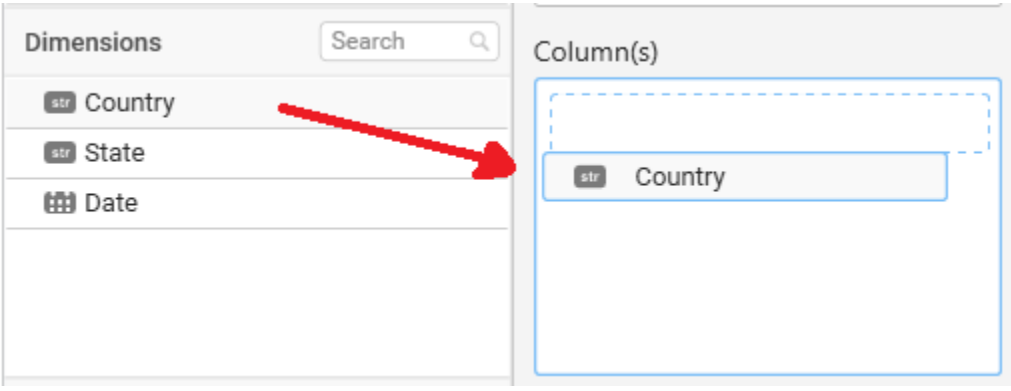
You can add more number values by drag drop the Measures into Value field.

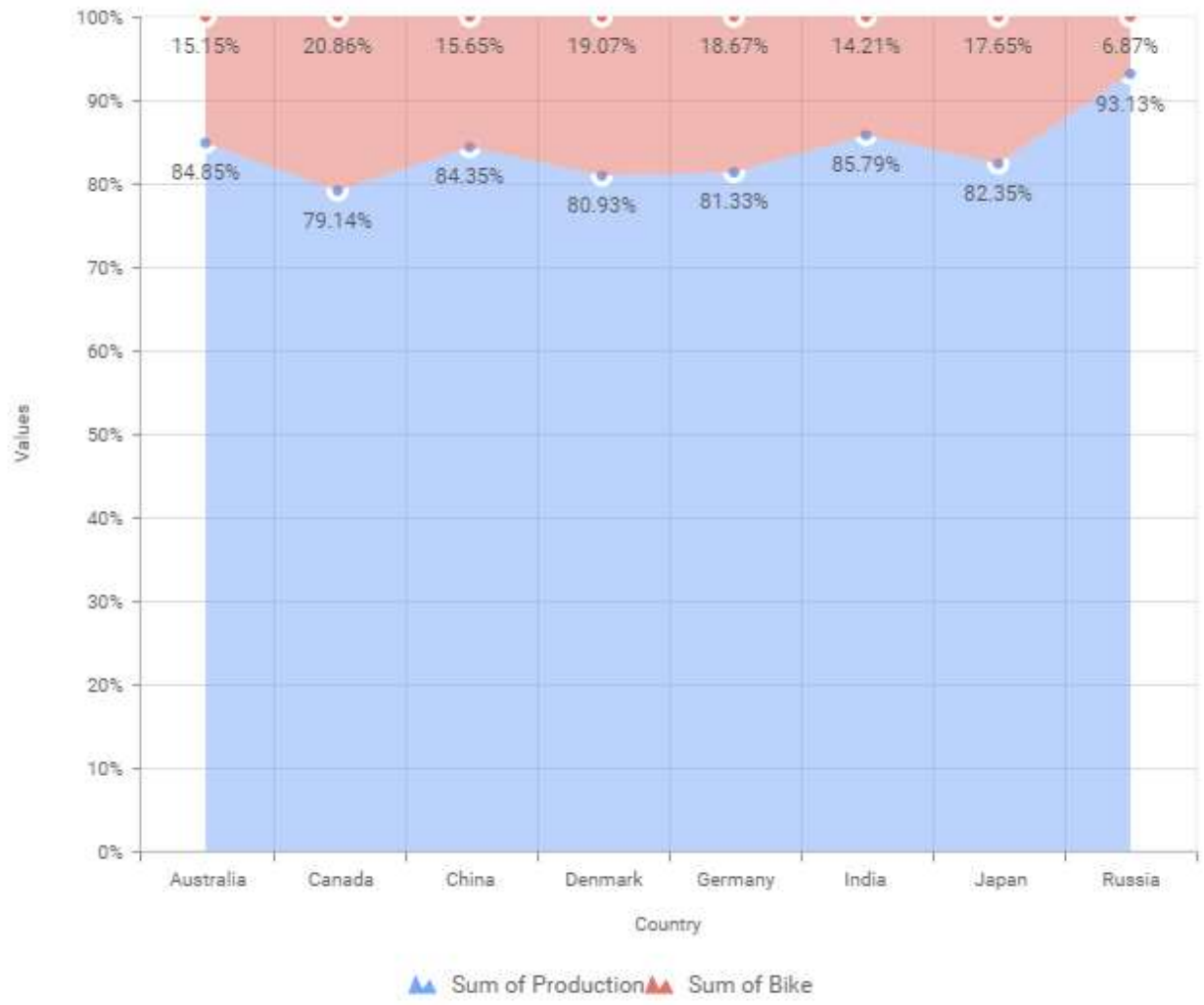




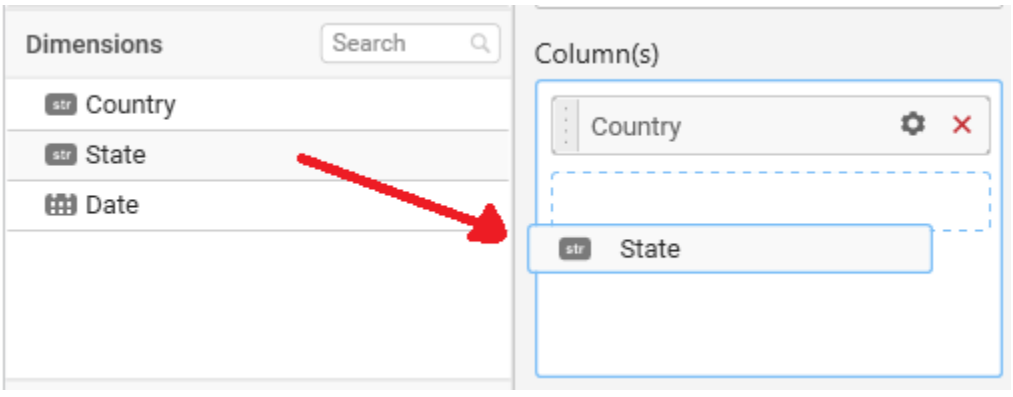
Assigning Column(s)

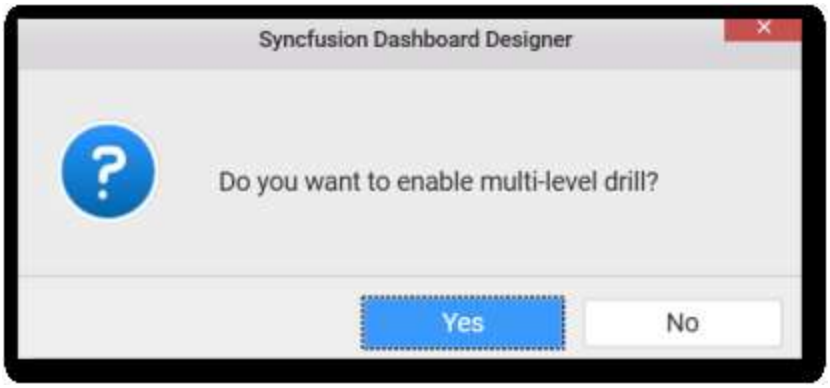
You can add the Dimension into Column field by drag and drop.





You have option to add more than one Column Value.

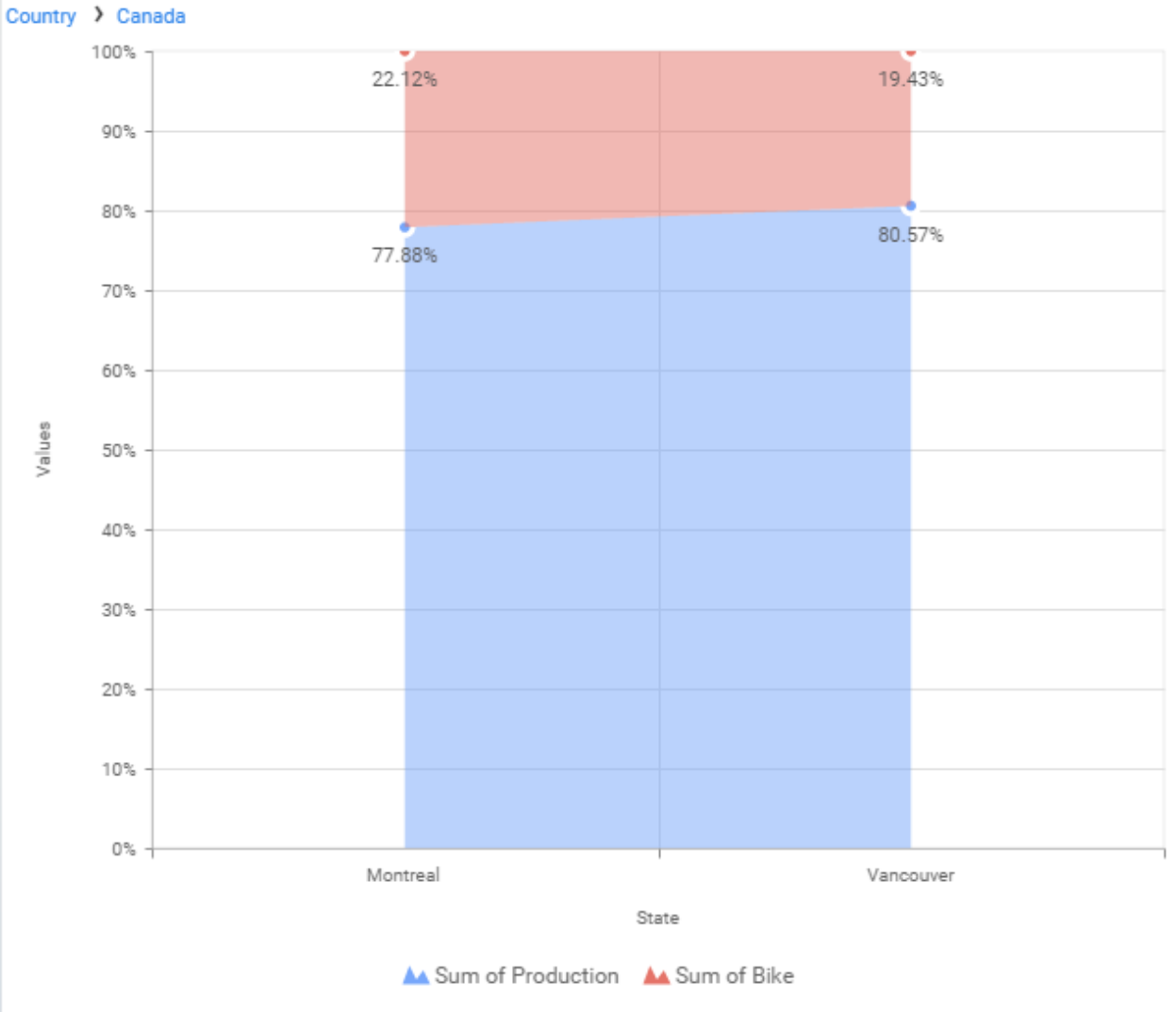




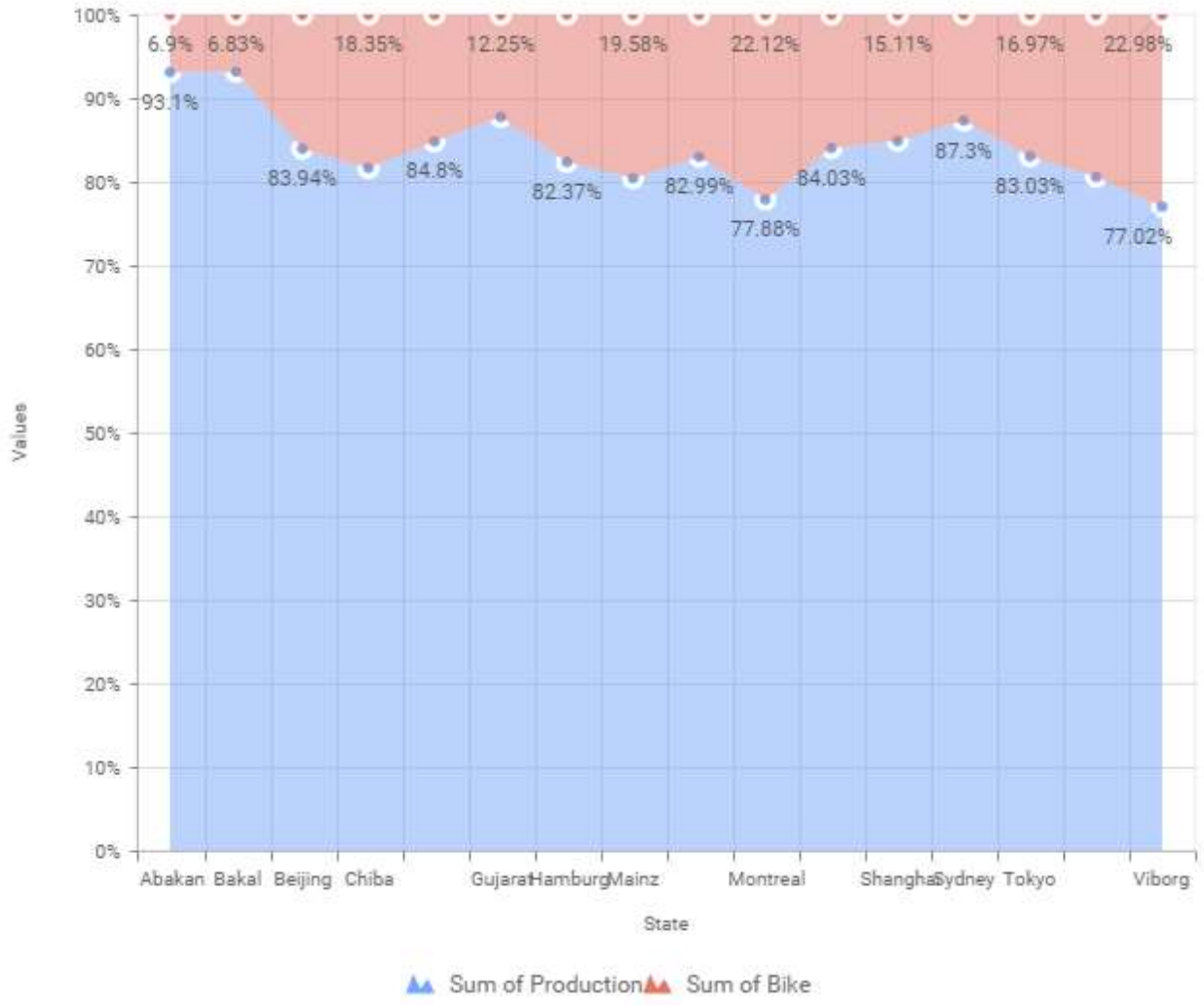
- If you choose Yes Drill down option will be enabled.

You can drill down the chart by clicking on the chart.

The drilled view of the chart is follows.

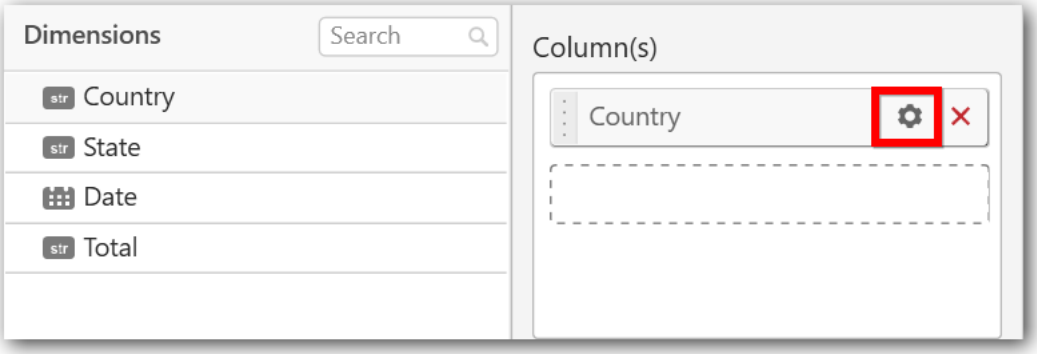


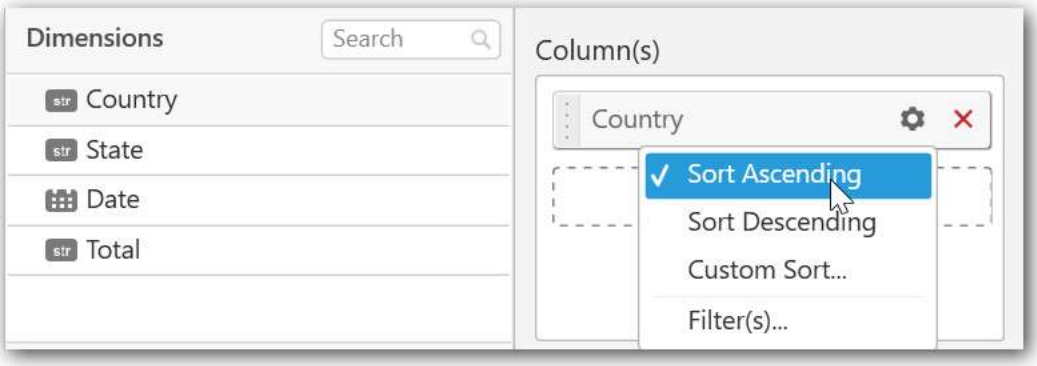
- If you click No the new Dimension value will replace old value.



You can also add Measures and Expression columns into Column(s) field.

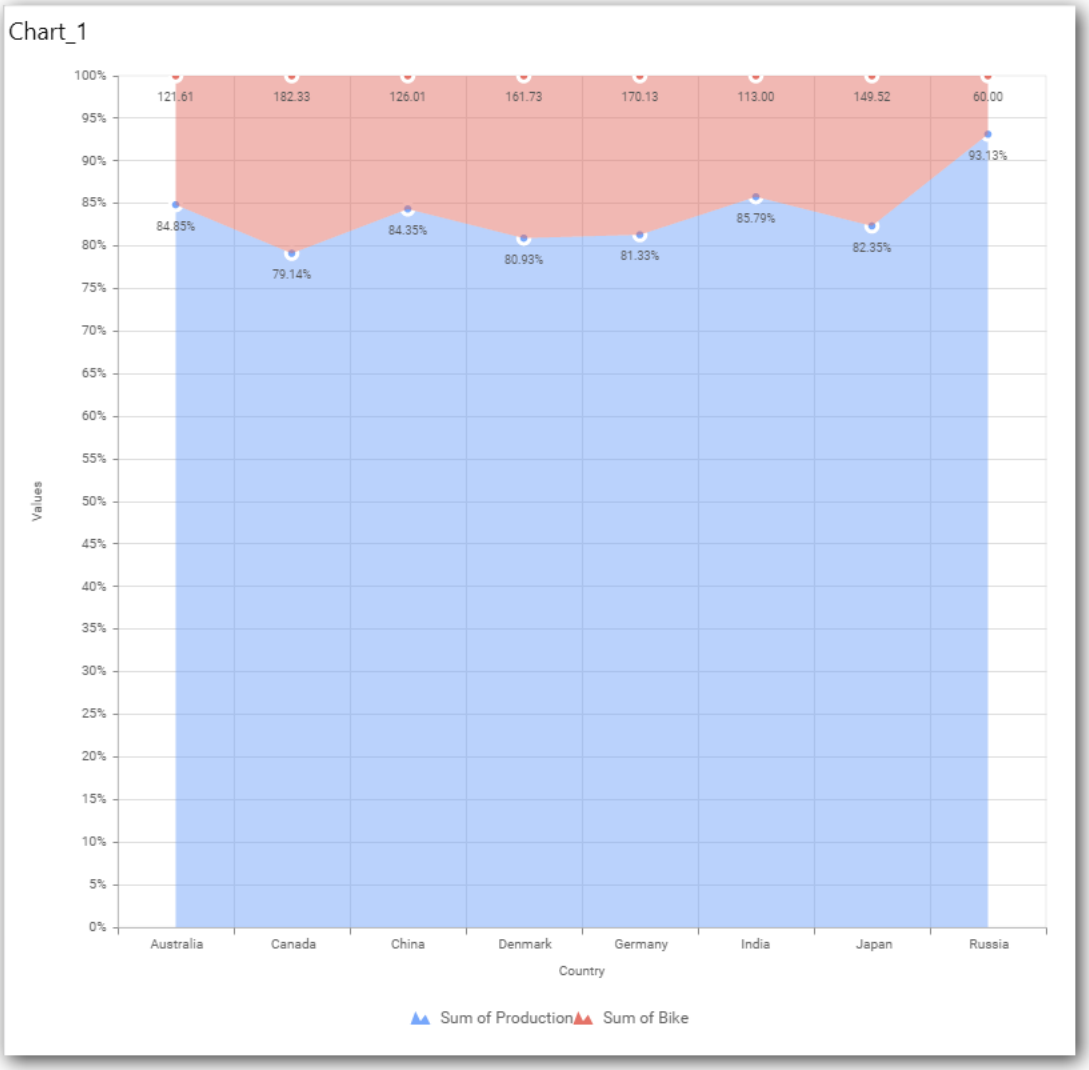
You have options to change the settings.



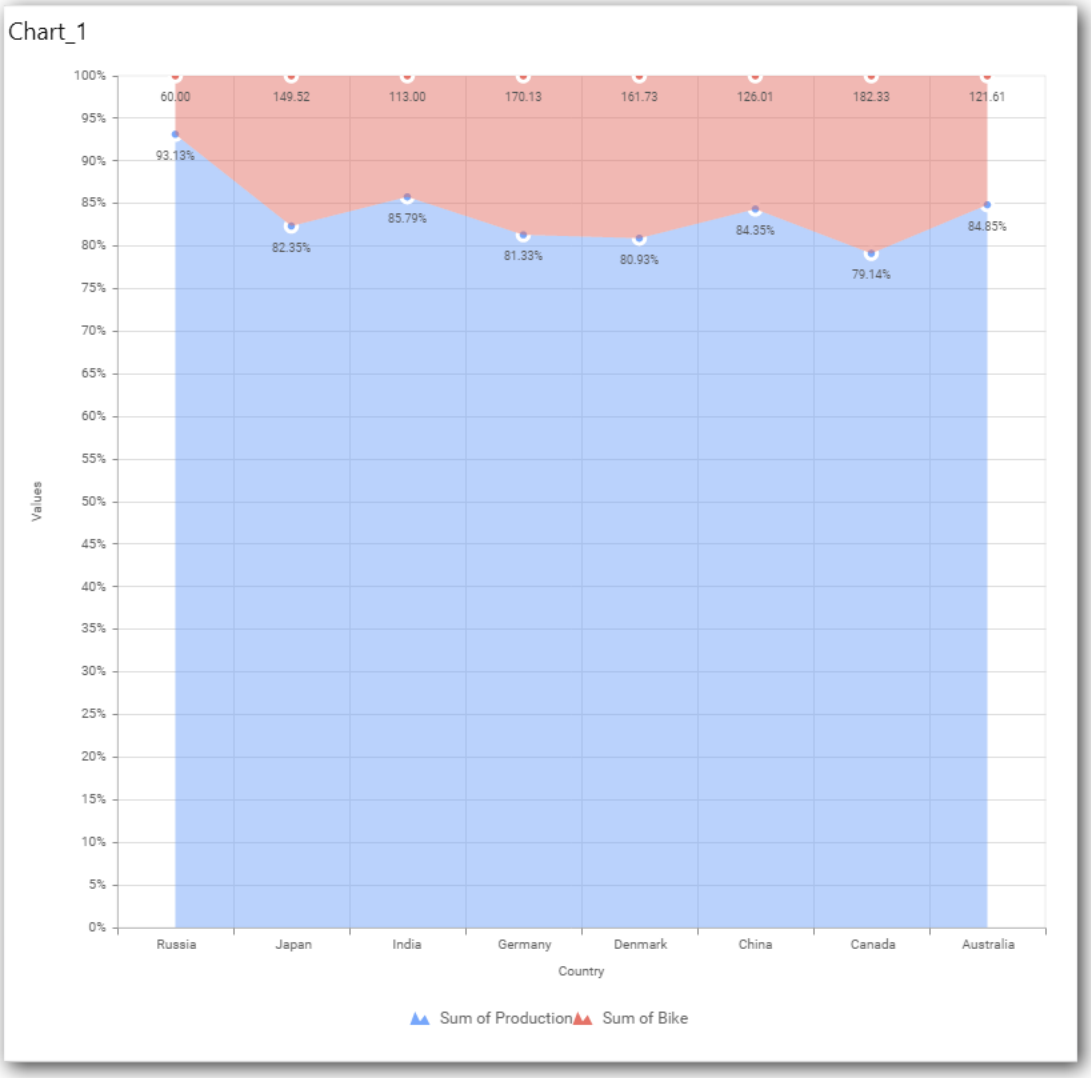


You can sort the chart either in Ascending or Descending series.

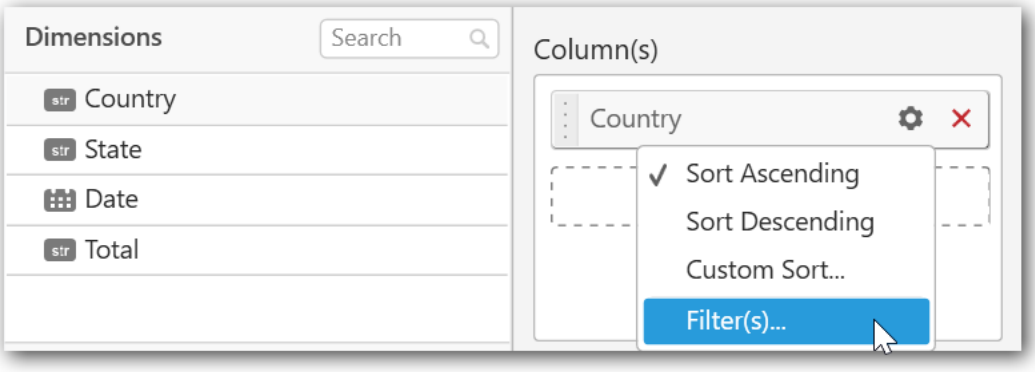
Ascending Order

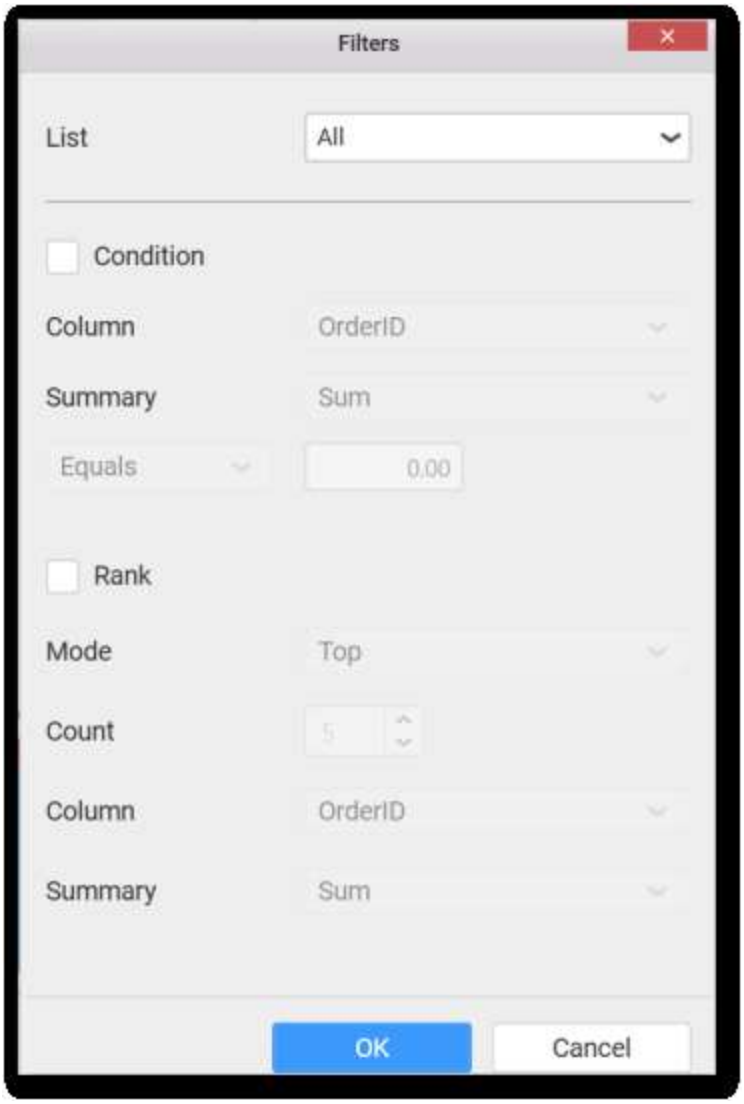


Descending order

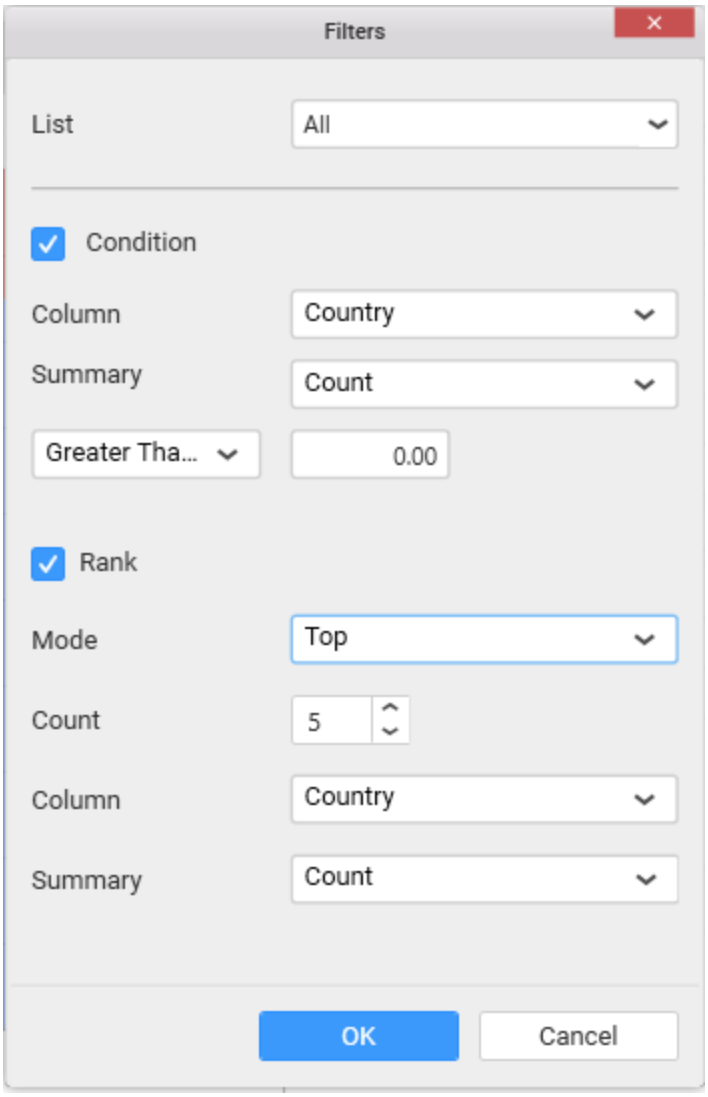


You can apply a filter.

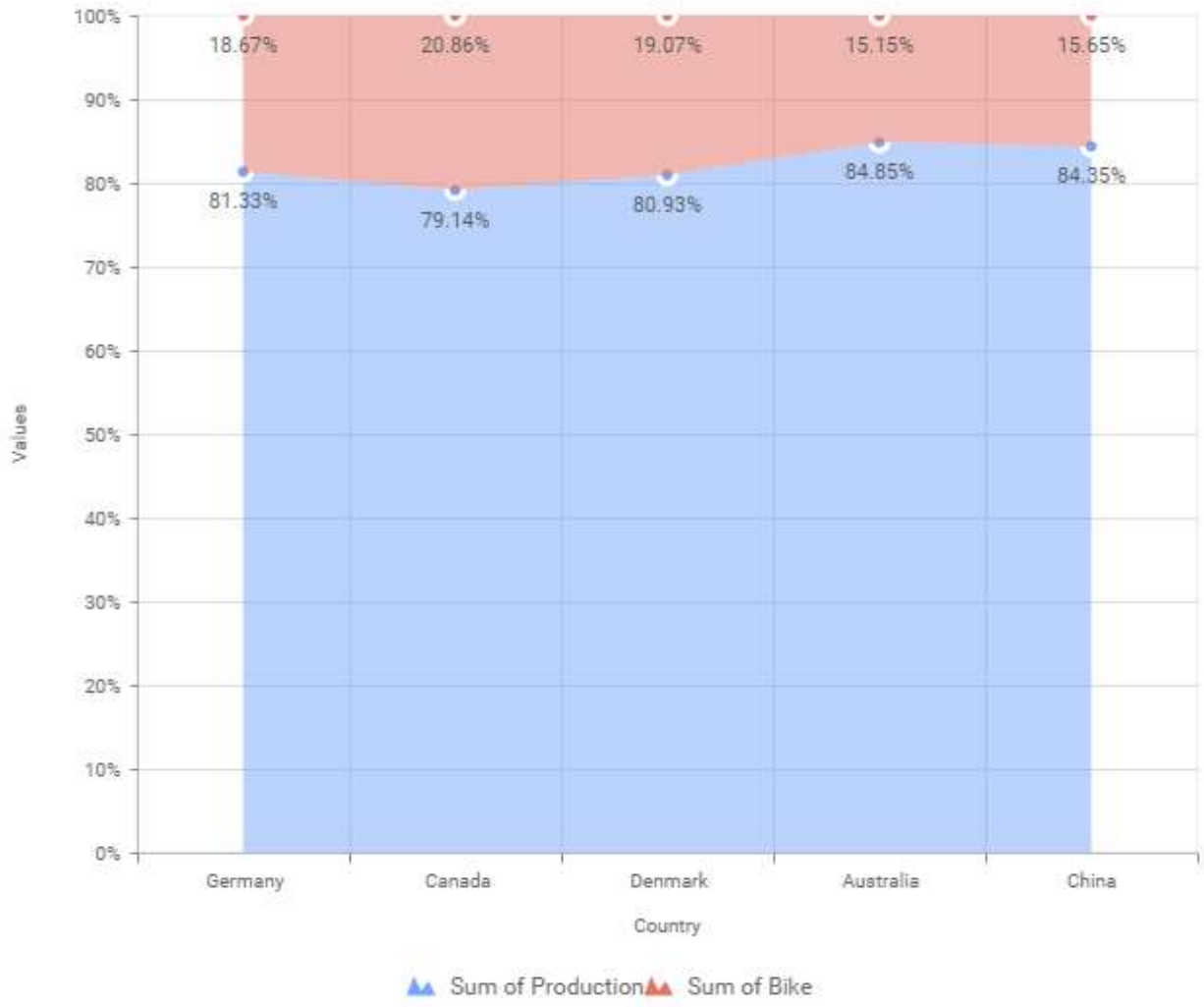




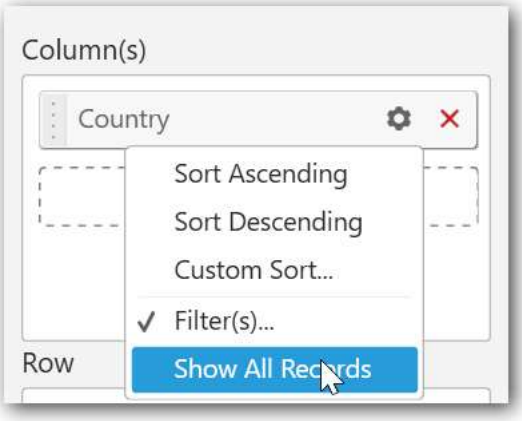
Select the Conditions and Rank you need.



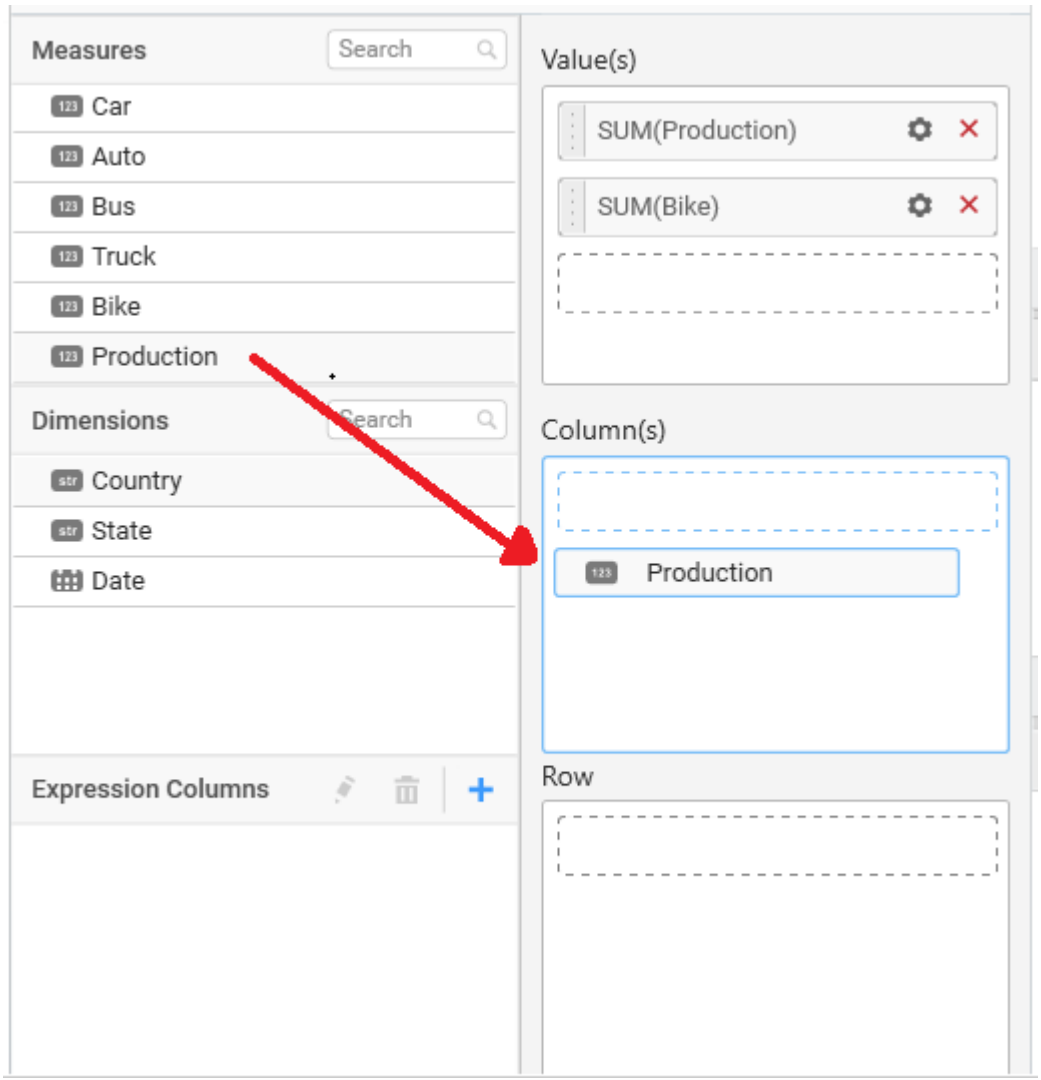
Now the chart will be rendered like this.

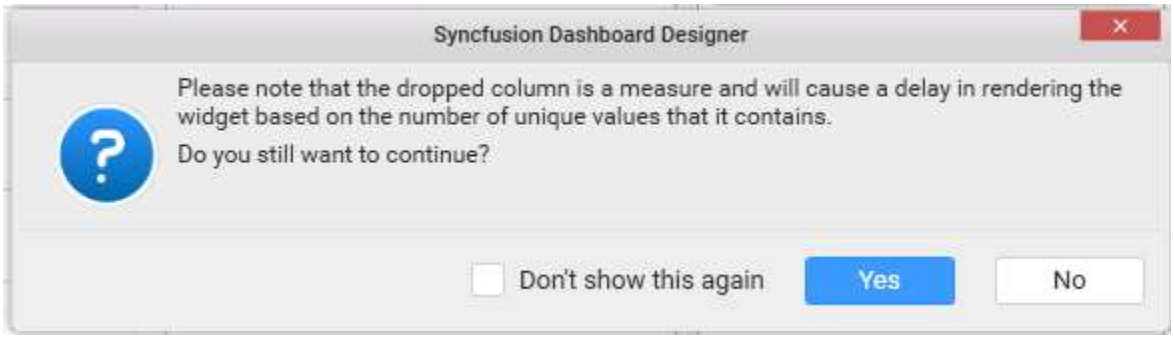


To show all records again click on Show All Records.



On adding measures into Column(s) will show the following alert.

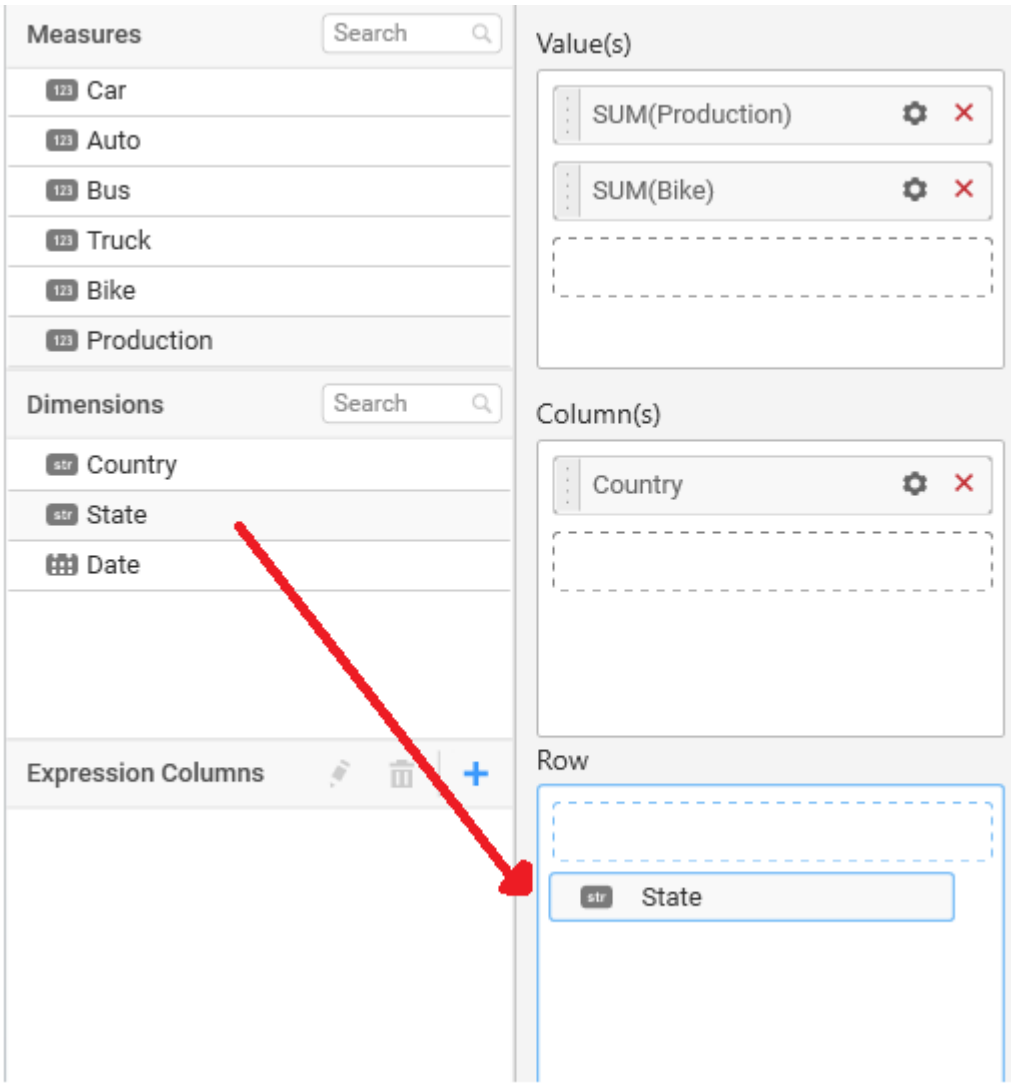




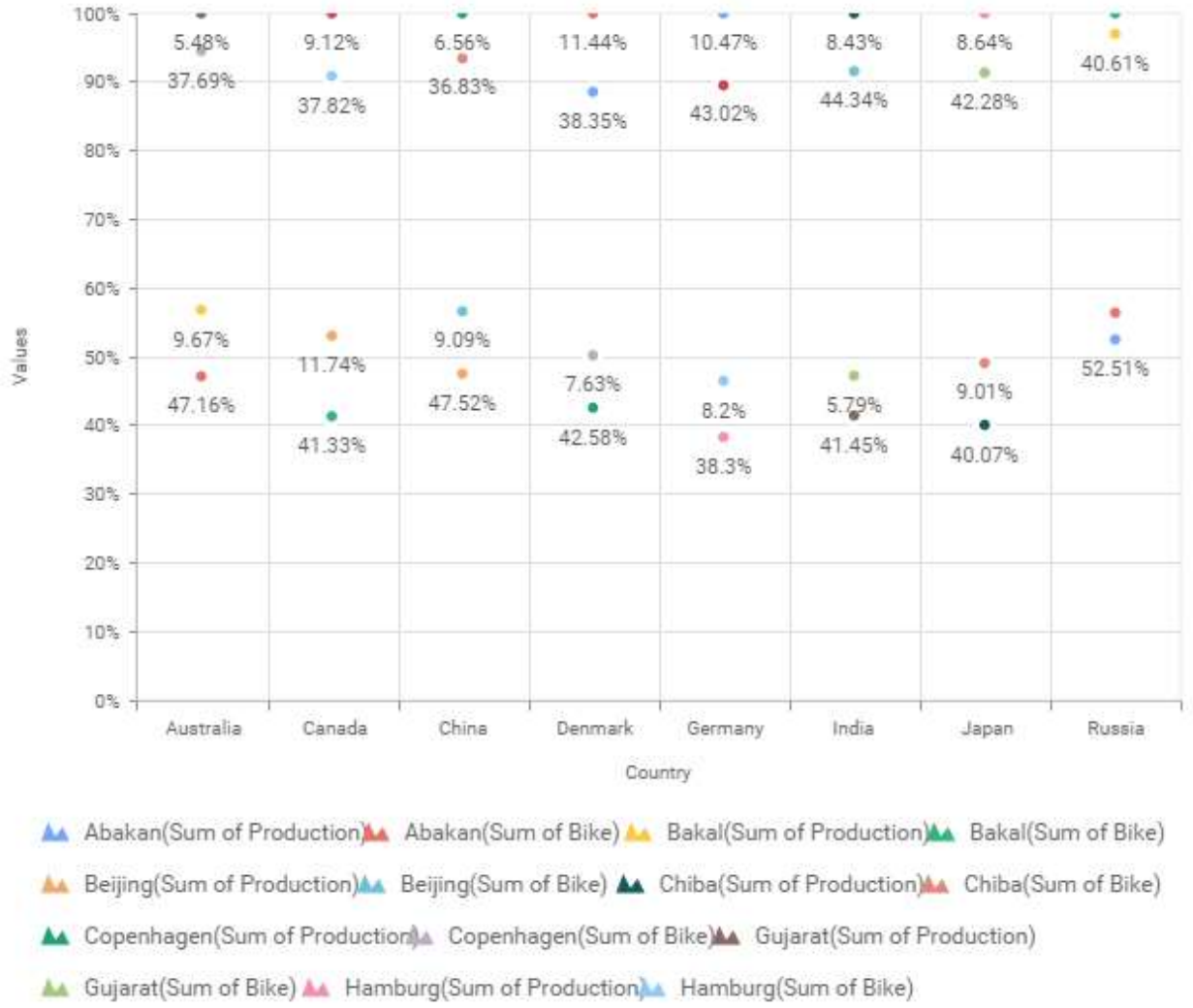
To continue select Yes, otherwise select No.

Assigning Row

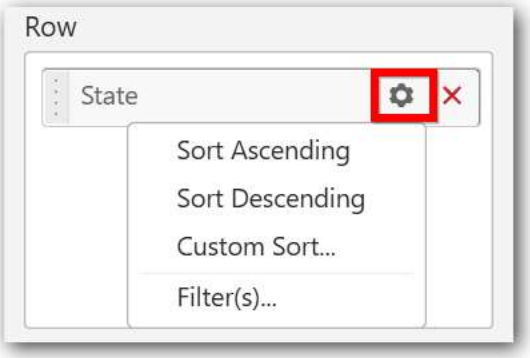
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s).



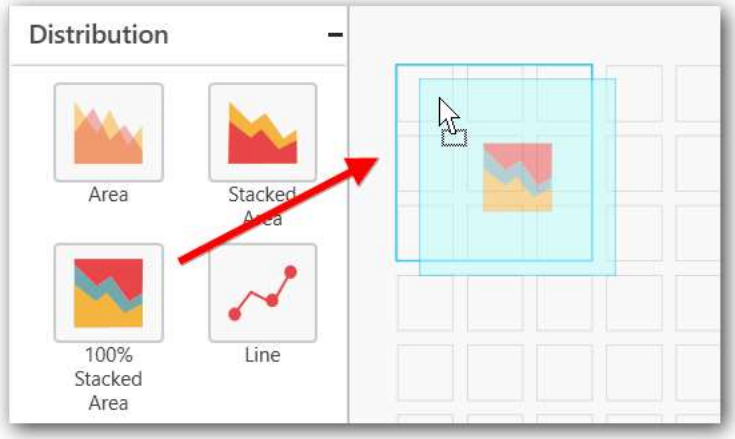
How to configure SSAS data to 100% stacked Area Chart?

100% Stacked Area Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you

would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuration of SSAS data to 100% Stacked Area chart

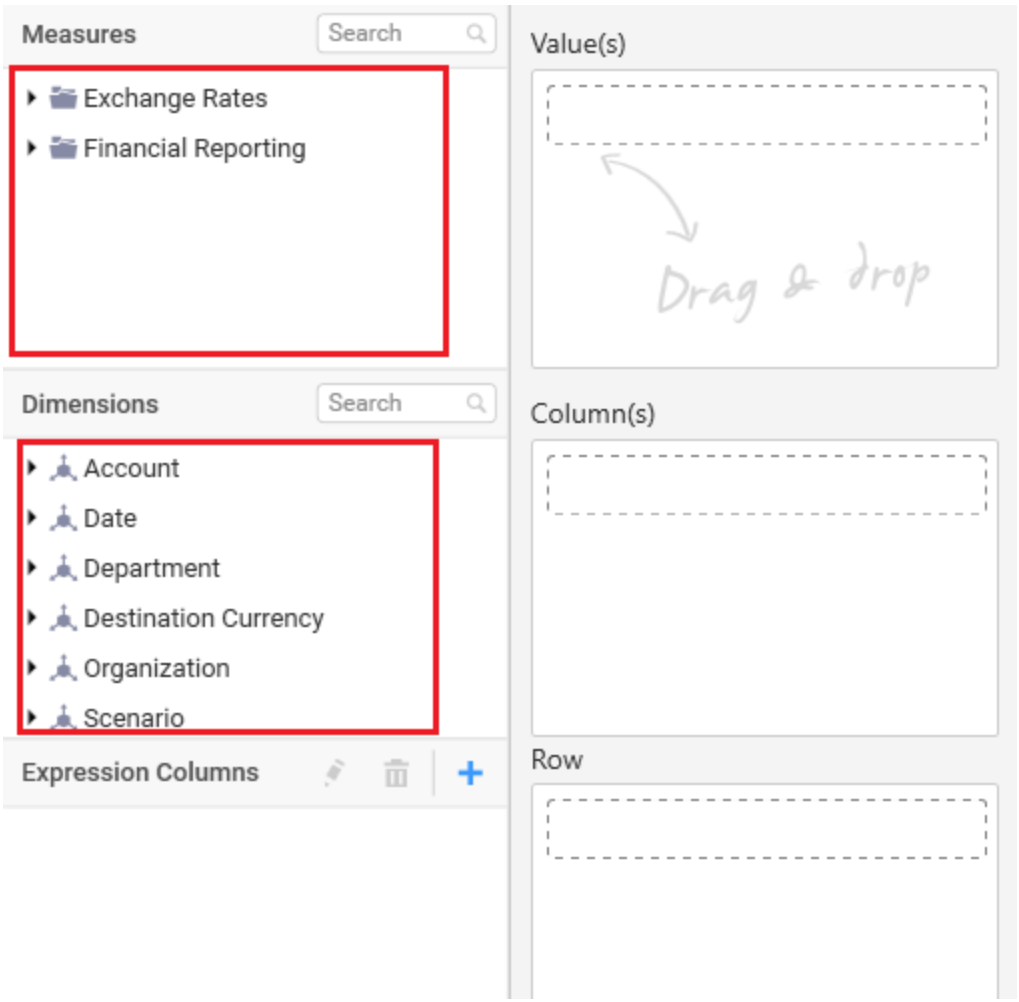
Drag and drop the 100% Stacked Area chart widget into canvas and resize into your required size.



Select the dropped widget using mouse.

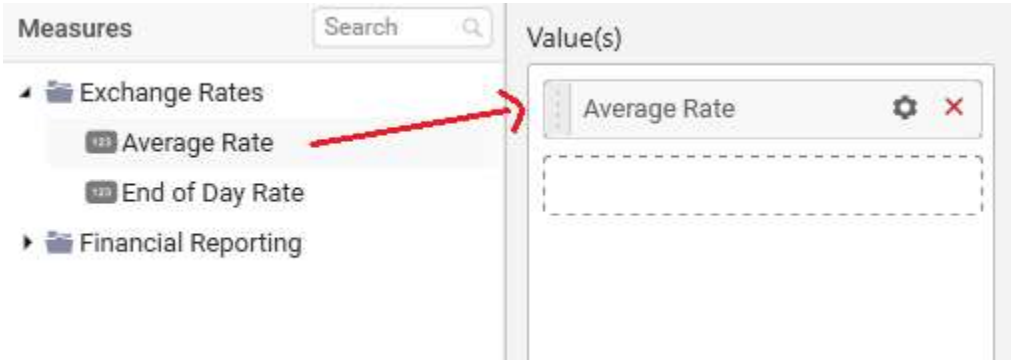
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions.

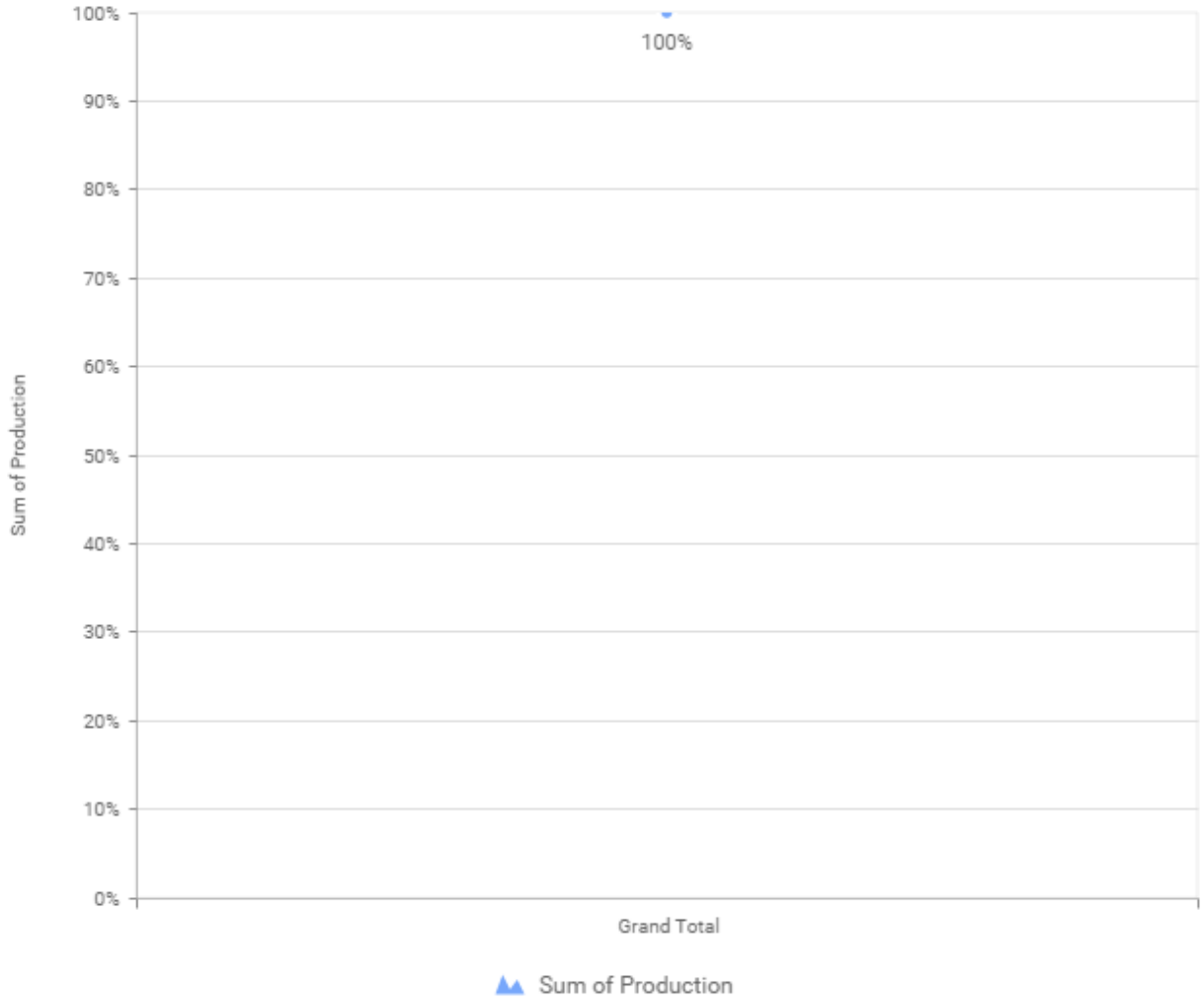


Assigning Value(s)

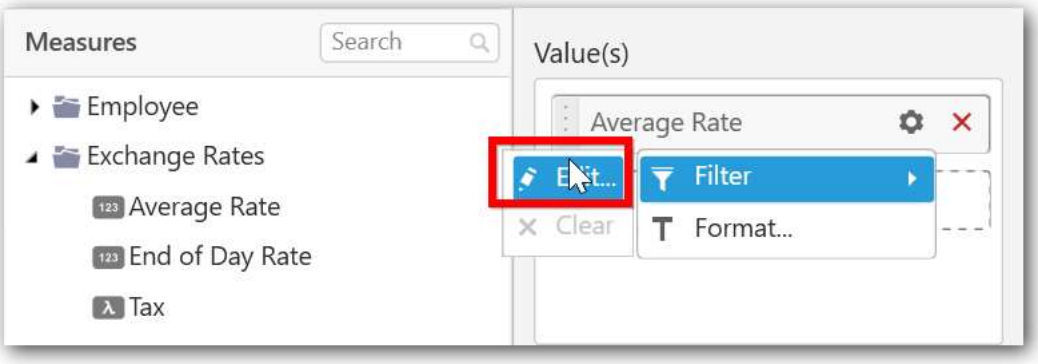
Drag and drop a column under Measures category into Value(s) section.



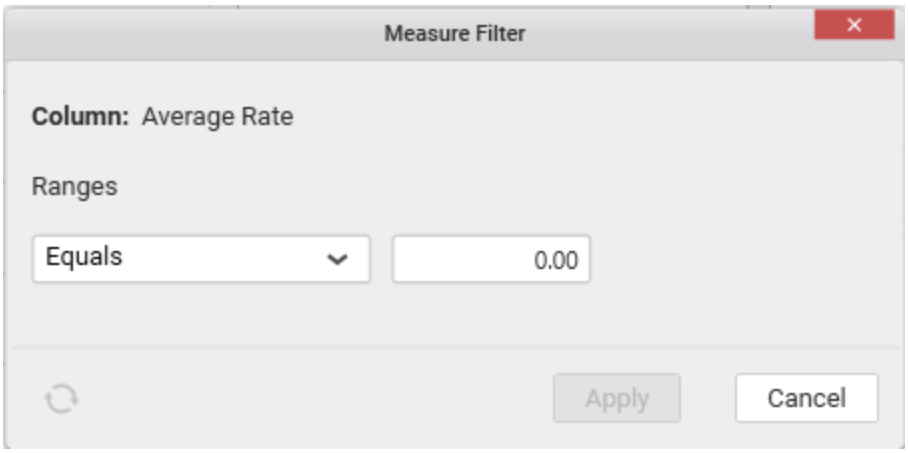
Now the chart will be rendered like this.



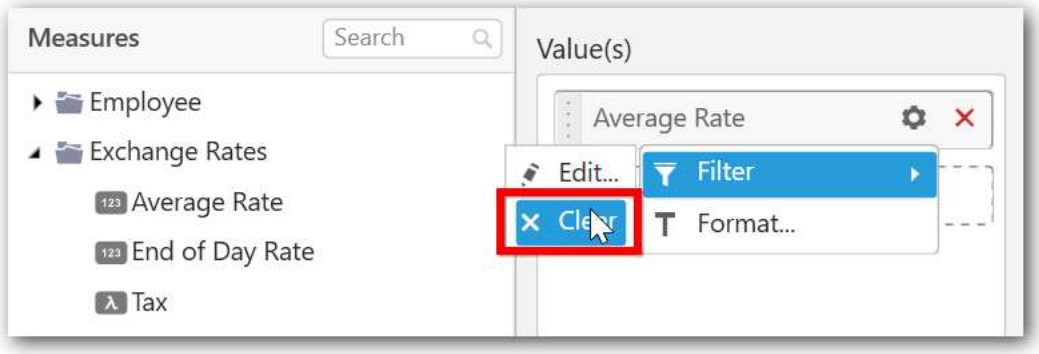
Define the filter criteria to match through choosing Edit option in Filter menu item.



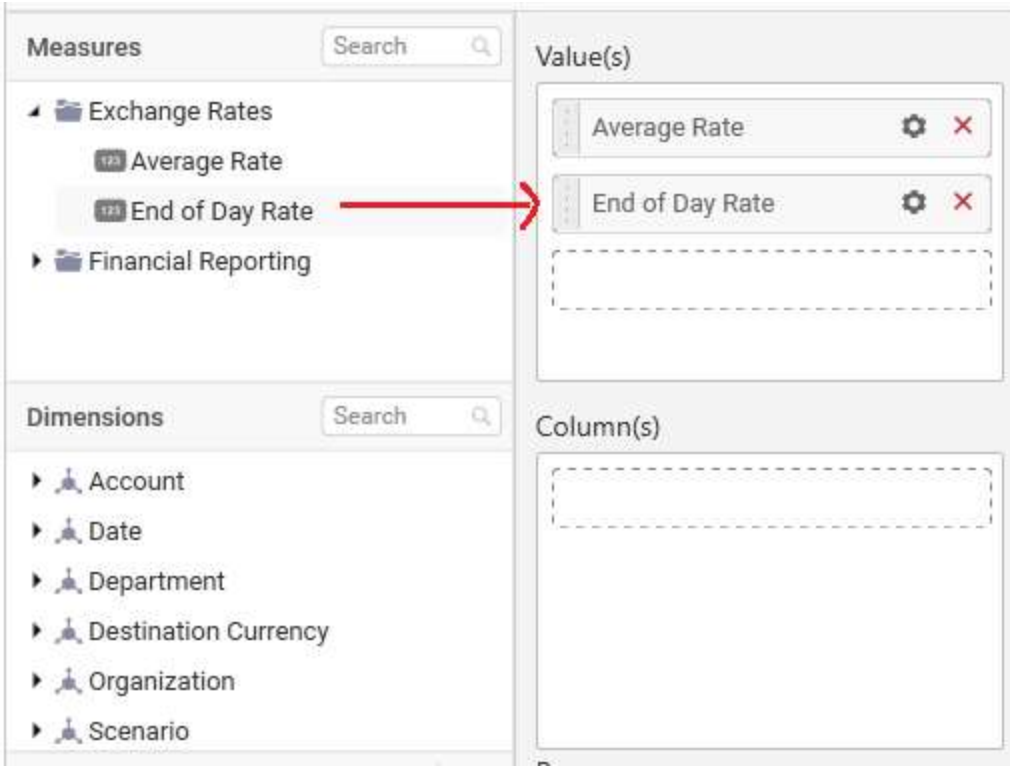
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.

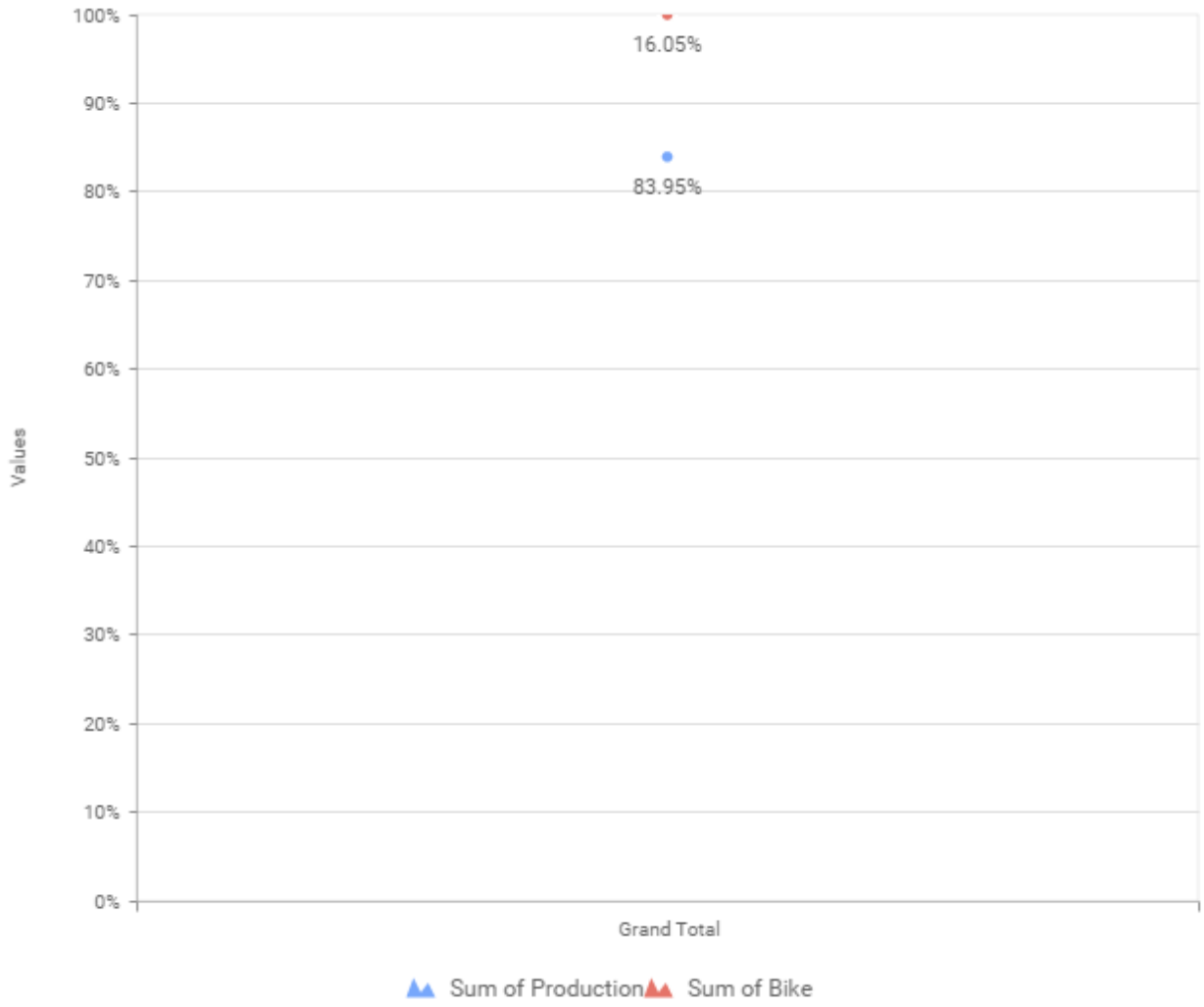


Select Clear option to clear the defined filter.



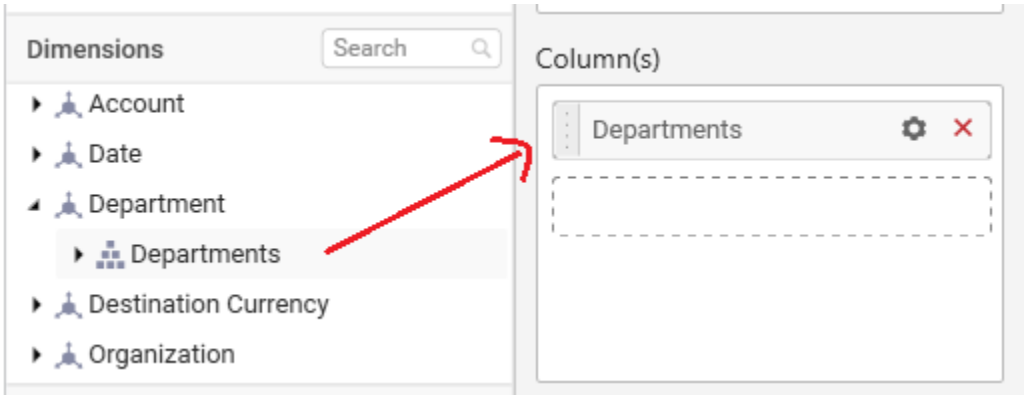
You can also add more than one column to the Value(s) section.



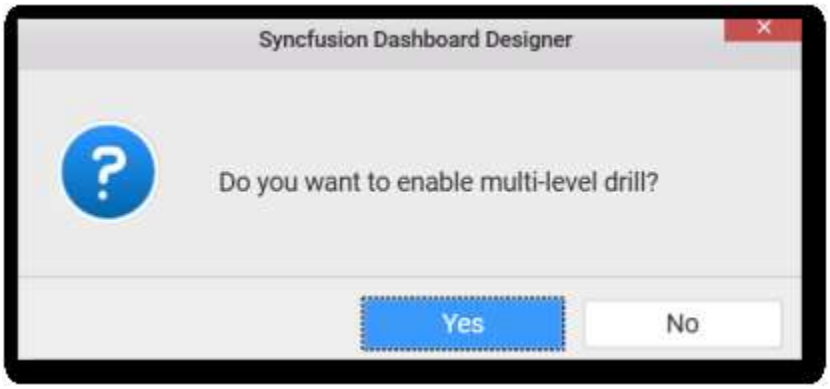


Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.

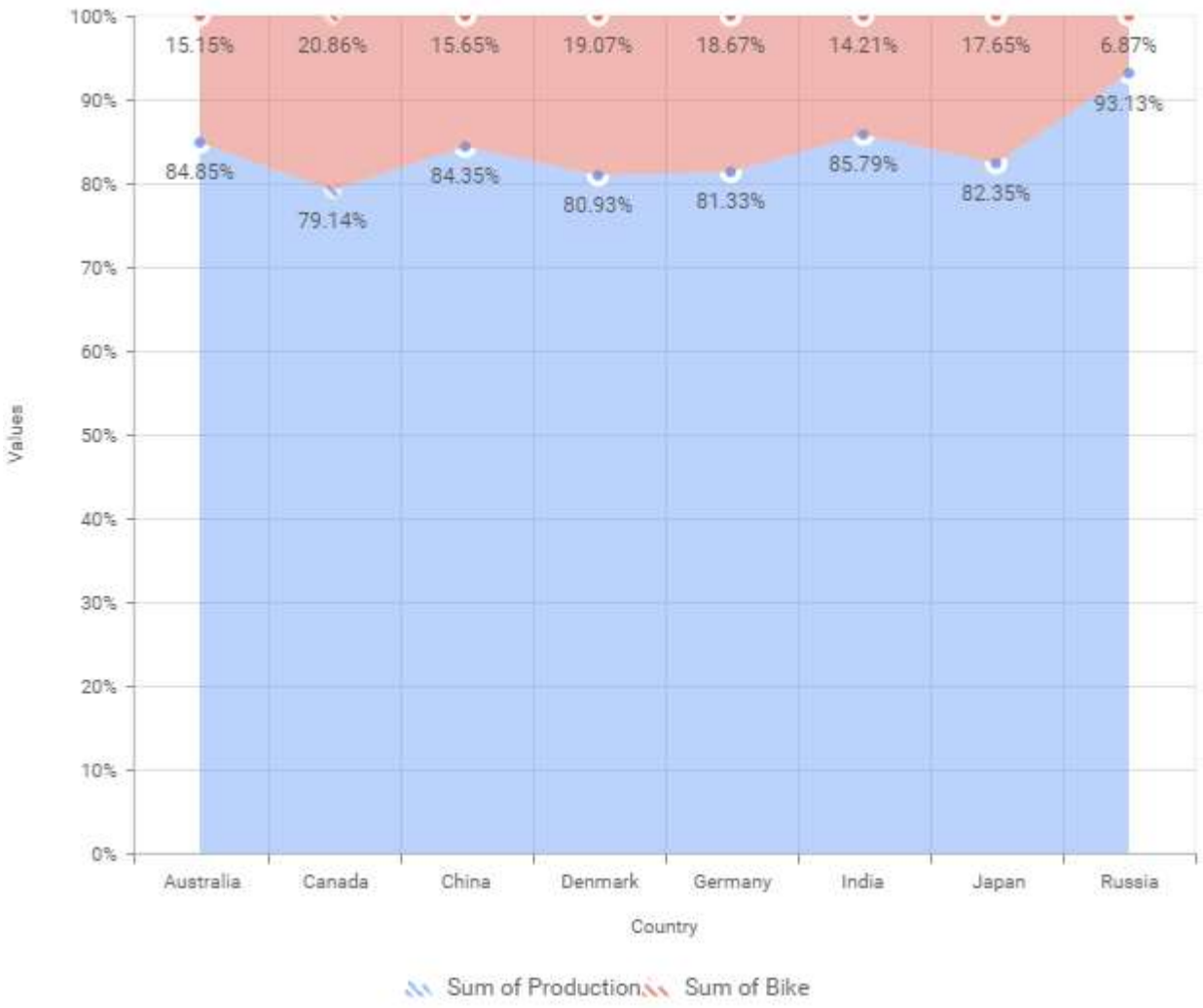


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

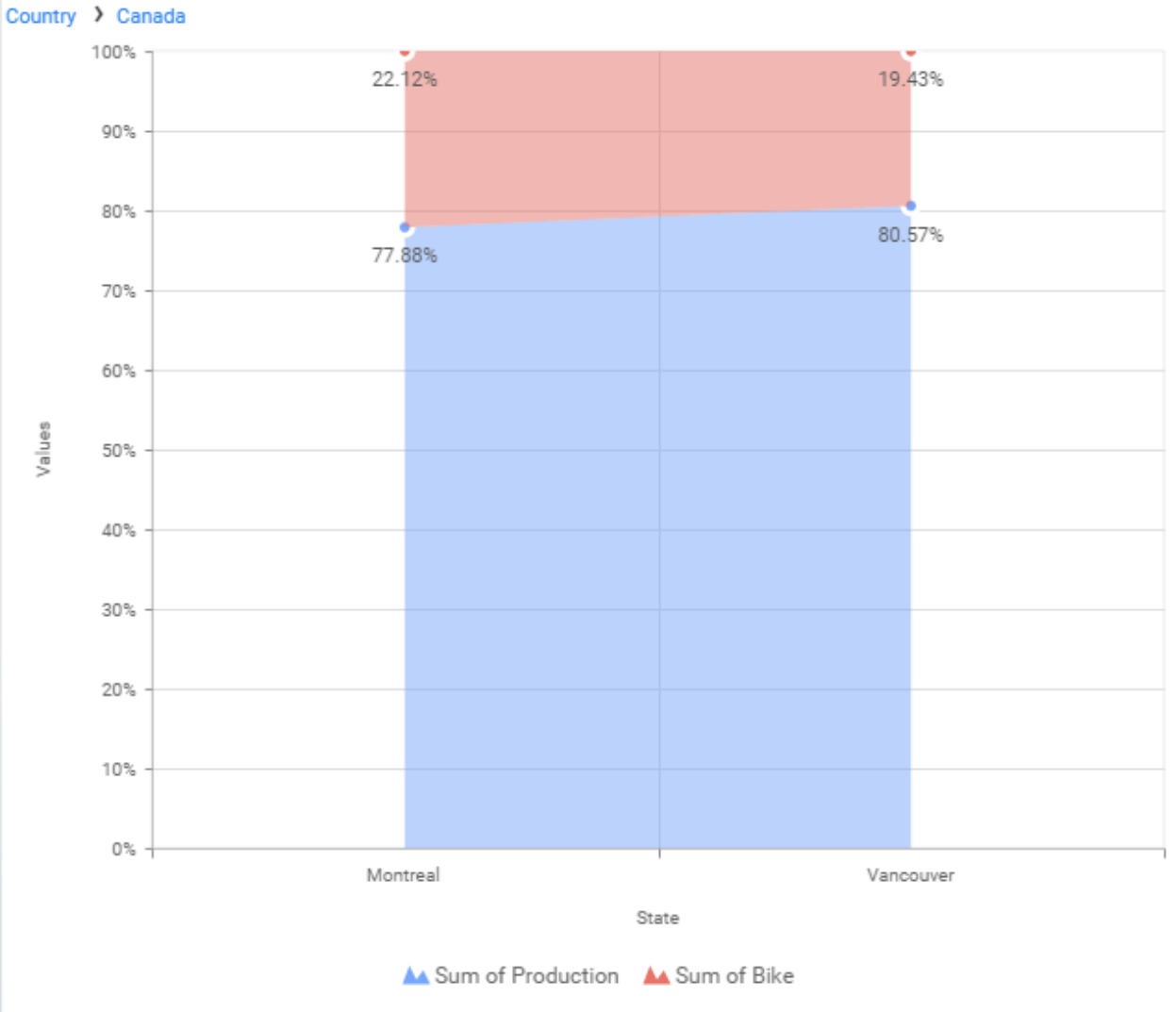


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

Click the respective data value marker in chart to drill into its inner level.

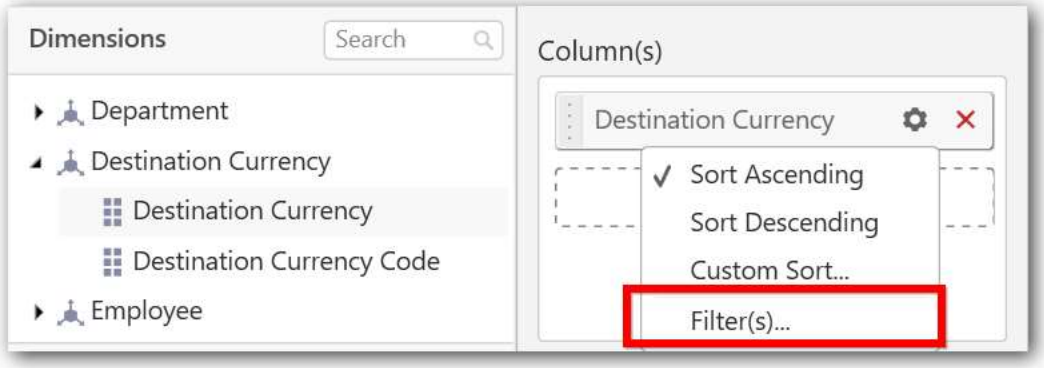


The drilled view of the chart is follows.

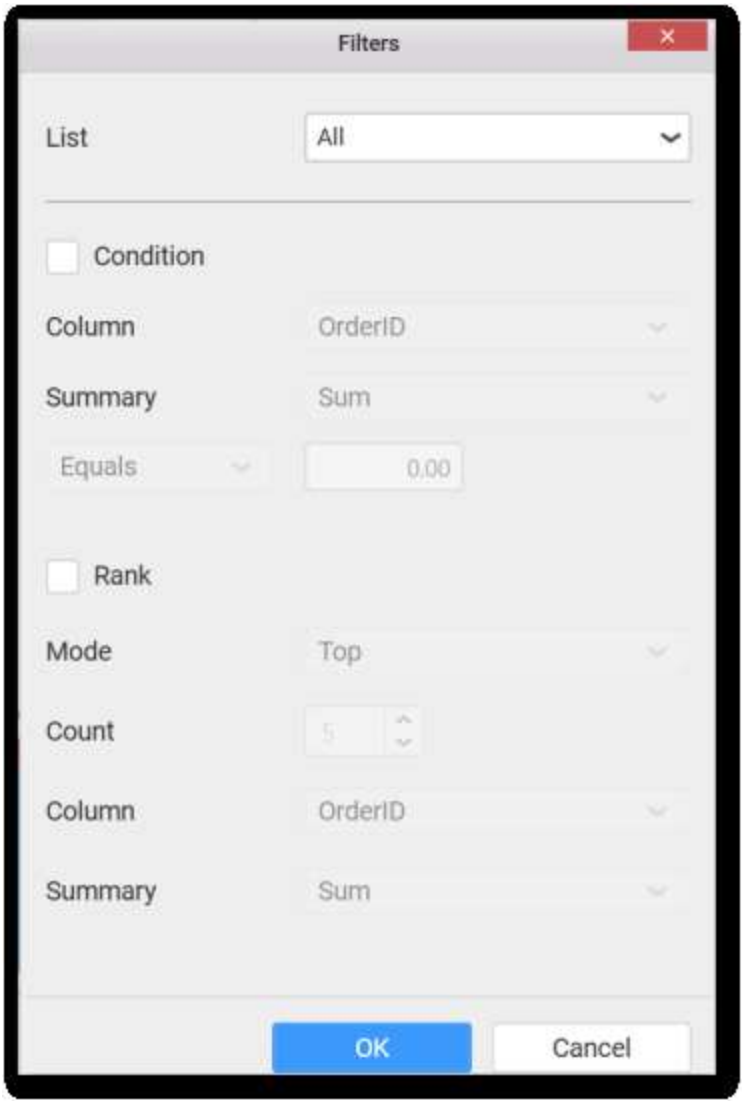


Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level.

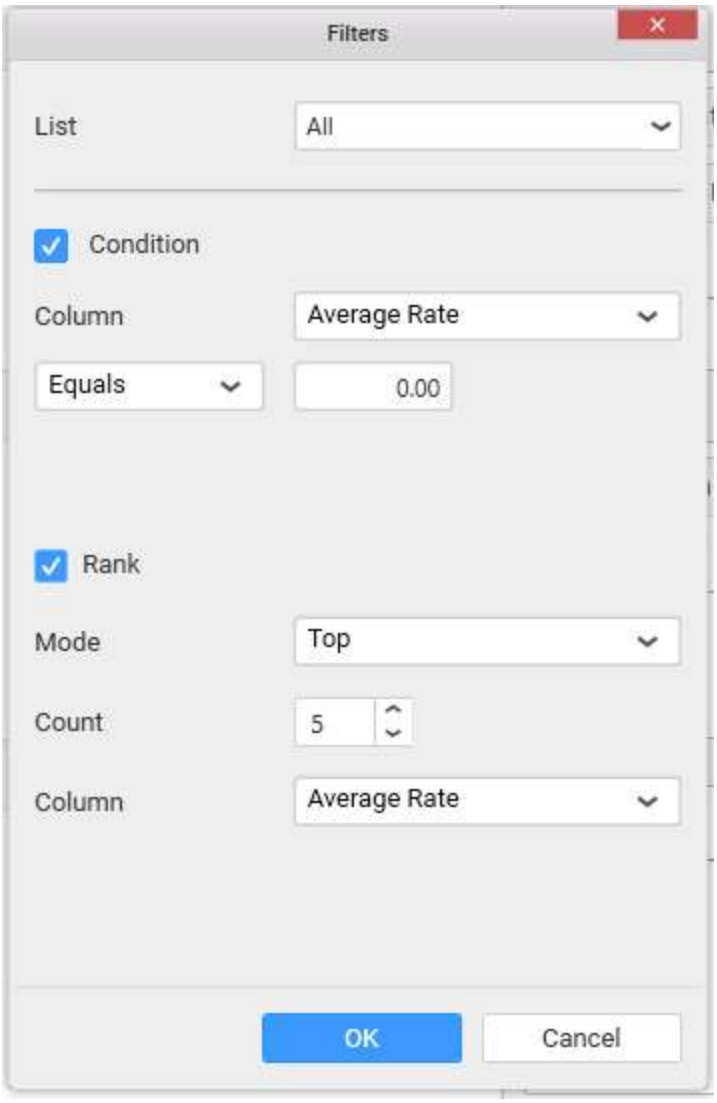
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



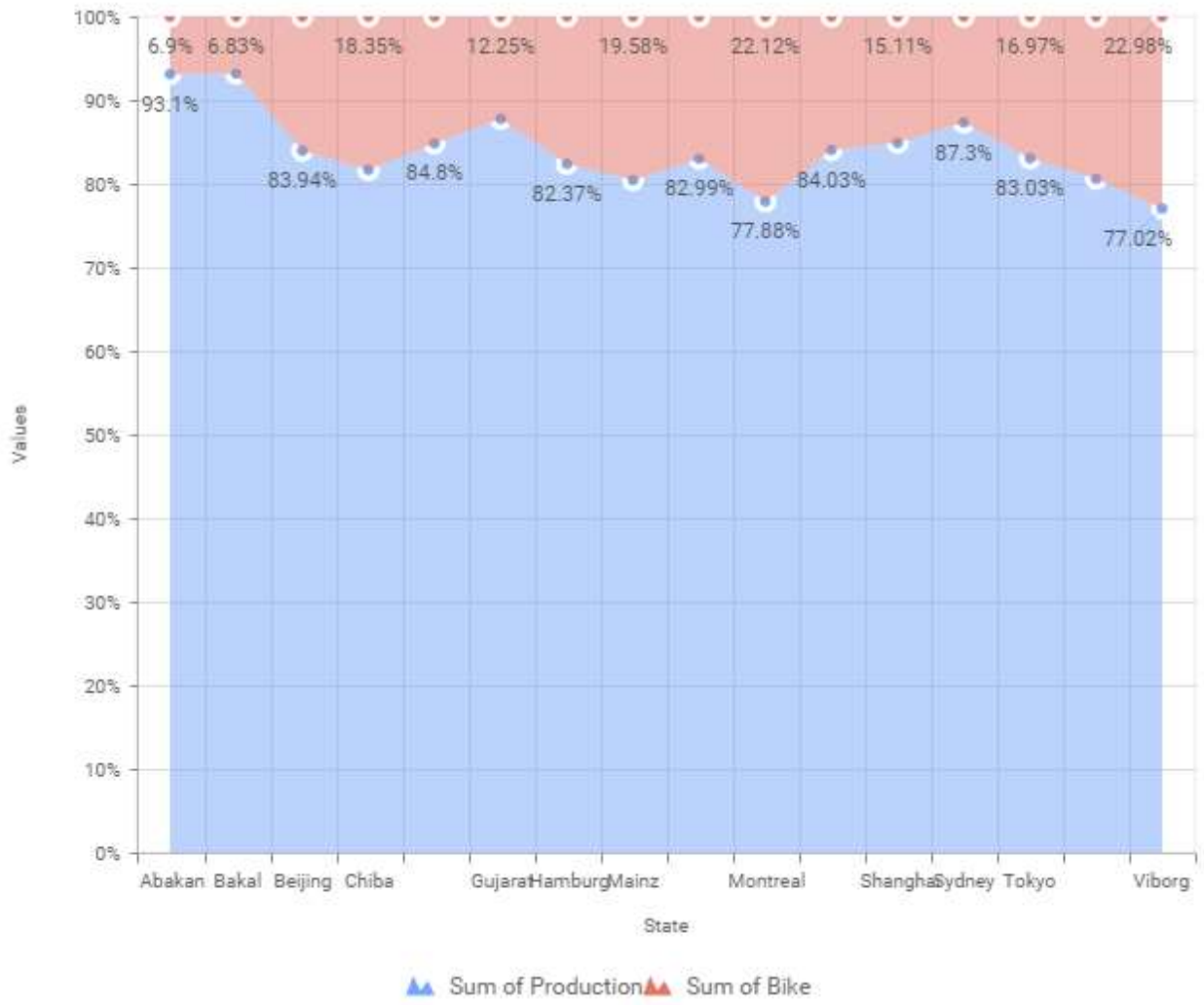
Select Filter(s)… to launch the Filters window.



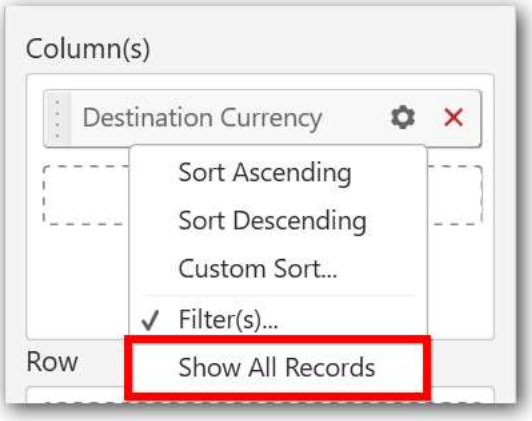
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

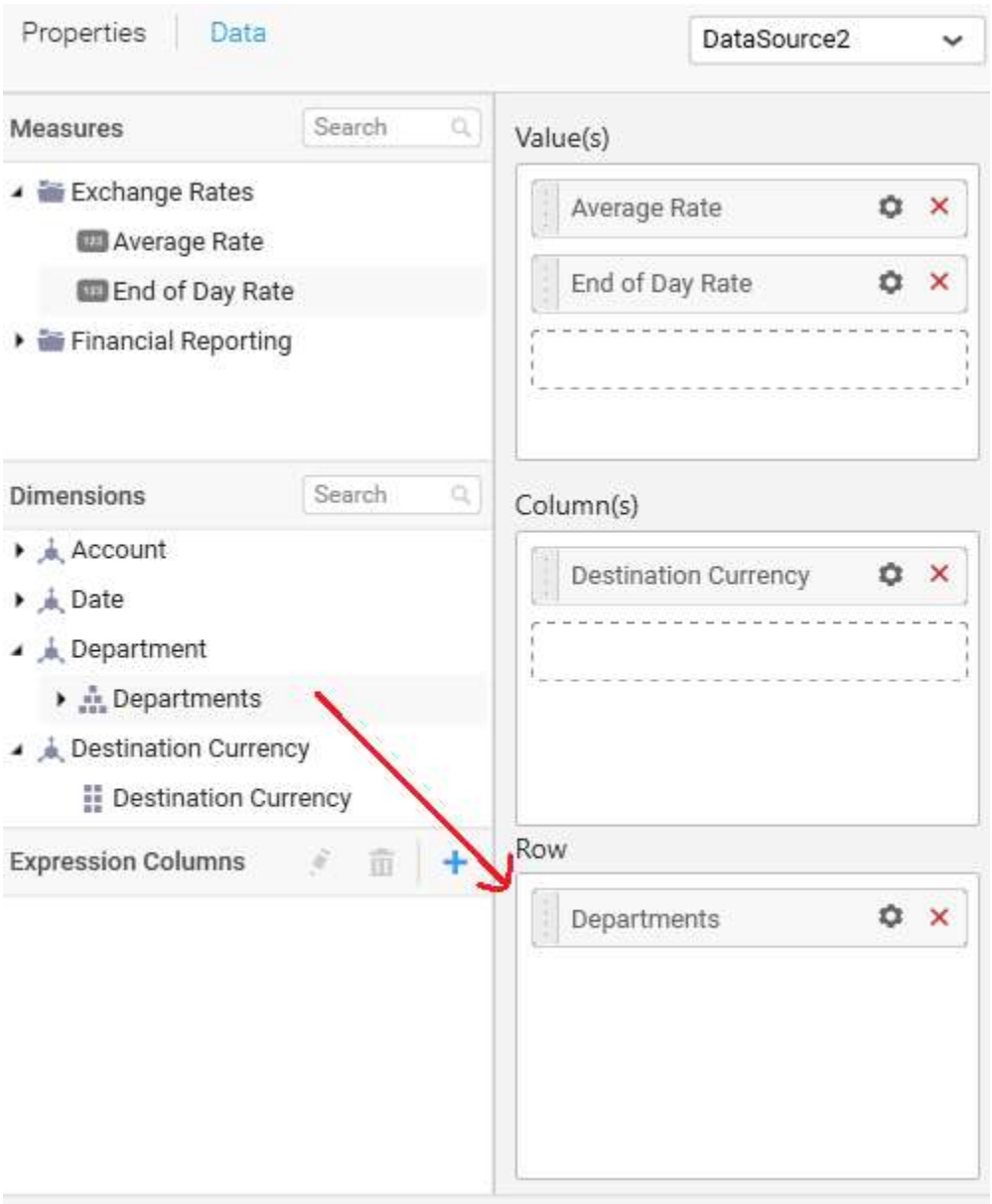


To show all records again click on Show All Records.

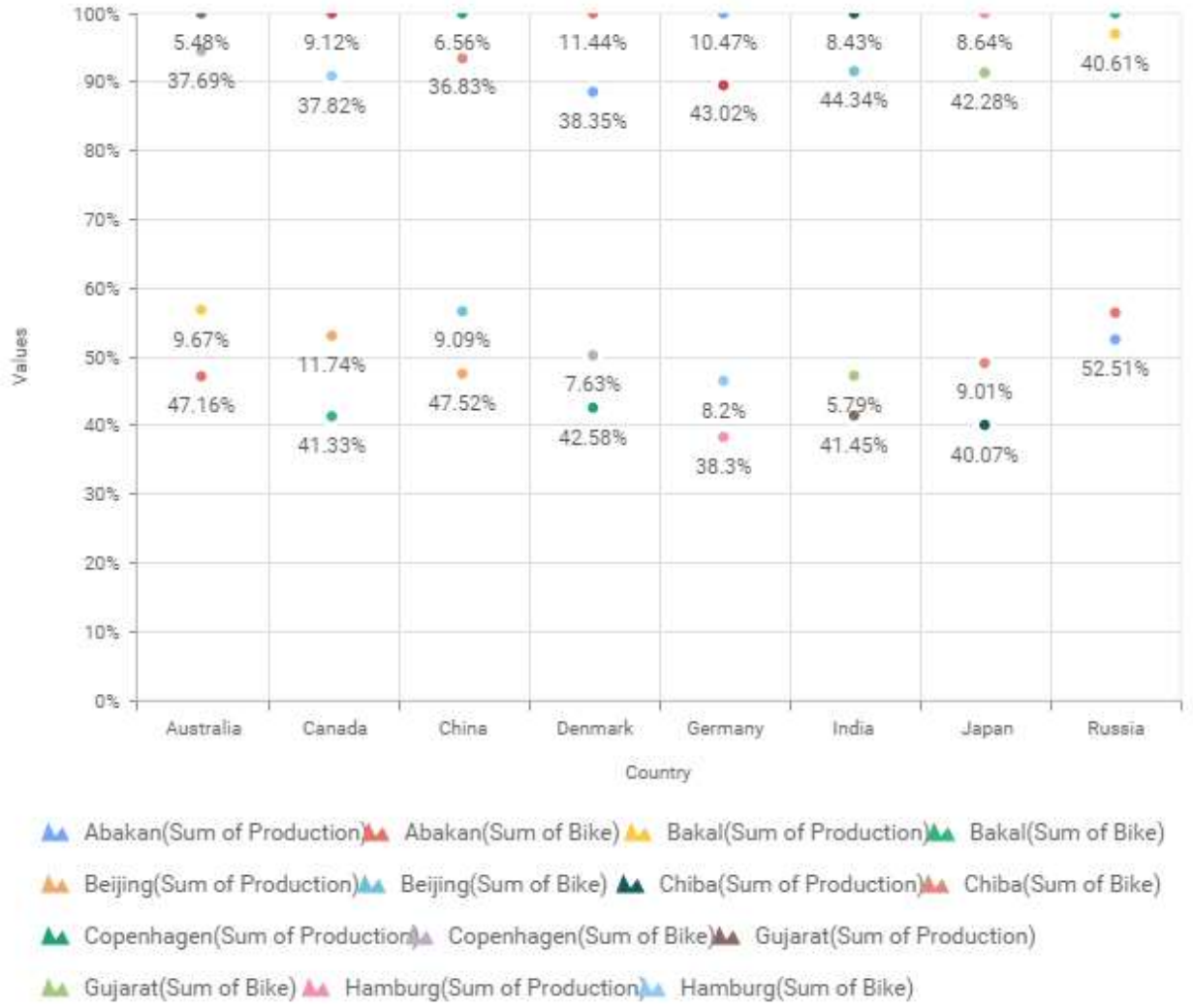


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.

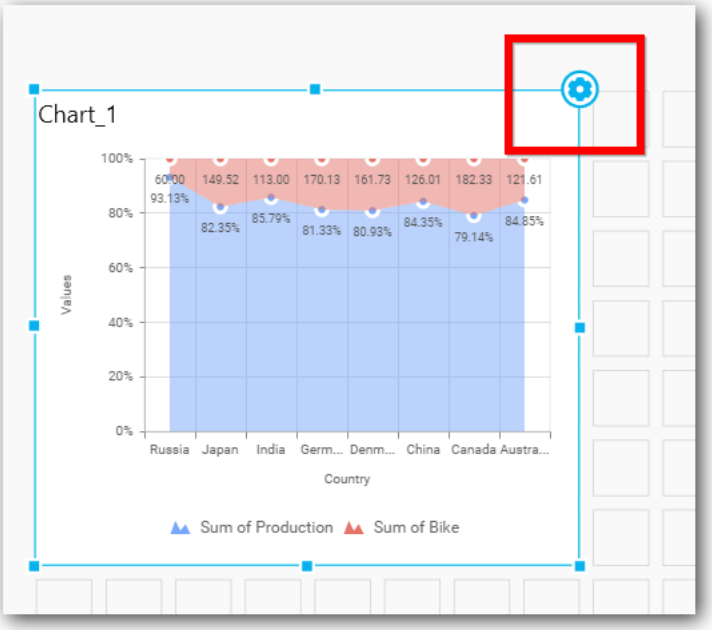


How to format 100% Stacked Area Chart?

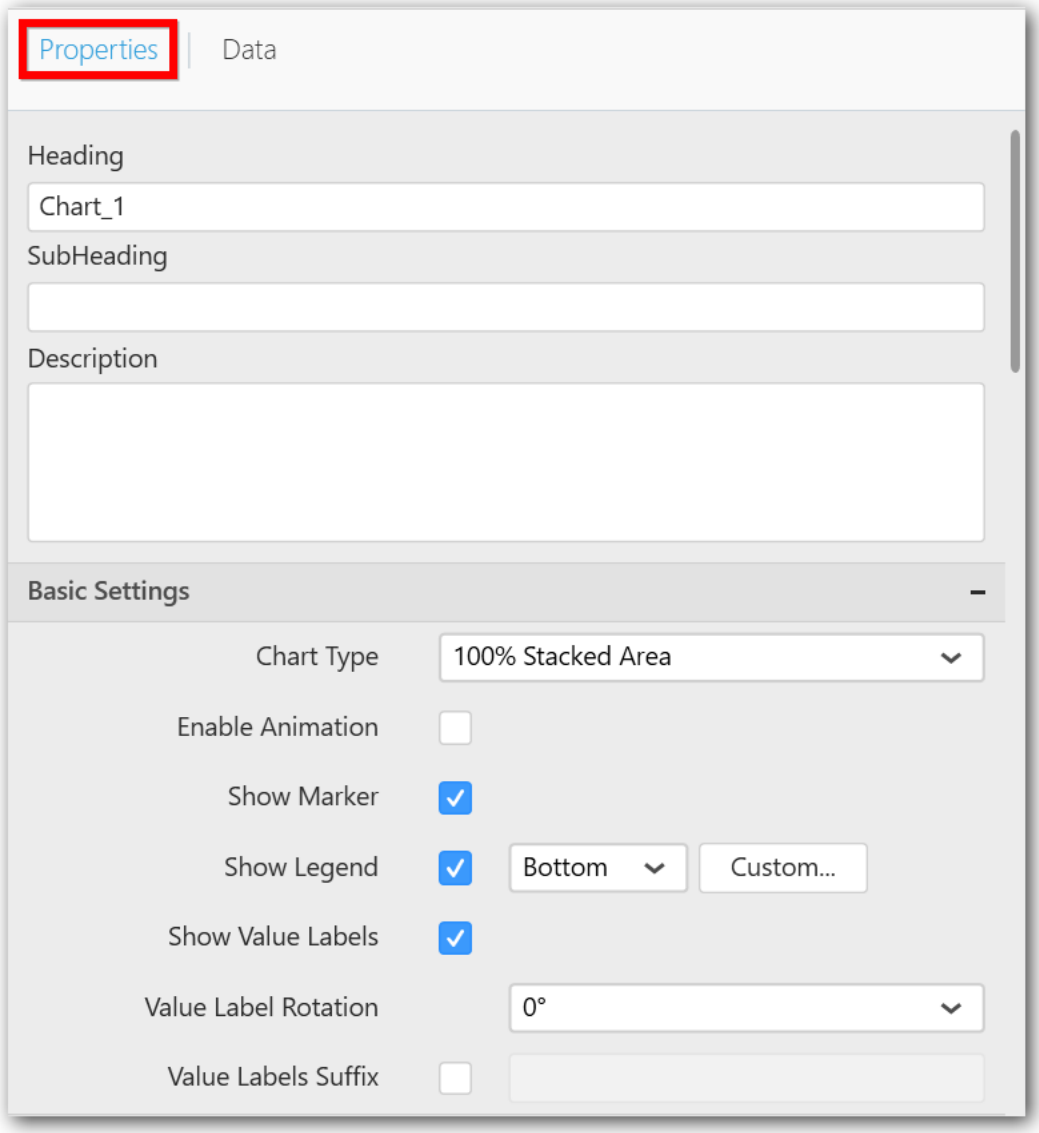
You can format the 100% stacked area chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into 100% stacked area chart follow the steps

1. Drag and drop the 100% stacked area chart into canvas and resize it to your required size.
2. Configure the data into 100% stacked area chart.
3. Focus on the 100% stacked area chart and Click on Widget Settings.

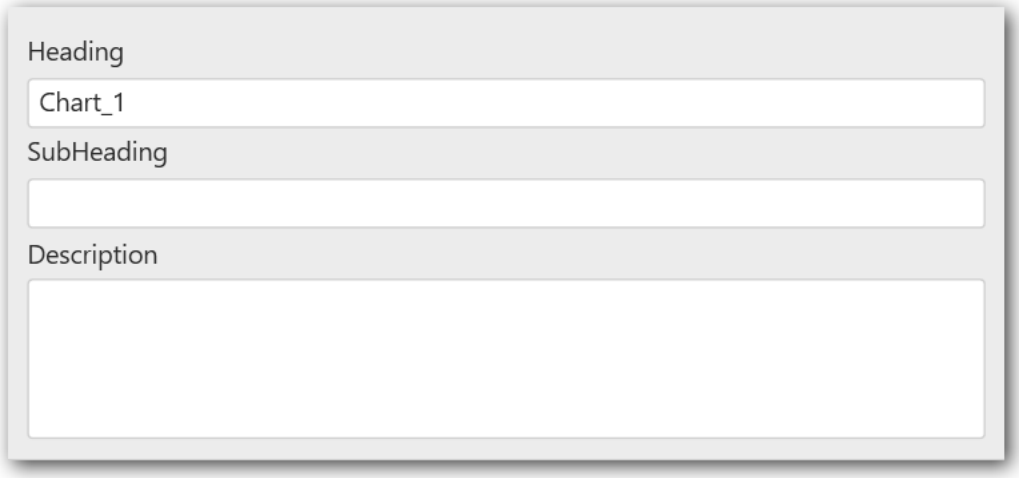


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this 100% stacked area chart widget.

SubHeading

This allows you to set sub-title for this 100% stacked area chart widget.

Description

This allows you to set description for this 100% stacked area chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

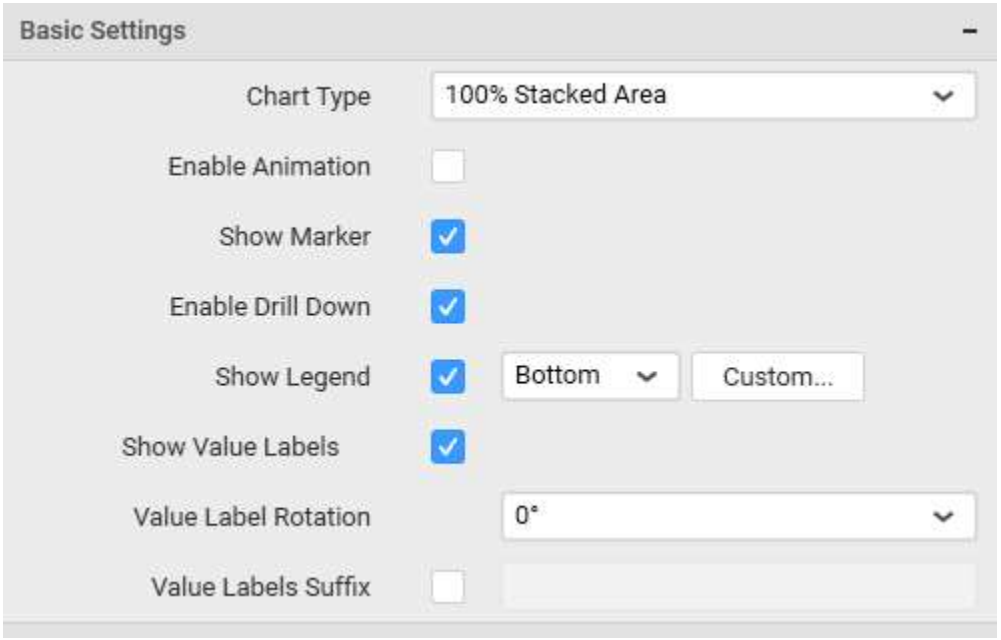


Chart Type

This allows you to switch the widget view from current chart type to another chart type. To selecting chart type through combo box.

Enable Animation

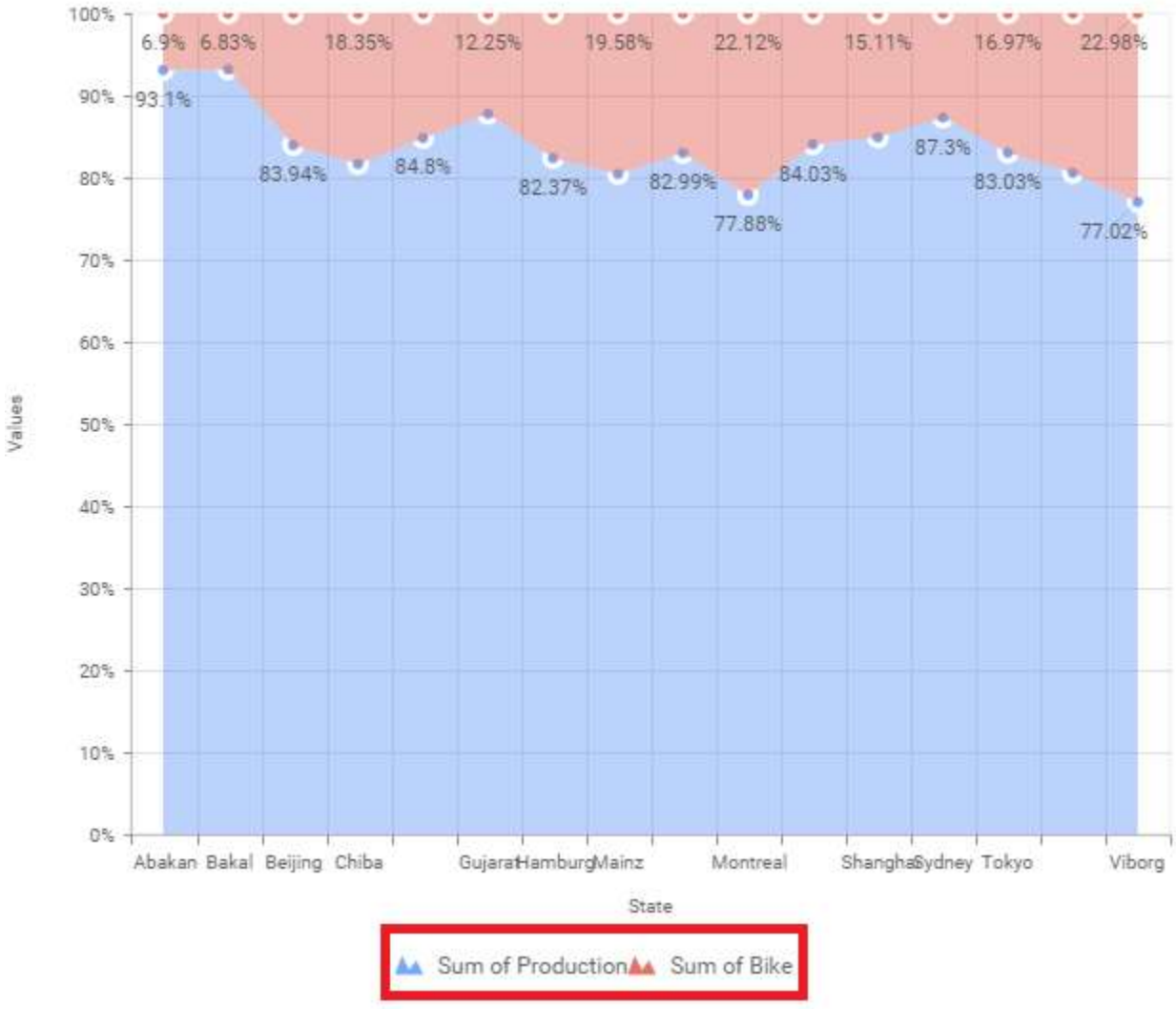
This allows you to enable the series rendering in animated mode.

Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling the option Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

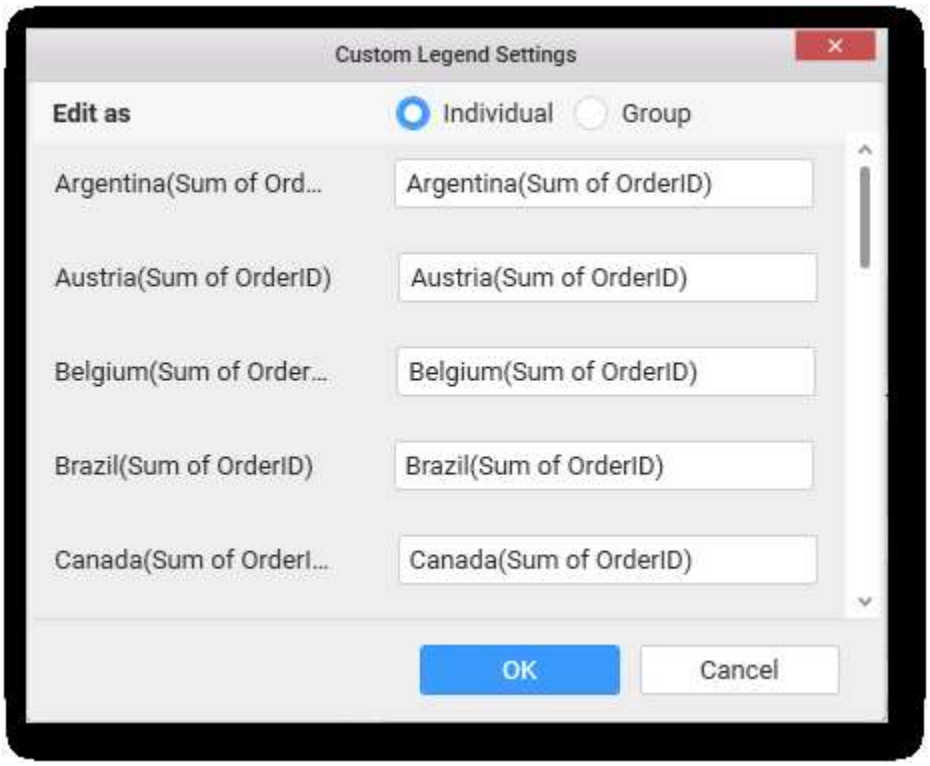
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

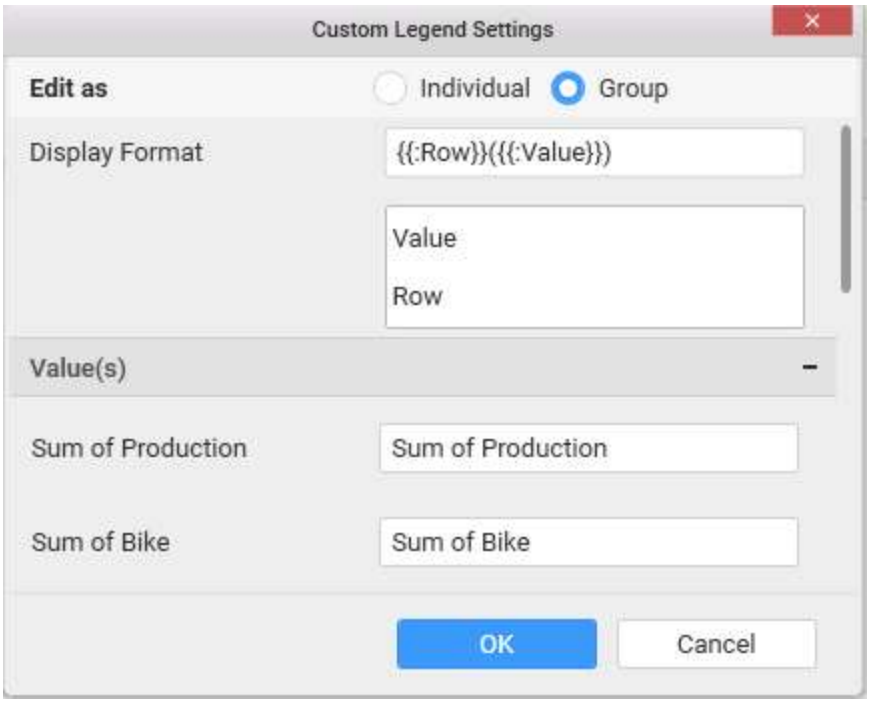
{{"{"}} : Row {{}}}} ({{"{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

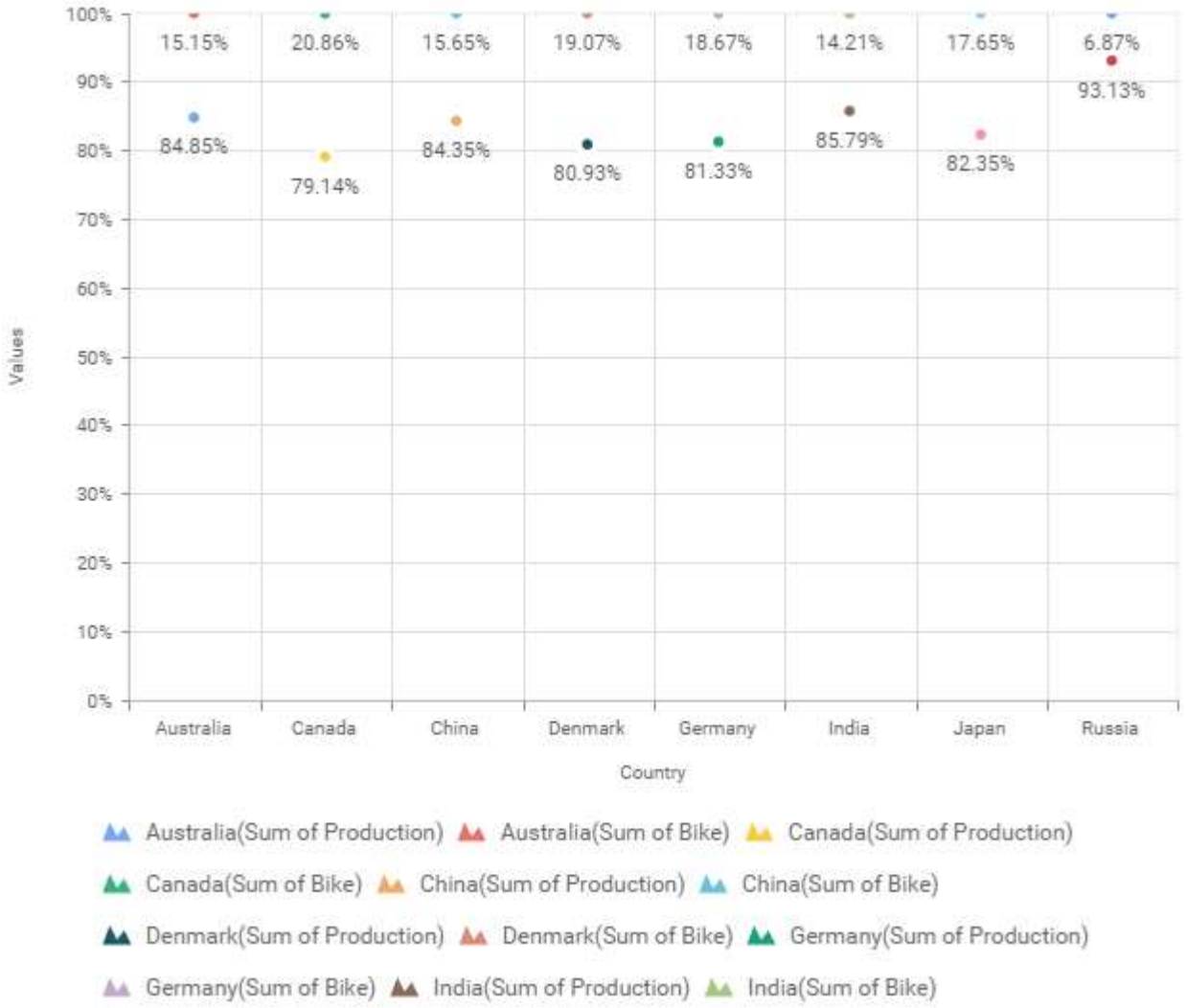


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

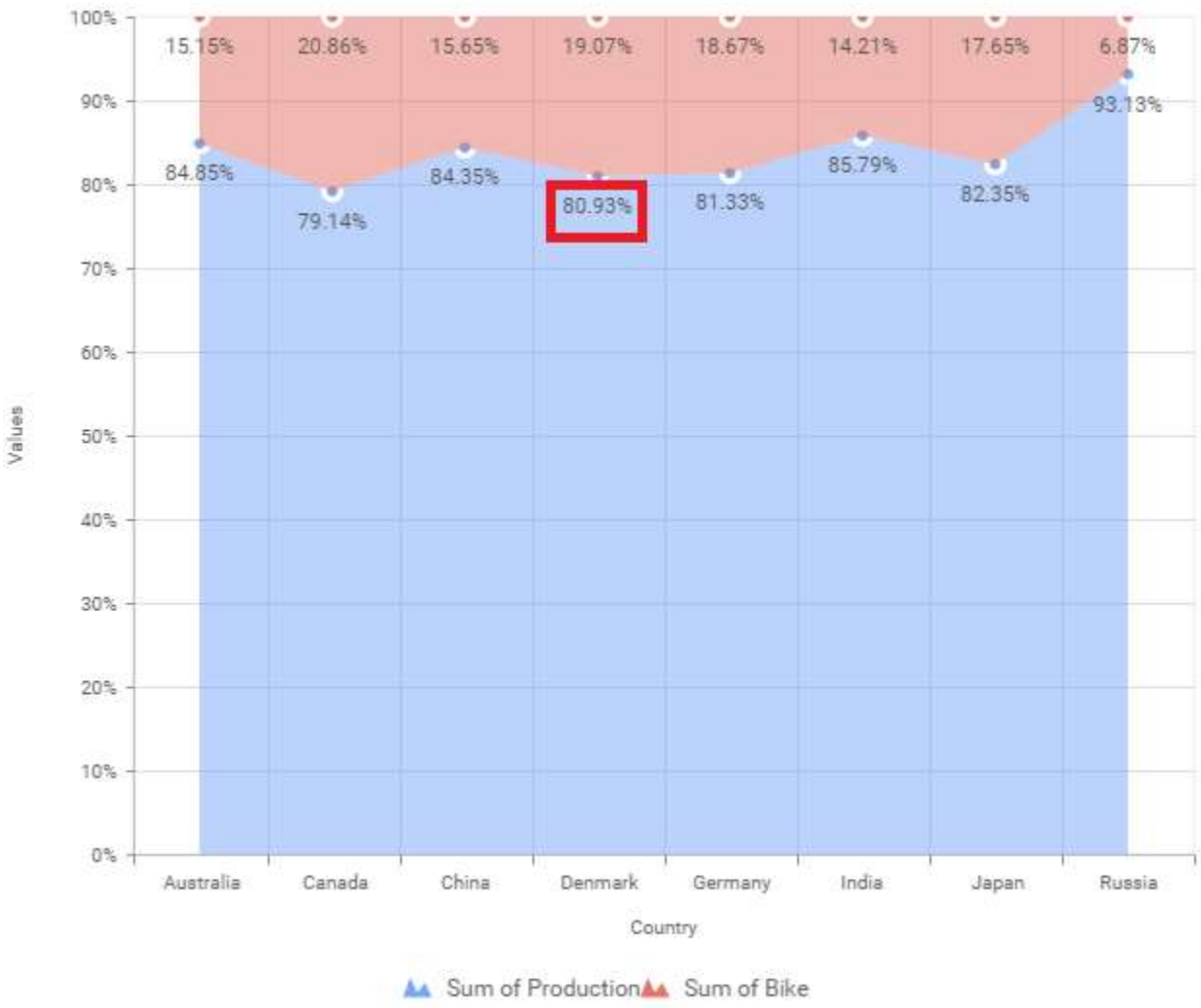


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



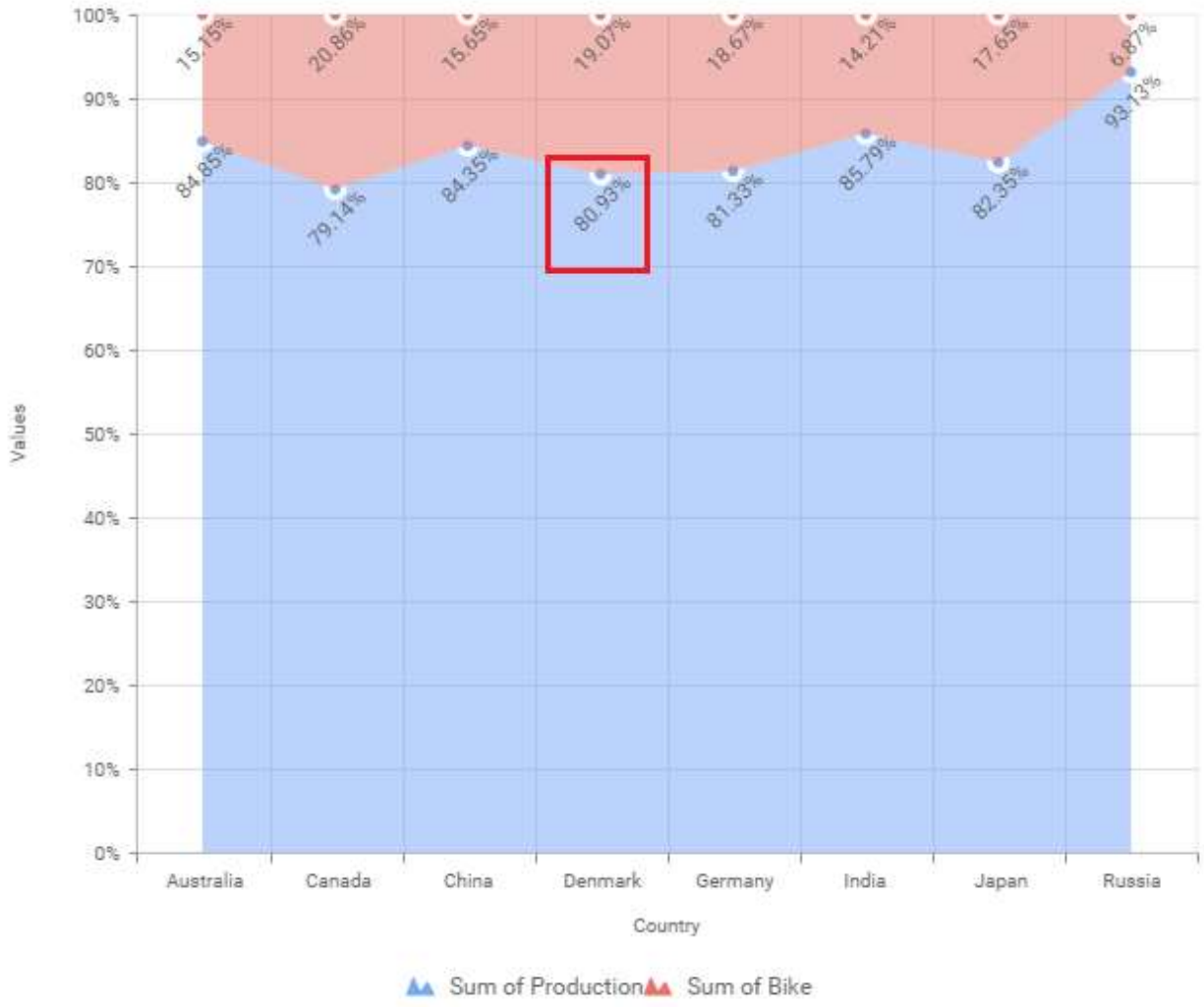
Show Value Labels

This allows you to toggle the visibility of value labels.



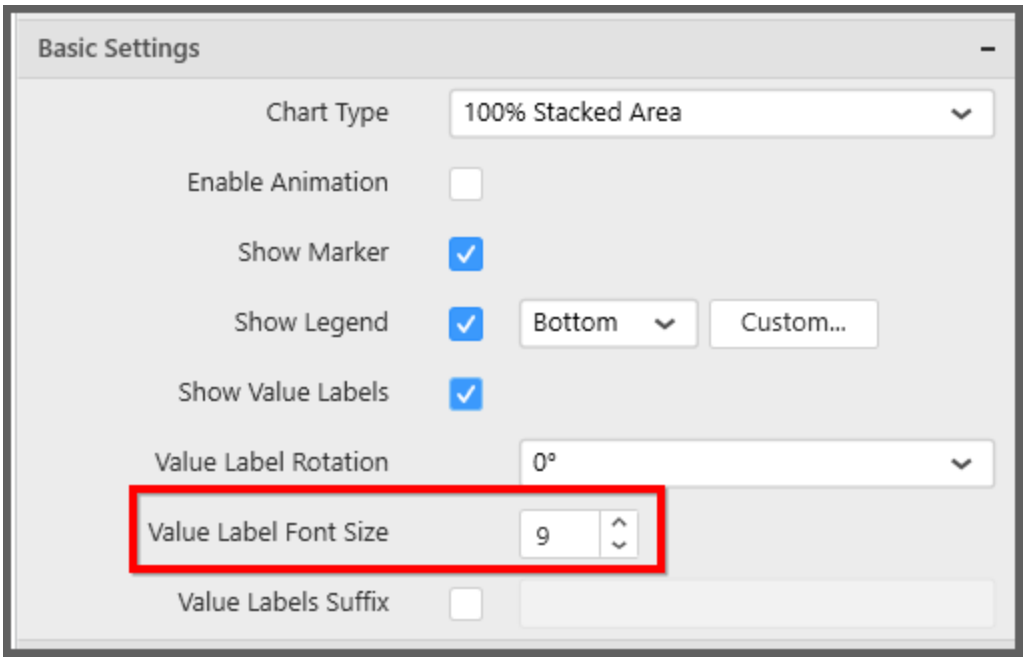
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



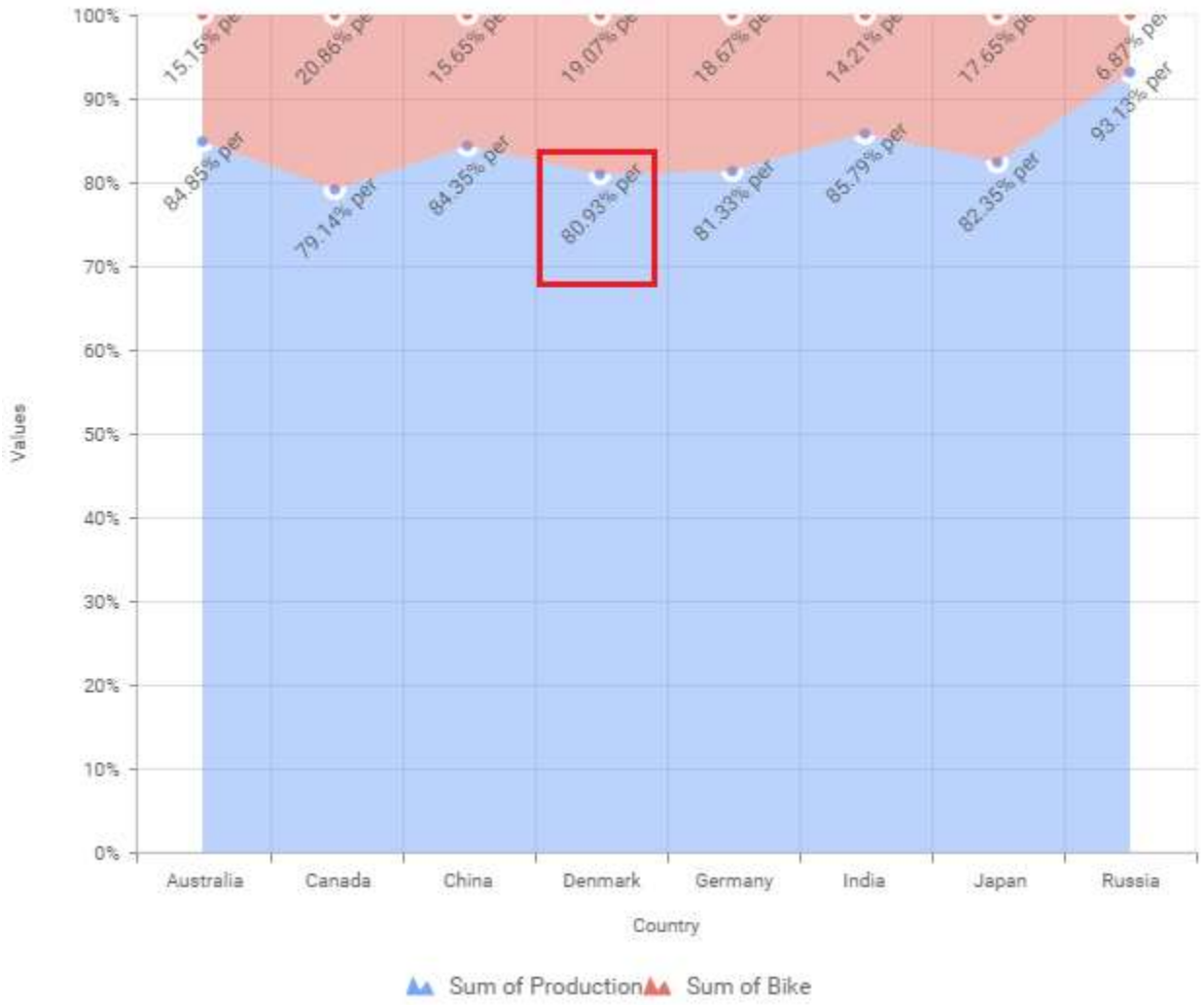
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

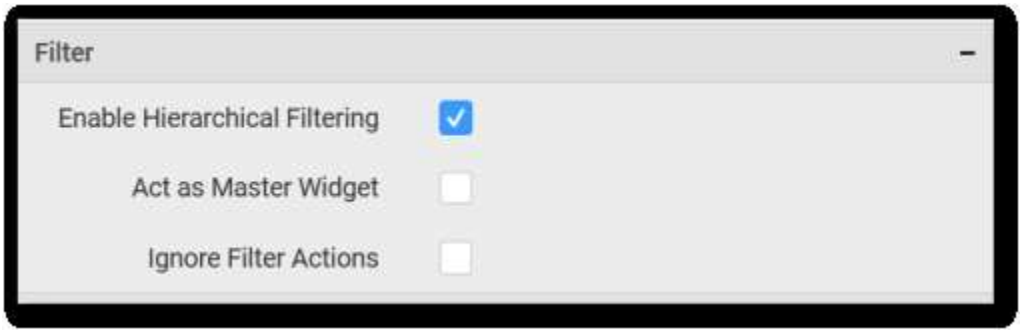


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter Settings



Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

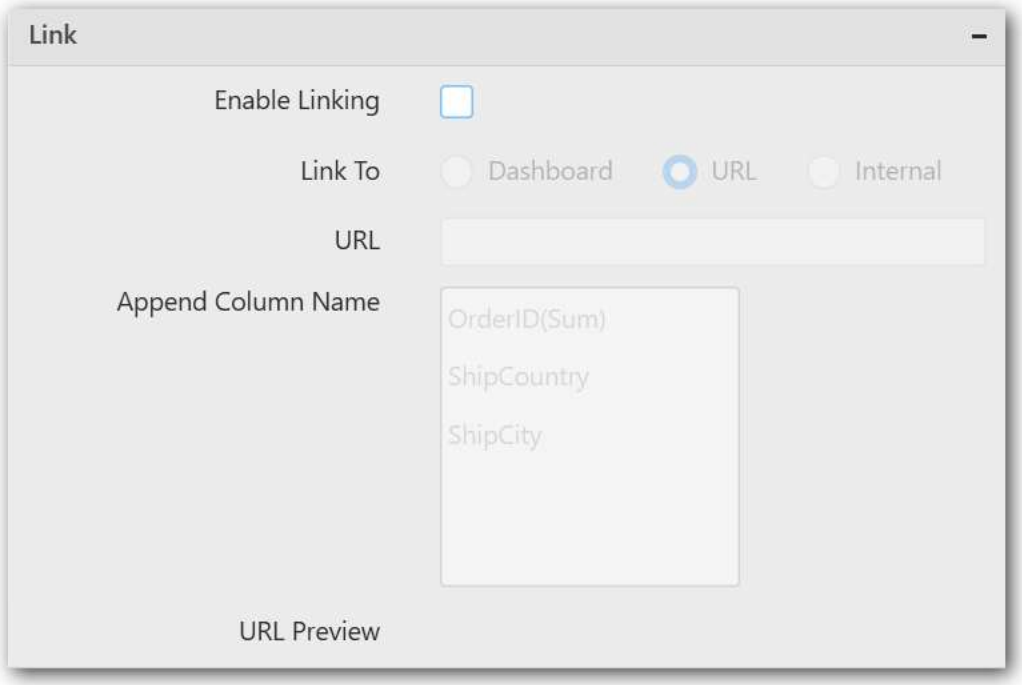
Act as Master Widget

This allows you to define this 100% stacked area chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

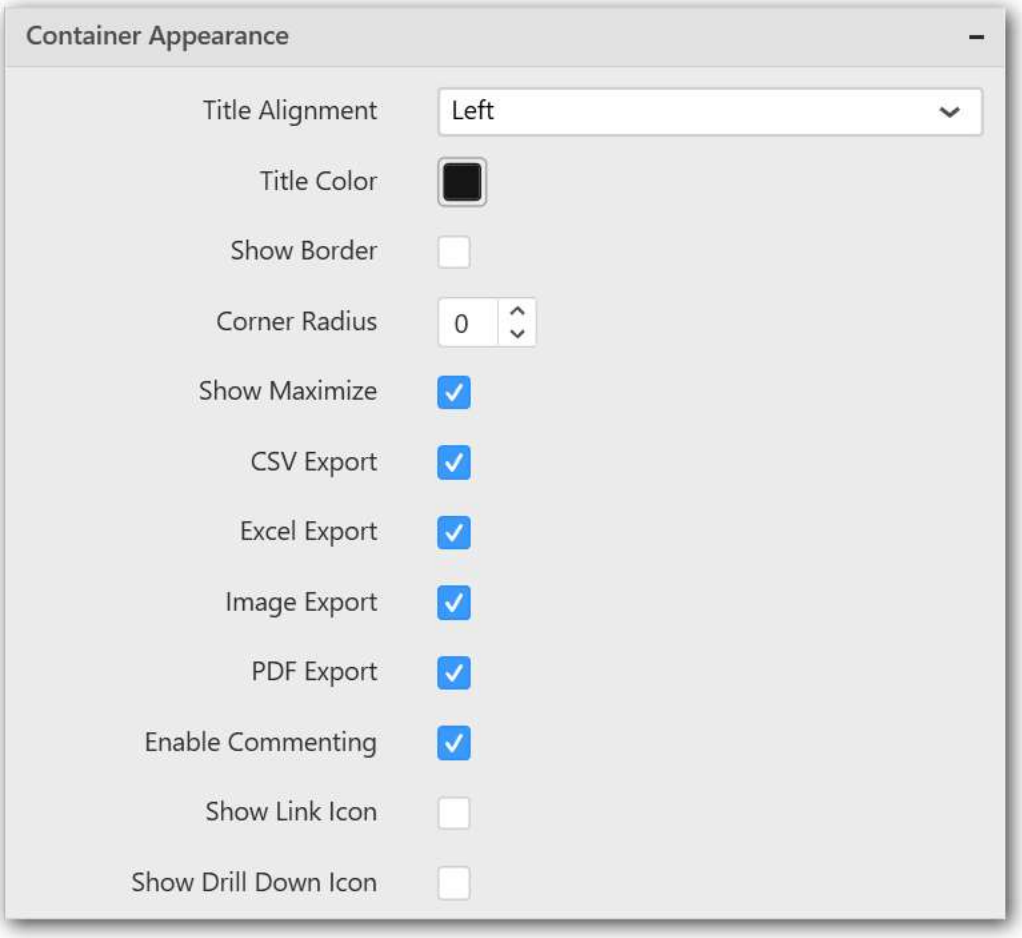
This allows you to define this 100% stacked area chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this 100% stacked area chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this 100% stacked area chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this 100% stacked area chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

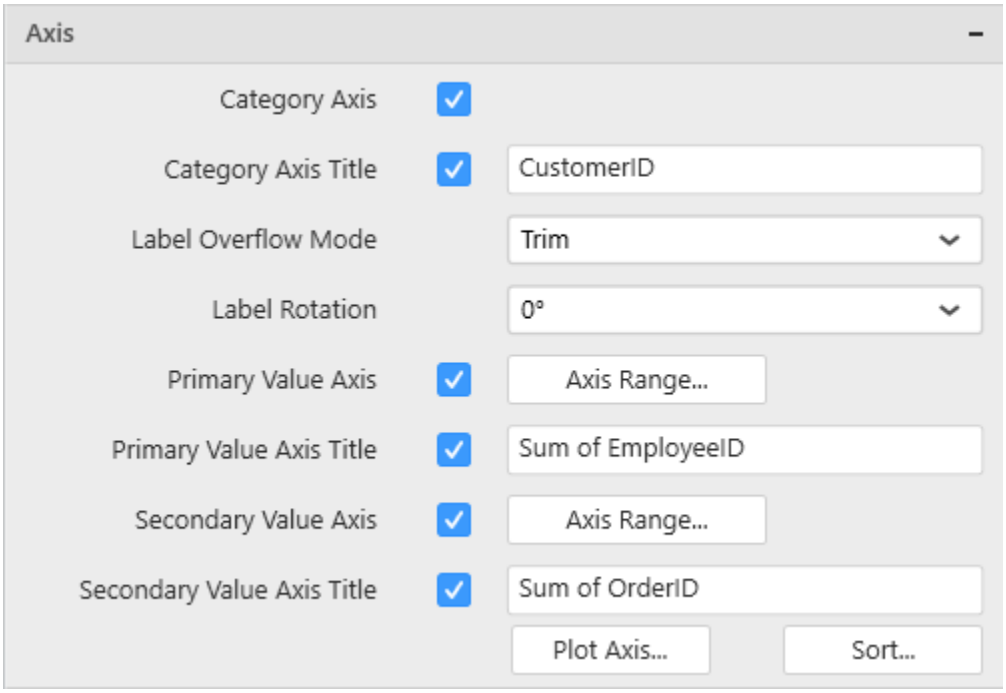
Image Export

This allows you to enable/disable the image export option for this 100% stacked area chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

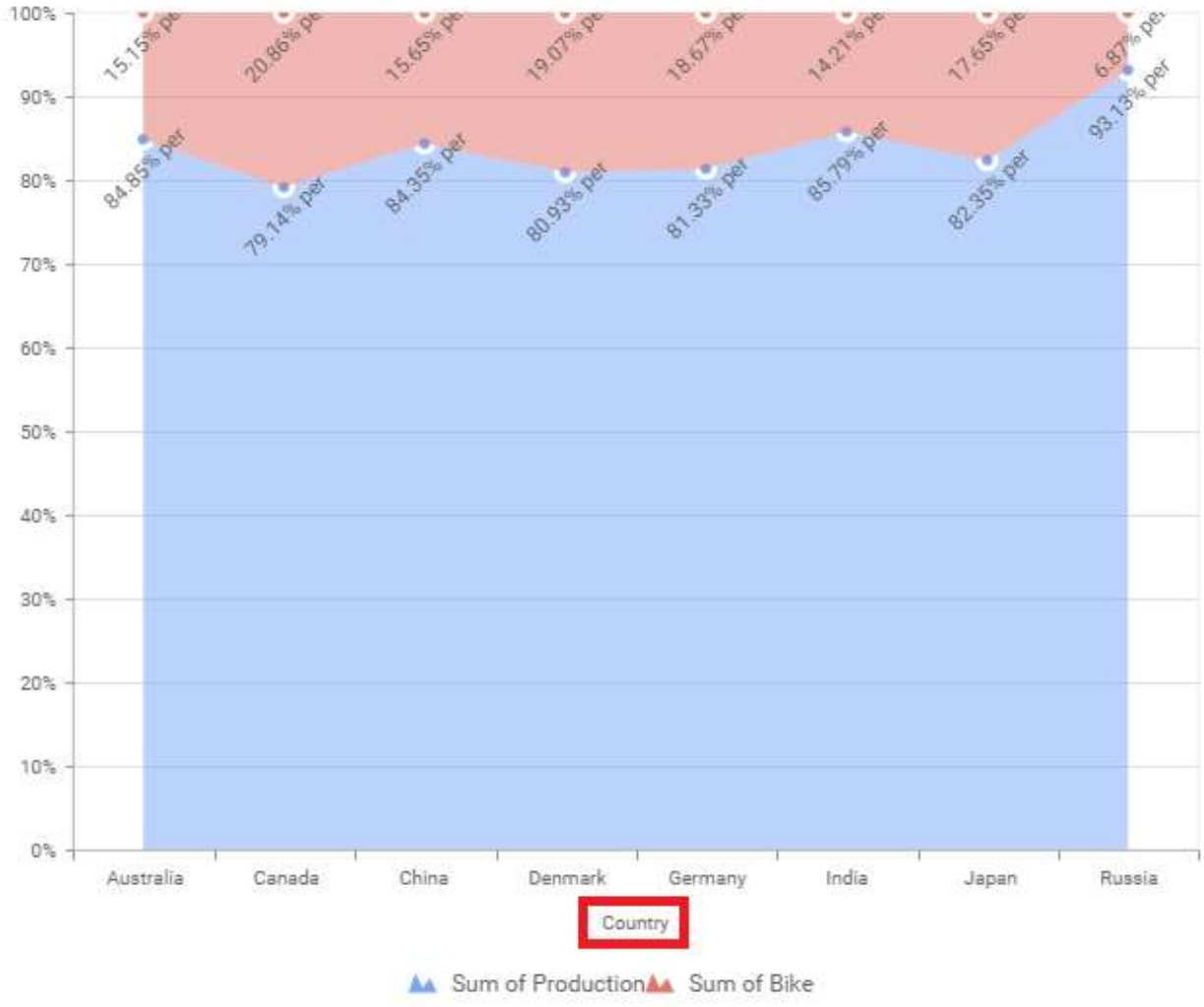
Axis Settings



This section allows you to customize the axis settings in chart.

Category Axis

This allows you to enable/edit the Category Axis title. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

Label overflow mode

This allows you to handle the display mode of the overlapping labels.

Trim

This option trims the end of overlapping label in the axis.

Hide

This option hides the overlapping label in the axis.

Wrap

This option wraps the lengthy label text in the axis.

Label Rotation

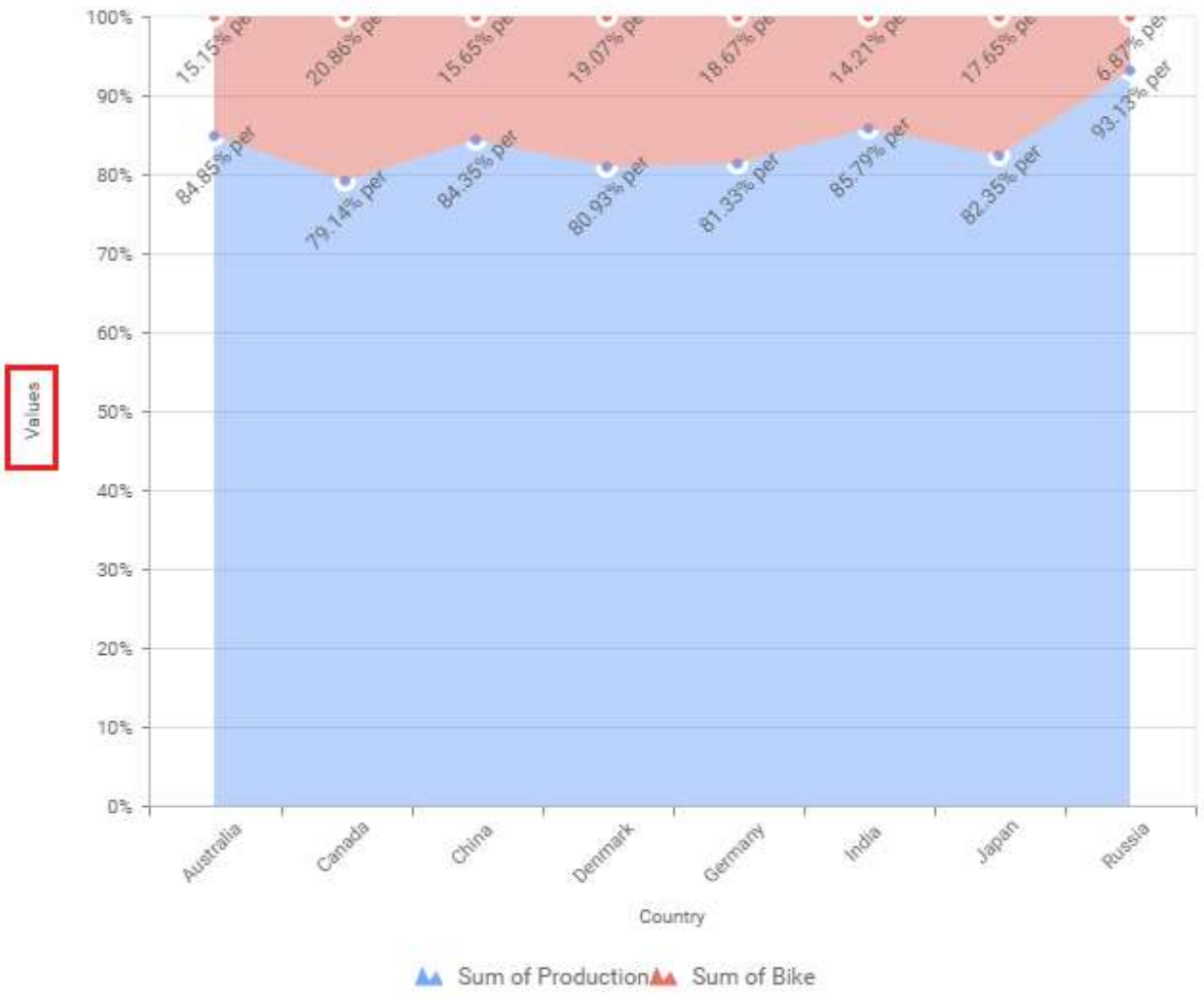
This allows you to define the rotation angle for the category axis labels to display.

Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.

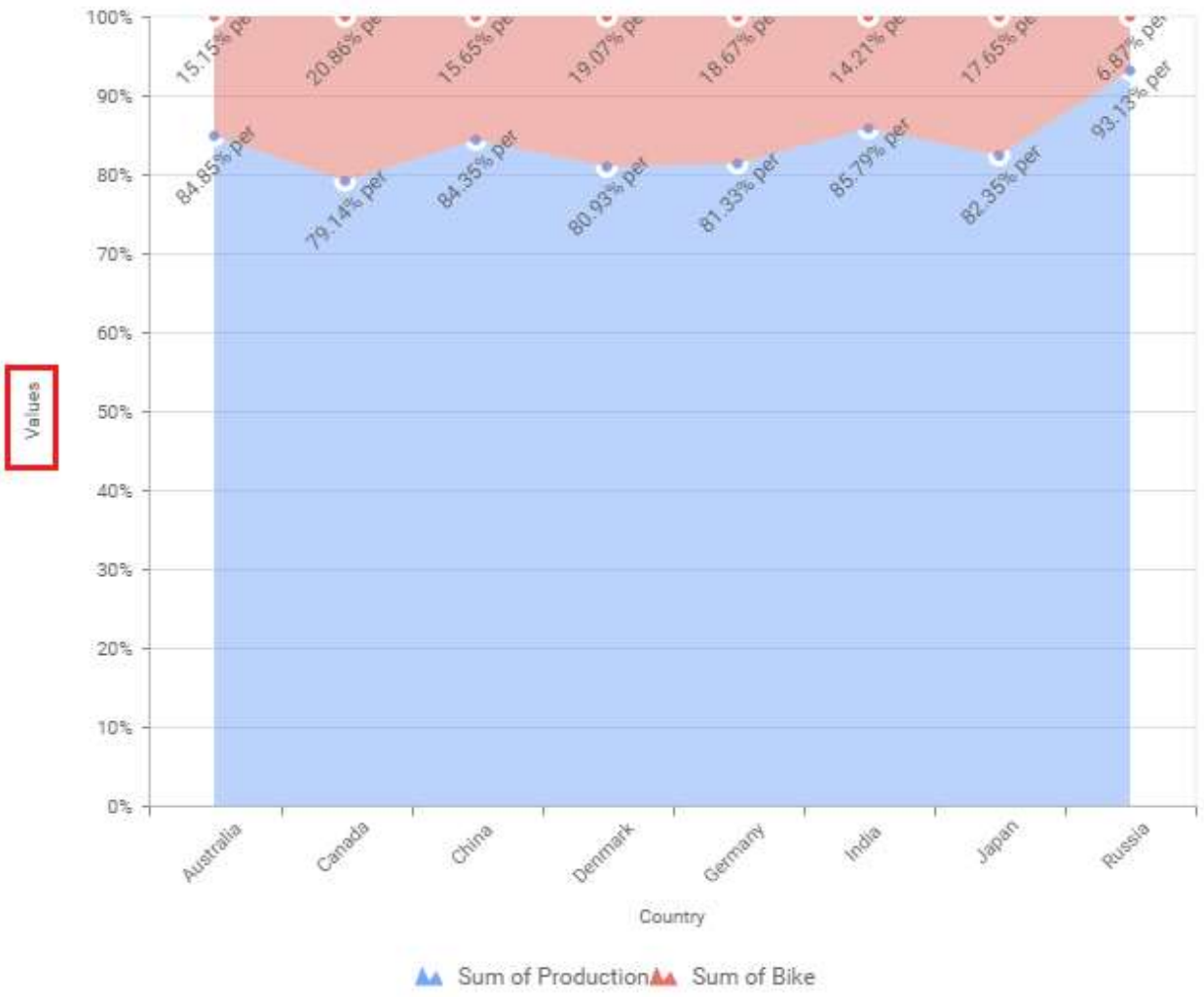
Primary Value Axis

This allows you to enable/edit the Primary Value Axis title. It will reflect in chart area y-axis name.



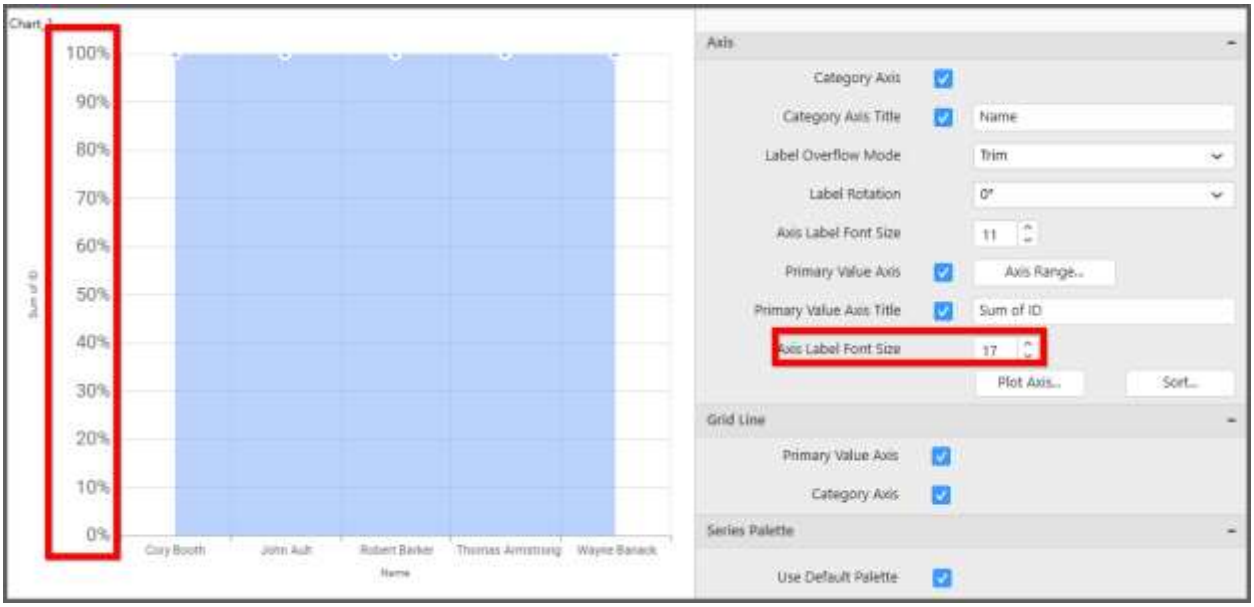
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

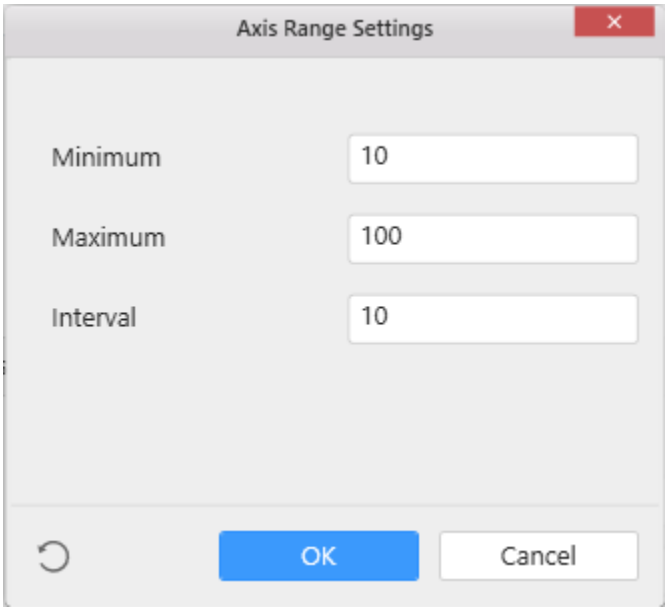


Primary Value Axis Range

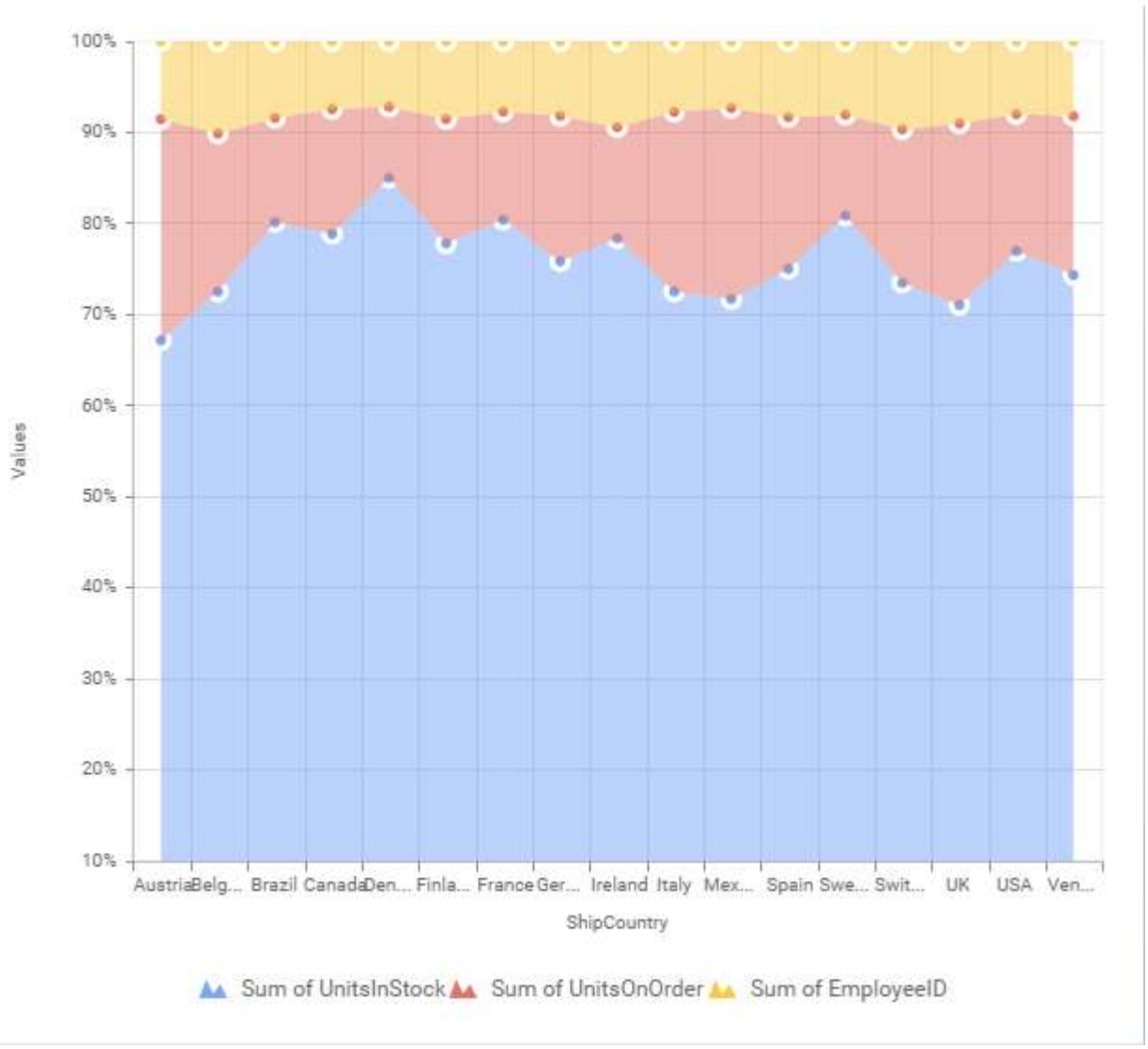
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.



Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



Secondary Value Axis

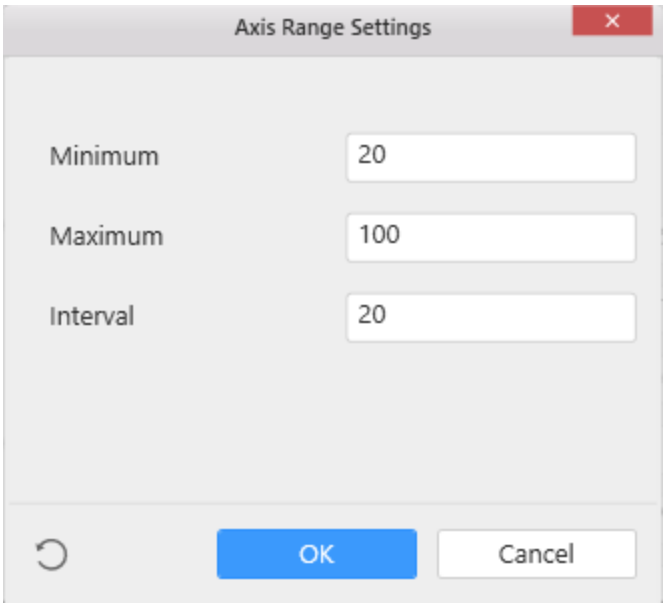
This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.

Secondary Value Axis Title

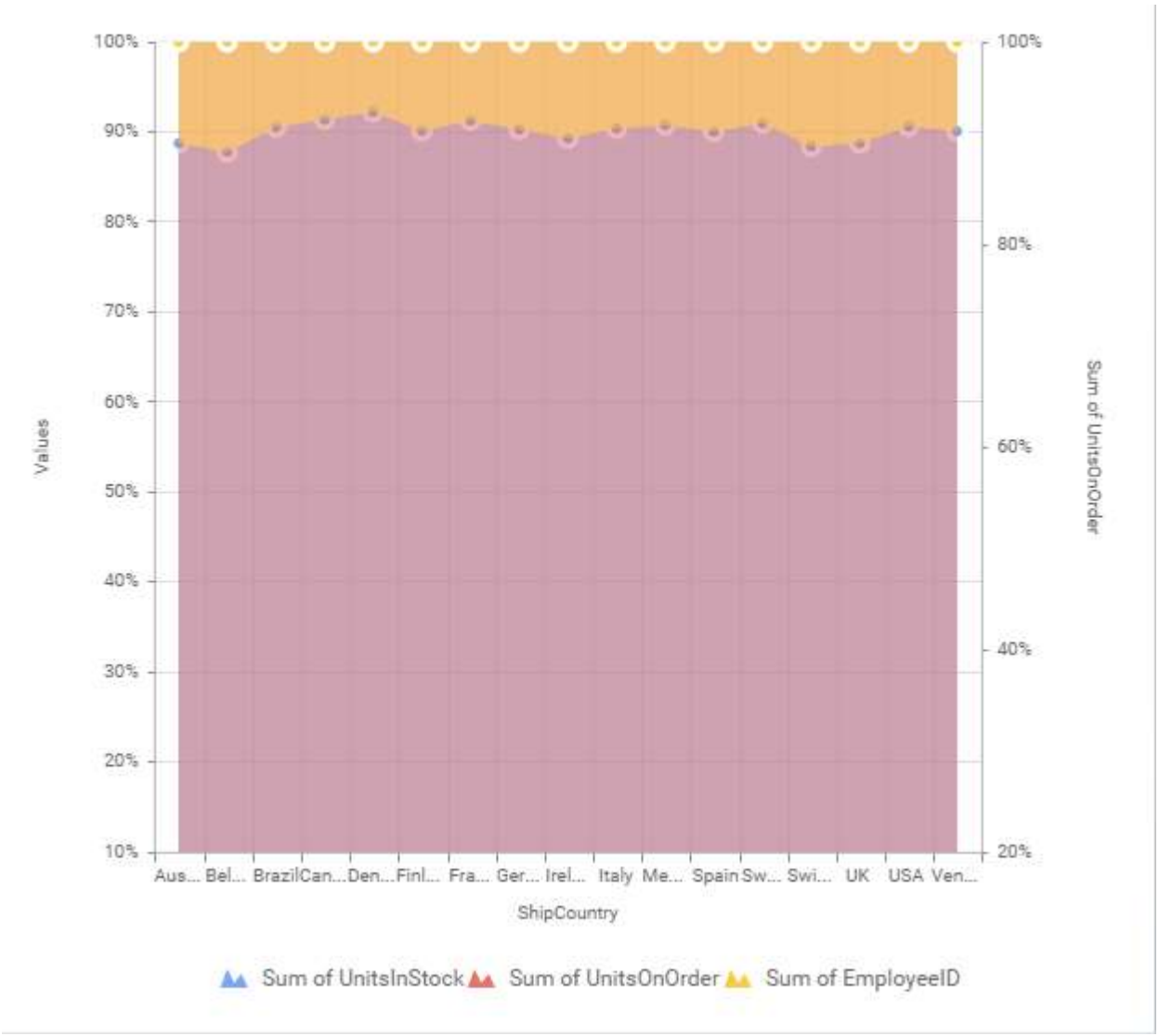
This allows you to toggle the visibility of secondary value axis title.

Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.



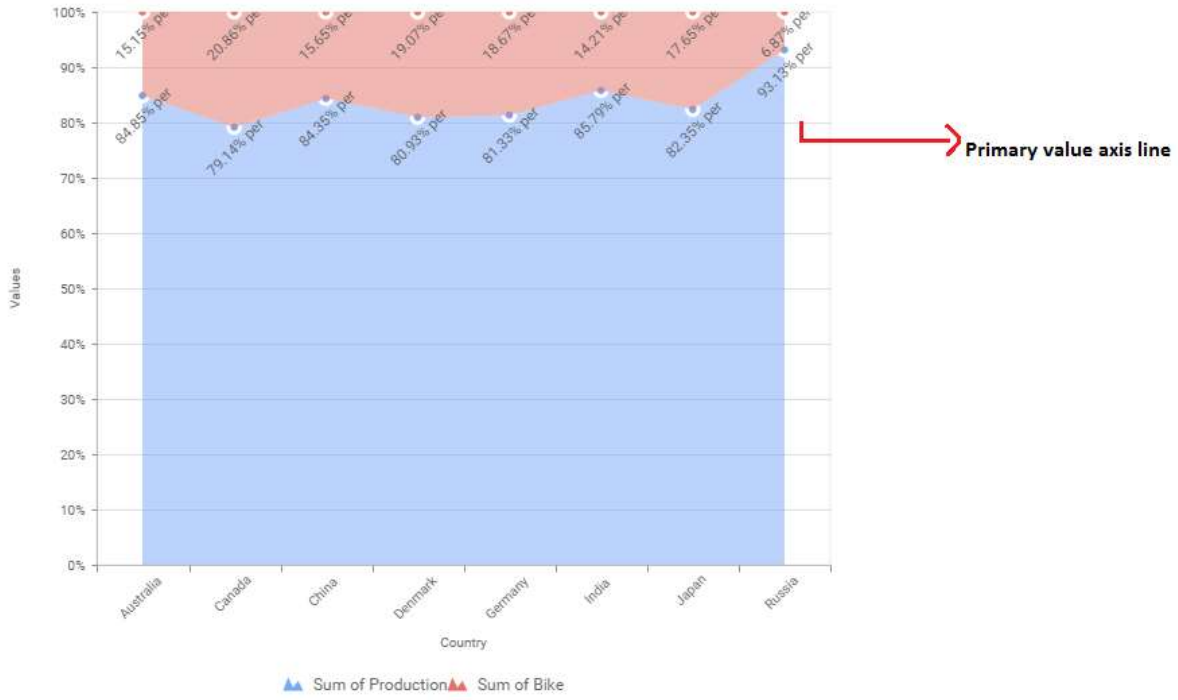
Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



Grid Line Settings

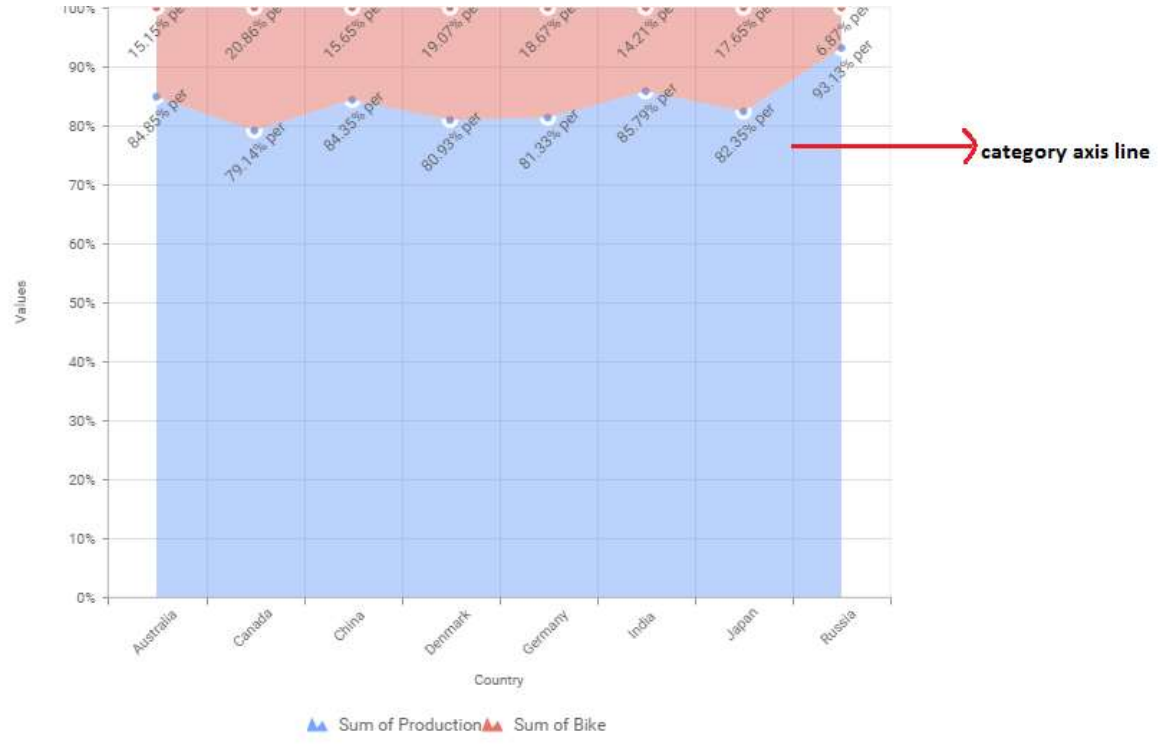
Primary value Axis

This allows you to enable the Primary Value Axis gridlines for the 100% stacked area chart.



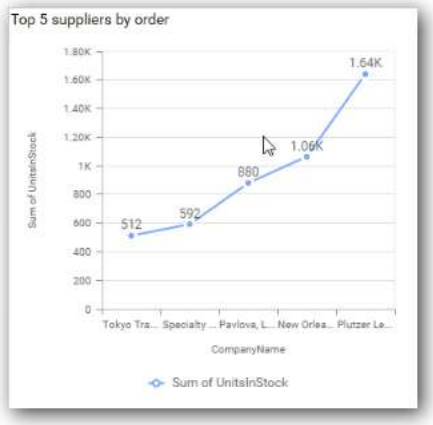
Category Axis

This allows you to define the visibility of Category axis gridlines.



Line Chart

Line Chart allows you to showcase trends for analysis over a time period with data points connecting using straight lines.

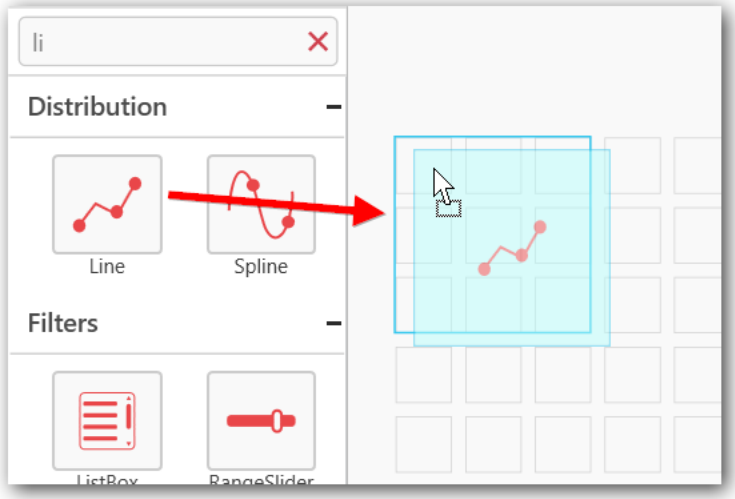


How to configure flat table data to Line Chart?

Line Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

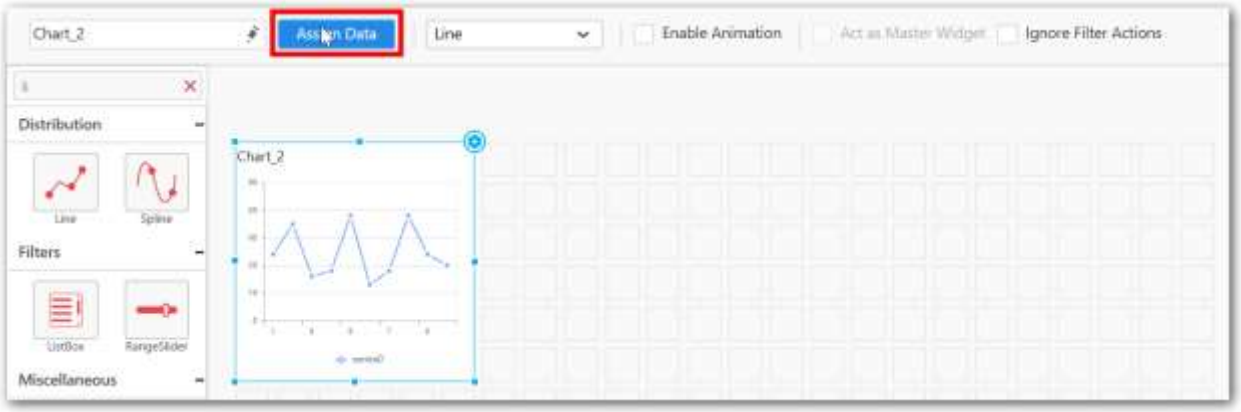
Follow the steps to configure data to Line chart

Drag and drop the control to canvas and resize it to your required size.

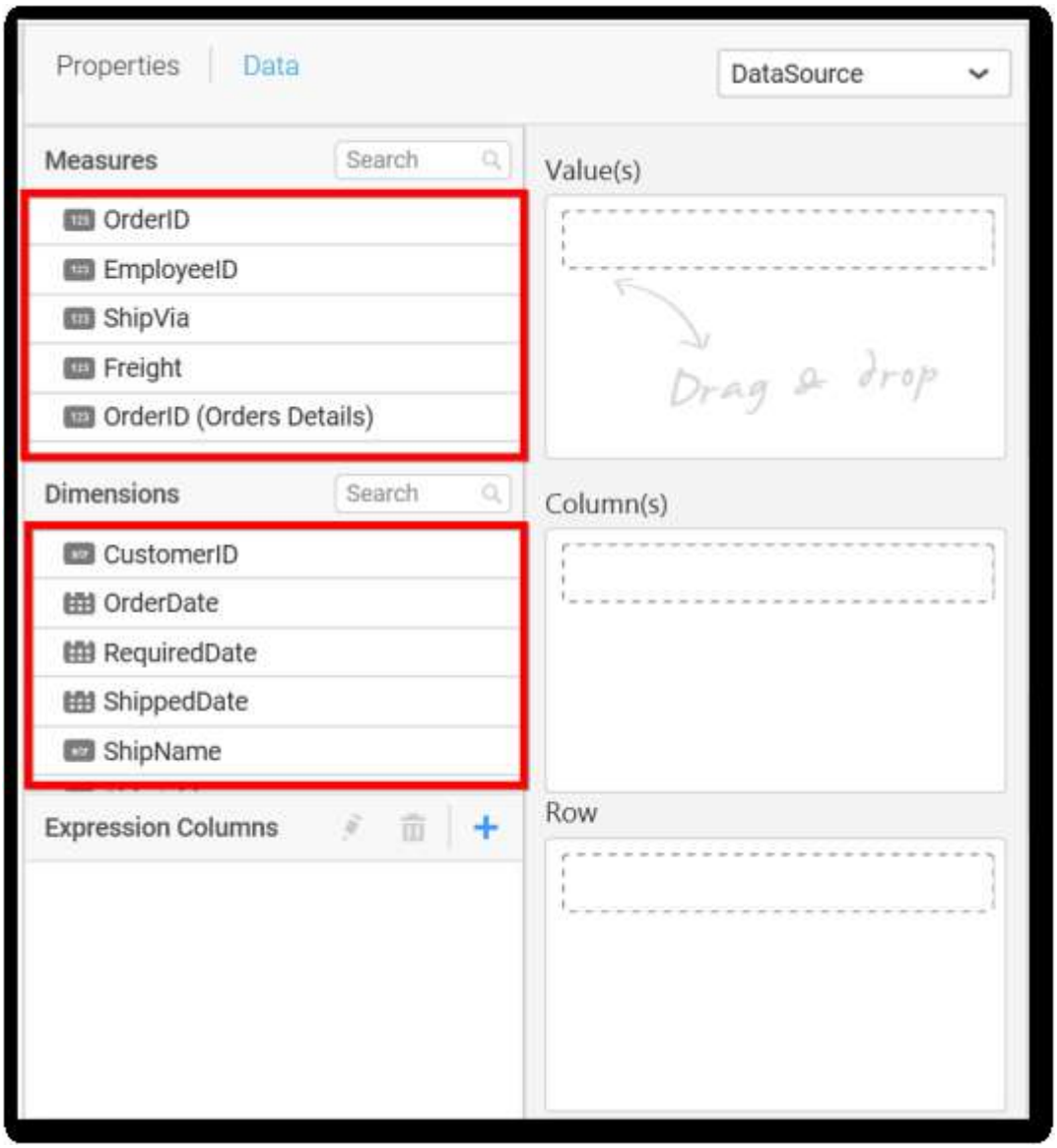


Connect to the data source.

Focus on the line widget and click on Assign Data.

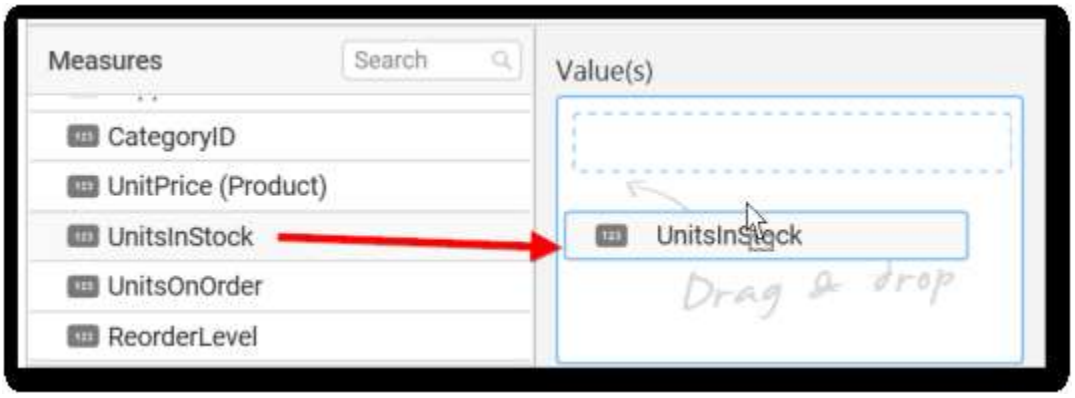


The data pane will be opened with available Measures and Dimensions from the connected data source.

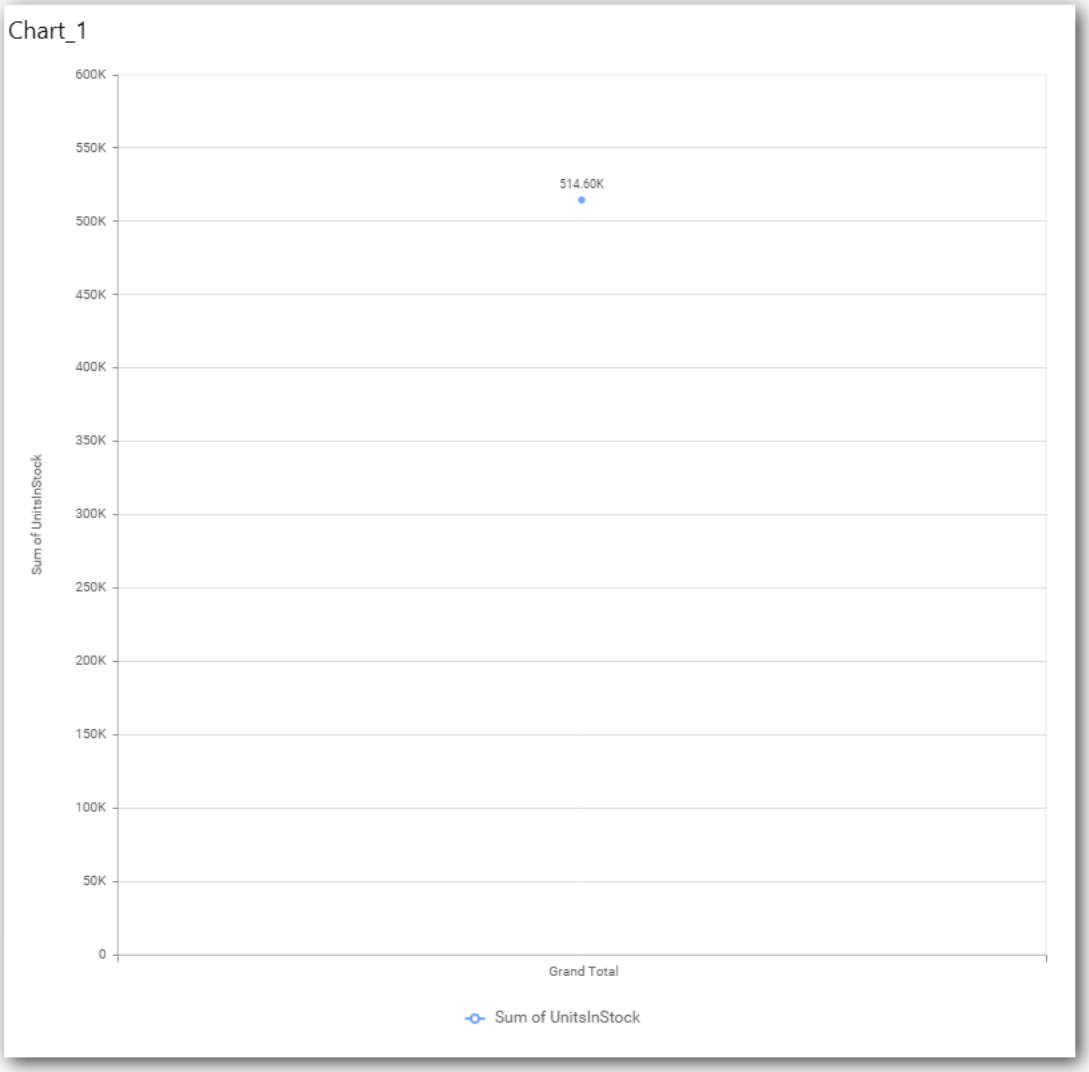


Assigning Value(s)

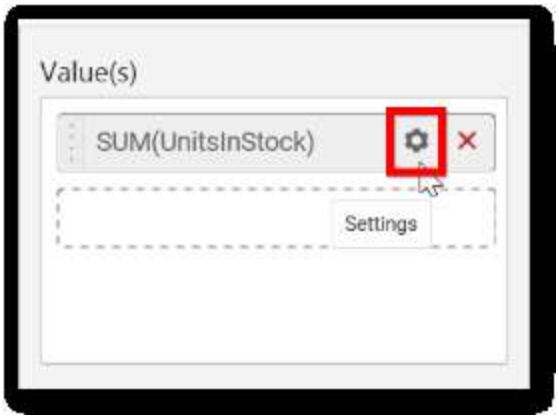
Drag and drop the Measure into Value.



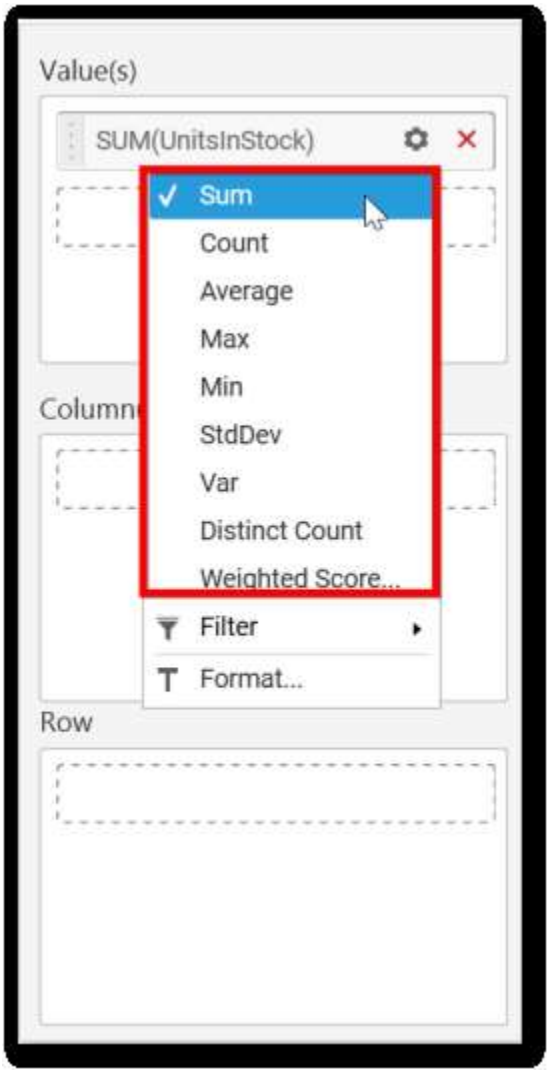
Now the chart will be rendered like this.



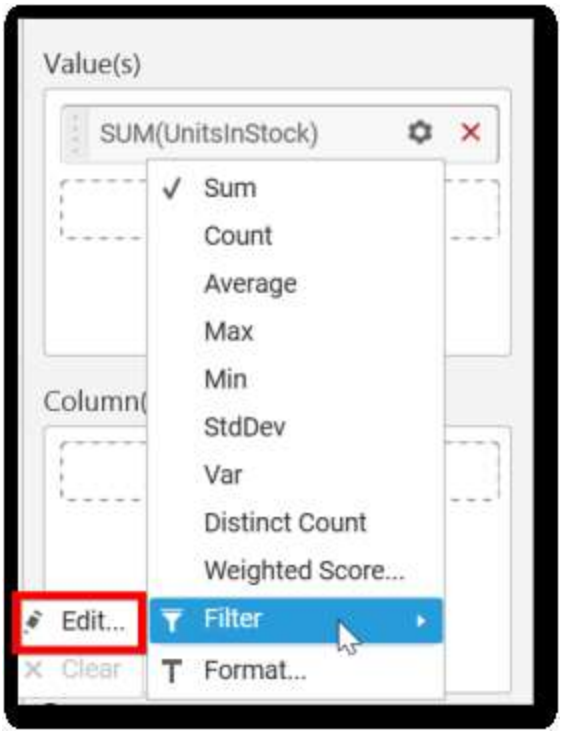
You can change the summary type of the value by clicking on Settings option.



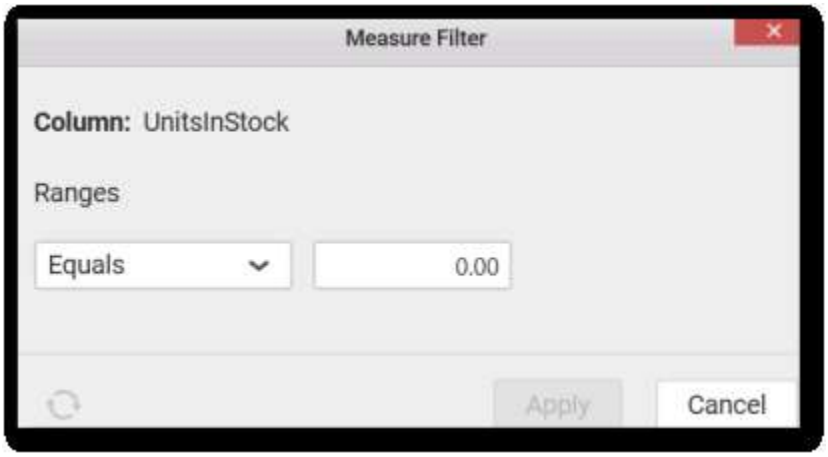
Select the required summary type from list.

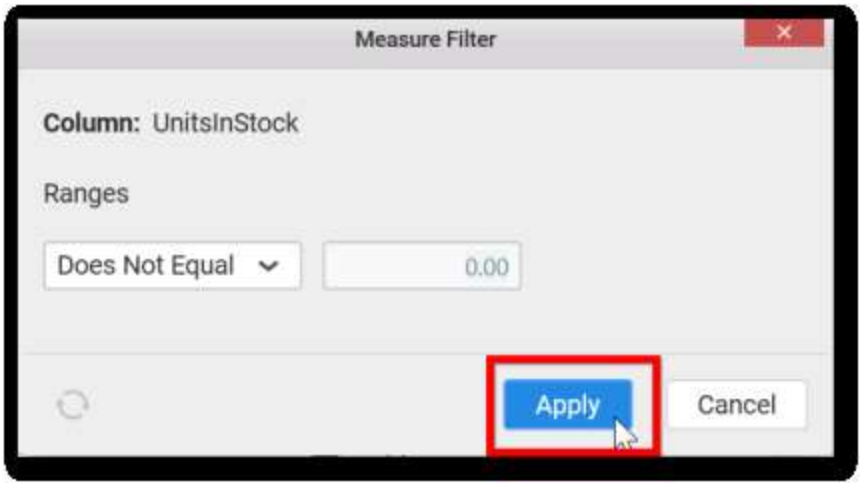


You can select what data to be displayed by choosing filter option.

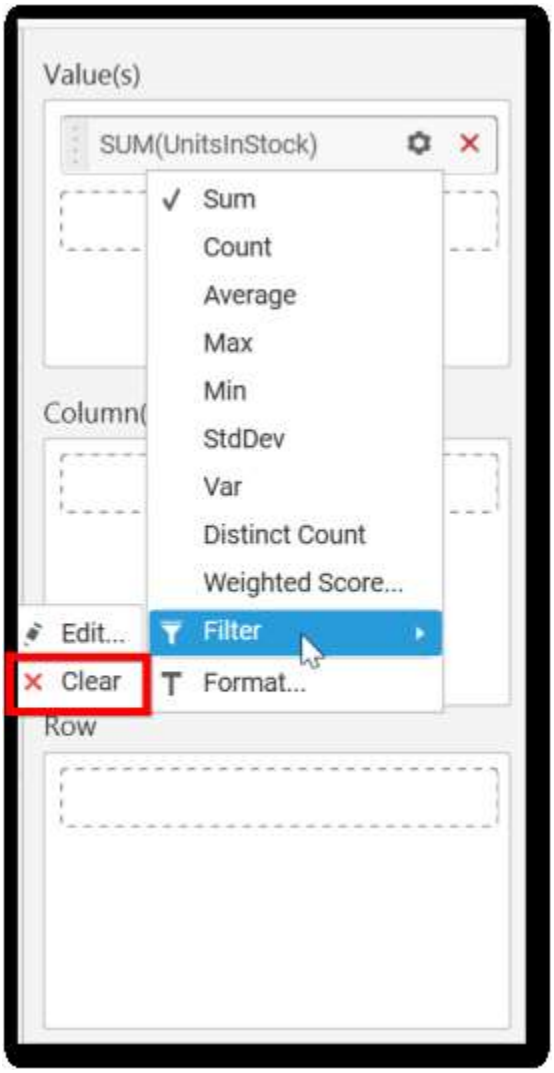


The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.



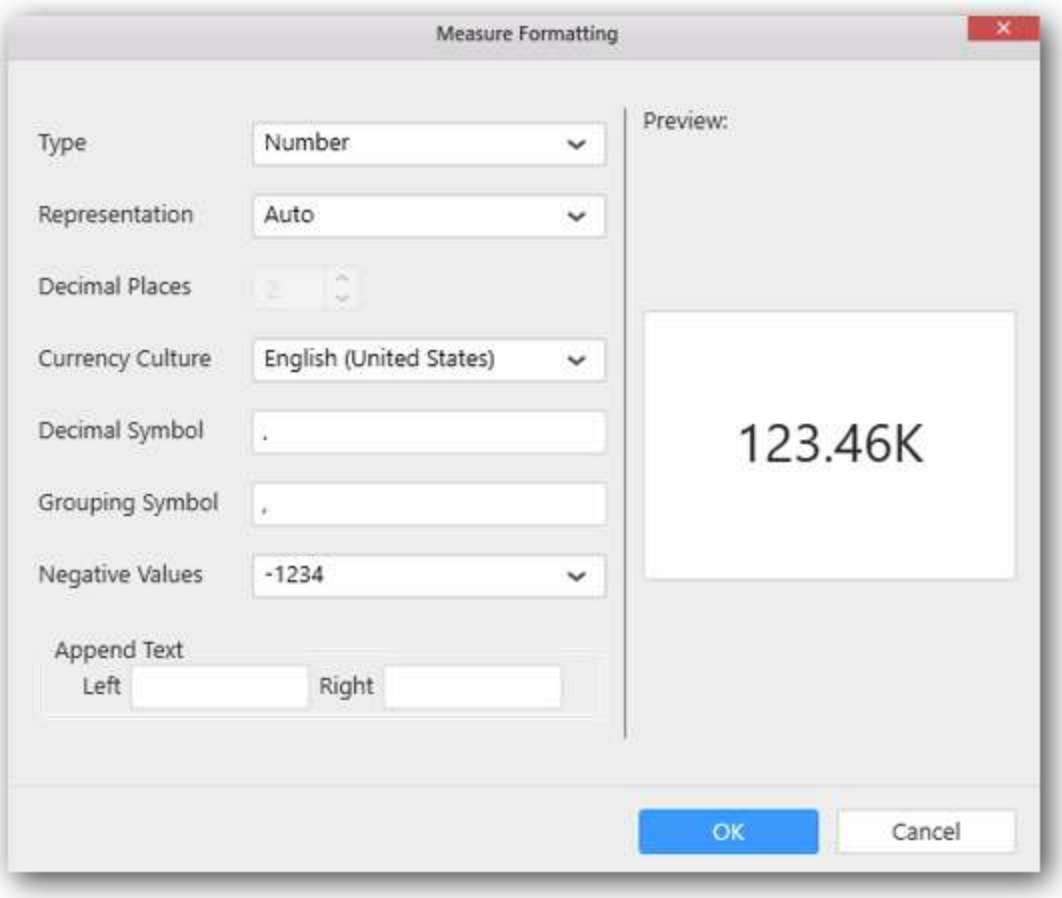


You can Clear the filter.

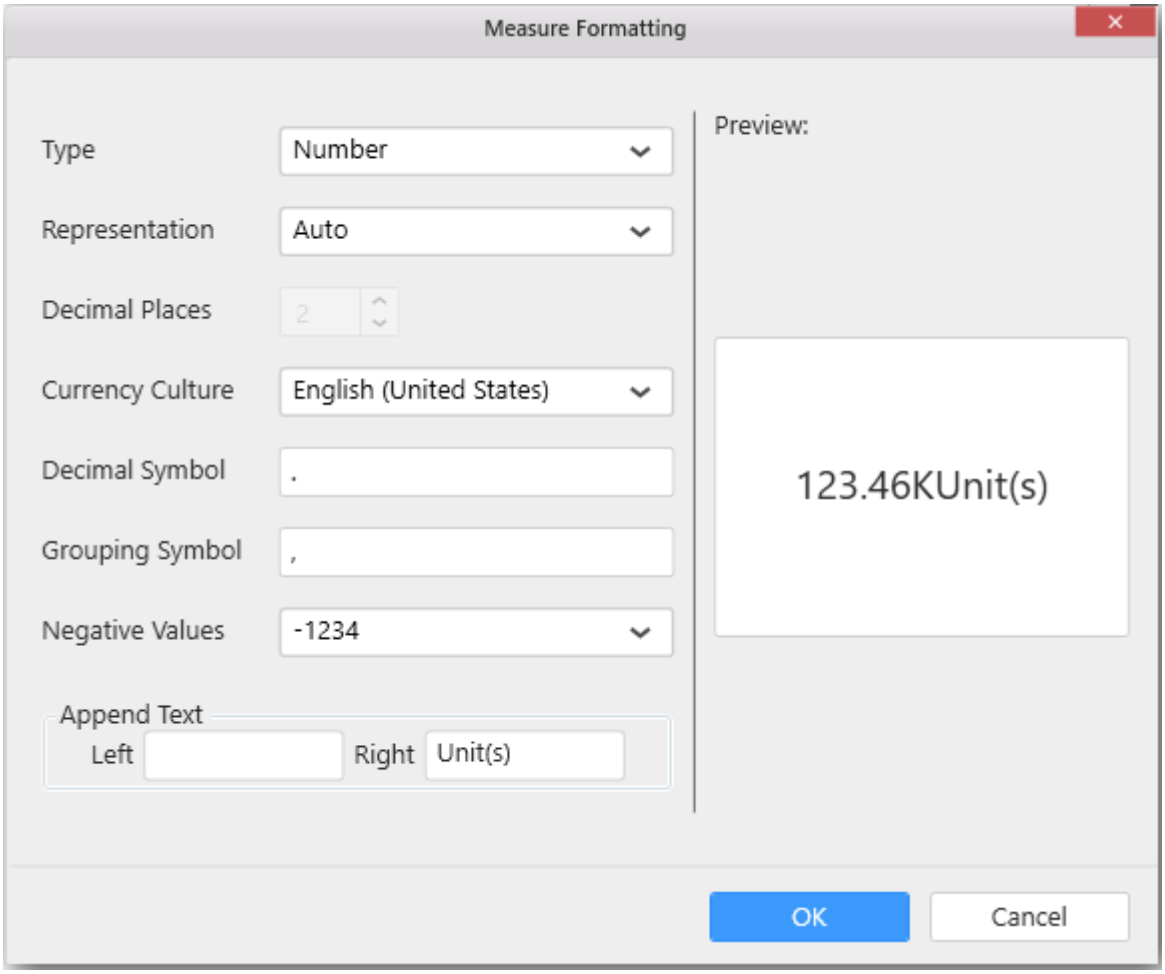


You can Format the value.

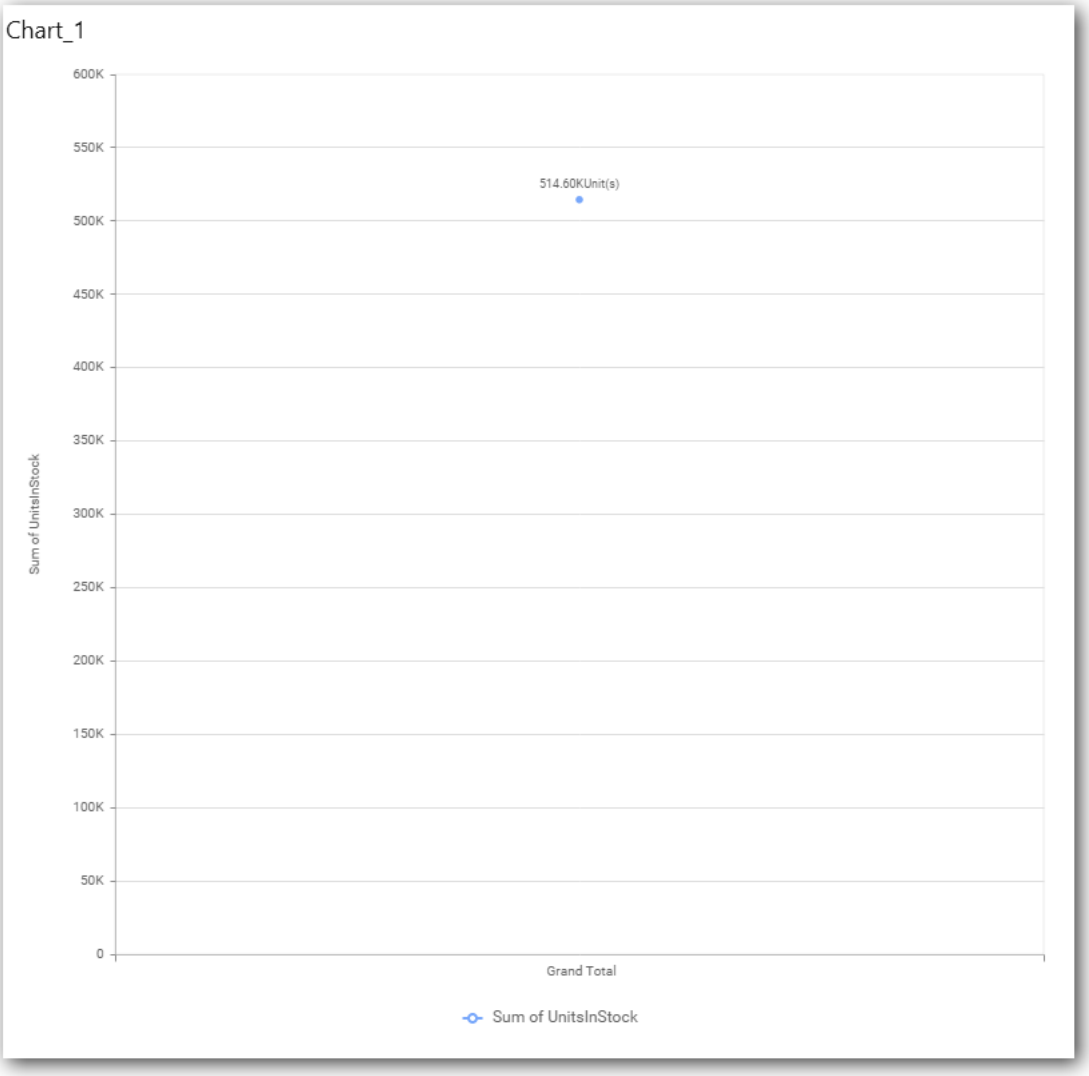
The format options will be shown.



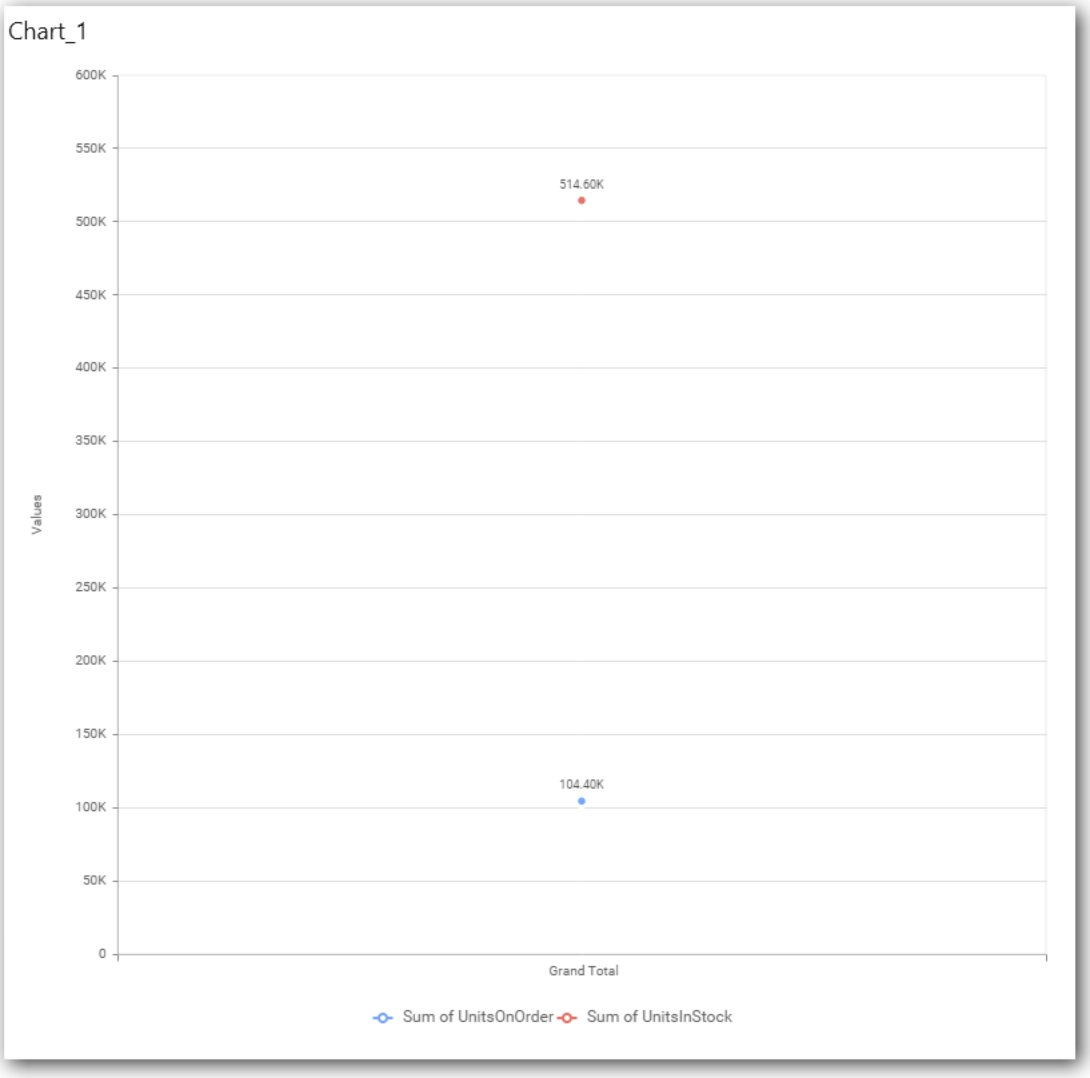
Choose the options you need and click OK.



Now the Chart will be rendered like this.



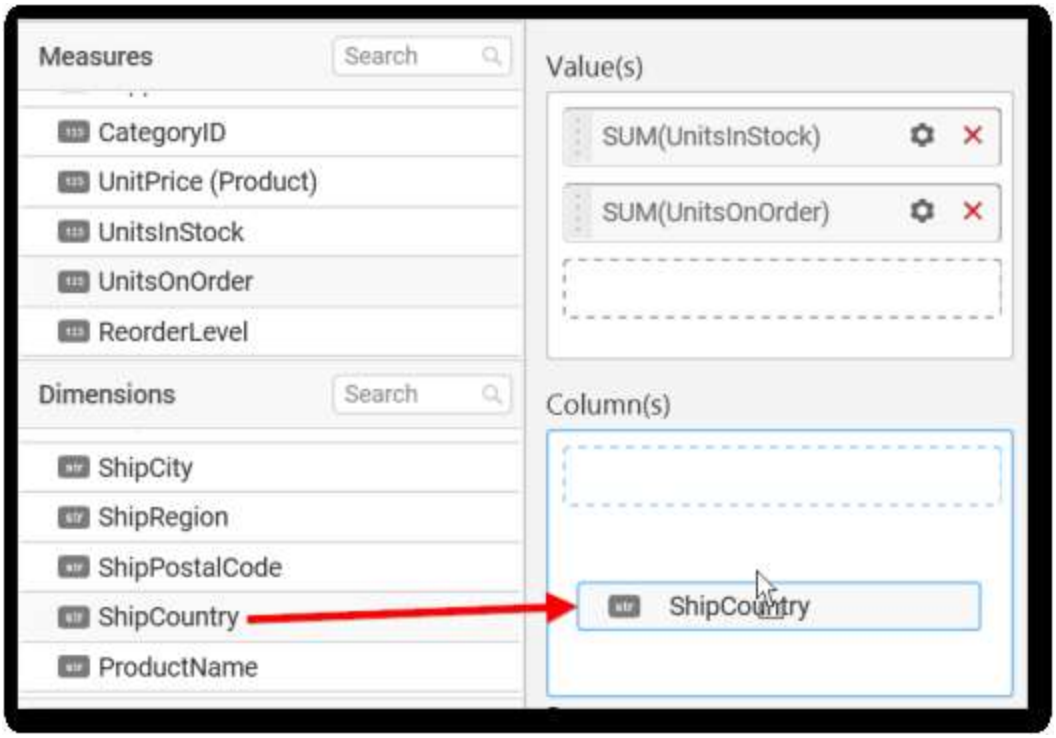
You can add more number of values by drag and drop the Measures into Value field.

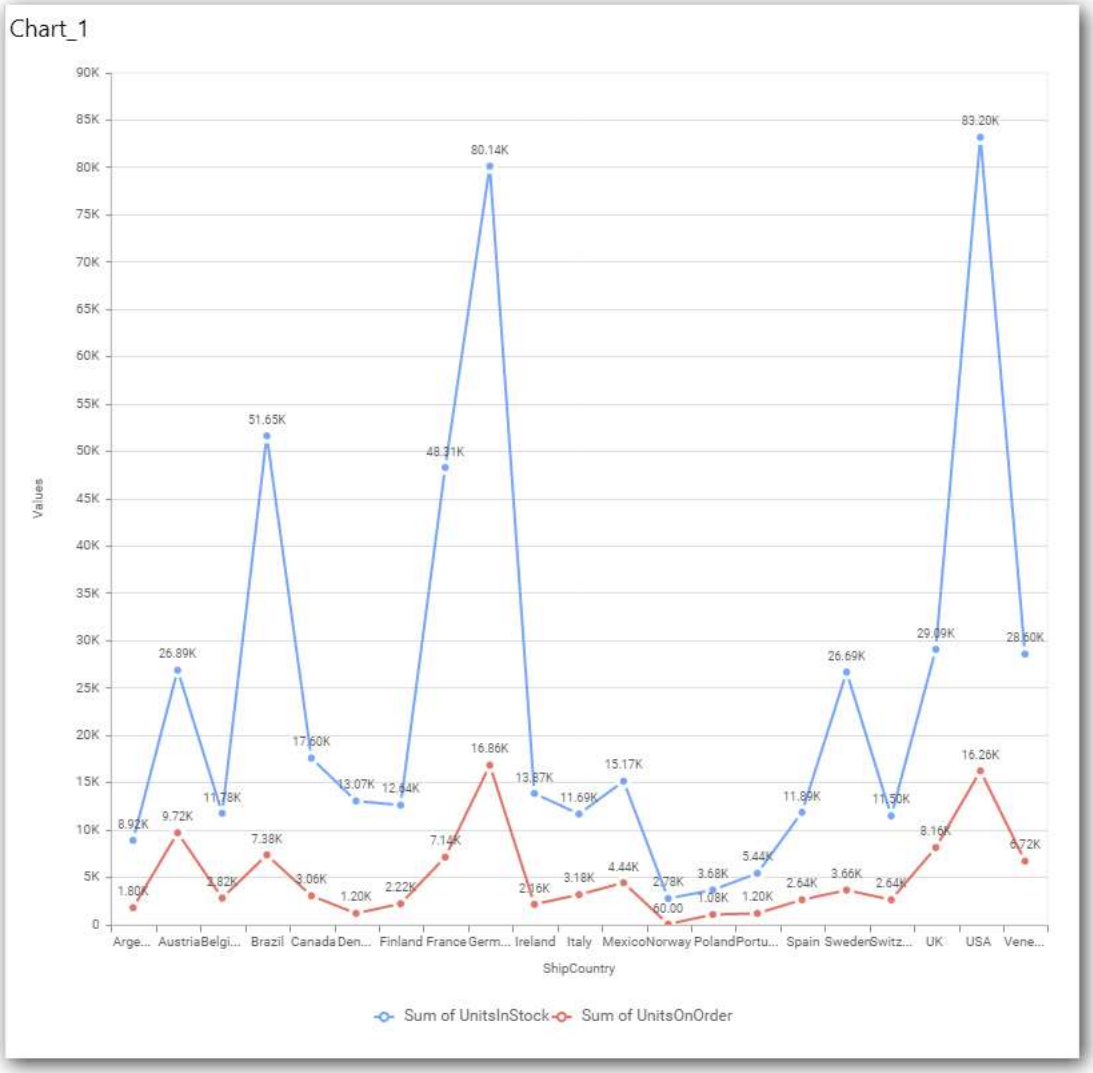


You can also add Dimensions and Columns to Value(s).

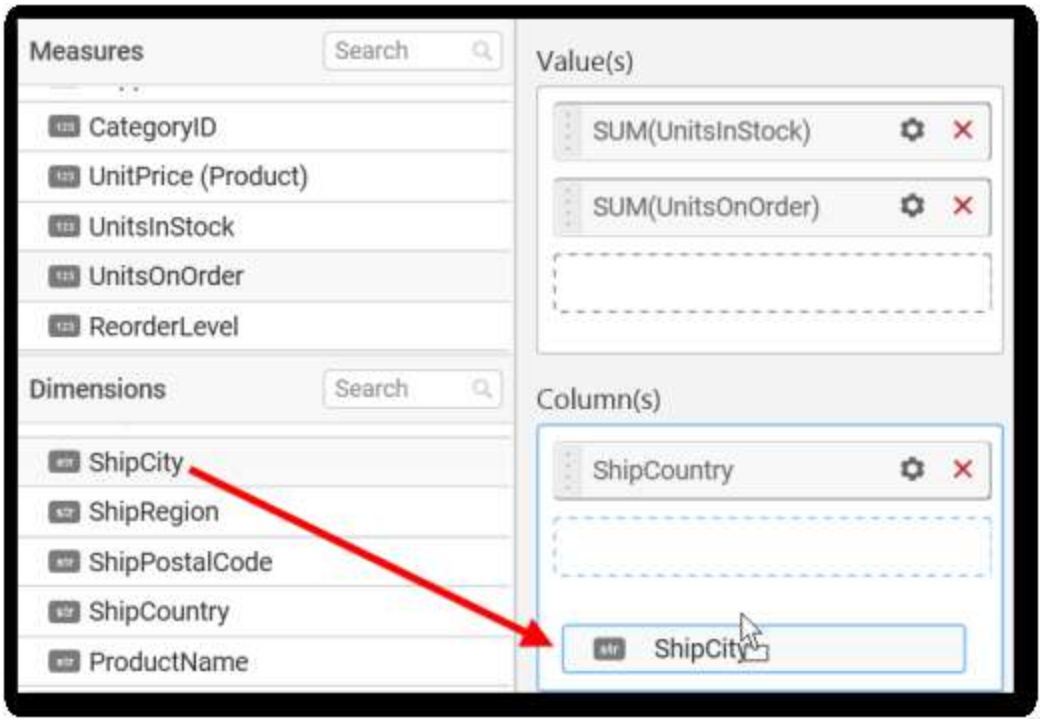
Assigning Column(s)

You can add the Dimension into Column(s) field by drag and drop.

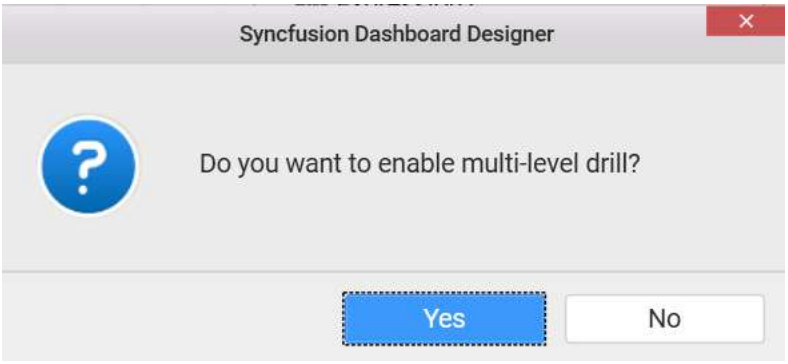




You have option to add more than one Column Value

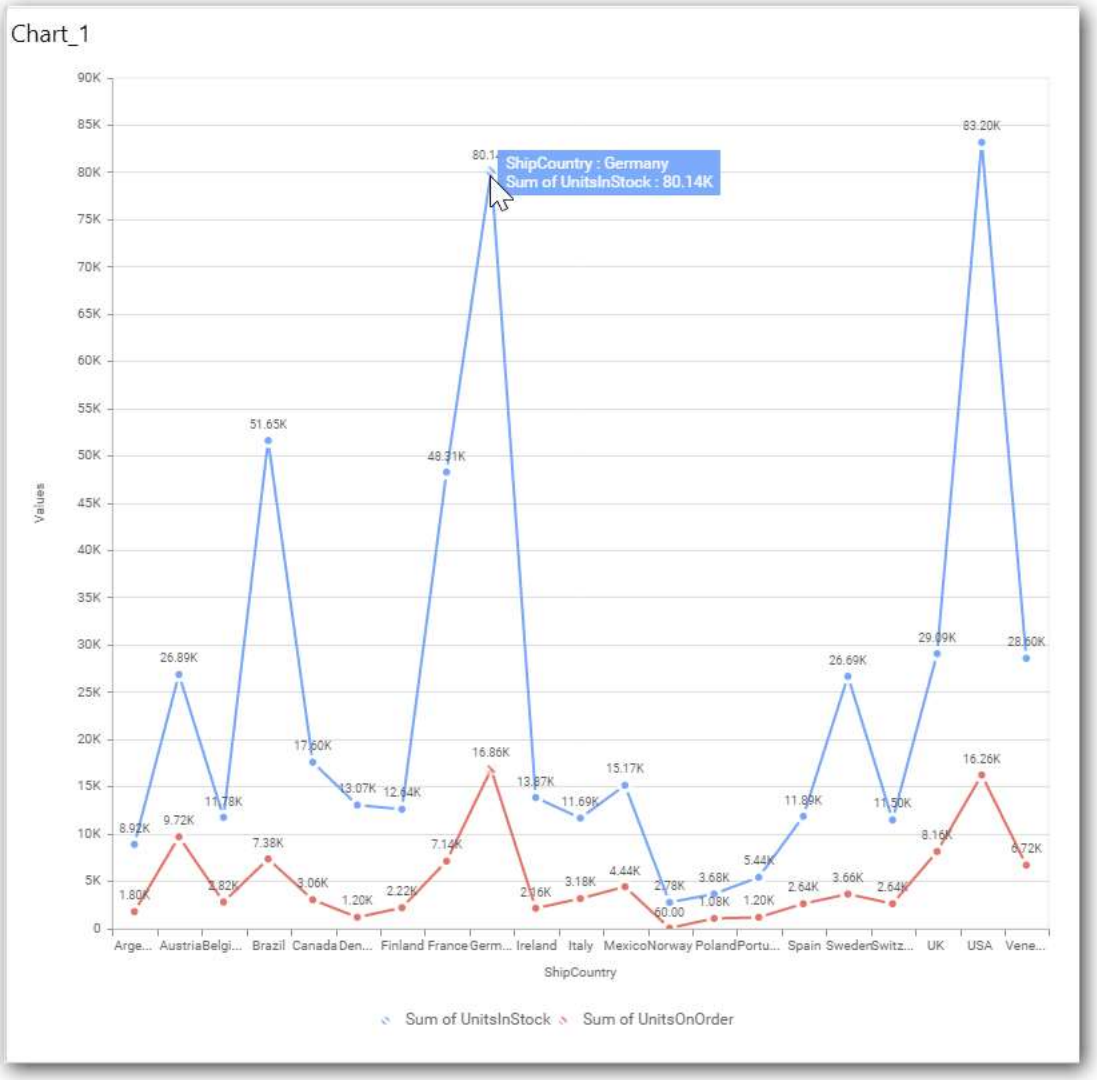


The following alert message will be shown.

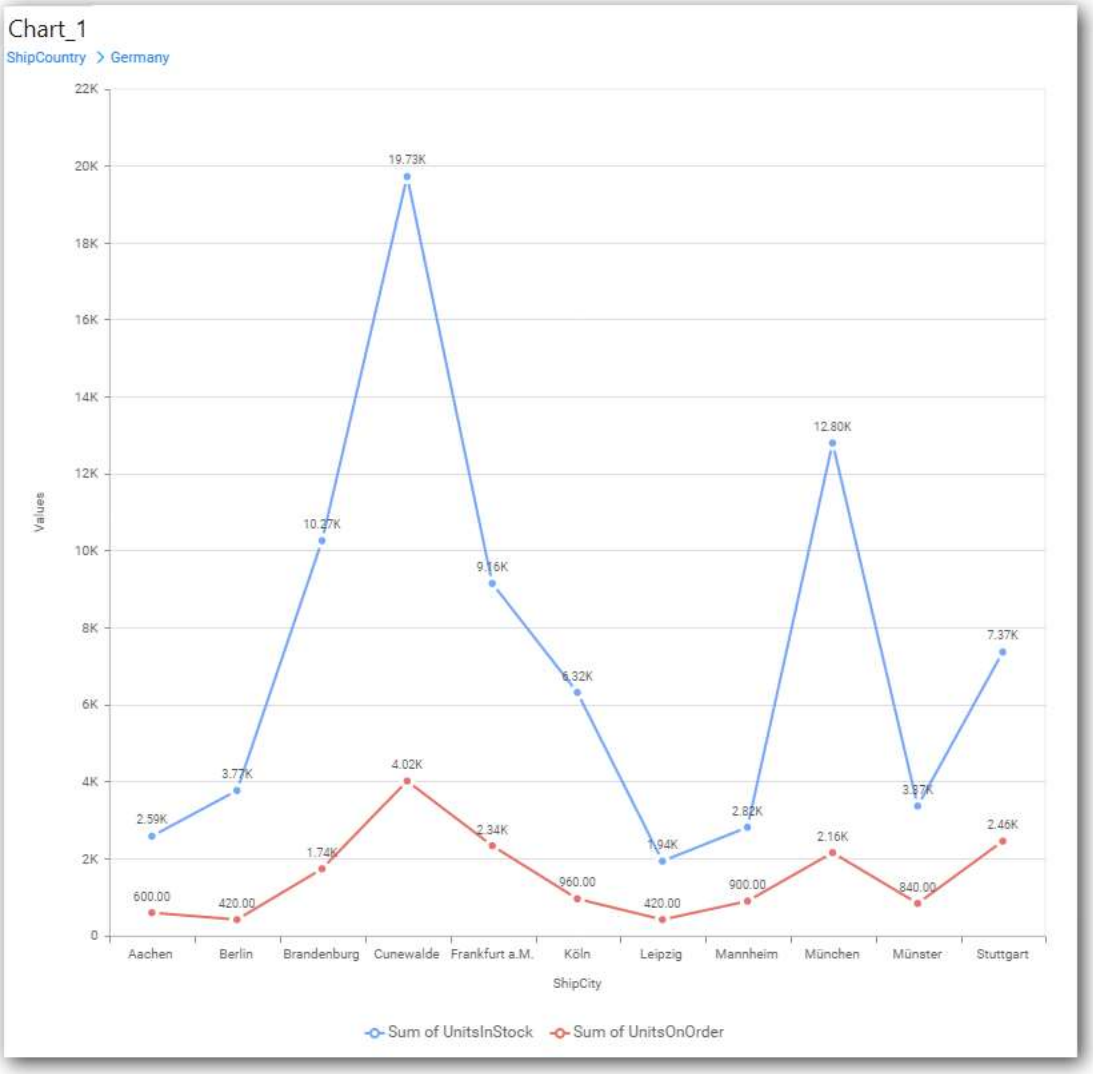


- If you choose Yes Drill down option will be enabled.

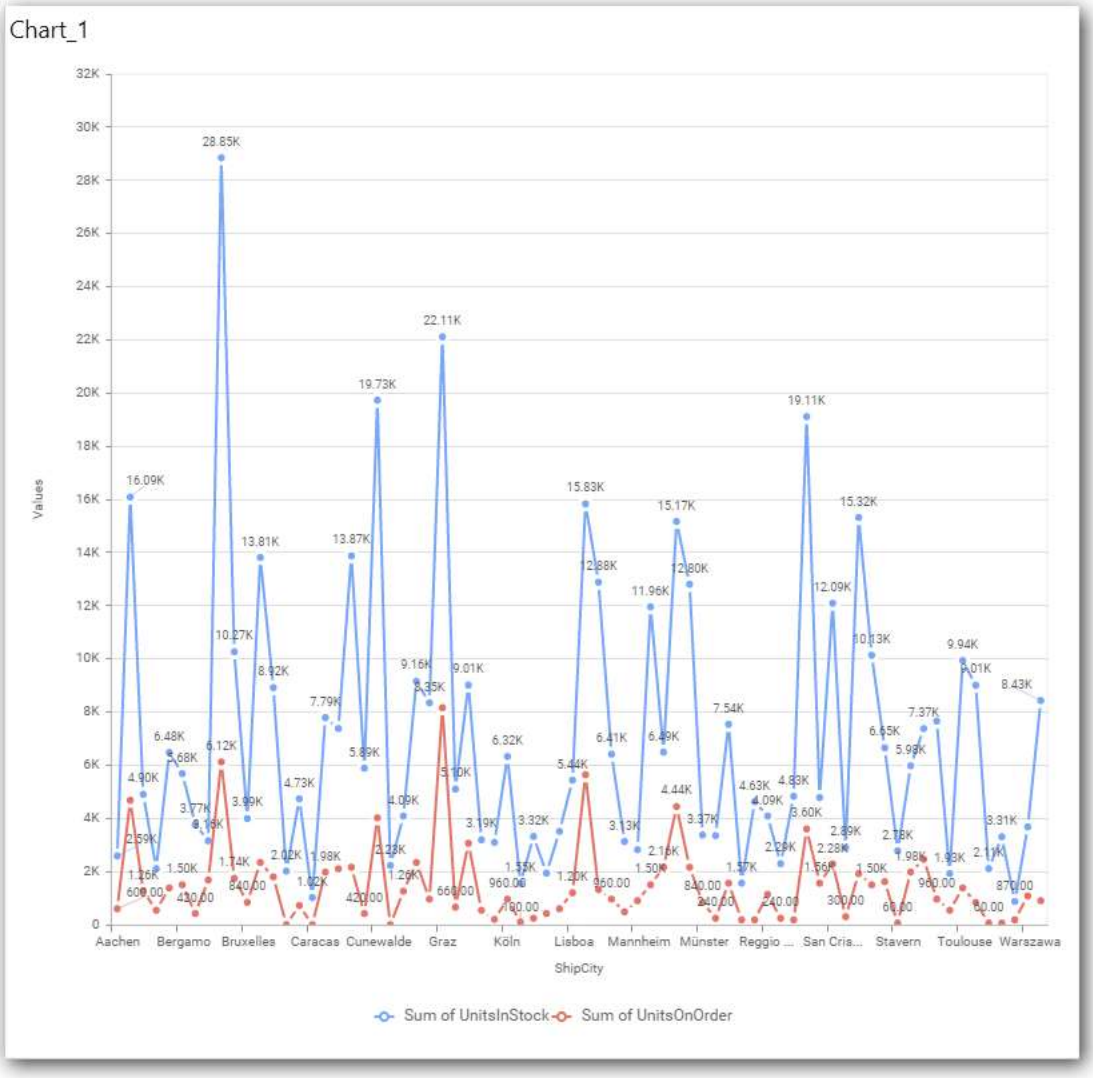
You can drill down the chart by clicking on the chart.



The drilled view of the chart is follows.

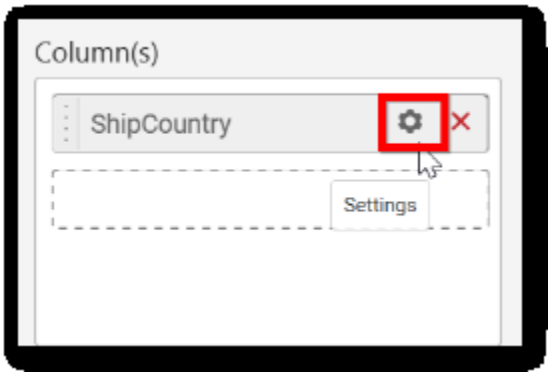


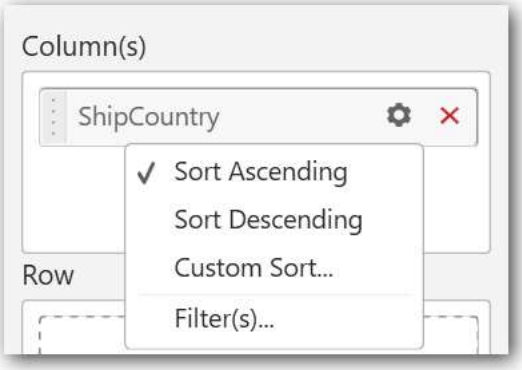
- If you click No the new Dimension value will replace old value.



You can also add Measures and Expression columns into Column(s) field.

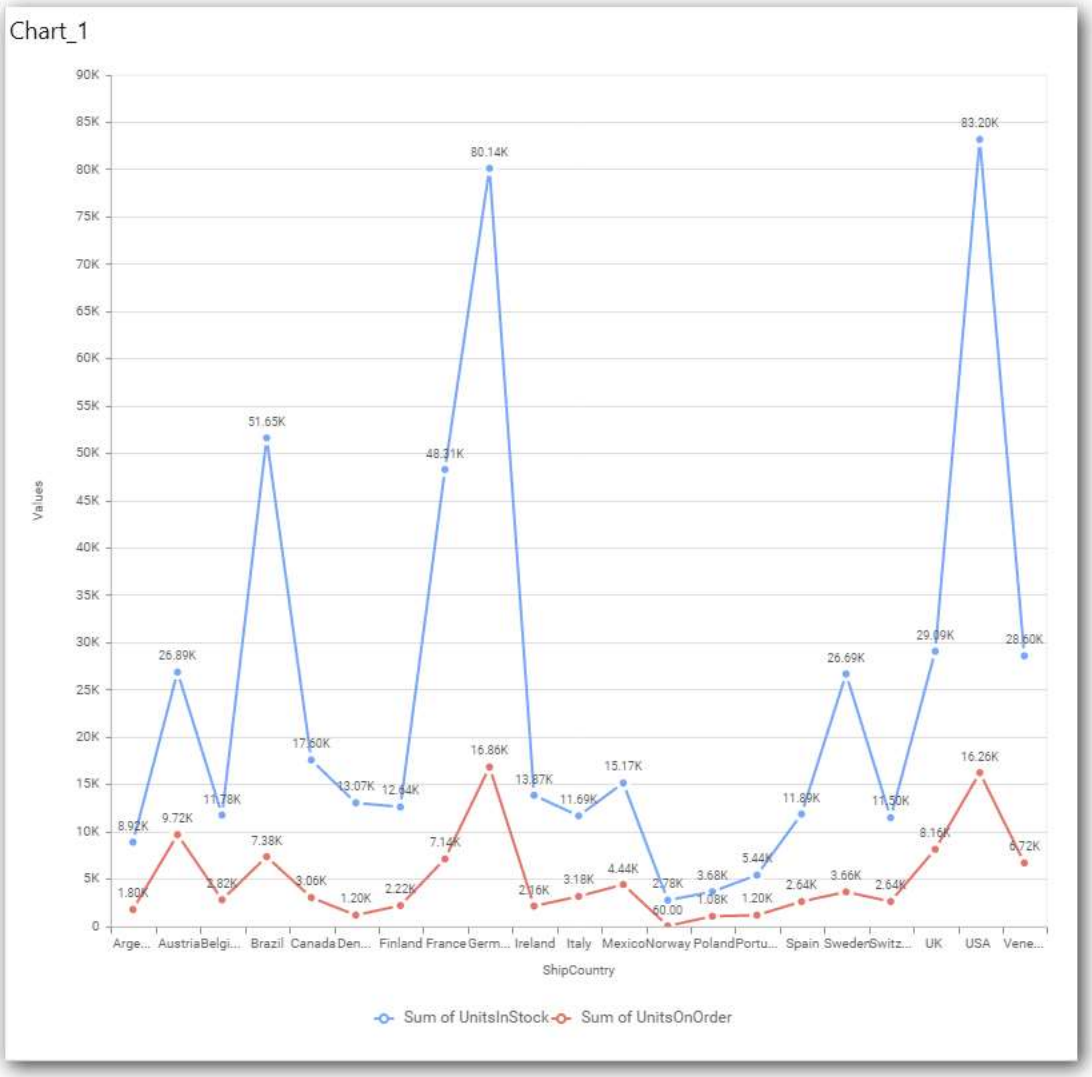
You have options to change the Settings.



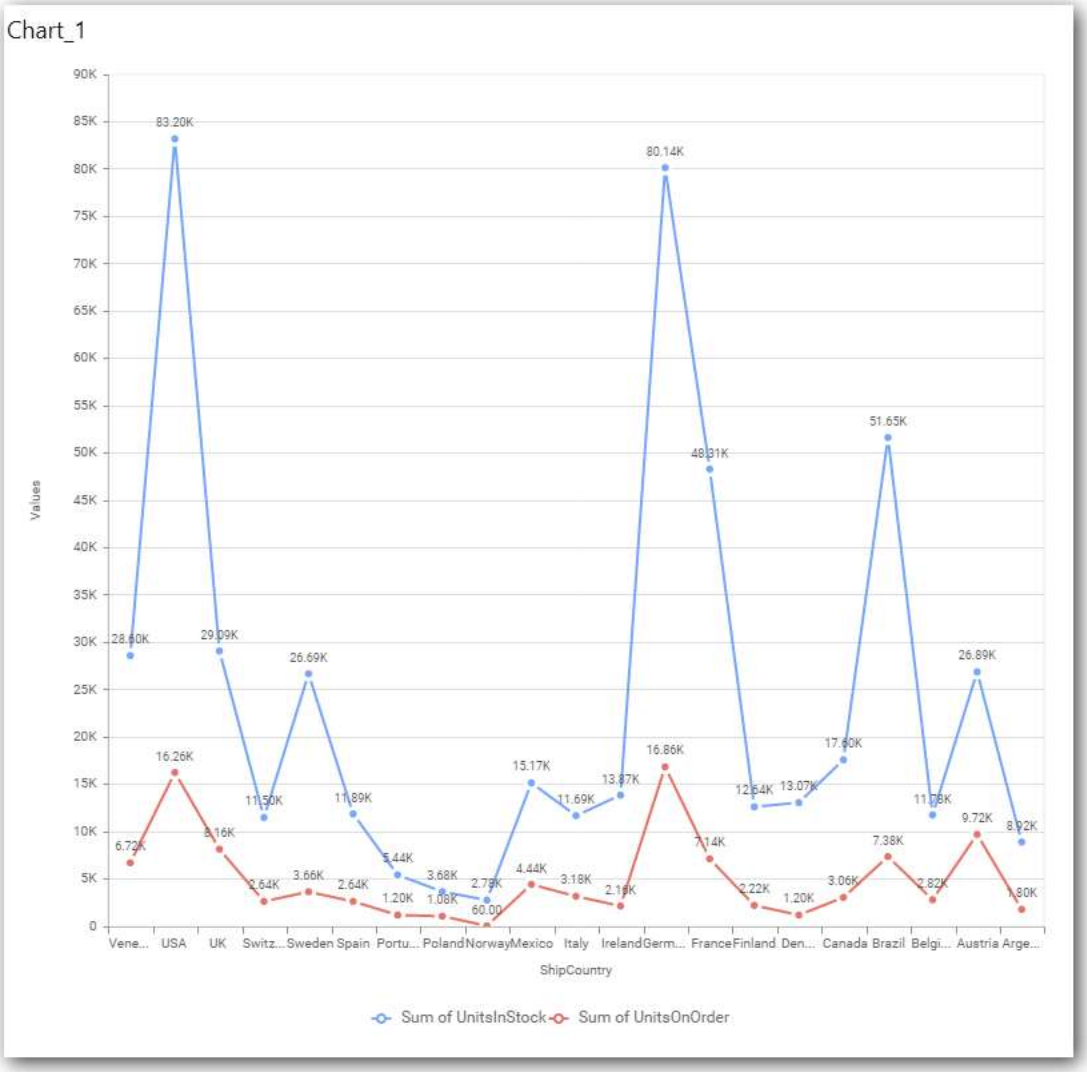


You can sort the chart either in Ascending or Descending series.

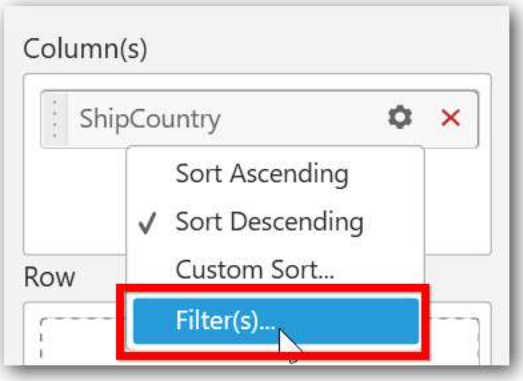
Ascending Order:

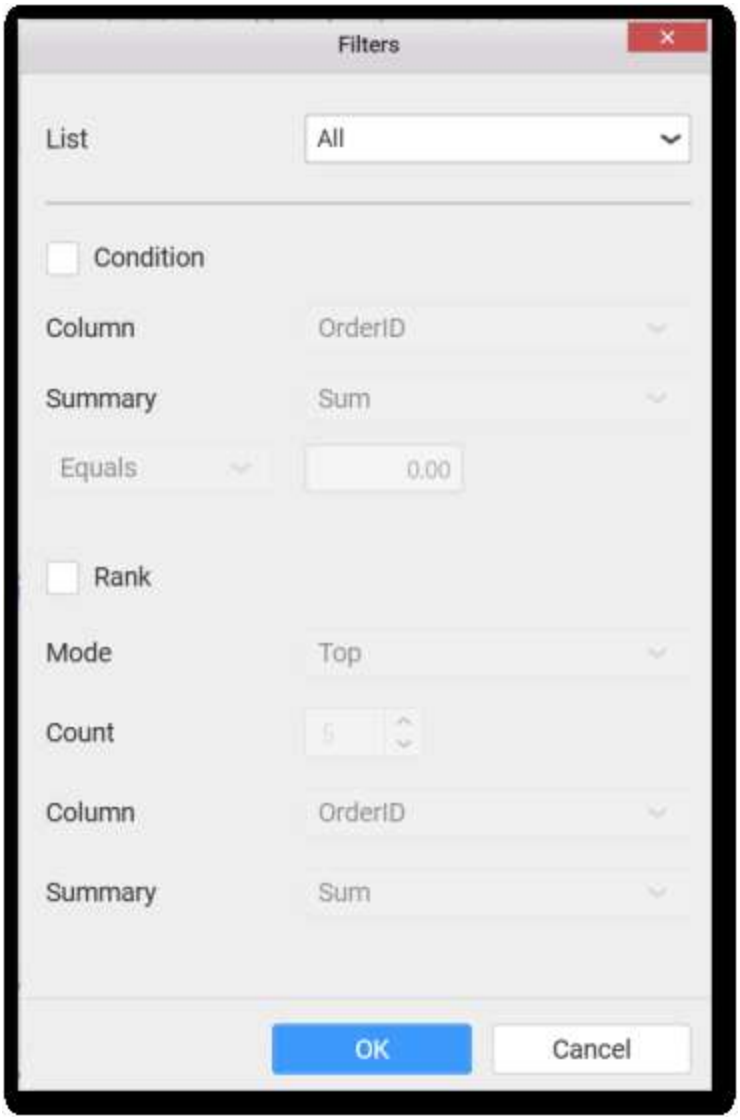


Descending order:

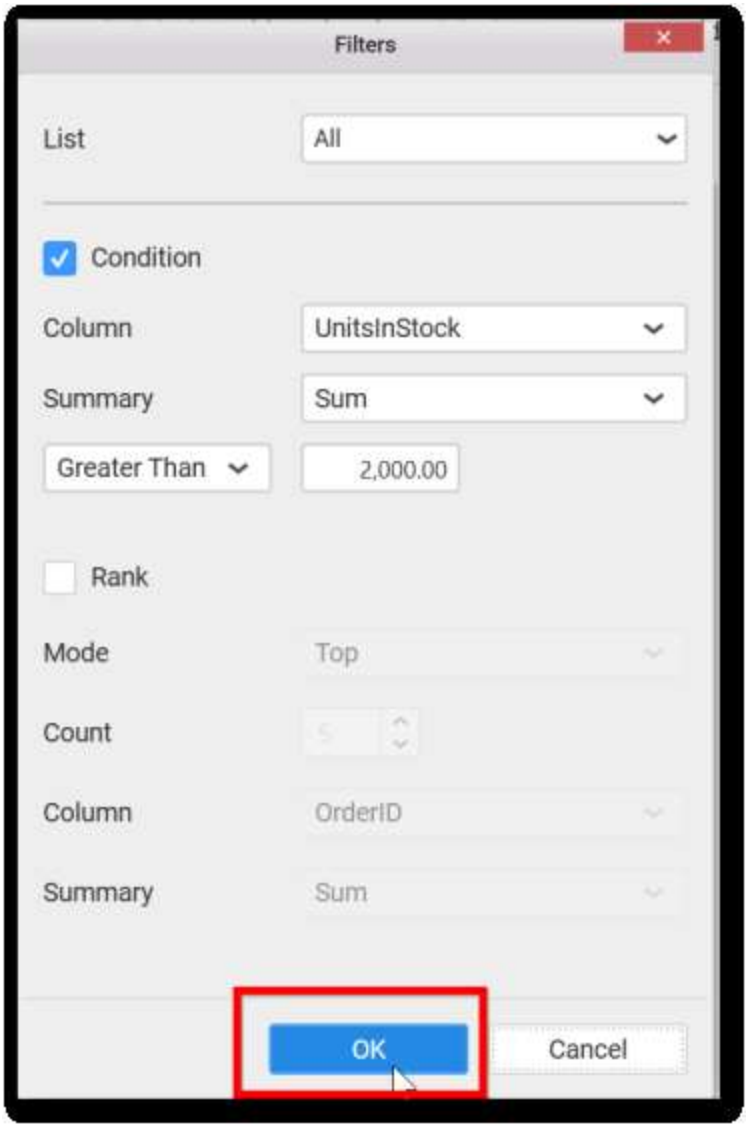


You can apply a Filter.

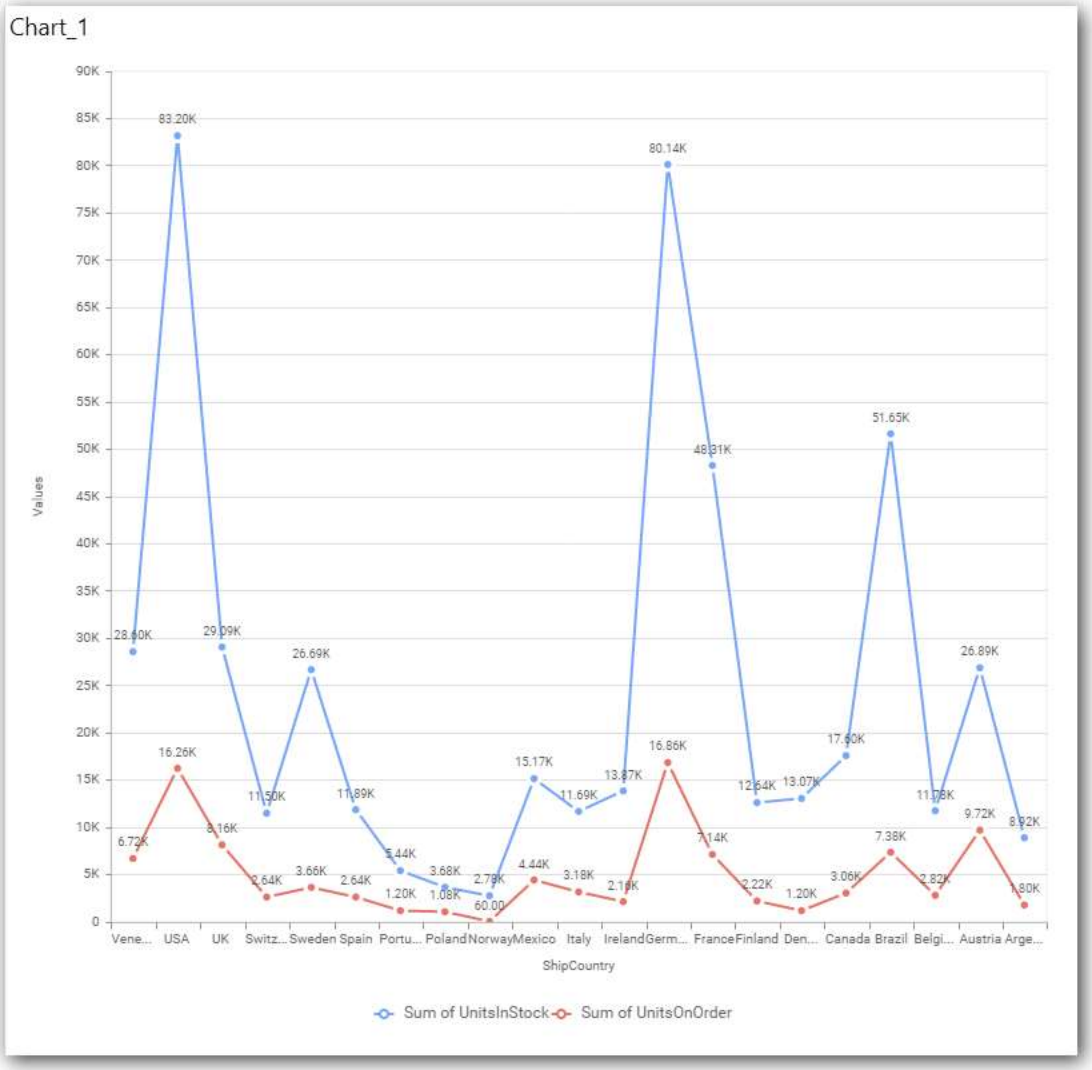




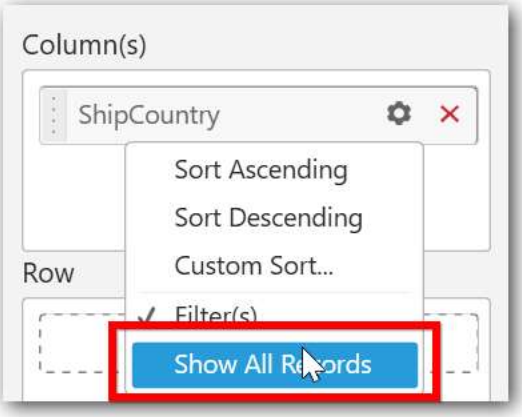
Select the Conditions and Rank you need.



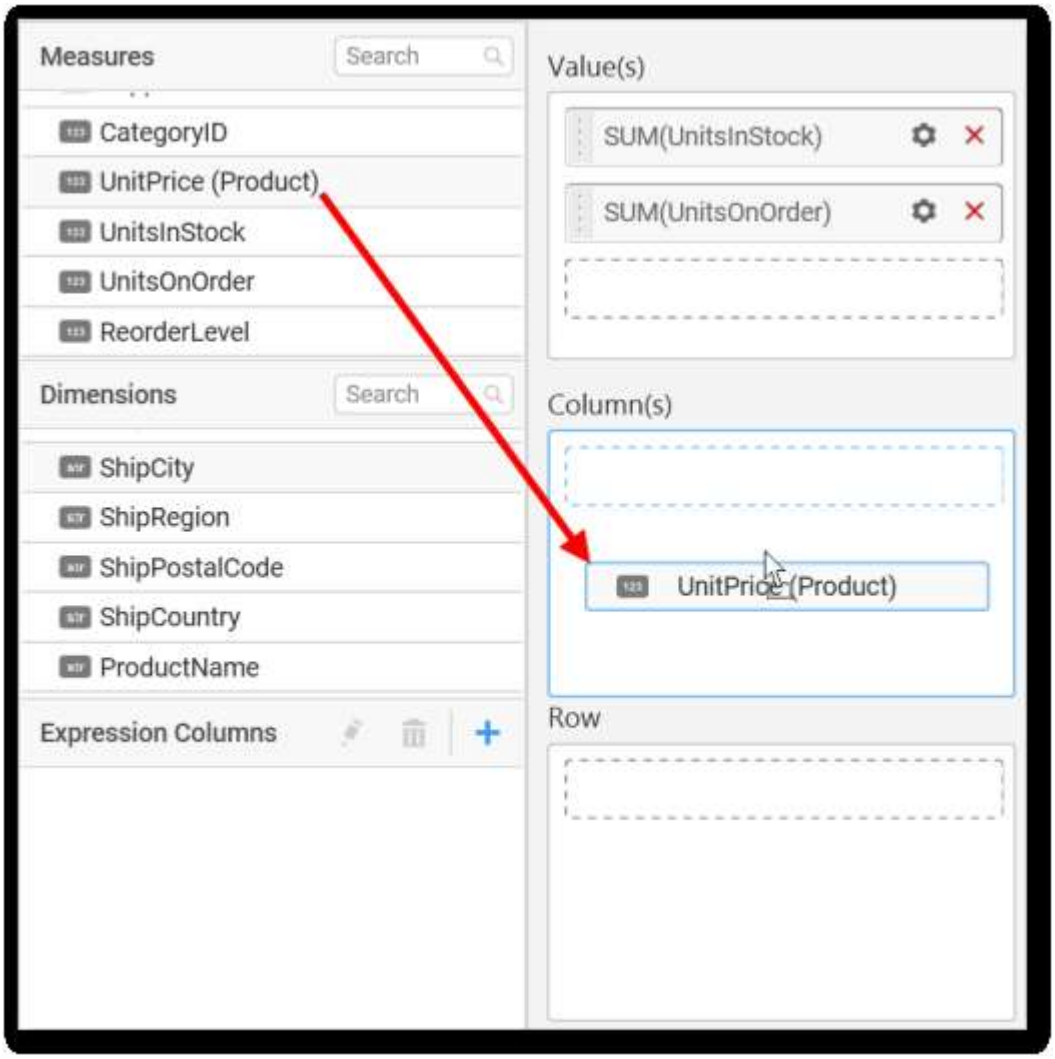
Now the chart will be rendered like this.



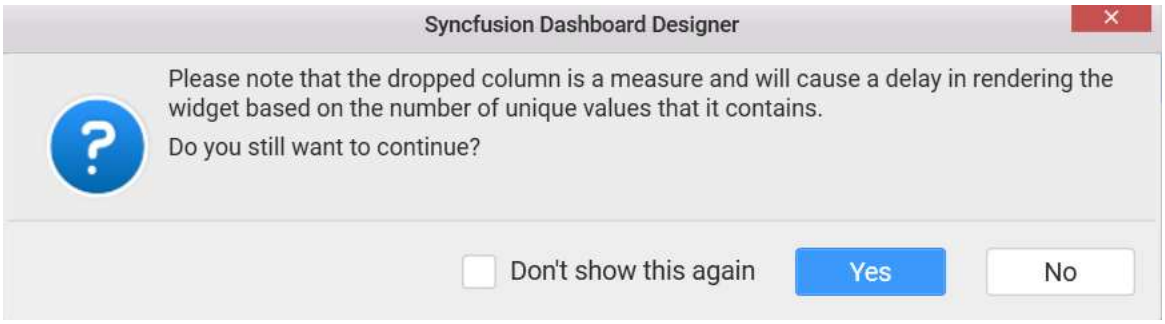
To show all records again click on Show All Records.



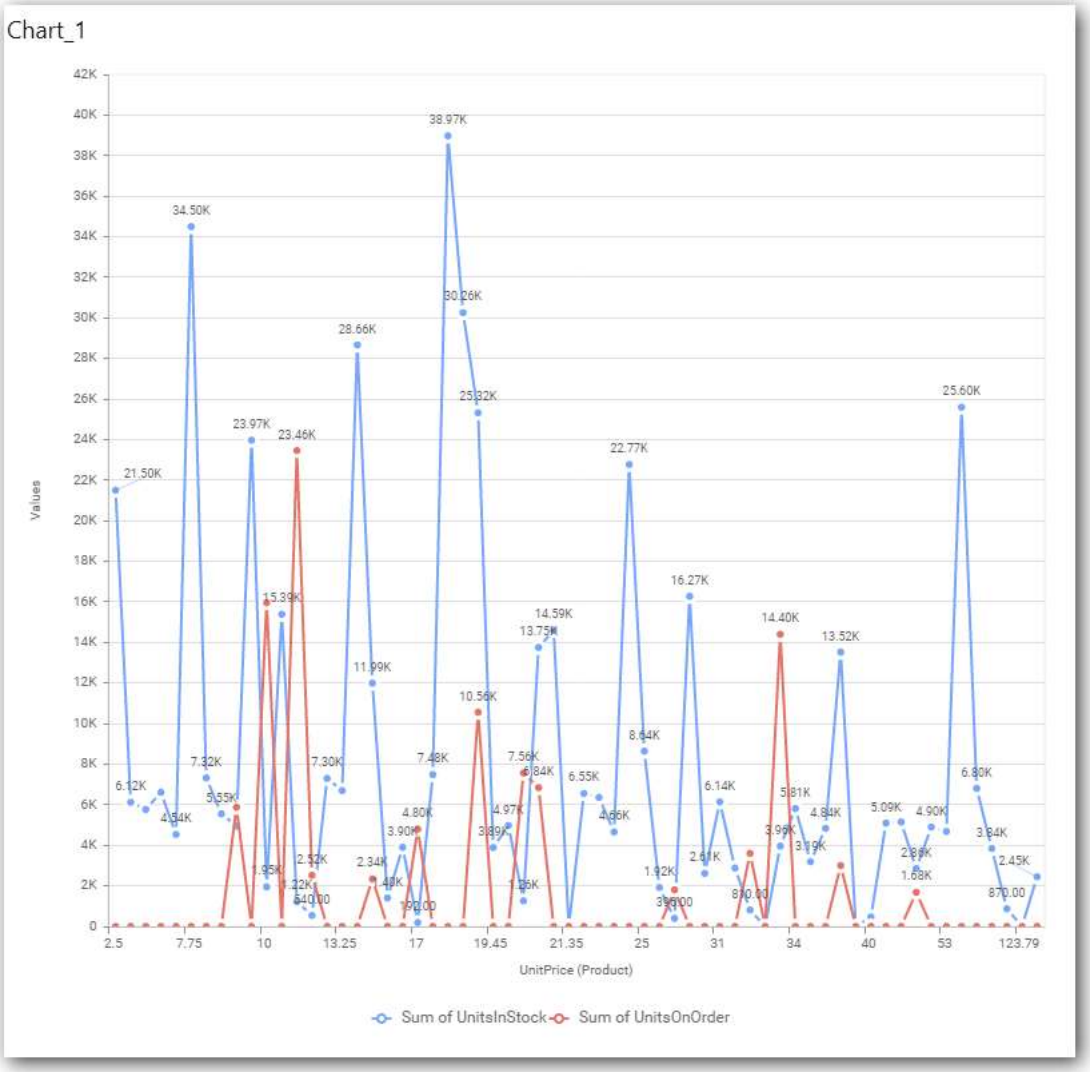
You can add the Measures into Column(s).



The following alert message will shown.

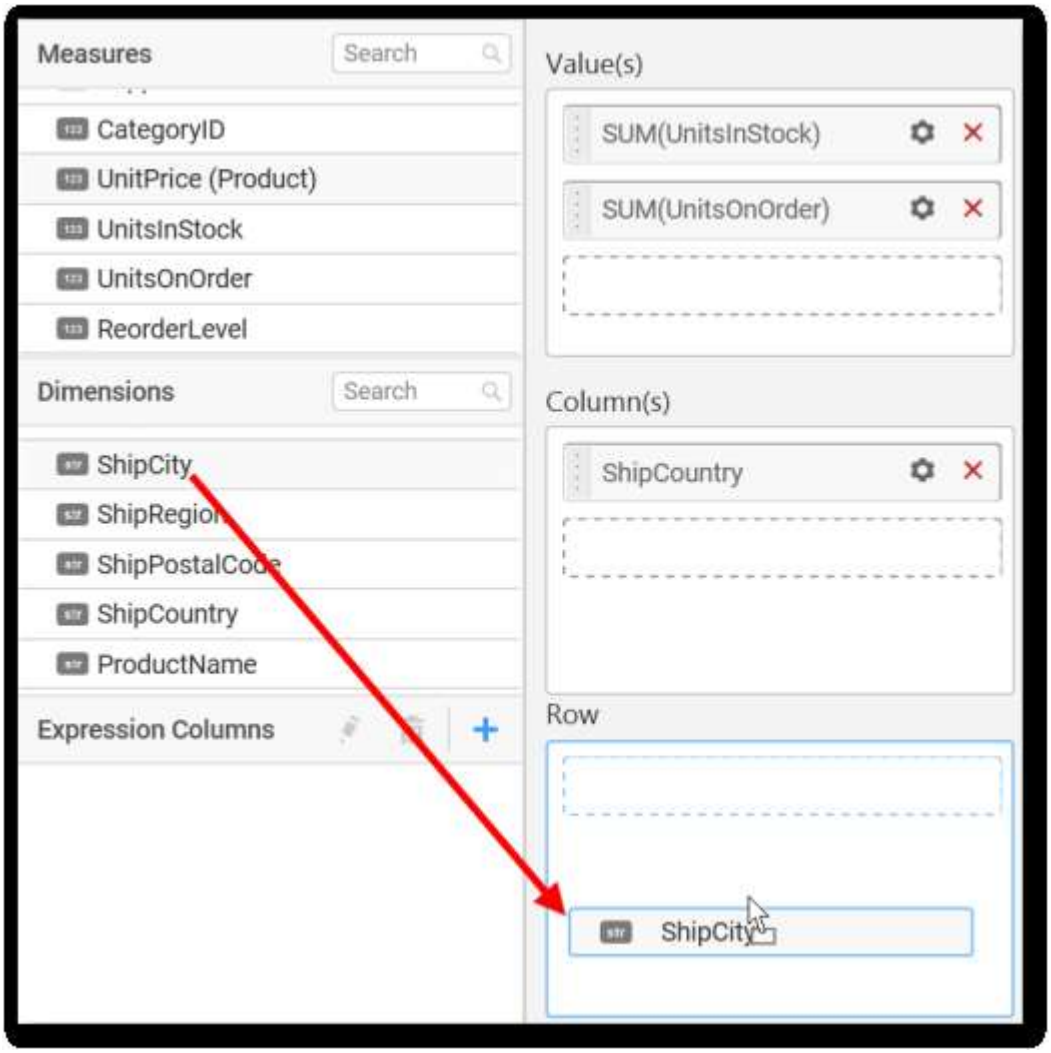


To continue click Yes, otherwise click No.

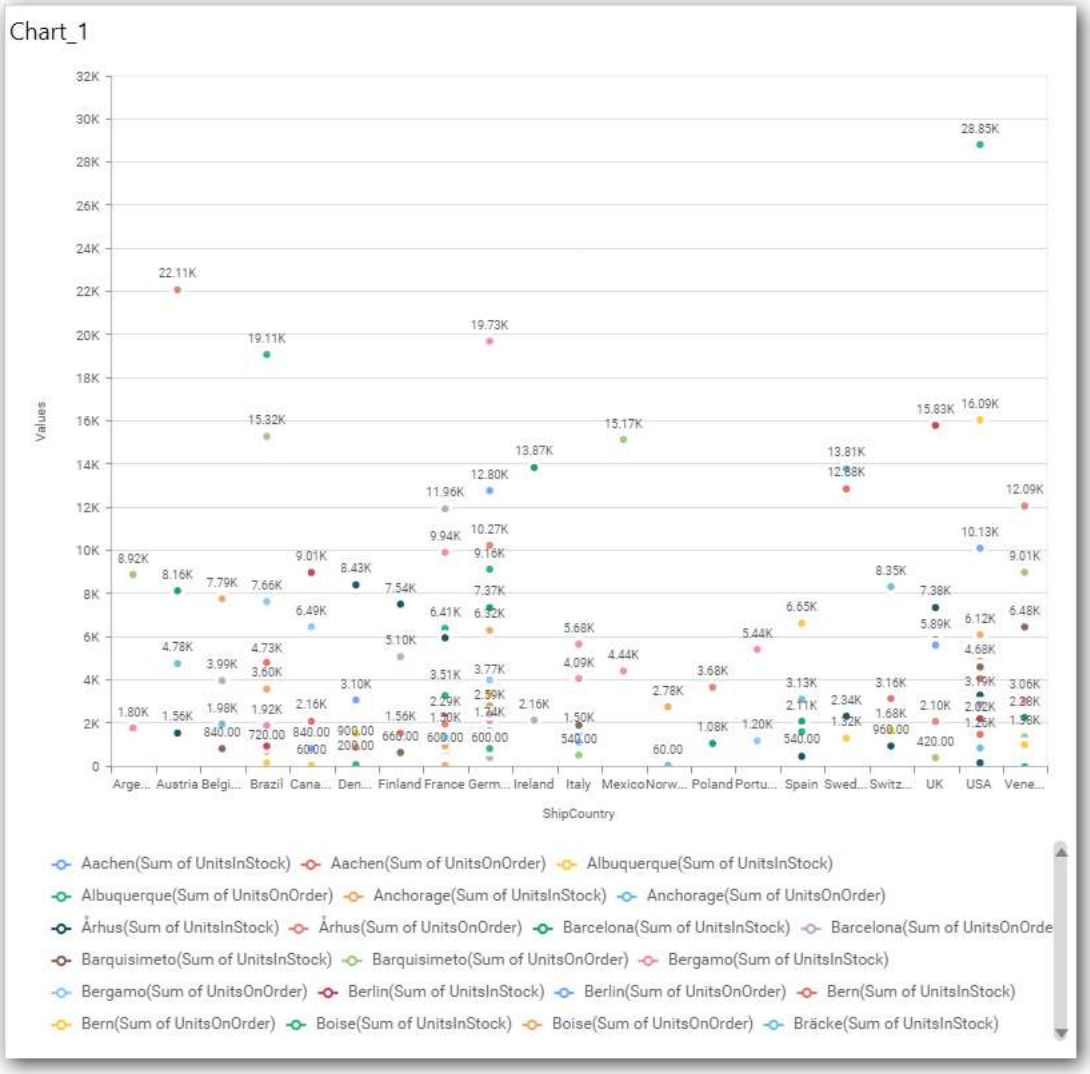


Assigning Row

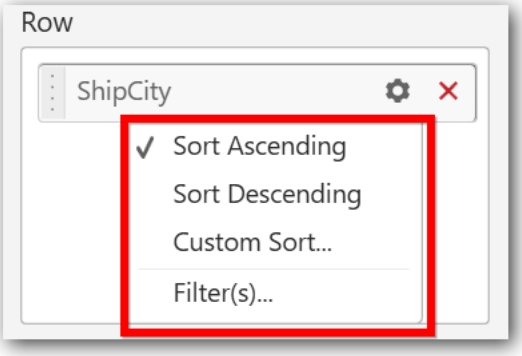
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s).



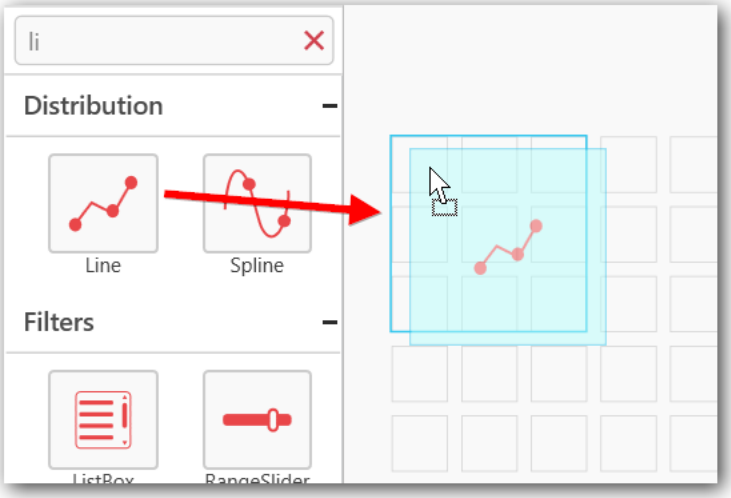
How to configure the SSAS data to Line Chart?

Line Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that

you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

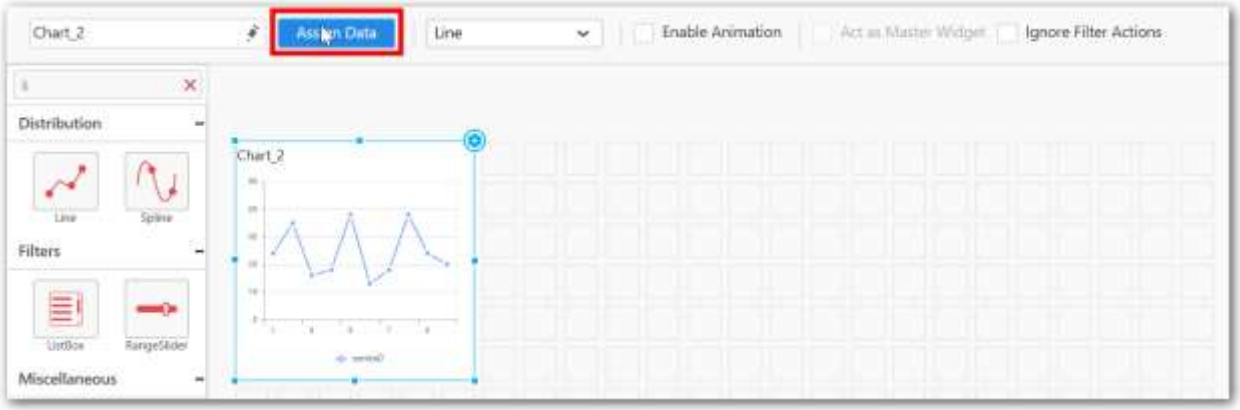
Following steps illustrates configuration of SSAS data to Line chart.

Drag and drop the Line chart to canvas and resize it to your required size.

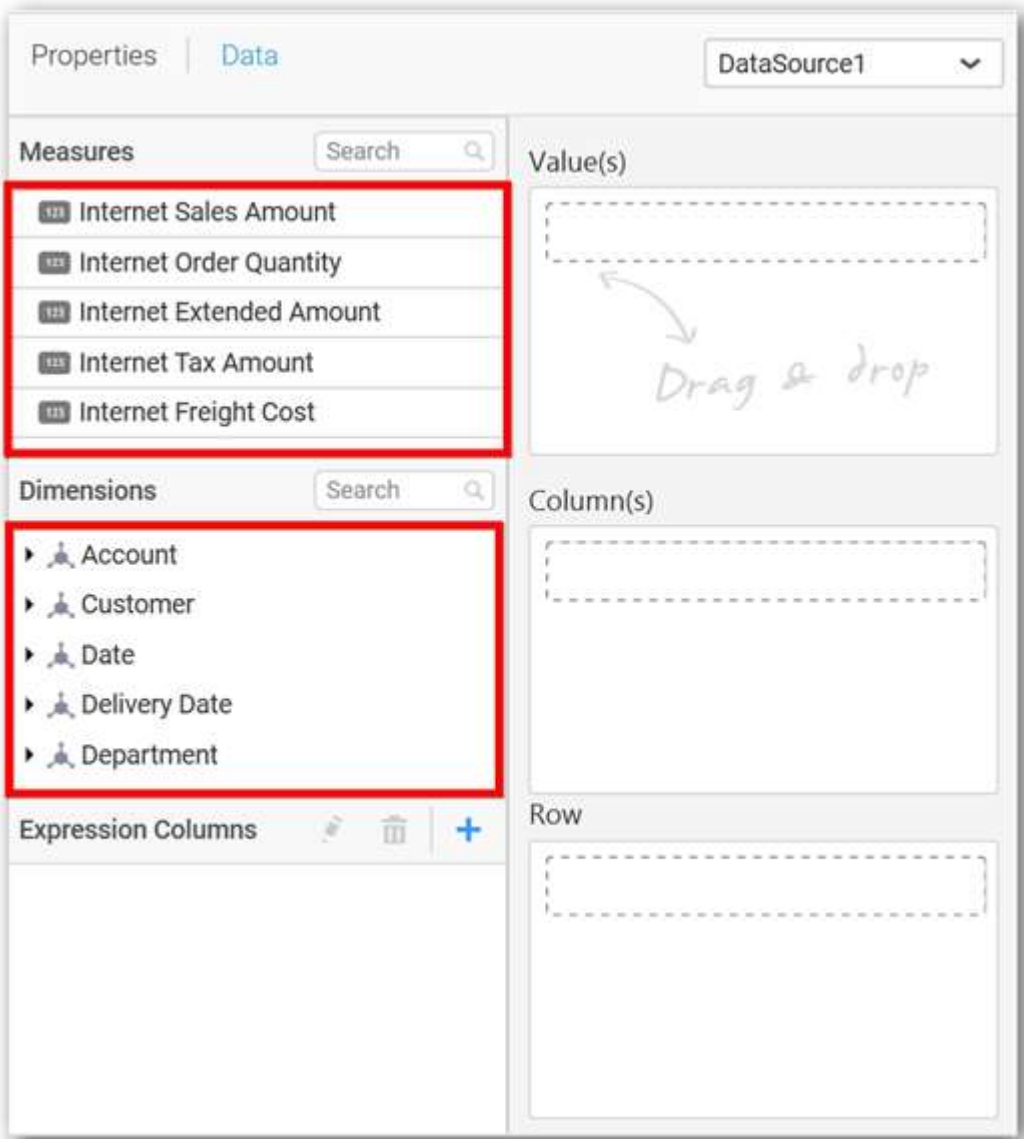


Select the dropped widget using mouse.

Click the Assign Data button in the toolbar.

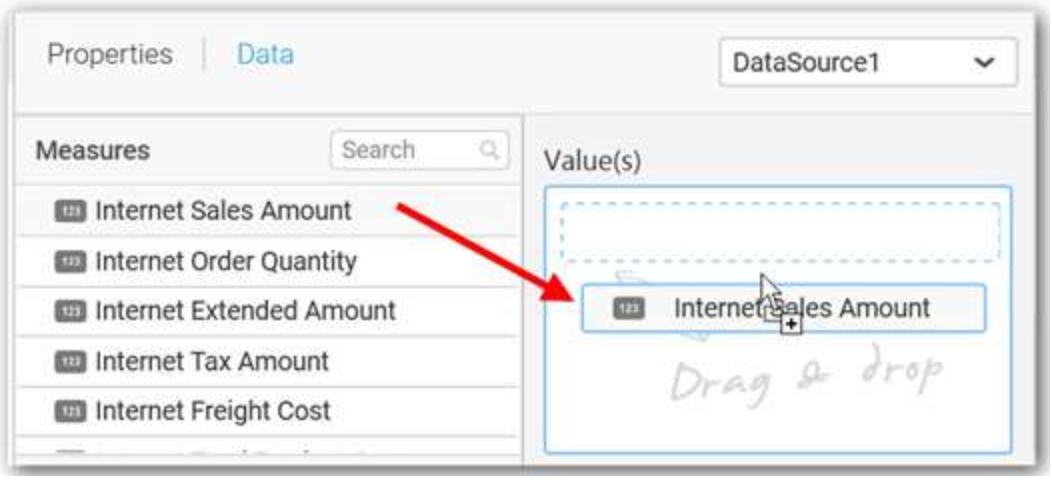


A Data pane will be opened with available Measures and Dimensions

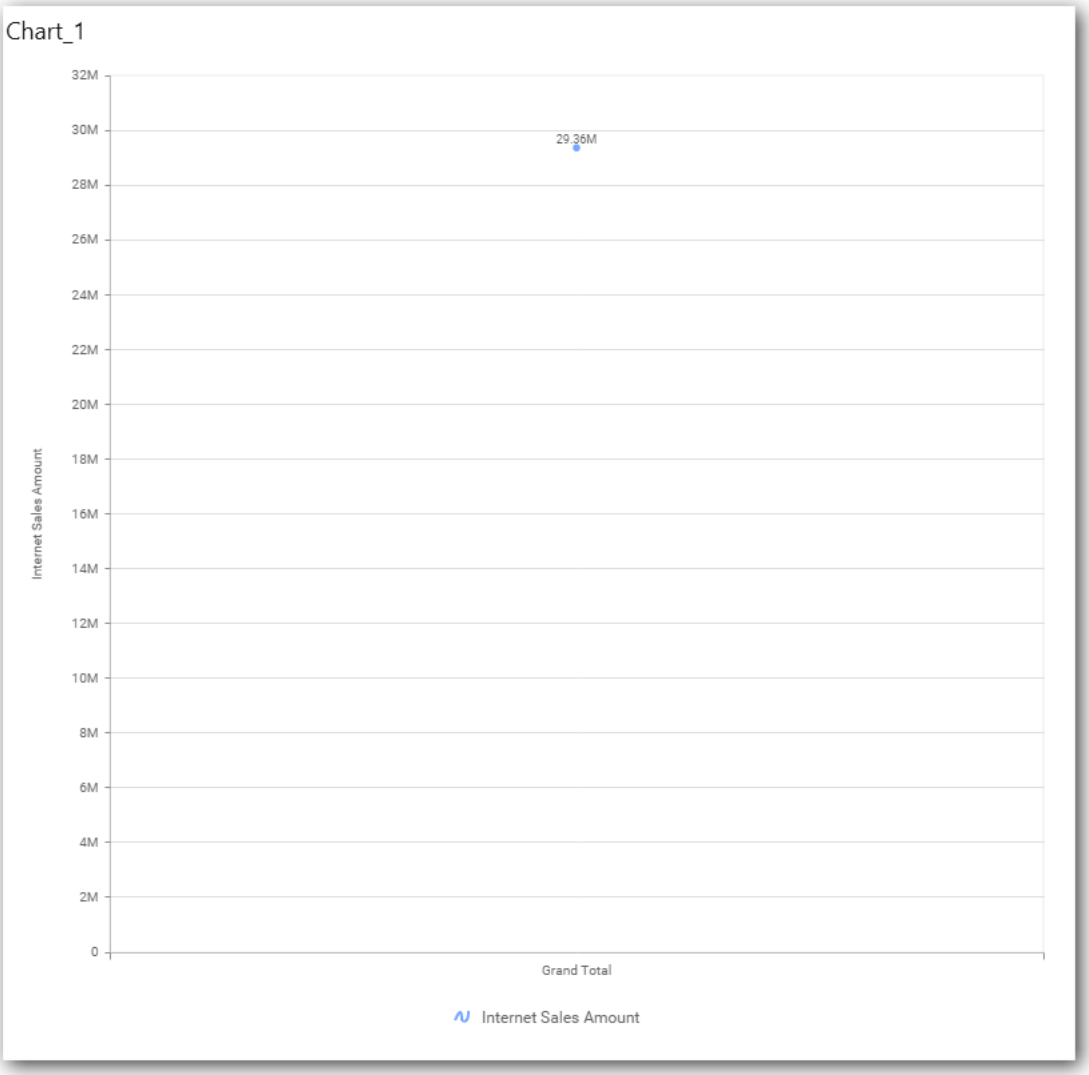


Assigning Value(s)

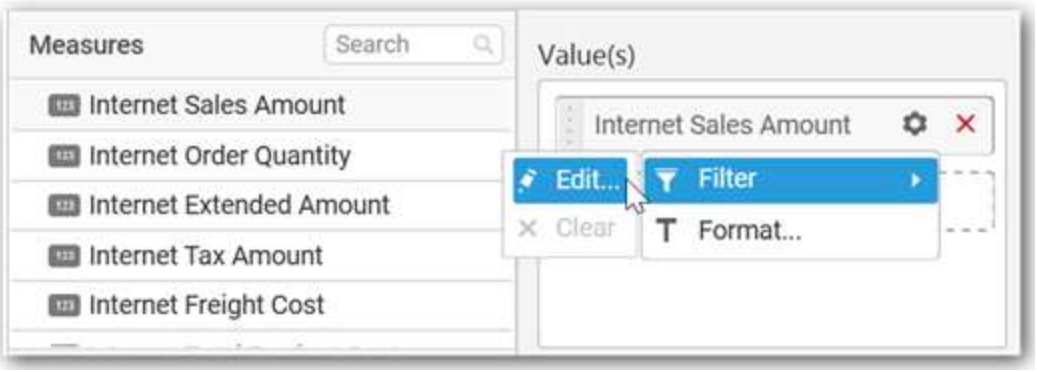
Drag and drop a column under Measures category into Value(s) section.



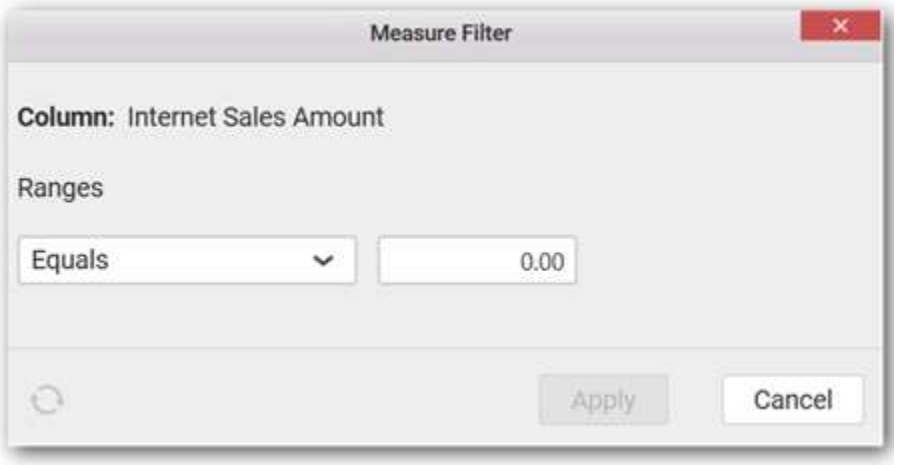
Now the chart will be rendered like this.



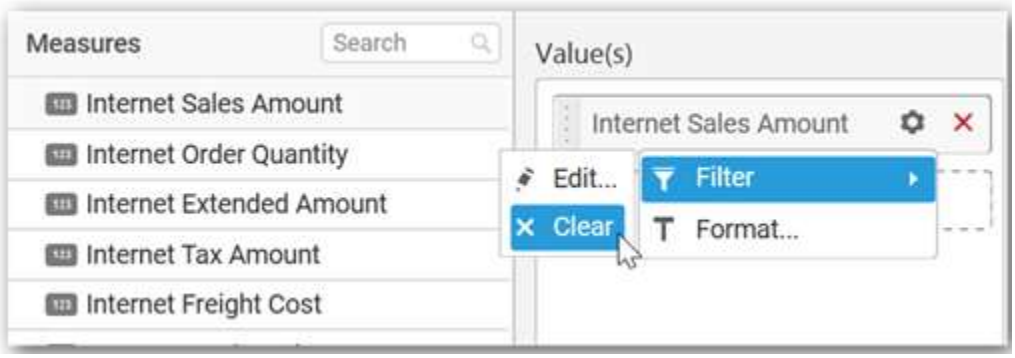
Define the filter criteria to match through choosing Edit option in Filter menu item.



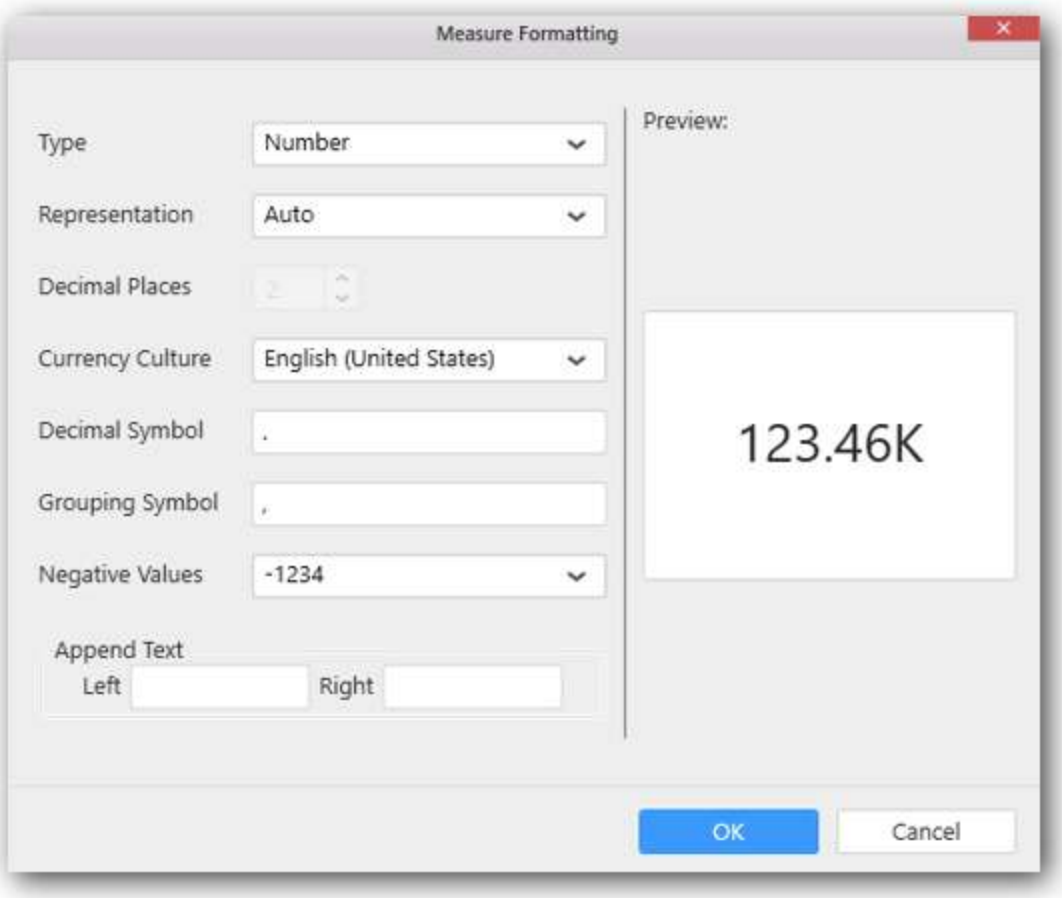
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



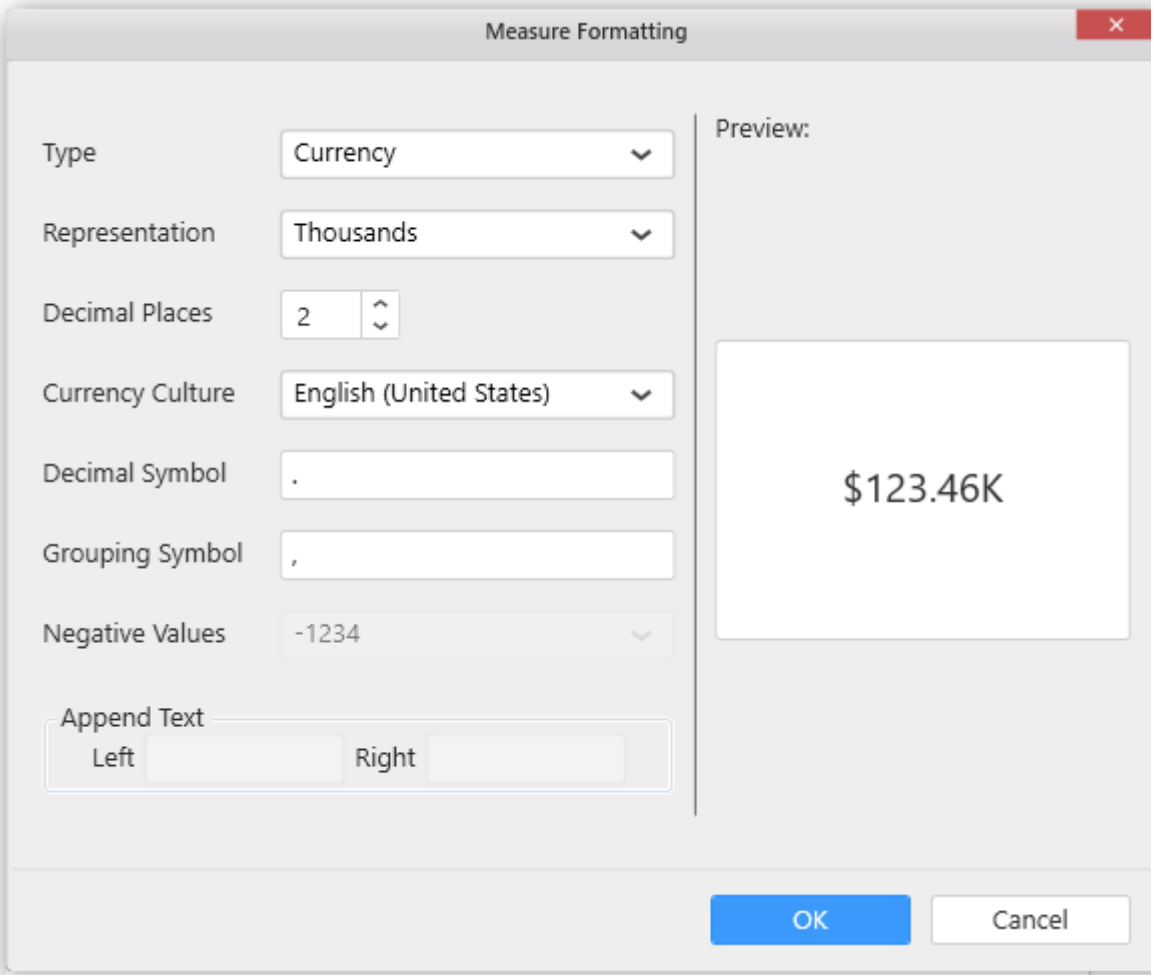
Select Clear option to clear the defined filter.



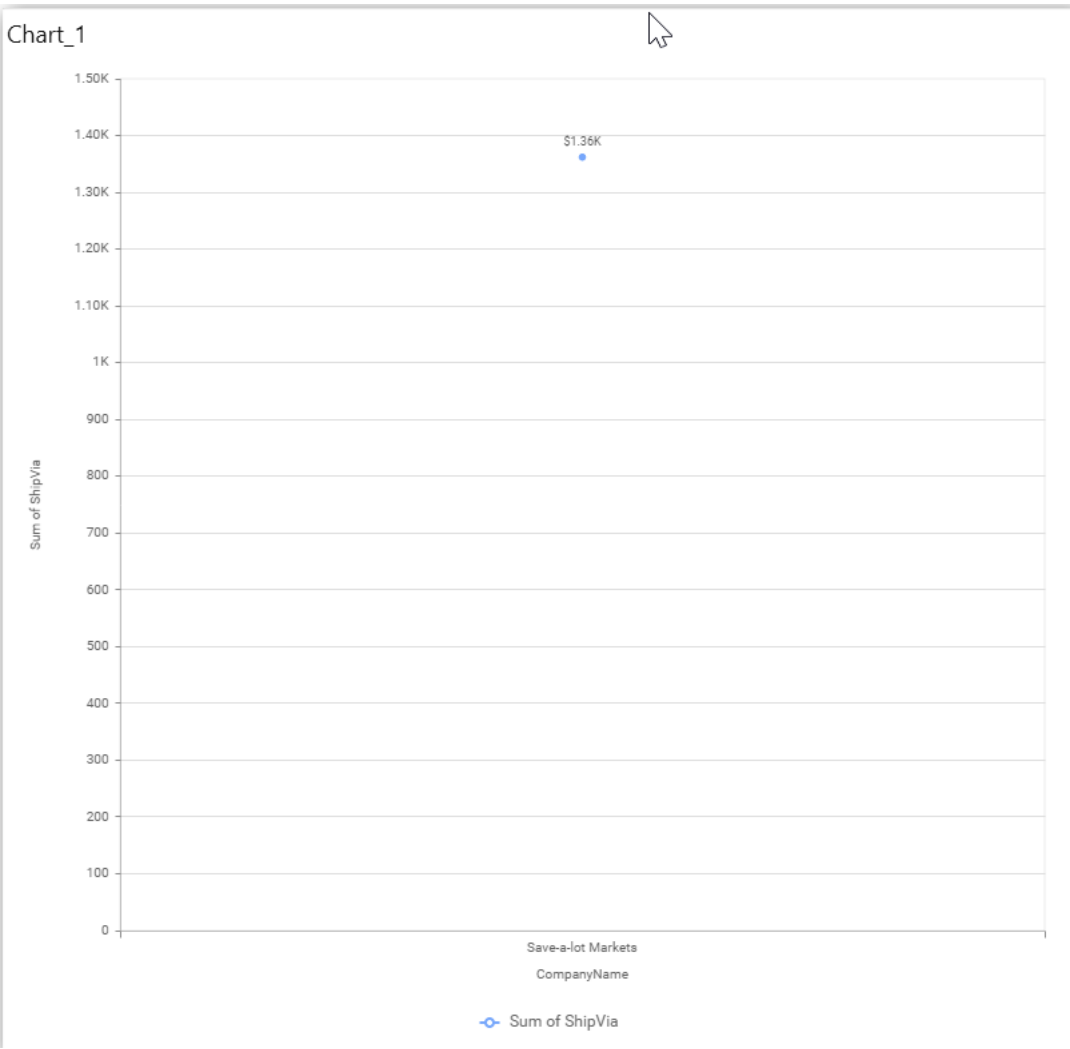
Select Format option to define the display format to the values in the column through Measure Formatting window.



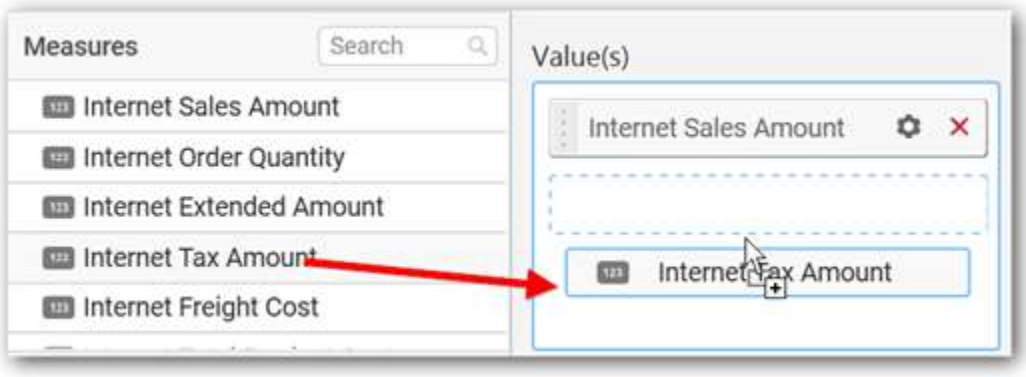
Choose the options you need and click OK.

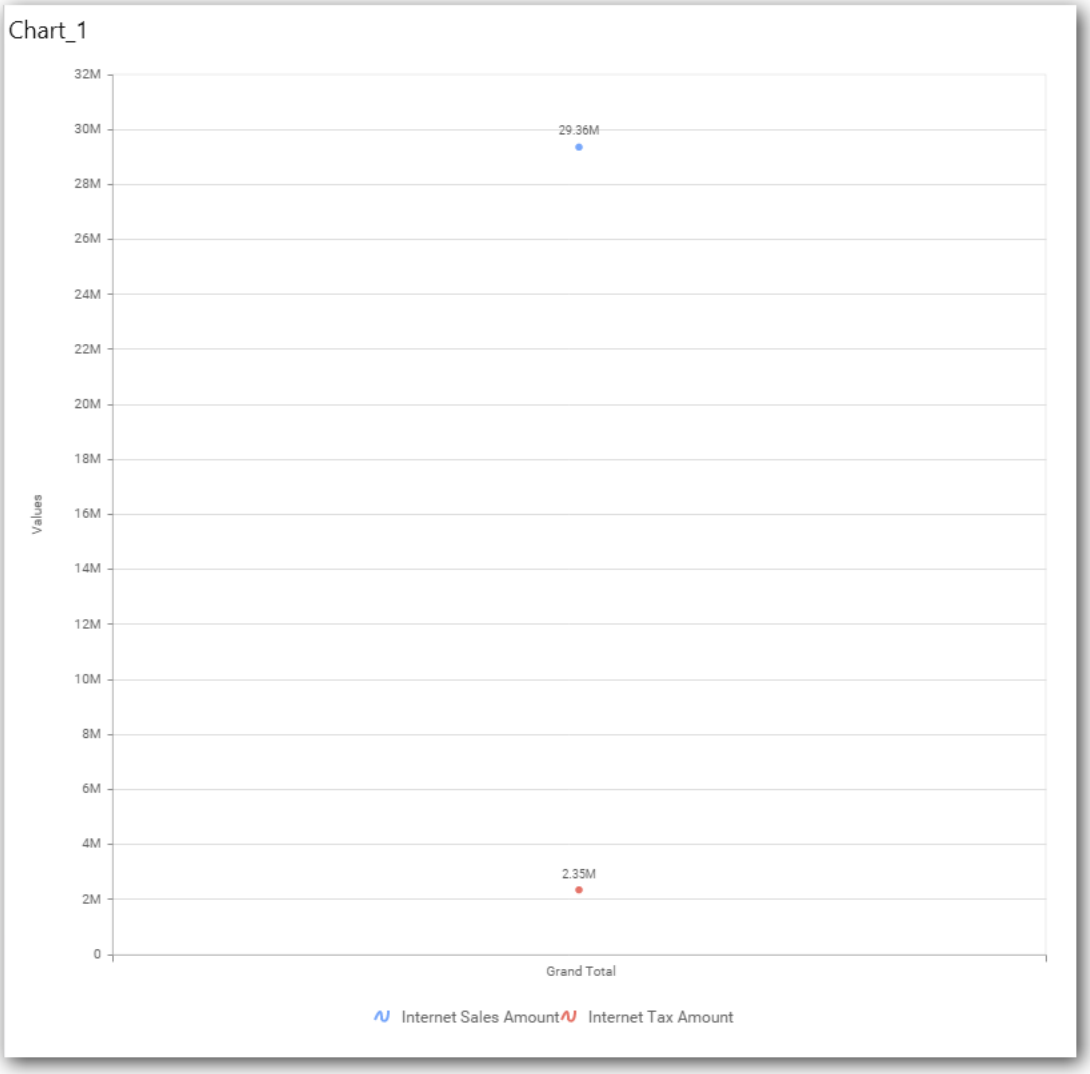


Now the Chart will be rendered like this.



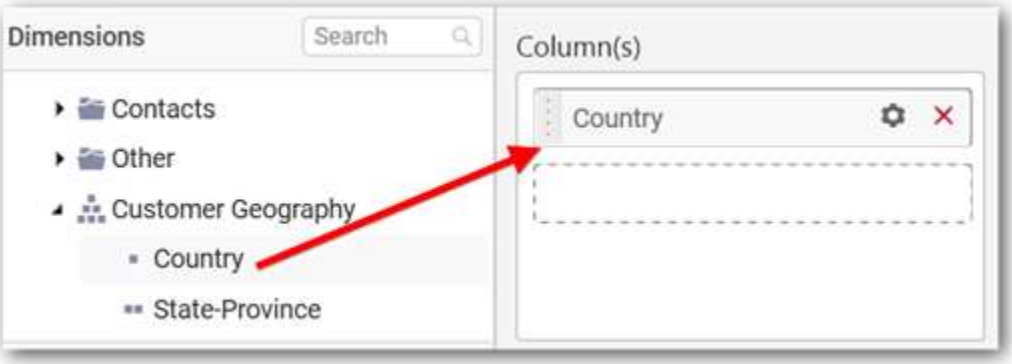
You can add more number values by drag drop the Measures into Value(s) field.

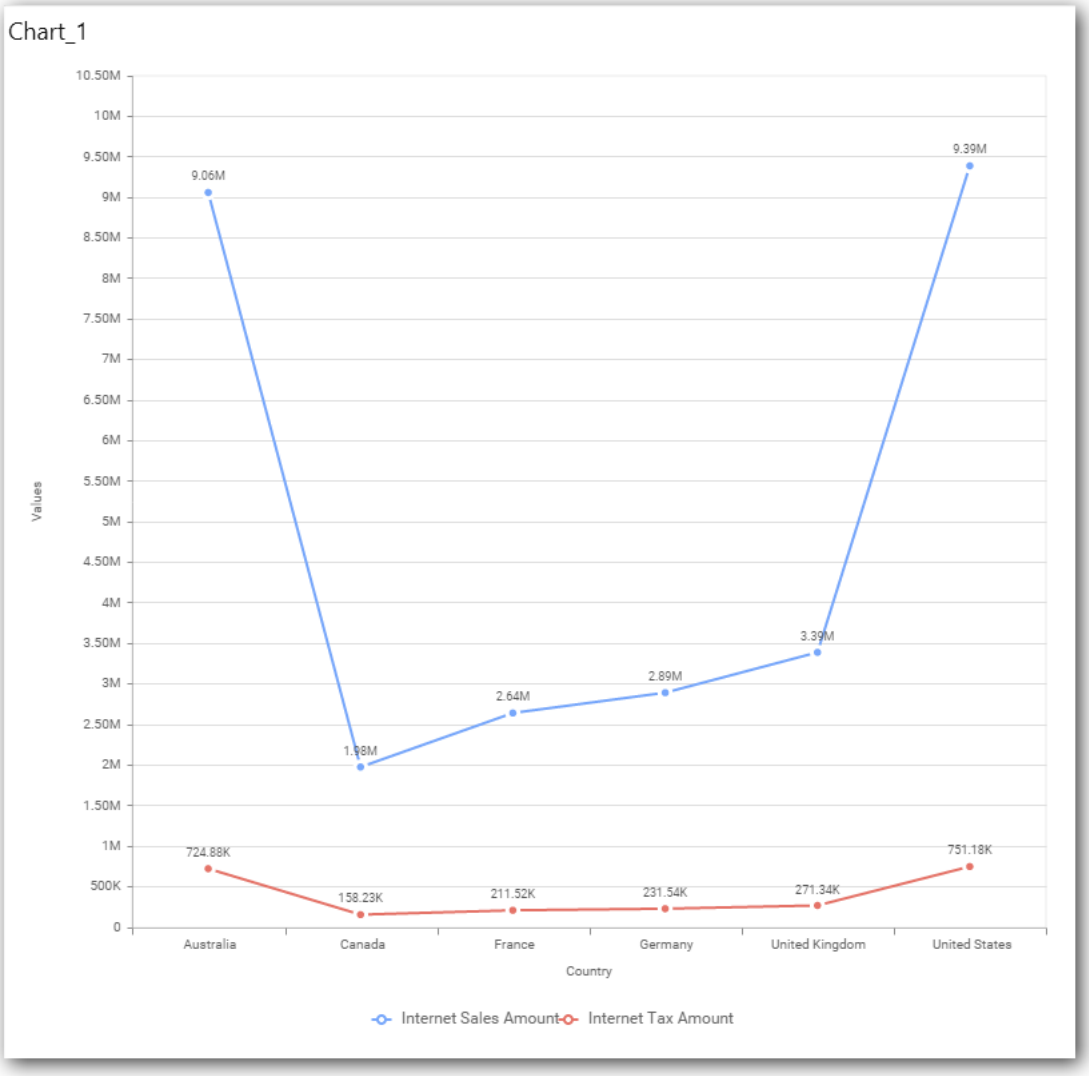




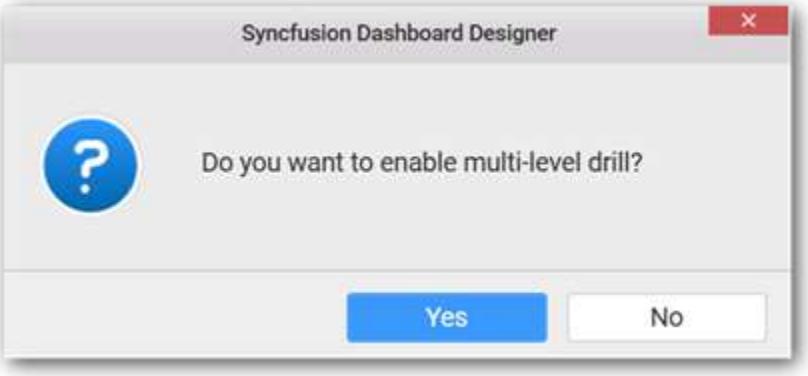
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.



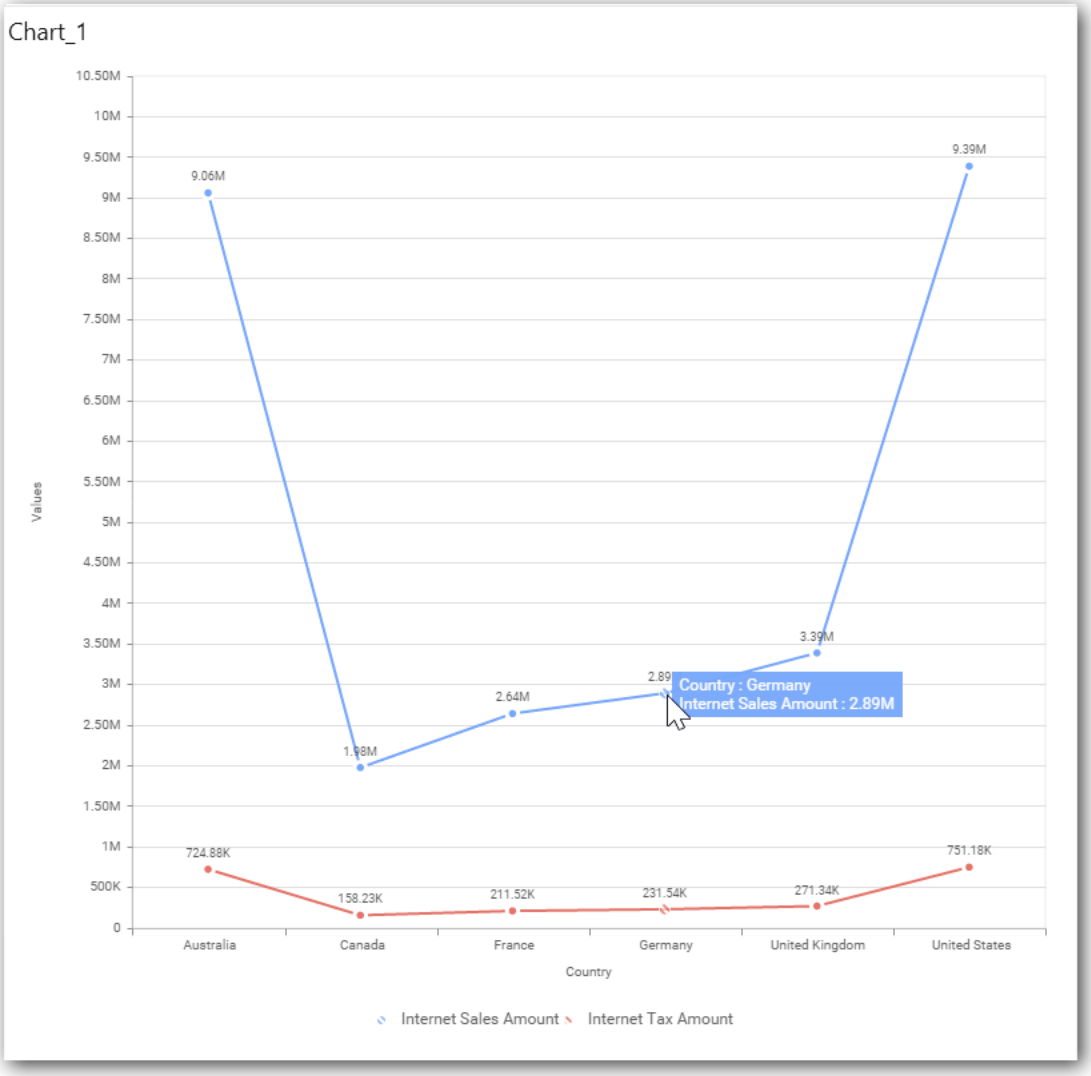


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

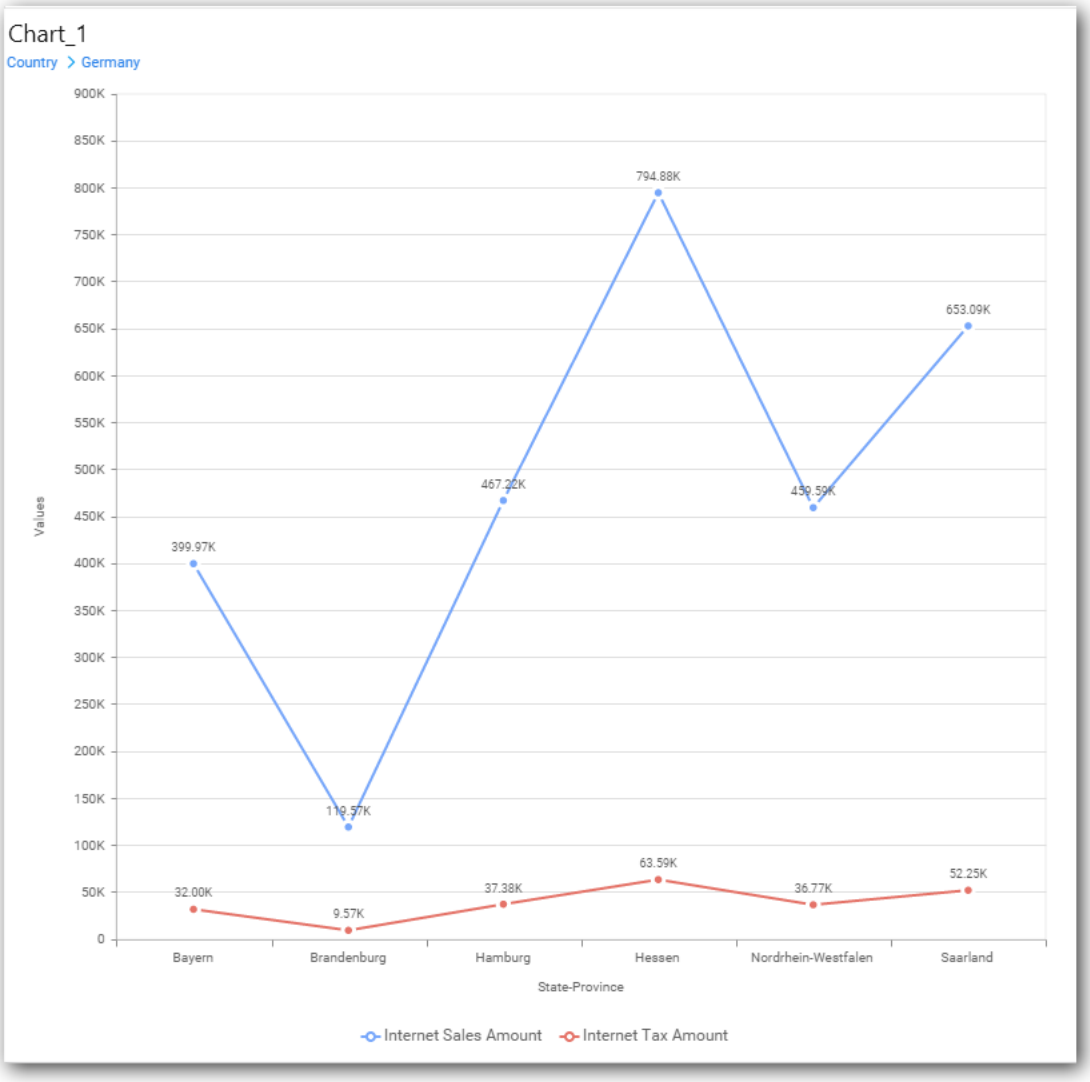


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

Click the respective data value marker in chart to drill into its inner level.

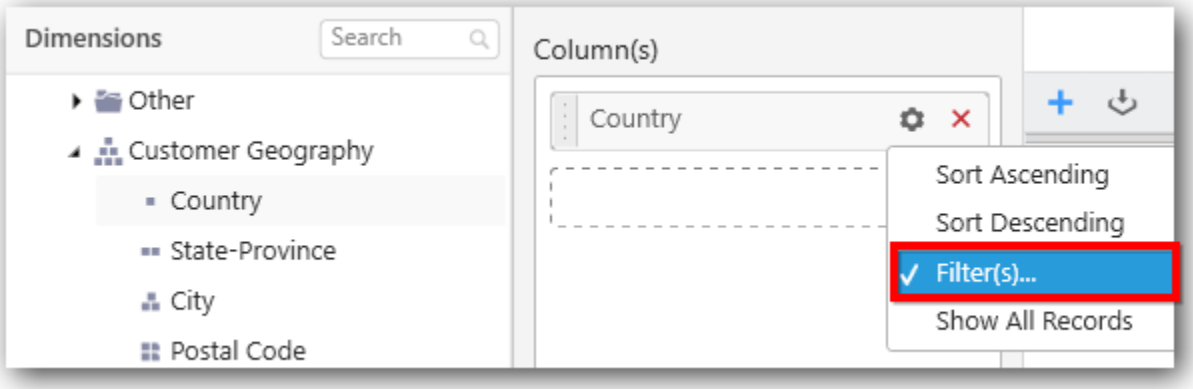


The drilled view of the chart is follows.

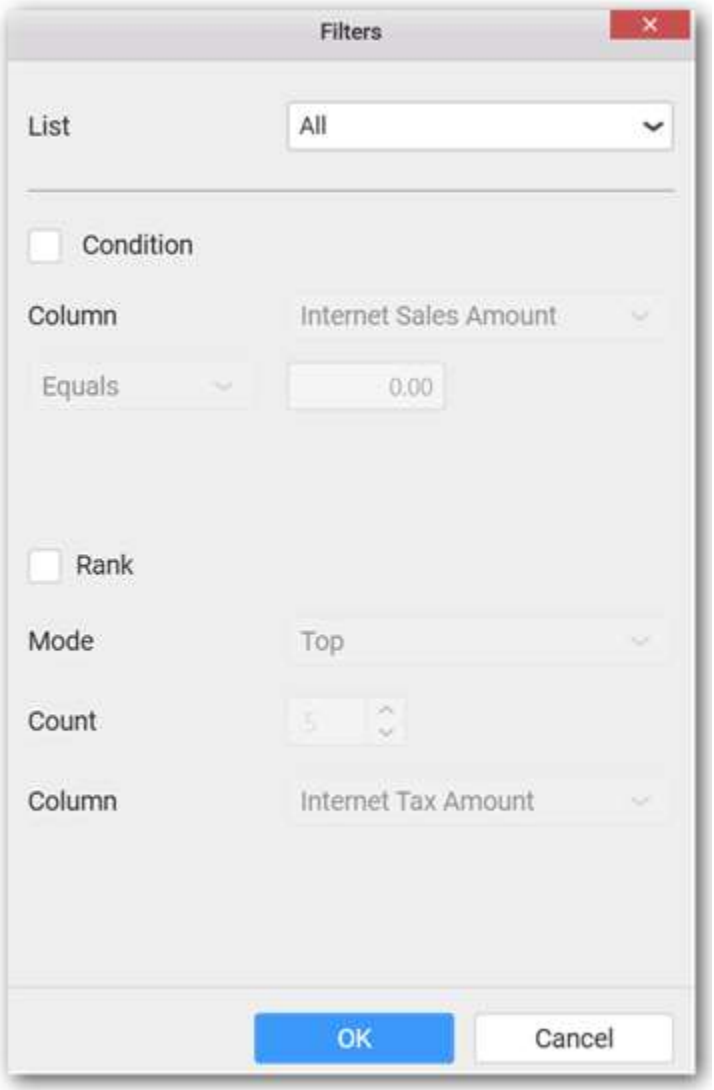


Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level

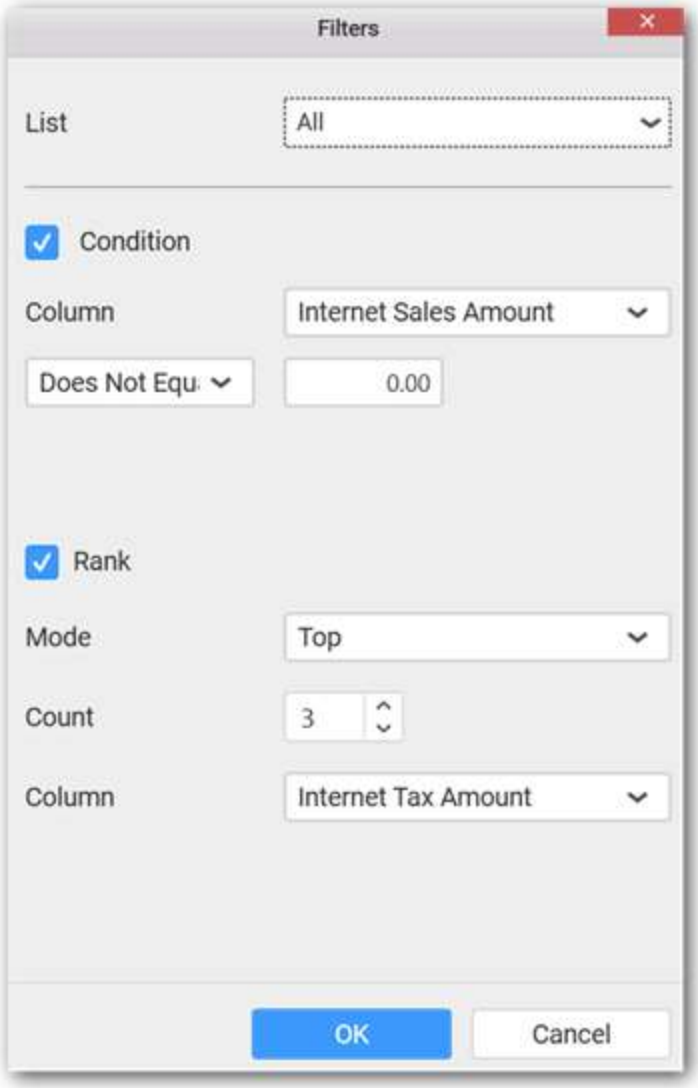
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



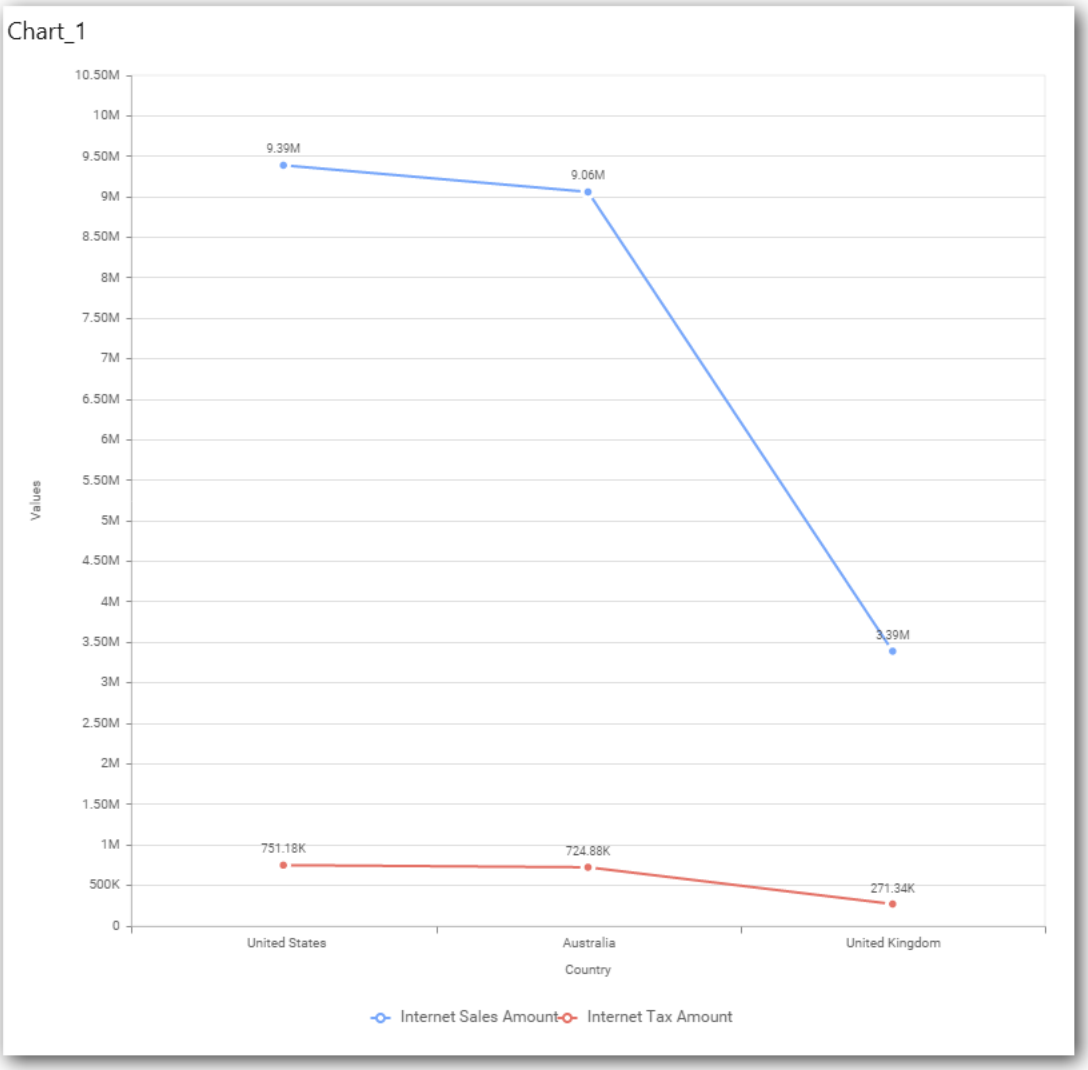
Select Filter(s)… to launch the Filters window.



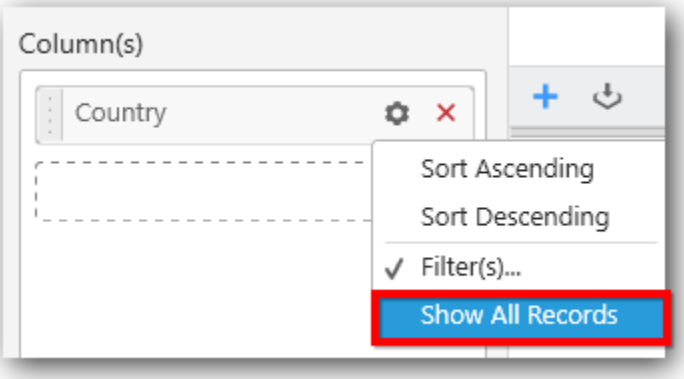
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

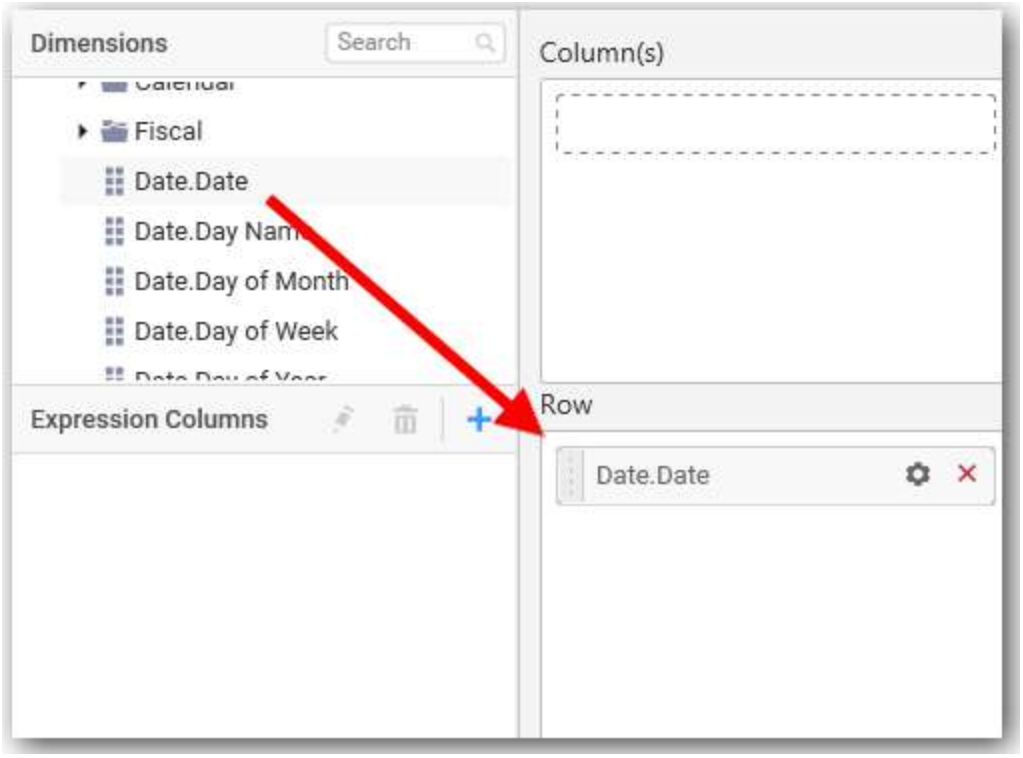


To show all records again click on Show All Records.

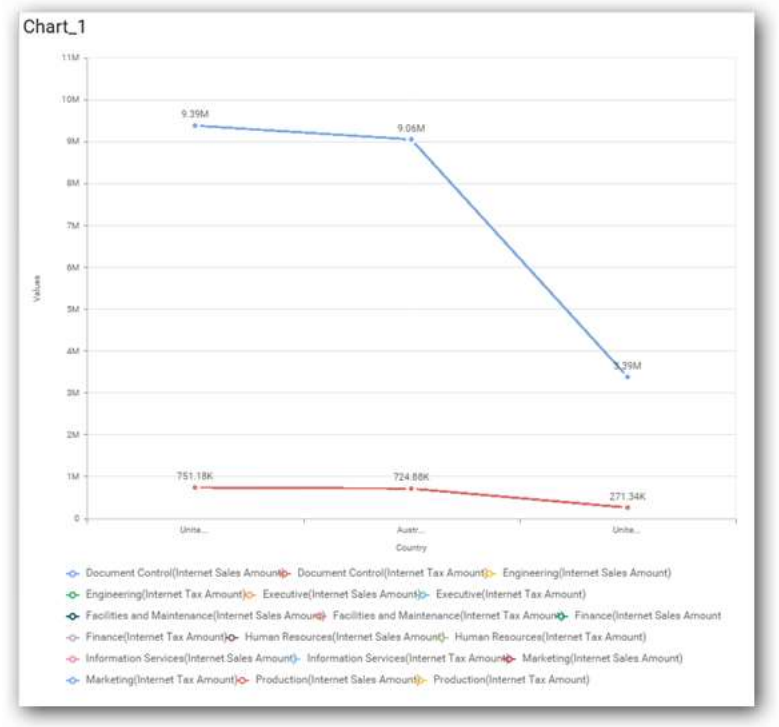


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.

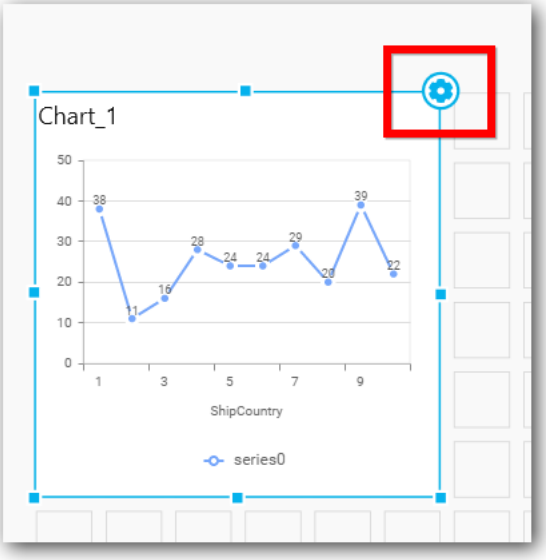


How to format Line Chart?

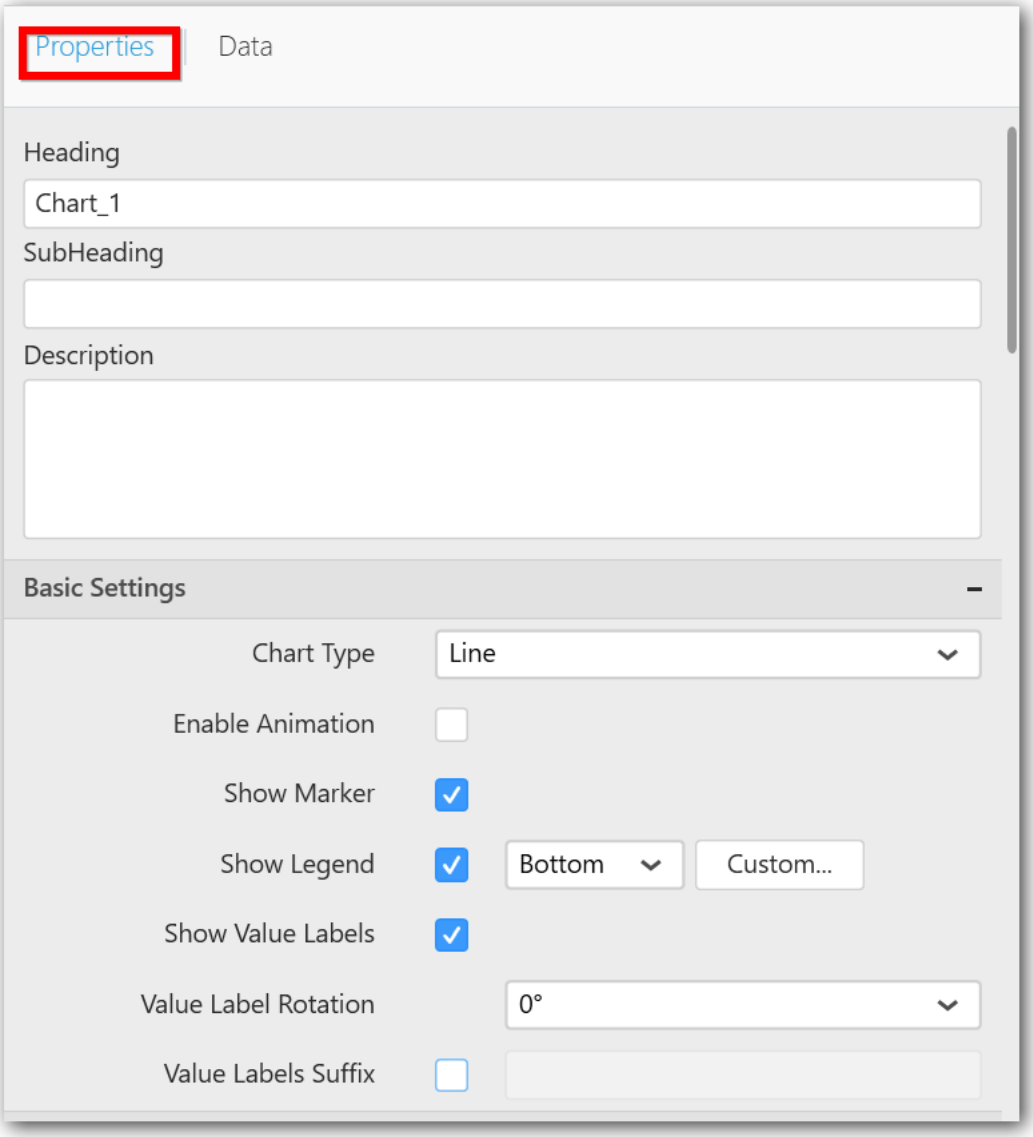
You can format the Line chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into line chart follow the steps

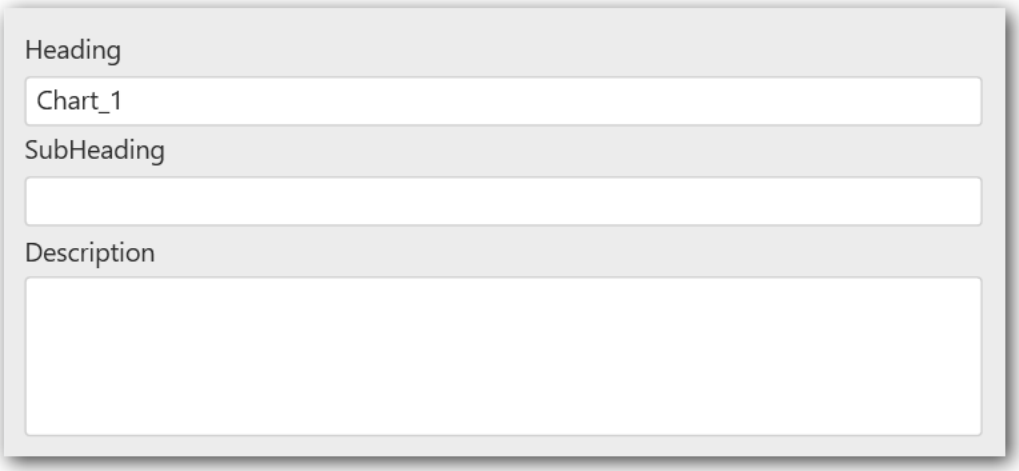
1. Drag and drop the line chart into canvas and resize it to your required size.
2. Configure the data into line chart.
3. Focus on the line chart and Click on Widget Settings.



The property window will be opened



General Settings



Header

This allows you to set title for this line chart widget.

SubHeading

This allows you to set sub-title for this line chart widget.

Description

This allows you to set description for this line chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

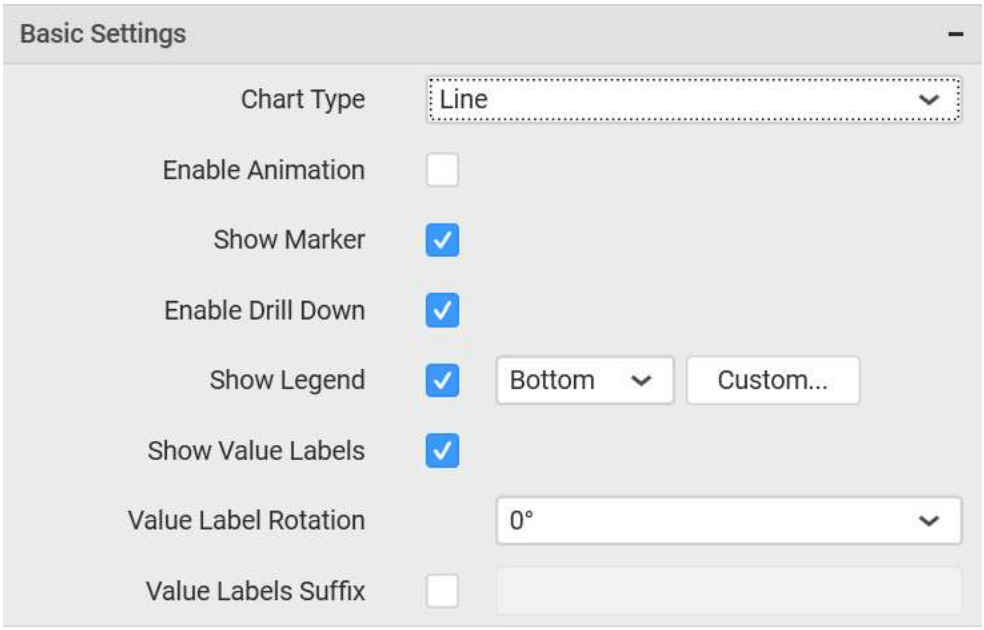


Chart Type

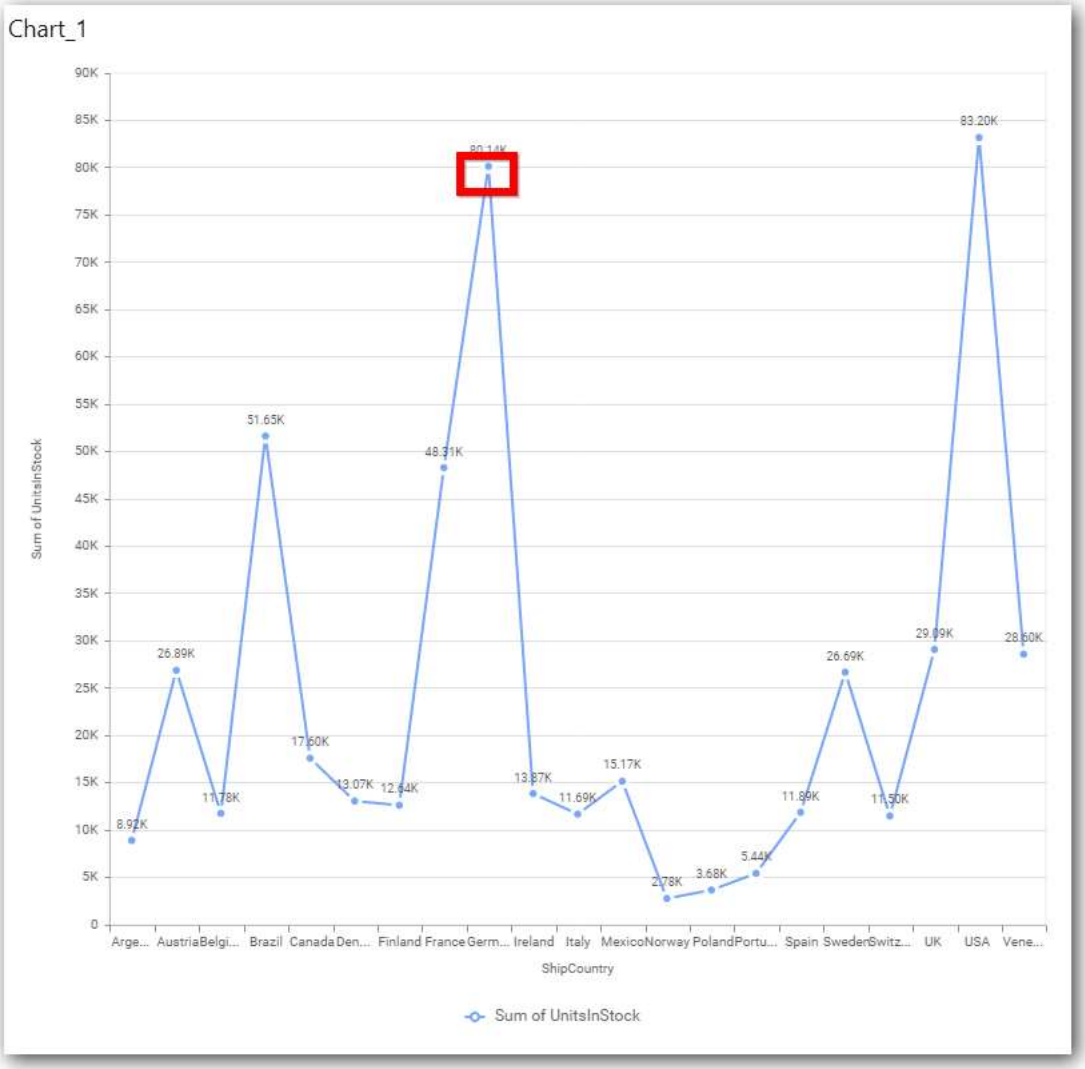
This allows you to switch the widget view from current chart type to another chart type. To selecting chart type through combo box.

Enable Animation

This allow to enable option the rendering of series in animated mode.

Show Data Marker

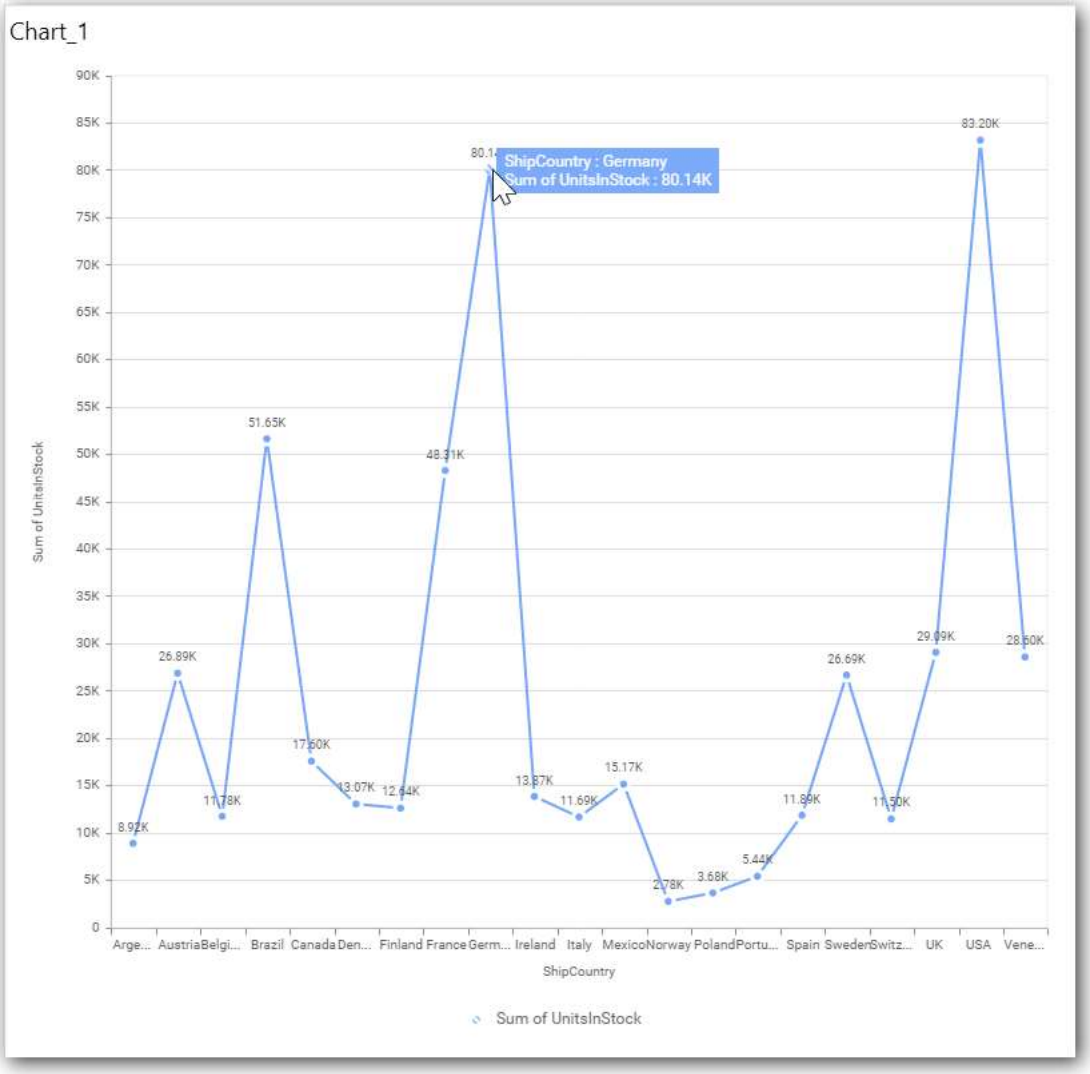
This allows you to toggle the visibility of marker from label to adorn each data point in chart series.



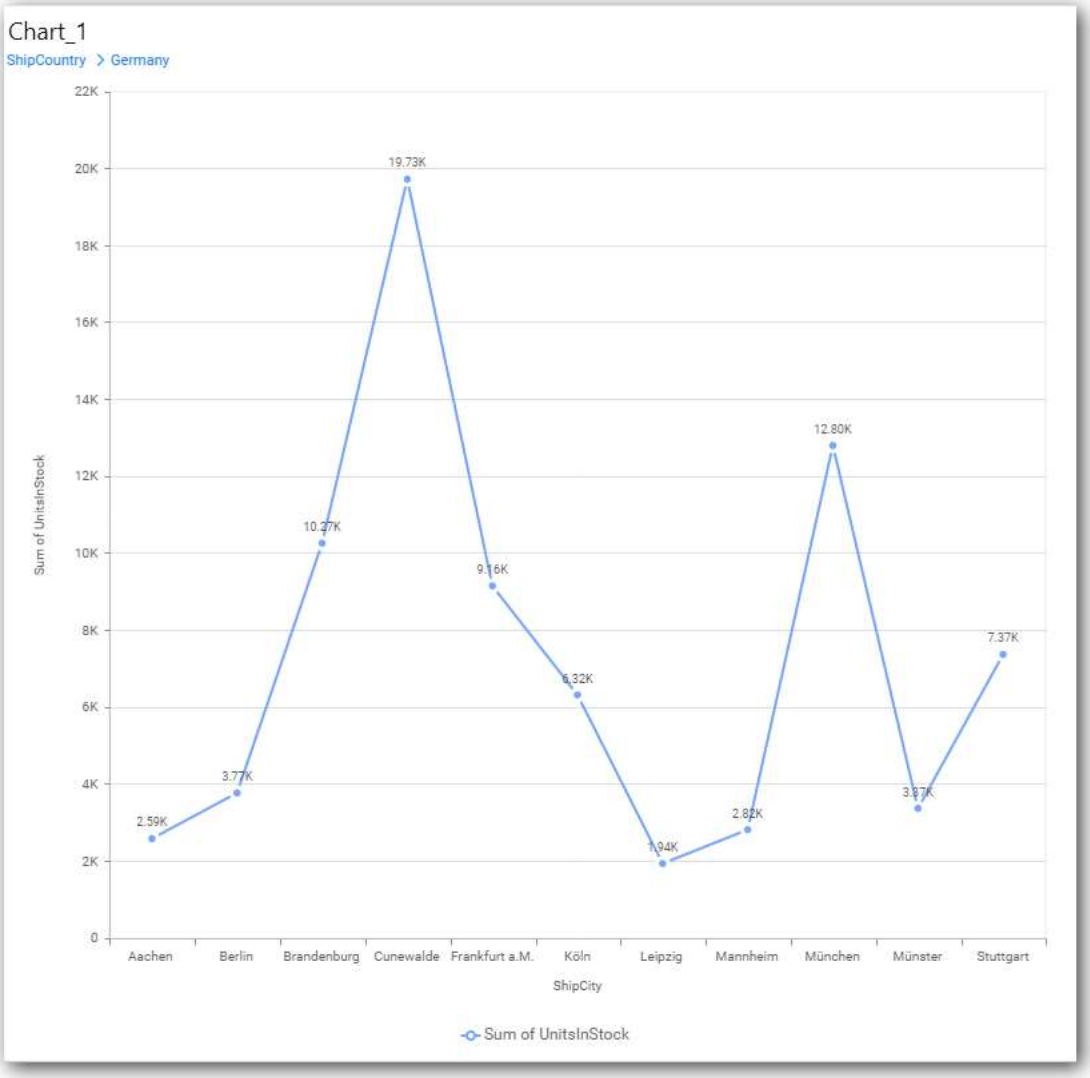
Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View

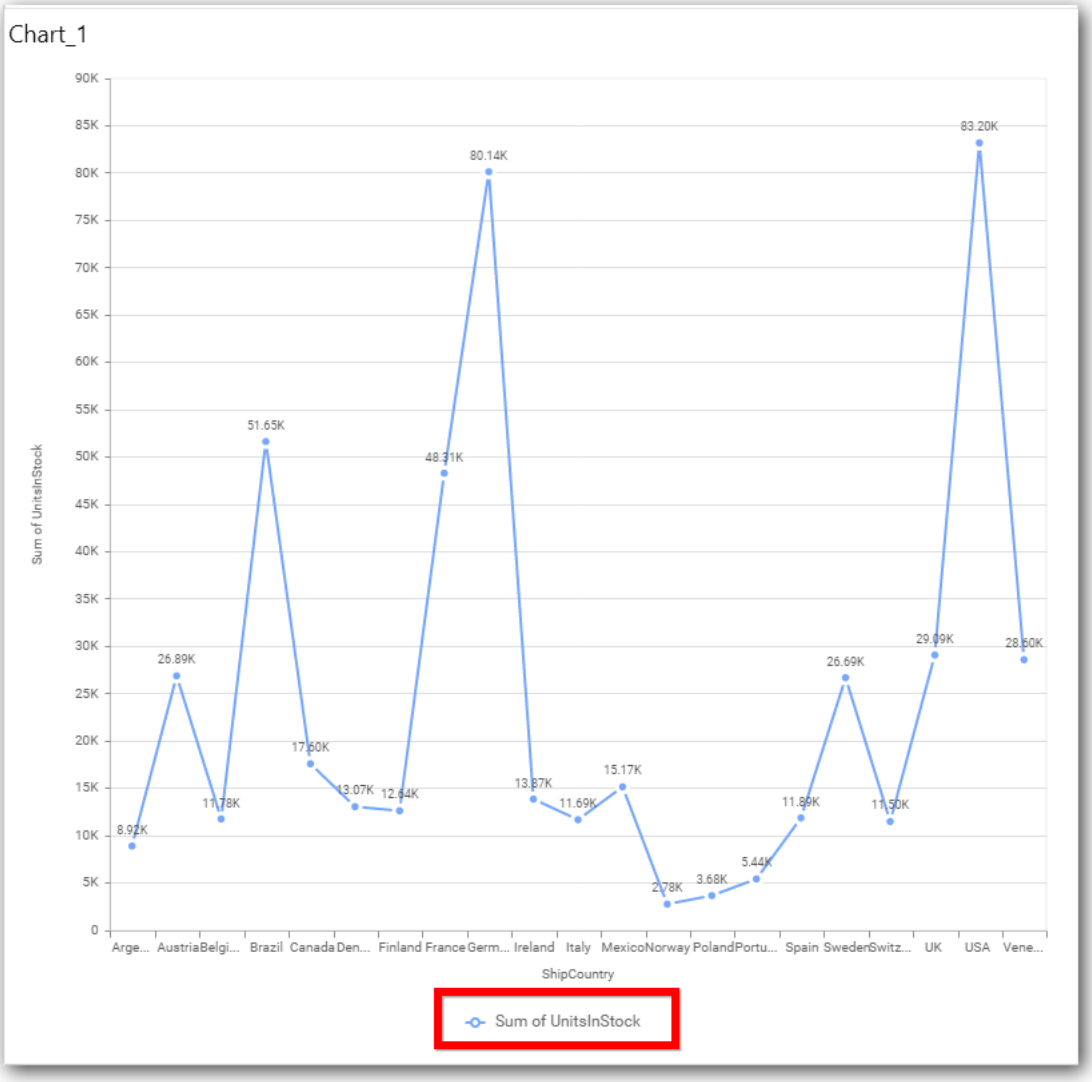


Drilled View



Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

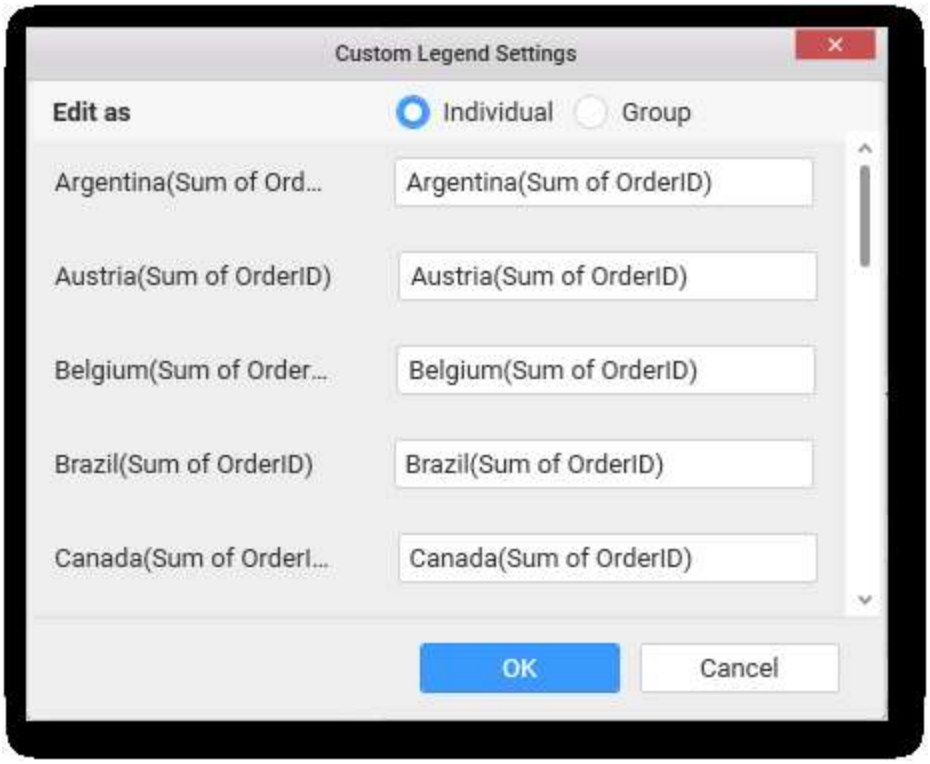
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

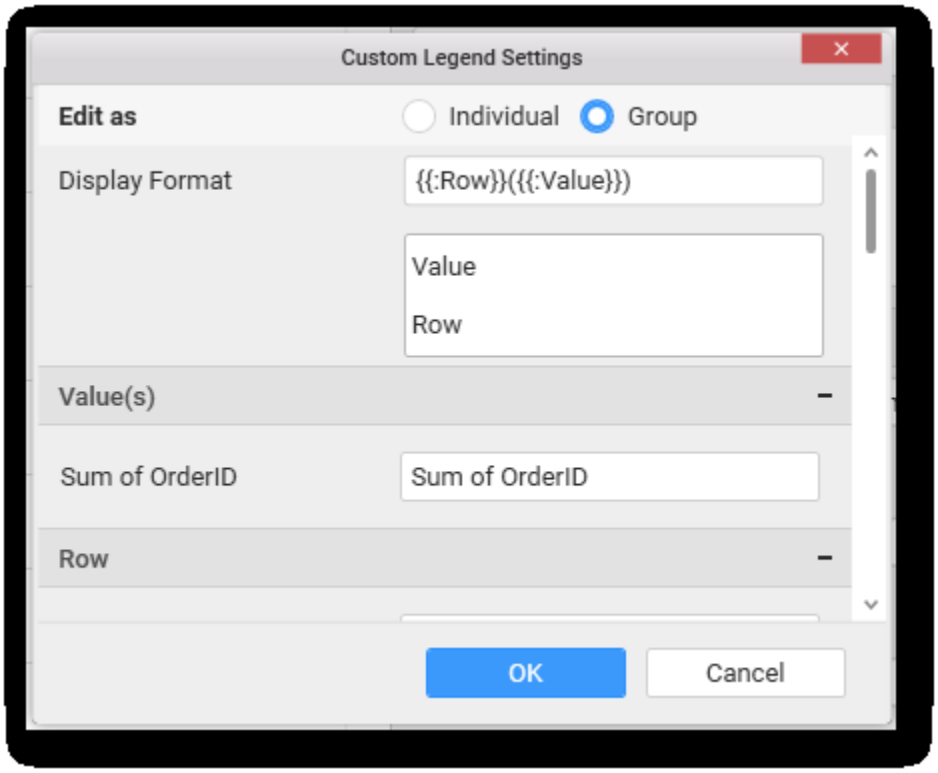
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

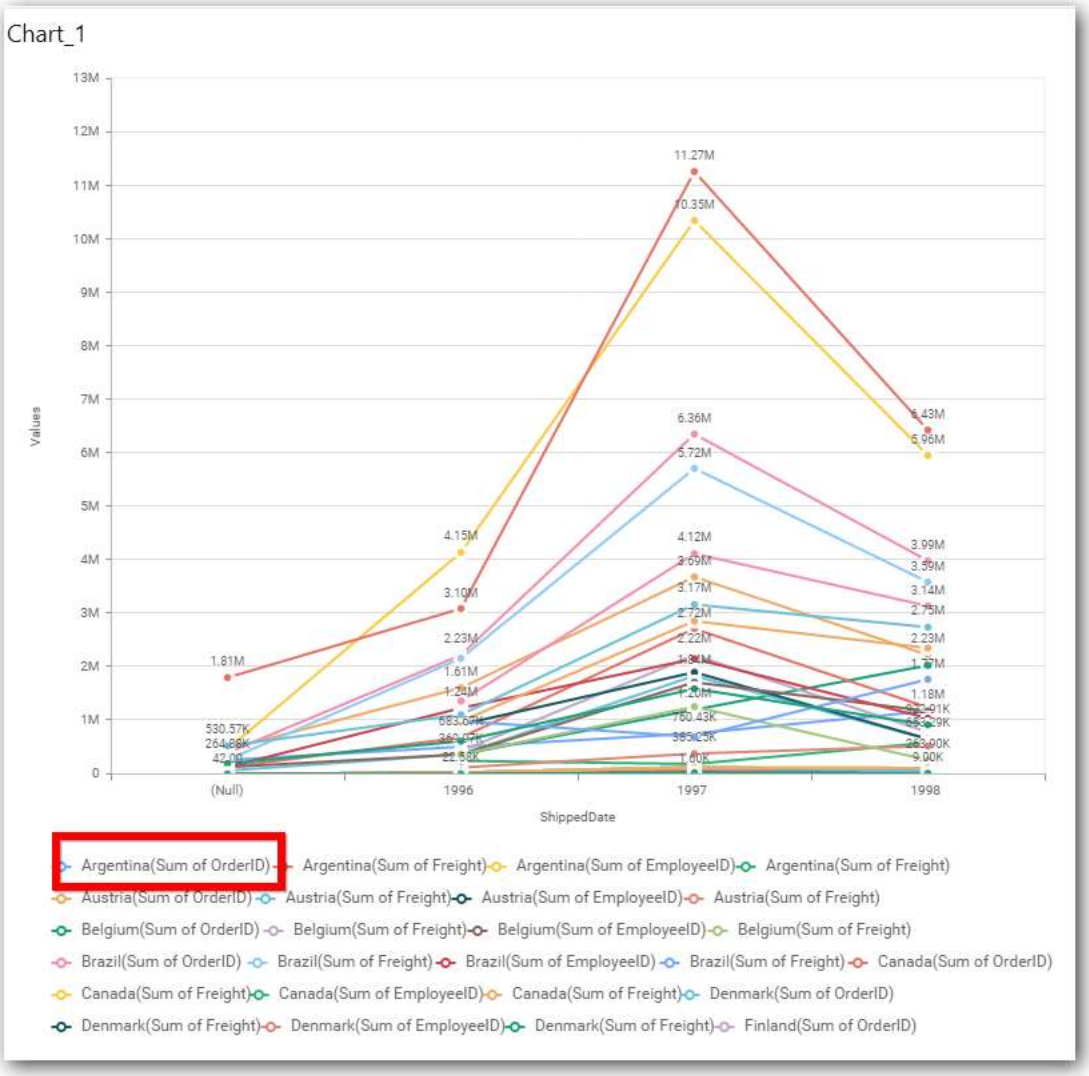


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

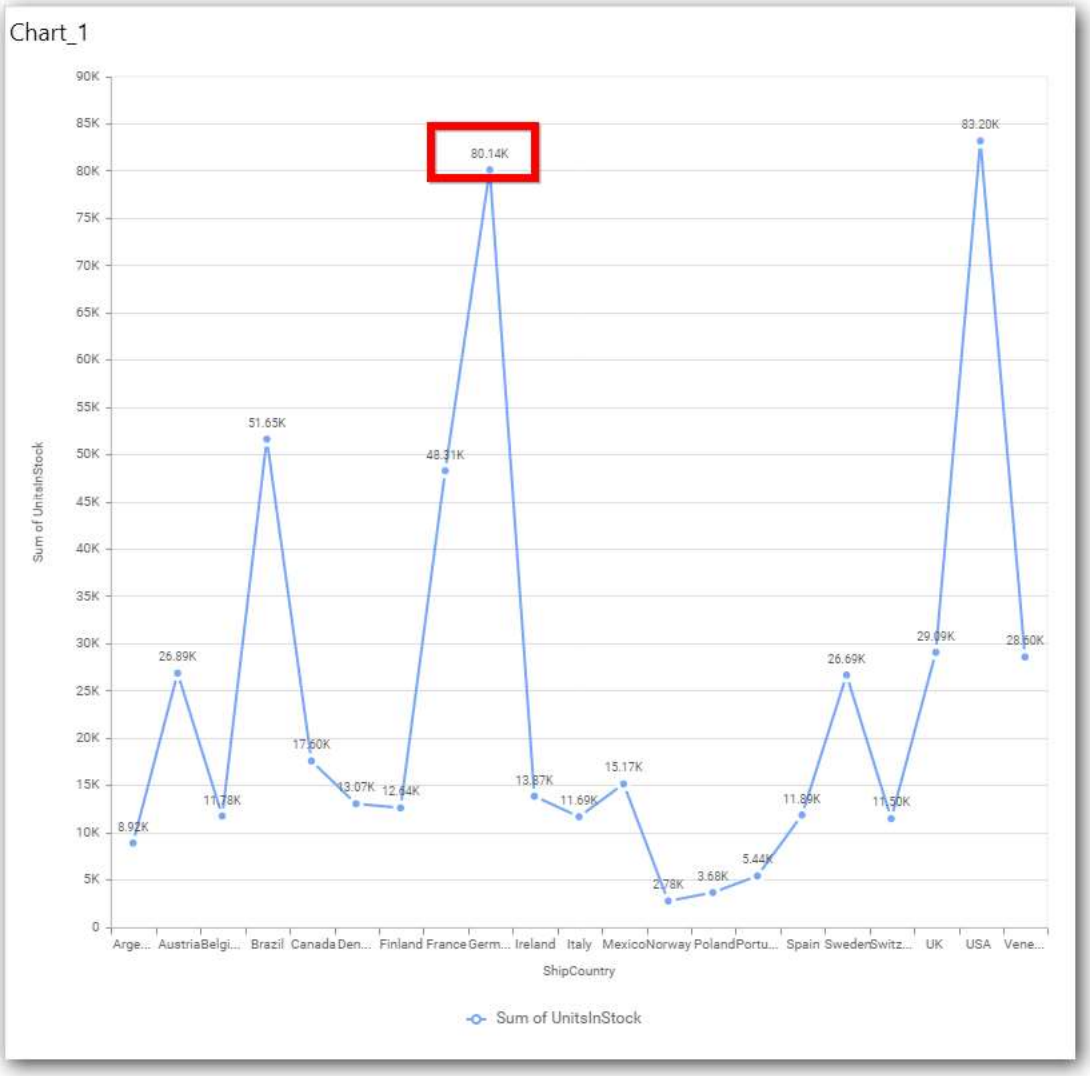


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



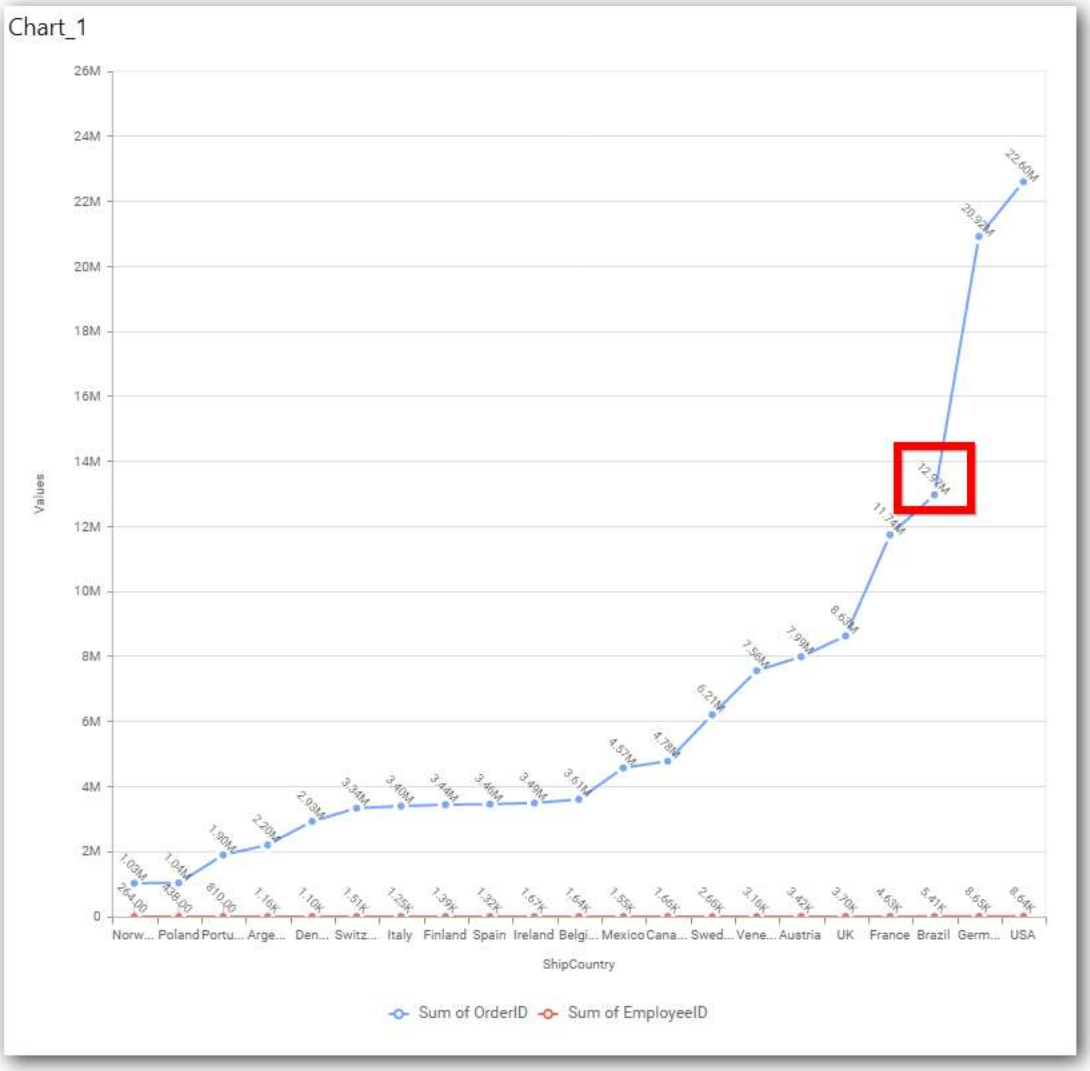
Show Value Labels

This allows you to toggle the visibility of value labels.



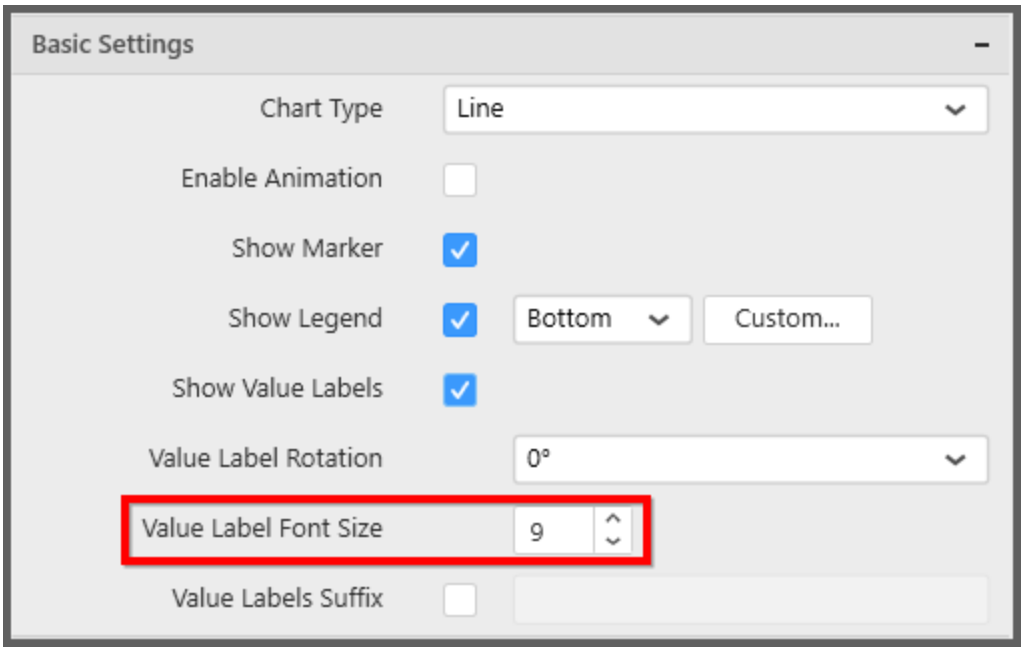
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



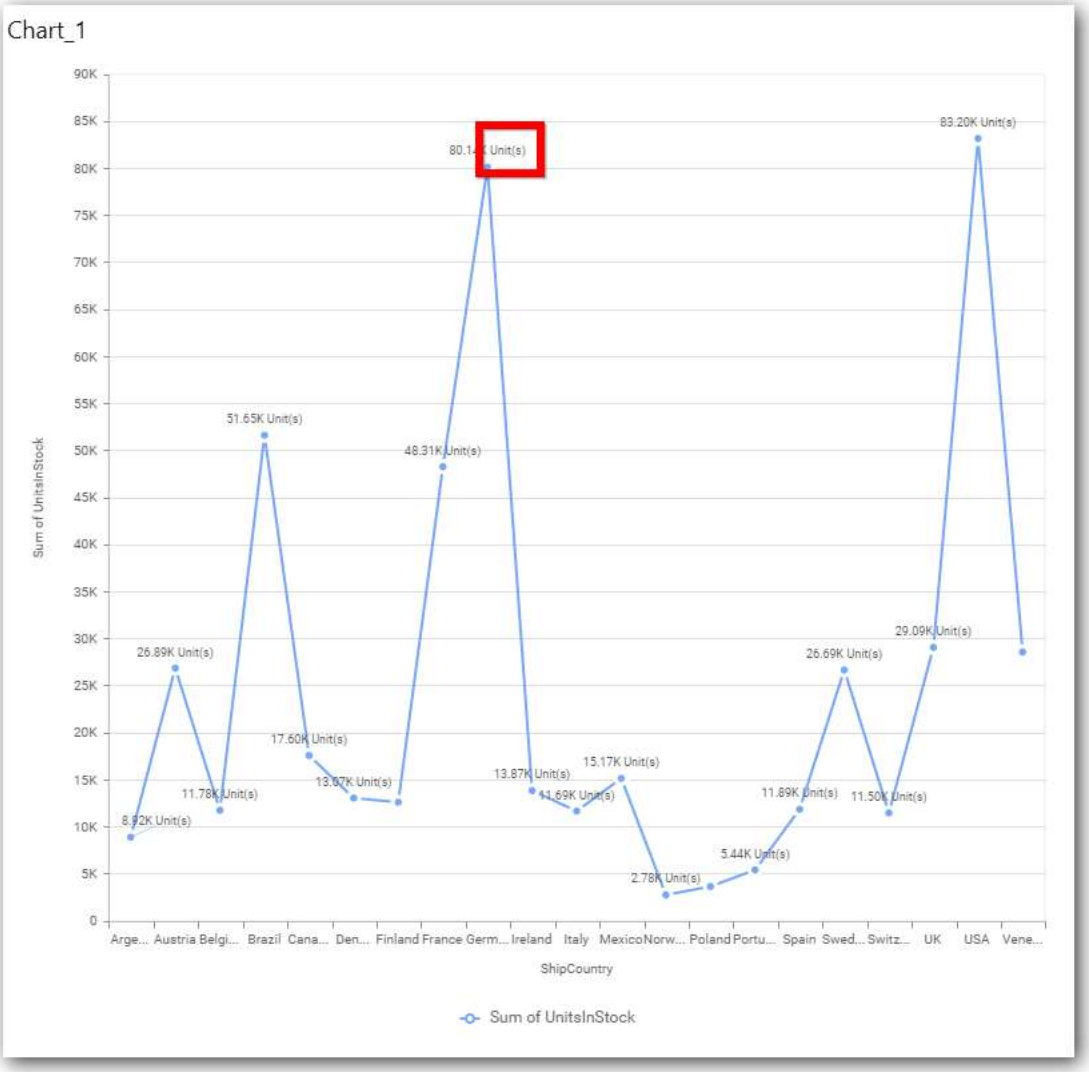
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

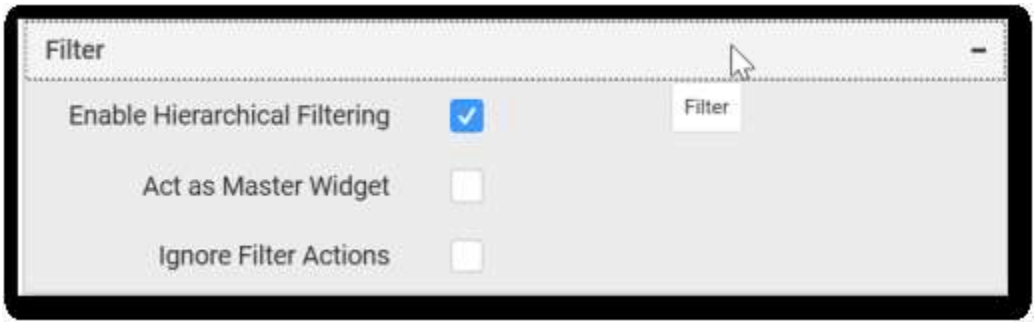


Value Labels Suffix

This allows you to set suffix to the value labels.



Filter Settings



Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical, in this line chart widget.

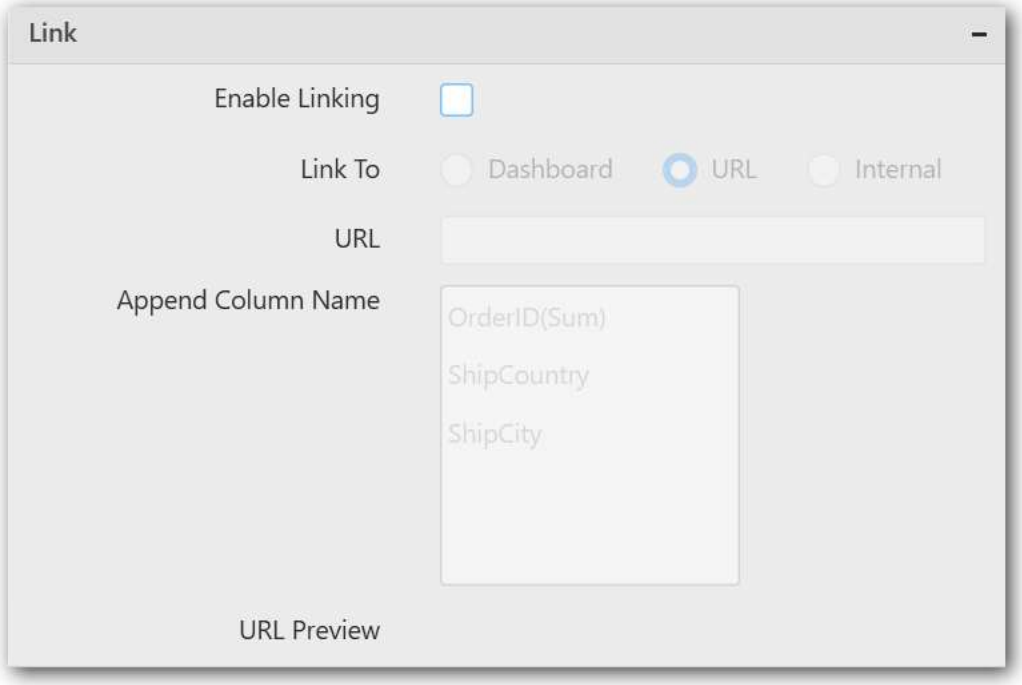
Act as Master Widget

This allows you to define this line chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

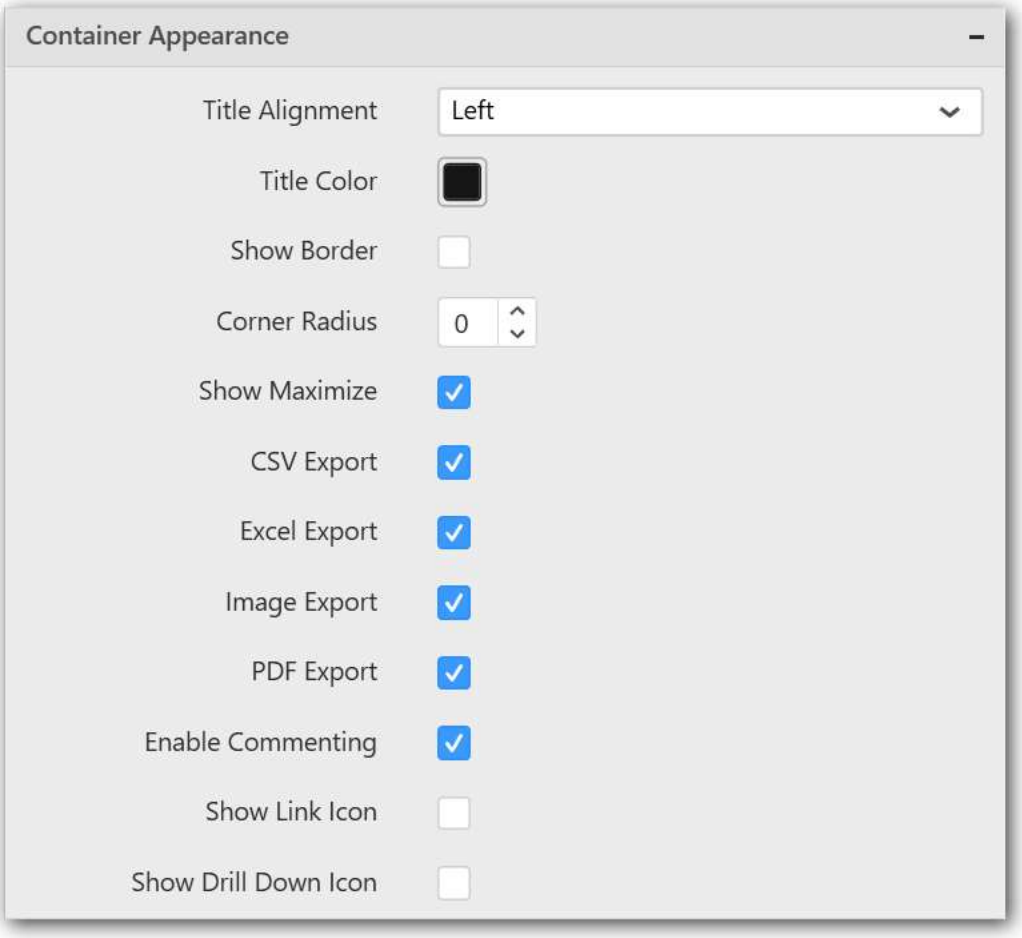
This allows you to define this line chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this line chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this line chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this line chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

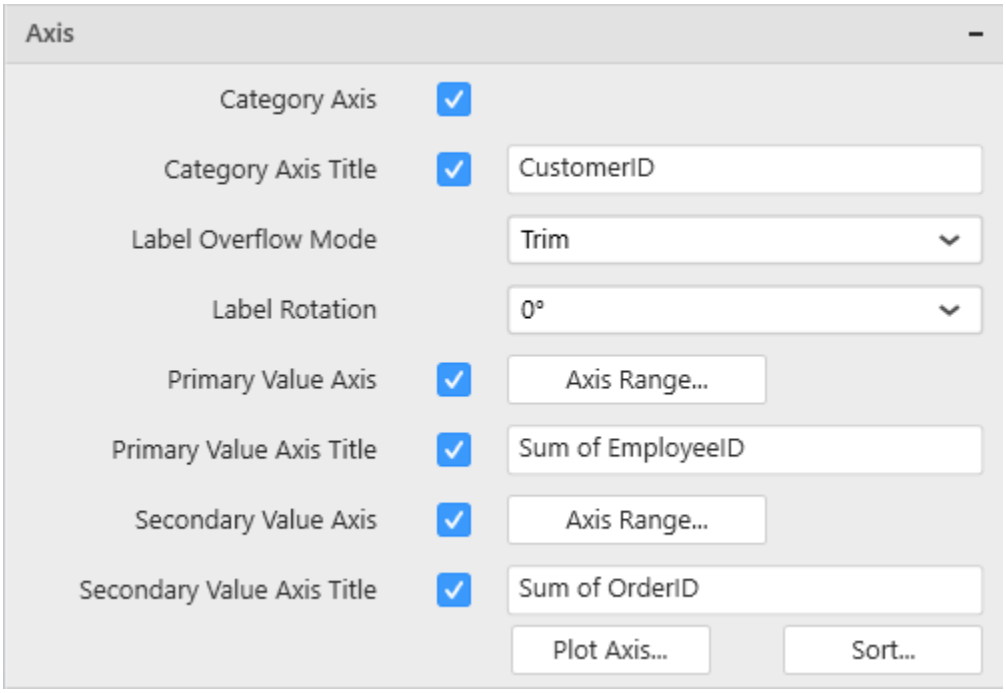
Image Export

This allows you to enable/disable the image export option for this line chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

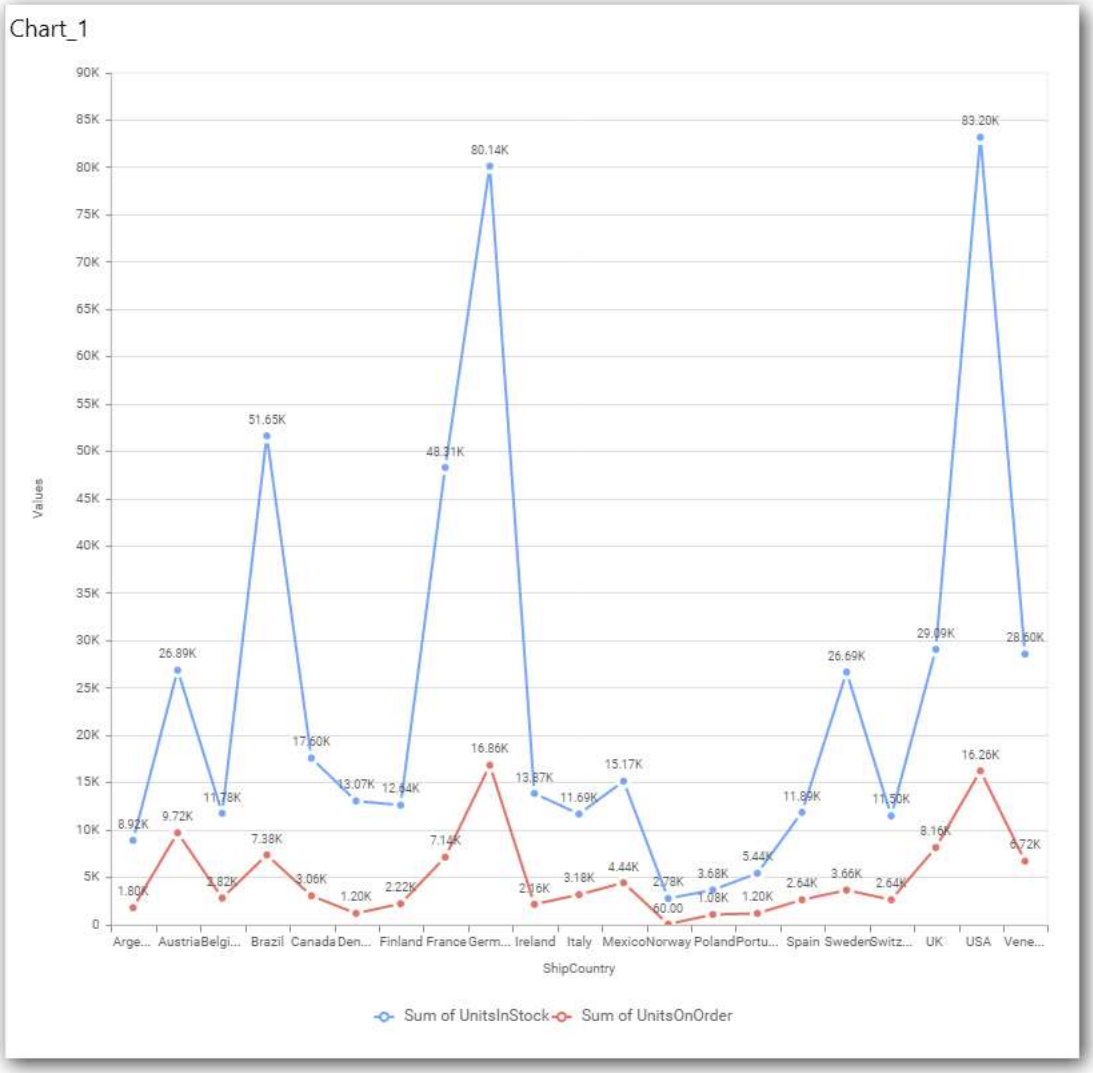
Axis Settings



This section allows you to customize the axis settings in chart.

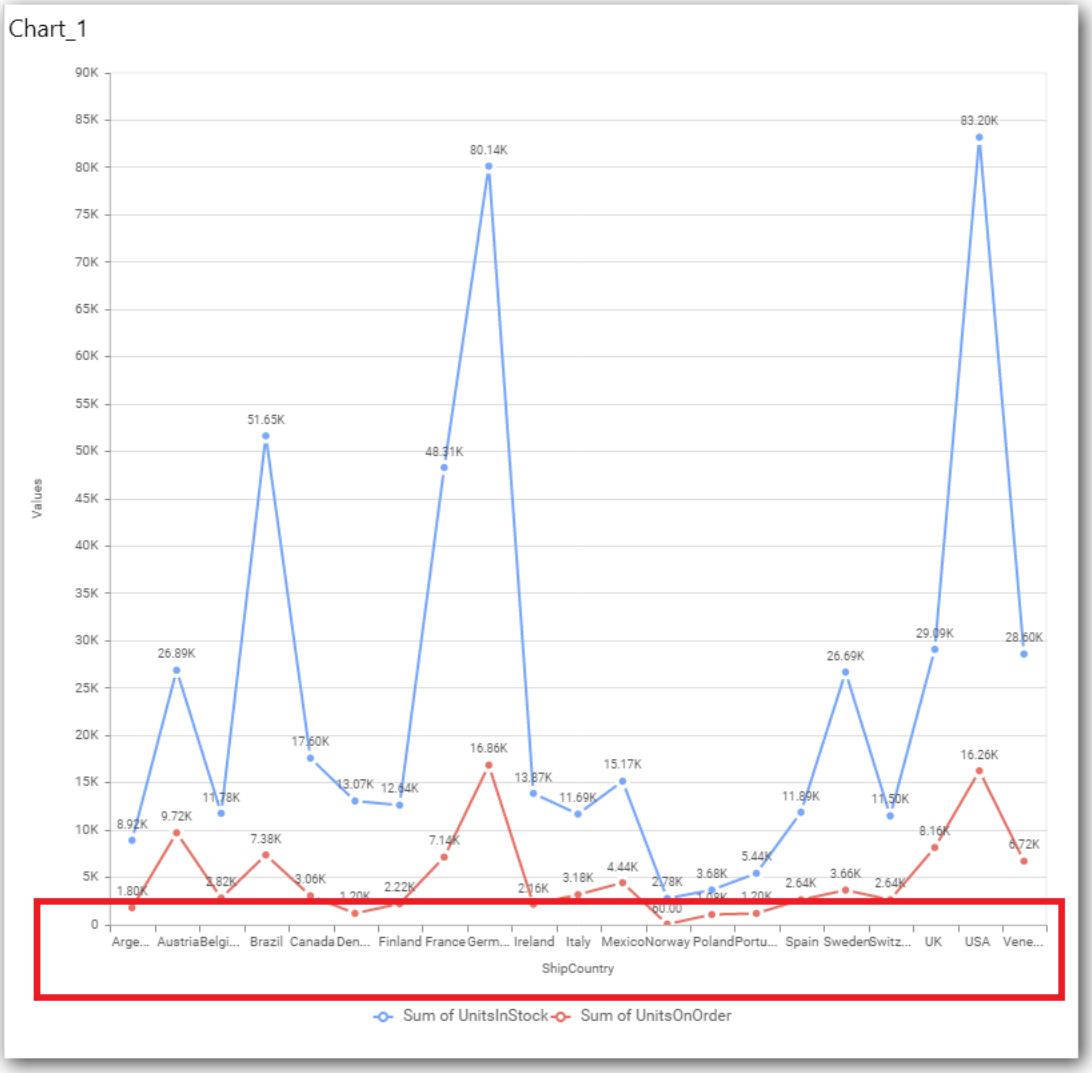
Category Axis

This allows to enable/edit the option of Category Axis. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

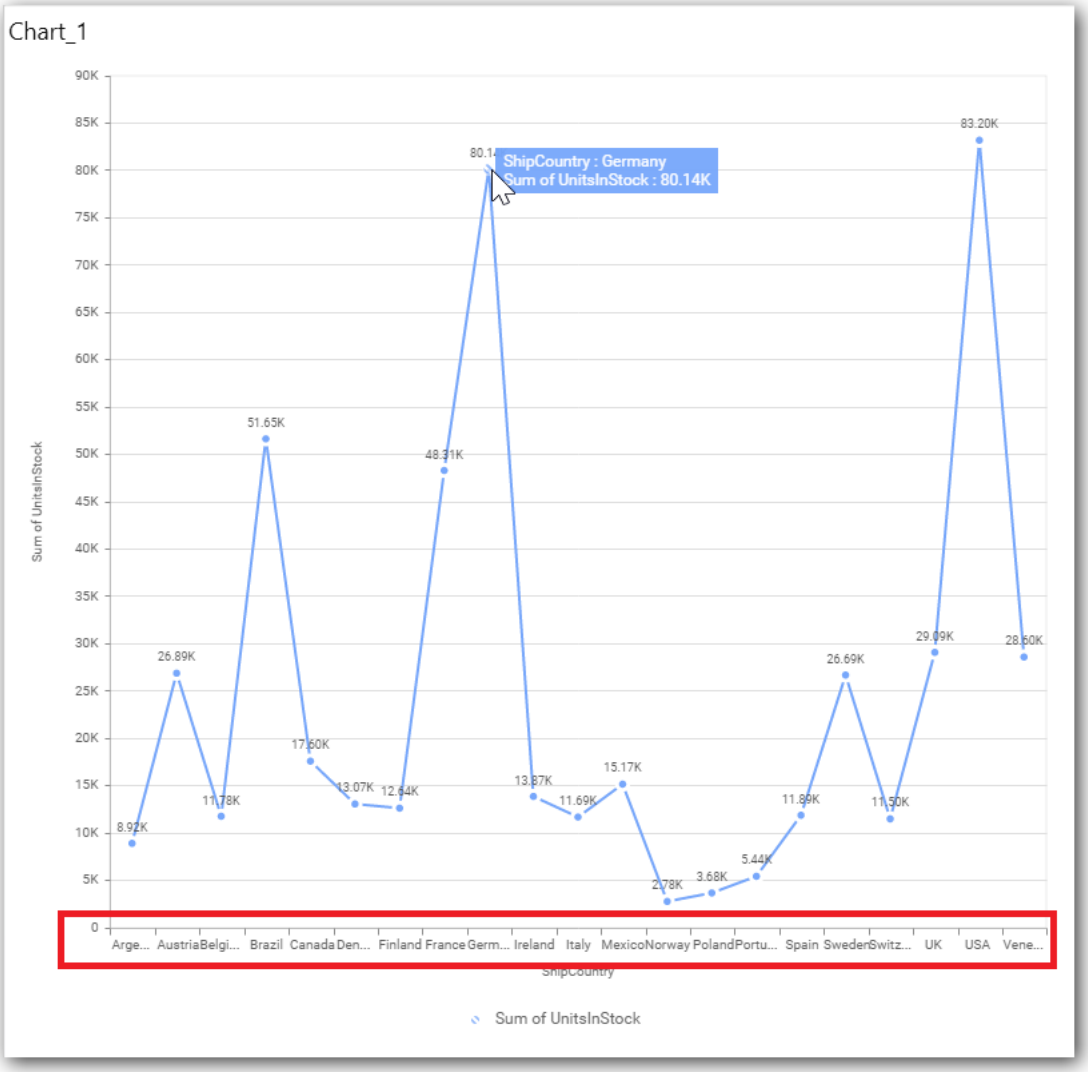


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

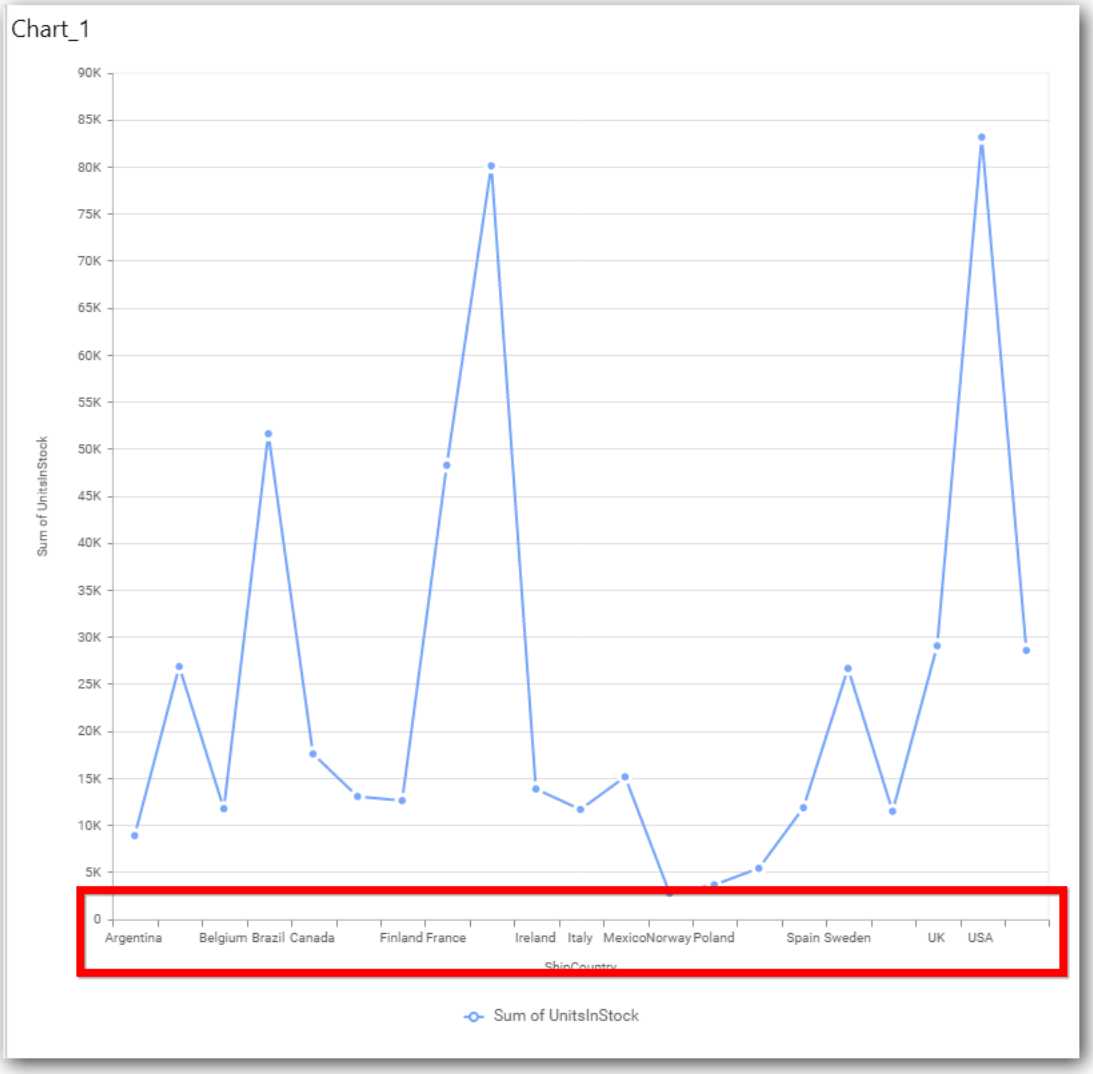
Trim

This option trims the end of overlapping label in the axis.



Hide

This option hides the overlapping label in the axis.

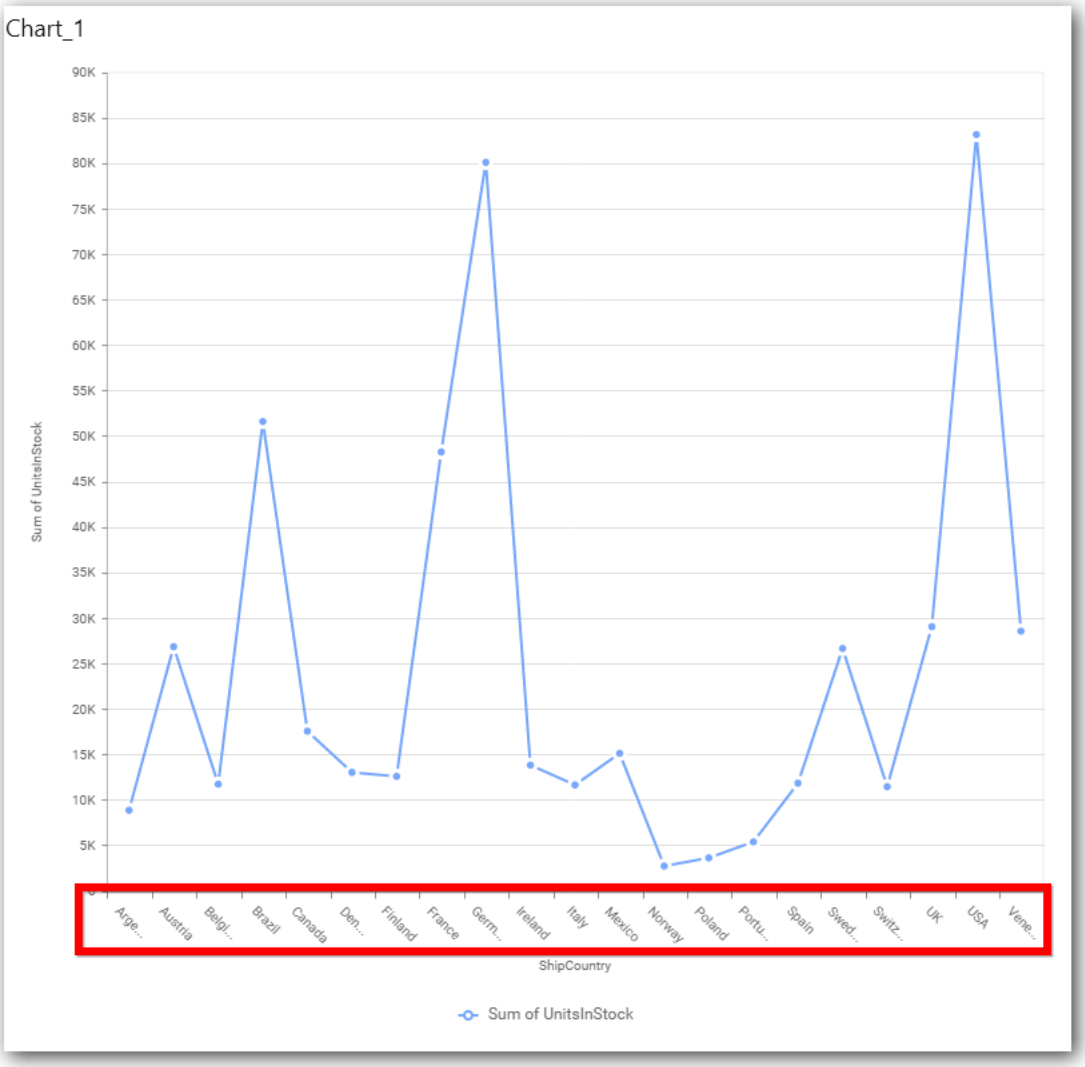


Wrap

This option wraps the lengthy label text in the axis.

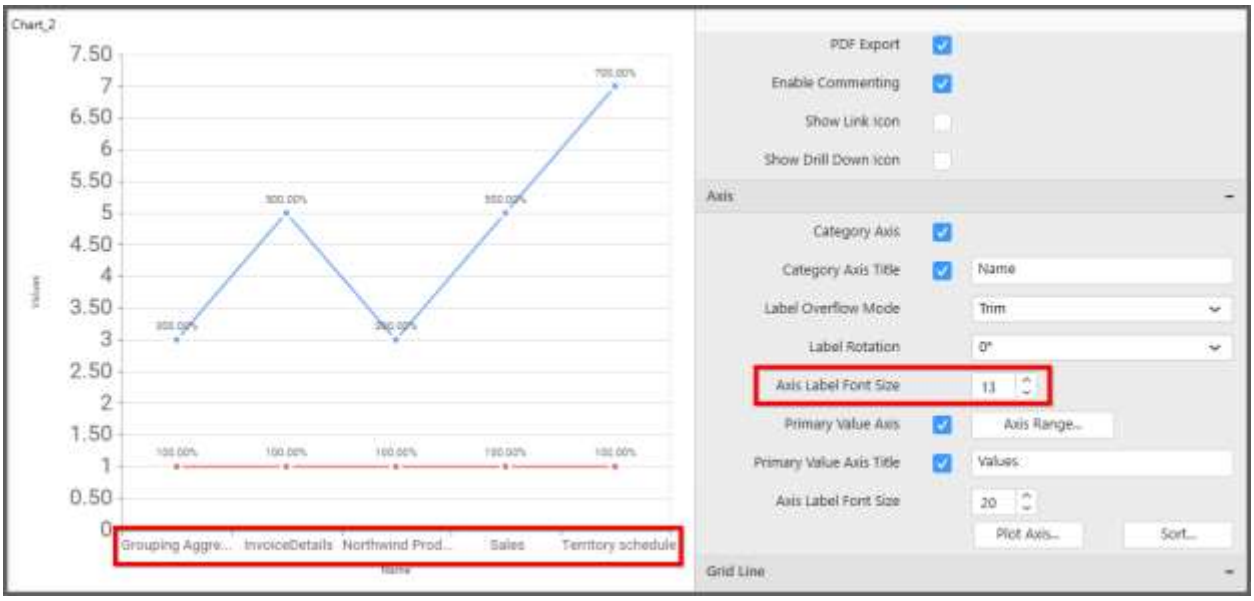
Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



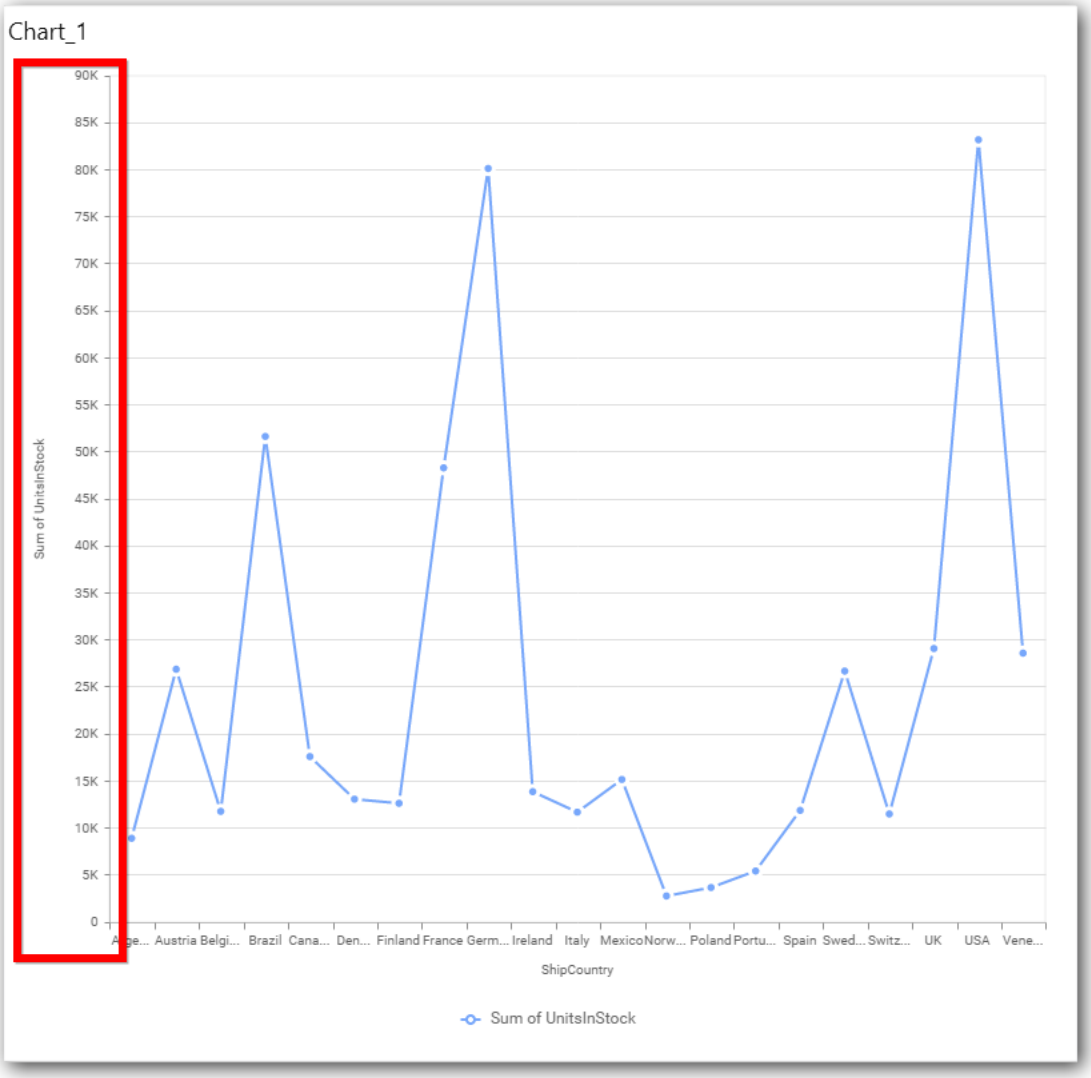
Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.



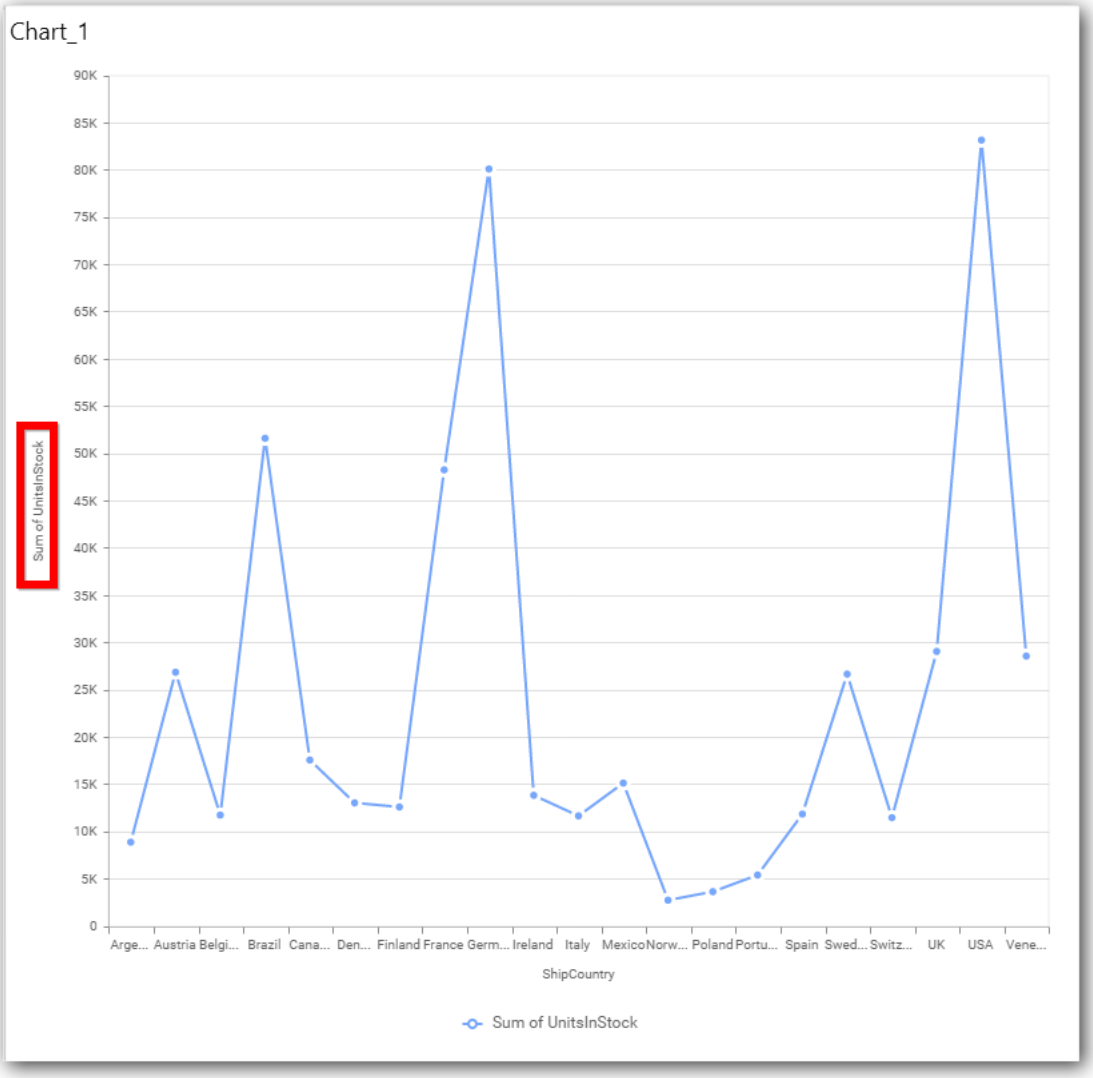
Primary Value Axis

This allows you to enable or edit the option of Primary Value Axis. It will reflect in chart area y-axis name.



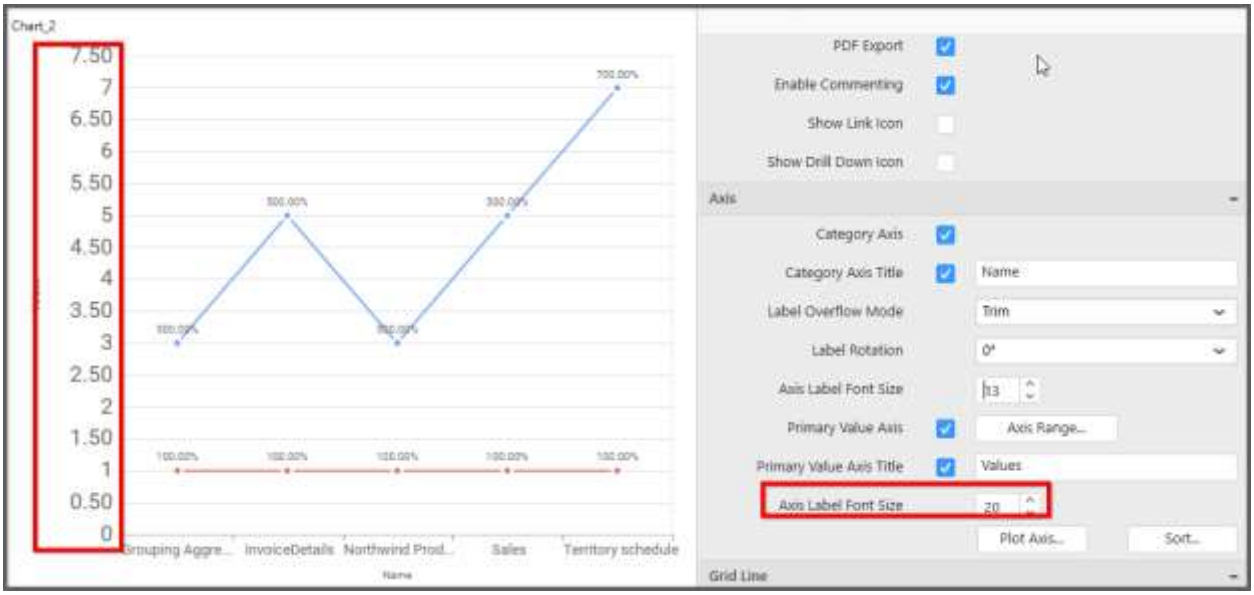
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

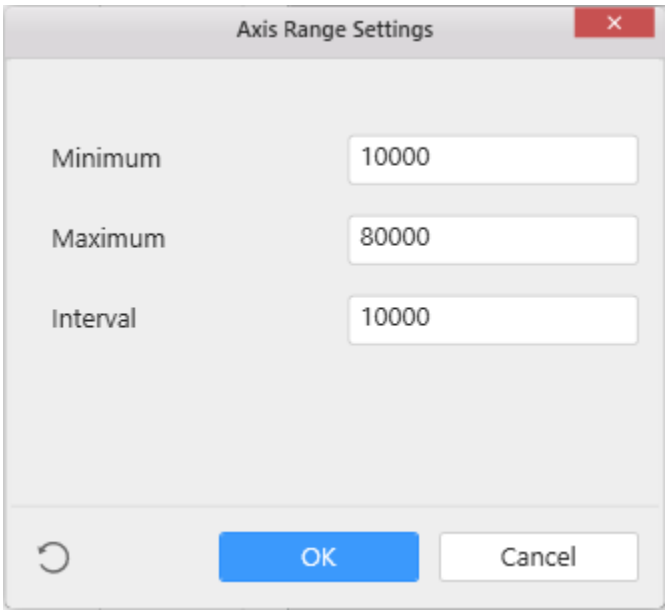


Primary Value Axis Range

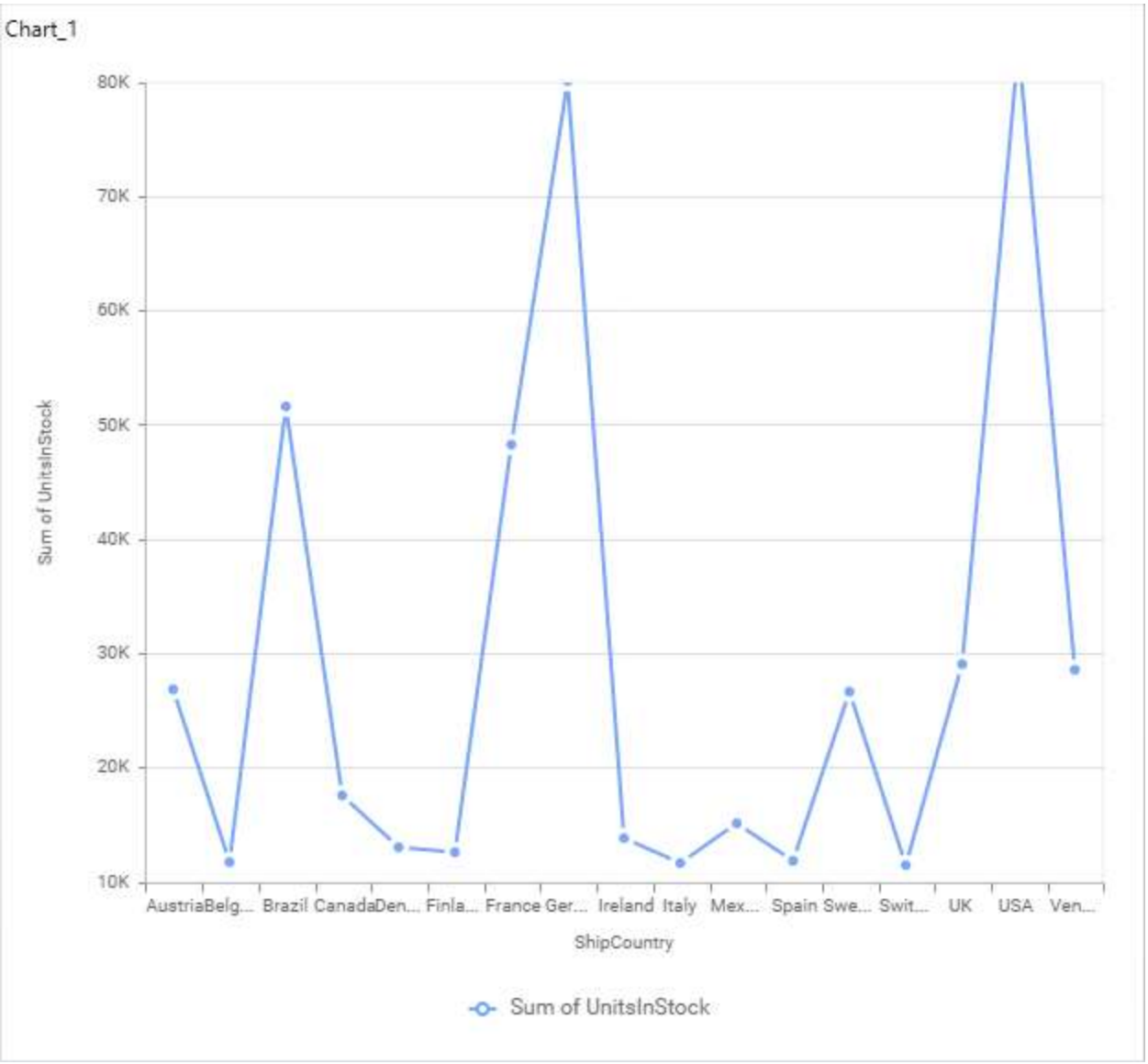
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.

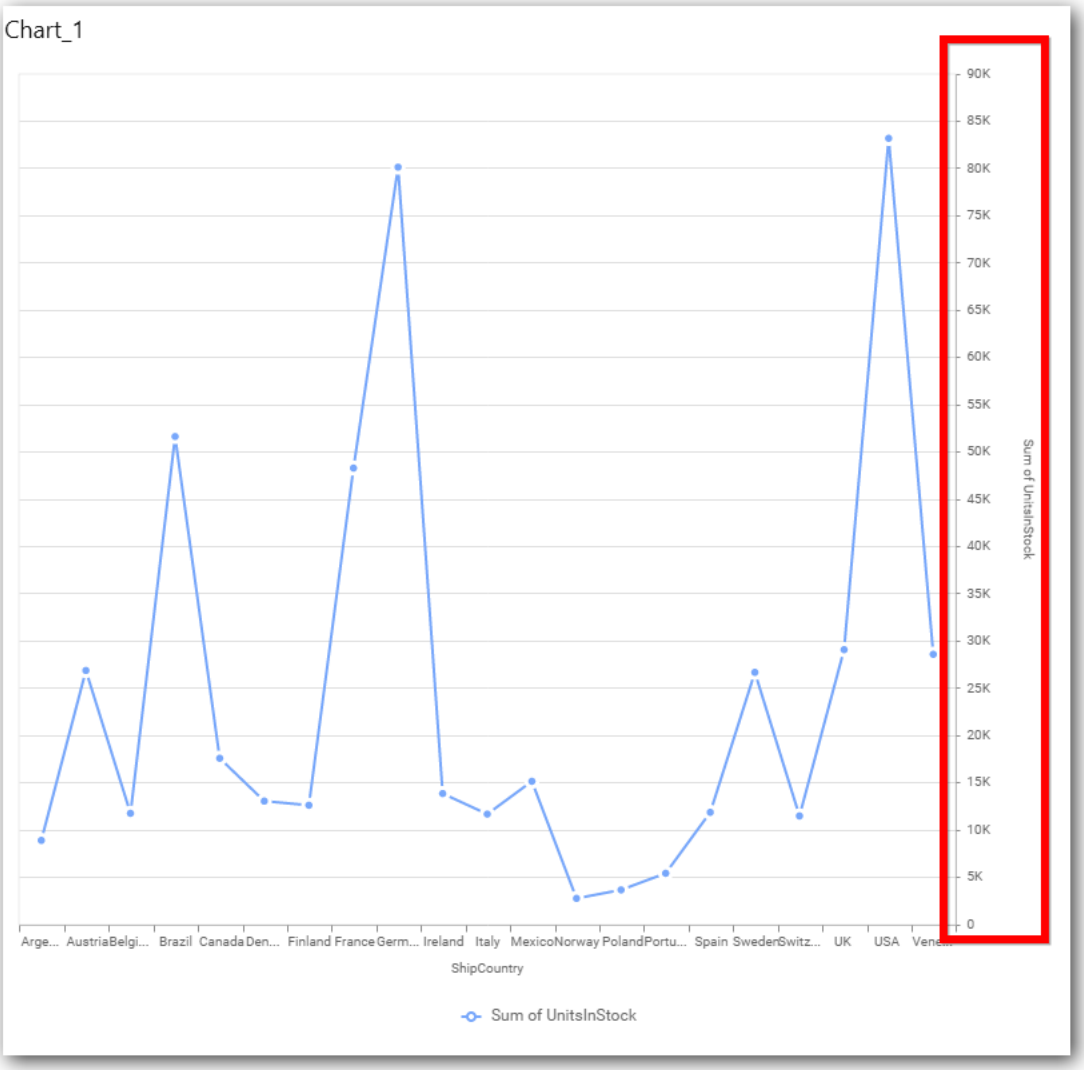


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



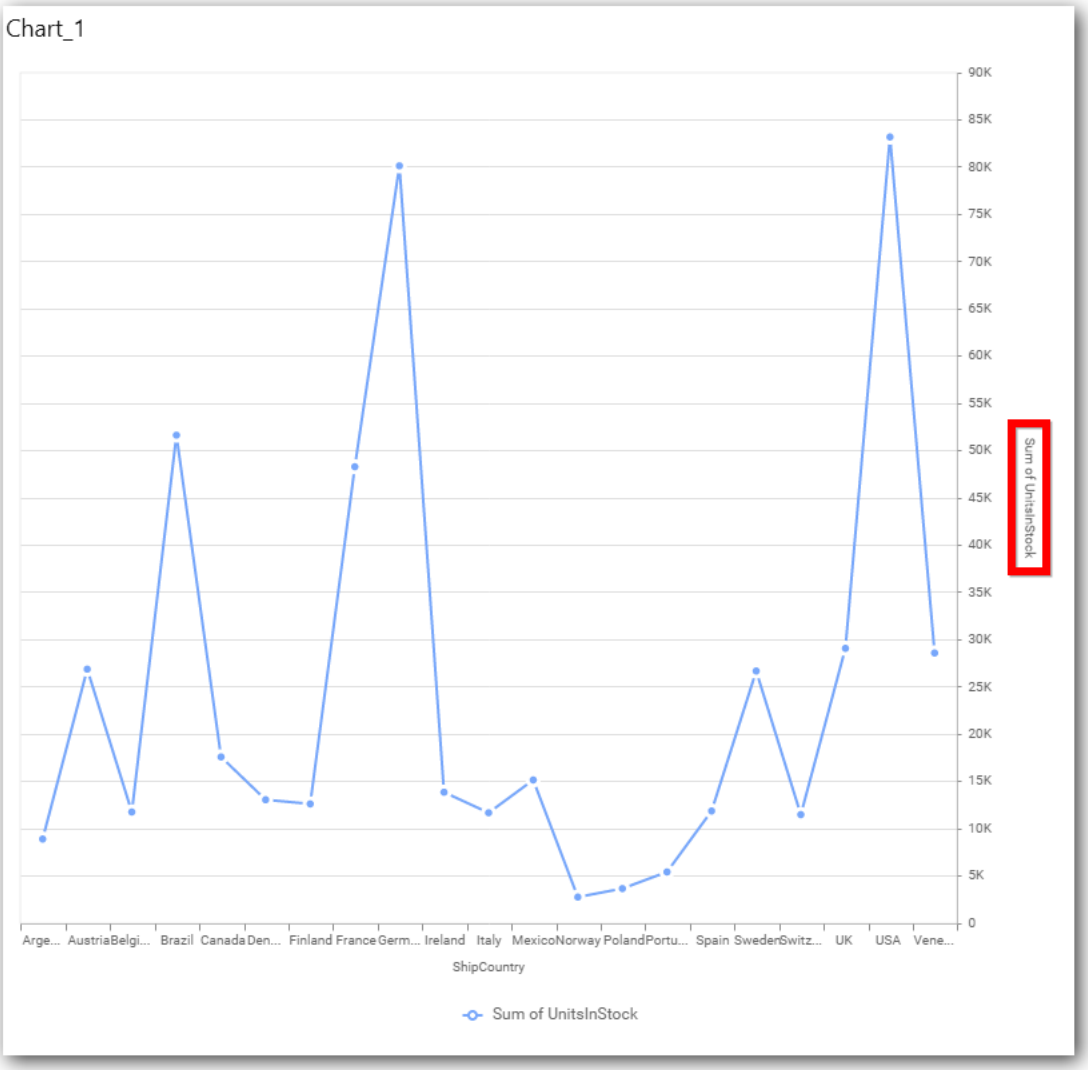
Secondary Value Axis

This allows to enable/edit the option of Secondary Value Axis. It will reflected in chart area secondary y-axis name.



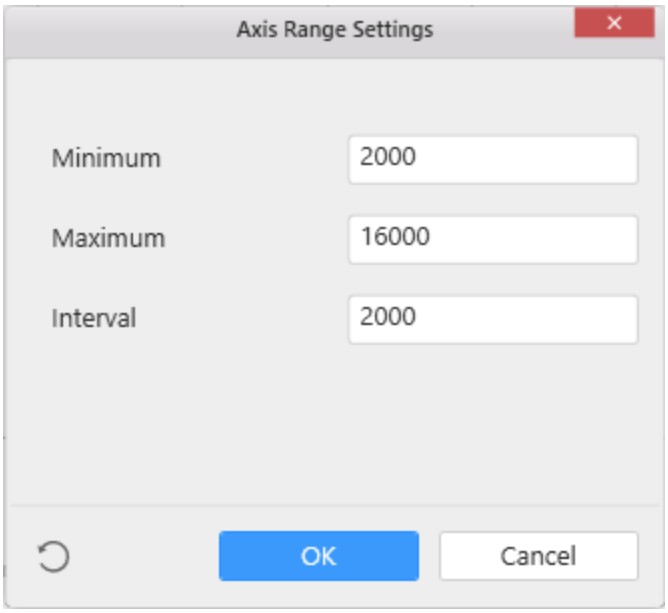
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

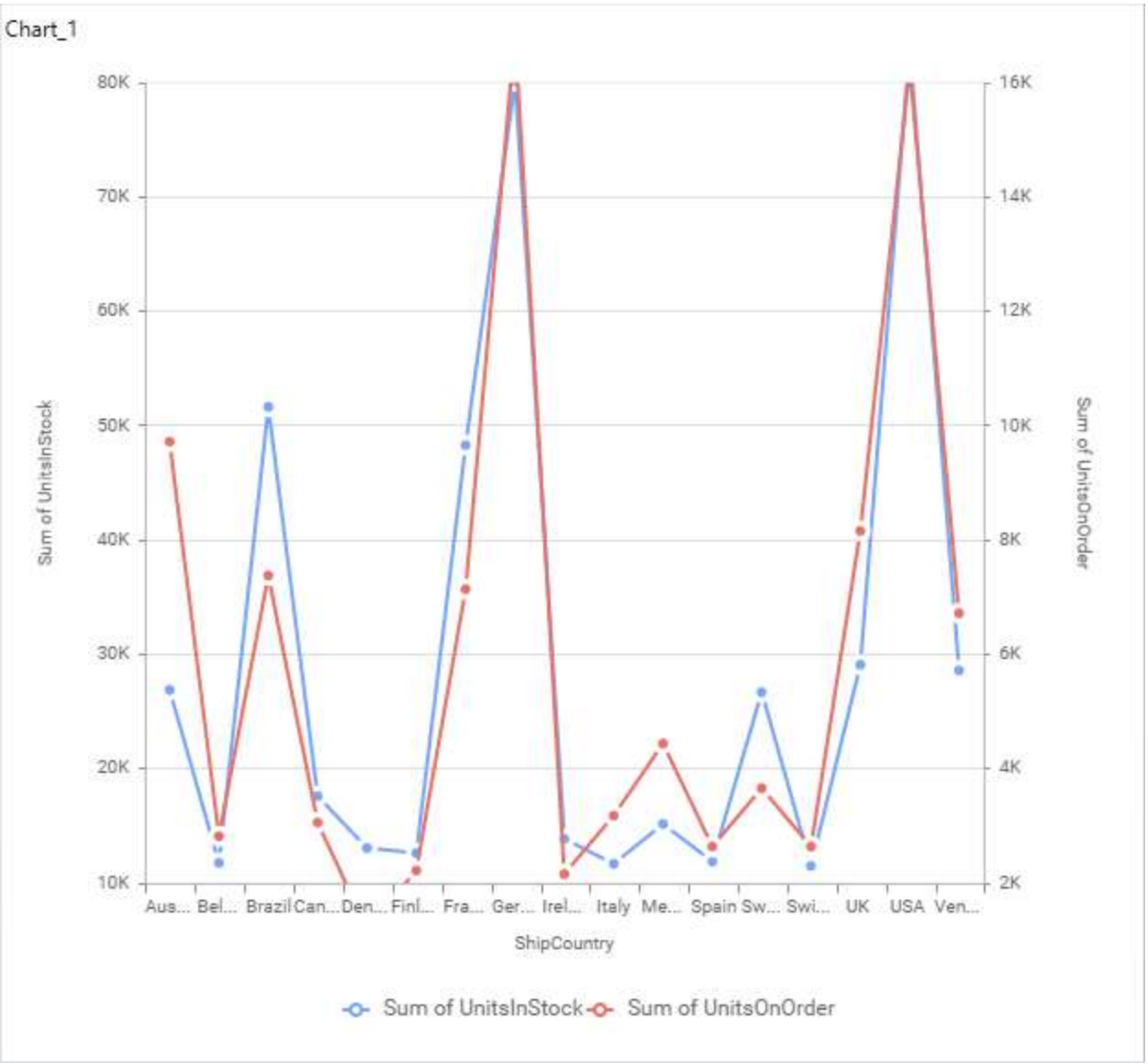


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

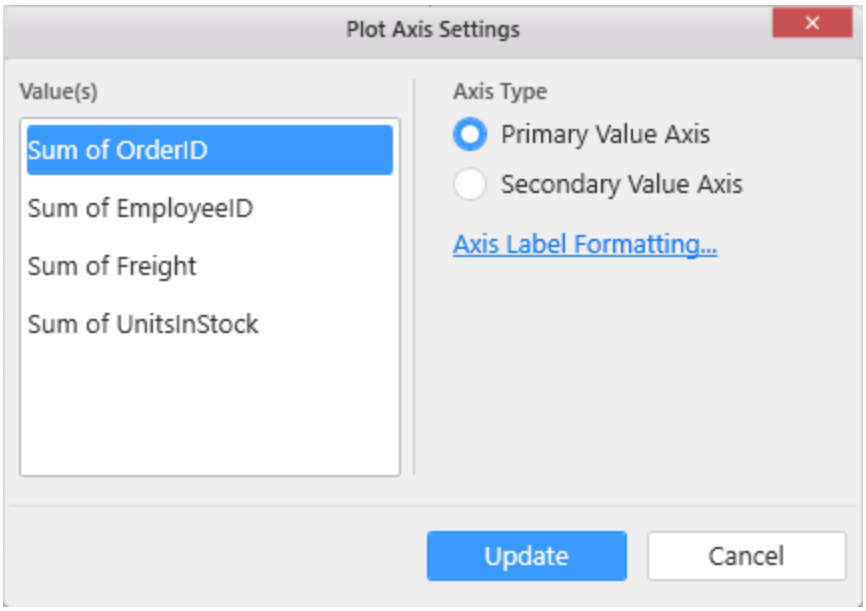


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

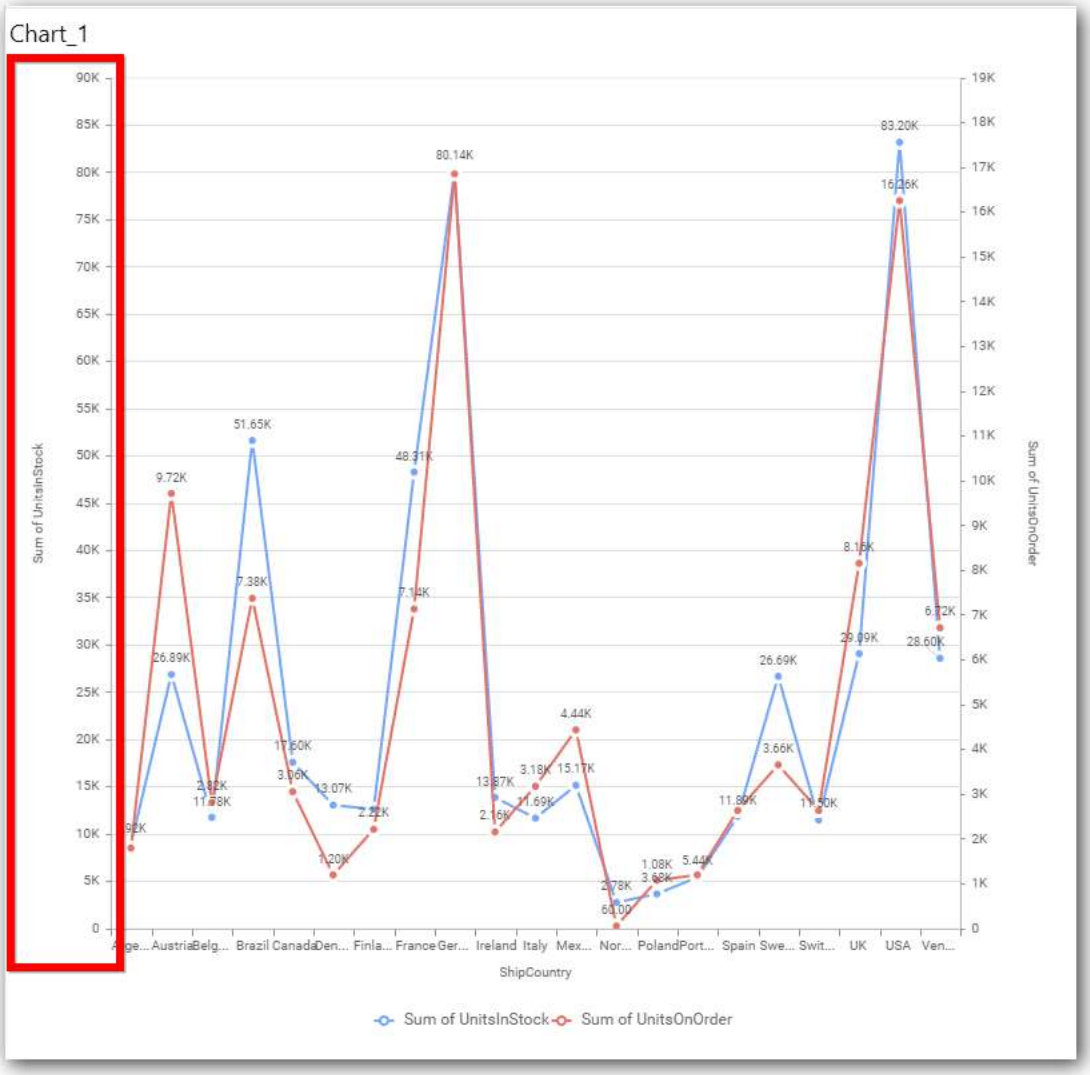


Plot Axis Settings

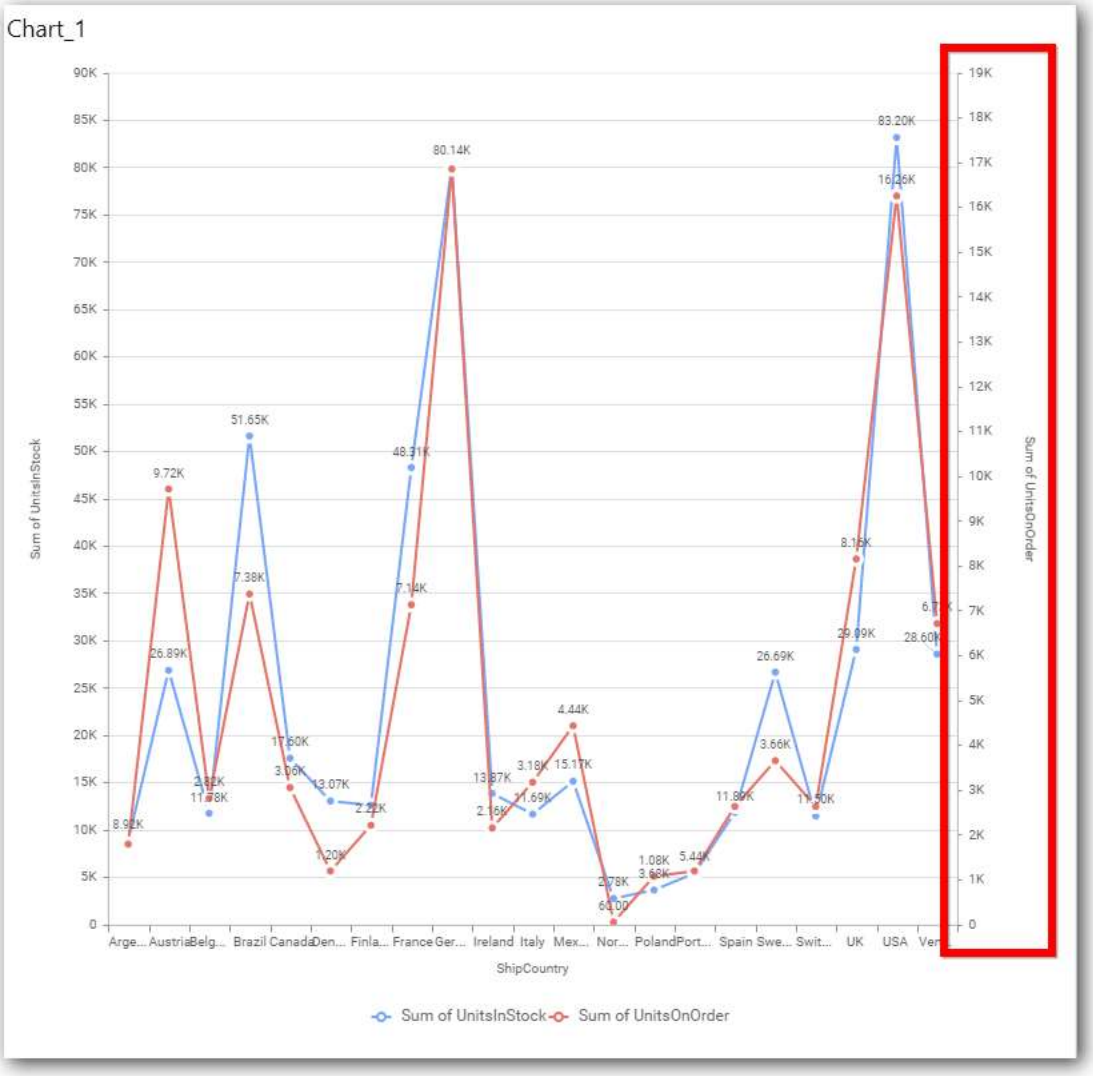
This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



Primary Value Axis

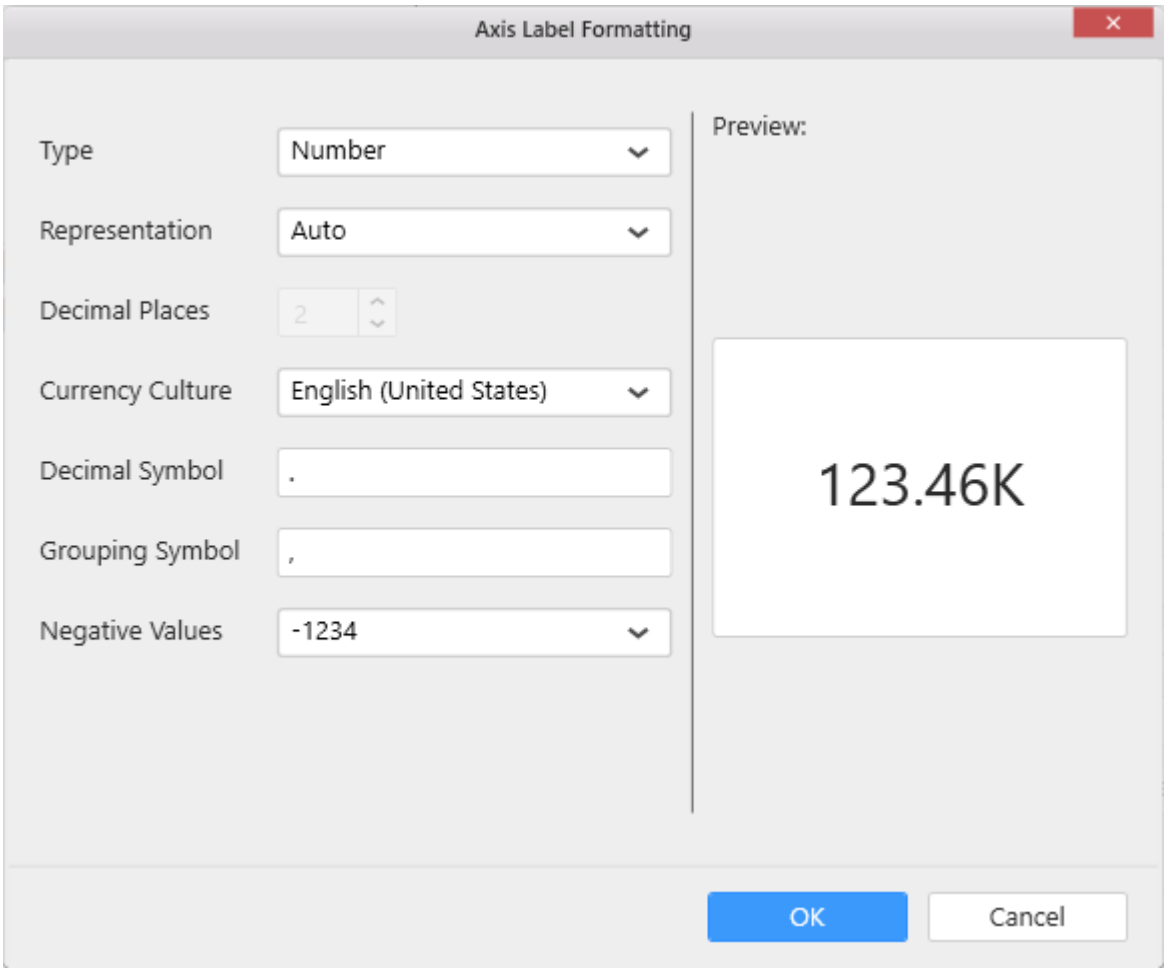


Secondary Value Axis



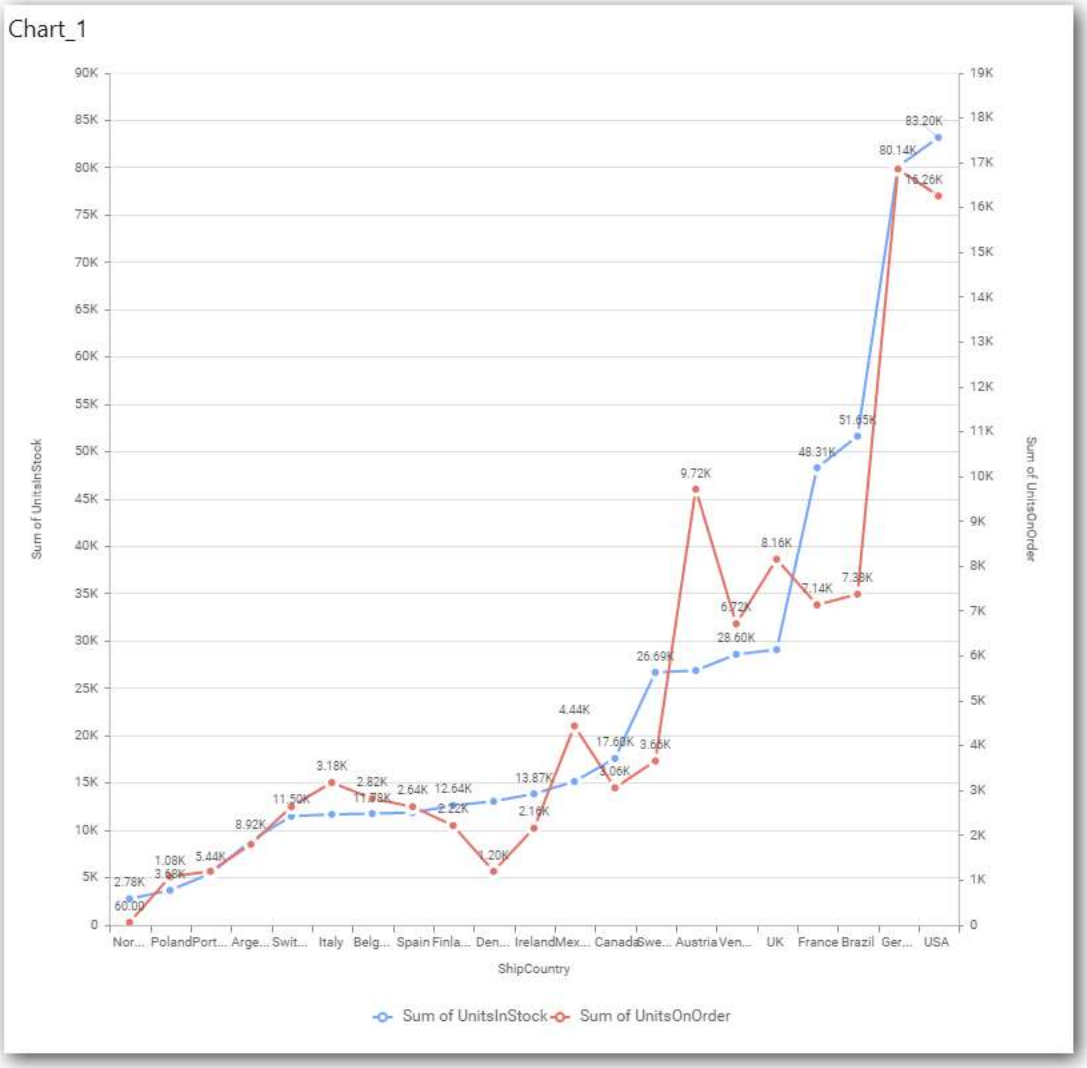
Axis Format Settings

This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.

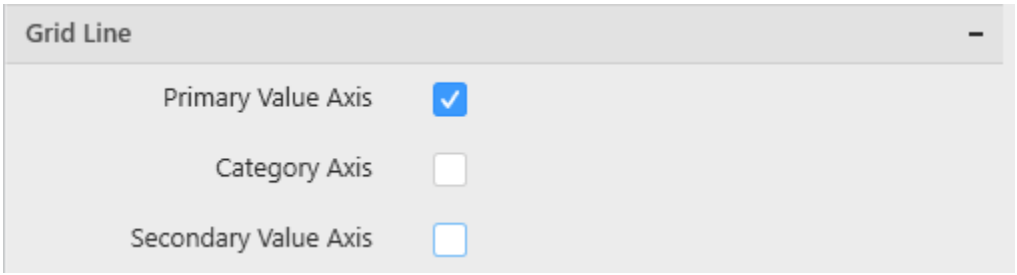


Sort Order

This allows you to define the sorting of chart based on any of the measures that you dropped and its order through this option. Following screenshot illustrates the Ascending sort order.



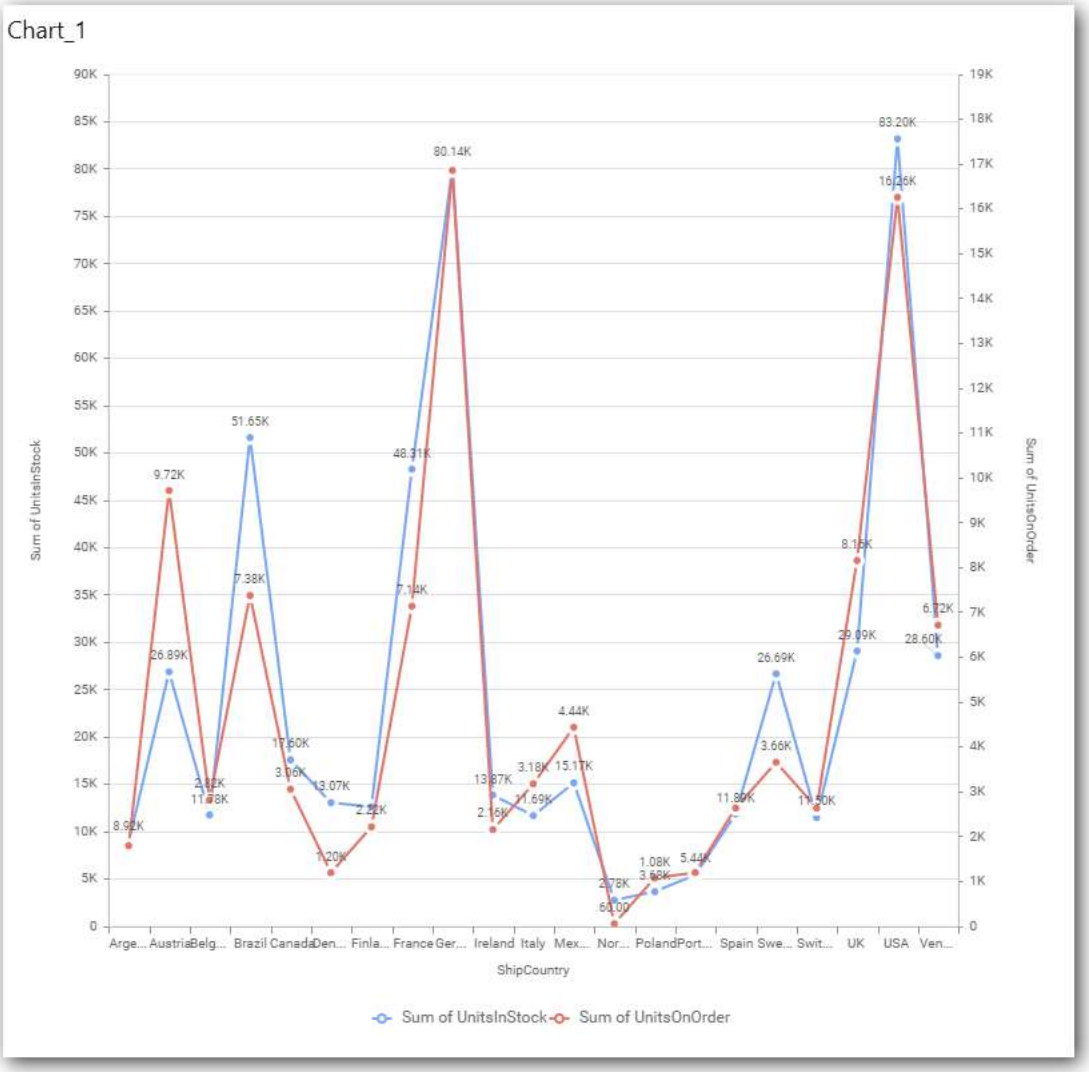
Grid Line Settings



To define the visibility of Category axis lines, Primary Value axis through the respective options. If Secondary Value Axis was enabled, you can define visibility of that line too.

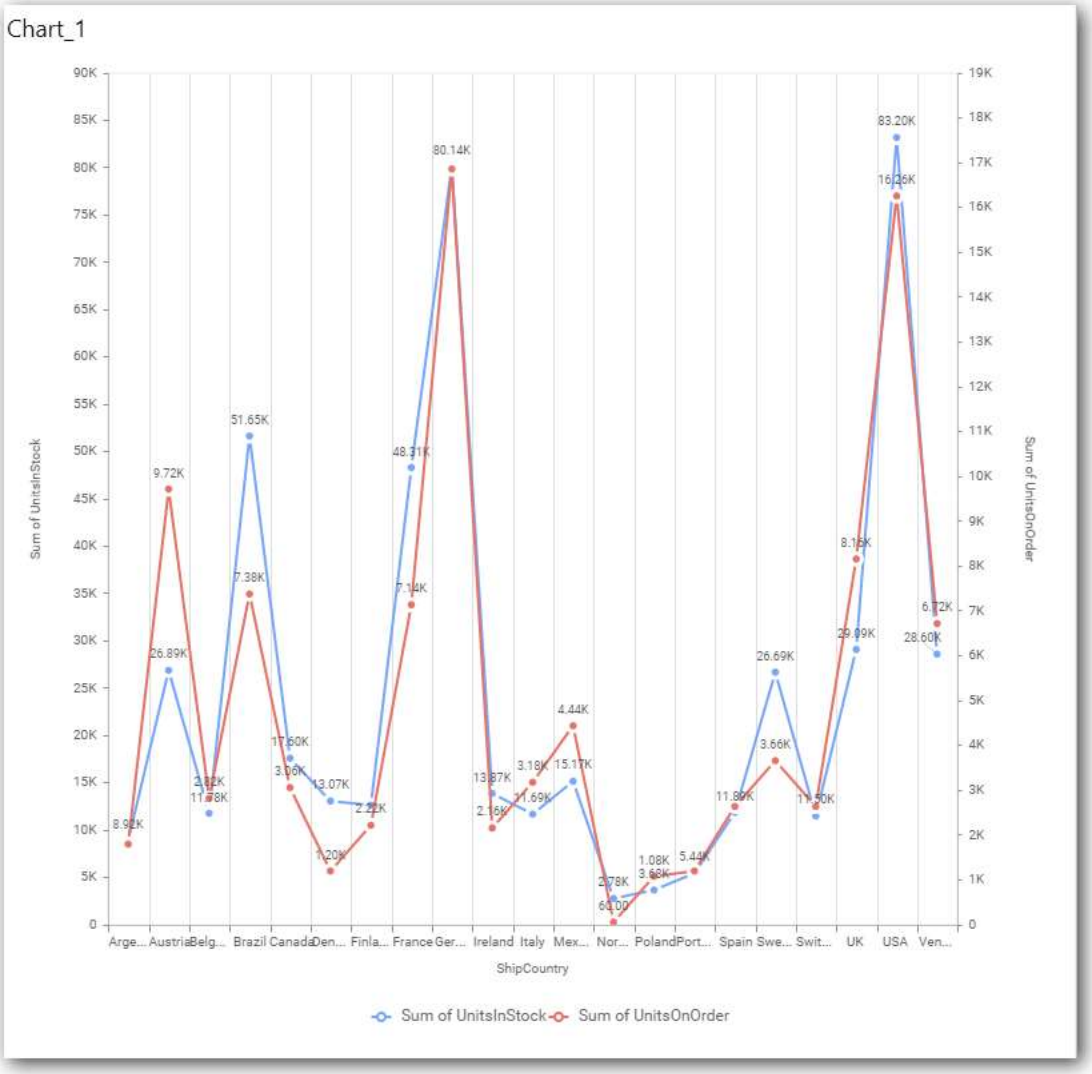
Primary Value Axis

This allows you to toggle the visibility of Primary Value Axis gridlines.



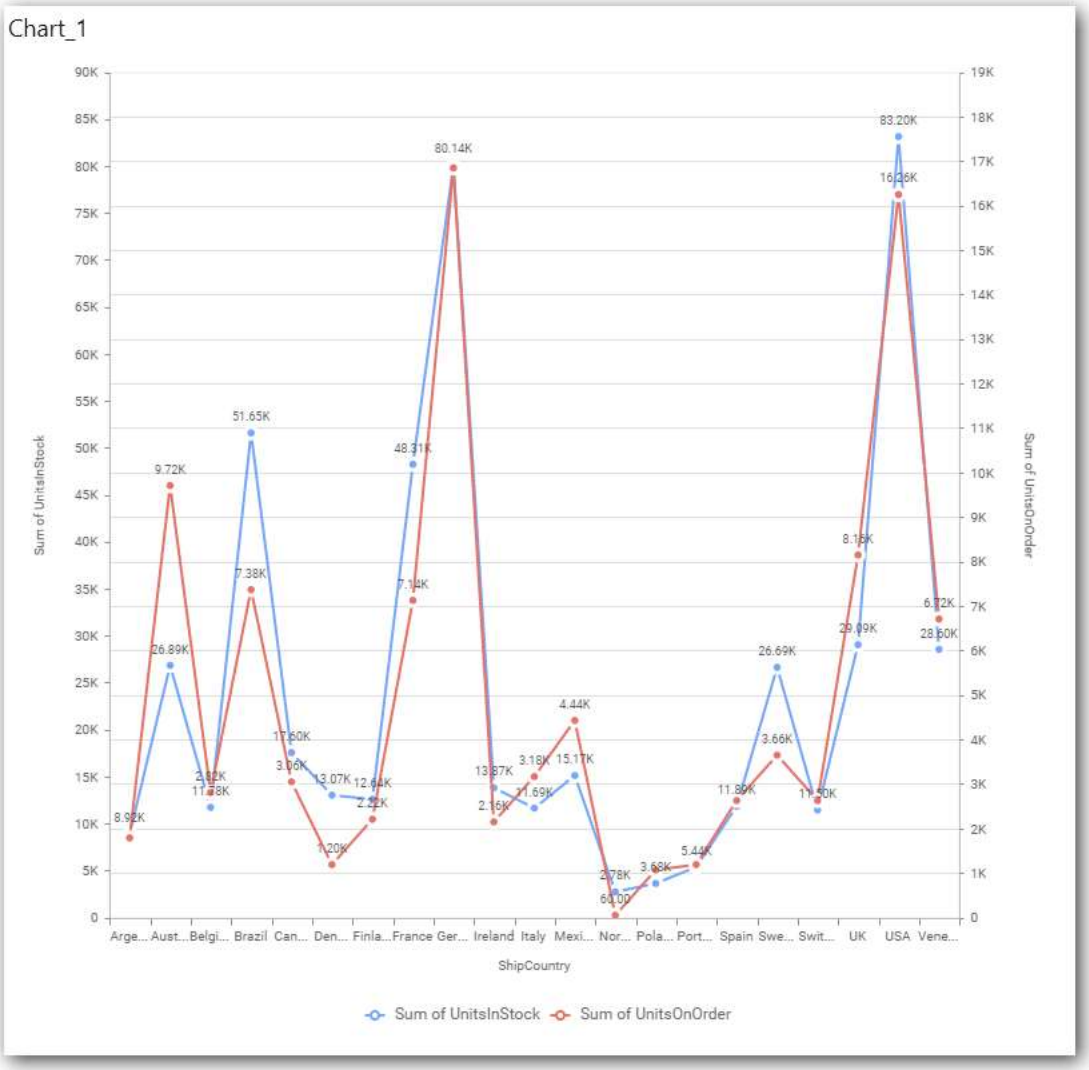
Category Axis

This allows you to toggle the visibility of Category Axis gridlines.

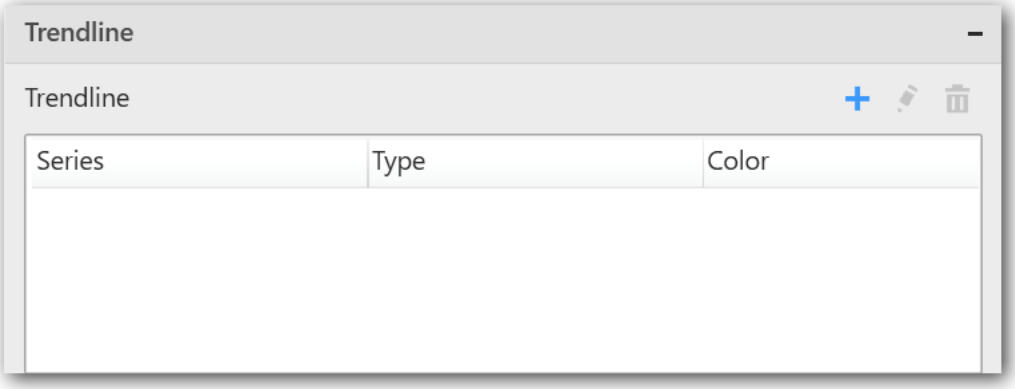


Secondary Value Axis

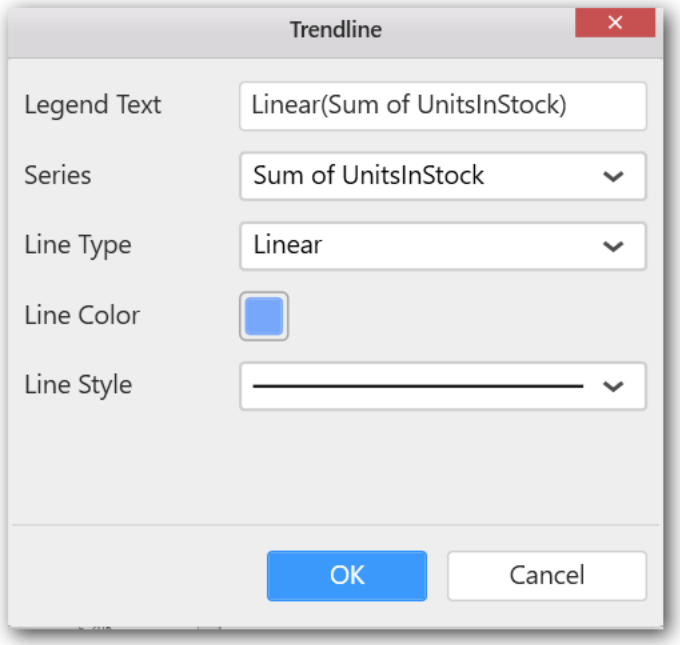
This allows you to toggle the visibility of Secondary Value Axis gridlines.



Trend Line Settings



You can add trend line to chart based on dropped measure that you select. You can also customize its legend text, line type and line color. Trend line is not visible, by default.



After applying these settings it will reflect in chart like below.

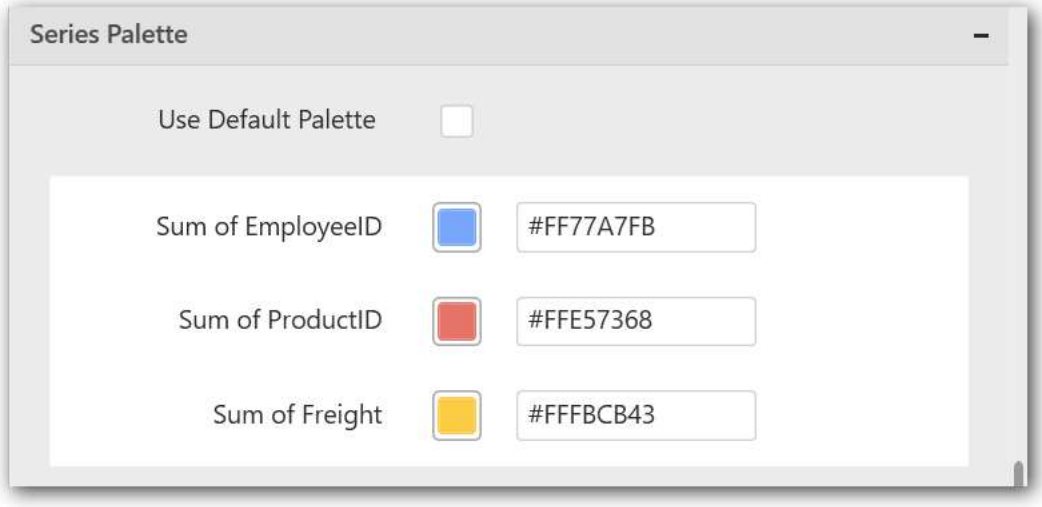
You have options to edit or delete the added trend lines.

Series Palette

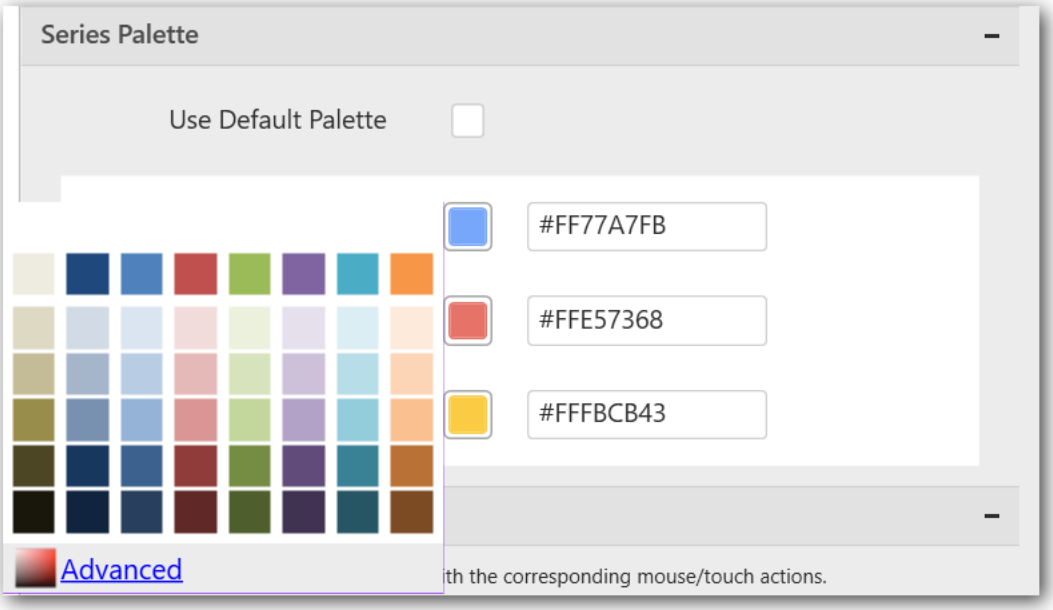
This allows you to customize the chart series color through Series Palette section.

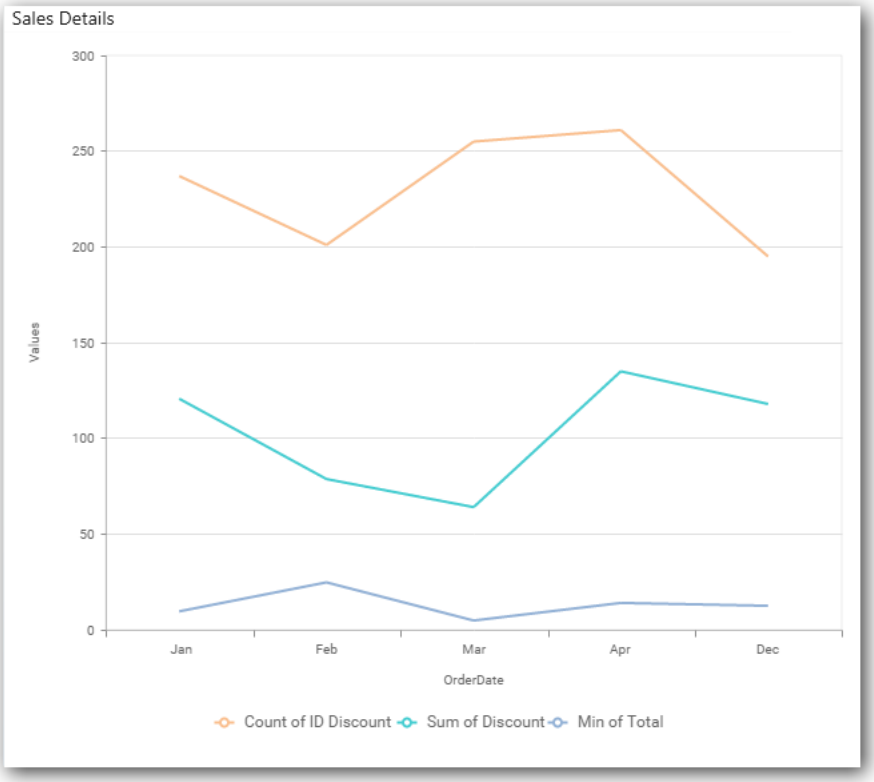
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Spline Chart

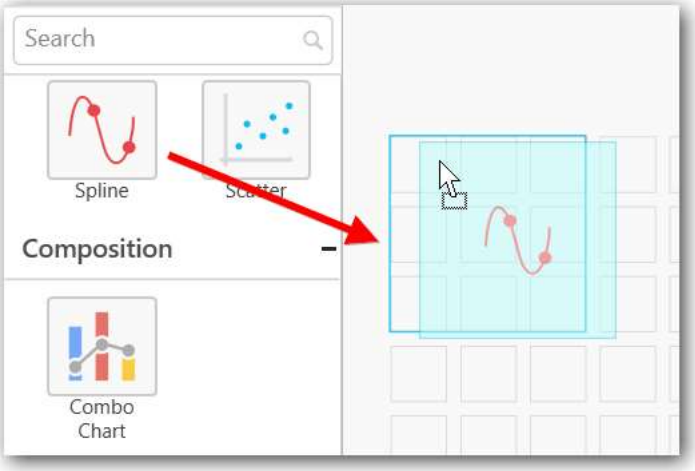
Spline Chart allows you to showcase trends for analysis over a time period with data points connected using splines.

How to configure flat table data to Spline Chart?

Spline Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

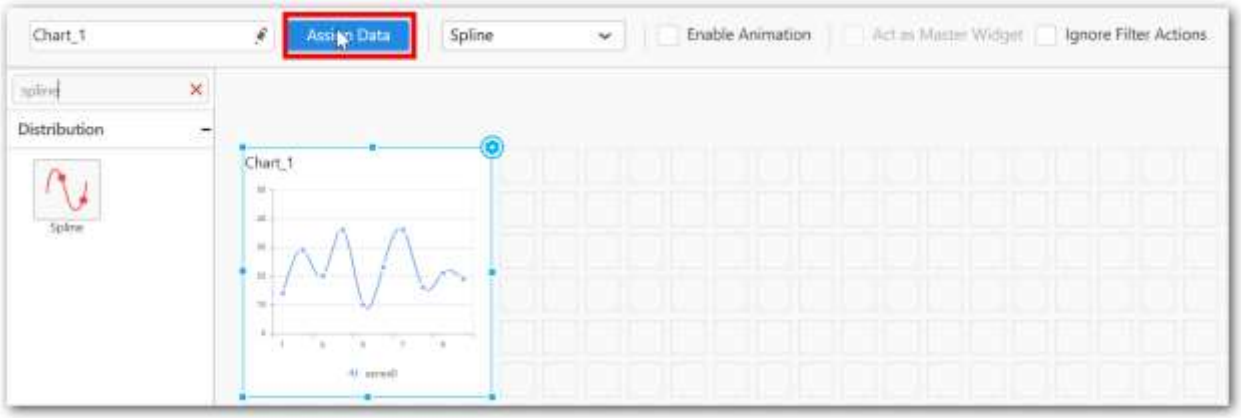
Follow the steps to configure data to spline charts

Drag and drop the spline chart into canvas and resize it into required size.

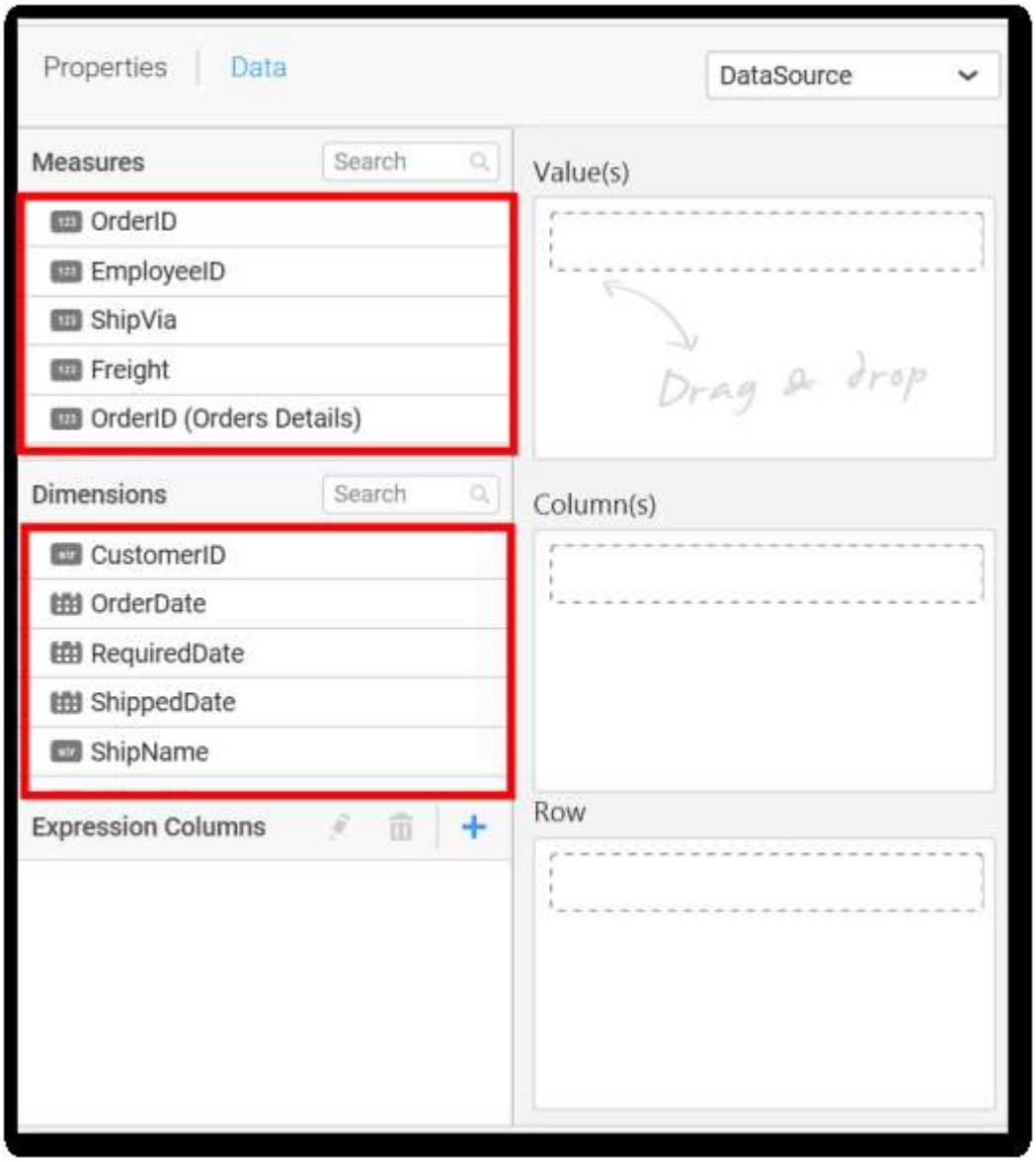


Connect to the data source.

Focus on the widget and click on Assign Data button.

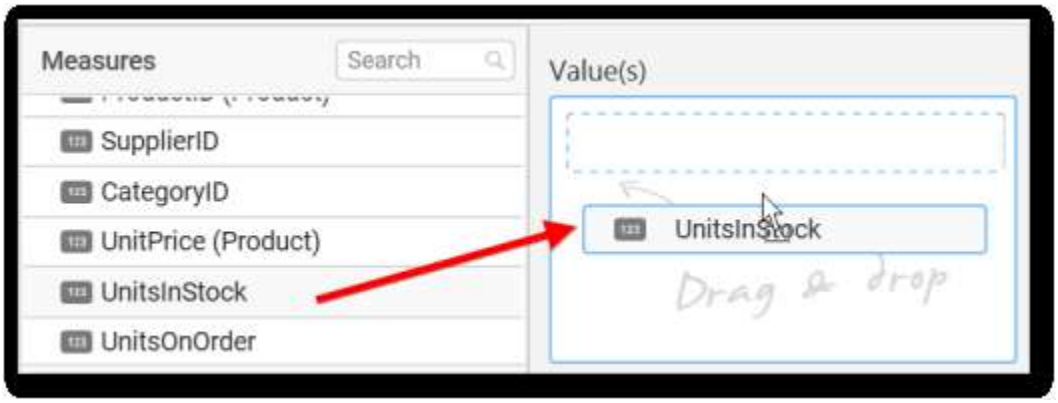


The data pane will be opened with available dimensions and measures from the connected data source

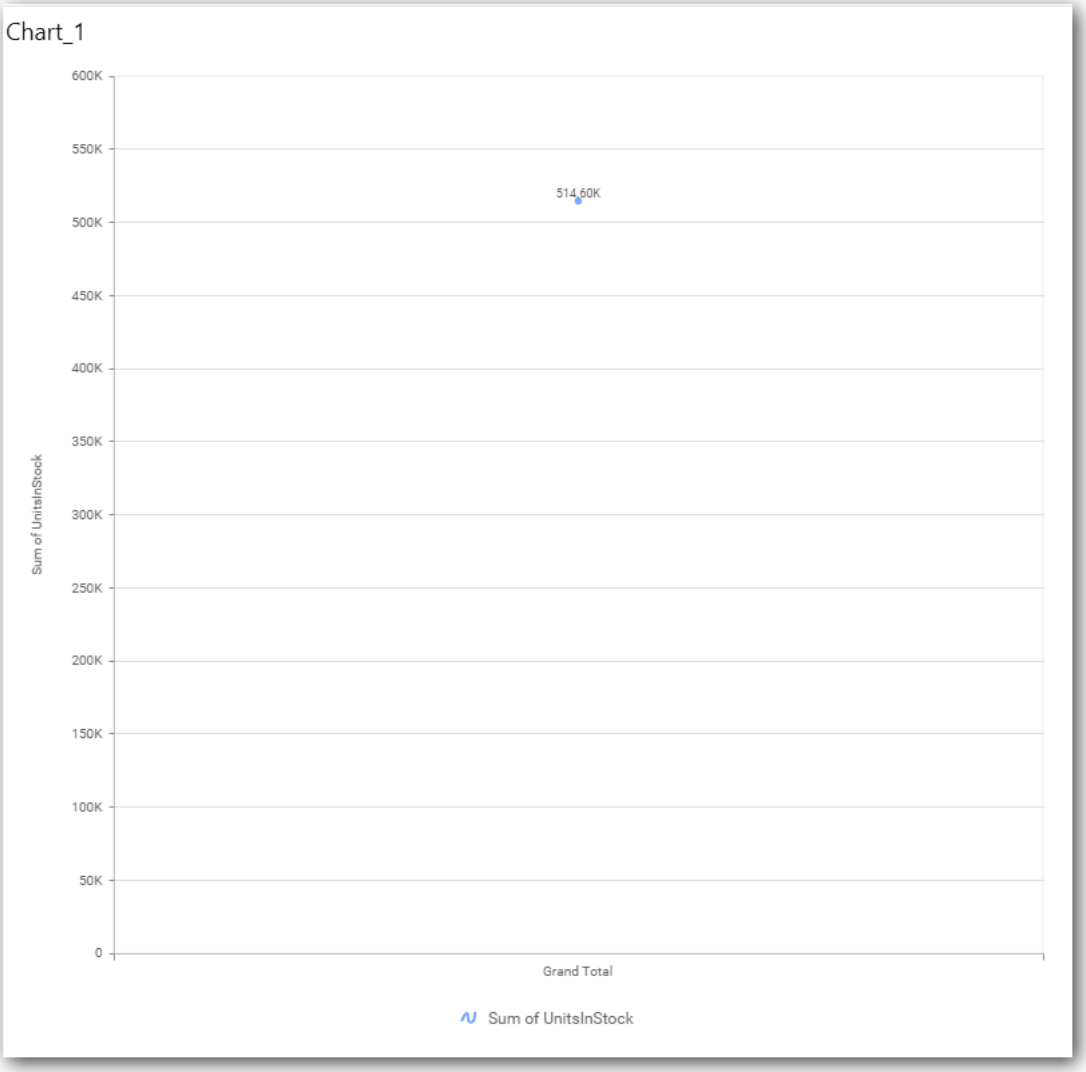


Assigning Value(s)

Drag and drop the Measure into Value.

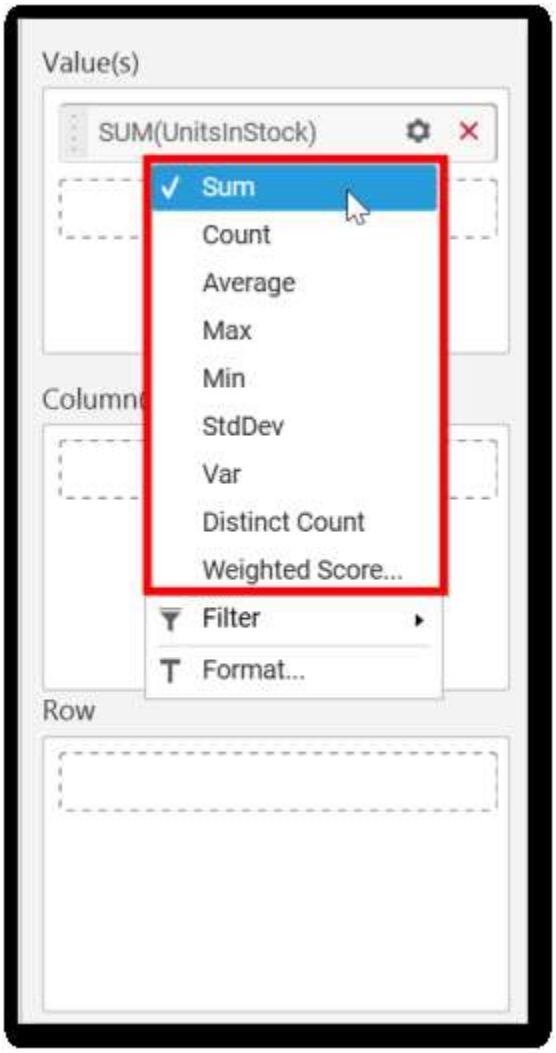


Now the chart will be rendered like this.

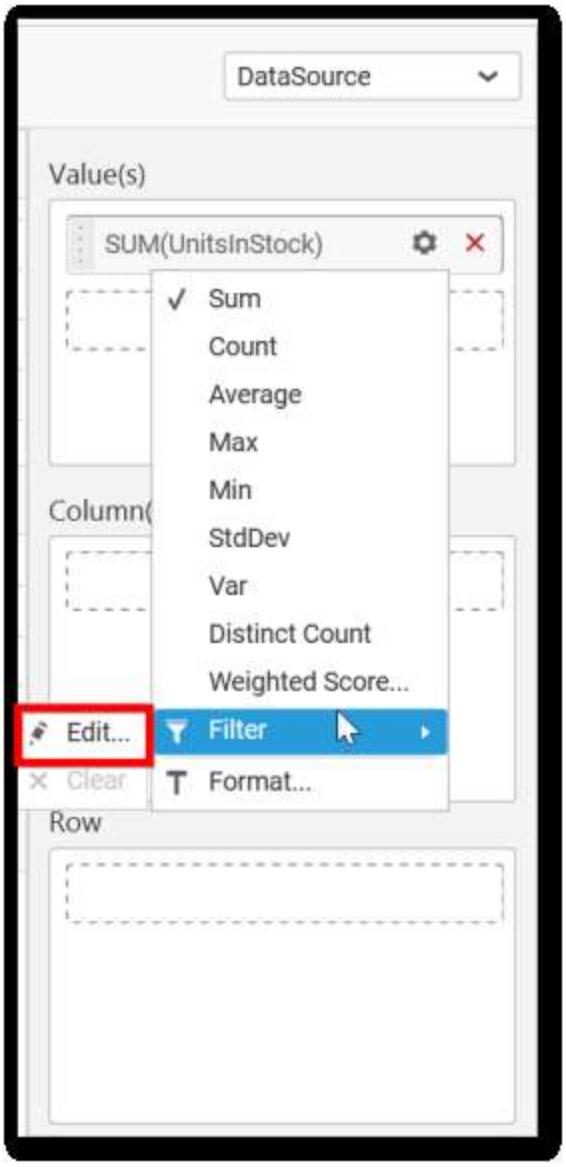


You can change the summary type of the value by clicking on Settings option.

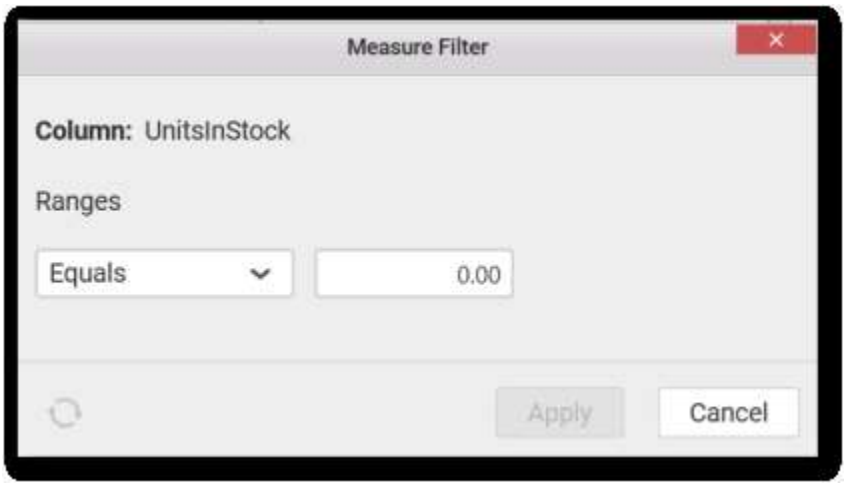
Select the required summary type from list.



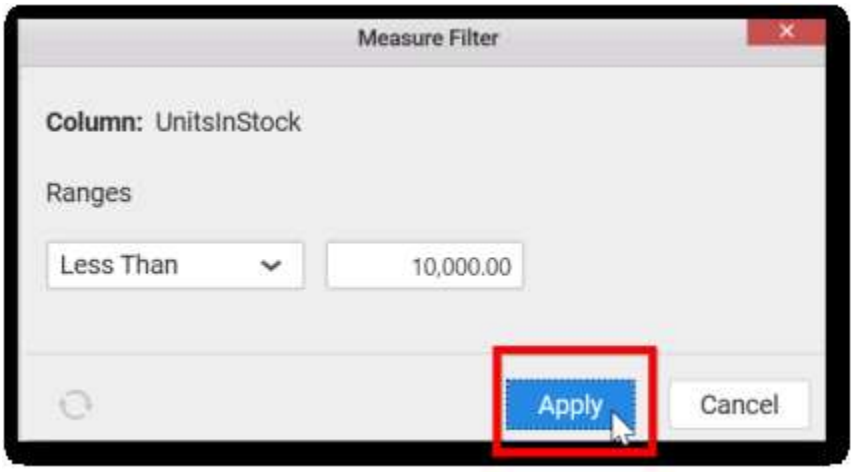
You can select what data to be displayed by choosing filter option.



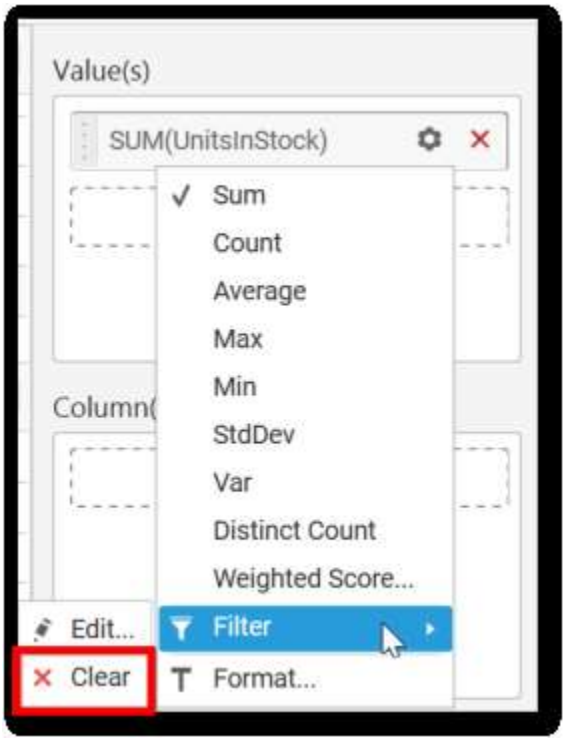
The Measure Filter option will be shown.



You can choose the filter condition and apply the condition value.

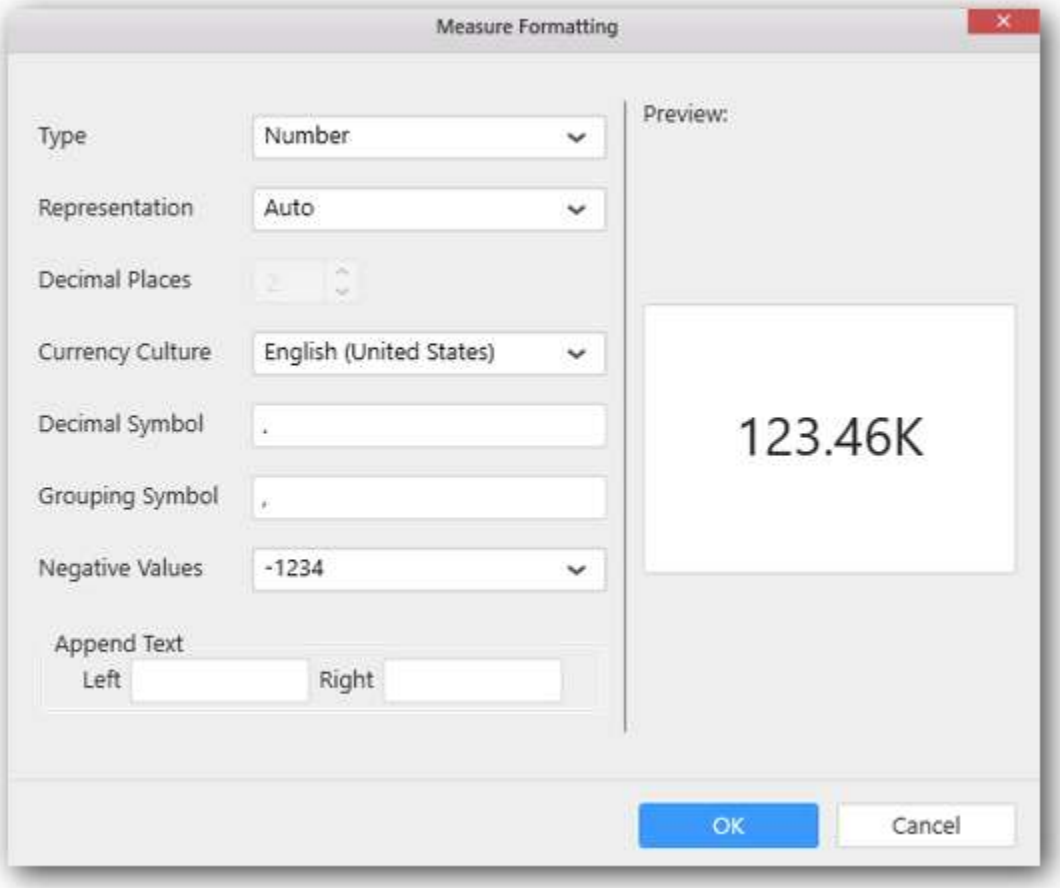


You can Clear the filter.

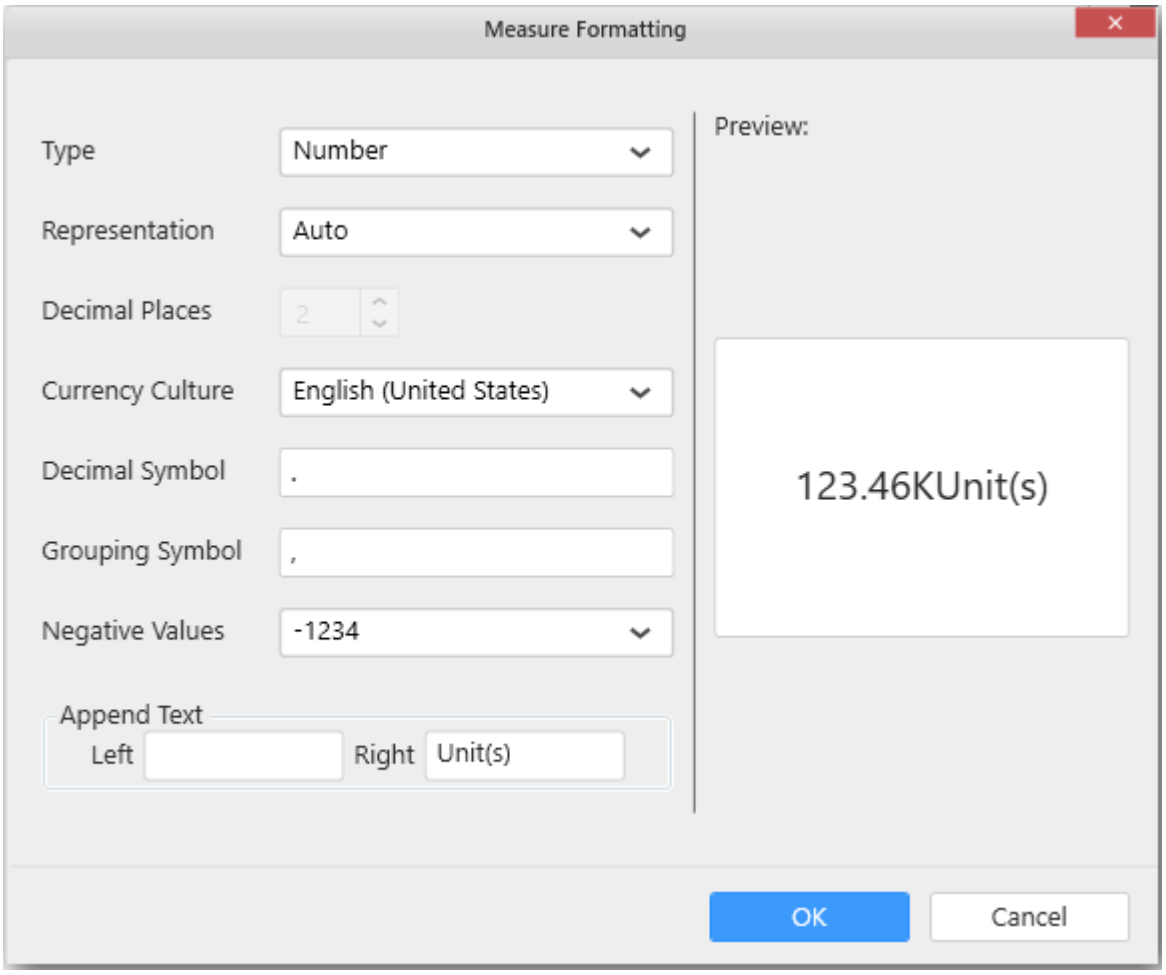


You can Format the value.

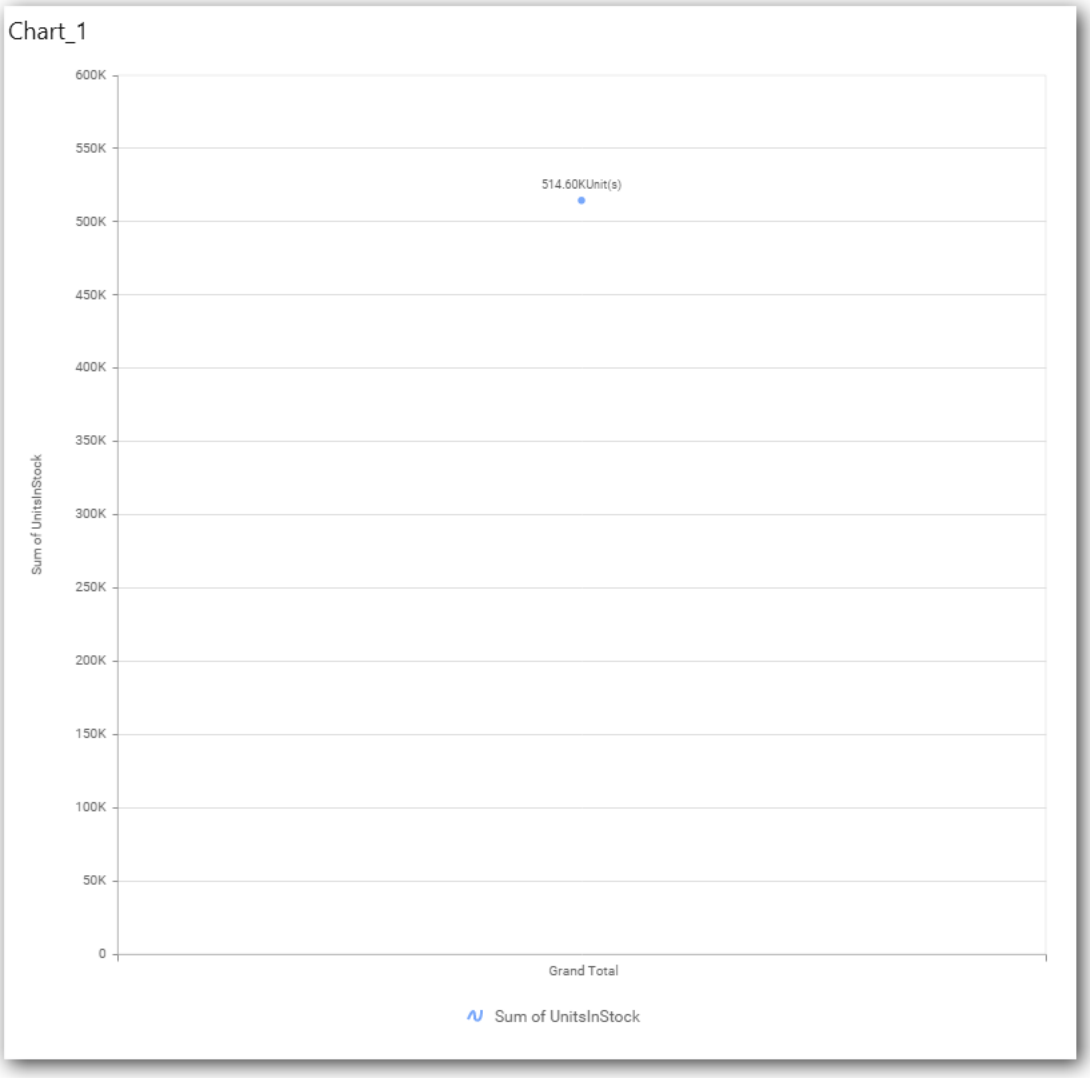
The format options will be shown



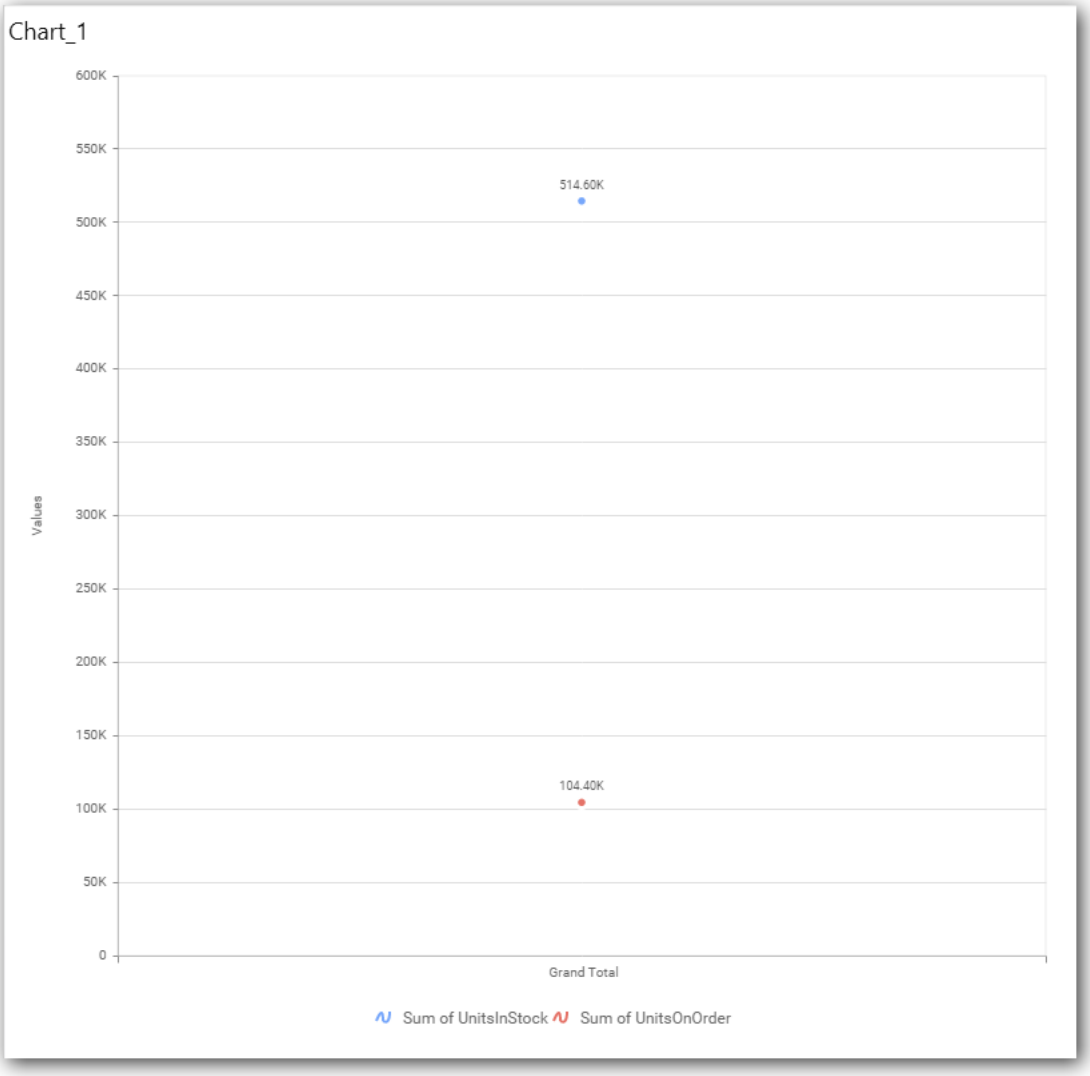
Choose the options you need and click OK.



Now the Chart will be rendered like this.



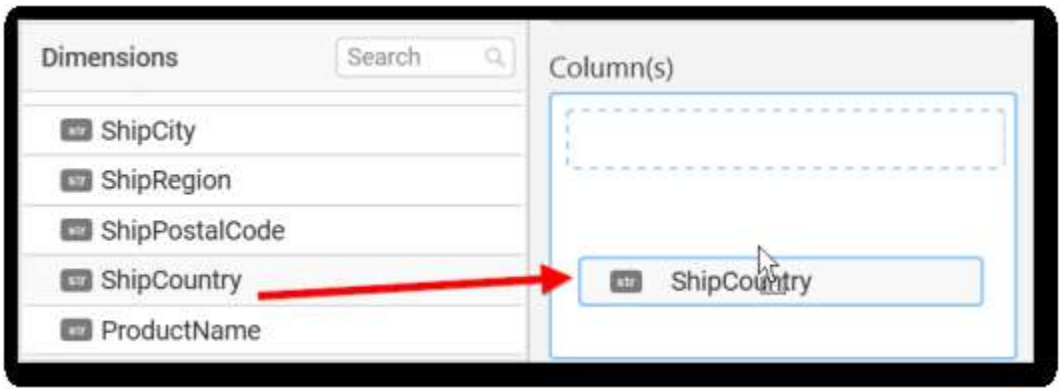
You can add more than one values by drag and drop the measures into value field.



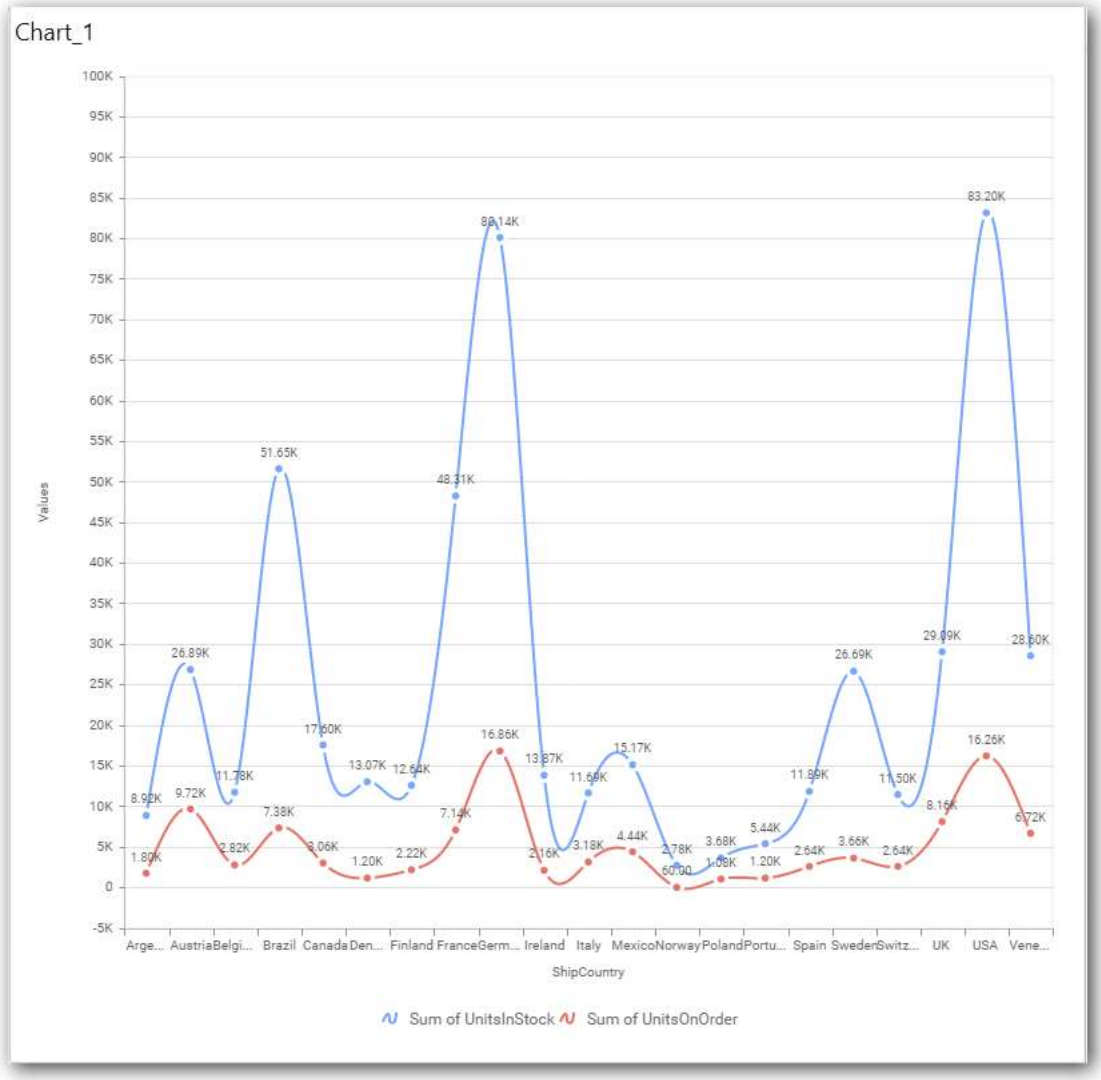
You can also add Dimensions and Columns to Value(s).

Assigning Column(s)

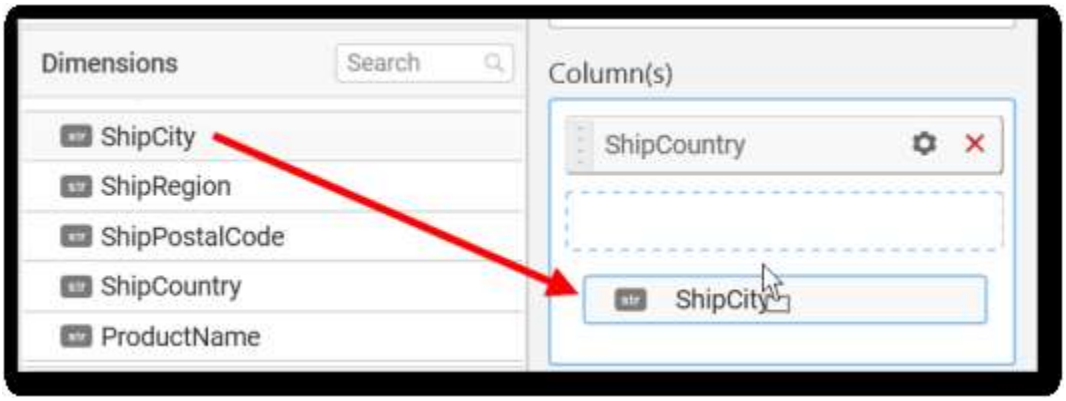
You can add the Dimension into Column field by drag and drop.



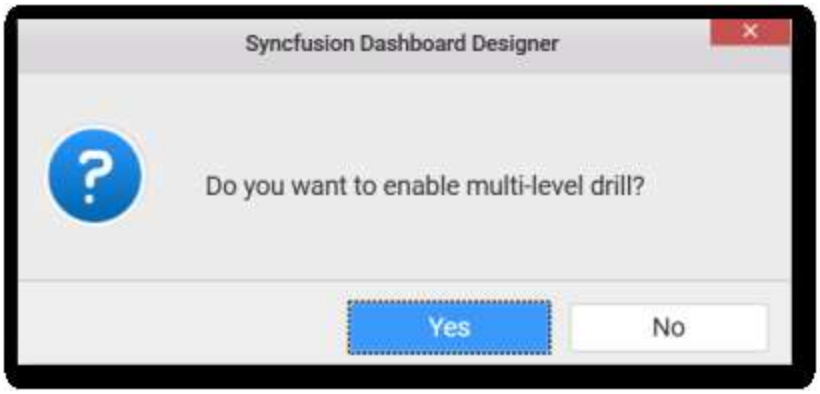
The chart will be rendered as shown in the following figure.



You have option to add more than one Column Value.

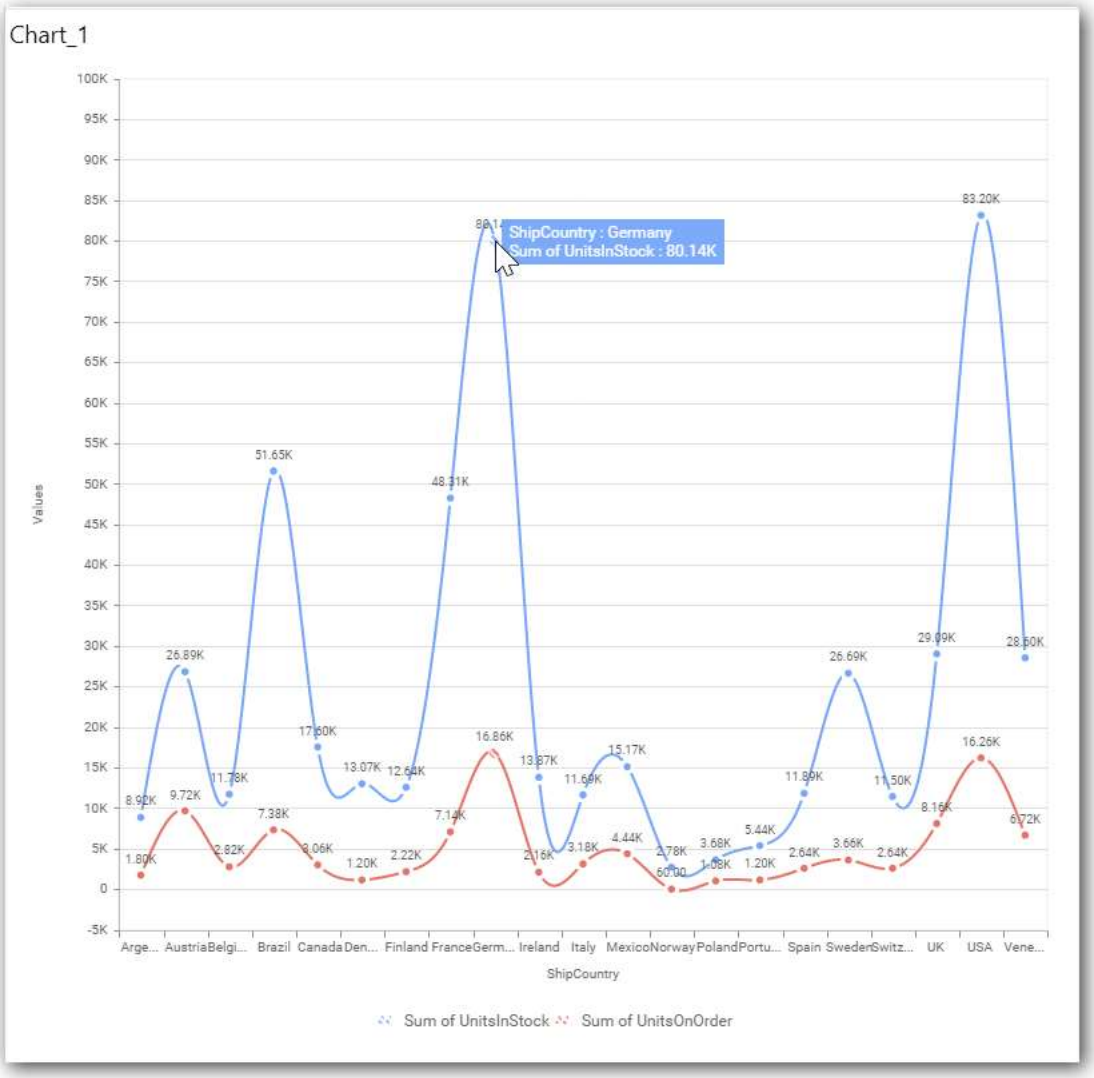


The following alert message will be shown.

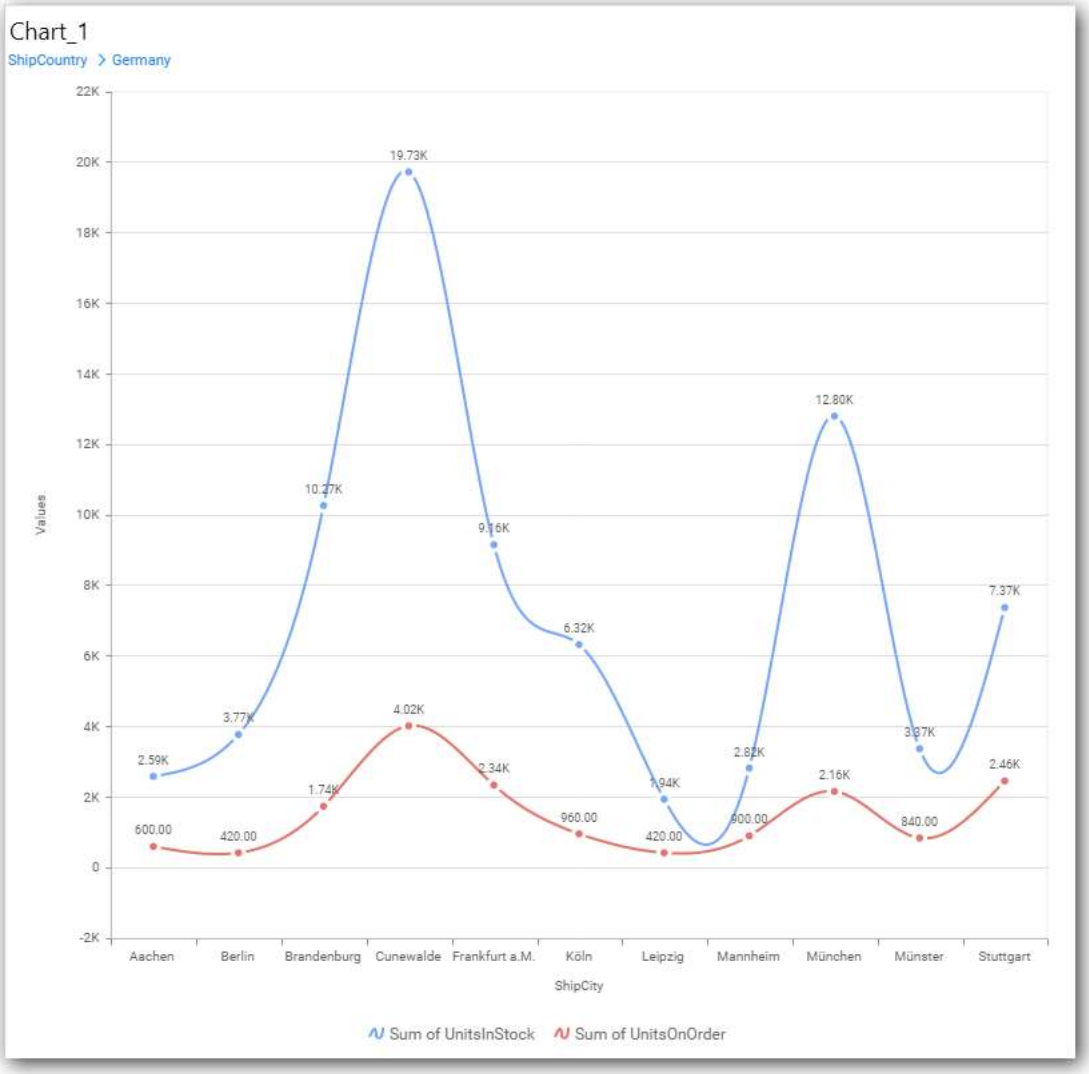


- If you choose **Yes** Drill down option will be enabled.

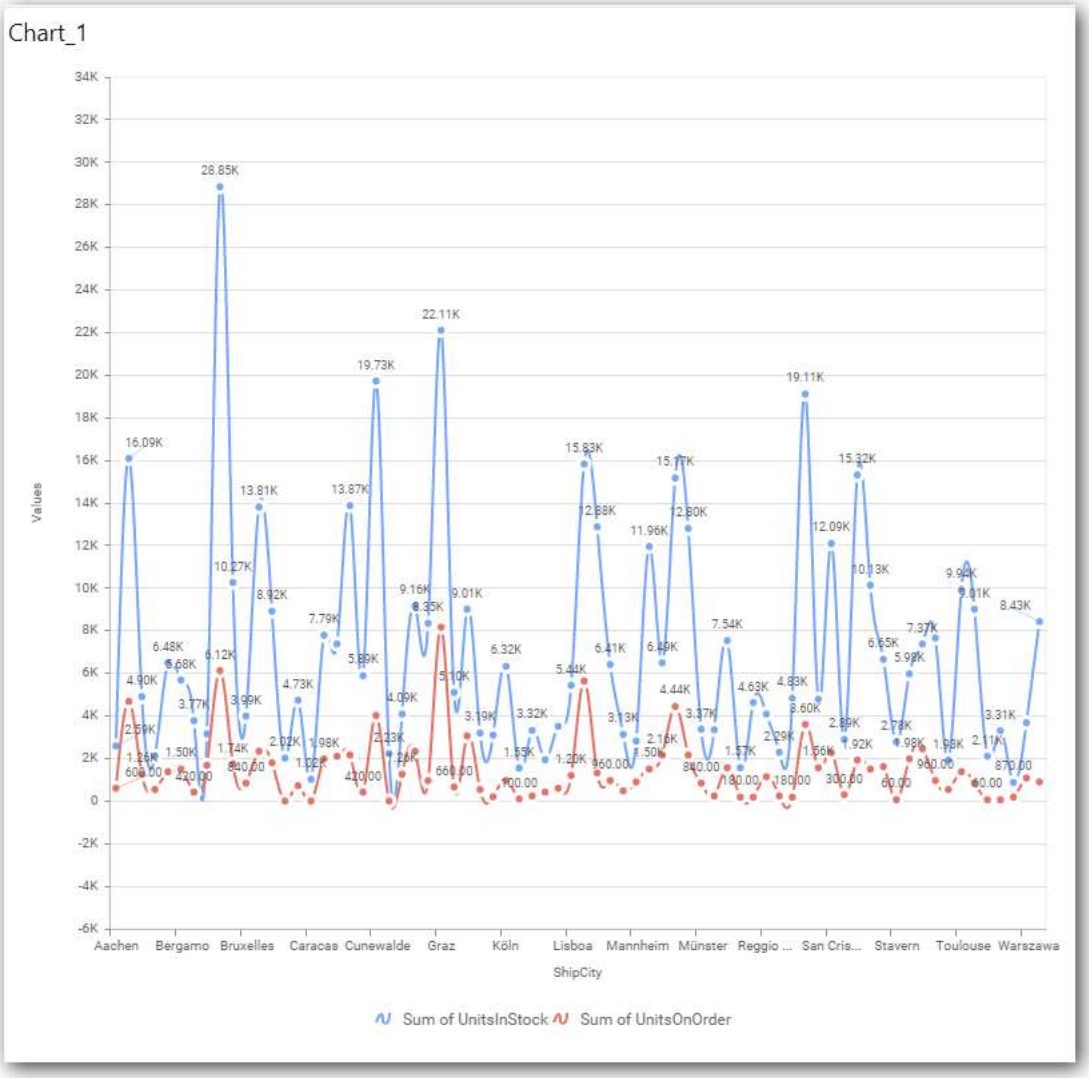
You can drill down the chart by clicking on the chart.



The drilled view of the chart is follows.

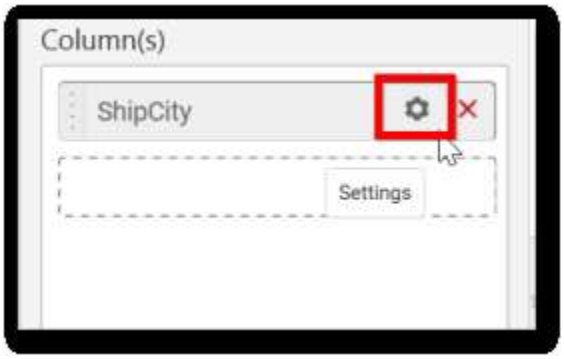


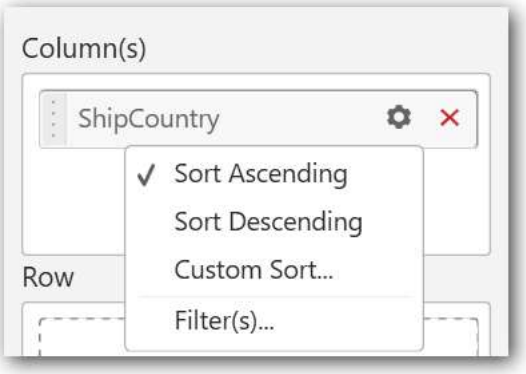
- If you click No the new Dimension value will replace old value.



You can also add Measures and Expression Columns into Column(s) field.

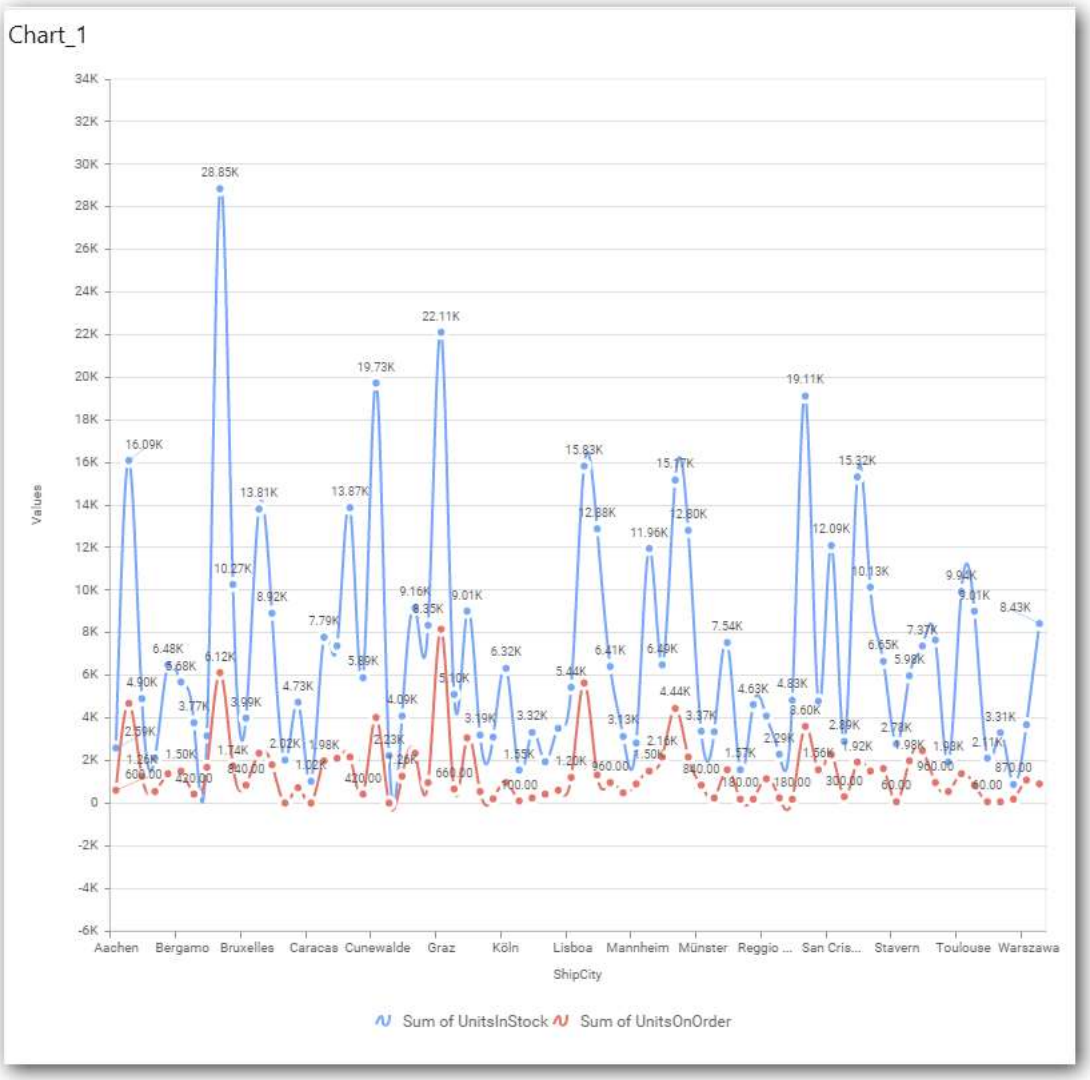
You have options to change the Settings.



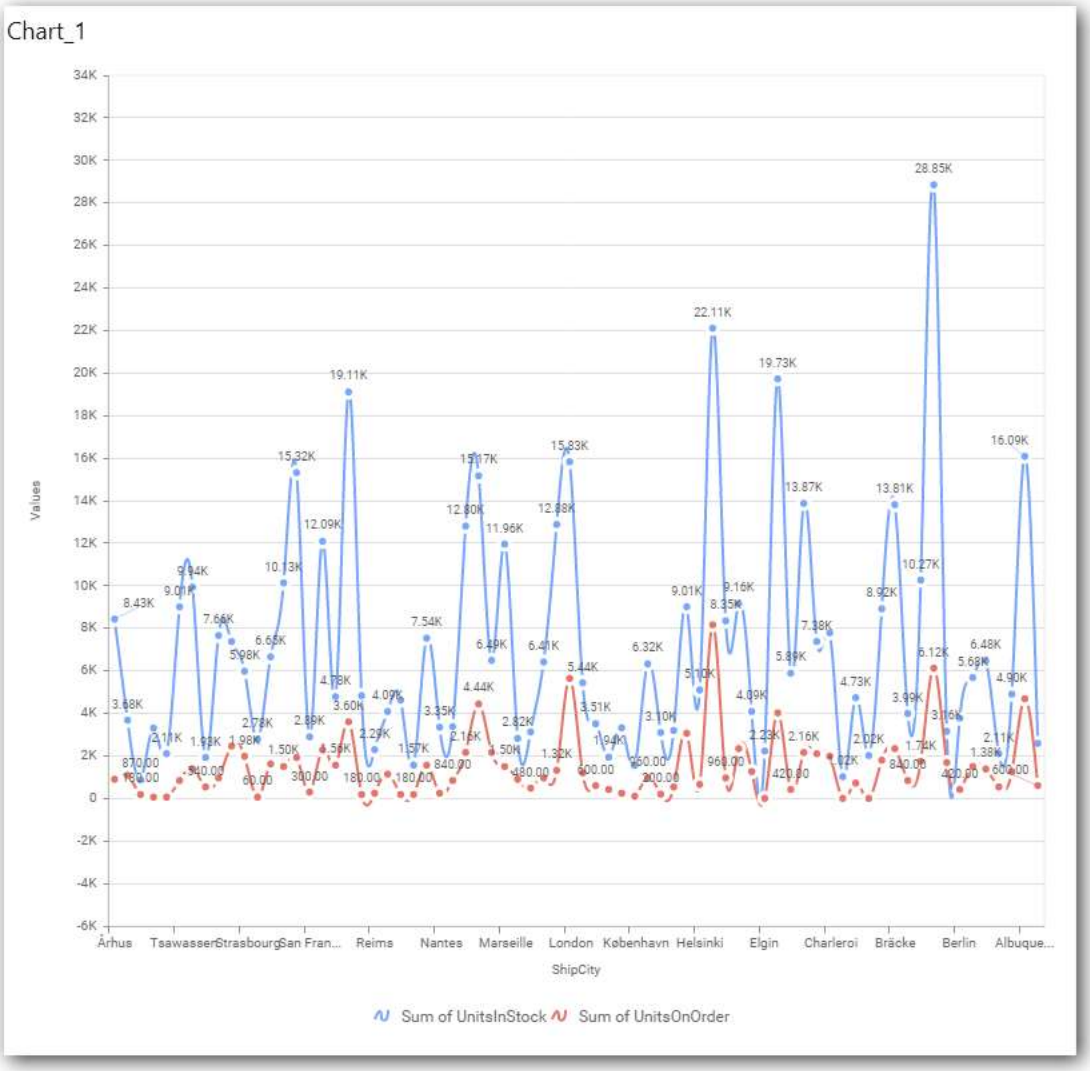


You can sort the chart either in Ascending or Descending series.

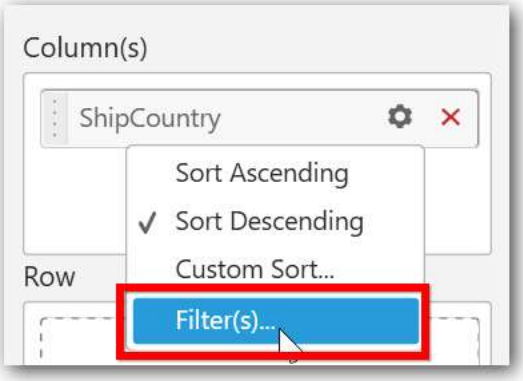
Ascending Order

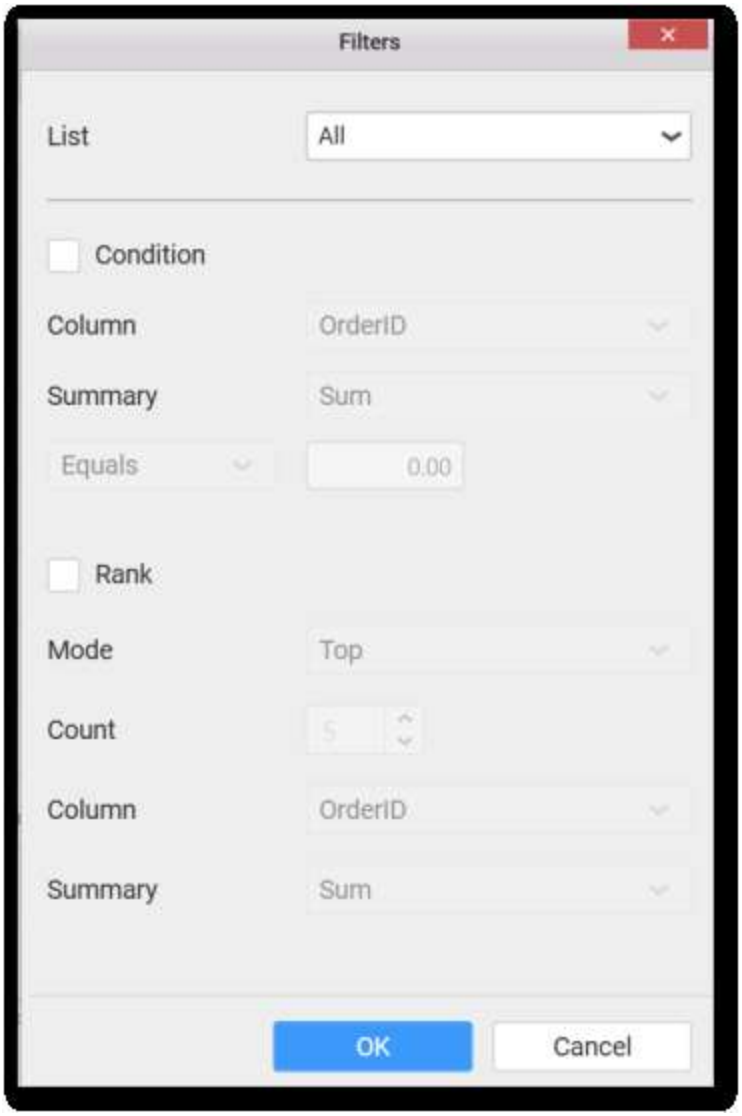


Descending order

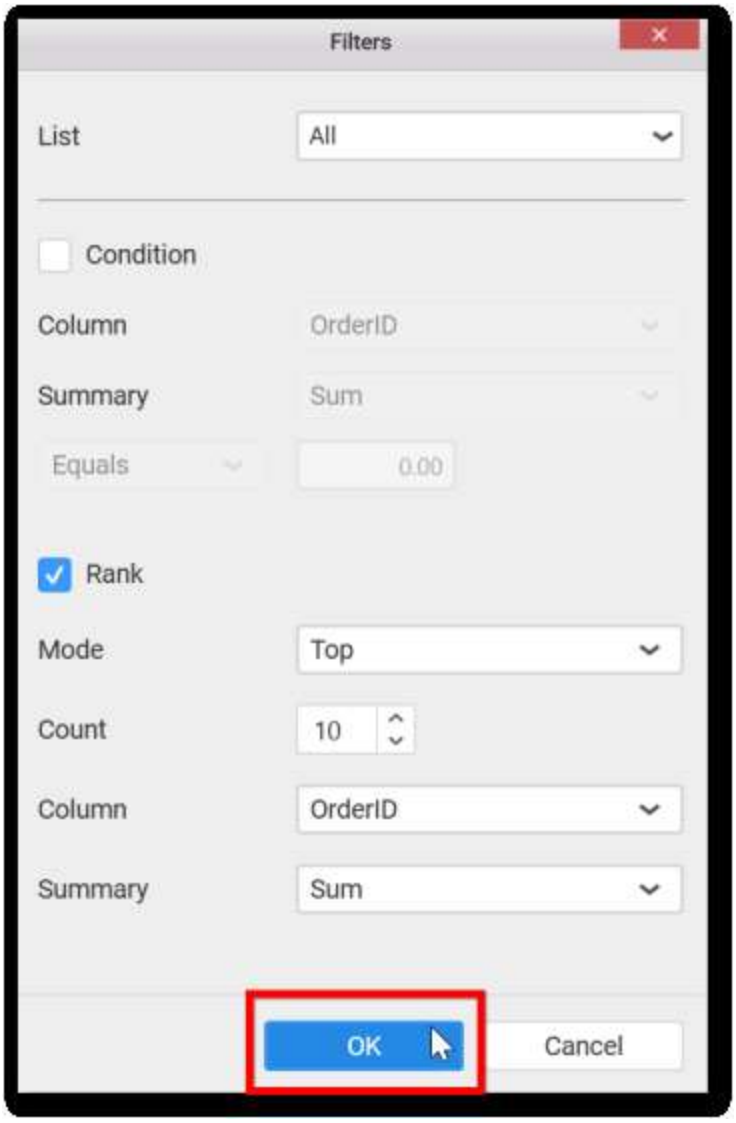


You can apply a filter.

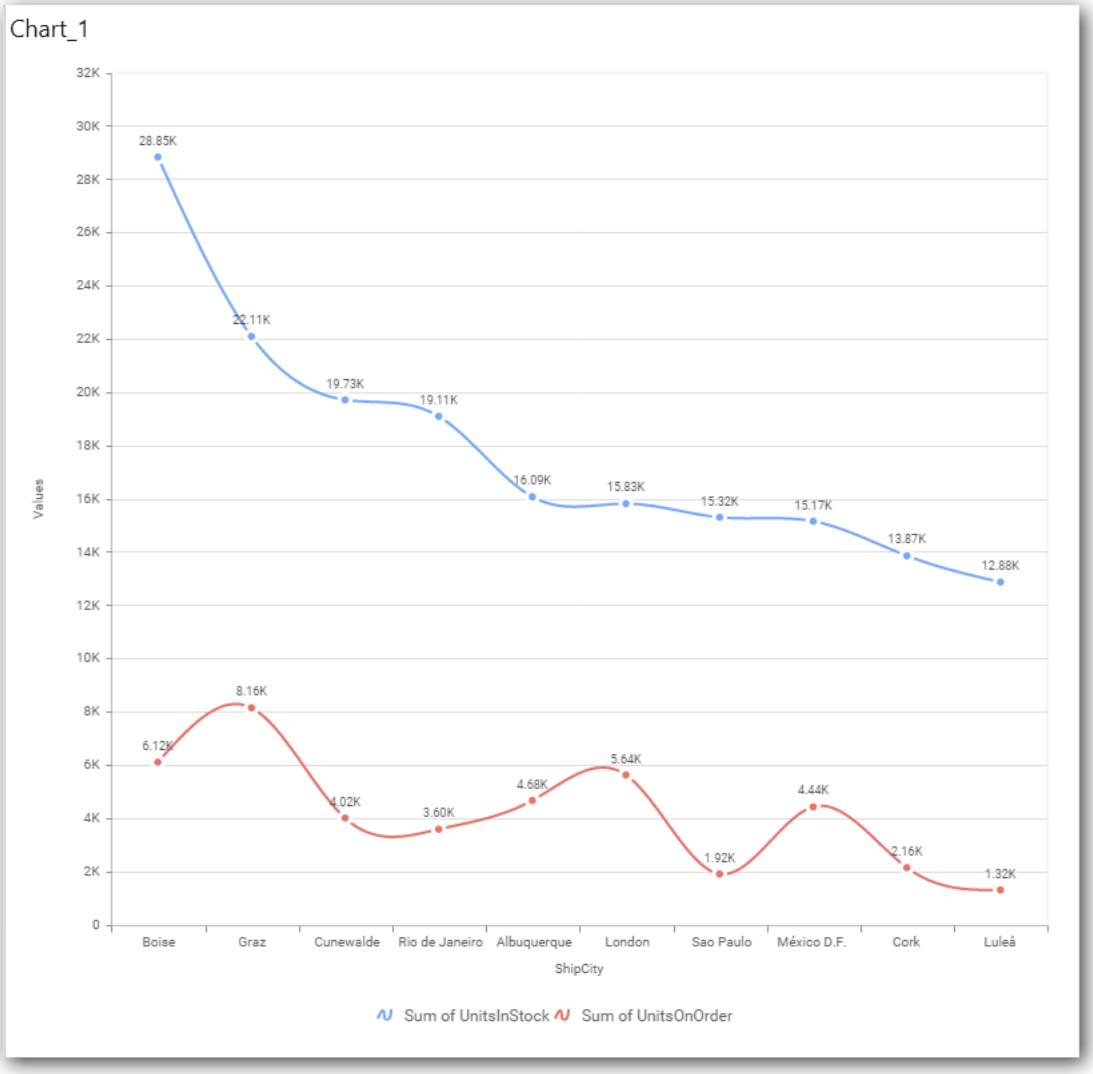




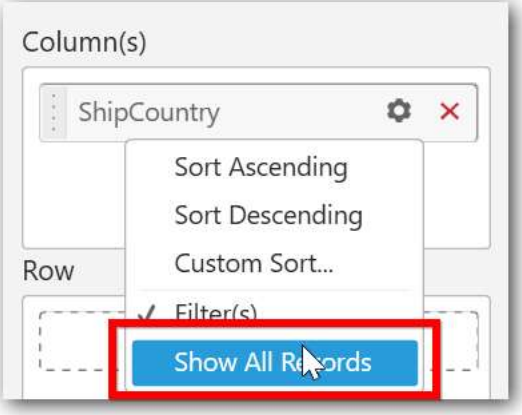
Select the Conditions and Rank you need.



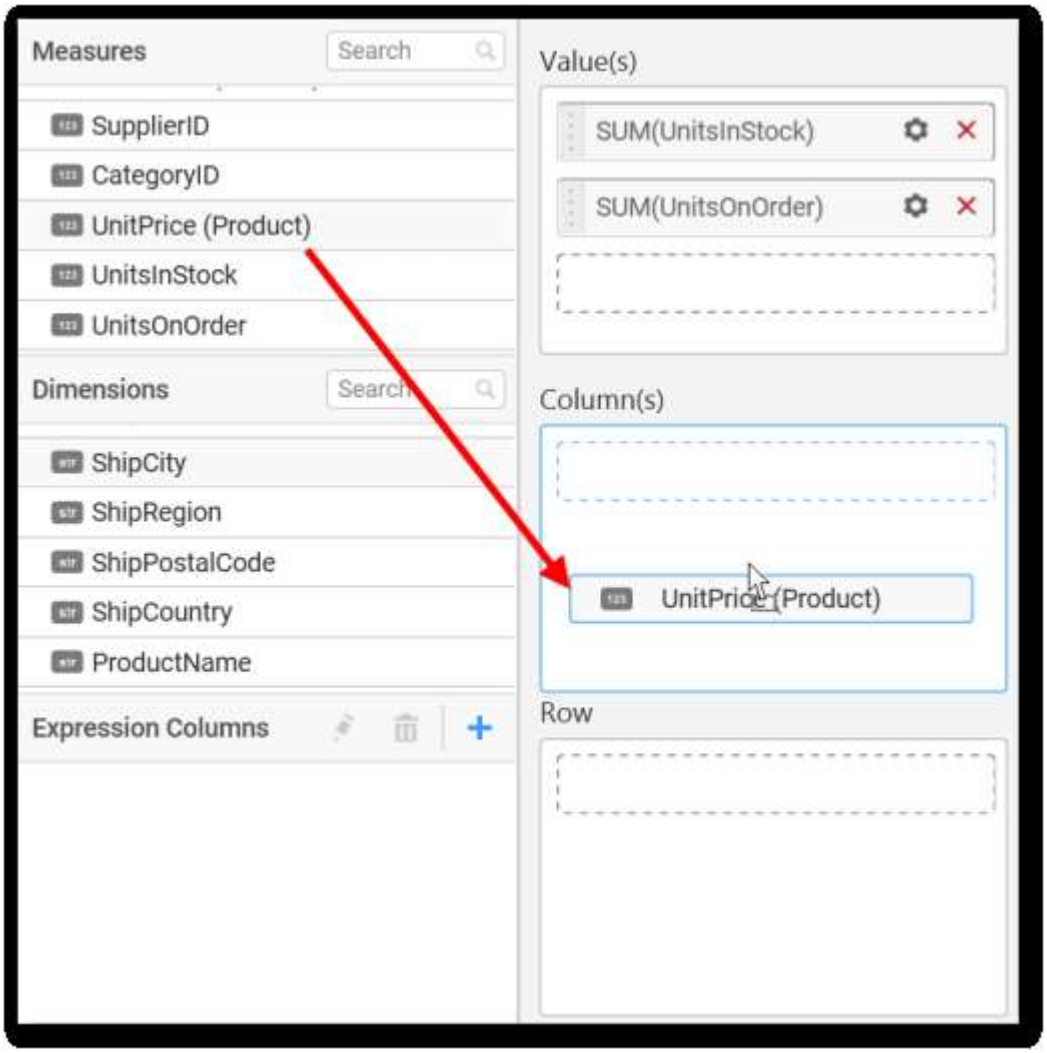
Now the chart will be rendered like this.



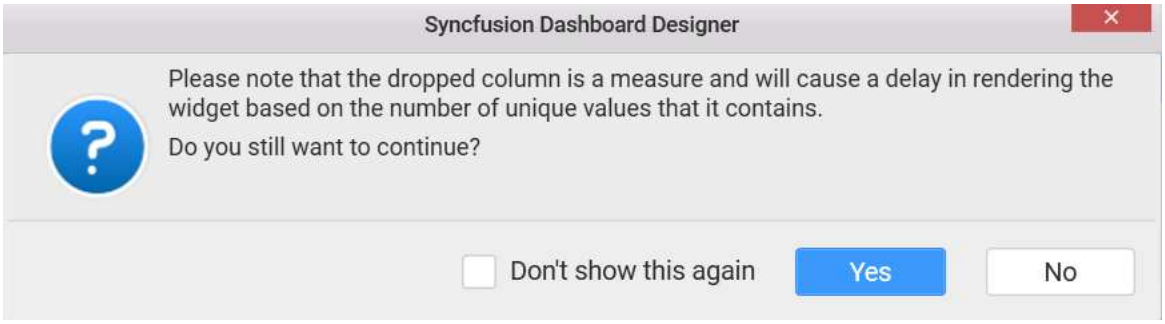
To show all records again click on Show all records.



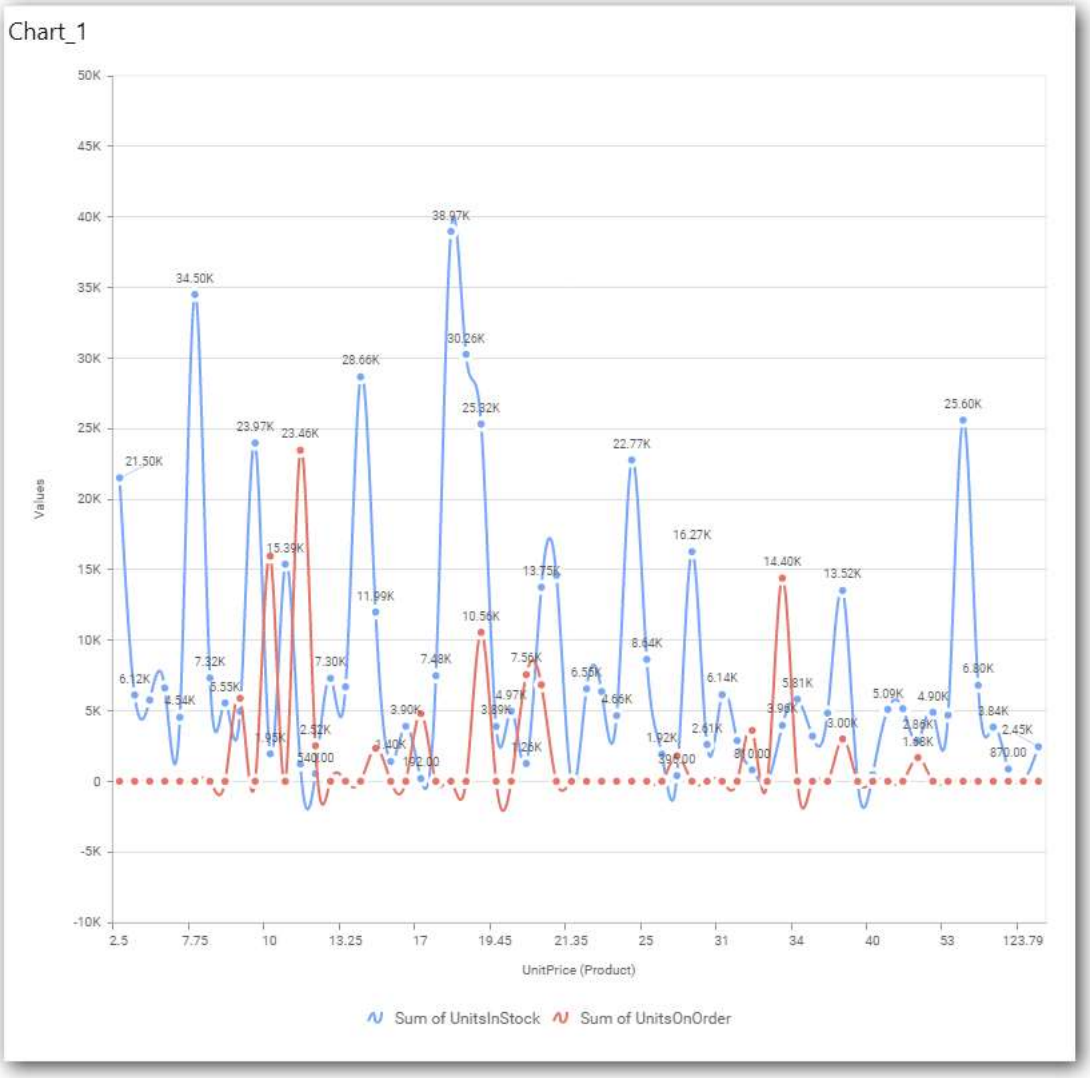
You can add Measures into Column(s).



The following alert will shown.

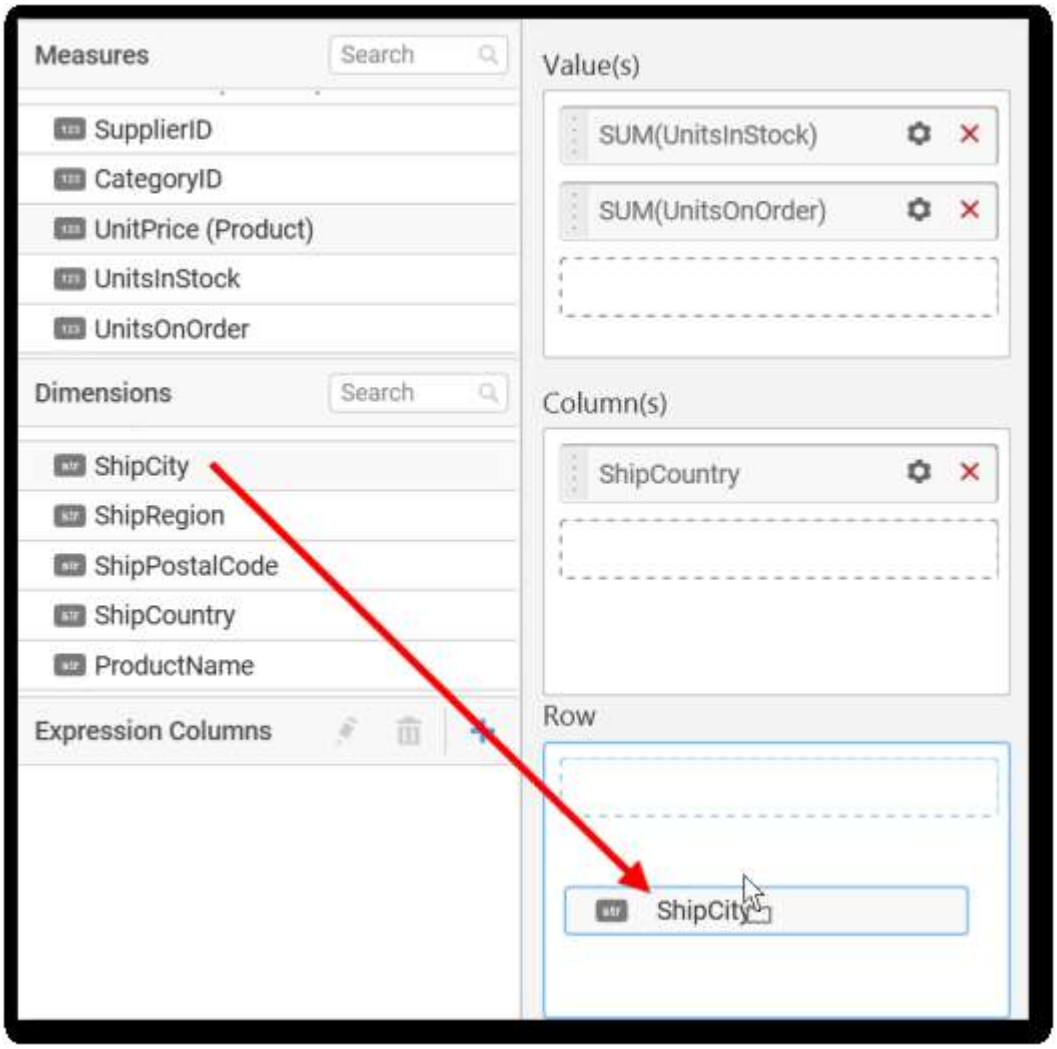


To continue click Yes otherwise click on NO.

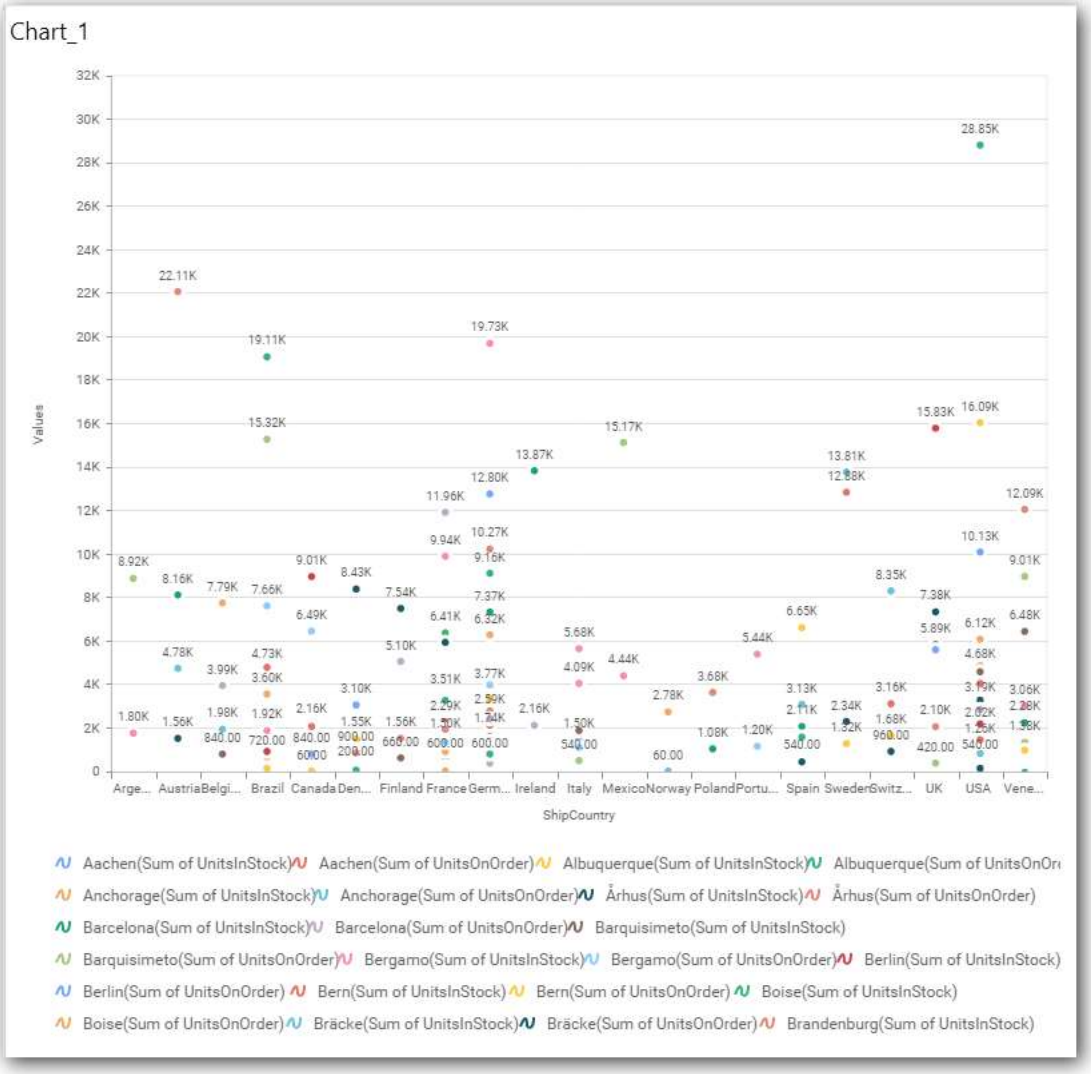


Assigning Row

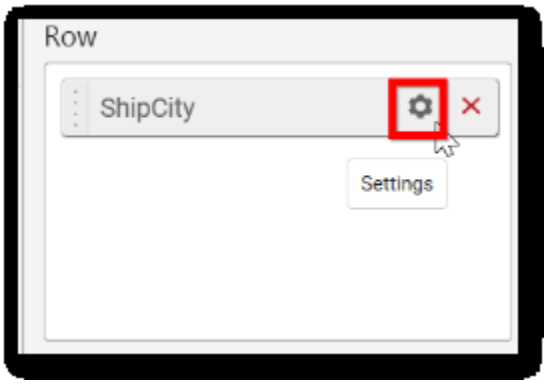
You can add Dimension into the Row field for series chart.

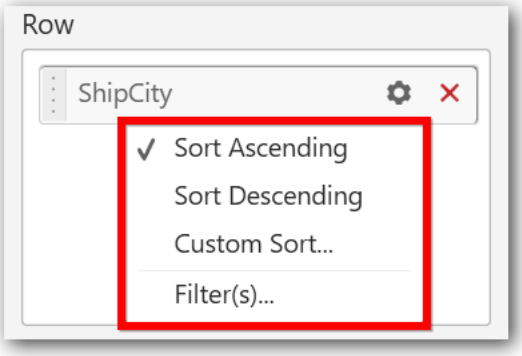


The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s).



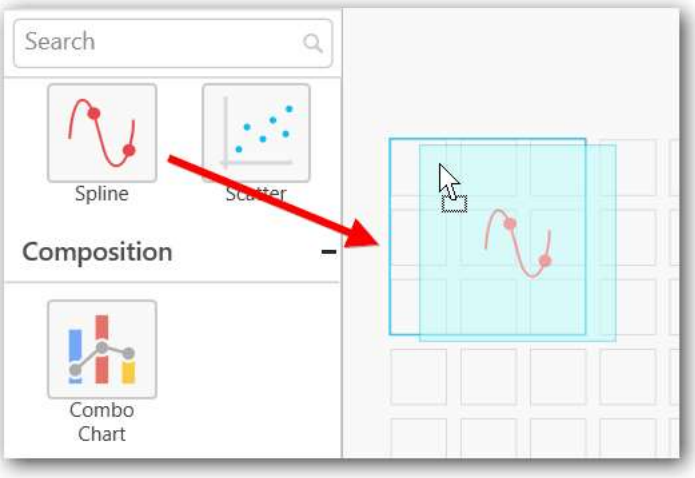


How to configure SSAS Data to Spline Chart?

Spline Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

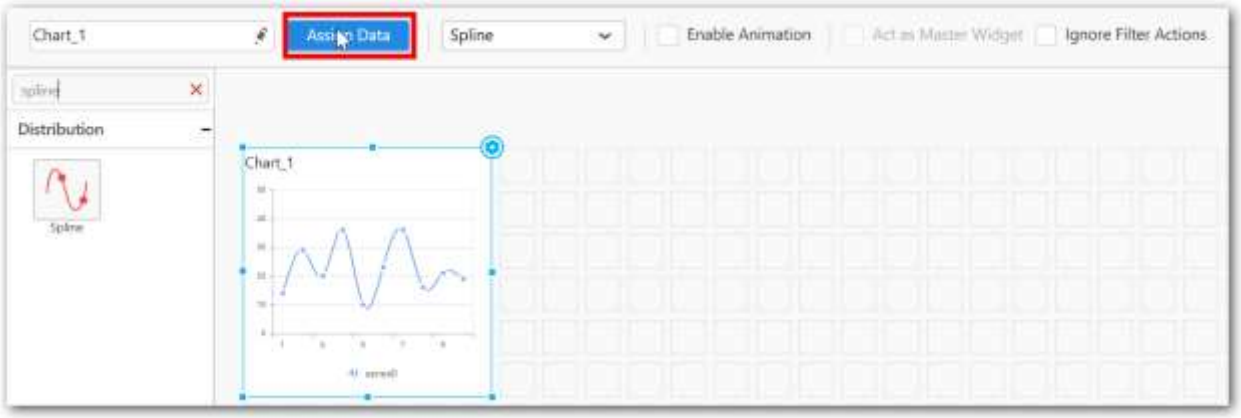
Following steps illustrates configuration of SSAS data to Spline chart

Drag and drop the Spline chart widget into canvas and resize into your required size.

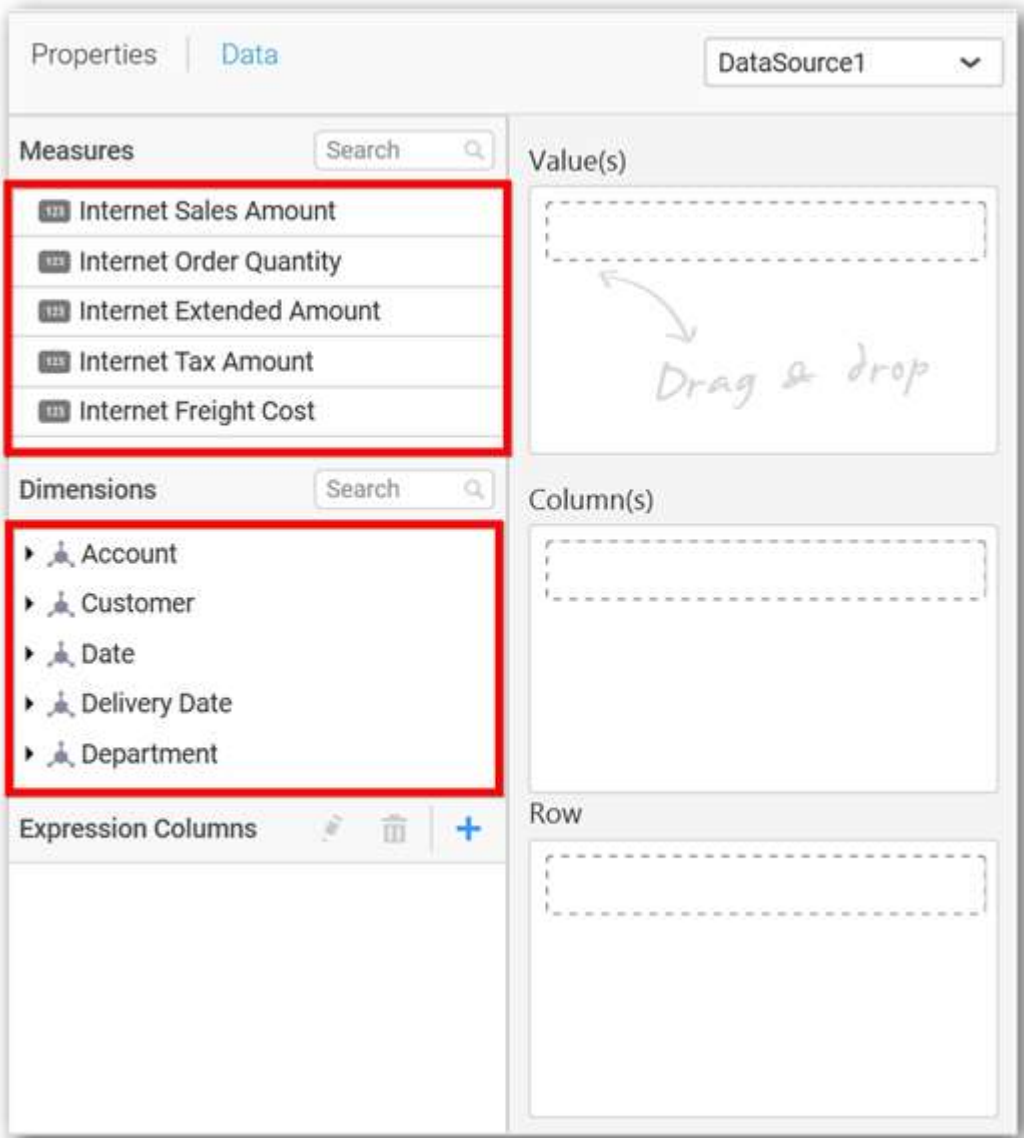


Select the dropped widget using mouse.

Click the Assign Data button in the toolbar.

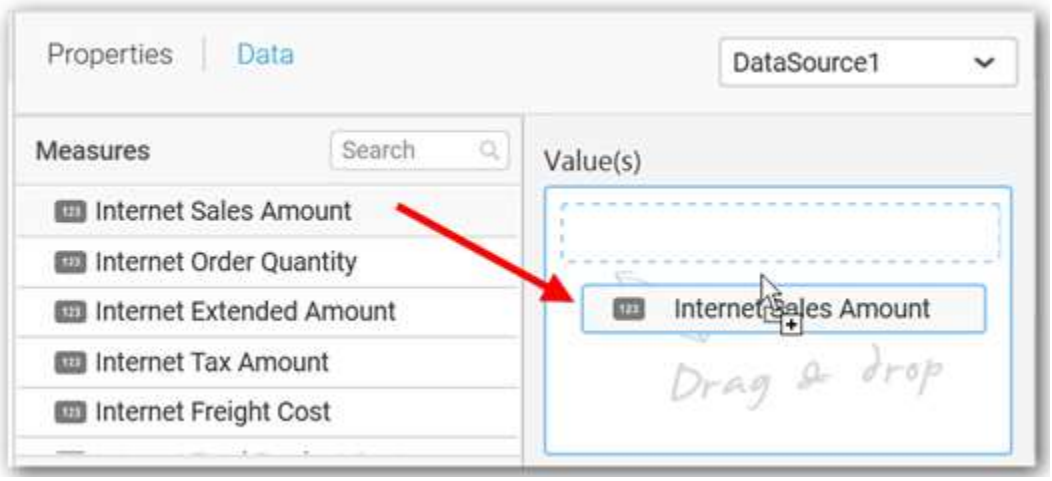


A Data pane will be opened with available Measures and Dimensions

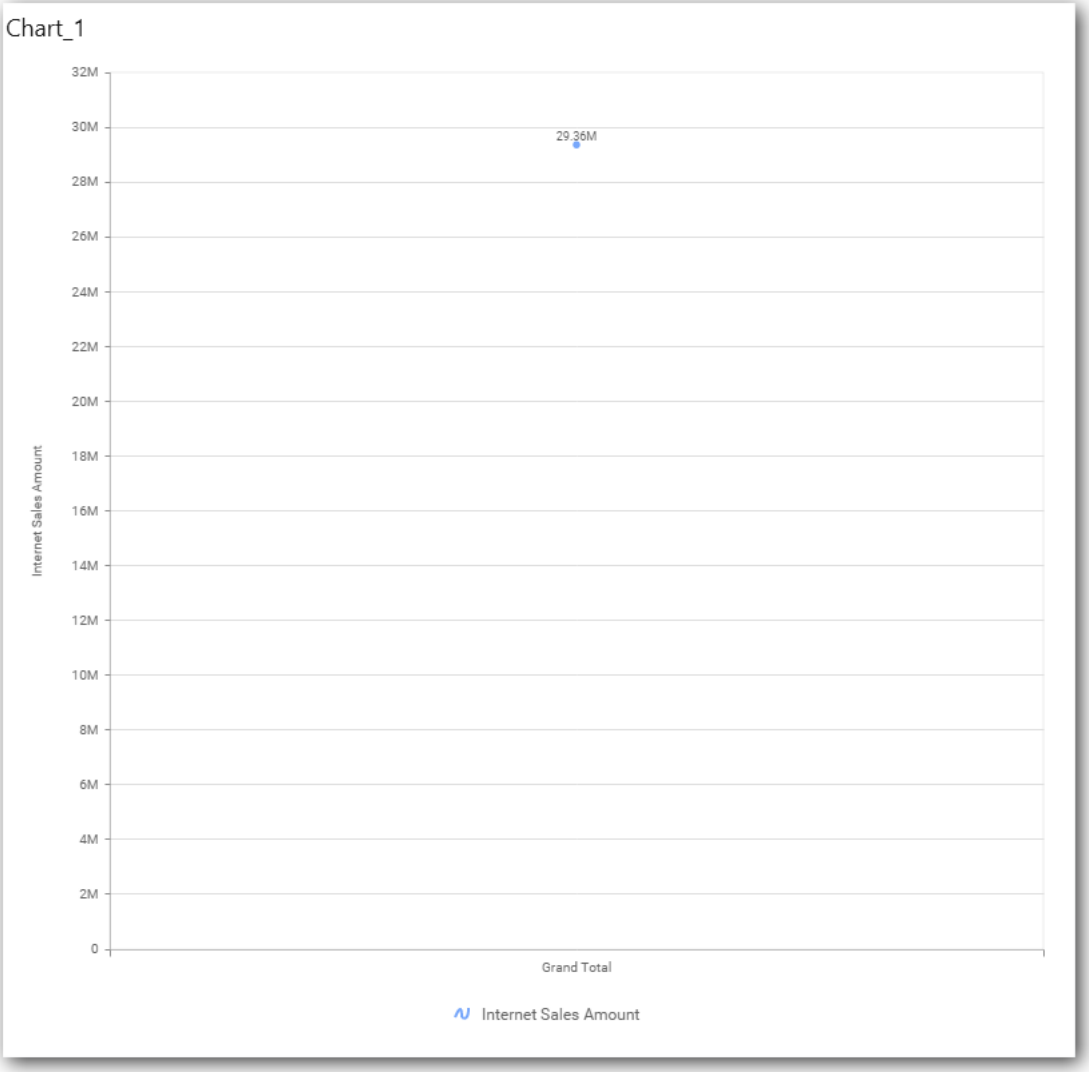


Assigning Value(s)

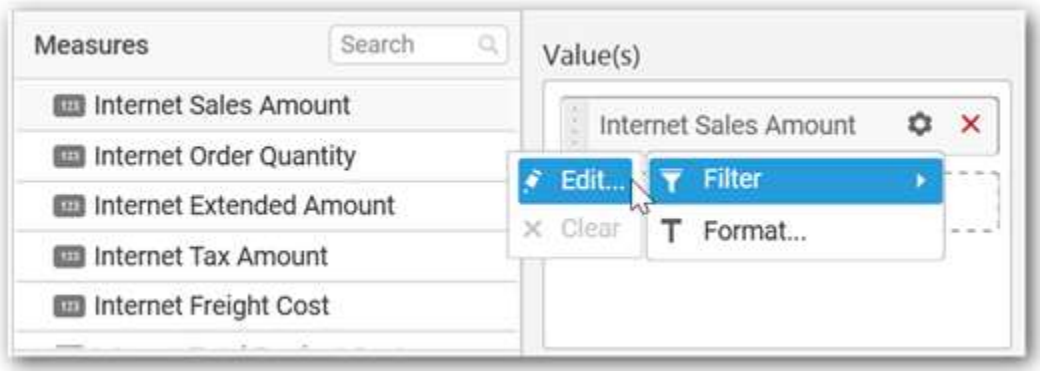
Drag and drop a column under Measures category into Value(s) section.



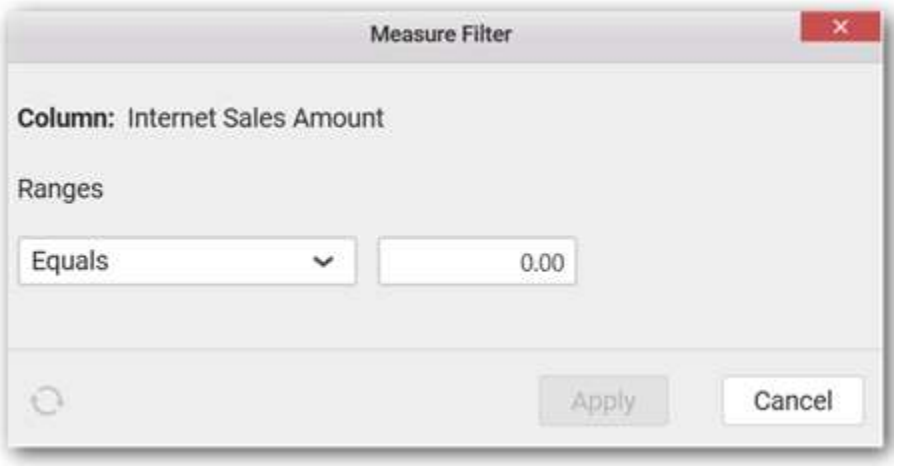
Now the chart will be rendered like this.



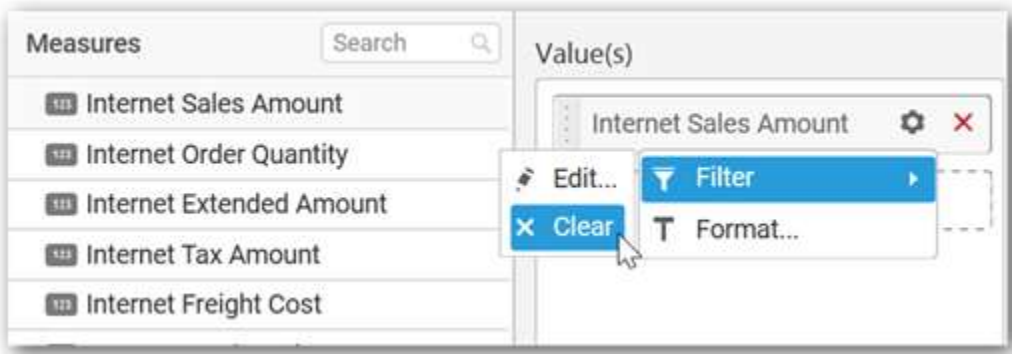
Define the filter criteria to match through choosing Edit option in Filter menu item.



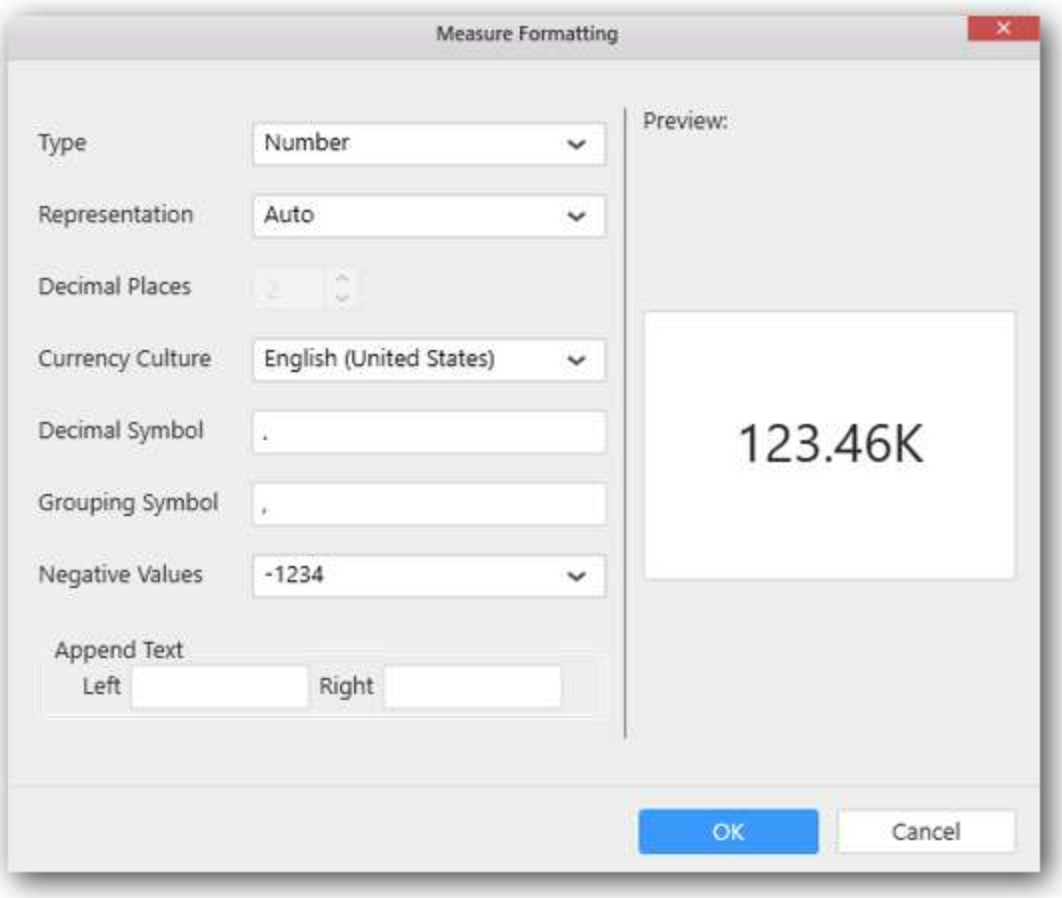
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



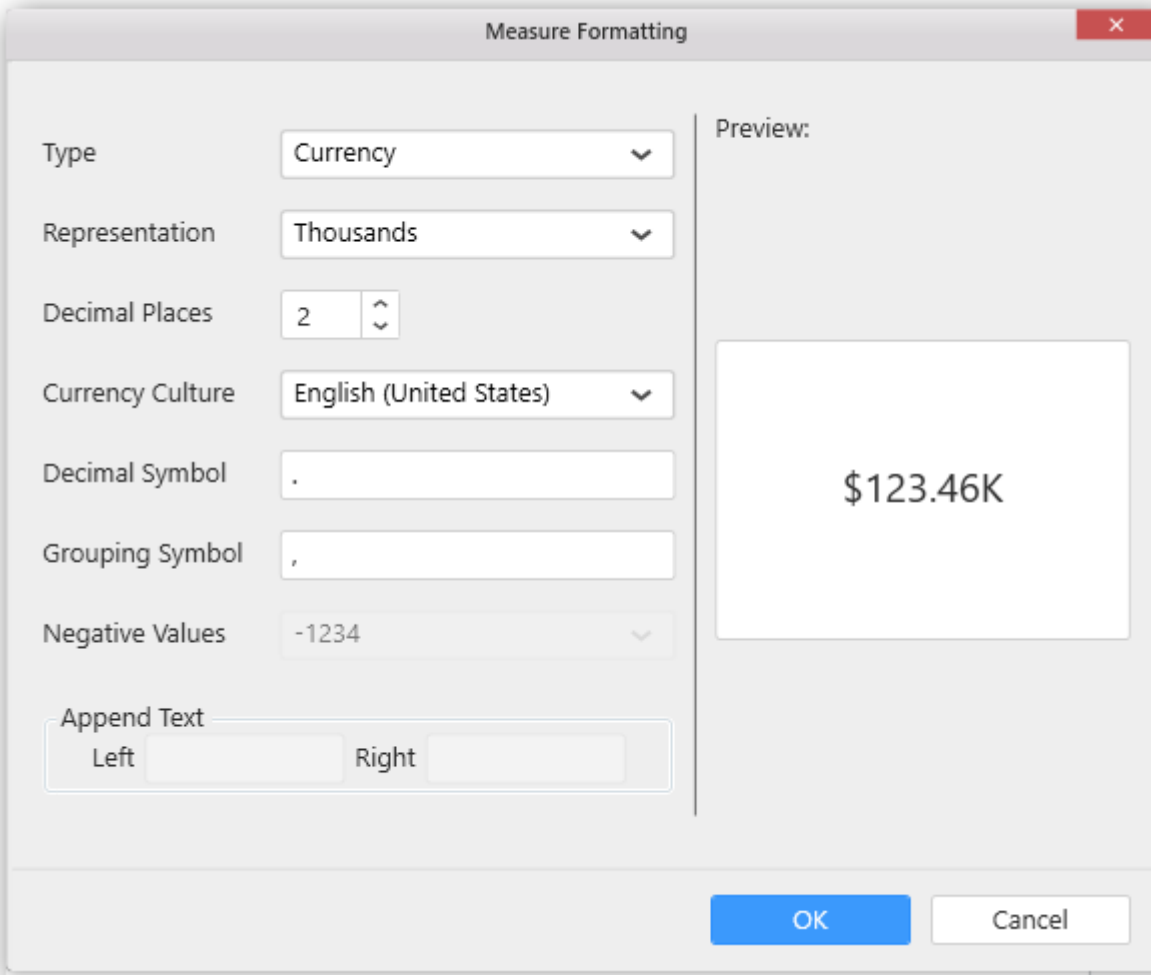
Select Clear option to clear the defined filter.



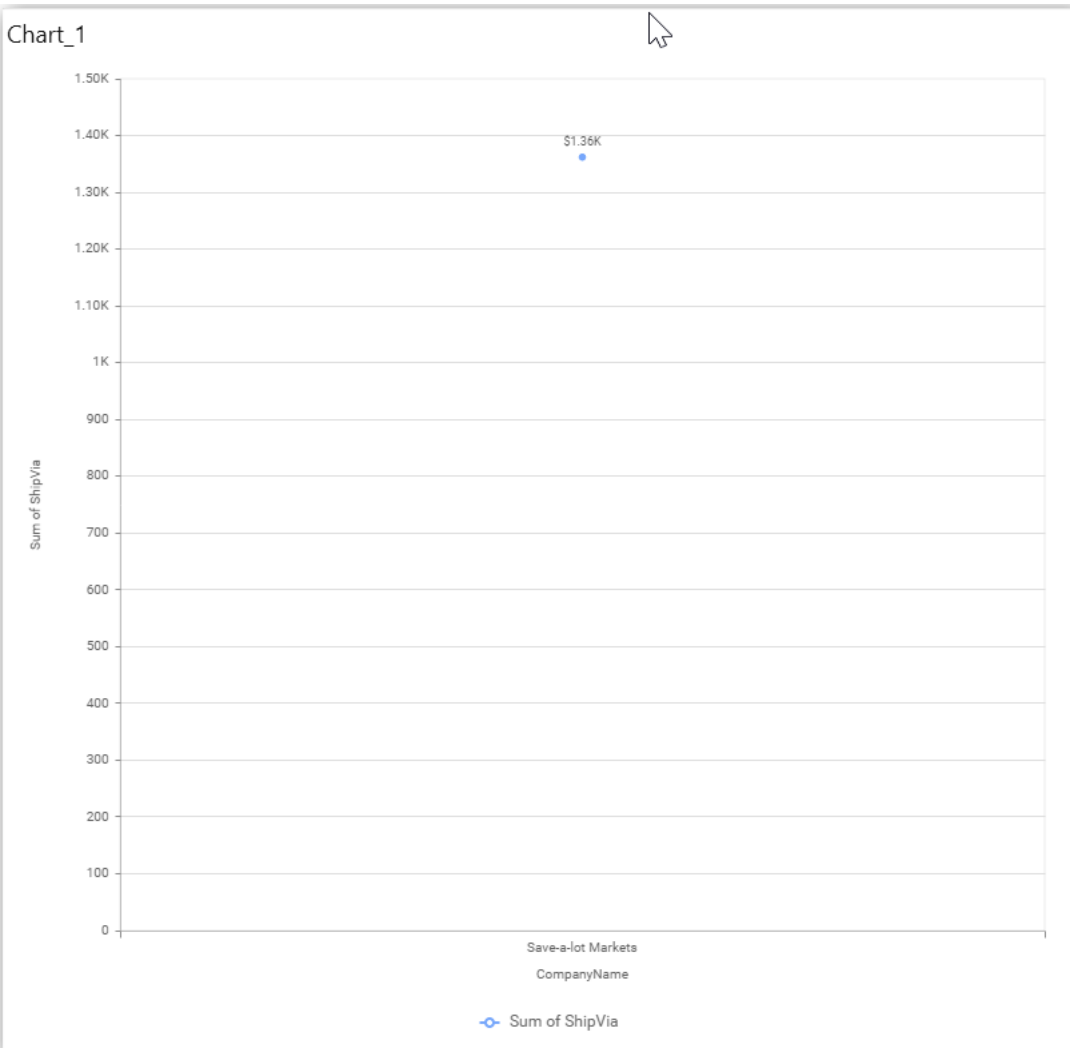
Select Format option to define the display format to the values in the column through Measure Formatting window.



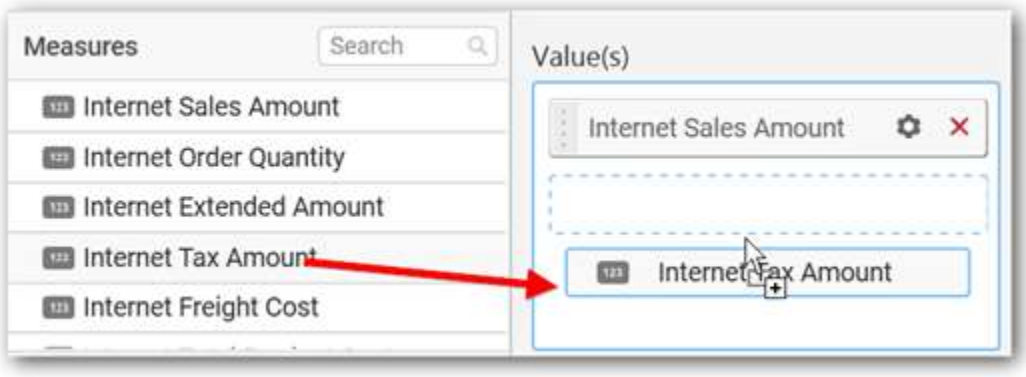
Choose the options you need and click OK.

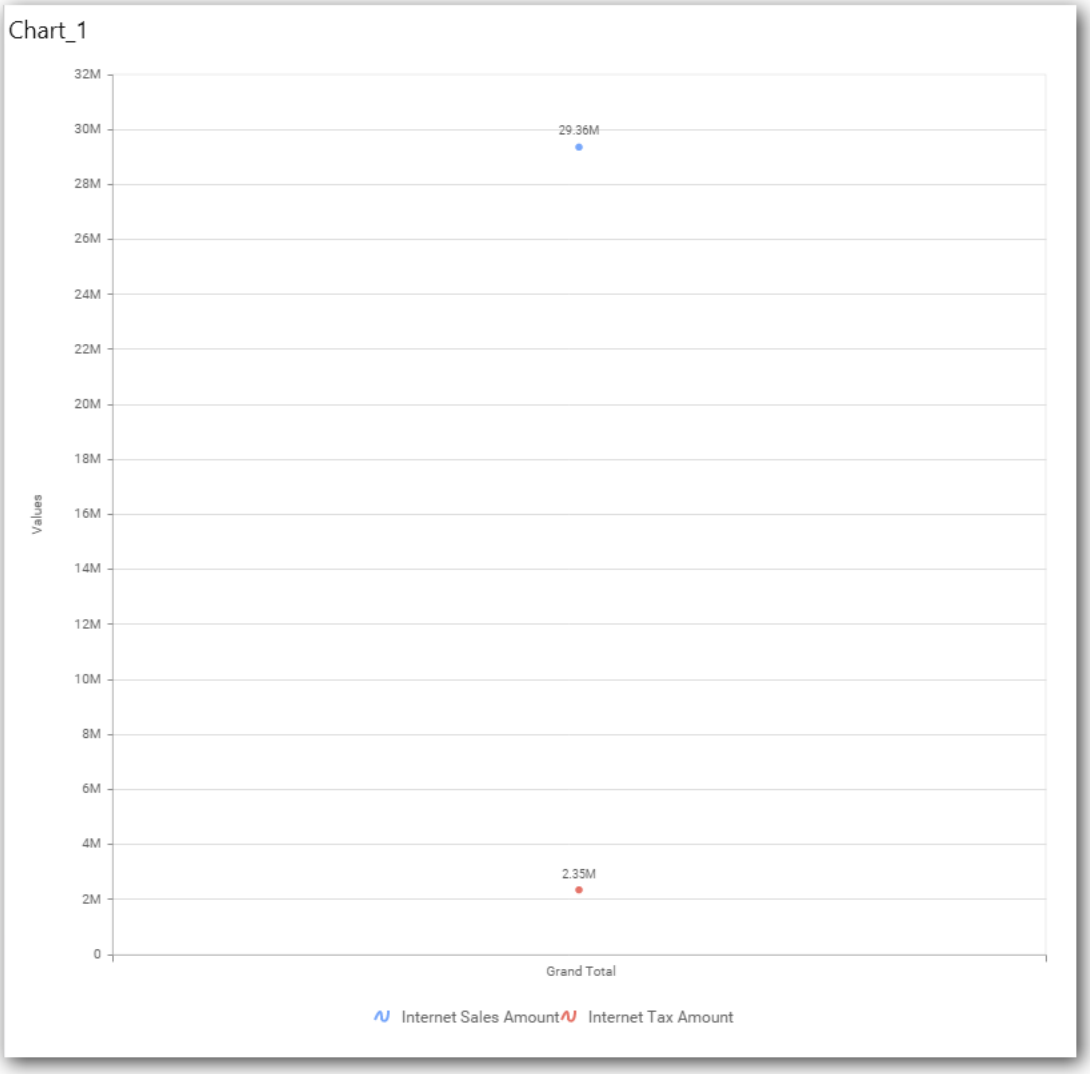


Now the Chart will be rendered like this.



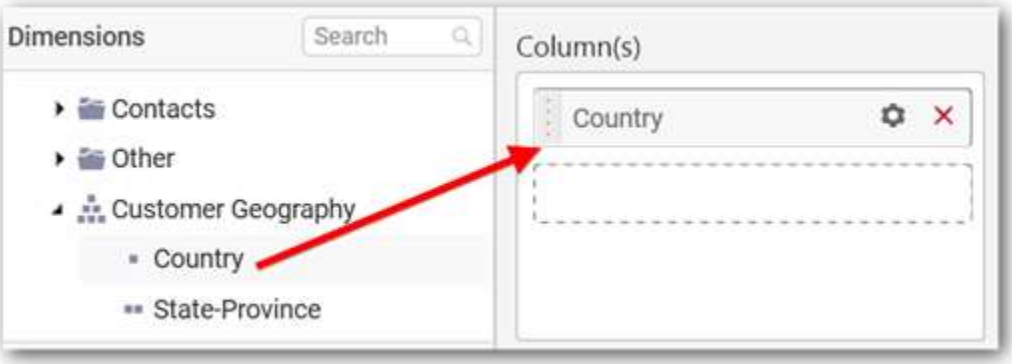
You can add more number values by drag drop the Measures into Value(s) field.

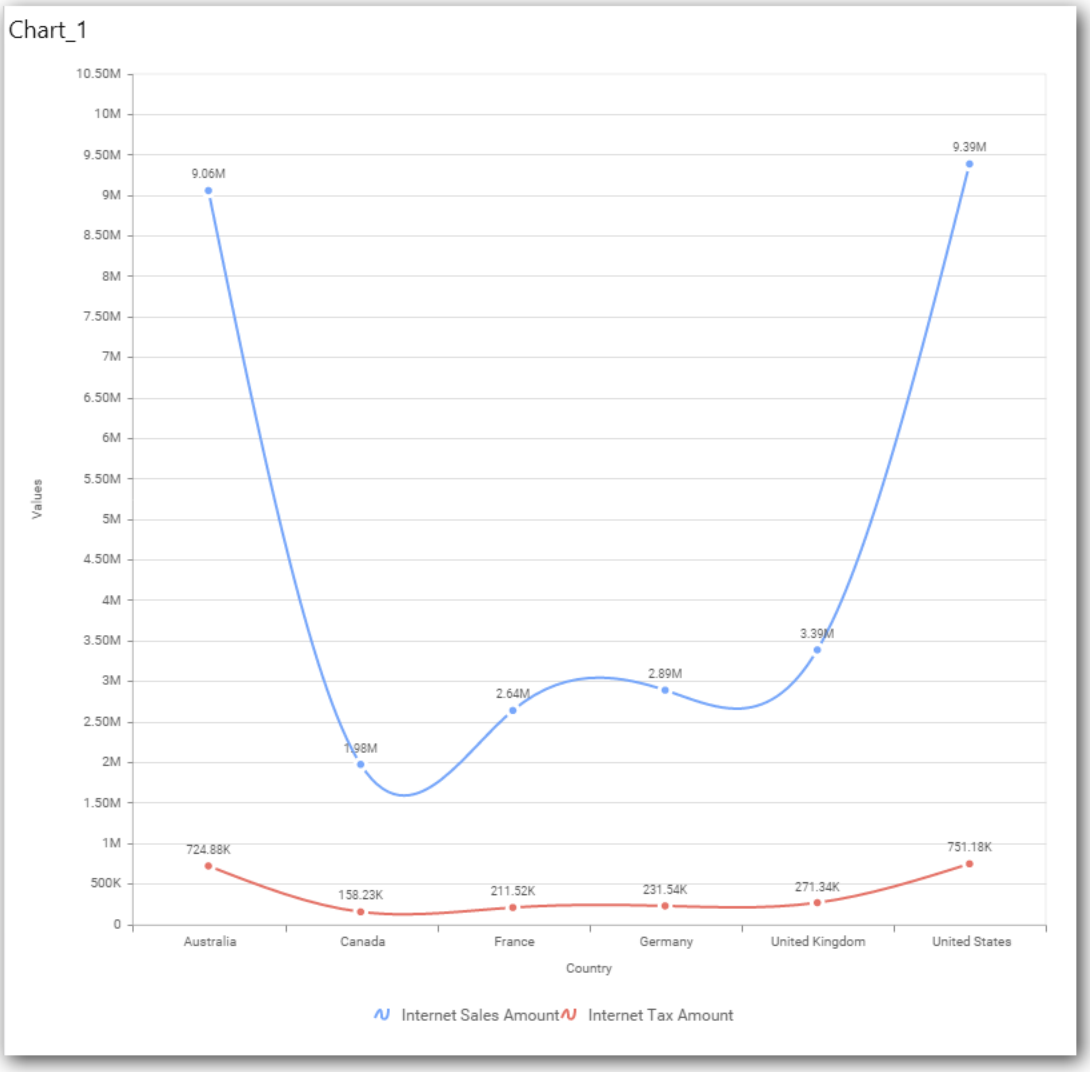




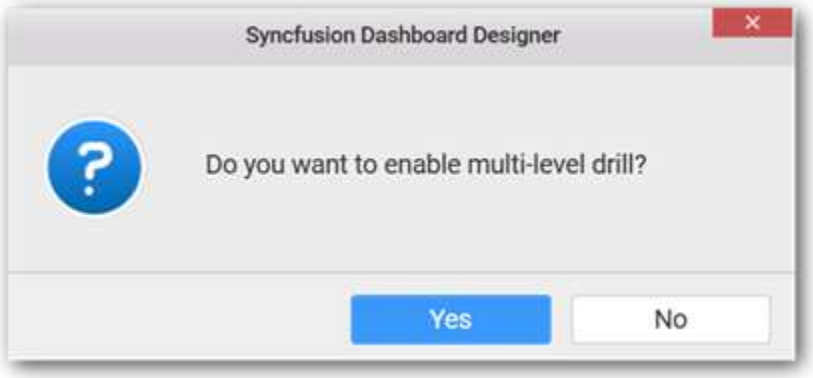
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.



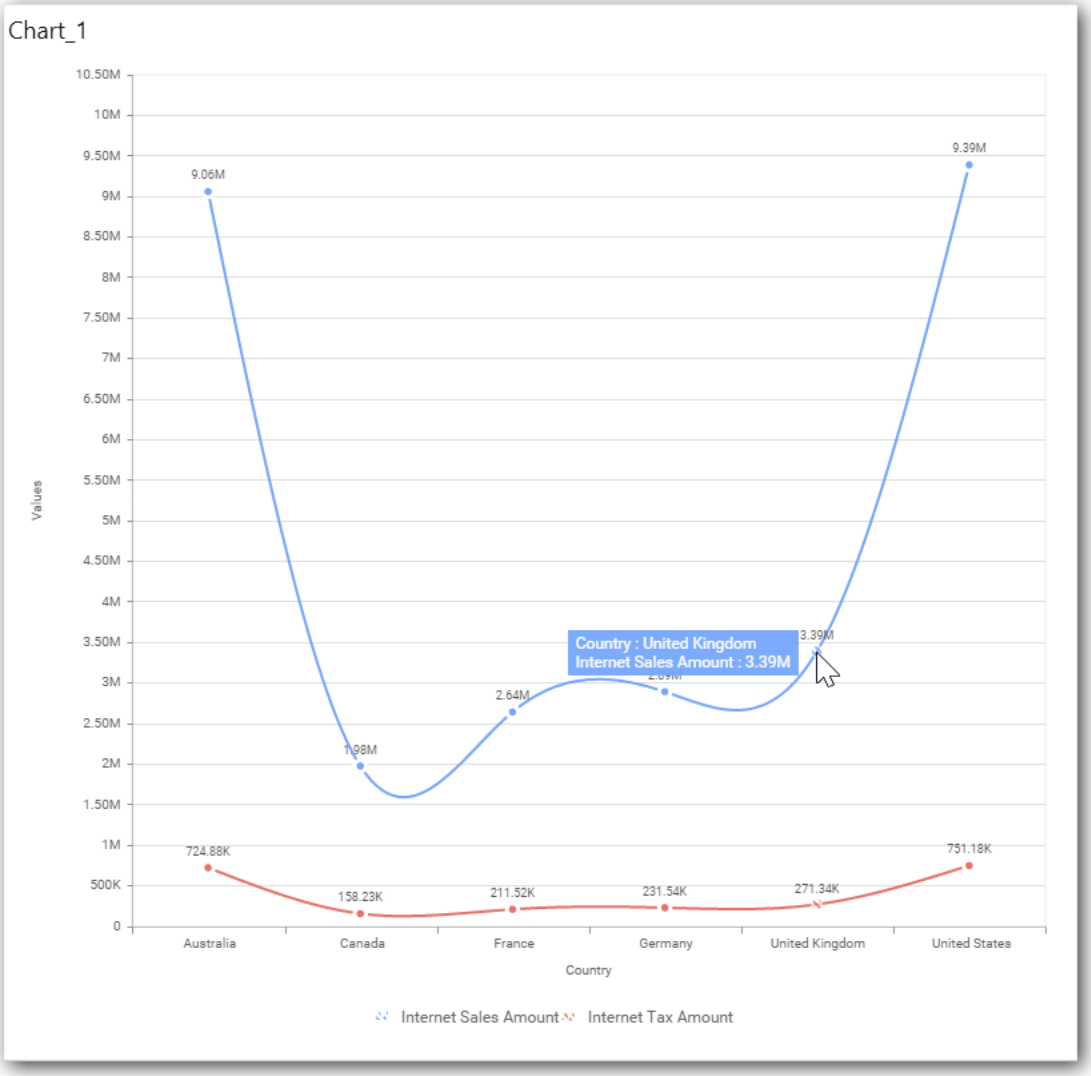


You may also add more than one column into Column(s) section. In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

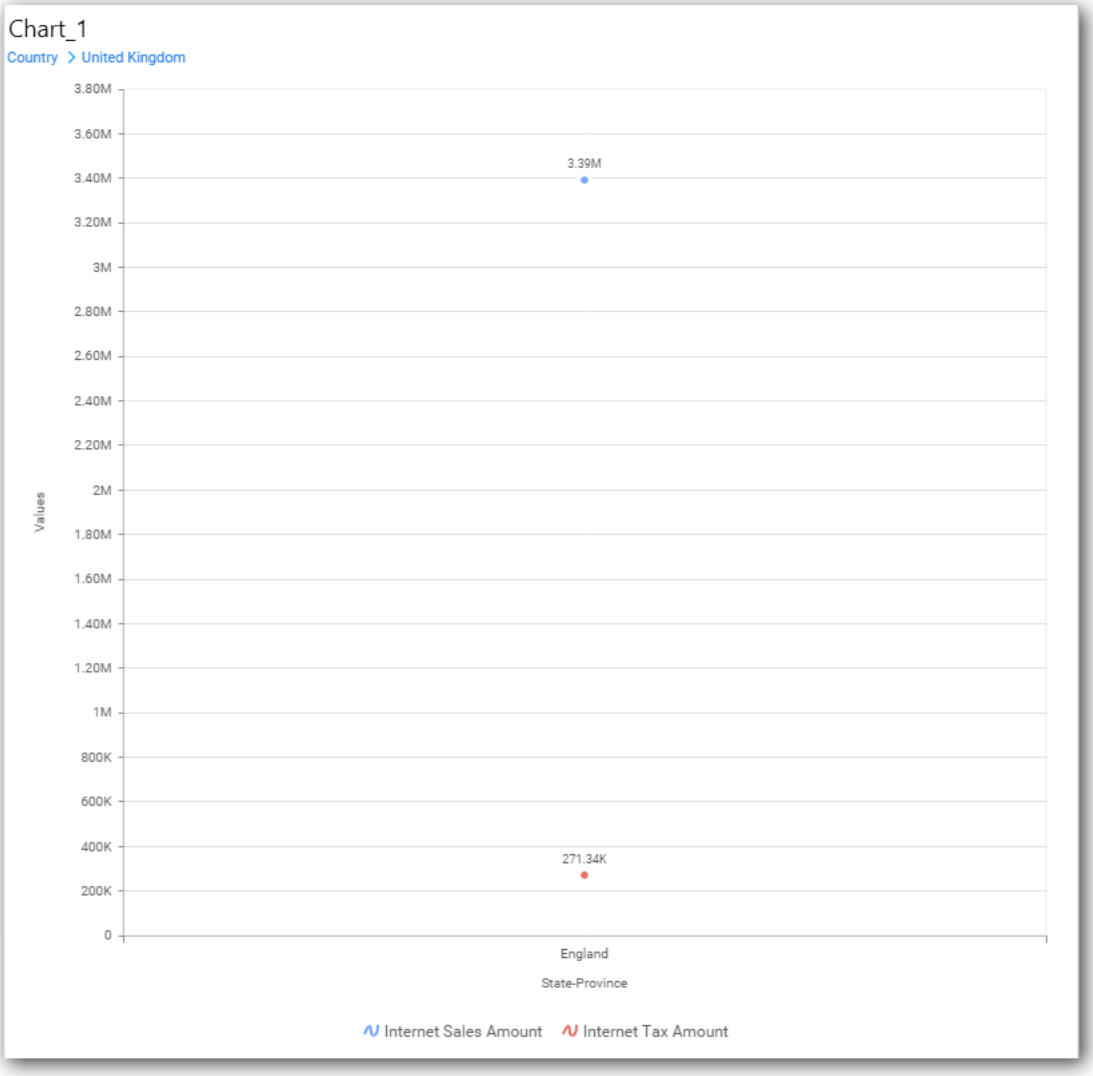


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

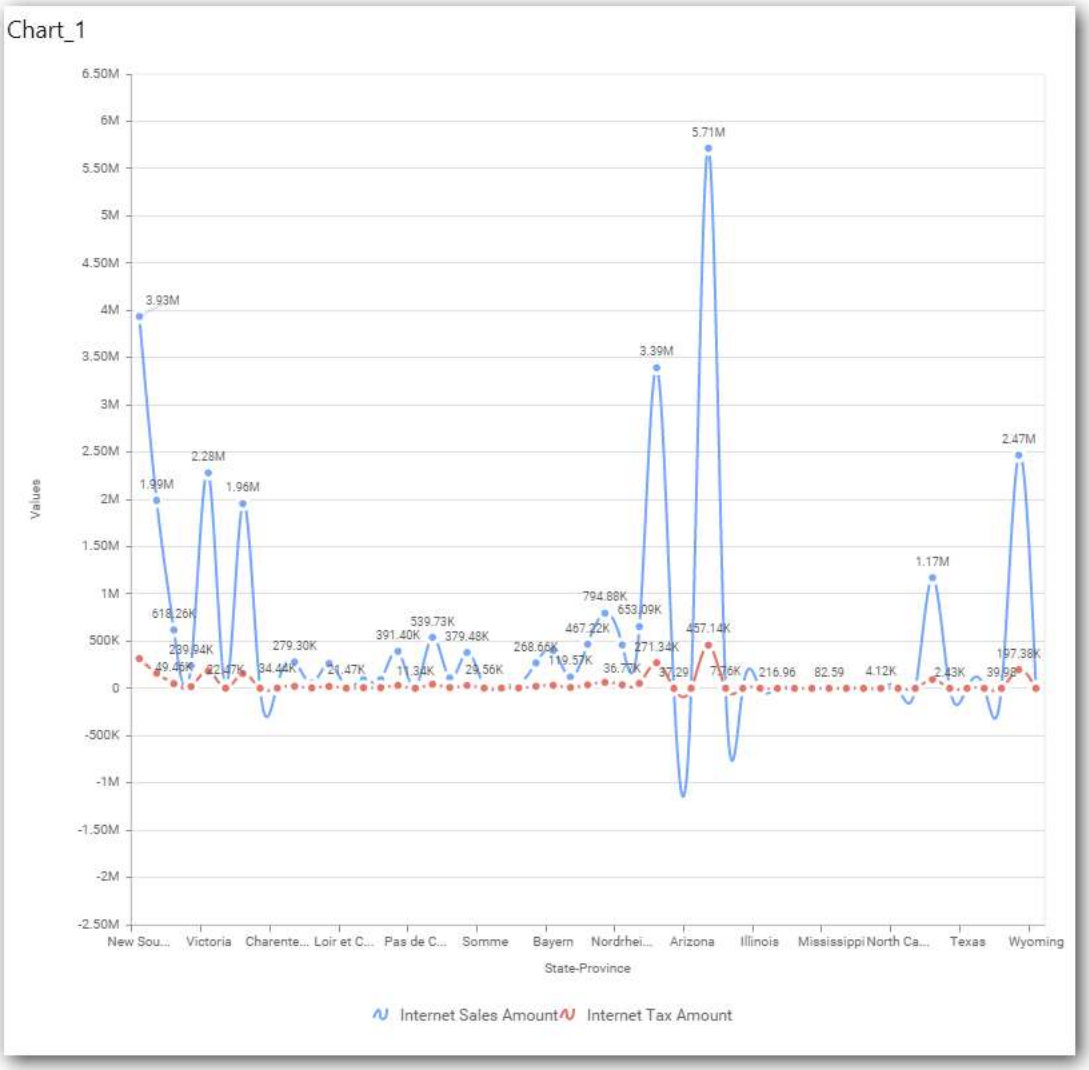
Click the respective data value marker in chart to drill into its inner level.



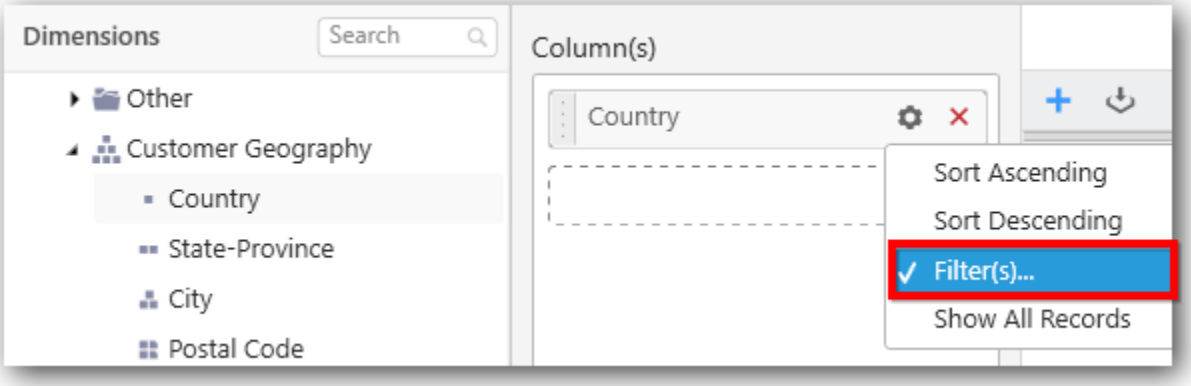
The drilled view of the chart is follows.



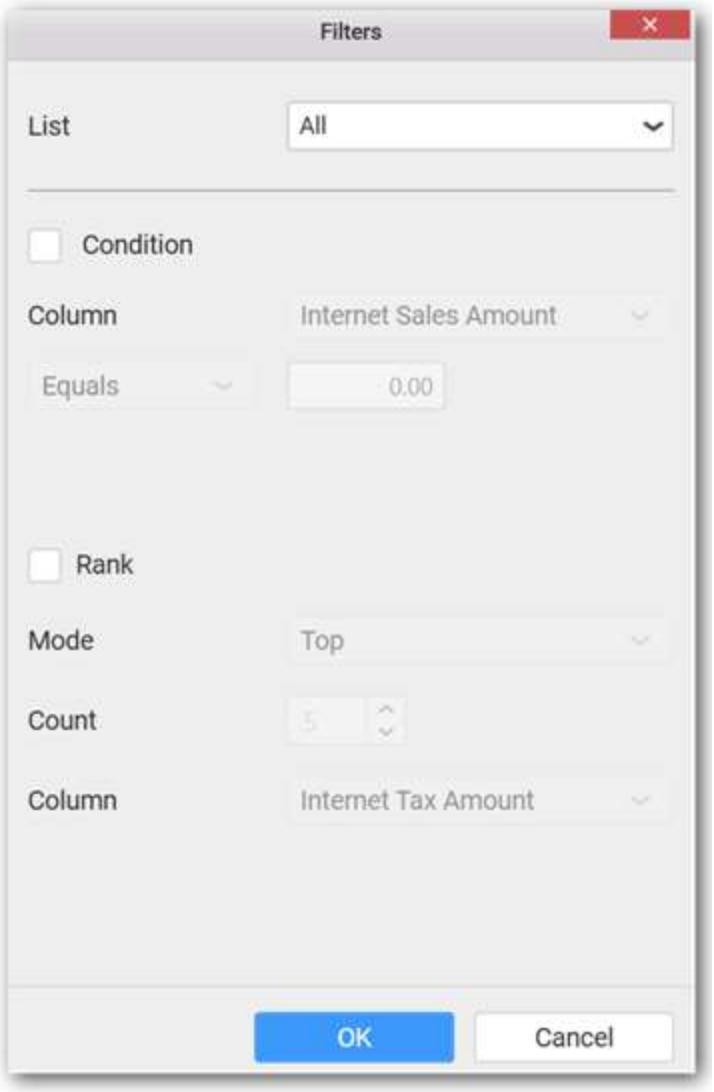
Through the breadcrumb at top, you may navigate to the outer or middle levels from your current inner level



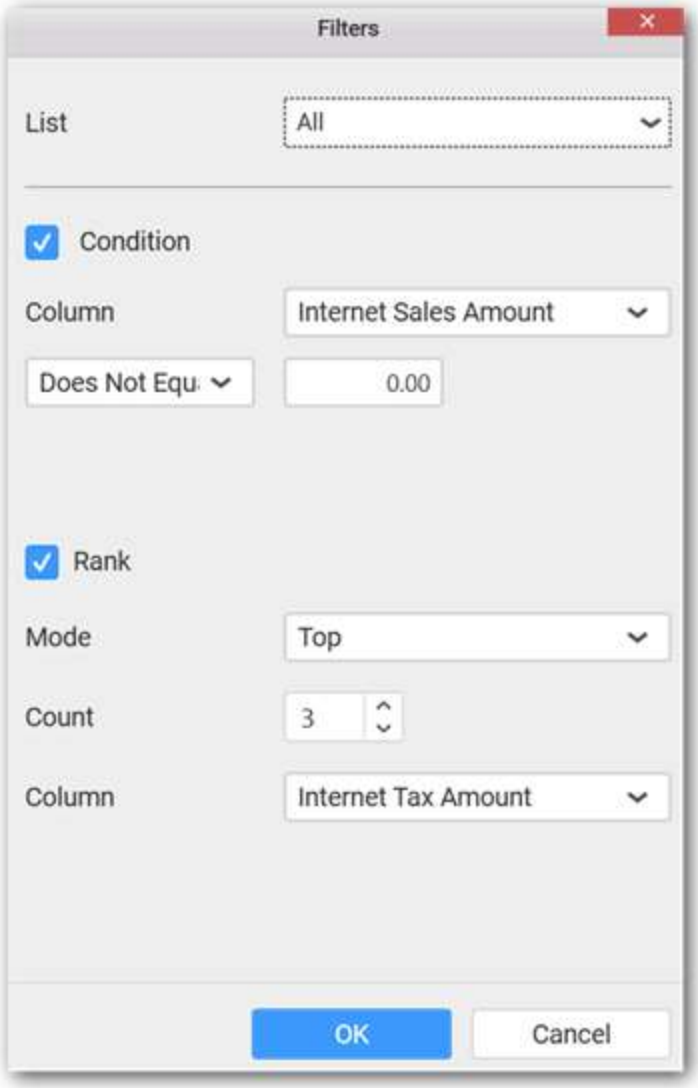
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



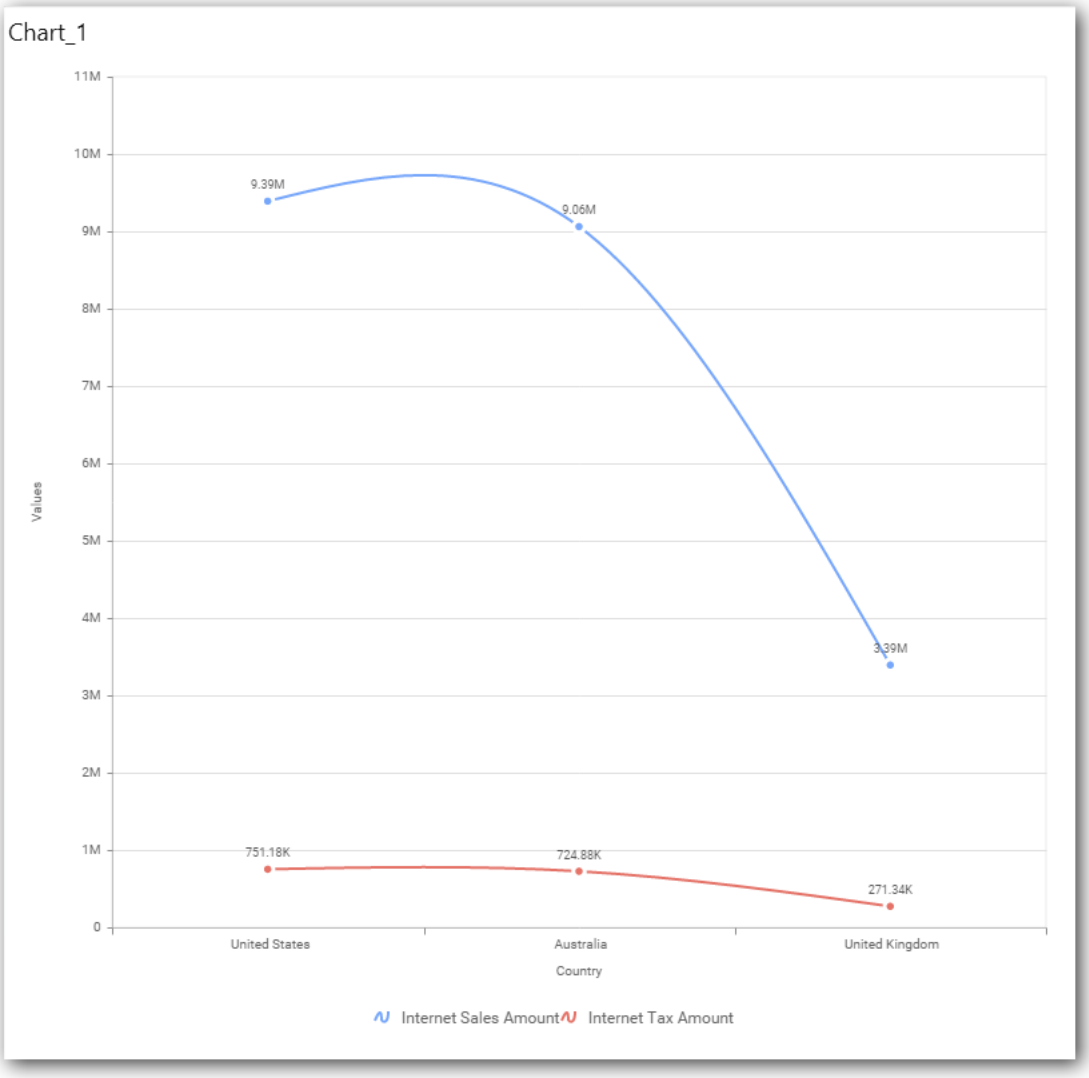
Select Filter(s)… to launch the Filters window.



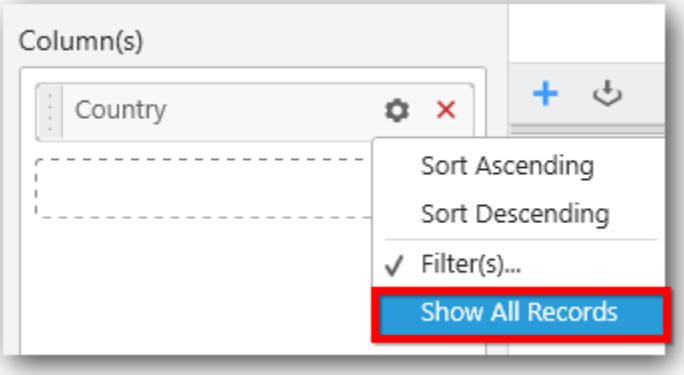
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

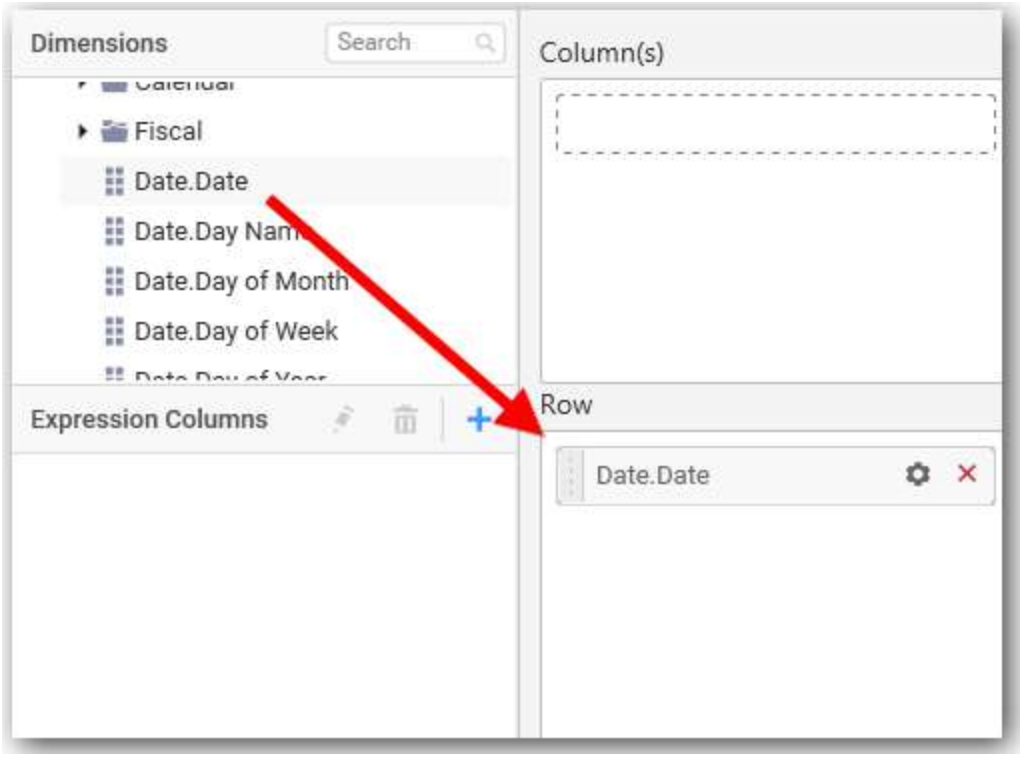


To show all records again click on Show All Records.

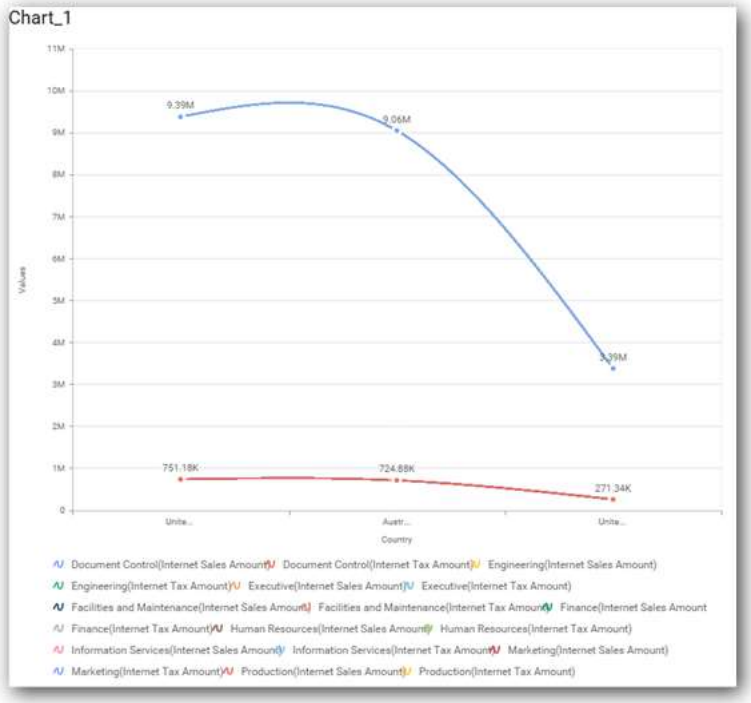


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



The chart will be rendered in series as shown in the image below.

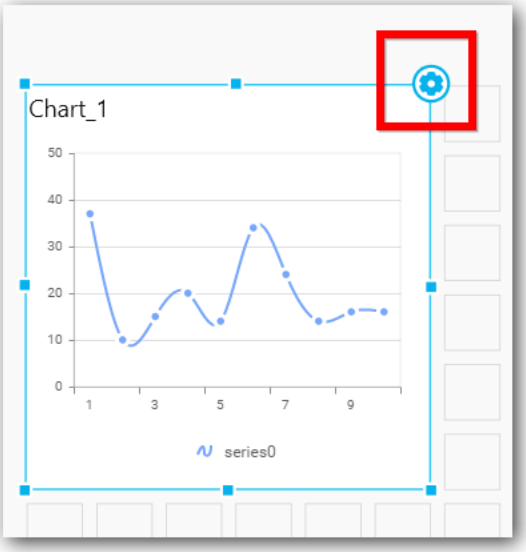


How to format Spline Chart?

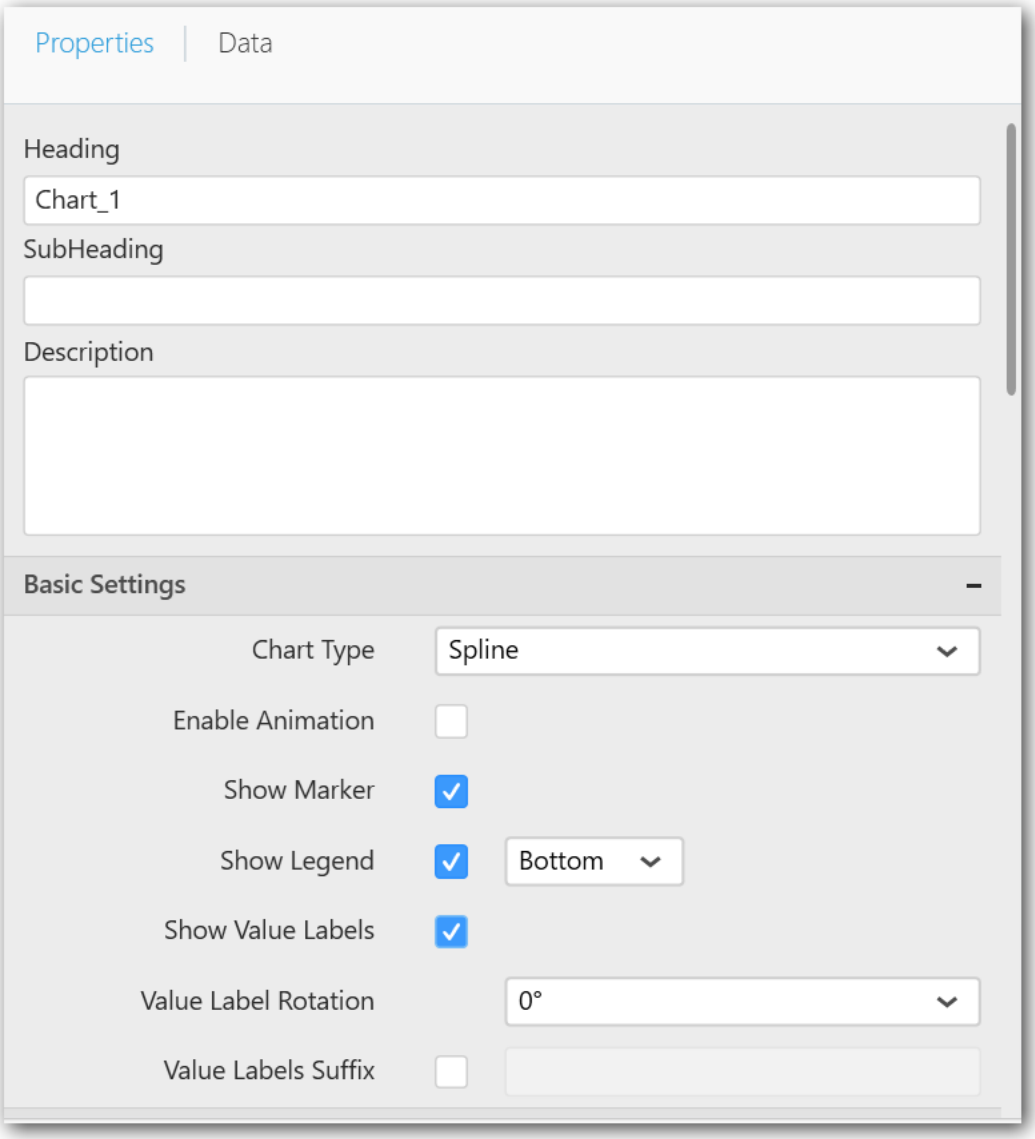
You can format the spline chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into spline chart follow the steps

1. Drag and drop the spline chart into canvas and resize it to your required size.
2. Configure the data into spline chart.
3. Focus on the spline chart and Click on Widget Settings.

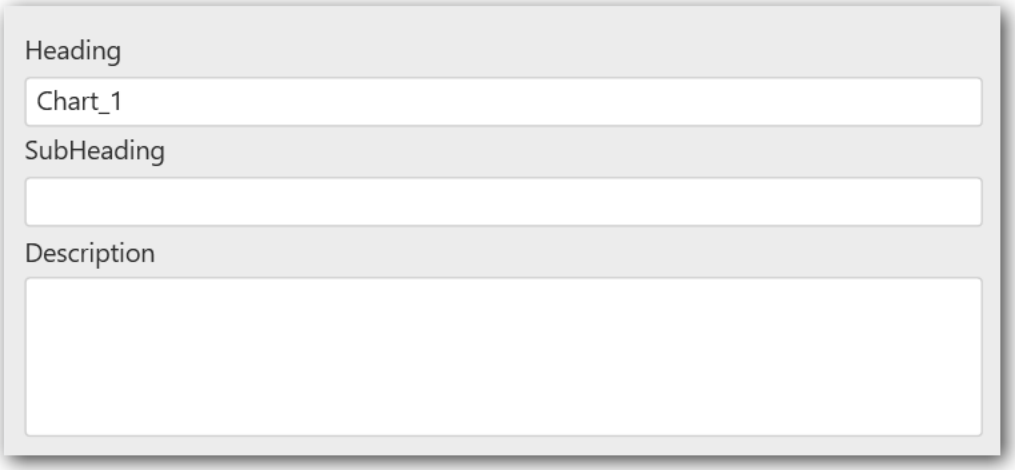


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this spline chart widget.

SubHeading

This allows you to set sub-title for this spline chart widget.

Description

This allows you to set description for this spline chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

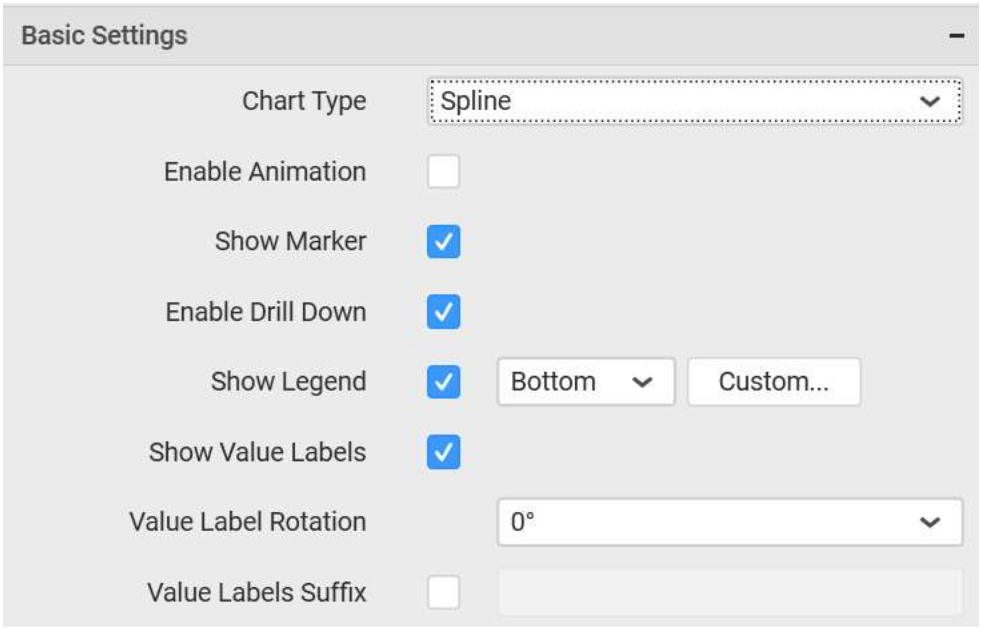


Chart Type

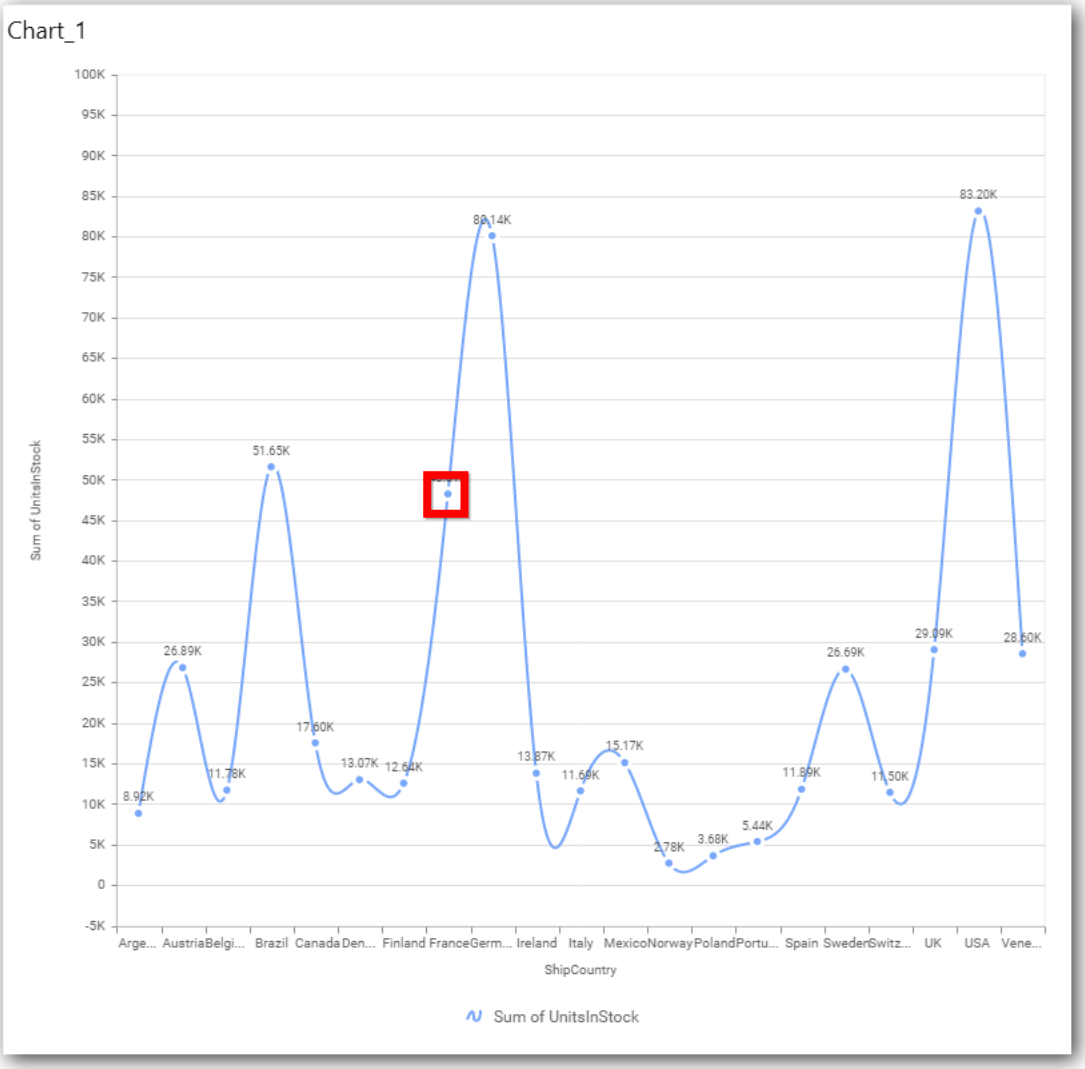
This allows you to switch the widget view from current chart type to another chart type. To selecting chart type through combo box.

Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Data Marker

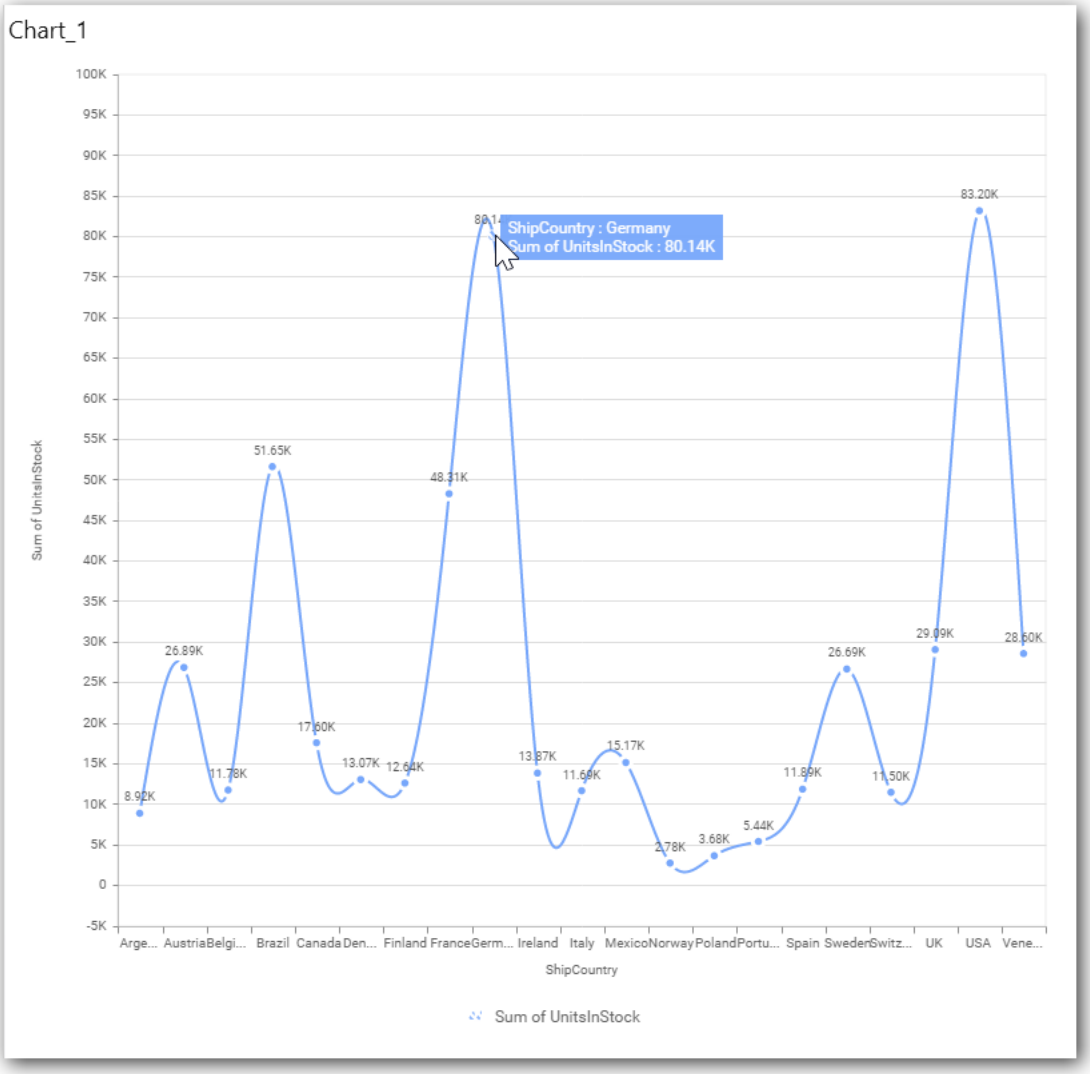
This allows you to toggle the visibility of marker from label to adorn each data point in chart series.



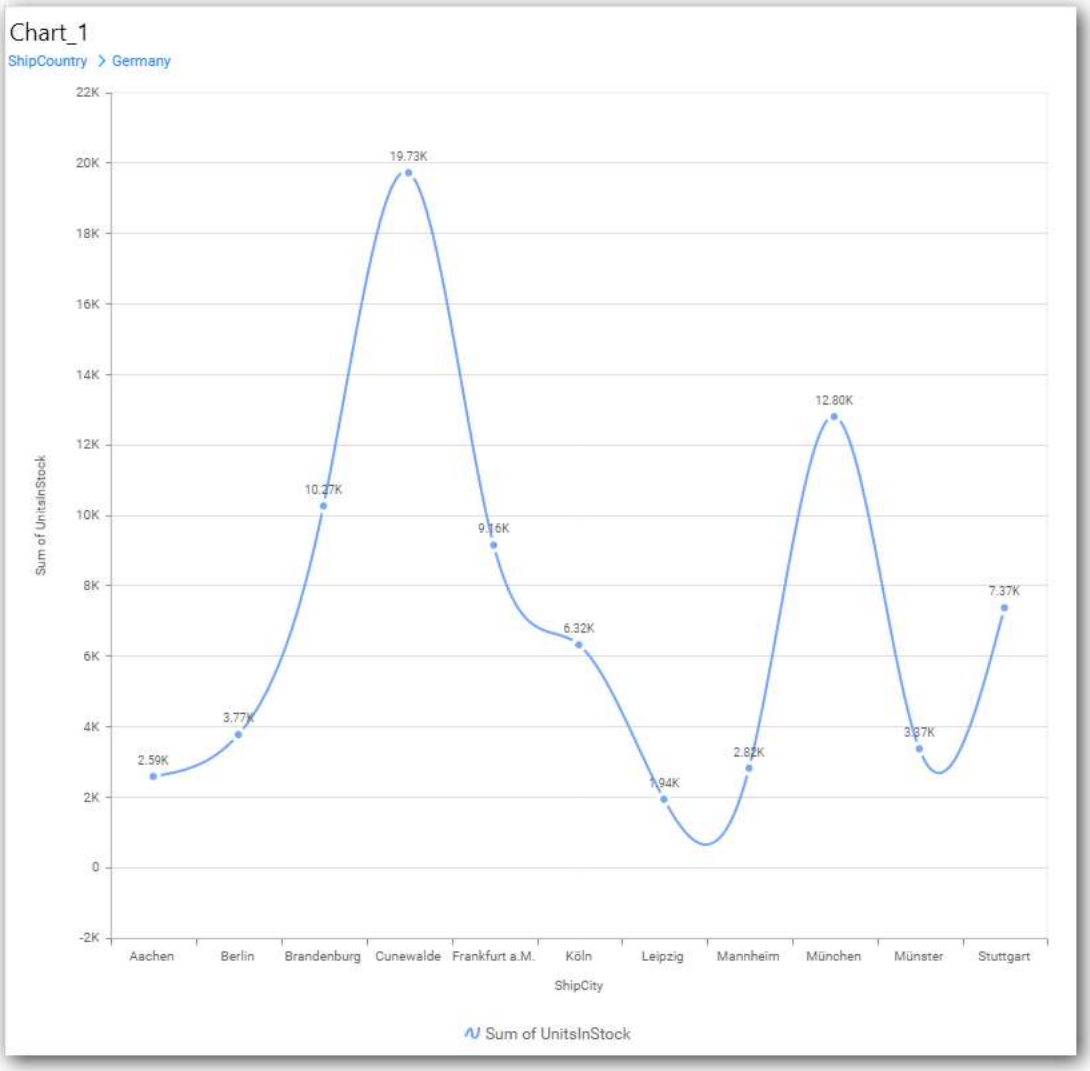
Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View

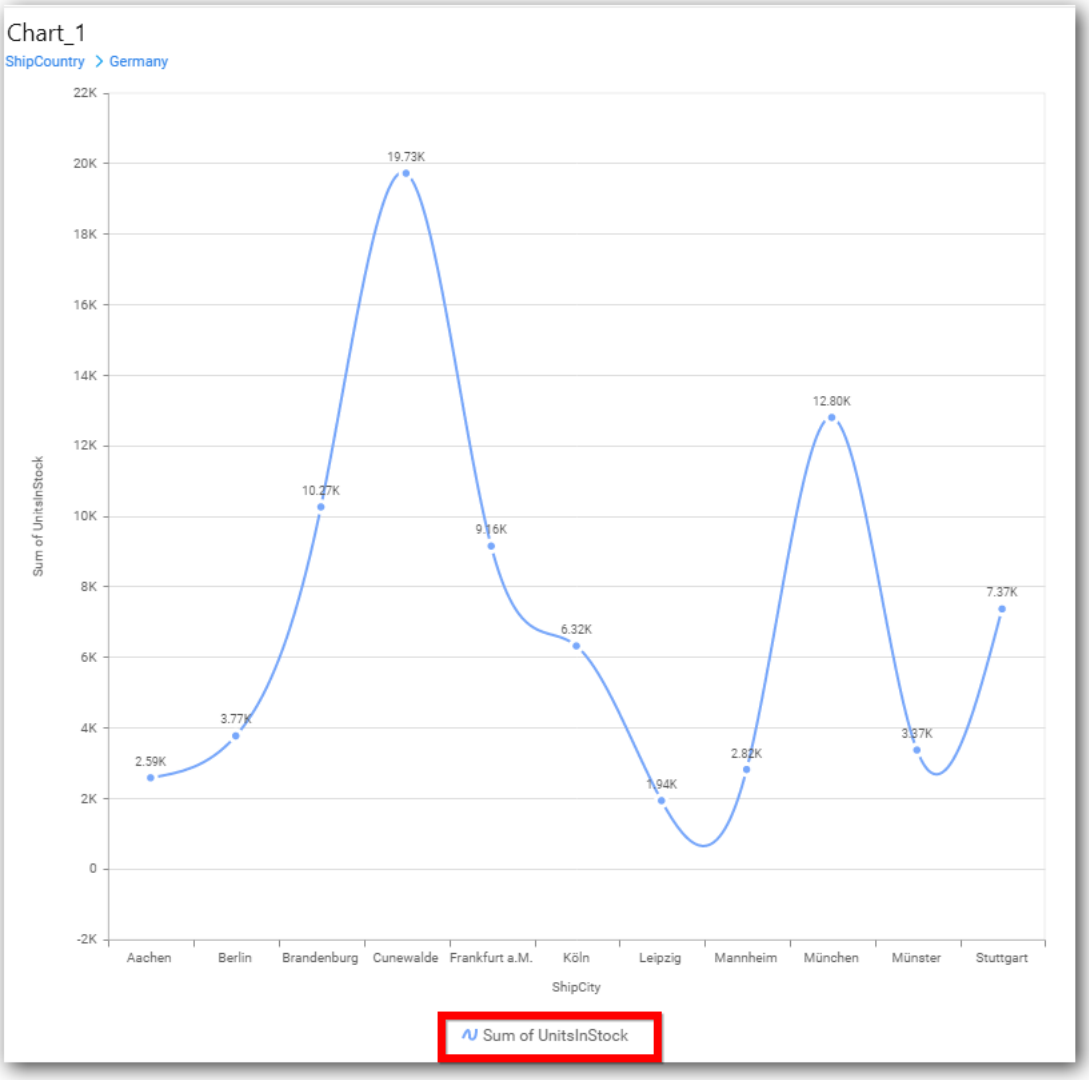


Drilled View



Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

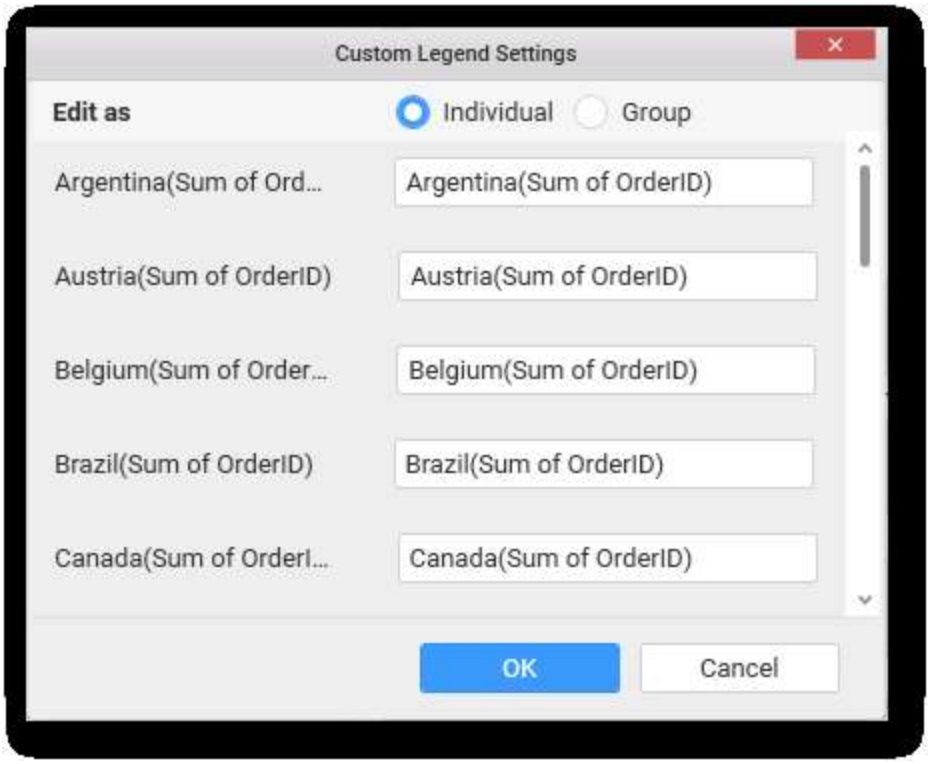
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

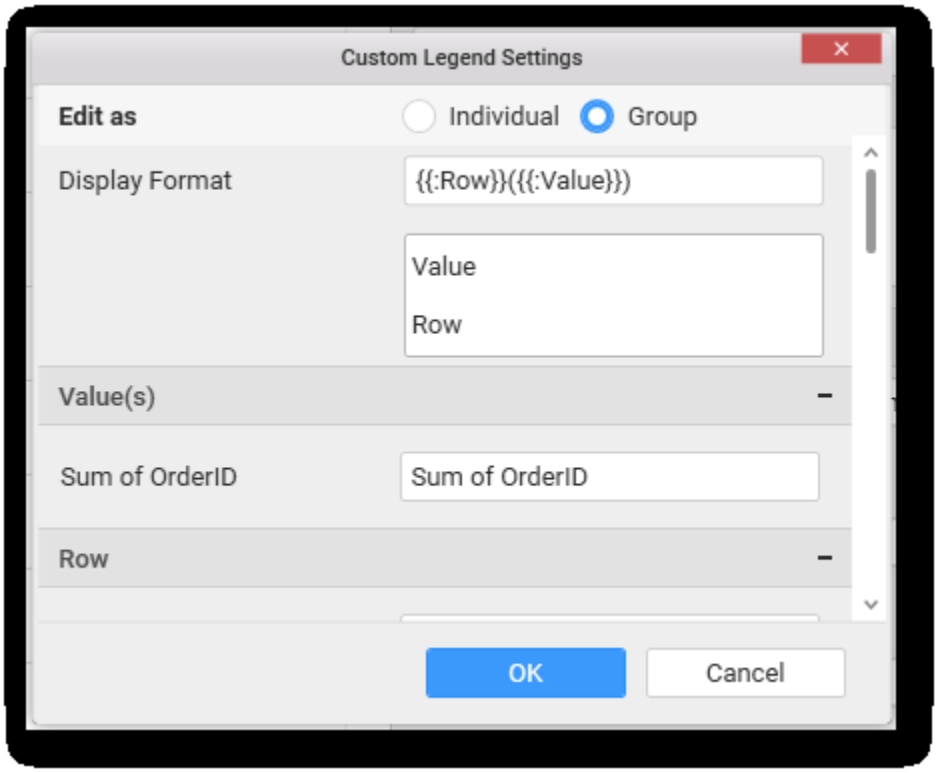
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

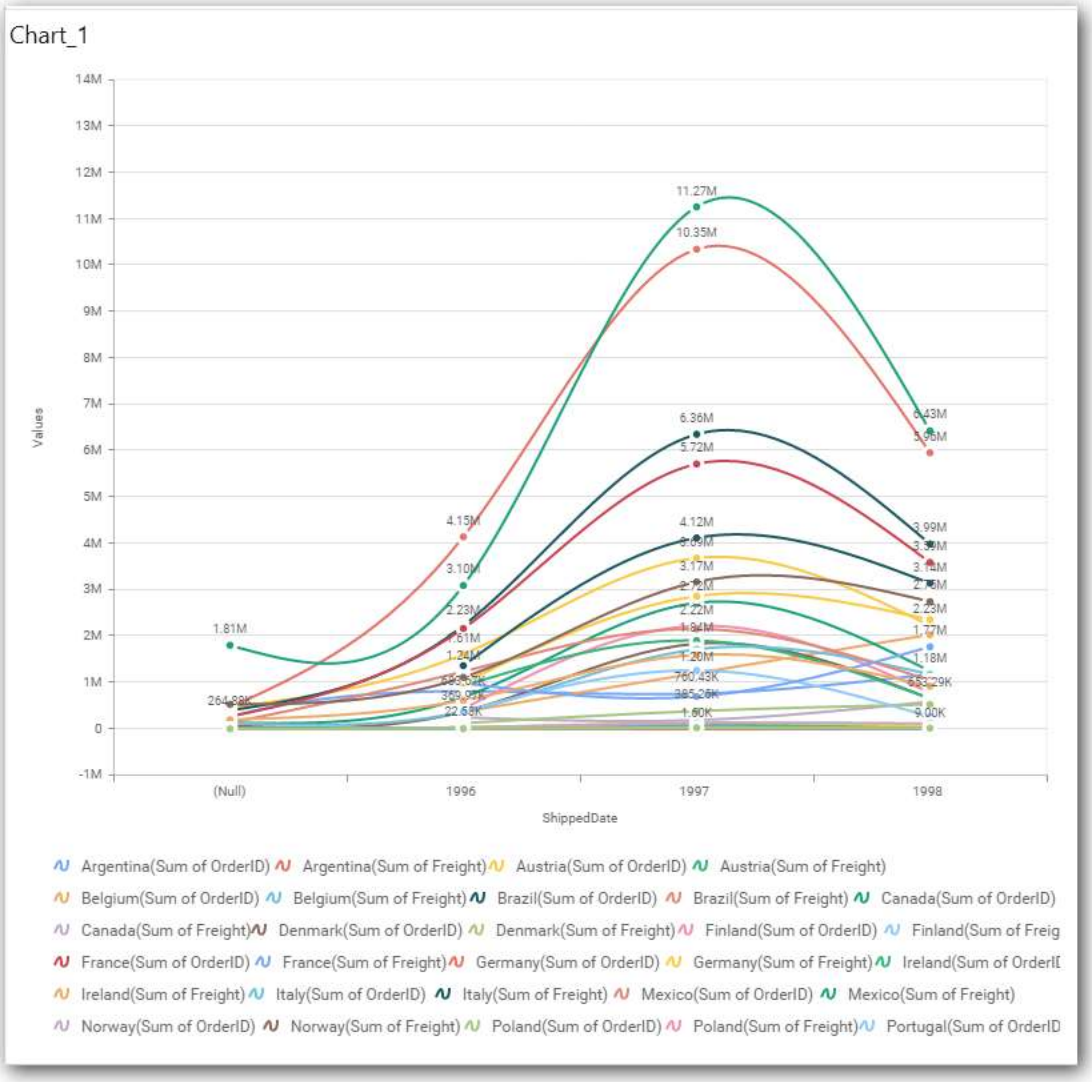


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

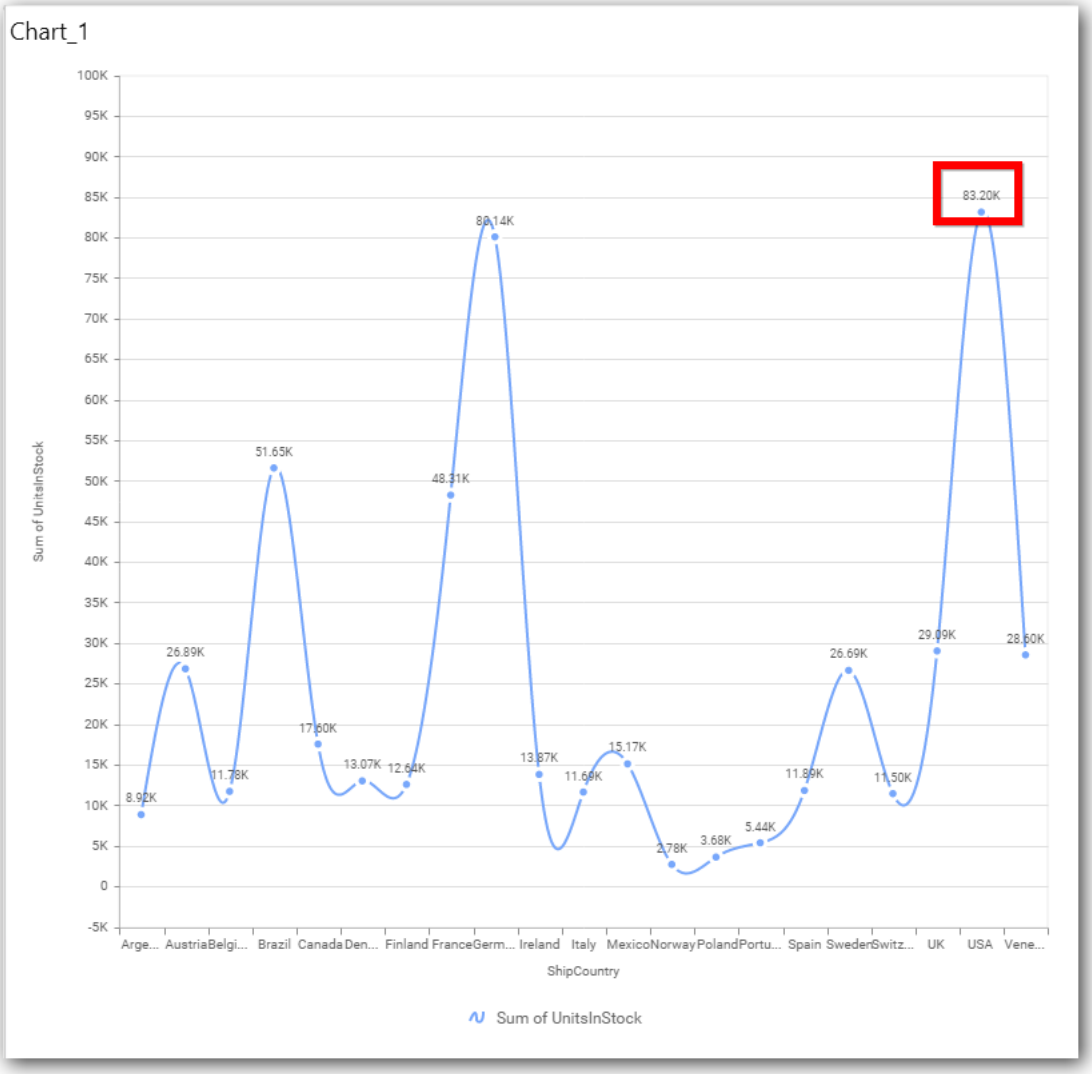


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



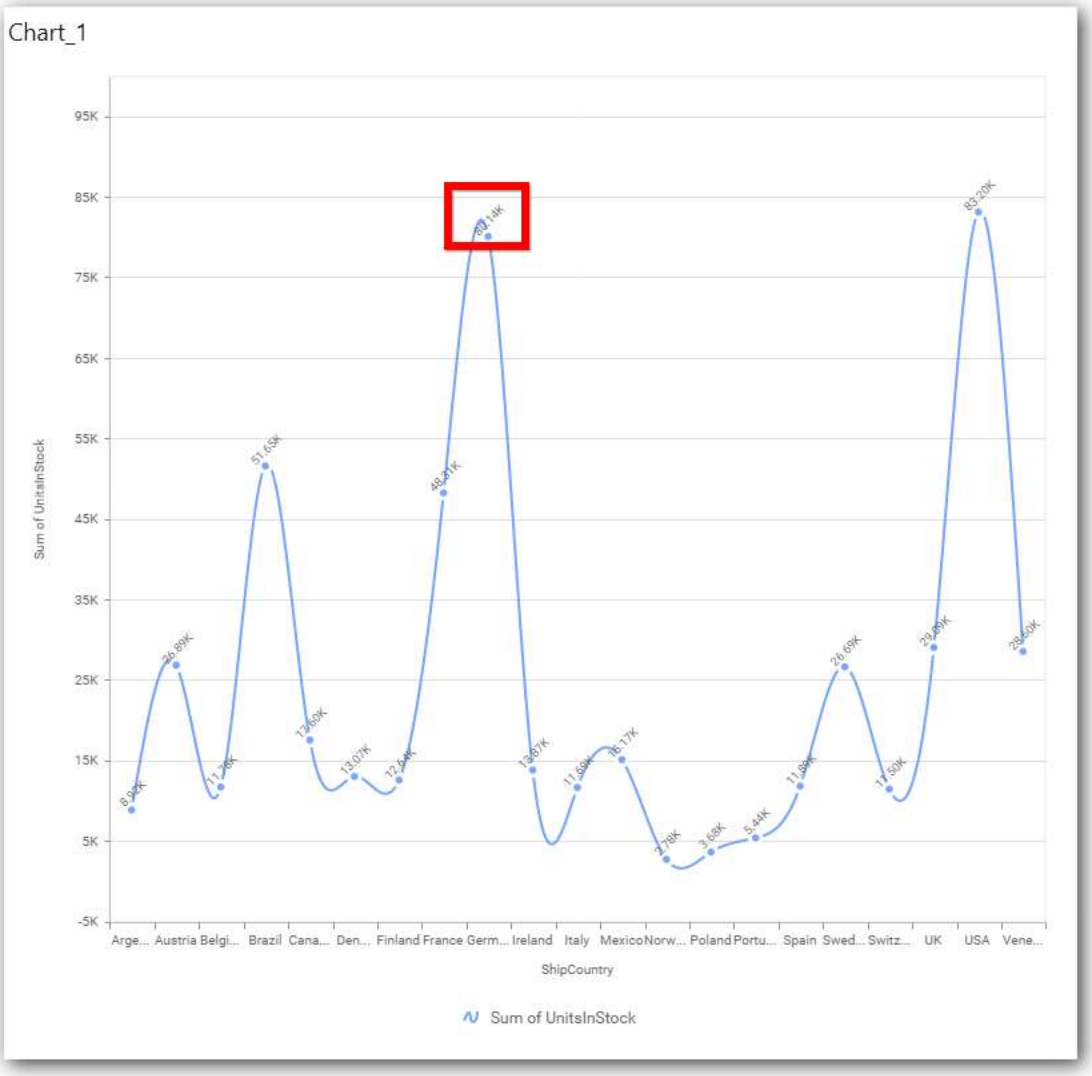
Show Value Labels

Through Show Value Labels, you can toggle the visibility of value labels.



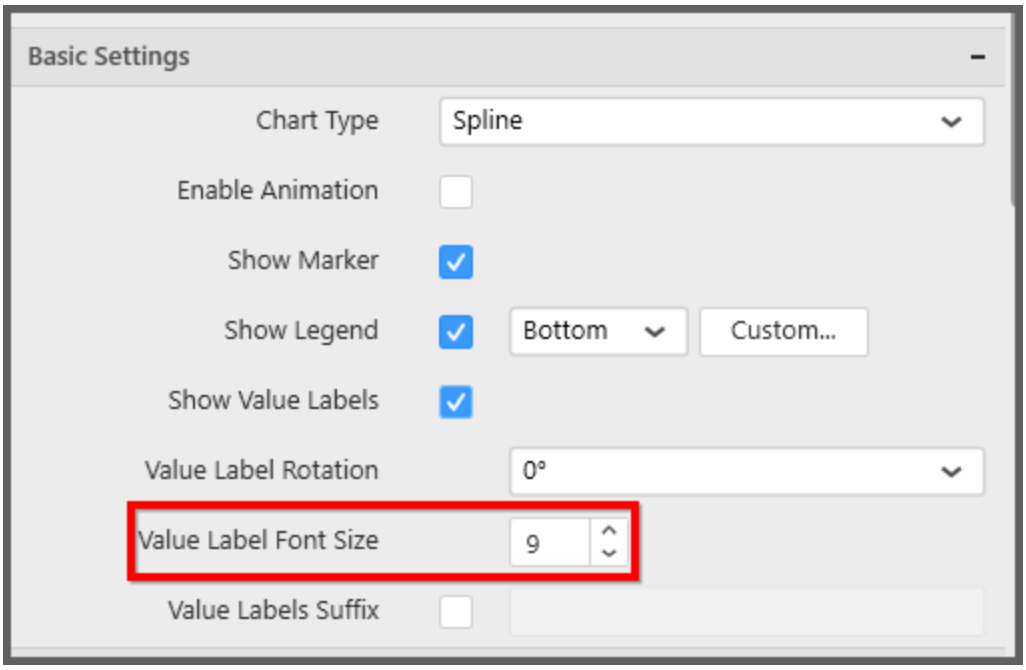
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



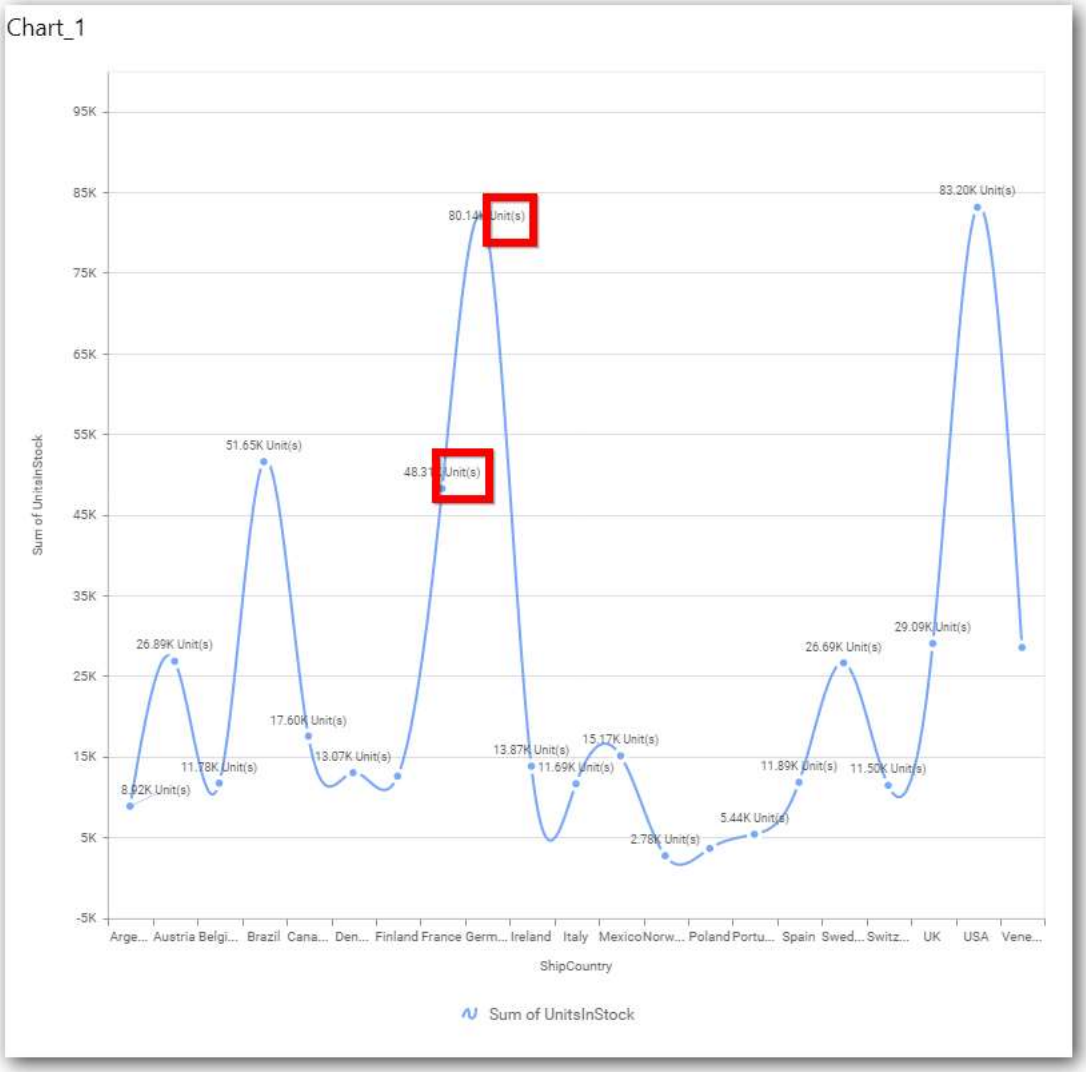
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

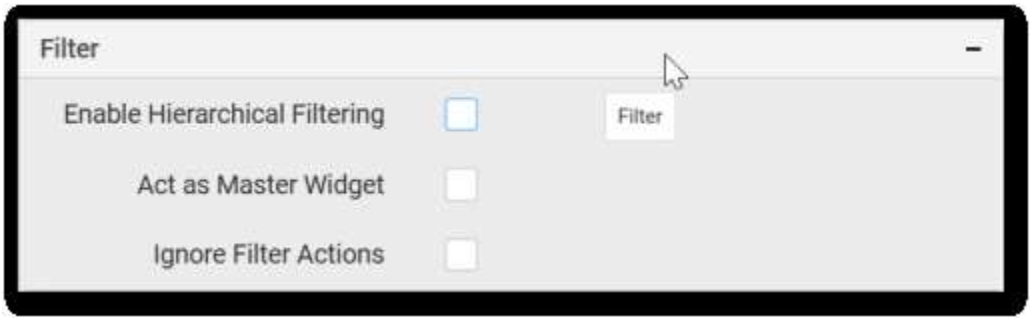


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

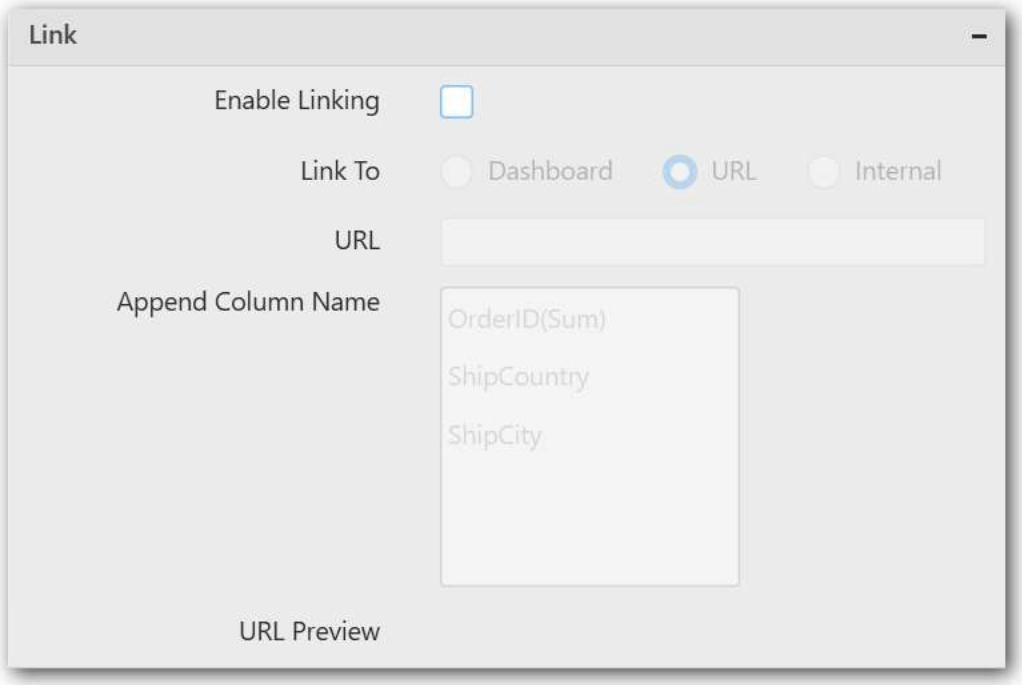
Act as Master Widget

This allows you to define this spline chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

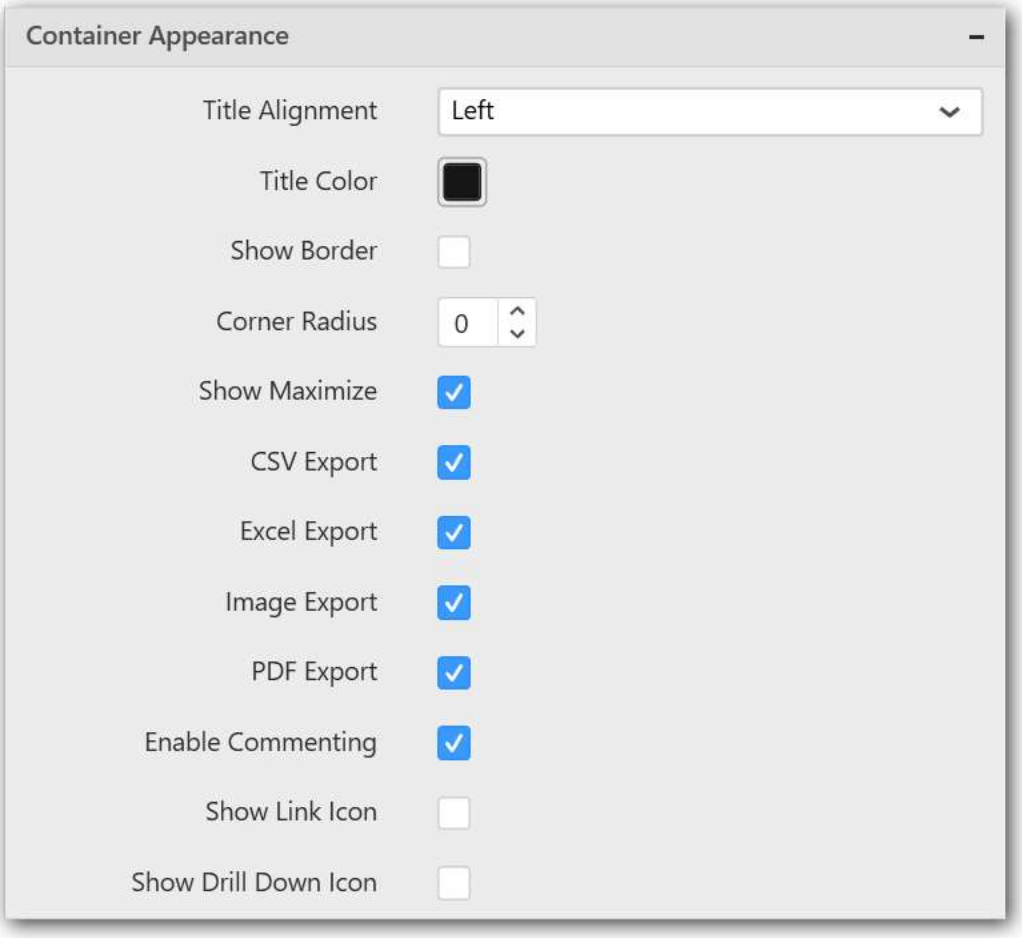
This allows you to define this spline chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this spline chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this spline chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer .

Excel Export

This allows you to enable/disable the Excel export option for this spline chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer .

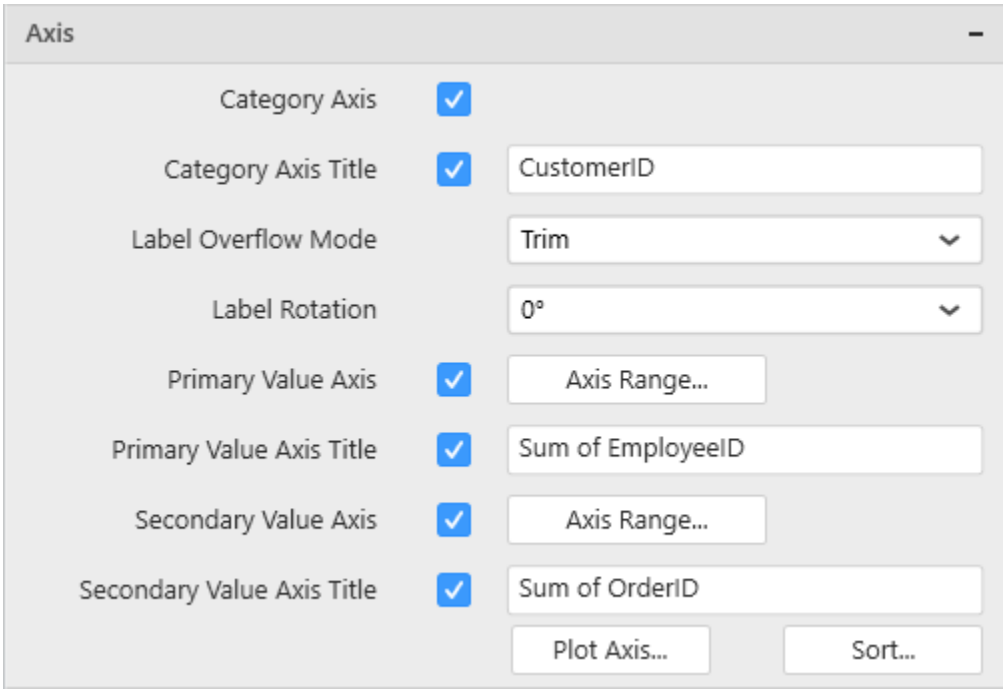
Image Export

This allows you to enable/disable the image export option for this spline chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Axis Settings



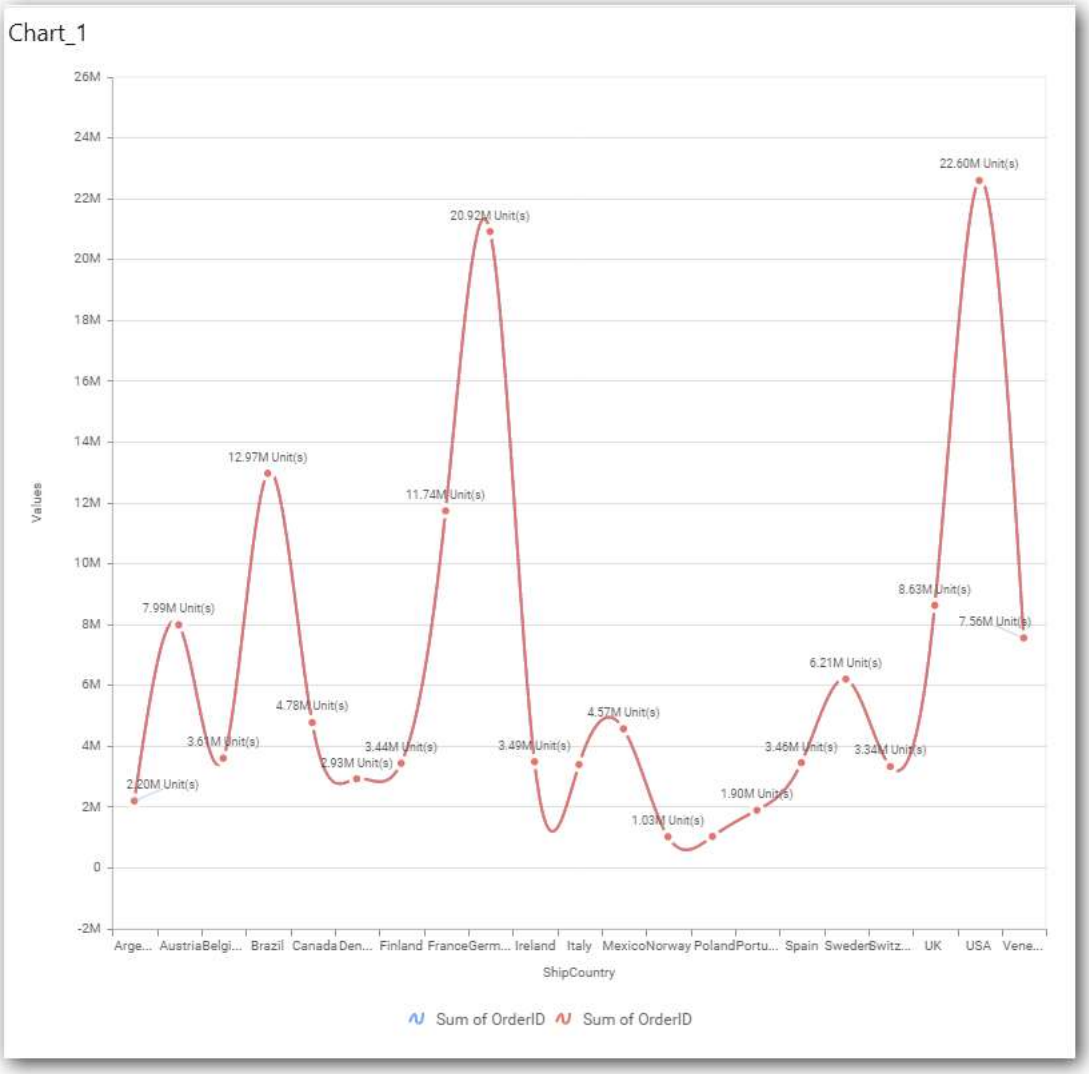
This section allows you to customize the axis settings in chart.

Category Axis

This allows to enable/edit the option of Category Axis. It will reflect in chart area x-axis name.

Category Axis Title

This allows you to toggle the visibility of Category axis title.

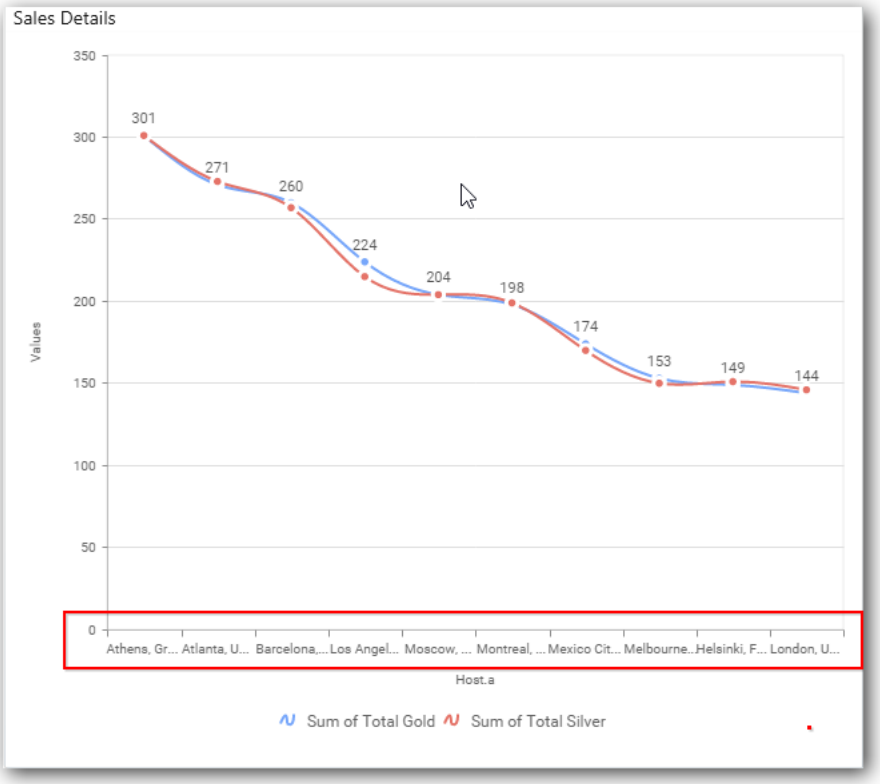


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

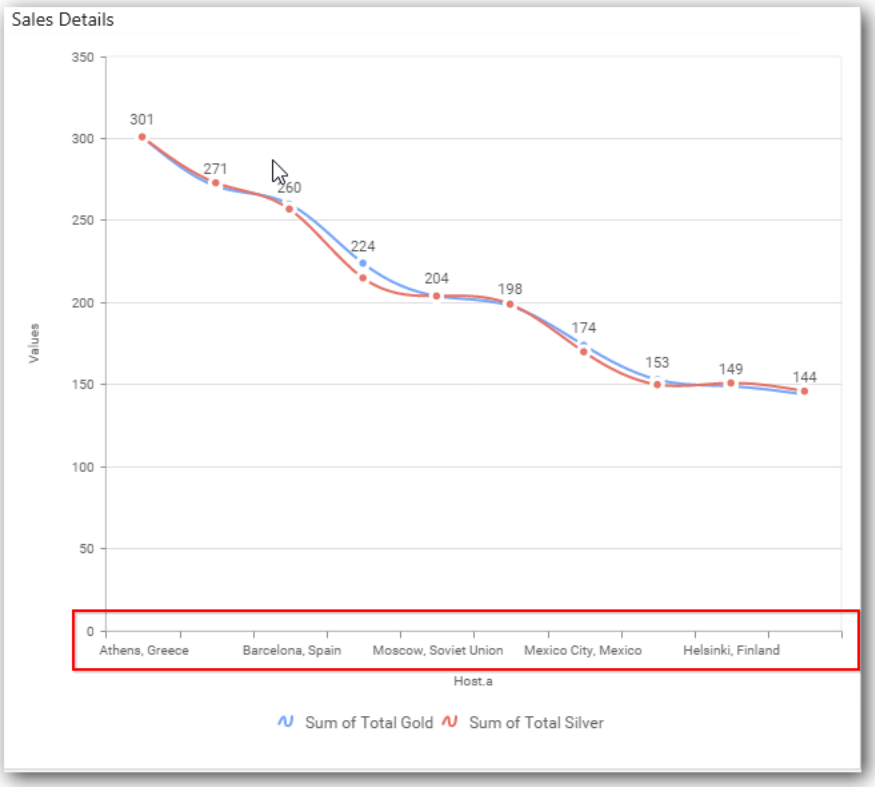
Trim

This option trims the end of overlapping label in the axis.



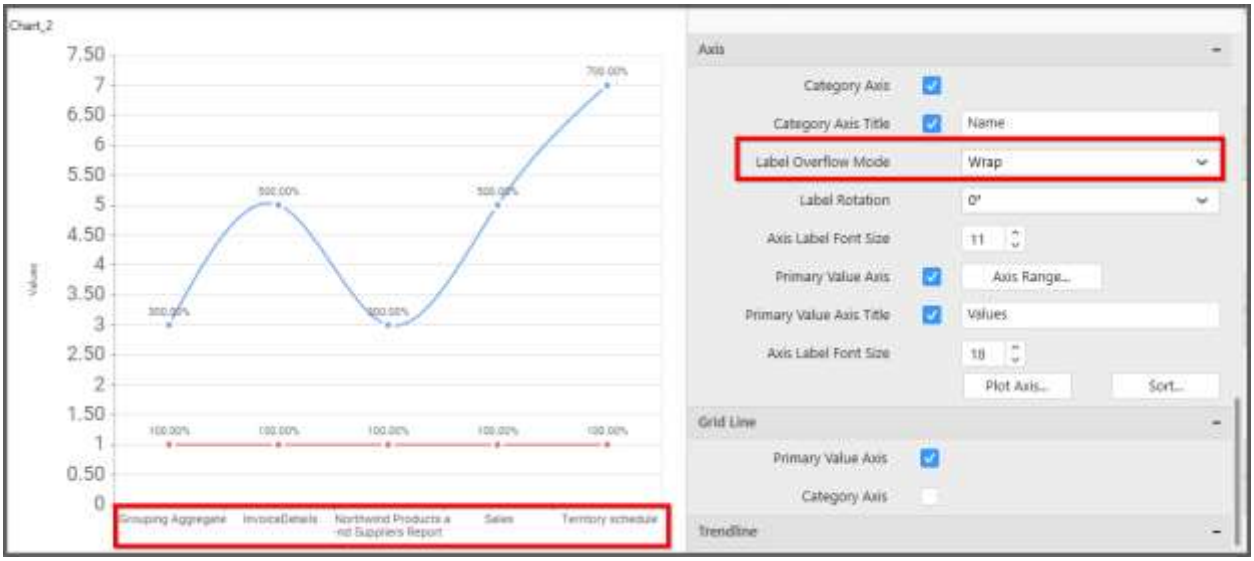
Hide

This option hides the overlapping label in the axis.



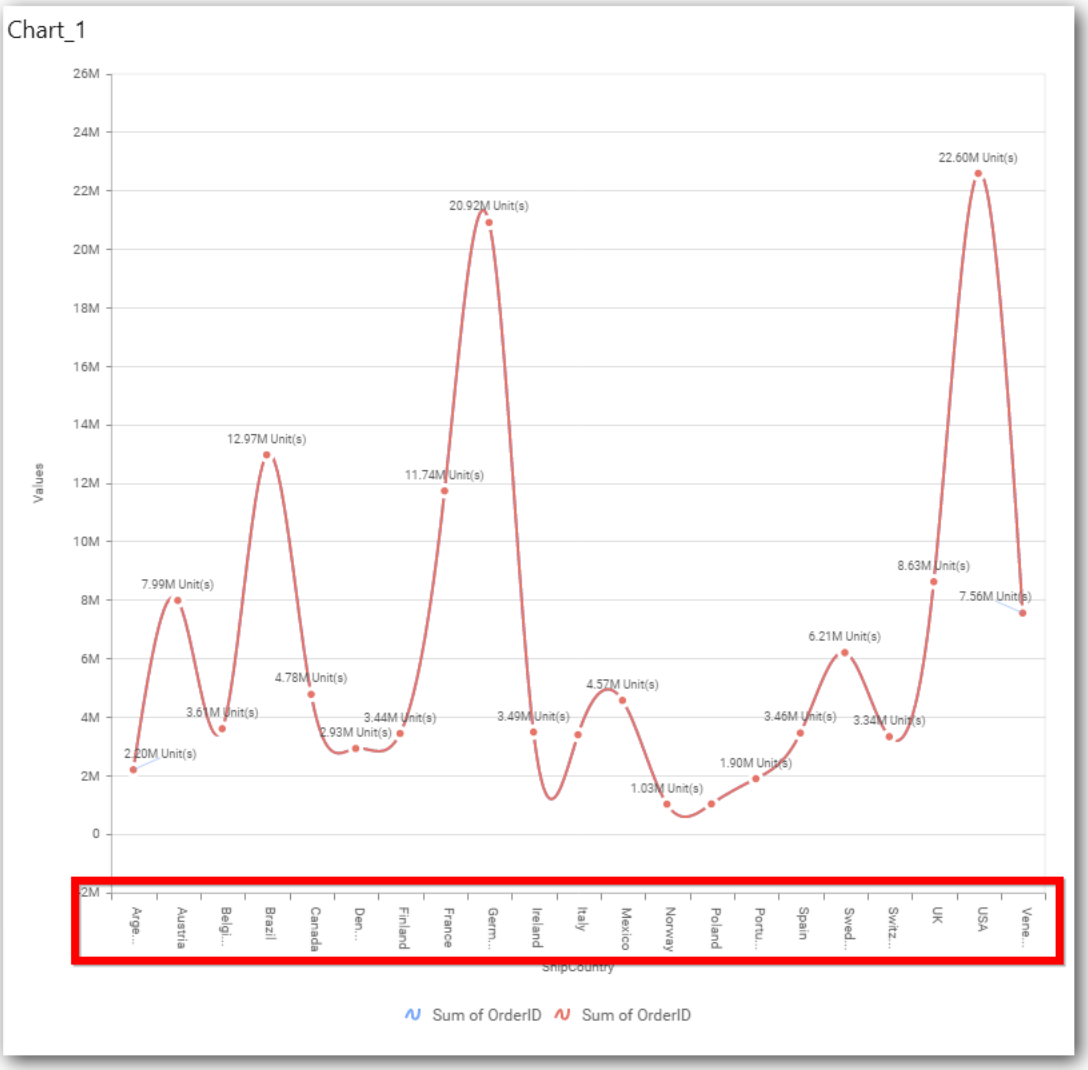
Wrap

This option wraps the lengthy label text in the axis.



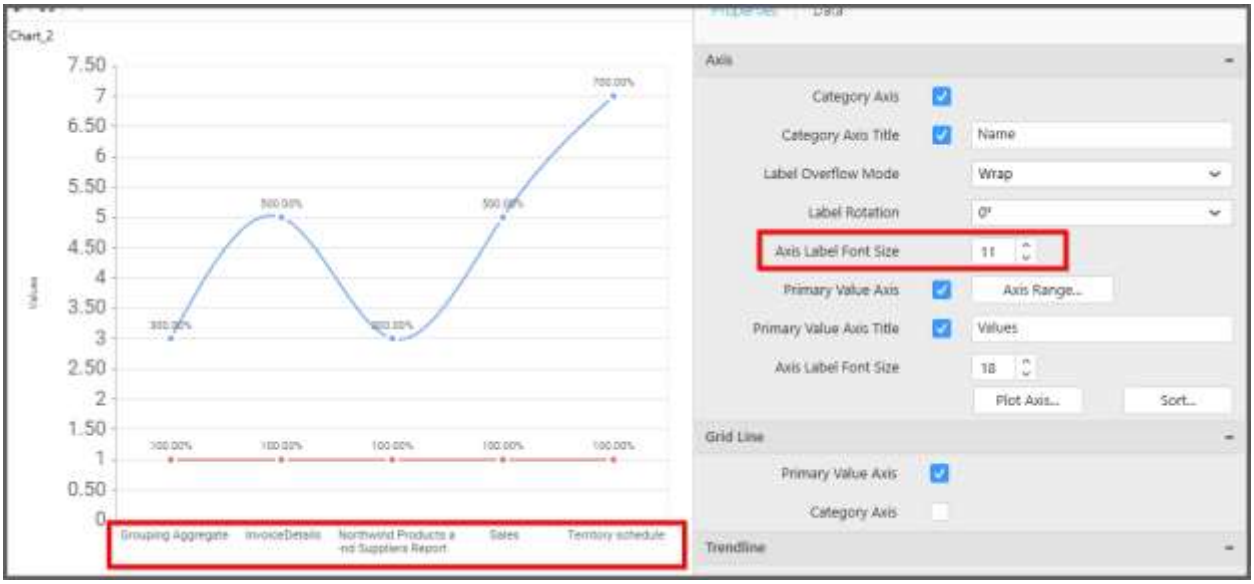
Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



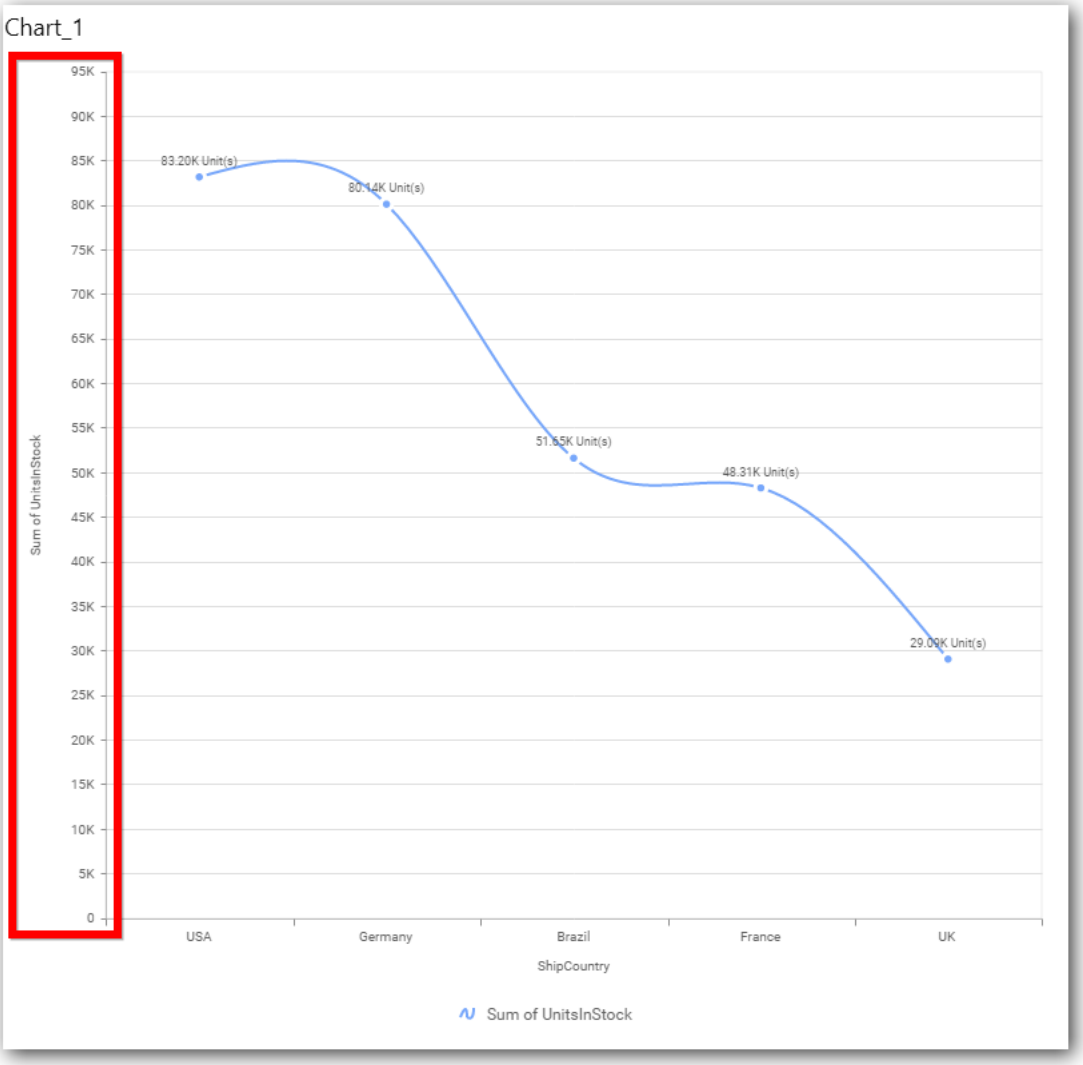
Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.



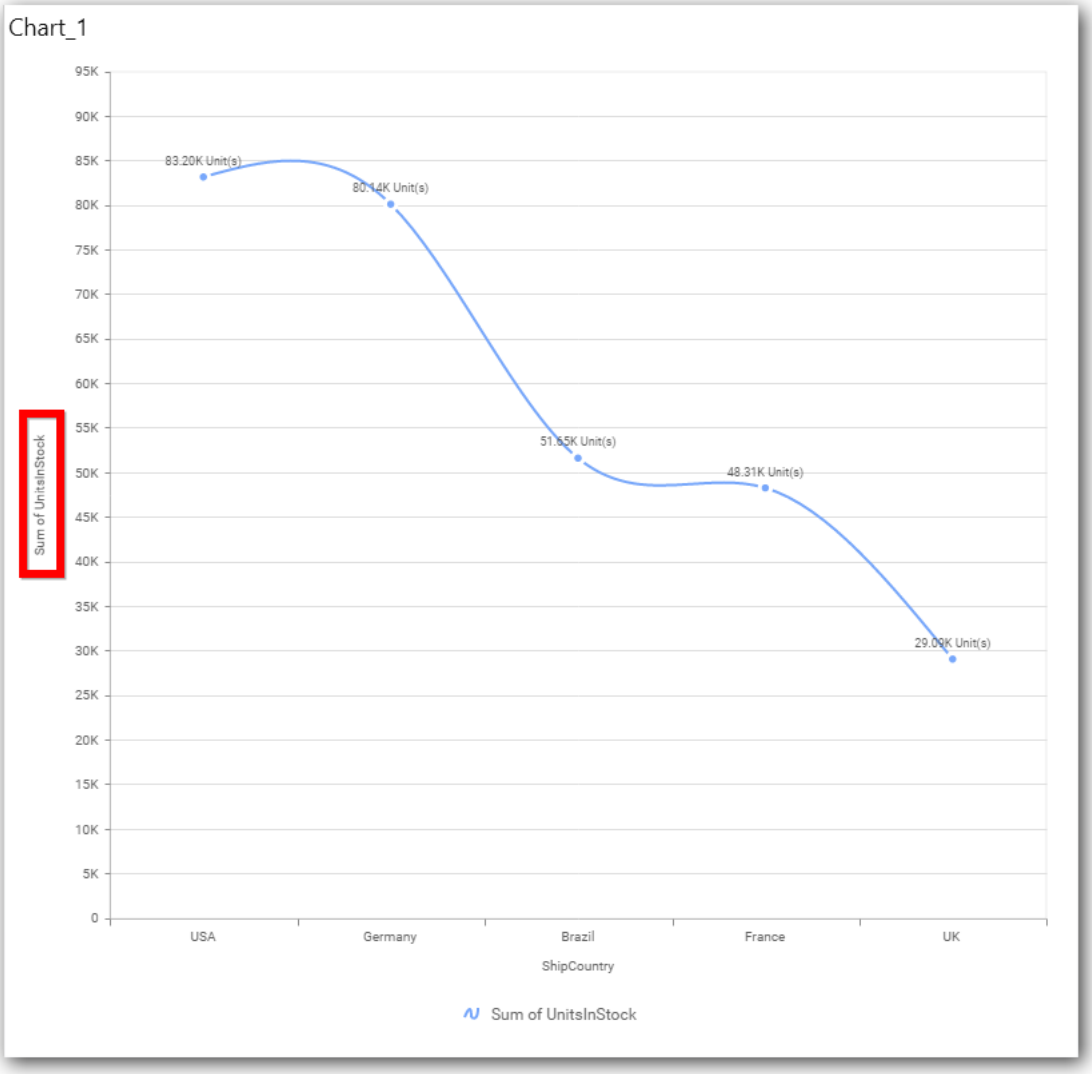
Primary Value Axis

This allows you to enable/edit the Primary Value Axis title. It will reflect in chart area y-axis name.



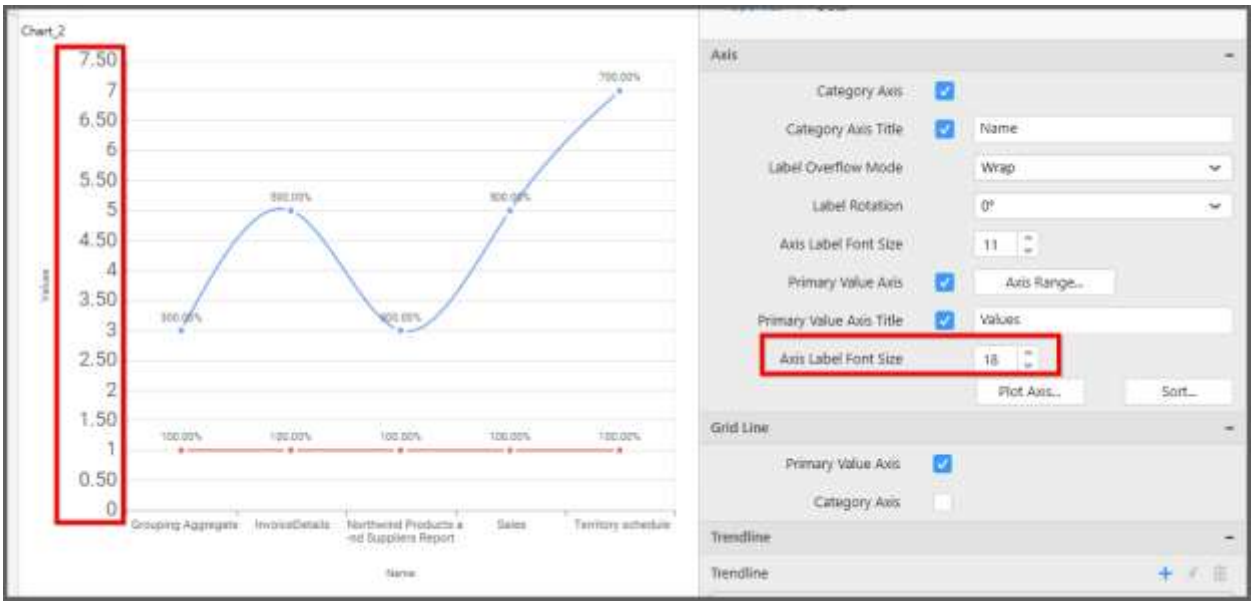
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

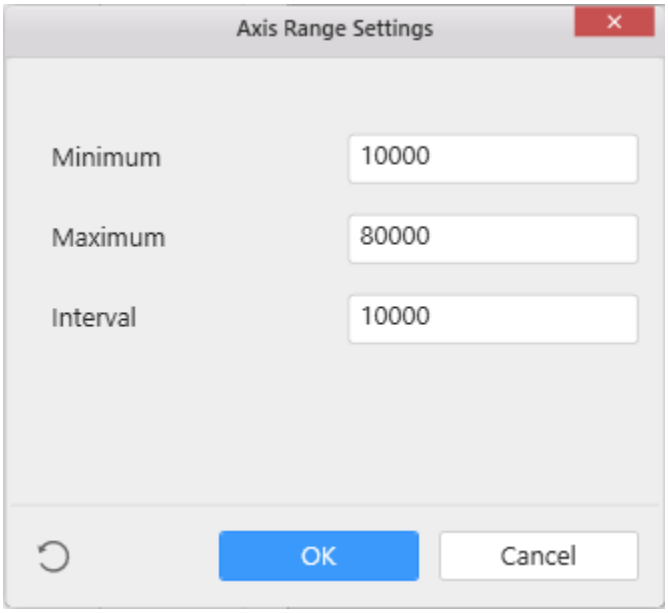


Primary Value Axis Range

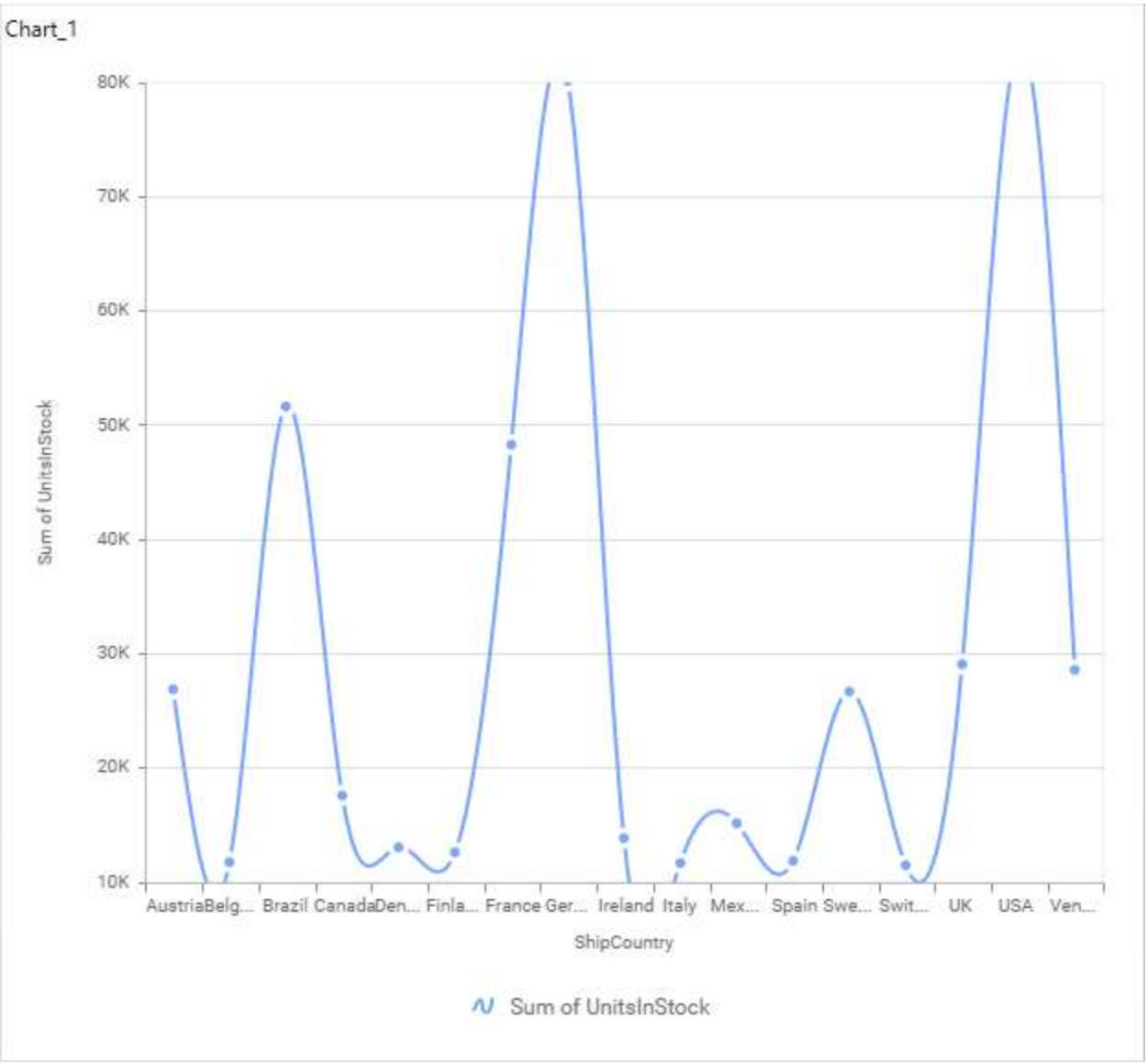
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.

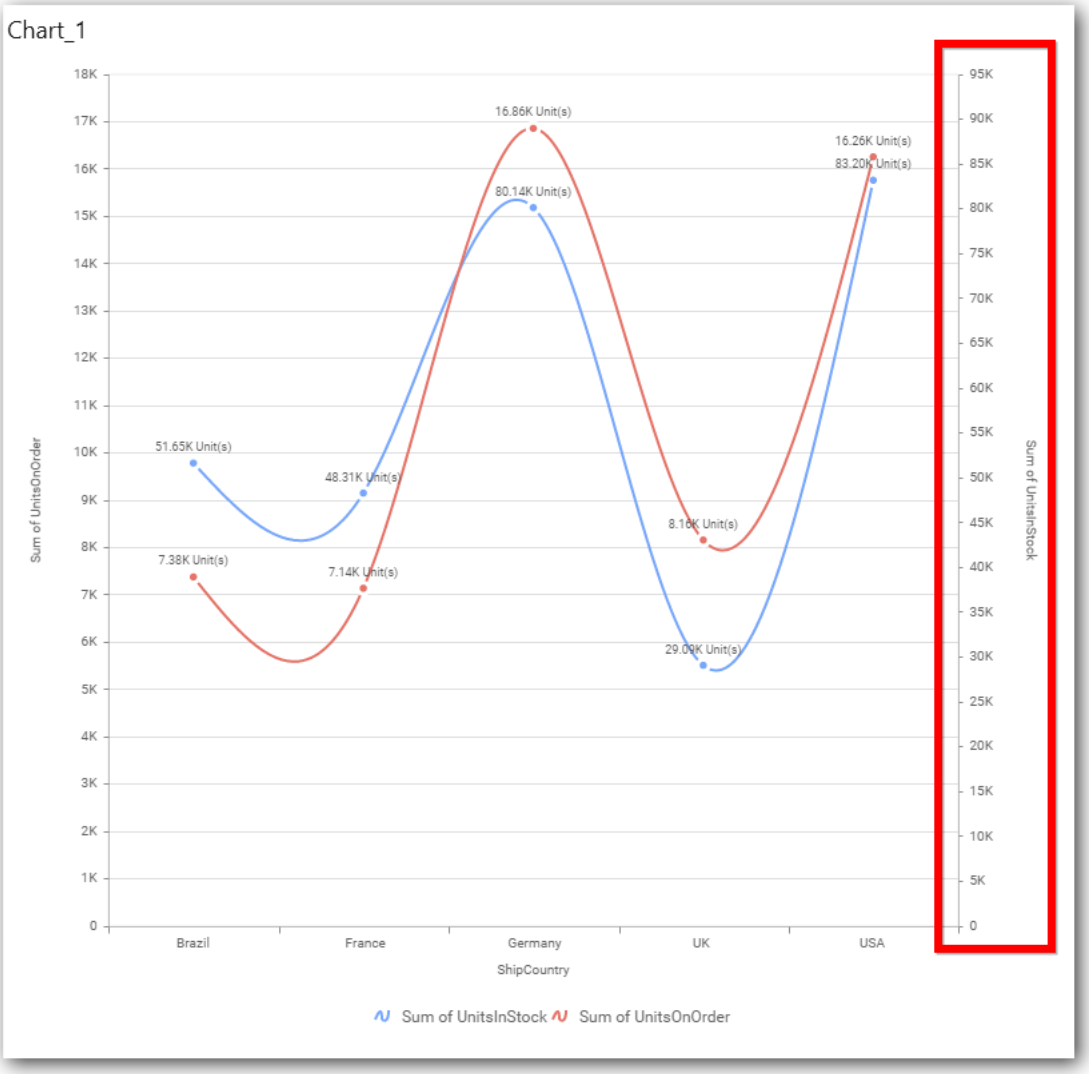


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



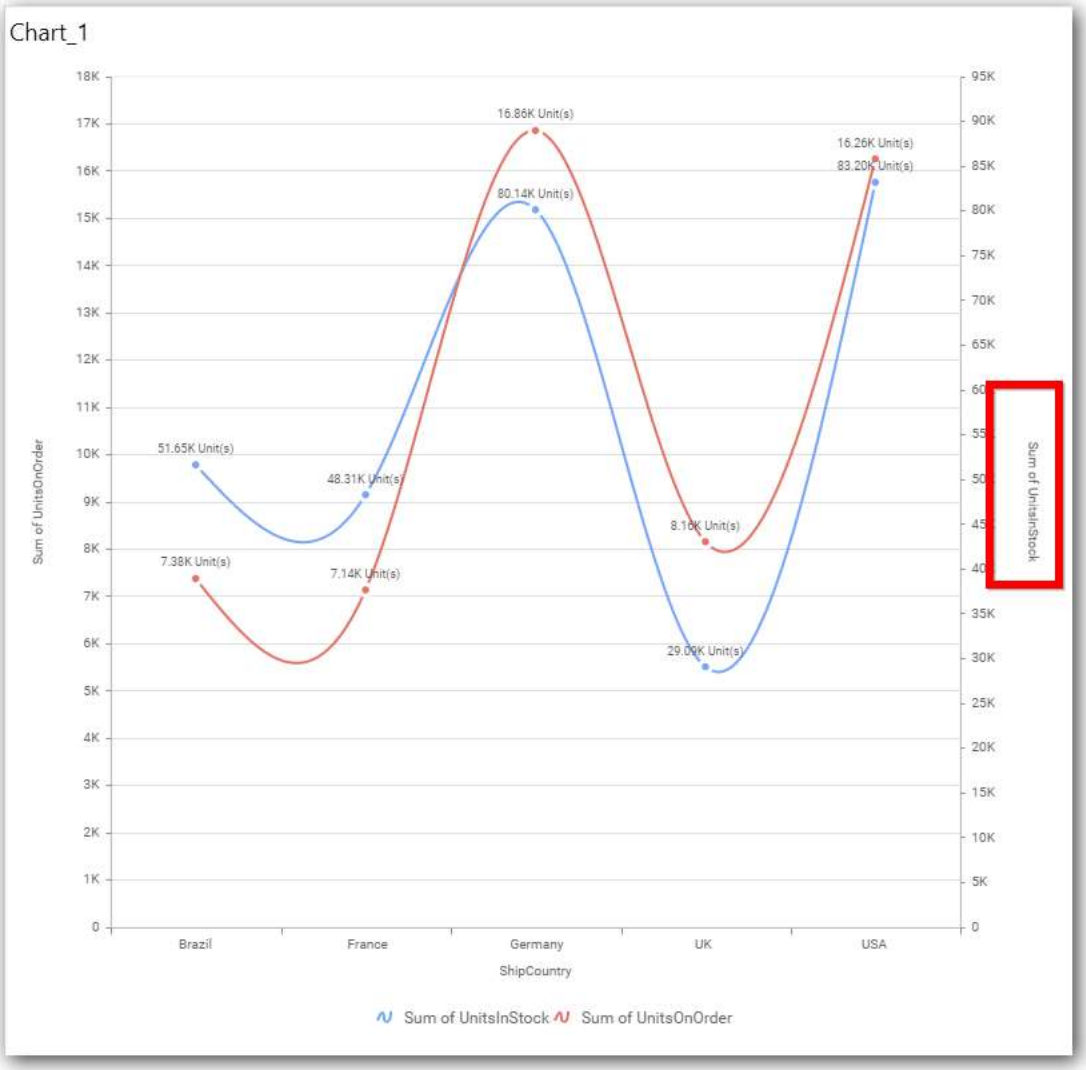
Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.



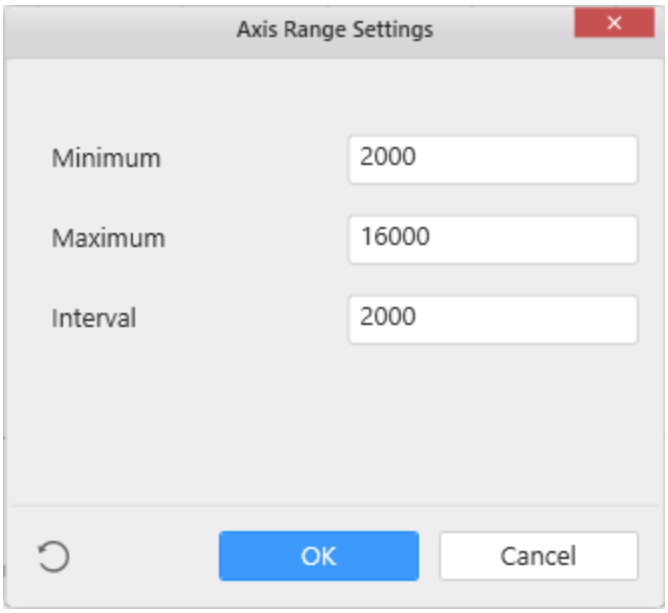
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

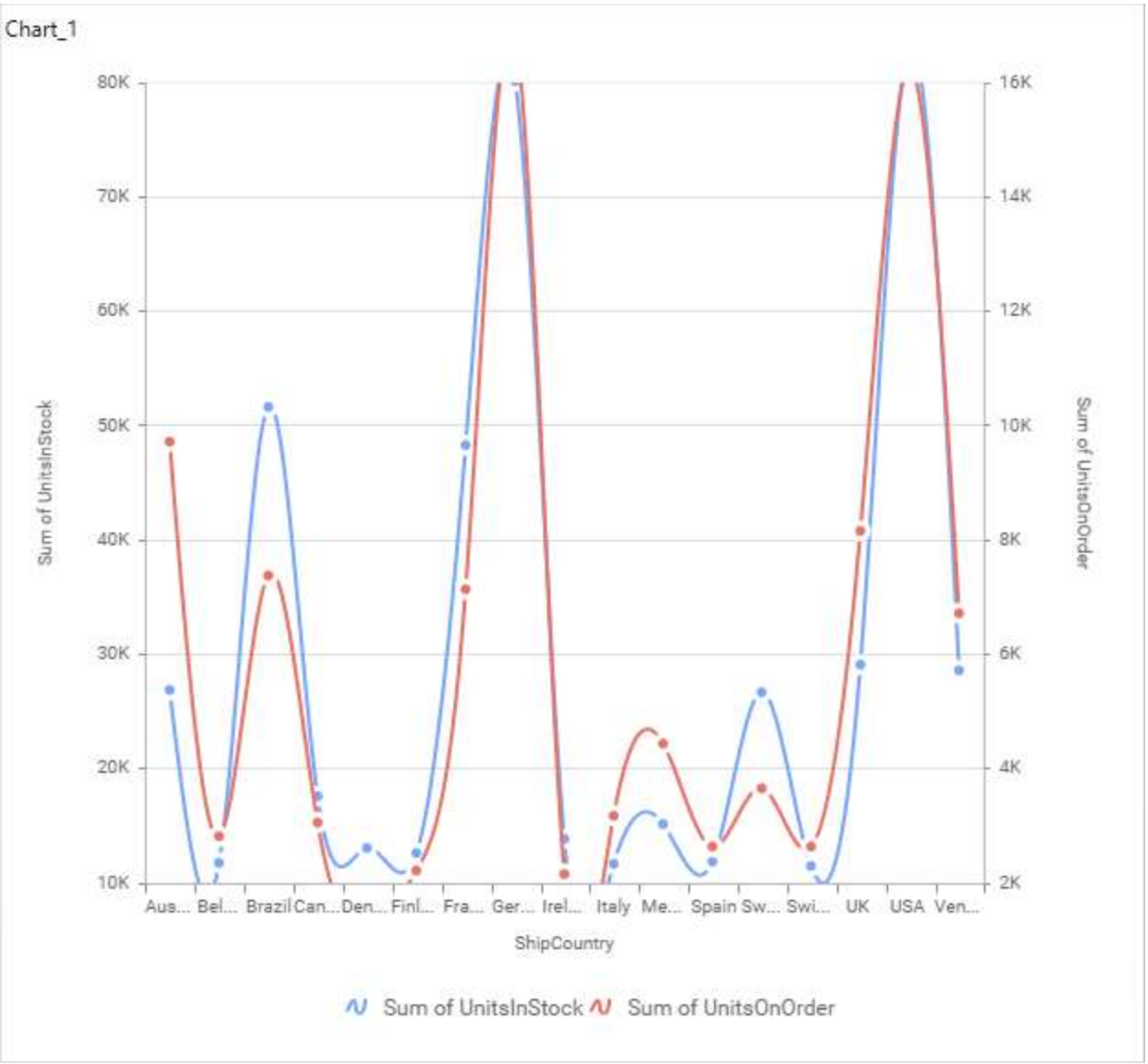


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

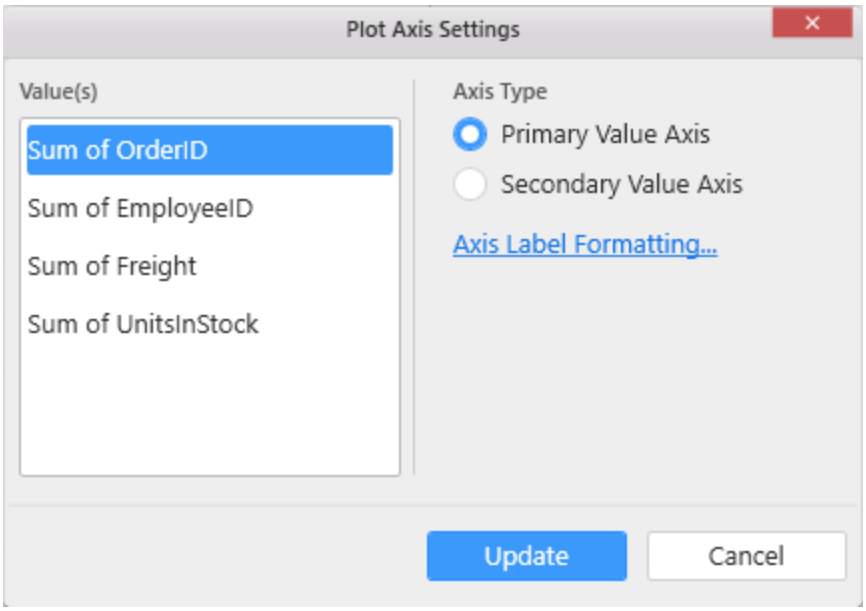


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

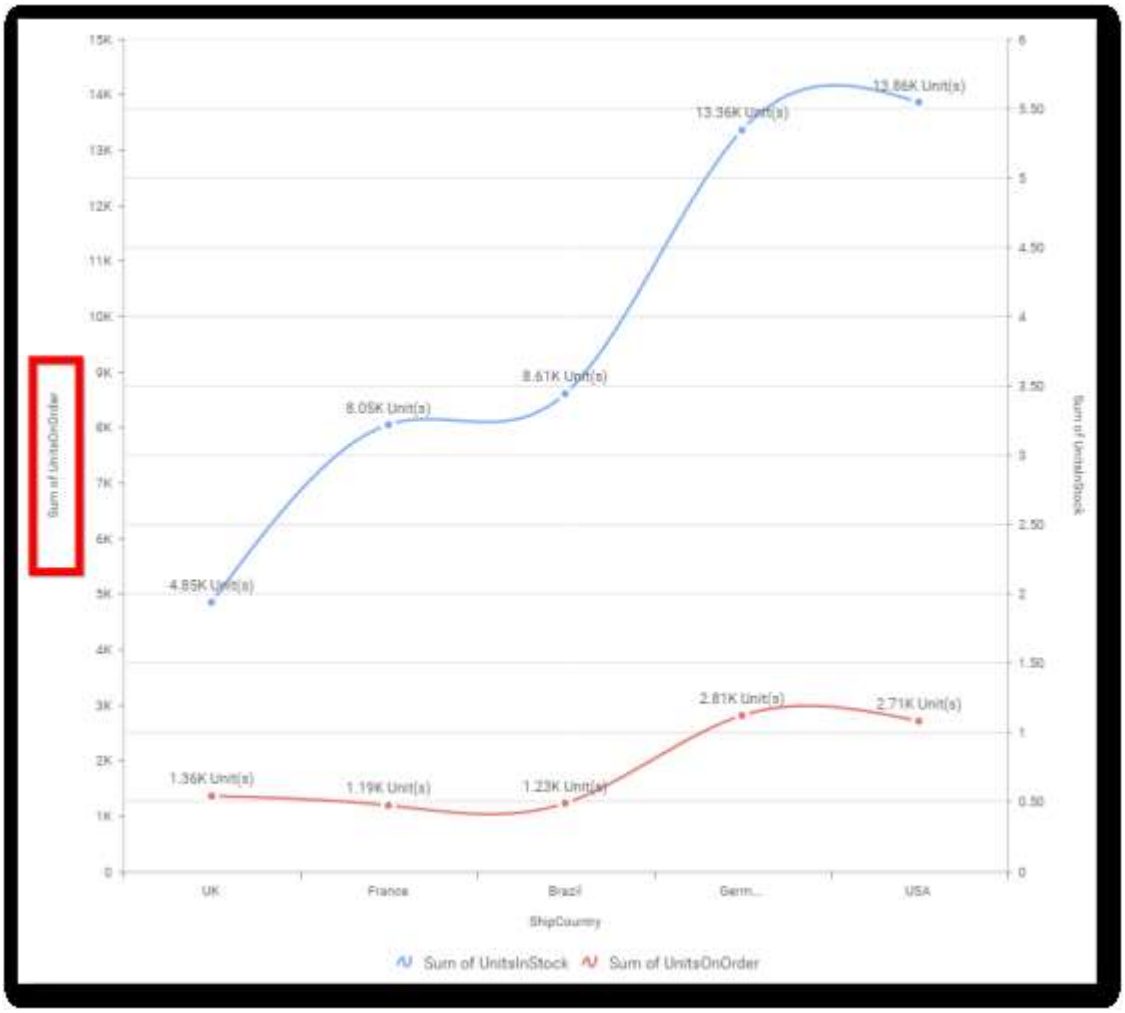


Plot Axis Settings

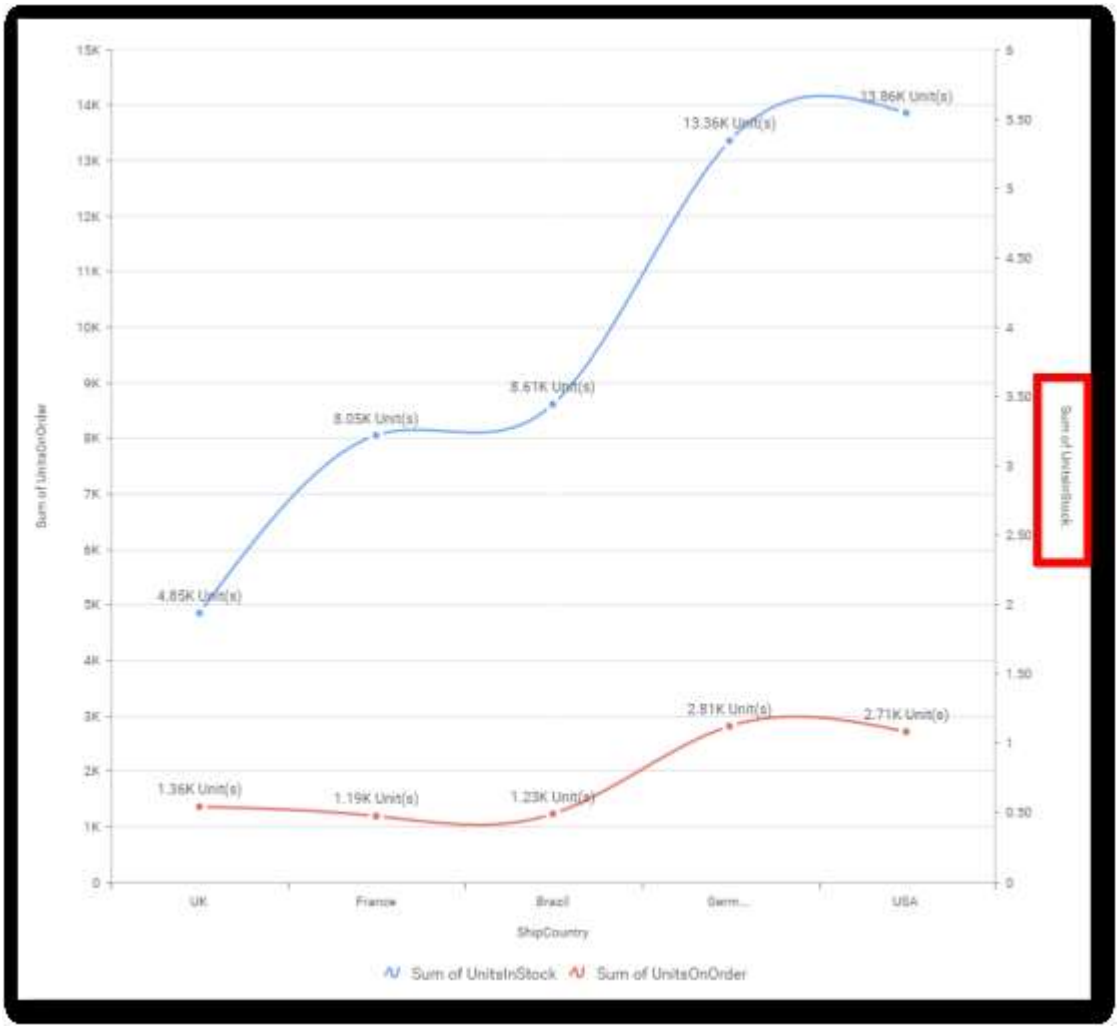
This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



Primary Value Axis

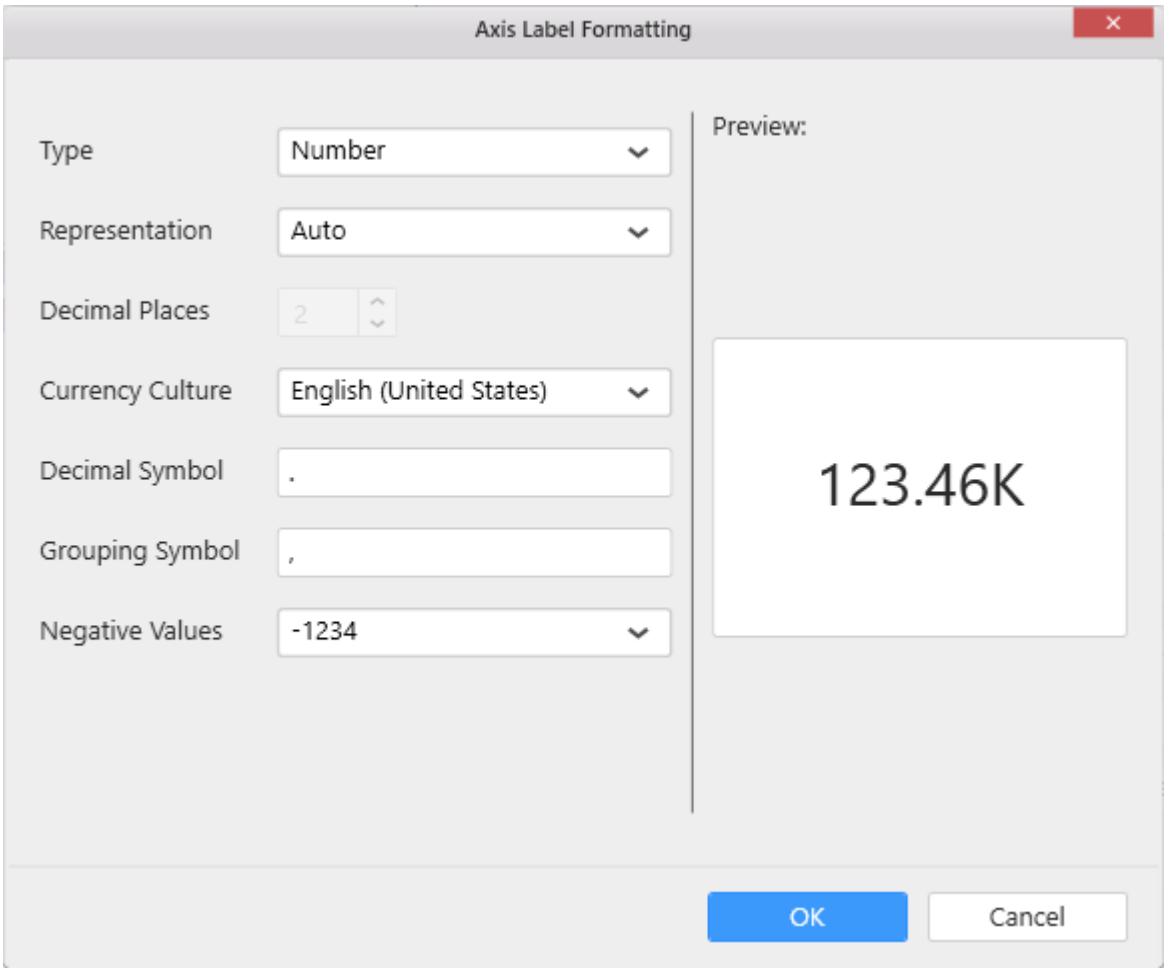


Secondary Value Axis



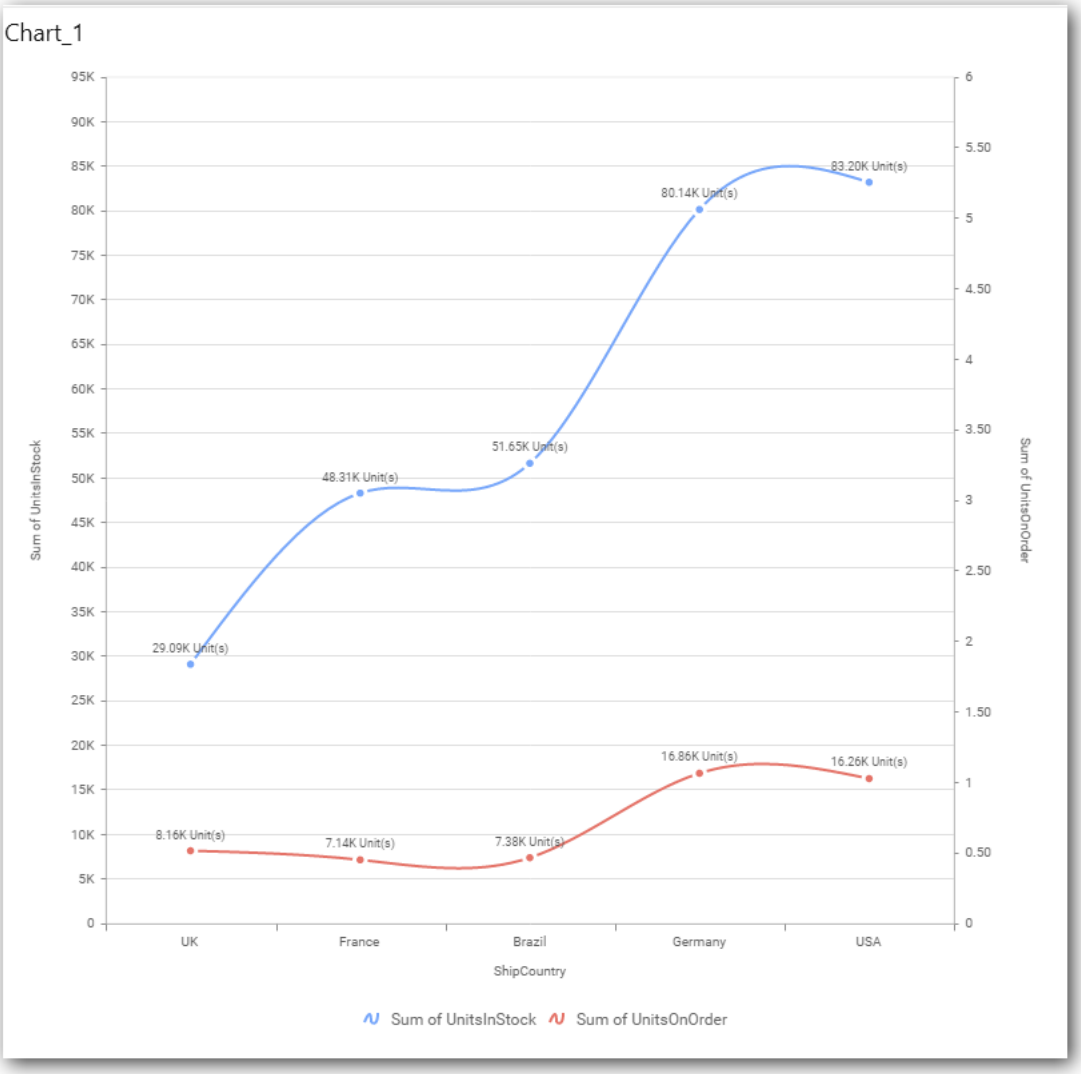
Axis Label Formatting

This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.

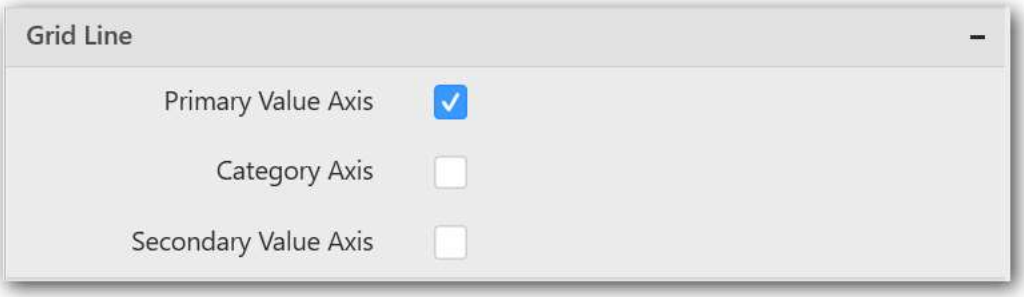


Sort Order

To define the sorting of chart based on any of the measures that you dropped and its order through this option. Following screenshot illustrates the Ascending sort order.

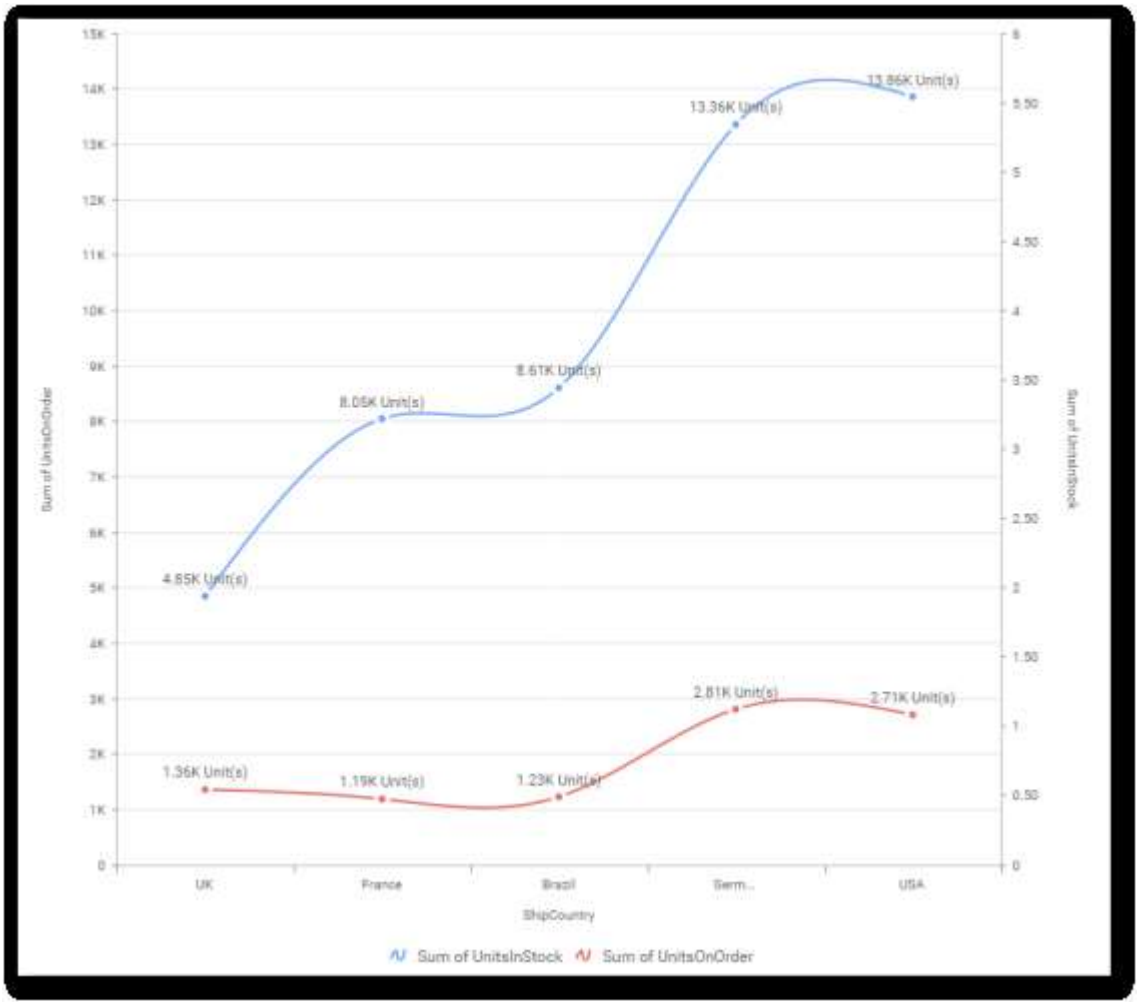


Grid Line Settings



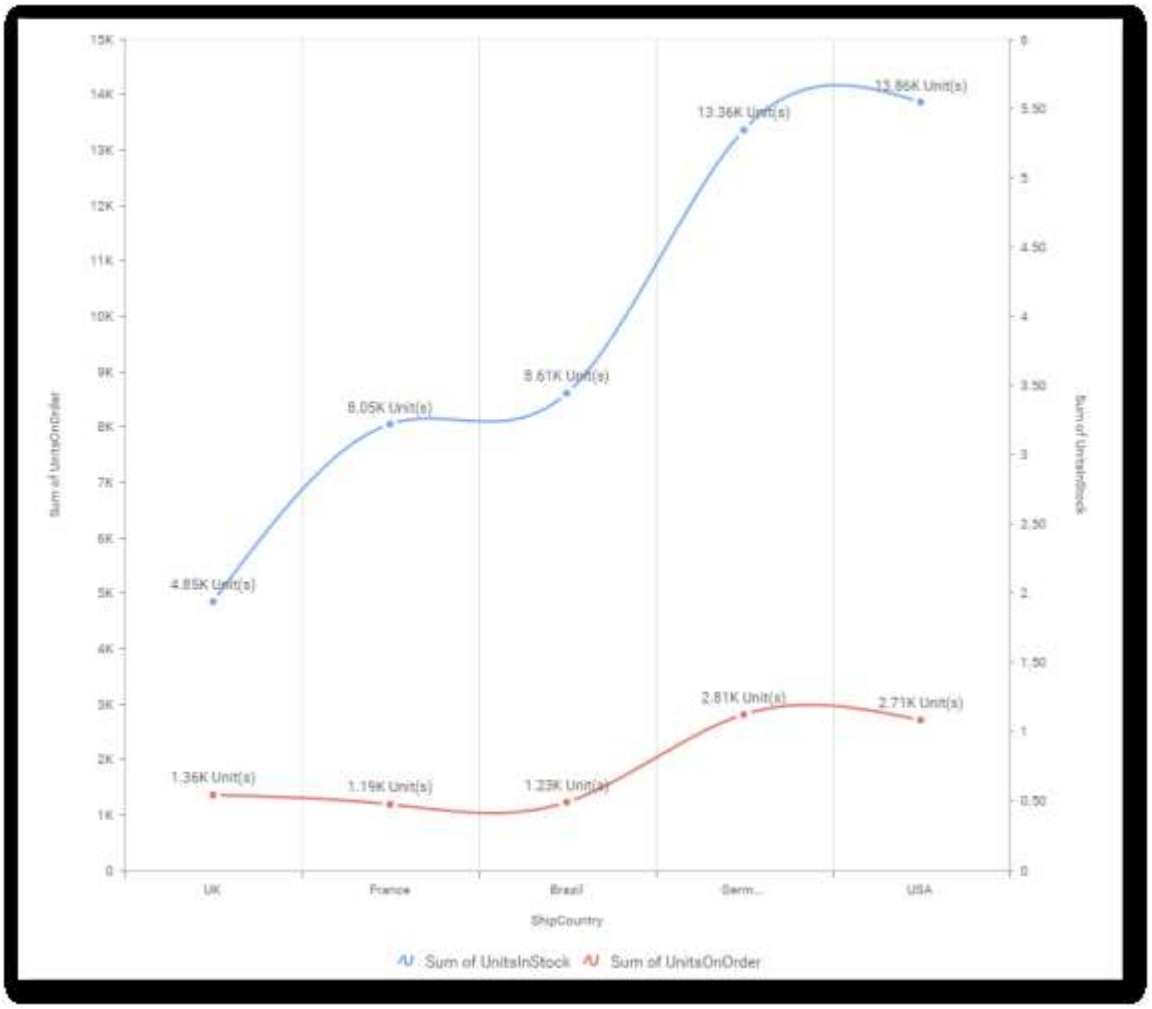
Primary Value Axis

This allows you to enable the Primary Value Axis gridlines.



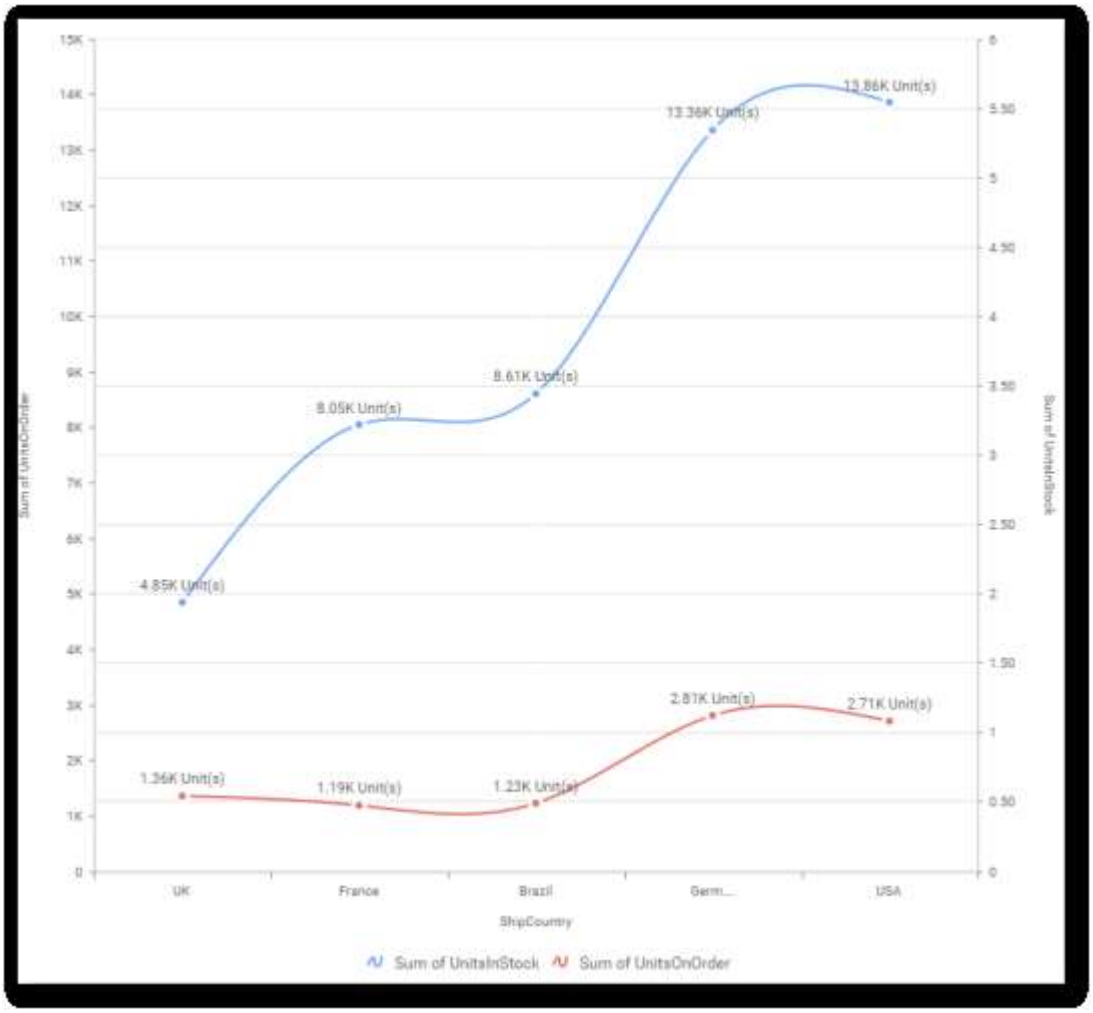
Category Axis

This allows you to toggle the visibility of Category Axis gridlines.



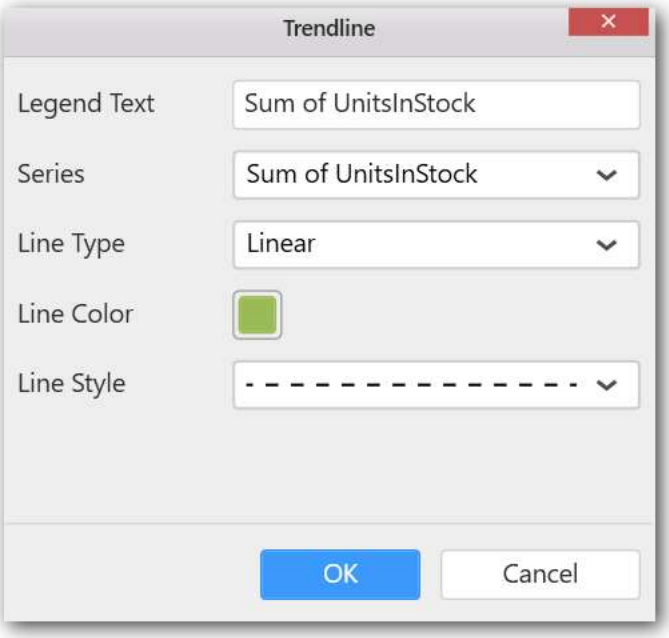
Secondary Value Axis

This allows you to toggle the visibility of Secondary Value Axis gridlines.



Trend Line Settings

You can add trend line to chart based on dropped measure that you select. You can also customize its, legend text, line type and line color. Trend line is not visible, by default.



After applying these settings, it will reflect in chart like below.

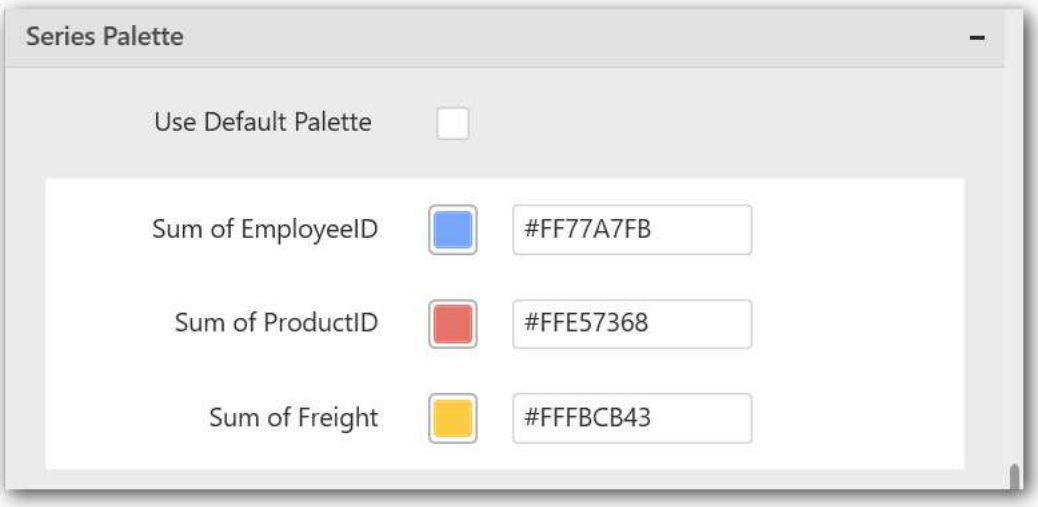
You have options to edit or delete the added trend lines.

Series Palette

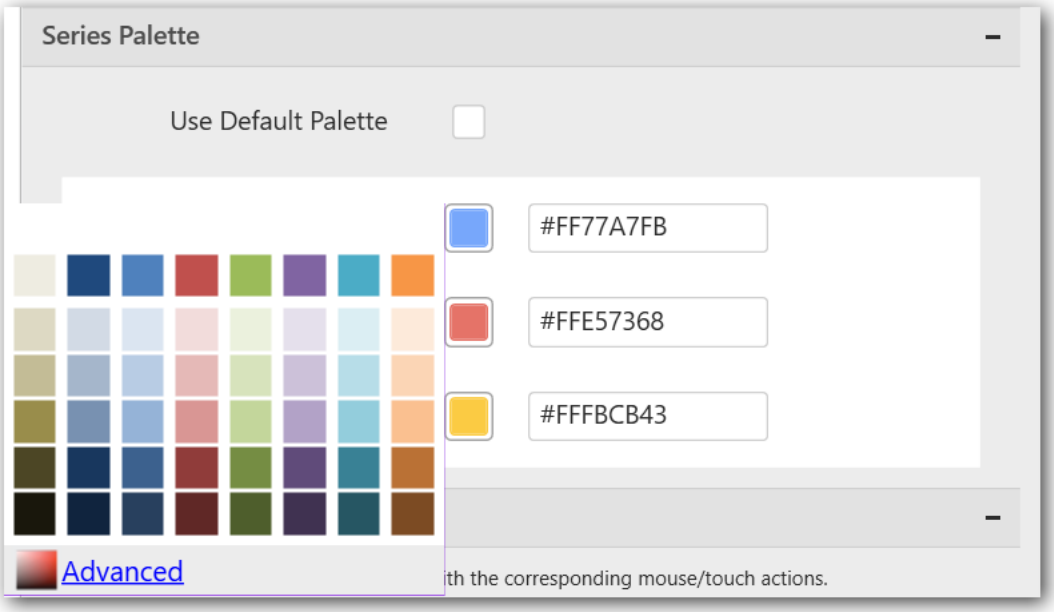
This allows you to customize the chart series color through Series Palette section.

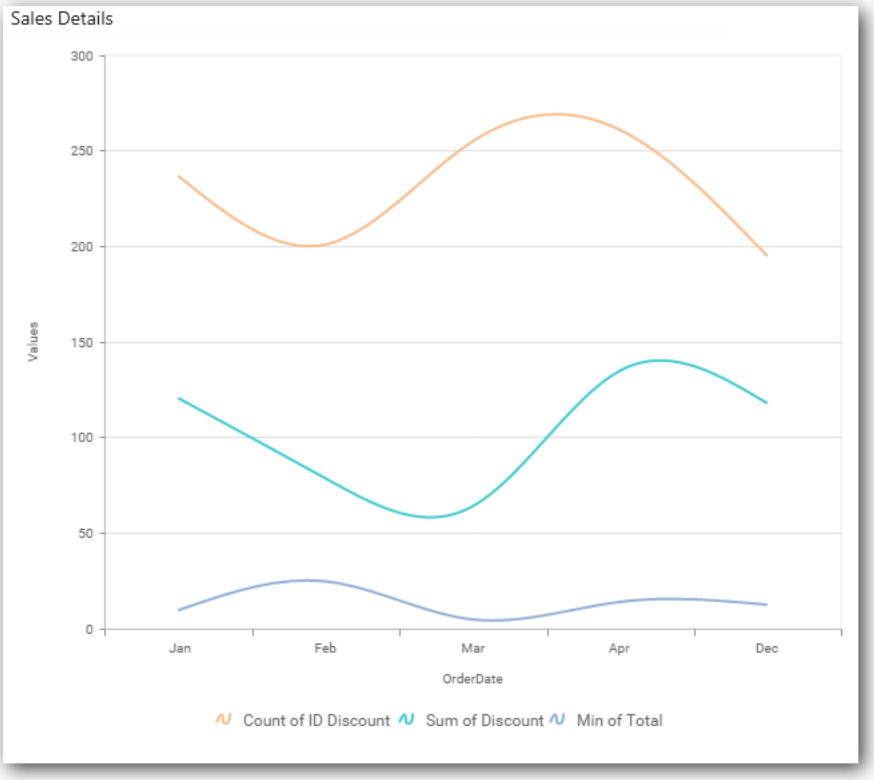
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



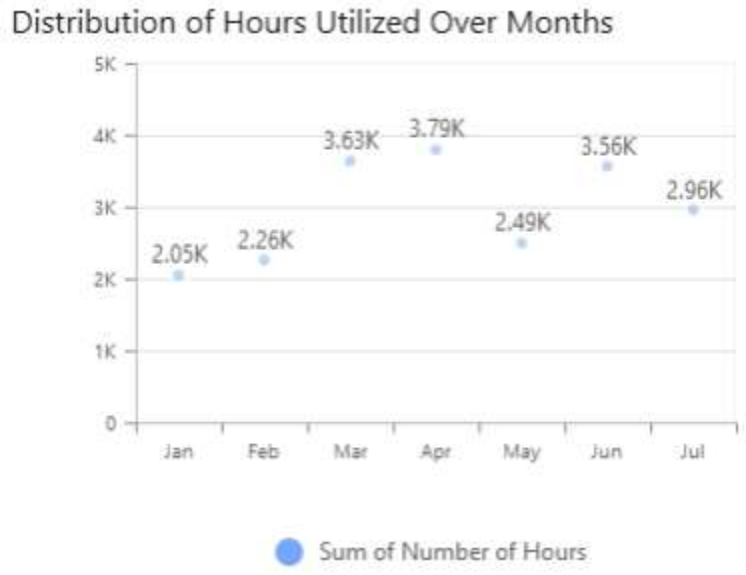
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Scatter Chart

Scatter Chart allows you to compare large number of data points represented as dots irrespective of time.



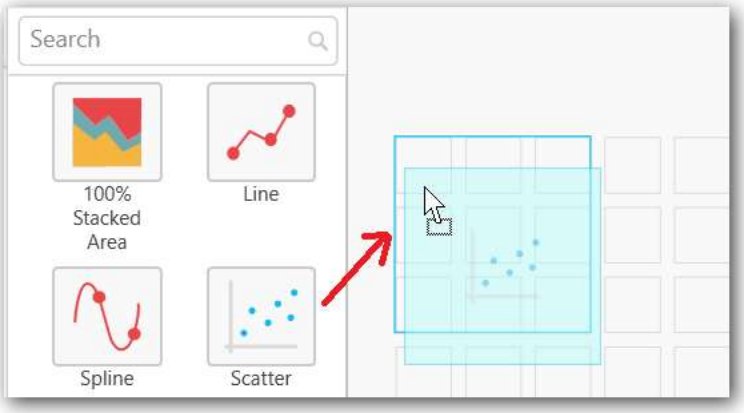
How to configure flat table data to Scatter Chart?

Scatter Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that

you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

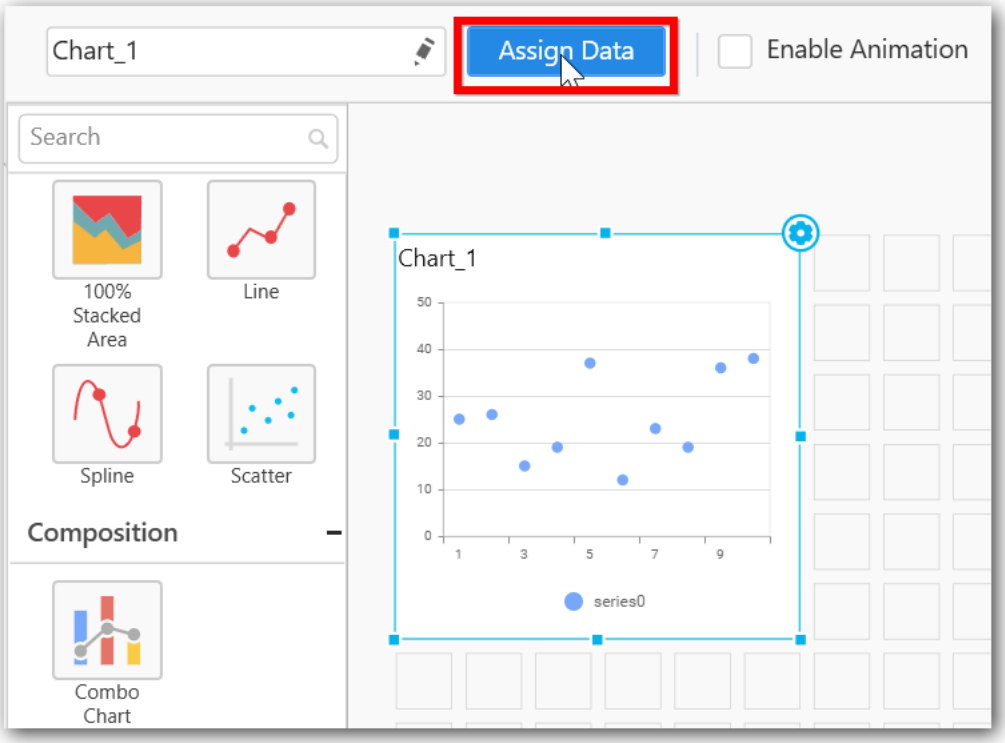
Follow the steps to configure data into Scatter chart

Drag and drop the Scatter chart into canvas and resize it to your required size.

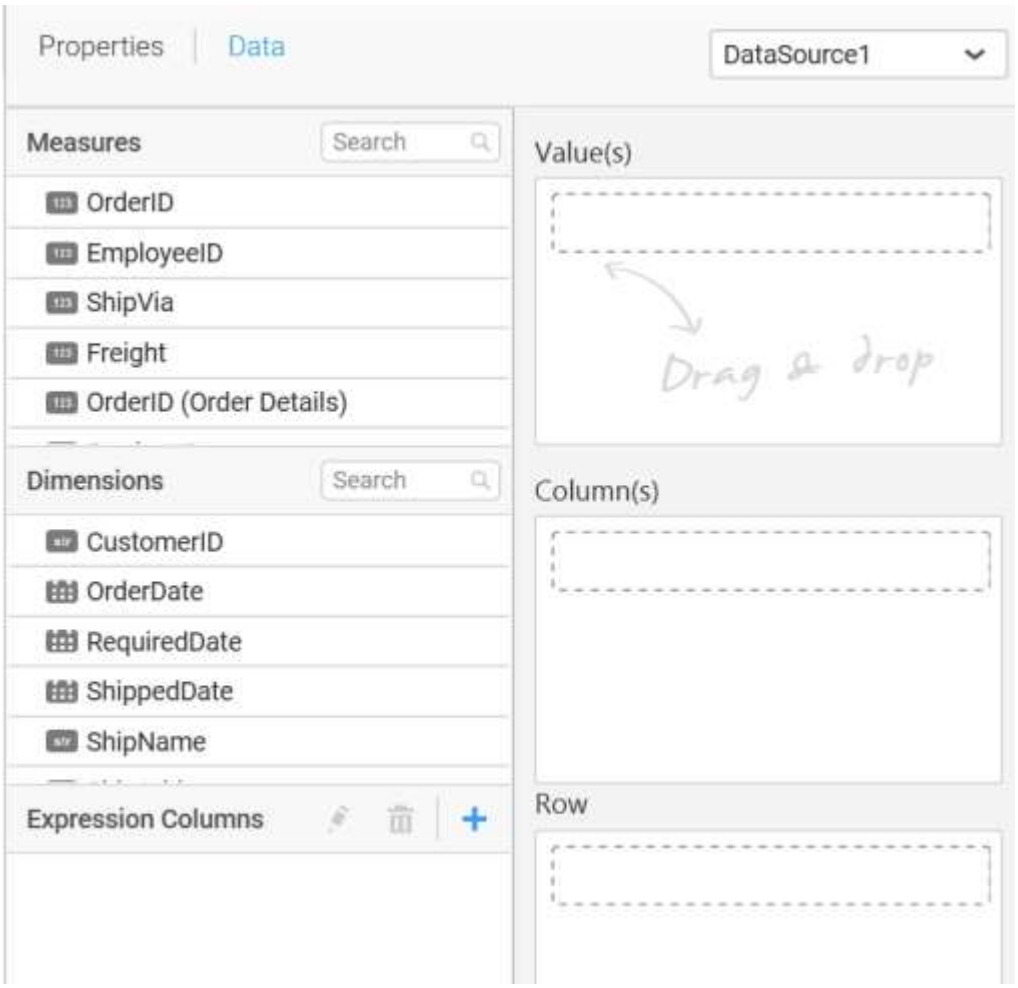


Connect to data source.

Focus on the Scatter chart and click on Assign Data button.

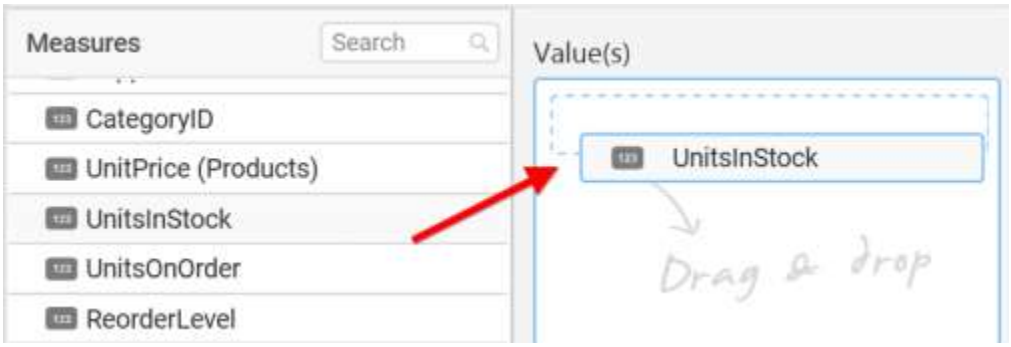


The data pane will be opened with available Measures and Dimensions from the connected data source.

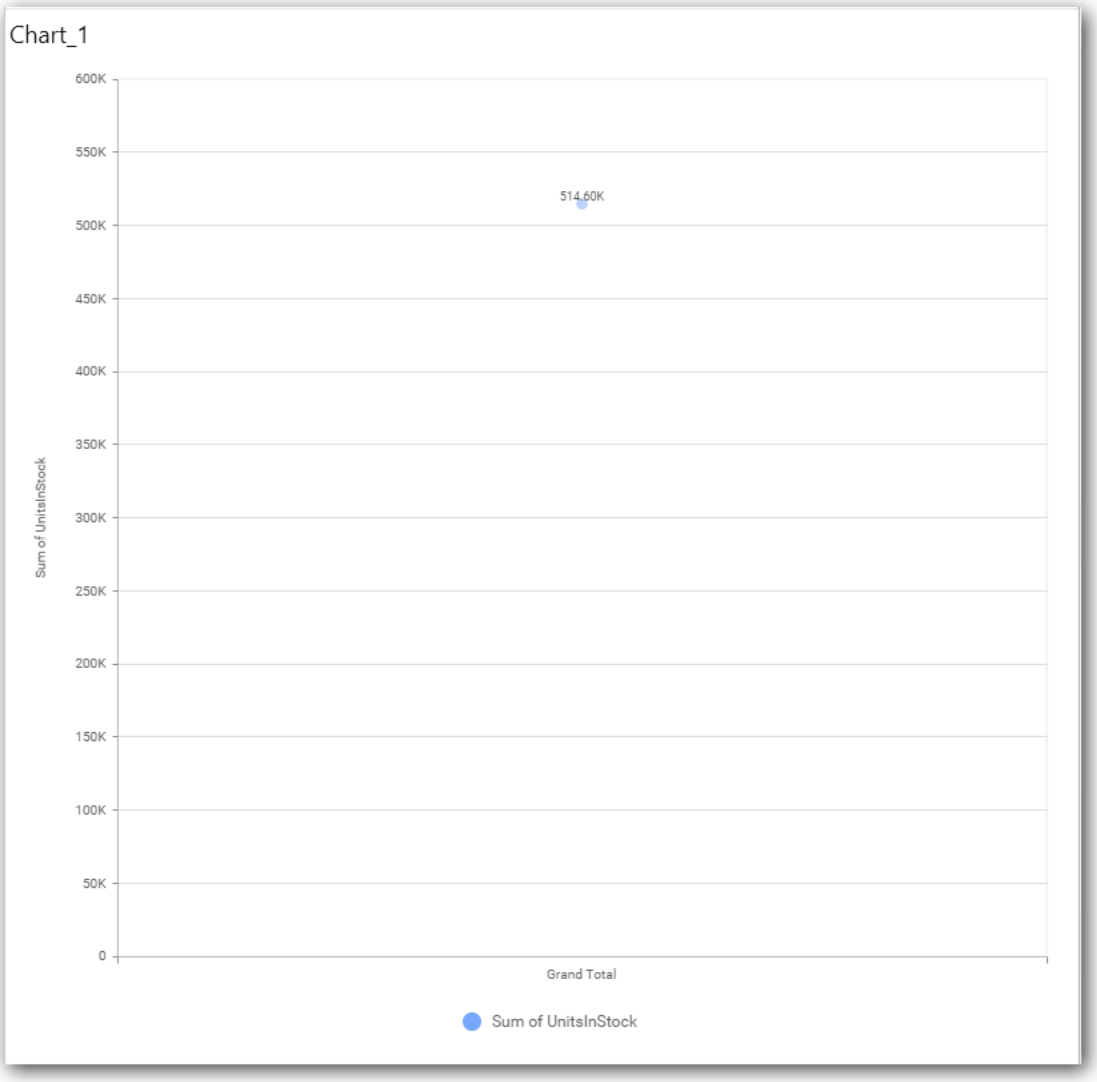


Assigning Value(s)

Drag and drop the Measure into Value(s).

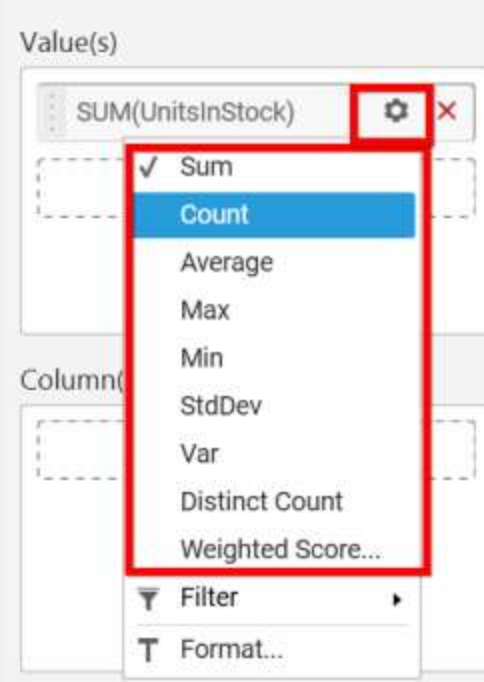


Now the chart will be rendered like this.

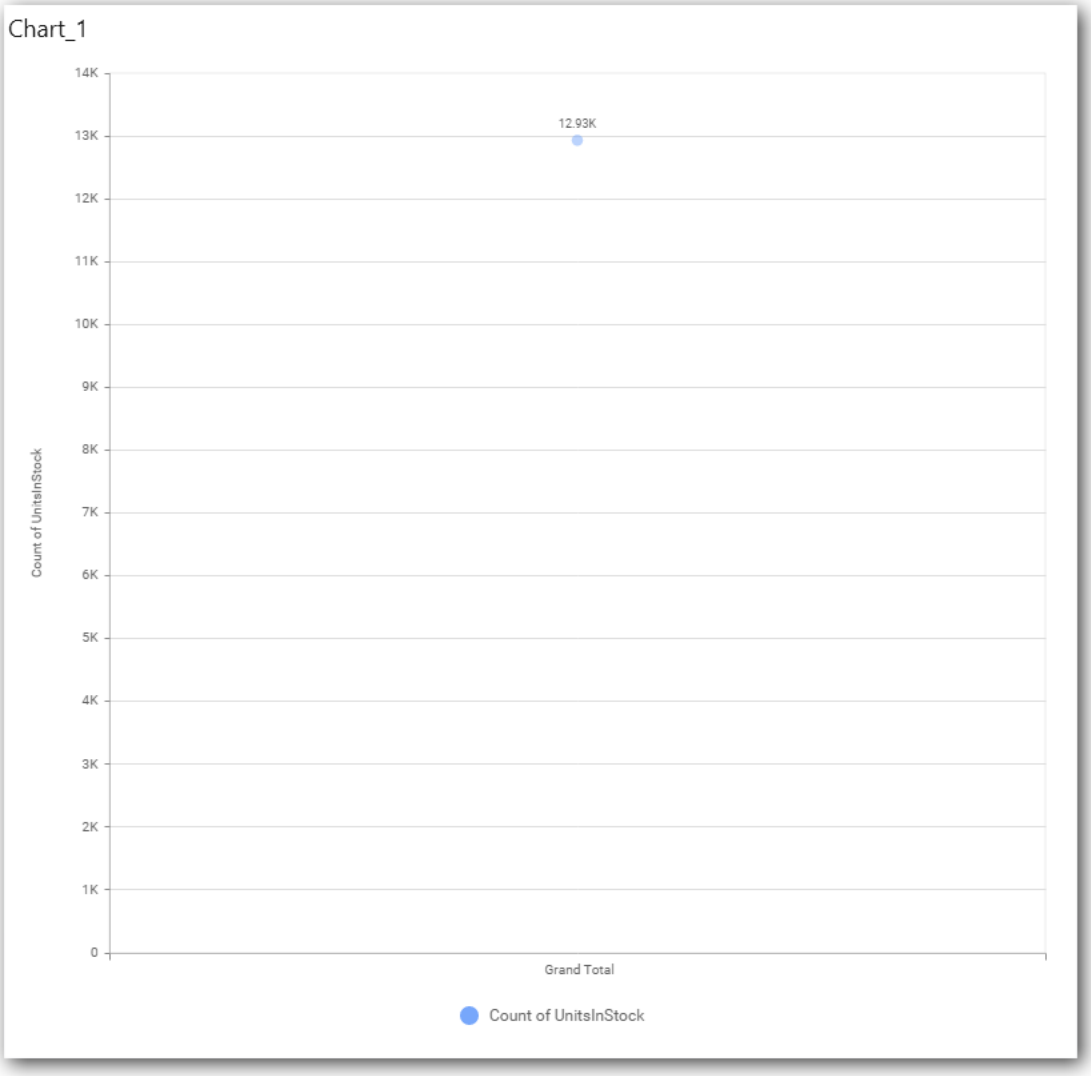


You can change the summary type of the value by clicking on Settings option.

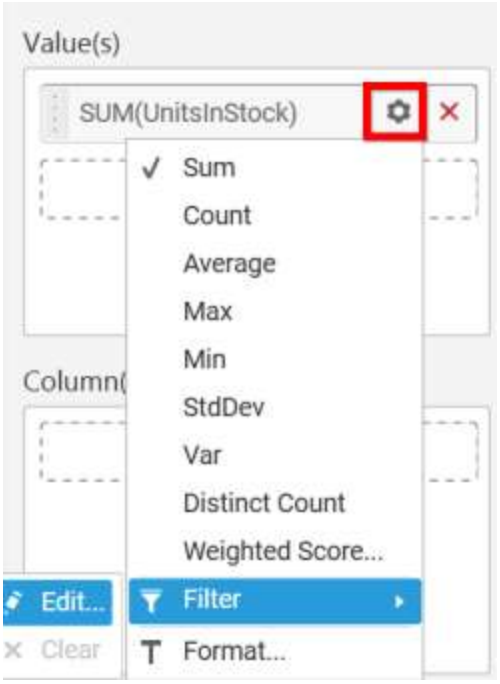
Select the required summary type from list.



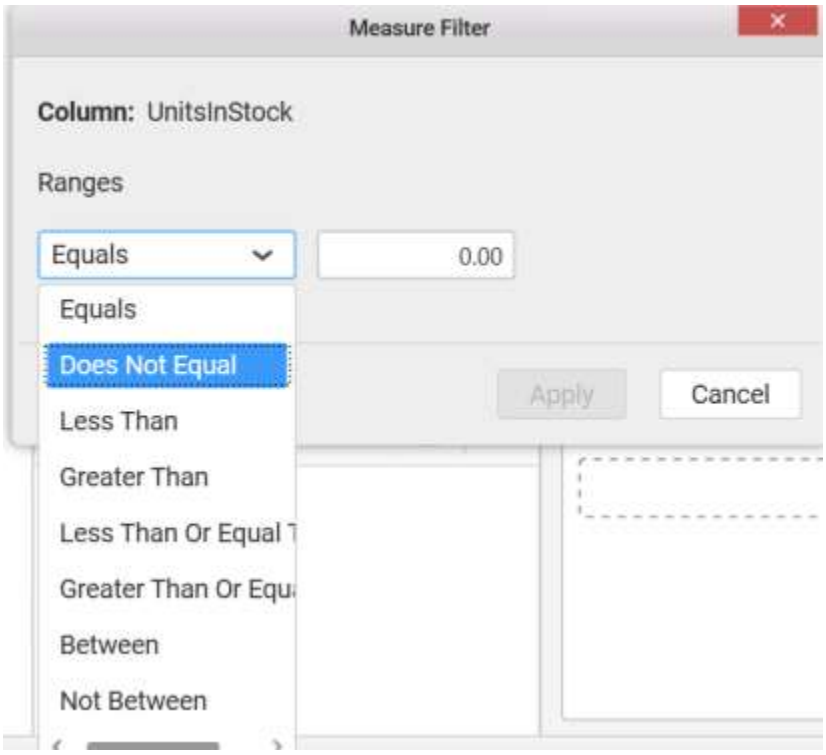
You can see the summary type changed to Count from Sum.



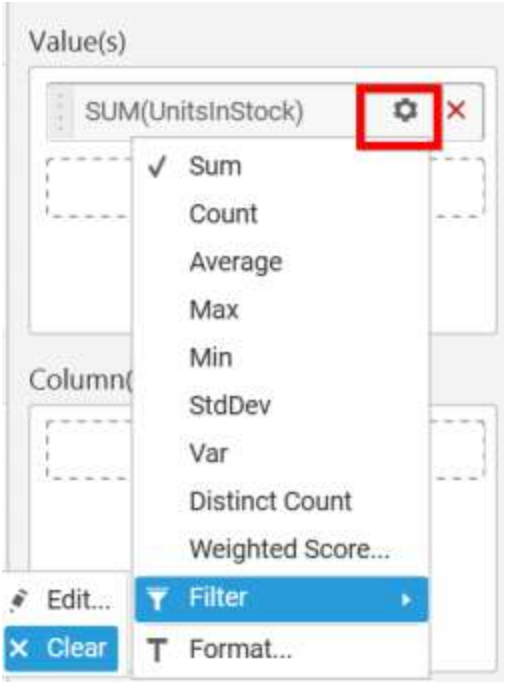
You can select what data to be displayed by choosing filter option.



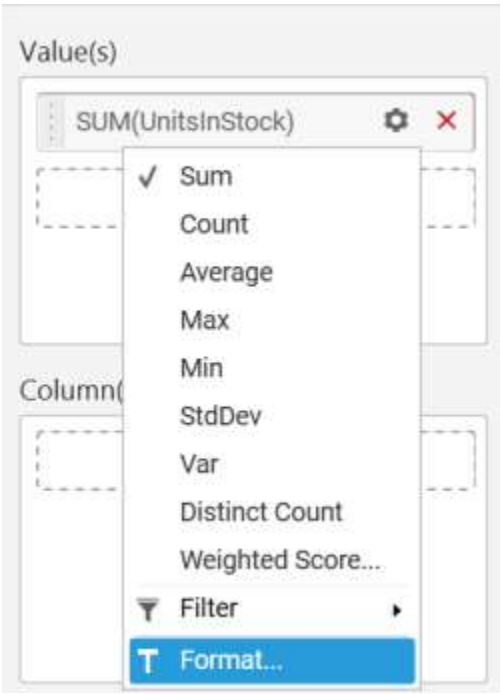
The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.



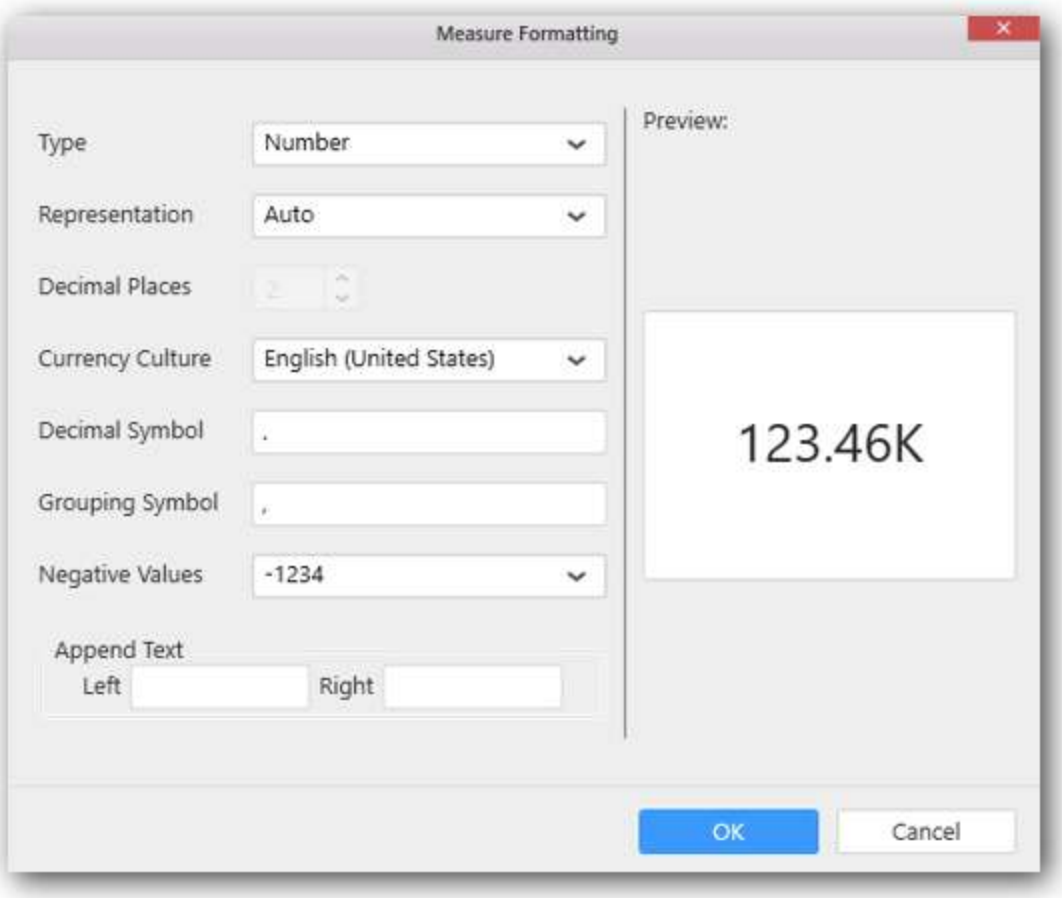
You can Clear the filter.



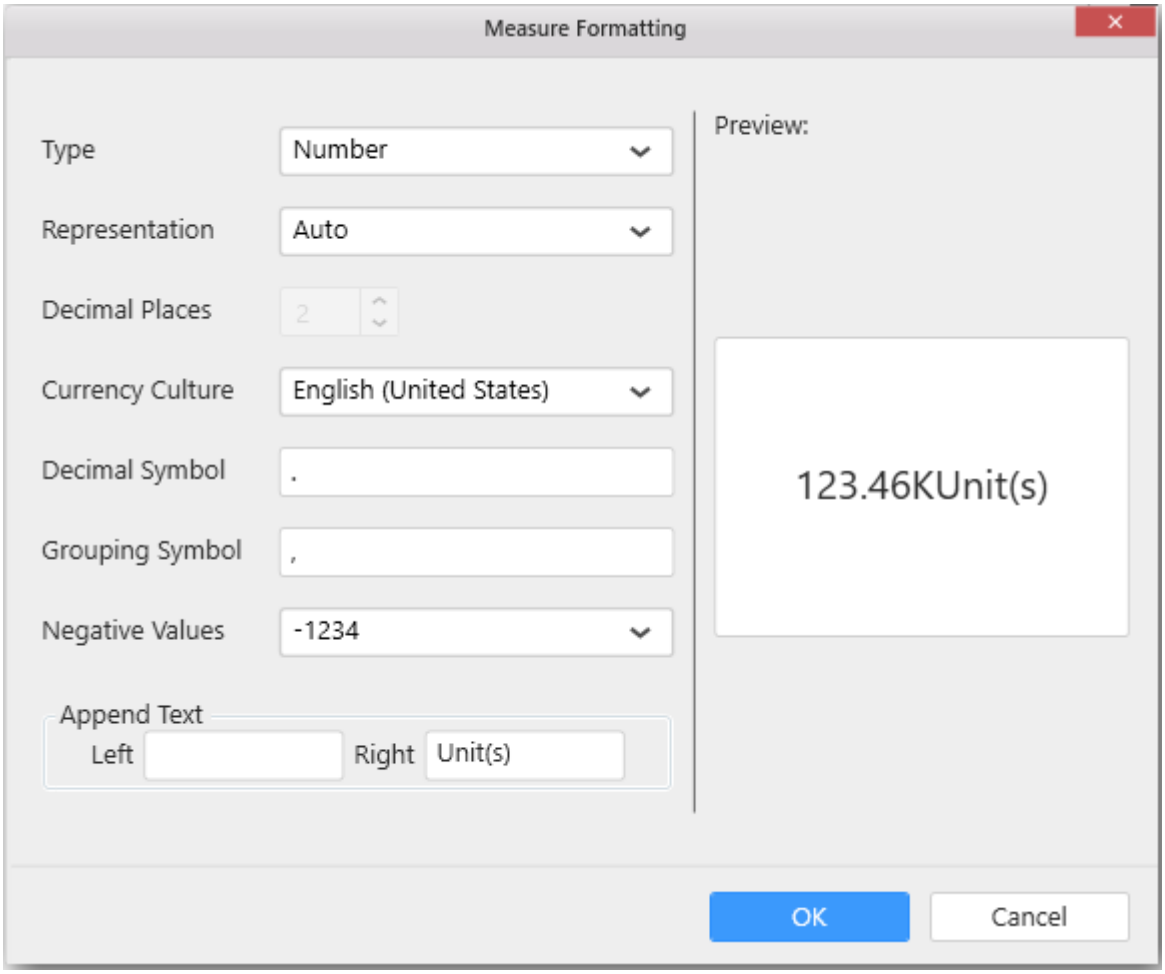
You can Format the value.



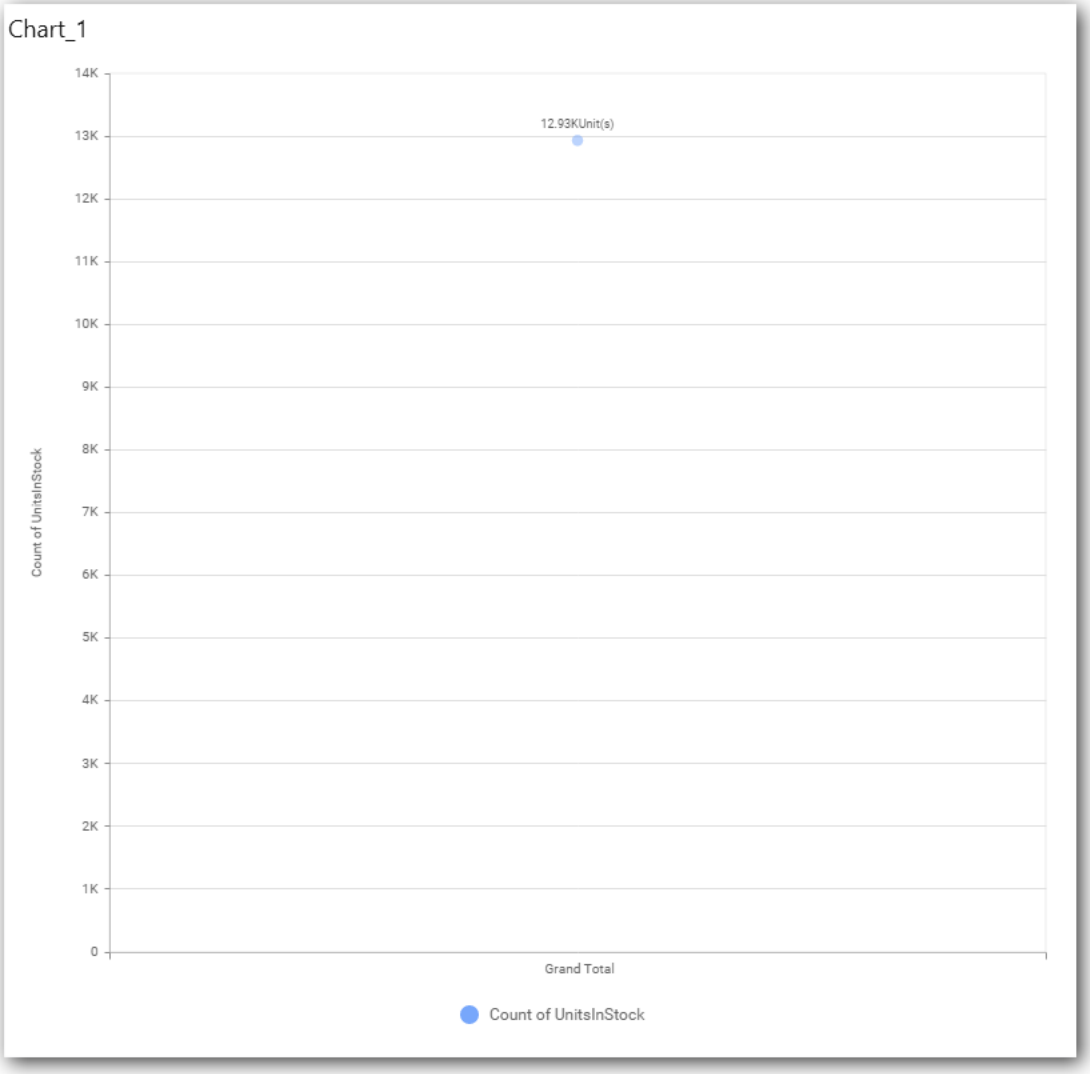
The format options will be shown.



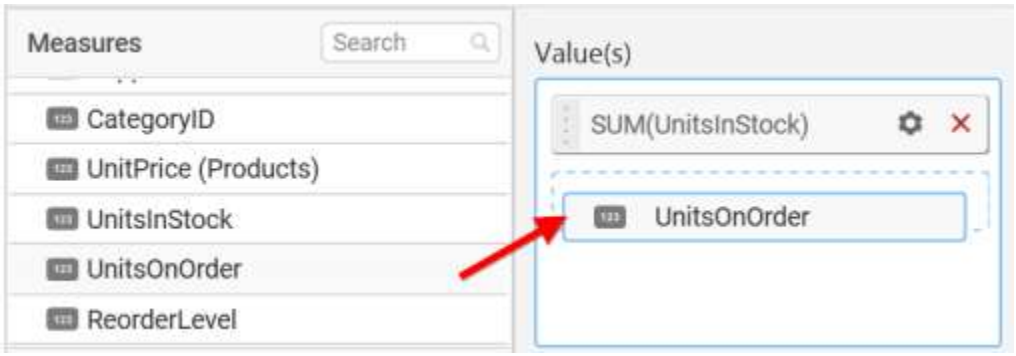
Choose the options you need and click OK.



Now the Chart will be rendered like this.



You can add more number of values by drag and drop the Measures into Value field.

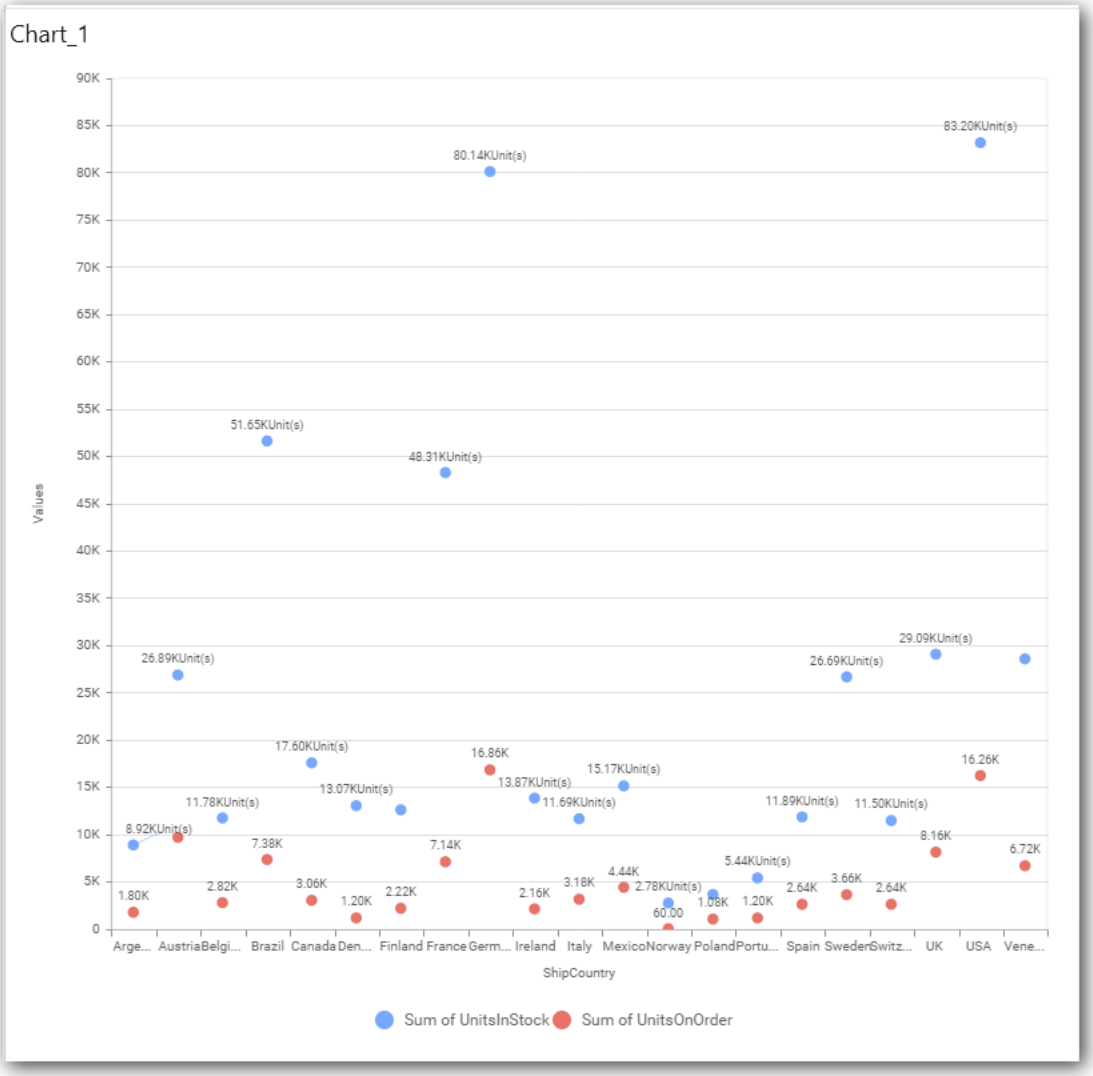


!Measures into Value` field[](images/scatterchart_img18.png)

You can also add Dimensions and Columns to Value(s).

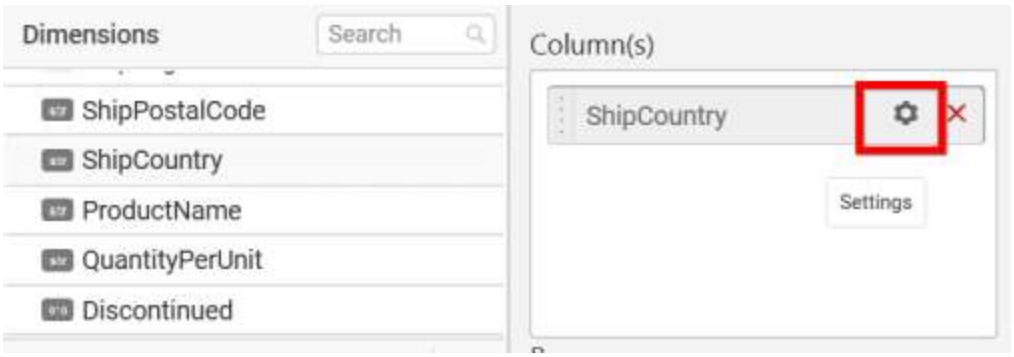
Assigning Column(s)

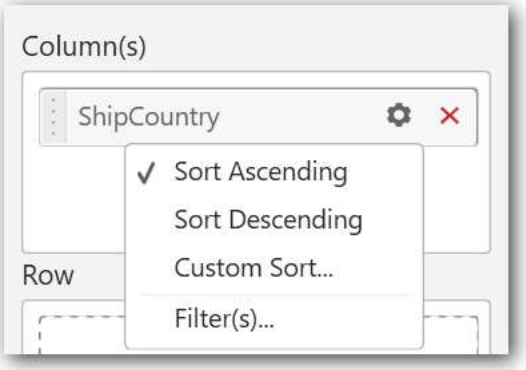
You can add the Dimension into Column field by drag and drop.



You can also add Measures and Expression Columns into Column(s) field.

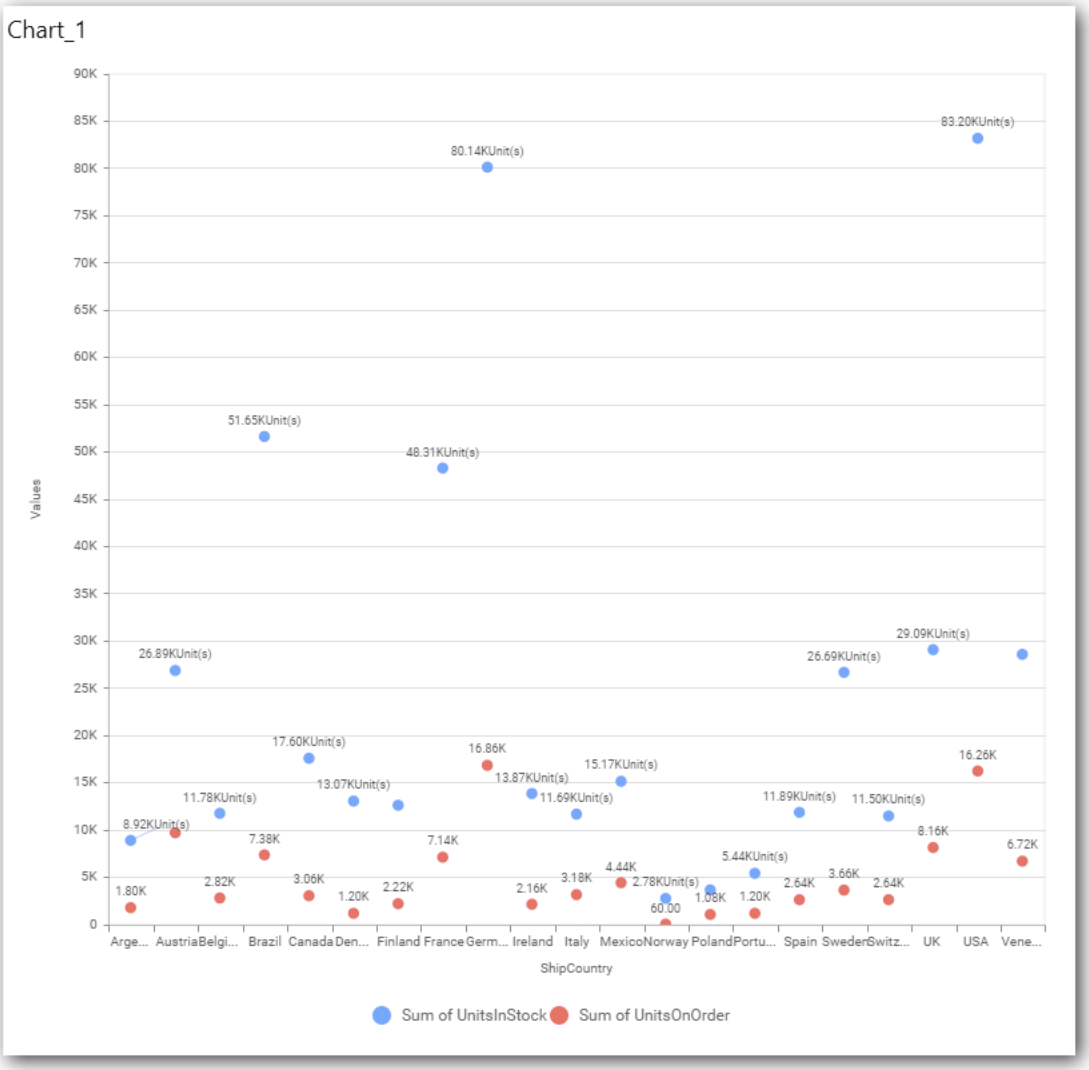
You have options to change the settings.



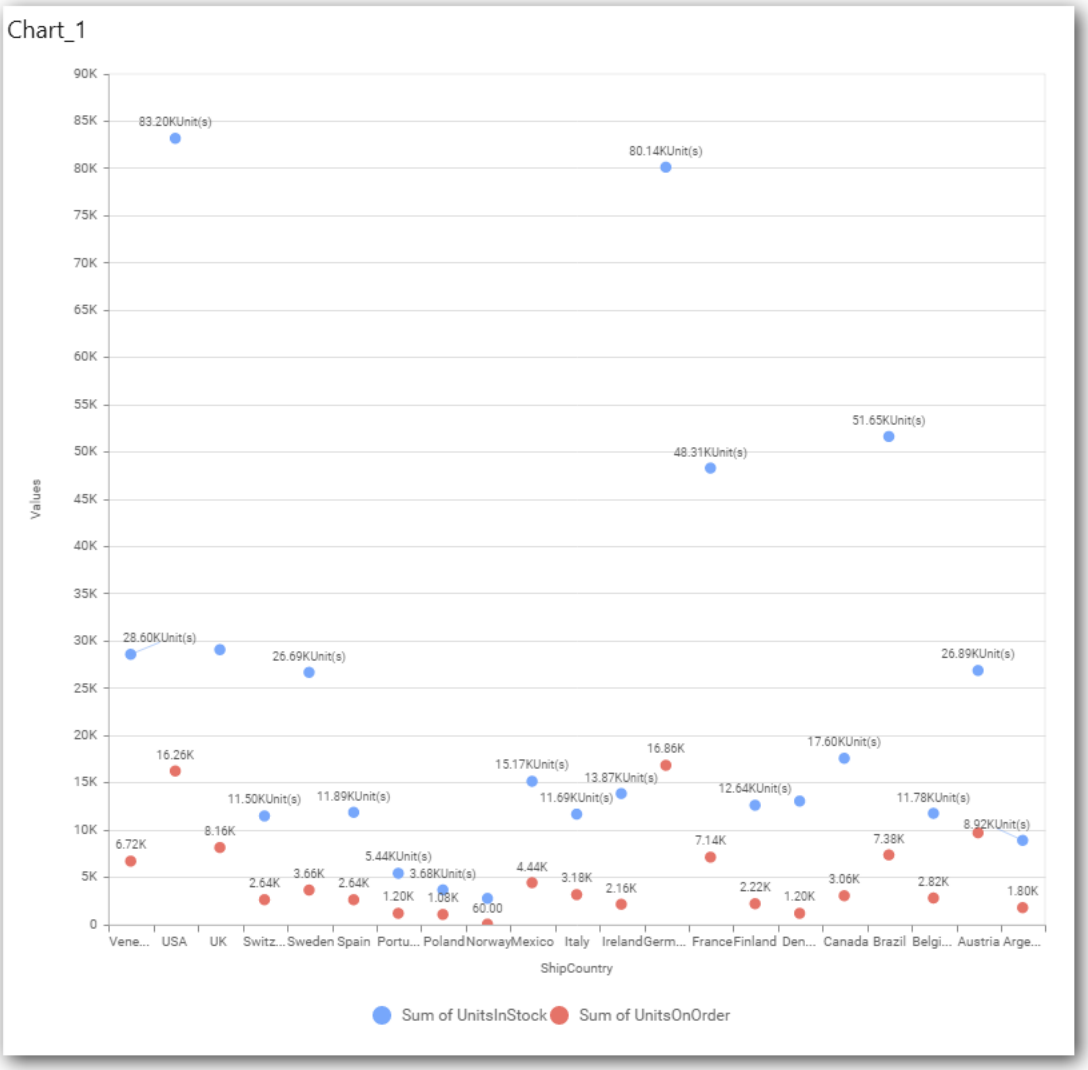


You can sort the chart either in Ascending or Descending series.

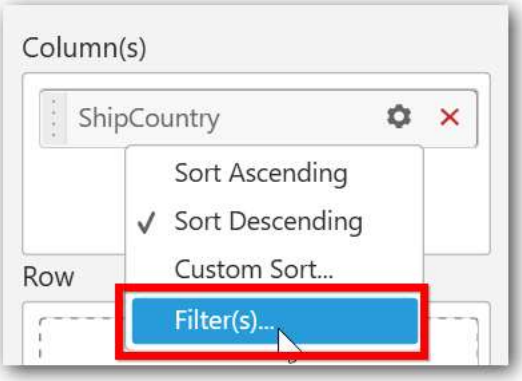
Ascending Order

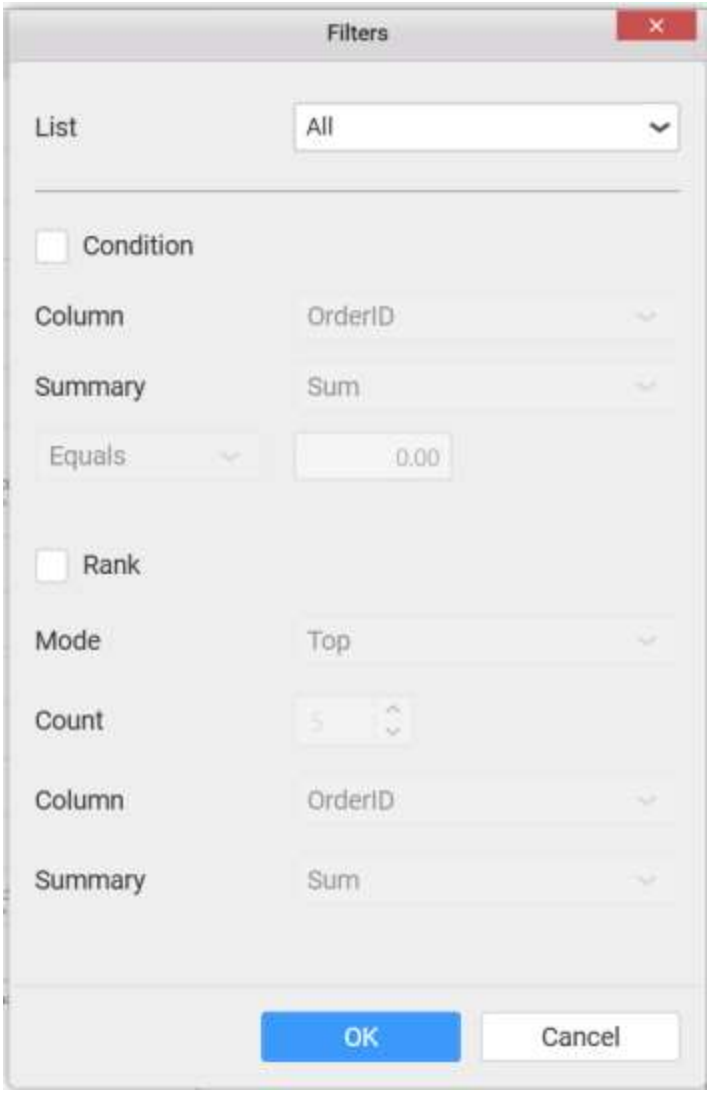


Descending order

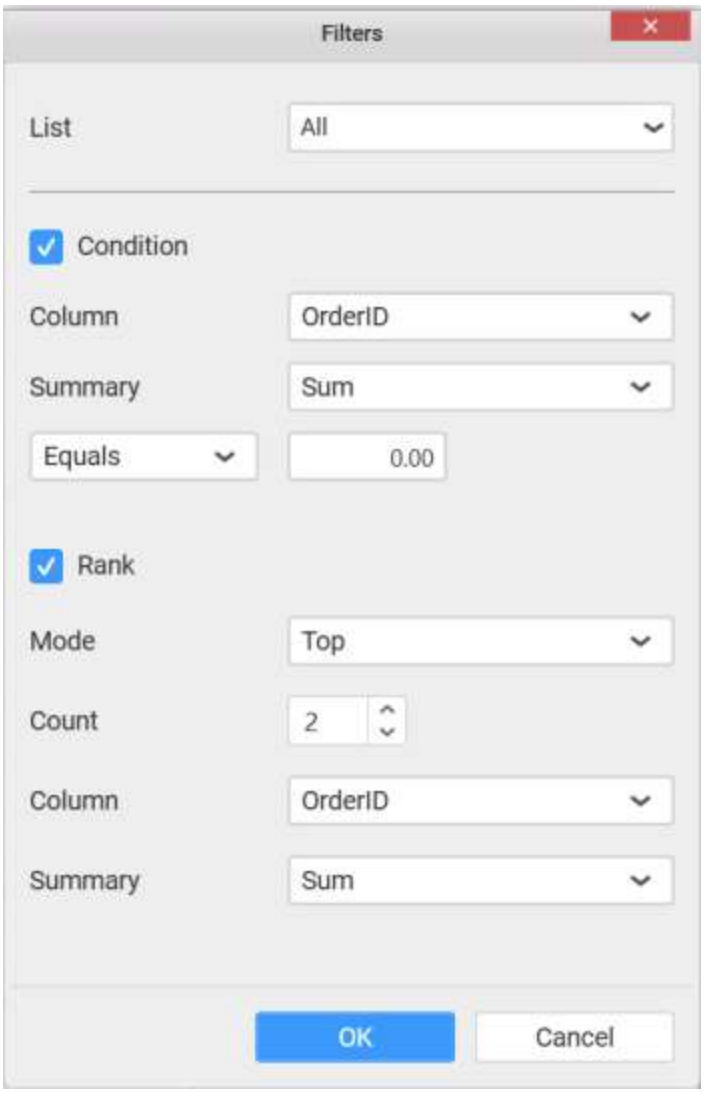


You can apply a filter.

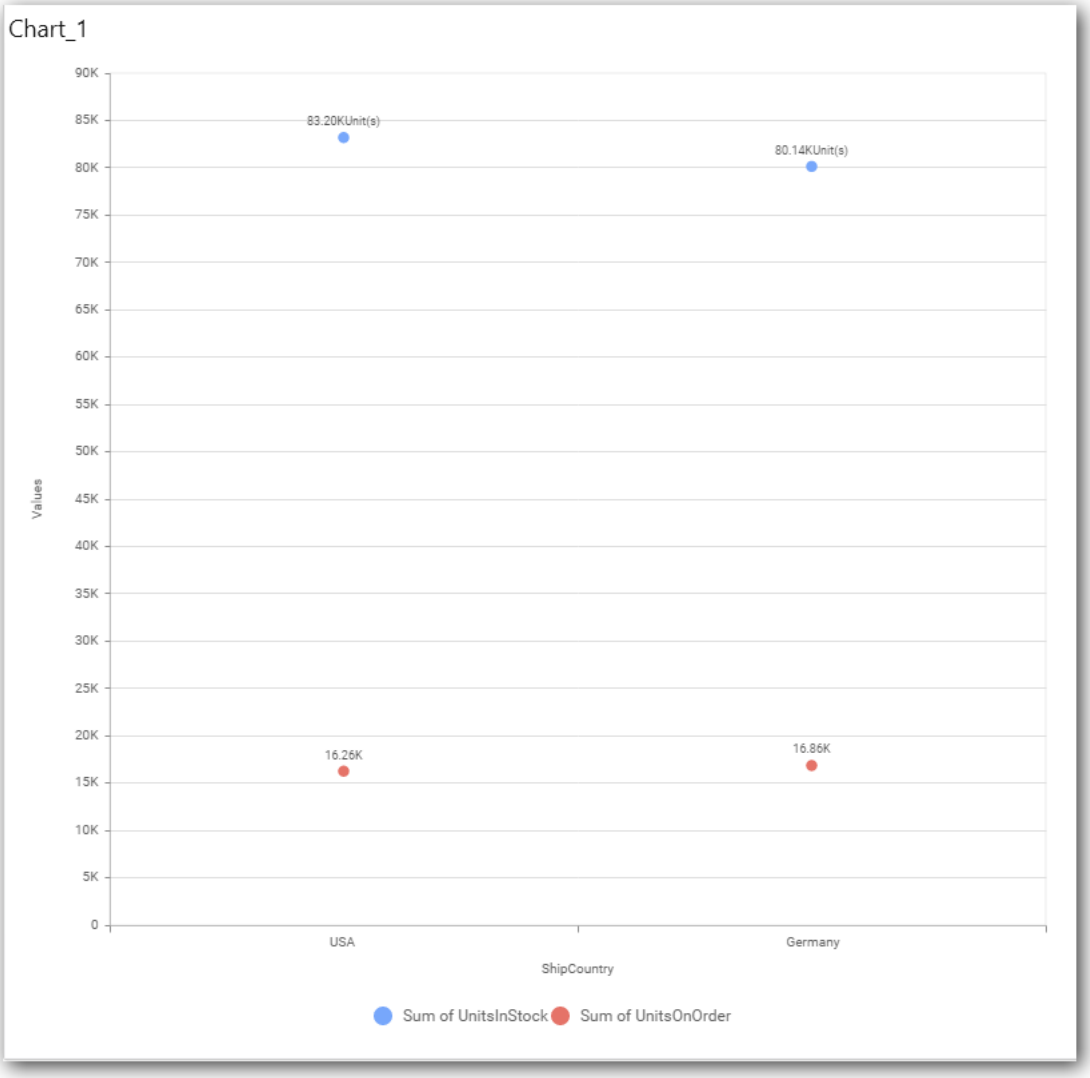




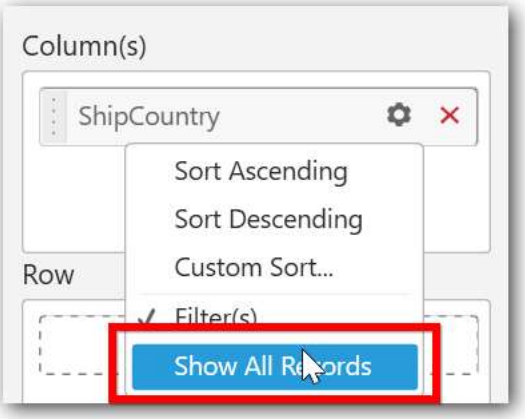
Select the Conditions and Rank you need.



Now the chart will be rendered like this.



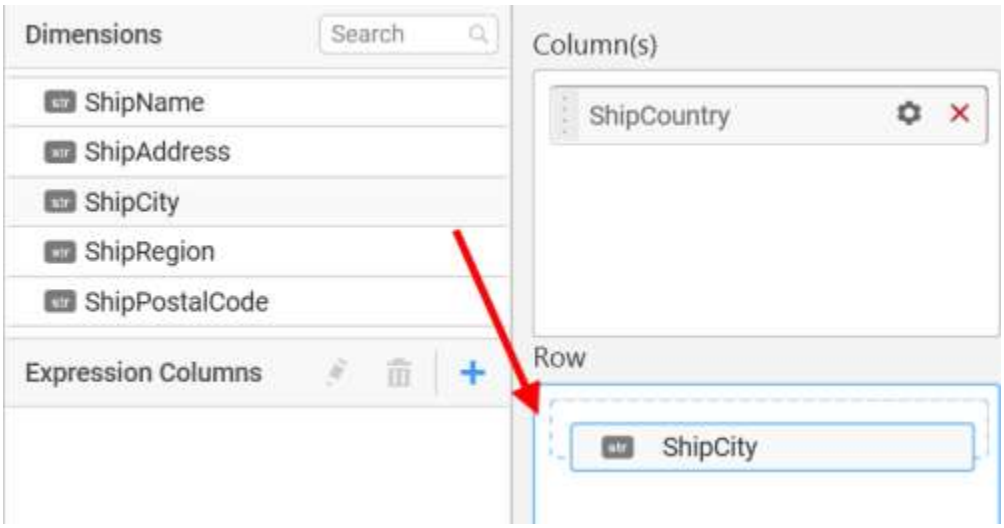
To show all records again click on Show All Records.



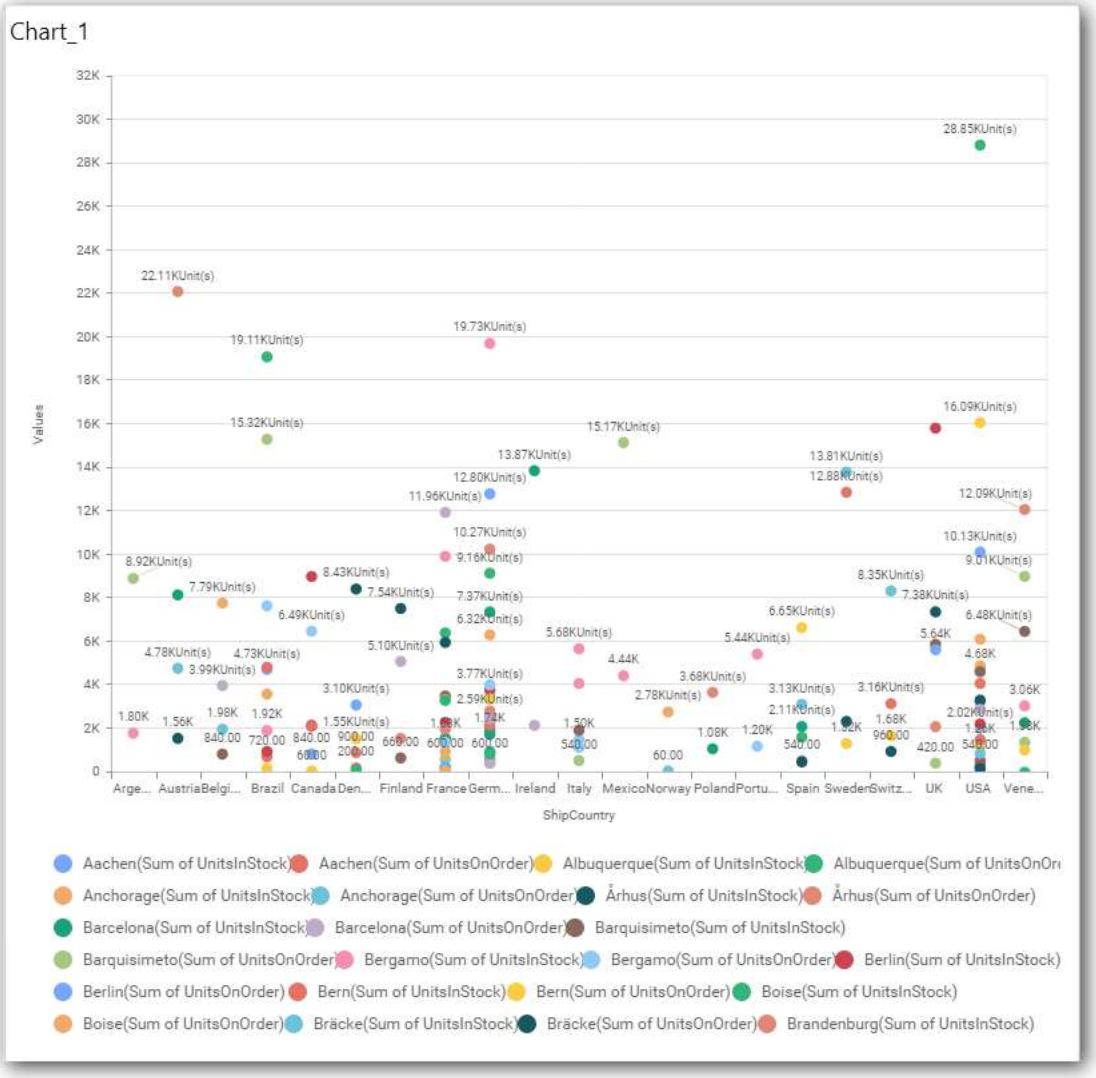
You can add **Measures** into **Column**

Assigning Row

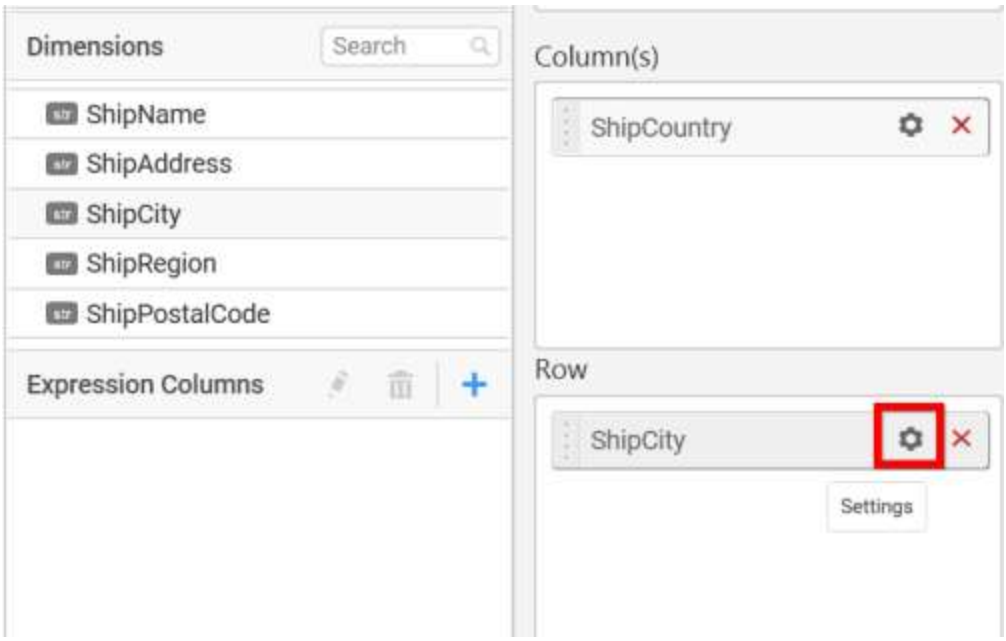
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s).

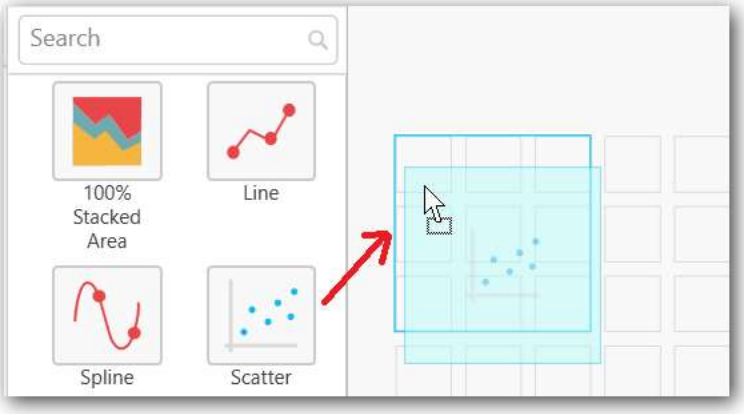


How to configure SSAS Data to Scatter Chart?

Scatter Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

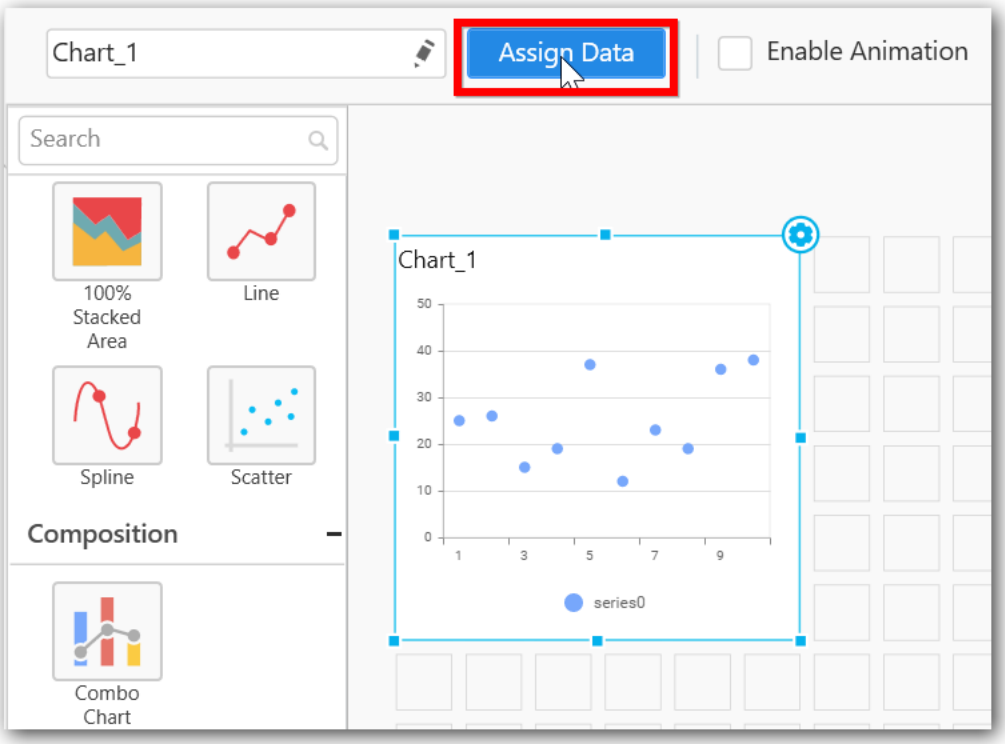
Following steps illustrates configuration of SSAS data to scatter chart

Drag and drop the Scatter chart widget into canvas and resize into your required size.

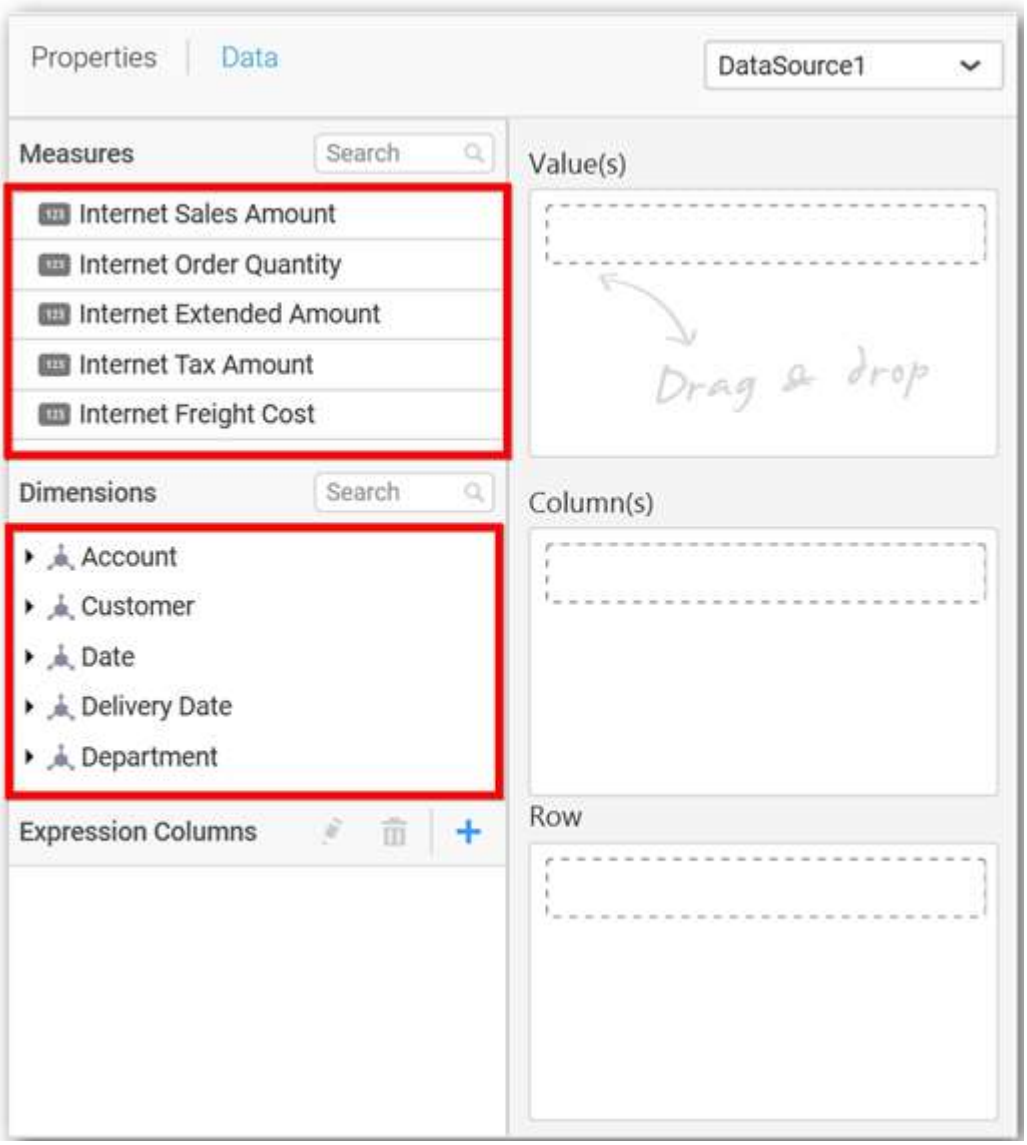


Select the dropped widget using mouse.

Click the Assign Data button in the toolbar.

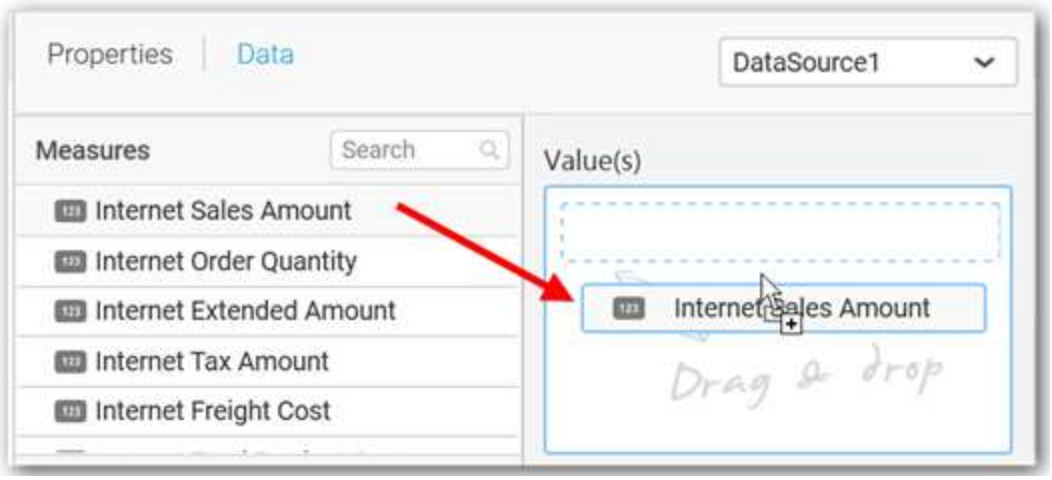


A Data pane will be opened with available Measures and Dimensions

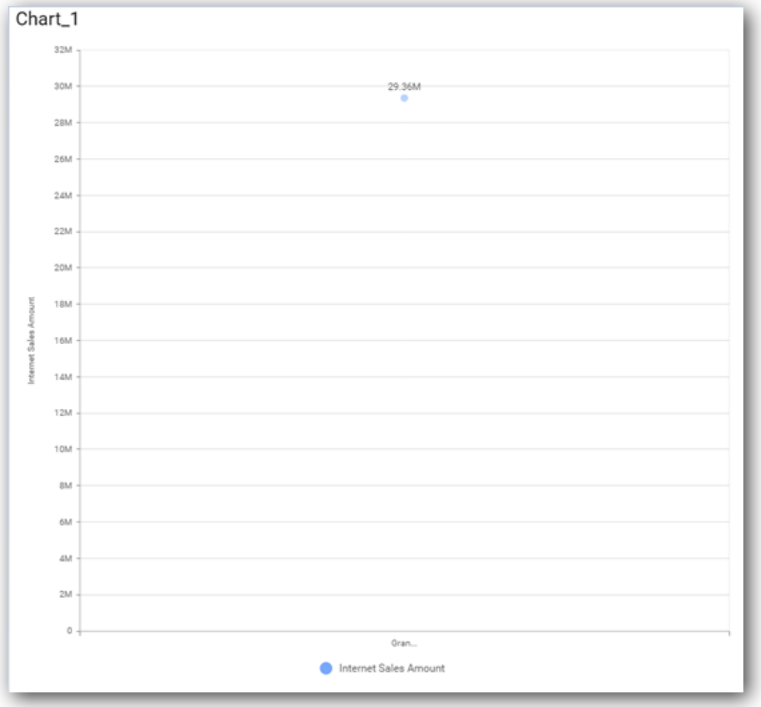


Assigning Value(s)

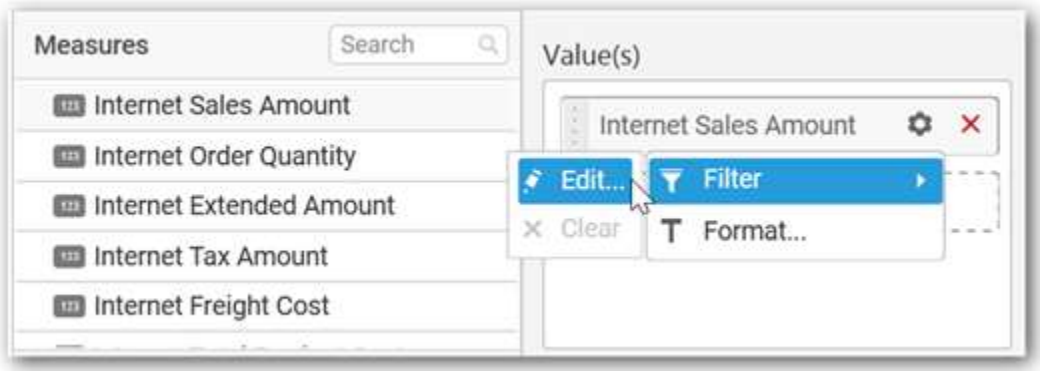
Drag and drop a column under Measures category into Value(s) section.



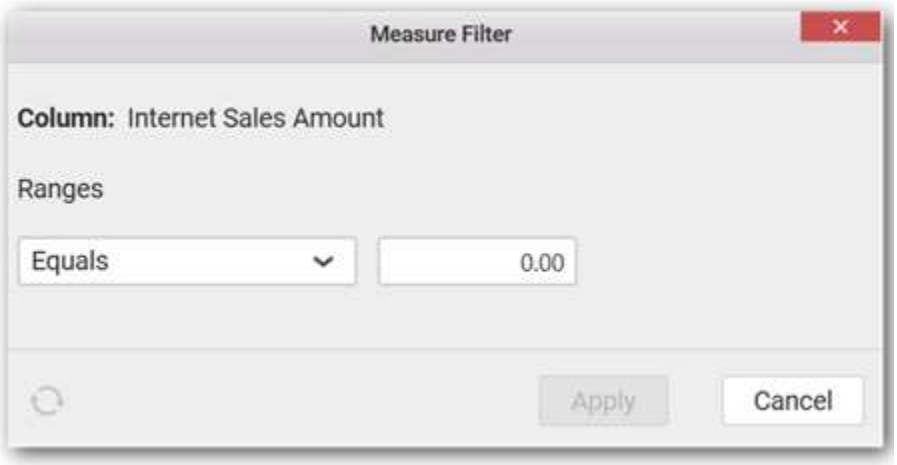
Now the chart will be rendered like this.



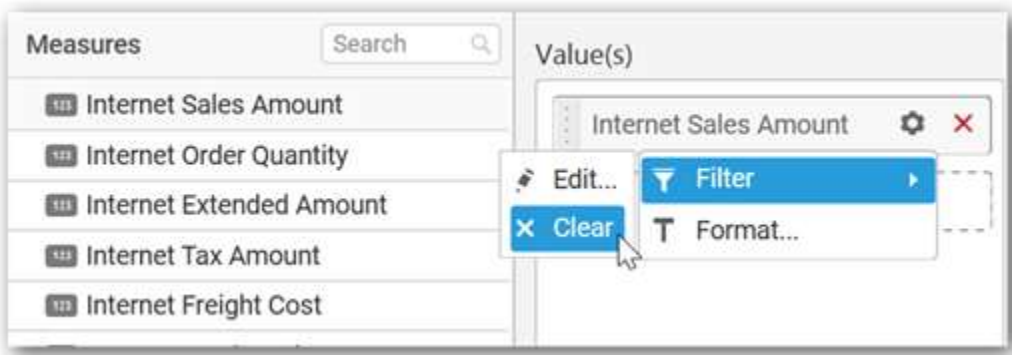
Define the filter criteria to match through choosing Edit option in Filter menu item.



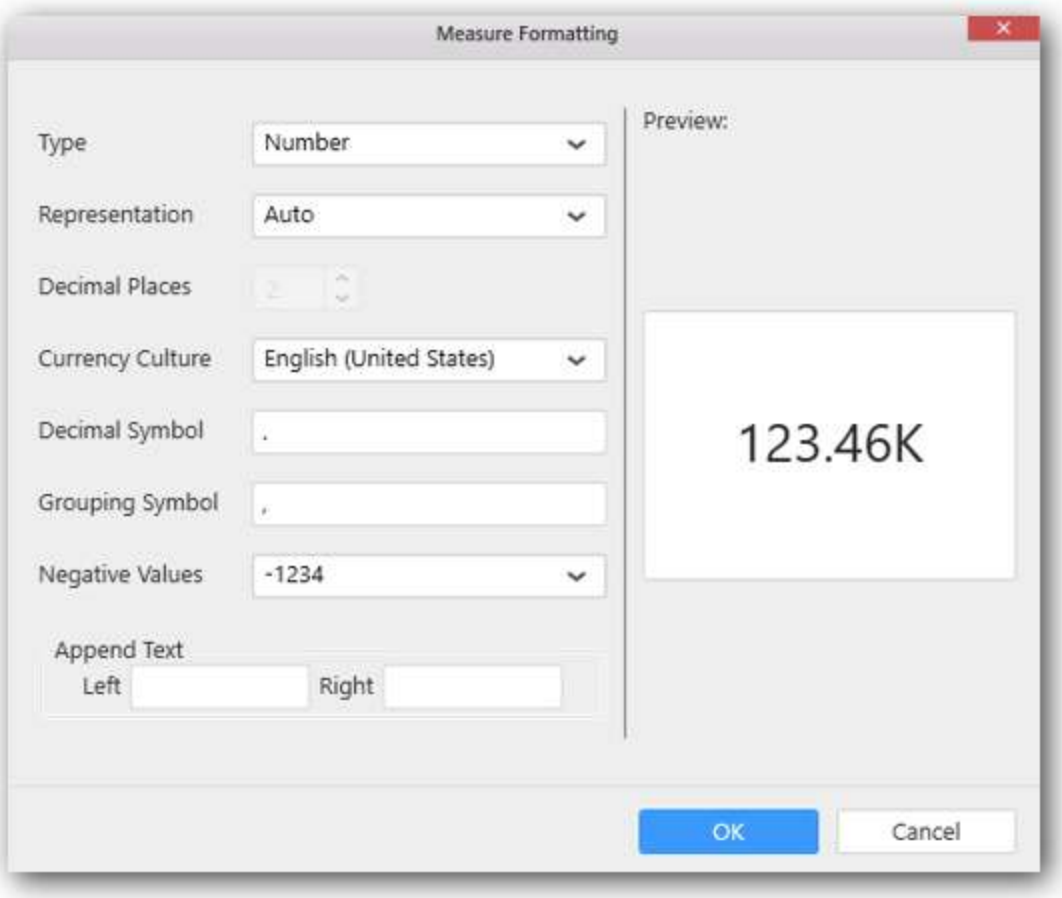
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



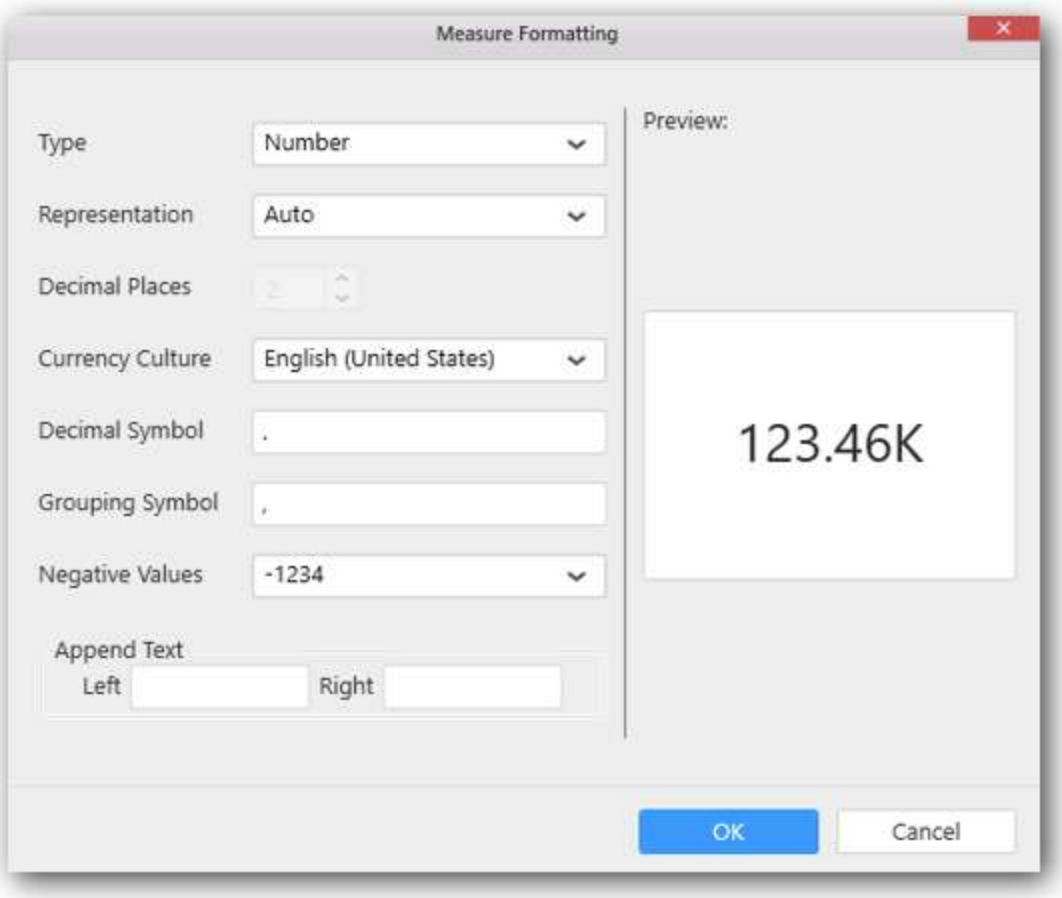
Select Clear option to clear the defined filter.



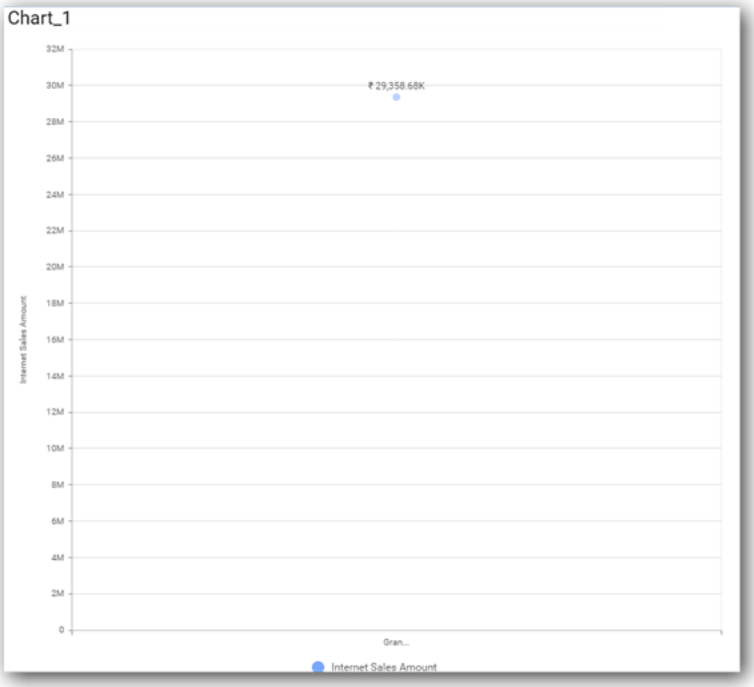
Select Format option to define the display format to the values in the column through Measure Formatting window.



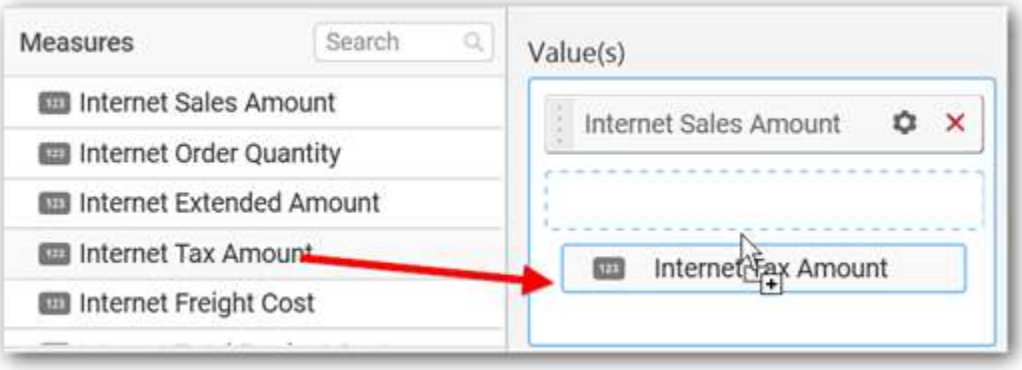
Choose the options you need and click OK.

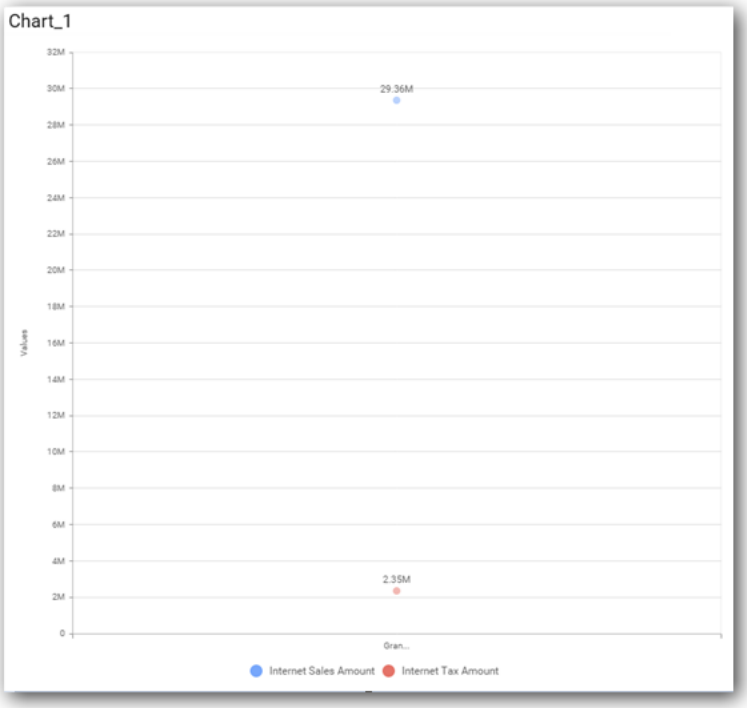


Now the Chart will be rendered like this.



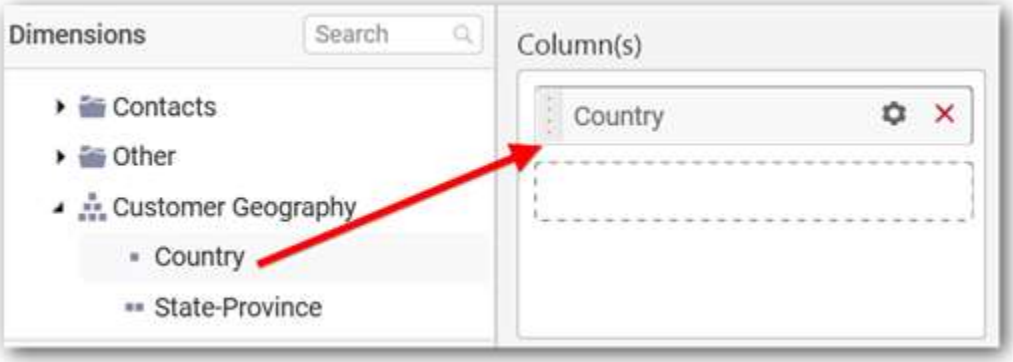
You can add more number values by drag drop the Measures into Value(s) field.

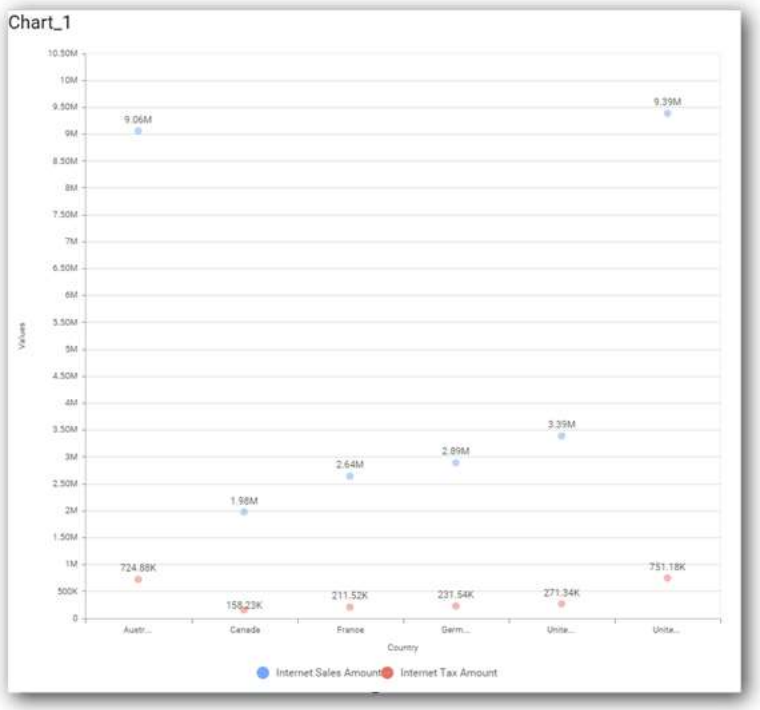




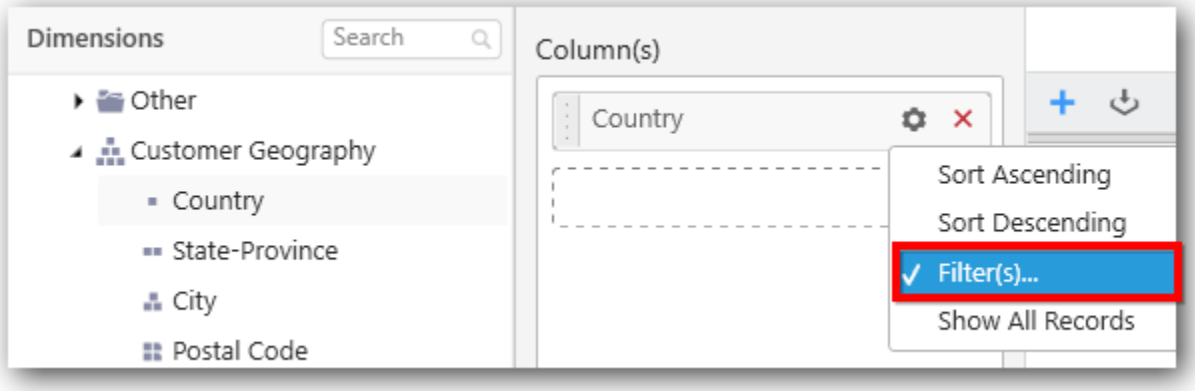
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.

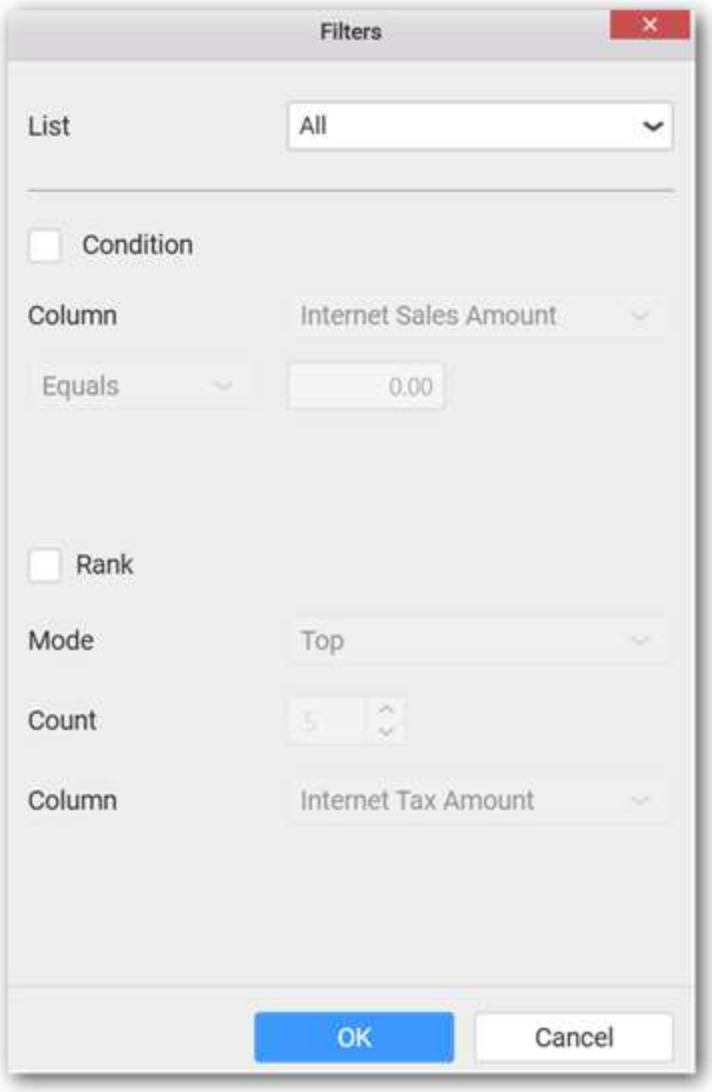




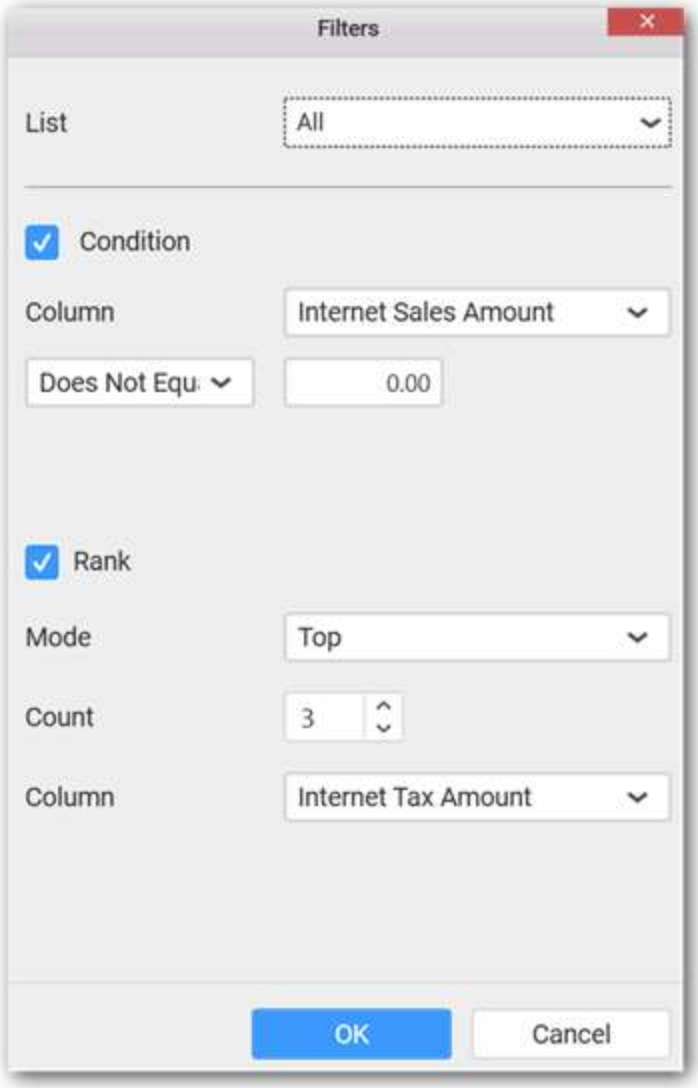
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



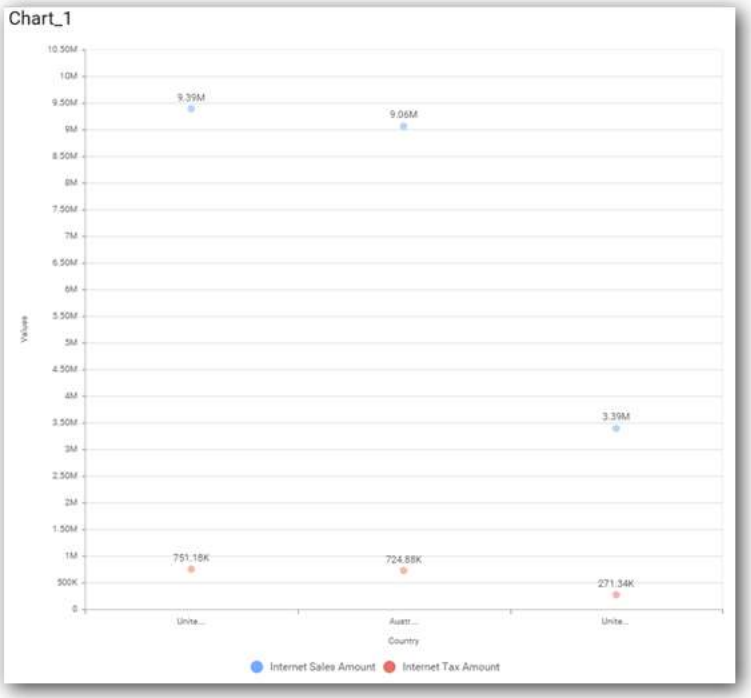
Select Filter(s)... to launch the Filters window.



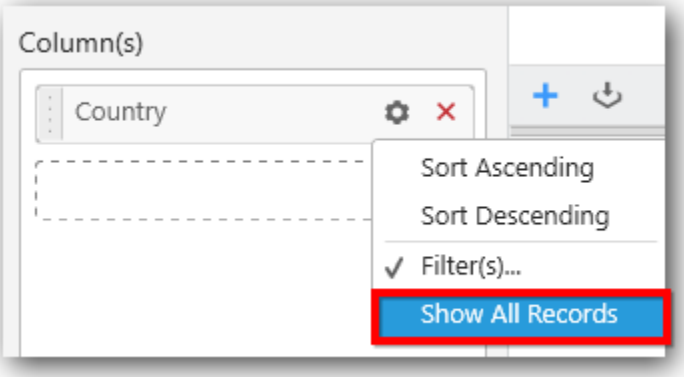
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

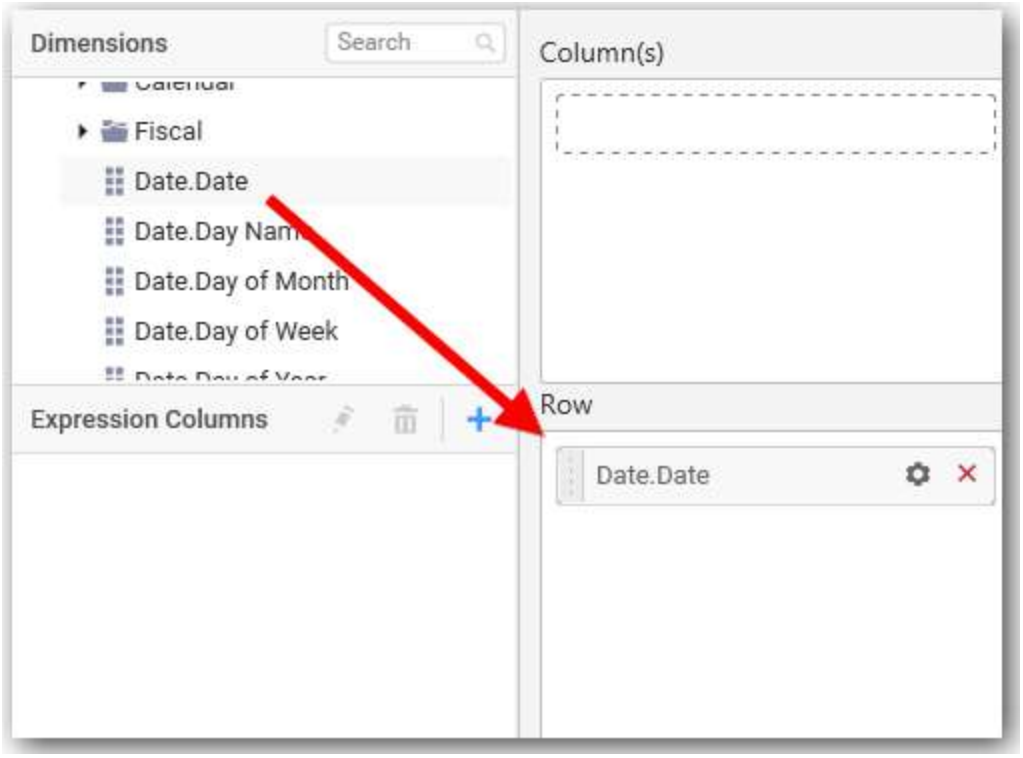


To show all records again click on Show All Records.



Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart



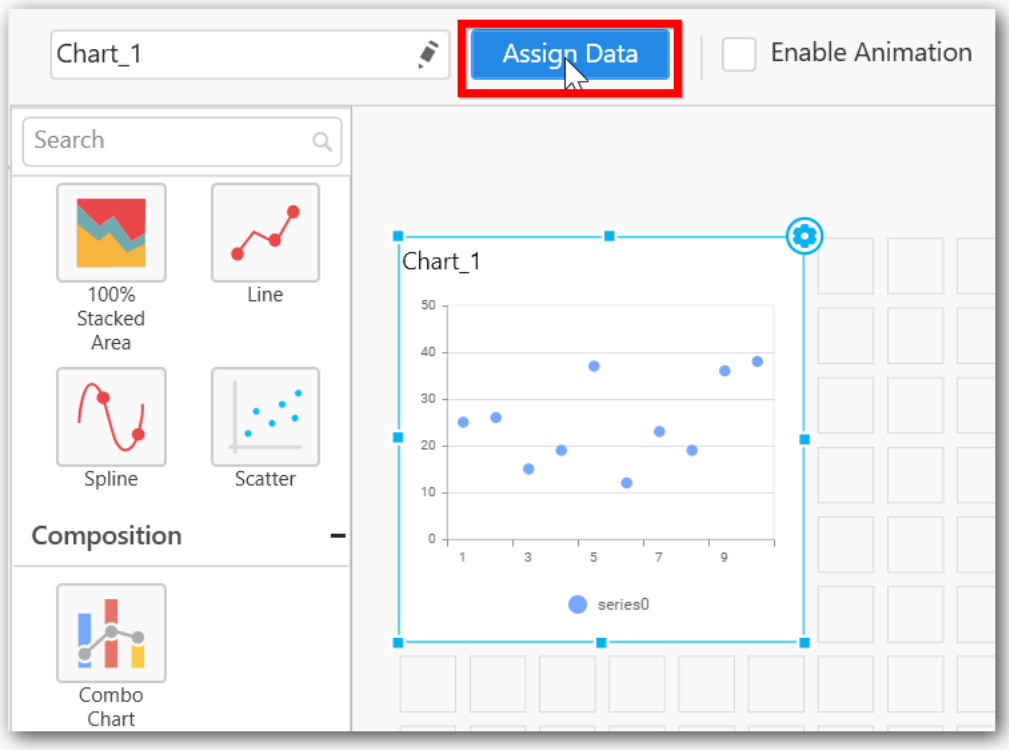
The chart will be rendered in series as shown in the image below.

How to format Scatter Chart?

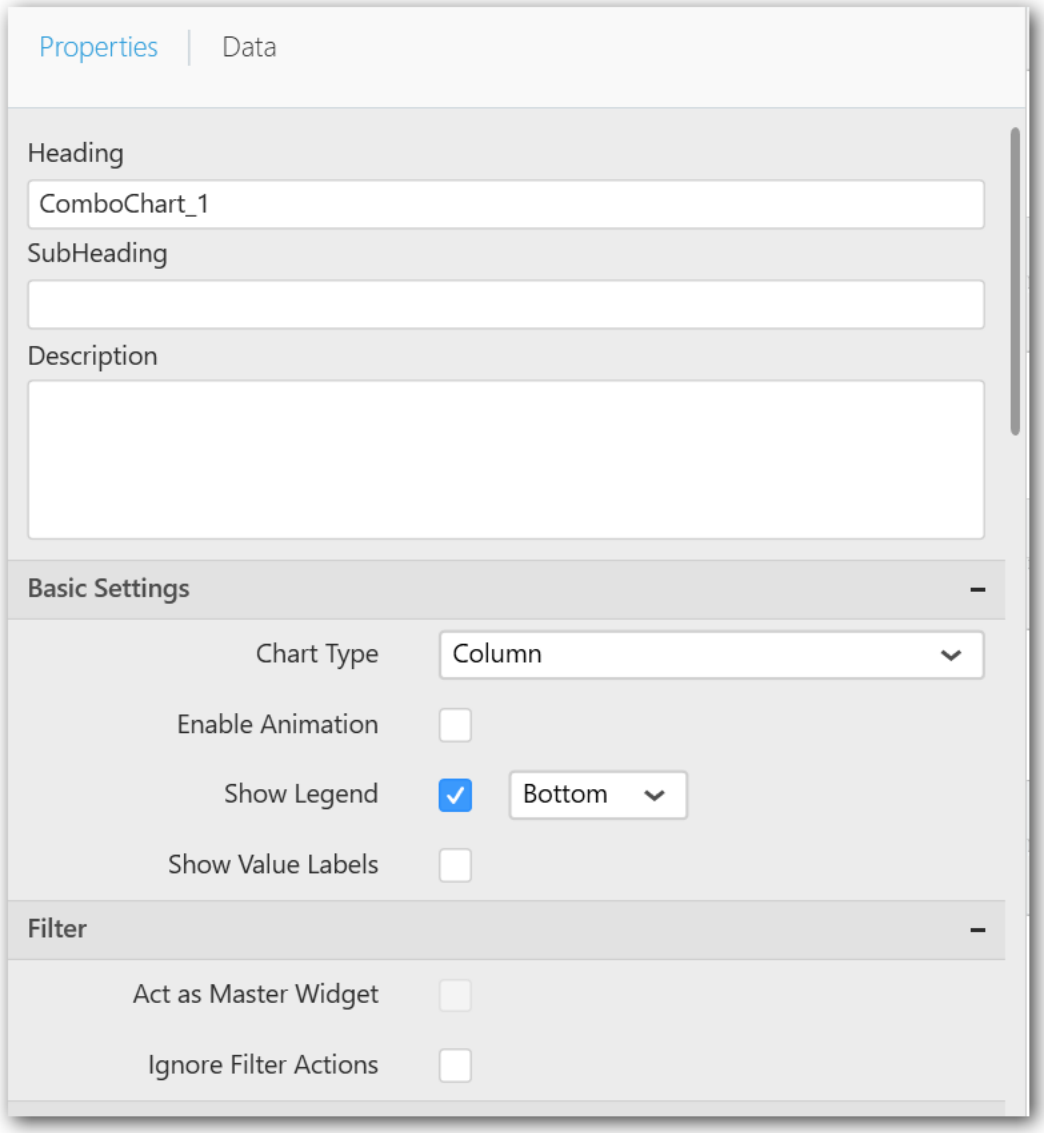
You can format the scatter chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into Scatter chart follow the steps

1. Drag and drop the Scatter chart into canvas and resize it to your required size.
2. Configure the data into Scatter chart.
3. Focus on the Scatter chart and Click on Widget Settings.

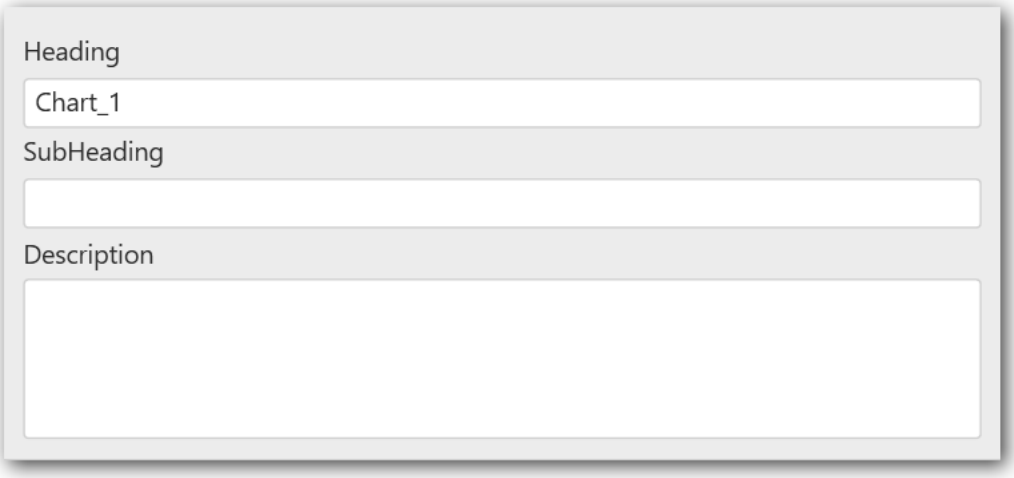


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this scatter chart widget.

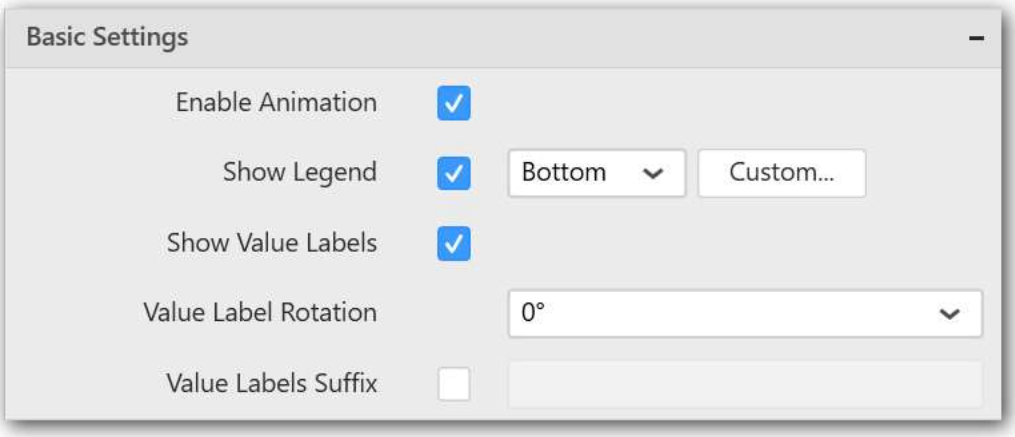
SubHeading

This allows you to set sub-title for this scatter chart widget.

Description

This allows you to set description for this scatter chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

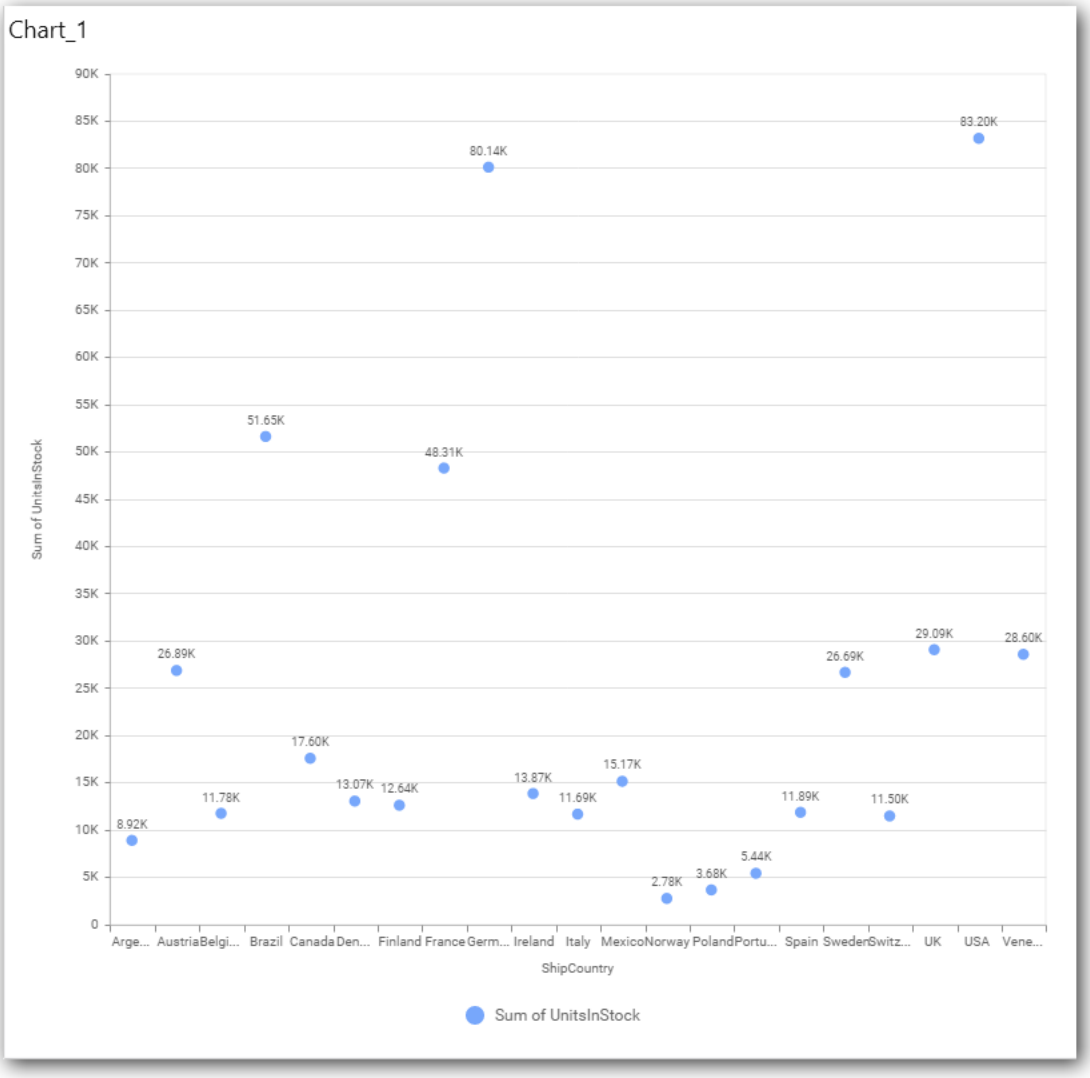


Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

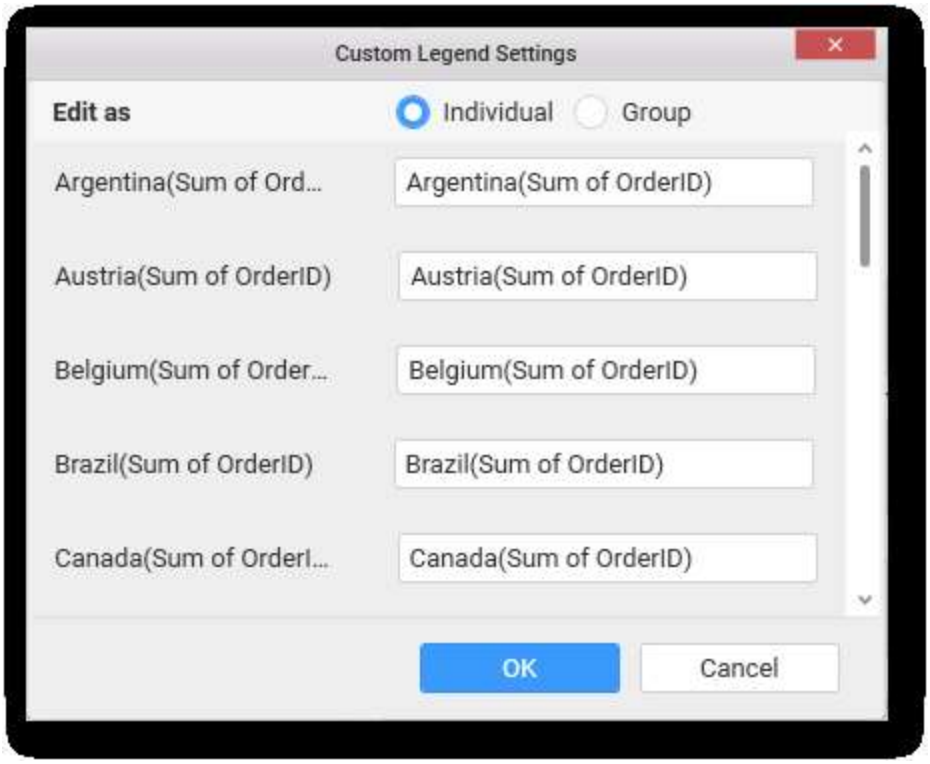
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

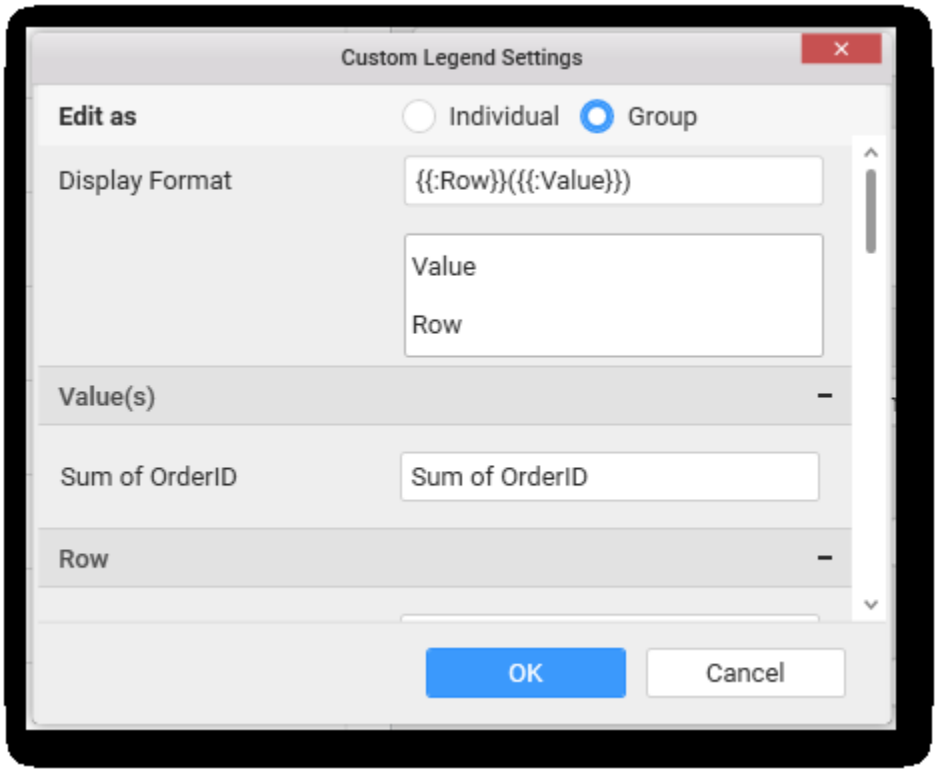
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

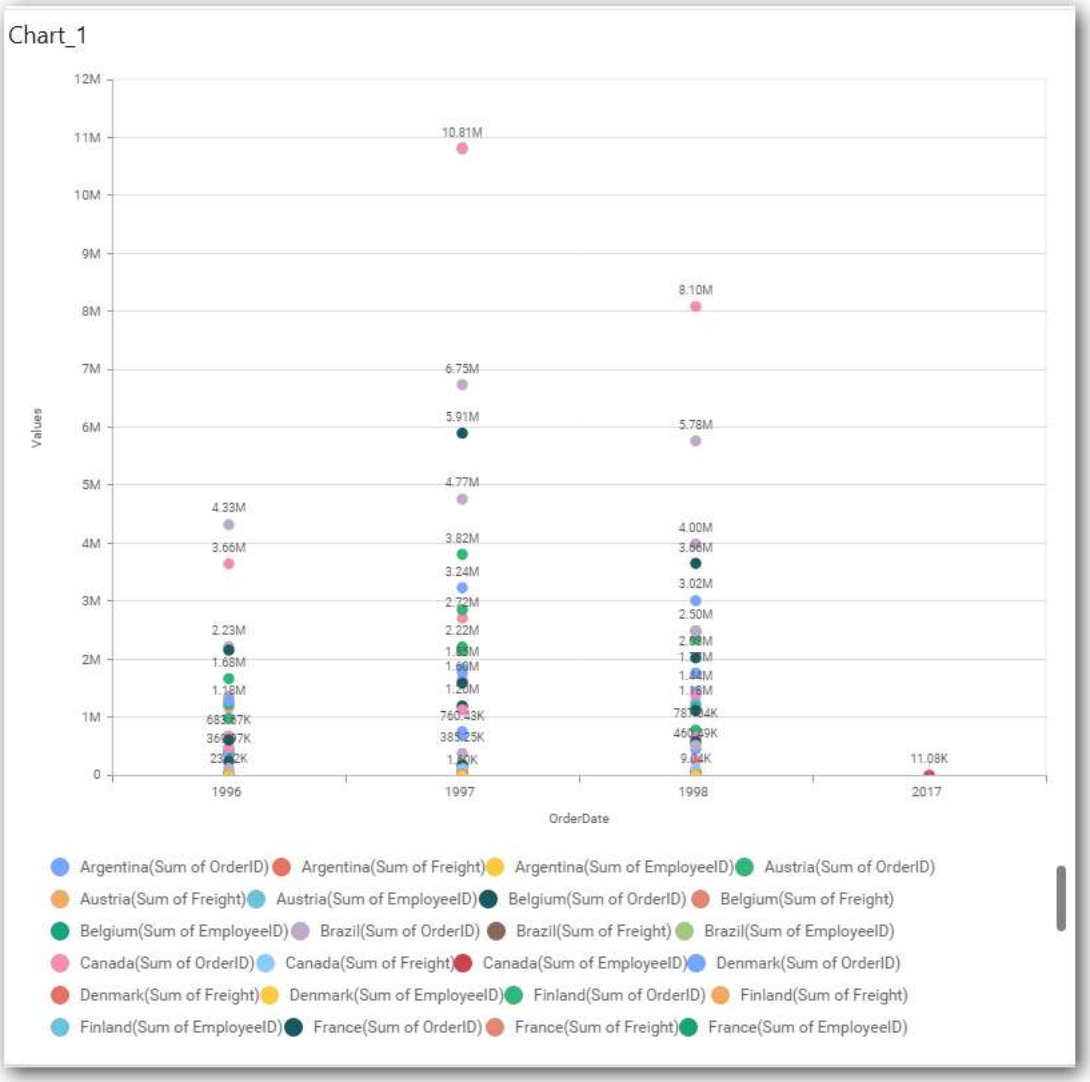


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

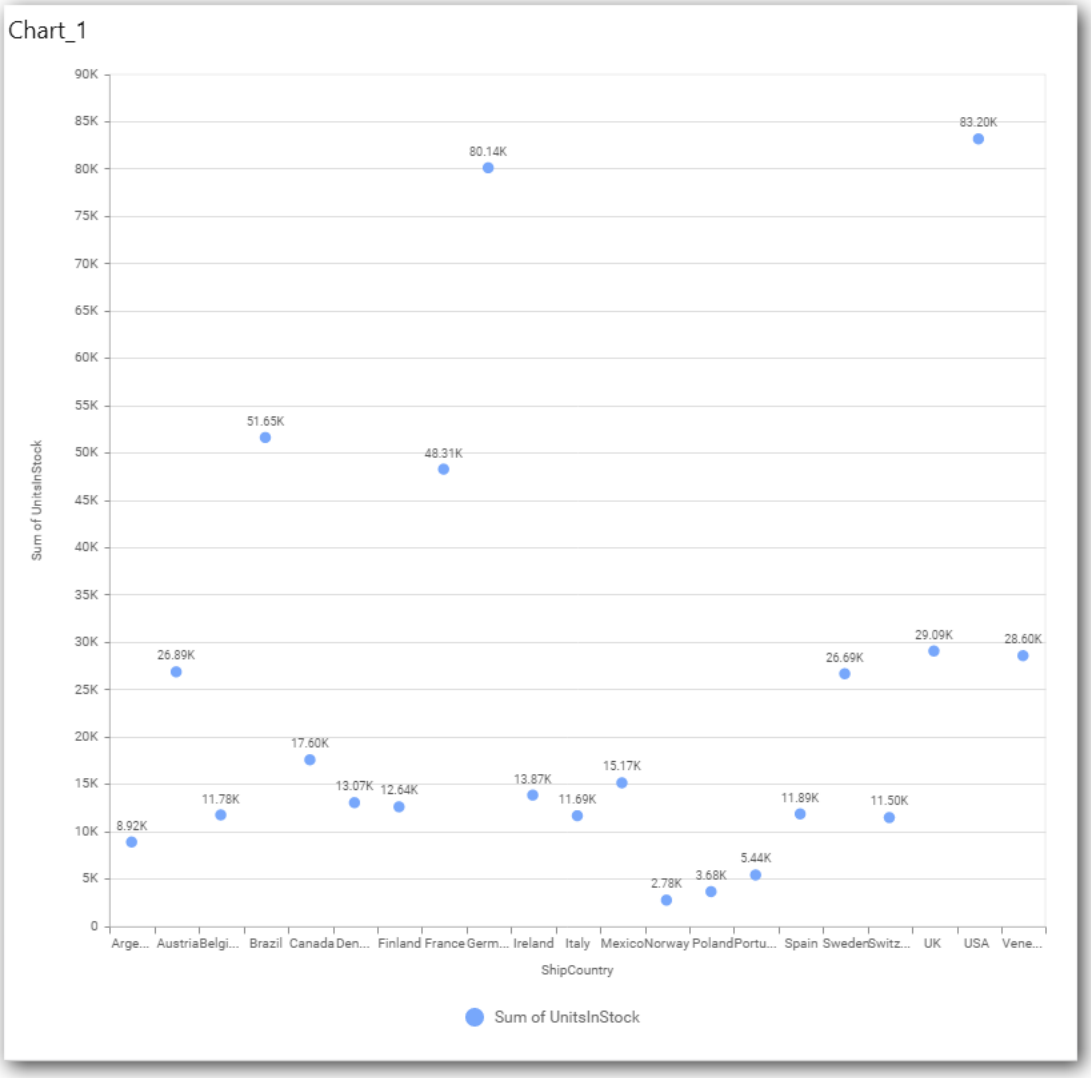


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



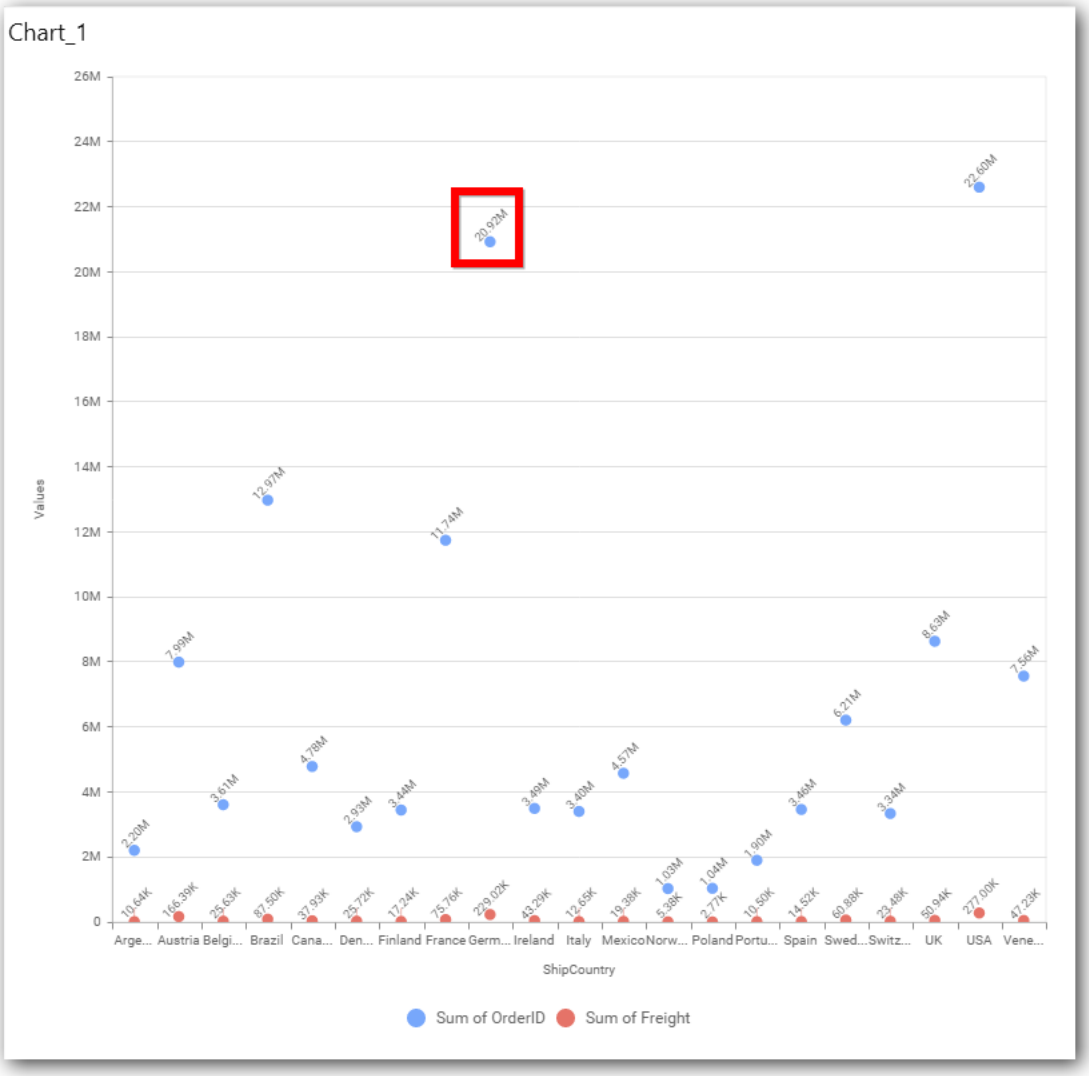
Show Value Labels

This allows you to toggle the visibility of value labels.



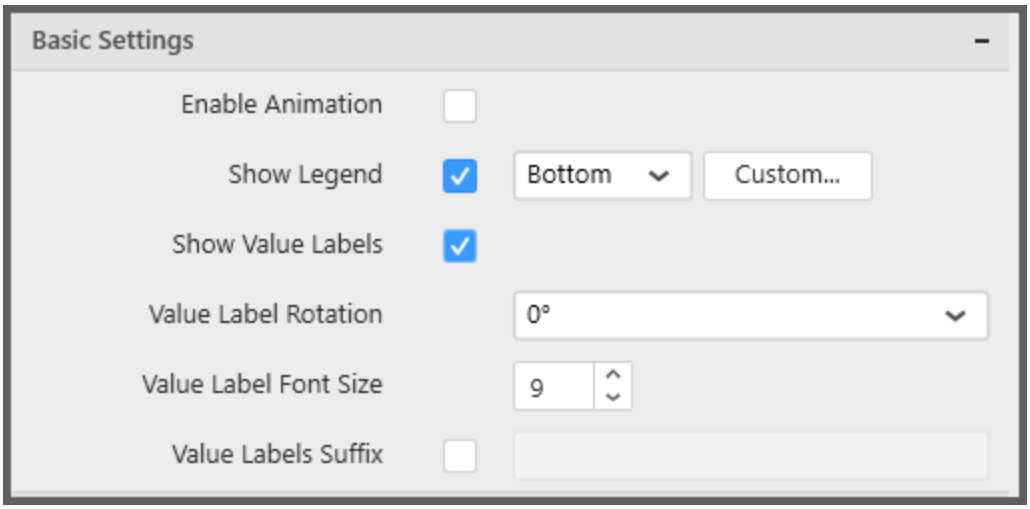
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



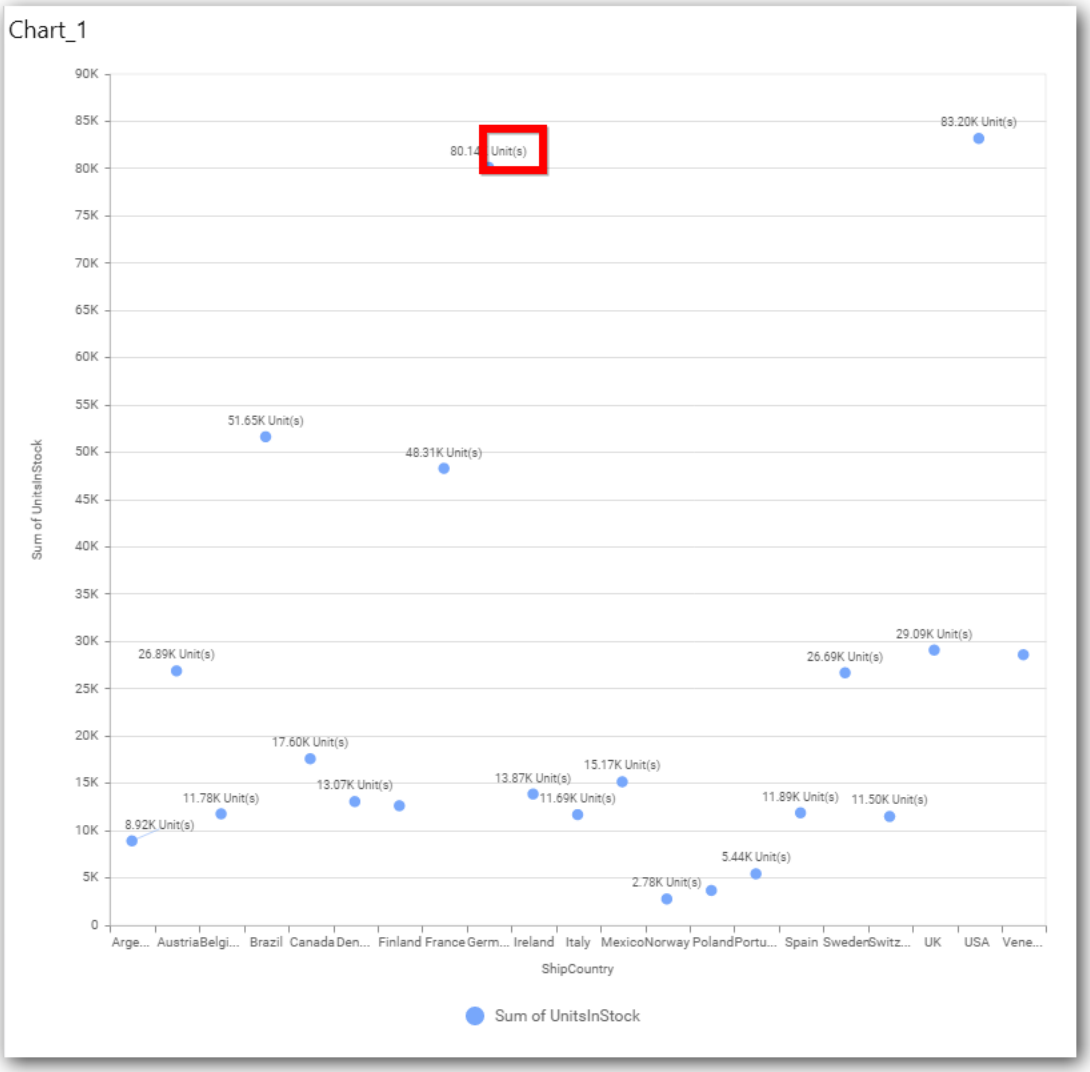
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

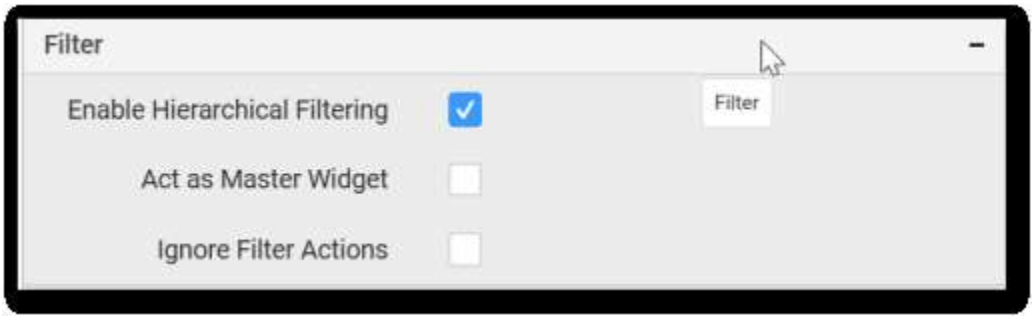


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

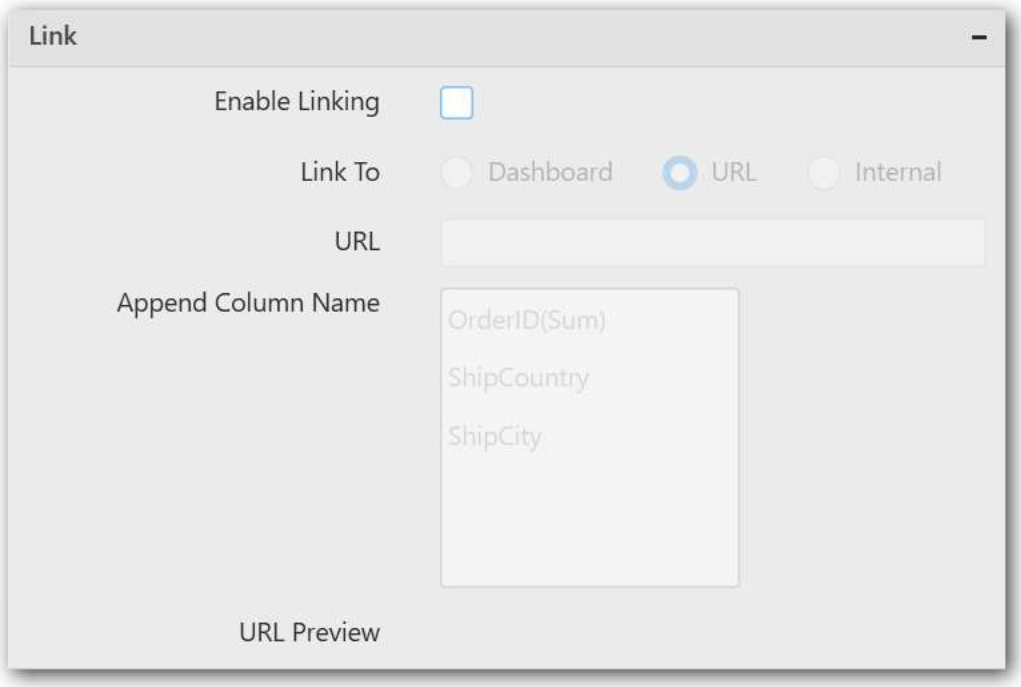
Act as Master Widget

This allows you to define this scatter chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

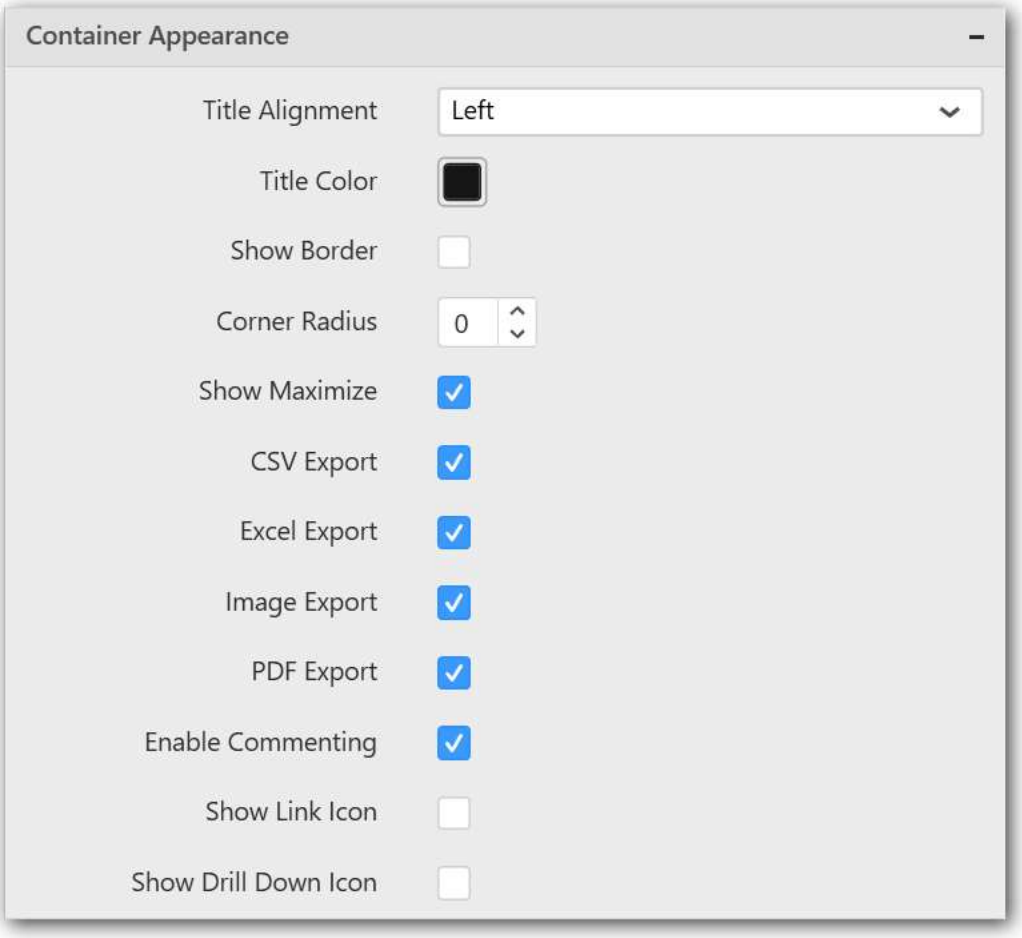
This allows you to define this scatter chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this scatter chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this scatter chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this scatter chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

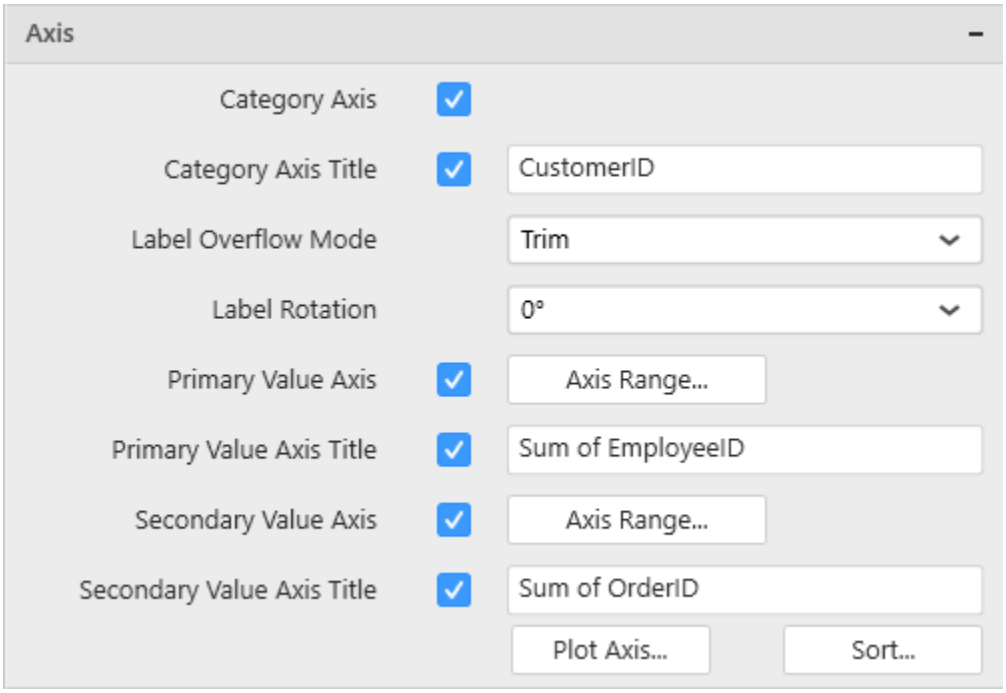
Image Export

This allows you to enable/disable the image export option for this scatter chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

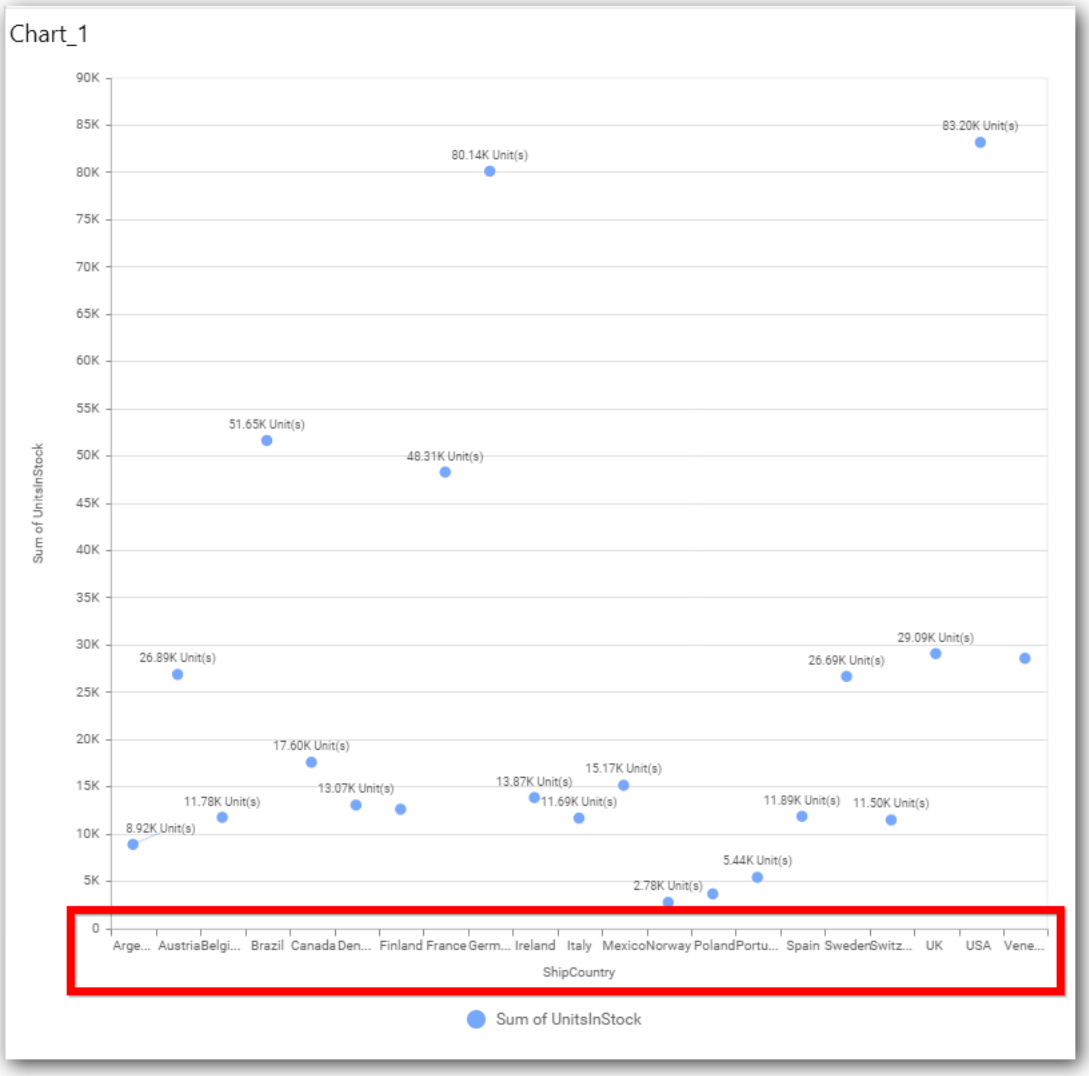
Axis Settings



This section allows you to customize the axis settings in chart.

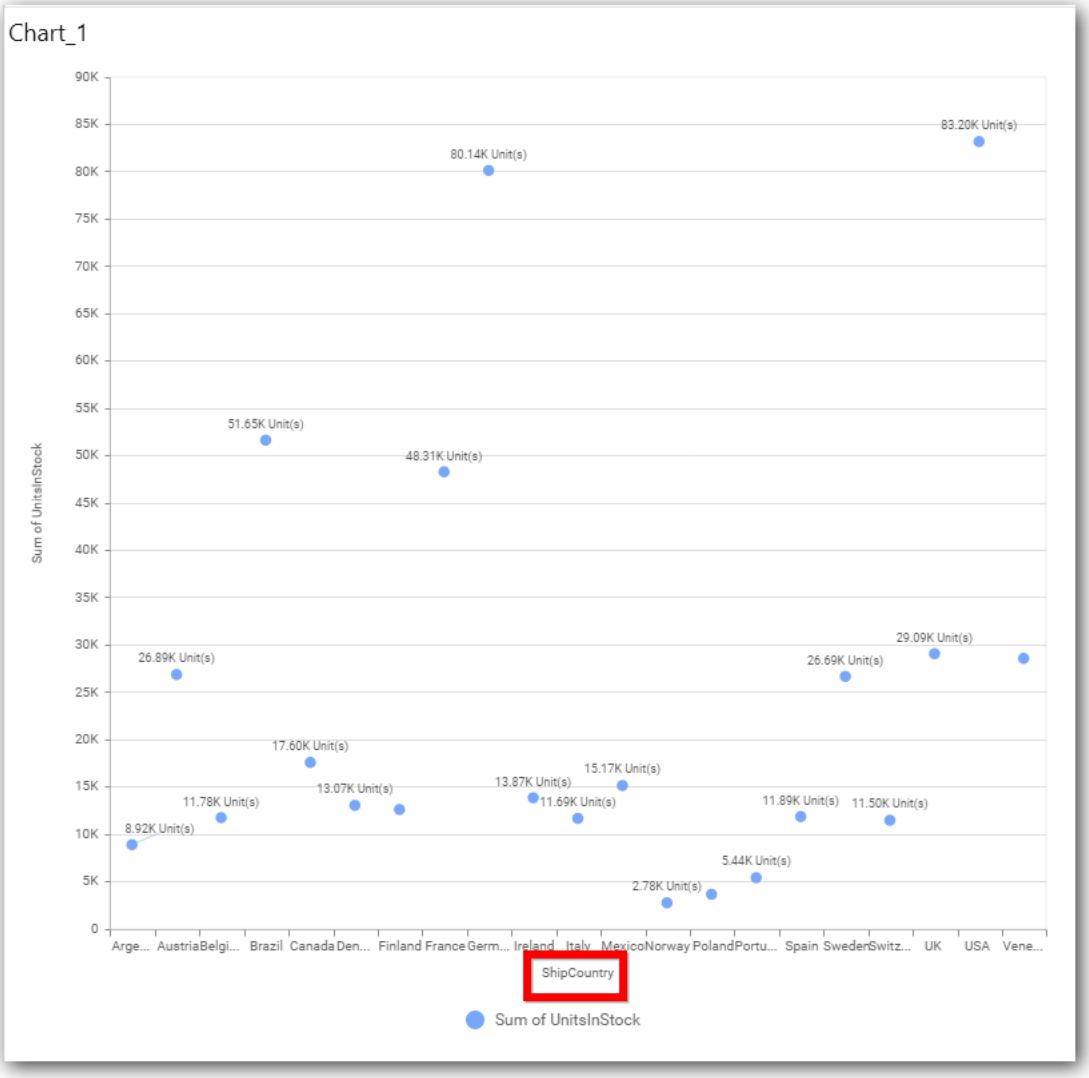
Category Axis

This allows to enable/edit the option of Category Axis. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

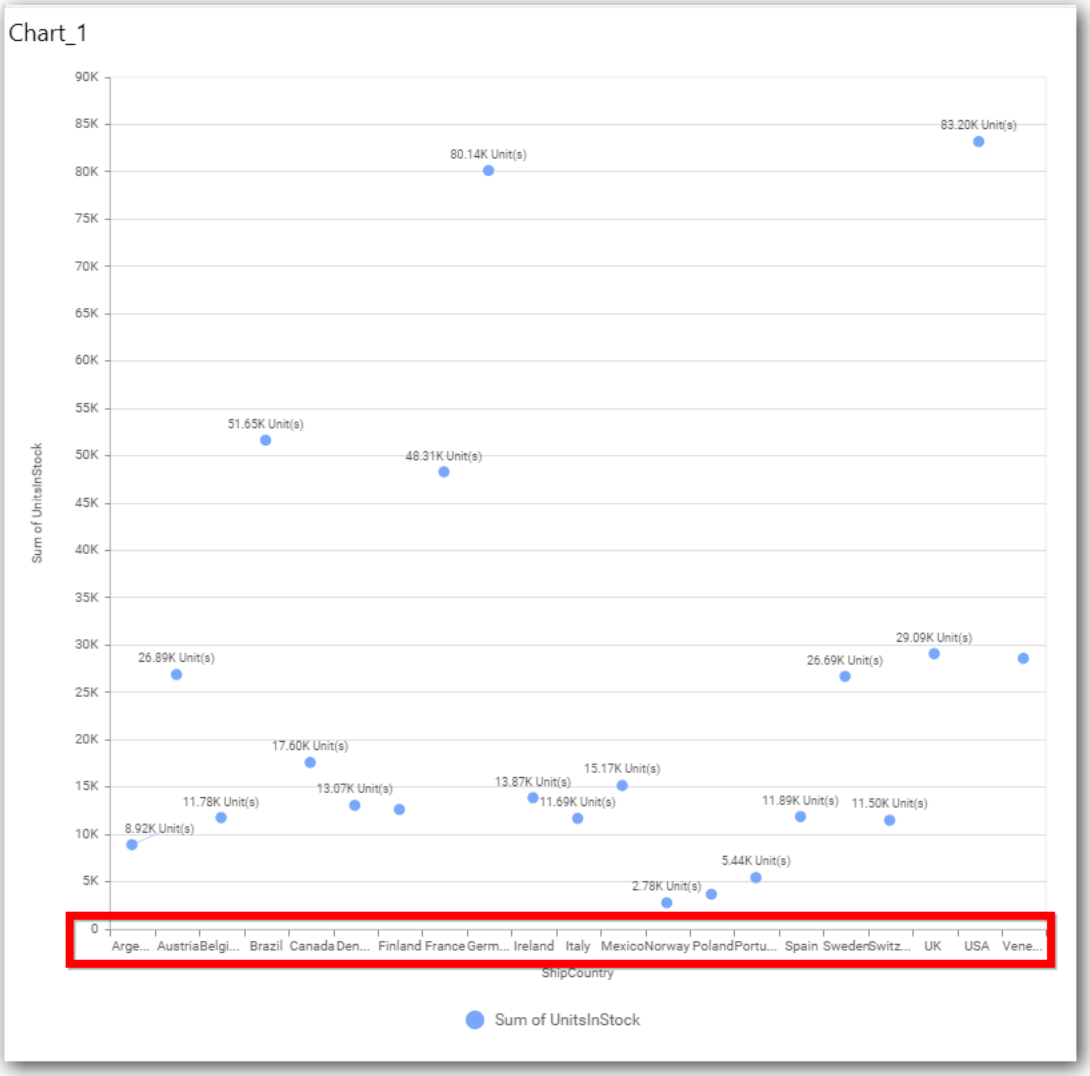


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

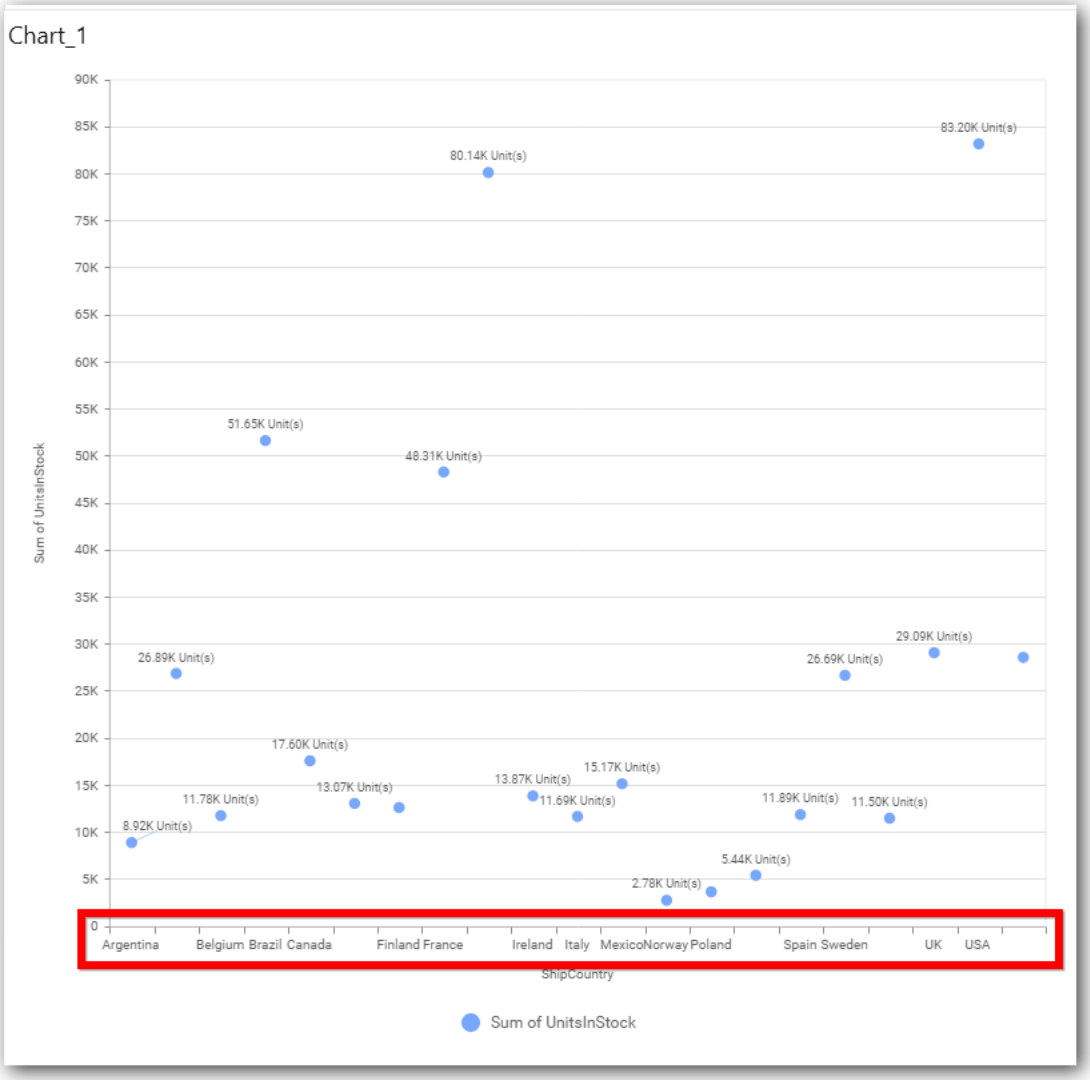
Trim

This option trims the end of overlapping label in the axis.



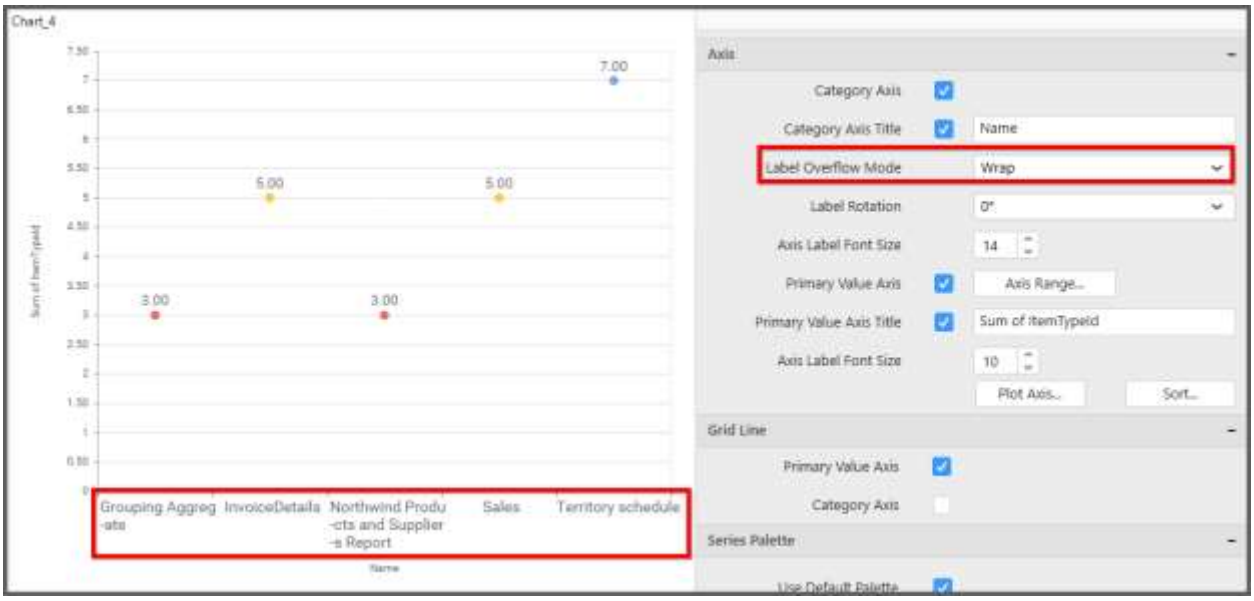
Hide

This option hides the overlapping label in the axis.



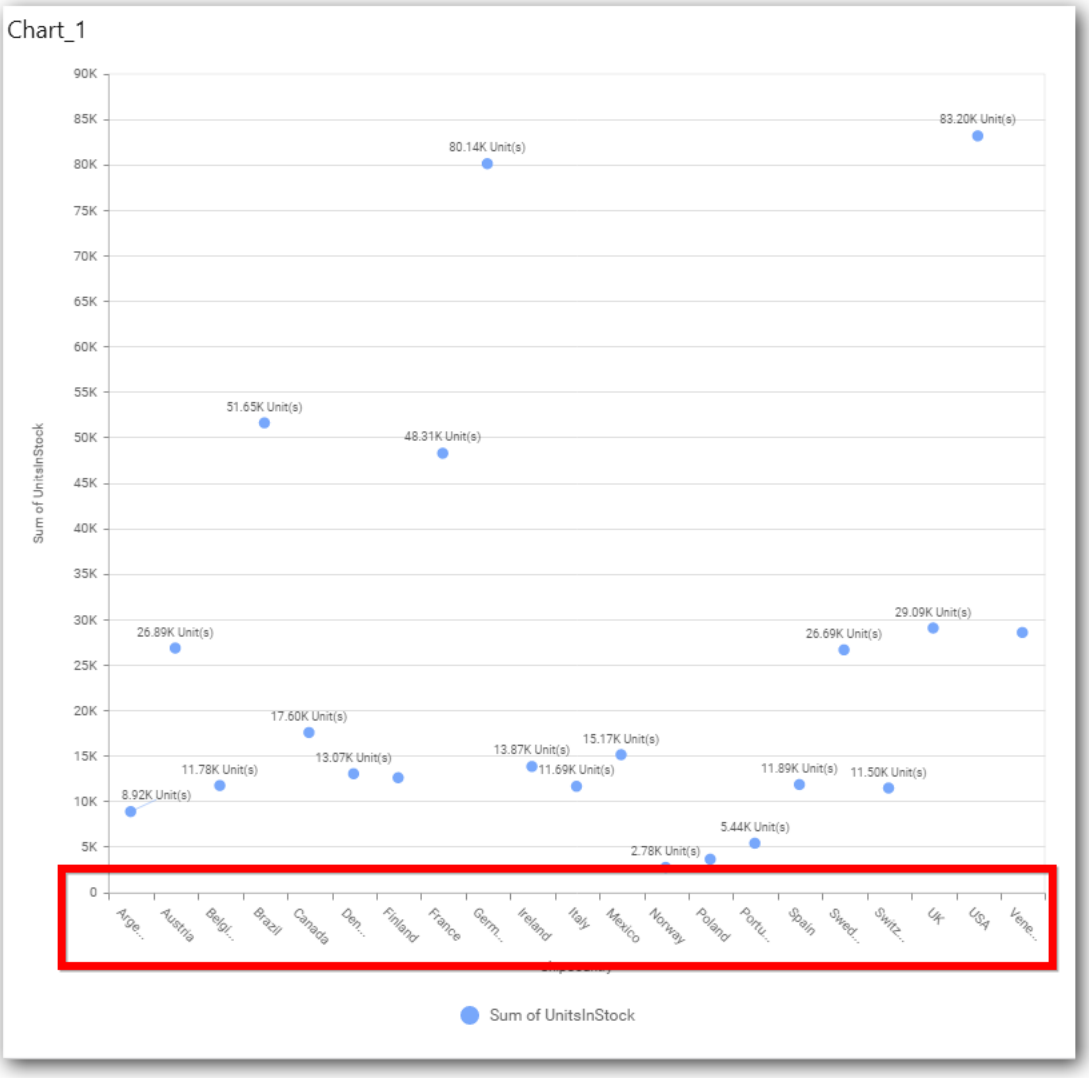
Wrap

This option wraps the lengthy label text in the axis.



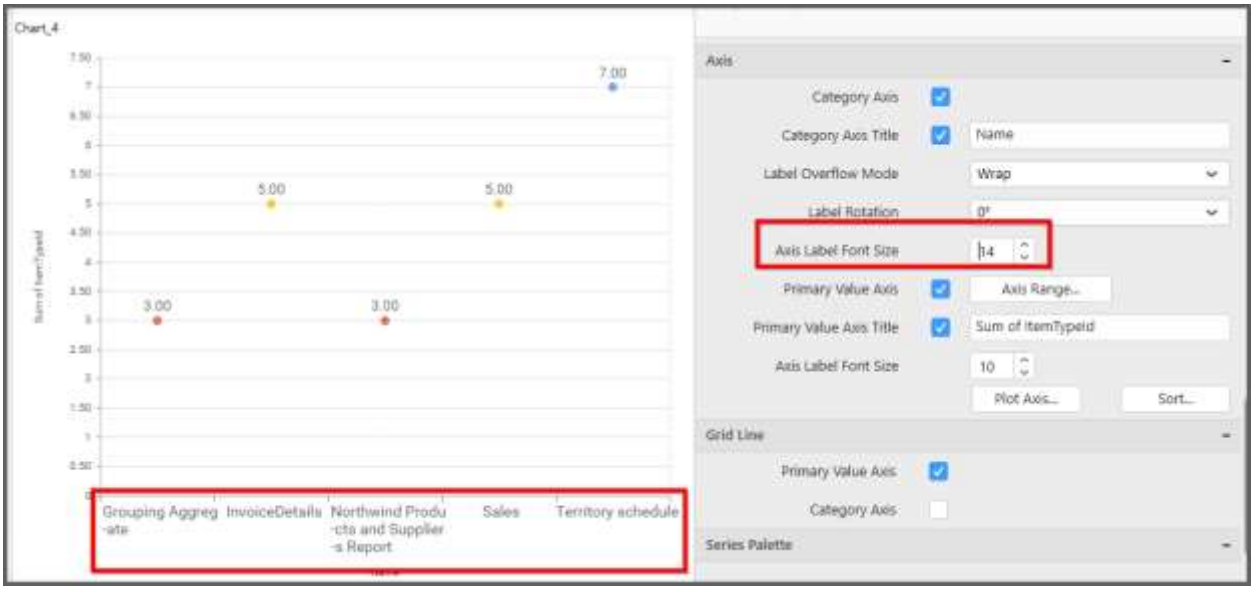
Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



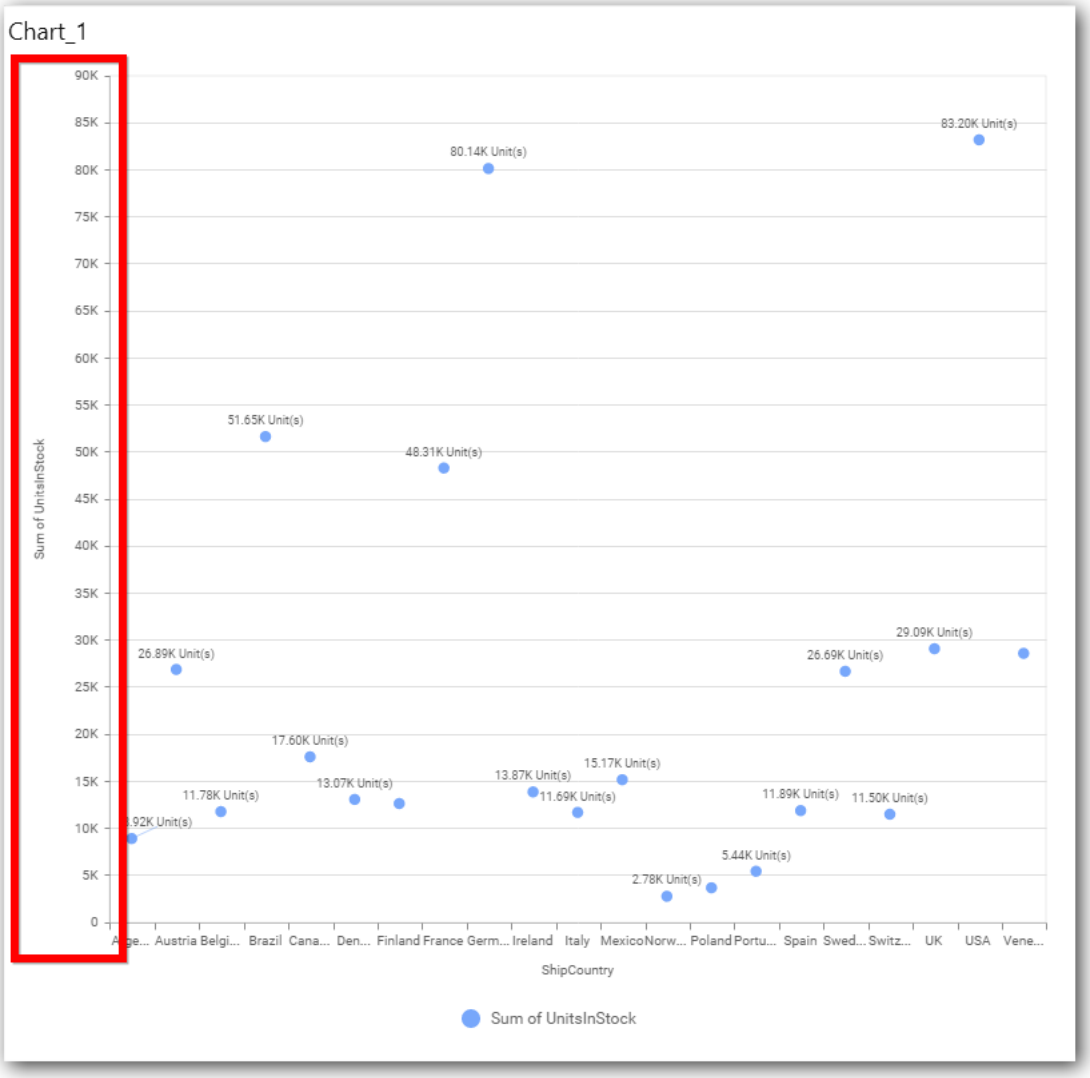
Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.



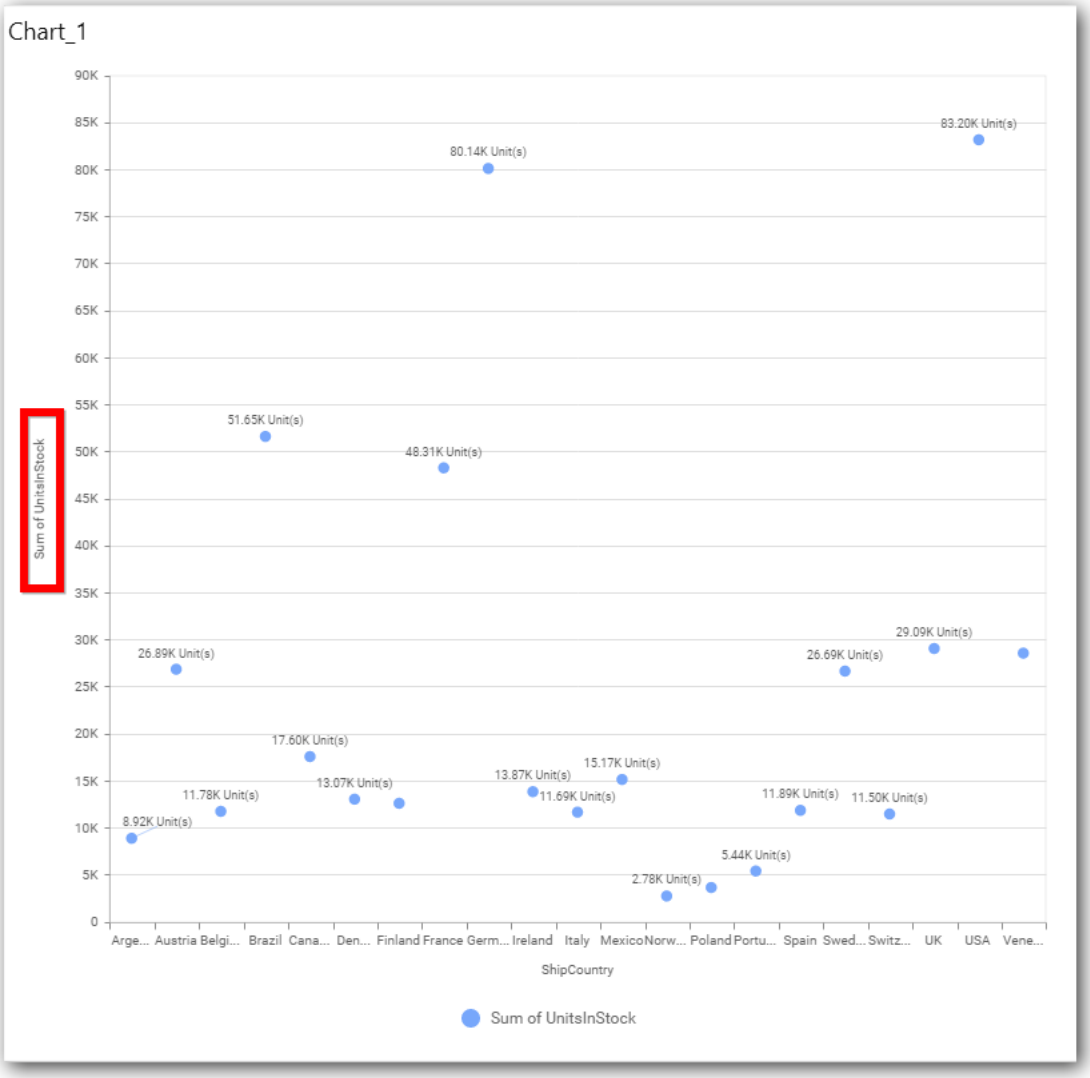
Primary Value Axis

This allows you to enable/edit the option of Primary Value Axis. It will reflect in chart area y-axis name.



Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

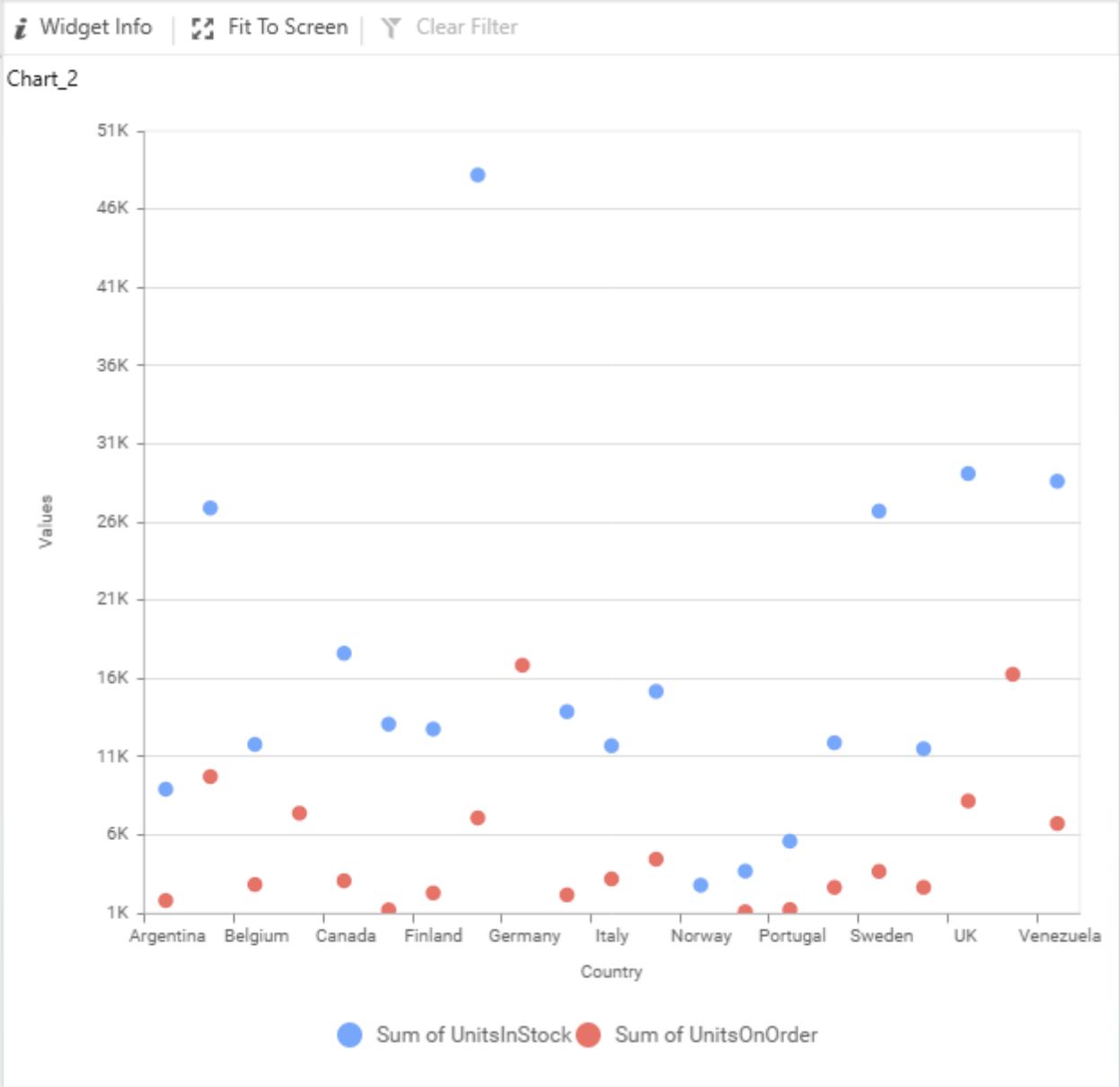
This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

Primary Value Axis Range

This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.



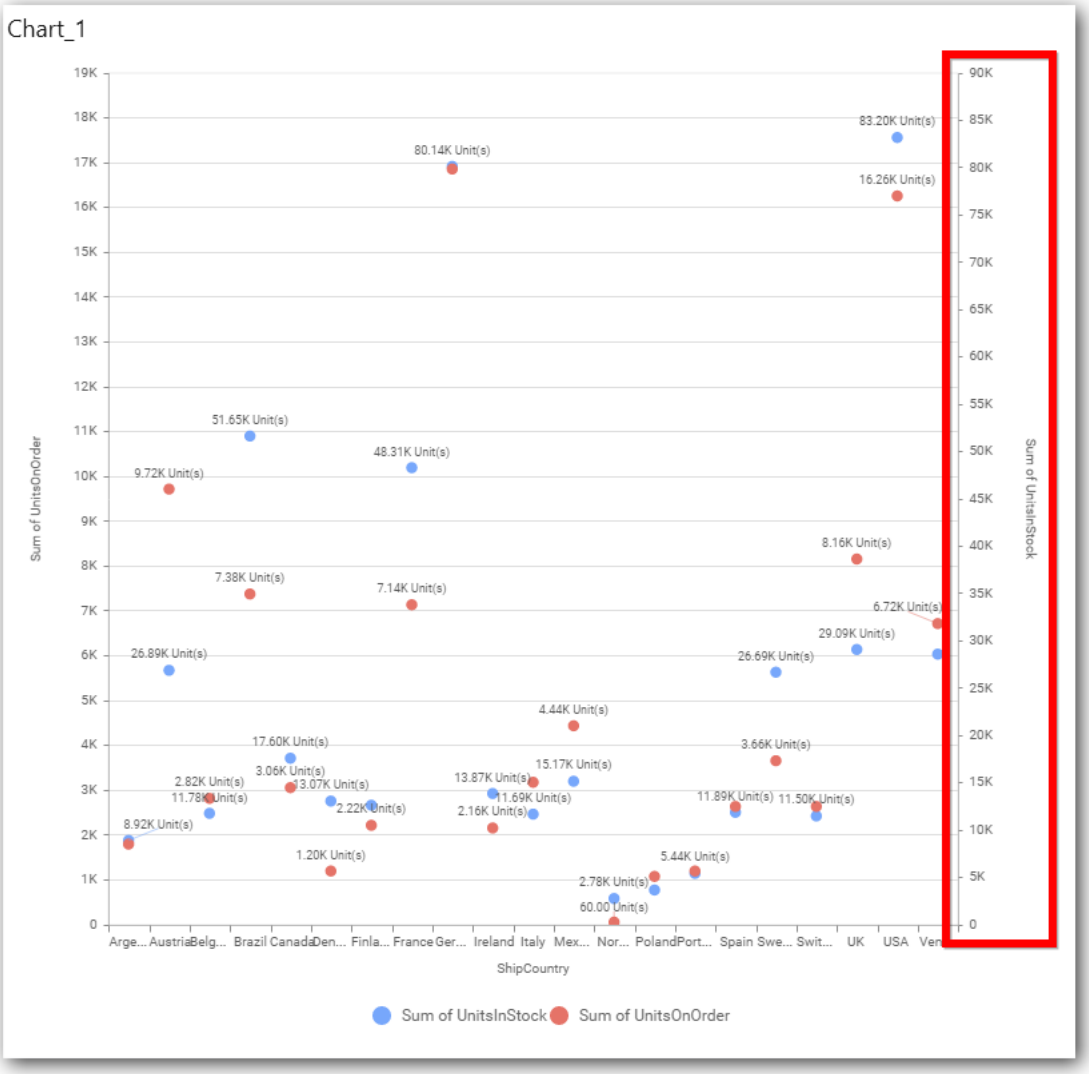
Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

![Chart illustration

](images/scatterchart_primaryvalueaxisrange.png)

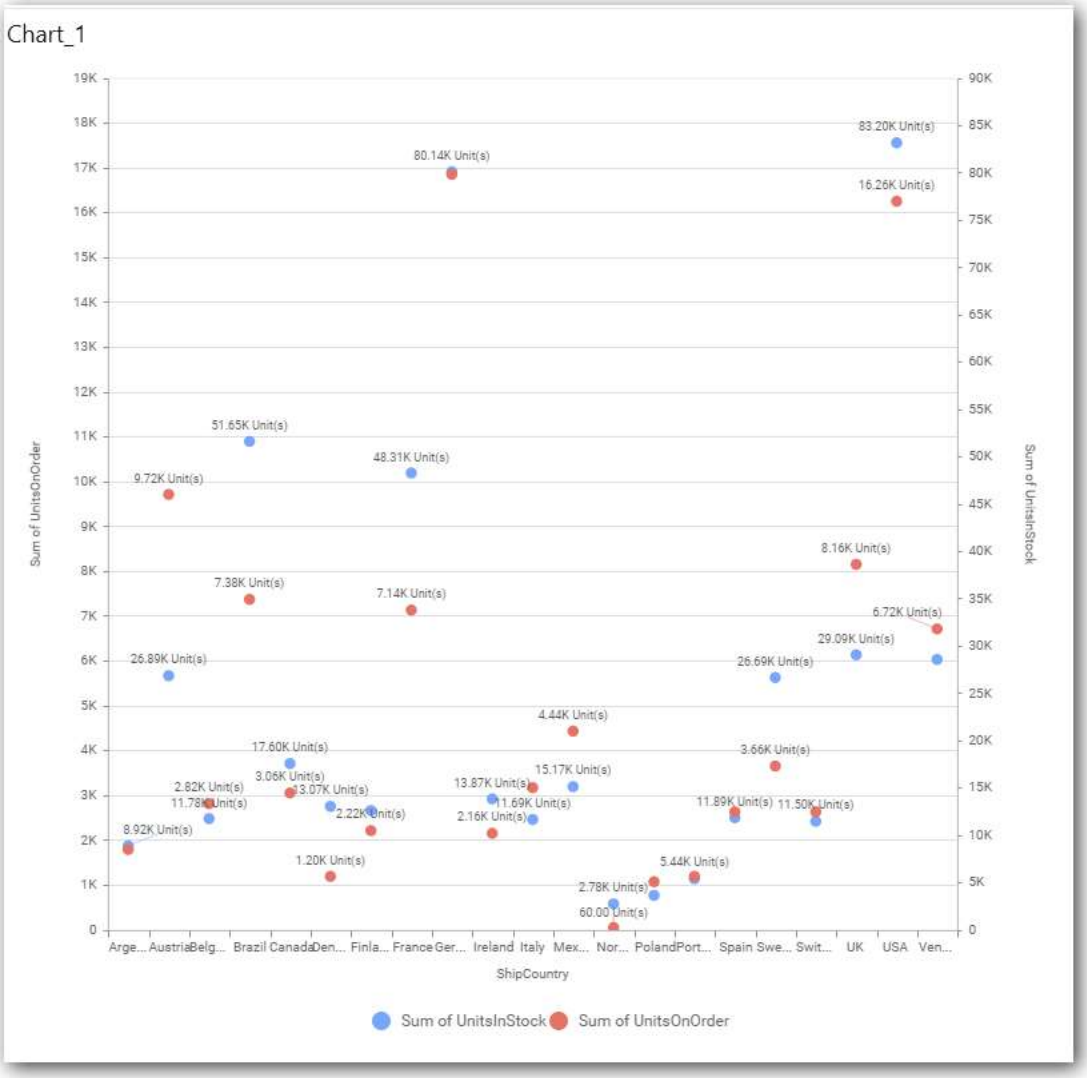
Secondary Value Axis

This allows you to enable/edit the option of Secondary Value Axis. It will reflect in chart area secondary y-axis name.



Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

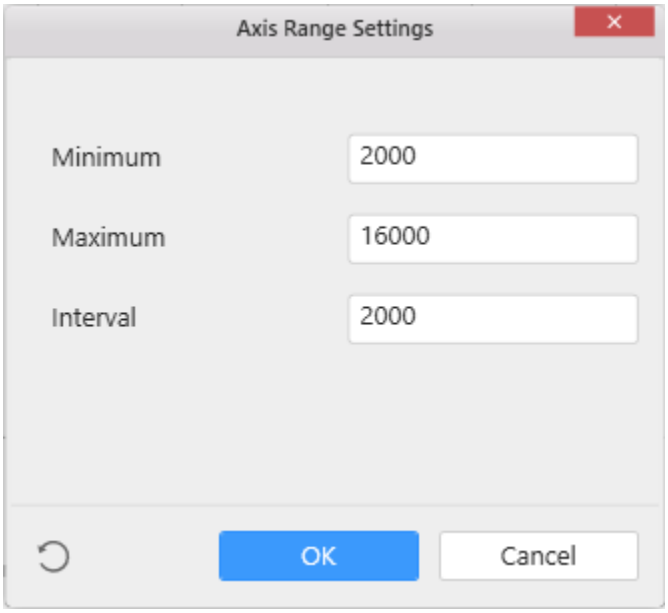


Axis Label Size

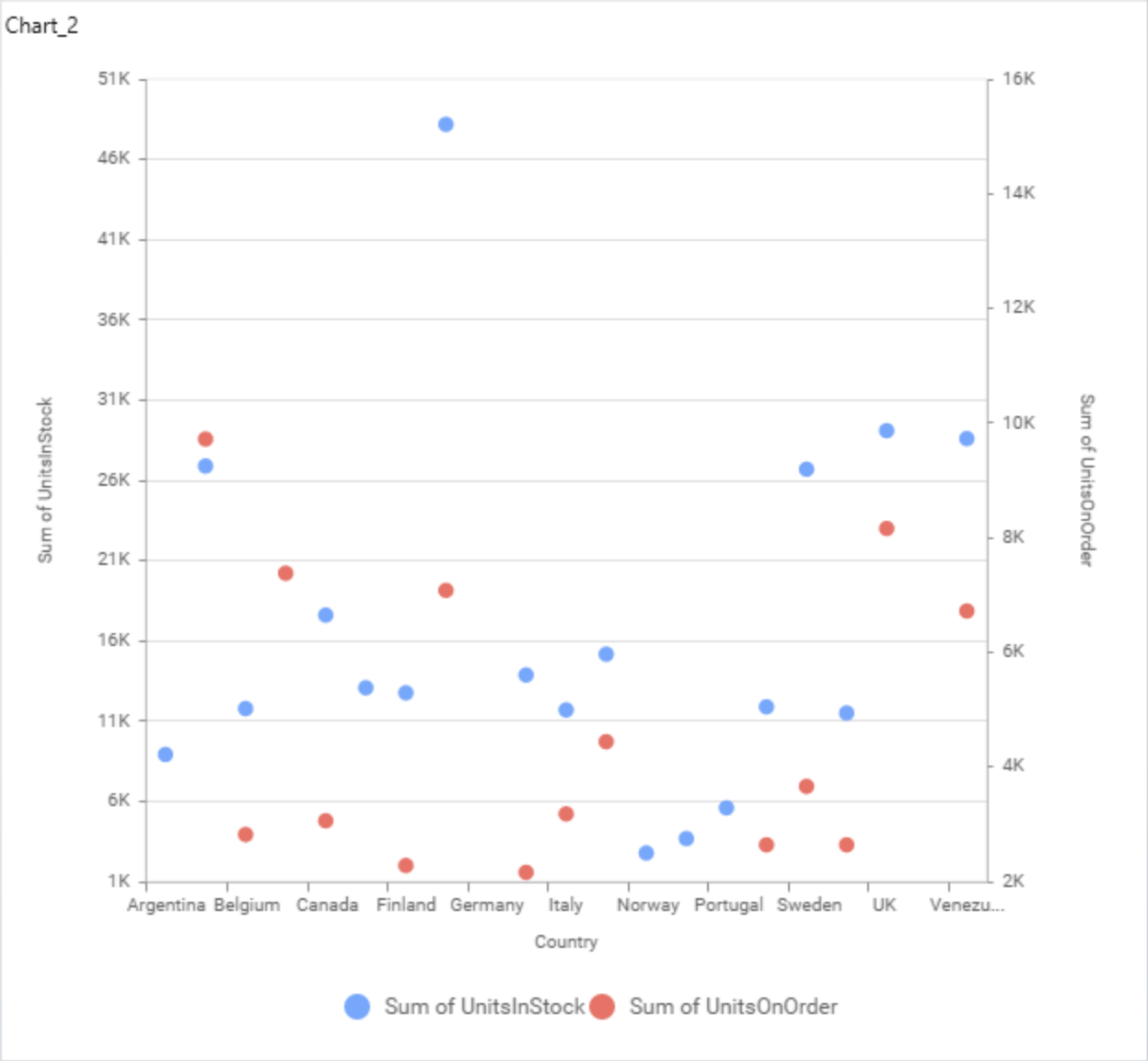
This allows you to increase or decrease the font size of the secondary axis label. Default font size for the secondary axis label was 10 pixels.

Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

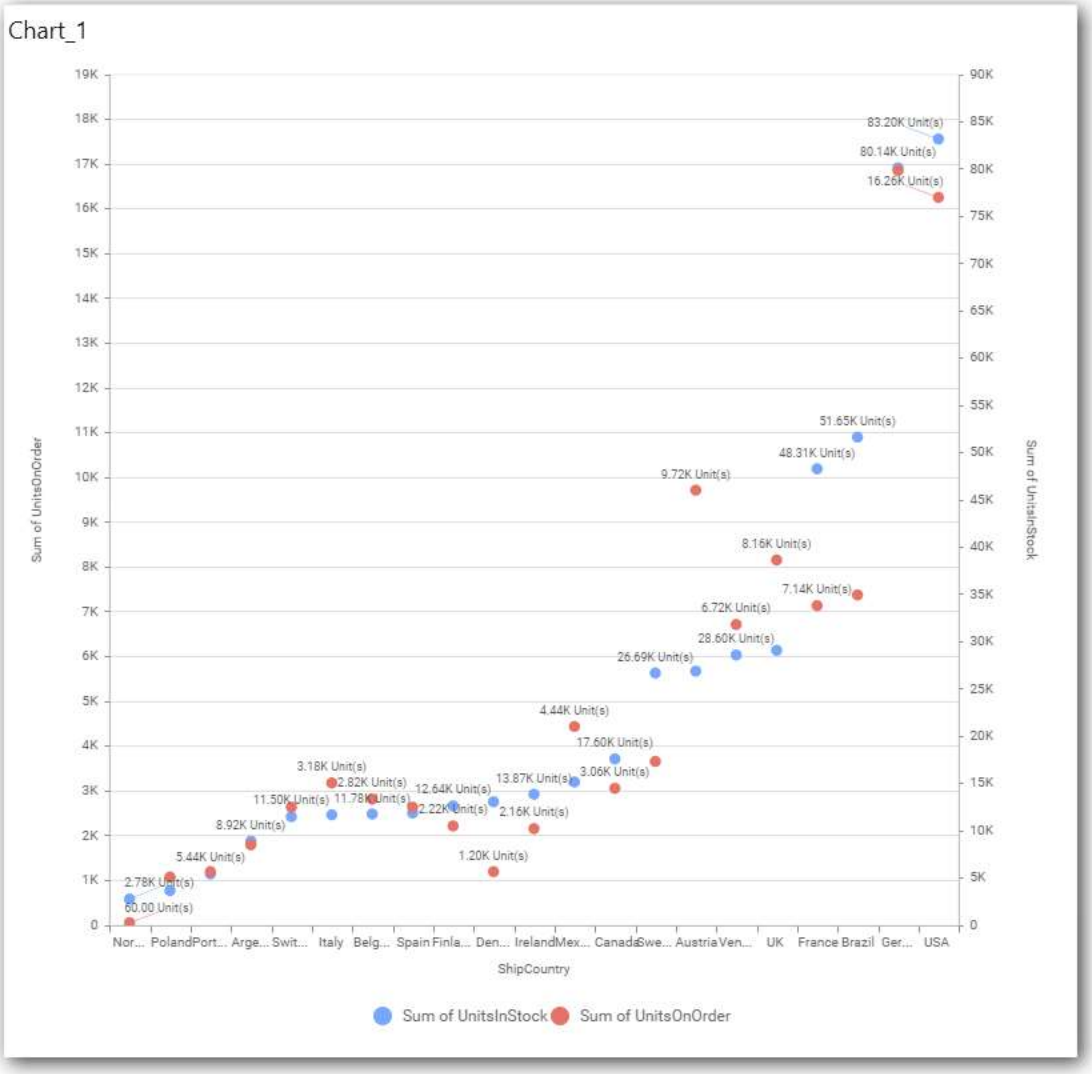


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



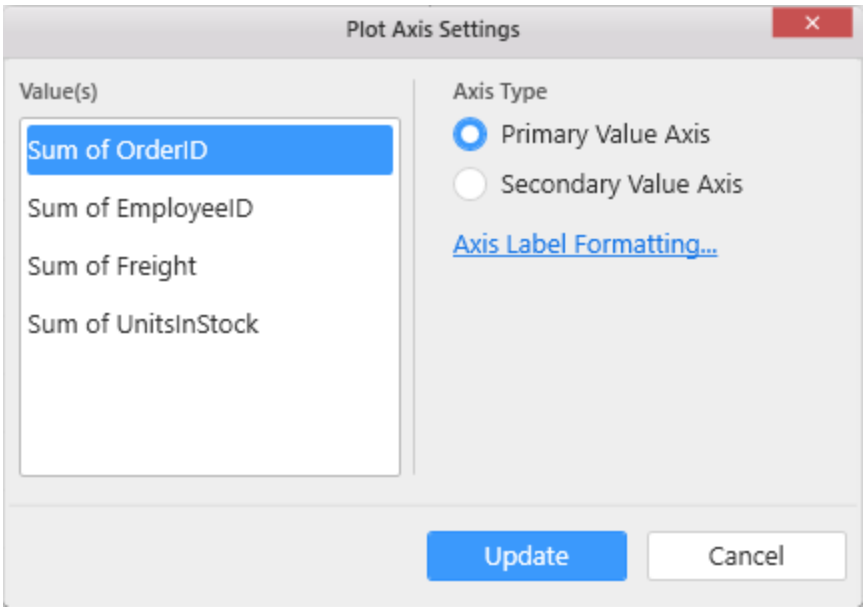
Sort Order

To define the sorting of chart based on any of the measures that you dropped and its order through this option. Following screenshot illustrates the Ascending sort order.

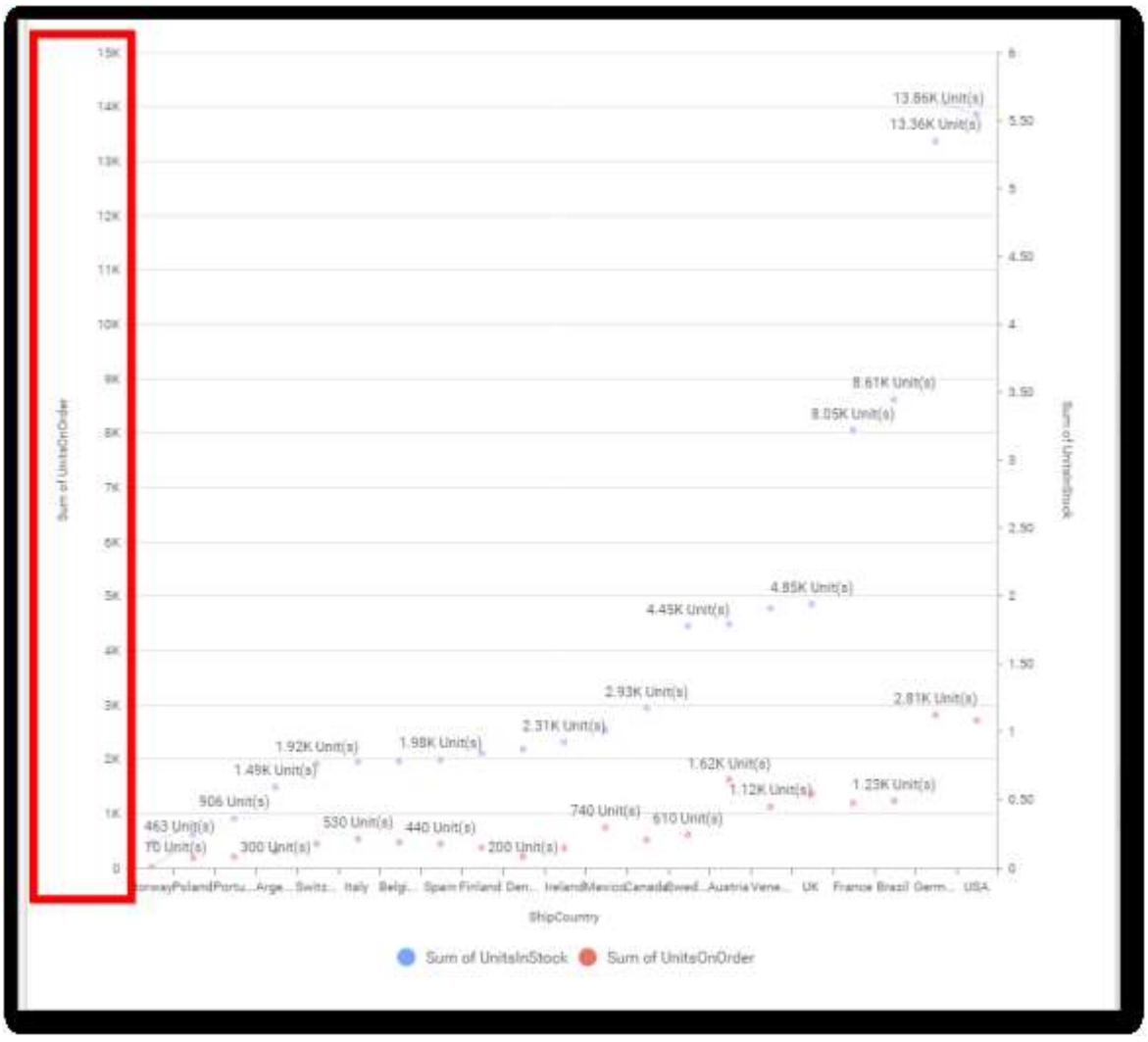


Plot Axis Settings

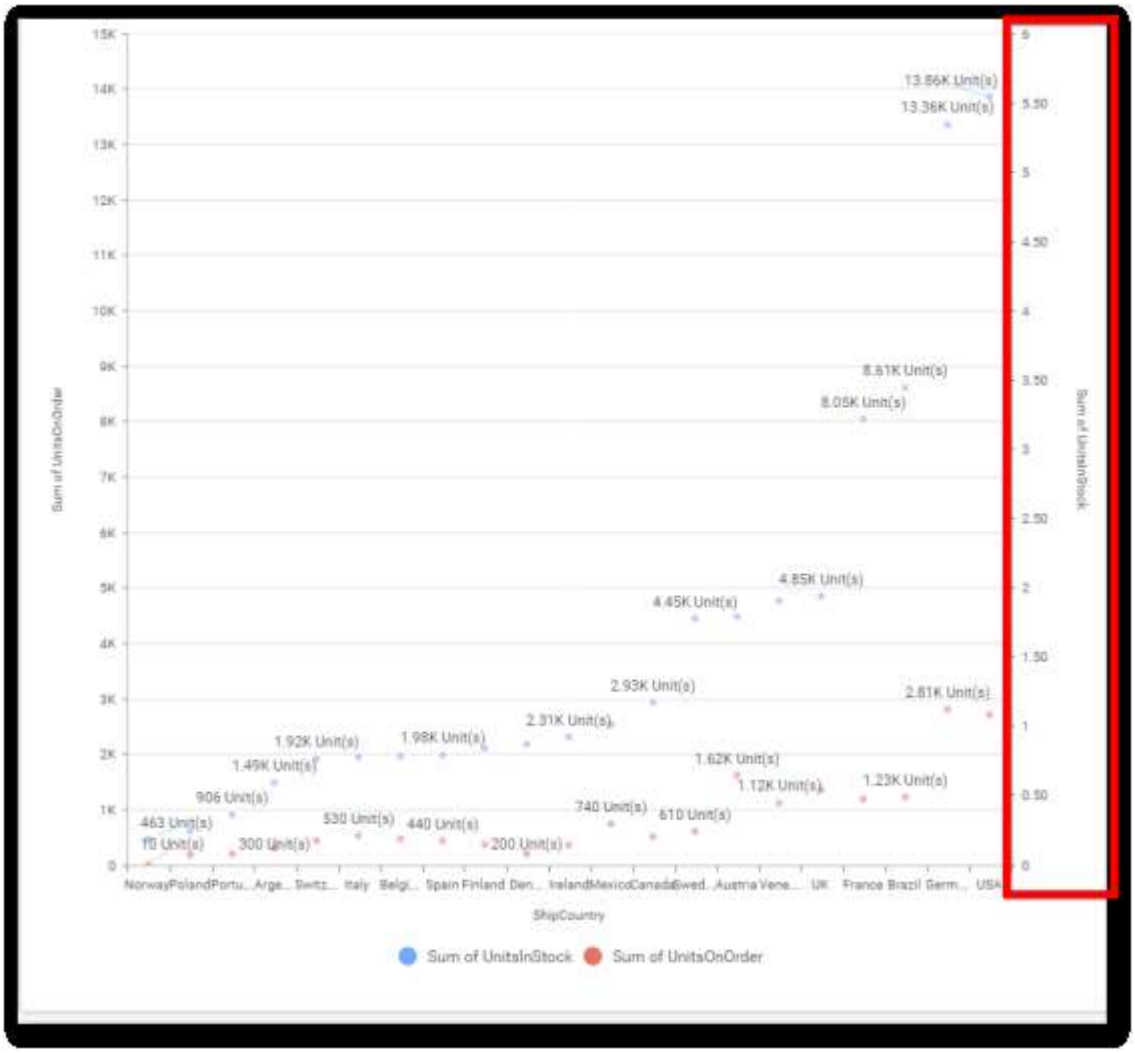
This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



Primary Value Axis

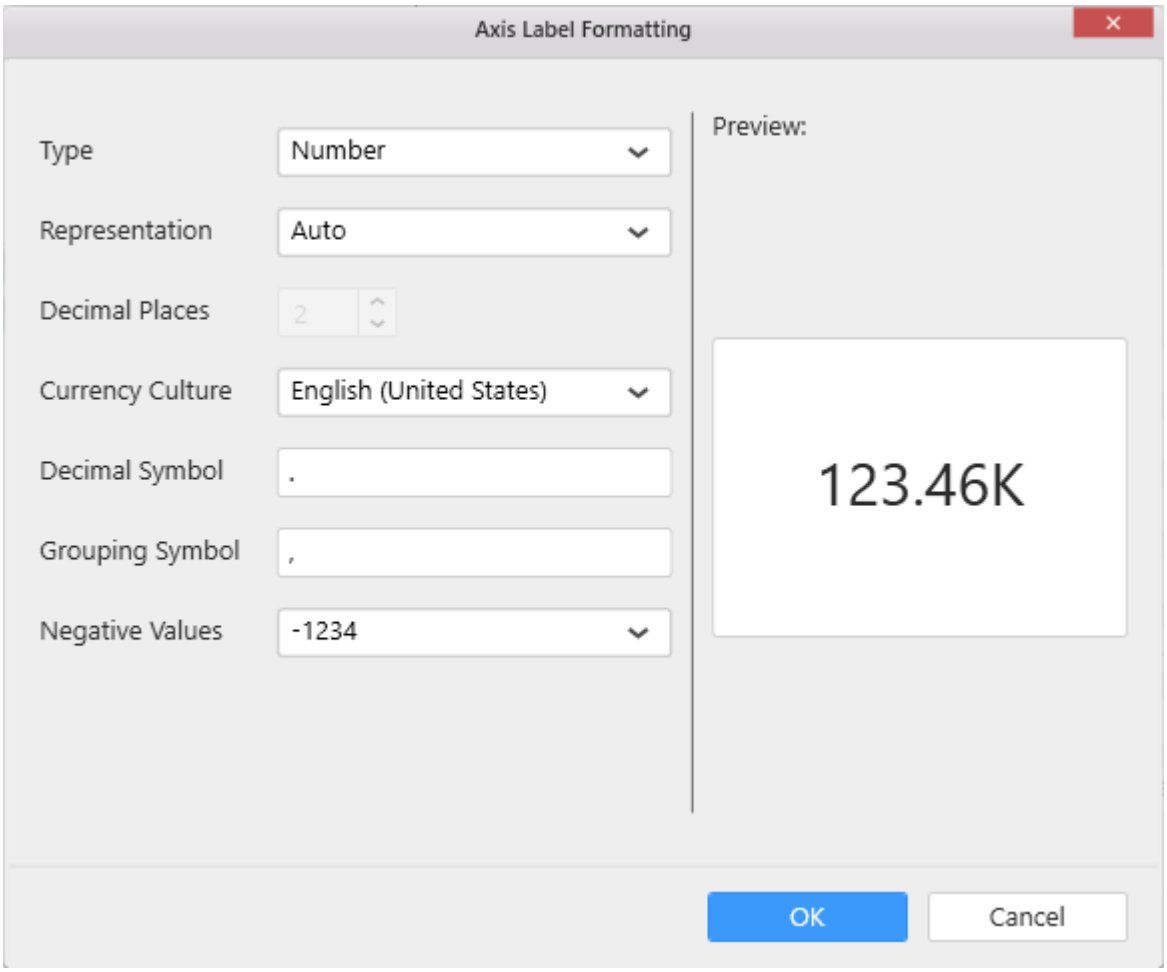


Secondary Value Axis



Axis Label Formatting

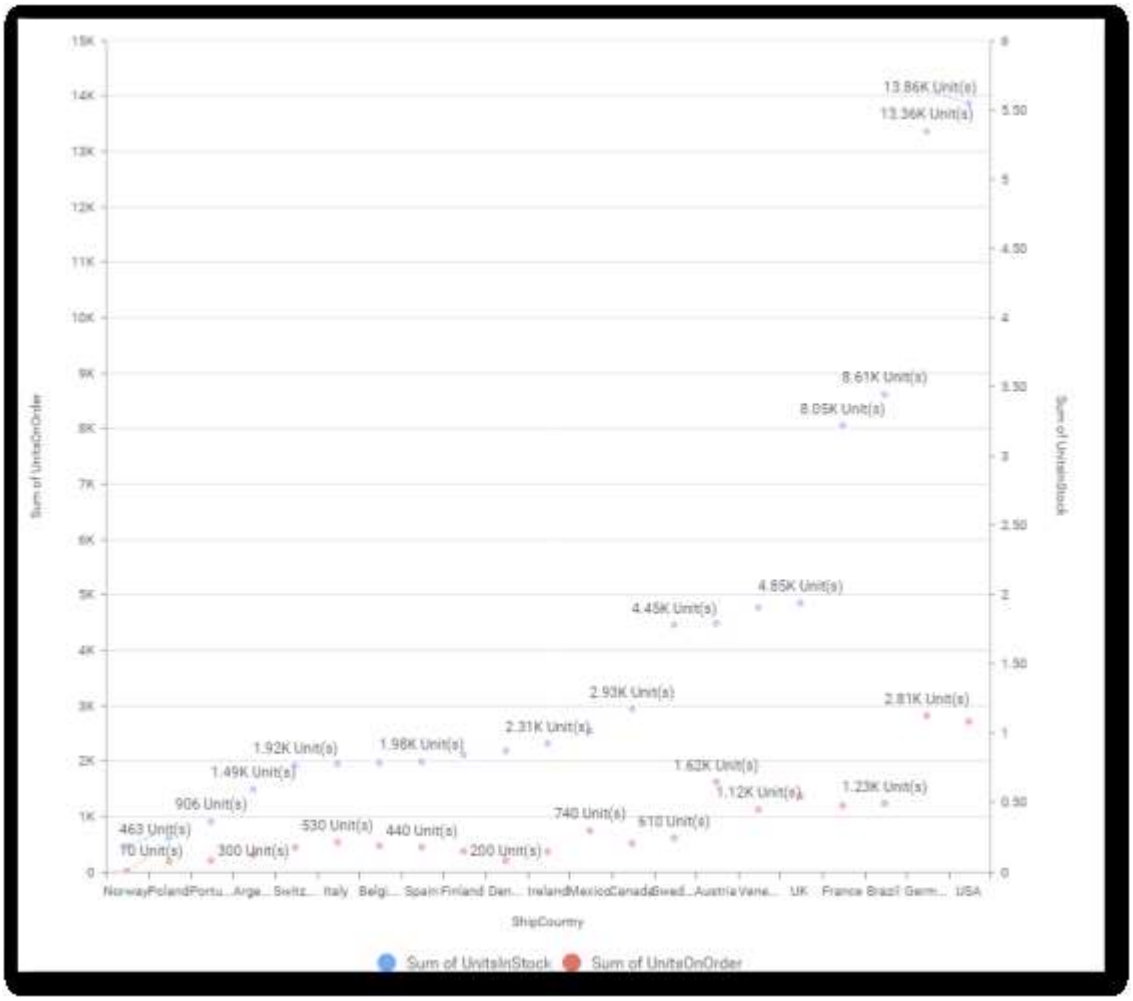
This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.



Grid Line Settings

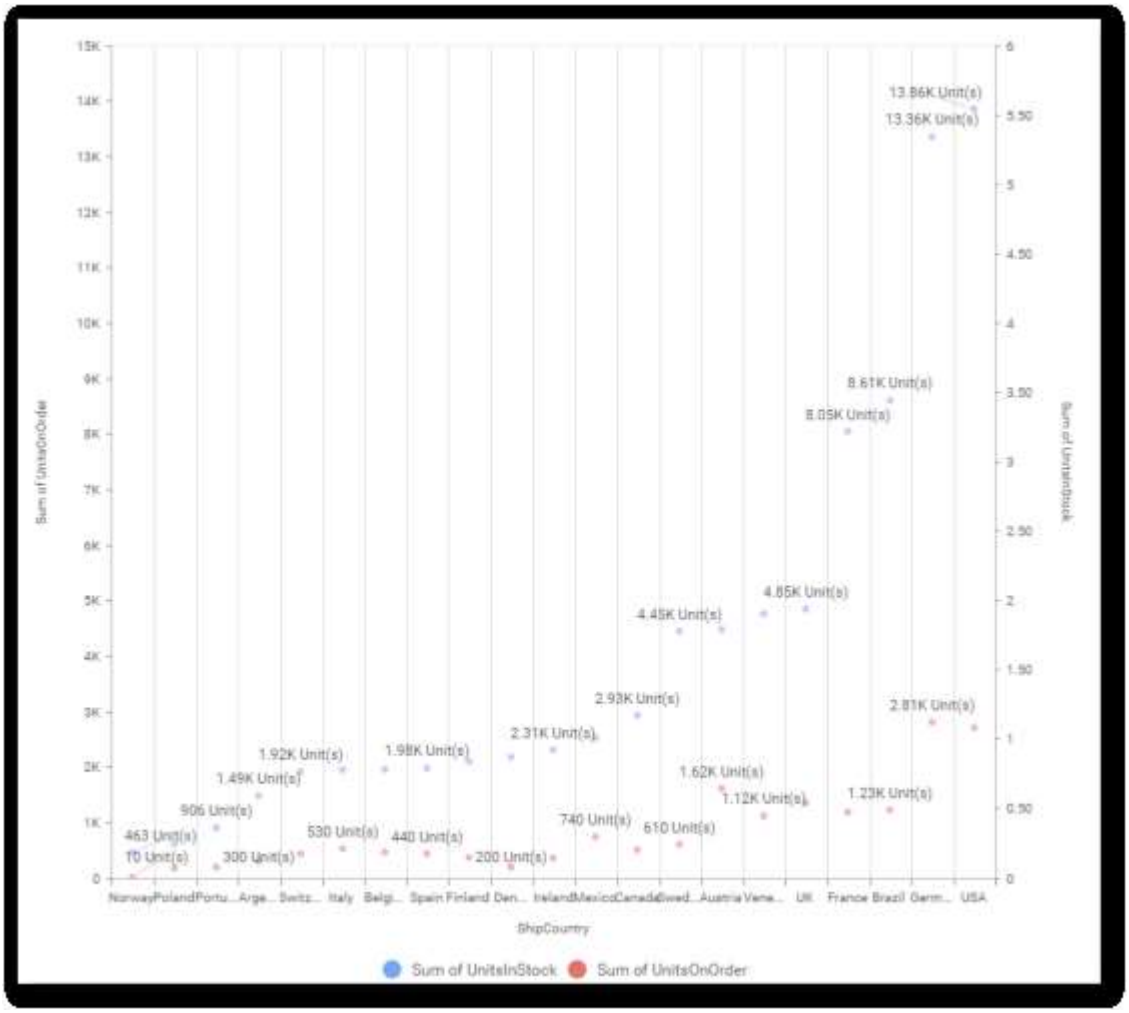
Primary Value Axis

This allows you to enable the Primary Value Axis gridlines.



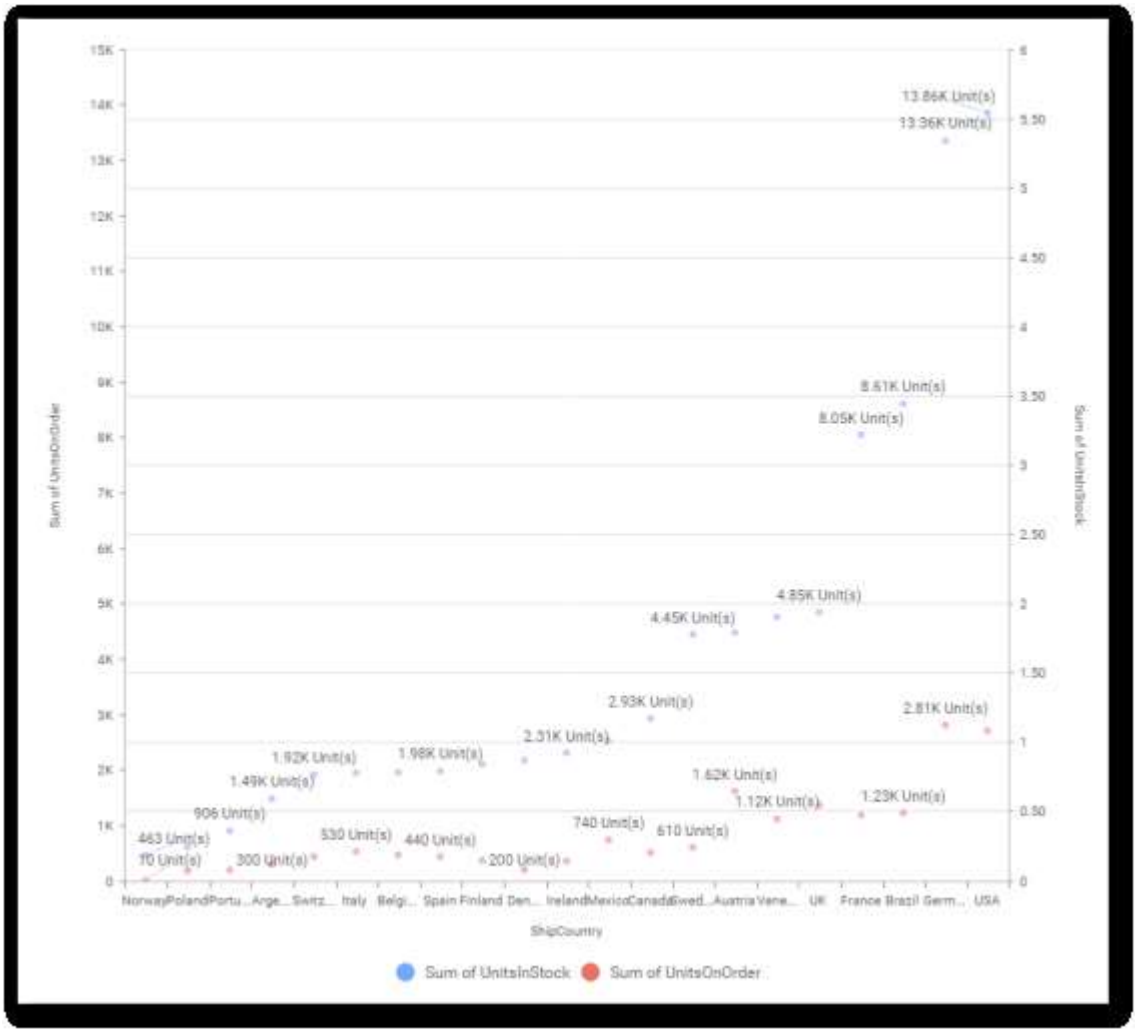
Category Axis

This allows you to toggle the visibility of Category Axis gridlines.



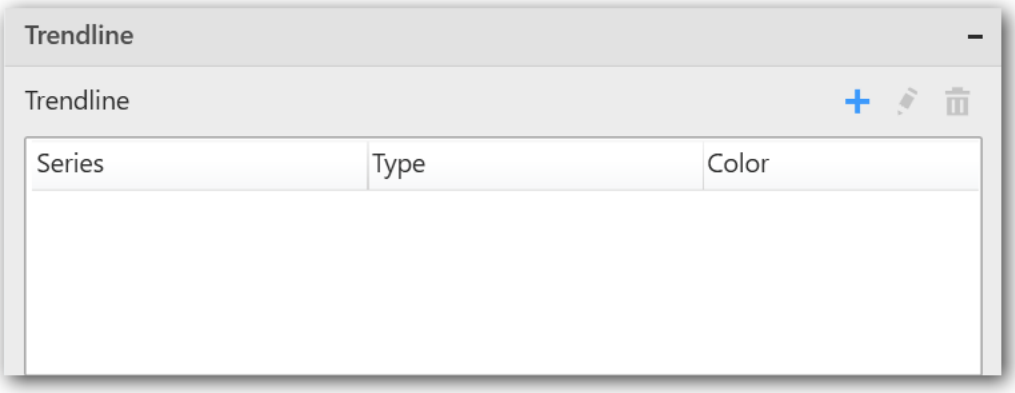
Secondary Value Axis

This allows you to toggle the visibility of Secondary Value Axis gridlines.

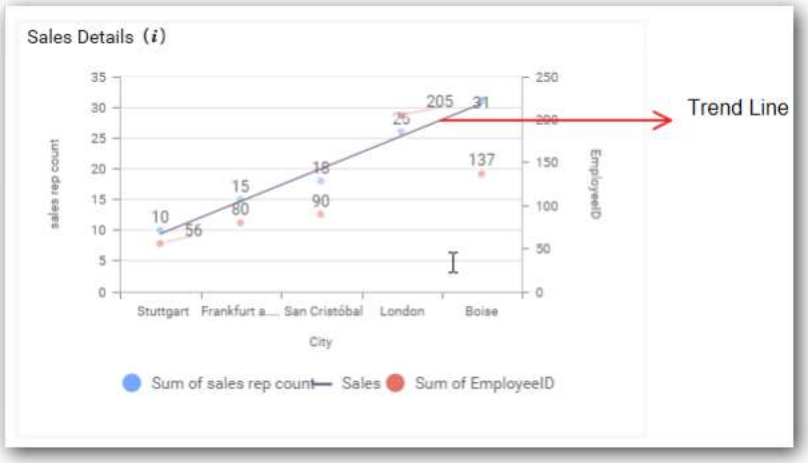


Trend Line Settings

You can add trend line to chart based on dropped measure that you select. You can also customize its legend text, line type and line color. Trend line is not visible, by default.



After applying these settings, it will reflect in chart like below.



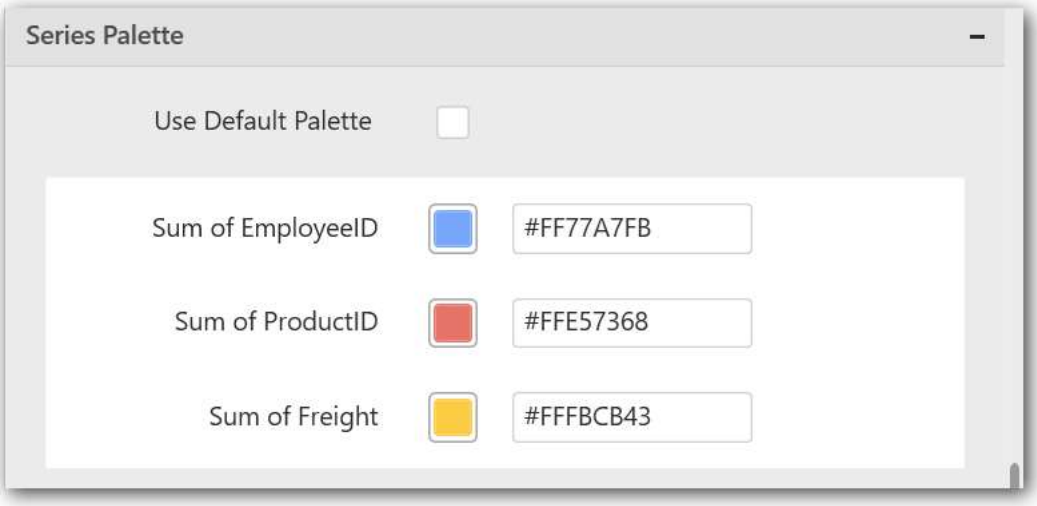
You have options to add or delete the added trend lines.

Series Palette

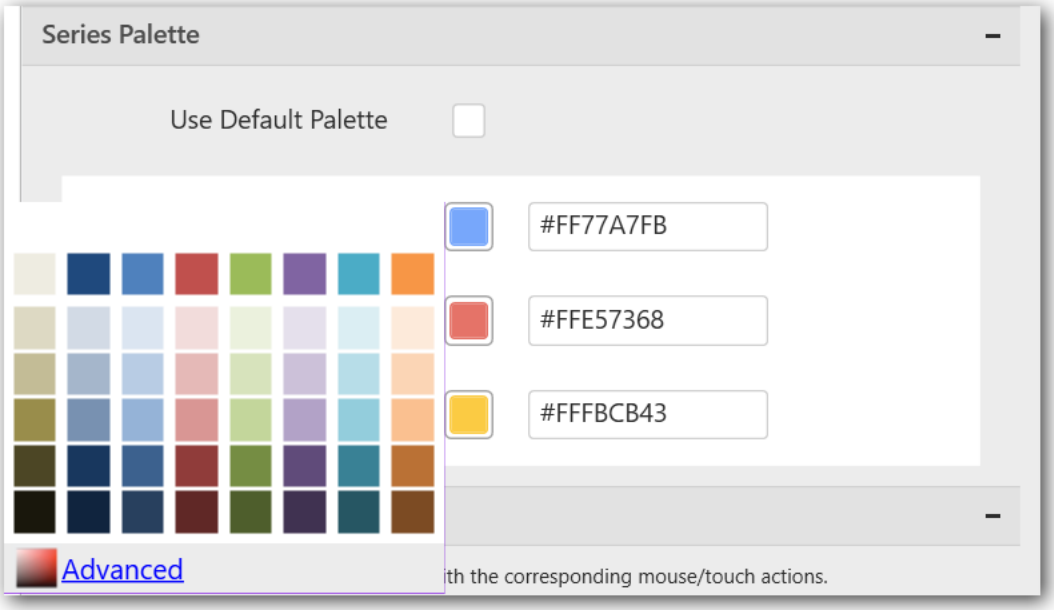
This allows you to customize the chart series color through Series Palette section.

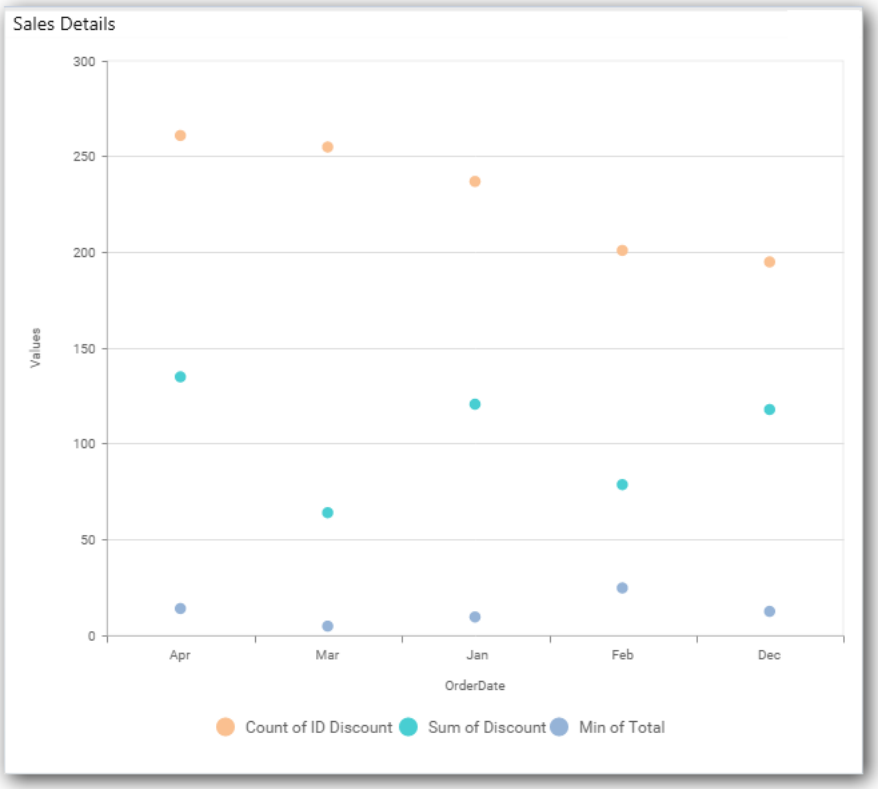
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



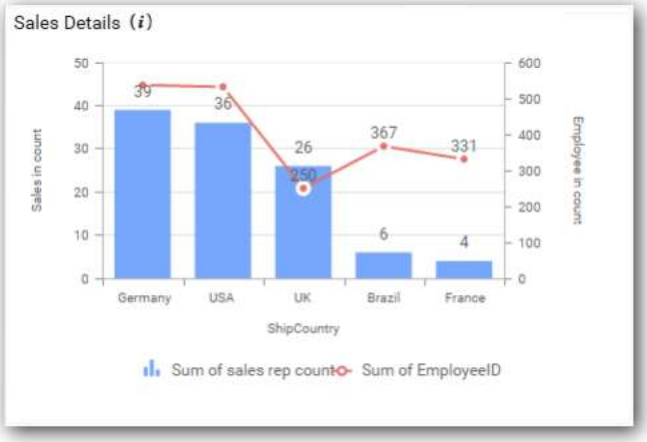
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Combo Chart

Combo Chart allows you to compare values across categories representing different data sets with different chart types and a secondary axis.

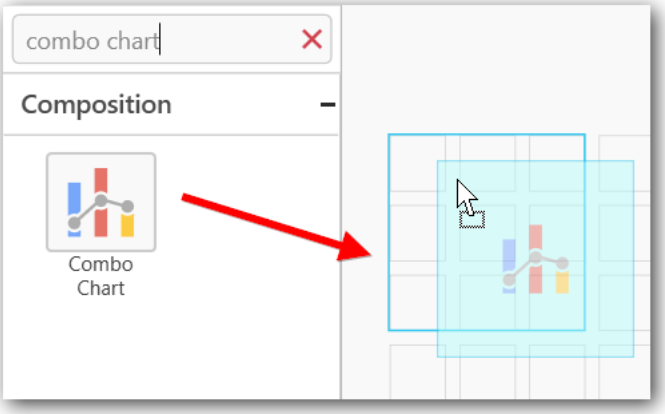


How to configure flat table data to Combo Chart?

To plot a combo chart, a minimum requirement of 1 value or line value and 1 column is needed. Combo charts differ in one case when compared to basic charts, which is, on adding series to Row section, for each distinct value of the series added, one combo chart will get rendered. W.r.t this chart, one additional block Line Value will be available in Data pane where we can drop the measure whose type will be line, which need to be compared against.

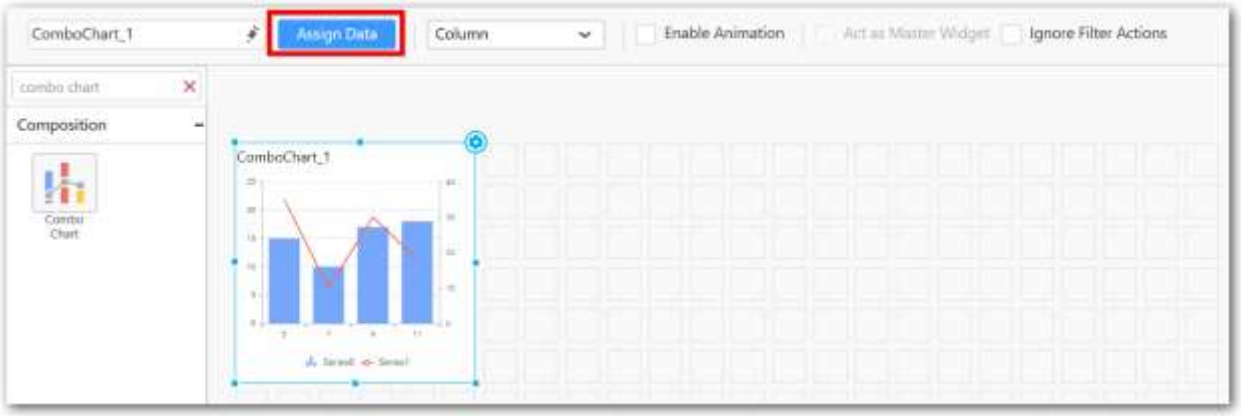
Follow the steps to configure data to combo chart

Drag and drop the combo chart widget to canvas and resize it to your required size.

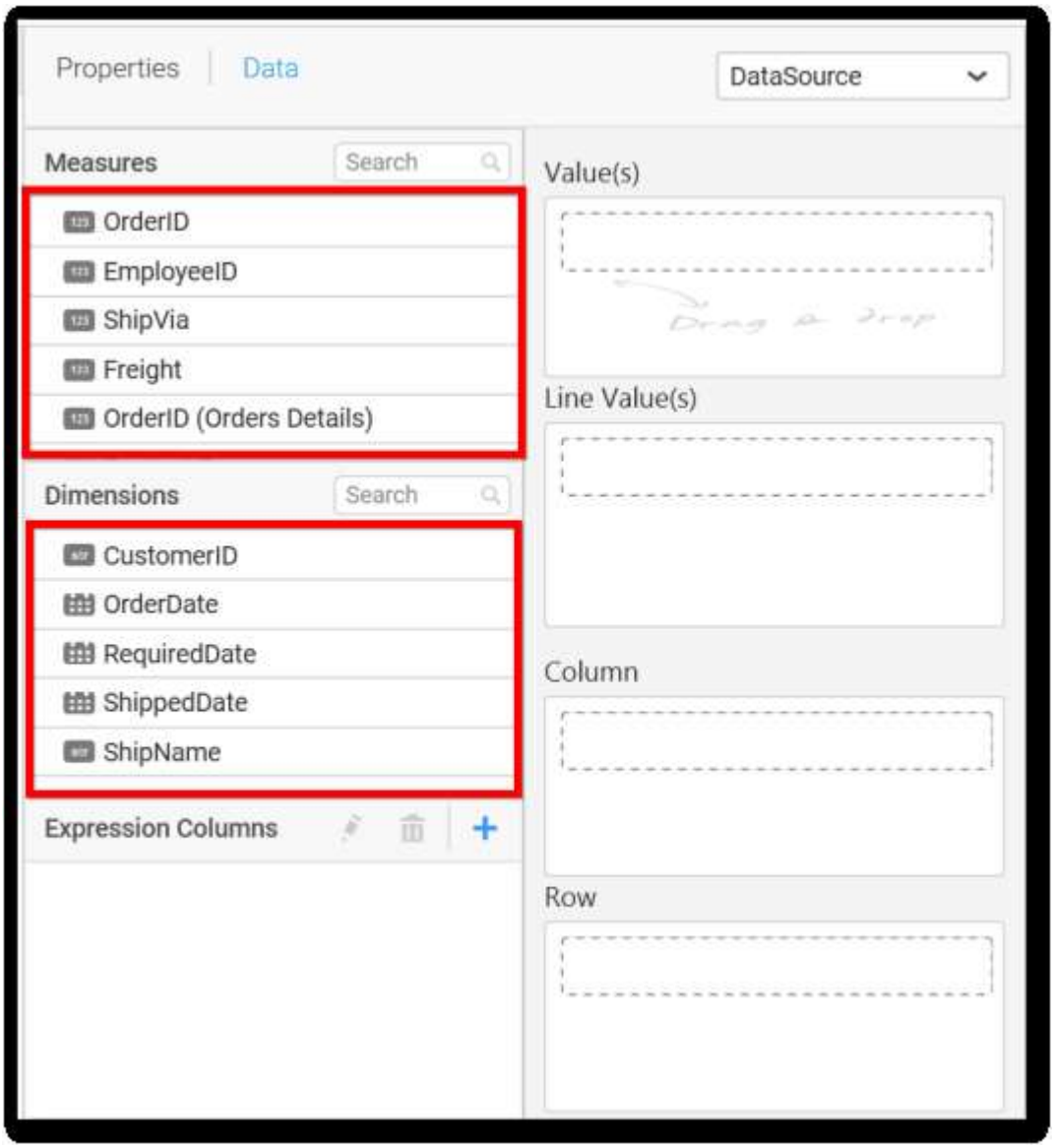


Connect to the data source.

Focus on the Combo chart and click on Assign Data.

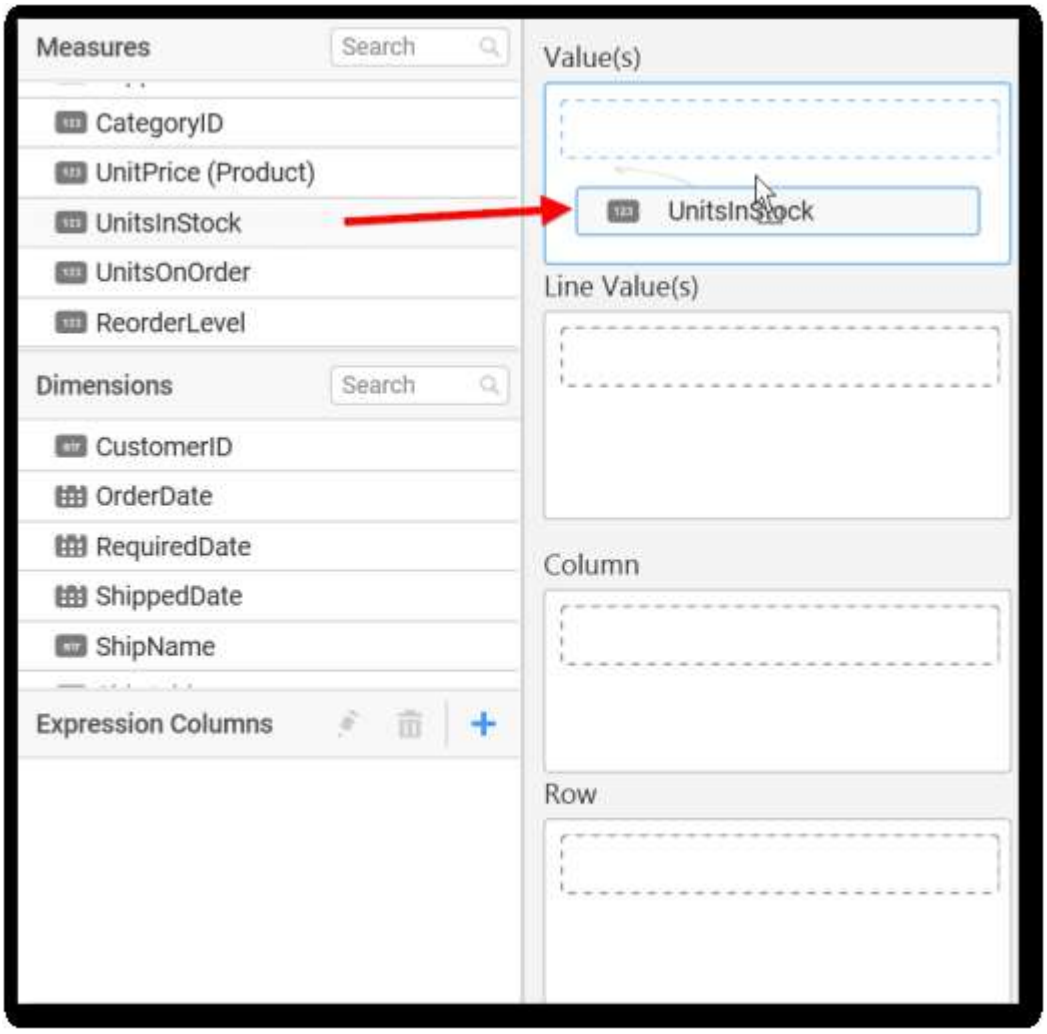


The data pane will be opened with available measures and dimensions from the connected data source.

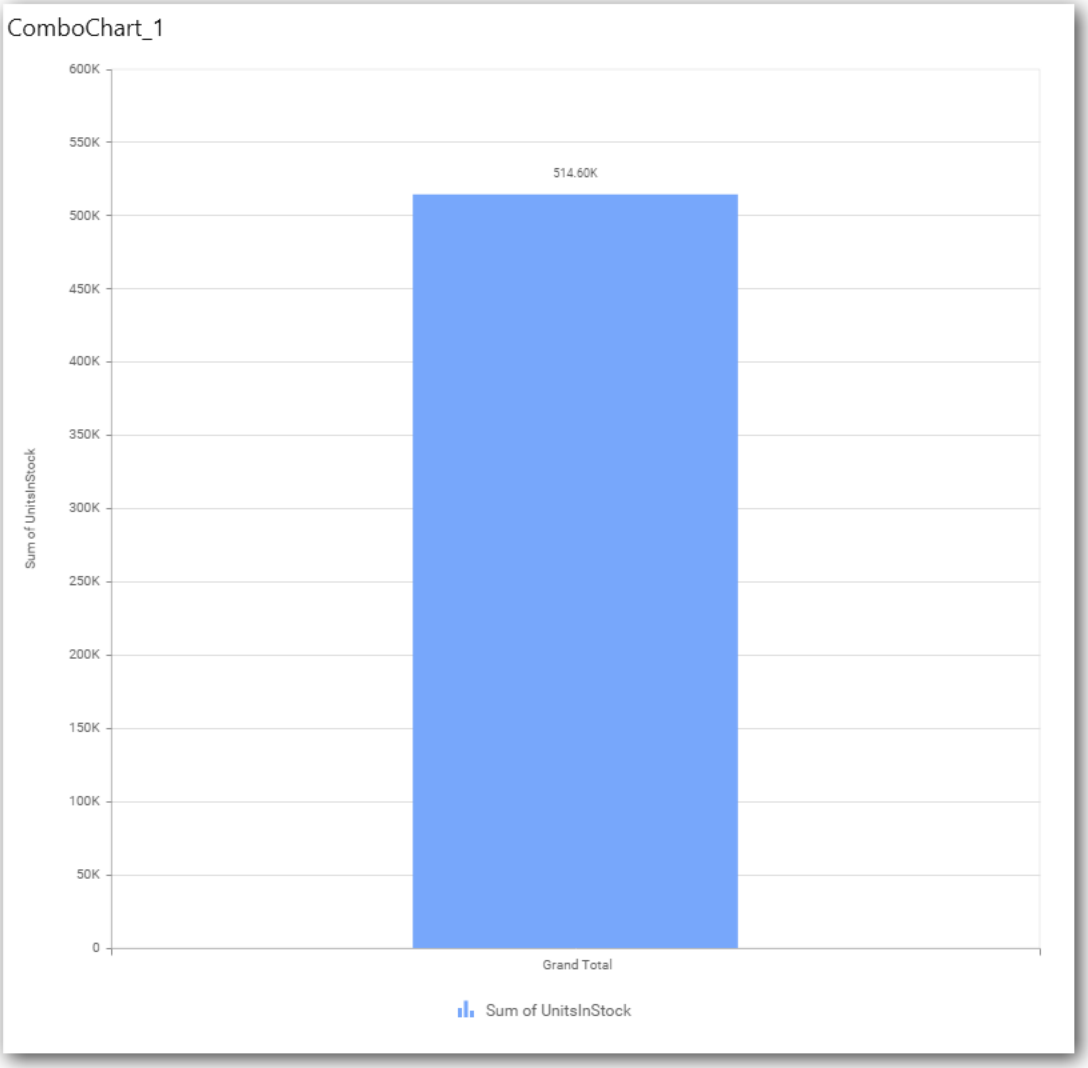


Assigning Value(s)

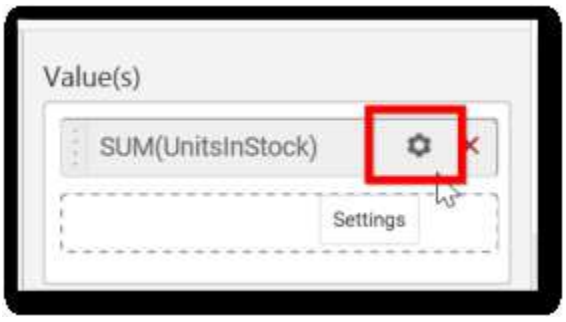
Drag and drop the Measure into Value.



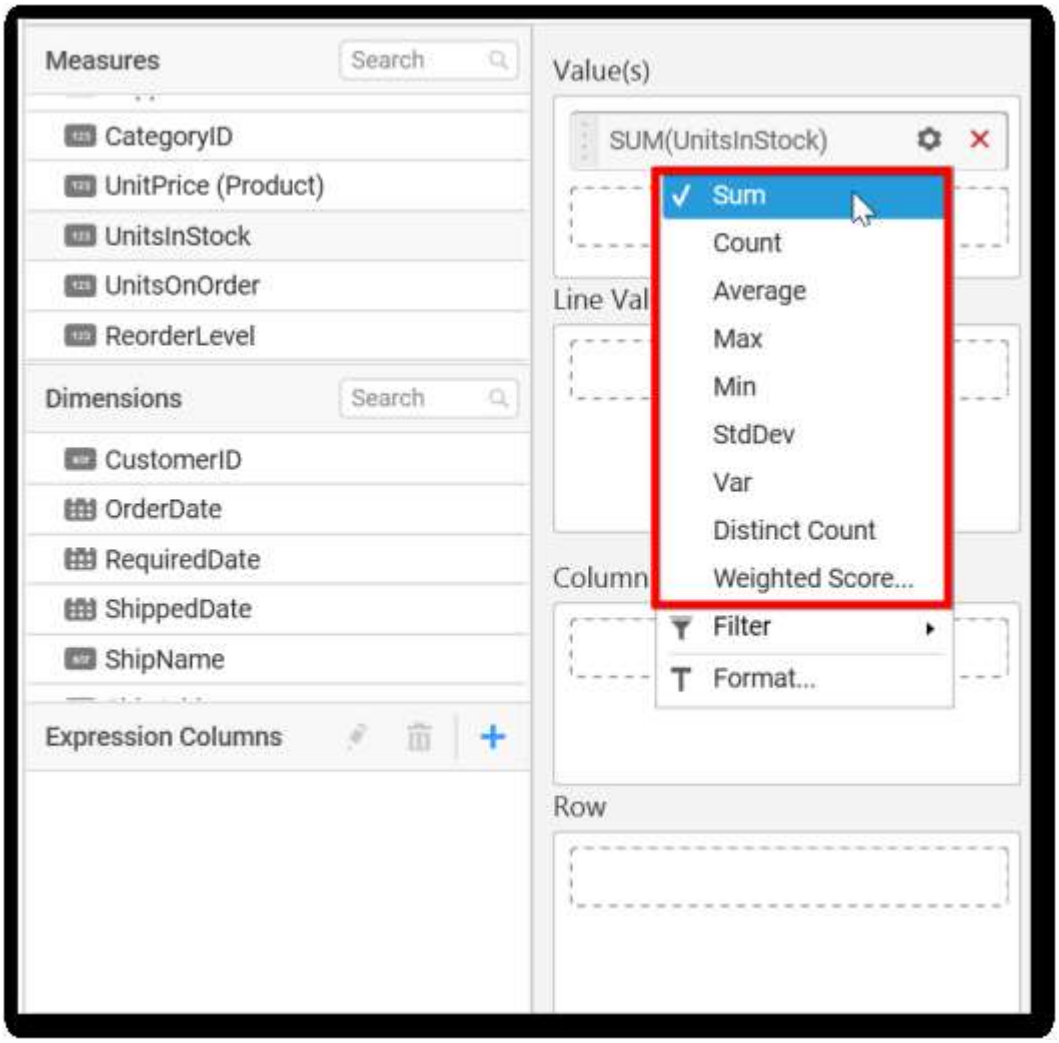
Now the chart will be rendered like this.



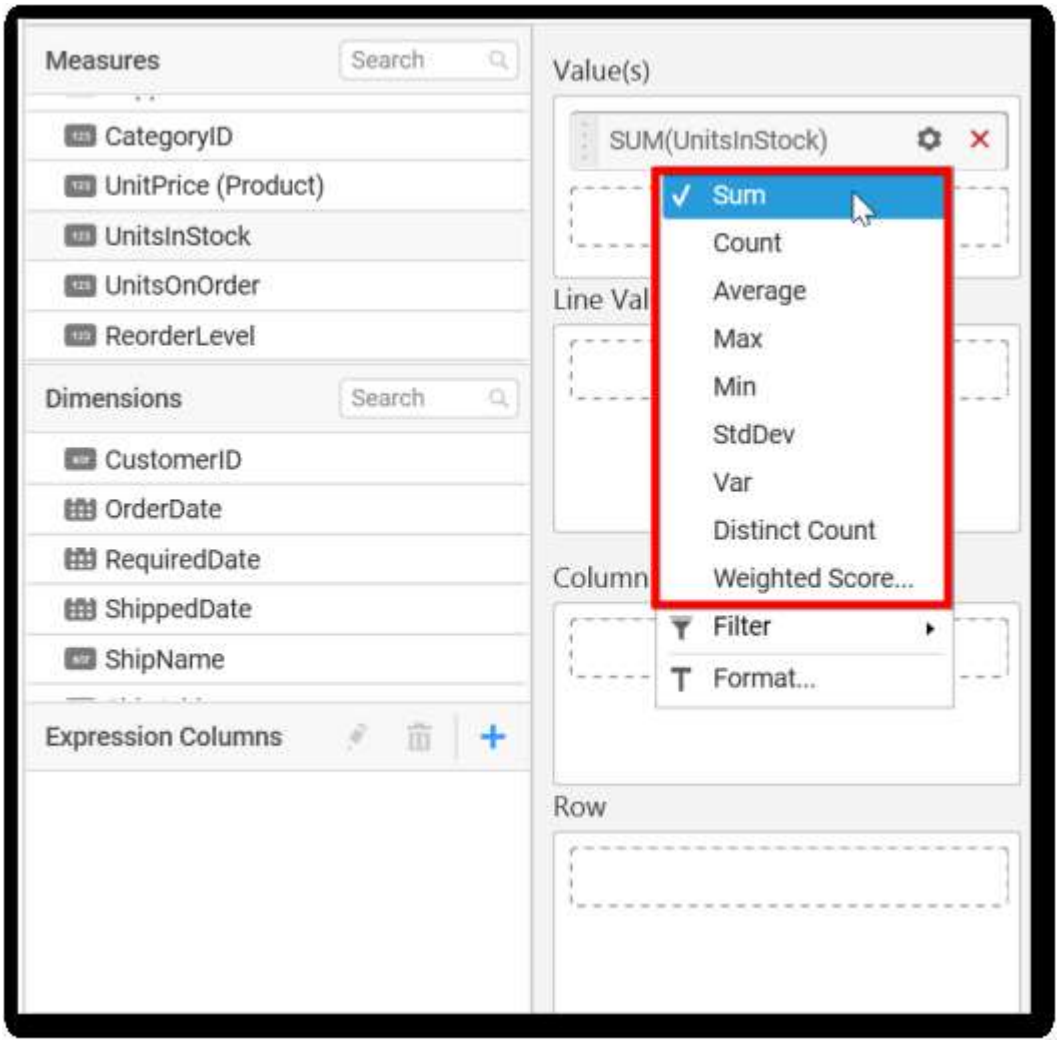
You can change the summary type of the value by clicking on Settings option.



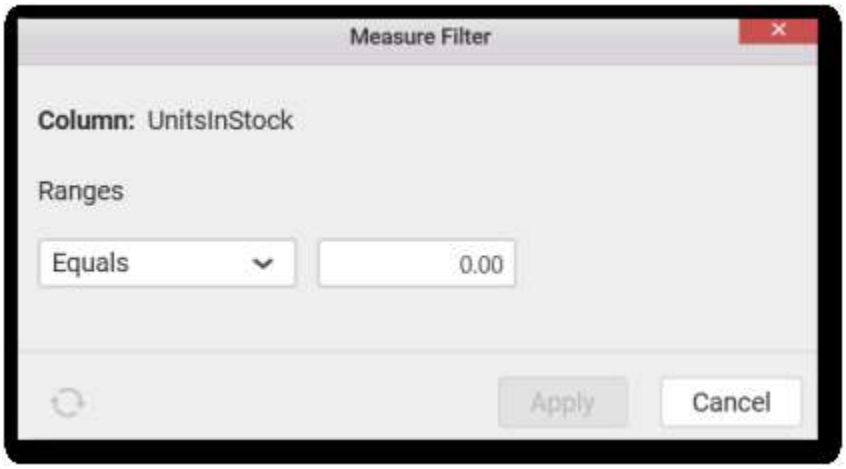
Select the required summary type from list.

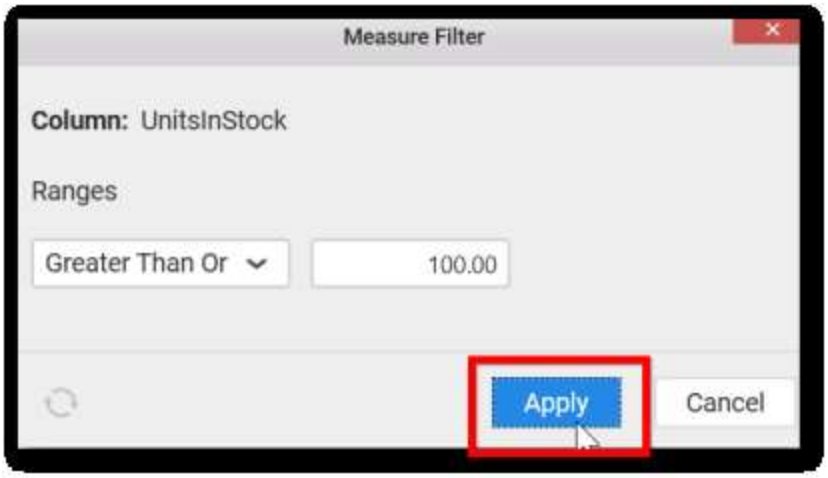


You can select what data to be displayed by choosing filter option.

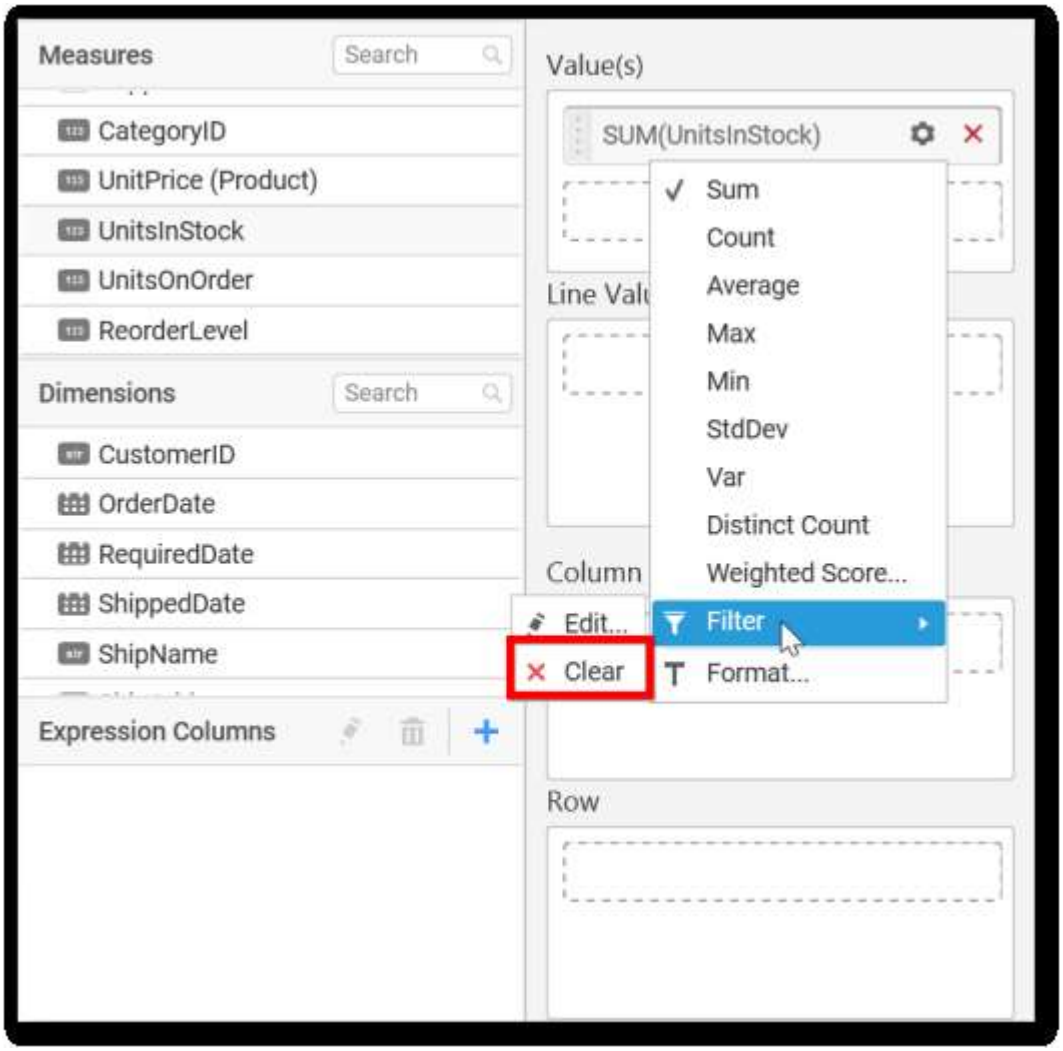


The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.

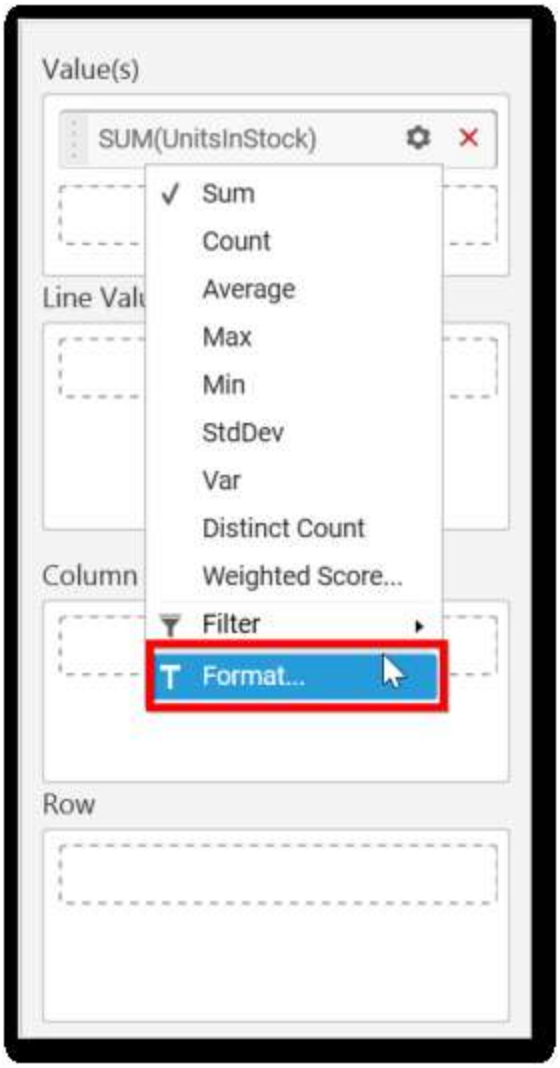




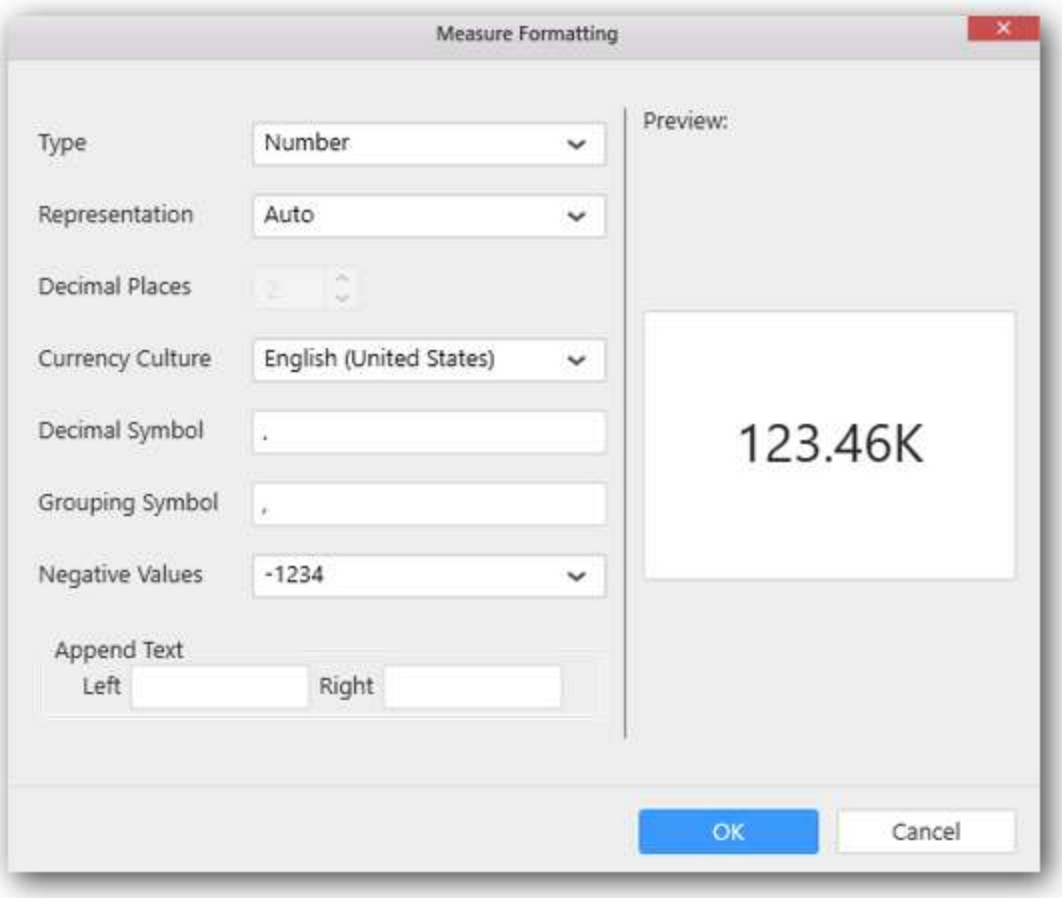
You can Clear the filter.



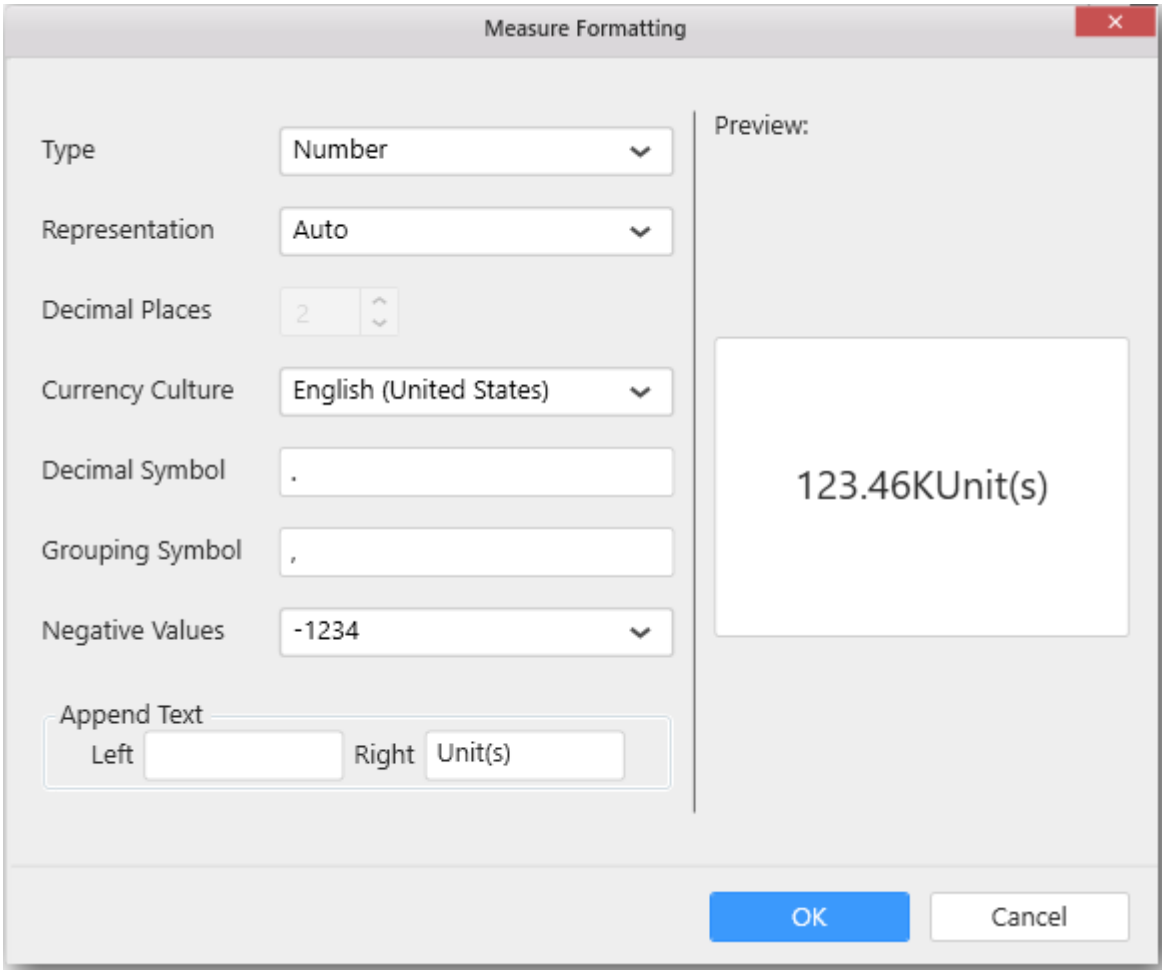
You can Format the value.



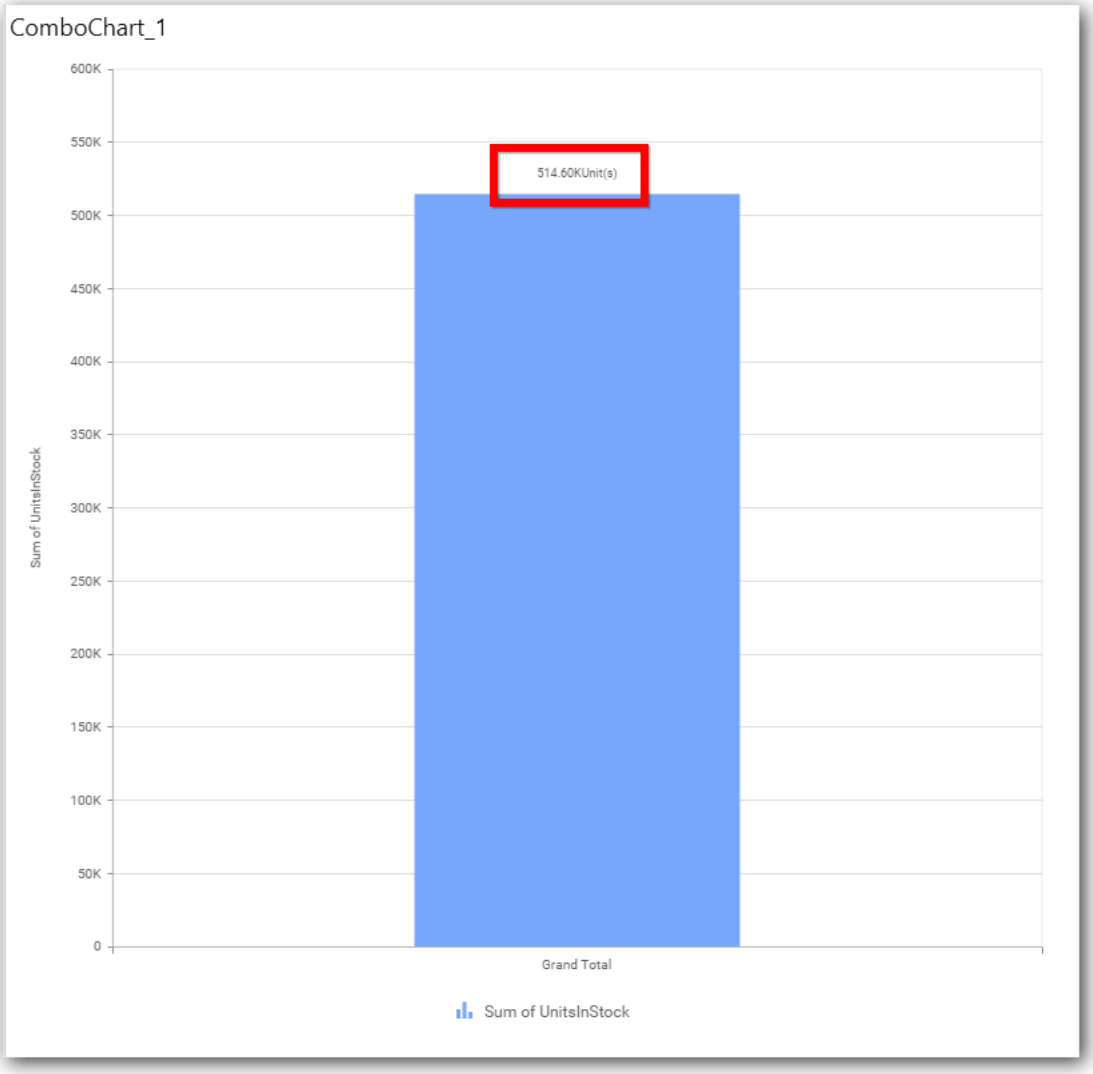
The format options will be shown.



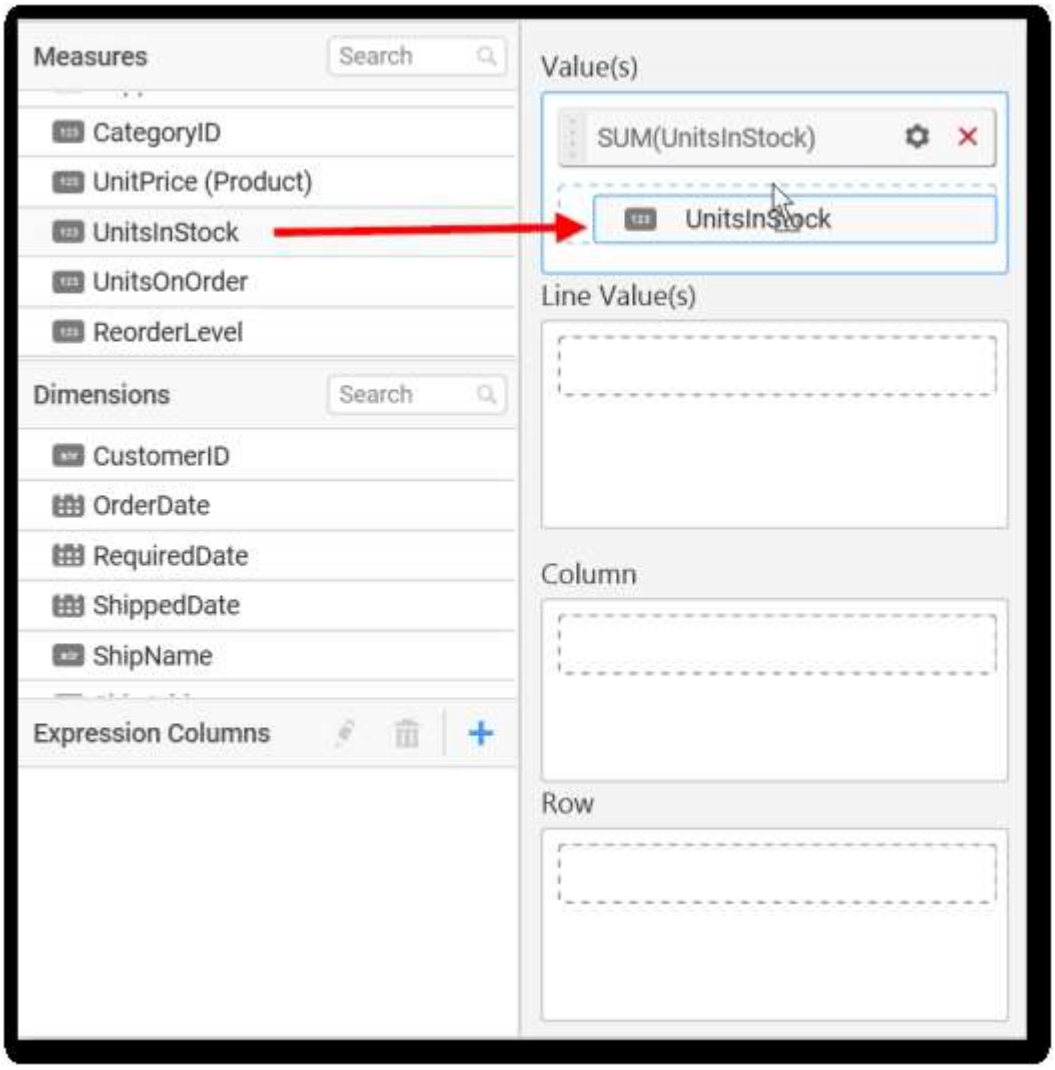
Choose the options you need and click OK.

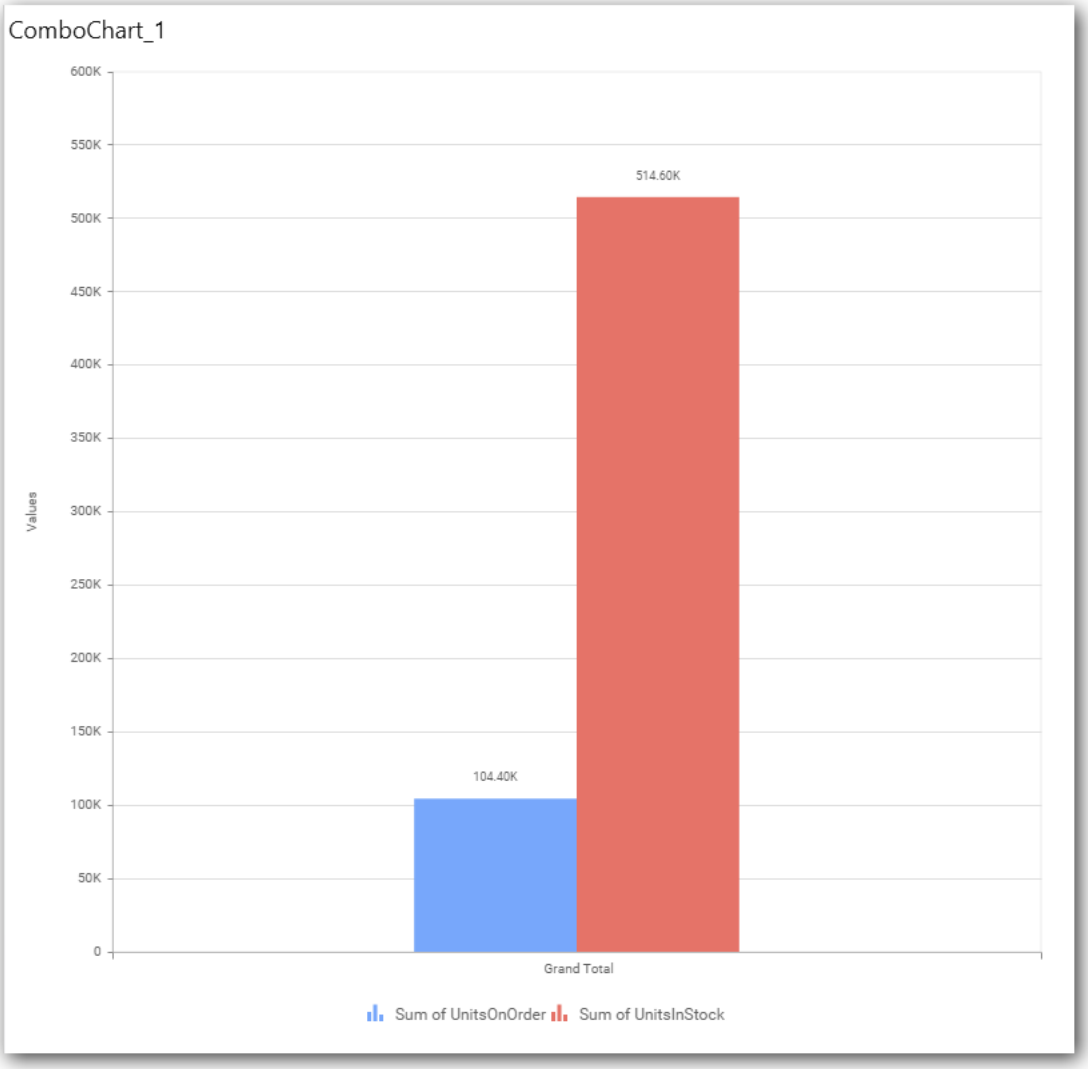


Now the Chart will be rendered like this.



You can add more number values by drag drop the Measures into Value field.

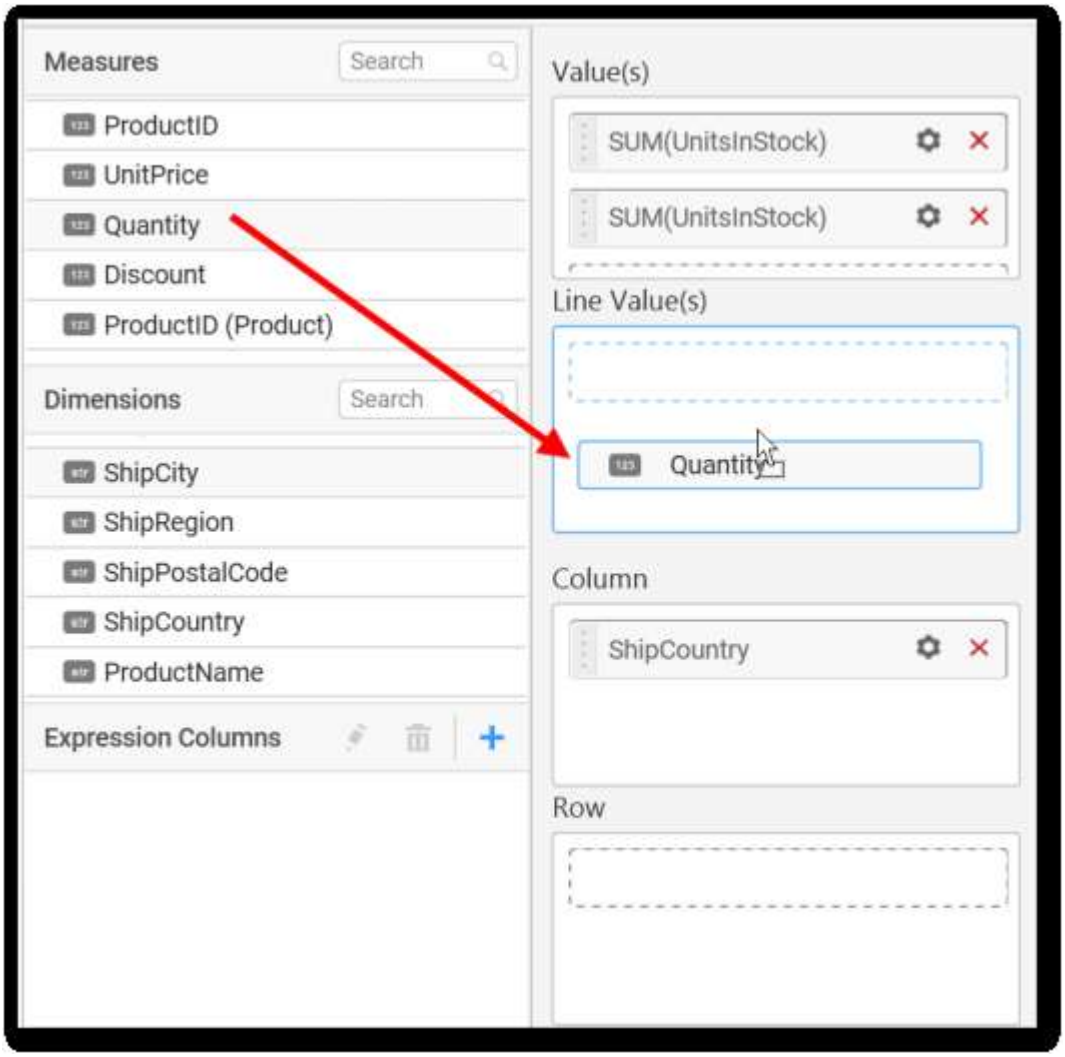


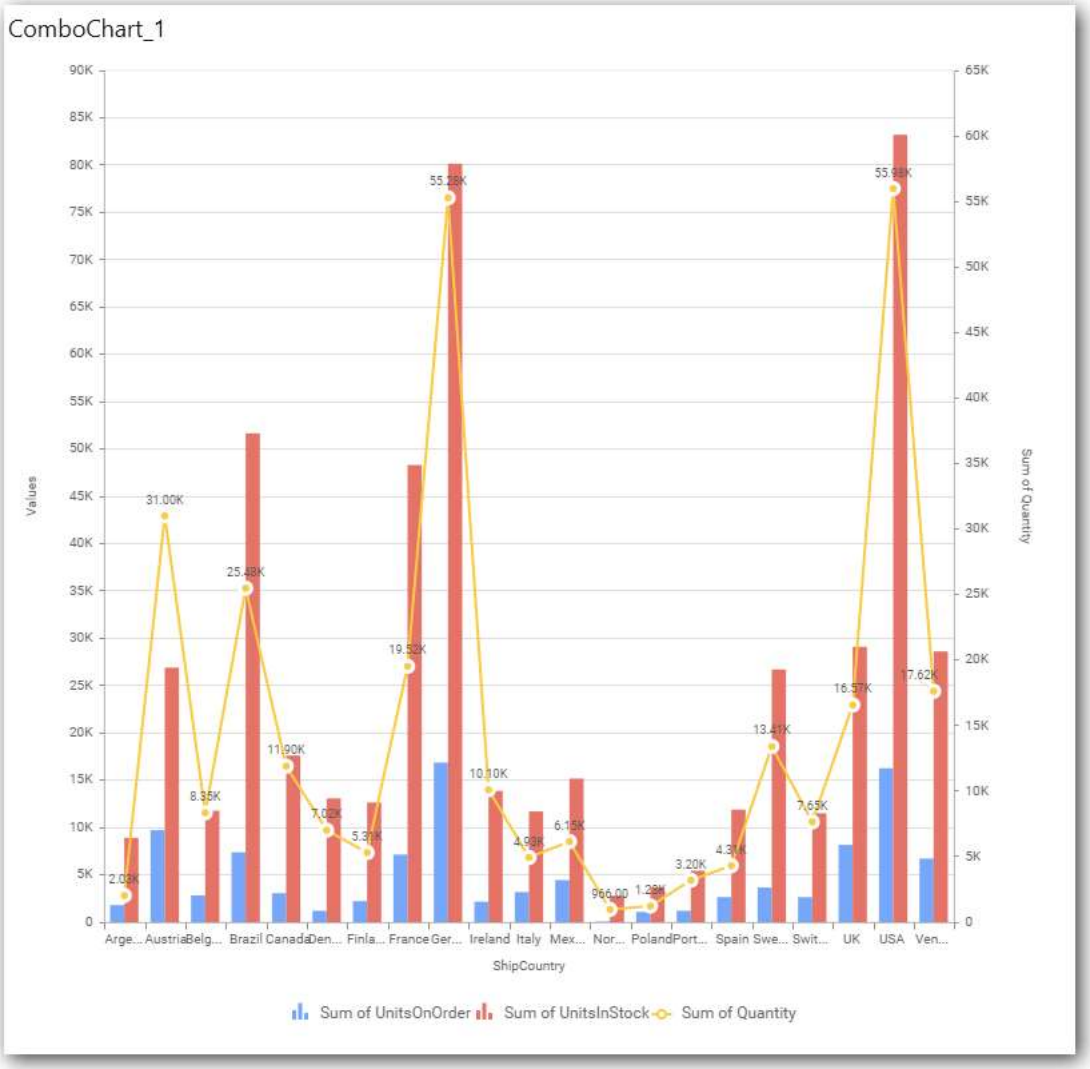


You can also add Dimensions and Columns to Value(s).

Assigning Line Value

You can add the Measures into Line values.

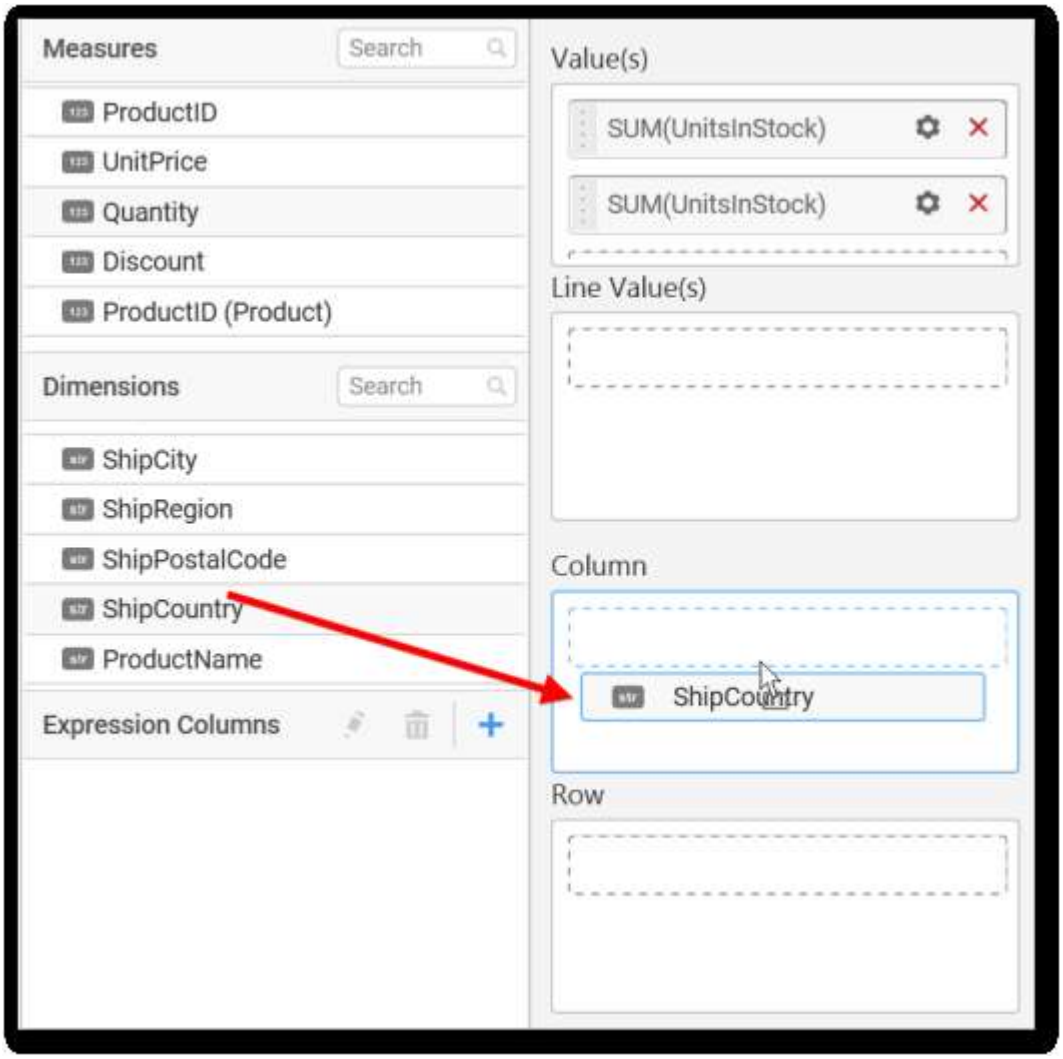


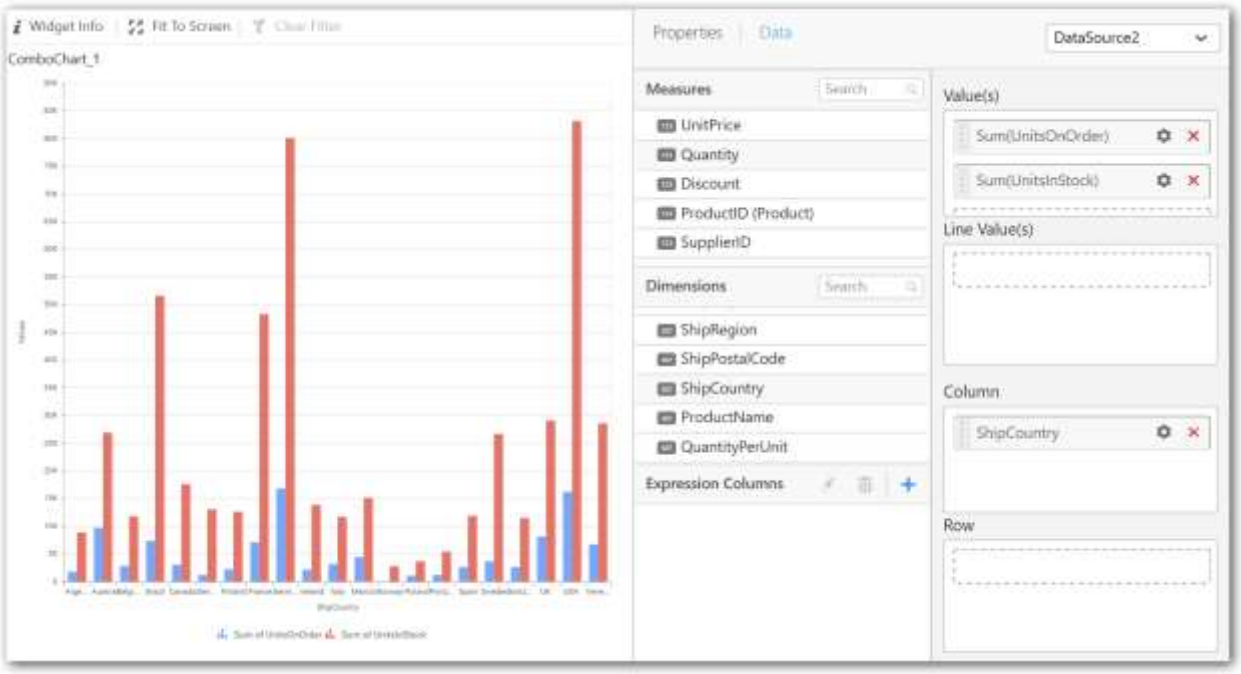


You can add more than one line values of Measure or Dimension, the settings are similar to Value(s) field.

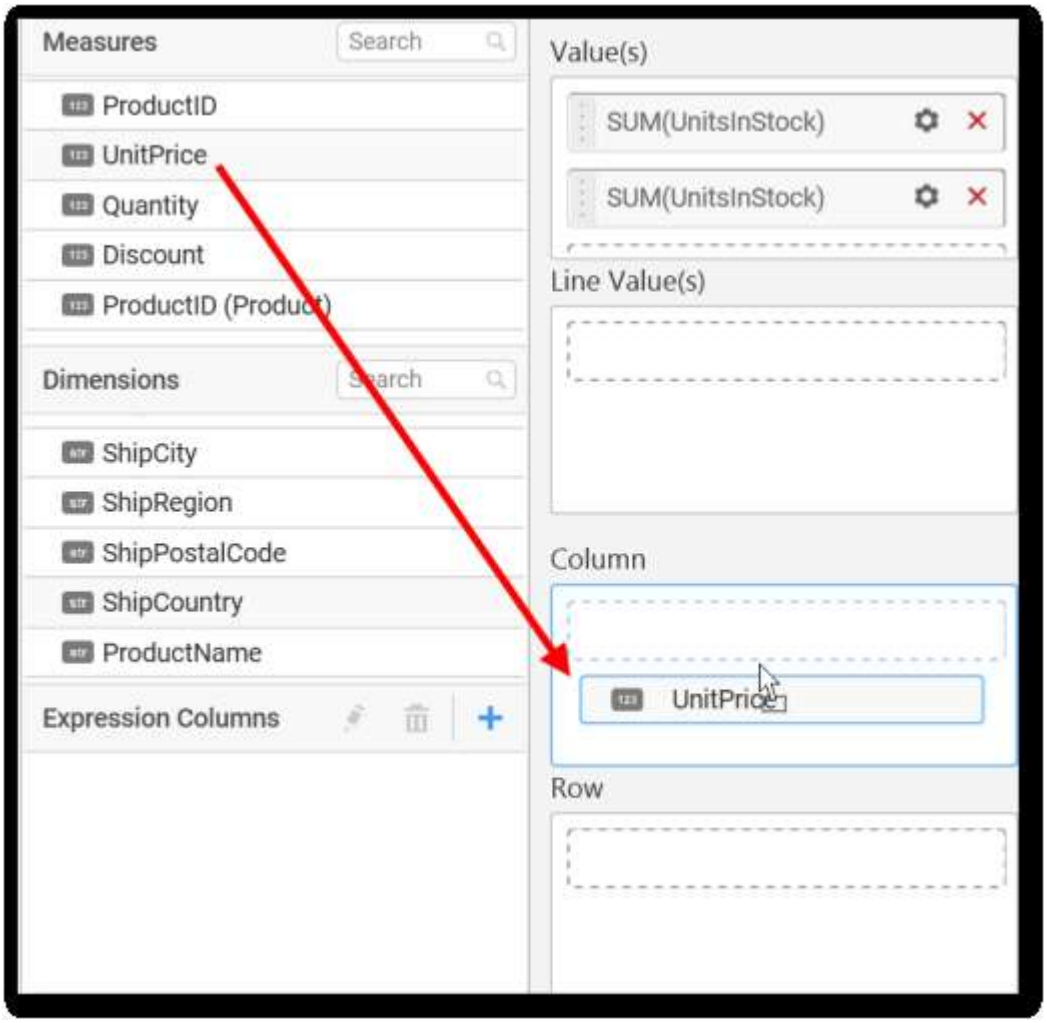
Assigning Column(s)

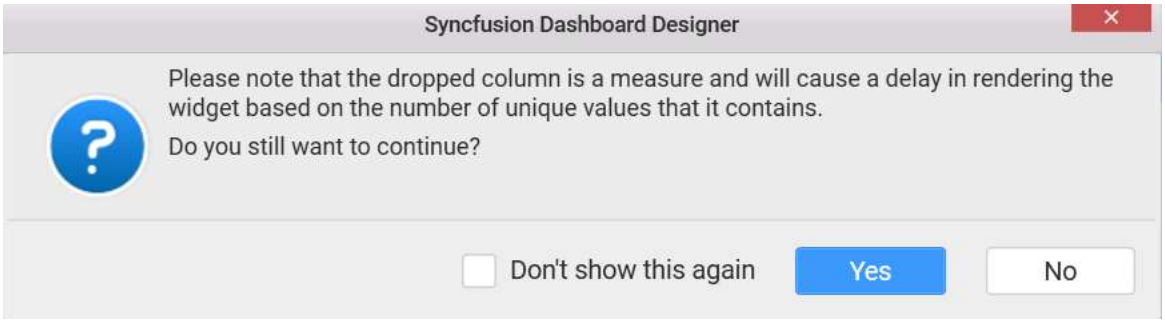
You can add the Dimension into Column field by drag and drop.



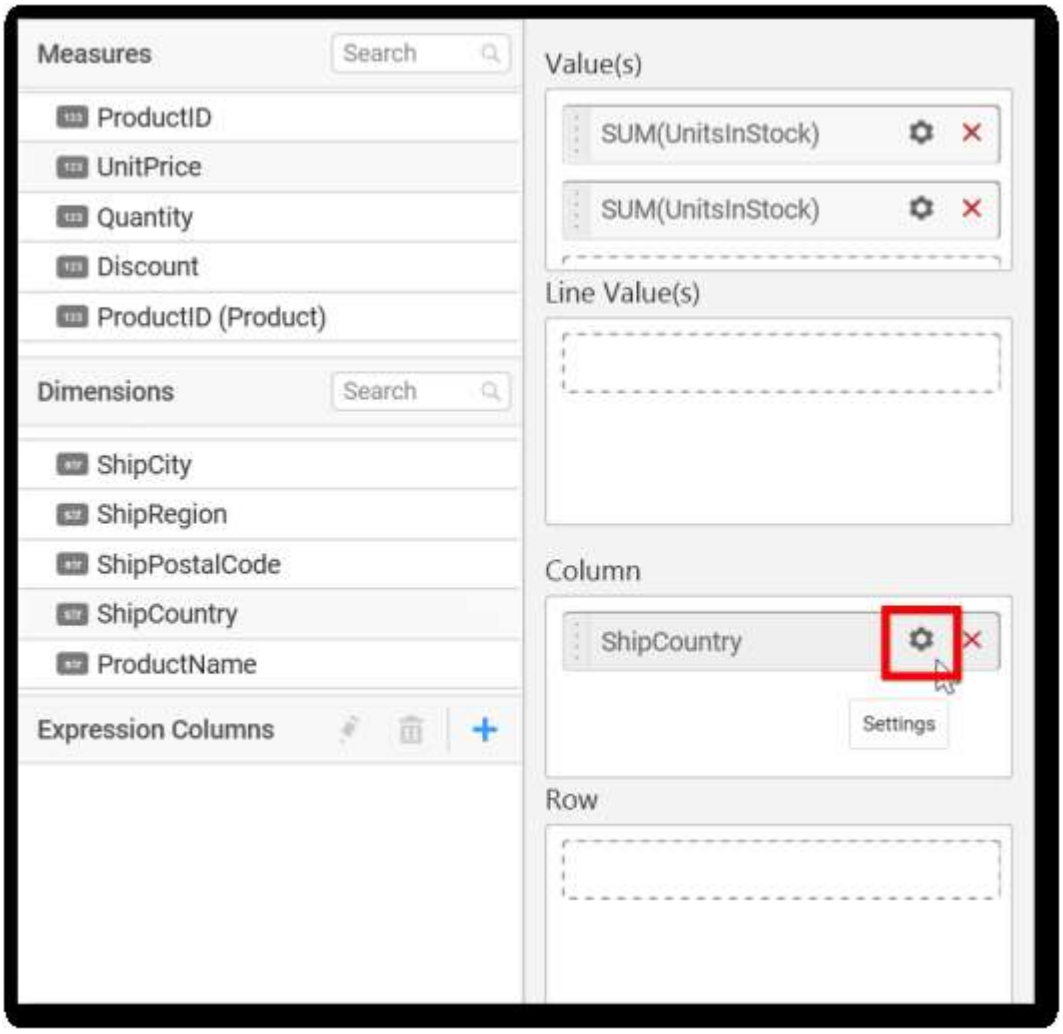


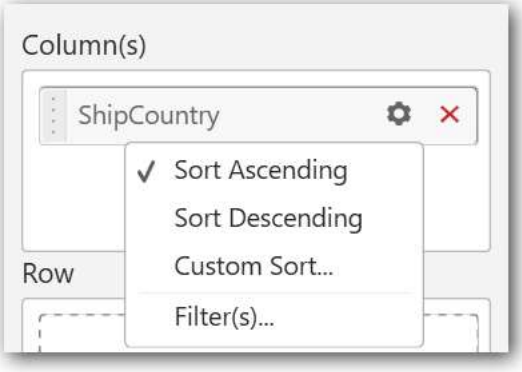
You can also add Measures and Expression Columns into Column(s) field.





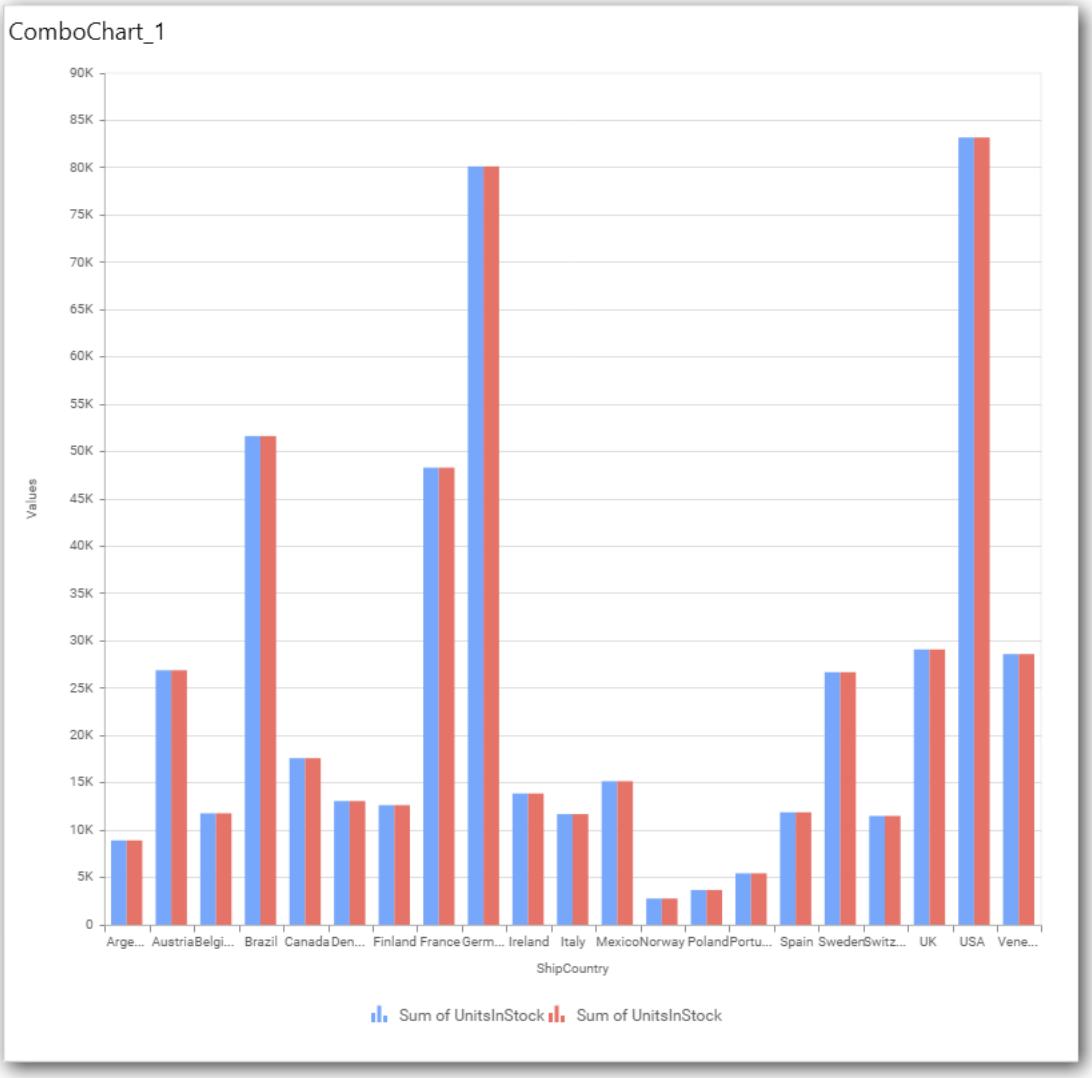
You have options to change the settings.



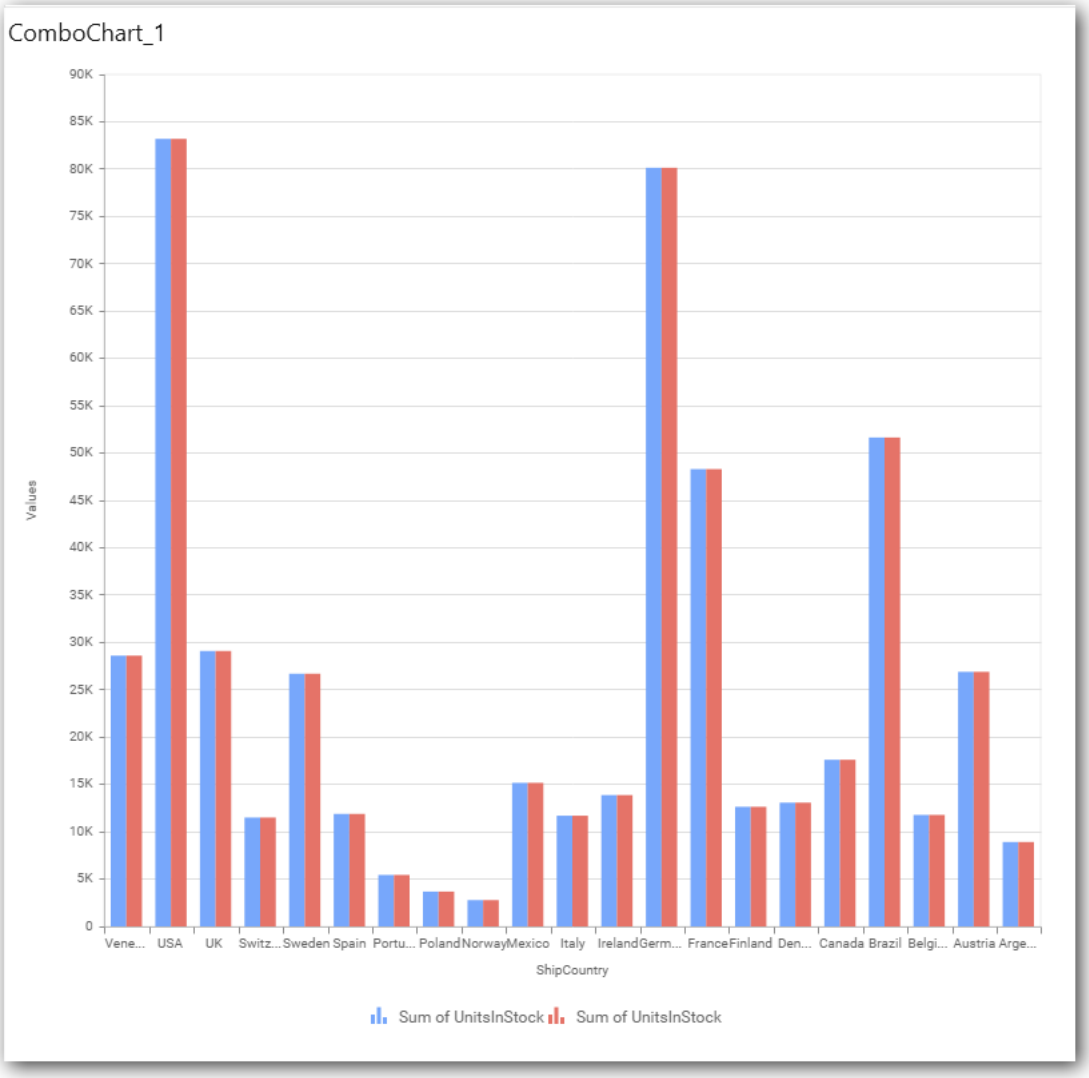


You can sort the chart either in Ascending or Descending series.

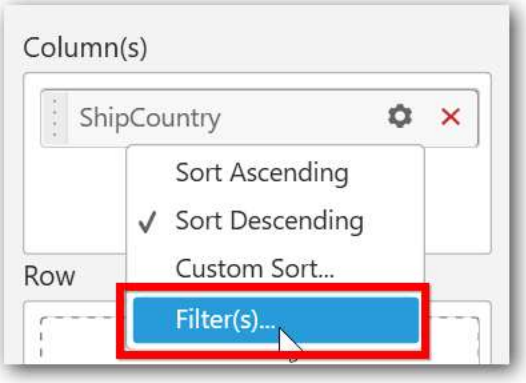
Ascending Order:



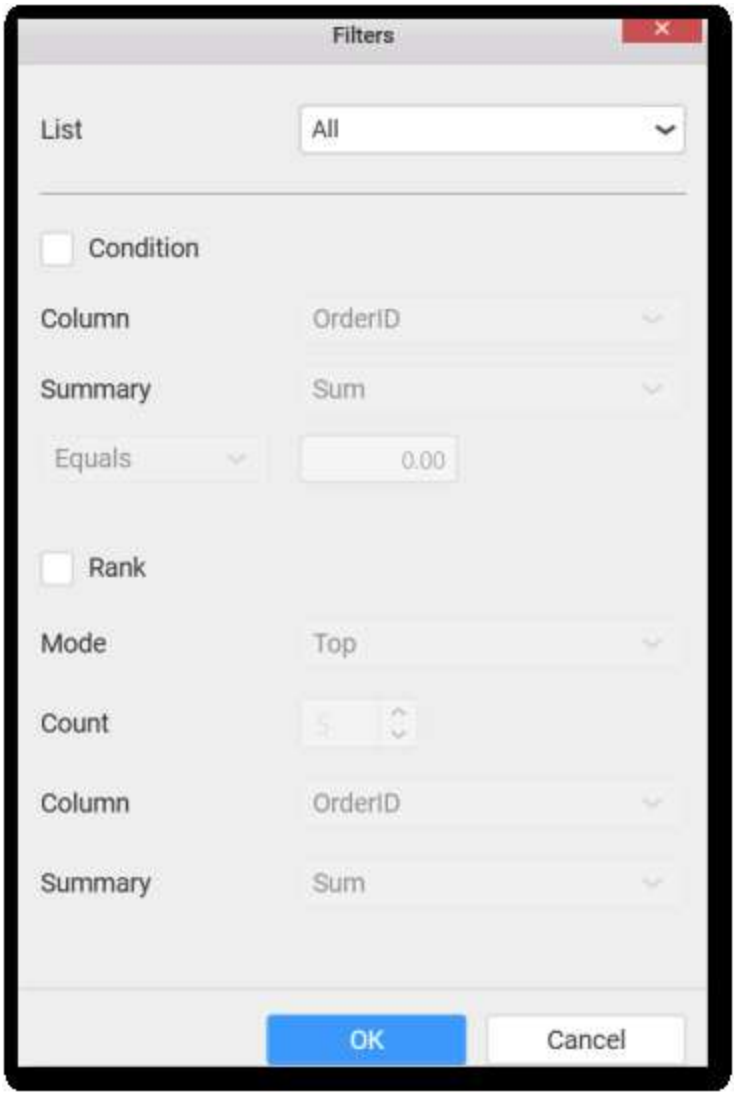
Descending order:

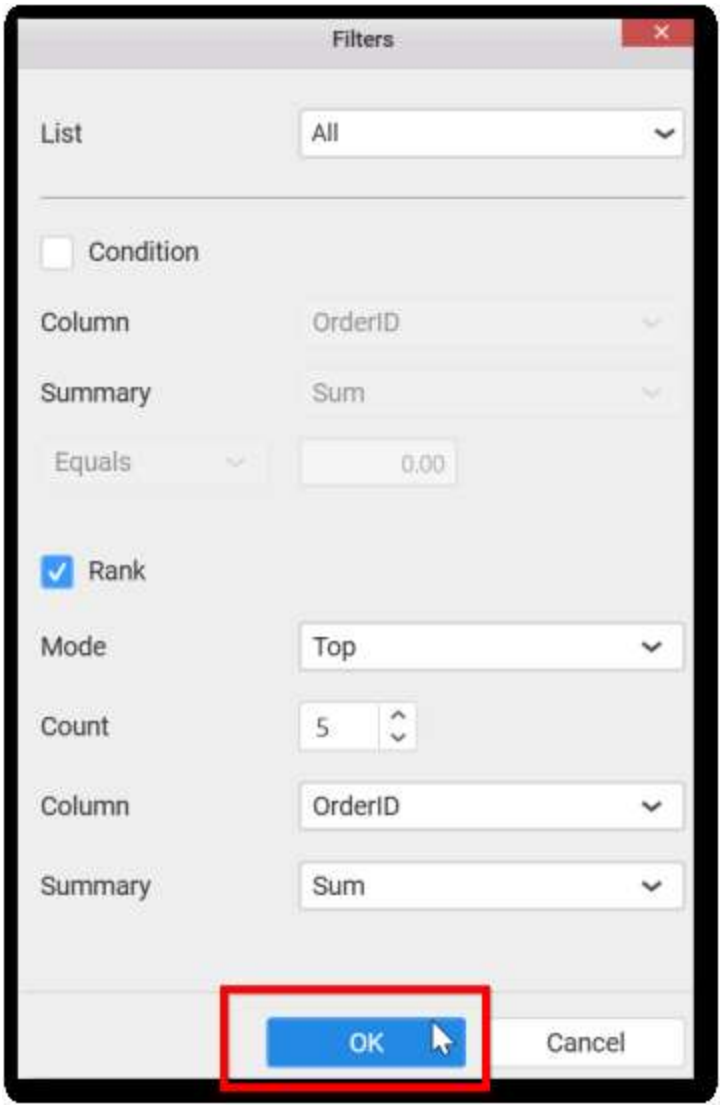


You can apply a filter.

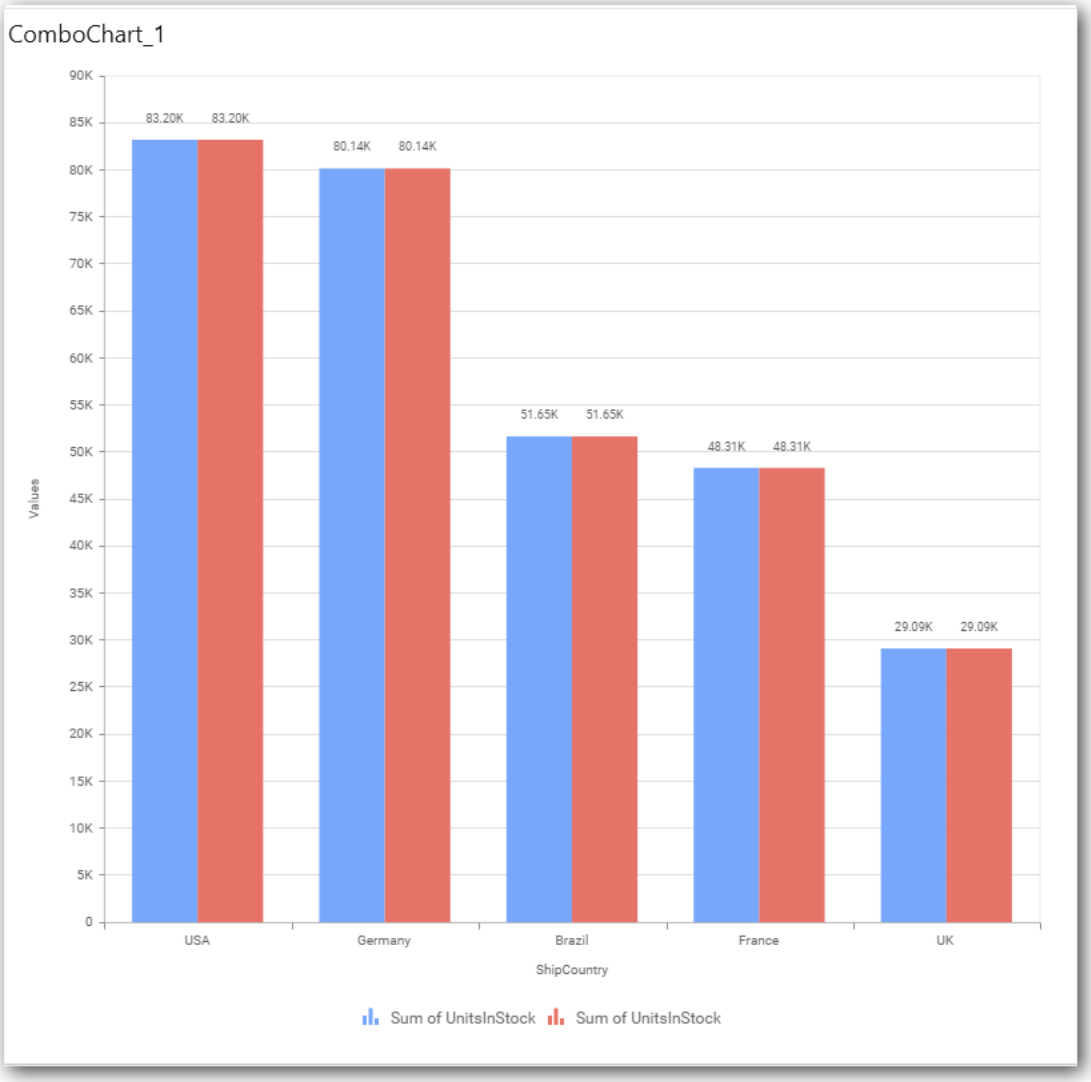


Select the Conditions and Rank you need.

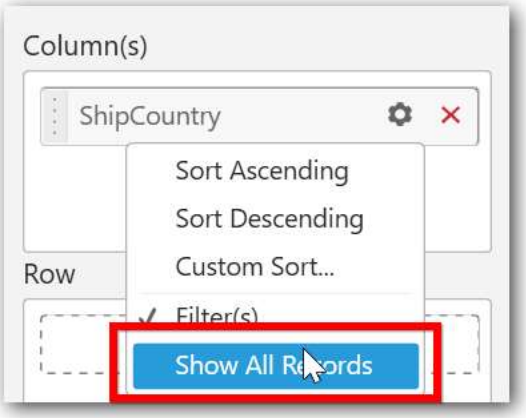




Now the chart will be rendered like this

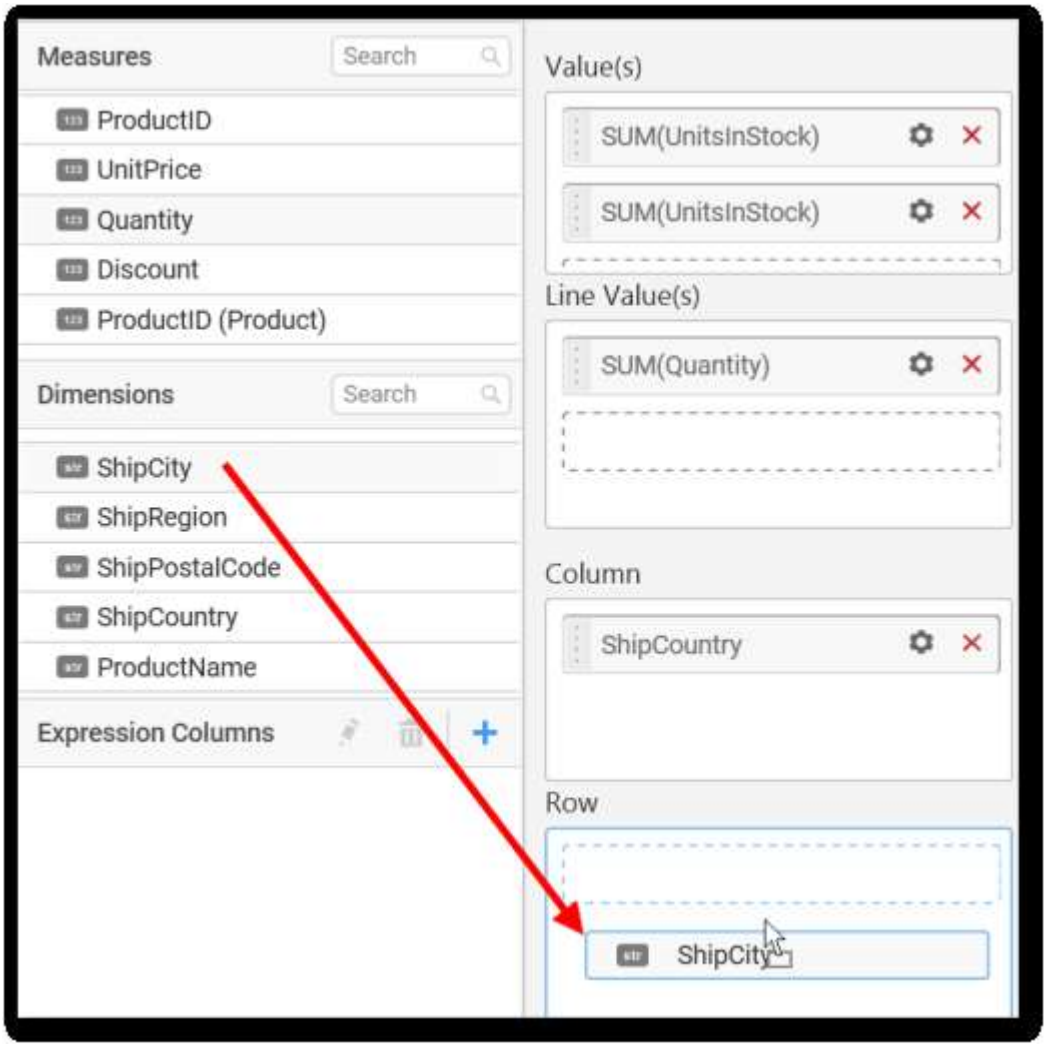


To show all records again click on Show All Records.

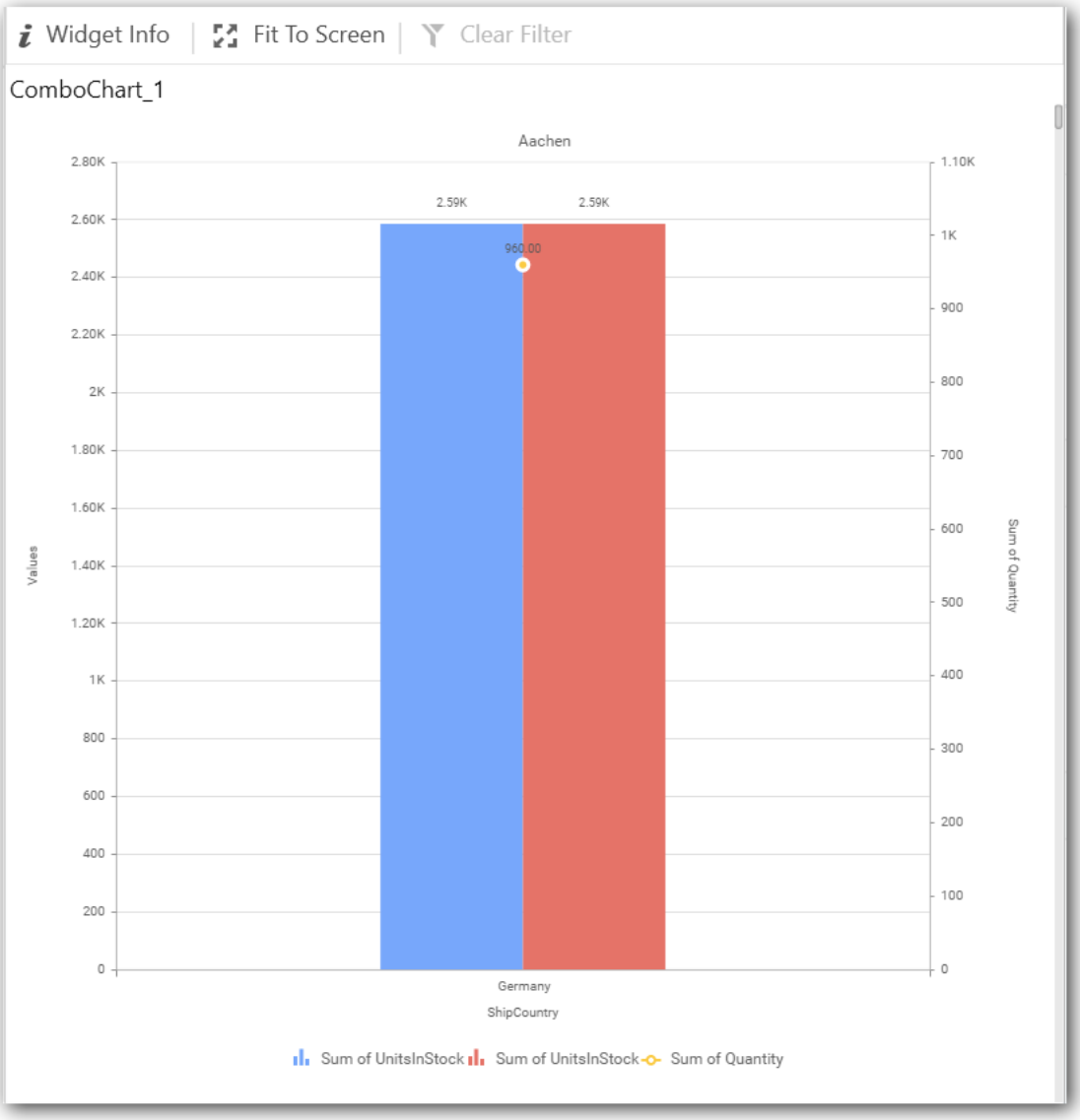


Assigning Row

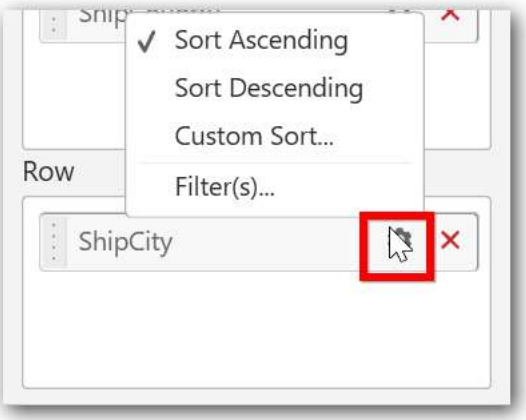
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



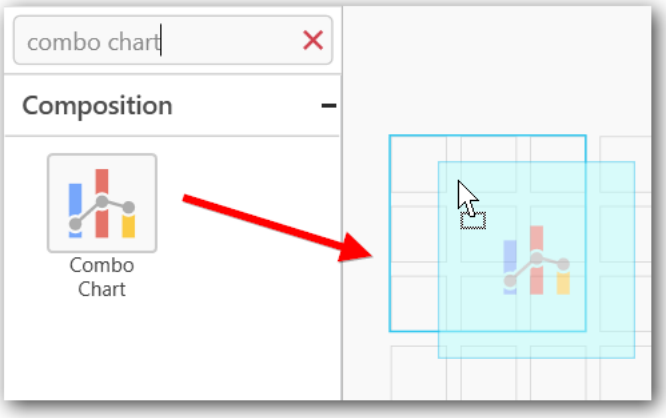
You have Settings options similar to Column(s).



How to configure the SSAS data to Combo Chart?

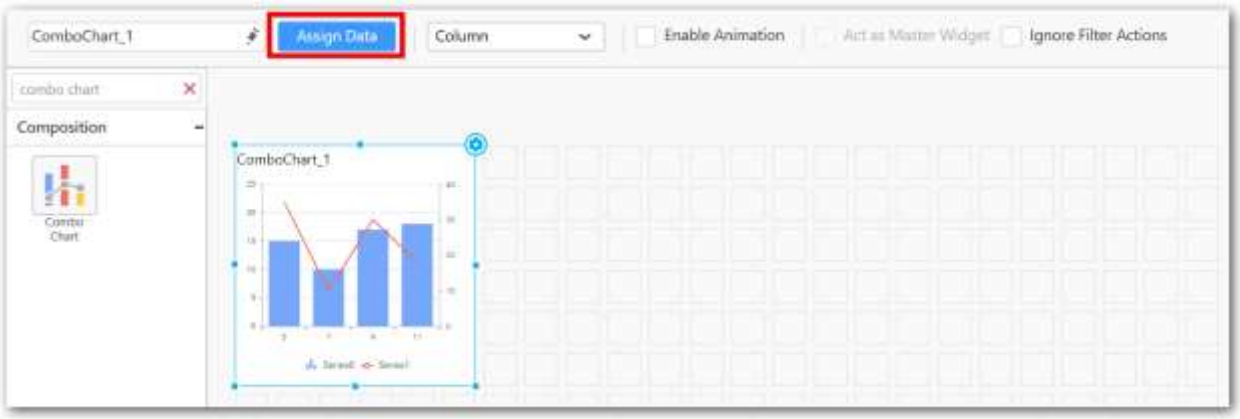
Following steps illustrates configuration of SSAS data to combo chart

Drag and drop the Combo Chart widget to canvas and resize it to your required size.

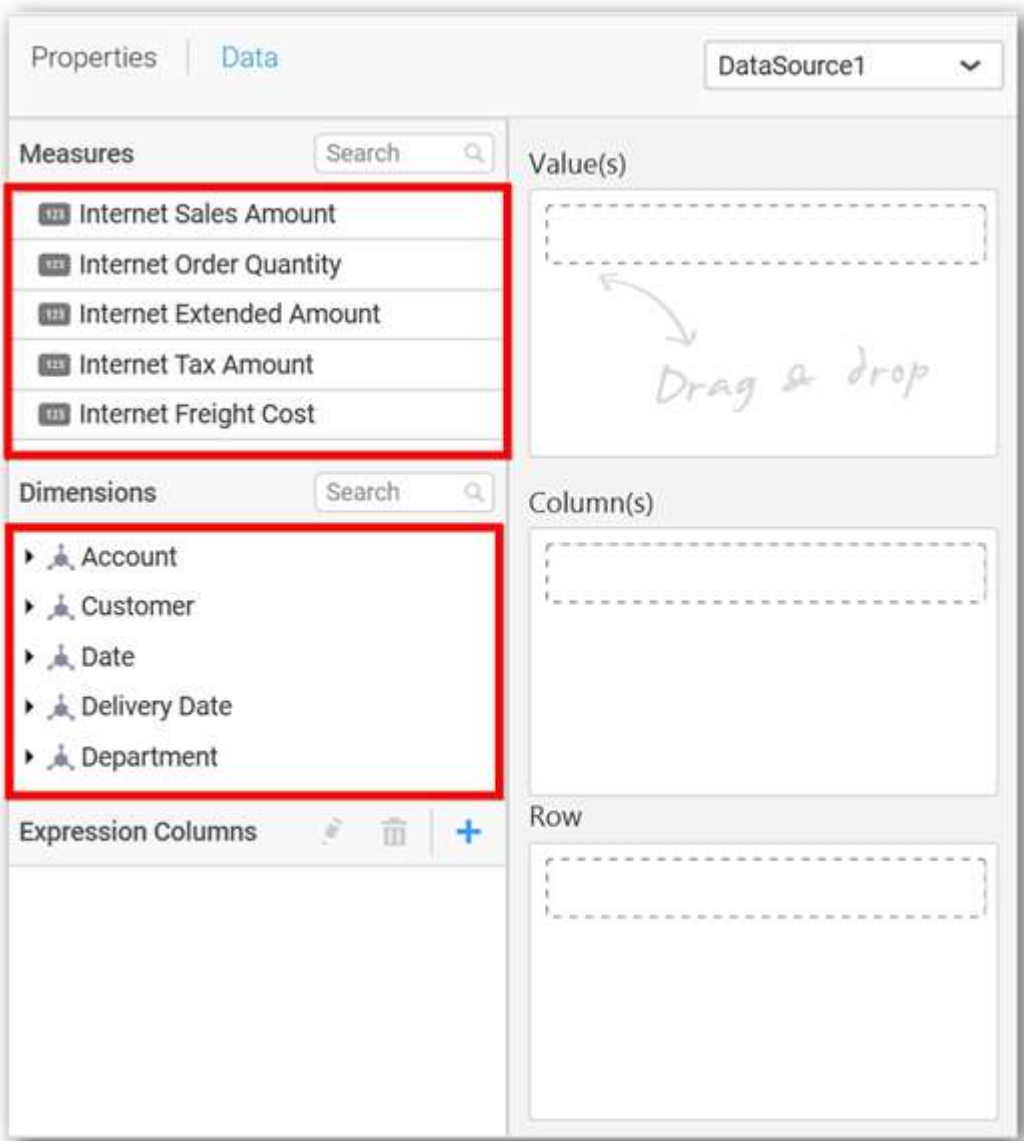


Select the dropped widget using mouse.

Click the Assign Data button in the toolbar.

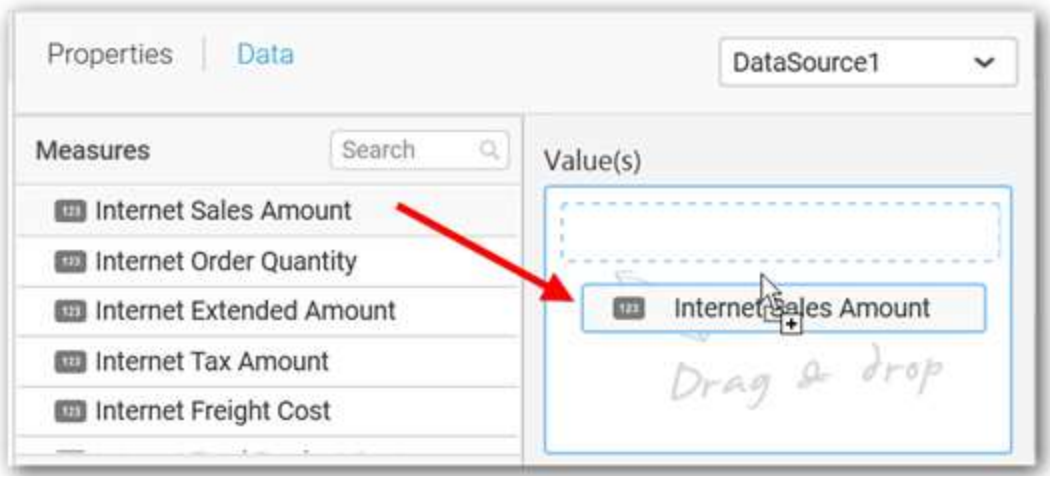


A Data pane will be opened with available Measures and Dimensions.

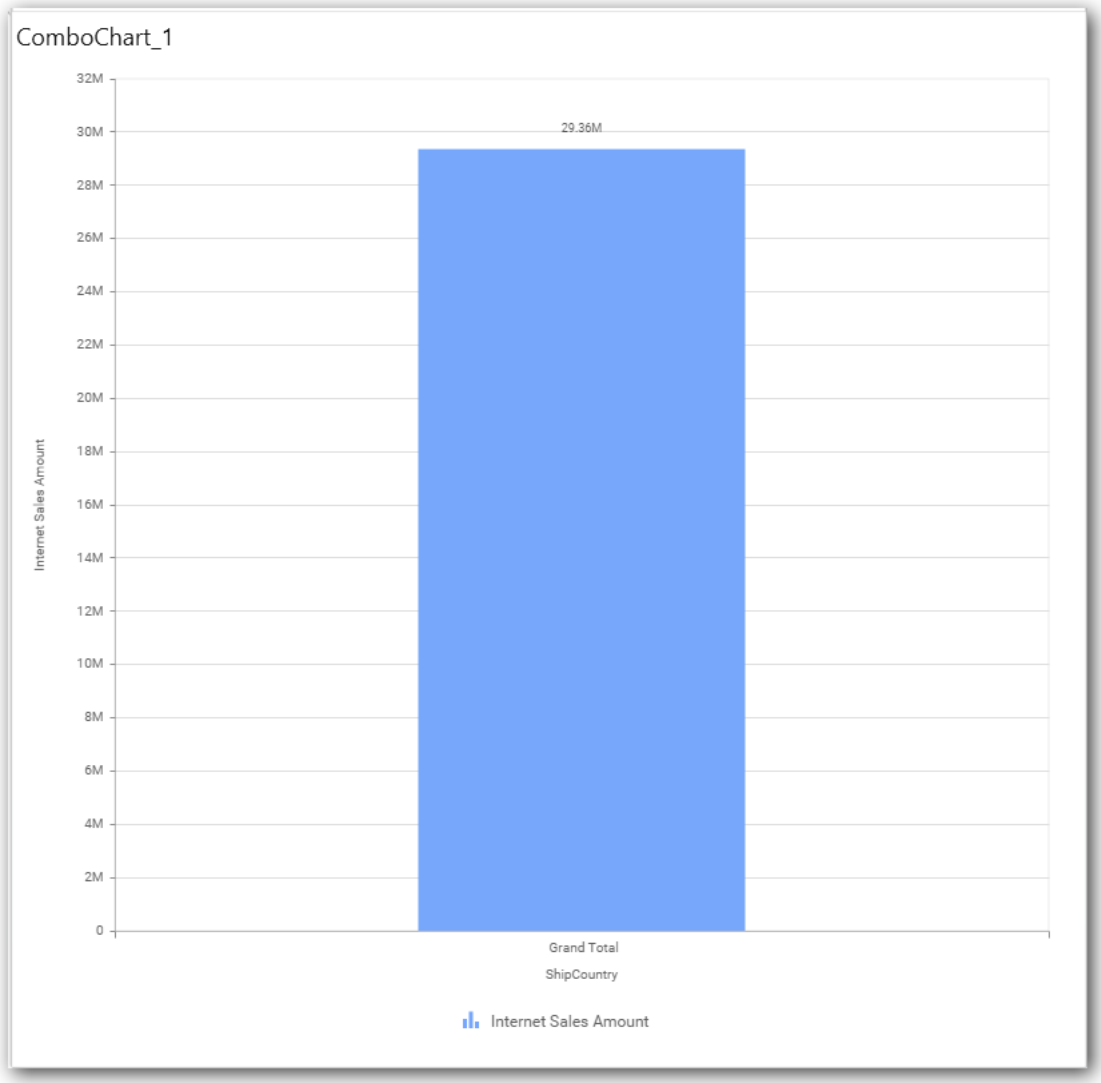


Assigning Value(s)

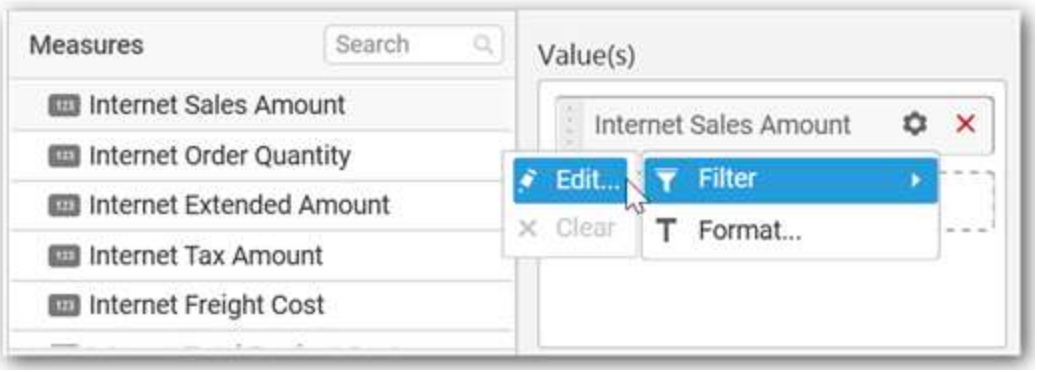
Drag and drop a column under Measures category into Value(s) section.



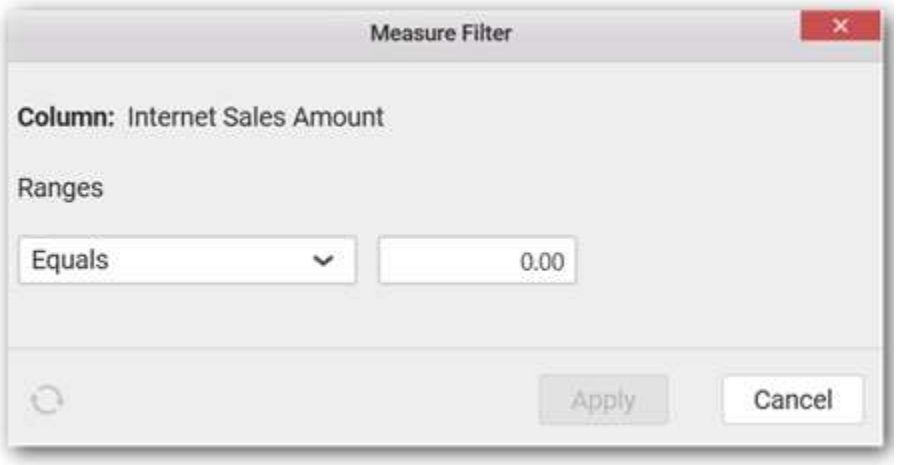
Now the chart will be rendered like this.



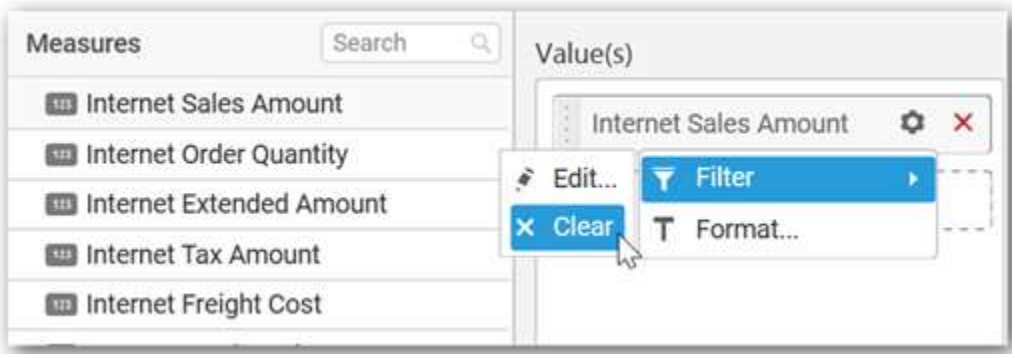
Define the filter criteria to match through choosing Edit option in Filter menu item.



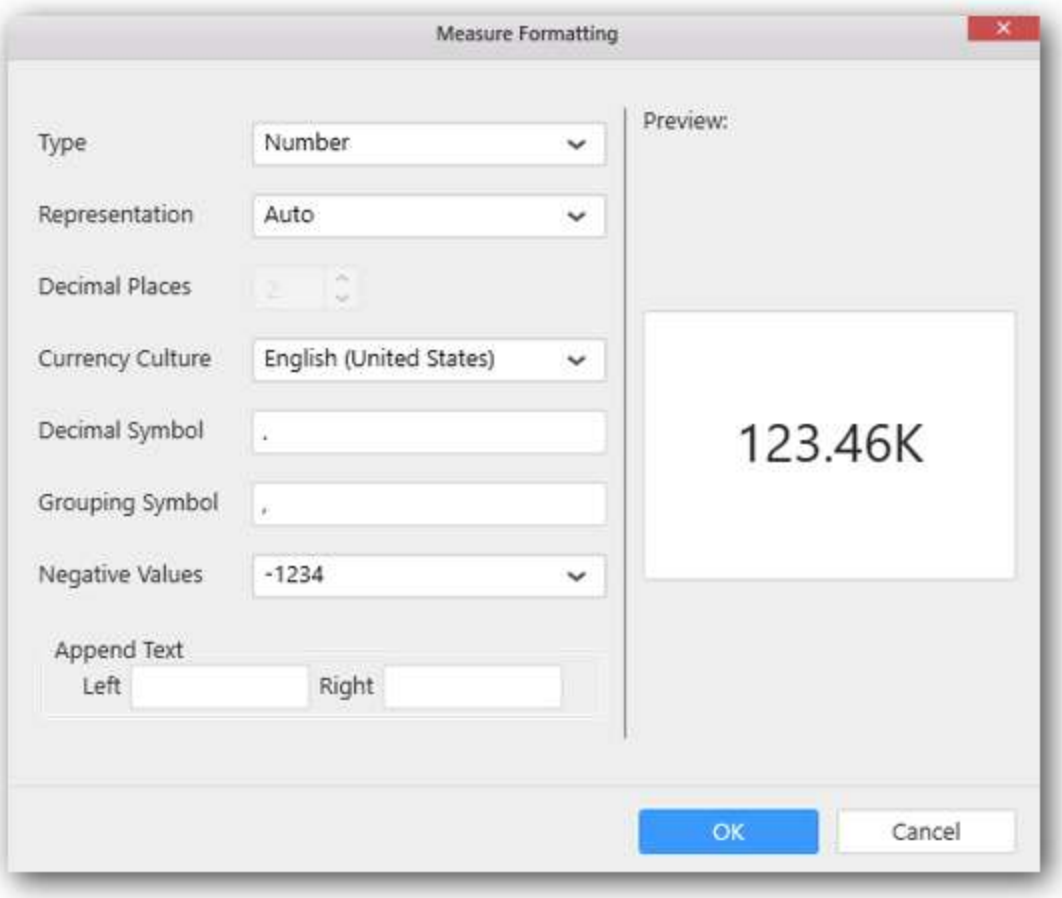
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



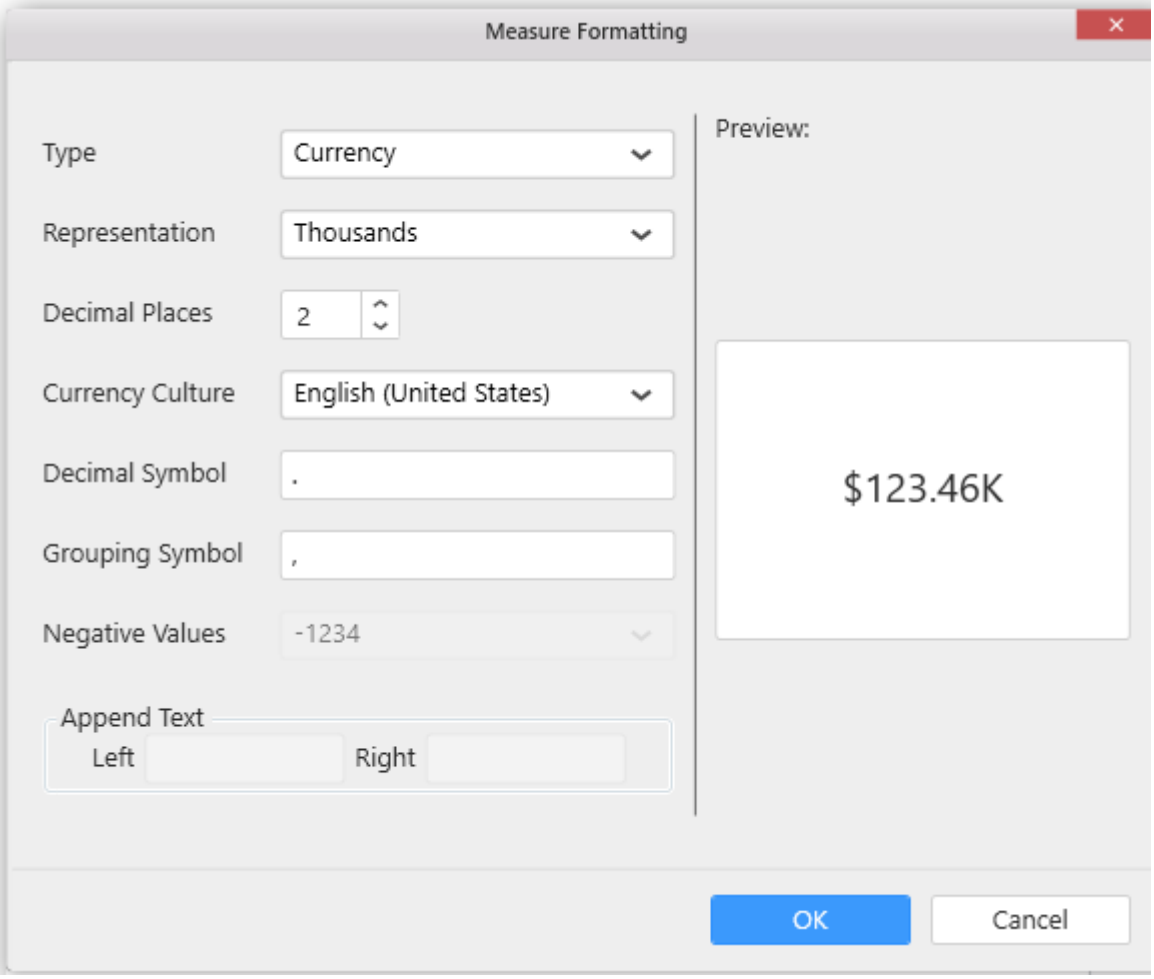
Select Clear option to clear the defined filter.



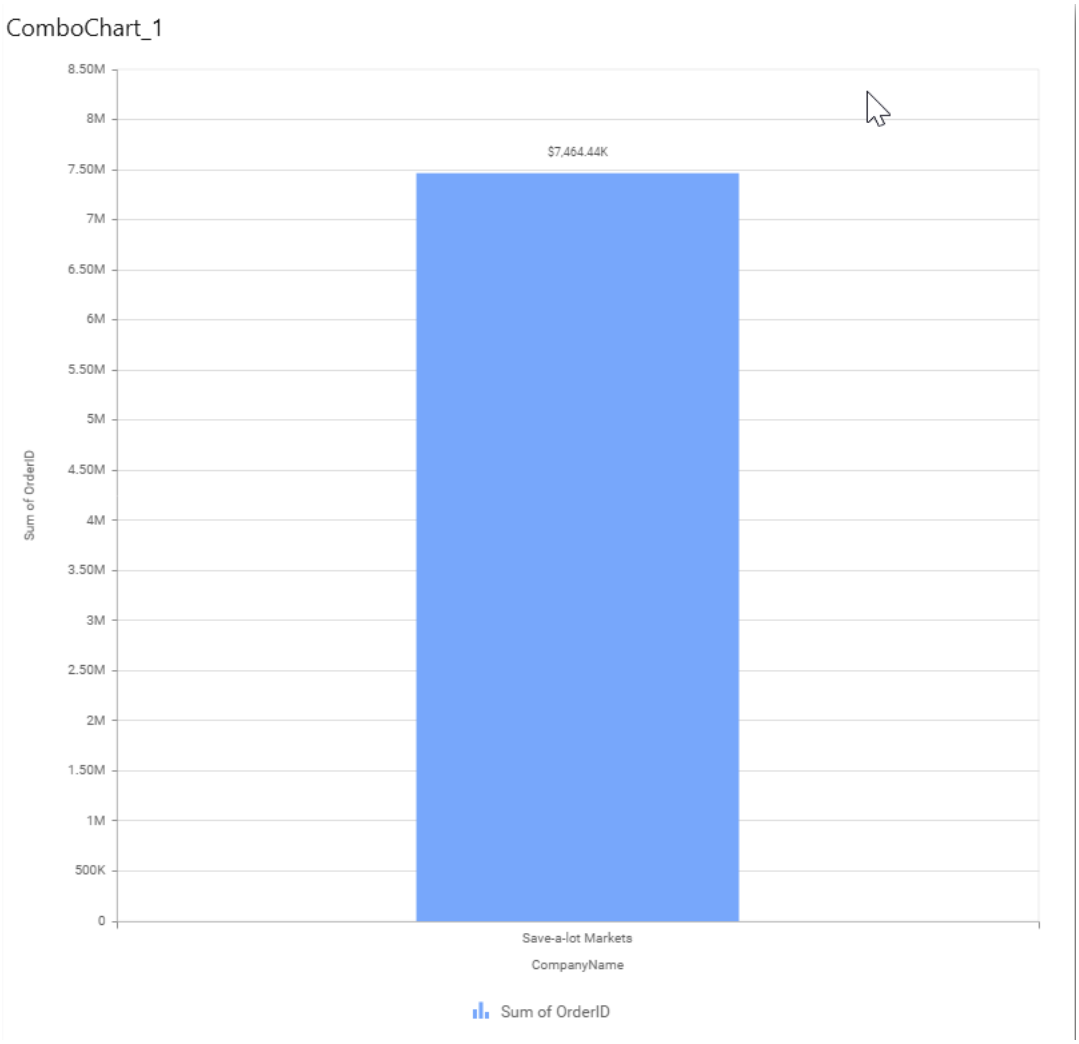
Select Format option to define the display format to the values in the column through Measure Formatting window.



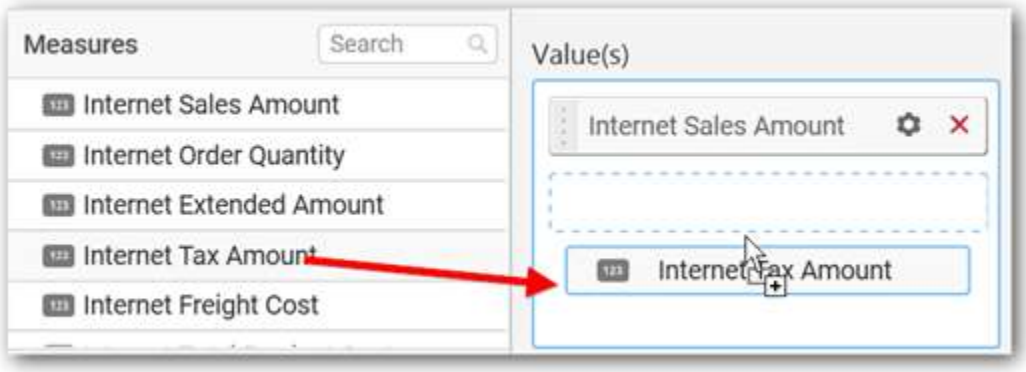
Choose the options you need and click OK.

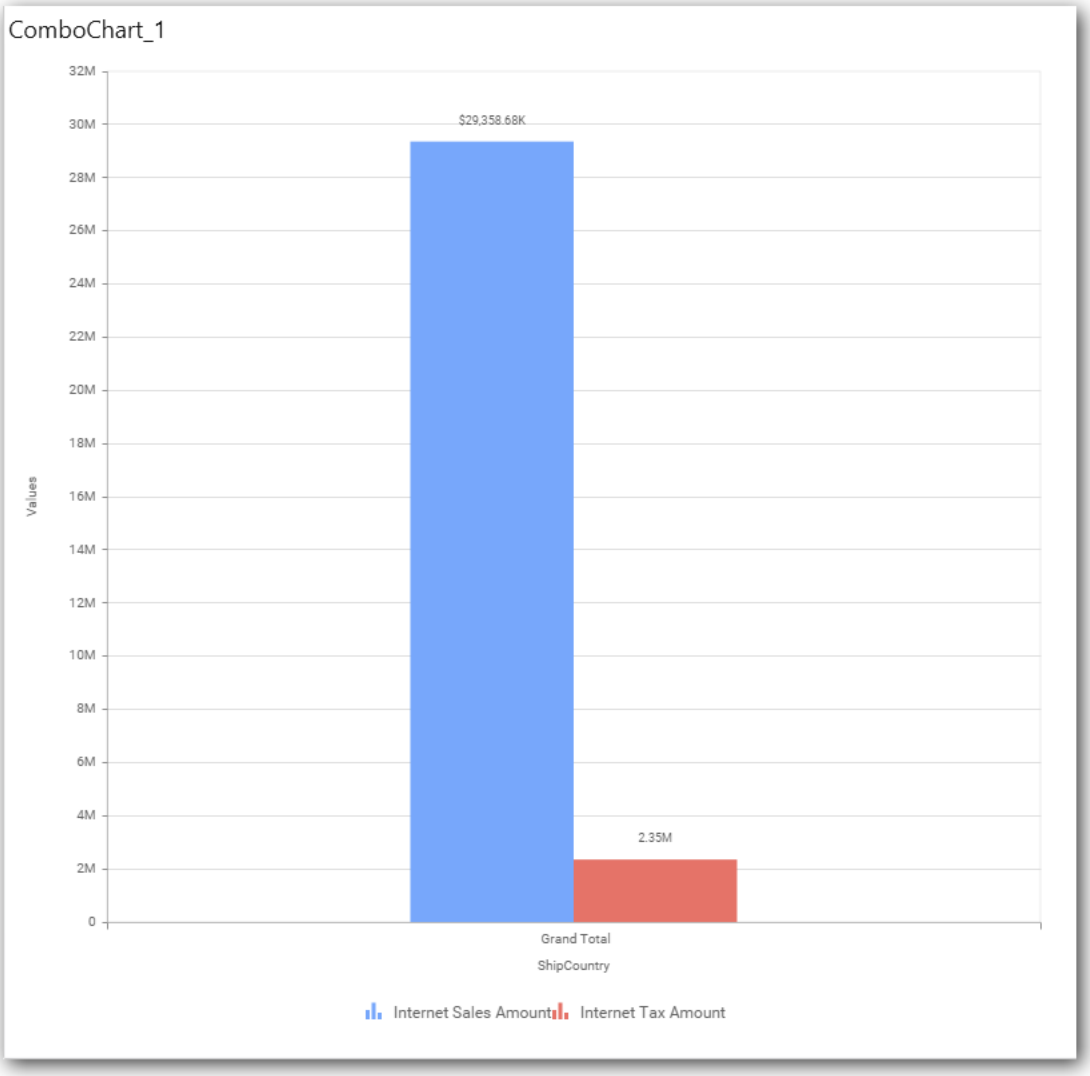


Now the Chart will be rendered like this.



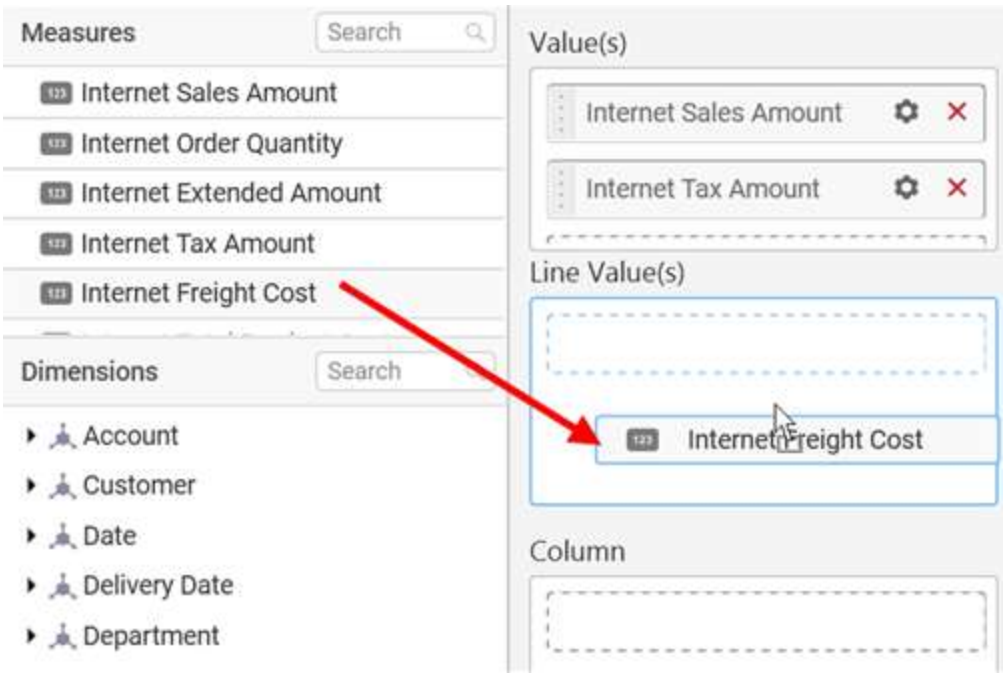
You can also add more than one column to the Value(s) section.

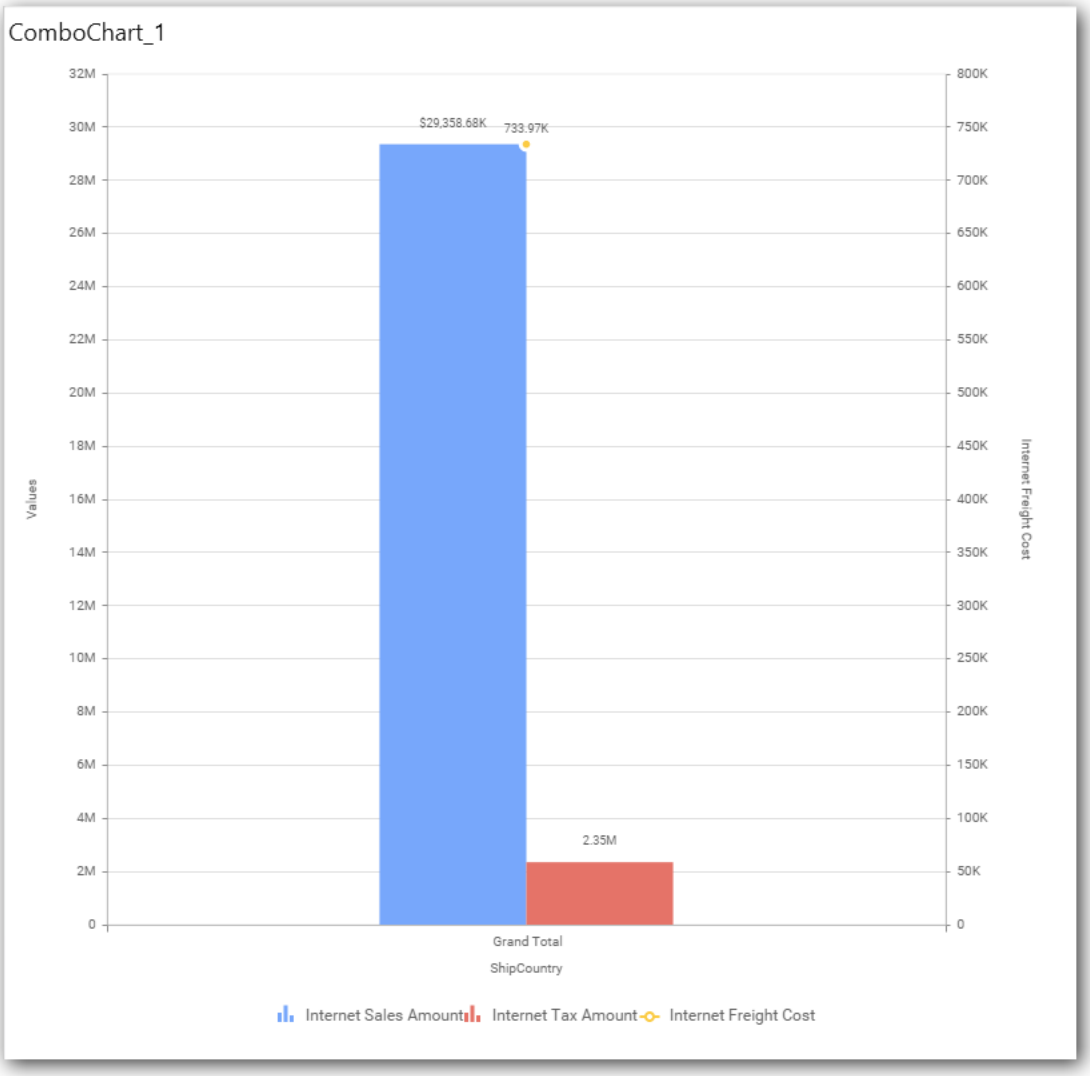




Assigning Line Value

Add a Measure into Line Value(s) section through drag and drop.

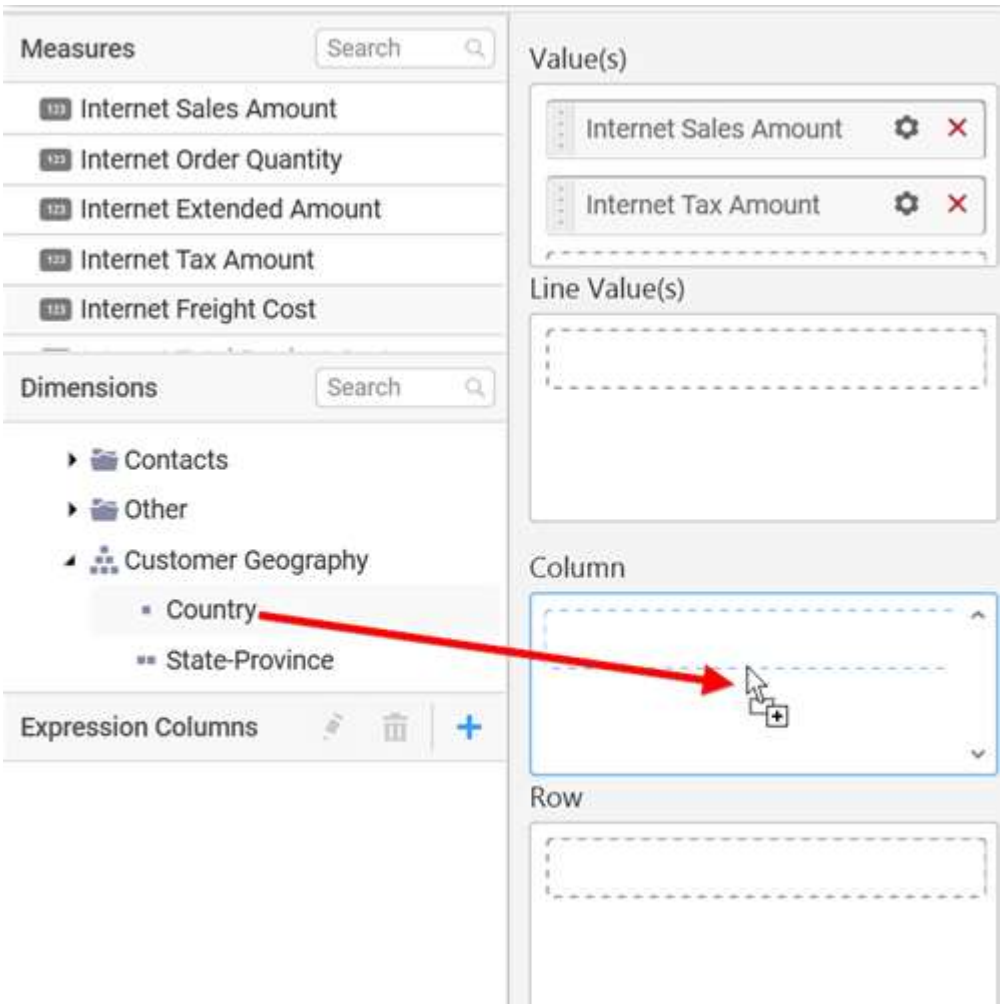


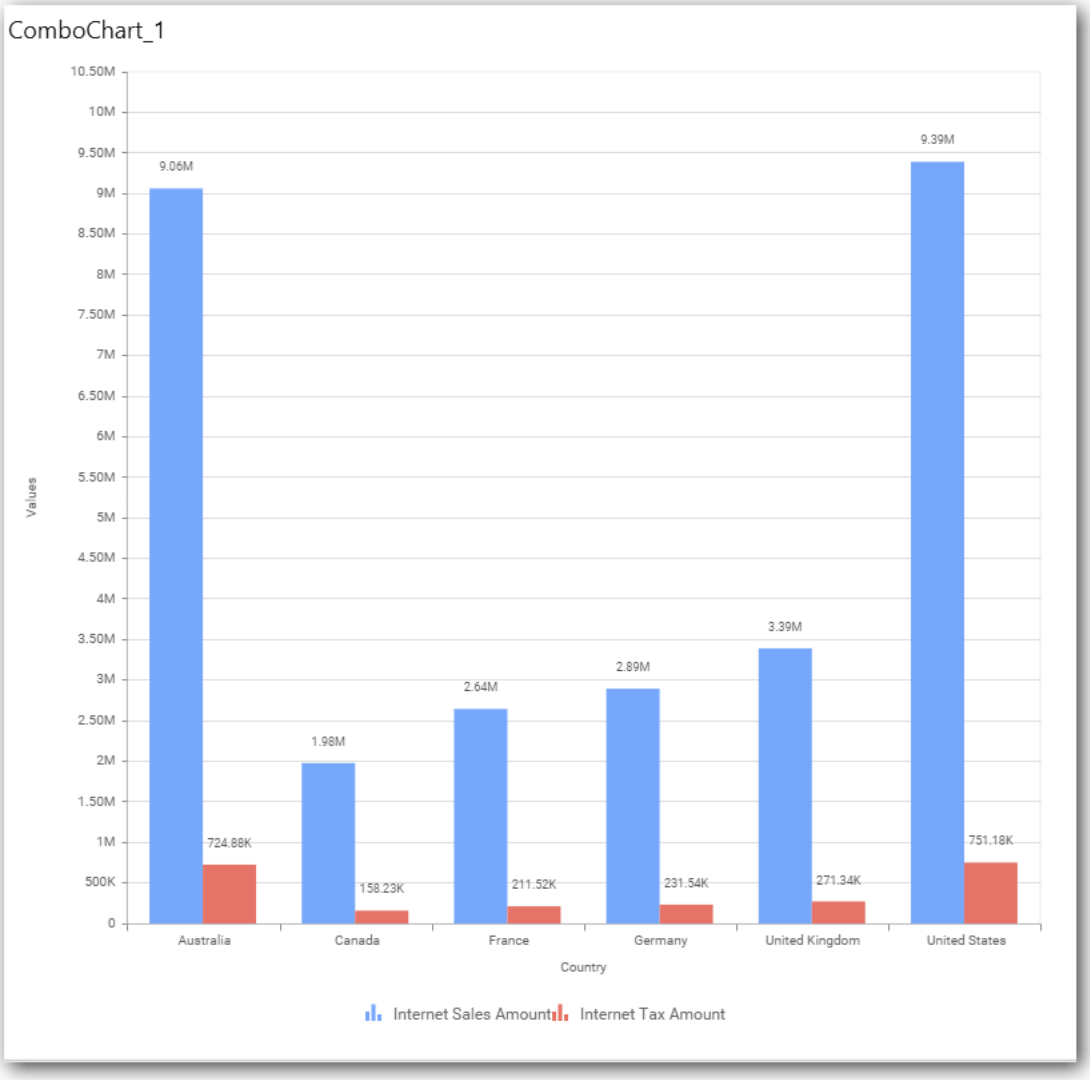


You can also add more than one column to the Line Value(s) section.

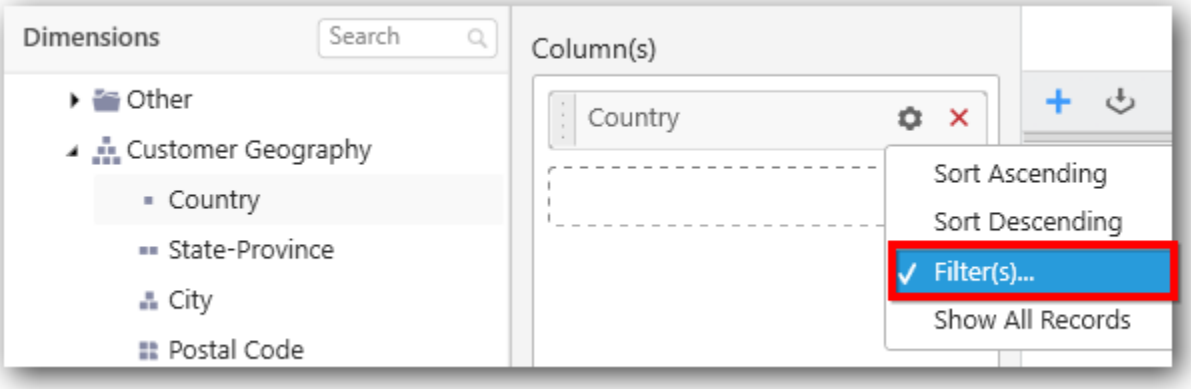
Assigning Column(s)

Add a dimension level or hierarchy into Column section through drag and drop

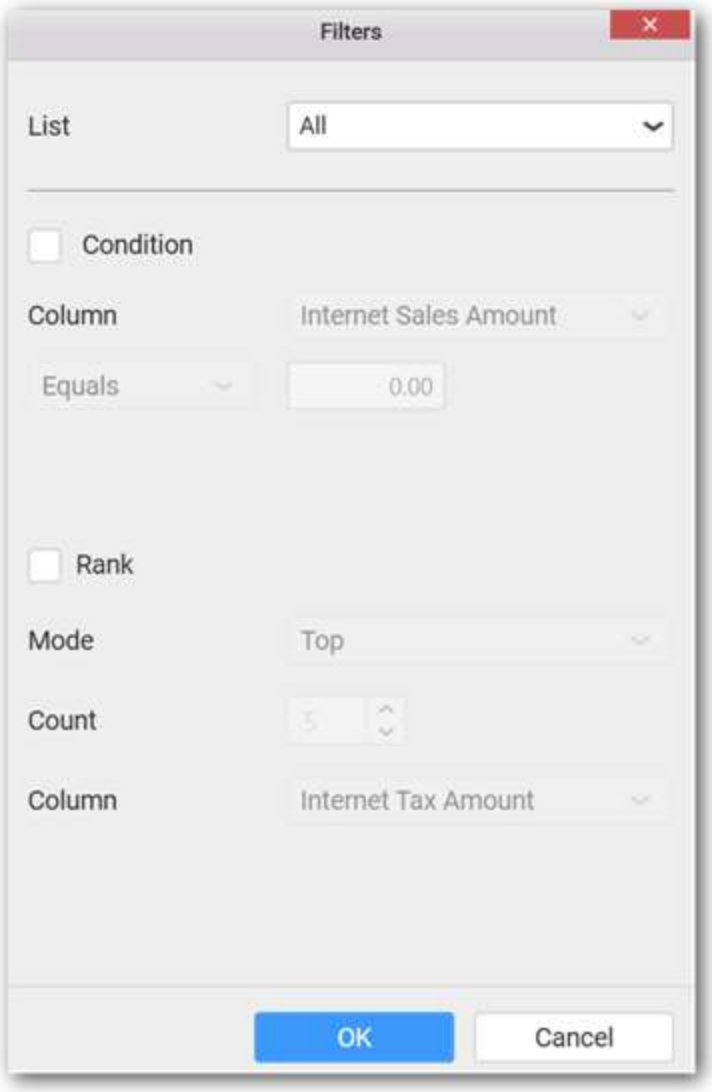




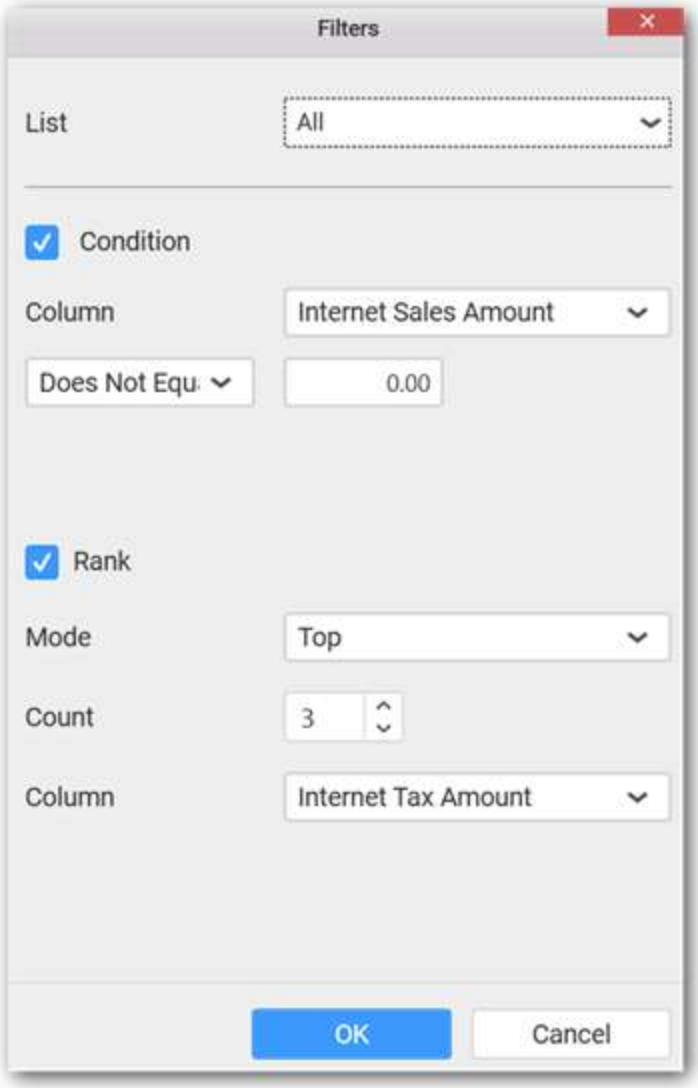
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



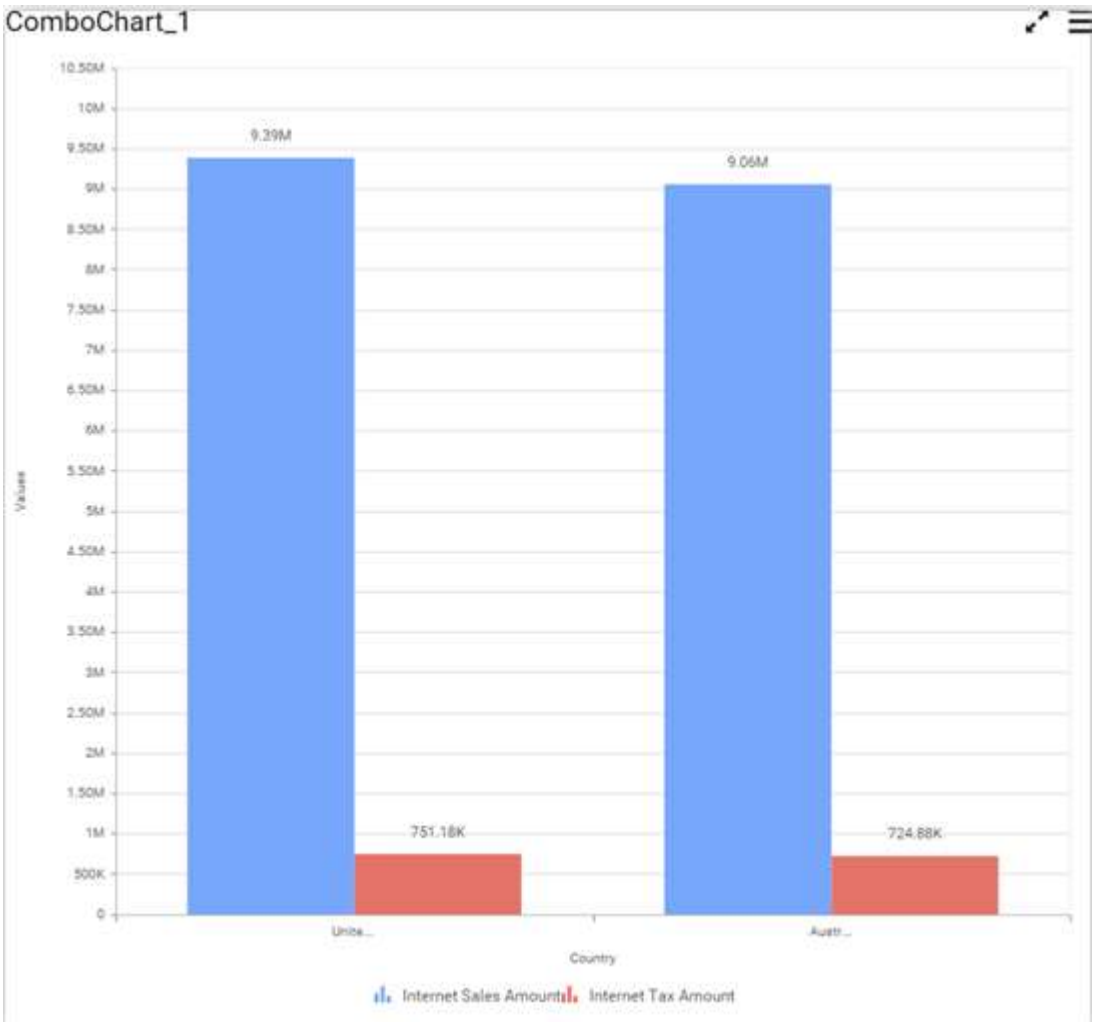
Select Filter(s)... to launch the Filters window.



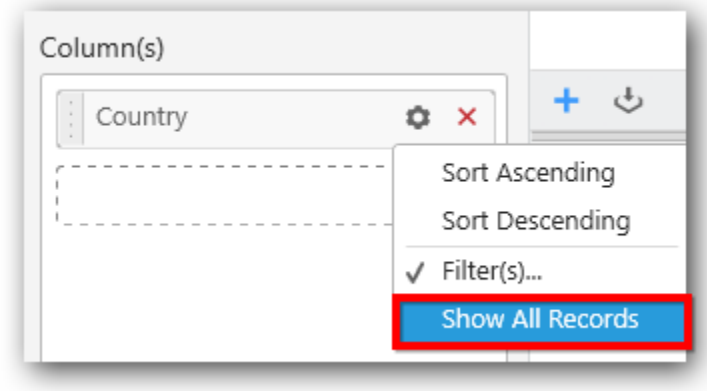
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

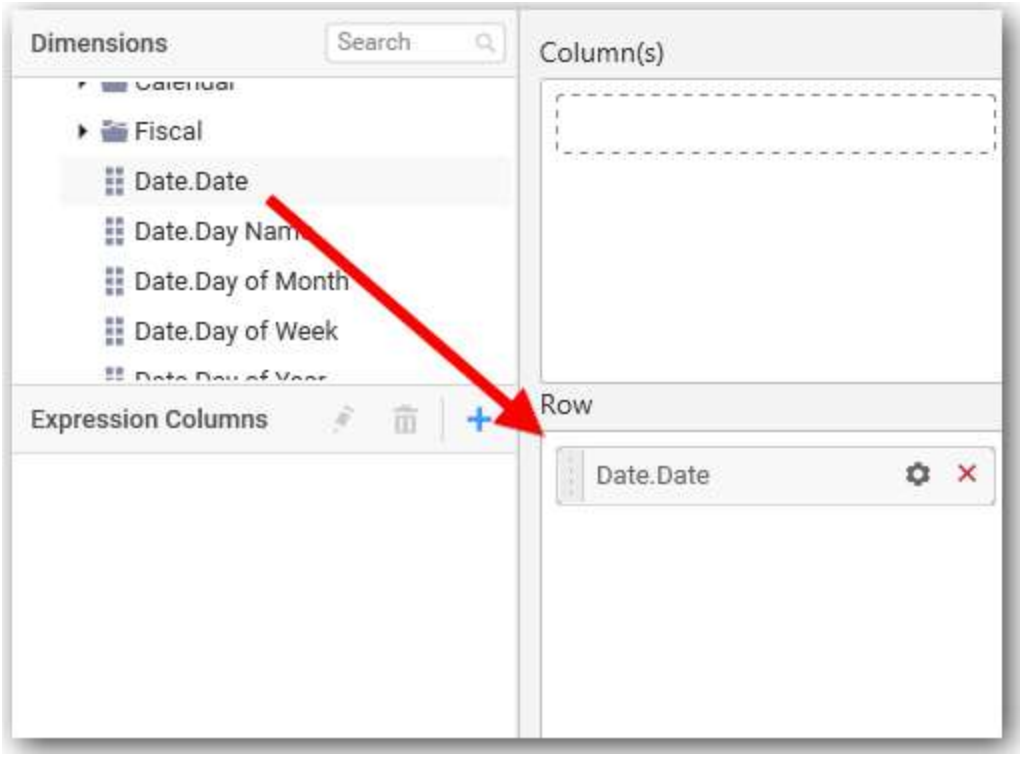


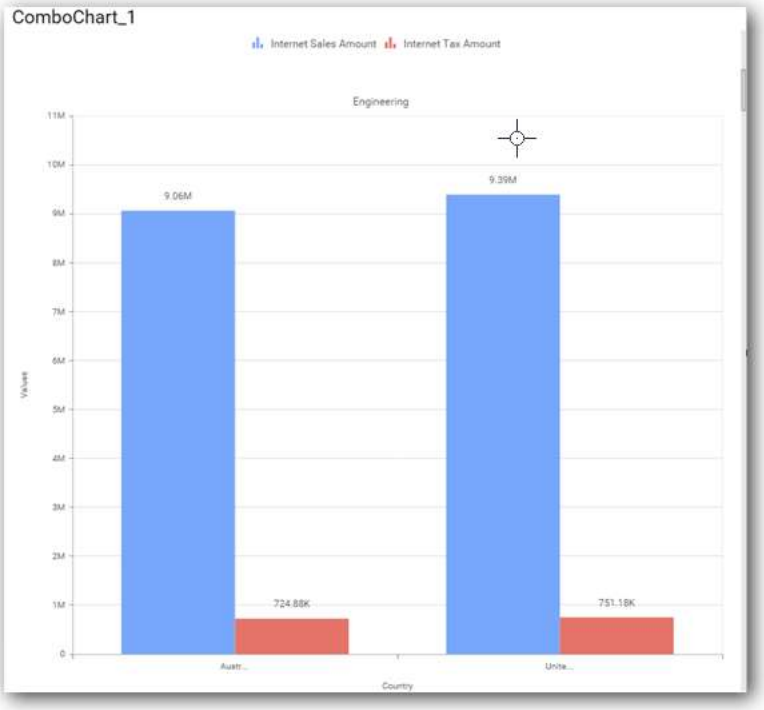
To show all records again click on Show All Records.



Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart.



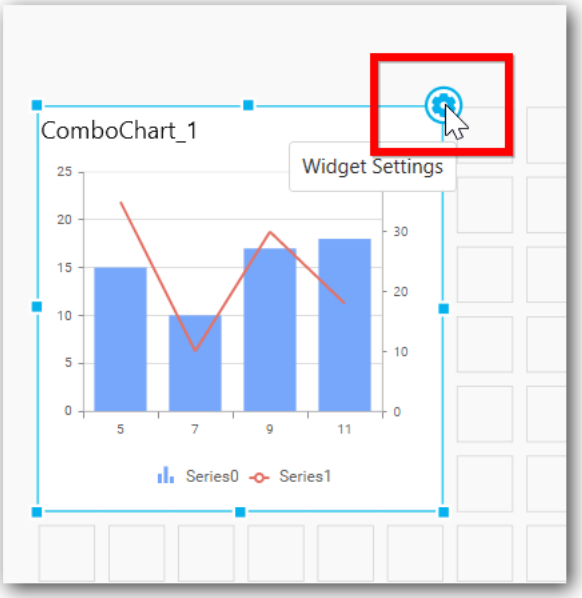


How to format Combo Chart?

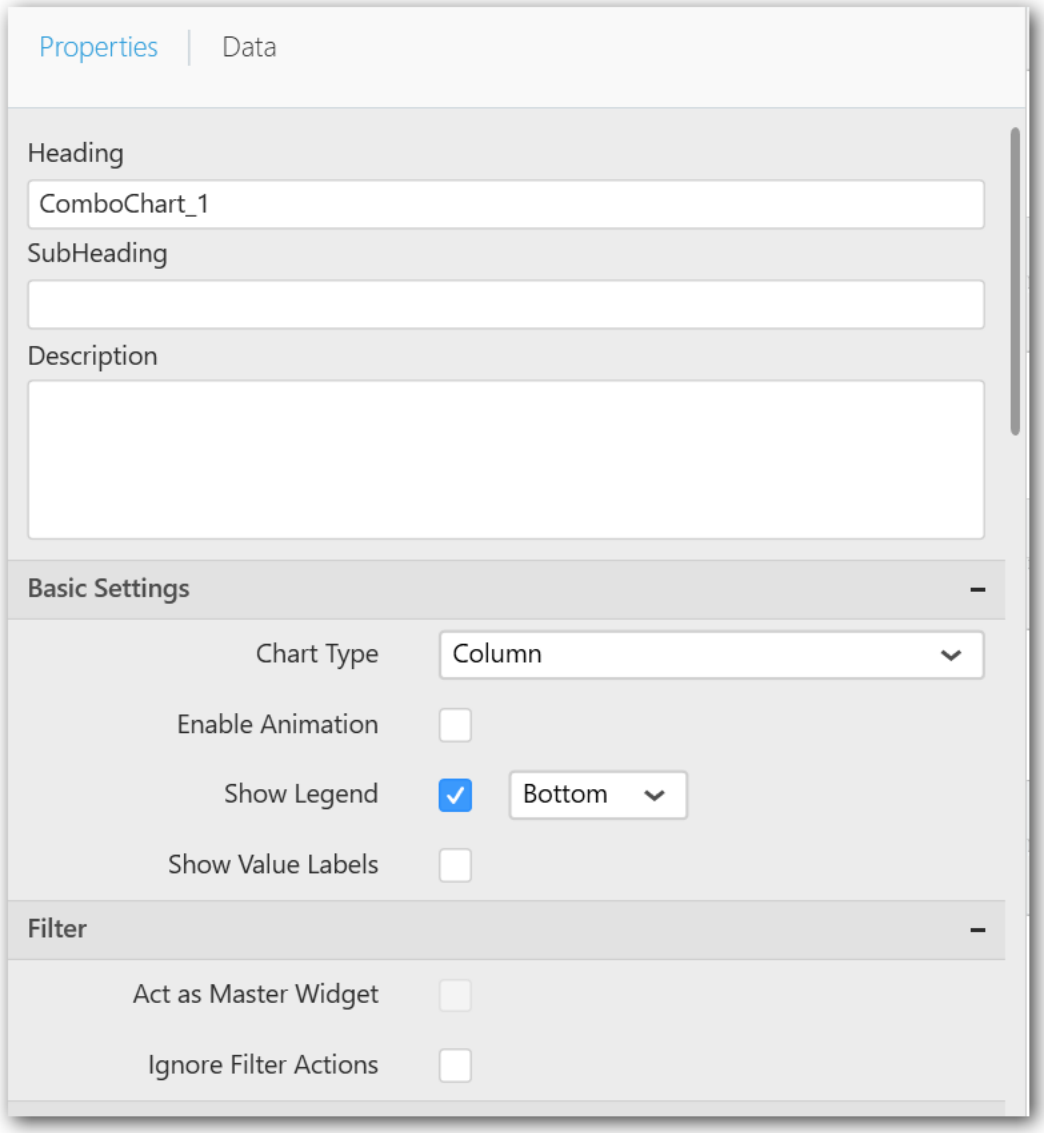
You can format the combo chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into combo chart follow the steps

1. Drag and drop the combo chart into canvas and resize it to your required size.
2. Configure the data into combo chart.
3. Focus on the combo chart and Click on Widget Settings.

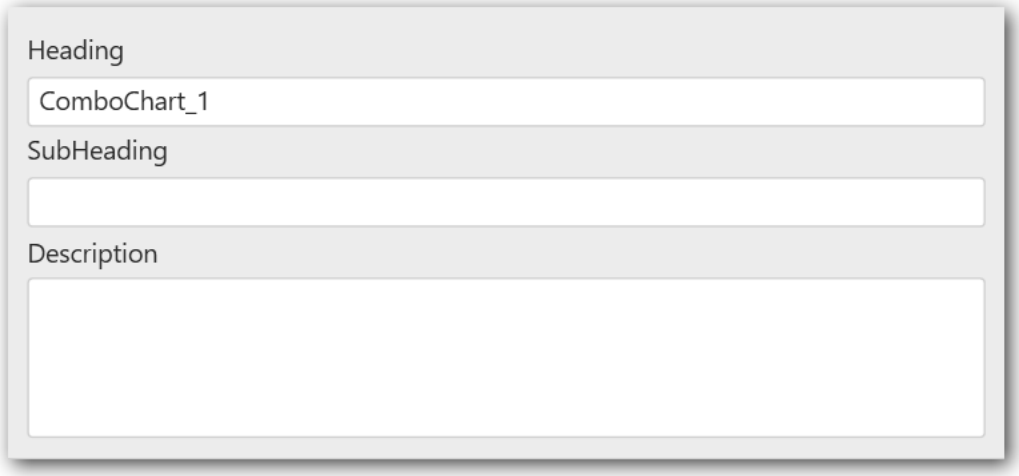


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this combo chart widget.

SubHeading

This allows you to set sub-title for this combo chart widget.

Description

This allows you to set description for this combo chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

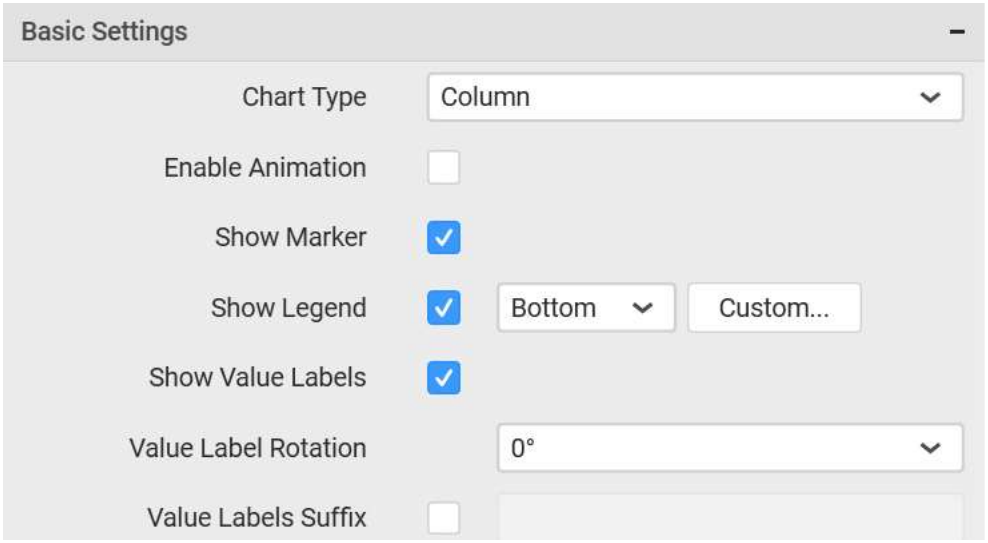


Chart Type

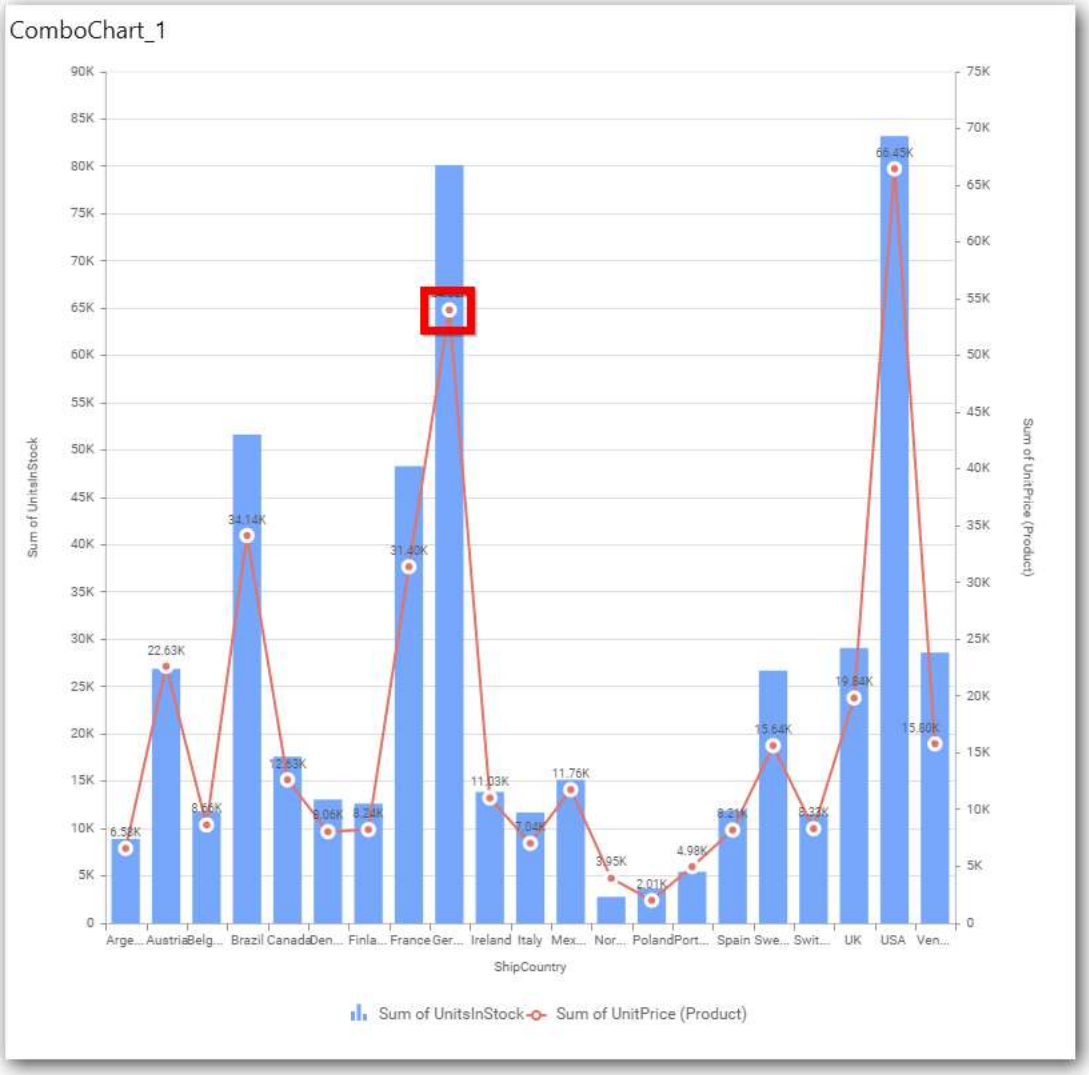
This allows you to switch the widget view from current chart type to another chart type. To selecting chart type through combo box.

Enable Animation

This allows you to enable the series rendering in animated mode.

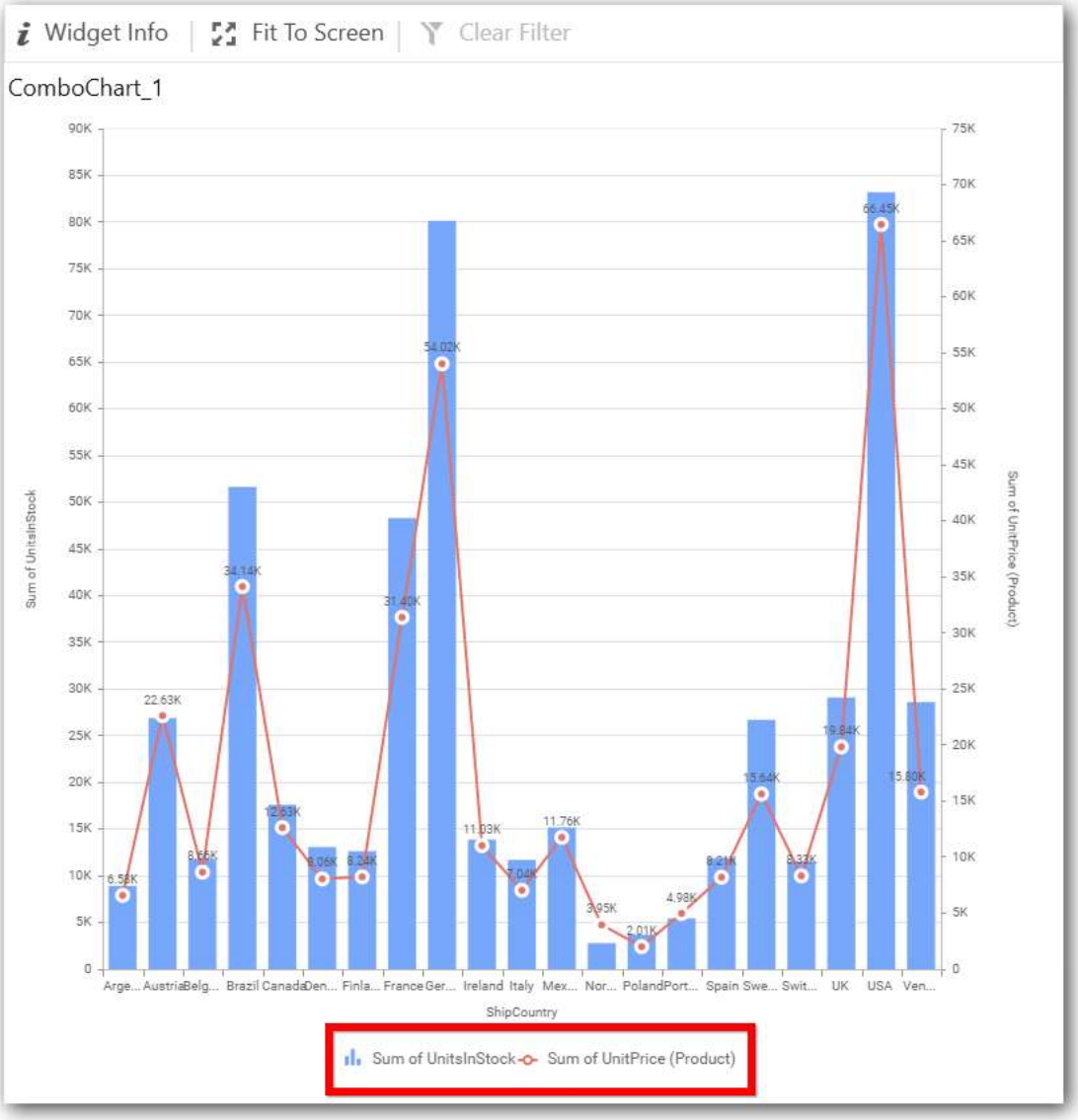
Show Data Marker

This allows you to toggle the visibility of marker from label to adorn each data point in chart series.



Show Legend

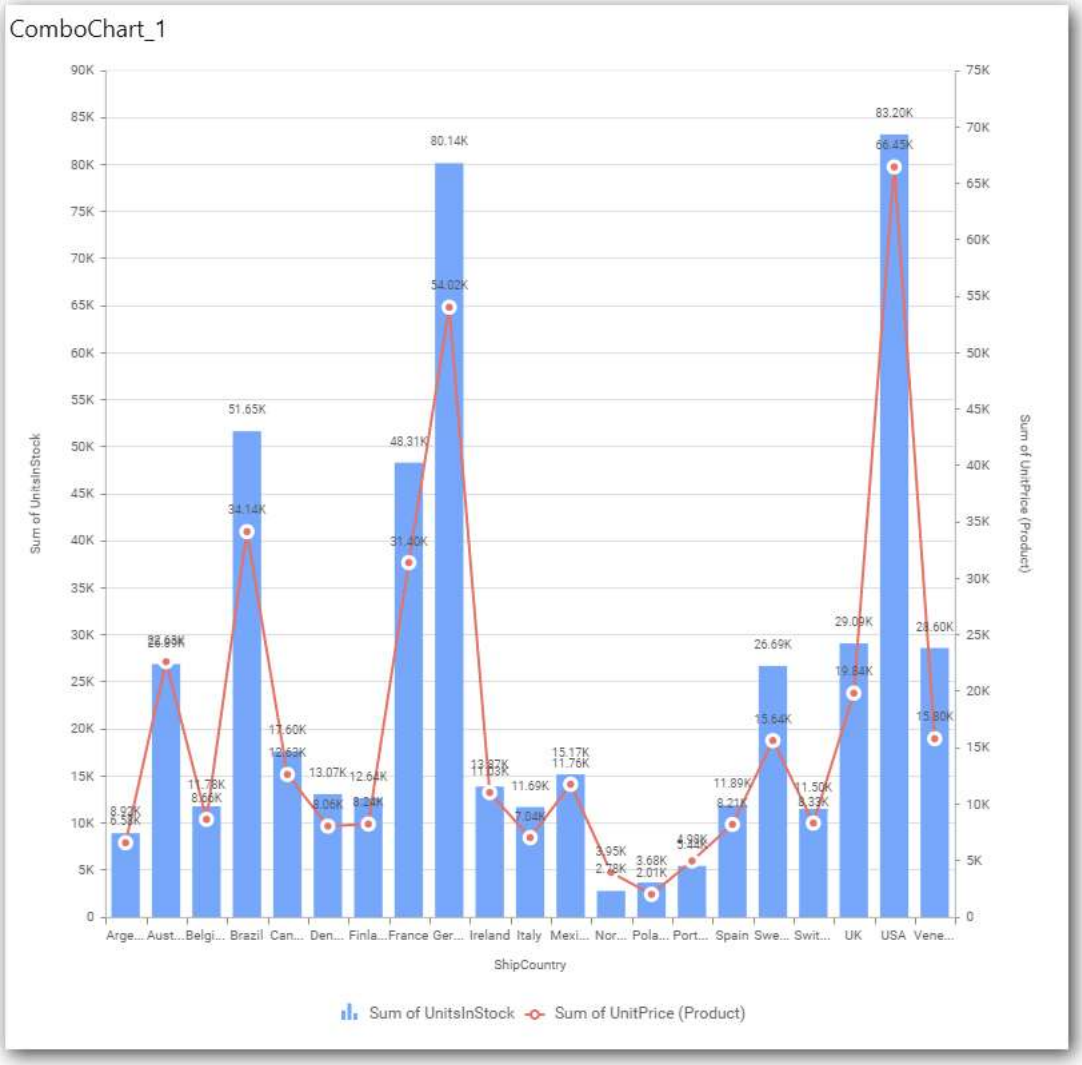
This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

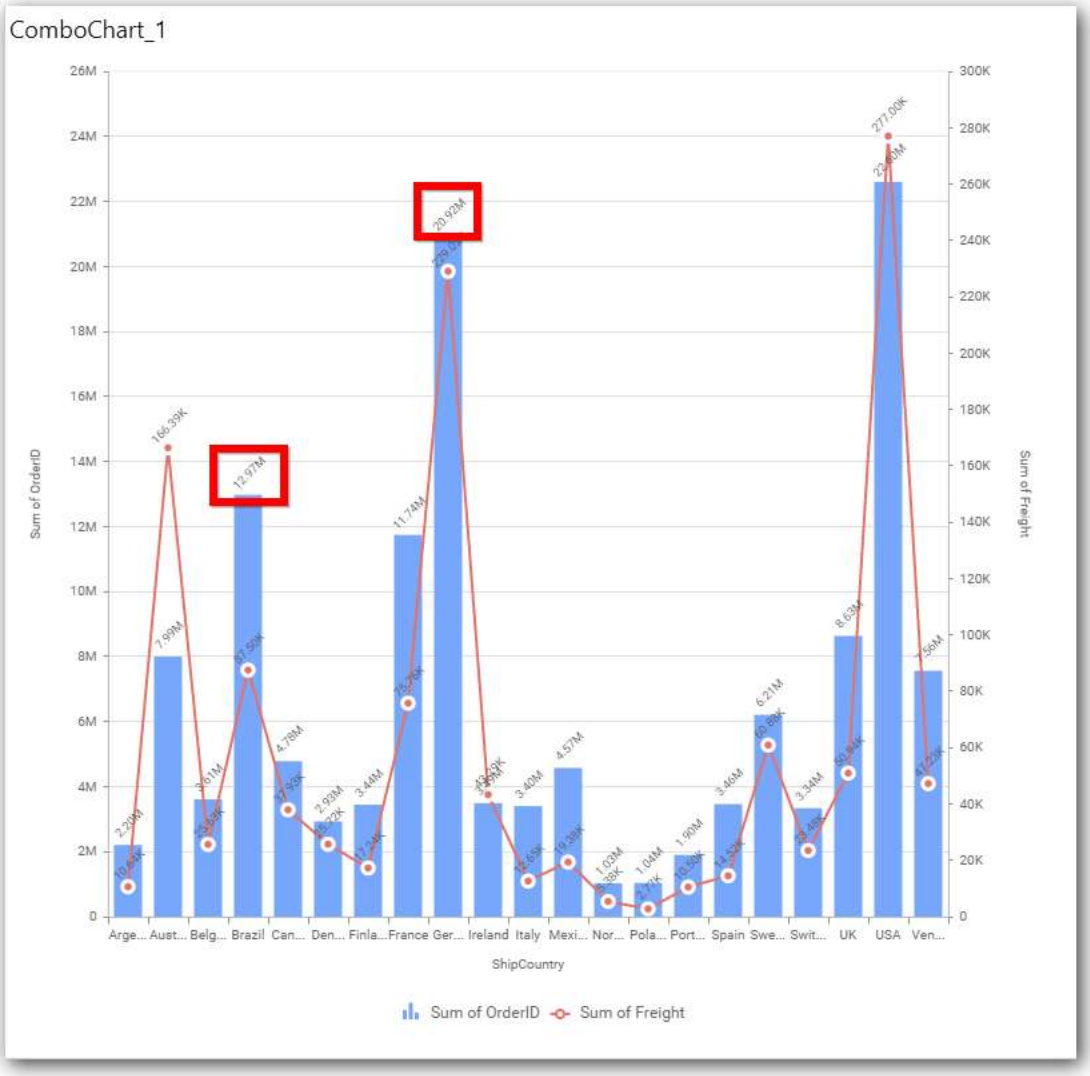
Show Value Labels

This allows you to toggle the visibility of value labels.



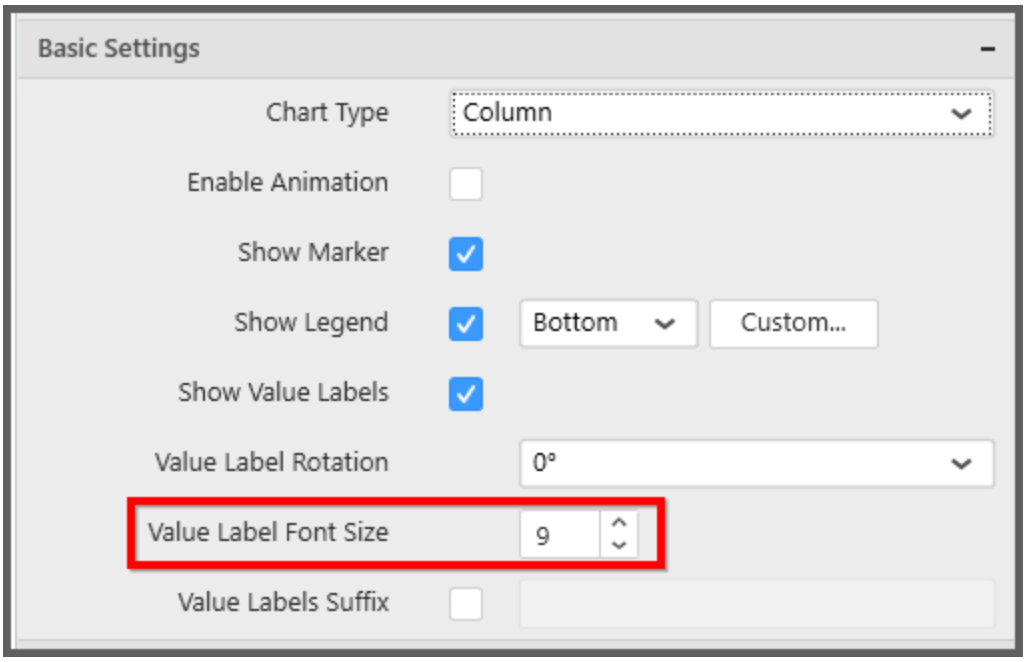
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



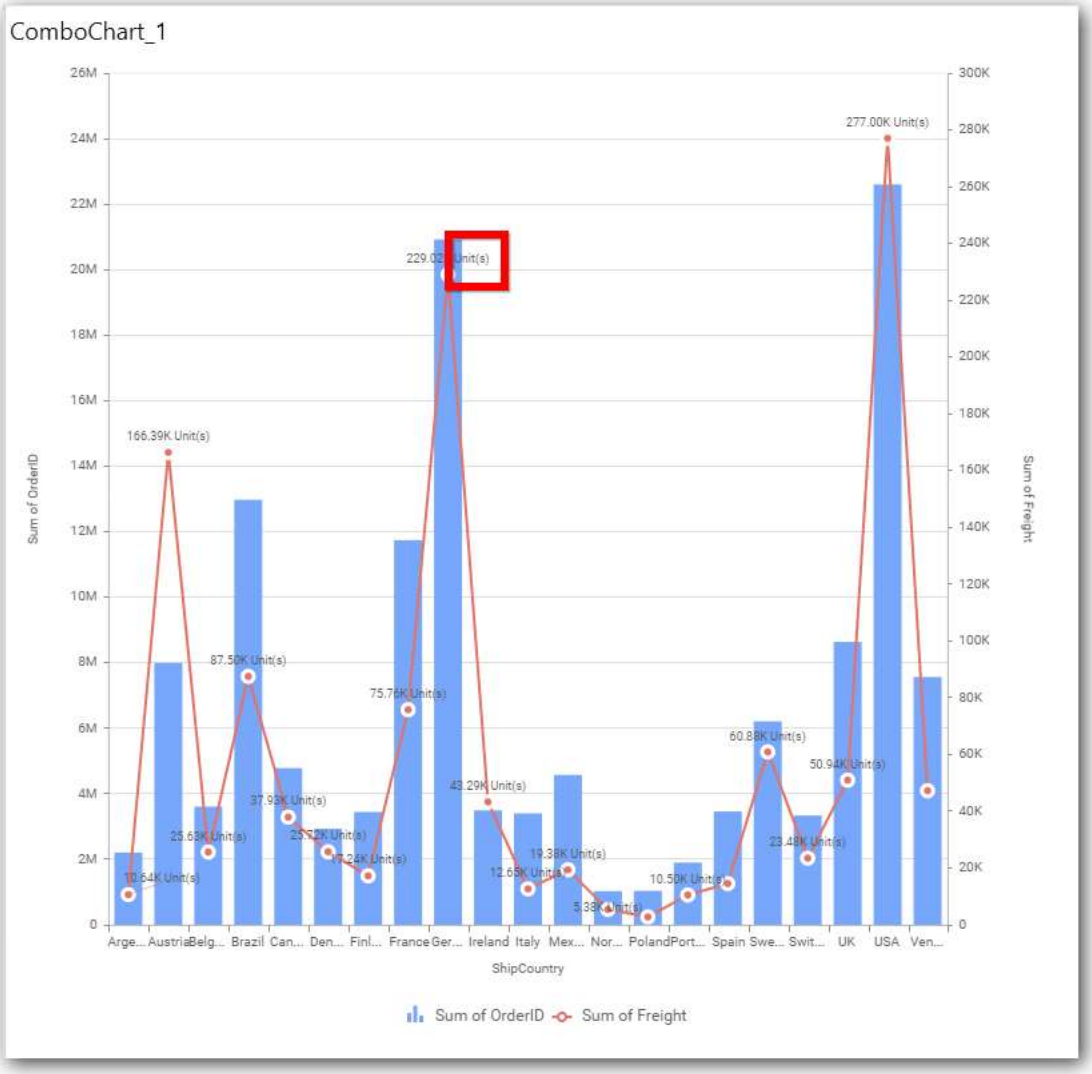
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

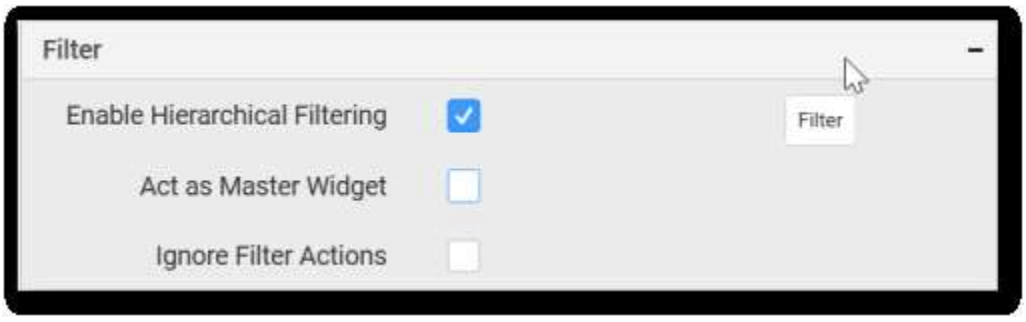


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

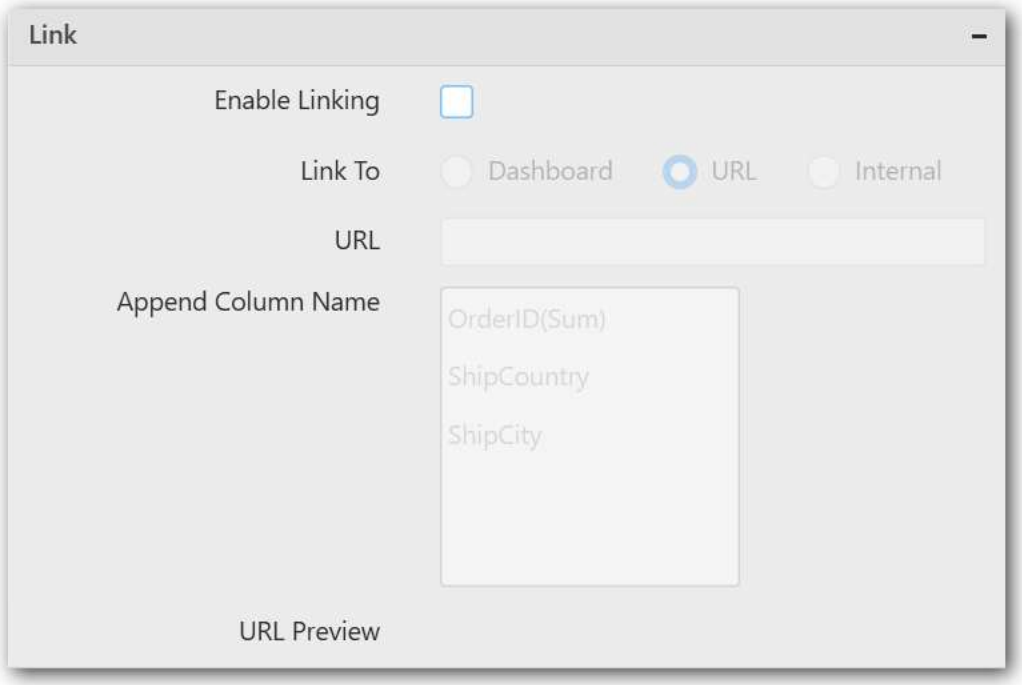
Act as Master Widget

This allows you to define this combo chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

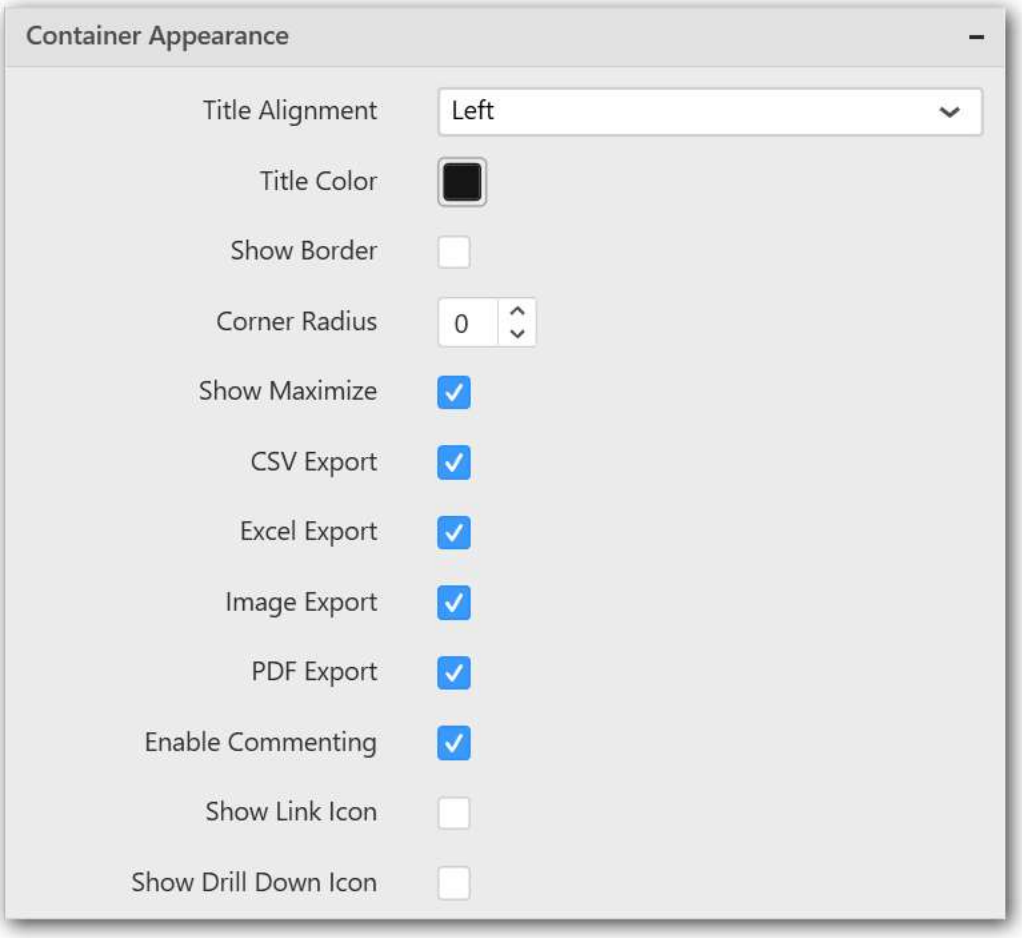
This allows you to define this combo chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this combo chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this combo chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this combo chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

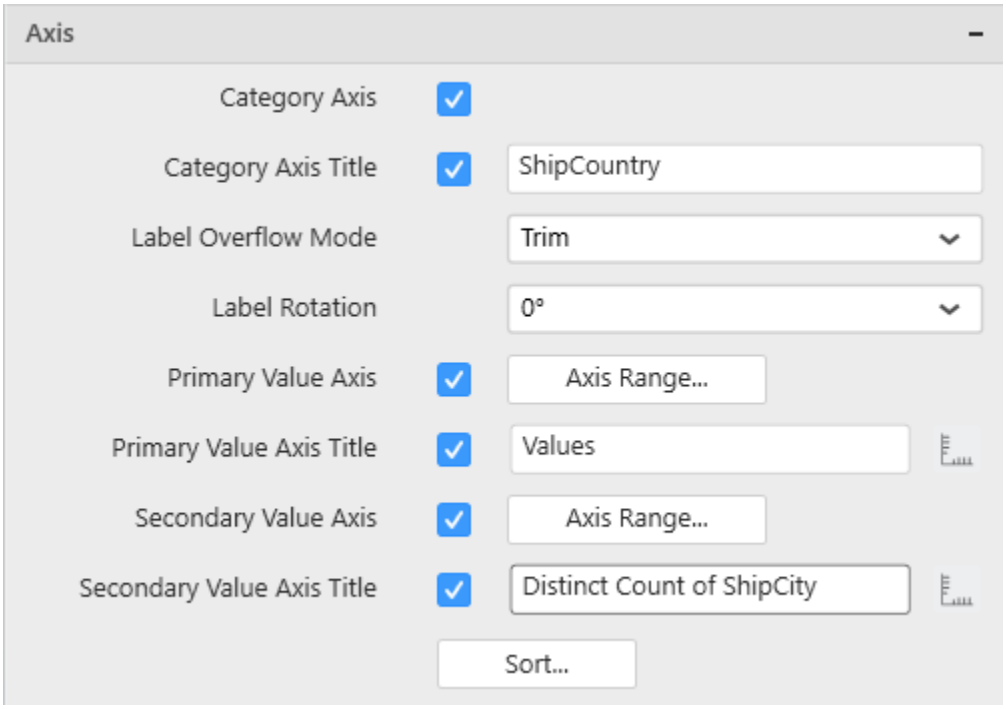
Image Export

This allows you to enable/disable the image export option for this combo chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

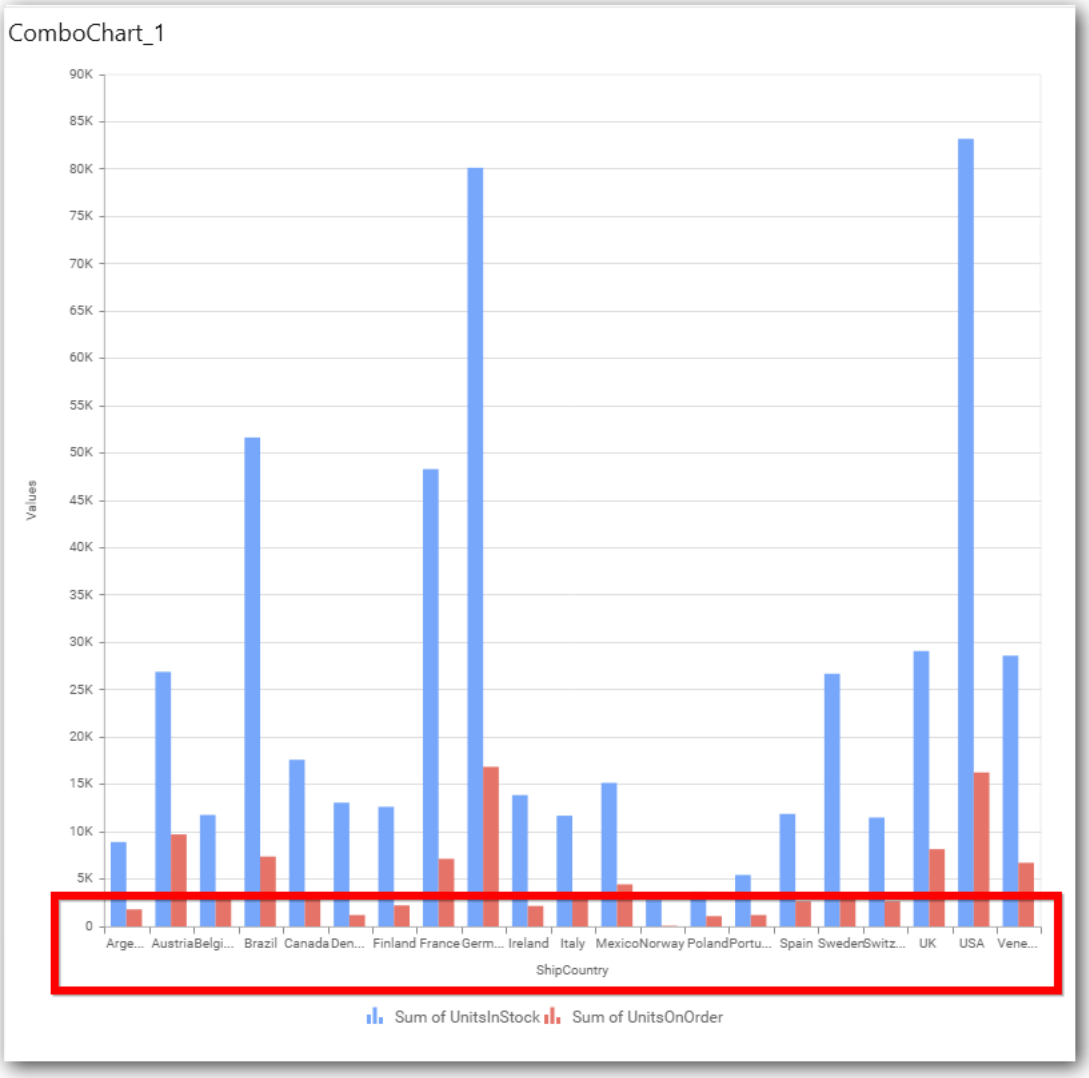
Axis Settings



This section allows you to customize the axis settings in chart.

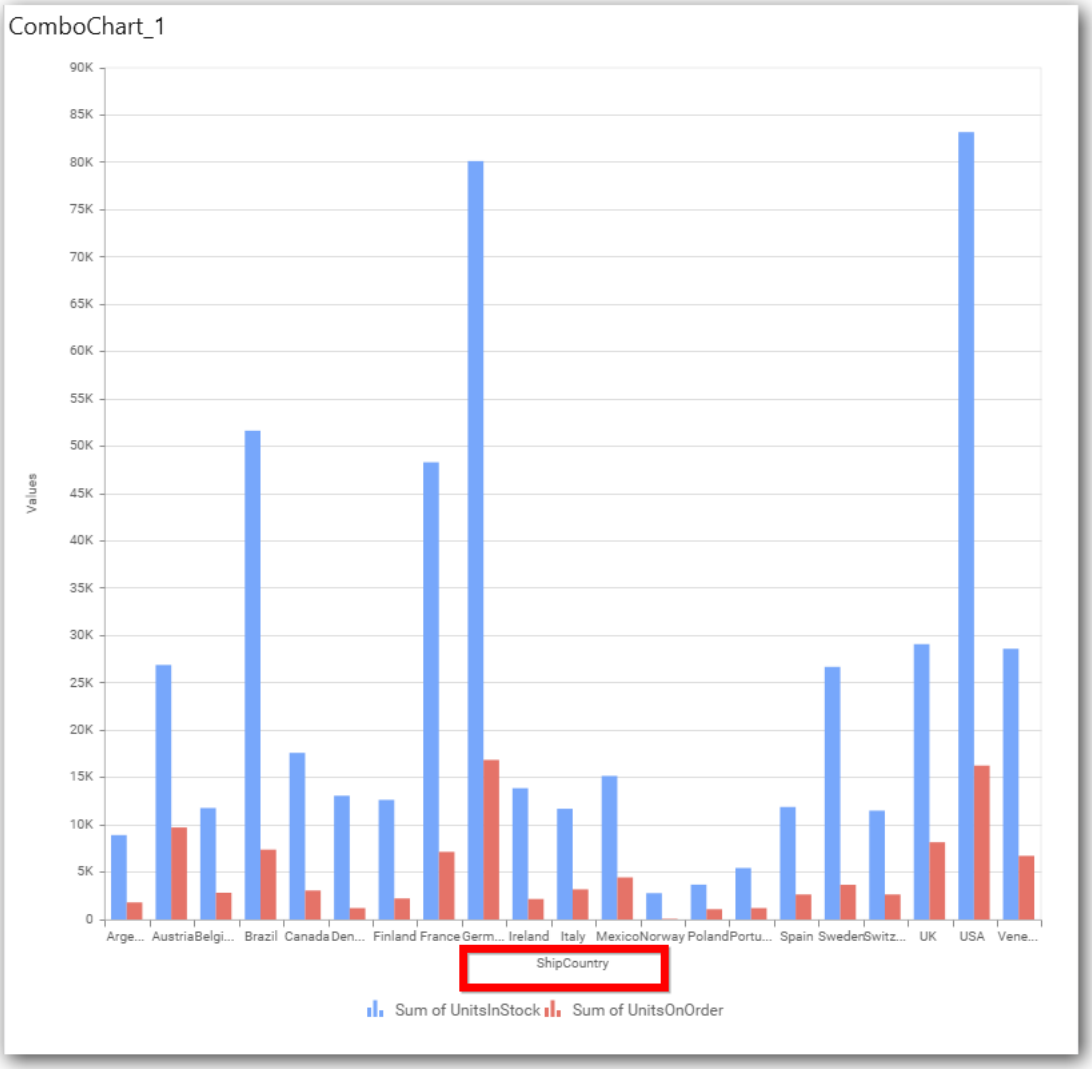
Category Axis

This allows you to enable/edit the option of Category Axis. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

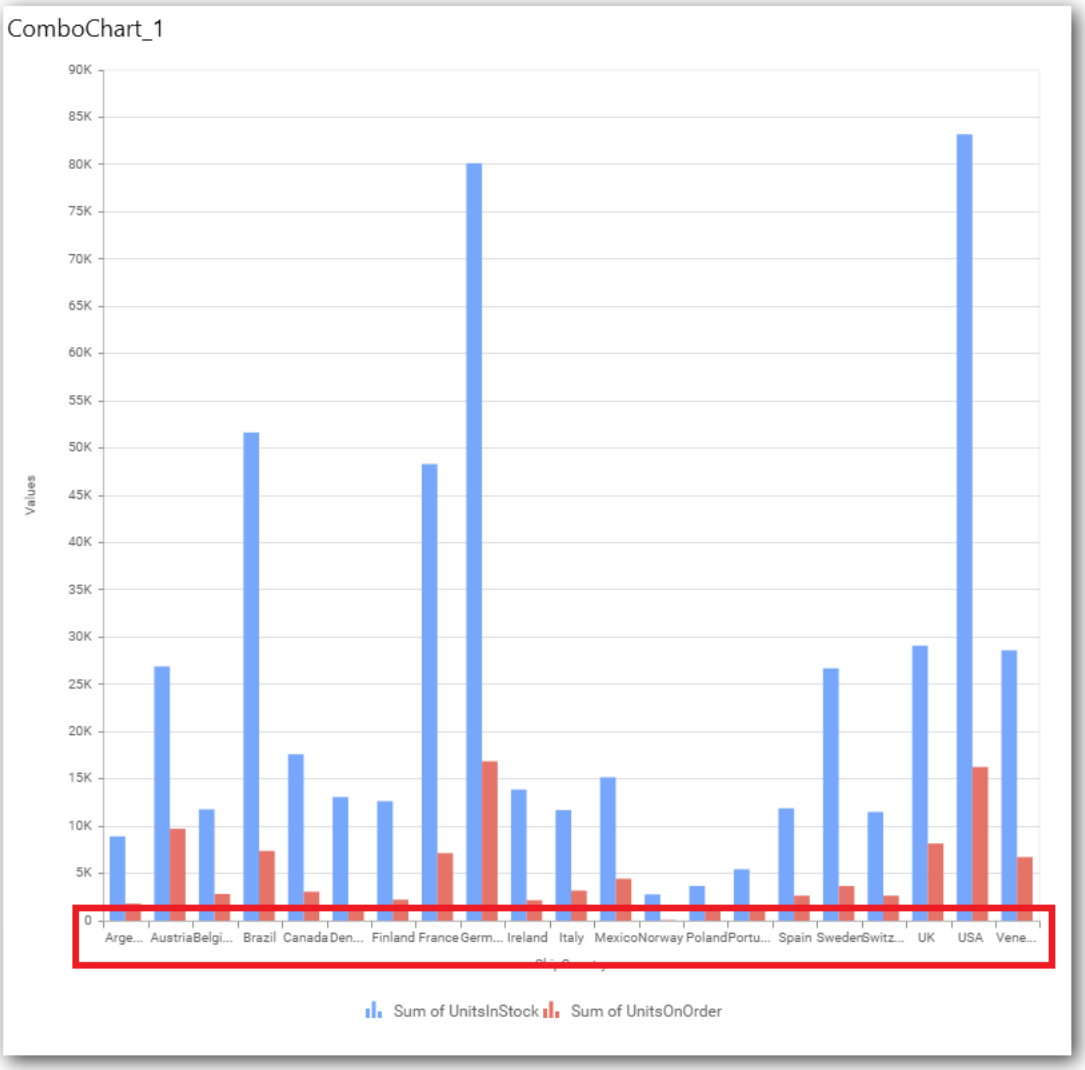


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

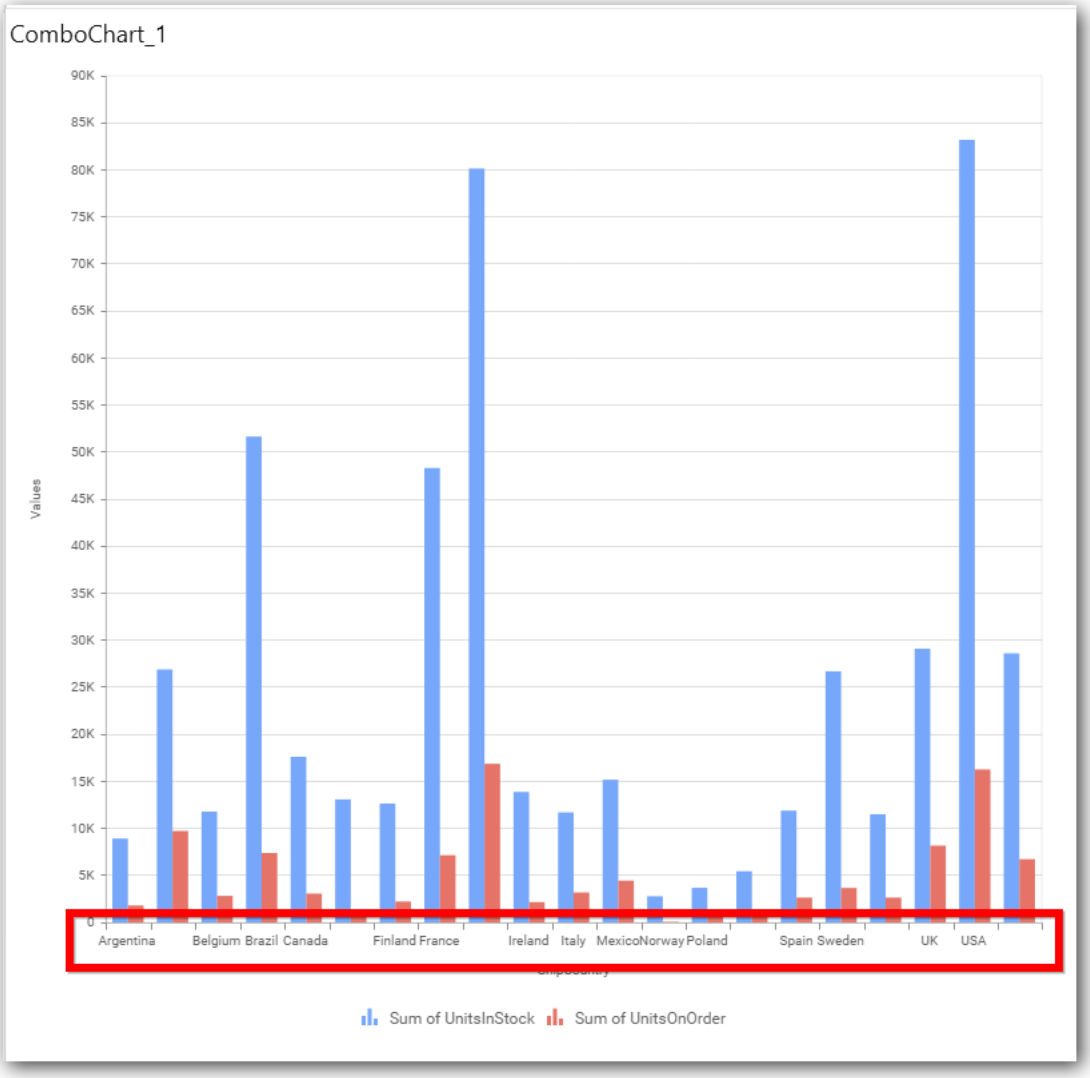
Trim

This option trims the end of overlapping label in the axis.



Hide

This option hides the overlapping label in the axis.

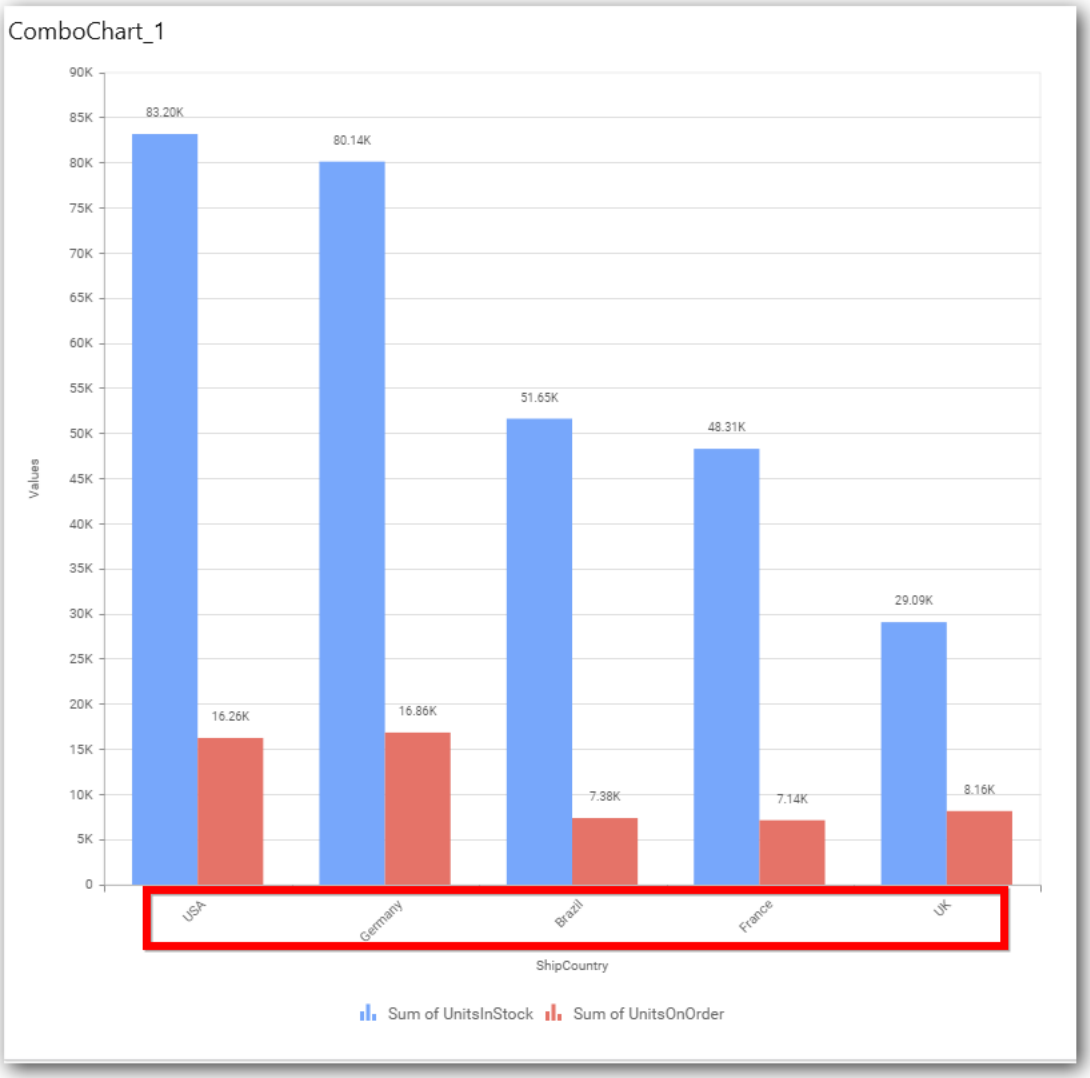


Wrap

This option wraps the lengthy label text in the axis.

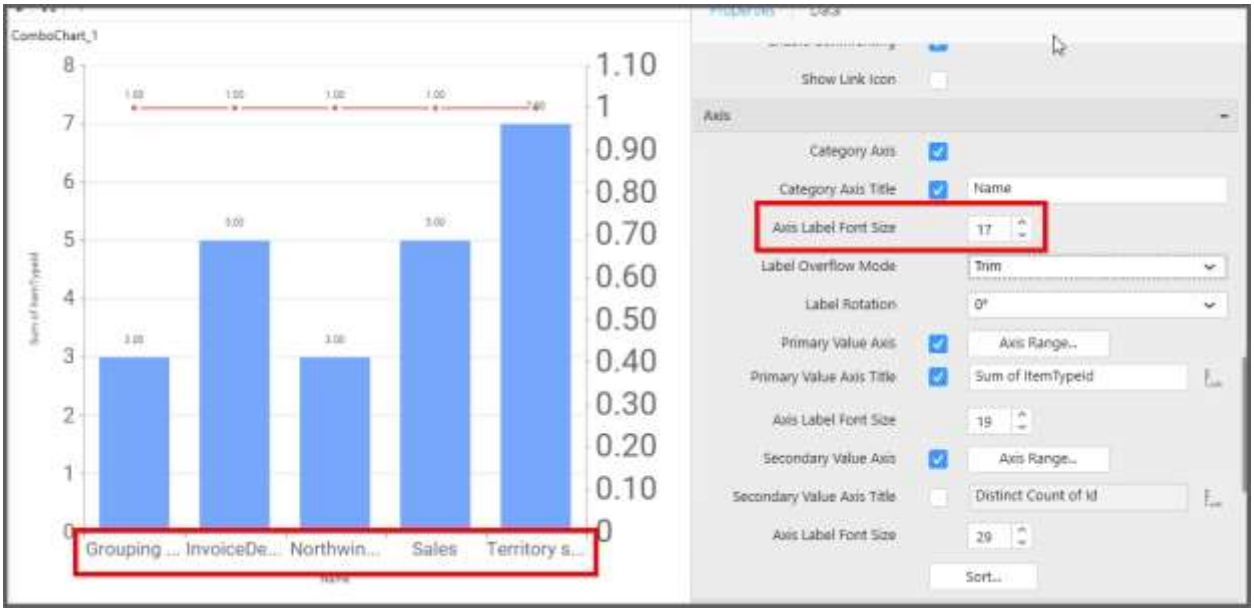
Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



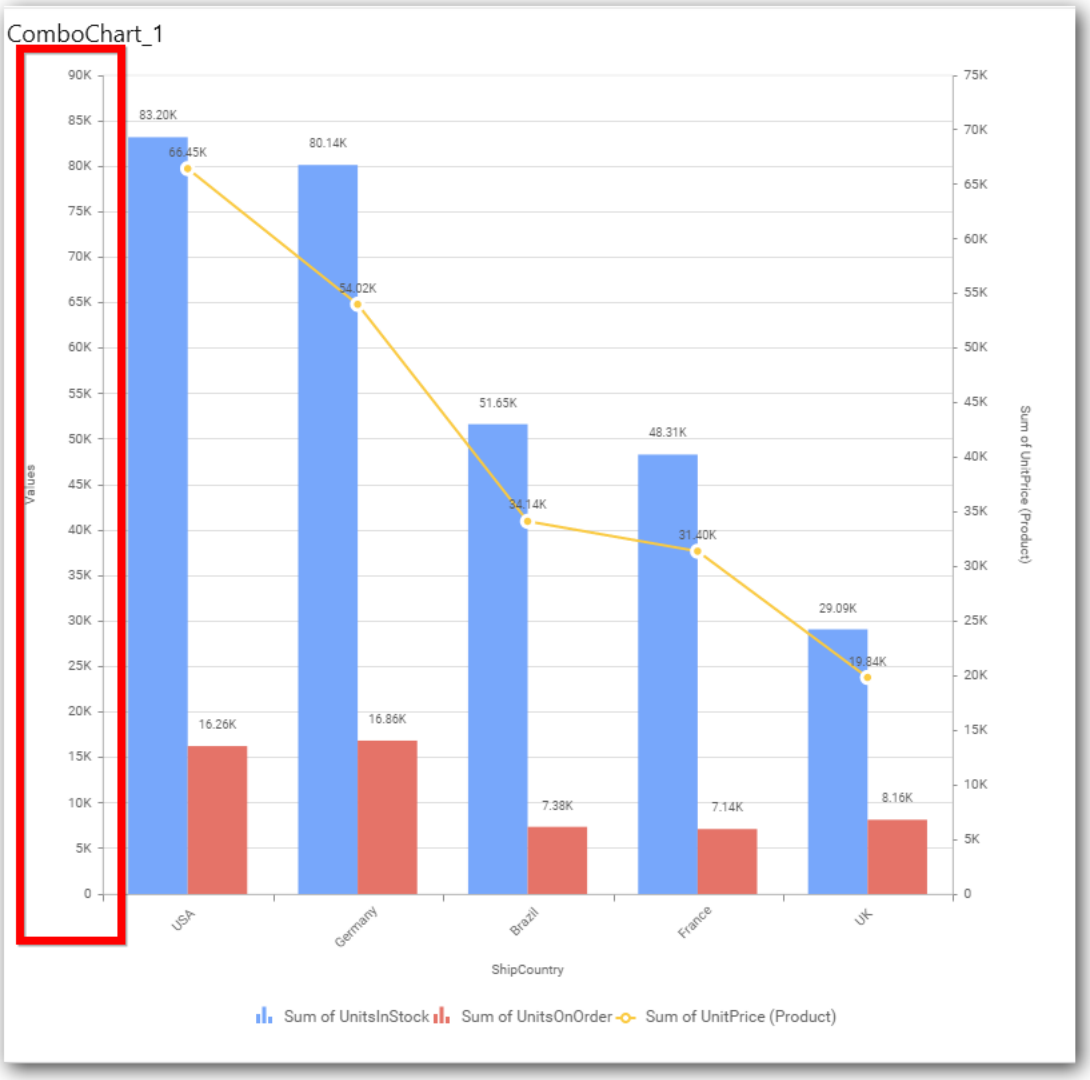
Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.



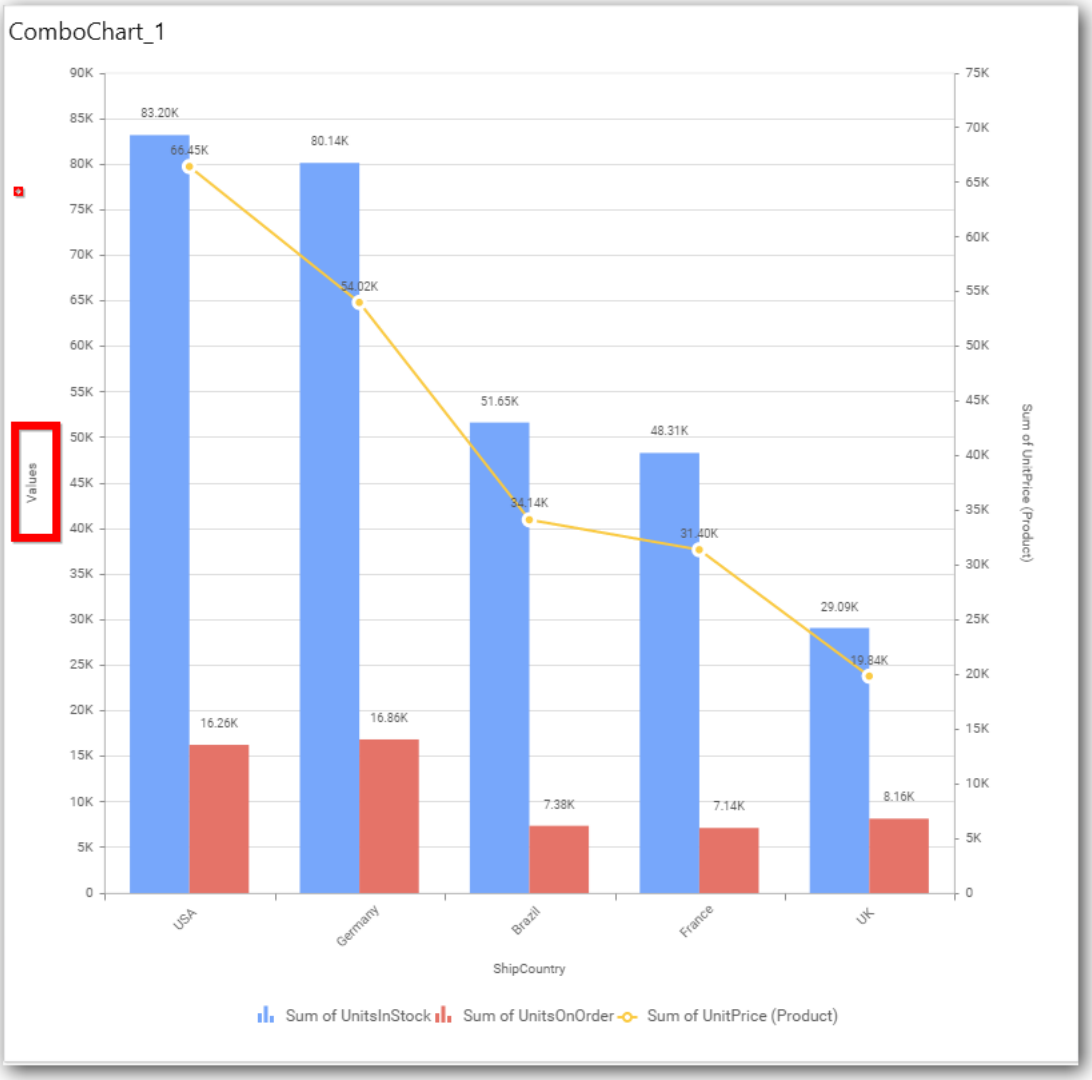
Primary Value Axis

This allows you to enable/edit the option of Primary Value Axis title. It will reflect in chart area y-axis name.



Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

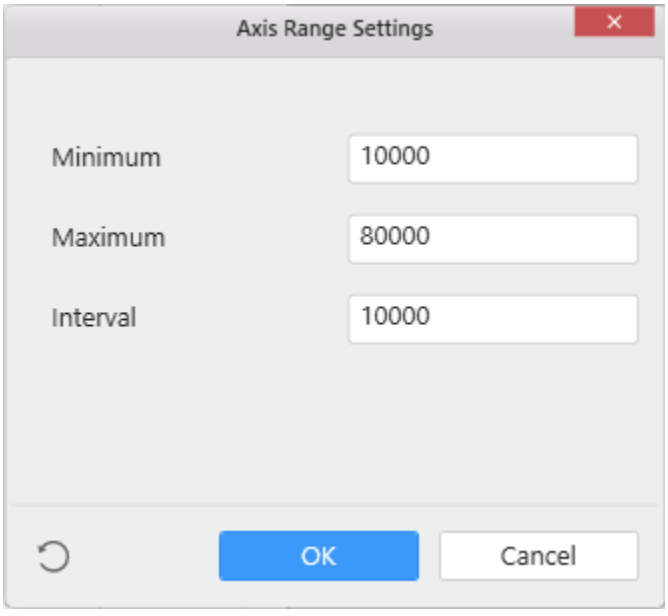
This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

Primary Value Axis Range

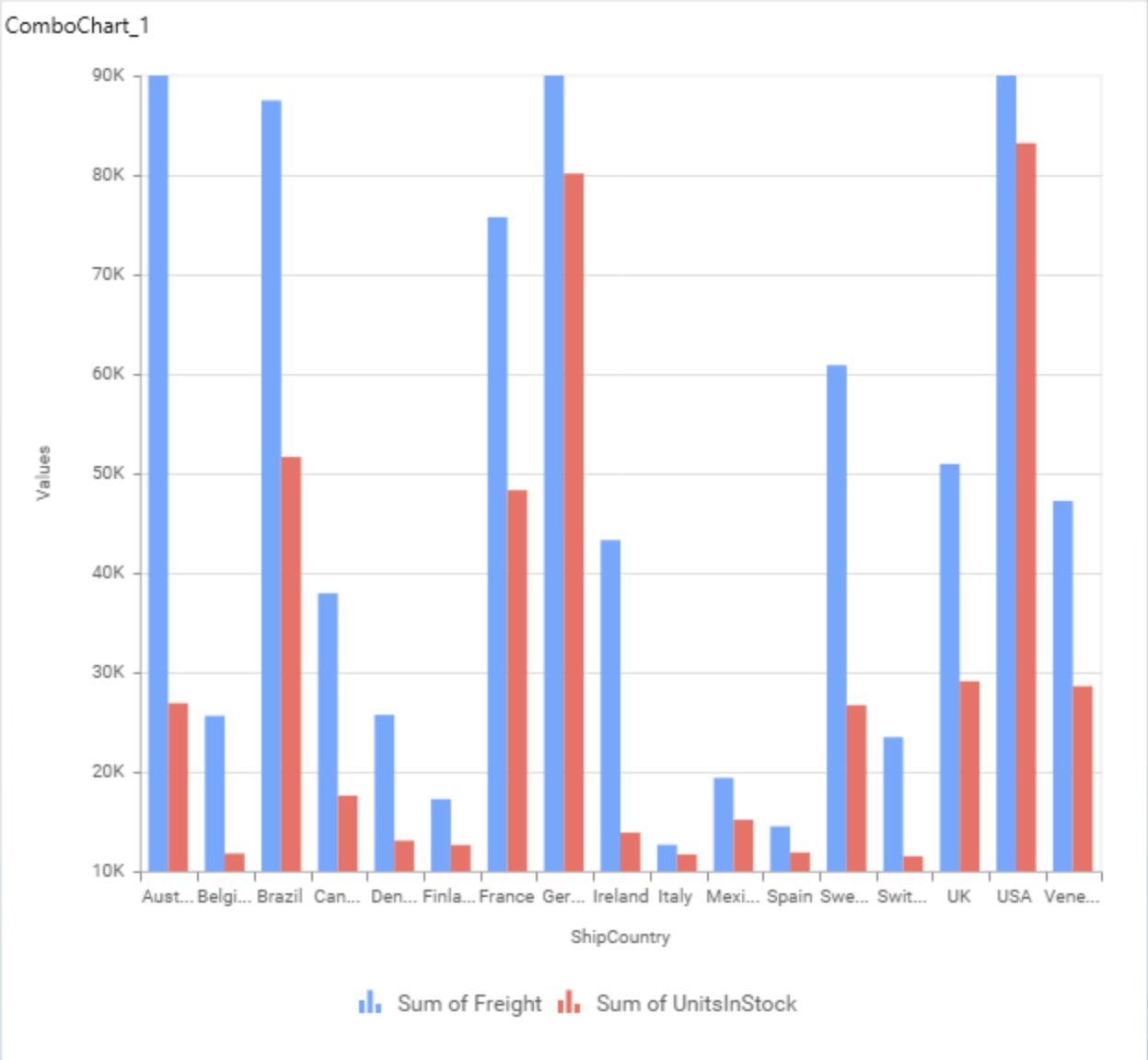
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.

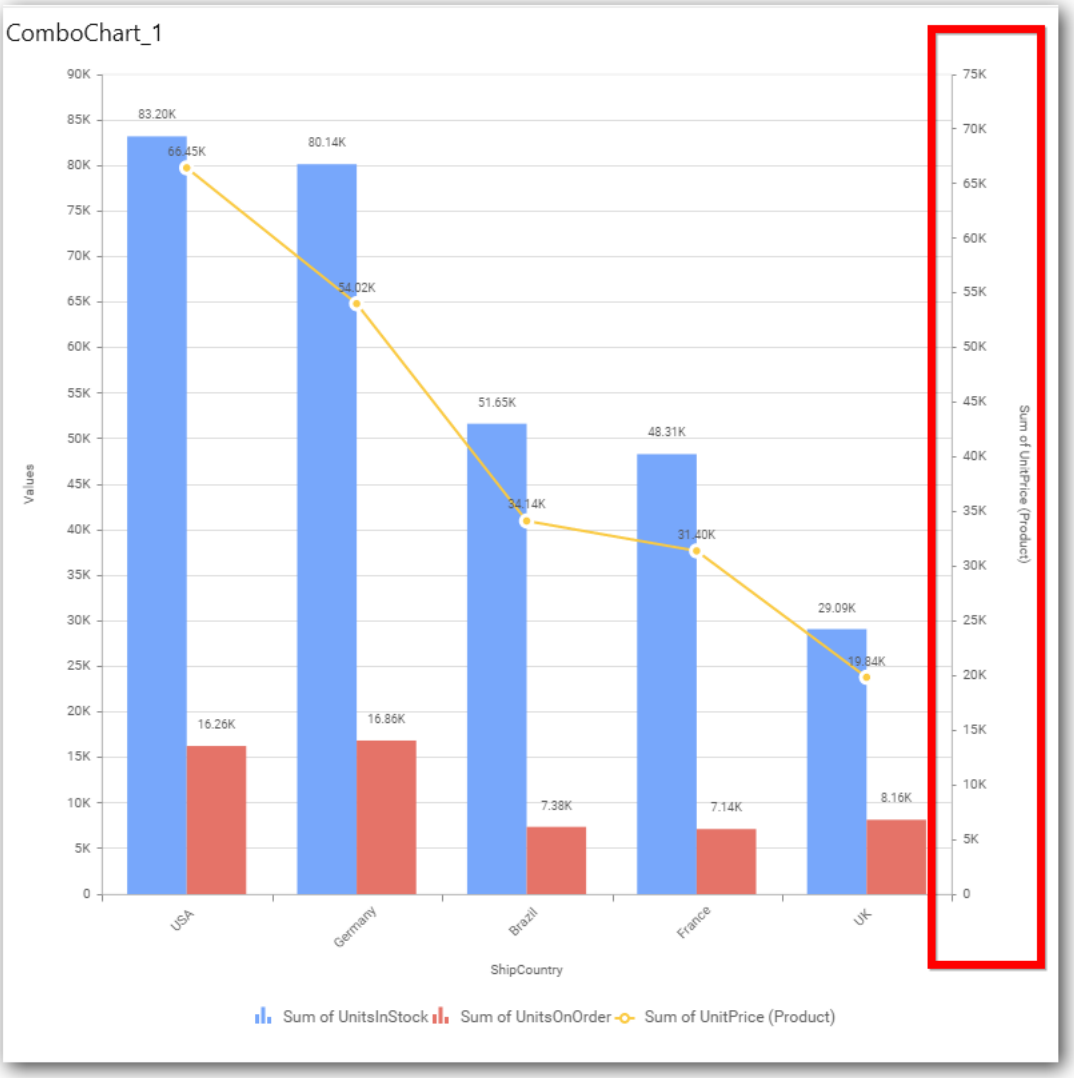


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



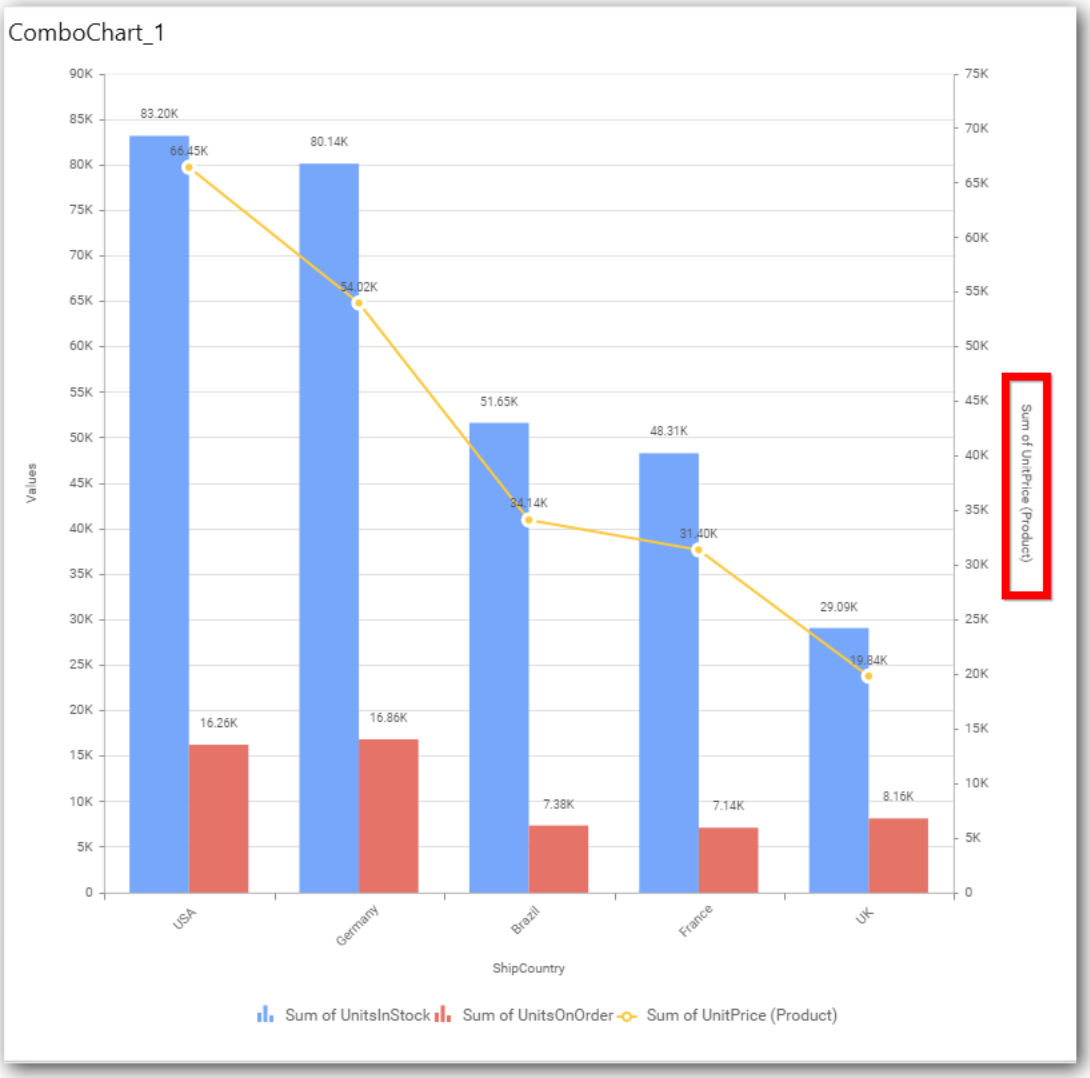
Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.



Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

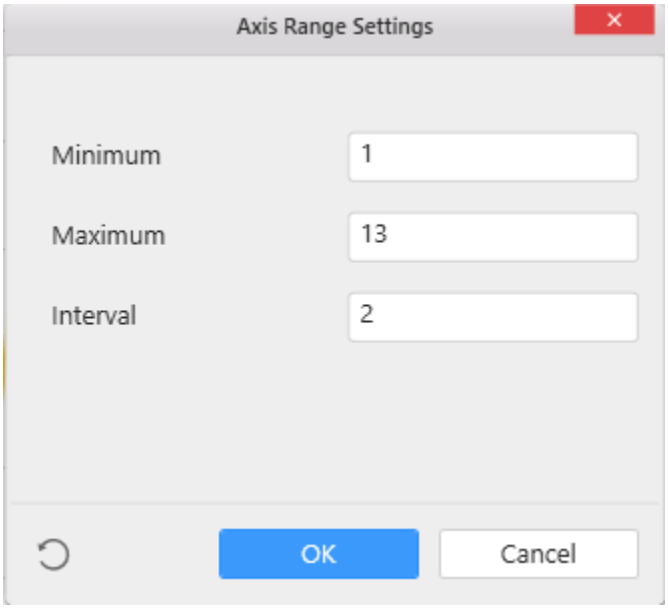


Axis Label Size

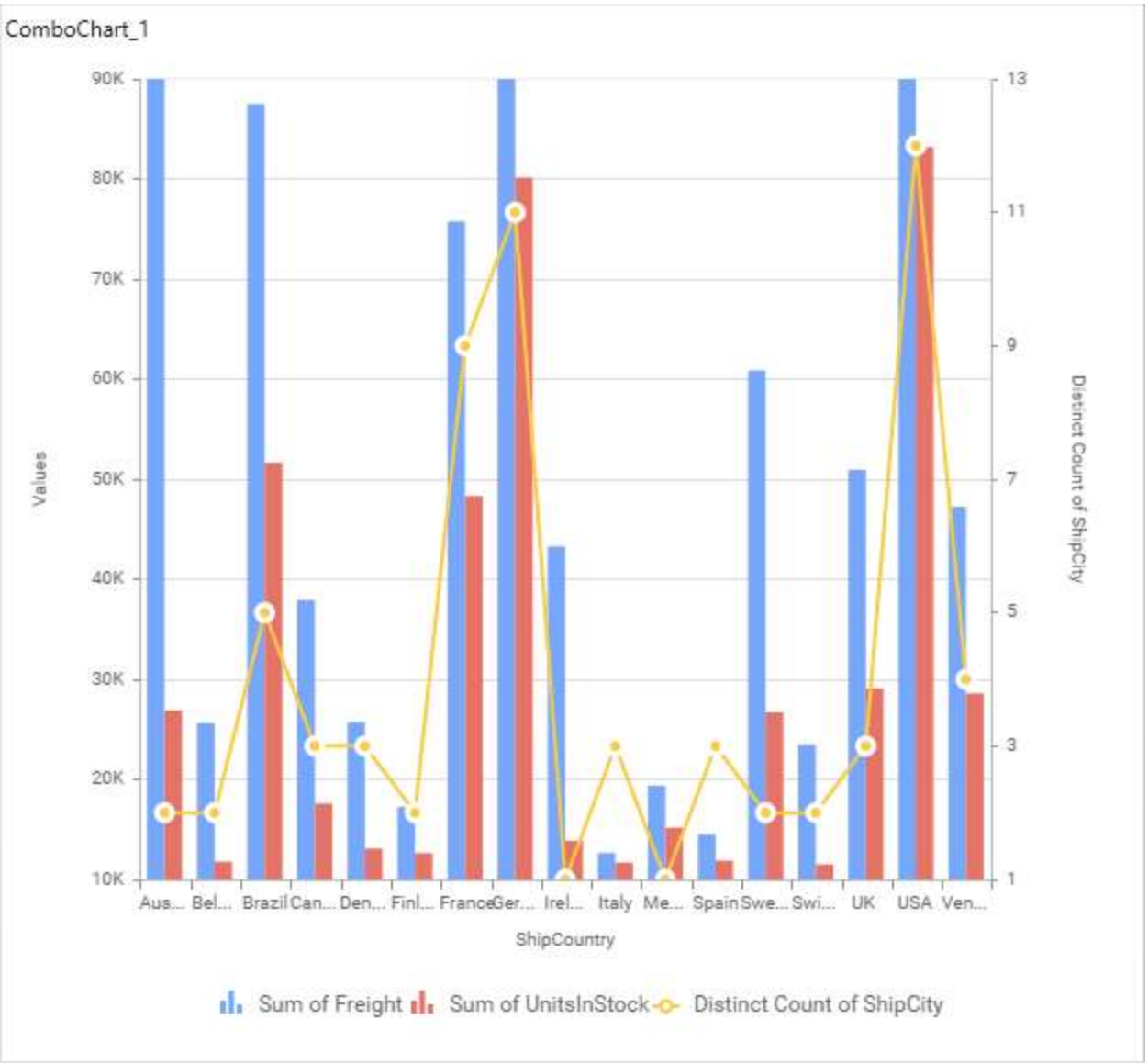
This allows you to increase or decrease the font size of the secondary axis label. Default font size for the secondary axis label was 10 pixels.

Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

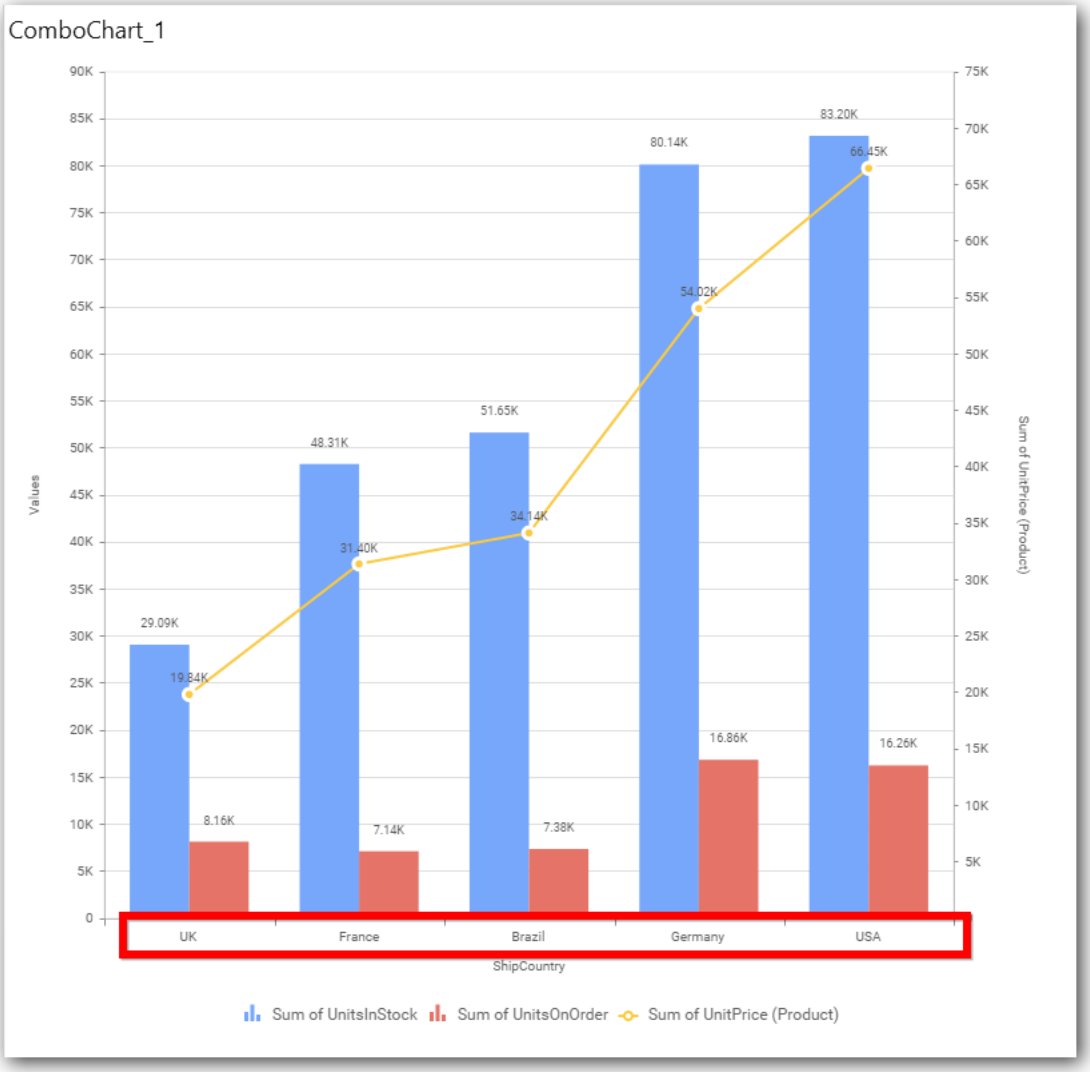


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.

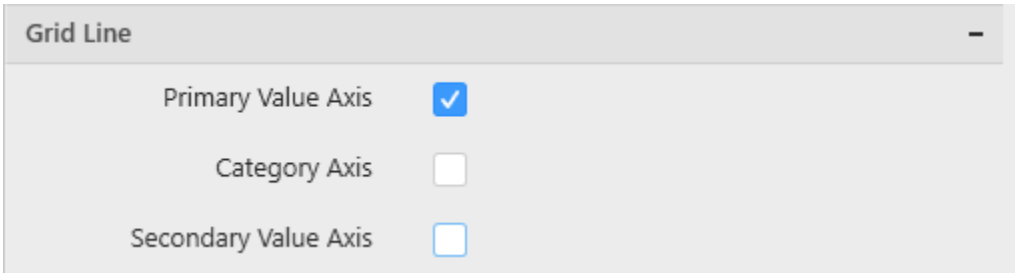


Sort Order

This allows you to define the sort order for each measure column added.



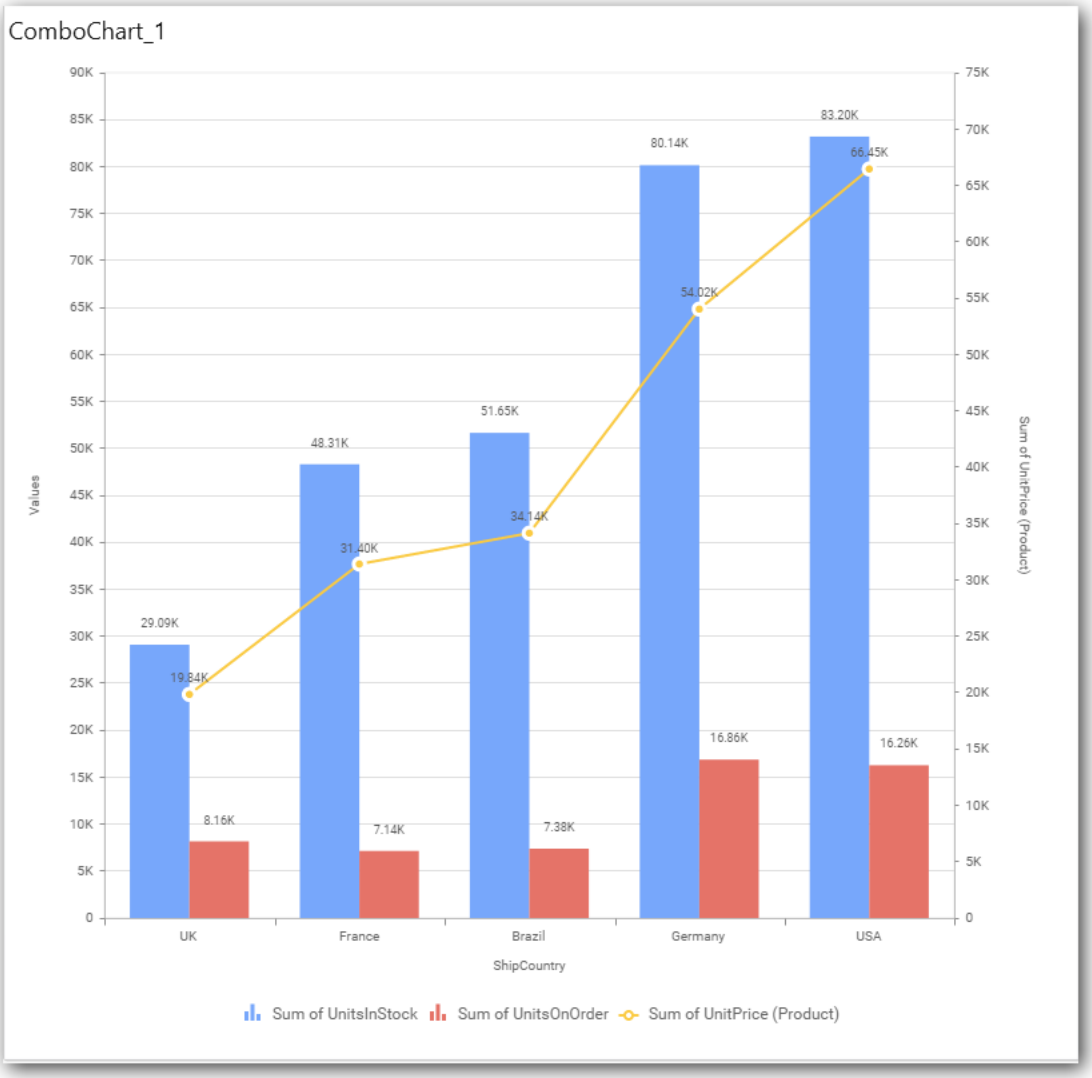
Grid Line Settings



Primary Value Axis

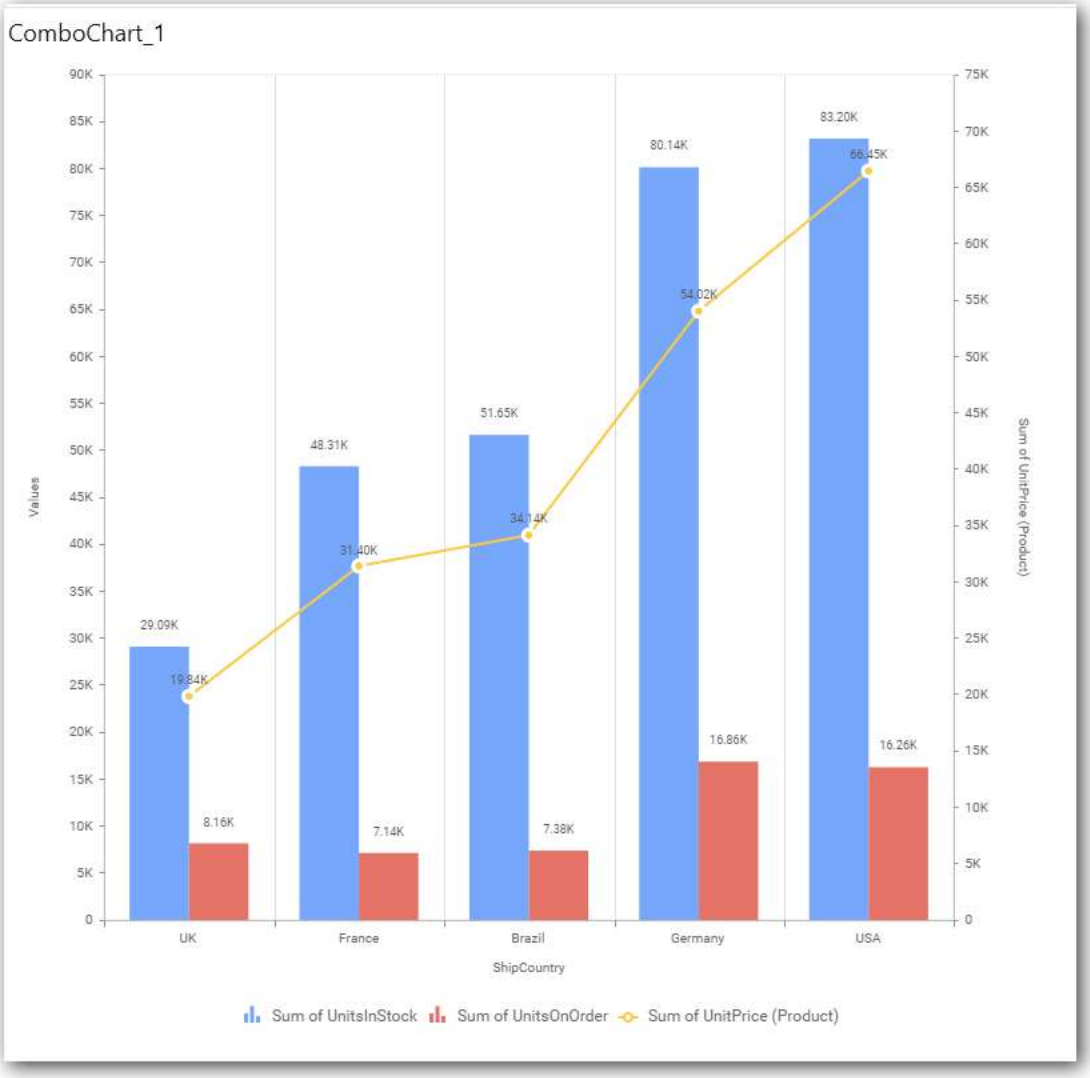
This allow you to enable the Primary Value Axis gridlines for the combo chart.

Primary Value Axis Line



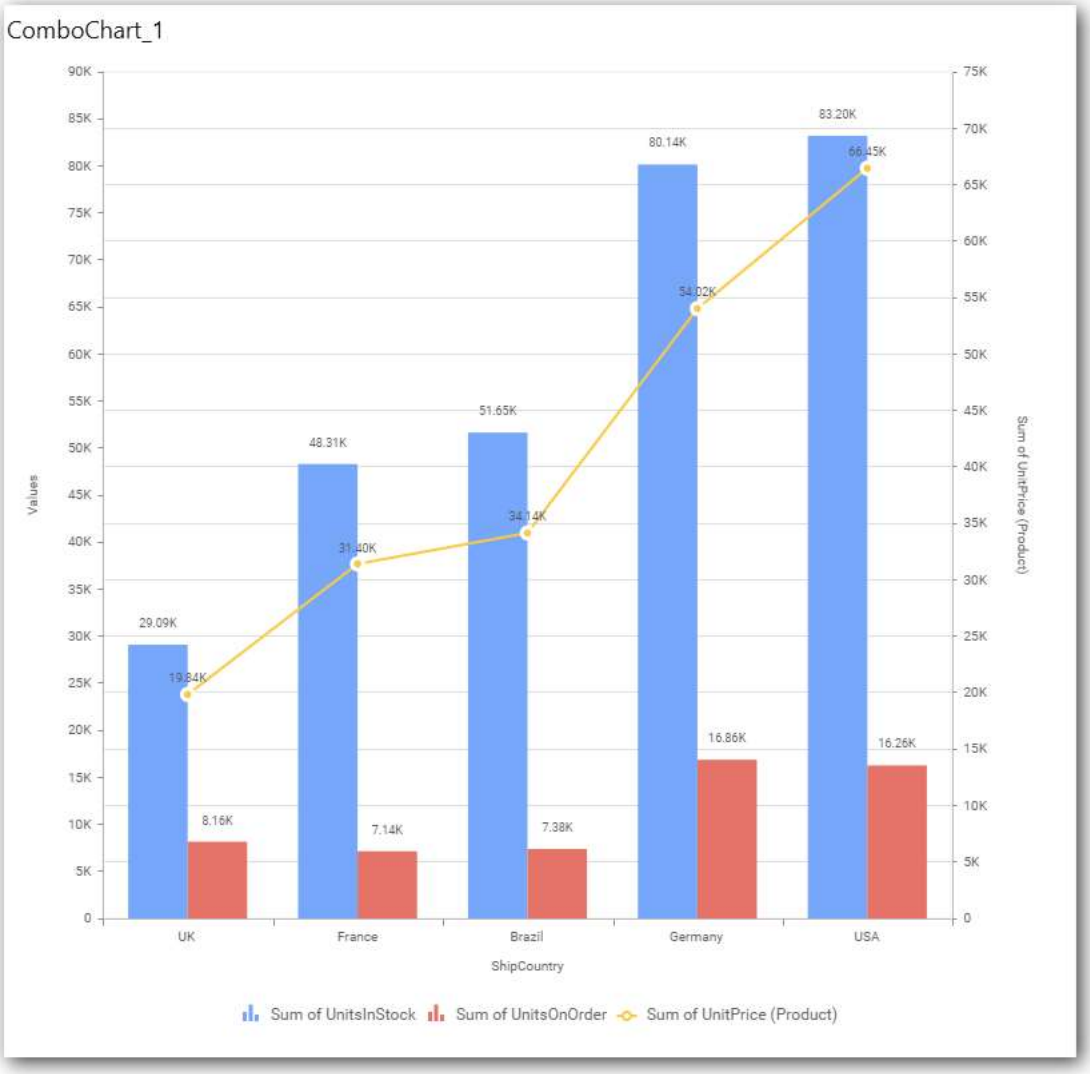
Category Axis

This allows you to toggle the visibility of Category Axis gridlines.



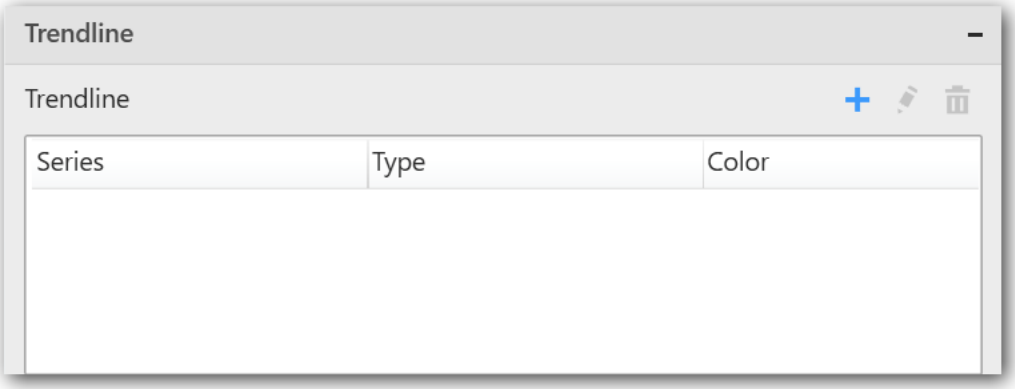
Secondary Value Axis

This allow you to toggle the visibility of Secondary Value Axis gridlines.

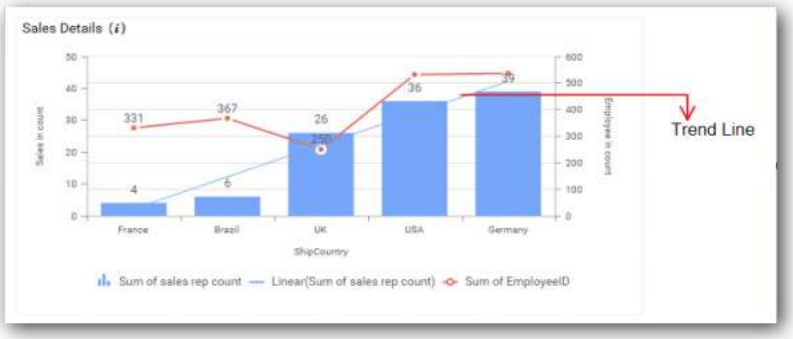


Trend line Settings

You can add trend line to chart based on dropped measure that you select. You can also customize its legend text, line type and line color. Trend line is not visible, by default.



After applying these settings, it will reflect in chart like below.



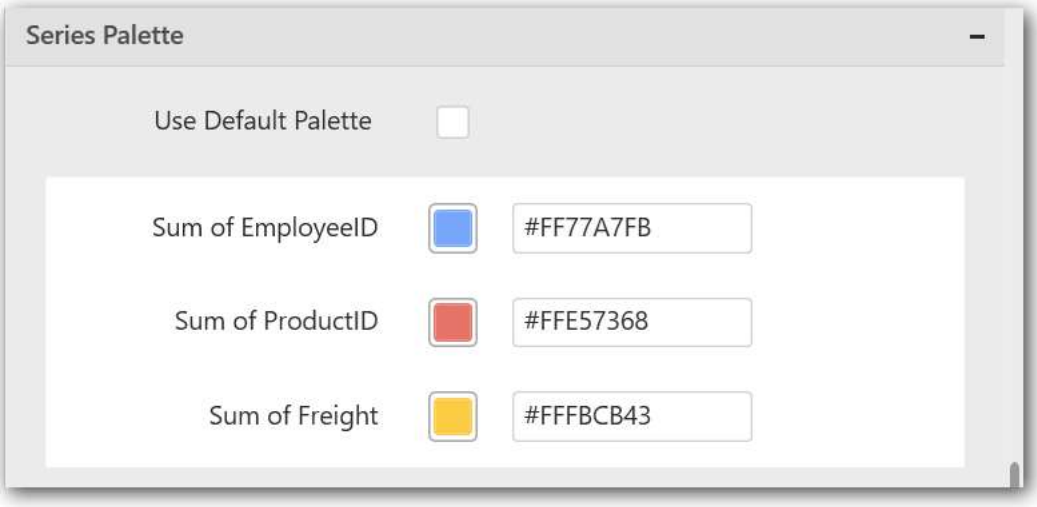
You have options to add or delete an added trend line.

Series Palette

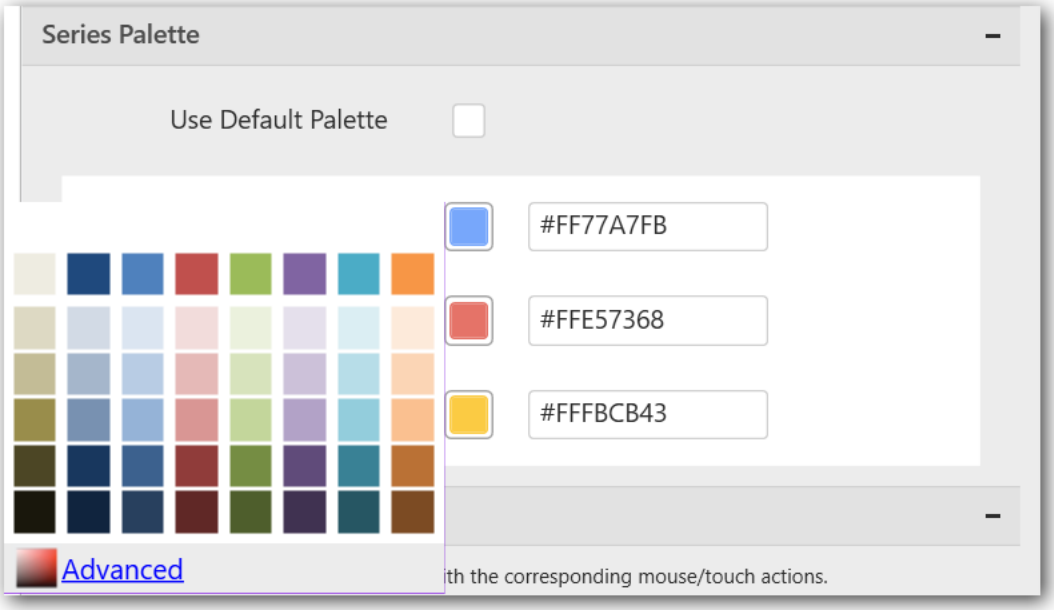
This allows you to customize the chart series color through Series Palette section.

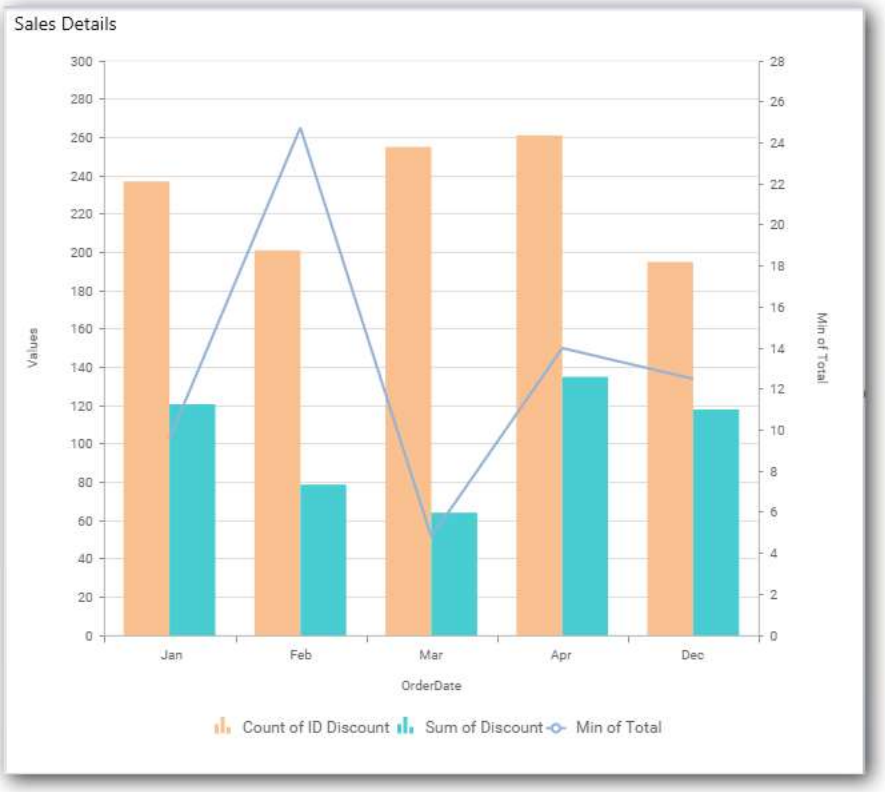
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



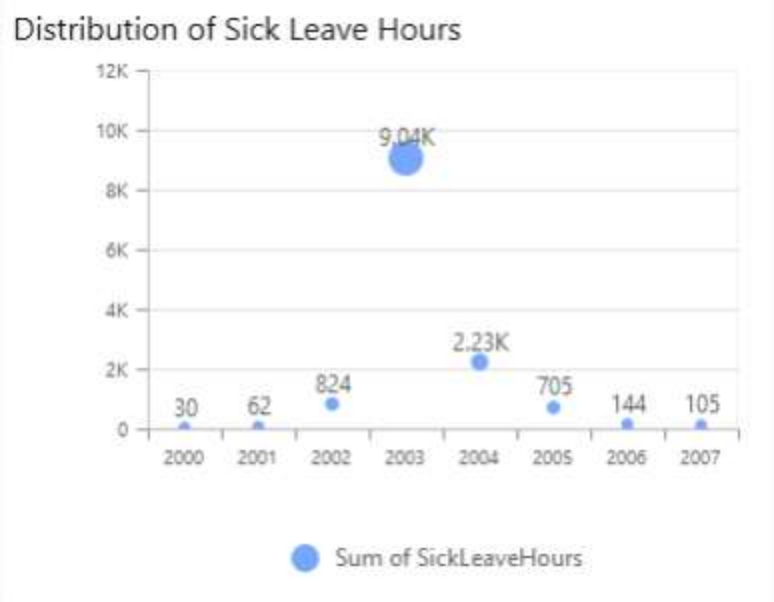
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Bubble Chart

Bubble Chart allows you to compare large number of data points represented as bubbles and showcase the difference through its size.

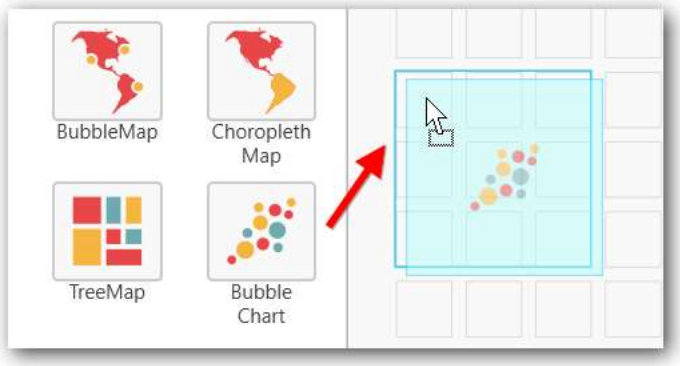


How to configure flat table data to Bubble Chart?

Bubble Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Follow the steps to configure data to bubble chart

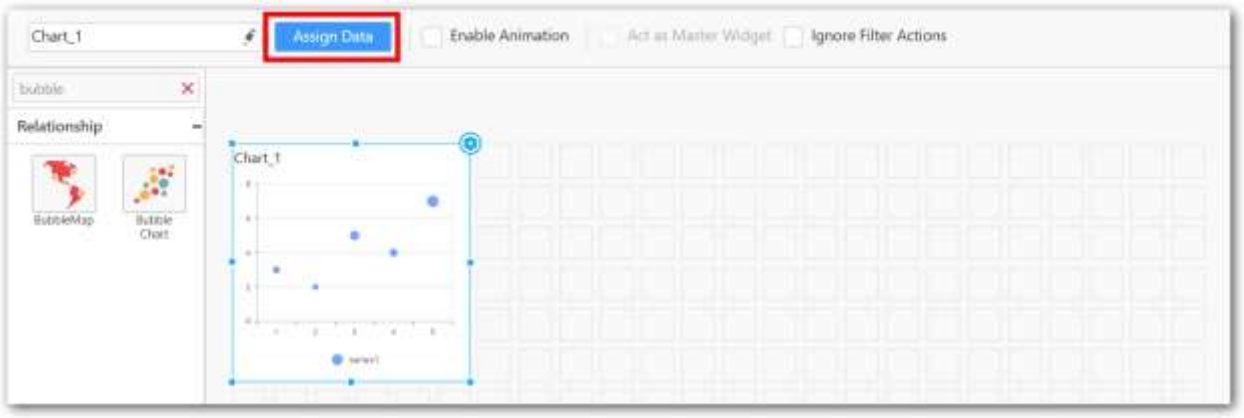
Drag and drop the bubble chart into canvas and resize it to your required size.



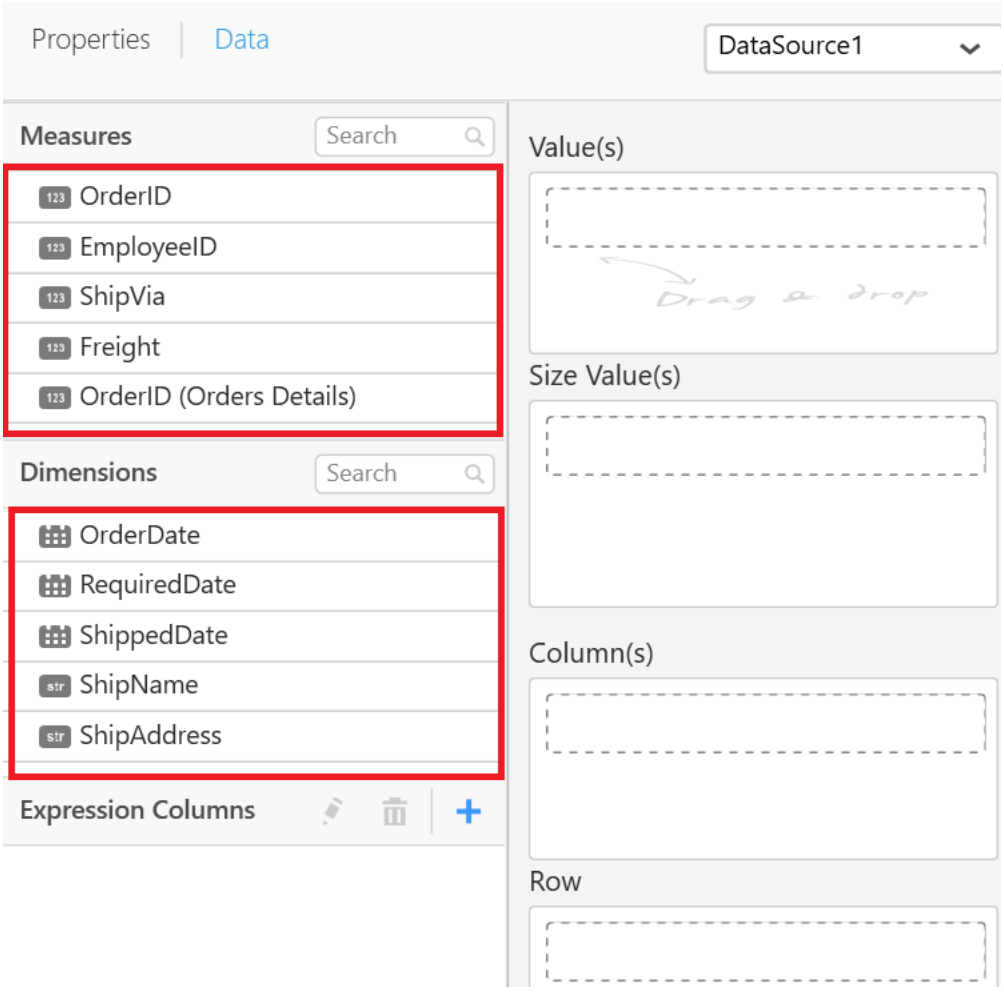
Connect to data source.

Focus on the bubble chart.

Click on Assign Data.

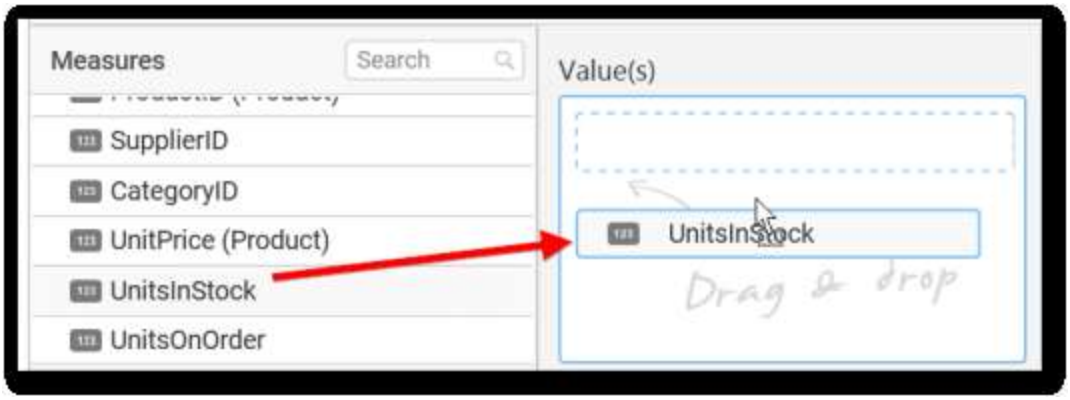


The data pane will be opened with available measures and dimensions from the connected data source.

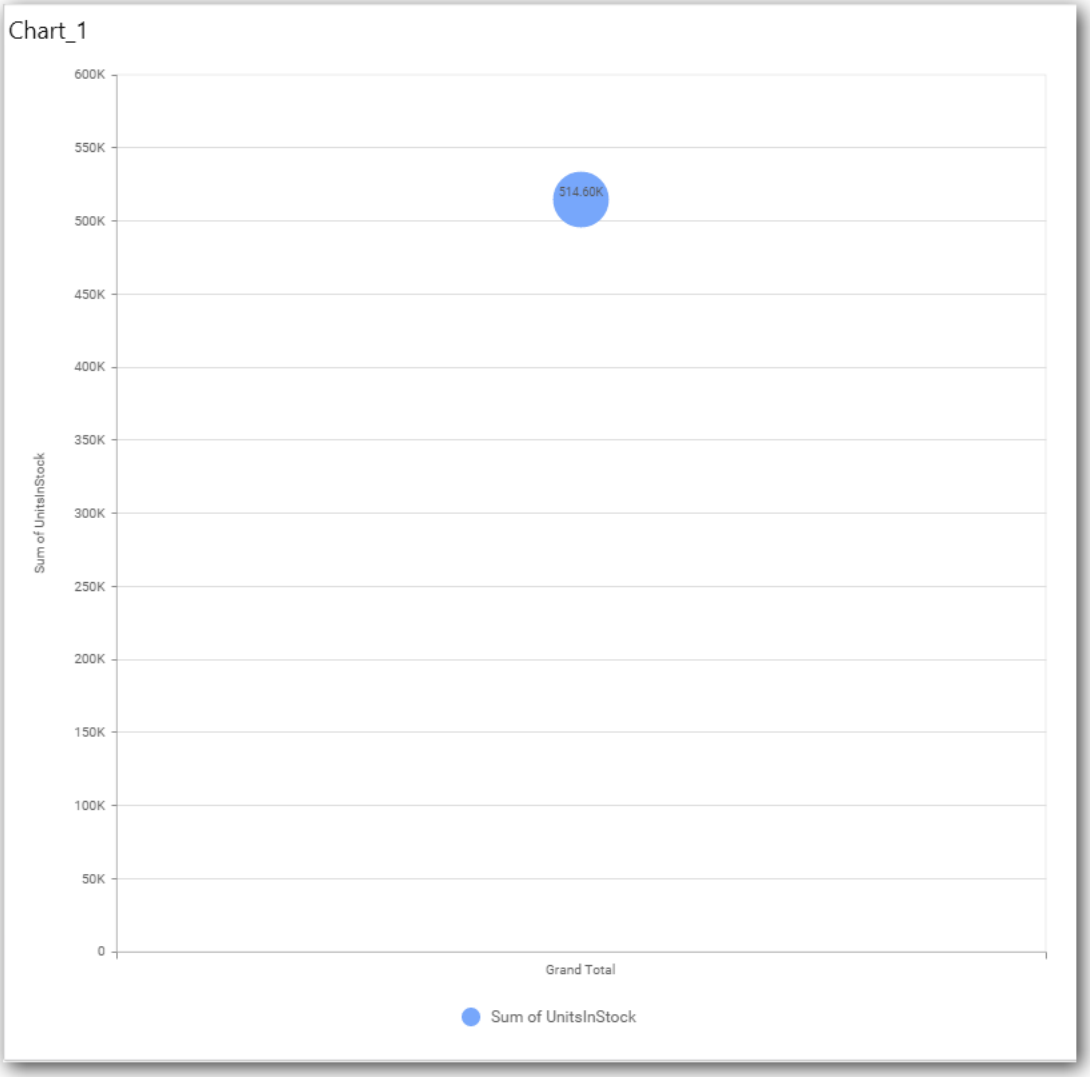


Assigning Value(s)

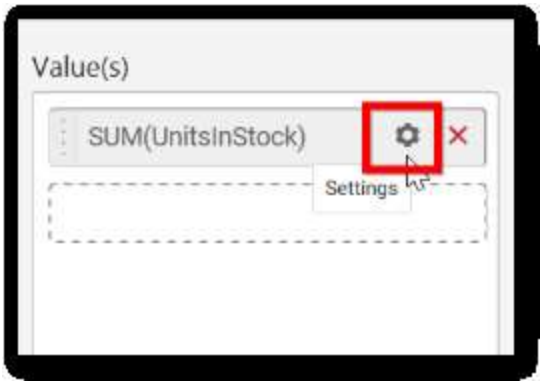
Drag and drop the Measure into Value.



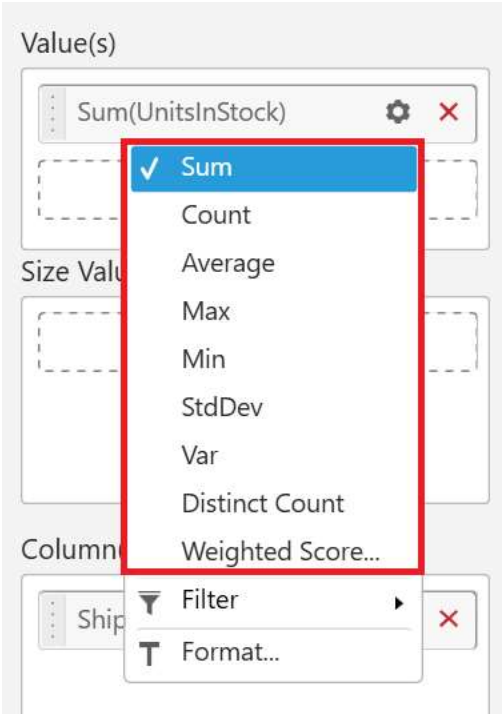
Now the chart will be rendered like this.



You can change the summary type of the value by clicking on Settings option.

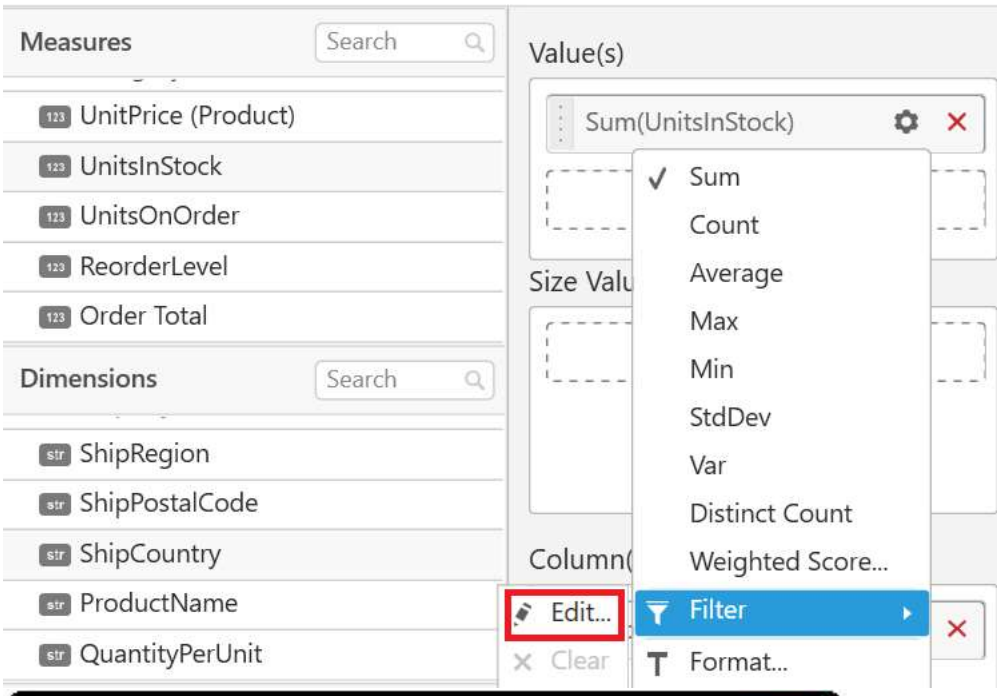


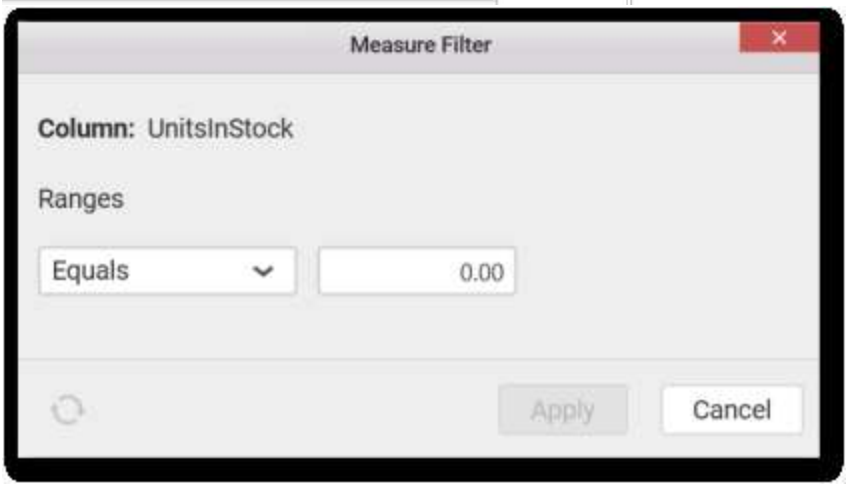
Select the required summary type from list.

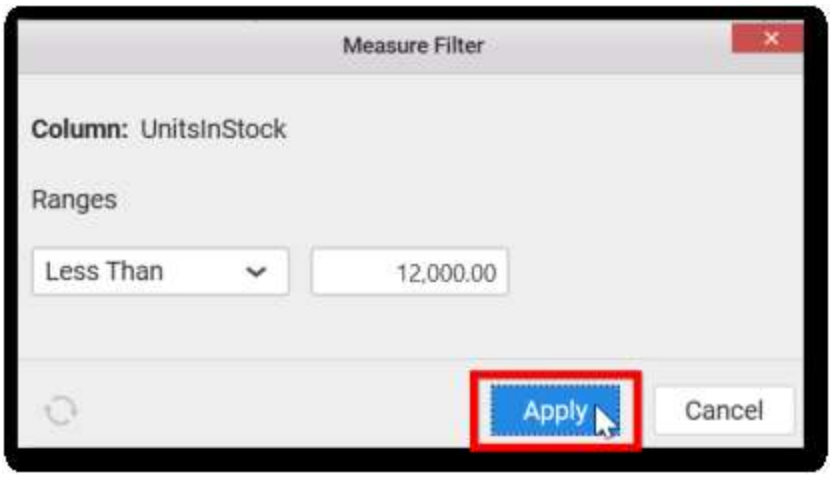


You can select what data to be displayed by choosing filter option.

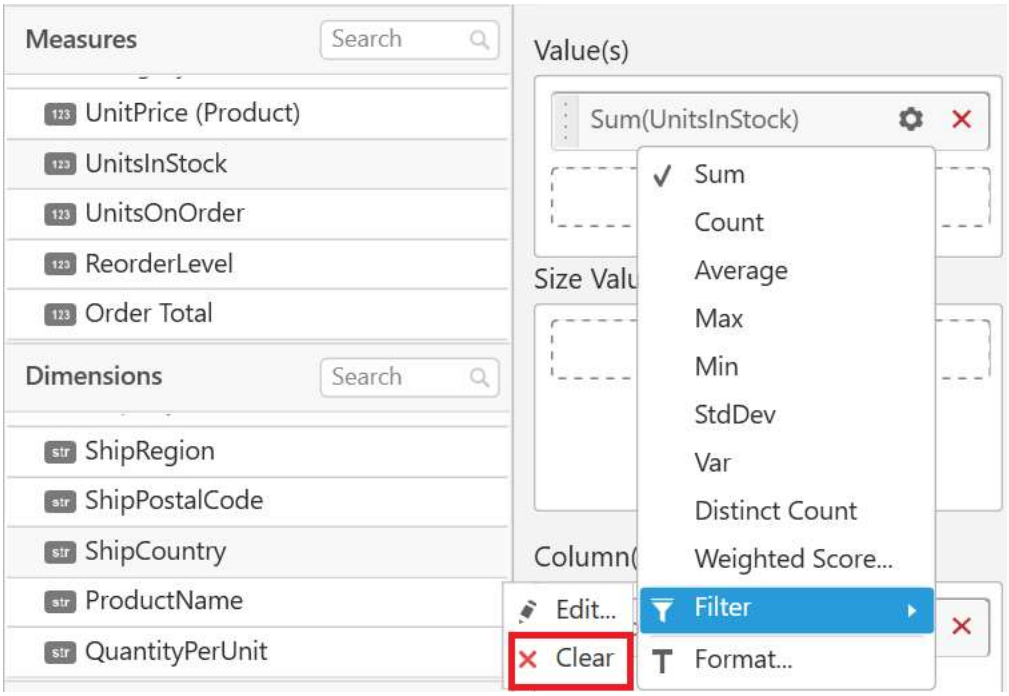
The Measure Filter option will be shown and you can choose the filter condition and apply the condition value.



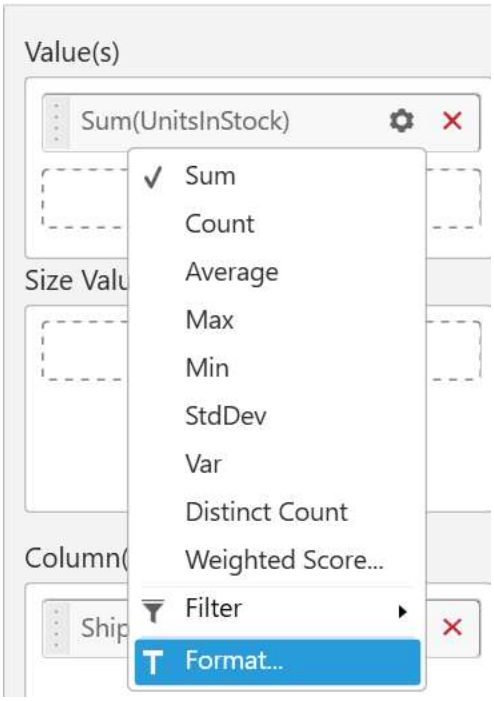




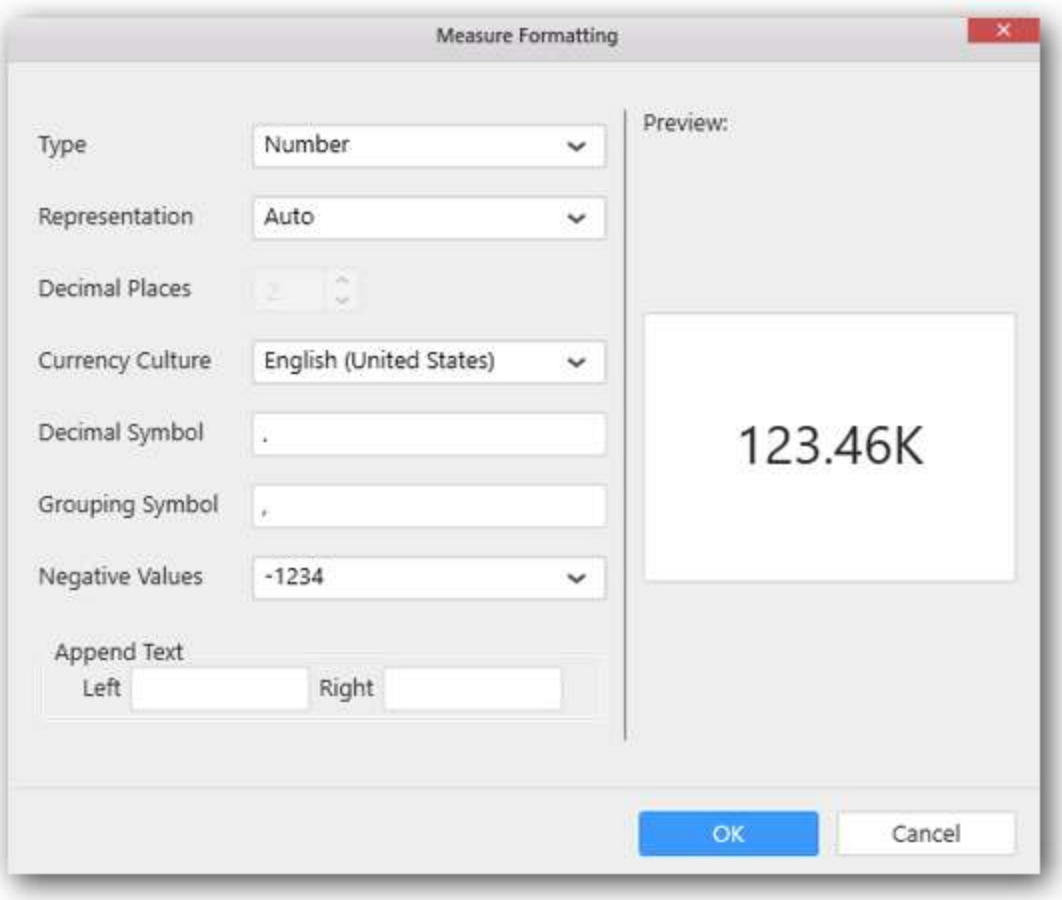
You can Clear the filter.



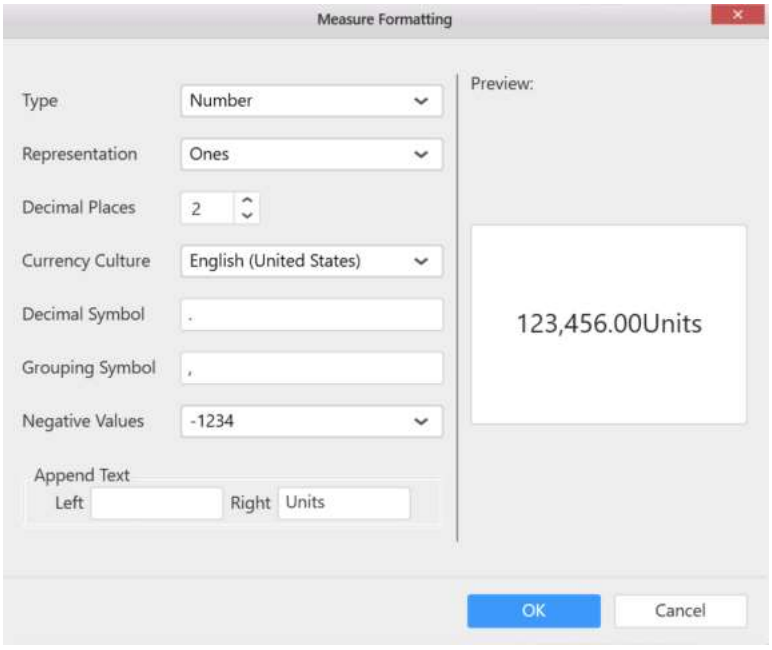
You can Format the value.



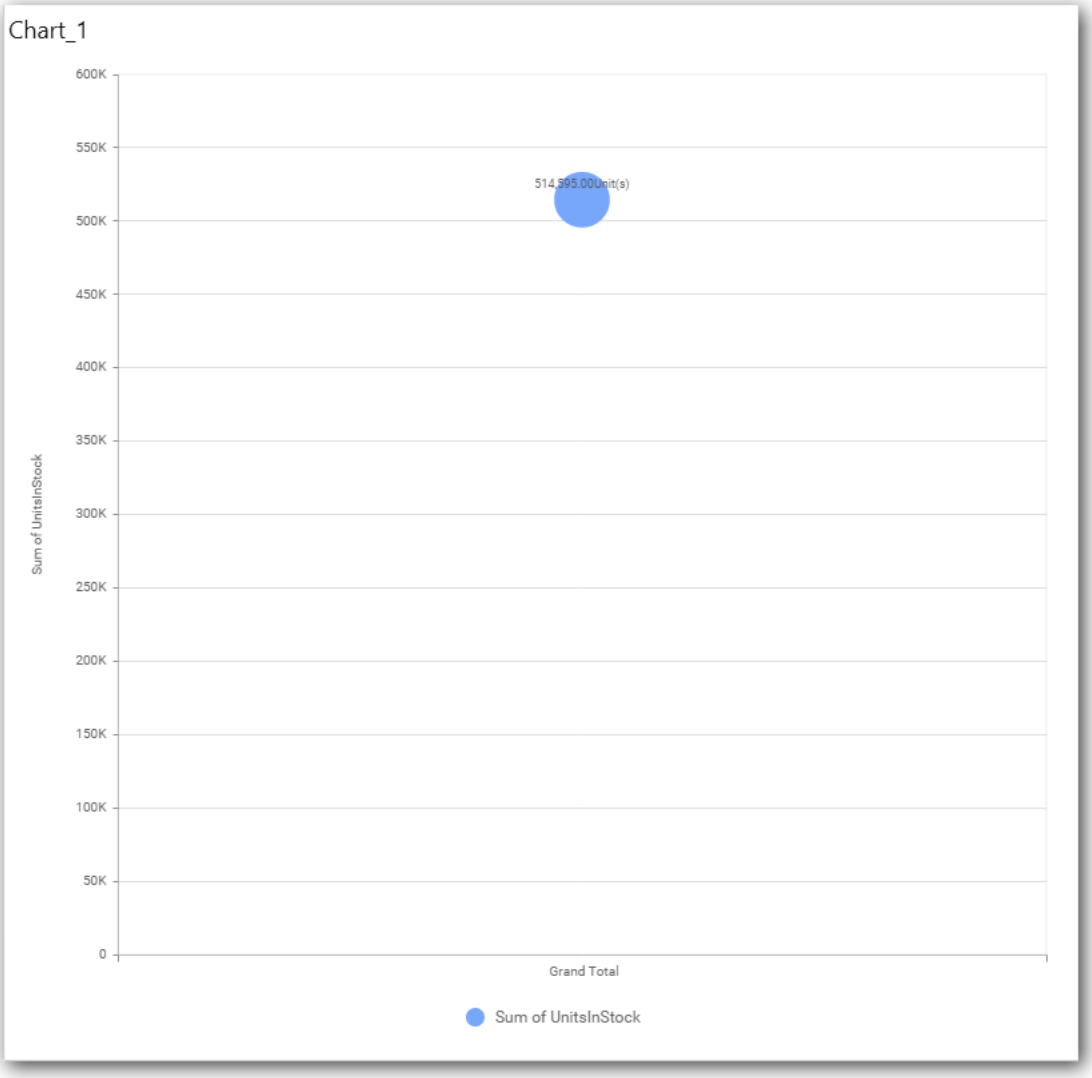
The format options will be shown.



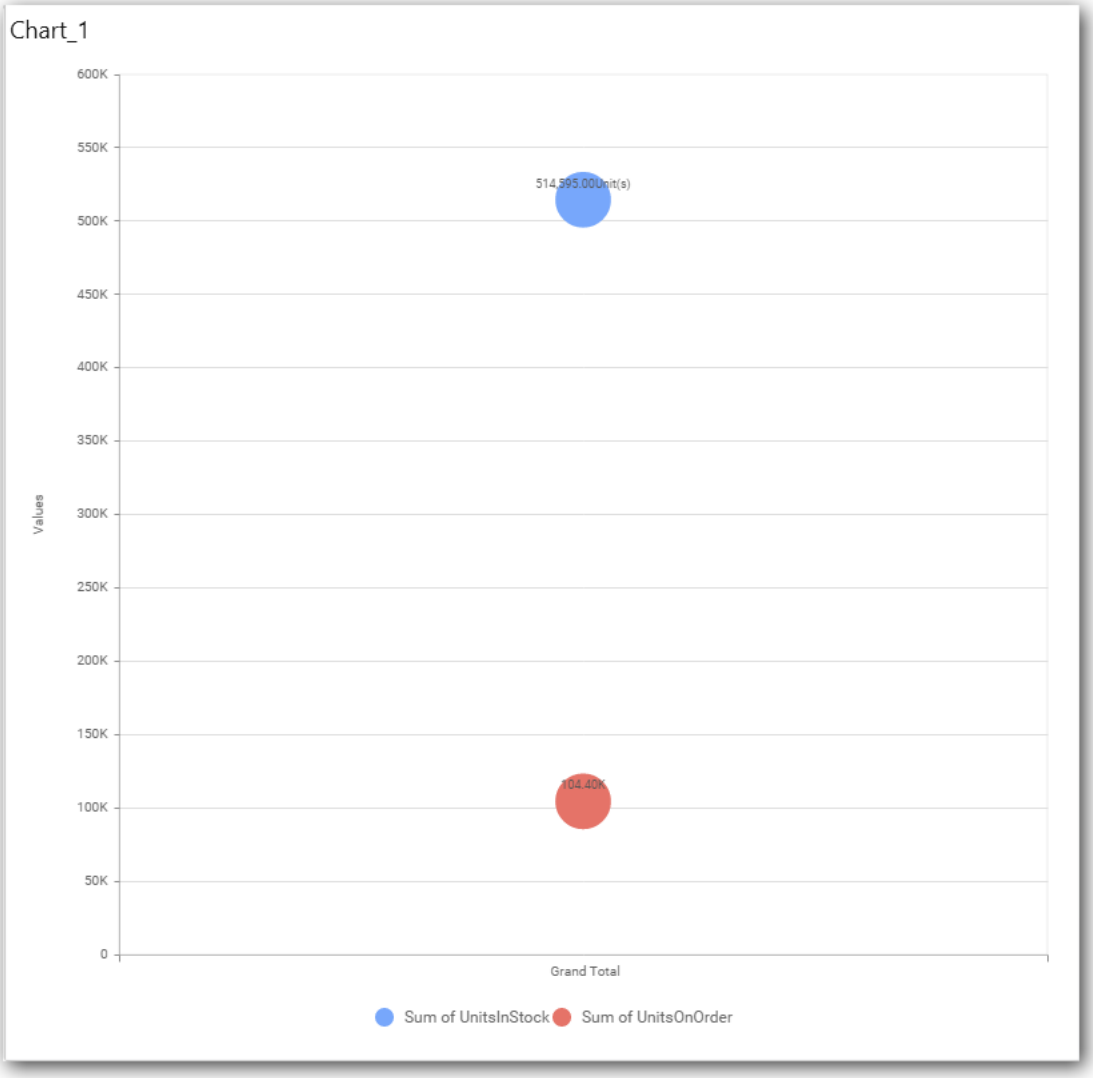
Choose the options you need and click OK.



Now the Chart will be rendered like this.



You can add more number of values by drag and drop the Measures into Value field.



You can also add Dimensions and Columns to Value(s).

Assigning Size Value(s)

You can add Measures into Size Value(s).

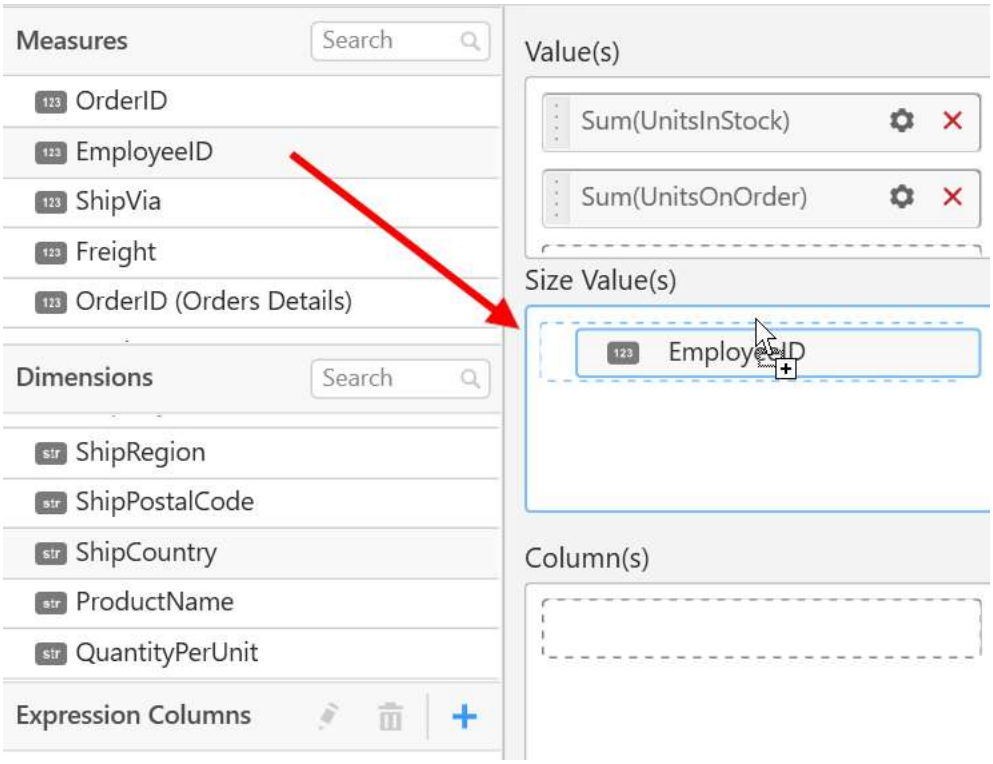


Chart bubble size will be calculated based on its corresponding size column values.

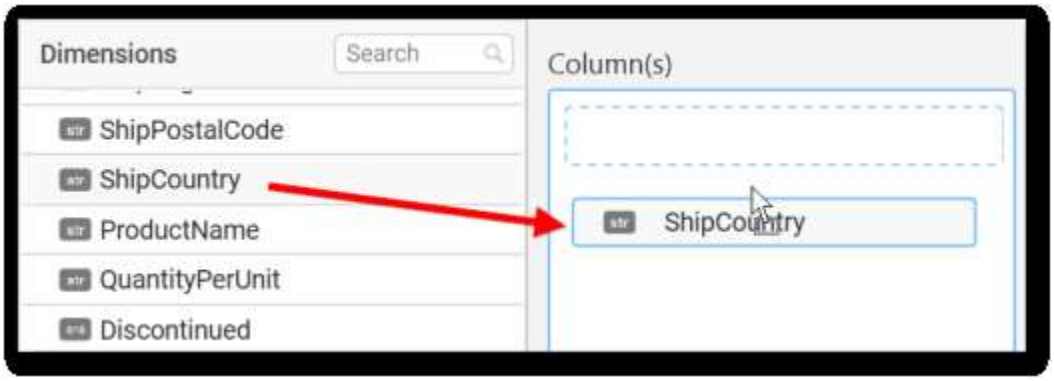
You can add any number of size values. First Value field will be calculated based on first Size Value(s). Second Value field will be calculated based on second Size Value(s).

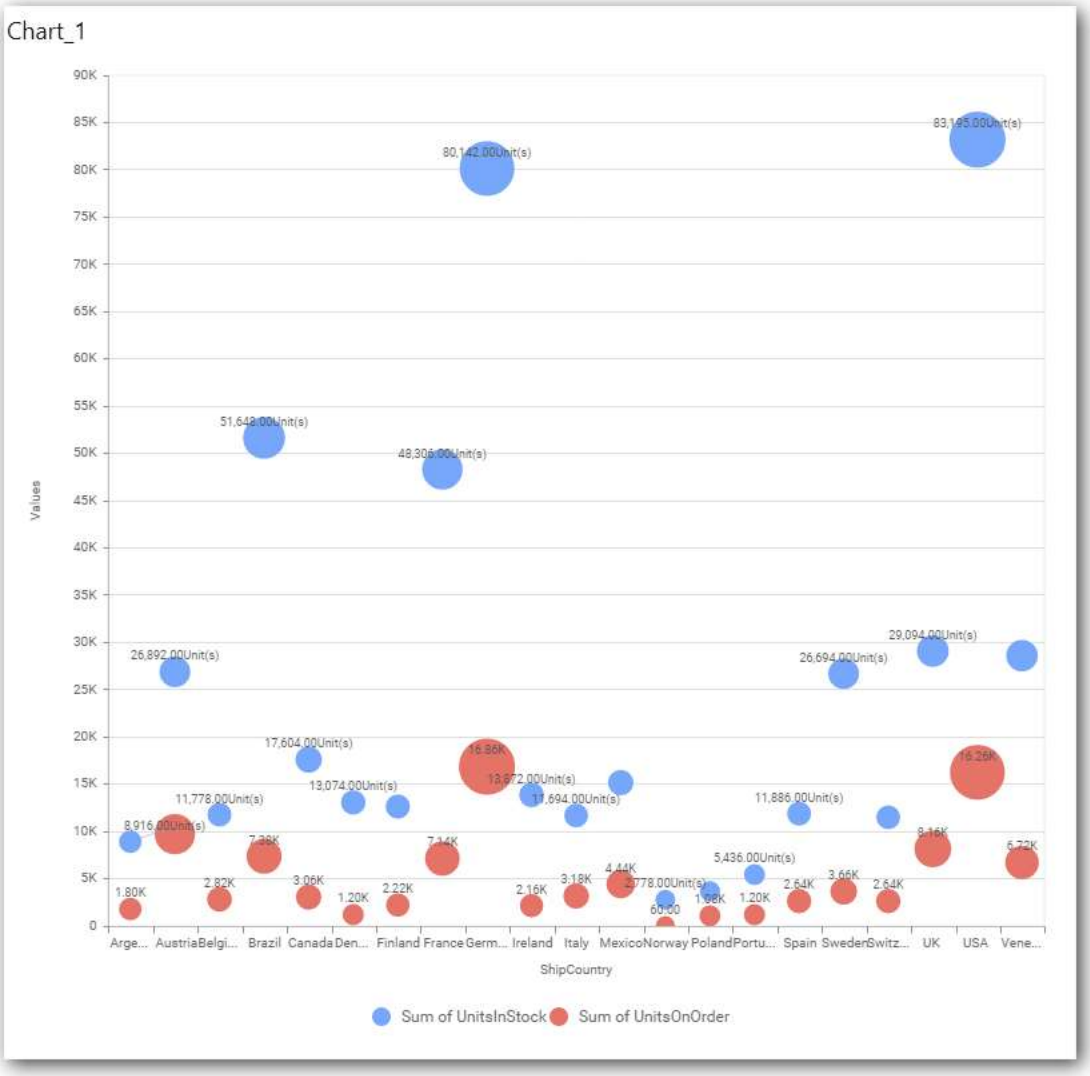
If there is no corresponding Size Value for Value field, then Value field size will be calculated based on Value field.

you can also add Dimension into Size Value(s).

Assigning Column(s)

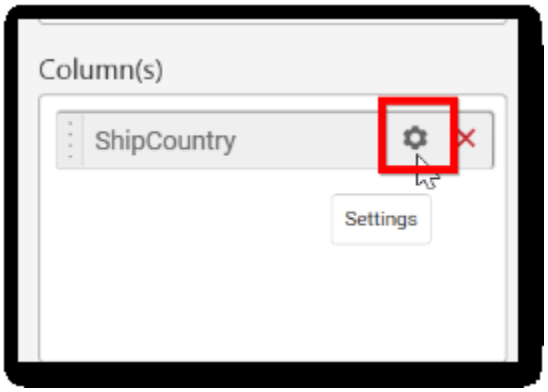
You can add the Dimension into Column field by drag drop.

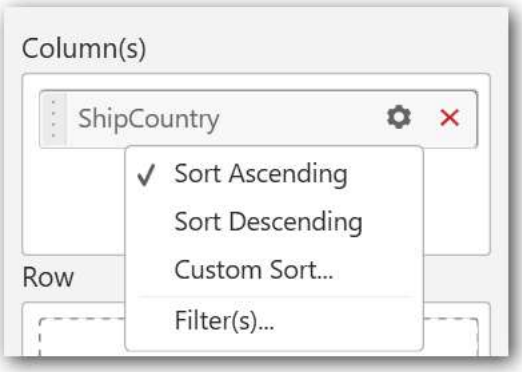




You can also add Measures and Expression Columns into Column(s) field.

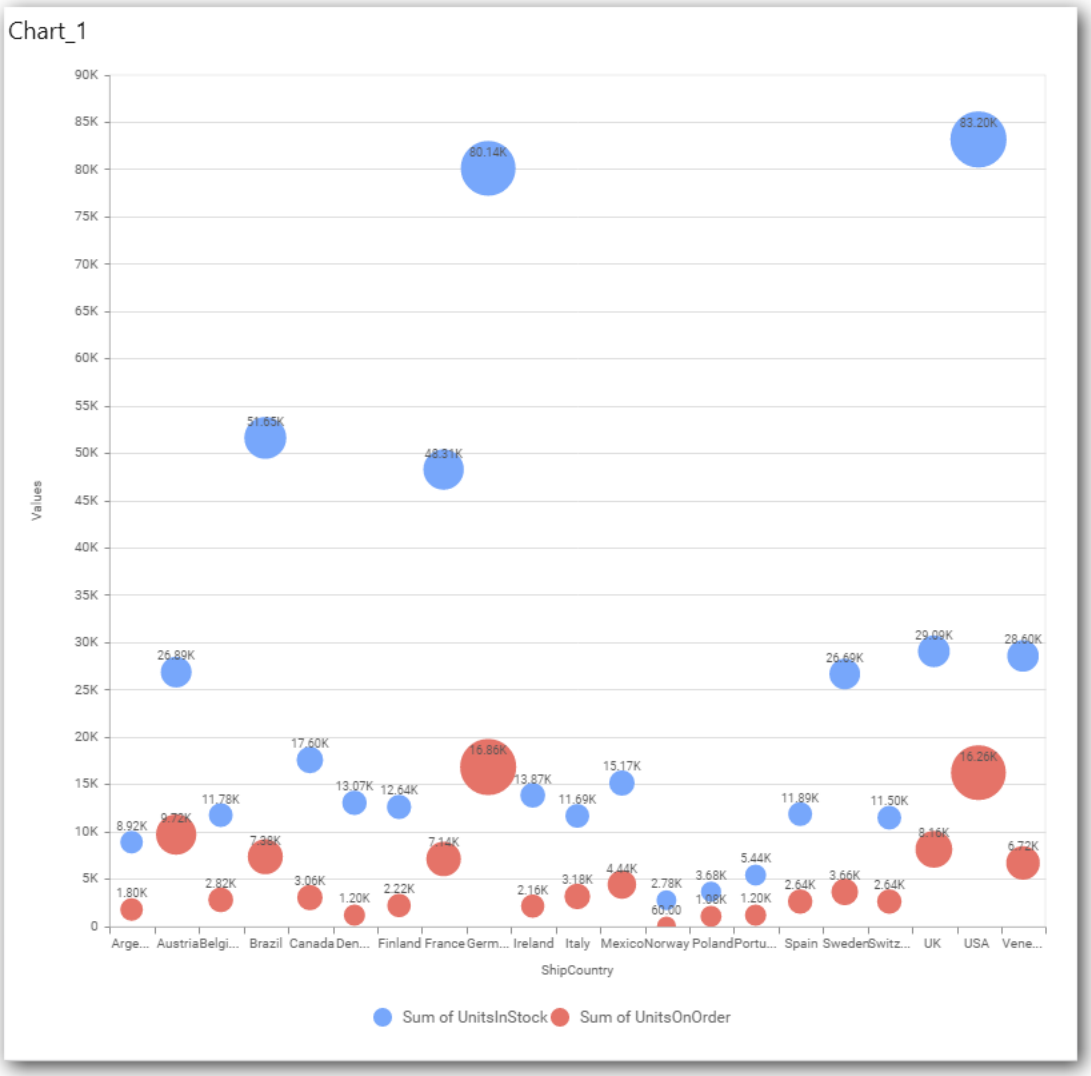
You have options to change the settings.



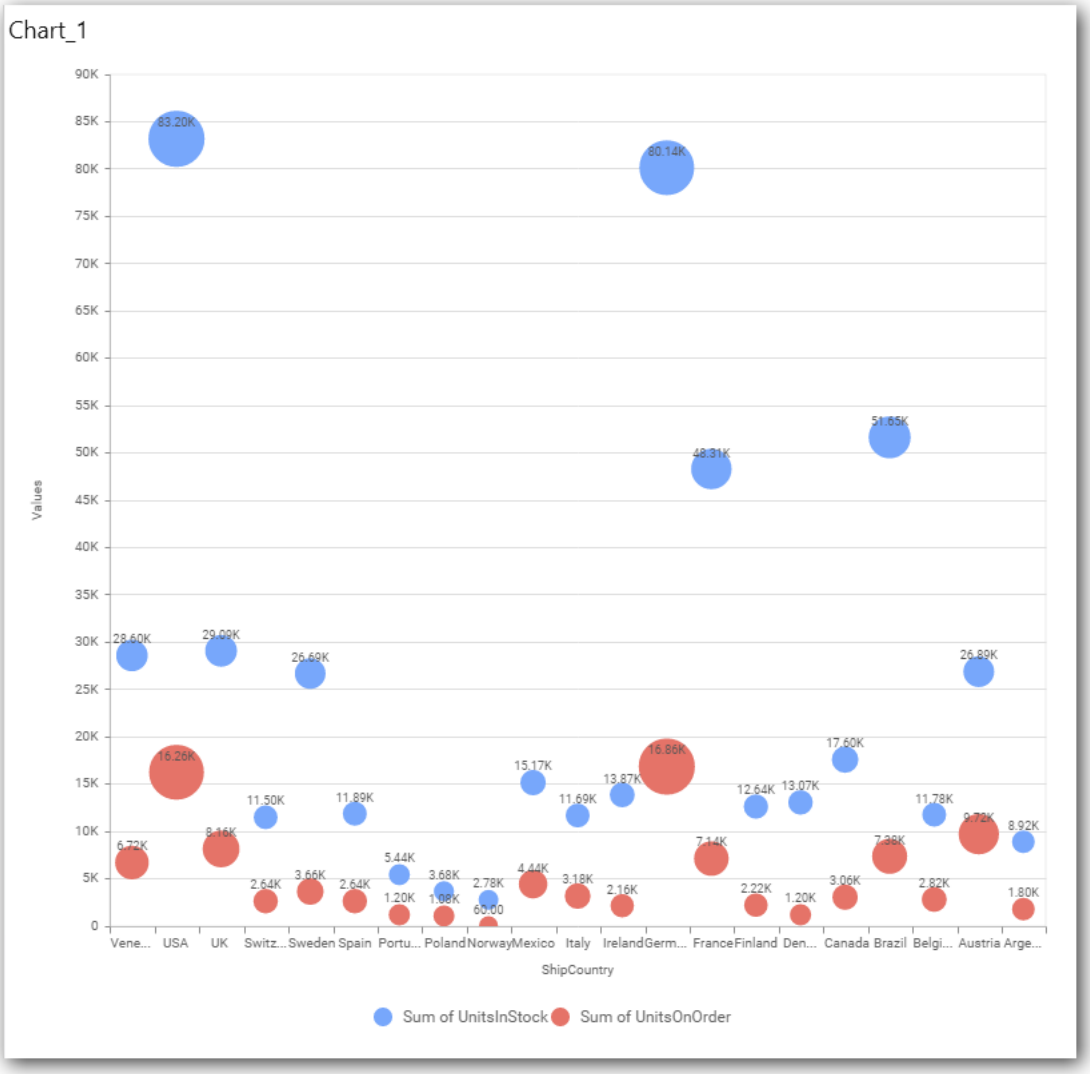


You can sort the chart either in Ascending or Descending series.

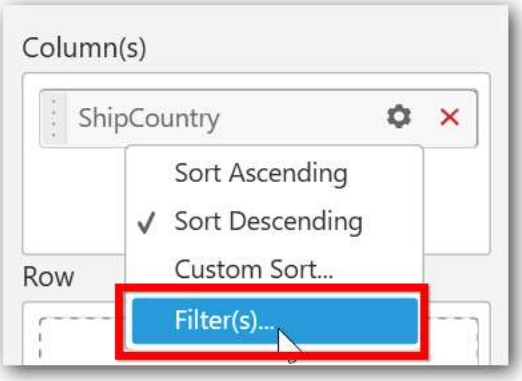
Ascending Order:

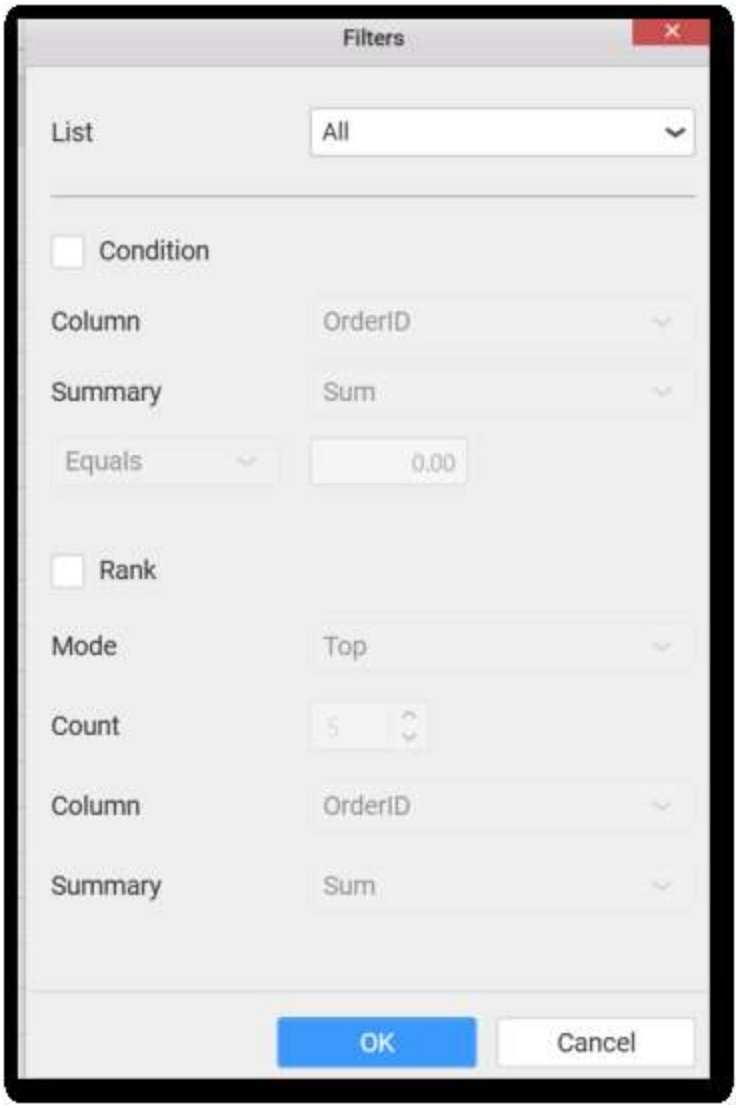


Descending order:

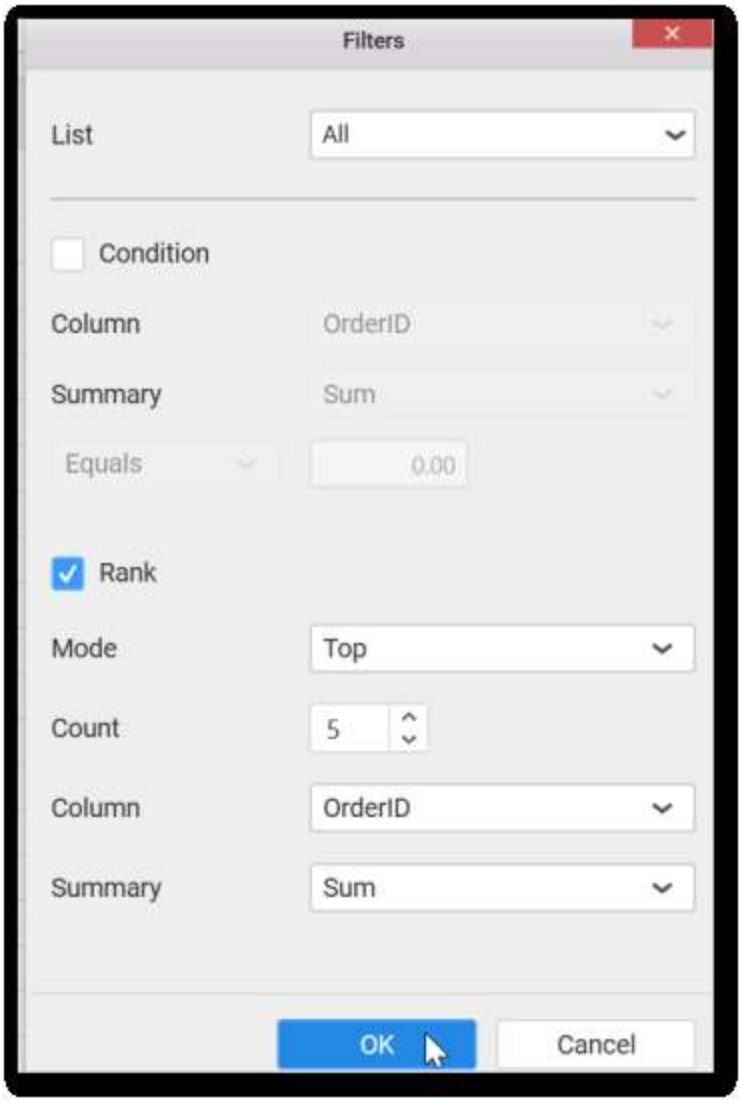


You can apply a filter

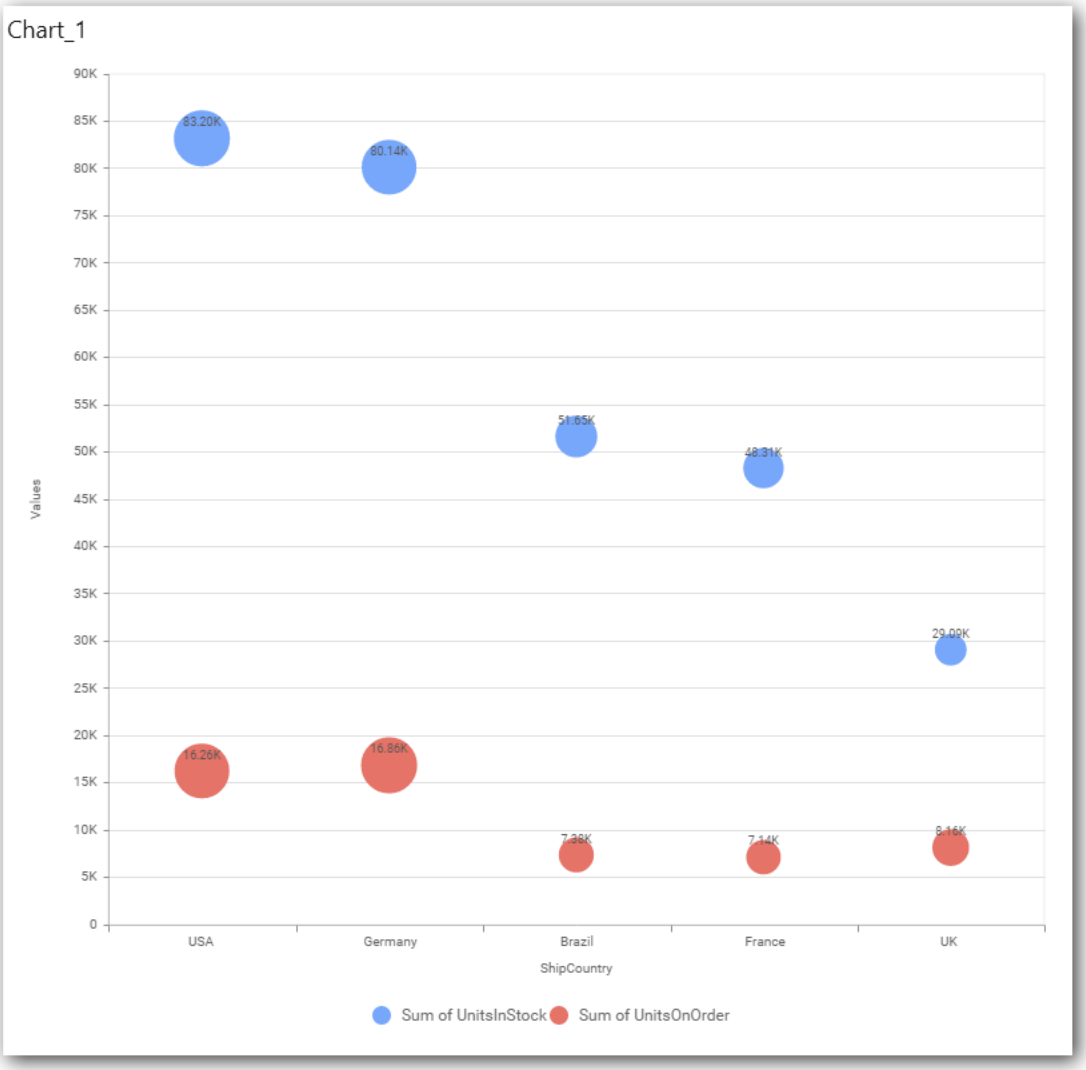




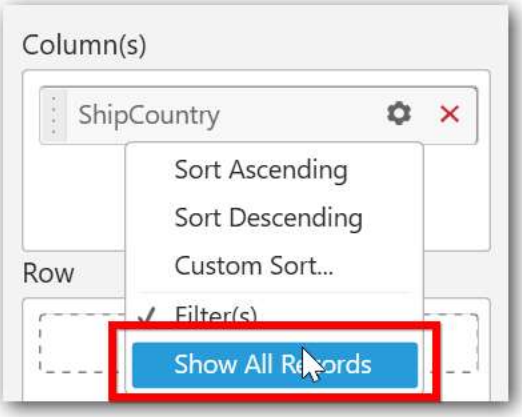
Select the Conditions and Rank you need.



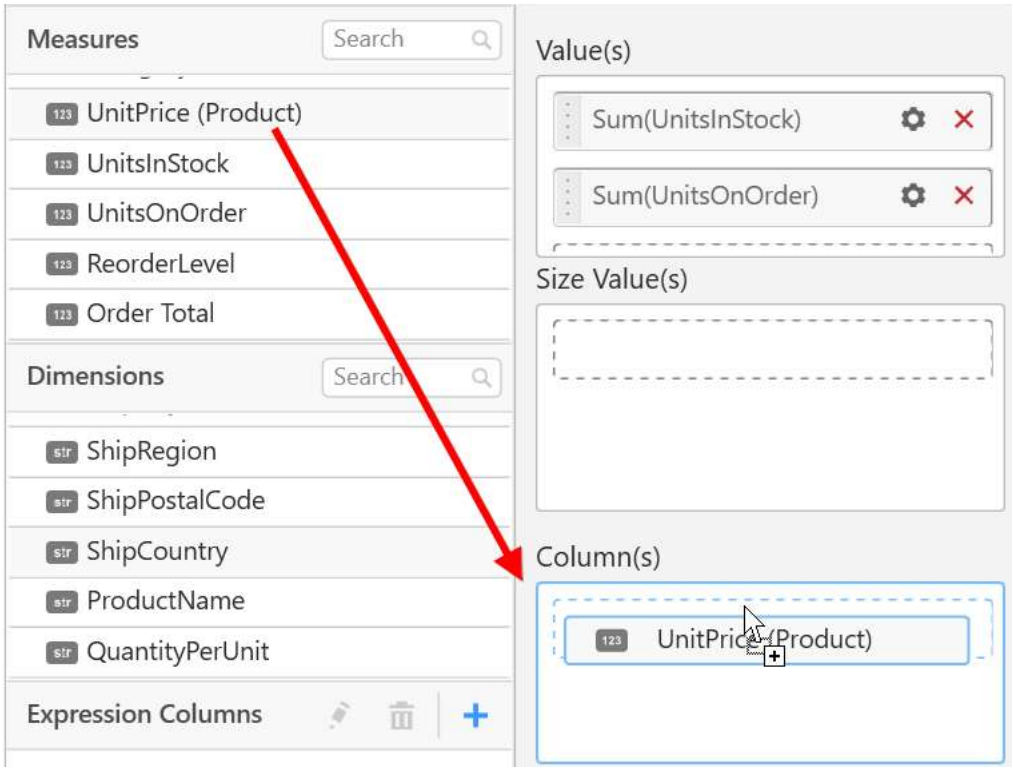
Now the chart will be rendered like this.



To show all records again click on Show All Records

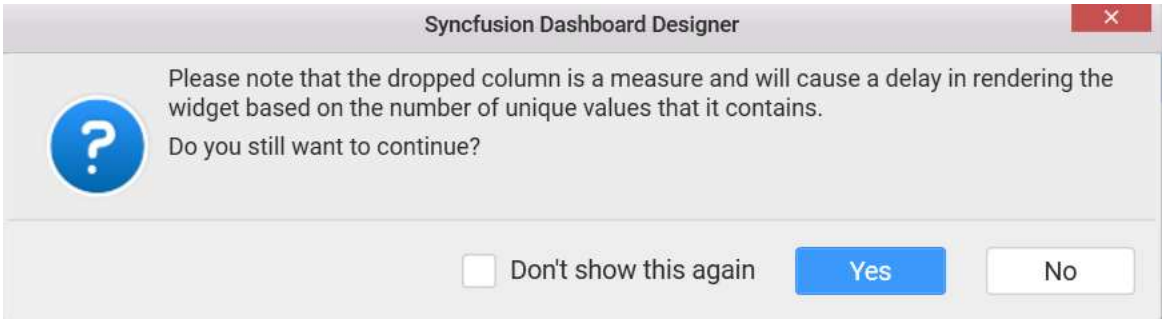


You can add Measures into Column(s).



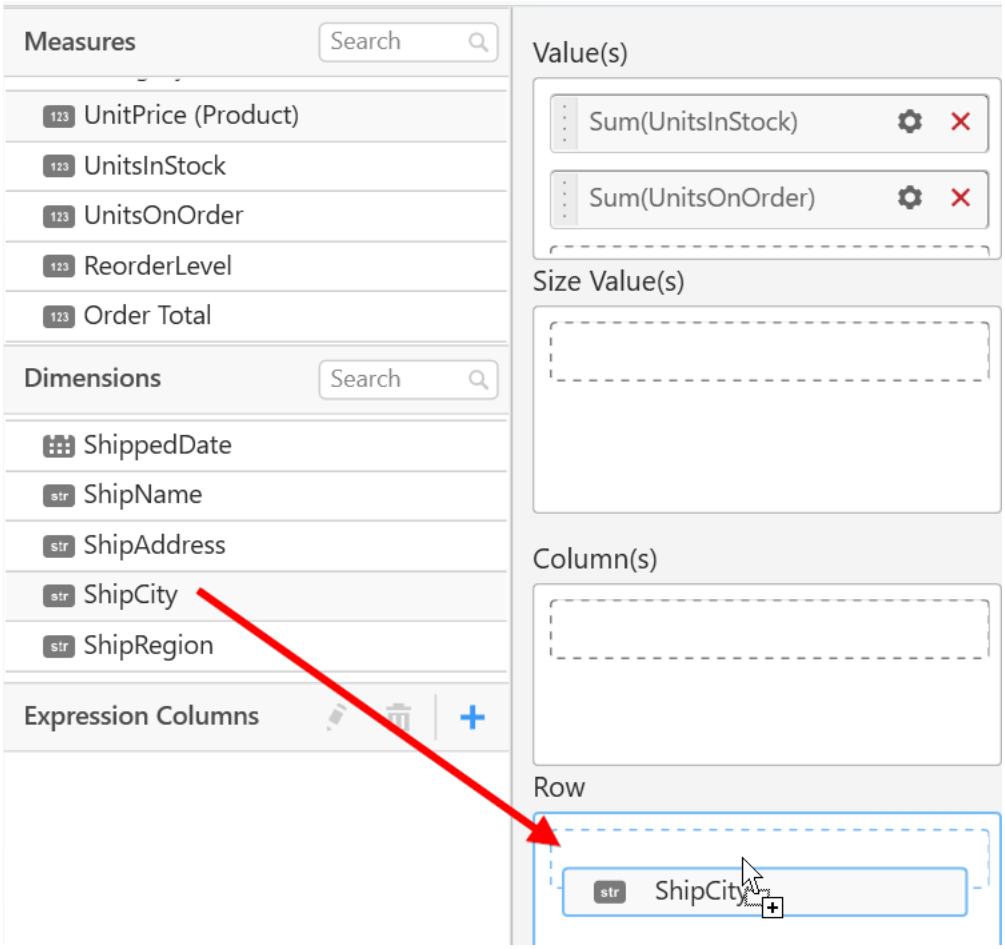
The following alert message will be shown.

To continue click Yes, otherwise choose No.

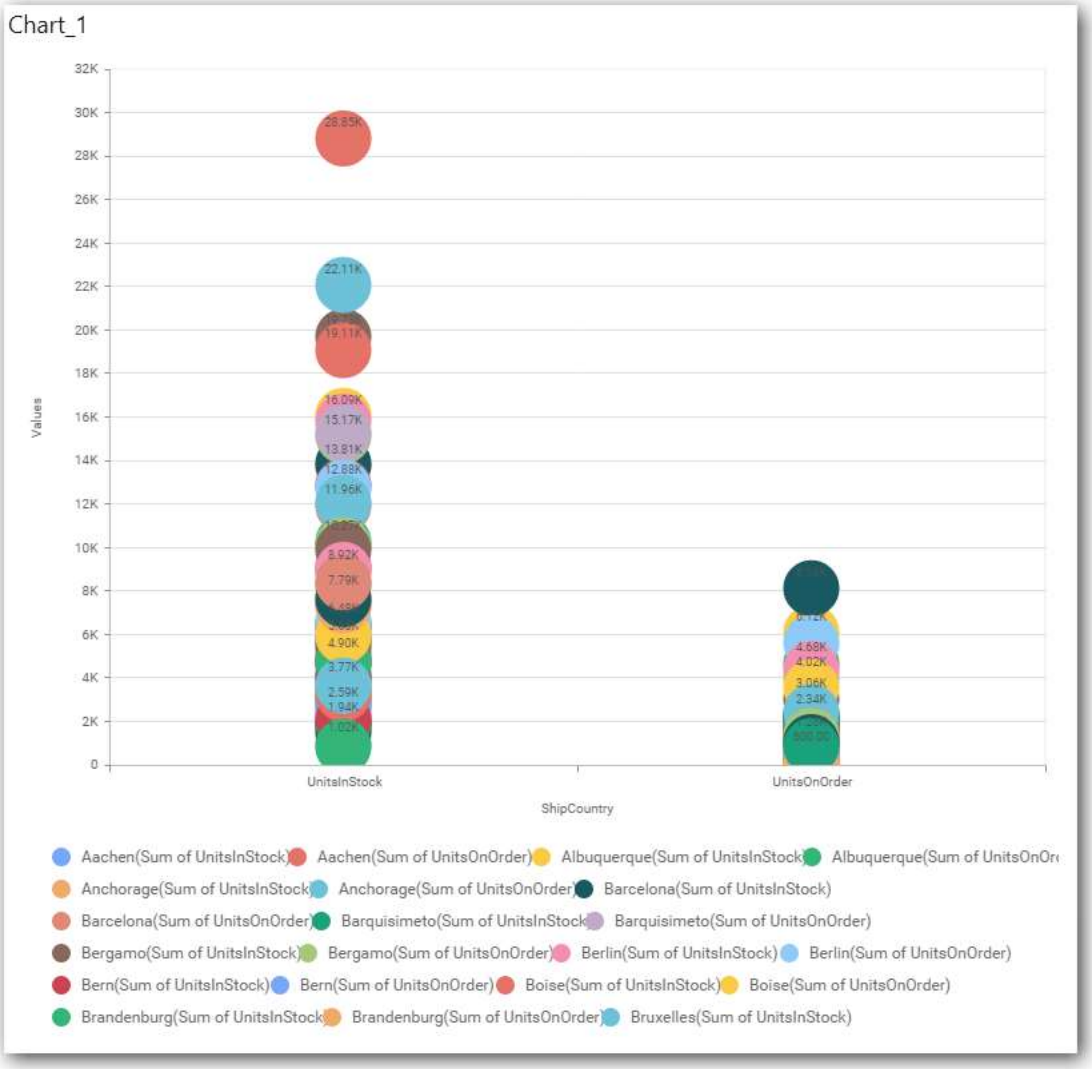


Assigning Row

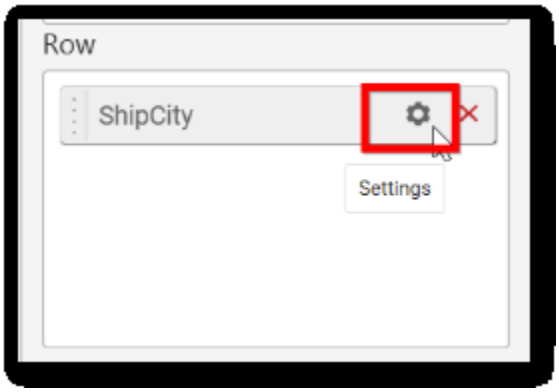
You can add Dimension into the Row field for series chart.



The chart will be rendered in series as shown in the image.



You have settings options similar to Column(s).



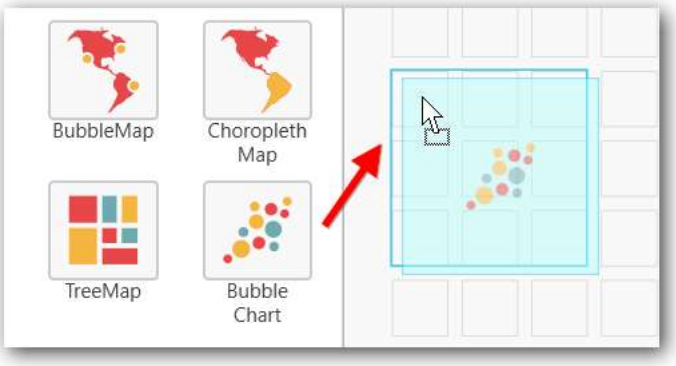
How to configure SSAS data to Bubble Chart?

Bubble Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that

you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

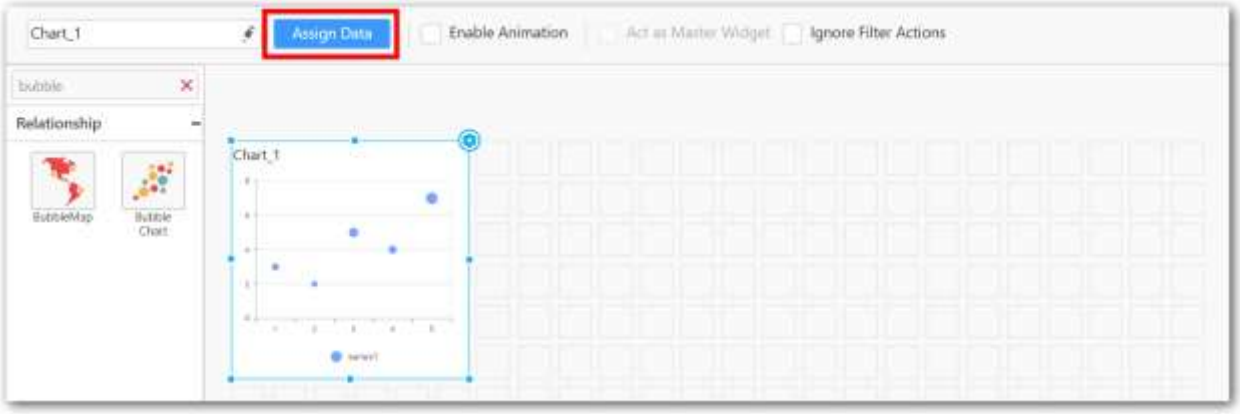
Following steps illustrates configuration of SSAS data to bubble chart.

Drag and drop the Bubble chart into canvas and resize it to your required size.

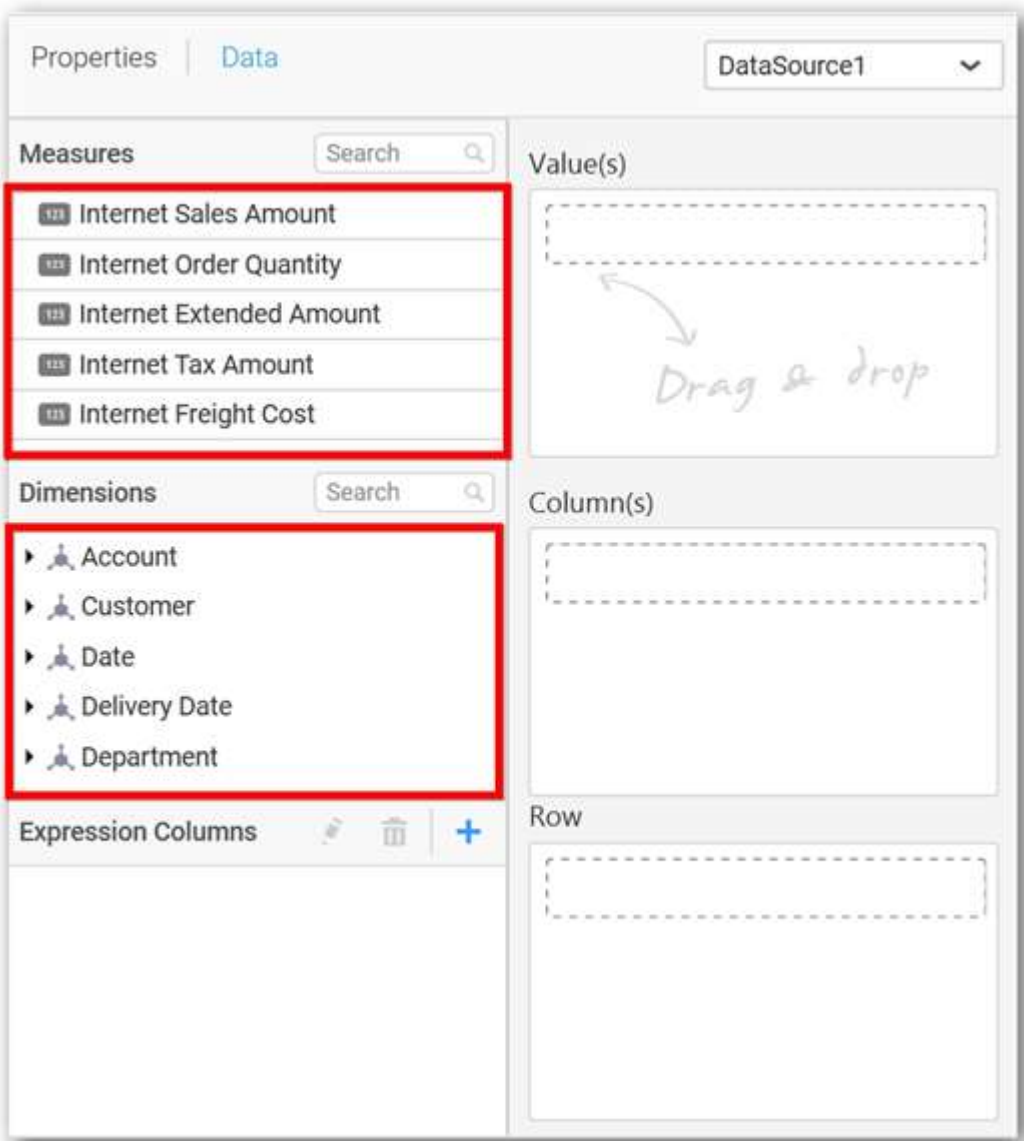


Select the dropped widget using mouse.

Click the Assign Data button in the toolbar.

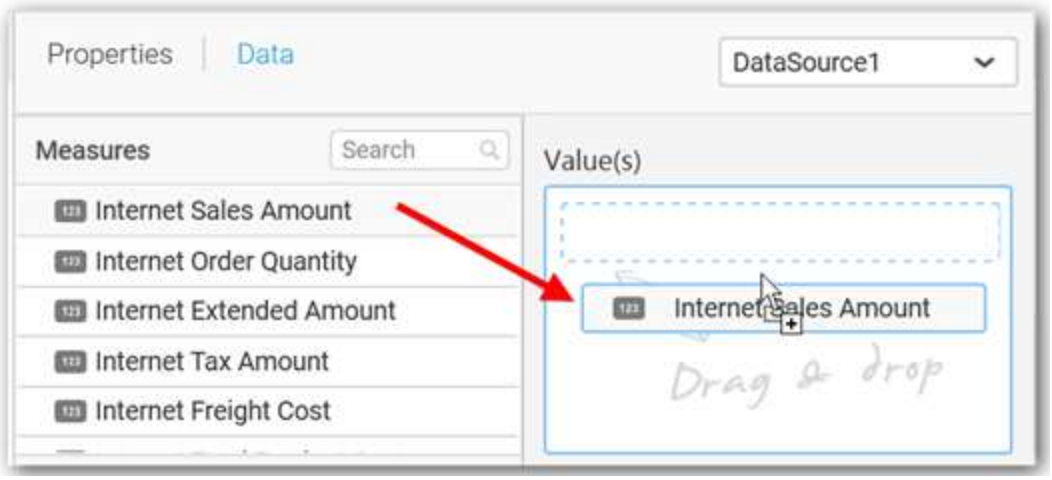


A Data pane will be opened with available Measures and Dimensions.

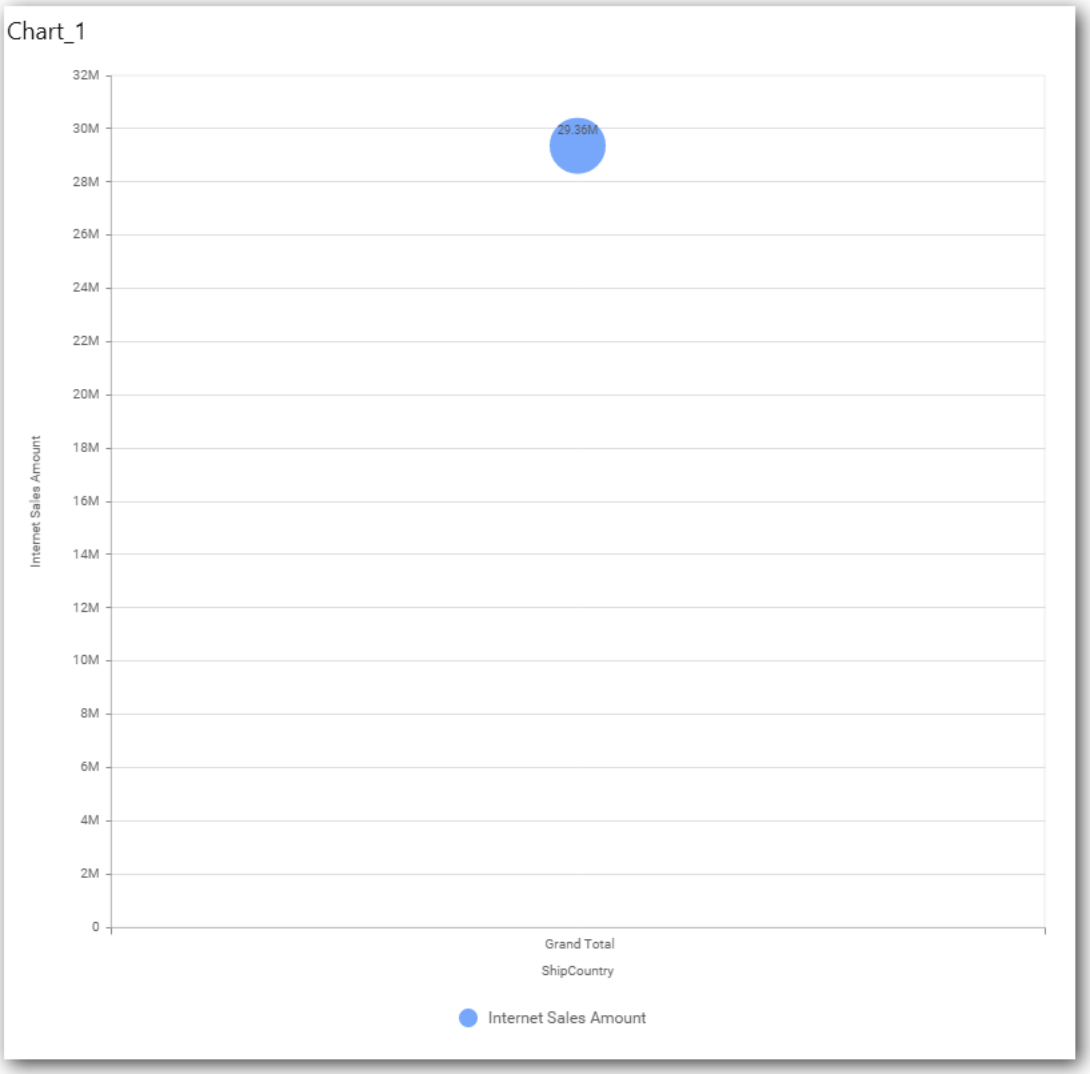


Assigning Value(s)

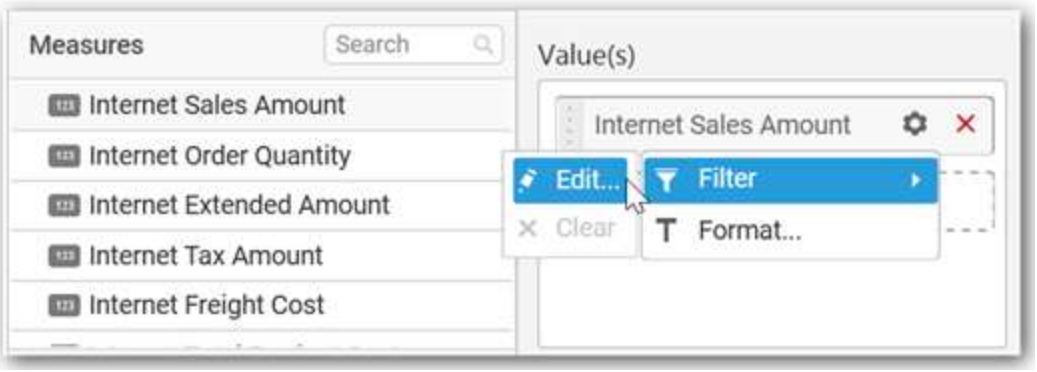
Drag and drop a column under Measures category into Value(s) section



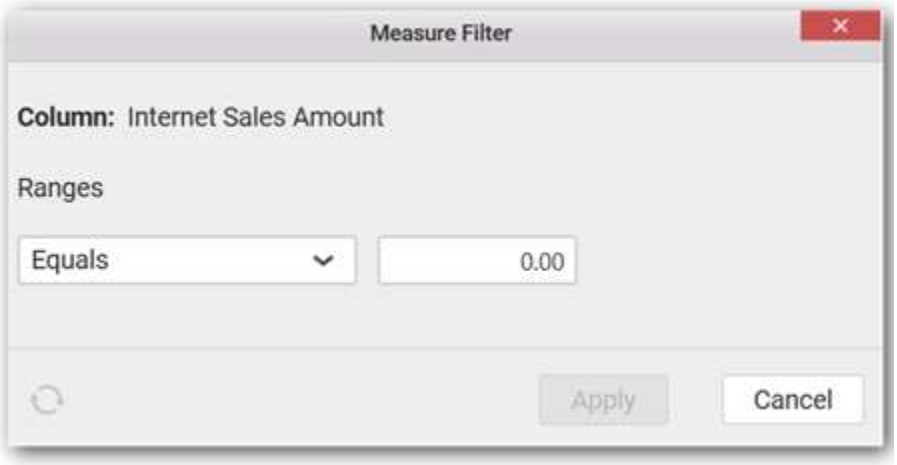
Now the chart will be rendered like this.



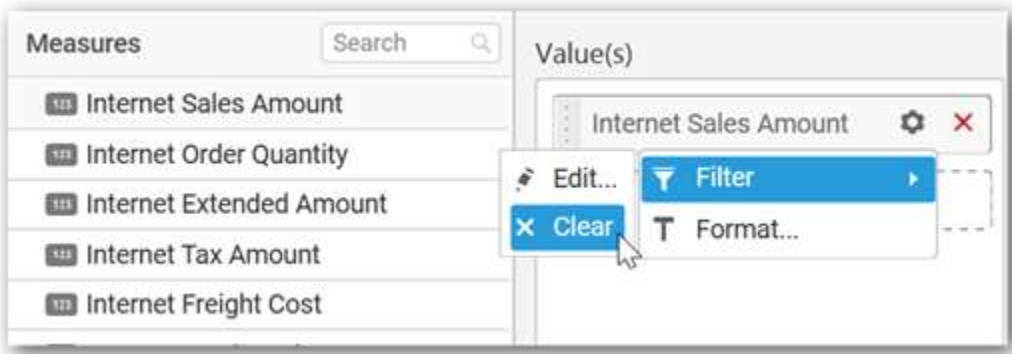
Define the filter criteria to match through choosing Edit option in Filter menu item.



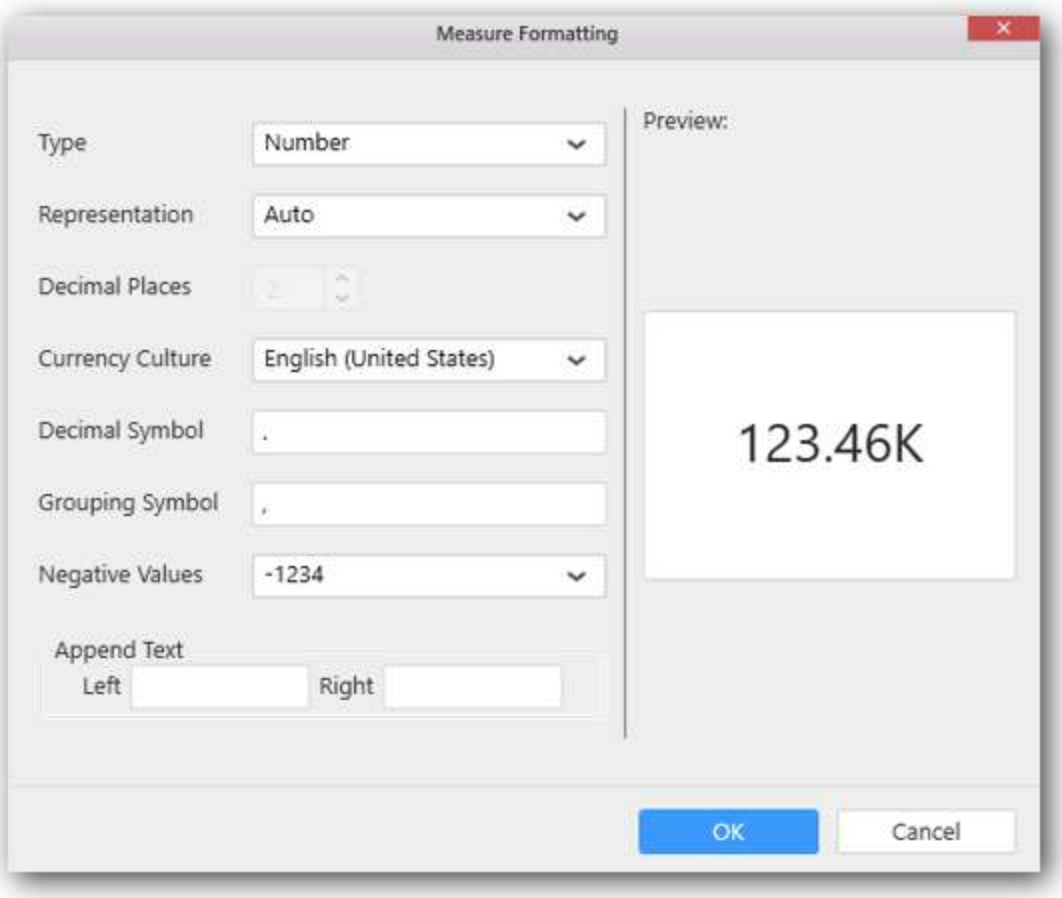
The Measure filter dialog will be shown and you can choose the filter condition and apply the condition value.



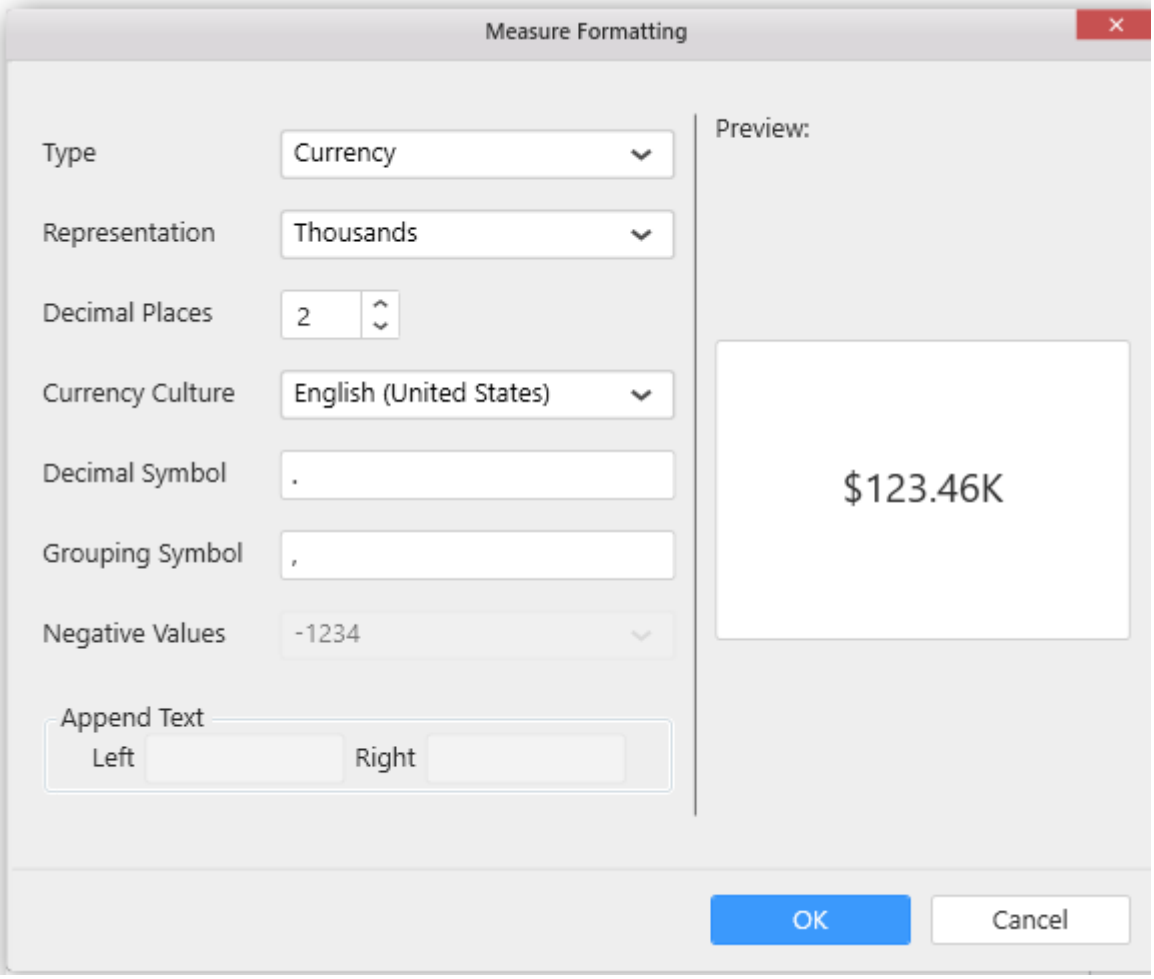
Select Clear option to clear the defined filter.



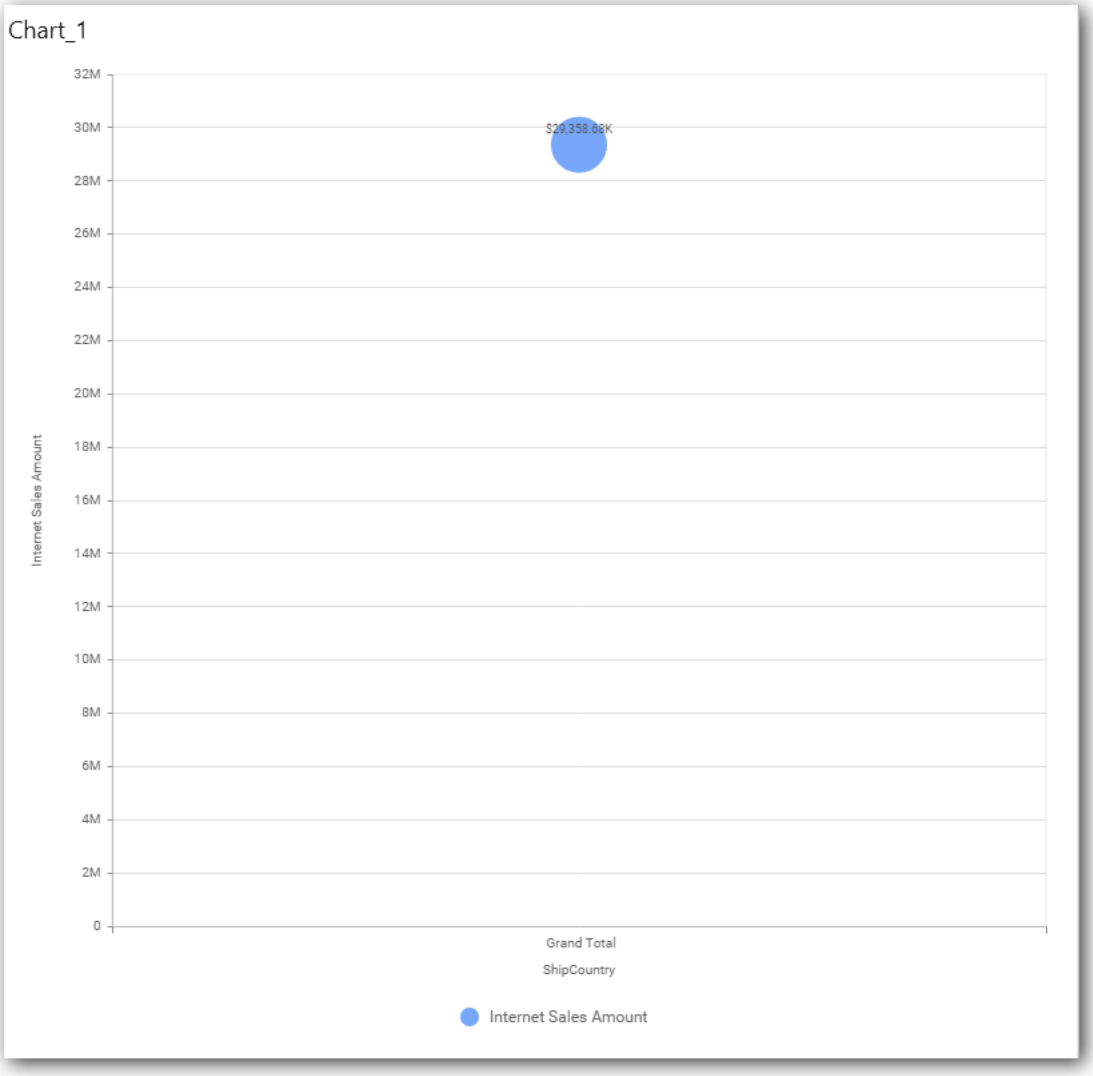
Select Format option to define the display format to the values in the column through Measure Formatting window.



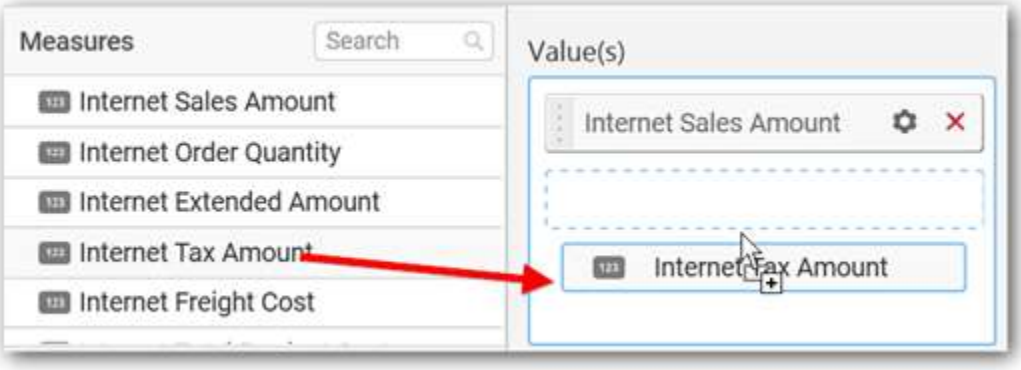
Choose the options you need and click OK.

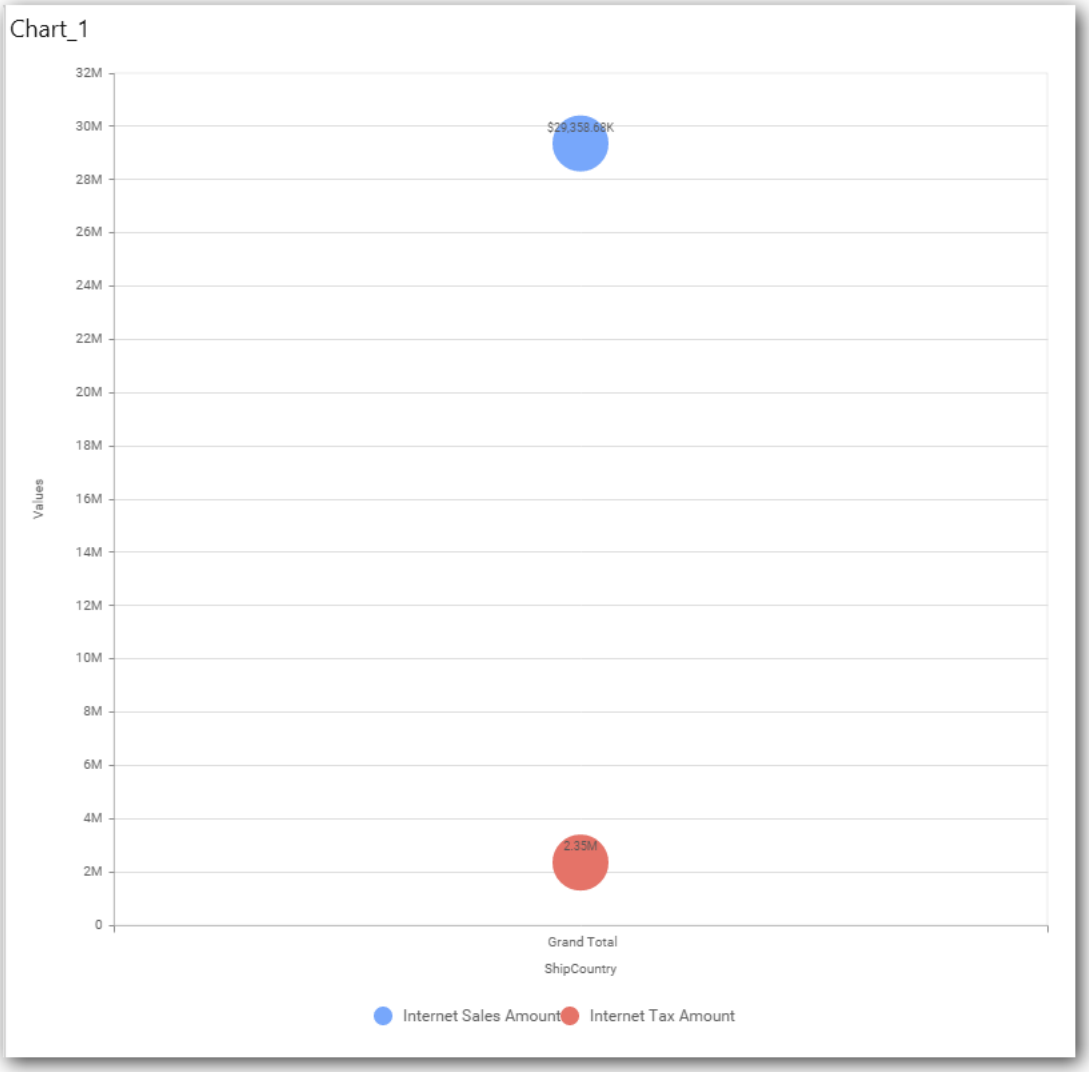


Now the Chart will be rendered like this.



You can also add more than one column to the Value(s) section.





Assigning Size Value(s)

You can add Measures into Size Value(s).

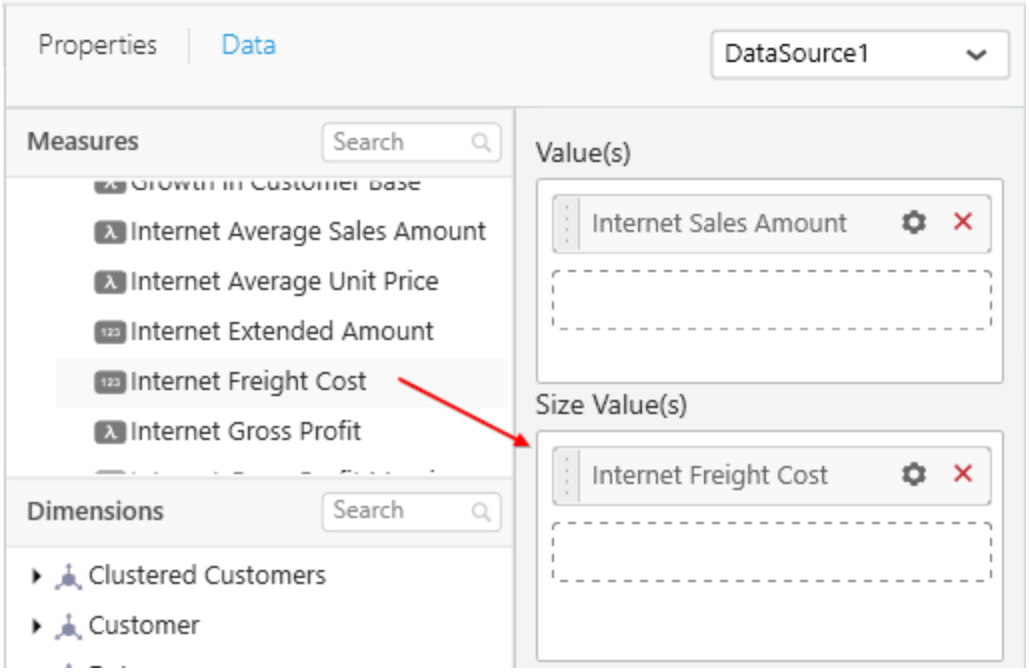


Chart bubble size will be calculated based on its corresponding size column values.

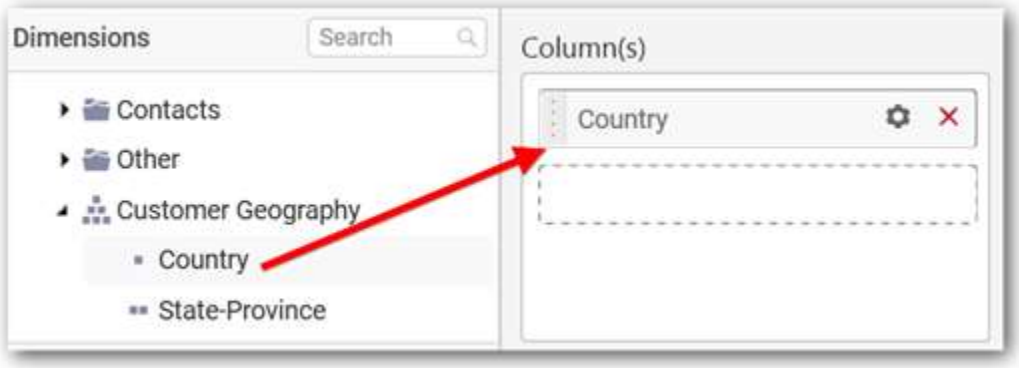
You can add any number of size values. First Value field will be calculated based on first Size Value(s). Second Value field will be calculated based on second Size Value(s).

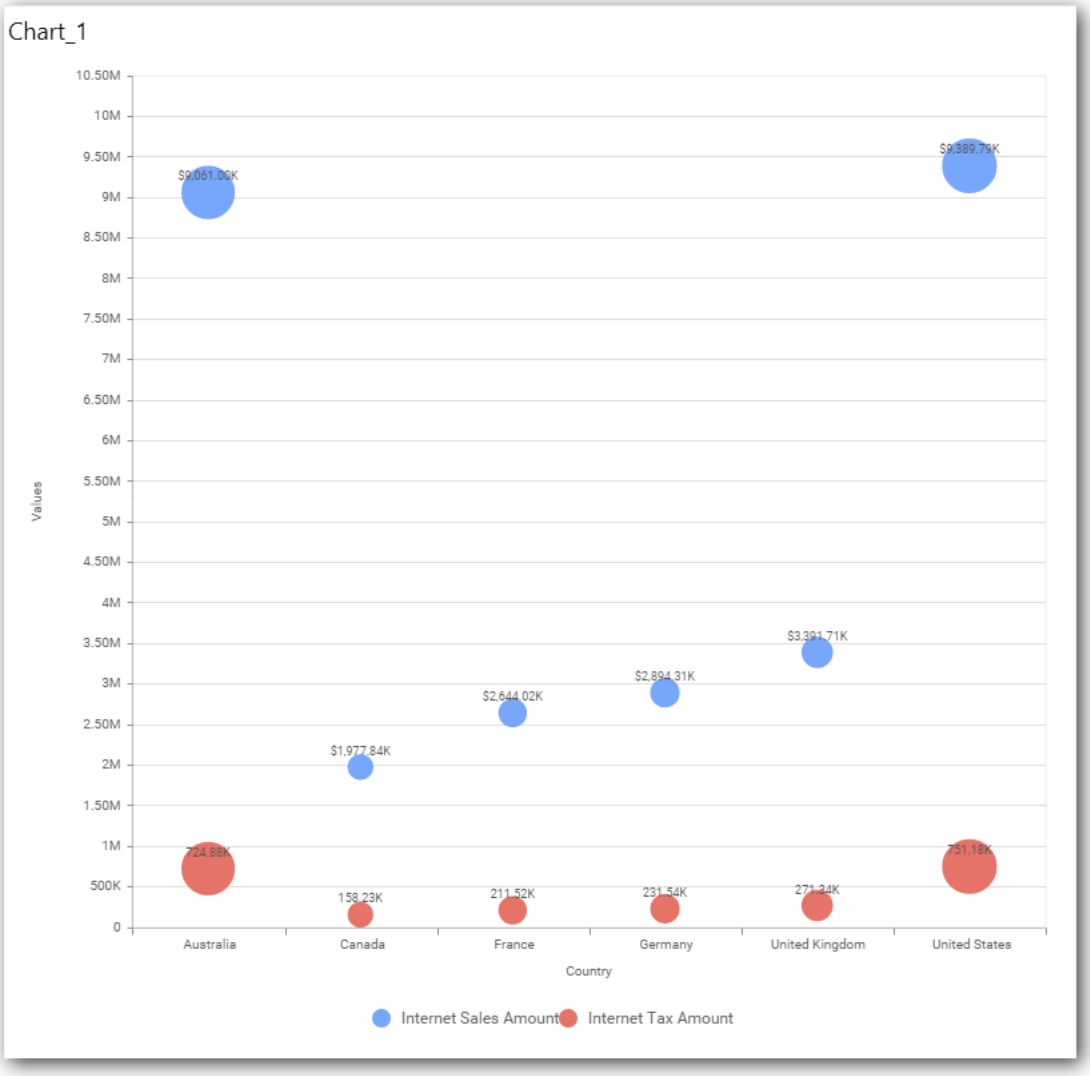
If there is no corresponding Size Value for Value field, then Value field size will be calculated based on Value field.

you can also add Dimension into Size Value(s).

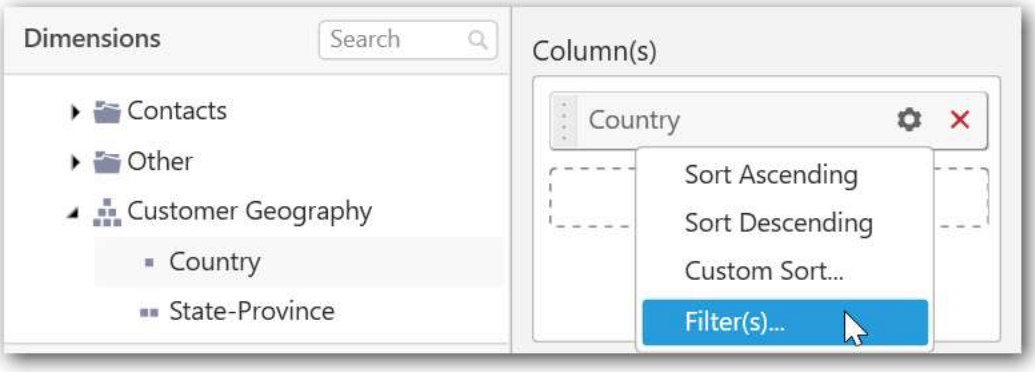
Assigning Column(s)

Add a dimension level or hierarchy into Column(s) section through drag and drop.

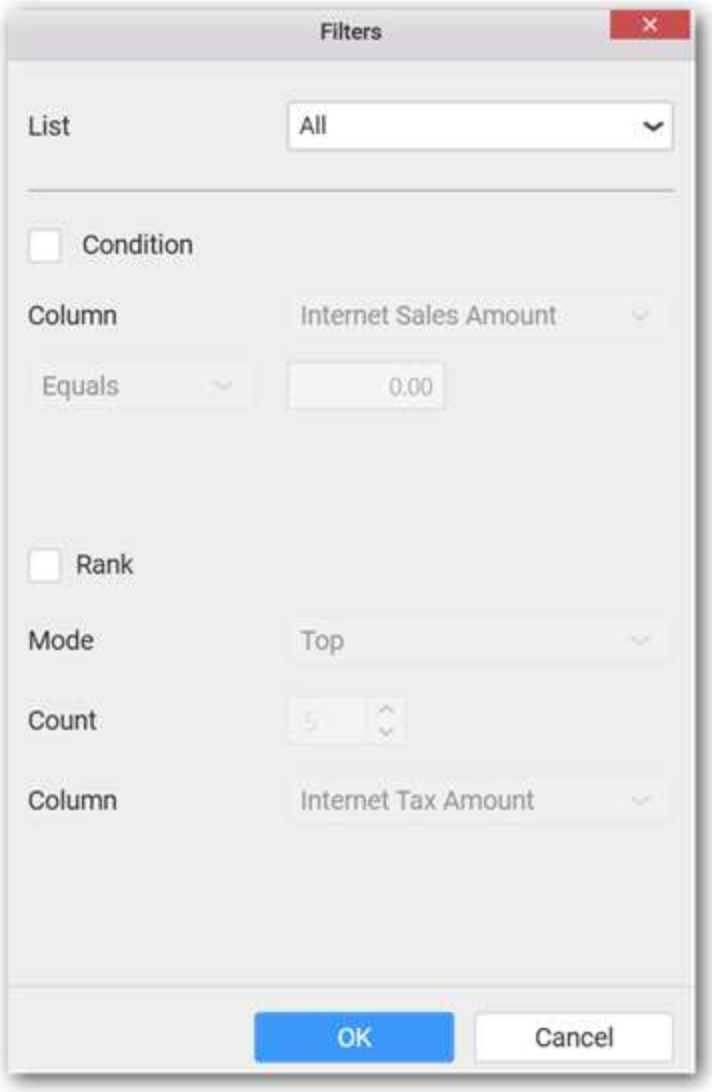




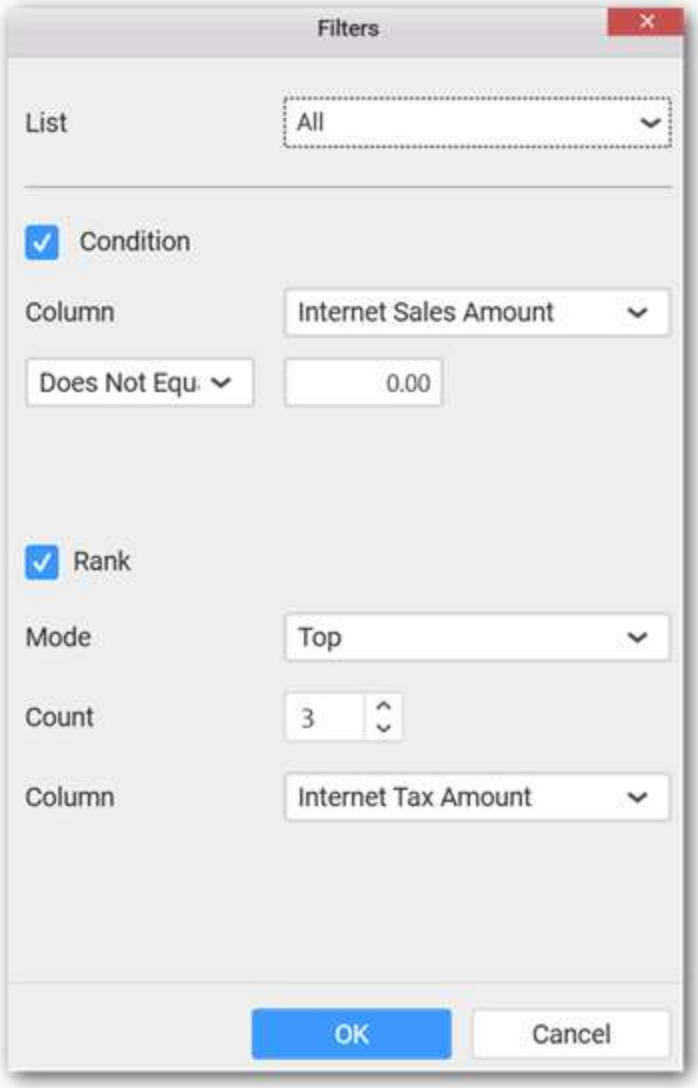
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



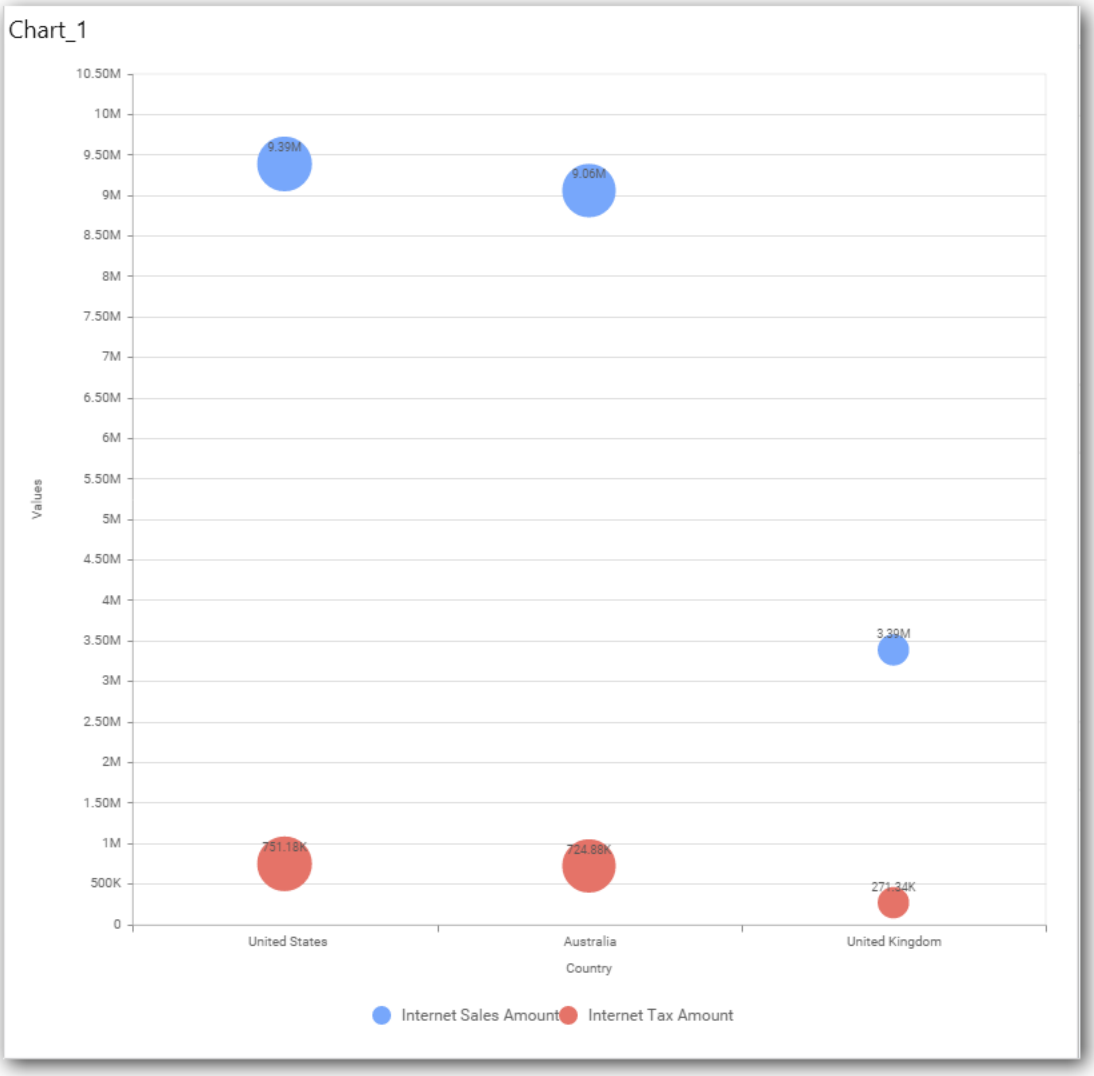
Select Filter(s)... to launch the Filters window.



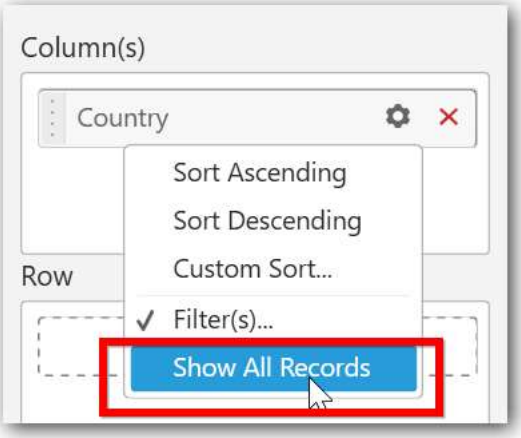
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

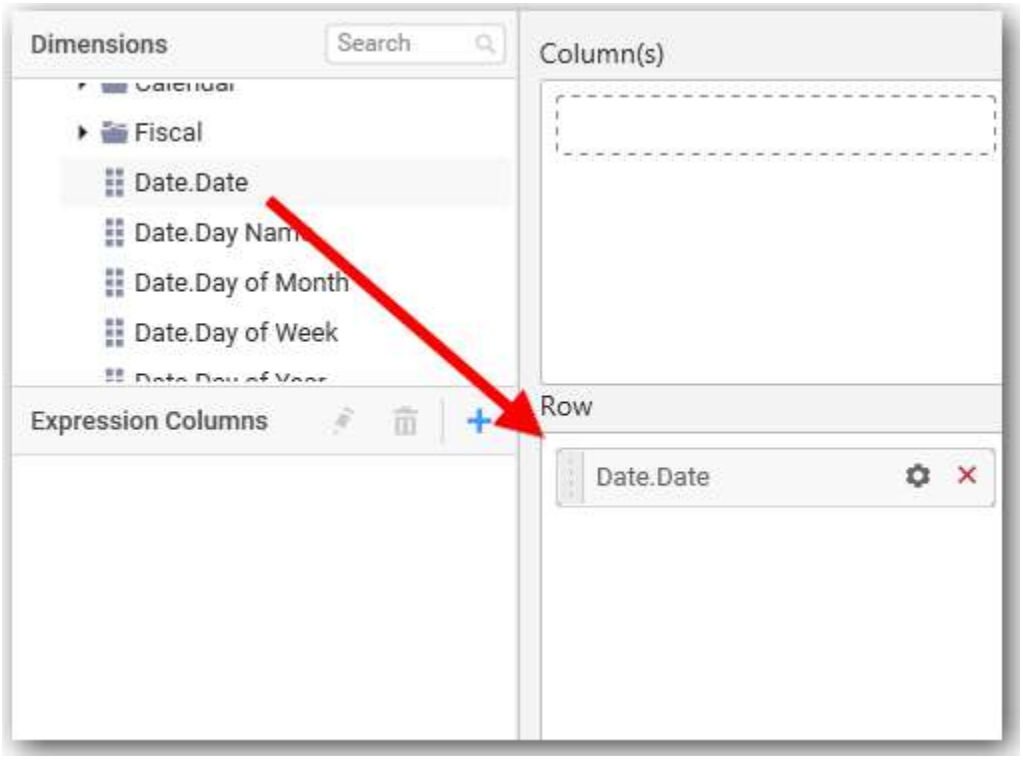


To show all records again click on Show All Records.

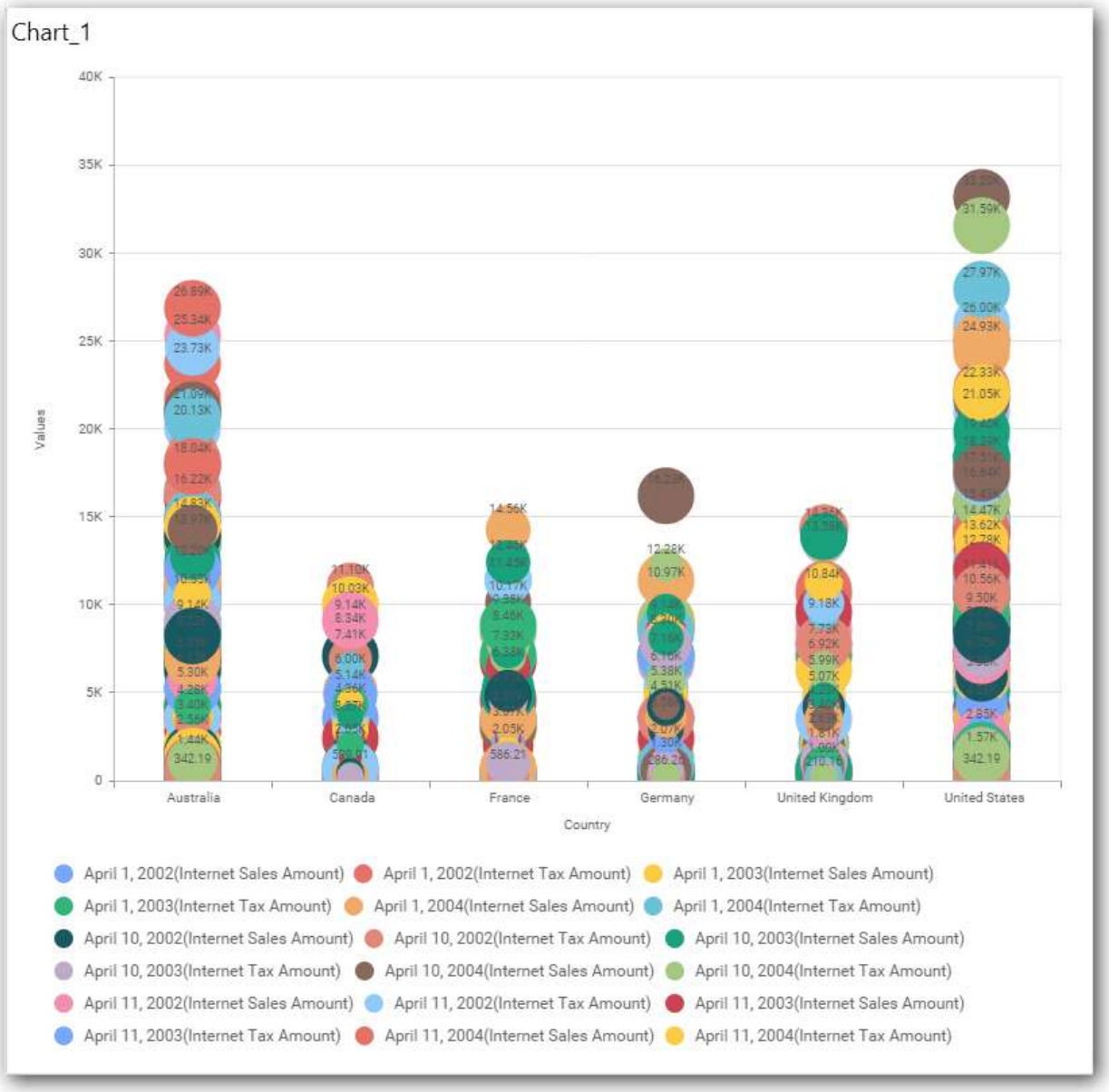


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart.



The chart will be rendered in series as shown in the image.

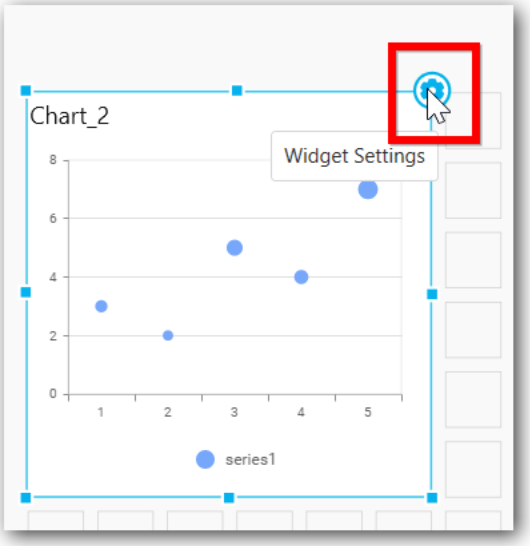


How to format Bubble Chart?

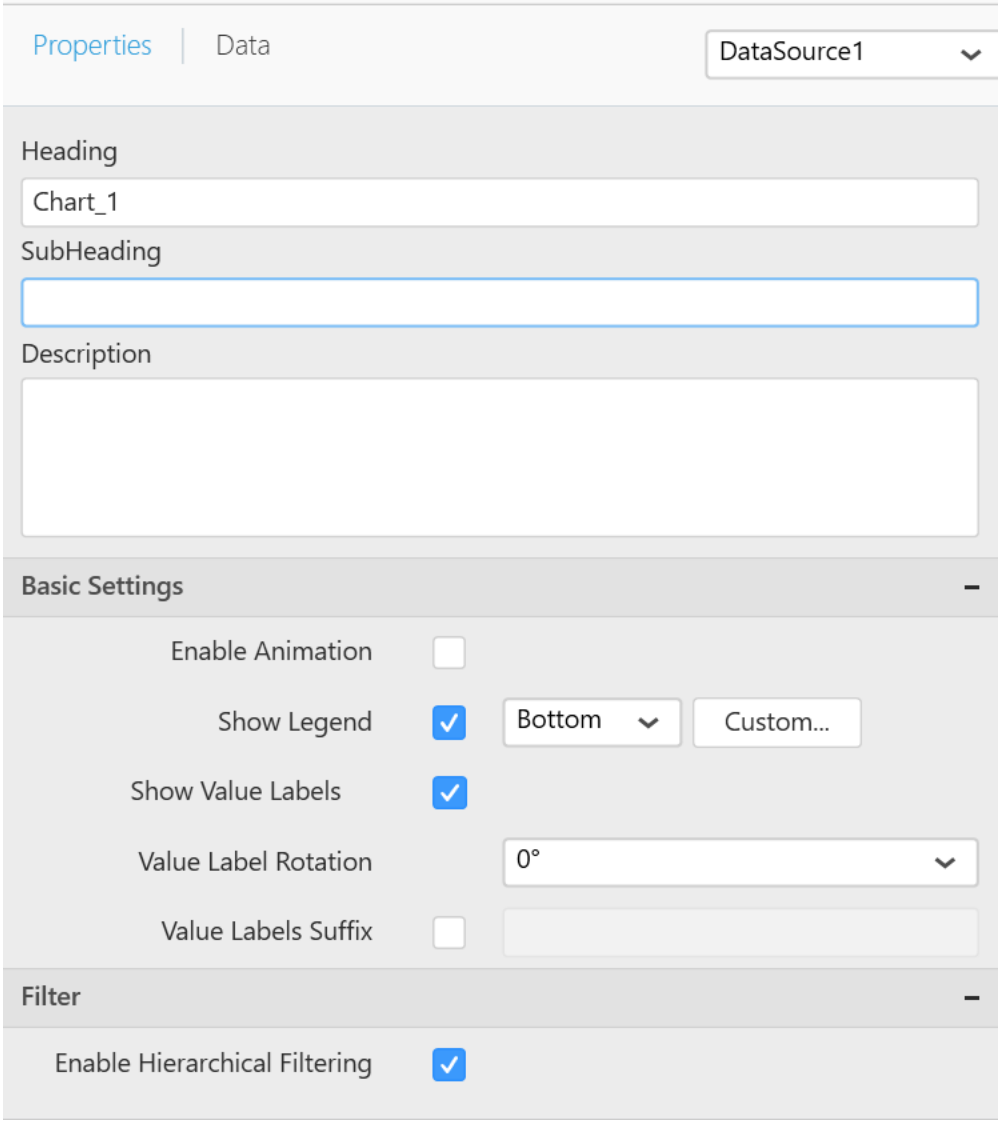
You can format the Bubble chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into bubble chart follow the steps

1. Drag and drop the bubble chart into canvas and resize it to your required size.
2. Configure the data into bubble chart.
3. Focus on the bubble chart and Click on Widget Settings.

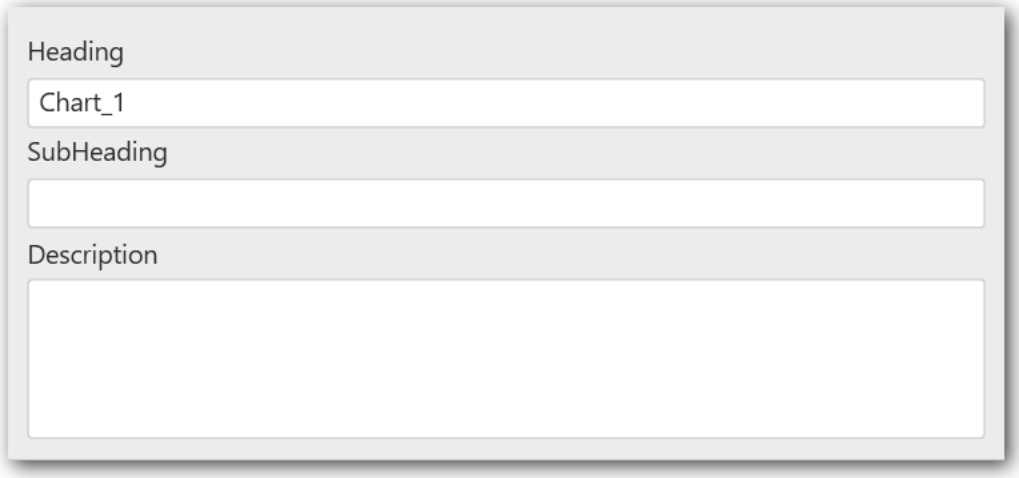


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this bubble chart widget.

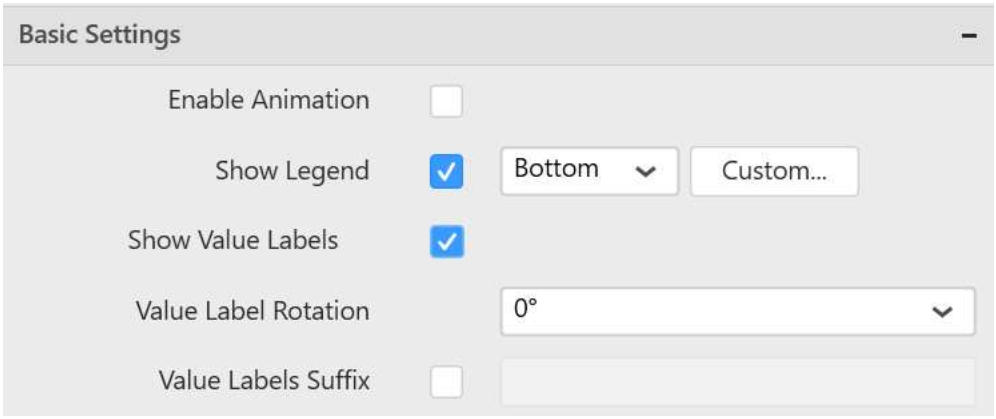
SubHeading

This allows you to set sub-title for this Bubble chart widget.

Description

This allows you to set description for this bubble chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

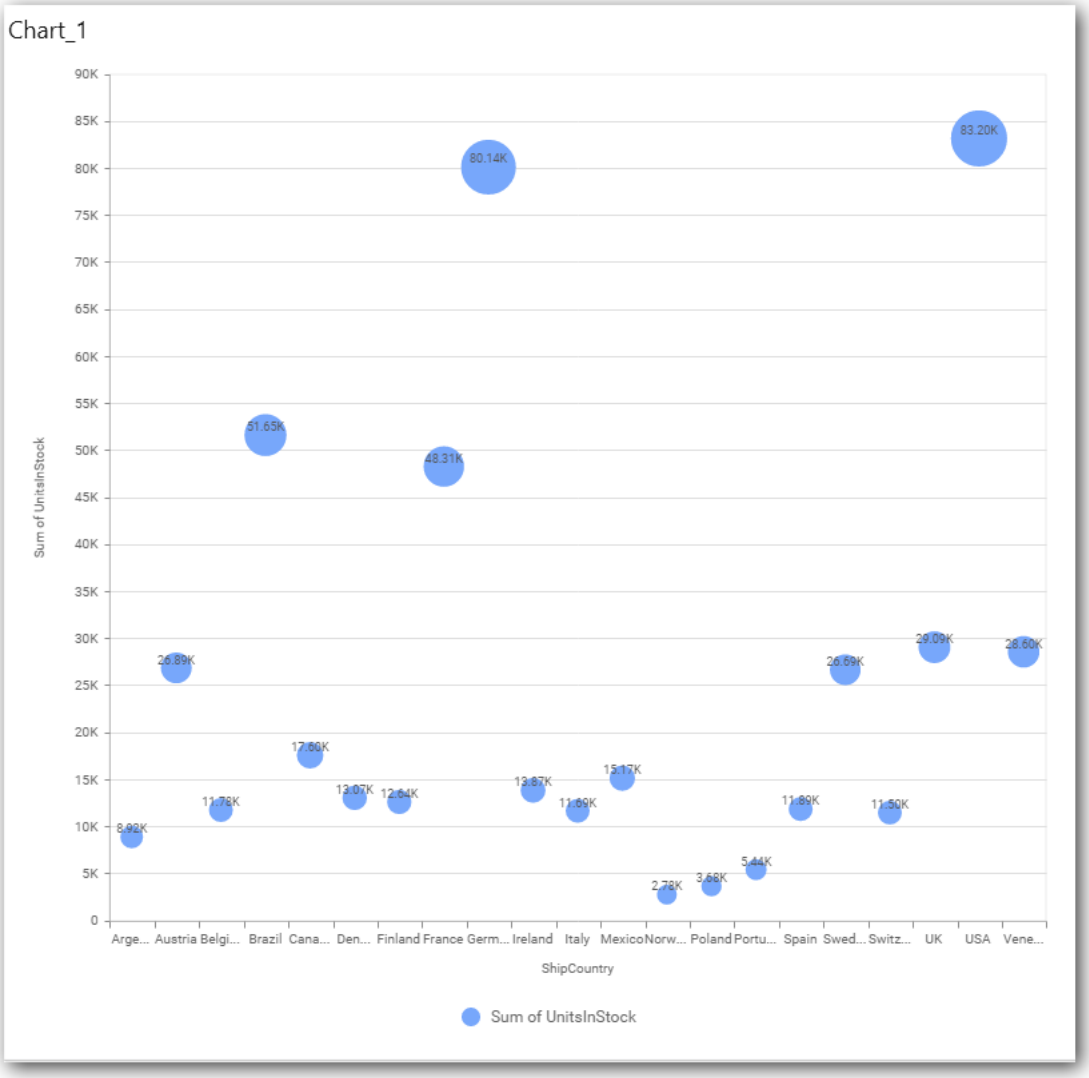


Enable Animation

This allows you to enable the rendering of series in animated mode.

Show Legend

This allows you to toggle the visibility of legend in chart and also changing the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Custom Legend Settings

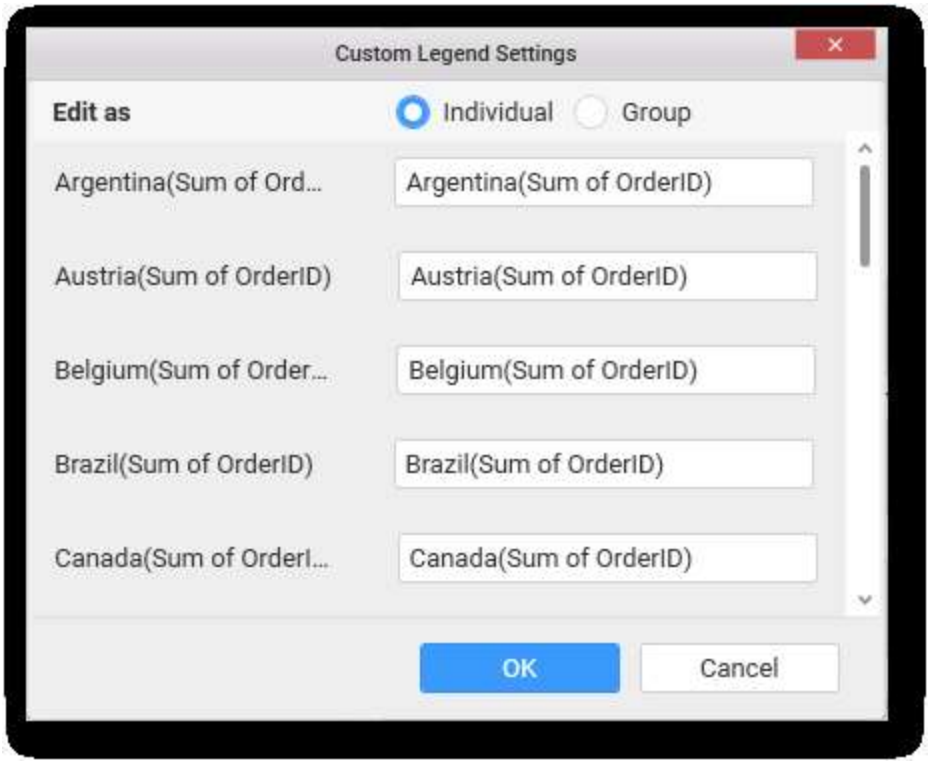
You can customize the legend text through the Custom Legend Settings dialog. This dialog will show the legend text list as labels at left and corresponding text area at right to add the formatted text to display instead. When a column is added into Row section, this dialog will show two options Individual and Group at top in addition, to toggle between.

Individual

Selecting Individual option will allow you to define a custom text (through the text area) to display for each legend series in chart with the default format:

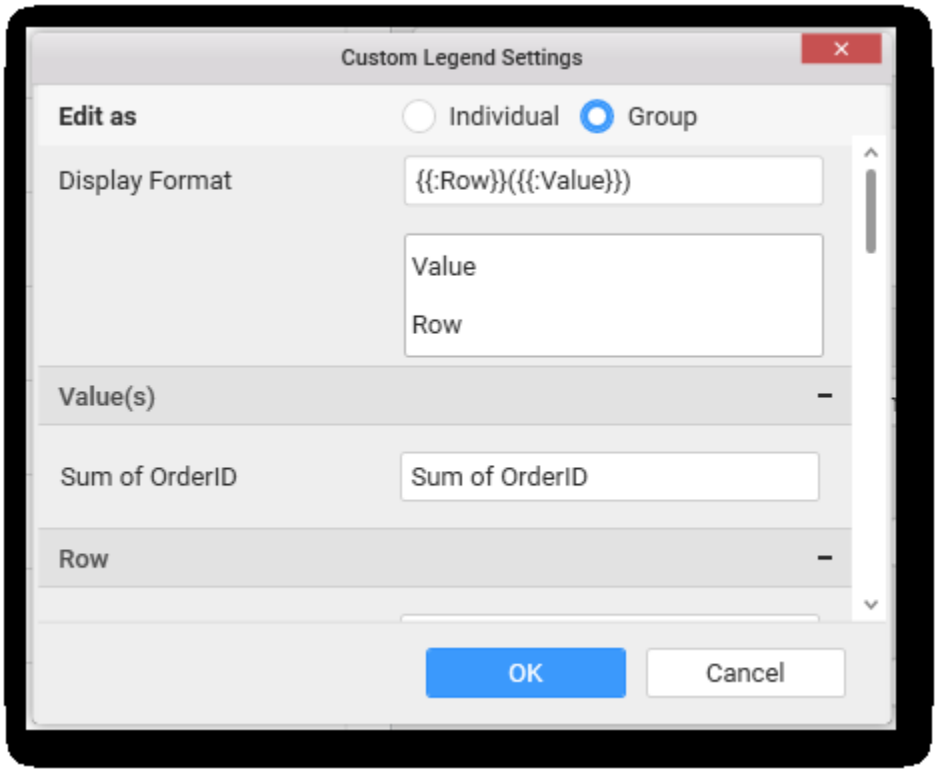
{{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}})

Where, Row represents the value of dimension column added to Row section and Value represents the value of the measure column added to Value section.

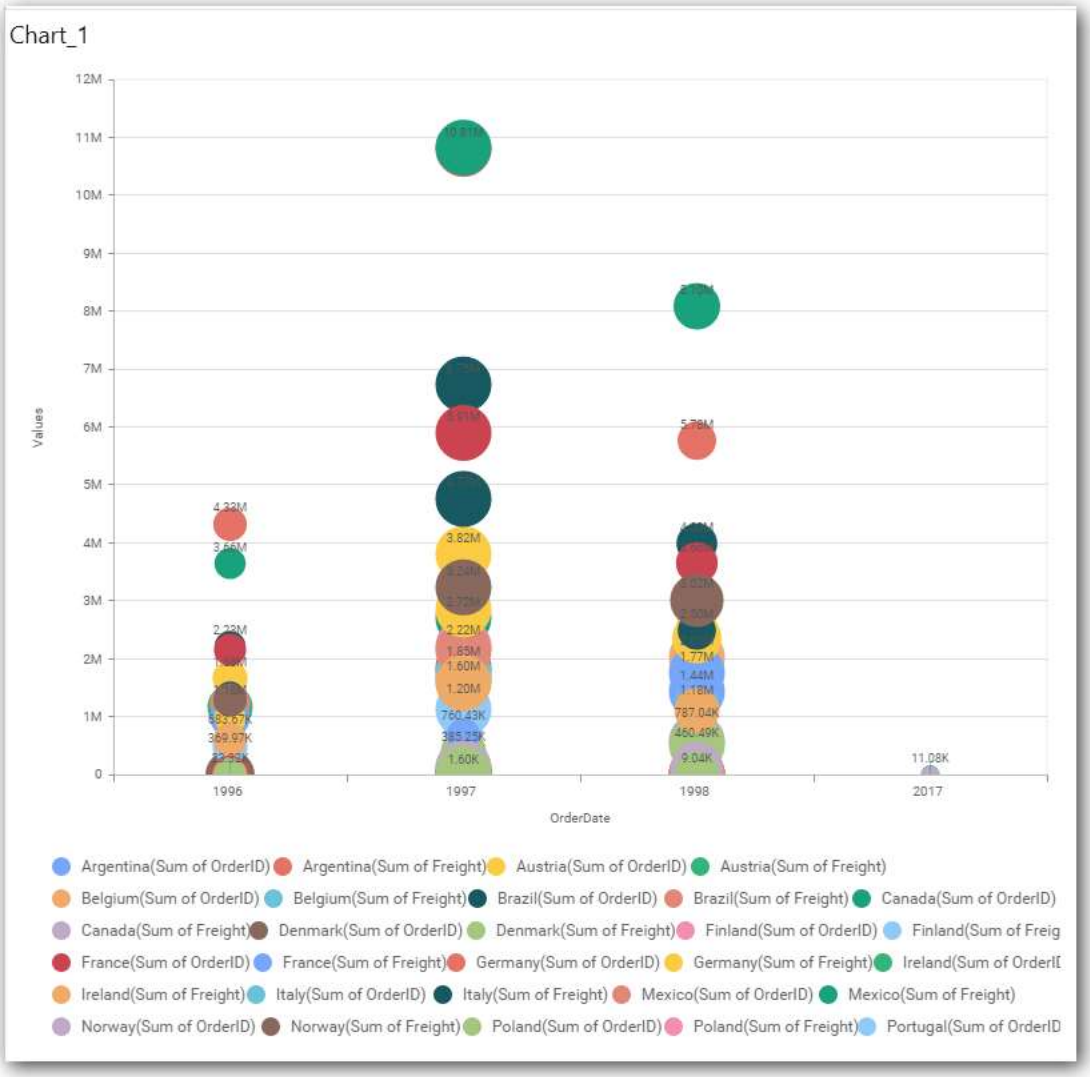


Group

Enabling Group option will allow you to set the display format and define a custom text (through the text area) to display for each legend series based on the specified format.

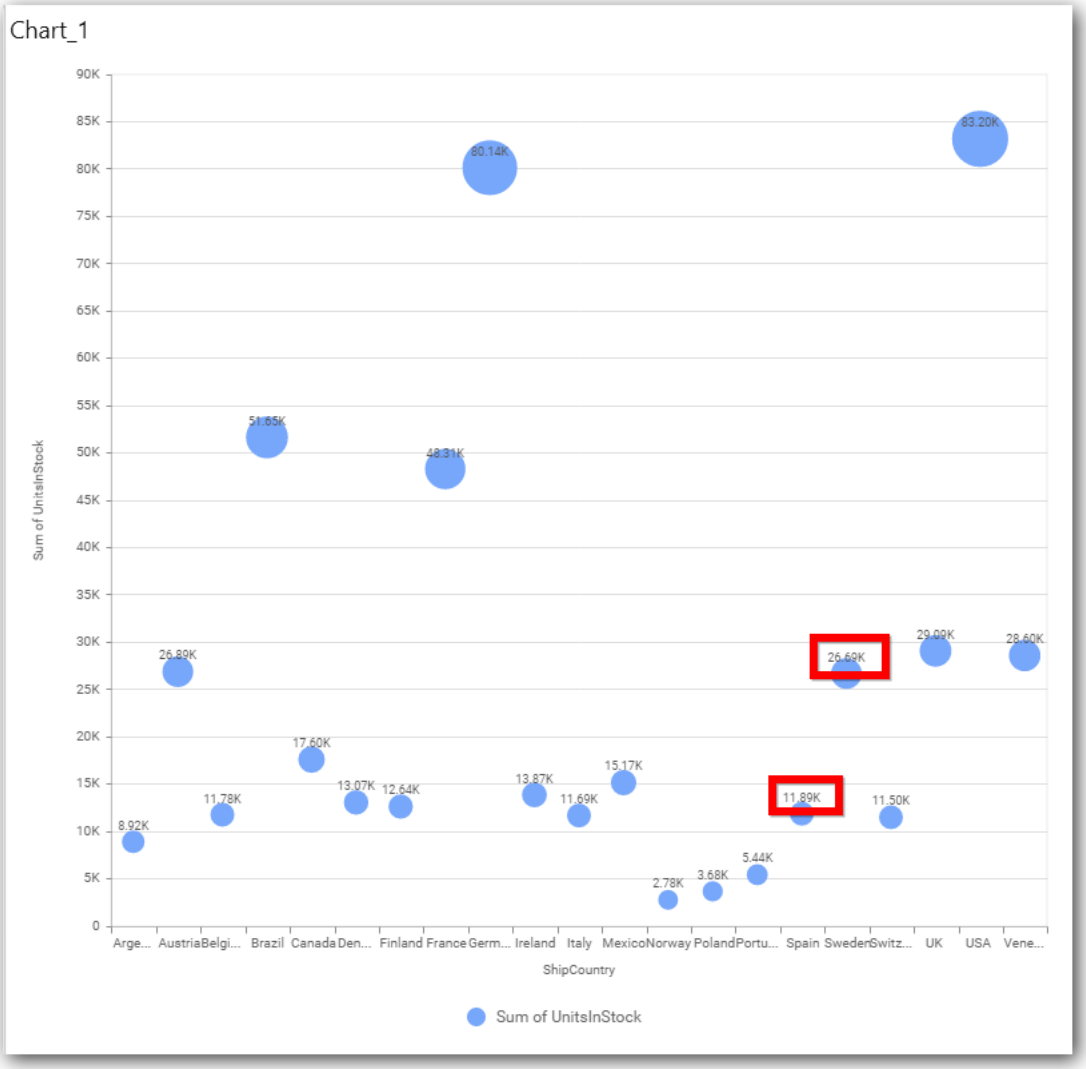


For example, If Display Format is {{"{{"}} : Row {{}}}} ({{"{{"}} : Value {{}}}}), then Legend series will display like Argentina (Sum of Order ID)



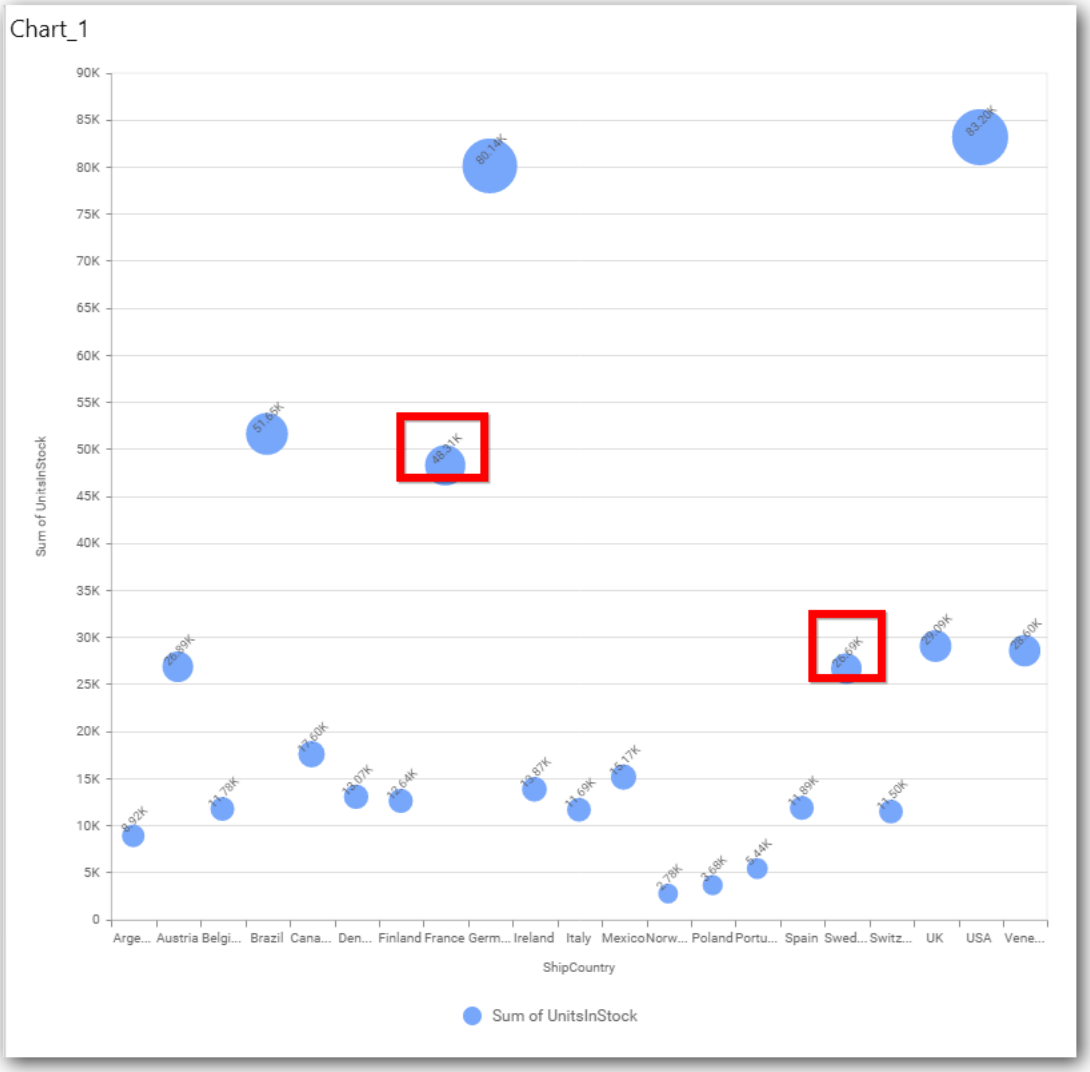
Show Value Labels

This allows you to toggle the visibility of value labels.



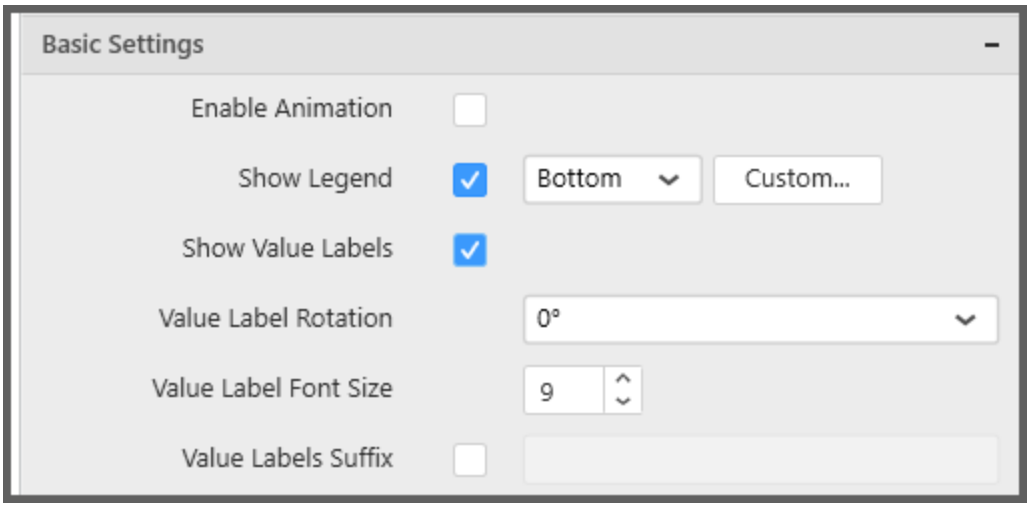
Value Label Rotation

This allows you to define the rotation angle for the value labels to display.



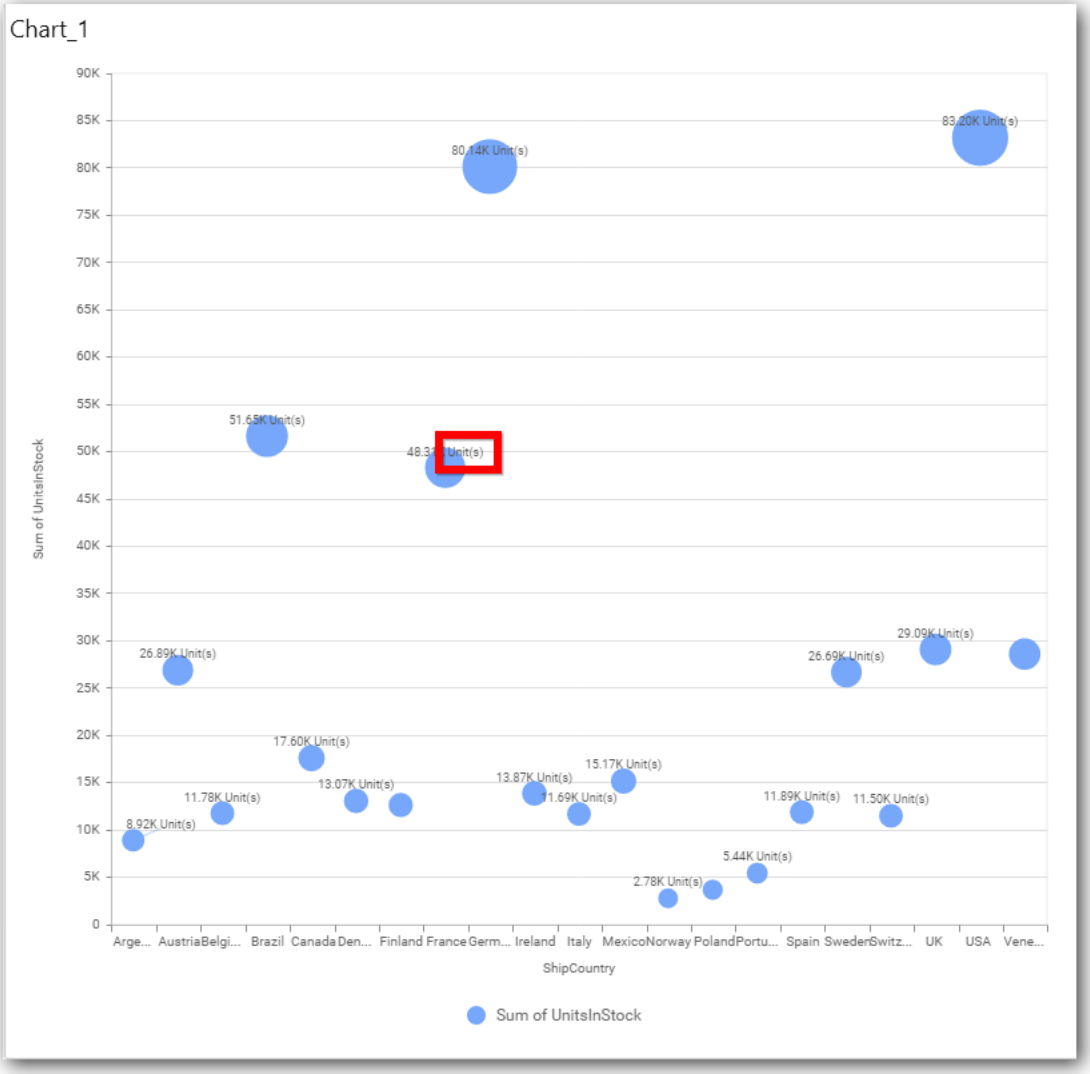
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.

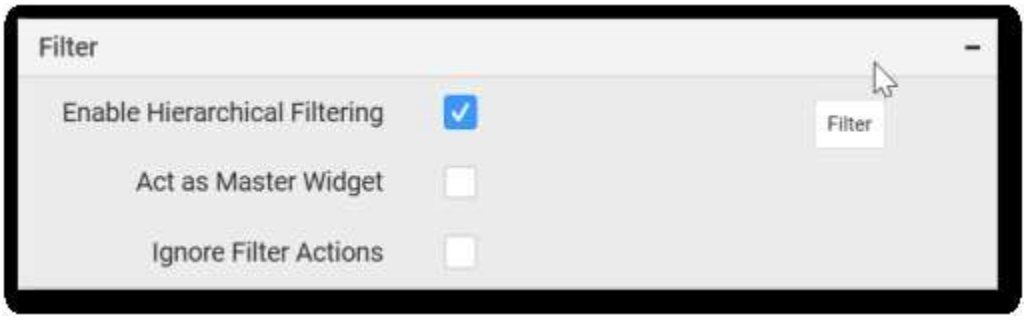


Value Labels Suffix

Allows you to set suffix to the value labels.



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

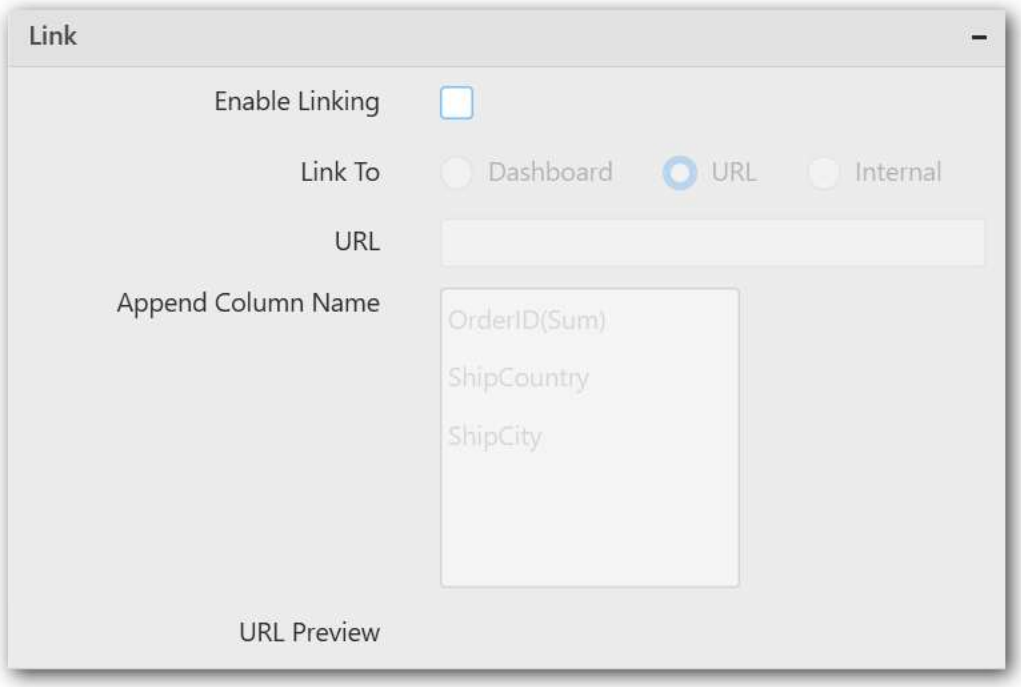
Act as Master Widget

This allows you to define this bubble chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

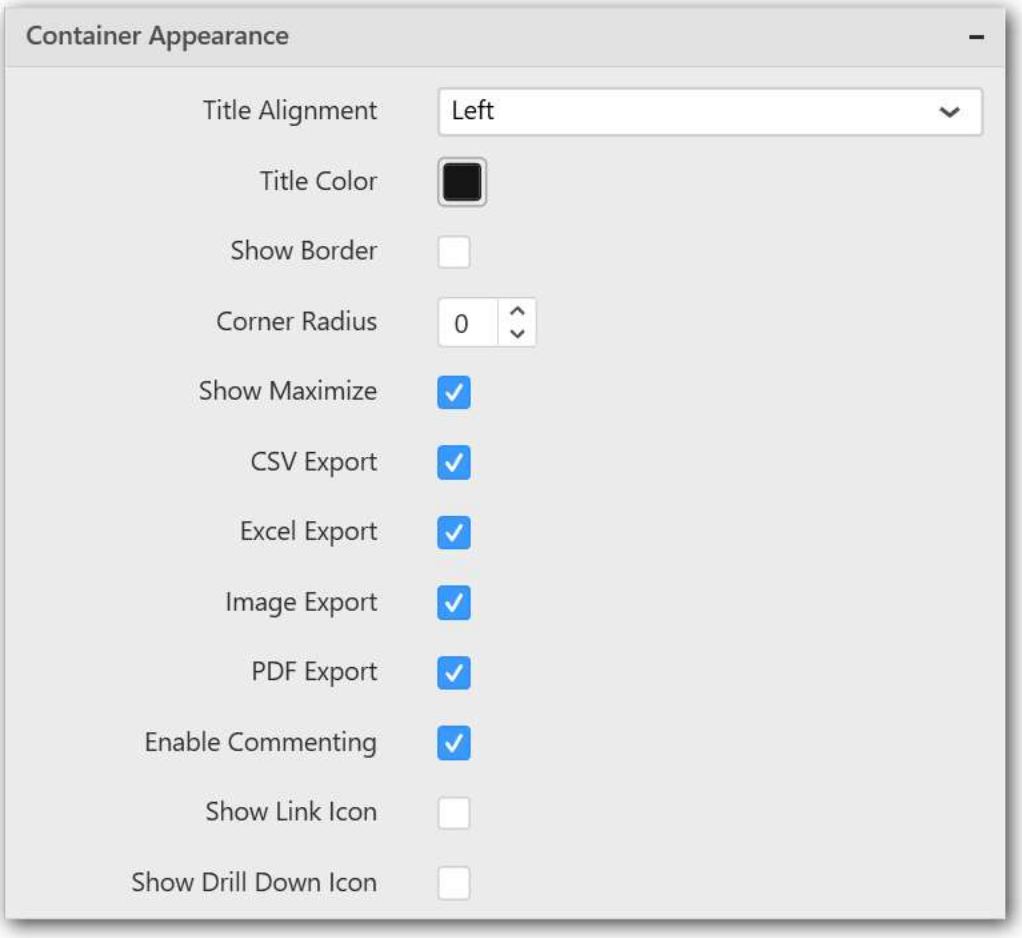
This allows you to define this bubble chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this bubble chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this bubble chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this bubble chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

Image Export

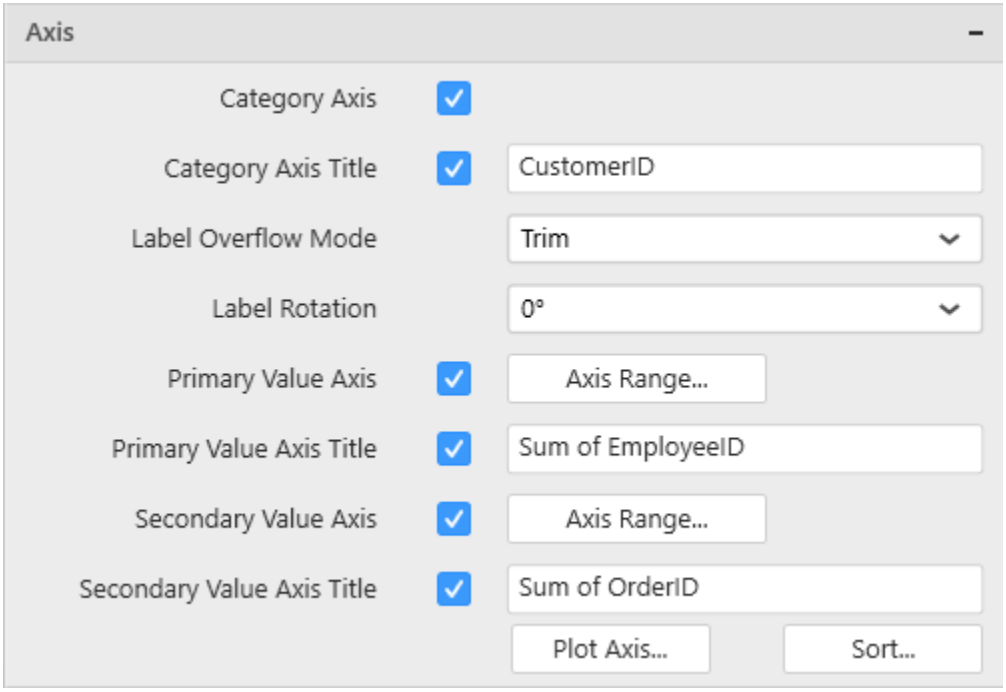
This allows you to enable/disable the image export option for this bubble chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

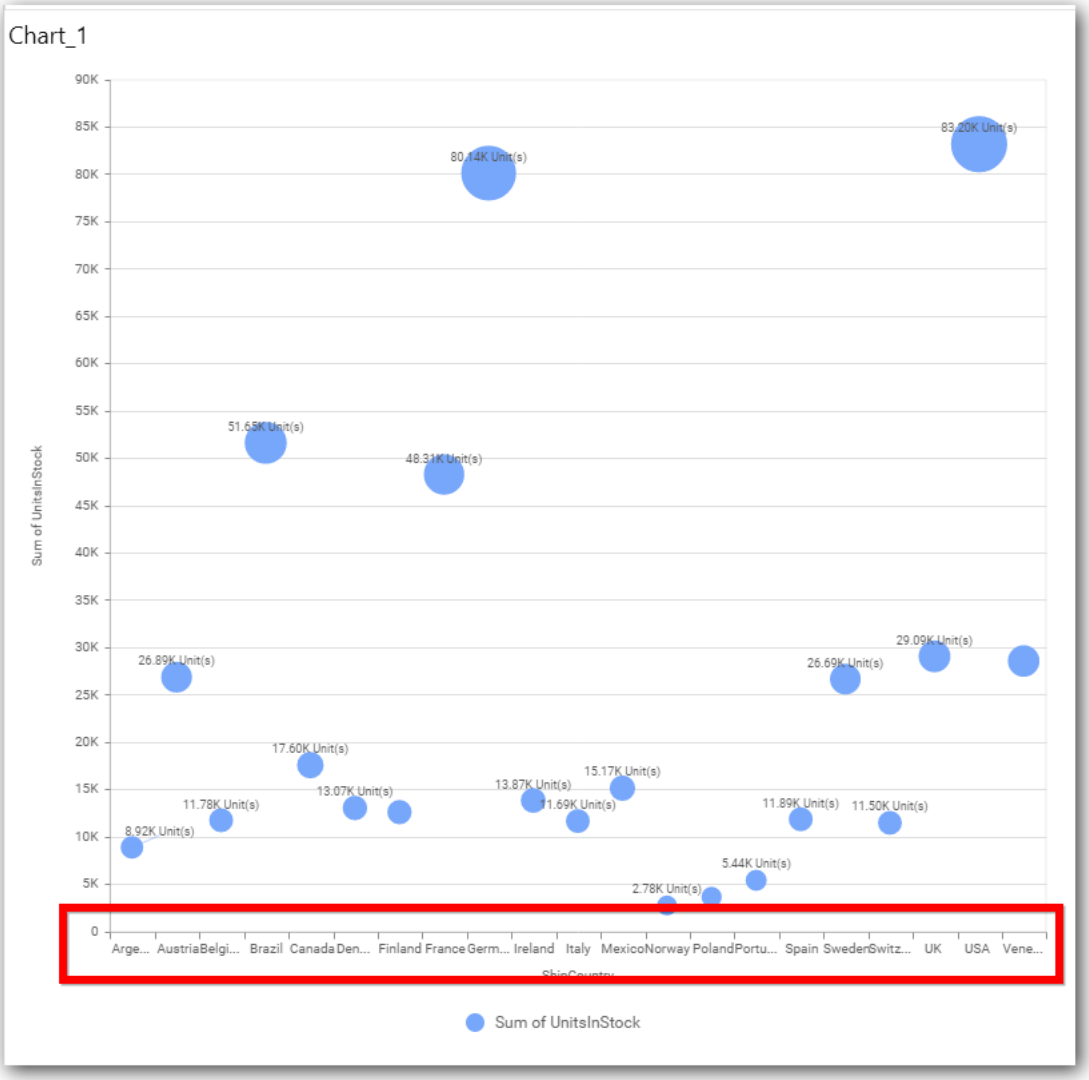
Axis Settings

This section allows you to customize the axis settings in chart.



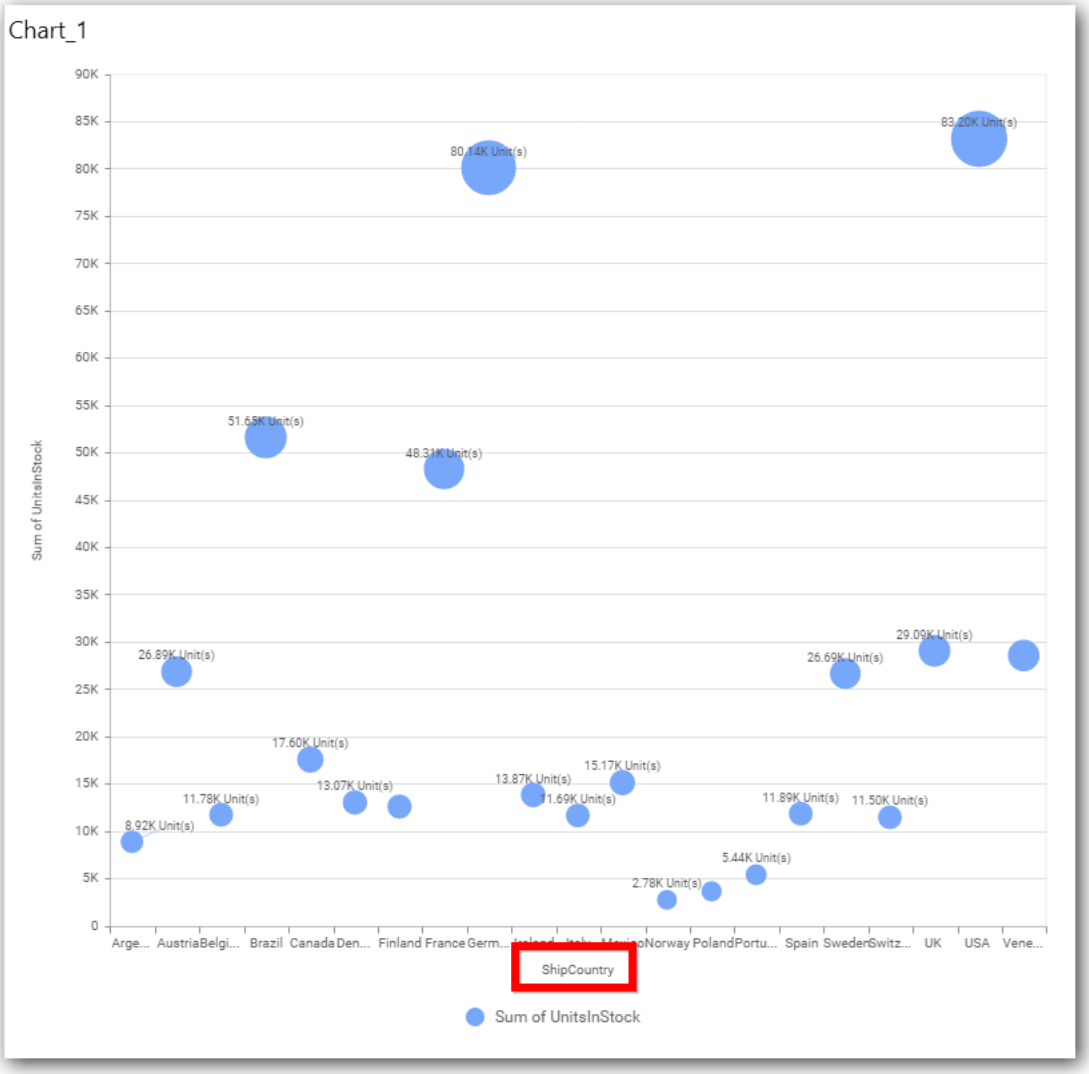
Category Axis

This allows to enable/edit the option of Category Axis. It will reflect in chart area x-axis name.



Category Axis Title

This allows you to toggle the visibility of Category axis title.

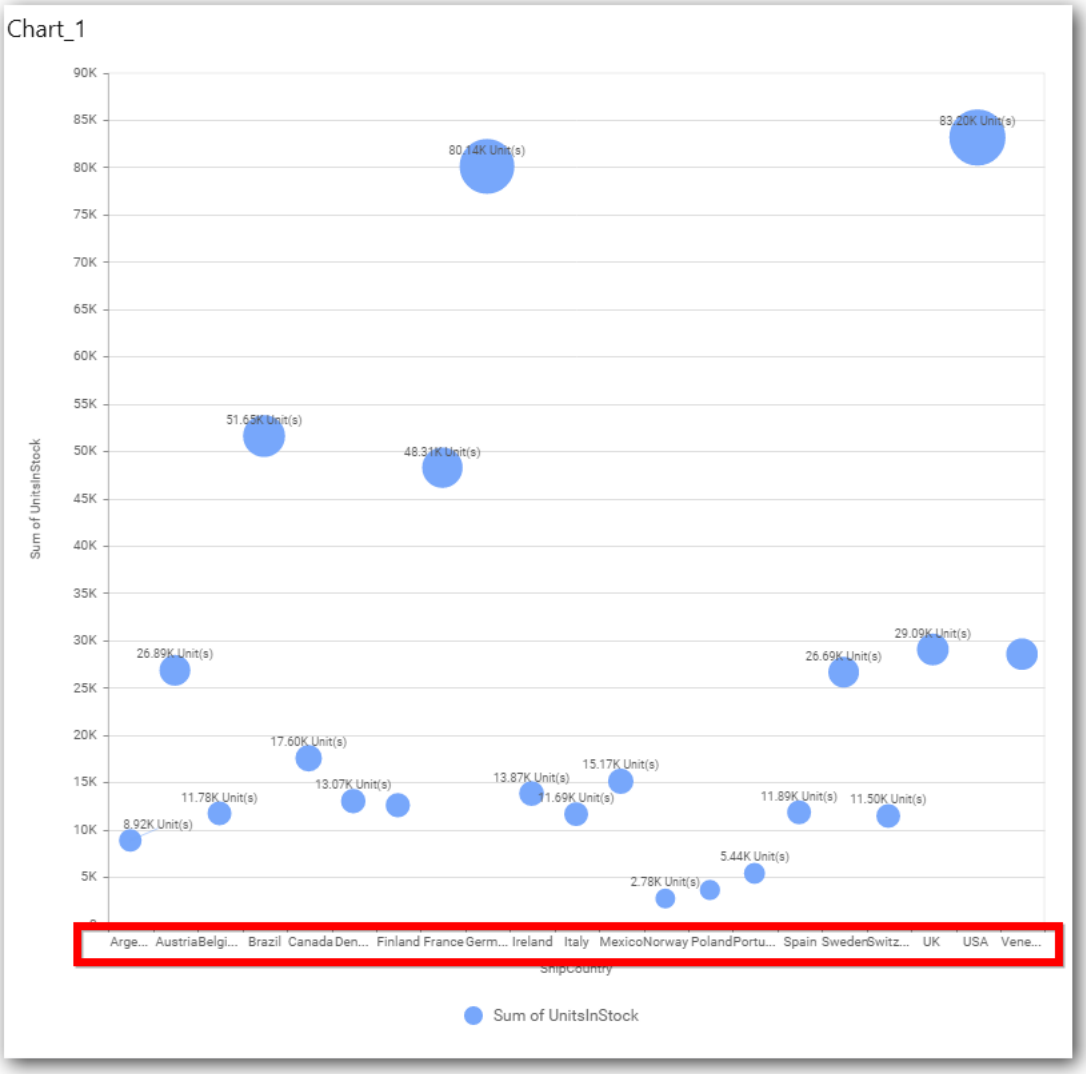


Label overflow mode

This allows you to handle the display mode of the overlapping labels.

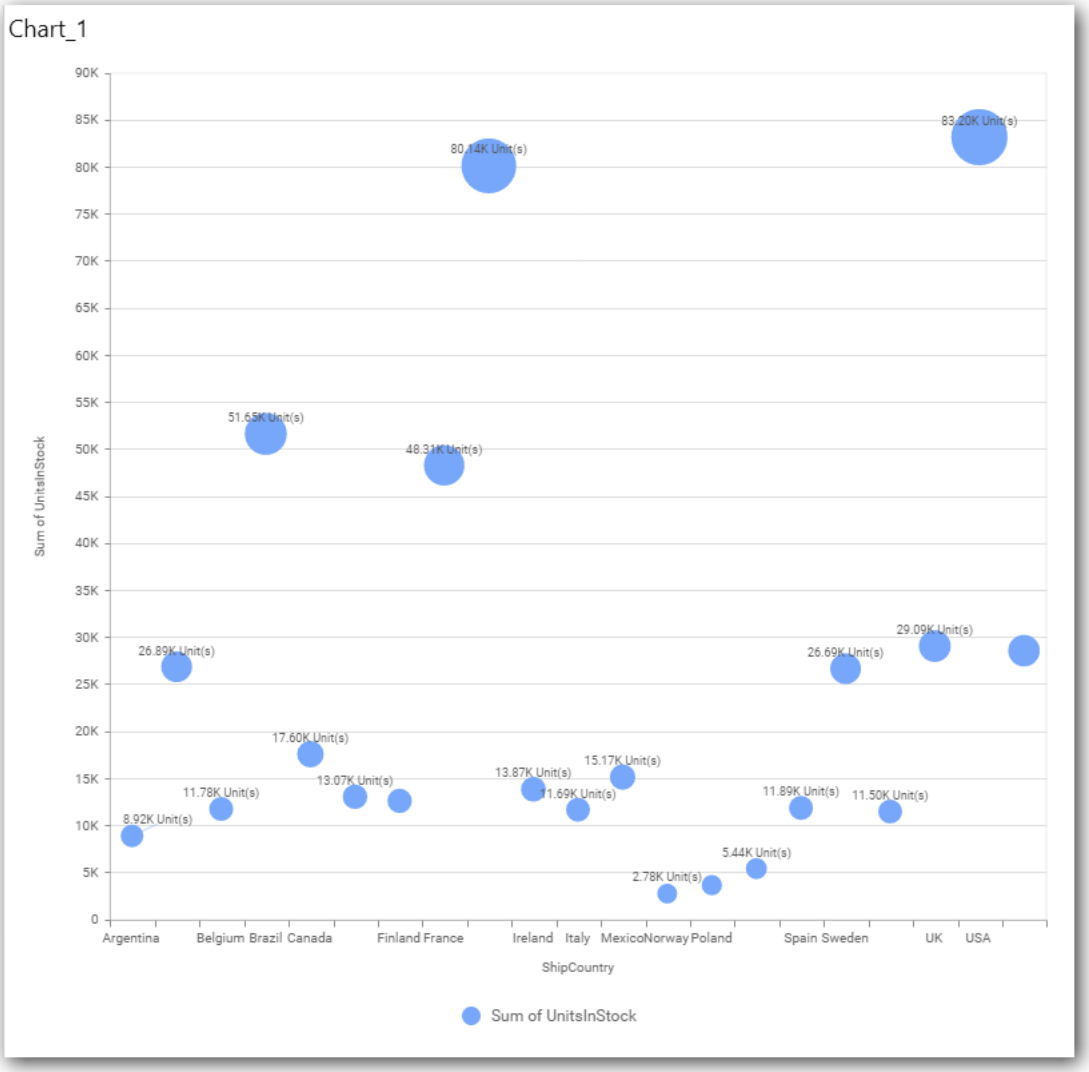
Trim

This option trims the end of overlapping label in the axis.



Hide

This option hides the overlapping label in the axis.

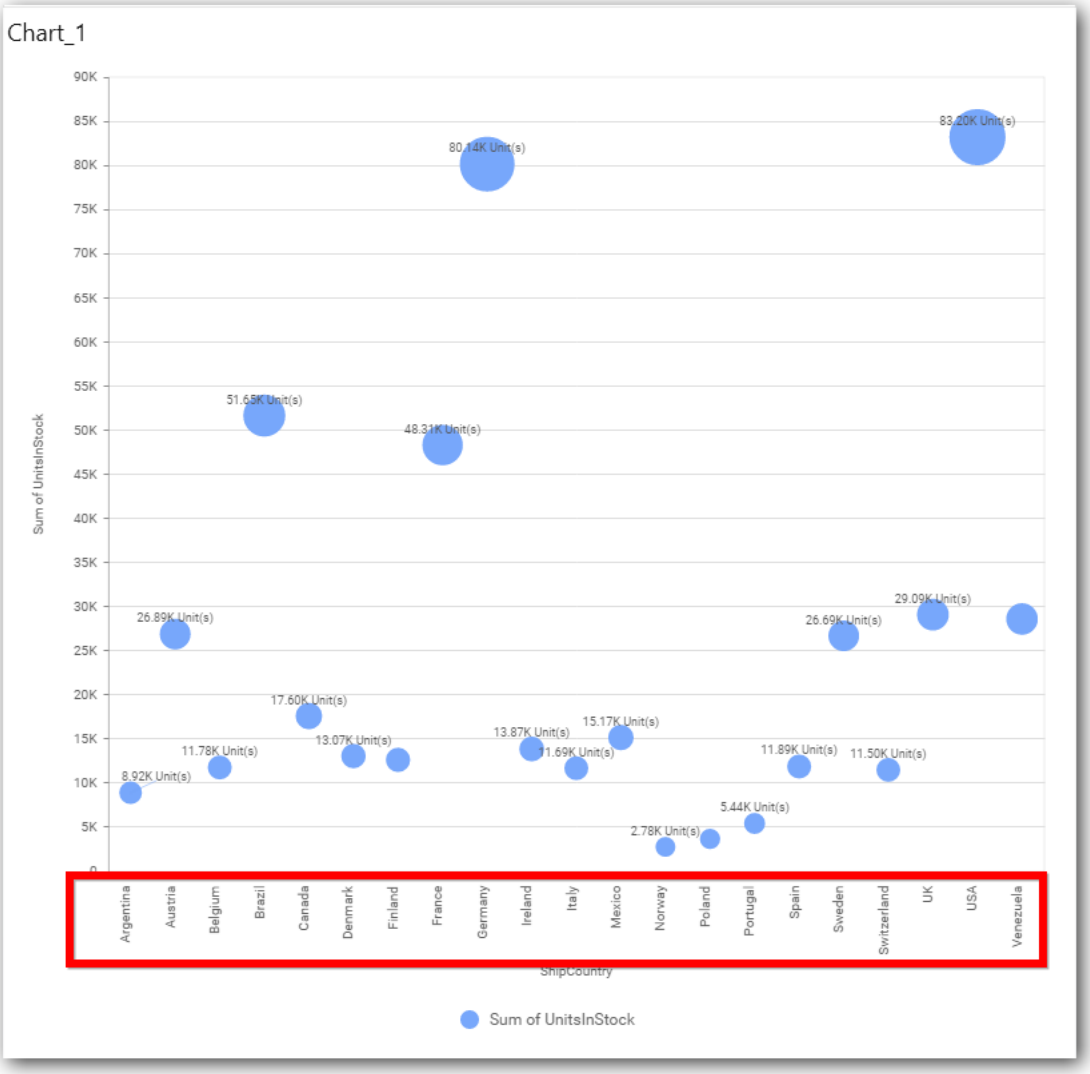


Wrap

This option wraps the lengthy label text in the axis.

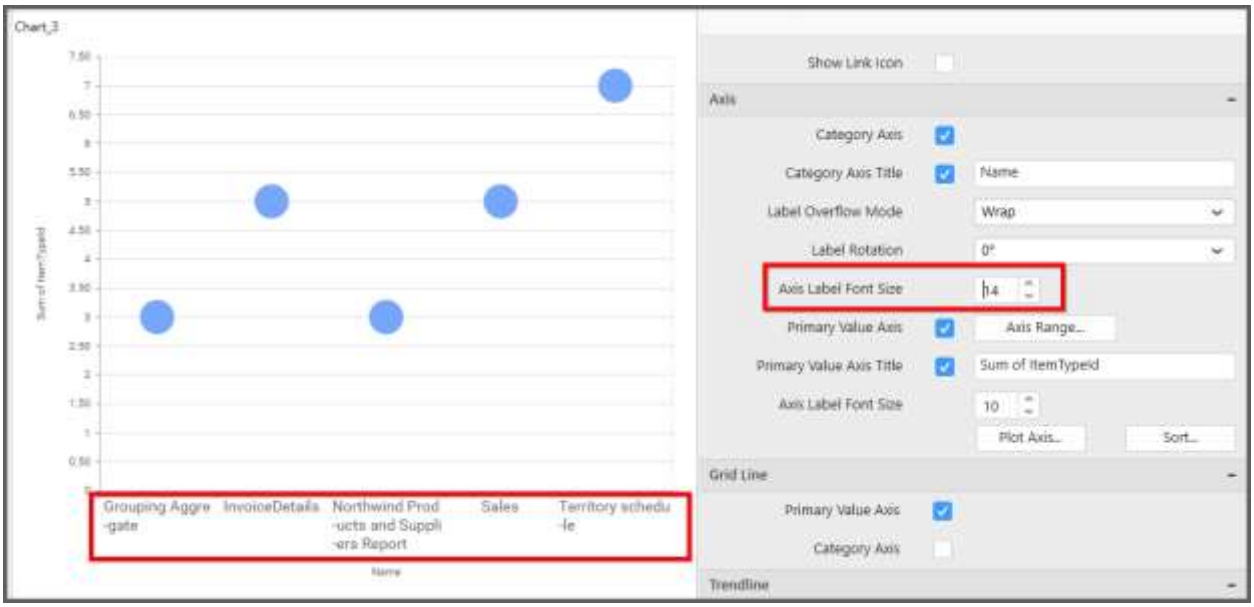
Label Rotation

This allows you to define the rotation angle for the category axis labels to display.



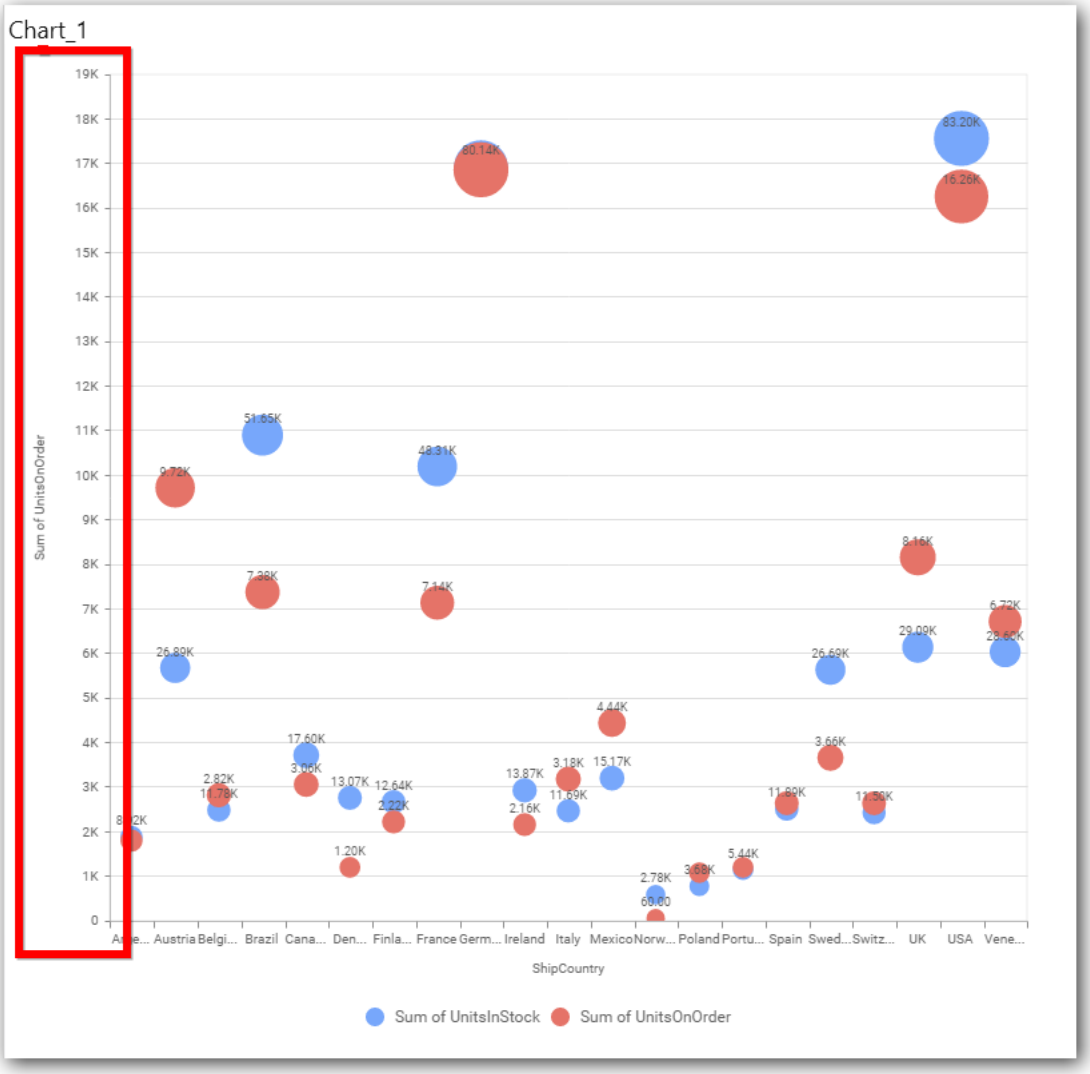
Axis Label Size

This allows you to increase or decrease the font size of the category axis label. Default font size for the category axis label was 10 pixels.



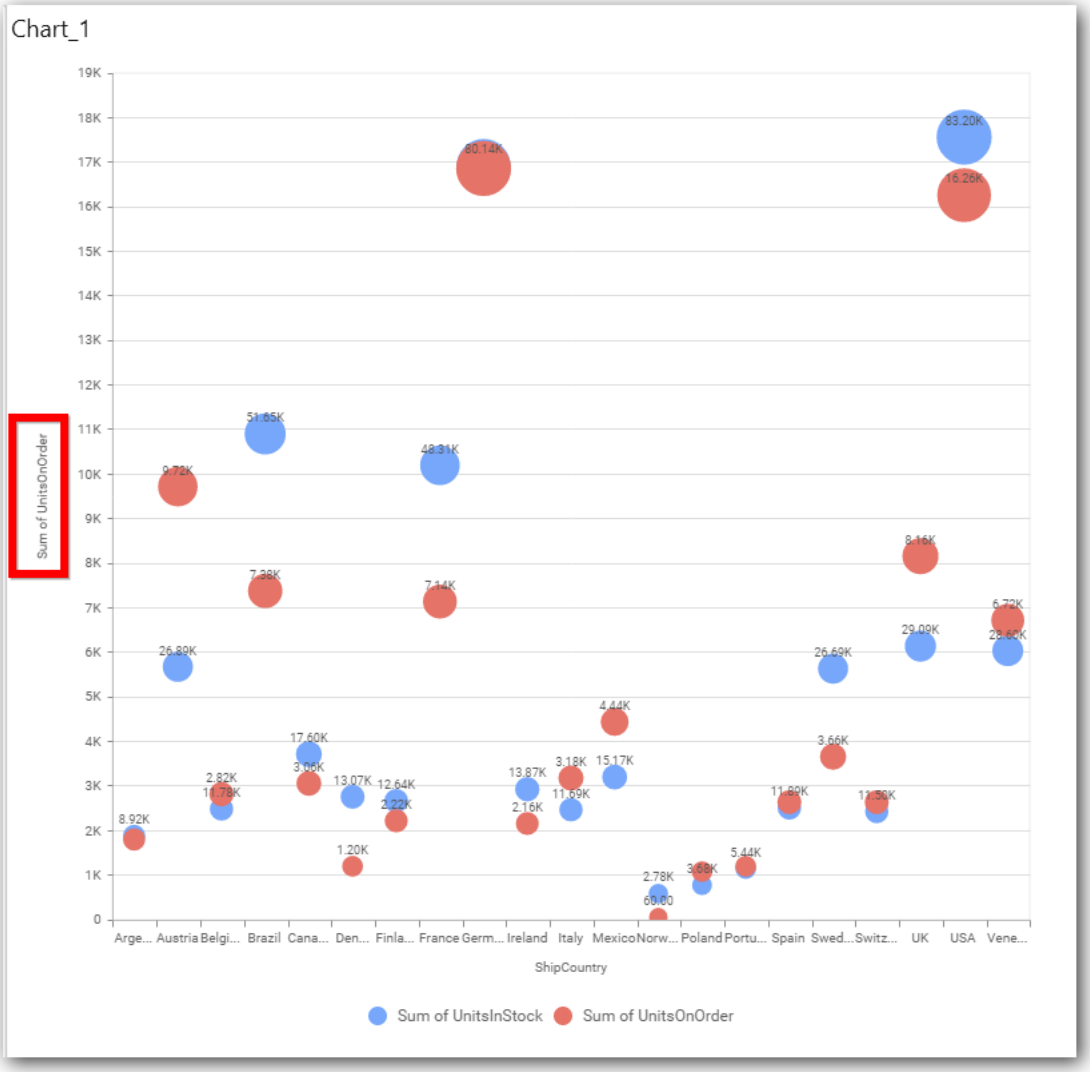
Primary Value Axis

This allows you to enable/edit the Primary Value Axis title. It will reflect in chart area y-axis name.



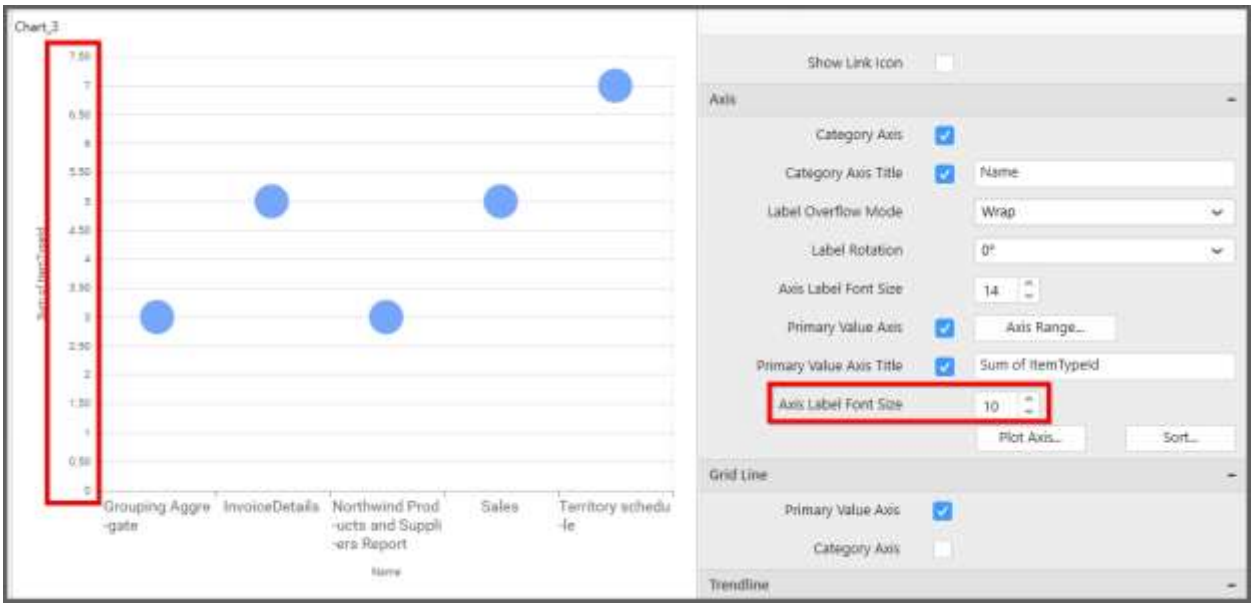
Primary Value Axis Title

This allows you to toggle the visibility of primary value axis title.



Axis Label Size

This allows you to increase or decrease the font size of the primary axis label. Default font size for the primary axis label was 10 pixels.

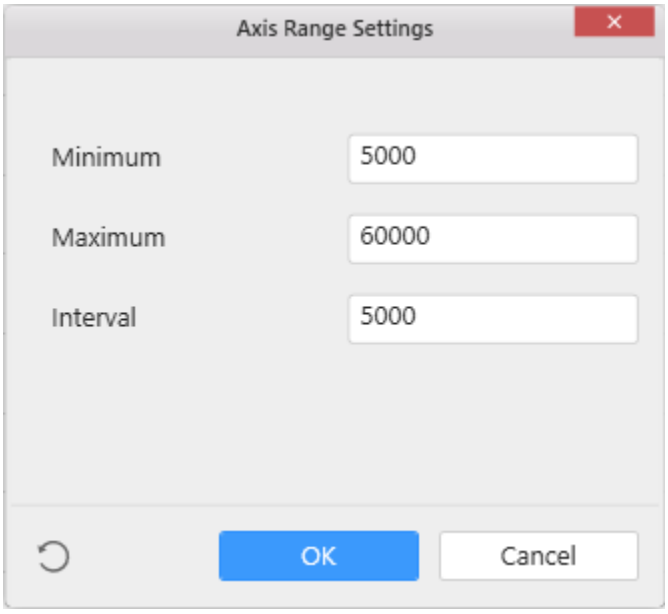


Primary Value Axis Range

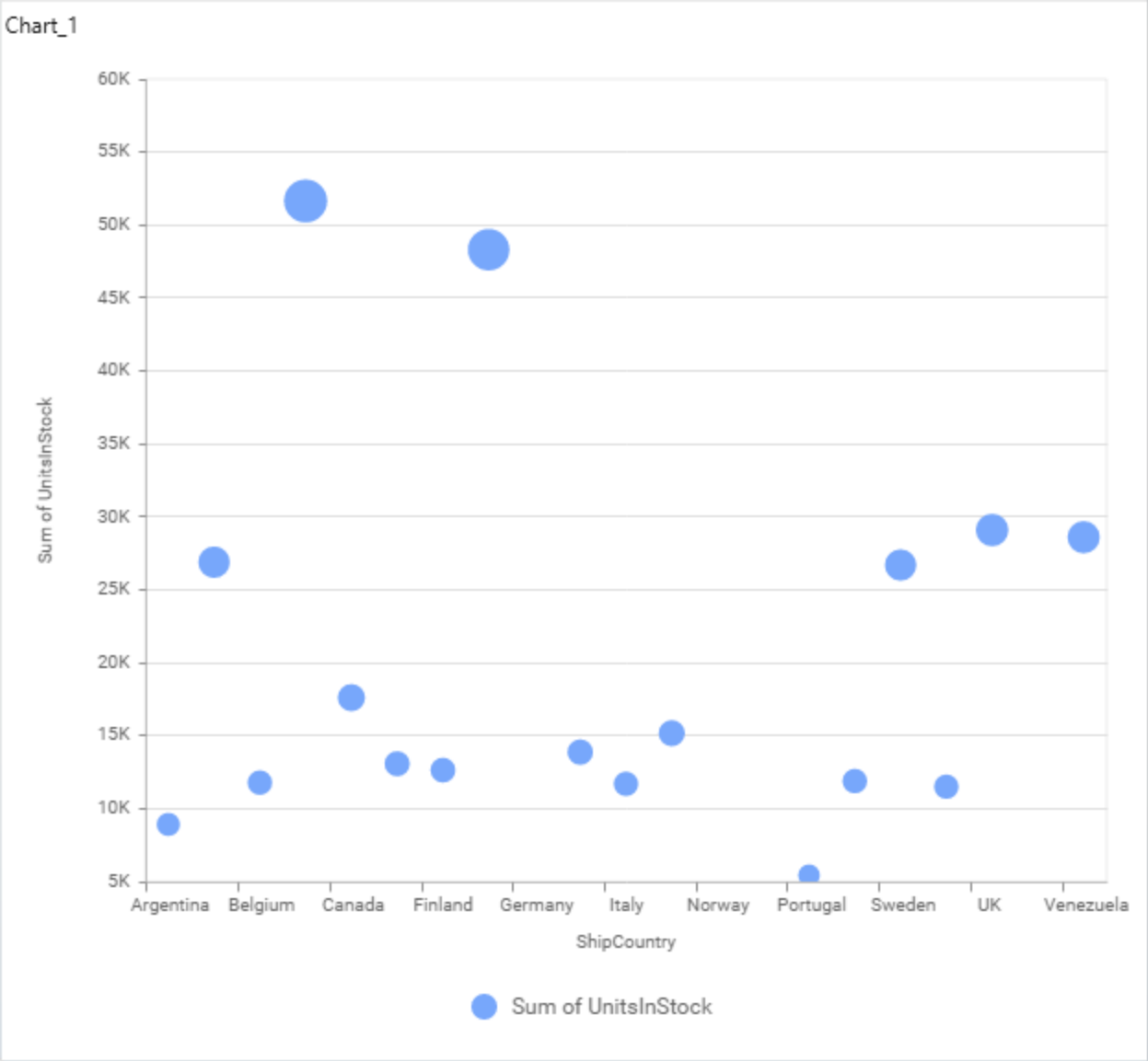
This allows you to set user-defined range with valid interval for primary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking the Axis Range button in primary axis property pane.

Axis Range Settings

You can customize the axis range values through Axis Range Settings dialog. This dialog will show the options to set minimum range, maximum range and interval values.

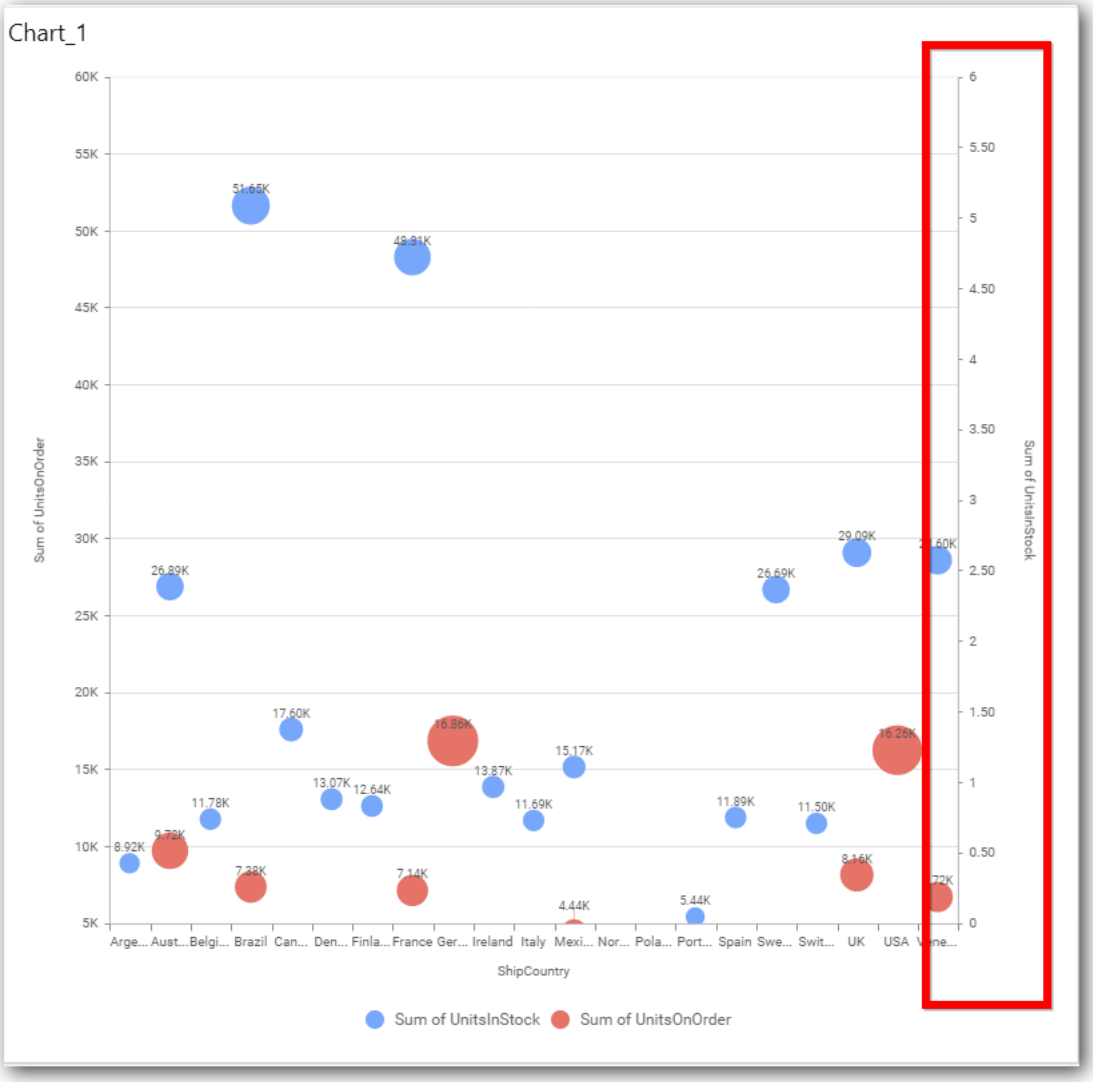


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



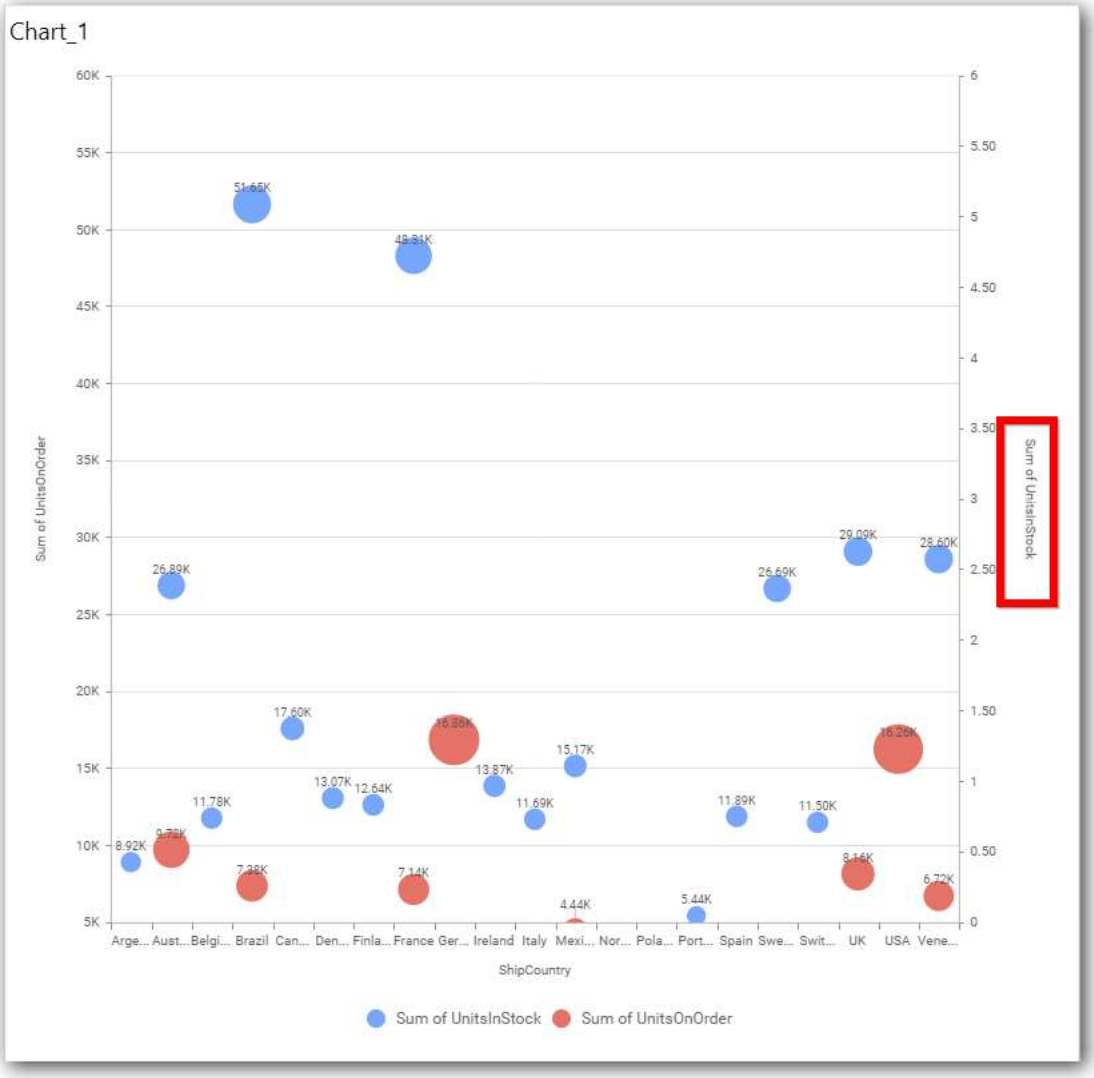
Secondary Value Axis

This allows you to enable/edit the Secondary Value Axis title. It will reflect in chart area secondary y-axis name.



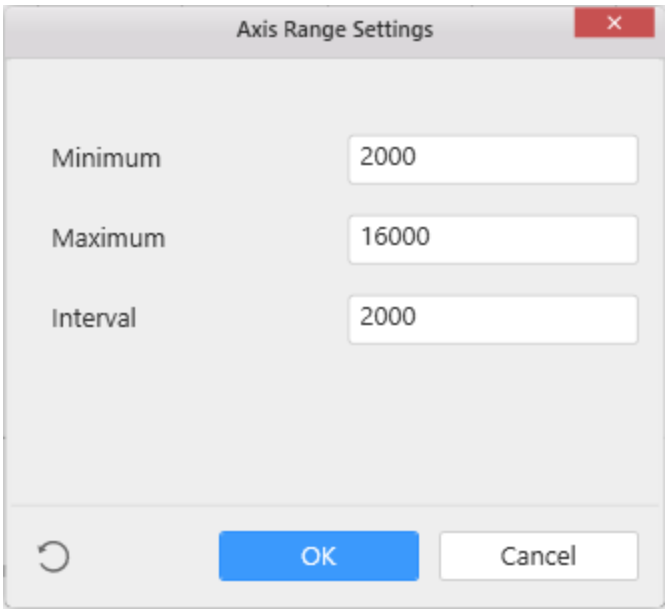
Secondary Value Axis Title

This allows you to toggle the visibility of secondary value axis title.

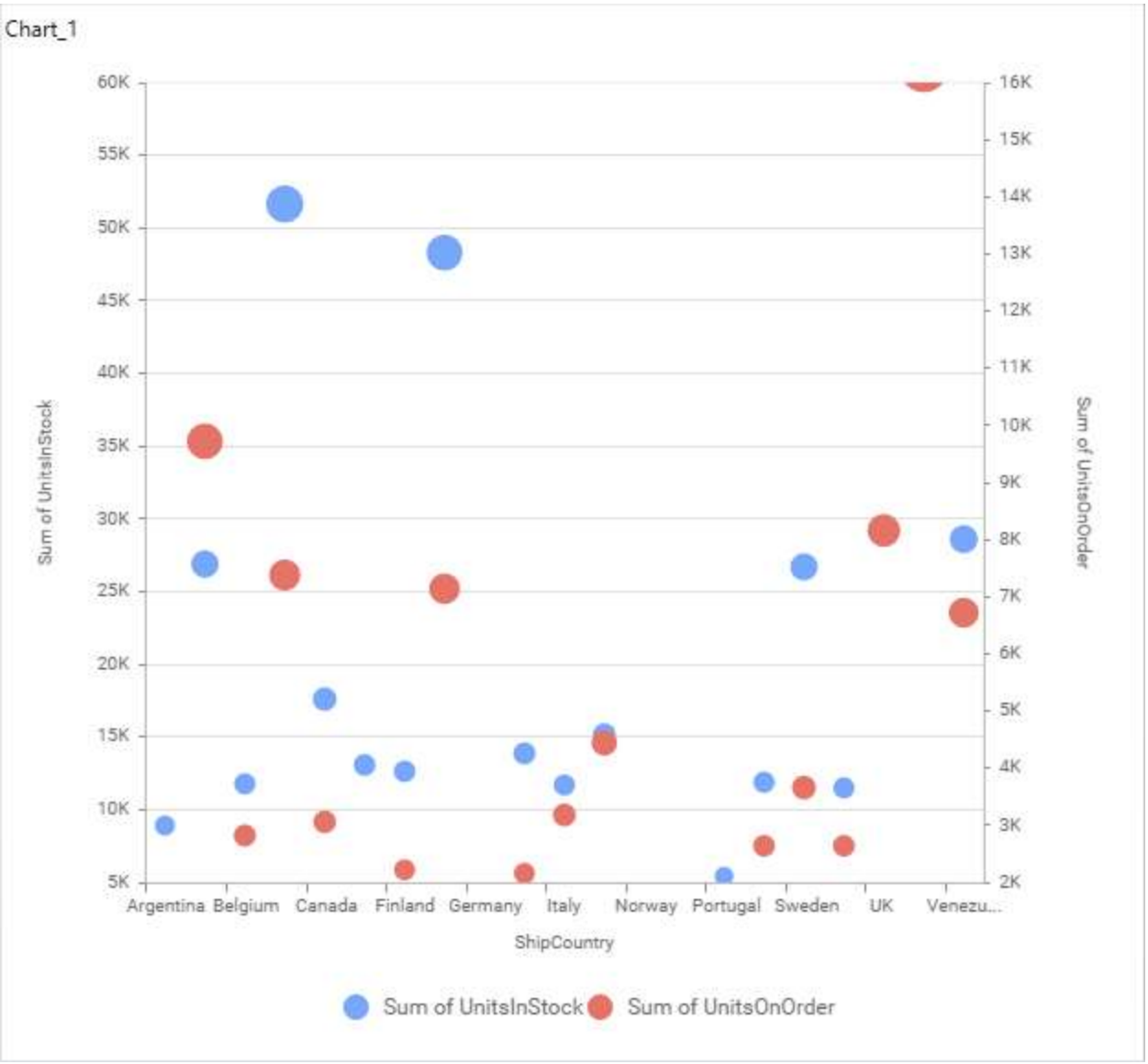


Secondary Value Axis Range

This allows you to set user-defined range with valid interval for secondary value axis. The axis range can be set from Axis Range Settings dialog and this can be opened on clicking Axis Range button in secondary axis property pane.

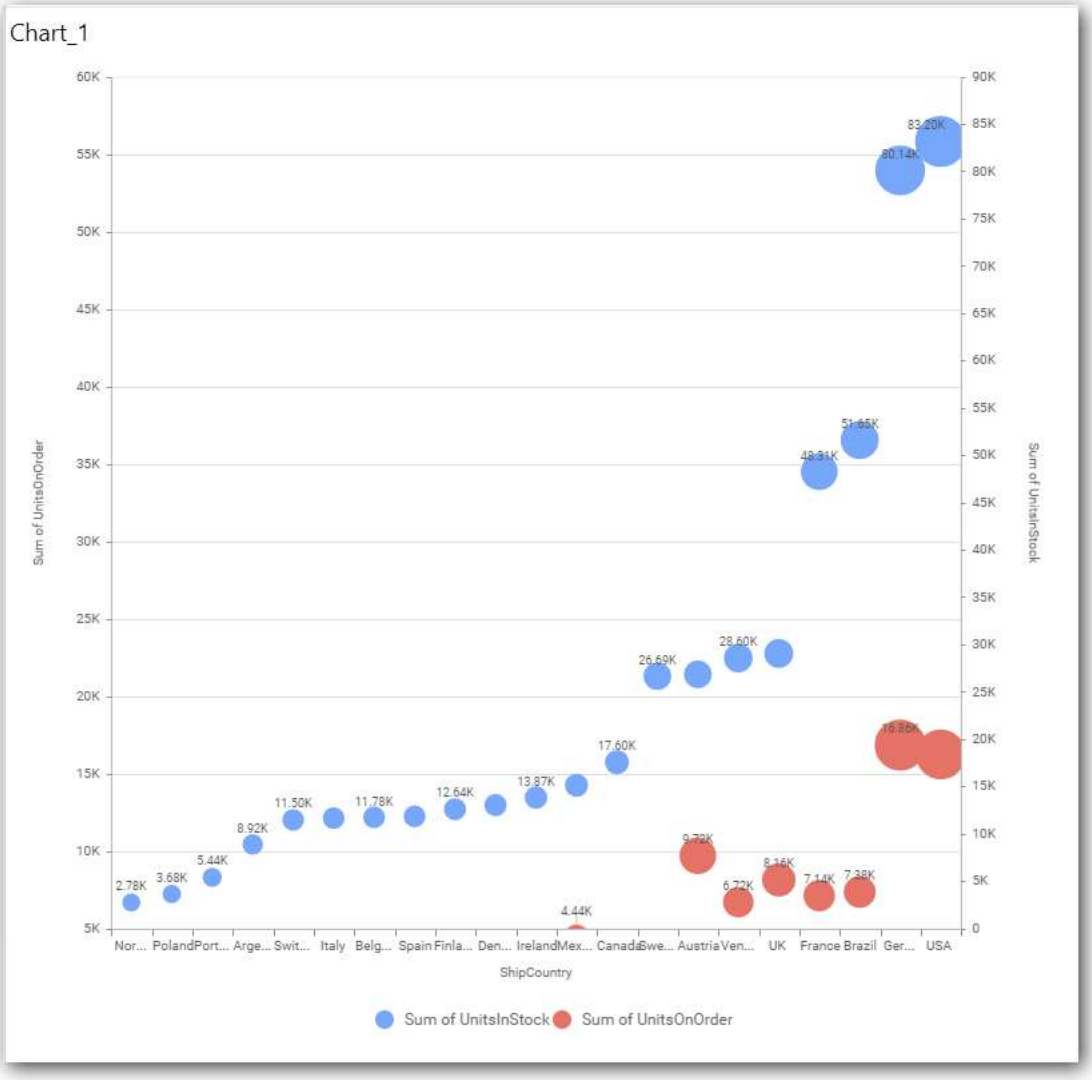


Below Chart demonstrate the Axis Range support, with the above shown min, max and interval values.



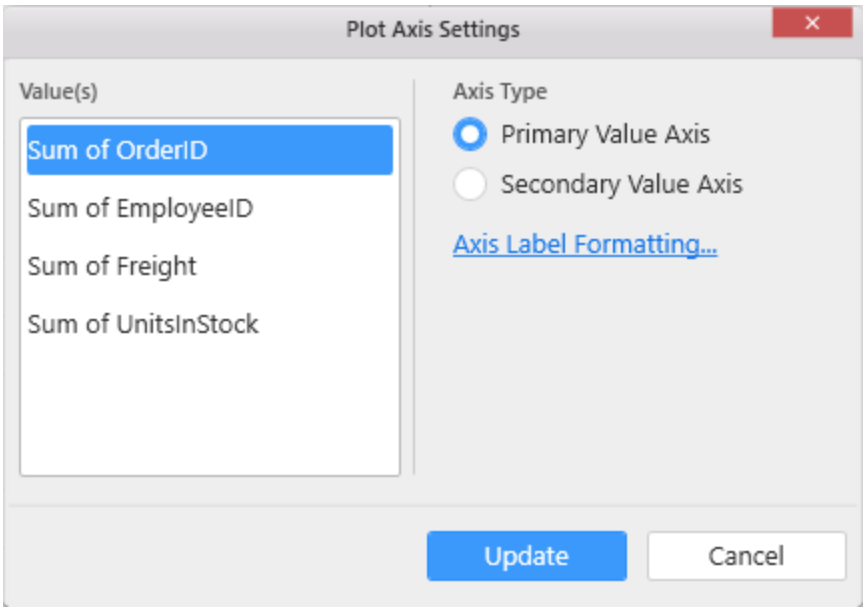
Sort Order

This allows you to define the sorting of chart based on any of the measures that you dropped and its order through this option. Following screenshot illustrates the Ascending sort order.

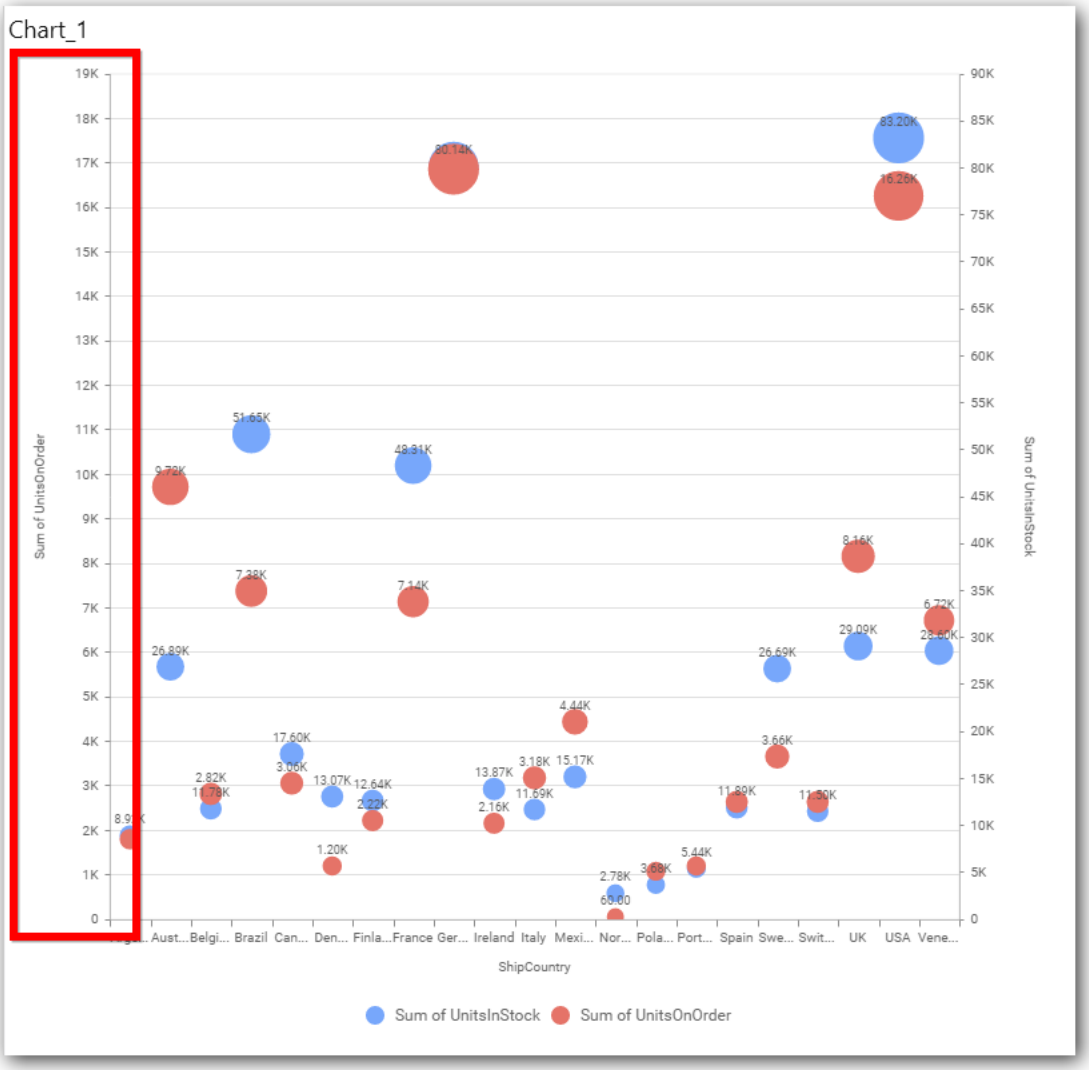


Plot Axis Settings

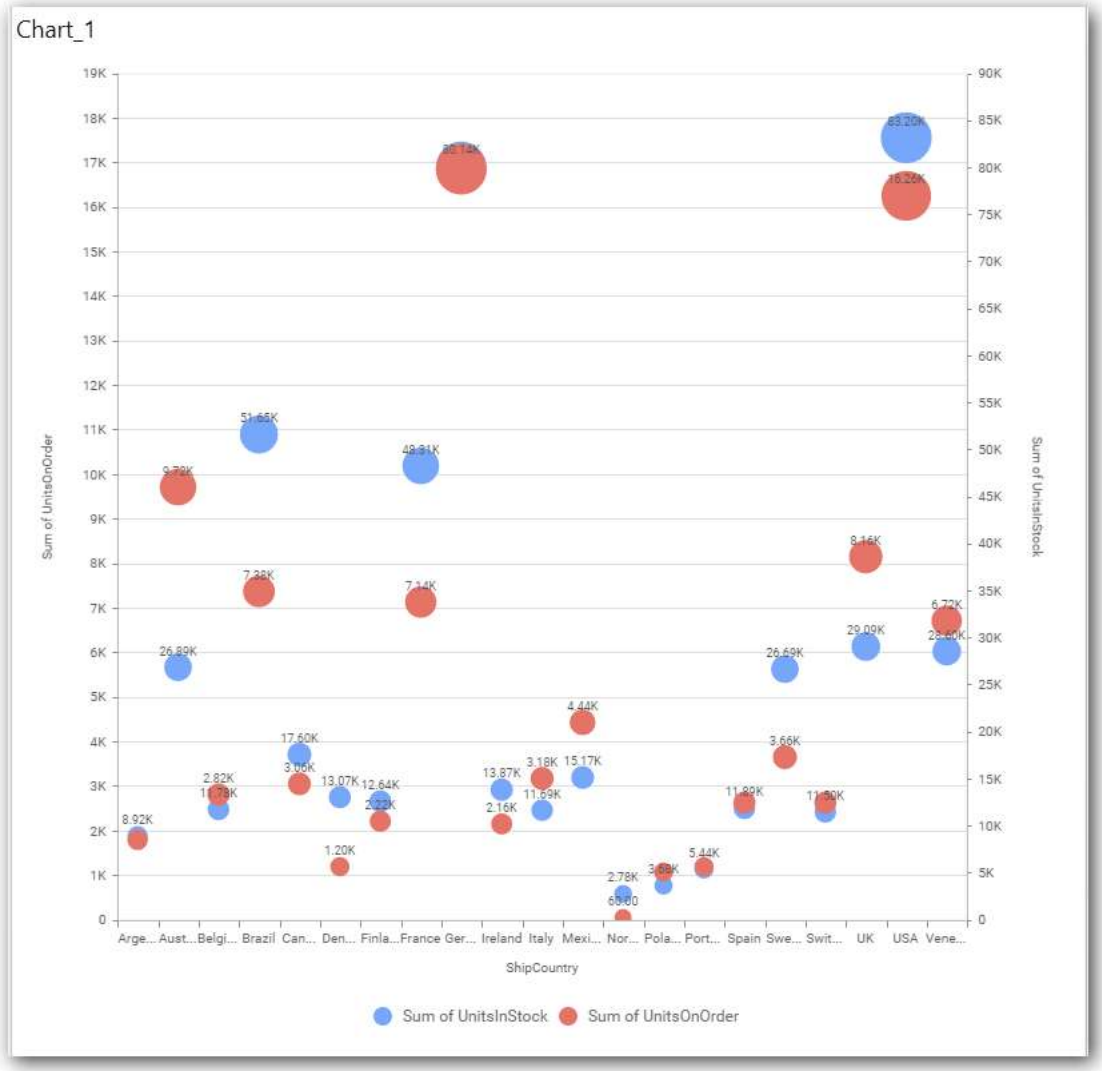
This allows you to define which measure column need to be plotted against which value axis (primary or secondary).



Primary Value Axis

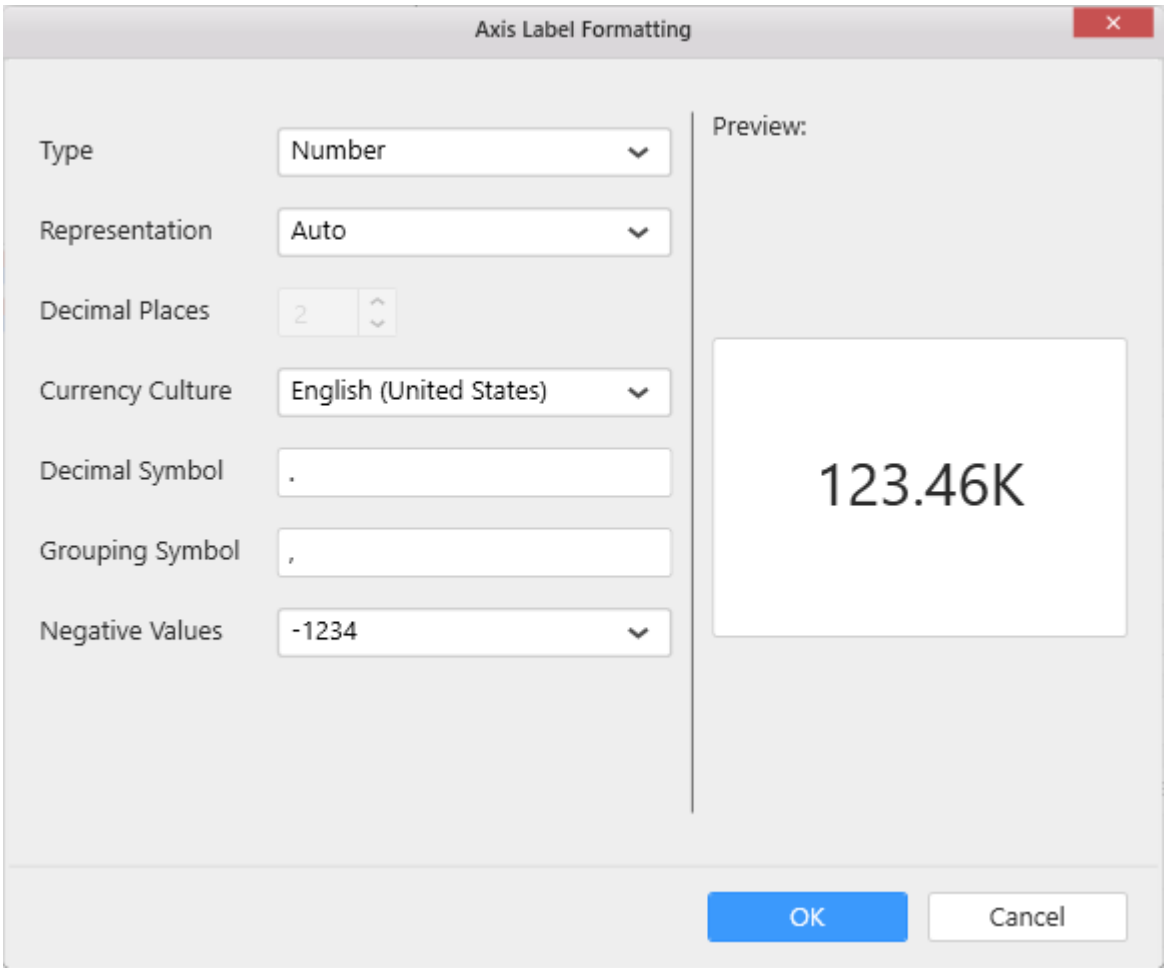


Secondary Value Axis



Axis Label Formatting

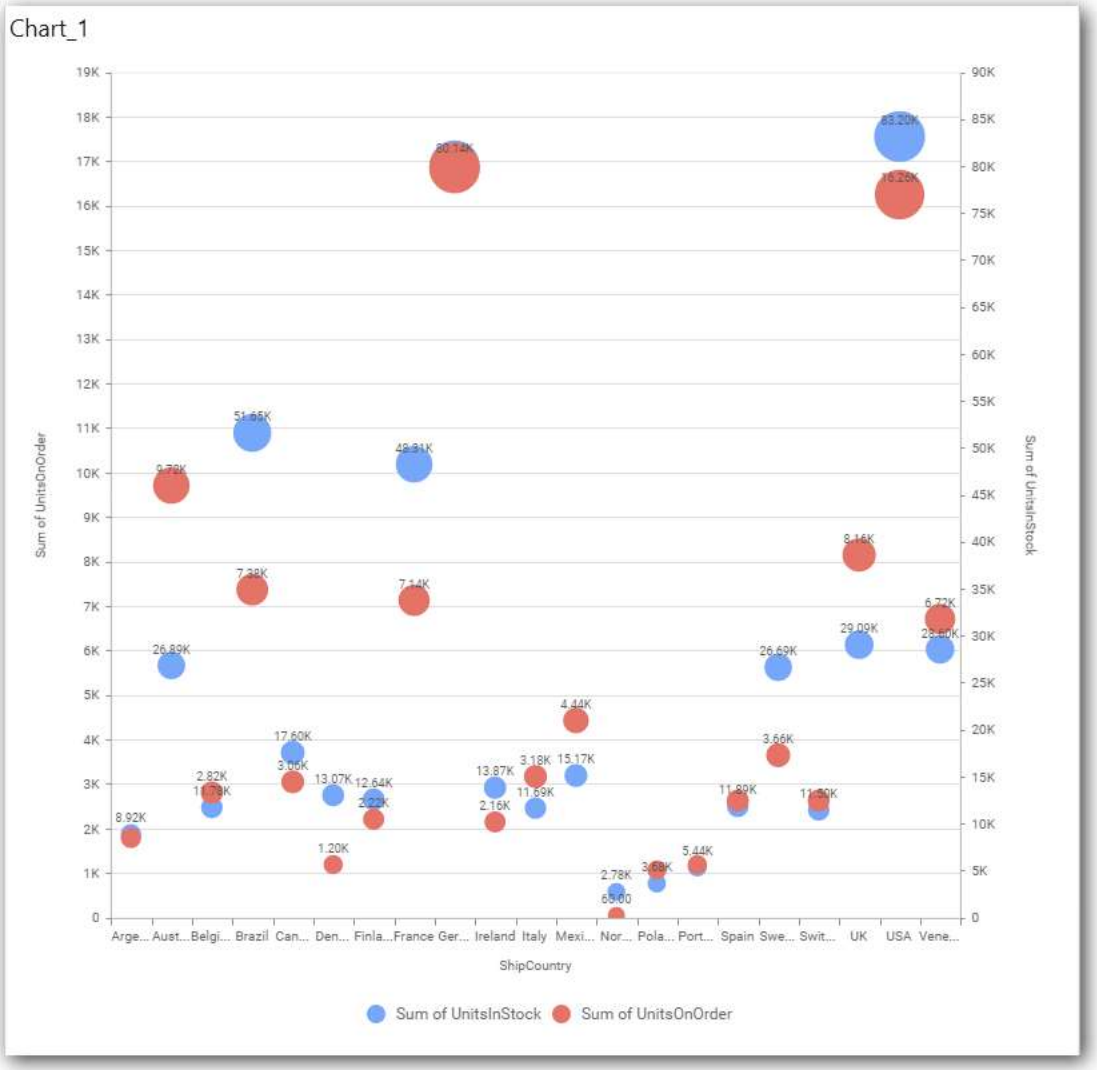
This allows you to handle different formatting options like display type, denominations, decimal places, currency culture and negative value display format to the value axis labels. Click on Axis Label Formatting button in Plot Axis Settings window will launch the following editor to configure settings.



Grid Line Settings

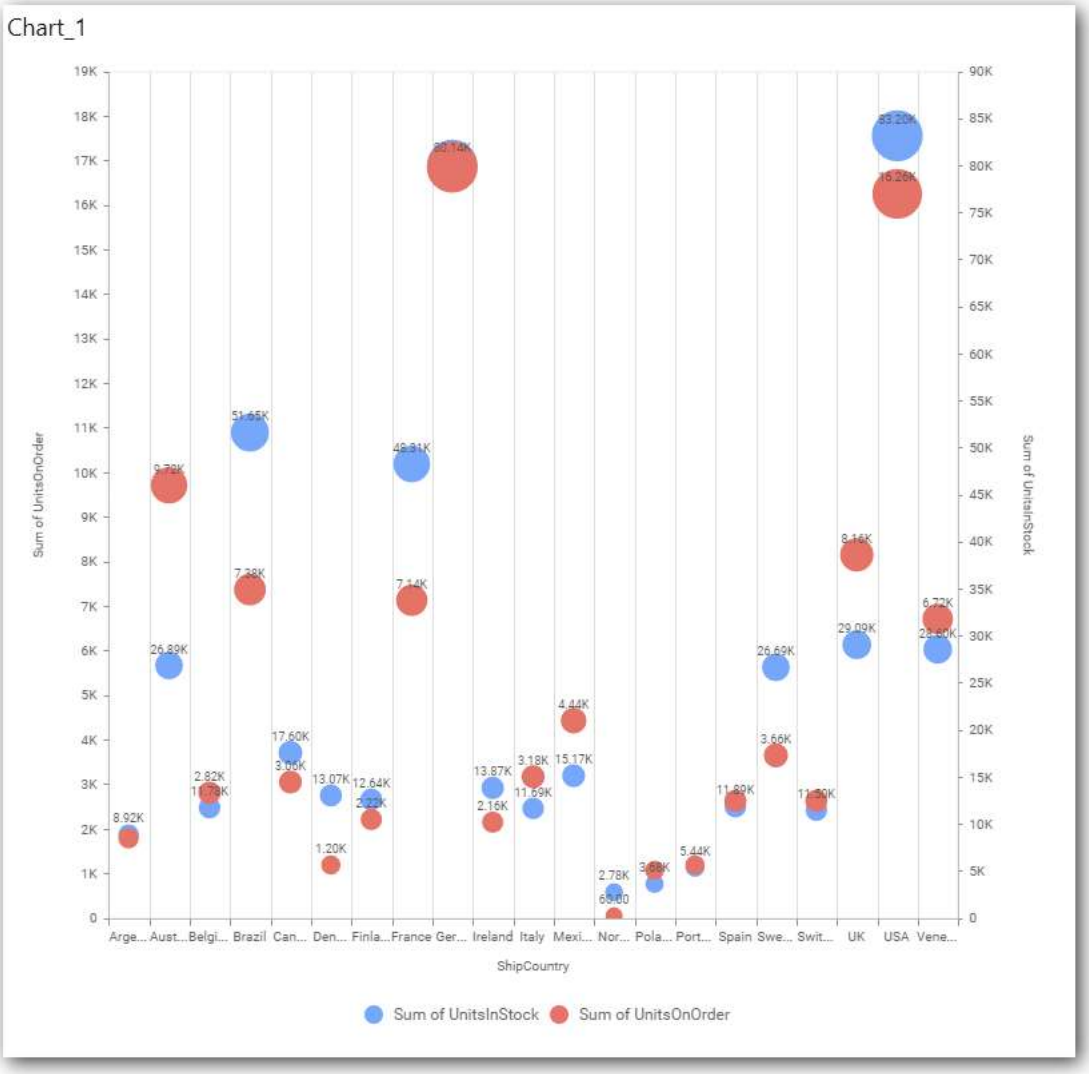
Primary Value Axis

This allows you to enable the Primary Value Axis gridlines.



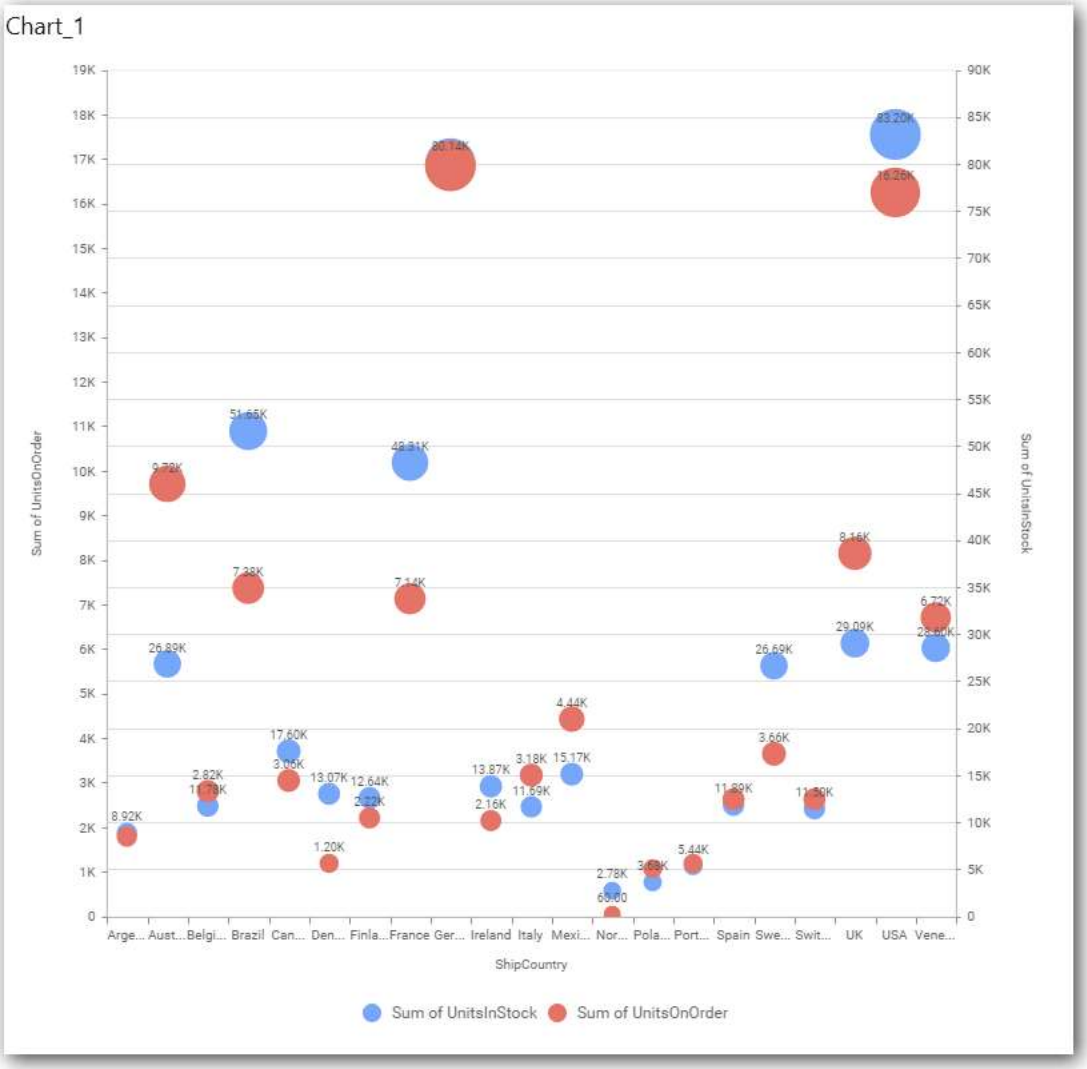
Category Axis

This allows you to toggle the visibility of Category axis gridlines.

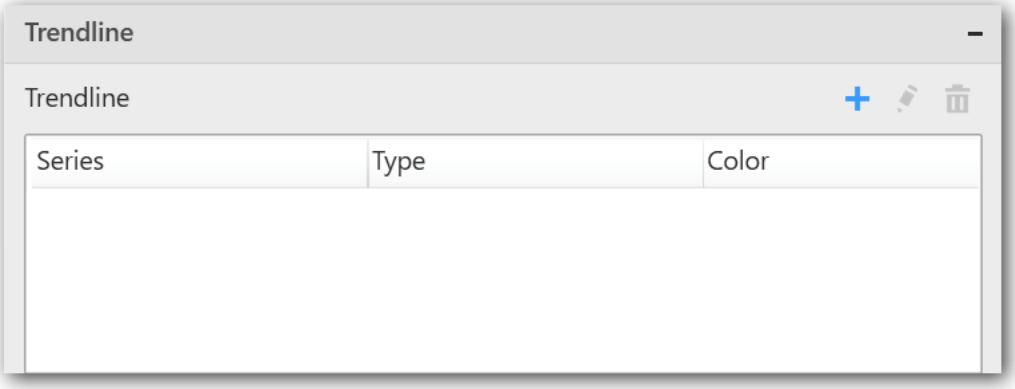


Secondary Value Axis

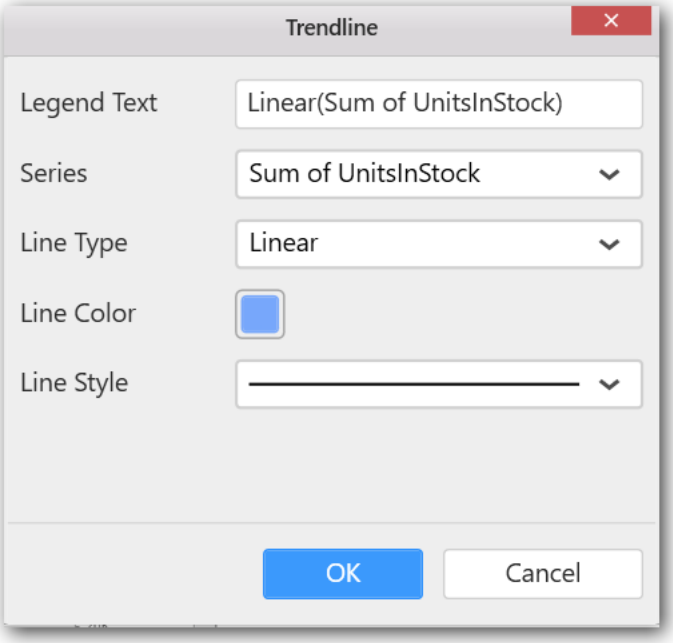
This allows you to toggle the visibility of Secondary Value Axis gridlines.



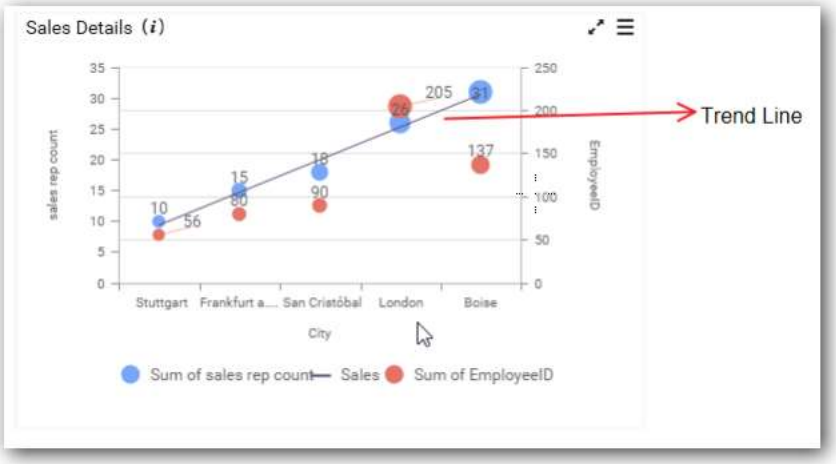
Trend Line Settings



You can add trend line to chart based on dropped measure that you select. You can also customize, the legend text, line type and line color. Trend line is not visible, by default.



After applying these settings, it will reflect in chart like below.



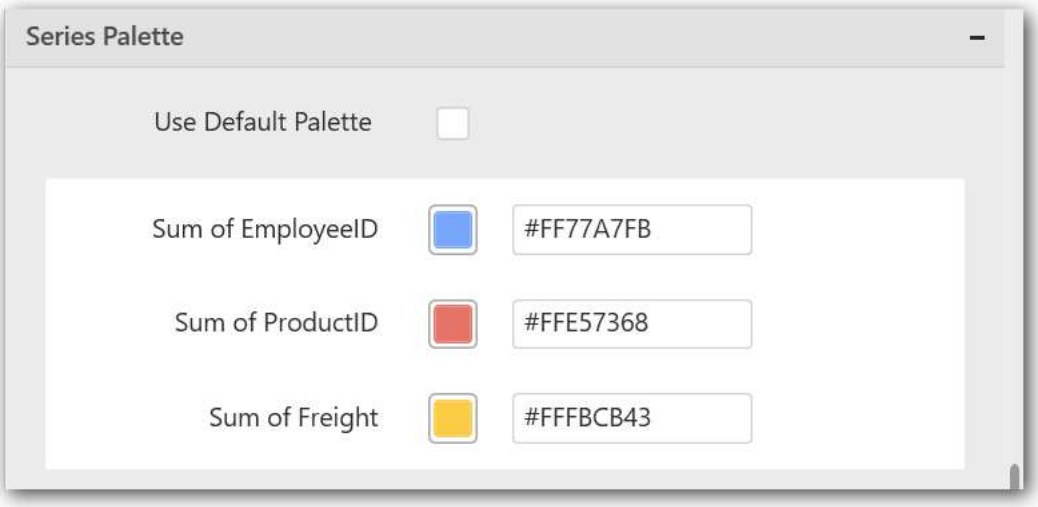
You have options to edit or delete added trend lines.

Series Palette

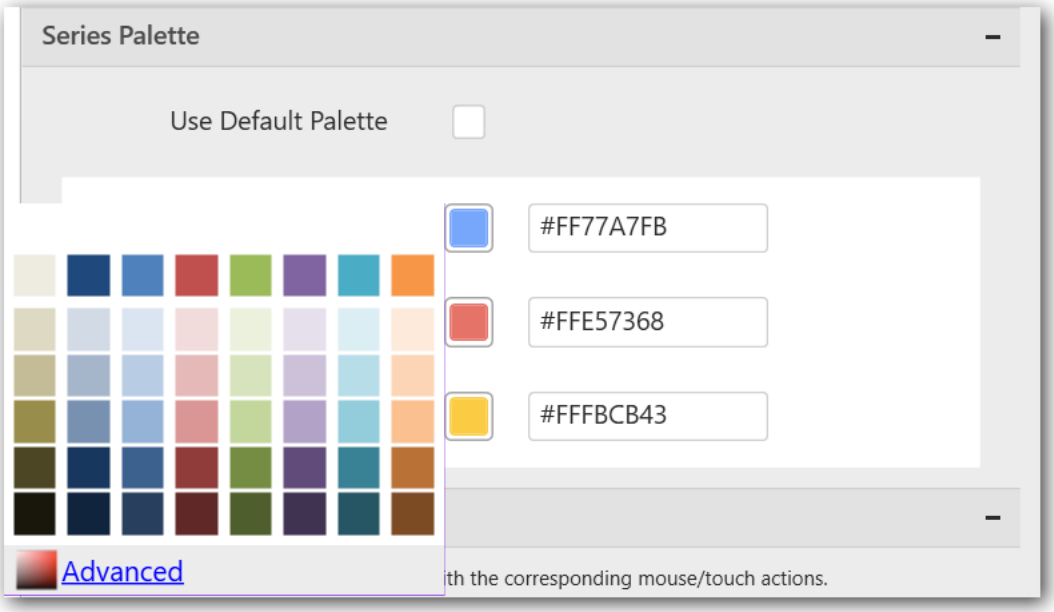
This allows you to customize the chart series color through Series Palette section.

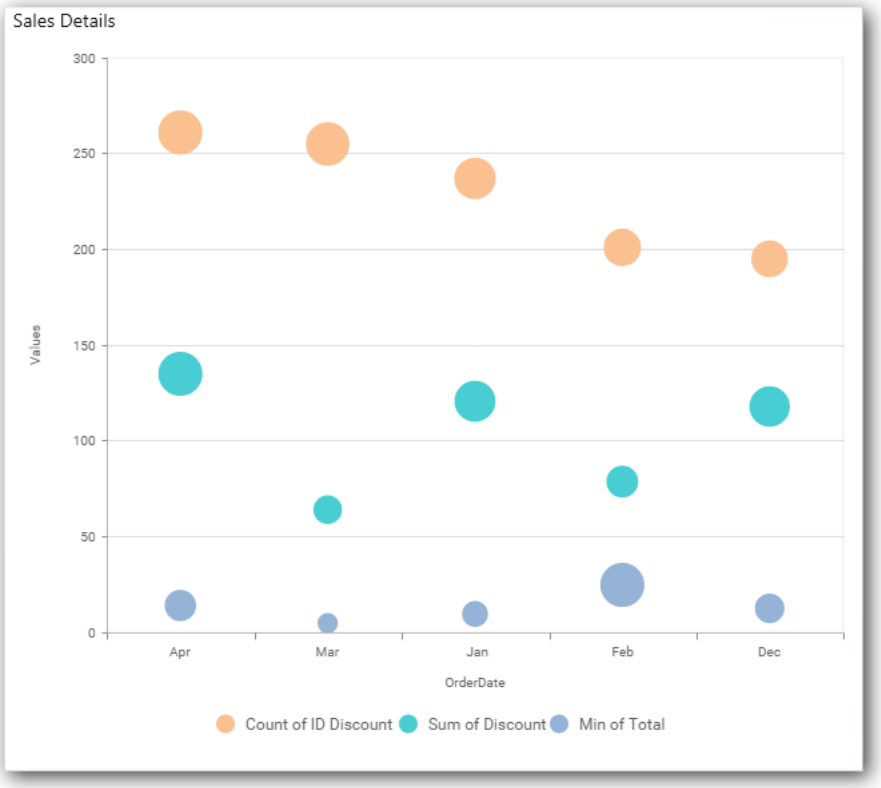
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to series.



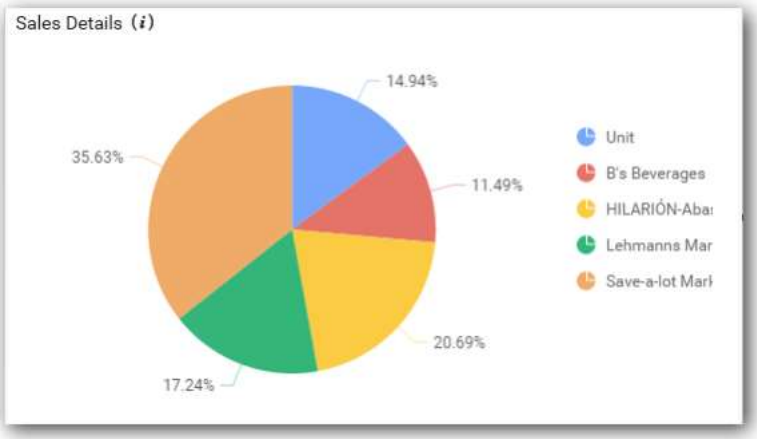
By toggle off the Use Default Palette, you can customize the series colors. This section shows, list of series' labels on the left-hand side and corresponding series color on the right-hand side. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Pie Chart

Pie Chart allows you to showcase proportionality of each item to the total in the form of pie-slices.

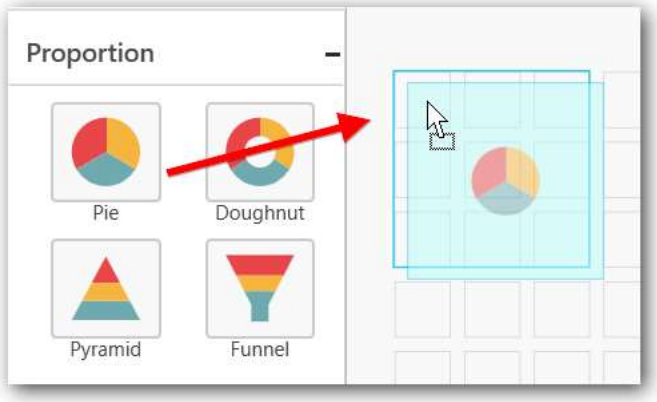


How to configure the flat table data to Pie Chart?

Pie Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

To configure data into pie chart follow the steps

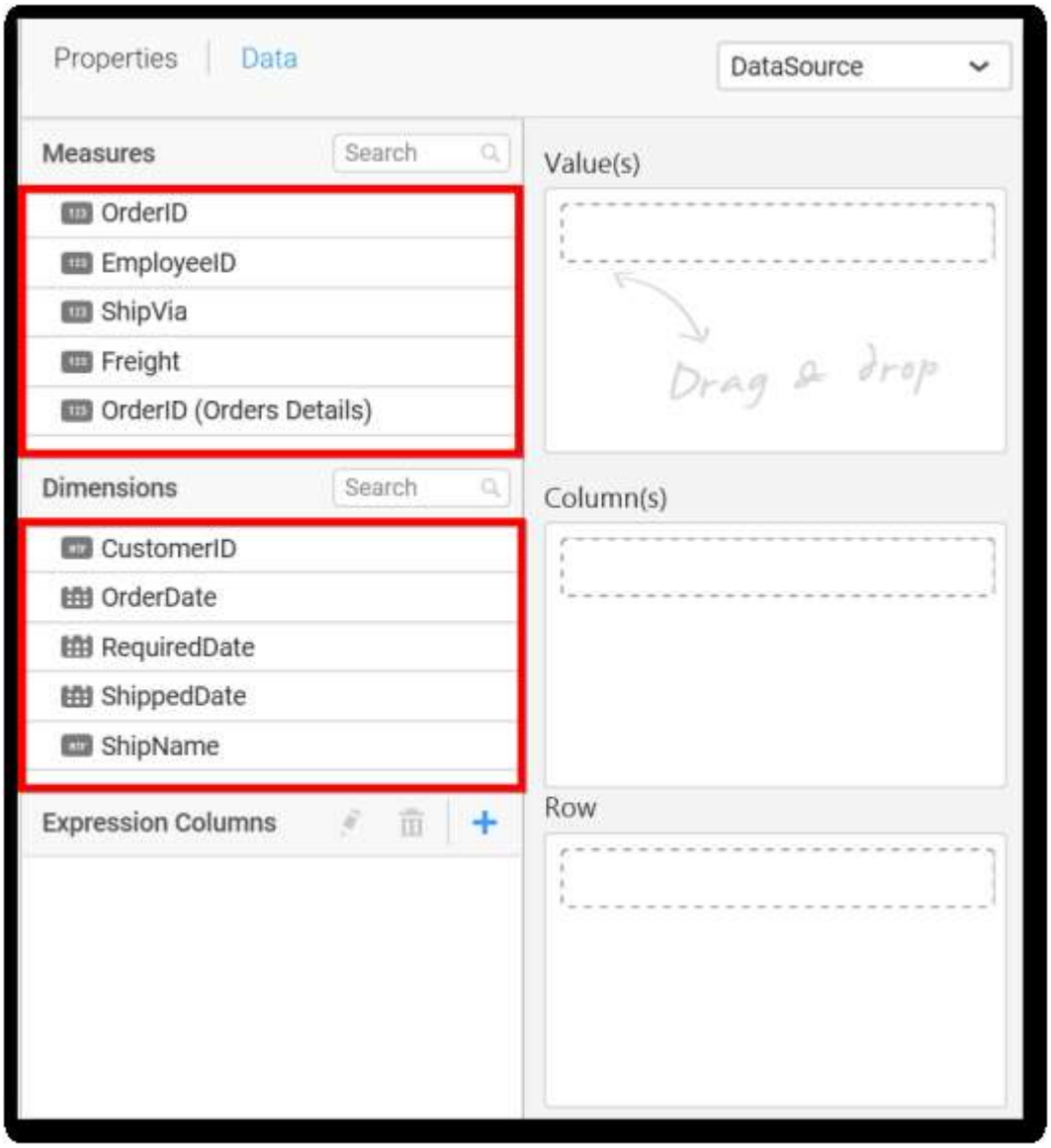
Drag and drop the Pie chart into canvas and resize it to your required size.



Connect to data source.

Focus on the Pie chart and Click on Assign Data.

The data pane will be opened with available Measures and Dimensions from the connected data source.

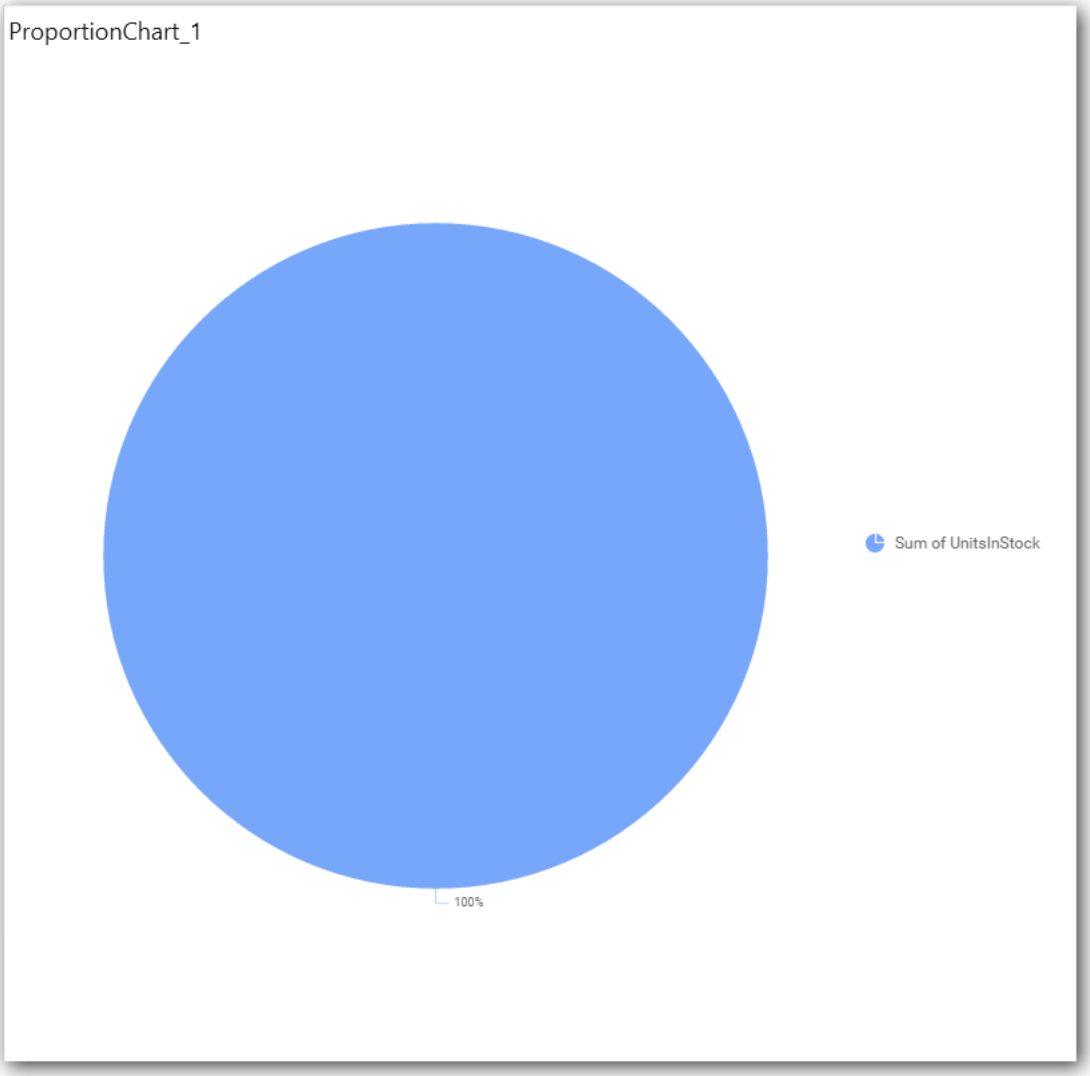


You can add the required data from Measures and Dimensions into required field, you can also create Expression Columns using Expression Designer

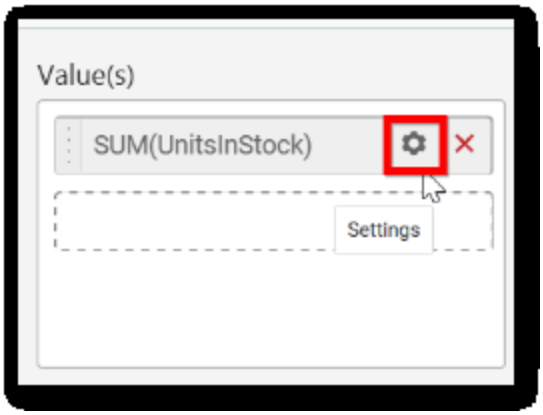
Adding Value(s)

You can add Measures into Value(s) field by drag and drop the required measure.

Now the Pie chart will be rendered like this.

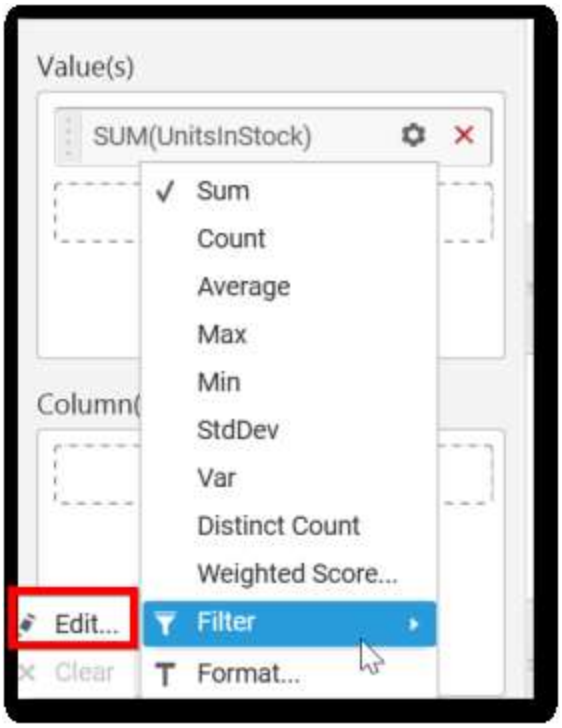


You can change the Settings option.

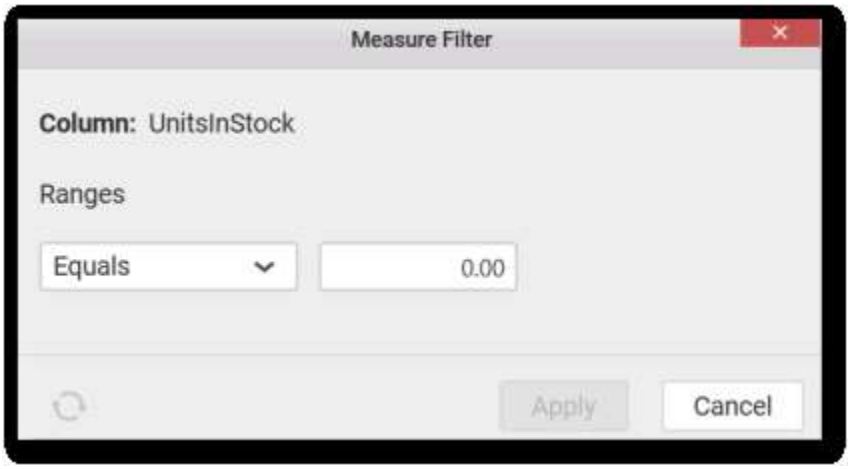


You can select the required summary type from the available summary types shown in Settings.

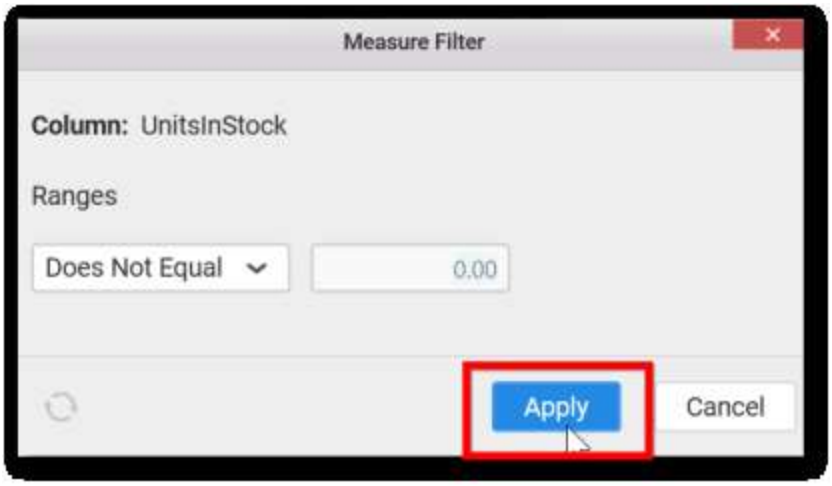
You can filter the data to be displayed in pie chart by using filter.



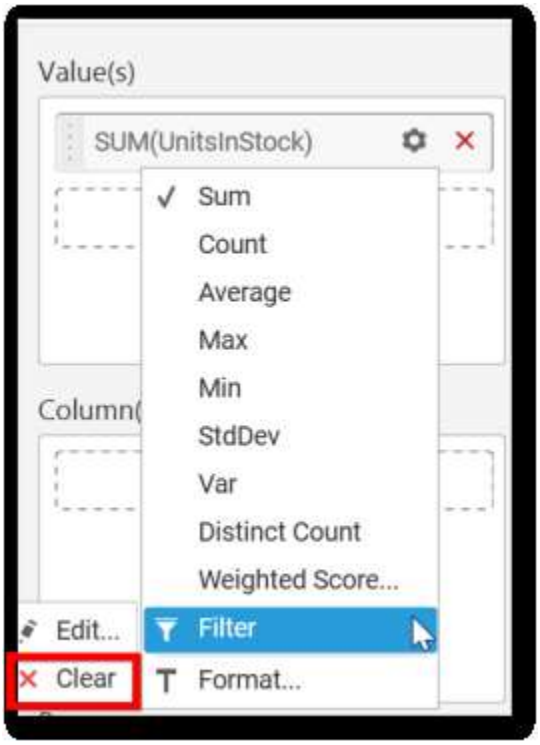
When you click the Measure Filter option will appear.



You can select the Condition to be applied in the shown list box and set the value in text box, select the condition and range and then click on Apply.

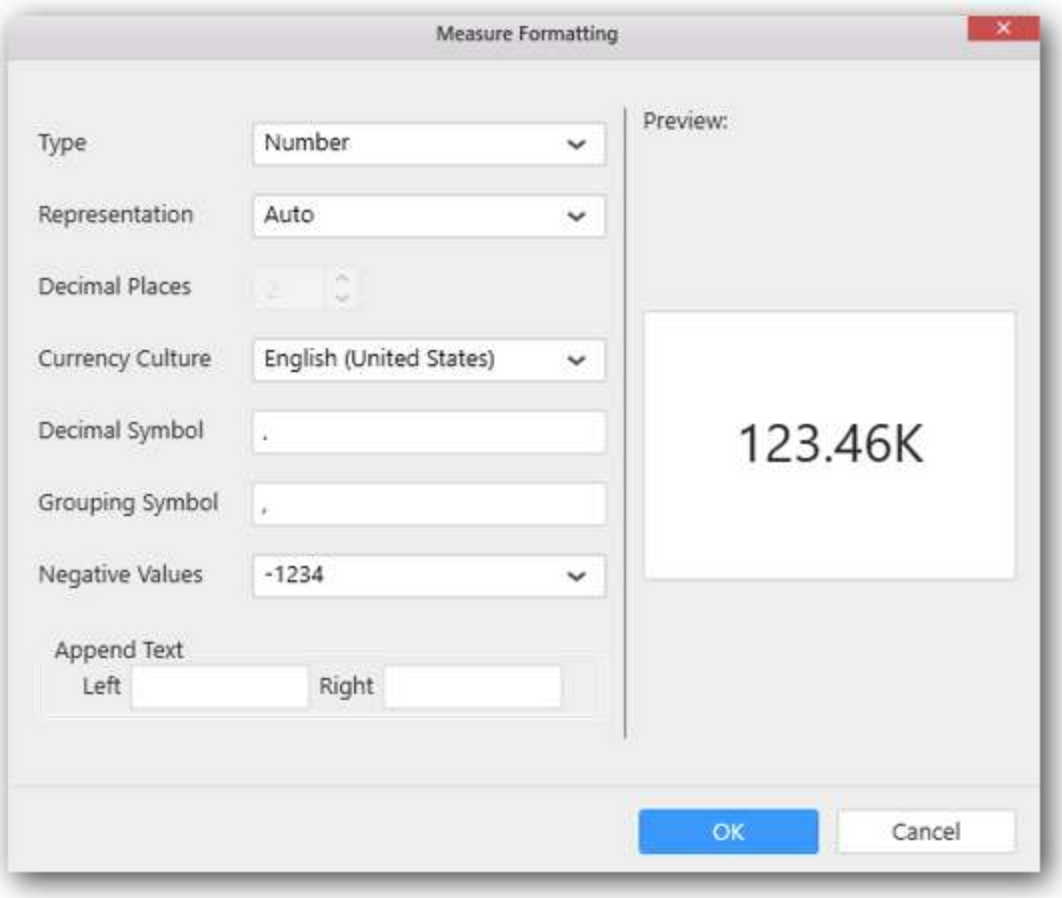


You have option to clear the applied filter. Click on clear to remove the filters.

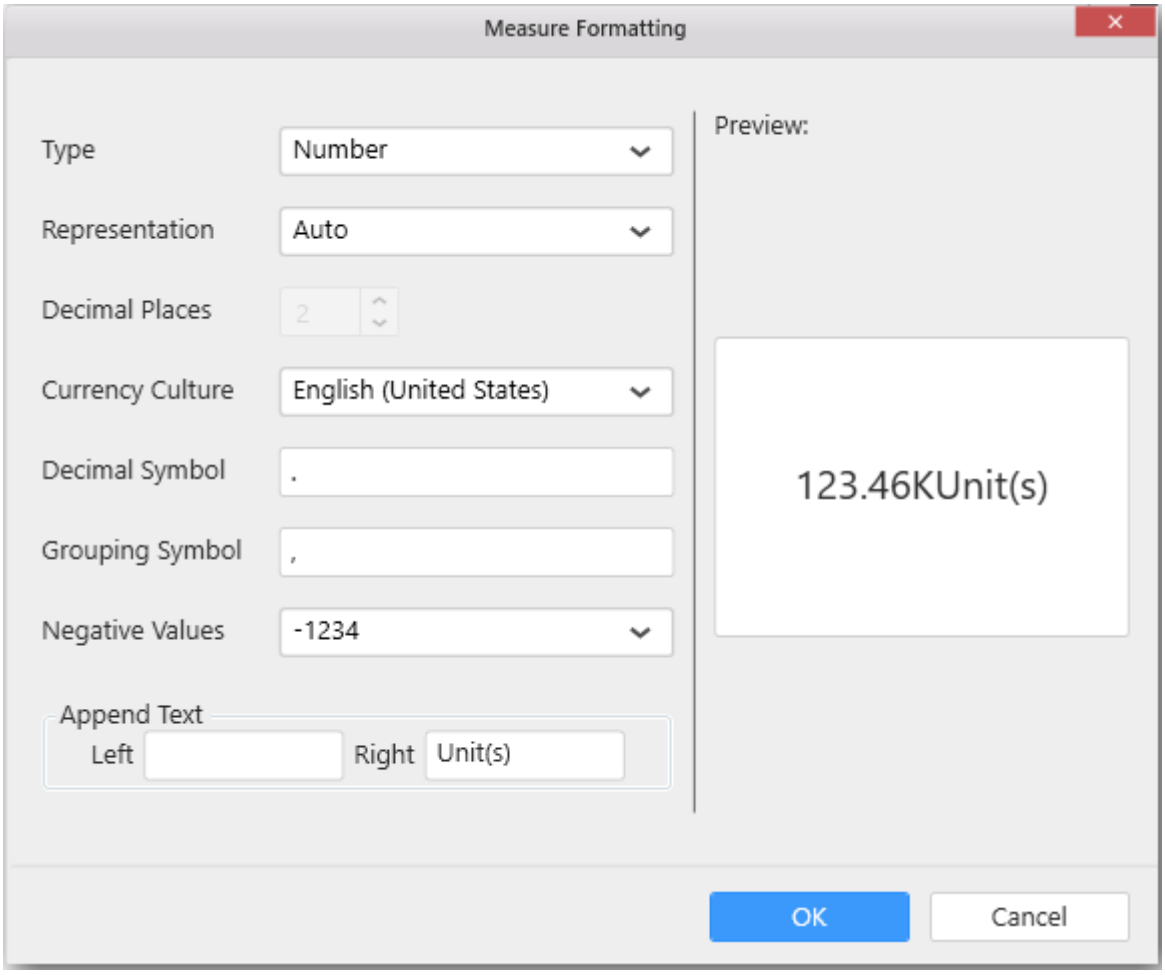


You can format the data to be displayed in the Pie chart by using format option.

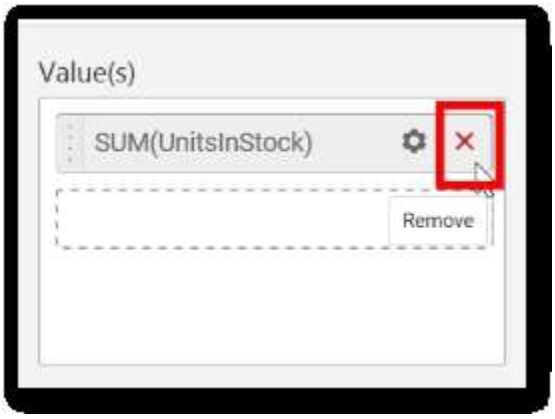
The measure formatting option will be shown, select the format that you want and click OK.



Select the required format and click on OK.

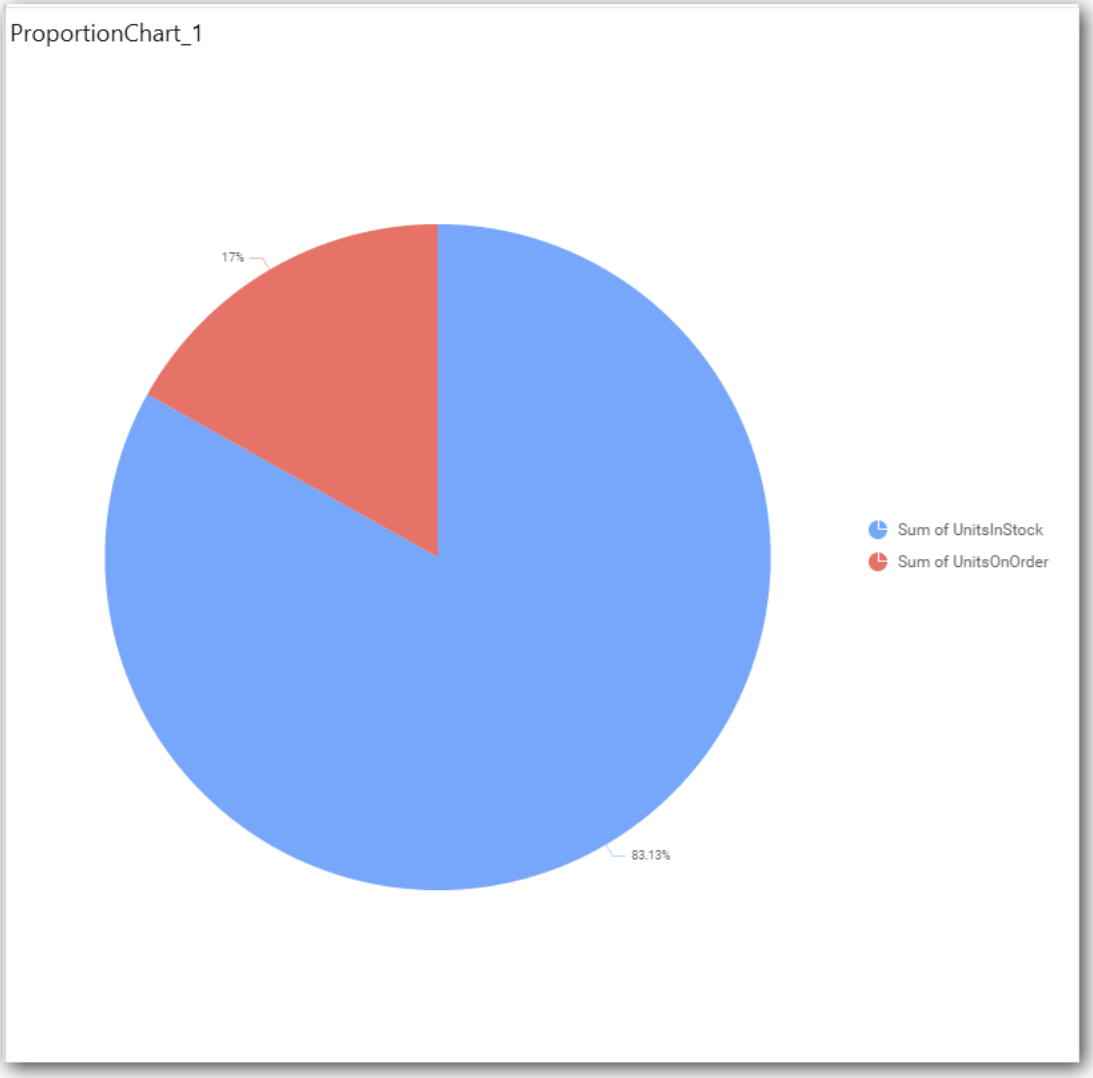


To remove the added value fields click on x button.



You can add multiple Measures into Value.

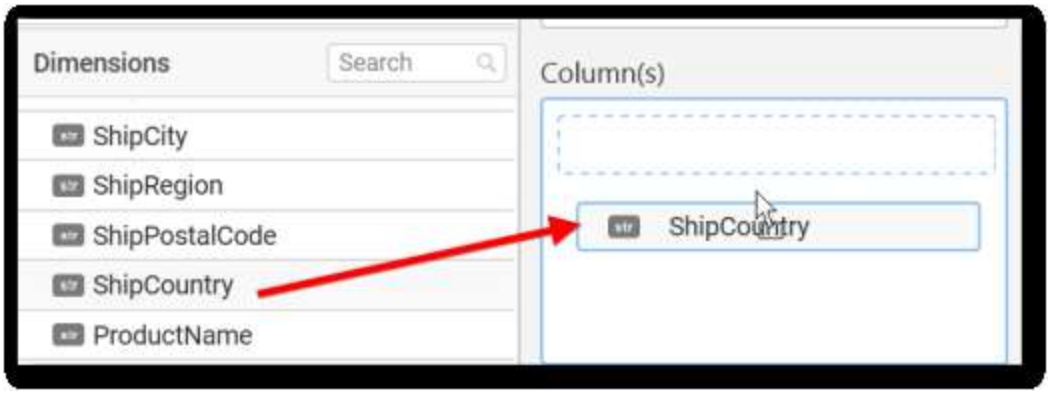
The chart will be rendered as shown in the following image.



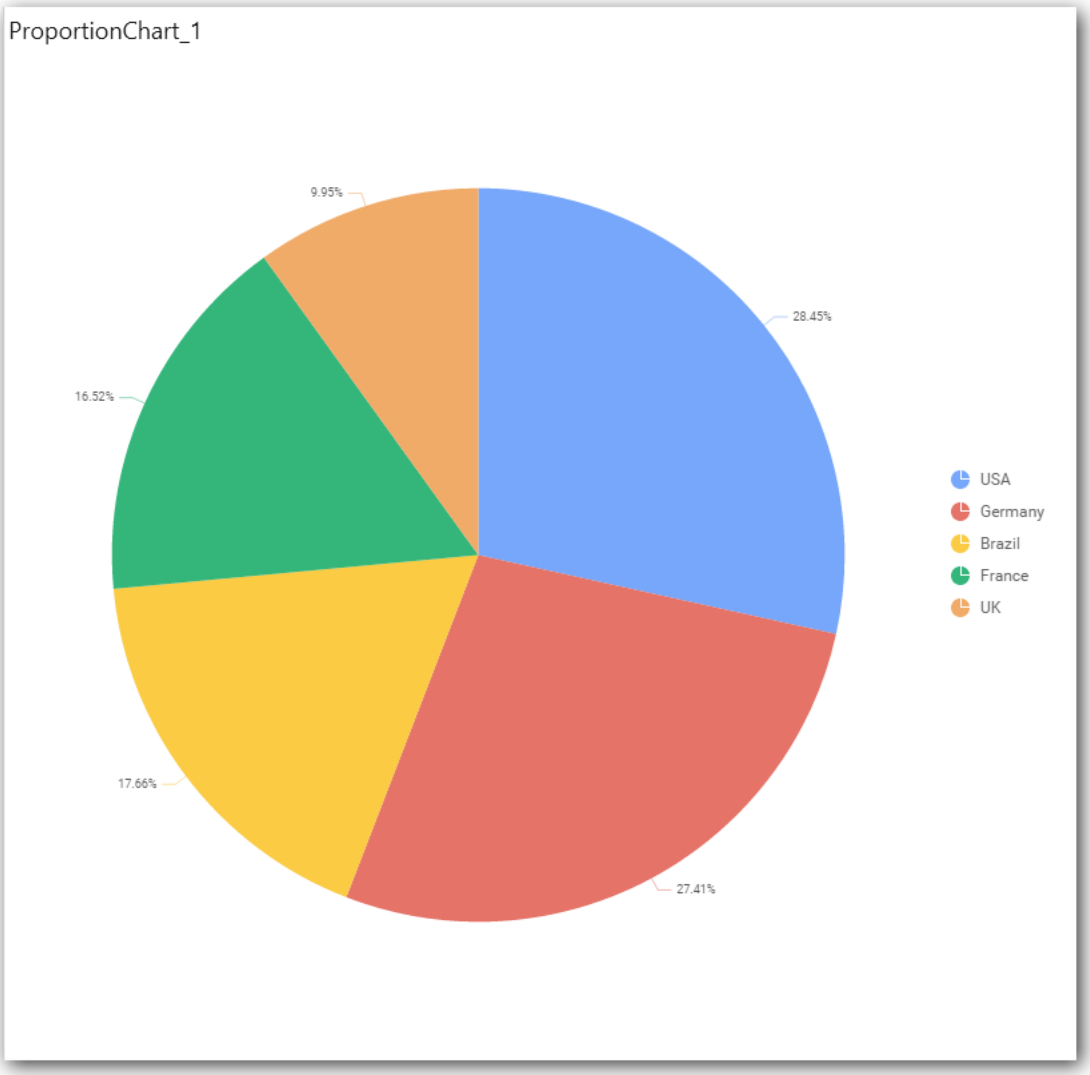
You can also drag and drop Dimension and Expression Column into Value(s).

Adding Column(s)

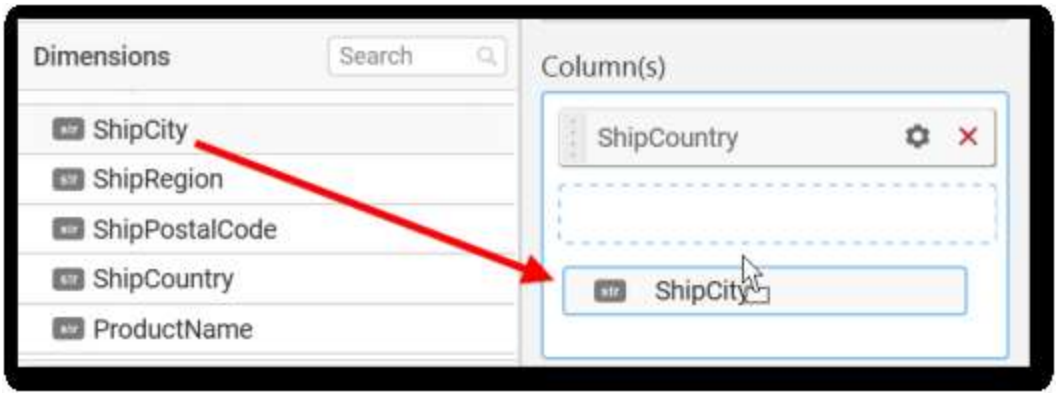
Drag and drop the Dimensions into Column(s).



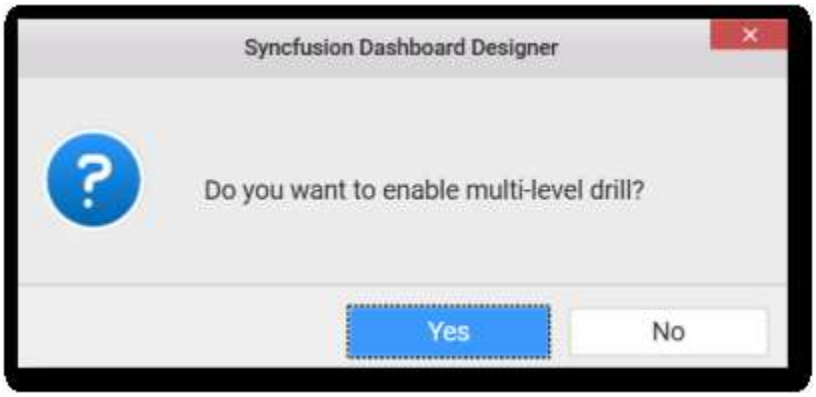
Pie chart will be rendered like this.



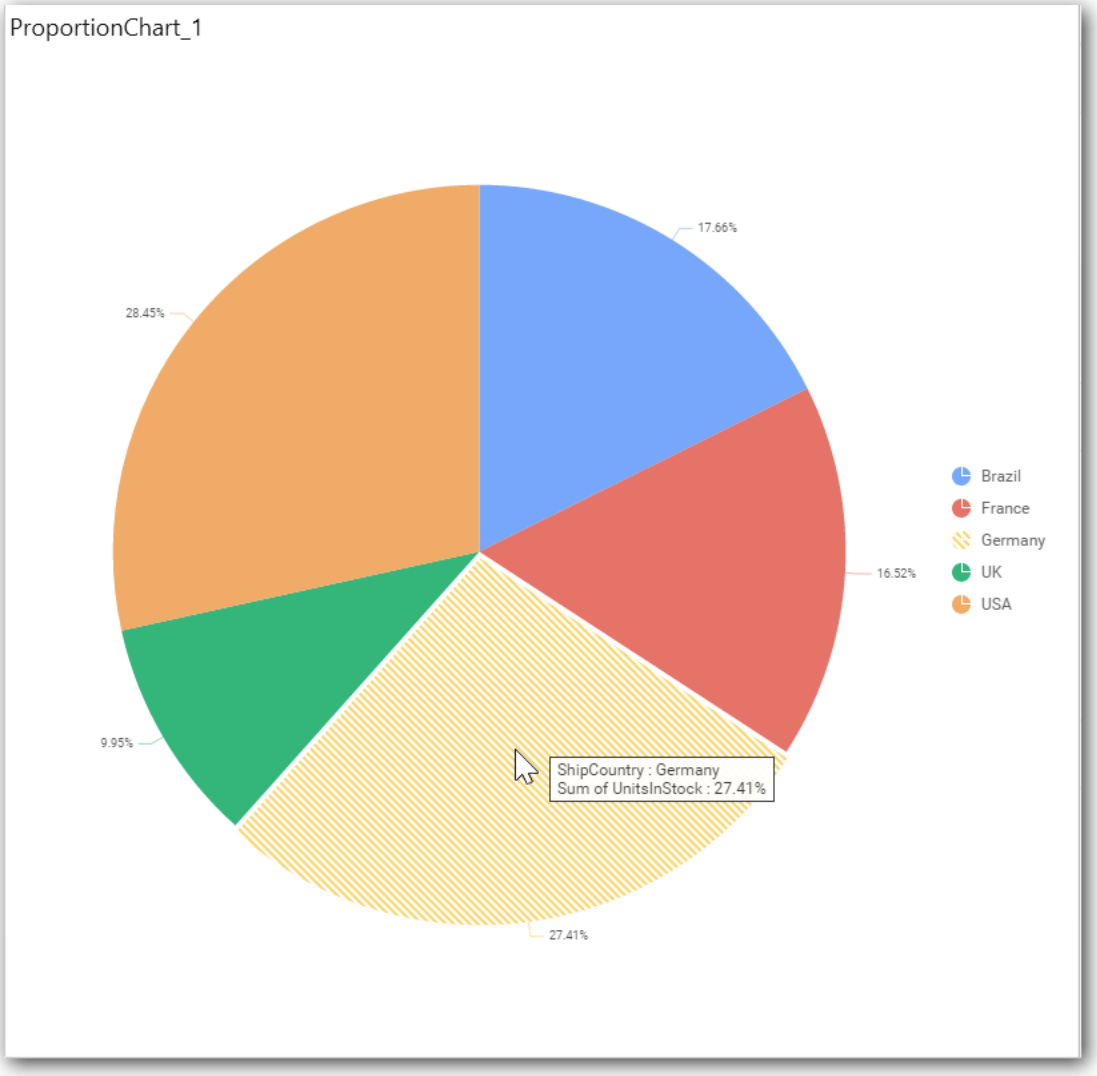
You can add more than one value into Column(s) field.



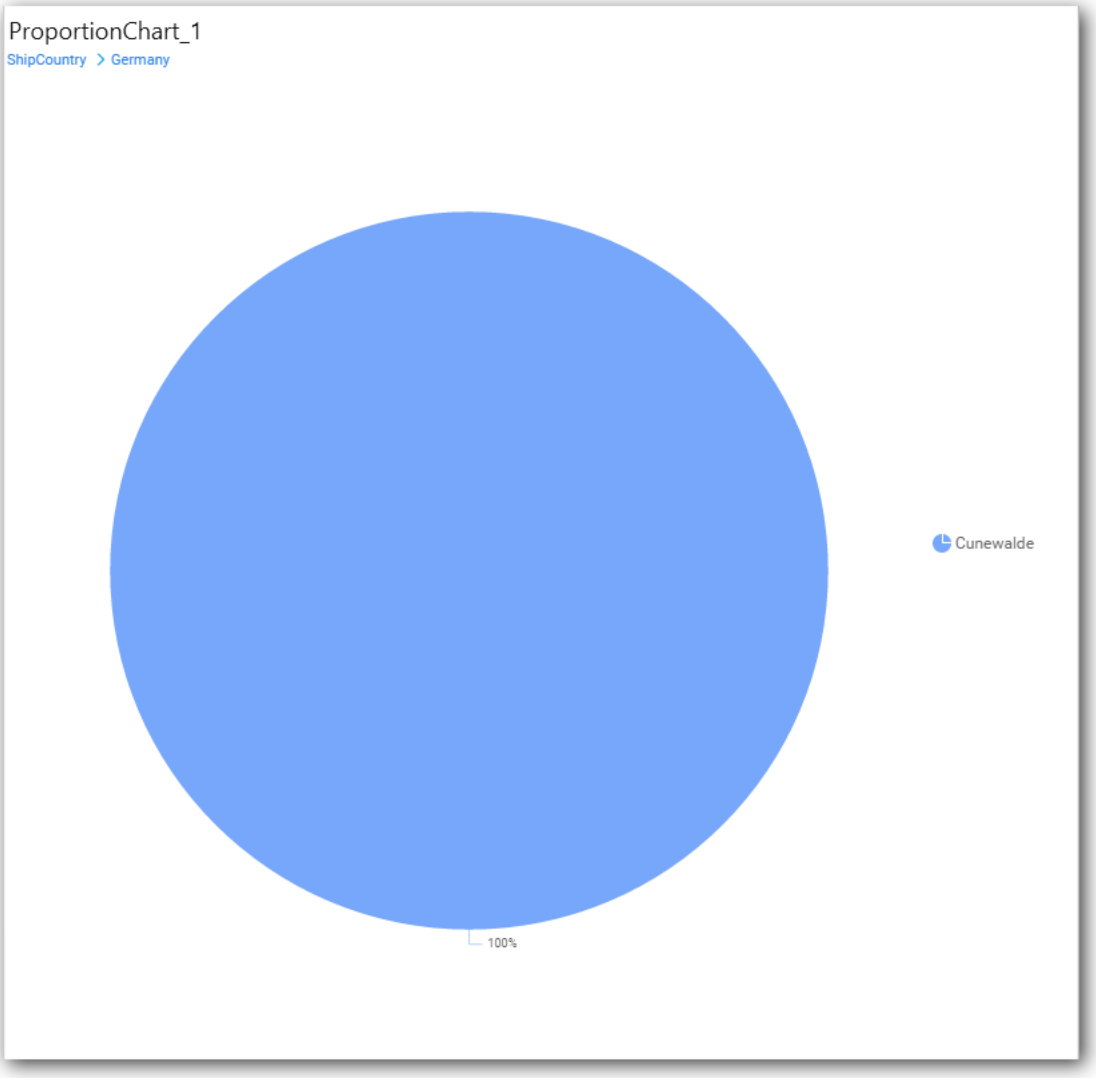
The following message will open.



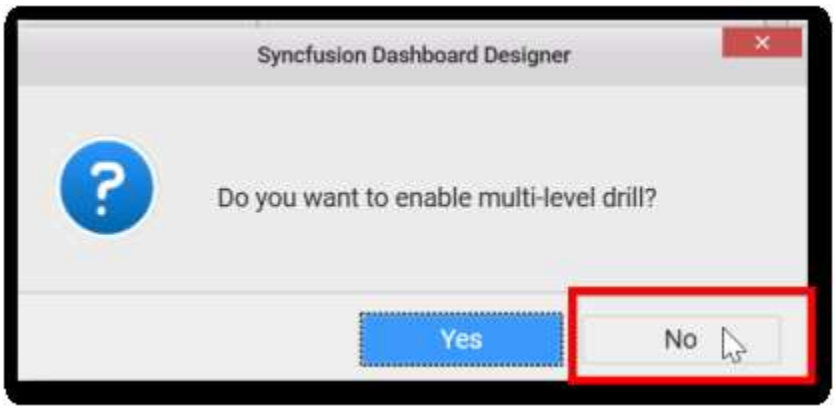
To enable drill down click yes.



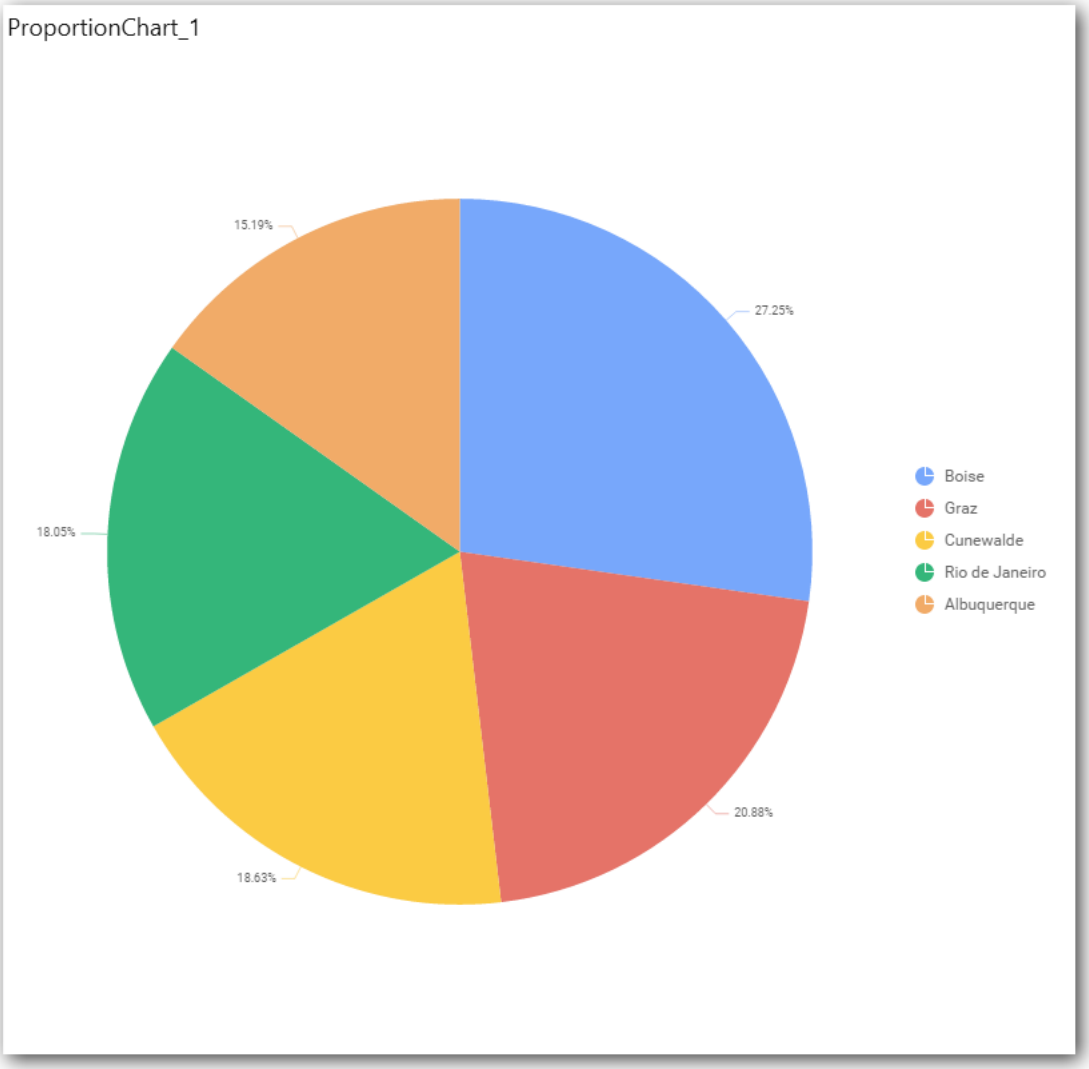
The drilled view of the chart region selected.



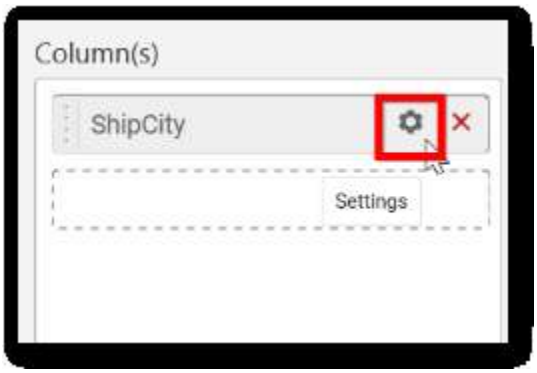
To continue without drill down click No.

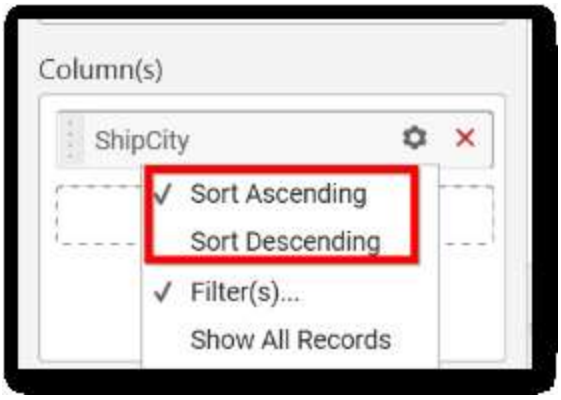


The old column value will be replaced by new column value.



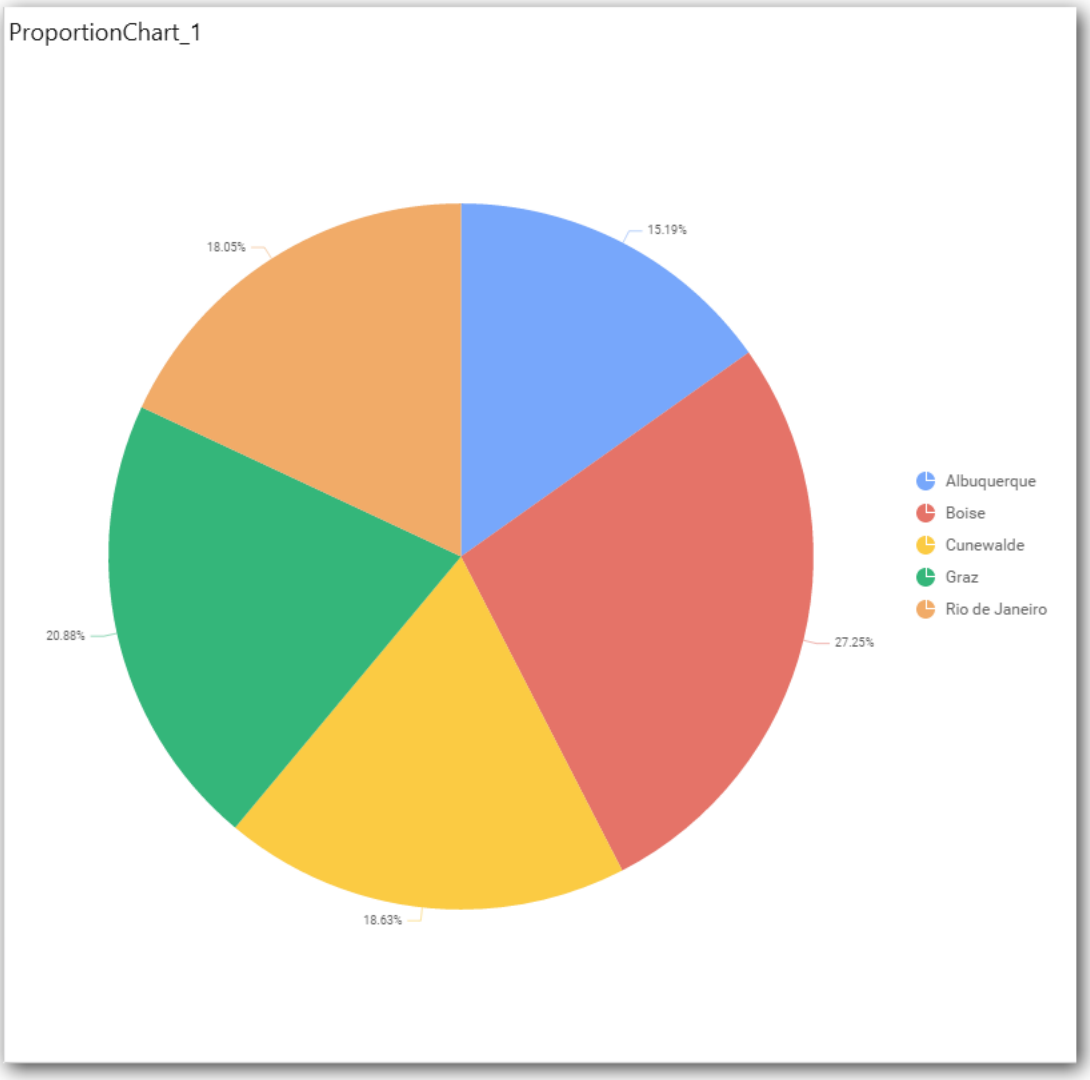
You can change the Settings.



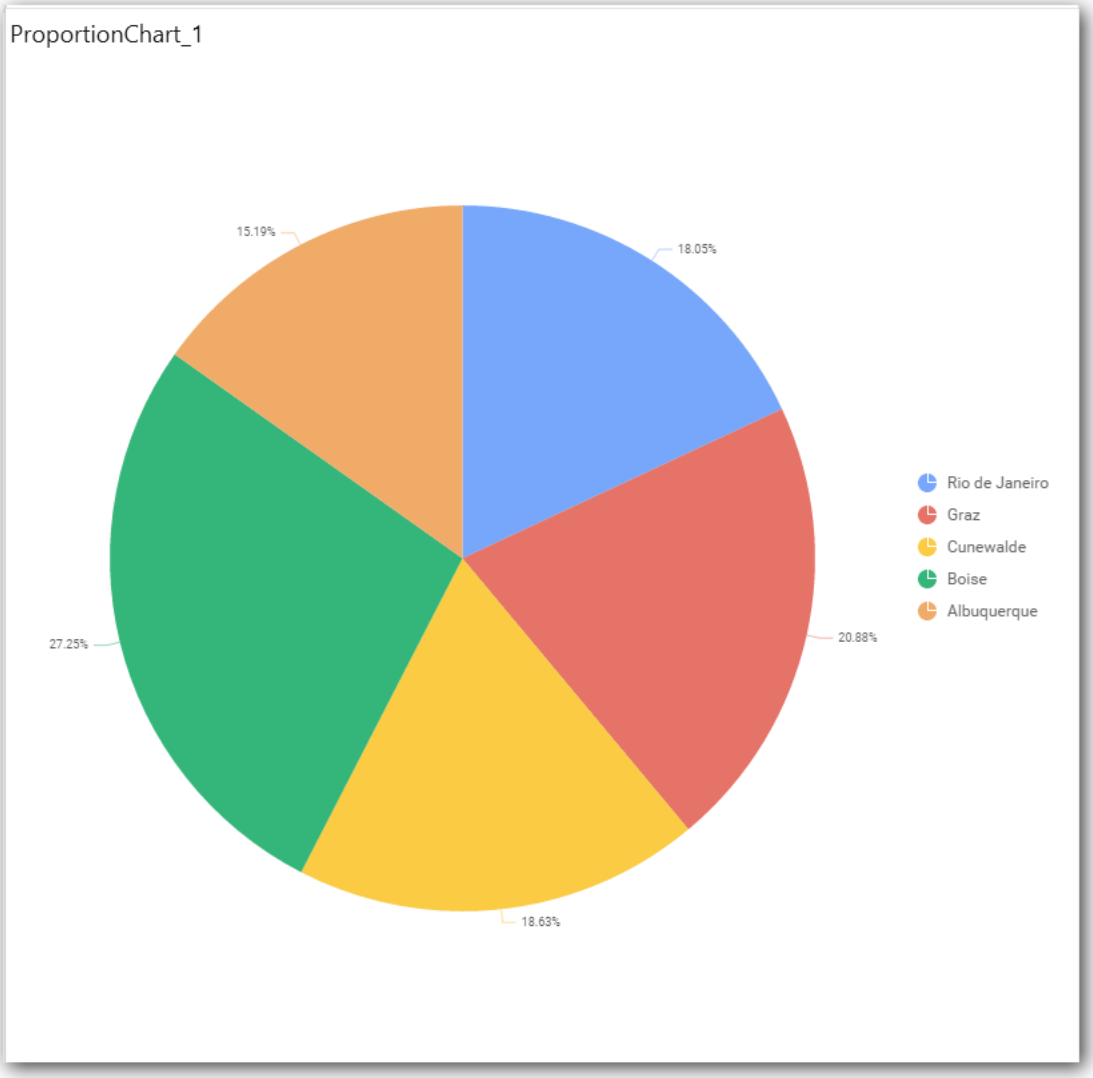


You can select sort Ascending or Descending.

Ascending order

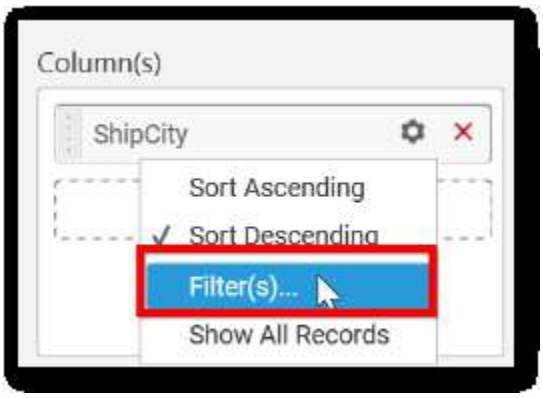


Descending order



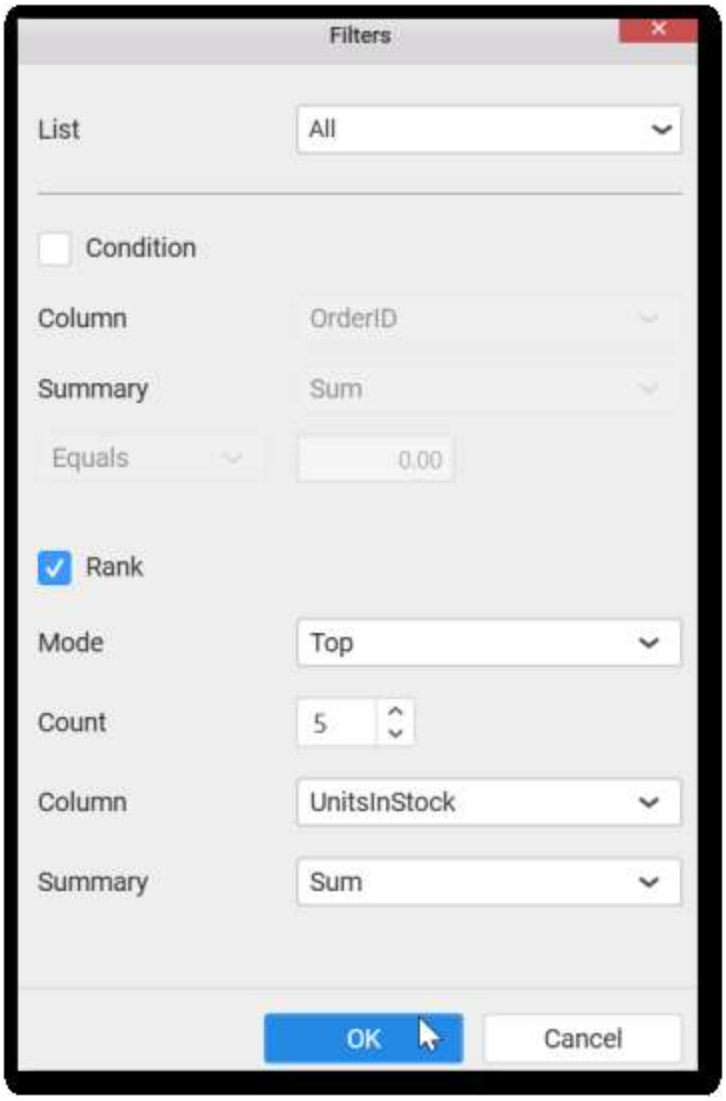
You can apply filters by selecting filter in settings

Note: Filter will be set by default for top 5 records.



The filters option will open.

Select the needed Conditions and rank and then click Ok.

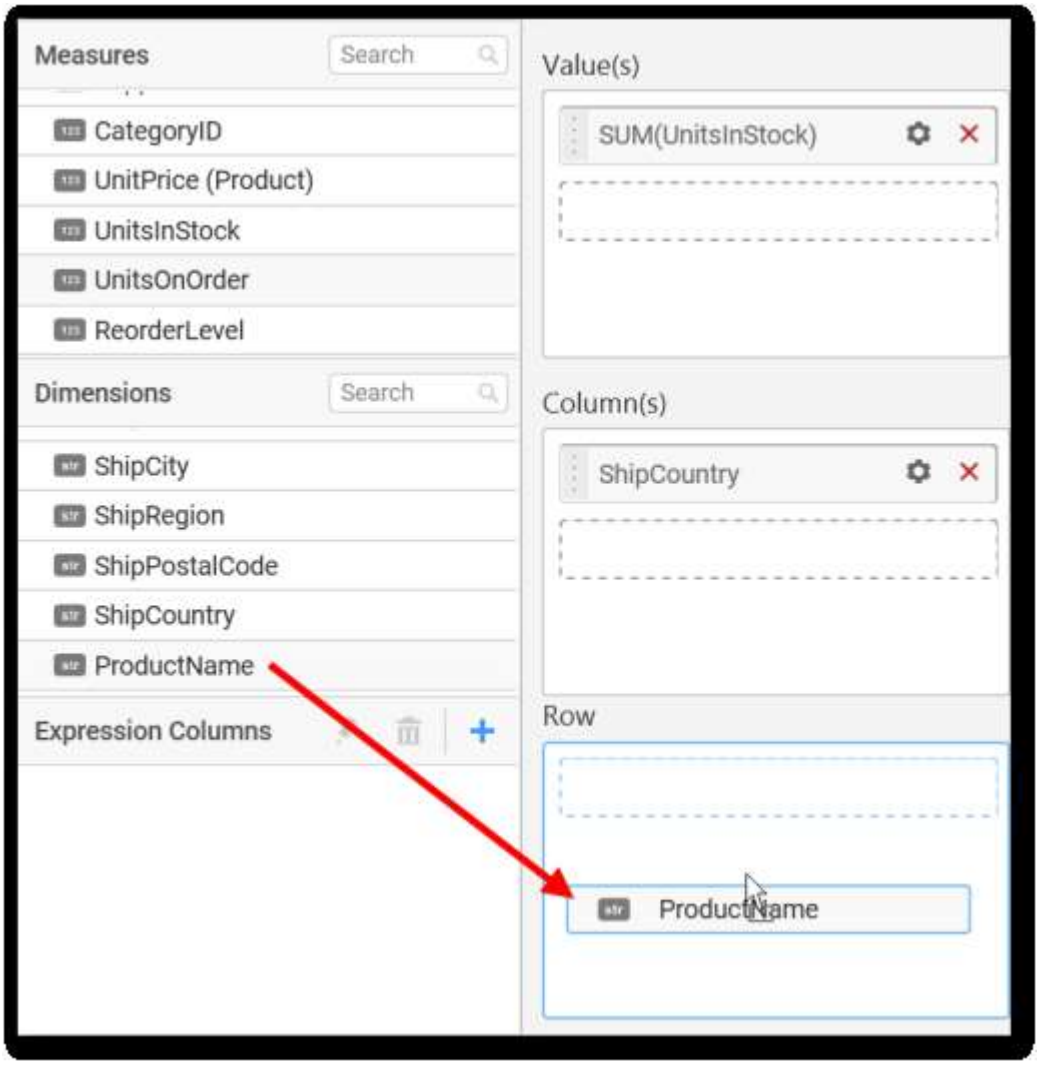


To show all records click on Show All Records.

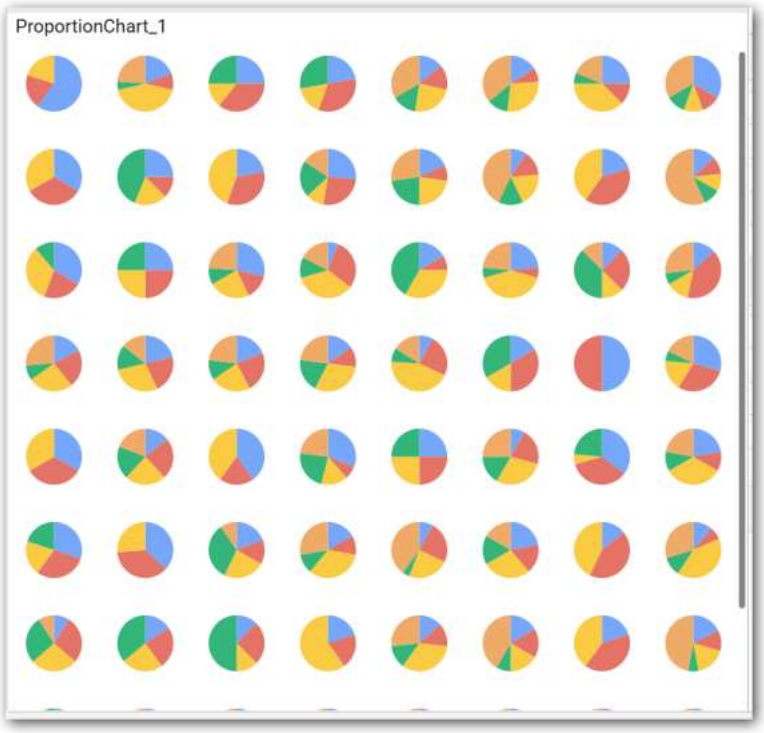
Similarly you can add the Measures and Expression Columns into column field.

Adding Row

You can drag and drop the Measure or Dimension into the Row field.



This will render pie chart in series.



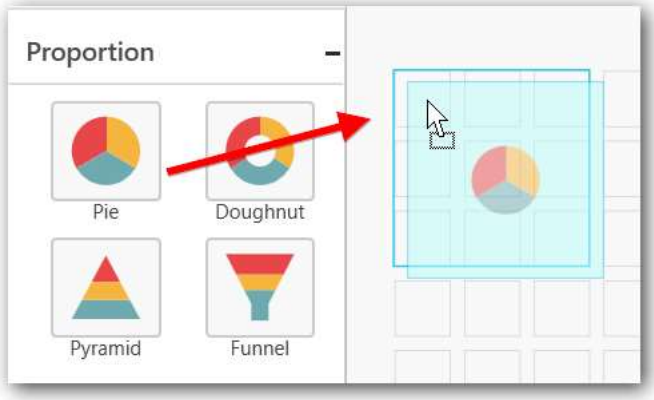
Scroll down to see all charts.

How to configure the SSAS data to Pie Chart?

Pie Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuration of SSAS data to Pie chart.

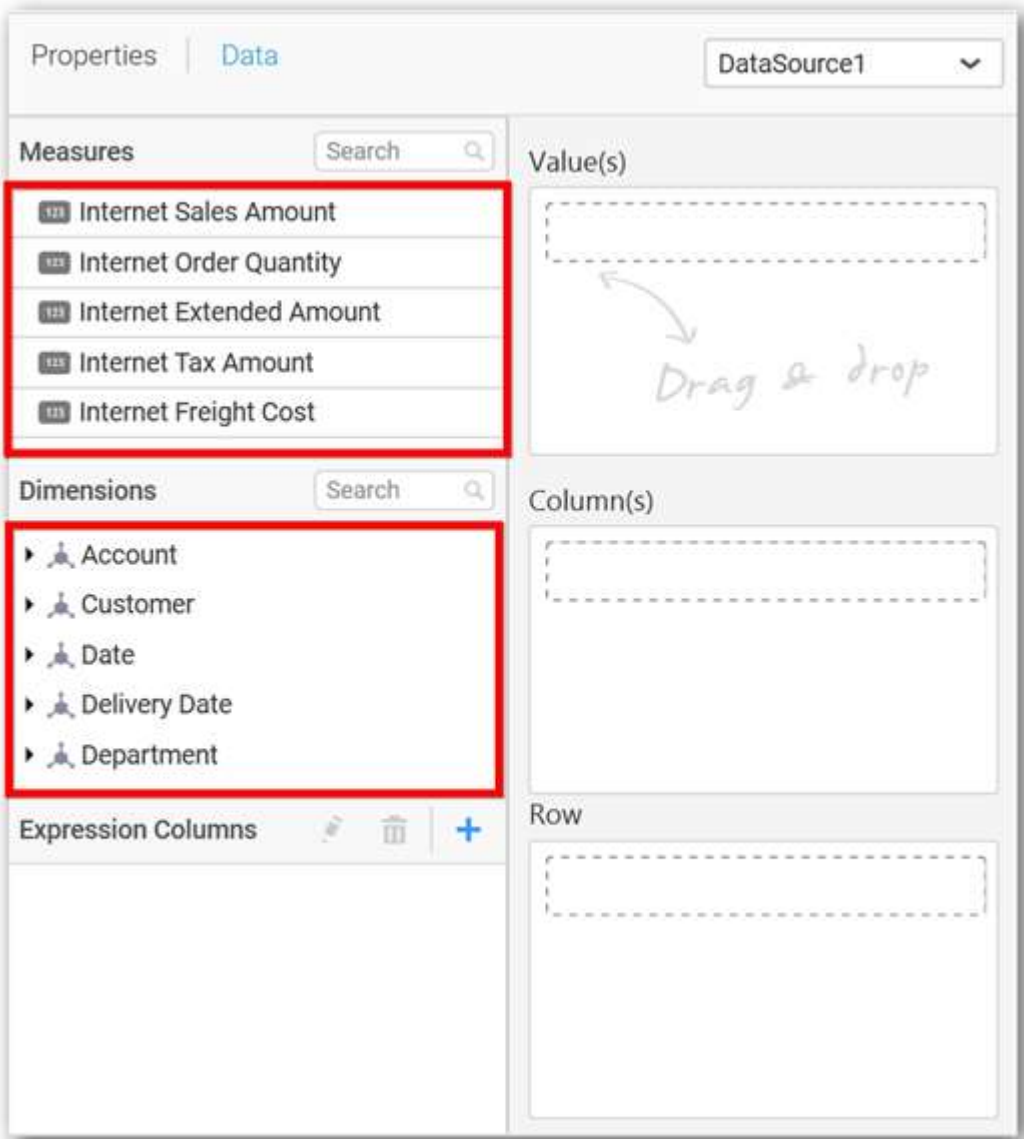
Drag and drop the Pie chart into canvas and resize it to your required size.



Select the dropped widget using mouse.

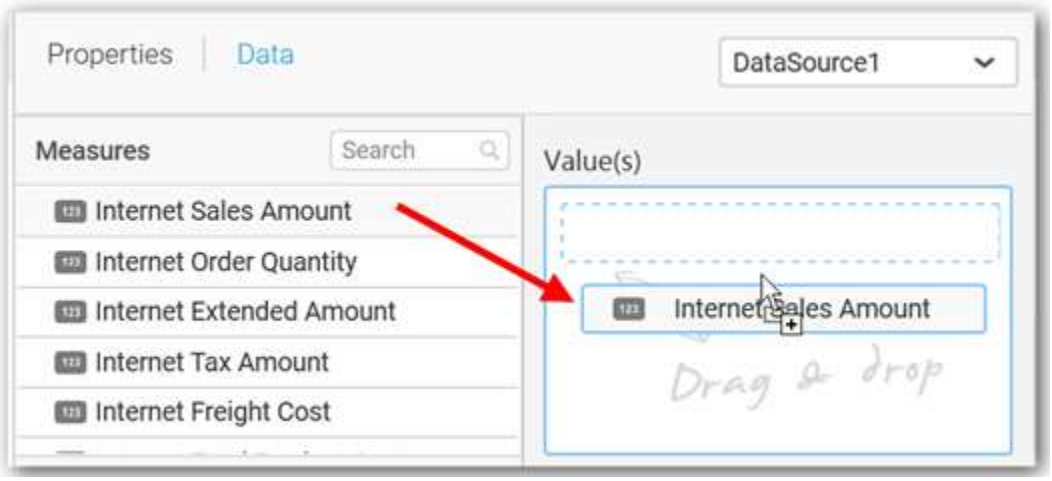
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions.

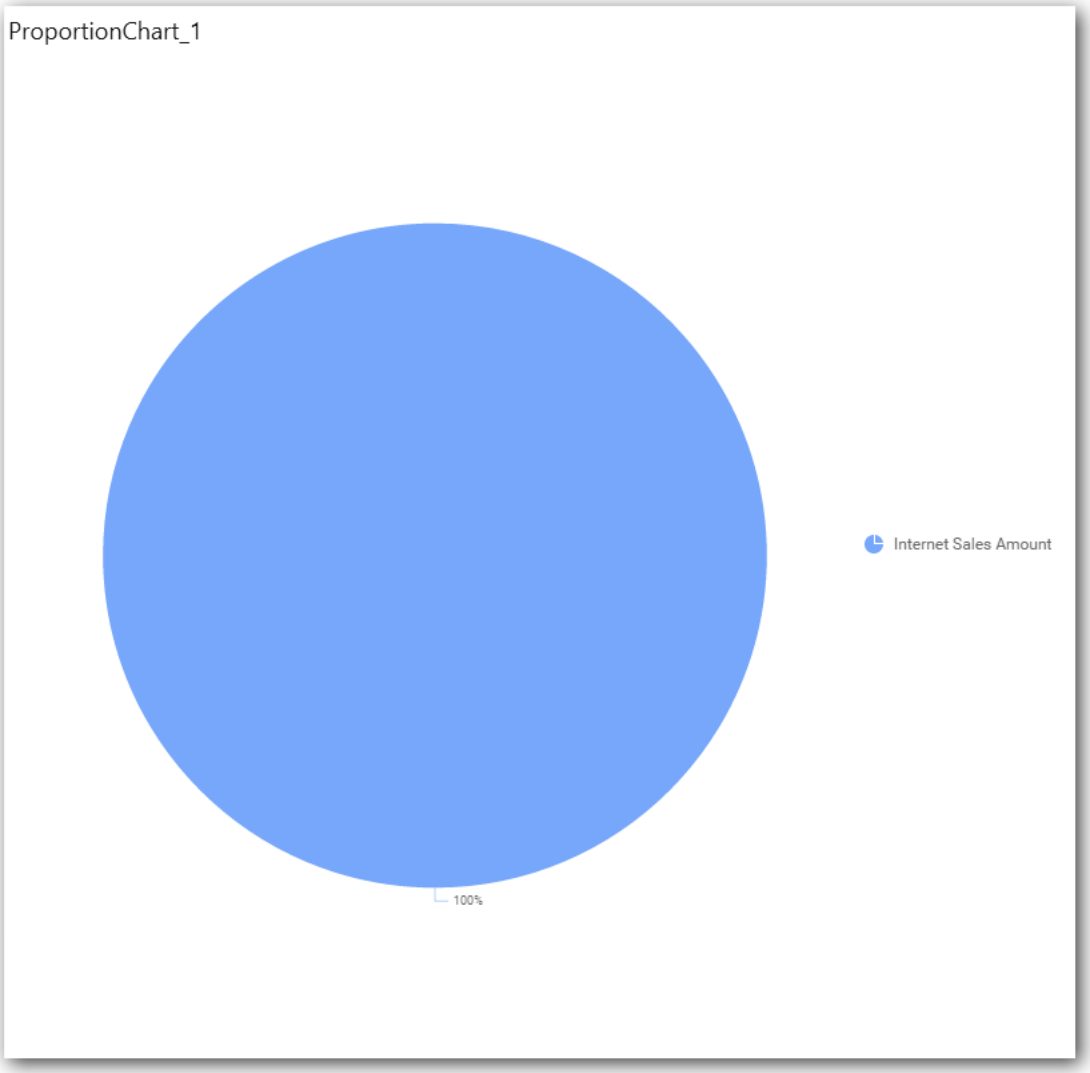


Assigning Value(s)

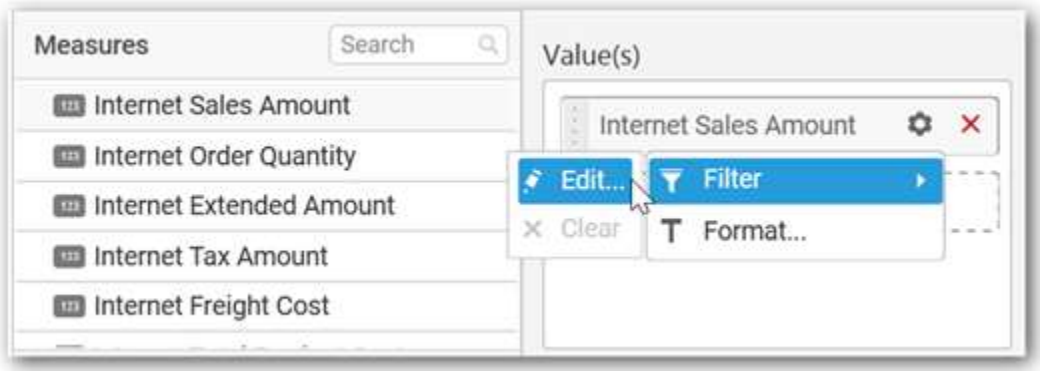
Drag and drop a column under Measures category into Value(s).



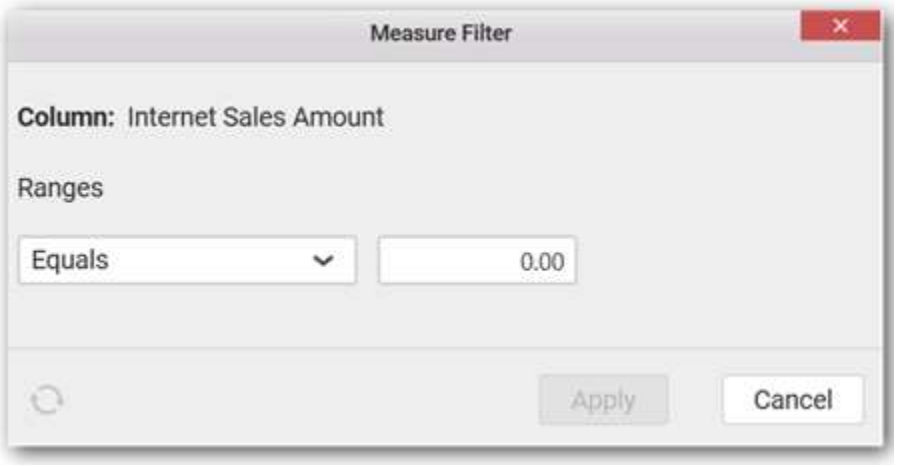
Now the chart will be rendered like this.



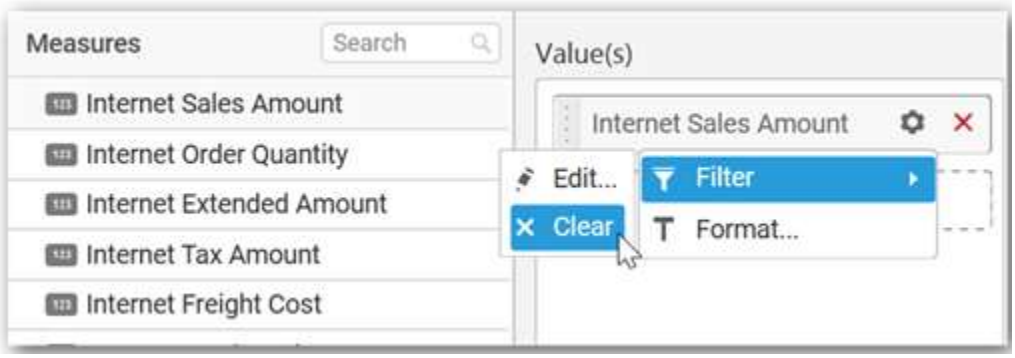
Define the filter criteria to match through choosing Edit option in Filter menu item.



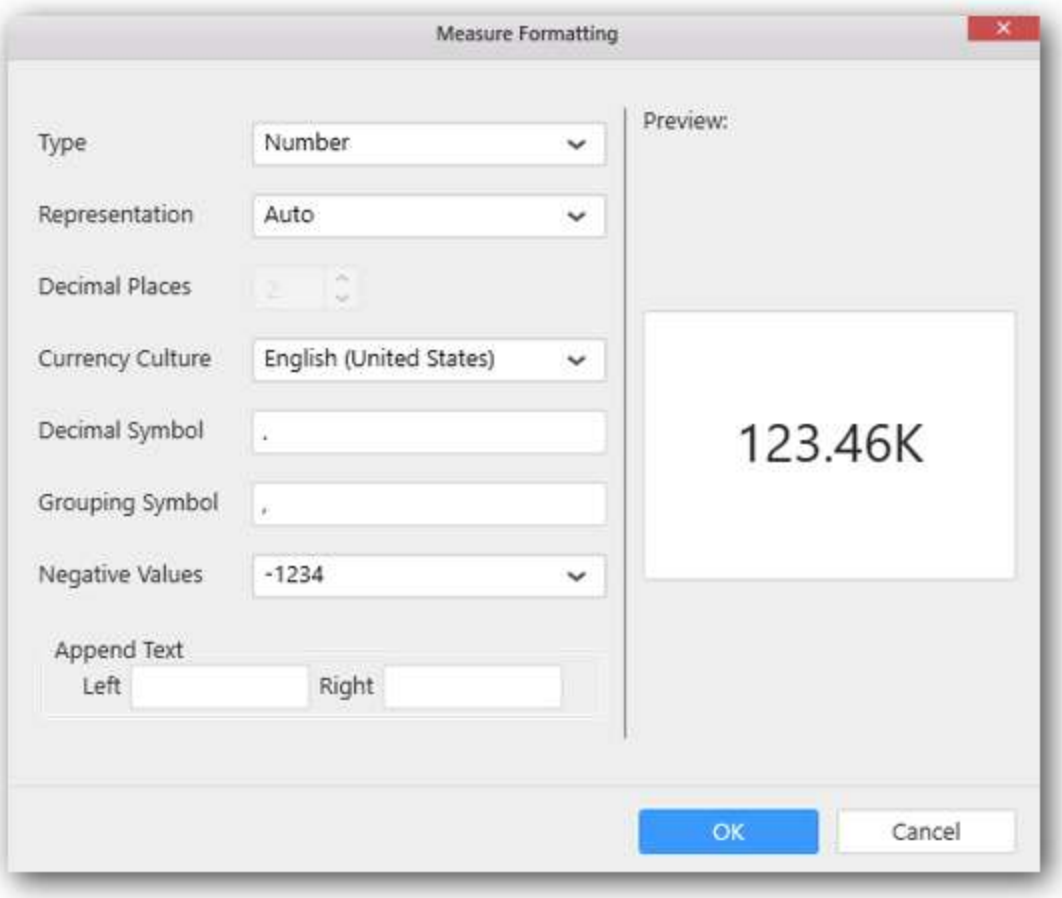
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



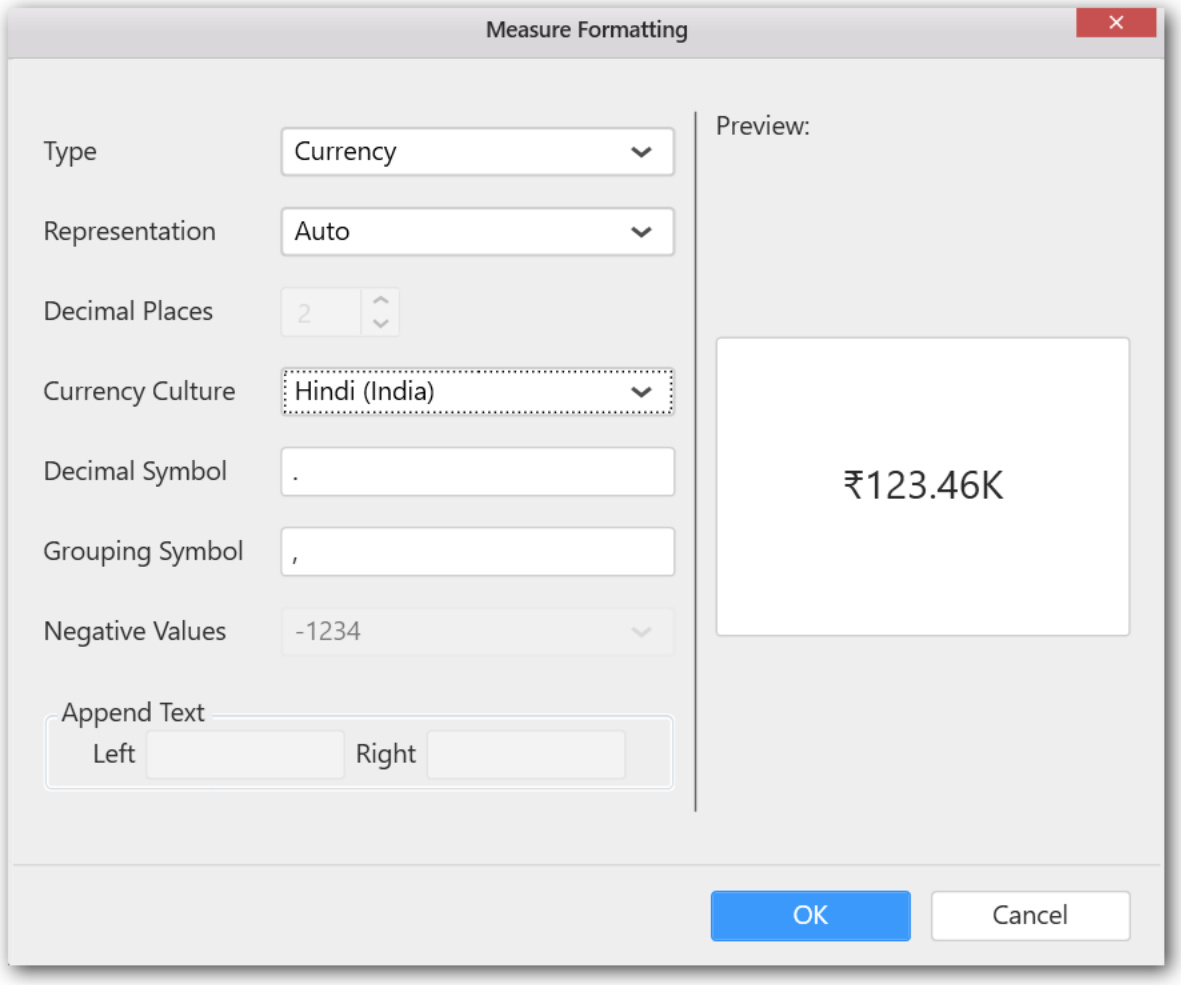
Select Clear option to clear the defined filter.



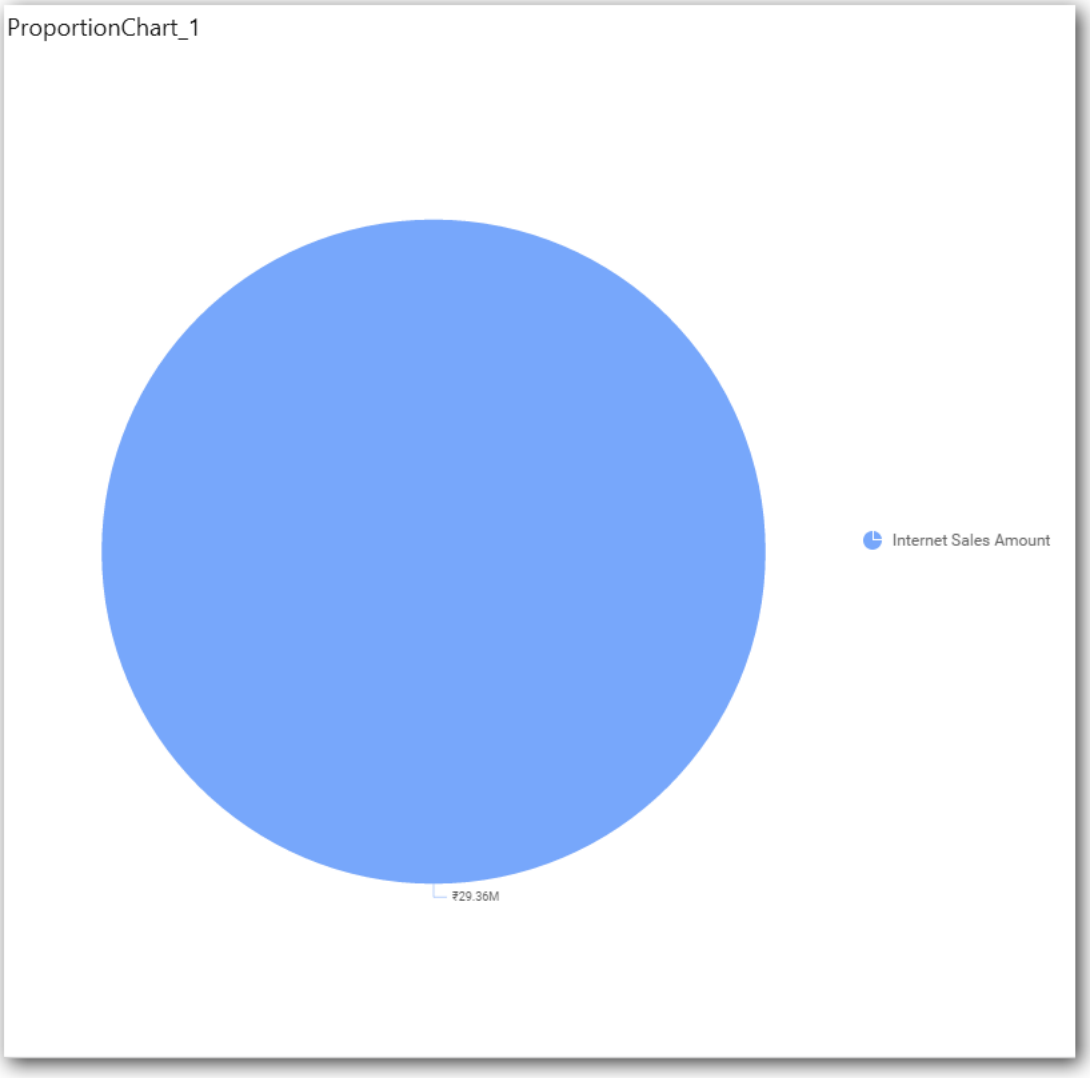
Select Format option to define the display format to the values in the column through Measure Formatting window.



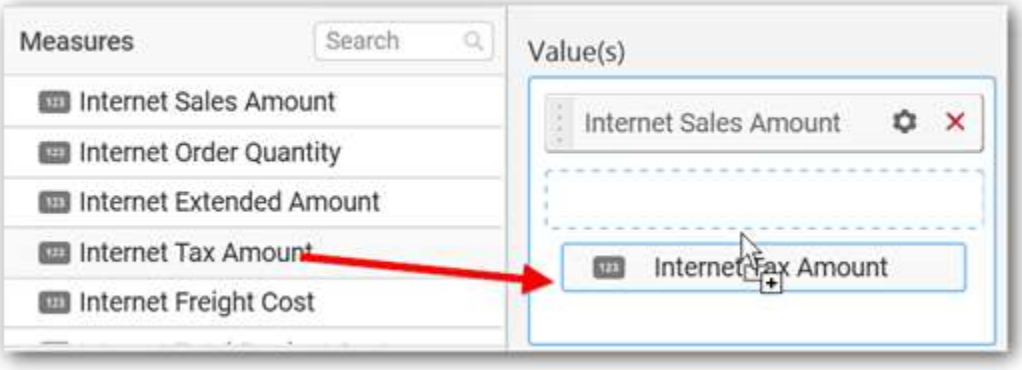
Choose the options you need and click OK.

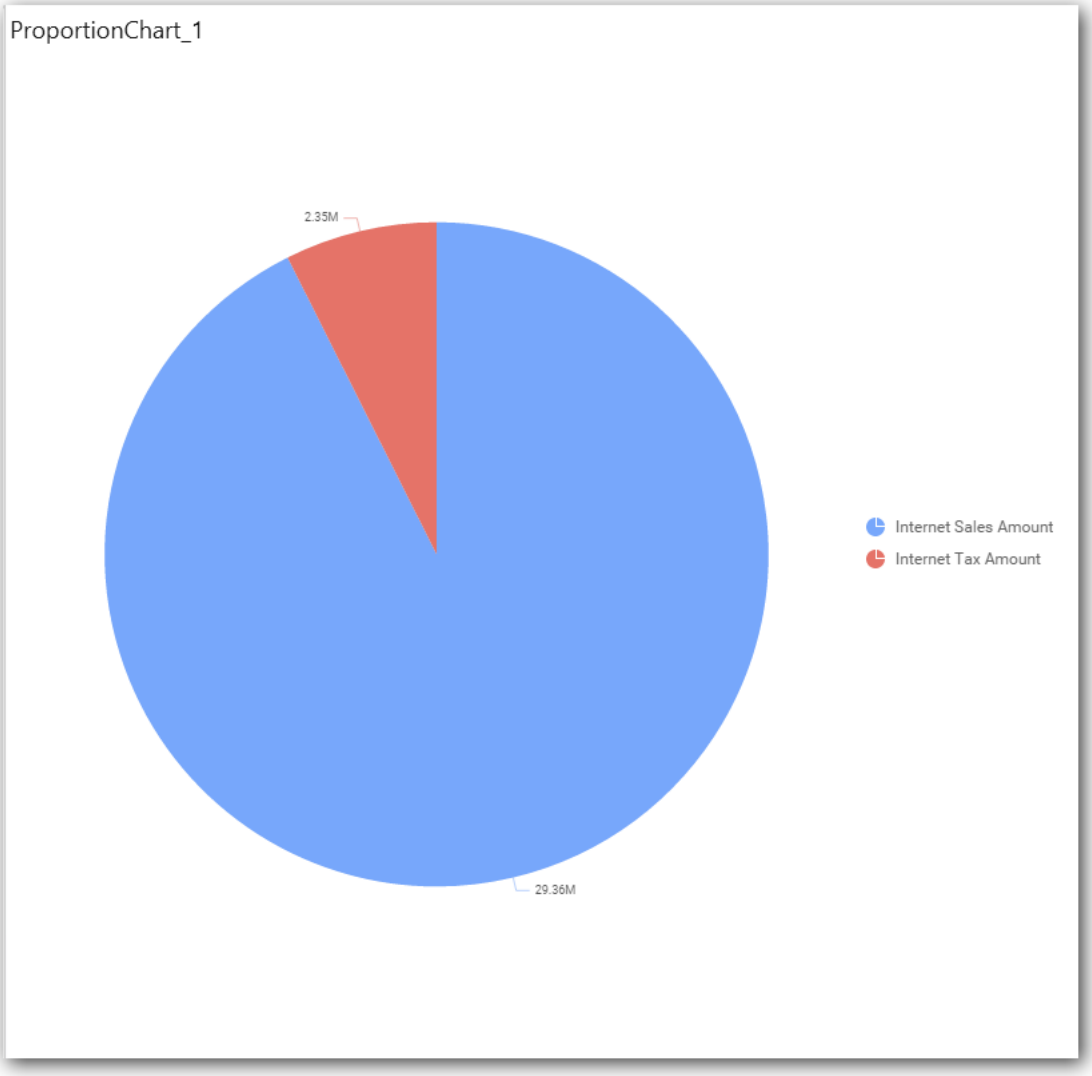


Now the Chart will be rendered like this.



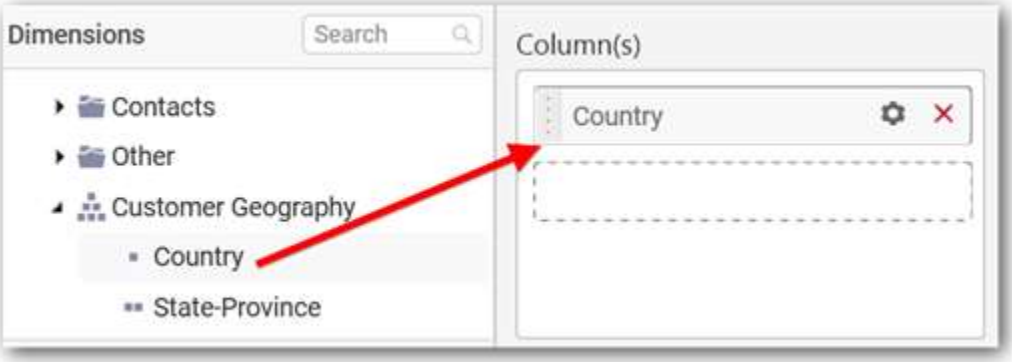
You can also add more than one column to the Value(s).

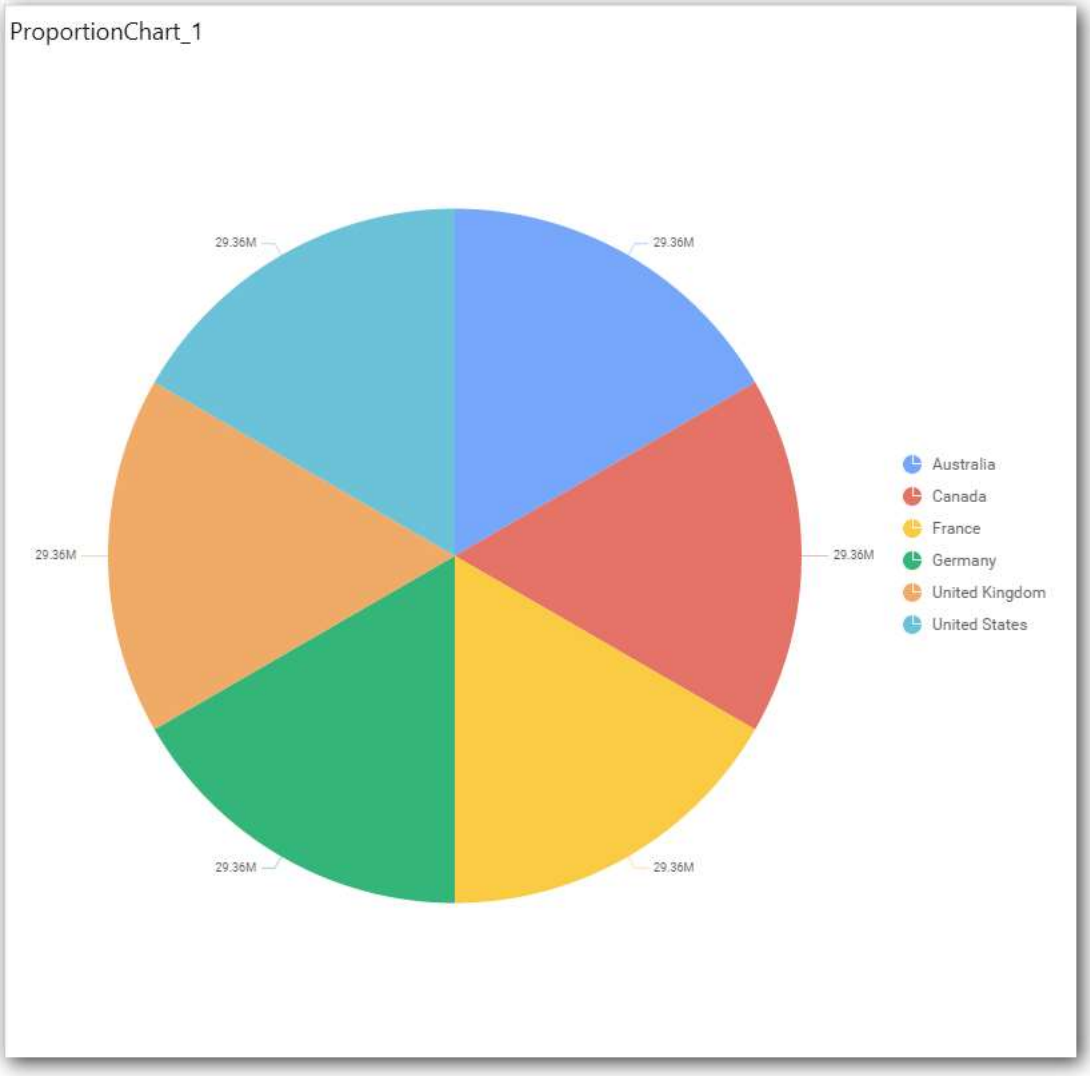




Assigning Column(s)

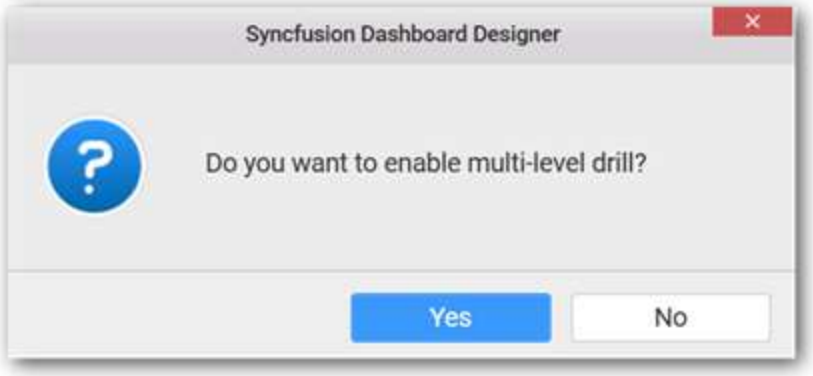
Add a dimension level or hierarchy into Column(s) section through drag and drop.





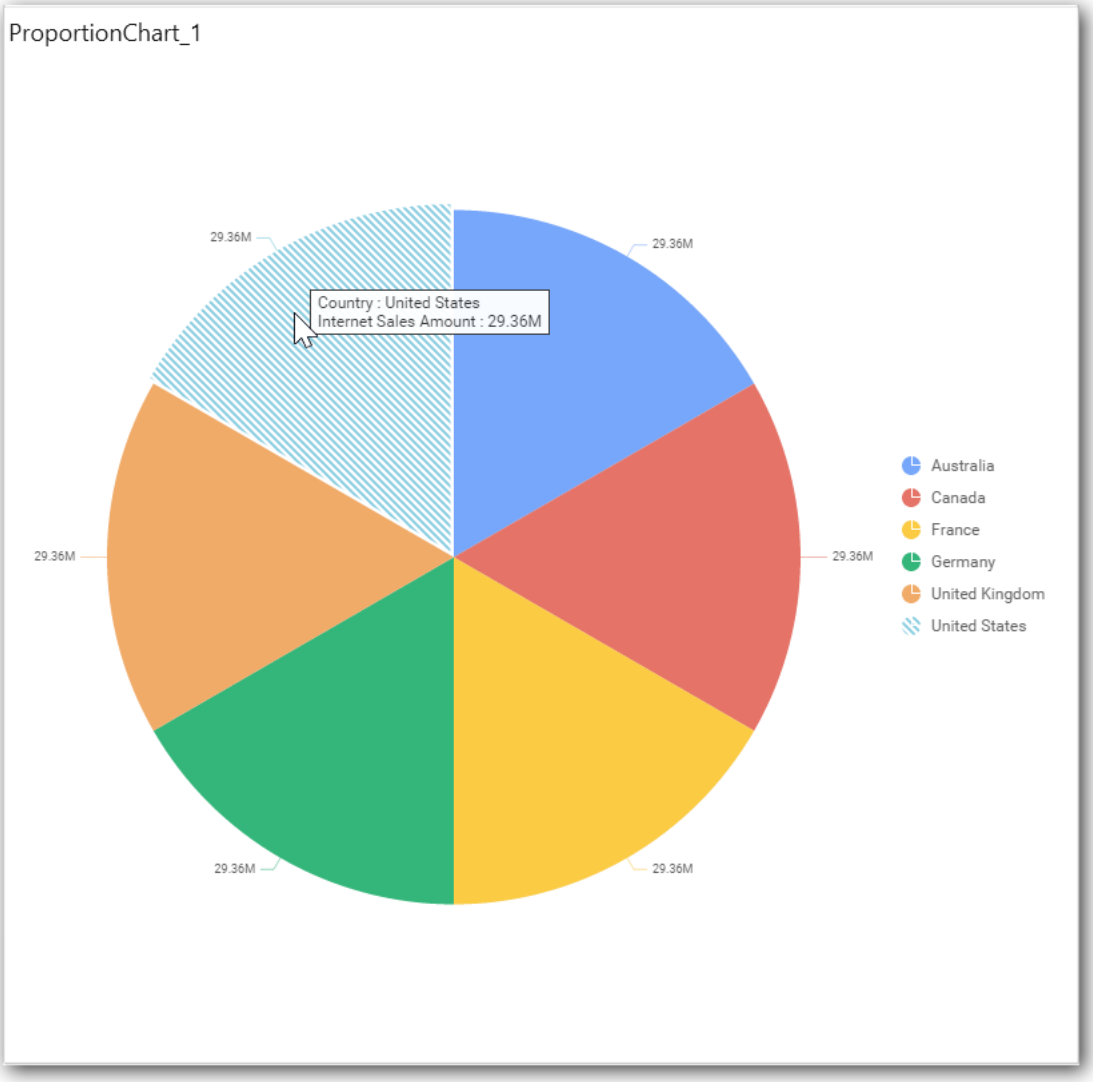
You may also add more than one column into Column(s) section.

In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

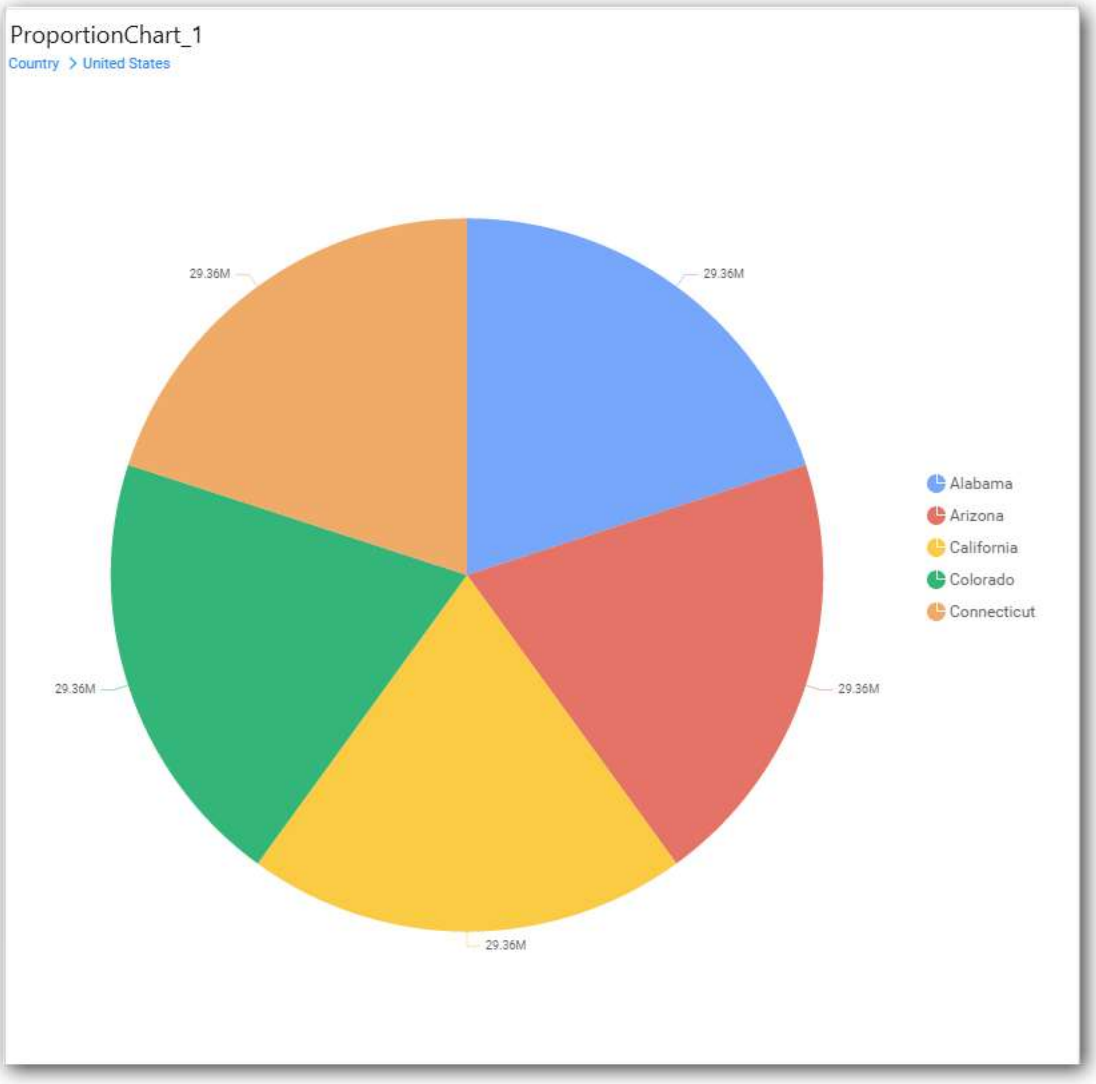


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

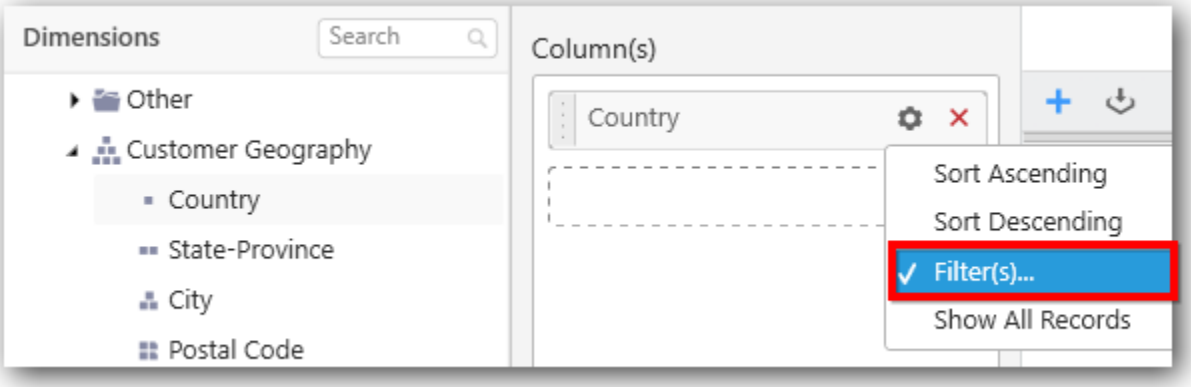
Click the respective data value marker in chart to drill into its inner level.



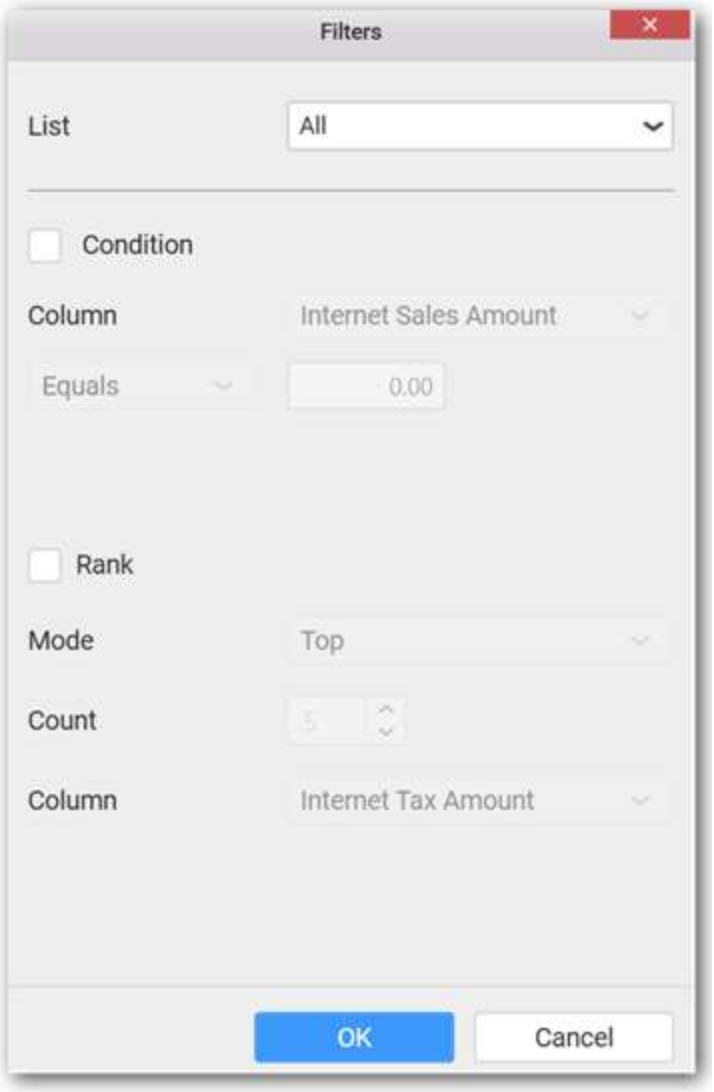
The drilled view of the chart is follows.



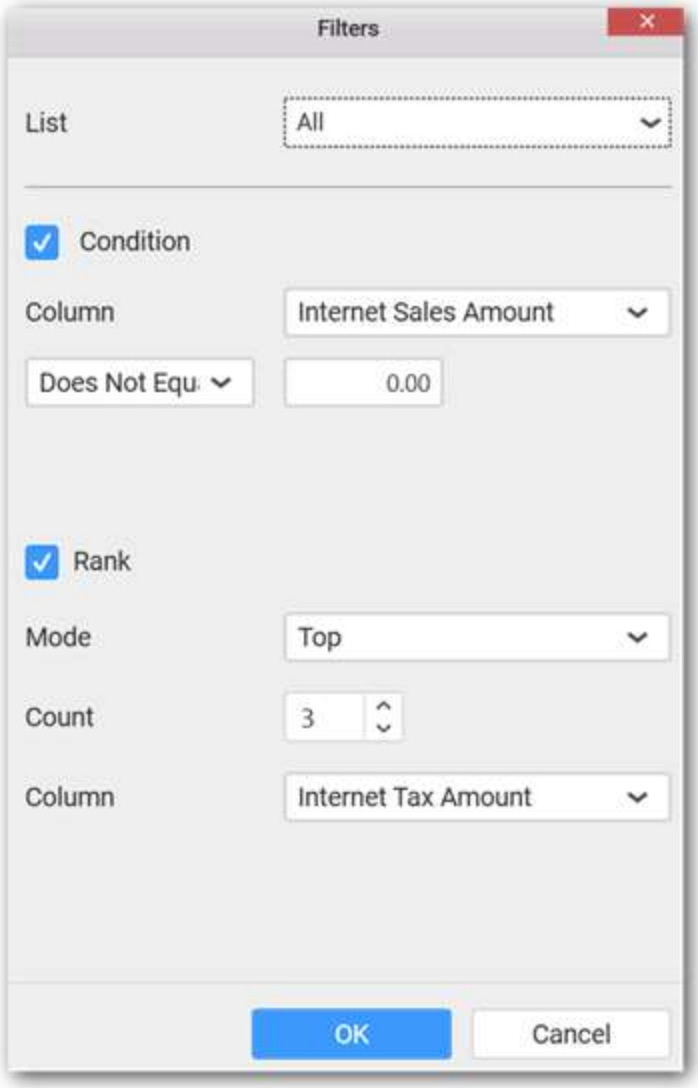
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



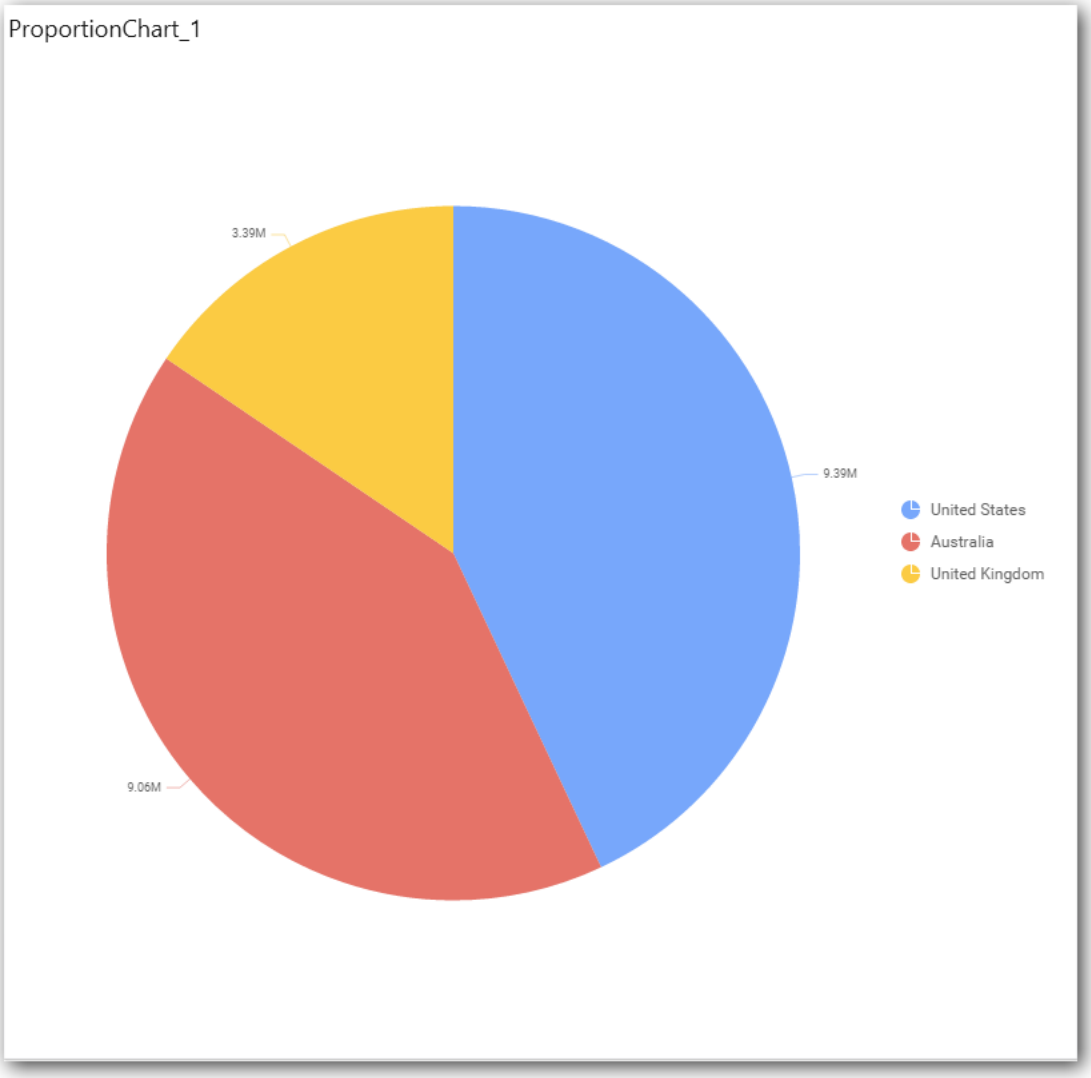
Select Filter(s)… to launch the Filters window.



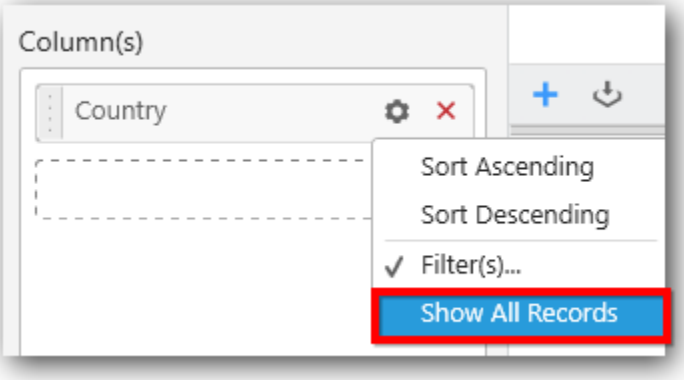
Define the filter condition and Rank and Click OK.



Now the chart will be rendered like this

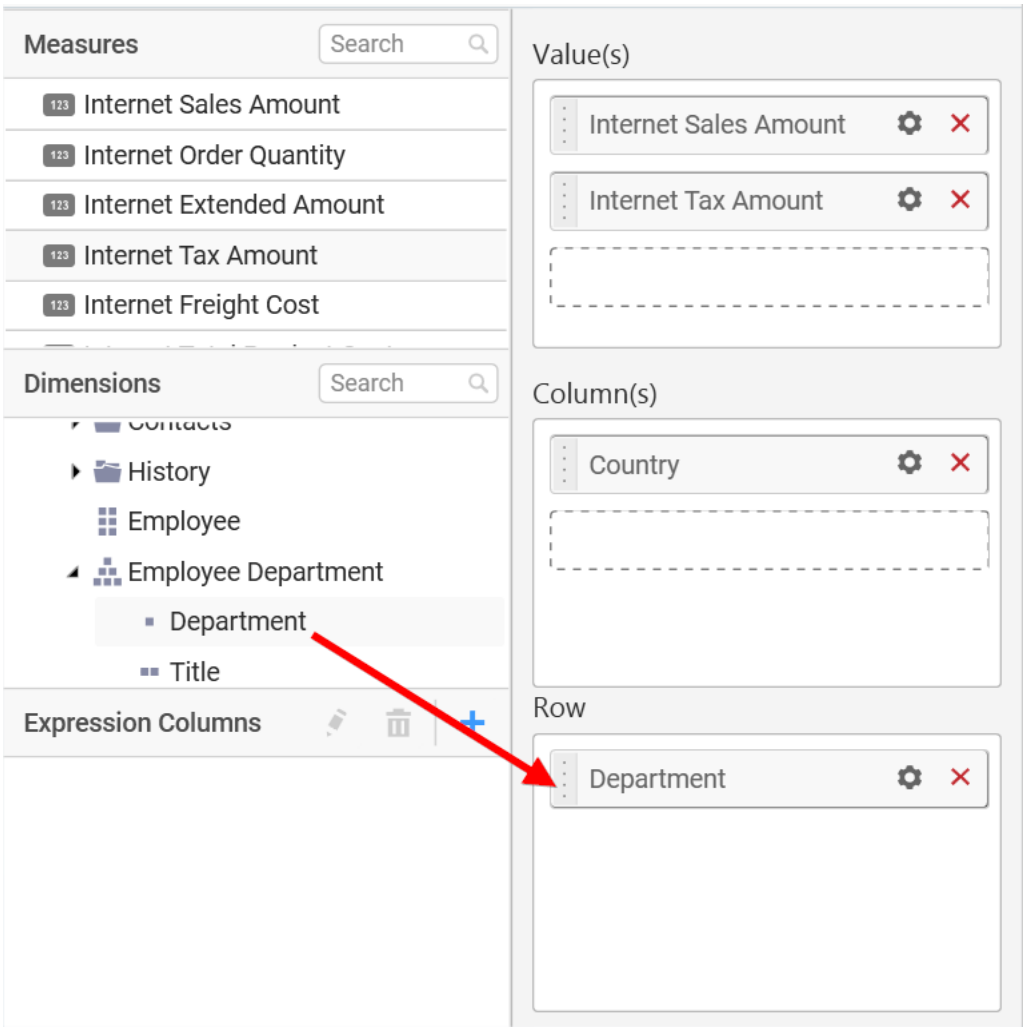


To show all records again click on Show All Records.

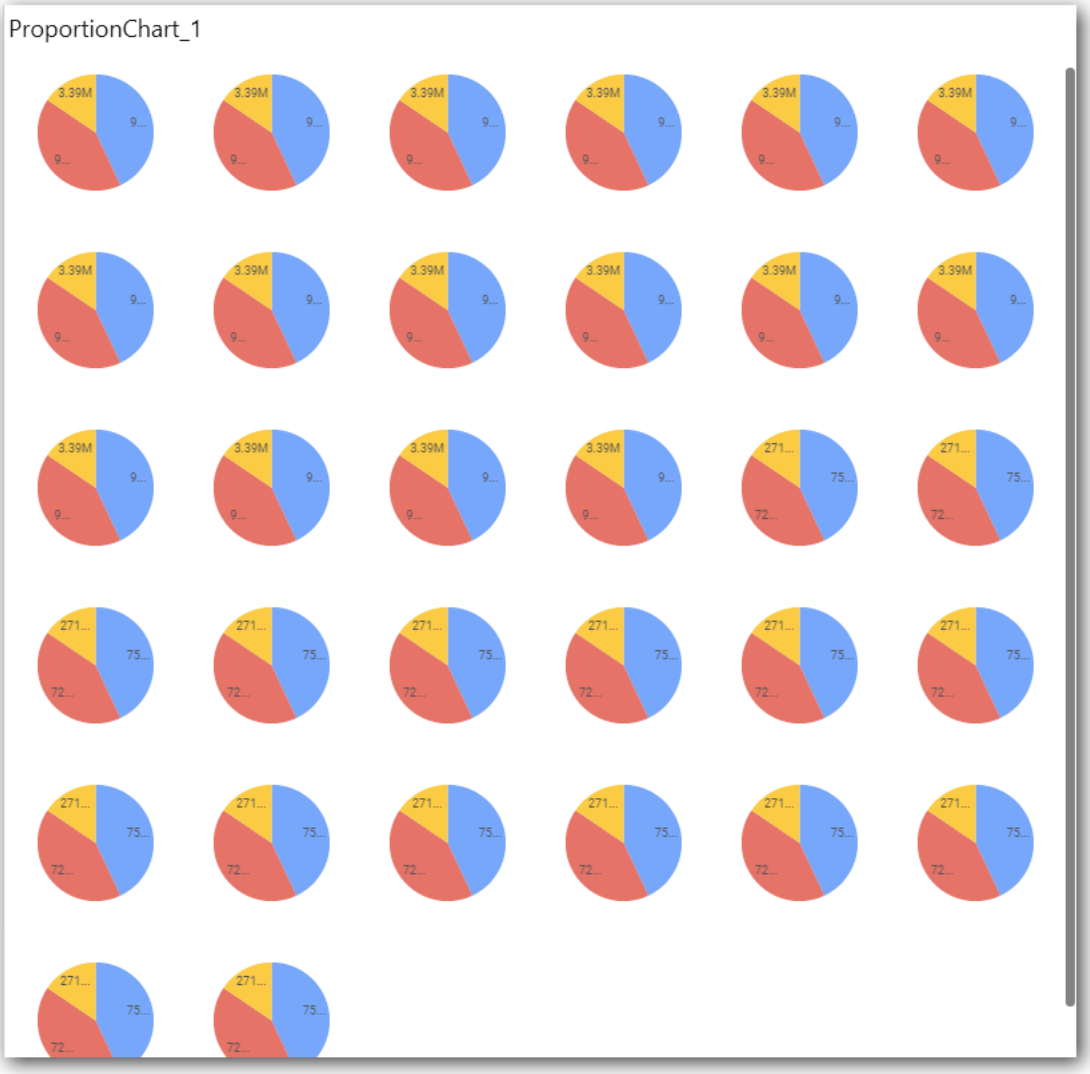


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart.



The chart will be rendered in series as shown in the image.



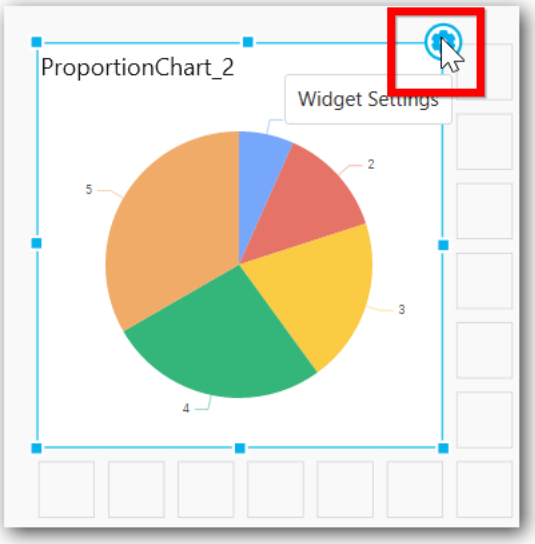
Scroll down to see all charts.

How to format Pie Chart?

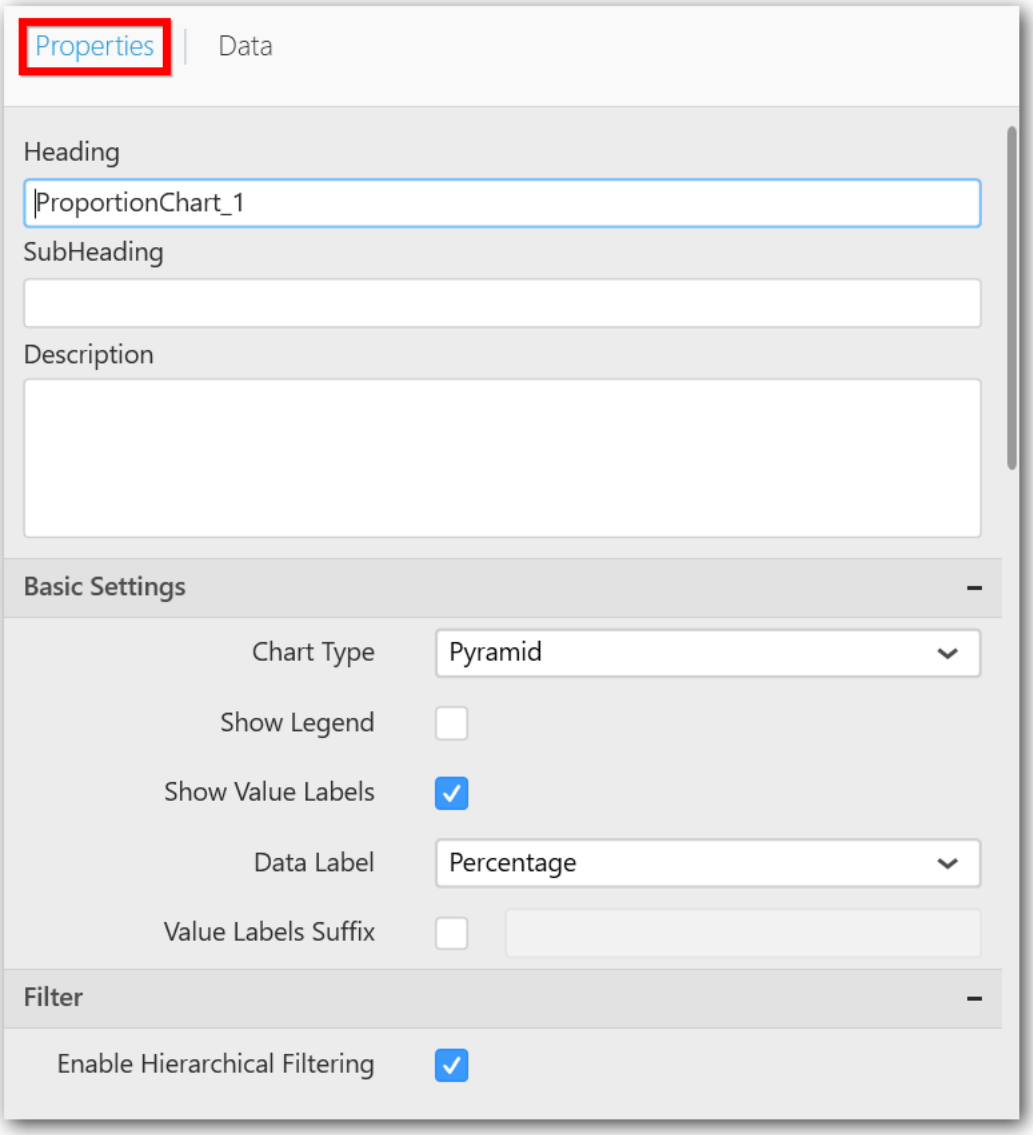
You can format the pie chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into pie chart follow the steps

1. Drag and drop the Pie chart into canvas and resize it to your required size.
2. Configure the data into Pie chart.
3. Focus on the Pie chart and Click on Widget Settings.

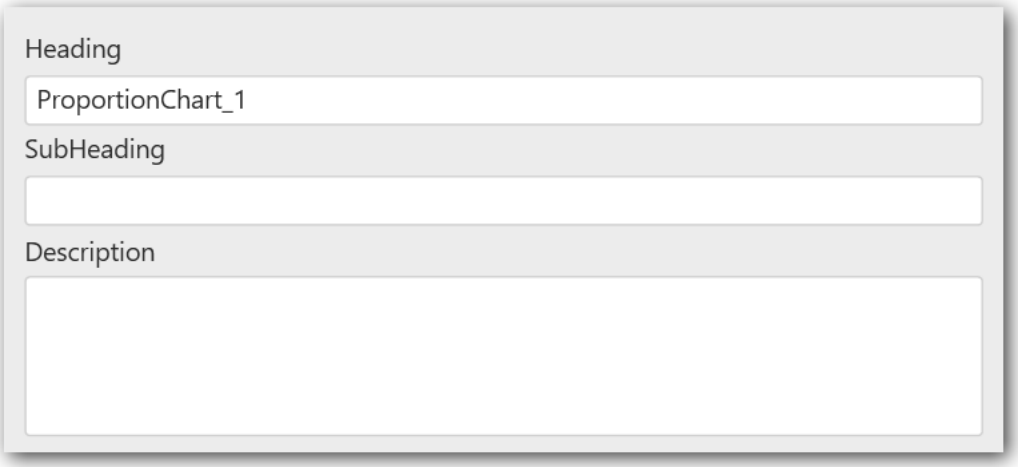


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this pie chart widget.

SubHeading

This allows you to set sub-title for this pie chart widget.

Description

This allows you to set description for this pie chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

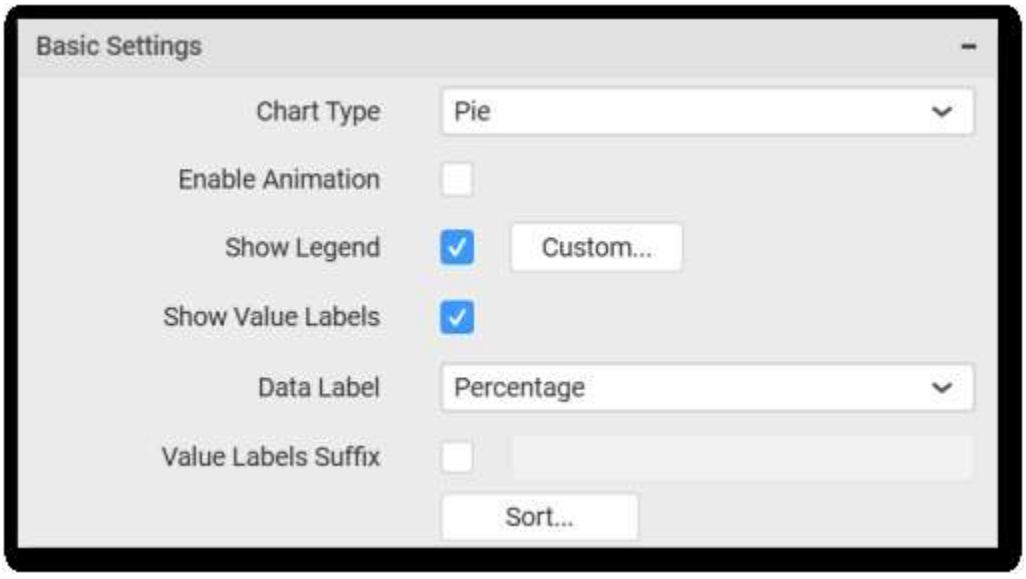


Chart Type

This allows you to switch the widget view from current chart type to another chart type.

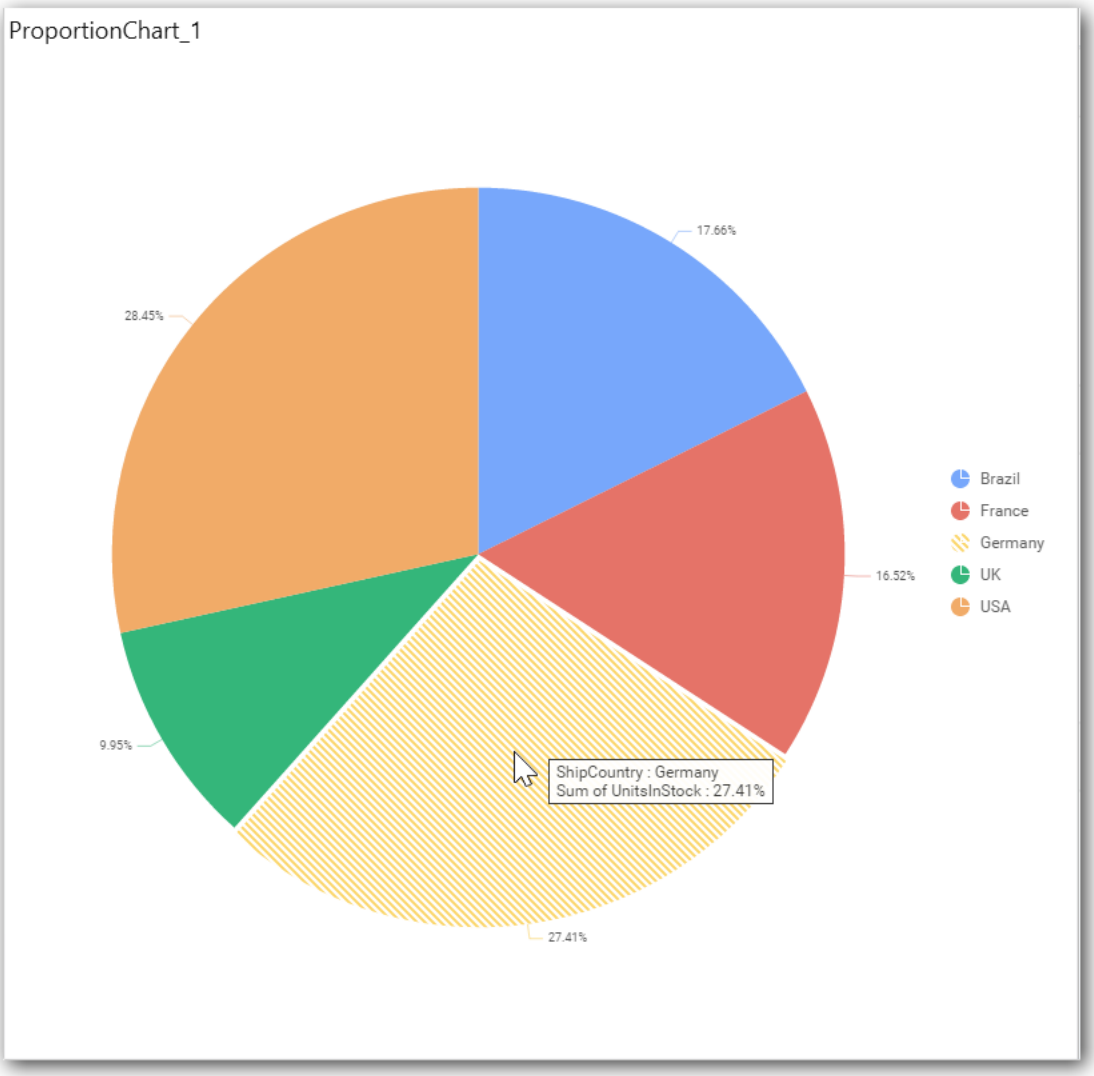
Enable Animation

This allows you to enable the series rendering in animated mode.

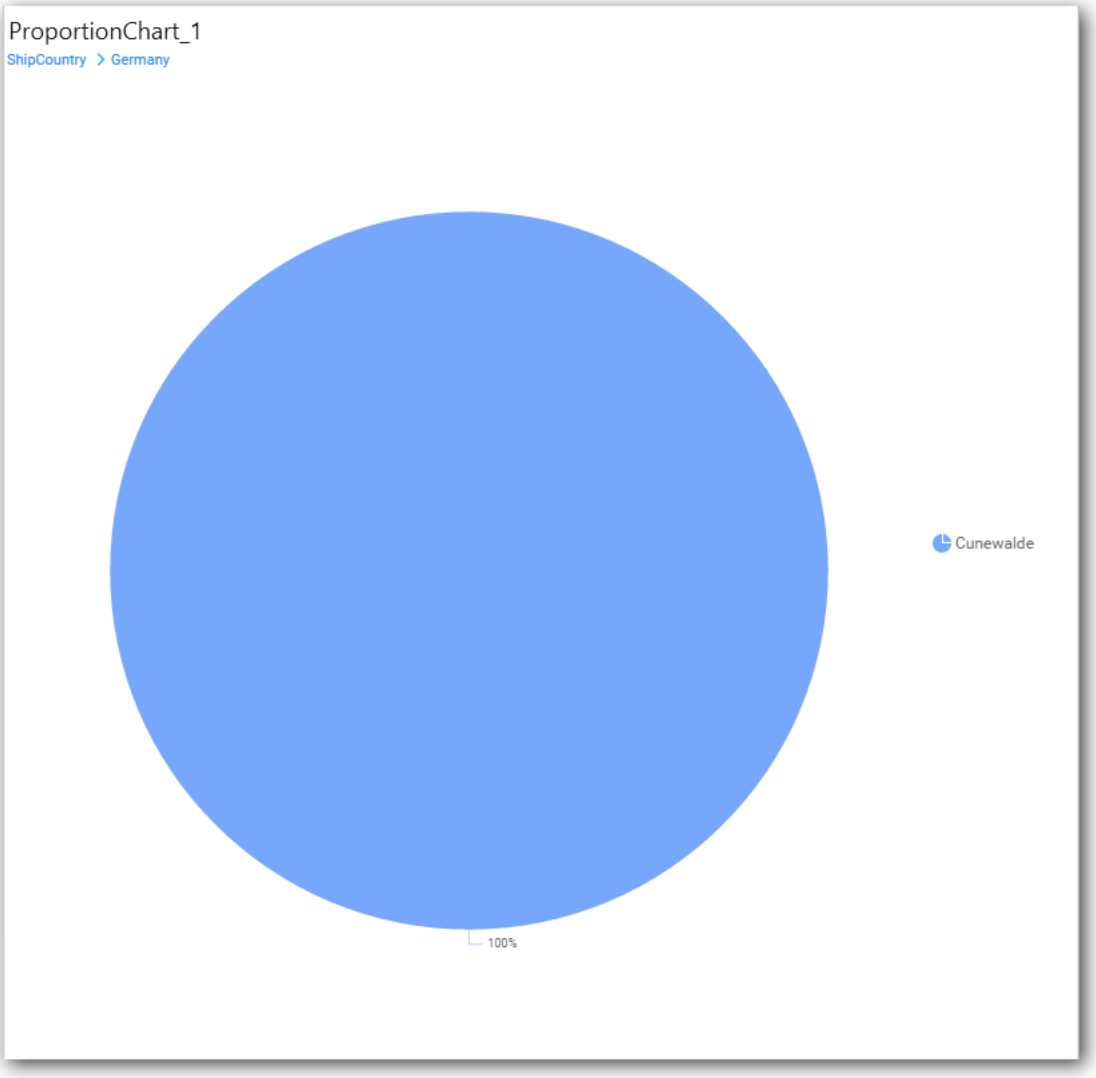
Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View:

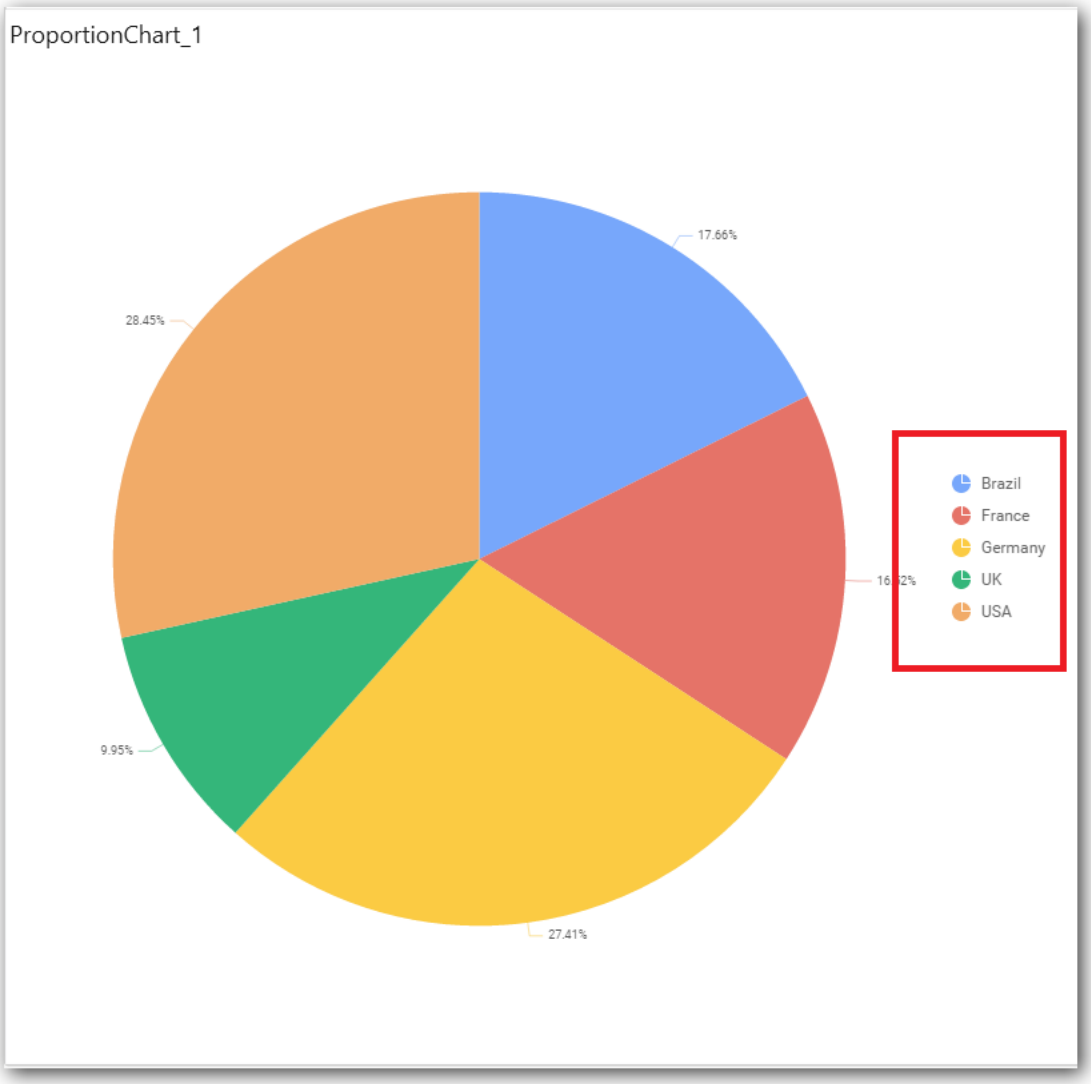


Drilled View



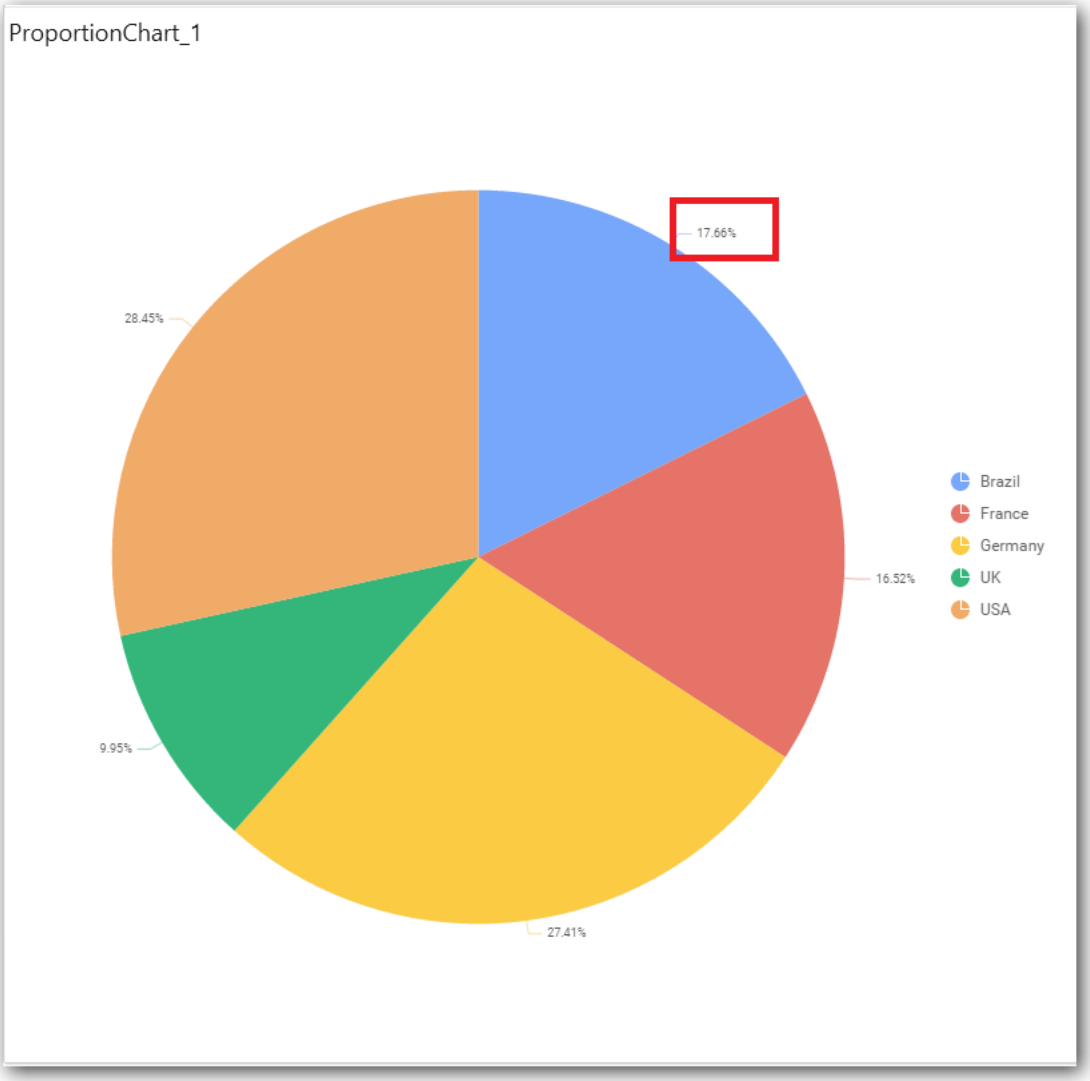
Show Legend

This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box). Enabling this option of Custom Legend Text will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.



Show Value Label

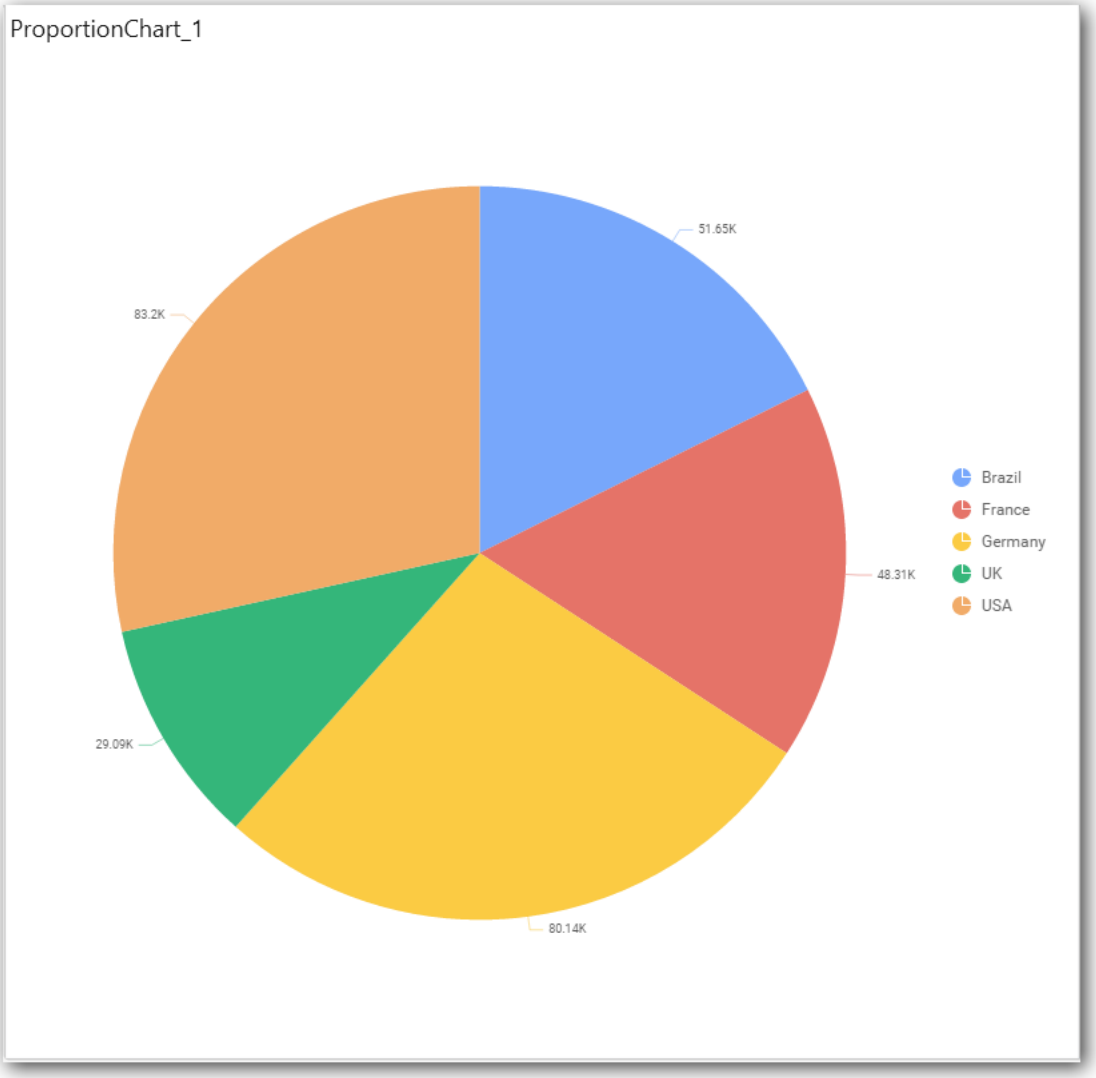
This allows you to toggle the visibility of value labels.



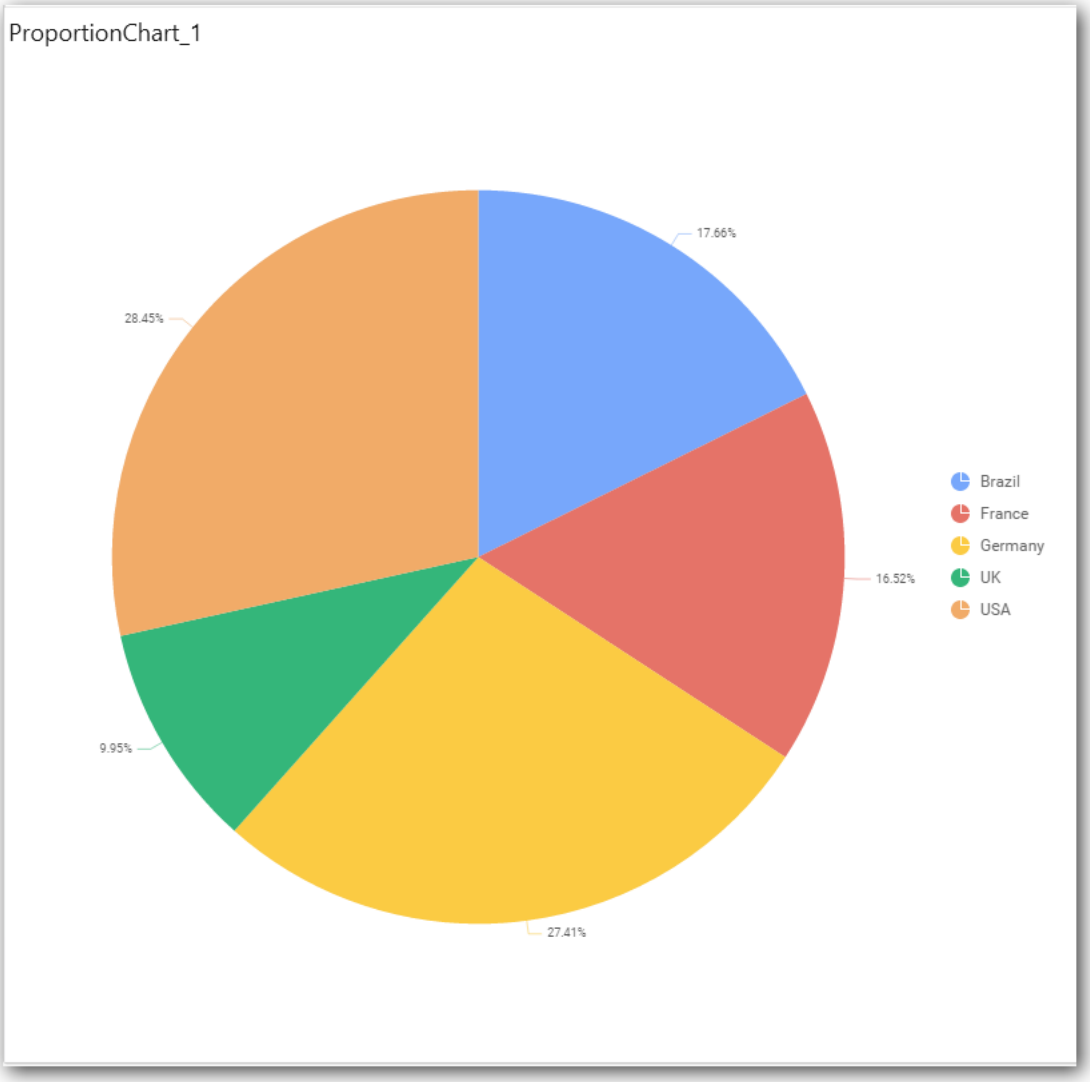
Data Label

This allows you to define the display format either as value or as percentage.

Value

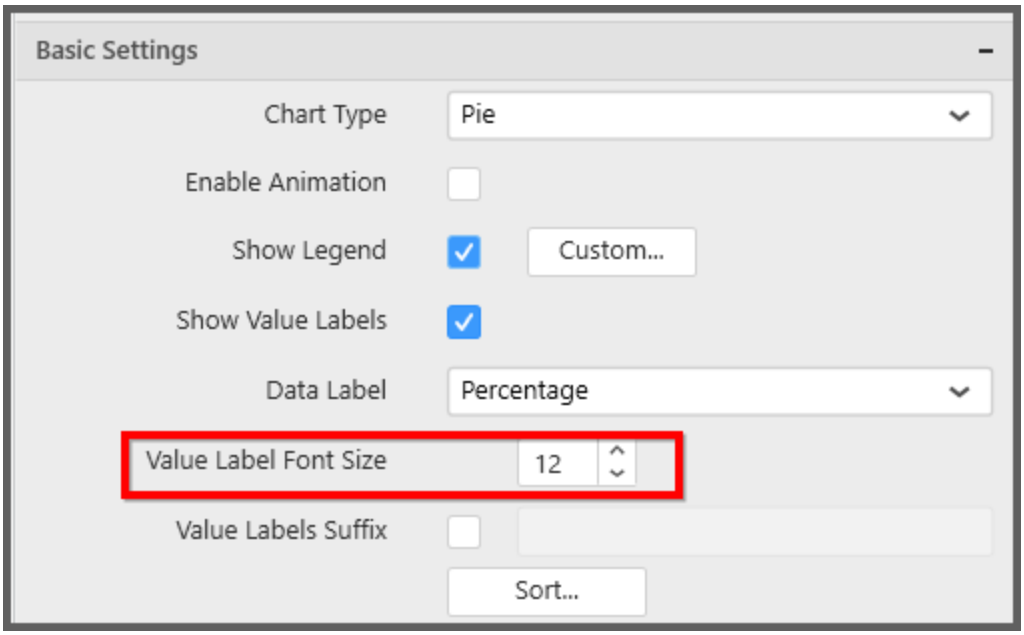


Percentage



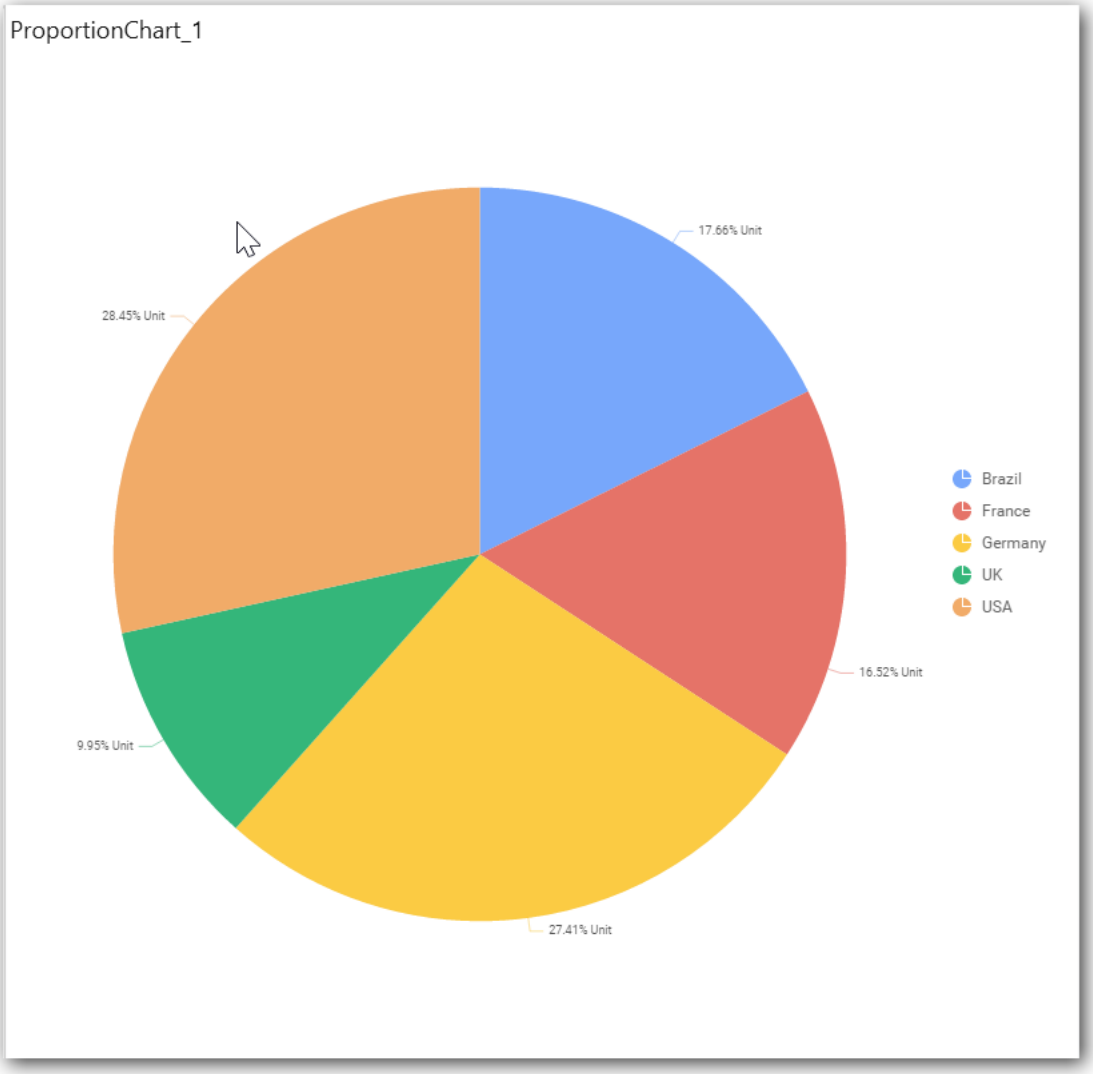
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.



Value Labels Suffix

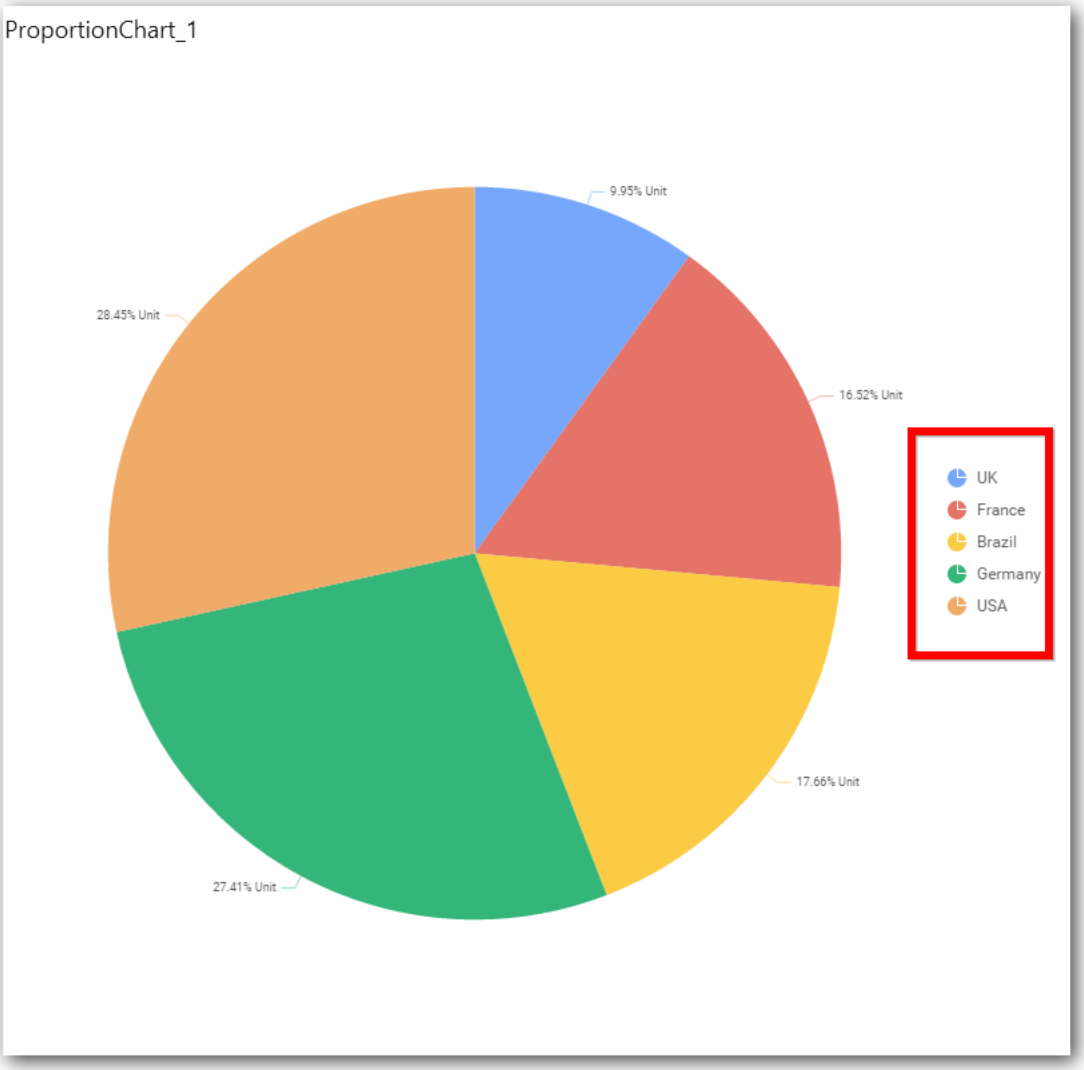
Allows you to set suffix to the value labels.



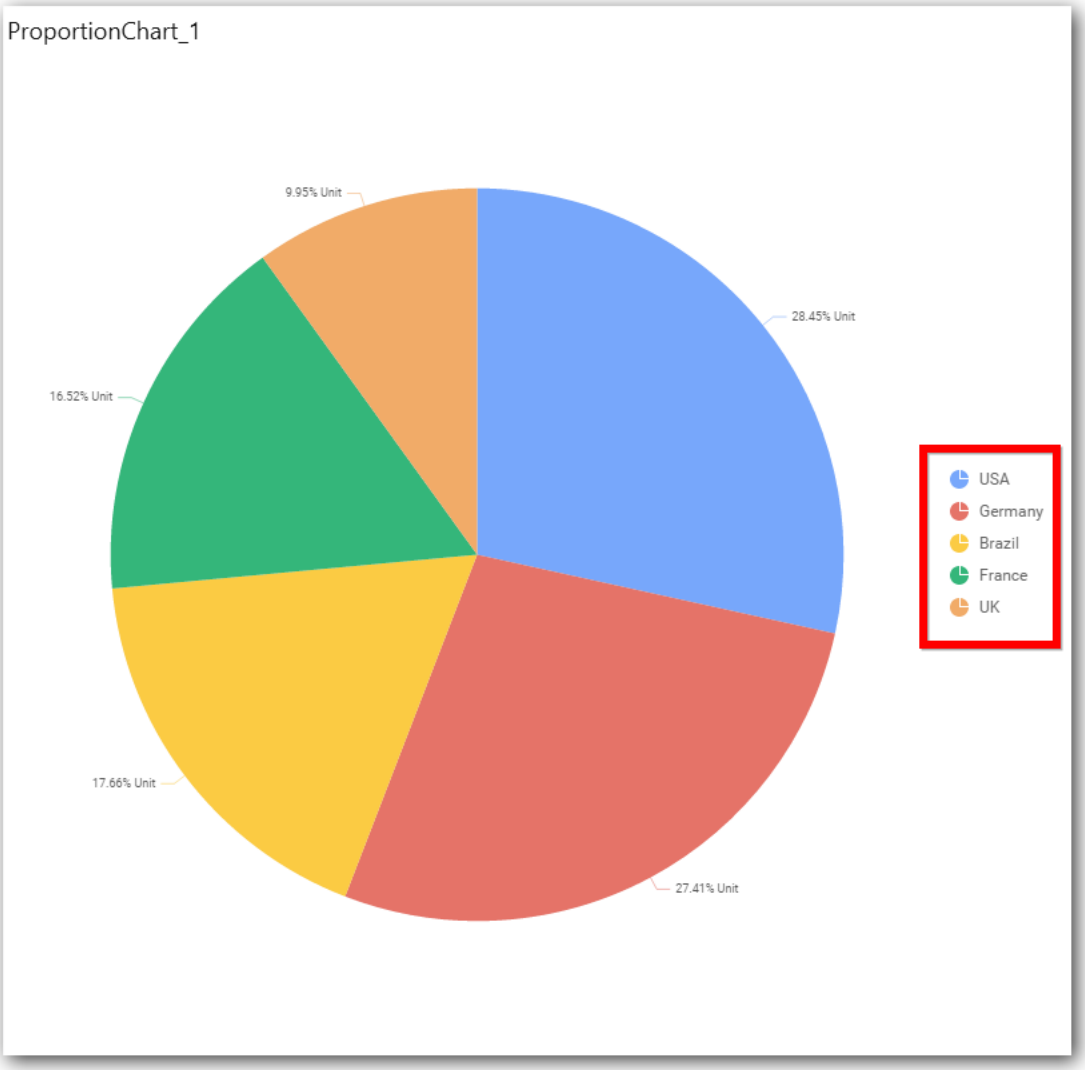
Sort Order

This allows you to define the sort order for each measure column added.

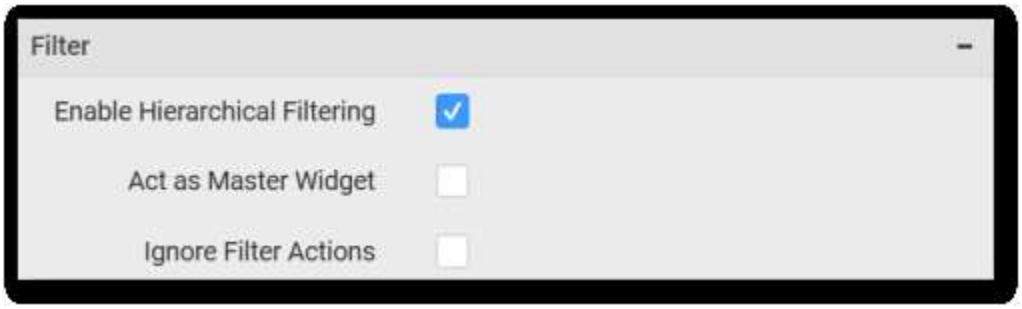
Ascending



Descending



Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

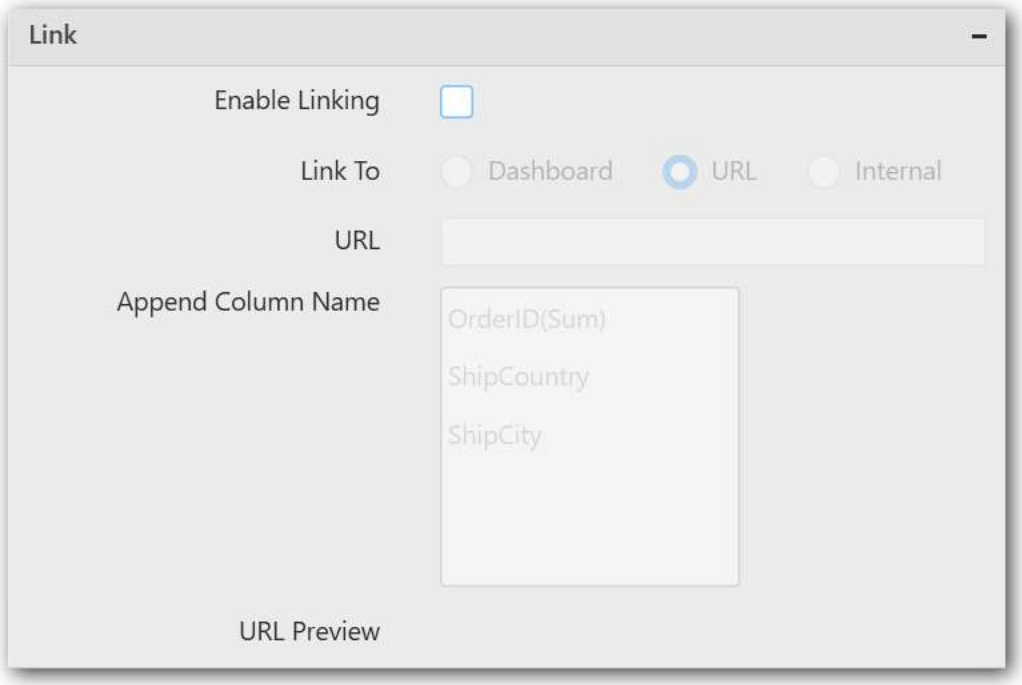
Act as Master Widget

This allows you to define this pie chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

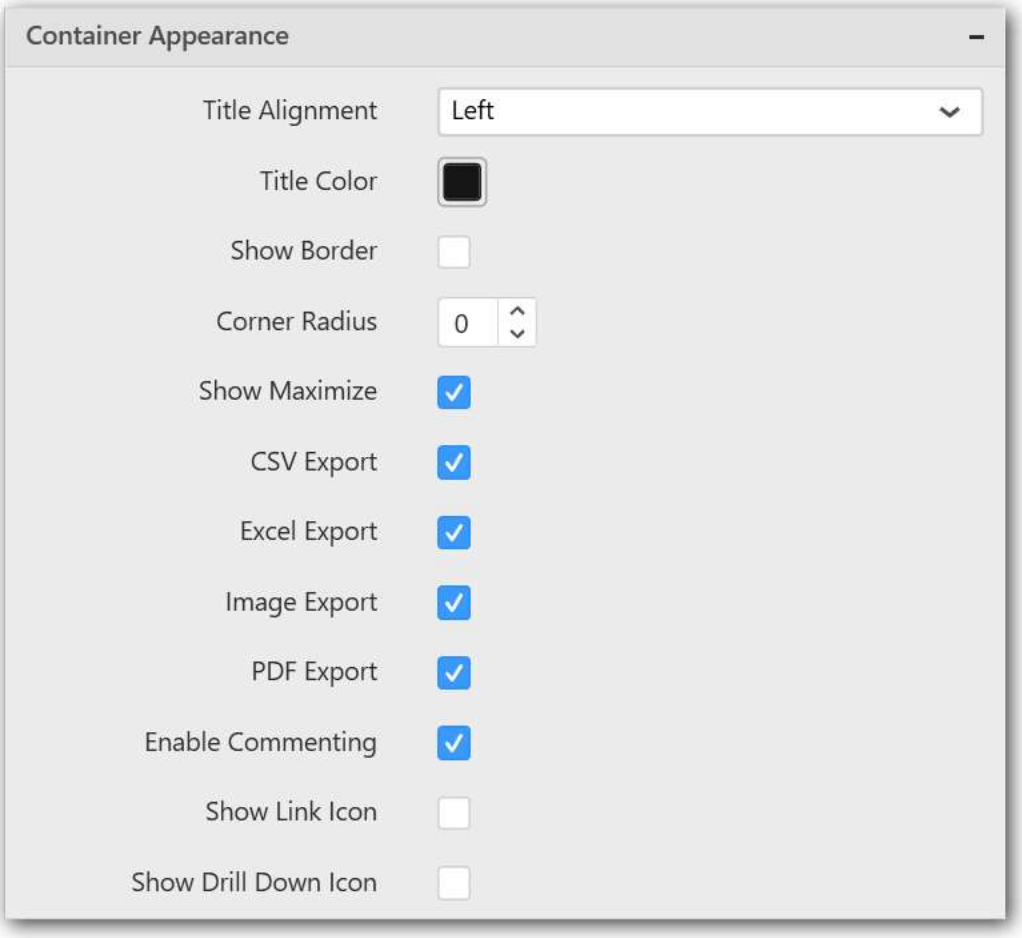
This allows you to define this pie chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this pie chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this pie chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this pie chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

Image Export

This allows you to enable/disable the image export option for this pie chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

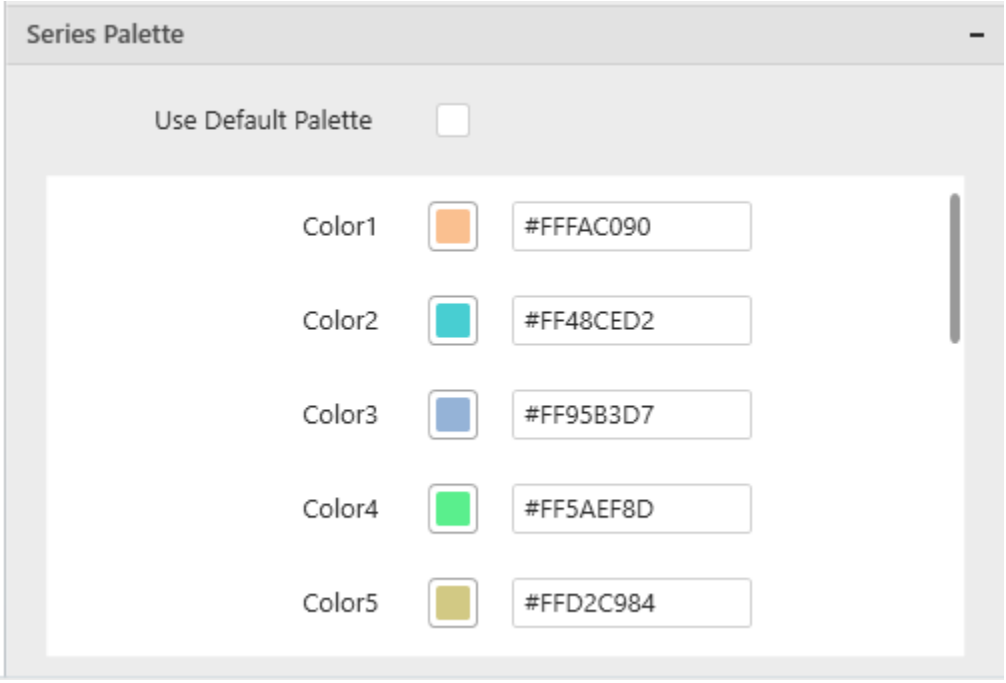
This allows you to enable/disable comment for dashboard widget. For more details refer here

Series Palette

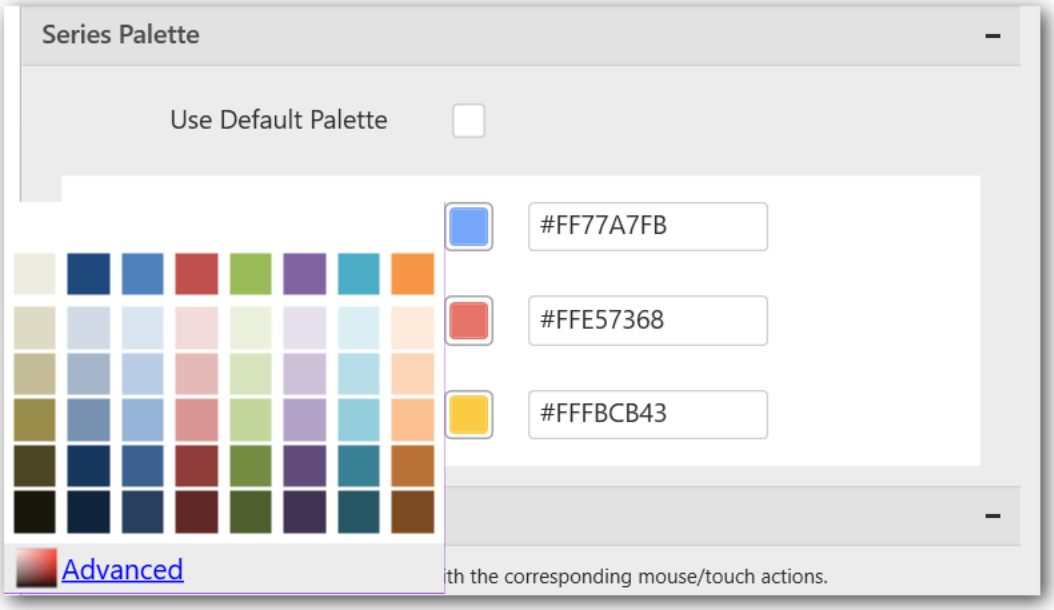
This allows you to customize the chart series color through Series Palette section.

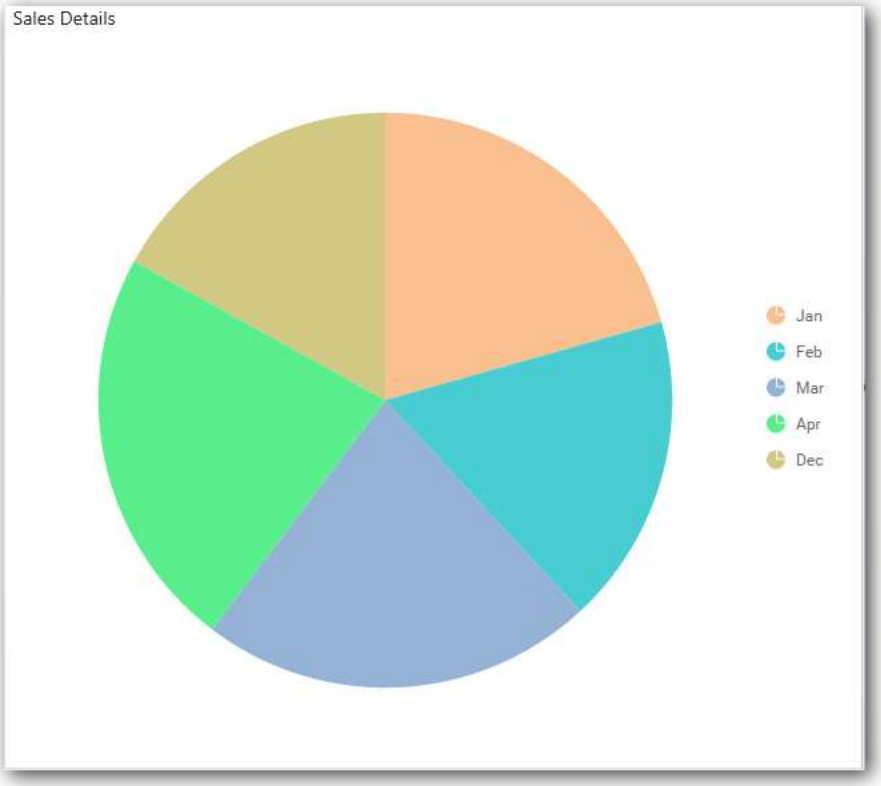
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to proportion series segments.



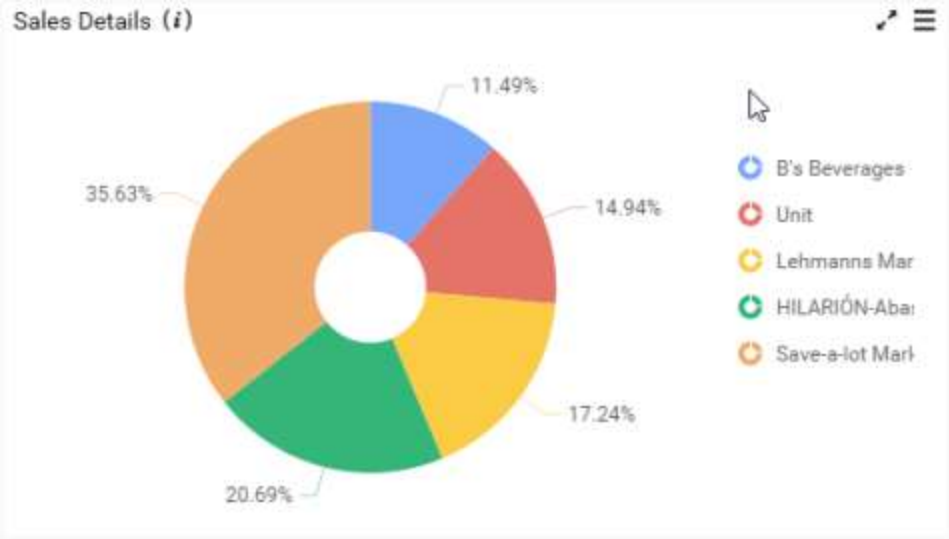
By toggle off the Use Default Palette, you can customize the proportion series segments' colors. This section shows a palette of colors. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Doughnut Chart

Doughnut Chart allows you to showcase proportionality of each item to the total in the form of donut-slices. To plot a doughnut chart, a minimum requirement of 1 value and 1 column is needed.

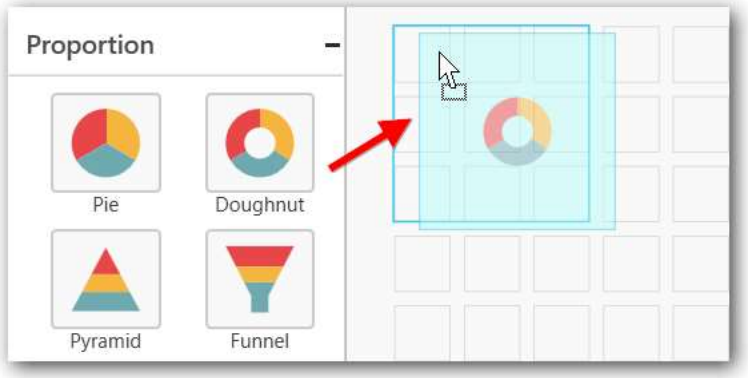


How to configure the flat table data to Doughnut Chart?

Doughnut Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

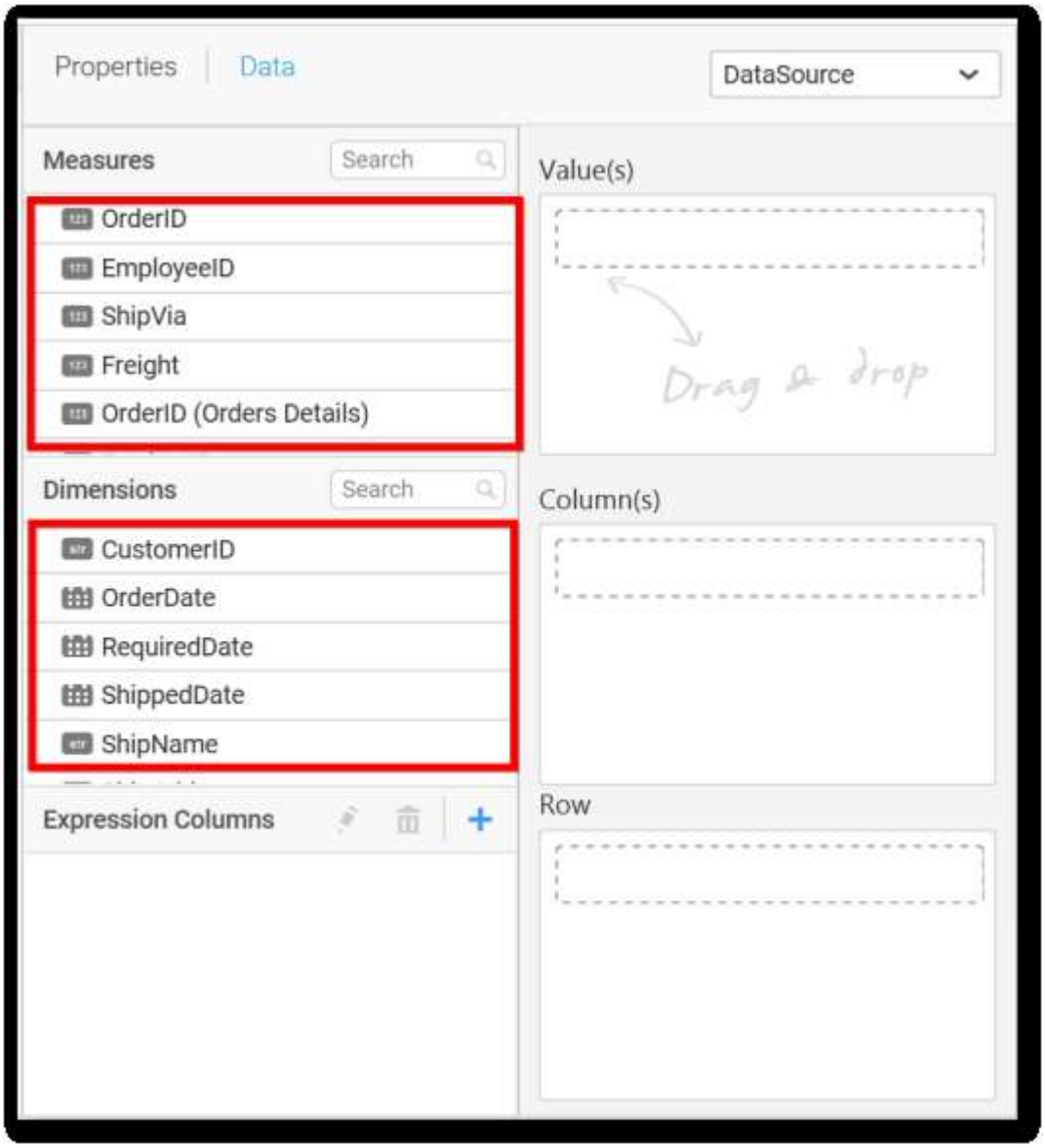
To configure data into Doughnut chart follow the steps

- Drag and drop the Doughnut chart into canvas and resize it to your required size.



- Connect to the Data source.
- Focus on the Doughnut chart and Click on Assign Data

The data pane will be opened with available Measures and Dimensions from the connected data source.

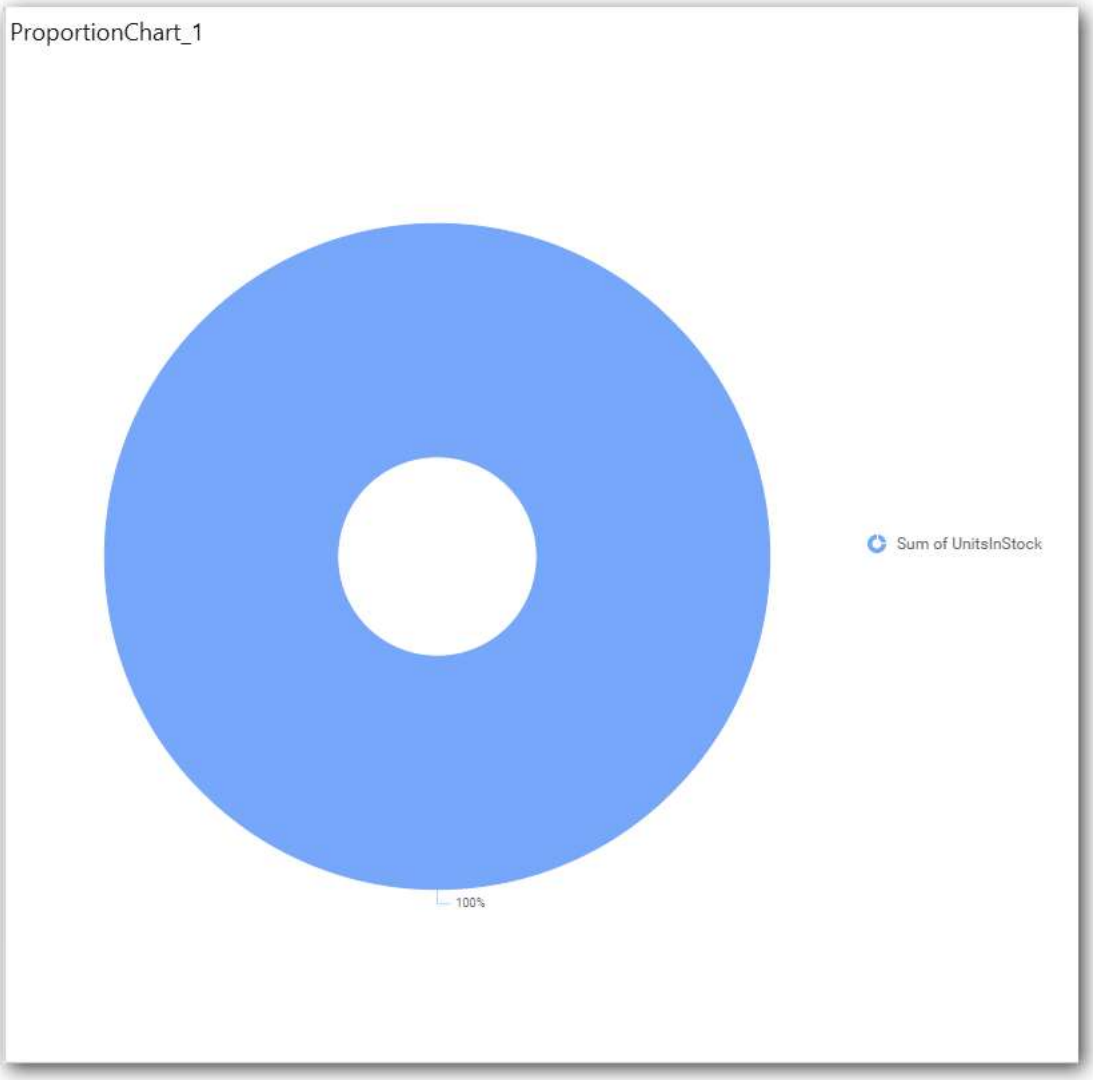


You can add the required data from Measures and Dimensions into required field, you can also create Expression Columns using Expression Designer.

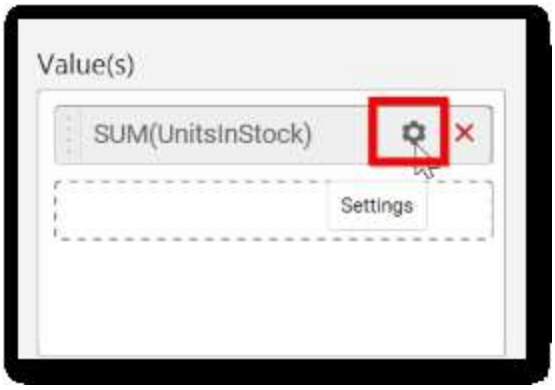
Adding Value(s)

You can add Measures into Value(s) field by drag and drop the required measure.

Now the Doughnut chart will be rendered like this.

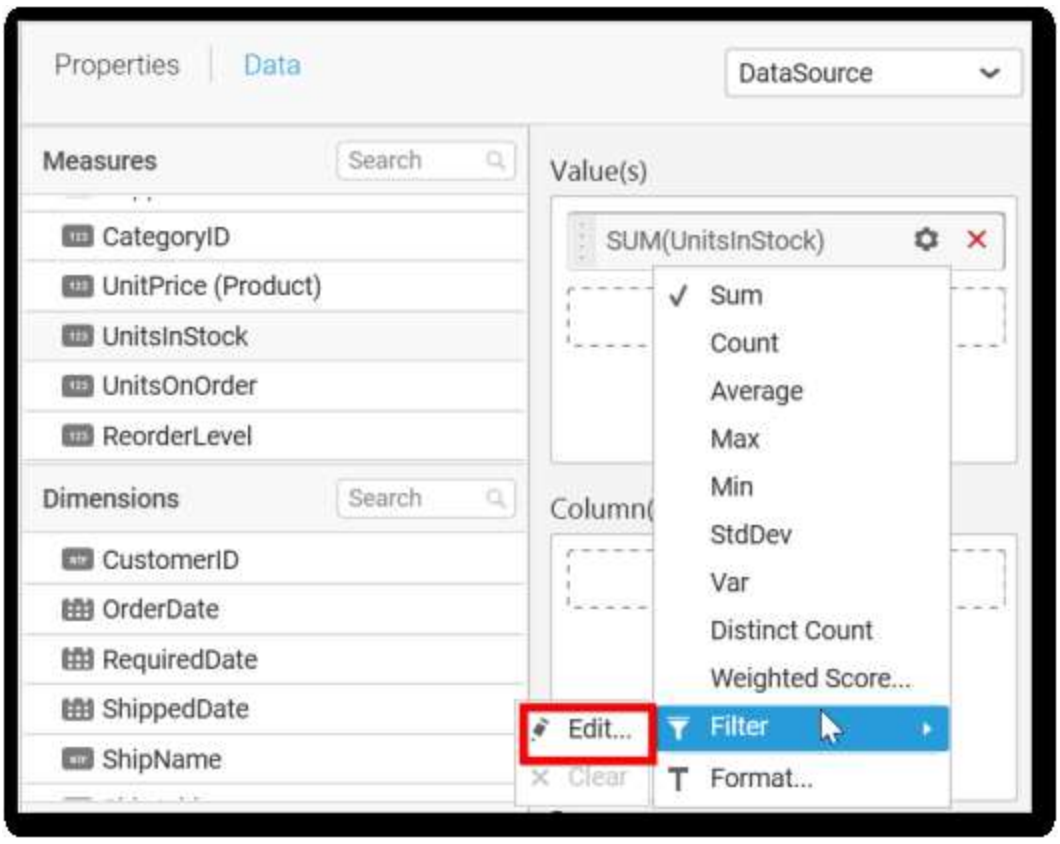


You can change the Settings option.

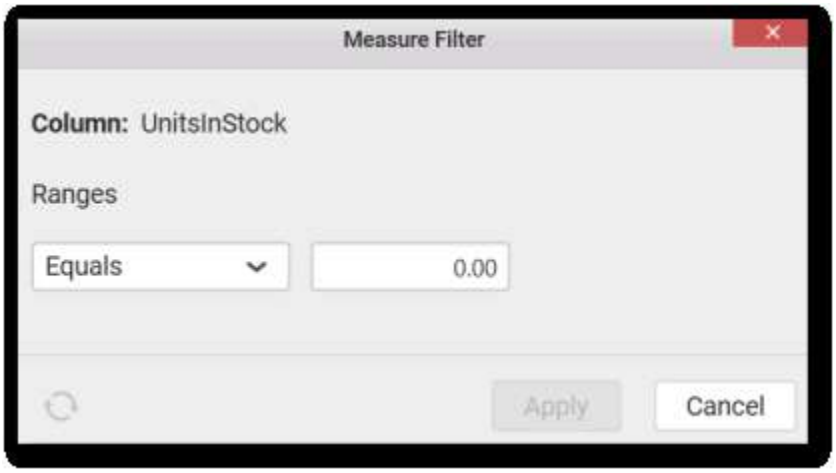


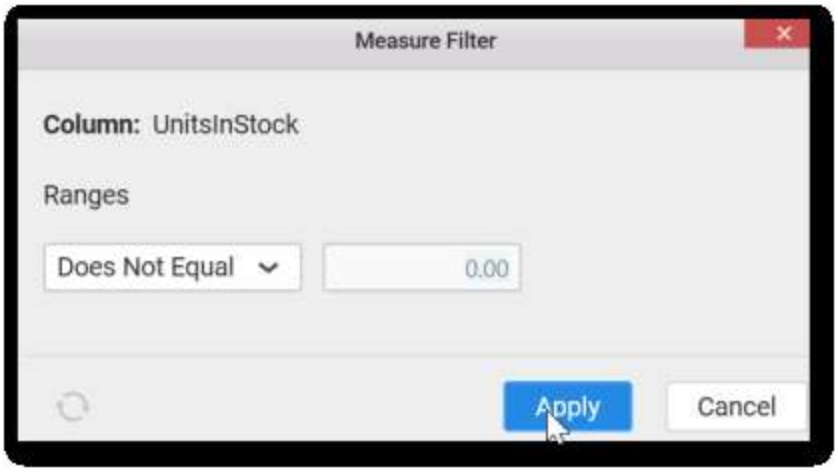
You can select the required summary type from the available summary types shown in settings.

You can filter the data to be displayed in doughnut chart by using filter.



When you click the Measure Filter option will appear.





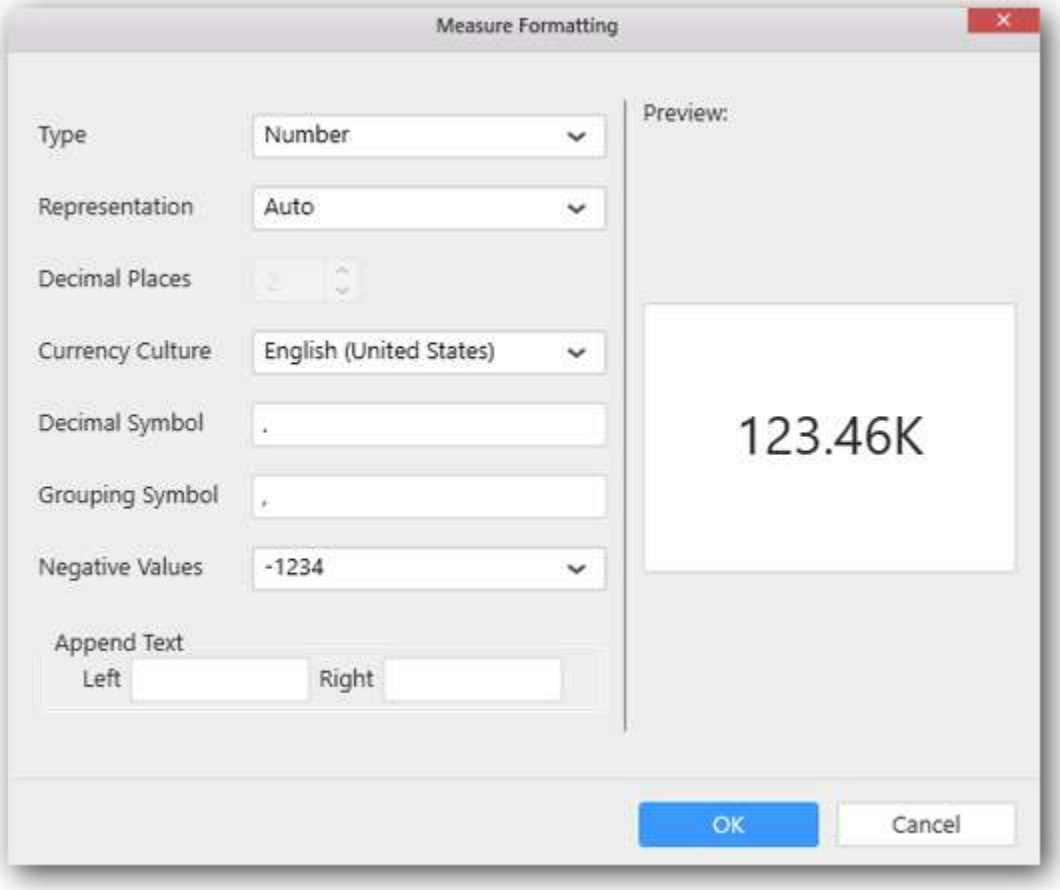
You can select the Condition to be applied in the shown list box and set the value in text box.

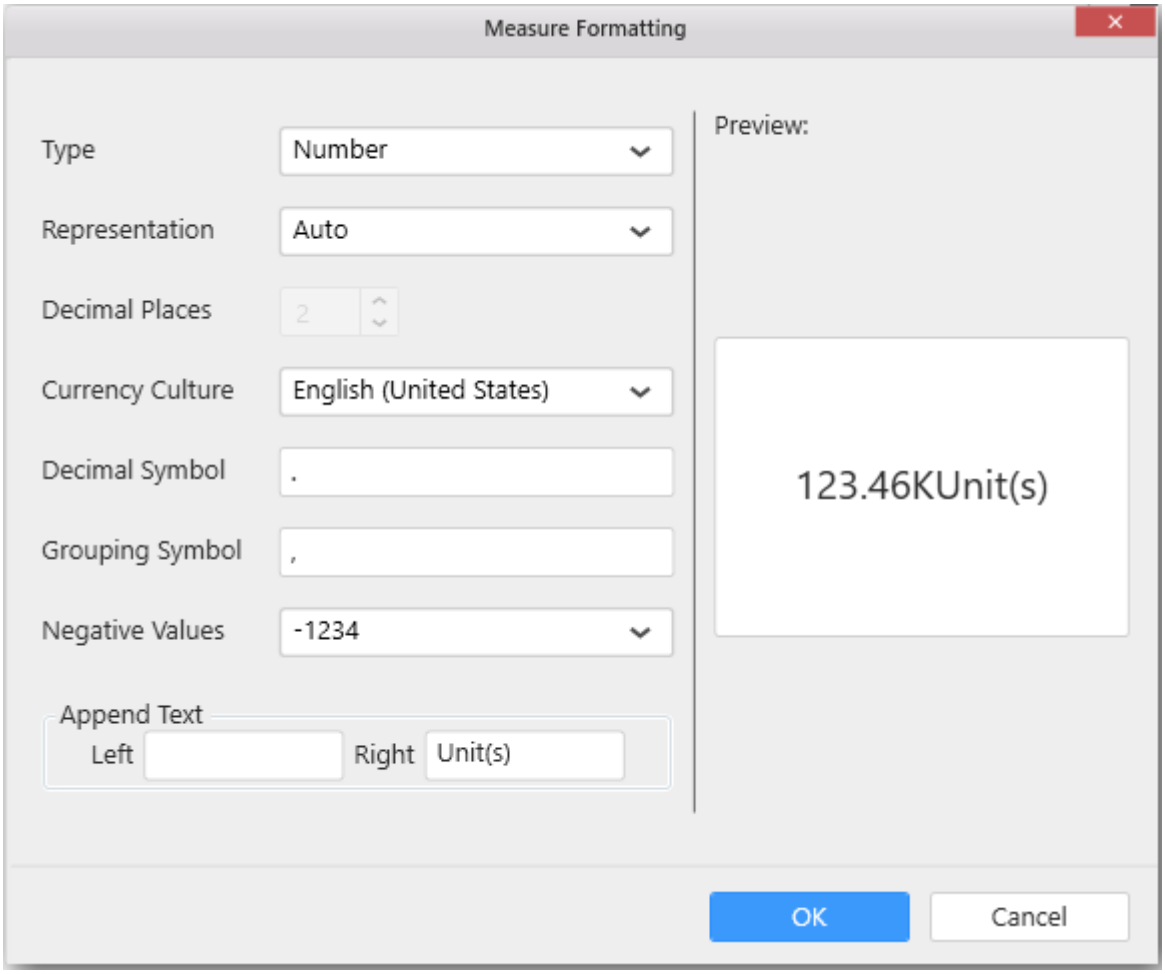
Click on Apply to see the changes.

You have option to Clear the applied filter. Click on clear to remove the filters

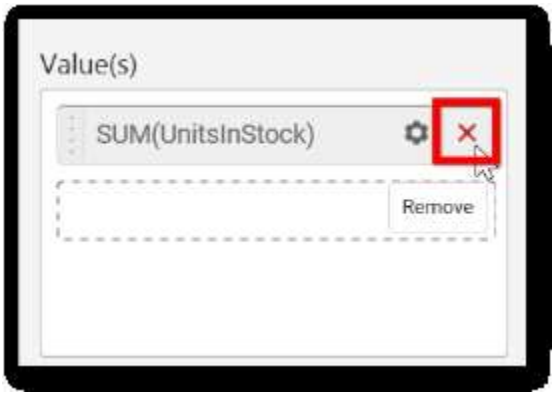
You can format the data to be displayed in the doughnut chart by using format option.

The Measure Formatting option will be shown, select the format that you want and click OK

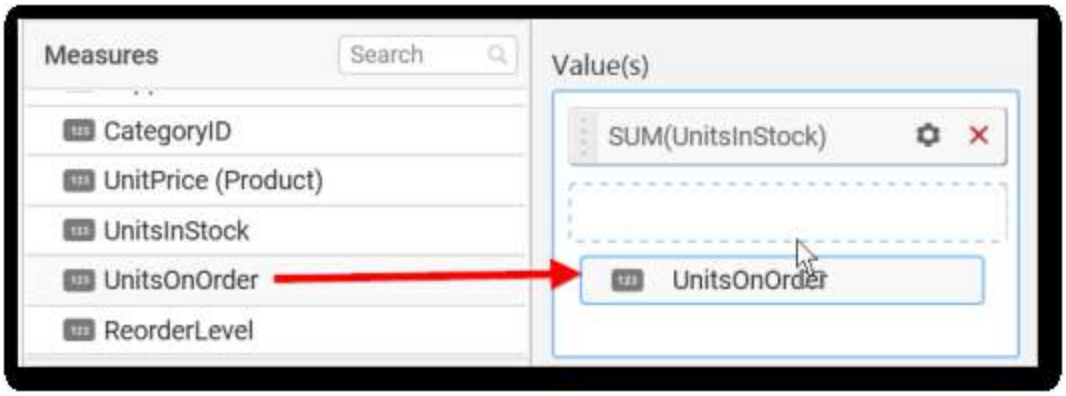


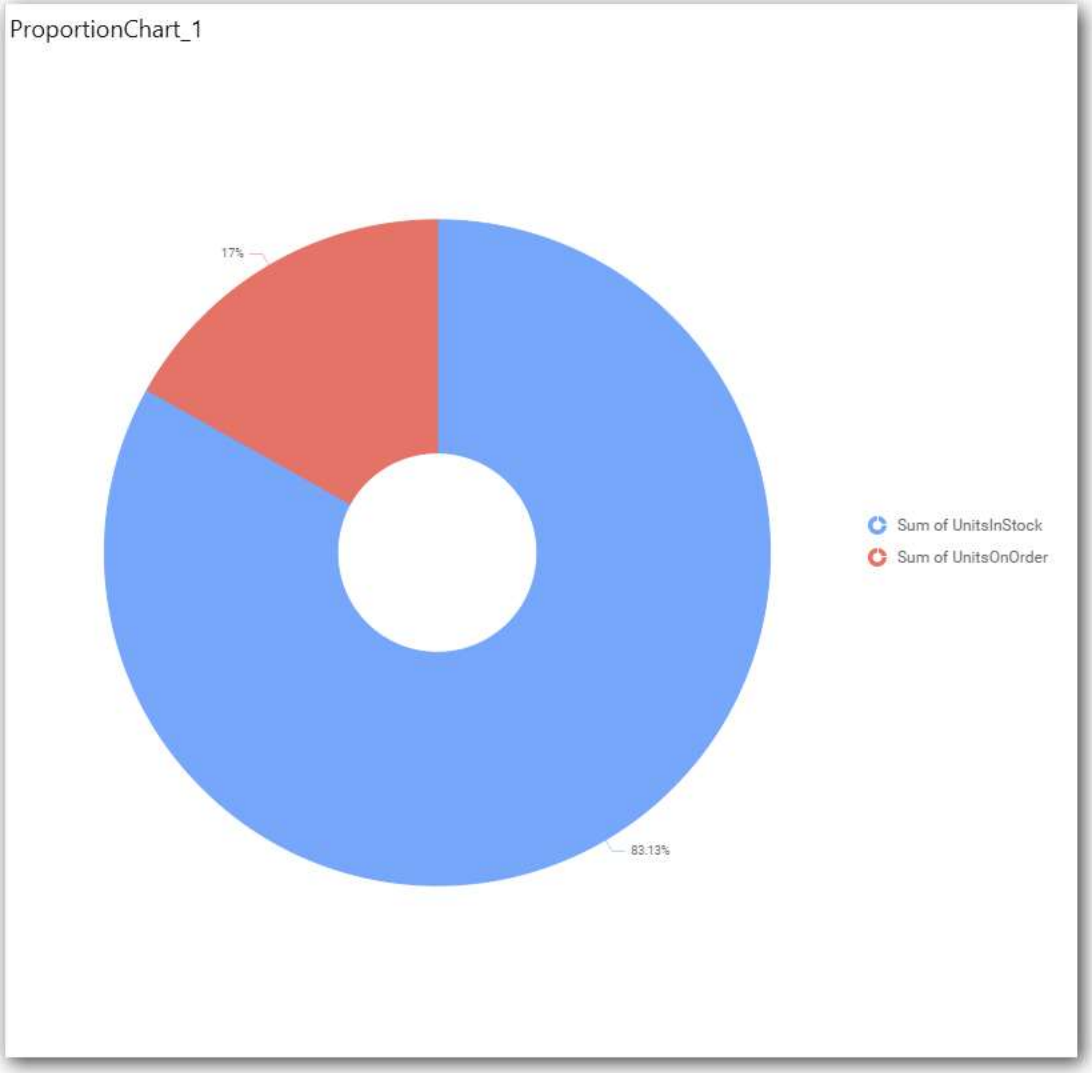


To remove the added value fields click on x button.



You can add multiple Measures into Value.

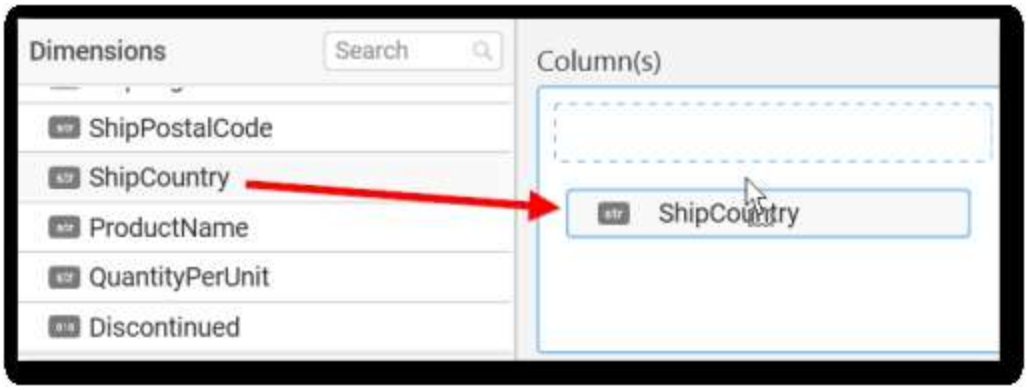




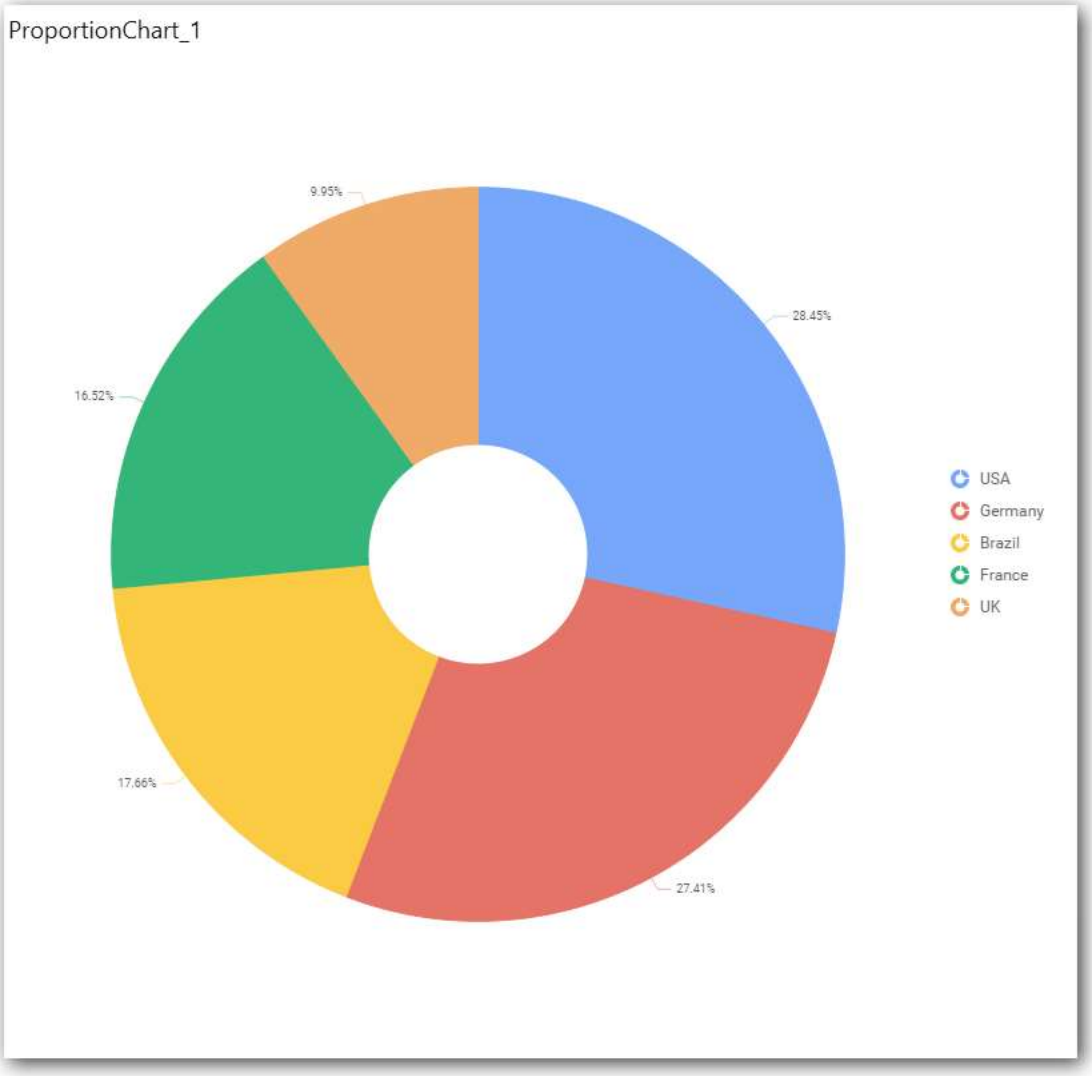
You can also drag and drop Dimension(s) and Expression Column(s) into Value(s)

Adding Column(s)

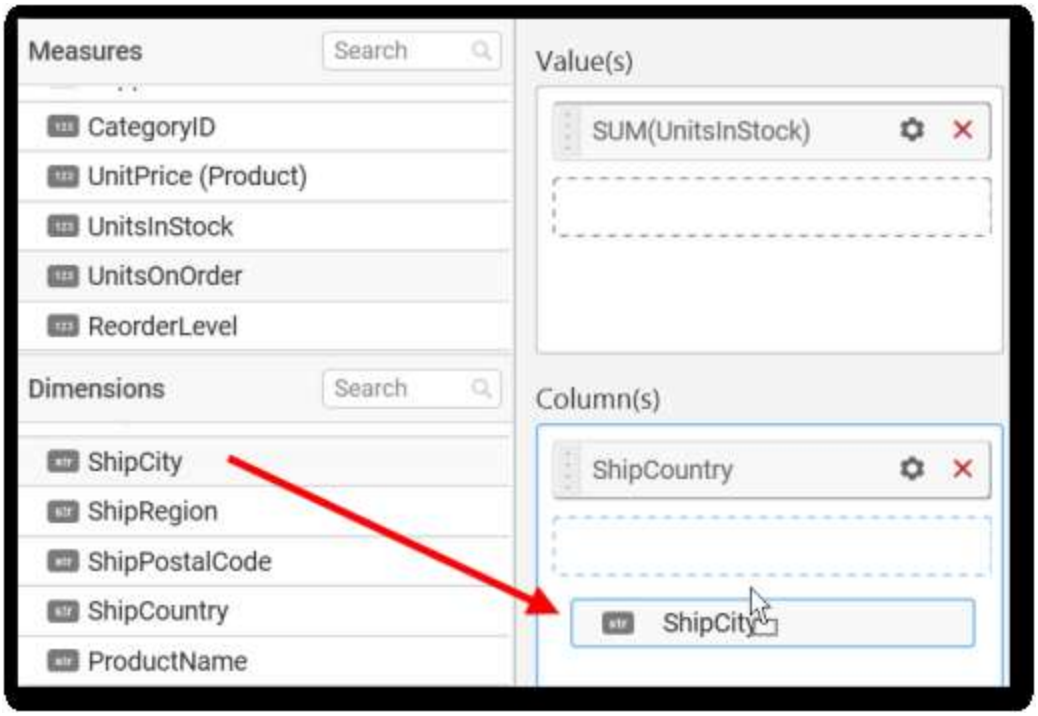
Drag and drop the Dimensions into Column(s)



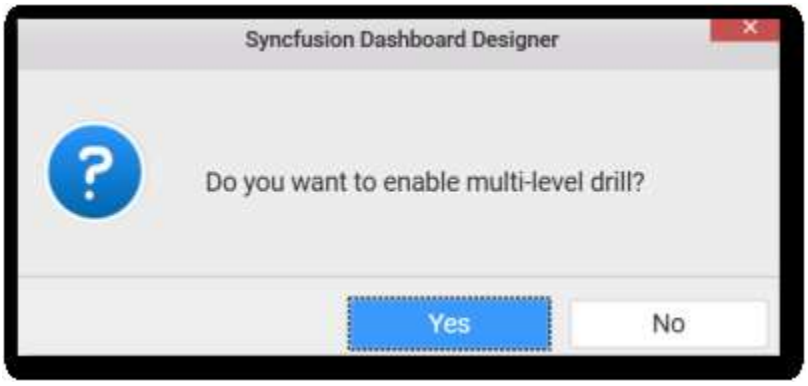
Doughnut chart will be rendered like this.



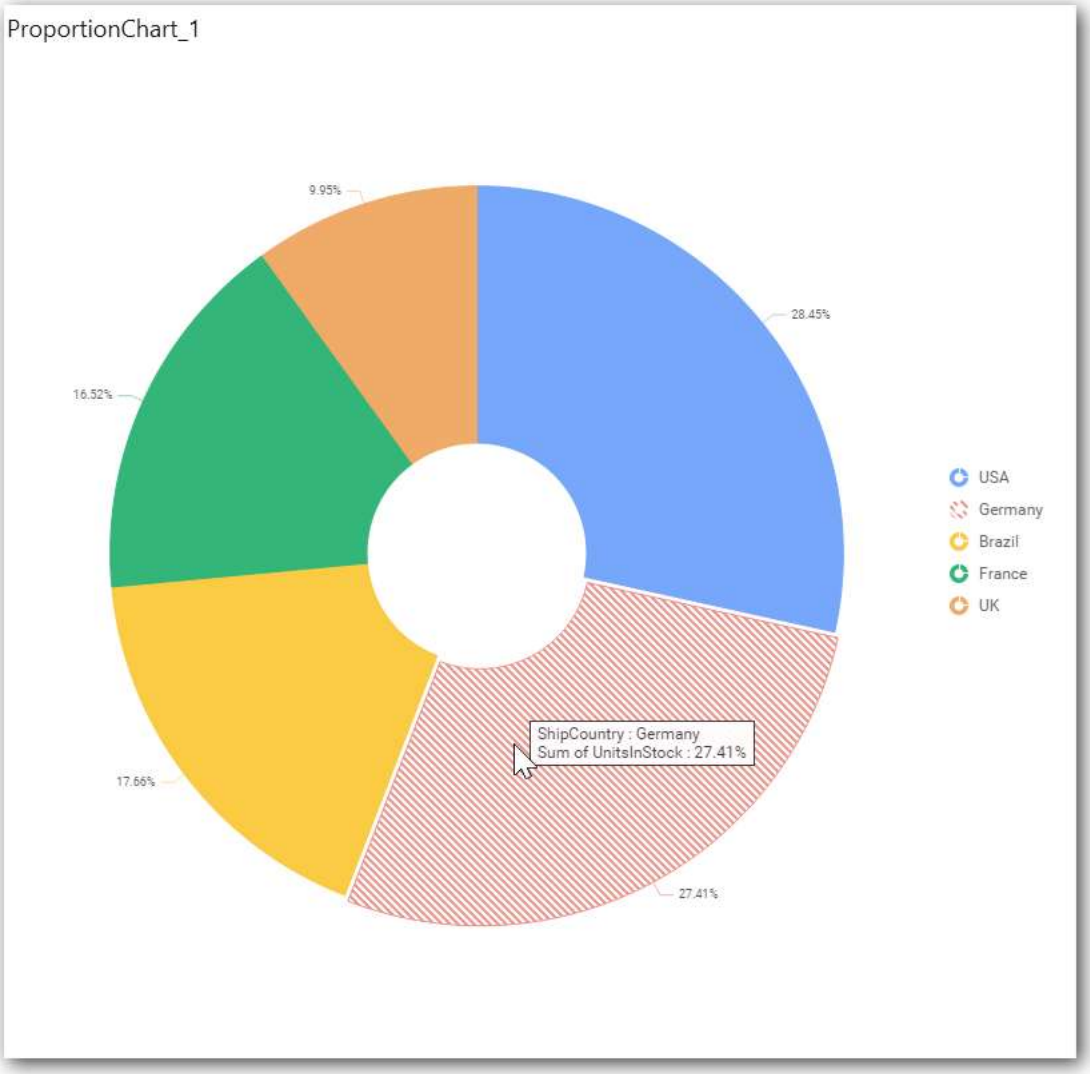
You can add more than one value into Column(s) field.



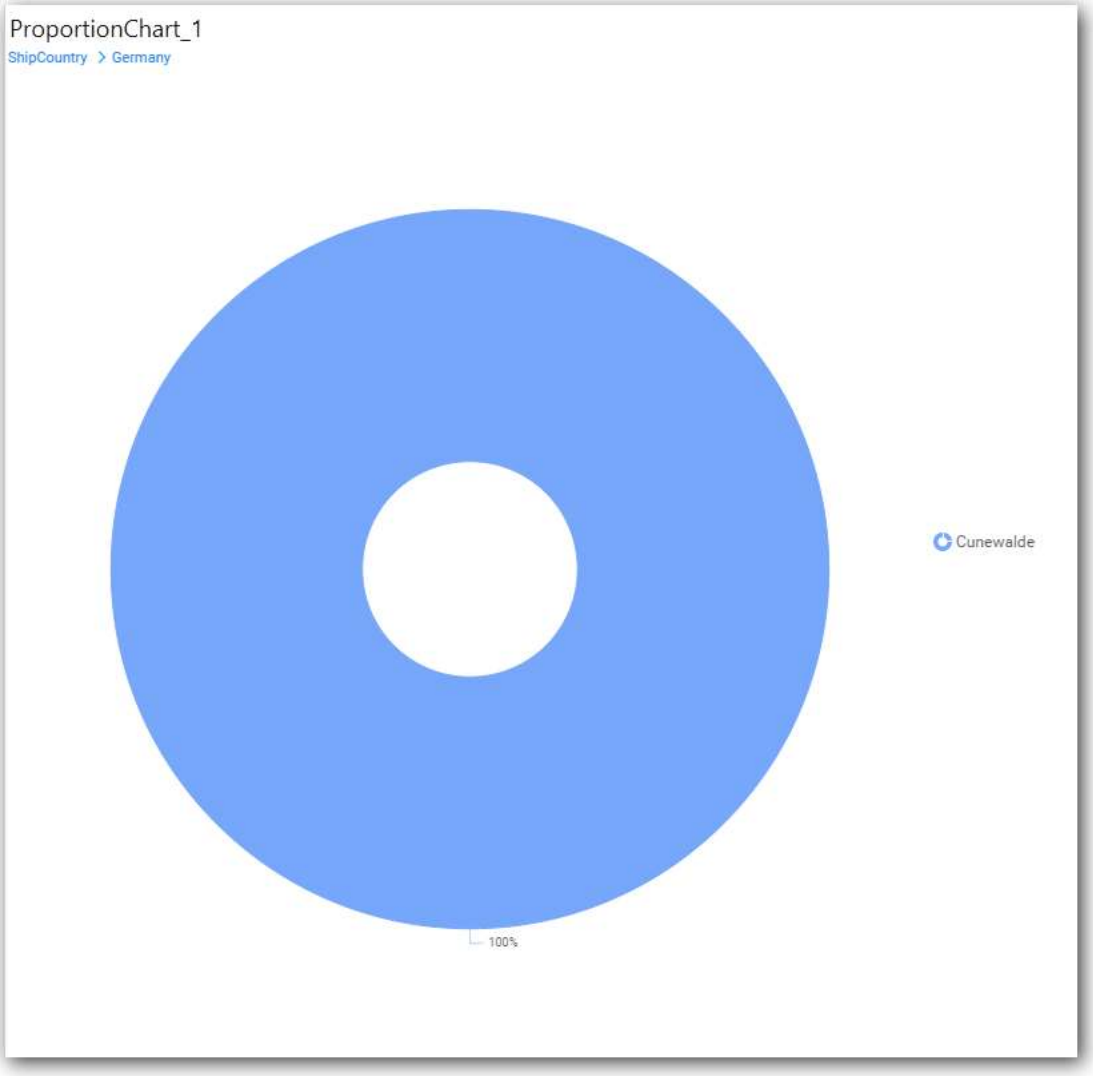
The following message will open



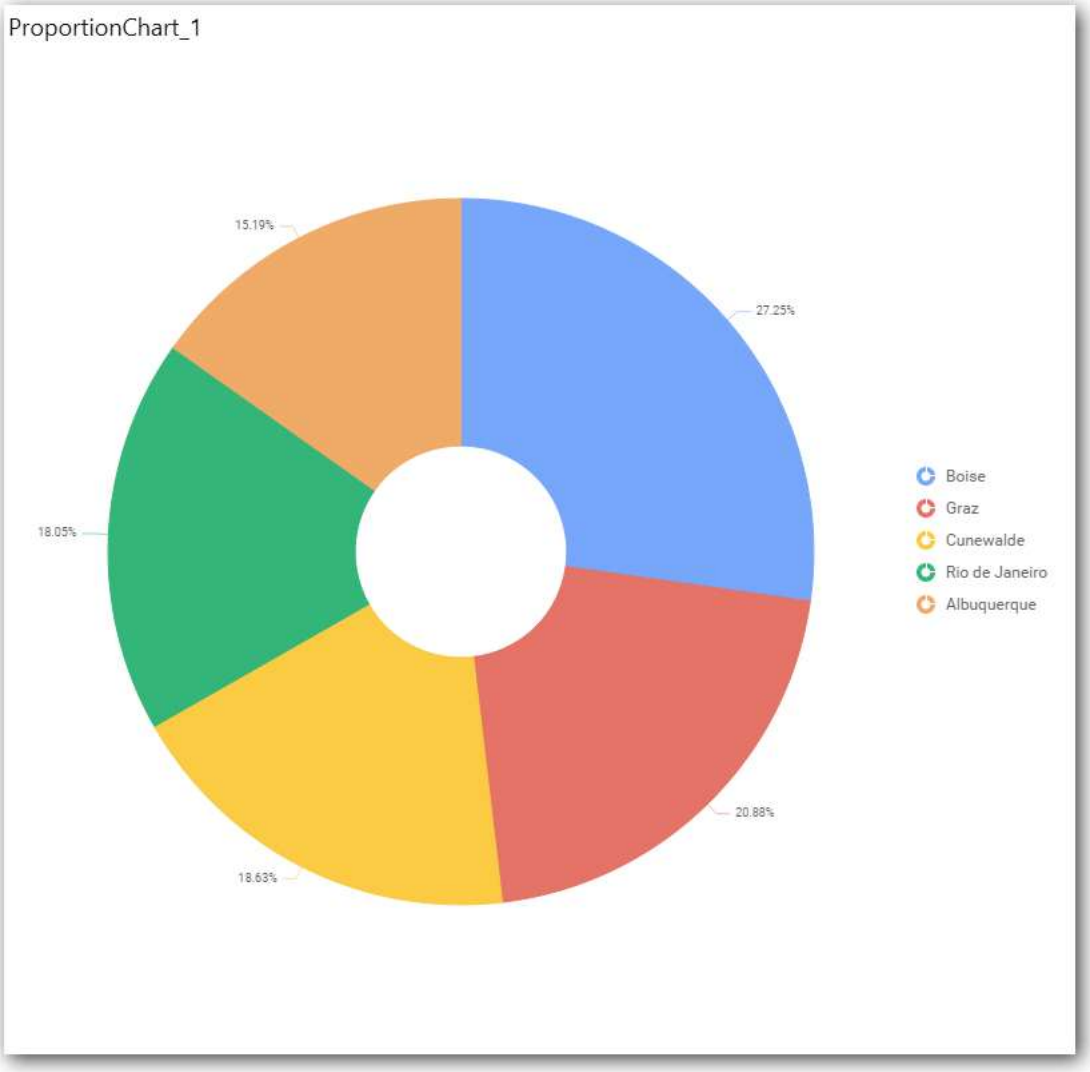
To enable drill down click Yes.



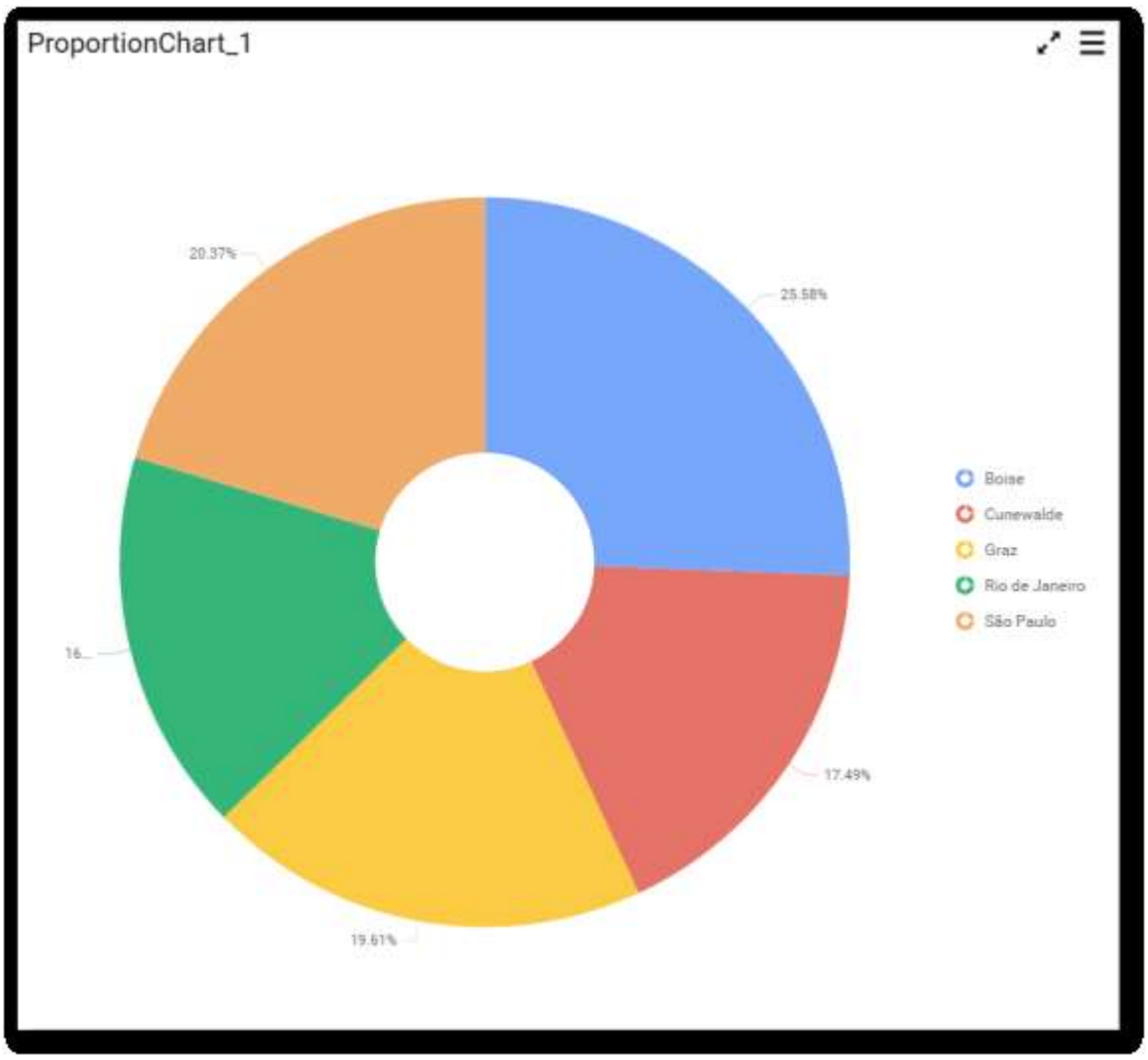
The drilled view of the chart region selected.



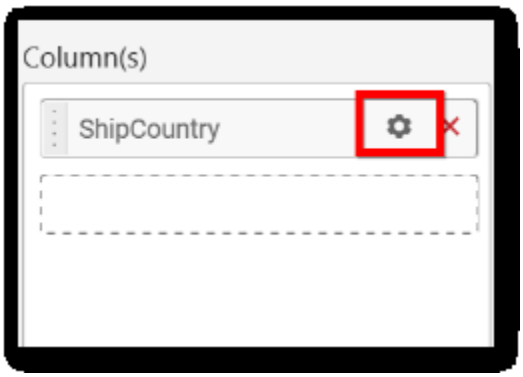
To continue without drill down click No.

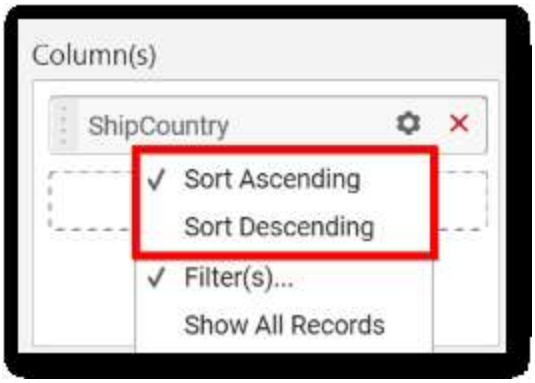


The old column value will be replaced by new column value.

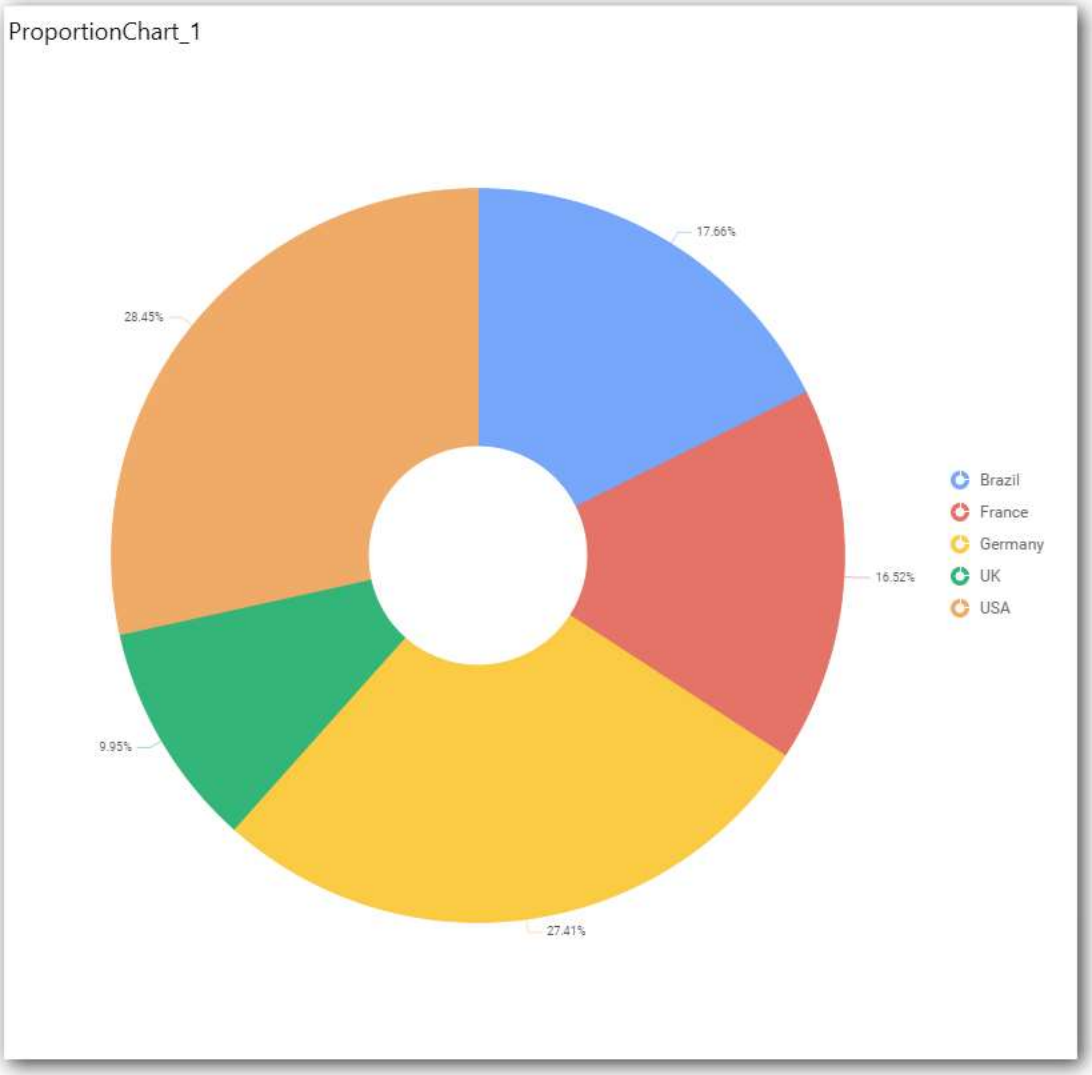


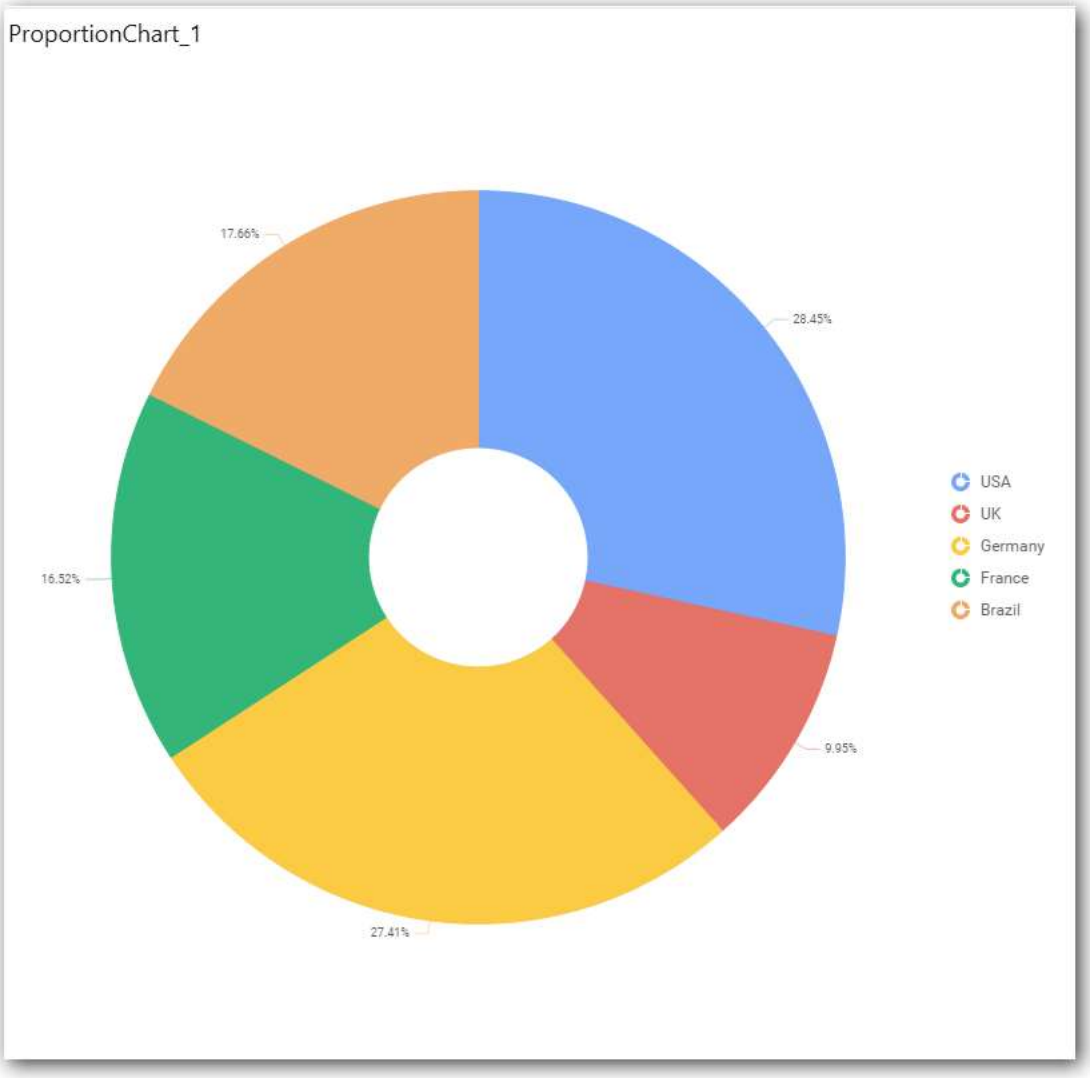
You can change the settings.





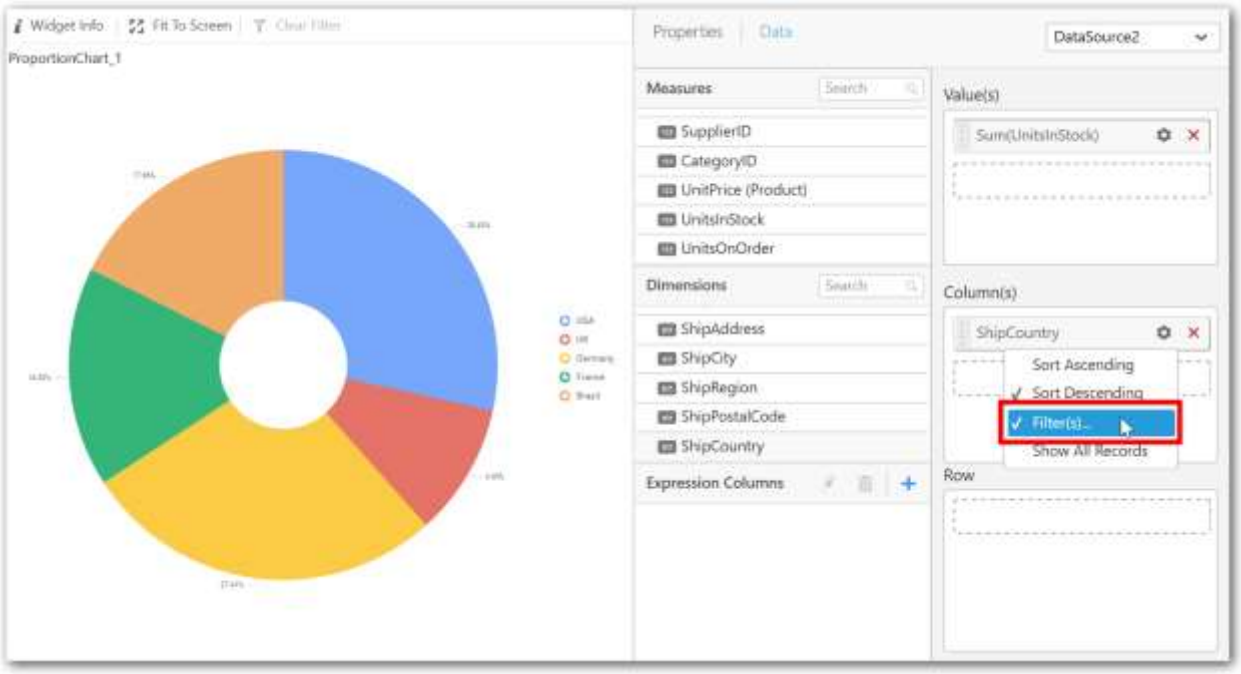
You can select sort Ascending or Descending.





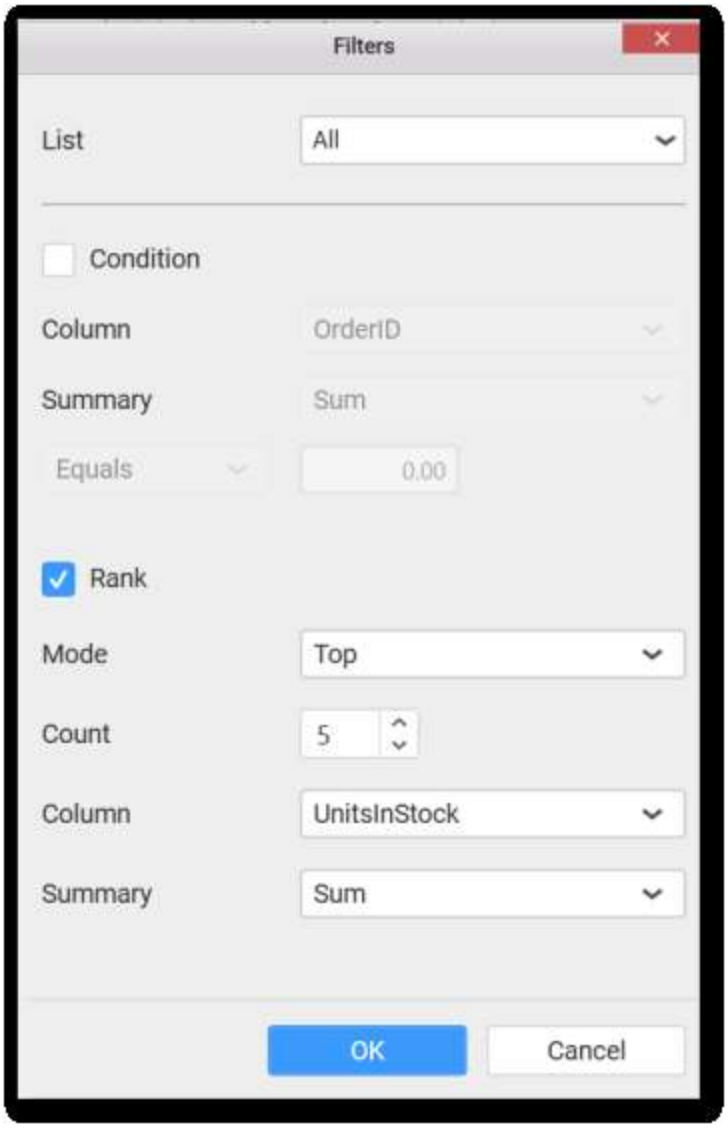
You can apply filters by selecting filter in settings.

Note: Filter will be set by default for top 5 records.



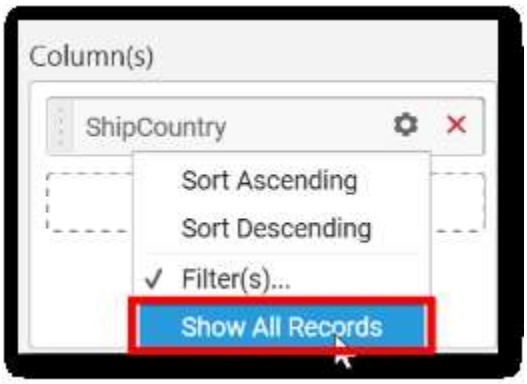
The filters option will open.

Select the needed Conditions and Rank and then click OK.



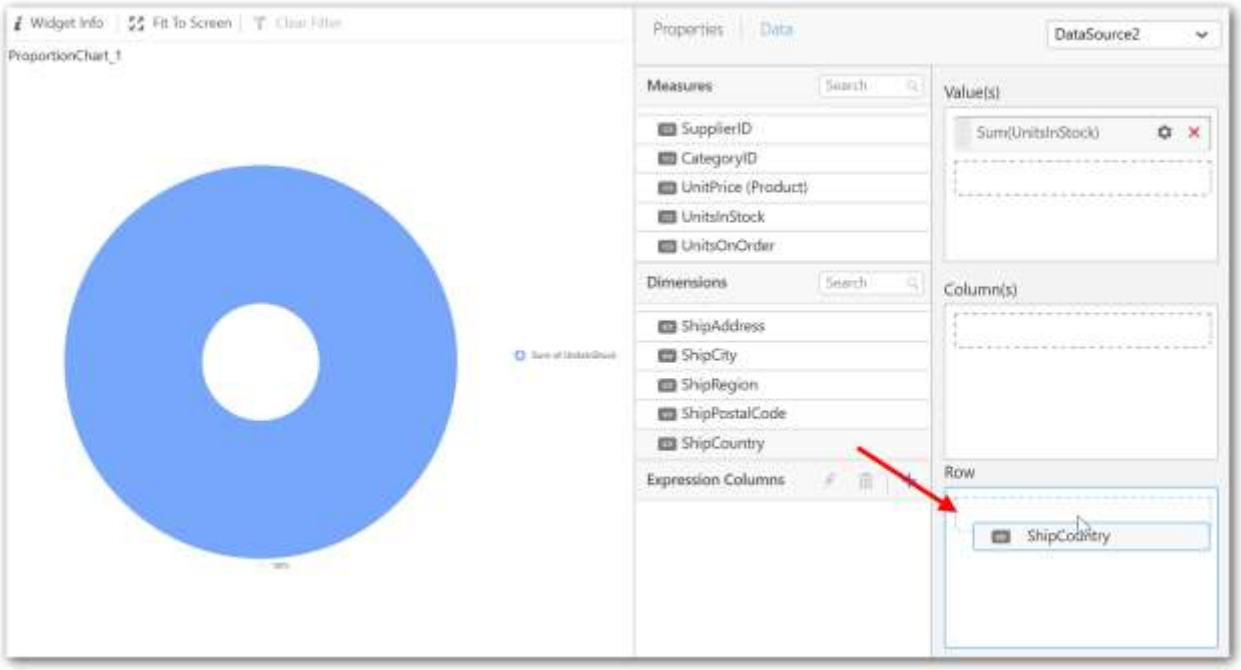
Similarly you can add the Measures and Expression Columns into Column(s) field.

To remove the filters applied, select Show All Records.

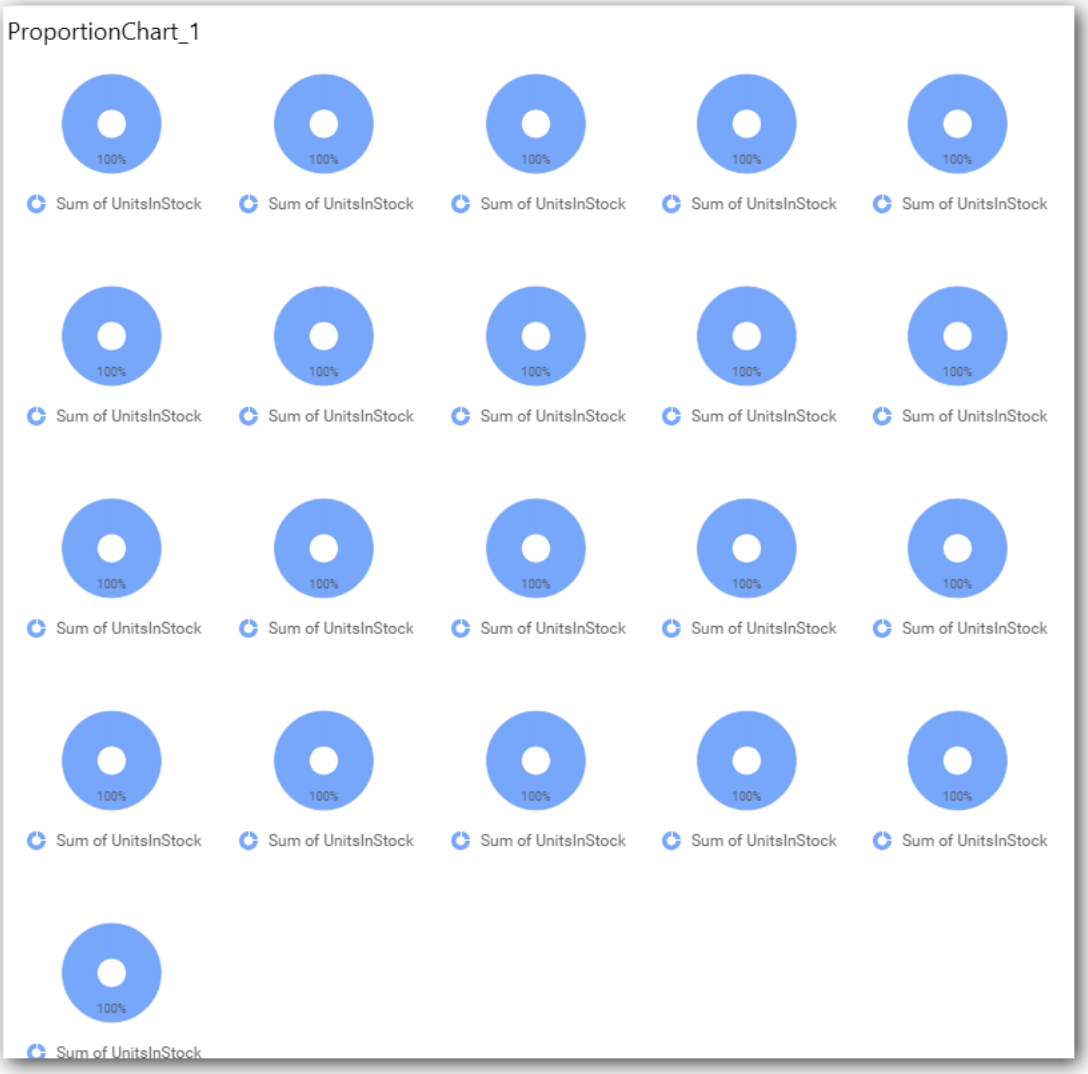


Adding Row

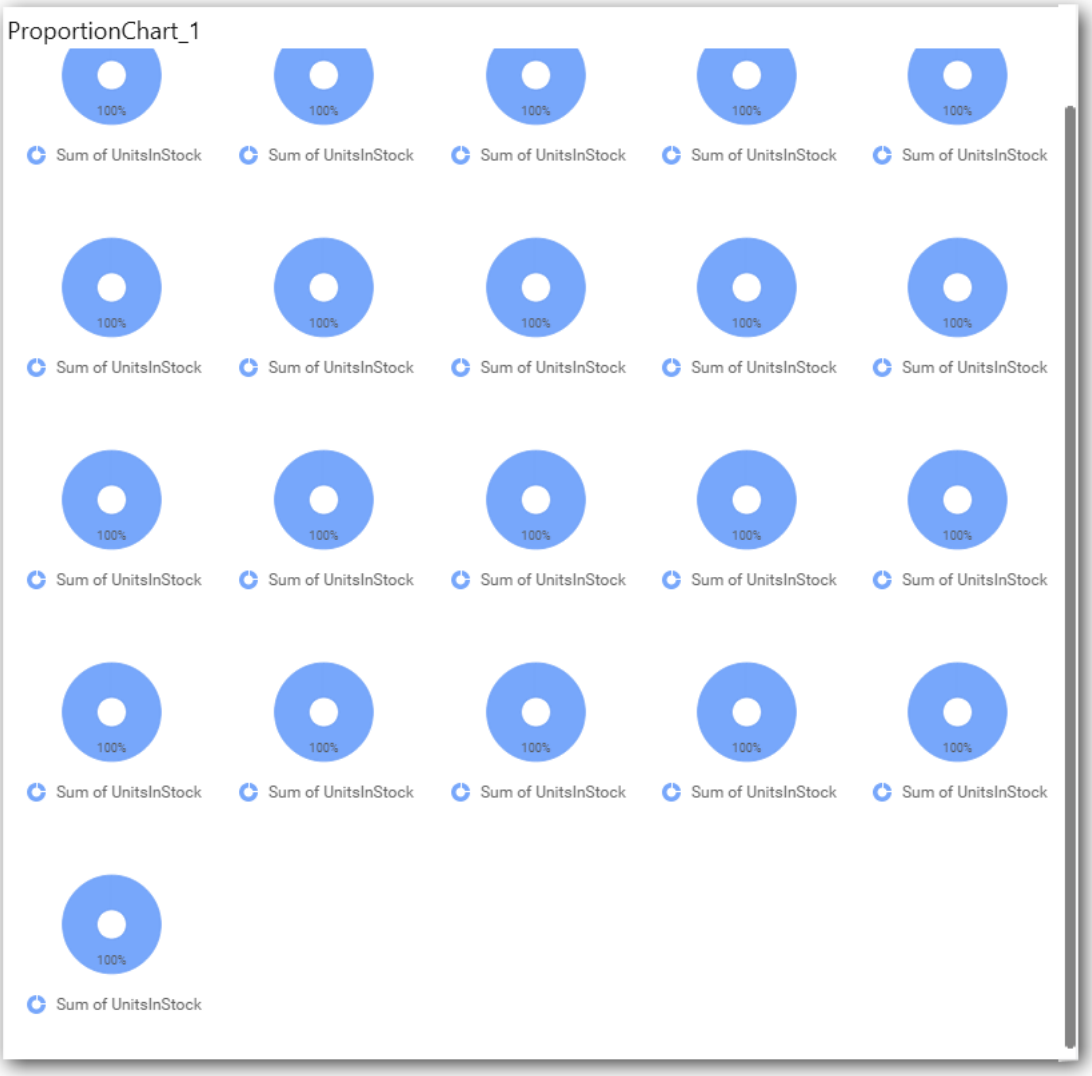
You can drag and drop the Measure or Dimension into the Row field.



This will render doughnut chart in series.



Scroll to see all charts.

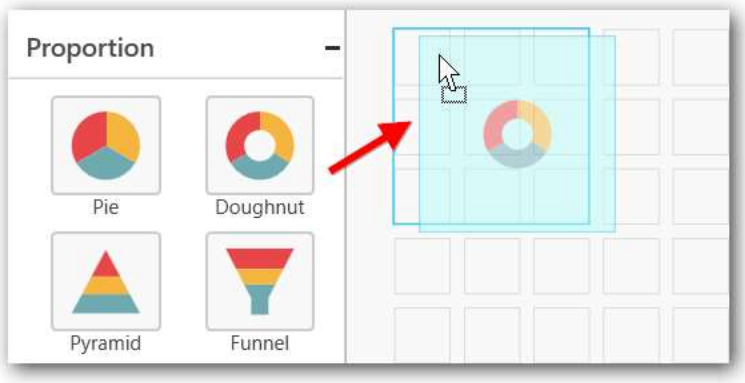


How to configure the SSAS data to Doughnut Chart?

Doughnut Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

Following steps illustrates configuration of SSAS data to Doughnut chart

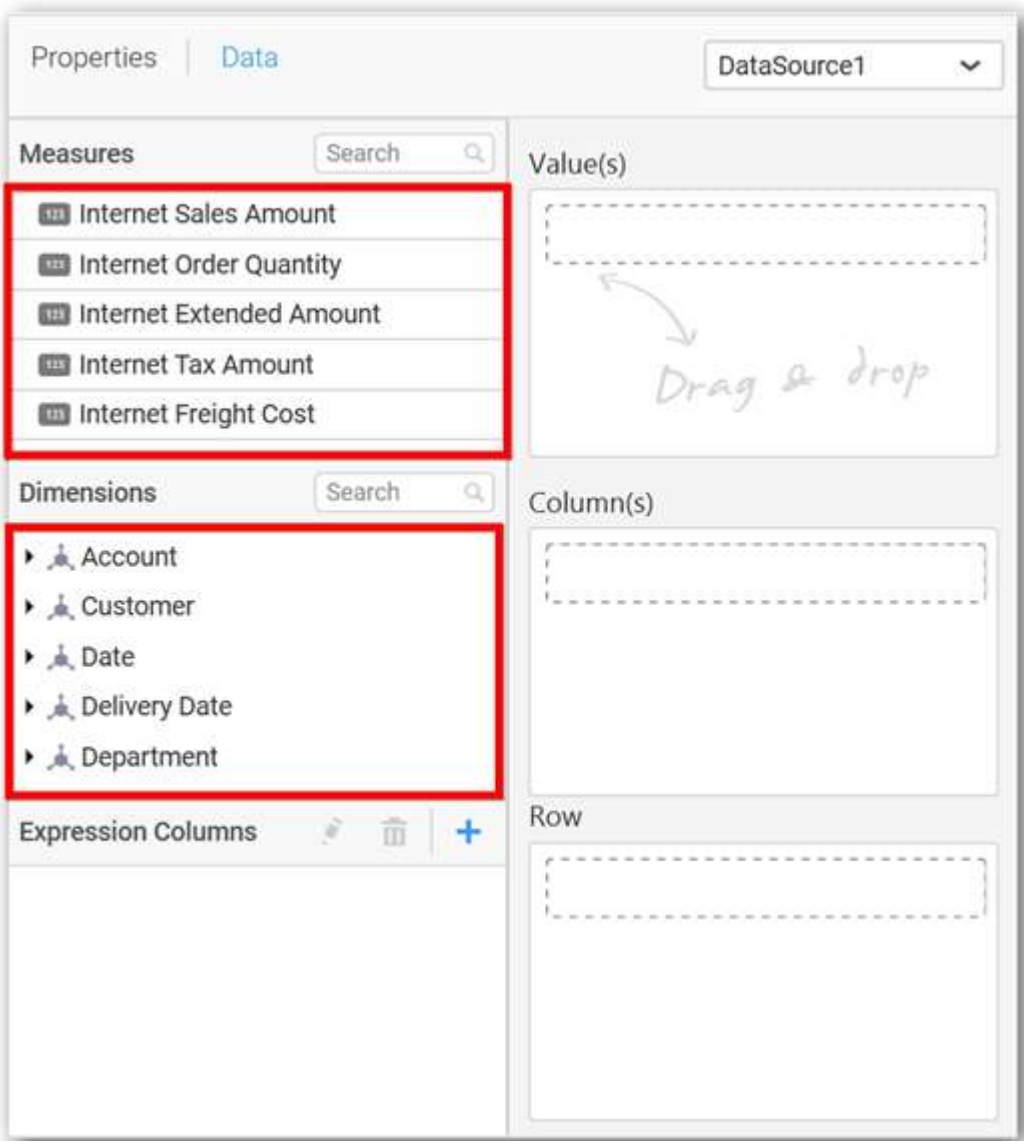
Drag and drop the Doughnut chart into canvas and resize it to your required size.



Select the dropped widget using mouse.

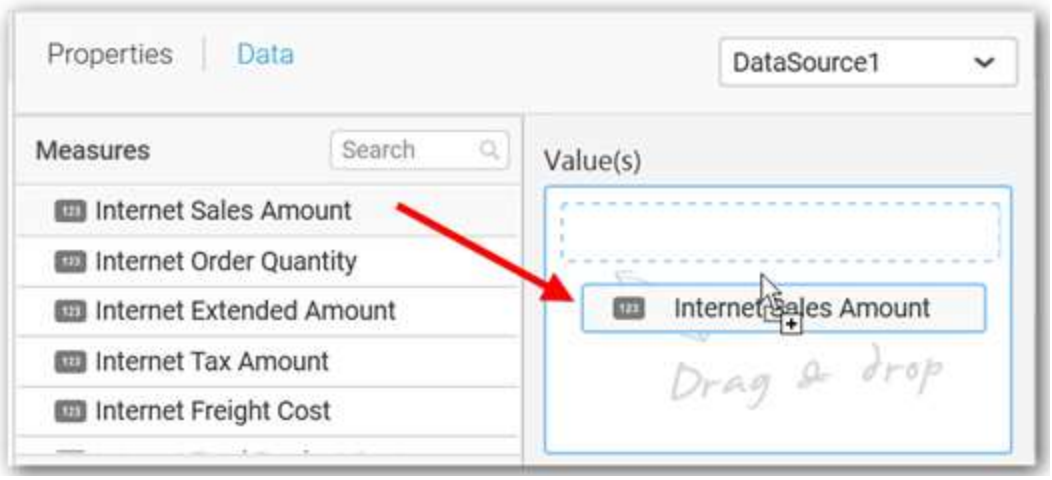
Click the Assign Data button in the toolbar.

A Data pane will be opened with available Measures and Dimensions.

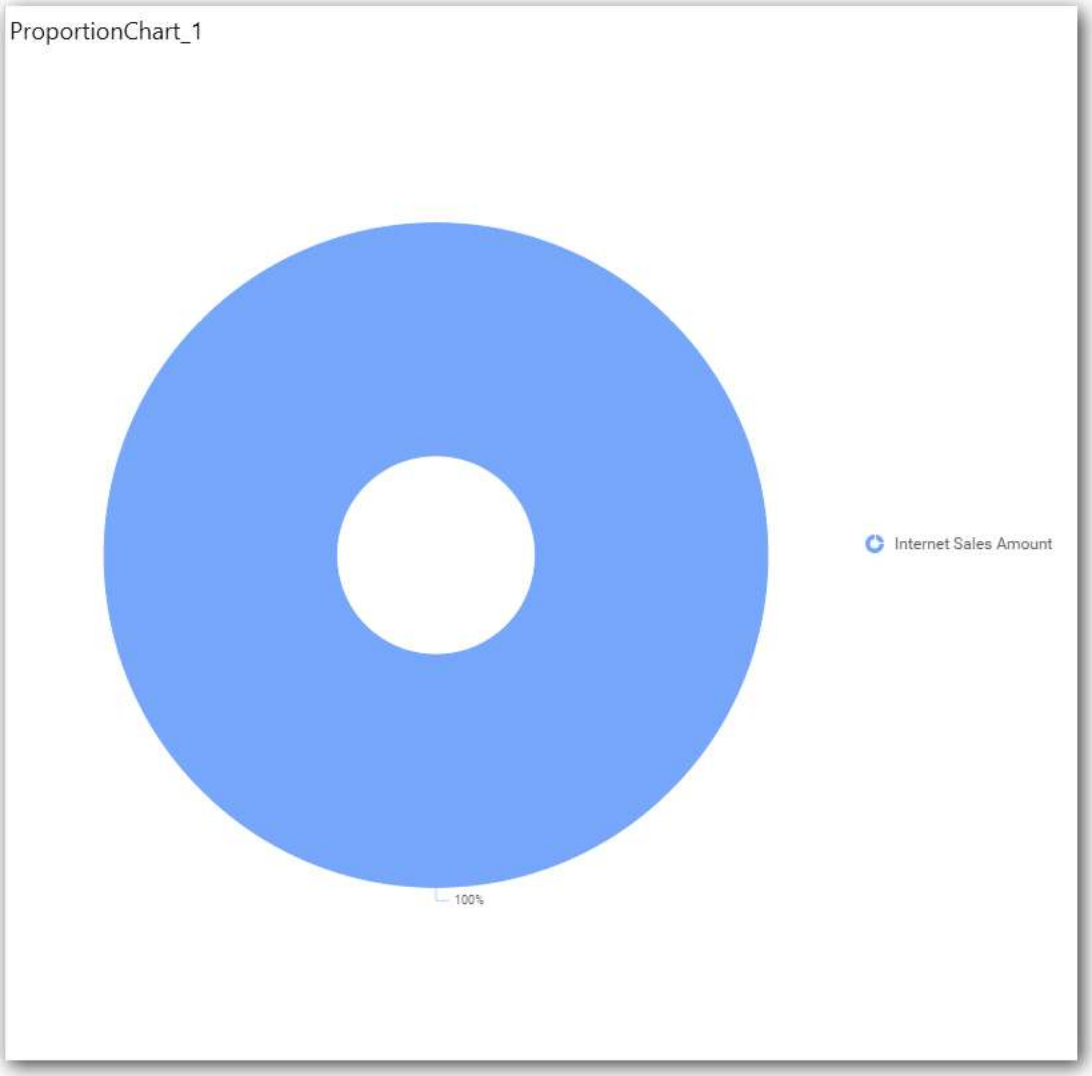


Assigning Value(s)

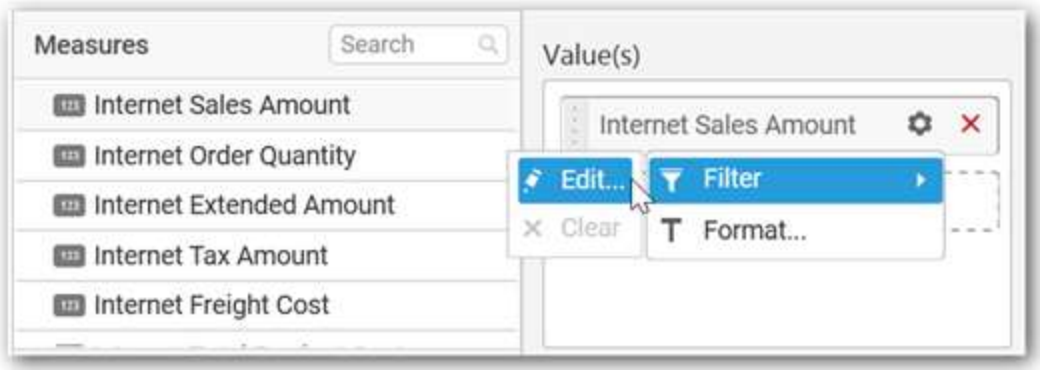
Drag and drop a column under Measures category into Value(s).



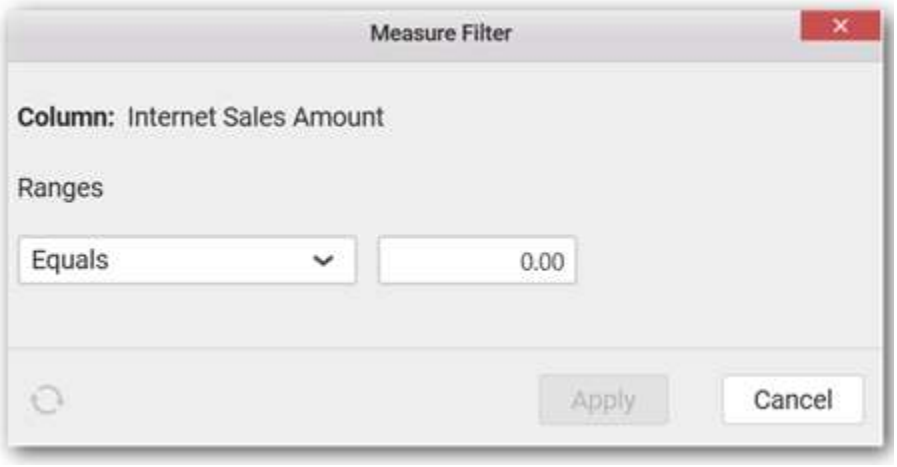
Now the chart will be rendered like this.



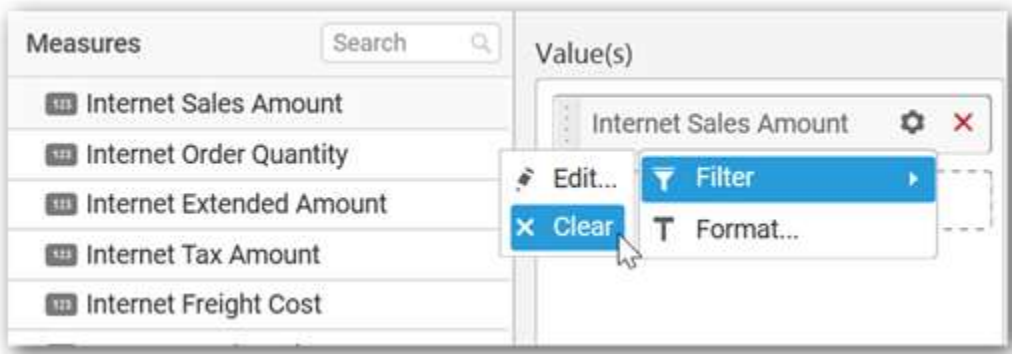
Define the filter criteria to match through choosing Edit option in Filter menu item.



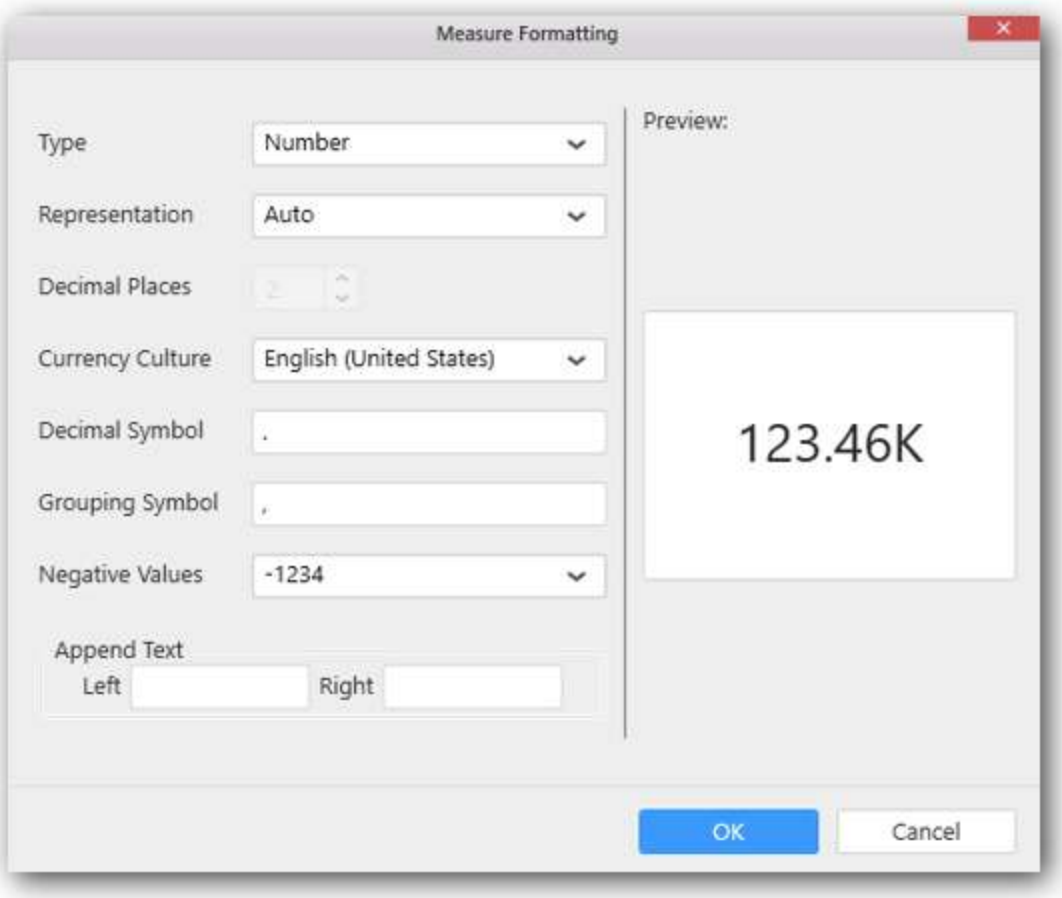
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



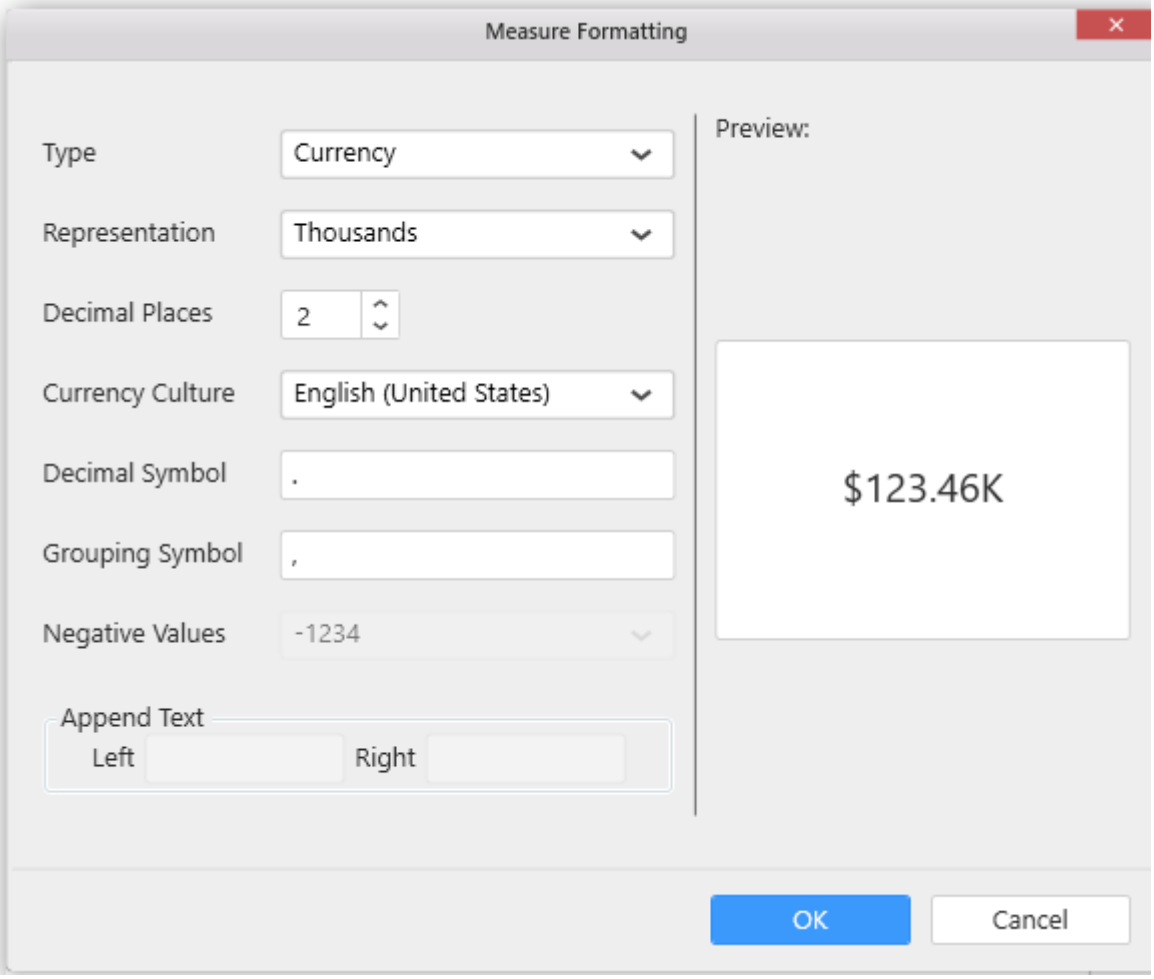
Select Clear option to clear the defined filter.



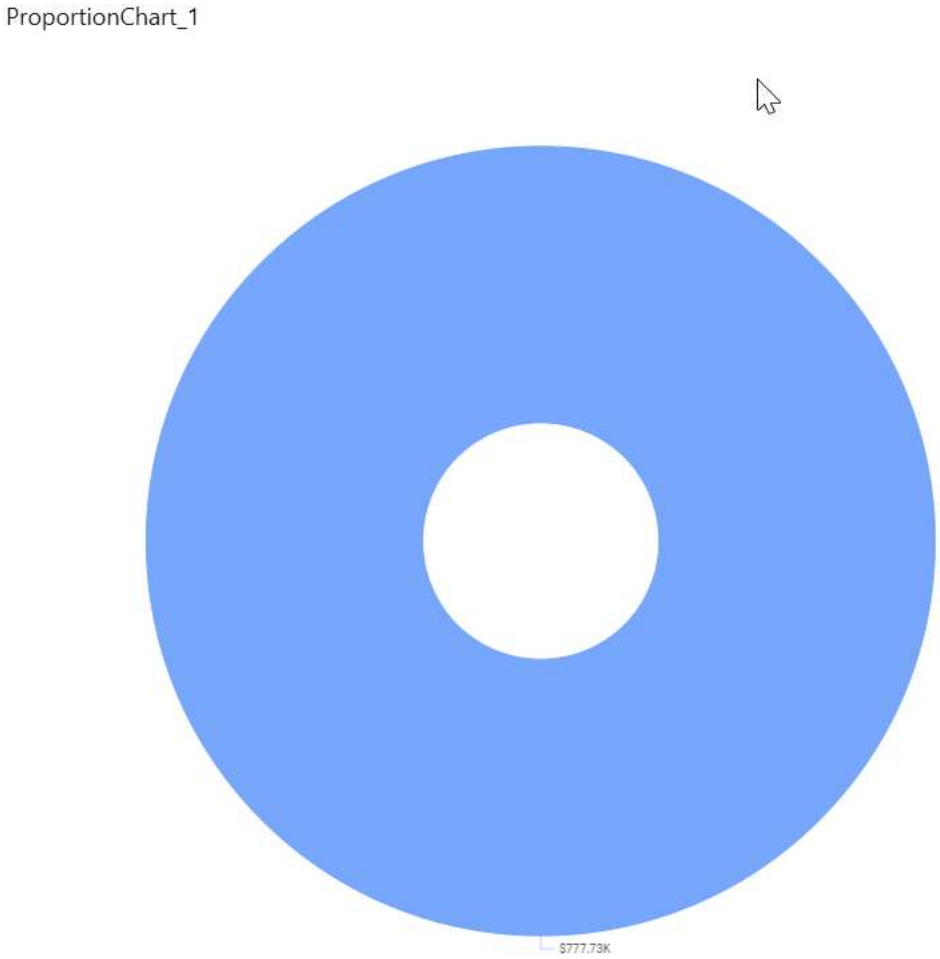
Select Format option to define the display format to the values in the column through Measure Formatting window.



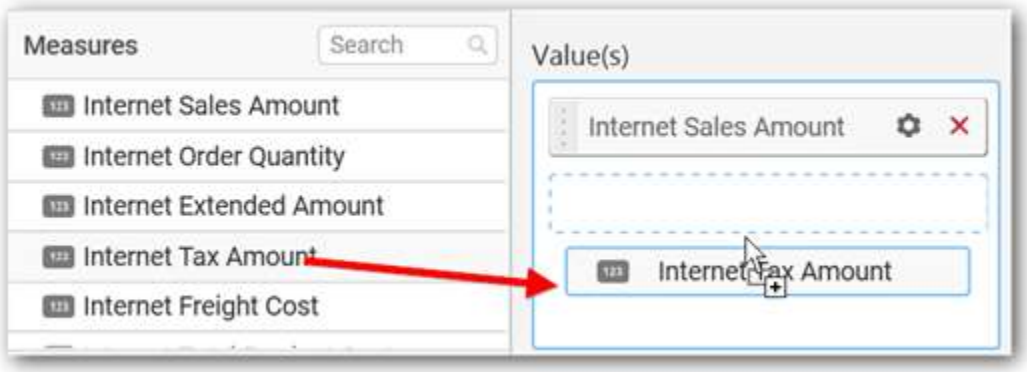
Choose the options you need and click OK.

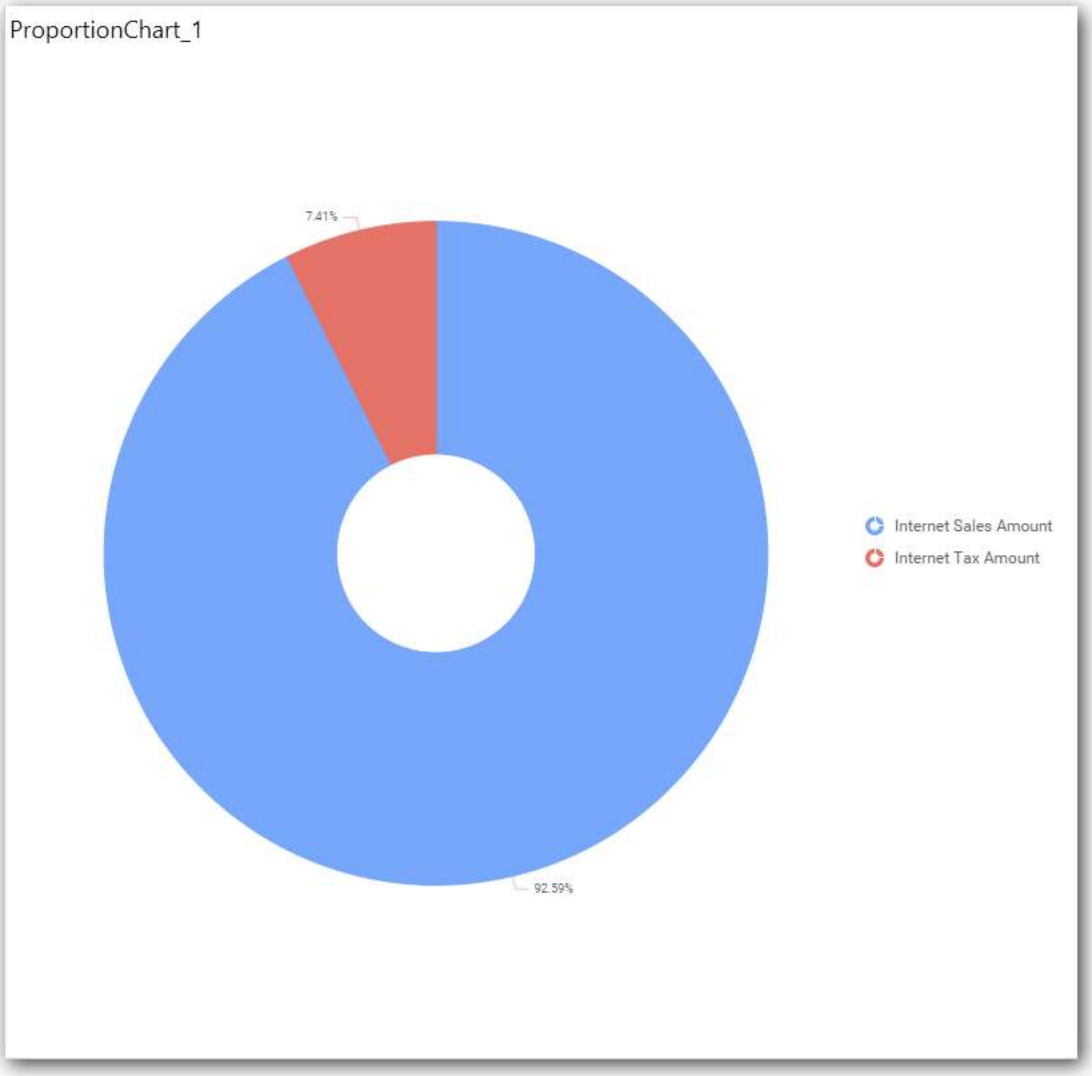


Now the Chart will be rendered like this.



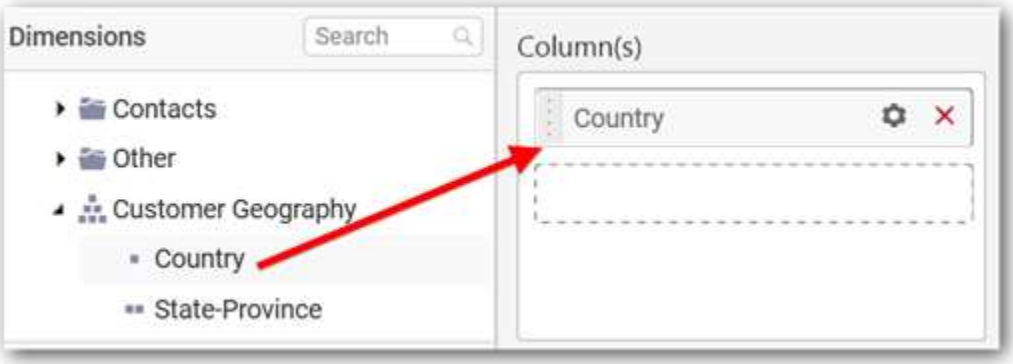
You can also add more than one column to the Value(s).

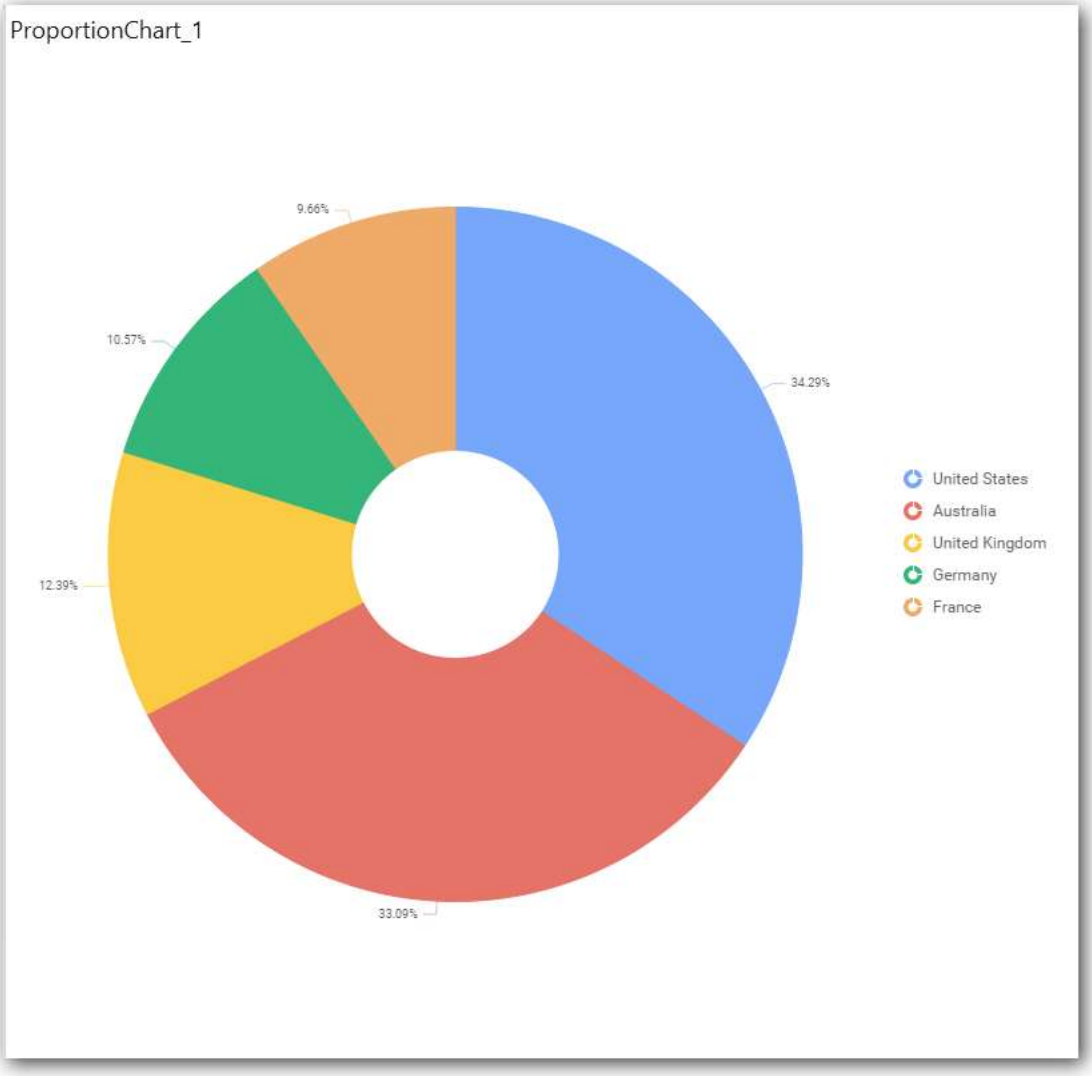




Assigning Column(s)

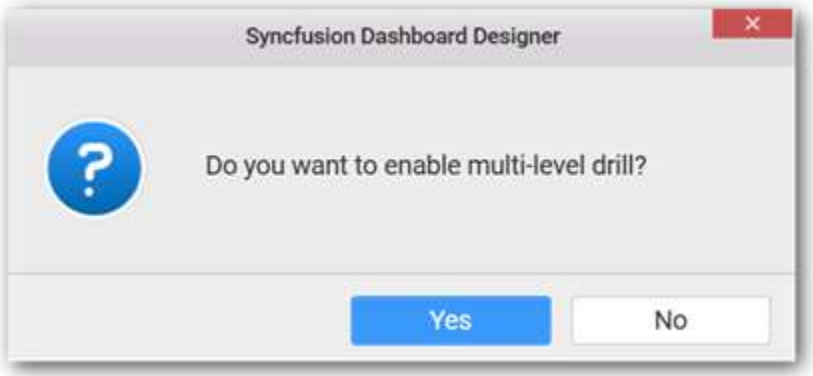
Add a dimension level or hierarchy into Column(s) section through drag and drop.





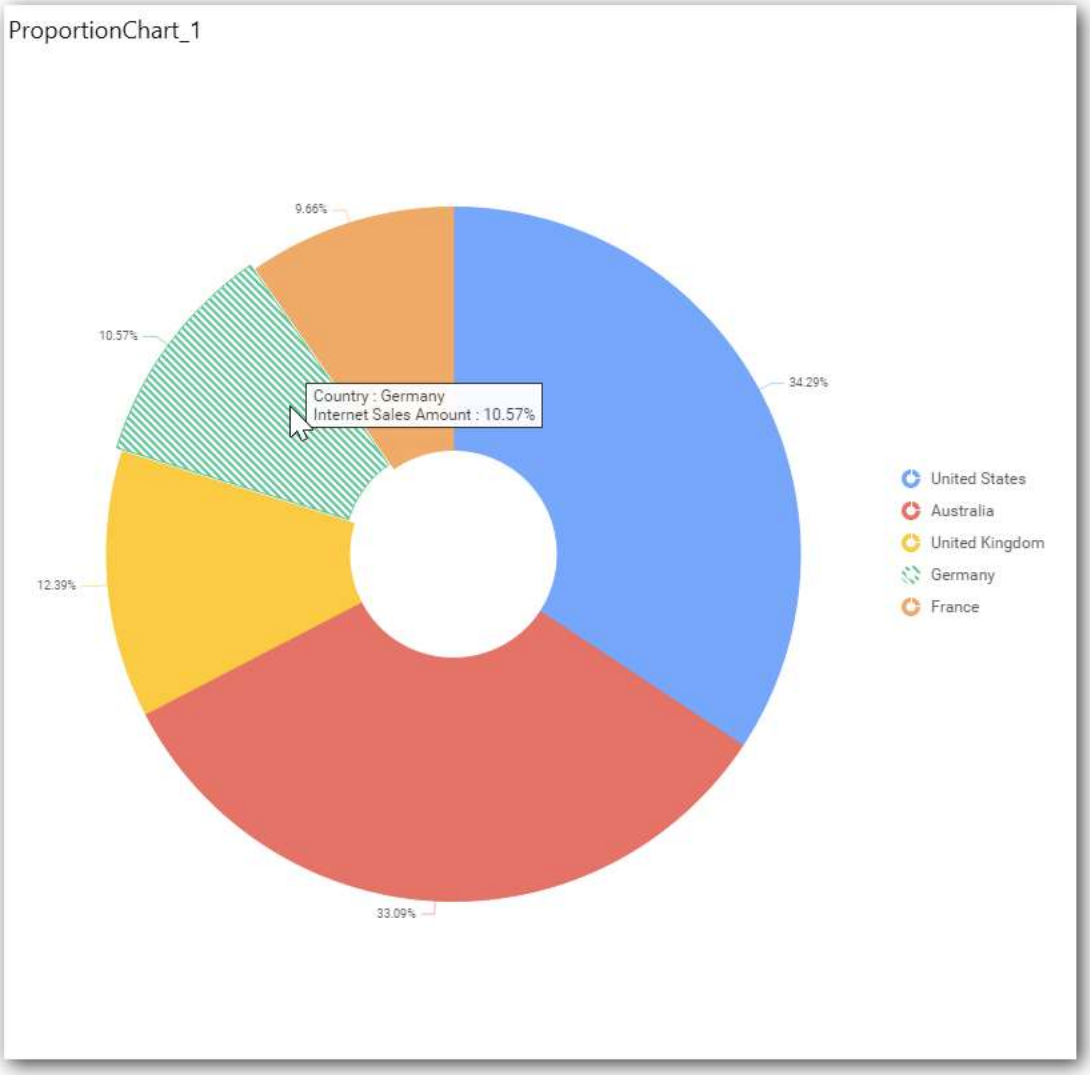
You may also add more than one column into Column(s) section.

In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

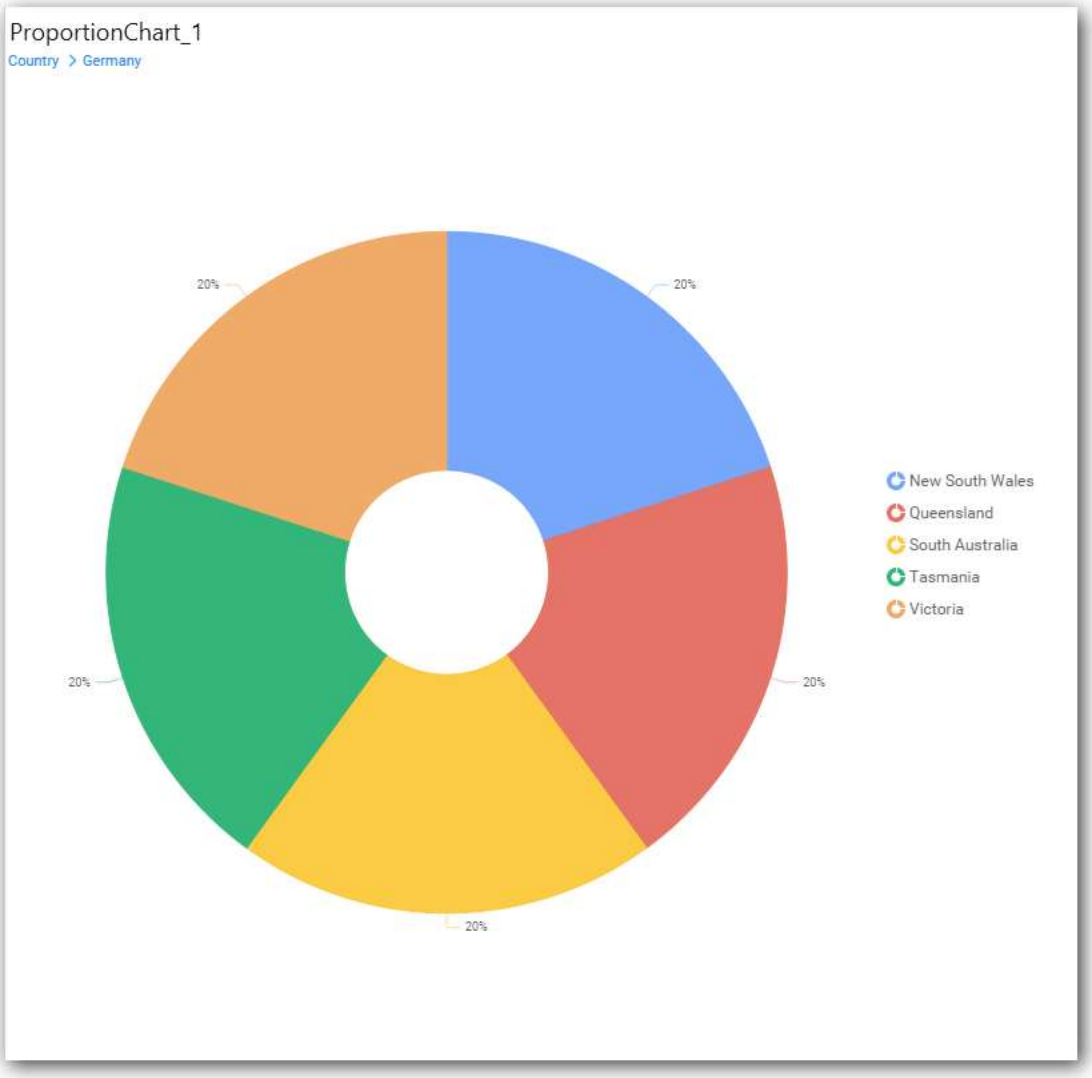


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

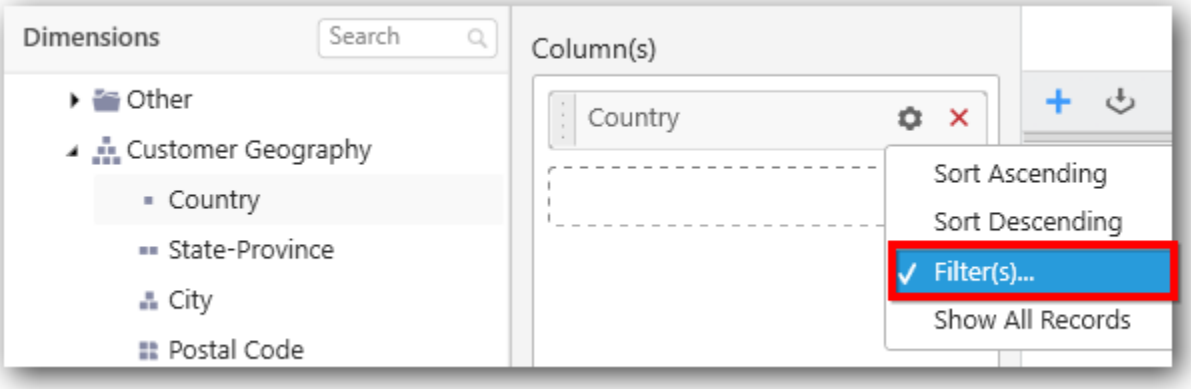
Click the respective data value marker in chart to drill into its inner level.



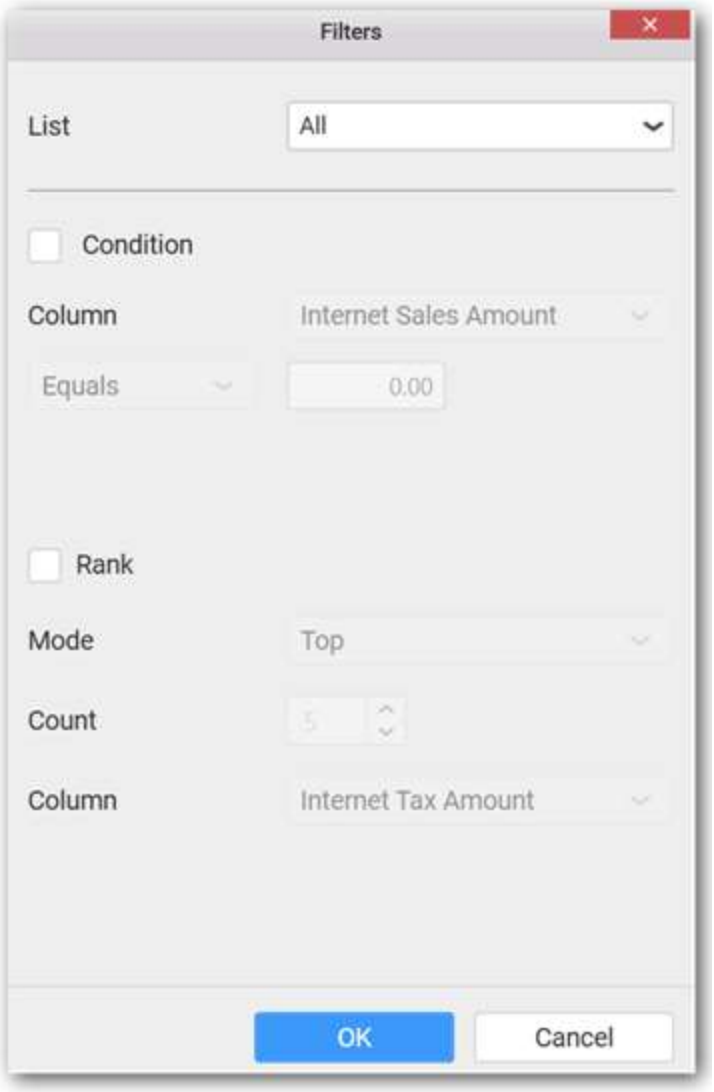
The drilled view of the chart is follows.



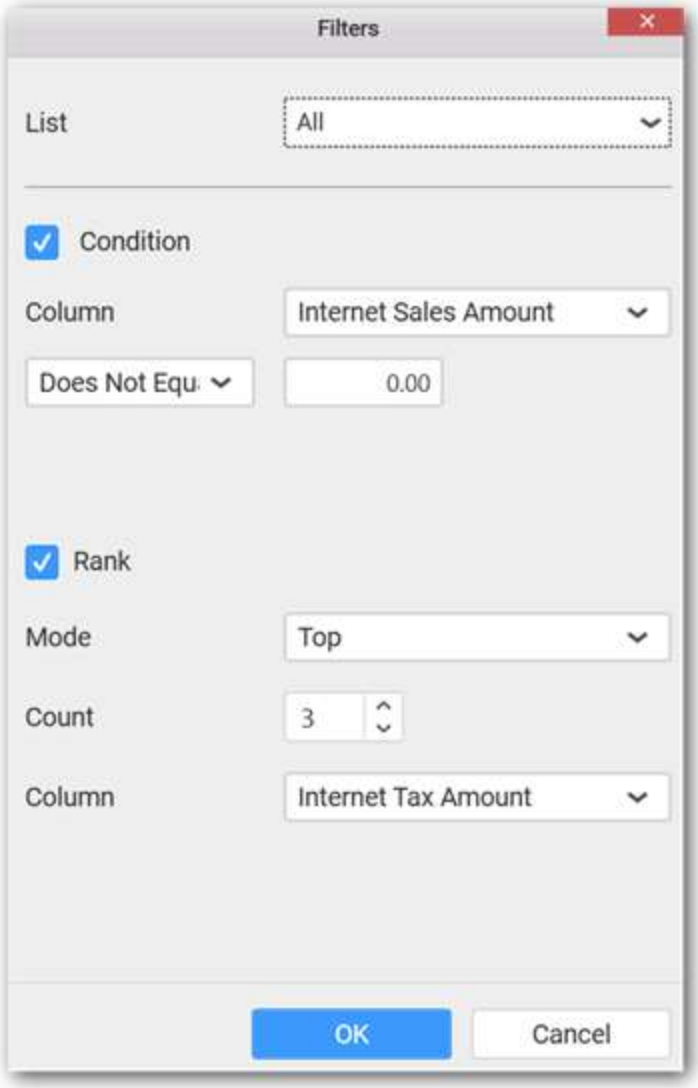
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



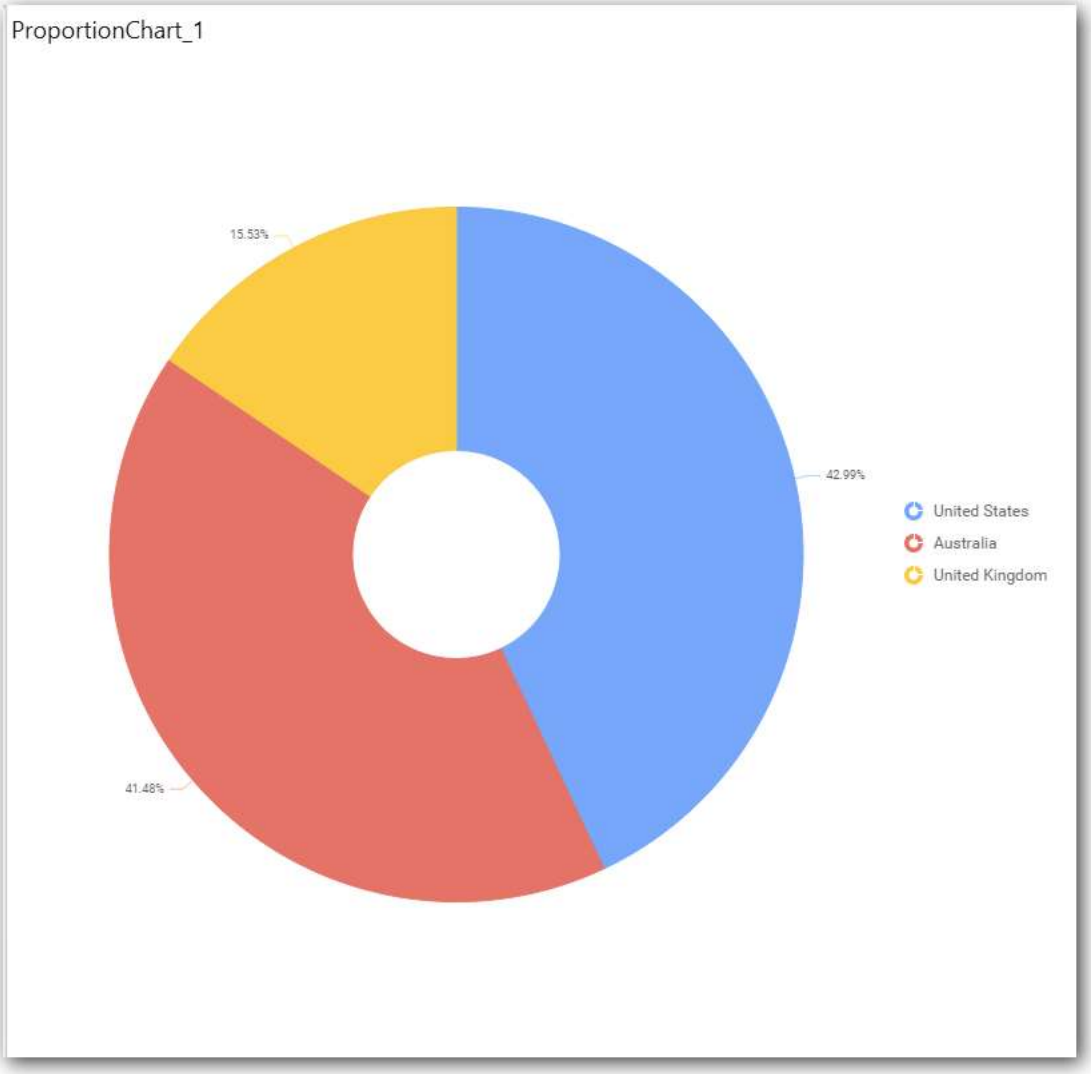
Select Filter(s)… to launch the Filters window.



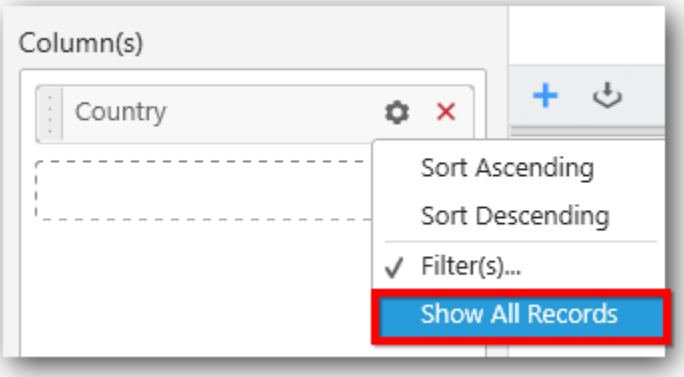
Define the filter condition and Rank and Click OK.



Now the chart will be rendered like this

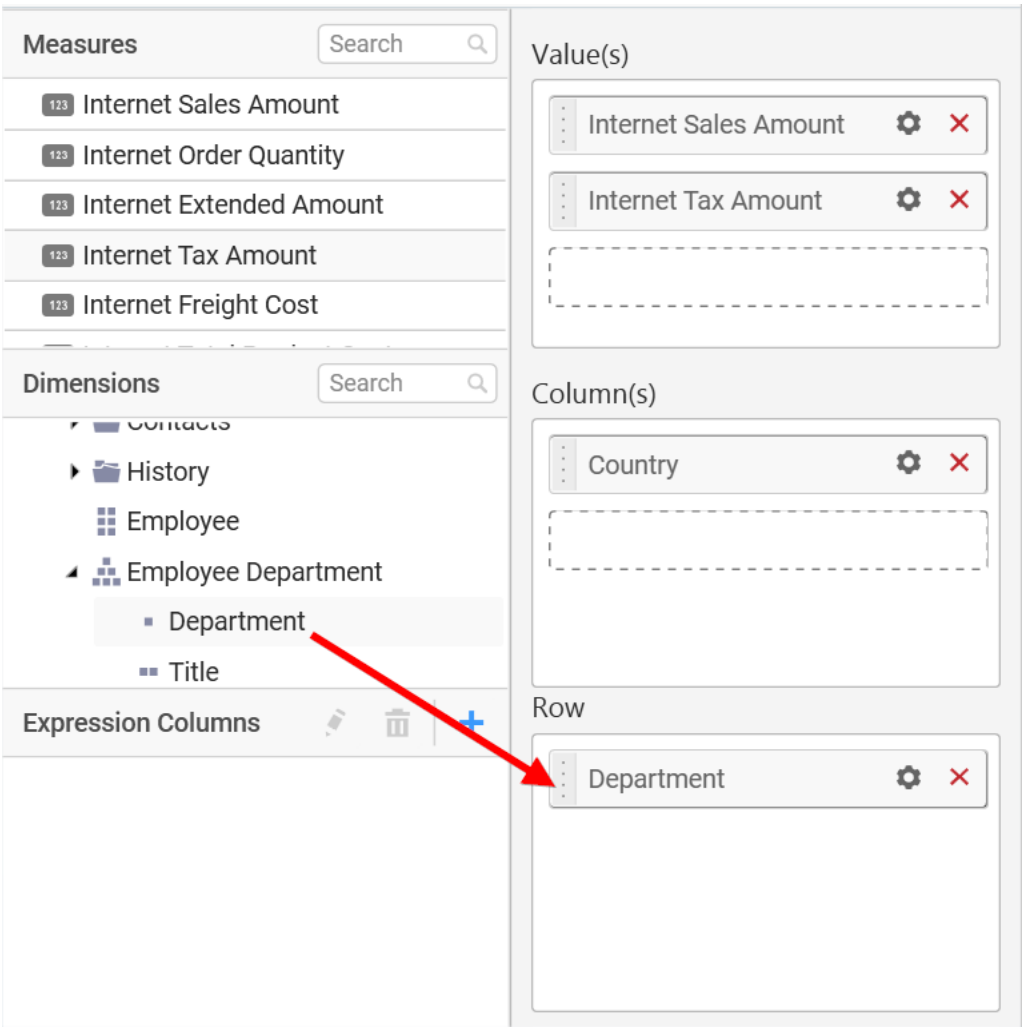


To show all records again click on Show All Records.



Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart.



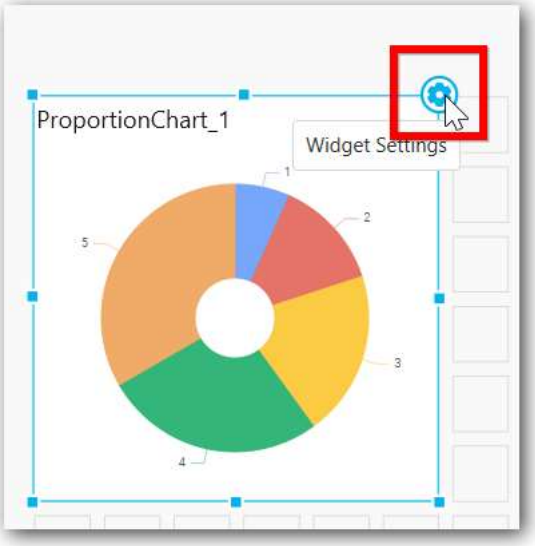
The chart will be rendered in series as shown in the image.

How to format Doughnut Chart?

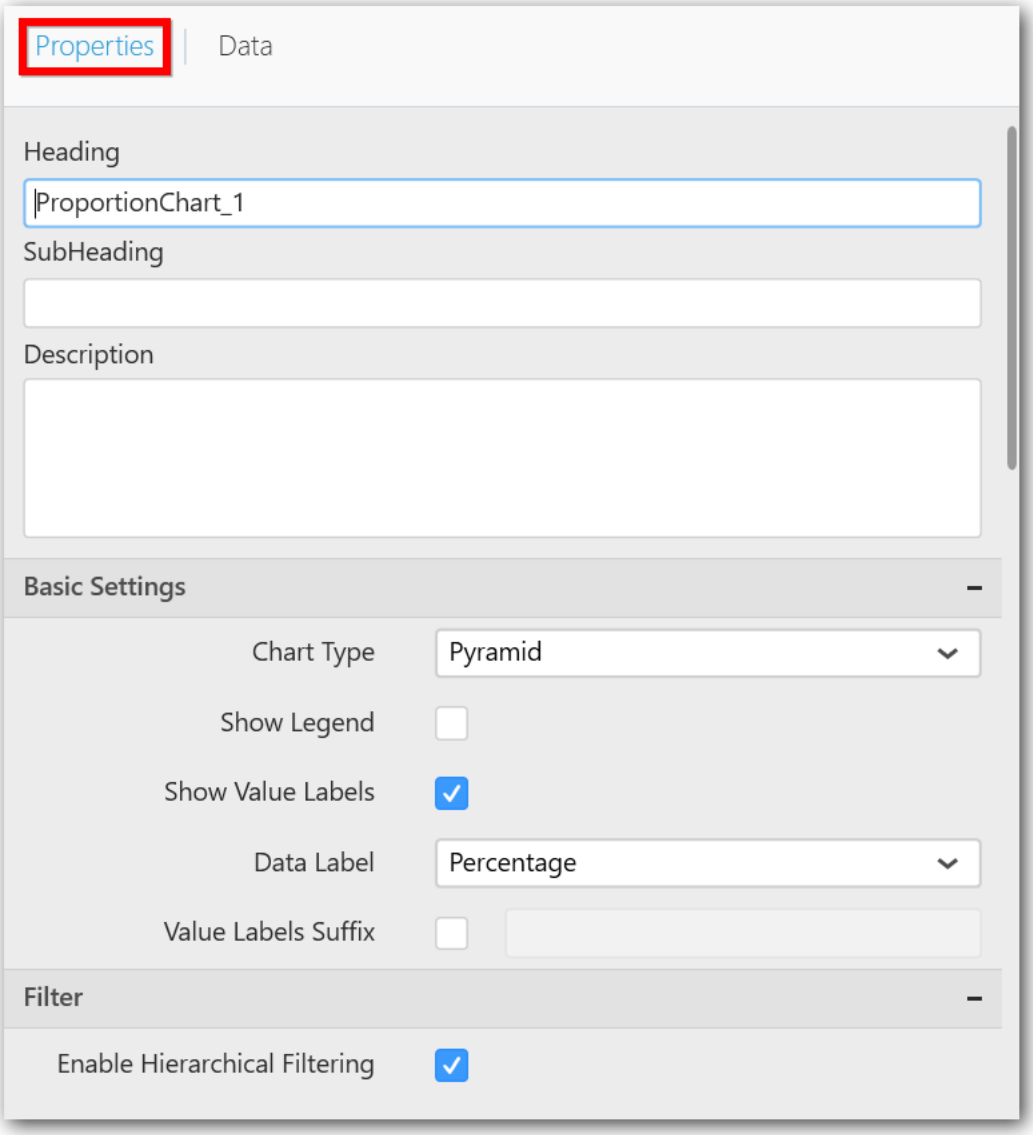
You can format the doughnut chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into doughnut chart follow the steps

1. Drag and drop the doughnut chart into canvas and resize it to your required size.
2. Configure the data into doughnut chart.
3. Focus on the doughnut chart and Click on Widget Settings.

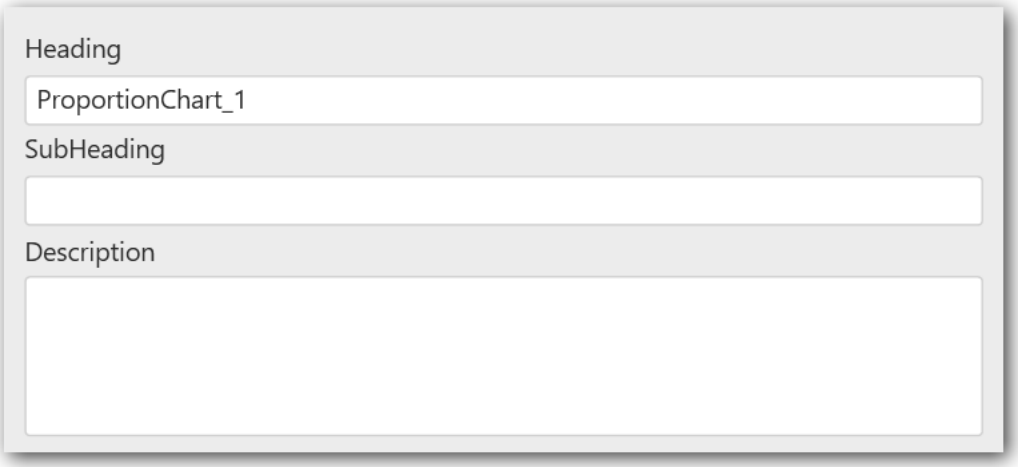


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this doughnut chart widget.

SubHeading

This allows you to set sub-title for this doughnut chart widget.

Description

This allows you to set description for this doughnut chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

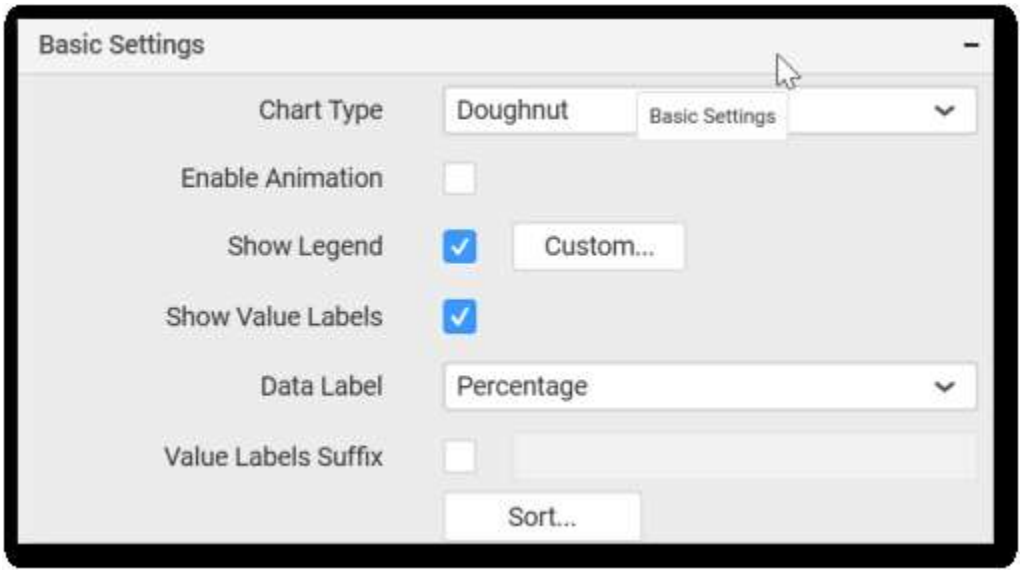


Chart Type

This allows you to switch the widget view from current chart type to another chart type.

Enable Animation

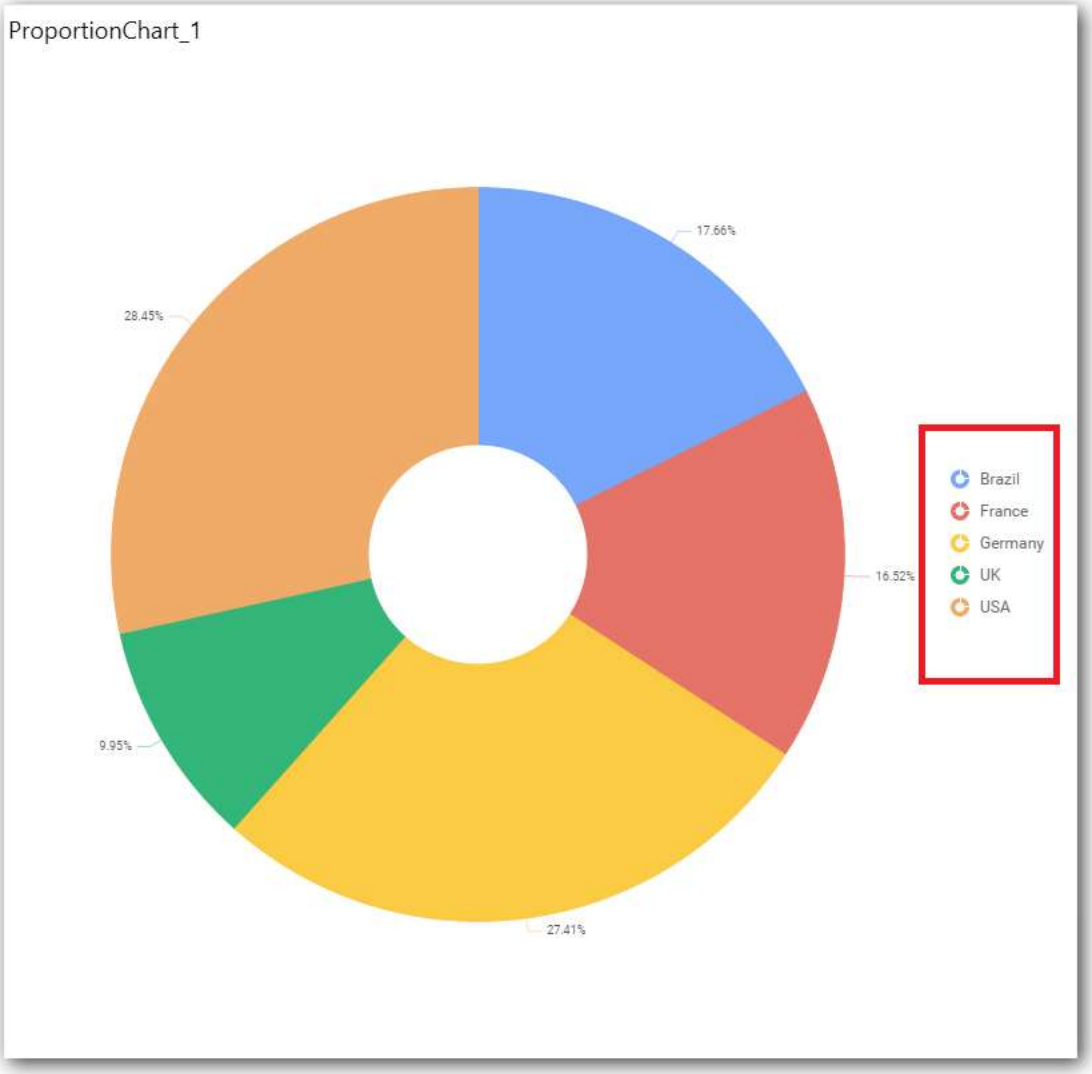
This allows you to enable the series rendering in animated mode.

Enable Drill Down

This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Show Legend

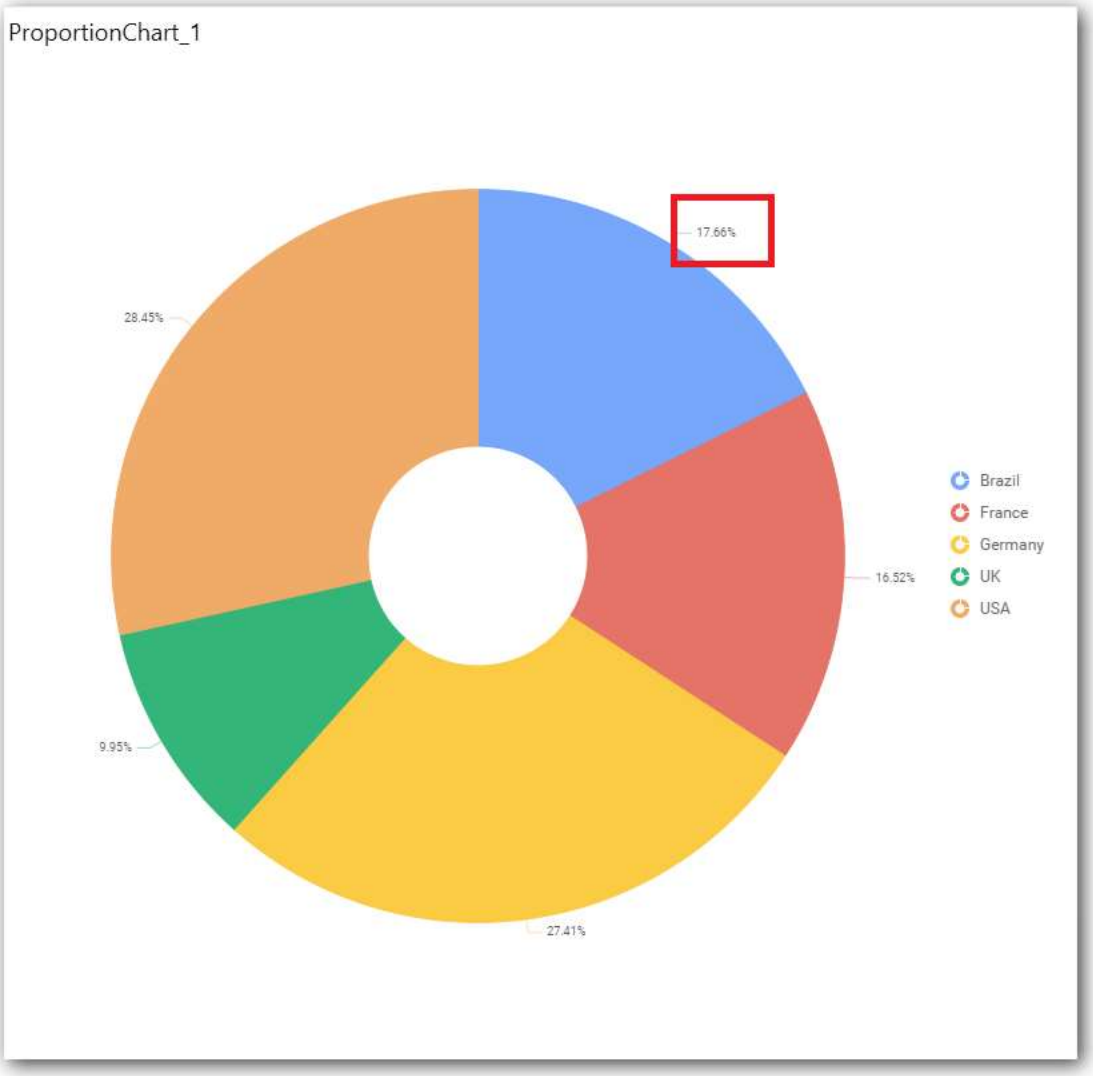
This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Show Value Labels

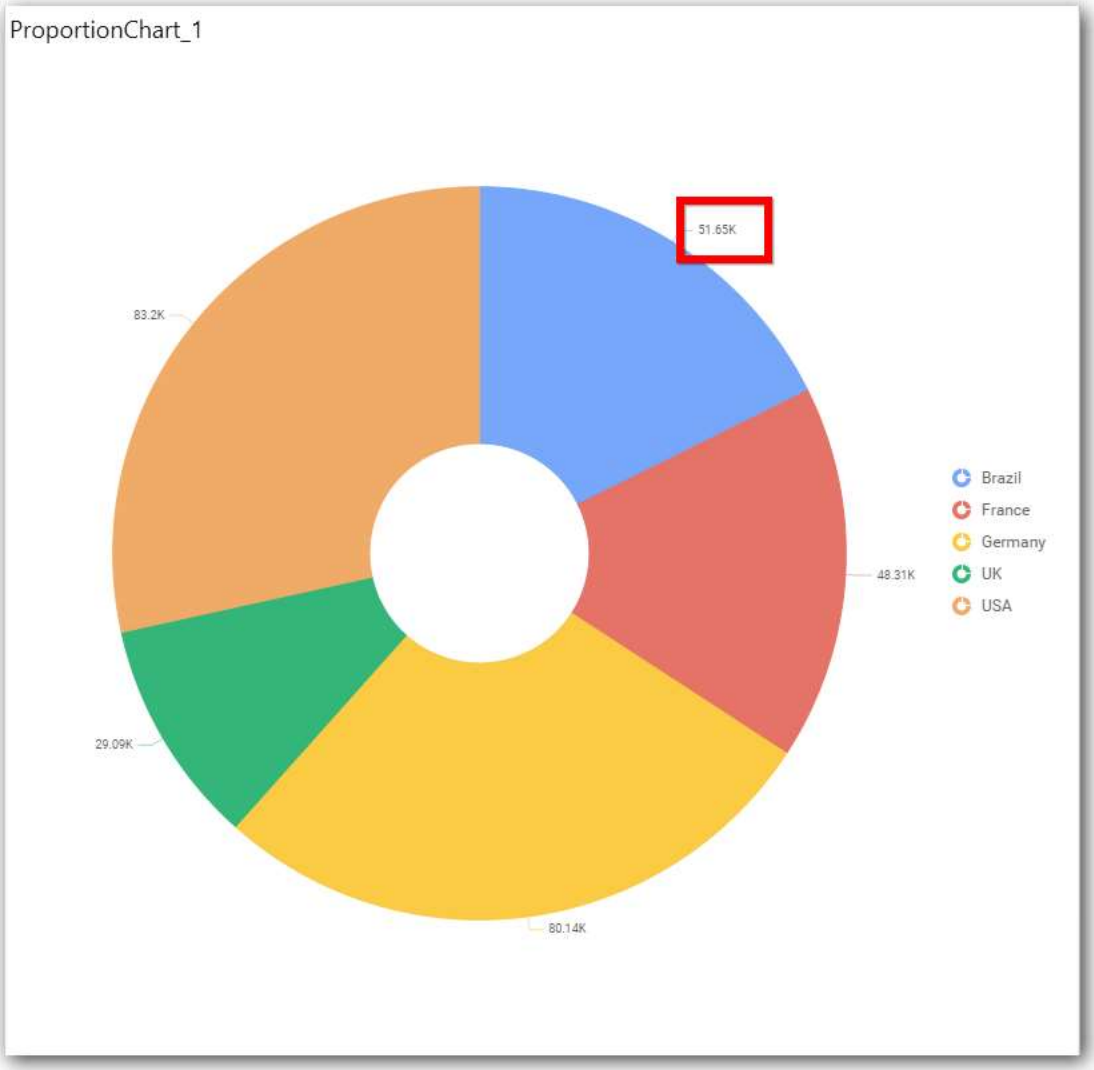
This allows you to toggle the visibility of value labels.



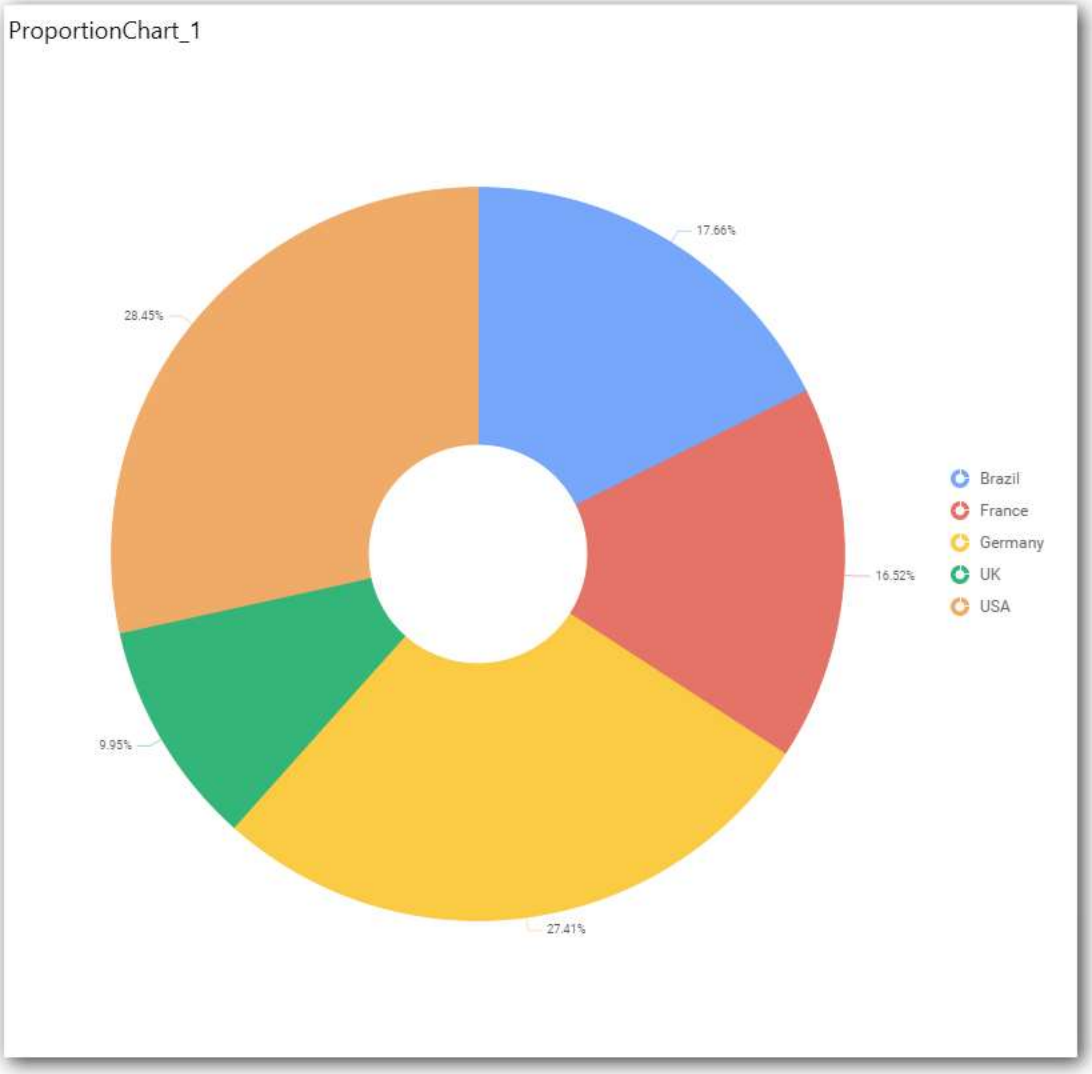
Data Label

This allows you to define the display format either as value or as percentage.

Value

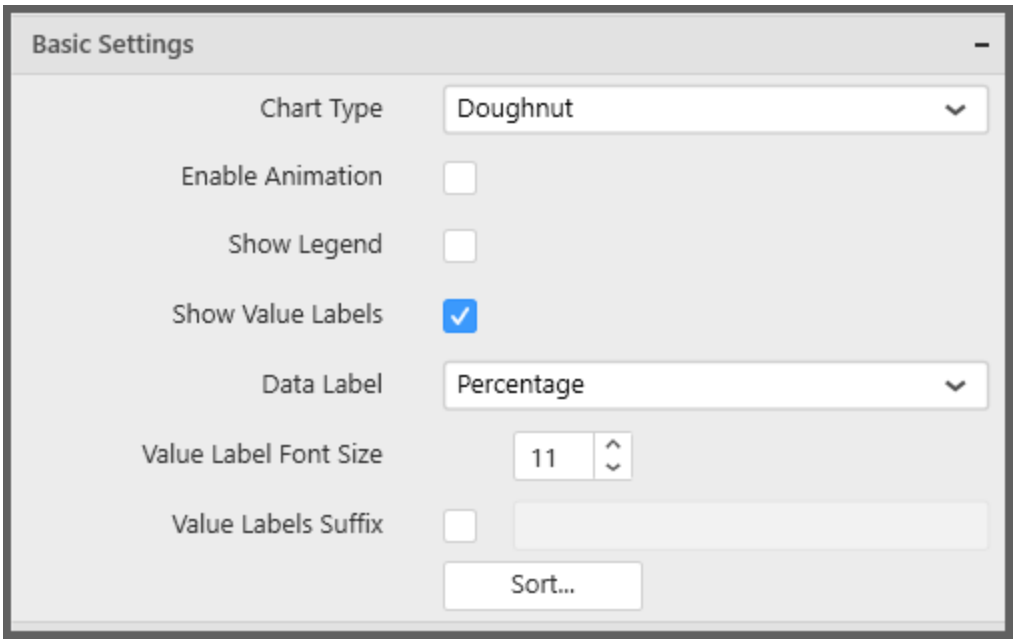


Percentage



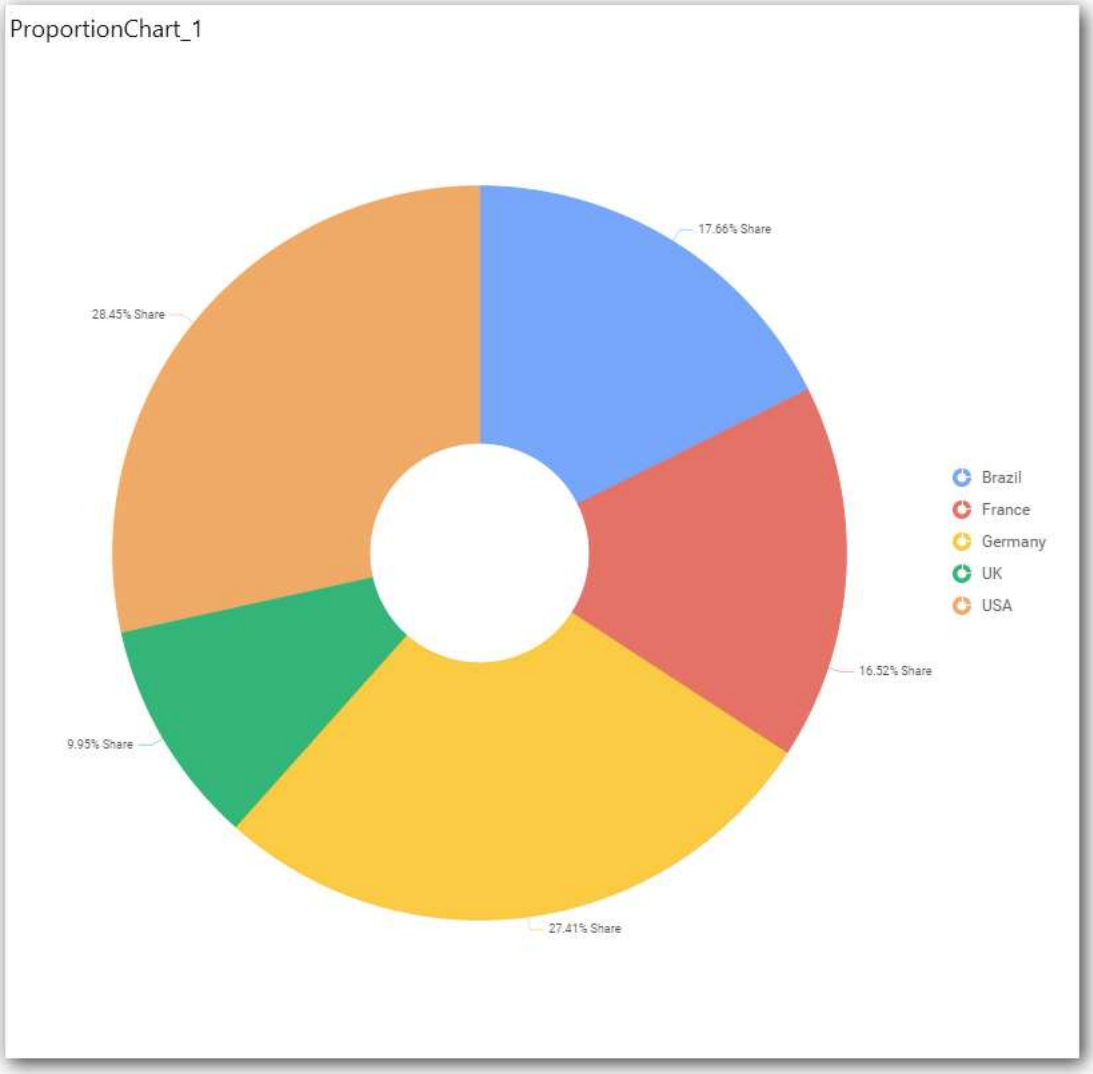
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.



Value Labels Suffix

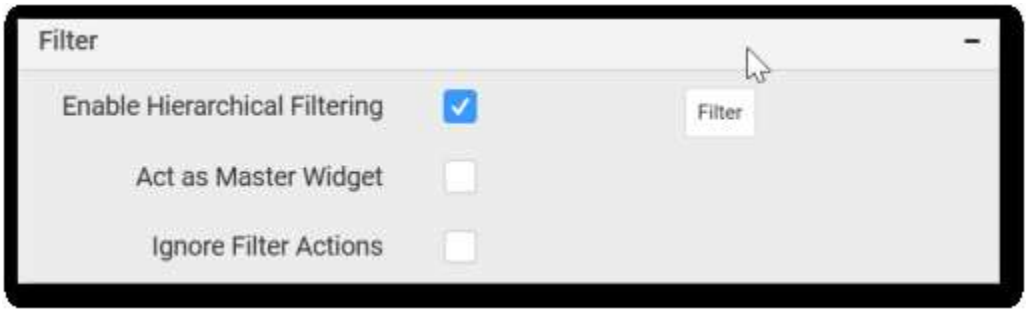
Allows you to set suffix to the value labels.



Sort Order

This allows you to define the sort order for each measure column added.

Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

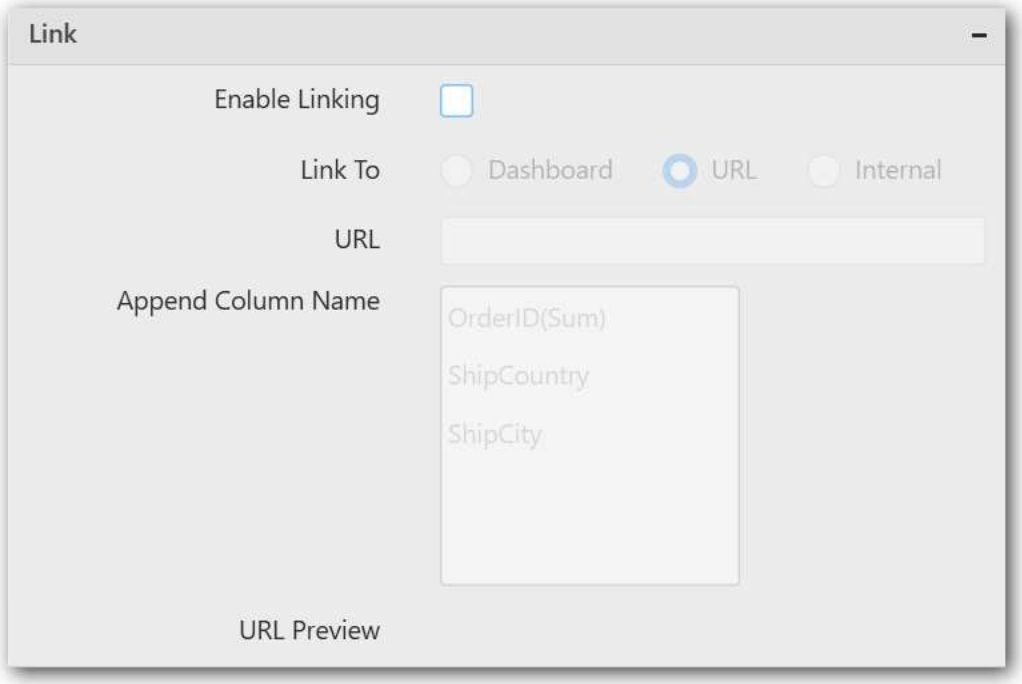
Act as Master Widget

This allows you to define this doughnut chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

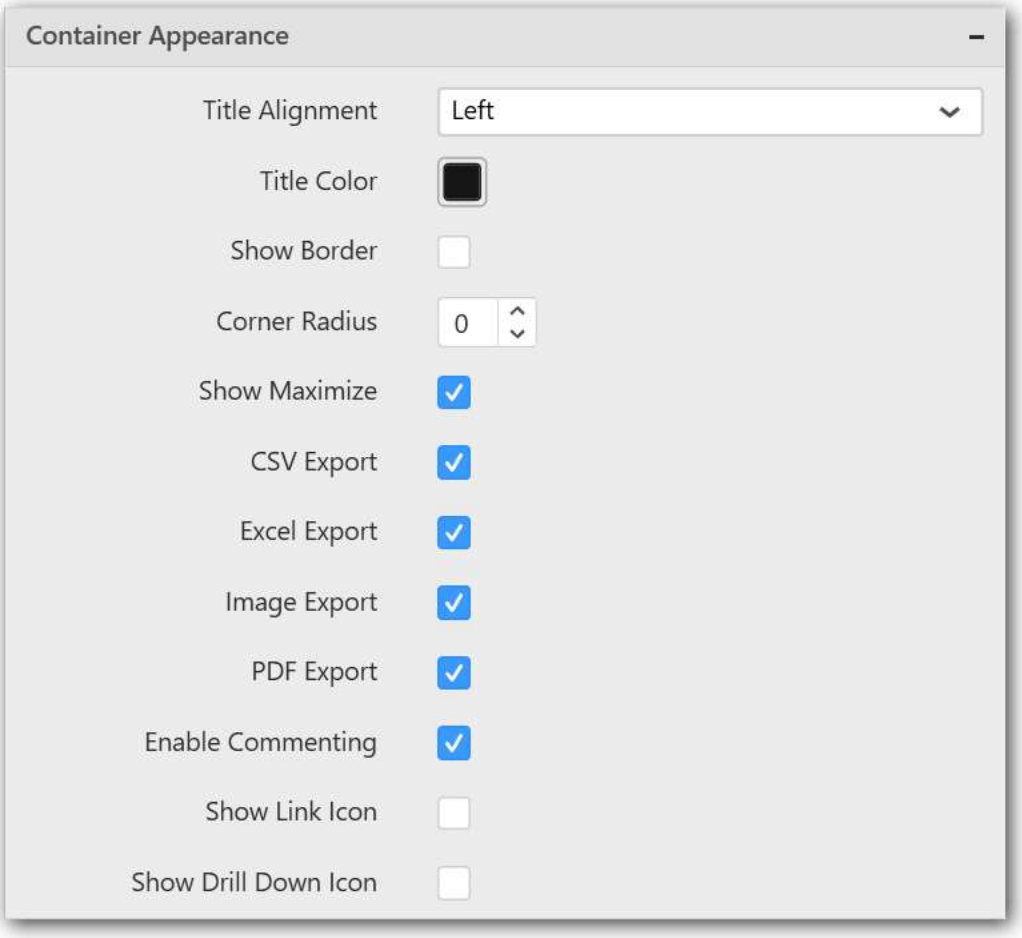
This allows you to define this doughnut chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this doughnut chart widget. The visibility of the maximize icon in widget header will be defined based on this setting.

CSV Export

This allows you to enable/disable the CSV export option for this doughnut chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this doughnut chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

Image Export

This allows you to enable/disable the image export option for this doughnut chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

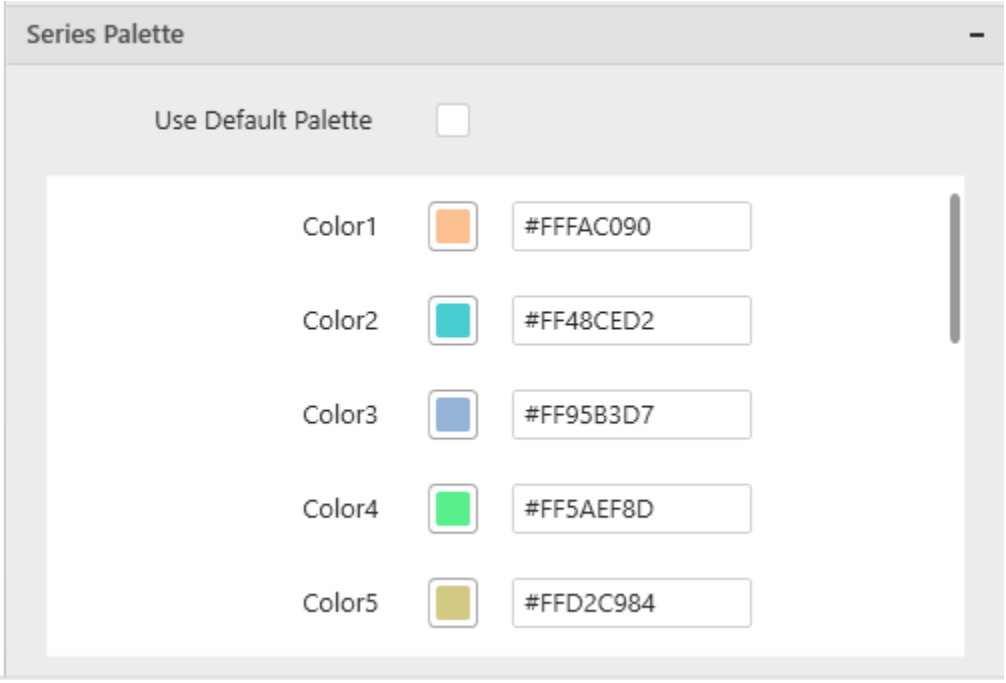
This allows you to enable/disable comment for dashboard widget. For more details refer here

Series Palette

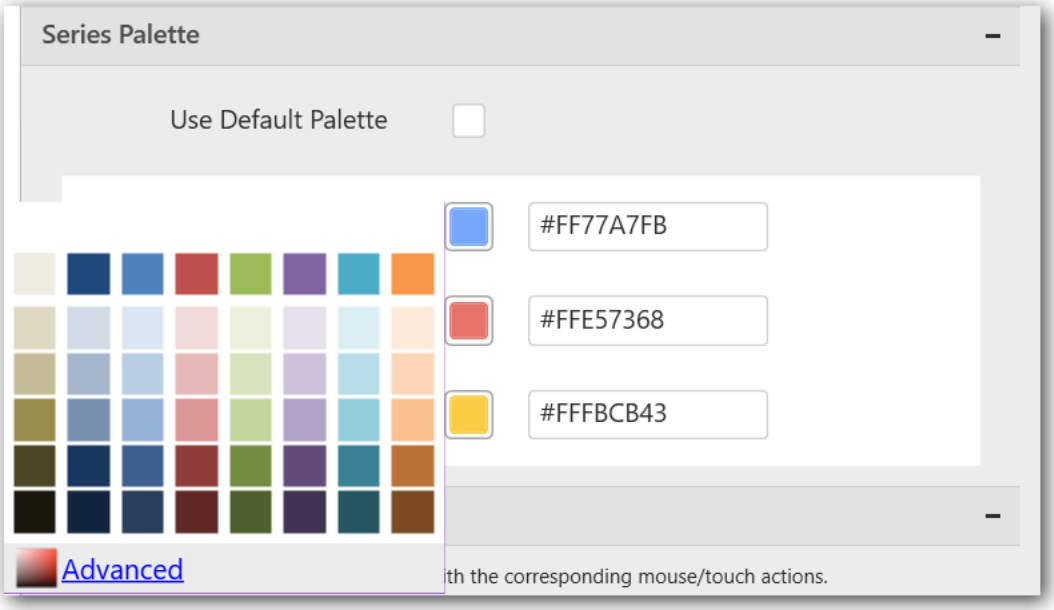
This allows you to customize the chart series color through Series Palette section.

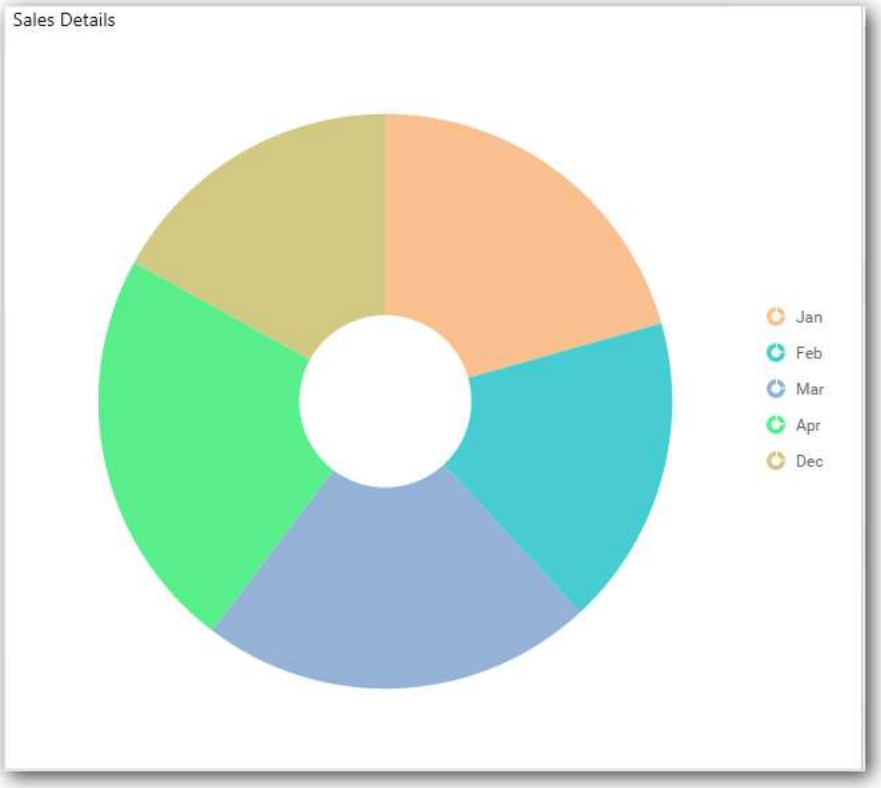
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to proportion series segments.



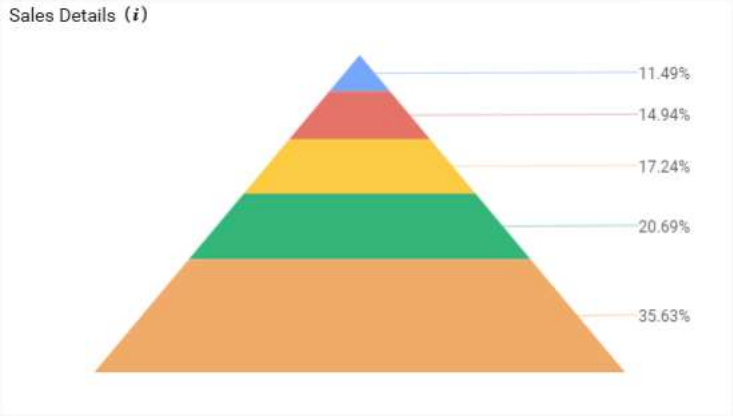
By toggle off the Use Default Palette, you can customize the proportion series segments' colors. This section shows a palette of colors. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Pyramid Chart

Pyramid Chart allows you to make proportional comparison between values showcased as progressively increasing manner. To plot a pyramid chart, a minimum requirement of 1 value and 1 column is needed.

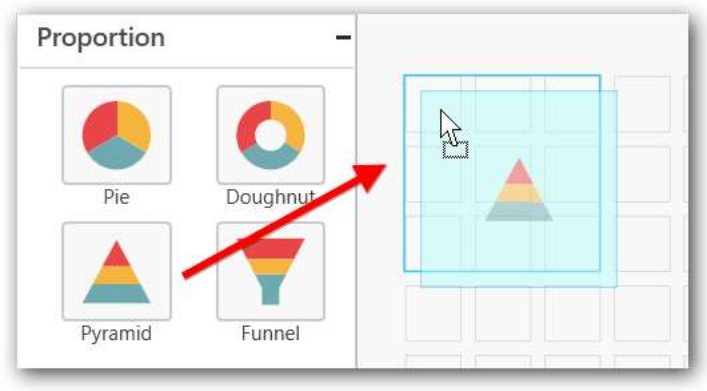


How to configure the flat table data to Pyramid Chart?

Pyramid Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

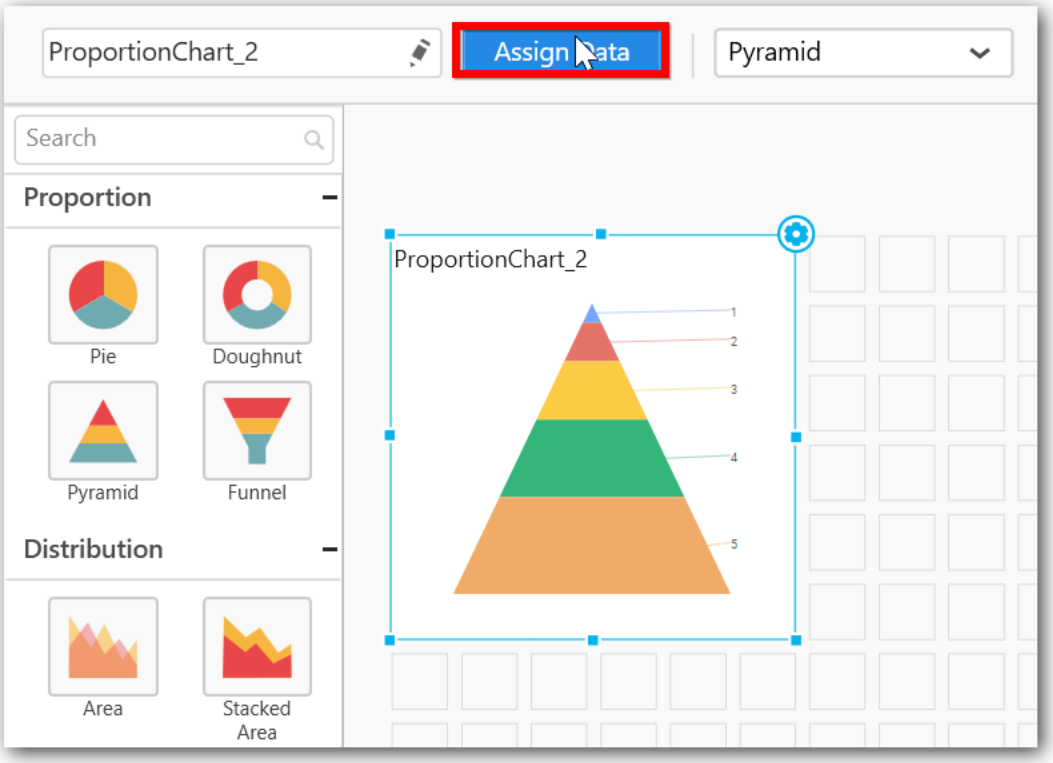
To configure data into pyramid chart follow the steps

Drag and drop the Pyramid chart into canvas and resize it to your required size.

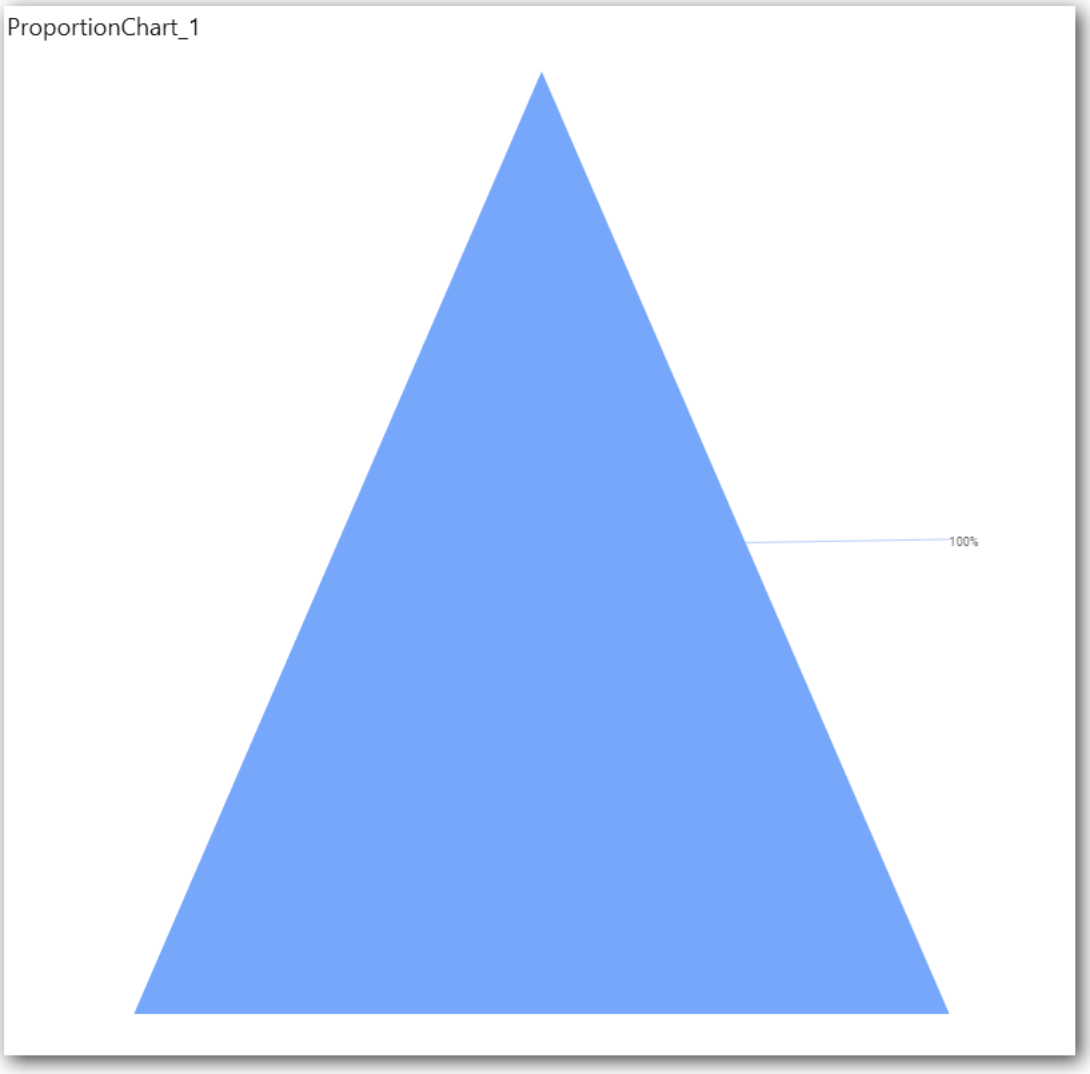


Connect to the Data source.

Focus on the dropped chart and Click on Assign Data.



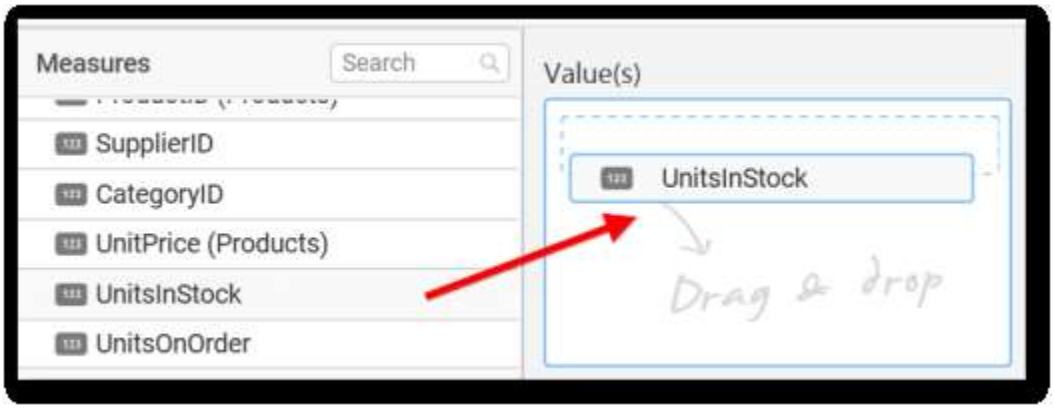
The following data pane will appear with available Measures and Dimensions from the connected data source.



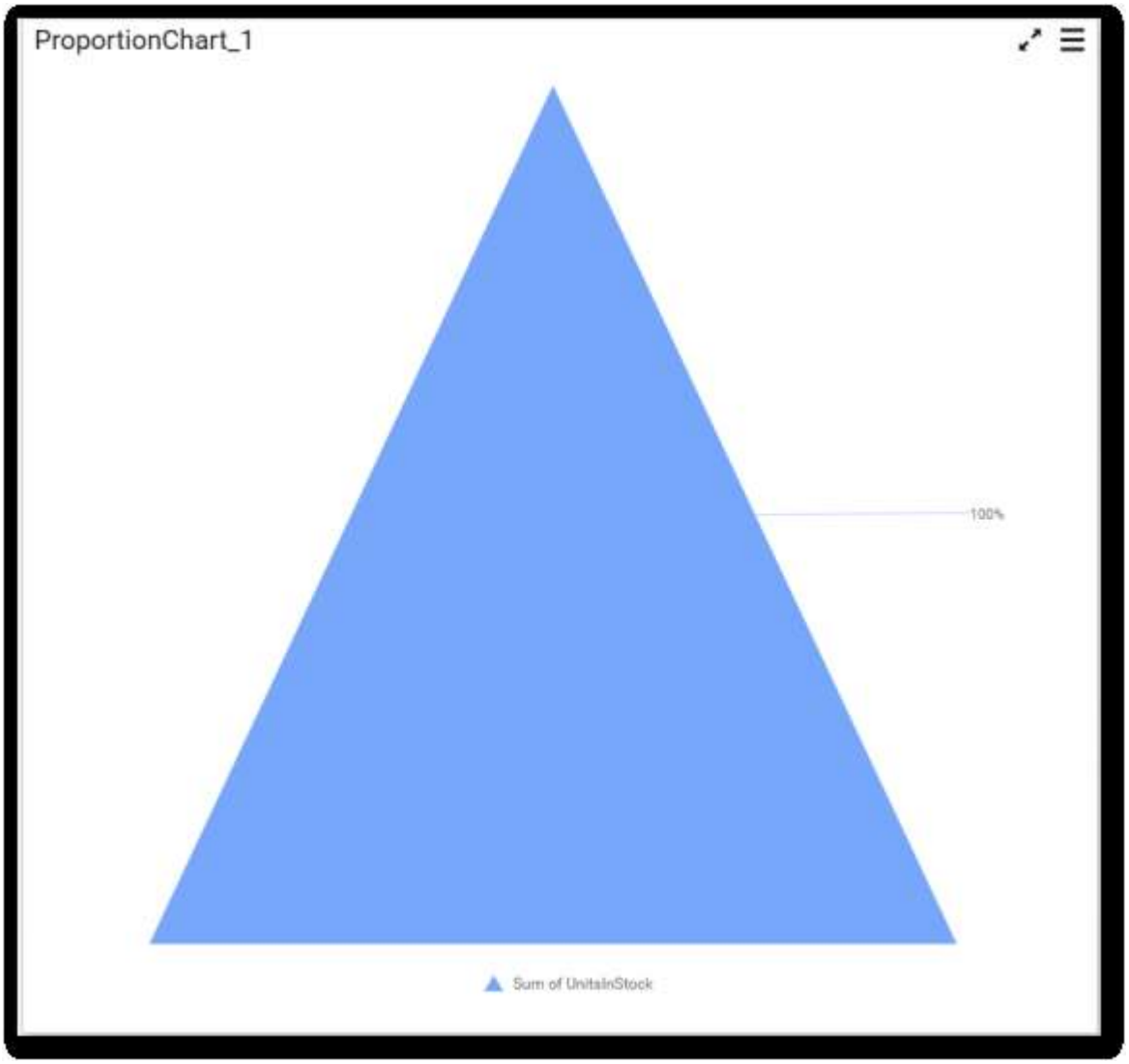
You can add the required data from Measures and Dimensions into required field. You can also create Expression Columns using Expression Designer.

Adding Value(s)

You can add Measures into Value(s) field by drag and drop the required measure.

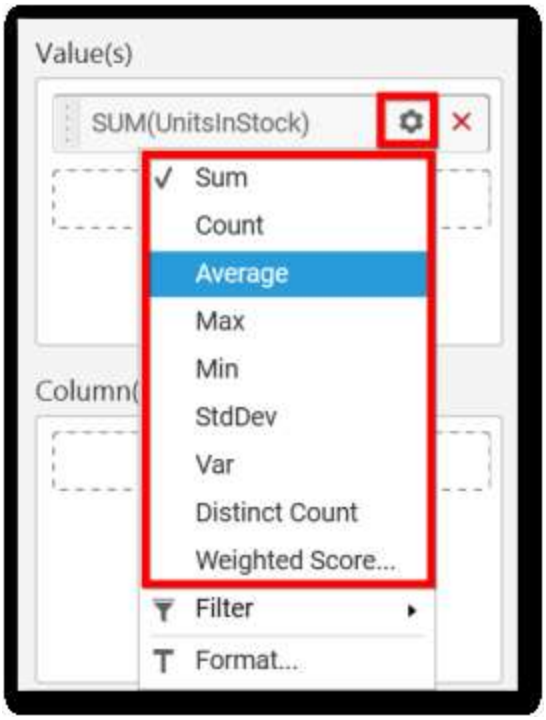


Now the Pyramid chart will be rendered like this

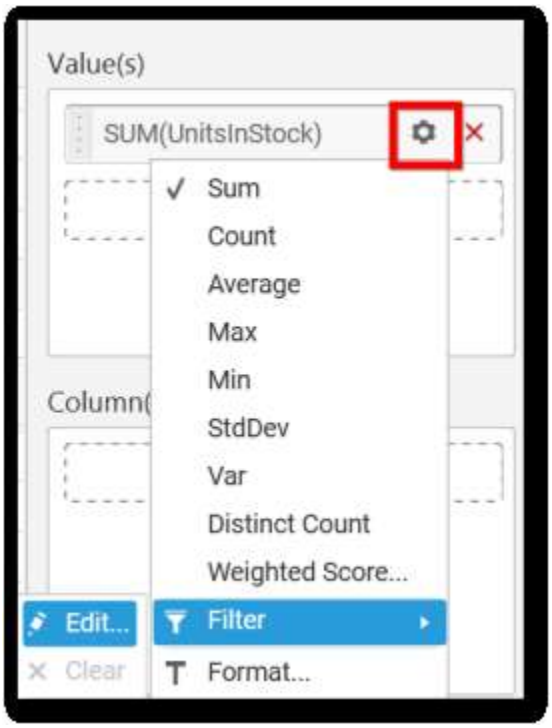


You can change the settings option.

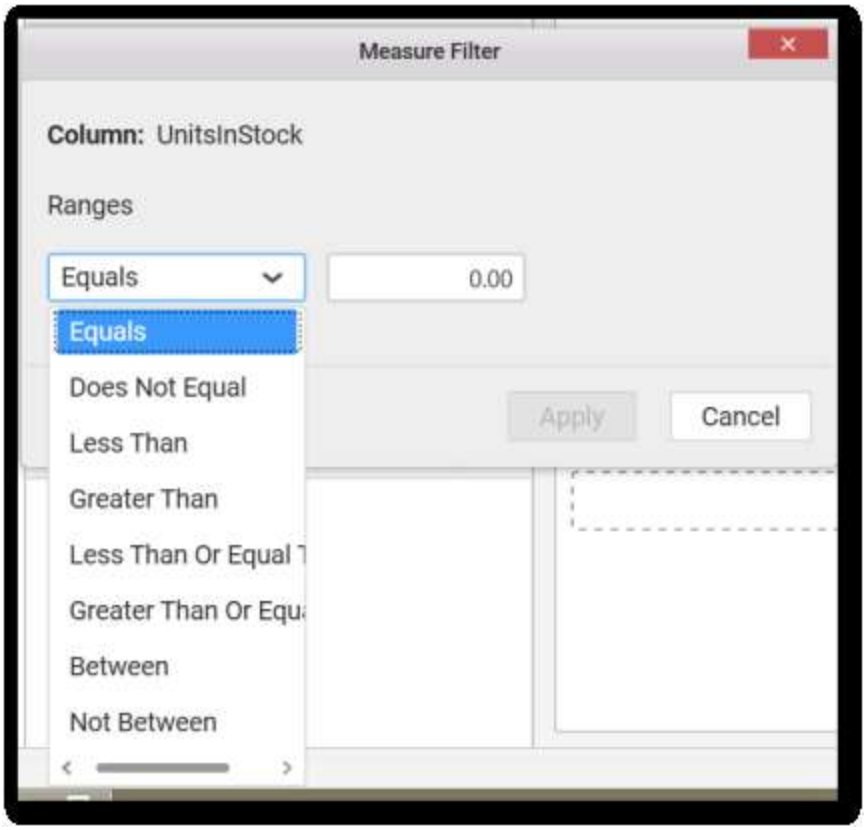
You can select the required summary type from the available summary types shown in settings.



You can filter the data to be displayed in pyramid chart by using Filter option.



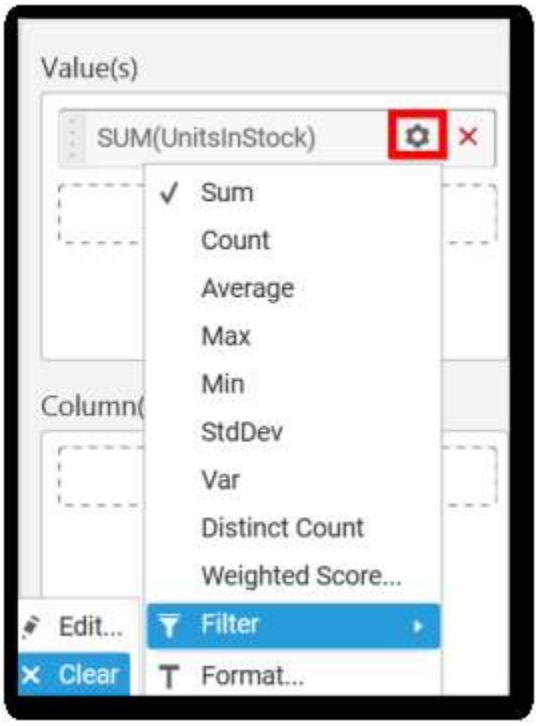
When you click the Measure Filter option will appear.



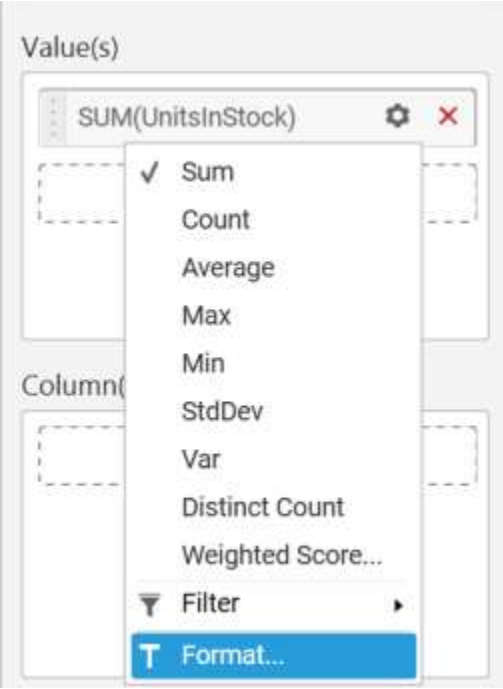
You can select the Condition to be applied in the shown list box and set the value in text box.

Click on Apply to see the changes.

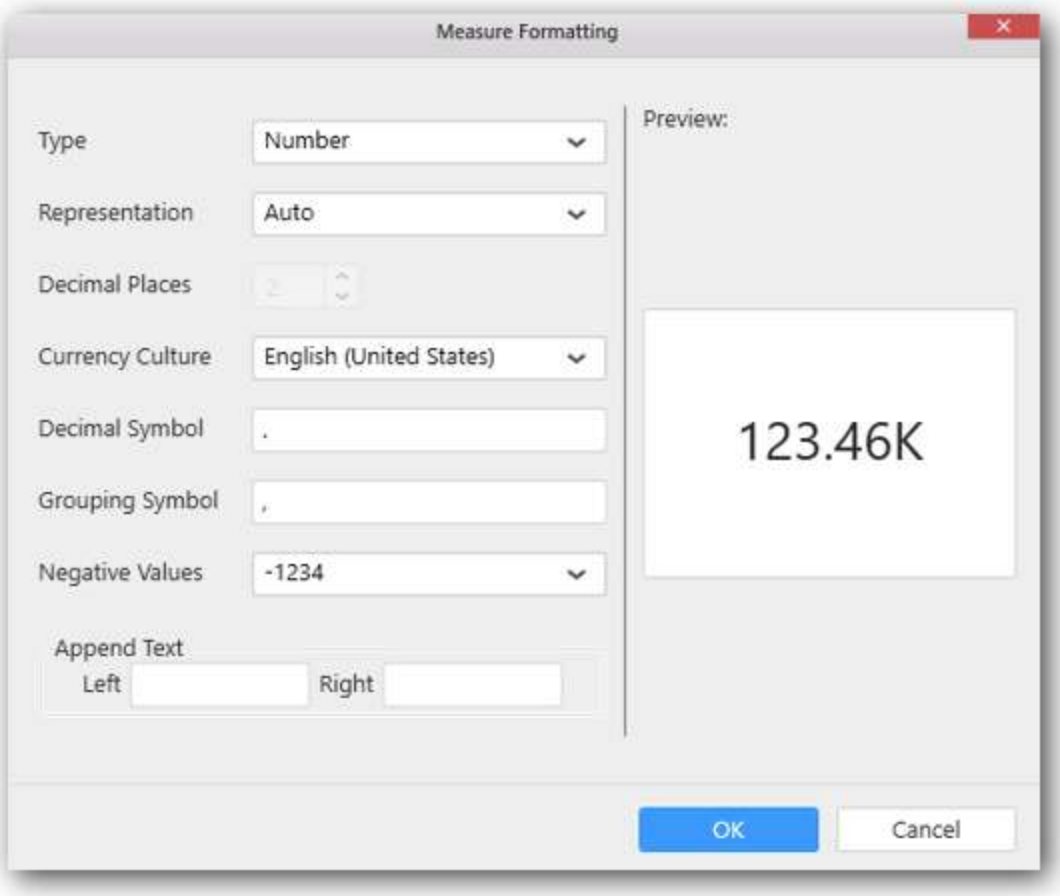
You have option to clear the applied filter. Click on clear to remove the filters

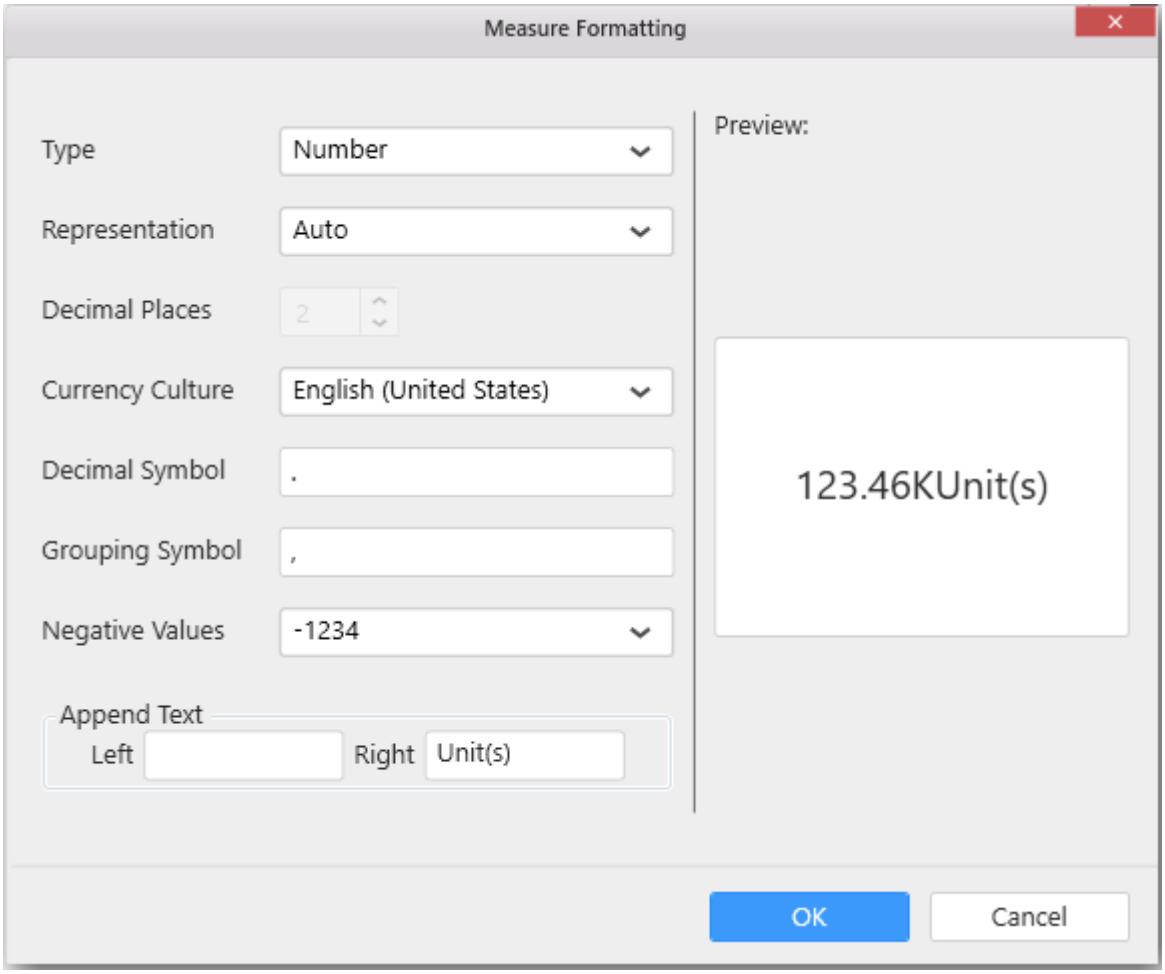


You can format the data to be displayed in the Pyramid chart by using format option.

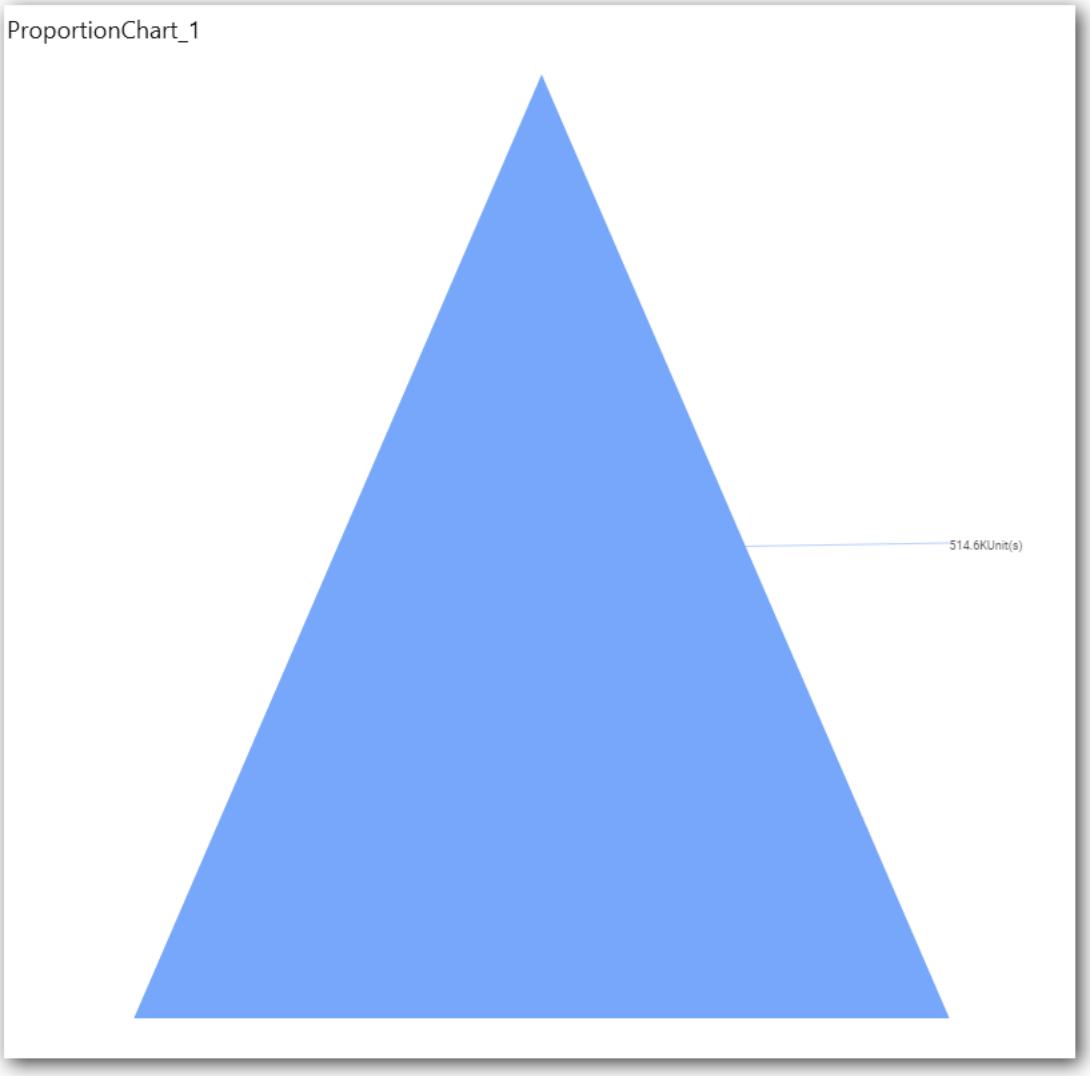


The Measure Formatting dialog will be shown, select the format that you want and click OK.

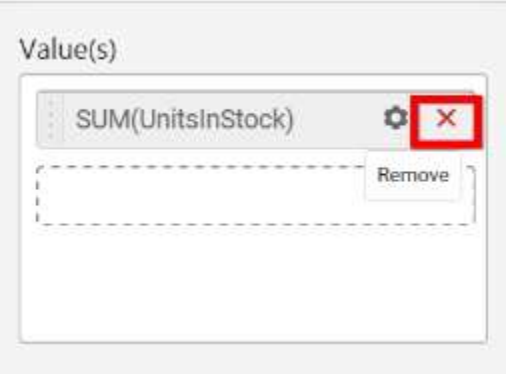

Measure Formatting
Type
Number
Representation
Auto
Decimal Places
Currency Culture
English (United States)
Decimal Symbol
.
Grouping Symbol
,
Negative Values
-1234
Append Text
Left
Right
Preview:
123.46K
OK
Cancel



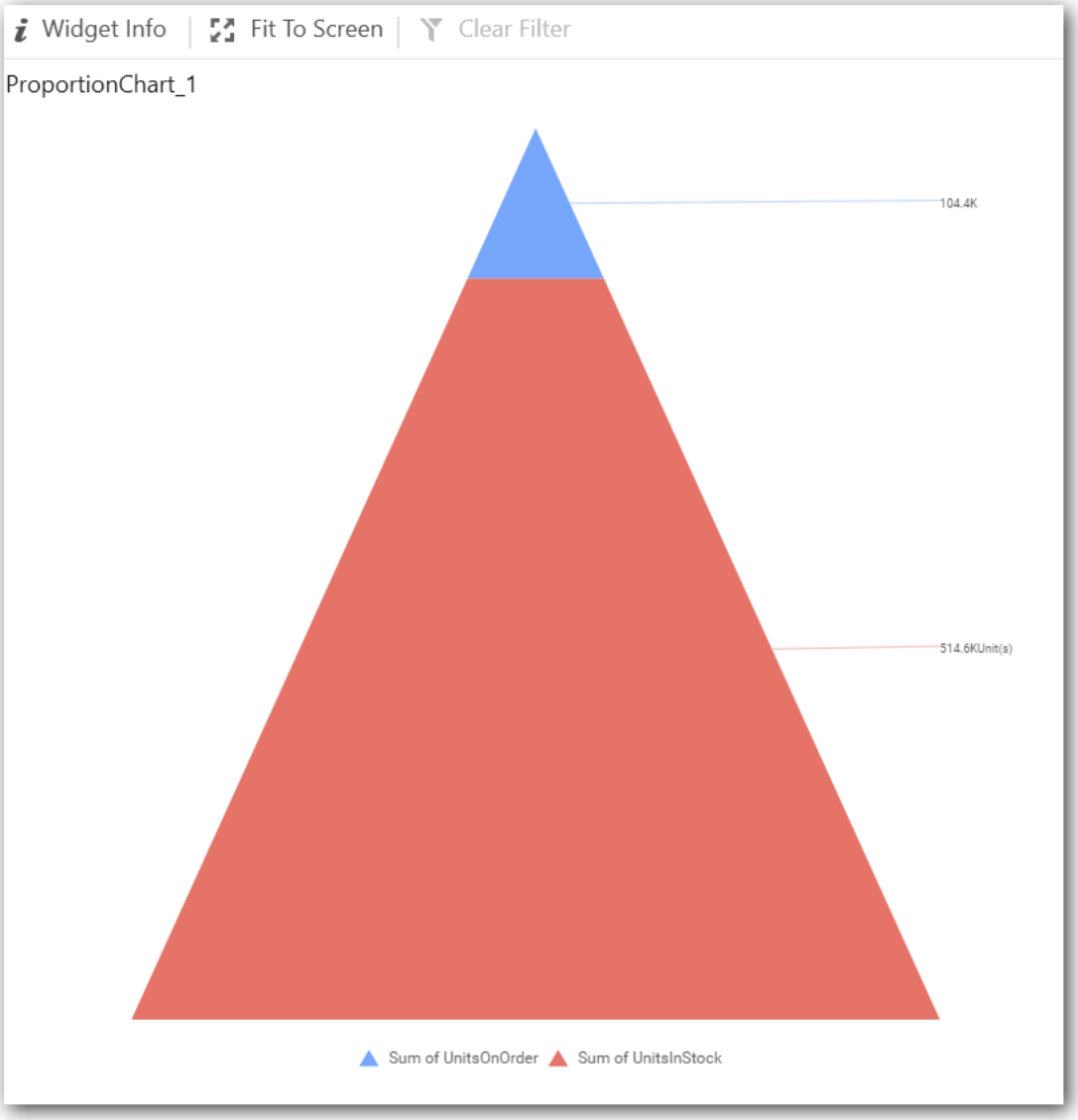
Here is an illustration,



To remove the added value fields click on x button.



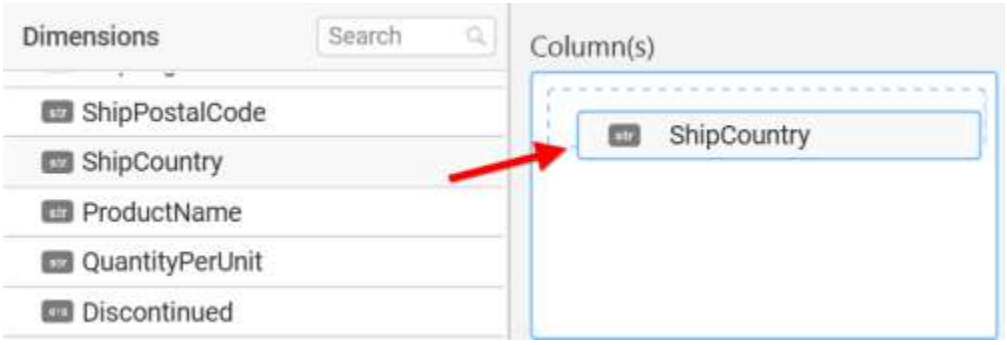
You can add multiple Measures into Value(s).



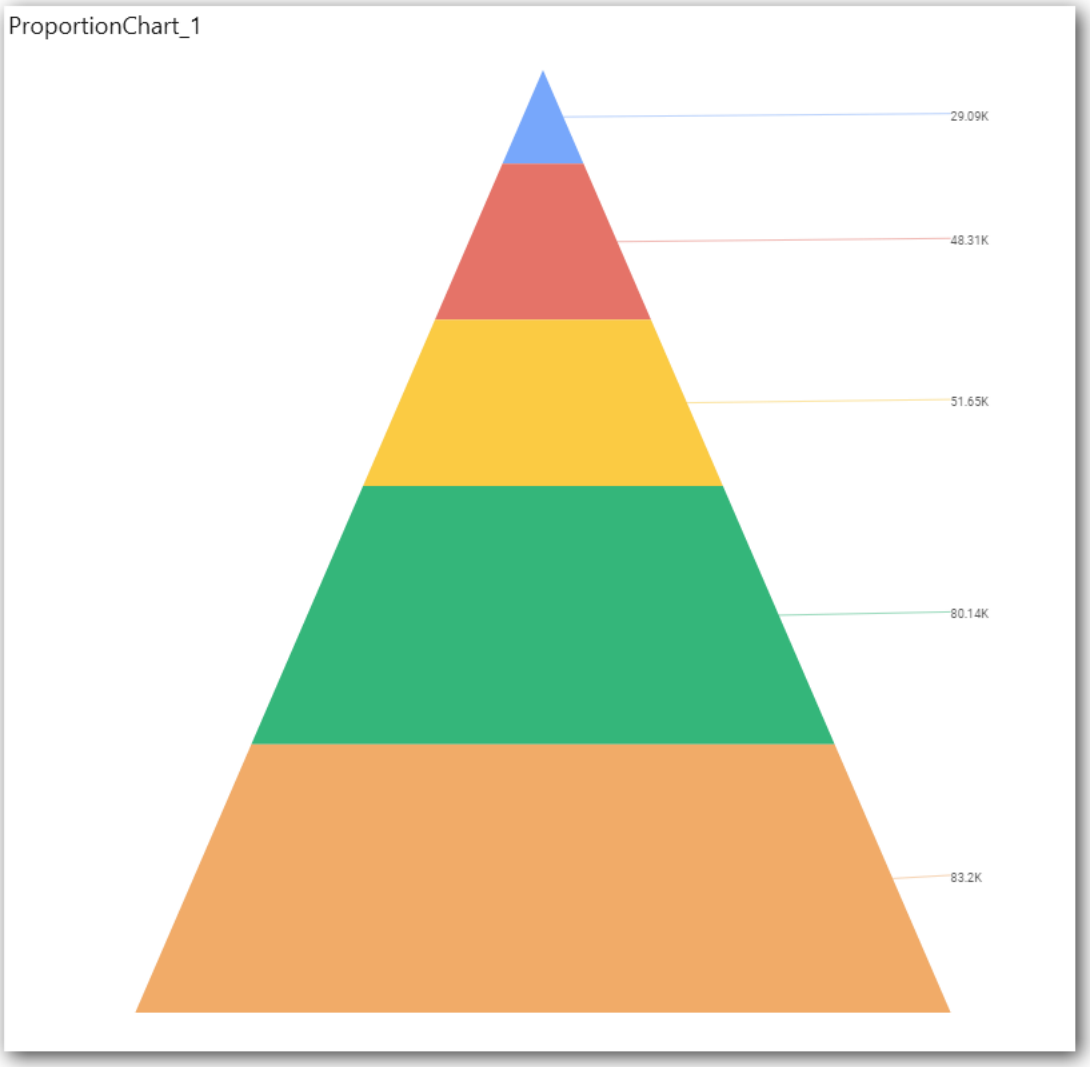
You can also drag and drop Dimension and Expression Column into Value(s).

Adding Column(s)

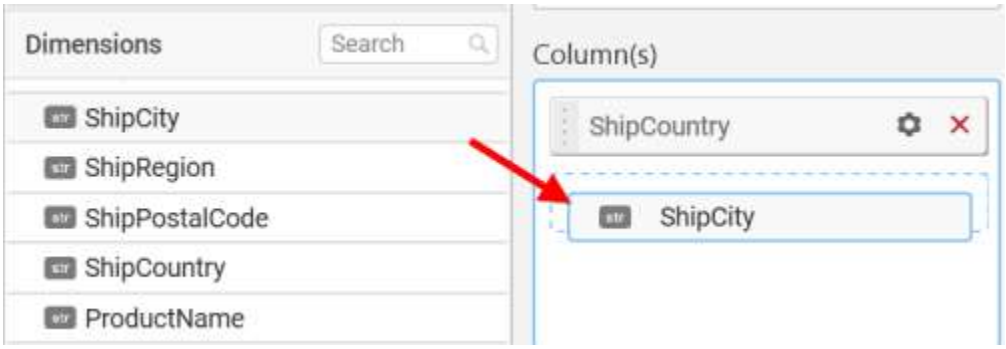
Drag and drop the Dimensions into Column(s).



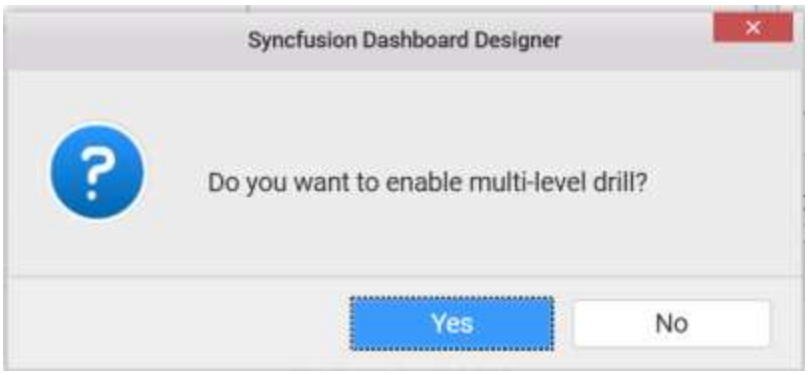
Pyramid chart will be rendered like this



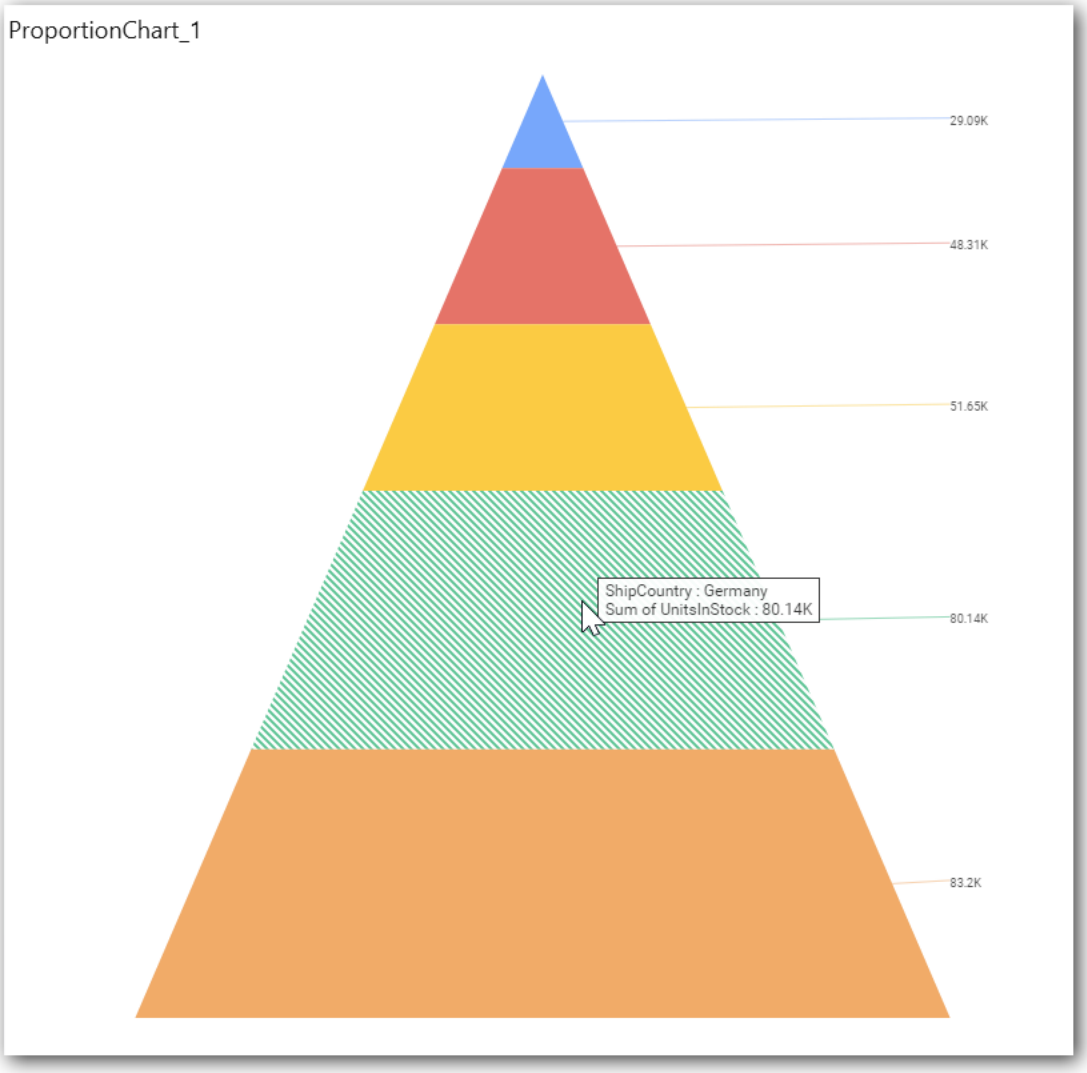
You can add more than one value into Column(s) field.



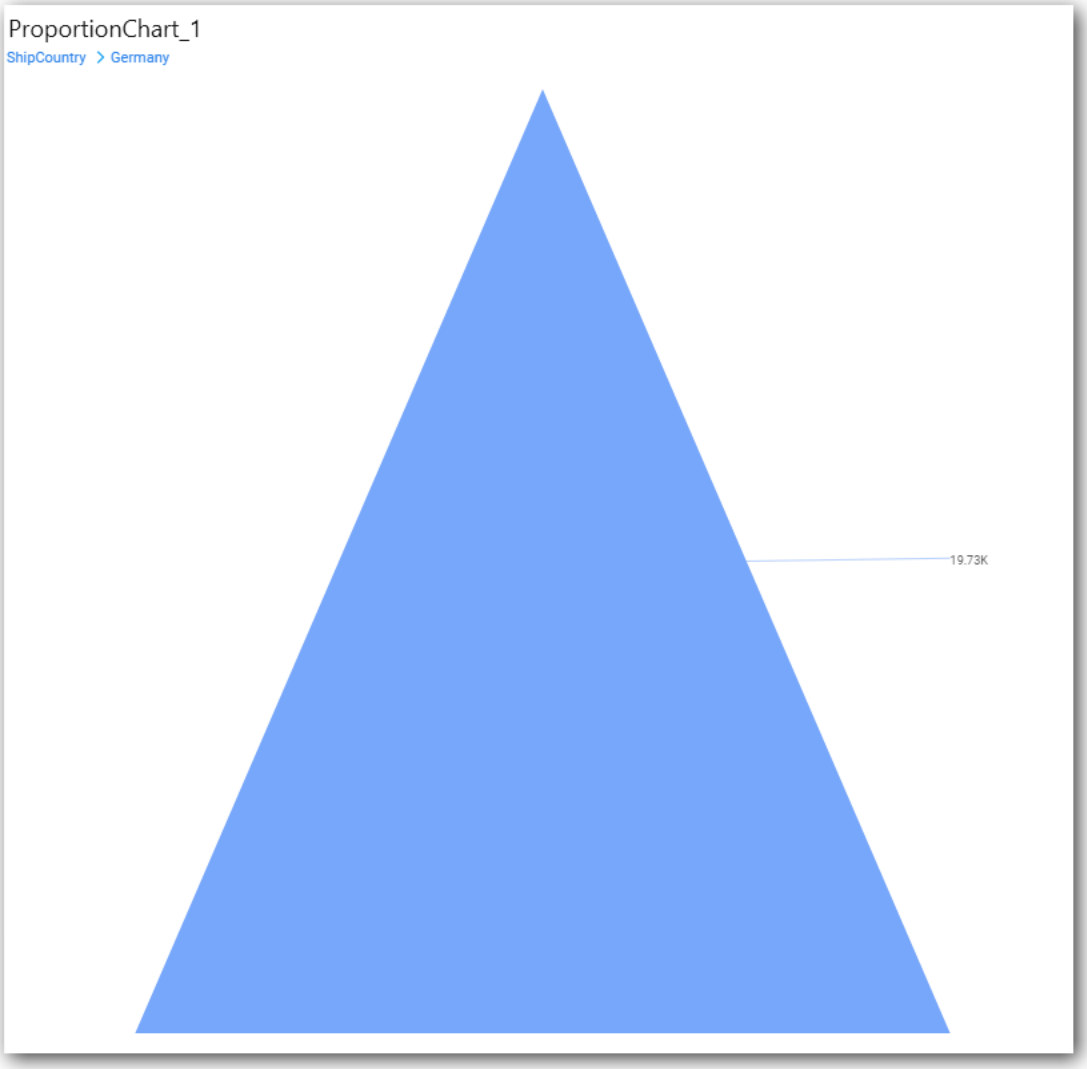
The following message will open



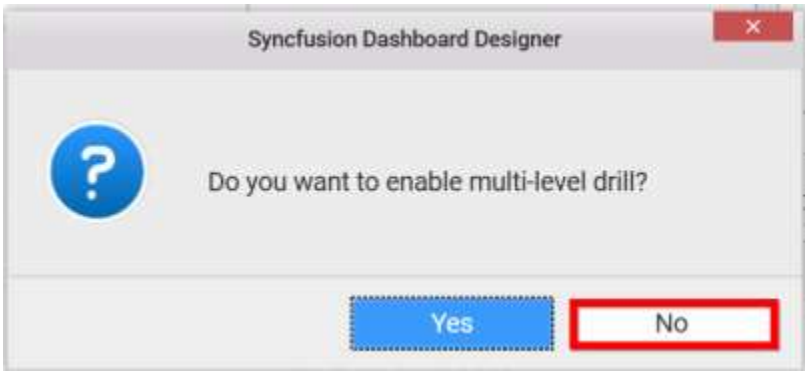
To enable drill down click Yes.



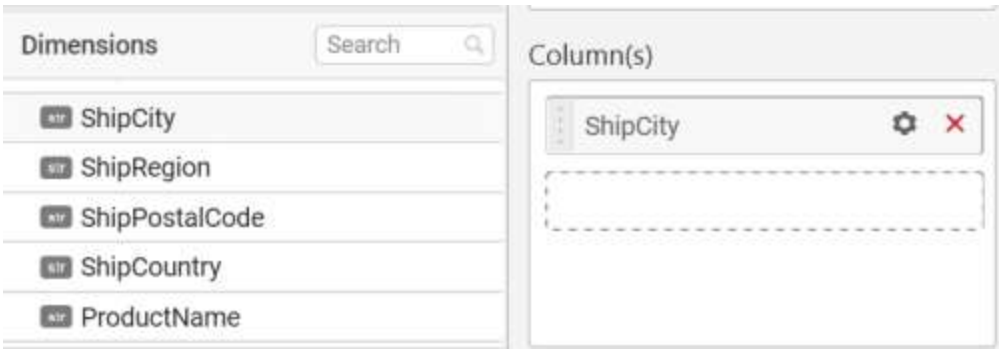
The drilled view of the chart region selected.



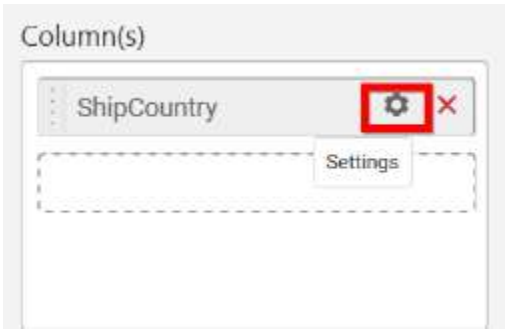
To continue without drill down click NO.



The old column value will be replaced by new column value

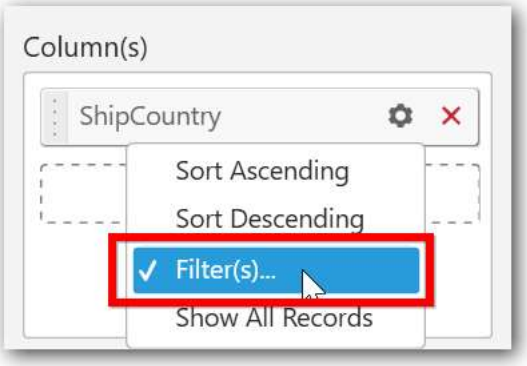


You can change the settings.



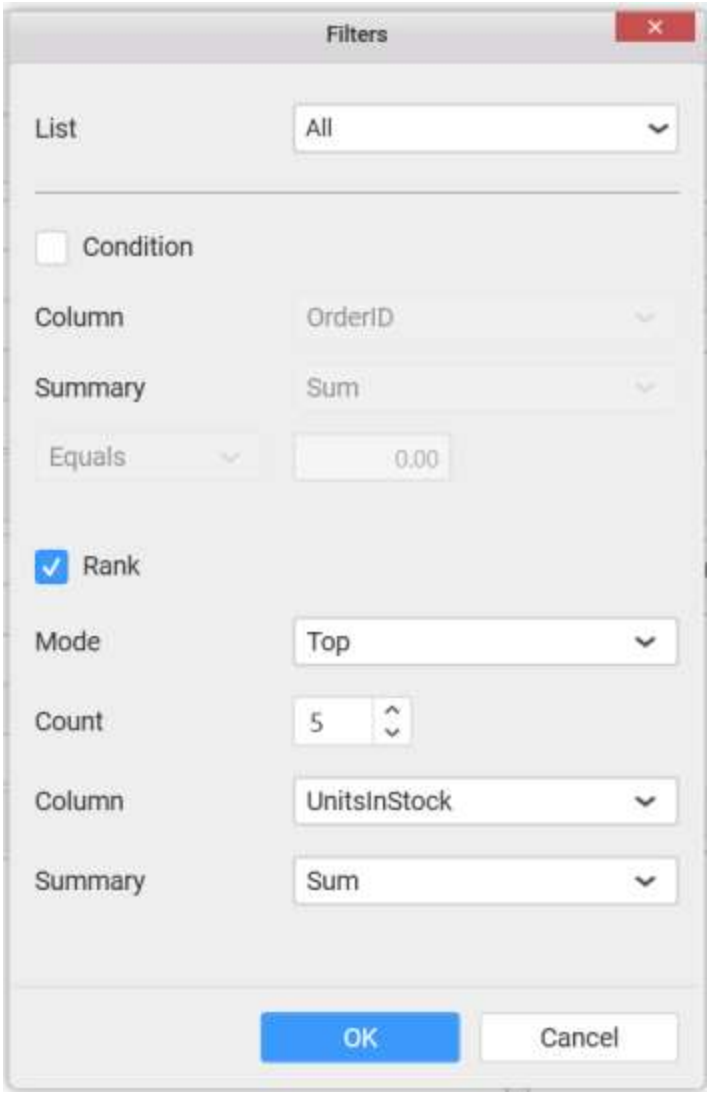
You can apply filters by selecting filter in settings.

Note: Filter will be set by default for top 5 records.



The filters option will open.

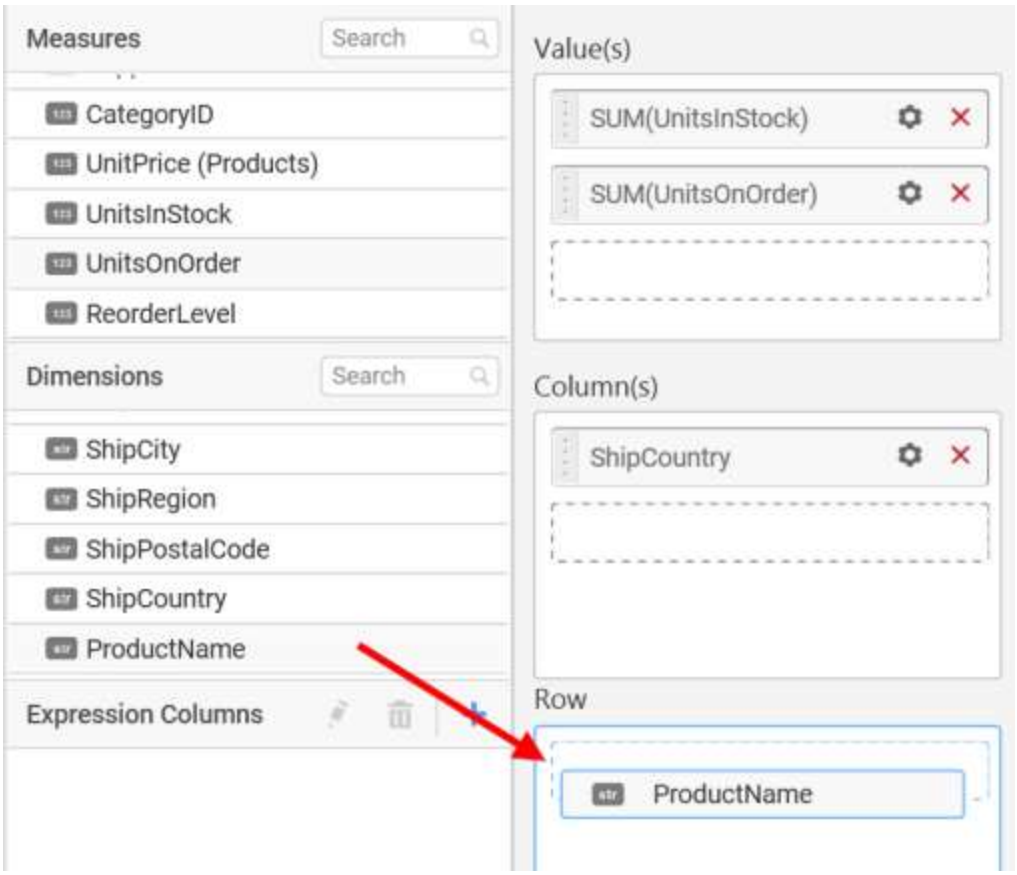
Select the needed Conditions and Rank and then click OK.



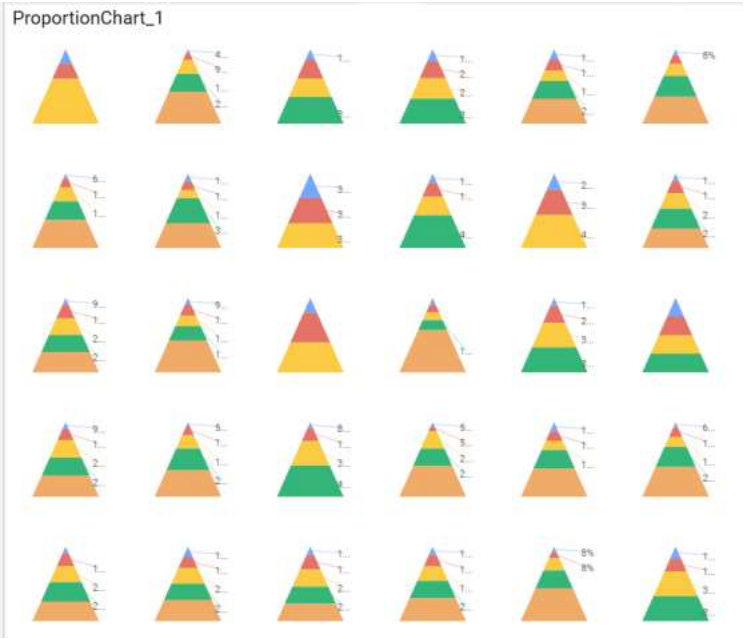
Similarly you can add the Measures and Expression Columns into Column field.

Adding Row

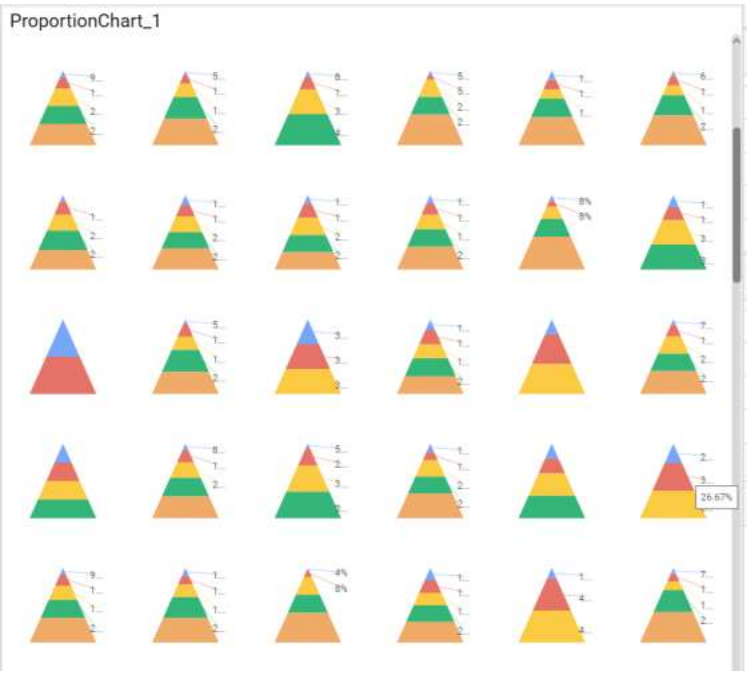
You can drag and drop the Measure or Dimension into the Row field.



This will render pyramid chart in series.



Scroll to see all charts

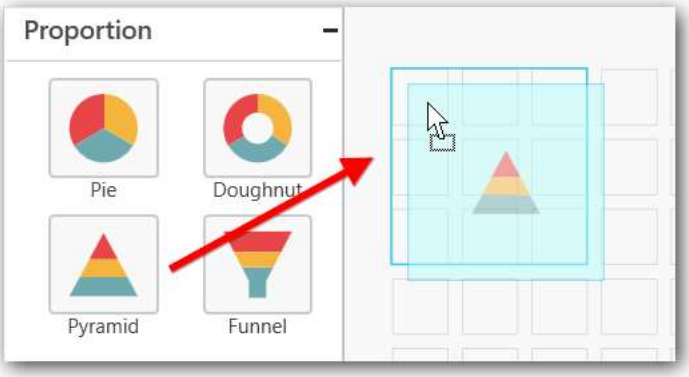


How to configure SSAS Data to Pyramid Chart?

Pyramid Chart need a minimum of 1 value element and 1 column element to showcase. The measure or expression field that you would like to analyze can be dropped into Value(s) block. The dimension that you would like to categorize the measure, can be dropped onto Column block. If you would like to categorize based on a series, then the respective dimension can be dropped onto Row block in addition.

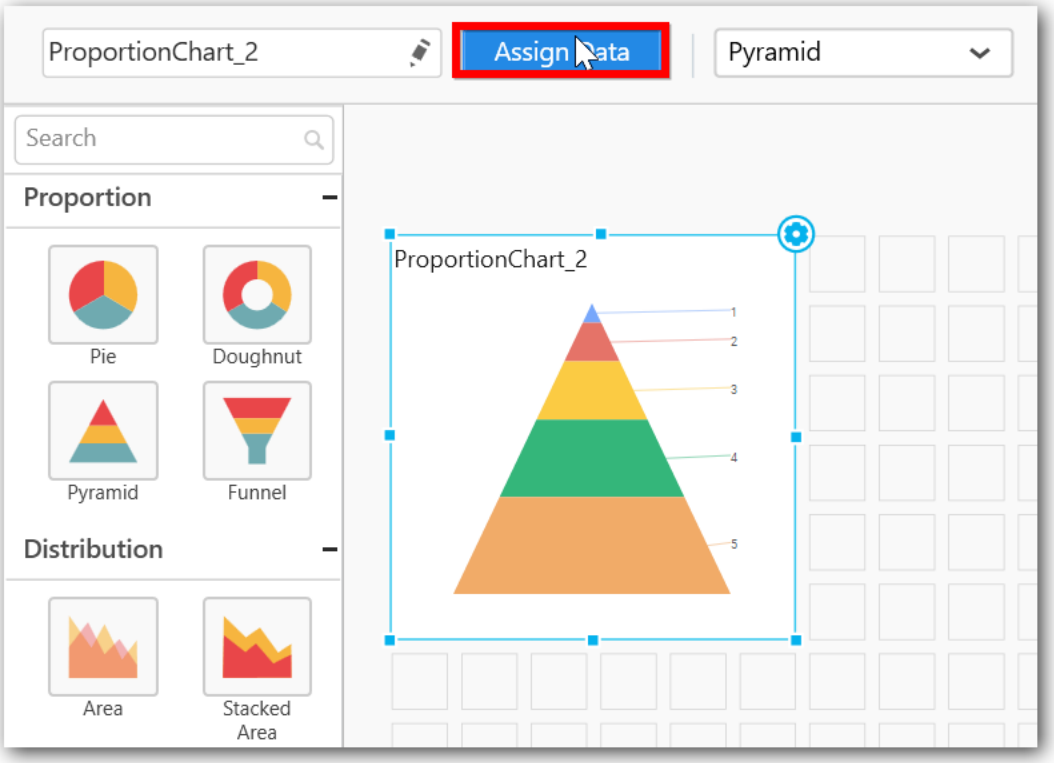
Following steps illustrates configuration of SSAS data to Pyramid Chart.

Drag and drop the Pyramid chart into canvas and resize it to your required size.

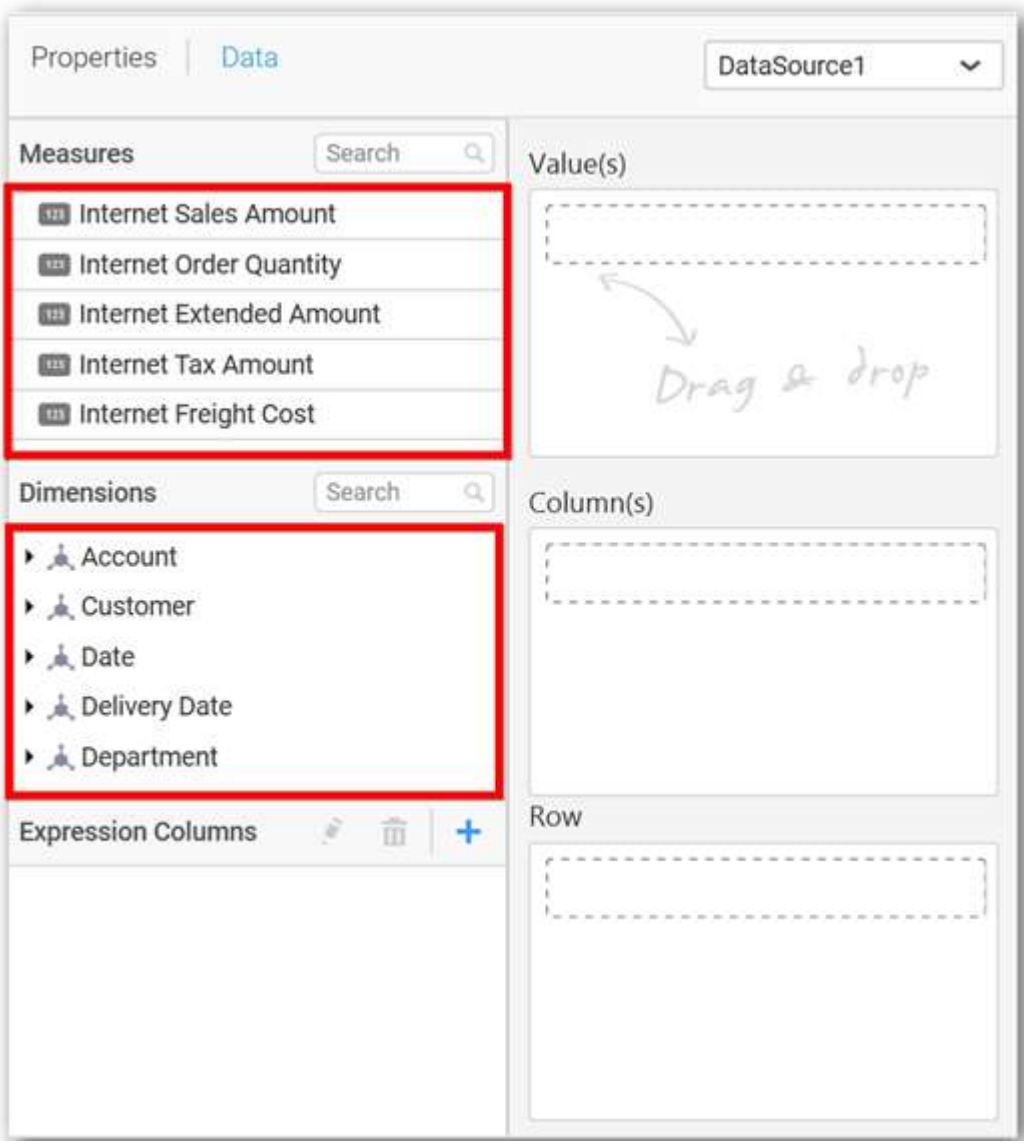


Select the dropped widget using mouse.

Click the Assign Data button in the toolbar.

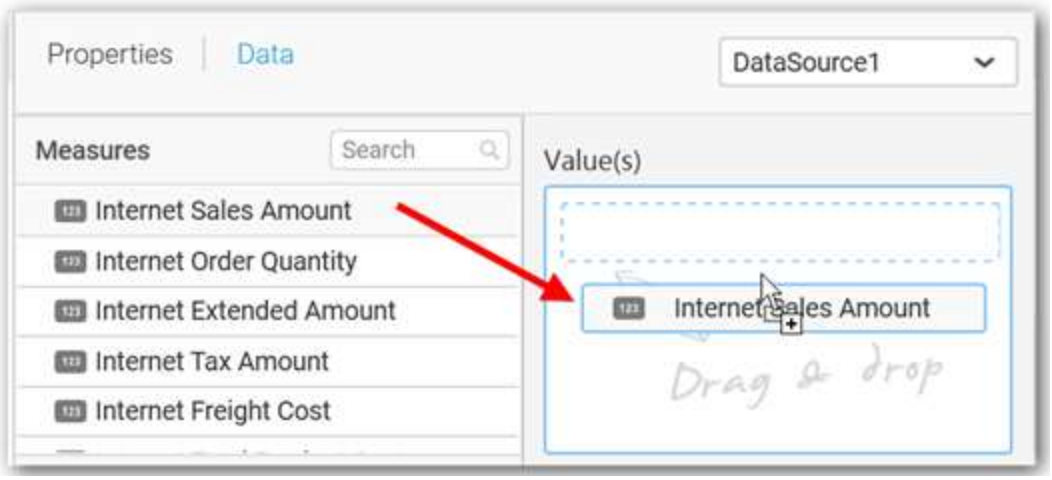


A Data pane will be opened with available Measures and Dimensions.

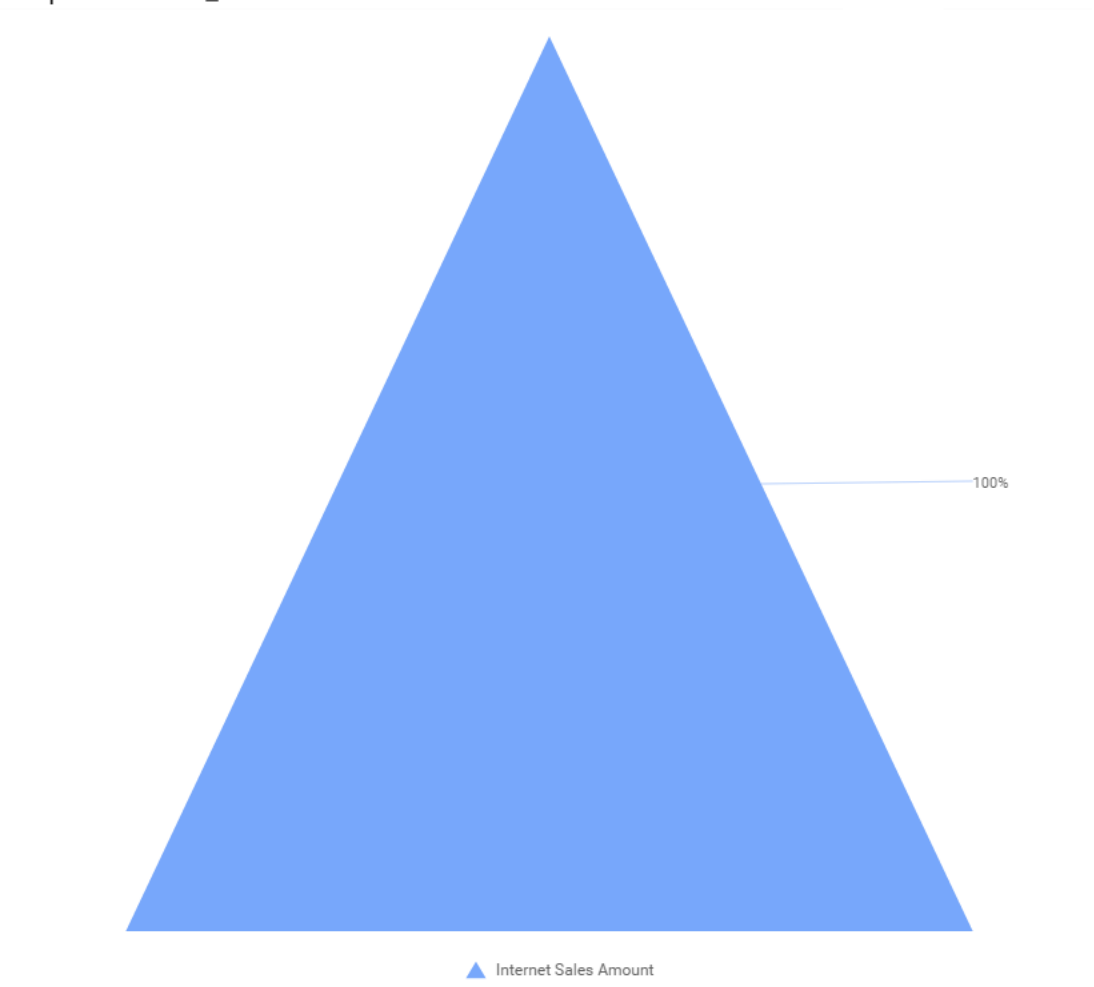


Assigning Value(s)

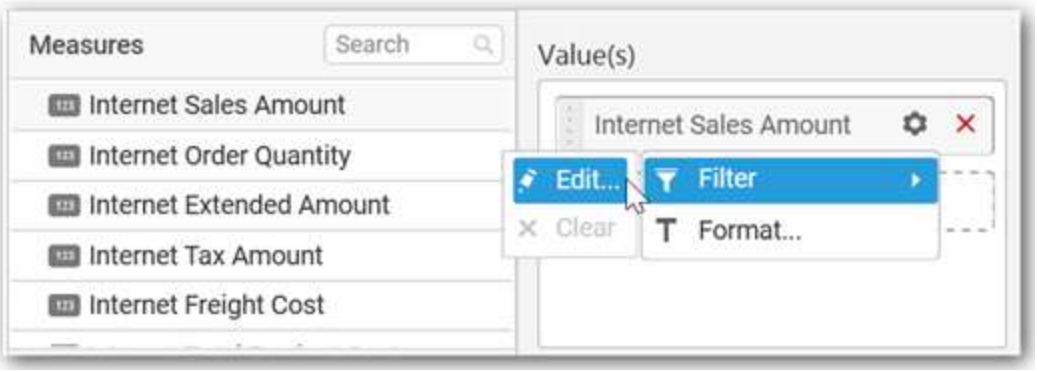
Drag and drop a column under Measures category into Value(s).



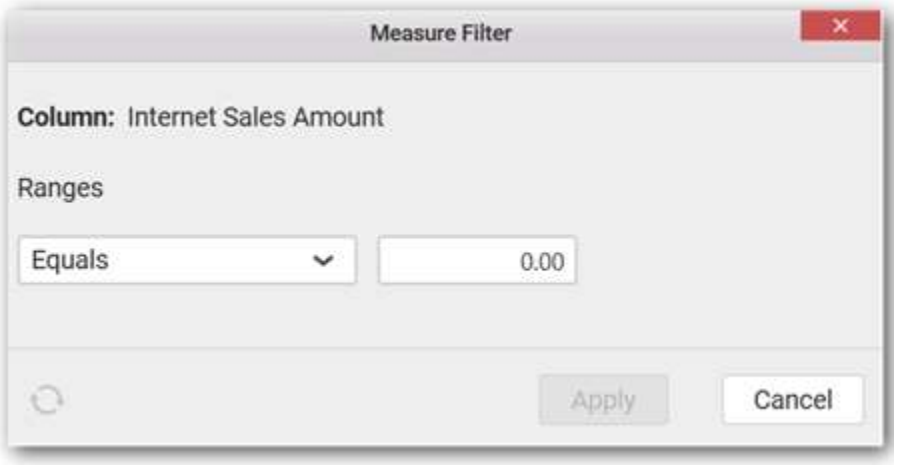
Now the chart will be rendered like this.



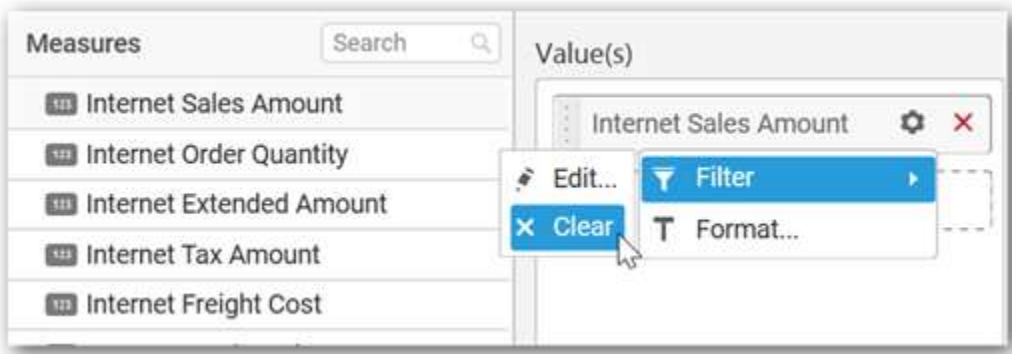
Define the filter criteria to match through choosing Edit option in Filter menu item.



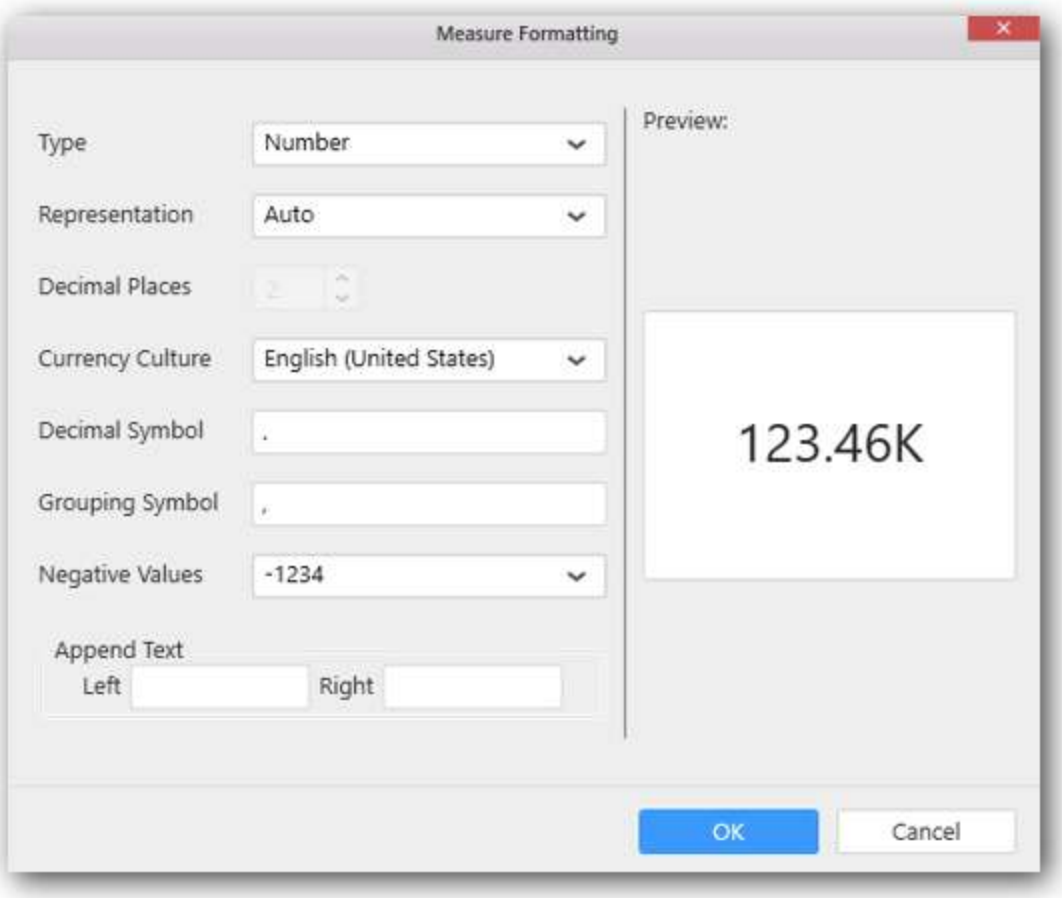
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



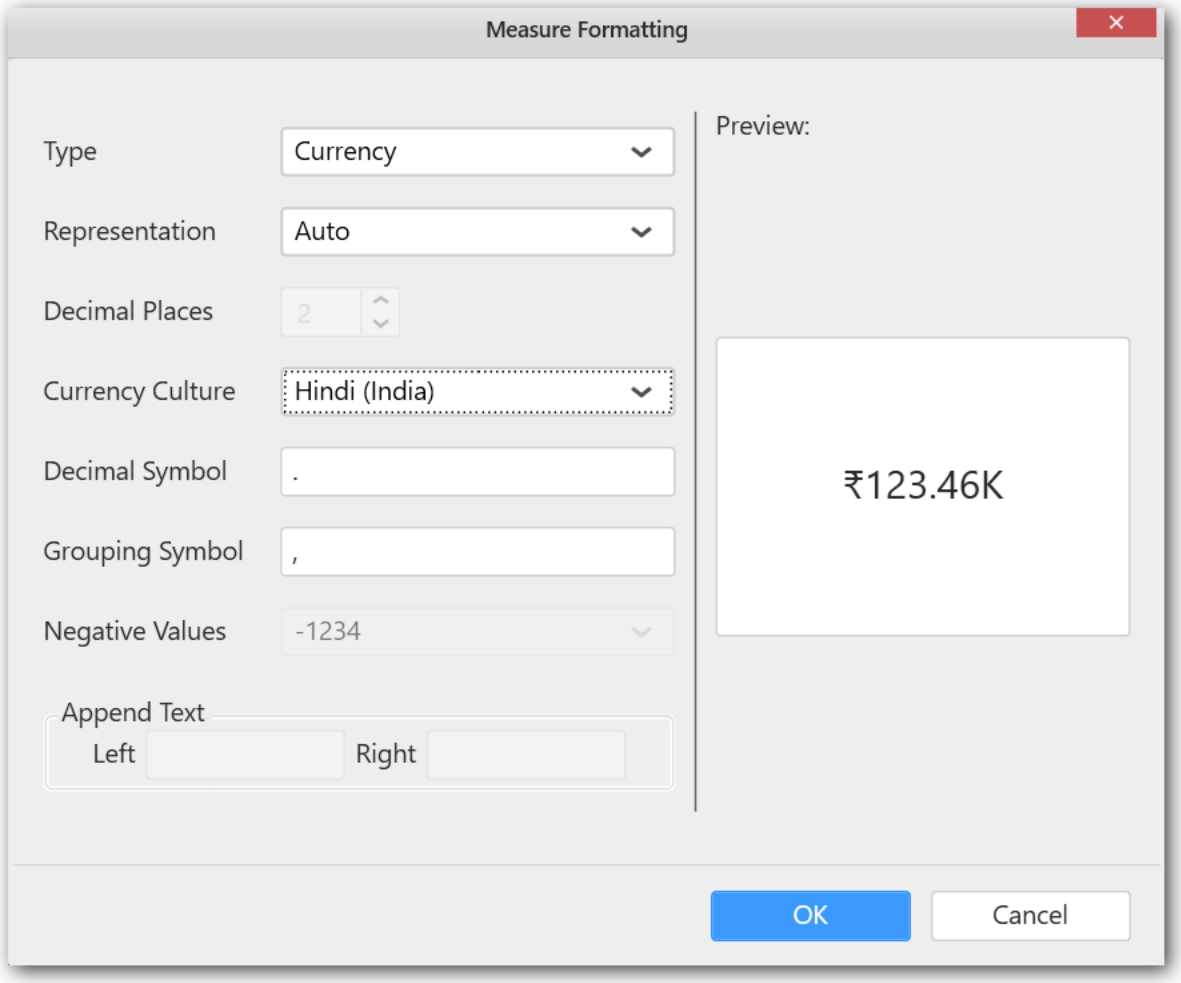
Select Clear option to clear the defined filter.



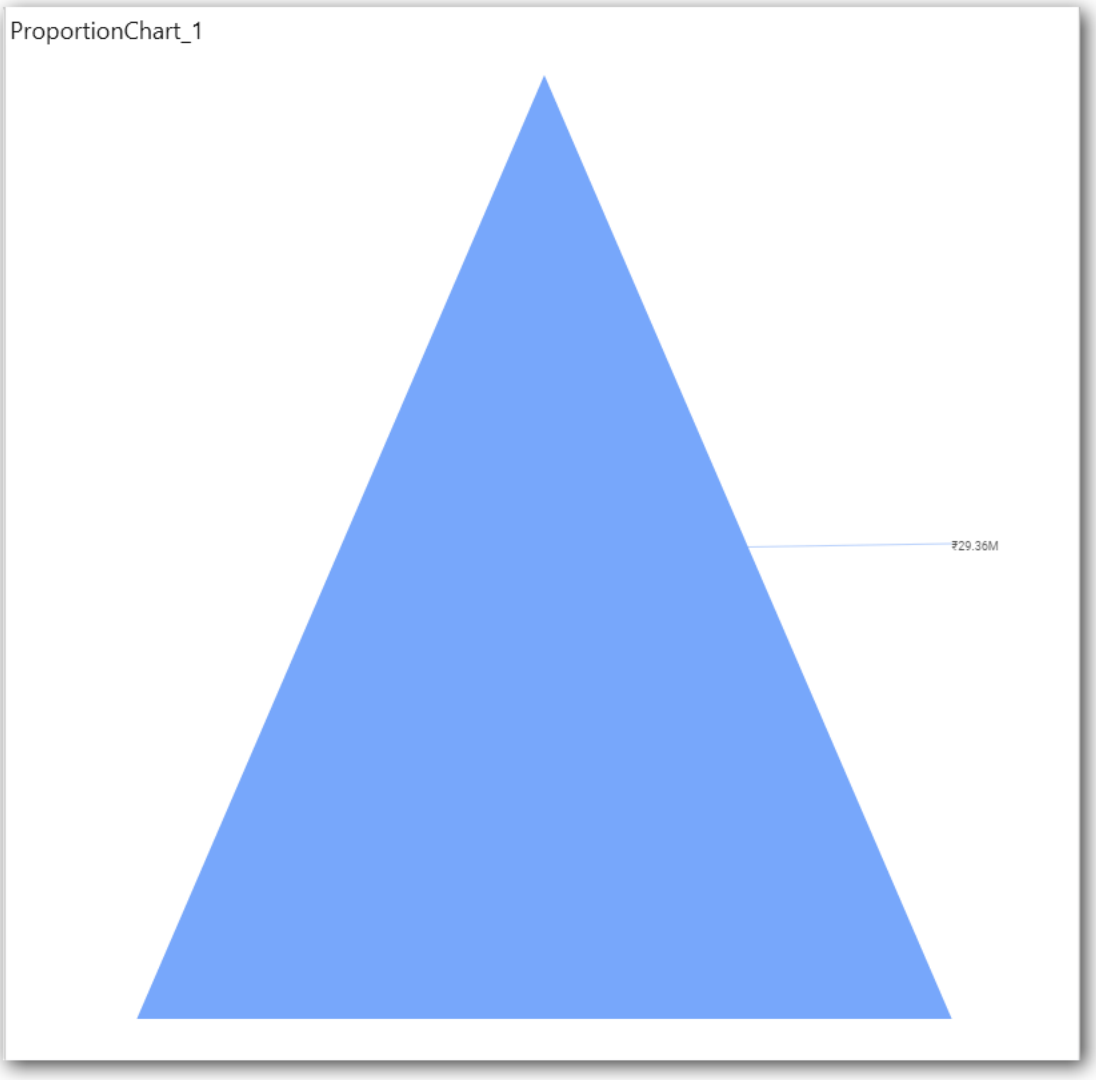
Select Format option to define the display format to the values in the column through Measure Formatting window.



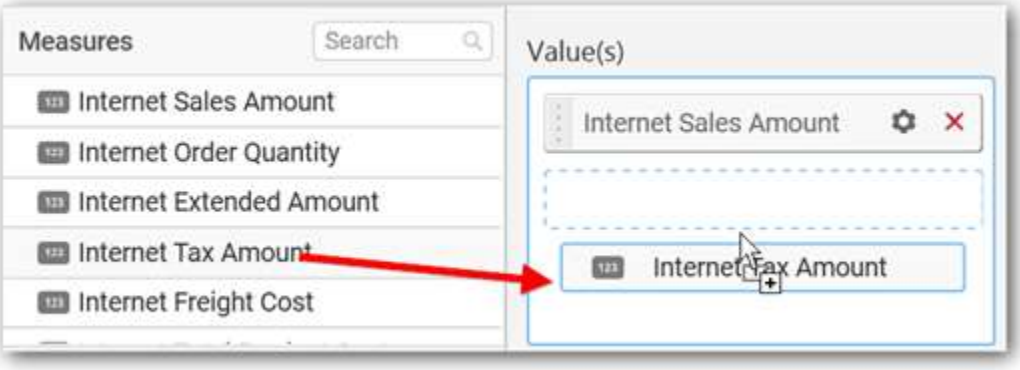
Choose the options you need and click OK.

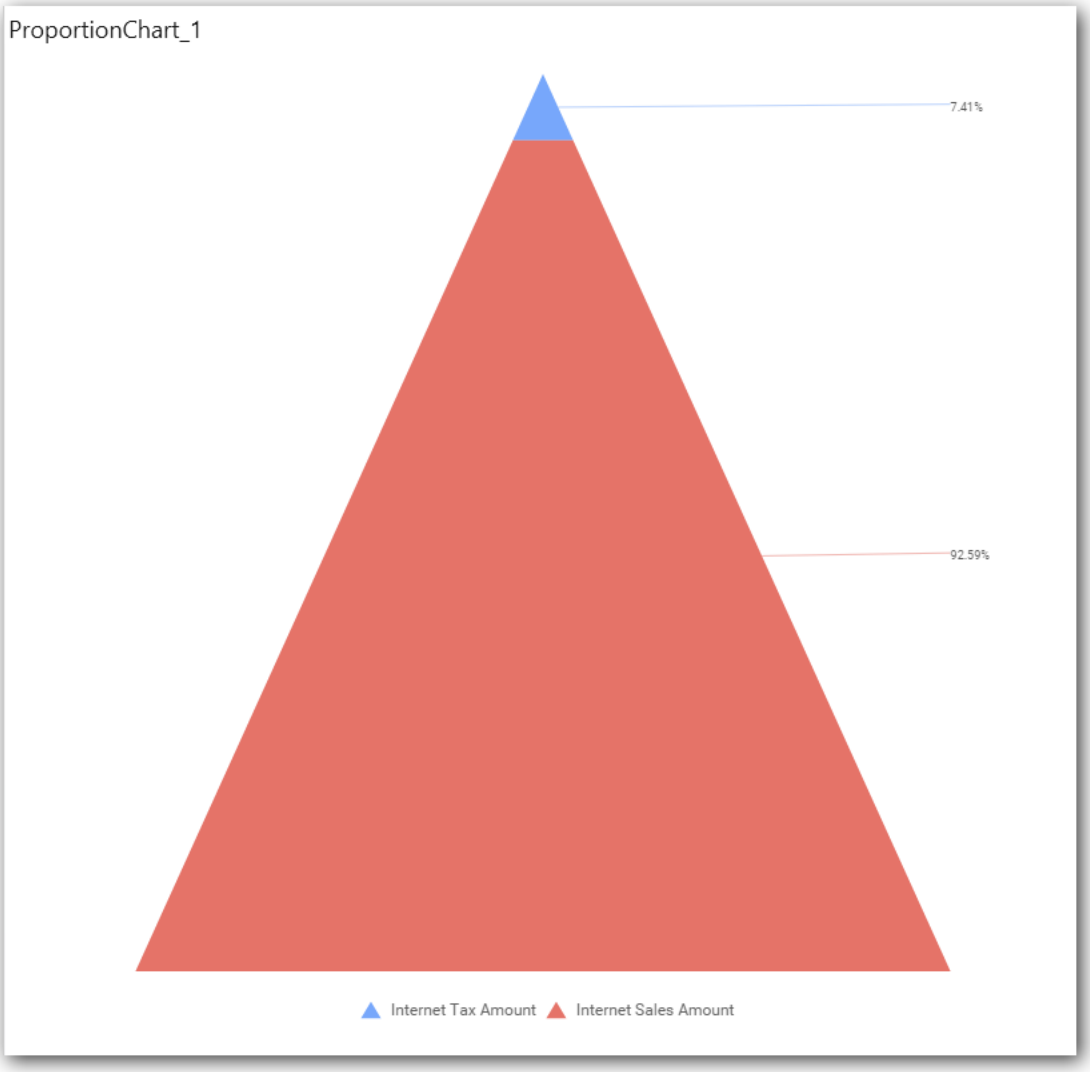


Now the Chart will be rendered like this.



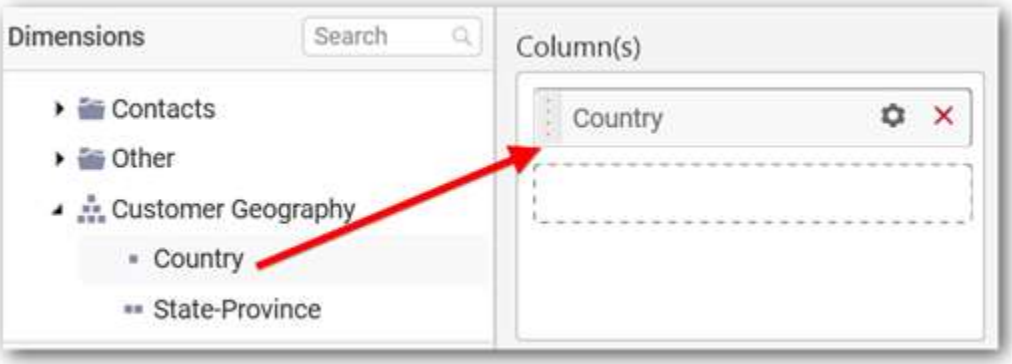
You can also add more than one column to the Value(s).

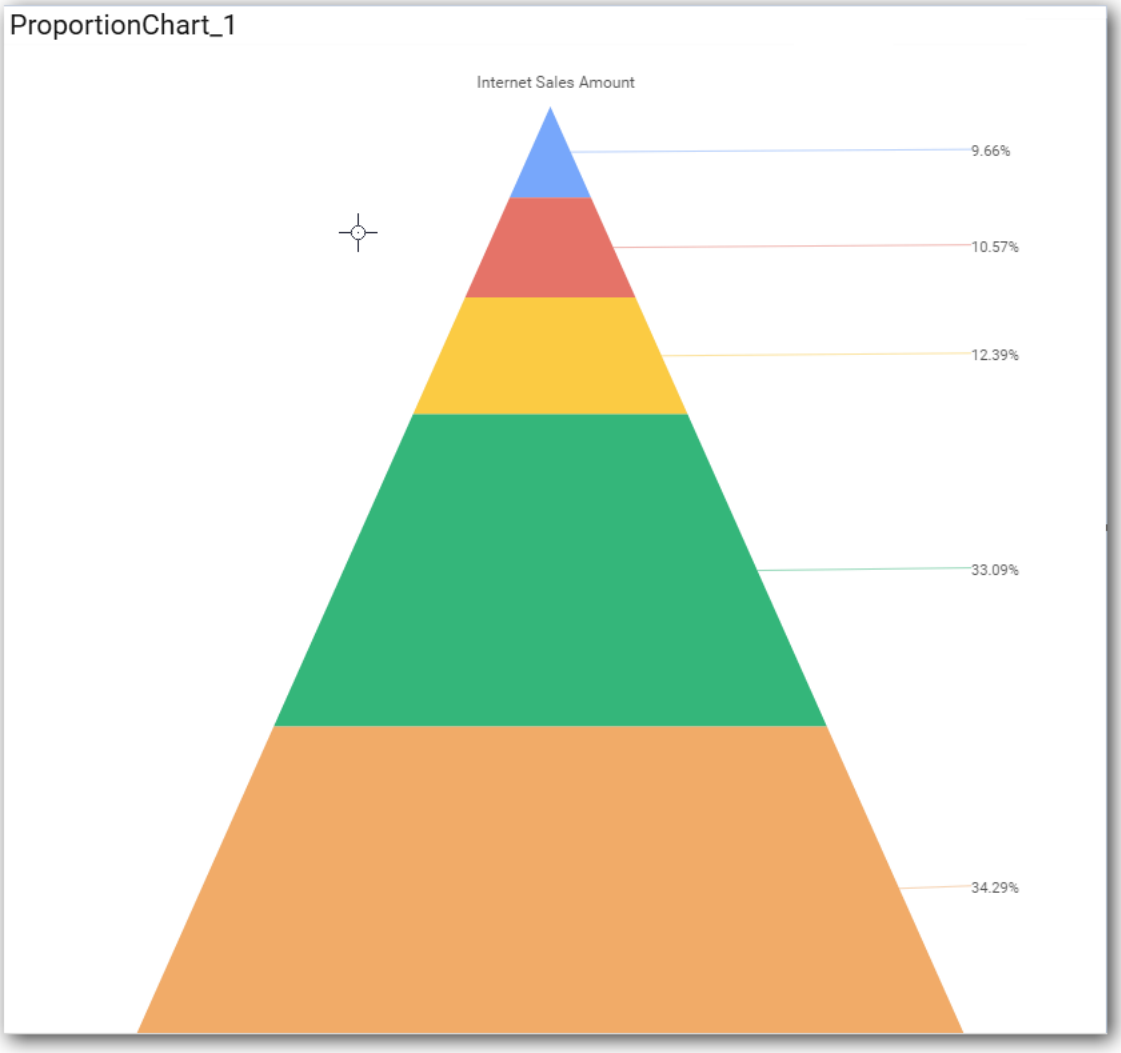




Assigning Column(s)

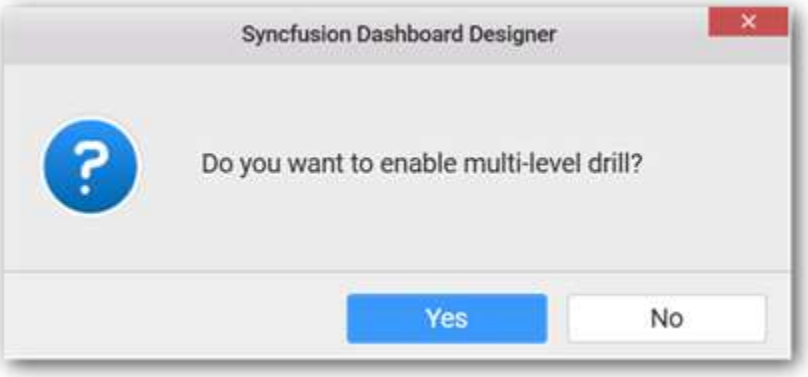
Add a dimension level or hierarchy into Column(s) section through drag and drop.





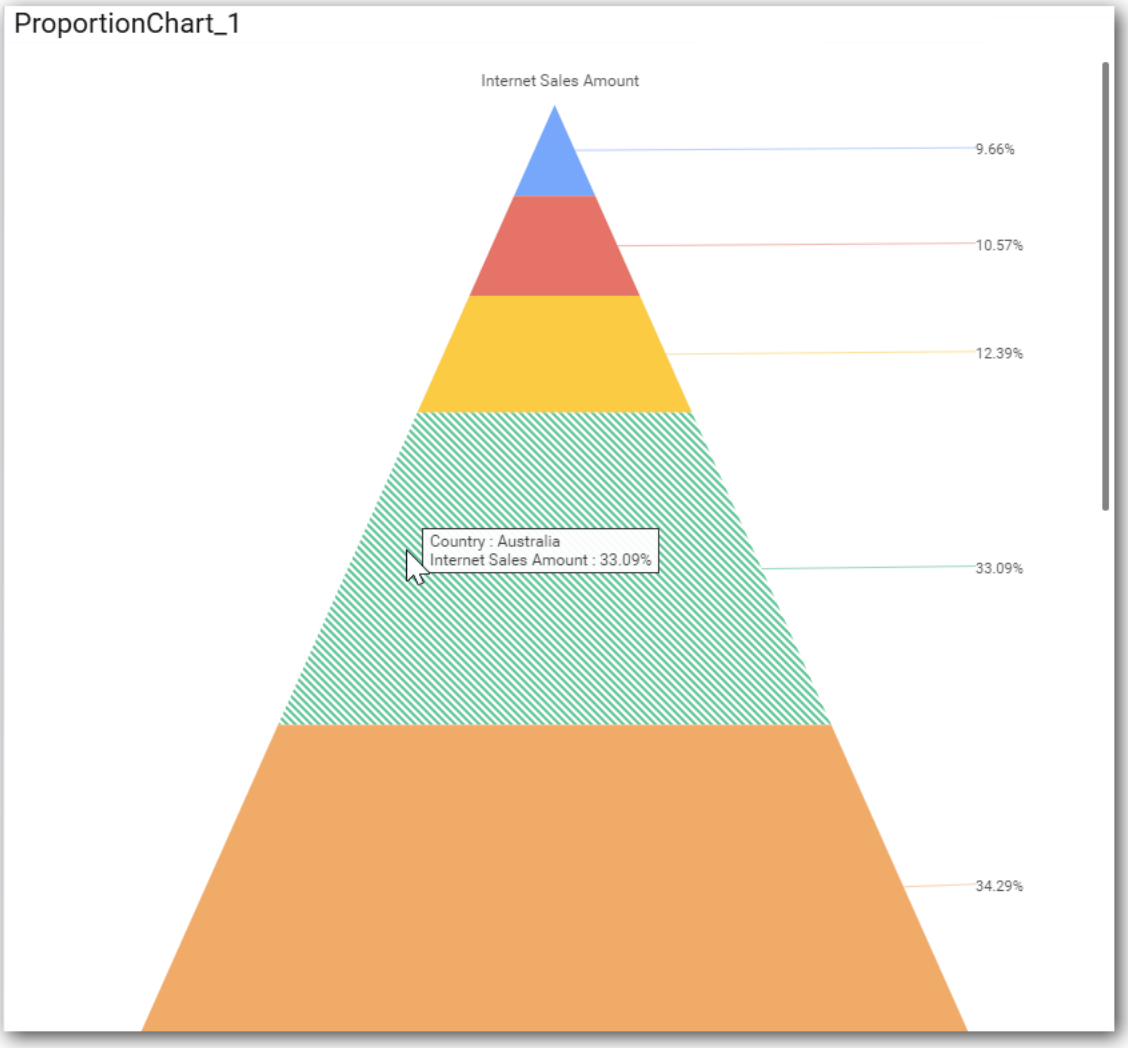
You may also add more than one column into Column(s) section.

In that case, you will be prompted with a message like below, asking for confirmation to enable drilling across the levels.

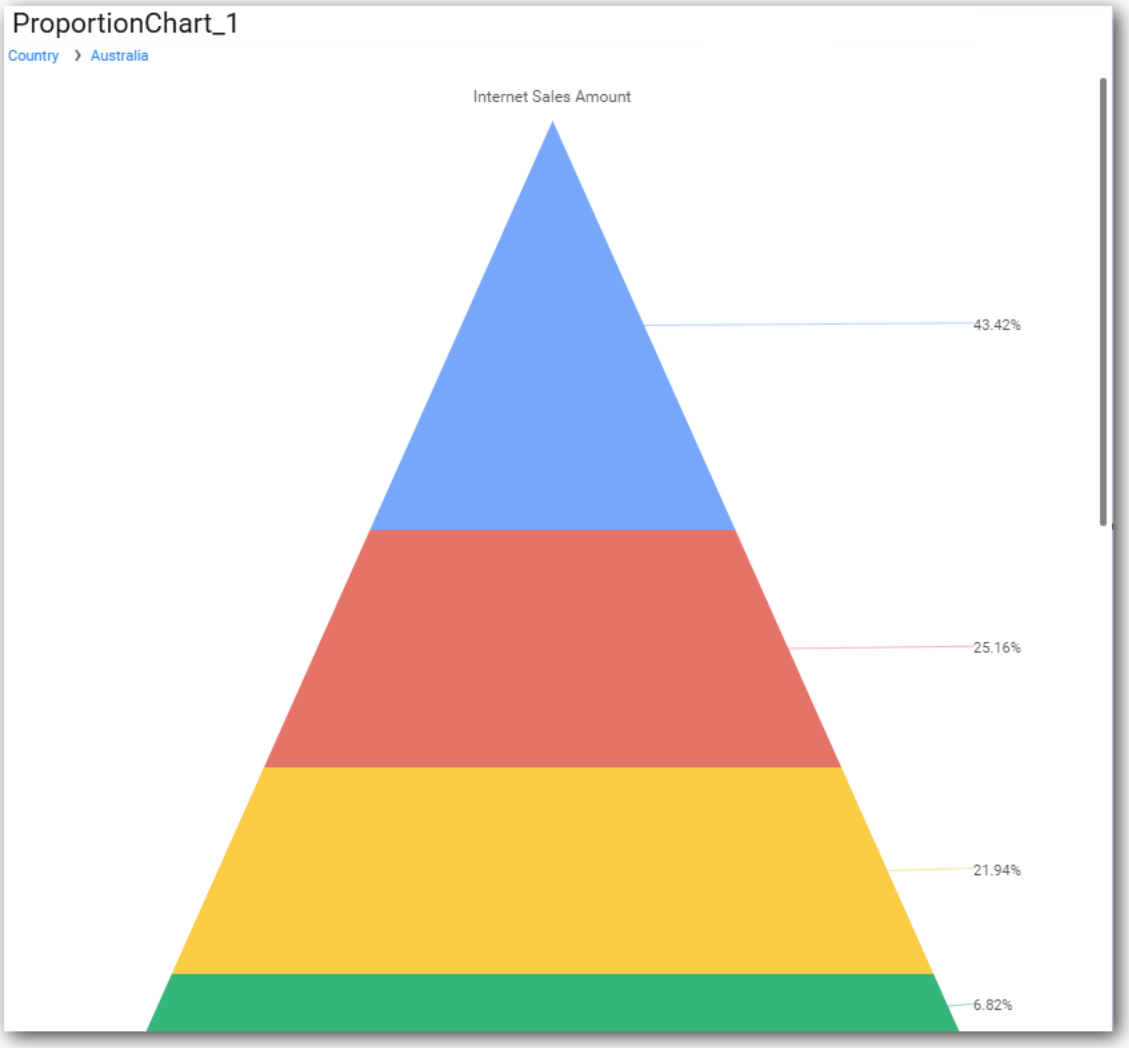


Select Yes to enable drill option in chart. Select No to replace the existing column with this one in the Column(s) section.

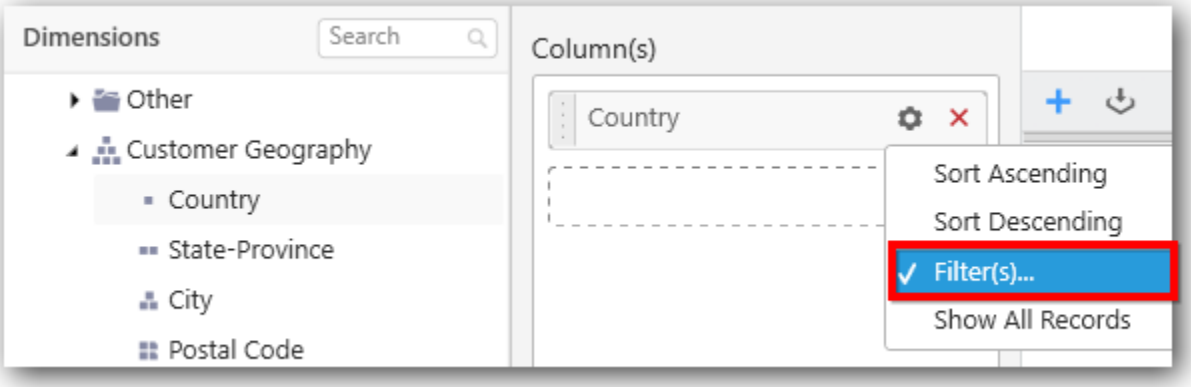
Click the respective data value marker in chart to drill into its inner level.



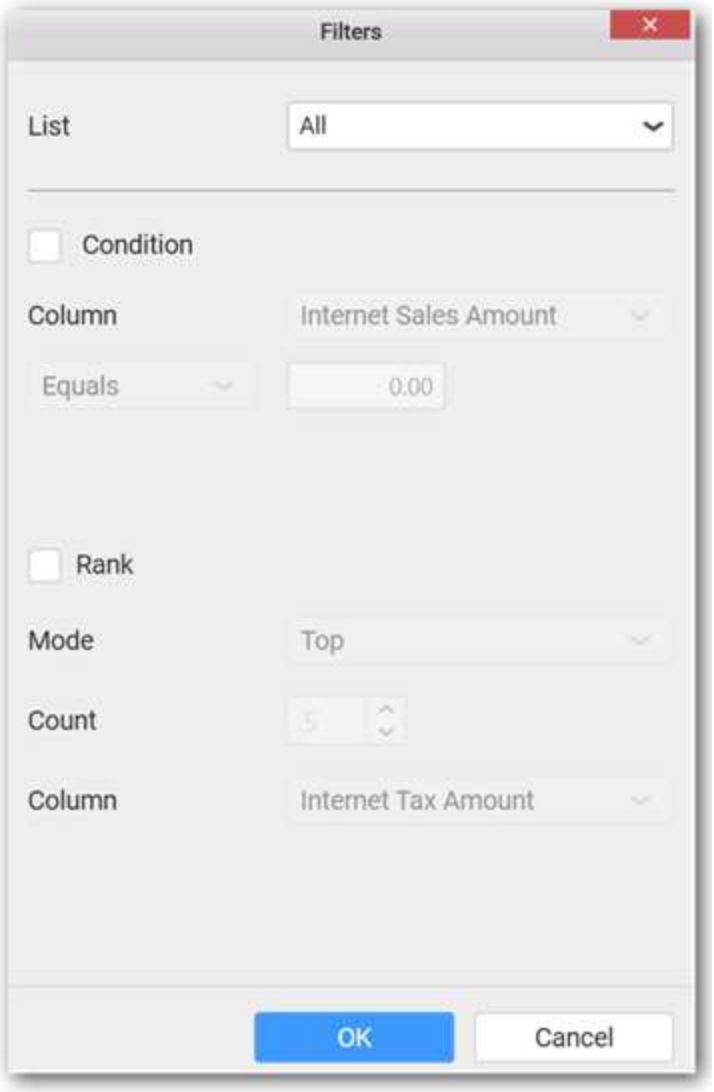
The drilled view of the chart is follows.



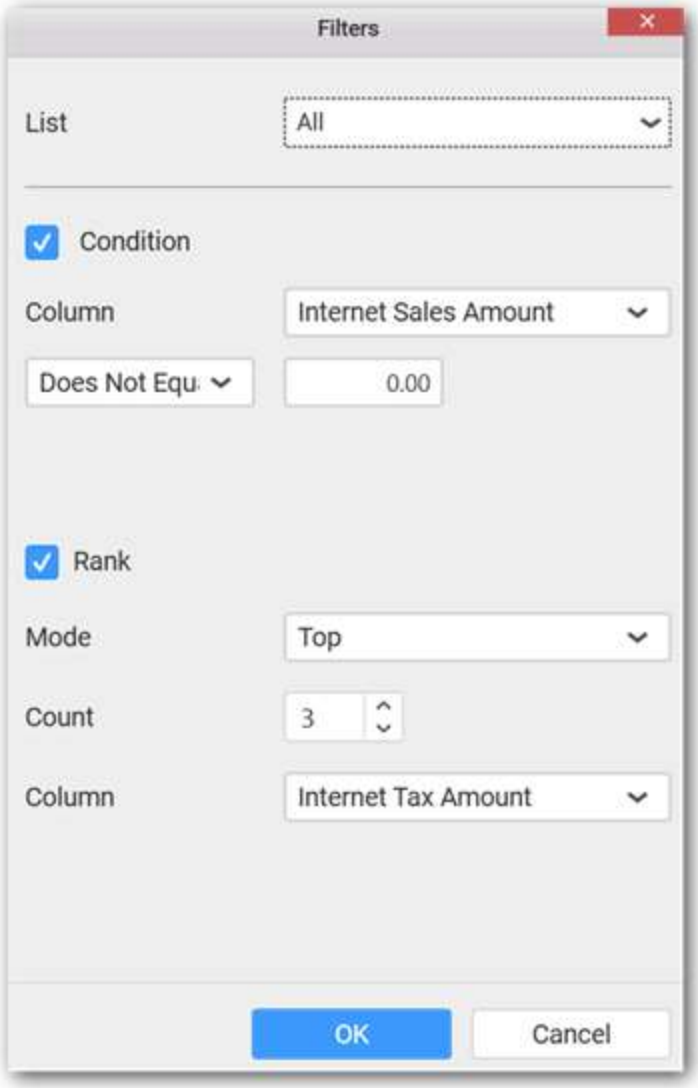
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



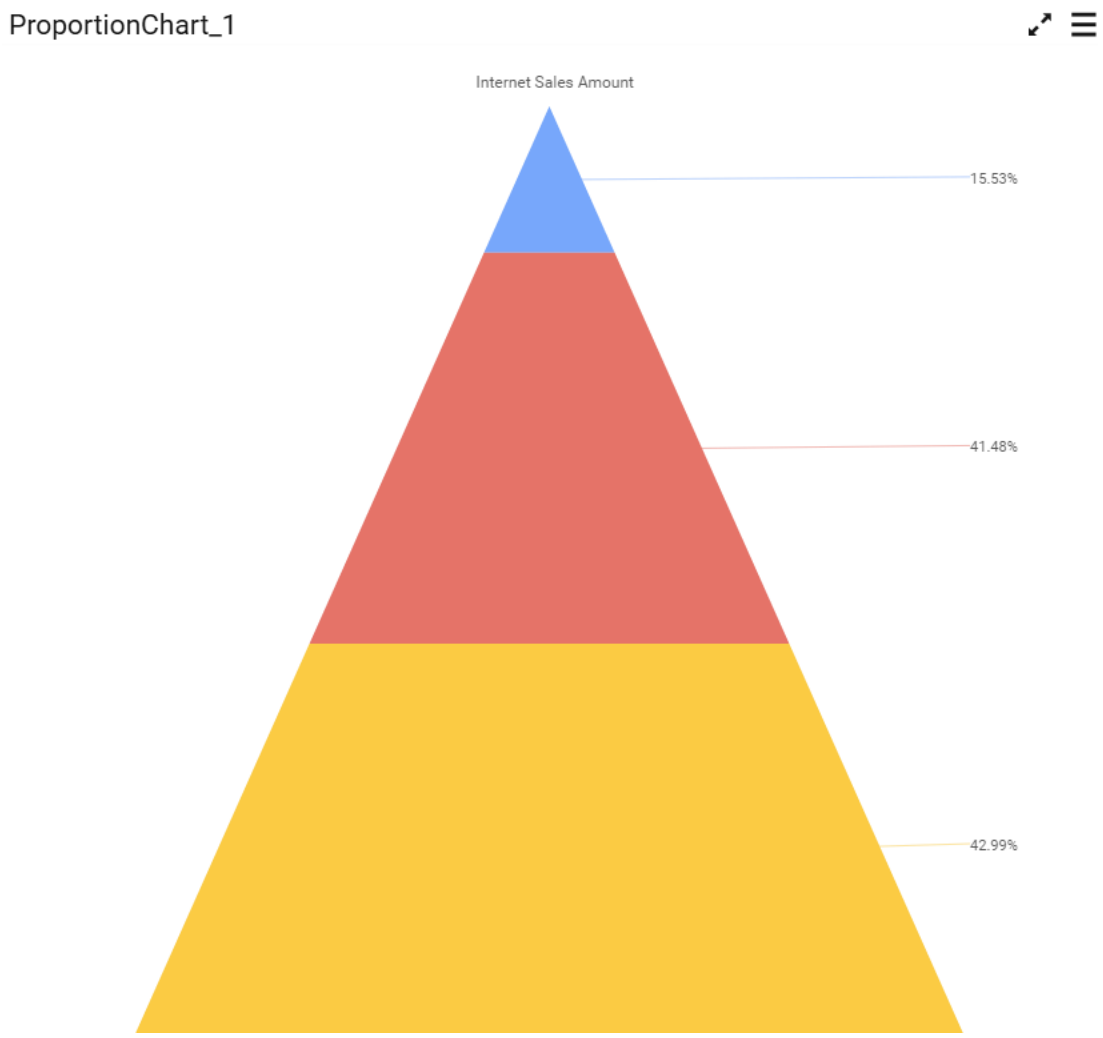
Select Filter(s)... to launch the Filters window.



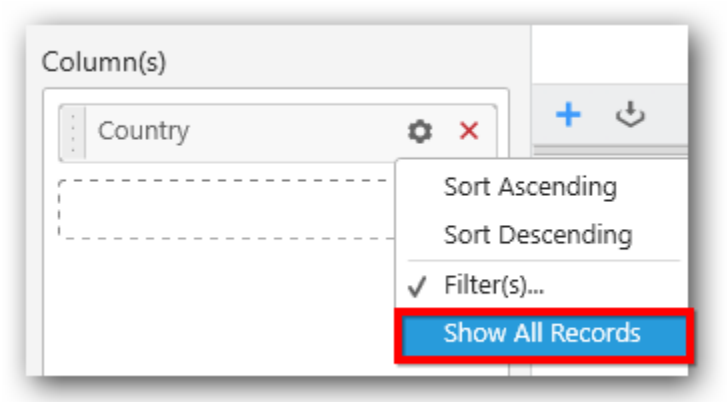
Define the filter Condition and Rank and Click OK.



Now the chart will be rendered like this

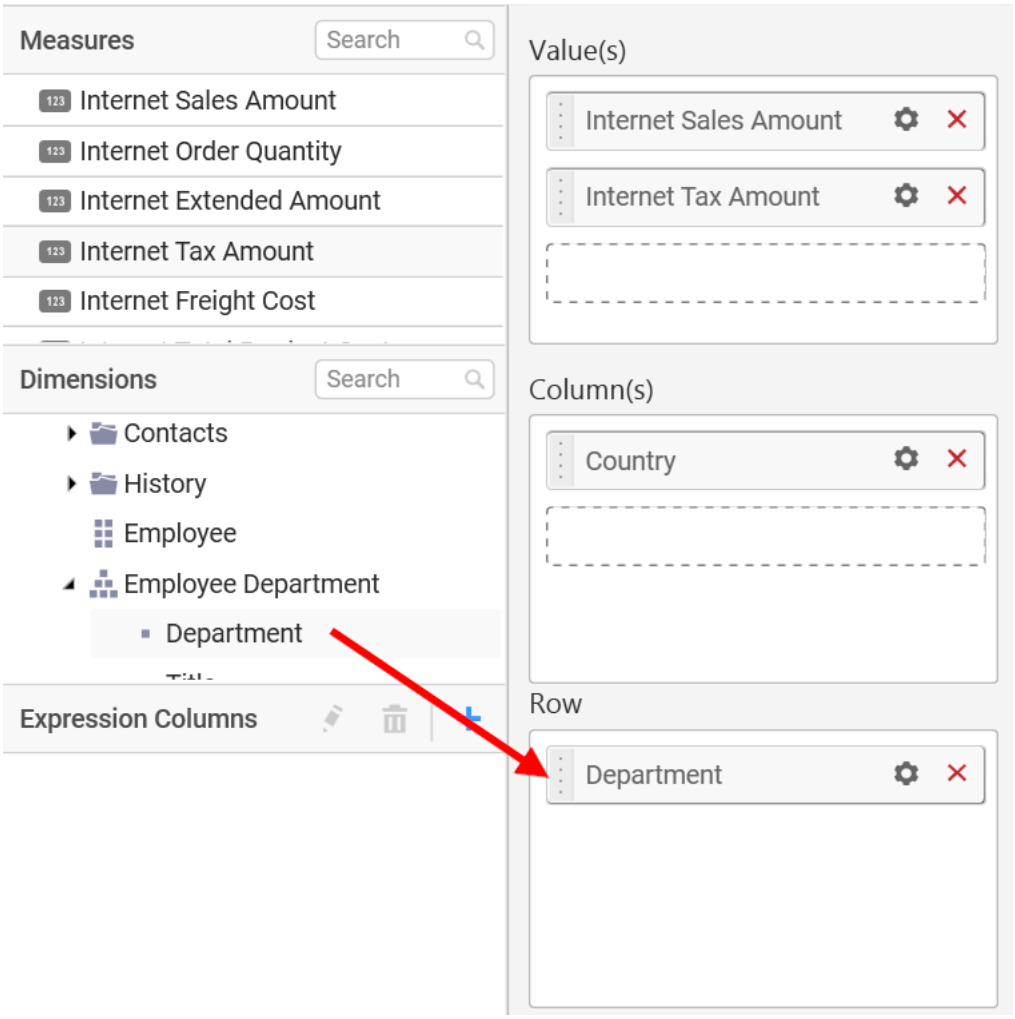


To show all records again click on Show All Records.

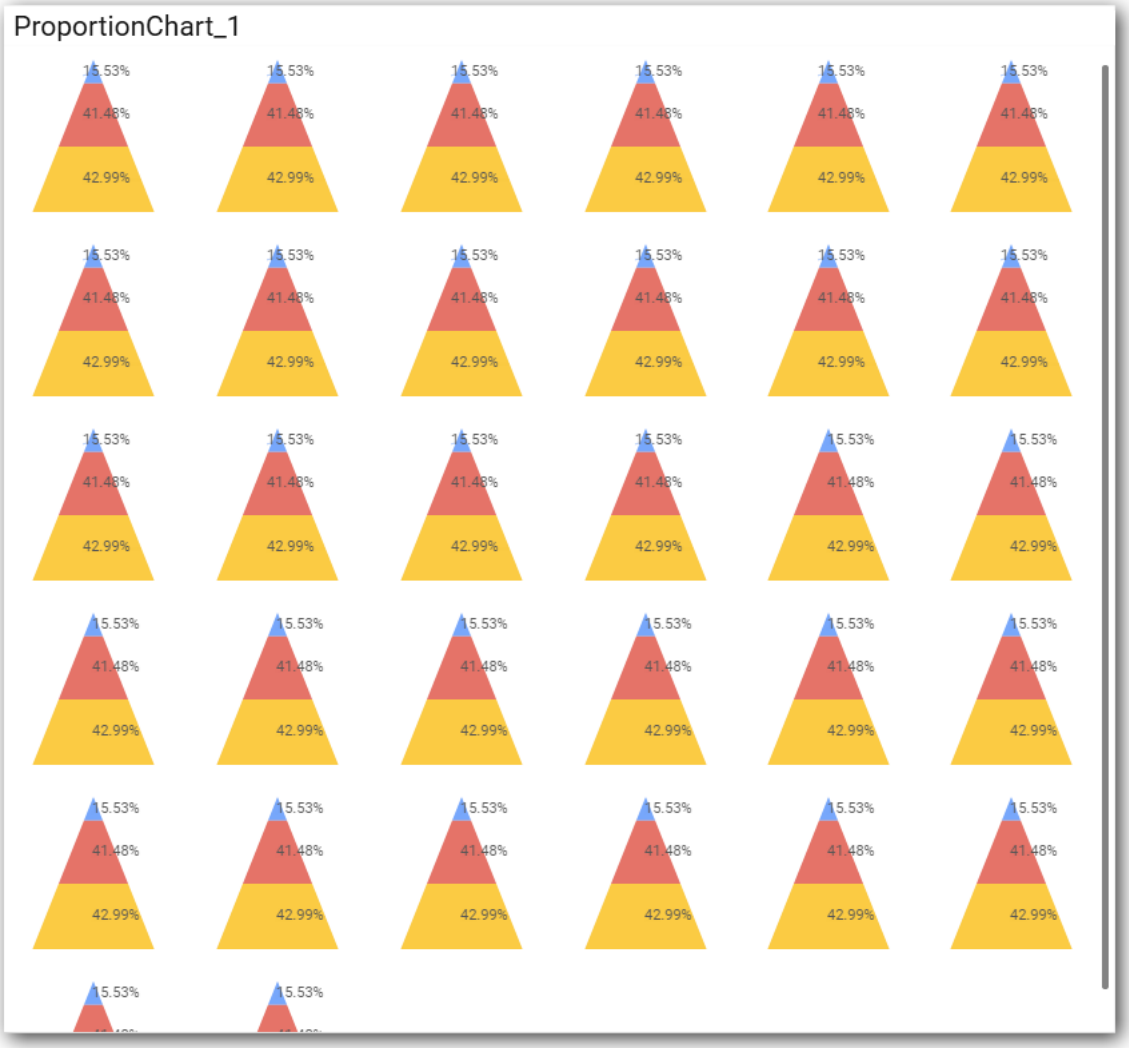


Assigning Row

You can add a dimension level or hierarchy to Row section for series rendering of chart.



The chart will be rendered in series as shown in the image.



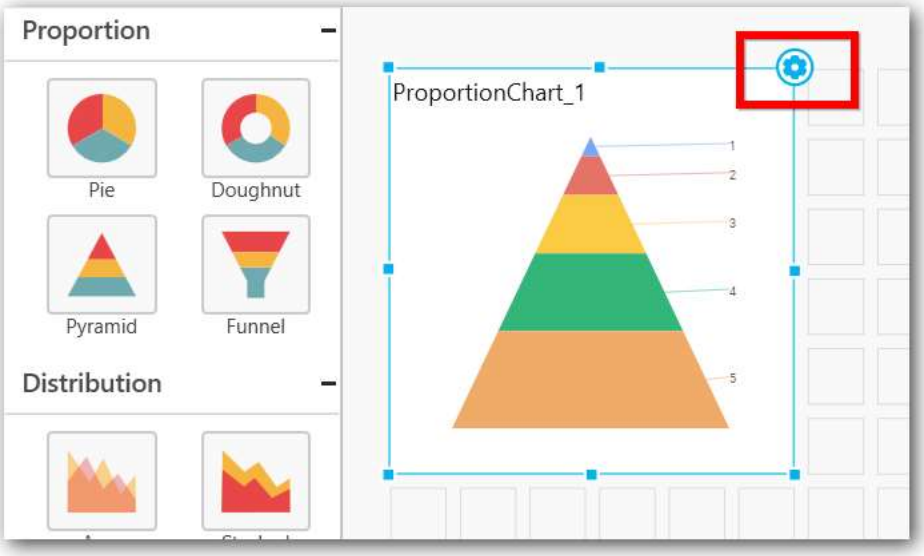
Scroll down to see all charts.

How to format Pyramid Chart?

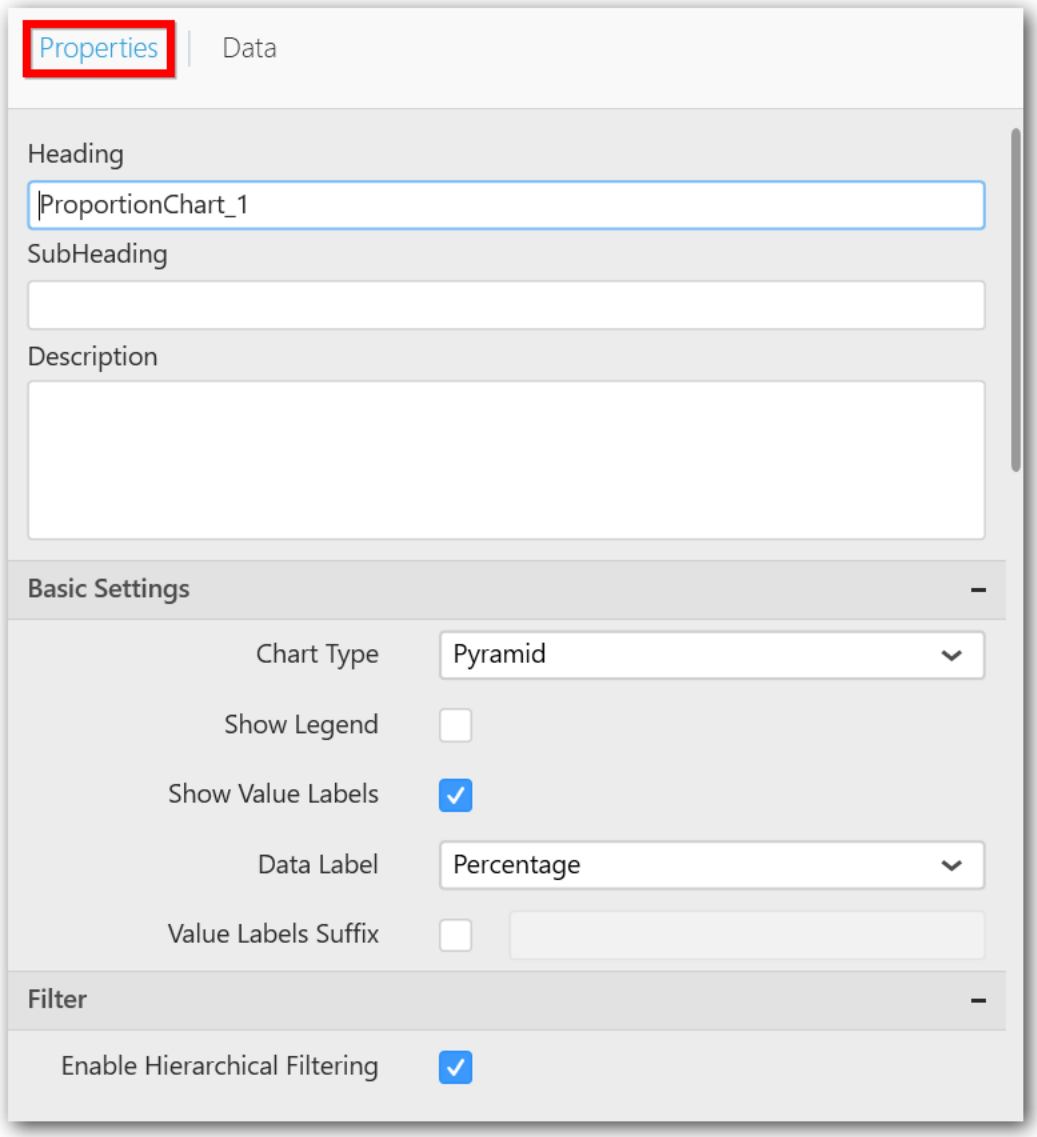
You can format the pyramid chart for better illustration of the view that you require, through the settings available in Properties pane.

To configure data into Pyramid chart follow the steps

1. Drag and drop the Pyramid chart into canvas and resize it to your required size.
2. Configure the data into Pyramid chart.
3. Focus on the Pyramid chart and Click on Widget Settings.

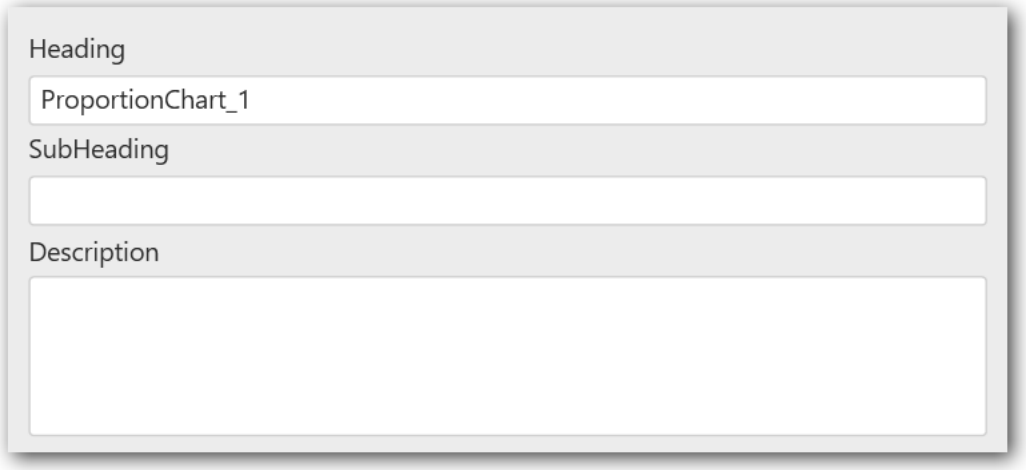


The property window will be opened.



You can see the list of properties available for the widget with default value.

General Settings



Header

This allows you to set title for this pyramid chart widget.

SubHeading

This allows you to set sub-title for this pyramid chart widget.

Description

This allows you to set description for this pyramid chart widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

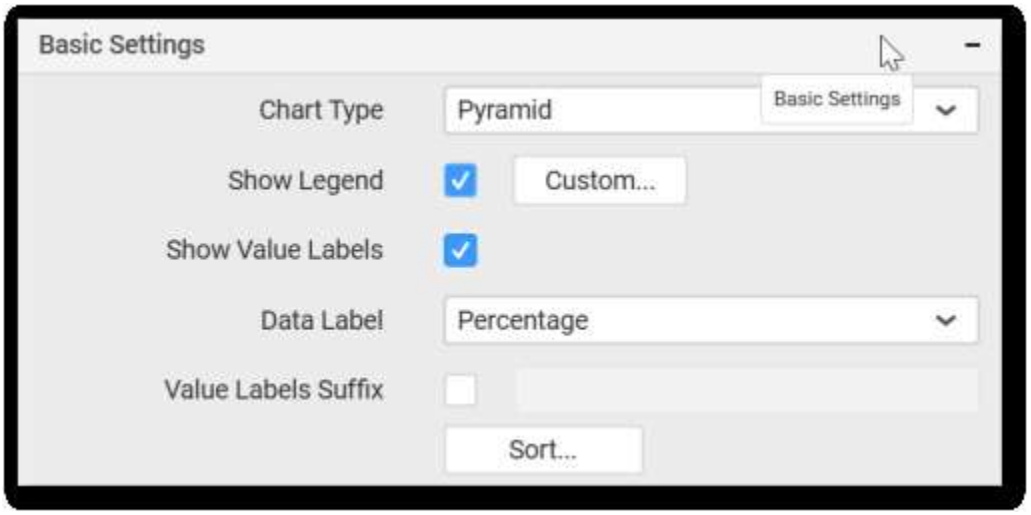


Chart Type

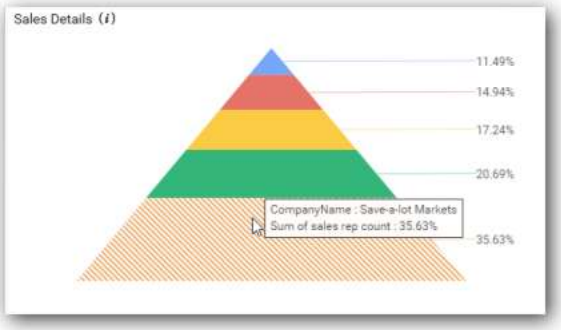
This allows you to switch the widget view from current chart type to another chart type.

Enable Drill Down

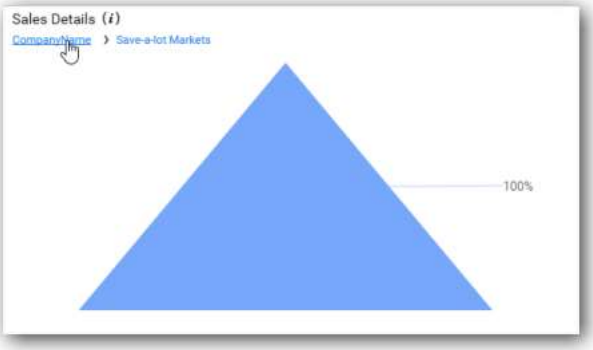
This allows you to add more than one dimension element to the Column block in Data Pane of Widget View such that, those form an hierarchy and each of its level can be navigated through clicking the

respective series drawn. In its disabled state, trying to add more than one element will replace the existing one.

Initial View

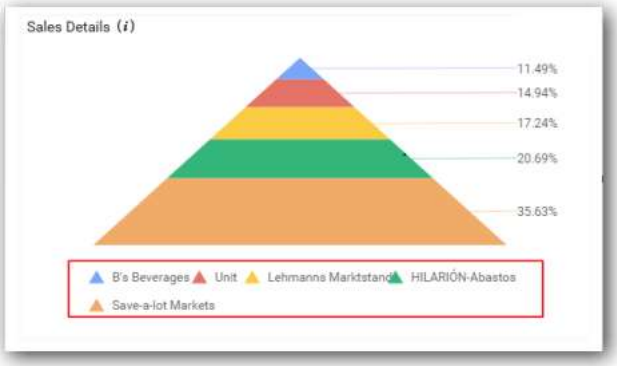


Drilled View



Show Legend

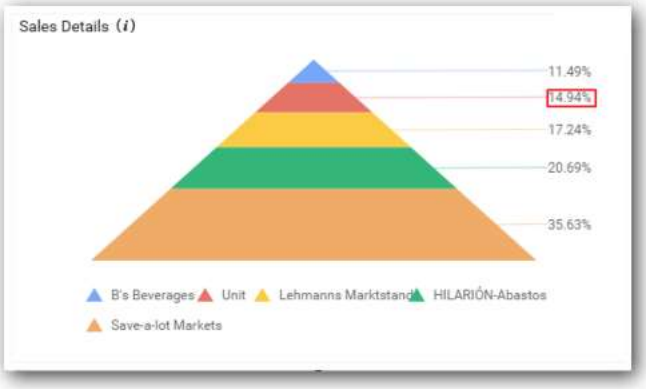
This allows you to toggle the visibility of legend in chart and also change the legend text position (selecting through combo box).



Enabling this option of **Custom Legend Text** will allow you to define a custom text (through the text area) to display for each legend series (selecting through the combo box) in chart.

Show Value Labels

This allows you to toggle the visibility of value labels.



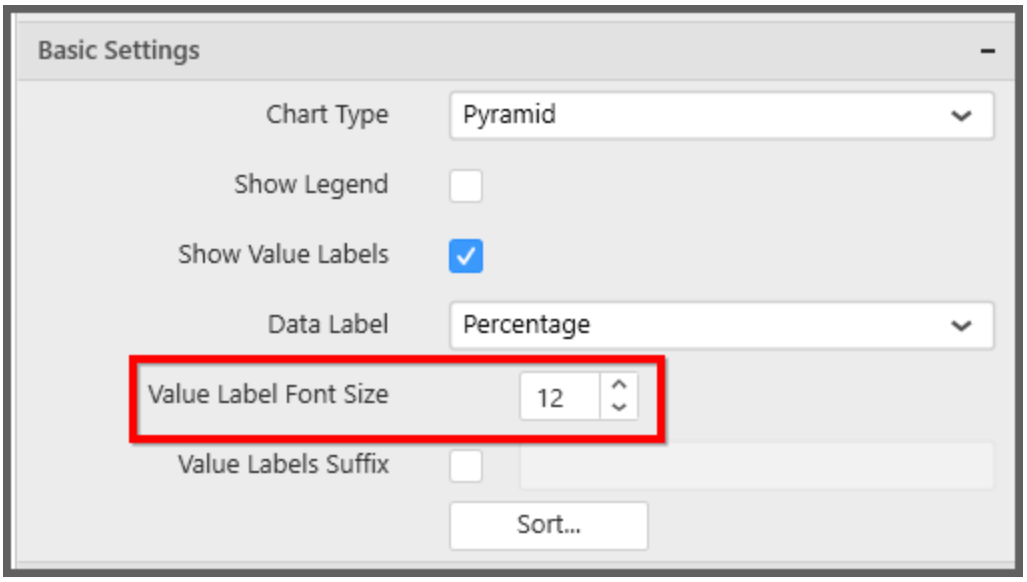
Data Label Value

This allows you to define the display format either as value or as percentage.

Percentage

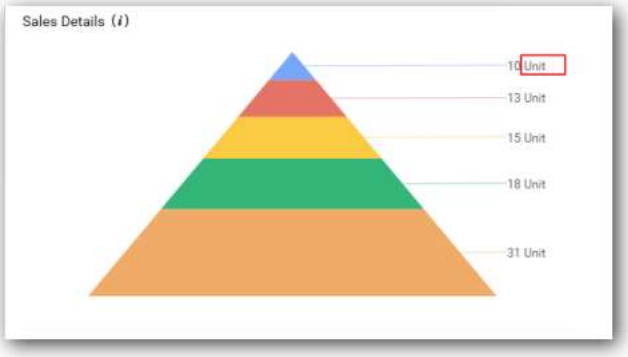
Value Label Font Size

This allows you to define the font size for the value labels to display. Text will get collapsed, if it exceeds the size of the bar. Default font size for the Value label is 9 pixels.



Value Labels Suffix

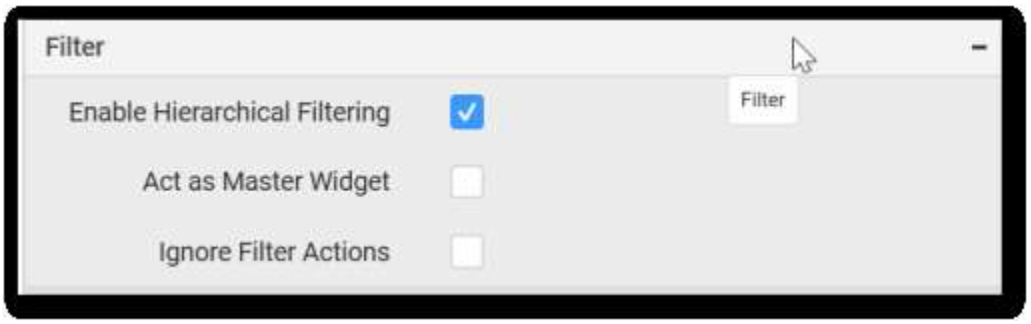
Allows you to set suffix to the value labels.



Sort Order

This allows you to define the sort order for each measure column added.

Filter Settings



Enable Hierarchical Filtering

This allows you to define the behavior of top n filtering which can be flat or hierarchical.

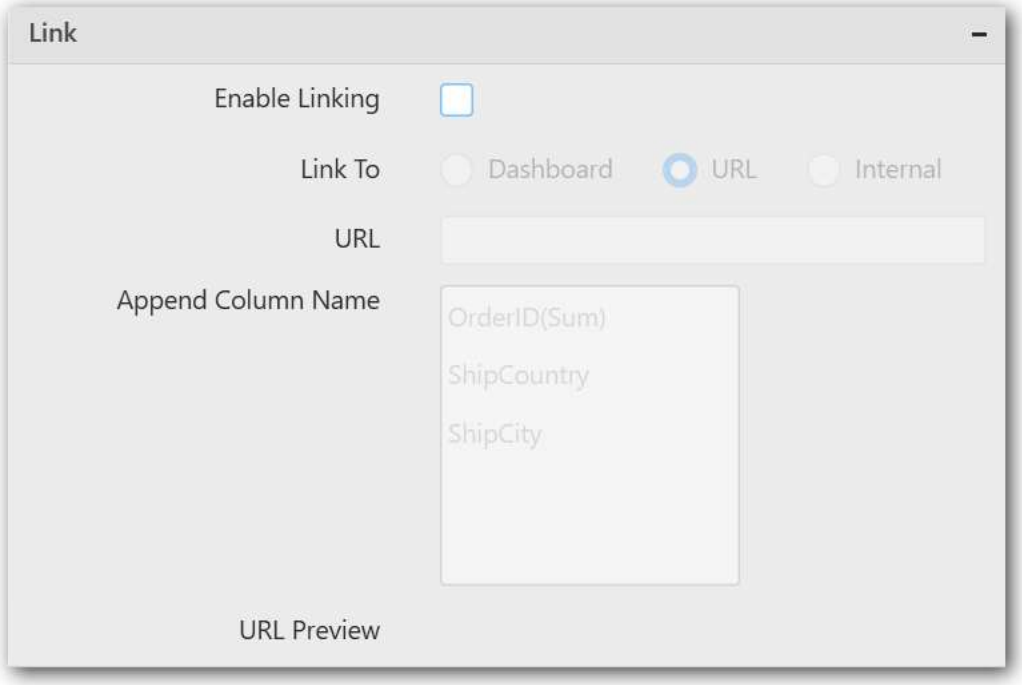
Act as Master Widget

This allows you to define this pyramid chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

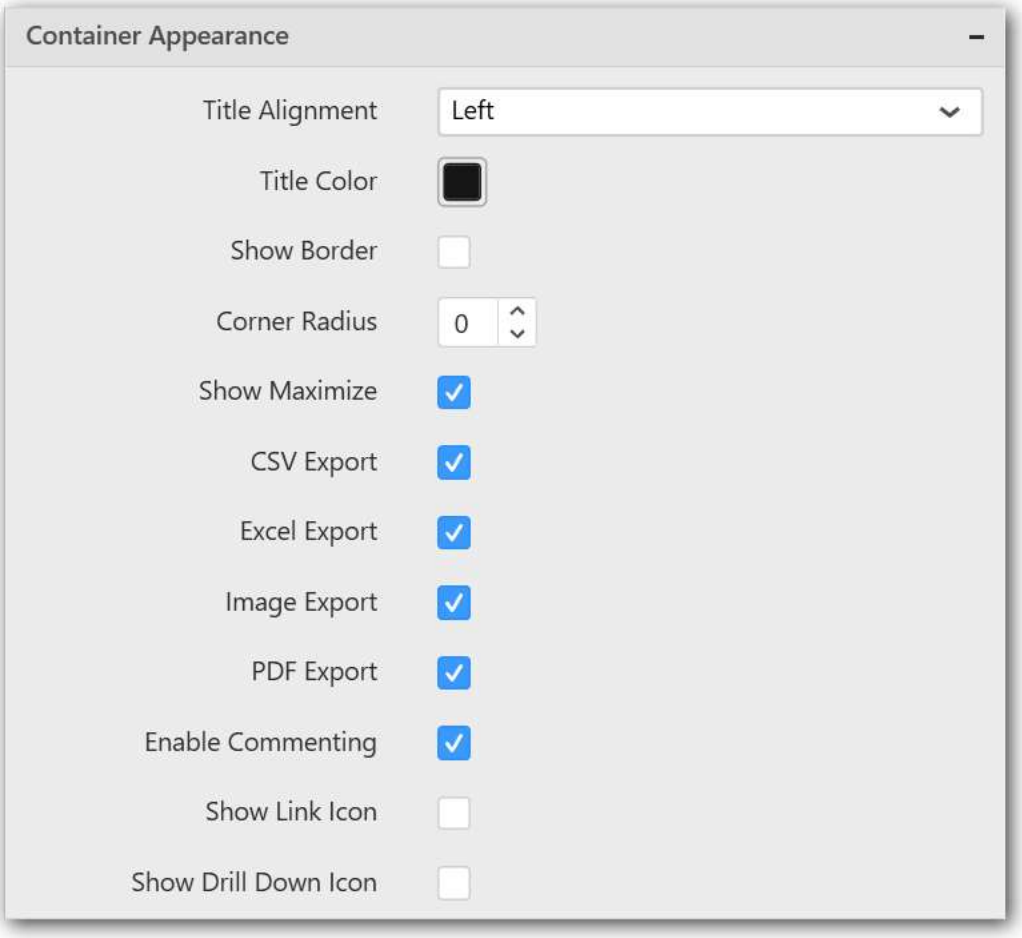
This allows you to define this pyramid chart widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings



To configure the linking to URL or dashboard with the widget through its settings. For more details, refer Linking.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this pyramid chart widget. The visibility of the maximize icon in widget header will be defined based on this setting in viewer.

CSV Export

This allows you to enable/disable the CSV export option for this pyramid chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel Export

This allows you to enable/disable the Excel export option for this pyramid chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

Image Export

This allows you to enable/disable the image export option for this pyramid chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

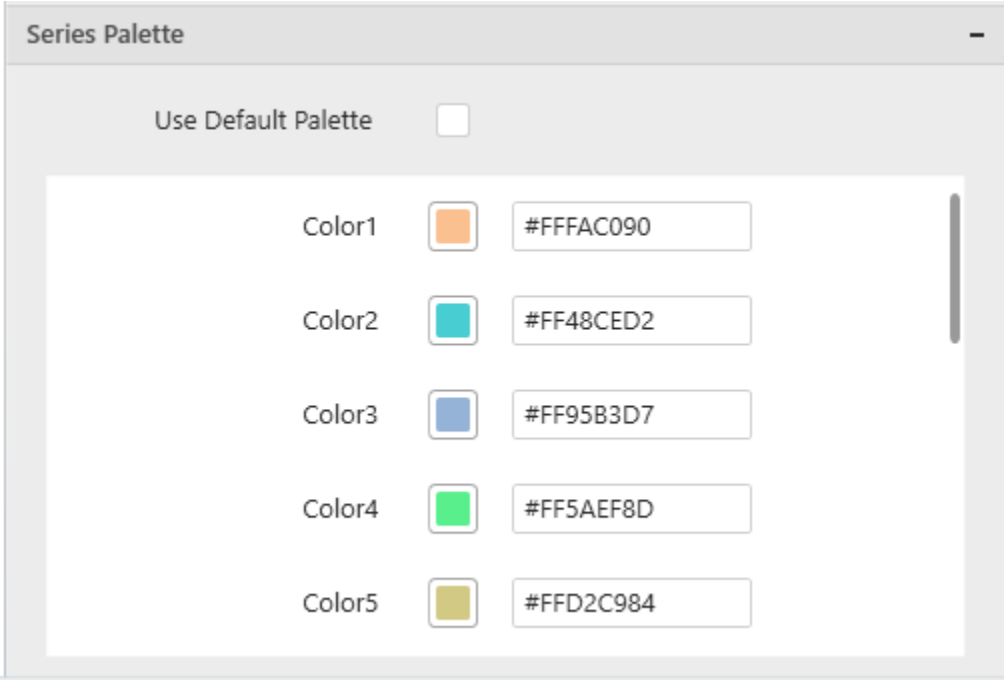
This allows you to enable/disable comment for dashboard widget. For more details refer here

Series Palette

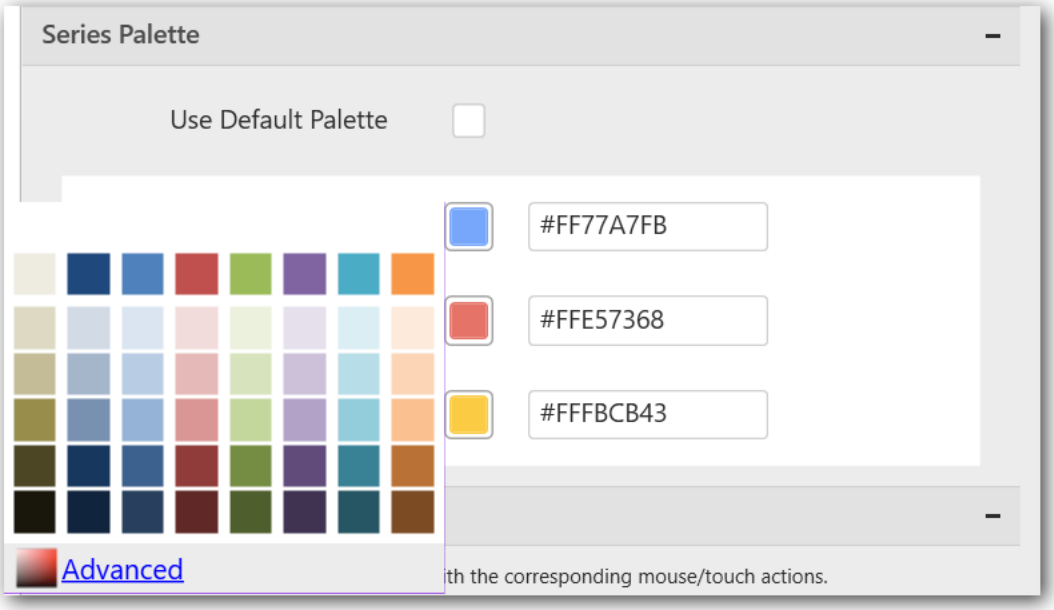
This allows you to customize the chart series color through Series Palette section.

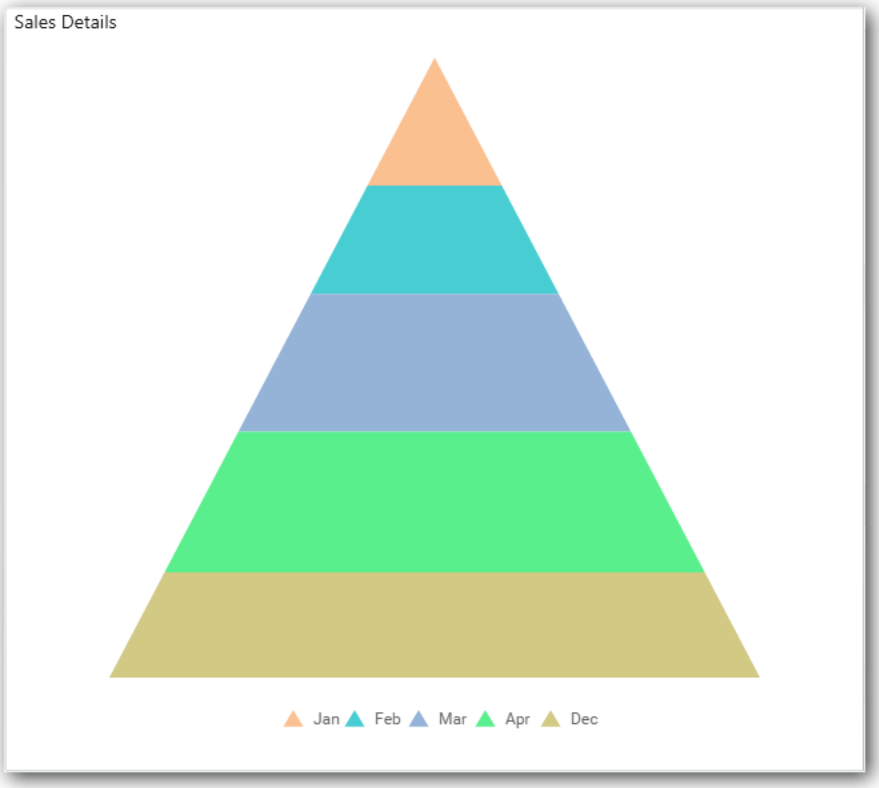
Use Default Palette

This allows you to toggle the series color between default palette and custom palette. By default, the property is toggled on and default palette will be applied to proportion series segments.



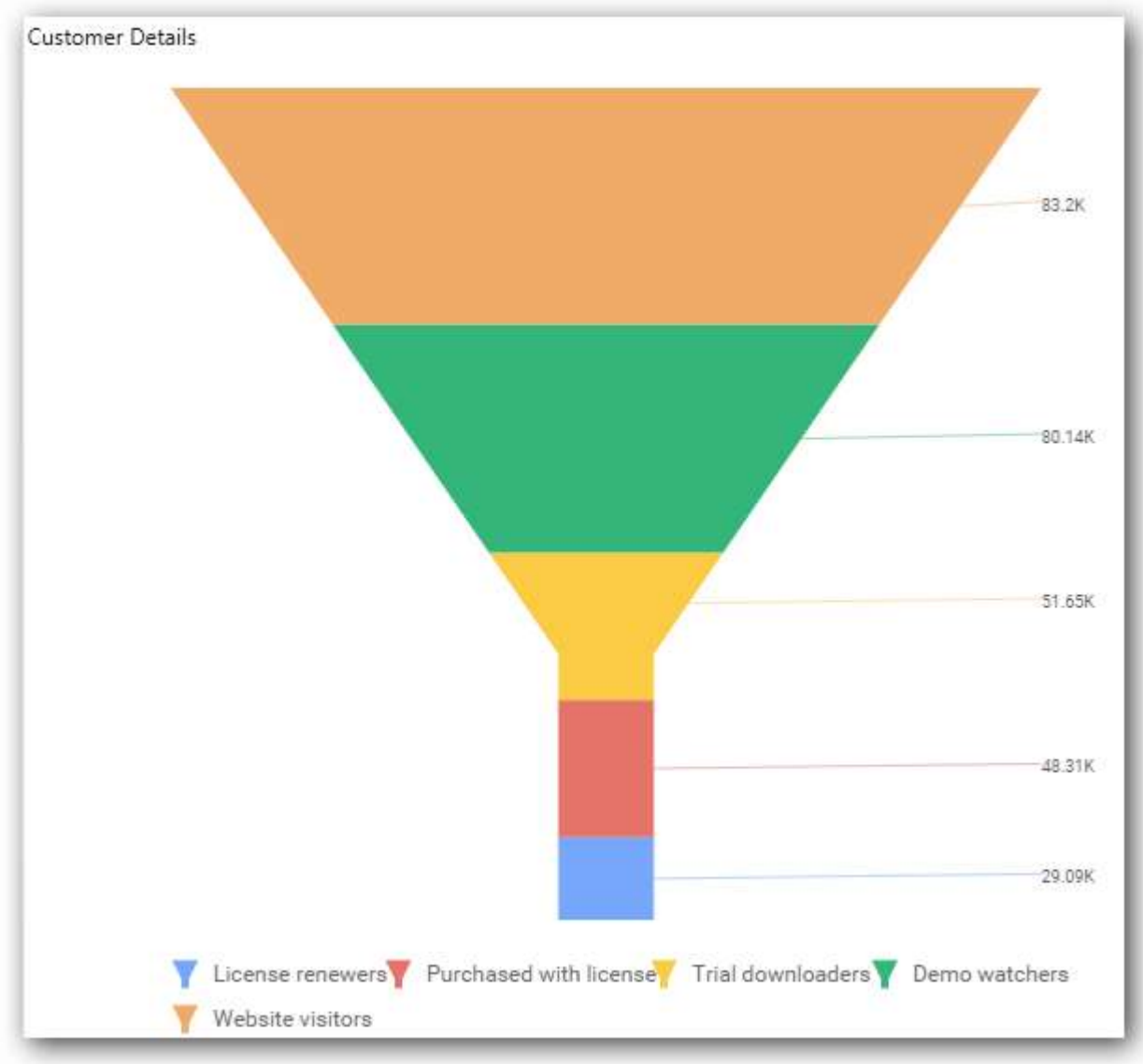
By toggle off the Use Default Palette, you can customize the proportion series segments' colors. This section shows a palette of colors. By clicking on the colored square, color picker will be opened. You can choose a color. And, you can also change the series color by changing the corresponding Hexadecimal value in the right-hand side.





Funnel Chart

Funnel Chart shows values across multiple stages in a process by highlighting different stages with different colors. It allows you make proportional comparison among values showcased in progressively decreasing manner.

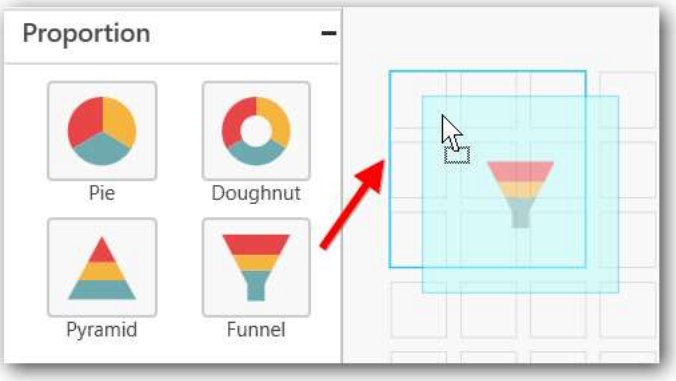


How to configure the flat table data to Funnel Chart

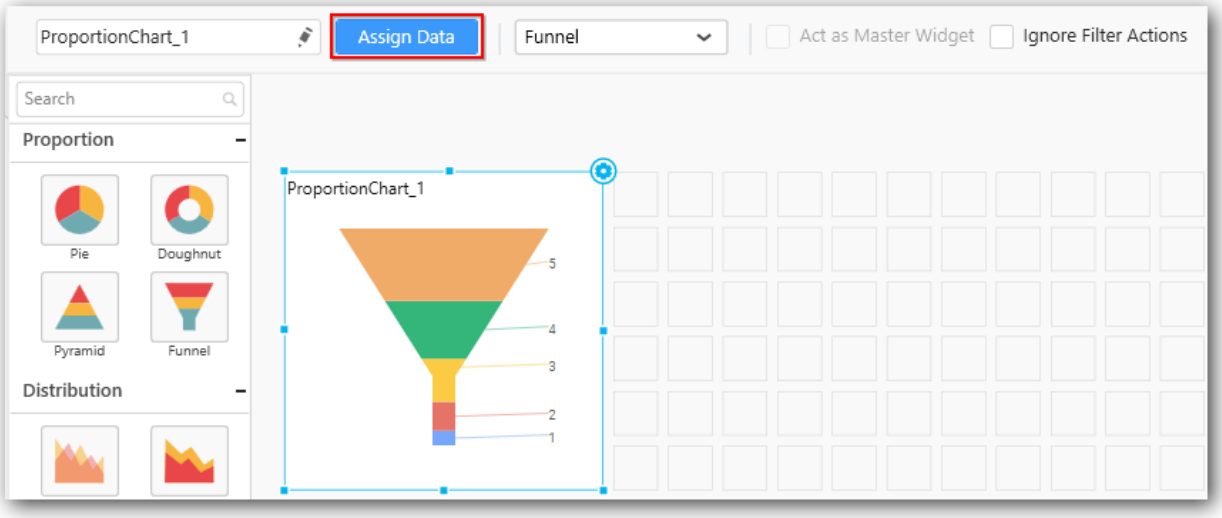
Funnel Chart needs a minimum of one value element and one column element to showcase. The measure or expression field that you want to analyze can be dropped into the Value(s) block. The dimension that you want to categorize the measure can be dropped into the Column block. To categorize the measure based on a series, drop the respective dimension into the Row block.

To configure data into the Funnel Chart, follow these steps:

1. Drag and drop the Funnel Chart into canvas and resize it to your required size.



2. Connect to the data source.
3. Focus the Funnel Chart and click **Assign Data**.

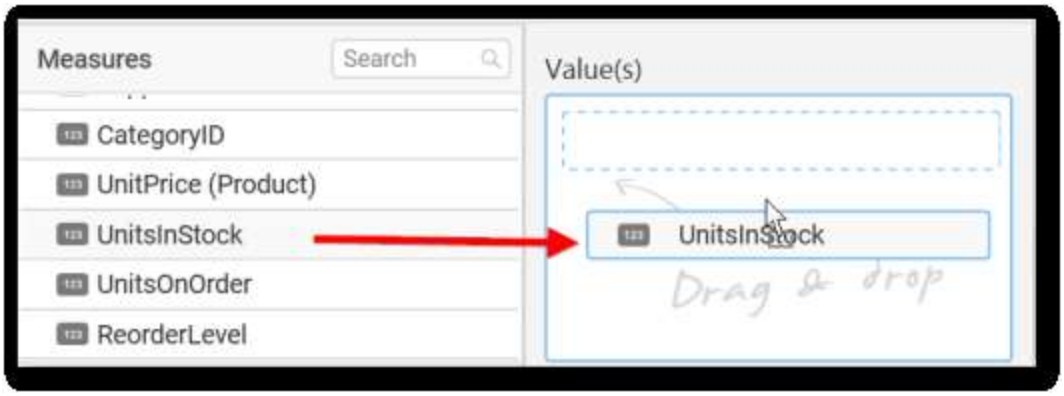


The data pane will be opened with available Measures and Dimensions from the data source.

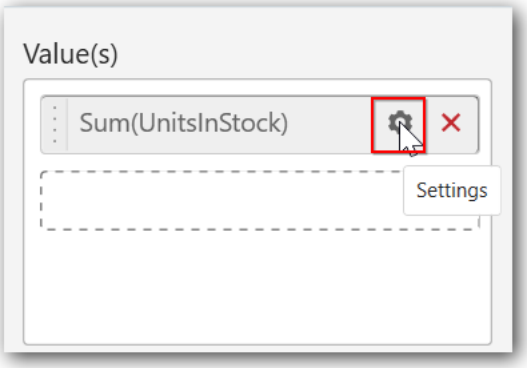
You can add the required data from Measures and Dimensions into the required field. You can create any mathematical, Boolean, or any expressions using columns, rows, or values in the Expression Columns using the Expression Designer.

Adding value(s)

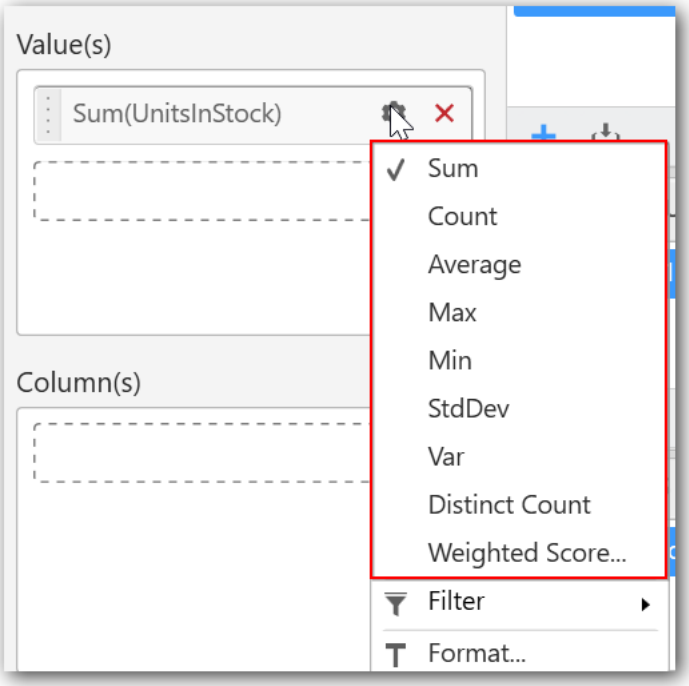
Add the Measures fields into Value(s) section by dragging and dropping the required measure fields.



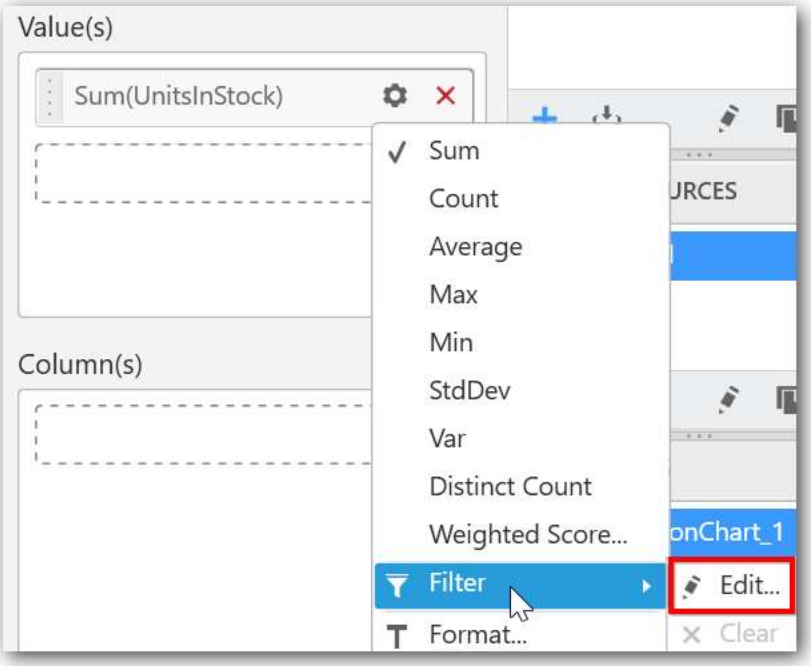
You can change the properties of the required field by using the settings option.



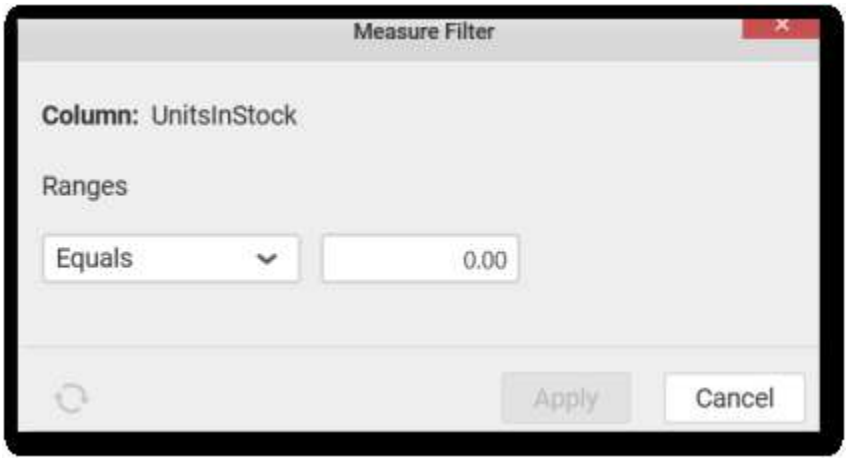
Select the required summary type from the available summary types shown in the settings.



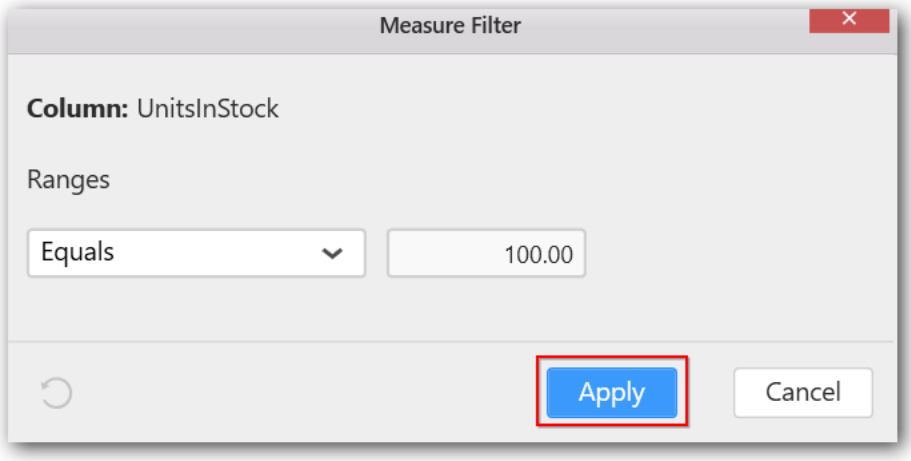
You can filter the data to be displayed in the Funnel Chart by using the filter. It is used to filter the values in numbers by setting the filter range as greater than, less than, etc.



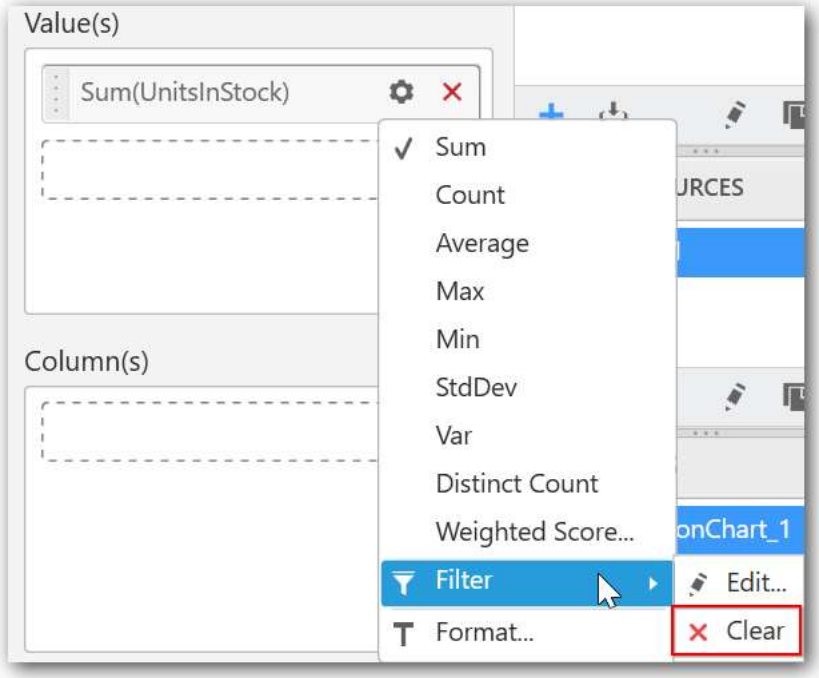
When you click **Edit**, the Measure Filter option will appear.



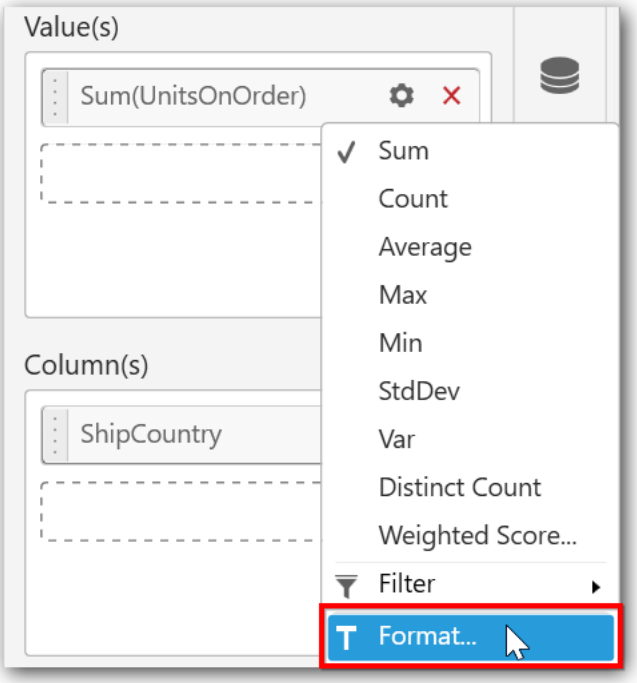
You can select the condition to be applied in the shown list box and set the value in text box, and then click **Apply**.



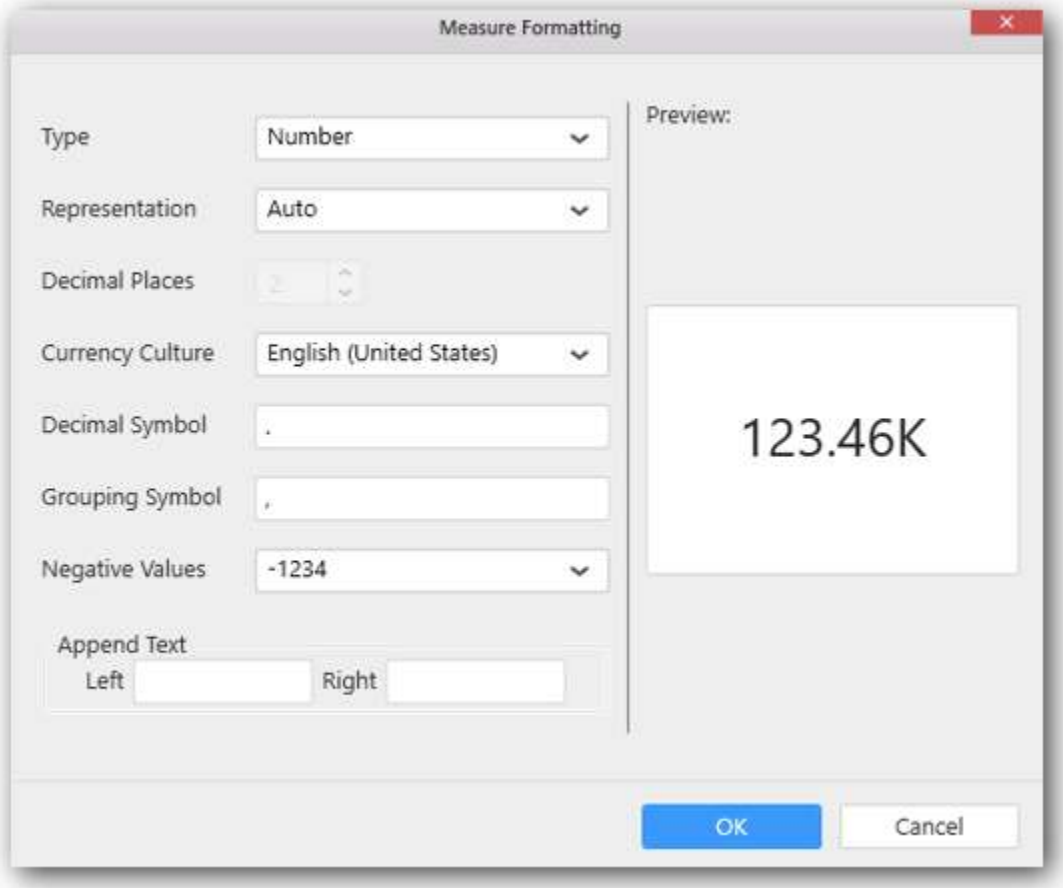
You have an option to clear the applied filter. Click **Clear** to remove the filters.



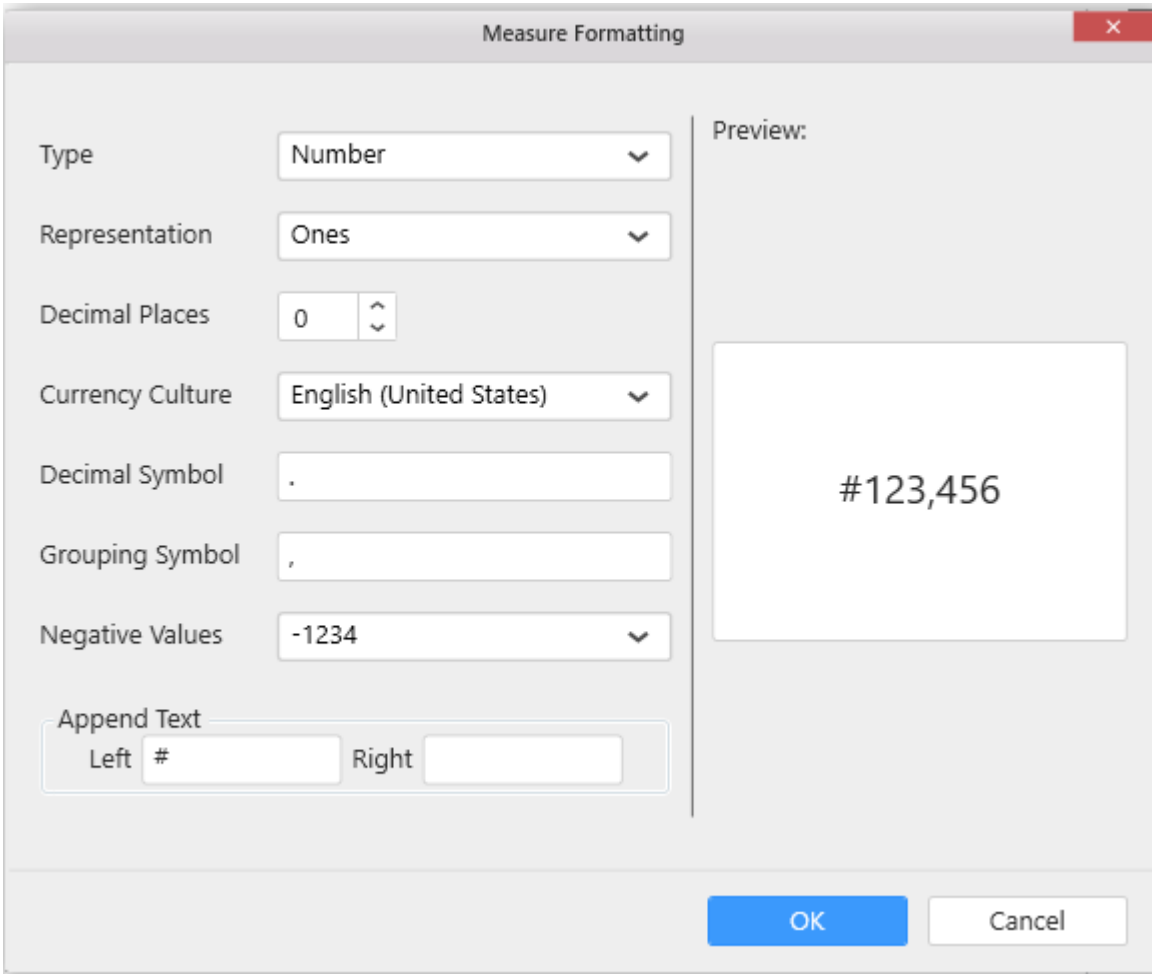
You can format the data to be displayed in the Funnel Chart by using the Format option.



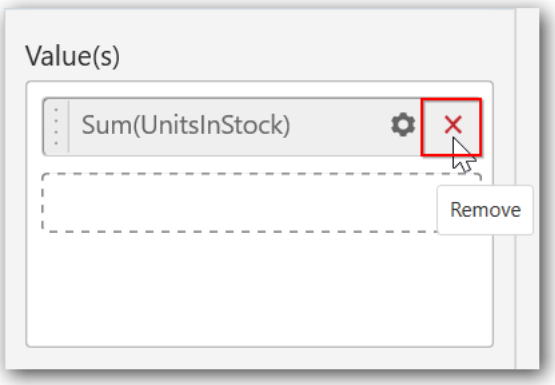
The Measure Formatting option will be shown as follows. Select the required format and click **OK**.



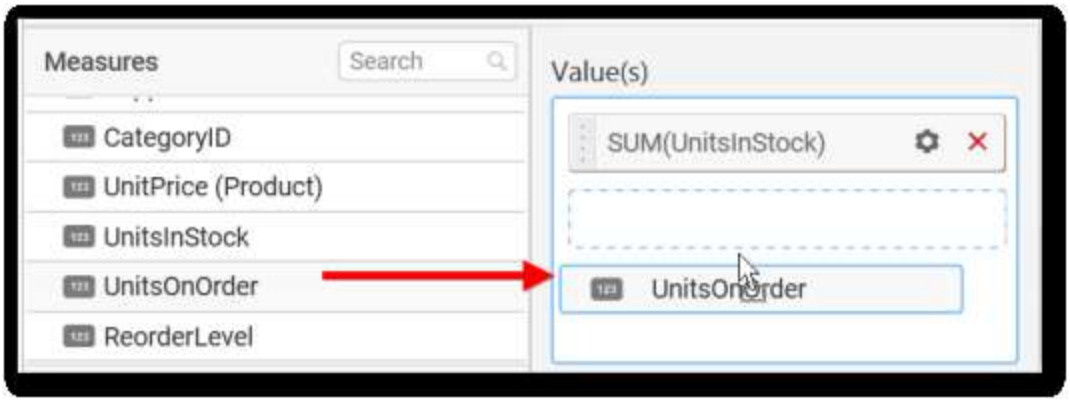
Enter the text to be displayed in the left side of the value in the Append Text as follows and click **OK** to proceed. Similarly, you can provide the required text in the right side of the value.



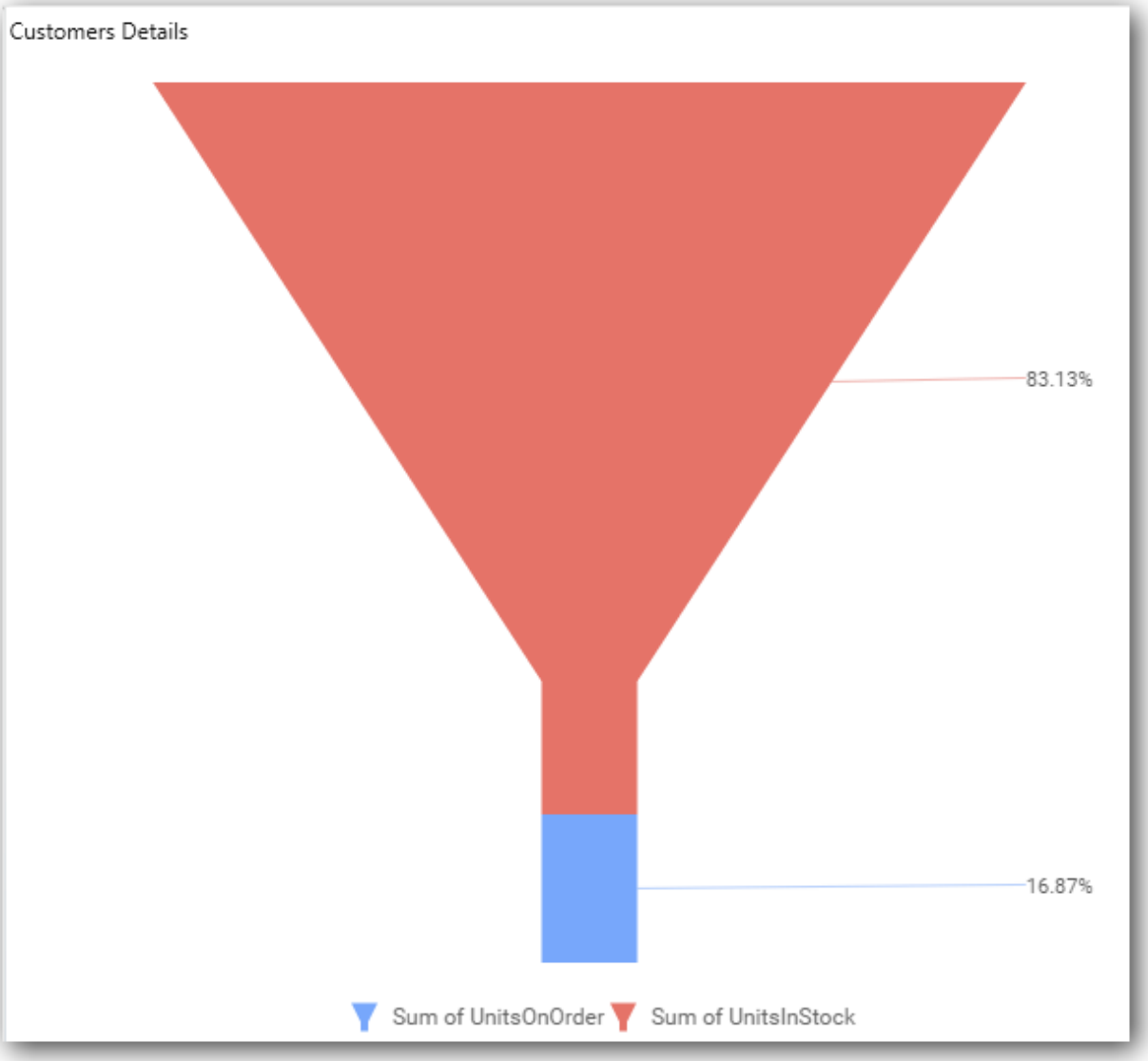
To remove the added value fields, click the x button.



You can add multiple Measures into Value(s) field.



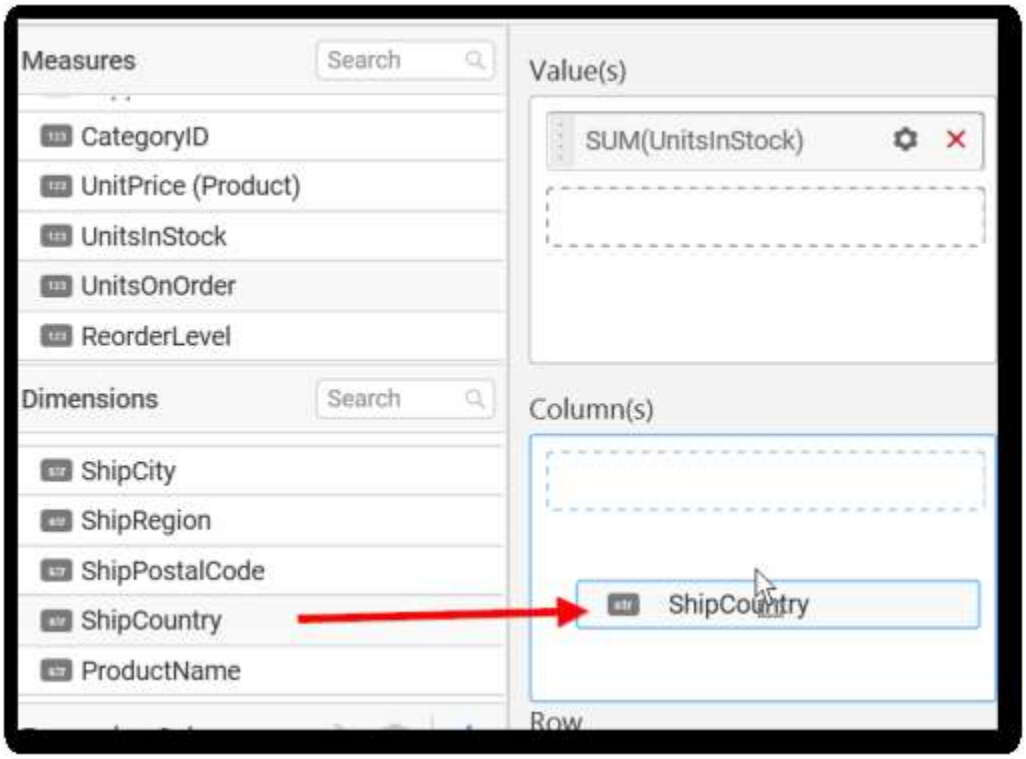
The chart will be rendered as follows.



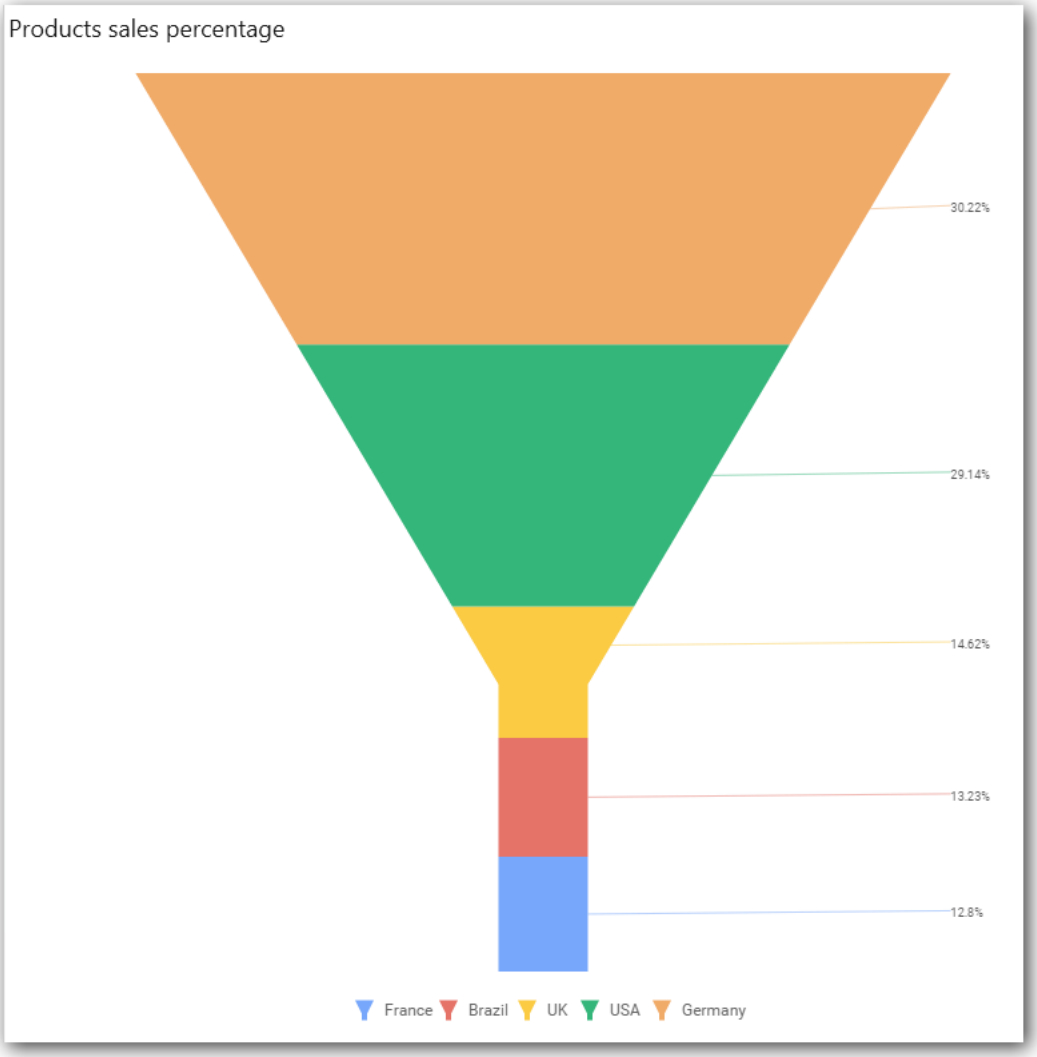
You can also drag the Dimensions and Expression Columns into Value(s).

Adding column(s)

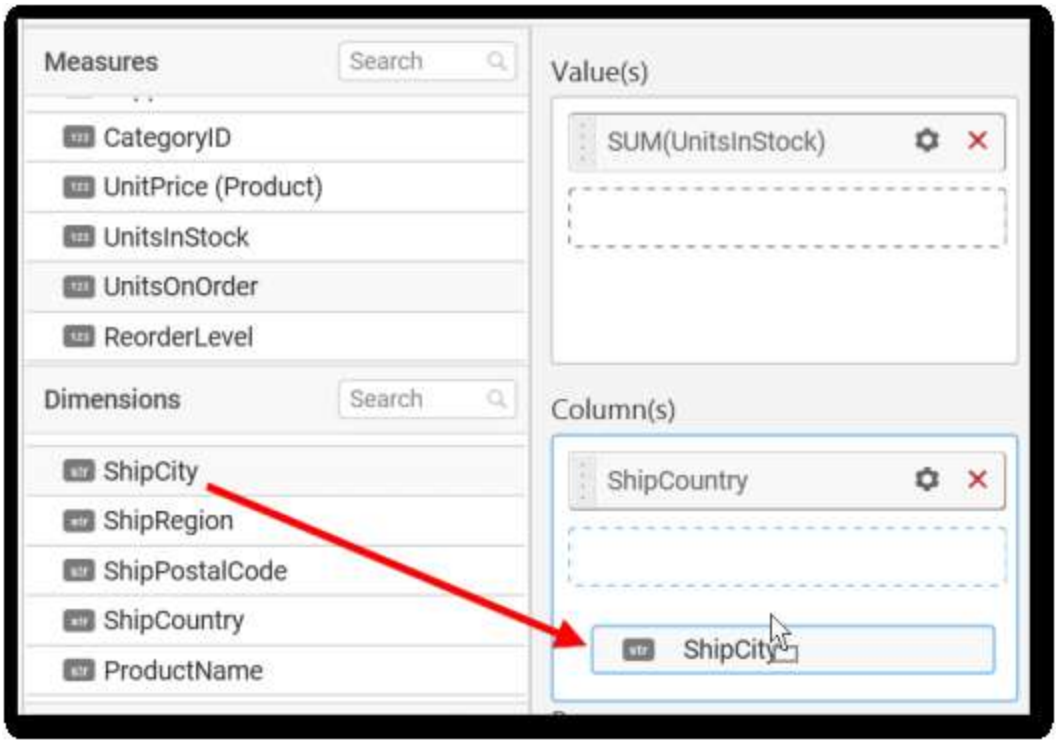
Drag and drop the Dimensions into Column(s).



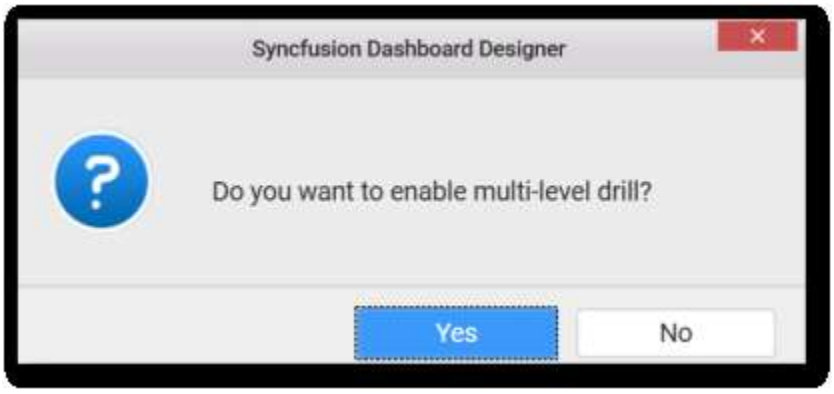
The Funnel Chart will be rendered as follows.



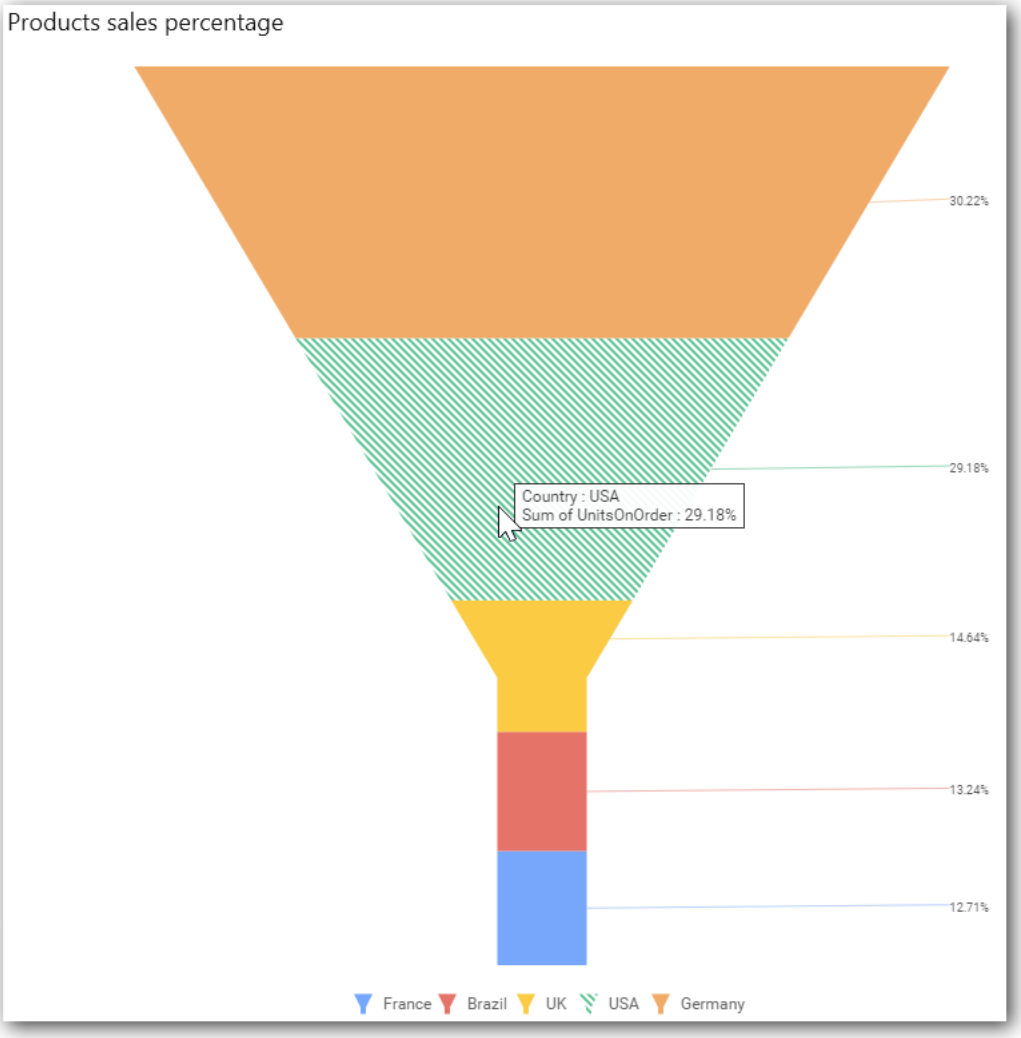
You can add more than one value into the Column(s) field.



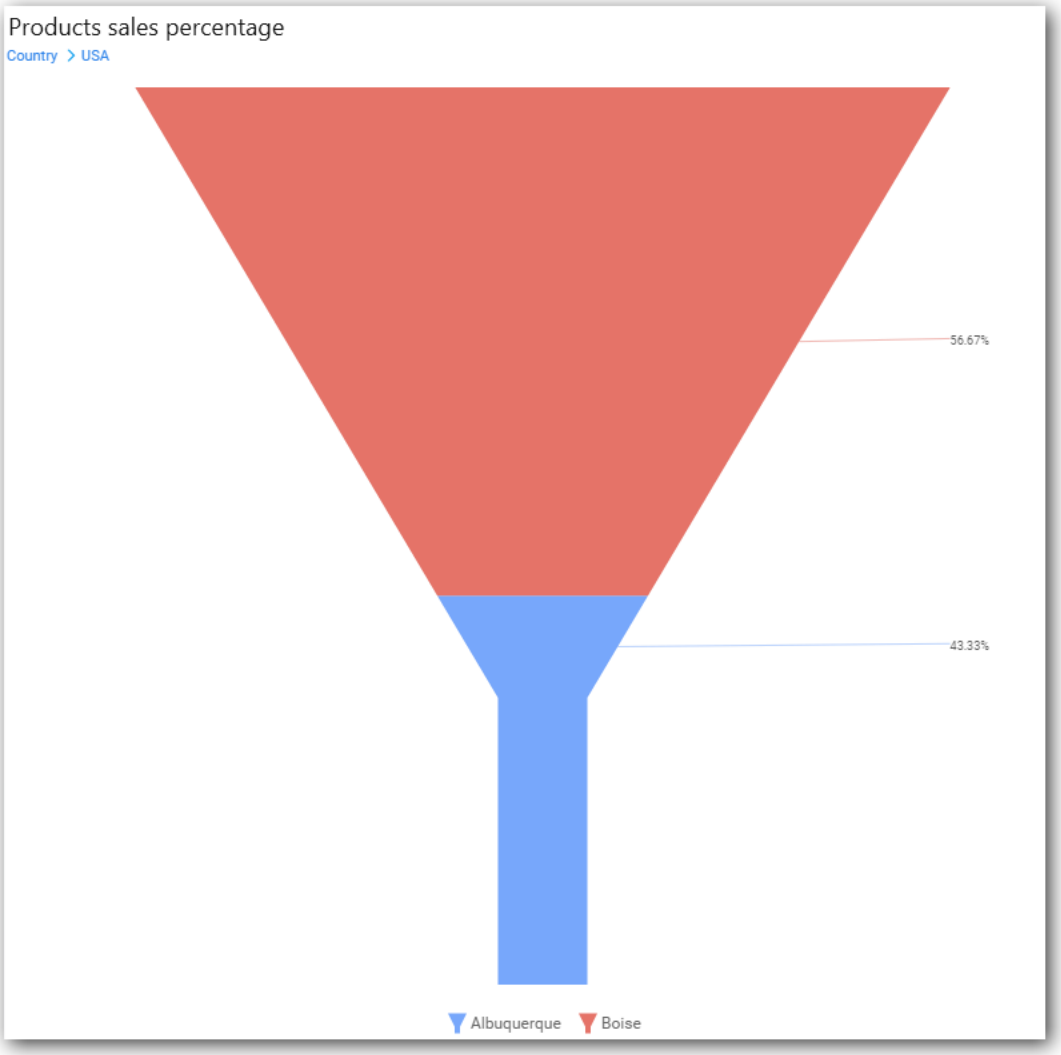
The following message will open.



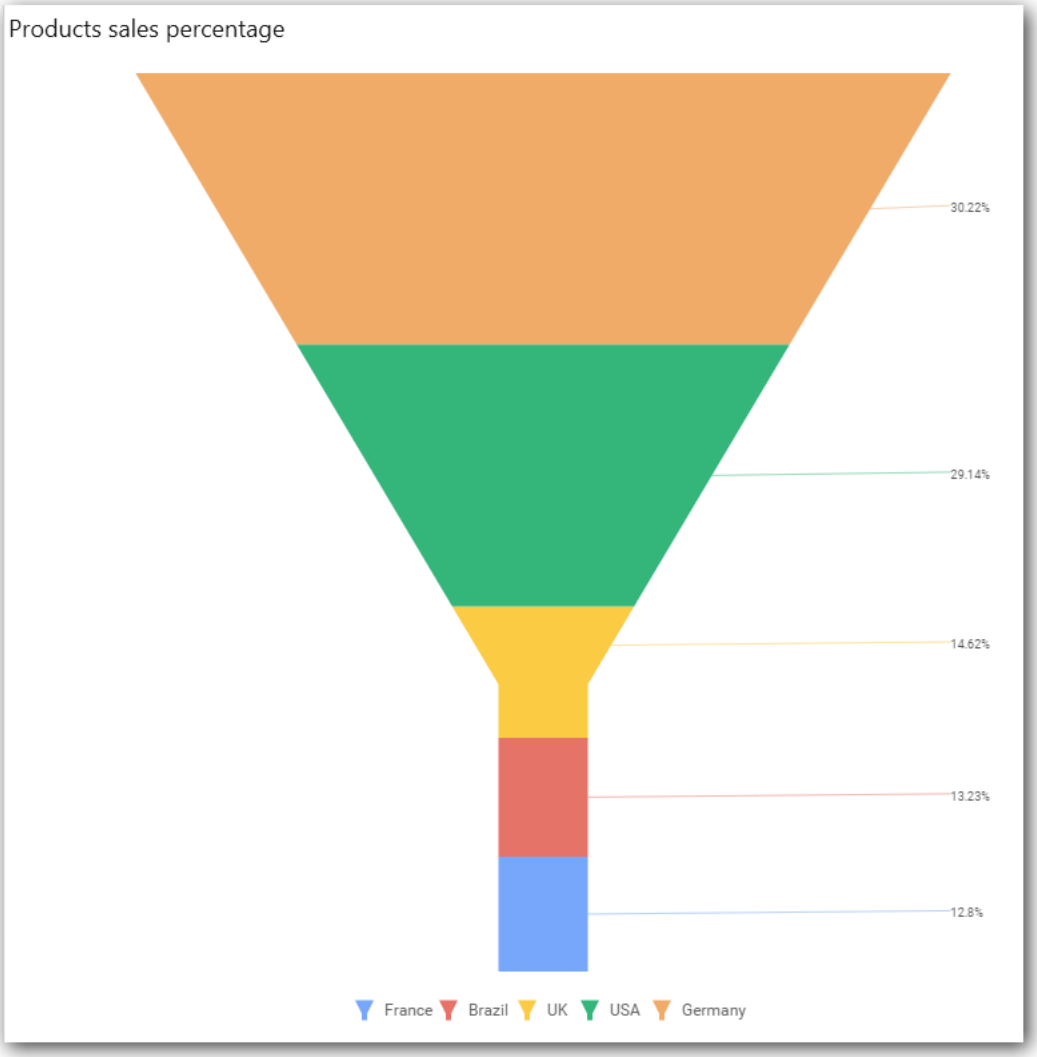
You can enable this option to get the further details about the selected chart region. To enable drill down, click **Yes**.



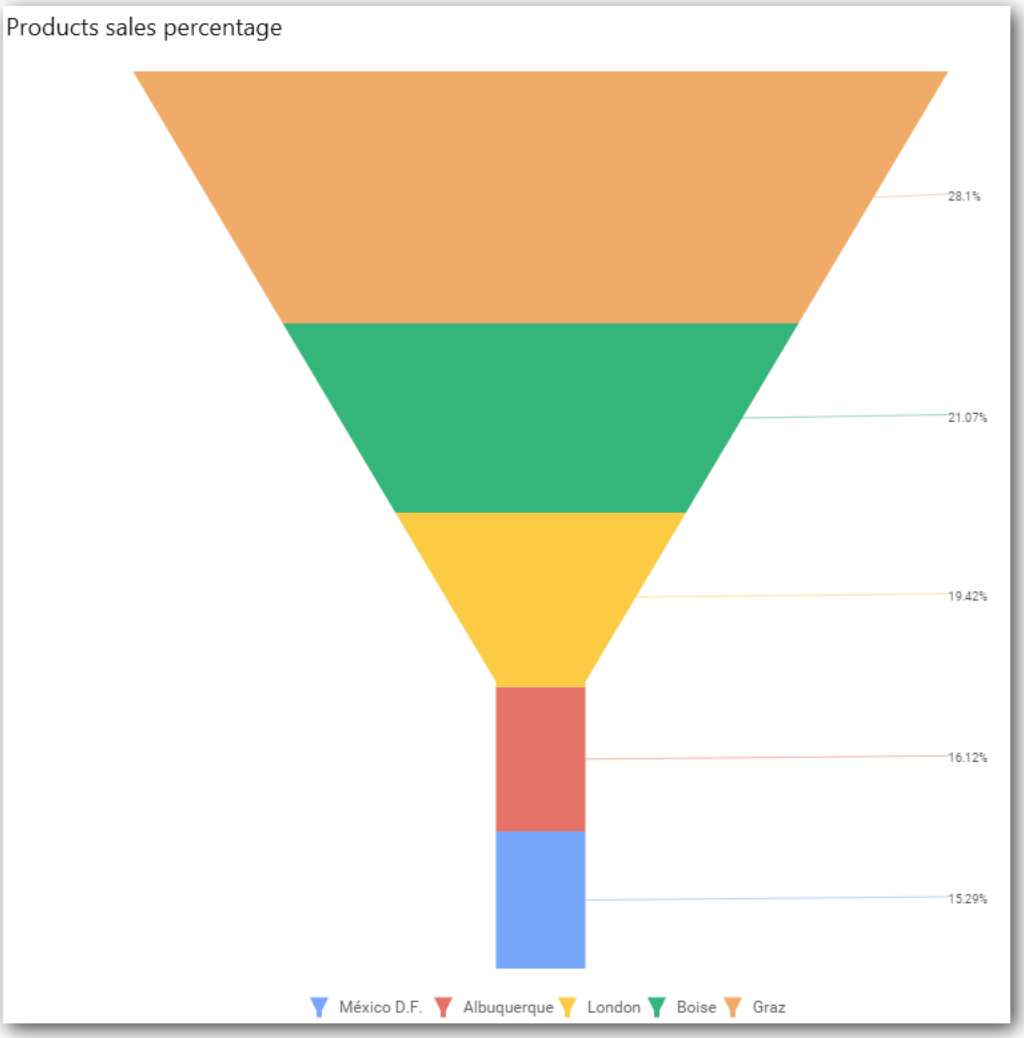
The drilled view of the selected chart region.



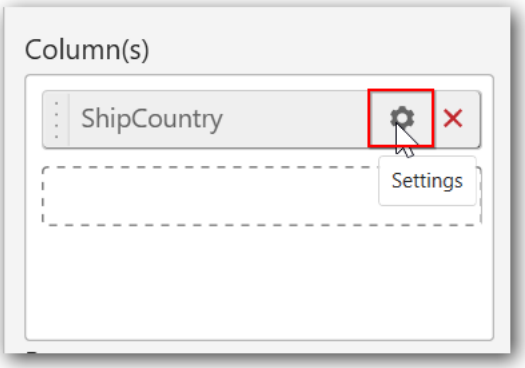
To continue without drill down, click **NO**.



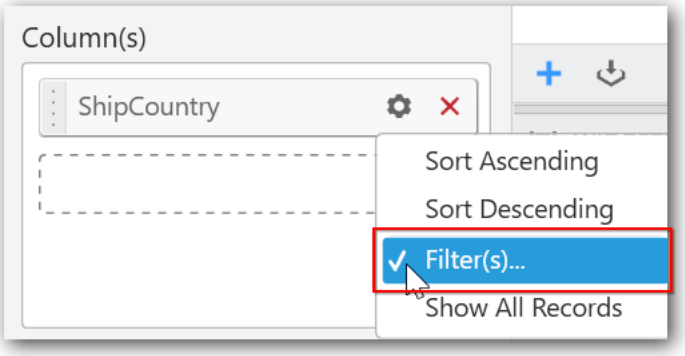
The old column value will be replaced by a new column value.



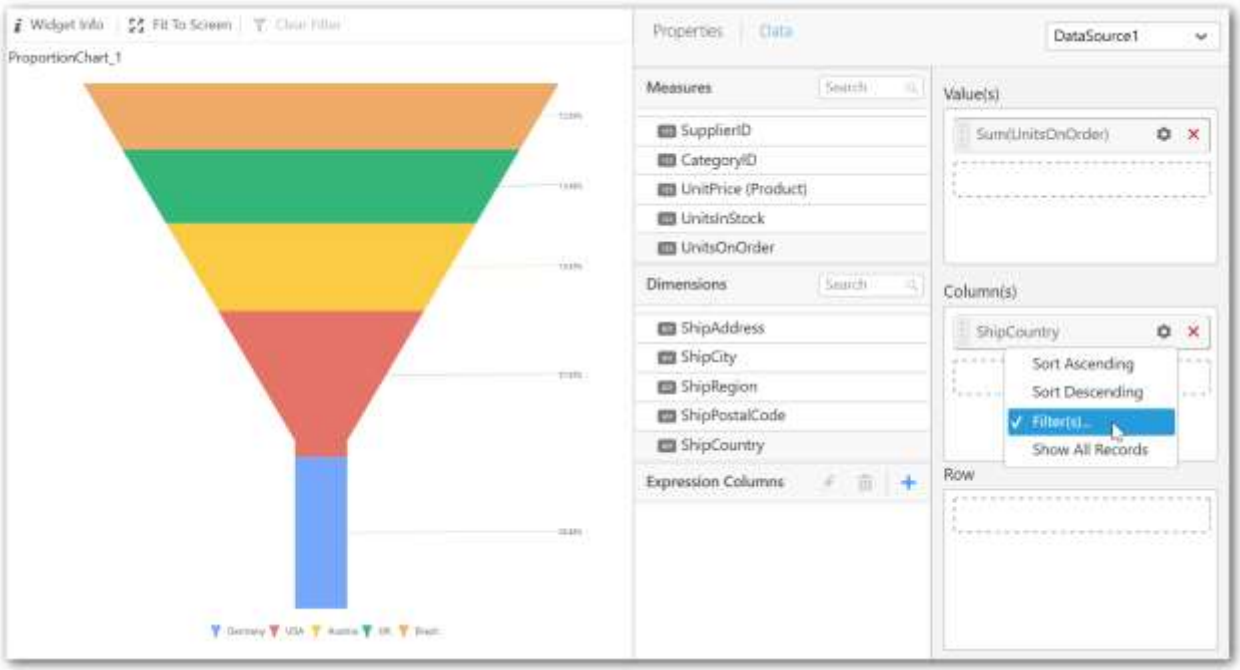
You can change the properties of the required field by using the Settings option.



You can apply filters by selecting the filter option in the settings. This filter option is used to filter the strings like top 5 records and bottom 5 records.

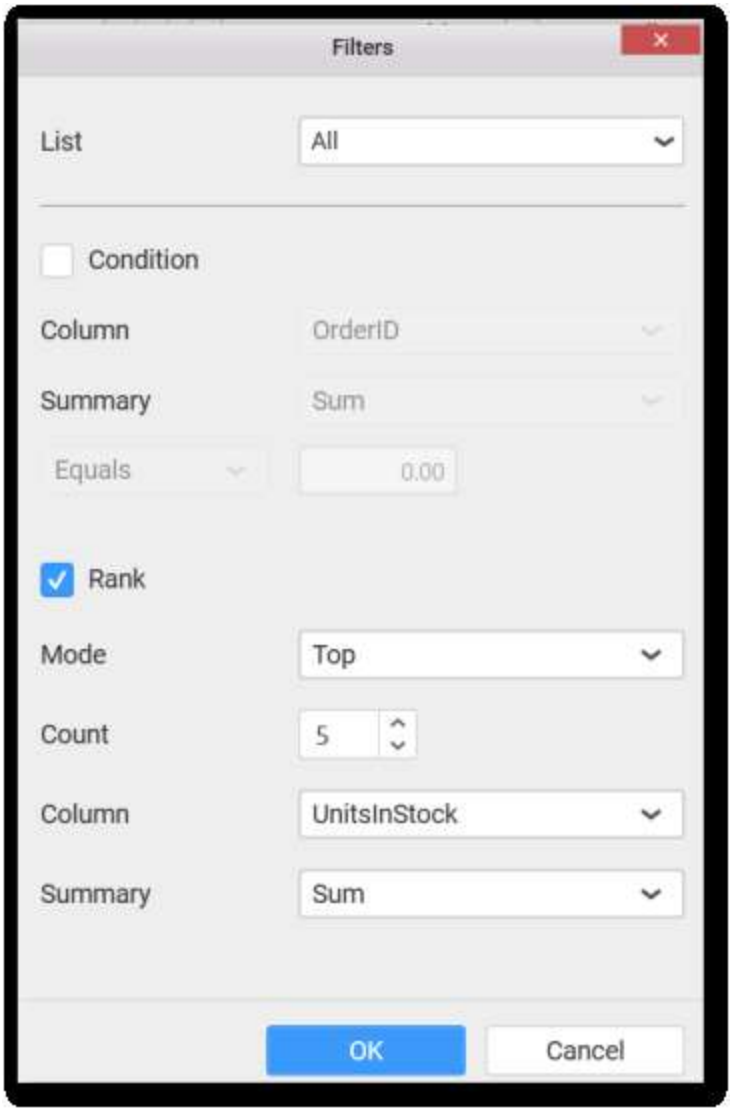


Note: By default, the filter will be set for top 5 records.



The filters option will open.

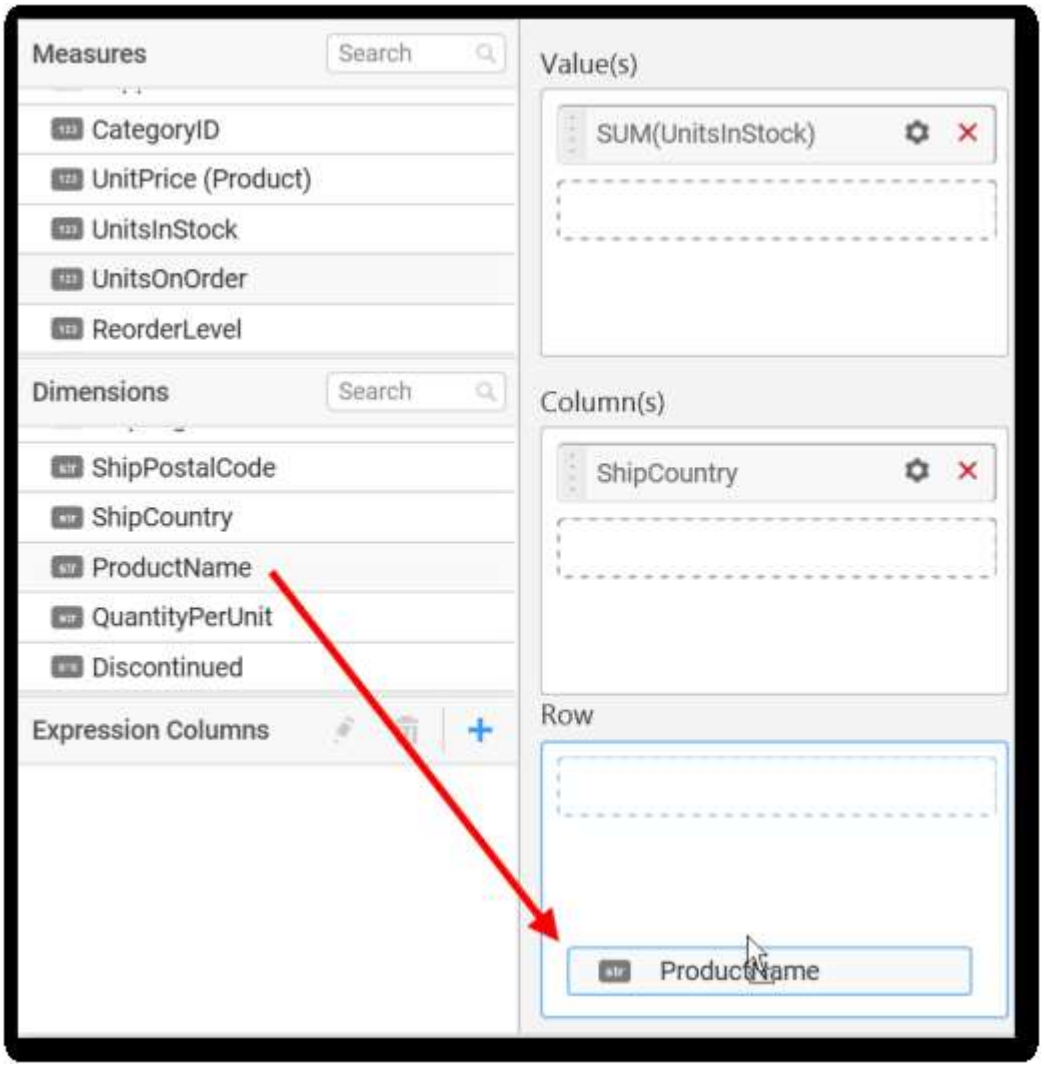
Select the required Conditions and Rank, and then click **OK**.



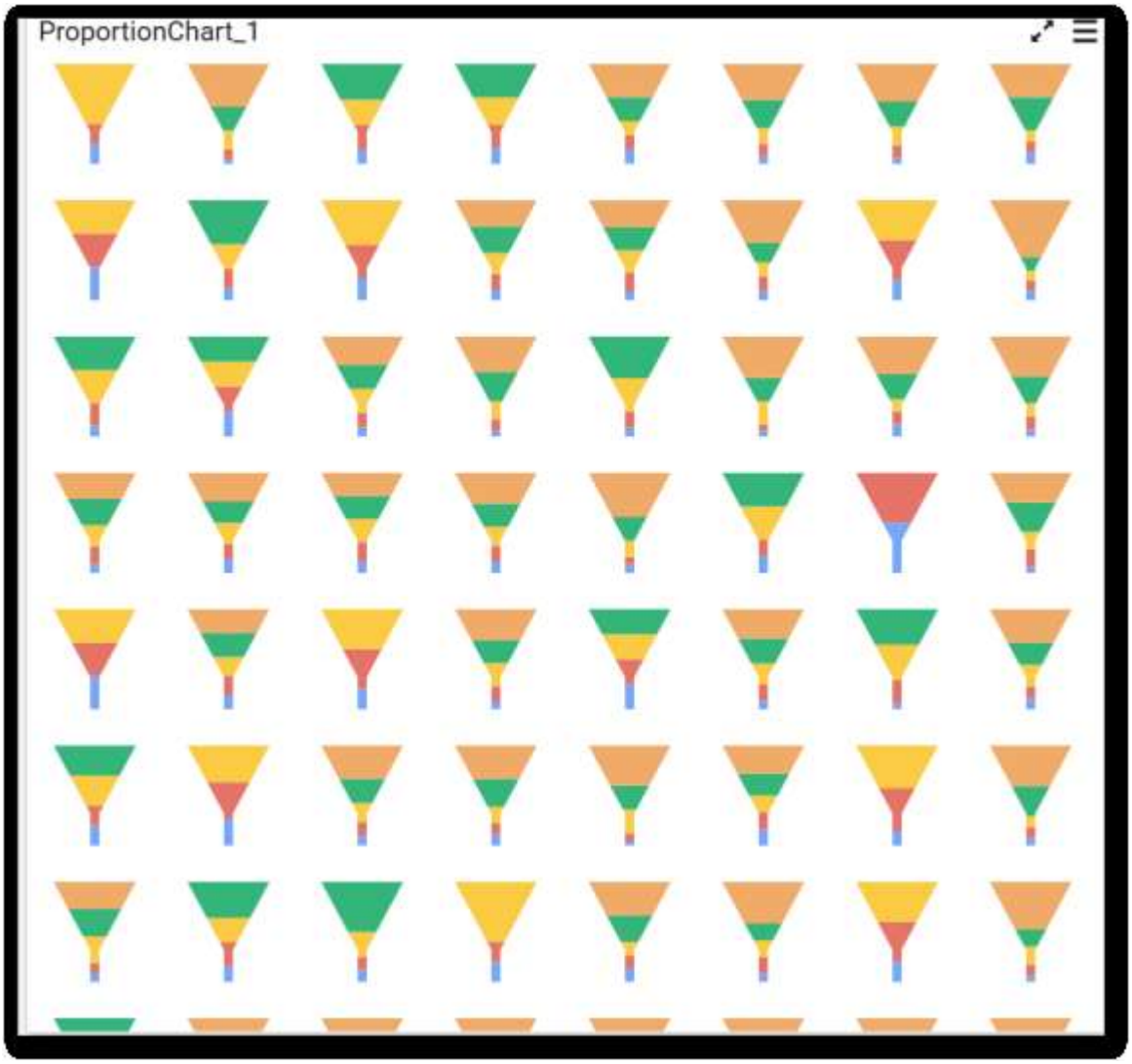
Similarly, you can add the Measures and Expression Columns into the Column field.

Adding row

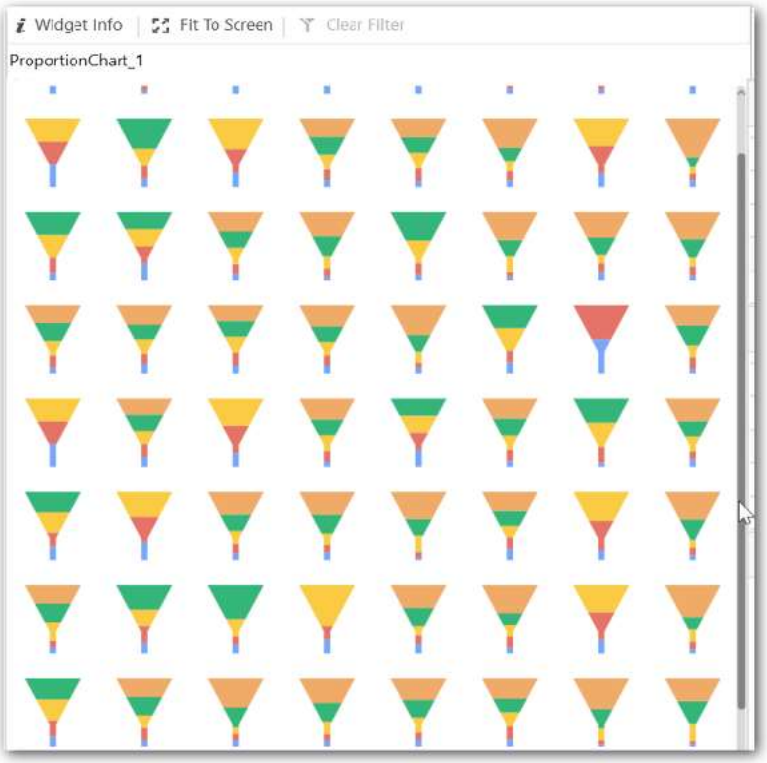
You can drag and drop the Measure or Dimension into the Row field.



This will render the Funnel Chart in series.



Scroll to see all charts.

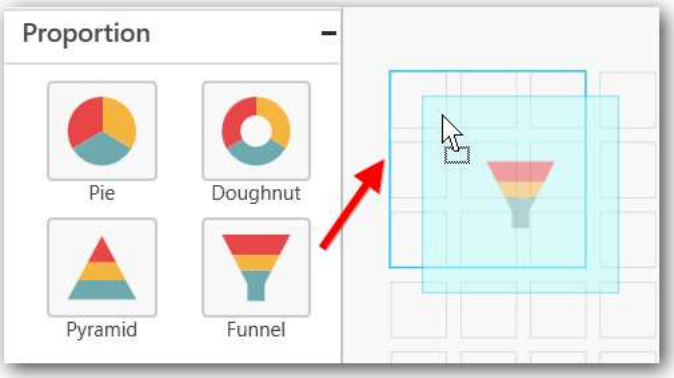


How to configure the SSAS data to Funnel Chart

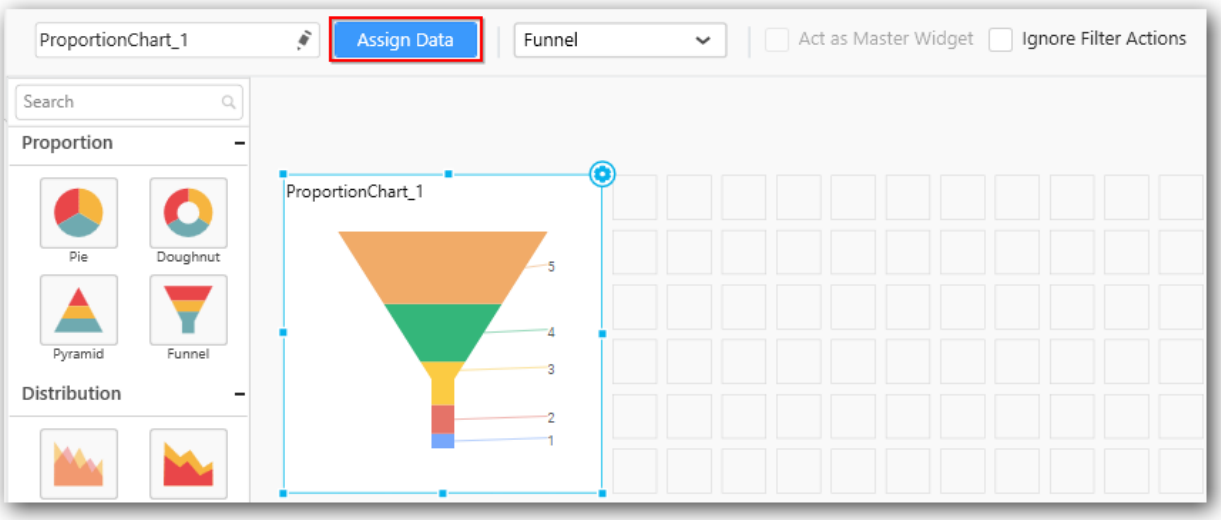
Funnel Chart needs a minimum of one value element and one column element to showcase. The measure or expression field that you want to analyze can be dropped into the Value(s) block. The dimension that you want to categorize the measure can be dropped into the Column block. To categorize the measure based on a series, drop the respective dimension into the Row block.

The following steps illustrate the configuration of SSAS data to Funnel chart.

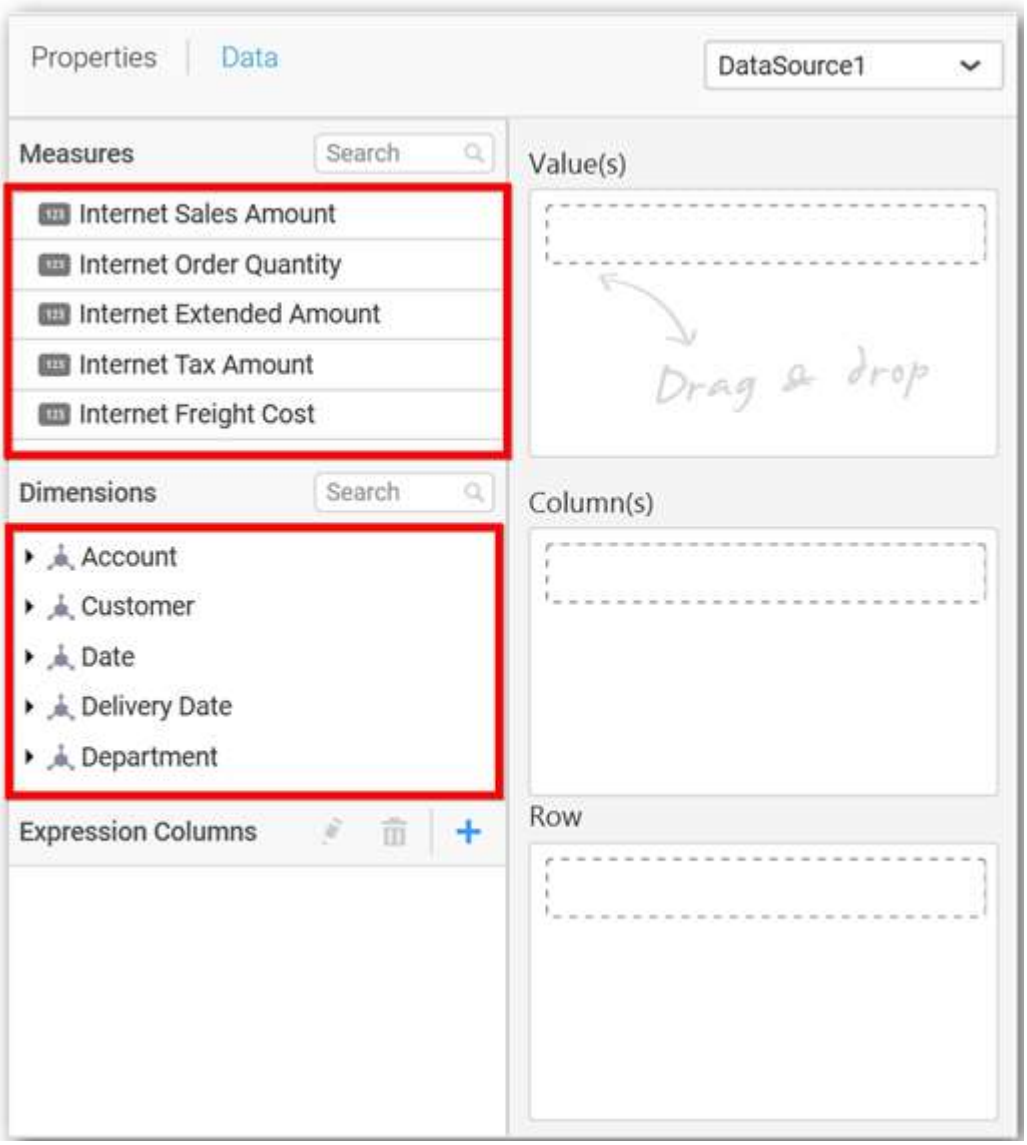
Drag and drop the Funnel chart into the canvas and resize it to your required size.



Select the dropped widget and click **Assign Data** in the toolbar.

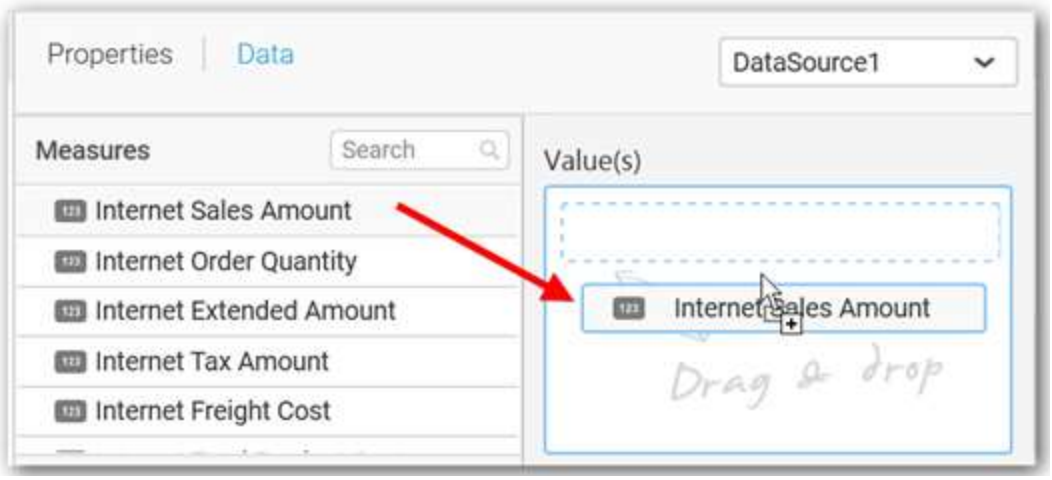


A data pane will be opened with available Measures and Dimensions.

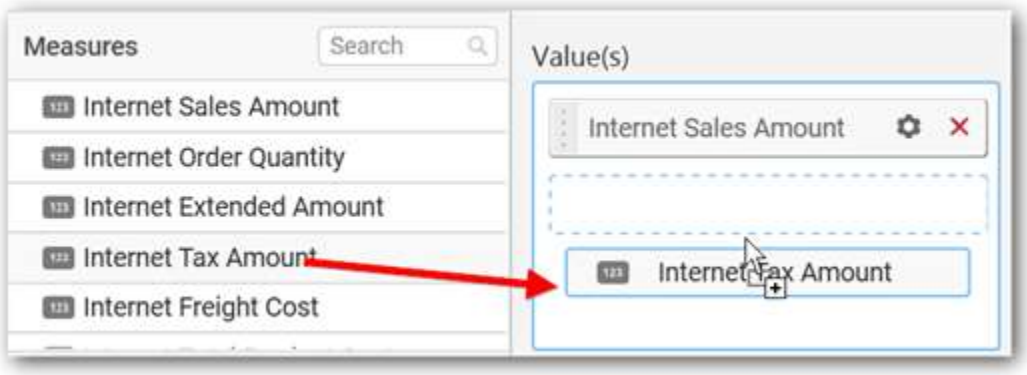


Assigning value(s)

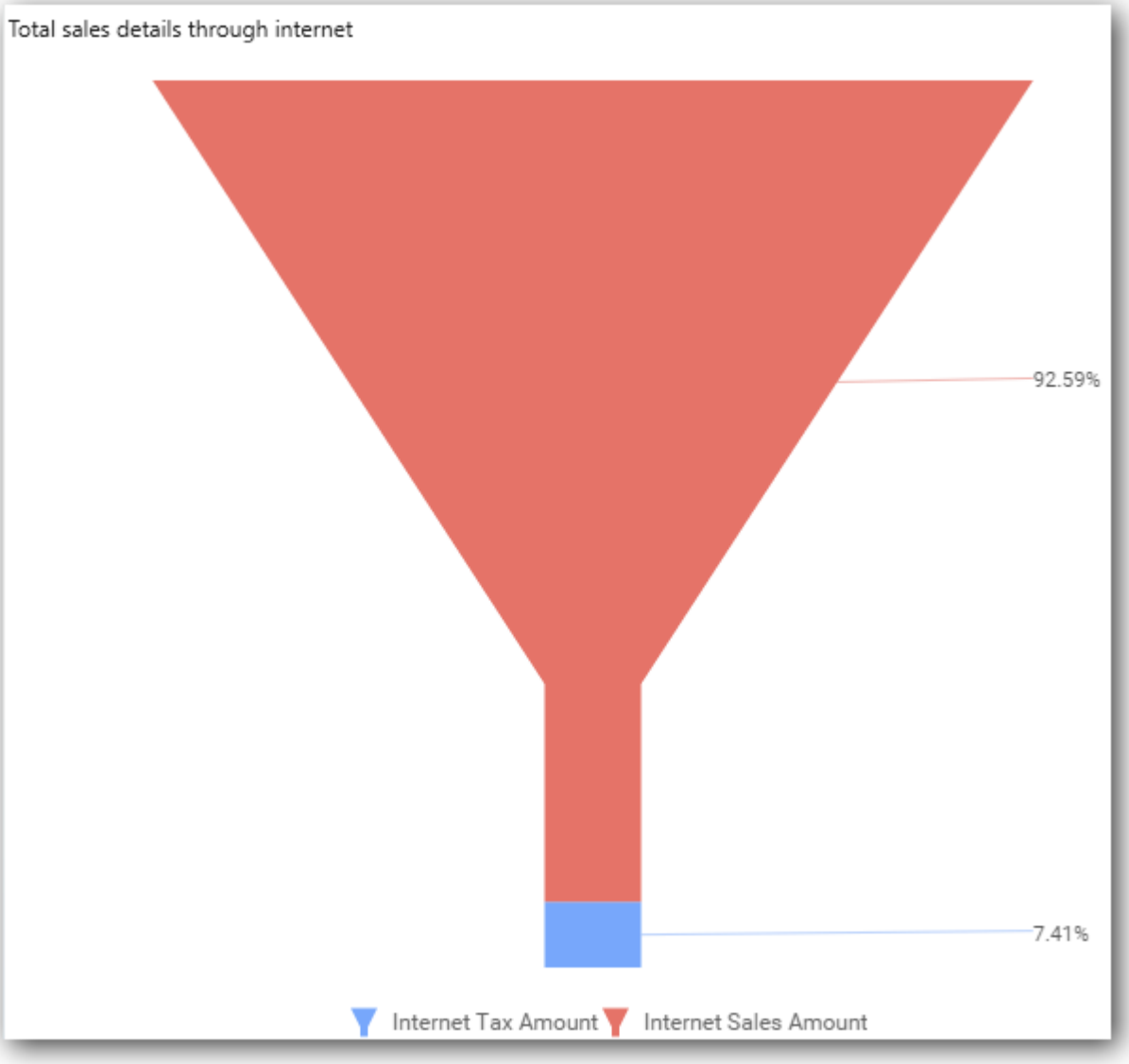
Drag and drop a column under Measures category into Value(s).



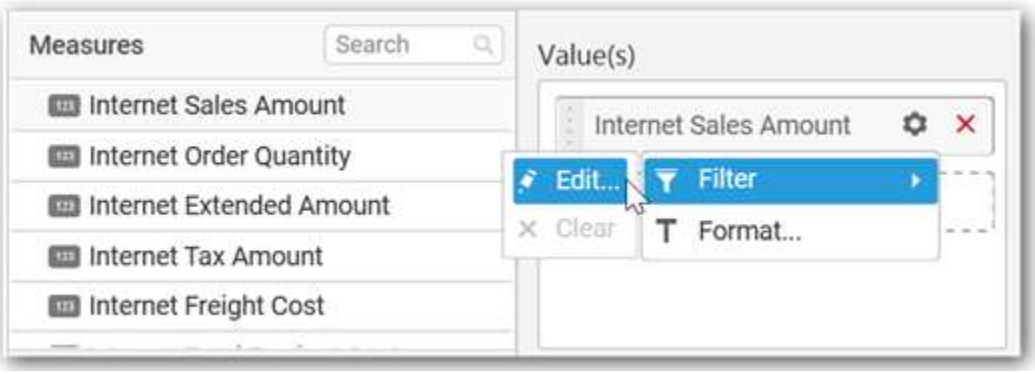
You can also add more than one column to the Value(s).



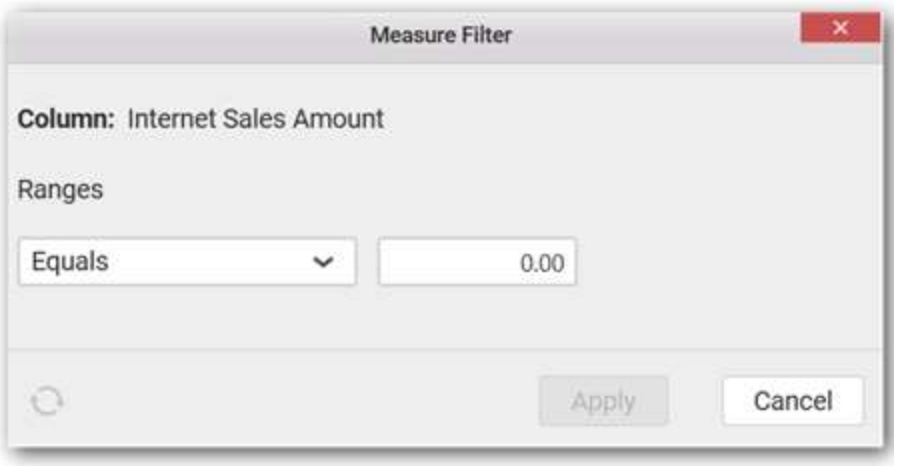
Now, the chart will be rendered as follows.



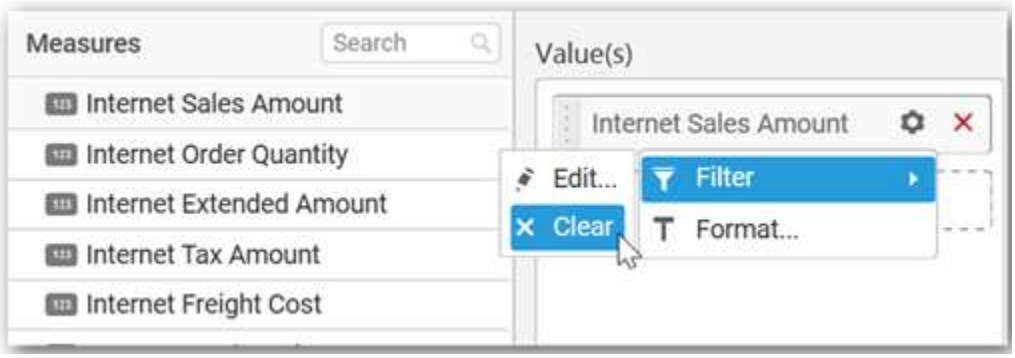
Define the filter criteria to match by choosing **Edit** in the Filter menu item.



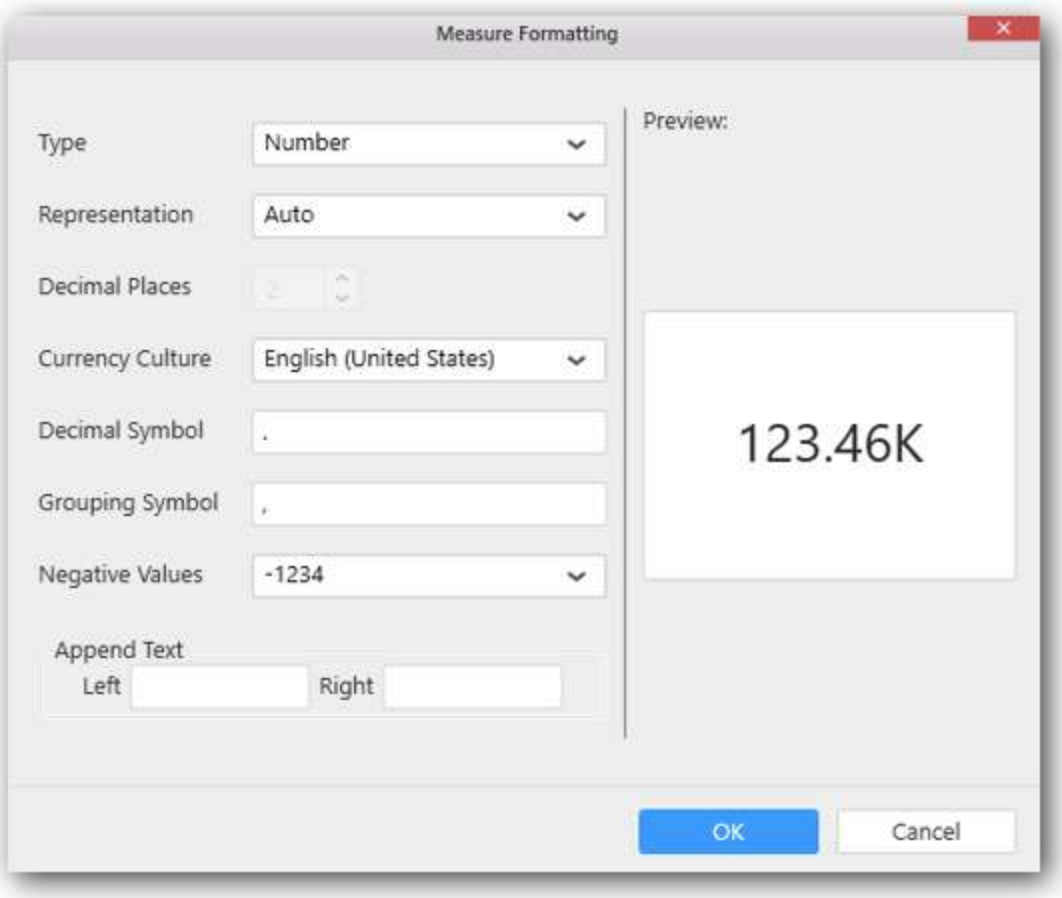
The Measure Filter dialog will be shown, where you can choose the filter condition and apply the condition value.



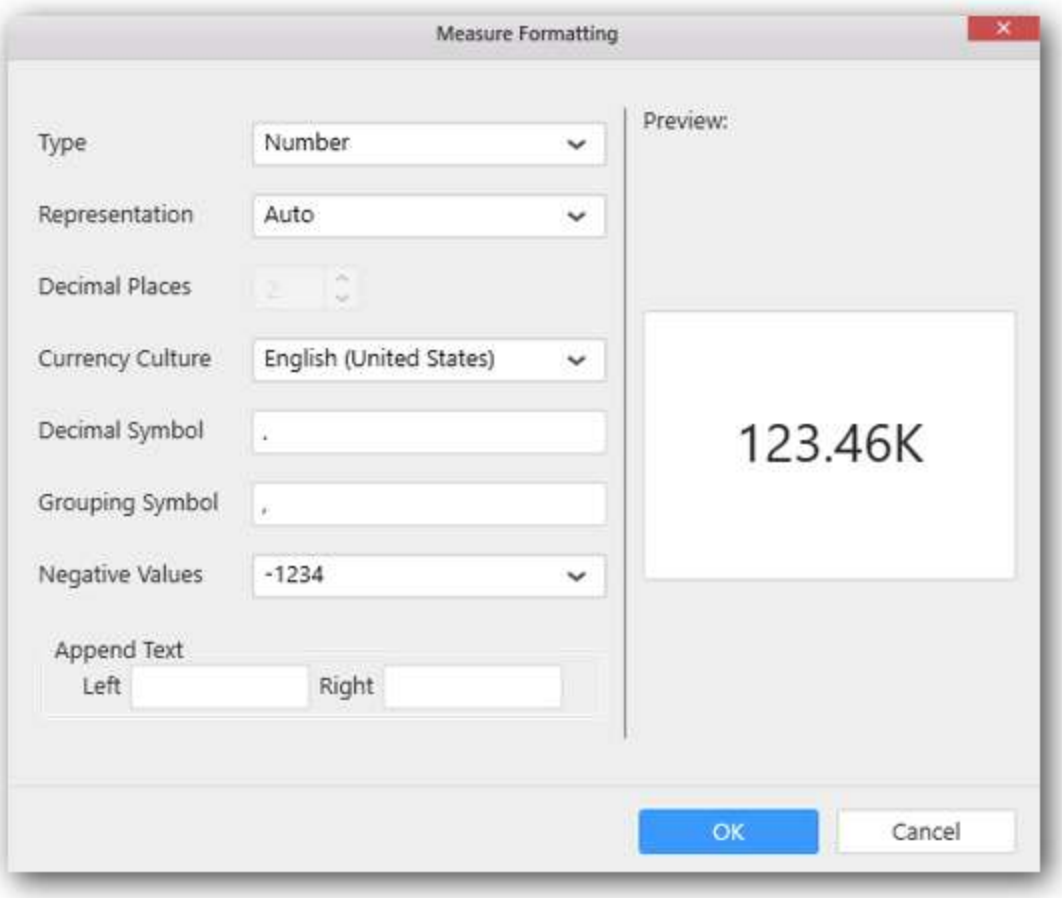
Select **Clear** to clear the defined filter.



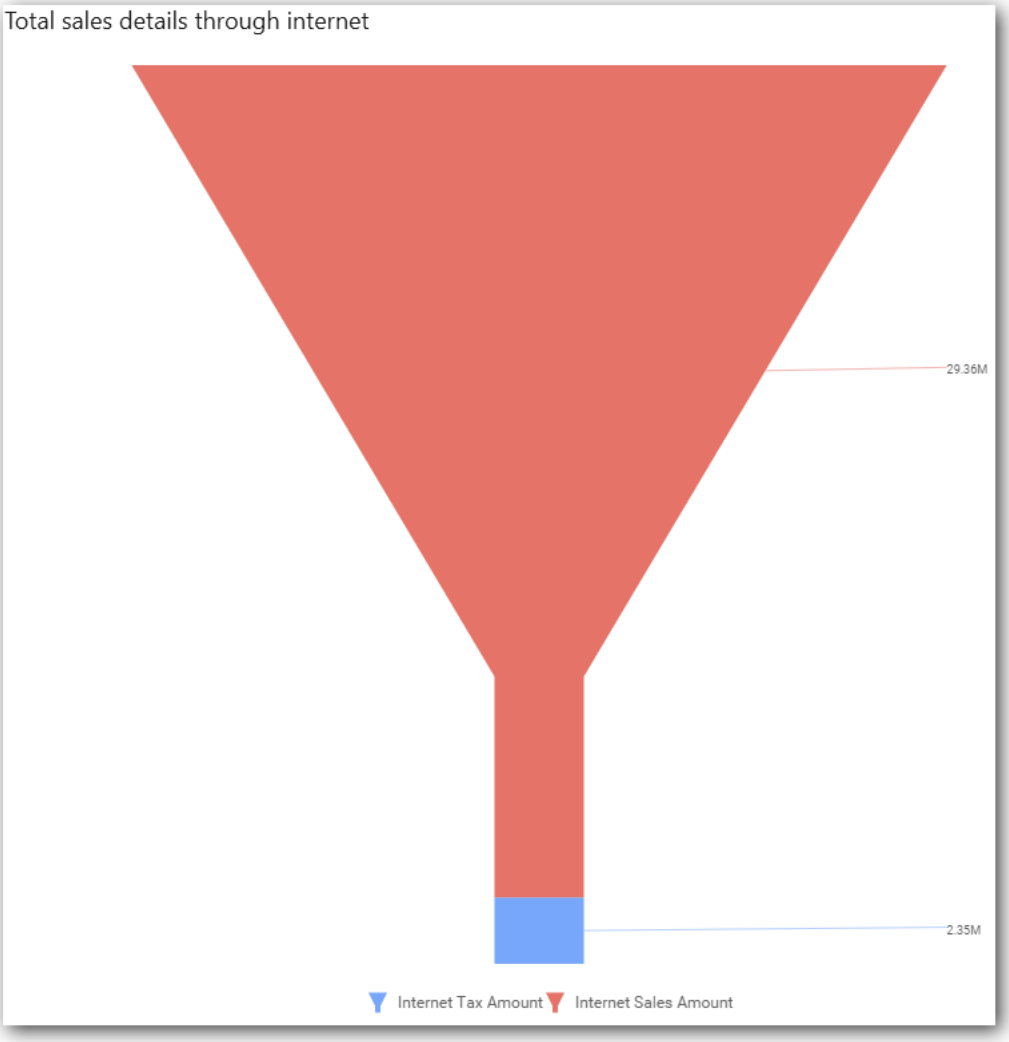
Select **Format** to define the display format to values in the column through the Measure Formatting window.



Choose the options that you need and click **OK**.

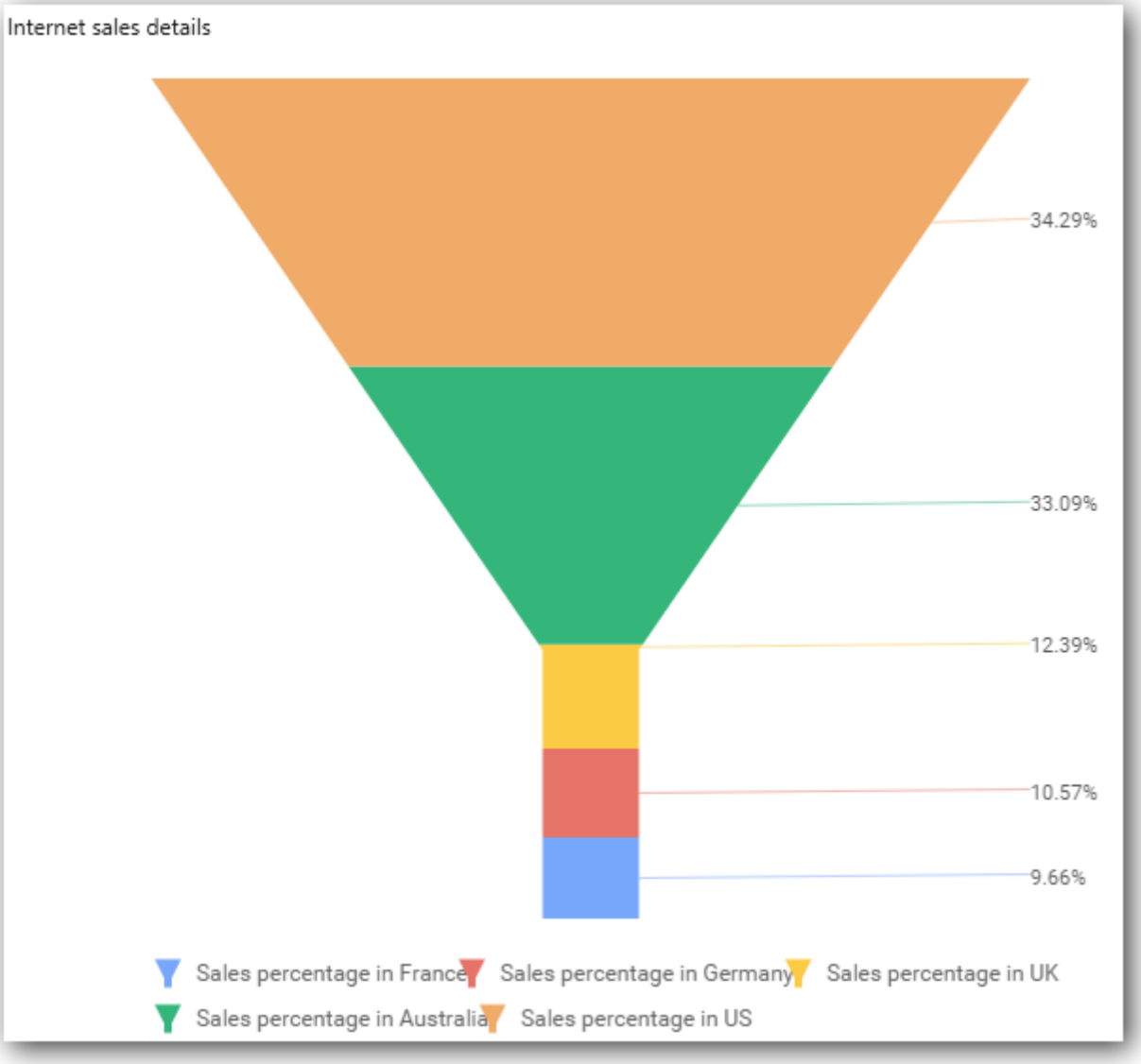


Now, the chart will be rendered as follows.

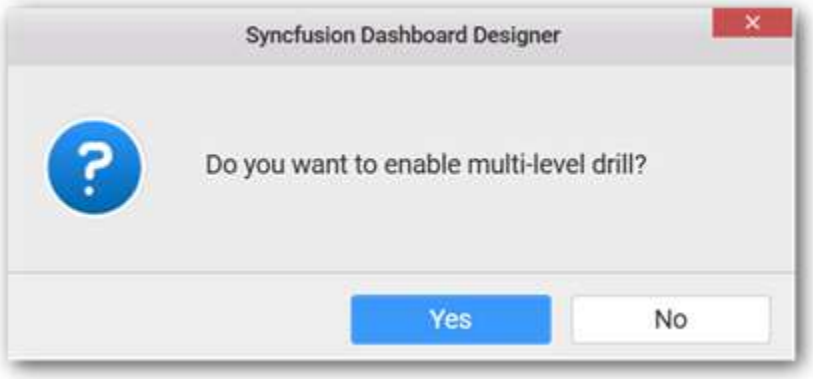


Assigning column(s)

Add a dimension level or hierarchy into Column(s) section by dragging it.

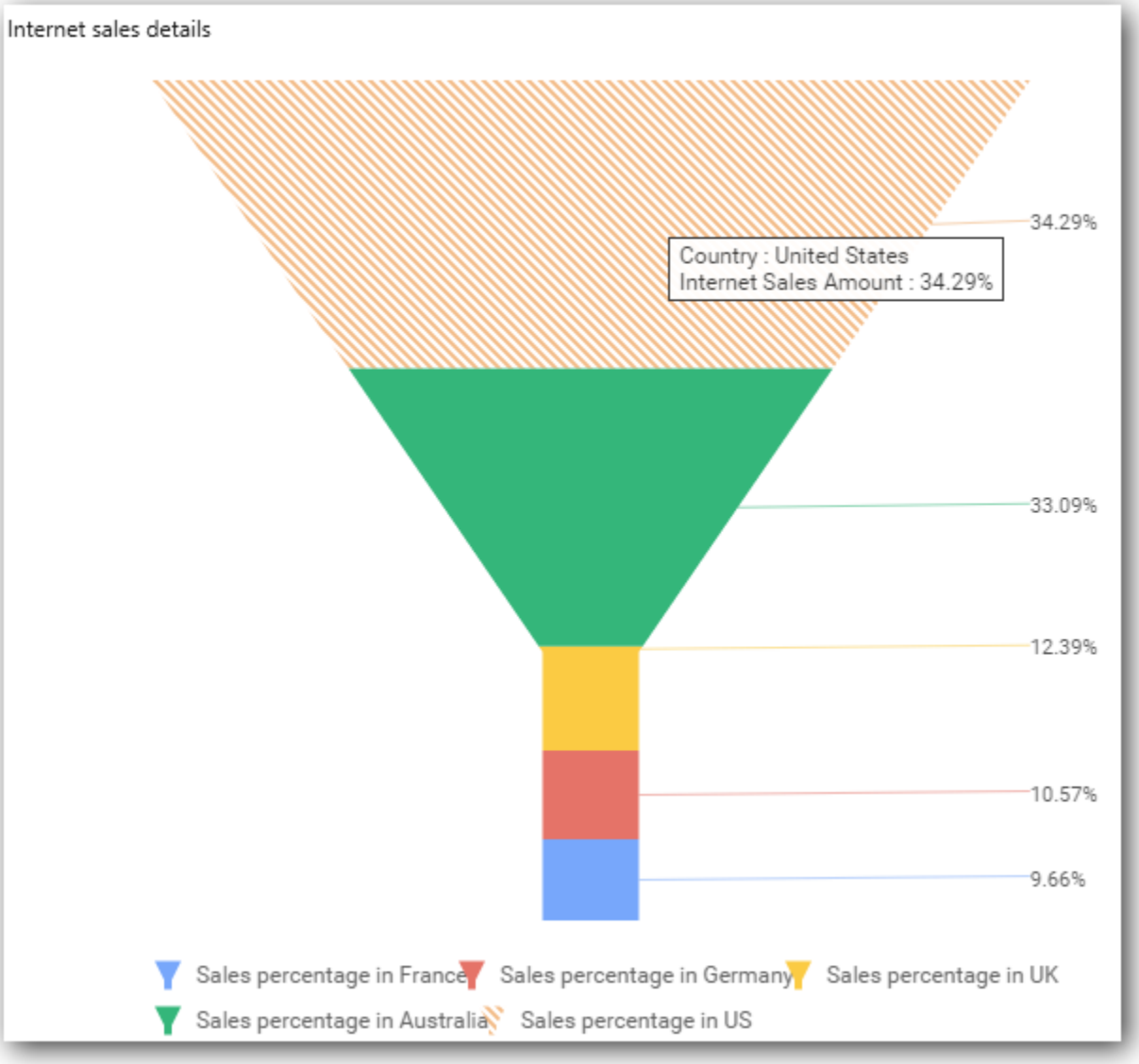


You may also add more than one column into Column(s) section. The following message will be prompted to enable drilling across multiple levels.

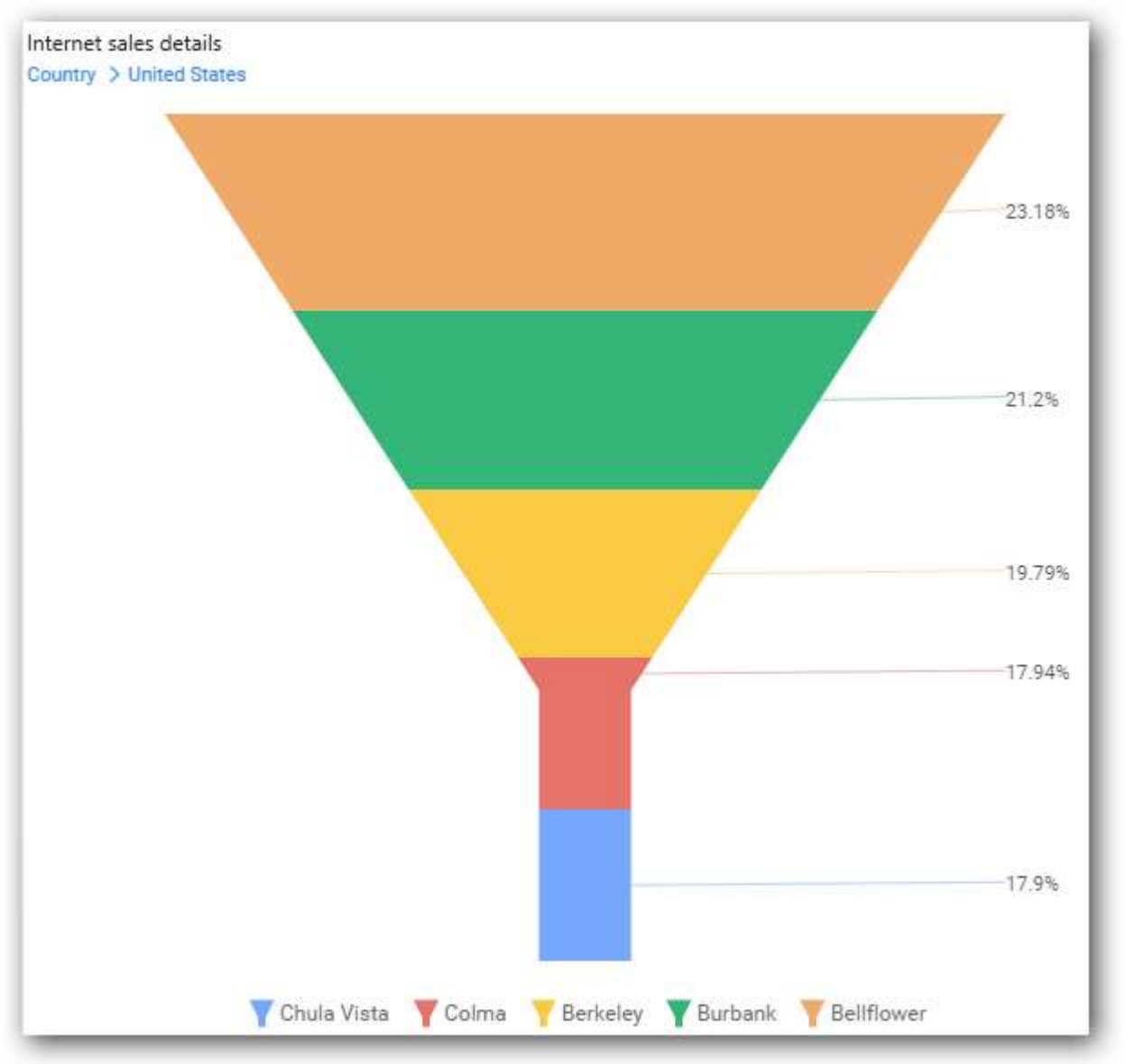


Select **Yes** to enable the drill option in the chart. Select **No** to replace the existing column with this one in the Column(s) section.

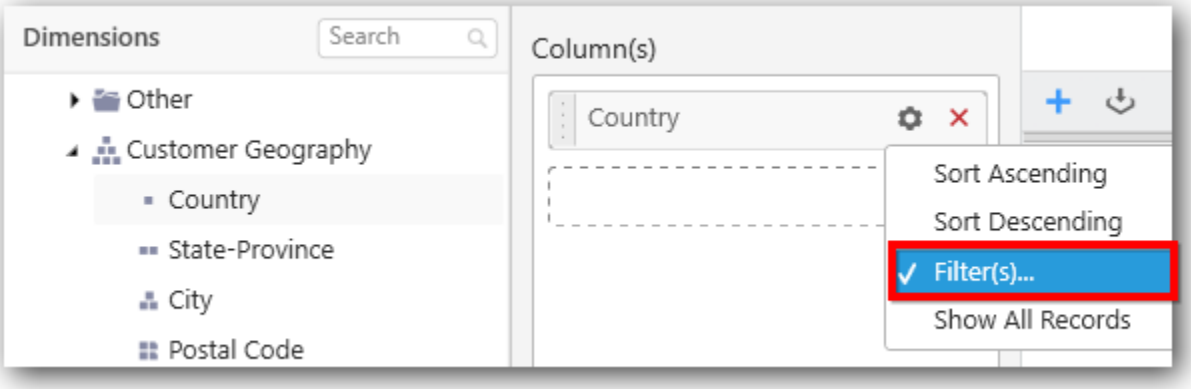
Click the respective data value marker in the chart to drill into its inner level.



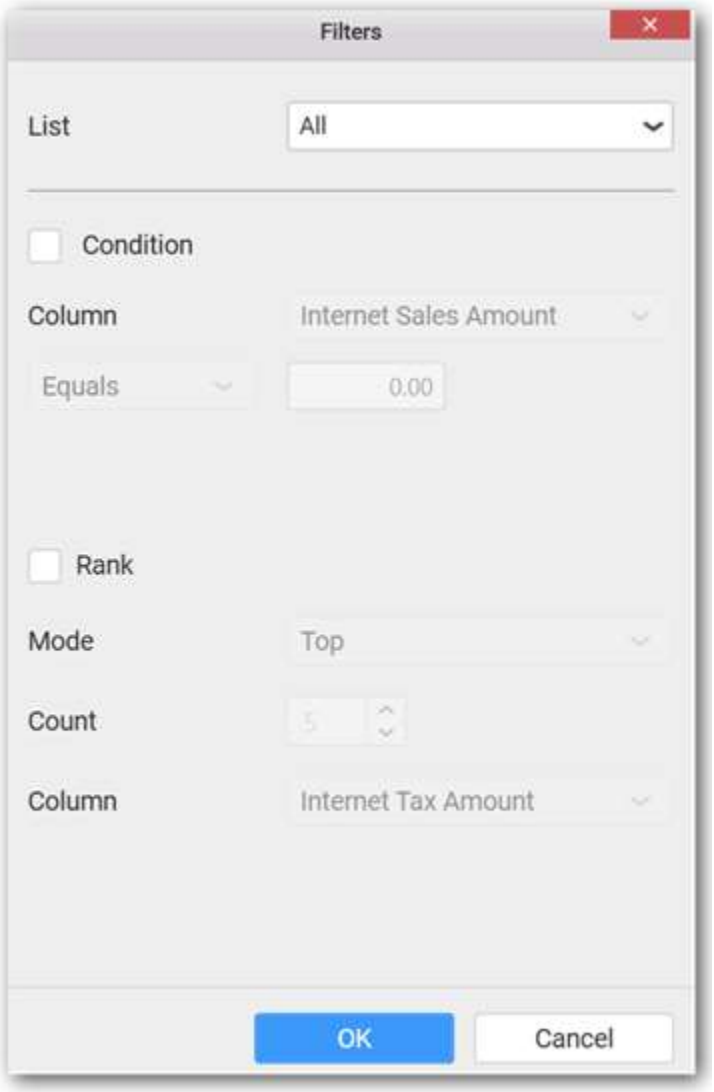
The drilled view of the chart is as follows.



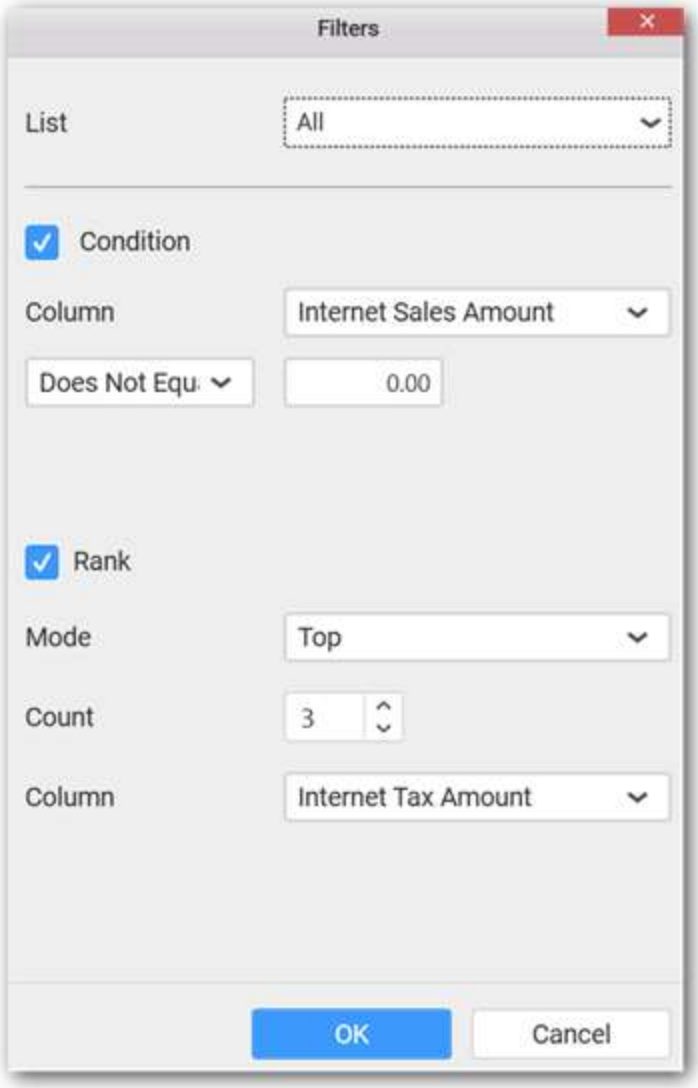
Define the filter criteria using the Filter(s)… menu item in the Settings drop-down menu.



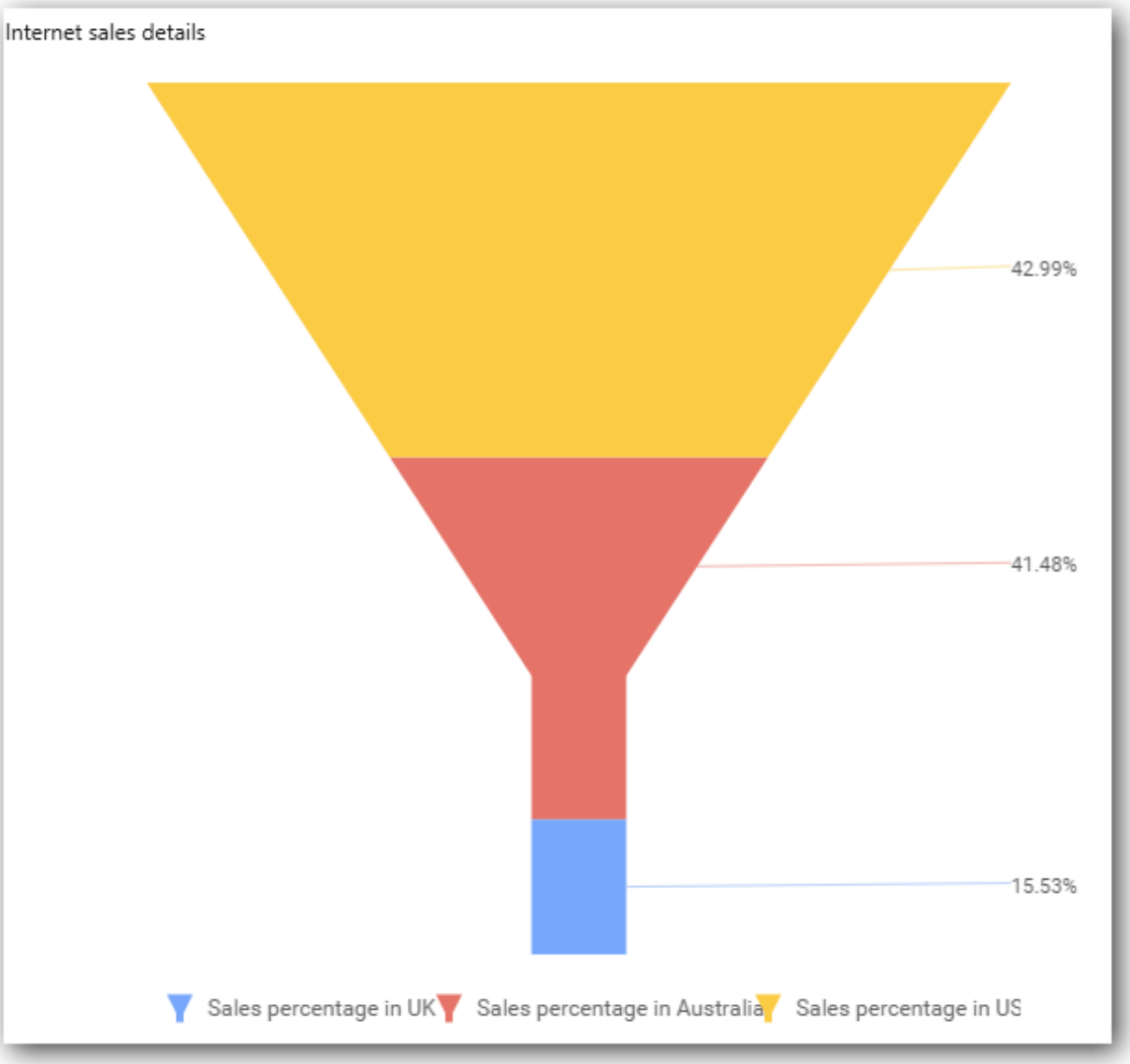
Select Filter(s)… to launch the Filters window.



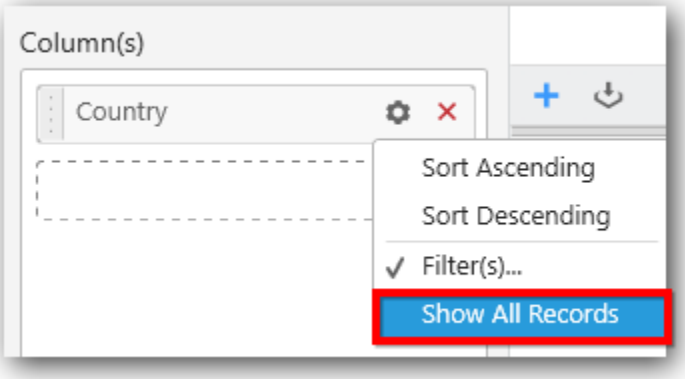
Define the filter condition and Rank and click **OK**.



Now, the chart will be rendered as follows.

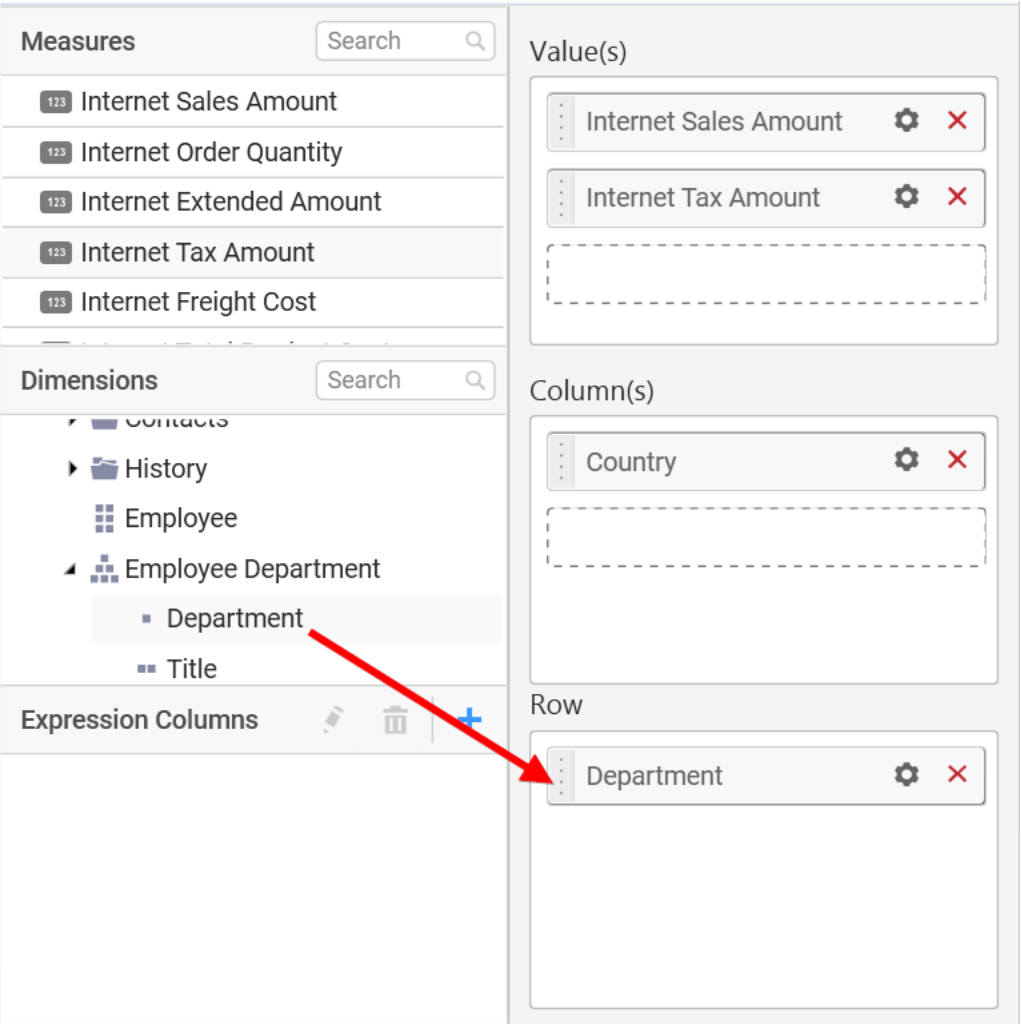


To show all records again, click Show All Records.

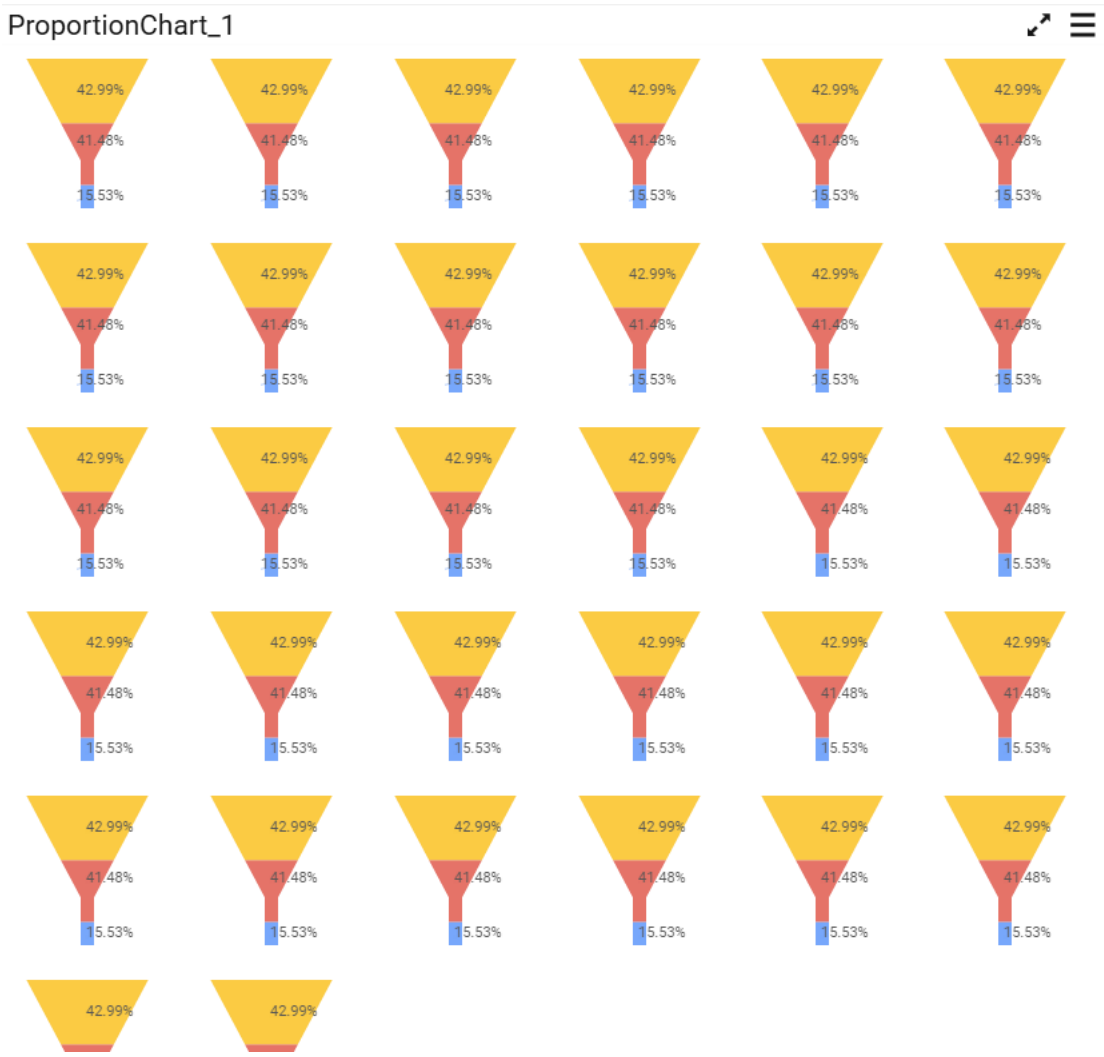


Assigning row

You can add a dimension level or hierarchy to the Row section for series rendering of the chart.



The chart will be rendered in series as shown in the image.



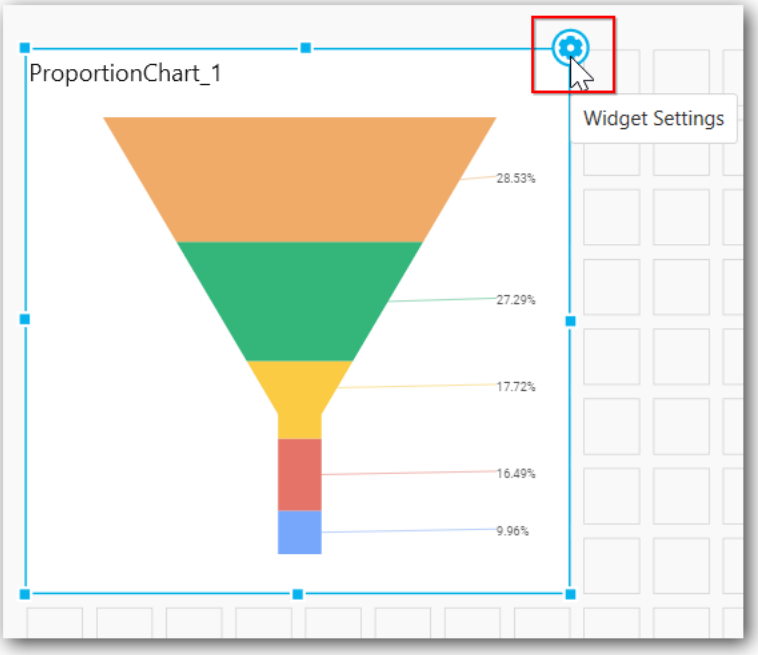
Scroll down to see all charts.

How to format a Funnel Chart

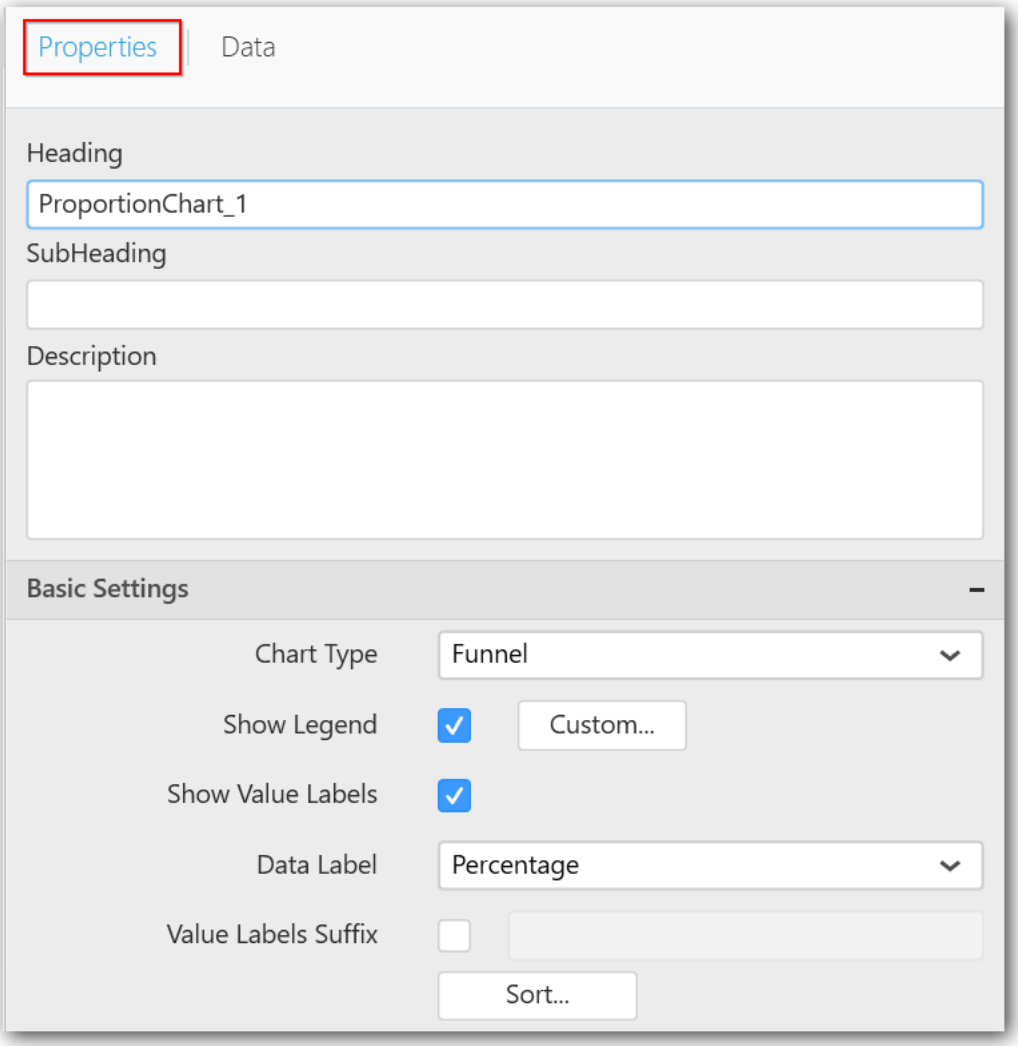
You can format a Funnel Chart for better illustration of the view by using the settings available in the Properties pane.

To configure data into the Funnel Chart, follow the steps:

1. Drag and drop the Funnel Chart into the canvas and resize it to your required size.
2. Configure data into the Funnel Chart.
3. Focus the Funnel Chart and click the Widget Settings icon.

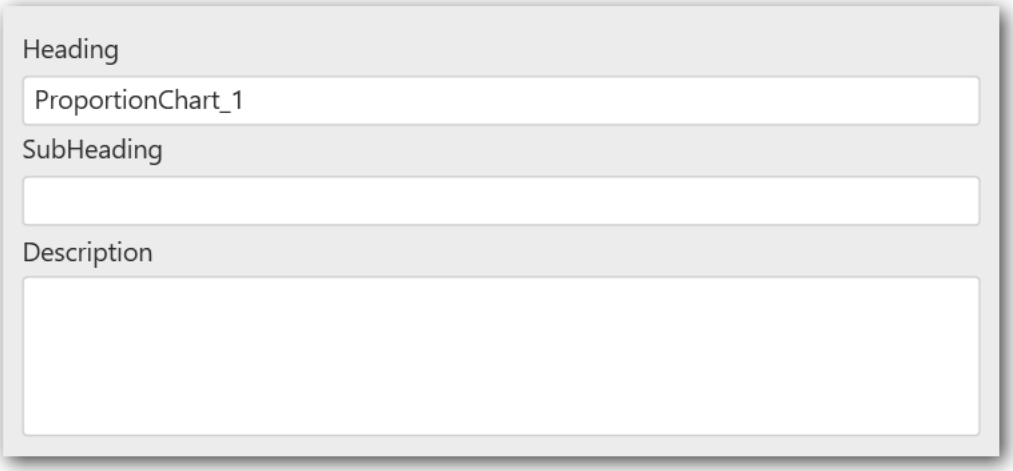


The property window will be opened as follows.



You can see the list of properties available for the widget with default value.

General settings



Header

Allows you to set a title for the Funnel Chart widget.

SubHeading

Allows you to set a subtitle for the Funnel Chart widget.

Description

Allows you to set a description for the Funnel Chart widget, whose visibility will be denoted by i icon, hovering which displays the description in tooltip.

Basic settings

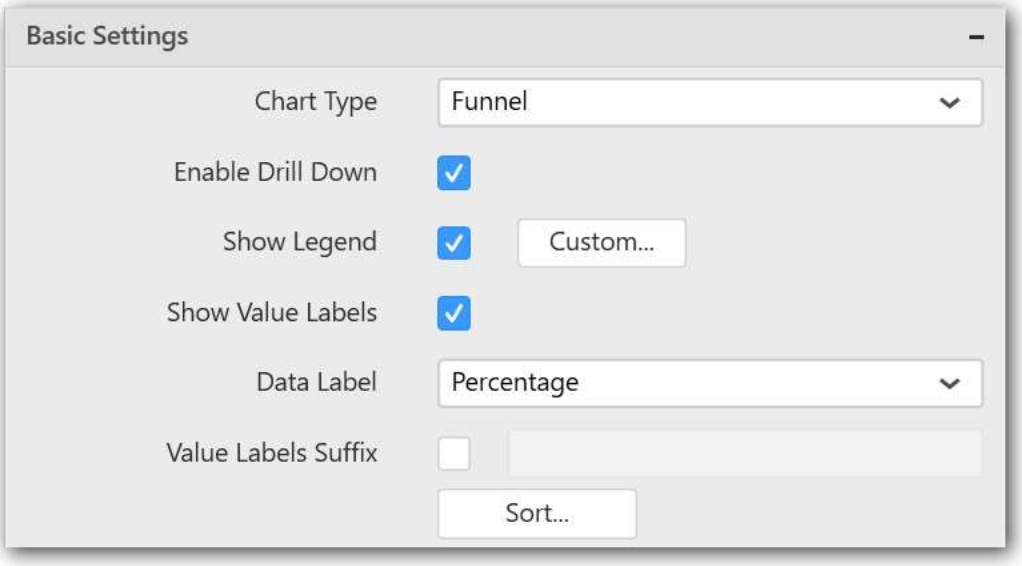


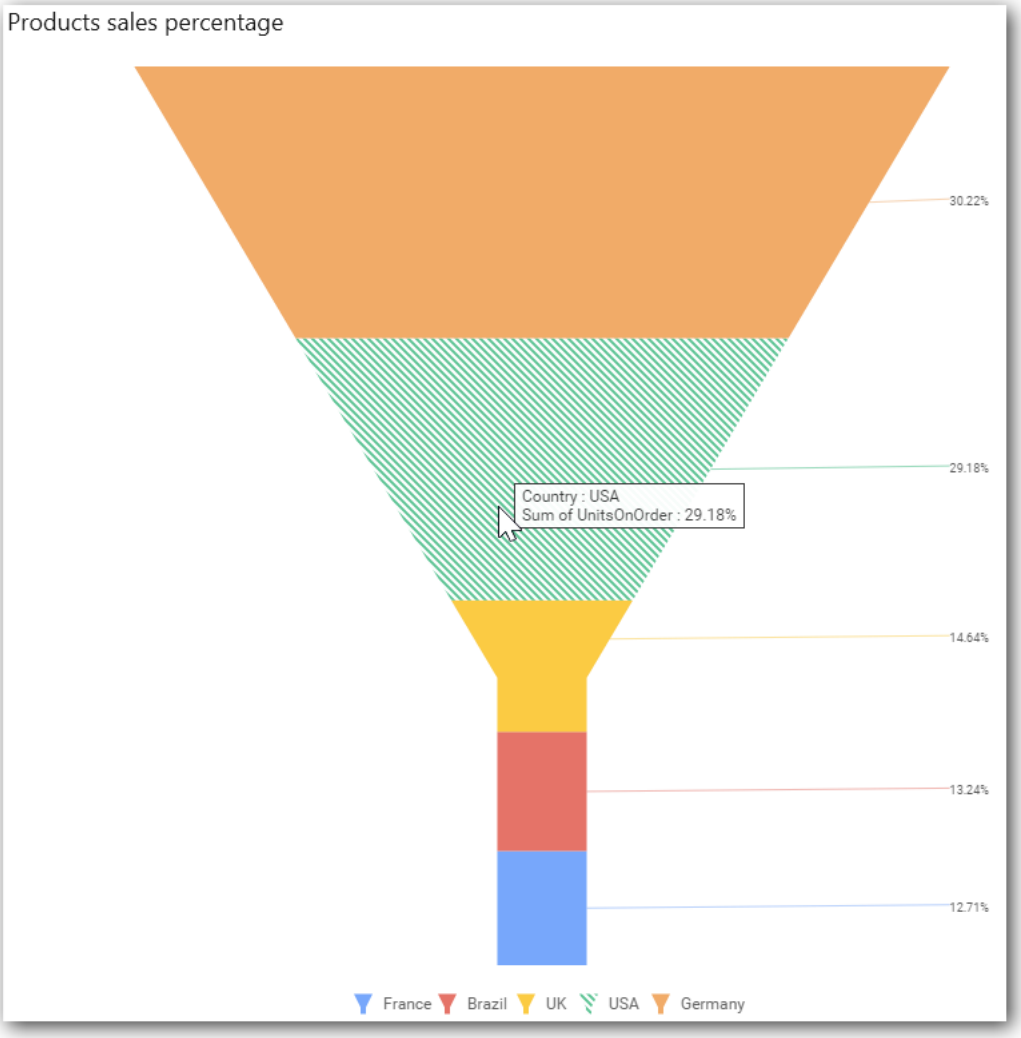
Chart type

Allows you switch the widget view from the current chart type to another chart type.

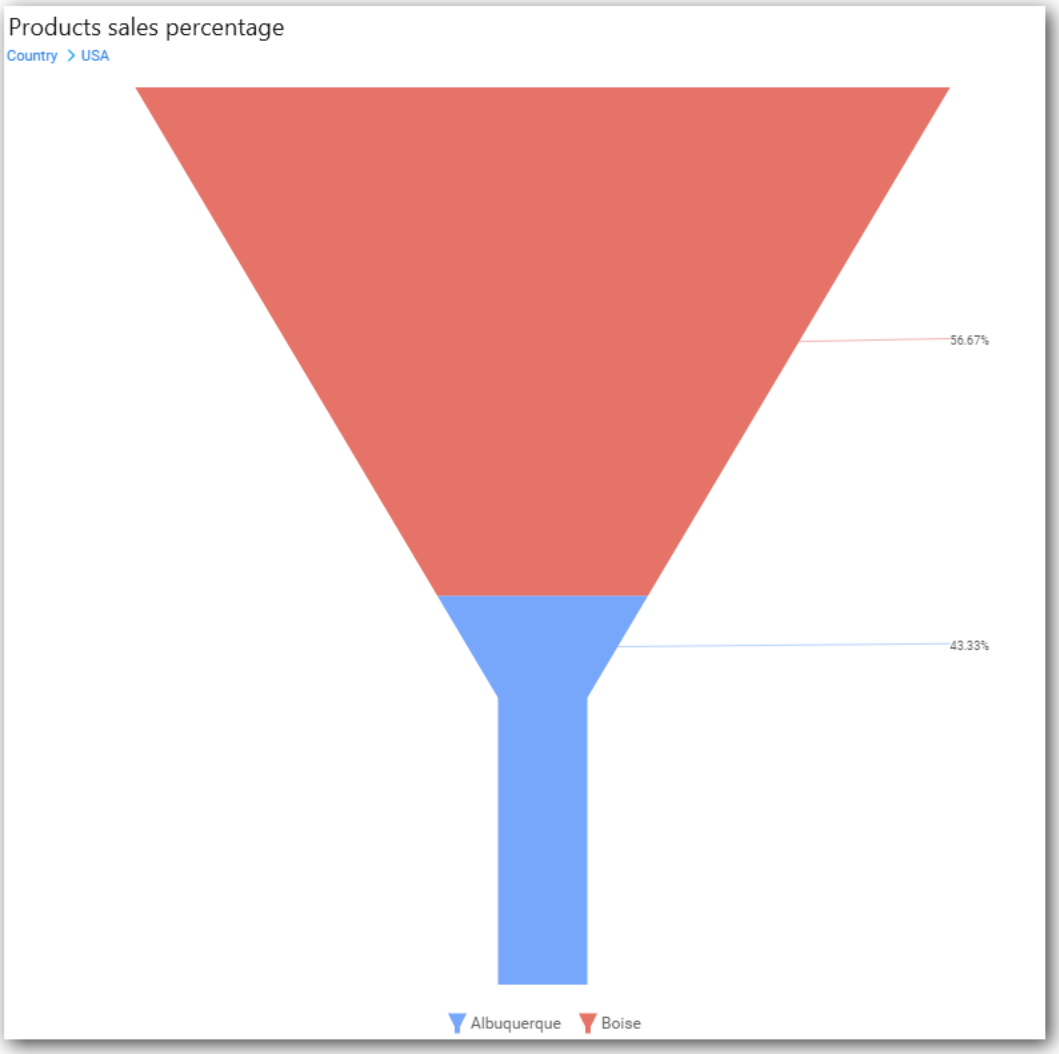
Enable drill down

Allows you add more than one dimension element to the Column block in the Data Pane of Widget View. It forms an hierarchy, and each of its level can be navigated by clicking the respective series drawn. In its disabled state, adding more than one element will replace the existing one.

Initial view

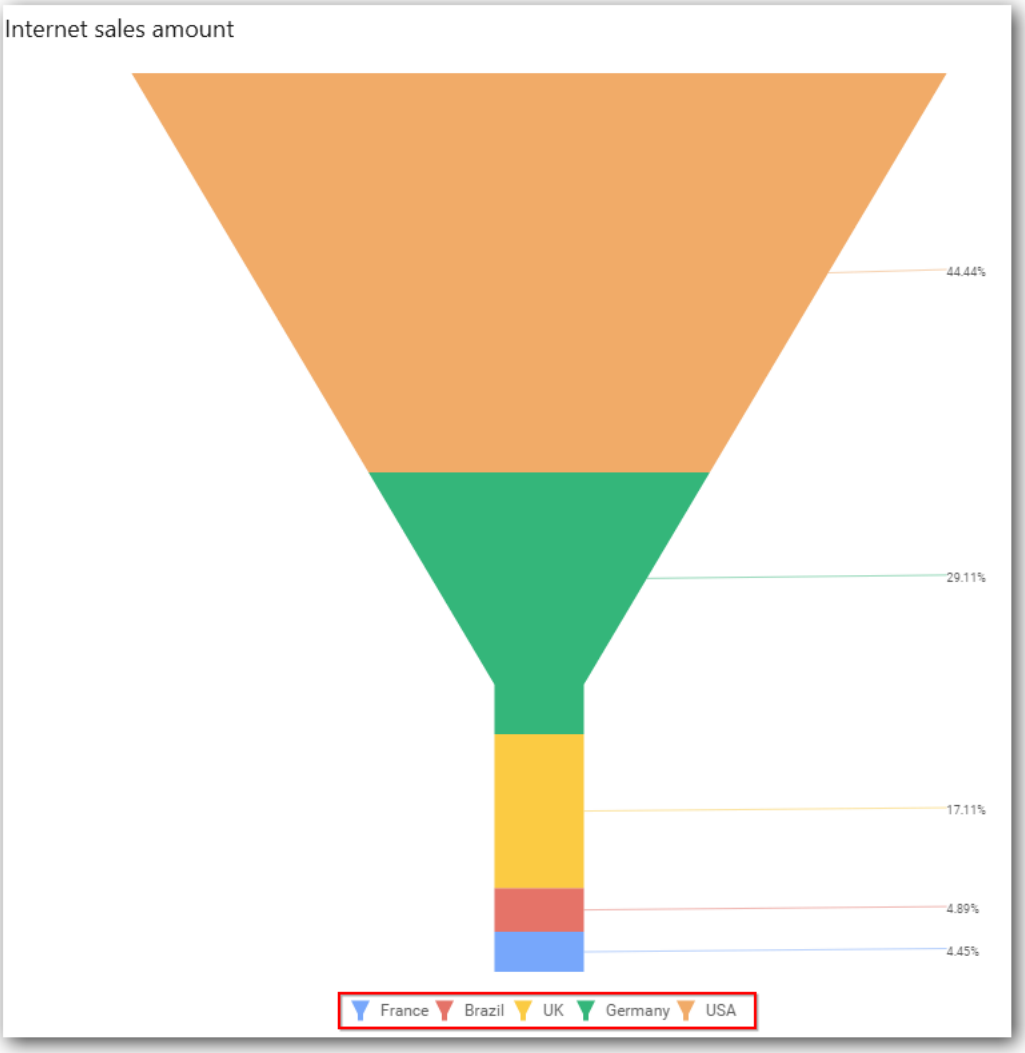


Drilled view



Show legend

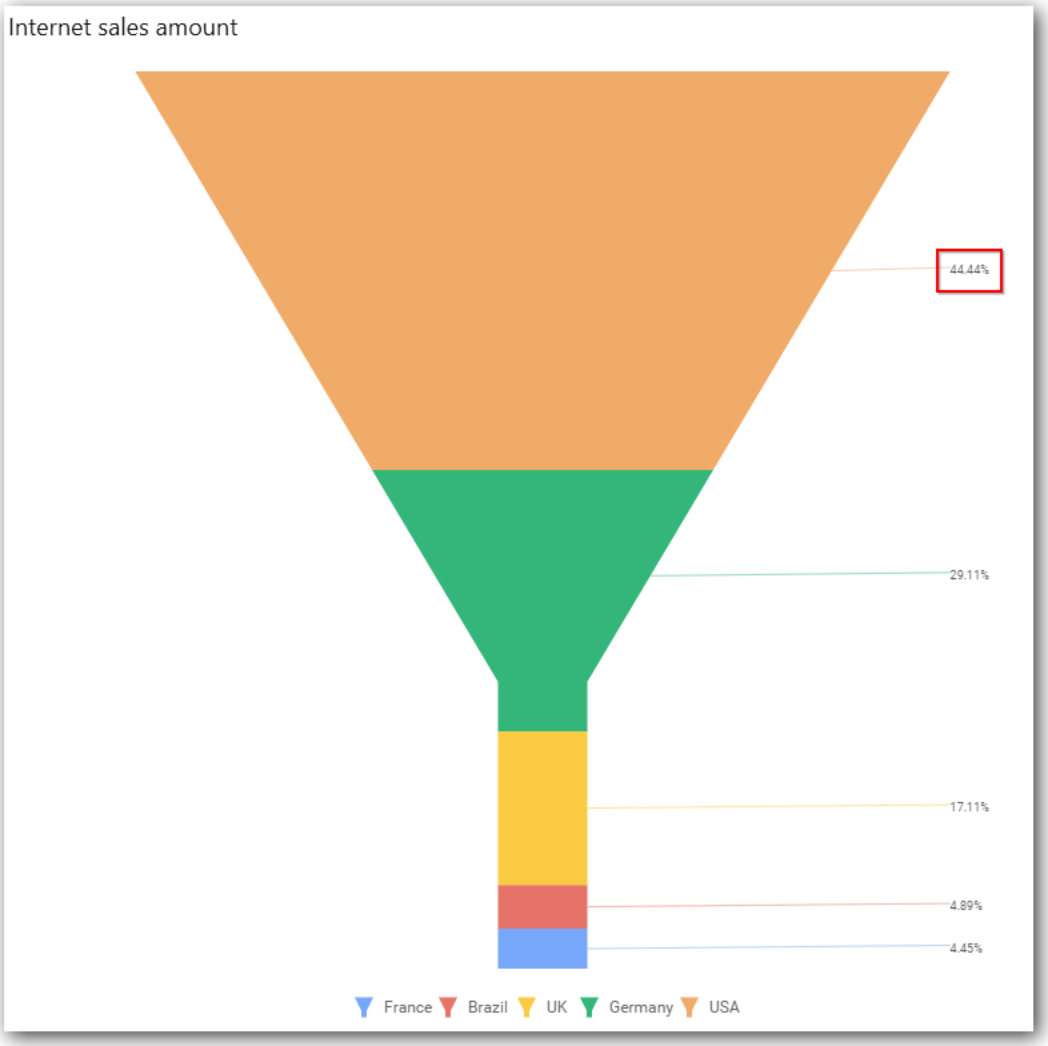
Allows you toggle the visibility of legend in the chart and change the legend text position (selecting through combo box).



Enabling the Custom Legend Text option allows you define a custom text (through the text area) to display for each legend series (selecting through the combo box) in the chart.

Show value labels

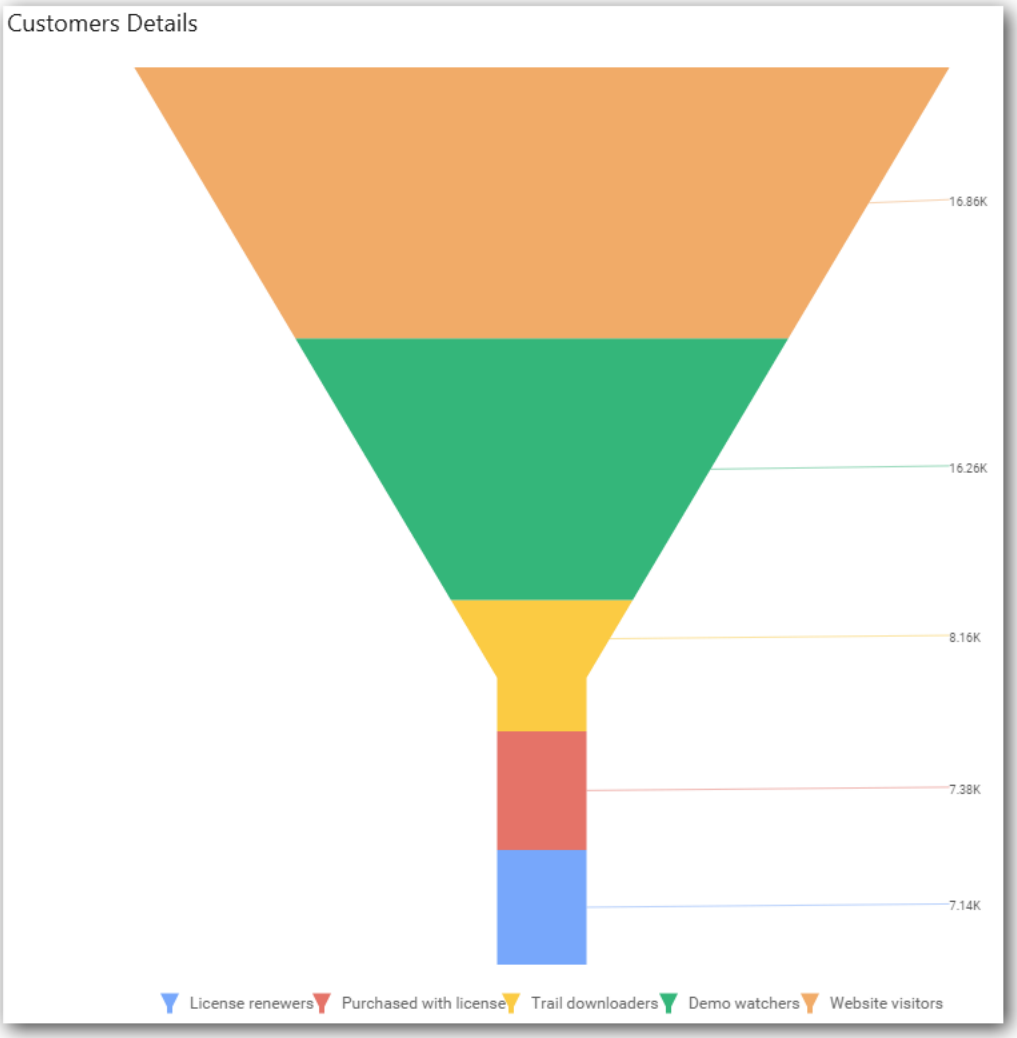
Allows you toggle the visibility of value labels.



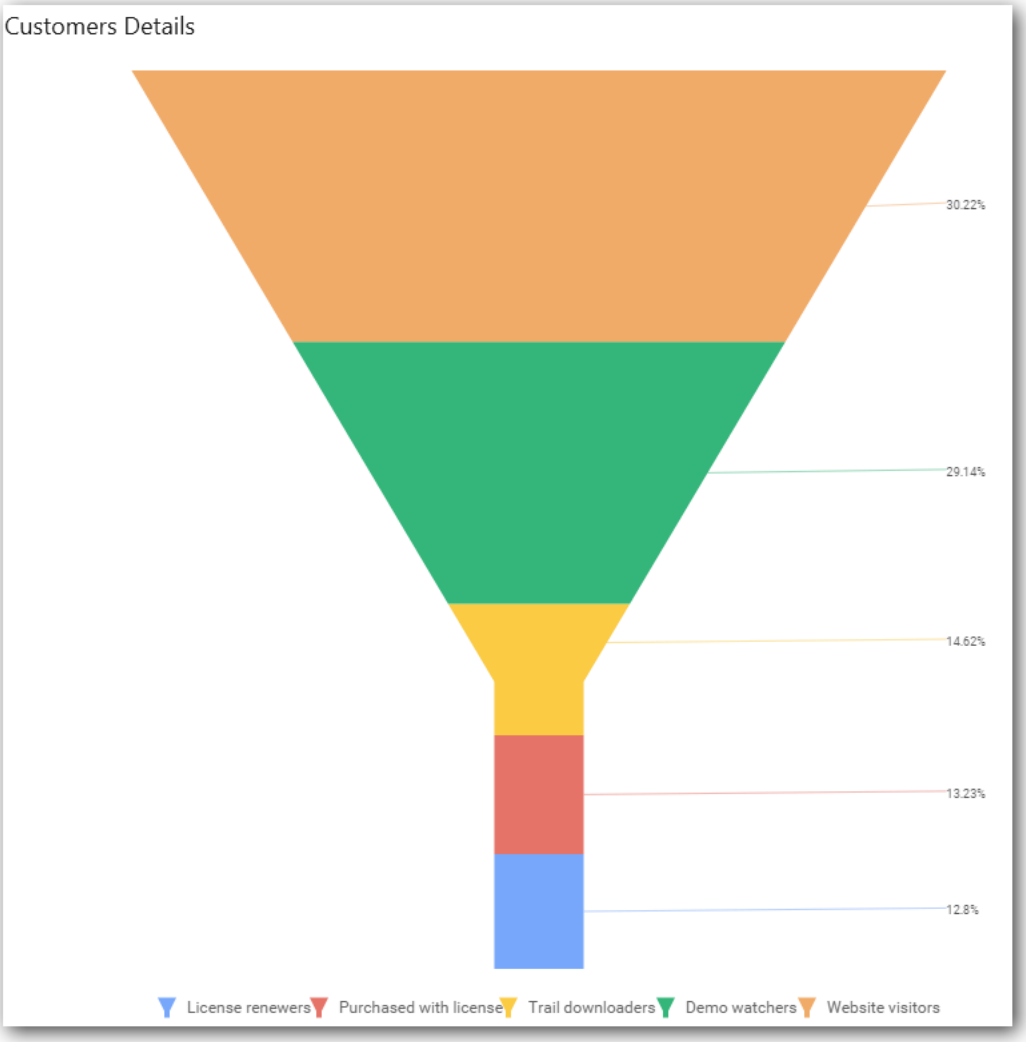
Data label

Allows you define the display format either as value or percentage.

Value

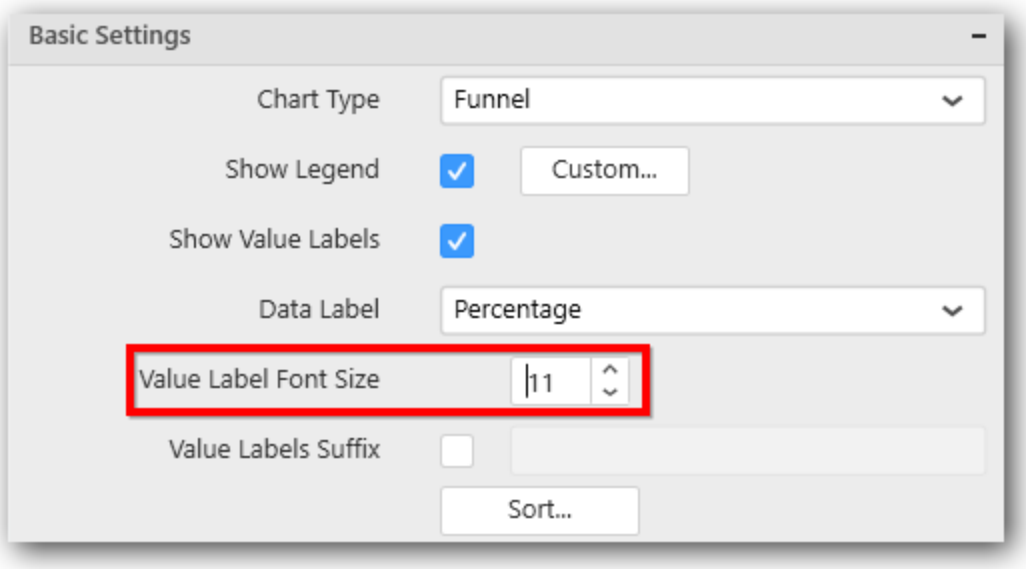


Percentage



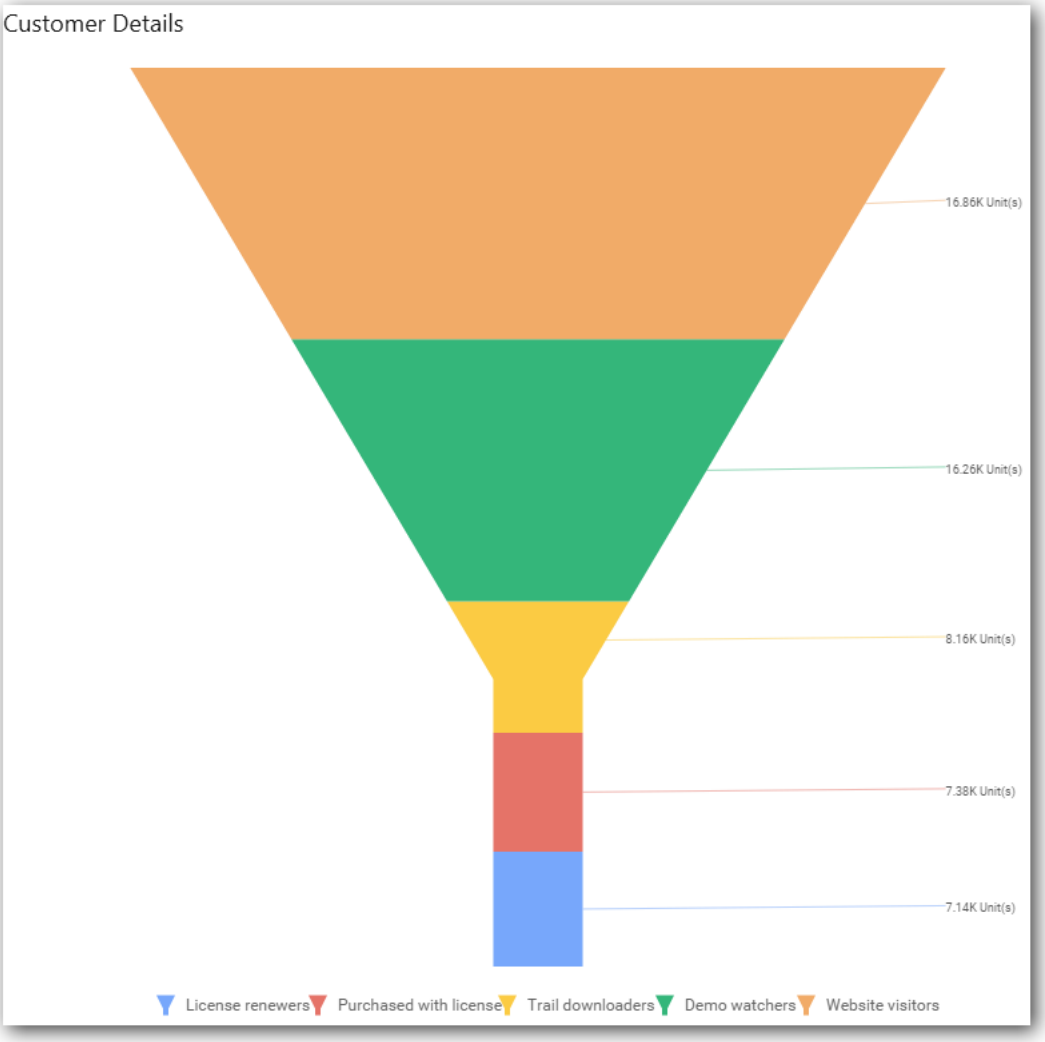
Value label font size

Allows you define the font size for the value labels to display the text. Text will collapse, if it exceeds the size of the bar. The default font size for the value label is 9 pixels.



Value labels suffix

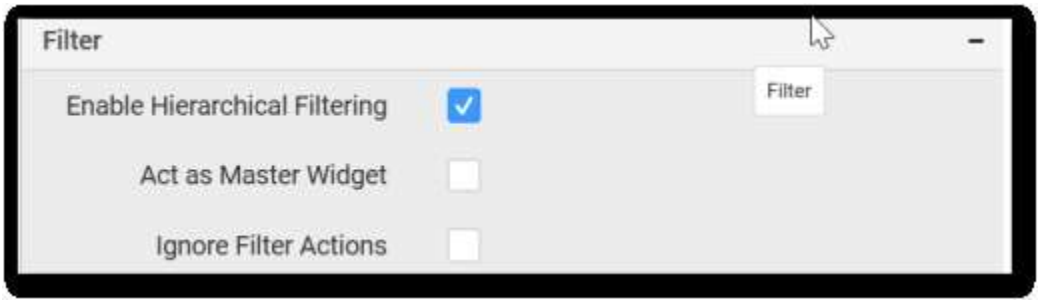
Allows you to set suffix to the value labels.



Sort order

Allows you define the sort order for each added measure column.

Filter settings



Hierarchical filtering

Allows you define the behavior of top n filtering which can be flat or hierarchical.

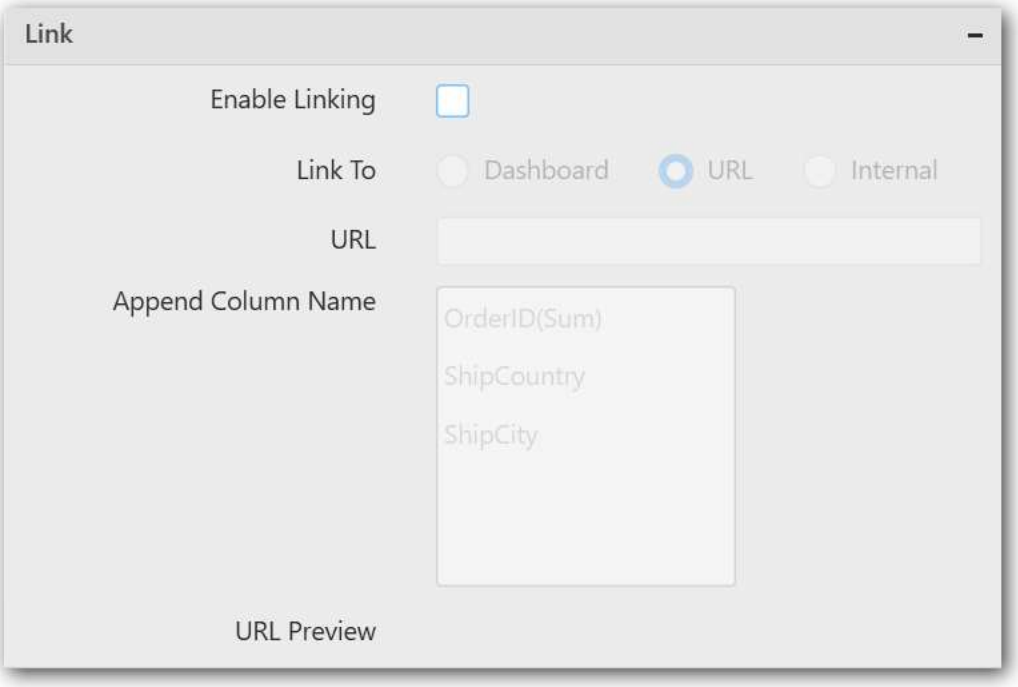
Act as master widget

Allows you define the Funnel Chart widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore filter actions

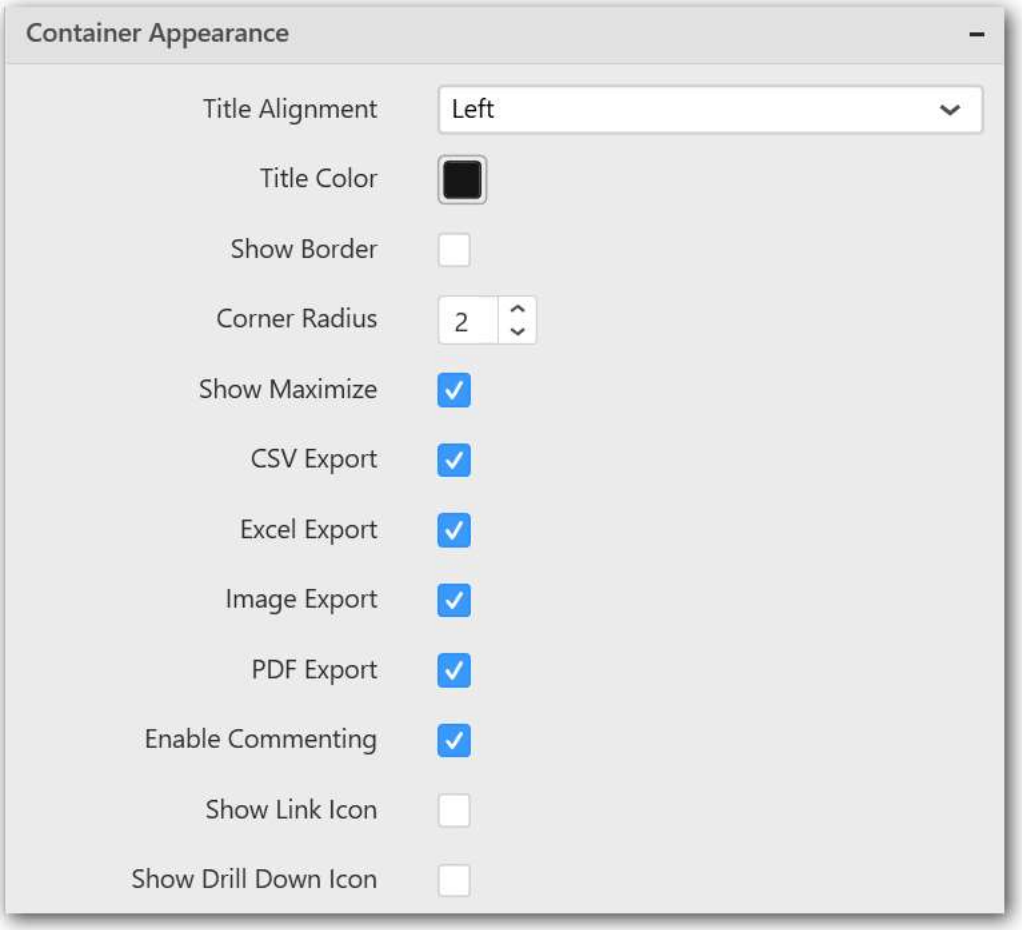
Allows you define the Funnel Chart widget to ignore responding to the filter actions applied on other widgets in the dashboard.

Link settings



Configure the linking to URL or dashboard with the widget through its settings. For more details, refer to Linking.

Container appearance



Title alignment

Allows you handle the alignment of widget title to either left, center, or right.

Title color

Allows you apply the text color to the widget title.

Show border

Allows you toggle the visibility of border surrounding the widget.

Corner radius

Allows you to apply the specified radius to widget corners. Value can be between 0 and 10.

Show maximize

Allows you enable or disable the maximized mode of the Funnel Chart widget. The visibility of the maximize icon in the widget header will be defined based on this setting in viewer.

CSV export

Allows you enable or disable the CSV export option for the Funnel Chart widget. Enabling this allows you to export the summarized data of the widget view to CSV format in viewer.

Excel export

Allows you enable or disable the Excel export option for the Funnel Chart widget. Enabling this allows you to export the summarized data of the widget view to XLSX format in viewer.

Image export

Allows you enable or disable the image export option for the Funnel Chart widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable comment

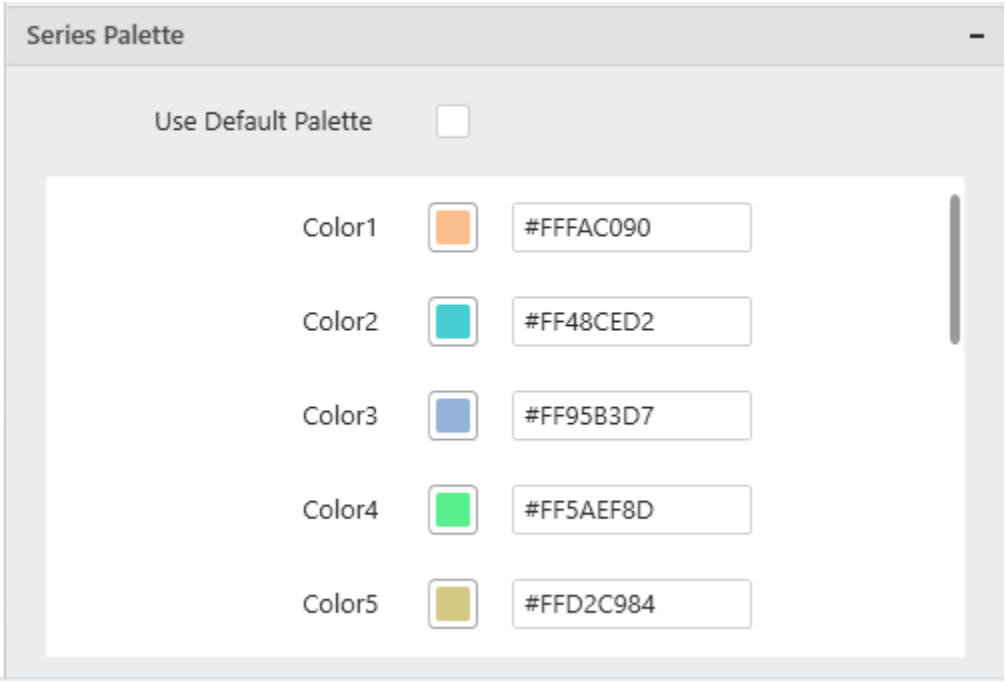
Allows you enable or disable the comment for dashboard widget. For more details refer to here

Series palette

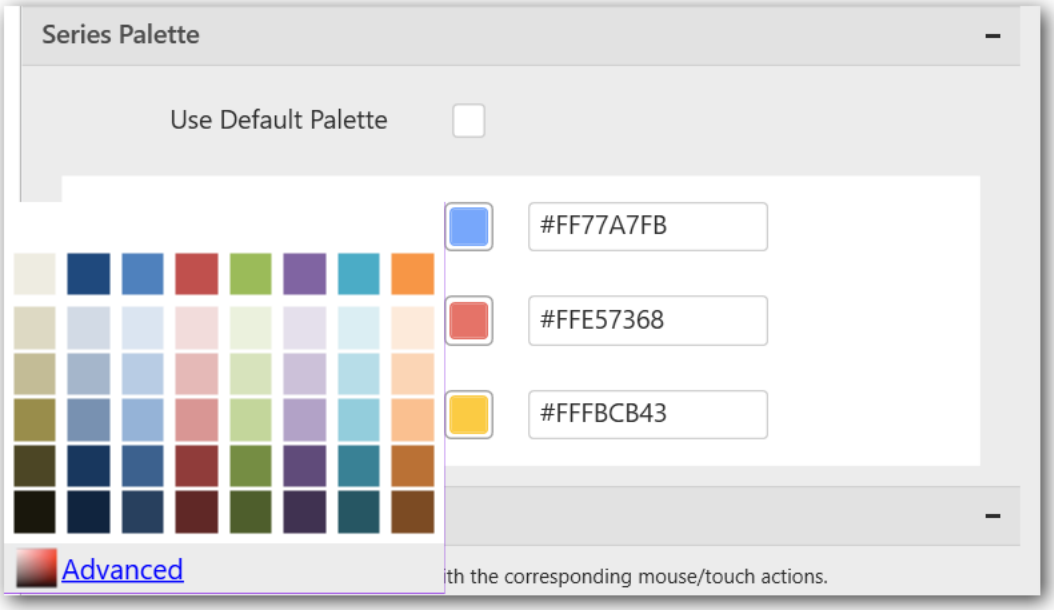
Allows you customize the chart series color through the Series Palette section.

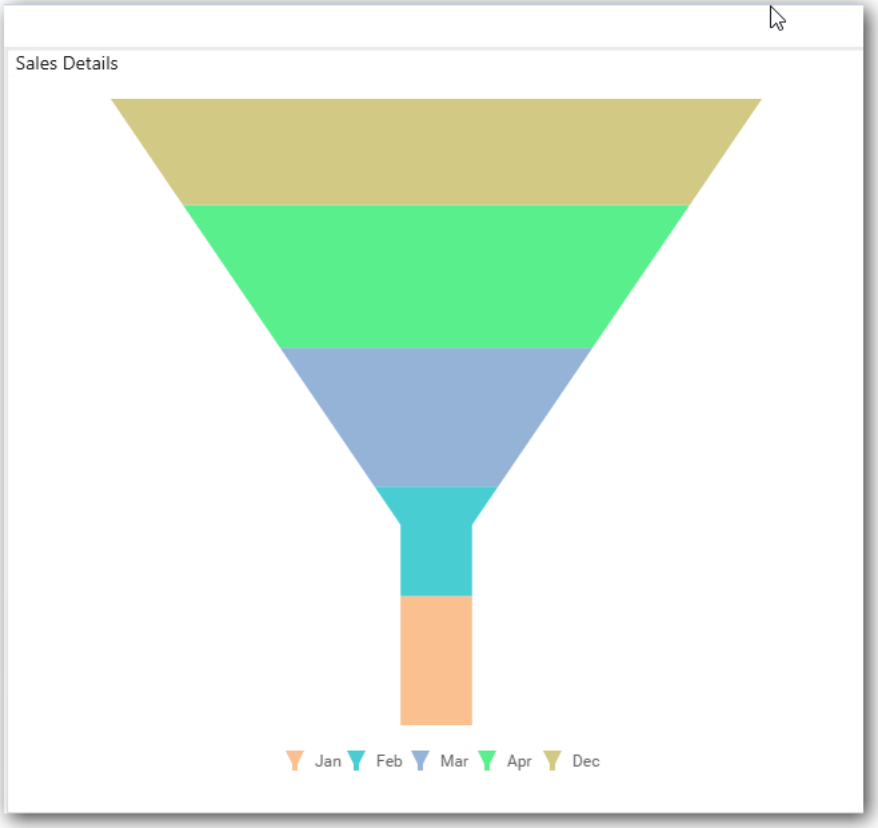
Use Default Palette

Allows you toggle the series color between the default palette and custom palette. By default, the property is toggled on and the default palette will be applied to proportion series segments.



By toggle off the Use Default Palette, you can customize the proportion series segments' colors. This section shows a palette of colors. By clicking the colored square, the color picker will be opened. You can choose a color. And, you can change the series color by changing the corresponding hexadecimal value in the right side.





Grid

Grid allows you to showcase ranking relationship through vertical arrangement of items, ordered from top to bottom.

Sales in City

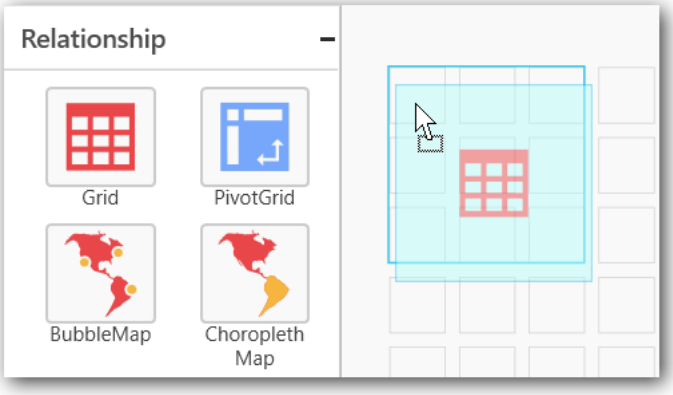
City	Total Sales	Profit
Aachen	3,763.21	+3.17K
Albuquerque	52,234.42	+45.60K
Anchorage	16,313.67	+13.76K

How to configure flat table data to Grid?

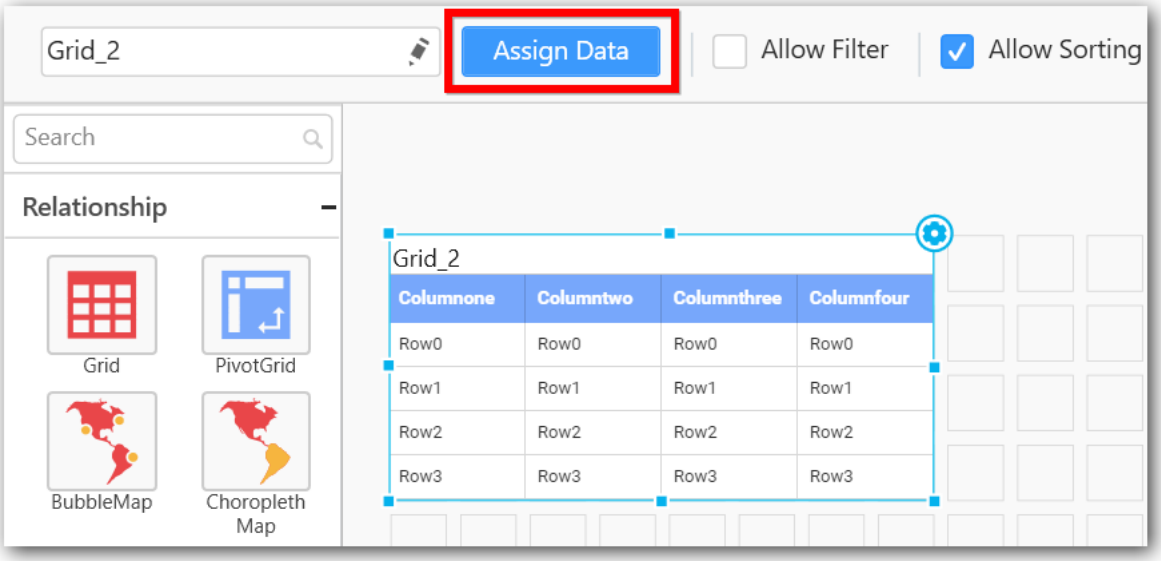
To construct a grid, a minimum requirement of 1 column is needed. You can visualize both measure, calculated measure and dimension column data in grid control. You can also add a column that is hidden from the view by adding the column in the hidden columns section. The data of these columns will be hidden from the view but can be used for filtering other widgets in the dashboard.

The following procedure illustrates data configuration of Grid.

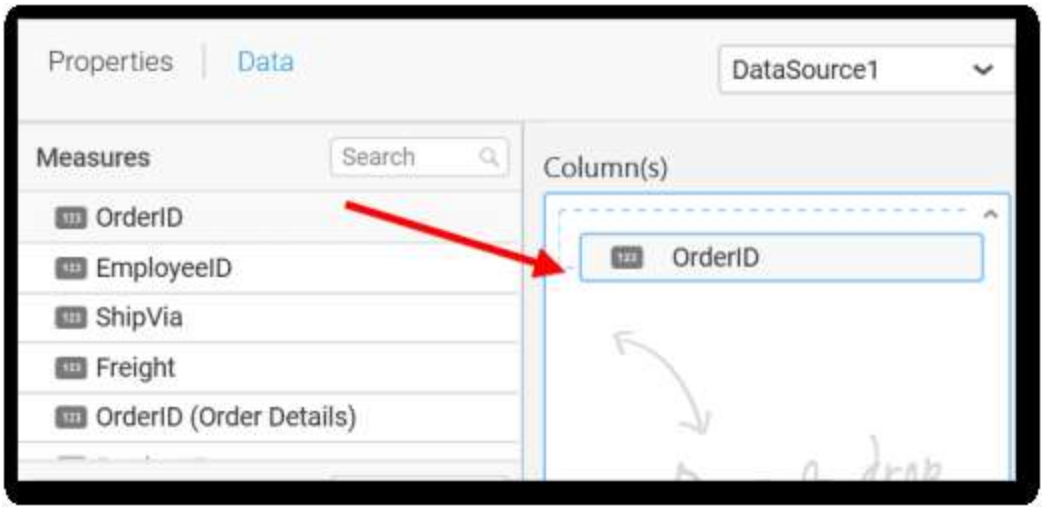
Drag and drop Grid control icon from the Tool box into design panel. You can find control in Toolbox by search.



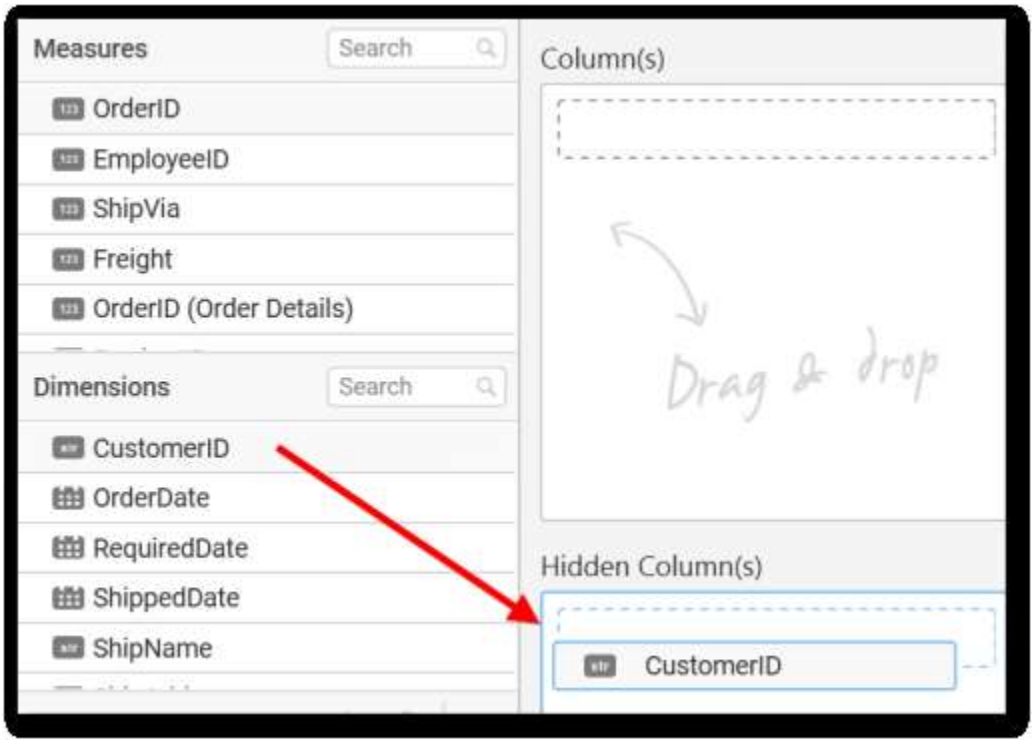
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



Bind column through drag and drop element from Measures section to Column(s) section.

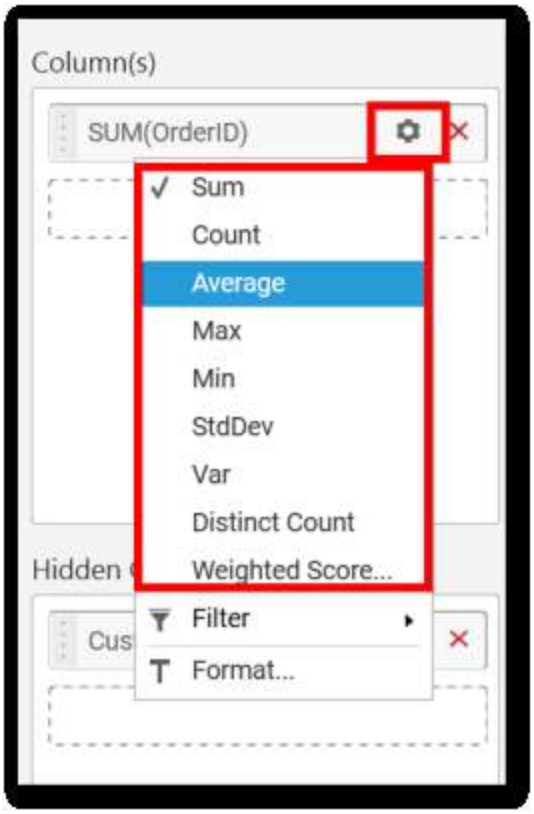


You can drag and drop the elements to Hidden Columns if required. Based on the Hidden columns elements the values will be shown in grid widget.

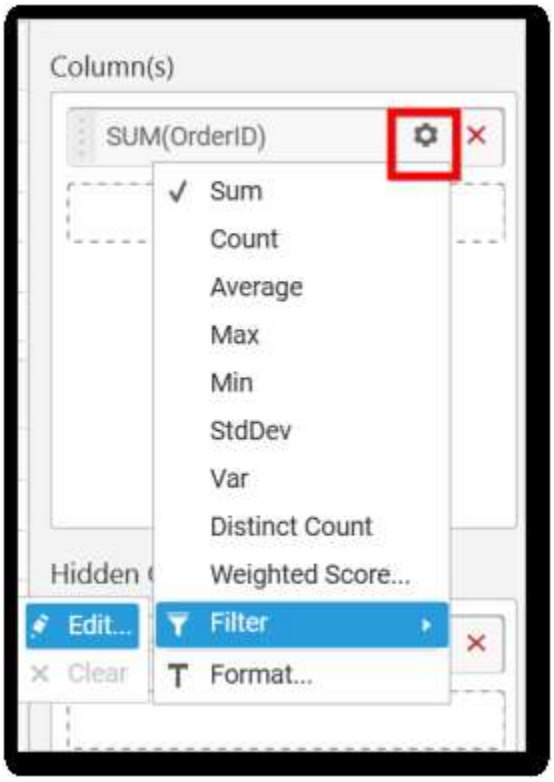


You can use aggregate function to change the column values by clicking the settings.

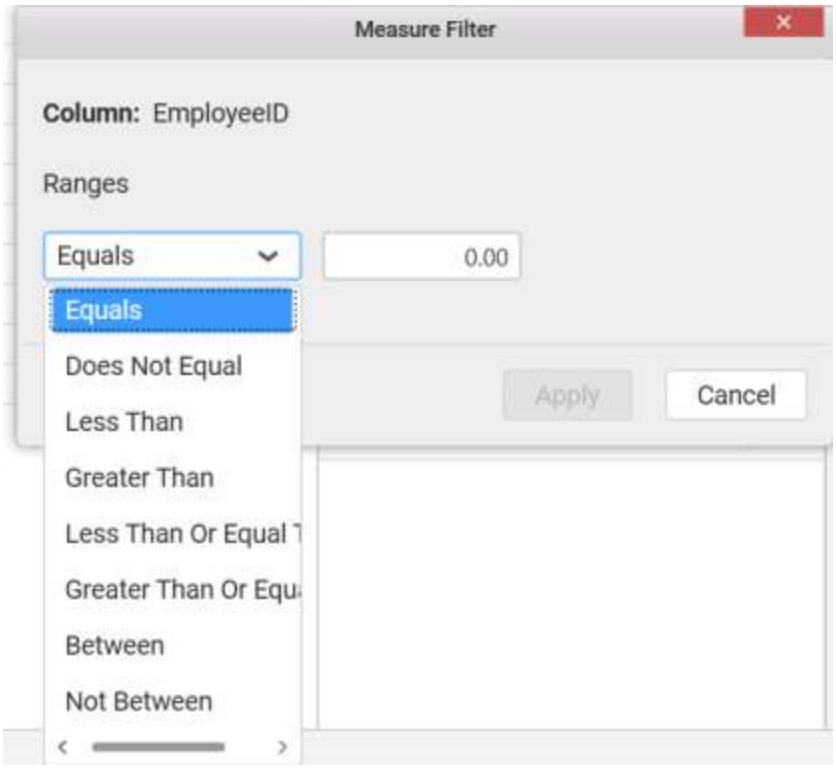
Note: "None" option allows you to view data without aggregation but grouped. ODBC ANSI SQL connection in Live Mode does not support this option.



You can use Filters to change the values by selecting the Edit option.



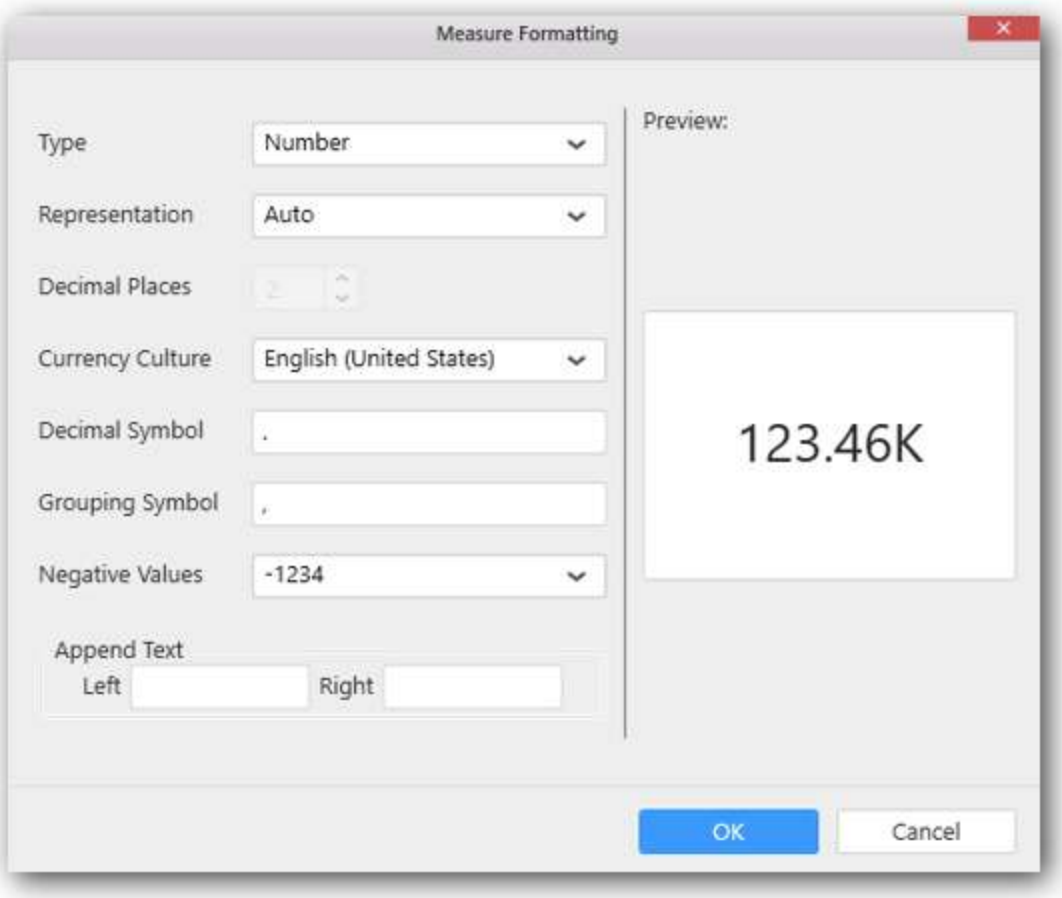
You can get the Measure Filter window and select the filter condition.



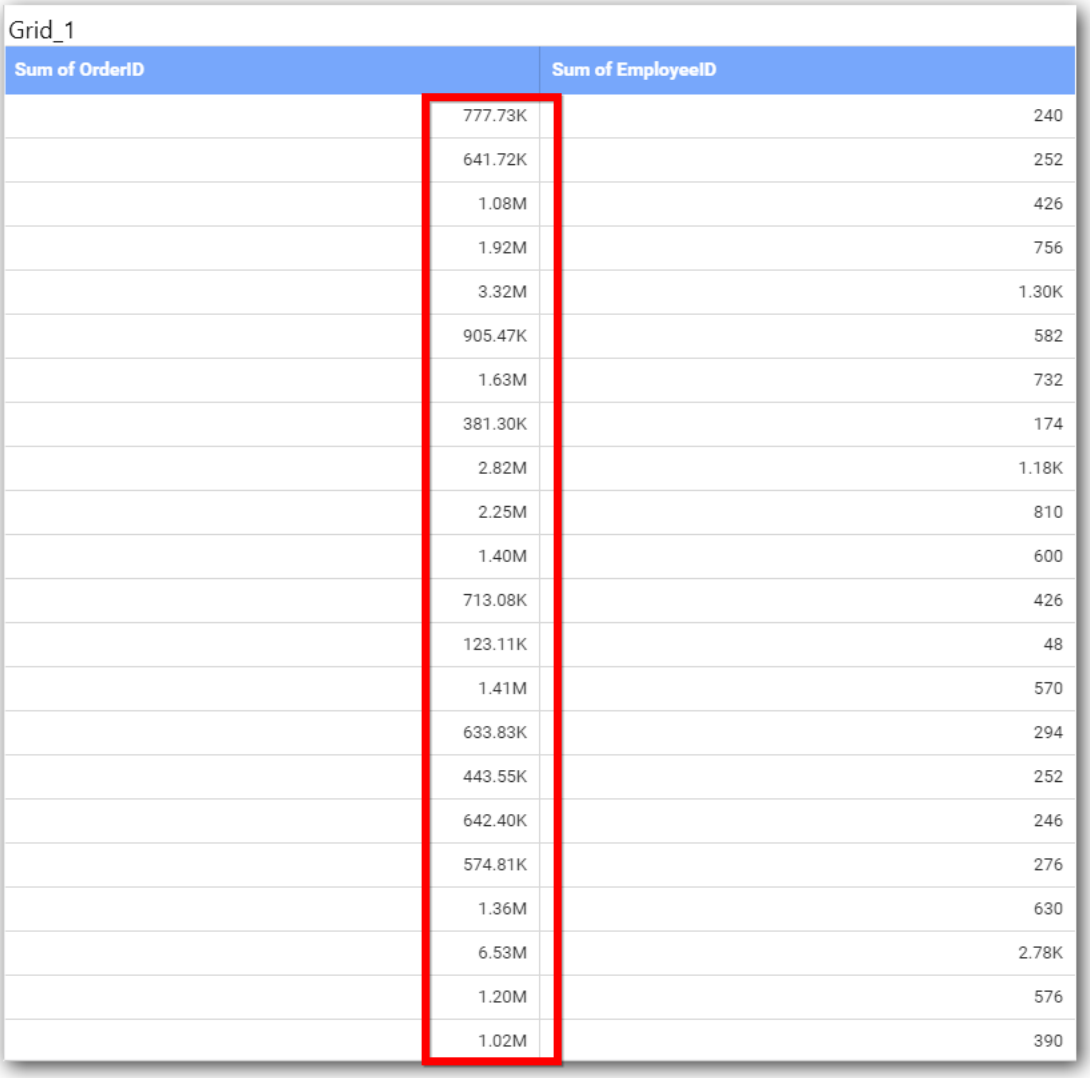
You can clear the filter by clicking the Clear option.

You can format the elements by selecting the Format option.

Measure Formatting window will be shown to change the measure.



Here is an illustration,

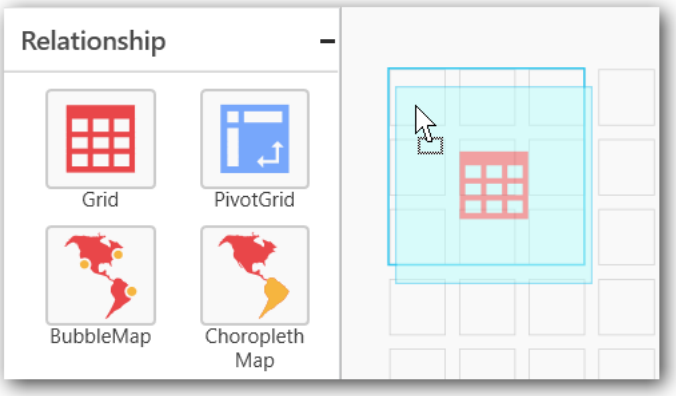


How to configure the SSAS data to Grid?

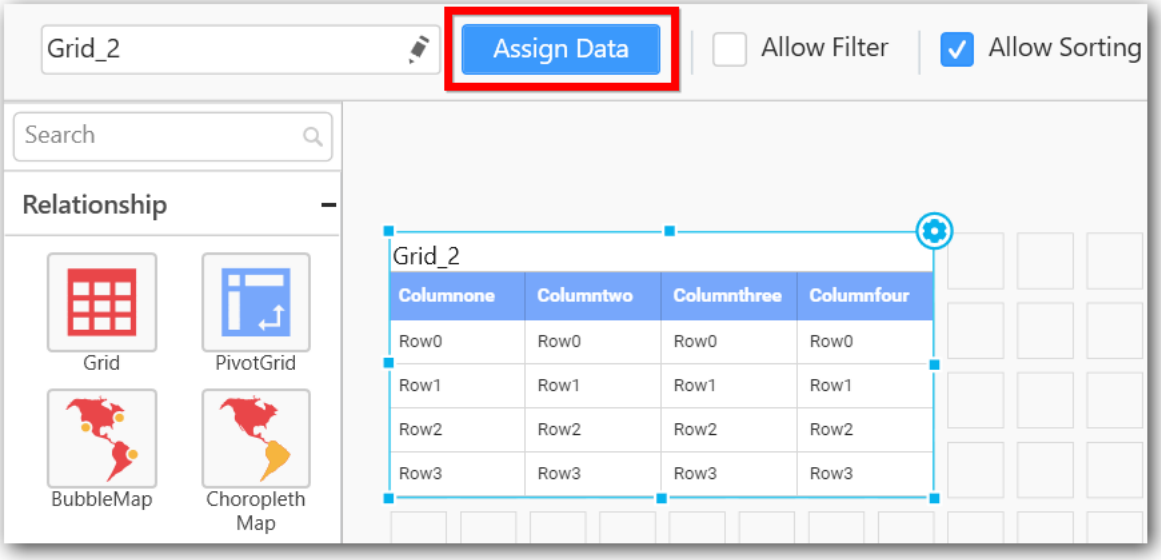
To construct a grid, a minimum requirement of 1 column is needed. You can visualize both measure, calculated measure and dimension column data in grid control. You can also add a column that is hidden from the view by adding the column in the hidden columns section. The data of these columns will be hidden from the view but can be used for filtering other widgets in the dashboard.

Following steps illustrates configuration of SSAS data to Grid.

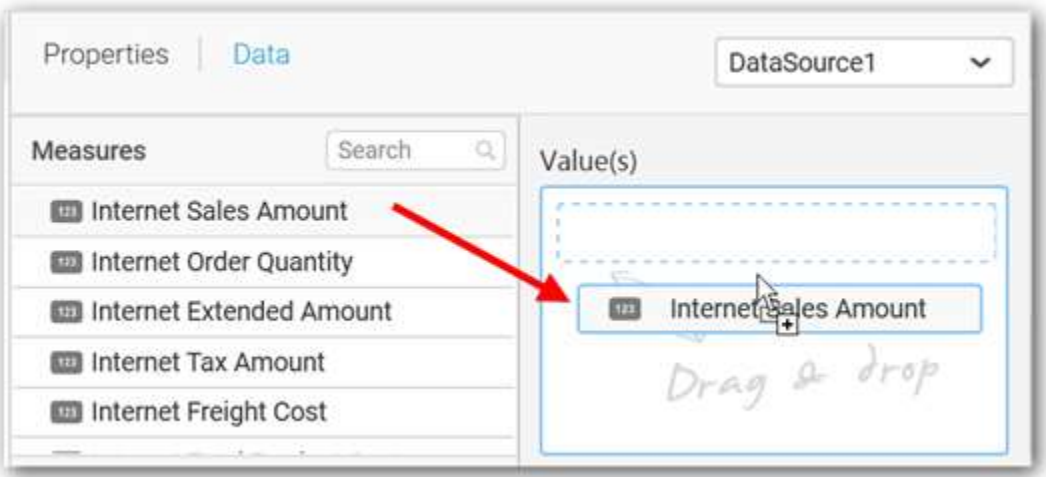
Drag and drop Grid control icon from the Tool box into design panel. You can find control in Toolbox by search.



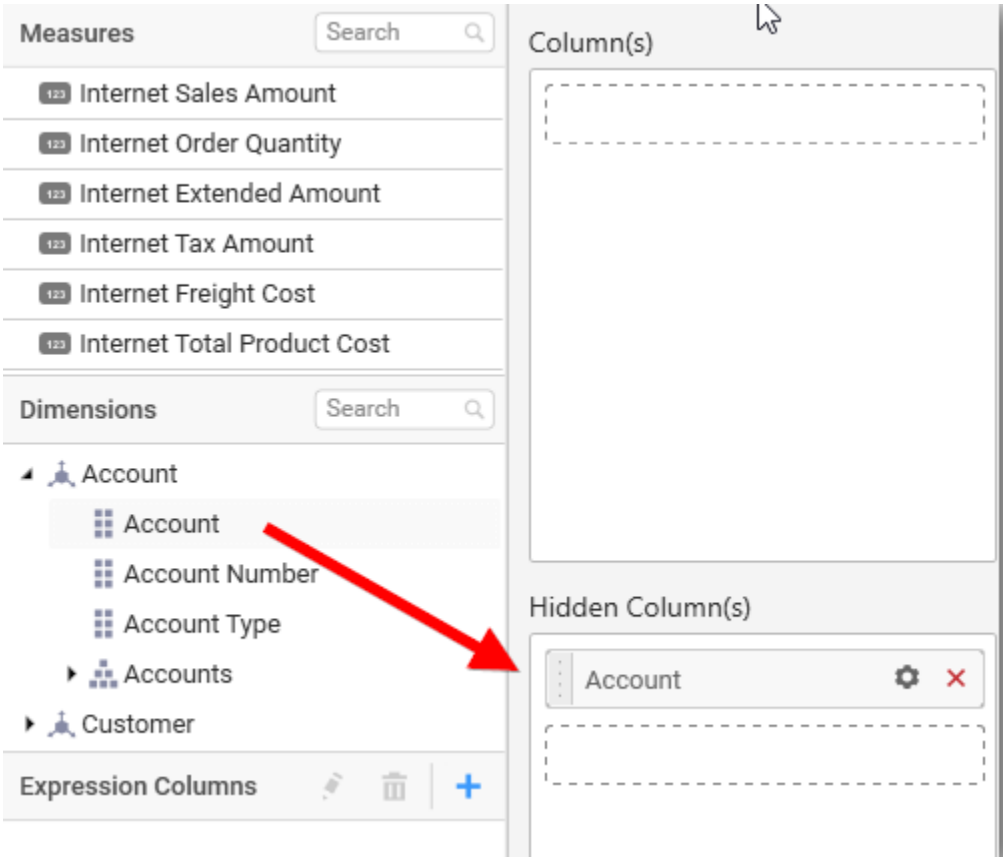
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



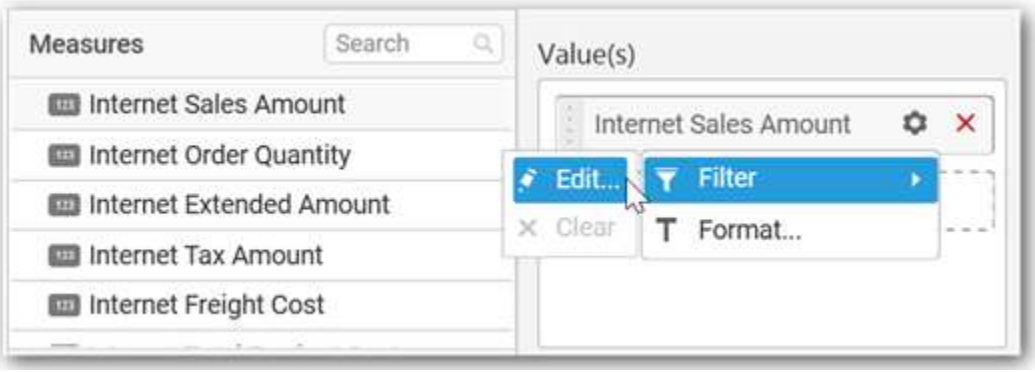
Drag and drop a column under Measures or Dimensions category into Column(s)



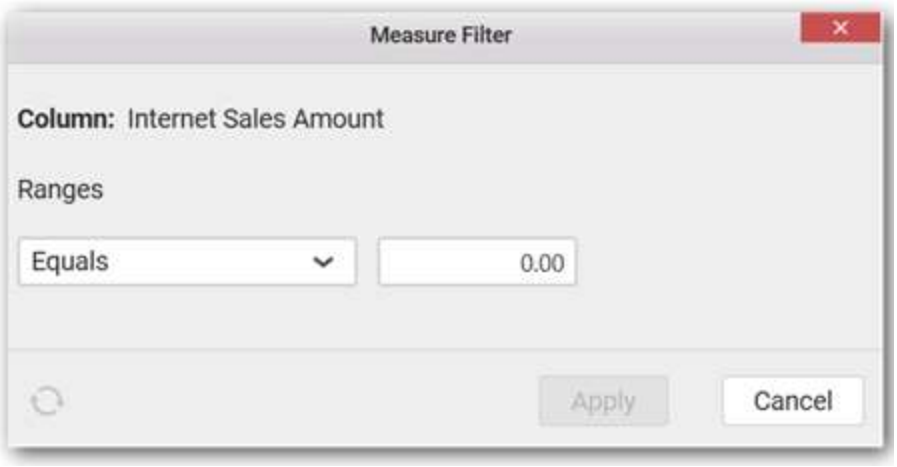
Drag and drop the elements to Hidden Columns if required. Based on the Hidden columns elements the values will be shown in grid widget.



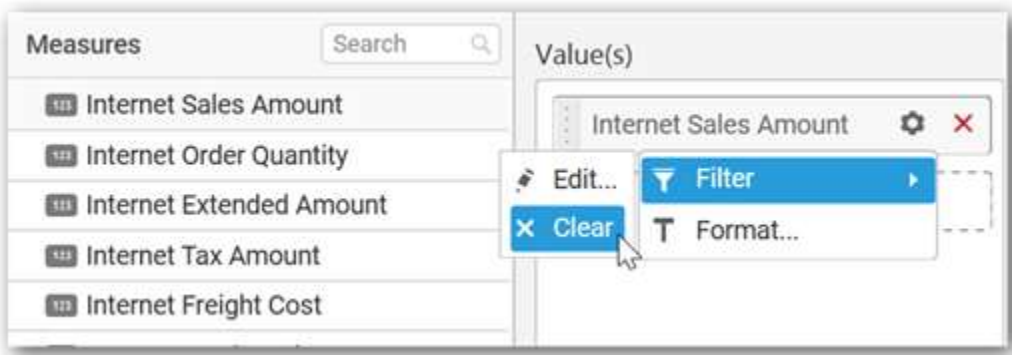
Define the filter criteria to match through choosing Edit option in Filter menu item.



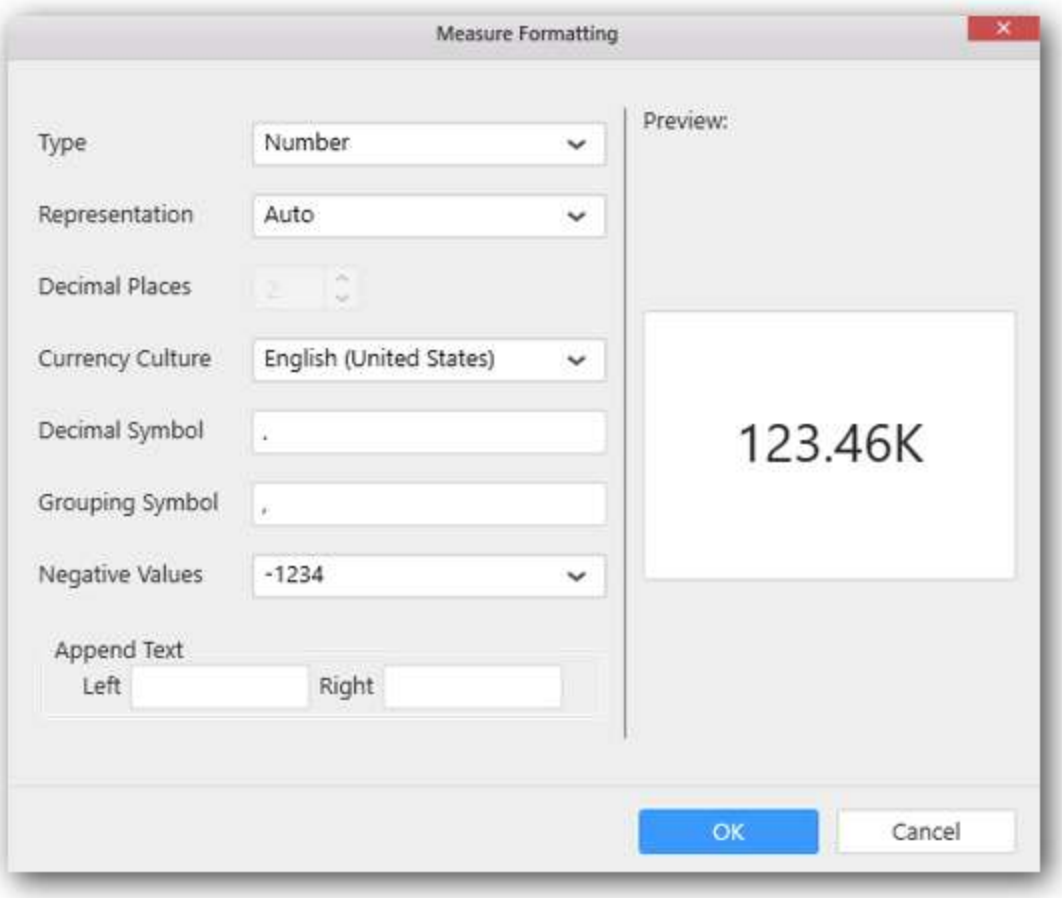
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



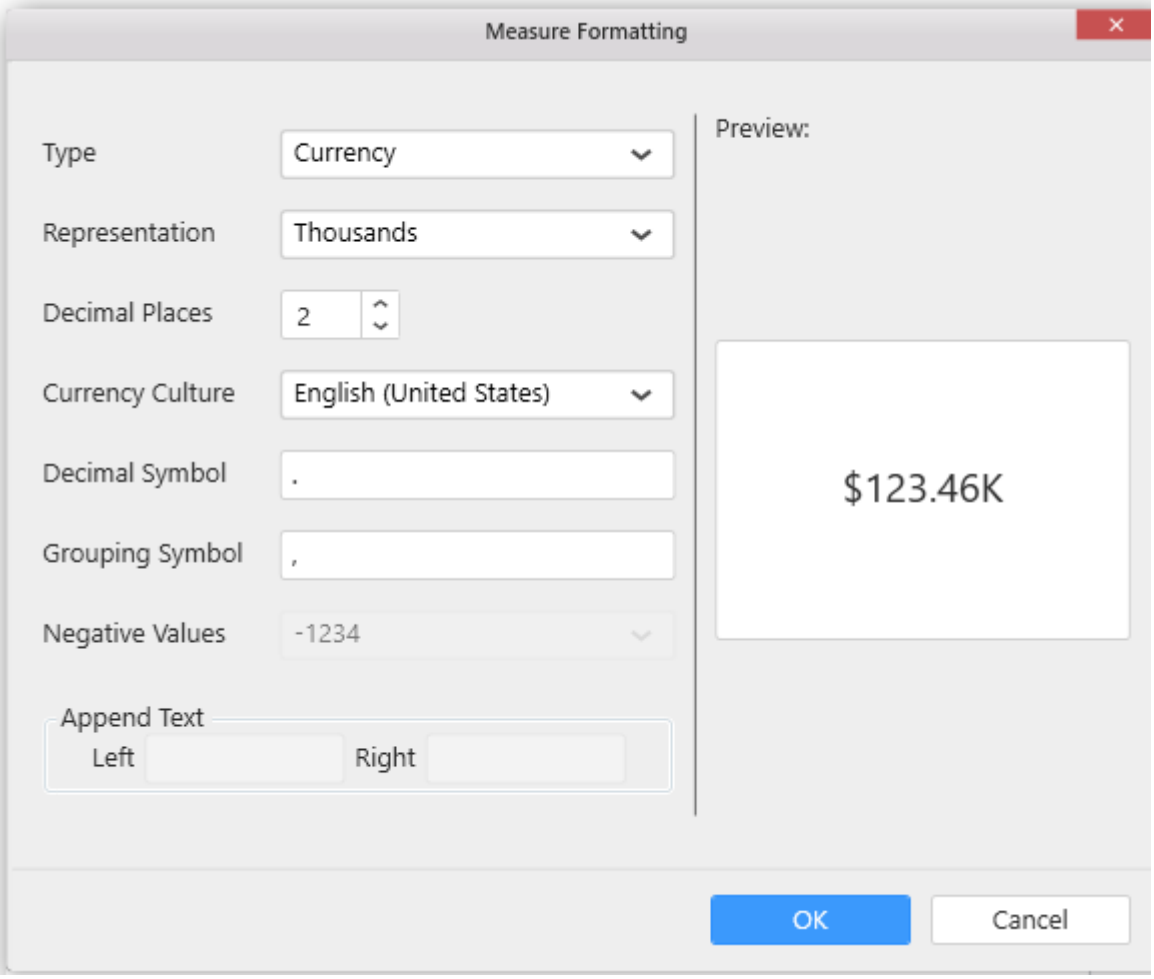
Select Clear option to clear the defined filter.



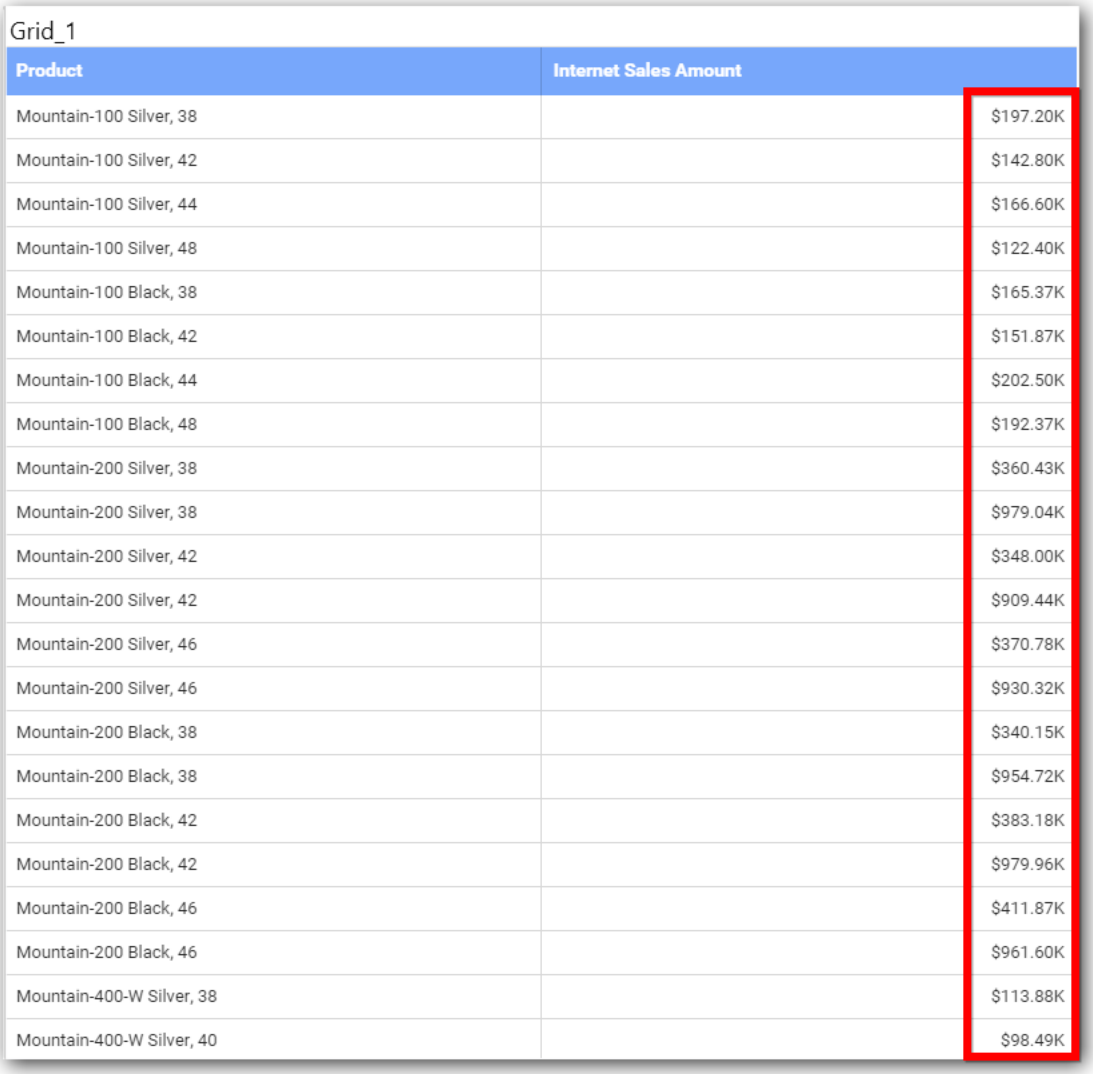
Select Format option to define the display format to the values in the column through Measure Formatting window.



Choose the options you need and click OK.



Here is an illustration,


Grid_1

Product	Internet Sales Amount
Mountain-100 Silver, 38	$197.20K
Mountain-100 Silver, 42	$142.80K
Mountain-100 Silver, 44	$166.60K
Mountain-100 Silver, 48	$122.40K
Mountain-100 Black, 38	$165.37K
Mountain-100 Black, 42	$151.87K
Mountain-100 Black, 44	$202.50K
Mountain-100 Black, 48	$192.37K
Mountain-200 Silver, 38	$360.43K
Mountain-200 Silver, 38	$979.04K
Mountain-200 Silver, 42	$348.00K
Mountain-200 Silver, 42	$909.44K
Mountain-200 Silver, 46	$370.78K
Mountain-200 Silver, 46	$930.32K
Mountain-200 Black, 38	$340.15K
Mountain-200 Black, 38	$954.72K
Mountain-200 Black, 42	$383.18K
Mountain-200 Black, 42	$979.96K
Mountain-200 Black, 46	$411.87K
Mountain-200 Black, 46	$961.60K
Mountain-400-W Silver, 38	$113.88K
Mountain-400-W Silver, 40	$98.49K

How to format Grid Widget?

You can format the Grid for better illustration of the view that you require, through the settings available in Properties pane.

General Settings

Header

This allows you to set title for this Grid widget.

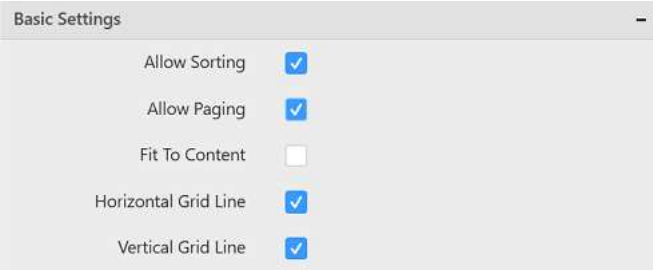
SubHeading

This allows you to set sub-title for this Grid widget.

Description

This allows you to set description for this Grid widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

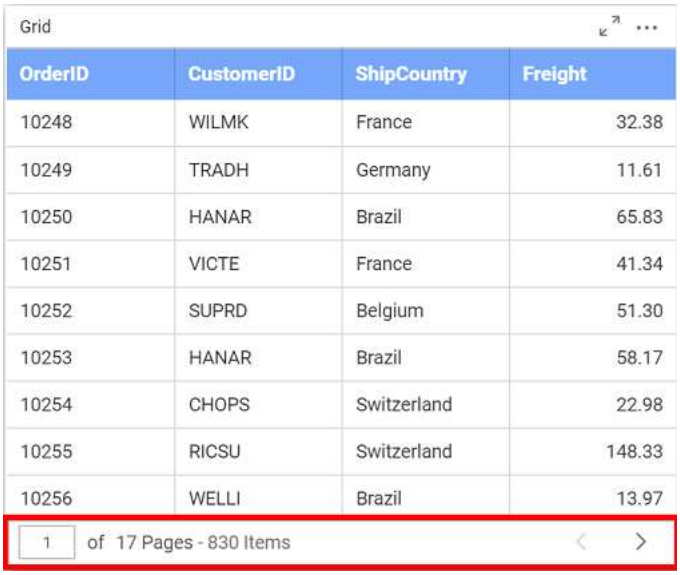


Allow Sorting

You can toggle the interactive sorting of columns in grid control using this. This option is enabled by default.

Allow Paging

This option allows you to enable/disable the Pager support in Grid control. After enabling this option you can find the current page number, total pages and total items information in the bottom of the Grid.



You can navigate to next and previous pages by clicking the left and right arrows respectively.

You also can jump to any page by entering the page number in the text box.

Fit To Content

The columns in the grid can be made to auto size based on the length of the content of the column.

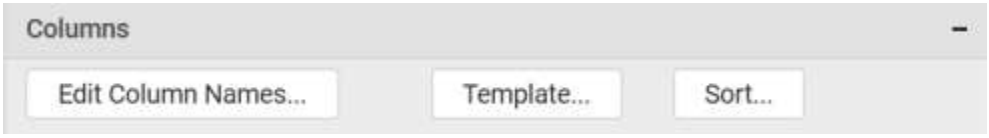
Horizontal Grid Line

You can enable/ disable horizontal grid lines in grid control. This option is enabled by default.

Vertical Grid Line

You can enable/ disable vertical lines in grid control. This option is enabled by default.

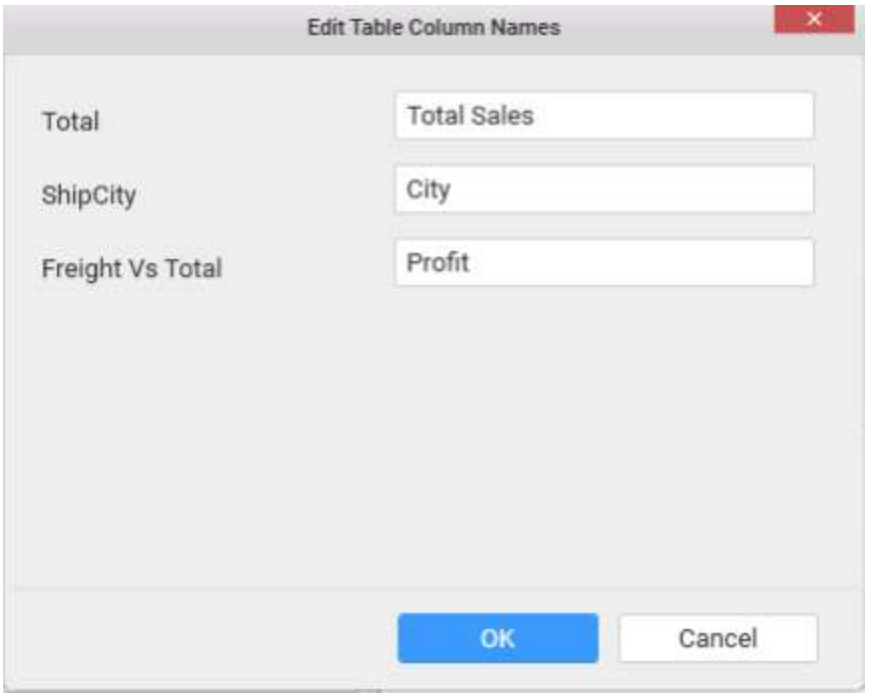
Column Settings



The Template and Sort options will be available only if you have a measure column added in Column(s) block.

Edit Column Names

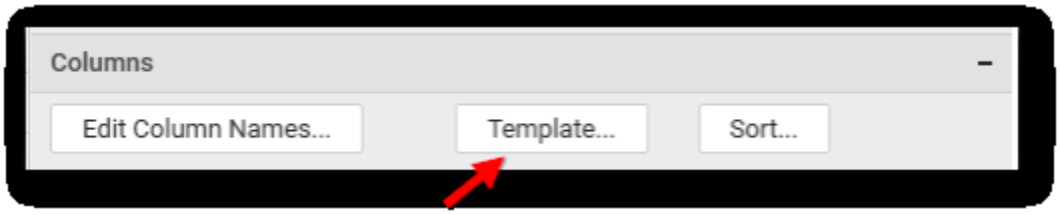
This allows you to customize the column names displayed in the grid by navigating to edit columns dialog from property panel. Through Edit Table Column Names window, a different name can be set to individual columns.



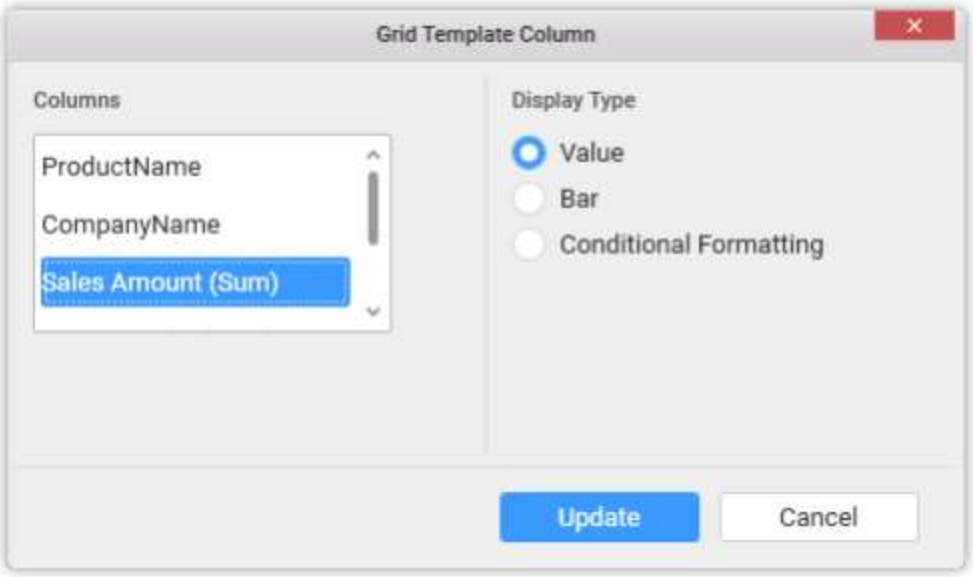
Once the changes are made, you can save by clicking the OK button.

Template

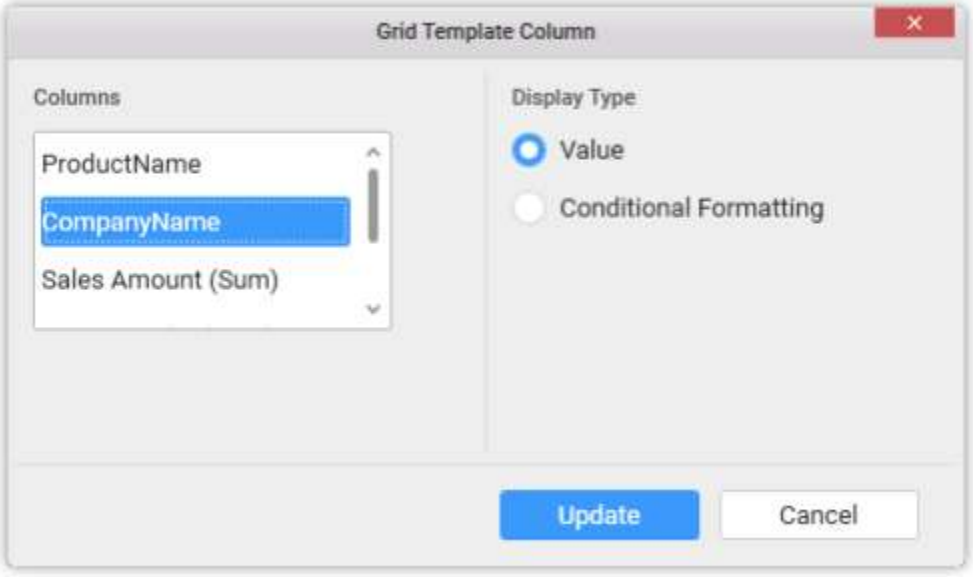
You can define the column value represented as text, bar or condition based coloring. Click Template to launch the Grid Template Column dialog. This lists out the columns added to the grid widget. For each of those columns, the value representation can be configured through options displayed at right.



For measure type column,



For dimension type column,



To define the value representation for a column, select the respective column from the Columns list in the left pane and select the display type for the same at right pane.

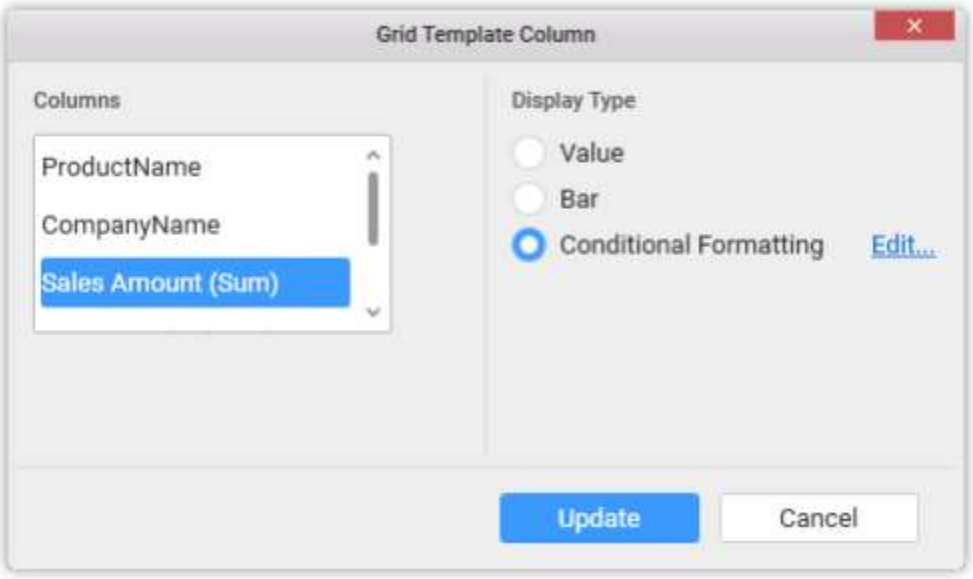
Select Value as display type, to get the column values represented as it is.

Product Name	Company Name	Sales Amount
Côte de Blaye	Aux joyeux ecclésiastiques	164,424
Mishi Kobe Niku	Tokyo Traders	108,640
Carnarvon Tigers	Pavlova, Ltd.	39,000
Gustaf's Knäckebröd	PB Knäckebröd AB	31,584
Ikura	Tokyo Traders	29,760

Select Bar as display type, to get the column values represented as progress bar.

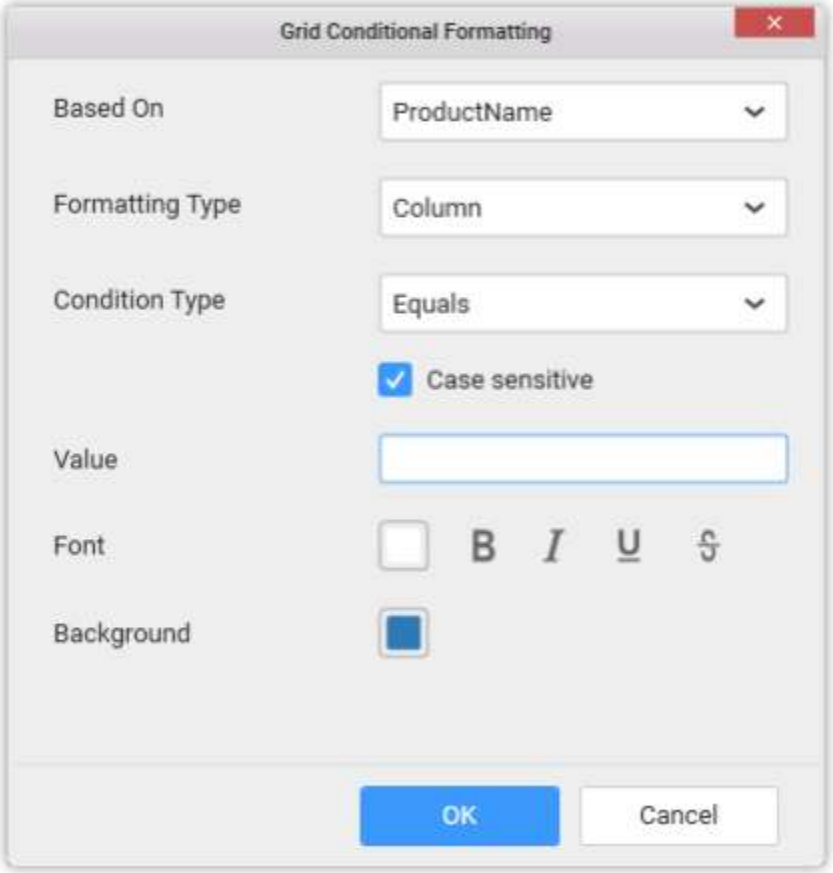
Product Name	Company Name	Sales Amount
Côte de Blaye	Aux joyeux ecclésiastiques	
Mishi Kobe Niku	Tokyo Traders	
Carnarvon Tigers	Pavlova, Ltd.	
Gustaf's Knäckebröd	PB Knäckebröd AB	
Ikura	Tokyo Traders	

Select Conditional Formatting to configure conditions and apply color to the cells based on that.



The Grid Conditional Formatting dialog can be opened through clicking the Edit link button.

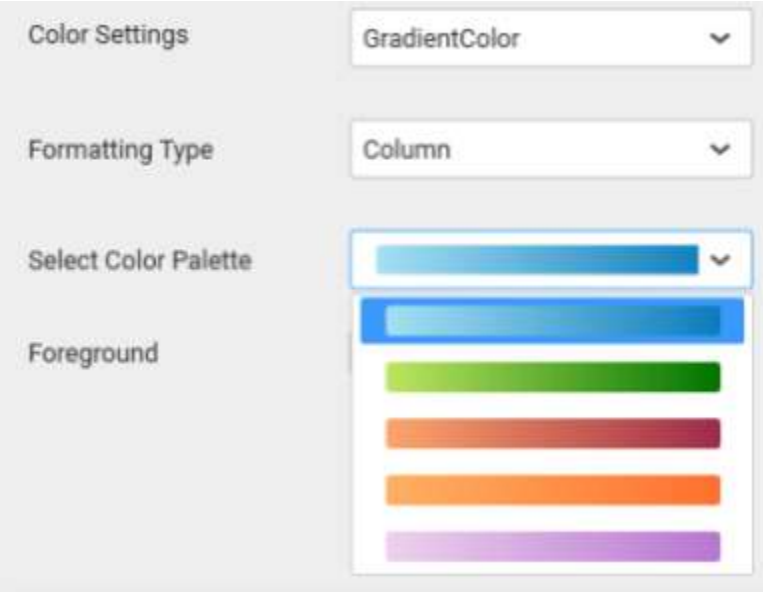
Add one or more conditions through clicking the + icon.



In this dialog,

Based On – Shows added column list out of which a column can be selected over which the condition need to be defined and applied to the column selected for template display.

Color Settings – It can be Single Color or Gradient Color. Selecting Single Color will apply the selected color to the rows/column that meets the filter criteria. Selecting Gradient Color will apply the selected palette to the rows/column that meets the filter criteria.



Formatting Type – It can be Row or Column which need to be applied with the formatting.

Column Formatted Grid

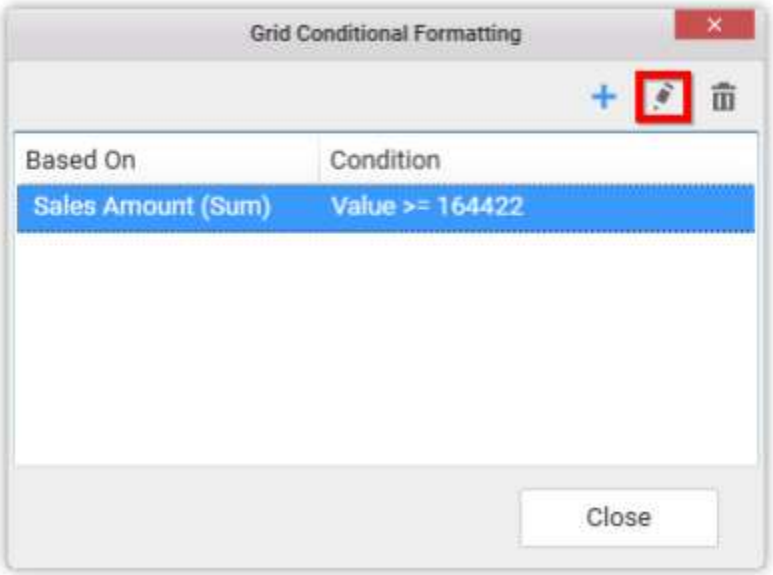
Row Formatted Grid

There is one more formatting type Indication. This type will be visible only when Gradient Color is selected. Selecting this type, will add a new column with indicator.

Condition Type – The compare operator can be set to compare values against.

Other font settings like color, style can be set.

Click OK to save the condition. You can also edit an already added condition through selecting the respective condition and click the Edit icon highlighted below.

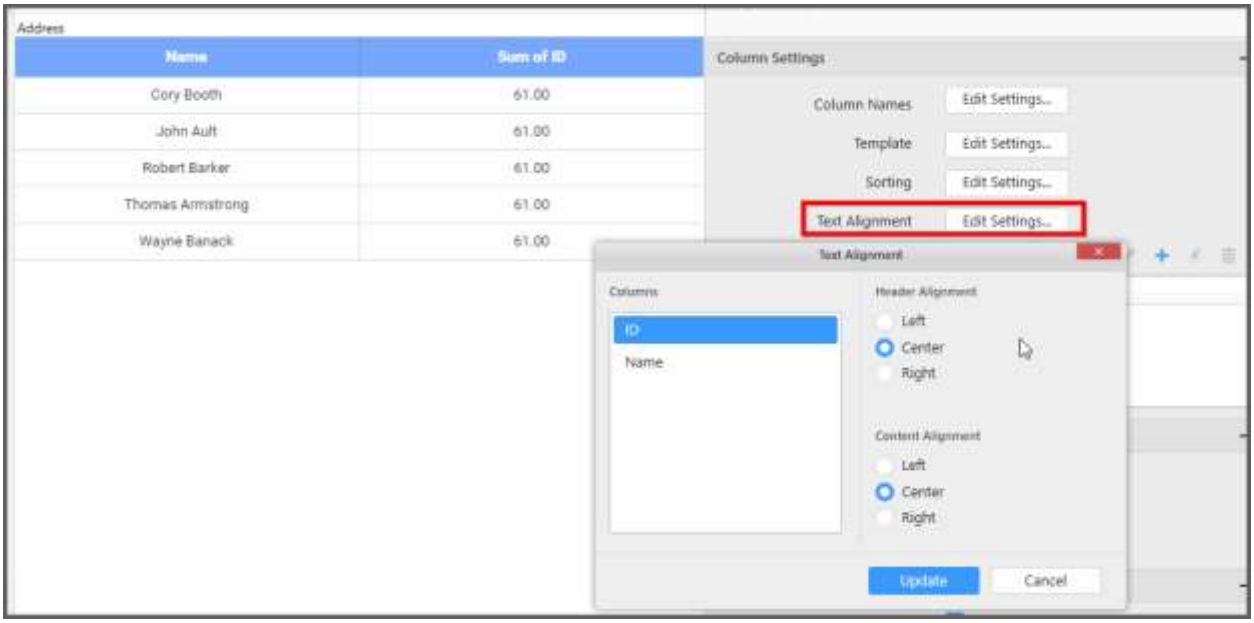


Sorting

You can sort the content in grid control based on the measure fields added by navigating to Grid Sort Settings dialog throughSort button click. The grid data can be sorted based on a single measure column only at a time.

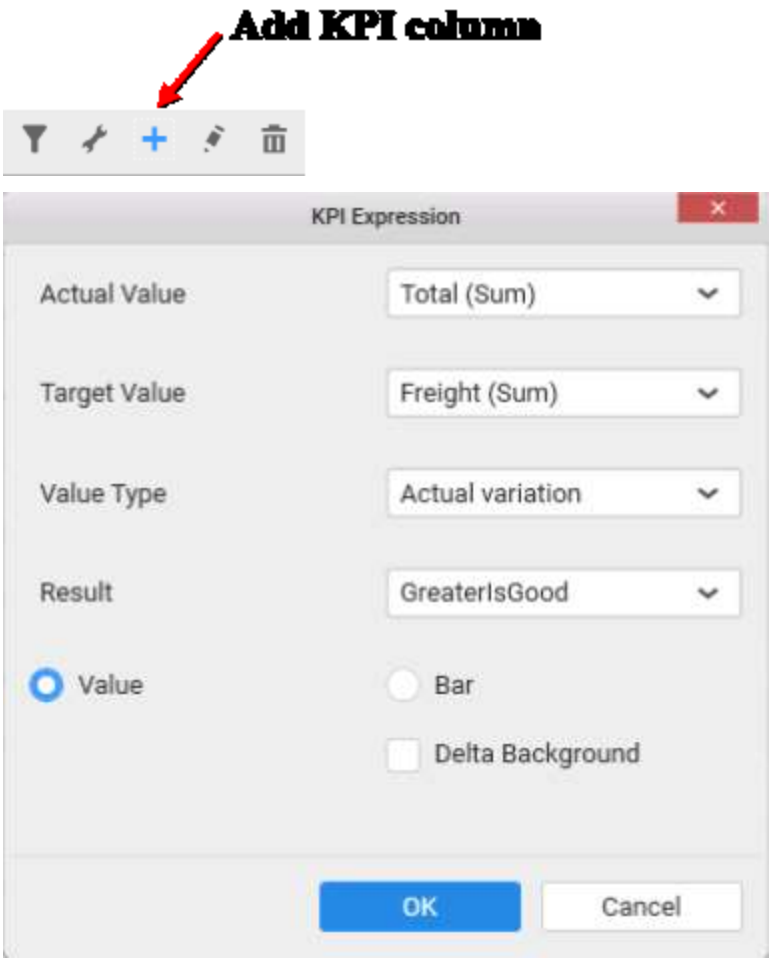
Text Alignment

You can handle the alignment of widget's header text and content text to either left, center or right in grid control.



Key Performance Indicator (KPI)

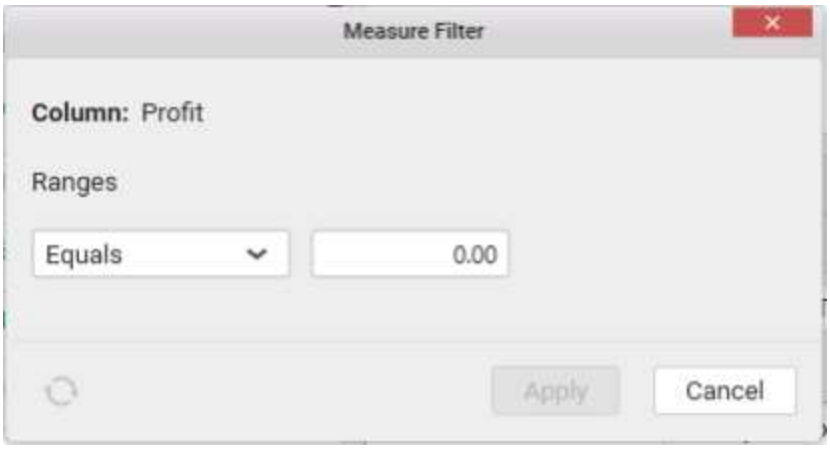
You can add Key Performance Indicator (KPI) columns in grid control by navigating to KPI Expression window by clicking Add KPI button from property panel at top.



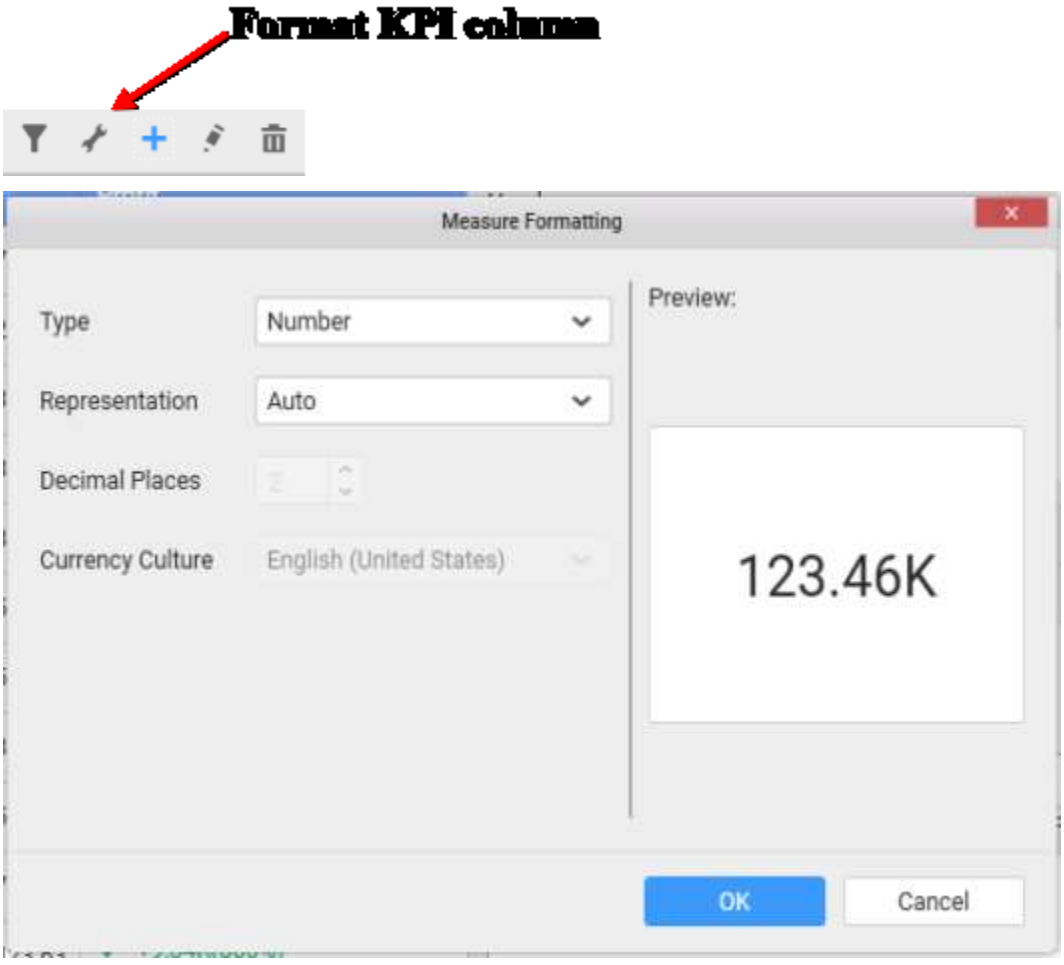
From the KPI Expression dialog, you can specify the column whose values need to be considered as actual value and the column that need to be considered as Target. The value type can be set based on which the KPI will be calculated. The following value types are available.

- Actual Variation (Default)
- Actual Value
- Percentage of variation
- Percentage of target
- Value and Percentage

The Result can be set to showcase the result as gain or loss based on which the value will be visualized. You can choose the type as value or graphical bar to showcase the data in the column.

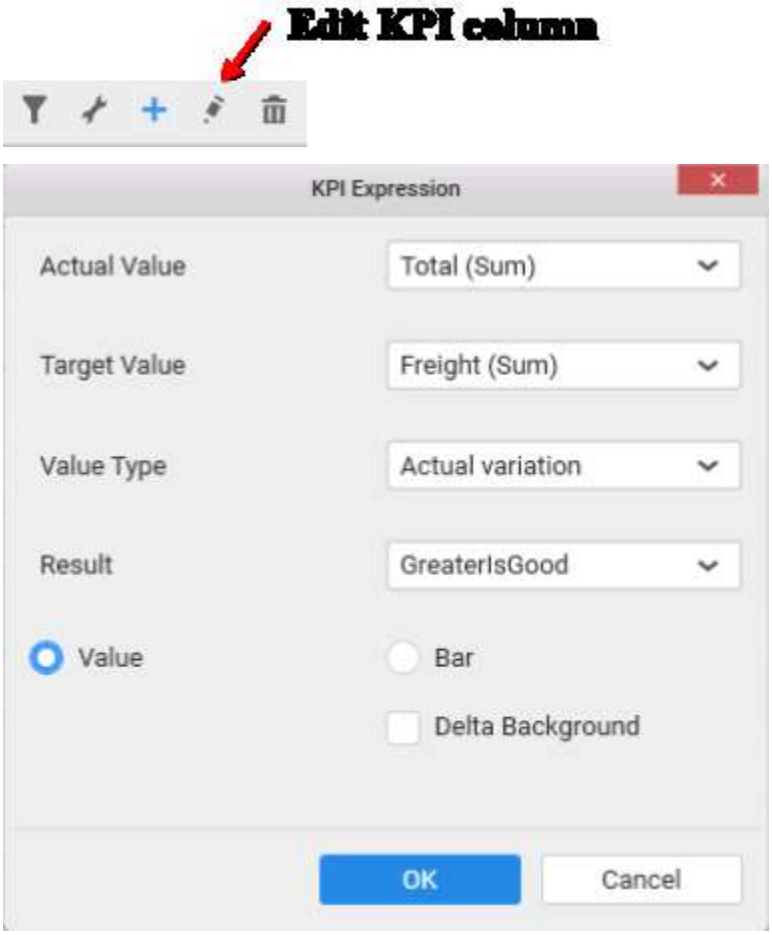


The KPI column can be filtered based on the measure values bind to the KPI. You can set measure filter by clicking the filter icon button, which will open the Measure Filter dialog from where you can specify the column and the condition for filtering the data showcased.



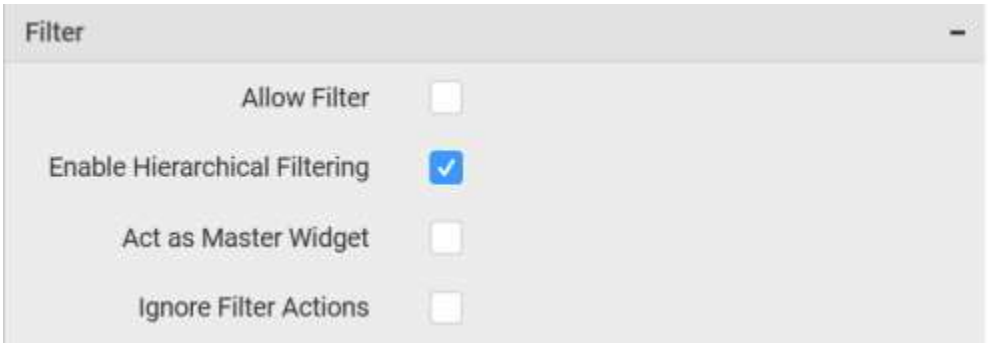
The values showcased in KPI column can be formatted just like any other measure column. You can open the Measure Formatting dialog box by clicking the Format KPI Column button. This allows you to handle different formatting options like display type, representation, decimal places and currency culture to the respective KPI column added.

You can Edit KPI column by clicking the Edit KPI column icon.



You can delete KPI column by clicking the Delete KPI column.

Filter Settings



Allow Filter

This allows you to enable a filter box for each column in the grid for easy filtering of data through this option.

Enable Hierarchical Filtering

Through this option, you can enable/disable hierarchical top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added.

When Flat is set, the least number set as top will be applied for the whole data. When Hierarchical is set, the Top N will be applied for each individual column separately based on the number set for each column.

Below example shows data of 3 Country and its 2 Cities where the sales is high.

Flat Top N

Sales

Country	City	Total Sales
Germany	Cunewalde	117,411.33
USA	Boise	115,560.28

Hierarchical Top N

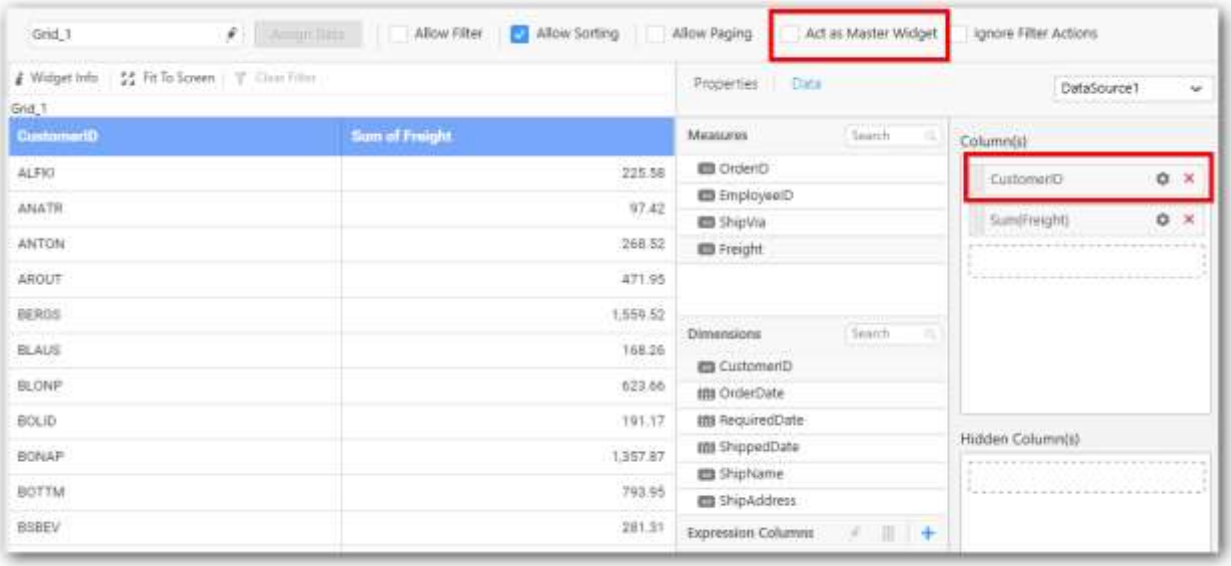
Sales

Country	City	Total Sales
Austria	Graz	113,153.06
Austria	Salzburg	26,228.64
Germany	Brandenburg	31,737.38
Germany	Cunewalde	117,411.33
USA	Albuquerque	52,234.42
USA	Boise	115,560.28

Act as Master Widget

This allows you to define this grid widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

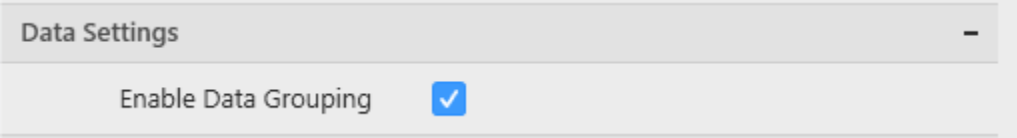
Act as Master Widget option will be enabled when you use dimension column in grid widget.



Ignore Filter Actions

This allows you to define this grid widget to ignore responding to the filter actions applied on other widgets in dashboard.

Data Settings



Enable Data Grouping

To be short, if you want to show the data as such from database without any aggregation or grouping then uncheck the Enable Data Grouping option under Data Settings.

In grouping mode, aggregation is applied for measure fields and grouping is applied for dimension fields.

In absence of grouping mode, raw data will be populated in the Grid widget. In such case, measure fields will show data without aggregation and dimension fields will show data without grouping.

When switching from data grouping mode to raw data mode, the following changes will happen in designer.

- Expressions which are created using aggregation function can't be configured in Grid widget in raw data mode. Such expressions will be automatically removed.
- KPI created with aggregation based expression would also be automatically removed.
- You won't be allowed to switch to raw data mode, if Data Filter is applied with Top N for the selected data source.

When switching from raw data mode back to data grouping mode, state maintenance happens in designer with few limitations.

- Aggregation function for measures will be restored back to "SUM" which is the default aggregation type.
- The applied condition filter, rank filter will be cleared.

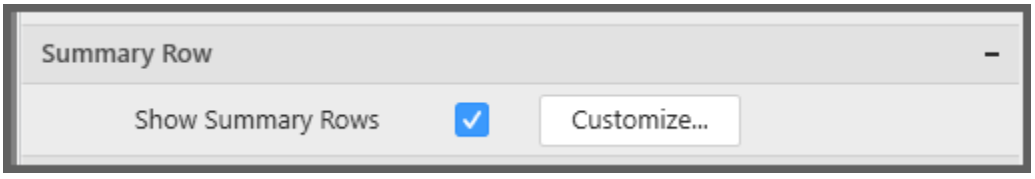
Note: ODBC ANSI SQL in Live Mode and SSAS connection does not support raw data.

Grid_1

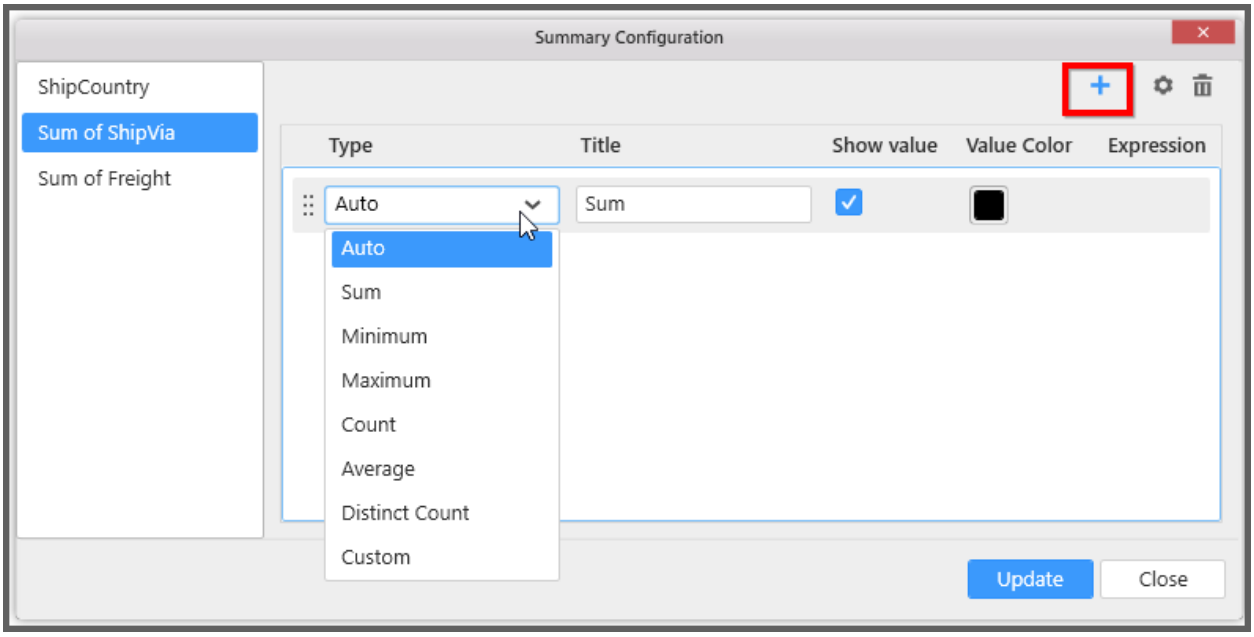
ShipCity	ShipCountry	ShippedDate	Freight
Aachen	Germany	1996	30.54
Aachen	Germany	1996	5.45
Albuquerque	USA	1996	2
Albuquerque	USA	1996	98.03
Albuquerque	USA	1996	147.26
Albuquerque	USA	1996	74.16
Albuquerque	USA	1996	150.15
Albuquerque	USA	1996	142.08
Albuquerque	USA	1997	12.51
Albuquerque	USA	1997	708.95
Albuquerque	USA	1997	13.75
Albuquerque	USA	1997	58.98
Albuquerque	USA	1997	44.42
Albuquerque	USA	1997	18.66

Summary Row

This allows you to show the summarized value of the column based on the selected type.



You can add multiple summary rows for a column, and also you can add a custom expression to summarize the value.



Type

This allows you to choose the functionality to be done in the column.

Title

This allows you to set as title for the summarized column value.

Show Value

This allows you to show the summarized value, if it was checked.

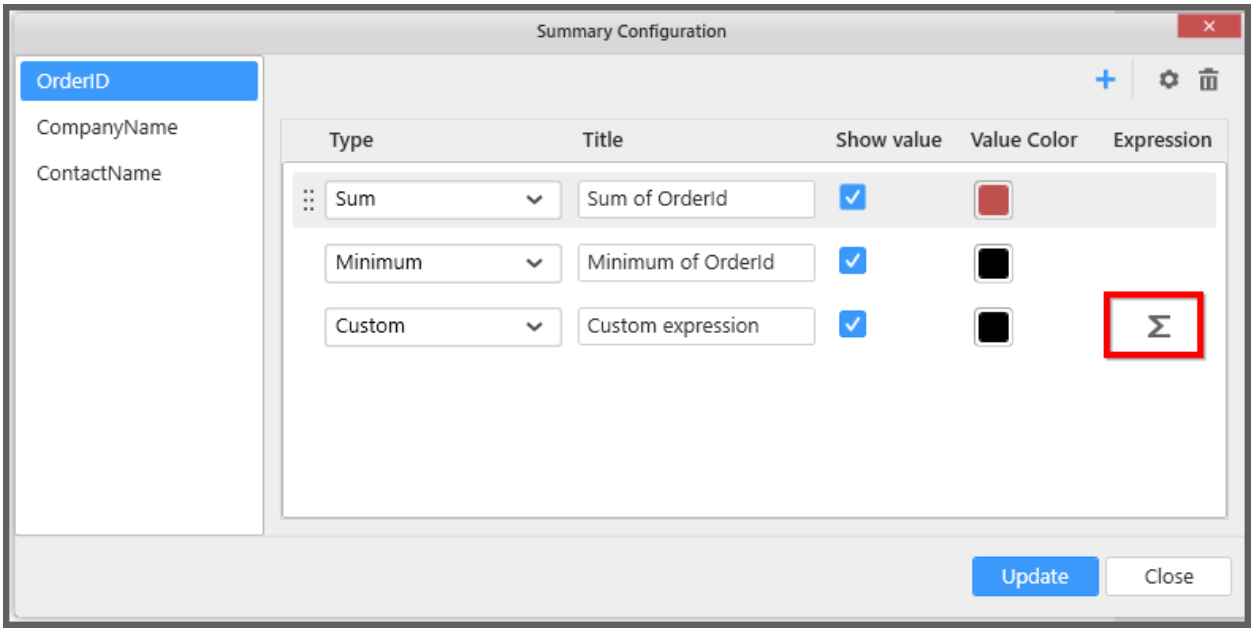
Value Color

This allows you to apply color for the Summarized value.

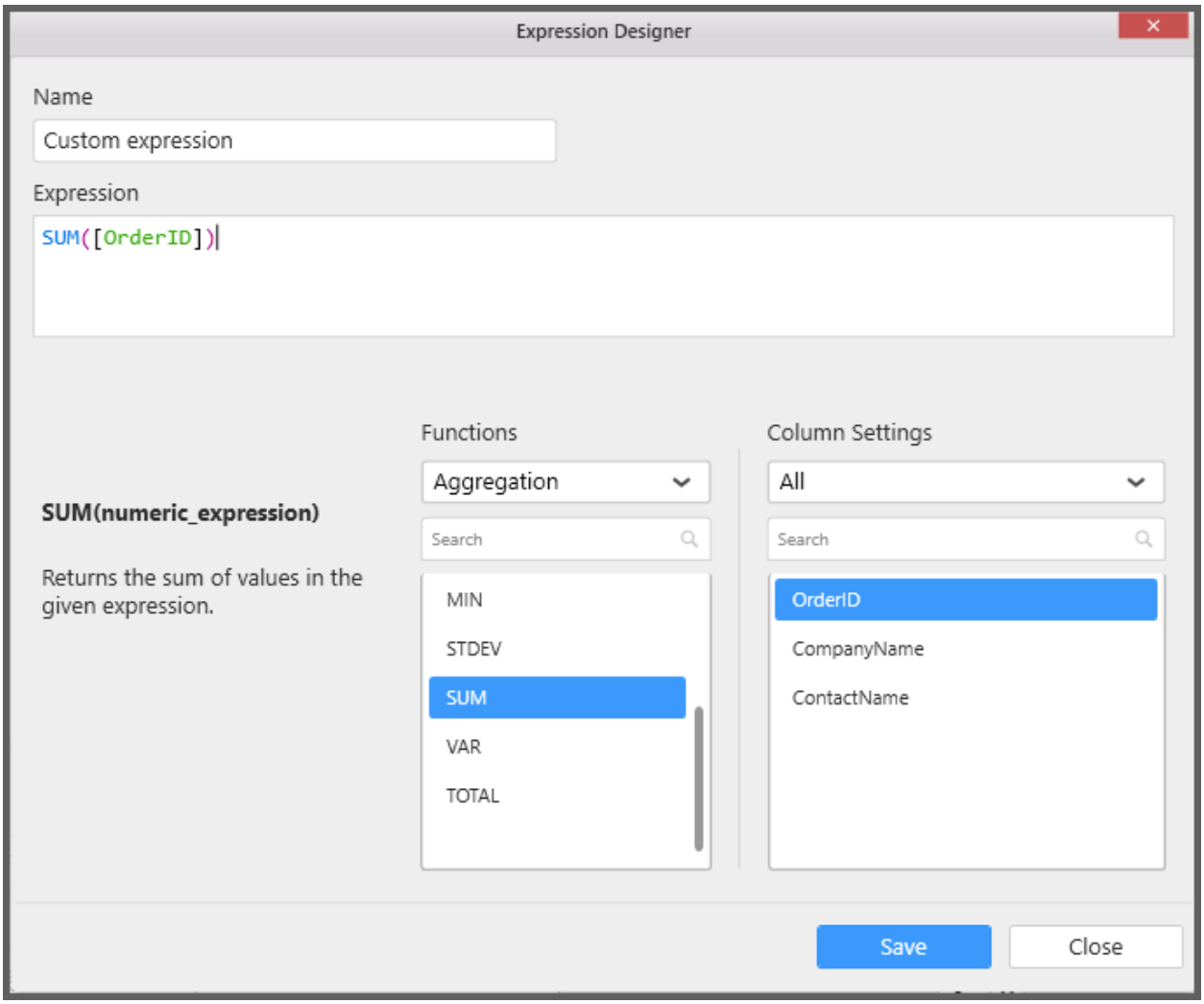
Expression

This allows you to have your own custom expression to summarize values.

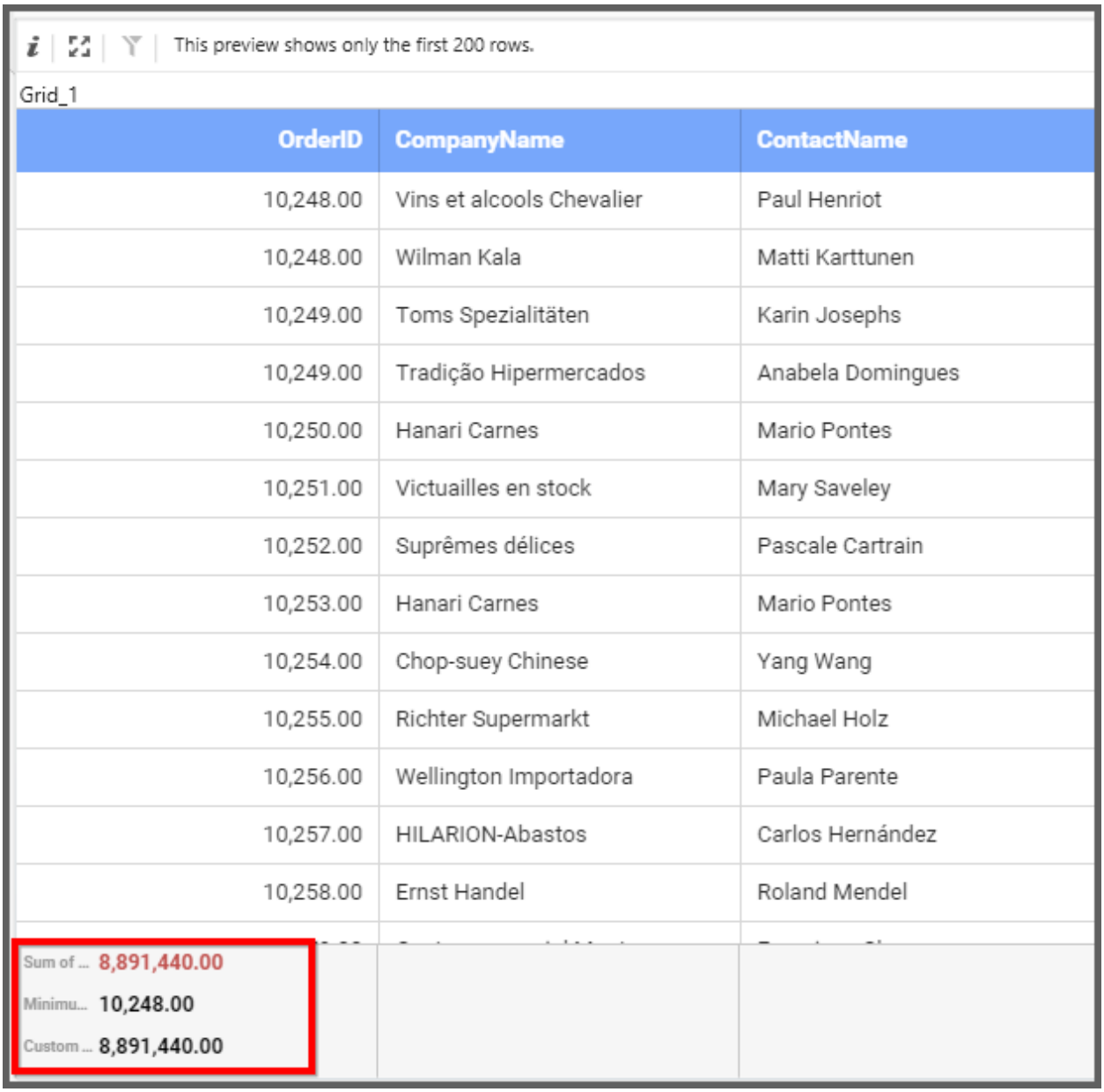
Select `Custom` type to write your own custom expression.



Click on the above highlighted sum icon to open expression designer.



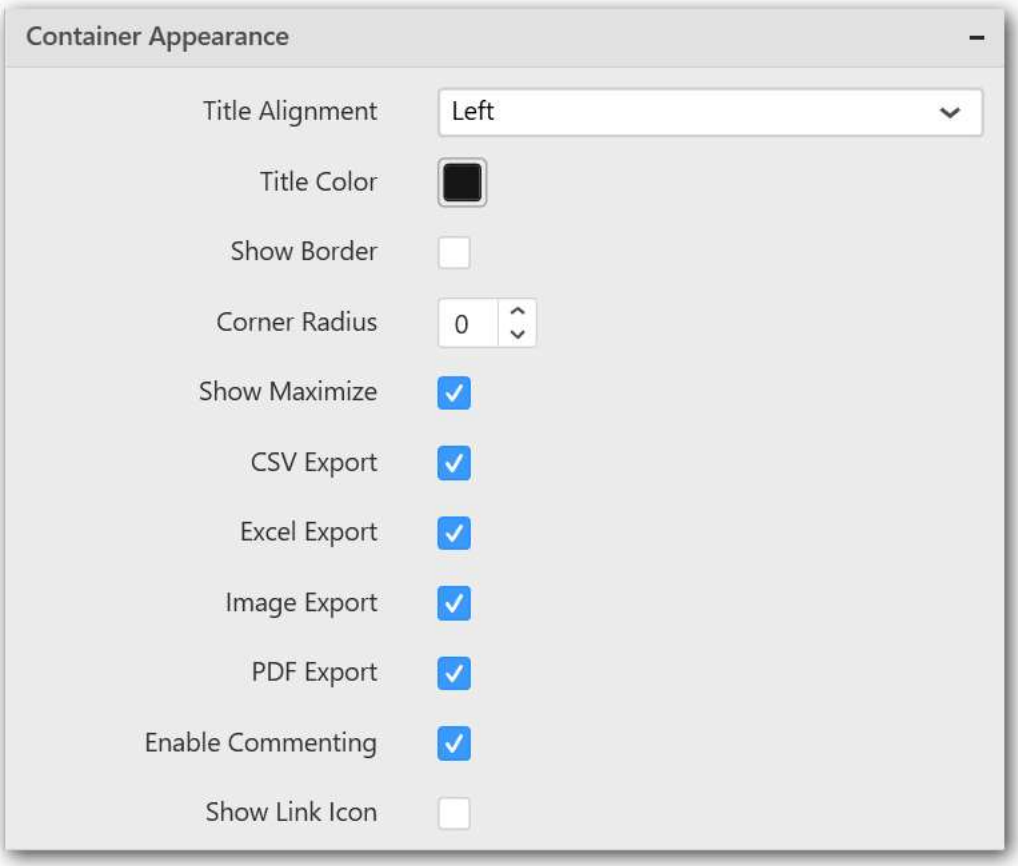
You can find the summarized output in the grid as below.



Link Settings

You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this grid widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the grid widget.

CSV Export

This allows you to enable/disable the CSV export option for this grid widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this grid widget. Enabling this allows you to export the summarized data of the widget view to XLSX format.

Image Export

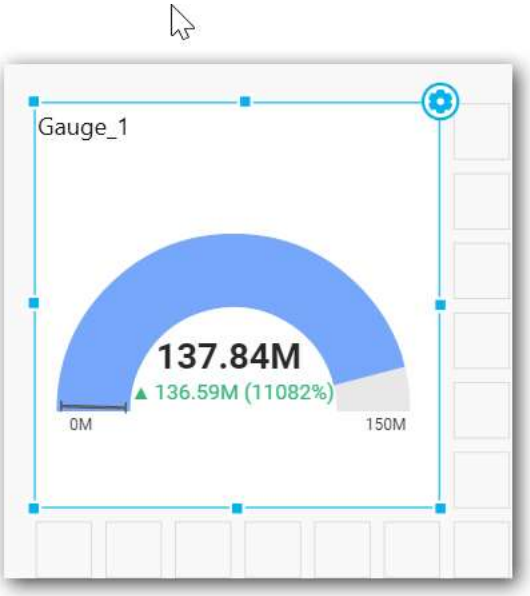
This allows you to enable/disable the image export option for this grid widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Radial Gauge

Radial Gauge allows you to measure processing efficiency through key performance indicators like Value and Goal. To showcase a radial gauge, a minimum requirement of 1 actual or target value is needed.

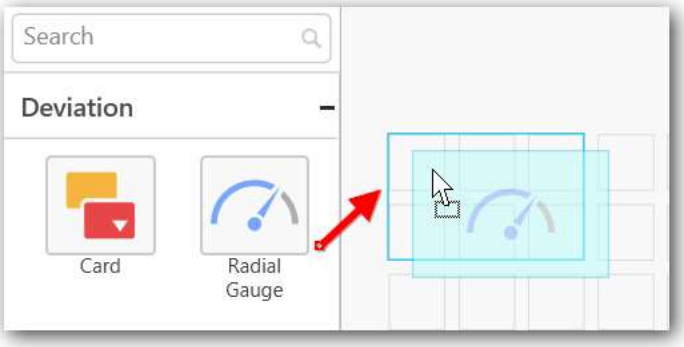


How to configure flat table data to Radial Gauge?

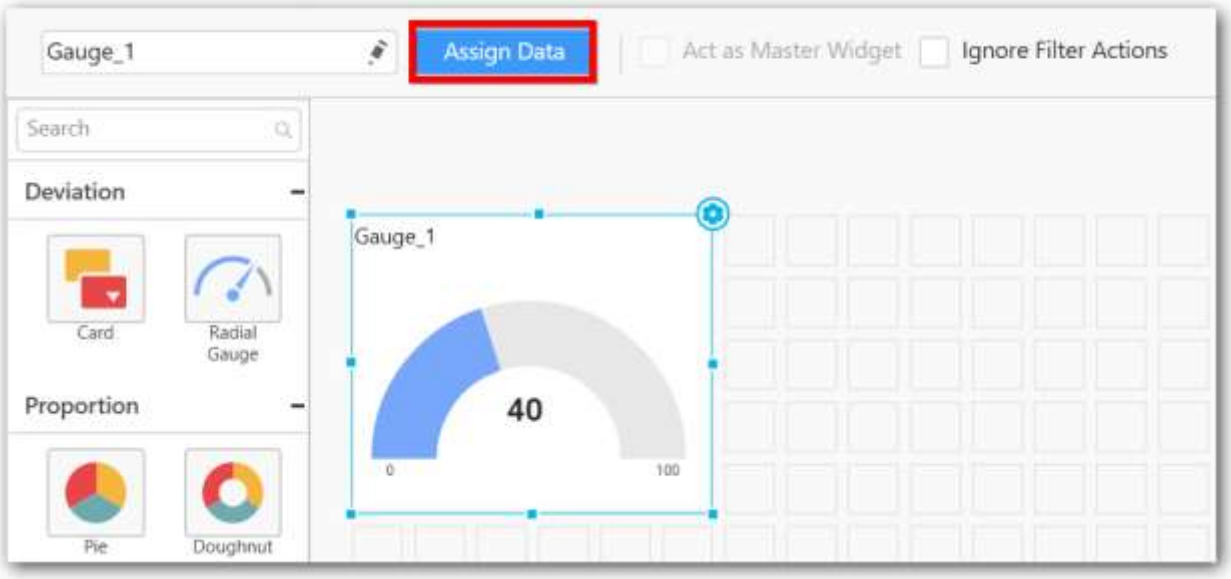
To construct a radial gauge, a minimum requirement of 1 column is needed. You can visualize both measure, calculated measure and dimension column data in radial gauge control.

The following procedure illustrates data configuration of Radial Gauge.

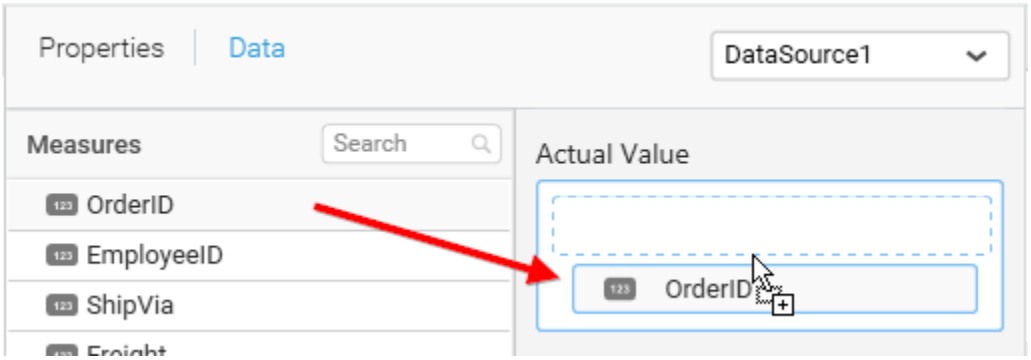
Drag and drop Radial Gauge control icon from the Tool box into design panel. You can find control in Toolbox by search.



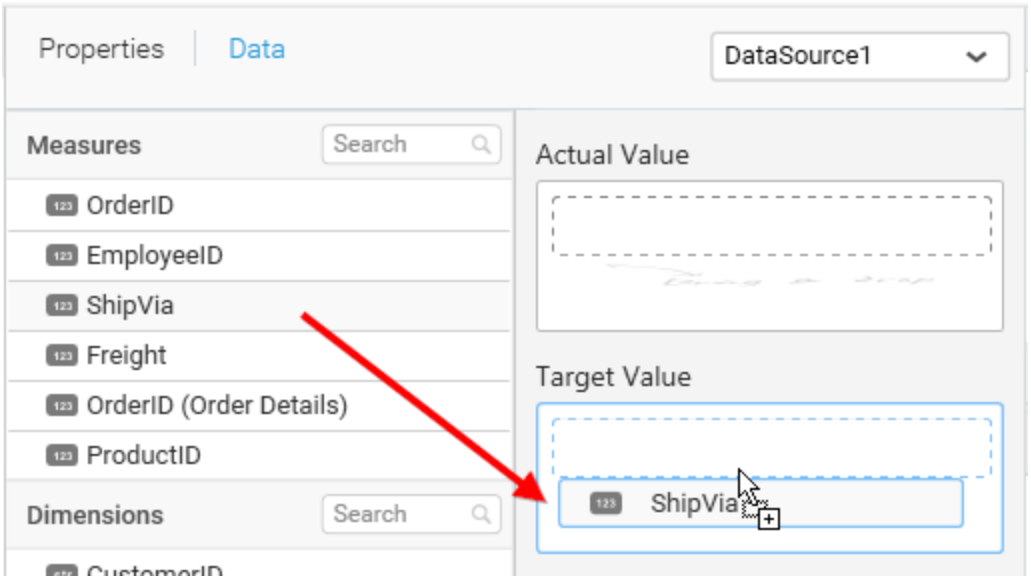
After control added in design panel, click **Assign Data** button at Design Tools Pane to open the Data configuration pane.



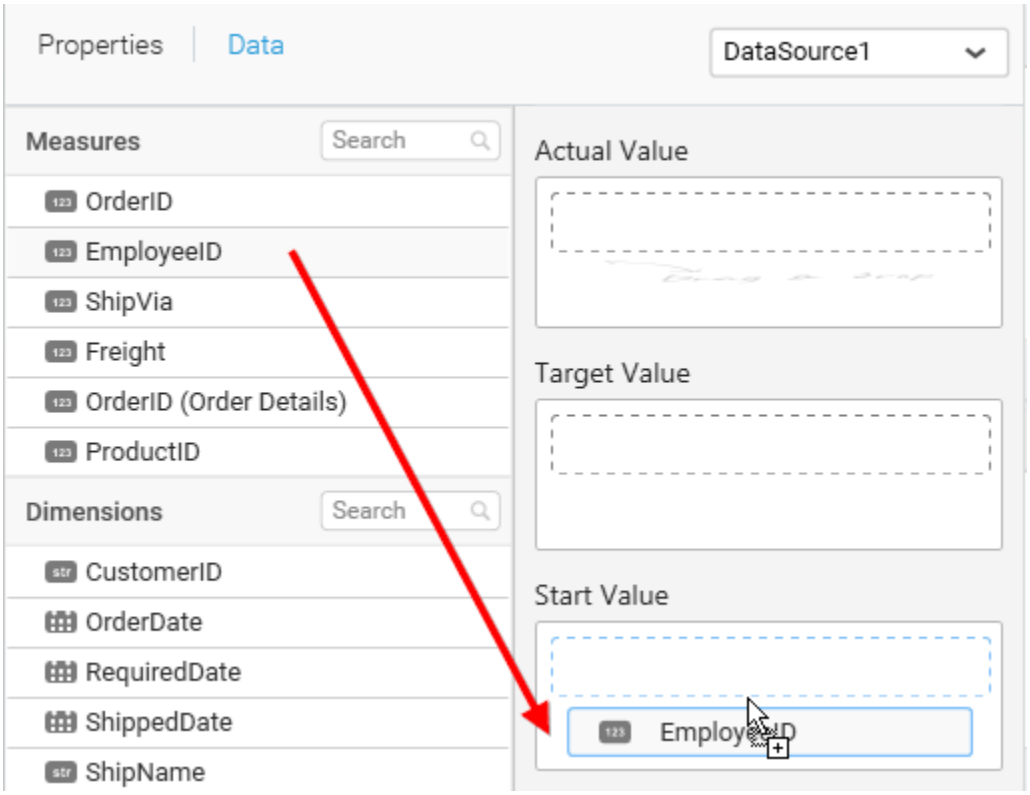
Bind column through drag and drop element from Measures section to Actual Value.



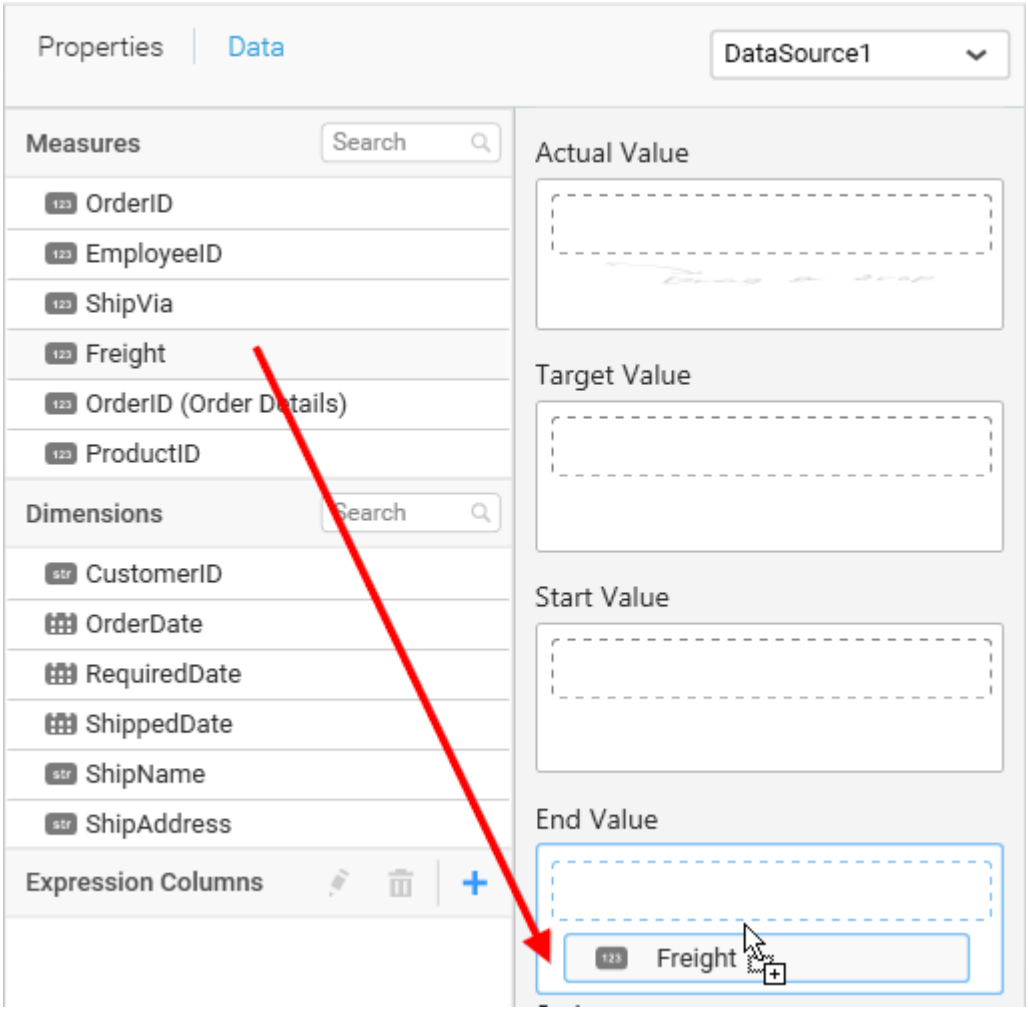
Drag and Drop the elements from section to Target Value.



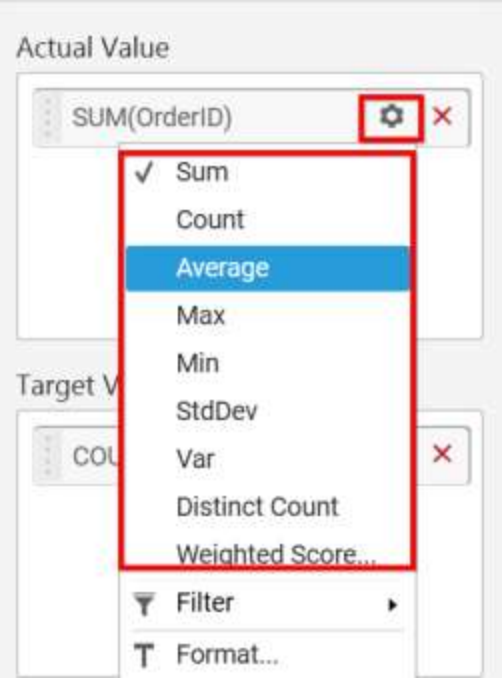
Drag and Drop the elements from section to Start Value.



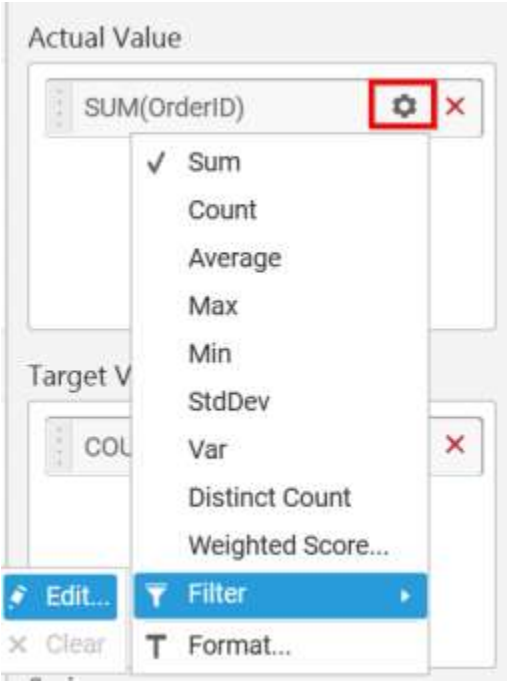
Drag and Drop the elements from section to End Value.



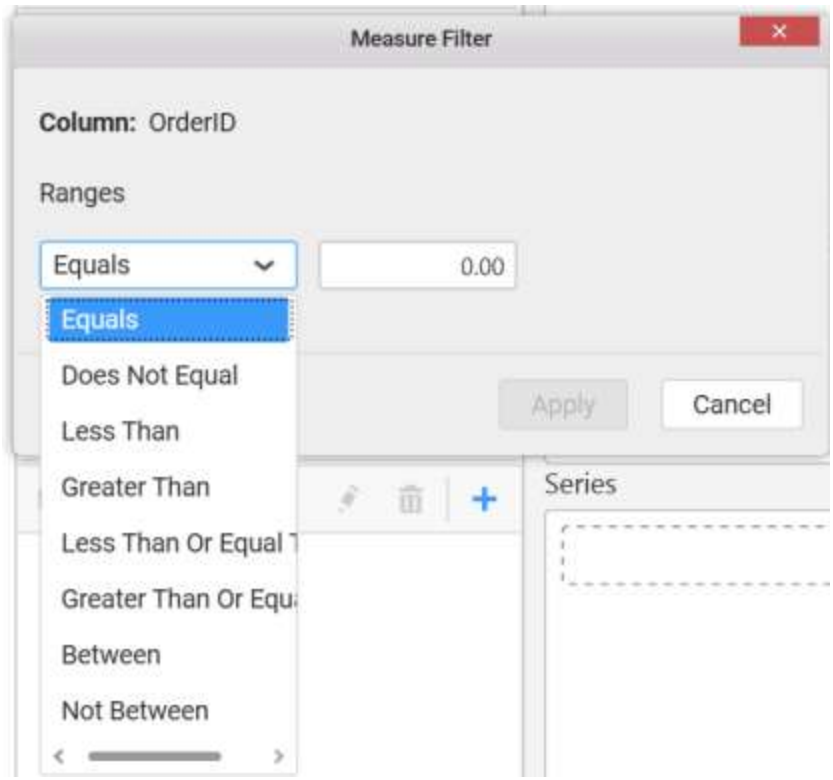
You can use aggregate function to change the Actual Value of the radial gauge widget.



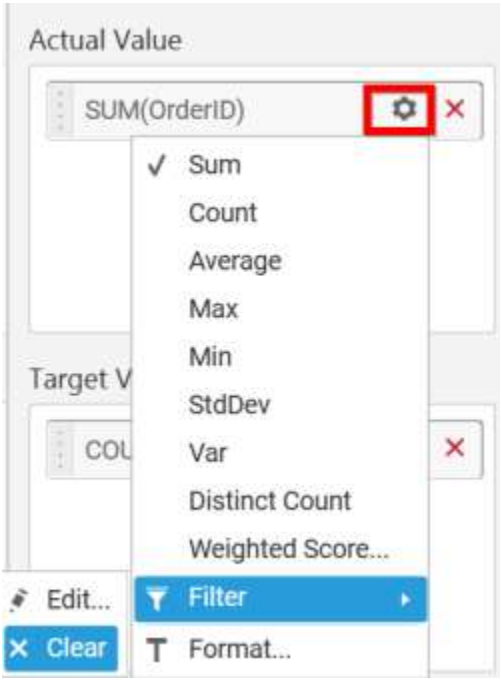
You can use Filter option to filter the data by specifying the filter condition.



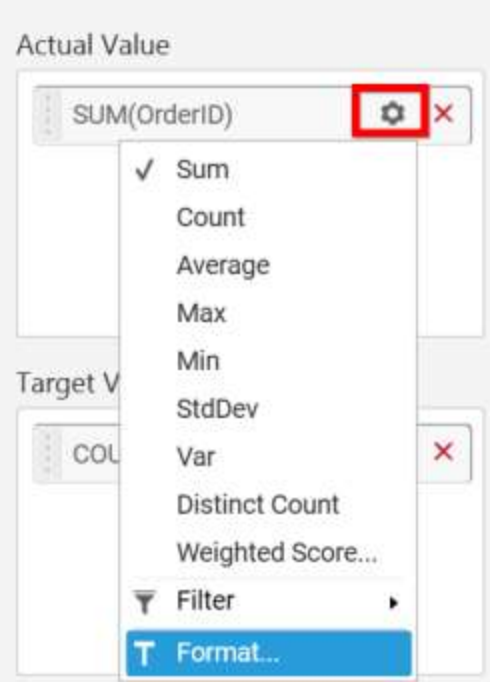
Measure Filter will be shown to set the Ranges.



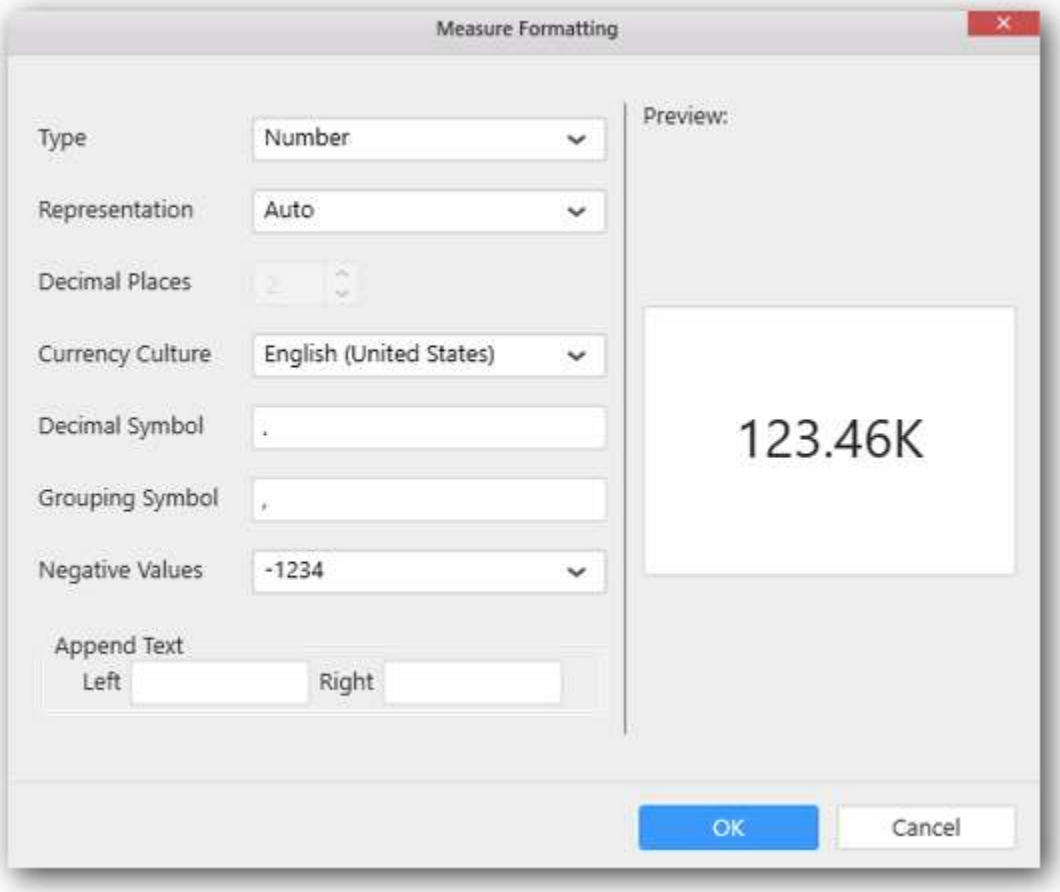
You can clear the filter by selecting the Clear option.



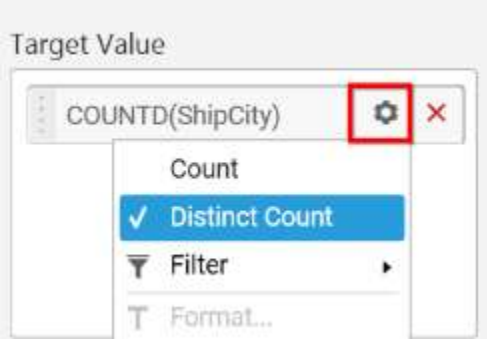
You can format the elements by selecting the Format option.



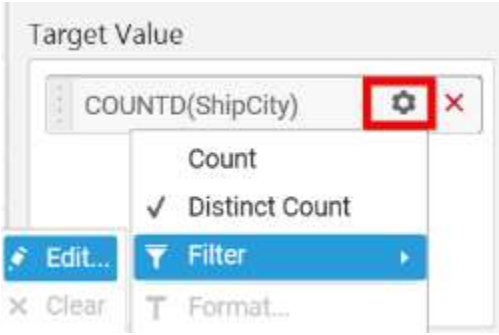
Measure Formatting window will be shown to change the measure.



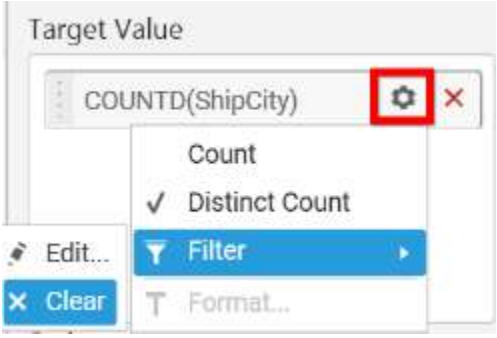
You can change the Target Value by changing the settings.



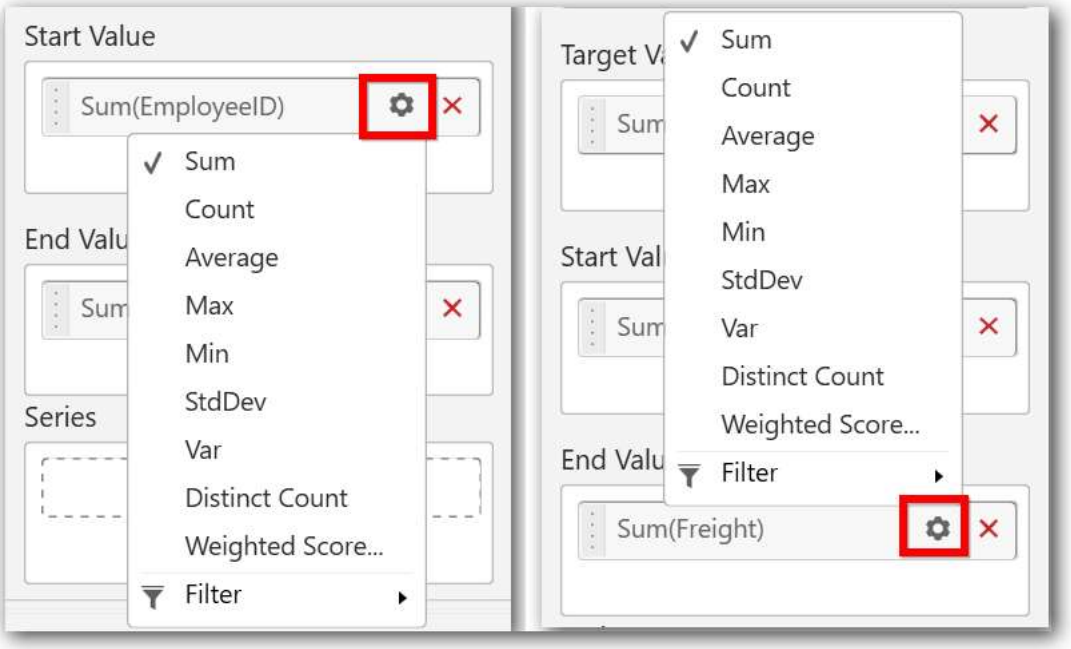
Filters can be applied to the Target Value by changing the filter condition.



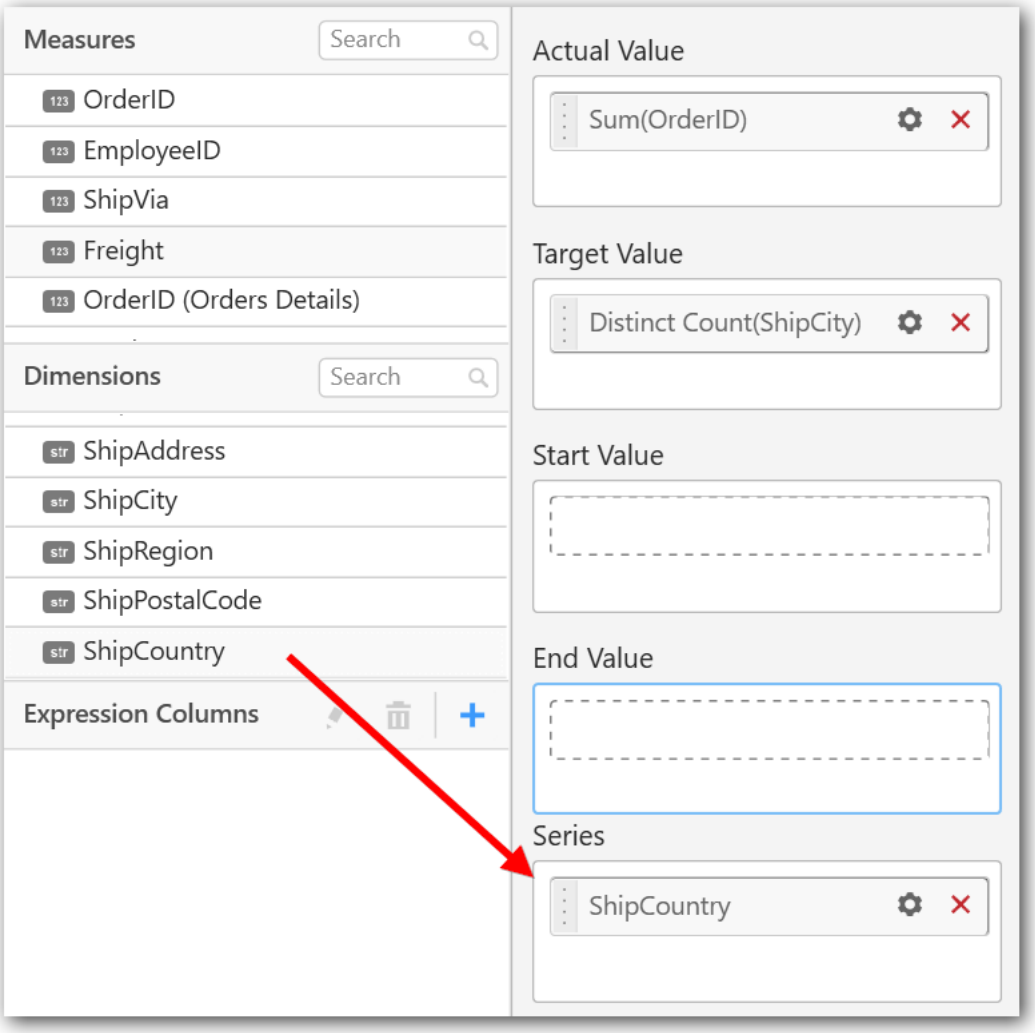
You can clear the filter by selecting the Clear option.



As like Target Value, You can change the Start Value, End Value by changing the filter condition and also You can apply and clear the filter to both Start Value and End Value.



Drag and Drop the elements from sections to Series.



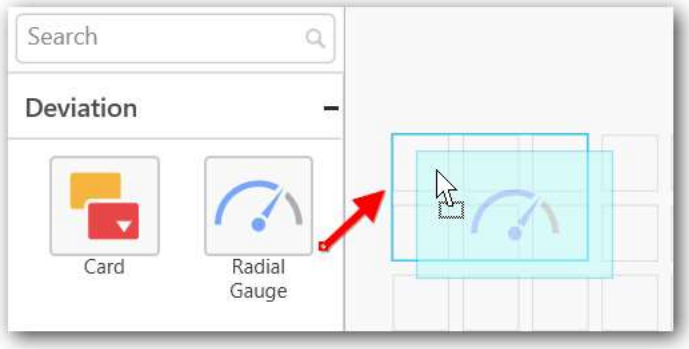
Here, is an illustration

How to configure the SSAS data to Radial Gauge?

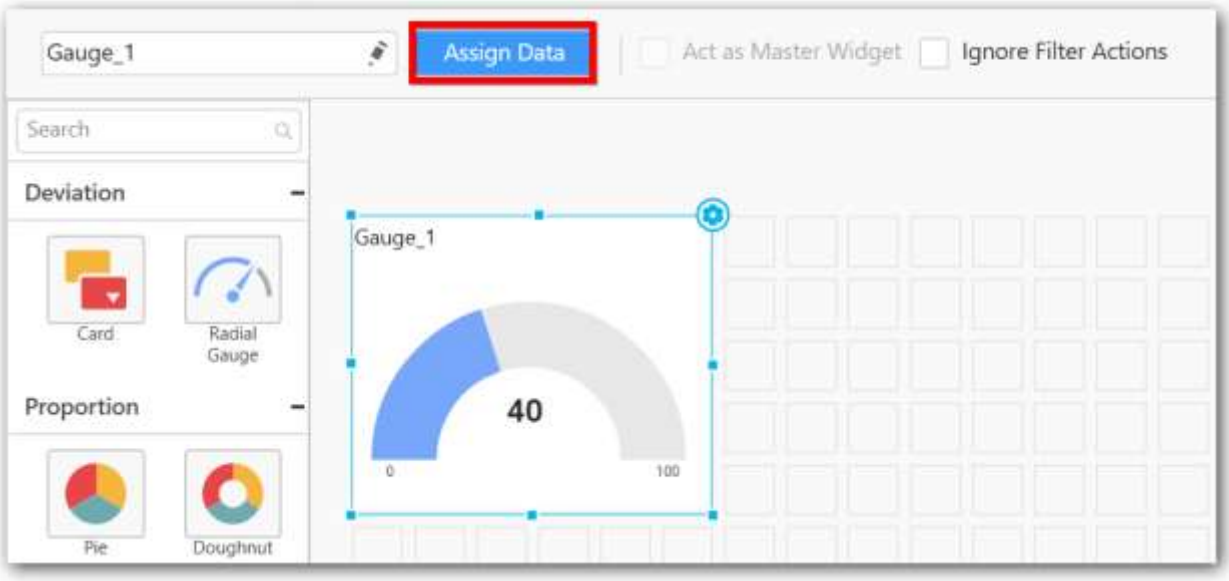
To construct a radial gauge, a minimum requirement of 1 column is needed. You can visualize both measure, calculated measure and dimension column data in radial gauge control.

Following steps illustrates configuration of SSAS data to Radial Gauge.

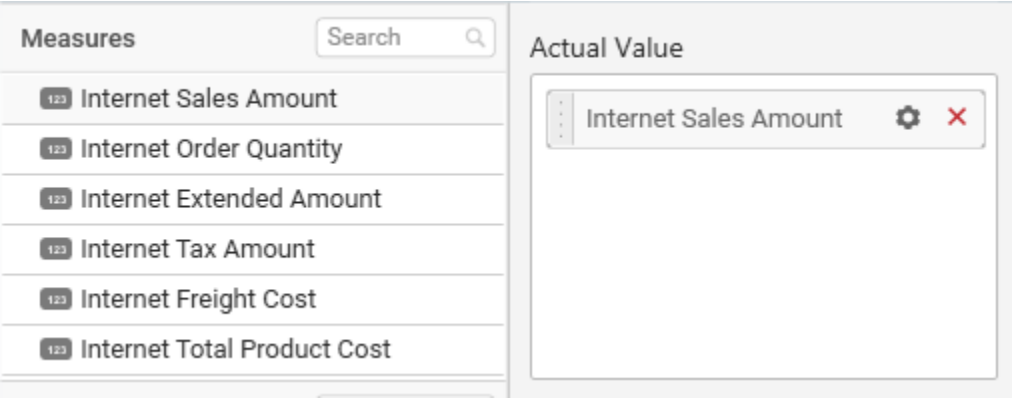
Drag and drop Radial Gauge control icon from the Tool box into design panel. You can find control in Toolbox by search.



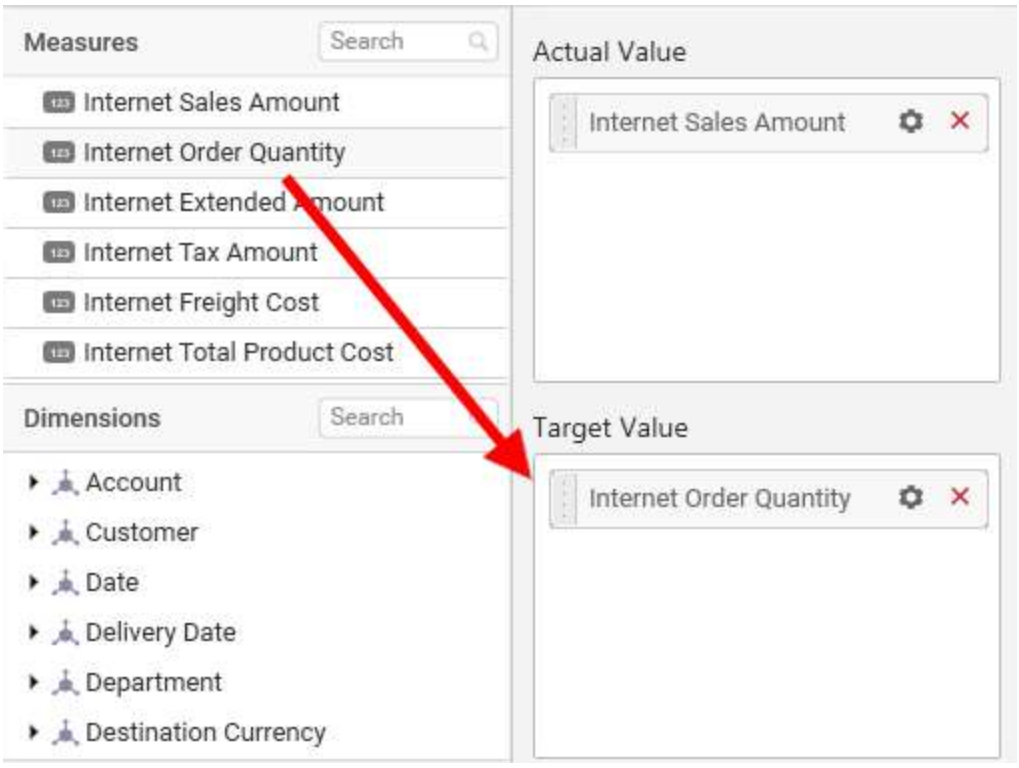
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



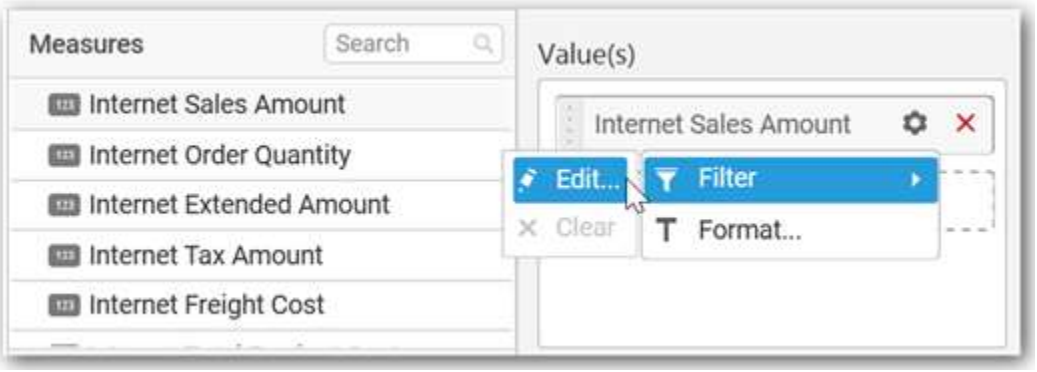
Drag and drop a column under Measures category into Actual Value.



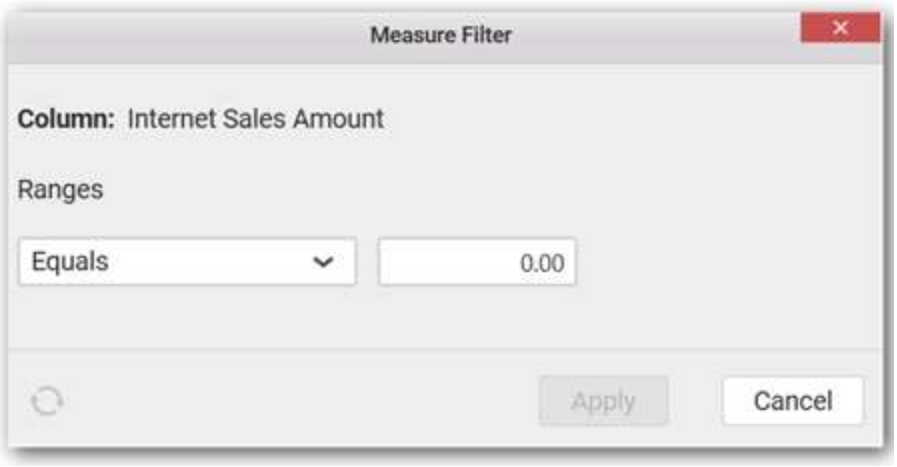
Drag and drop a column under Measures category into Target Value.



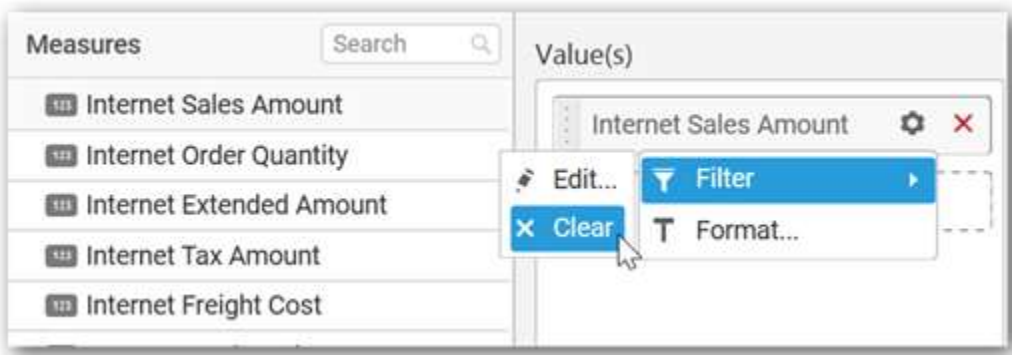
Define the filter criteria to match through choosing Edit option in Filter menu item.



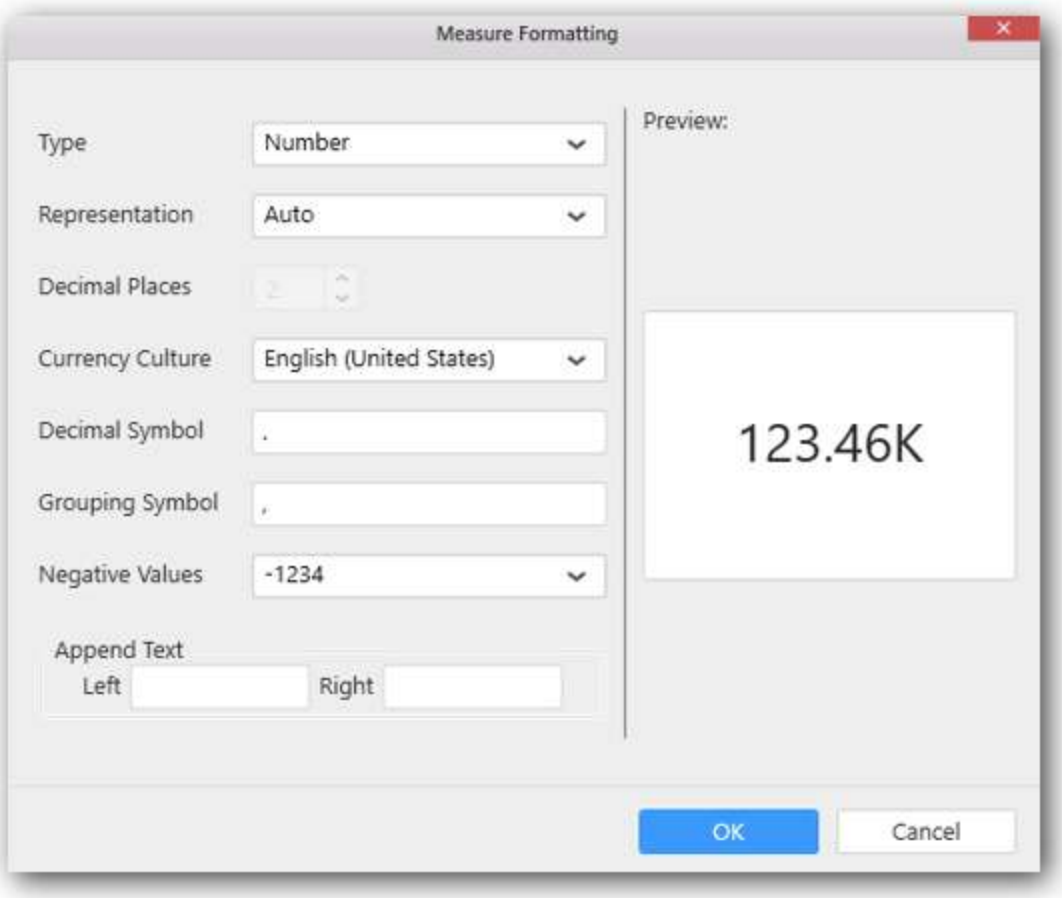
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



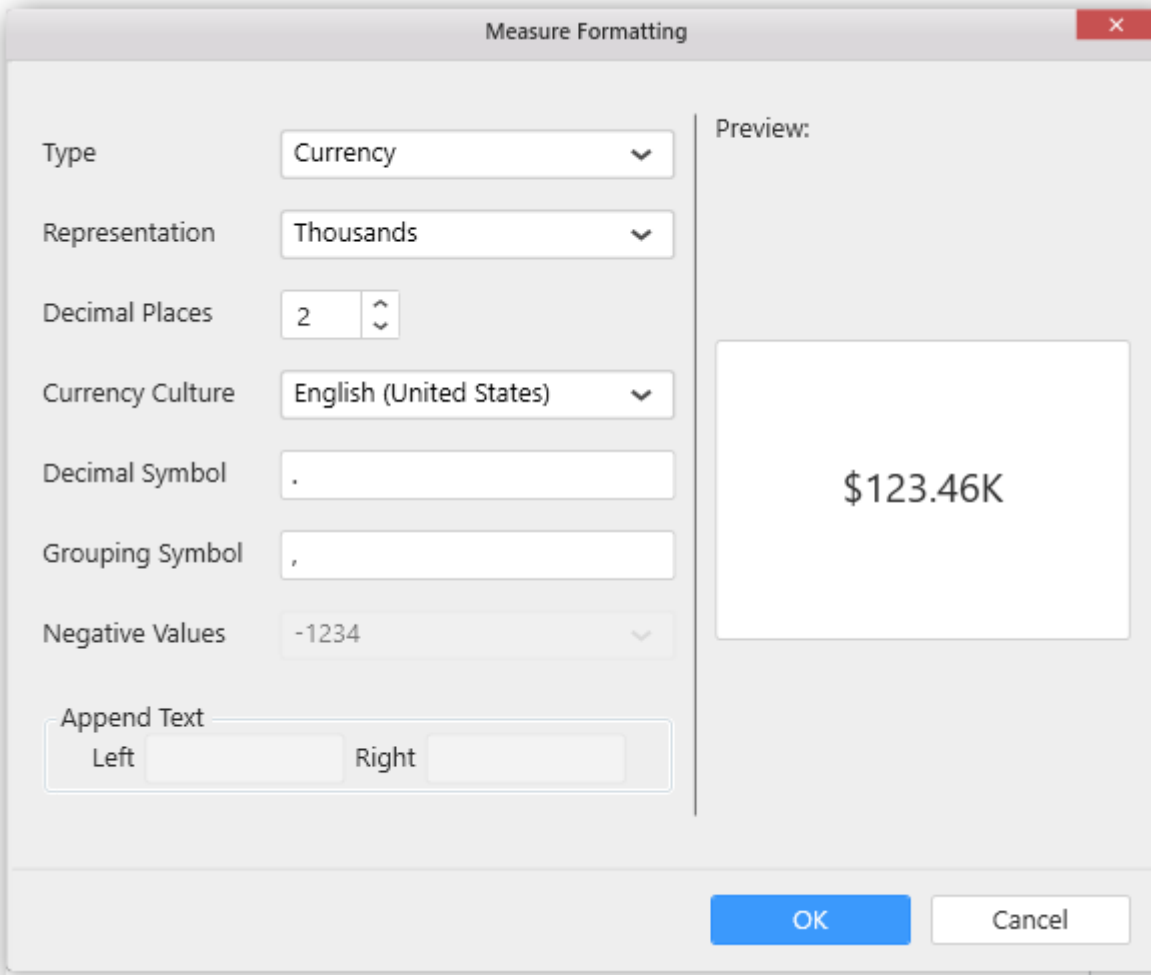
Select Clear option to clear the defined filter.



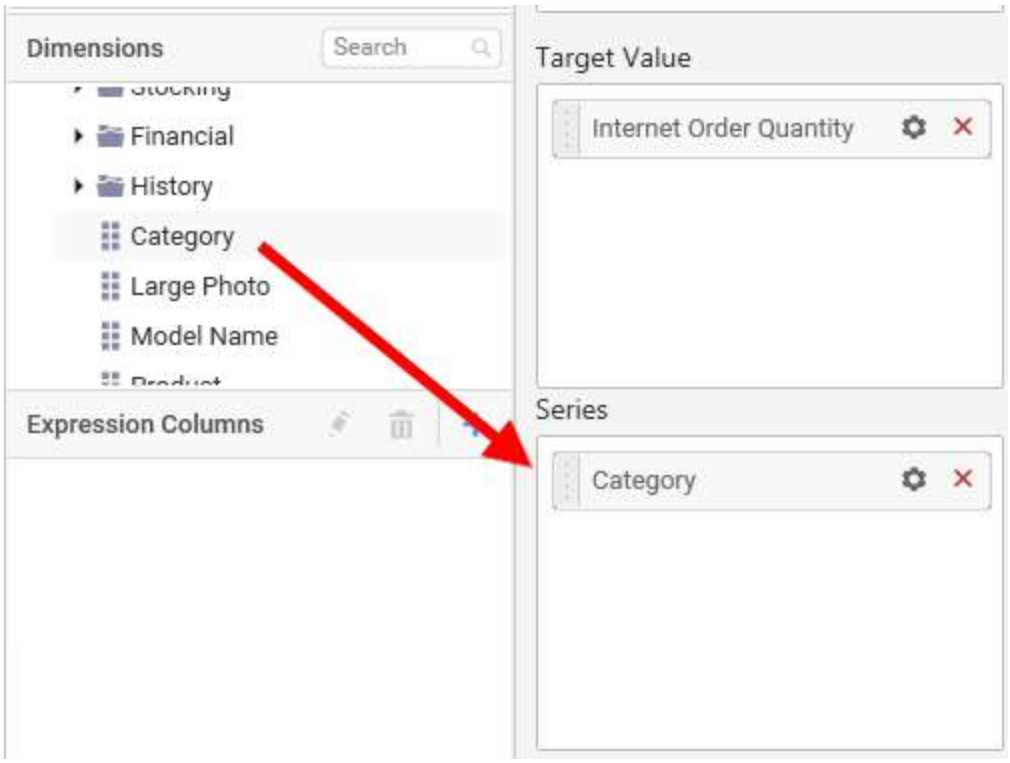
Select Format option to define the display format to the values in the column through Measure Formatting window.



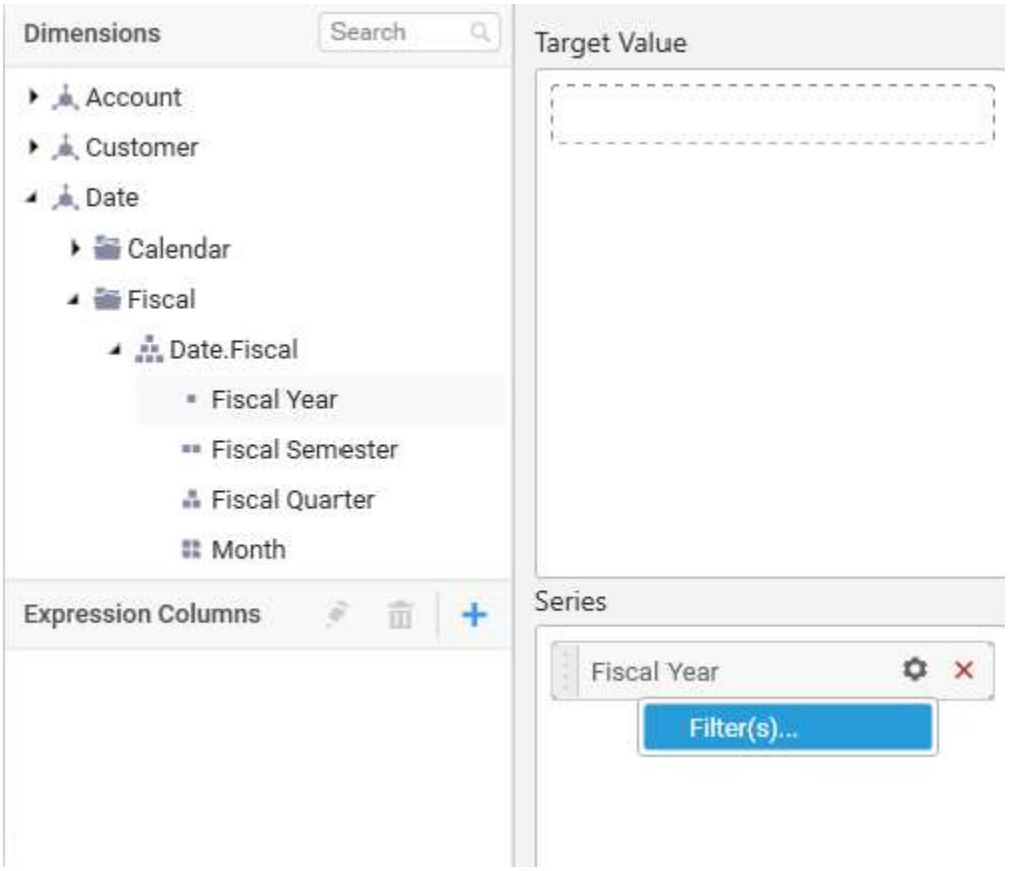
Choose the options you need and click OK.



Add a dimension level or hierarchy into Series section through drag and drop.

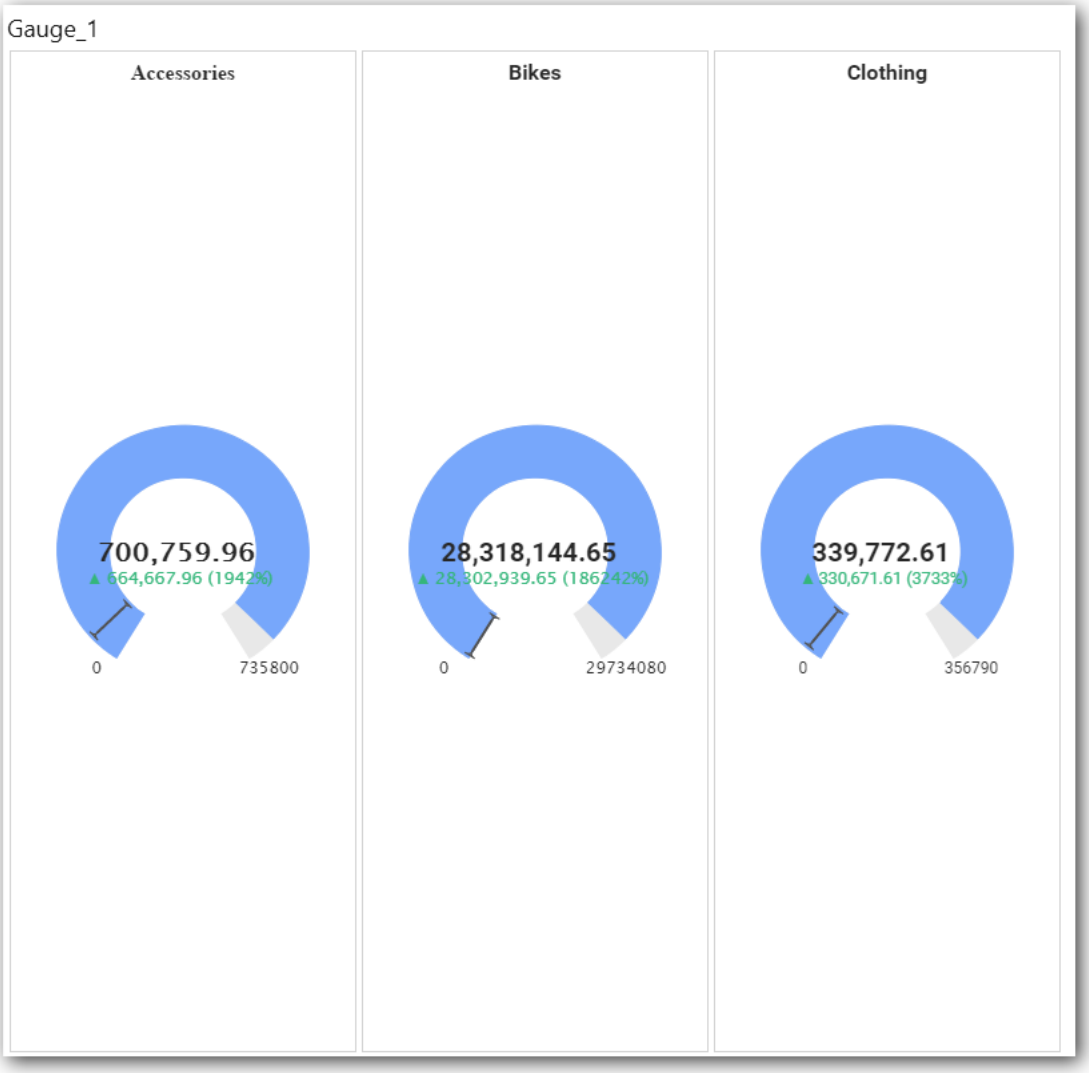


Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



To know more about filters, refer here.

Here, is an illustration



How to format Radial Gauge?

You can format the radial gauge for better illustration of the view that you require, through the settings available in Properties pane.

General Settings

Header

This allows you to set the container header text.

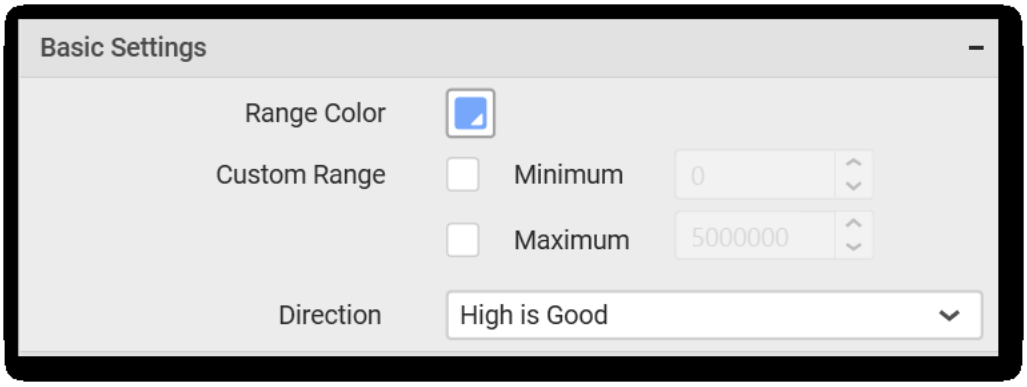
SubHeading

This allows you to set sub-title for this Radial Gauge widget.

Description

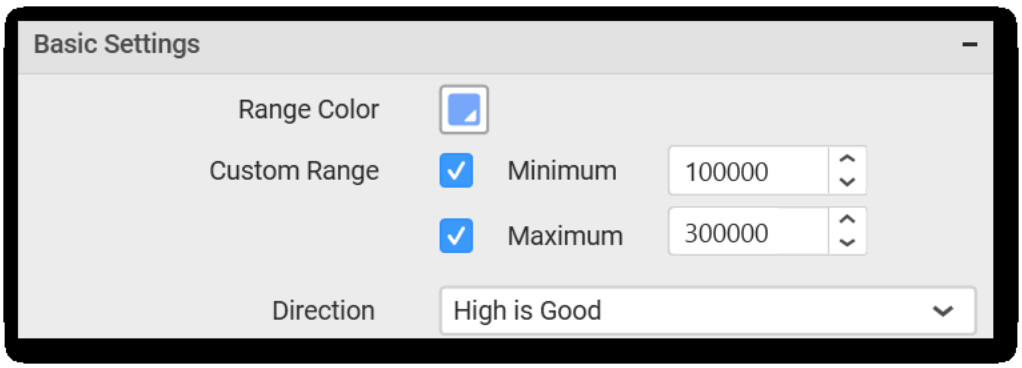
This allows you to set description for this radial gauge widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings



Range Color setting allows you to customize the range color of Gauge widget. Click the color to drop down the list of palette colors and select one to apply to the range.

Custom Range setting allows you to customize the range start and range end values of Gauge widget. Enable the Minimum and/or Maximum value boxes for editing through selecting respective checkbox.



Note: Range Start Value should be always greater than range end value.

Direction setting allows you to define the quality of value through specifying whether higher value should be treated as good or bad.

Filter Settings

Act as Master

This allows you to define this gauge widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard. This option is available only when a column in Series section is configured.

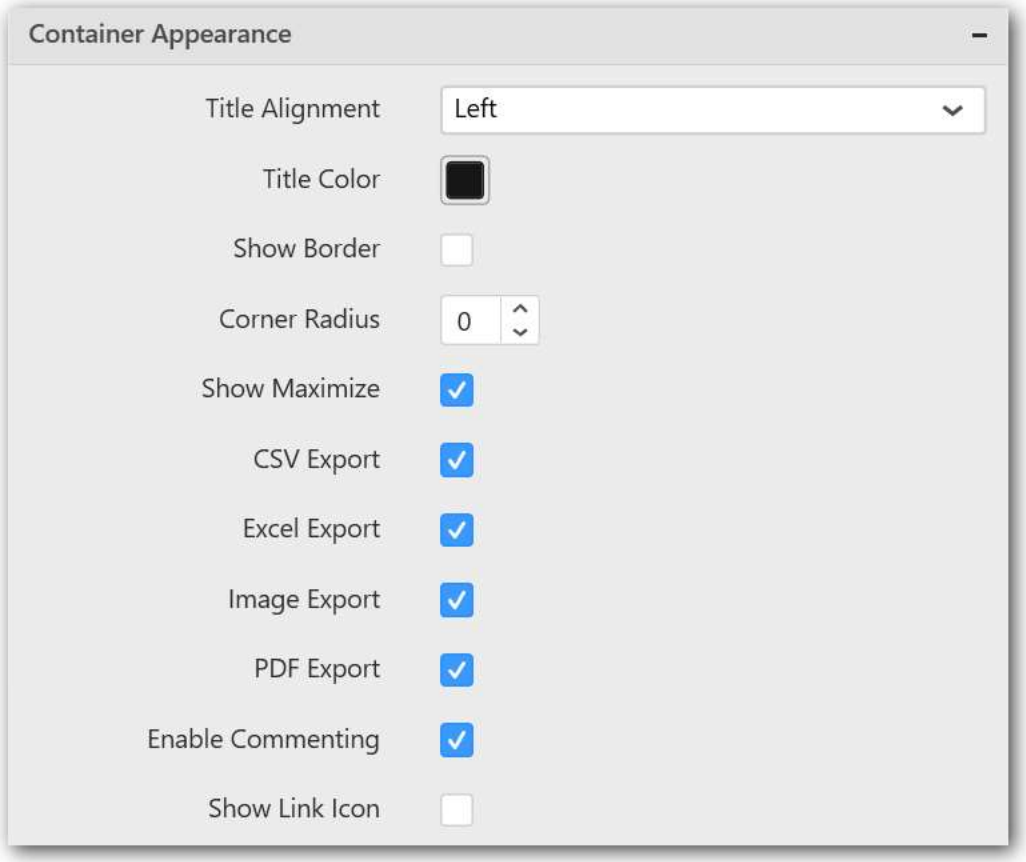
Ignore Filter Actions

This allows you to define this gauge widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings

You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this radial gauge widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the radial gauge widget.

CSV Export

This allows you to enable/disable the CSV export option for this radial gauge widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this radial gauge widget. Enabling this allows you to export the summarized data of the widget view to XLSX format.

Image Export

This allows you to enable/disable the image export option for this radial gauge widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Card

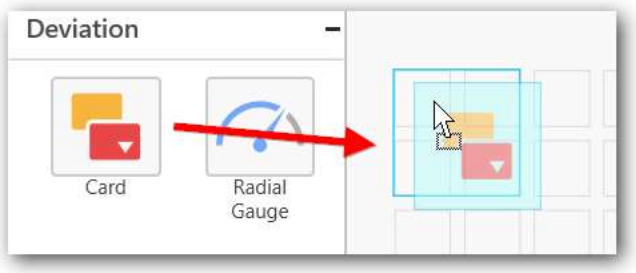
Card allows you to measure trends through key performance indicators (KPIs) like Value and Goal.

How to configure flat table data to Card Widget?

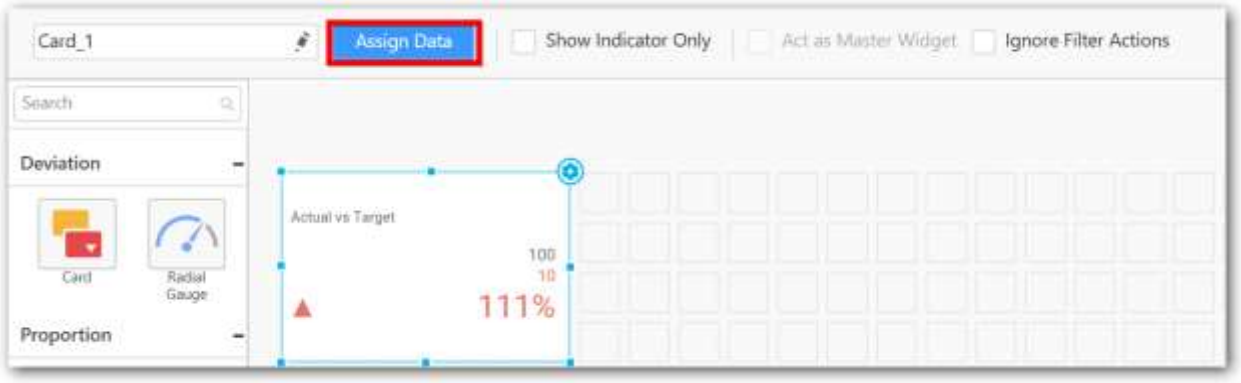
To showcase a card, a minimum requirement of 1 actual and/or target value is needed.

The following procedure illustrates data configuration of Card.

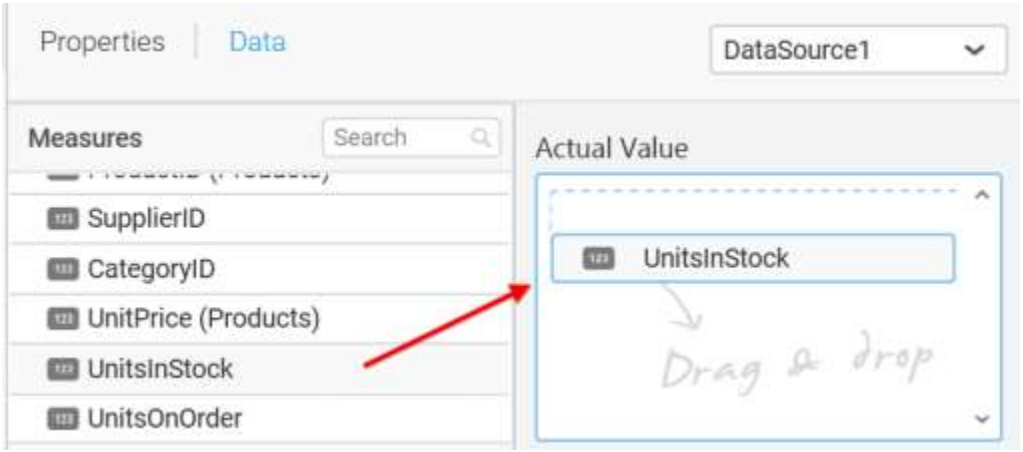
Drag and drop Card control icon from the Tool box into design panel. You can find control in Toolbox by search.



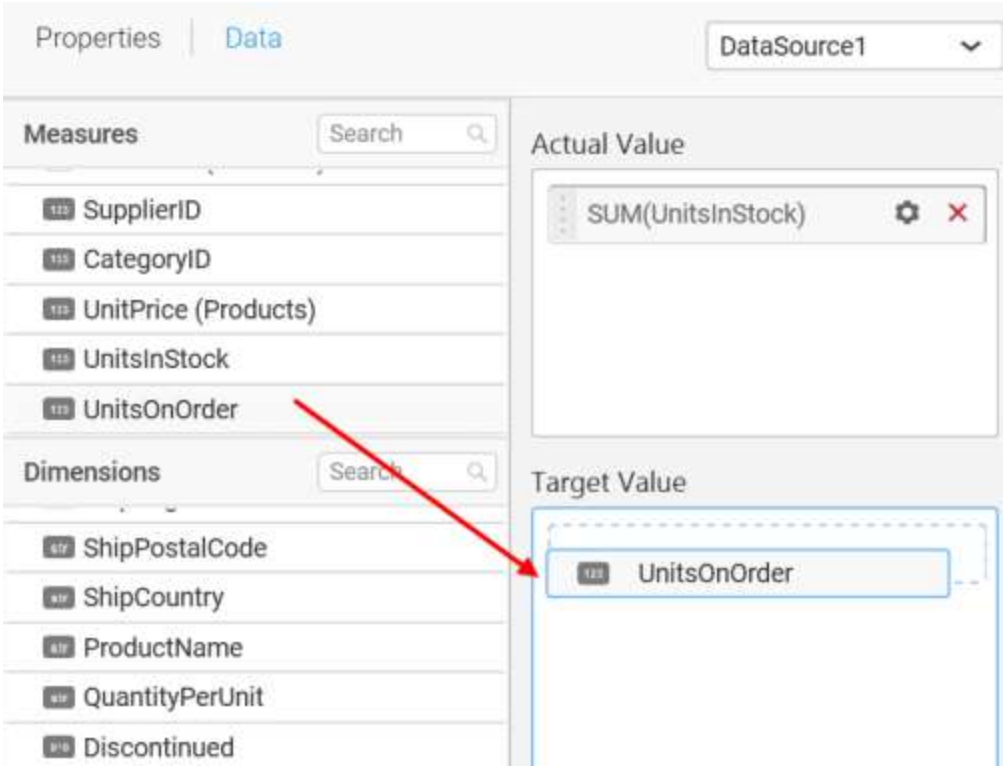
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



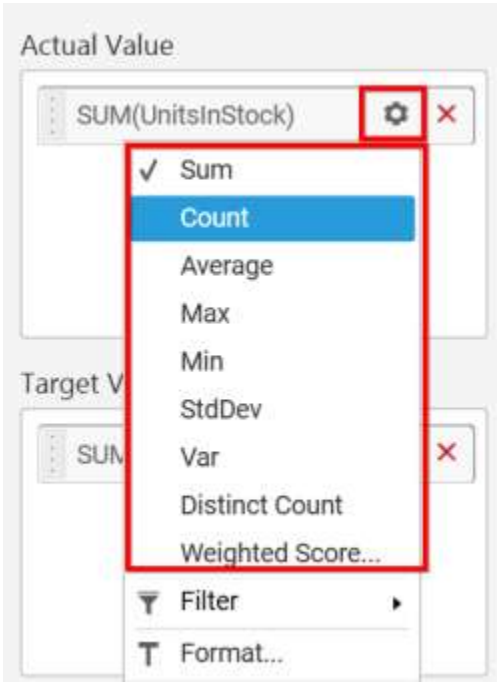
Bind column through drag and drop element from Measures section to Actual Value.



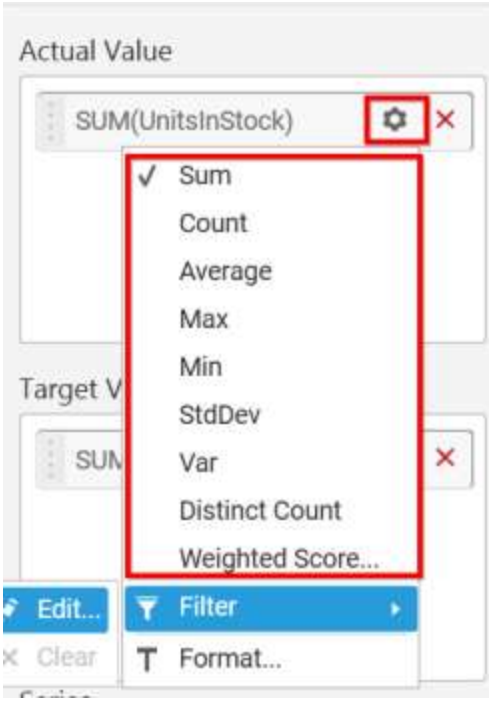
Drag and Drop the elements to Target Value.



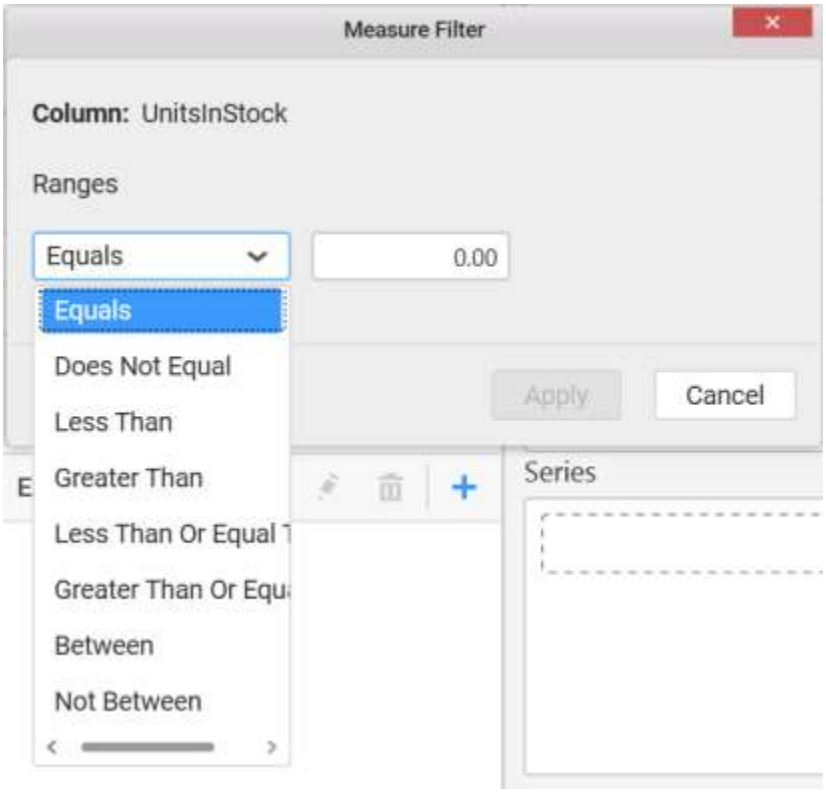
You can use aggregate function to change the Actual Value of the card.



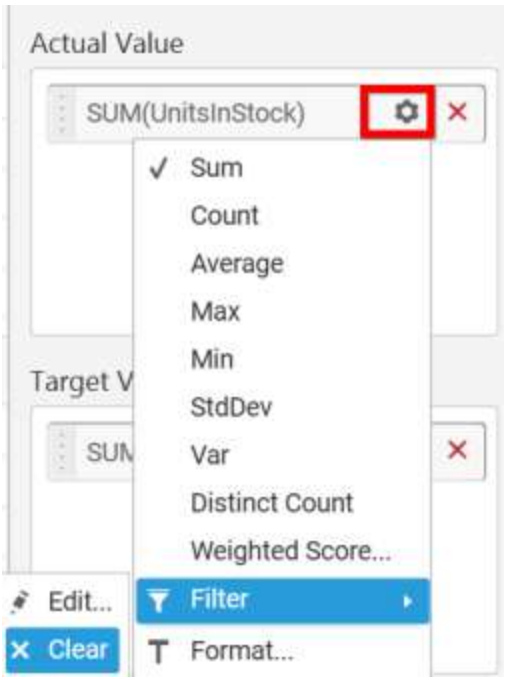
You can use Filter option to filter the data by specifying the filter condition.



Measure Filter window will be shown to set the Ranges.

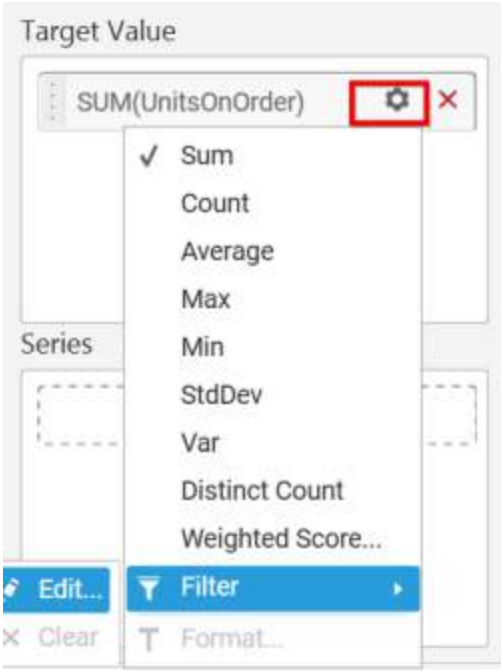


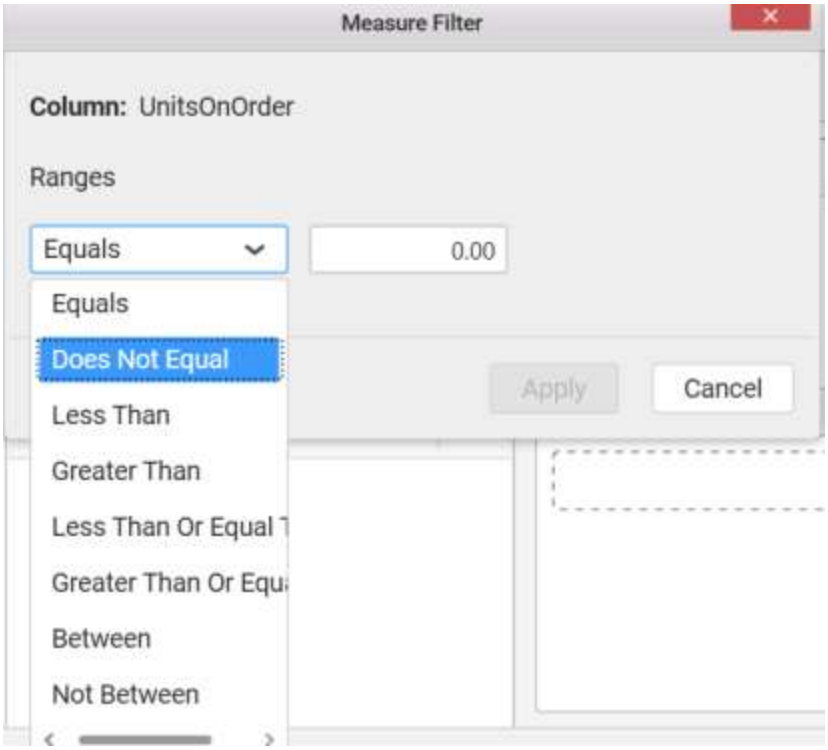
You can clear filter by selecting the Clear option for Actual Value.



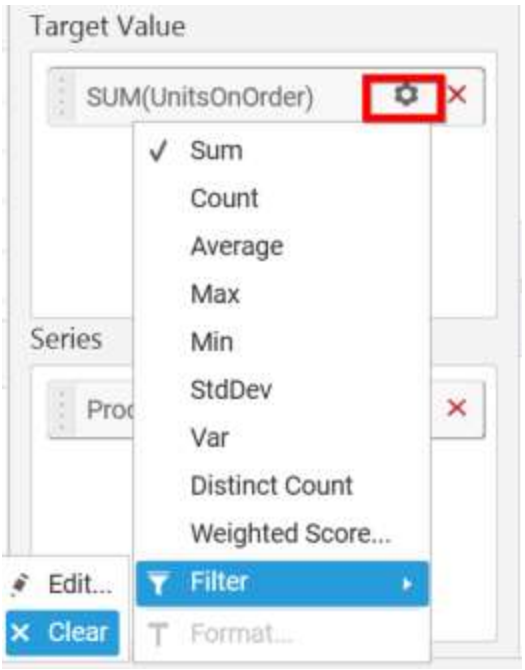
You can the change the Target value by selecting aggregate function.

You can use Filter option to filter the data by specifying the filter condition.

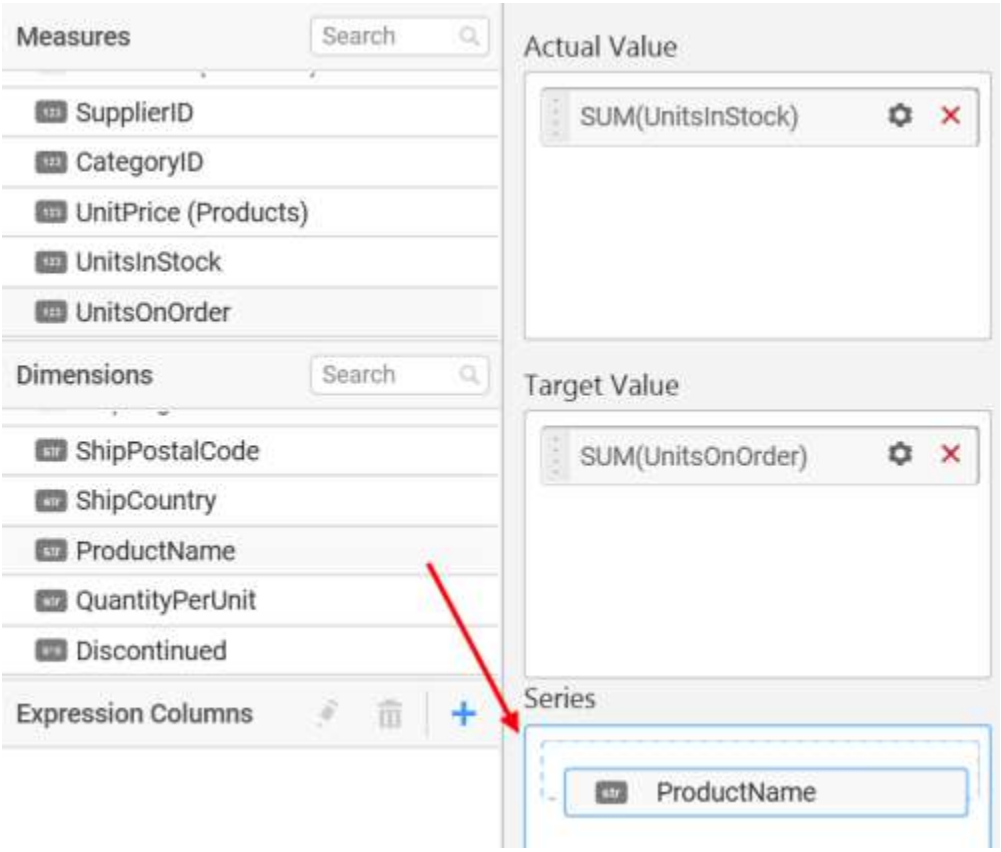




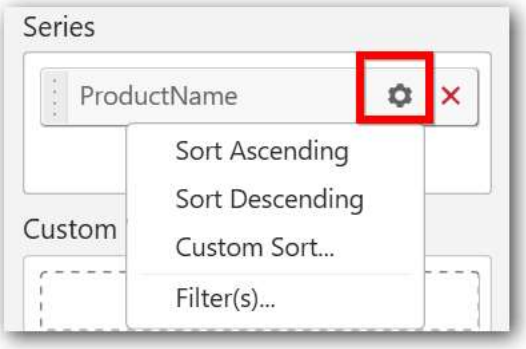
You can clear filter by selecting the Clear option for Target Value.



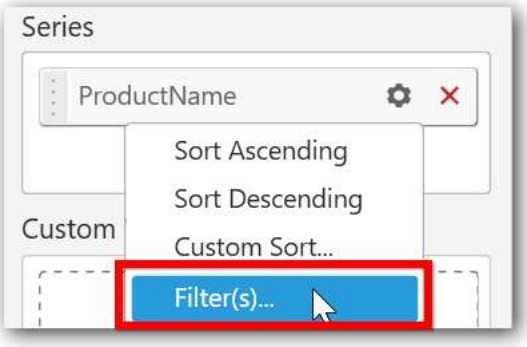
Drag and Drop the elements from sections to Series.



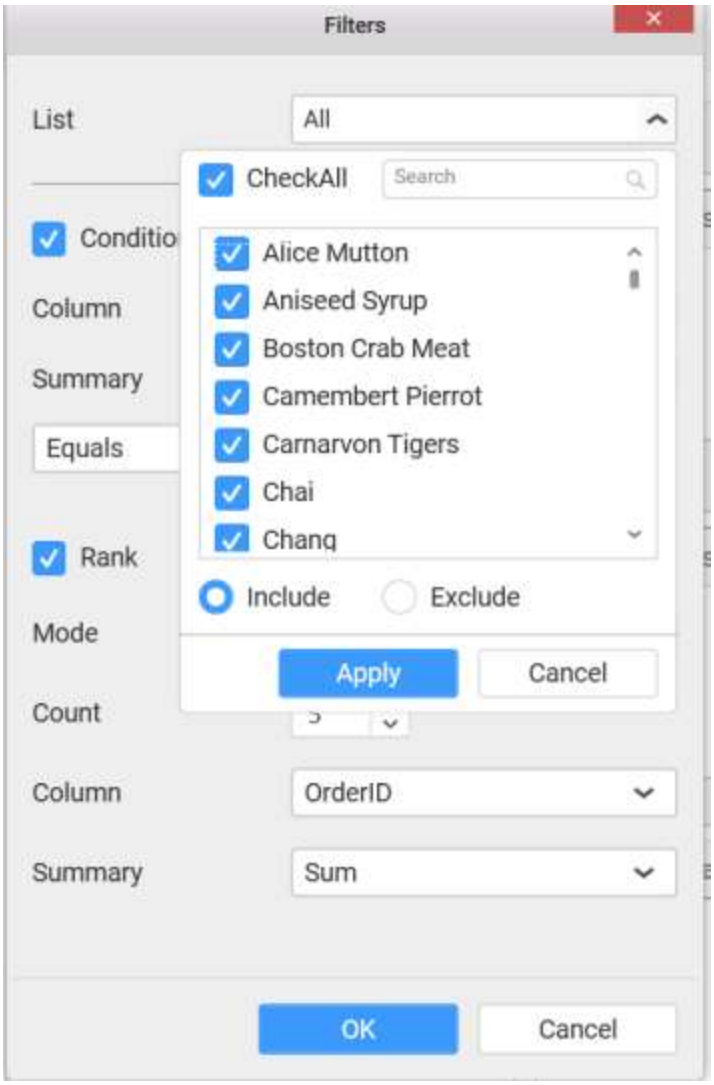
You can change the Series value of the card by changing the setting.



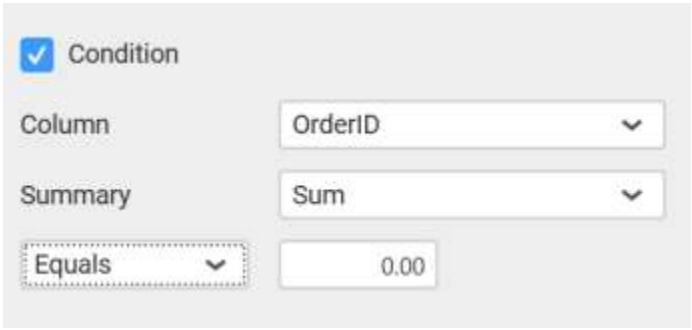
You can use the filters by selecting the Filter(s)... option to rank the elements.

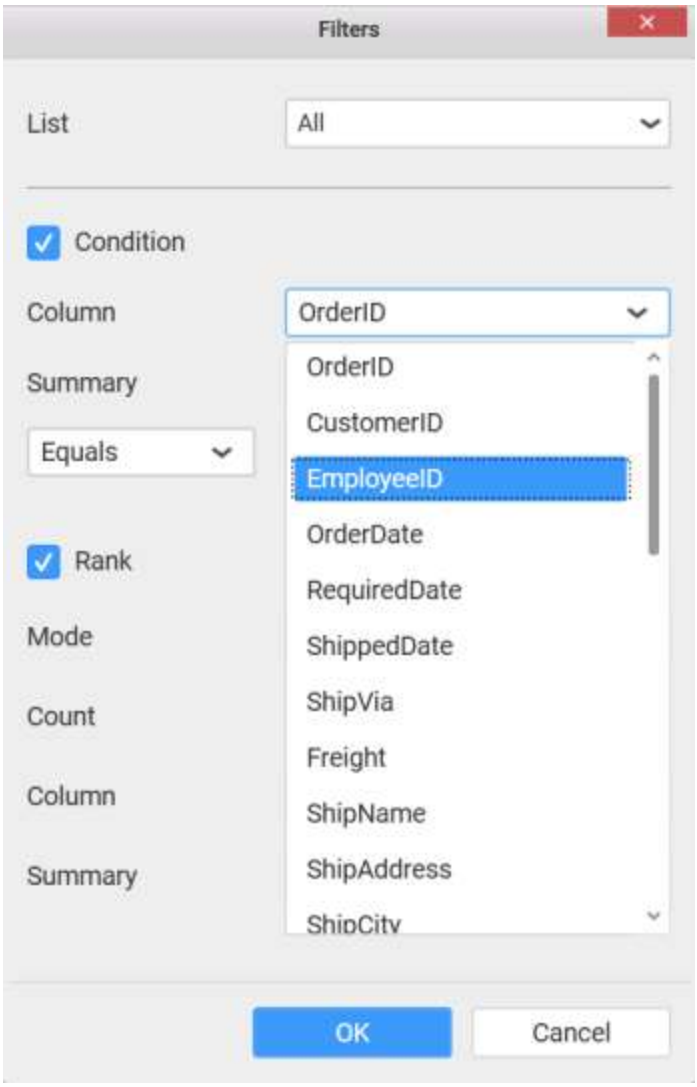


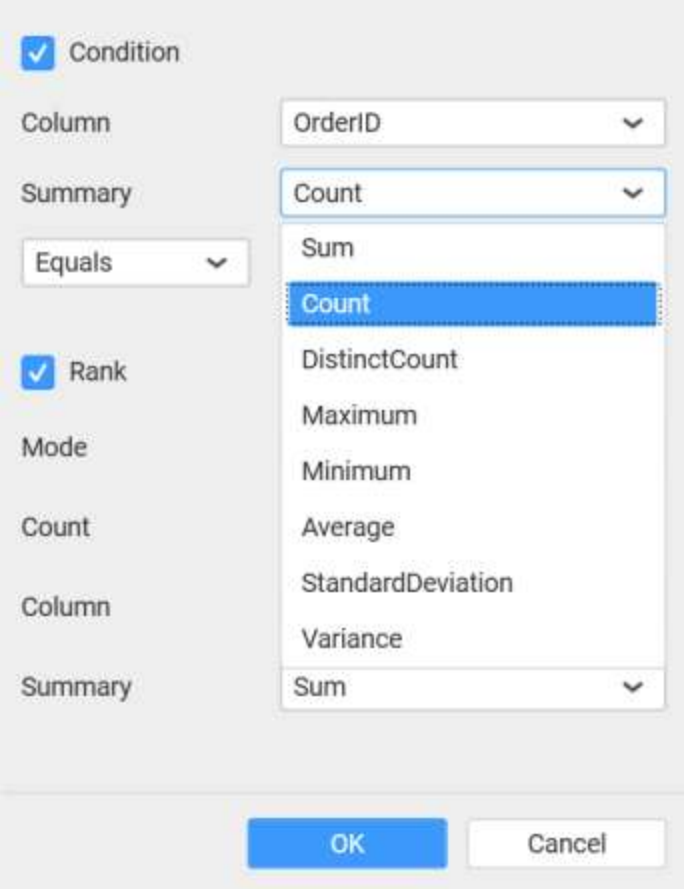
You can select the specific country to filter the element and CheckAll is used either to check all the data or to select the specific data. Include and Exclude is used to include and exclude the elements by selecting the radio button and click the Apply button.



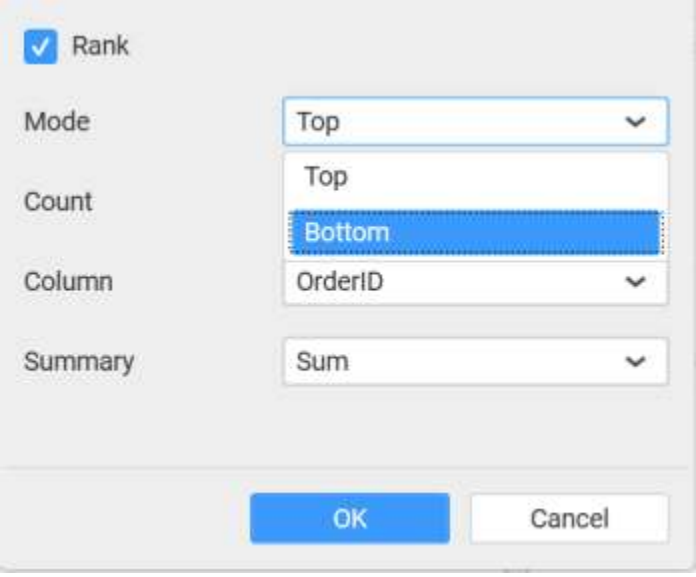
You can select the Condition option to change the Column elements and Summary type by selecting required column name and summary type.



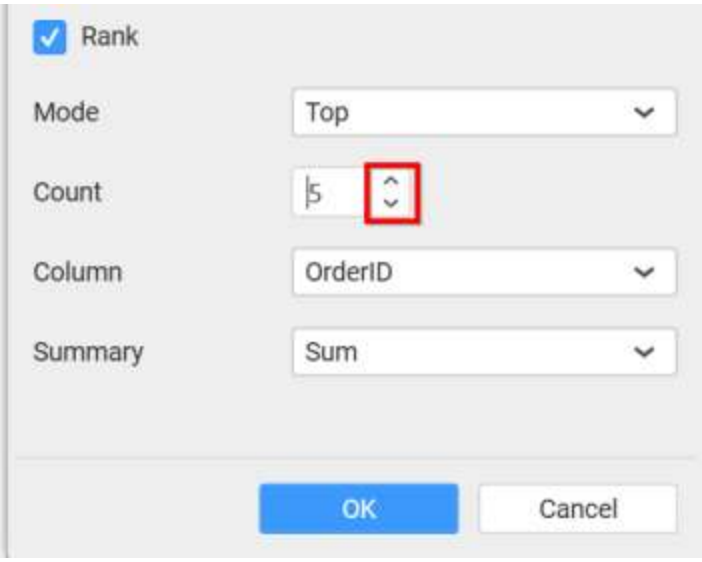




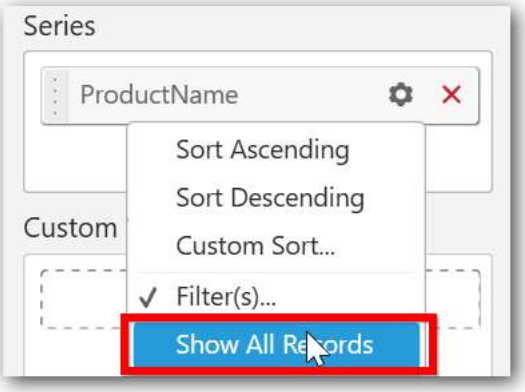
You can select the Rank option to enable filters and select the Mode either top or bottom.



You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.

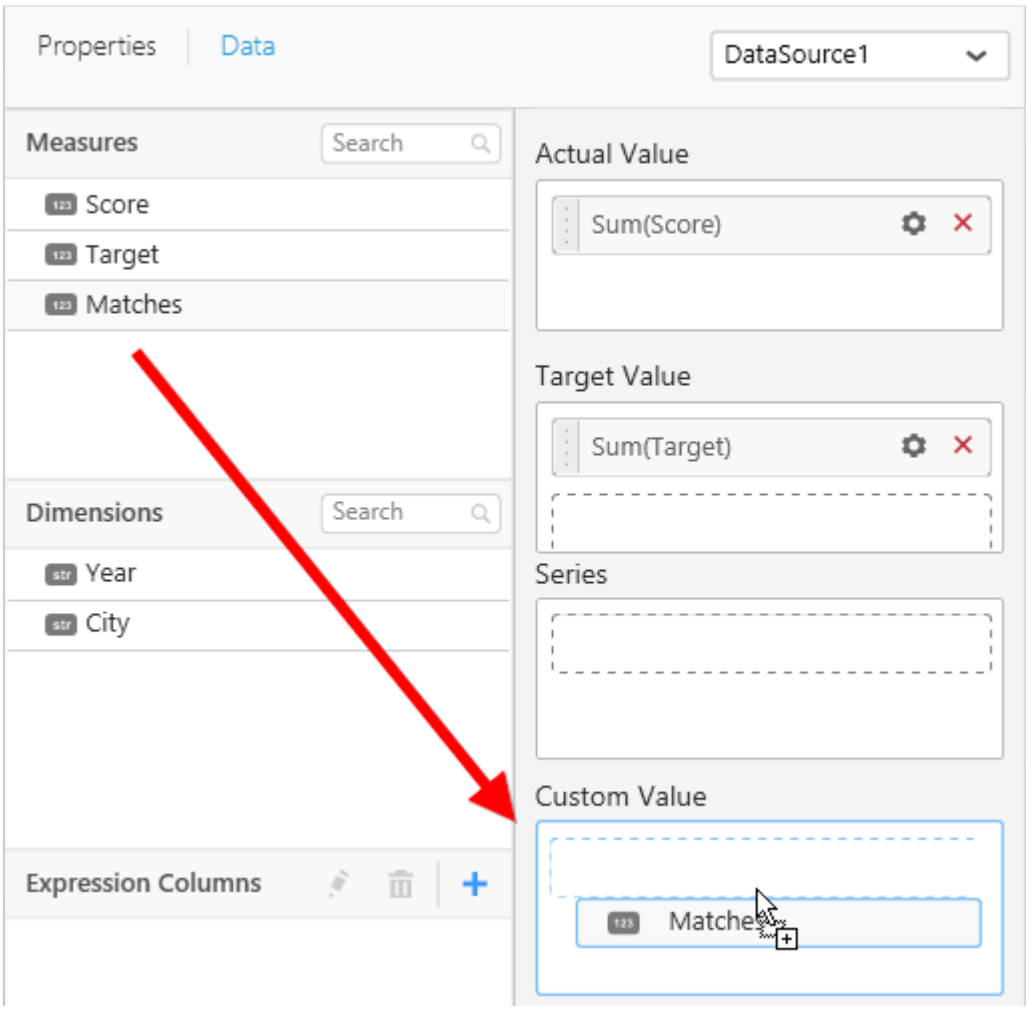


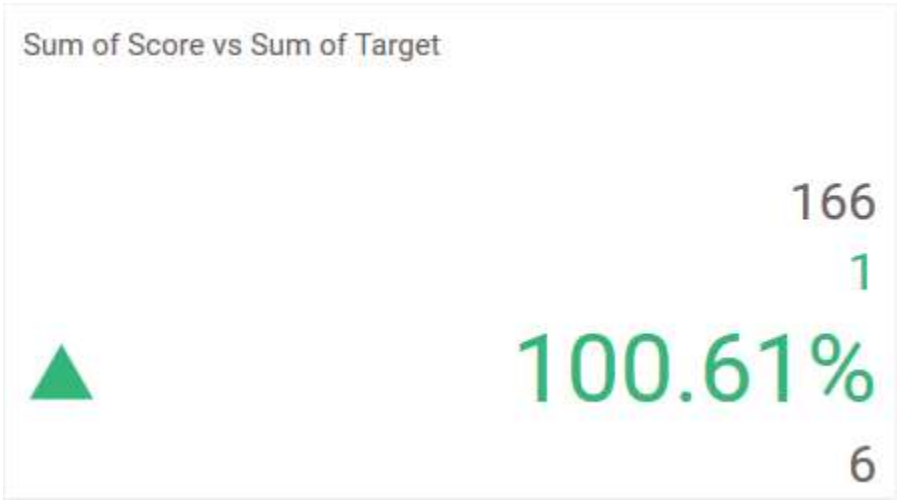
You can clear filter by selecting Show All Records for Series section.



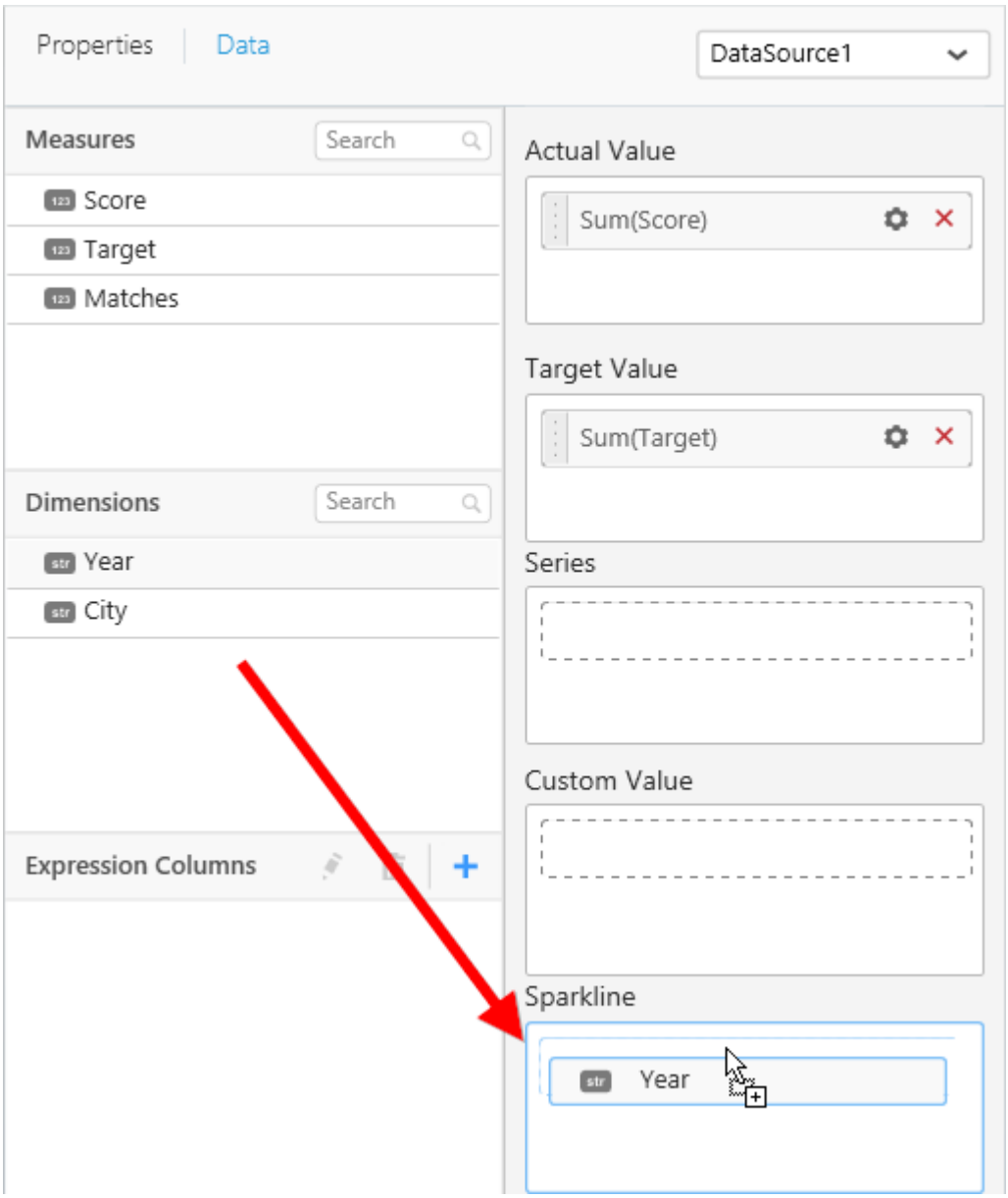
Here is an illustration,

You can add an additional value to the card if needed. Drag and Drop the item to Custom Value section.



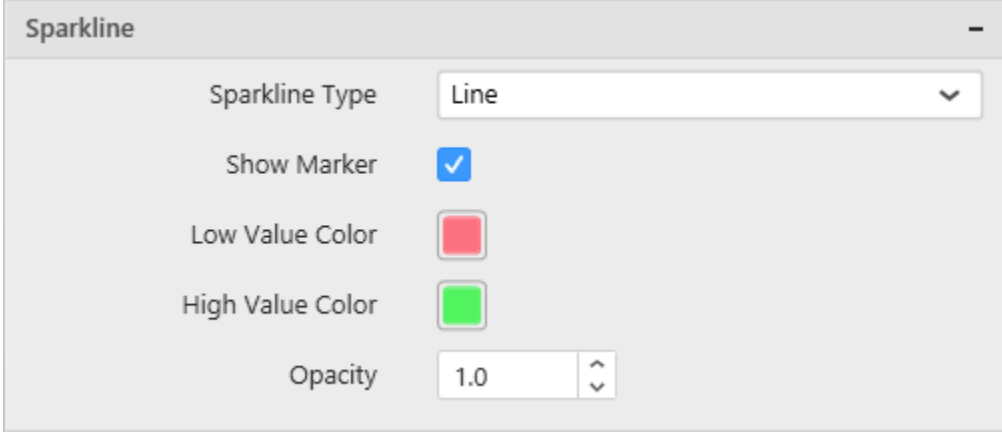


To showcase Sparkline in card widget, drag and drop a dimension field into the Sparkline section.



Sparkline shows the actual value trend with respect to added field.

You can customize the properties of Sparkline



Sparkline Type - Allows to change the type of Sparkline among Line, Area and Column

Show Marker - Allows to enable/disable the marker displayed on Sparkline

Low Value Color - Sets the color of marker for the least value

High Value Color - Sets the color of marker for the highest value

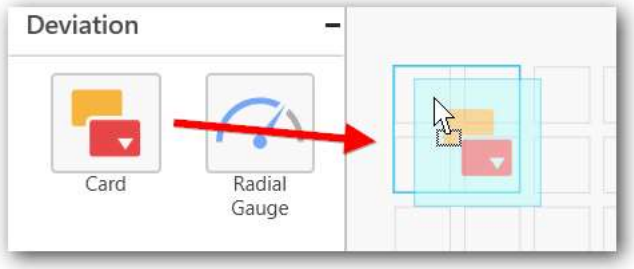
Opacity - Sets the opacity of Sparkline

How to configure SSAS data to Card Widget?

To showcase a card, a minimum requirement of 1 actual and/or target value is needed.

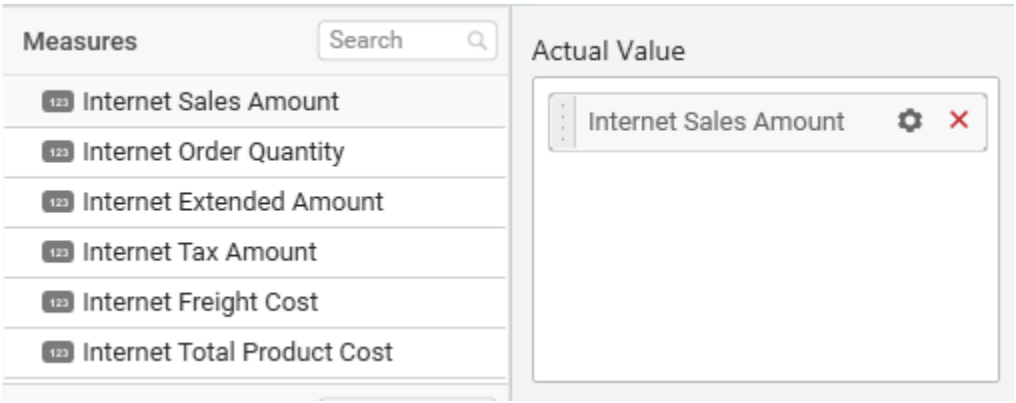
Following steps illustrates configuration of SSAS data to Card.

Drag and drop Card control icon from the Tool box into design panel. You can find control in Toolbox by search.

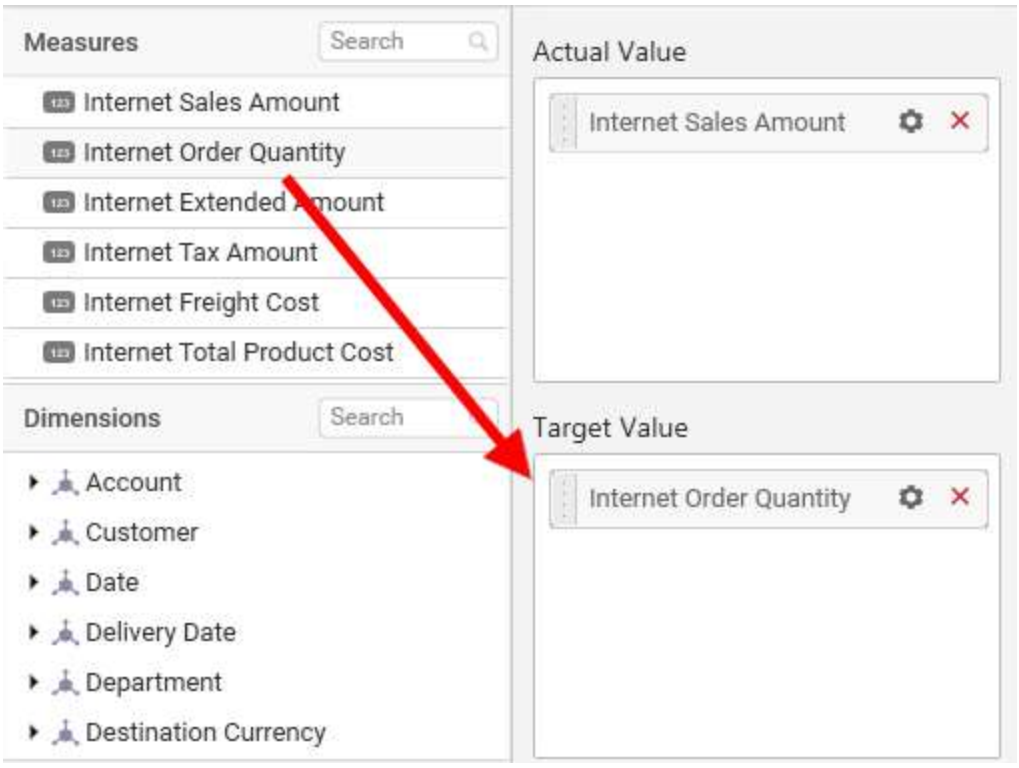


After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.

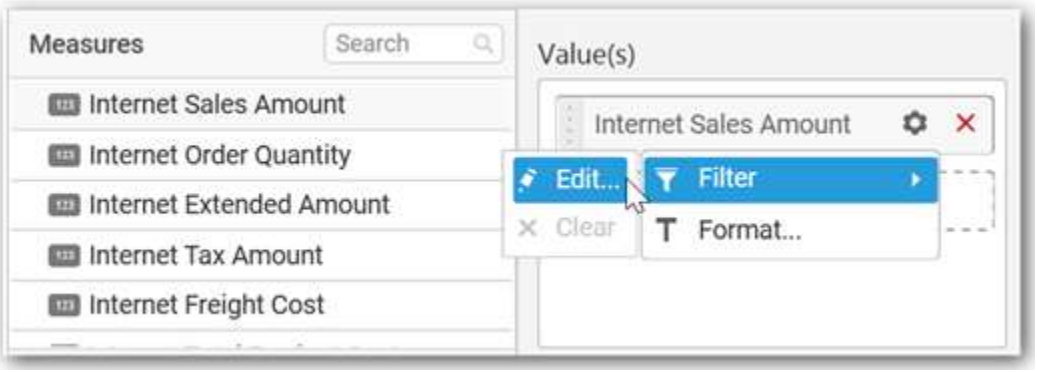
Drag and drop a column under Measures category into Actual Value.



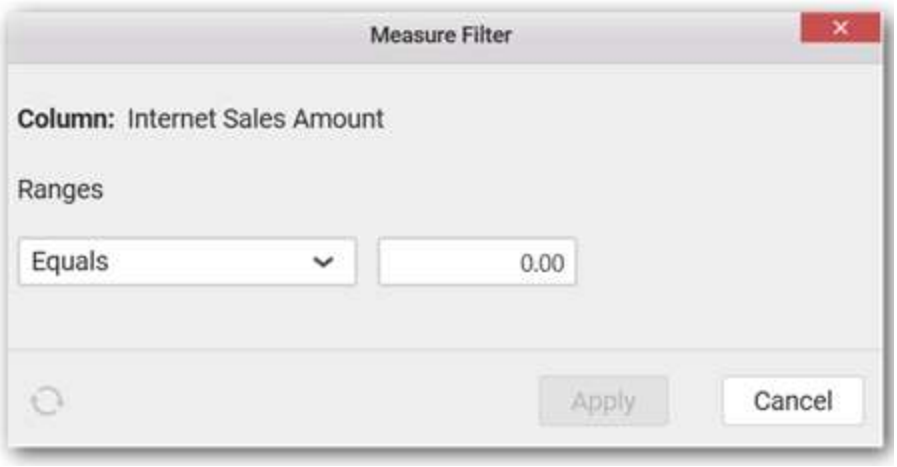
Drag and drop a column under Measures category into Target Value.



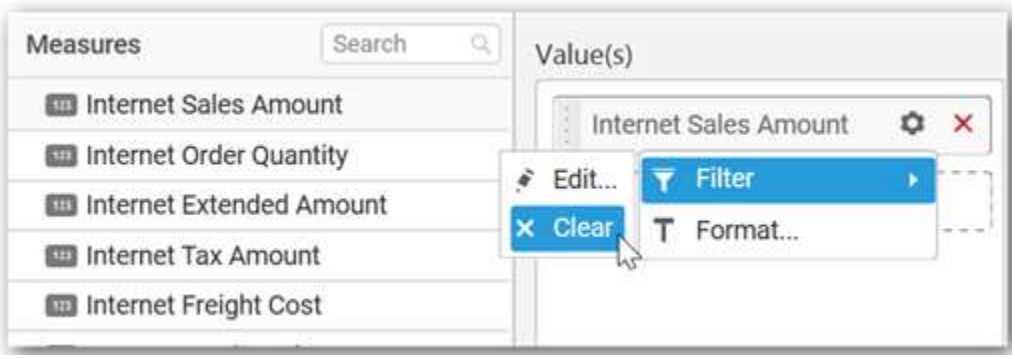
Define the filter criteria to match through choosing Edit option in Filter menu item.



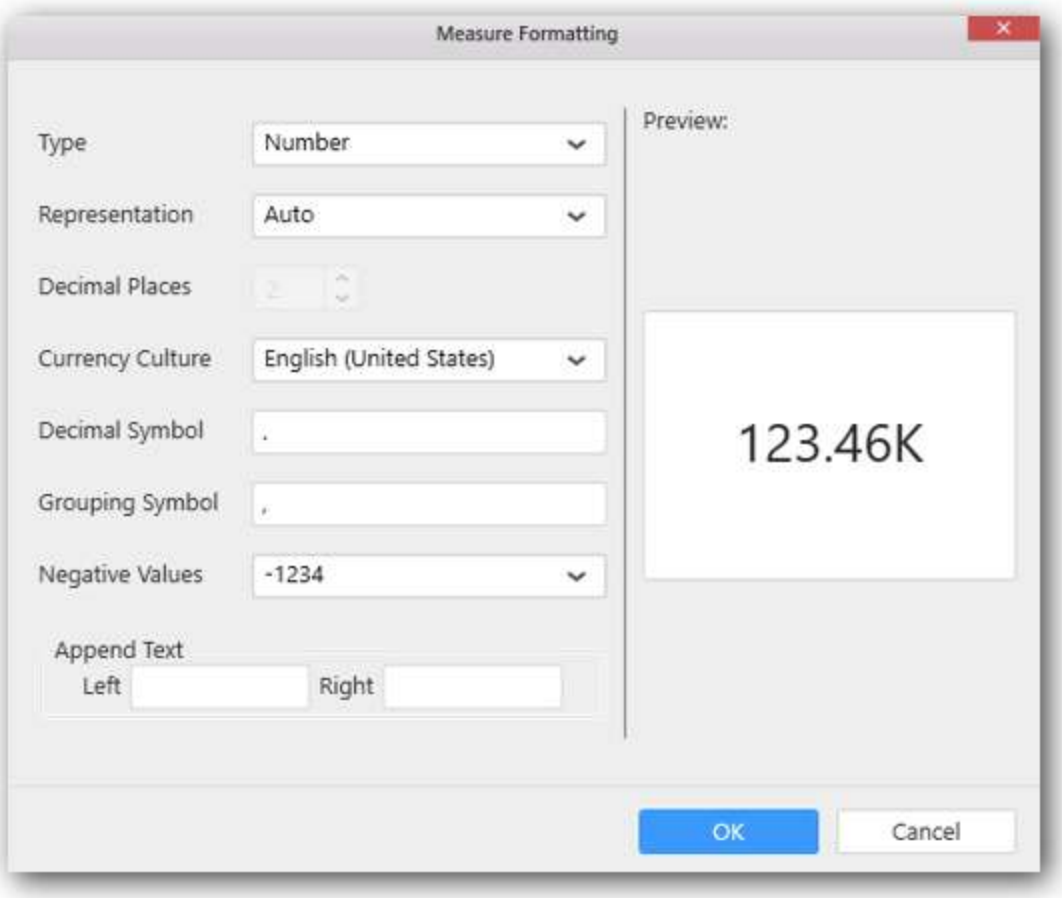
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



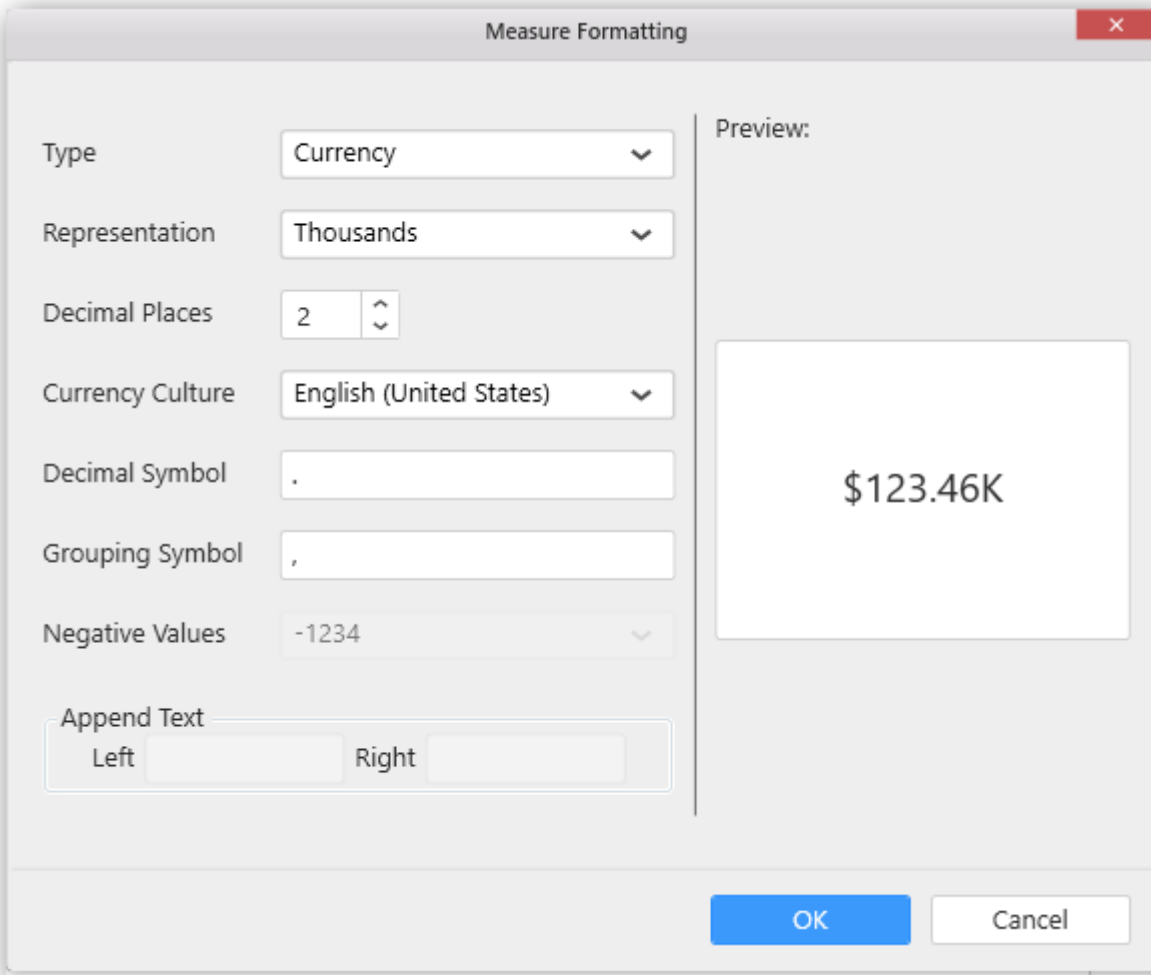
Select Clear option to clear the defined filter.



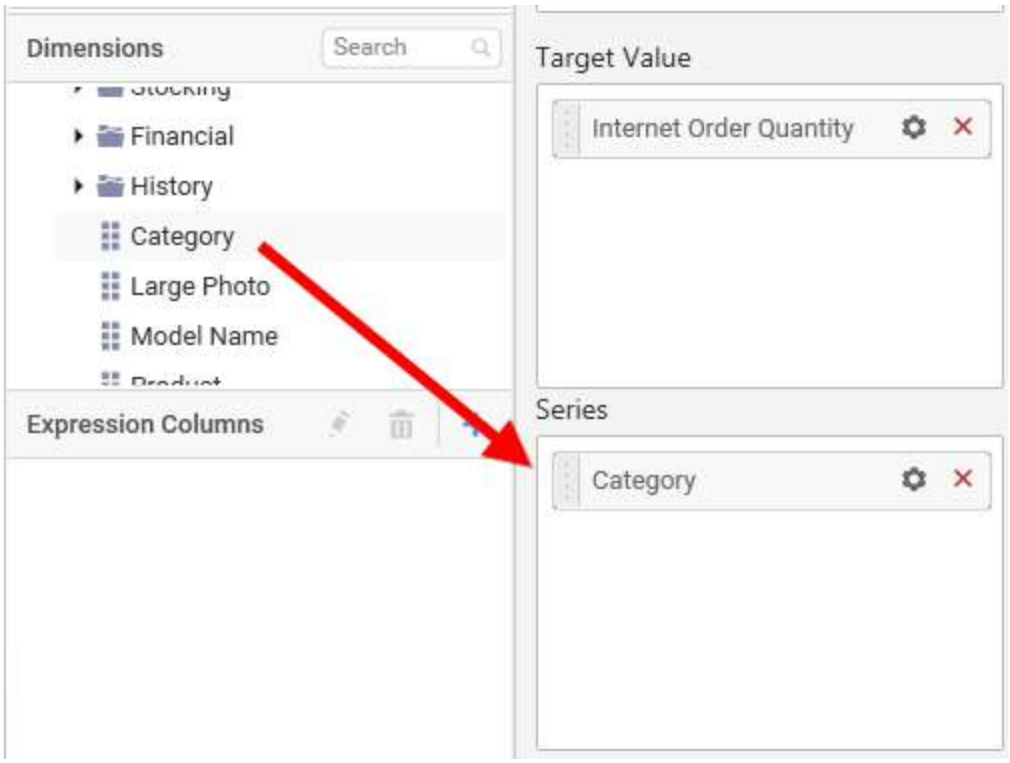
Select Format option to define the display format to the values in the column through Measure Formatting window.



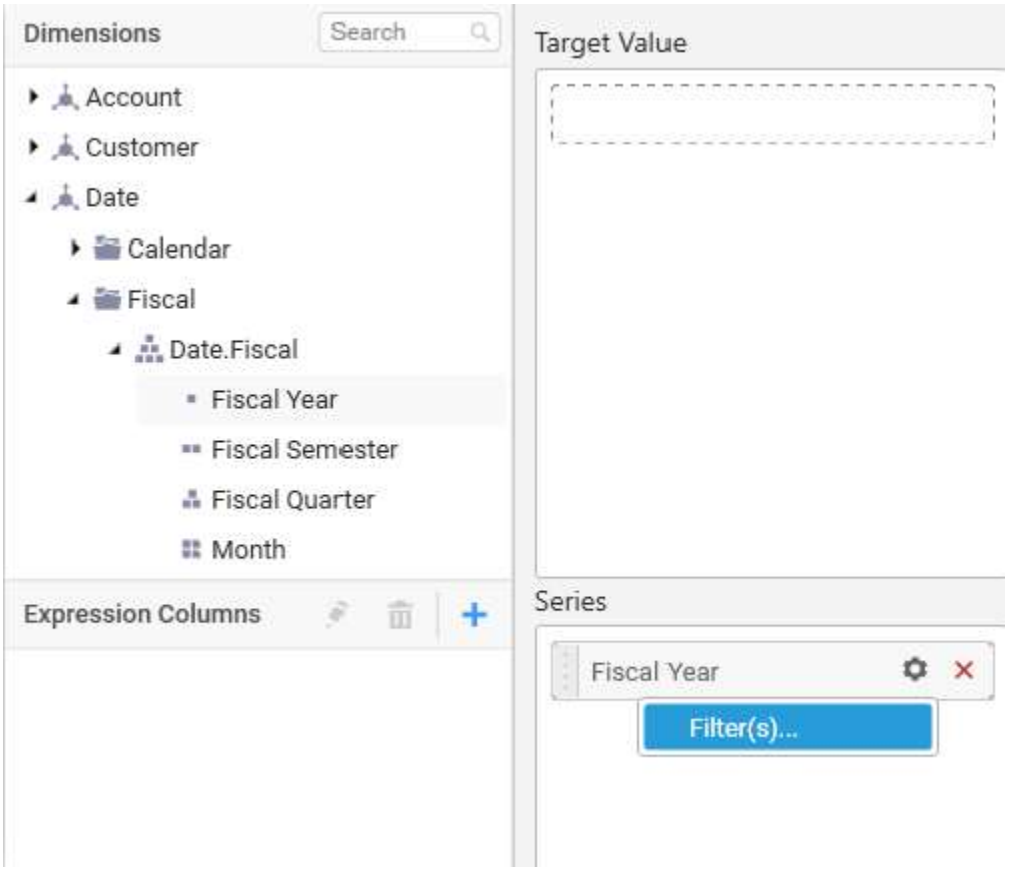
Choose the options you need and click OK.



Add a dimension level or hierarchy into Series section through drag and drop.

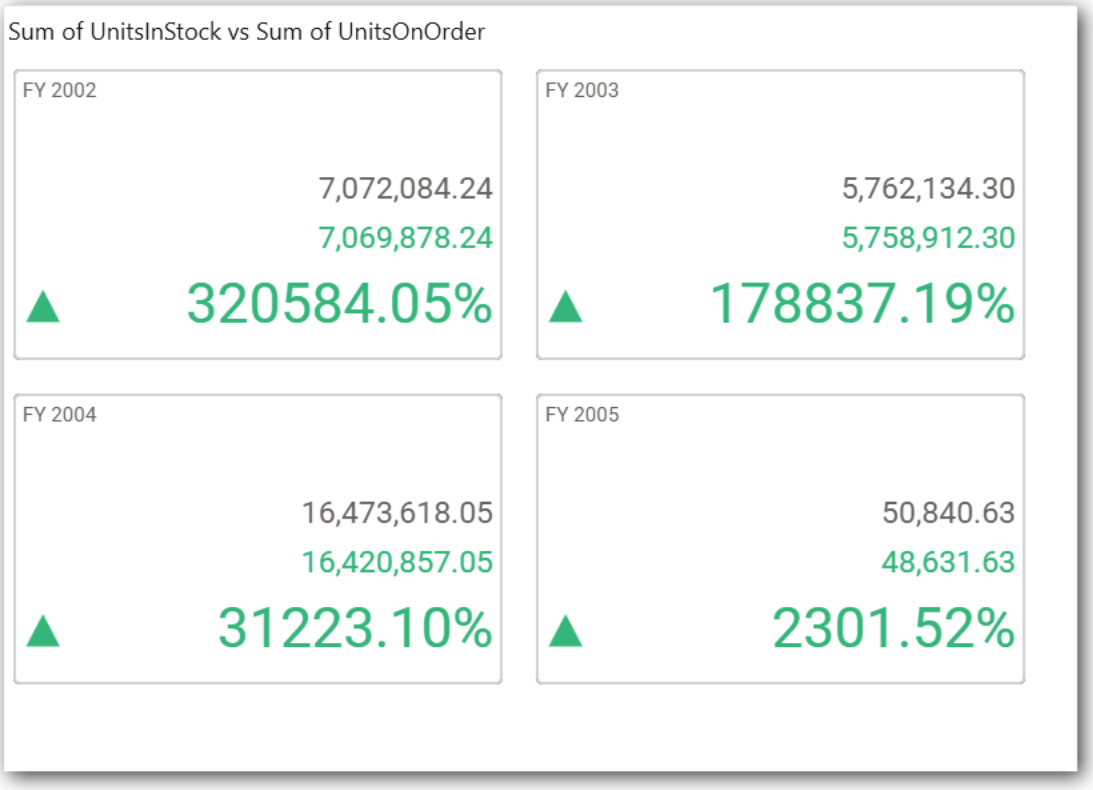


Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



To know more about filters, refer here.

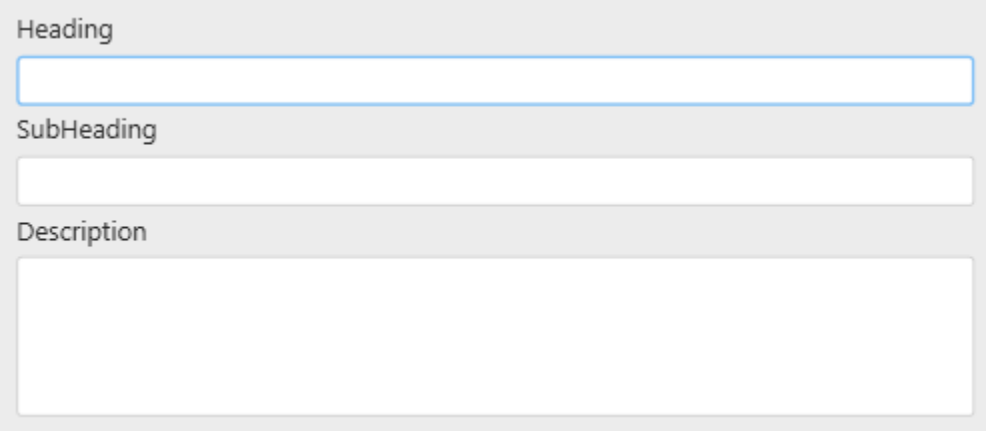
Here is an illustration,



How to format Card Widget?

You can format the card for better illustration of the view that you require, through the settings available in Properties pane.

General Settings



Heading

This allows you to set title for this card widget.

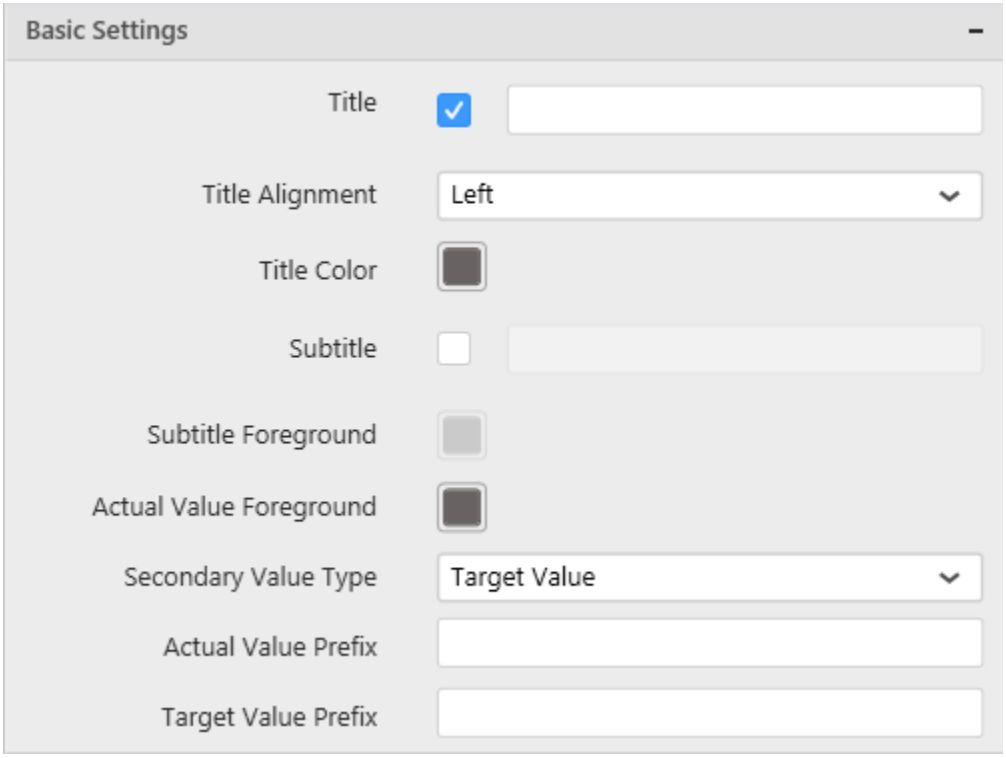
SubHeading

This allows you to set a sub title for the series of cards.

Description

This allows you to set description for this card widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings



Title

You can set a custom name as card title.

Title Alignment

The title can be aligned left, center, or right.

Title Color

The color of title text can be customized.

Subtitle

A sub title can be added to the card control providing a suitable text. Subtitle text will be displayed at the bottom of the title text.

Subtitle Foreground

The text color of subtitle text can be customized.

Actual Value Foreground

You can customize the color of card value. This option will be disabled when Show Indicator Only option was enabled.

Secondary Value Type

This allows you to set the Secondary Value Type based on the Variation and Target Value.

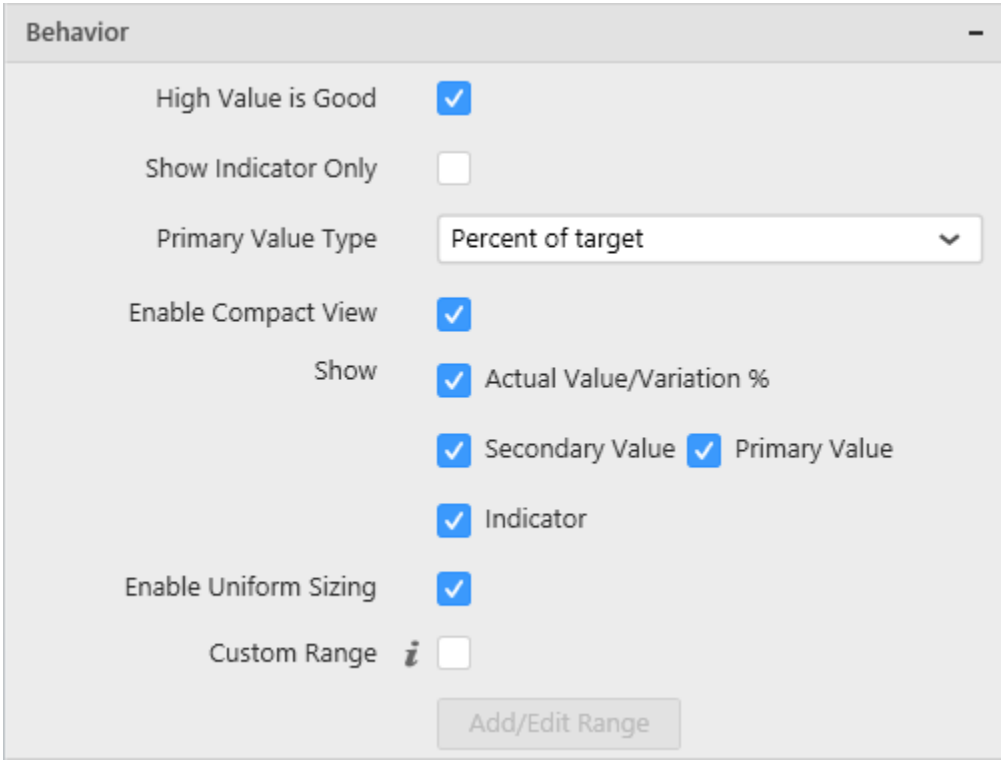
Actual Value Prefix

This allows you to set the actual value prefix.

Target Value Prefix

This allows you to set the target value prefix.

Behavior Settings



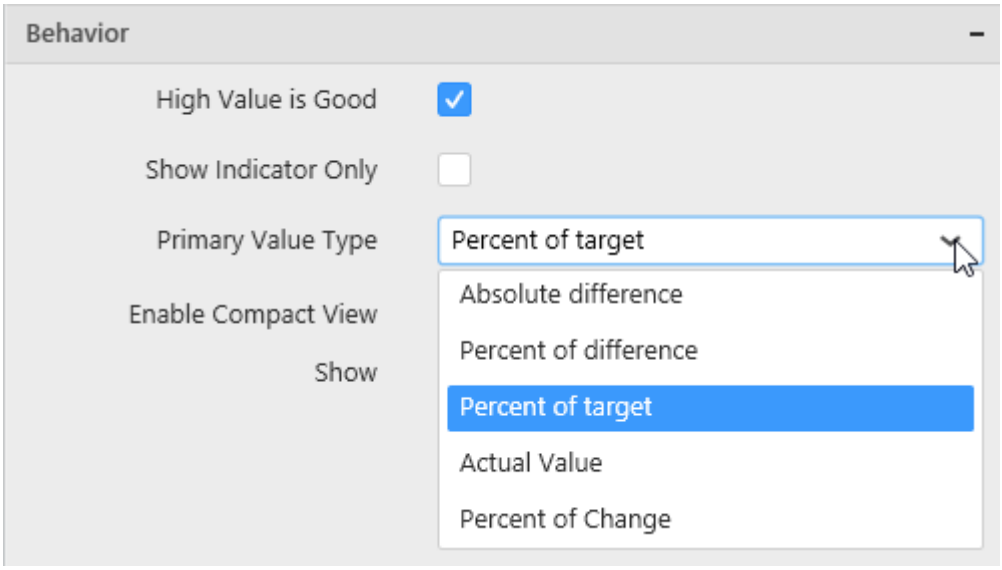
High Value is Good

The card visualization can be customized through specifying whether higher value should be treated as good or bad.

Show Indicator Only

Enabling this allows you to show indicator representation alone in card. This option will be available only if columns are added to both Actual Value and Target Value sections.

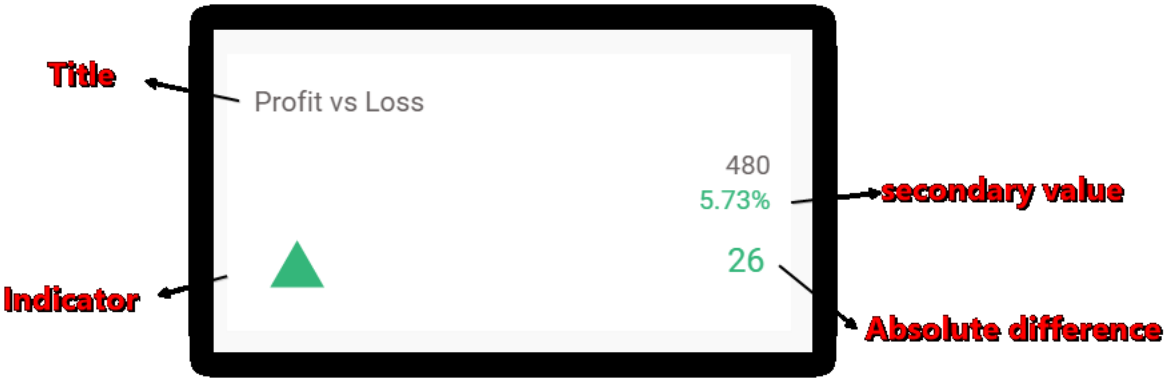
Primary Value Type



This allows you to customize the data showcased in card control by switching the available Primary Value Types.

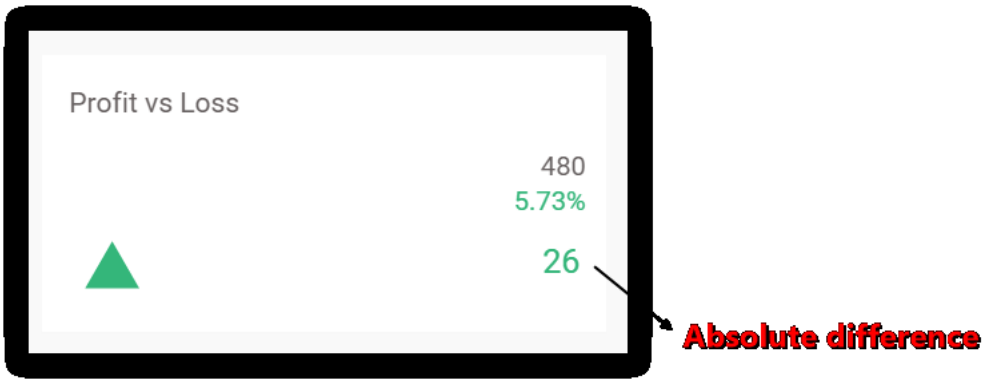
- Absolute difference
- Percent of difference
- Percent of target
- Actual Value
- Percent of Change

Default layout



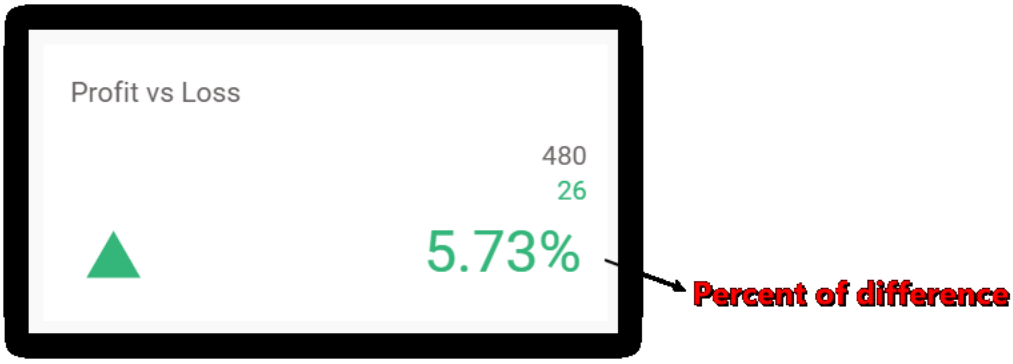
Absolute difference

Absolute difference = Actual Value – Target Value



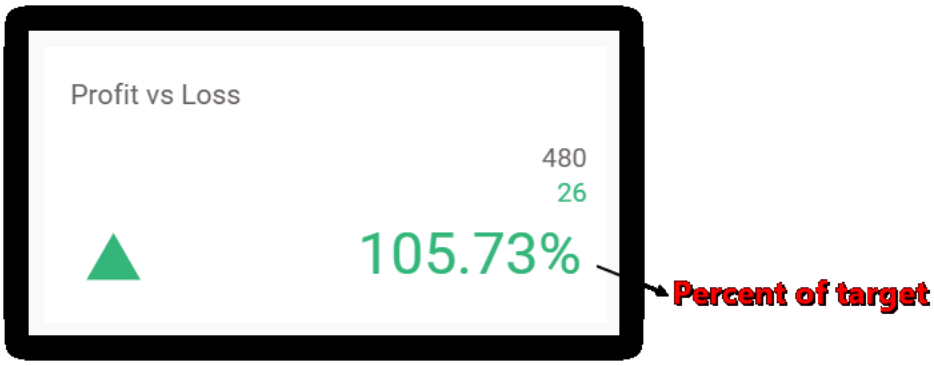
Percent of difference

Percent of difference = [([Actual Value / Target Value] 100)-100]*

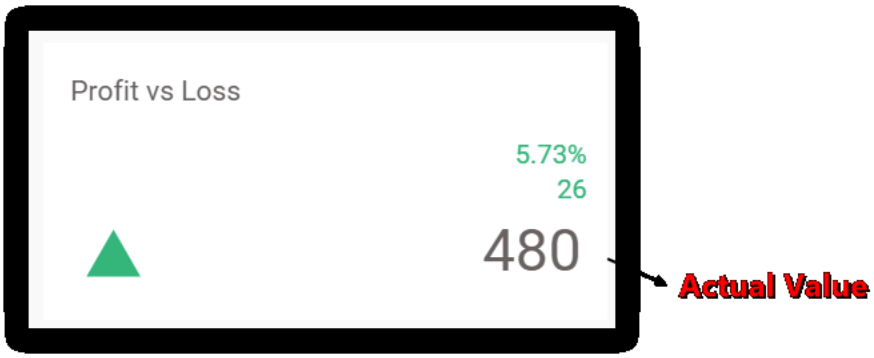


Percent of target

Percent of target= [([Actual Value / Target Value] 100)]*



Actual value

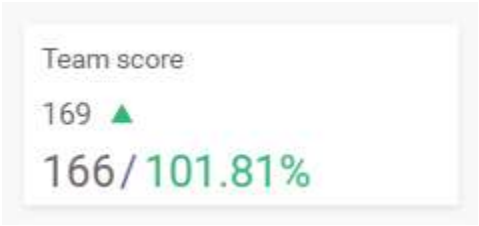


Percent of Change

Percent of Change = [((Actual Value - Target Value) / Actual Value) 100]*

Compact View

With minimal space, this option allows the user to showcase the values in a legible manner.



Visibility of Elements

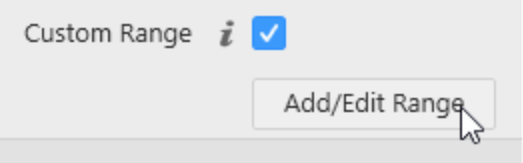
This option allows to select the items to be displayed in card as desired.

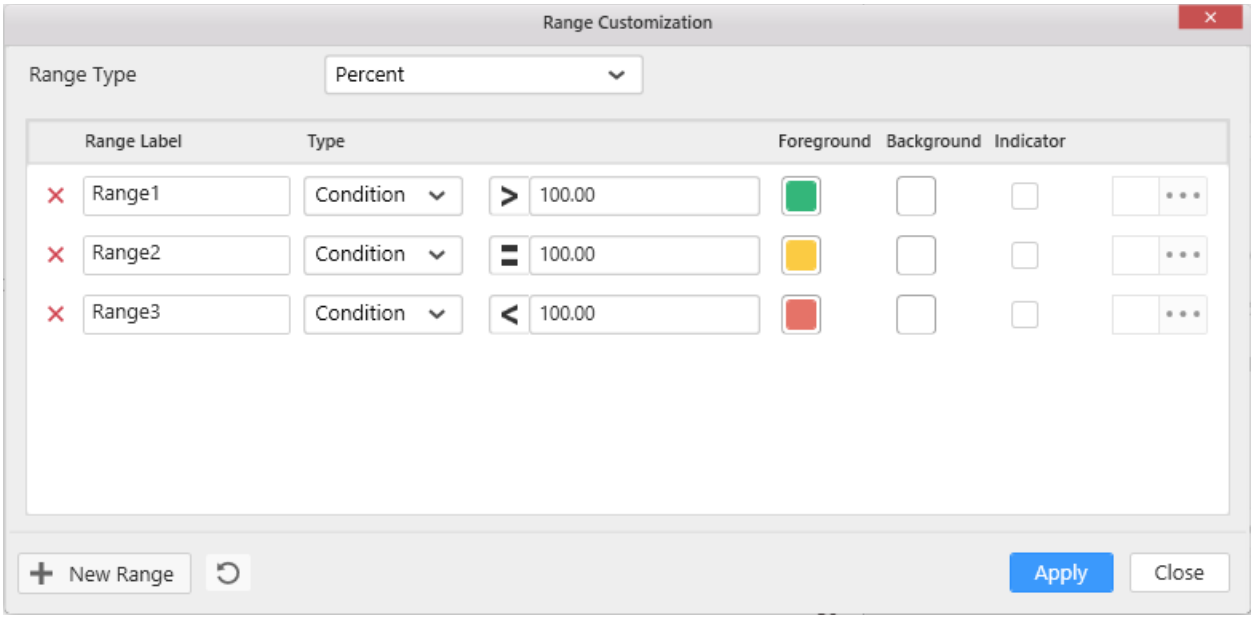
Uniform Sizing

Displays the header and values in even font size for the cards designed with same height and width.

Custom Range

This option allows you to customize the colors and ranges of Card with the help of a window



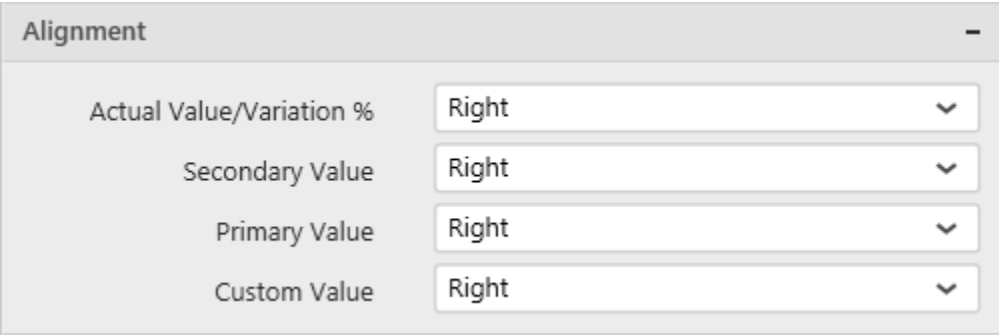


This window allows you to perform the following actions

- To set a name for each range
- To add a new range
- To customize the range limits
- To delete an existing range
- To set an image as indicator, foreground and background colors for each range
- To apply the range based on absolute values or percentage of its original value

Alignment Settings

This settings allows to you align the values at the Left, Right and Center positions.



Filter Settings

Act as Master

This allows you to define this grid widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

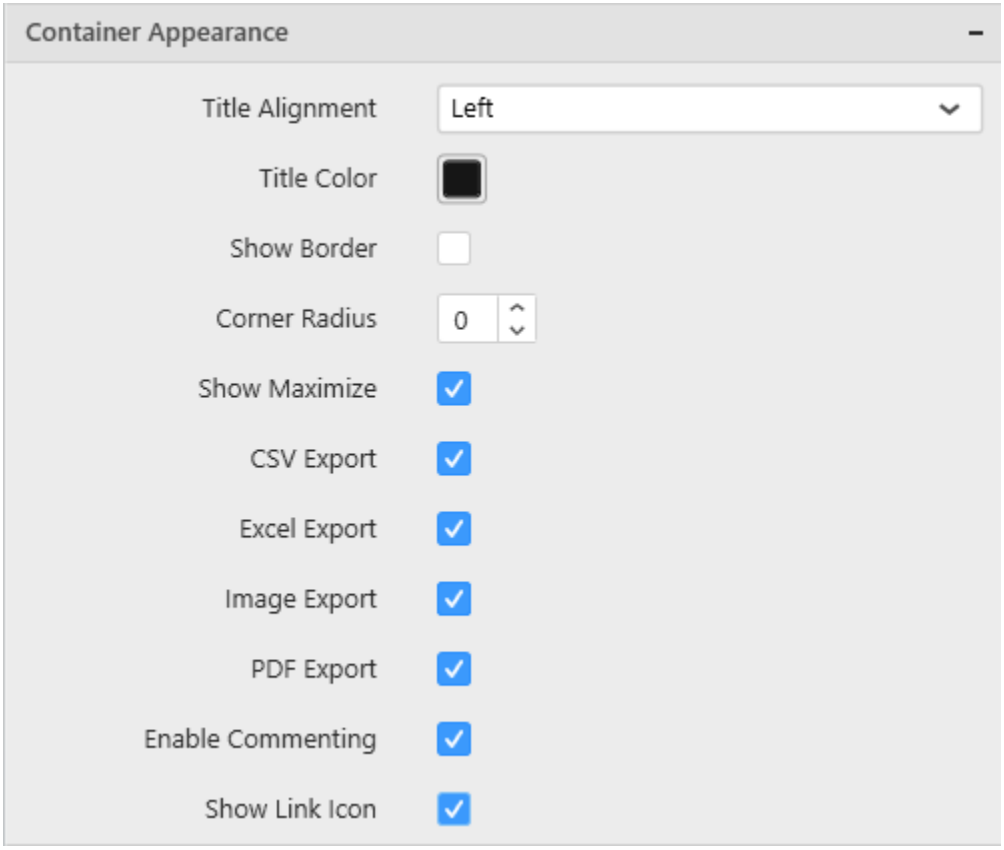
Ignore Filter Actions

This allows you to define this grid widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings

You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this card widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the grid widget.

CSV Export

This allows you to enable/disable the CSV export option for this card widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this card widget. Enabling this allows you to export the summarized data of the widget view to XLSX format.

Image Export

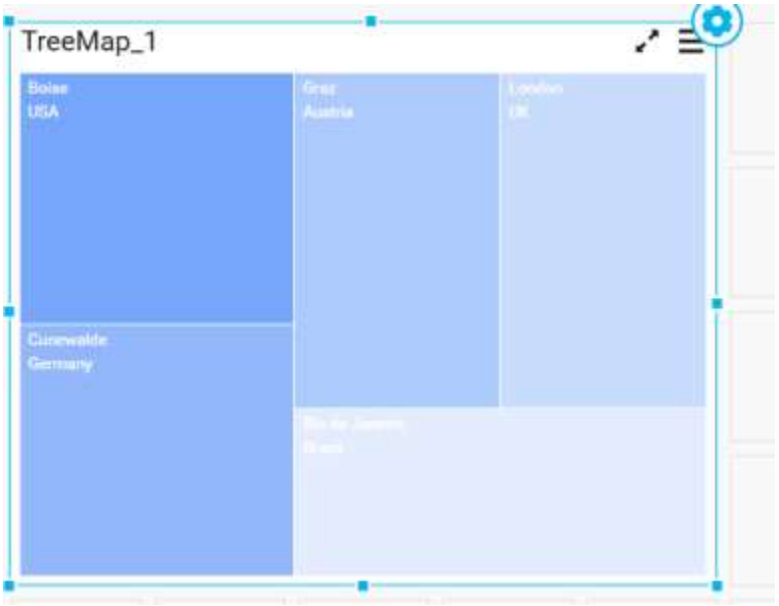
This allows you to enable/disable the image export option for this card widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Tree Map

Tree Map allows you to visualize large data through its proportional shelves and color scales.

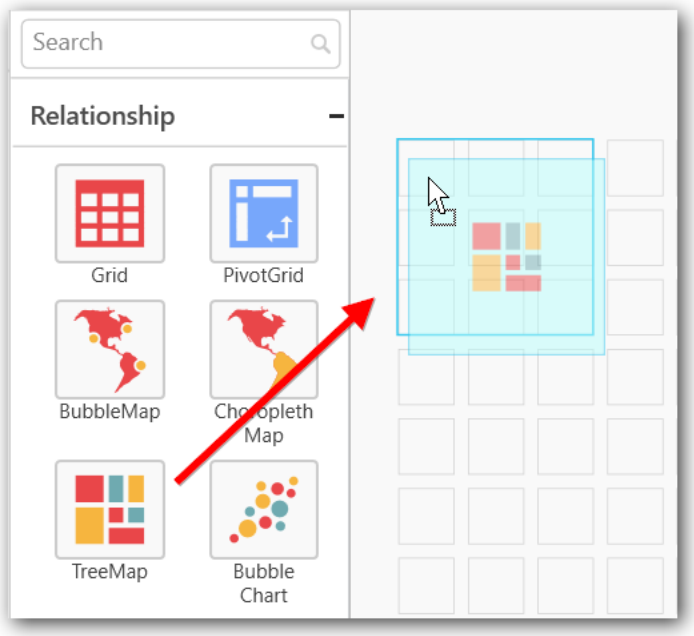


How to configure flat table data to Tree map widget?

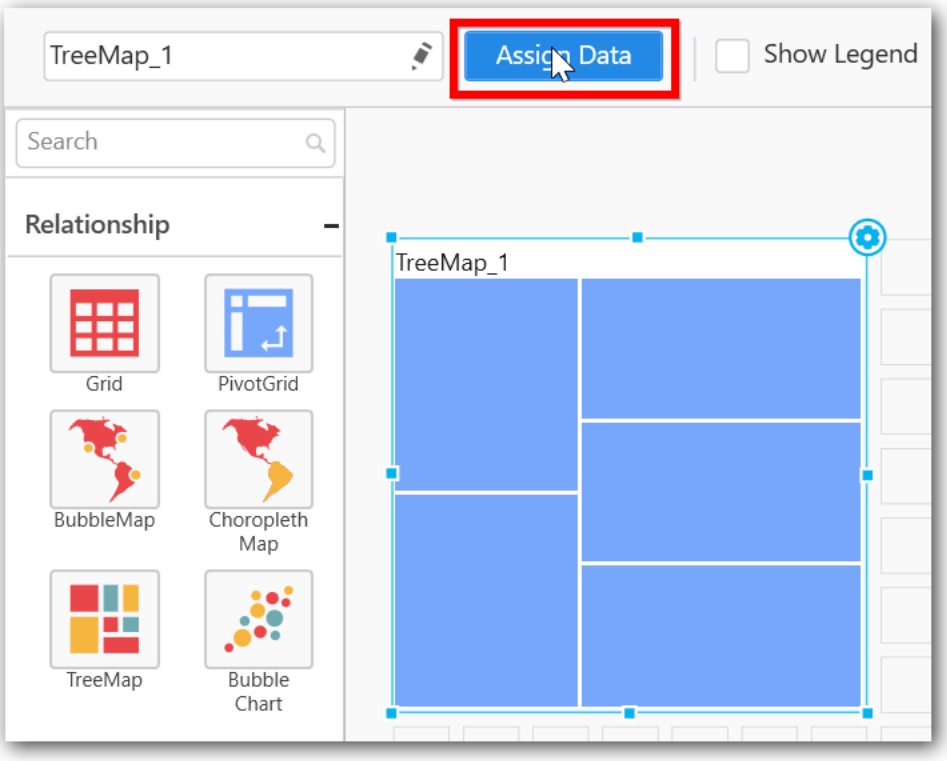
To showcase a tree map, a minimum requirement of 1 value and 1 group by field is needed.

The following procedure illustrates data configuration of Tree Map.

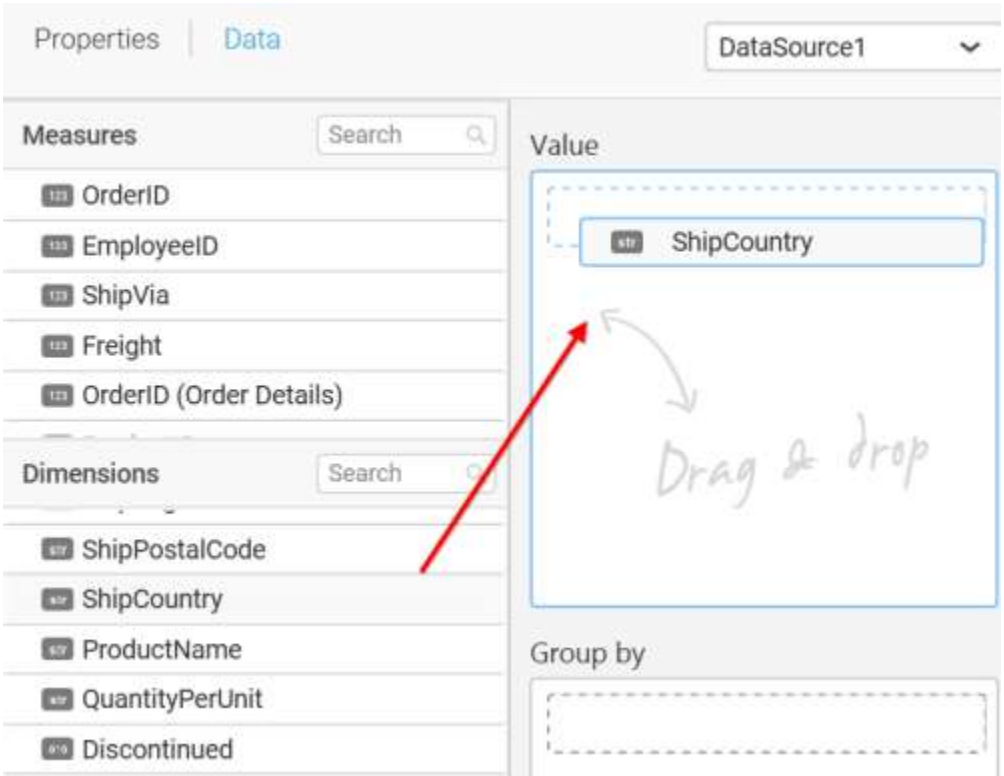
Drag and drop TreeMap control icon from the Tool box into design panel. You can find control in Toolbox by search.



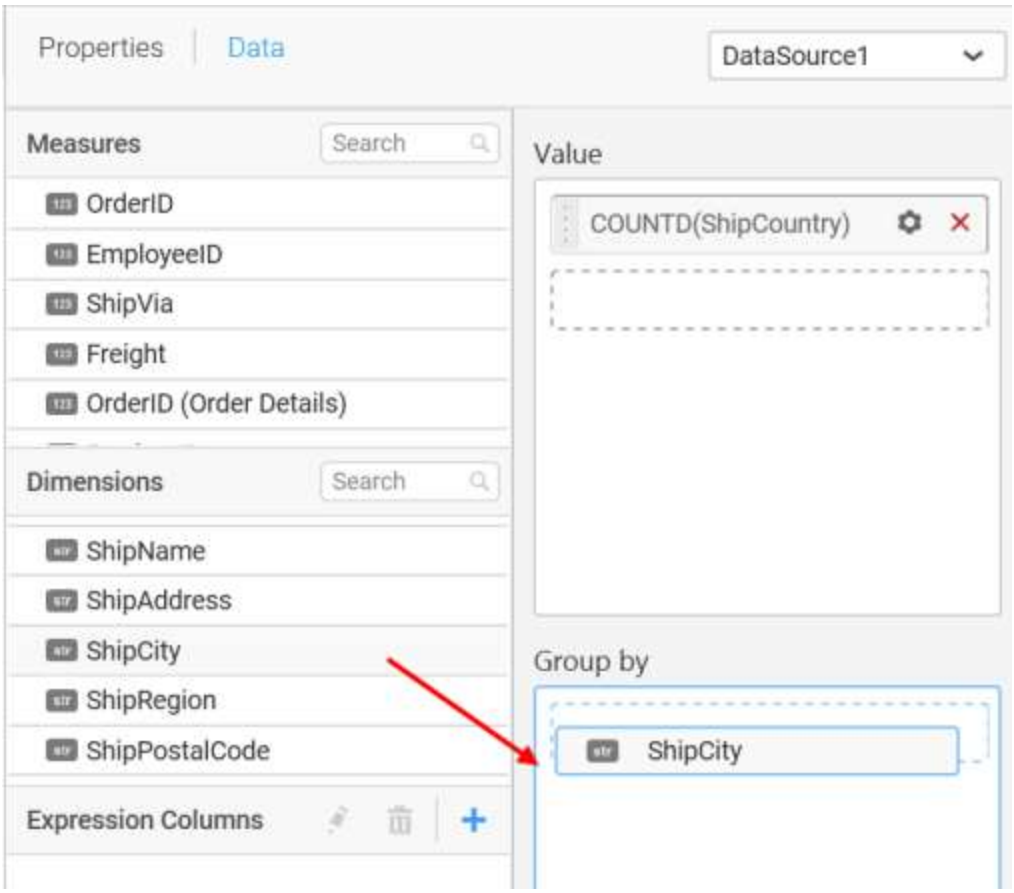
After control added in design panel, click **Assign Data** button at Design Tools Pane to open the Data configuration pane.



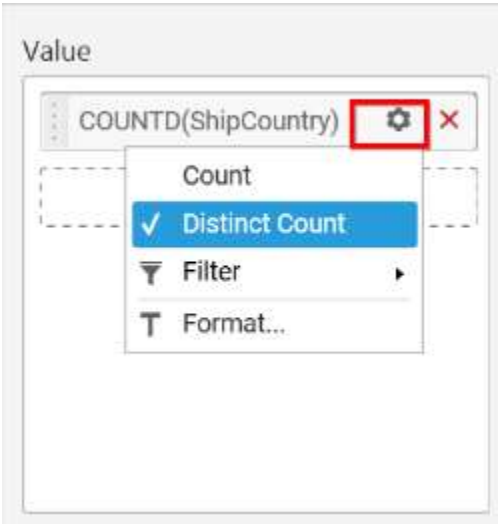
Bind column through drag and drop element from sections to Value.



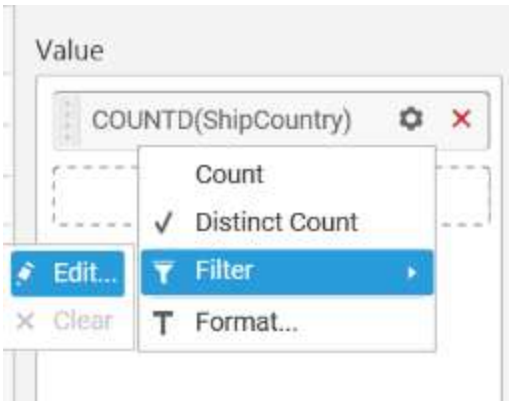
Drag and Drop the elements from sections to Group by.



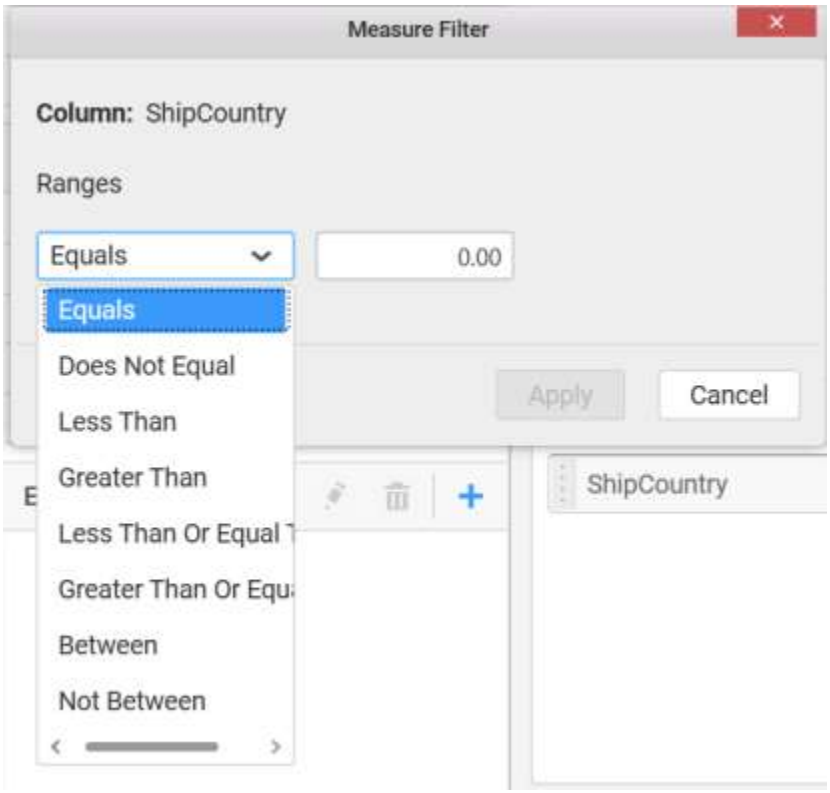
You can use aggregate function to change the Value of the column.



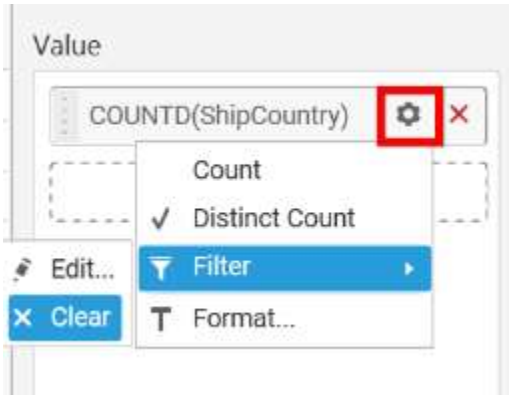
You can use Filter option to filter the data by specifying the filter condition.



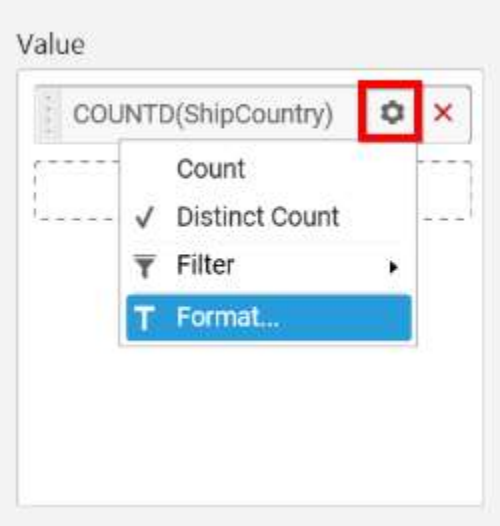
Measure Filter window will be shown to edit the filter condition.



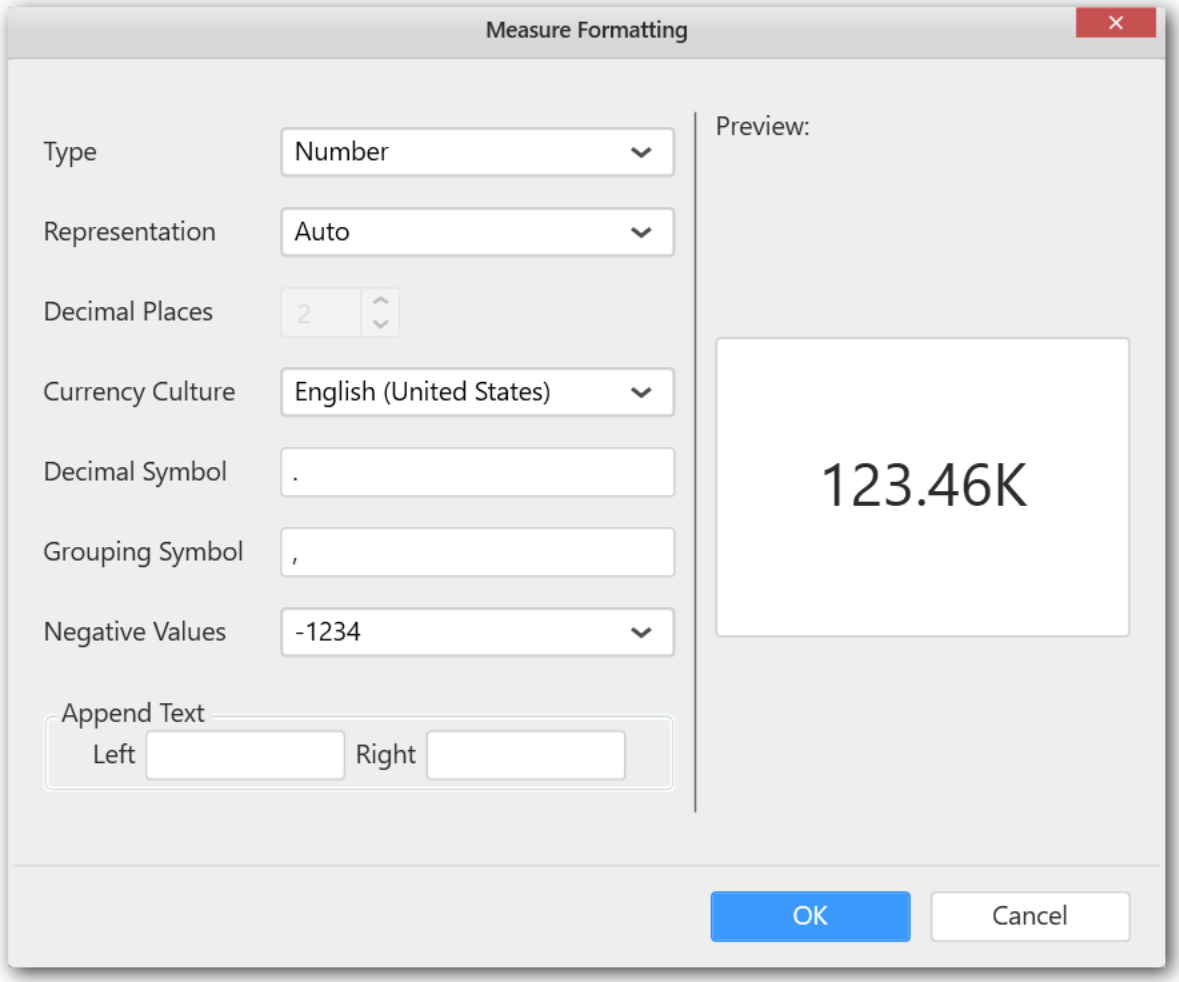
You can clear filters by selecting the Clear option for Value section.



You can format the values by selecting the Format option.

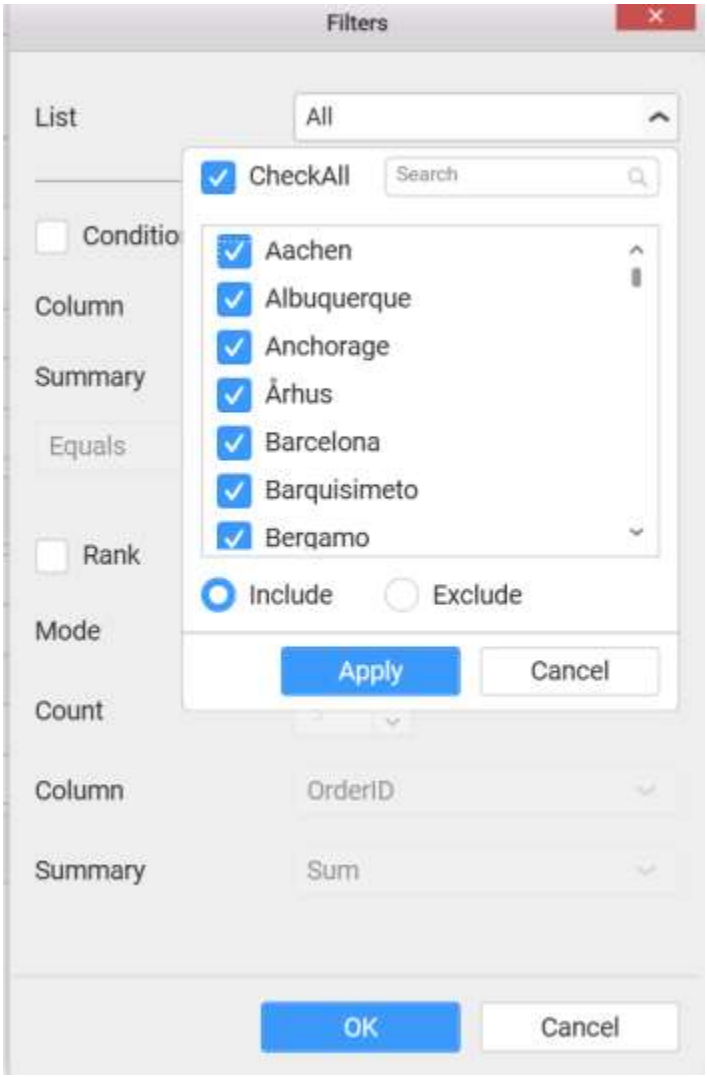


Measure Formatting window will be shown.

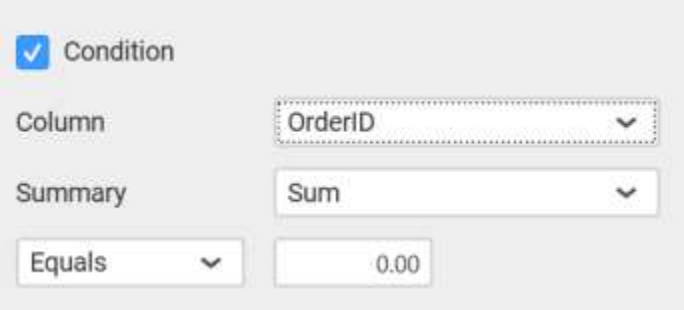


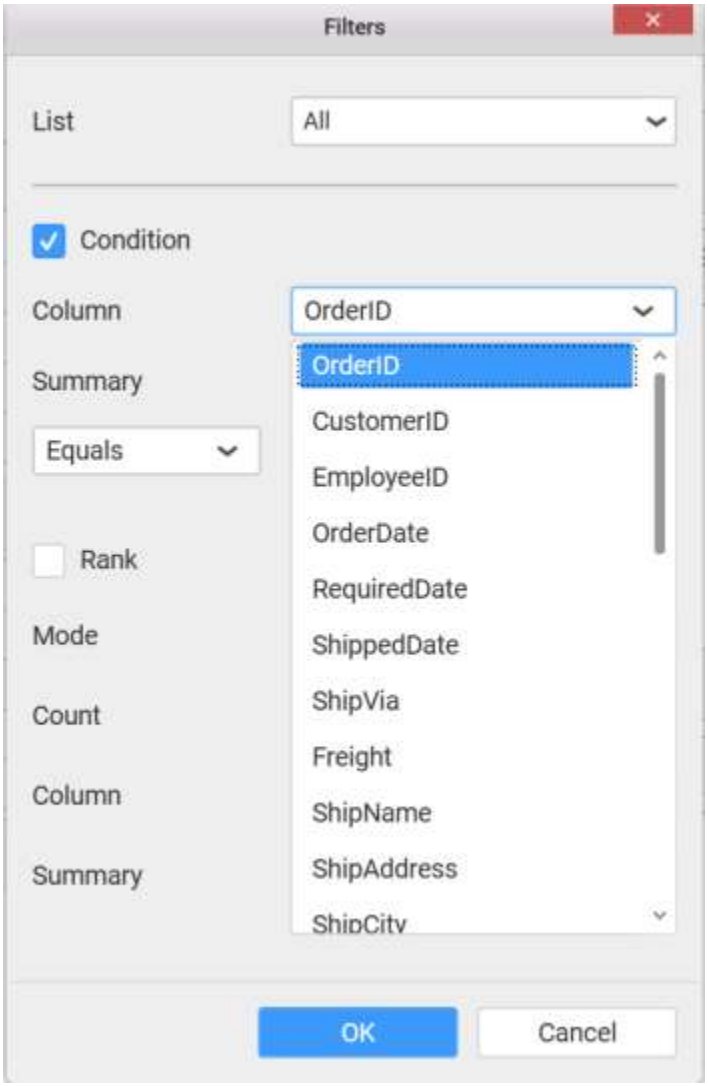
You can use the filters by selecting the Filter(s)... option to rank to the elements.

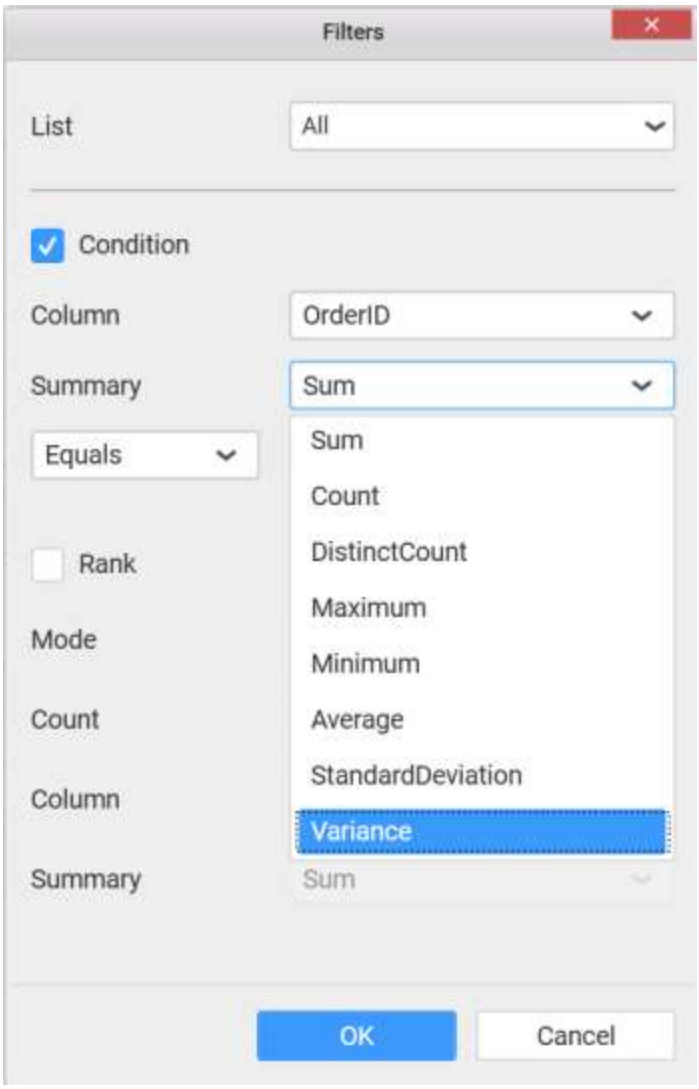
You can select the specific city to filter the element and CheckAll is used either to check all the data or to select the specific data. Include and Exclude is used to include and exclude the elements by selecting the radio button and click the Apply button.



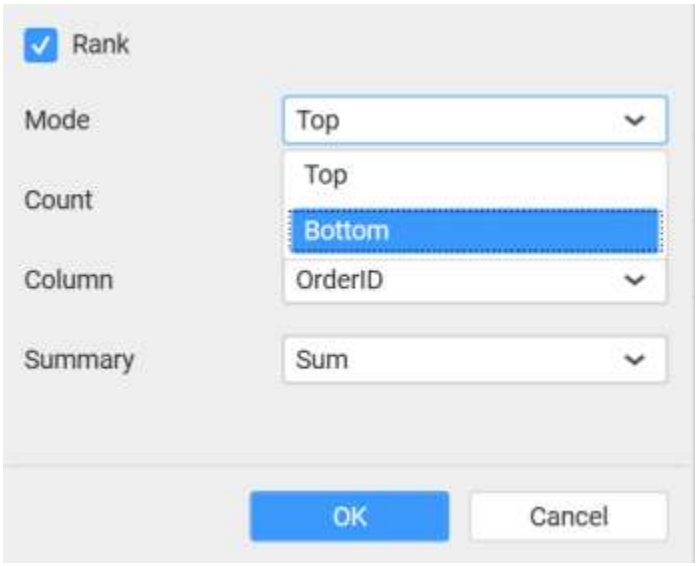
You can select the Condition option to change the Column elements and Summary type by selecting the required column name and summary type.



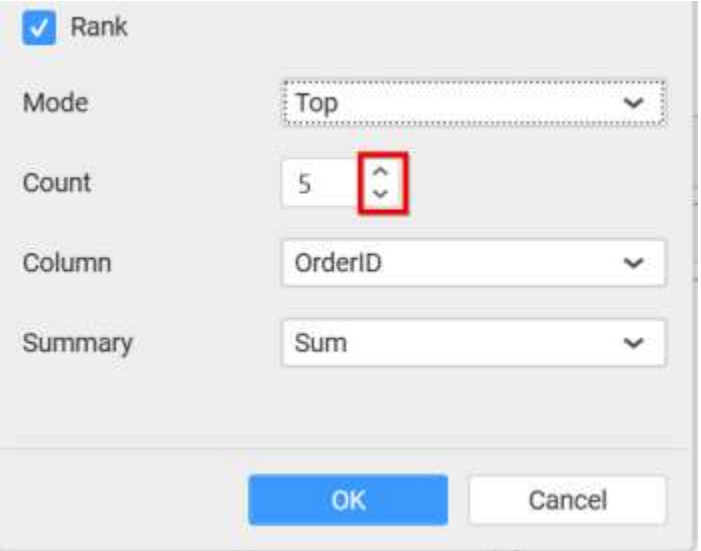




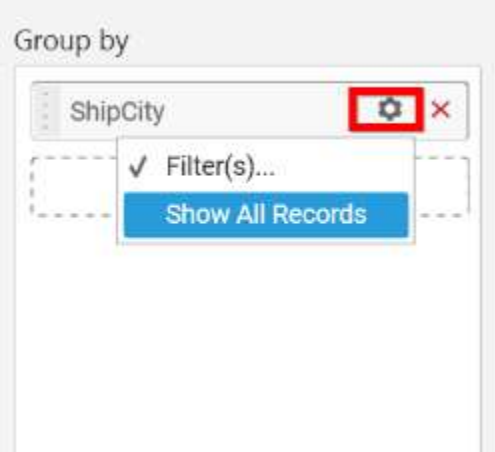
You can select the Rank option to enable filters and select the Mode either top or bottom.



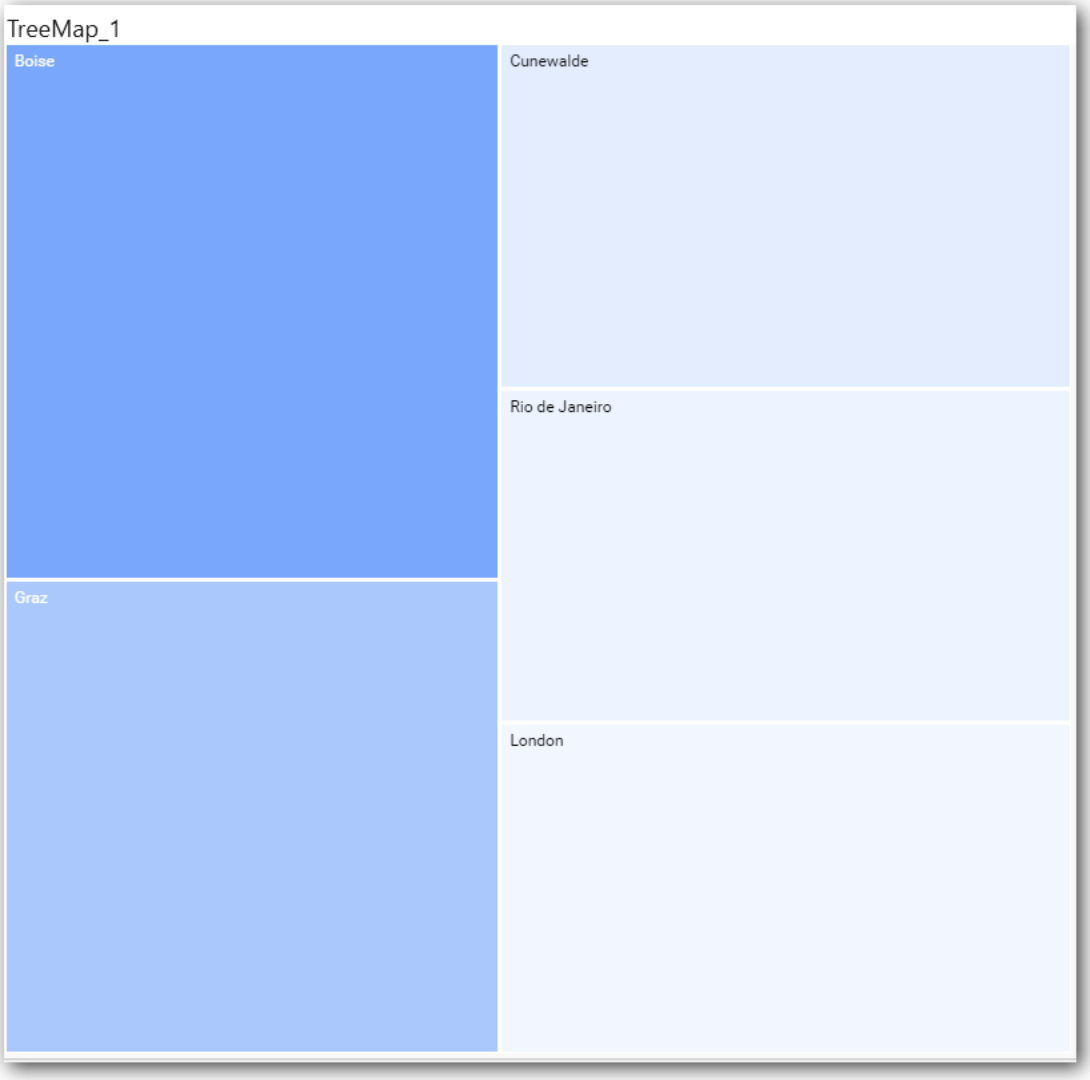
You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.



You can clear filters by selecting the Show All Records.



Here is an illustration,

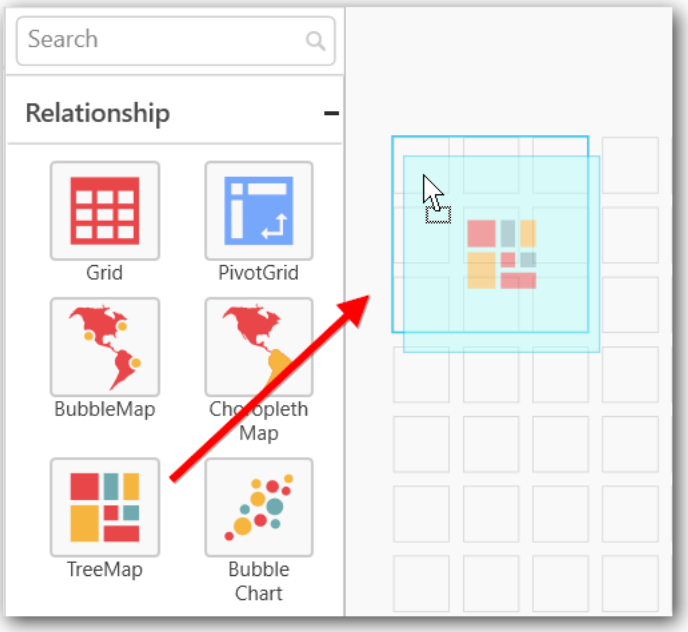


How to configure the SSAS data to Treemap?

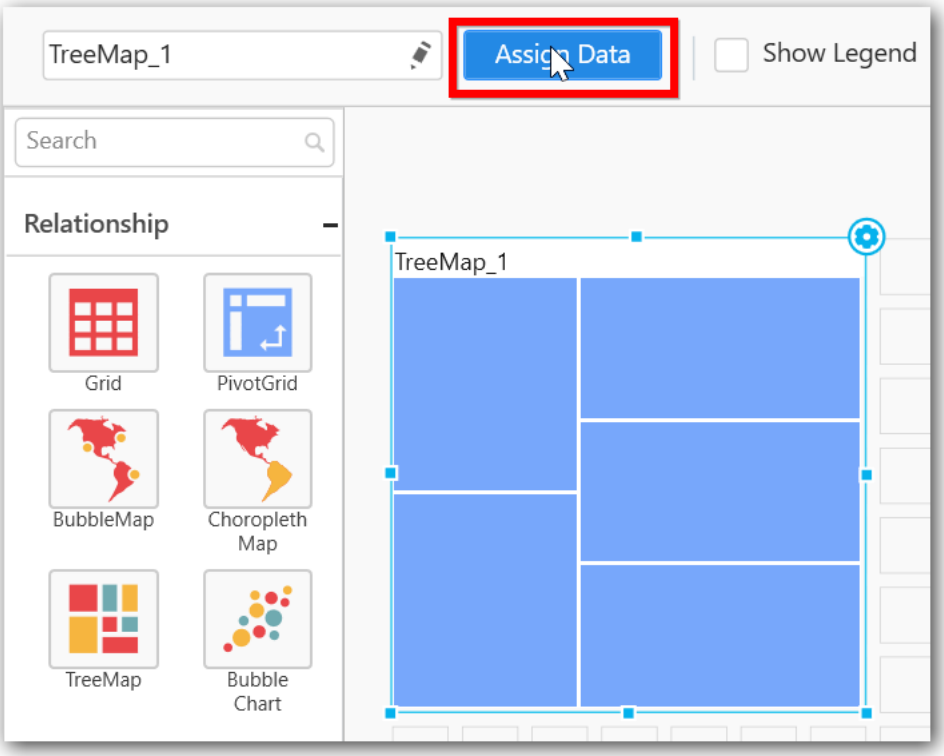
To showcase a tree map, a minimum requirement of 1 value and 1 group by field is needed.

Following steps illustrates configuration of SSAS data to Tree map

Drag and drop the TreeMap widget into canvas and resize into your required size.

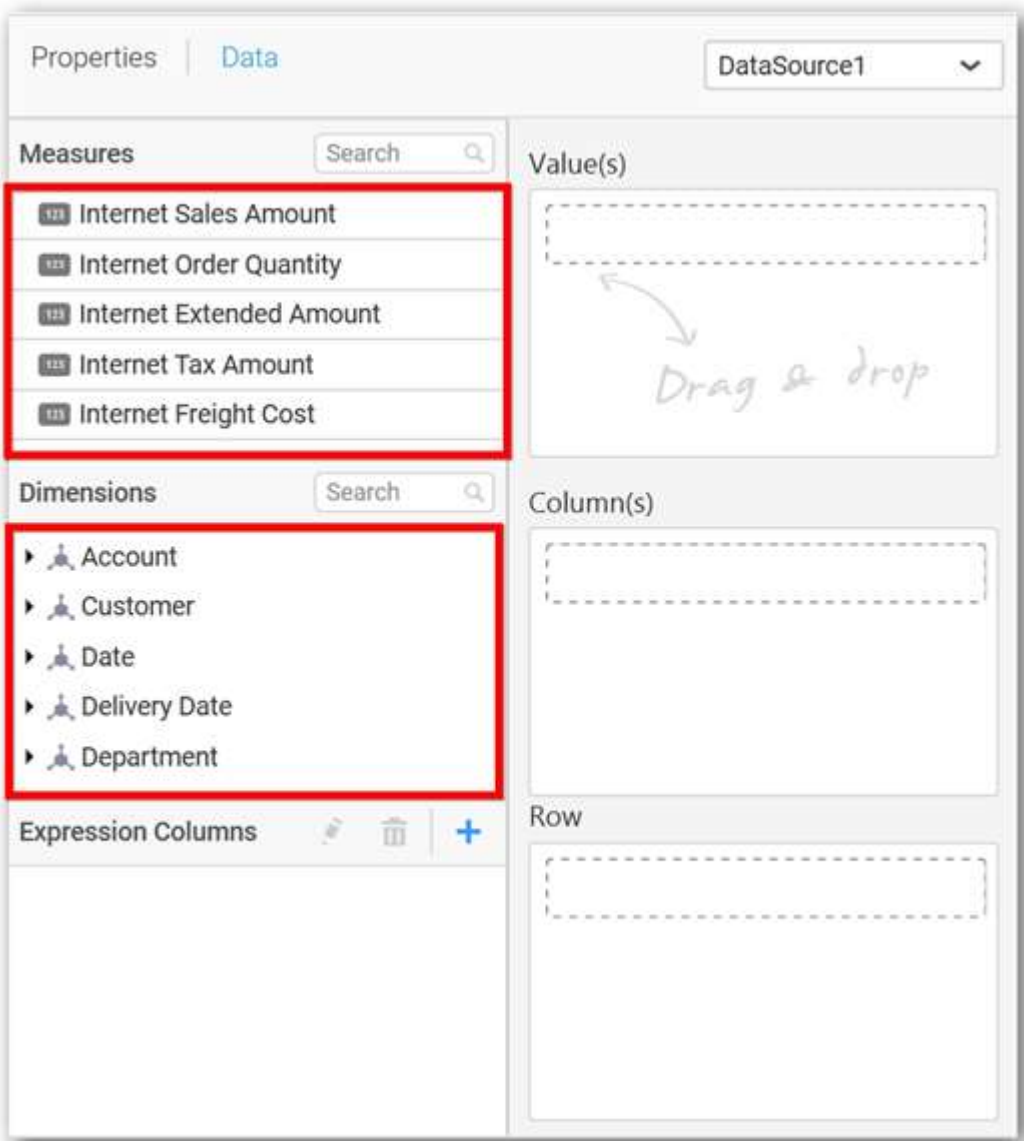


Select the dropped widget using mouse.

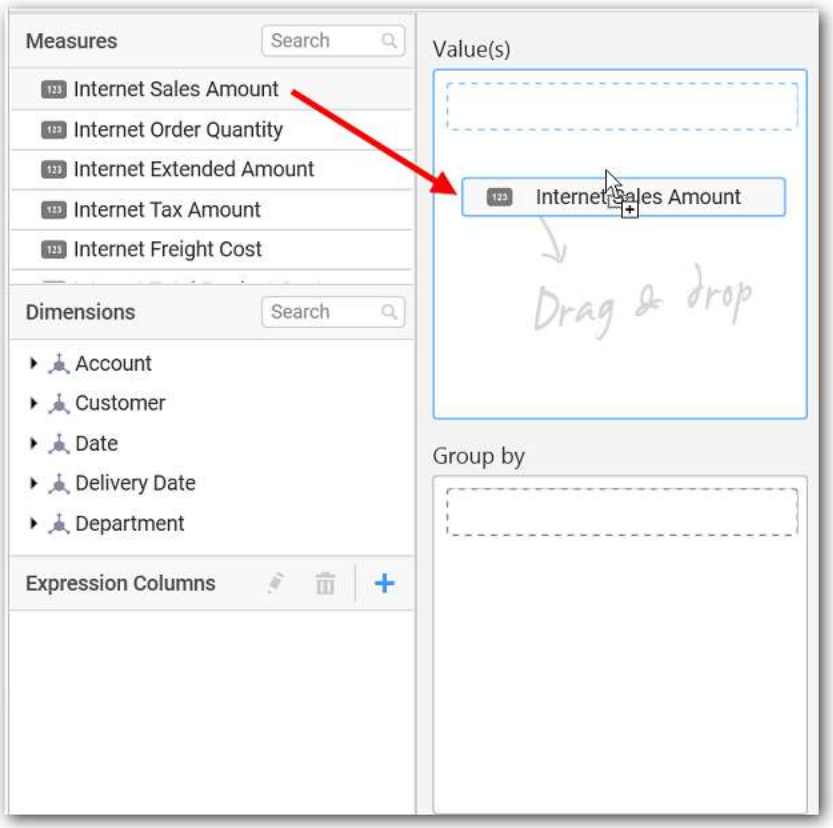


Click the Assign Data button in the toolbar.

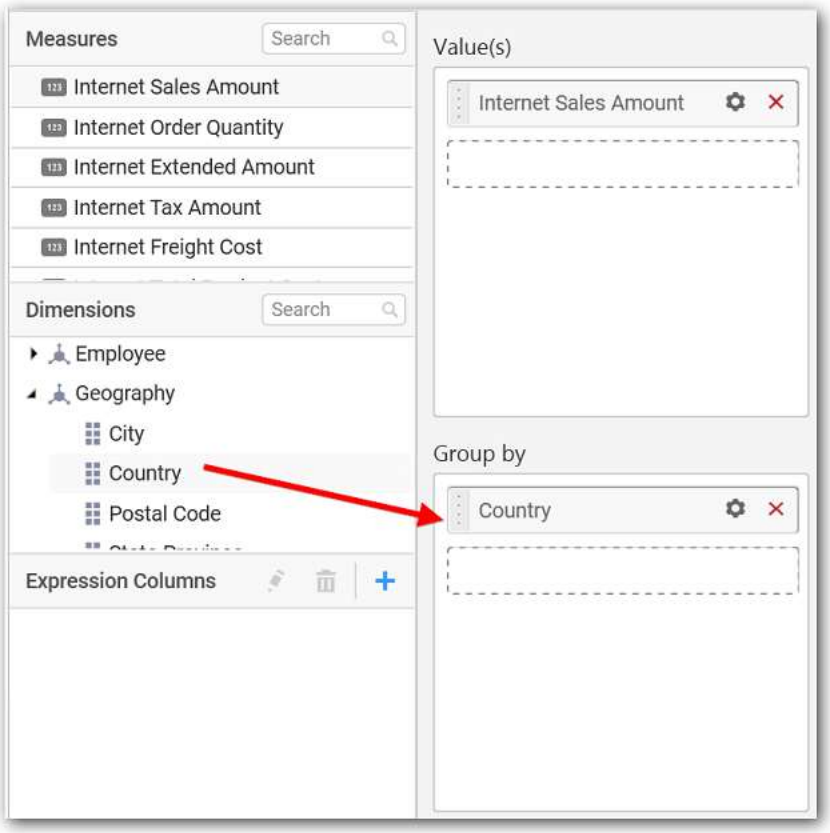
A Data pane will be opened with available Measures and Dimensions.



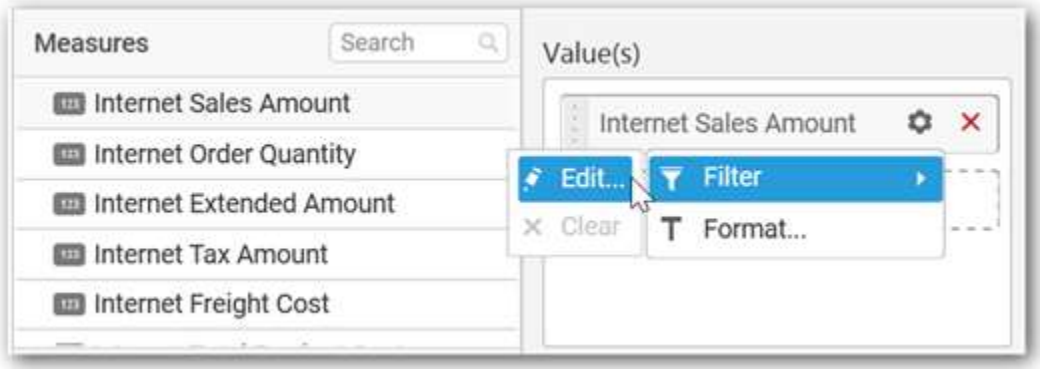
Drag and drop a column under Measures category into Value(s) section.



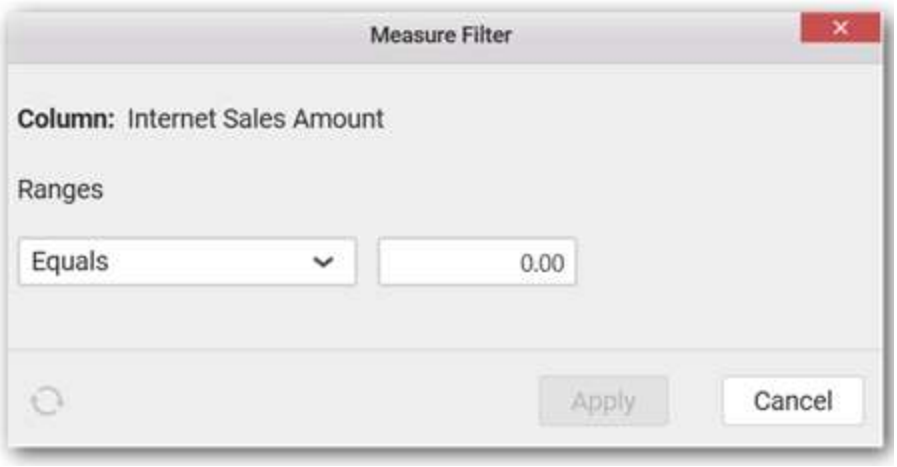
Drag and Drop the elements from sections to Group by.



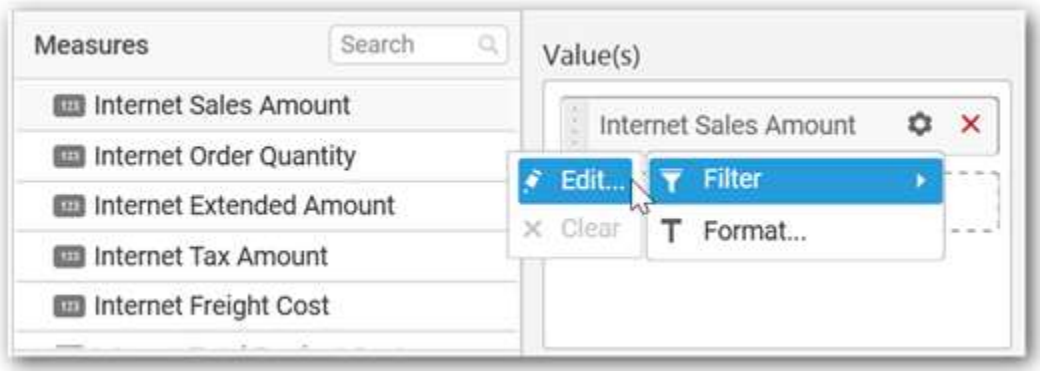
Define the filter criteria to match through choosing Edit option in Filter menu item.



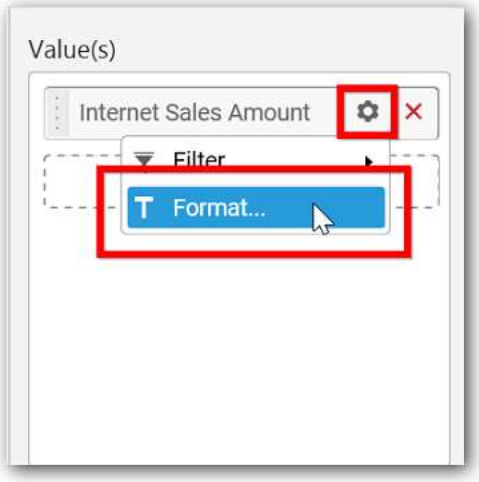
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



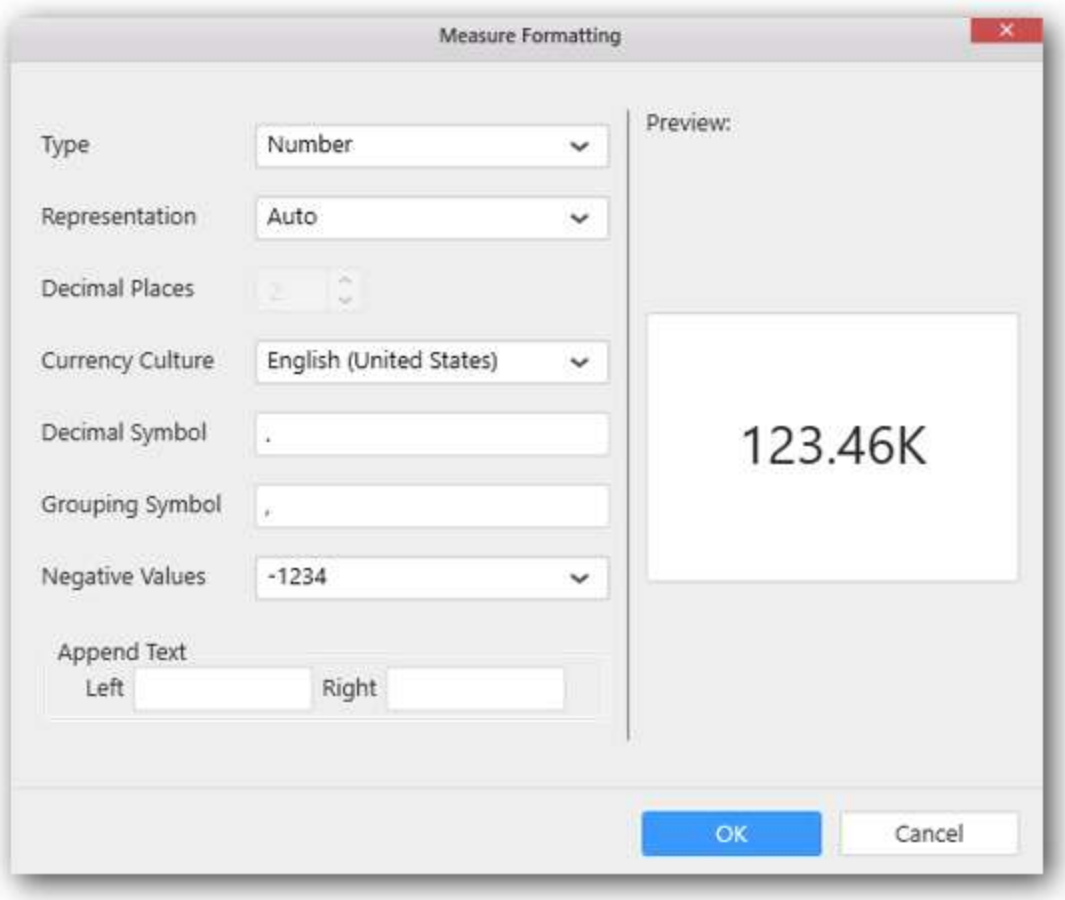
Select Clear option to clear the defined filter.



Select Format option to define the display format to the values in the column through Measure Formatting window.

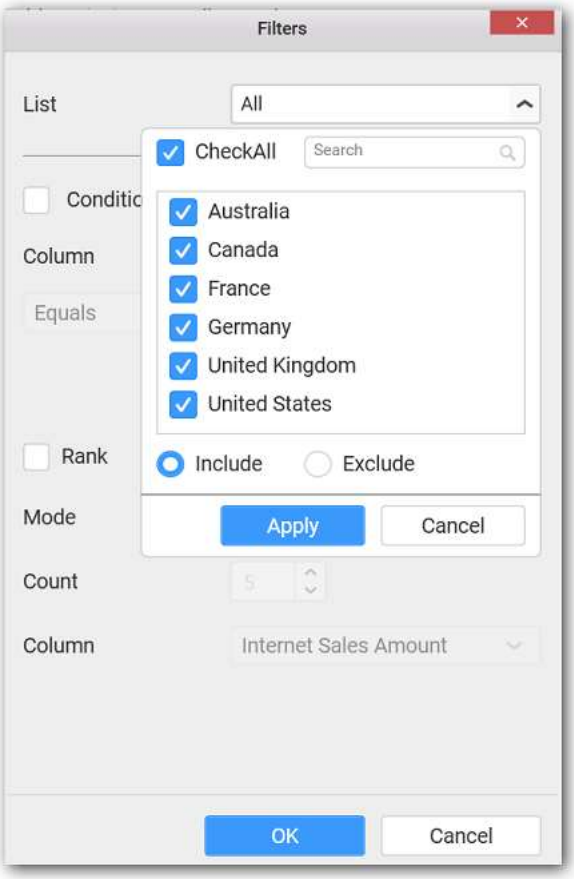


Choose the options you need and click OK.

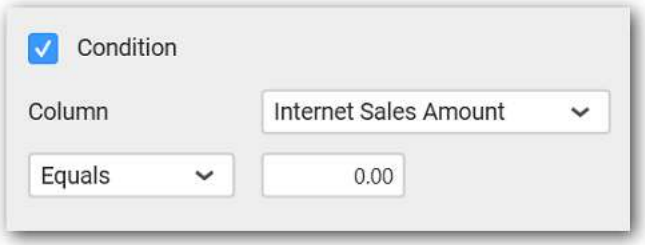


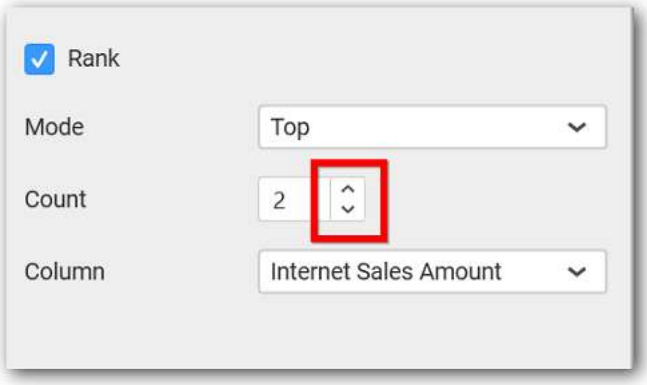
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.

Select Filter(s)... to launch the Filters window.

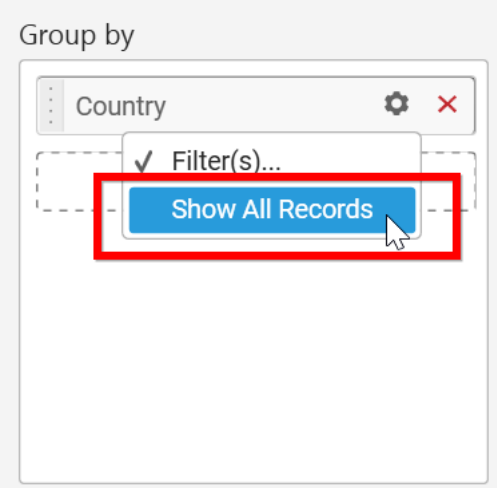


Define the filter Condition and Rank and Click OK..

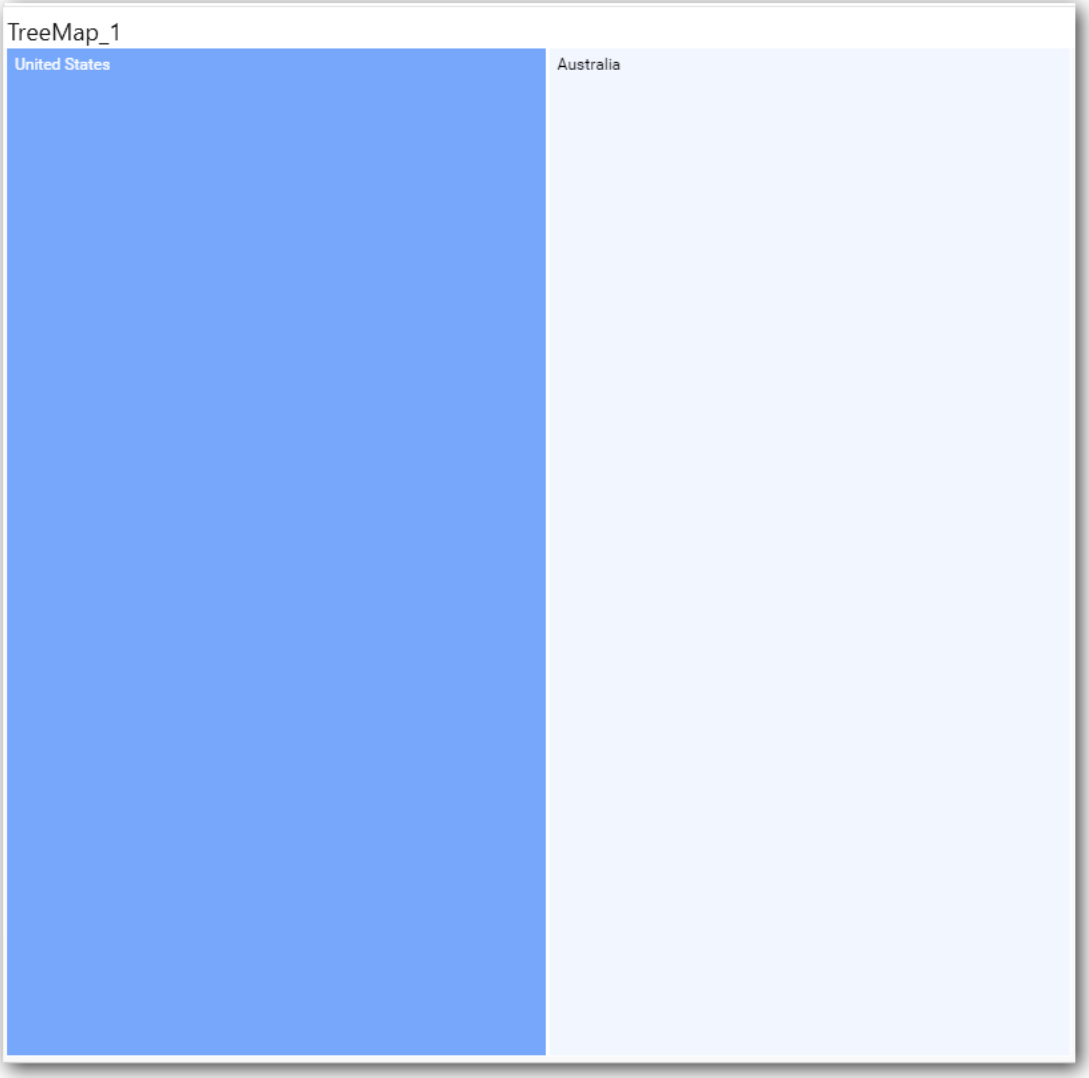




To show all records again click on Show All Records.



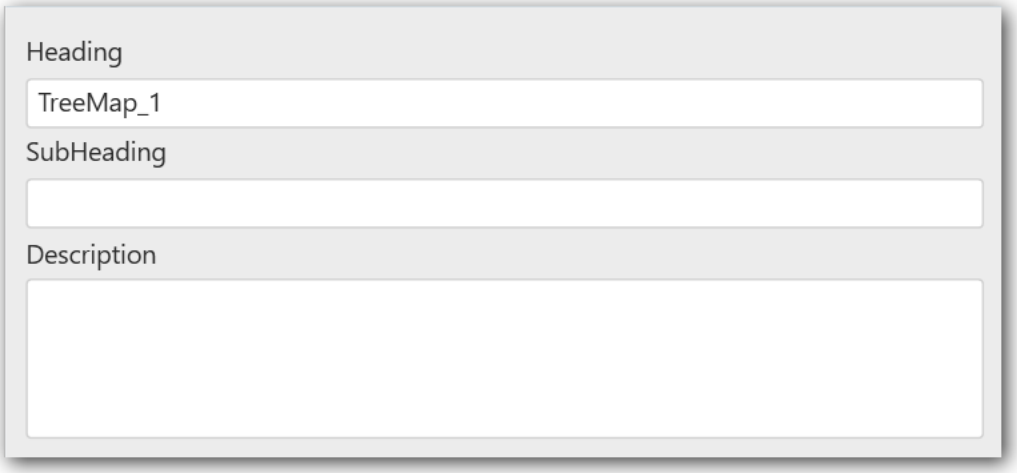
Here is an illustration,



How to format Tree map widget?

You can format the Tree map for better illustration of the view that you require, through the settings available in Properties pane.

General Settings



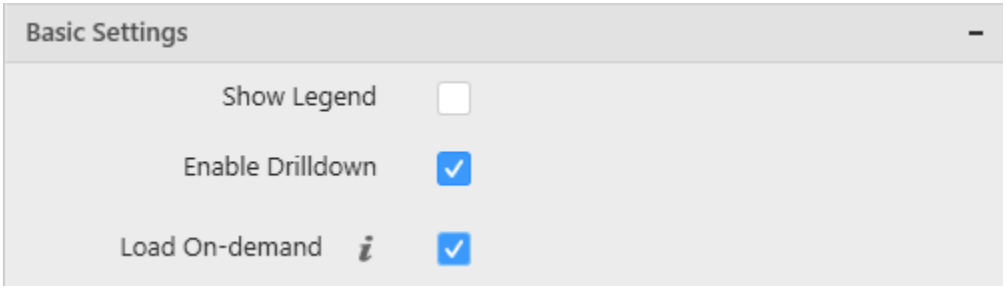
Header

This allows you to set title for this tree map widget.

Description

This allows you to set description for this Tree map widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings



Show Legend

Allows to show/hide the legend for Treemap.

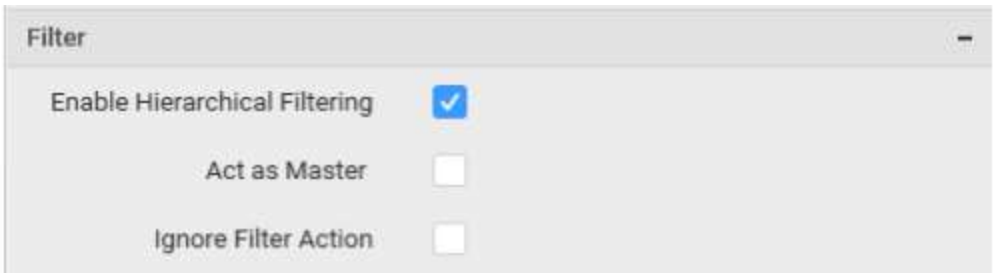
Enable Drilldown

In case of hierarchical view, multiple levels will get rendered in the same view. This can be switched to drill down view through enabling this setting.

Load on demand

Allows to load the subsequent level data dynamically on drilldown. This improves the performance of Treemap widget.

Filter Settings



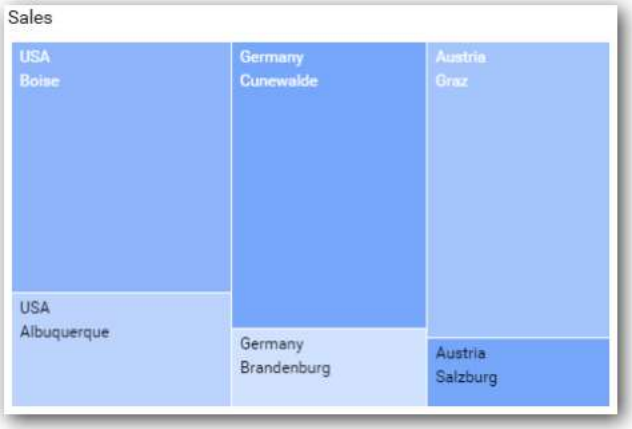
Enable Hierarchical Filtering

Through this option, you can enable/disable hierarchical Top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added. When Flat is set, the least number set as top will be applied for the whole data. When Hierarchical is set, the Top N will be applied for each individual column separately based on the number set for each column.

Below example shows data of 3 Country and its 2 Cities where the sales is high.

Flat Top N

Hierarchical Top N



Act as Master

This allows you to define this tree map widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

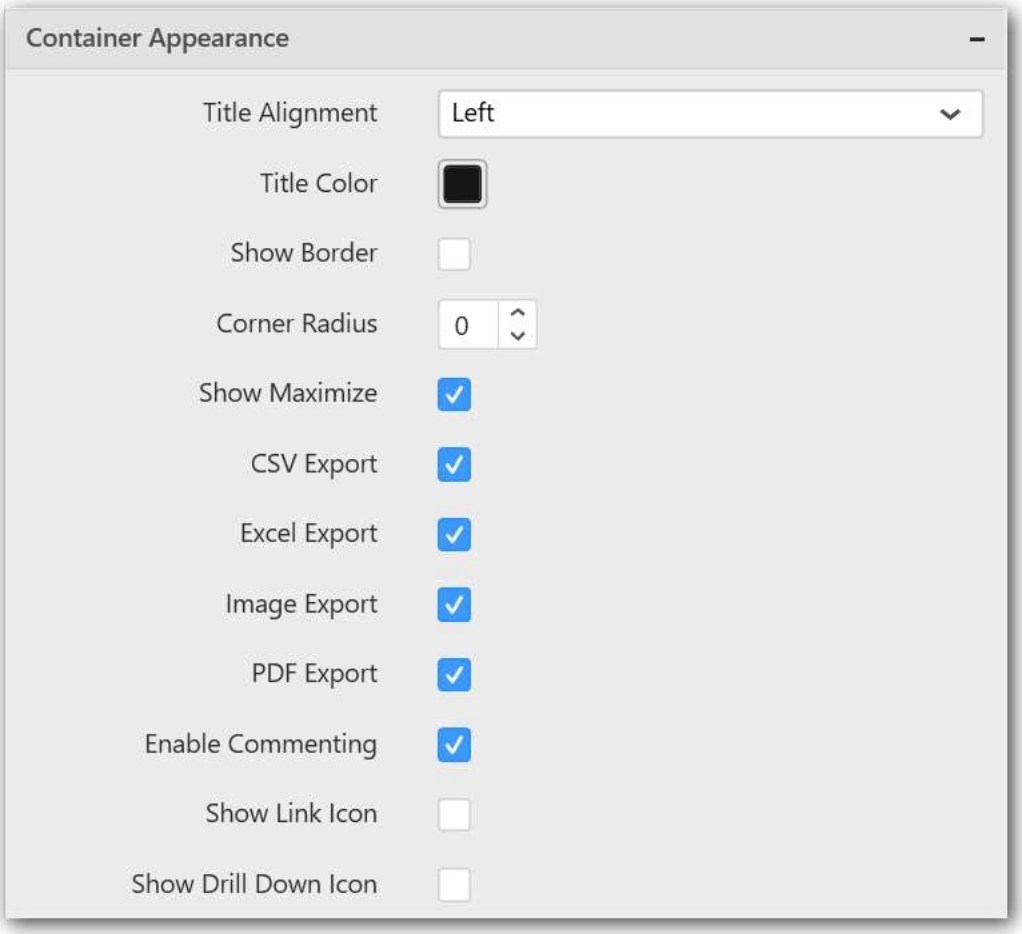
Ignore Filter Action

This allows you to define this tree map widget to ignore responding to the filter actions applied on other widgets in dashboard.

Link Settings

You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this tree map widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the tree map widget.

CSV Export

This allows you to enable/disable the CSV export option for this tree map widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this tree map widget. Enabling this allows you to export the summarized data of the widget view to XLSX format.

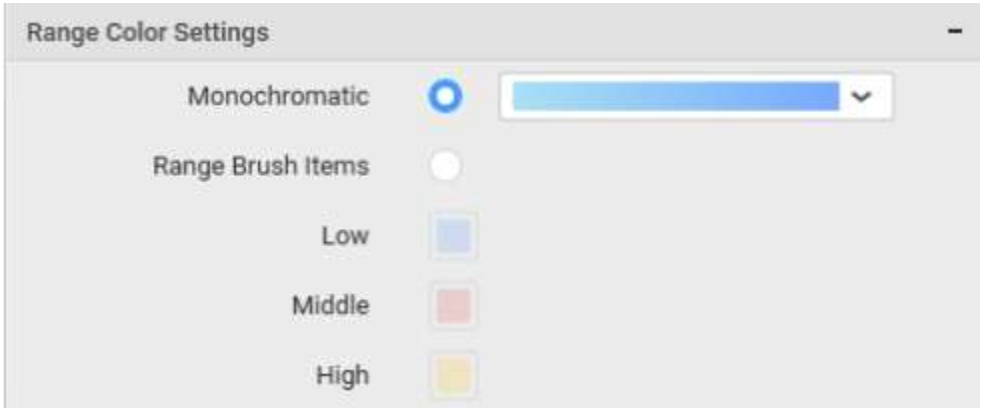
Image Export

This allows you to enable/disable the image export option for this tree map widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Range Color Settings



Monochromatic

This allows you to configure a single color palette whose saturation will be varied based on the value density.

Range Brush Items

This allows you to configure three different colors that allocates itself to the value range accordingly.

Low

Let you choose single color for low values when Range Brush Items is selected.

Middle

Let you choose single color for middle values when Range Brush Items is selected.

High

Let you choose single color for high values when Range Brush Items is selected.

Bubble Map

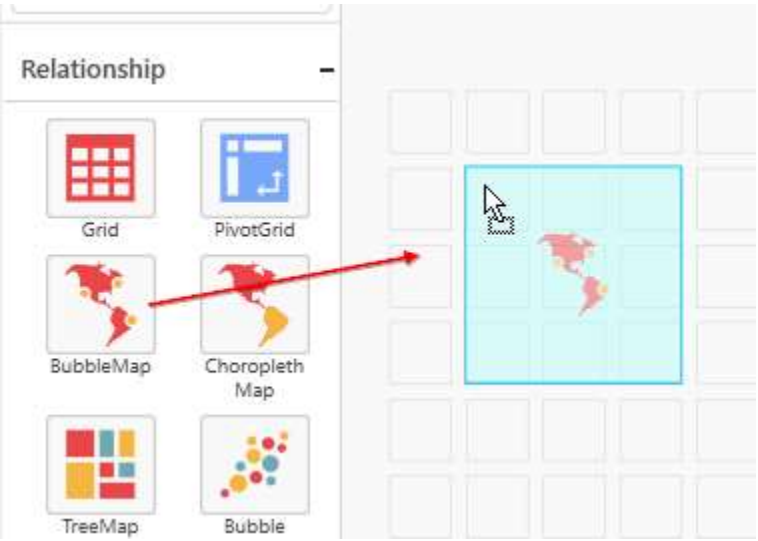
Bubble Map allows you to showcase quantitative values encoded through bubble size.

How to configure flat table data to Bubble Map?

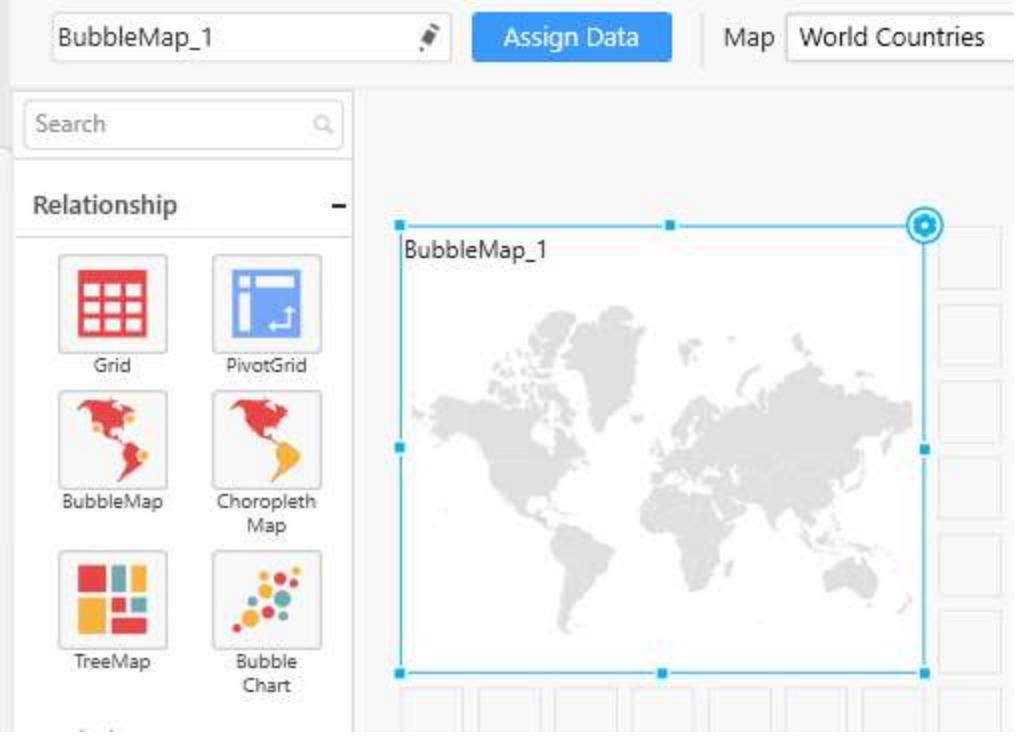
To plot a Bubble Map, a minimum requirement of 1 value and 1 shape is needed.

The following procedure illustrates data configuration of Bubble Map.

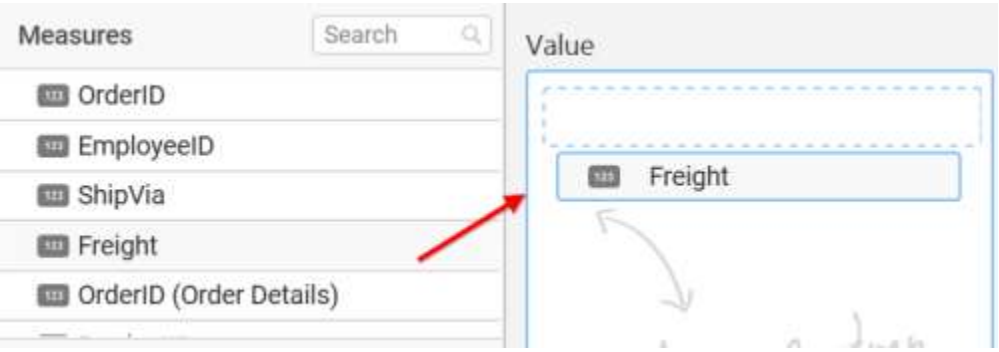
Drag and drop Bubble Map control icon from the Tool box into design panel. You can find control in Toolbox by search.



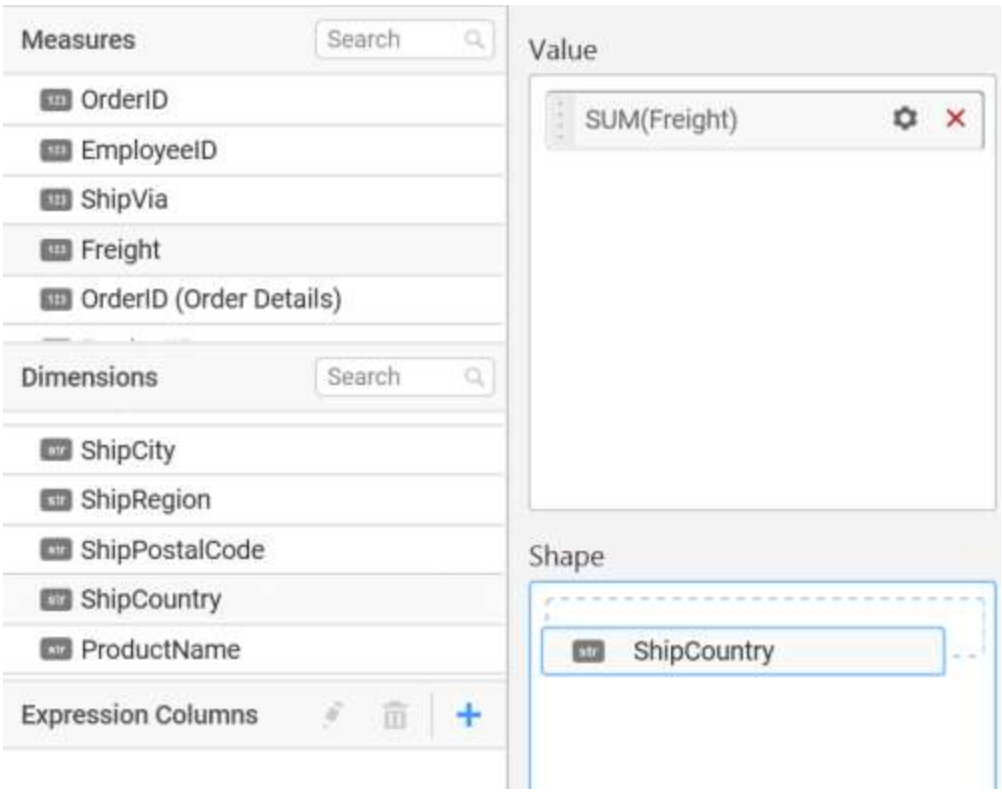
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



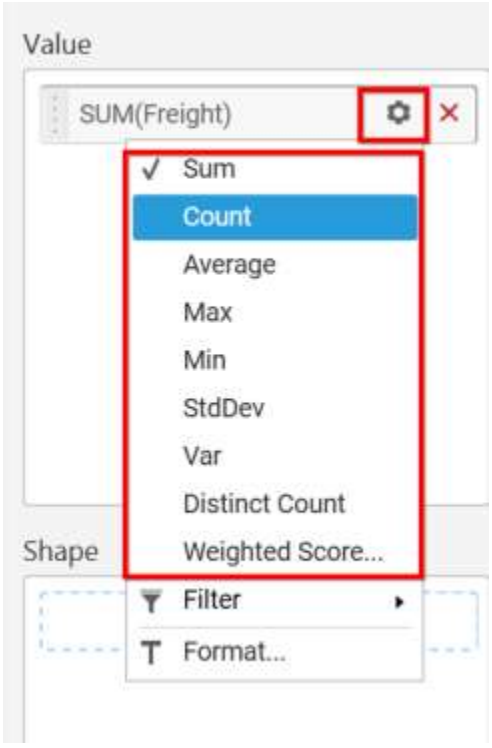
Bind column through drag and drop element from Measures section to Value.



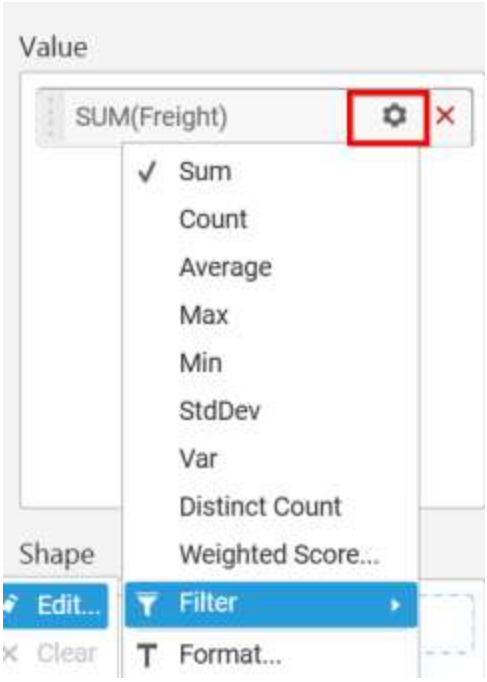
Drag and Drop the elements from sections to Shape.



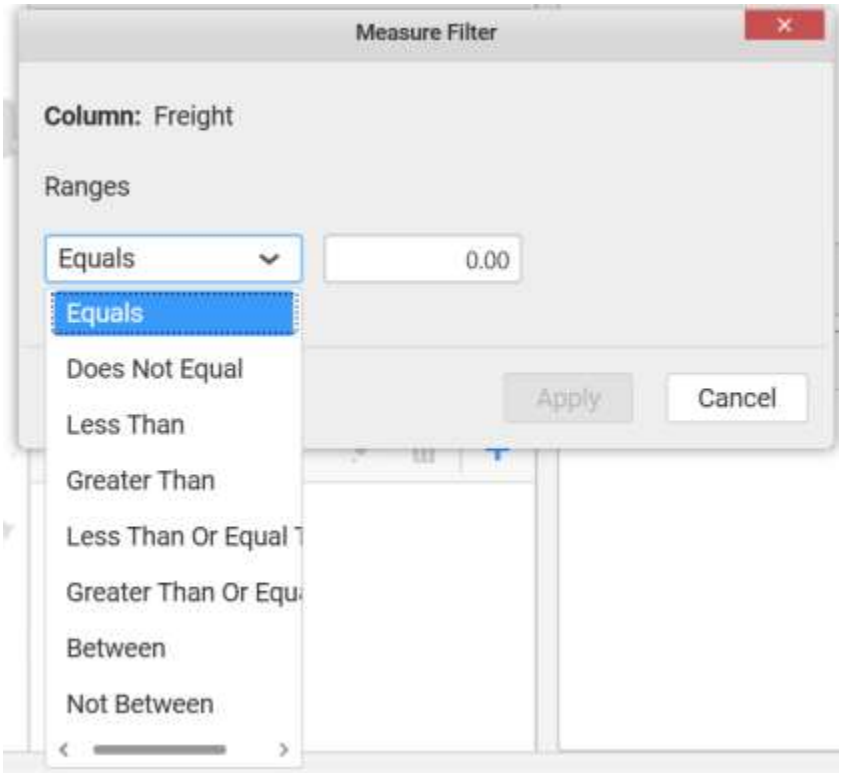
You can use the aggregation function to change the Value of the column.



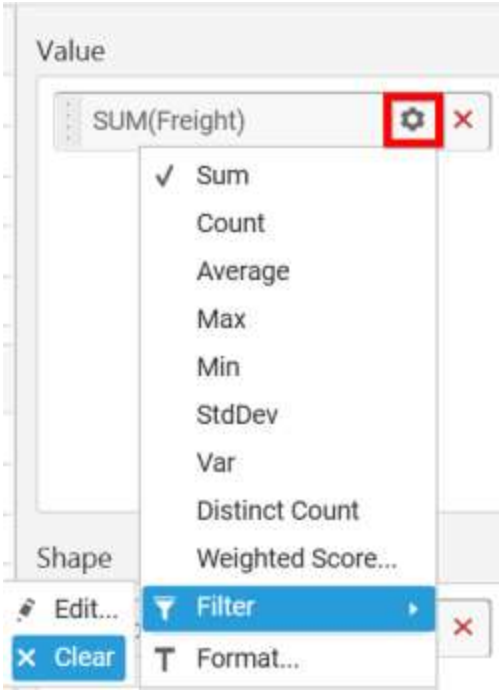
You can use the filter option to enable the Filters by selecting the Edit option.



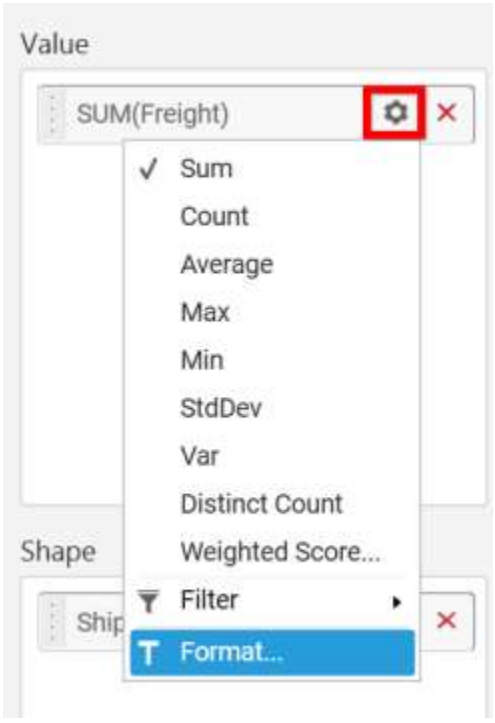
Measure Filters window will be shown to select filter condition.



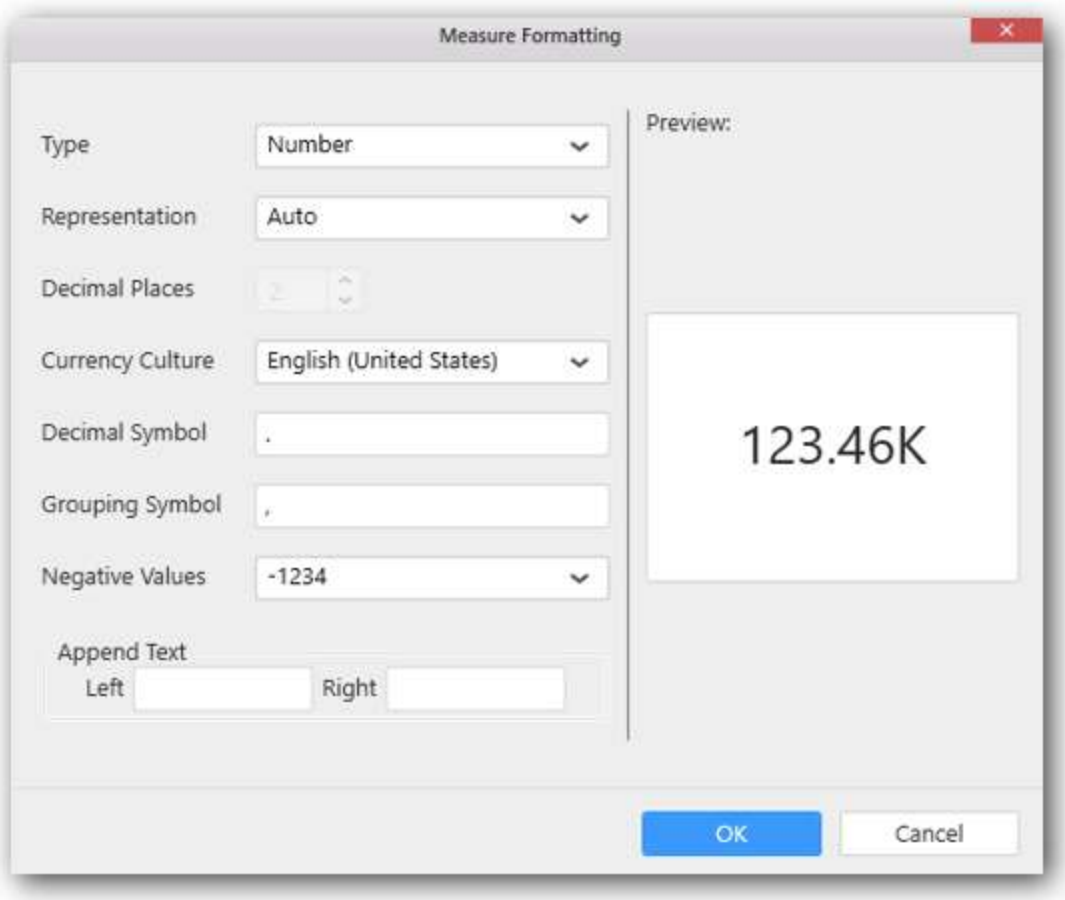
You can clear the filter by selecting the Clear option.



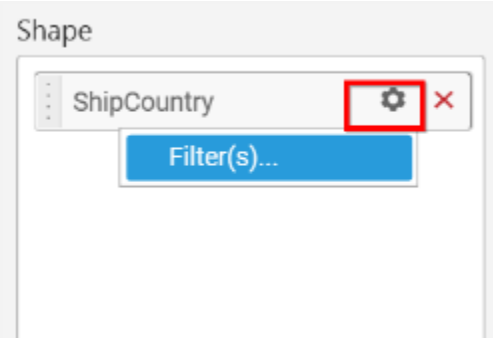
You can format the values by selecting the Format option.



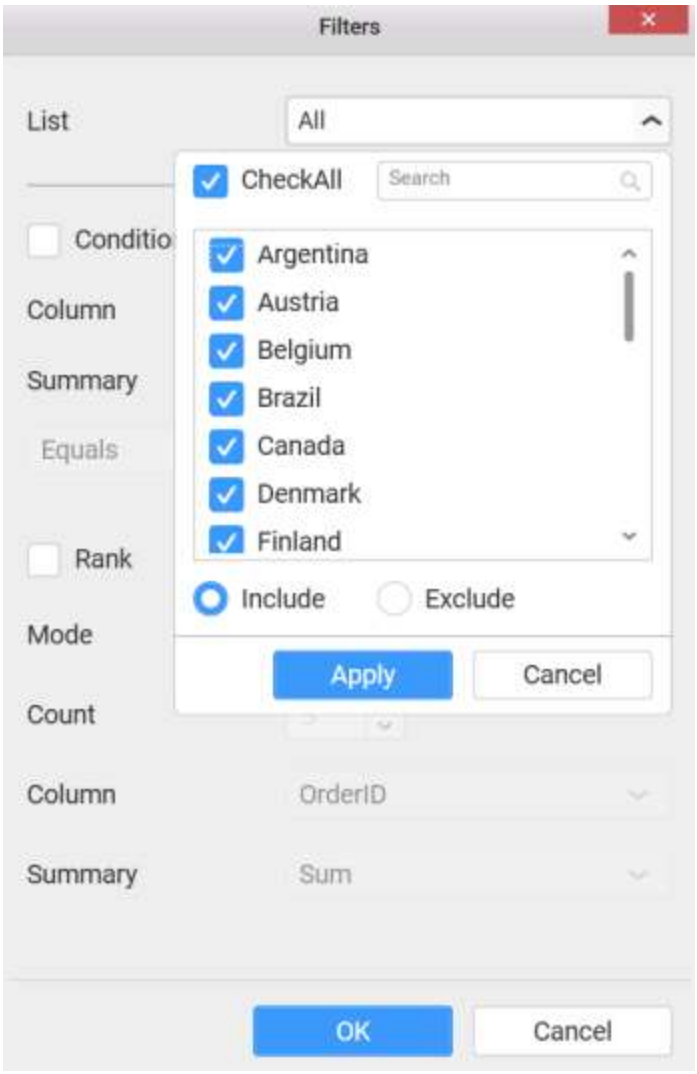
Measure Formatting window will be shown.



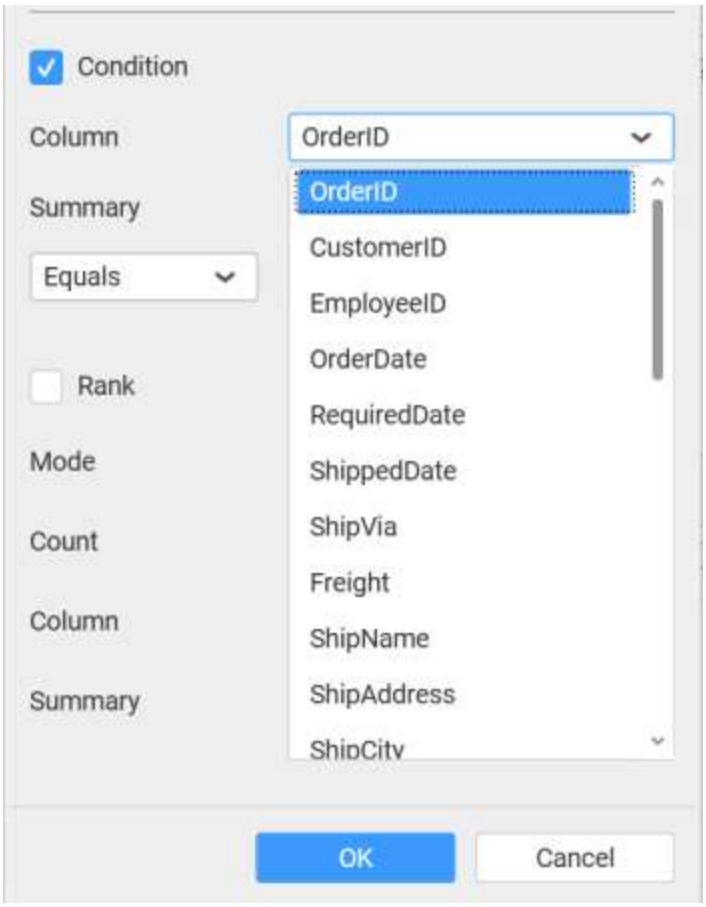
You can use the filters by selecting the Filter(s)... option to rank the elements.

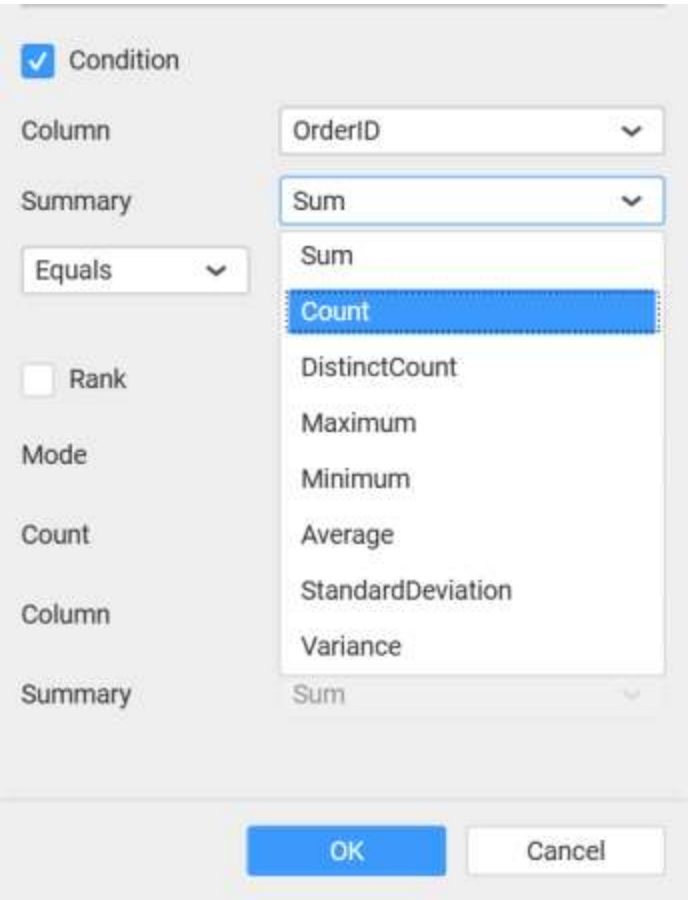


You can select the specific country to filter the element and CheckAll is used either to check all the data or to select the specific data. Include and Exclude is used to include and exclude the elements by selecting the radio button and click the Apply button.

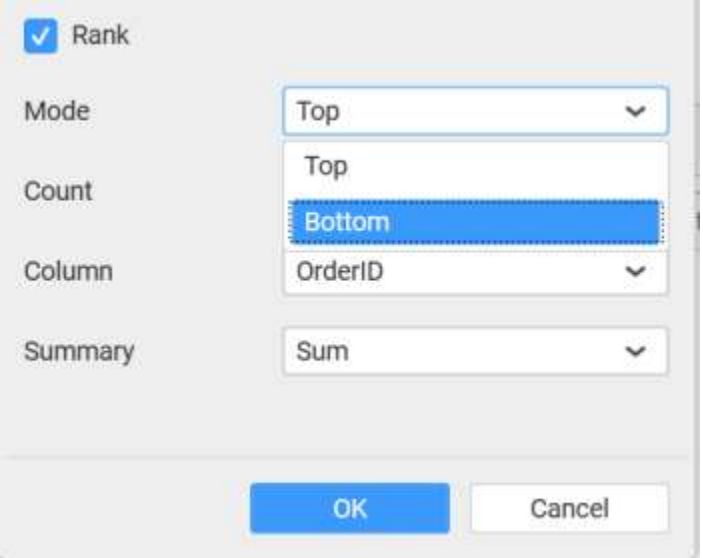


You can select the Condition option to change the Column elements and Summary type by selecting the required column name and summary type.

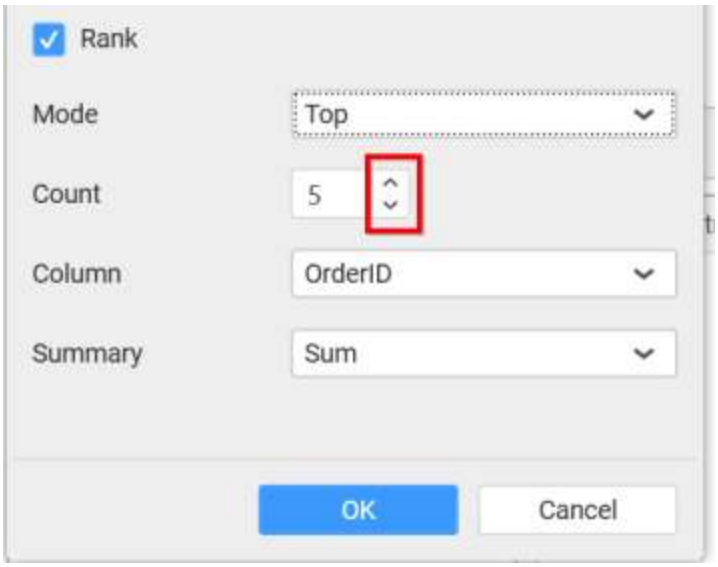




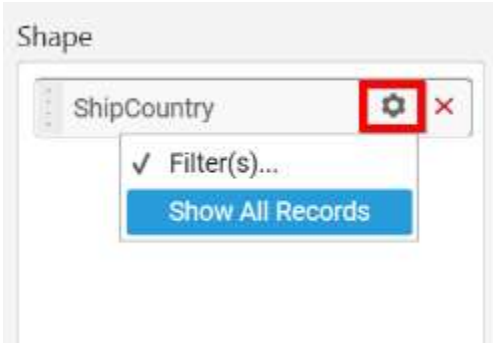
You can select the Rank option to enable filters and select the Mode either top or bottom.



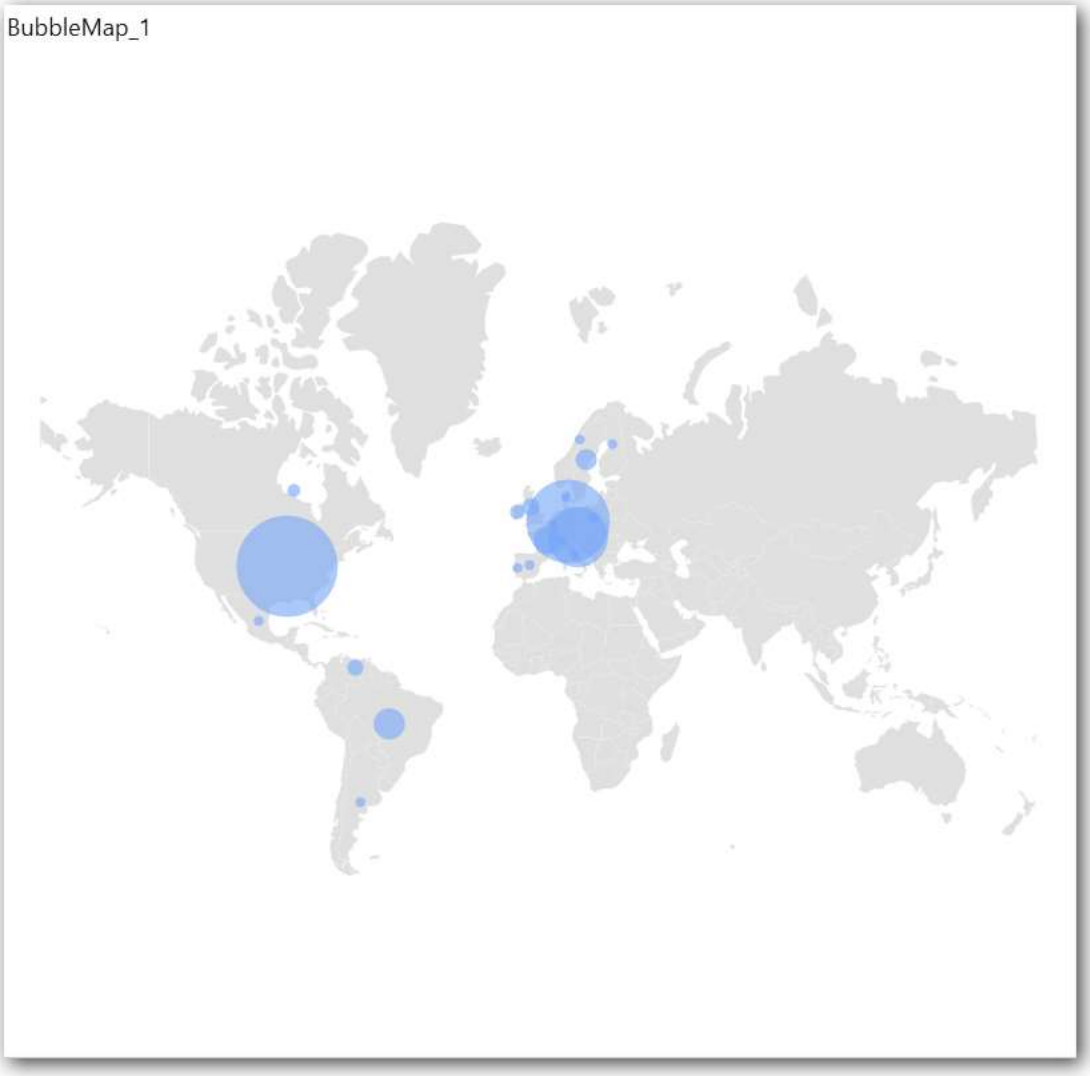
You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.



You can clear the filters by selecting the Show All Records options.



Here is an illustration,

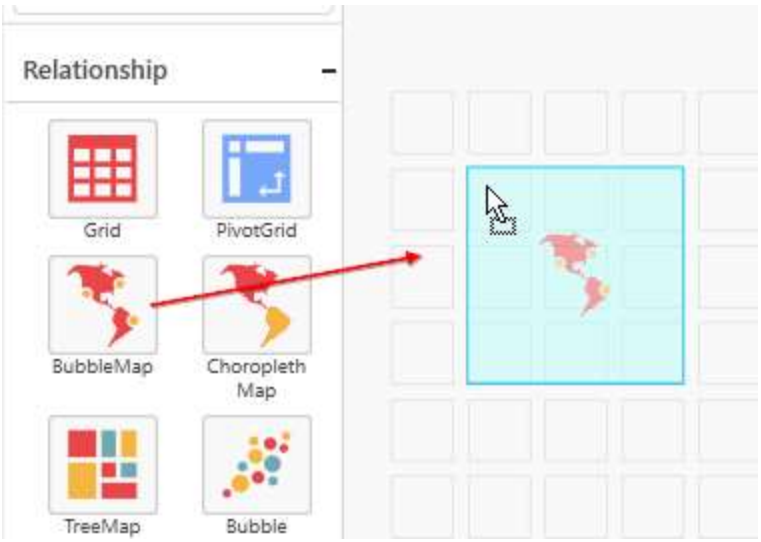


How to configure SSAS data to Bubble Map?

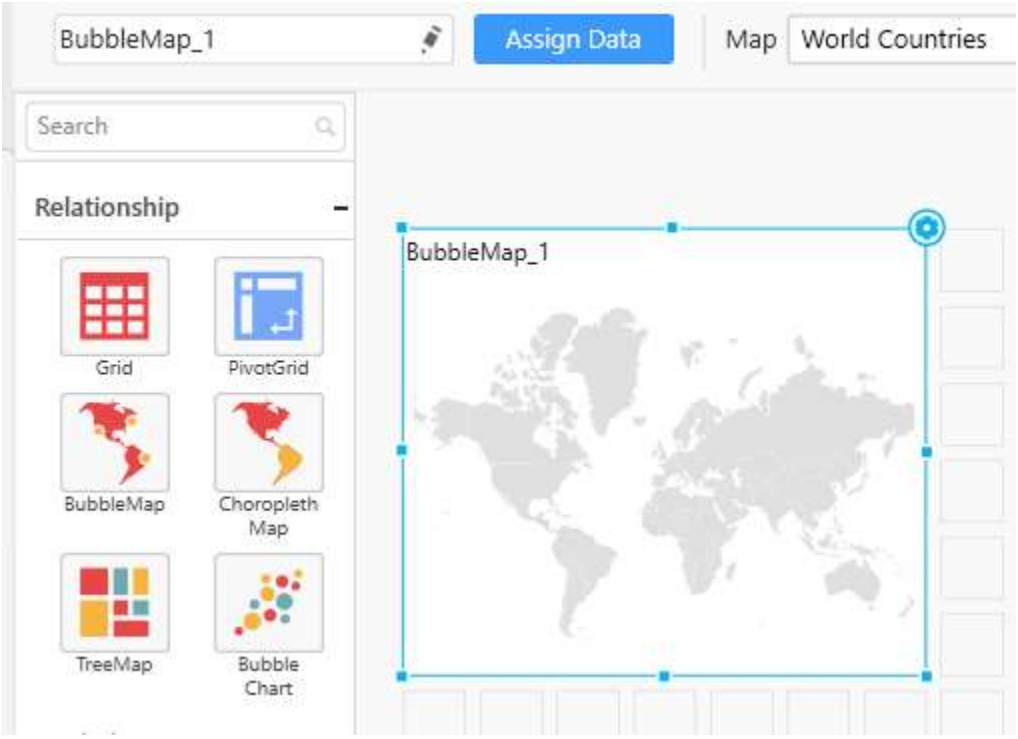
To plot a Bubble Map, a minimum requirement of 1 value and 1 shape is needed.

Following steps illustrates configuring SSAS data to Bubble Map.

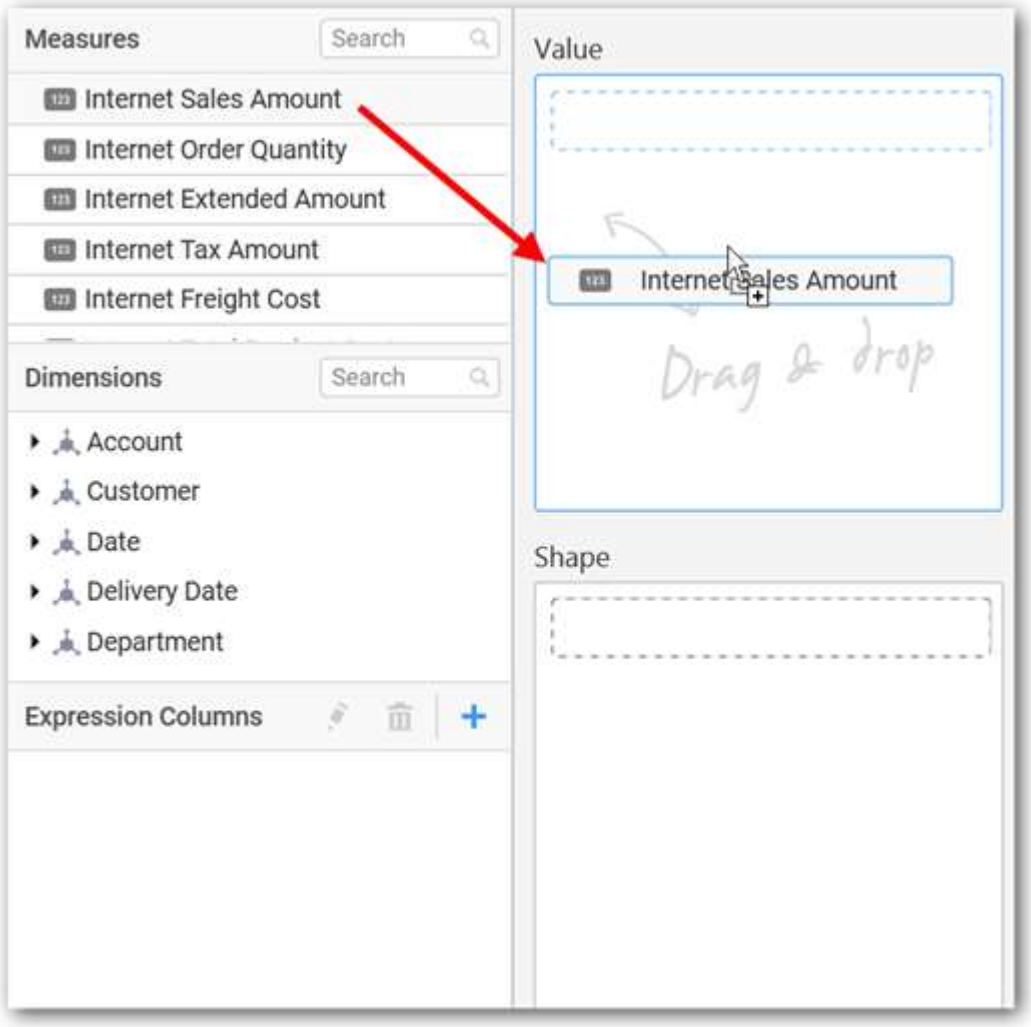
Drag and drop Bubble Map control icon from the Tool box into design panel. You can find control in Toolbox by search.



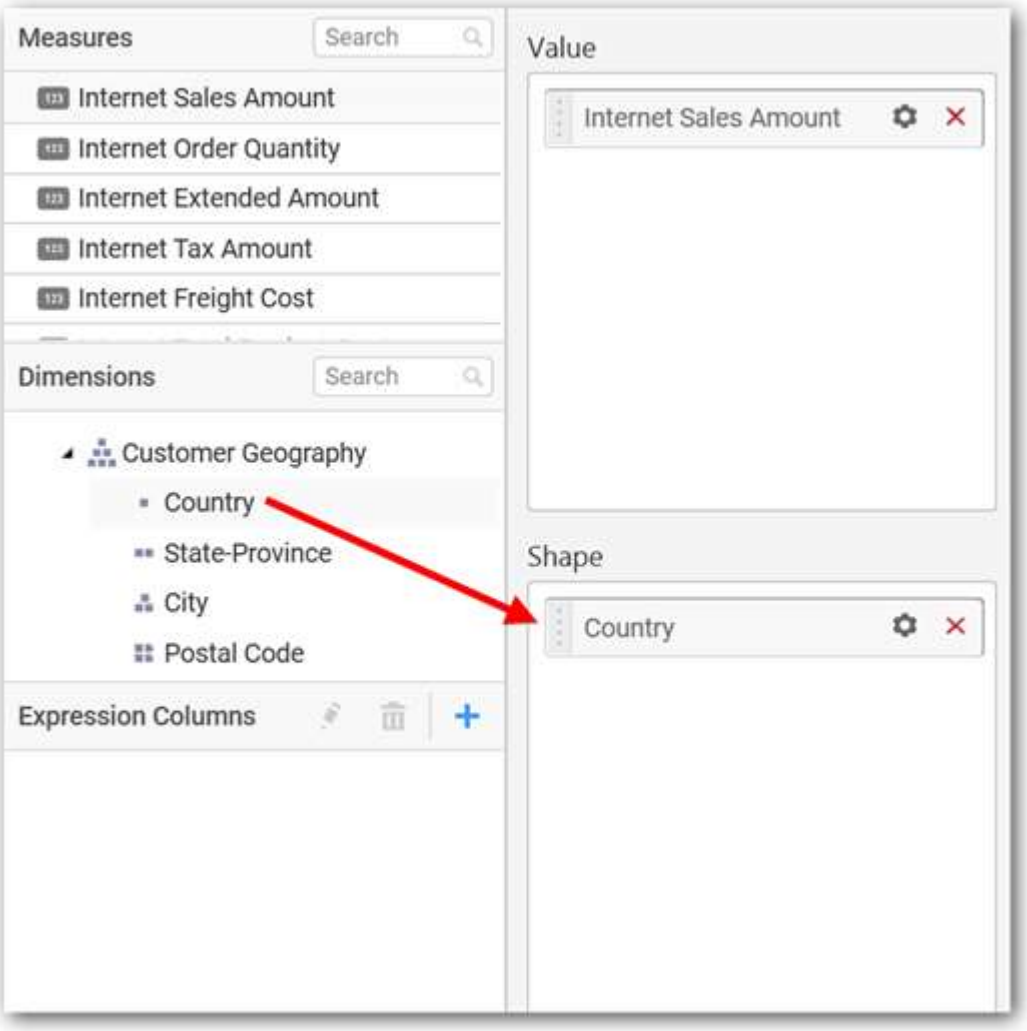
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



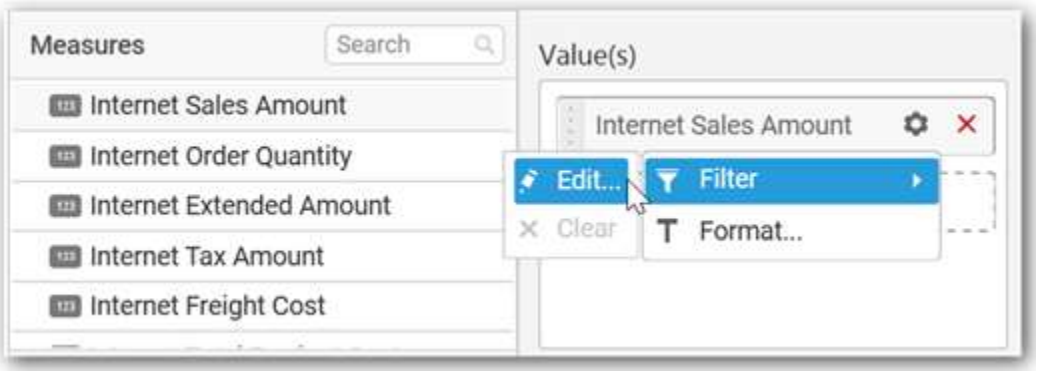
Drag and drop a column under Measures category into Value.



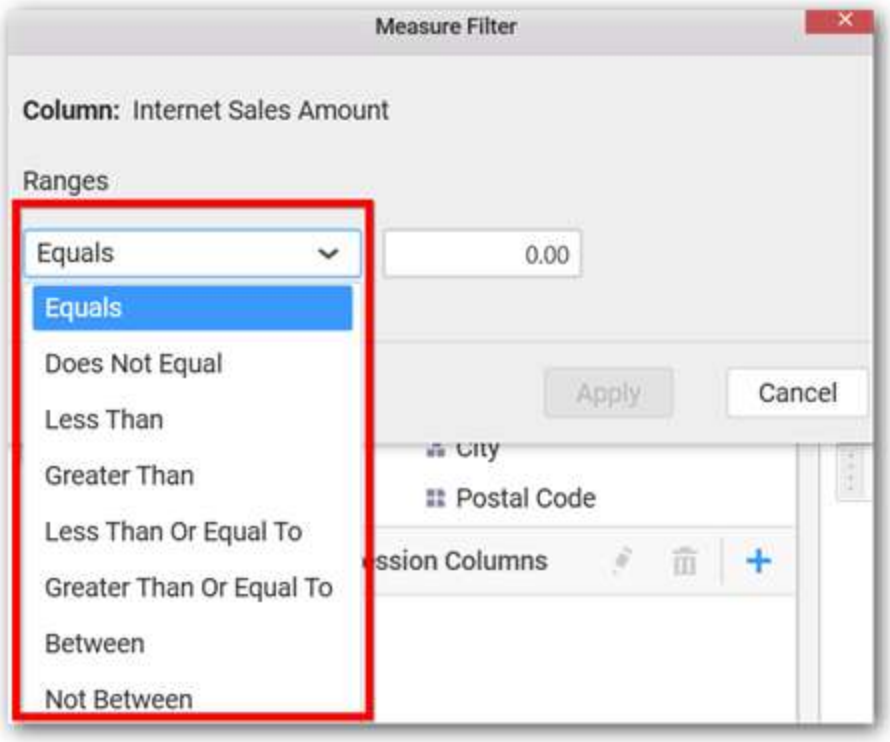
Drag and drop a dimension level or hierarchy column under Dimensions category into Shape.



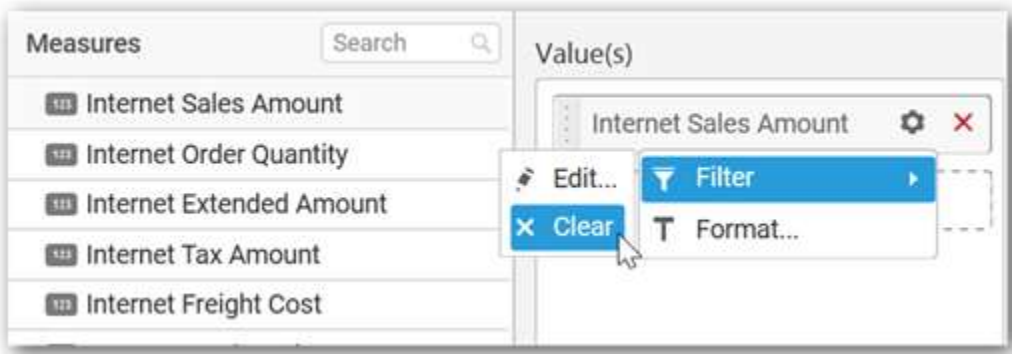
Define the filter criteria to match through choosing Edit option in Filter menu item.



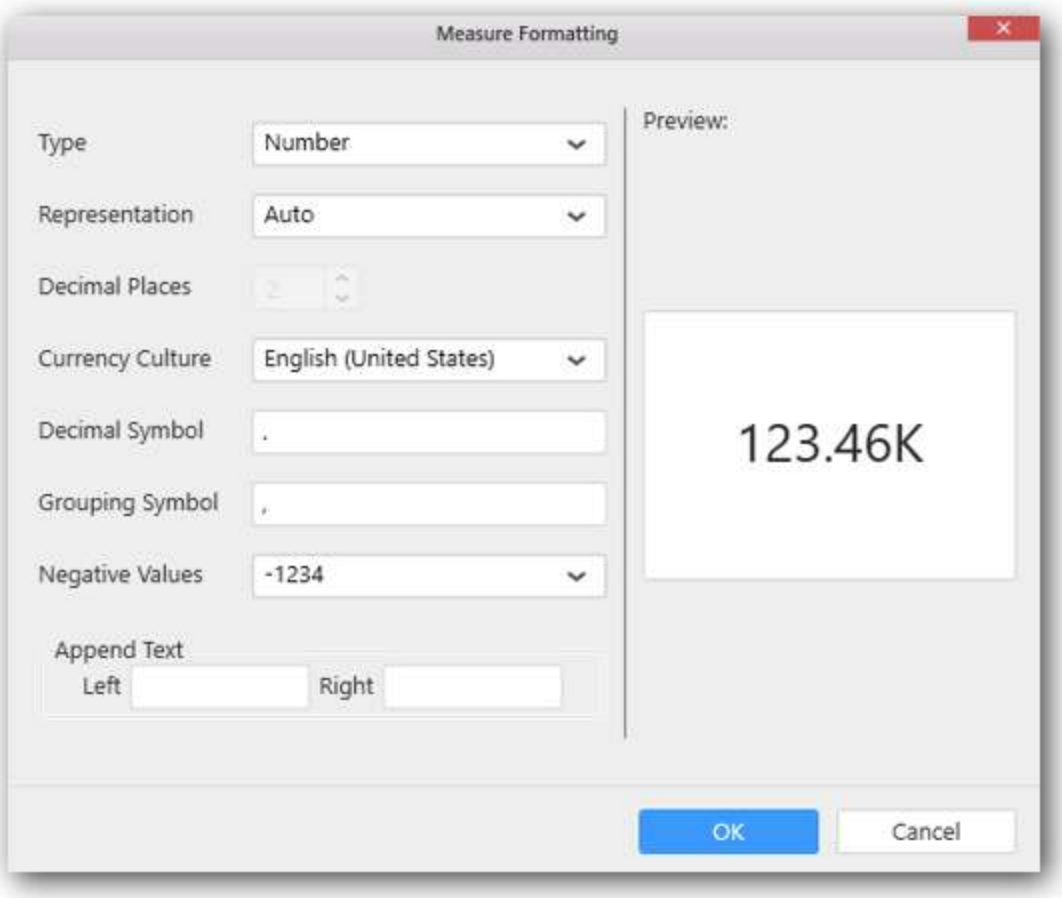
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



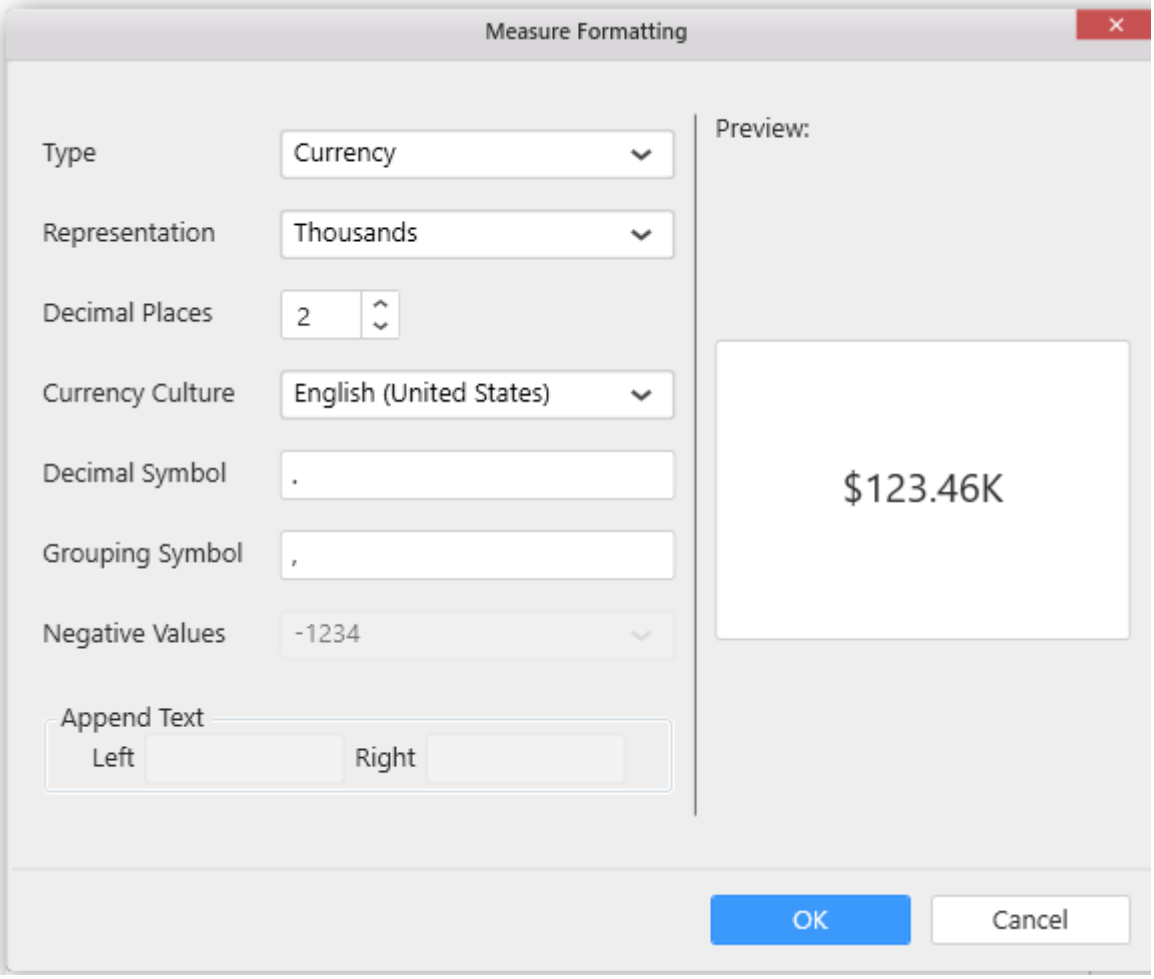
Select Clear option to clear the defined filter.



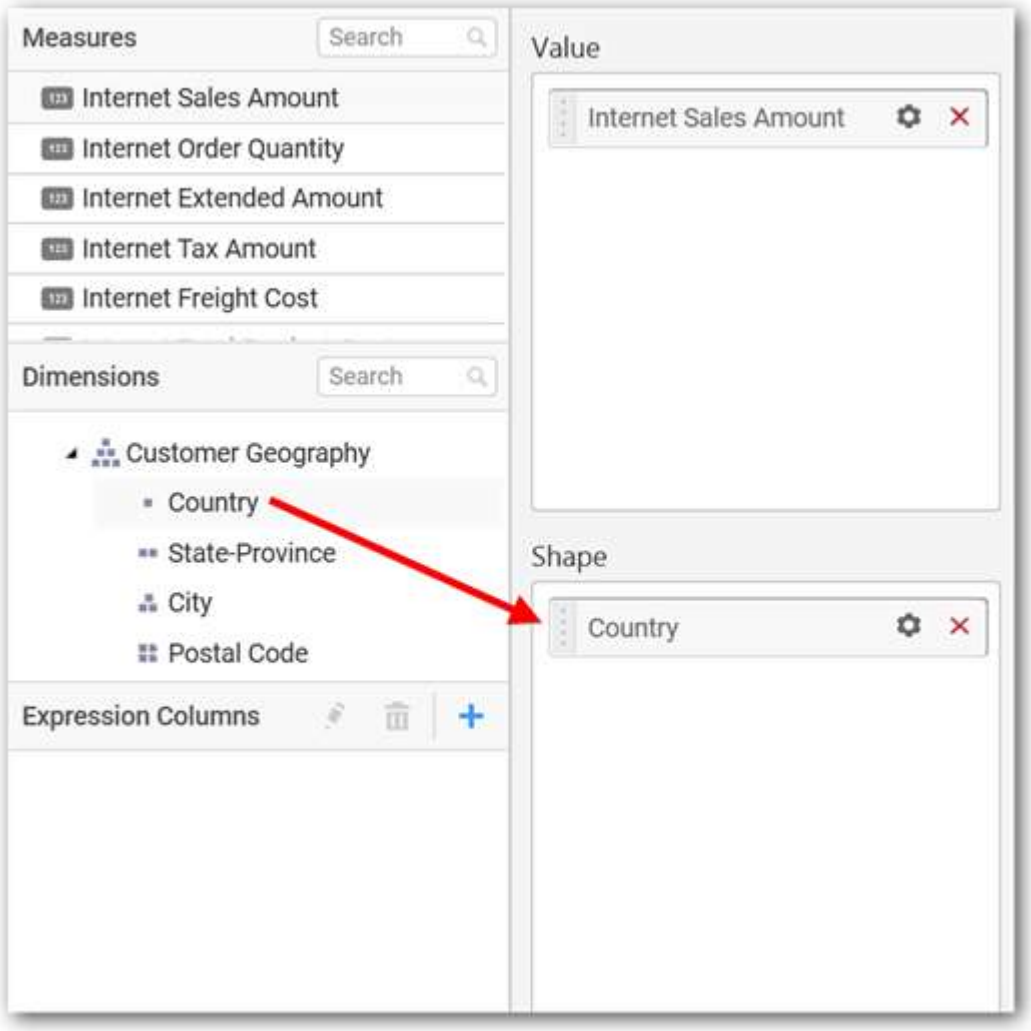
Select Format option to define the display format to the values in the column through Measure Formatting window.



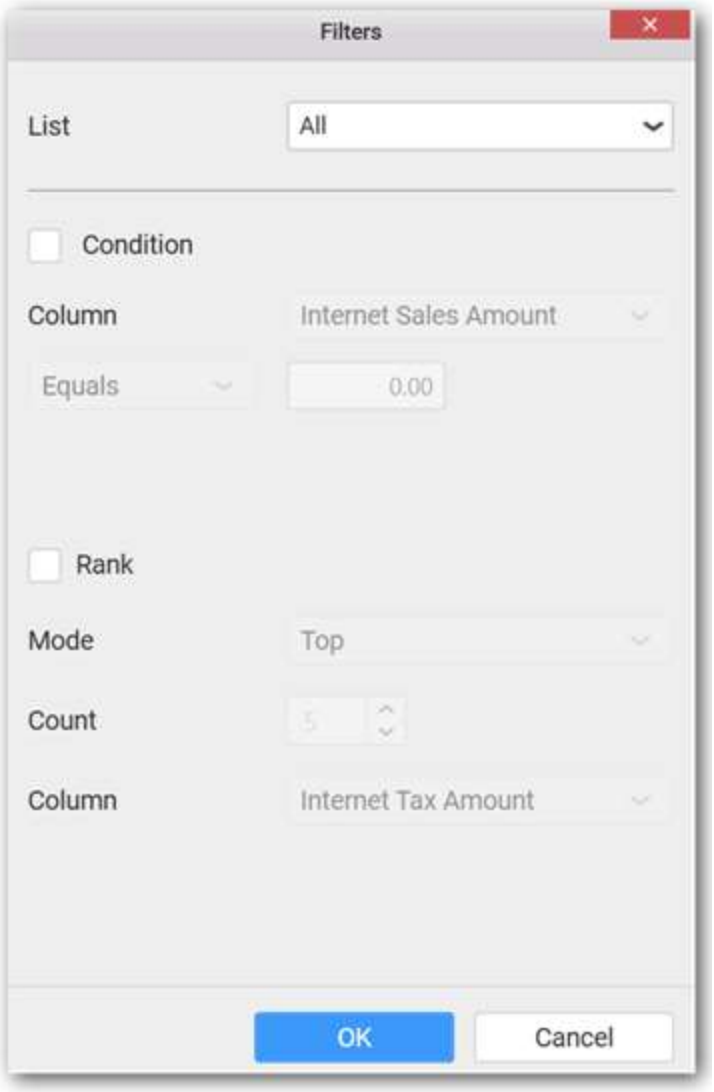
Choose the options you need and click OK.



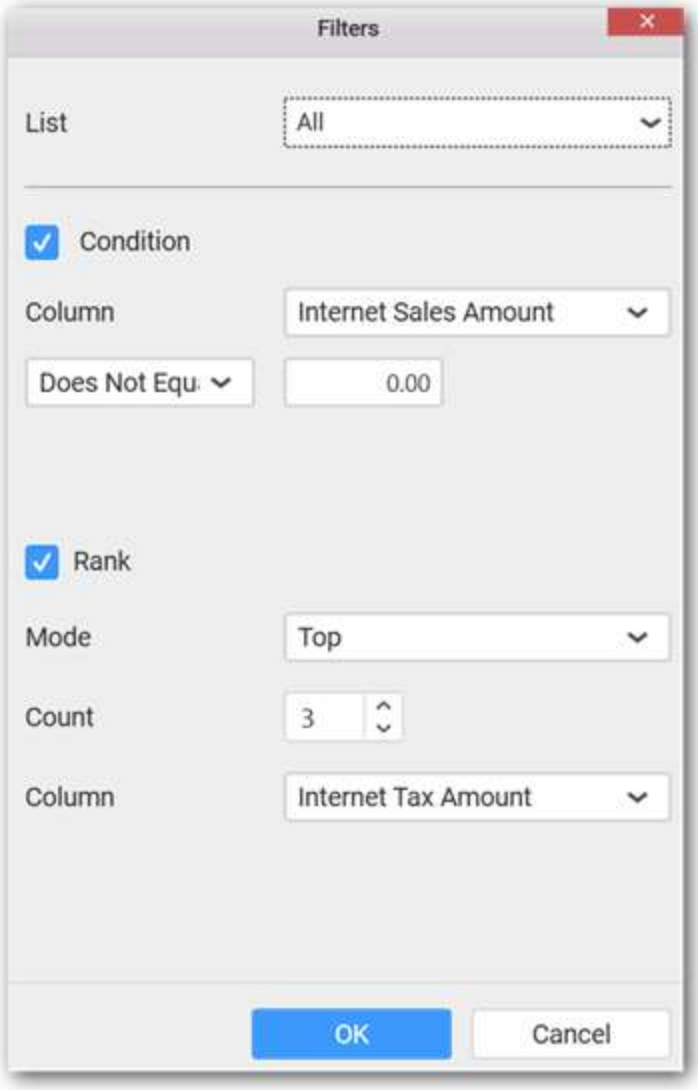
Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



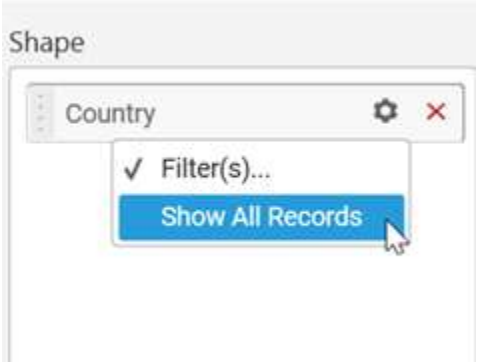
Select Filter(s)… to launch the Filters window.



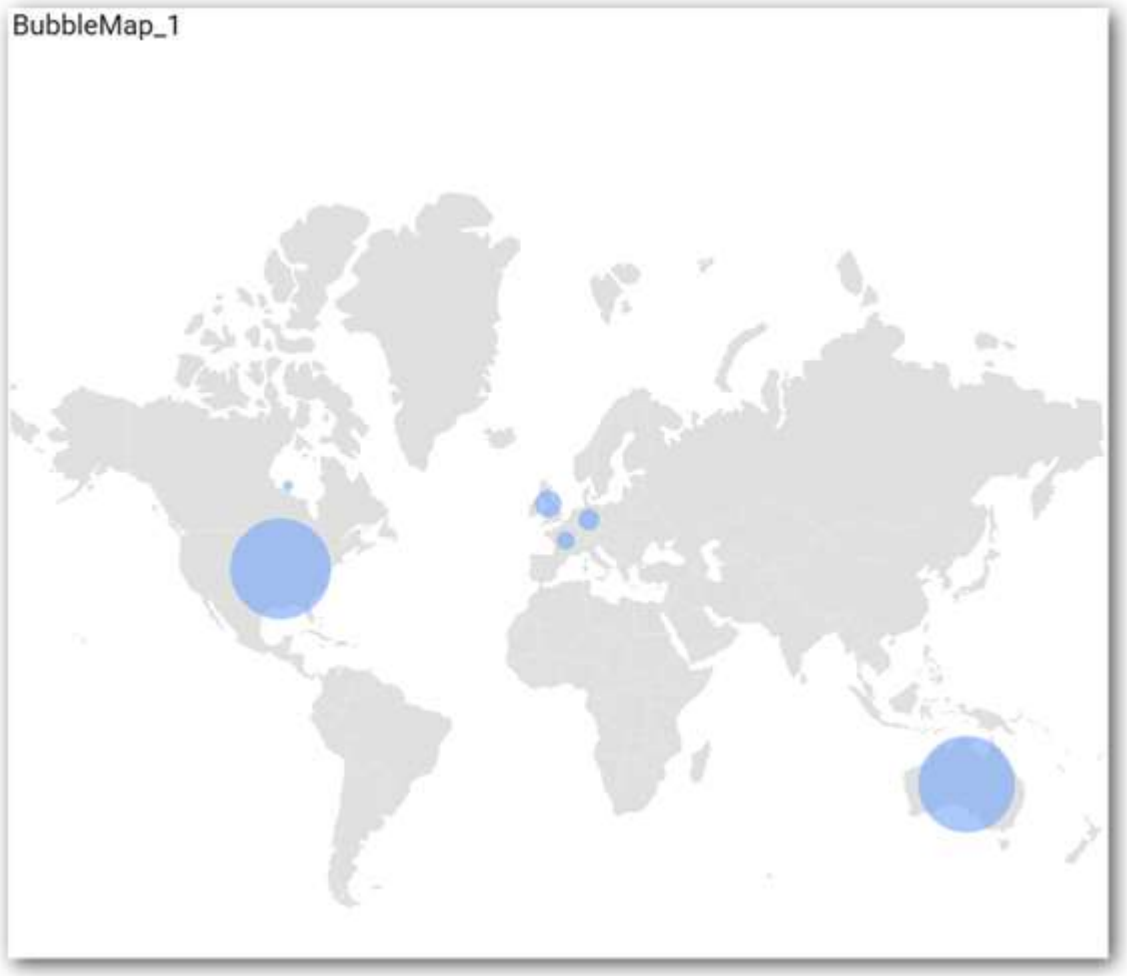
Define the filter Condition and Rank and Click OK.



To show all records again click on Show All Records.



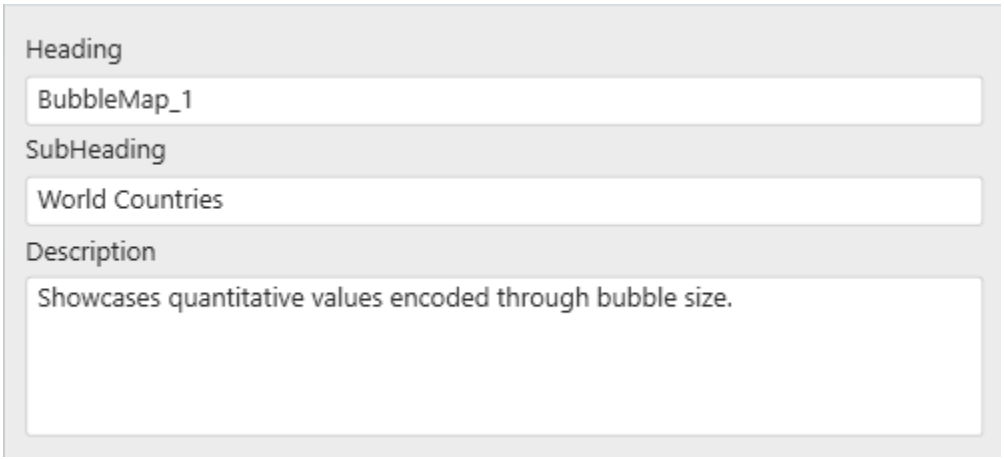
Here is an illustration,



How to format Bubble Map widget?

You can format the Bubble Map for better illustration of the view that you require, through the settings available in Properties pane.

General Settings



Heading

This allows you to set title for this Bubble Map widget.

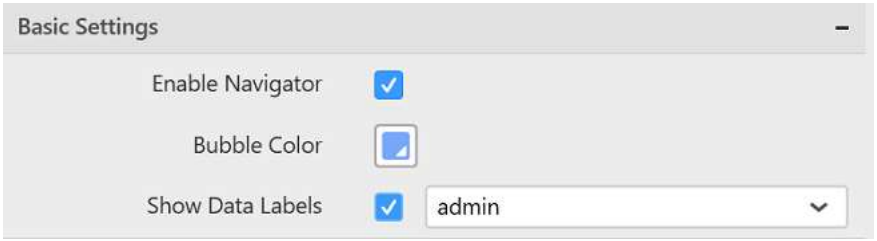
SubHeading

This allows you to set sub-title for this Bubble Map widget.

Description

This allows you to set description for this Bubble Map widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings



Enable Navigator

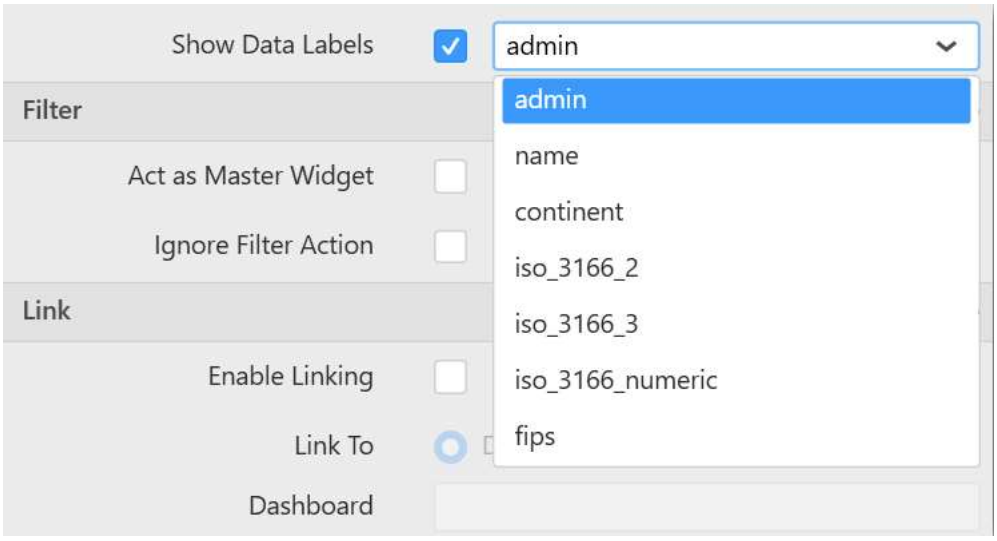
You can enable/ disable the map navigator in Bubble Map.

Bubble Color

You can customize the bubble color from the set of predefined color palette.

Show Data Labels

You can enable or disable the data labels in map widget by using Show Data Labels property. After enabling this property, a dropdown menu is enabled for choosing the shape properties (i.e. name, admin, iso_3166, etc.) which is going to be displayed as data label.



The properties displayed in the DropDownList are changed based on selected shapes.

Filter Settings

Act as Master Widget

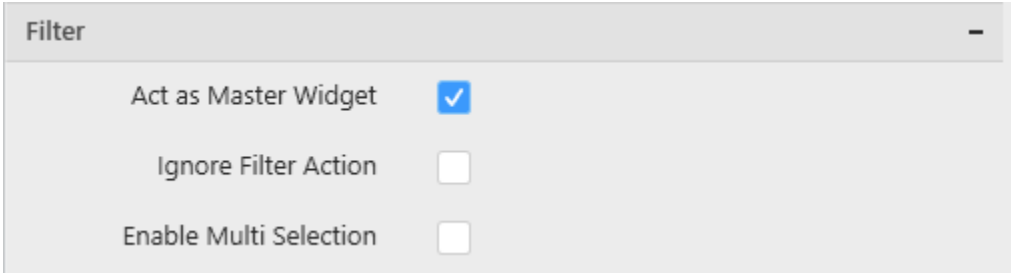
This allows you to define this Bubble Map widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this Bubble Map widget to ignore responding to the filter actions applied on other widgets in dashboard.

Enable Multi Selection

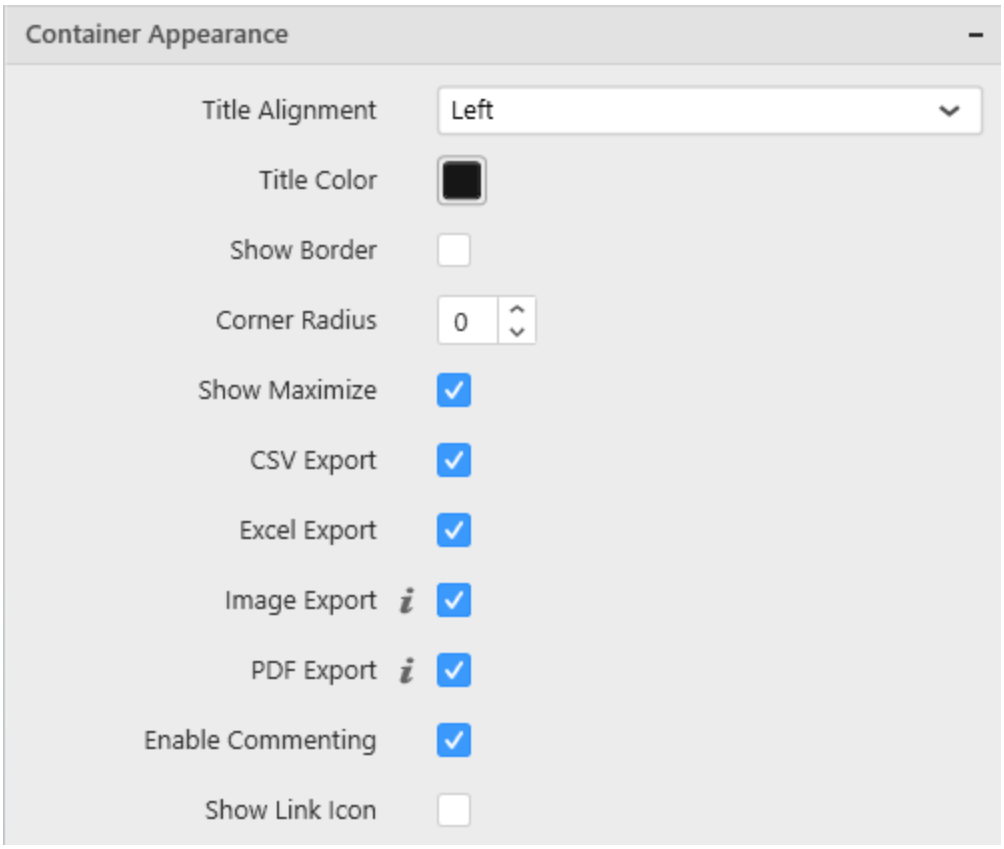
Enable Multi Selection is displayed only when you enable the Act as Master Widget option. This allows you to select multiple shapes in the Bubble Map widget.



Link Settings

You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this Bubble Map widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the Bubble Map widget.

CSV Export

This allows you to enable/disable the CSV export option for this Bubble Map widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this Bubble Map widget. Enabling this allows you to export the summarized data of the widget view to XLSX format.

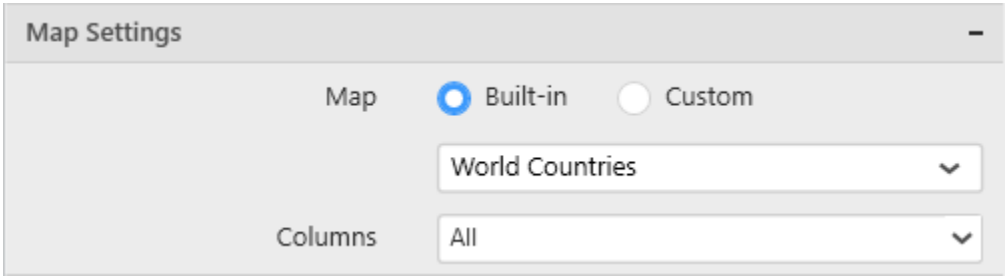
Image Export

This allows you to enable/disable the image export option for this Bubble Map widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Map Settings

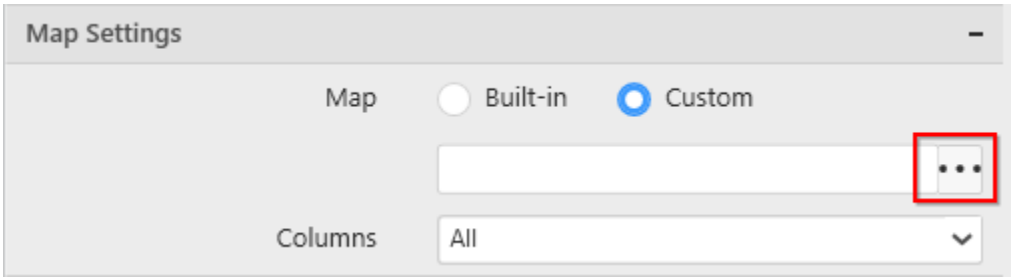


Built-in

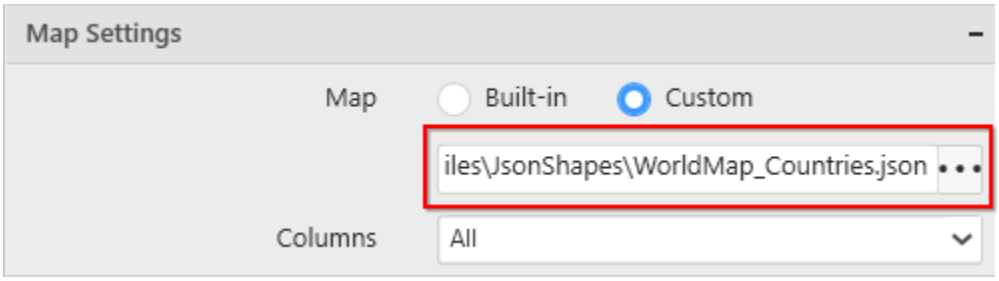
You can choose from the available 14 map shape files shipped along with dashboard designer for visualizing map data.

Custom

This option allows you to load any map shape files (i.e. .json files) into Bubble Map widget. For loading custom map files, you need to choose Custom option from Map settings. After enabling this option, browse icon is enabled for loading custom shape file.

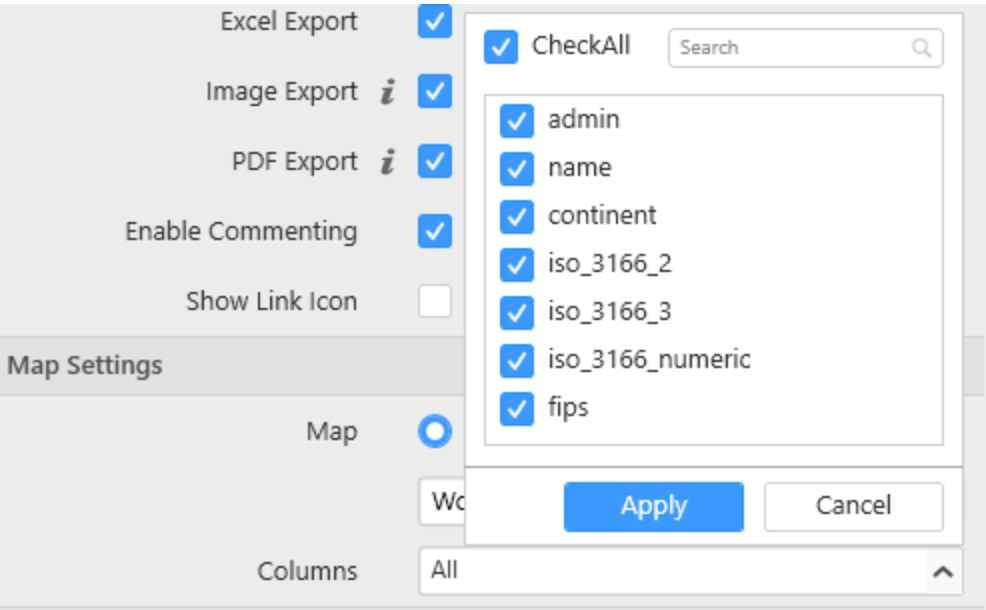


Using browse icon, we can browse and load the required shape file in Map.



Columns

You can choose multiple column fields from the list of field names supported by the loaded map shape that can be mapped with the data source connected. By default all column fields have been selected and you can select/unselect from columns drop down menu.



Choropleth Map

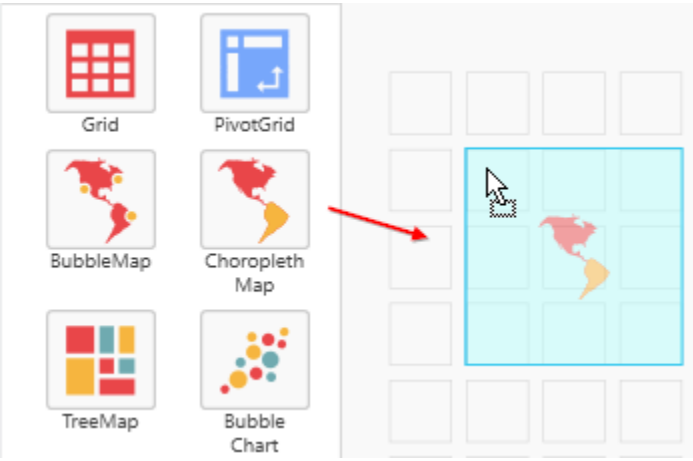
Choropleth Map allows you to showcase quantitative values encoded through color scale.

How to configure flat table data into Choropleth Map?

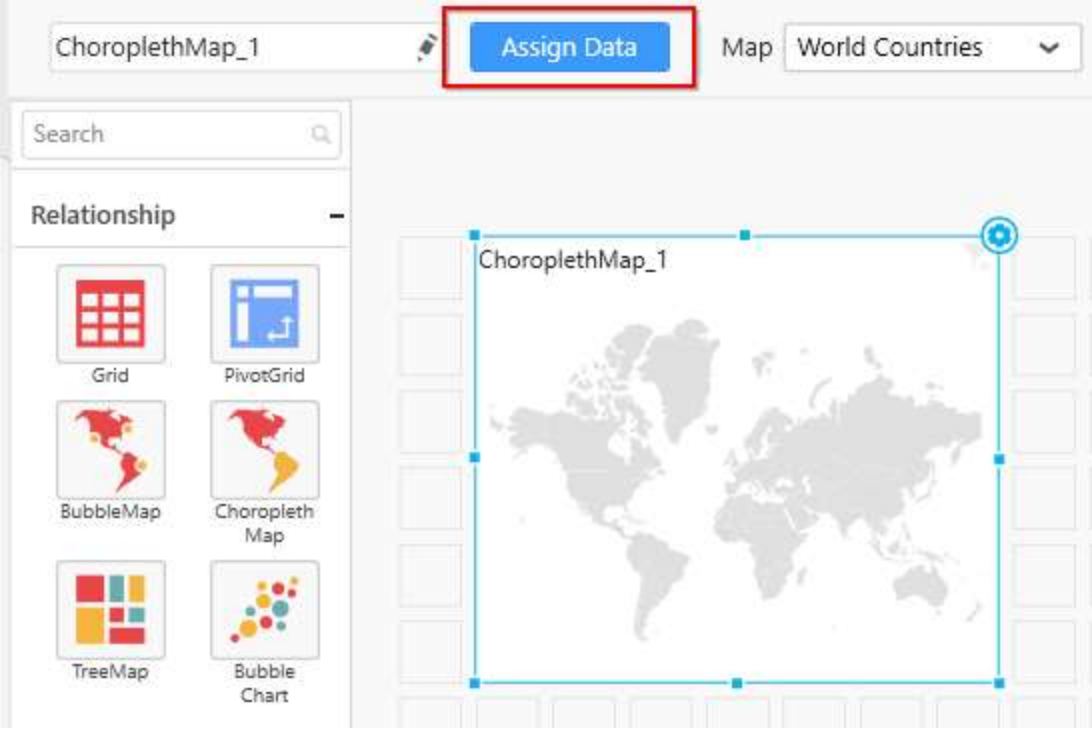
To plot a Choropleth Map, a minimum requirement of 1 value and 1 shape is needed. This holds one additional block Calculations in Data Pane. Dropping a dimension will display each region split by each of its item.

The following procedure illustrates data configuration of Choropleth Map.

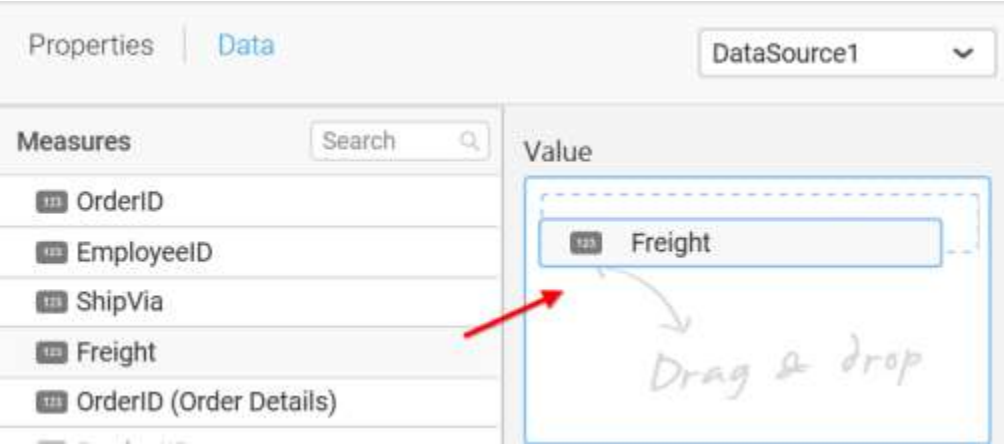
Drag and drop Choropleth Map control icon from the Tool box into design panel. You can find control in Toolbox by search.



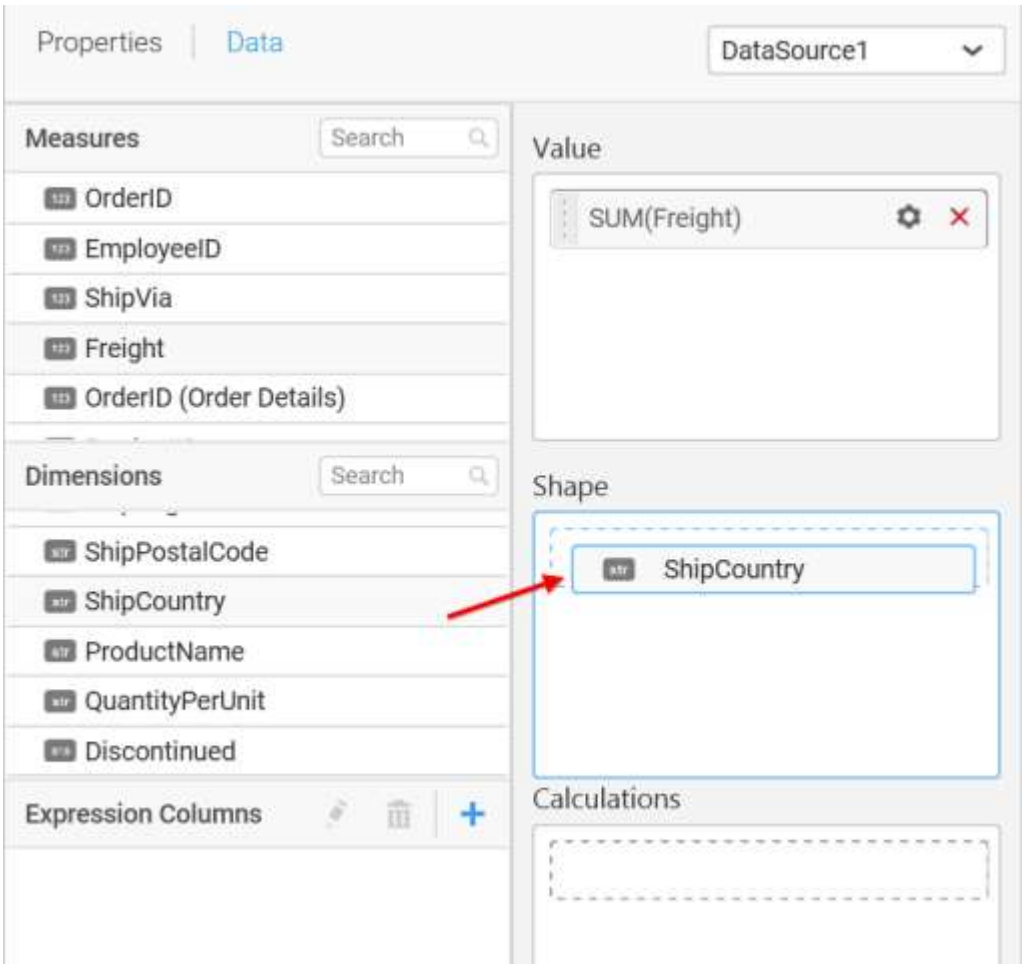
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



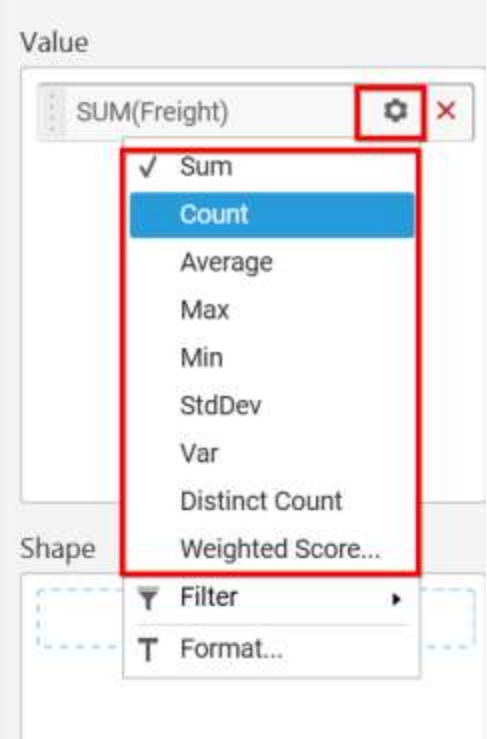
Bind column through drag and drop element from Measures section to Value.



Drag and Drop the elements from sections to Shape.

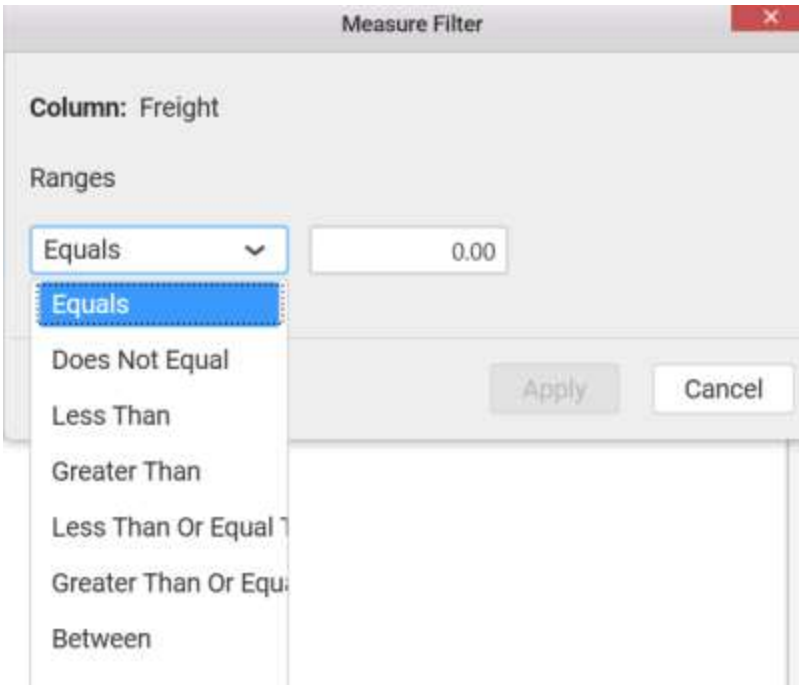


You can use the aggregation function to change the Value of the column.

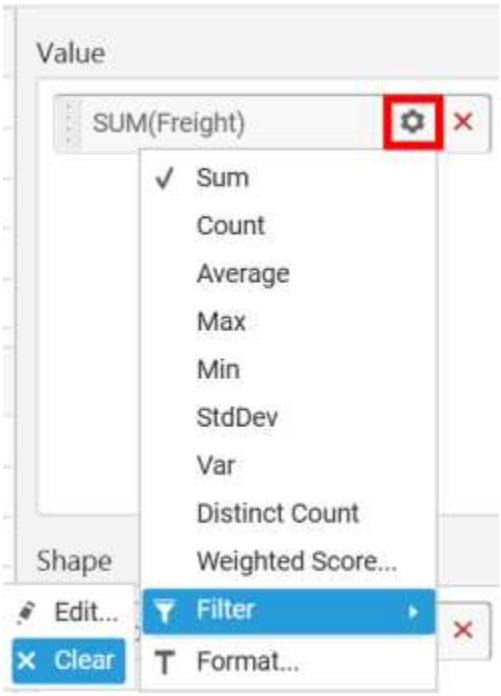


You can use the Filter option to enable filters by selecting the Edit option.

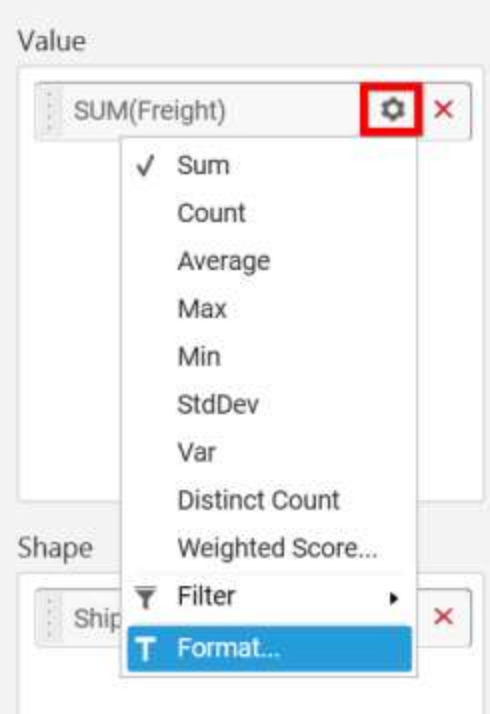
Measure Filter window will be shown.



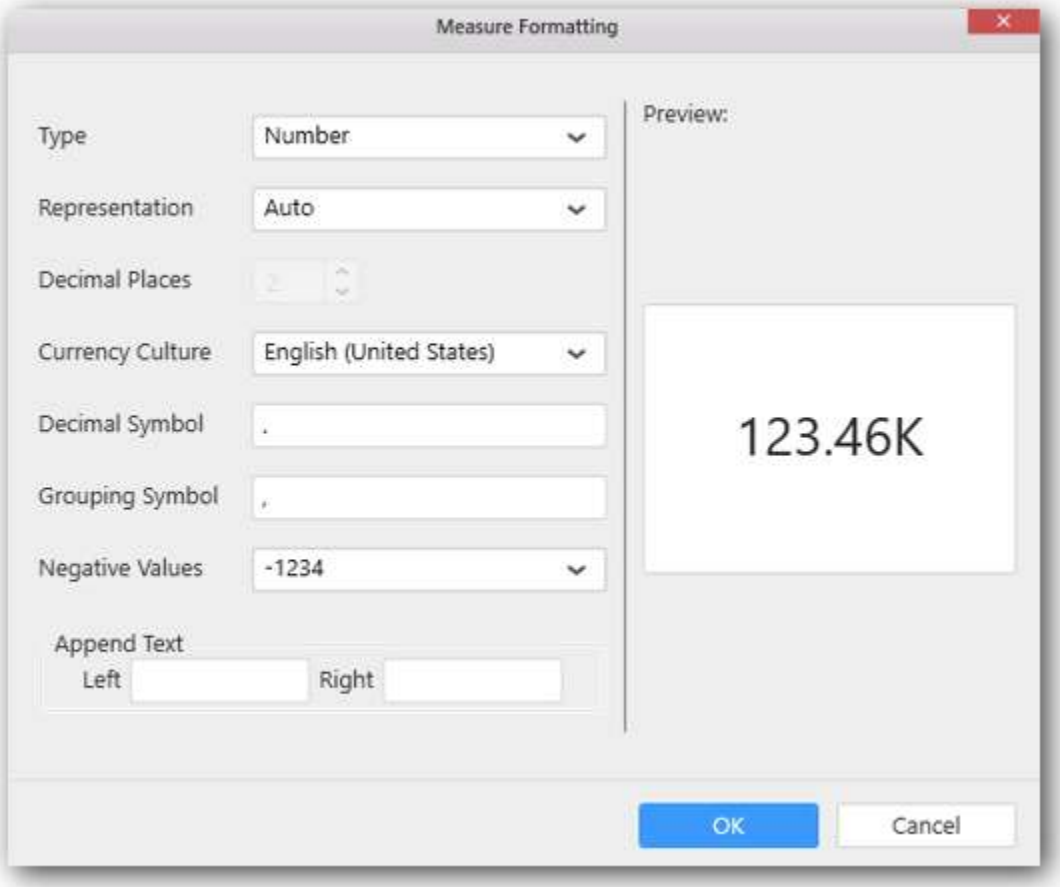
You can clear the filter by selecting the Clear option.



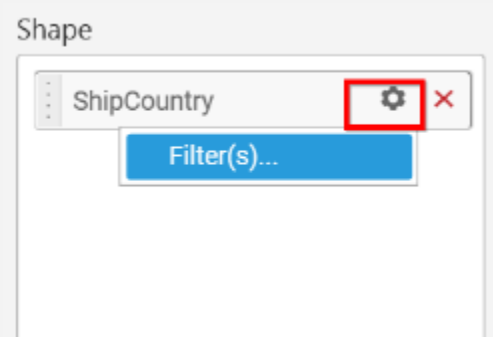
You can format the values by selecting the Format option.



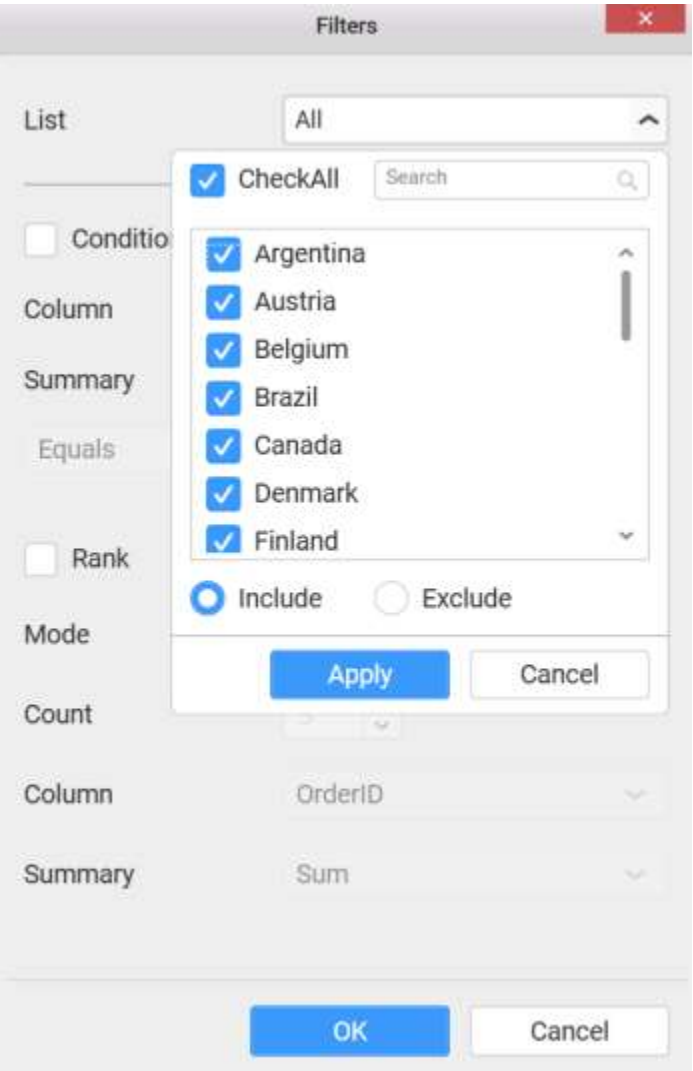
Measure Formatting window will be shown.



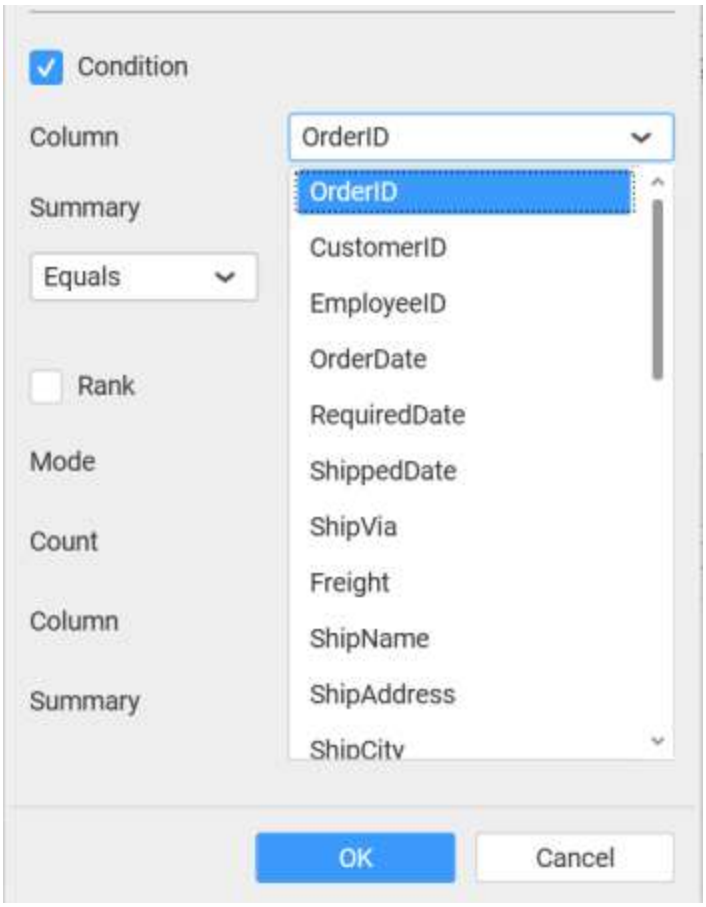
You can use the filters by selecting the Filter(s)... option to rank to the elements.

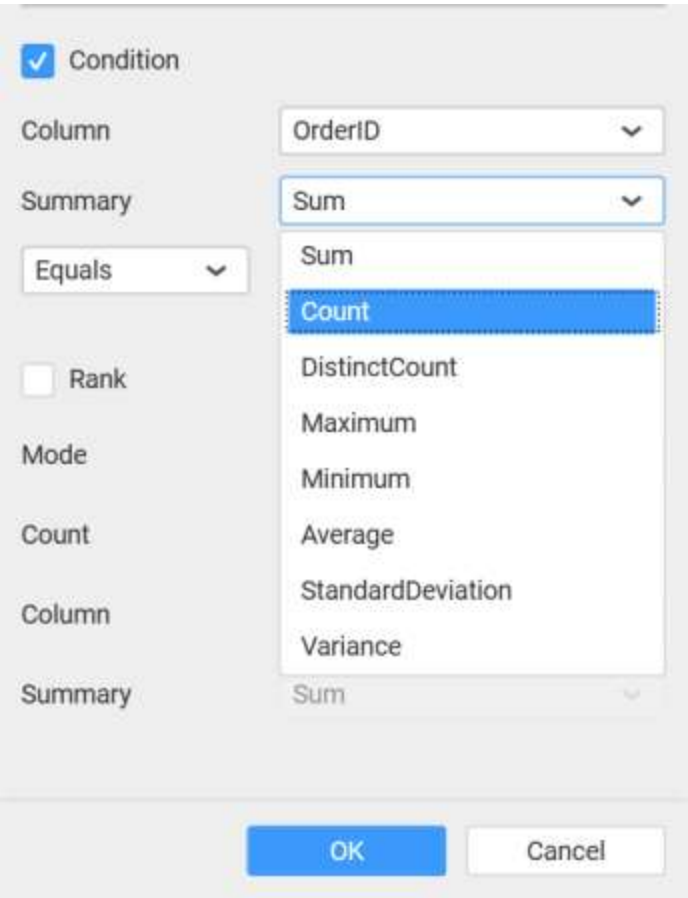


You can select the specific country to filter the element and CheckAll is used either to check all the data or to select the specific data. Include and Exclude is used to include and exclude the elements by selecting the radio button and click the Apply button.

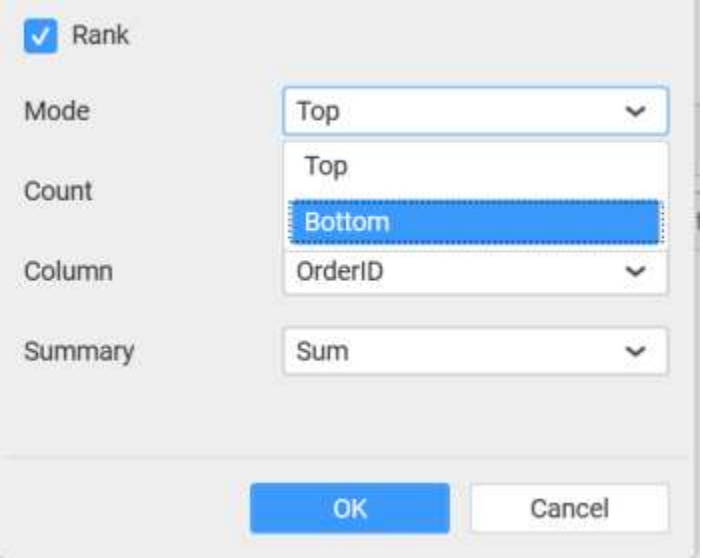


You can select the Condition option to change the Column elements and Summary type by selecting the required column name and summary type.

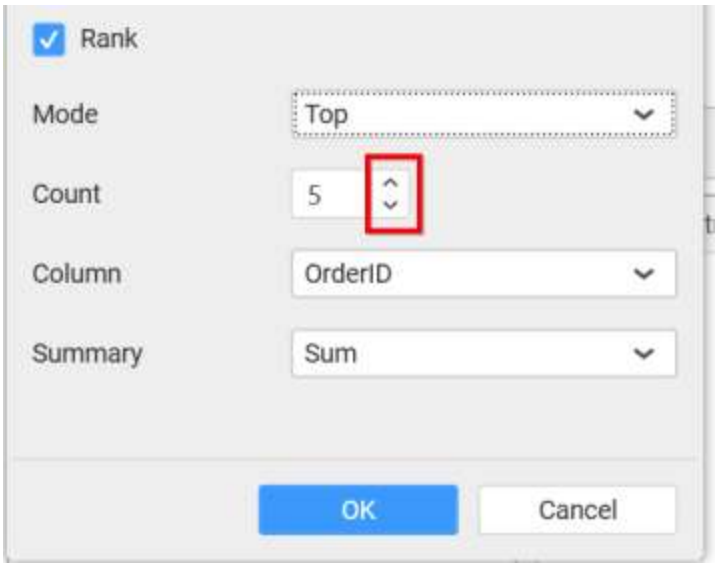




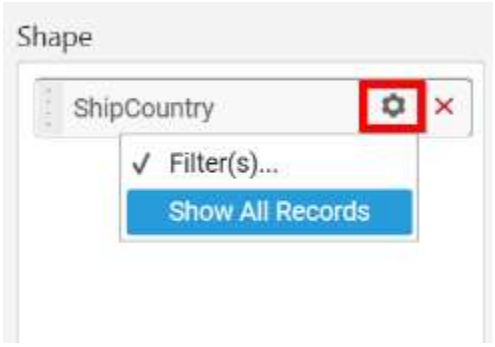
You can select the Rank option to enable filters and select the Mode either top or bottom.



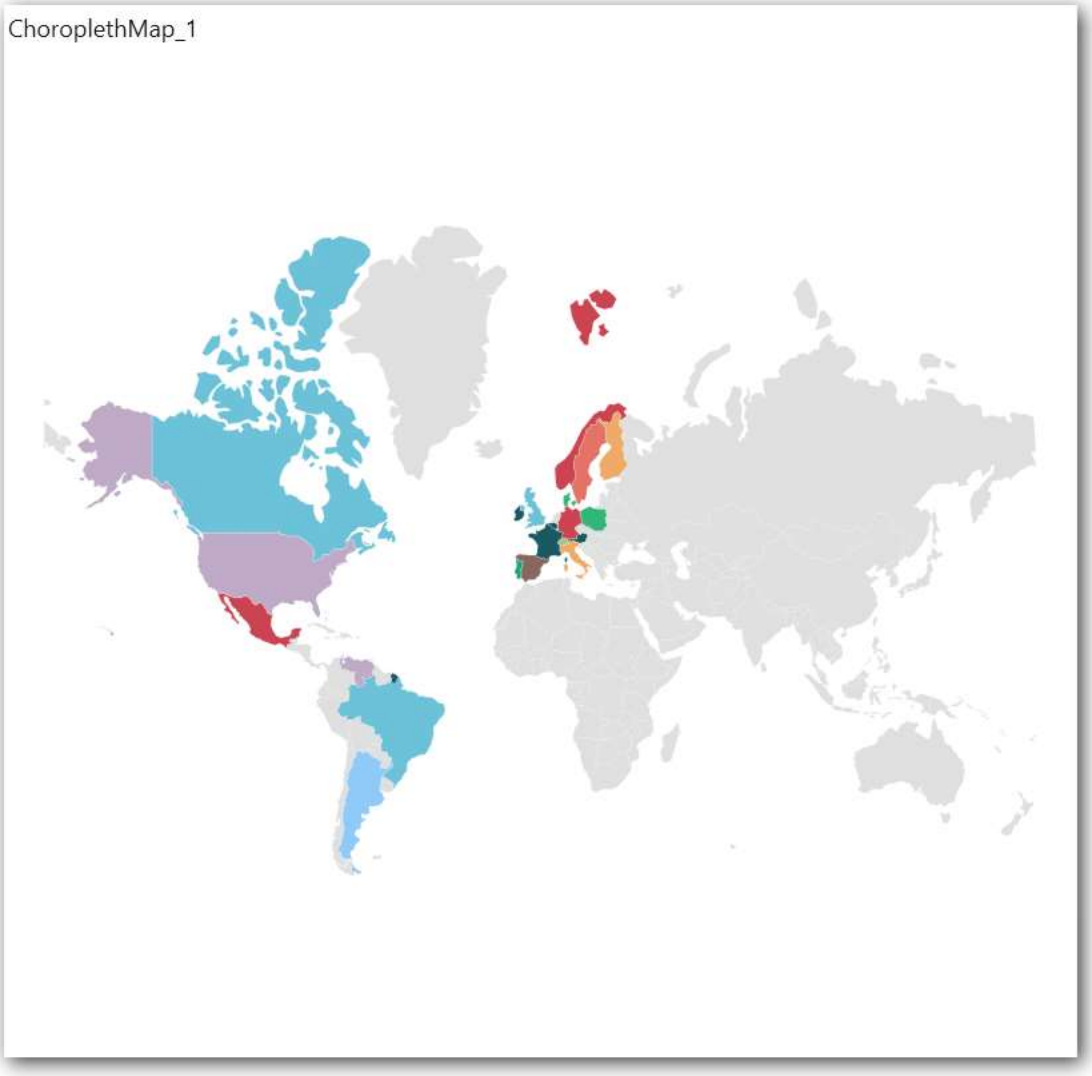
You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.



You can clear the filters by selecting the Show All Records options.



You can set colors to countries.

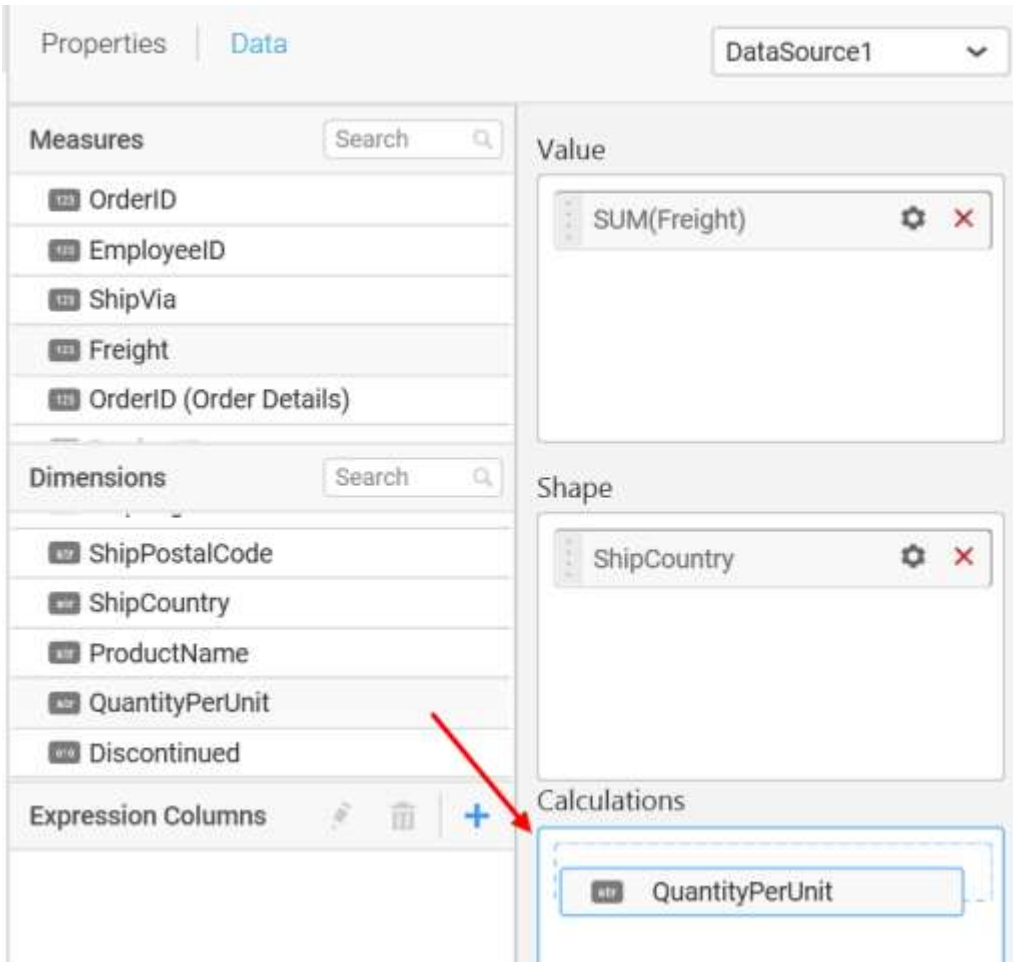


In the above screenshot, colors are applied to countries based on calculations. The default color palette is used to set the colors.

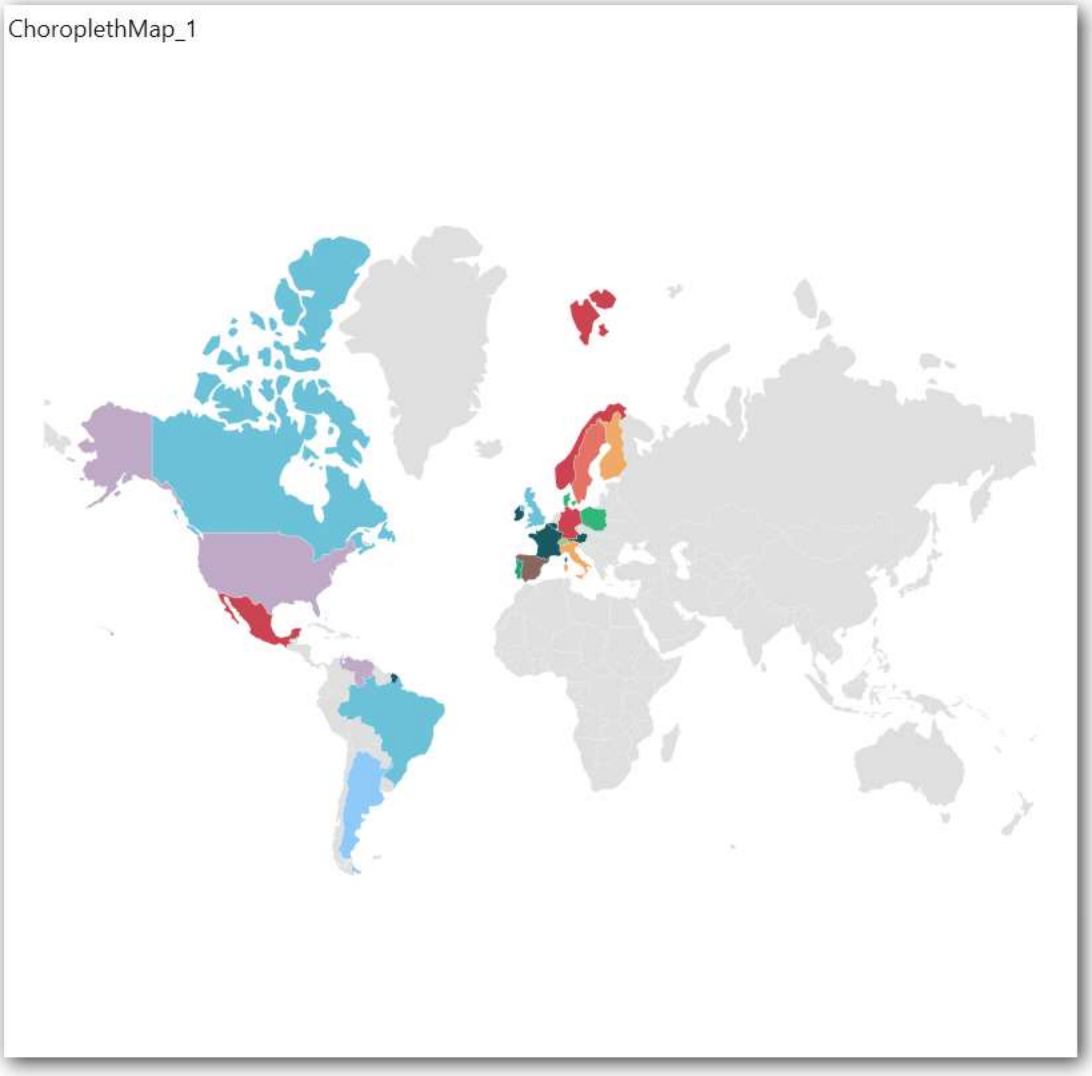
On hovering the mouse over a country, the tooltip will display its details of measure, shape, and calculations.

To achieve this, follow the given steps:

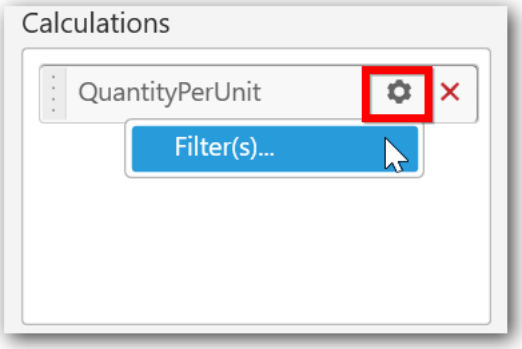
Drag and drop the elements from dimensions section to Calculations.



Here is an illustration,



You have settings options similar to Column(s).

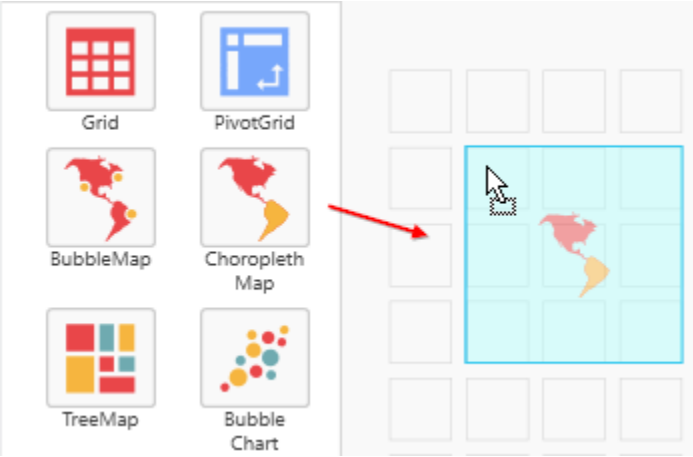


How to configure SSAS data to Choropleth map?

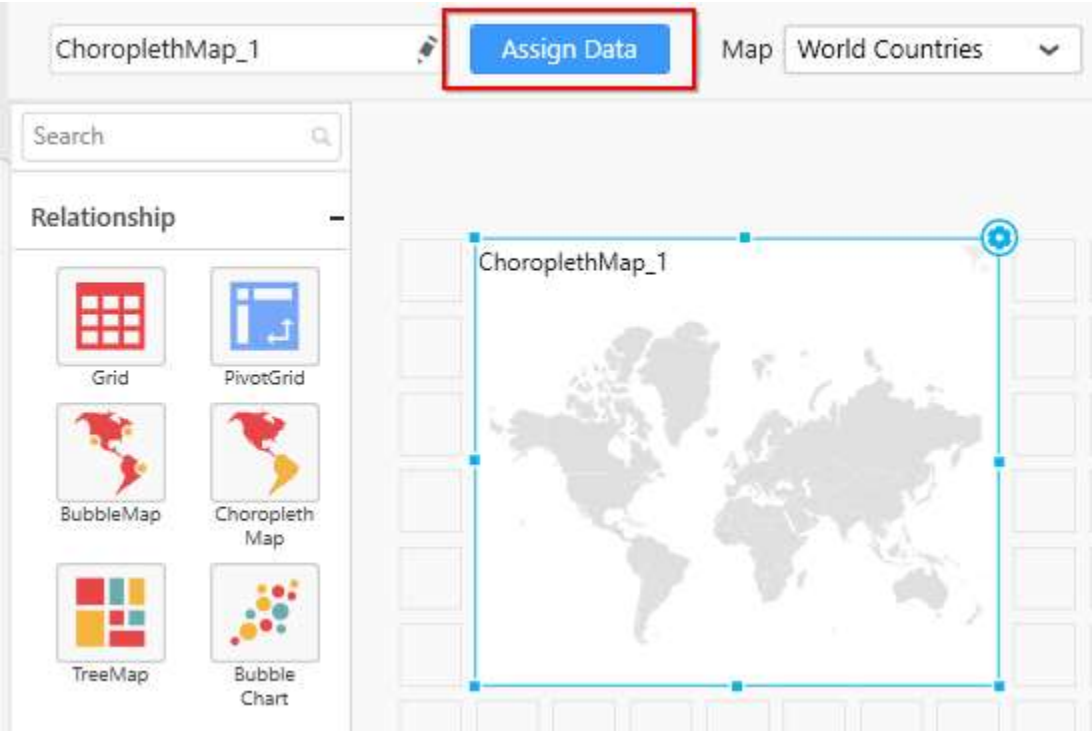
To plot a Choropleth Map, a minimum requirement of 1 value and 1 shape is needed. This holds one additional block Calculations in Data Pane. Dropping a dimension will display each region split by each of its item.

Following steps illustrates configuration of SSAS data to Choropleth Map.

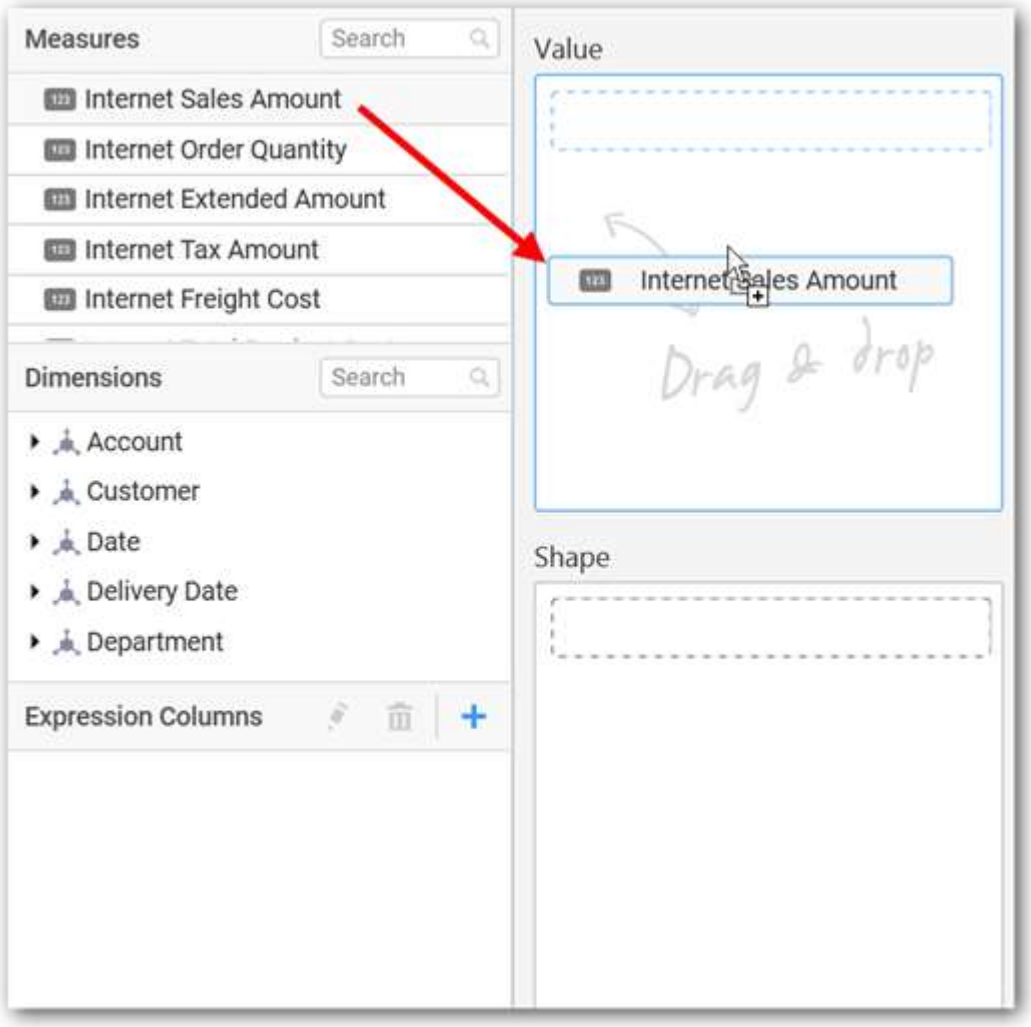
Drag and drop Choropleth Map control icon from the Tool box into design panel. You can find control in Toolbox by search.



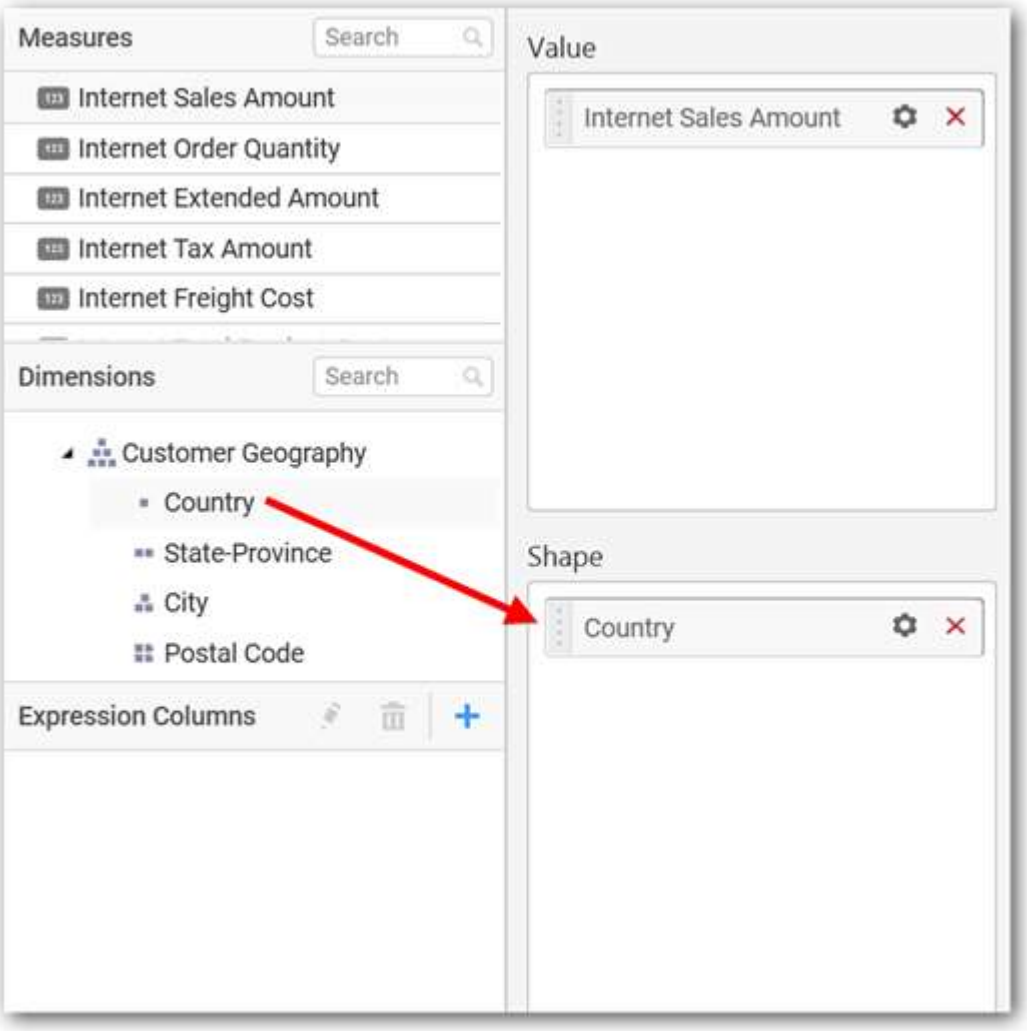
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



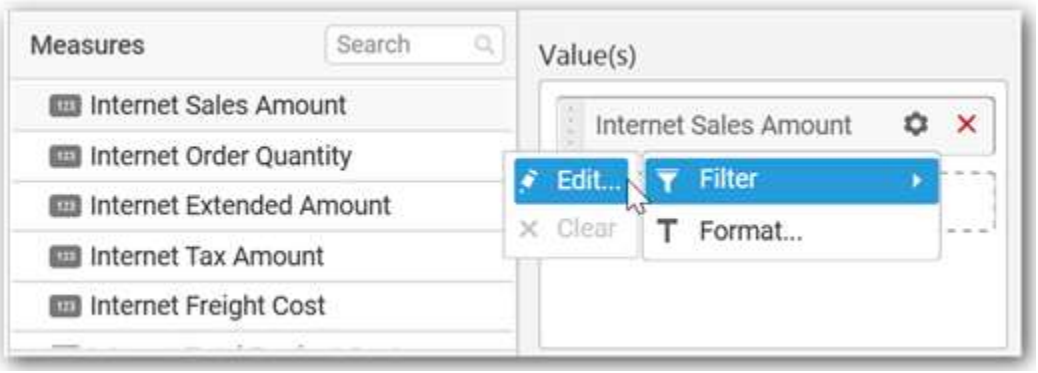
Drag and drop a column under Measures category into Value.



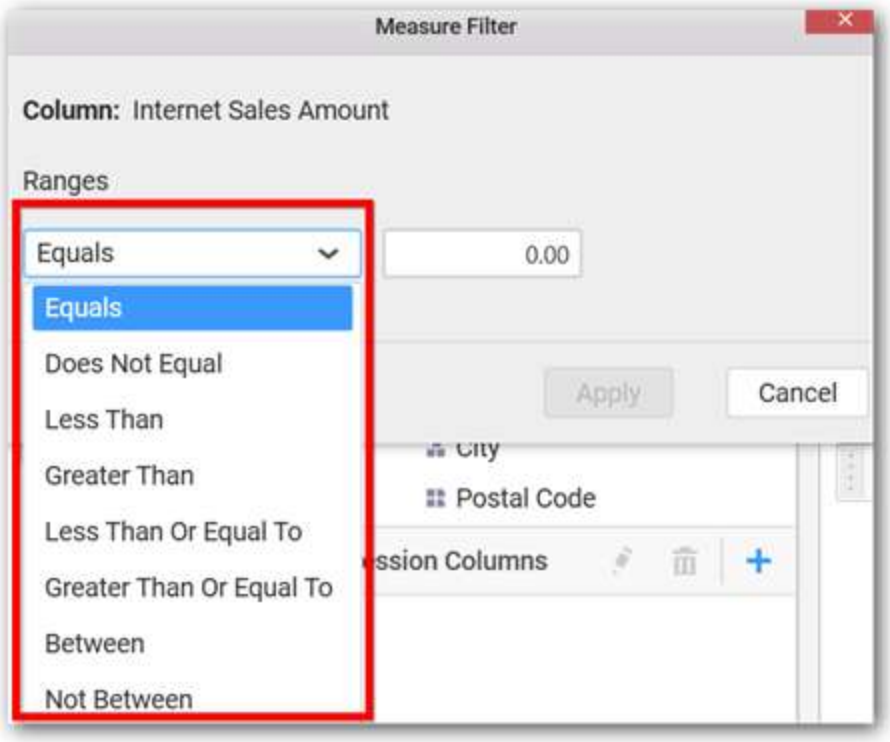
Drag and drop a dimension level or hierarchy column under Dimensions category into Shape.



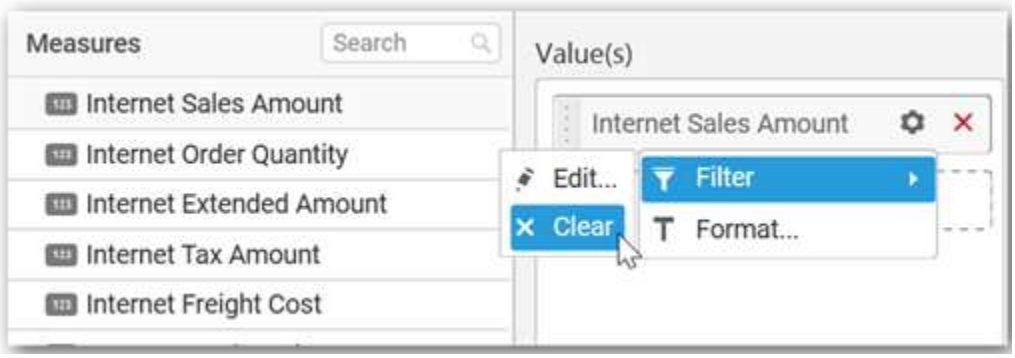
Define the filter criteria to match through choosing Edit option in Filter menu item.



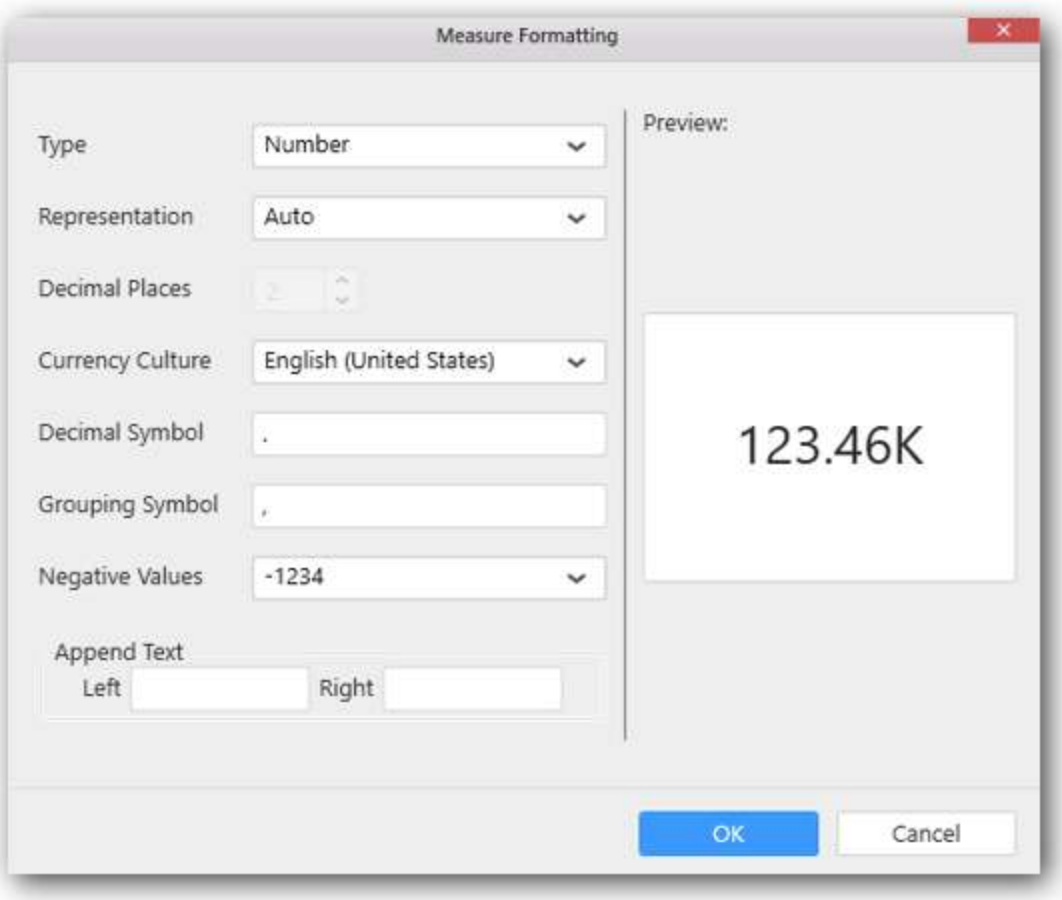
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



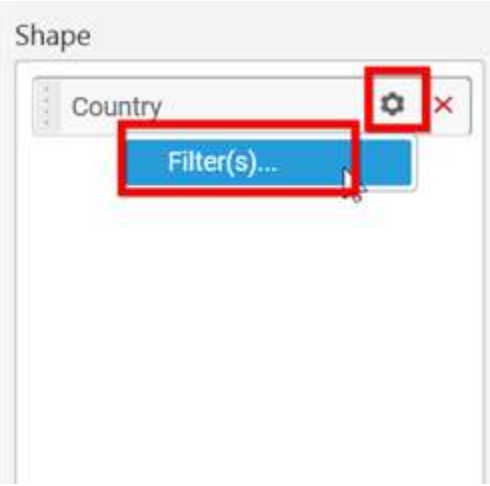
Select Clear option to clear the defined filter.



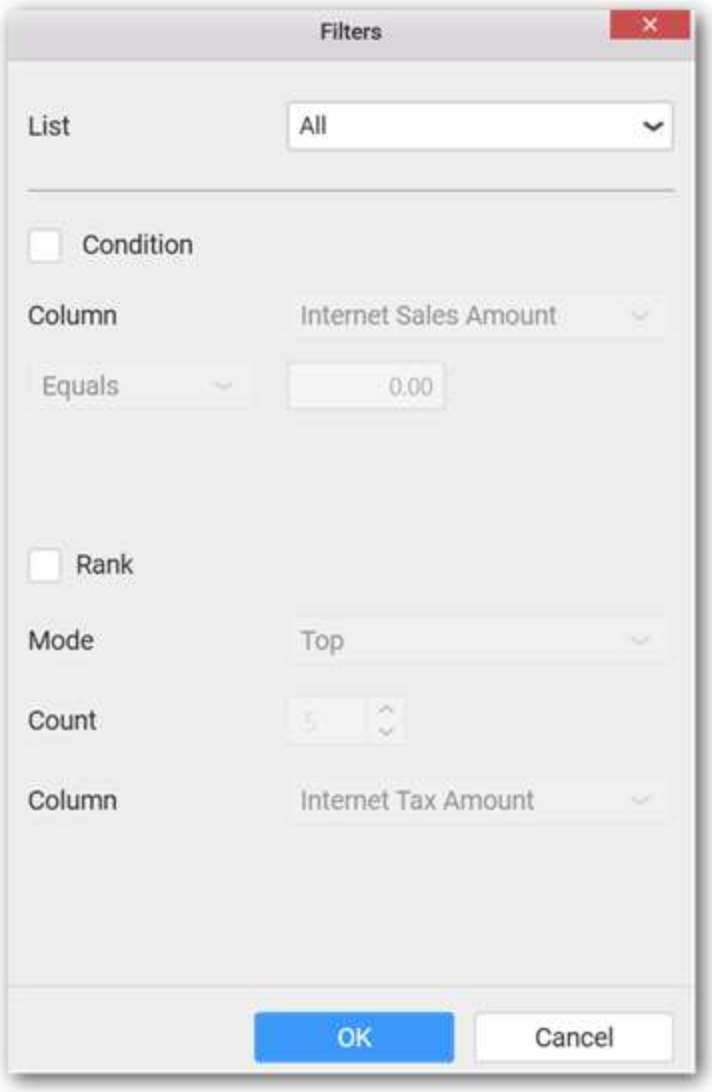
Select Format option to define the display format to the values in the column through Measure Formatting window.



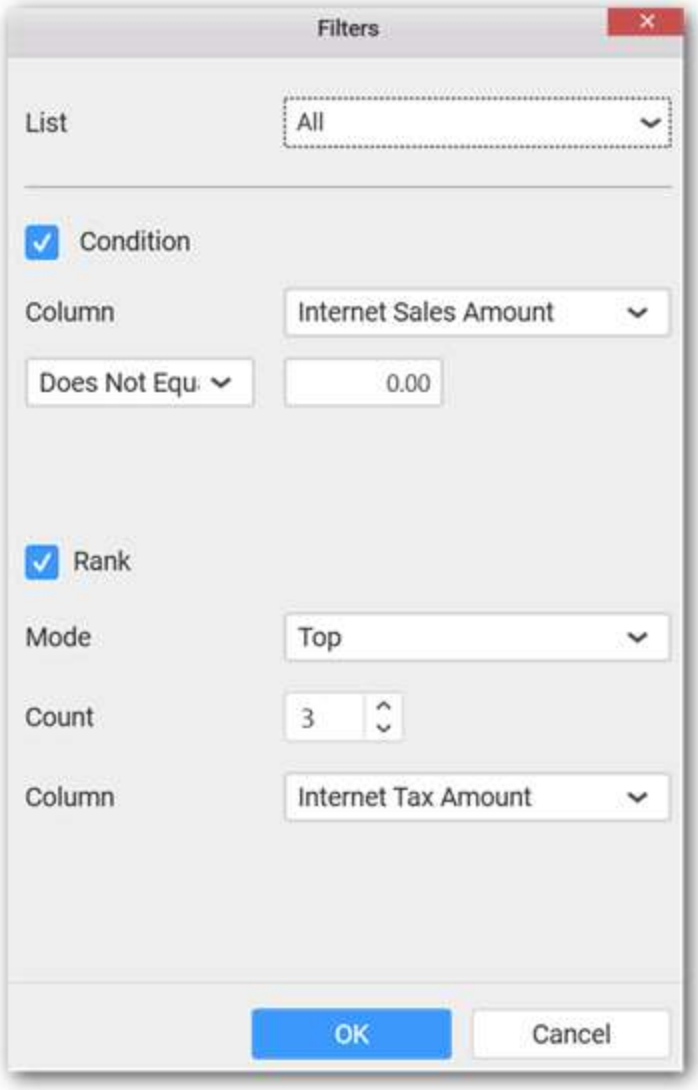
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



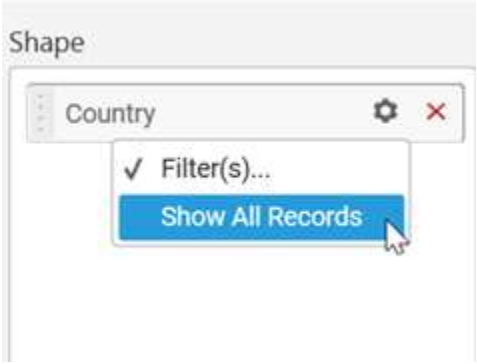
Select Filter(s)... to launch the Filters window.



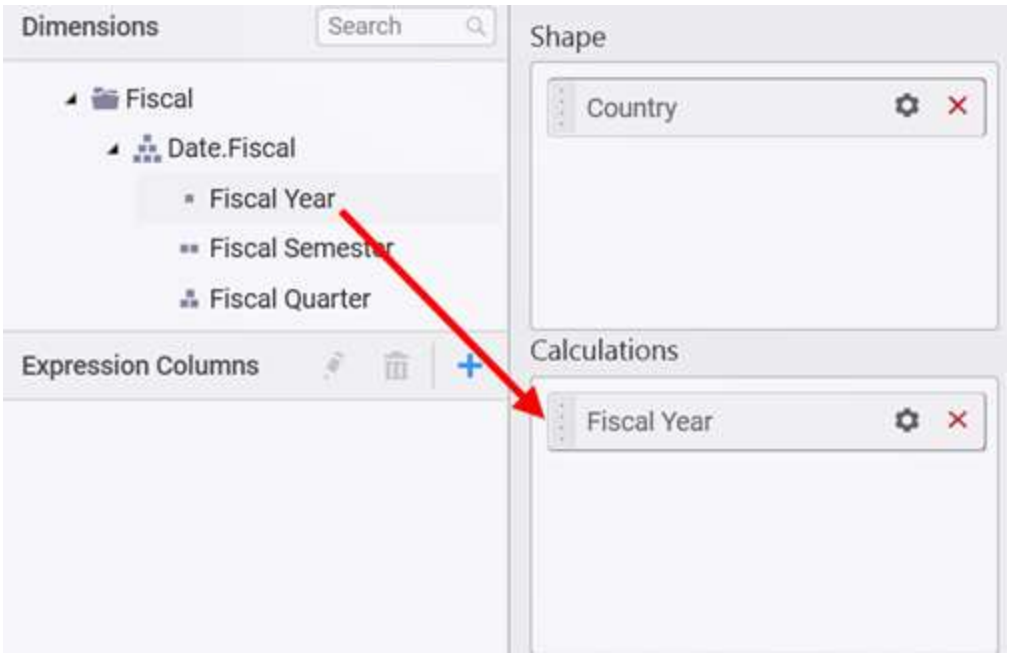
Define the filter Condition and Rank and Click OK.



To show all records again click on Show All Records.



Drag and drop a dimension level or hierarchy column under Dimensions category into Calculations.

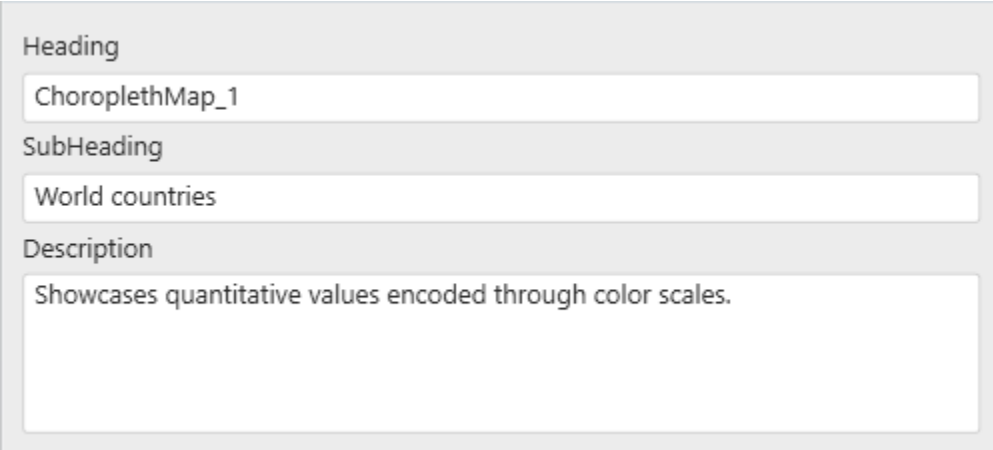


Here is an illustration,

How to format Choropleth Map?

You can format the Choropleth Map for better illustration of the view that you require, through the settings available in Properties pane.

General Settings



Heading

This allows you to set title for this Choropleth Map widget.

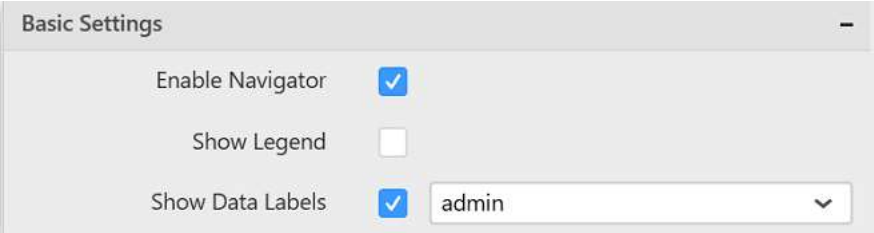
SubHeading

This allows you to set sub-title for this Choropleth Map widget.

Description

This allows you to set description for this Choropleth Map widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings



Enable Navigator

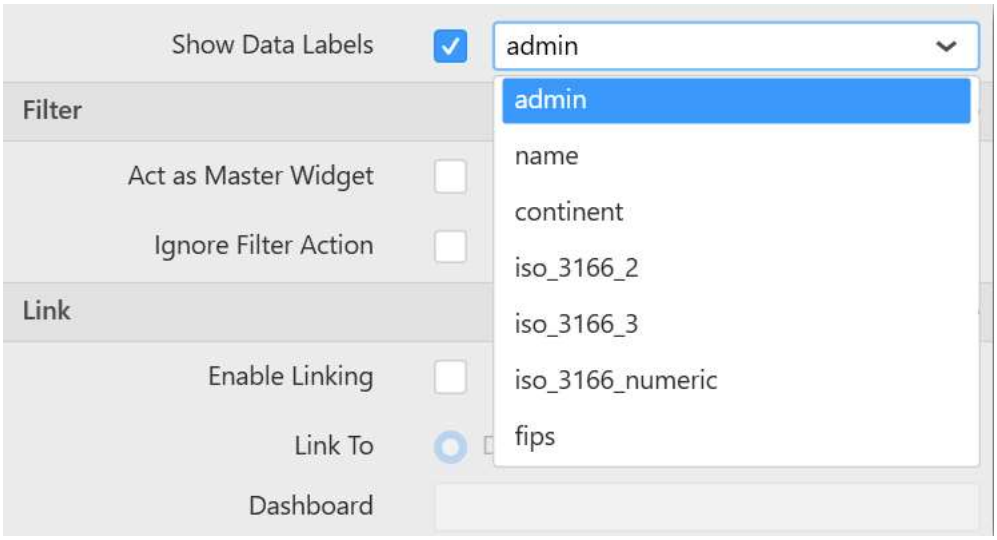
You can enable/ disable the map navigator in Choropleth Map.

Show Legend

You can toggle the visibility of legend in Choropleth Map.

Show Data Labels

You can enable or disable the data labels in map widget by using Show Data Labels property. After enabling this property, a dropdown menu is enabled for choosing the shape properties (i.e. name, admin, iso_3166, etc.) which is going to be displayed as data label.



The properties displayed in the DropDownList are changed based on selected shapes.

Filter Settings

Act as Master Widget

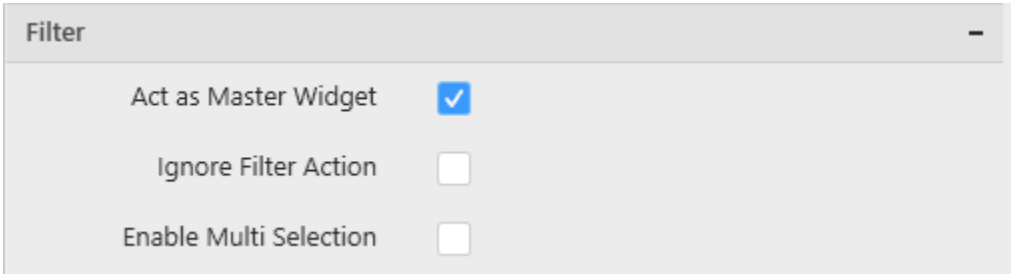
This allows you to define this Choropleth Map widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this Choropleth Map widget to ignore responding to the filter actions applied on other widgets in dashboard.

Enable Multi Selection

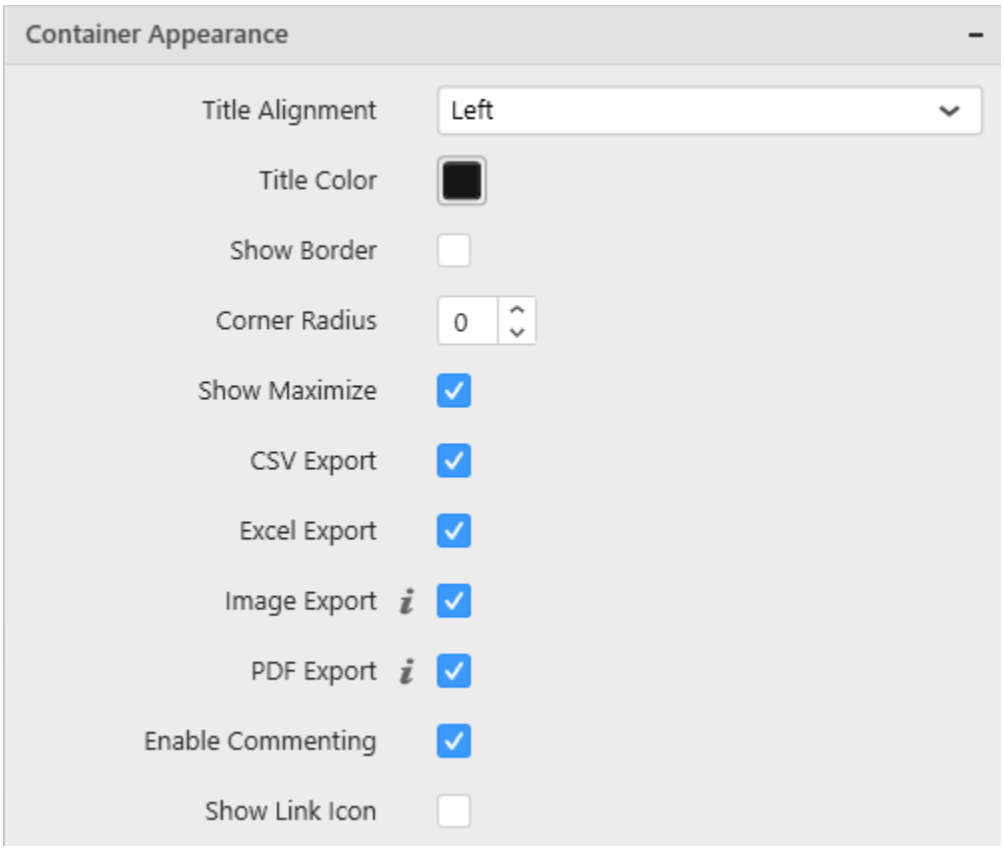
Enable Multi Selection is displayed only when you enable the Act as Master Widget option. This allows you to select multiple shapes in the Choropleth Map widget.



Link Settings

You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this Choropleth Map widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the Choropleth Map widget.

CSV Export

This allows you to enable/disable the CSV export option for this Choropleth Map widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this Choropleth Map widget. Enabling this allows you to export the summarized data of the widget view to XLSX format.

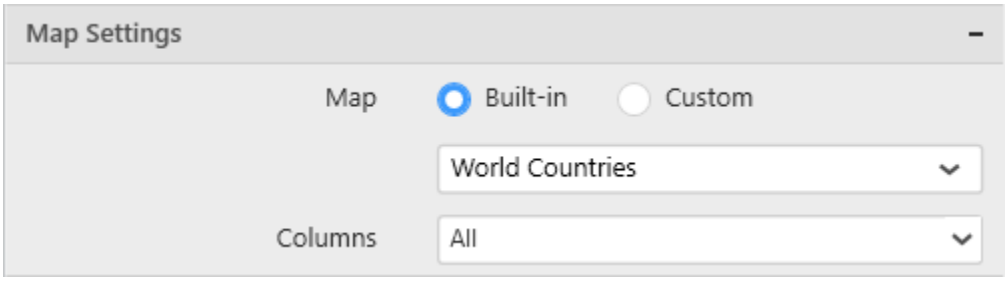
Image Export

This allows you to enable/disable the image export option for this Choropleth Map widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Map Settings

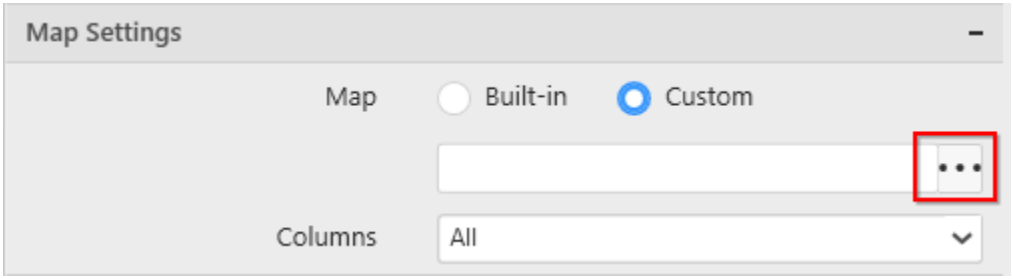


Built-in

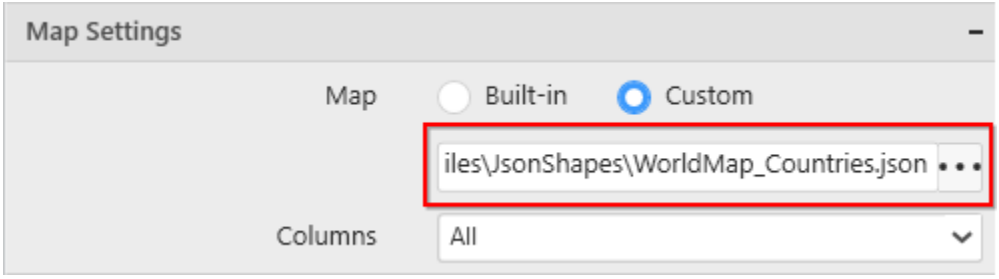
You can choose from the available 14 map shape files shipped along with dashboard designer for visualizing map data.

Custom

This option allows you to load any map shape files (i.e. .json files) into Map widget. For loading custom map files, you need to choose Custom option from Map settings. After enabling this option, browse icon is enabled for loading custom shape file.

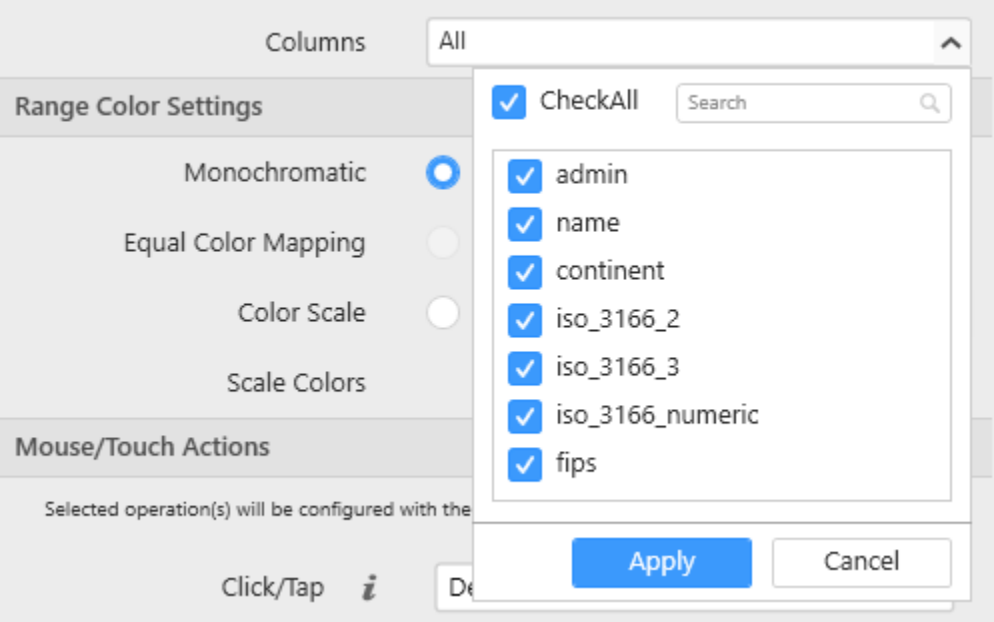


Using browse icon, we can browse and load the required shape file in Map.

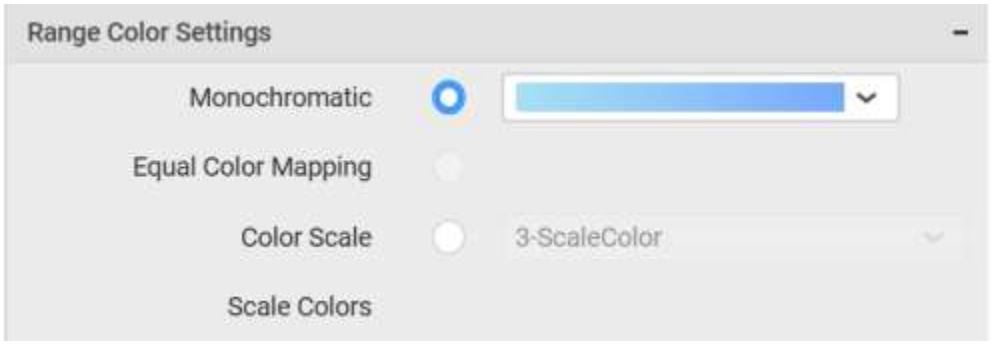


Columns

You can choose multiple column fields from the list of field names supported by the loaded map shape that can be mapped with the data source connected. By default all column fields have been selected and you can select/unselect from columns drop down menu.

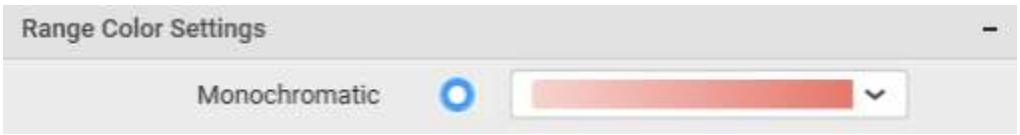


Range Color Settings

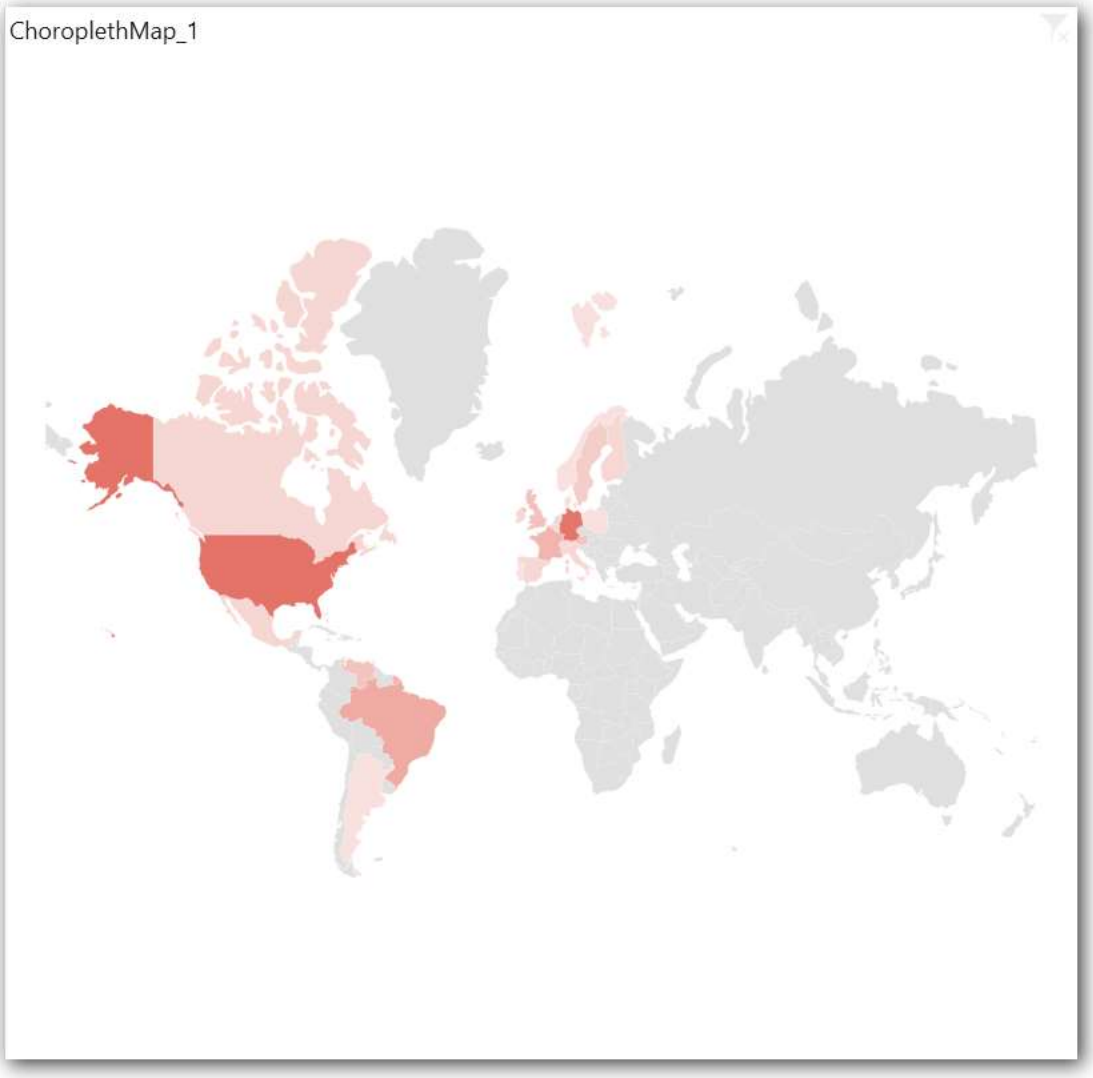


Monochromatic

You can configure a single color palette whose saturation differs based on the value density.



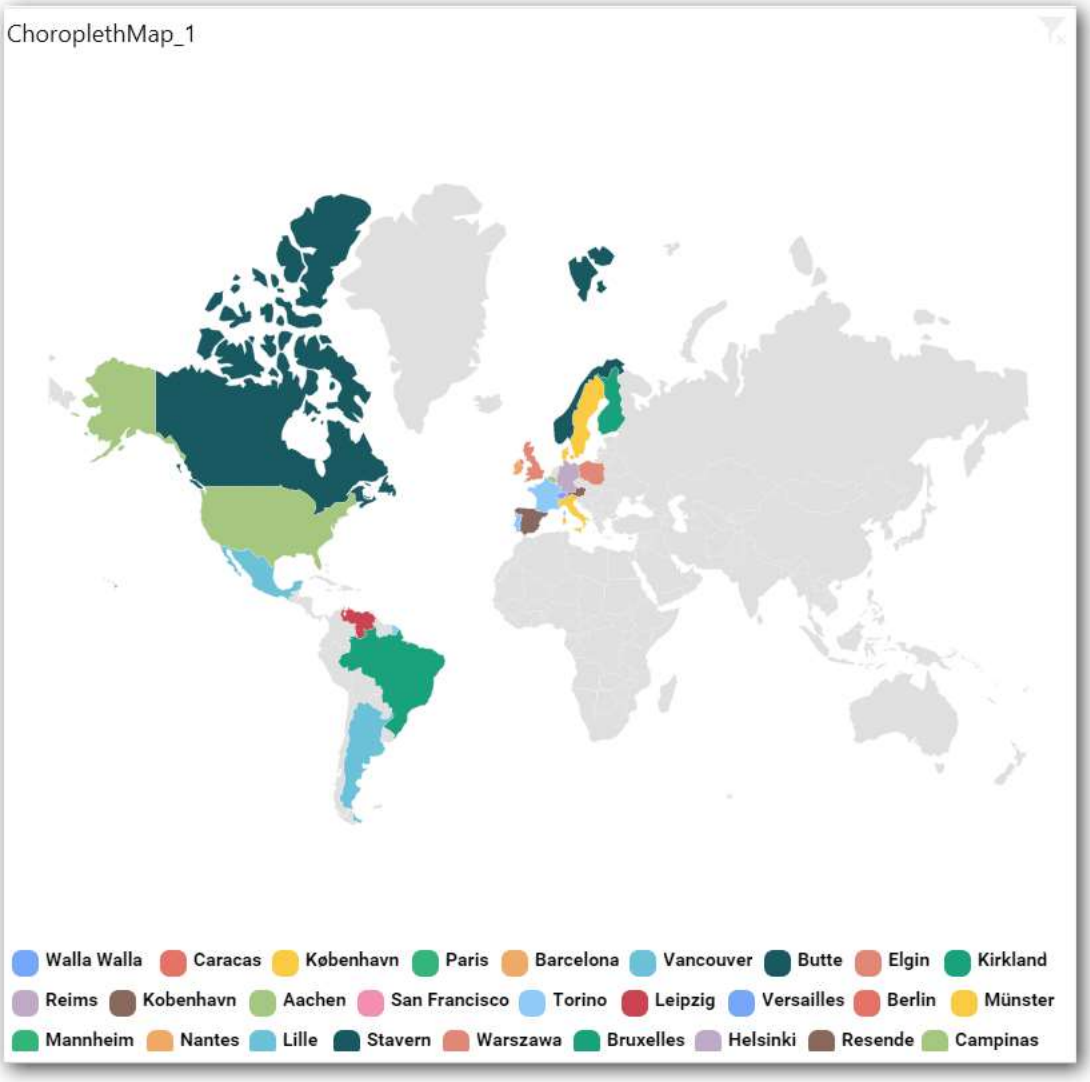
Here is an illustration,



Equal Color Mapping

This option will be available only when you add any element to Calculations block in Data pane. Switching to any other range color scheme option, will remove the added column from Calculations block.

Here is an illustration,



Color Scale

You can choose the number of scales the color need to be split into (3, 5, 7, or 9). Based on the selected color scale, the color palette will be displayed.

Scale Colors

You can customize the colors in the selected color scale.

Pivot Grid

Pivot Grid allows you to display summarized data in cross tabular format.

Year wise sales per country

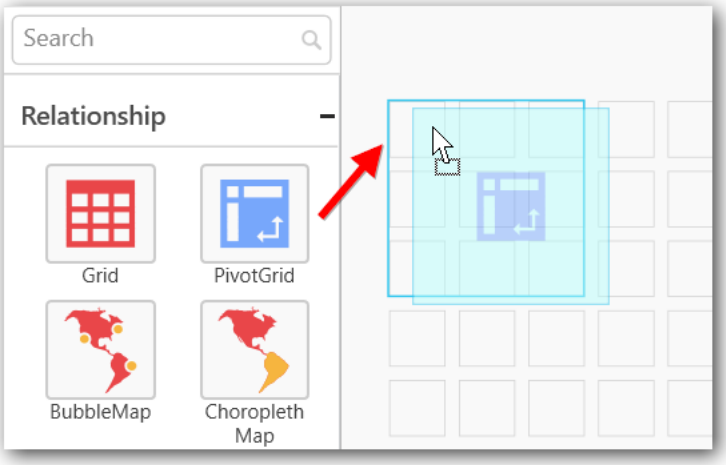
	1996	1997	1998	Grand Total
Austria	29,314.49	63,094.48	46,972.73	139,381.7
Brazil	23,814.52	44,524.42	46,551.34	114,890.28
France	17,627.33	47,879.38	19,950.65	85,457.36
Germany	37,780.63	124,101.83	82,614.62	244,497.07
UK	9,650.2	27,825.01	23,124.85	60,600.06
USA	41,869.78	120,975.78	100,541.6	263,387.16
Venezuela	10,424.76	28,153.57	22,196.52	60,774.85
Grand Total	170,481.71	456,554.46	341,952.3	968,988.47

How to configure flat table data to Pivot Grid?

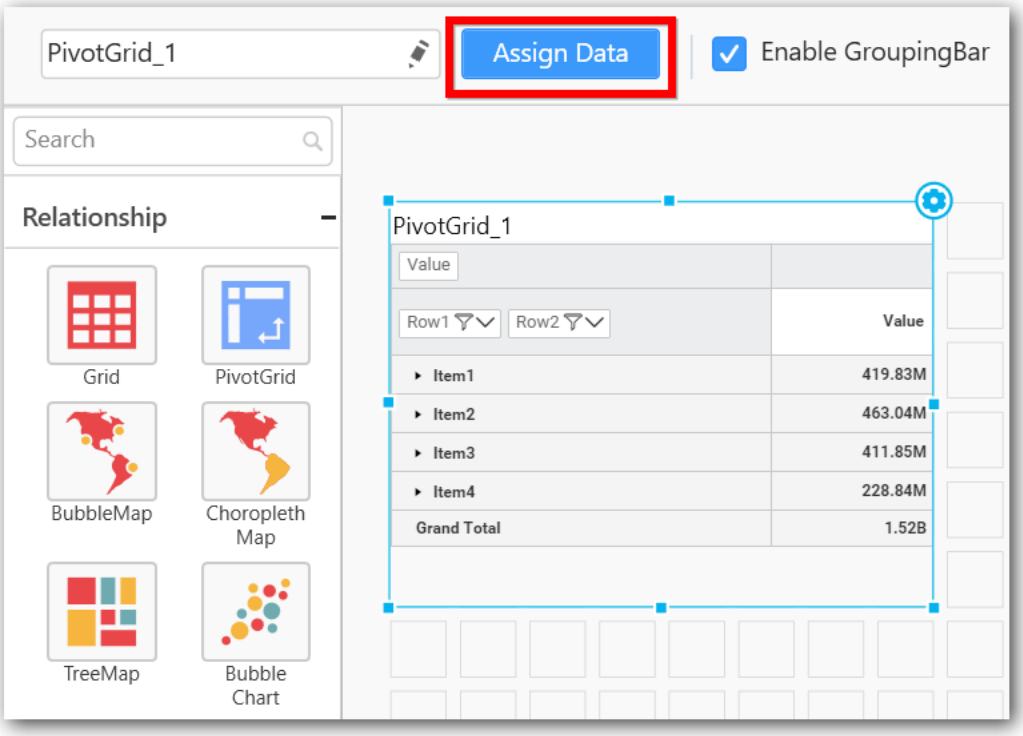
To construct a pivot grid, a minimum requirement of 1 value and 1 row/column is needed.

The following procedure illustrates data configuration of Pivot Grid.

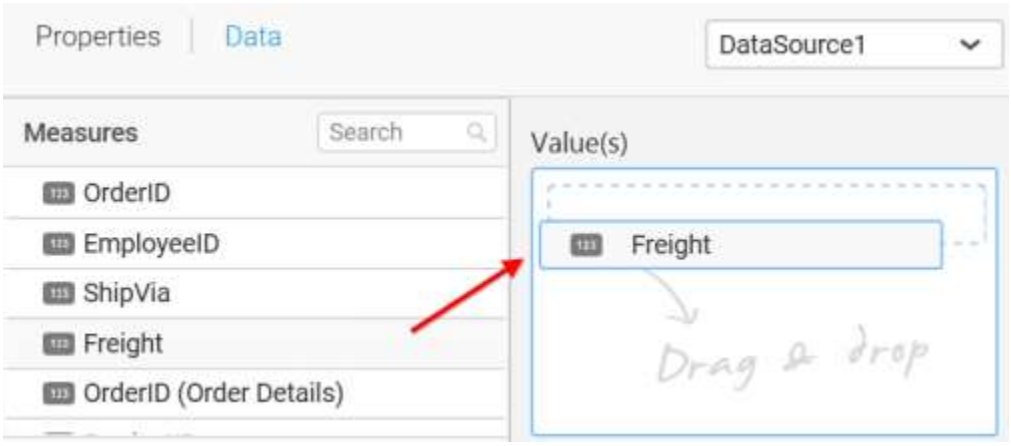
Drag and drop Pivot Grid control icon from the Tool box into design panel. You can find control in Toolbox by search.



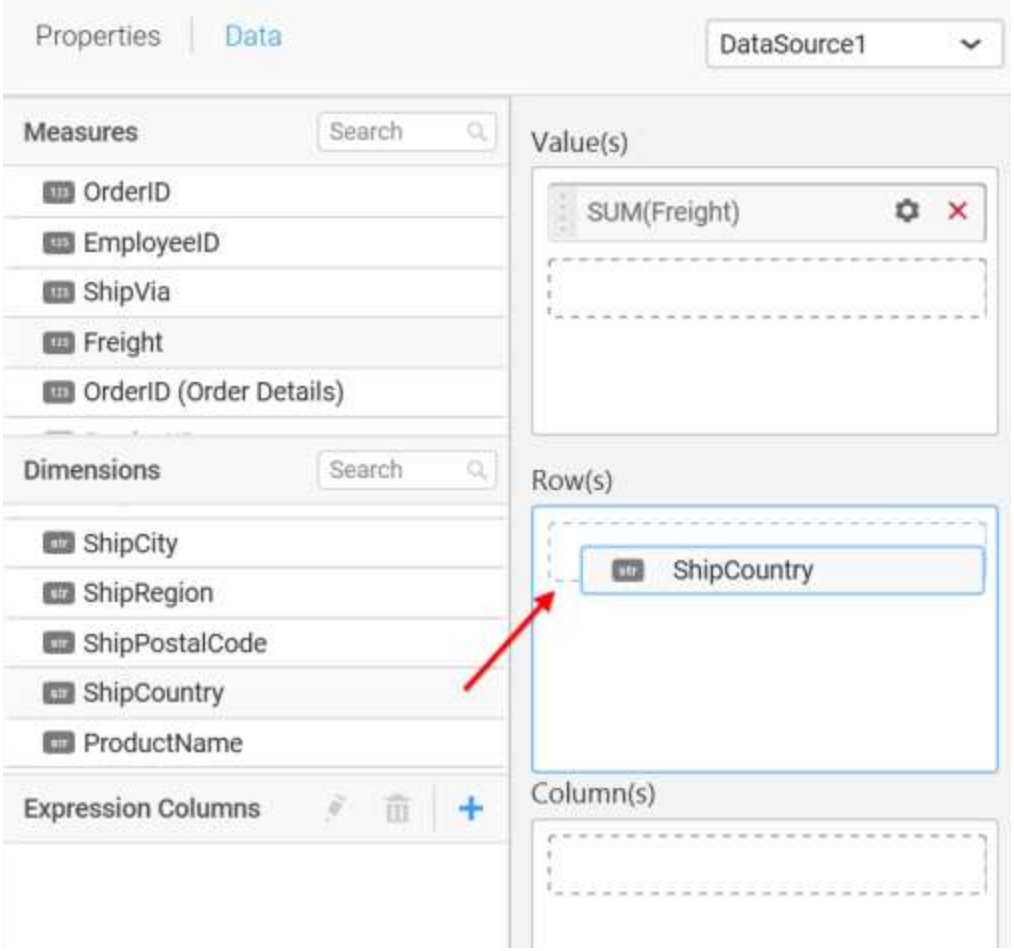
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



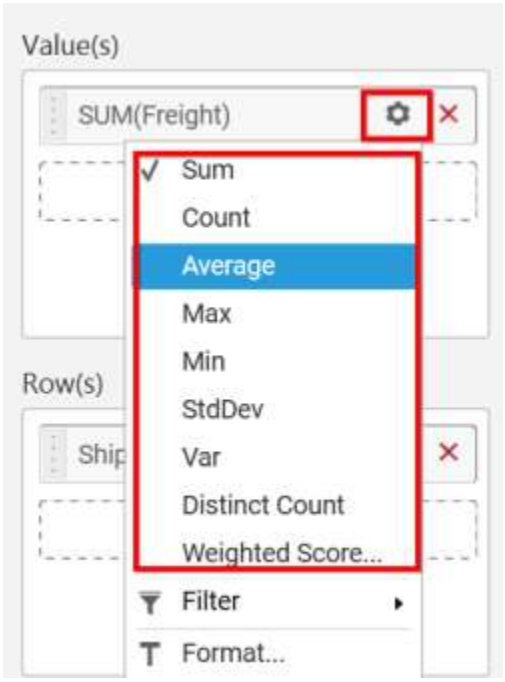
Drag and Drop the elements from Measures to Value(s).



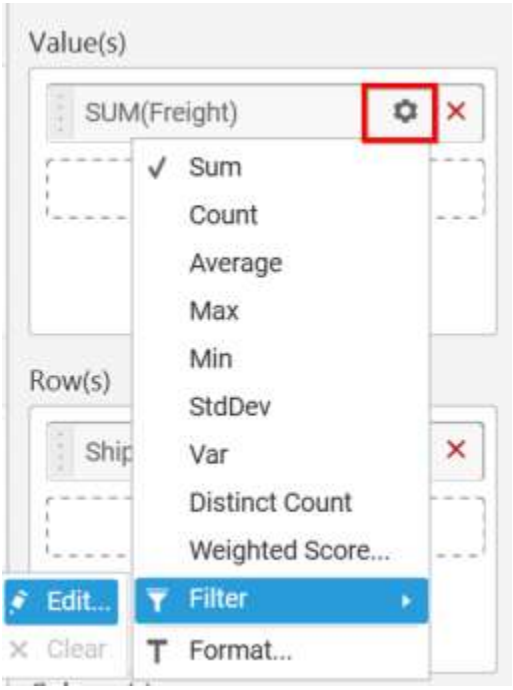
Drag and Drop the elements from sections to Row(s).



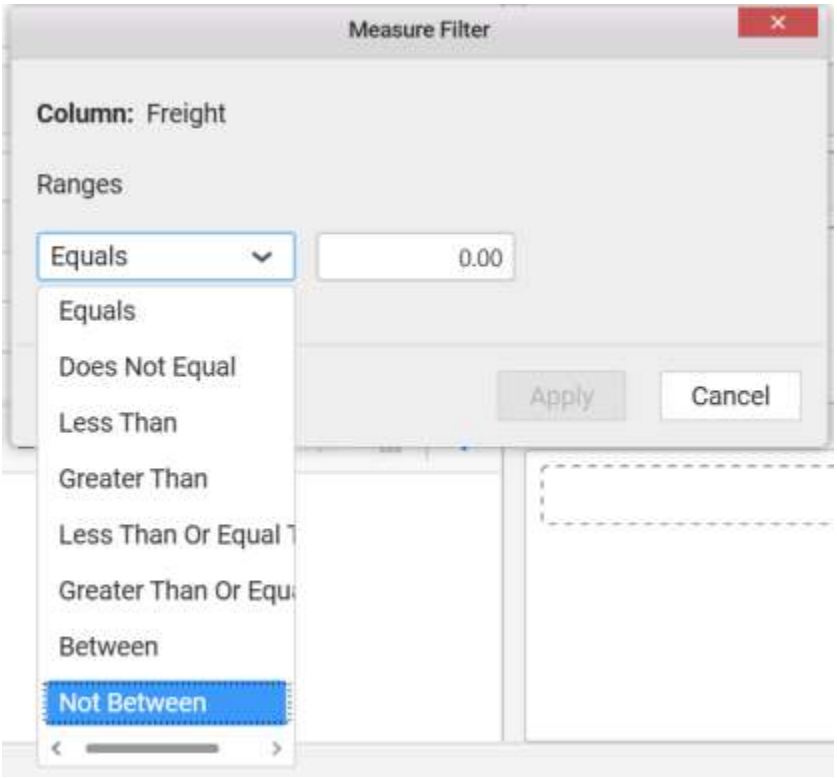
You can use the aggregation function to change the Value(s) of the column.



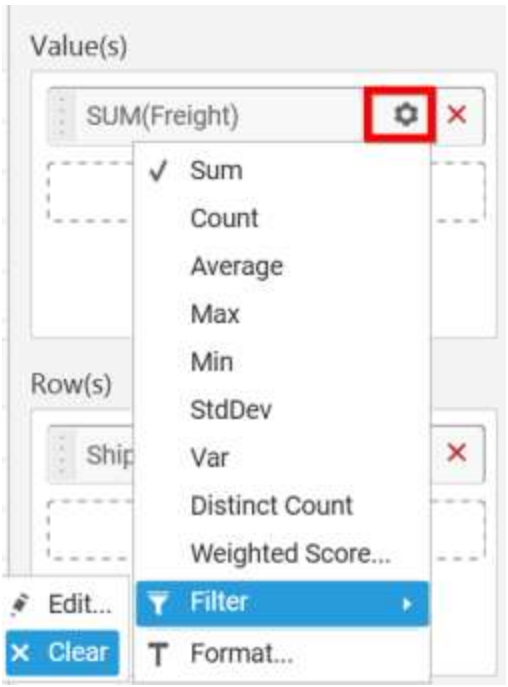
You can use the Filter option to enable filters by selecting the Edit option.



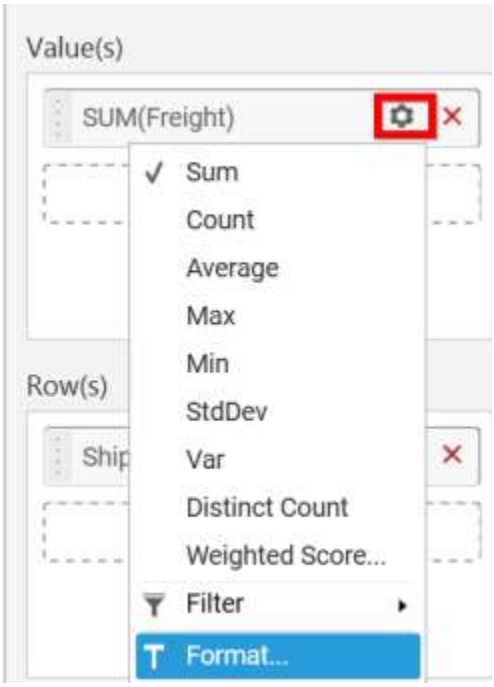
Measure Filter window will be shown to set the Ranges.



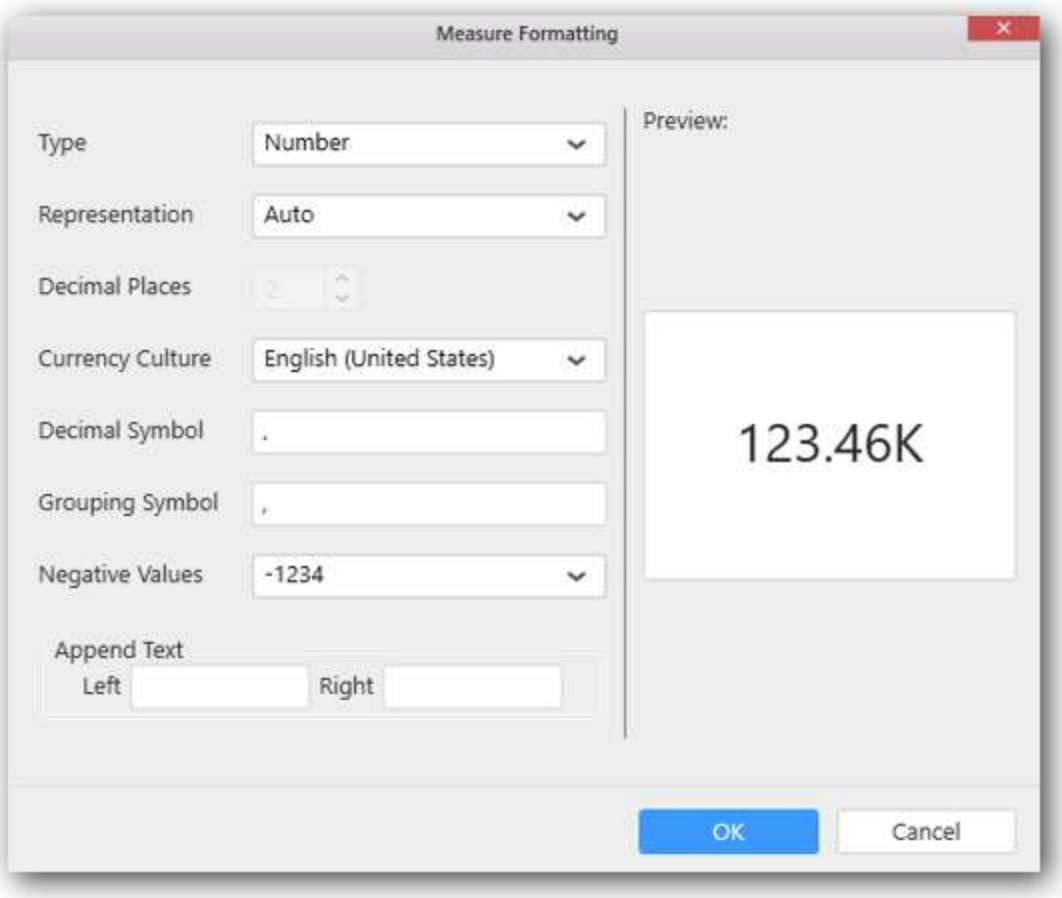
You can clear the filters by selecting the Clear option.



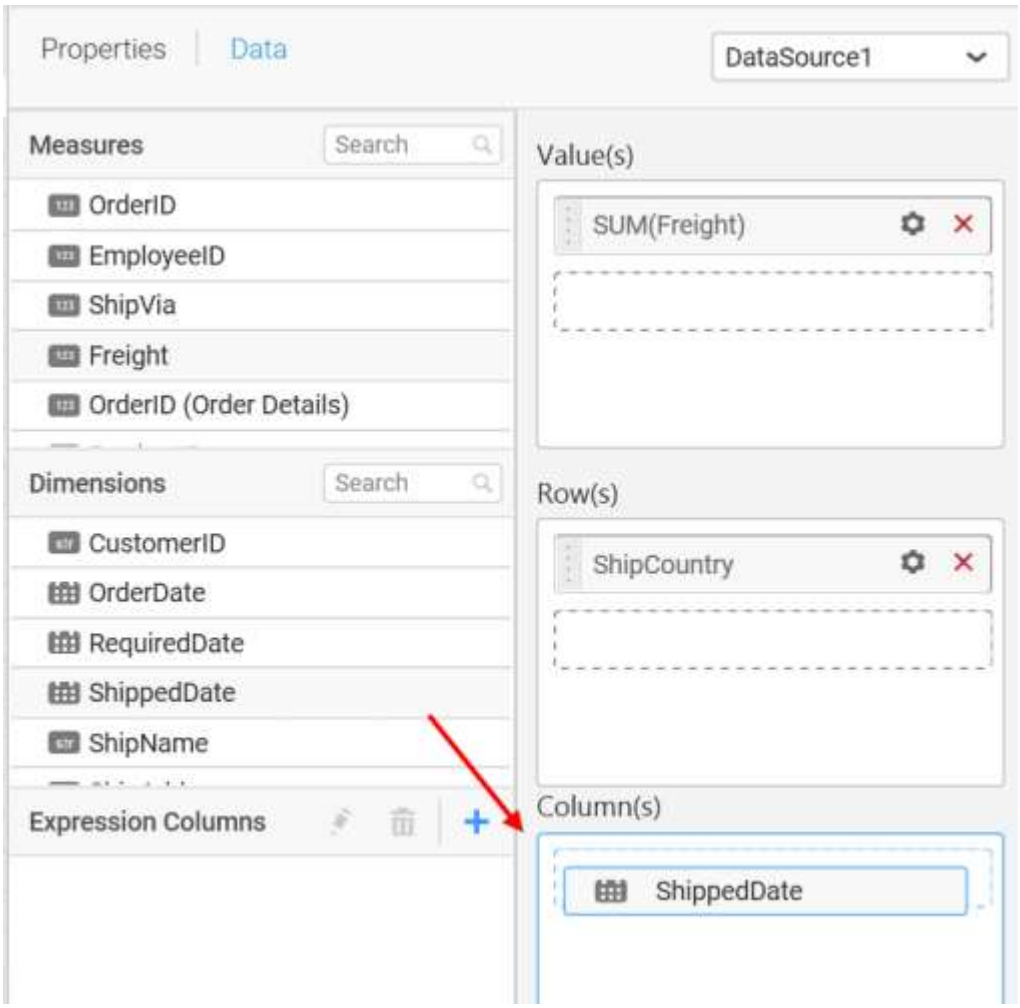
You can format the values by selecting the Format option.



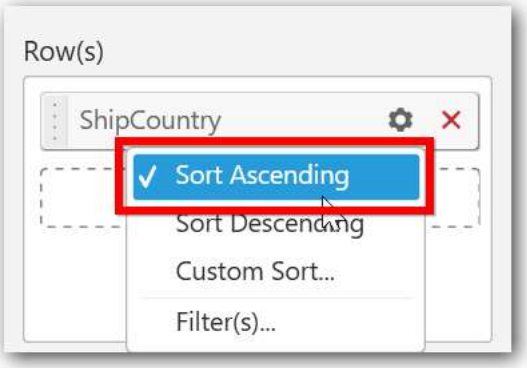
Measure Formatting window will be shown.



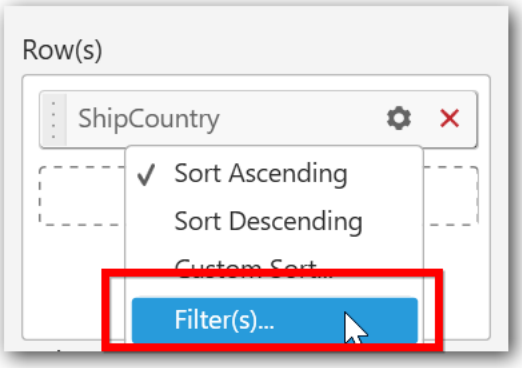
Drag and Drop the elements from sections to Column(s).



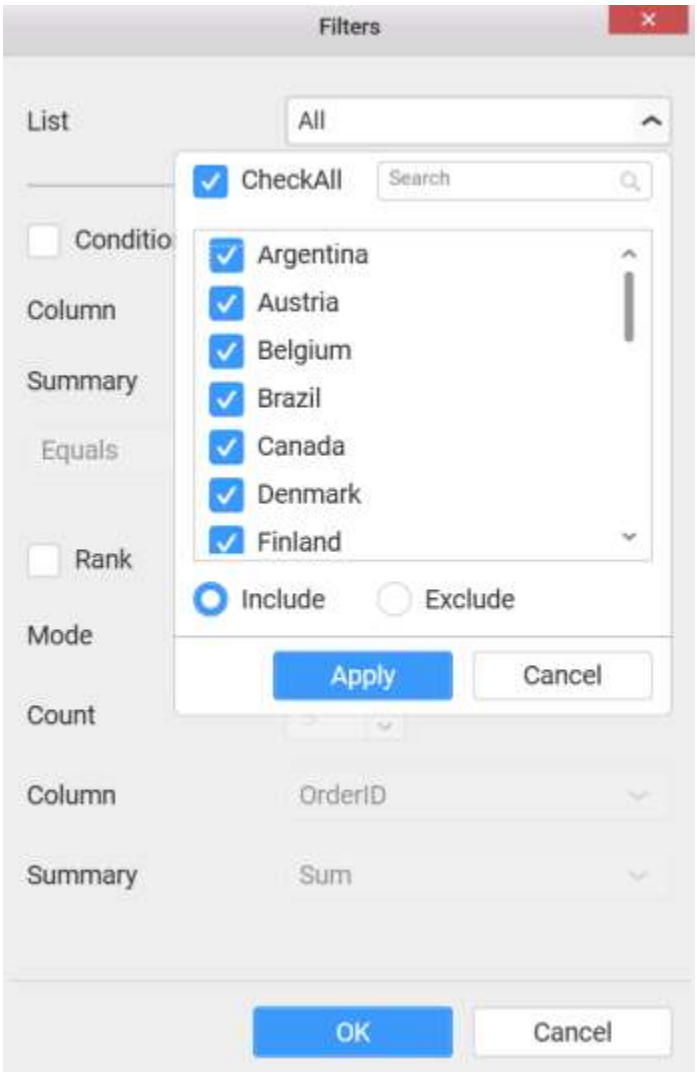
You can select the settings to sort the data either Ascending or Descending



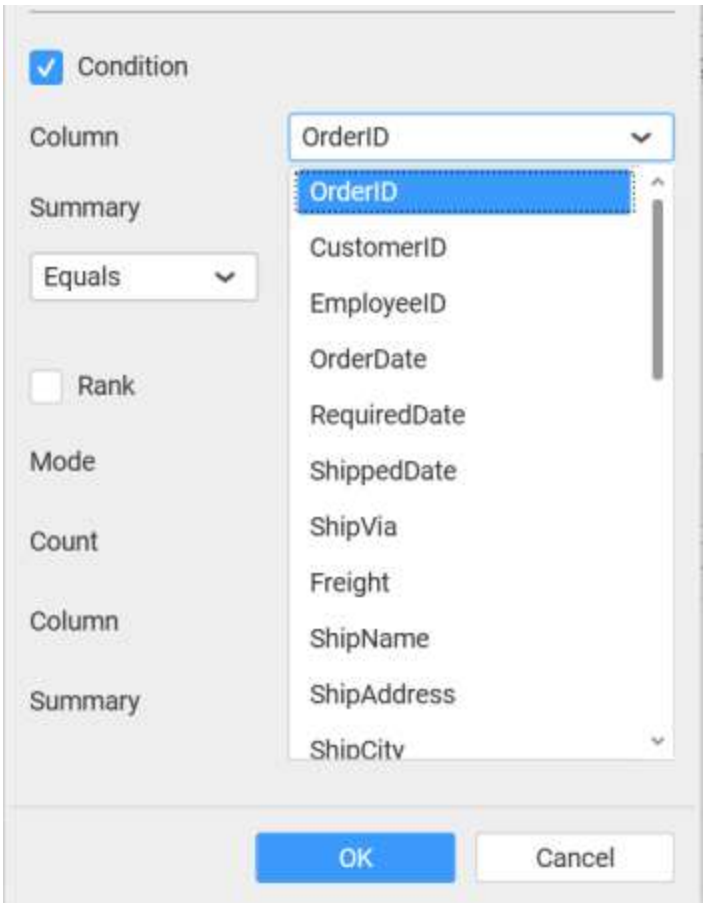
You can use the filters by selecting the Filter(s)... option to rank to the elements.

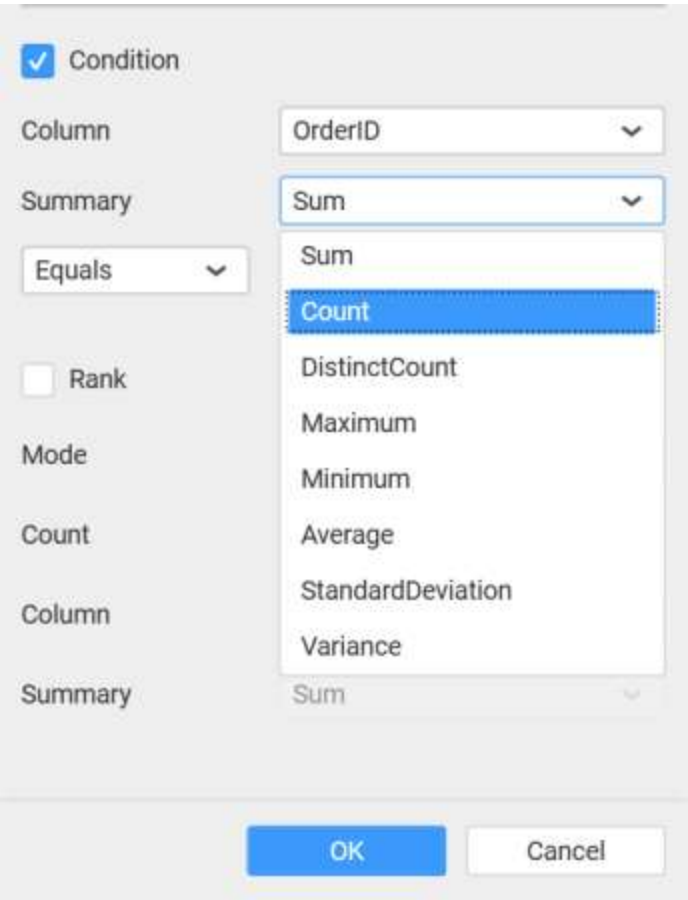


You can select the specific country to filter the element and CheckAll is used either to check all the data or to select the specific data. Include and Exclude is used to include and exclude the elements by selecting the radio button and click the Apply button.

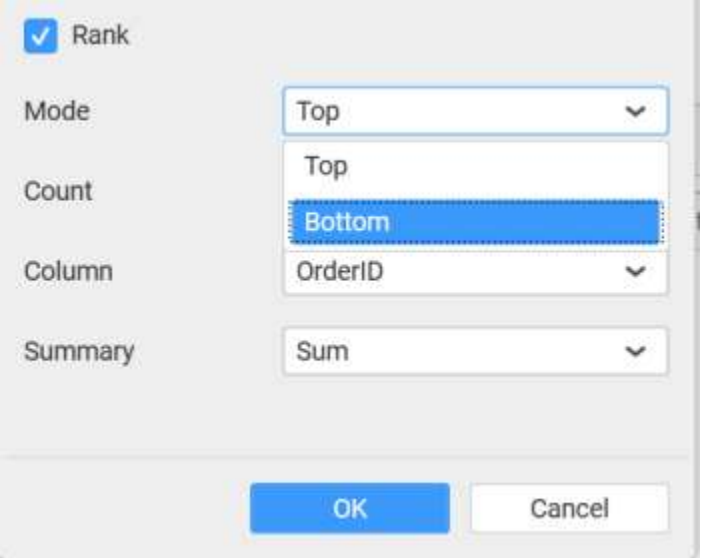


You can select the Condition option to change the Column elements and Summary type by selecting the required column name and summary type.

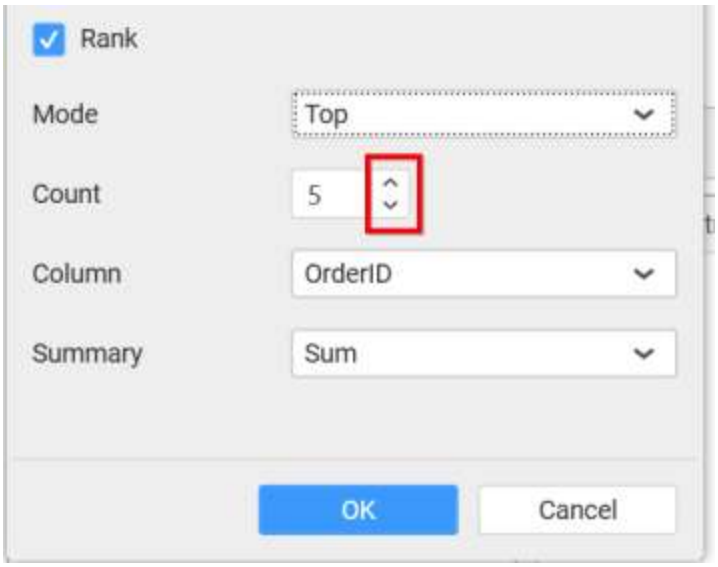




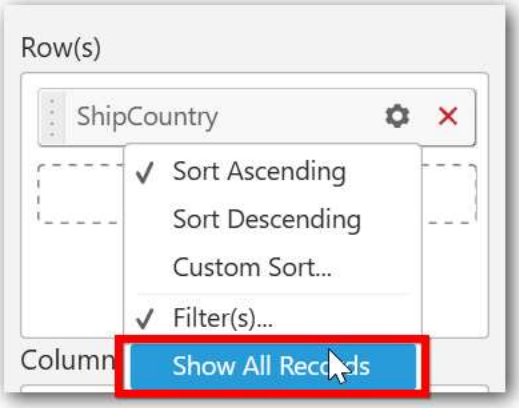
You can select the Rank option to enable filters and select the Mode either top or bottom.



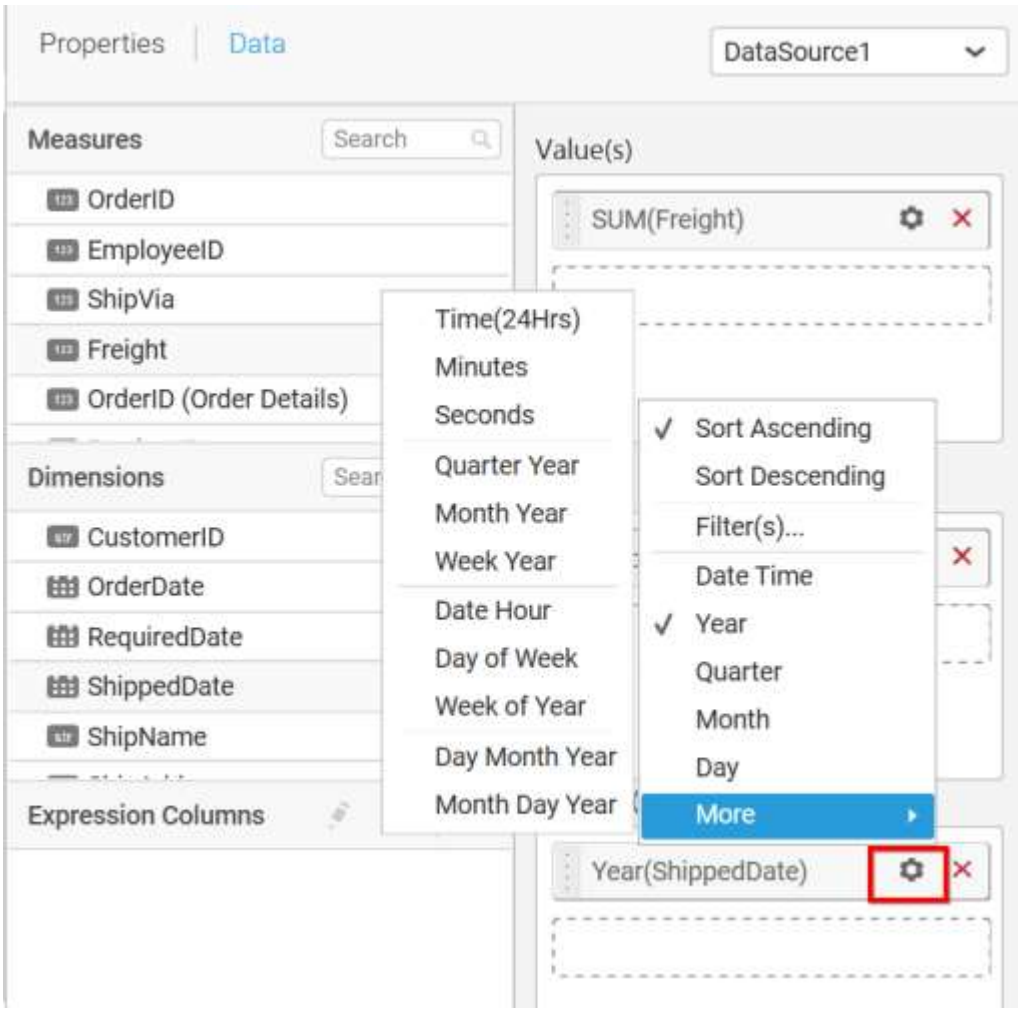
You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.



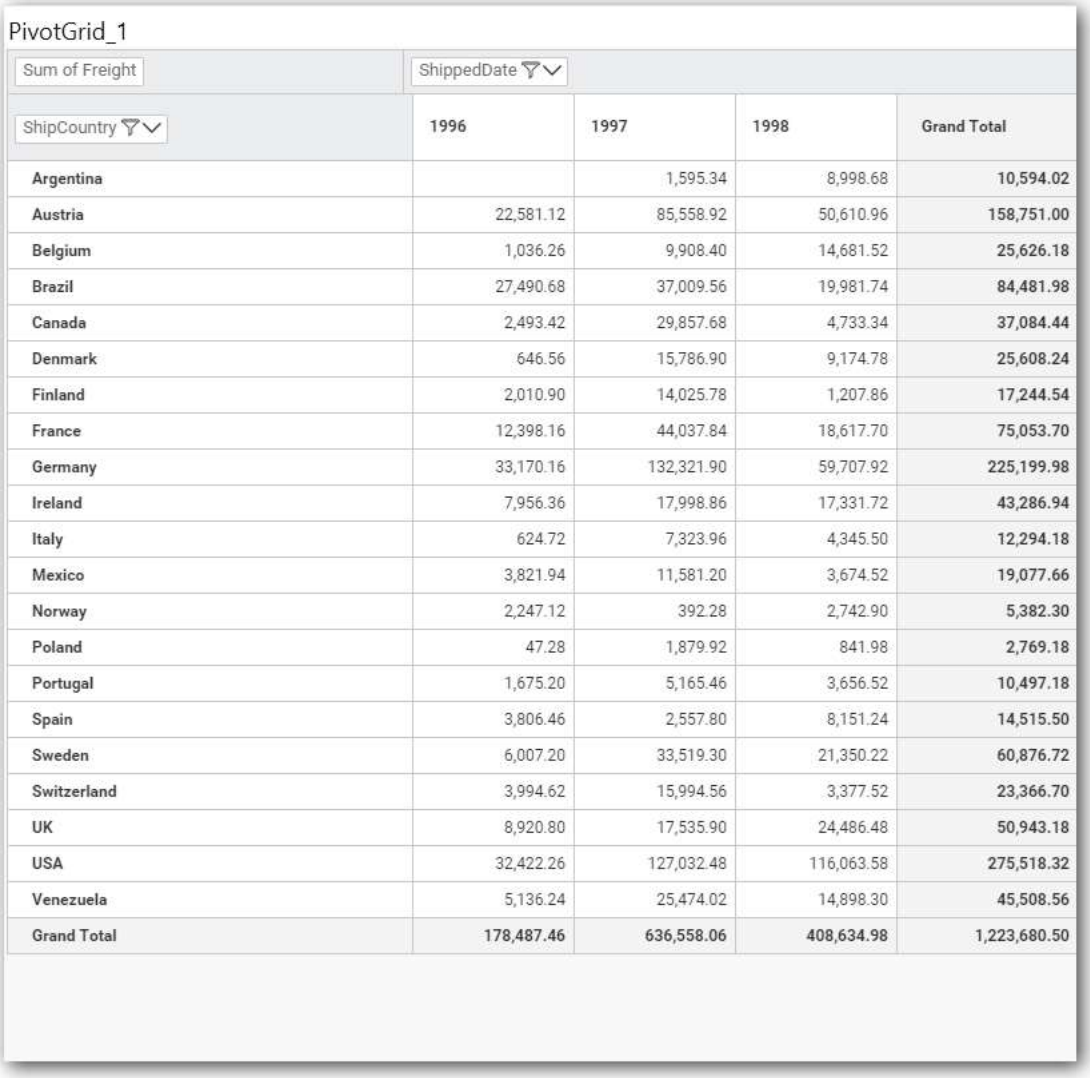
You can clear the filters by selecting the Show All Records.



You can sort the values by selecting either Ascending or Descending. Filter(s)... option can be used to edit the filter condition. More option is used to set the Time and Date in different format.



Here is an illustration,



PivotGrid_1

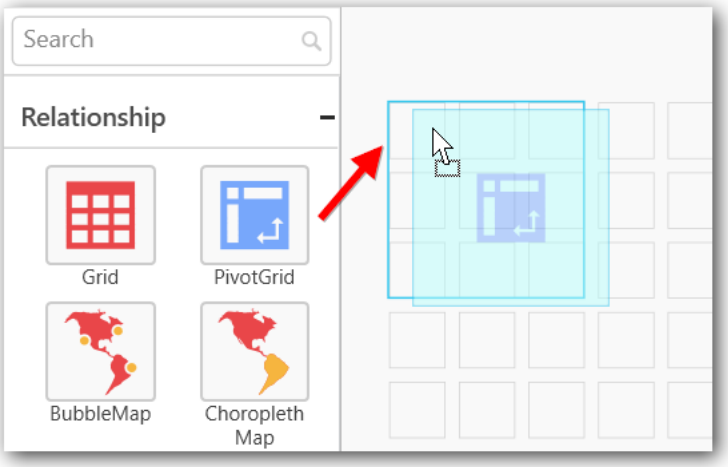
Sum of Freight	ShippedDate			
ShipCountry	1996	1997	1998	Grand Total
Argentina		1,595.34	8,998.68	10,594.02
Austria	22,581.12	85,558.92	50,610.96	158,751.00
Belgium	1,036.26	9,908.40	14,681.52	25,626.18
Brazil	27,490.68	37,009.56	19,981.74	84,481.98
Canada	2,493.42	29,857.68	4,733.34	37,084.44
Denmark	646.56	15,786.90	9,174.78	25,608.24
Finland	2,010.90	14,025.78	1,207.86	17,244.54
France	12,398.16	44,037.84	18,617.70	75,053.70
Germany	33,170.16	132,321.90	59,707.92	225,199.98
Ireland	7,956.36	17,998.86	17,331.72	43,286.94
Italy	624.72	7,323.96	4,345.50	12,294.18
Mexico	3,821.94	11,581.20	3,674.52	19,077.66
Norway	2,247.12	392.28	2,742.90	5,382.30
Poland	47.28	1,879.92	841.98	2,769.18
Portugal	1,675.20	5,165.46	3,656.52	10,497.18
Spain	3,806.46	2,557.80	8,151.24	14,515.50
Sweden	6,007.20	33,519.30	21,350.22	60,876.72
Switzerland	3,994.62	15,994.56	3,377.52	23,366.70
UK	8,920.80	17,535.90	24,486.48	50,943.18
USA	32,422.26	127,032.48	116,063.58	275,518.32
Venezuela	5,136.24	25,474.02	14,898.30	45,508.56
Grand Total	178,487.46	636,558.06	408,634.98	1,223,680.50

How to configure the SSAS data to Pivot Grid?

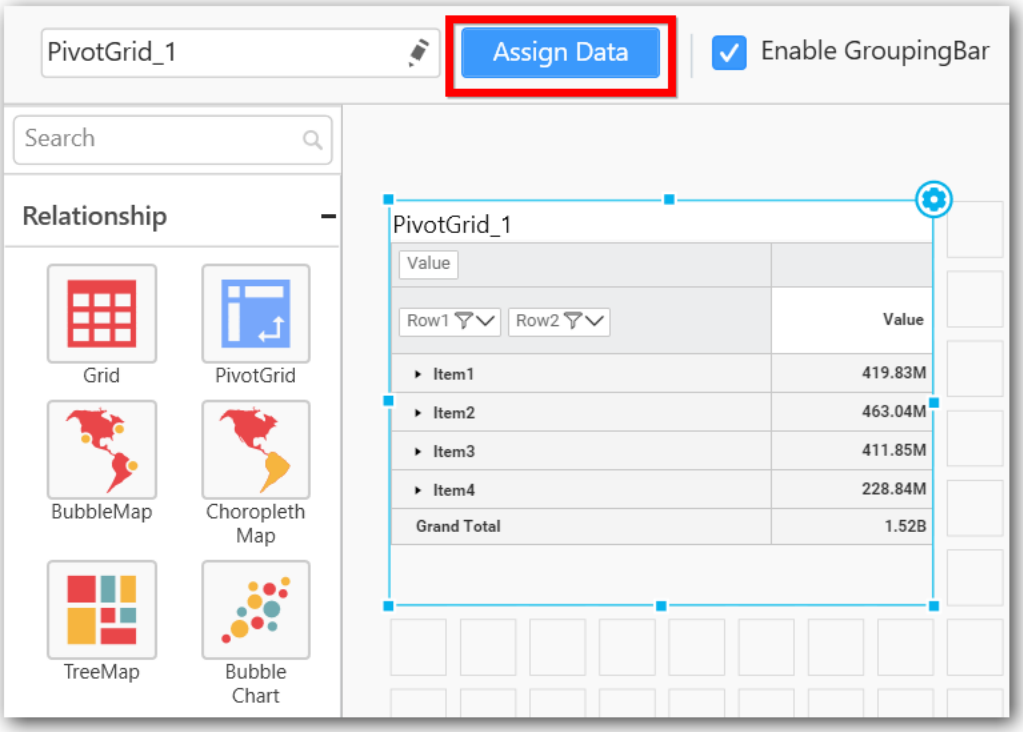
To construct a pivot grid, a minimum requirement of 1 value and 1 row/column is needed.

Following steps illustrates configuration of SSAS data to Pivot Grid.

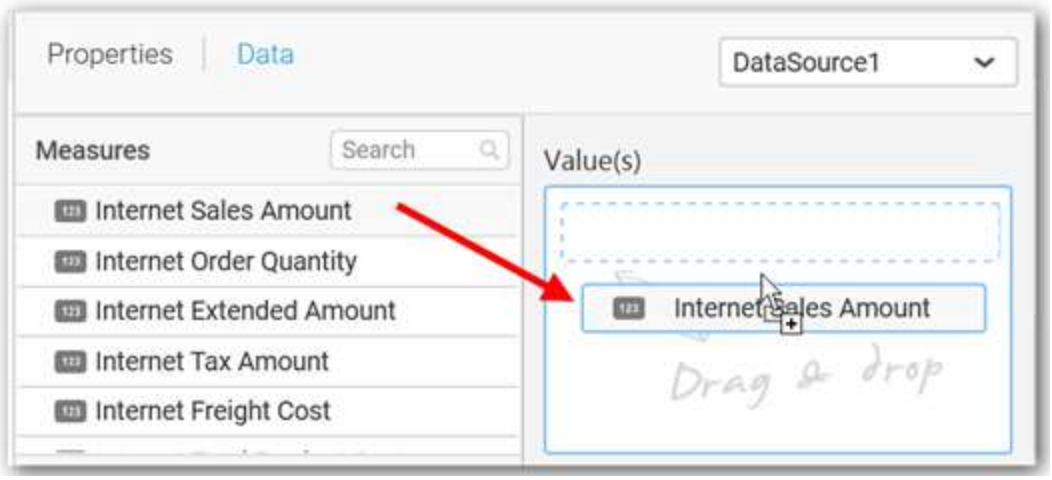
Drag and drop Pivot Grid control icon from the Tool box into design panel. You can find control in Toolbox by search.



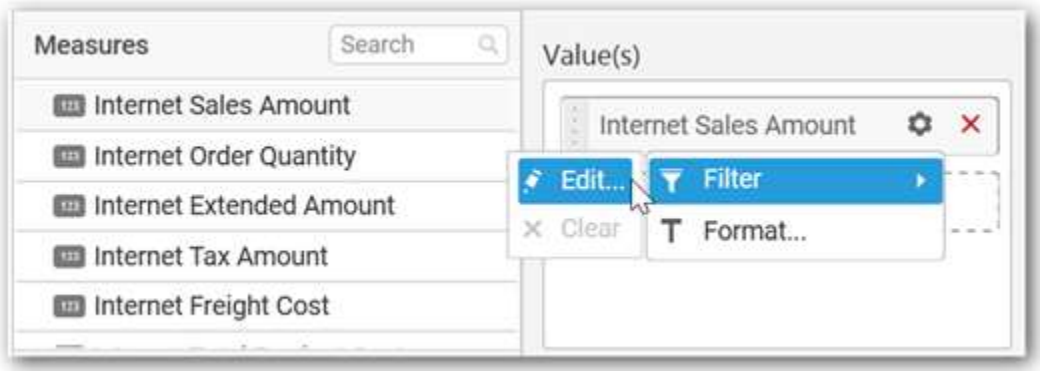
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



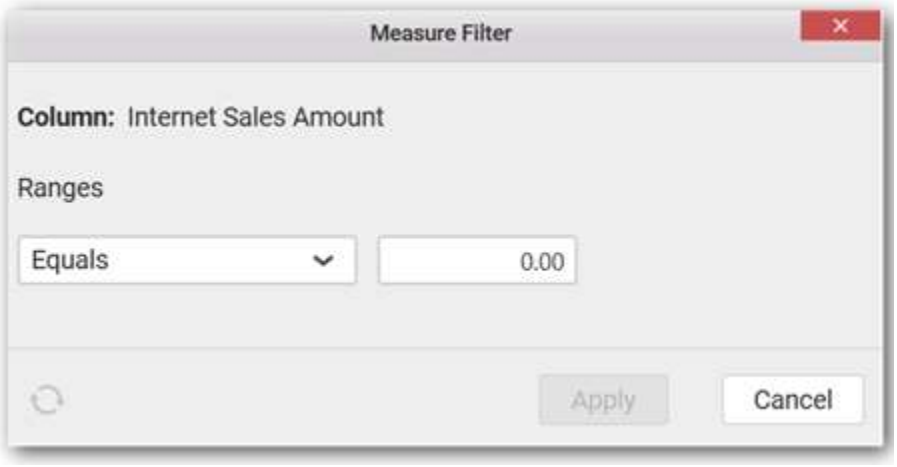
Drag and drop a column under Measures category into Value(s).



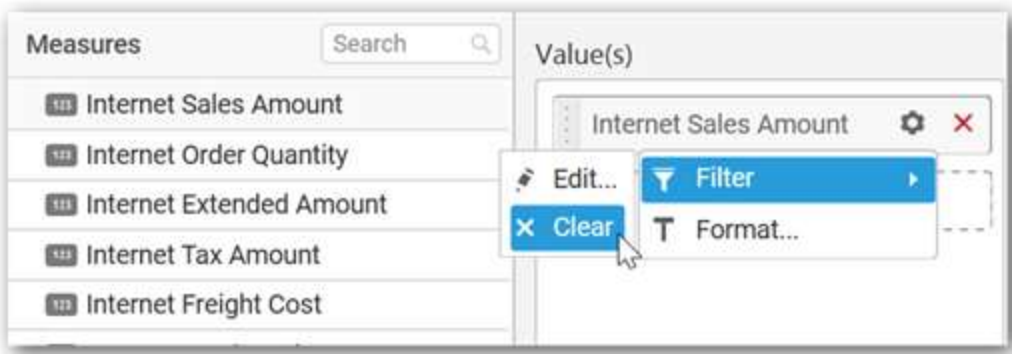
Define the filter criteria to match through choosing Edit option in Filter menu item.



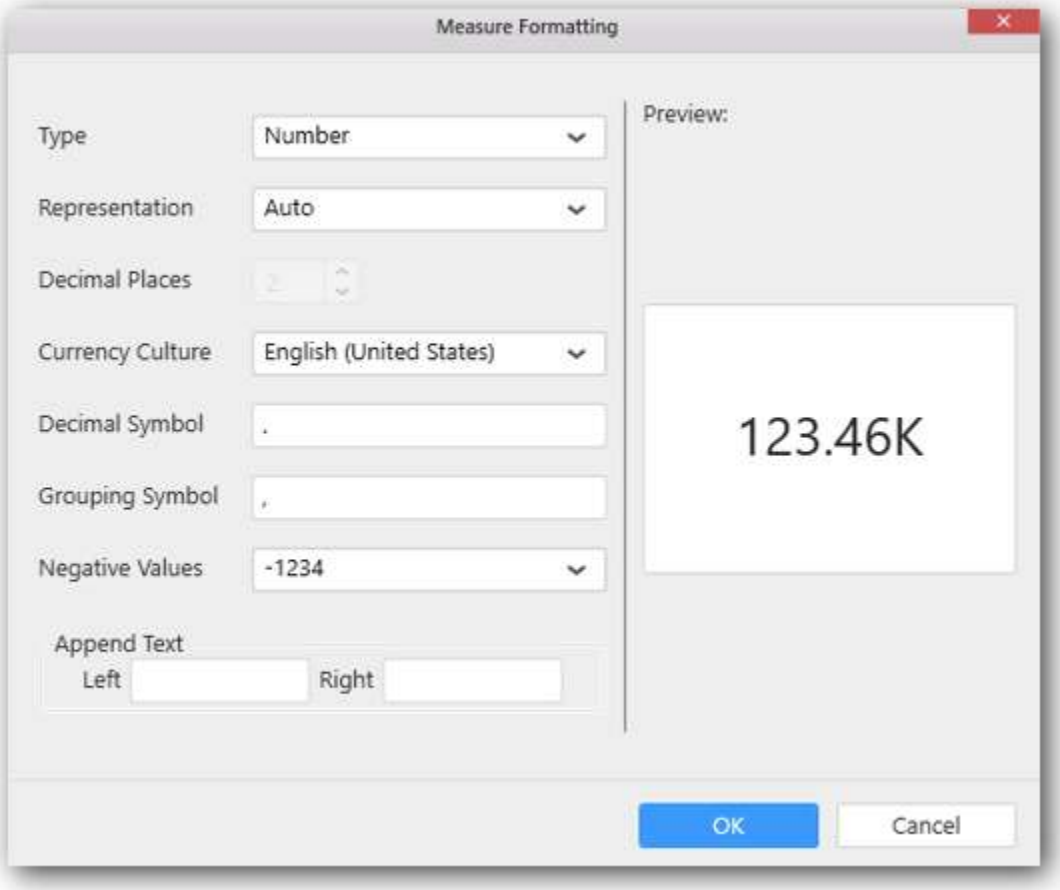
The Measure filter dialog will be shown where you can choose the filter condition and apply the condition value.



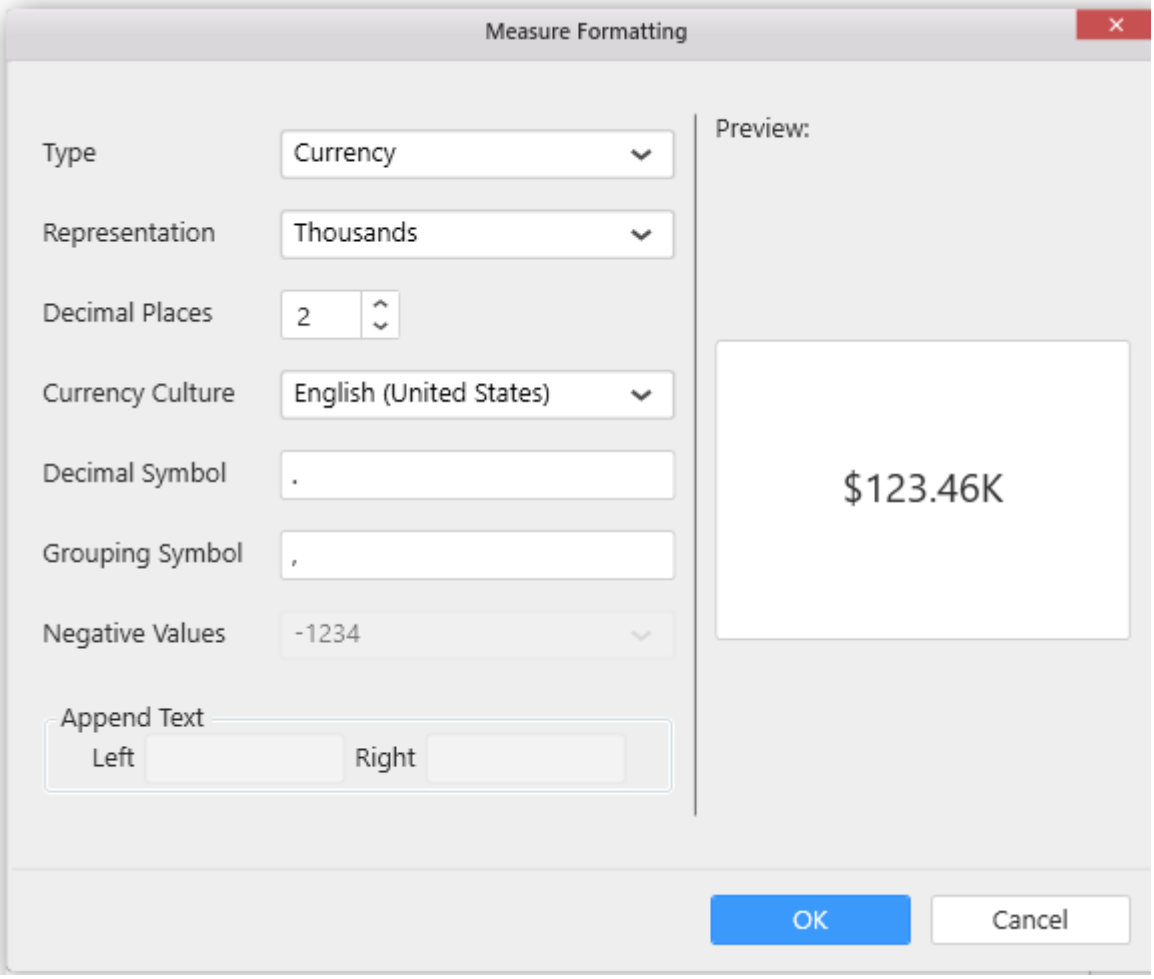
Select Clear option to clear the defined filter.



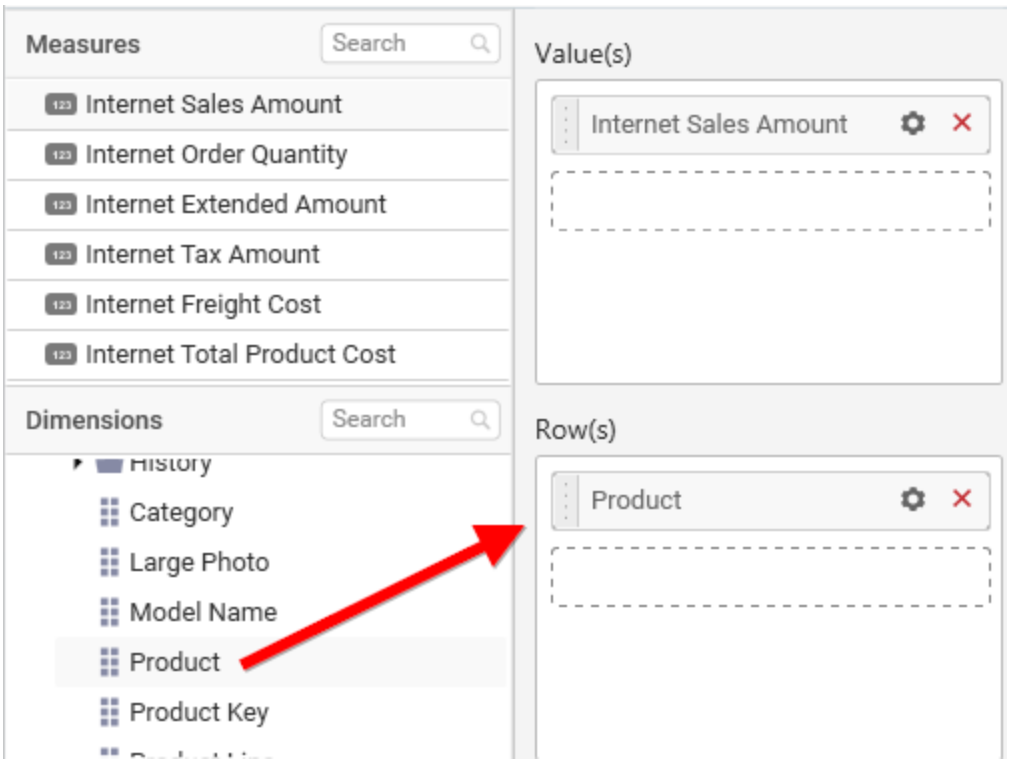
Select Format option to define the display format to the values in the column through Measure Formatting window.



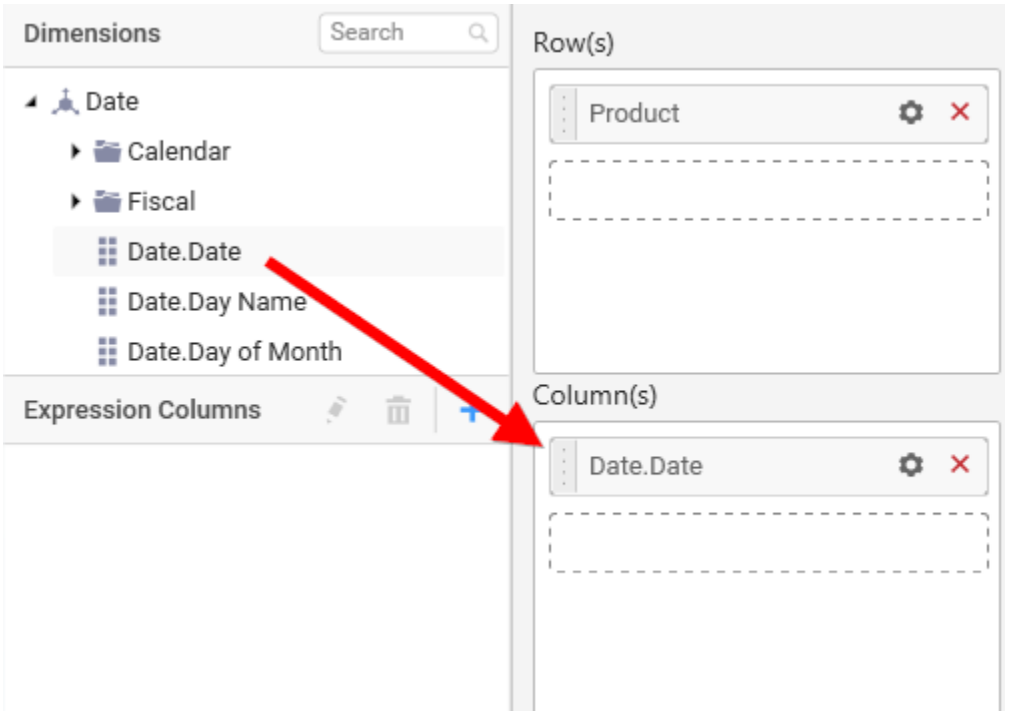
Choose the options you need and click OK.



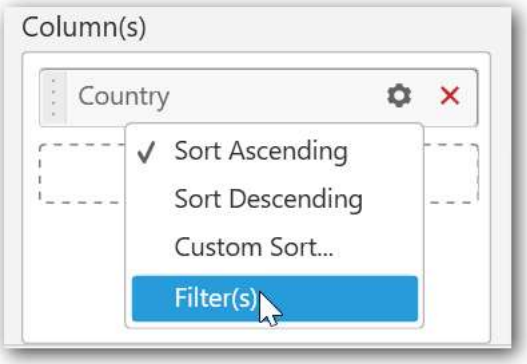
Drag and drop a dimension level or hierarchy element under Dimensions category into Row(s).



Drag and drop a dimension level or hierarchy element under Dimensions category into Column(s).



Define filter criteria through Filter(s)… menu item in the Settings drop down menu.



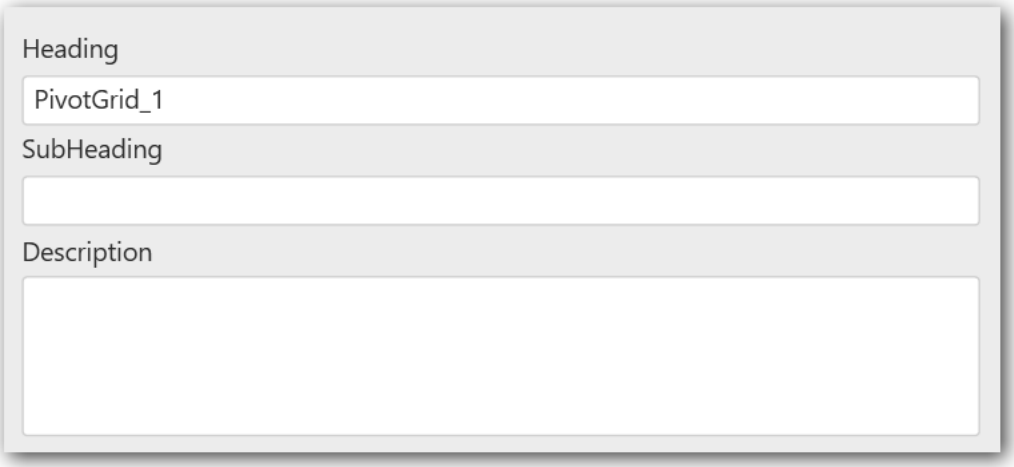
To know more about filters, refer here.

Here is an illustration,

How to format Pivot Grid?

You can format the Pivot grid for better illustration of the view that you require, through the settings available in Properties pane.

General Settings



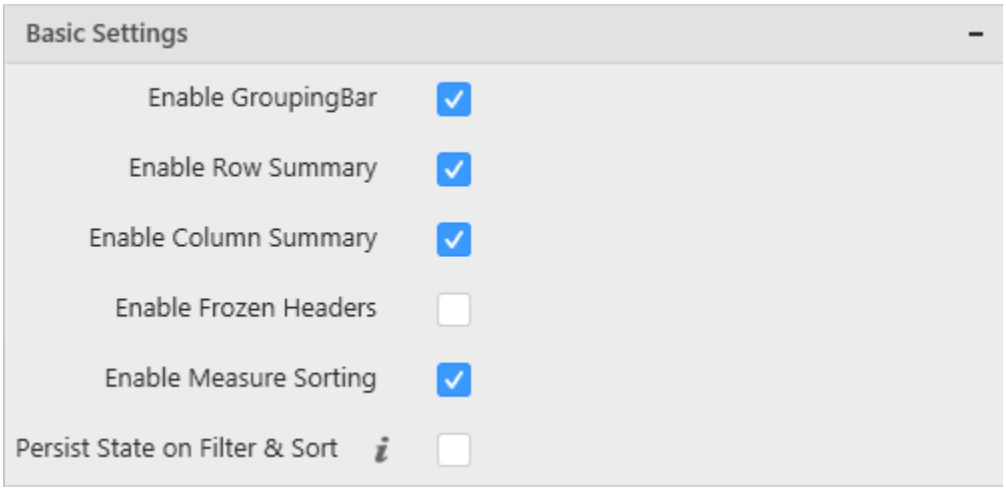
Header

This allows you to set title for this pivot grid widget.

Description

This allows you to set description for this pivot grid widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings



Enable GroupingBar

You can enable/disable the grouping bar which contains in line filtering and sorting option.

Enable Row Summary

You can enable/disable the grand total summary row.

Enable Column Summary

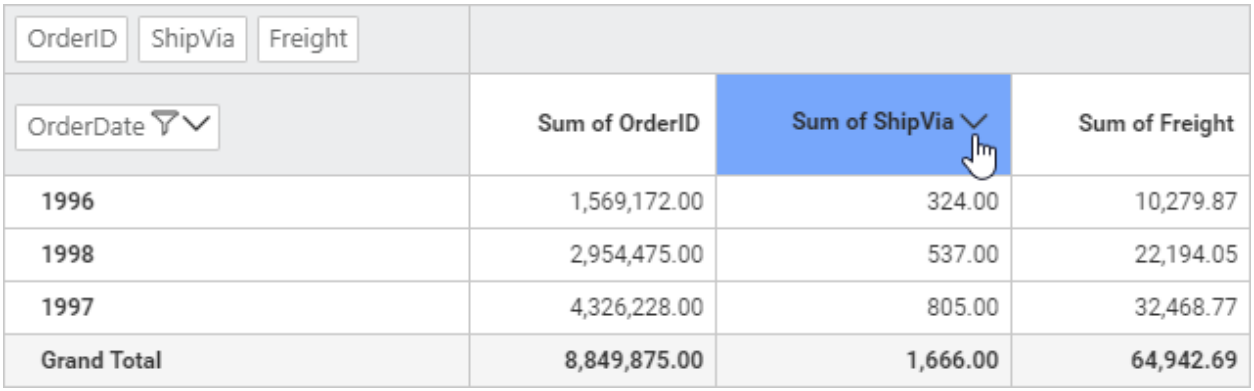
You can enable/disable the grand total summary column.

Enable Frozen Headers

Allows you to freeze the header of the Grid so that it will be always visible when scrolling the content with a large number of rows or columns providing a precise view.

Enable Measure Sorting

You can sort the rows based on the measure values associated with a specific column. This can be achieved by clicking on the respective headers.



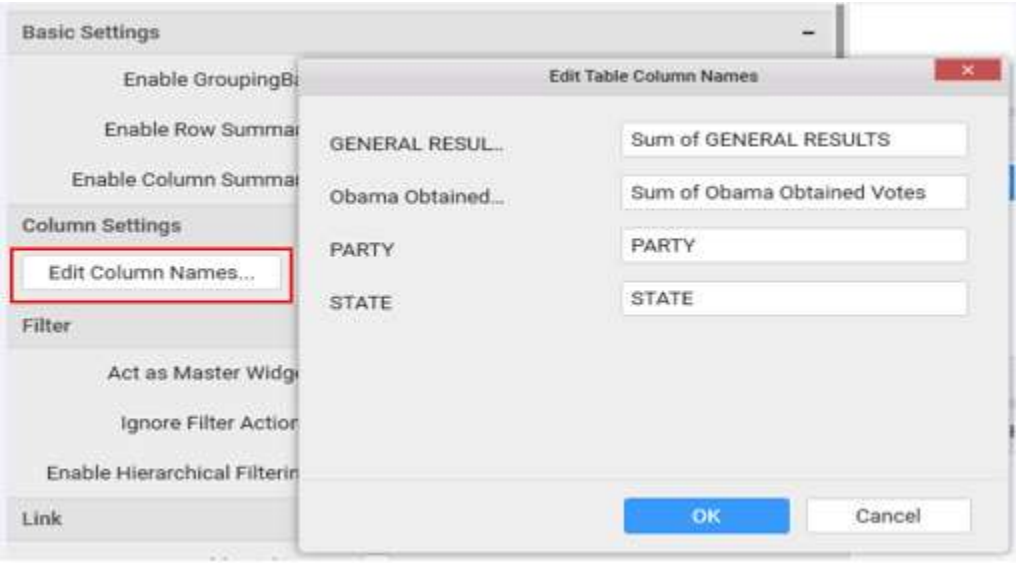
Persist State on Filter & Sort

PivotGrid widget will persist the drilled states for initial rendering, maximizing and export operations. This property allows to persist the drilled states also for Filtering and Sorting operations.

Columns Settings

Edit Column Name

You can edit the binded column name displayed in the widget.



Filter settings

Act as Master Widget

This allows you to define pivot grid widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this pivot grid widget to ignore responding to the filter actions applied on other widgets in dashboard.

Enable Hierarchical Filtering

Through this option, you can enable/disable hierarchical Top N filtering. While applying Top N filter with multiple dimension columns, the data returned can be customized based on whether the filtering need to be done as flat or based on the hierarchy of dimension columns added. When Flat is set, the least number set as top will be applied for the whole data. When Hierarchical is set, the Top N will be applied for each individual column separately based on the number set for each column.

Below example shows data of 3 Country and its 2 Cities where the sales is high.

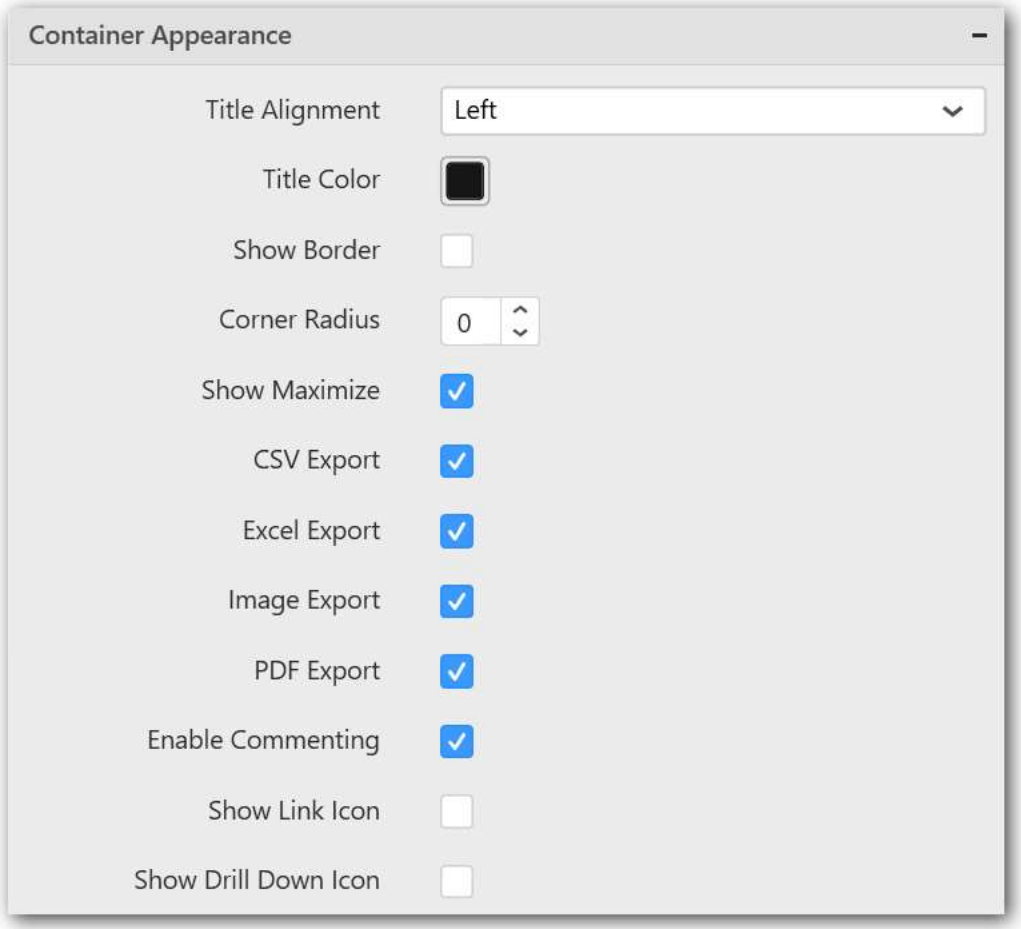
Flat Top N

Hierarchical Top N

Link Settings

You can enable linking and configure to navigate either to a published dashboard URL or to a general URL with or without parameters. For more details, refer Linking.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Show Maximize

This allows you to enable/disable the maximized mode of this pivot grid widget. The visibility of the maximize icon in widget header will be defined based on this setting. Clicking this icon in viewer will show the maximized view of the pivot grid widget.

CSV Export

This allows you to enable/disable the CSV export option for this pivot grid widget. Enabling this allows you to export the summarized data of the widget view to CSV format.

Excel Export

This allows you to enable/disable the Excel export option for this pivot grid widget. Enabling this allows you to export the summarized data of the widget view to XLSX format.

Image Export

This allows you to enable/disable the image export option for this pivot grid widget. Enabling this allows you to export the view of the widget to image format (*.JPG) in viewer.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

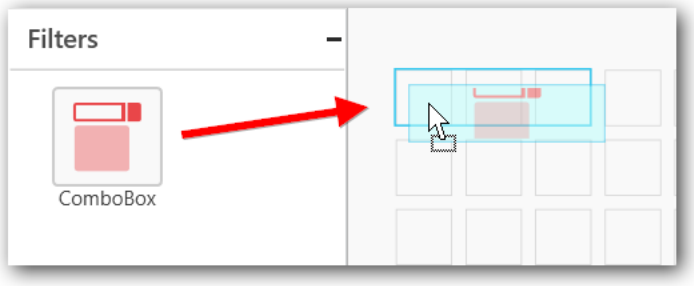
ComboBox

ComboBox enables you to filter based on single or multiple items selection in dropdown list. To bind a ComboBox, a minimum requirement of 1 column is needed.

How to configure flat table data to ComboBox?

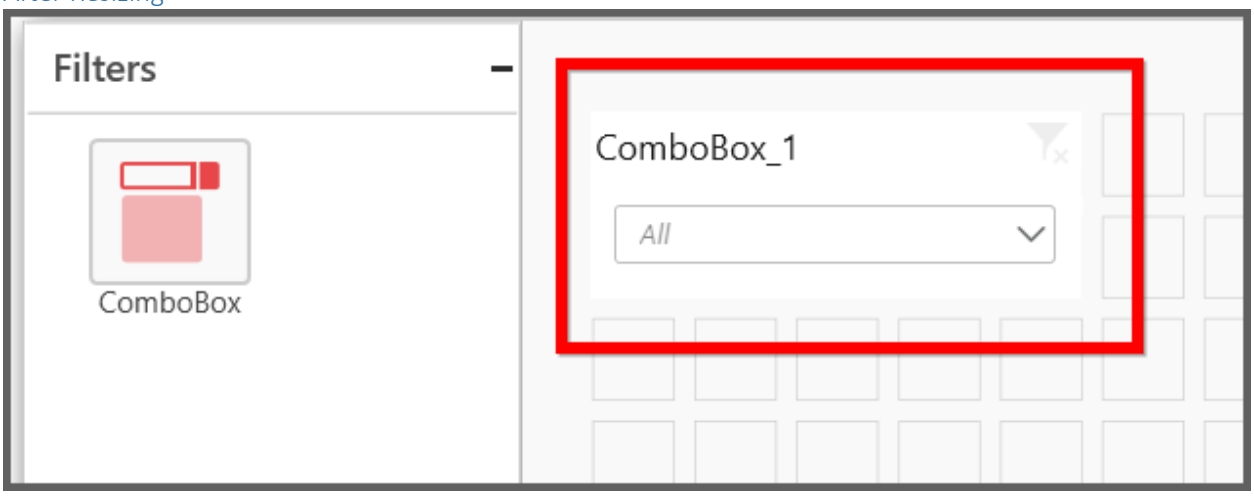
The following procedure illustrates data configuration of ComboBox.

Drag and drop ComboBox widget from the Toolbox into design panel and resize into your required size. You can find widget in Toolbox by search.

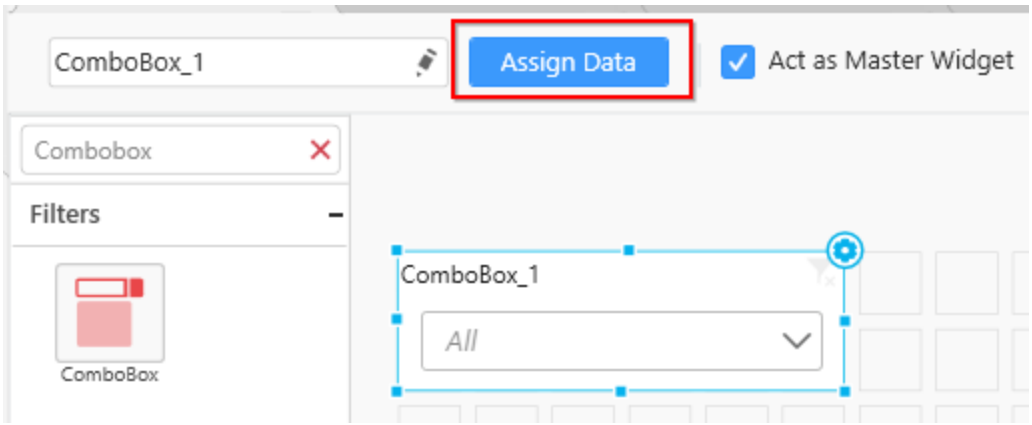


Before Resizing

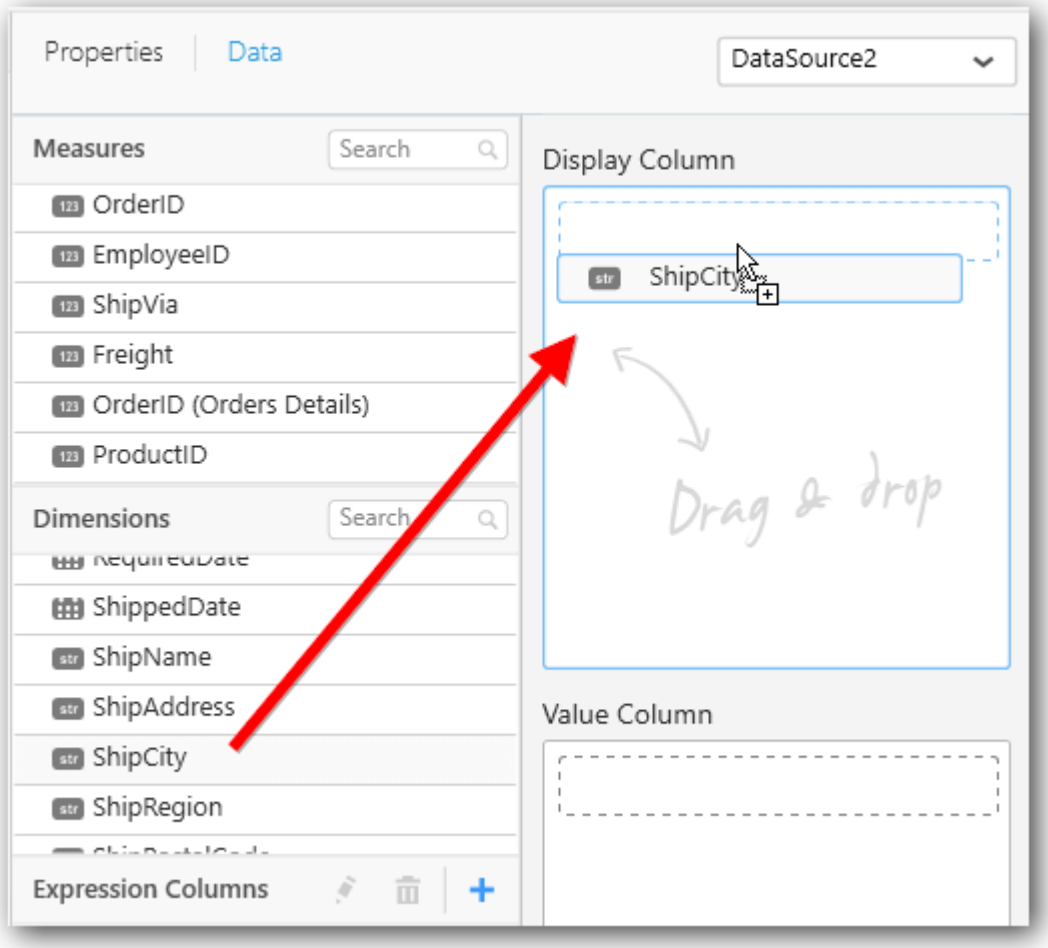
After Resizing



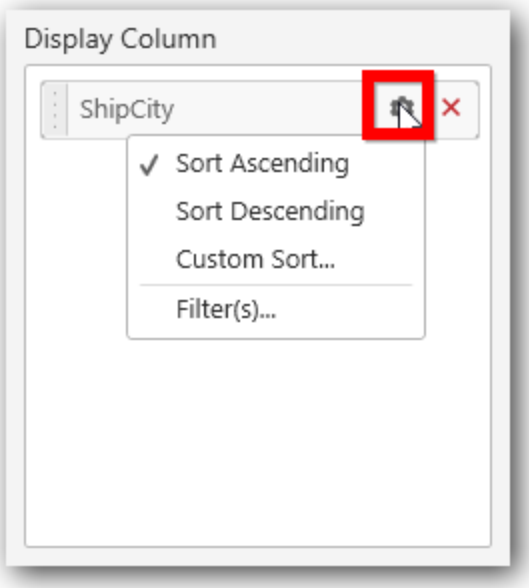
Select the dropped widget using mouse and click the Assign Data button at Design Tools Pane to open the Data configuration pane.



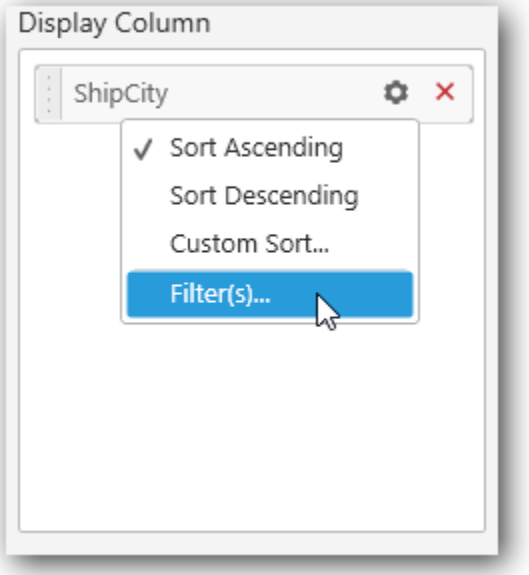
Drag and drop a column from Measures or Dimensions or Expression Columns category to Display Columns section.



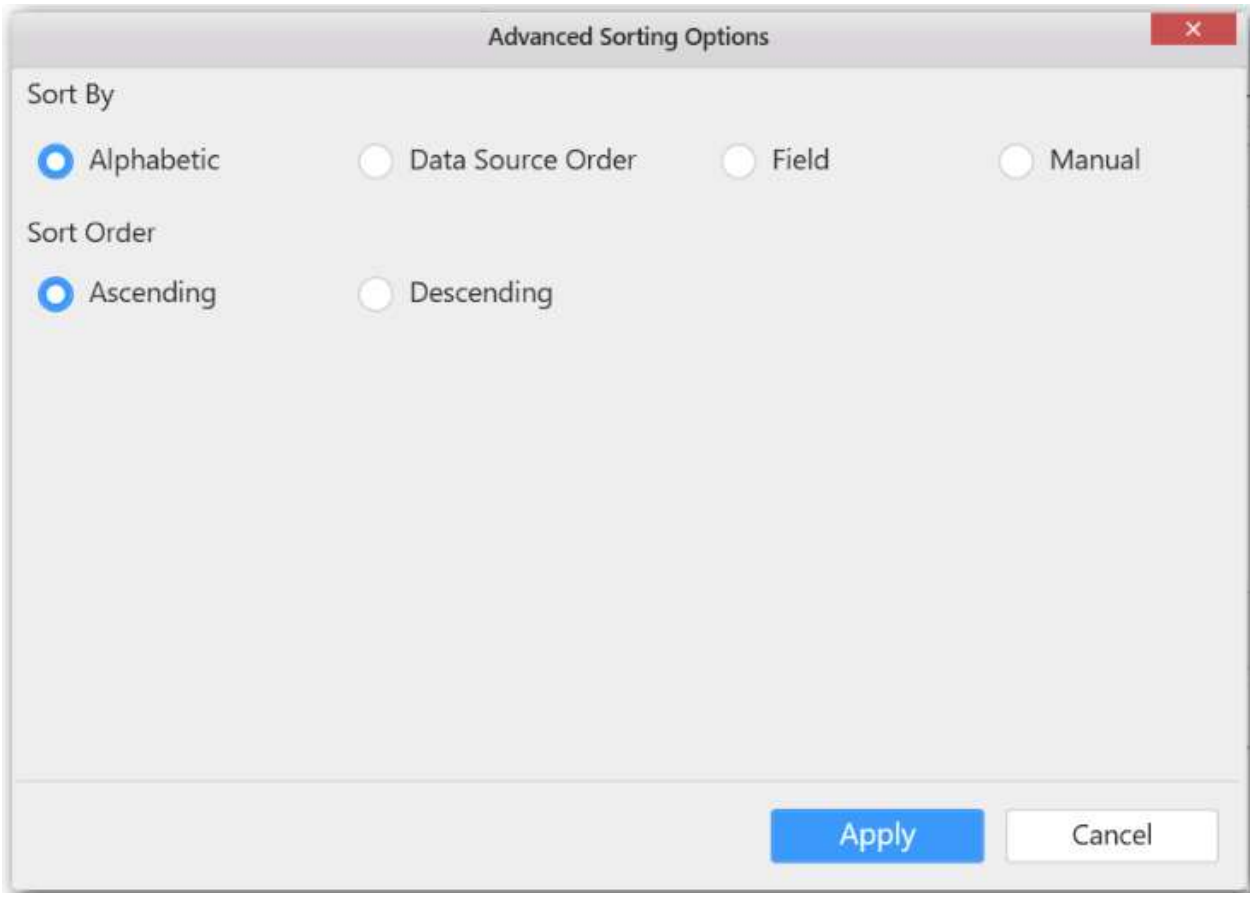
Define the sort order of the dropped column through the Settings drop down menu.



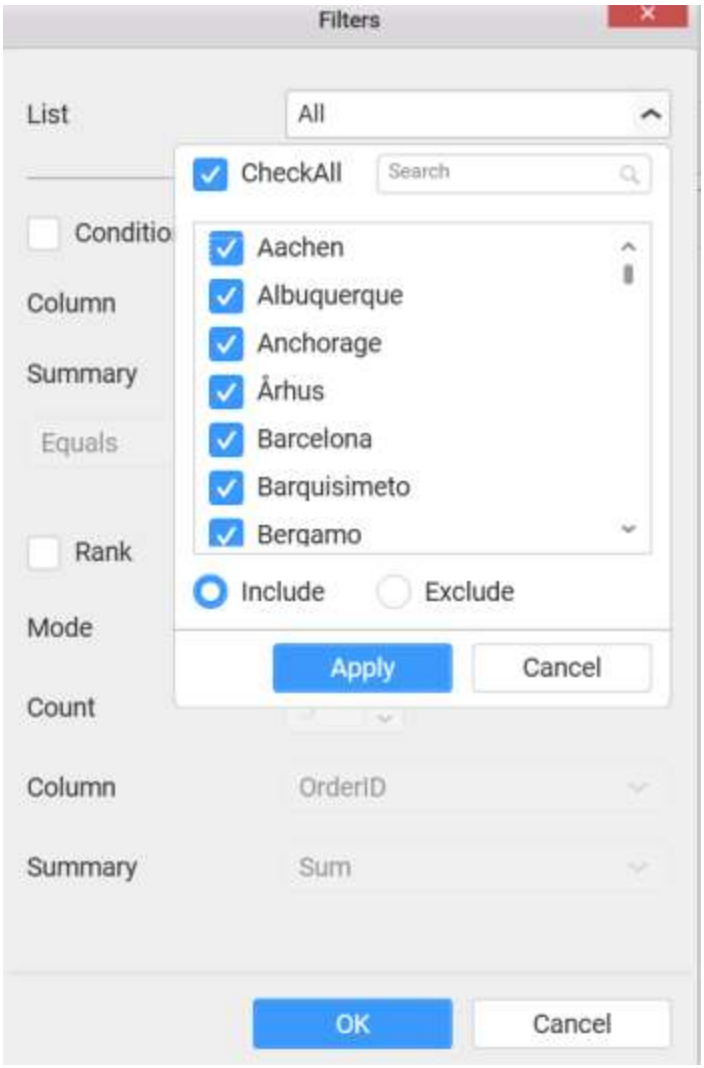
Define filter criteria through the Filter(s)... menu item in the Settings drop down menu.



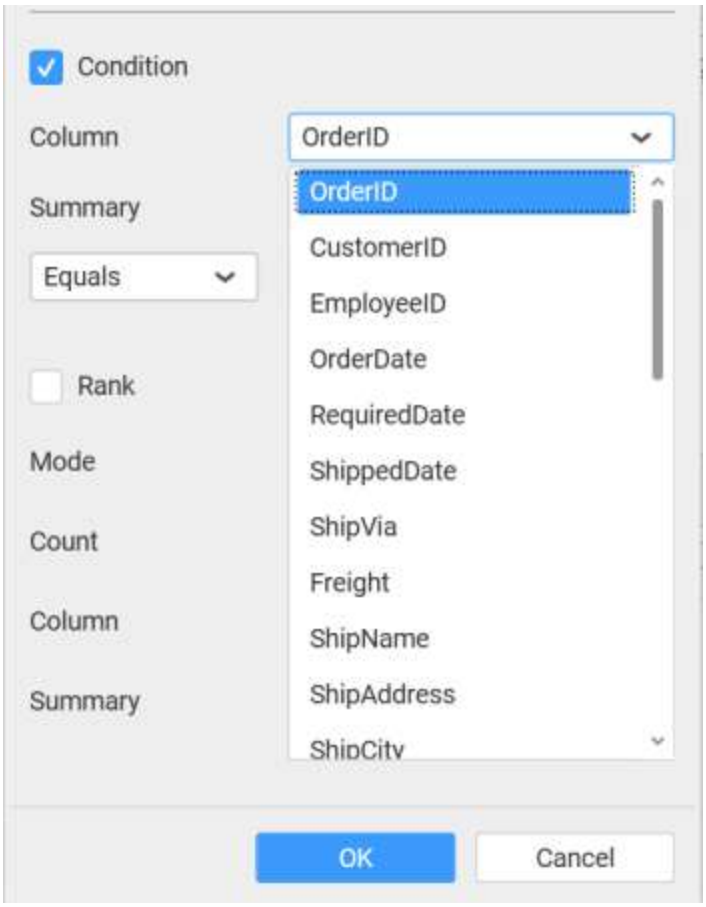
Custom sort

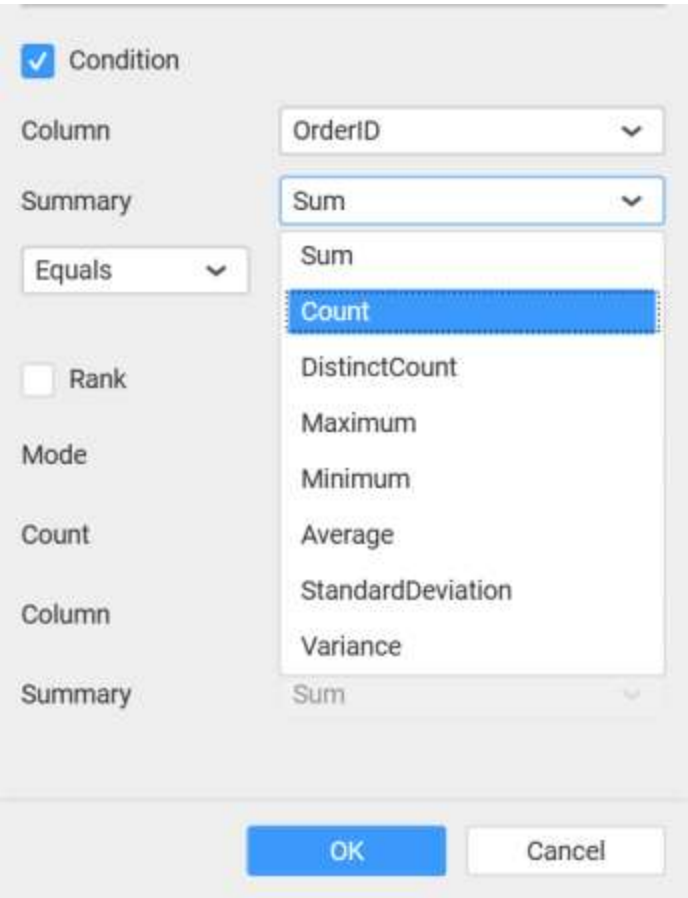


Select the specific item to filter the element and `CheckAll` is used either to check all the item or to select the specific item. `Include` and `Exclude` is used to include and exclude the selected elements. Click the `Apply` button to apply the selection.

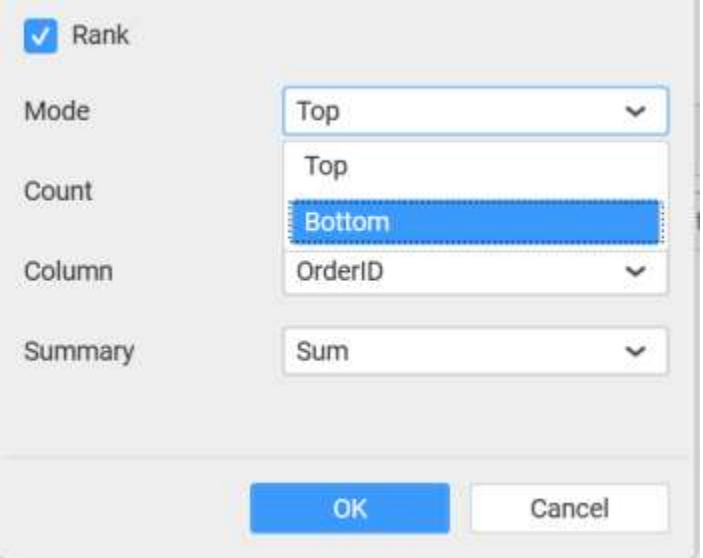


Select the Condition option to change the Column elements and Summary type by selecting the required column name and summary type.

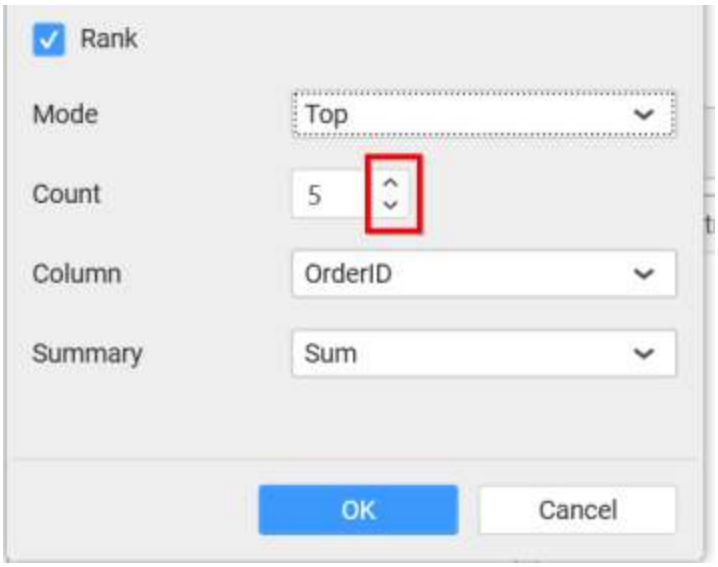




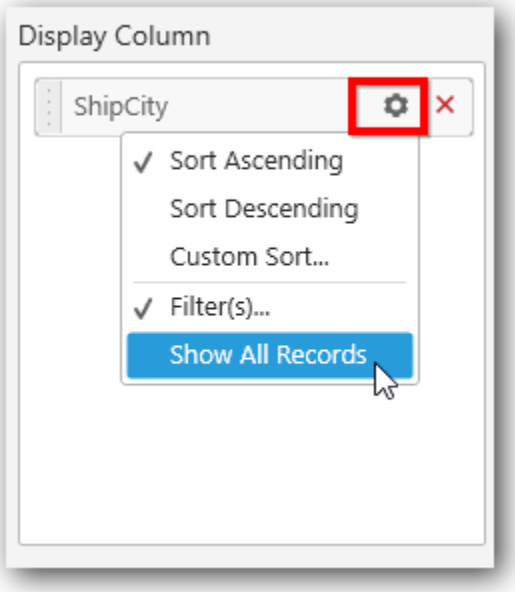
Select the Rank option to enable filters and select the Mode as either top or bottom.



You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.



Clear the filters by selecting the Show All Records in the Settings dropdown menu.



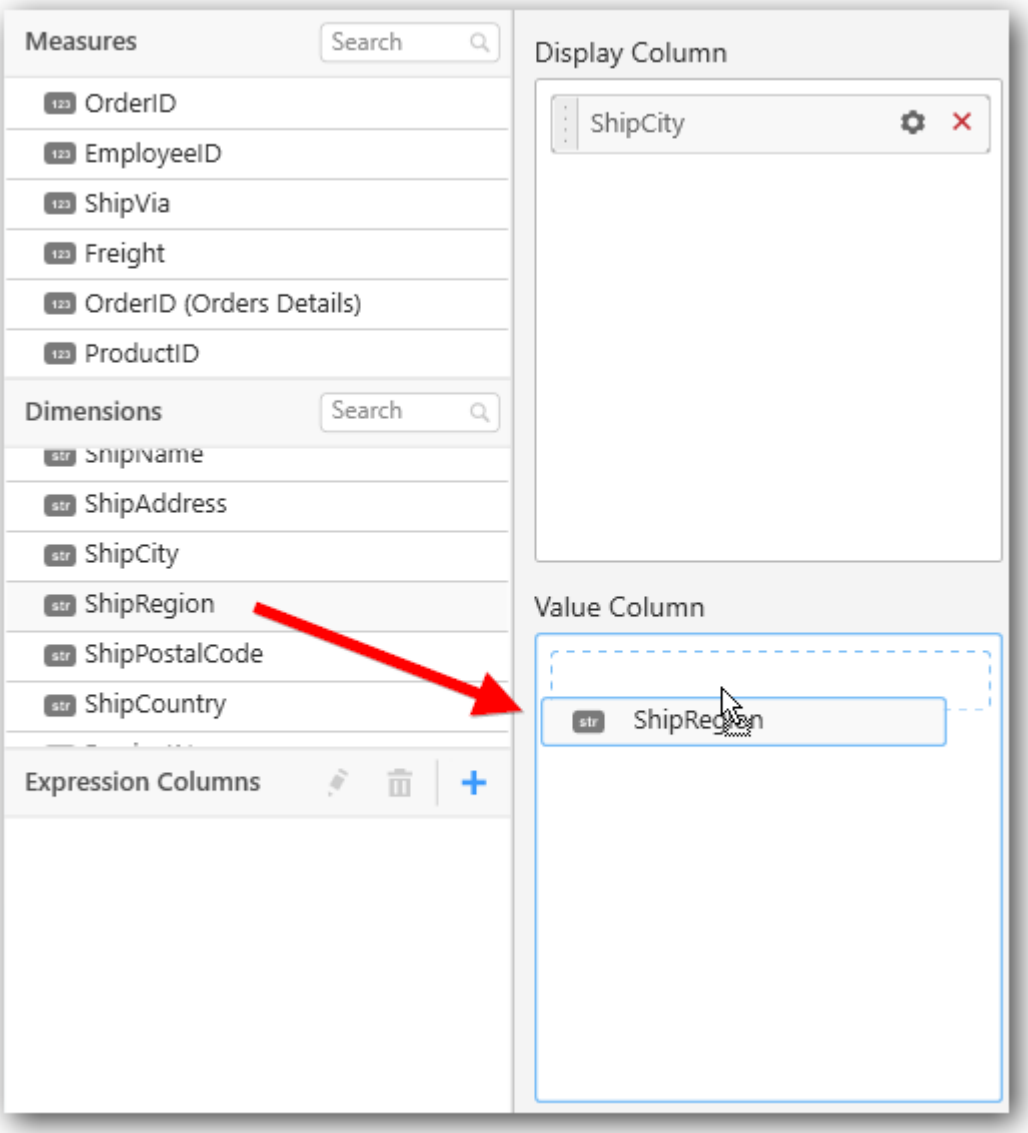
Value Column

We can drag and drop a column from Measures or Dimensions or Expression Columns category to Value Columns section.

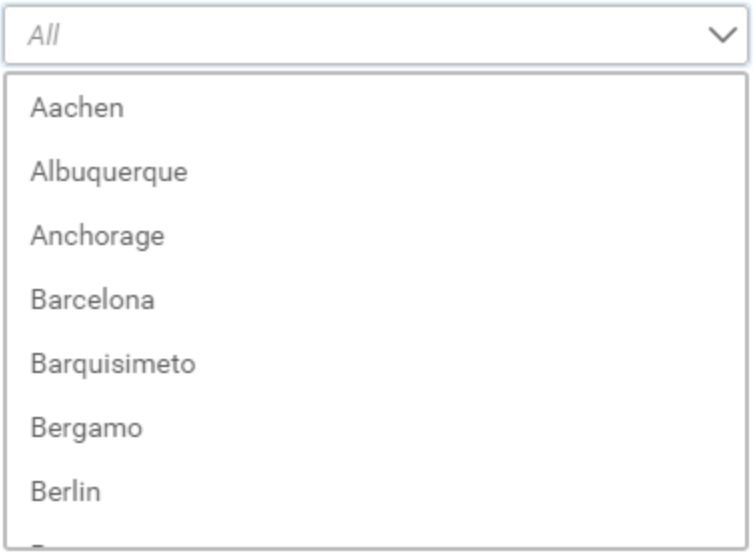
Note: This column section value must be unique.

Display column values will be shown visually in the widget but internal operation based on value column.

Value column will not be applicable for custom format and relative date format.



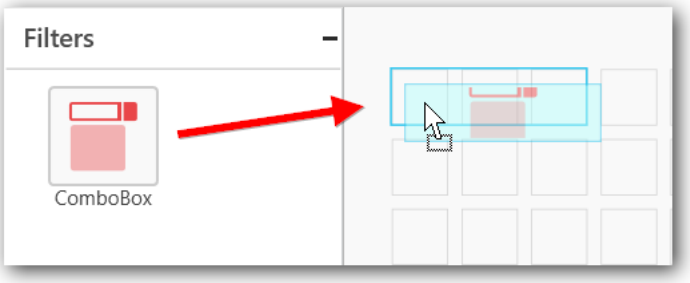
Here is an illustration,



How to configure the SSAS data to ComboBox?

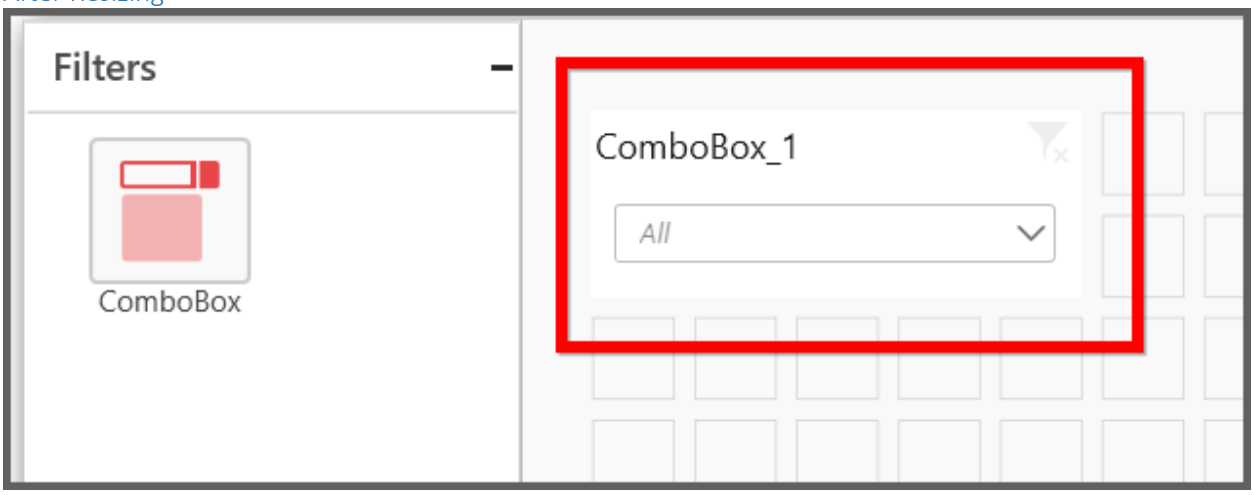
Following steps illustrates configuration of SSAS data to ComboBox.

Drag and drop ComboBox widget from the Toolbox into design panel and resize into your required size. You can find widget in Toolbox by search.

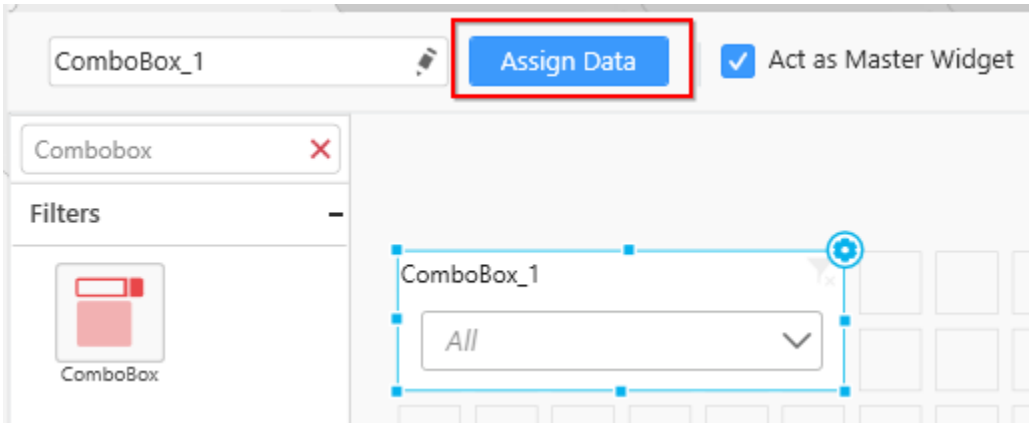


Before Resizing

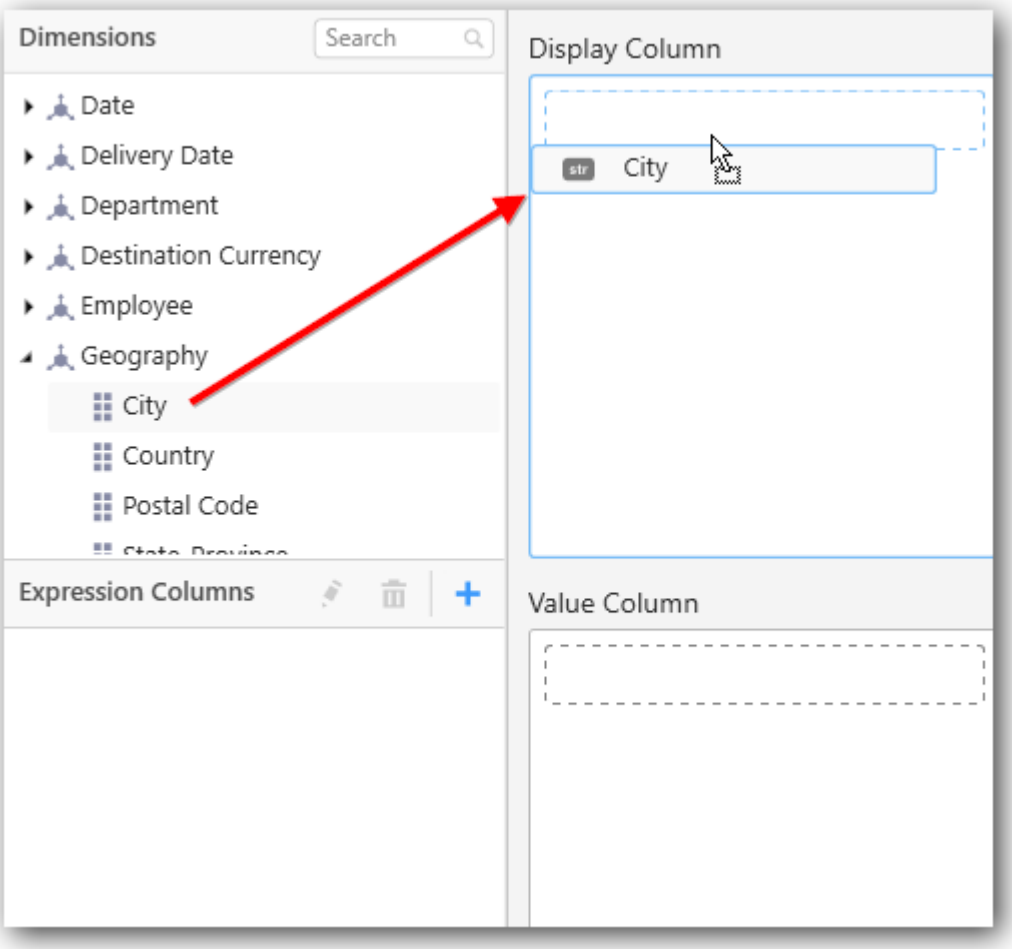
After Resizing



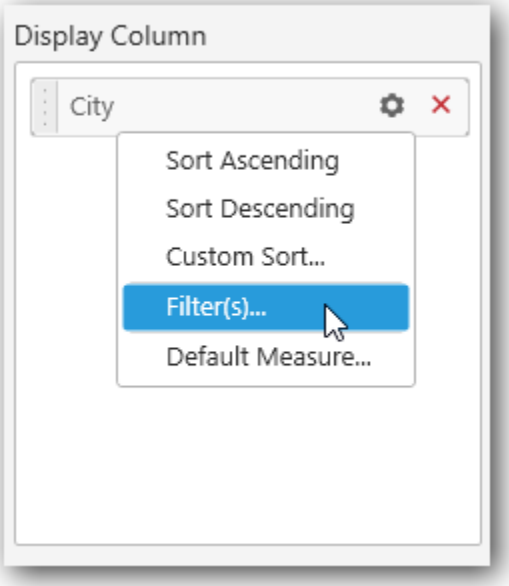
Select the dropped widget using mouse and click the Assign Data button at Design Tools Pane to open the Data configuration pane.



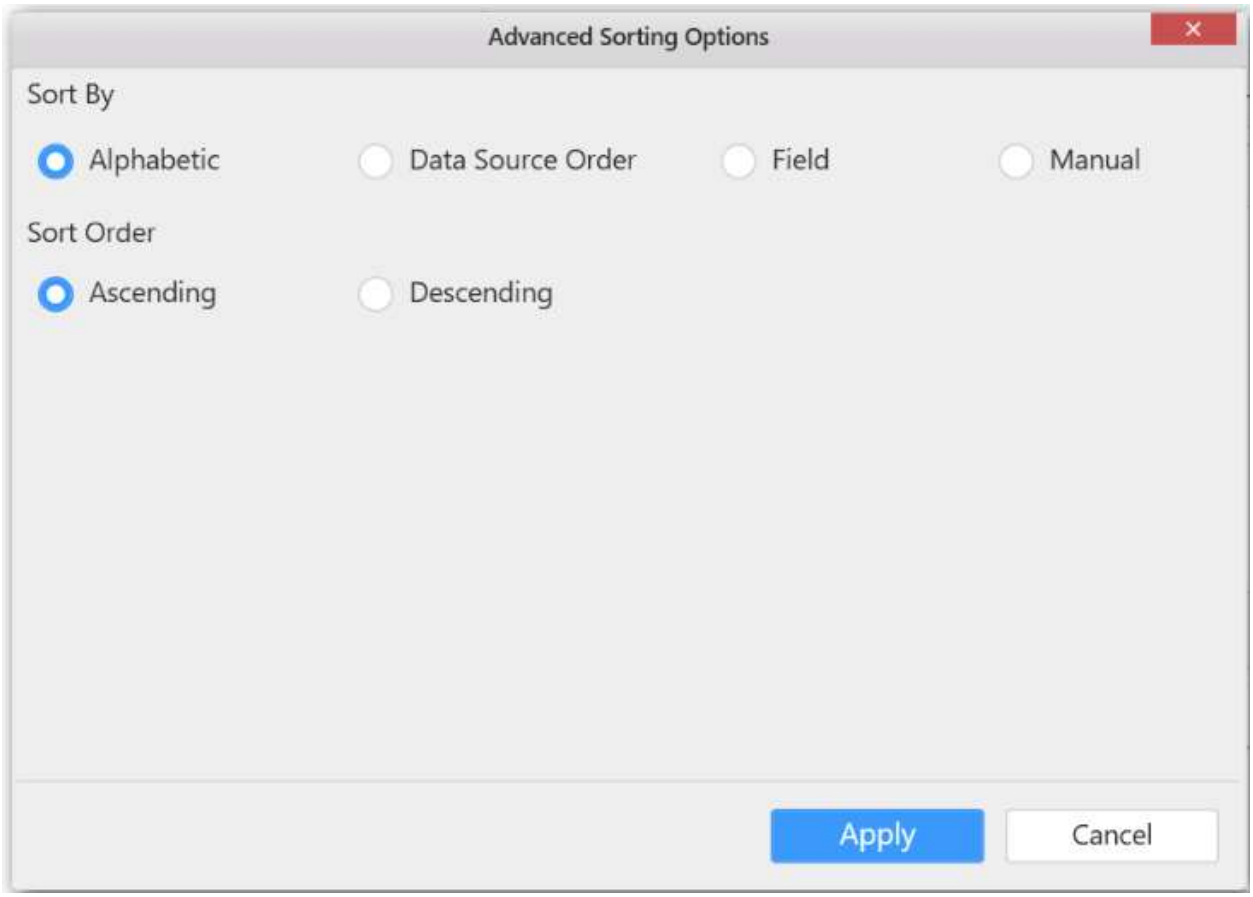
Drag and drop a dimension level or hierarchy column under Dimensions category into Display Column section.



Define filter criteria through Filter(s)... menu item in the Settings drop down menu.

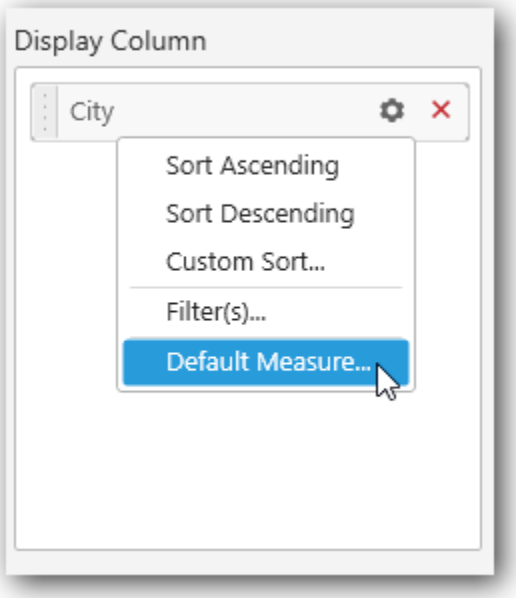


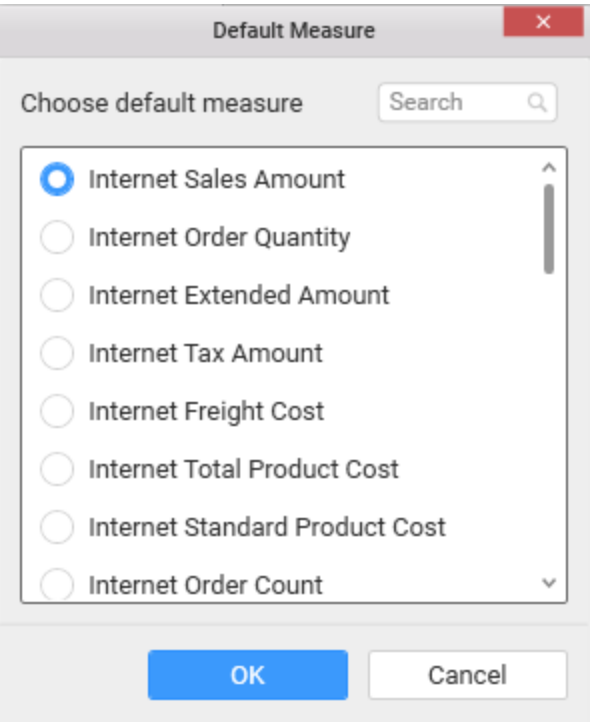
Custom sort



To know more about filters, refer here.

Define a default measure to the dropped dimension through the Settings dropdown menu against which dimension values need to be categorized.





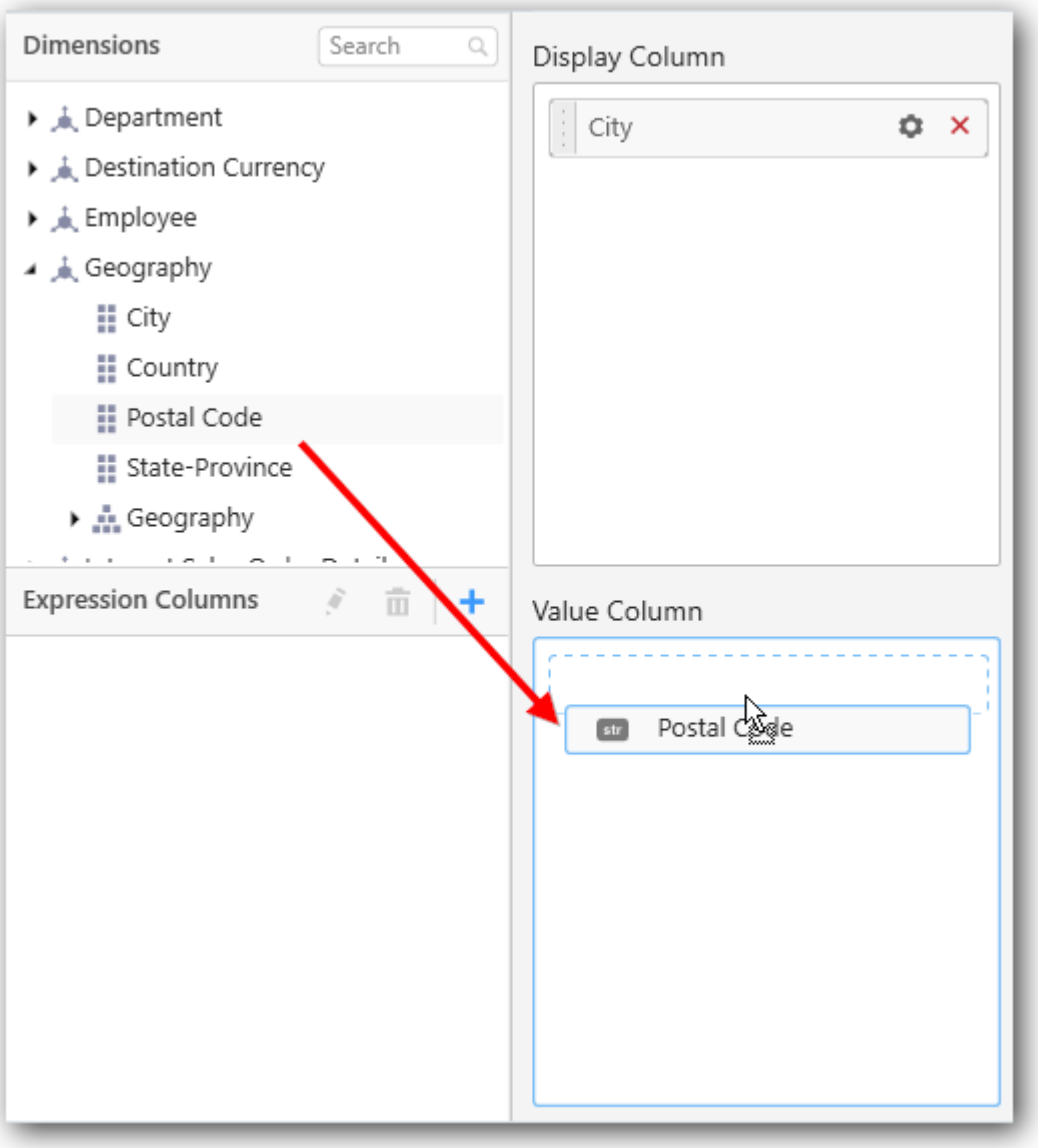
Value Column

We can drag and drop a column from Measures or Dimensions or Expression Columns category to Value Columns section.

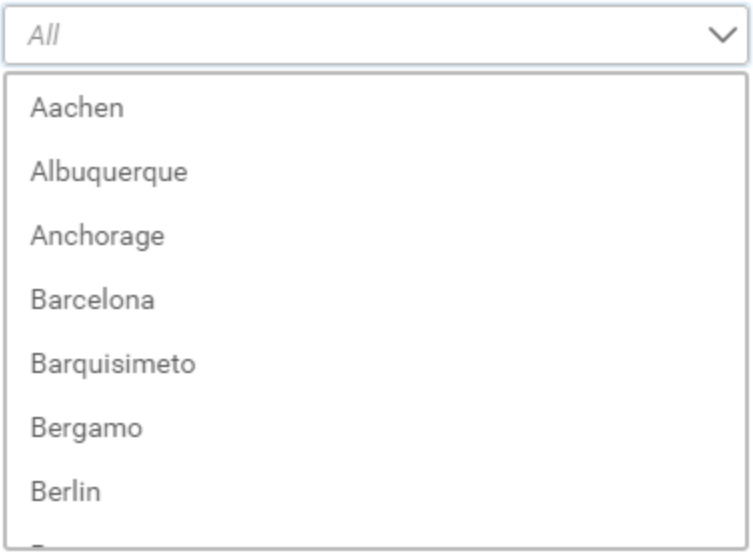
Note: This column section value must be unique.

Display column values will be shown visually in the widget but internal operation based on value column.

Value column will not be applicable for custom format and relative date format.



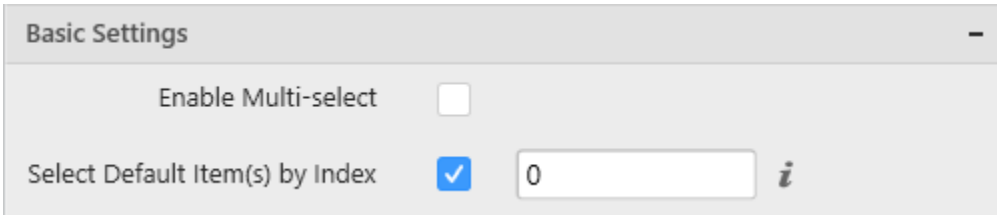
Here is an illustration,



How to format ComboBox?

You can format the ComboBox for better illustration of the view that you require, through the settings available in Properties pane. This pane can be opened from design view through clicking the Settings icon at top right corner of the widget.

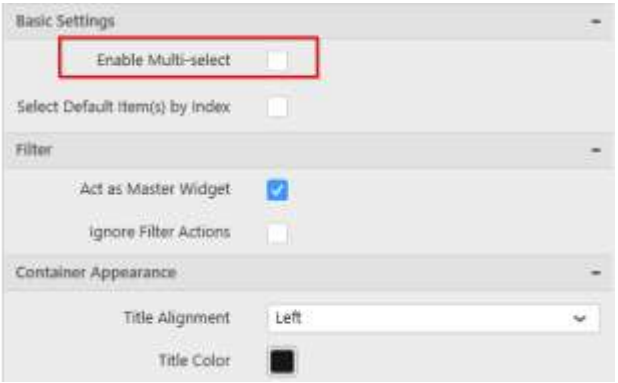
Basic Settings



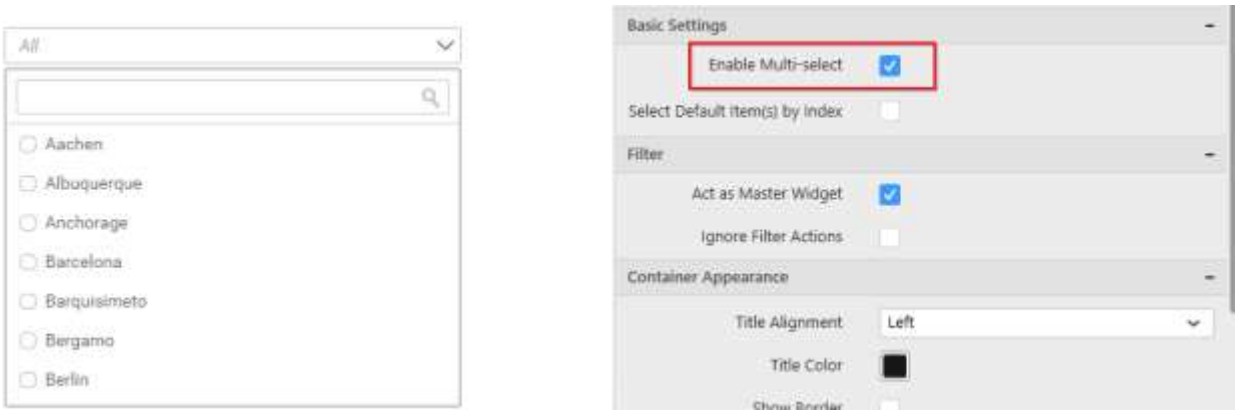
Enable Multi-select

This allows you to define single/multiple item selection in dropdown list.

Single Selection



Multiple Selection



Select Default Item(s) By Index

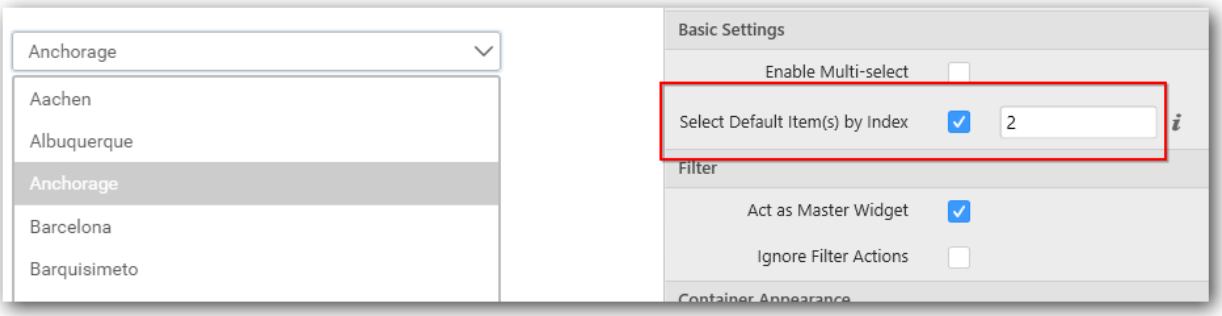
This allows you to select one or more values by default, based on their corresponding indices mentioned. In single selection mode, you can select only one index and in multi-selection mode, you can select more than one values and also by range.

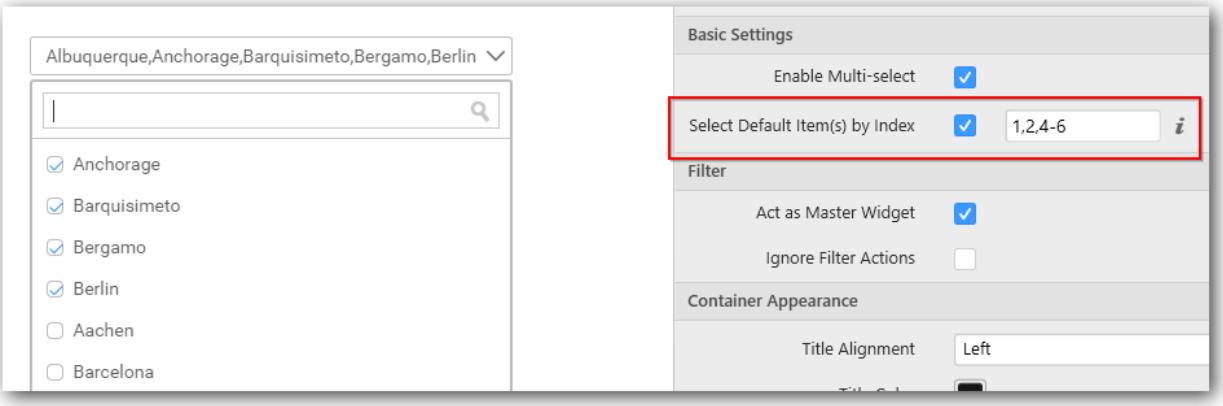
Indexing Patterns

- Supports only positive integer values starting from 0.
- Comma(,) is used to separate the indices. Example: 1,2,5.
- Hyphen(-) is used to define the index range. Example: 1-4,7,9-11.

Note: You can select maximum index value of 100 and selection is not applicable more than 100.

In multi-selection mode, by default the selected indices are sorted to the top order.





General Settings

Heading

This allows you to set title for this ComboBox widget.

SubHeading

This allows you to set sub-title for this ComboBox widget.

Description

This allows you to set description for this ComboBox widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Filter Settings

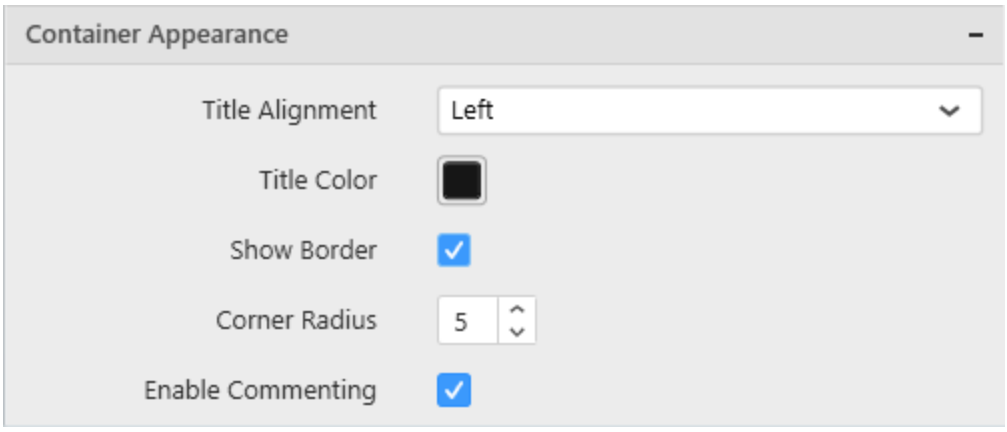
Act as Master Widget

This allows you to define this ComboBox widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this ComboBox widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

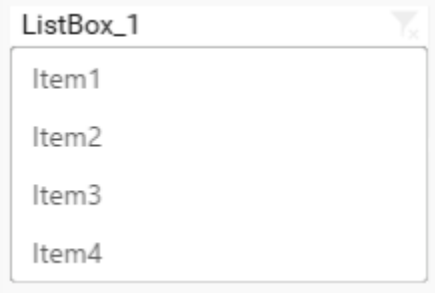
This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

List Box

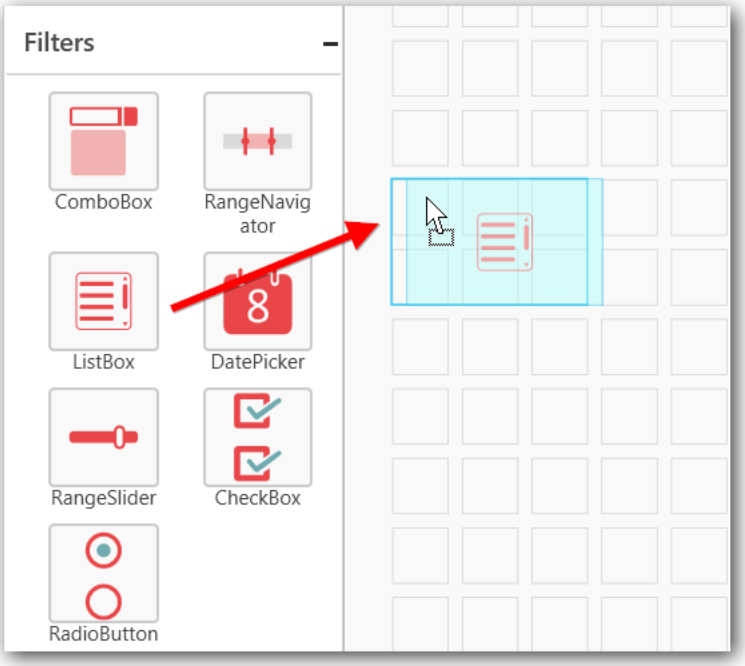
List Box enables you to filter based on single or multiple items selection in a list. To configure a list box, a minimum requirement of 1 column is needed.



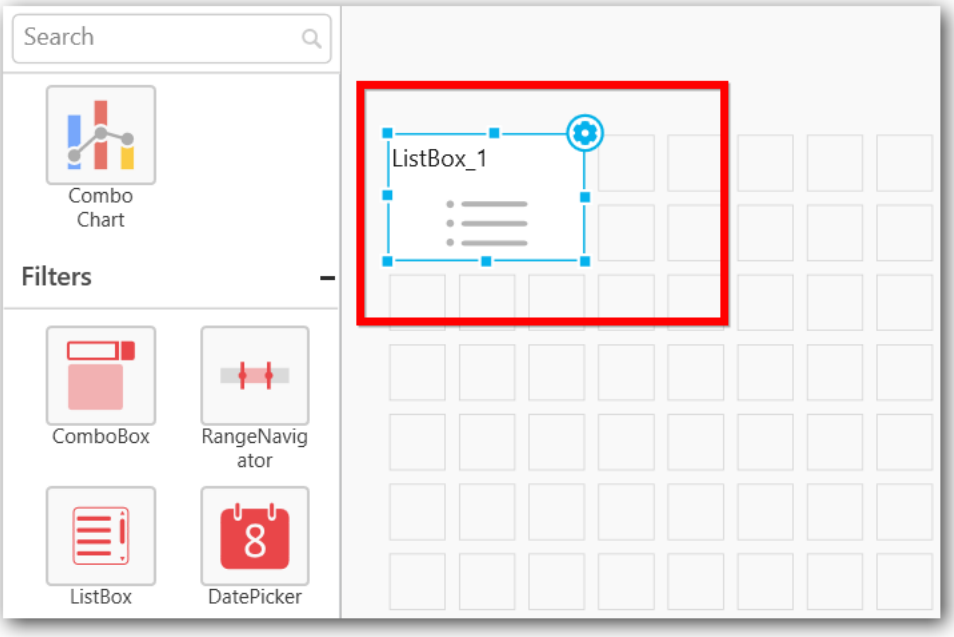
How to configure the flat table data to ListBox?

The following procedure illustrates data configuration of ListBox.

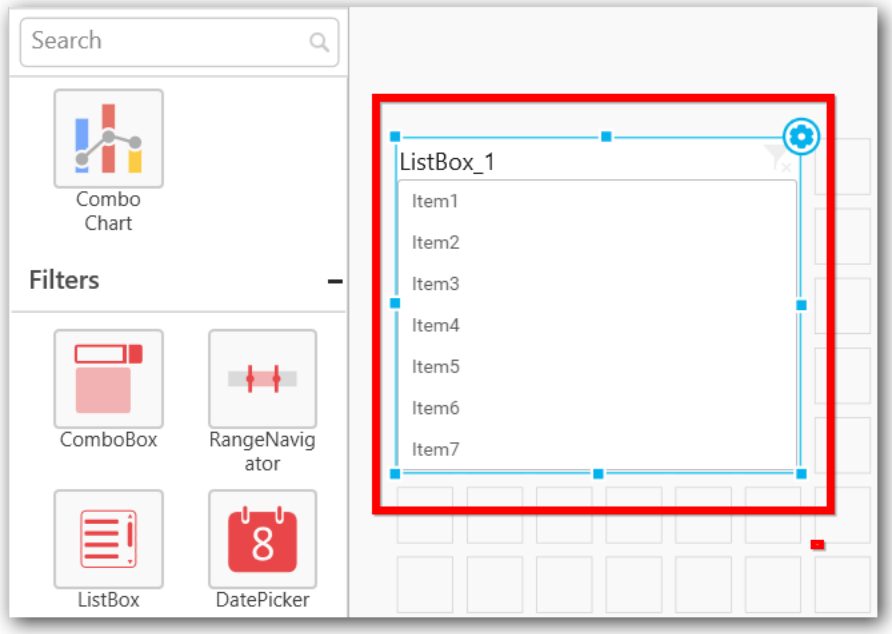
Drag and drop List Box control icon from the Tool box into design panel. You can find control in Toolbox by search.



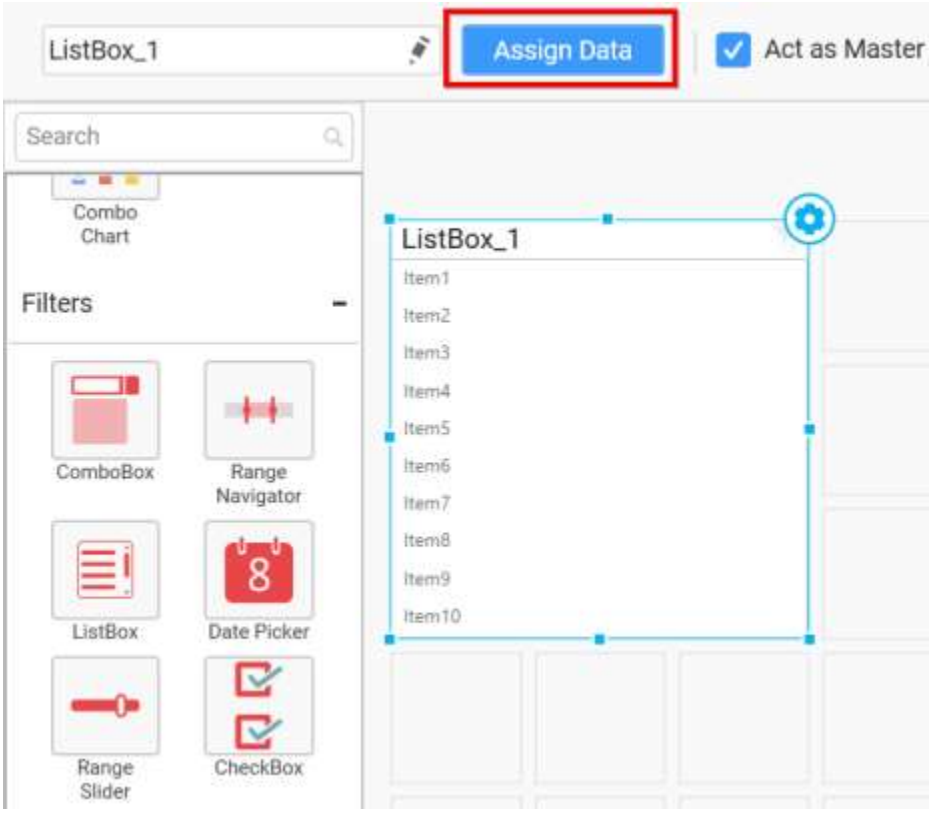
Before Resizing



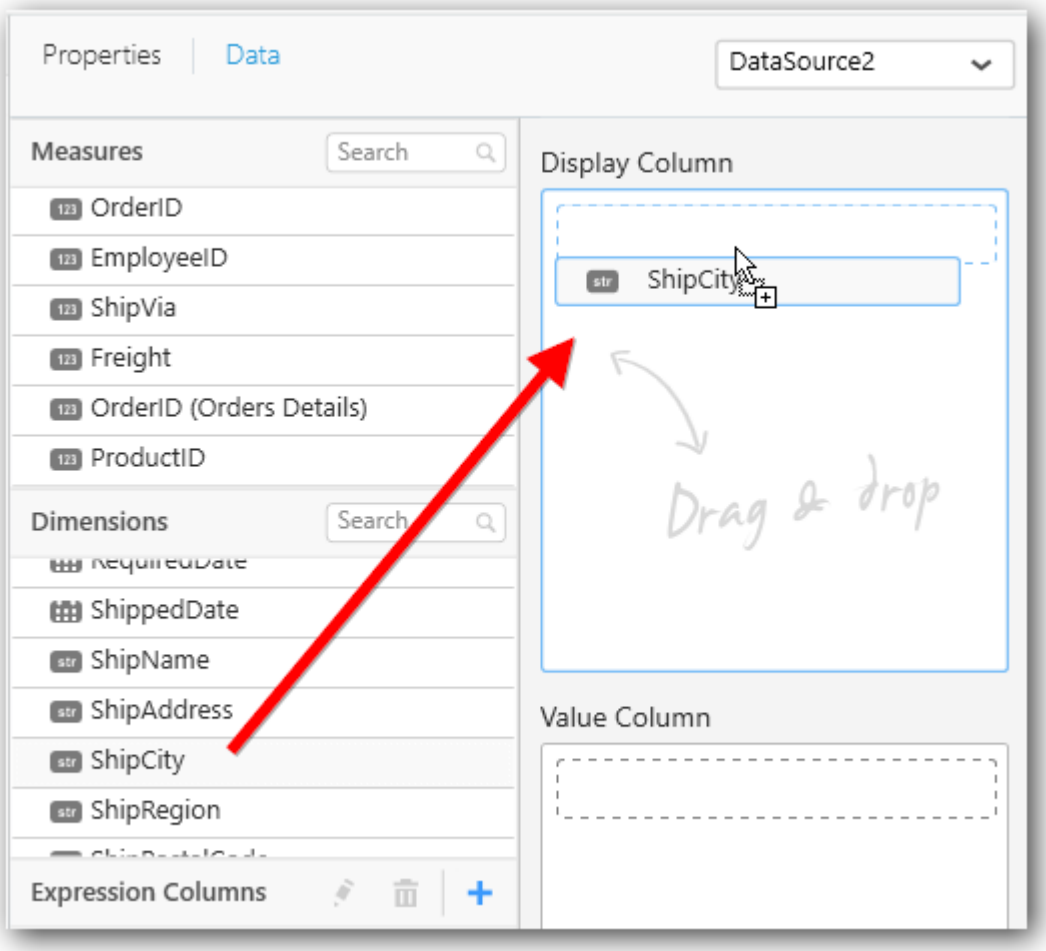
After Resizing



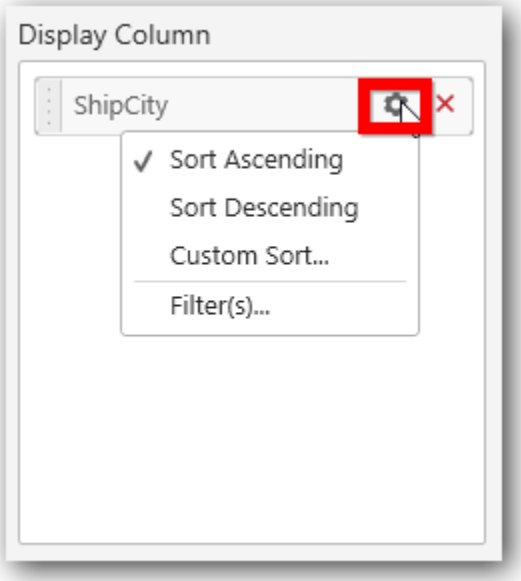
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



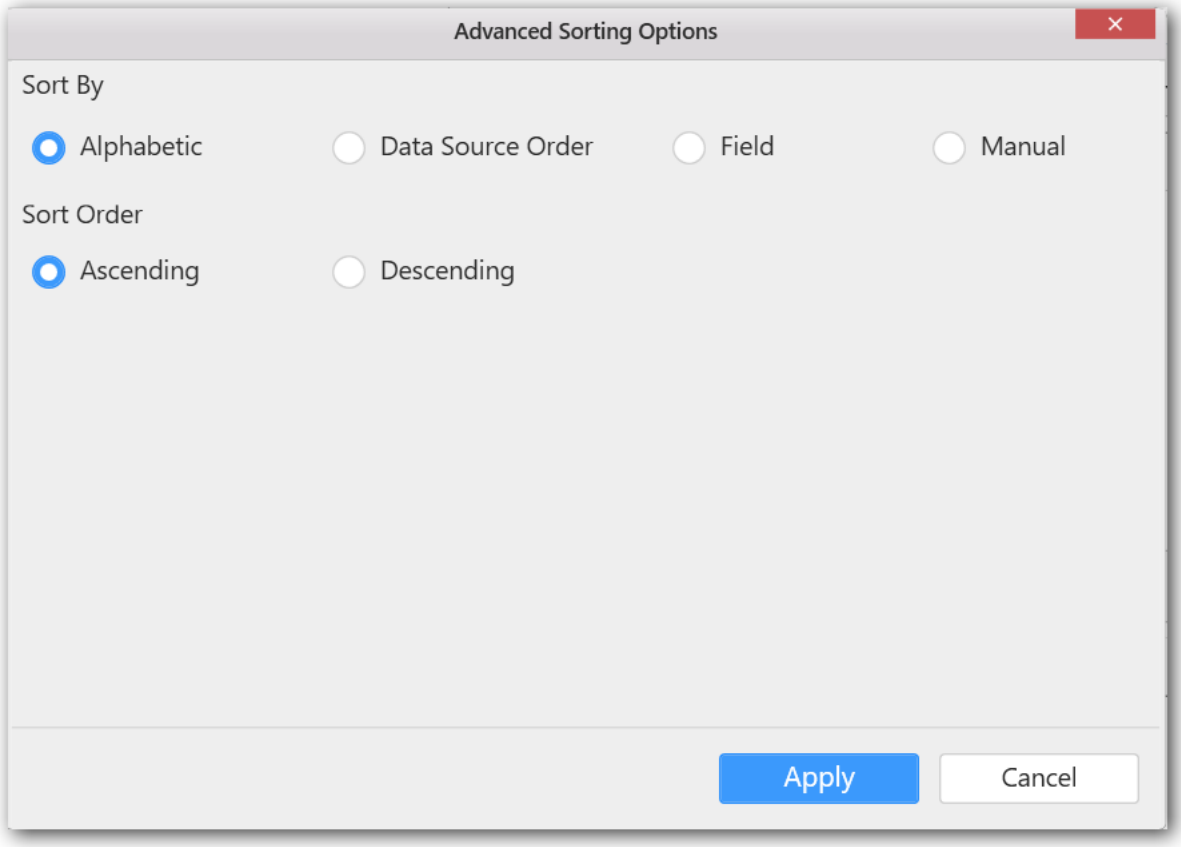
Bind column through drag and drop element from sections to Display Column section.



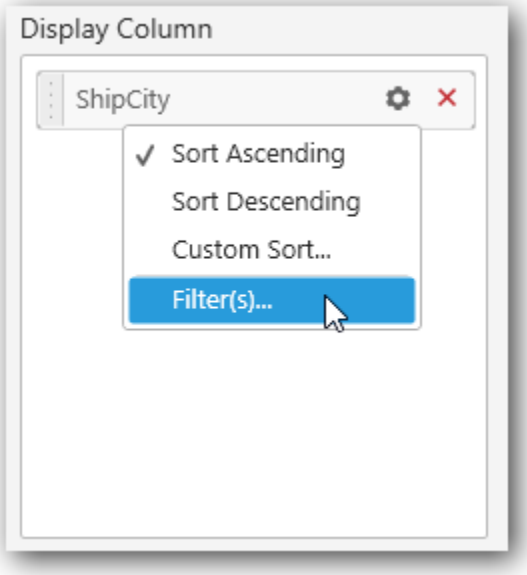
You can select the settings to sort the data either Ascending or Descending.



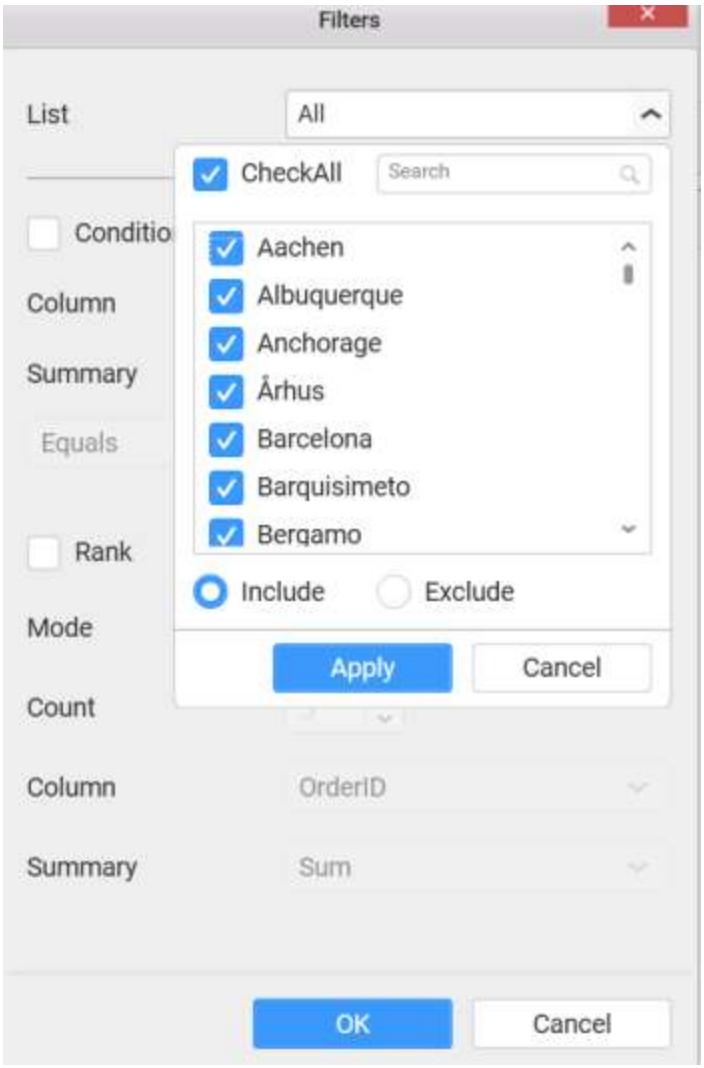
Custom sort



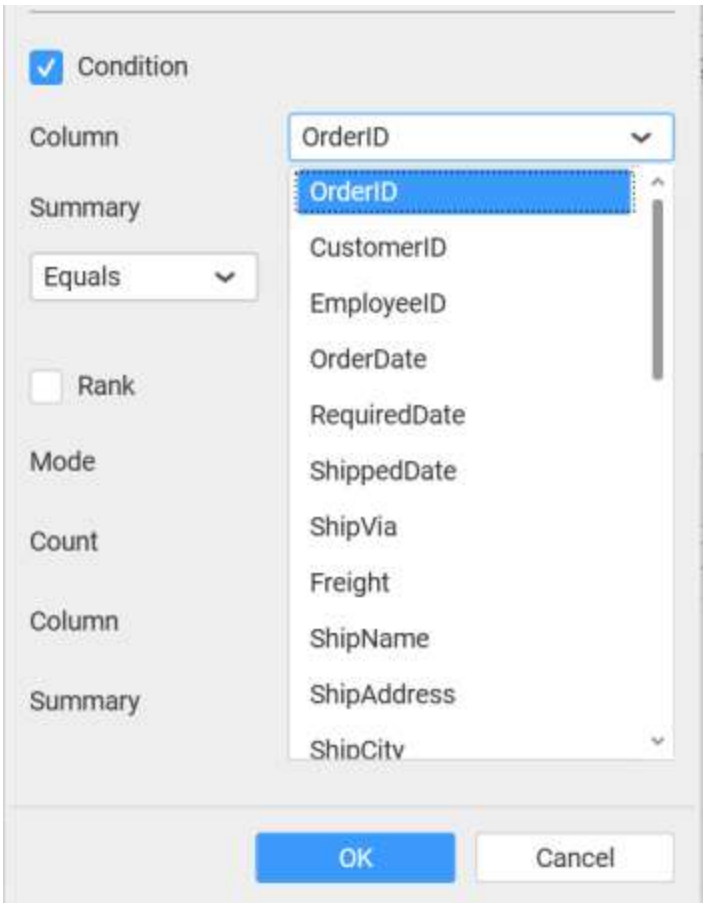
You can use the filters by selecting the Filter(s)... option to rank the elements.

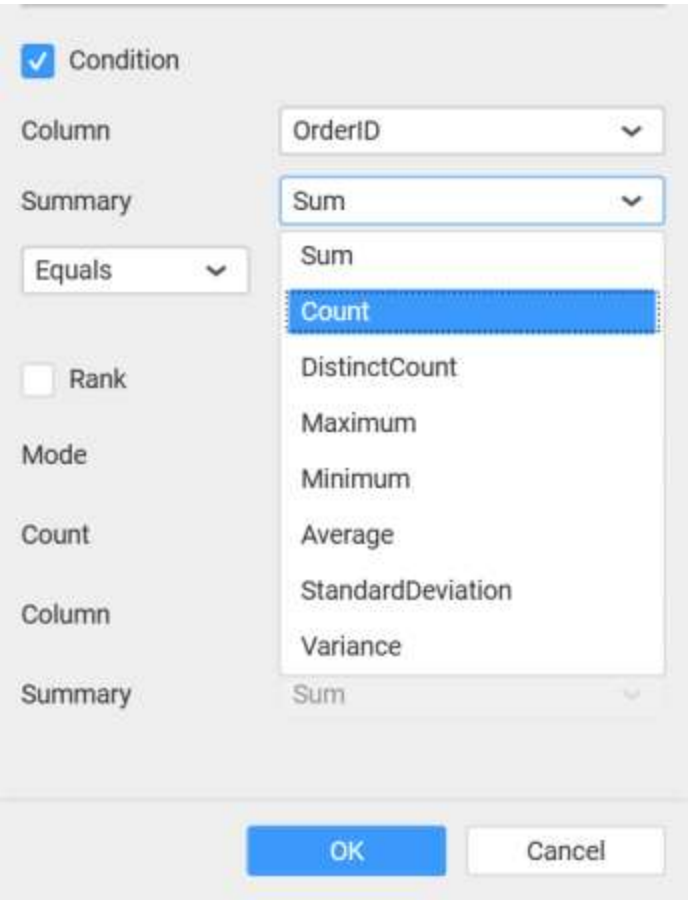


You can select the specific city to filter the element and CheckAll is used either to check all the data or to select the specific data. Include and Exclude is used to include and exclude the elements by selecting the radio button and click the Apply button.

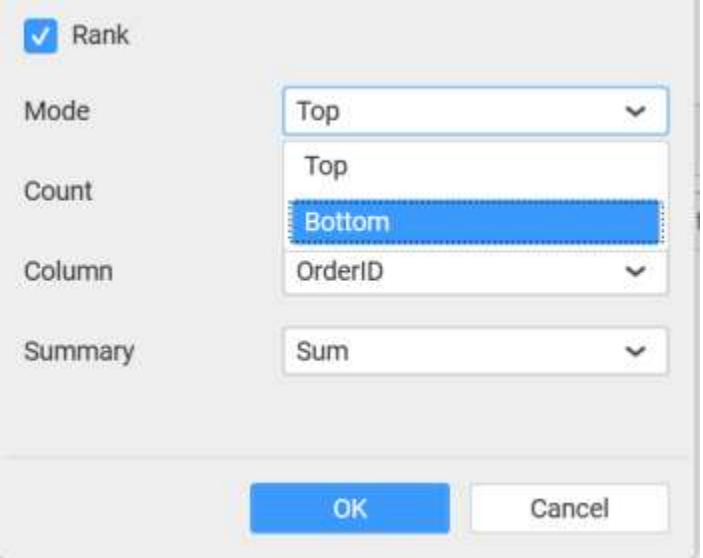


You can select the Condition option to change the Column elements and Summary type by selecting the required column name and summary type.

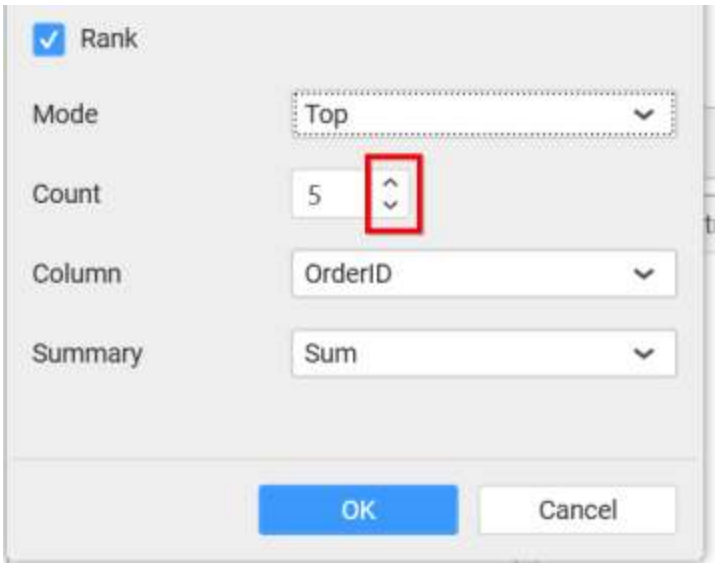




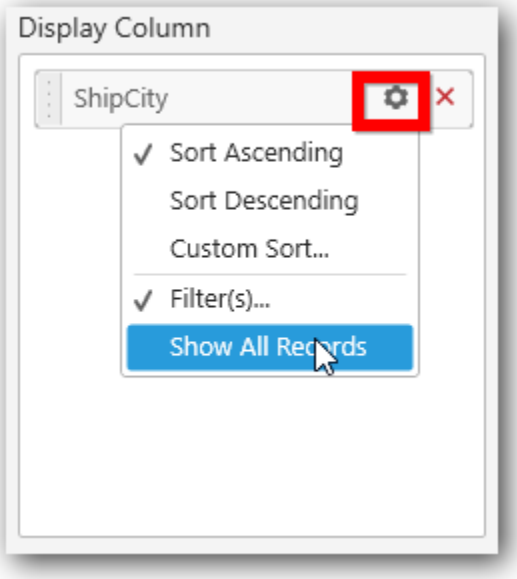
You can select the Rank option to enable filters and select the Mode either top or bottom.



You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.



You can clear the filters by selecting the Show All Records option.



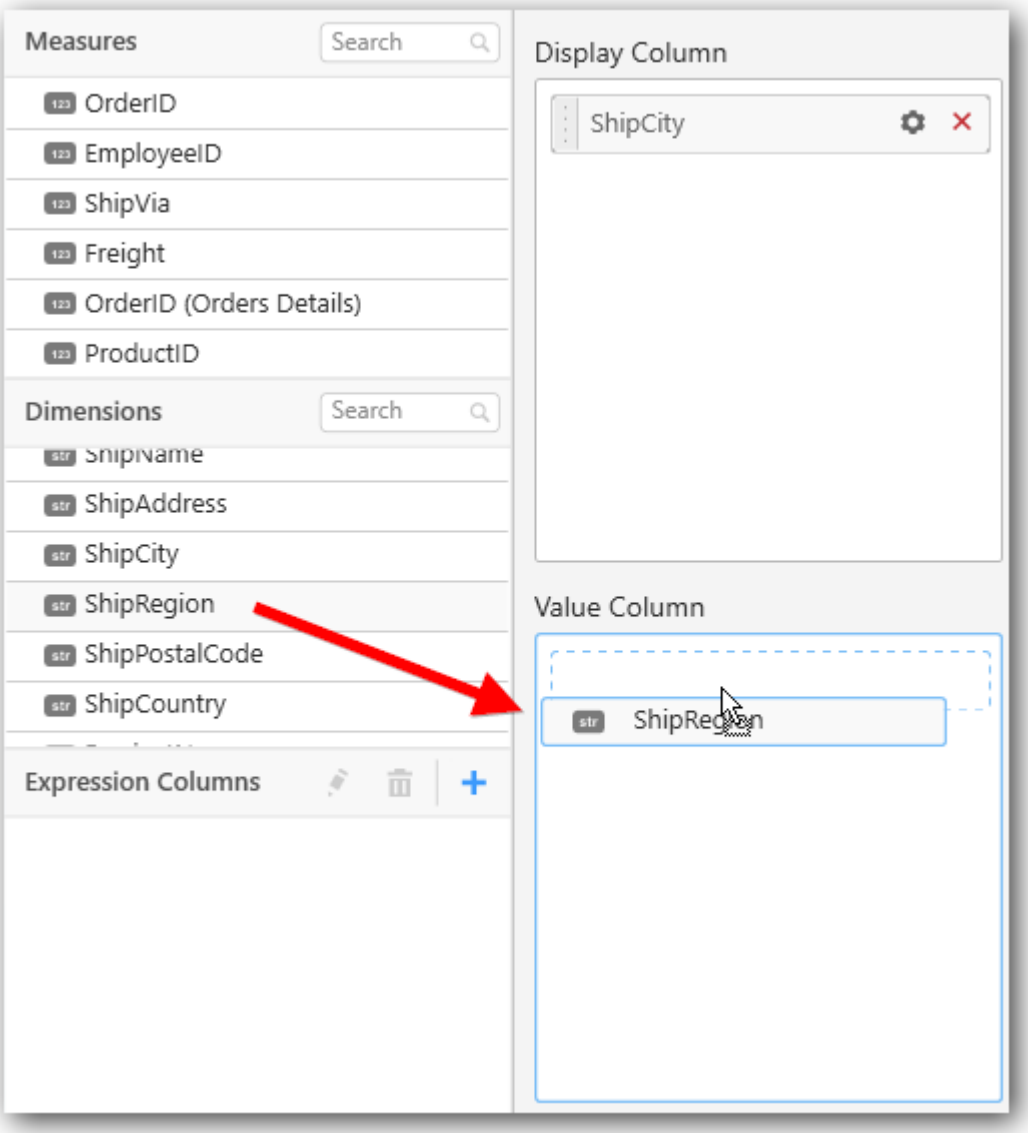
Value Column

We can drag and drop a column from Measures or Dimensions or Expression Columns category to Value Columns section.

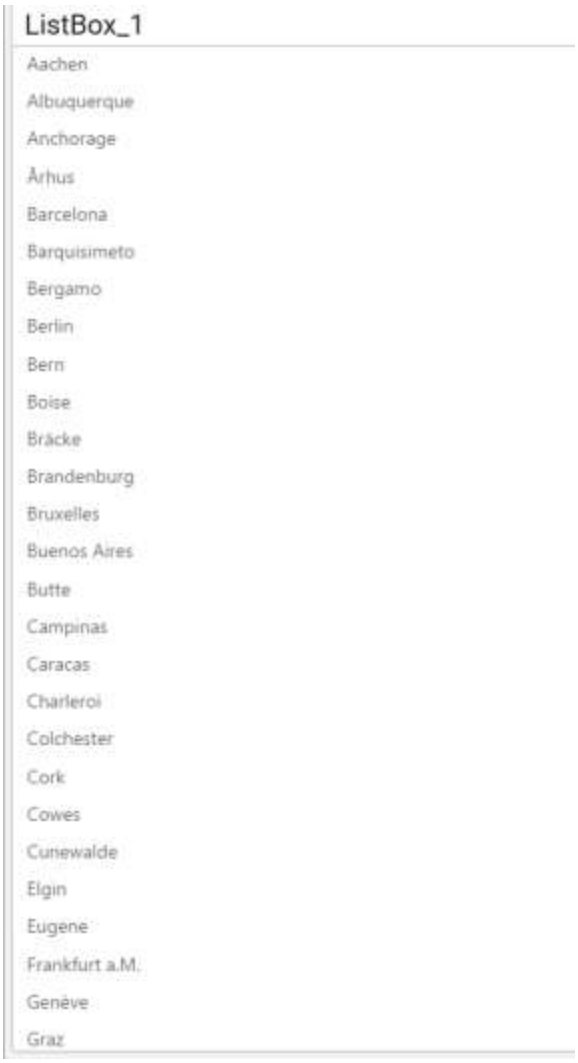
Note: This column section value must be unique.

Display column values will be shown visually in the widget but internal operation based on value column.

Value column will not be applicable for custom format and relative date format.



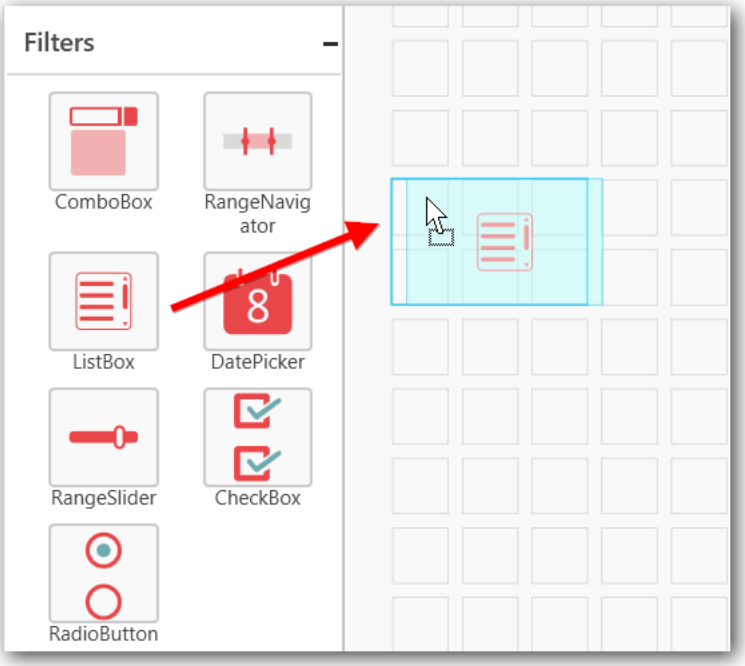
Here is an illustration,



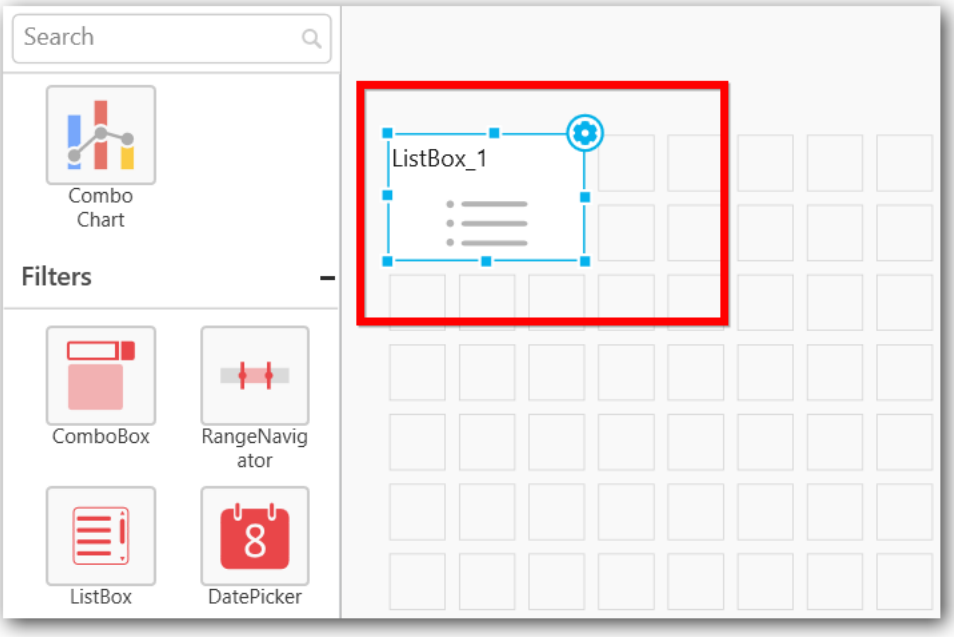
How to configure the SSAS data to ListBox?

Following steps illustrates configuration of SSAS data to List Box.

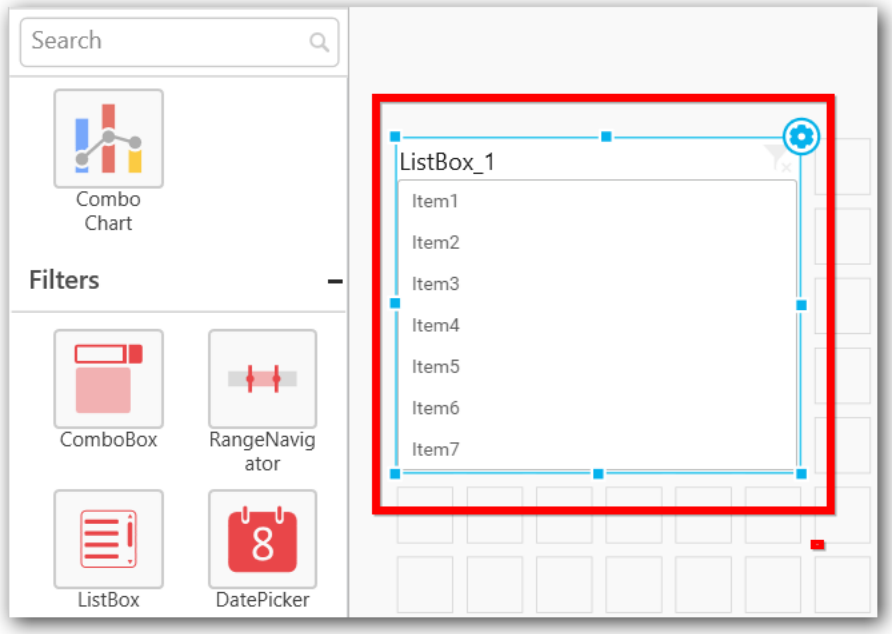
Drag and drop ListBox control icon from the Tool box into design panel. You can find control in Toolbox by search.



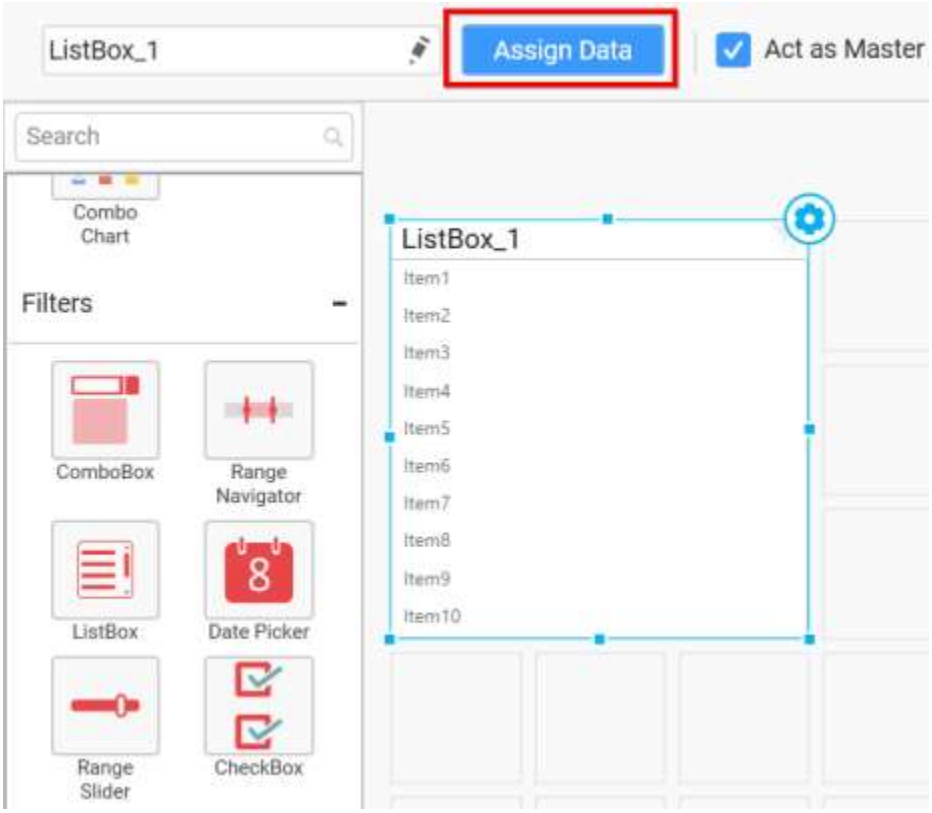
Before Resizing



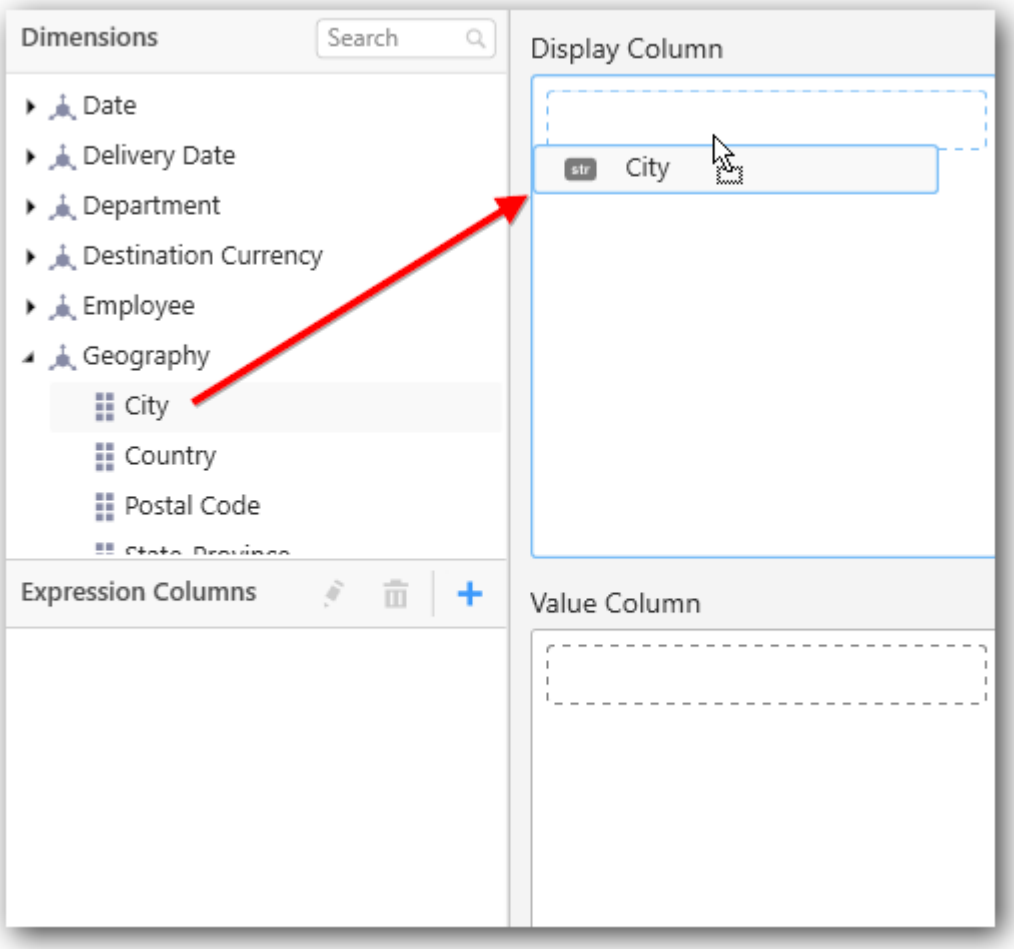
After Resizing



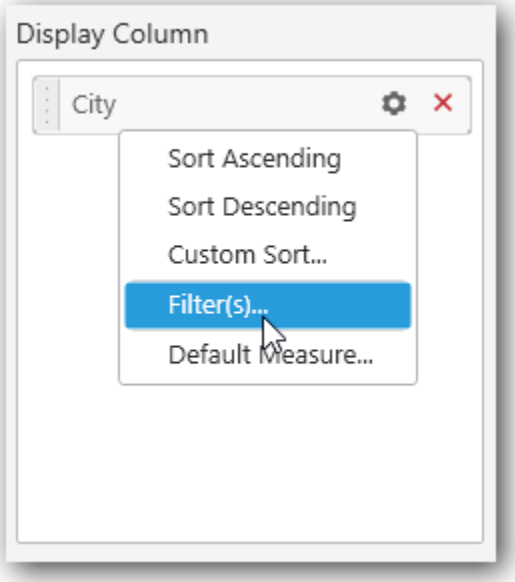
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



Drag and drop a dimension level or hierarchy under Dimensions category into Display Column.

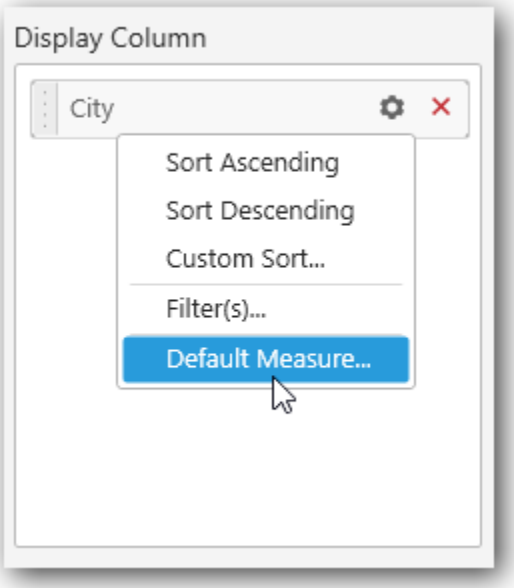


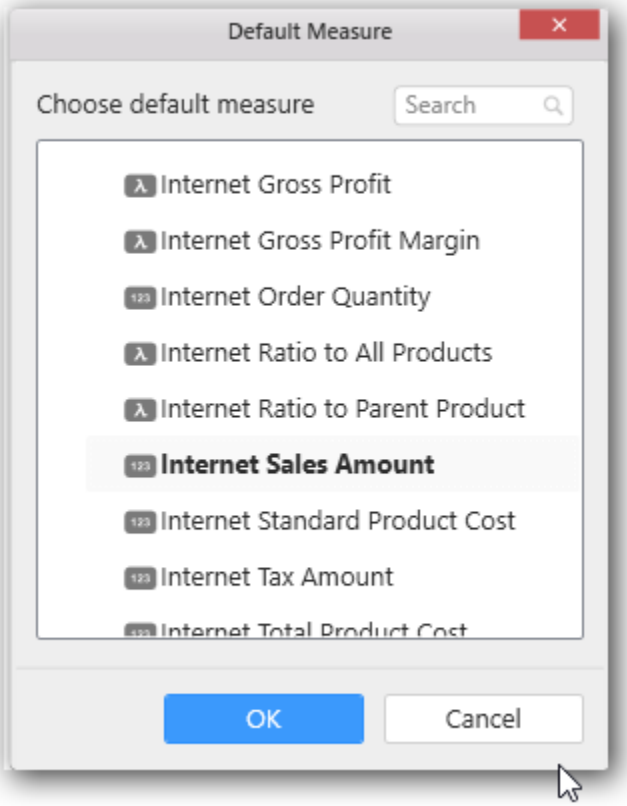
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



To know more about filters, refer here

Define default measure to the dropped dimension through Default Measure menu item to retrieve exact result for that dimension.





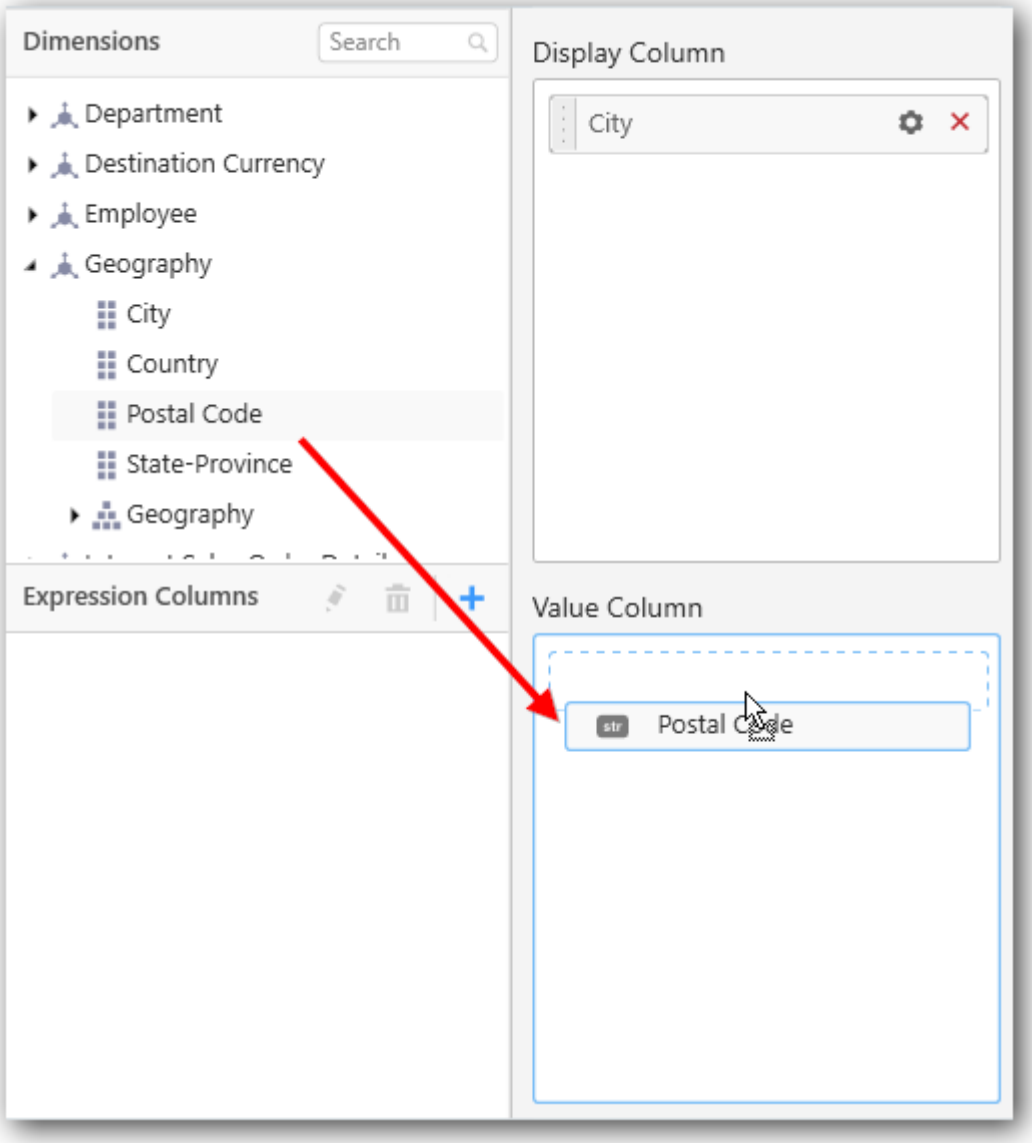
Value Column

We can drag and drop a column from Measures or Dimensions or Expression Columns category to Value Columns section.

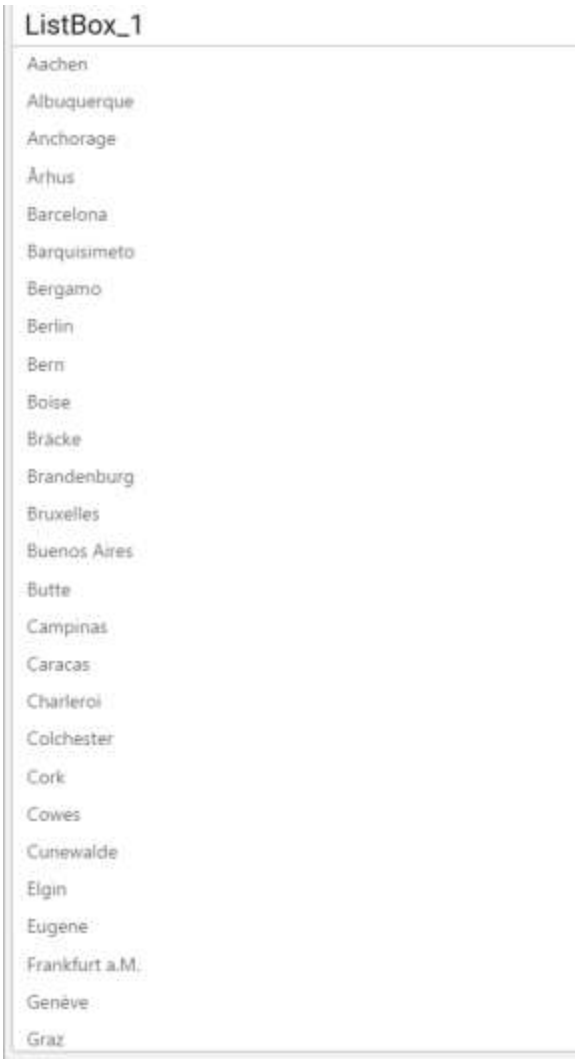
Note: This column section value must be unique.

Display column values will be shown visually in the widget but internal operation based on value column.

Value column will not be applicable for custom format and relative date format.



Here is an illustration,



How to format ListBox?

You can format the List box for better illustration of the view that you require, through the settings available in Properties pane. This pane can be opened from design view through clicking the Settings icon at top right corner of the widget.

General Settings

Header

This allows you to set title for this list box widget.

SubHeading

This allows you to set sub-title for this list box widget.

Description

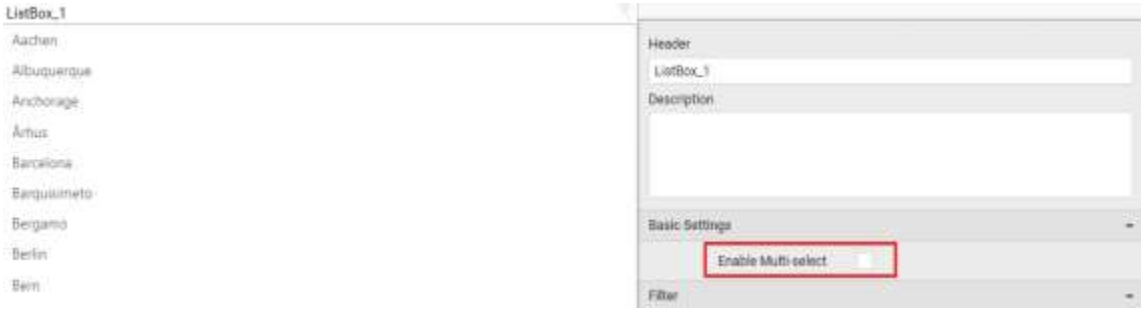
This allows you to set description for this list box widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

Enable Multi-select

This allows you to define single/multiple item selection in List Box.

Single Selection



Multiple Selection

Filter Settings

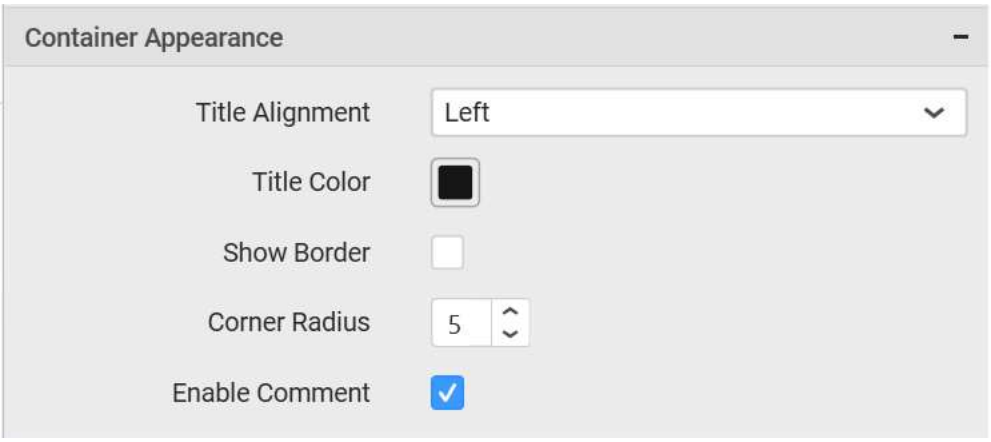
Act as Master Widget

This allows you to define this list box widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this list box widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

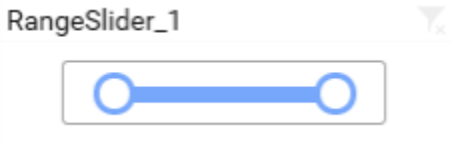
This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Range Slider

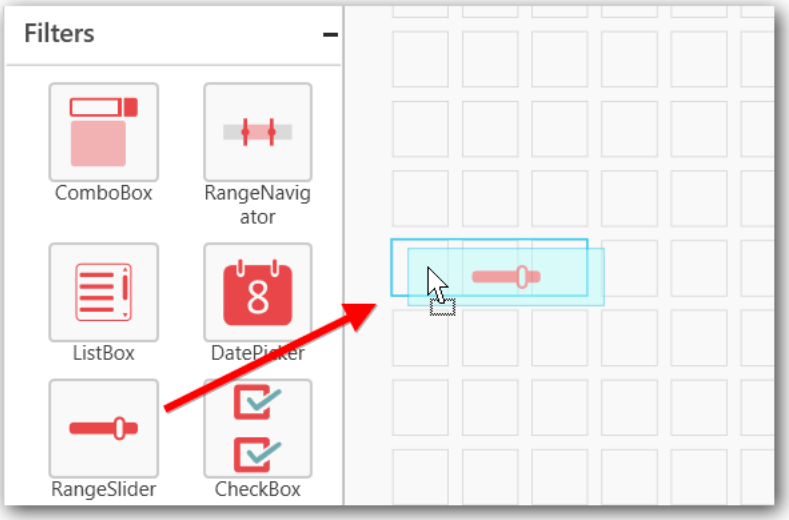
Range Slider enables you to filter based on value or date range set through sliders. To configure a range slider, a minimum requirement of 1 column is needed.



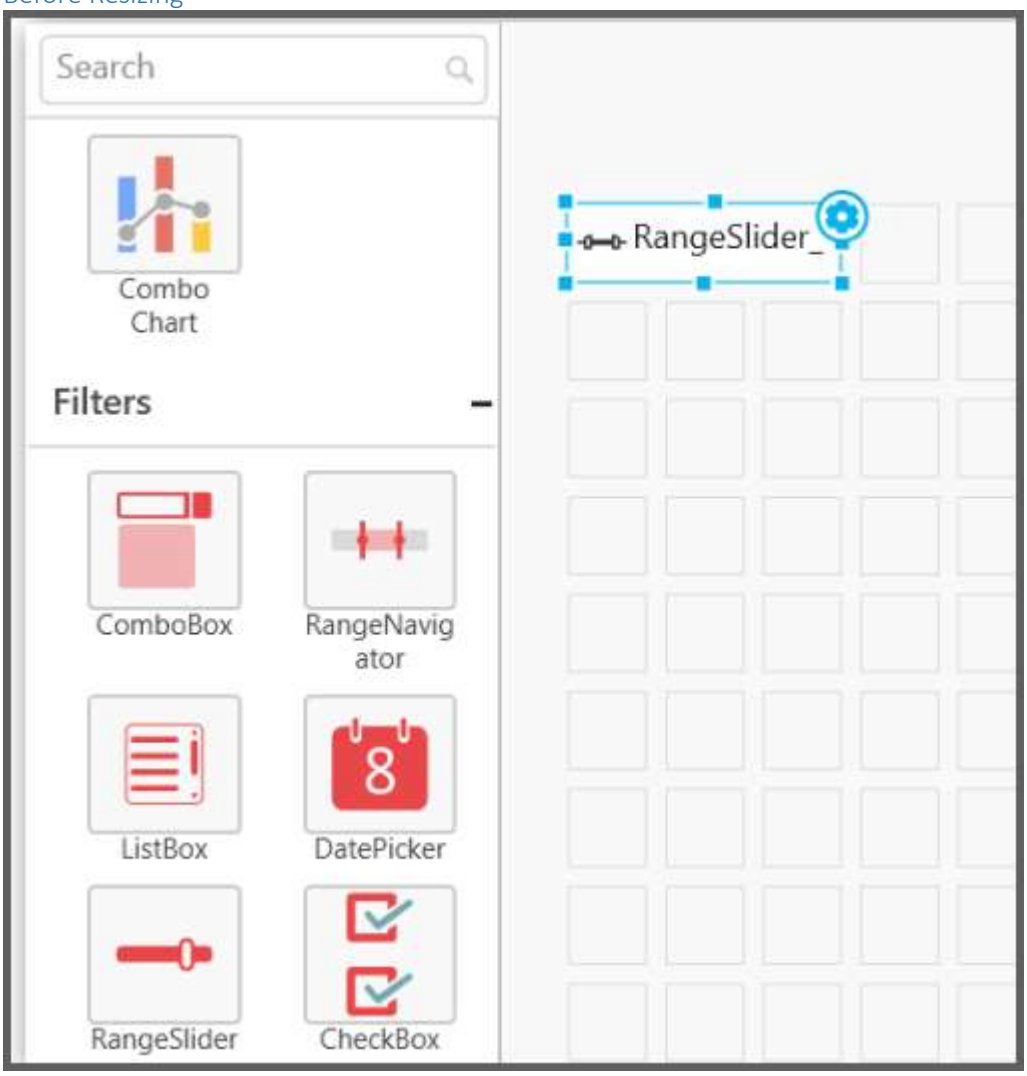
How to configure the flat table data to Range Slider?

The following procedure illustrates data configuration of Range Slider.

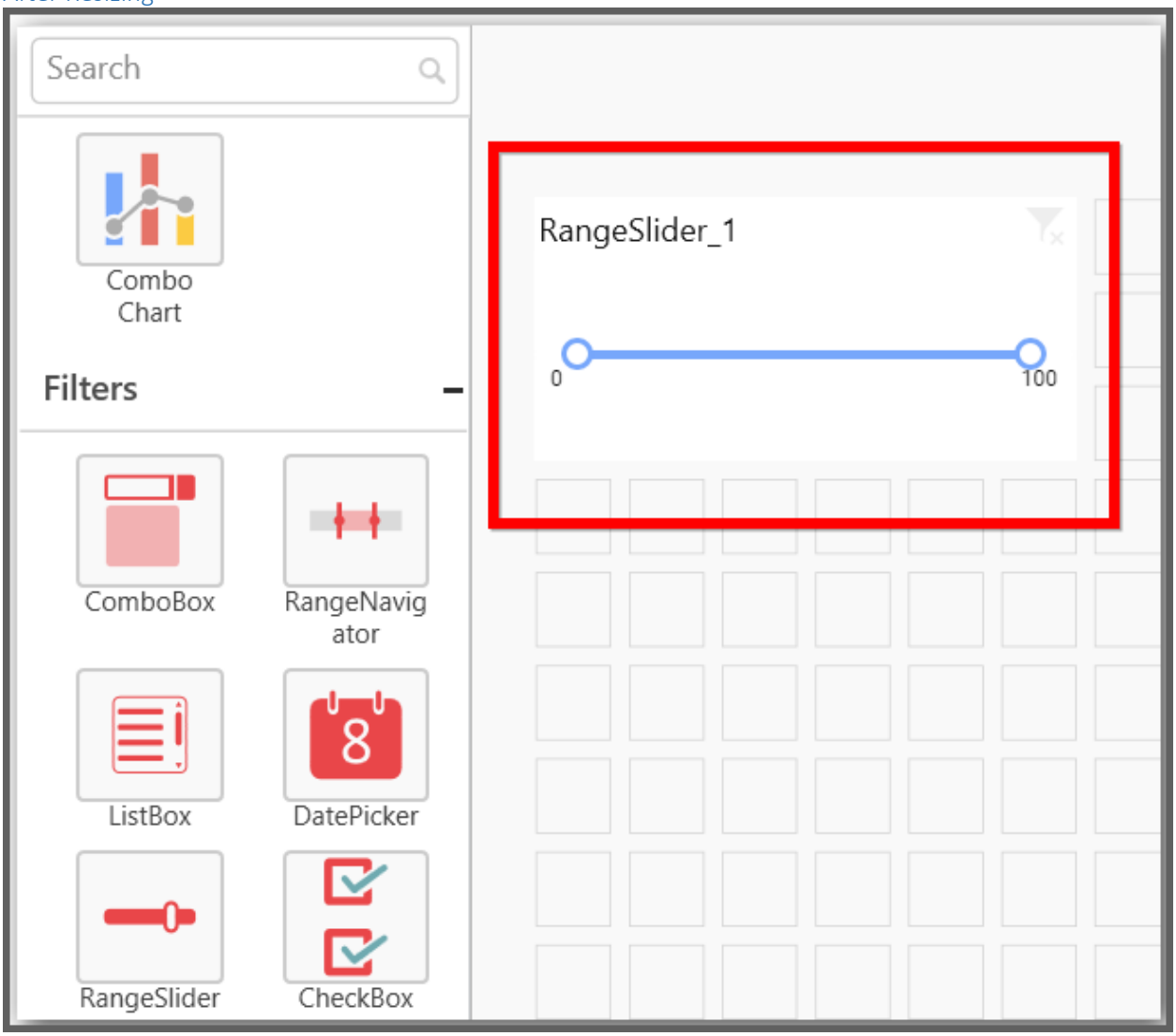
Drag and drop Range Slider control icon from the Tool box into design panel. You can find control in Toolbox by search.



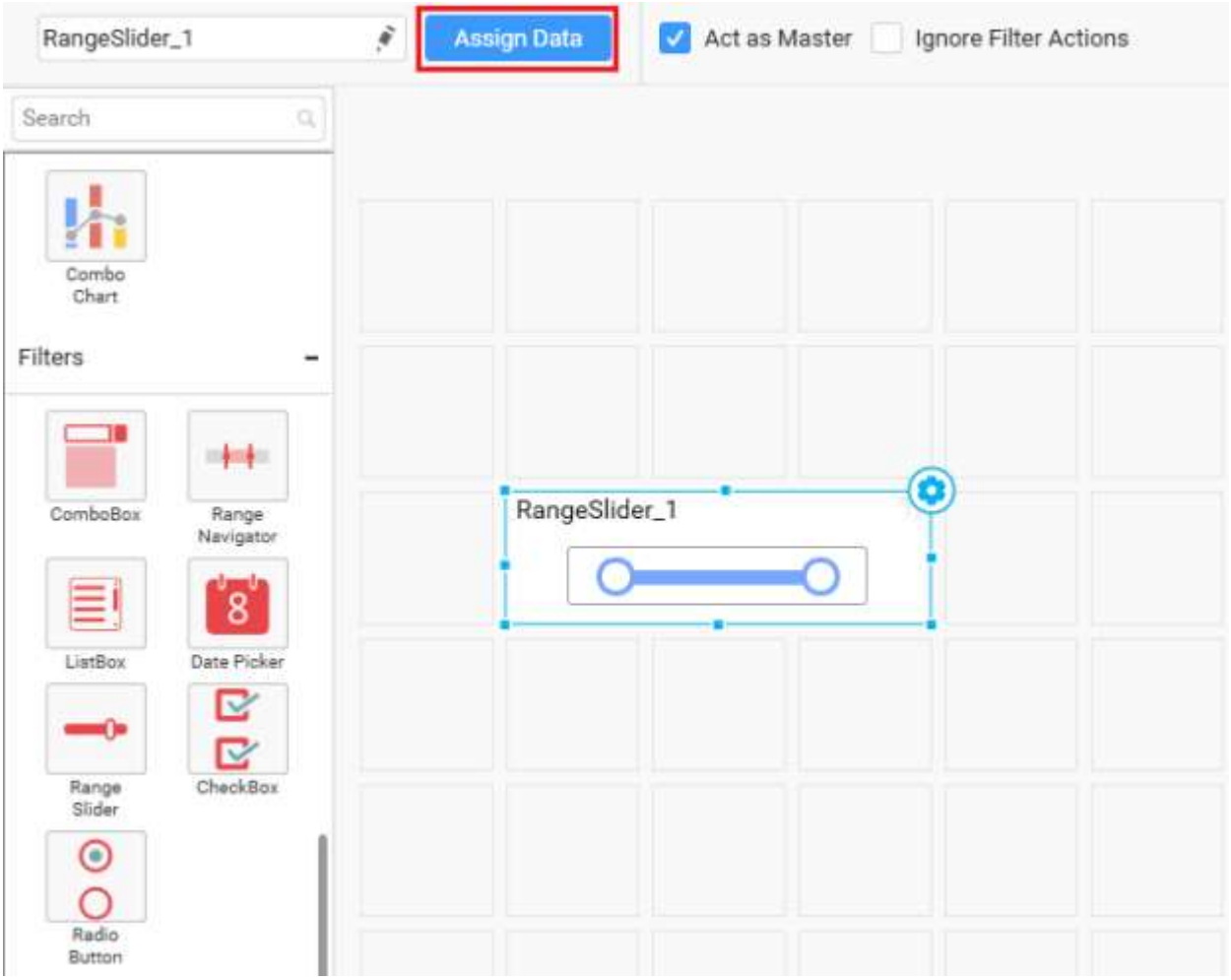
Before Resizing



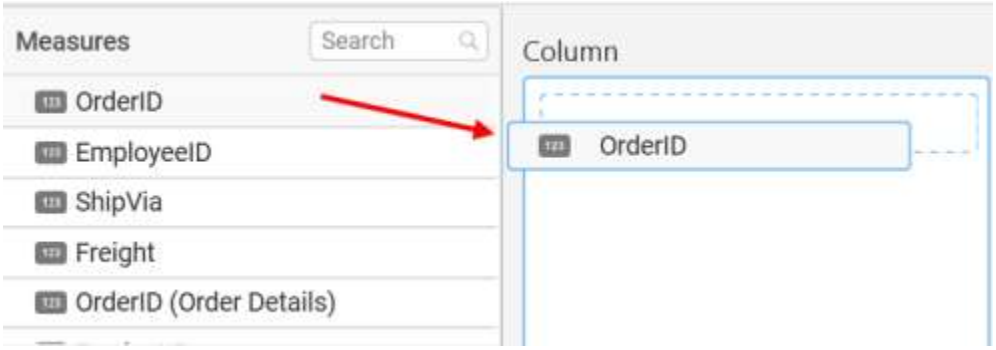
After Resizing



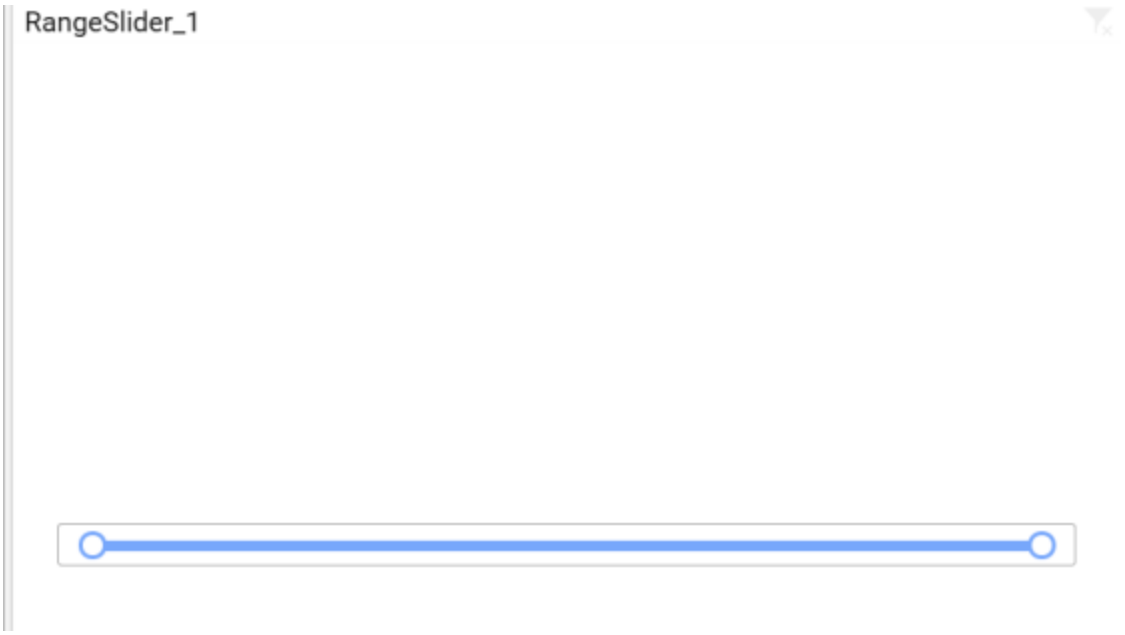
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



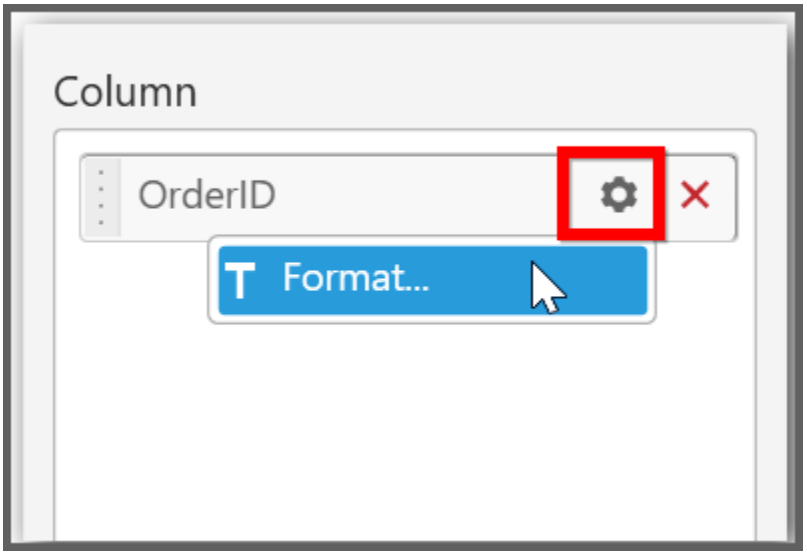
Bind column through drag and drop element from sections to Column section.



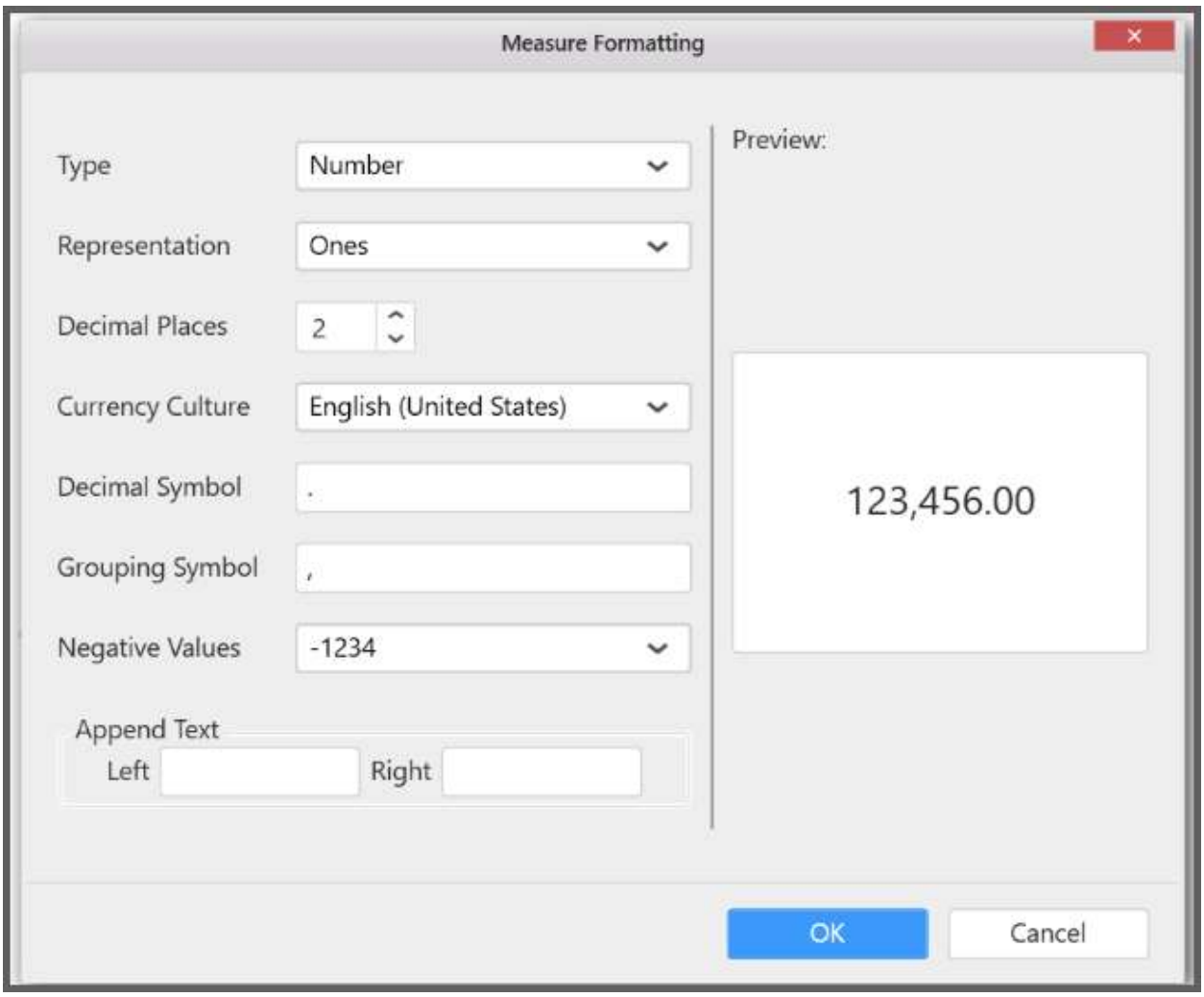
Here is an illustration,



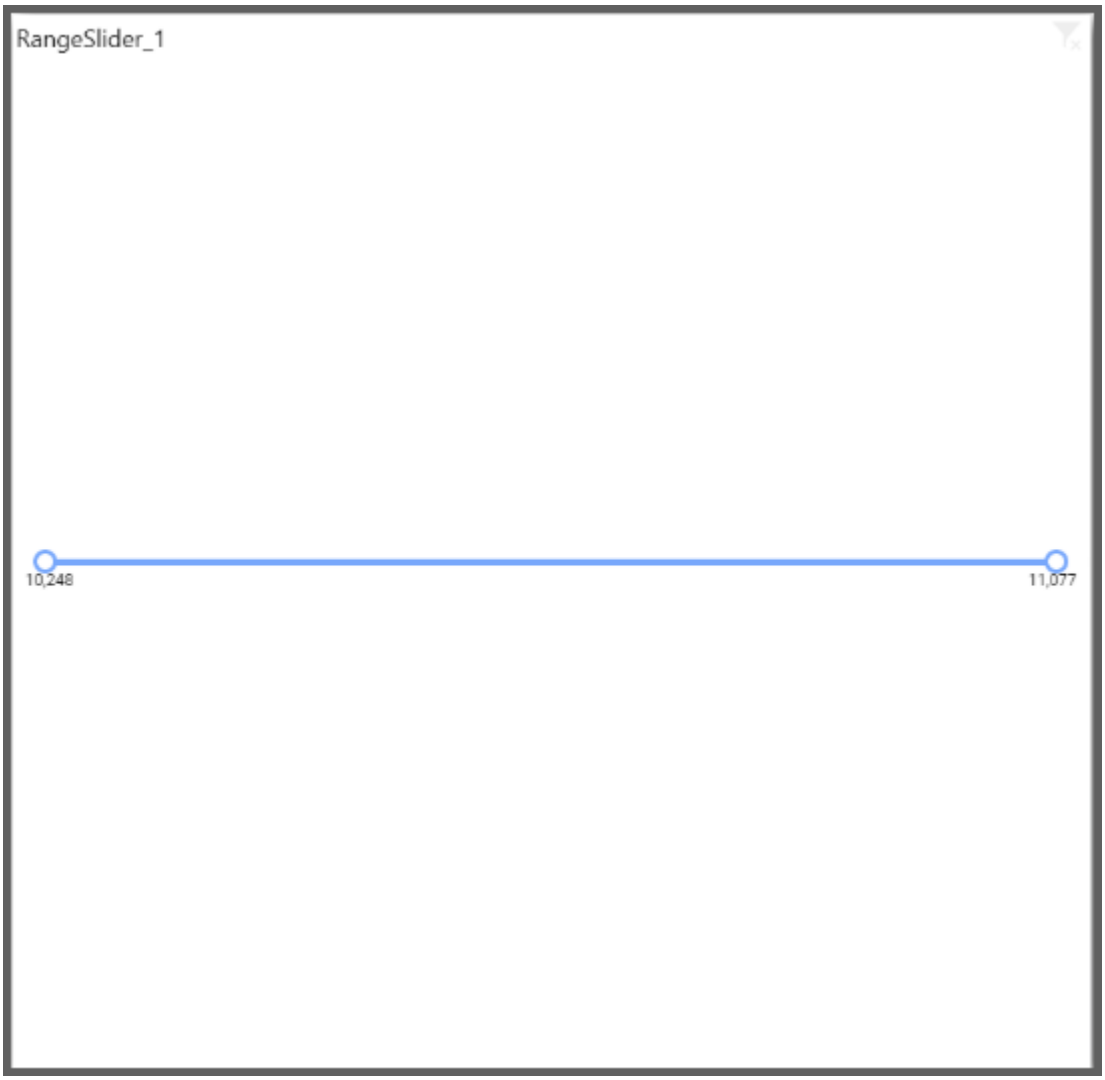
Select Format option to define the display format to the values in the column through Measure Formatting window.



The Measure Formatting window will open, you can select the values you want and click ok

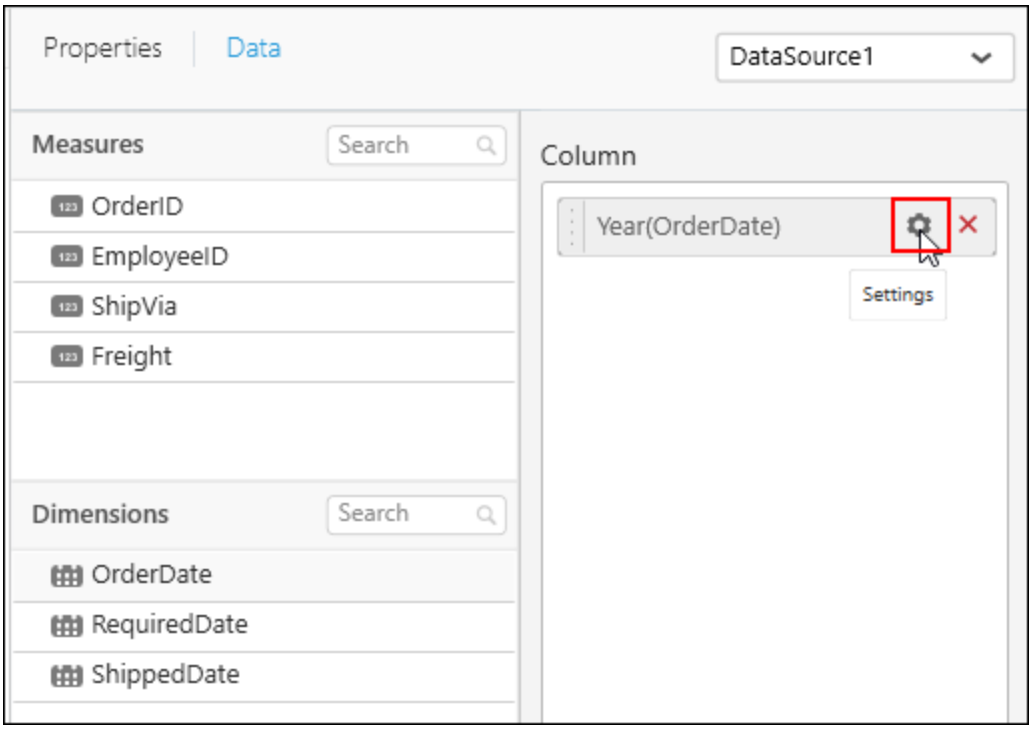


Here is an illustration,

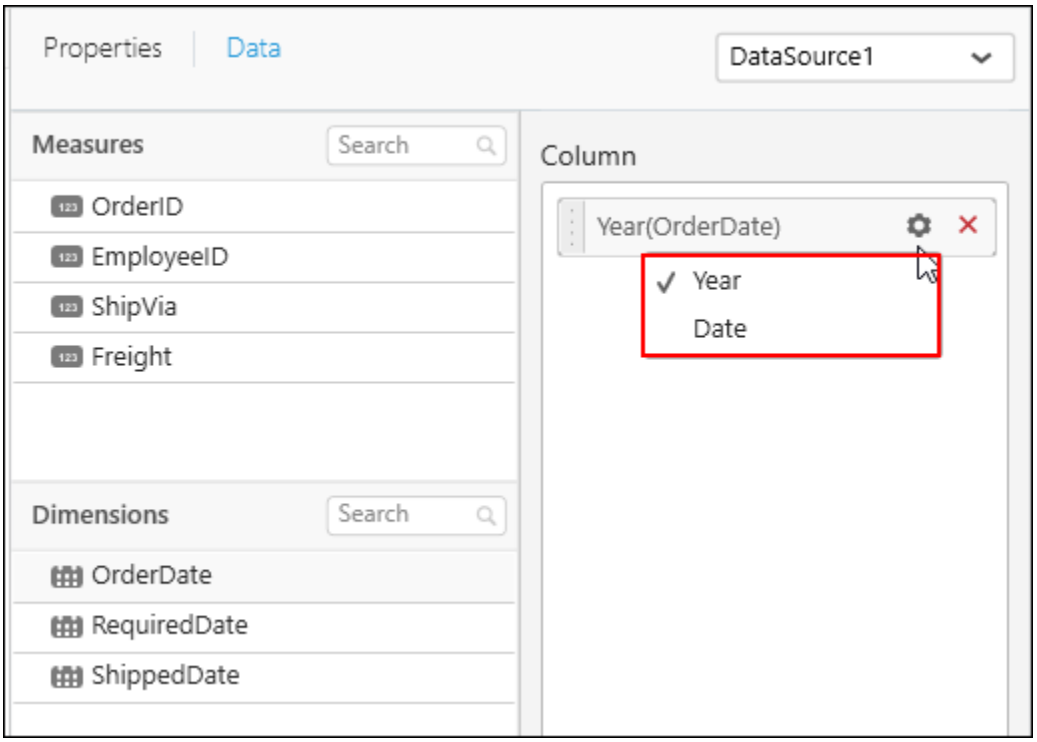


Configuring Date/Year settings in Range Slider

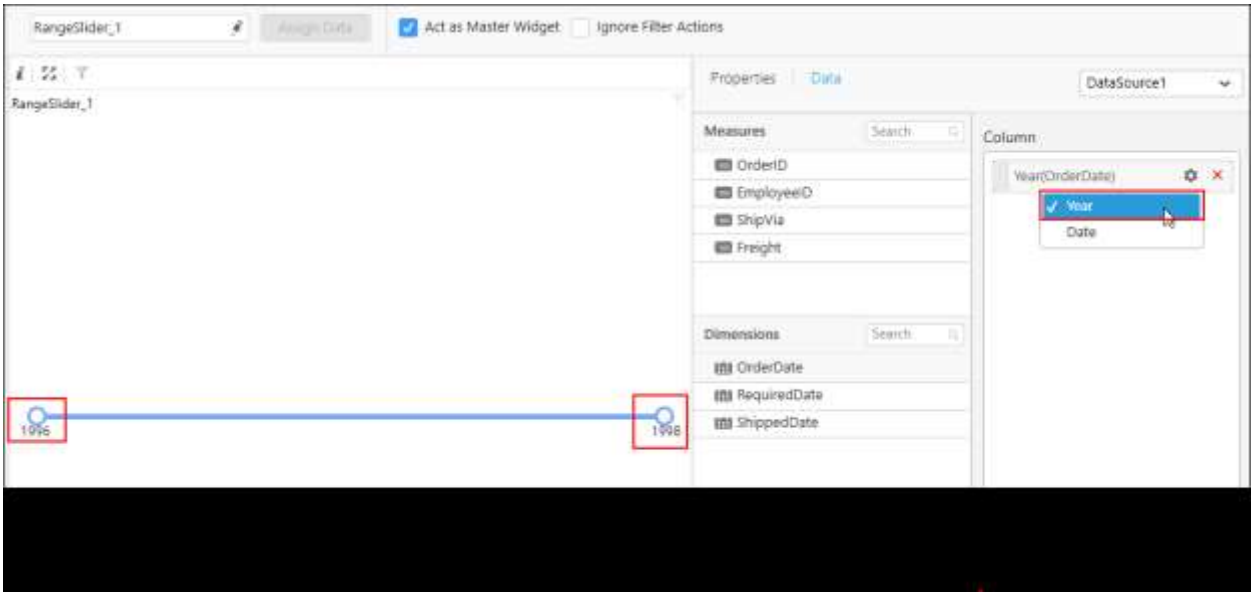
Drag and Drop a Dimension column into Column(s) section.



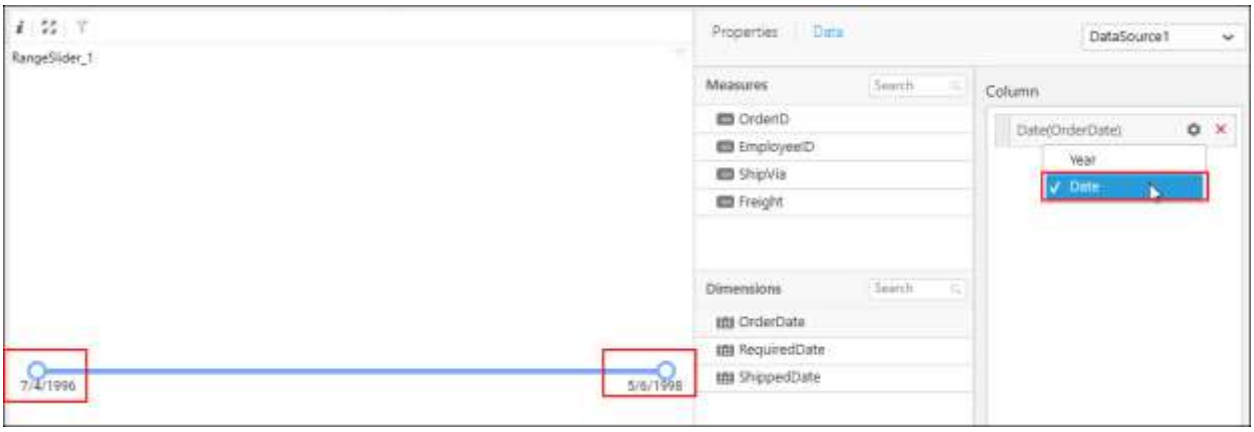
Click on the column field Settings to view the Year Setting as well as Date setting for the dimension column.



Here is an illustration of selecting Year field. You can filter the data according to year.



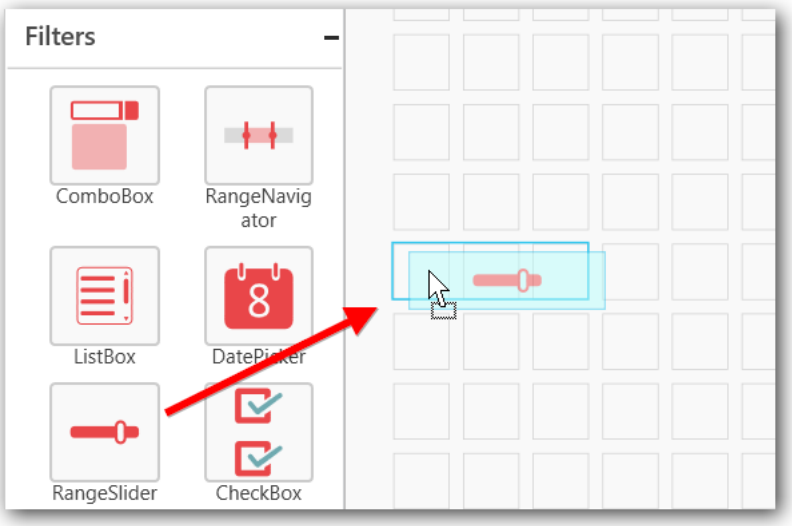
Here is an illustration of selecting Date field. You can filter the data depends upon the date.



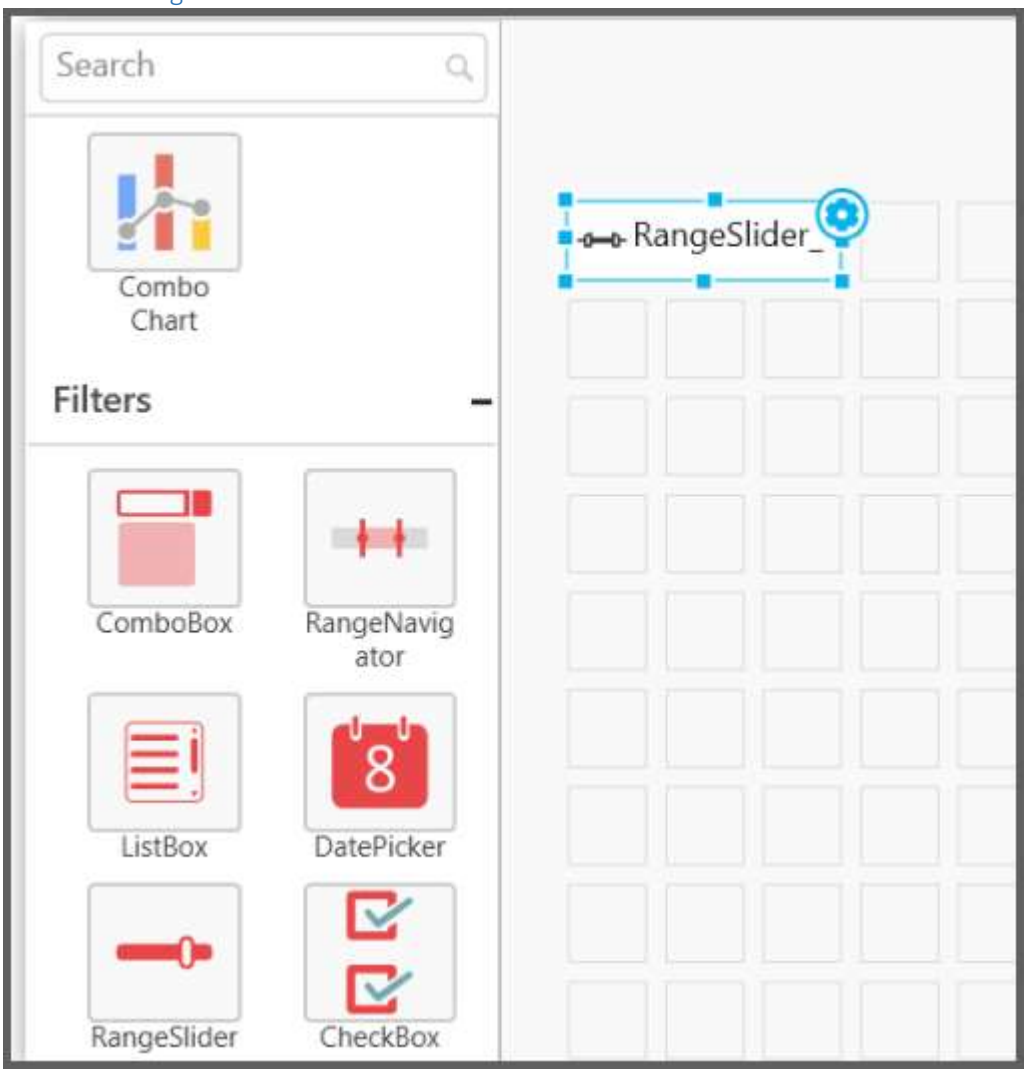
How to configure the SSAS data to Range Slider?

Following steps illustrates configuration of SSAS data to Range Slider.

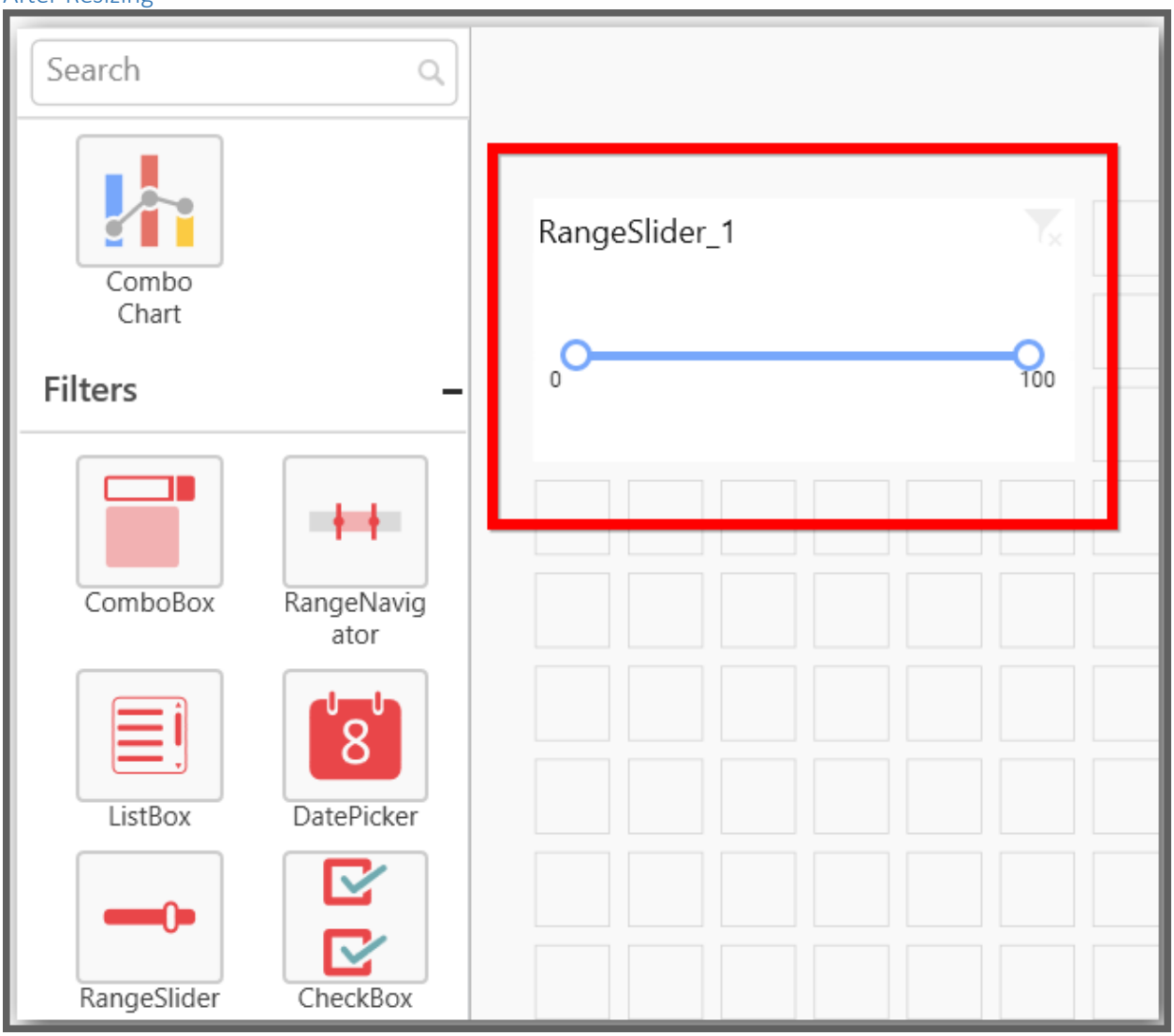
Drag and drop the Range Slider widget into canvas and resize into your required size.



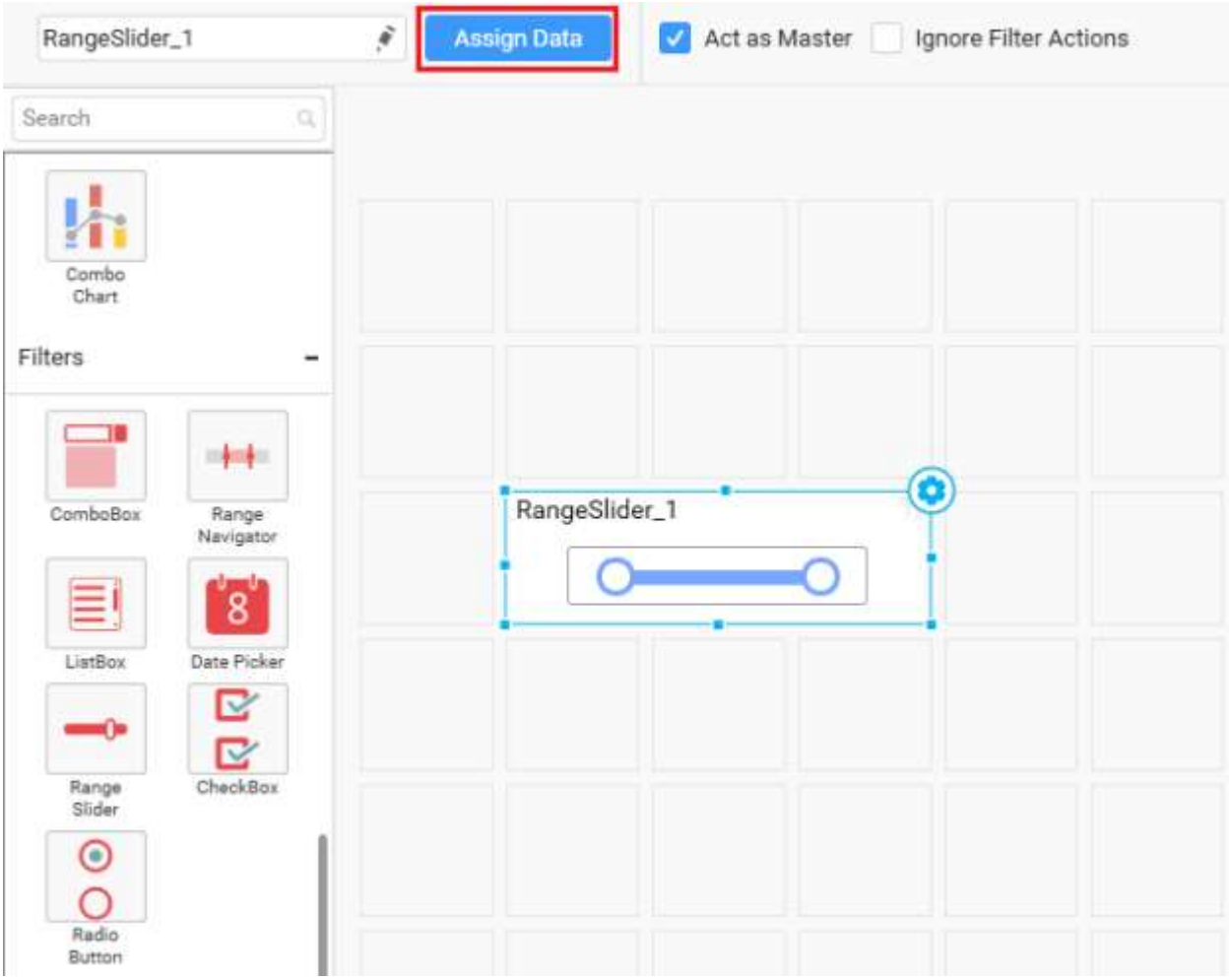
Before Resizing



After Resizing

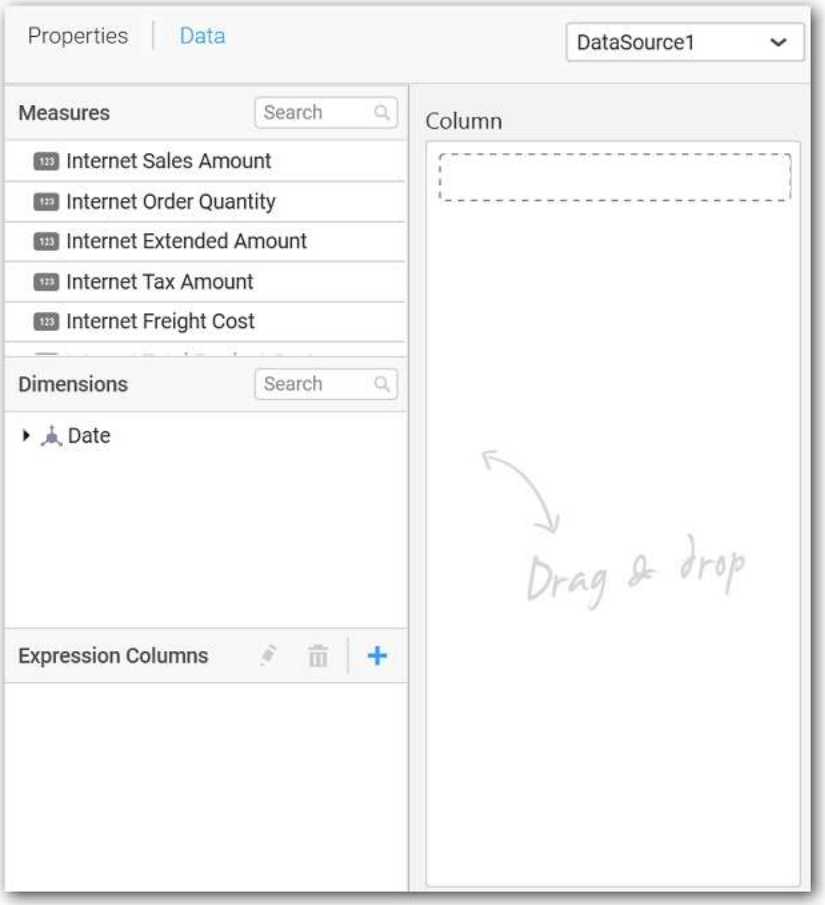


Select the dropped widget using mouse.

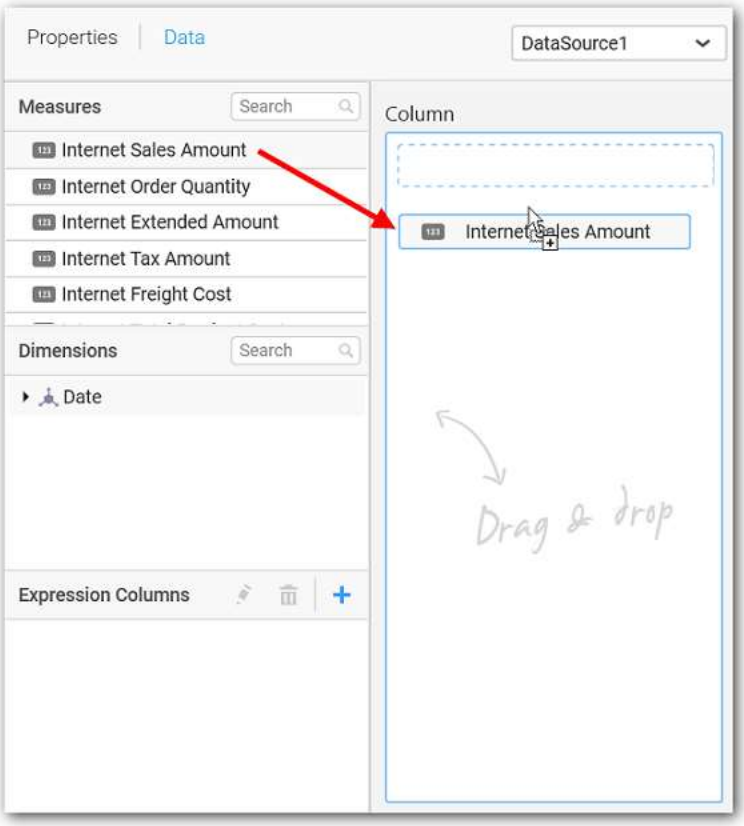


Click the Assign Data button in the toolbar.

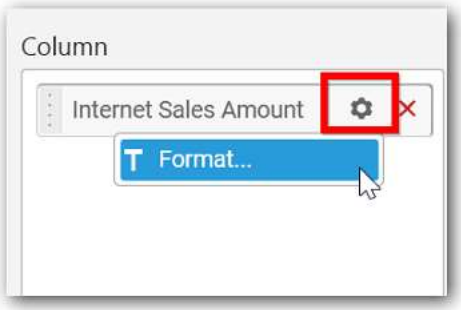
A Data pane will be opened with available Measures and Dimensions.



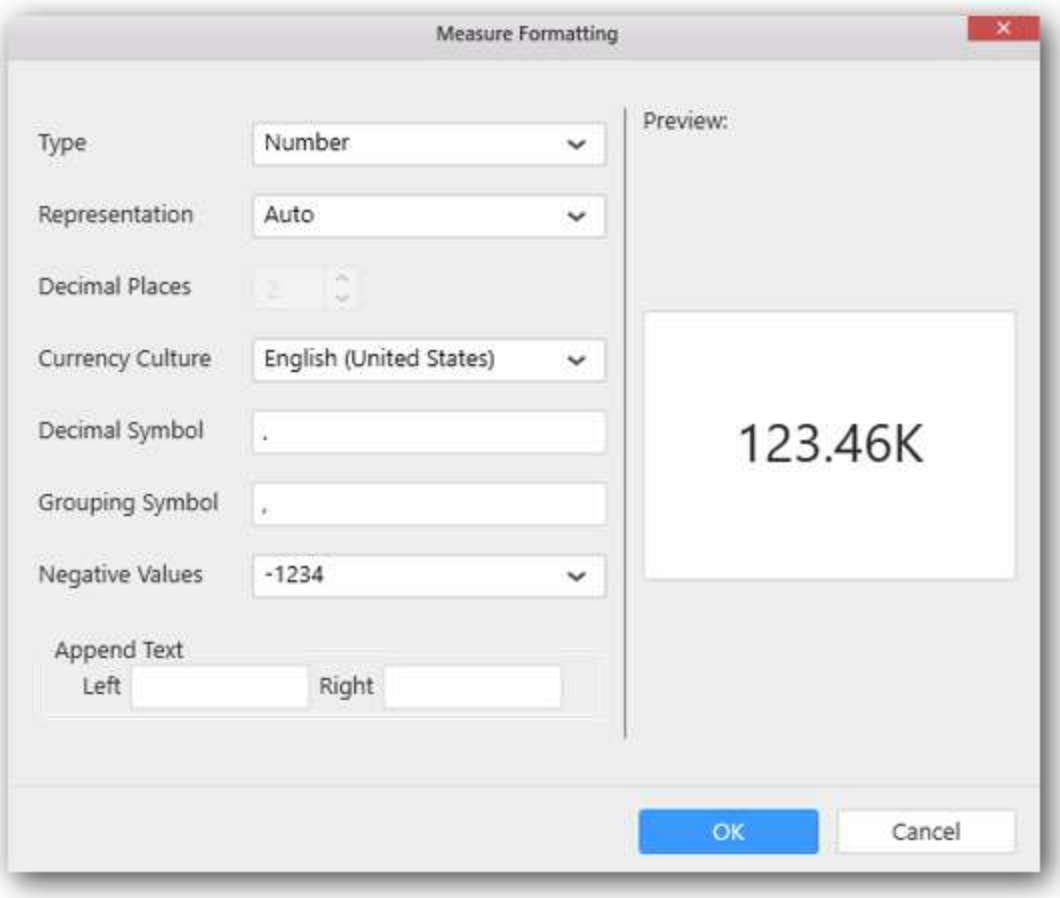
Drag and drop a column into Column(s) section.



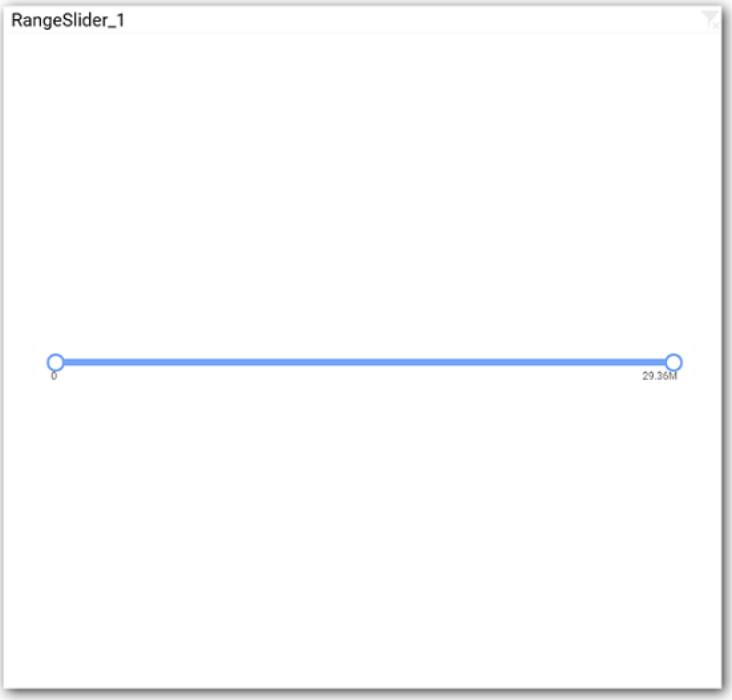
Select Format option to define the display format to the values in the column through Measure Formatting window.



The Measure Formatting window will open, you can select the values you want and click ok

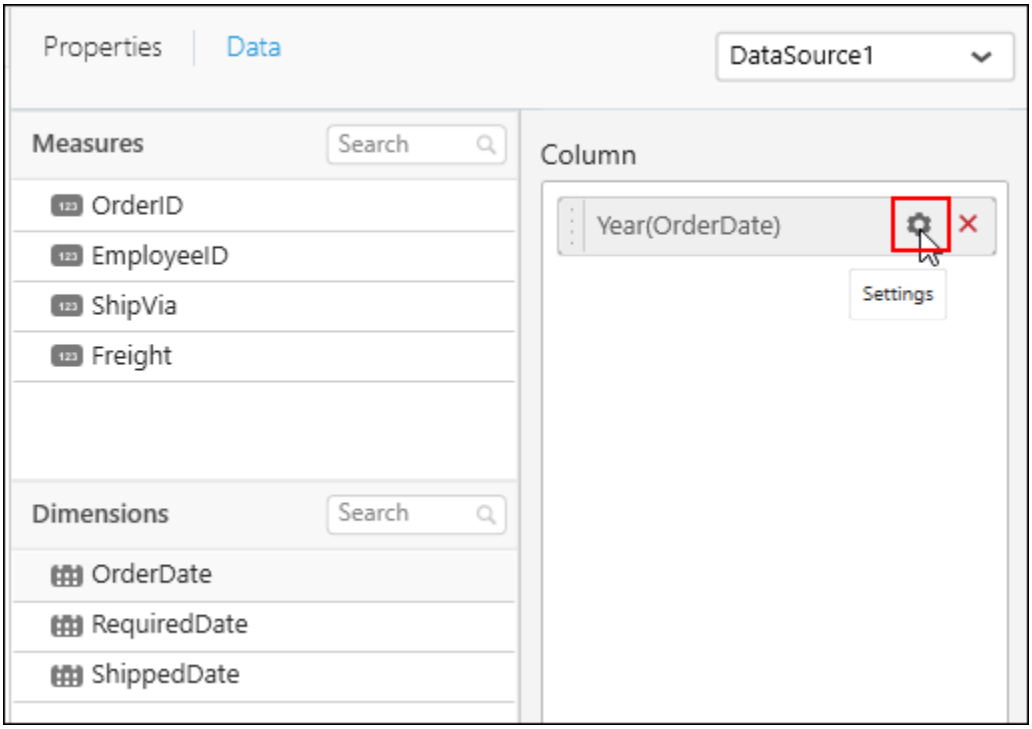


Here is an illustration,

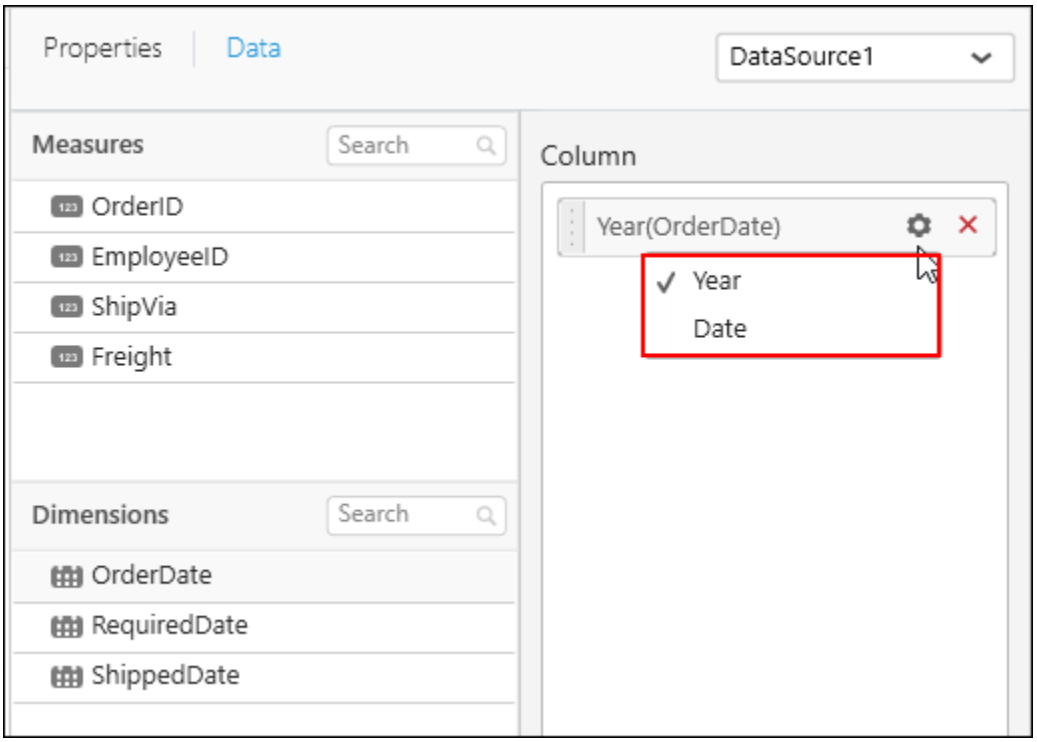


Configuring Date/Year settings in Range Slider

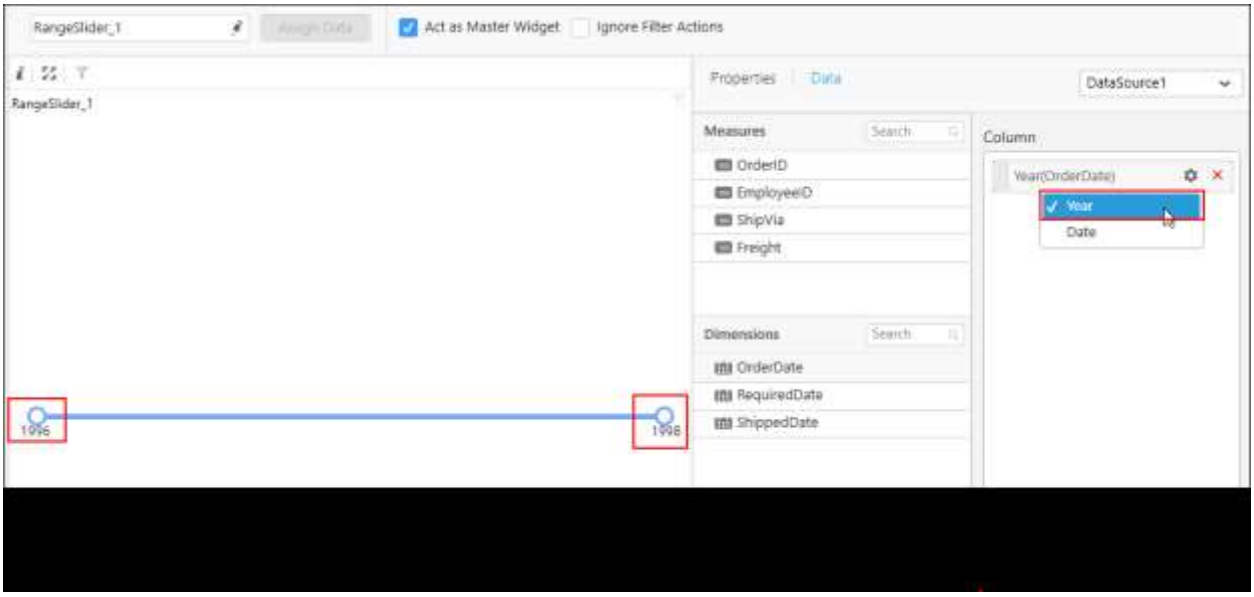
Drag and Drop a Dimension column into Column(s) section.



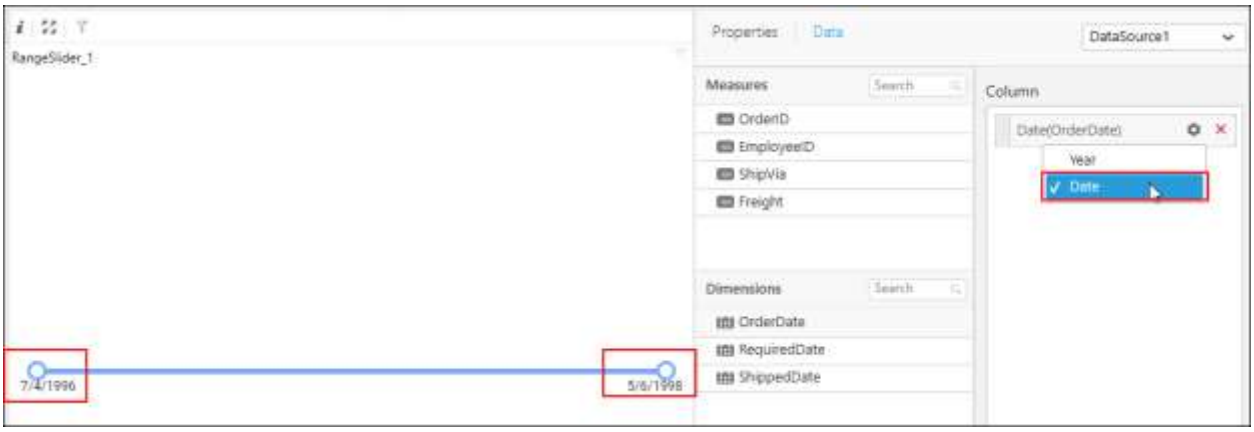
Click on the column field Settings to view the Year Setting as well as Date setting for the dimension column.



Here is an illustration of selecting Year field. You can filter the data according to year.



Here is an illustration of selecting Date field. You can filter the data depends upon the date.



How to format Range Slider?

You can format the Range Slider for better illustration of the view that you require, through the settings available in Properties pane.

General Settings

Header

This allows you to set title for this range slider widget.

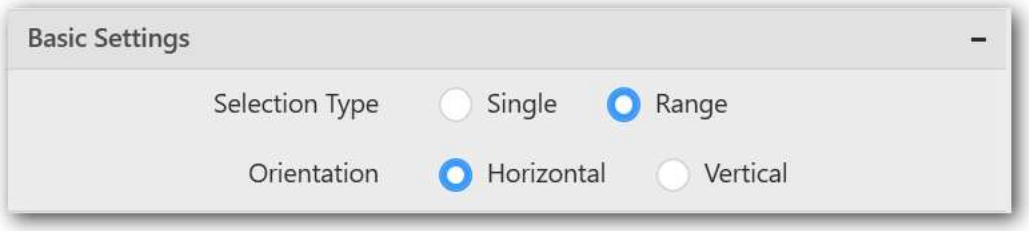
SubHeading

This allows you to set sub-title for this range slider widget.

Description

This allows you to set description for this range slider widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings



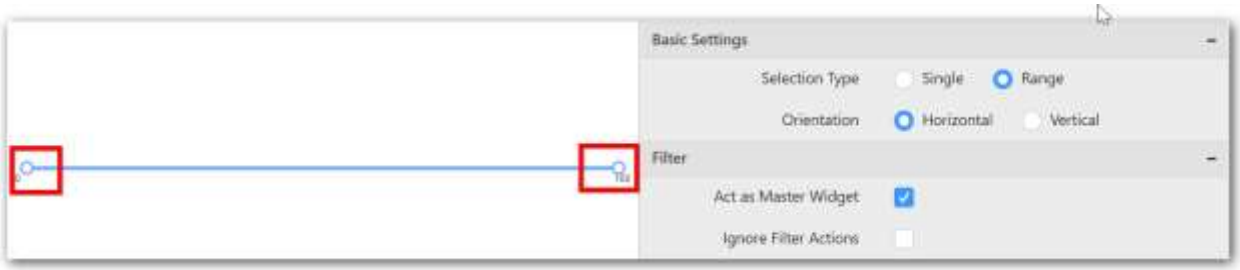
Selection Type

Single – Single value can be bounded.

Range Slider with Single Pointer

Range – A range (two values) can be bounded.

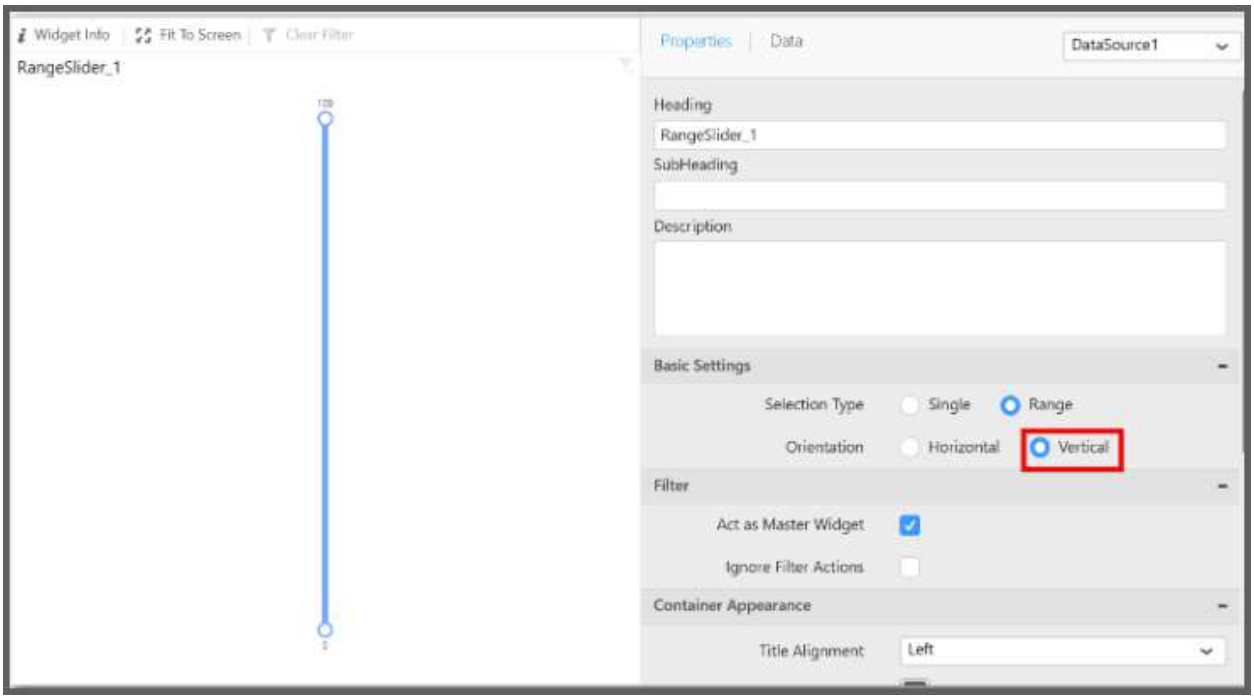
Range Slider with Range Pointer



Orientation

Orientation as Horizontal

Orientation as Vertical



Filter Settings

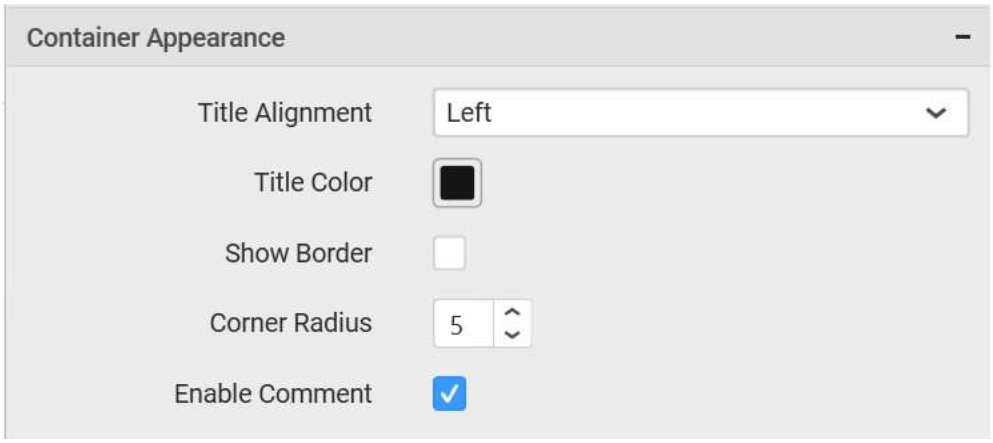
Act as Master Widget

This allows you to define this range slider widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this range slider widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

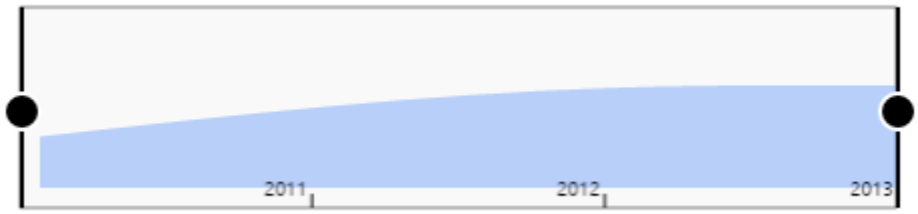
This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

Range Navigator

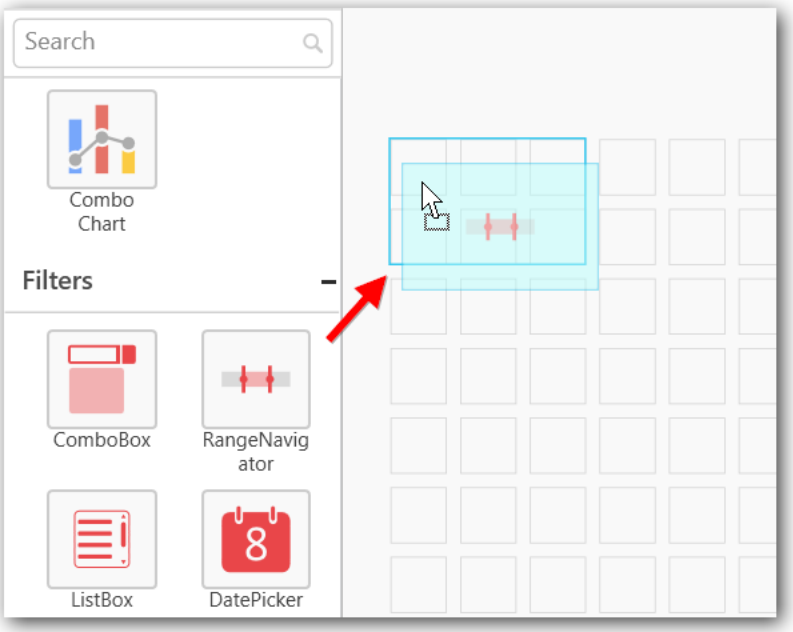
Range Navigator allows you to filter based on the date range dynamically set through navigation bars.



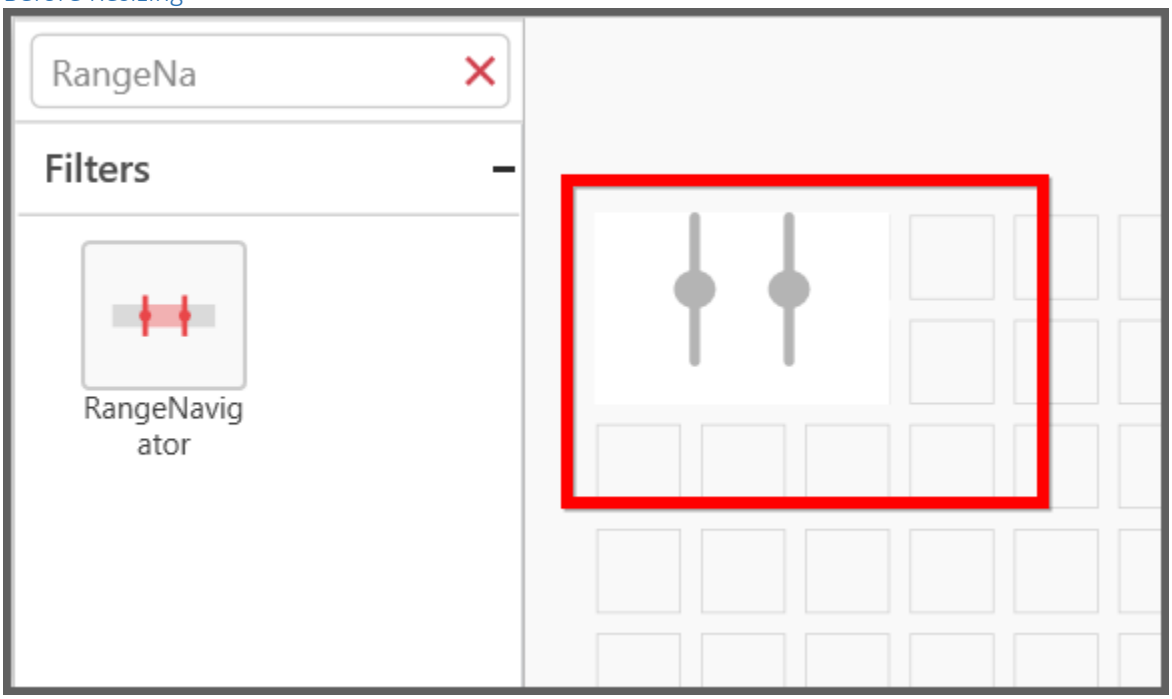
How to configure flat table data to Range Navigator?

The following procedure illustrates data configuration of Range Navigator.

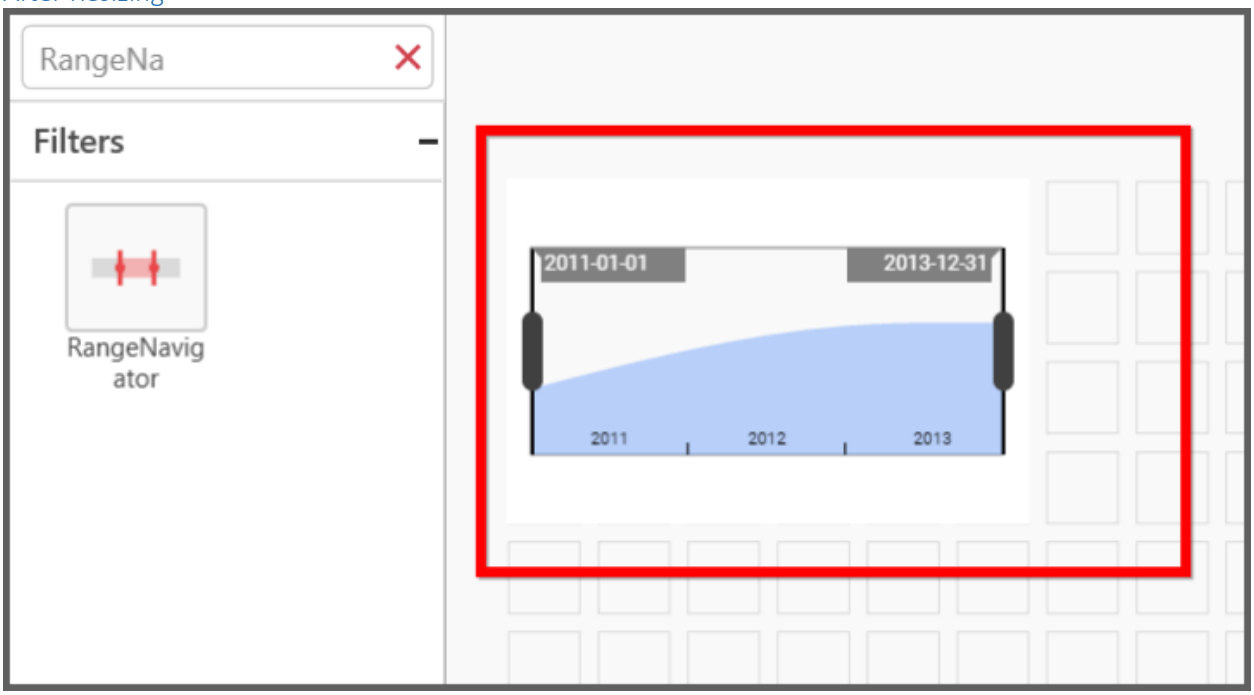
Drag and drop Range Navigator control icon from the Tool box into design panel. You can find control in Toolbox by search.



Before Resizing

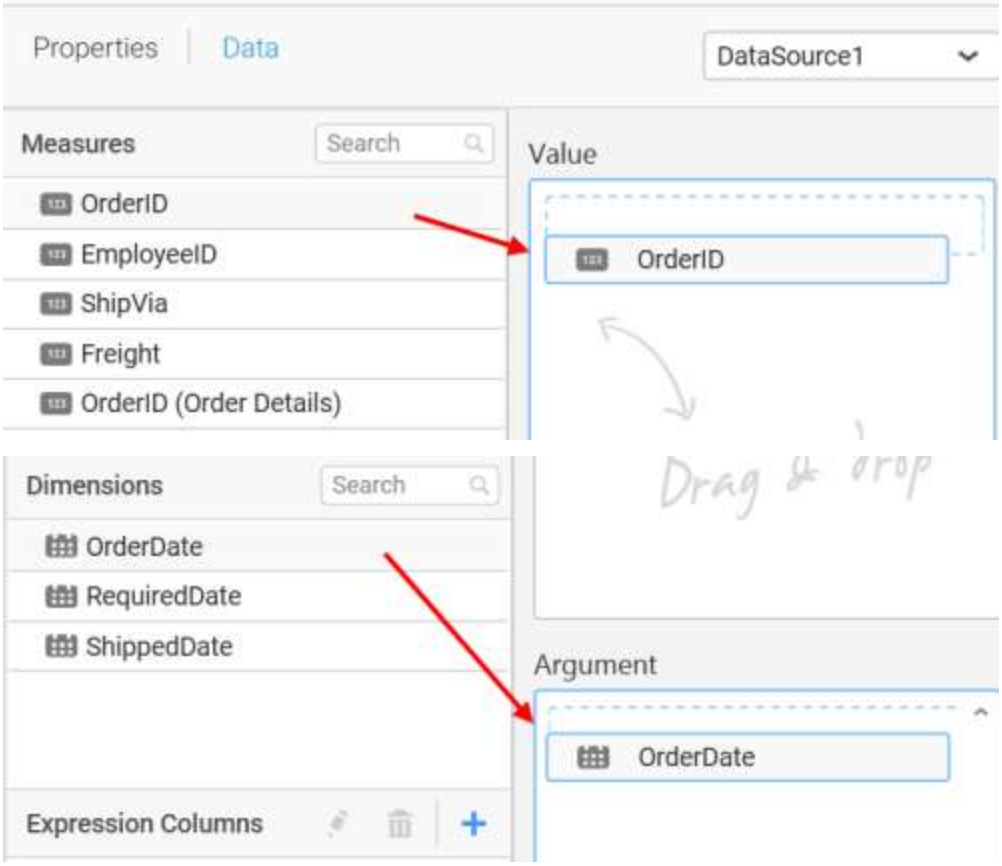


After Resizing

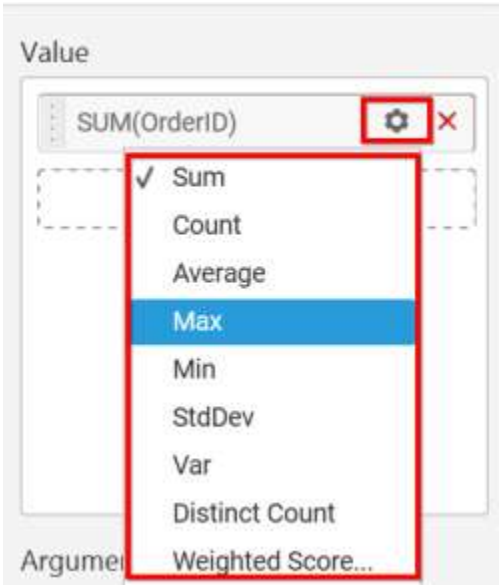


After control added in design panel, click **Assign Data** button at Design Tools Pane to open the Data configuration pane.

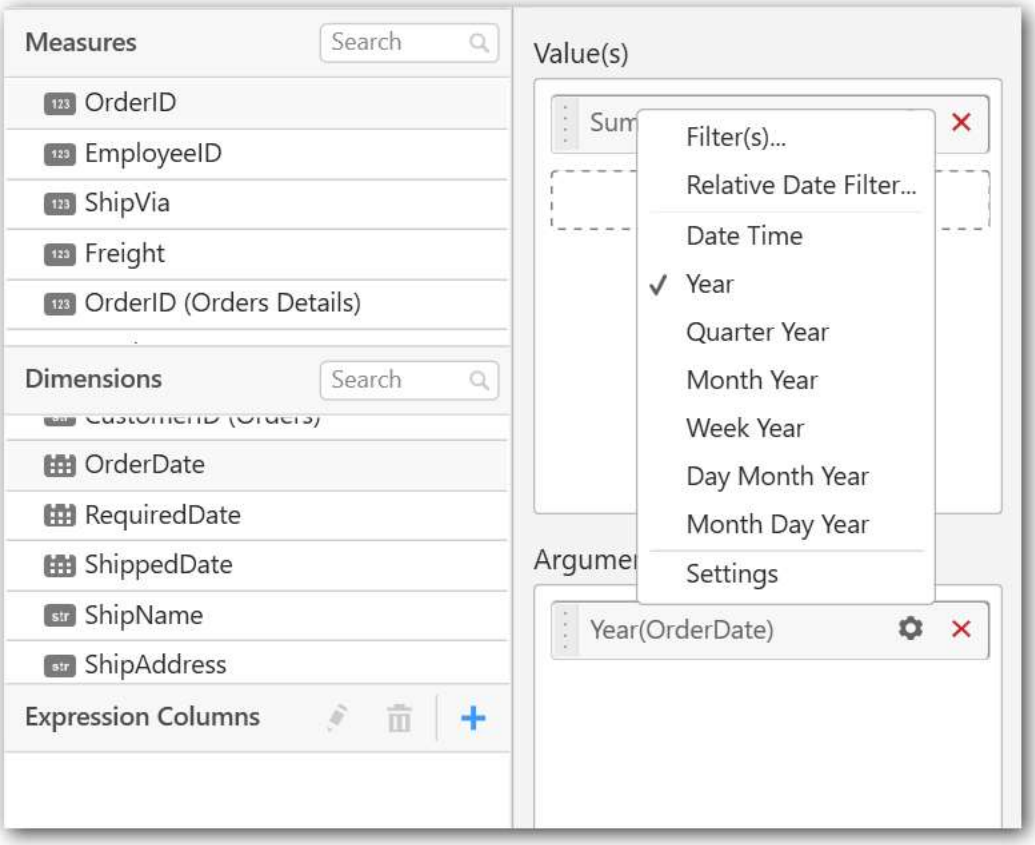
Bind a Measure column to Value section and a Date type column to Argument section.



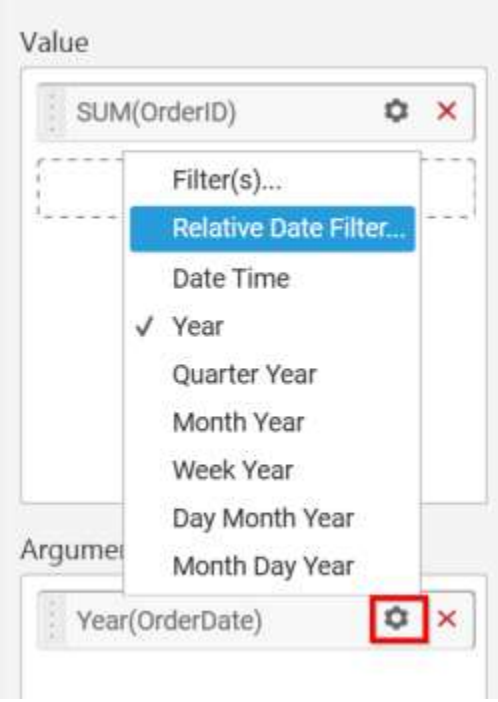
You can use the aggregation function to change the values of the elements.

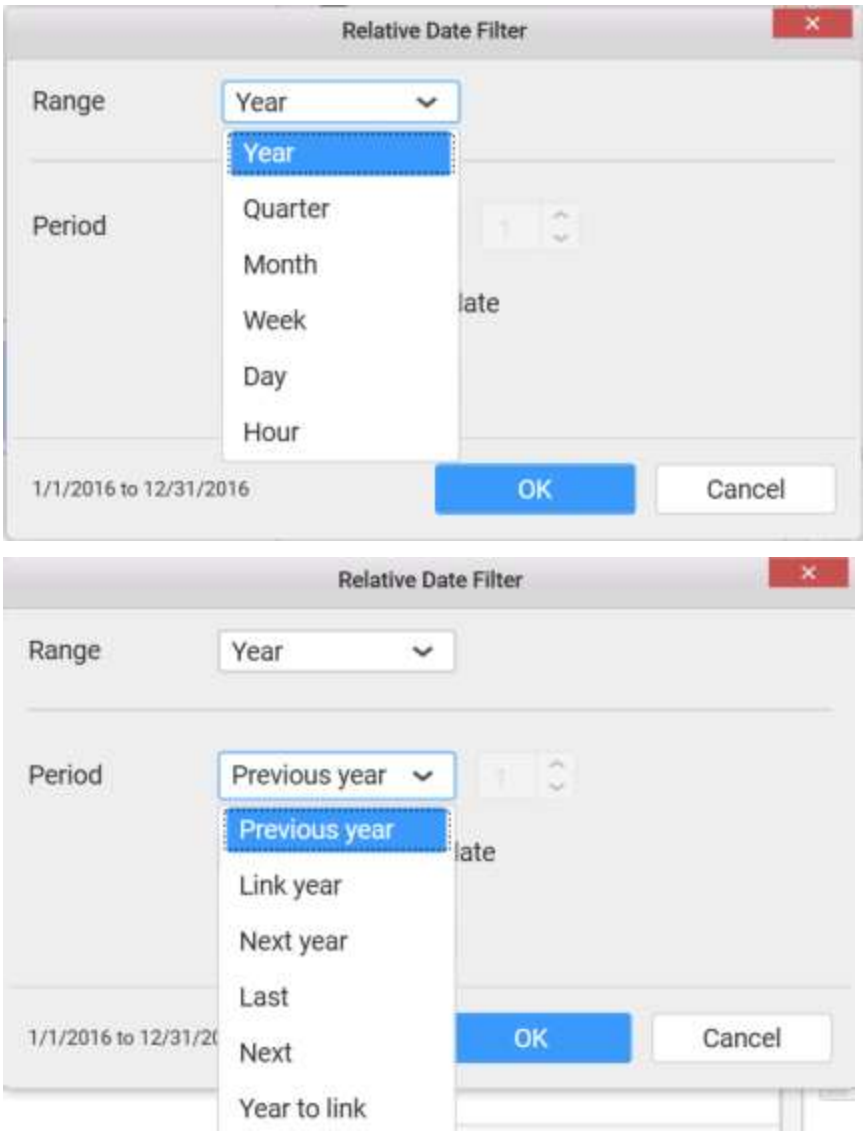


You can select the type of Date and Time as required by clicking the Setting button.

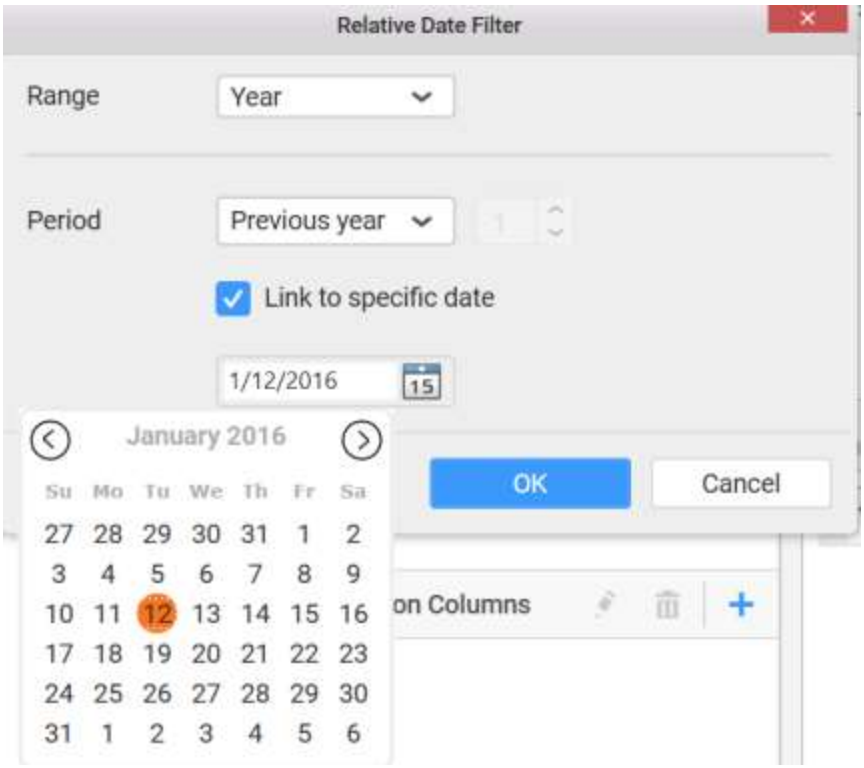


You can use the Relative Date Filter option to filter the Range and Period.

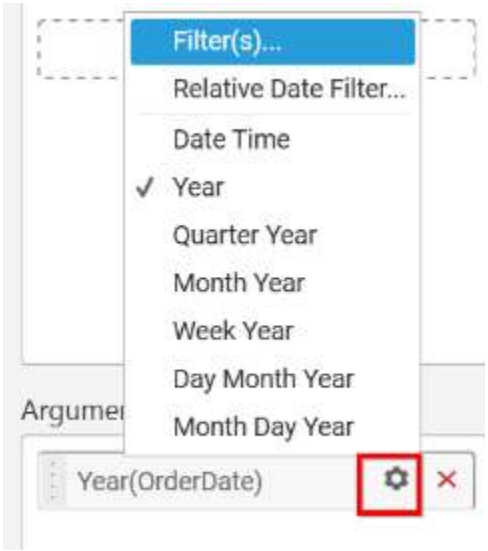




Select the Link to specific date option to select the specific date.

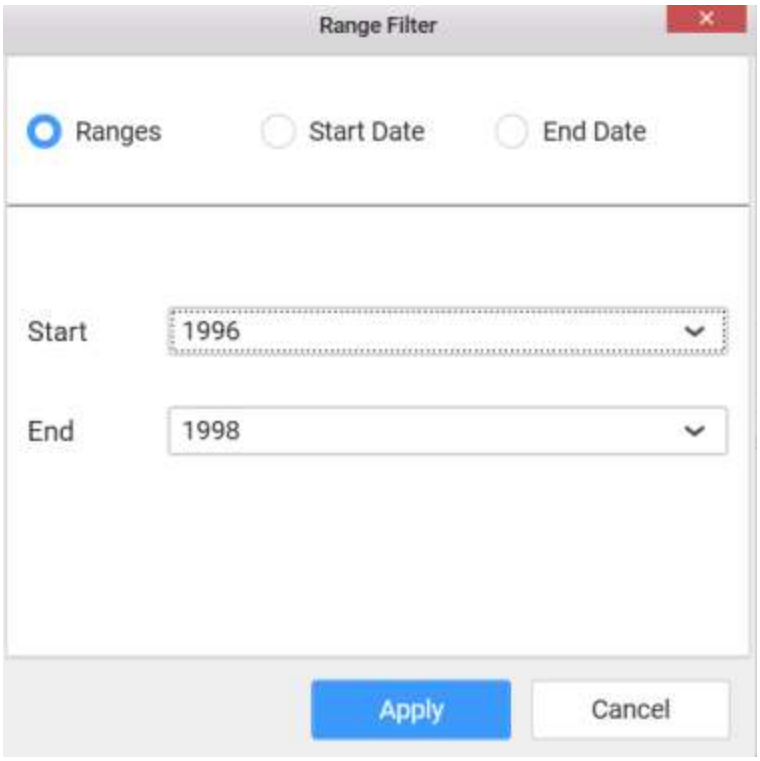


You can use the Filter(s)… option to apply the filter condition for date.

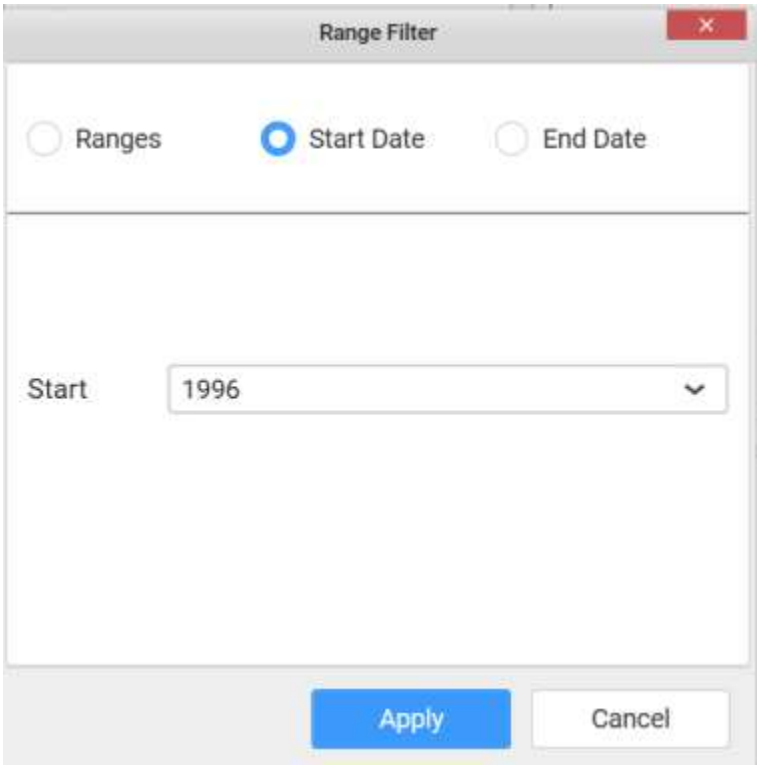


Range Filter will be shown to set either the Range, Start Date or End Date and click Apply.

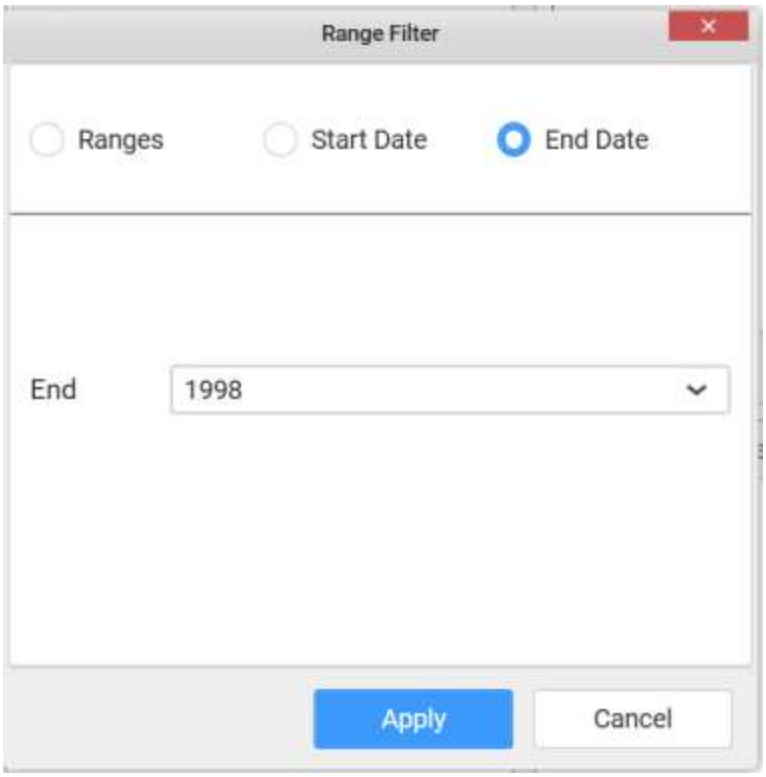
Ranges



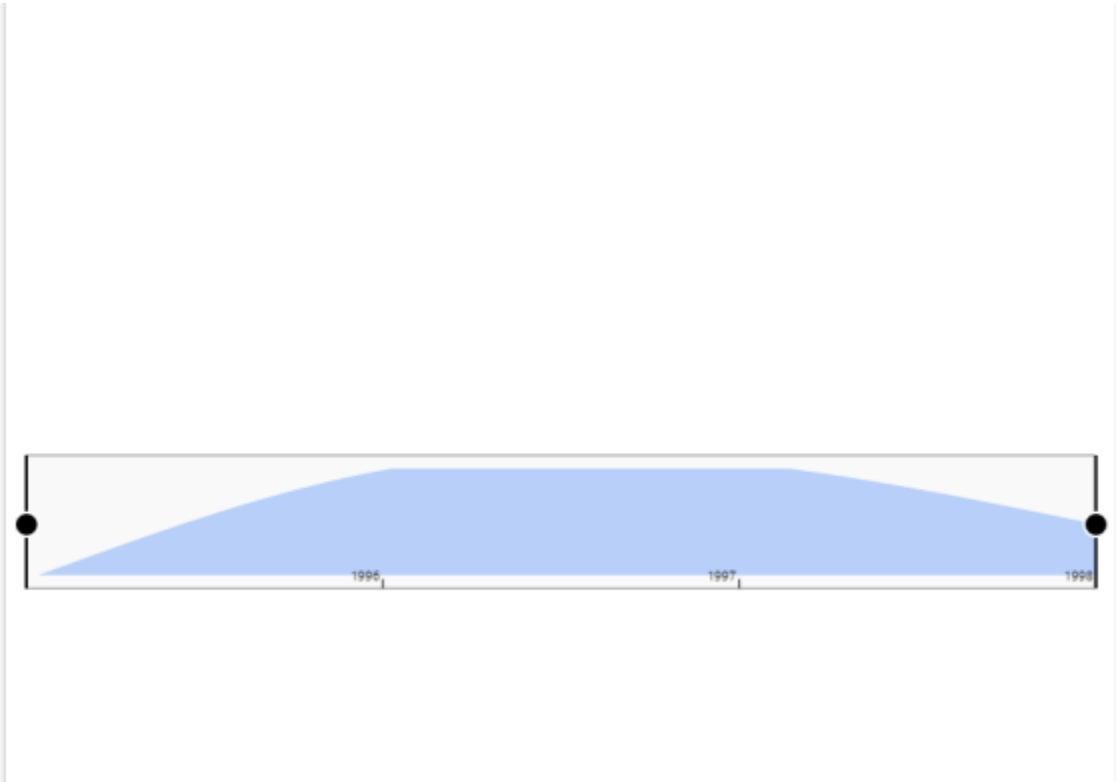
Start Date



End Date



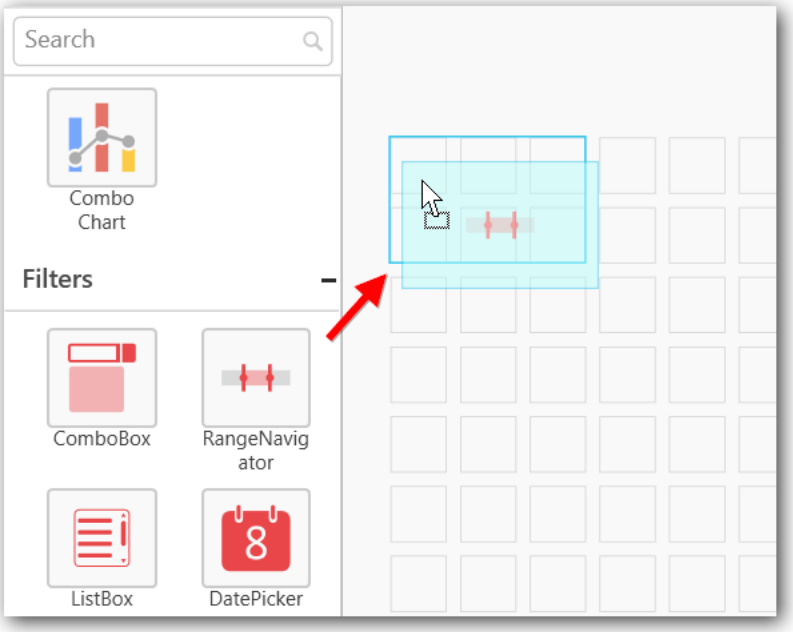
Here is an illustration,



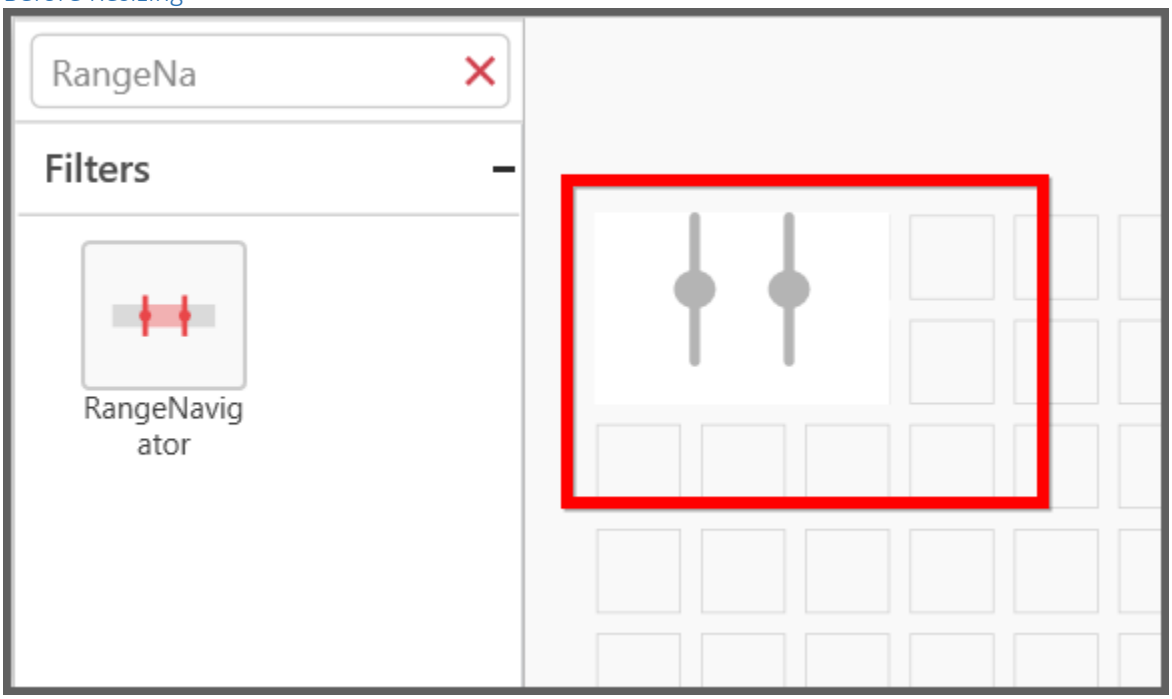
How to configure SSAS data to Range Navigator?

Following steps illustrates configuration of SSAS data to Range Navigator.

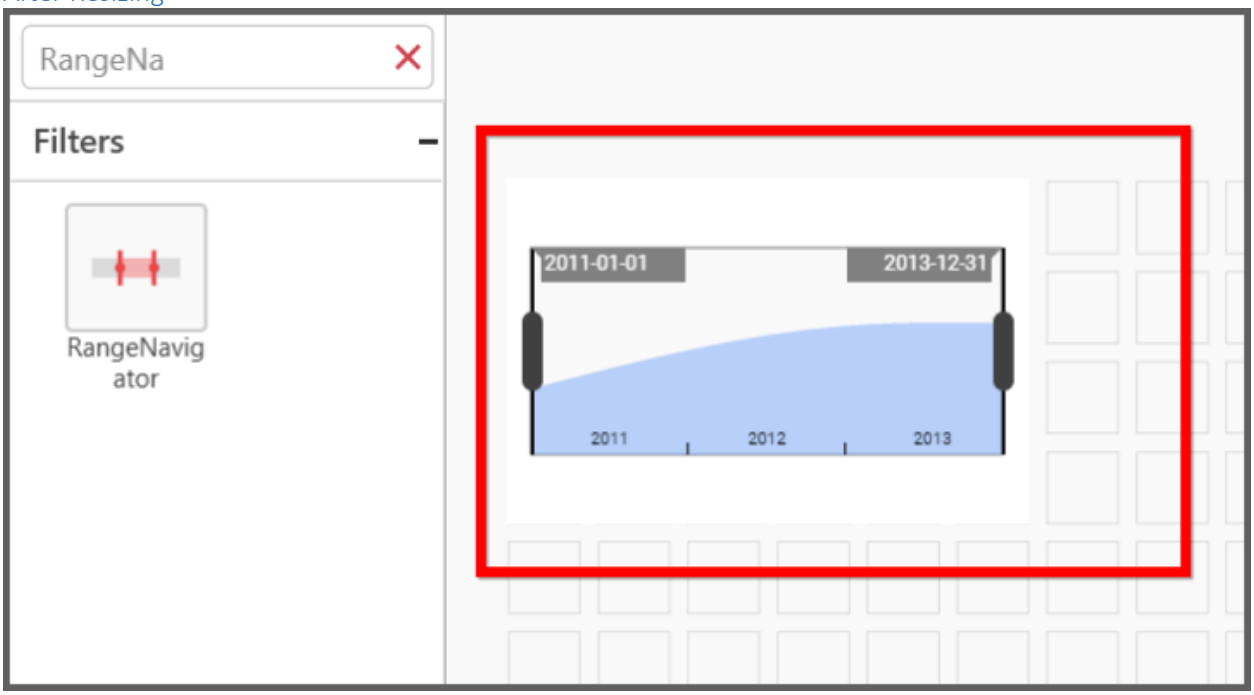
Drag and drop the Range Navigator widget into canvas and resize into your required size.



Before Resizing



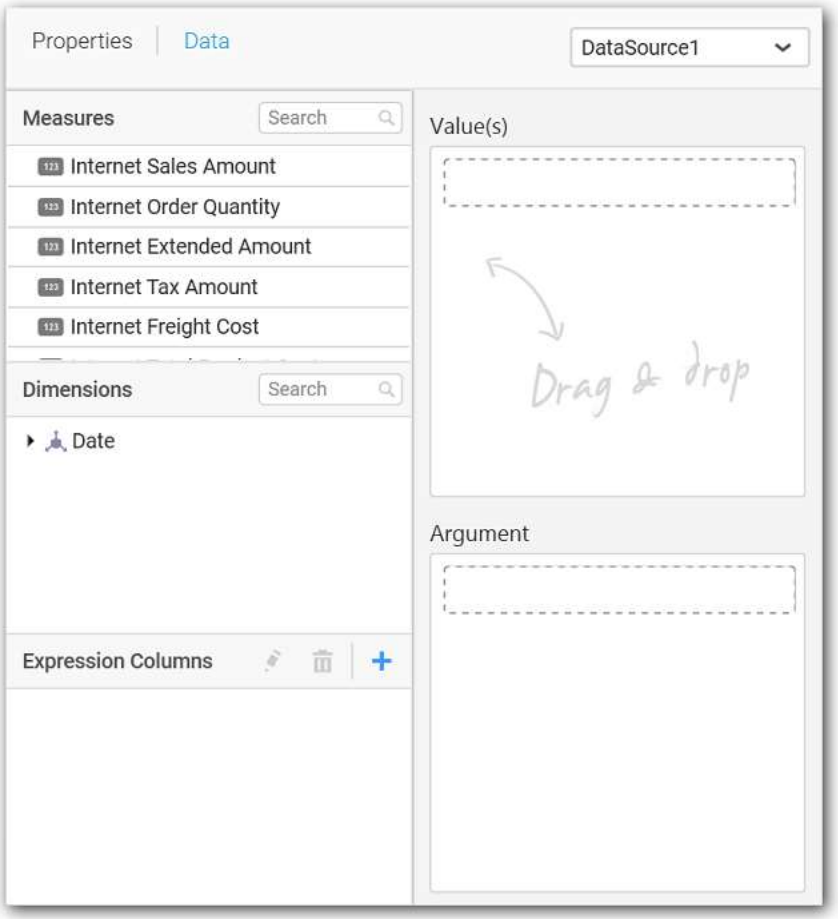
After Resizing



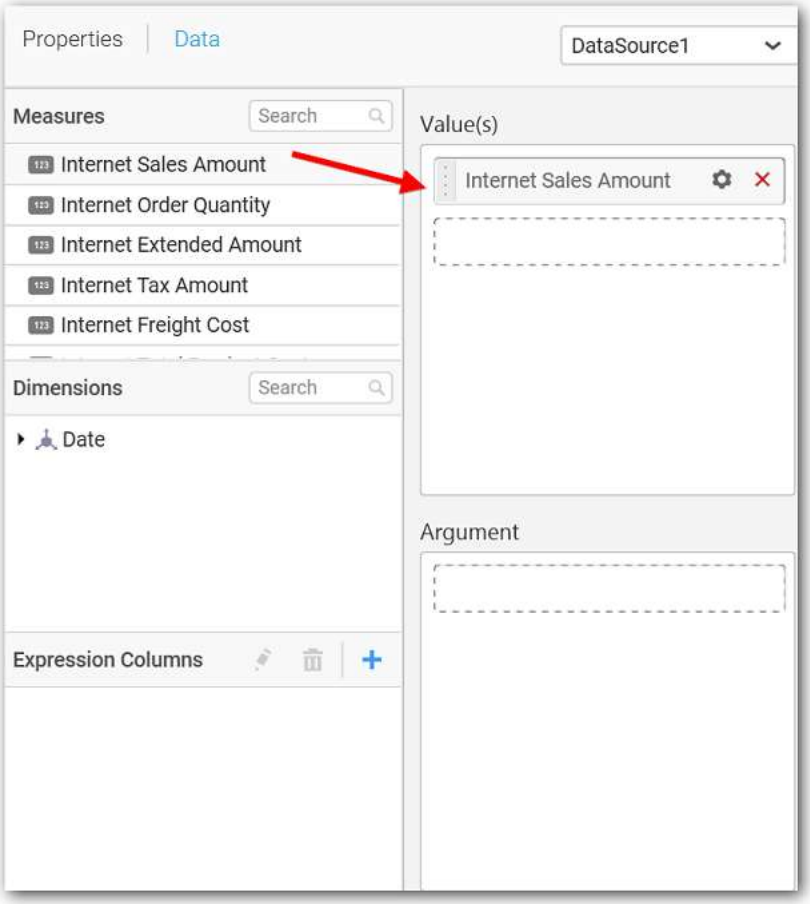
Select the dropped widget using mouse.

Click the Assign Data button in the toolbar.

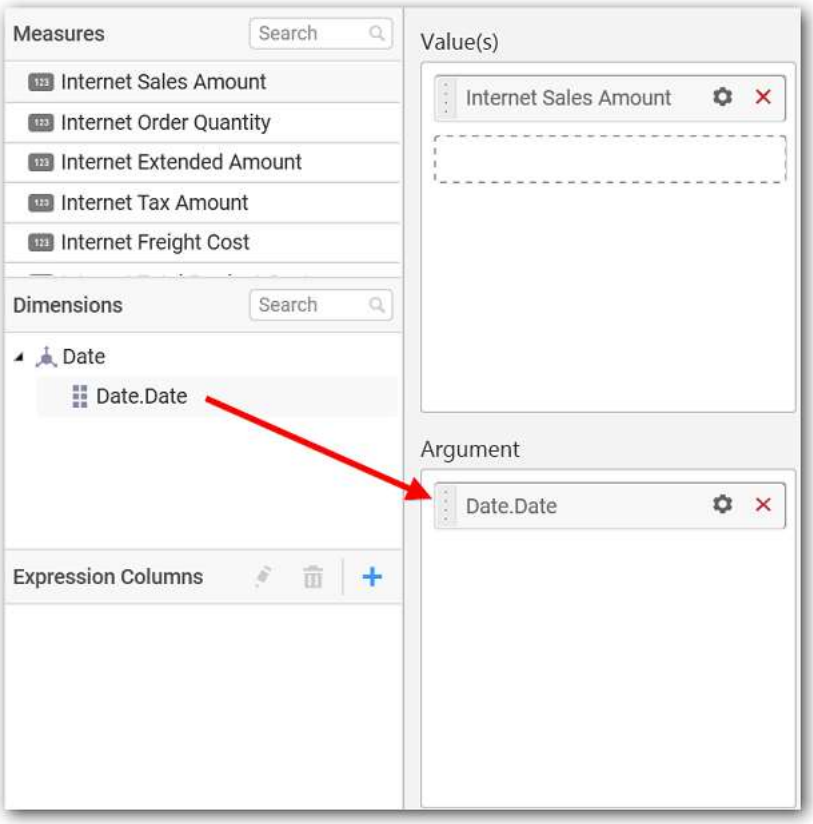
A Data pane will be opened with available Measures and Dimensions.



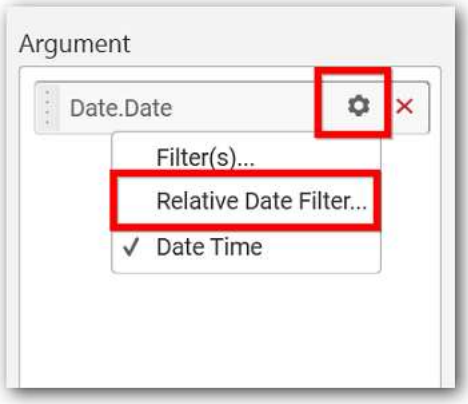
Drag and drop a column under Measures category into Value(s) section.

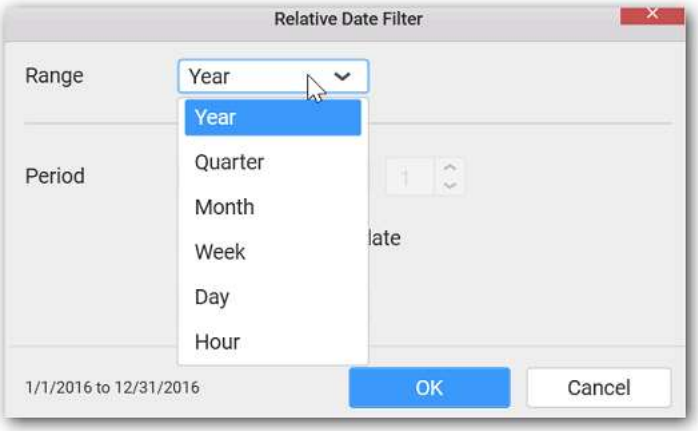


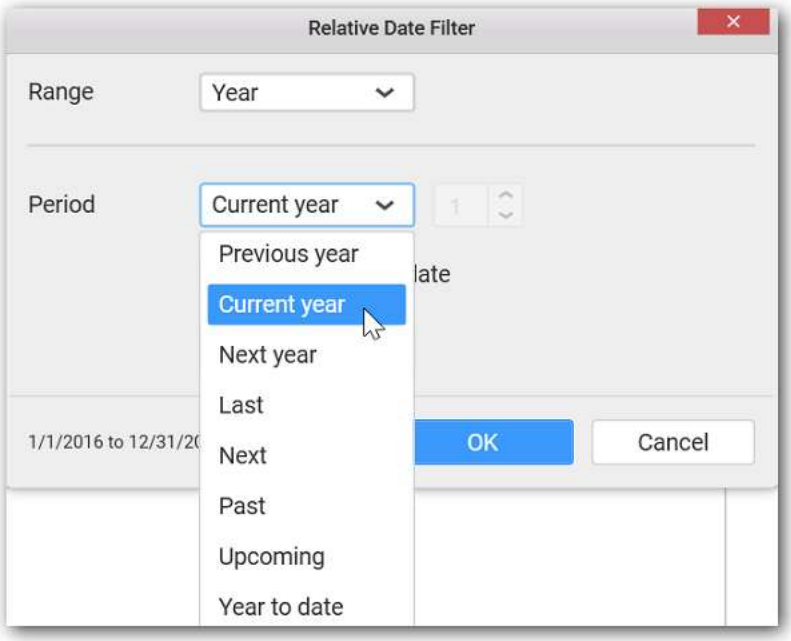
Drag and drop a column under Dimensions category into Argument section.



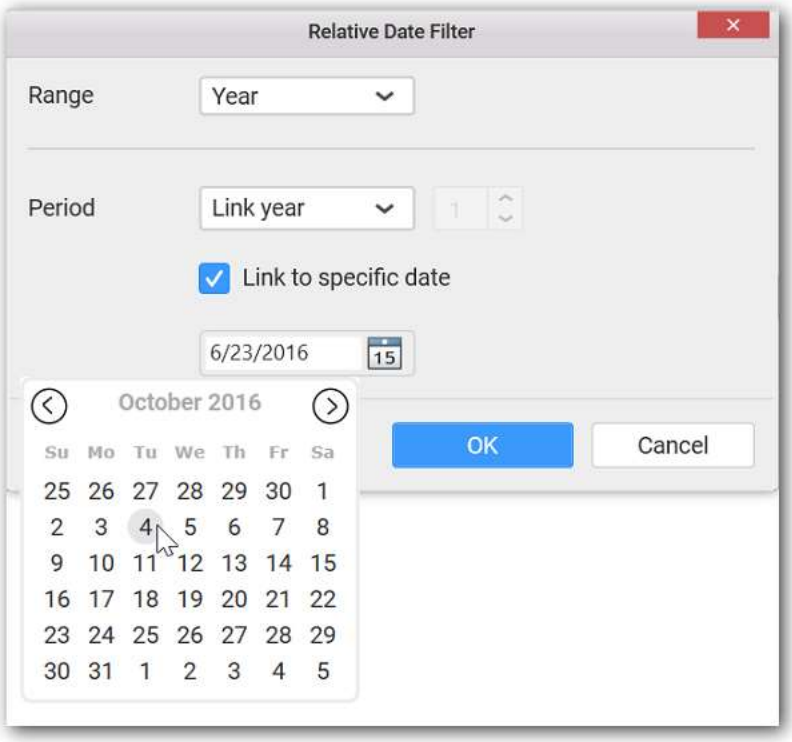
Define relative date filter criteria through Relative Date Filter menu item in the Settings drop down menu.



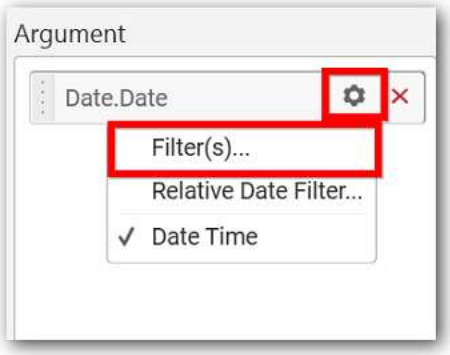




Select the Link to specific date option to select the specific date



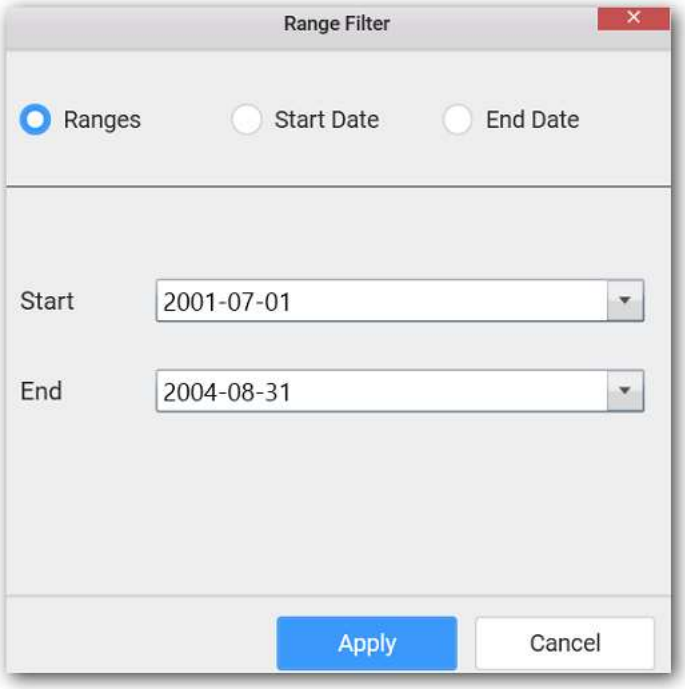
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.



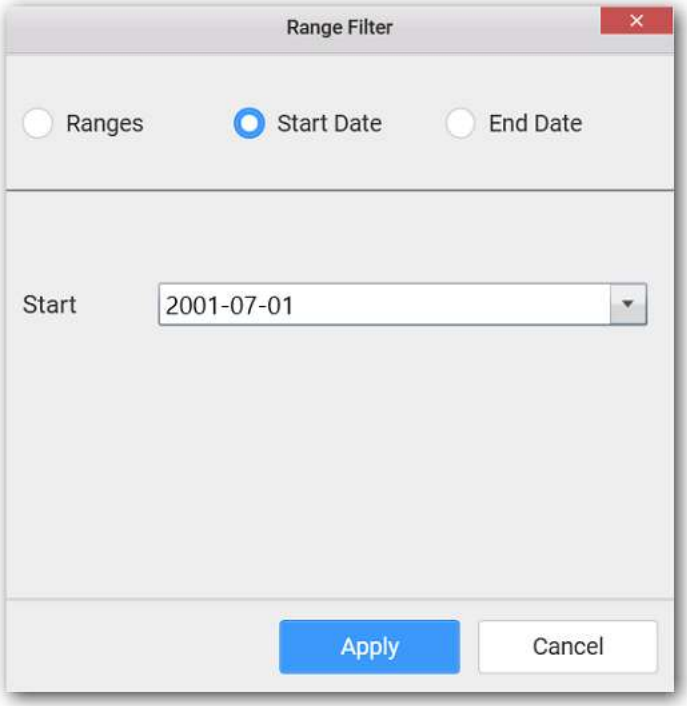
Select Filter(s)... to launch the Filters window.

set either the Range, Start Date or End Date and click Apply.

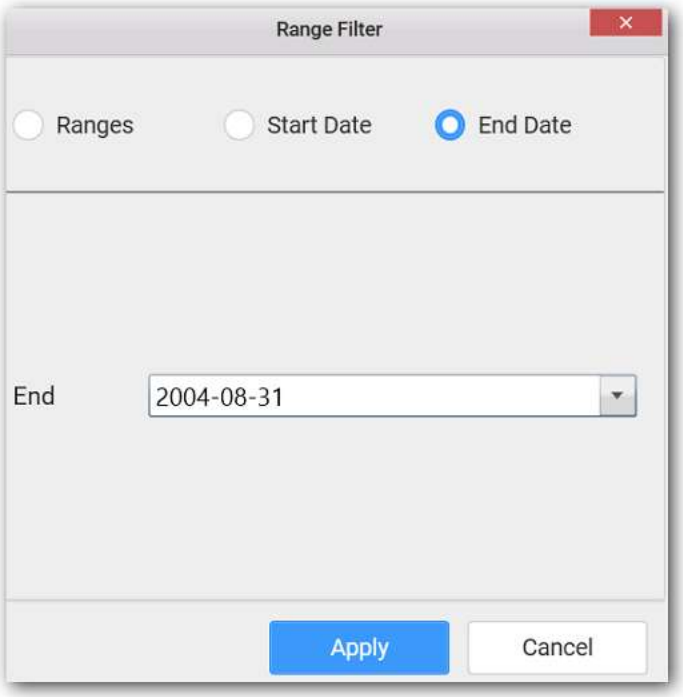
Ranges



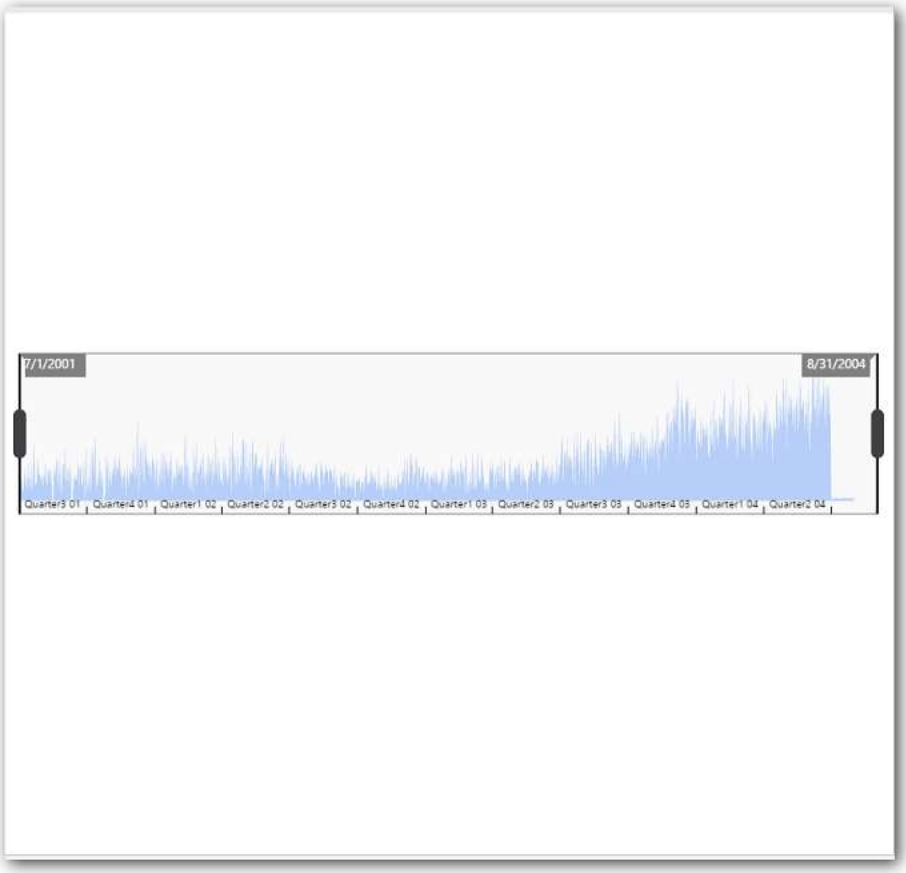
Start Date



End Date



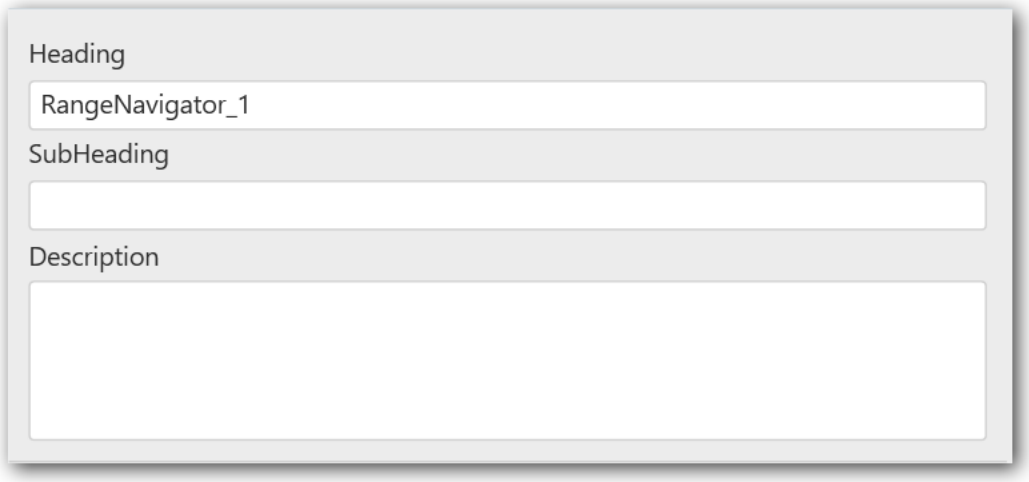
Here is an illustration,



How to Format Range Navigator widget?

You can format the Range Navigator for better illustration of the view that you require, through the settings available in Properties pane.

General Settings



Header

This allows you to set title for this range navigator widget.

SubHeading

This allows you to set sub-title for this ComboBox widget.

Description

This allows you to set description for this range navigator widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

Series Type

You can change the series type to any chart type to render in range navigator.

Filter settings

Act as Master Widget

This allows you to define this range navigator widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Options

This allows you to define this range navigator widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

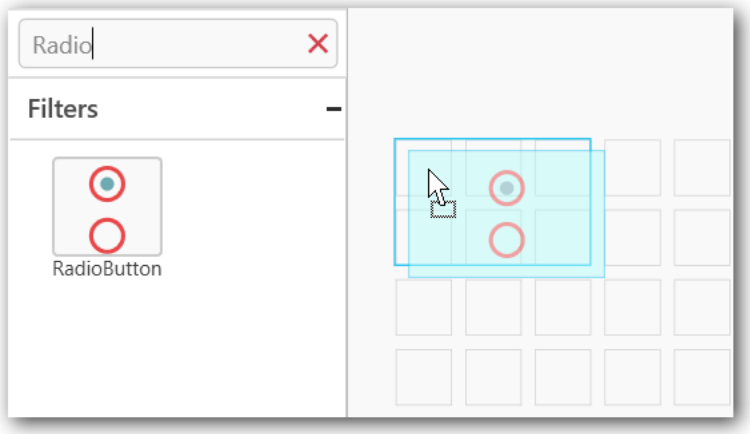
Radio Button

Radio Button allows you to filter based on single items selection in a group. To configure a radio button, a minimum requirement of 1 column is needed.

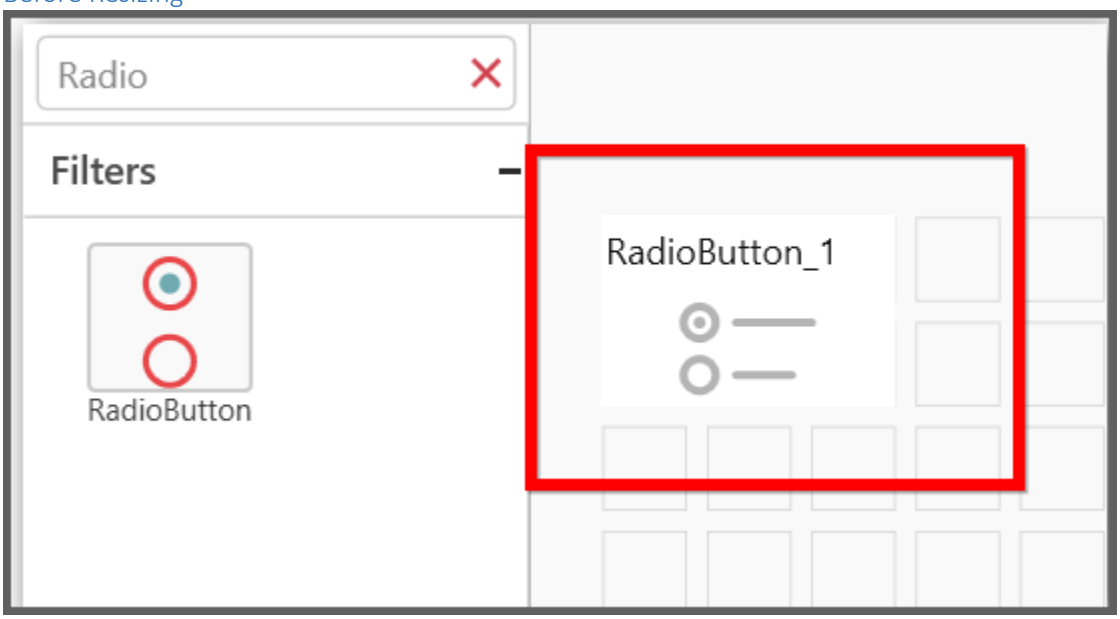
How to configure the flat table data to Radio Button?

The following procedure illustrates data configuration of Radio Button.

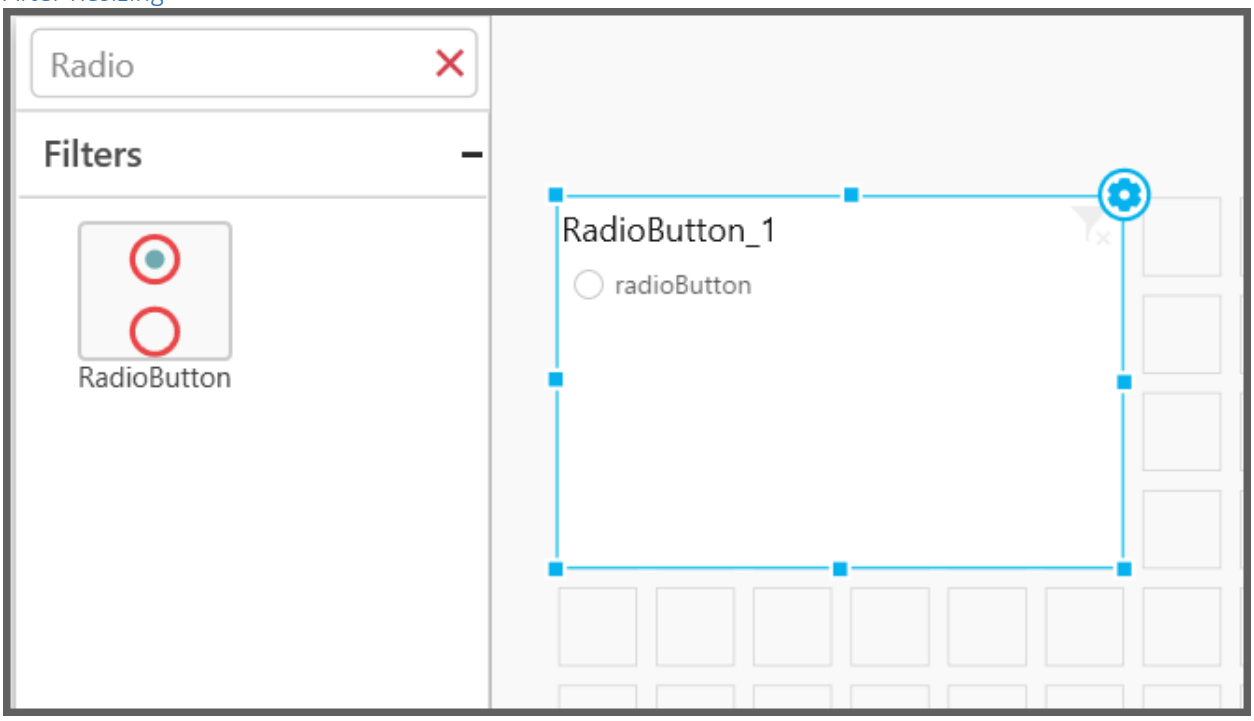
Drag and drop Radio Button control icon from the Tool box into design panel. You can find control in Toolbox by search.



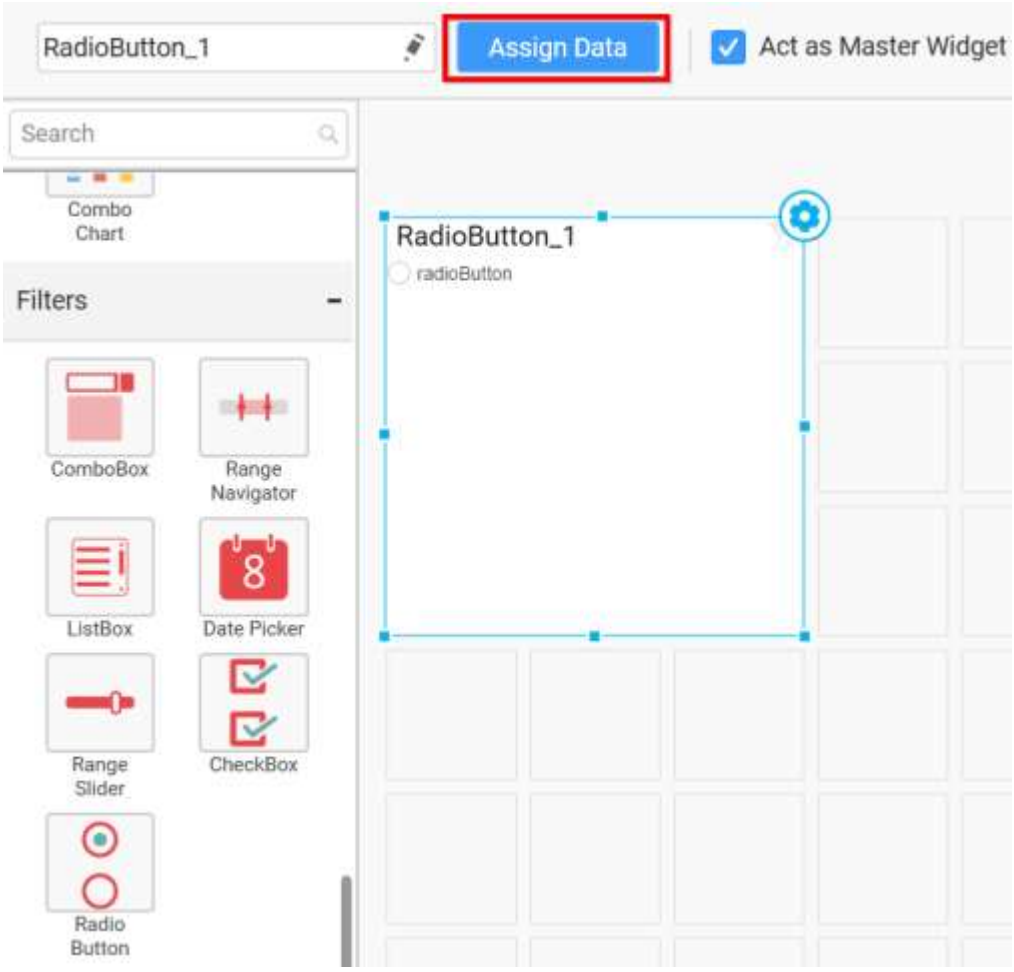
Before Resizing



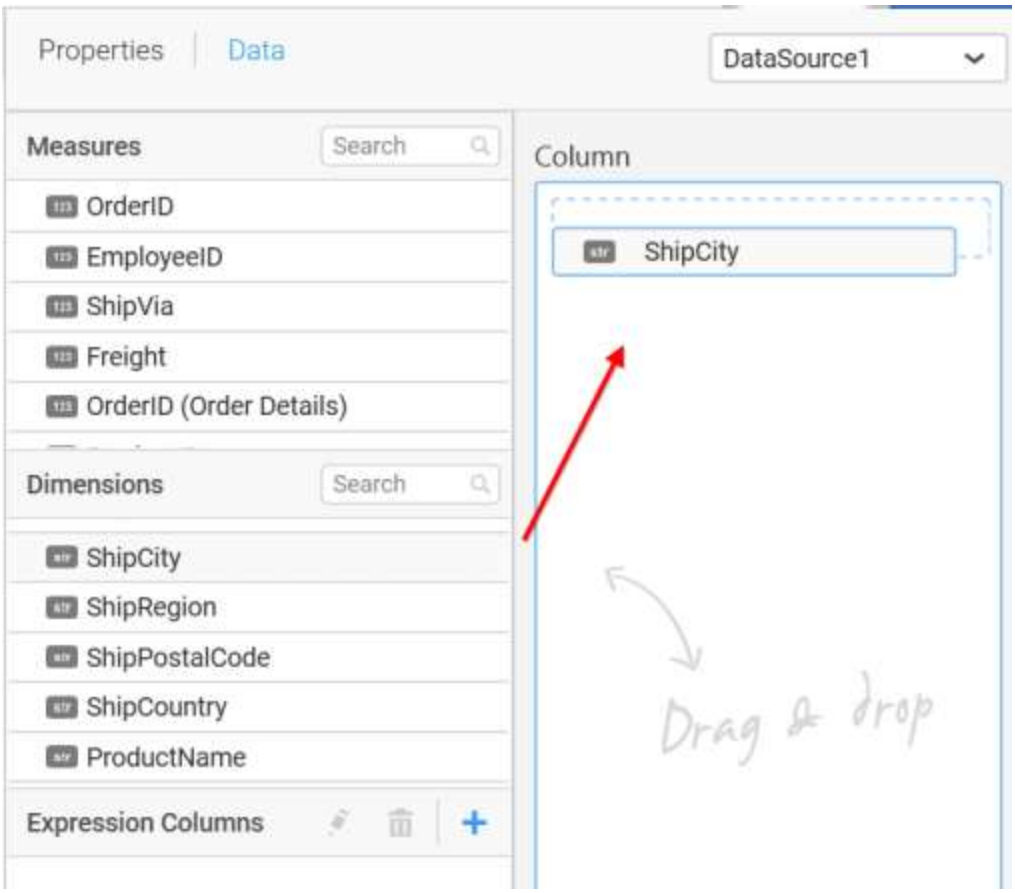
After Resizing



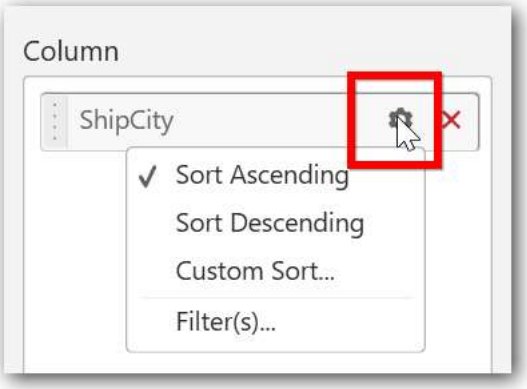
After control added in design panel, click Assign Data button at Design Tools Pane to open the Data configuration pane.



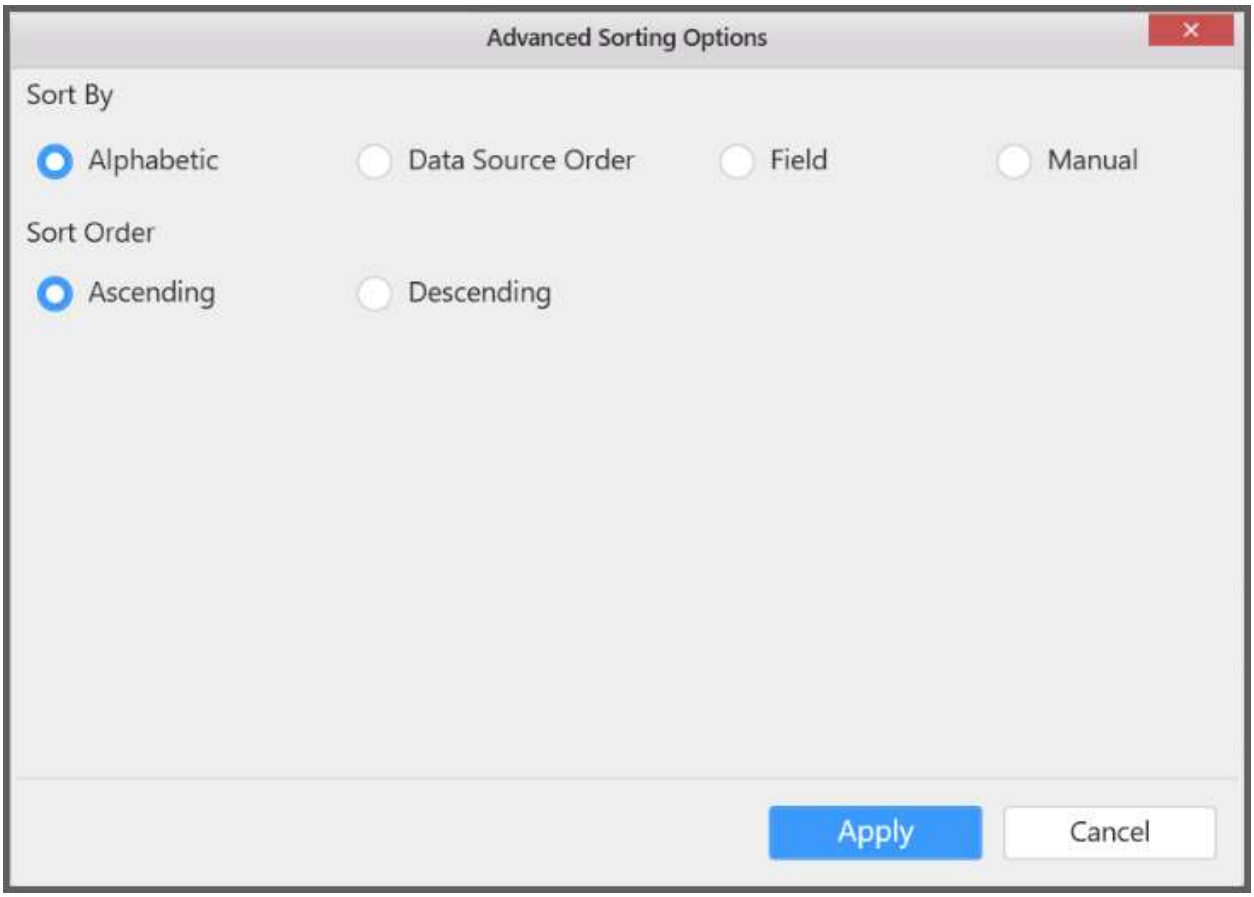
Bind column through drag and drop element from sections to Column section.



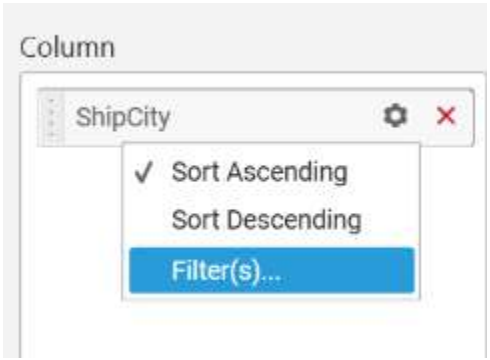
You can select the settings to sort the data either Ascending or Descending.



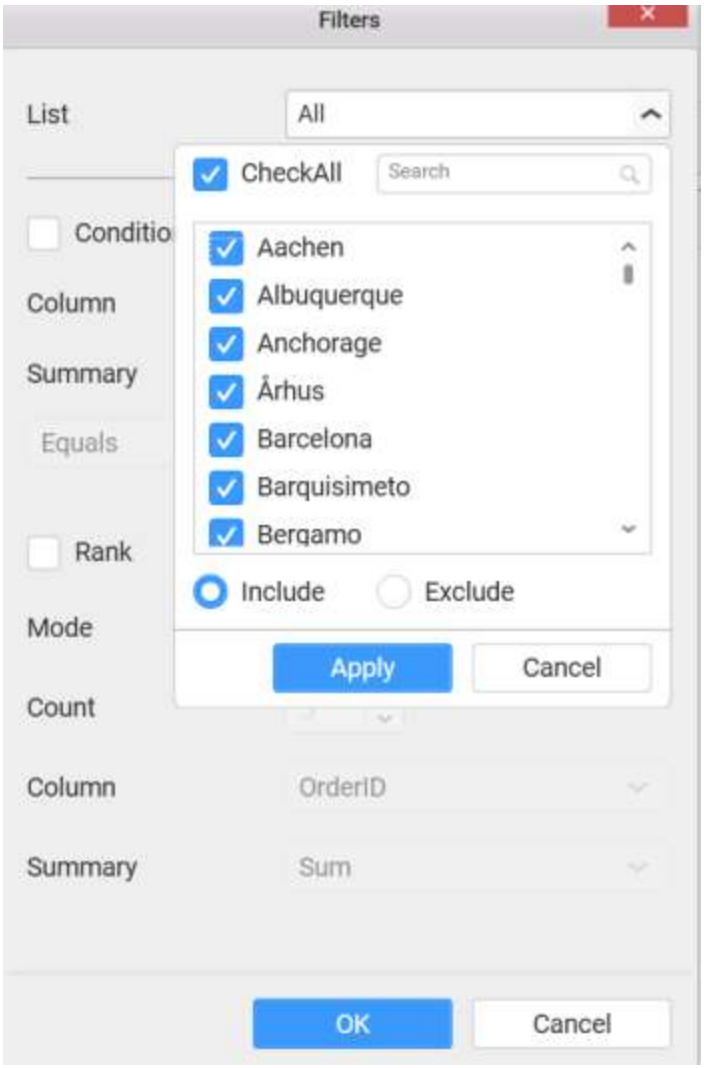
You can also select custom sorting.



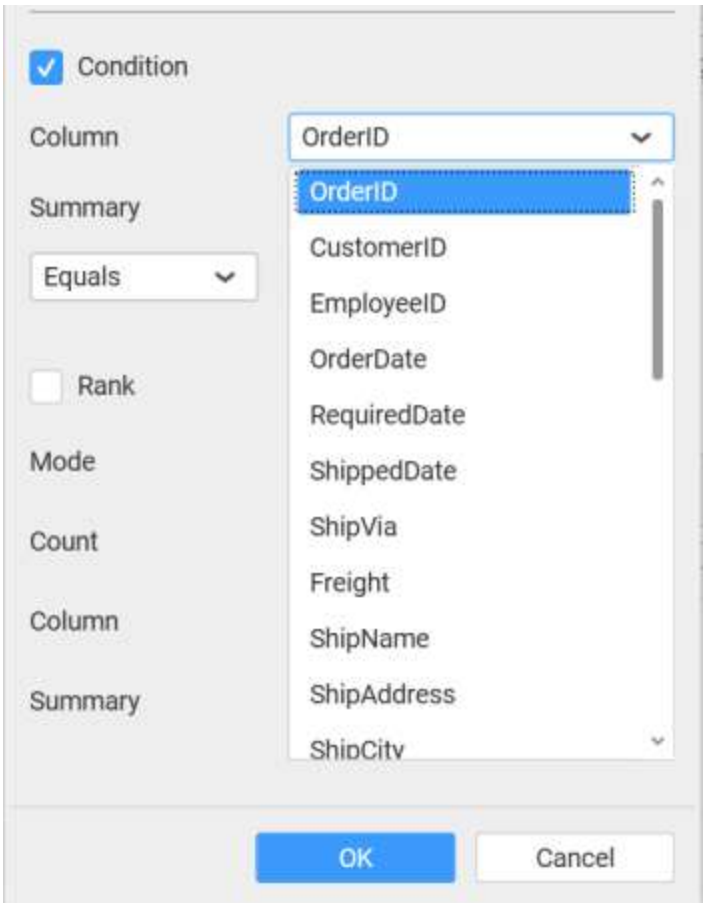
You can use the filters by selecting the Filter(s)... option to rank the elements.

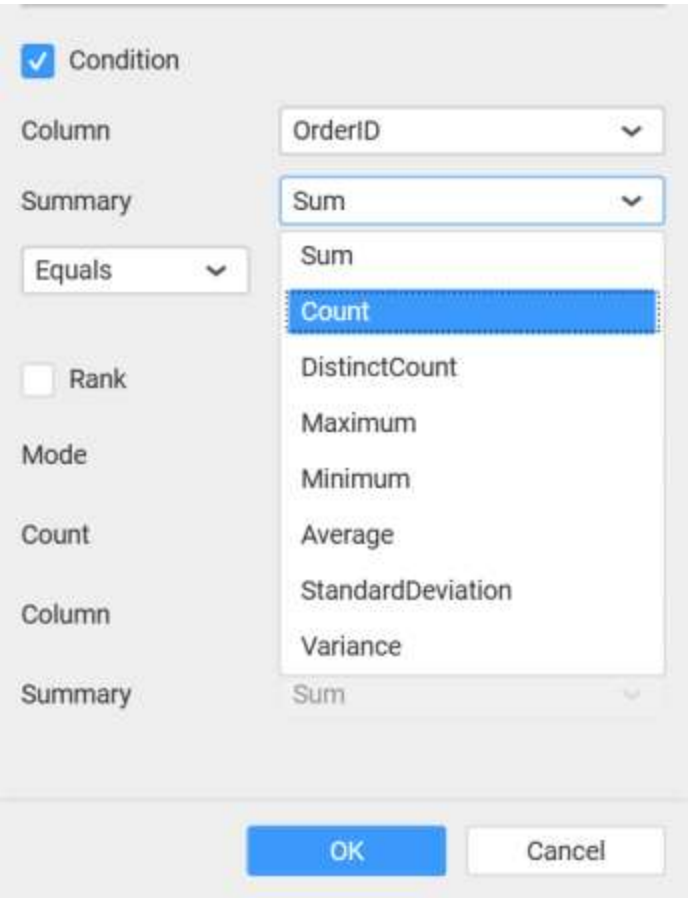


You can select the specific city to filter the element and CheckAll is used either to check all the data or to select the specific data. Include and Exclude is used to include and exclude the elements by selecting the radio button and click the Apply button.

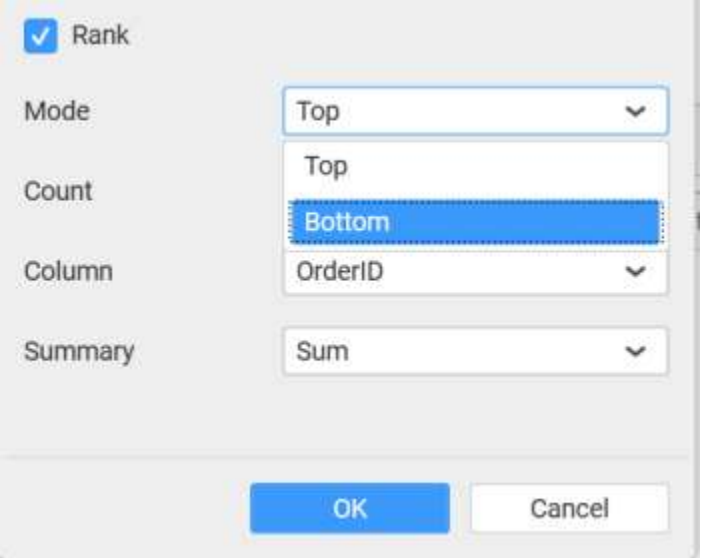


You can select the Condition option to change the Column elements and *Summary type by selecting the required column name and summary type.

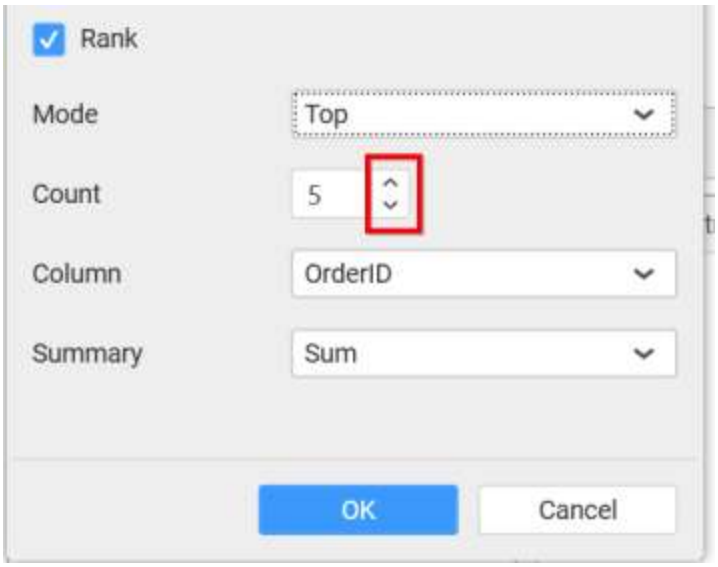




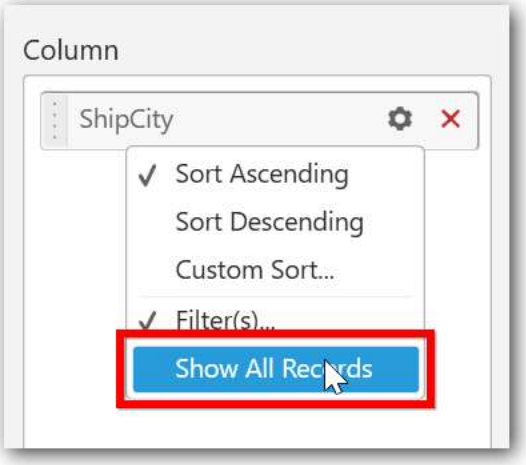
You can select the Rank option to enable filters and select the Mode either top or bottom.



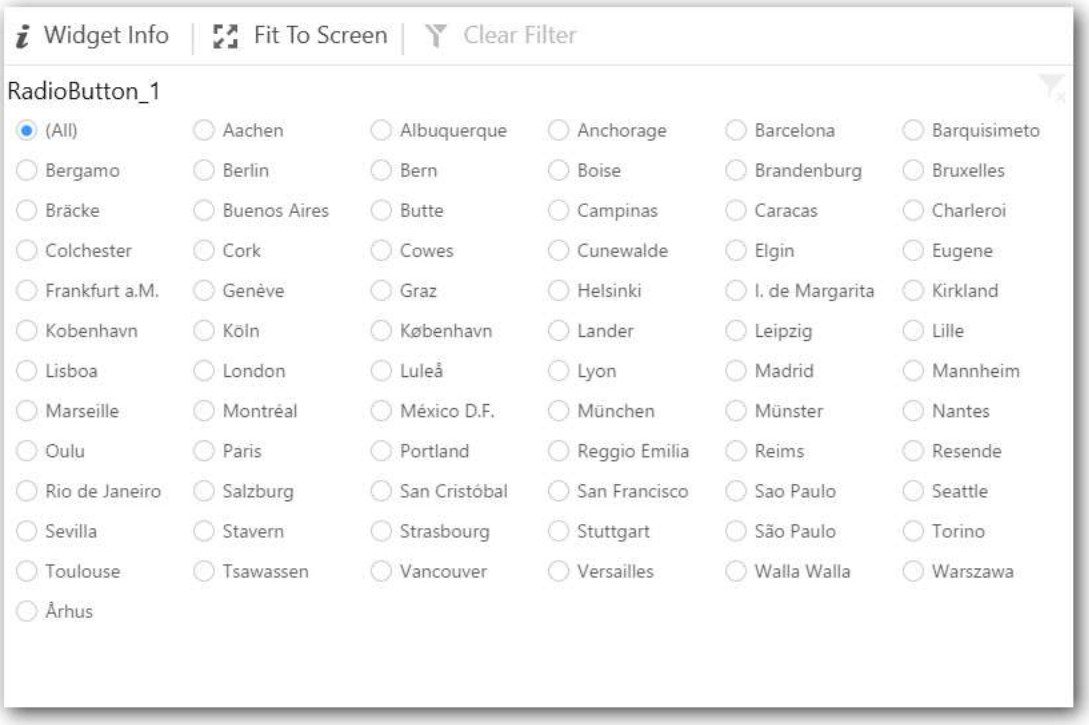
You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.



You can clear the filters by selecting the Show All Records option.



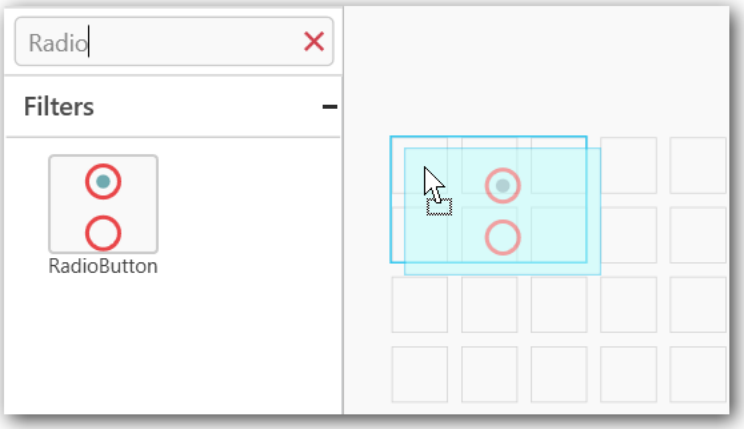
Here is an illustration,



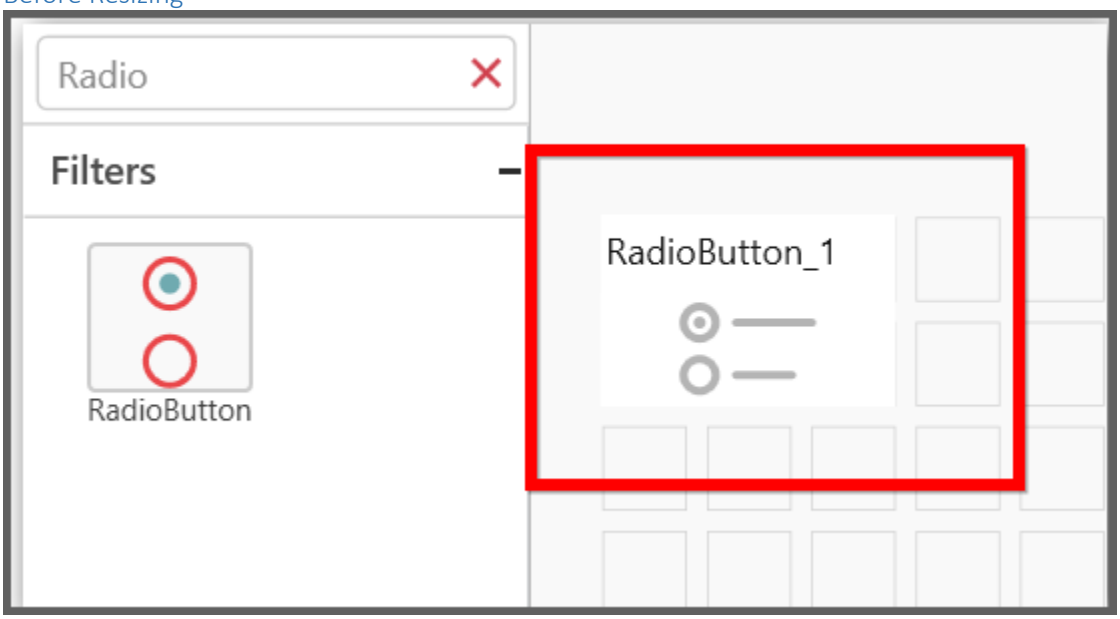
How to configure the SSAS data to Radio Button?

Following steps illustrates configuration of SSAS data to Radio Button.

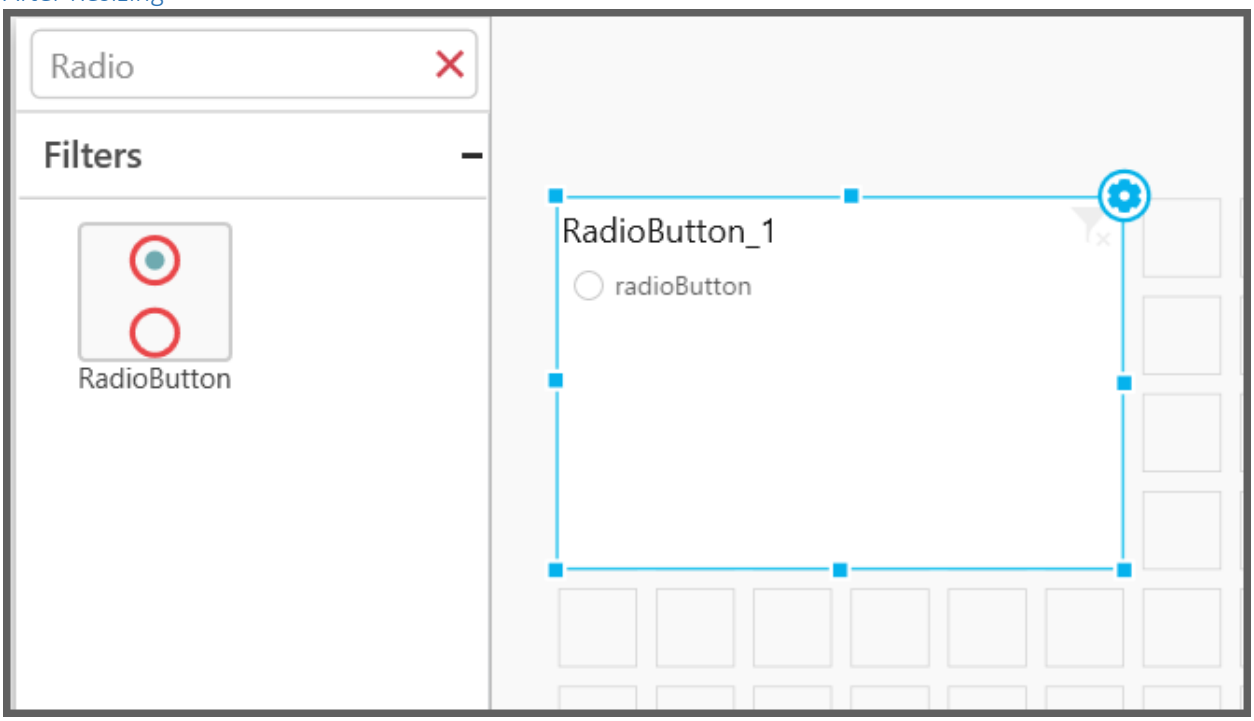
Drag and drop the Radio Button widget into canvas and resize into your required size.



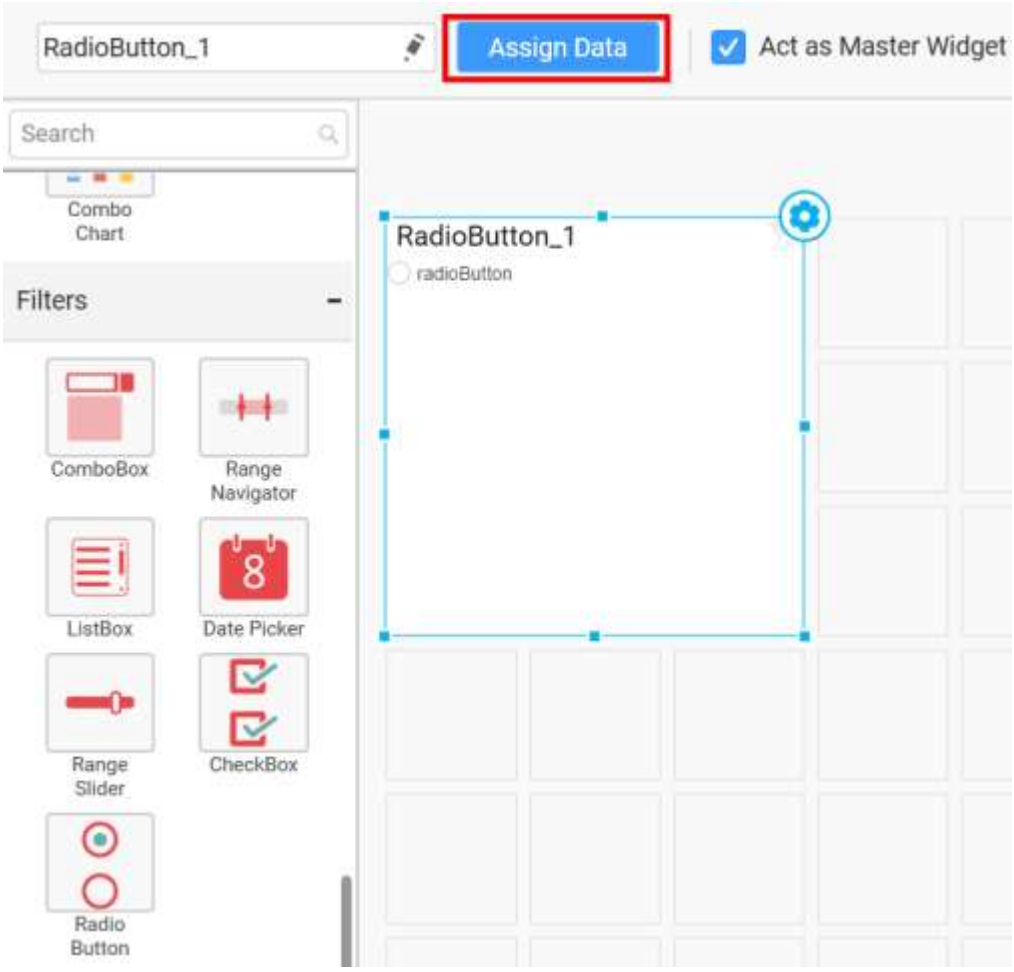
Before Resizing



After Resizing

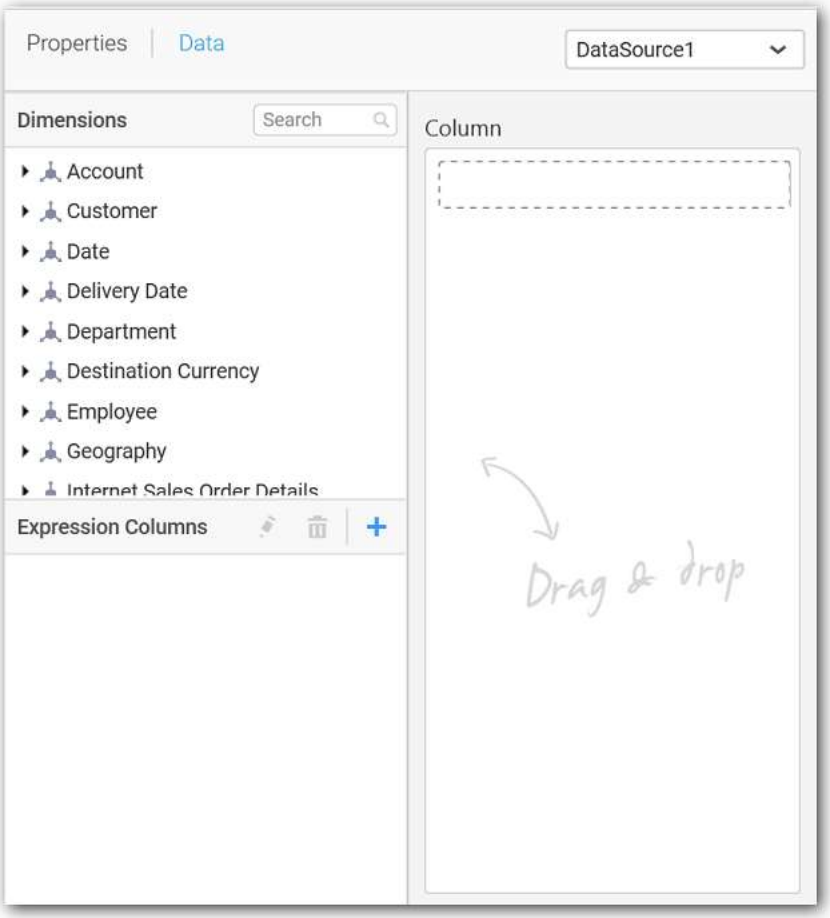


Select the dropped widget using mouse.

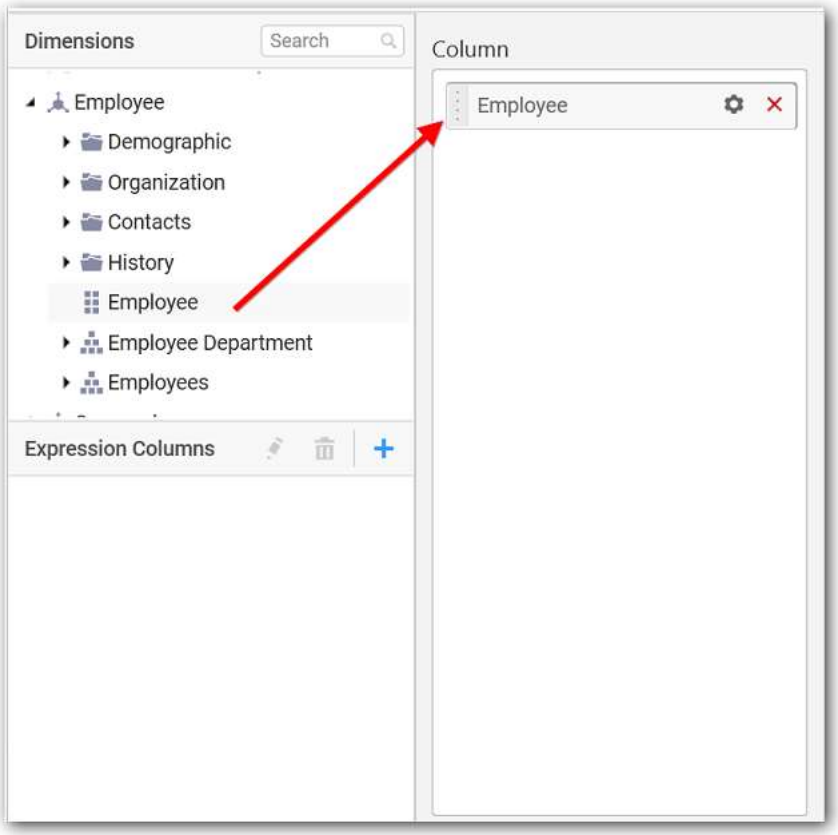


Click the Assign Data button in the toolbar.

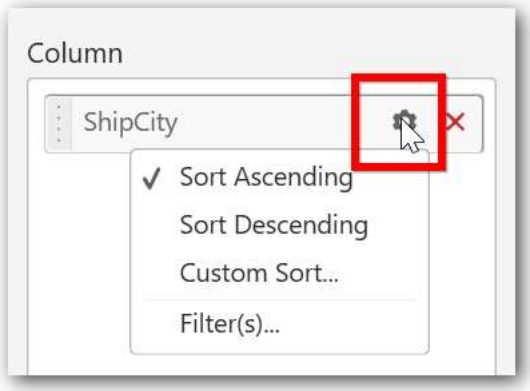
A Data pane will be opened with available Dimensions.



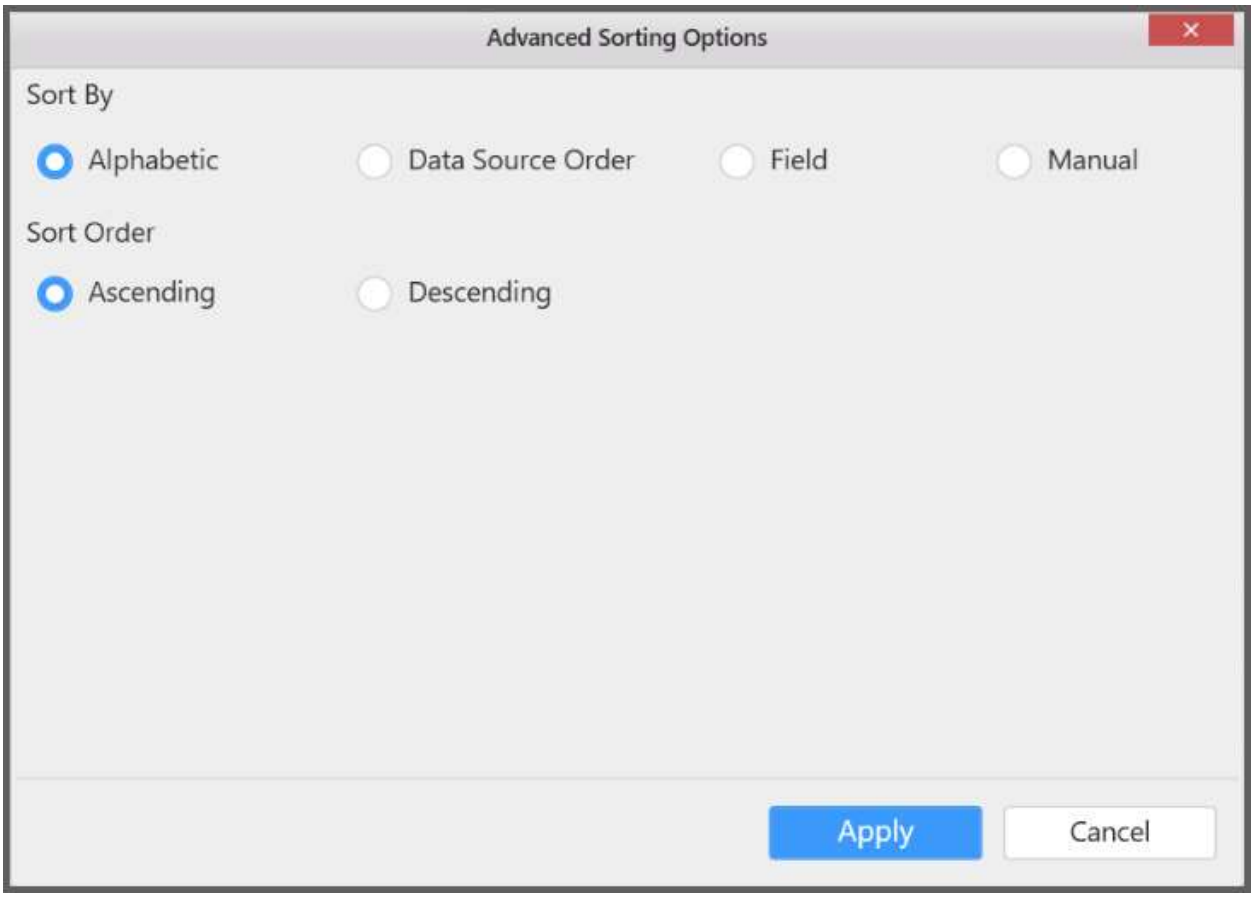
Add a dimension level or hierarchy into Column(s) section through drag and drop.



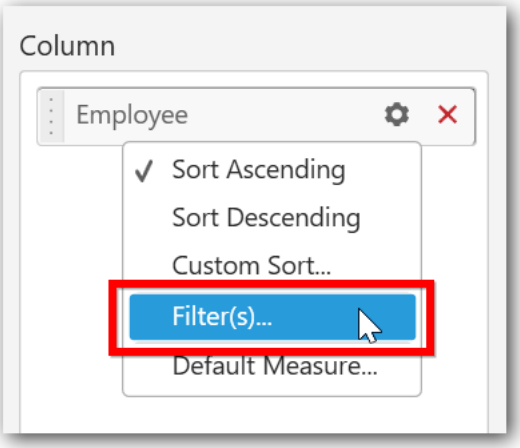
You can select the settings to sort the data either Ascending or Descending.



You can also select custom sorting.

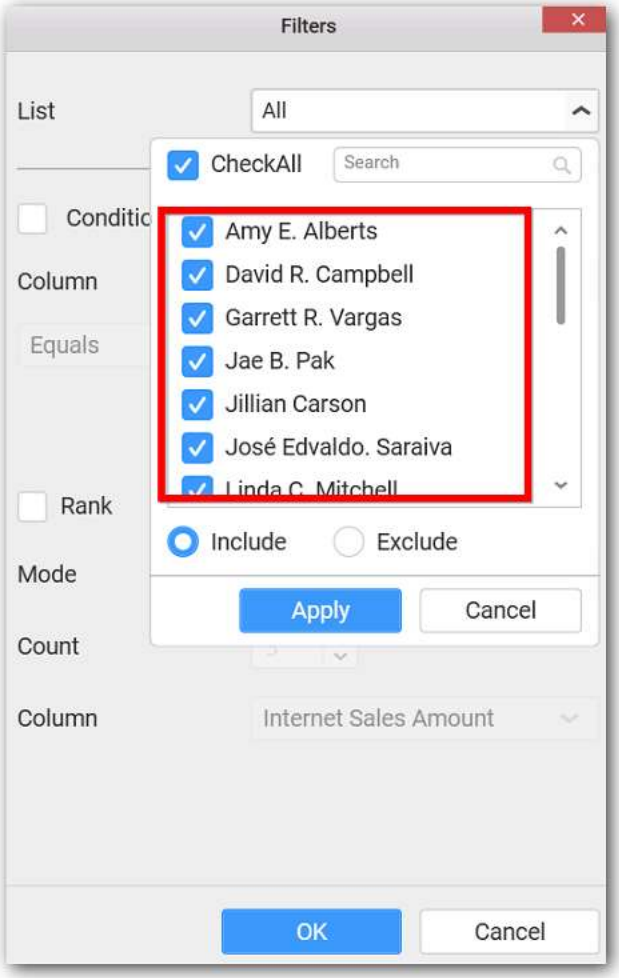


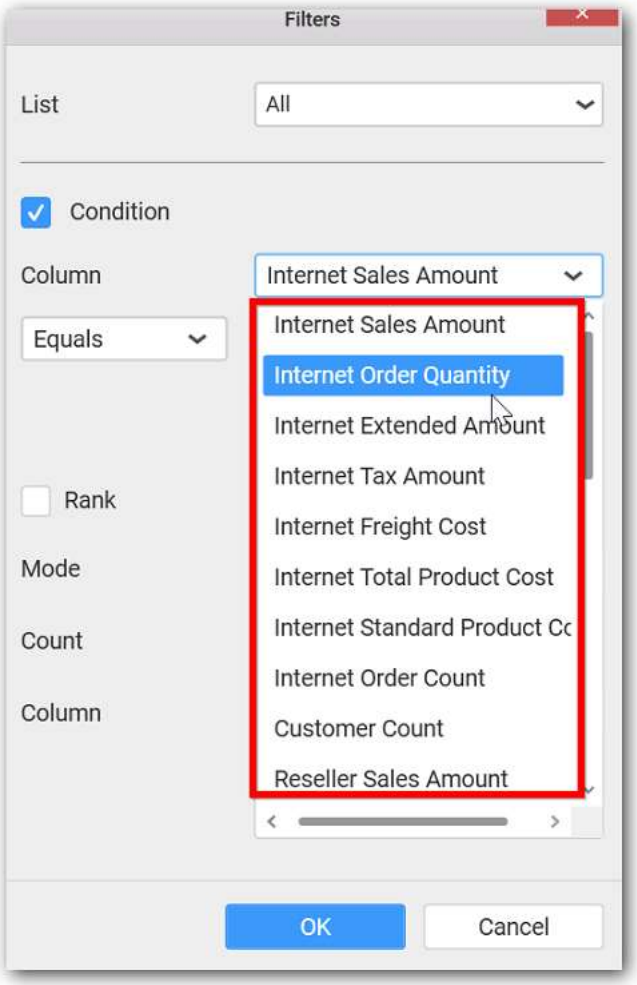
Define filter criteria through Filter(s)... menu item in the Settings drop down menu.

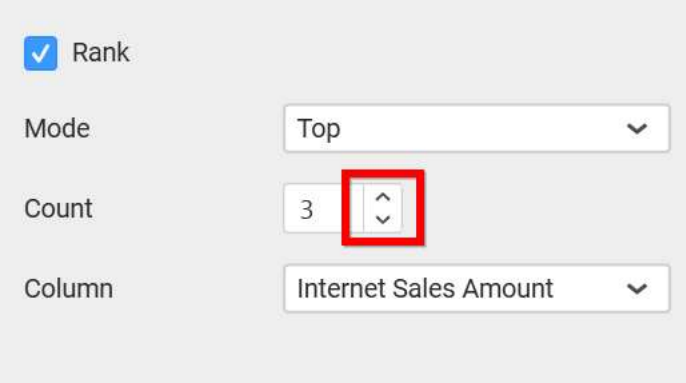


Select Filter(s)... to launch the Filters window.

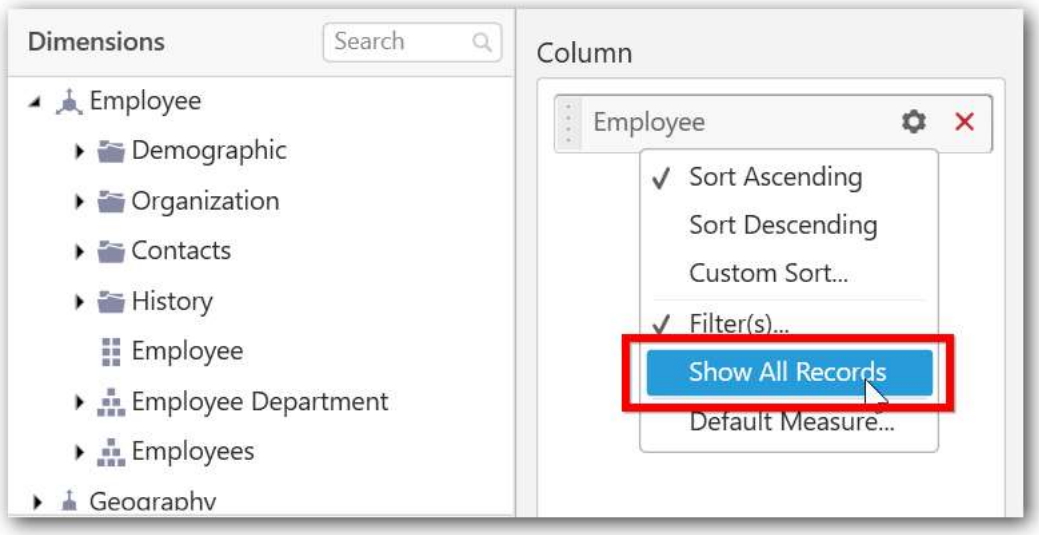
Define the filter Condition and Rank and Click OK.



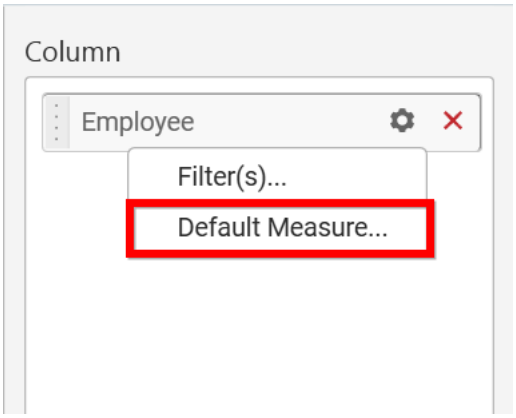


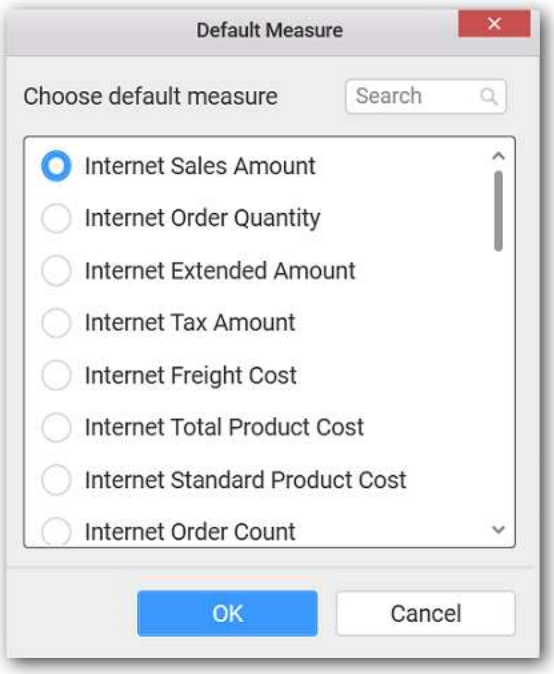


To show all records again click on Show All Records.



You can add default measure to the dropped dimension to retrieve exact result for that dimension.



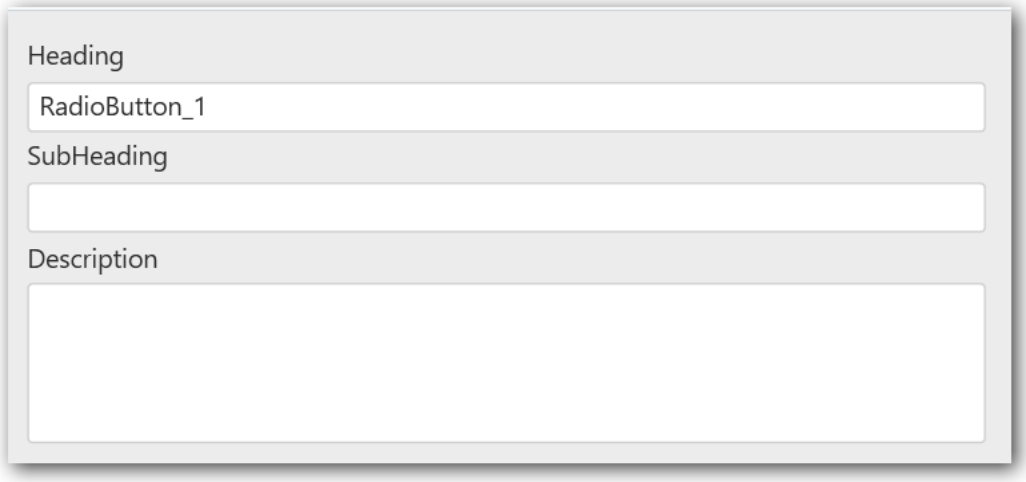


Here is an illustration,

How to format Radio Button?

You can format the Radio Button for better illustration of the view that you require, through the settings available in Properties pane.

General Settings



Header

This allows you to set title for this radio button widget.

SubHeading

This allows you to set sub-title for this radio button widget.

Description

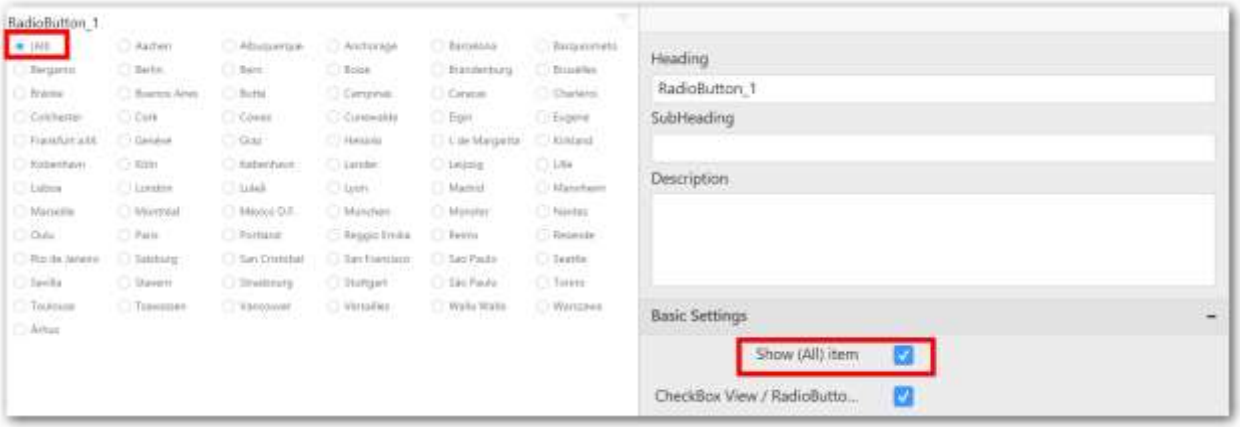
This allows you to set description for this radio button widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

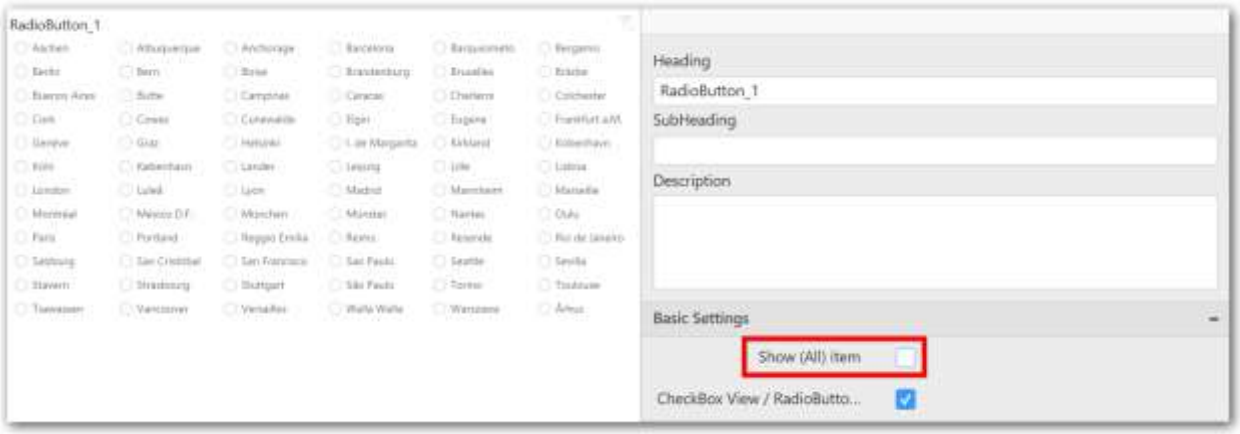
Show (All) item

This allows you to enable selection or deselection of entire items by adding All item.

Radio Button with Show All



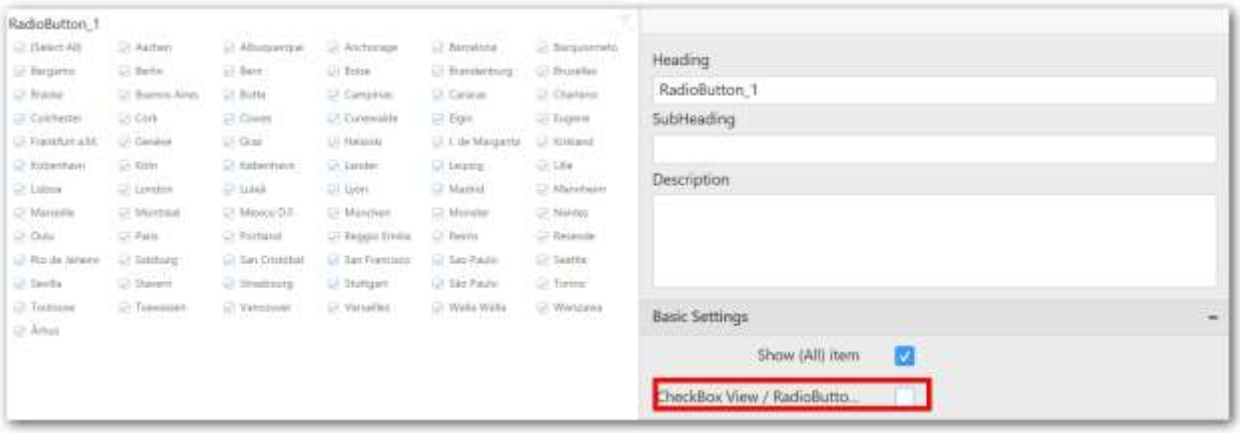
Radio Button without All



CheckBox View/RadioButton View

This allows you to change the view of the widget to checkbox.

Radio Button as Check Box View



Filter Settings

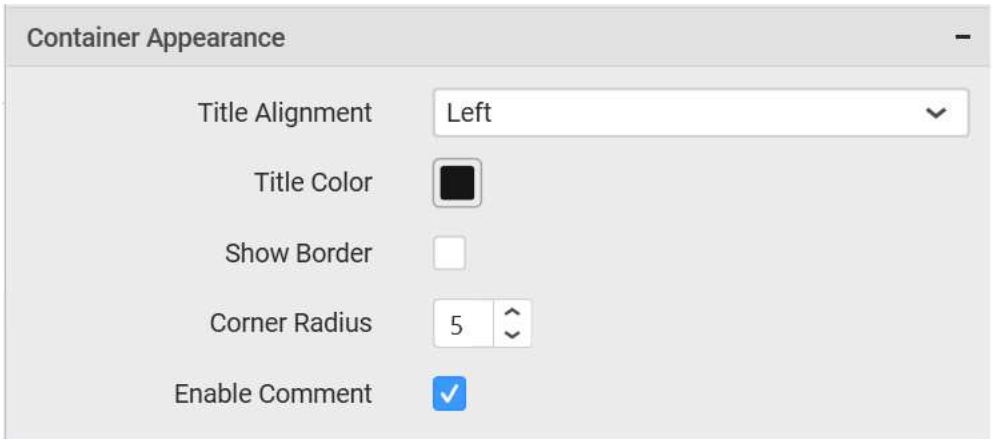
Act as Master Widget

This allows you to define this radio button widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this radio button widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

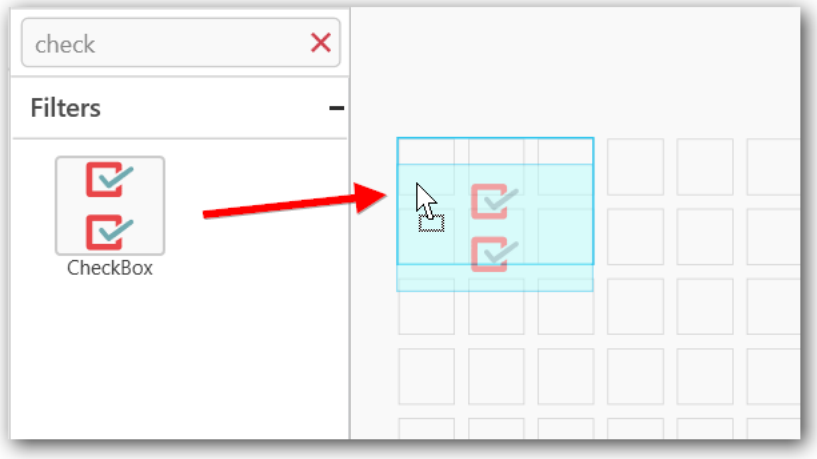
Check Box

Checkbox enables you to filter based on single or multiple items selection in a group. To configure a checkbox, a minimum requirement of 1 column is needed.

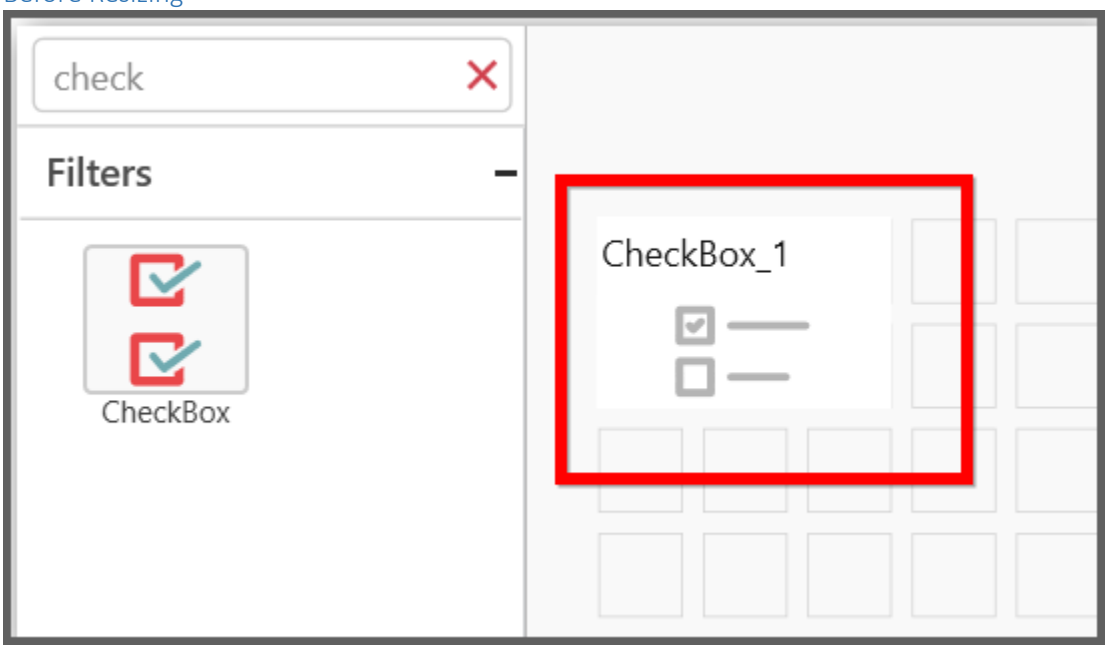
How to configure flat table data to Check Box?

The following procedure illustrates data configuration of Checkbox.

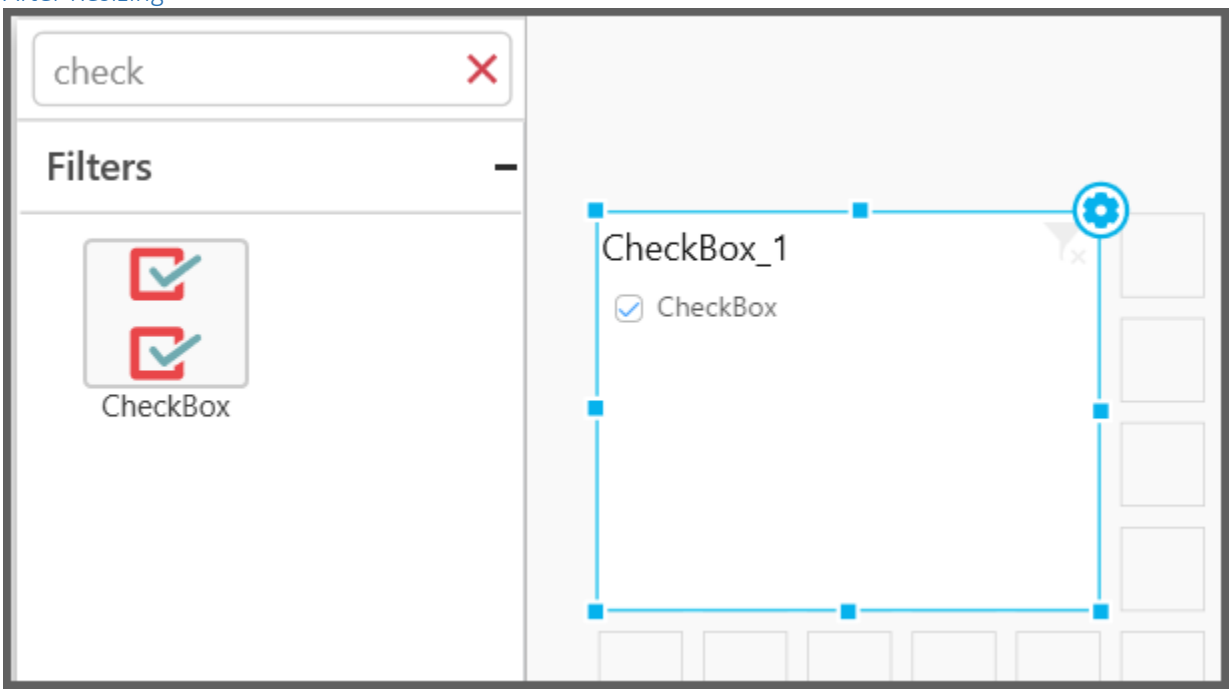
Drag and drop CheckBox widget from the Toolbox into design panel and resize into your required size. You can find widget in Toolbox by search.



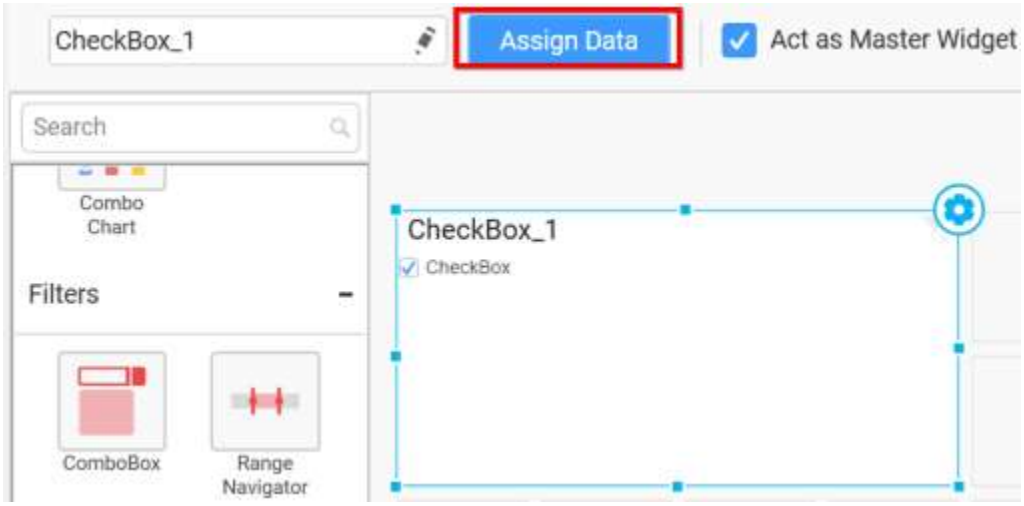
Before Resizing



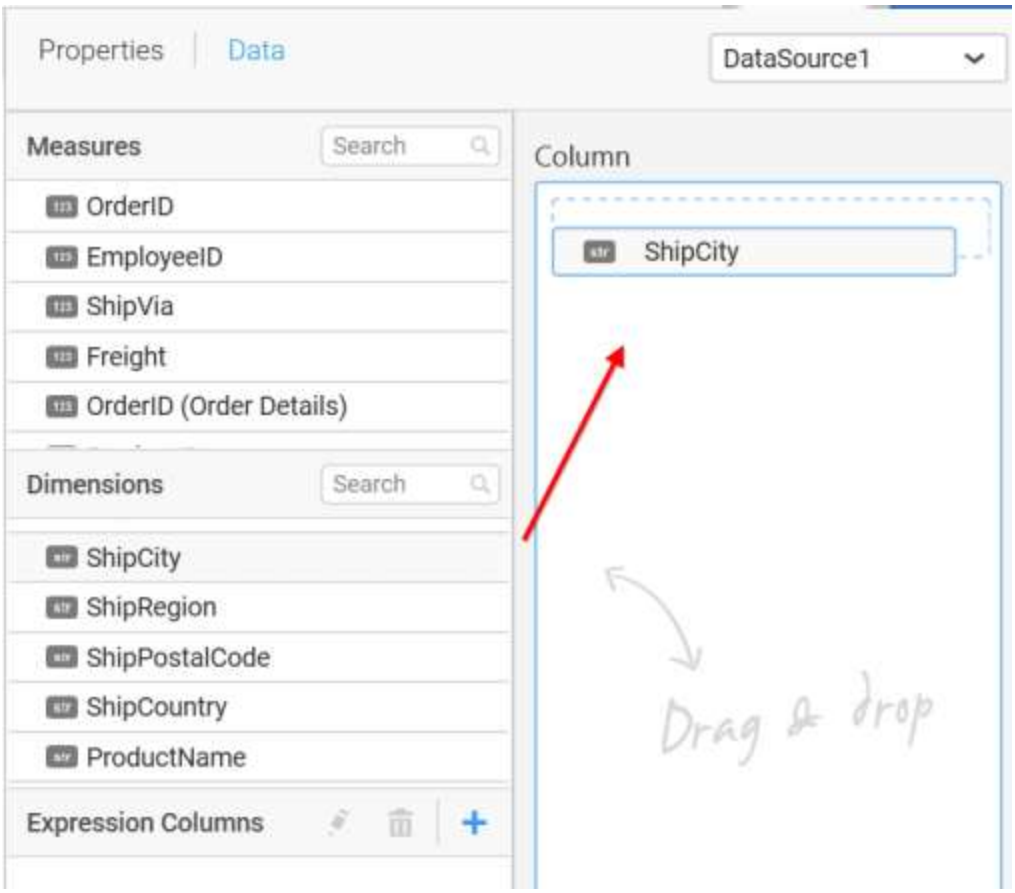
After Resizing



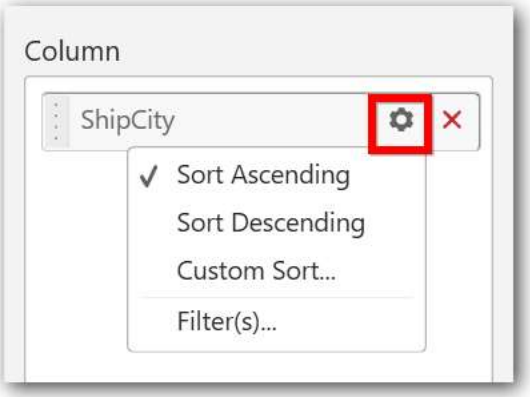
Select the dropped widget using mouse and click the Assign Data button at Design Tools Pane to open the Data configuration pane.



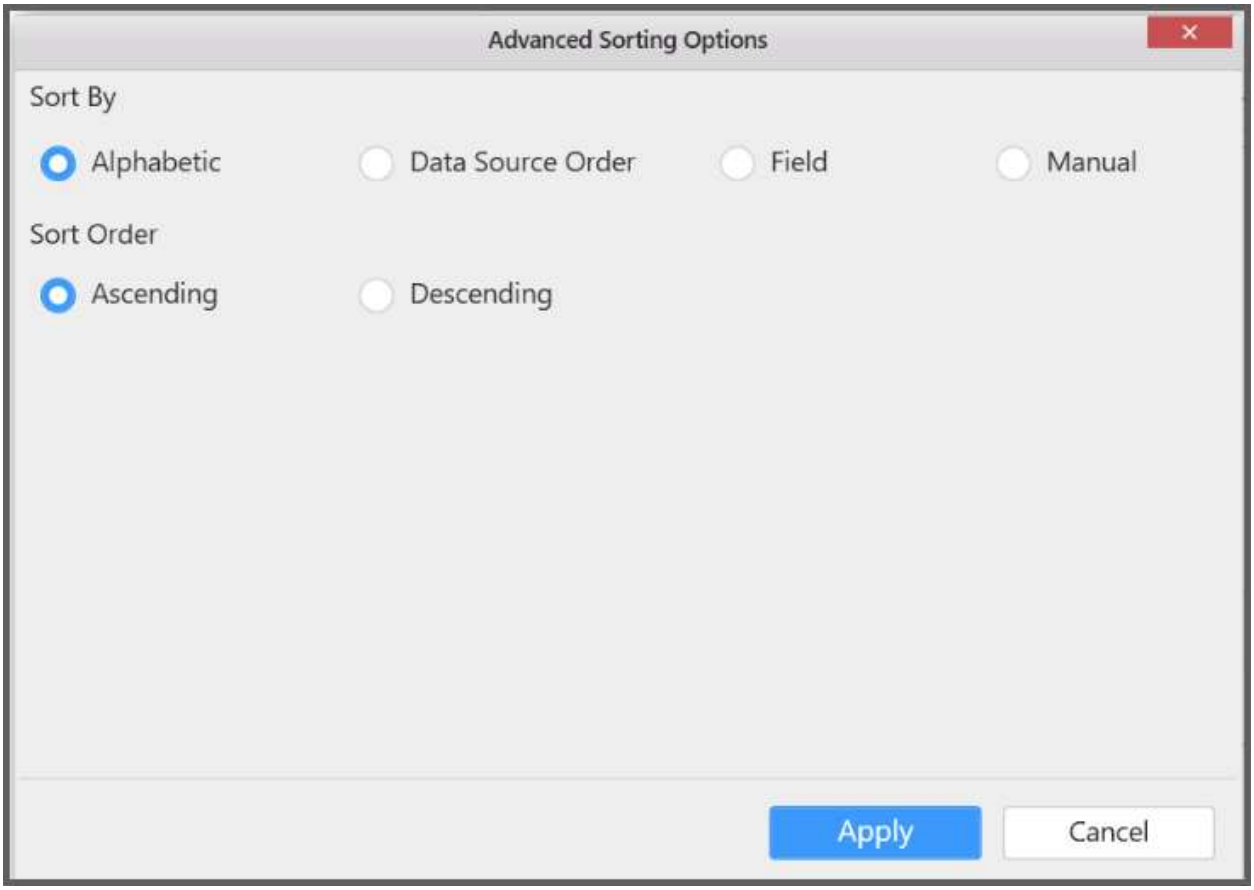
Drag and drop a column from Measures or Dimensions or Expression Columns category to Column section.



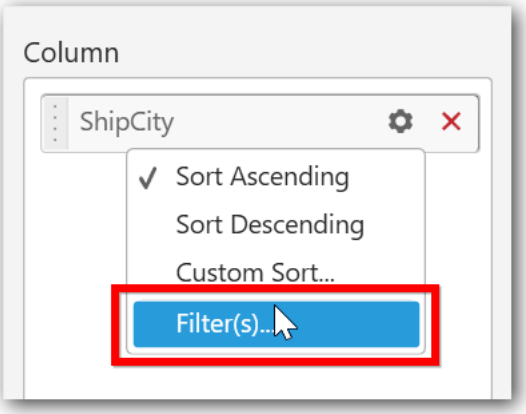
Define the sort order of the dropped column through the Settings drop down menu.



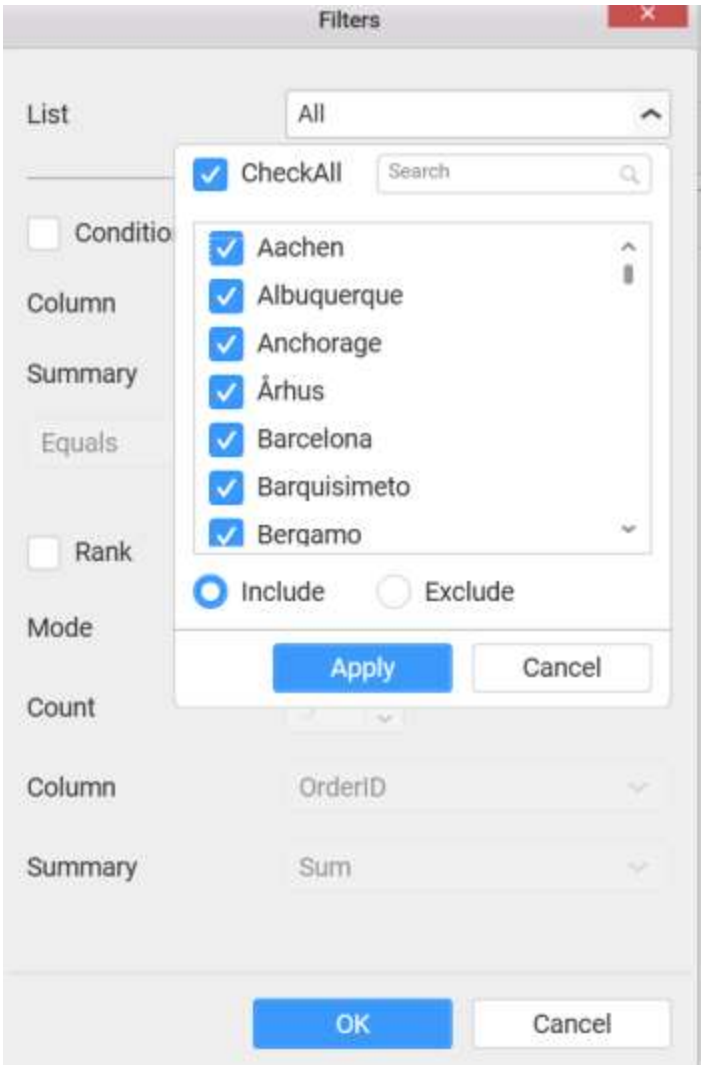
Custom sort



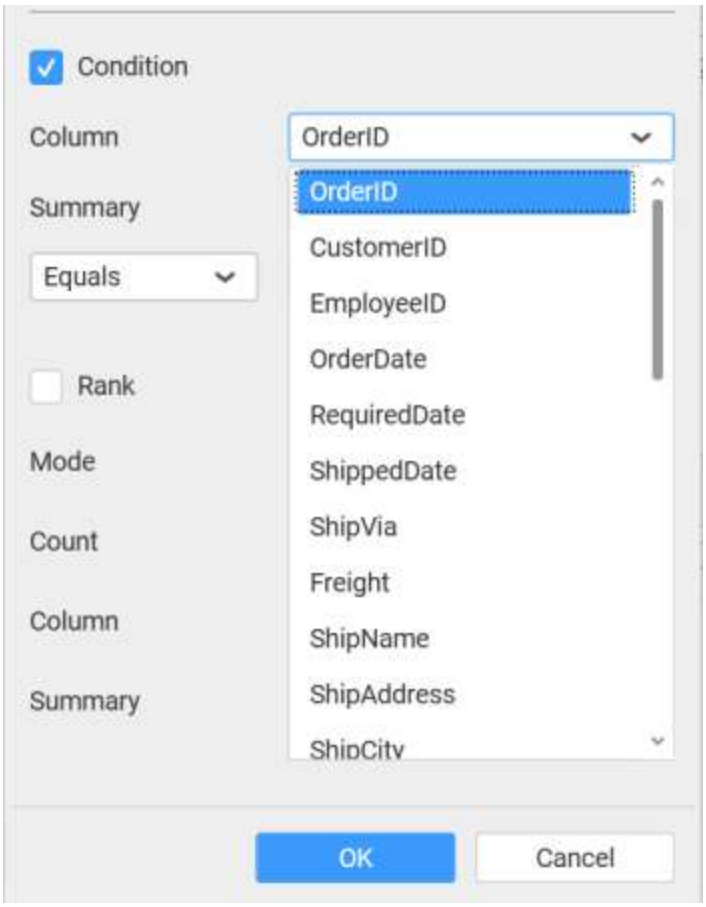
Define filter criteria through the Filter(s)... menu item in the Settings drop down menu.

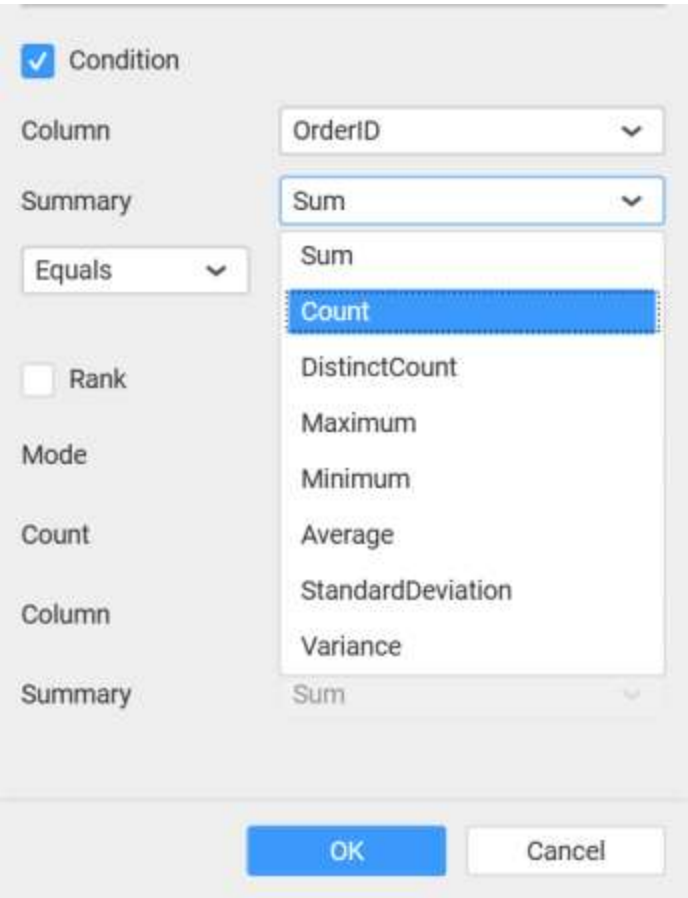


Select the specific item to filter the element and CheckAll is used either to check all the item or to select the specific item. Include and Exclude is used to include and exclude the selected elements. Click the Apply button to apply the selection.

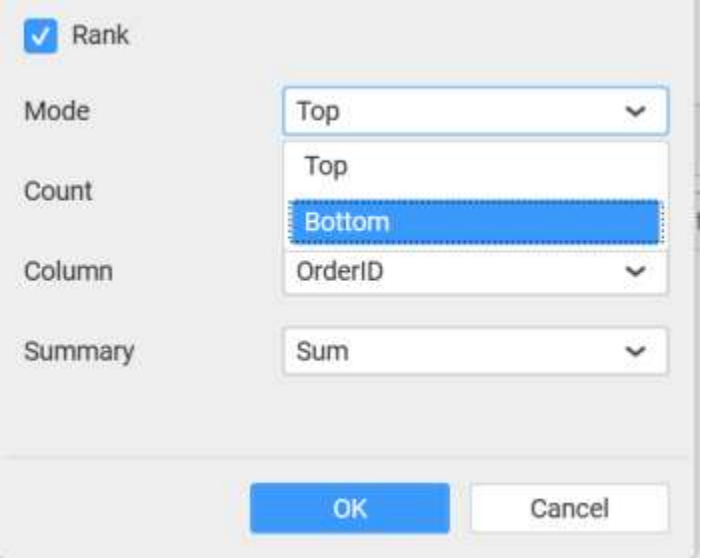


Select the Condition option to change the Column elements and Summary type by selecting the required column name and summary type.

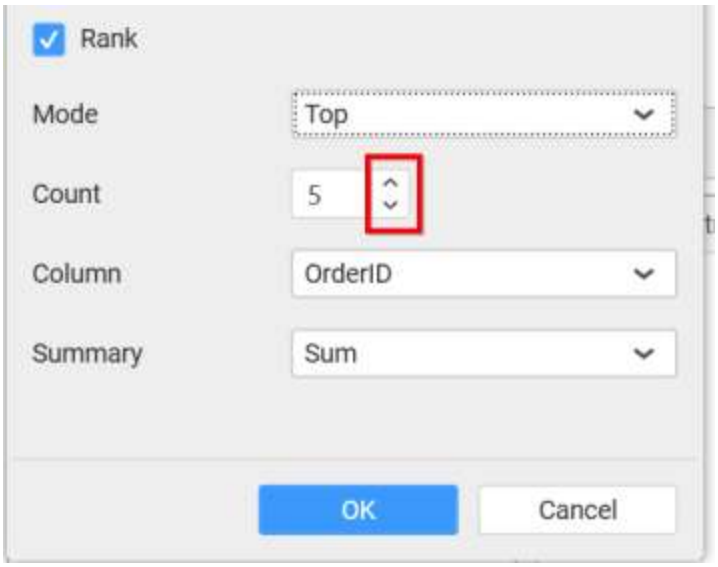




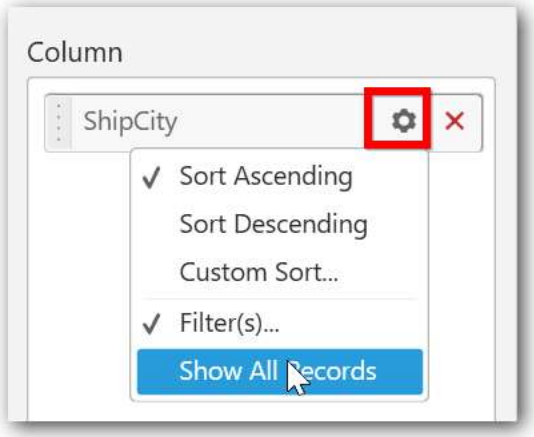
Select the Rank option to enable filters and select the Mode as either top or bottom.



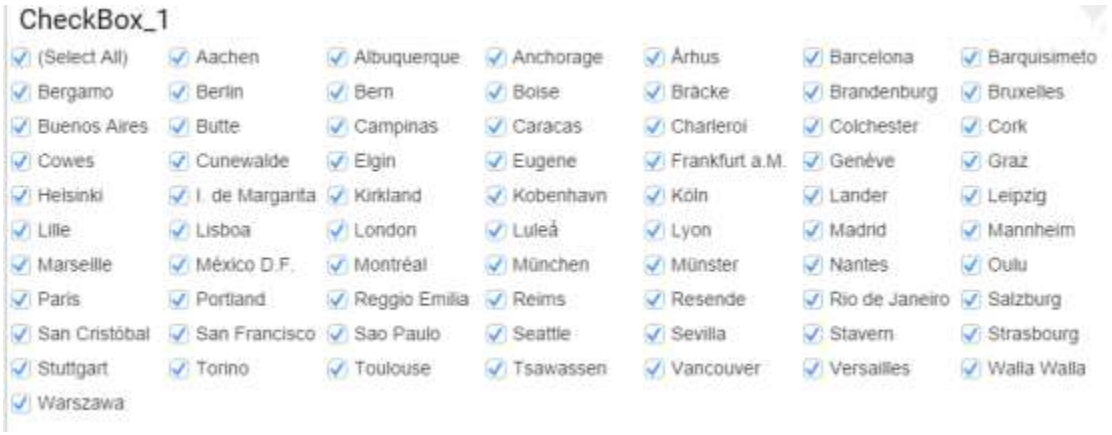
You can change the Count value to filter the top elements and change the column and summary type as required and click OK button.



Clear the filters by selecting the Show All Records in the Settings dropdown menu.



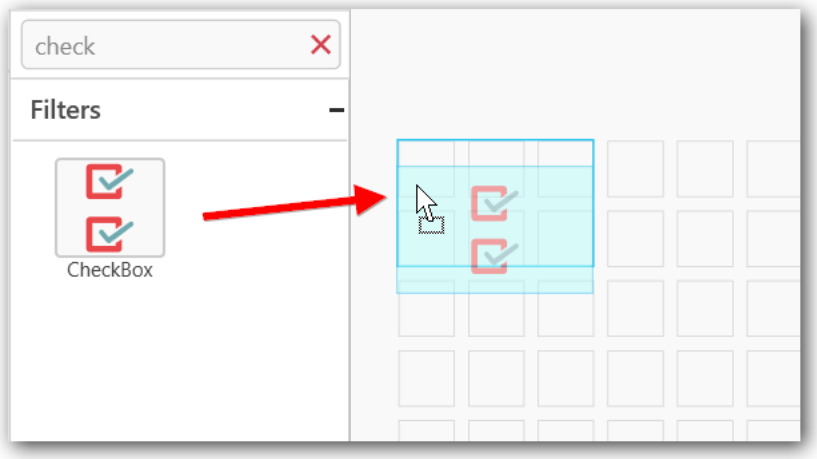
Here is an illustration,



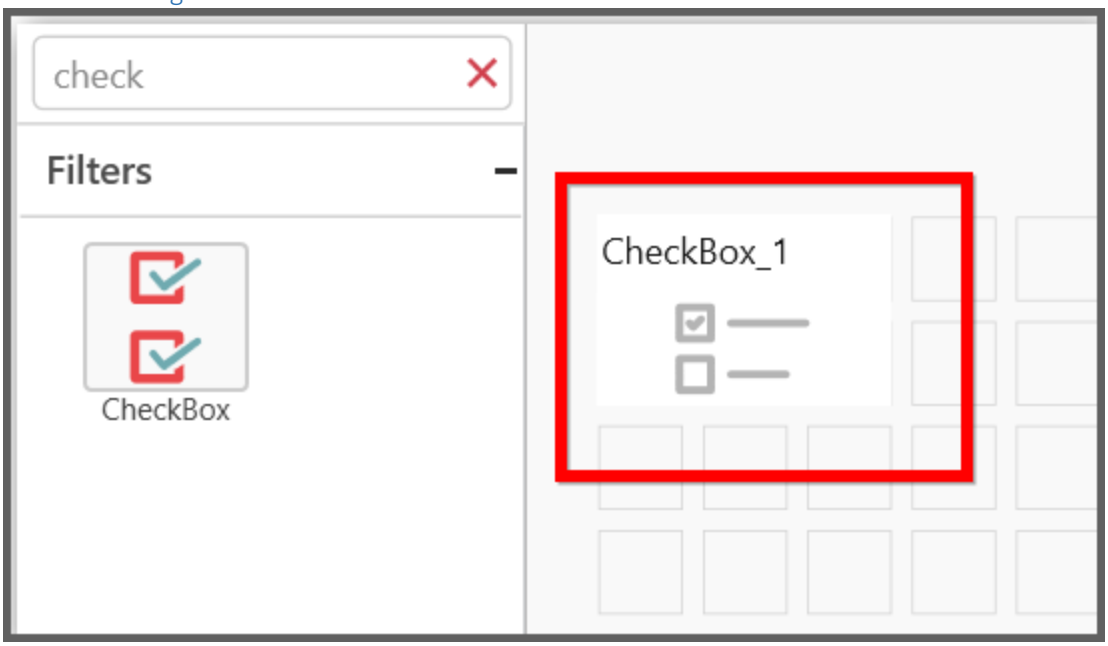
How to configure the SSAS data to Check Box?

Following steps illustrates configuring SSAS data to Checkbox.

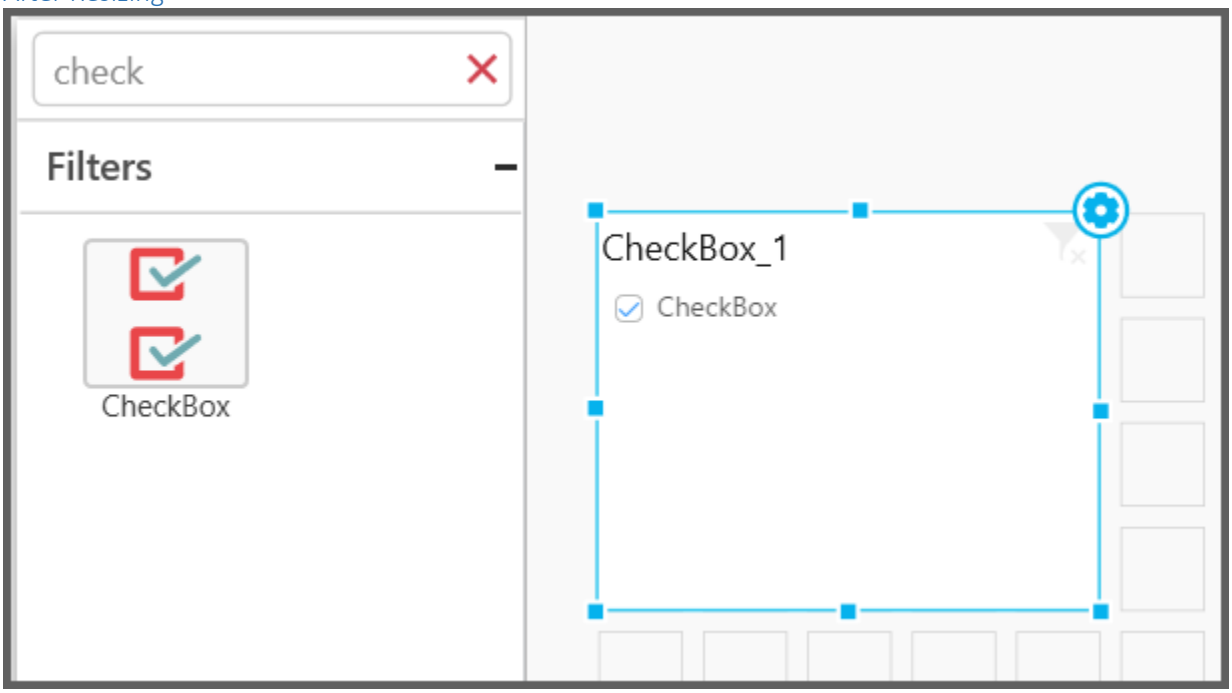
Drag and drop CheckBox widget from the Toolbox into design panel and resize into your required size. You can find widget in Toolbox by search.



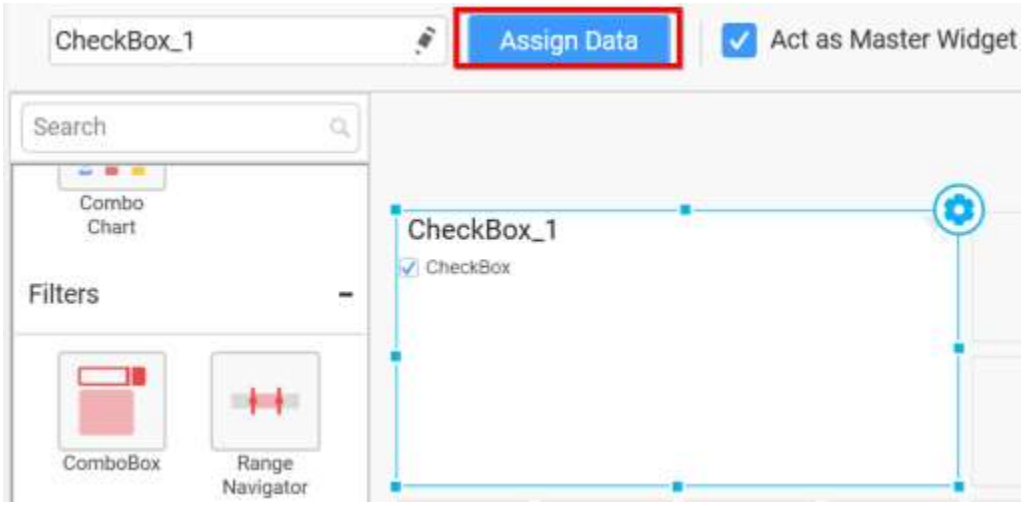
Before Resizing



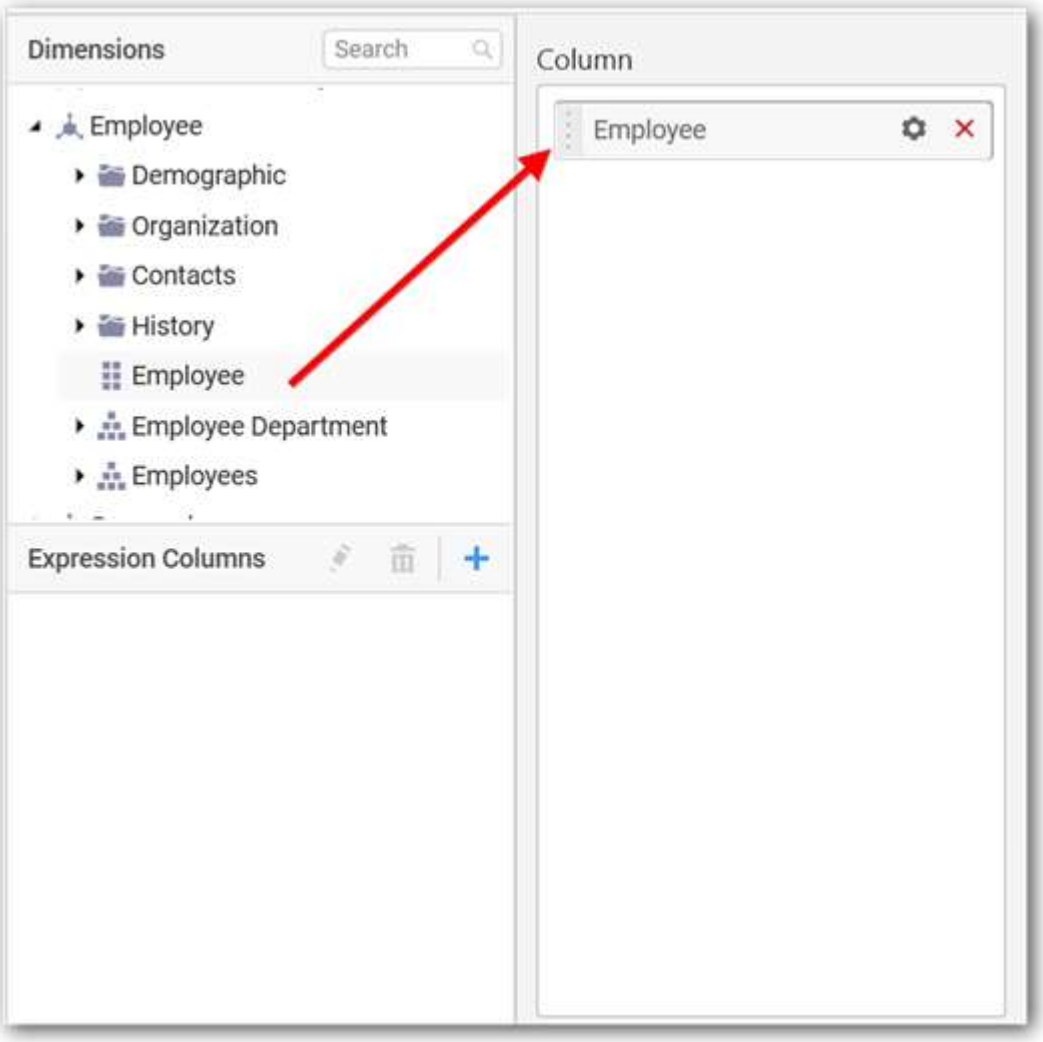
After Resizing



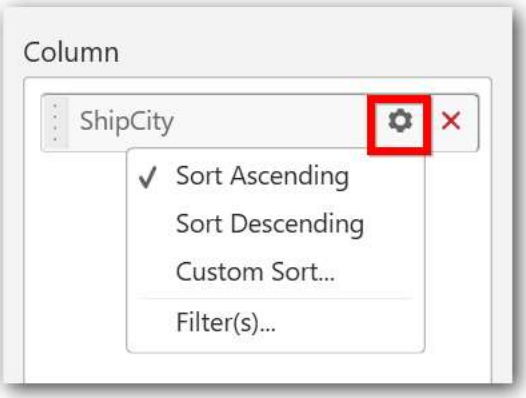
Select the dropped widget using mouse and click the Assign Data button at Design Tools Pane to open the Data configuration pane.



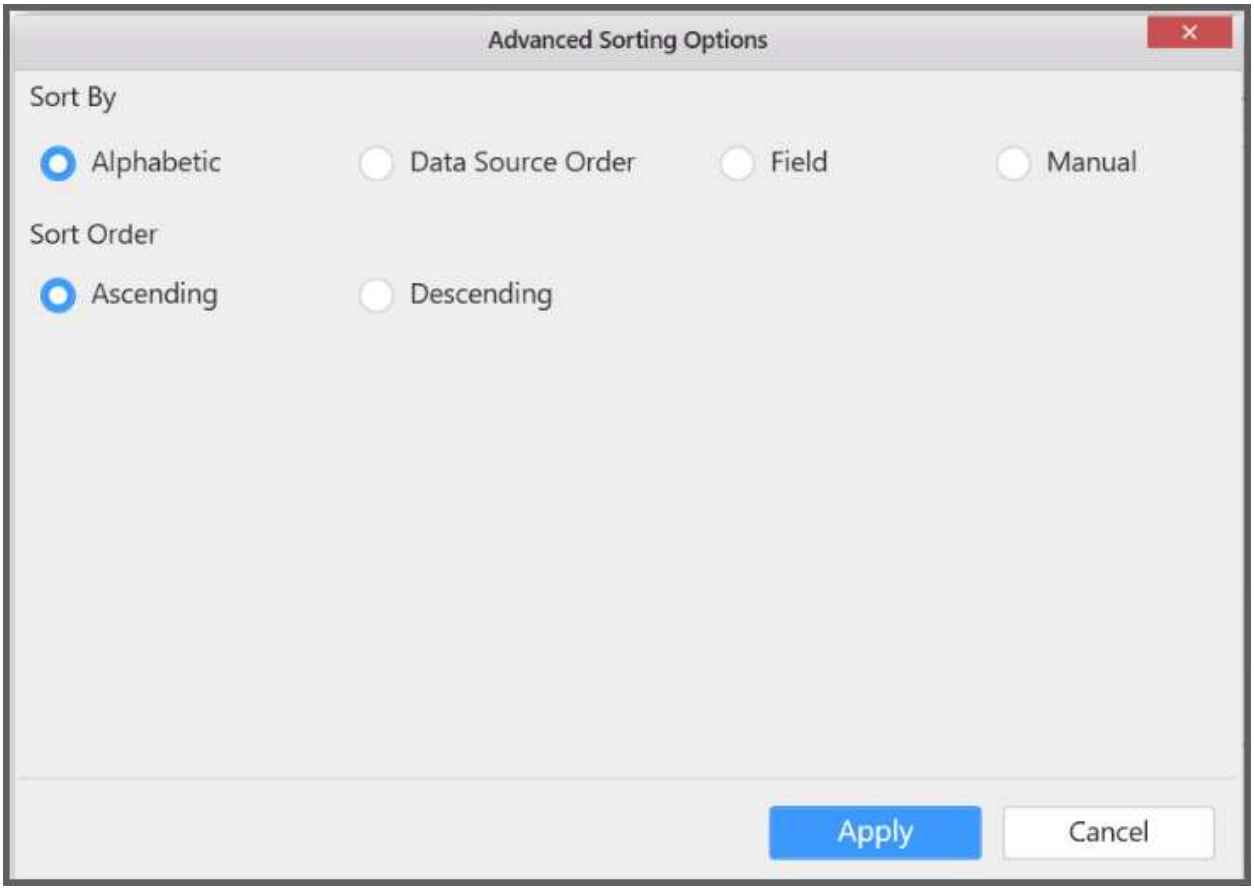
Drag and drop a dimension level or hierarchy column under Dimensions category into Column section.



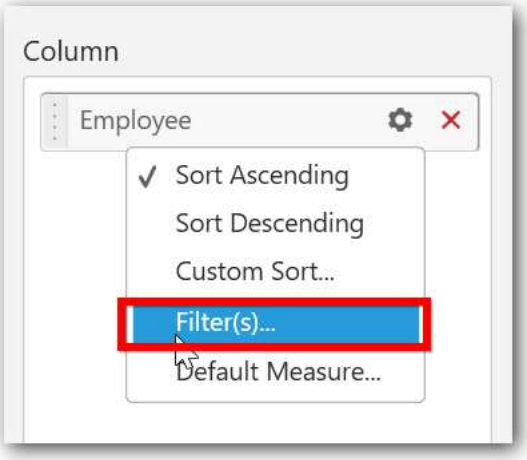
Define the sort order of the dropped column through the Settings drop down menu.



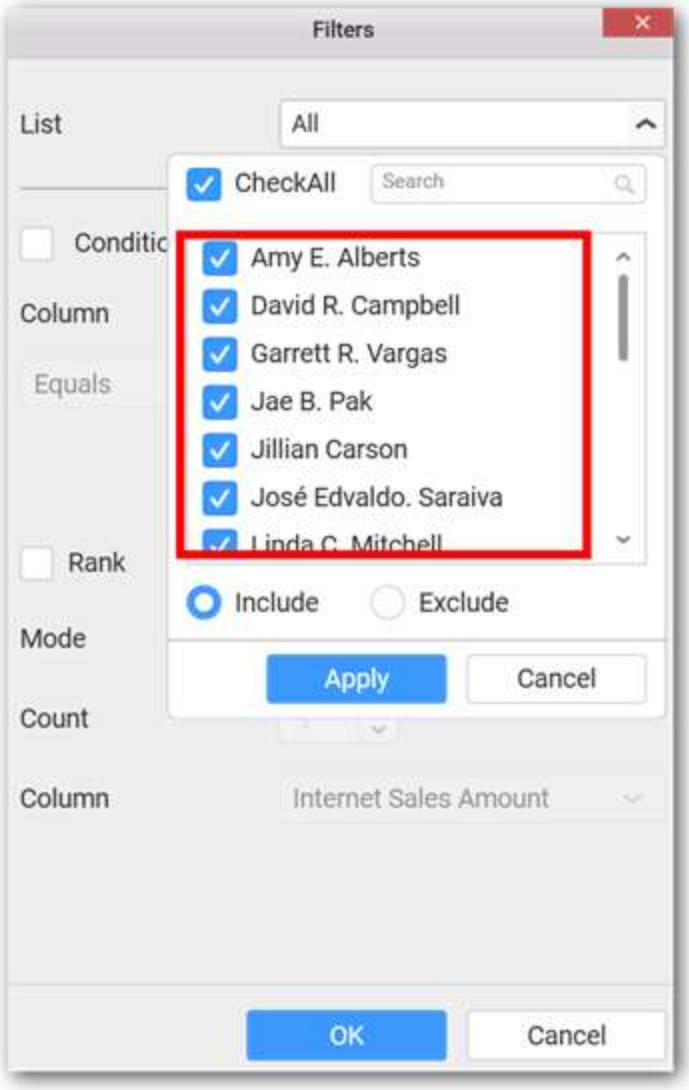
Custom sort



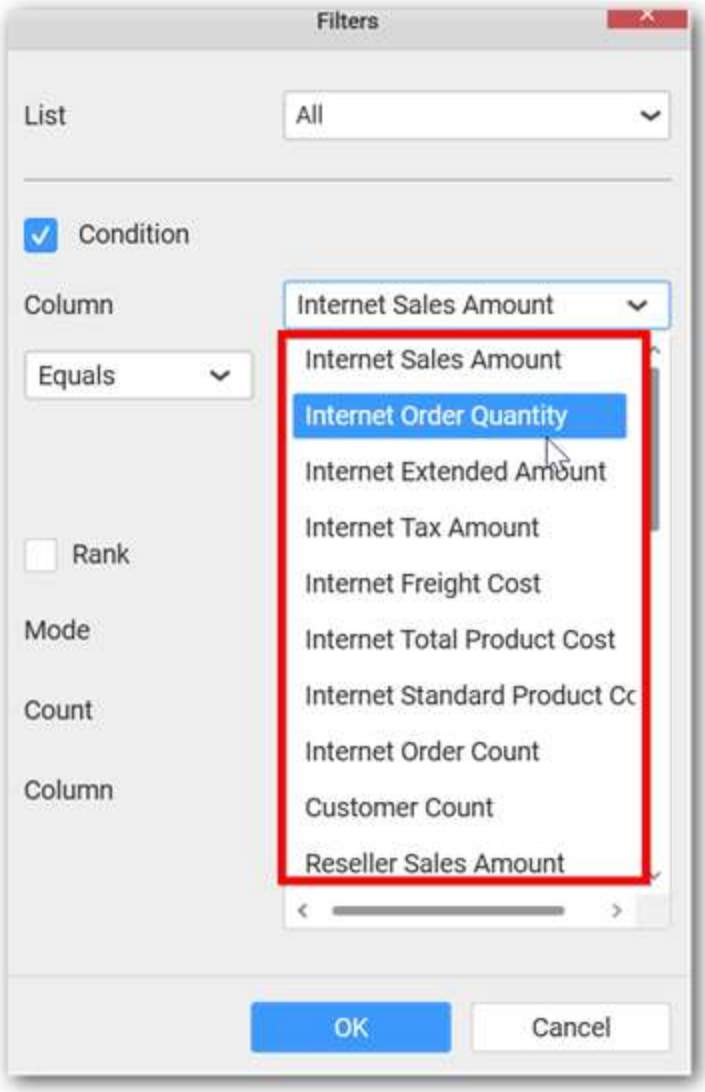
Define filter criteria through `Filter(s)...` menu item in the Settings drop down menu.



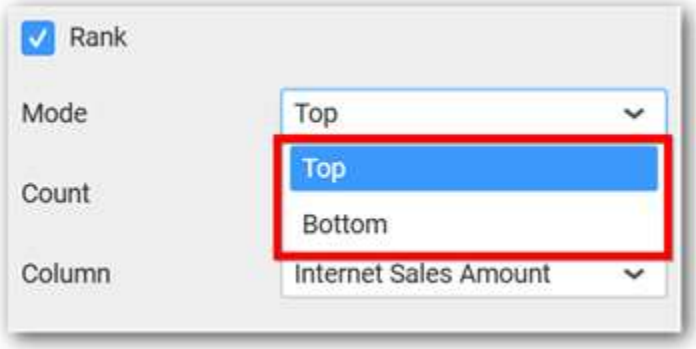
Select the specific item to filter the element and `CheckAll` is used either to check all the item or to select the specific item. `Include` and `Exclude` is used to include and exclude the selected elements. Click the `Apply` button to apply the selection.



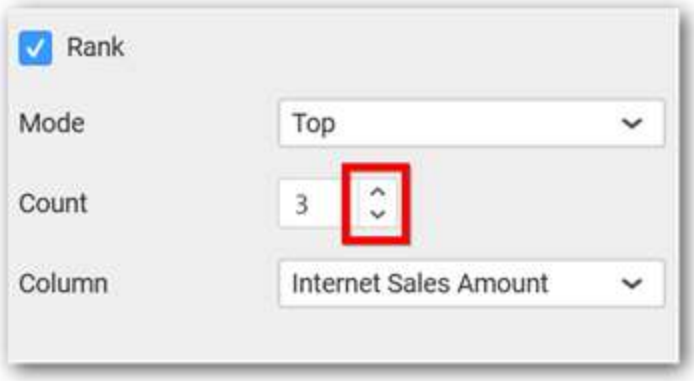
Select the Condition option to change the Column elements by selecting the required column name.



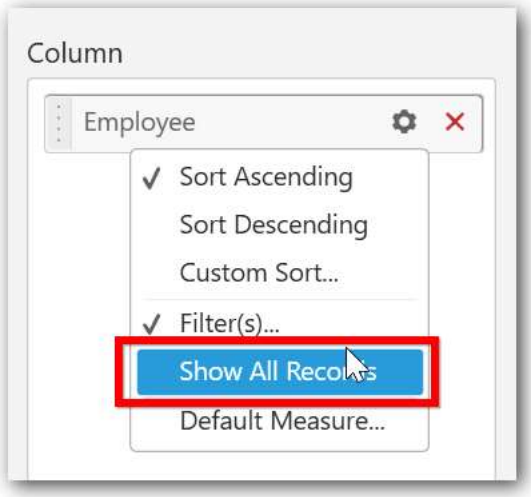
Select the Rank option to enable filters and select the Mode as either top or bottom.



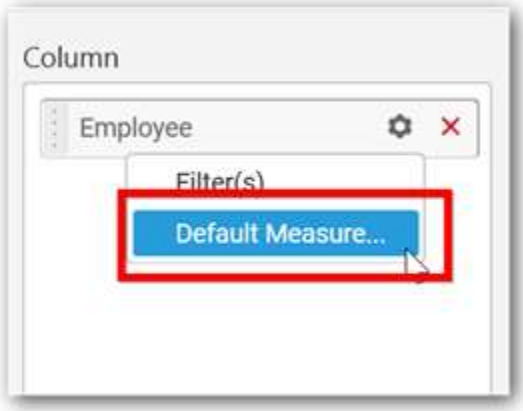
Select the Count value to filter the top elements and change the column as required and click OK button.

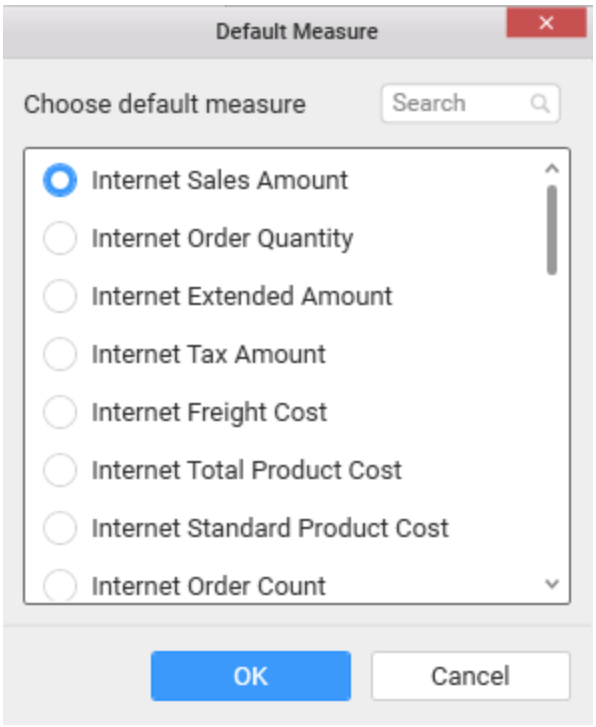


Clear the filters by selecting the Show All Records in the Settings dropdown menu.

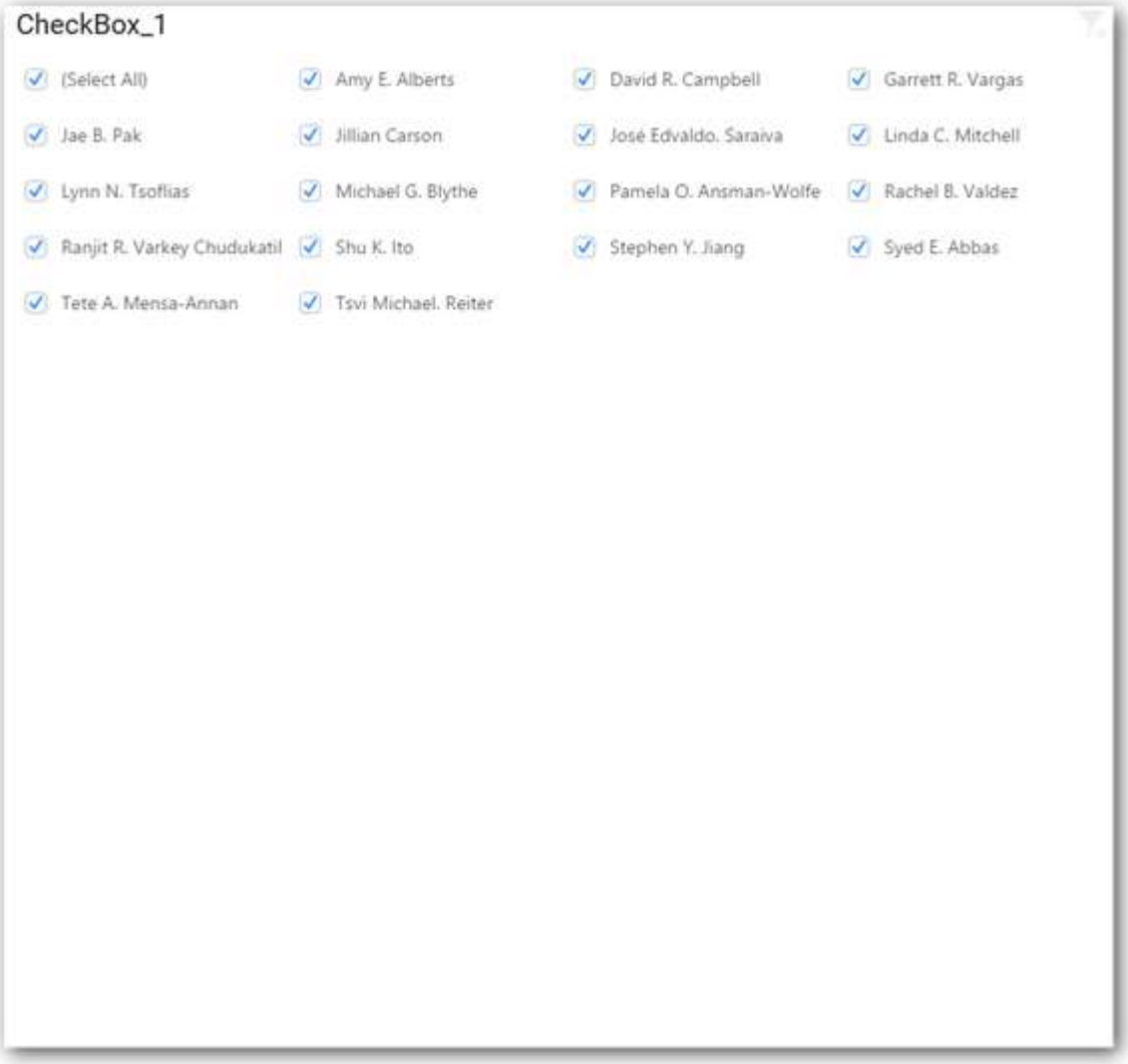


Define a default measure to the dropped dimension against which dimension values need to be categorized.





Here is an illustration,



How to format Check Box?

You can format the Check box for better illustration of the view that you require, through the settings available in Properties pane.

General Settings

Header

This allows you to set title for this check box widget.

SubHeading

This allows you to set sub-title for this ComboBox widget.

Description

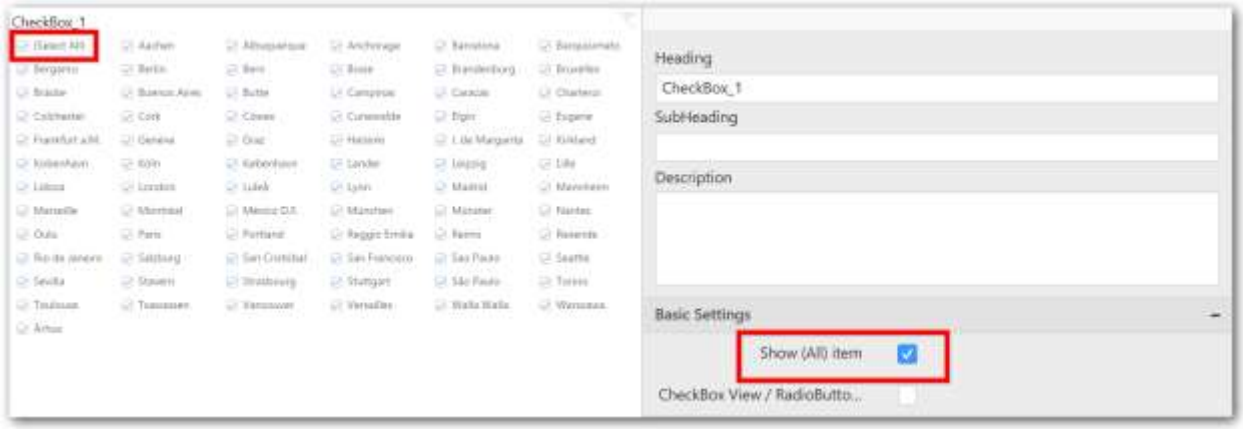
This allows you to set description for this check box widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

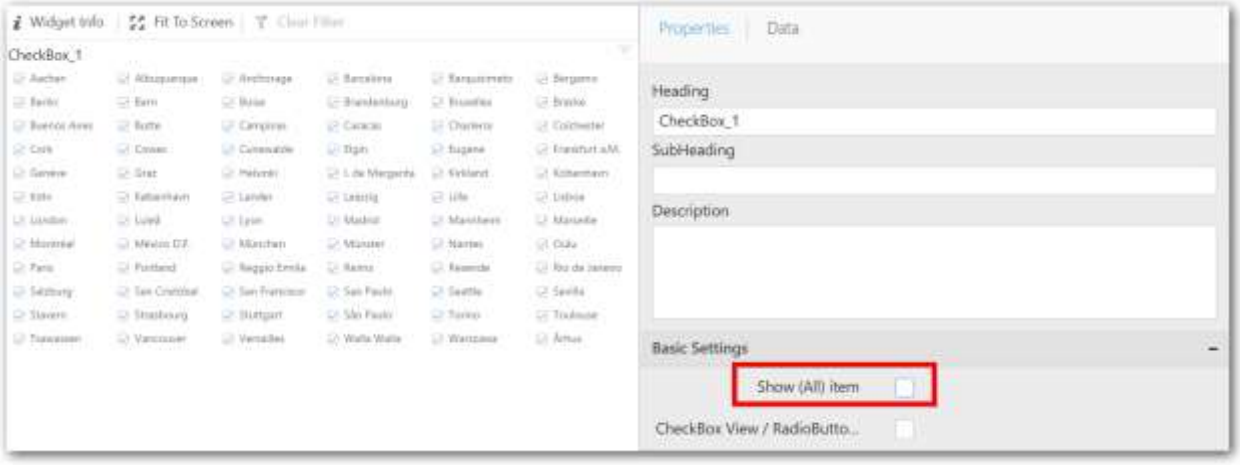
Show (All) item

This allows you to enable selection or deselection of entire items by adding Select All item.

Check Box with Select All



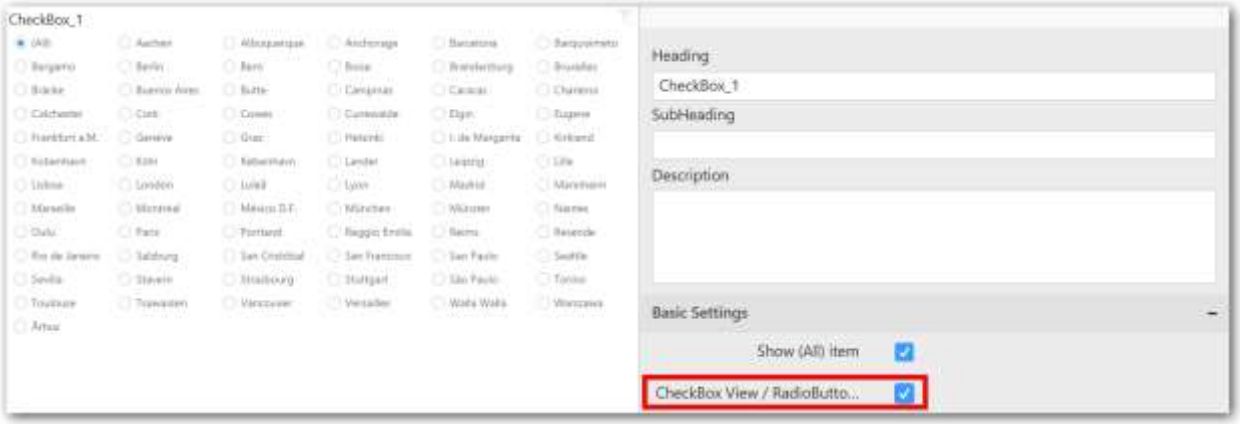
Check Box without Select All



CheckBox View/ RadioButton View

This allows you to change the view of the widget to radio button.

Check Box with Radio Button View



Filter Settings

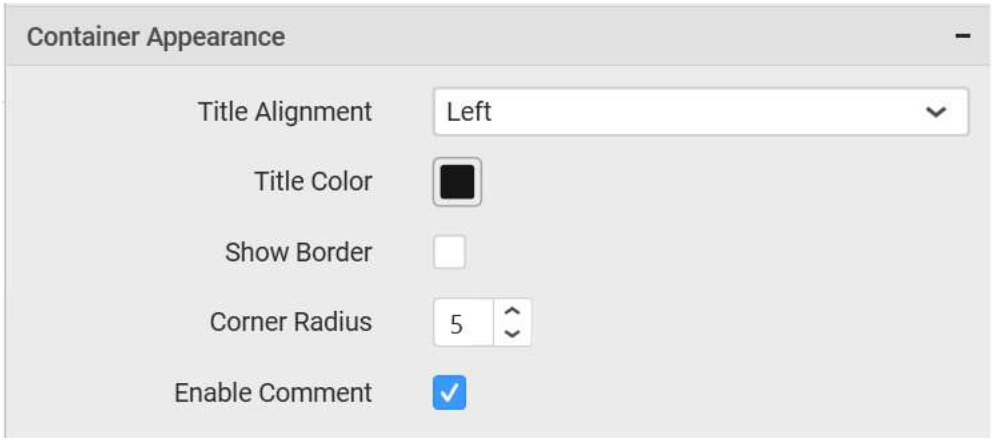
Act as Master Widget

This allows you to define this check box widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this check box widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Appearance



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comment

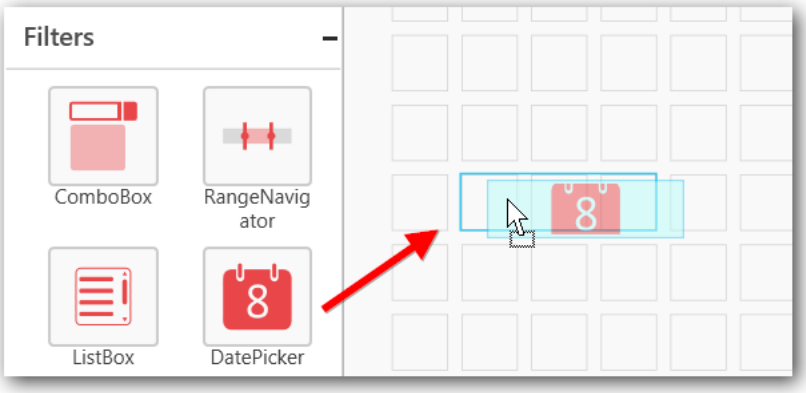
This allows you to enable/disable comment for dashboard widget. For more details refer here

Date Picker

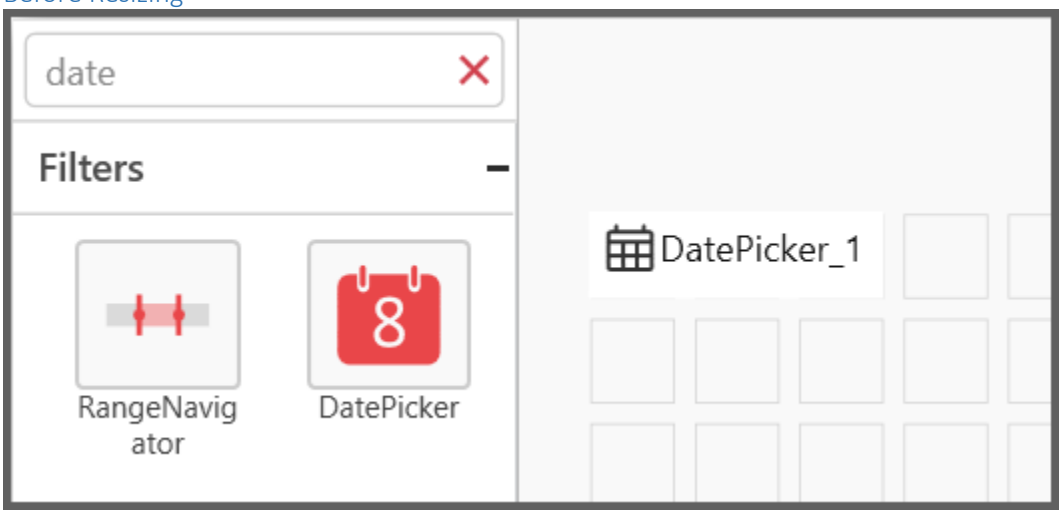
Date Picker enables you to filter based on the single or range of date selection.

How to configure flat table data to date picker?

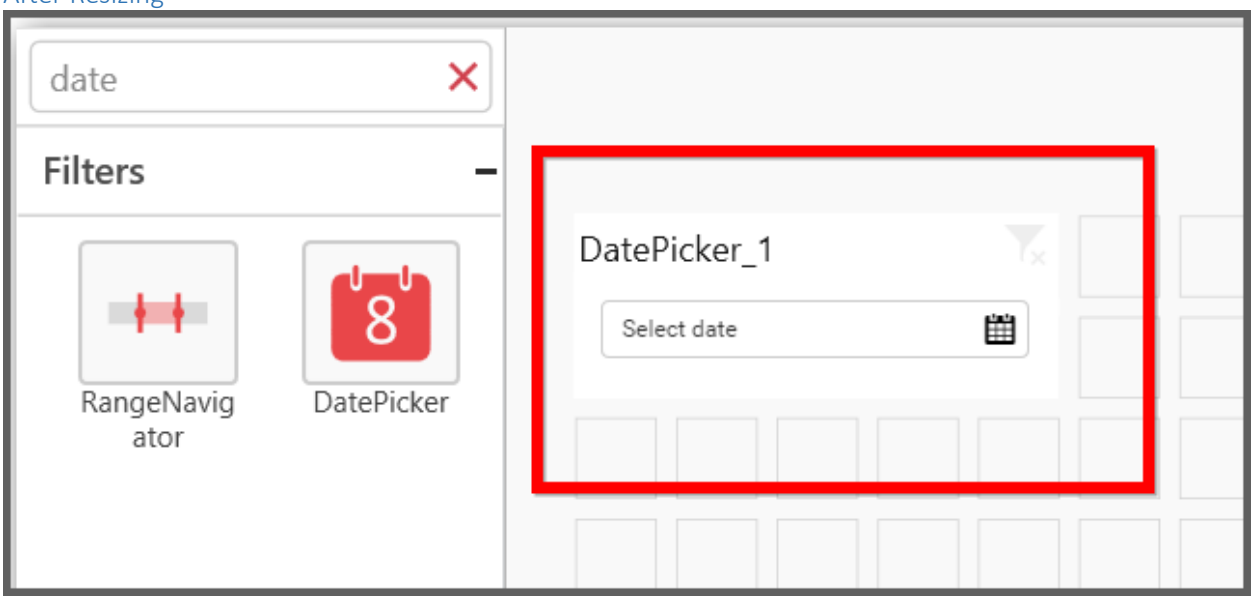
Drag and drop the Date Picker from toolbox at left into design canvas and resize it to your required size.



Before Resizing

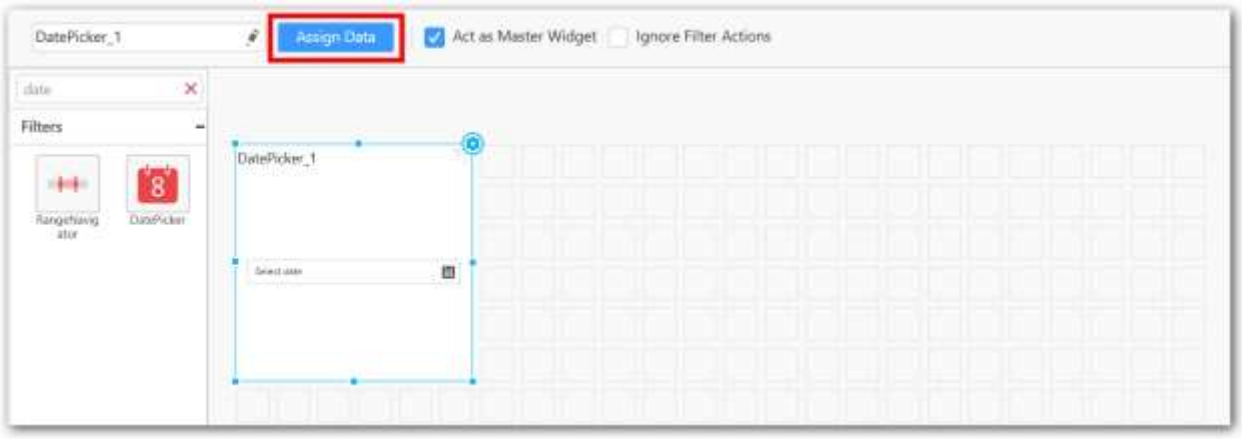


After Resizing

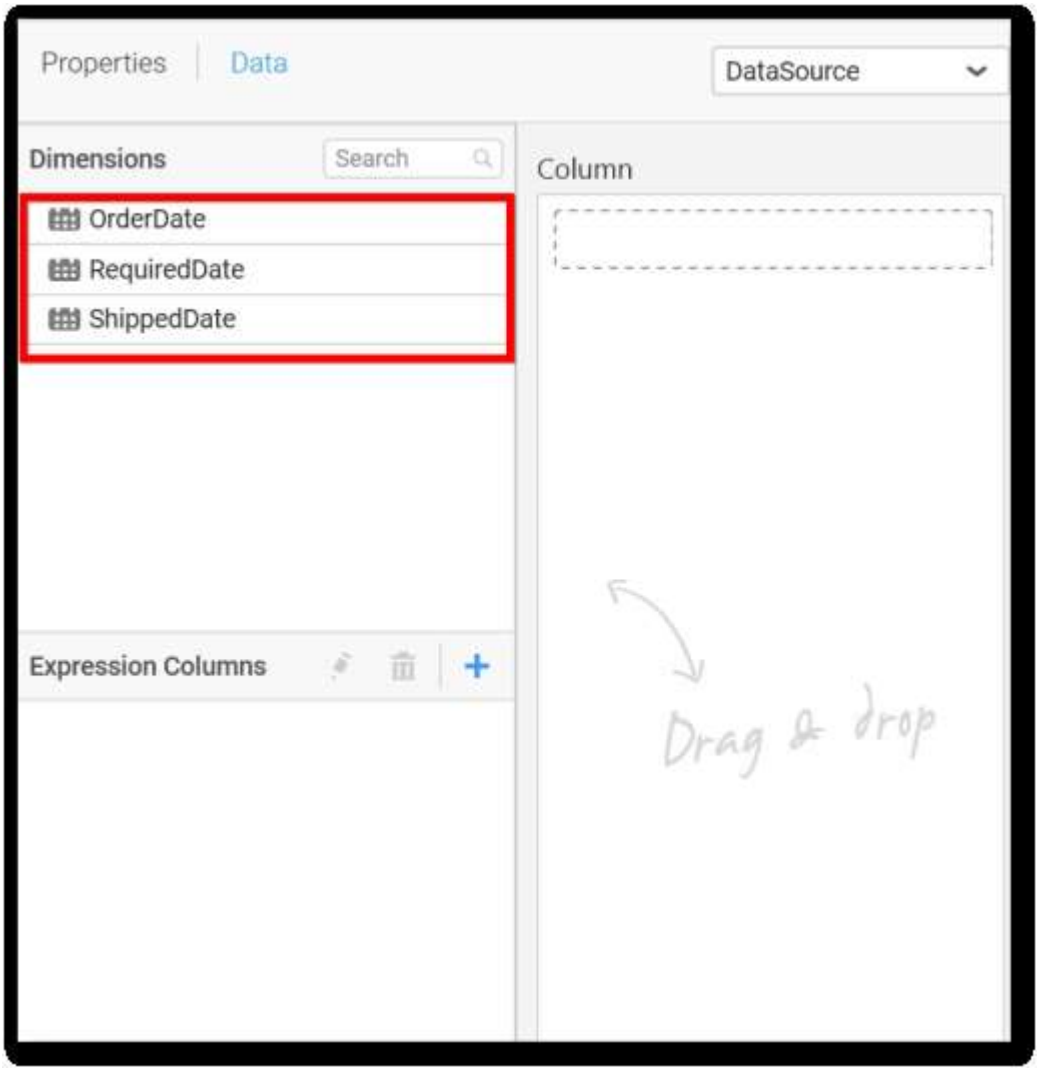


Keep the selection focus on the dropped date picker widget.

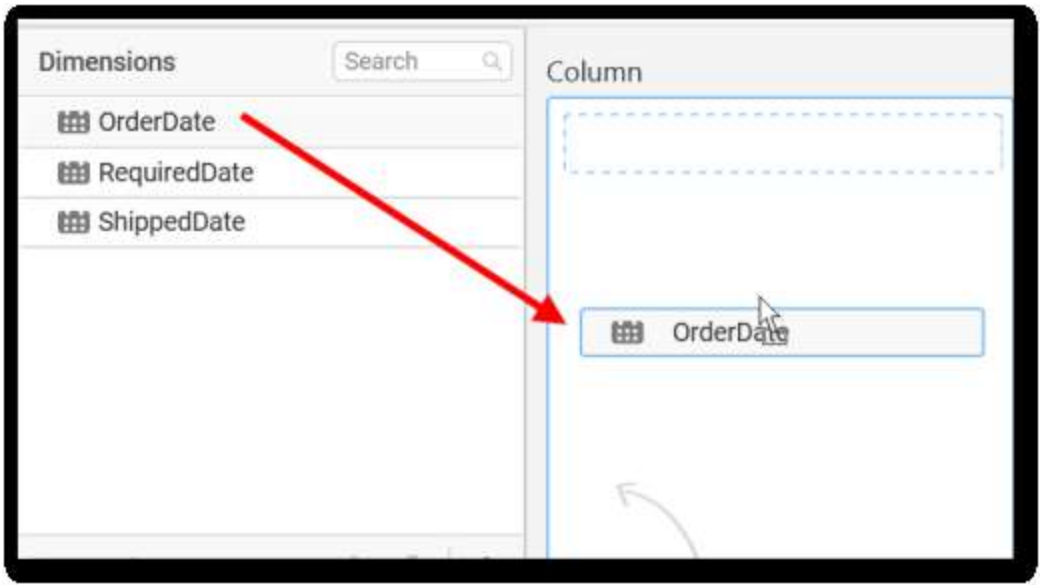
Click on Assign Data button in the tool bar.



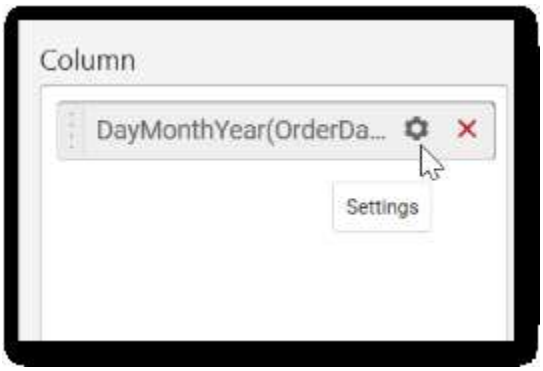
Now, the Data window will be opened with available date fields in Dimension from the connected data source.

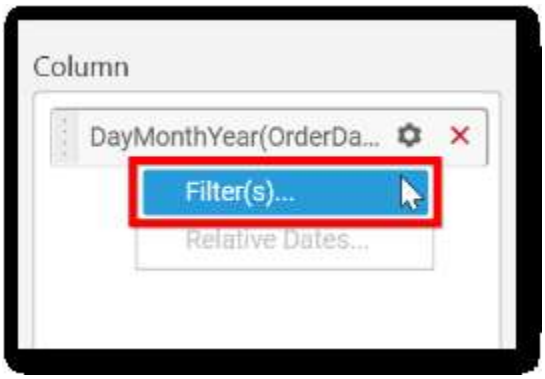


Drag and drop a date field from Dimensions into Column section.



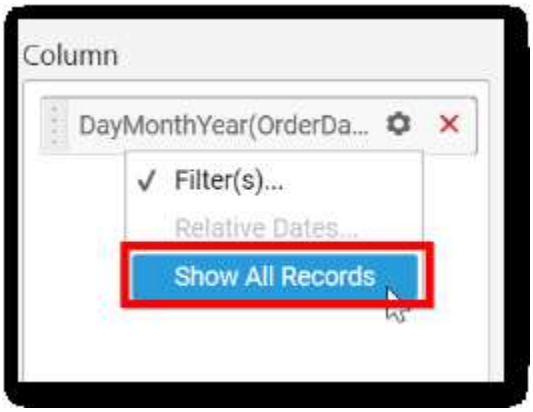
To filter out a specific range of dates, click the Settings icon in the dropped column and select Filter(s)... in the drop down menu.





Now the Range Filter window get displayed. Configure the date range to be filtered out and click Apply.

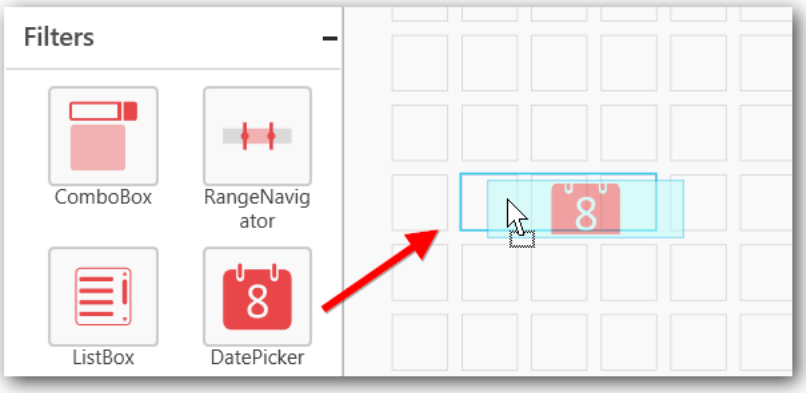
To clear the filter applied, click on Show All Records item in the drop down menu.



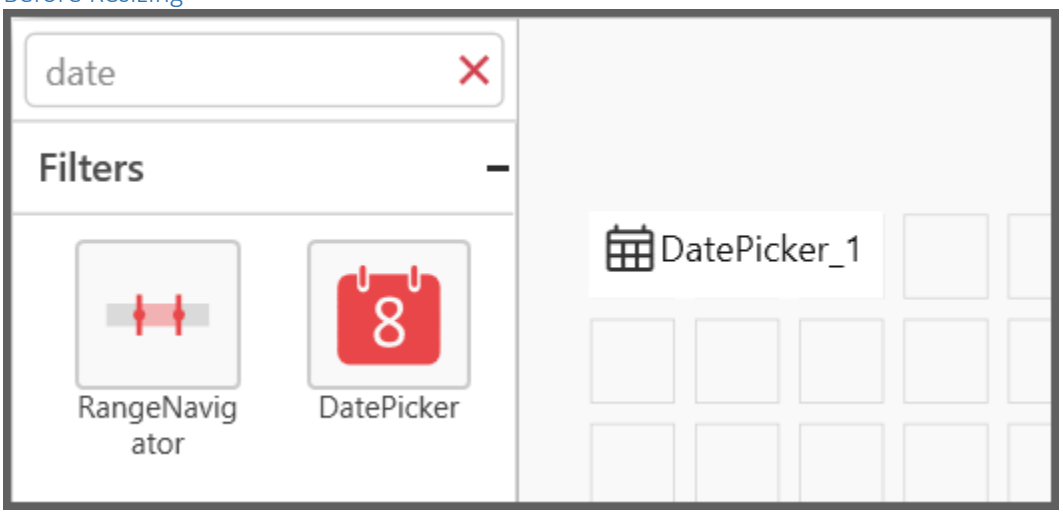
How to configure the SSAS data to date picker?

Following steps illustrates configuration of SSAS data to Date Picker.

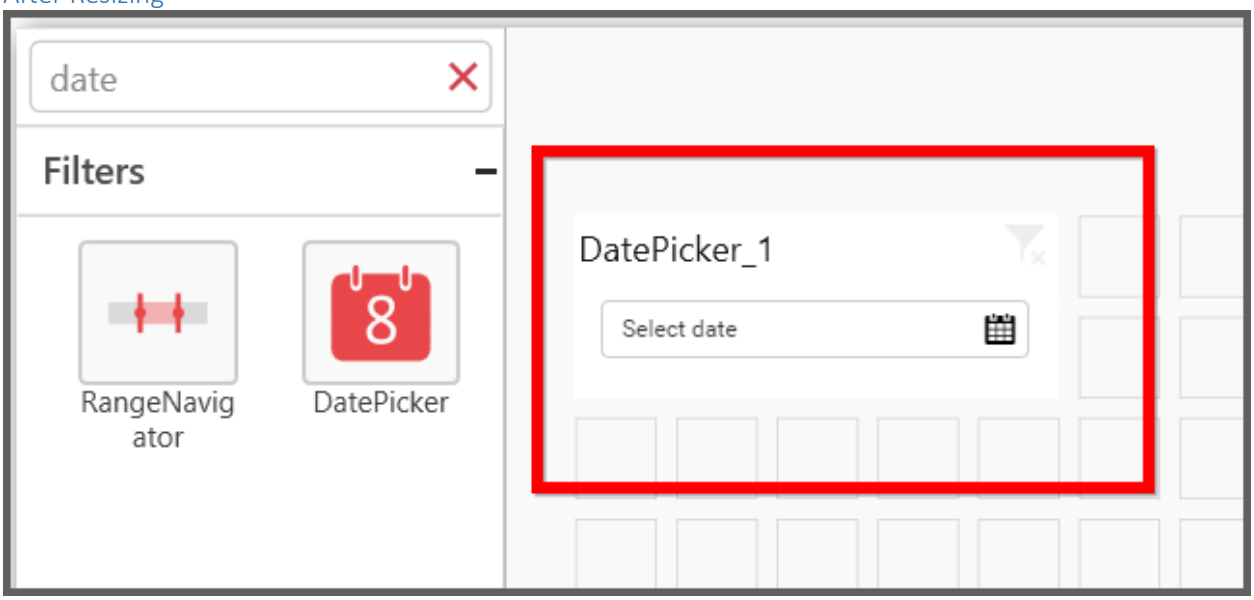
Drag and drop the Date Picker from toolbox at left into design canvas and resize it to your required size.



Before Resizing

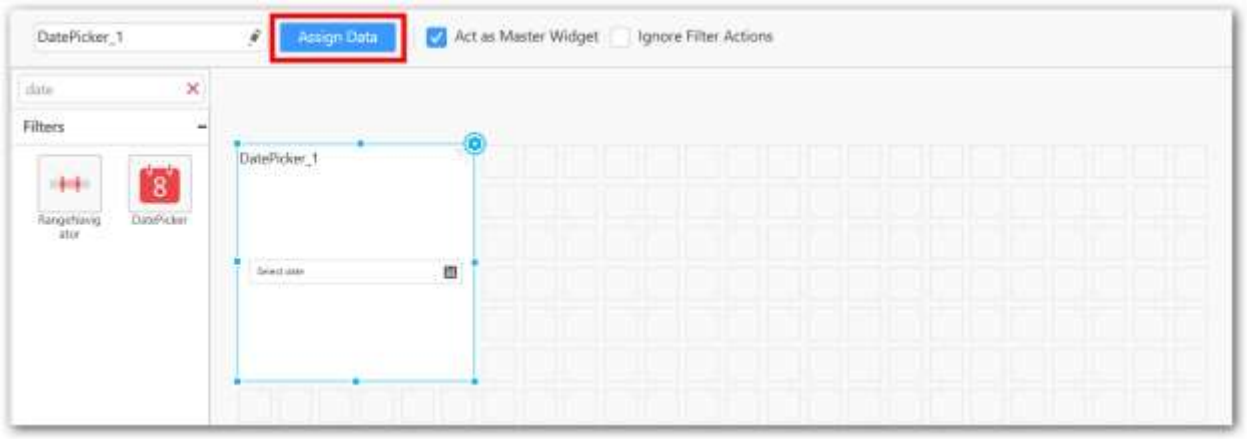


After Resizing



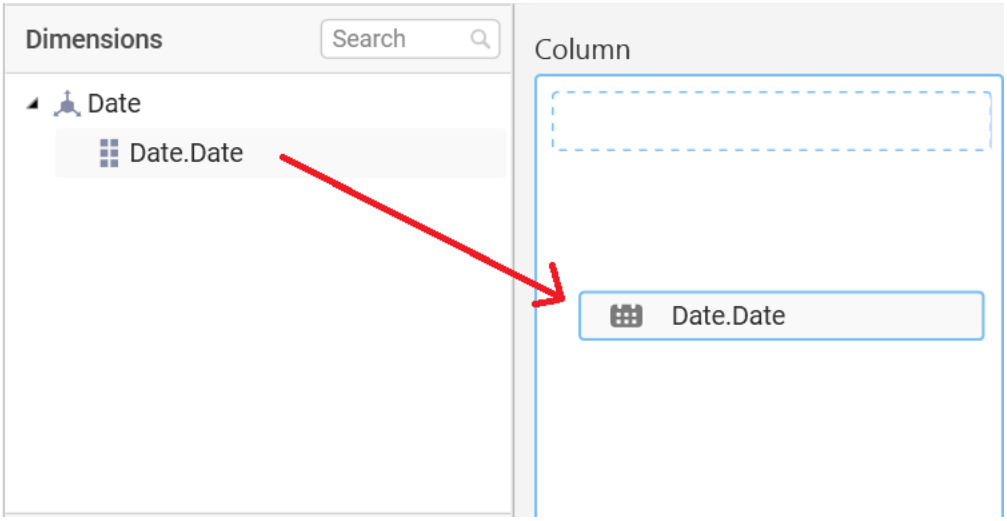
Select the dropped widget using mouse.

Click on Assign Data button in the tool bar.

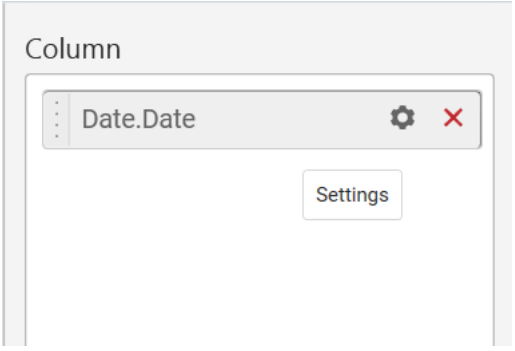


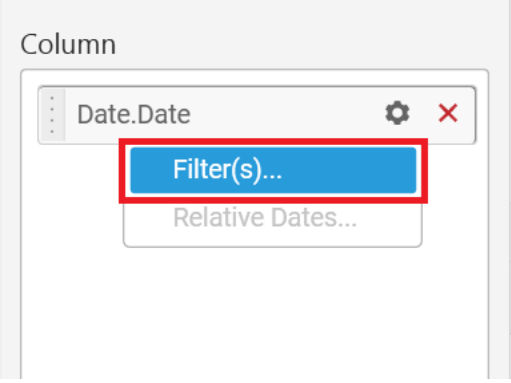
A Data pane will be opened with available date Dimensions.

Drag and drop a column under Dimensions category into Column.

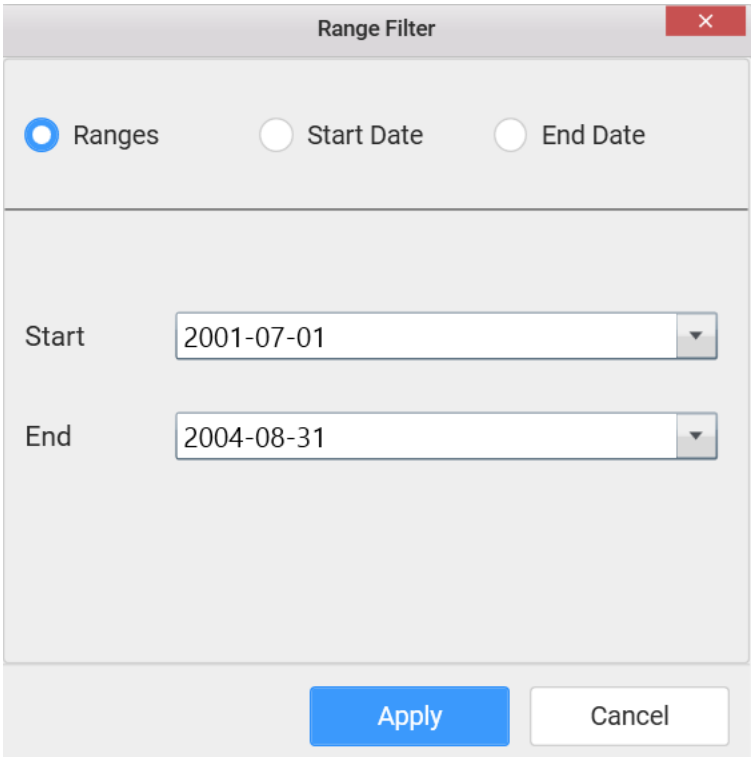


Define filter with specific range of dates by click the Settings icon in the dropped column and select Filter(s)... in the drop down menu.





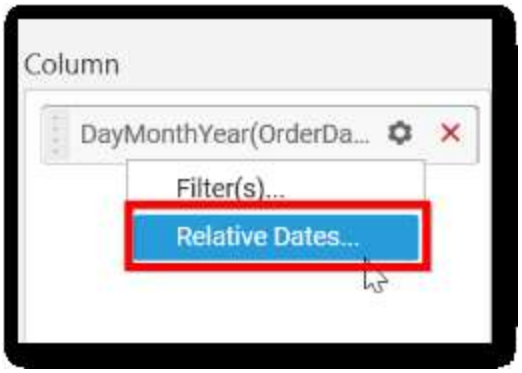
Define the date range in Range Filter dialog and click Apply.



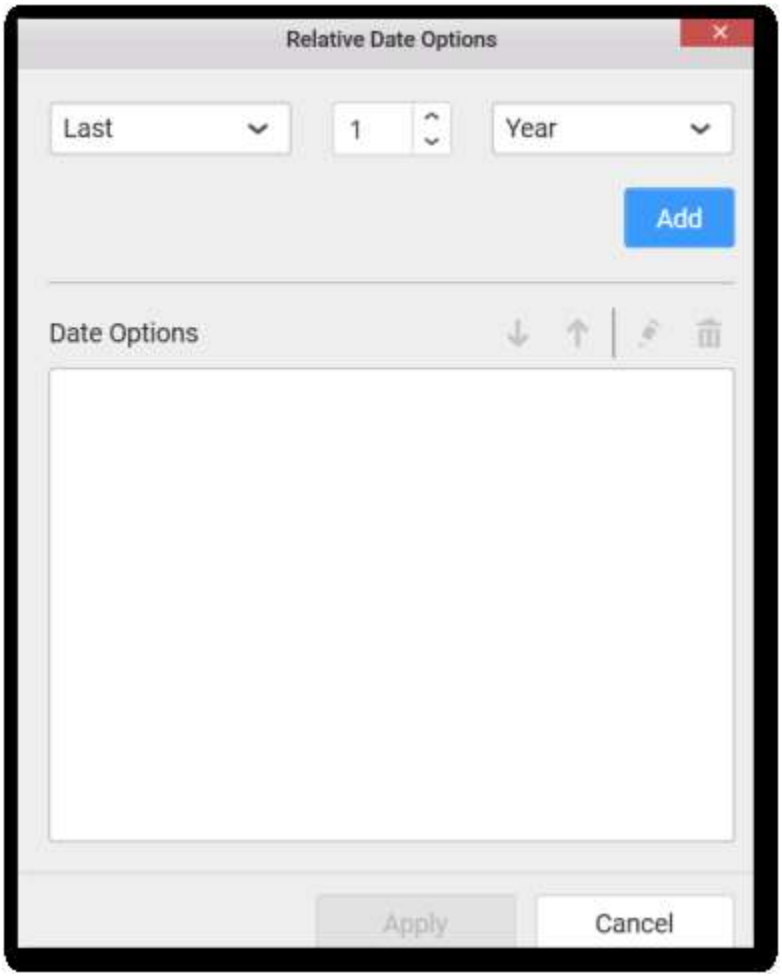
To show all records again click on Show All Records.

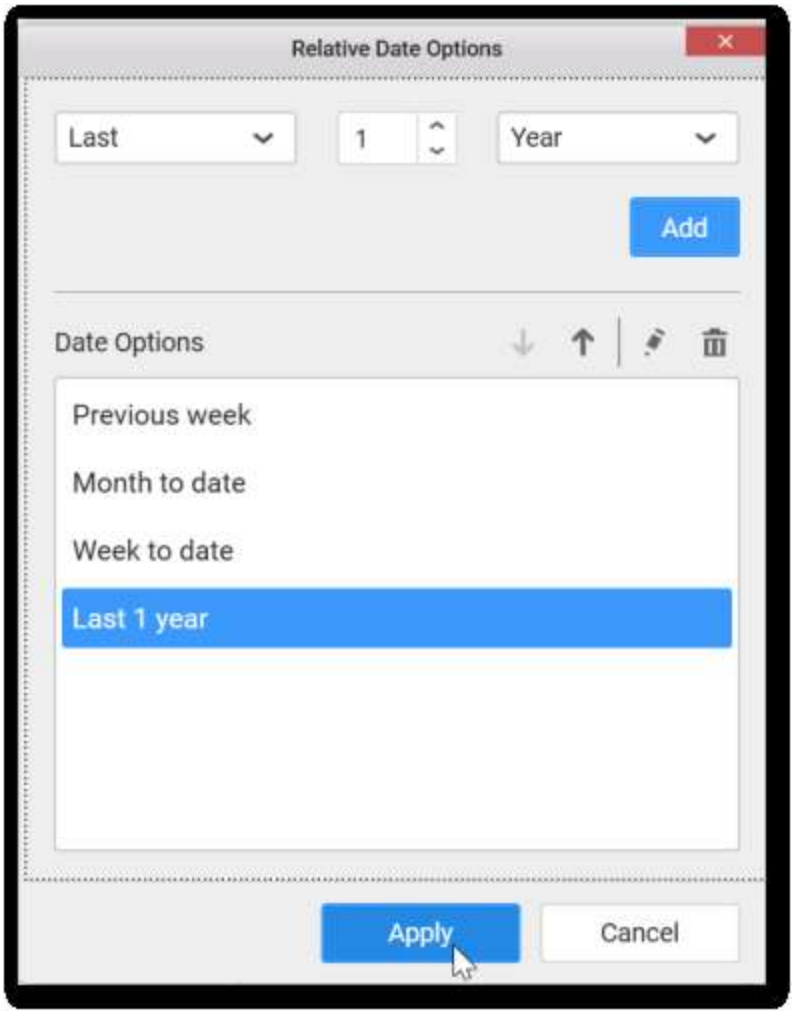
How to configure relative dates to date picker?

Switch to the Properties pane; Set the Selection Type as Range; Switch back to Data pane; Click the Settings icon in the dropped date column and select Relative Dates... in the drop down menu.

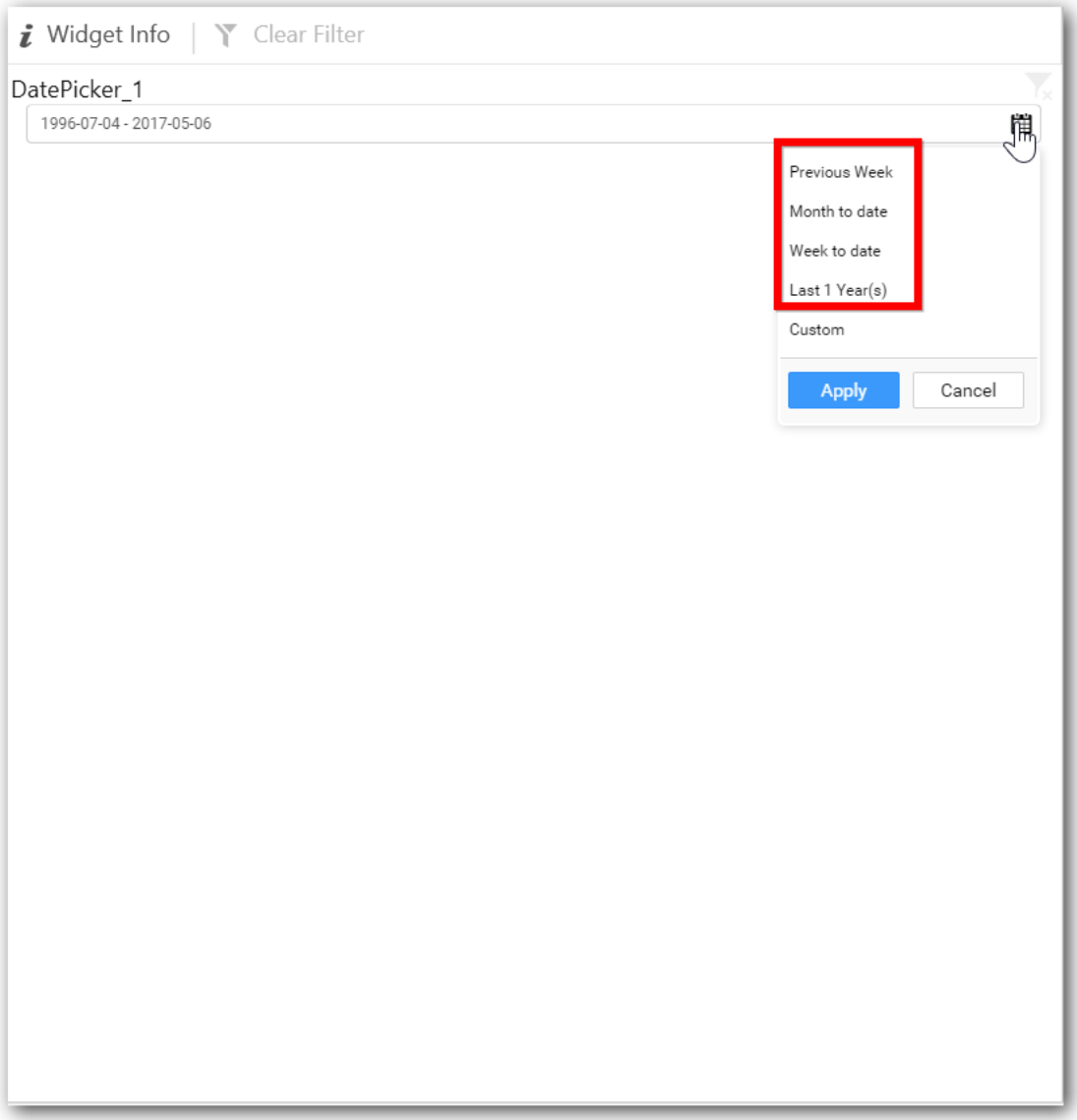


In the launched Relative Date Options window, configure the relative date and click Add. Repeat the same till the required set of relative dates added.



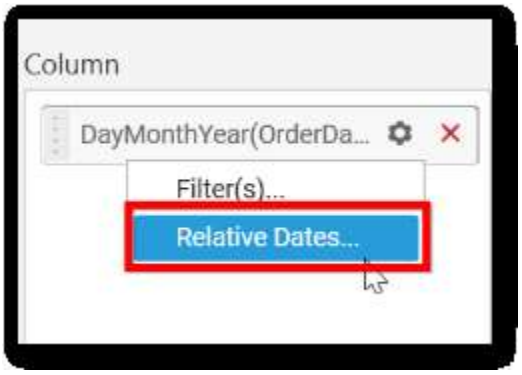


You can see the added relative dates in date picker like below.

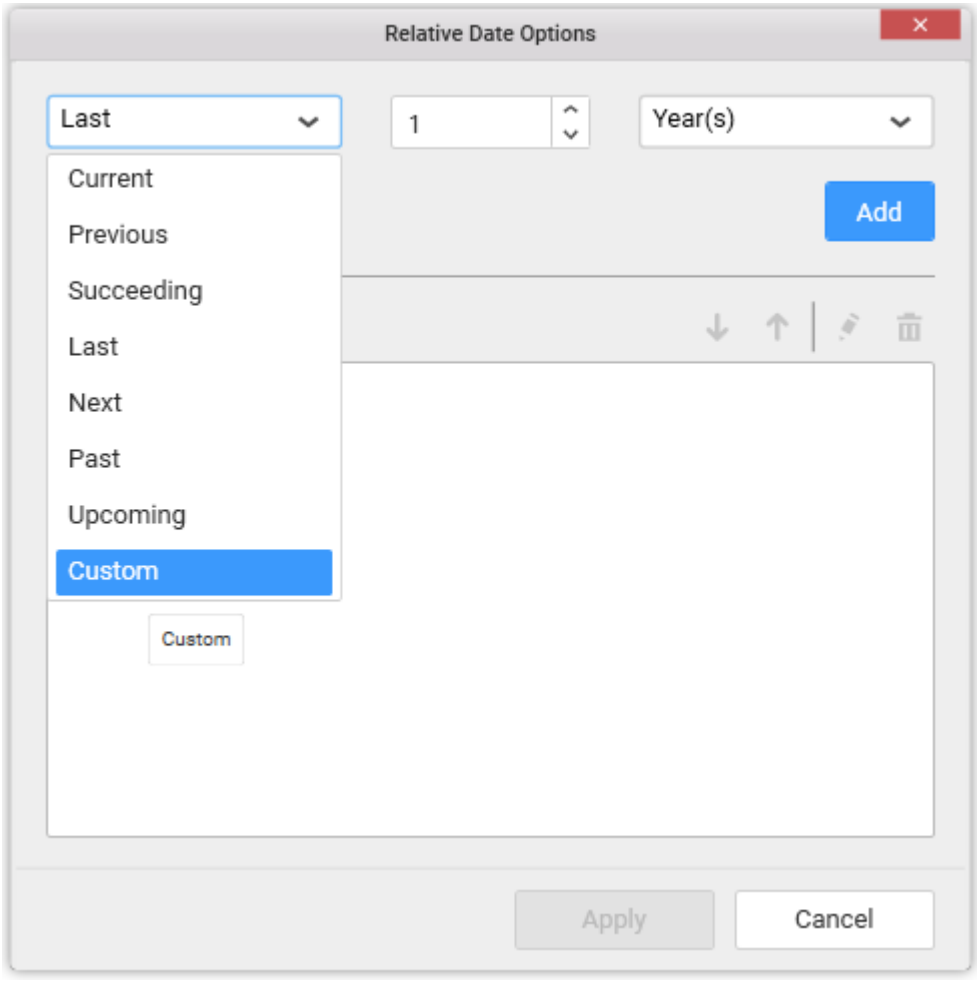


How to configure custom relative dates to date picker?

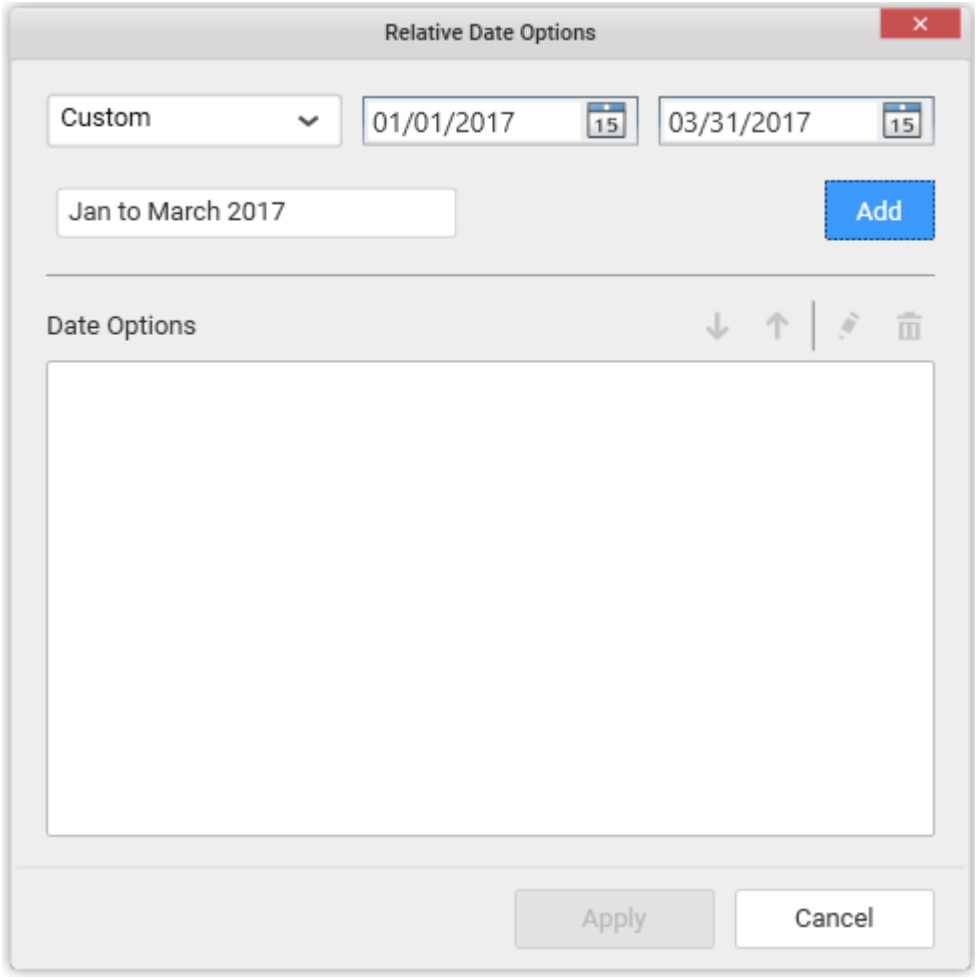
Switch to the Properties pane; Set the Selection Type as Range; Switch back to Data pane; Click the Settings icon in the dropped date column and select Relative Dates... in the drop down menu.

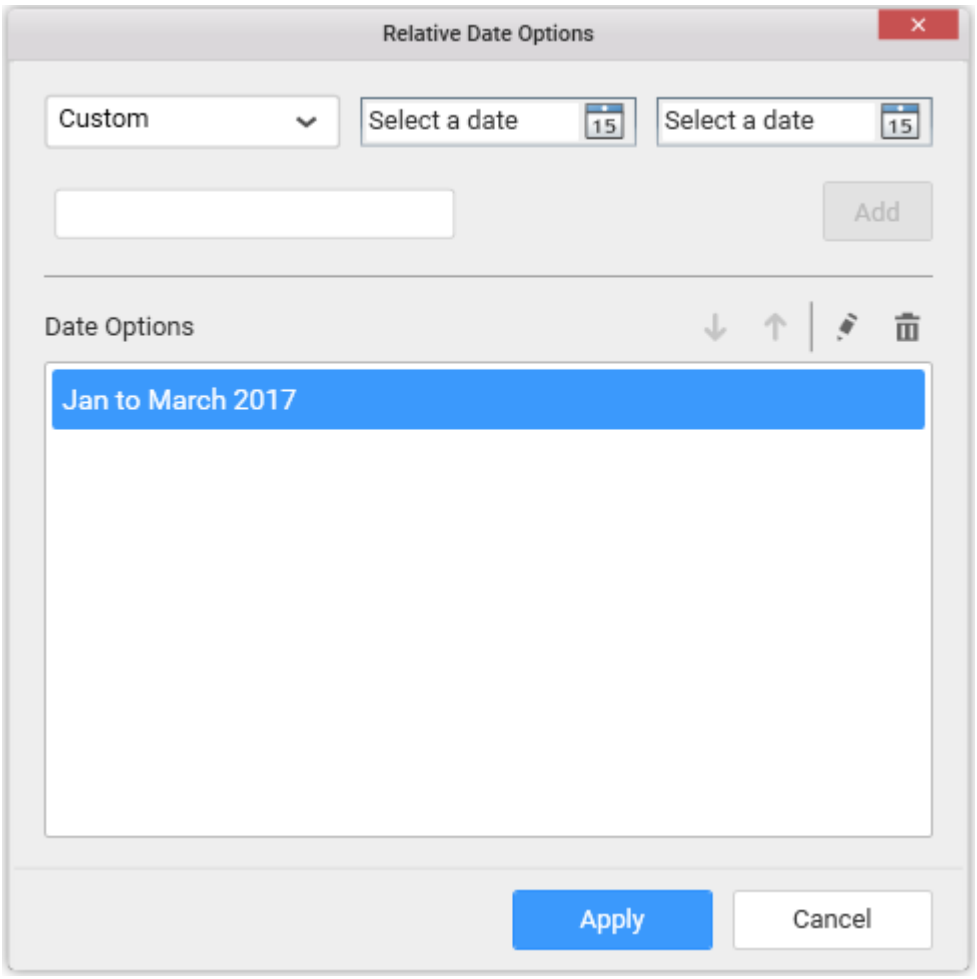


In the launched Relative Date Options window, select Custom option from DropDownList.



Choose the start and end dates which you like to set as custom range, also set the name for the custom range and click Add. Repeat the same till the required set of custom relative dates added.

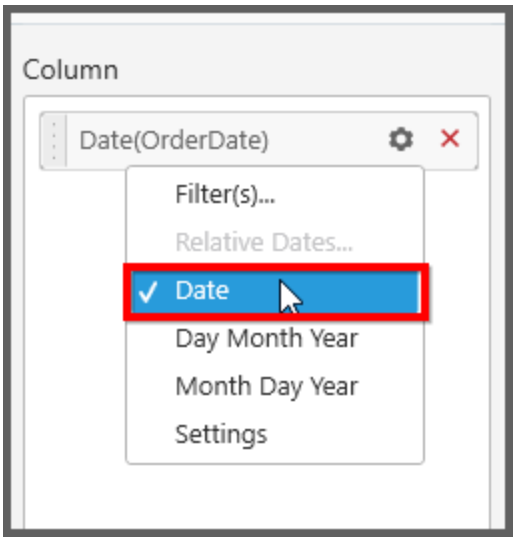




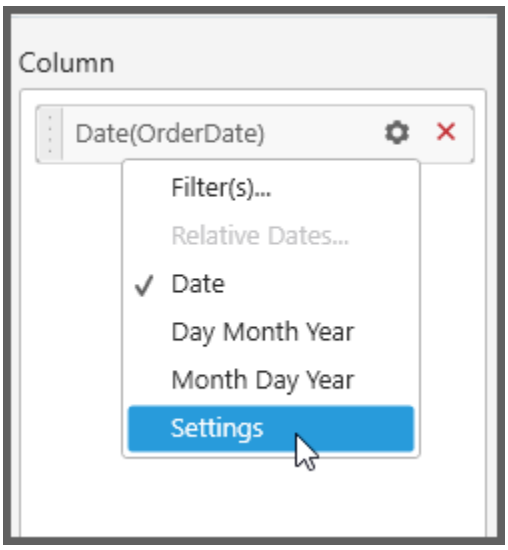
You can see the added custom relative dates in date picker like below.

How to customize date format in date picker?

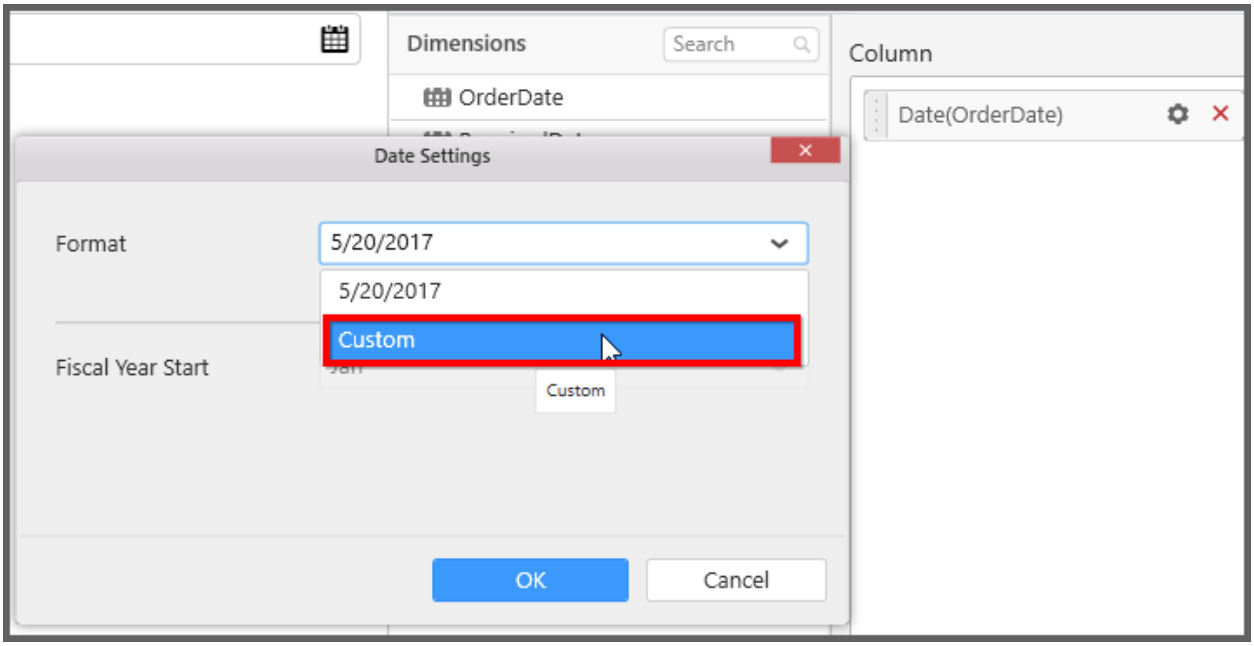
This helps you to set your own custom date formats only for "Date" format. Switch to the Properties pane and select date format.



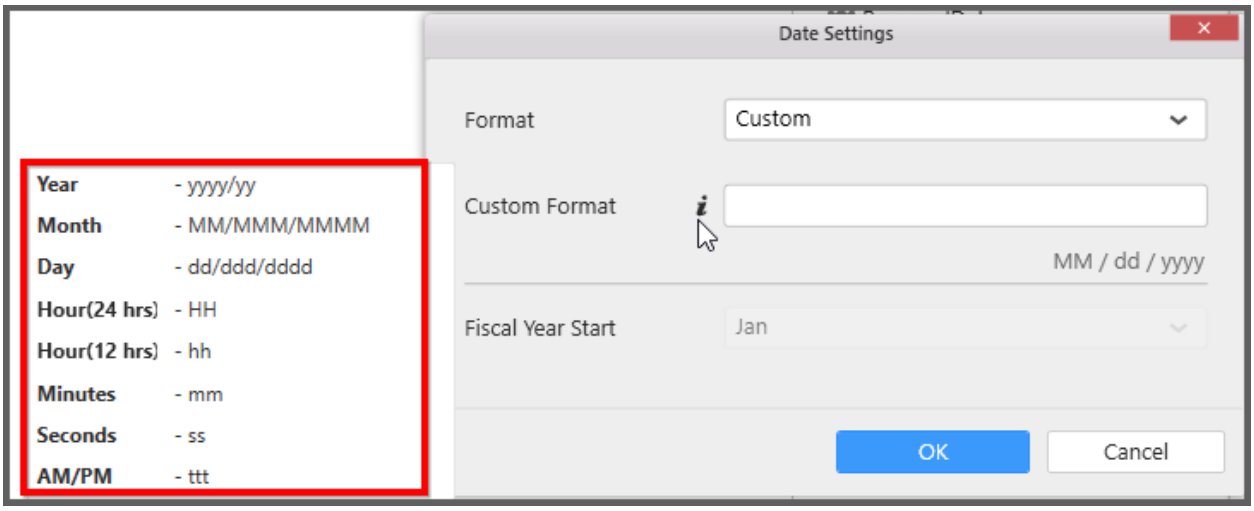
You can see the "Settings" option in the dropdown, after enabling "Date" format.



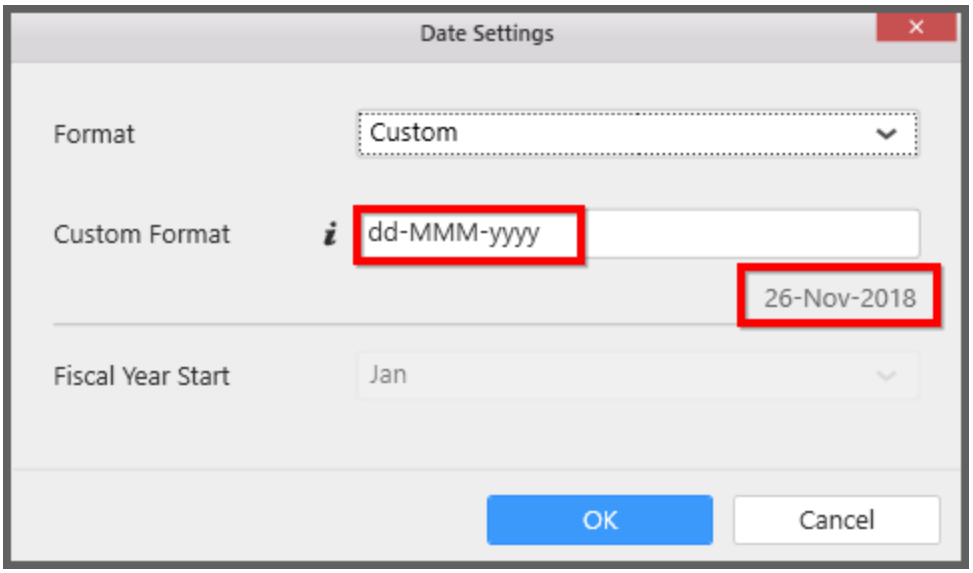
Select "Custom" option to customize the date format.



You can see the list of standard format to customize your date in Date picker control.



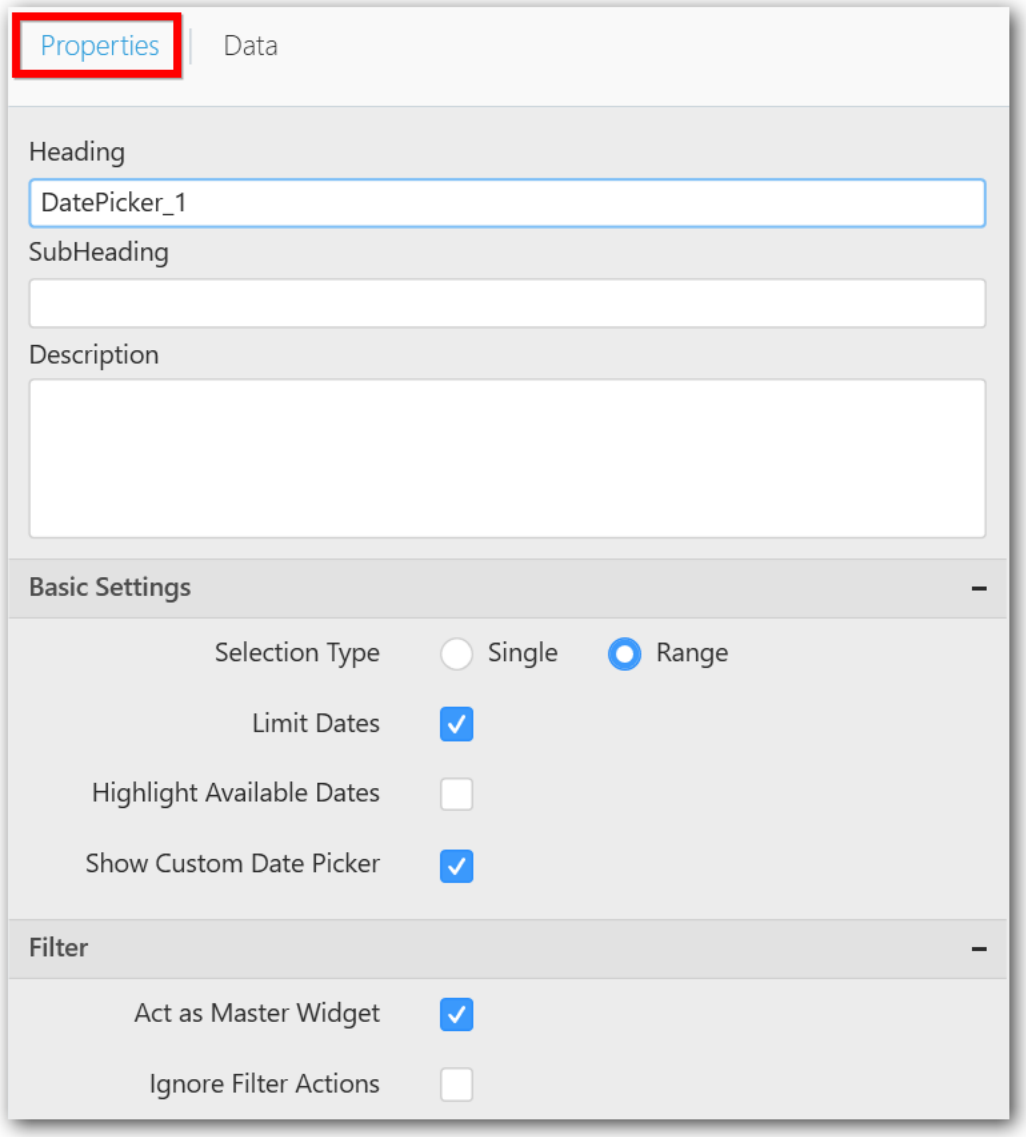
you can use the standard format as below.



How to format date picker?

Keep the selection focus on the date picker and Click on Widget Settings icon.

The property window will be opened like below.



You can see the list of properties available for the widget with default value.

General Settings

Header

This allows you to set title for this Date Picker widget.

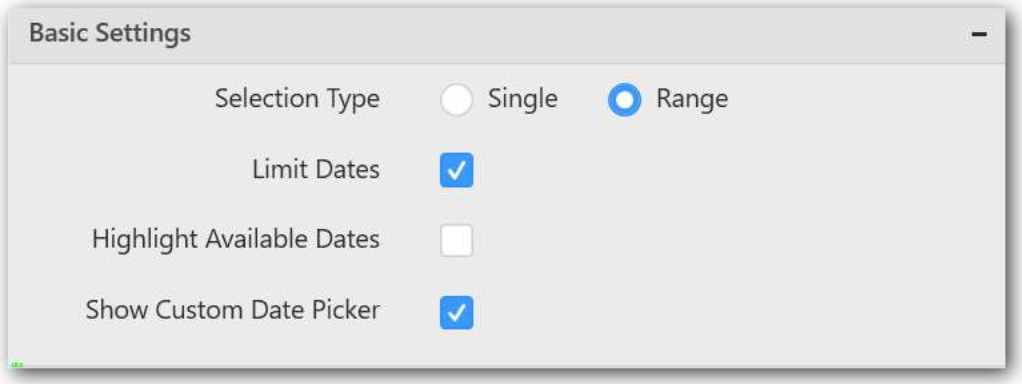
SubHeading

This allows you to set sub-title for this Date Picker widget.

Description

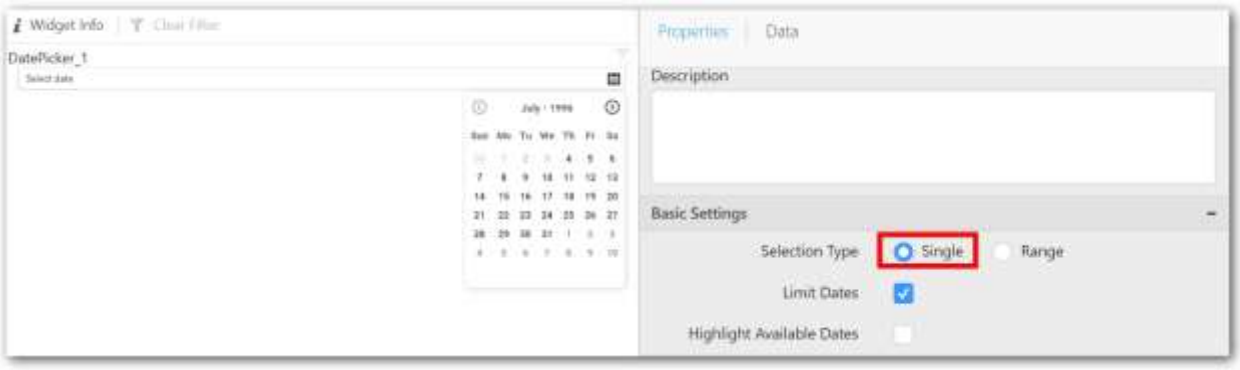
This allows you to set description for this Date Picker widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings

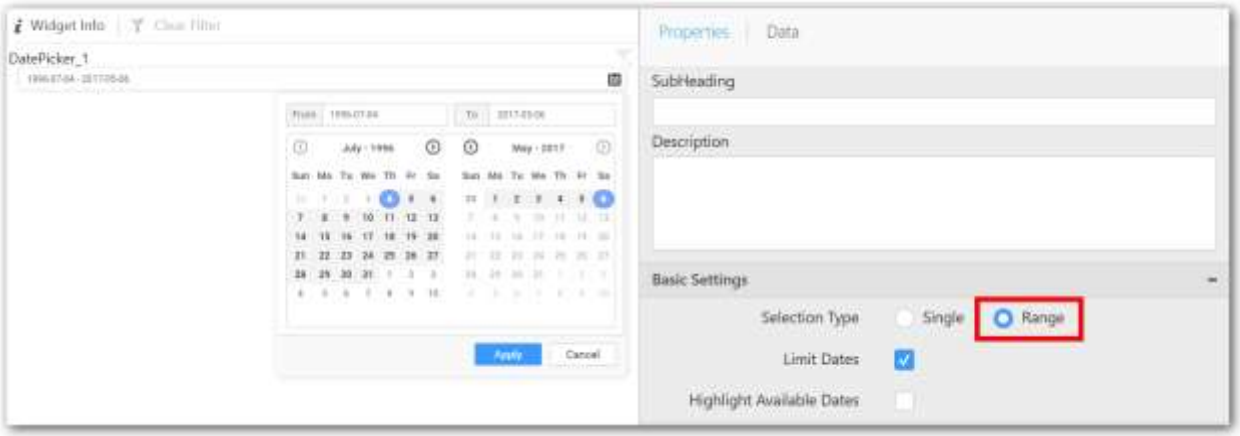


Selection Type

Single – Single date can be bounded.



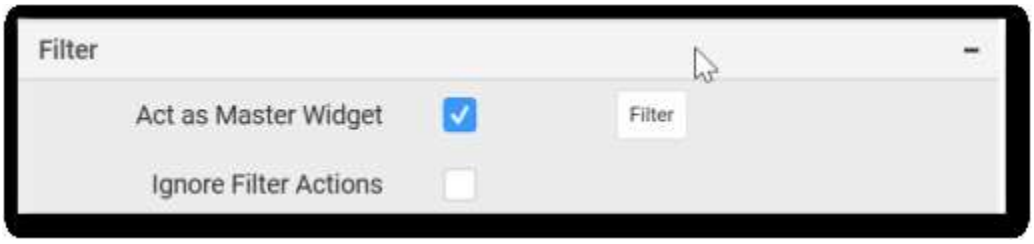
Range – A range of dates (two dates) can be bounded.



Highlight Available Dates

This allows you to enable the highlighting of available dates in date picker.

Filter Settings



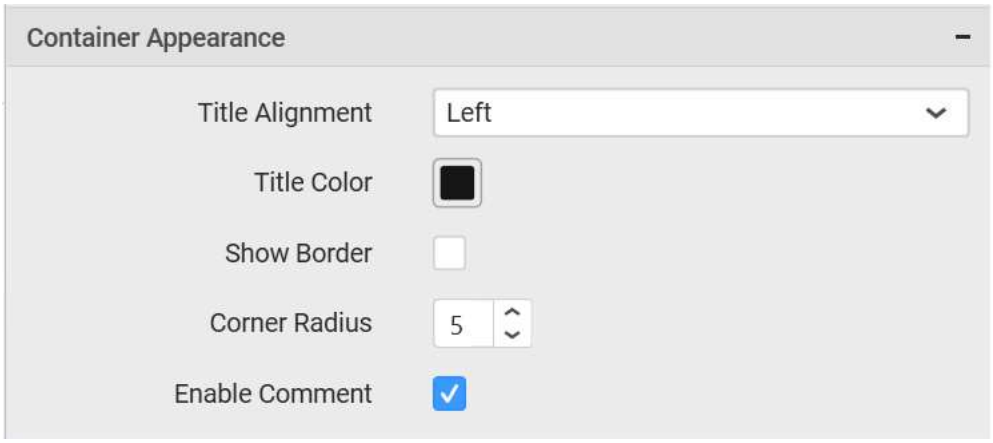
Act as Master Widget

This allows you to define this date picker widget as a master widget such that its filter action can be made to listen by other widgets in the dashboard.

Ignore Filter Actions

This allows you to define this date picker widget to ignore responding to the filter actions applied on other widgets in dashboard.

Container Settings



Title Alignment

This allows you to handle the alignment of widget title to either left, center or right.

Title Color

This allows you to apply text color to the widget title.

Show Border

This allows you to toggle the visibility of border surrounding the widget.

Corner Radius

This allows you to apply the specified radius to the widget corners. Value can be between 0 and 10.

Enable Comment

This allows you to enable/disable comment for dashboard widget. For more details refer here

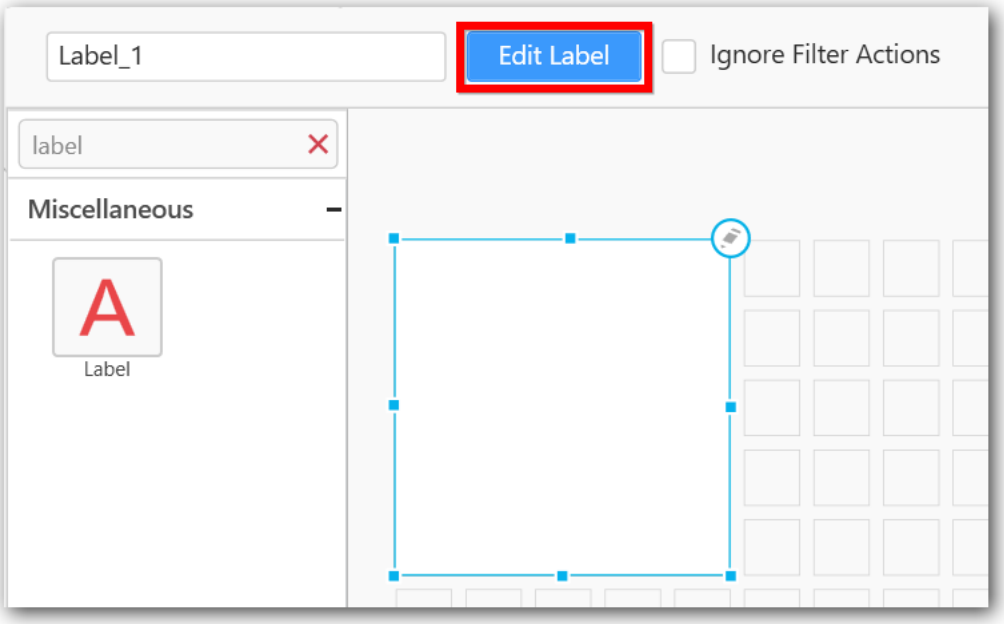
Label

Label allows you to display value set in rich text format, hyperlinked and through parameter placeholders.

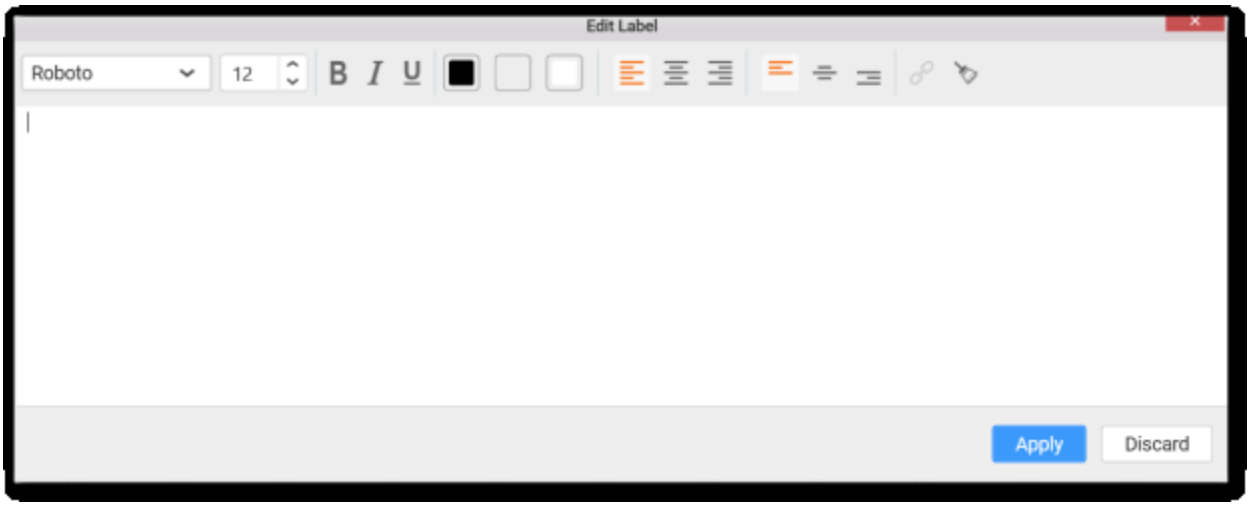
How to add text and format label?

The following procedure illustrates data configuration of Label.

Drag and drop Label control icon from the Tool box into design panel. You can find control in Toolbox by search.

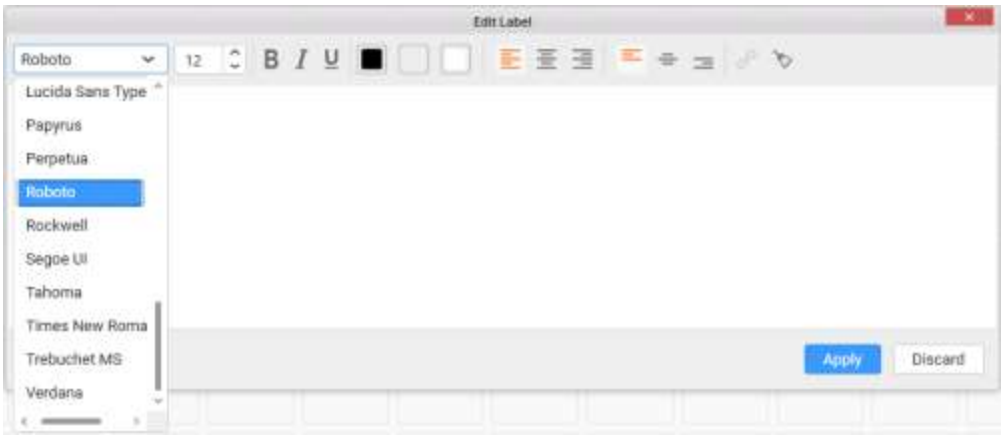


After control added in design panel, you can click edit icon at top right corner/ click Edit Label in Design Tools pane, to open the label editor.

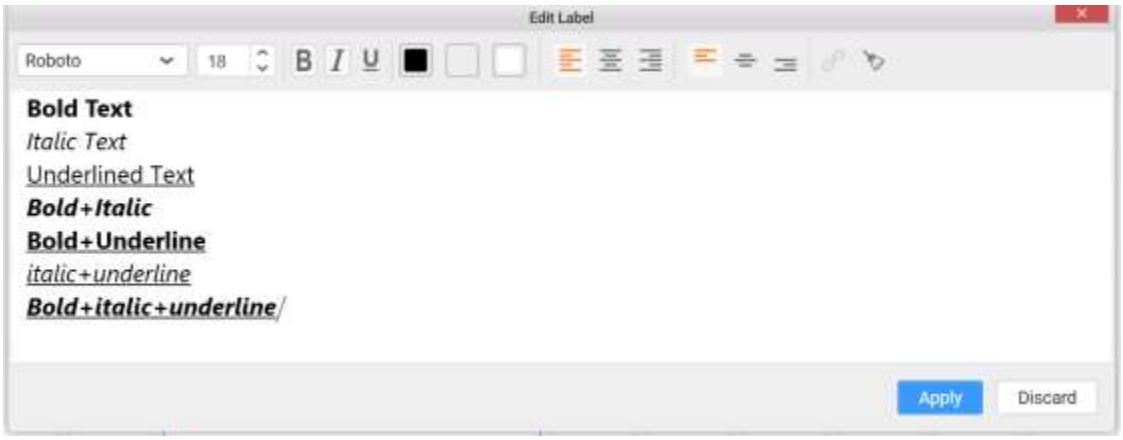


Font Family Selection

You can select font family of the text.

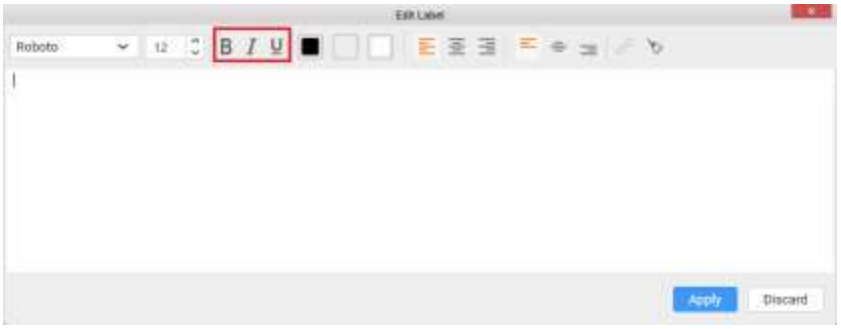


Emphasis



Font Style Selection

You can able to change font style of the text.

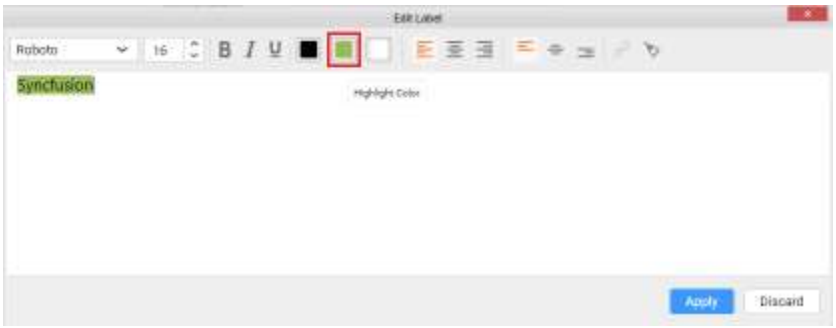


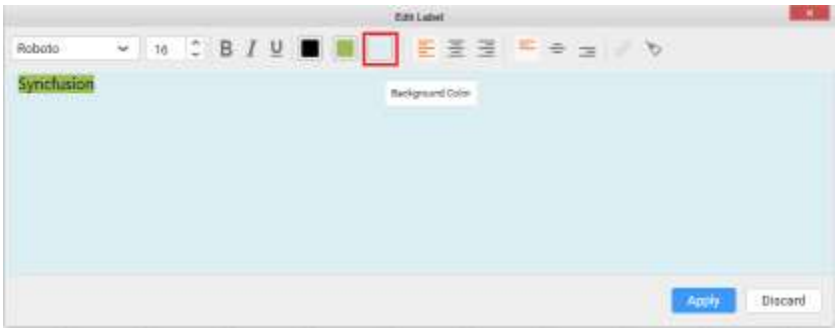
Color Customization

You can customize foreground and highlighting color of selected text and Background color of the label editor.

Foreground Color of label

Highlight Color of label





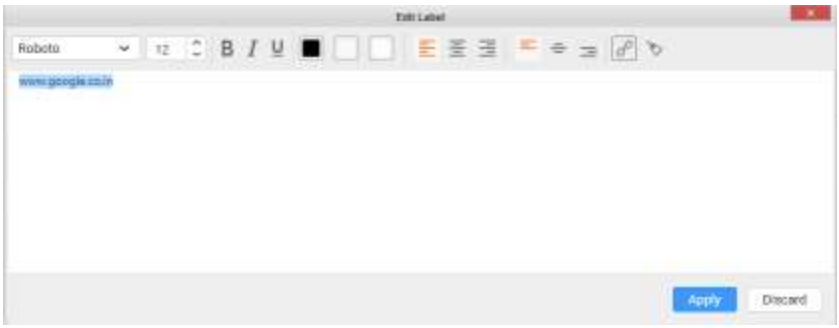
Text Alignment

You can change vertical and horizontal alignment of the text.

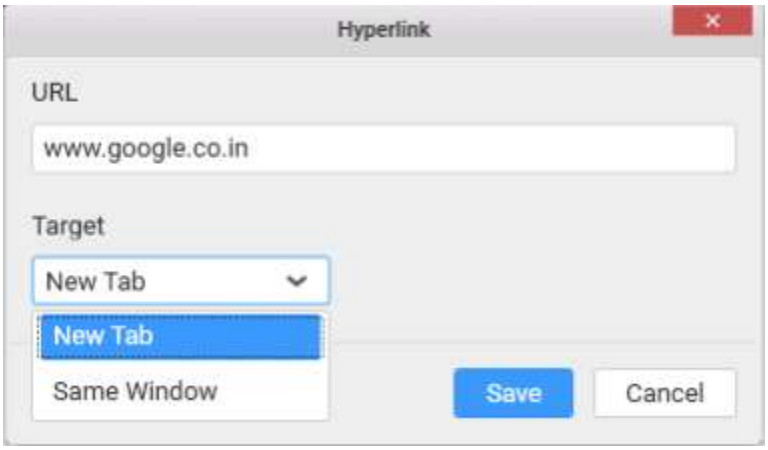
Hyperlink

The following procedure illustrates how to add hyperlink.

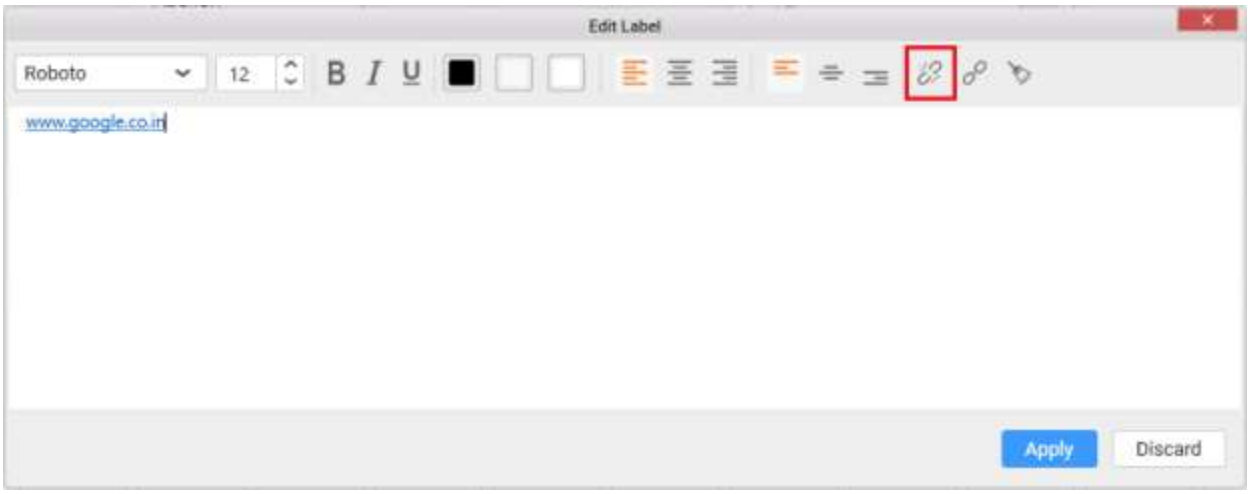
Add and select the URL link, Add Hyperlink option should be enabled.



Click the option, you can customize hyperlink settings.

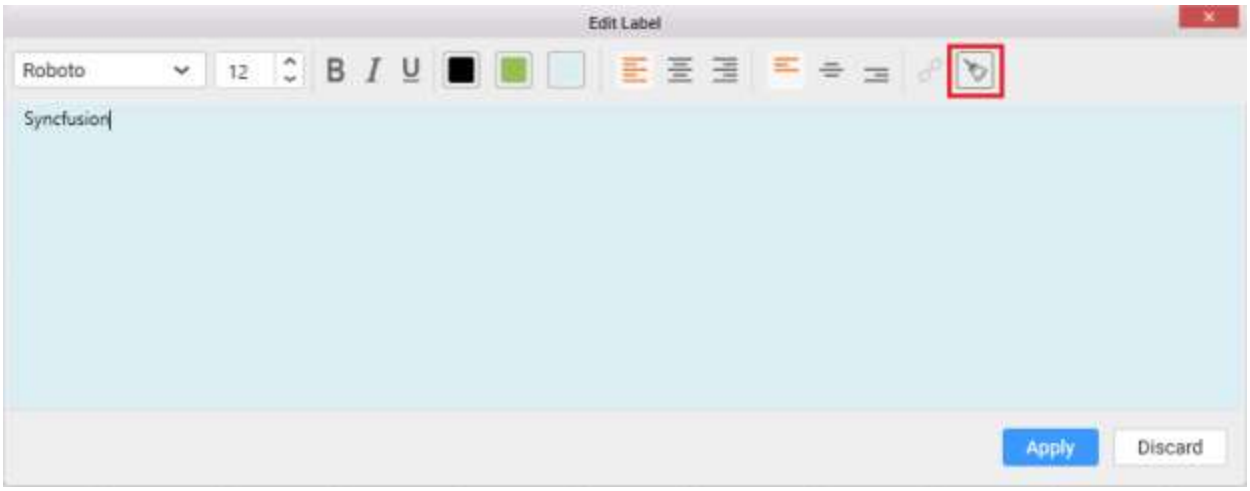


To remove hyperlink, click Remove Hyperlink which will be shown after adding hyperlink.



Reset Formatting

You can reset all the format settings applied to the text.



Image

Image allows you to display a both static and dynamic image within defined mode (default, fill, uniform and uniform to fill).

You may add image of supported formats including, BMP, JPG, JPEG, GIF, EMF, JFIF, JPE, PNG, RLE, TIF, TIFF, WMF, DIB, and ICO from your local machine or column binding to the image widget.

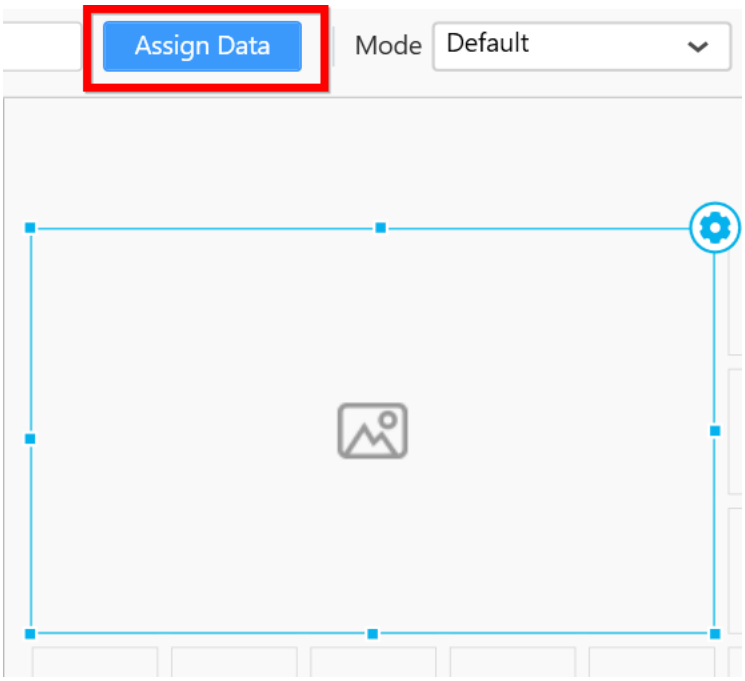
The following steps represents to add Image to dashboard.

Drag and drop the image widget into the Canvas.

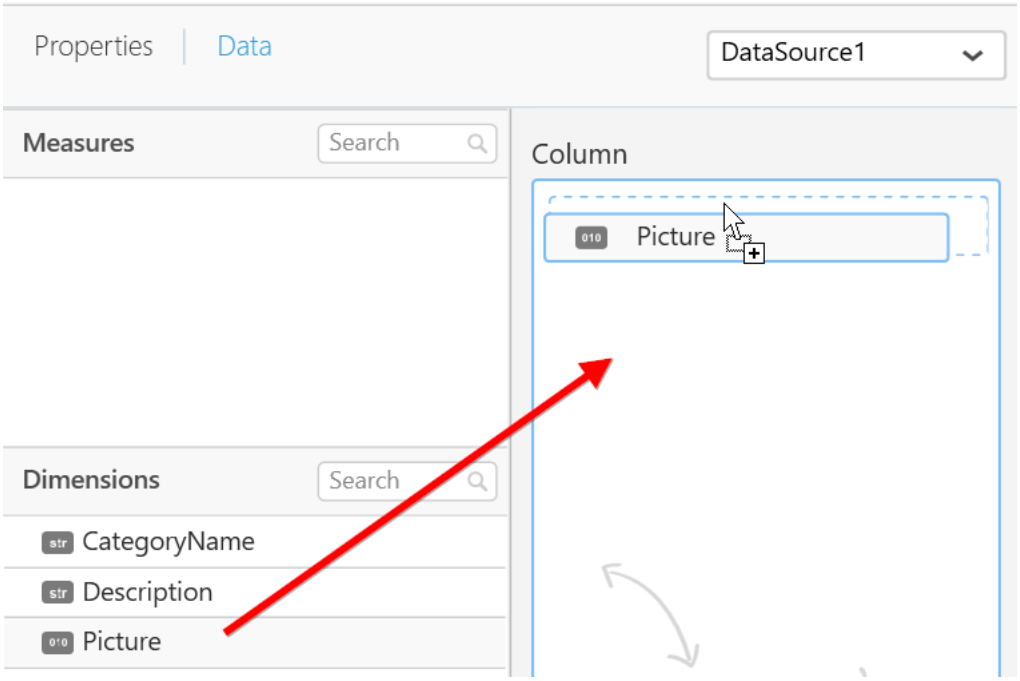
How to configure the data to Image widget?

You may browse the image or bind a datasource column for Image widget. Binding column must be in binary or string format for image widget.

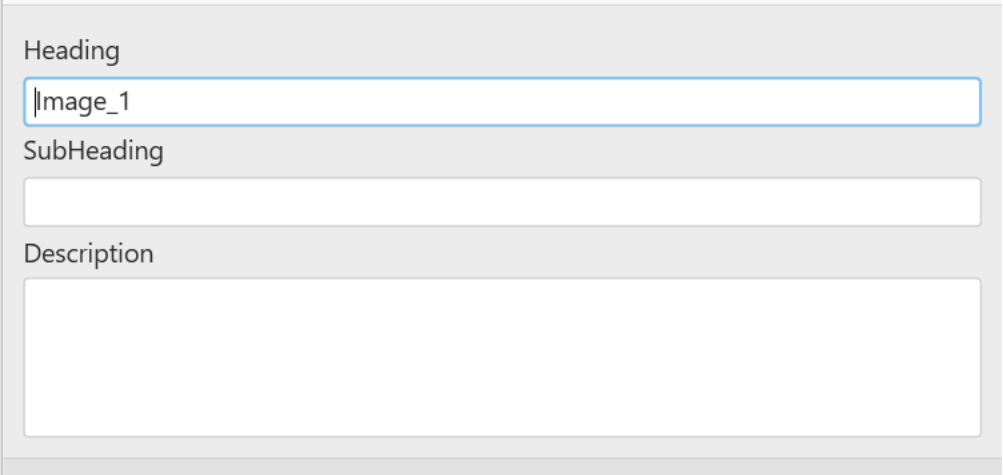
You may assign a data by clicking **Assign Data** button.



Drag and drop image column from dimension section to Column(s) section.



You can format the image for better illustration of the view that you require, through the settings available in Properties pane.



Header

This allows you to set title for this Image widget.

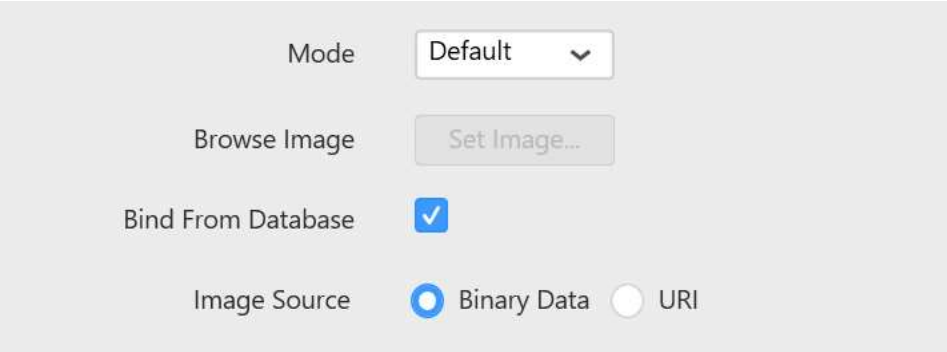
SubHeading

This allows you to set the subtitle for Image widget.

Description

This allows you to set description for this Image widget, whose visibility will be denoted by i icon, hovering which will display this description in tooltip.

Basic Settings



Mode

You can customize the image showcase style through Mode setting in the Design Tools pane or properties pane.

Default

The image will be displayed in its original size.

Fill

The image will be filled in the available space.

Uniform to Fill

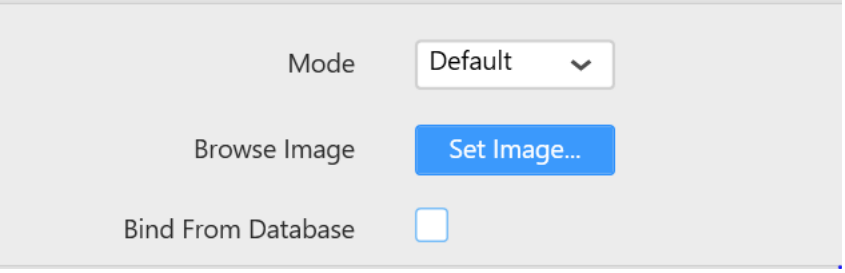
The image will be uniformly occupying the space but gets clipped, if it is larger than control

Uniform

The image sizes proportionally (without clipping) to best fit to the widget area.

Browse Image

You can browse the image from your local system.



Note: Image that having special characters in the file name is not supported in Dashboard Application.

Bind From Database

you can able to bind image from the datasource through this support. First image of bound column will be shown in the image widget

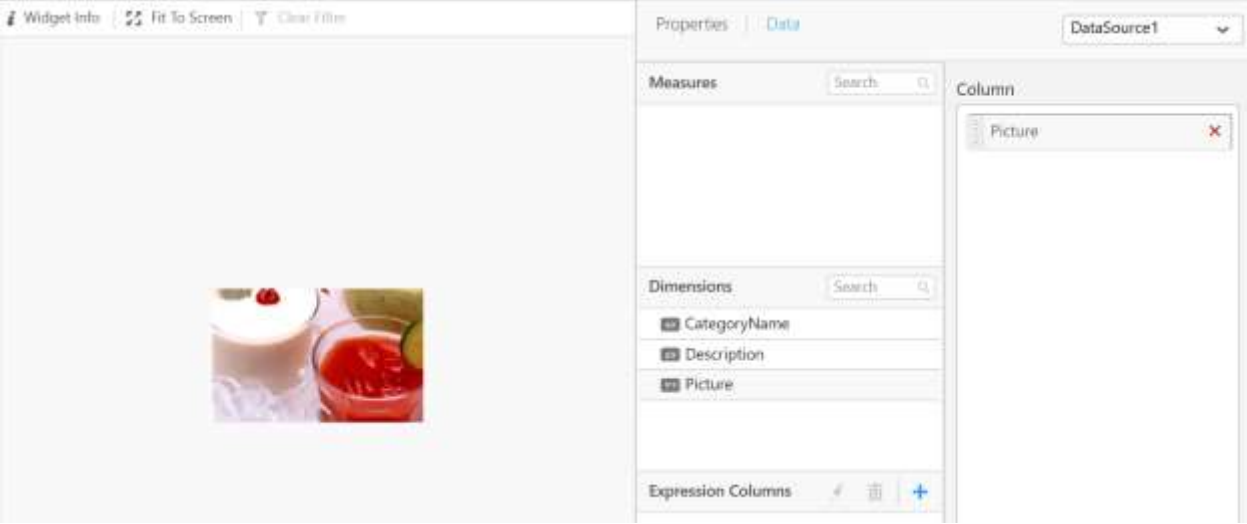
Image Source

Image datasource can be binded in two ways.

- Binary Data
- URI

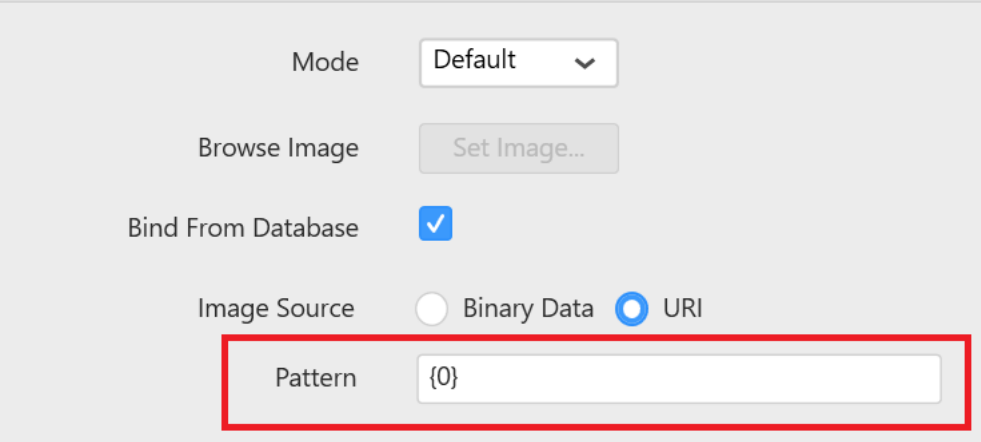
Binary Data

you can able to view binary data image through this option.



URI

URI images can be bind through this option and we can also able to add different columns value as a placeholder through the placeholder textbox.

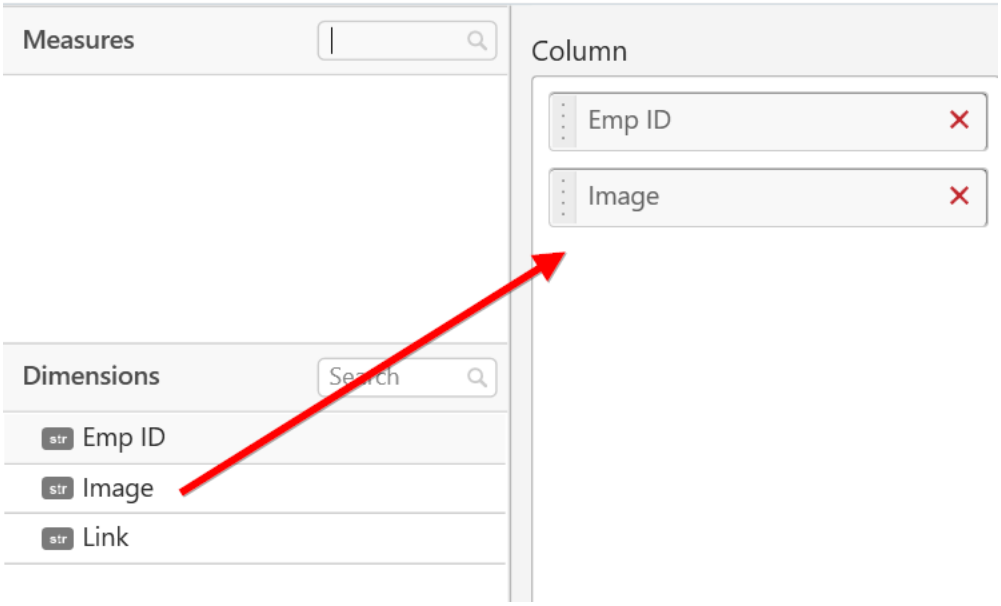


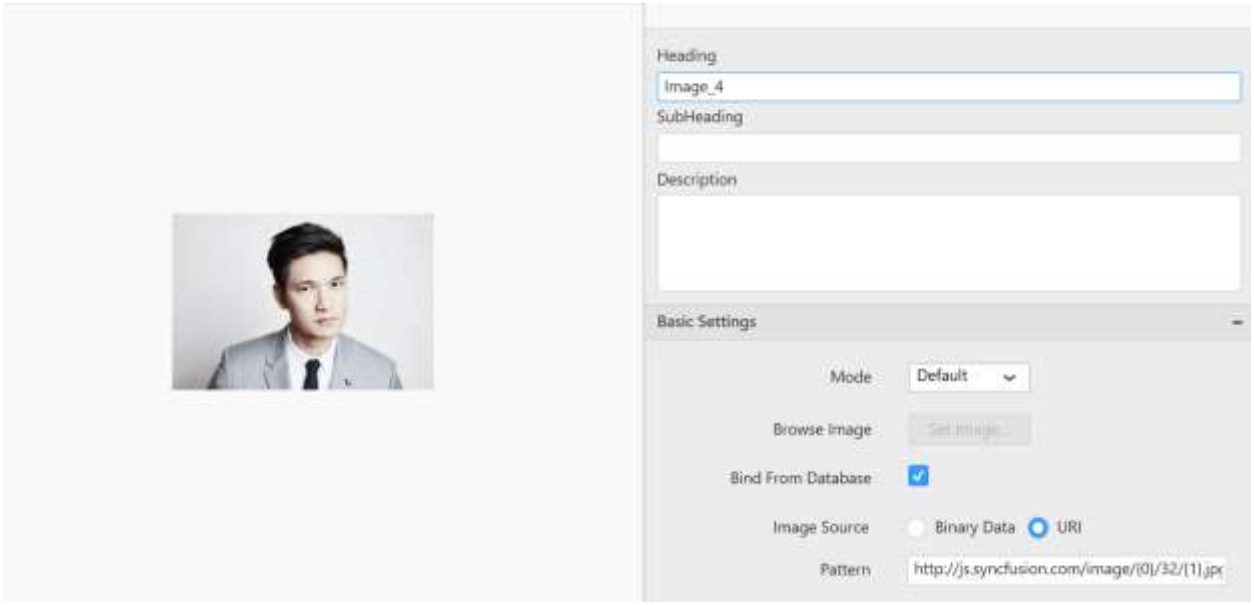
Binding direct image URI

Ex: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTxlVqMSd30Qah-lV4zXbDimmEc0-BYovuJMdDvvAhNIU6l79Nd

Forming URI through placeholder

EX: http://js.syncfusion.com/image/{0}/32/{1}.jpg





Filter Settings

IgnoreFilterActions

You can ignore the filter actions by enabling IgnoreFilterActions property. Browse Image will not act as a slave widget.

Container Appearance

Image Export

You can export the image widget as image.

PDF Export

You can export the image widget as PDF.

Syncfusion Custom Widget Authoring

Overview

The Syncfusion Dashboard custom widget allows you to add any user-defined widget, d3 controls, Syncfusion controls, or any JS widgets in a dashboard, and it can perform like the normal widget.

You can download the Syncfusion controls source files from the following links

- EJ1 controls
- EJ2 controls

To learn more about the custom widget creation, refer the video link.

Prerequisites and knowledge required for handling the custom widget authoring

This section covers the mandatory requirements to develop the custom widget.

Development Environment	Dashboard Designer and any web browser as mentioned in the link
Knowledge Required	Basic knowledge about the HTML and JavaScript

Syncfusion ships the following custom widgets along with the Syncfusion Dashboard Designer.

- Sparkline.sycw
- SunburstChart.sycw
- Radar.sycw
- Waterfall.sycw

And these custom widgets are available in the below location also.

Location: C:\ProgramData\Syncfusion\DashboardDesigner\Custom Widgets

Creating new widget

Syncfusion Custom Widget Utility (sycw.exe) is used to create and manage the custom widget and it will be shipped along with the Dashboard Designer setup installation in the following location.

Location: [Local drive installation location]\ Syncfusion\Dashboard Designer\DashboardDesigner\sycw.exe

Create command is used to create the widget in the specified path with provided name along with its unique GUID.

Format

create [widget creation path]

Example

create e:\widgets\sunburstchart

After the create command execution, you can see the widget named folder that is created as below:

Rules

- If you need to modify the widget name, display name, or GUID, it should be modified in both the widget configuration file(widgetconfig.json) and the source file(sourcefile.js) without mismatching. But modifying the GUID is not recommended.

Configuring widget using manifest file

After creating the widget, you can configure the widget by using the manifest file.

The manifest file is used to specify the widget's configuration. This file has to be placed in the custom widgets root directory with the name widgetconfig.json.

![Custom*widget sunburst folder structure](images/Custom*widget sunburst folder.structure.png)

Custom widget can be configured using the following settings. Refer the structure of widgetconfig.json

JS

```
{
"widgetName" : "Name of the widget.",
"displayName" : "Display name of the widget to be displayed in designer
toolbox.",
"enableIFrame" : "Used to render the widget within IFrame tag",
"guid": "Unique GUID for the widget.",
"category" : "Category in which the widget should be displayed in the
designer.",
"description" : "Description of the widget.",
"srcFile"  : "Path of the source file through which widget can be embedded
to
dashboard",
"dependencies" : "Specify the dependency scripts and CSS files to be
referred for
the widget",
"dataFields" : " Specifies the widget fields required for binding data from
a data
source. Each field can be bound to single or multiple
measure/dimension fields in a data source. When dataFields is not
specified, this widget acts as a non-data bound widget. ",
"functionalities" : " Custom properties for the widget can be specified
through the
functionalities API. ",
"filterSettings": "Filter settings are used to configure whether the widget
can take
part in filter interactions between the widgets.",
"enableLinking": "Used to enable or disable the linking support for the
widget",
"linkSettings": "Used to configure linking related settings for the widget"
}
```

This file contains the data necessary for the designer to import the custom widget and serialize the custom widget into *.sydx package while publishing the dashboard.

Setting name for the widget

Name for the widget can be set and specified through the widgetName API.

Format

widgetName: "name of widget"

Example

widgetName: "sunburst"

Rules

- It is a mandatory field for the widget.
- It should not contain any special characters and wide spaces.

Setting display name for the widget

Display name for the widget in the designer toolbar can be set through the displayName API.

![Custom*widget sunburst](images/Custom*widget sunburst.png)

Format

displayName: "display name of the widget in designer toolbar"

Example

displayName: "Sunburst Chart"

Rules

- It is not a mandatory field for the custom widget.
- It can contain special character and wide spaces.

Setting IFrame for the widget

Enable IFrame is used to specify whether the custom widget should be render within the IFrame tag or directly within the DashboardViewer. This support is used to resolve Scripts and CSS conflicts that occurs between the DashboardViewer and the custom widget. Currently exporting to image and PDF options are not provided for IFrame widgets.

Format

enableIFrame: "IFrame can be enabled or disabled for the widget"

Example

enableIFrame: "true"

Rules

- It is not a mandatory field for the custom widget.

Setting category for the widget

This allows you to specify the category in which widget should be displayed. If the specified category name is not exist, then a new category will be created on the provided name and the widget will be added under the new category.

If the name of the category is not provided for the widget, then it will be added under the miscellaneous category (default category).

![Custom*widget sunburst](images/custom*widget relationshoi sunburst.png)

Format

category: "name of the category in which the widget should be displayed"

Example

category: "Relationship"

Rules

- It is not a mandatory field for the custom widget.

Setting description for the widget

You can describe the widget. The following message will be shown on hovering the widget icon in the expander toolbox of the designer.

![Custom*widget description](images/custom*widget description.png)

Format

description: "description about the widget"

Example

description: "Sunburst Chart is used to visualize the hierarchical data. The center circle represents the root level of the hierarchy and outer circles represents higher levels of the hierarchy."

Rules

- It is not a mandatory field for the custom widget.

Setting source file location

Source file location should be specified here. Refer the use of source file in its module.

Format

srcFile: "location of the source file"

Example

srcFile: "src/sunburst.js"

Rules

- It is a mandatory field for the widget.

API version

Version is used to denote the current API version of the created widget and this value should not be modified.

Setting GUID for the widget

GUID is used to set unique identification for the custom widget, and it can be set through the guid API.

Format

guid: "used for unique identify the widget"

Example

guid: "74926583-8493-3333-6382-863428678492"

Rules:

- It is a mandatory API for the widget.
- Should satisfy the GUID format.

Setting dependency files for the widget

Dependency script and CSS files, which have to be referred for the widget, and should be specified in the dependencies section.

Format

JS

```
"dependencies": {
"scripts": [
"src/dependency1.js",
"src/ dependency2.js",
….
….
],
"styles": [
"style/ dependency1.css",
" style / dependency2.css",
….
….
]
}
```

Example

JS

```
"dependencies": {
"scripts": [
"src/ej.sunburstchart.js",
"src/ej.helper.js"
],
"styles": [
"style/ej.sunburstchart.css"
]
}
```

Rules

- It is not the mandatory API for the widget.
- Files should be specified in the reference order.

Limitation

jQuery reference (“jquery-1.10.2.min.js”) will be added as built-in reference for the custom widget when the IFrame is disabled, So that you are not allowed to specify the jQuery files in dependencies. If you want to add any specific version of jQuery, then you should enable the IFrame for the custom widget. For more information refer the link.

Specifying data fields

This allows you to specify the data fields with its type for custom widgets.

JS

```
dataFields:[
{
"displayName" : "display name of the data field1",
"valueType" : "type of the data field1, can be measure or dimension",
"name" :  "name of the data field1",
"min" : "minimum number of block which has to be dropped for the
datafield1",
"max" : "maximum number of block can be dropped for the datafield2",
"optional" : "specify whether it is mandatory data field"
},
{
"displayName" : " display name of the data field2",
"valueType" :" type of the data field2, can be measure or dimension",
"name" : " name of the data field2",
"min" : "minimum number of block which has to be dropped for the
datafield2",
"max": "maximum number of block can be dropped for the datafield2",
"optional" : "specify whether it is mandatory data field"
}
…
]
```

Example

JS

```
dataFields:[
{
"displayName" : "Value",
"valueType" : "measure",
"name" :  "Value",
"min" : 1,
"max" : 1,
"optional" :false
},
{
"displayName" : "Levels",
"valueType" : "dimension",
"name" : "Levels",
"min" : 1,
"max":4,
"optional" : false
}]
```

Rules

- It is not the mandatory API for the widget.

![Custom*widget schema](images/custom*widget schema.png)

Adding functionalities

Custom properties for the widget can be set and specified through the functionalities API.

Format

JS

```
"functionalities": [
{
"header": "name of the header",
"properties": [
{
"displayName": "display name of the properties",
"controlType": "type of control to be created in
properties window under this group",
"name": "name of property",
"defaultValue": "default value of the property",
"listItems": "list of items if the control type have list."
},
{
"displayName": " display name of the properties ",
"controlType": " type of control to be created in
properties window under this group ",
"name": " name of property ",
"defaultValue": " default value of the property "
}
……
……
]
},
{
"header": "name of the header",
"properties": [
{
"displayName": "display name of the properties",
"controlType": "type of control to be created in
properties window under this group",
"name": "name of property",
"defaultValue": "default value of the property",
"listItems": "list of items if the control type have list."
},
{
"displayName": " display name of the properties ",
"controlType": " type of control to be created in
properties window under this group ",
"name": " name of property ",
"defaultValue": " default value of the property "
}
……
……
]
}
……..
……..
]
```

Various control type sample

- Checkbox

![Custom*widget checkbox](images/custom*widget checkbox.png)

Format

JS

```
{
"displayName": "display name for the checkbox",
"controlType": "checkbox",
"name": "name of the property",
"defaultValue": "either true or false"
```

Example

JS

```
{
"displayName": "Enable Animation",
"controlType": "checkbox",
"name": "enableAnimation",
"defaultValue": "true"
}
```

- Textbox

![Custom*widget Textbox](images/custom*widget first widget.png)

Format

JS

```
{
"displayName": " display name for the textbox ",
"controlType": "textbox",
"name": "name of property",
"defaultValue": "default text to be displayed in the textbox"
}
```

Example

JS

```
{
"displayName": "Control Text",
"controlType": "textbox",
"name": " subtitle ",
"defaultValue": "My first widget"
}
```

- Combobox

![Custom*widget Combobox](images/custom*widget combobox.png)

Format

JS

```
{
"displayName": "display name of the property",
"controlType": "combobox",
"name": "name of property",
"listItems": ["listitem1"," listitem2"…],
"defaultValue": " listitem2"
}
```

Example

JS

```
{
"displayName": "Legend Position",
"controlType": "combobox",
"name": "legendPosition",
"listItems": ["top", "bottom", "left", "right"],
"defaultValue": "left"
}
```

- ColorPicker

![Custom*widget color picker](images/custom*widget color picker.png)

Format

JS

```
{
"displayName": "display name of the property ",
"controlType": "colorpicker",
"name": "name of property",
"defaultValue": "default color code"
}
```

Example

JS

```
{
"displayName": "label Color",
"controlType": "colorpicker",
"name": "labelColor",
"defaultValue": "#463859"
}
```

- Updown

![Custom*widget updown](images/custom*widget updown.png)

Format

JS

```
{
"displayName": " display name of the property ",
"controlType": "updown",
"name": " name of property ",
"defaultValue": "10",
"min" : "minimum value can be set",
"max" : "maximum value can be set"
}
```

Example

JS

```
{
"displayName": "Text Size",
"controlType": " updown ",
"name": "textSize",
"defaultValue": "10",
"min" : "1",
"max" : "50"
}
```

Filter settings

If the widget should take part in filter interactions that occur between the widgets, then the filter settings can be used.

Format

JS

```
"filterSettings": {
"masterFilter": {
"visible": "boolean value to specify whether can be act as master ",
"defaultValue": " boolean value to specify whether widget is master by
default"
},
"ignoreMasterFilter": {
"visible": " boolean value to specify whether can be updated on
interaction from other widgets",
"defaultValue": " boolean value to specify whether widget ignore master
action by default "
}
}
```

Example

JS

```
"filterSettings": {
"masterFilter": {
"visible": "true",
"defaultValue": "true"
},
"ignoreMasterFilter": {
"visible": "true",
```



```
"defaultValue": "false"
}
}
```

Rules:

- It is not a mandatory field for the widget.

To configure Linking for the widget

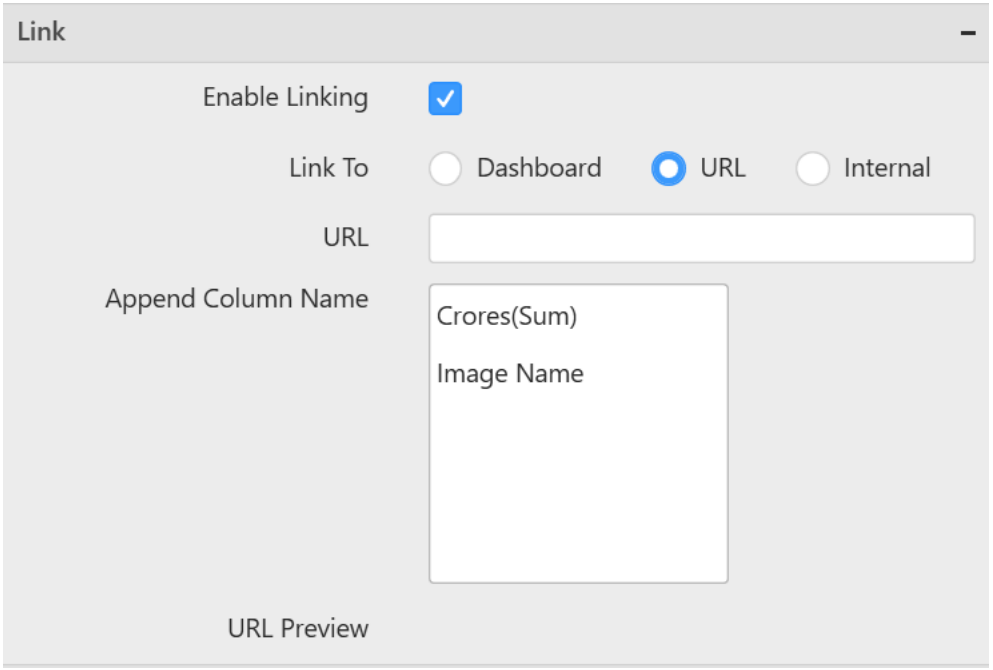
If the widget need to configured with navigation support, then linking can be used.

Format

enableLinking: "Linking can be enabled or disabled for the widget"

Example

enableLinking: "true"



Rules

It is not a mandatory field for custom widget

Link settings

If you need to configure each bounded column with unique links, then enableTabularUrlLinking of linkSettings can be used.

Format

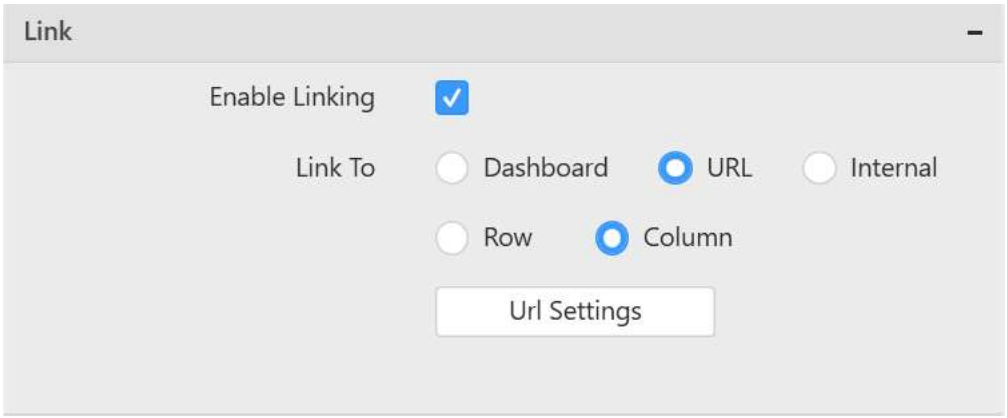
JS

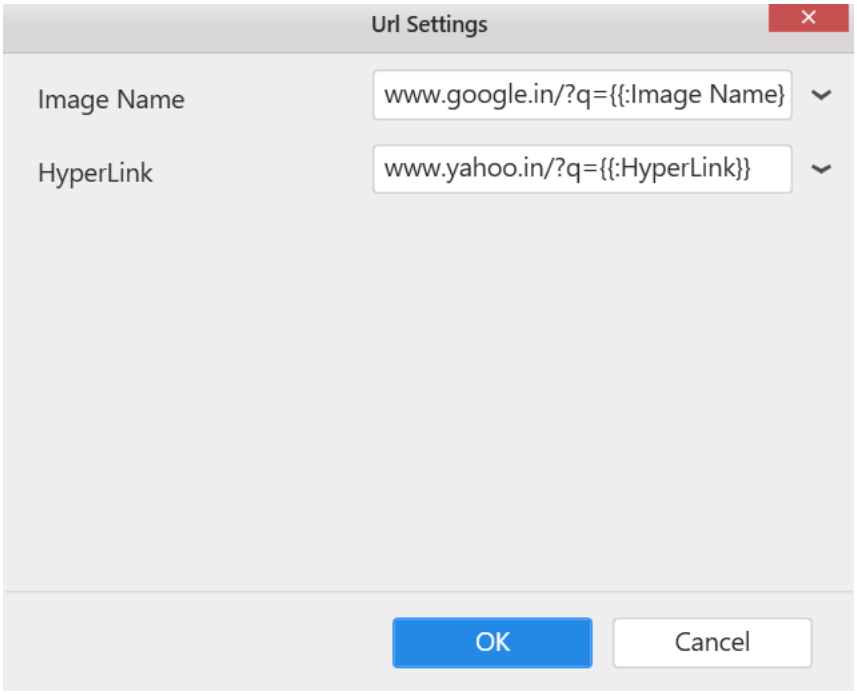
```
"linkSettings": {
enableTabularUrlLinking: "tabular url linking can be enabled or disabled"
}
```


Example

JS

```
"linkSettings": {
enableTabularUrlLinking: "true"
}
```





Rules

- It is not a mandatory field for custom widget.

Packing the widget

After configuring the widget, you can pack the widget by using pack comments.

To import the custom widget in the dashboard designer, you should pack the widget in the .sycw format. The pack command is used to pack the widget which can be imported.

Format

pack [path of the widget to be packed]

Example

Specify the root path of the widget to be packed in the pack command.

![Custom*widget](images/Custom*widget folder.png)

SunburstChart.sycw file will be created in the output folder as below:

![Custom*widget Sunburst chart](images/Custom*widget sunburst folder.png)

Rules

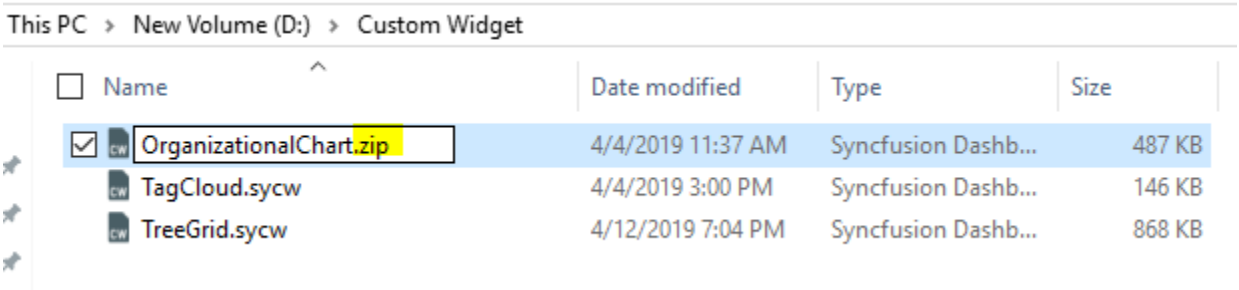
- If you need to modify the widget, you can modify the widget and import it again in the dashboard designer to reflect the changes.

Unpacking the widget

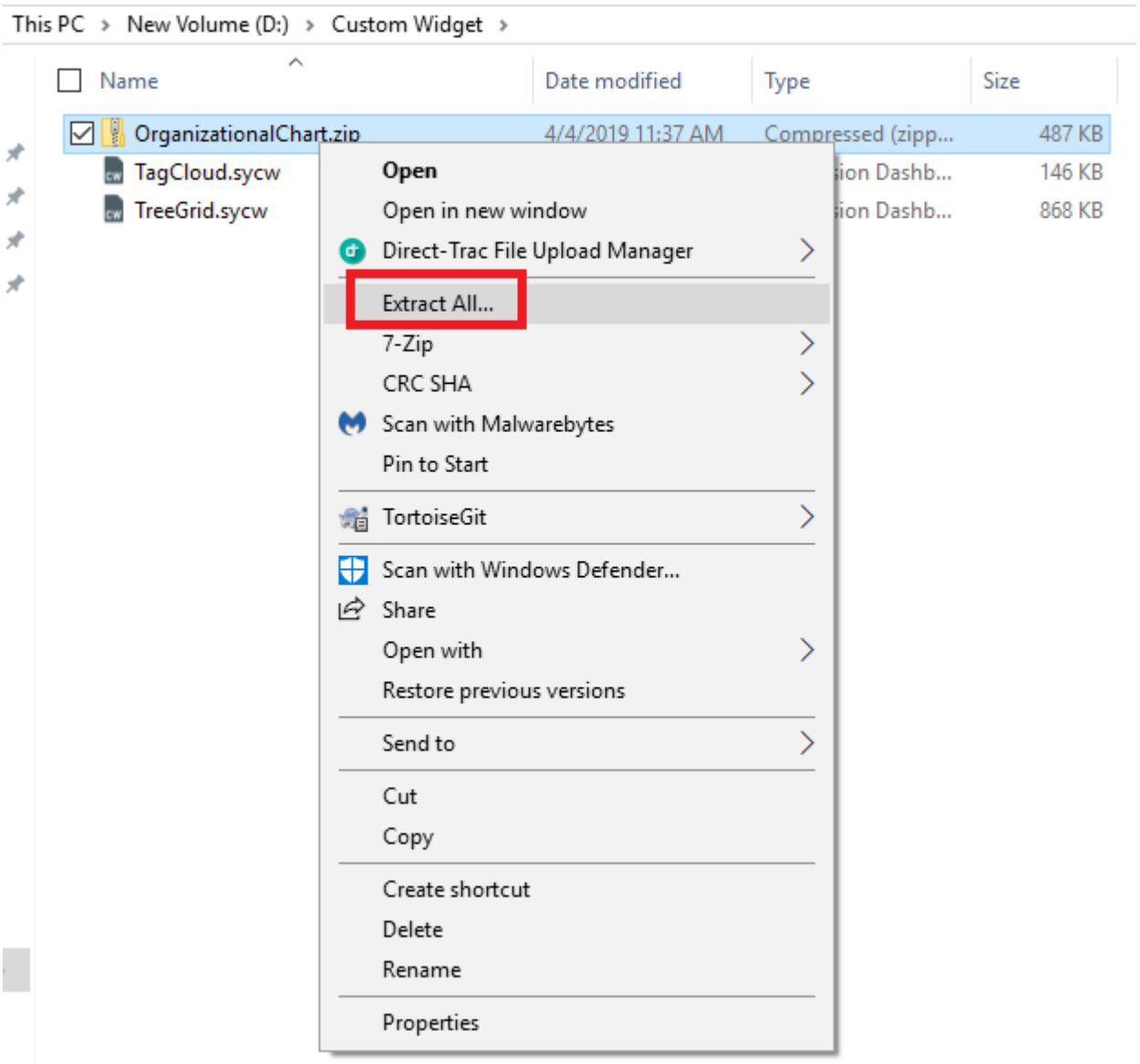
To modify or to reconfigure the custom widget, you should unpack the custom widget which is in the .sycw format.

Steps to unpack the custom widget

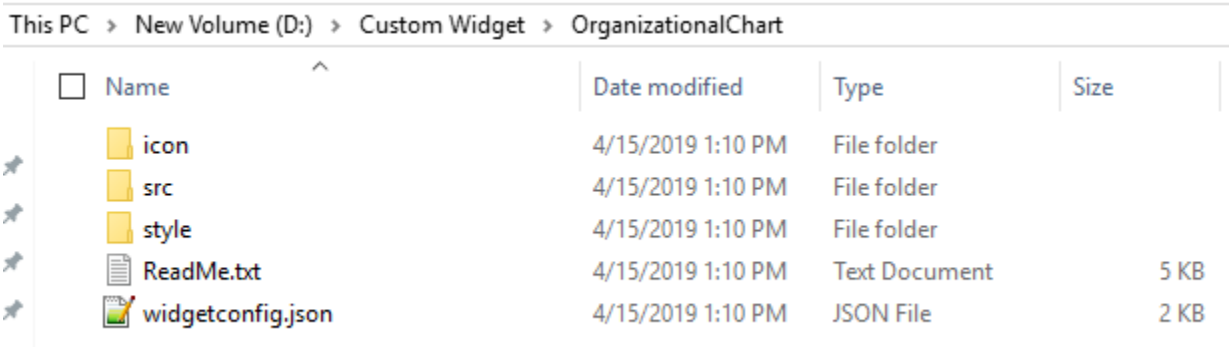
1. Rename the extension of the provided '.sycw' into '.zip' extension as shown in the below image.



2. Now right click on the renamed widget as in the image. Click 'Extract All' to view the source file.



3. You can find the unpacked Organization chart custom widget and it's source file as in the below image.



Rules

- If you need to modify the widget, you can modify the widget and import it again in the dashboard designer to reflect the changes.

Importing widget in designer

From the 'Widget' menu in the designer, select 'Manage Custom Widgets' to import, rename, and delete the widgets. You can import the custom widget file with the extension *.sycw through the add widget option.

![Custom*widget Import](images/custom*widget mange custom.png)

![Custom*widget Import](images/custom*widget manage 2.png)

Debug the widget through HTML file

After importing the custom widget in the designer, you can debug the widget in following cases:

- At initial rendering through the init method.
- Updation request through update method.

Render initially through the init method

![Custom*widget init method](images/custom*widget init method.png)

Update widget through update method

Update request for the widget will be triggered in the following cases.

- On widget container resizing.

![Custom*widget resize](images/custom*widget resize.png)

- On datasource update

![Custom*widget update](images/custom*widget update.png)

- On property change

![Custom*widget change](images/custom*widget change.png)

After initial rendering of the widget, you can debug the property through the browser. Use the following steps for debugging:

- Drag and drop the imported widget in the designer canvas, after this HTML file will be created at the location **%appdata%\Syncfusion\DashboardDesigner\versionnumber\Html\[custom widget report_id].html**
- Run the custom widget HTML file.
- Change the needed property like in the below specified format.
- You can debug the changed property in the custom widget source file.

Format

JS


```
ej.dashboard.setProperty(document.getElementById("container"),{ "name":
"name of the property to be changed", "value": "value of the property" });
```

document.getElementById("container") – widget container element.

name - name of the property specified in the widgetconfig.json under functionalities properties.

value – depends on the control type specified in the widgetconfig.json under functionalities properties.

![Custom*widget Sunburst functions](images/custom*widget functionalities.png)

Example

JS

```
ej.dashboard.setProperty (document.getElementById("container"),{ 'name':
'showTitle, 'value': true});
```

![Custom*widget show title](images/custom*widget show title.png)

Rendering widget through the source file

Source file is used to embed the user defined widget within the dashboard. Refer the below API available in the source file.

Format

JS

```
ej.dashboard.registerWidget(
{
guid : "Specifies the GUID used in widgetconfig.json",
widgetName : "Specifies the name of the widget used in widgetconfig.json",
init : function () {
/* init method will be called when the widget is initialized. */
},
update : function (option) {
update method will be called when any update needs to be performed in the
widget. */
if (option.type == "resize") {
/* update type will be 'resize' if the widget is being resized. */
}
else if (option.type == "refresh") {
/* update type will be 'refresh' when the data is refreshed. */
}
else if (option.type == "propertyChange") {
update type will be 'propertyChange' when any property value is changed in
the designer. */
}
}
});
```

Example

JS

```
ej.dashboard.registerWidget(
{
guid : "b0d5348d-f625-4b78-8db9-c5ed9d38eb45",
widgetName : "SunburstChart",
```



```
init : function () {
var widget =  document.createElement("div").setAttribute("id",
this.element.getAttribute("id") + "_widget");
widget.innerHTML = "Widget is created successfully";
this.element.appendChild(widget);
},
update : function (option) {
var widget = document.getElementById(this.element.getAttribute("id") +
"_widget");
if(option.type == "resize") {
widget.innerHtml = "Widget is resized."
}
else if (option.type == "refresh") {
widget.innerHtml = "Widget data need to be refreshed."
}
else if (option.type == "propertyChange") {
widget.innerHtml = "Widget property have been changed."
}
}
}
);
```

The following information will be available in this scope.

Syntax	Uses
this.element	this.element Container div element in which the custom widget should be embedded.
this.model.dataSource	Holds the datasource for the custom widget in JSON format
this.model.boundColumns	Holds the information about the bounded column to the control. • columnName denotes the actual column name in the data source. • uniqueColumnName should be used at the time of interaction. • dataType of the column. • dateTimeFormat holds the datetime format like dd/mm/yyyy etc.
this.model.boundColumns.datafield[columnIndex].columnName	
this.model.boundColumns.datafield[columnIndex].uniqueColumnName	
this.model.boundColumns.datafield[columnIndex].dataType	
this.model.boundColumns.datafield[columnIndex].valueType	
this.model.boundColumns.datafield[columnIndex].dateTimeFormat	
this.model.boundColumns.datafield[columnIndex].dateTimeComponent	
this.model.boundColumns.datafield[columnIndex].culture	
this.model.boundColumns.datafield[columnIndex].currencyCode	
this.model.boundColumns.datafield[columnIndex].decimalPoints	
this.model.boundColumns.datafield[columnIndex].decimalSeparator	

Syntax	Uses
this.model.boundColumns.datafield[columnIndex].groupSeparator this.model.boundColumns.datafield[columnIndex].measureType this.model.boundColumns.datafield[columnIndex].negativeValueFormat this.model.boundColumns.datafield[columnIndex].prefix this.model.boundColumns.datafield[columnIndex].suffix this.model.boundColumns.datafield[columnIndex].summaryType this.model.boundColumns.datafield[columnIndex].unit	• dateTimeComponent like date, datetime, quarter etc. • culture holds the culture value chosen in the formatting window. • currencyCode holds the currency code value for the culture chosen in the formatting window. • decimalPoints holds the number of decimal points value chosen in the formatting window. • decimalSeparator holds the decimal symbol value chosen in the formatting window. • groupSeparator holds the grouping symbol value chosen in the formatting window. • measureType holds the measure type value chosen in the formatting window. • negativeValueFormat holds the negative value format chosen in the formatting window. • prefix holds the prefix value chosen in the formatting window. • suffix holds the suffix value chosen in the formatting window. • summaryType holds the summaryType

Syntax	Uses
	value chosen in the formatting window. • unit holds the unit value chosen in the formatting window.
this.model.properties	Holds the properties window properties.
this.model.baseURL	this.model.baseURL Holds the base url of custom widget package, used to fetch the asserts (images etc..) in the custom widget folder..
this.model.initMode	this.model.initMode Value will be set according to the rendering mode. (designtime - when rendering in designer, runtime -when render in viewer, export â€“ When the widget rendered for export.)
this.model.isDataBound	this.model.isDataBound Boolean value will be set as true when the widget satisfies the Min Max condition in data fields.

Rules:

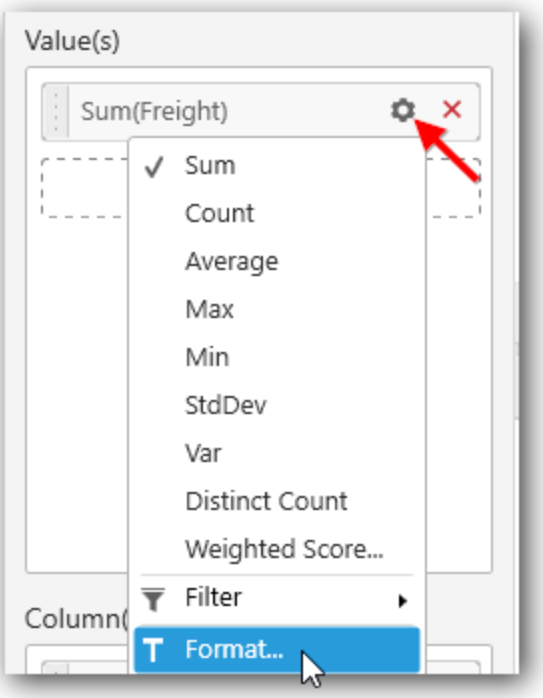
- Widget API's should be in JSON format and need to pass as parameter for **ej.dashboard.registerWidget** method.
- The following are the mandatory API's (**guid**, **widgetName**, **init**, **update**) for the custom widget.

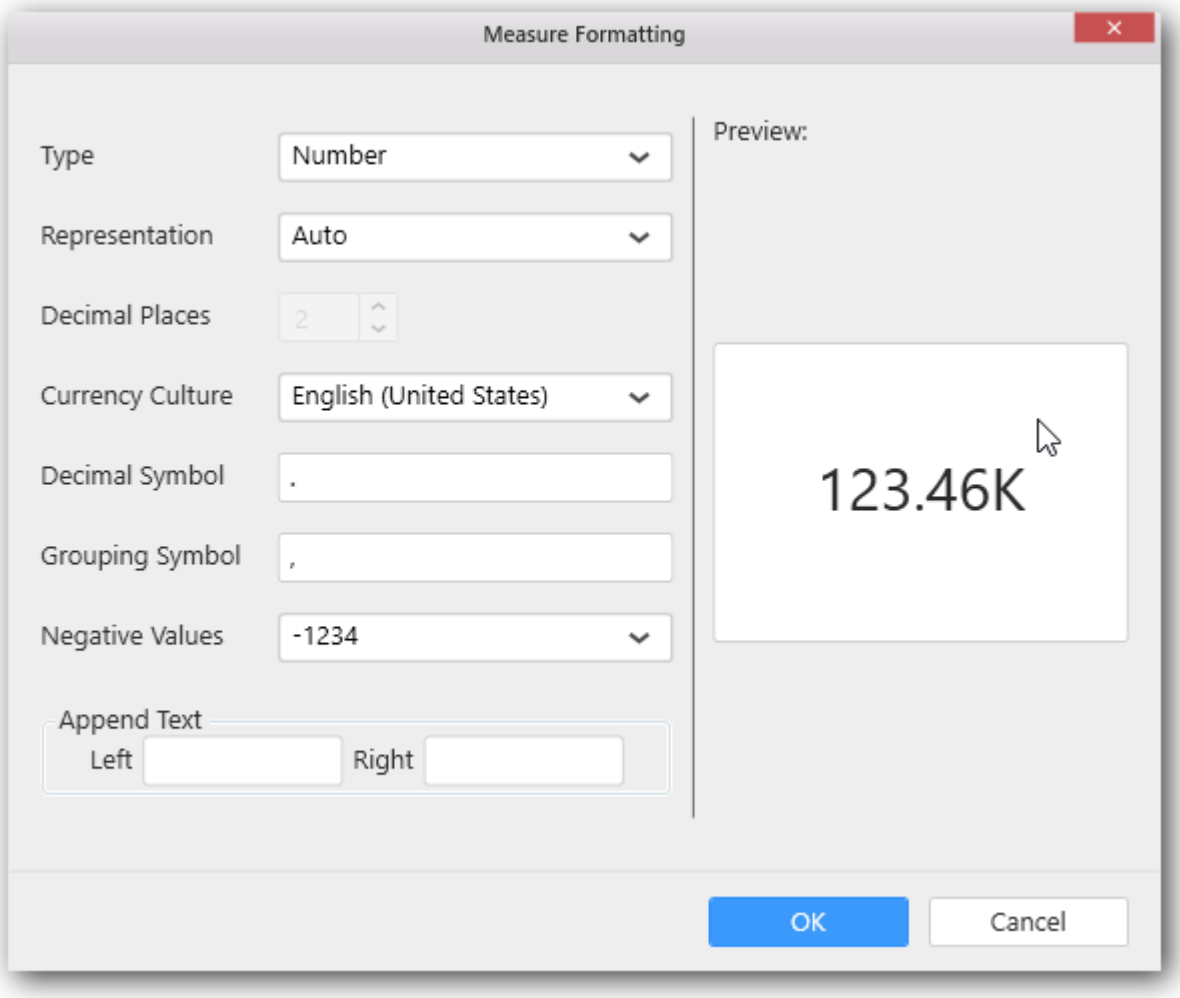
Formatting widget data

The custom widget data can be formatted as like built-in widgets.

Measure Format

Click on the Settings icon and then select Format option to format the measure values.





The values chosen in the formatting window will be stored in the respective column of **this.model.boundColumns**. Refer the above table for API reference.

Measure value can be formatted in two ways.

- formatNumberThroughSchema() - to format the value based on the options selected in the formatting window
- formatNumber() - to format the value based on custom parameters.

formatNumberThroughSchema()

To invoke this method, instance for Internationalization() should be created with the selected culture.

JS

```
var instance = new
ej.dashboard.Internationalization(this.model.boundColumns.columnName[index].
culture);
```

formatNumberThroughSchema() should be invoked using created instance with value to be formatted and the corresponding column information as a method parameter.

JS

```
instance.formatNumberThroughSchema(value,
this.model.boundColumns.columnName[index]);
```

It will return the formatted value based on the options selected in the measure formatting window.

formatNumber()

If we want to format the number with custom parameters then we can make use of formatNumber()

JS

```
instance.formatNumber(value,  {format:measureType, currency:currencyCode,
unit:unitValue, decimal:decimalSeparatorValue, group:groupSeparatorValue,
negative:negativeValueFormat, prefix:prefixValue, suffix:suffixValue});
```

value - Number which we need to format.

currencyCode: Specifies the currency code (used only when the format is set as currency)

decimalSeparatorValue: Specifies the decimal separator.

groupSeparatorValue: Specifies the group separator.

measureType: Specifies the type of the measure like Number, Decimal, Percentage.

Examples: "N0", "C0", "P0" where N, C, P represents Number, Currency and Percentage respectively and "0" represents the number of decimal points.

negativeValueFormat: Specifies the negative value format.

Examples: 0 -> -1234, 1 -> (1234) 2 -> 1234-

prefixValue: Specifies the prefix value.

suffixValue: Specifies the suffix value.

unitValue: Specifies the representation of the value

Possible values: "auto", "ones", "thousands", "millions", "billions" etc.

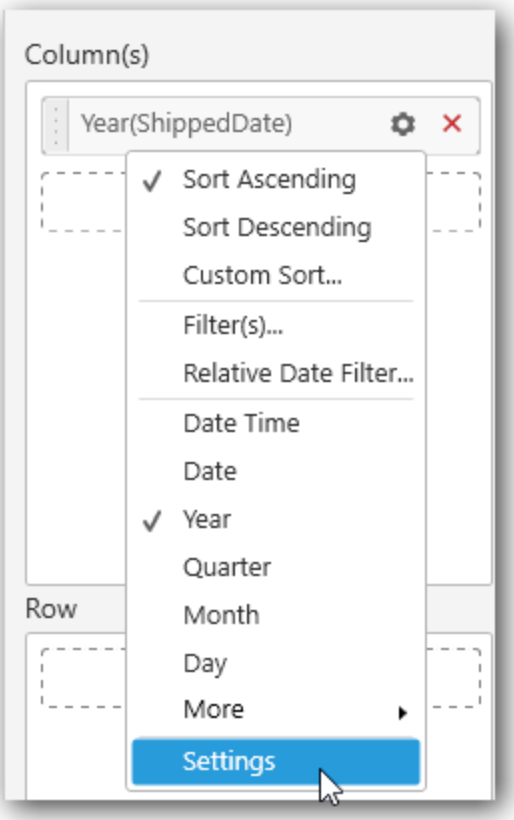
Examples:

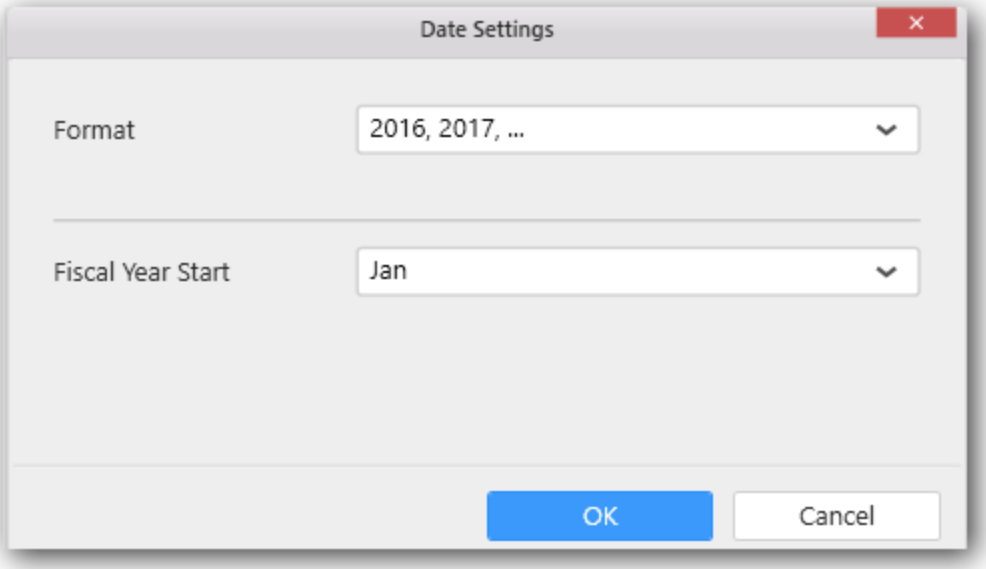
JS

```
var instance = new ej.dashboard.Internationalization("en-US");
instance.formatNumber(123456,  {format:"N2", unit:"ones", decimal:"&",
group:" ", negative:0, prefix:"s", suffix:"Unit(s)"});
// this will return "s123 456&00Unit(s)"
instance.formatNumber(123456,  {format:"C2", currency:"USD", unit:"ones",
decimal:"&", group:" ", negative:0});
// this will return "$123 456&00"
instance.formatNumber(123456,  {format:"P2", unit:"ones", decimal:"&",
group:" ", negative:0});
"
// this will return "12 345 600&00%"
```

Date Formats

Click on the Settings icon and select Settings option to format the date values





The values chosen in the formatting window will be stored in the respective column of **this.model.boundColumns**. Refer the above table for API reference.

Date values can be formatted in two ways.

- formatDateThroughSchema() - to format the date value based on the options selected in the date settings window
- formatDate() - to format the date value value based on custom format.

formatDateThroughSchema()

To invoke this method, instance for Internationalization() should be created.

JS

```
var instance = new ej.dashboard.Internationalization();
```

formatDateThroughSchema() should be invoked using created instance with value to be formatted and the corresponding column information as a method parameter.

JS

```
instance.formatDateThroughSchema(dateValue.toString(),
this.model.boundColumns.columnName[index]);
```

It will return the formatted date value with the based on option selected in the date settings window.

formatDate()

In case if we format the date with a custom format then we can make use of formatDate()

JS

```
instance.formatDate(dateObj,{format:dateFormat});
```

Examples:

JS

```
var date = new Date();
date.setYear(2016);
instance.formatDate(date,{format:"yy"});
// this will return "16"
date.setMonth(2);
instance.formatDate(date,{format:"MMMM"});
// this will return "March"
date.setDate(27);
this.formatDate(date,{format:"dddd"});
// this will return "Tuesday"
```

Configuring widget for interaction

The custom widget can take part in the filter interaction like existing widgets. Refer the following API's to perform the communication between the widgets.

Format

JS

```
var selectedColumnsFilter = [];
var filterColumn = new ej.dashboard.selectedColumnInfo();
filterColumn.condition = "condition";
filterColumn.uniqueColumnName = "unique column name";
filterColumn.values =["value1", "value2", "value3"…] ;
```



```
selectedColumnsFilter.push(filterinfo);
……..
……..
ej.dashboard.filterData(this, selectedColumnsQuery); /*
selectedColumnsFilter is the list of selected column queries send from
custom widget for interaction. */
```

Various types of column

- Dimension type column

Values in the dimension column will be a string, date related, or boolean format.

- Measure type column

Values in the measure column will be in number format.

Date type of column

Data type of a bounded column can be obtained through the below API.

Format

JS

```
this.element.boundColumns.[data field name][index].dateType
```

Example

JS

```
this.element.boundColumns.Levels.[0].dateType
```

Dimension type column

Various conditions available for the dimension type column other than the DateTime data type:

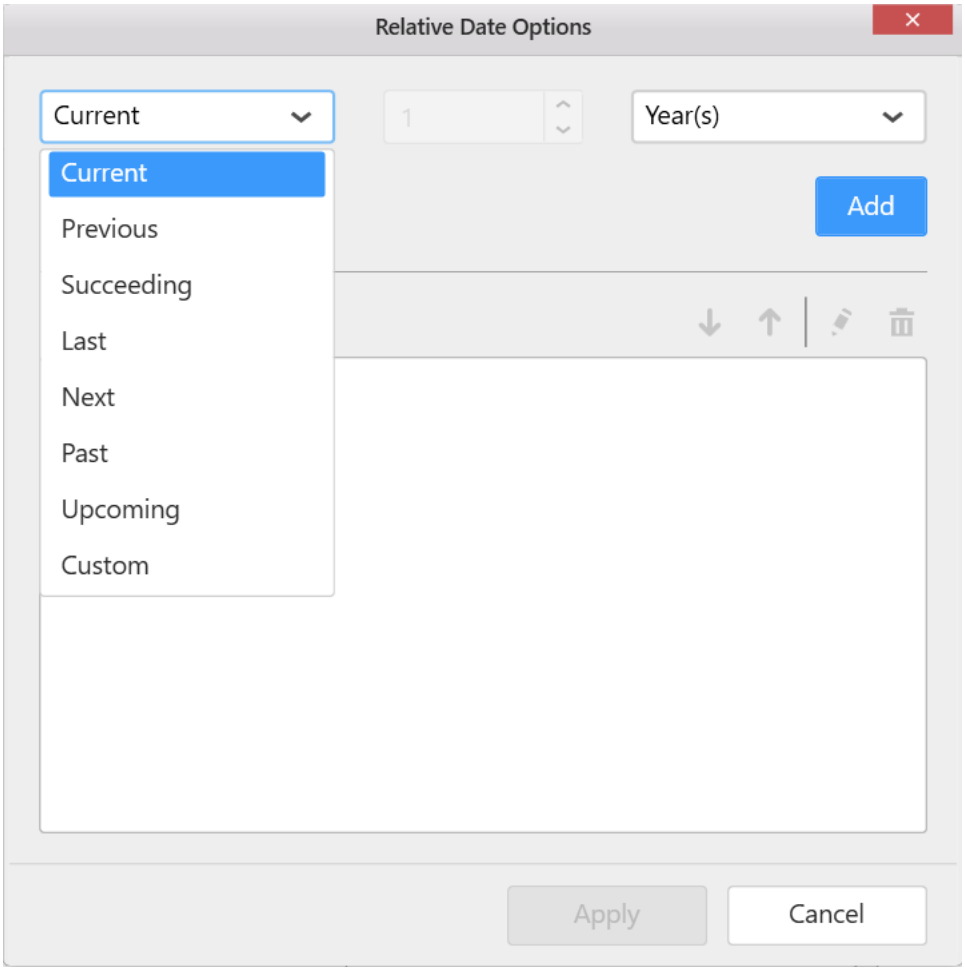
- **include**
- **exclude**

Various conditions available for the dimension type column with the DateTime data type:

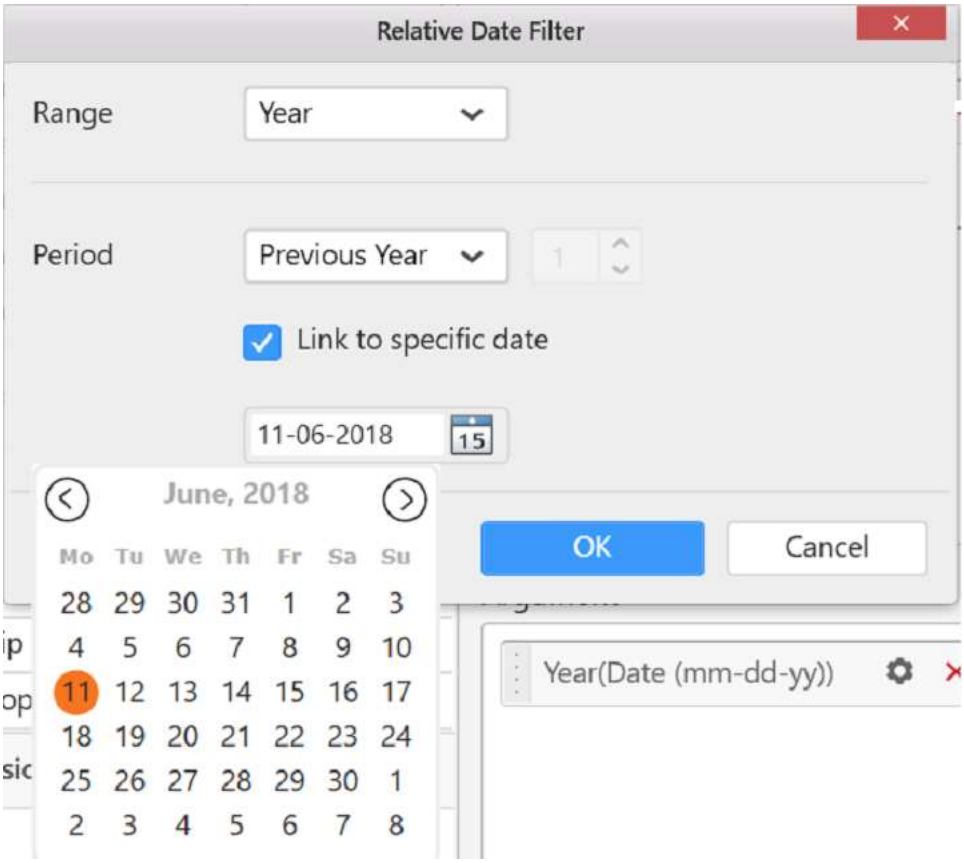
- **range**
- **include**
- **exclude**
- **relative**

Note: If the widget has only one data field block and its limit is one, the Relative Dates will be enabled. Or else Relative Date Filter will be enabled in Dashboard Designer.

Relative Dates:



Relative Date Filter:



Example for dimension to filter the selected value.

JS

```
var selectedColumnsFilter = [];
var filterColumn = new ej.dashboard.selectedColumnInfo();
filterinfo.condition = "include";
filterinfo.values =["india", "china"] ;
selectedColumnsFilter.push(filterColumn);
ej.dashboard.filterData(this, selectedColumnsFilter);
```

Measure type column

Various condition types for measure type

- **equals**
- **notequals**
- **lessthan**
- **greaterthan**
- **lessthanorequals**
- **greaterthanorequals**
- **isbetween**
- **isnotbetween**

Example for measure to specify the in-between condition.

JS

```
var selectedColumnsFilter = [];
var filterColumn = new ej.dashboard.selectedColumnInfo();
filterColumn.condition = "isbetween";
filterColumn.uniqueColumnName =
this.model.boundColumns.Value[0].uniqueColumnName;
filterColumn.values =[12,100] ;
selectedColumnsFilter.push(filterColumn);
ej.dashboard.filterData(this, selectedColumnsFilter);
```

Configuring widget for linking

You can configure to navigate either to a published dashboard or to a general URL with or without parameters. For more information, Linking.

Format

JS

```
// Following code should be implemented in sourcefile.js.
ej.dashboard.navigateThroughLinking(this, selectedColumnsInfo); /*
selectedColumnsInfo is the list of selected column queries send from custom
widget for linking. */
For more information about selectedColumnsInfo, Refer
[FilterInteraction](#configuring-widget-for-interaction).
```

Example

JS

```
var selectedColumnsFilter = [];
var filterColumn = new ej.dashboard.selectedColumnInfo();
filterinfo.condition = "include";
filterinfo.values =["india", "china"] ;
selectedColumnsFilter.push(filterColumn);
ej.dashboard.navigateThroughLinking(this, selectedColumnsFilter);
```

Configuring widget for tabular url linking

By using tabular linking, we can configure unique links to each bounded columns of widget.

Example

JS

```
/* Code to find the unique link configured with selected value
and it should be implemented in sourcefile.js file*/
if (this.model.linkSettings.enableTabularUrlLinking) {
for (var i = 0; i < this.model.linkSettings.urlPatterns.length; i++) { //
urlPatterns contains bounded column details along with configured links.
if (this.model.linkSettings.urlPatterns[i].uniqueColumnName ===
selectedFilterInfos[0].uniqueColumnName ) {
configuredUrl = this.model.linkSettings.urlPatterns[i].urlLink;
}
}
}
ej.dashboard.navigateThroughLinking(this,selectedFilterInfos,configuredUrl);
/* configuredUrl is link configured to selected column in widget */
```


Samples

We have created the following custom widgets from Syncfusion controls. You can download the sample custom widgets from below links:

- Sparkline.sycw
- SunburstChart.sycw
- Radar.sycw
- Waterfall.sycw

Adding Syncfusion ejSunburstChart as custom widget

Follow the below steps to add the sunburst chart as custom widget for the dashboard. You can download the actual widget from this link.

- Refer the commands to create new custom widget files.

![Custom*widget Sunburst](images/custom*widget sunburst folder 2.png)

- Configure the widget through the widgetconfig.json file.
- Description - You can provide the Description about the Sunburst chart.
- SrcFile - Path of the source file which is required for the Sunburst Chart.
- Dependencies - Dependency script and CSS files should be specified in the dependencies section.

JS

```
"description": "Sunburst Chart is useful for visualizing hierarchical
data.",
"srcFile": "src/sunburst.js",
"dependencies": {"scripts": [ "src/ej.sunburstchart.js"] },
```

- As sunburst chart requires two data fields, we have configured **values** and **levels** as shown below and the properties section contains the properties related to the sunburst chart. Refer this link for configuring widgetconfig.json.

JS

```
"dataFields": [
{
"displayName": "Value",
"valueType": "measure",
"name": "Value",
"min": 1,
"max": 1,
"optional": false
},
{
"displayName": "Levels",
"valueType": "dimension",
"name": "Levels",
"min": 1,
```

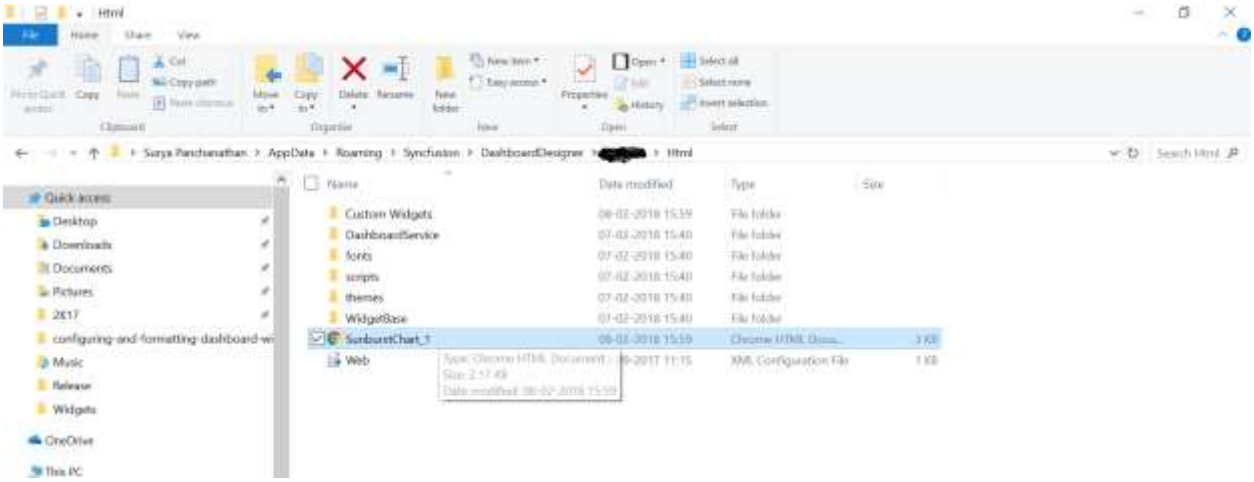


```
"max": 4,
"optional": false
}
],
```

- ej.sunburstchart.js dependency file can be downloaded from this link.
- Pack the widget and import it in the designer.
- Then, configure the widget with the necessary data.

![Custom*widget Sunburst](images/custom*widget schema 2.png)

- Run the sunburstchart_xx.html (%appdata%\Syncfusion\DashboardDesigner\versionnumber\Html) generated in the location, and then you can start rendering sunburst through the **init** method in the source file.



- **this.element** is the container(div) provided to render our actual widget.
- So, we are creating our sunburst chart in **this.element** by using the data in **this.model.datasource**.
- In renderSunburstChart(), We are creating a new space (div) for rendering our sunburst chart custom widget and appending into the container as you can see in the below code.

JS

```
renderSunburstChart : function () {
this.widget = document.createElement("div");
this.widget.setAttribute("id", this.element.getAttribute("id") + "_widget");
this.element.appendChild(this.widget);
```

- The width and height of the widget is assigned from the container's width and height respectively.

JS


```
$(this.widget).css({"width":$(this.element).width(),
"height":$(this.element).height()});
```

- Configure the datasource with some default values for initial rendering of sunburst Chart. **groupMemberPath** and **levels** are the API's of ejSunburstChart where we have to configure the **uniqueColumnName** of the columns that we bind in the levels section.

JS

```
var dataSource = [{ Item: "Item1", Value: 50 },{ Item: "Item2", Value: 60
},{ Item: "Item3", Value: 70 },{ Item: "Item4", Value: 80 },{ Item: "Item5",
Value: 90 },{ Item: "Item6", Value: 90 },{ Item: "Item7", Value: 90 },{
Item: "Item8", Value: 90 },{ Item: "Item9", Value: 90 },{ Item: "Item10",
Value: 90 },{ Item: "Item11", Value: 90 }];
var levels = [];
var valueMember = "Value";
levels = [ { groupMemberPath: "Item" }];
```

- this.model.boundColumns.value and this.model.boundColumns.length contains the columns bounded in the values and levels respectively. By checking the below condition, we can find whether the widget is configured with columns or not.
- Using for loop, the groupMemberPath value for each levels is assigned.

JS

```
if (this.model.boundColumns.Value.length > 0 &&
this.model.boundColumns.Levels.length > 0) {
levels = [];
dataSource = this.model.dataSource;
valueMember = this.model.boundColumns.Value[0].uniqueColumnName;
for (var level = 0; level < this.model.boundColumns.Levels.length; level++)
{
levels.push({ groupMemberPath :
this.model.boundColumns.Levels[level].uniqueColumnName });
}
}
```

- Properties for the custom widget will be available in **this.model.properties**. For example showLegend, titleText, animationType, etc.
- sunburst chart is rendered by configuring the required API'S and assign the values from the **this.model.properties**

JS

```
$(this.widget).ejSunburstChart({
dataSource: dataSource,
valueMemberPath: valueMember, levels: levels,
tooltip: { visible: true},
margin: (this.marginVisibility())? { left: 10, top: 10, bottom: 10, right:
10} :{ left: 0, top: 0, bottom: 0, right: 0} ,
```



```
border: { width: (this.marginVisibility())? 2:0 },
load: $.proxy(this.sunburstChartLoad),
enableAnimation: (this.model.initMode === "export") ? false :
this.model.properties.enableAnimation,
animationType: this.getAnimationType(this.model.properties.animationType),
innerRadius: 0.2,
title: { text:  this.model.properties.titleText , visible:
this.model.properties.showTitle},
zoomSettings:{enable:this.model.properties.enablezoom, zoomOnClick:
this.model.properties.enablezoom  },
dataLabelSettings:{visible:this.dataLabelVisibility(),labelRotationMode:
this.getRotationMode(this.model.properties.dataLabelRotationMode),
labelOverflowMode : "hide", font: { size: "10px" }},
size: { height: $(this.element).height(), width: $(this.element).width()},
legend: { visible:  (this.legendVisibility()) , position:
this.legendPosition() },
pointRegionClick : $.proxy(this.selectionChange,this),
highlightSettings: {enable: true},
selectionSettings: {enable: true, mode : "parent"},
dataLabelRendering : $.proxy(this.dataLabelFormat, this),
legendItemRendering : $.proxy(this.legendFormat, this),
tooltipInitialize : $.proxy(this.tooltipFormat,this)
});
});
```

- For ejSunburstChart related API references, refer the link.

marginVisibility()

- **marginVisibility** function controls the visibility of the margins by returning a boolean value based on the widget and height of the container.

JS

```
marginVisibility : function () {
return ($(this.widget).width() > 200 && $(this.widget).height() > 200);
}
```

getAnimationType()

- **getAnimationType** function returns the type of animation based on the type we select in the widget property window.

JS

```

```


dataLabelVisibility()

- **dataLabelVisibility** function controls the visibility of the data labels by returning a boolean value based on the width and height of the container.

JS

```
dataLabelVisibility : function (e) {
return !($(this.element).width() < 300 || $(this.element).height() < 300);
}
```

getRotationMode

- **getRotationMode** function returns the type based on the type we selected in the widget properties window.

JS

```
getRotationMode : function(rotationMode) {
if (rotationMode === "Normal")
return "normal";
else if(rotationMode === "Angle")
return "angle";
}
```

legendVisibility()

- **legendVisibility** function controls the visibility of the data labels by returning a boolean value based on the width and height of the container.

JS

```
legendVisibility : function (e) {
return ($(this.element).width() > 300 && $(this.element).height() > 300 &&
this.model.properties.showLegend);
}
```

legendPosition()

- **legendPosition** function is used to place the legend position based on the condition **if ($(this.element).width() > $(this.element).height())**. (i.e) If the the size of the container width is greater than the length then the legend displays on **right**. Otherwise, it will display on left of the **widget**.

JS

```
legendPosition : function (e) {
if ( $(this.element).width() > $(this.element).height())
return "right";
else
return "top";
```



```
}
```

- Sunburst widget will be rendered as follows:

![Custom*widget sunburst](images/custom*widget sunburst chart.png)

selectionChange

- Other widgets in the dashboard that are configured as a slave for sunburst chart can be updated based on the selected data in the sunburst chart by using below code. ej.dashboard.filterData(this, selectedFilterInfos) is used to send filter data request.
- On triggering **pointRegionClick** the selectionChange will gets invoked. In selectionChange(), we get the chart data and get the parent element using **findSelectionParent** method and push the parent element into **data[]**. Then create a **selectedFilterInfos** object of **ej.dashboard.selectedColumnInfo()** type with the information present in the data[] and pass the **selectedFilterInfos** into the **ej.dashboard.filterData(this,selectedFilterInfos)**.

JS

```

```

Note: findSelectionParent is a internal API of ejSunburstChart, which is used to get the parent element. It not exposed in the sunburst chart documentation.

dataLabelFormat()

- **dataLabelFormat()** is used to customize the data label based on the formatting options.

JS

```
dataLabelFormat : function(e){
if(this.model.boundColumns.Levels.length > 0 &&
this.model.boundColumns.Value.length > 0 &&
!ej.isNullOrUndefined(e.data.Text)){
var instance = new ej.dashboard.Internationalization();
e.data.Text = this.model.boundColumns.Levels[e.data.level - 1].dataType ==
"datetime" ? instance.formatDateThroughSchema(e.data.Text.toString(),
this.model.boundColumns.Levels[e.data.level - 1]) : e.data.Text ;
}
}
```

legendFormat()

- **legendFormat()** is used to customize the legend text based on the formatting options.

JS

```
legendFormat : function(e){
```



```
if(this.model.boundColumns.Levels.length > 0 &&
this.model.boundColumns.Value.length > 0 &&
!ej.isNullOrUndefined(e.data.text)){
var instance = new ej.dashboard.Internationalization();
e.data.text = this.model.boundColumns.Levels[e.data.level - 1].dataType ==
"datetime" ? instance.formatDateThroughSchema(e.data.text.toString(),
this.model.boundColumns.Levels[e.data.level - 1]) : e.data.text ;
}
}
```

tooltipFormat()

- **tooltipFormat** is used to customize the tooltip text based on the formatting options.

JS

```
tooltipFormat : function (e){
if(this.model.boundColumns.Levels.length > 0 &&
this.model.boundColumns.Value.length > 0 &&
!ej.isNullOrUndefined(e.data.currentText)){
var instance = new
ej.dashboard.Internationalization(this.model.boundColumns.Value[0].culture);
e.data.currentText = this.model.boundColumns.Levels[e.data.level -
1].columnName + " : " + (this.model.boundColumns.Levels[e.data.level -
1].dataType == "datetime" ?
instance.formatDateThroughSchema(e.data.currentText.split(":
")[0],this.model.boundColumns.Levels[e.data.level - 1]) :
e.data.currentText.split(": ")[0] ) + "<br>" +
this.model.boundColumns.Value[0].summaryType + " of " +
this.model.boundColumns.Value[0].columnName + " : " +
instance.formatNumberThroughSchema(e.data.currentText.split(":
")[1]*1,this.model.boundColumns.Value[0]);
}
}
```

- Update method will be triggered for below operations and we have to do the required changes by checking the update types(resize, refresh, propertyChange) and widget will be updated accordingly by invoking the respective API of the widget. 1) Resize 2) Refresh 3) property change

JS

```
update : function (option) {
if(option.type == "resize") {  // option type will be resize when the
browser is resized
this.resizeWidget(option.size);  // widget will be resized through this
method
}
else if (option.type == "refresh") {  // option type will be refresh when
the widget data gets modified
var widgetObj = $(this.element).data("ejSunburstChart");
widgetObj.model.dataSource = this.model.dataSource;  // updating new data to
the control
widgetObj.redraw();
```



```
}
else if (option.type == "propertyChange") {  // option type will be
propertyChange when widget properties changed in the Dashboard Designer
var widgetObj = $(this.element).data("ejSunburstChart");
switch (option.property.name)  // option.property contains the property name
and value of the modified property.
{
case "enableAnimation":
widgetObj.model.enableAnimation = option.property.value;
break;
case "animationType":
widgetObj.model.animationType = option.property.value;
break;
case "titleText":
widgetObj.model.title.text =  option.property.value;
case "showTitle":
widgetObj.model.title.visible =  option.property.value;
break;
case "enablezoom":
widgetObj.model.zoomSettings.enable =  option.property.value;
case "showLegend":
widgetObj.model.legend.visible = option.property.value;
break;
}
widgetObj.redraw(); // widget will be updated based on the property changes.
}
}
```

Note: Find the sample sunburst chart widget from the location C:\ProgramData\Syncfusion\DashboardDesigner\Custom Widgets.

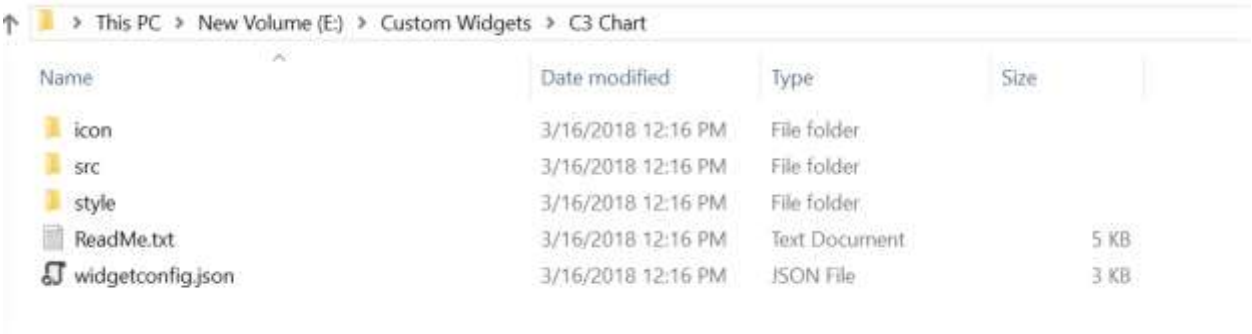
Adding C3 chart as custom widget

Here we are creating C3 Chart using D3 Library.

Note: C3 Chart is an open source and it is under MIT licensed.

Follow the below steps to add the third party widget(C3 Chart)as custom widget for the dashboard.

- Refer the commands to create new custom widget files.



- Configure the widget through the widgetconfig.json file.

- As C3 Chart requires two data fields, we have configured **column** and **values** as shown below and the properties section contains the properties related to the C3 Chart. Refer this link for configuring widgetconfig.json.

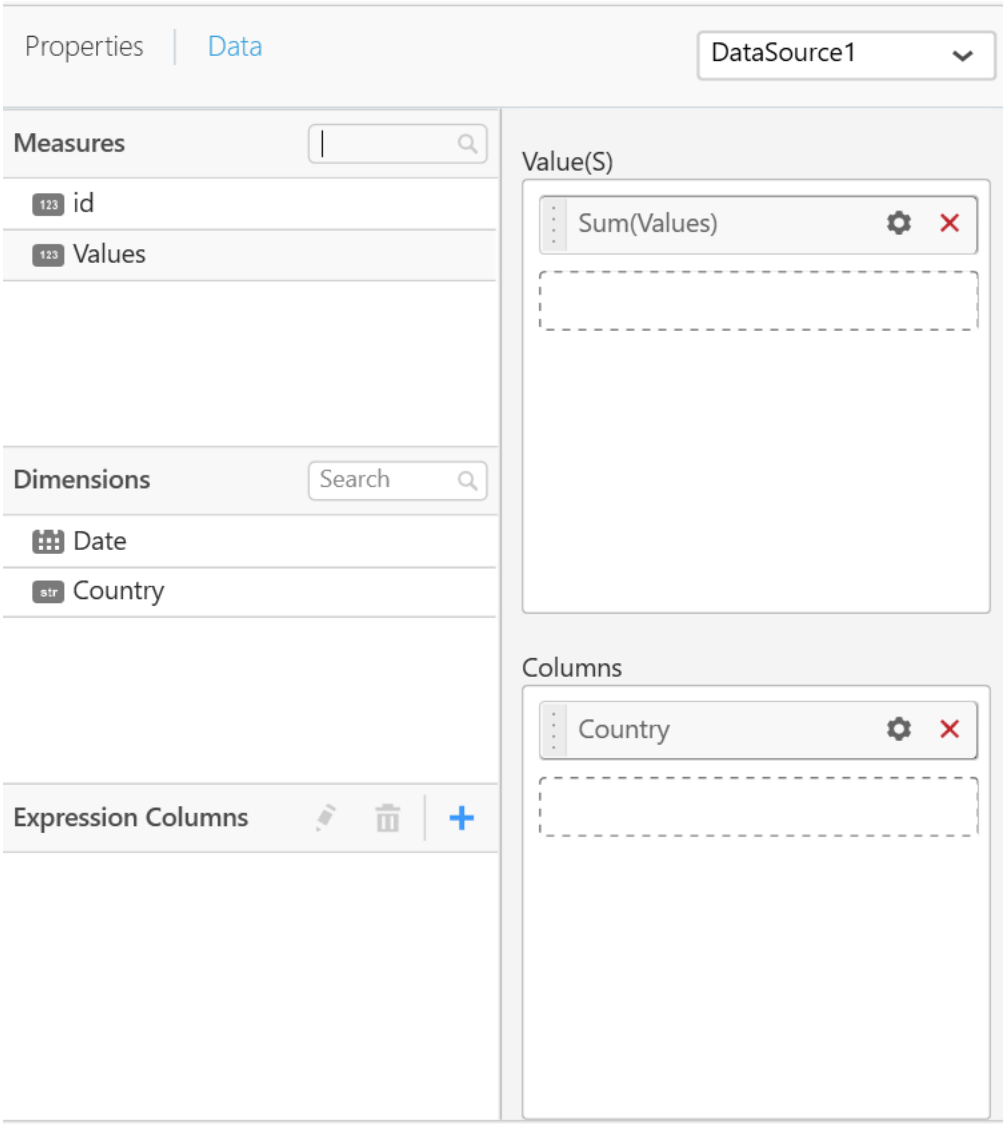
JS

```
"widgetName": "C3Chart",
"displayName": "C3 Chart",
"guid": "b0d5348d-f625-4b78-8db9-c5ed9d38eb45",
"category": "Comparison",
"description": "Compare values for a set of unordered items across
categories",
"srcFile": "src/sourcefile.js",
"dependencies": {
"scripts": [
"src/d3.js",
"src/c3.js"
]
},
"dataFields": [
{
"displayName": "Value",
"valueType": "measure",
"name": "Value",
"min": 1,
"max": 1,
"optional": false
},
{
"displayName": "Column",
"valueType": "dimension",
"name": "Column",
"min": 1,
"max": 1,
"optional": false
}
],
"functionalities": [
{
"header": "Basic Settings",
"properties": [
{
"displayName": "Show Tooltip",
"controlType": "checkbox",
"name": "showTooltip",
"defaultValue": "true"
},
{
"displayName": "Show Label",
"controlType": "checkbox",
"name": "showLabel",
"defaultValue": "true"
},
{
"displayName": "Color",
"controlType": "colorpicker",
```

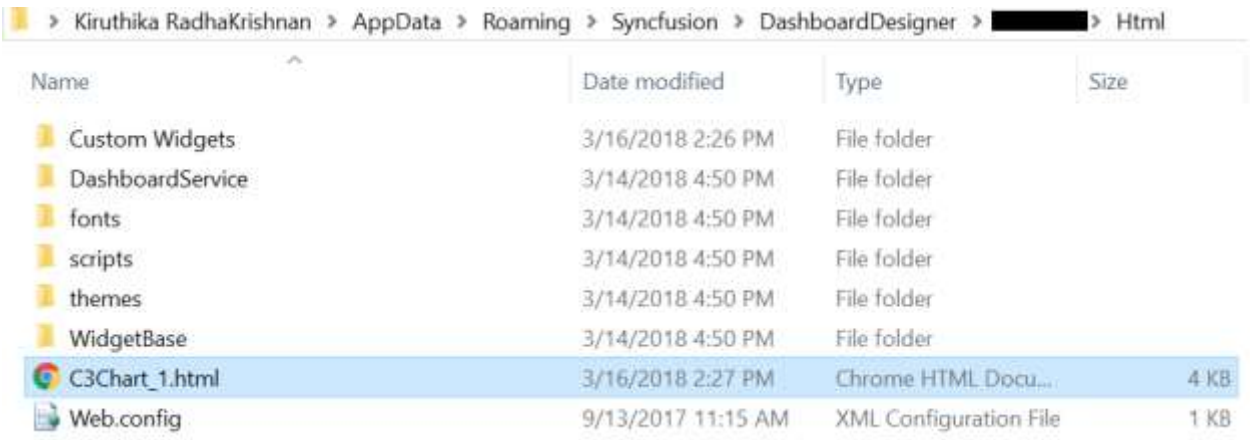


```
"name": "barColor",
"defaultValue": "#2196F3"
}
]
},
],
"filterSettings": {
"masterFilter": {
"visible": true,
"defaultValue": true
},
"ignoreMasterFilter": {
"visible": true,
"defaultValue": false
}
}
```

- d3.js dependency file can be downloaded from D3 website.
- c3.js dependency file can be downloaded from C3 website.
- Pack the widget and import it in the designer.
- Then, configure the widget with the necessary data.



- Run the C3Chart_xx.html (%appdata%\Syncfusion\DashboardDesigner\versionnumber\Html) generated in the location, and place the break point in the **init** method in the source file of the widget.



- **this.element** is the container(div) provided to render our actual widget.
- So, we are creating our C3 Chart in **this.element** by using the data in **this.model.datasource**.
- In Init(), We are creating a new space (div) for rendering our C3 Chart custom widget and appending into the container as you can see in the below code.
- The width and height of the widget is assigned from the container's width and height respectively.

JS

```
init: function () { /* init method will be called when the widget is
initialized */
this.widget = document.createElement("div");
this.widget.setAttribute('id', 'chart');
var width = document.getElementById('container').style.width;
var height = document.getElementById('container').style.height;
this.element.appendChild(this.widget);
var that = this;
```

- Configure the datasource with some default values for initial rendering of C3 Chart.

JS

```
var columns =[];
columns =[{column:'Item1',values: '50'},{column:'Item2',values:
'20'},{column:'Item3',values: '30'}];
```

- this.model.boundColumns.Value and this.model.boundColumns.length contains the columns bounded in the columns and values respectively. By checking the below condition, we can find whether the widget is configured with columns or not.
- Converting datasource structure required by C3 Chart.

JS

```
if (this.model.boundColumns.Value.length > zero) {
for (var i = 0; i < this.model.dataSource.length; i++) {
```



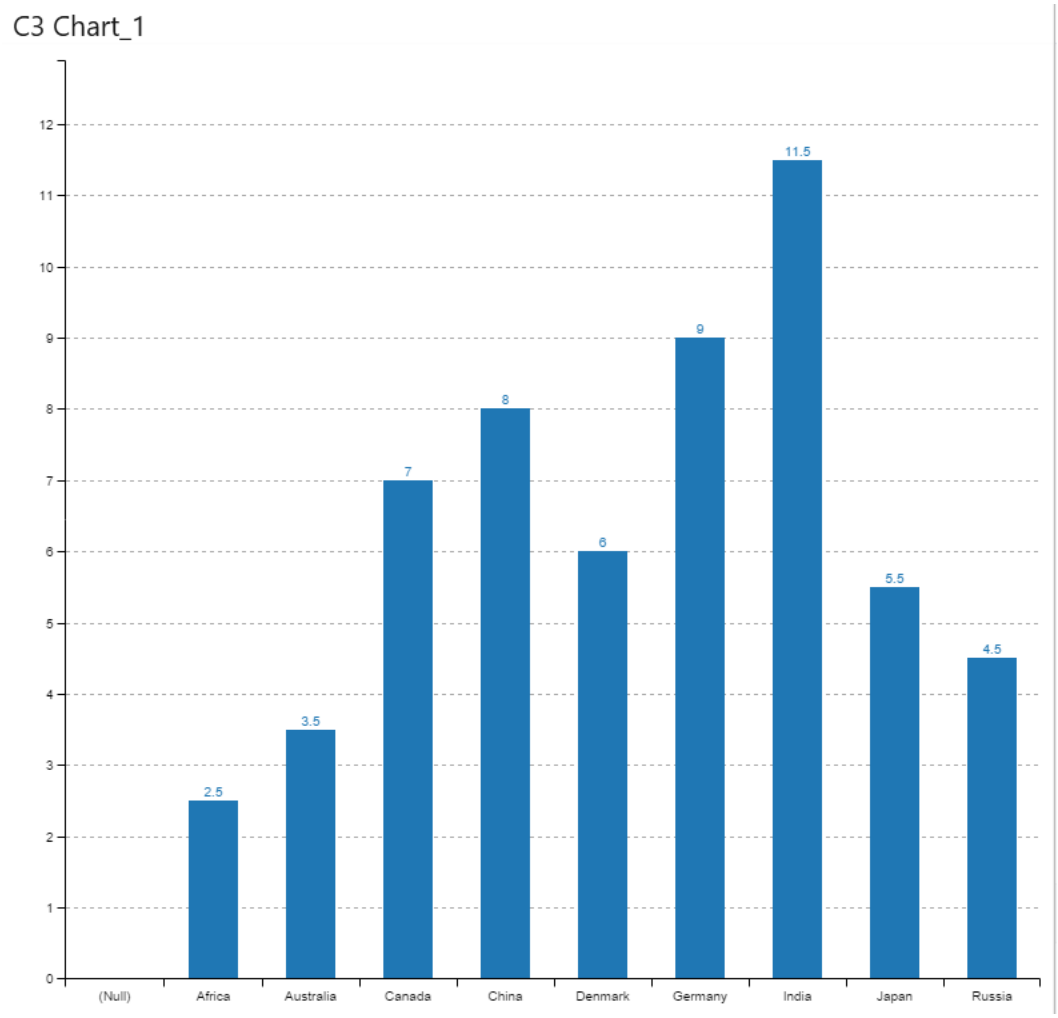
```
columns.push({column:
this.model.dataSource[i][this.model.boundColumns.Column[0].uniqueColumnName]
, values:
this.model.dataSource[i][this.model.boundColumns.Value[0].uniqueColumnName]}
);
}
```

- To render the C3 Chart, configure the required API's and assign the data to it.

JS

```
var chart = c3.generate({
data:
{
type: 'bar',
json: columns,
keys:
{
x: 'column',
value: ['values']
},
labels: true,
selection:
{
enabled: true,
multiple: false,
draggable: true
},
onclick: function(d) {
that.selectionChange(d)
},
},
axis:
{
x:
{
type: 'category'
}
},
grid:
{
x: {show: false},
y: {show: true}
},
bar:
{
width:
{
ratio: 0.5
}
},
interaction: {enabled: true}
});
this.widget = chart;
```


- C3 Chart widget will be rendered as follows:



selectionChange

- Other widgets in the dashboard that are configured as a slave for C3 Chart can be updated based on the selected data in the C3 Chart by using below code. ej.dashboard.filterData(this, selectedFilterInfos) is used to send filter data request.
- On triggering **onclick** the selectionChange will gets invoked. In selectionChange(), we get the chart data and push the selected value into **data[]**. Then create a **selectedFilterInfos** object of **ej.dashboard.selectedColumnInfo()** type with the information present in the data[] and pass the **selectedFilterInfos** into the **ej.dashboard.filterData(this,selectedFilterInfos)**.

JS

```
selectionChange: function (e) {
var data = [];
data.push(this.model.dataSource[e.x][this.model.boundColumns.Column[0].uniqu
eColumnName]);
var zero = 0;
var selectedFilterInfos = [];
if (data.length > zero)
```



```
{
var filterinfo = new ej.dashboard.selectedColumnInfo();
filterinfo.condition = "include";
filterinfo.uniqueColumnName =
this.model.boundColumns.Column[0].uniqueColumnName;
filterinfo.values.push(data[0]);
selectedFilterInfos.push(filterinfo);
}
ej.dashboard.filterData(this, selectedFilterInfos);
},
```

- Update method will be triggered for below operations and we have to do the required changes by checking the update types(resize, refresh, propertyChange) and widget will be updated accordingly by invoking the respective API of the widget. 1) Resize 2) Refresh 3) property change

JS

```
update: function (option) {
var widget = this.widget;
/* update method will be called when any update needs to be performed in the
widget. */
if (option.type == "resize") {
/* update type will be 'resize' if the widget is being resized. */
widget.resize(option.size);
}
else if (option.type == "refresh") {
/* update type will be 'refresh' when the data is refreshed. */
this.refreshChart();
}
else if (option.type == "propertyChange") {
/* update type will be 'propertyChange' when any property value is changed
in the designer. */
switch (option.property.name) {
case "showTooltip":
widget.internal.config.tooltip_show = option.property.value;
break;
case "showLabel":
widget.internal.config.data_labels = option.property.value;
widget.flush();
break;
case "barColor":
widget.internal.config.data_colors.values = option.property.value;
widget.flush();
break;
}
}
}
```

JS

```
refreshChart : function (e) {
var columns =[];
if (this.model.boundColumns.Value.length > 0)
{
```



```
for( var i=0; i< this.model.dataSource.length; i++)
{
columns.push({column:this.model.dataSource[i][this.model.boundColumns.Column
[0].uniqueColumnName],values:this.model.dataSource[i][this.model.boundColumn
s.Value[0].uniqueColumnName]});
}
} else {
columns =[{column:'A',values: '50'},{column:'B',values:
'20'},{column:'C',values: '30'}];
}
var chart = c3.generate({
data: {
type : 'bar',
json: columns,
keys: {
x: 'column',
value: ['values']
},
labels:this.model.properties.show label,
selection:
{
enabled: true,
multiple: false,
draggable:true
},
onclick: function(d) {
that.selectionChange(d)
},
},
axis:
{
x: {
type: 'category'
}
},
grid:
{
x:{ show : true},
y:{show: false}
},
bar:
{
width: { ratio: 0.5 }
},
interaction: {enabled: true}
});
this.widget = chart;
},
```

Aggregating Value Columns Based on Type

Each of the value column configured to widget, can be aggregated individually based on the type you define. Following table illustrates the aggregation types and their use.

Aggregation Type	Result

Sum	Calculates the sum of values of all members.
Count	Returns the count of all members.
Average	Calculates the average of values of all members.
Max	Returns the highest value for all members.
Min	Returns the lowest value for all members.
StdDev	Calculates the standard deviation of all members.
Var	Calculates the variance of all members.
DistinctCount	Returns the distinct count of all members.
Weighted Score	Calculates the weighted average of all members based on another column defined.
None	Allows you to view data without aggregation but grouped

Note: SSAS data source don't have this option enabled for value columns as these columns were pre-aggregated in cube itself.

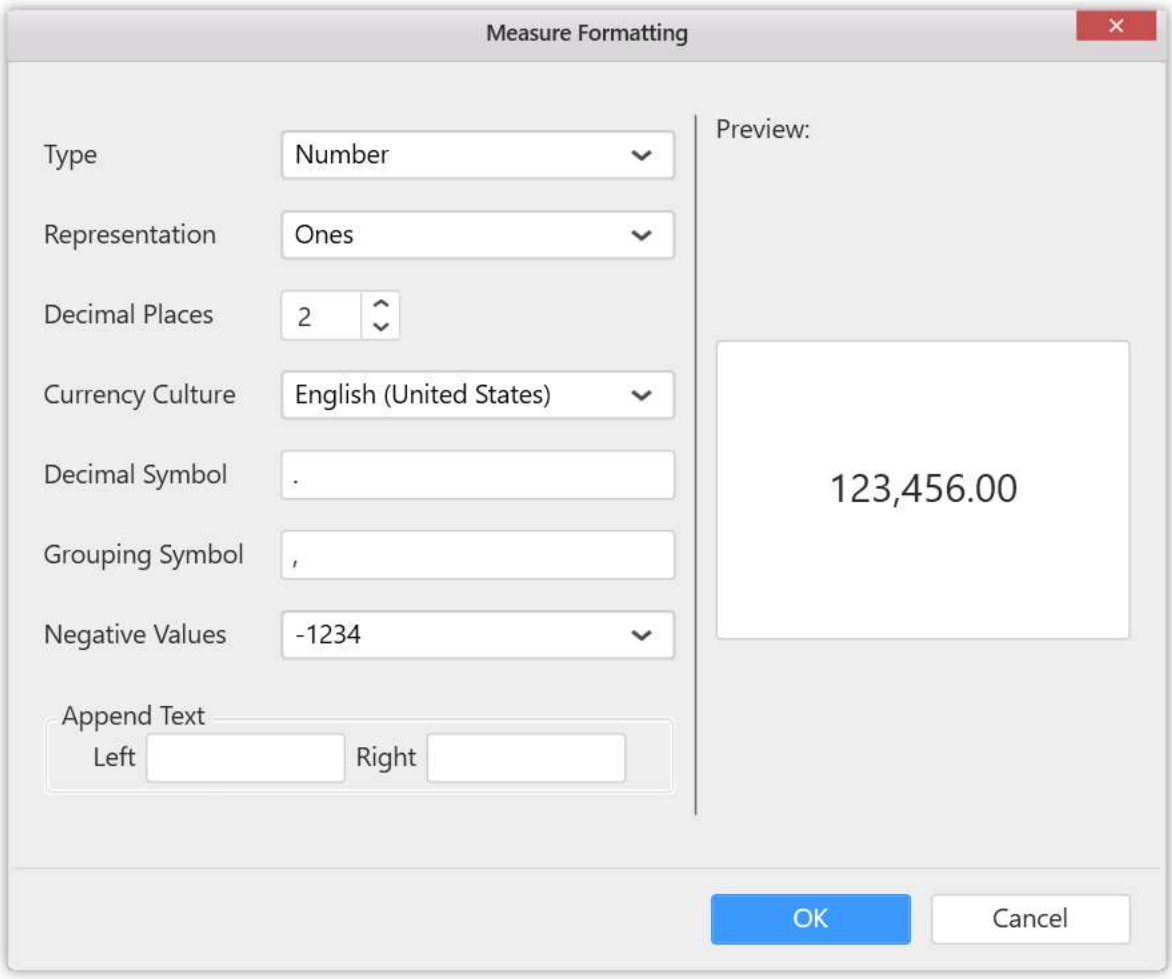
Information: None option is only applicable to the Grid widget and the ODBC ANSI SQL connection in Live Mode does not support this option.

Note: Refer the Configuring and formatting dashboard widgets sections for changing the aggregation types for the added fields in the widgets.

Formatting Measure Type Column

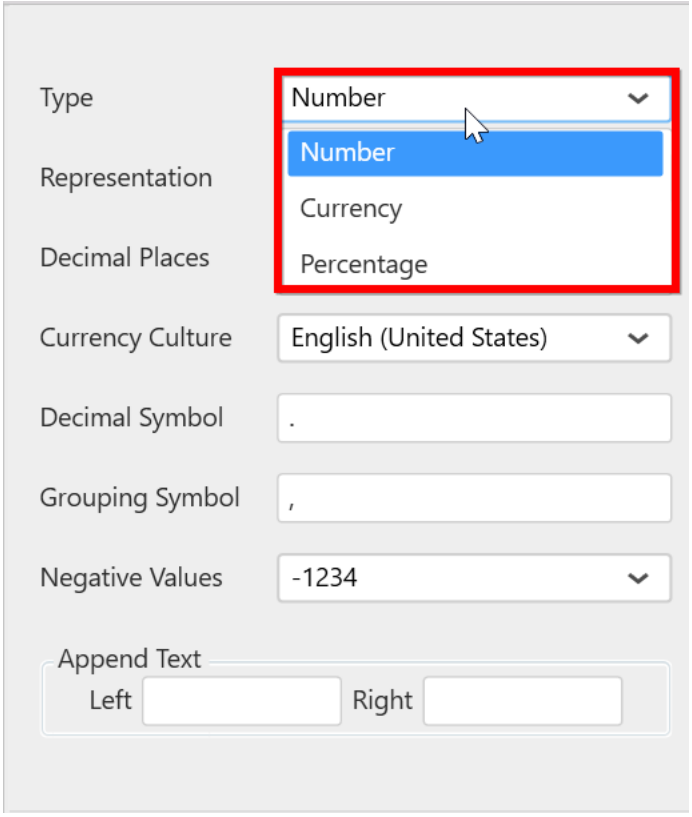
Measure type column values can be formatted based on the following different options:

The measure formatting is displayed in the following screenshot and refer this section for formatting the widgets.



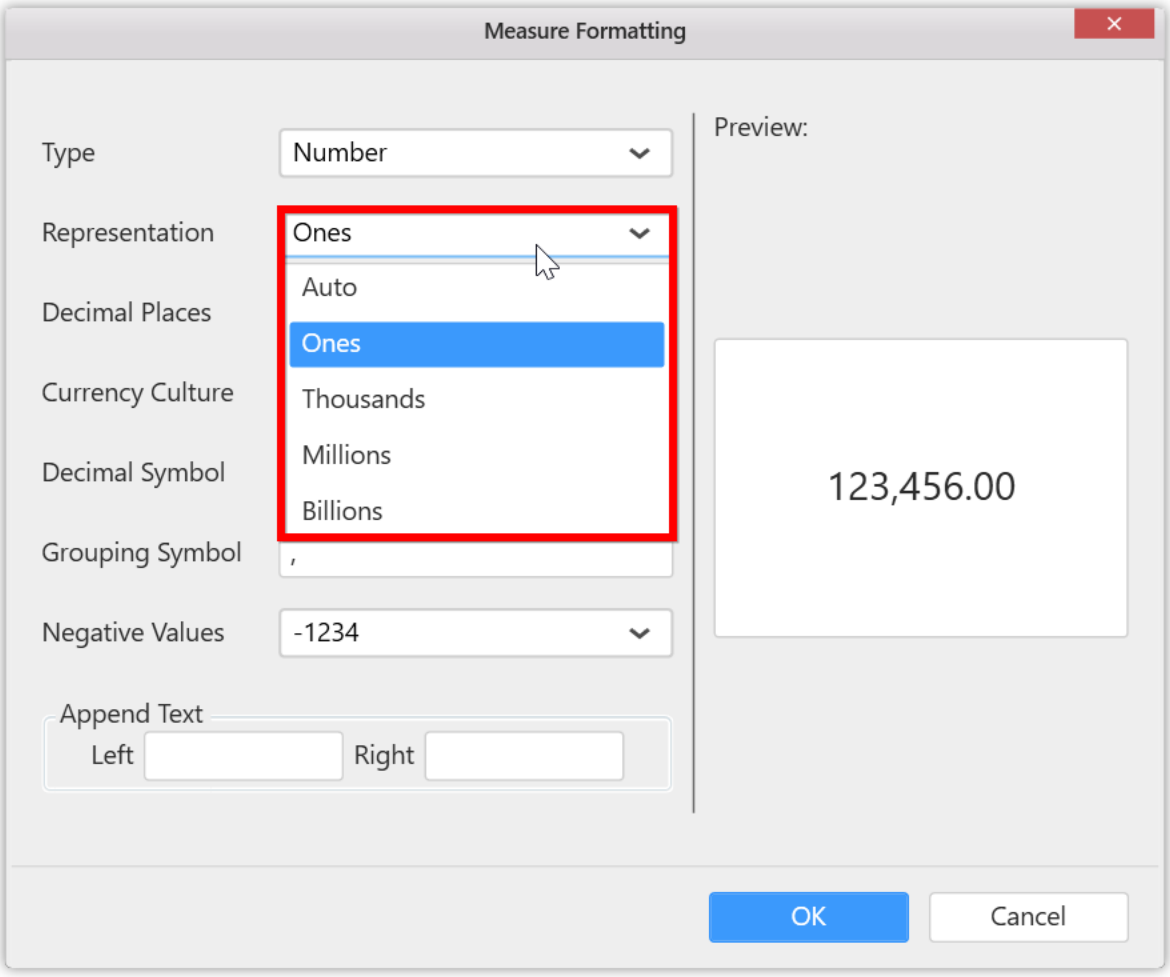
Type

The measure column values display type can be defined through this based on the data displayed. For example, if you are displaying sales amount column, then type can be defined as Currency.



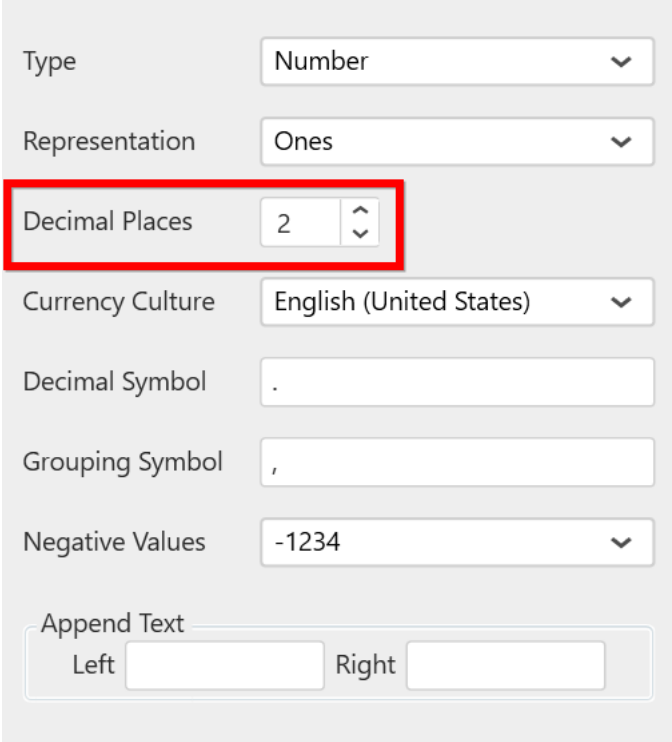
Representation

The value display format representation can be defined through this. For example, through selecting Thousands, value 10,000 will be displayed as 10K.



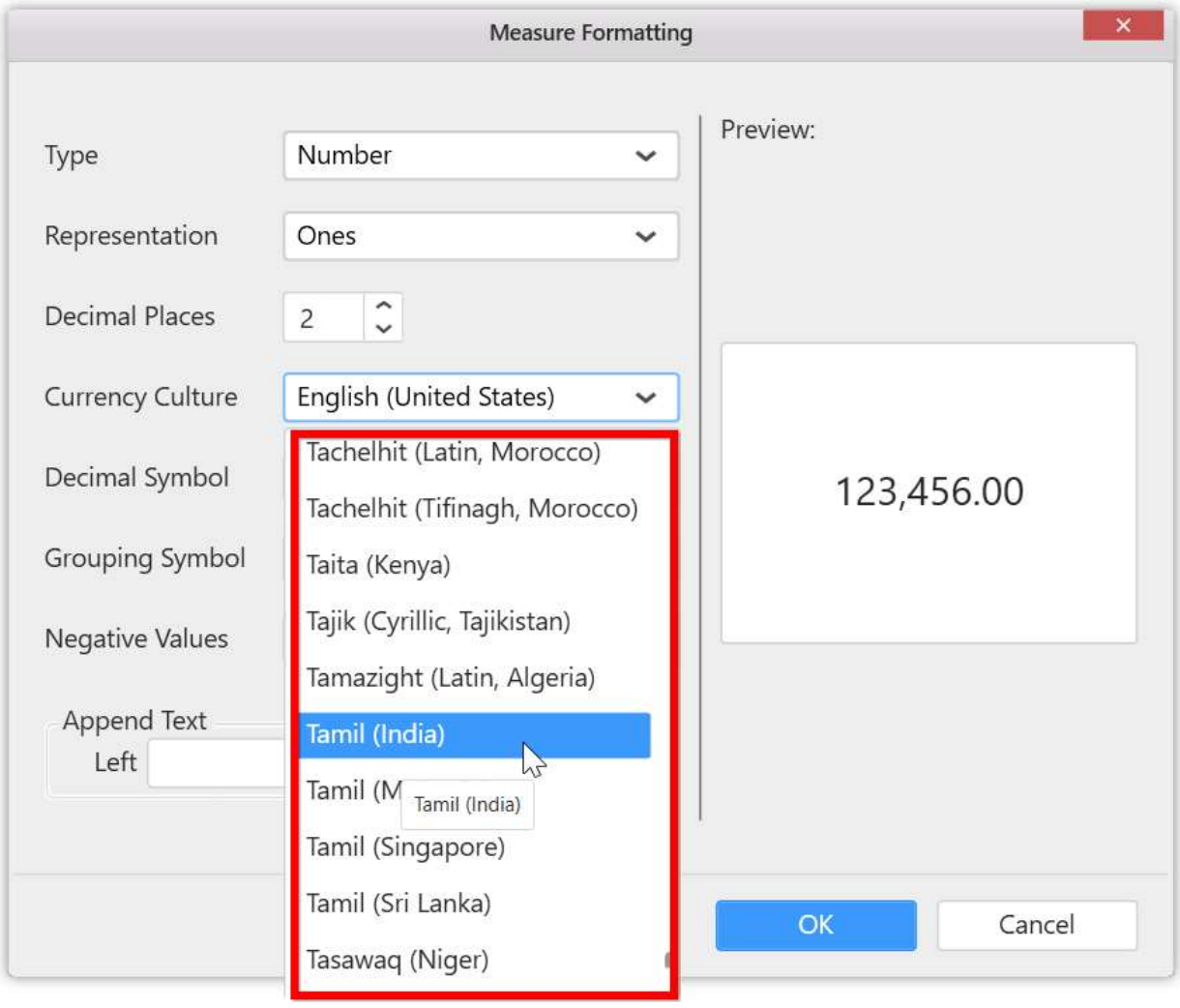
Decimal Places

You can set the decimal places explicitly when the Representation was set with options other than Auto.



Currency Culture

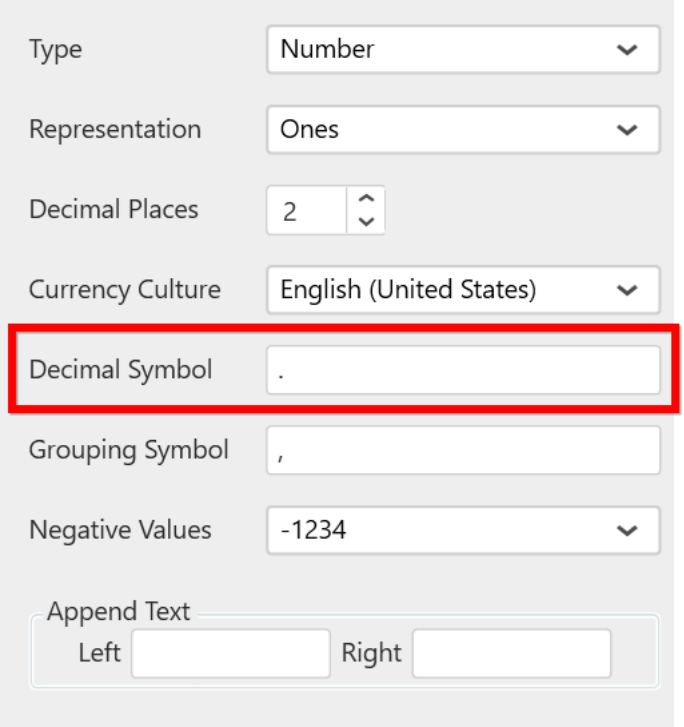
You can set the currency value culture when the value display type was set as Currency.



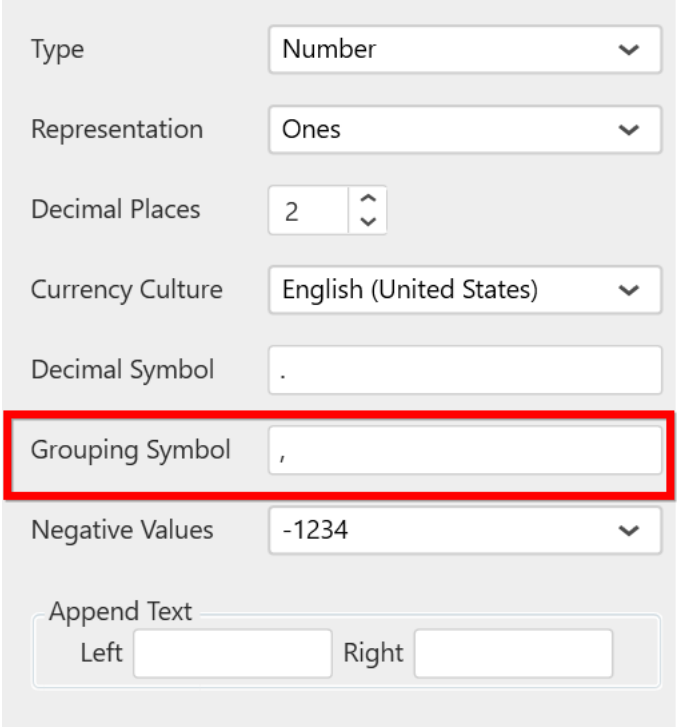
Decimal and Group separator

You can set the decimal and group separators for the value.

Decimal separator:



Group separator:

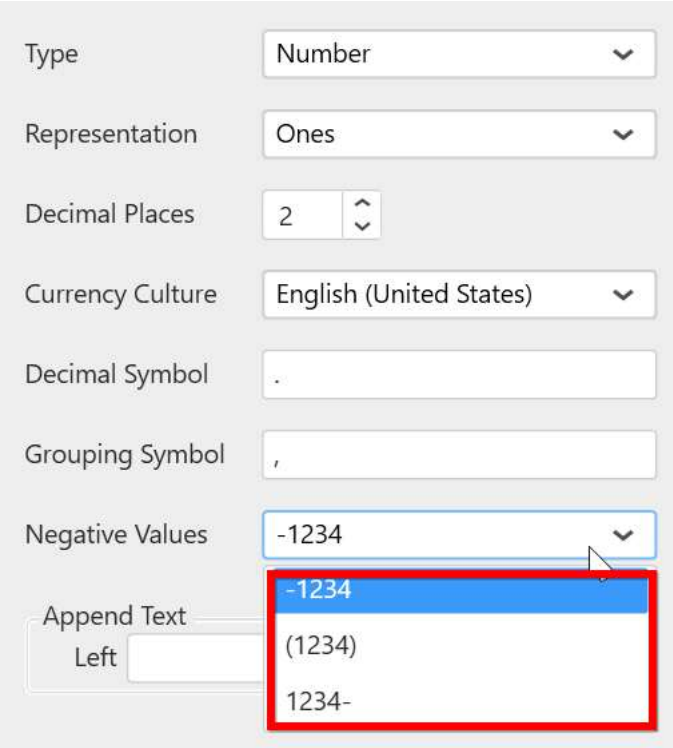


The decimal separator should be replaced with only symbols and special characters and the Group separator can be replaced with any value.

Note: You shouldn't use the same values in decimal and group separators.

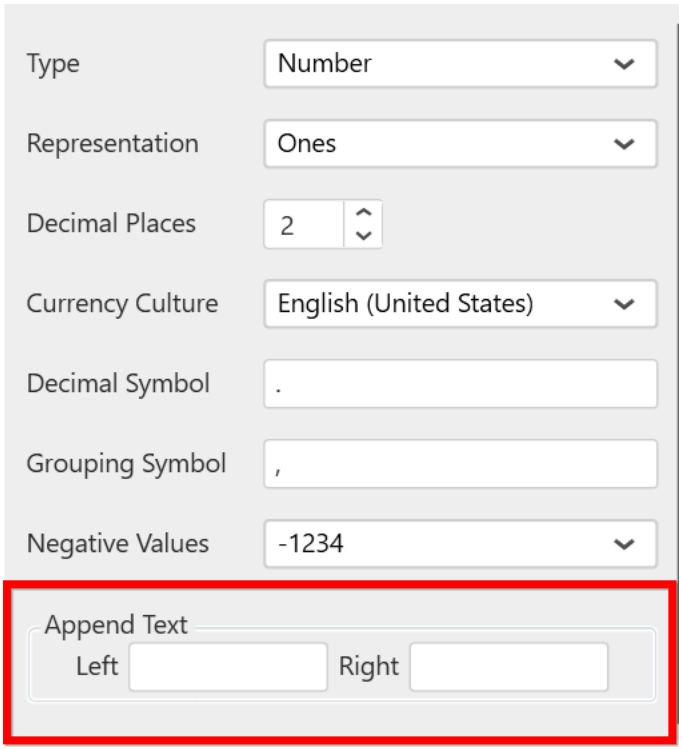
Negative Values

You can set the negative value display format for number representation.



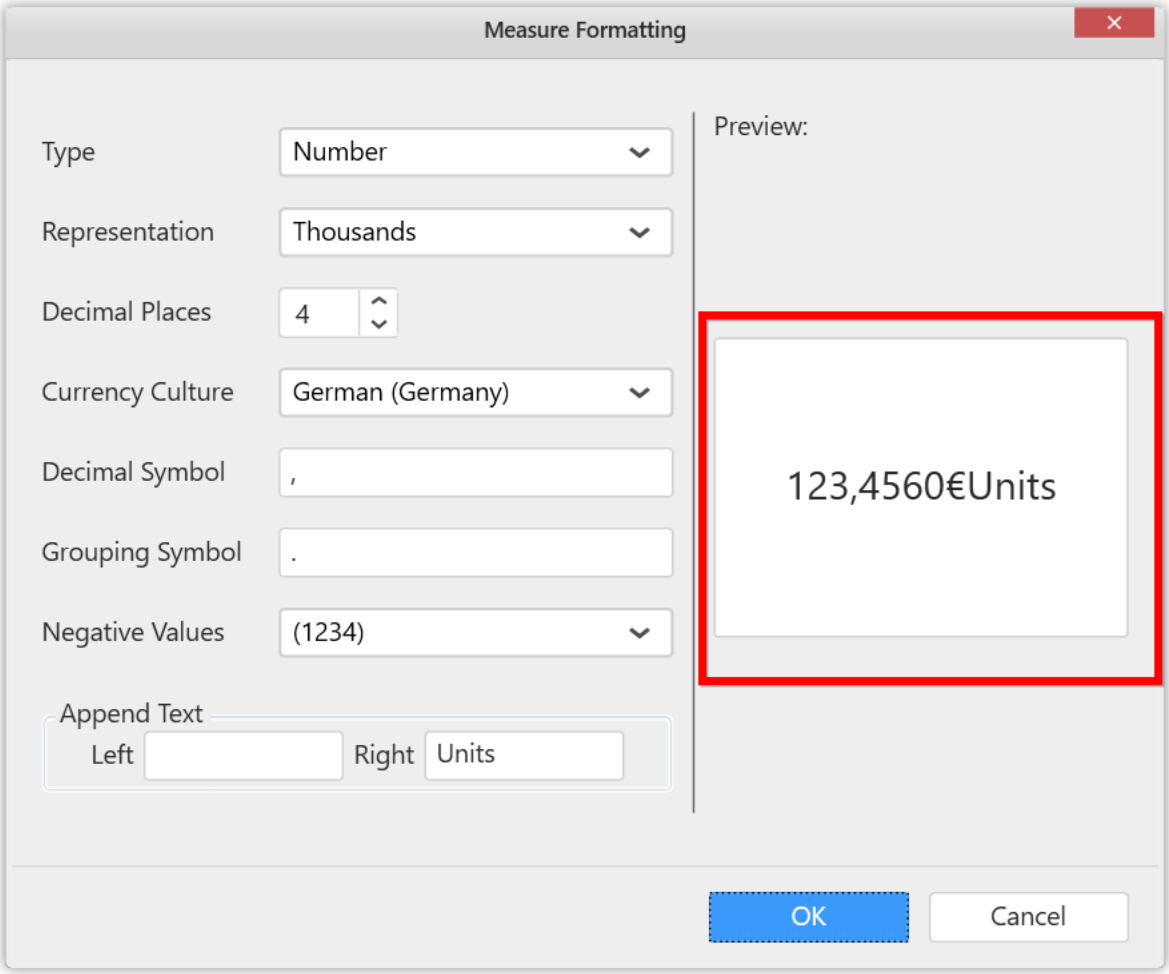
Append Text

You can append text, character, number or symbol either at start or at the end of the values.



Preview

This pane at right of Measure Formatting dialog will preview the display value based on the settings you change.

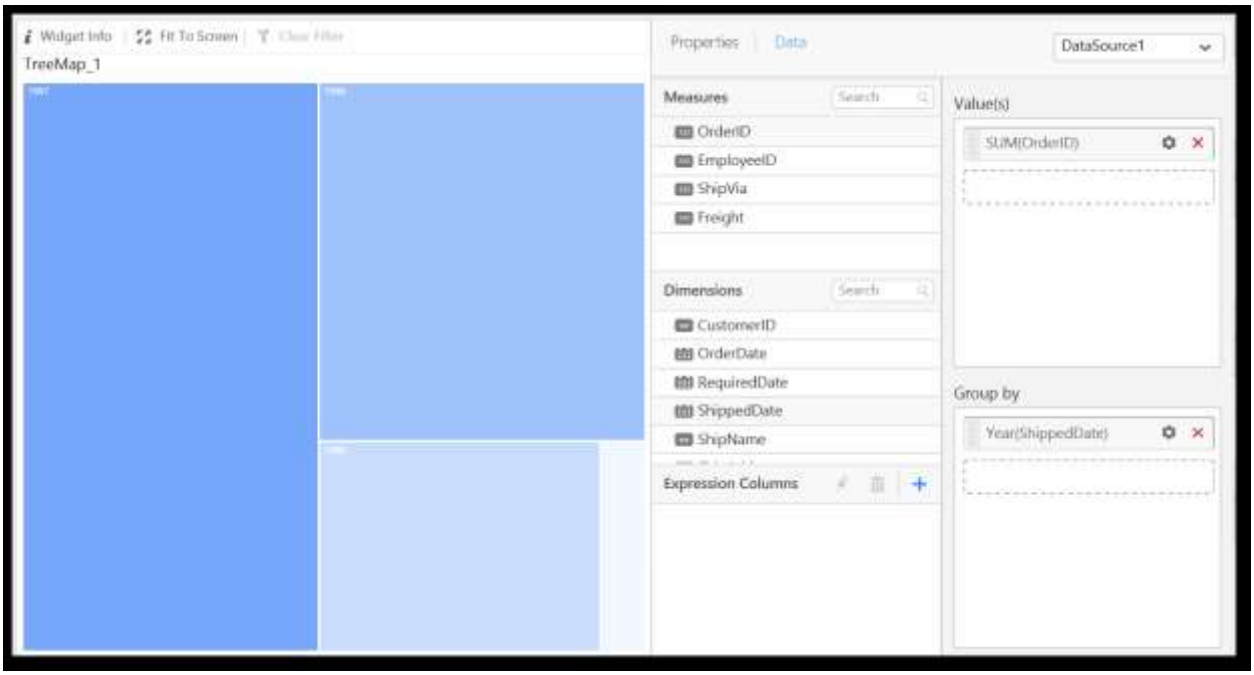


Custom Date/Time/Datetime Formatting

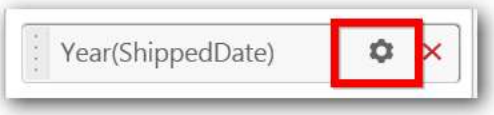
You can customize the display format of dates in dashboard widgets based on your need. You can choose from the wide range of date formats available including, custom date format.

How to format the Date field value shown in the Widgets

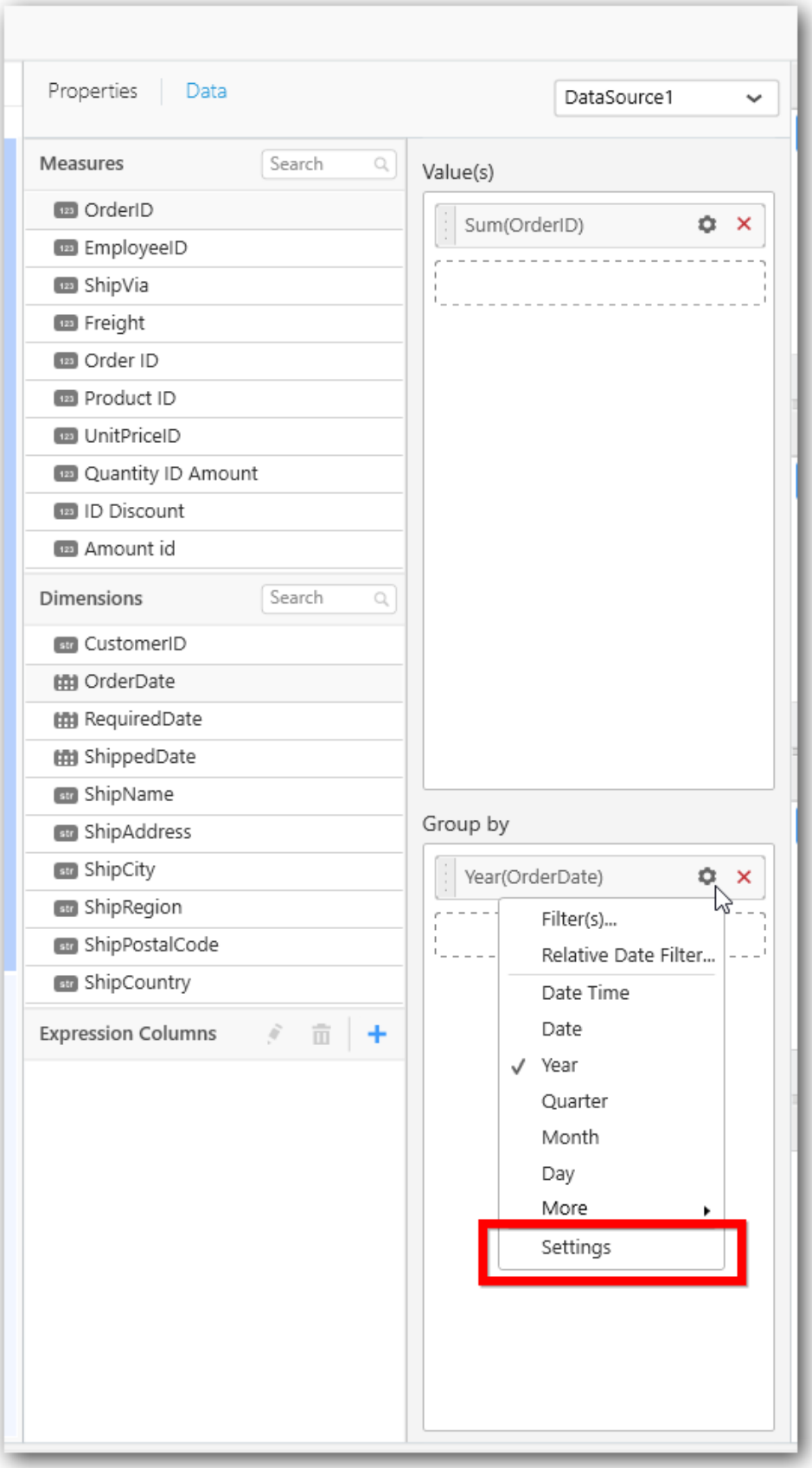
Bind the date field to the widget as shown in the image below.



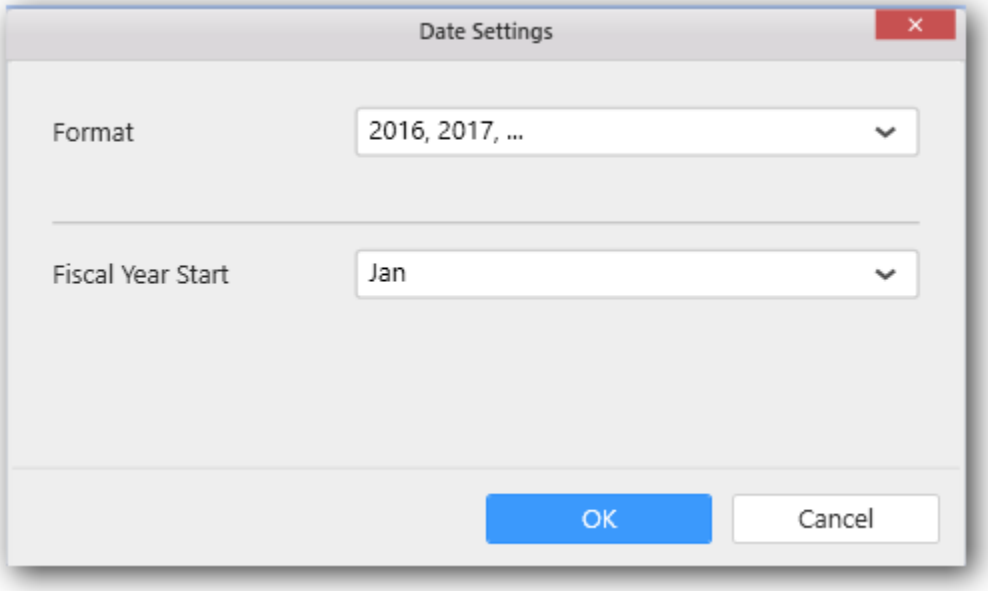
Click on the settings button in the added field.



Format option is provided in the context menu. Click on the Settings menu item to open the window.



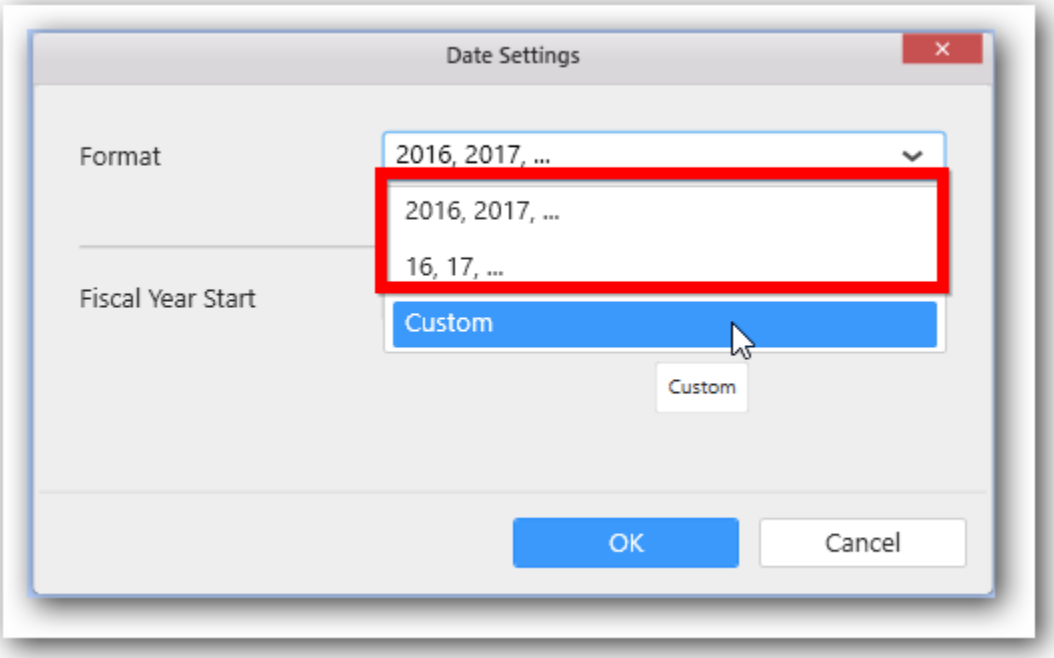
Date Settings window will be opened as shown here:



Format shows the predefined available options provided for the selected date/time format.

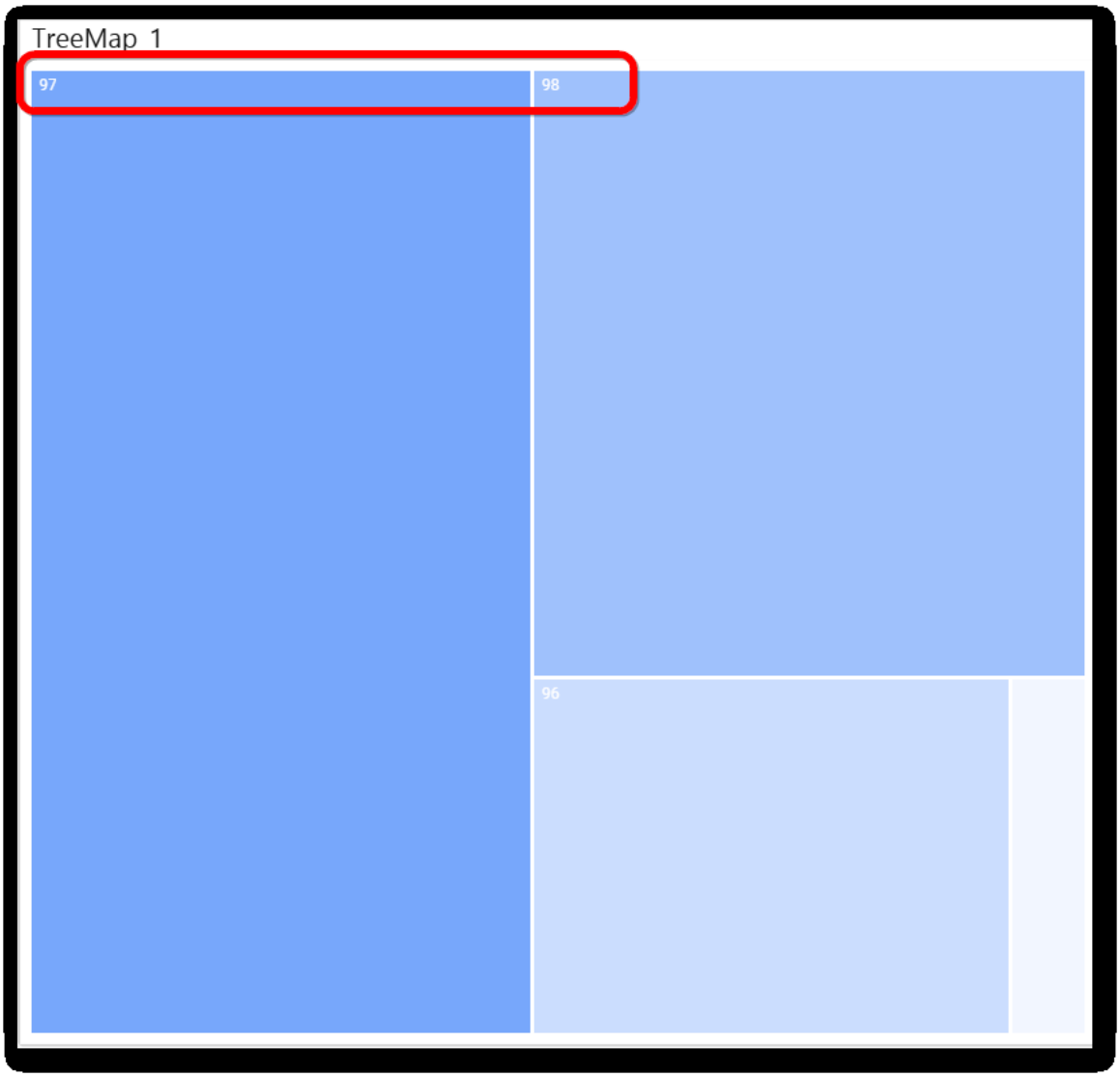
For example,

If Year is selected, then the following options will be shown.



Select the formatting option you want; it will be reflected in the widget.

For example, if we select 16,17,... option then the widget will be rendered like shown in the below image:



You can find the predefined formats in the below table.

Type	Formats available
Year	2016,2017,... 16,17,...
Quarter	Quarter 1,Quarter 2 Q1,Q2,Q3,Q4 1,2,3,4
Month	Jan,Feb,Mar January,February Ja,F,Mar,A 1,2,3,...12

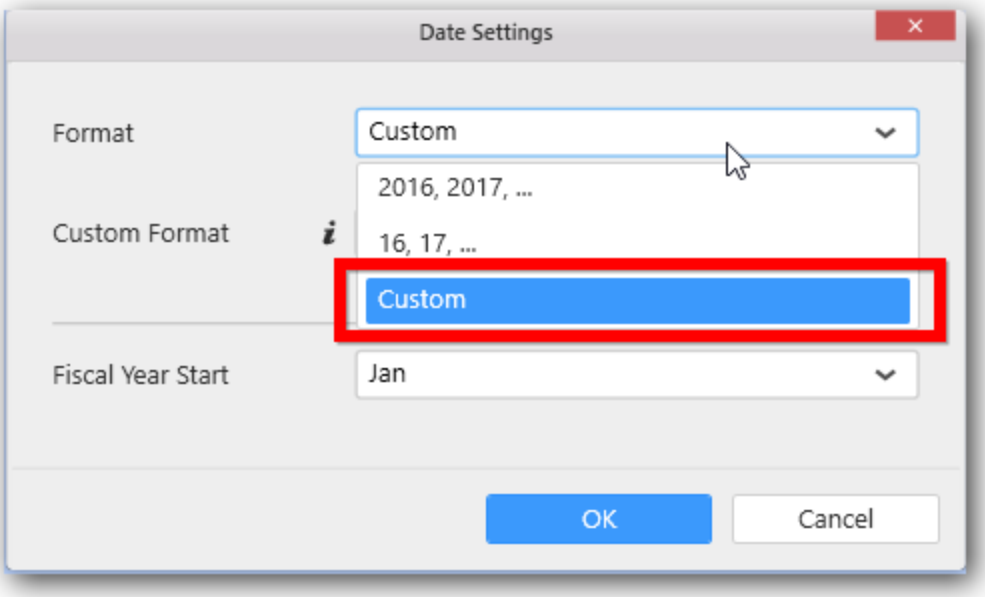
	01,02,03,...12
Hour	00,01,02,03,.. 1 AM,2 PM Ja,F,Mar,A 0,1,2,3,...
Minutes	00,01,02,03,.. 0,1,2,3,...
Seconds	00,01,02,03,.. 0,1,2,3,...
Quarter Year	Quarter 1 2017
Month Year	Jan 2016,May 2017
Date Hour	23/07/2017 00
Week of Year	00,01,02,03.. Week 0,Week 1,Week 2,.. Week 00,Week 01,Week 02,..
Day	00,01,02,03,..31 1,2,3,..31

Note: Date time formatting option is not available for the Time,Week Year,Day Month Year, Month Day Year types.

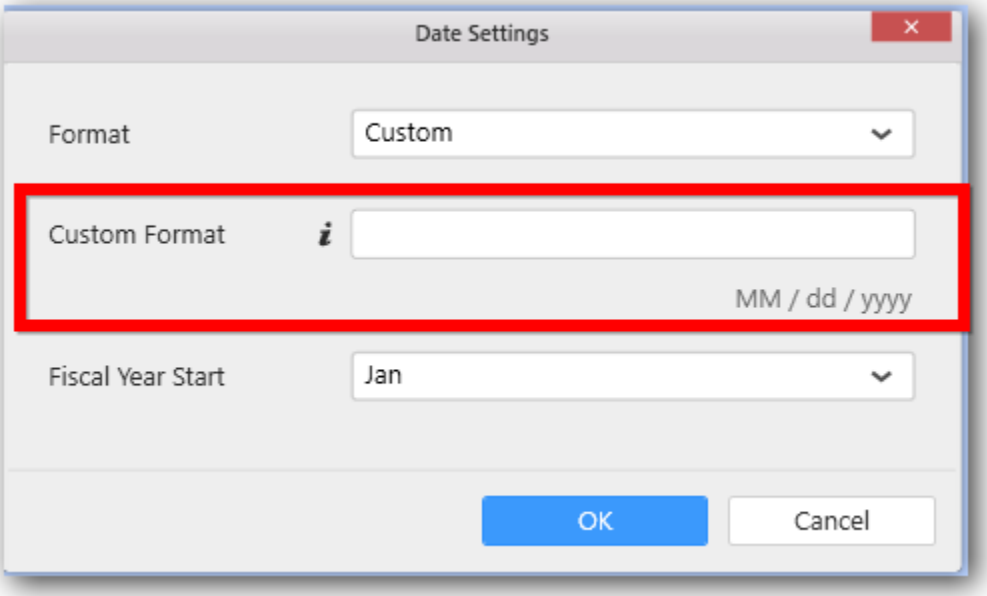
Also, the date/time formatting option is not available for the Chart and Range navigator widget types.

Custom formatting the date time format

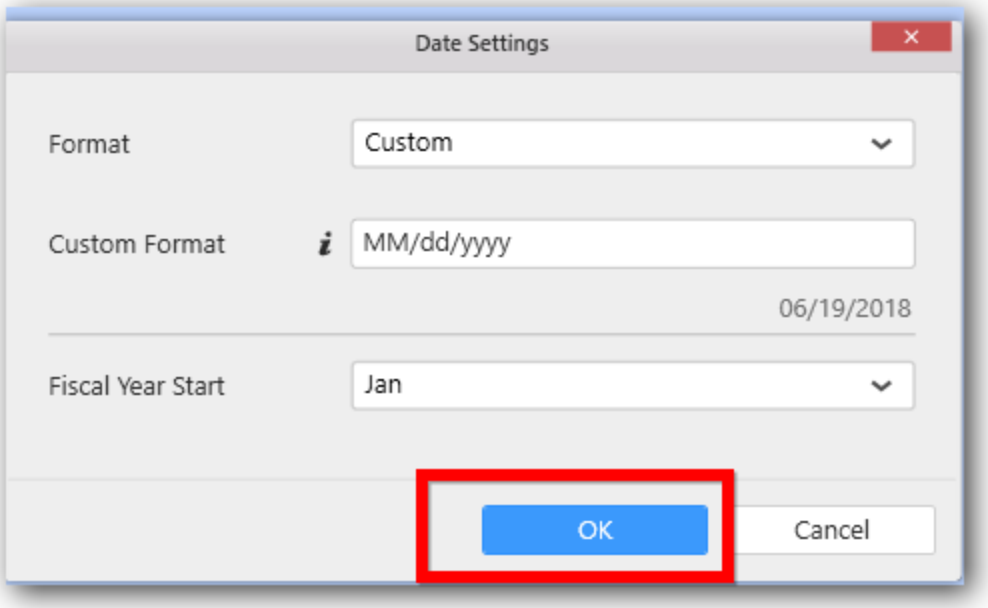
Select the custom option in the format window.



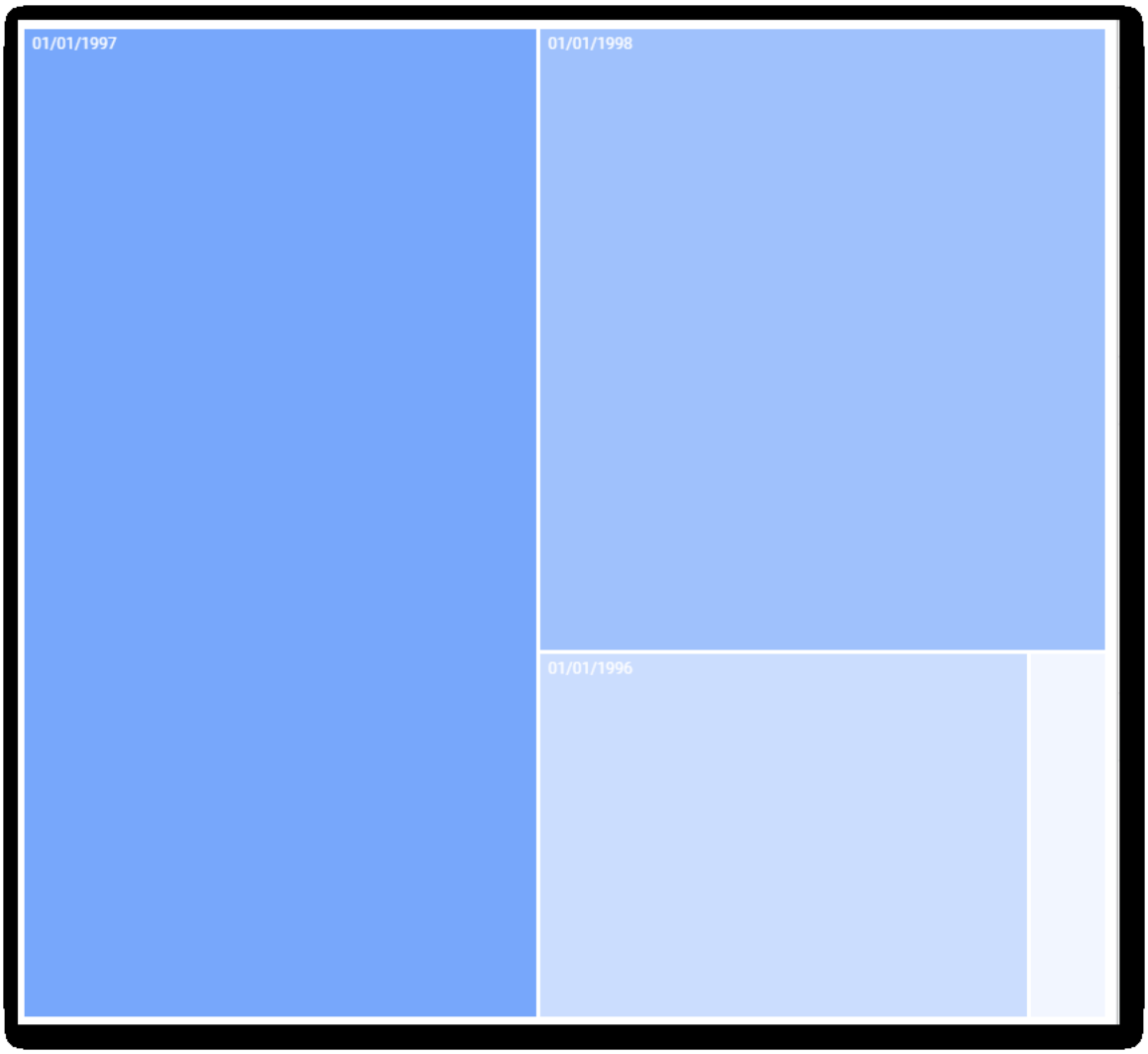
Now the Custom Format text box will be enabled.



Specify the custom format in the textbox and hit OK button.

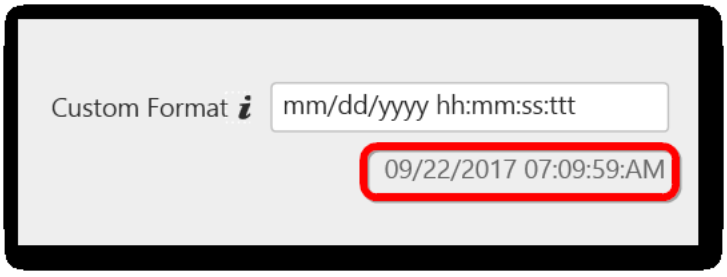


Widget will be re-rendered like shown here:



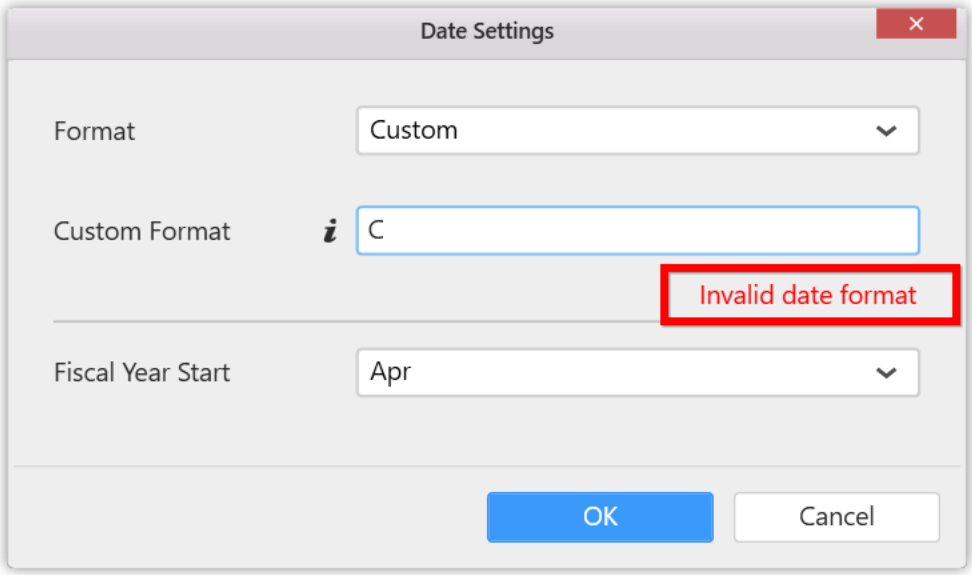
Proper custom date time format:

An example date time value will appear in the preview area as per the date time format specified in the textbox.



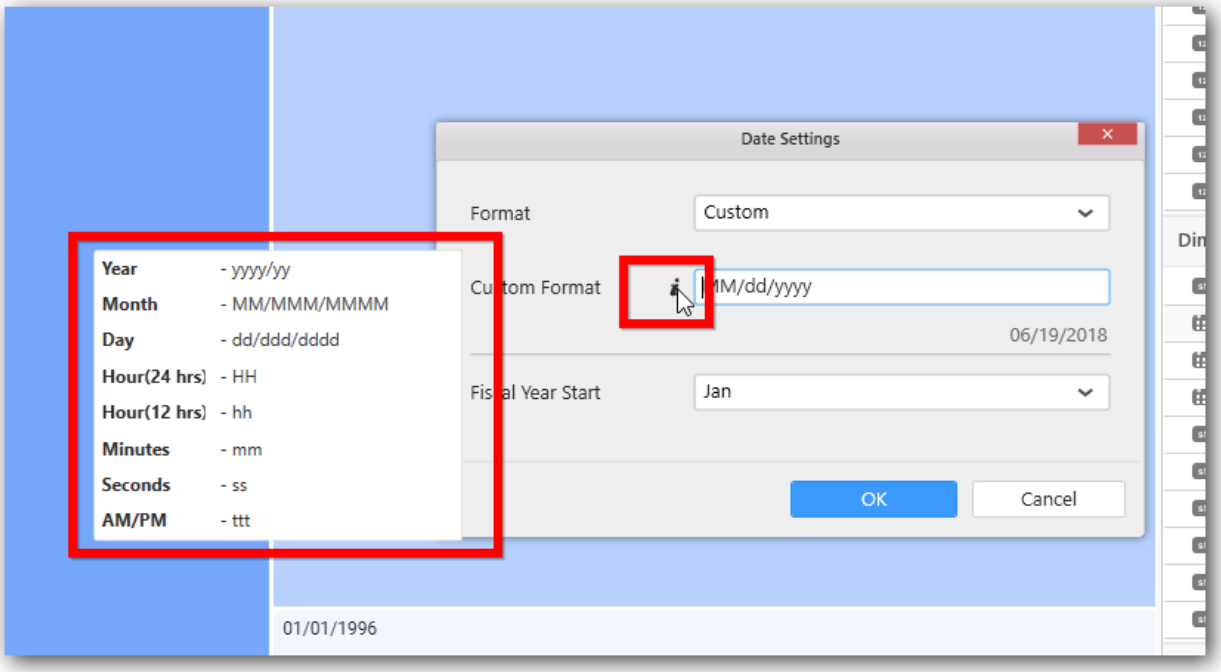
Invalid format:

For invalid date formats, it will show the below alert message:



The list of valid format specifiers can be referred from table mentioned in this page

You can find the formatting strings by hovering the i icon in the format window.



The valid format strings are listed in the below table.

Format String	description
yy	Displays the year, from 00 to 99.
yyyy	Display the year as a four-digit number.
MM	Display the month, from 01 through 12.

MMM	Display the abbreviated name of the month
MMMM	Display the full name of the month.
HH	Display the hour, using a 24-hour clock from 00 to 23.
hh	Display the hour, using a 12-hour clock from 01 to 12.
mm	Display the minute, from 00 through 59.
ss	Display the second, from 00 through 59.
ttt	Display the AM/PM.

Note: Custom formatting option is only available for the below Date Time formats:

* Date time

* Date

* Year

* Month Year

* Date Hour

Fiscal Year Start

Fiscal year start allows users to set fiscal format to the date field. It enables them to design dashboard based on fiscal dates. In some situations, a date field needs to be expressed in terms of an organization's fiscal year.

Way to apply the fiscal format

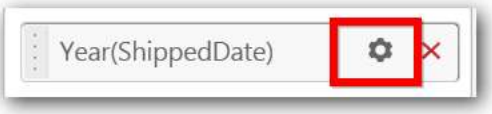
This setting can be applied either in data source level or in widget level. For data source level you can refer here. For widget level, please refer the below steps.

Fiscal year start in widget level

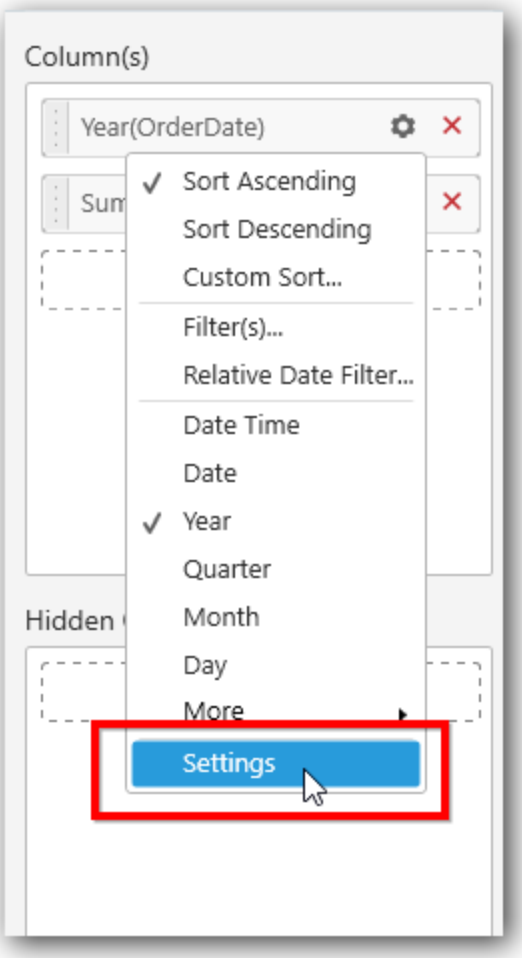
You can apply the fiscal format on date fields in dashboard widgets based on your need.

How to apply Fiscal Year Start in widget level

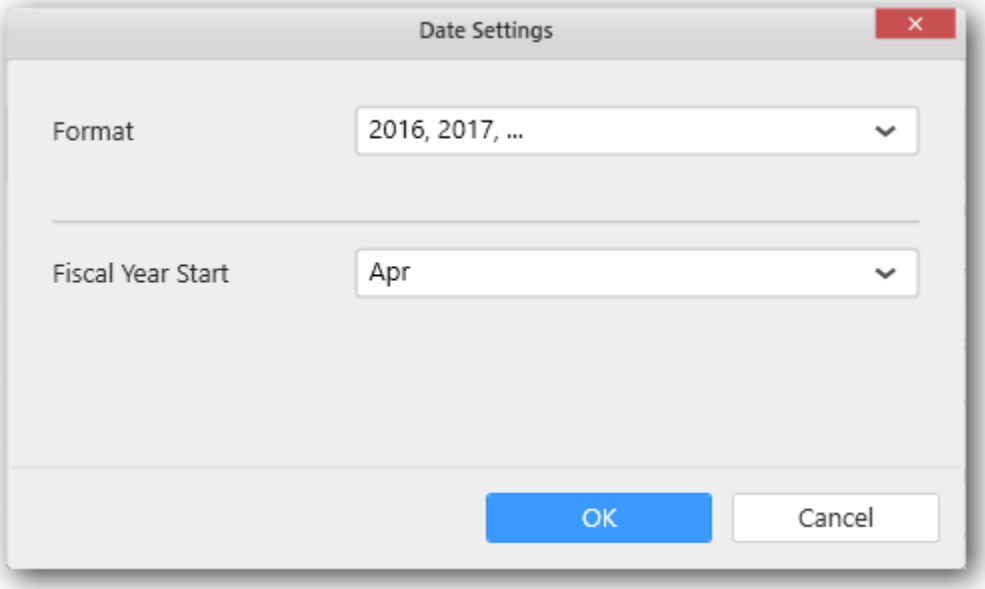
Click on the settings button in the added field.



Settings option is provided in the context menu. Click on the Settings menu item to open the window.

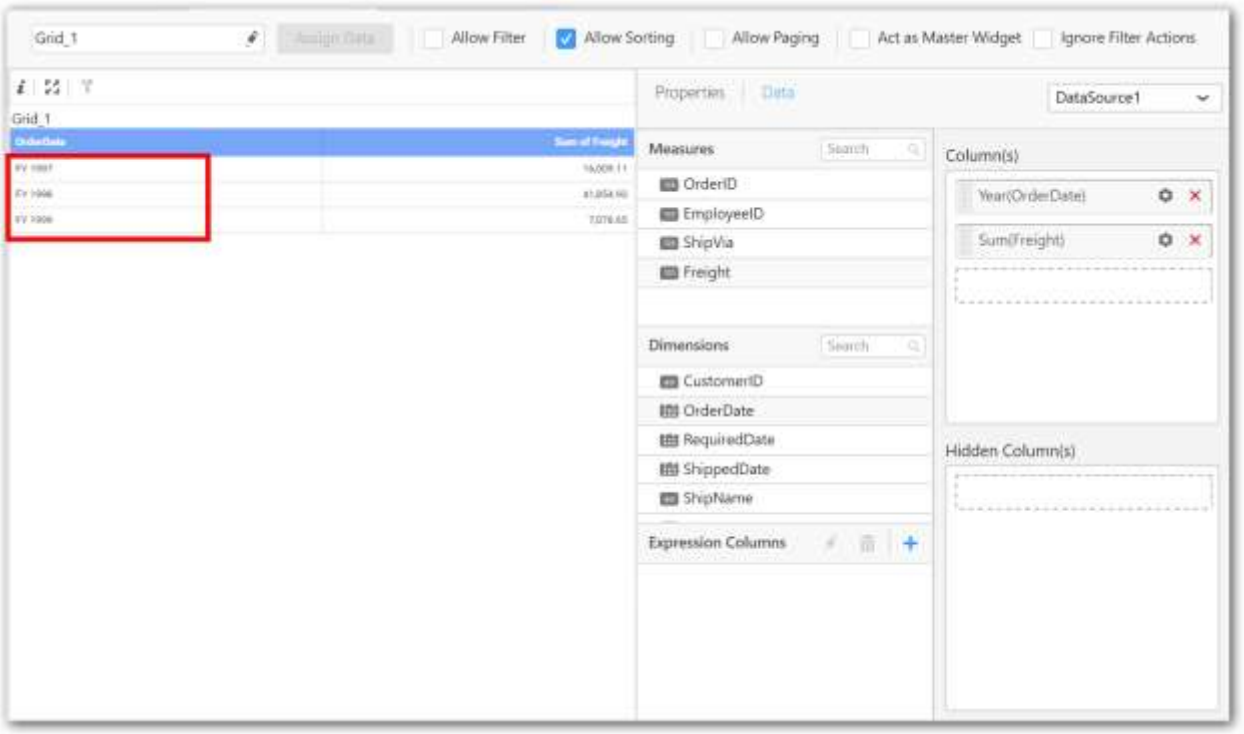


The following Date Settings window will be opened..



Now, you can choose the starting month of the fiscal year from the Fiscal Year Start drop-down.

The Widget will render as follows.

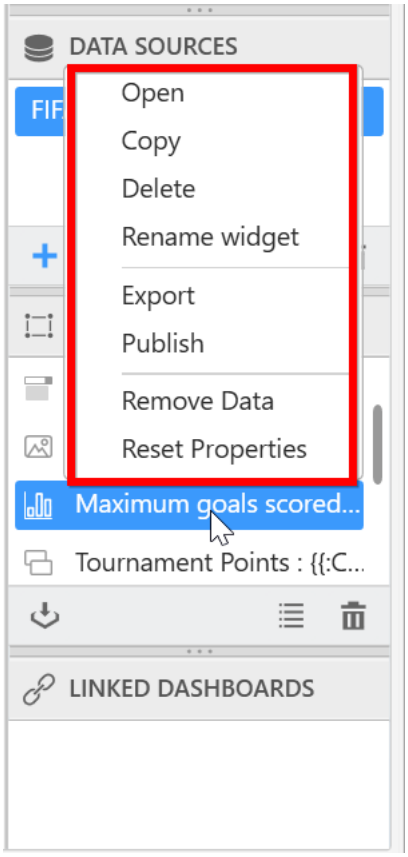


Saving a dashboard widget

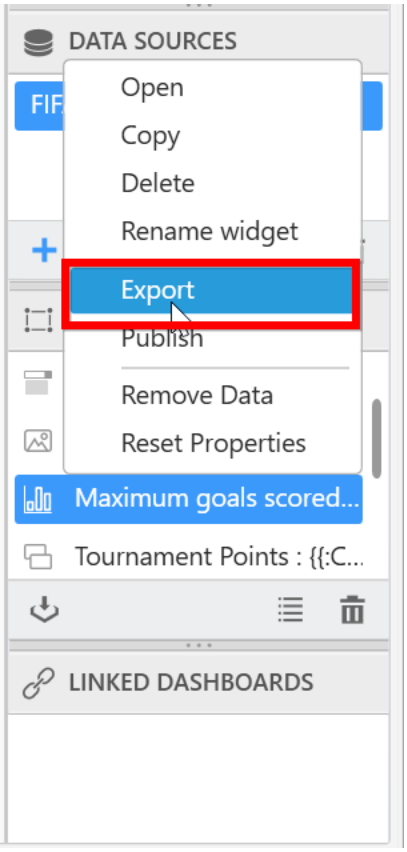
You can save any dashboard widget used in the dashboard report into your local by following the below steps.

1. Select the widget you want to export in the widget's container.

2. Right click on the widget name to open the context menu.



3. Select the Export option in the menu and save in your required location.

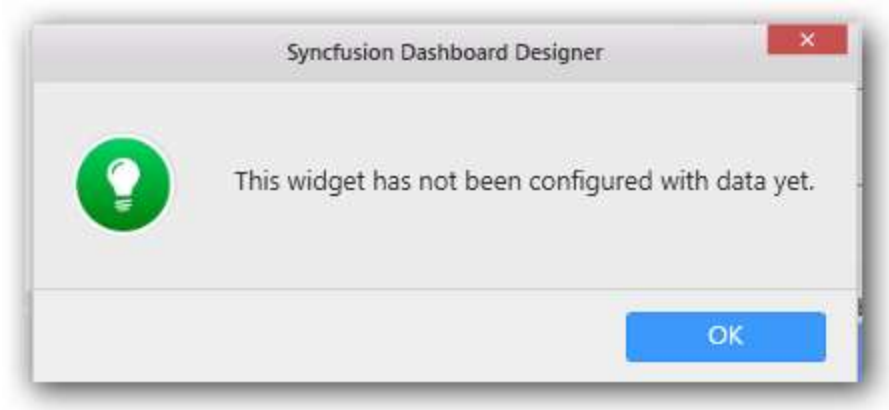


Note: The widget and its corresponding data sources will be exported in the Syncfusion widget(.sydw) file format. You can import the sydw files in Dashboard Designer application any time or you can also add the file to your Dashboard Server application.

Viewing widget bounded data

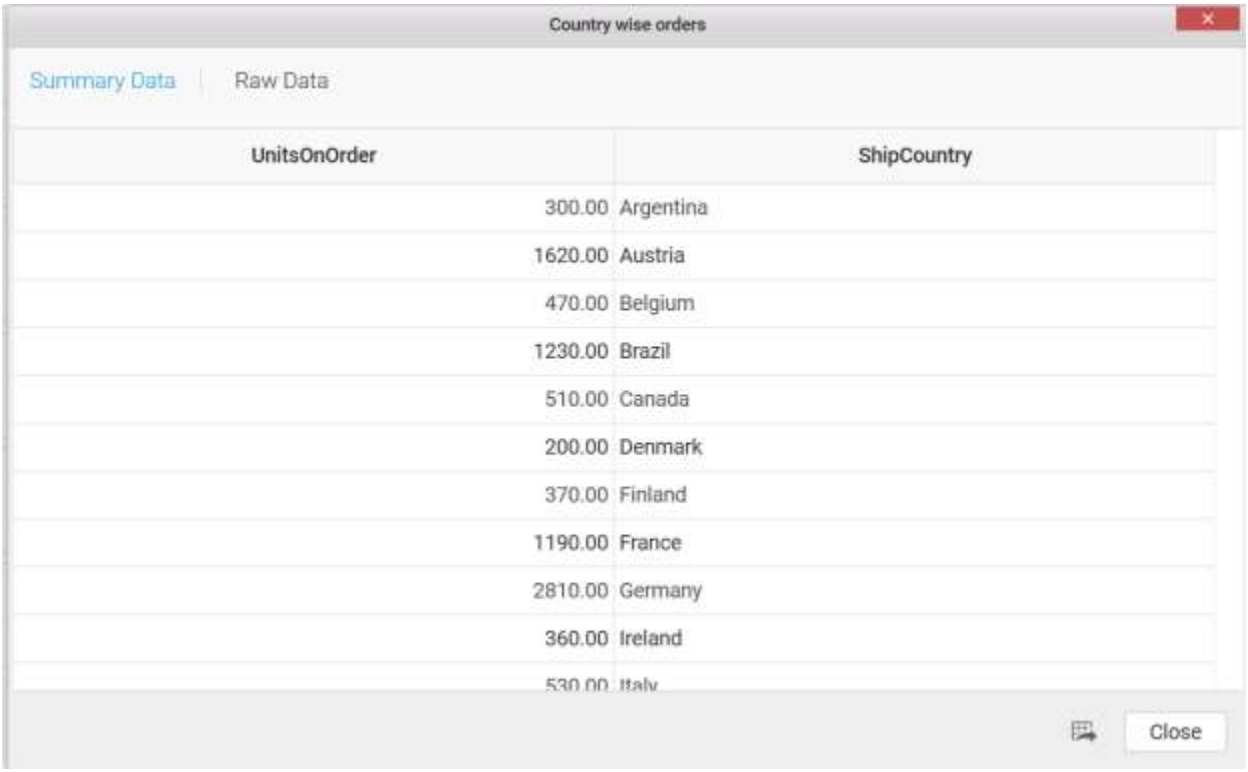
You can view the widget bounded data by clicking the view data icon in WIDGETS pane and you will get the window which contains the Summary Data and Raw Data.

If the selected widget was not bounded with data, it will show the alert message.

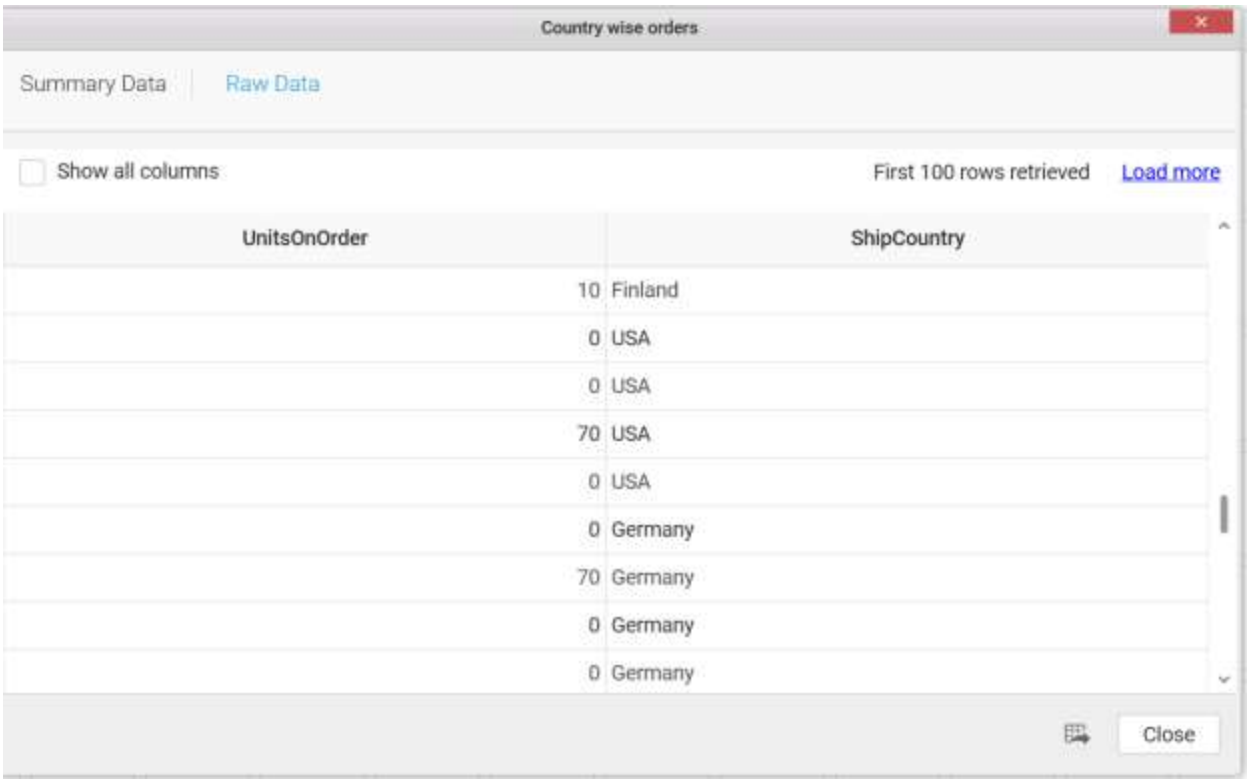


Note: Any widget bounded with SSAS data source don't have the Raw Data tab displayed in View Data window as the cube itself holds the aggregated data.

Summary Data



Raw Data

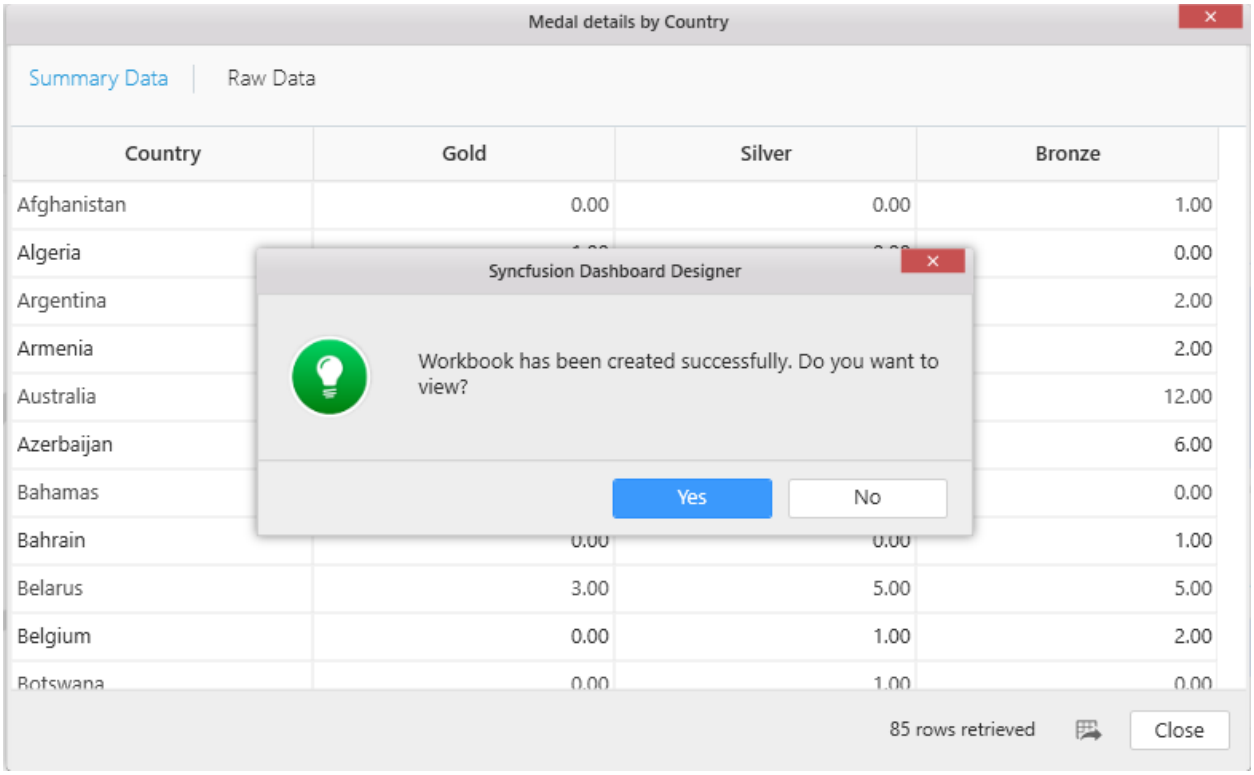


By following below procedure, you can export the data to excel file which is displayed in the view data grid.

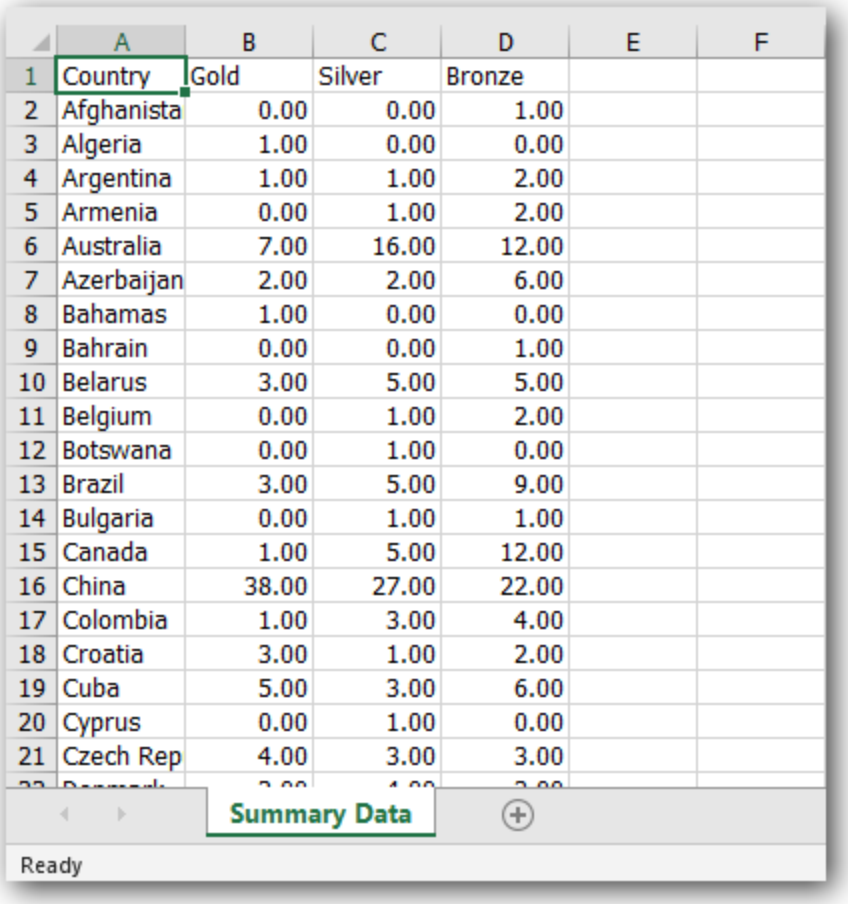
Click Export icon in view data grid.

Now Save as window will open. Enter a suitable name and choose the specified location. By clicking the Save button, you can save the data in excel format.

Once data saved successfully, you will get message as follows.



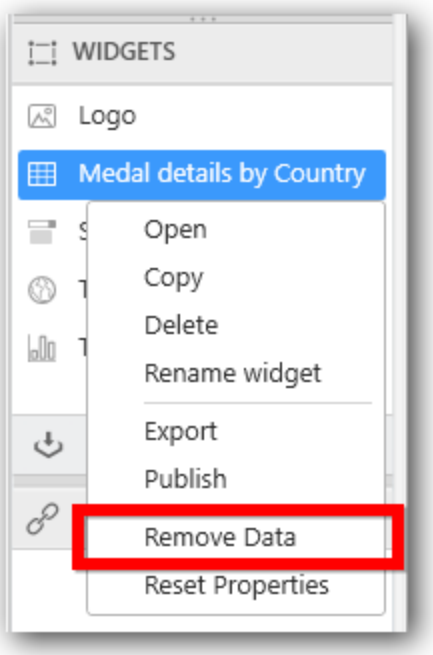
Click Yes to view the saved data in excel file as follows.



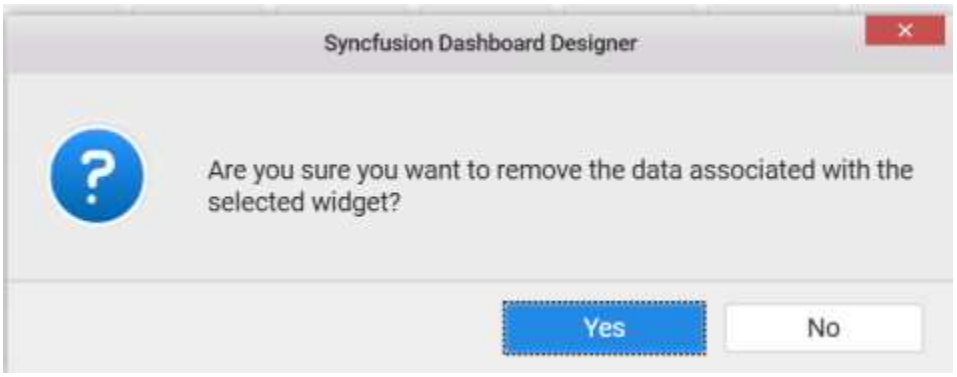
	A	B	C	D	E	F
1	Country	Gold	Silver	Bronze		
2	Afghanista	0.00	0.00	1.00		
3	Algeria	1.00	0.00	0.00		
4	Argentina	1.00	1.00	2.00		
5	Armenia	0.00	1.00	2.00		
6	Australia	7.00	16.00	12.00		
7	Azerbaijan	2.00	2.00	6.00		
8	Bahamas	1.00	0.00	0.00		
9	Bahrain	0.00	0.00	1.00		
10	Belarus	3.00	5.00	5.00		
11	Belgium	0.00	1.00	2.00		
12	Botswana	0.00	1.00	0.00		
13	Brazil	3.00	5.00	9.00		
14	Bulgaria	0.00	1.00	1.00		
15	Canada	1.00	5.00	12.00		
16	China	38.00	27.00	22.00		
17	Colombia	1.00	3.00	4.00		
18	Croatia	3.00	1.00	2.00		
19	Cuba	5.00	3.00	6.00		
20	Cyprus	0.00	1.00	0.00		
21	Czech Rep	4.00	3.00	3.00		

Removing bounded data

You can remove the data by right click on the widget name in WIDGETS pane and select the Remove Data option.

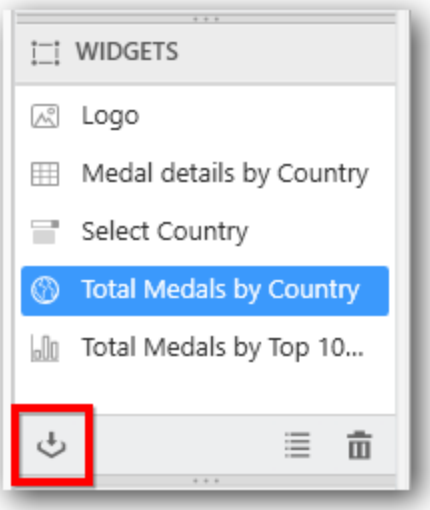


You will get the alert message to remove the bounded data and click Yes button.

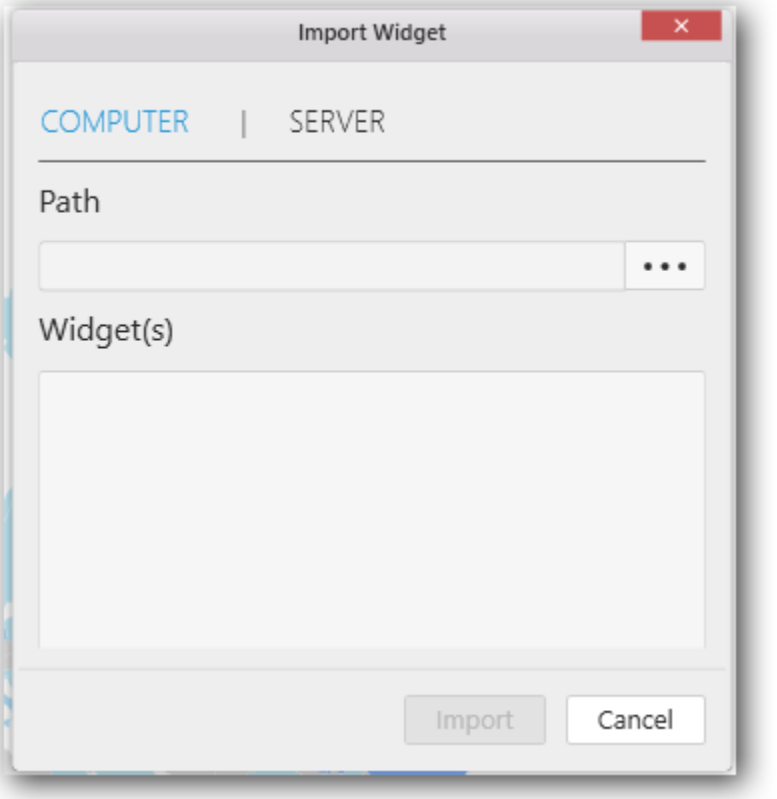


Using an existing dashboard widget

You can use an existing dashboard widget by clicking the Import widget icon in WIDGETS Pane and you will get the Import Widget window.

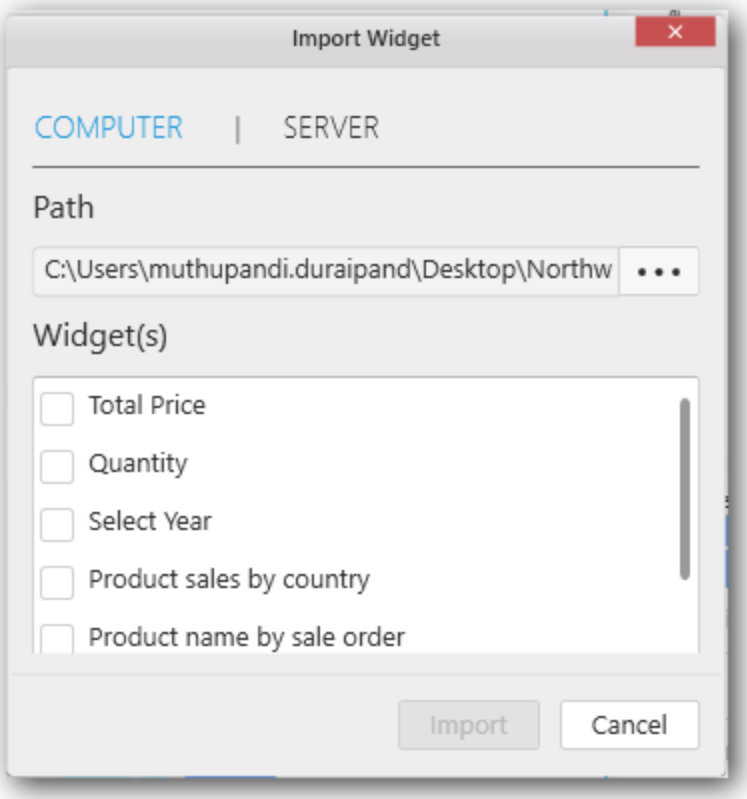


The Import widget window opens as follows.

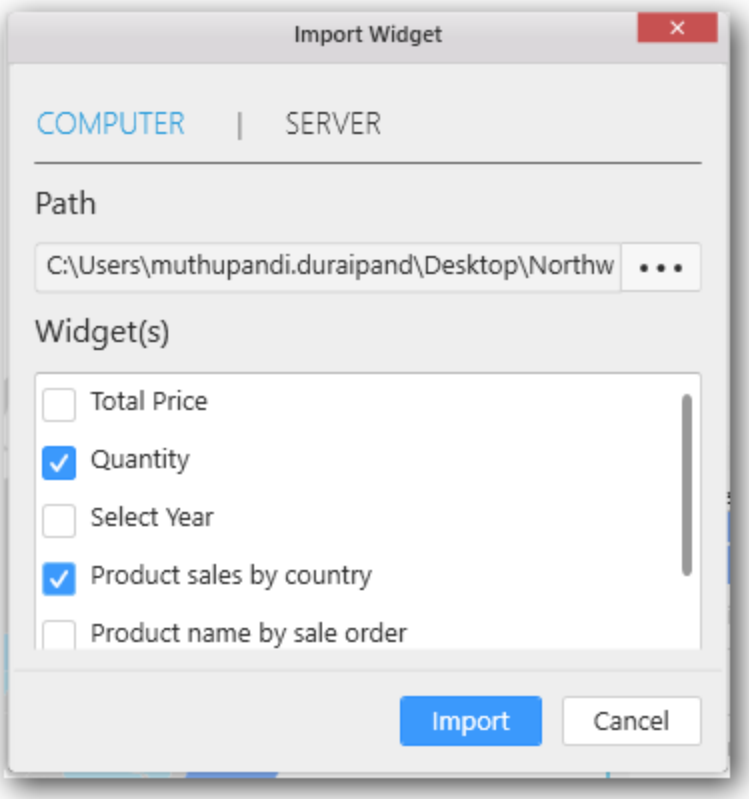


COMPUTER

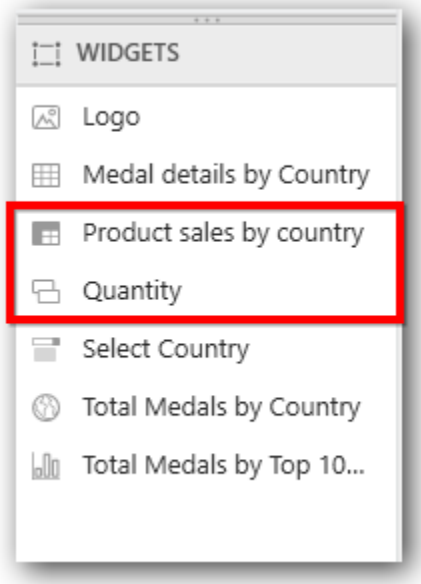
Using this option, you can browse the widget file from the local machine. Here, you can import widgets from SYDX/SYDW formatted files. When selecting a file, you can get a list of available widgets displayed as in the following.



Select the required widgets to import into your dashboard and click Import.



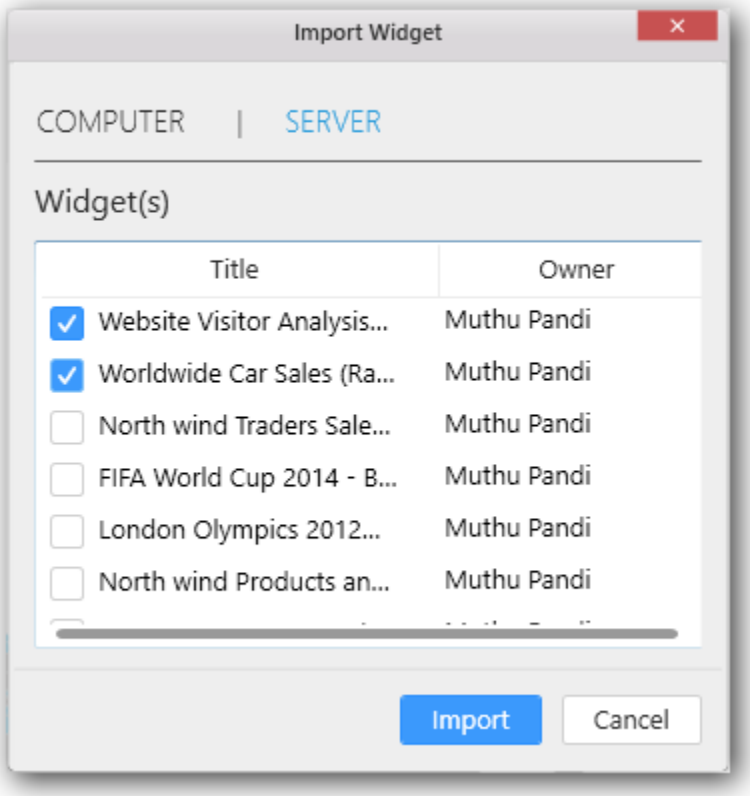
Now the respective widget(s) will be imported into the WIDGETS container as follows.



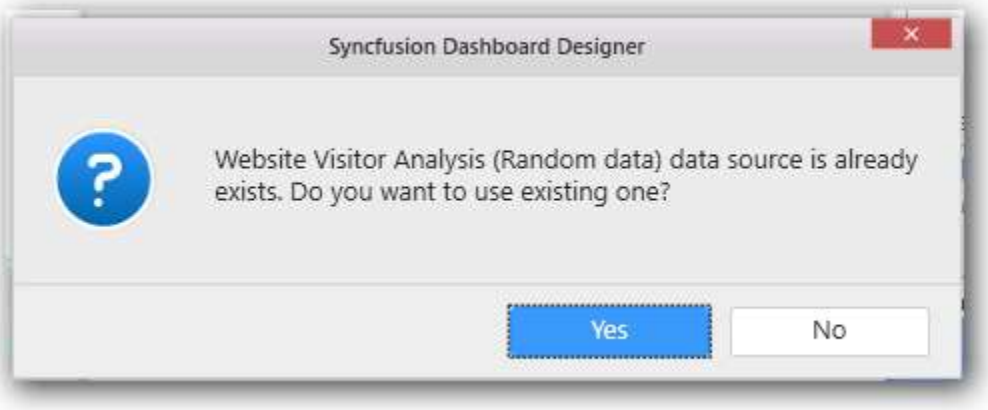
SERVER

You can import widgets from the Dashboard Server using the following ways. To display the widgets that are present in the server, you should sign in the Dashboard Server account in the Dashboard Designer.

Select the SERVER tab. You will get a list of available Widgets that are present in the Dashboard Server. Choose the widgets and click Import.

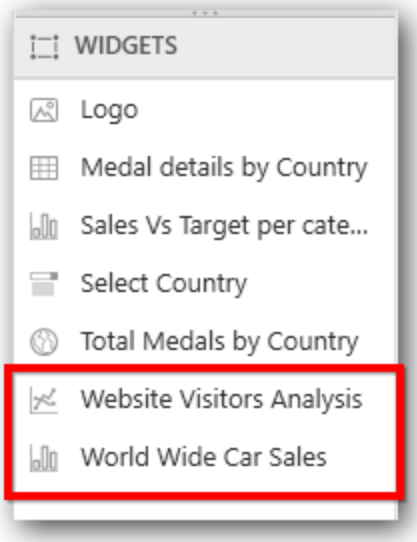


While importing widget, if bounded data source already existed in the data source container, it will show below information message.



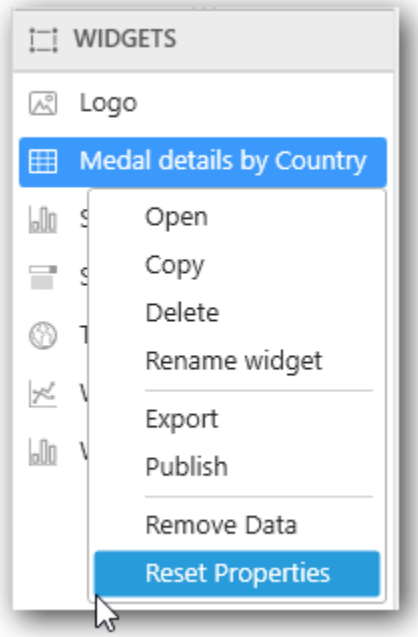
If click yes it will reuse the data source which exists in the data source container. otherwise, it will import another one data source.

Now the respective widget(s) will be imported into the WIDGETS container as in the following.

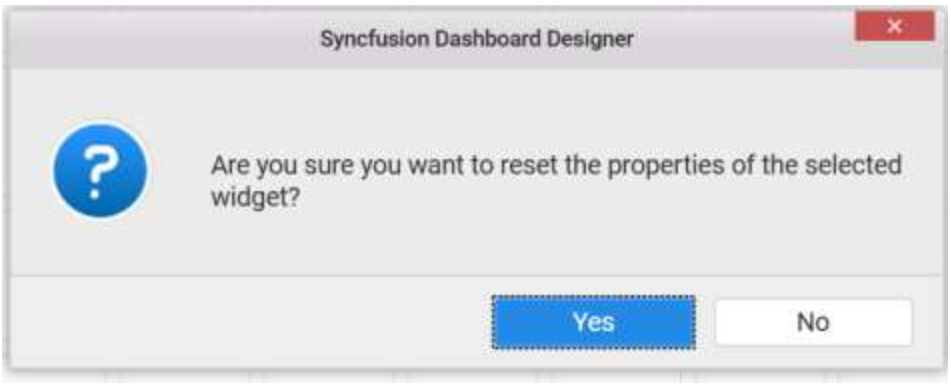


Resetting the widget properties

You can reset the widget properties by right click on the widget and select the Reset Properties.

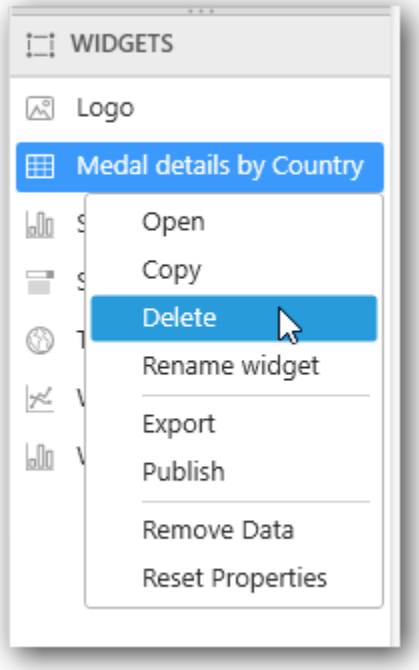


You will get the alert message to reset the properties of the selected widget and select Yes button to reset it.

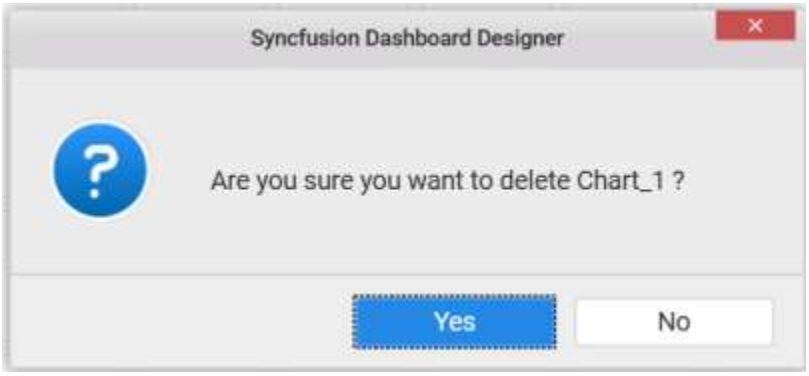


Deleting a dashboard widget

You can delete a dashboard widget by right click on the widget name in WIDGETS pane and select the Delete option.

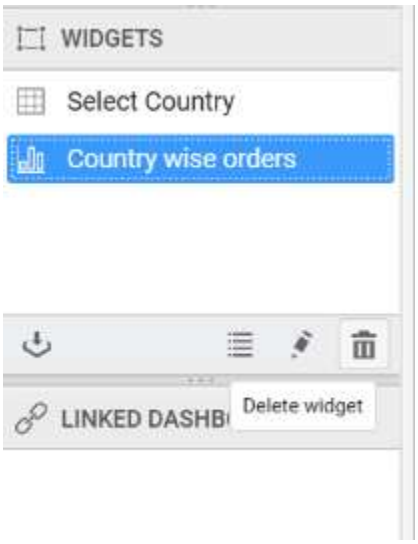


You will get the alert message to delete the widget and click Yes button.

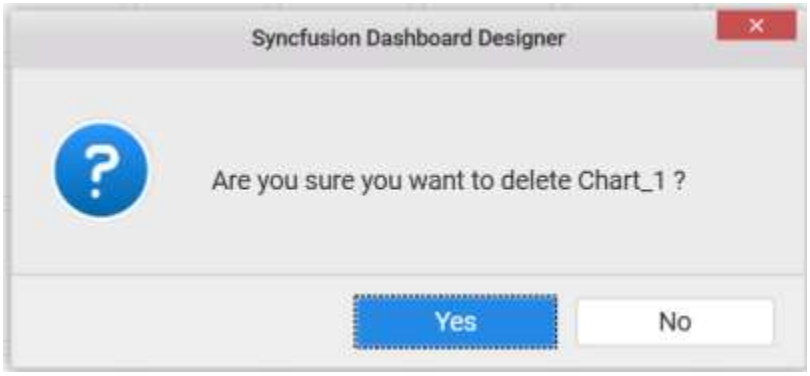


Click No if required, not to delete the widget.

You can also delete the widget by clicking the Delete Widget icon in WIDGETS pane.



You will get the alert message to delete the widget and click Yes button.



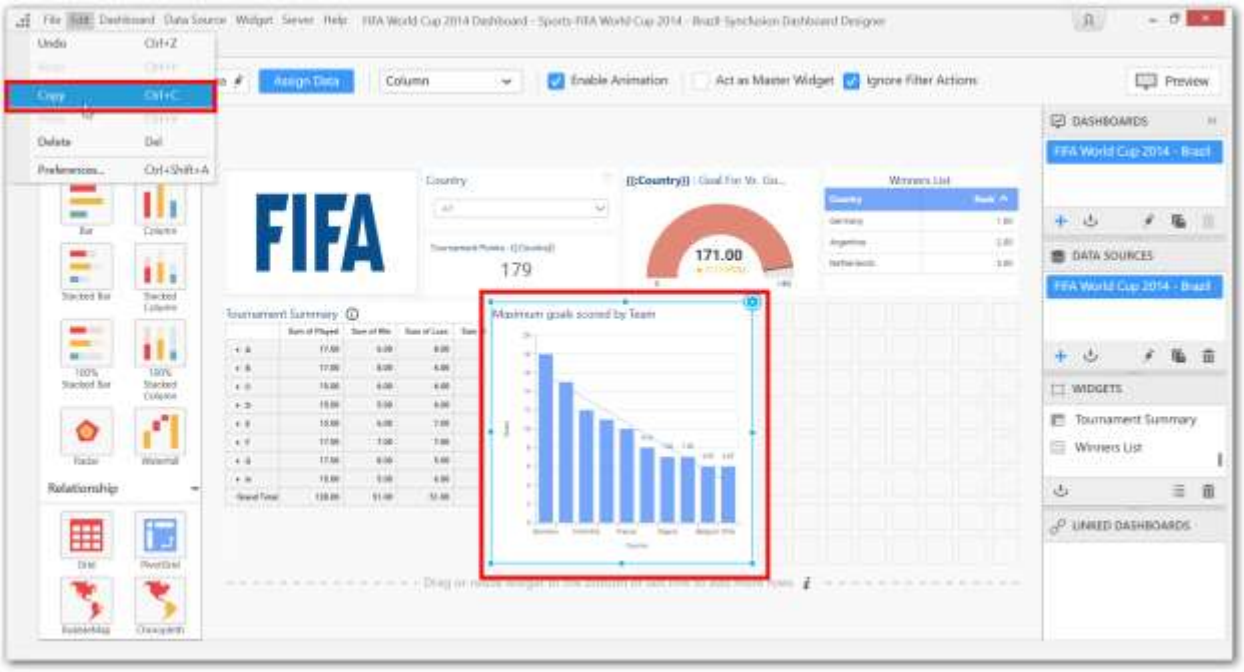
Duplicating a Dashboard Widget

Using menu options

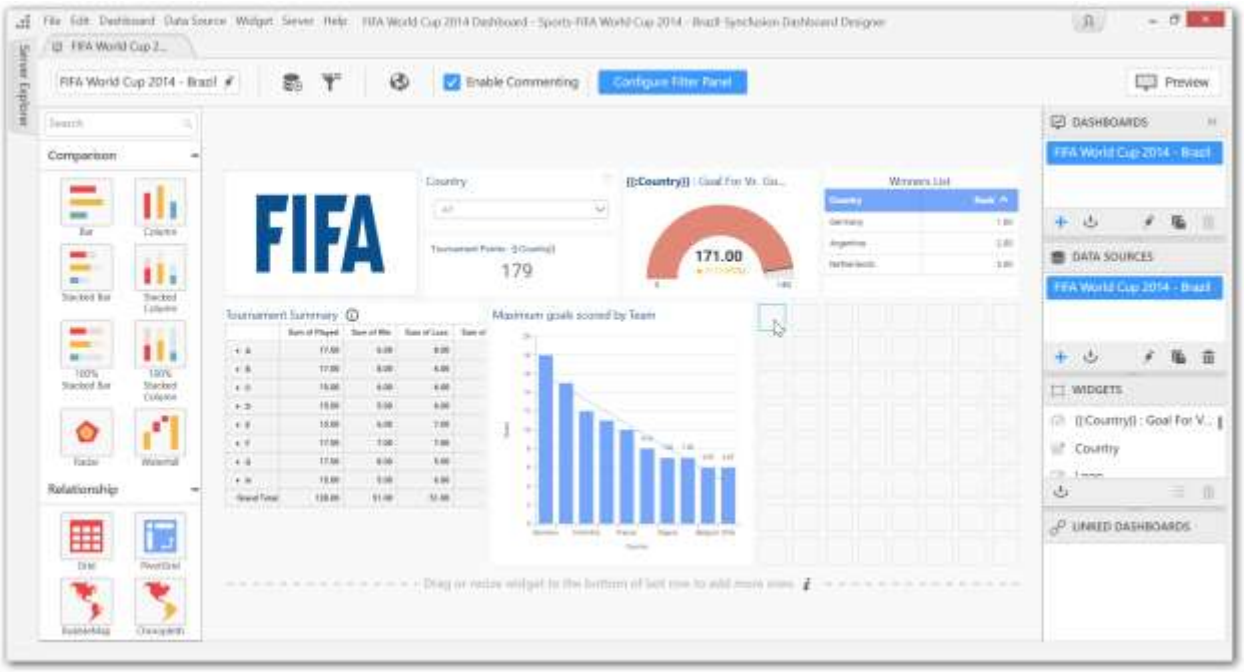
You can copy or duplicate a particular dashboard widget with in the same dashboard tab using Copy and Paste option provided in the Edit menu or using keyboard shortcuts.

1. Select the dashboard widget you want to copy in the Widgets container or in the Design canvas.

2. Open on the Edit menu option and click the Copy menu option.

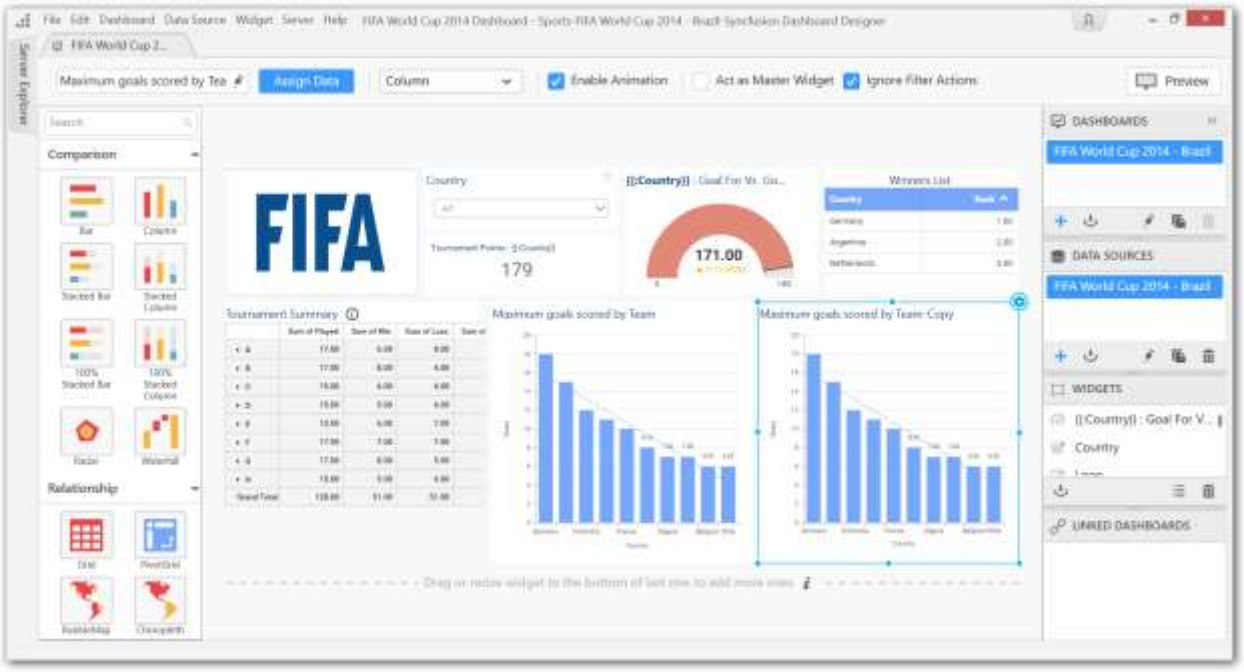


3. Now select an empty tile in the Design canvas where you want to paste the copied widget.



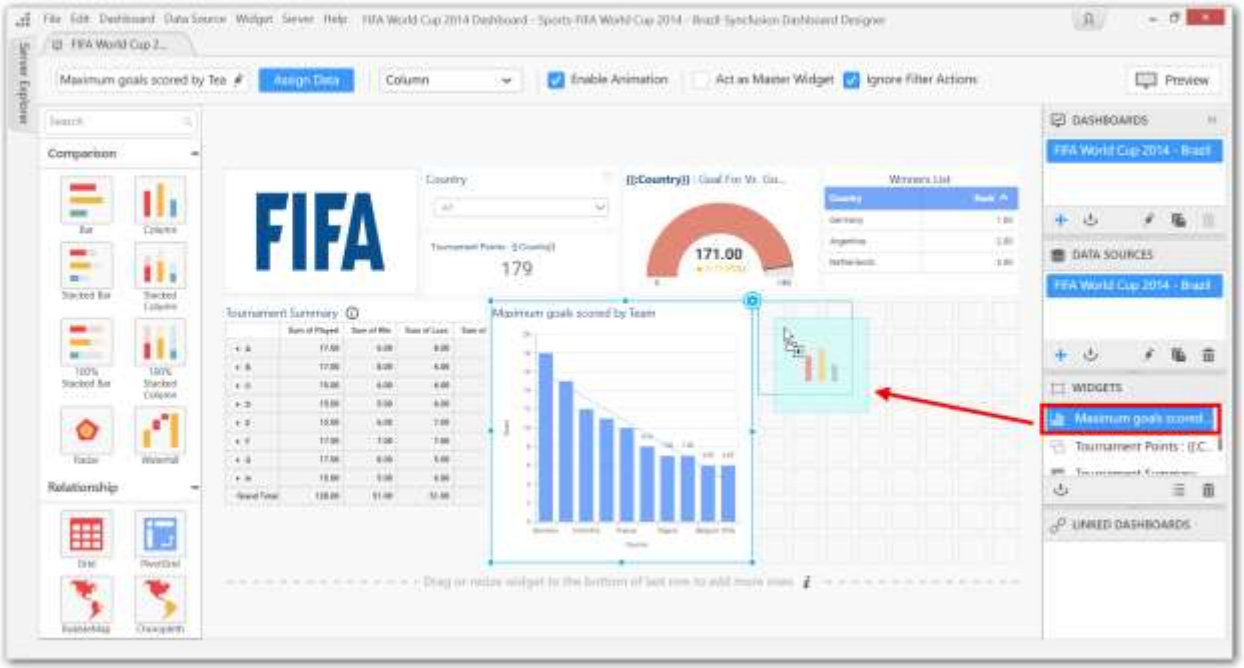
4. Use the keyboard shortcut key Ctrl + V for pasting the widget or make use of the Edit - Paste option from menu.

Now the widget will be copied as shown in the following screenshot:



Using drag and drop option

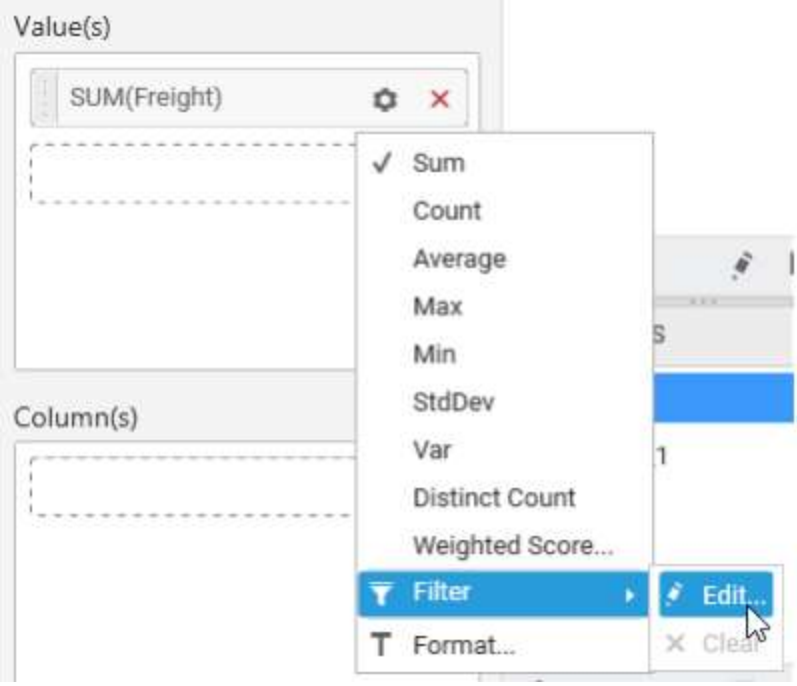
You can also perform the duplicate option by drag and drop the Selected widget from the Widgets container into the empty tiles in the Design canvas like shown below:



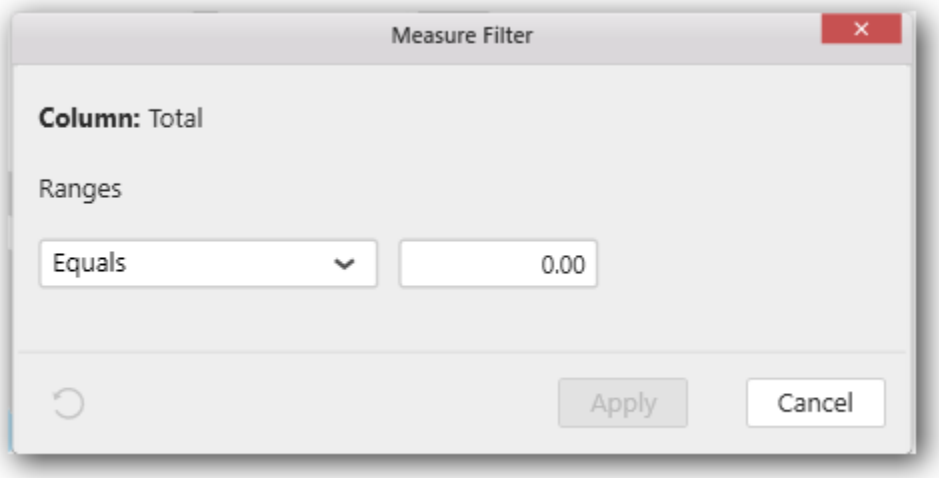
Configuring Widget Filters

Configuring Filter for Measure Column

Filter for measure column can be configured through opening the Measure Filter dialog from the Filter option in the Settings drop down menu.



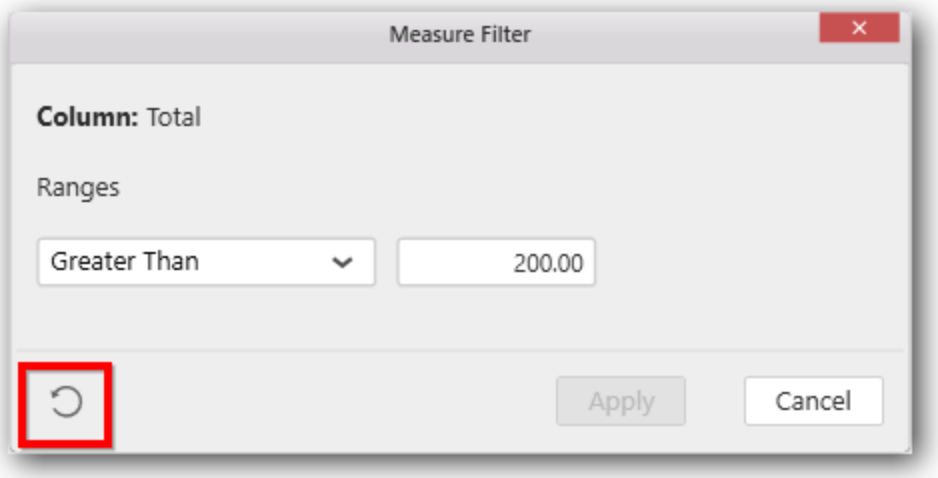
Measure Filter dialog will open like below on clicking the Edit… menu item.



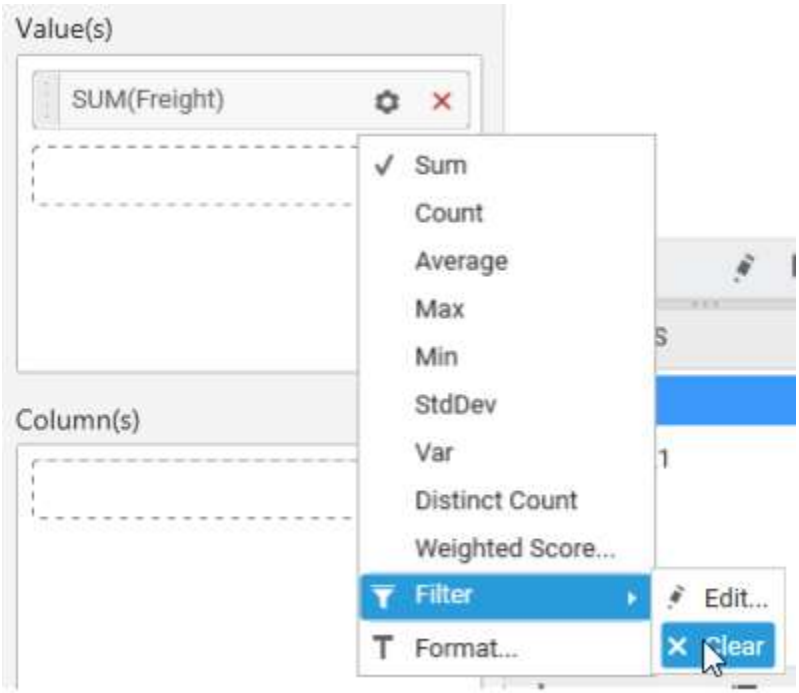
Configure the compare operator and the value to be compared against the selected column values. Click Apply to apply the filter settings to the widget. Now, the applied settings get saved. These applied settings will be retained on reopening this dialog.

Click Reset to reset the changes made in the dialog. Doing so, will reset the filter applied to that column before.

Note: Reset will be in enabled state only when there was a filter applied already. Apply will be in enabled state only when there are pending changes in the dialog to save.

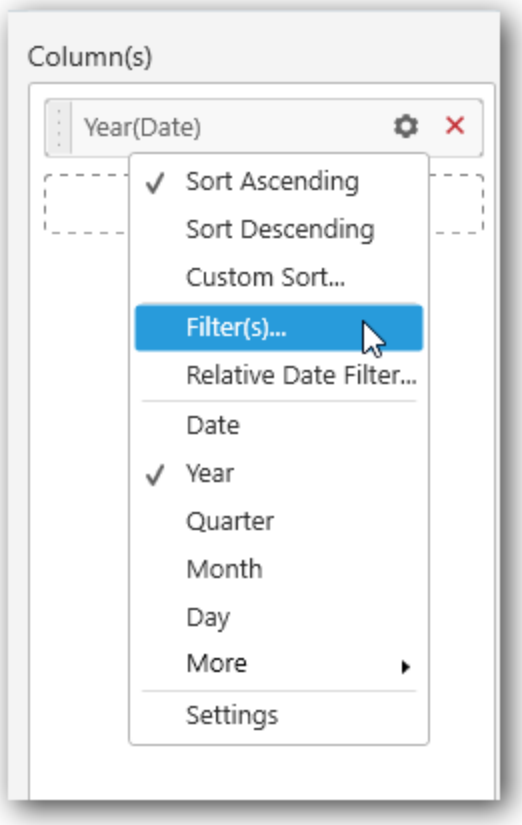


To clear the filter applied to a measure column, click the Clear menu item in the Settings drop down menu. This menu item will be in enabled state only when there was filter configured already to that column.

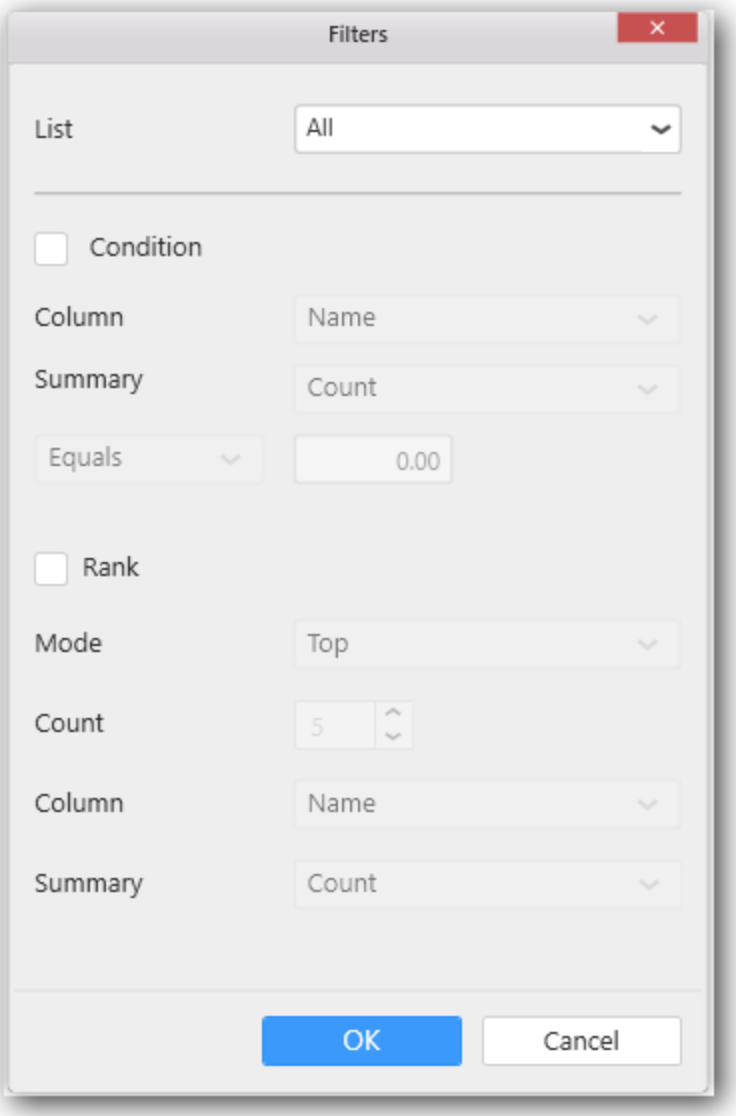


Configuring Filter for Dimension Column

Filter for dimension column can be configured through opening the Filters dialog from the Filters… option in the Settings drop down menu.



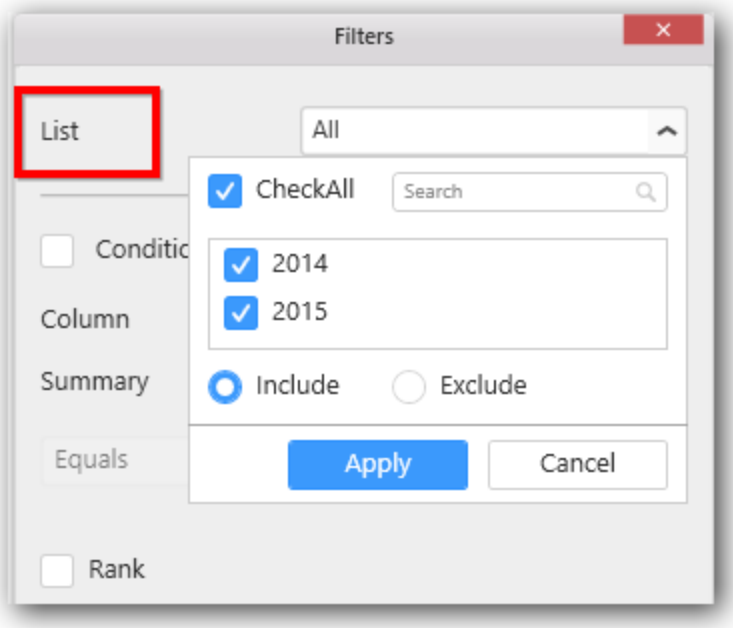
Filters dialog will open like below.



This dialog consist of three different filters, which can be applied individually or in combined manner.

Item-based Filtering

Through this filtering, you can filter out the specific items from consideration.



Click the button at right to drop down the list holding the individual values of that column. You can check or uncheck each of those or as a whole (through CheckAll option).

Include and Exclude options, lets you choose whether to consider checked items for inclusion or for exclusion in filter respectively.

The Search text box helps you to filter the view in case of larger list and required to search for specific one.

Click Apply to save the changes you made in the filter drop down list.

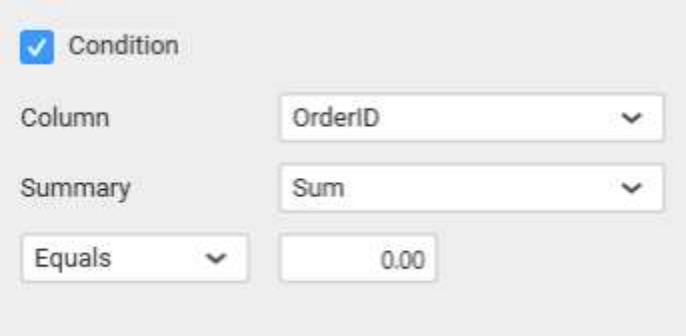
Click Cancel, if you require to cancel the changes made in the filter drop down list.

Click OK in the Filters dialog to save the changes made w.r.t this item-based filtering.

Click Cancel in the Filters dialog, if required, to ignore the changes made.

Condition-based Filtering

Through this filtering, you can impose a condition based on which the filter need to be applied. This filtering option is disabled, by default. You can enable it by clicking the Condition checkbox. Doing again, will disable it.



Set the column near the Column label, by which the filter criteria need to be defined. Set the summary type near Summary label, based on which aggregation need to be applied over the selected column. Set the Compare Operator and the value to compare against the column values.

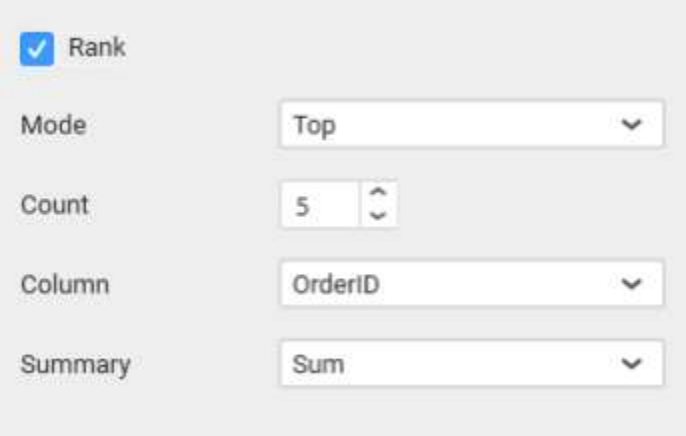
Note: The Summary setting is not applicable for widget bounded with SSAS data source.

Click OK in the Filters dialog to save the changes made w.r.t condition-based filtering.

Click Cancel in the Filters dialog, if required, to ignore the changes made.

Rank-based Filtering

Through this filtering, you can filter top or bottom n items based on a different column with the selected aggregated calculation applied. You can enable it by clicking the Rank checkbox. Doing again, will disable it.



Set the top or bottom mode to consider, near the Mode label. Set the number of records to filter near the Count label. Set the column name based on which the filter need to be applied, near the Column label. Set the summary type based on which aggregation need to be handled whose output should have been compared with corresponding value in widget bounded data, near the Summary label.

Note: The Summary setting is not applicable for widget bounded with SSAS data source.

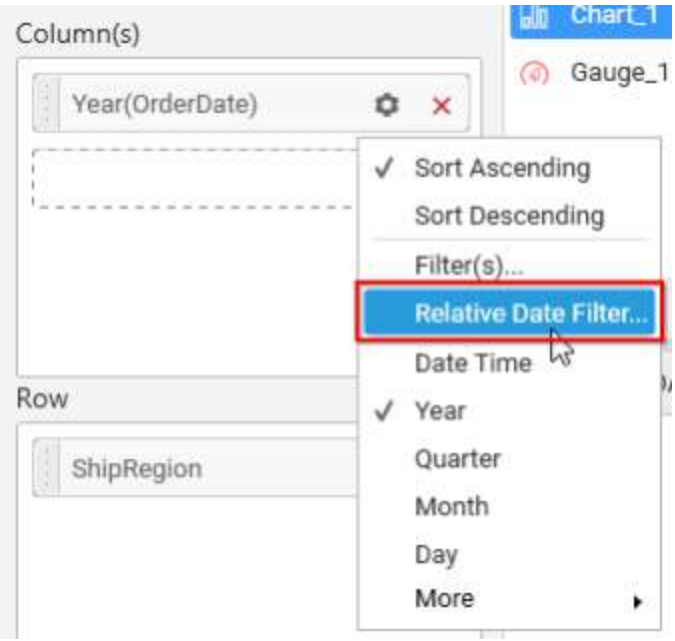
Click OK in the Filters dialog to save the changes made w.r.t rank-based filtering.

Click Cancel in the Filters dialog, if required, to ignore the changes made.

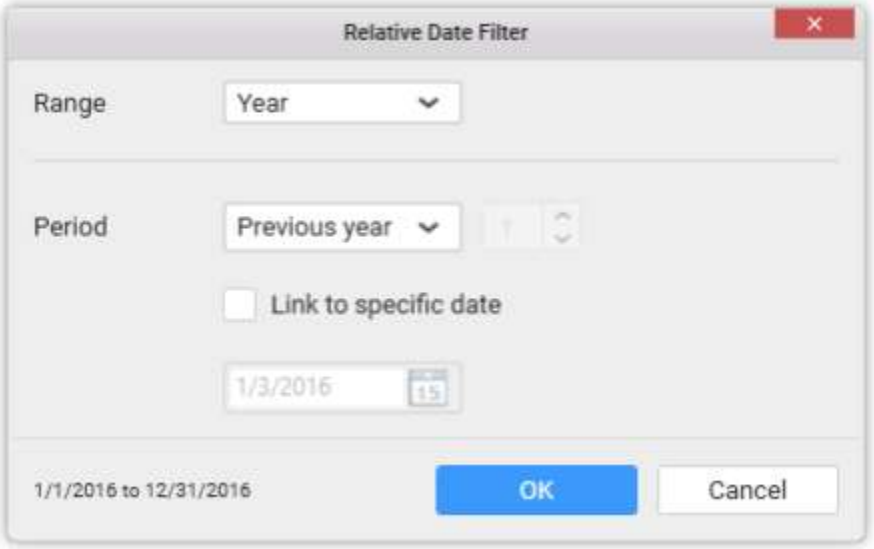
Note: When all these three filters were configured and applied, the records satisfying the criteria of all these filters will be considered by widget into it.

Filtering based on Relative Dates

This filtering is applicable only for date time type dimension columns. This is an added option for date time columns and hence can be found only for them in Settings drop down menu.

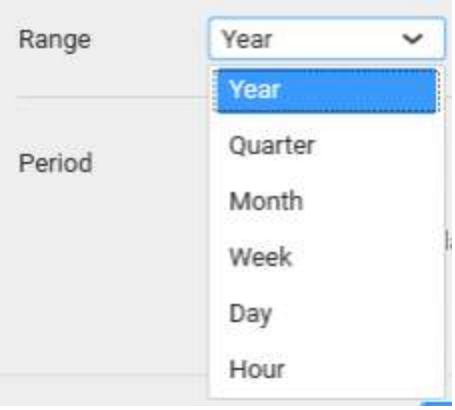


Click the Relative Date Filter… option in Settings drop down menu corresponding to that date time column, to open the Relative Date Filter dialog.

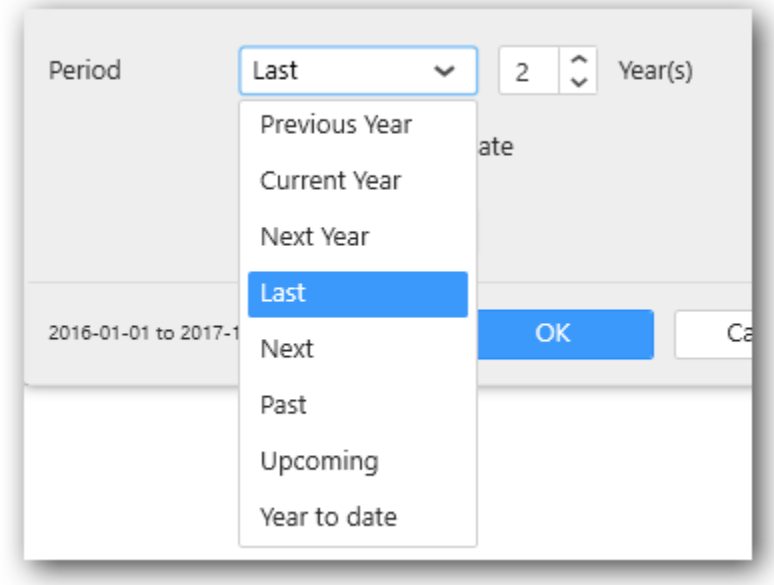


This dialog allows you to configure the filter for date time column relatively. i.e. if you configure Previous 1 week as filter, it will get you the last 1 week data in that widget. After a week, if you check, it will show you the previous 1 week from the date you are viewing. Through this, you no longer required to set the filter statically and change week by week to see the previous week results.

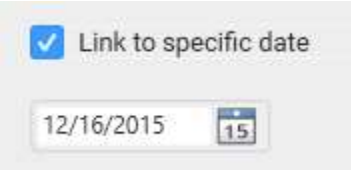
In this dialog, set the range you are going to look over through the widget. It may be year, month, week, etc. like below, near the Range label.



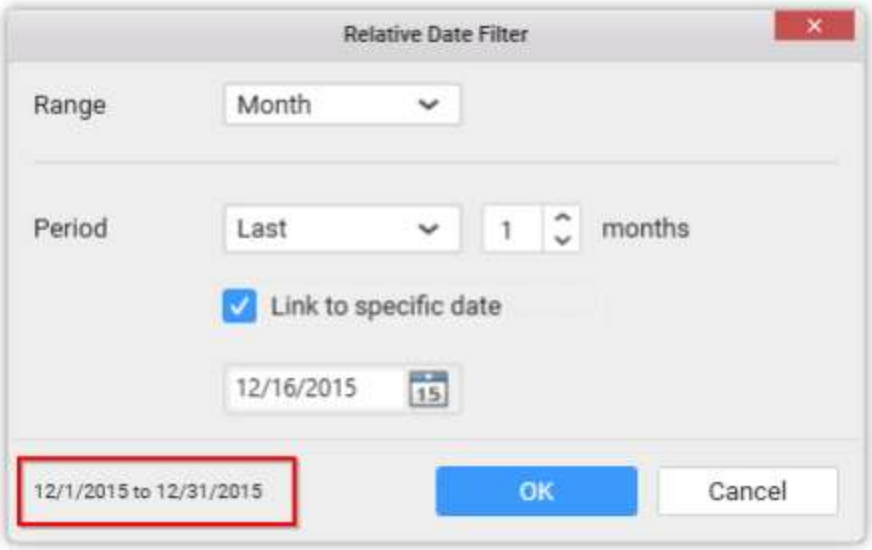
Set the period you would like to see in the defined range, near the Period label like below.



You can even pin the relative date filter to a specific date such that, it will then behave statically like other normal filters. i.e. last 1 month returns the period w.r.t the specific date such that, it always return the data of the same month even after 1 month.



You can see the preview date (highlighted below) in the dialog based on the values you set.



Click OK to save the changes and apply the filter.

Click Cancel if required, to ignore the changes made.

Configuring Label Parameters

Note: Label parameters are not supported for the SSAS data source currently.

Using label parameters in the dashboard widgets title section

You can configure the label parameters by using the field name in the Header of widget. Use the below format to configure the label parameter.

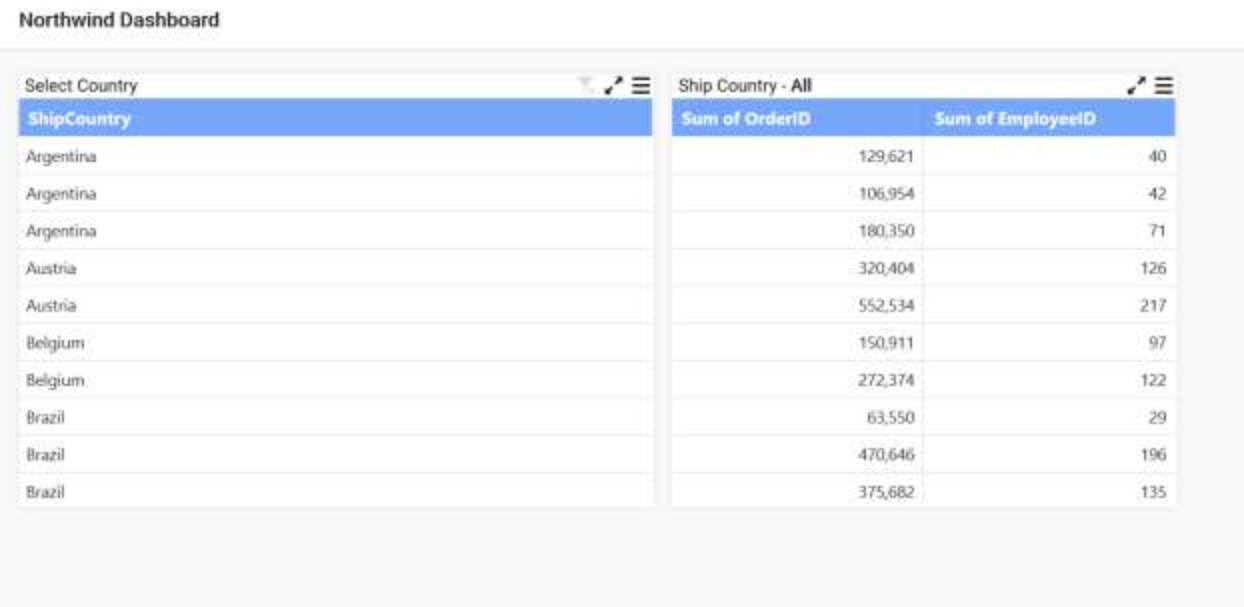
Syntax: {{"{{"}}:Column_Name{{}}}} **when single data source is present**

Or

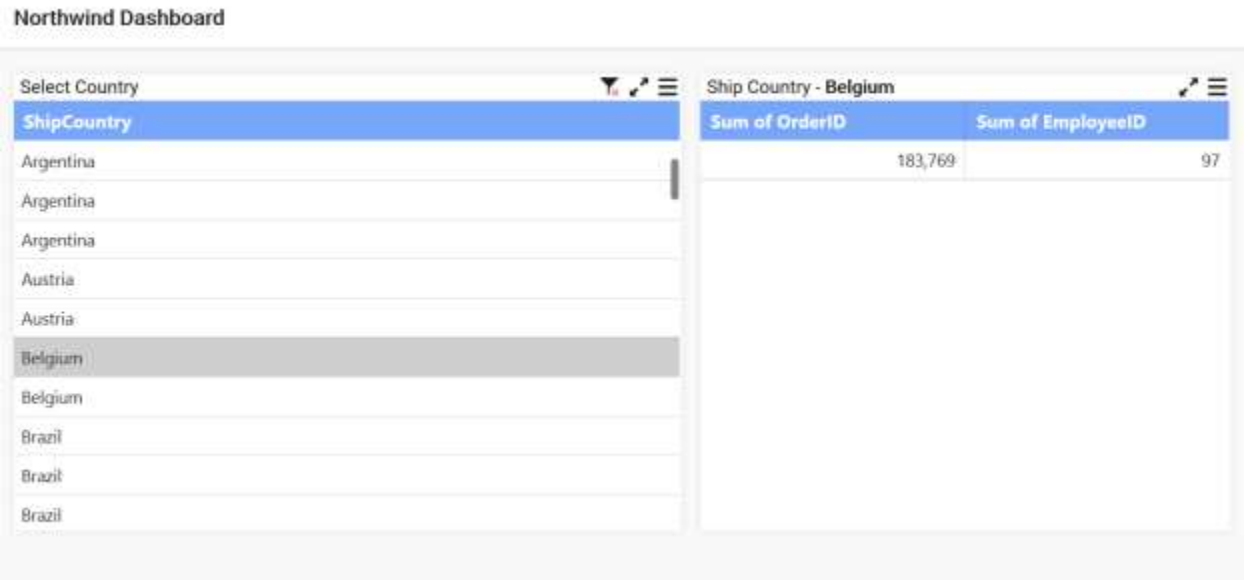
Syntax: {{"{{"}}:DataSource*Name.Column*Name{{}}}} **when more than one data source is present**.

For example, The header text of the Grid widget as: Ship Country - {{"{{"}}:ShipCountry{{}}}.

Now, the dashboard will show the label parameter for all the countries like Ship Country - All



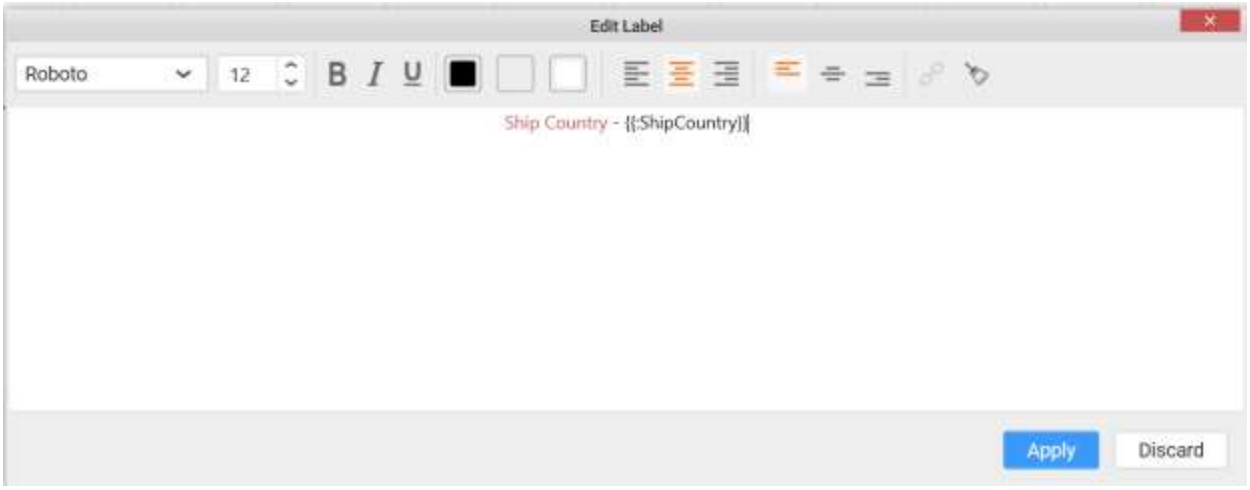
You can select the required country to display by selecting the country name in the grid widget. Based on the selected country the values will be displayed on the widget.



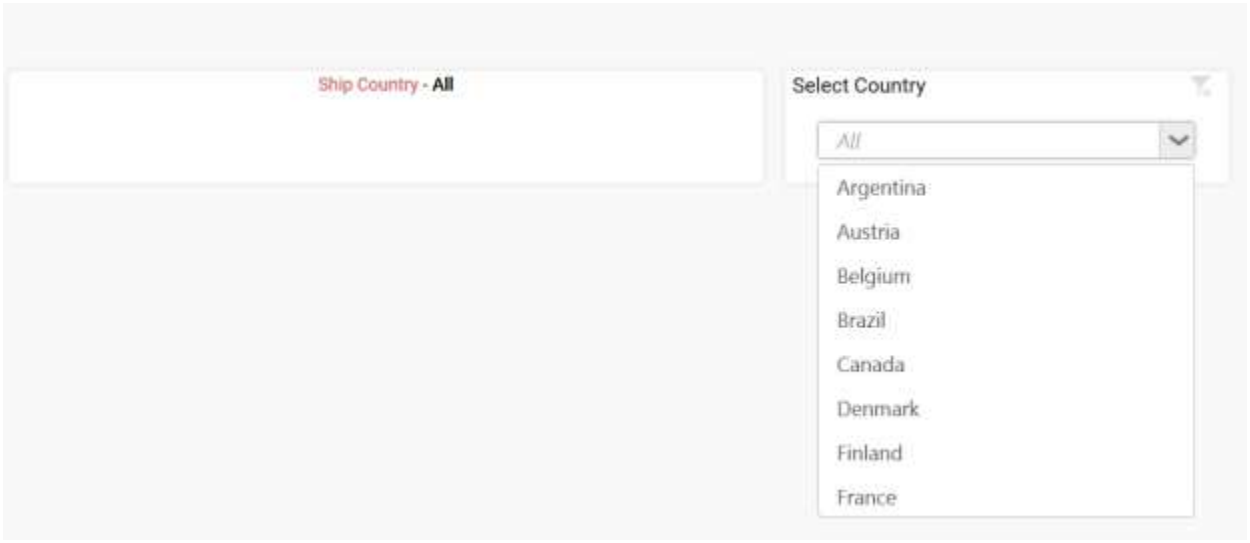
Using label parameters in the label widget

You can also use label widget to configure the label parameters.

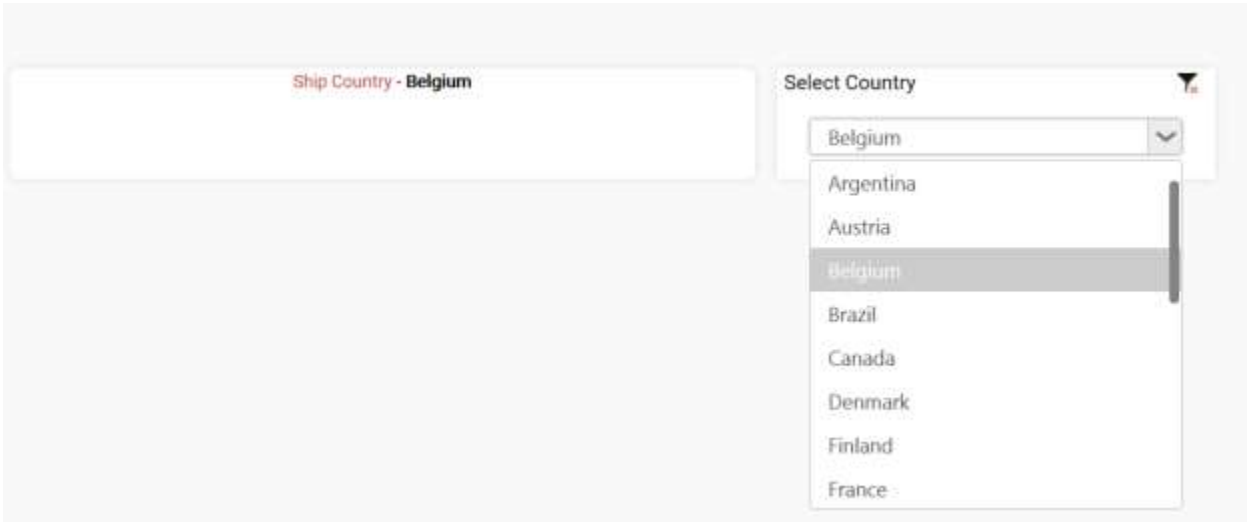
Drag and drop the label widget into the design pane and click the edit label to add the label parameters.



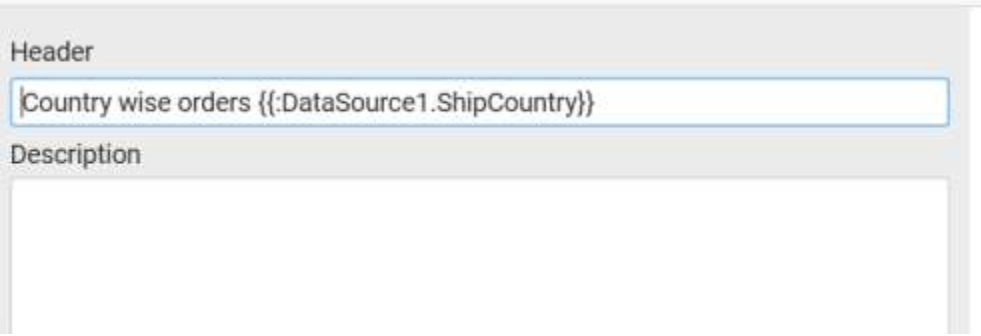
While previewing the dashboard, initially the parameter of the ship country will be shown as All in the label widget.



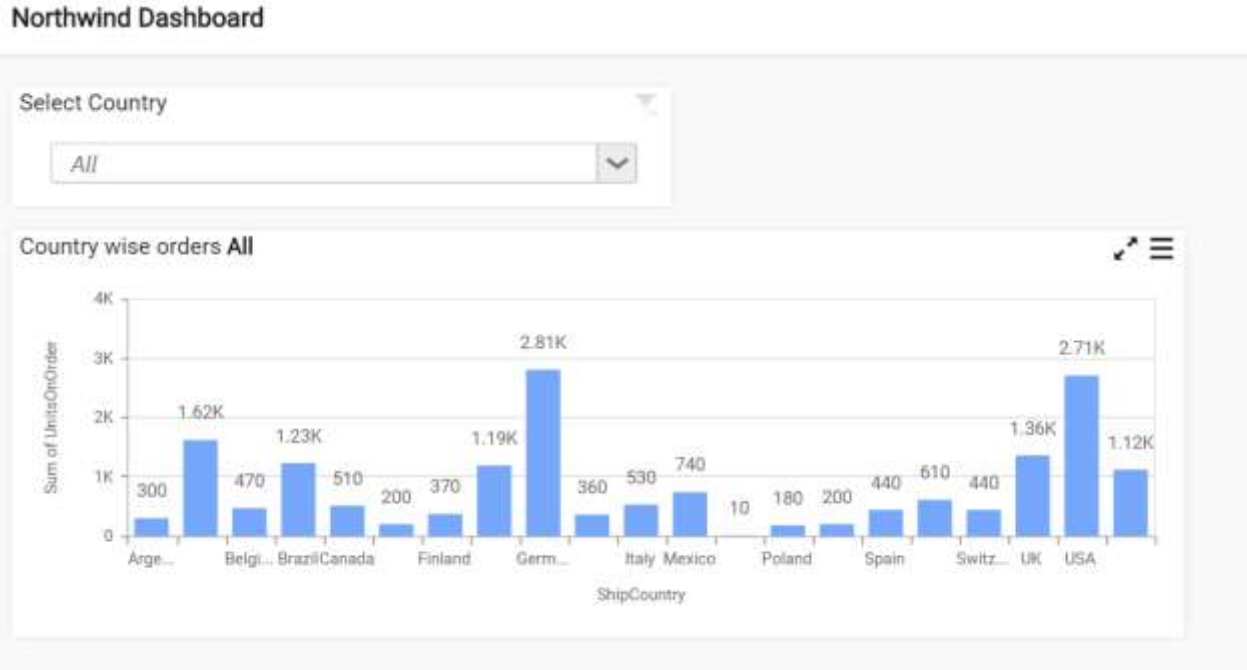
While selecting the particular country, it will show that Country name in the label widget.



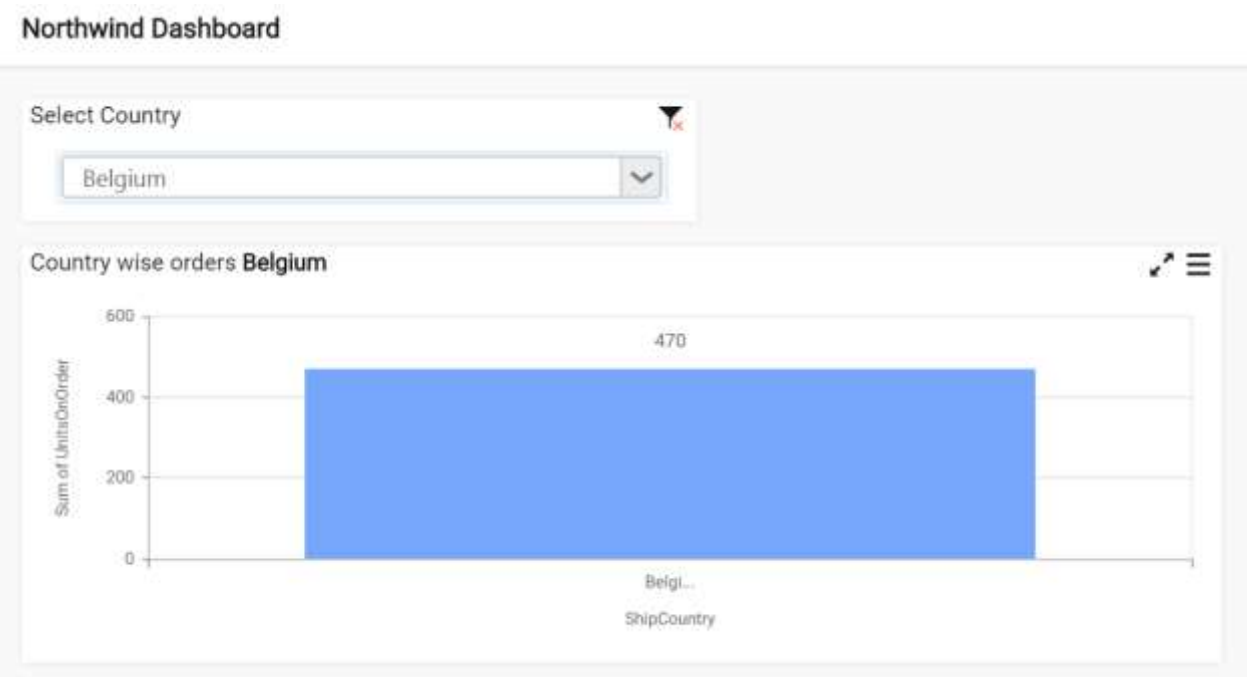
You can add the data source before the parameters in Header of the properties pane, when more than one data source can be present in the dashboard.



While previewing the dashboard, initially the values of the parameters will be shown as All.



While selecting the particular country, you can show that country name in the widget.



Functions supported in label parameters

You can configure the label parameters using the column name with functions like Sum, Count, Average, Min, Max, StdDev, Var, Distinct Count, Weighted Score for numeric values.

For Date Time type, the supported functions are Date, Year, Month, Quarter, Day, Day Month Year, Minutes, Second, Month Year, Date Hour, Day of Week and Week of Year.

For Text type, the supported functions are Count and Distinct Count.

Use the below format to configure the label parameter.

Syntax: {{"{{"}}:function(Column_Name){{}}}} **when single data source is present**

Or

Syntax: {{"{{"}}:function(DataSource*Name.Column*Name){{}}}} **when more than one data source is present**.

Function Name	Supported Type(s)	Description	Example(s)
Sum	Number	This function will return the summation of the given column in number format.	Total Quantity - {{"{{"}}:Quantity{{}}}} or Total Quantity - {{"{{"}}:sum(Quantity){{}}}}
Average	Number	This function will return the average of the given column in number format.	Average freight amount is {{"{{"}}:avg(Freight){{}}}} or {{"{{"}}:average(Freight){{}}}}

Count	Number, Text, Date Time	This function will return the count of the given column in numeric format.	Number of records : {{"{{"}}:count(Quantity){{}}}} or Number of records : {{"{{"}}:count(ShipCountry){{}}}} or Total Transactions : {{"{{"}}:count(InvoiceDate){{}}}}
DistinctCount	Number, Text, Date Time	This function will return the count of distinct values in the given column in number format.	Number of unique records are {{"{{"}}:dcount(OrderID){{}}}} or {{"{{"}}:distinctcount(OrderID){{}}}} Countries count is {{"{{"}}:dcount(ShipCountry){{}}}} or {{"{{"}}:distinctcount(ShipCountry){{}}}}Total unique transaction count is, {{"{{"}}:dcount(InvoiceDate){{}}}}
StandardDeviation	Number	This function will return the standard deviation of the column values in number format.	Standard Deviation of Quantity : {{"{{"}}:stdev(Quantity){{}}}}
Minimum	Number	This function will return the minimum value in the given column in number format.	Minimum value of Quantity: {{"{{"}}:min(Quantity){{}}}}
Maximum	Number	This function will return the highest value of the given column in number format.	Maximum value of Quantity: {{"{{"}}:max(Quantity){{}}}}
WeightedAverage	Number	This function will return the weighted average of the first column based on the weight given by the second column in number format.	Weighted Average for sales amount against freight is {{"{{"}}:WeightedAvg([SalesAmount],[Freight]){{}}}}
Date	Date Time	This function will return the date value as string, formatted based on the current system culture.	Sales done on {{"{{"}}:date(ShippedDate){{}}}}
DayMonthYear	Date Time	This function will return the date value as string formatted in	Sales done on - {{"{{"}}:daymonthyear(ShippedDate){{}}}}

		DD/MM/YYYY format.	
MonthDayYear	Date Time	This function will return the date value as string formatted in MM/DD/YYYY format.	Sales done on - {{"{{"}}:monthdayyear(ShippedDate){{}}}}
Year	Date Time	This function will return the year in number format.	Revenue for the Year - {{"{{"}}:year(ShippedDate){{}}}}
Month	Date Time	This function will return the month name as MMM format on selected row.	Weather in NYC - {{"{{"}}:monthname(ShippedDate){{}}}}
Quarter	Date Time	This function will return the â€˜Quarter 1/2/3/4â€™ based on calendar year on selected row.	Total sales for the Quarter - {{"{{"}}:quarter(ShippedDate){{}}}}
Day	Date Time	This function will return the day value (1-28/29/30/31) on selected row.	Store rate for the day - {{"{{"}}:day(ShippedDate){{}}}}
Minutes	Date Time	This function will return the minute value (00-59) on selected row.	Stock rate in - {{"{{"}}:minutes(ShippedDate){{}}}} minutes
QuarterYear	Date Time	This function will return the â€˜Quarter 1/2/3/4â€™ and the year based on calendar year on selected row.	Total sales in - {{"{{"}}:quarteryear(ShippedDate){{}}}}
MonthYear	Date Time	This function returns the Month and Year as MMM YYYY format for the selected record.	Weather in NYC on {{"{{"}}:monthyear(ShippedDate){{}}}}

DateHour	Date Time	This function returns the Date and Hour values for the selected record.	Total sales in - {{"{{"}}:datehour(ShippedDate){{}}}}
DayOfWeek	Date Time	This function returns the day value of the week (Sunday to Saturday) for the selected record.	Weather in NYC on {{"{{"}}:dayofweek(ShippedDate){{}}}}
WeekOfYear	Date Time	This function returns the week value of the respective year as number in selected record.	Week of the year is, {{"{{"}}:weekofyear(ShippedDate){{}}}}

Note: Function name of label parameters are case insensitive.

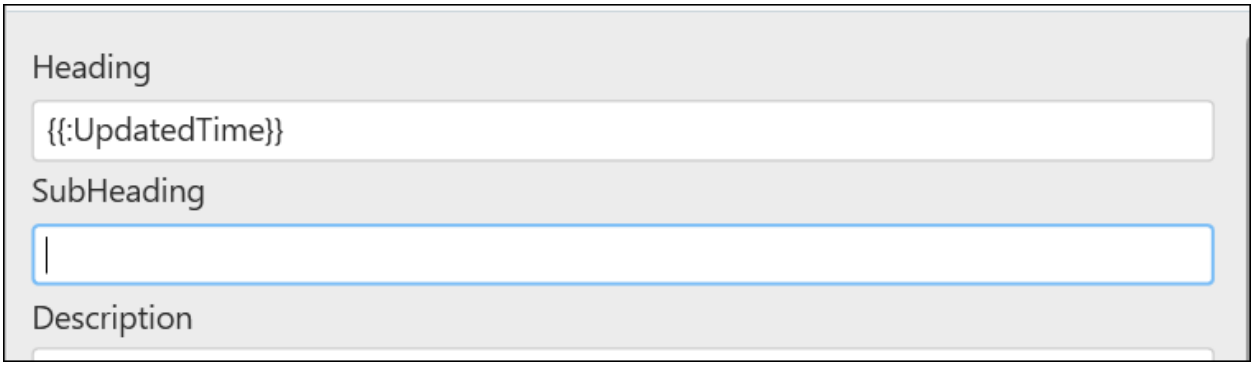
Last refresh time for dashboard/widgets:

You can display last refresh time using the keyword UpdatedTime in the Heading or Subheading section of widgets or dashboards. Use the below format to configure the label parameter.

Note: Syntax: {{"{{"}}:UpdatedTime{{}}}}

For example, The Heading or Subheading text of the Grid widget as:

{{“{{“}}:UpdatedTime{{}}}}.



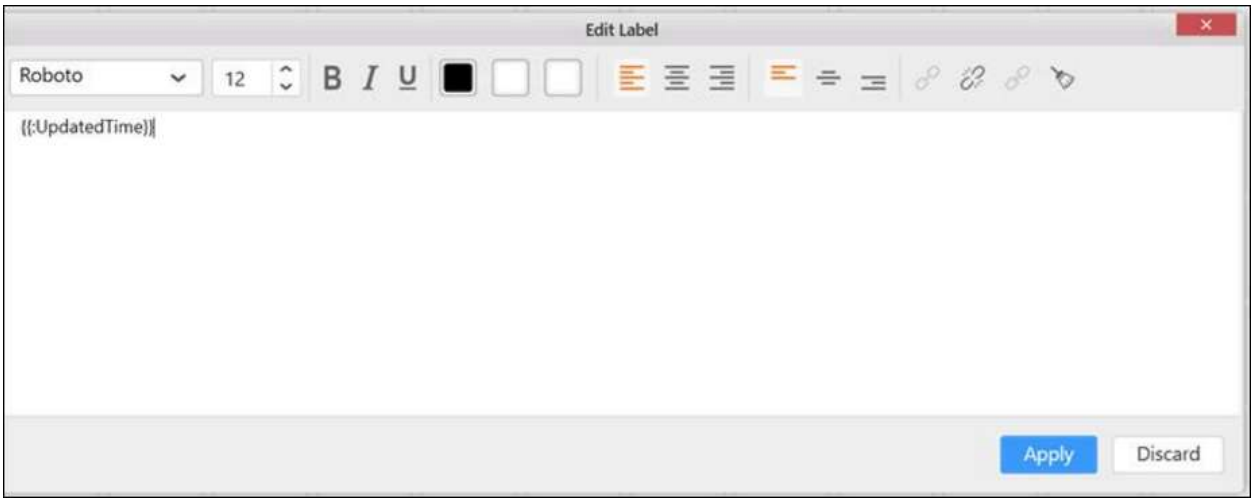
Now, the Heading shows last refresh time of that widget. You can also use other label parameters in Heading and Subheading.

Display last refresh time in label widget

Label widget can also be used for showing last refresh time.

Drag and drop the label widget into the design pane and click the edit label to add the below label parameters.

Syntax: {{"{{"}}:UpdatedTime{{}}}}



Note: To know the list of functions supported by the **UpdatedTime** refer to this Knowledge base article.

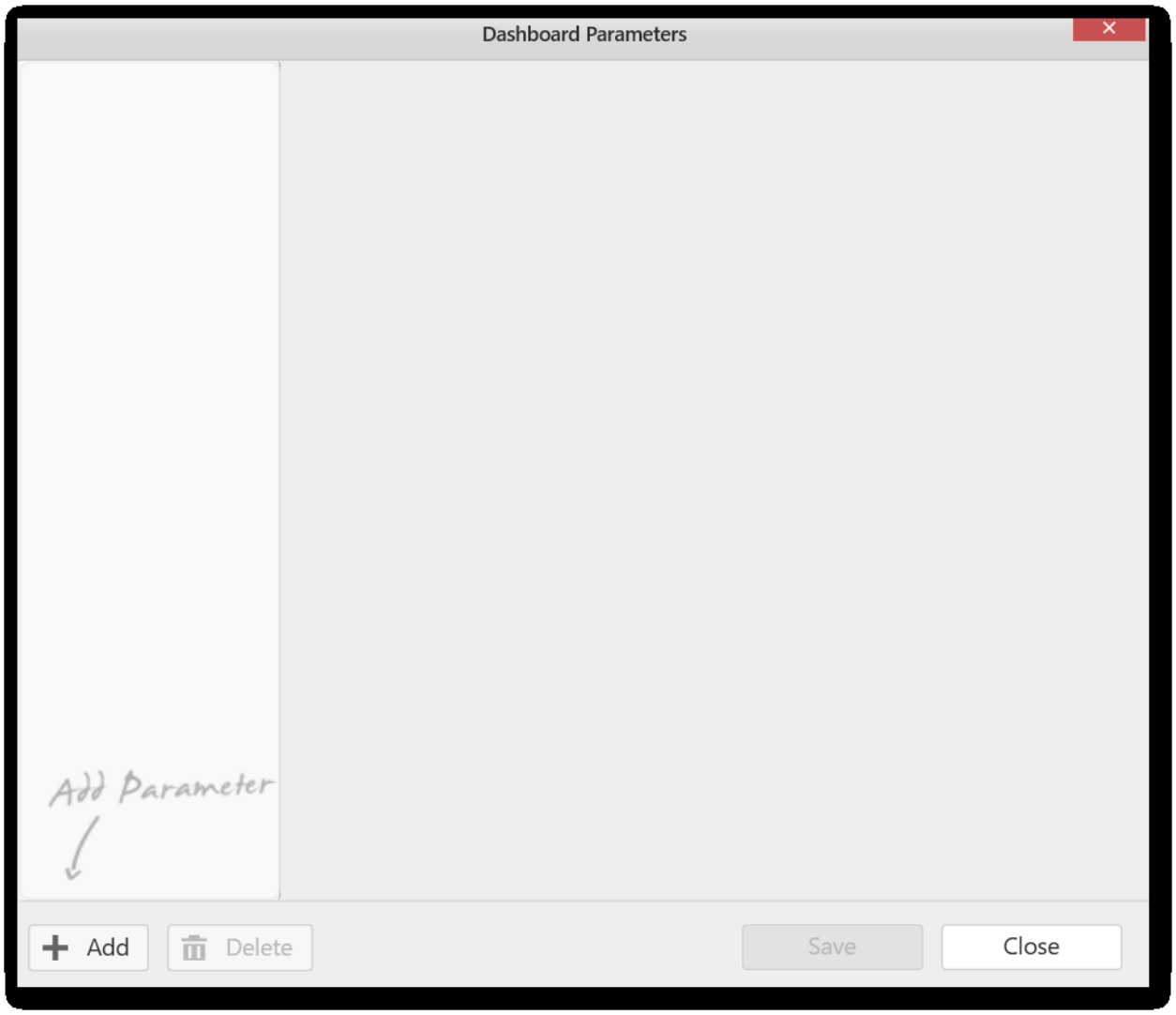
Configuring Dashboard Parameters

Dynamic parameter allows users to execute the custom query/stored procedure dynamically based on the parameter while viewing in the Dashboard Viewer or Dashboard Server. Users can also use the parameter in the expression columns and map the parameter values in user based filters.

Add dashboard parameters

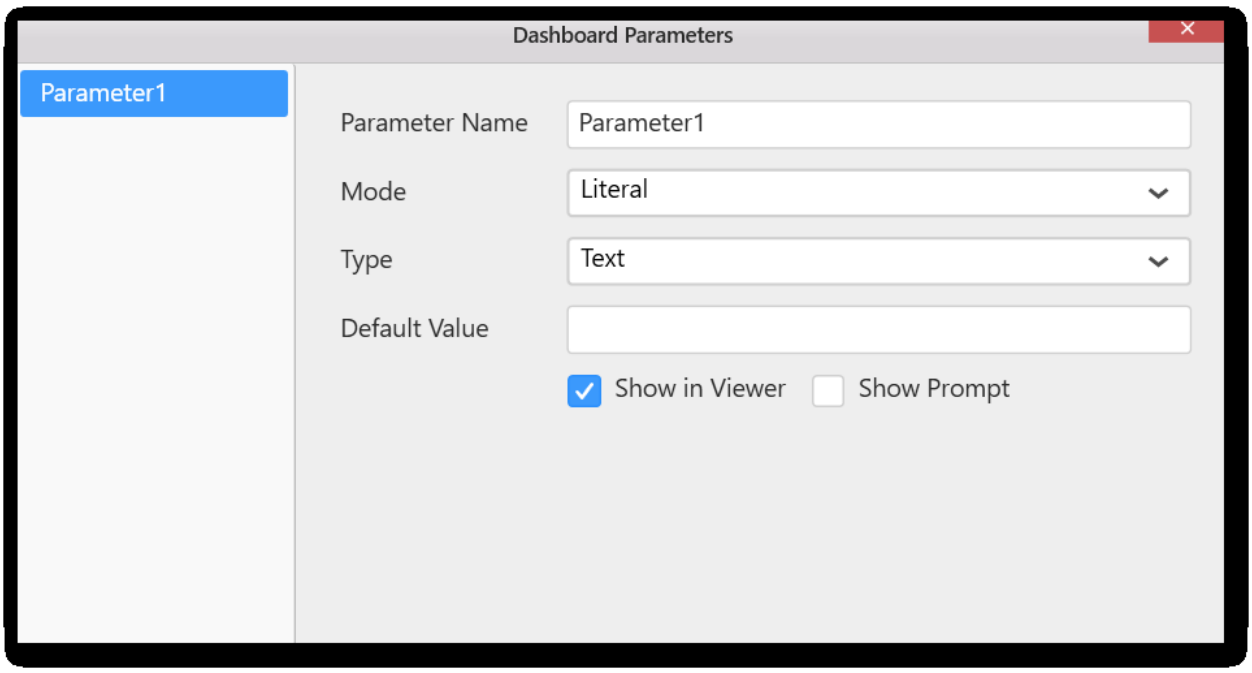
The Dashboard Parameters option is provided in the Dashboard menu as shown in the following image.

Click the **Parameters…** menu item to launch the Dashboard Parameters window. This can be also achieved by using the keyboard shortcut **Ctrl+Shift+D**.



Click **Add** to add a new parameter.

A new dashboard parameter will be added as follows.



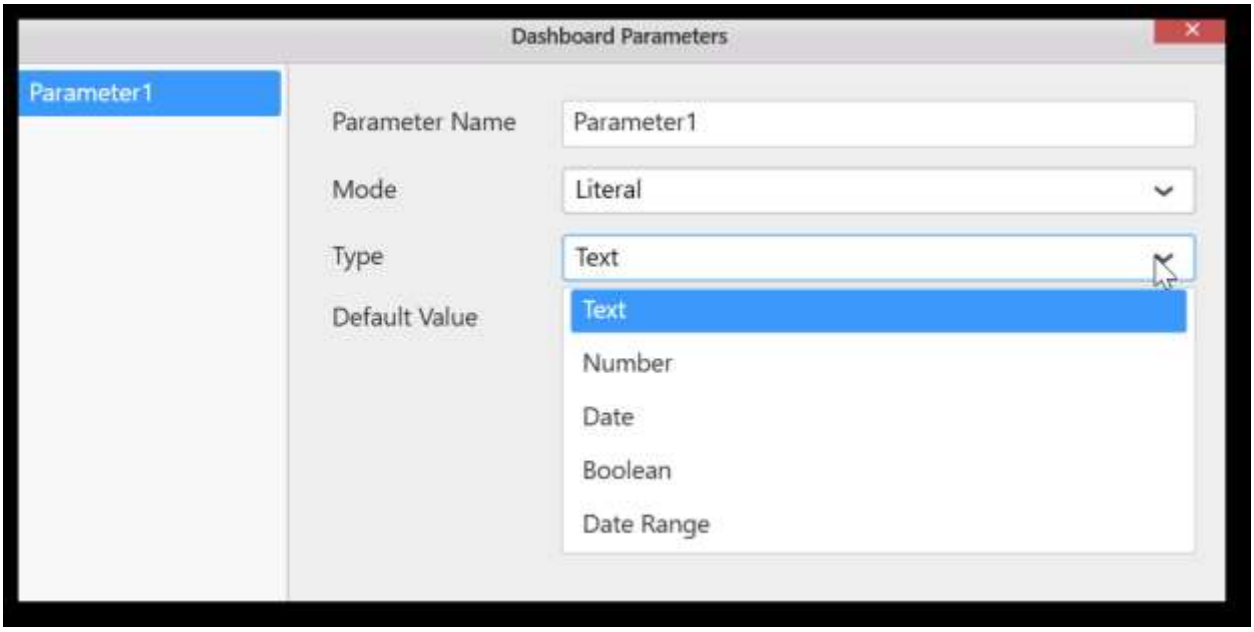
Modes

There are three different modes in the dashboard parameters as follows:

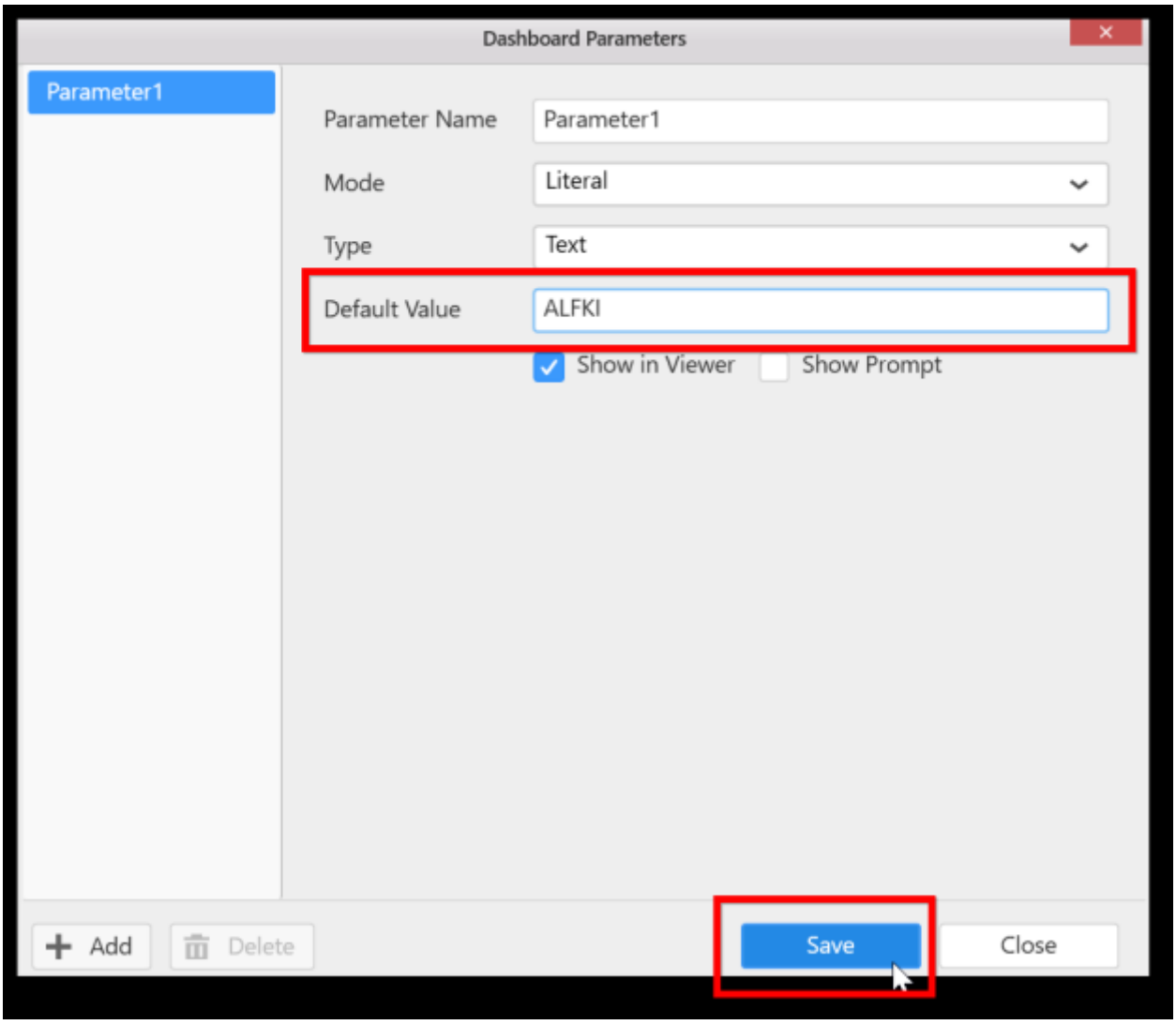
1.Literal: Defines the static value of the parameter. User can give only a single value and can pass the parameter to different modules. 2.List: Users can add values to a collection. The default value will be used for initial execution. 3.Data Column: Users can add a list of values from an existing column of a data source. The values will be copied to the parameter. The default value will be used for initial execution.

Literal mode

The literal mode allows you add the text, number, date , Boolean, and date-range-data type of values.



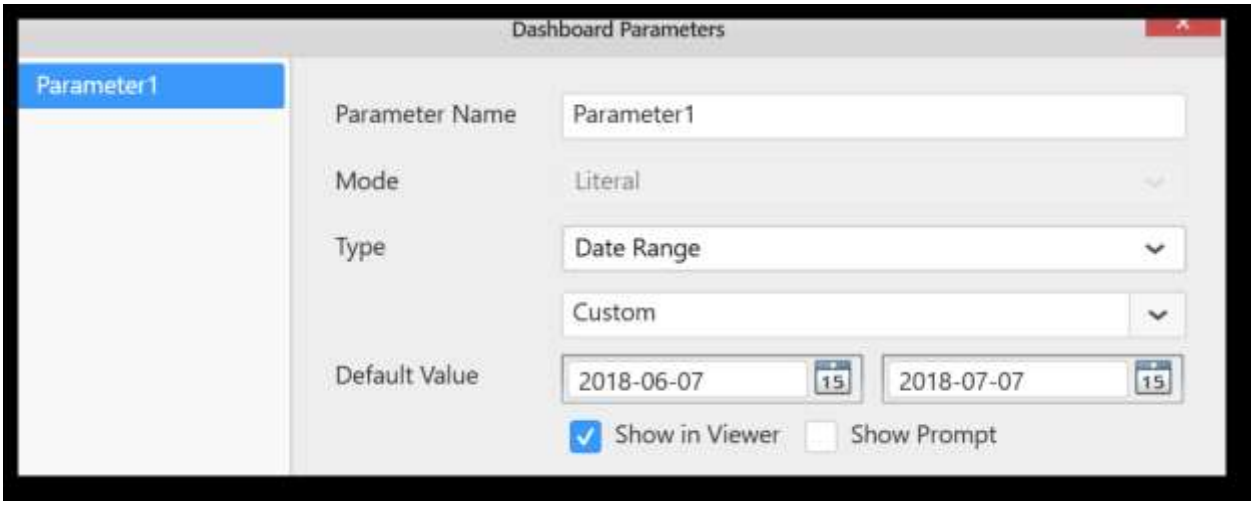
Select the Type and enter the Default Value as shown in the following screenshot.



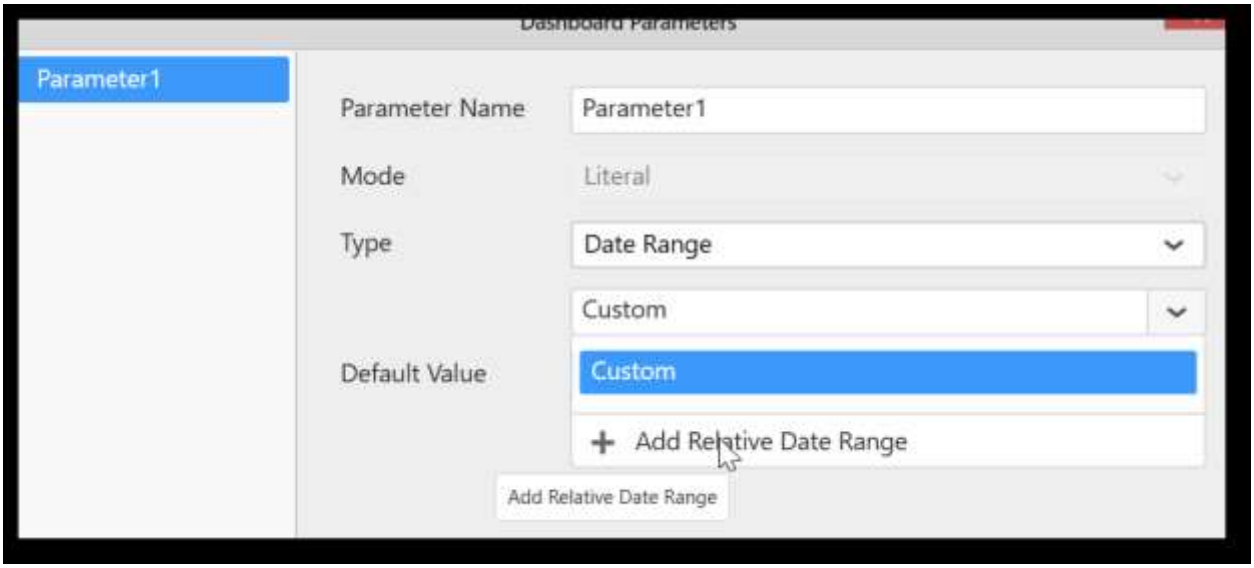
Click **Save** to save the added parameter.

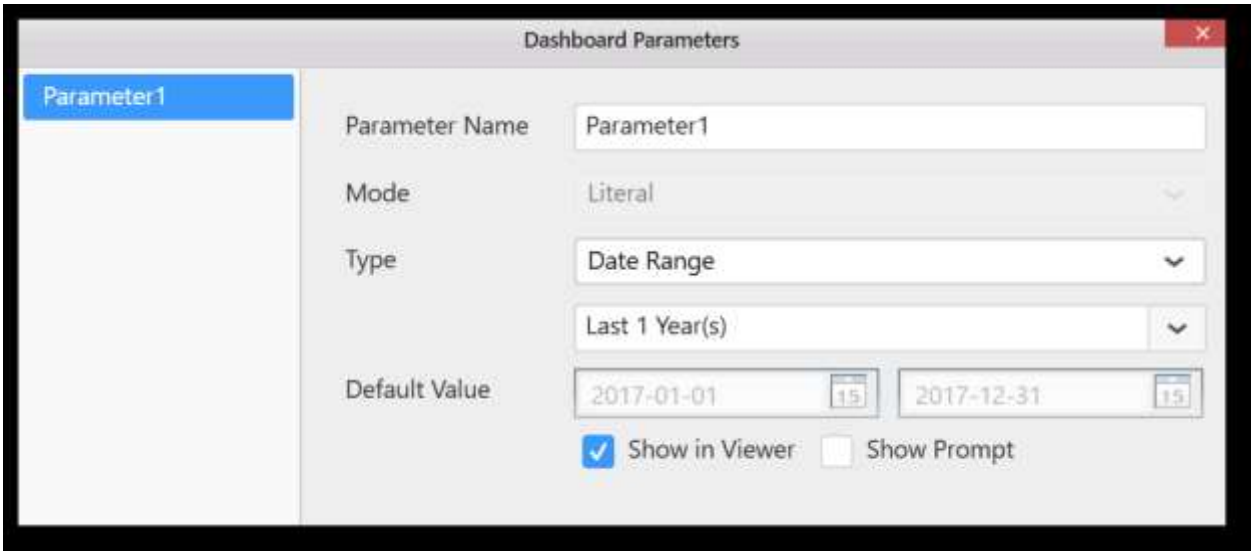
Date range

The date range parameter defines start date and end date that can pass to different modules for execution. Also, the date-range type provides the relative date ranges support.



To add relative date range, click **Add Relative Date Range** option.



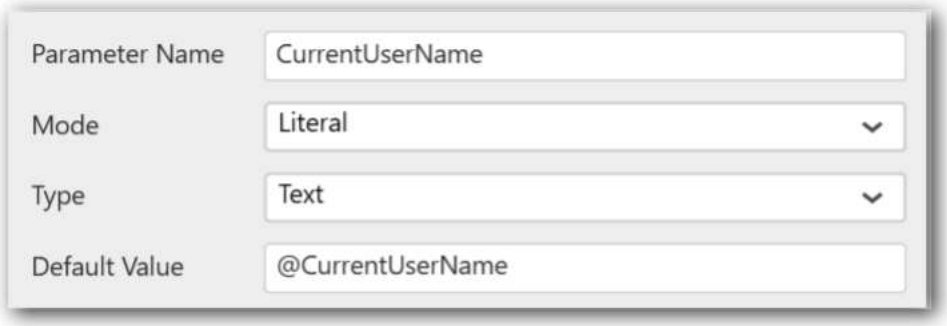


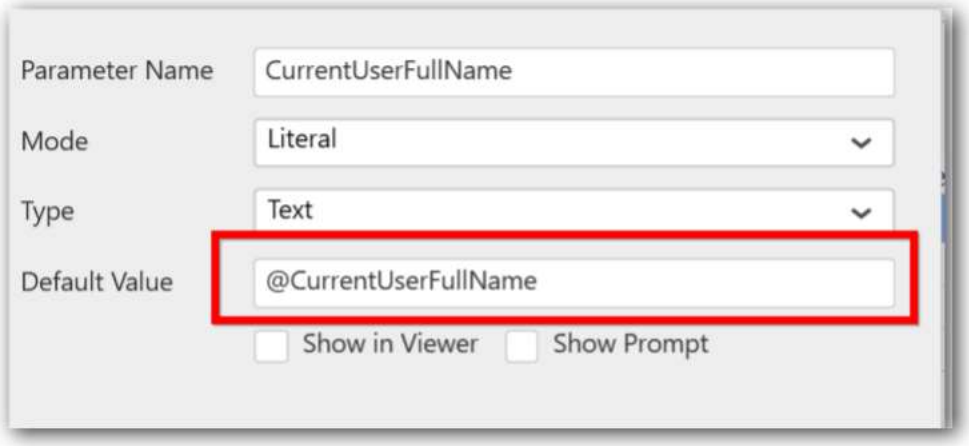
Dashboard Server logged in user details

You can use the name, full name, and email address of the logged in user as a dashboard parameter. To achieve this, use the **@CurrentUserName**, **@CurrentUserFullName**, and **@CurrentUserEmail** in the literal mode.

Information: In Public dashboards , the current user related dashboard parameters are not applicable and if any current user parameter is configured it will be ignored.

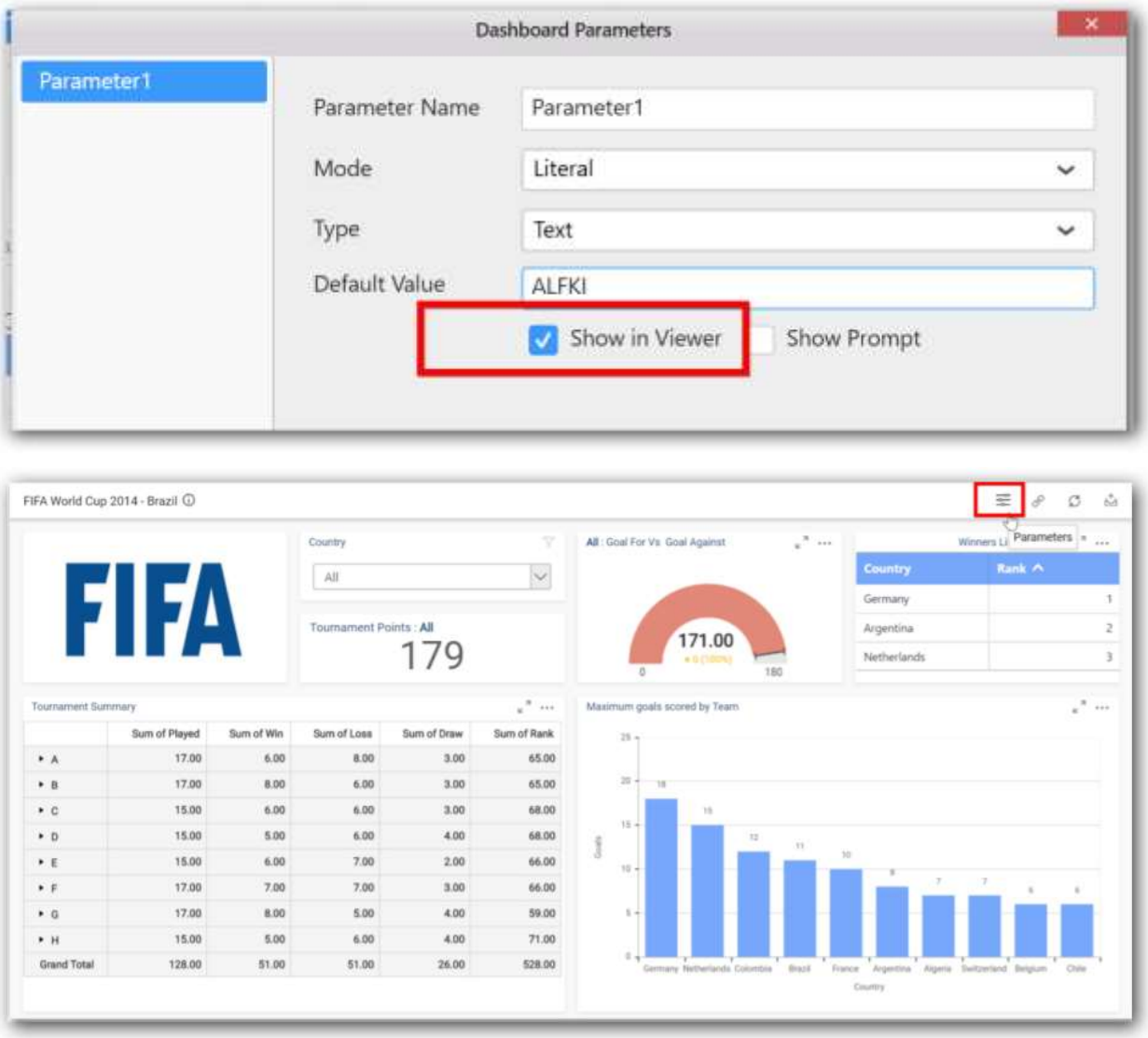
Note: The keywords are case sensitive. While using these syntax, make sure that the Show in Viewer and Show Prompt should be in unchecked state.



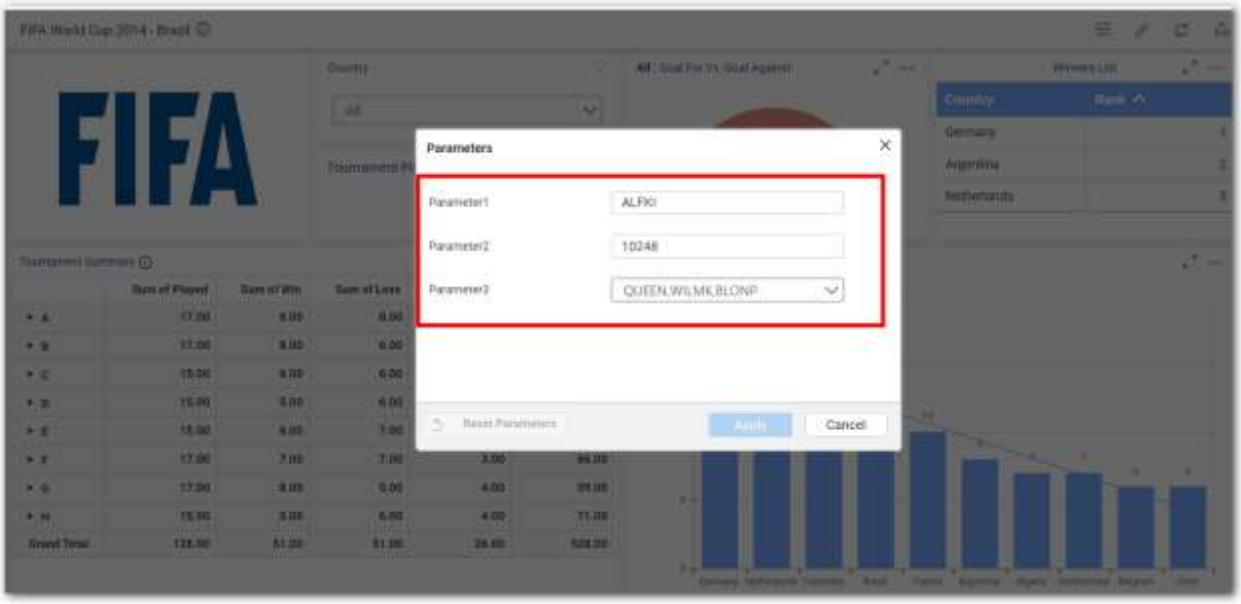


How to view or change the dashboard parameters in Dashboard Viewer

When the Show in Viewer option is enabled in the Dashboard Parameters window, then the parameters added in the dashboard will be available in the viewer tool bar.

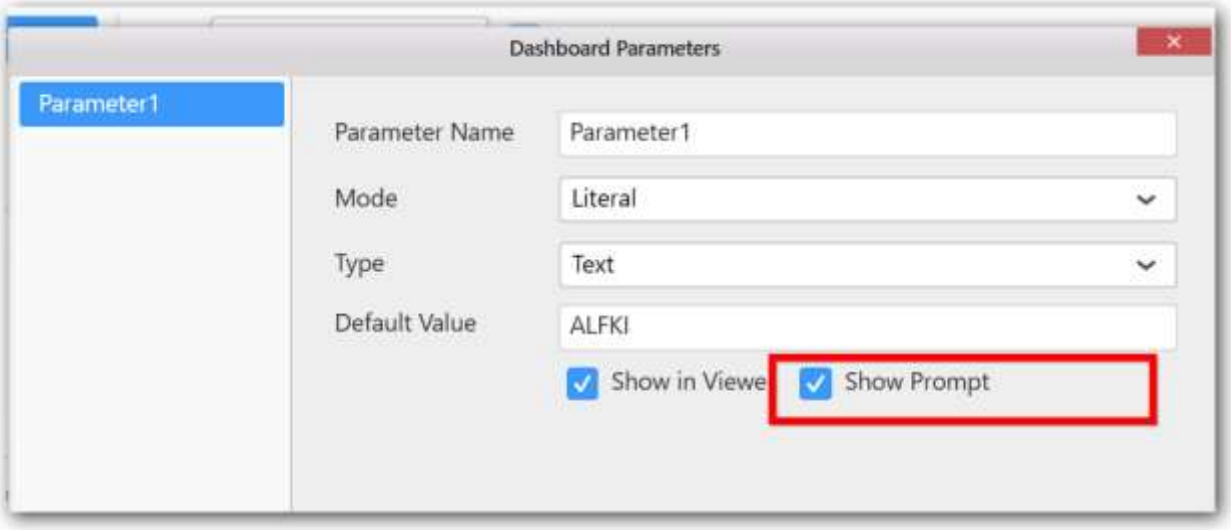


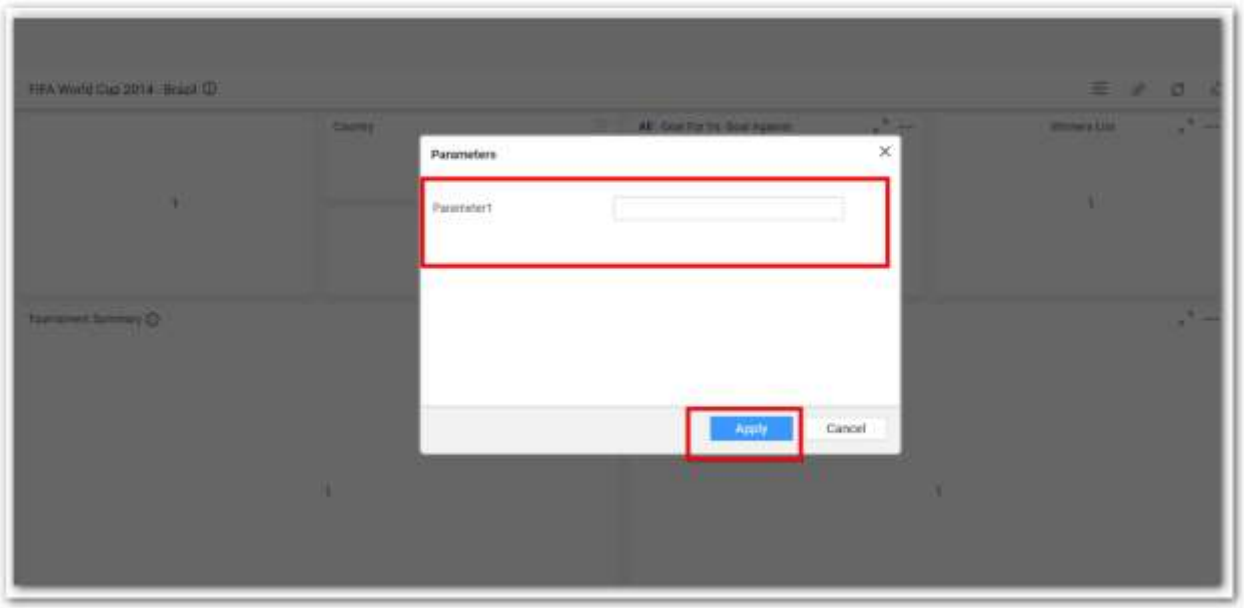
Now, the list of added parameters will be shown in the following dialog.



You can change or select the values and apply the values.

When the Show Prompt option is enabled in the Dashboard Parameters window, the parameters window will be shown as a prompt window while loading the dashboard in the viewer.

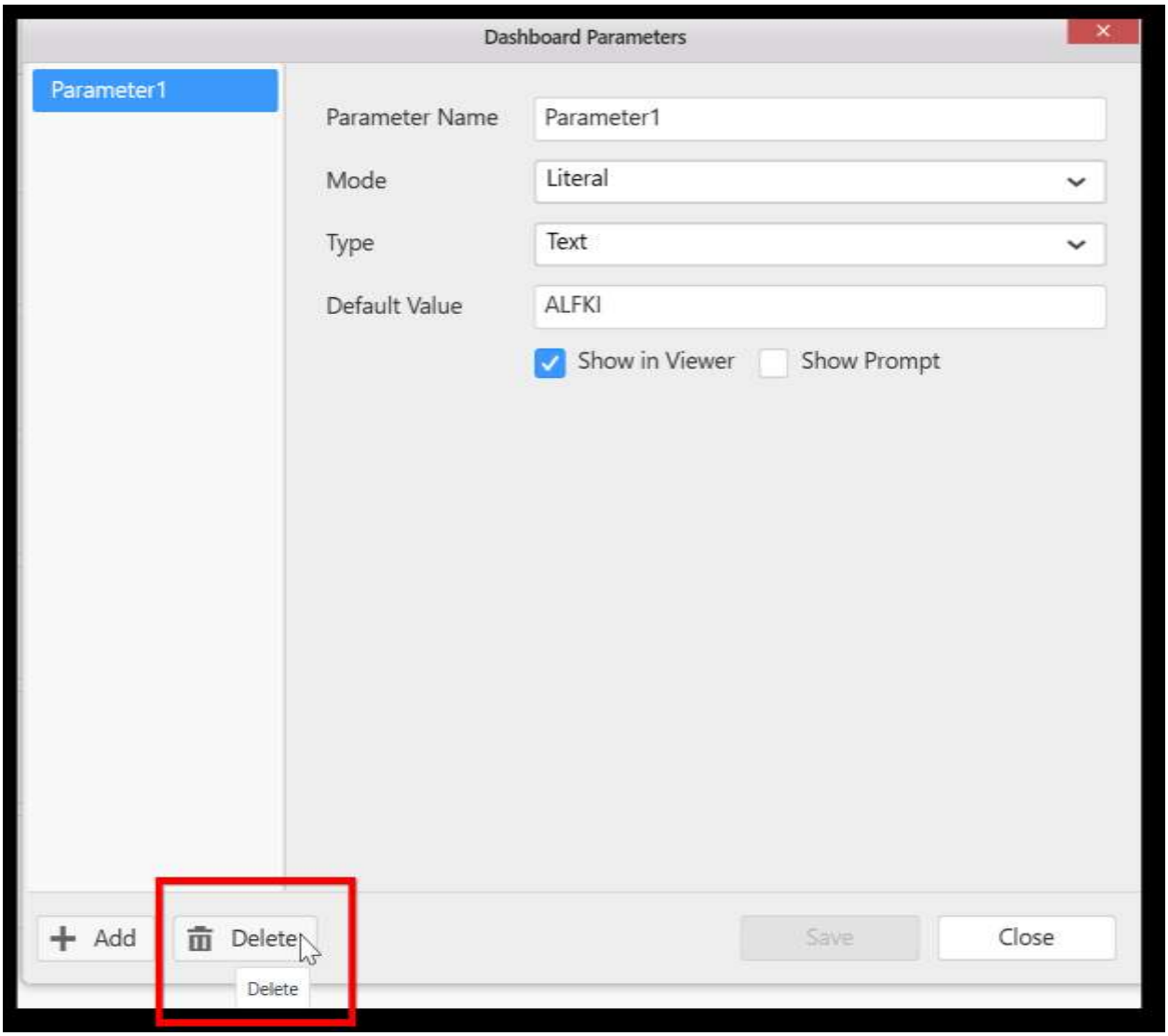




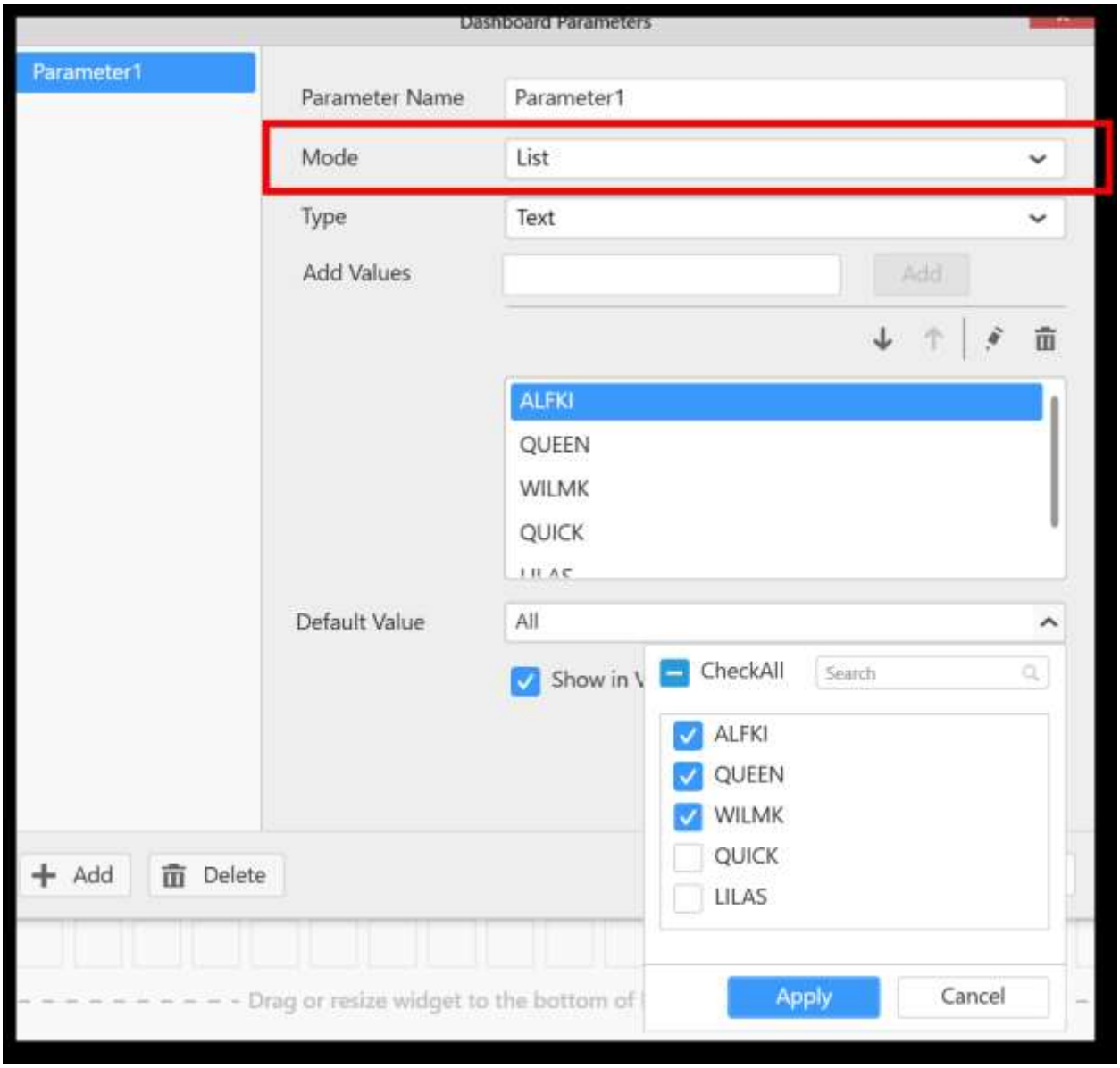
You can select and apply the required value from the window.

Deleting a parameter

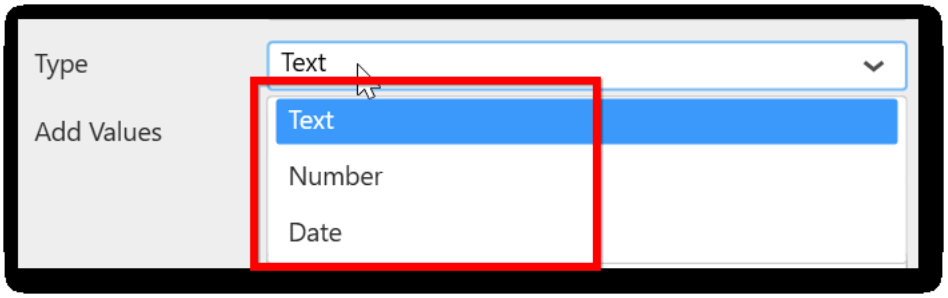
You can delete a saved parameter using the **Delete** option.



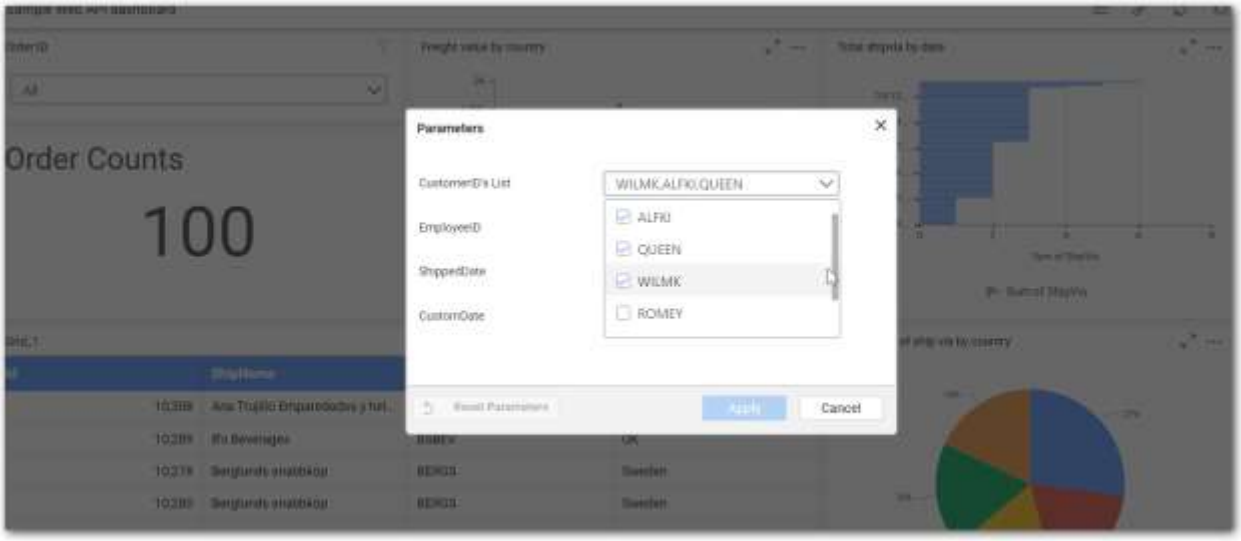
List mode



The list mode allows you add any number of values from a same data type, which can be a text, number, or date.



The list will be shown as drop-down list in the Dashboard Viewer, and you can select a value from the list in the viewer.



Conditions to use list-mode type with multiple selected values

a. Query view: You can use this dashboard parameter with multiple values only using IN query.

Example: Select *from Table_Name where ColumnName in (@{{:Parameter1}})

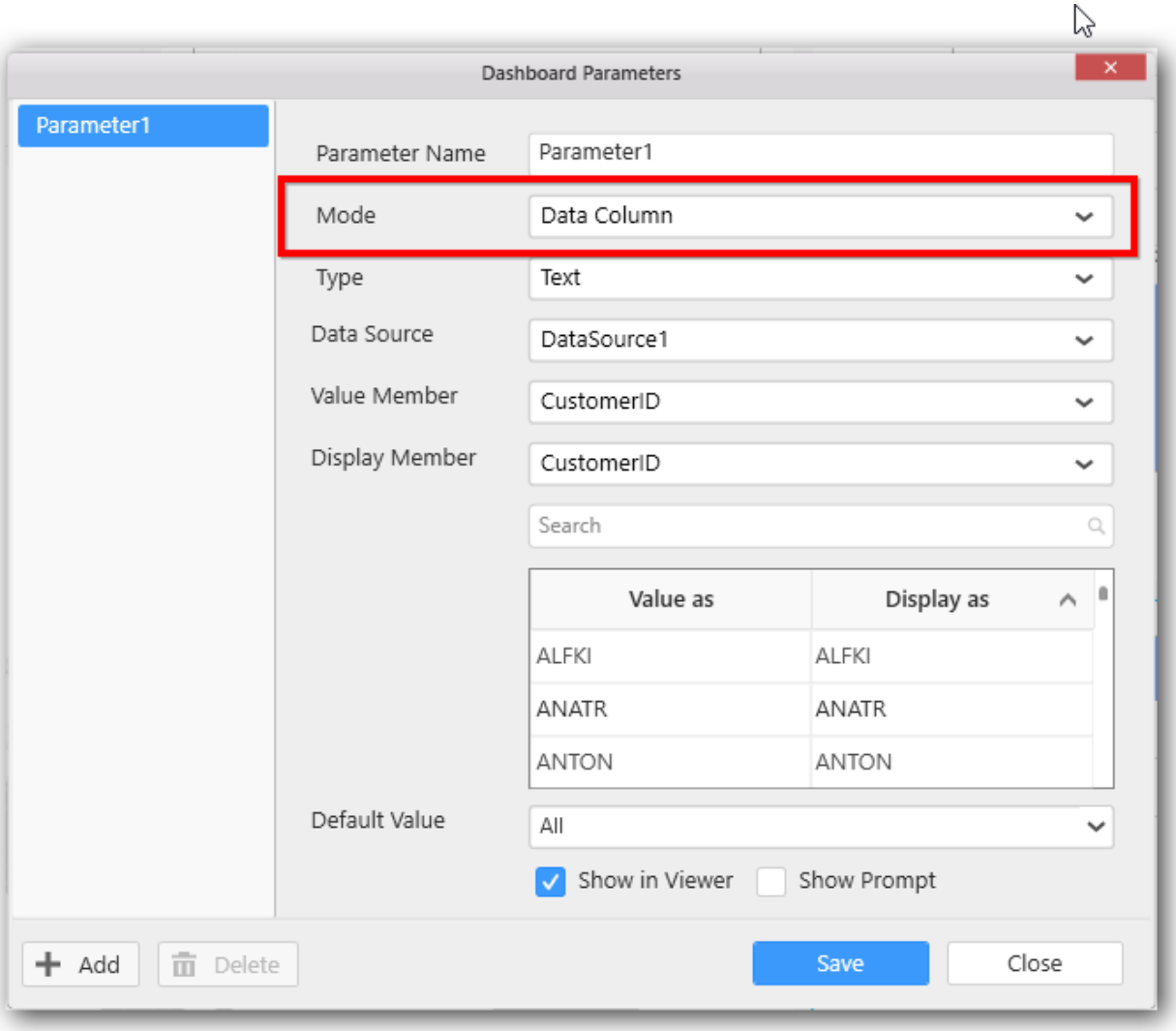
b. Stored Procedure: You should create a stored procedure using the logic trick. Because, SQL stored procedure accepts only single object as parameter. However, you can pass multiple values as comma separated single string to the parameter by a trick. This link explains about the trick.

c. Expression Editor: You can prefer the 'CONCAT' function.

Example: CONCAT([Parameter1])

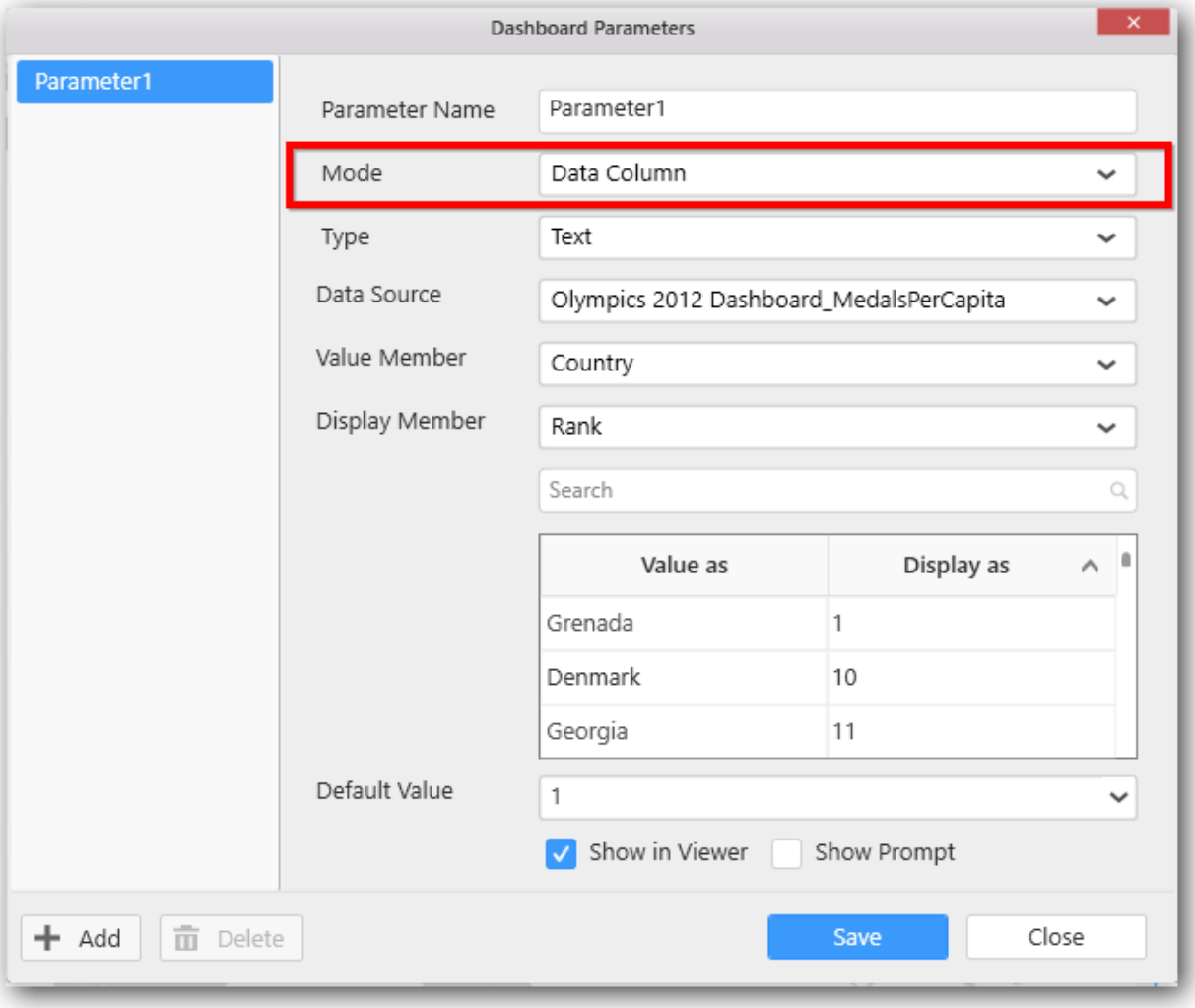
Data column mode

This mode allows you add the fields available in the added data sources.

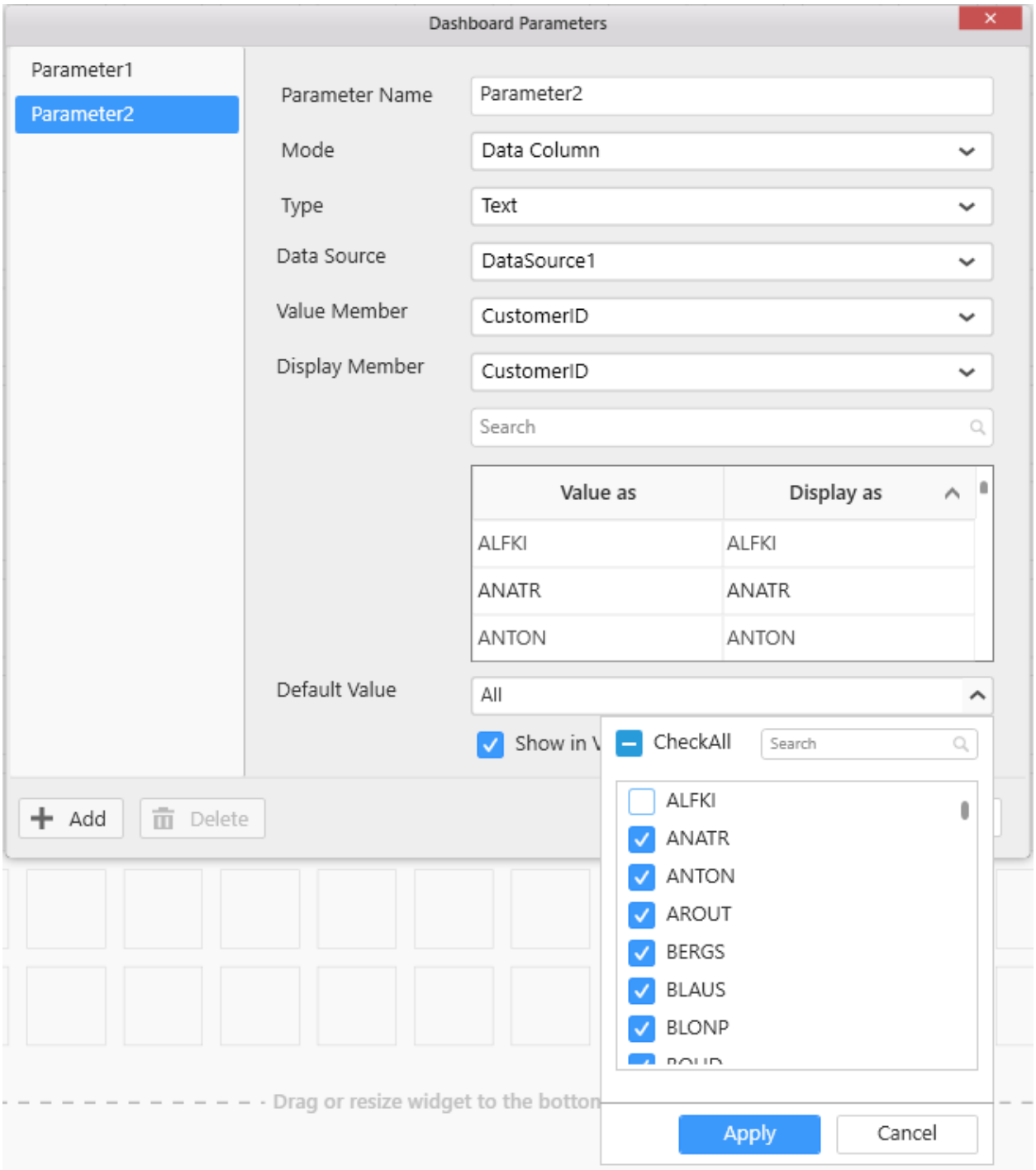


Value and display members

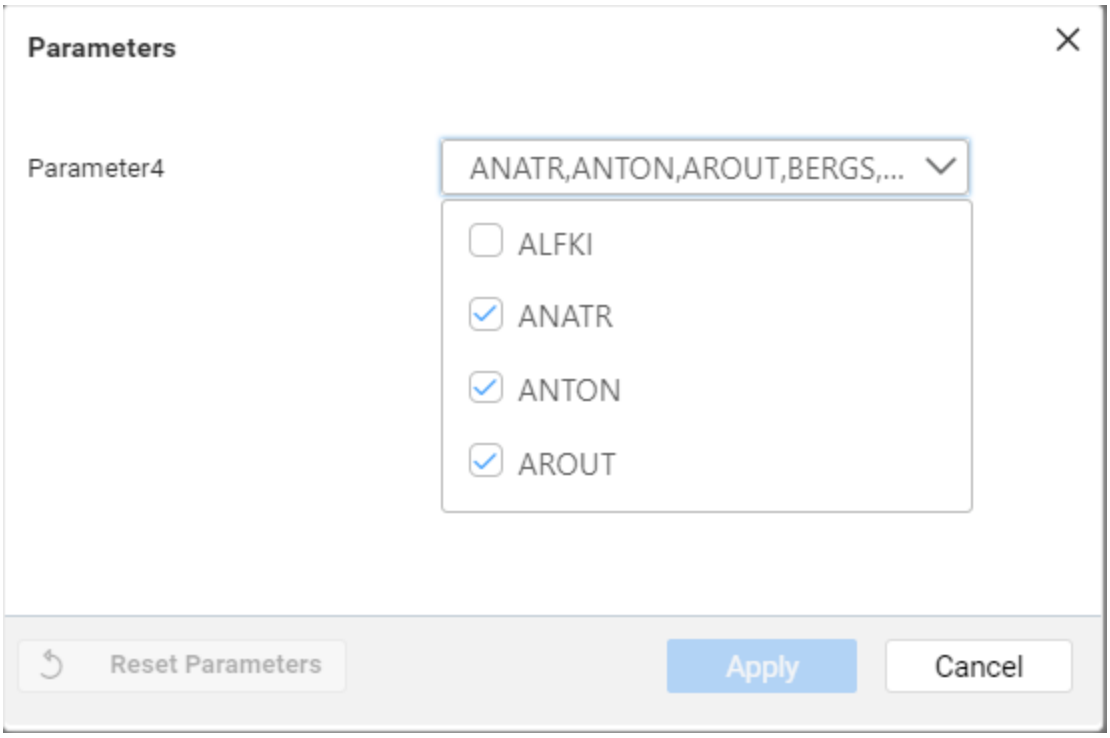
You can select a different Display member for a value member, the display member will be shown in the preview and server:



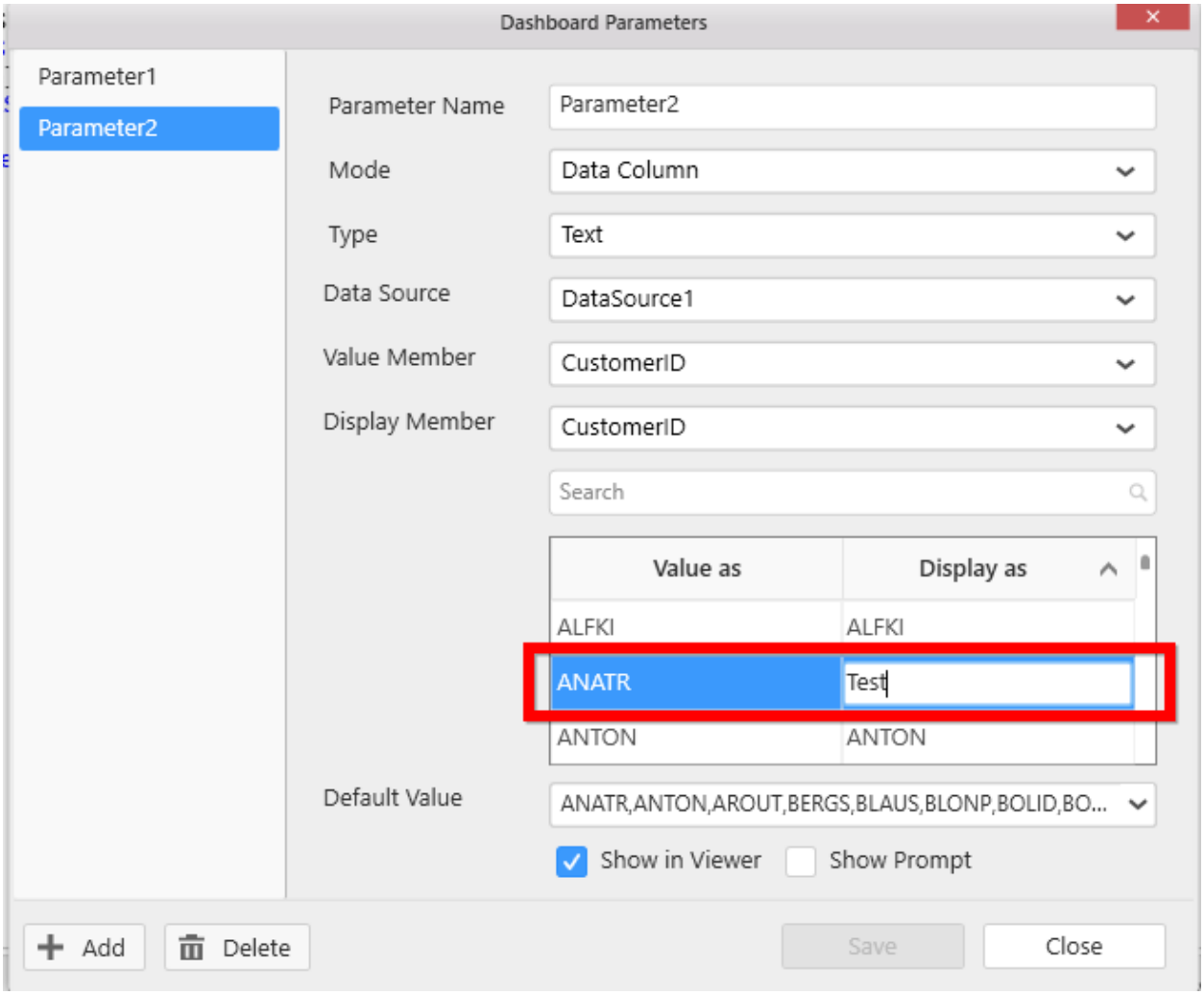
The data column mode allows you select any number of values from a same data type which can be a text, number, or date.



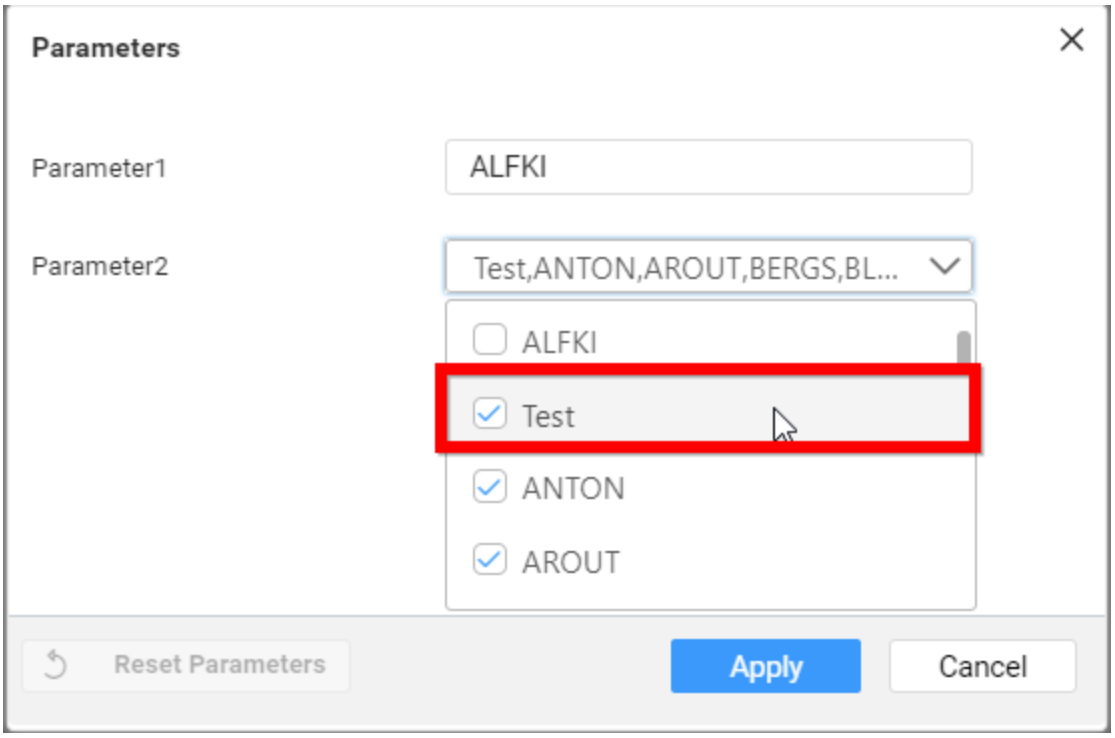
The list will be shown as drop-down list in the Dashboard Viewer, and you can select a value from the list in the viewer.



If you want to edit the display member values, you can able to edit like below.



It will be reflected in the viewer as shown in the following screenshot.



Conditions to use data-column-mode type with multiple selected values

a. Query view: You can use this dashboard parameter with multiple values only using IN query.

Example: Select * from Table_Name where ColumnName in (@{{:Parameter1}})

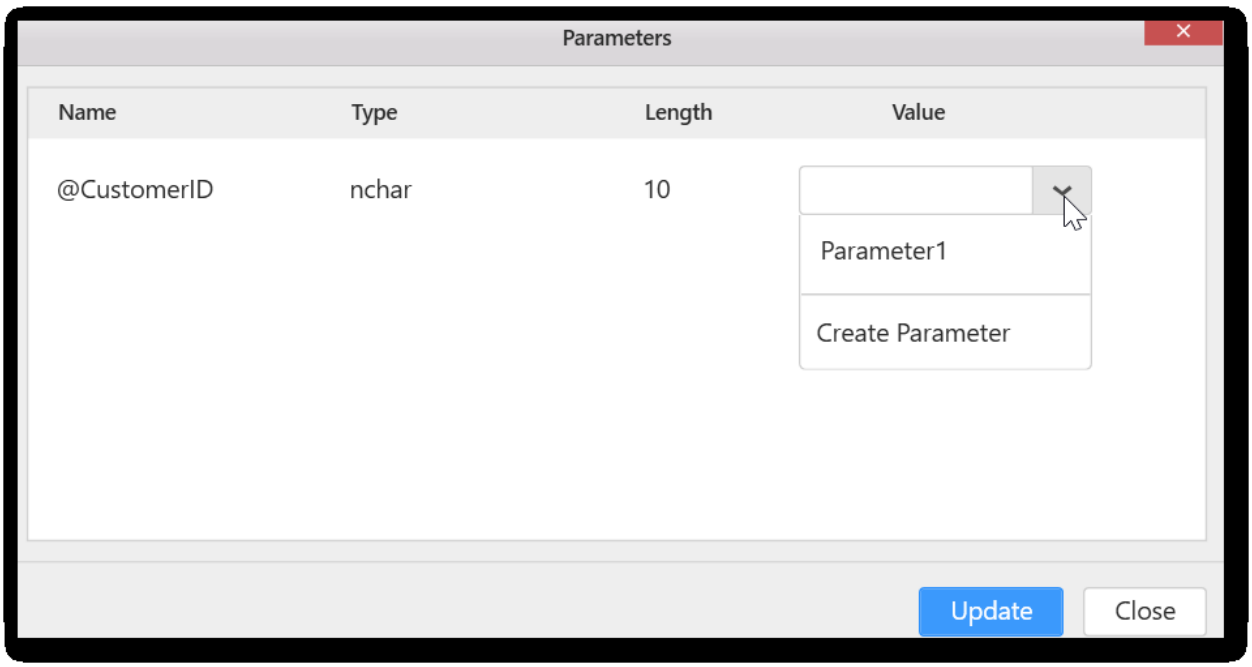
b. Stored Procedure: You should create a stored procedure using the logic trick. Because, SQL stored procedure accepts only a single object as parameter. However, you can pass multiple values as comma-separated single string to the parameter by a trick. Refer to this link for more details.

Uses of dashboard parameters

The dashboard parameters can be used in the stored procedures, Query view,Expression editor, and user based filters.

Stored procedure

The added parameters are listed in the value-drop-down box in the parameters window.

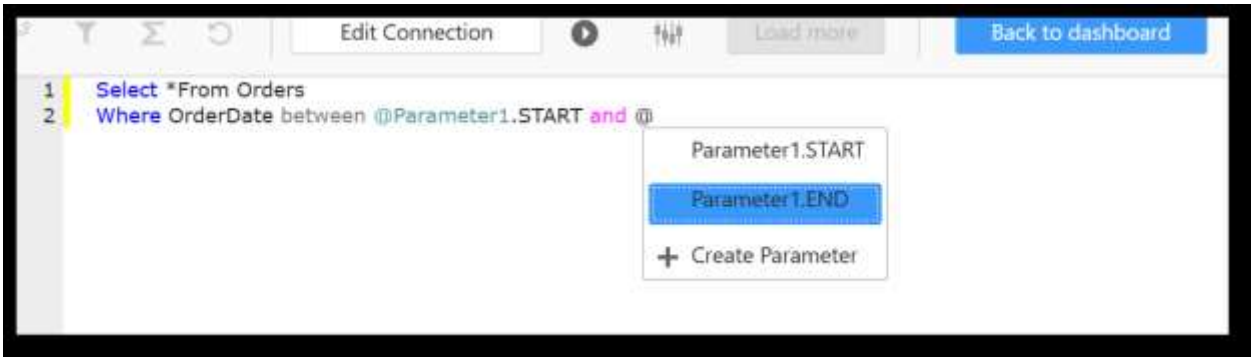


You can use Date range parameter in Stored procedure as follows.

Query view

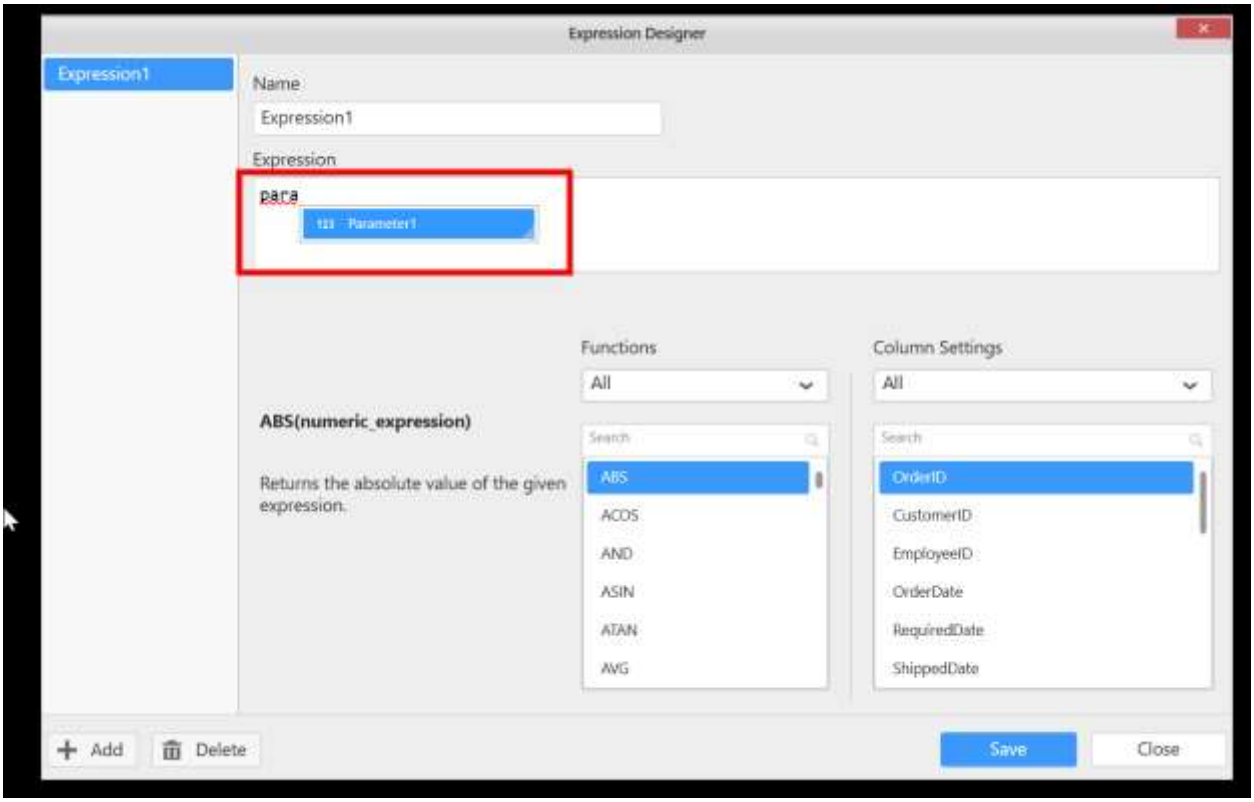
The parameters can be used in the query view, and the number of available parameters will be listed in the query editor when you press the @ key.

You can use Date range parameter in Query view as follows.

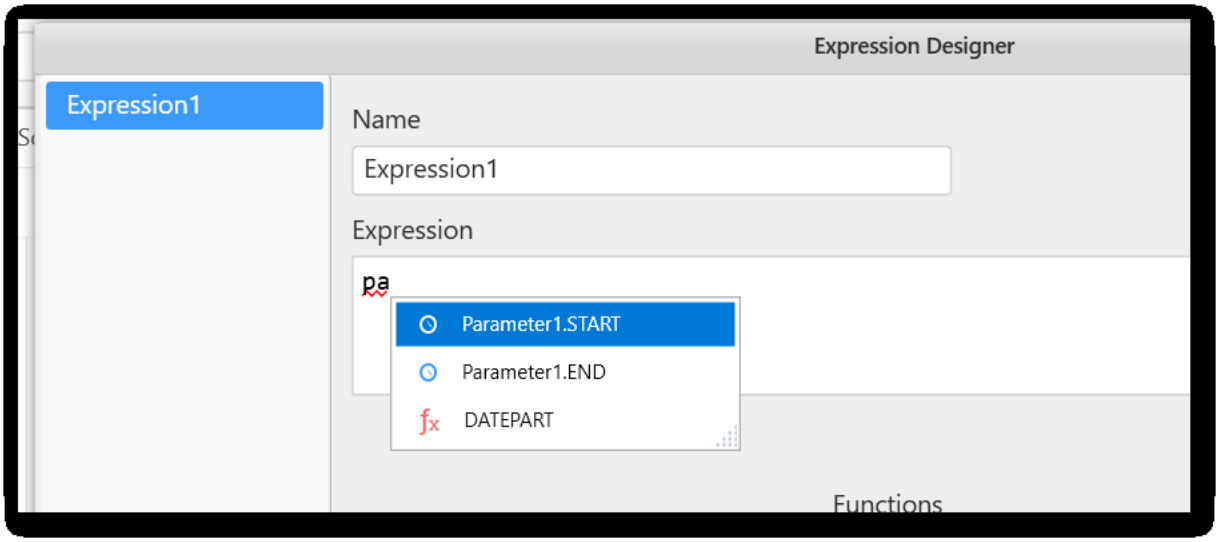


Expression editor

The dashboard parameters can be used in the Expression editor.

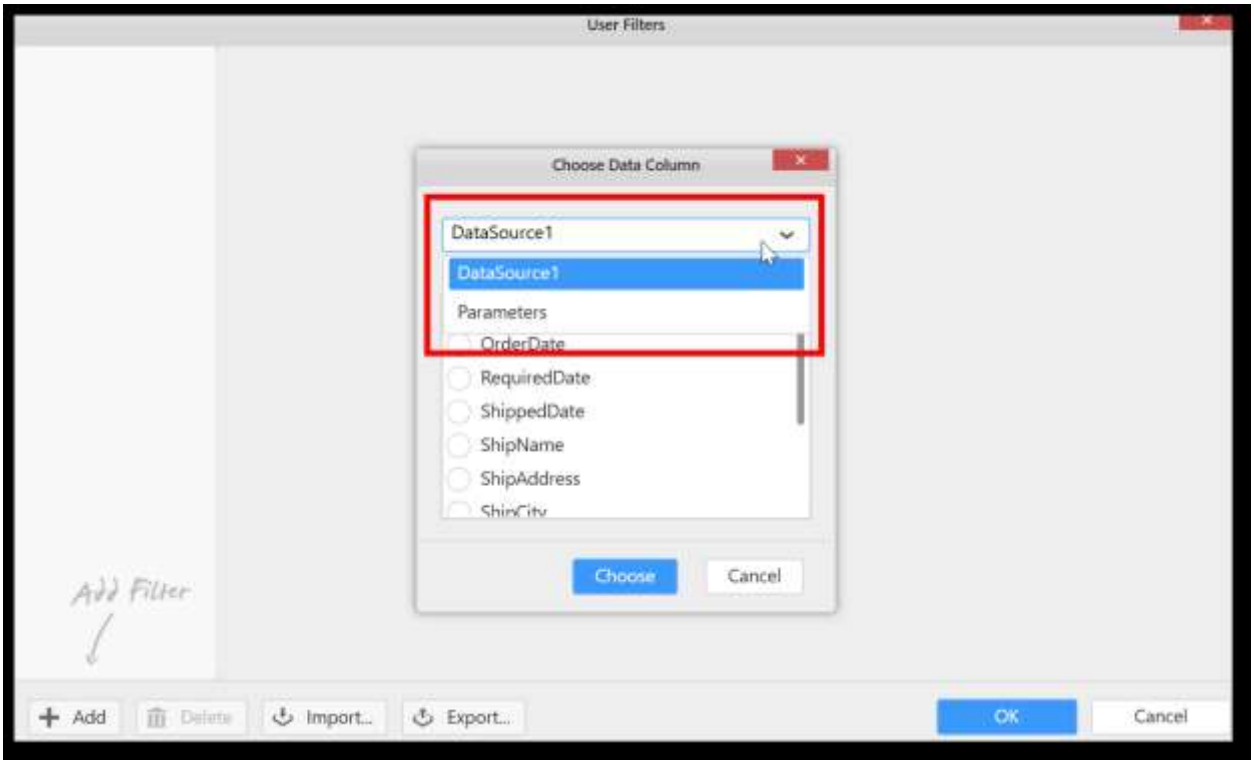


You can use Date range parameter in Expression Editor as follows.

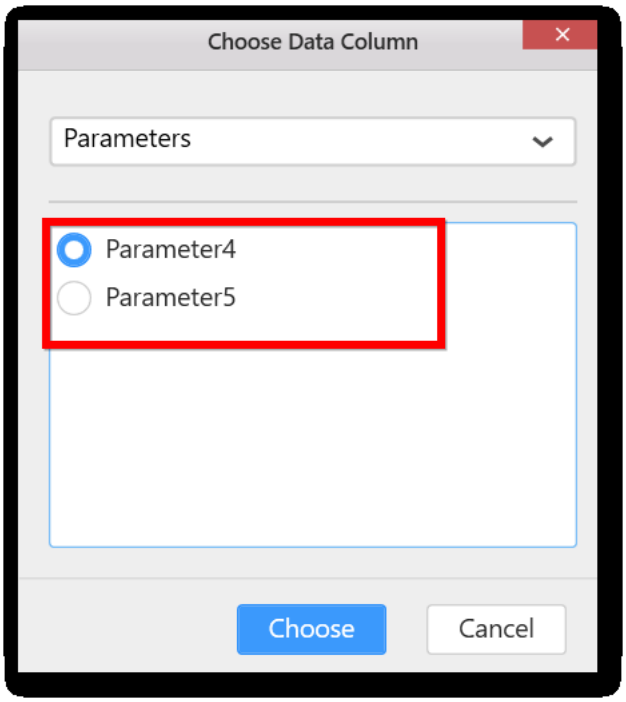


User based filters

The parameters option will be listed with the added data sources in the Choose Data Column window.



While clicking the parameters item in the drop-down list, the list mode and data-column-mode parameters can be used in the user-based filters.



Troubleshooting

Passing a comma-delimited parameter to a stored procedure

To create a stored procedure, follow these steps:

Step 1

Create a database as follows.

```
CREATE DATABASE TestParserLogic

GO

USE TestParserLogic

GO
```

Create a parser counter table. This table is used for parsing the comma-delimited parameter.

```
CREATE TABLE tableToolsStringParserCounter

(

ID INT

)
```

You have to populate the table once, when the table is not dropped from the database. For example: In the following, a loop is used from 1 to 1000 and each value is inserted during the looping process.

```
DECLARE @i INT

SELECT  @i = 1

WHILE (@i <= 1000)
```



```
BEGIN
INSERT INTO tableToolsStringParserCounter SELECT @i
SELECT @i = @i + 1
END
GO
```

Step 2

Here is the query that parses the comma-delimited string in the rows:

```
SELECT  Convert(Int, NullIf(SubString(',' + @IDs + ',' ,
ID , CharIndex(',' , ',' + @IDs + ',' , ID) - ID) , '')) AS IDList
FROM    tableToolsStringParserCounter
WHERE   ID <= Len(',' + @IDs + ',') AND
SubString(',' + @IDs + ',' , ID - 1, 1) = ','
AND     CharIndex(',' , ',' + @IDs + ',' , ID) - ID > 0
```

The results of the statement, resemble the output from executing a single column select statement. Now, you can use the given query.

```
DECLARE @IDs var char(100)
SELECT  @IDs = '429,446,552,1001, 332 , 471'
SELECT  Convert(Int, NullIf(SubString(',' + @IDs +
',' , ID , CharIndex(',' , ',' + @IDs + ',' , ID) -
ID) , '')) AS IDList
FROM    tableToolsStringParserCounter
WHERE   ID <= Len(',' + @IDs + ',') AND SubString(',' +
@IDs + ',' , ID - 1, 1) = ','
AND     CharIndex(',' , ',' + @IDs + ',' , ID) - ID > 0
```

Step 3

You can make some test data up by creating tables (TABLE tableCity and TABLE tableSalesman), and then populating them with data.

```
CREATE TABLE tableCity
(    CityID Int IDENTITY (1, 1) NOT NULL,
City var char(12) NOT NULL
)
GO
INSERT INTO tableCity (City) VALUES ('Houston')
INSERT INTO tableCity (City) VALUES ('New Orleans')
```



```
INSERT INTO tableCity (City) VALUES ('Atlanta')
INSERT INTO tableCity (City) VALUES ('Orlando')
CREATE TABLE tableSalesman
(    SalesmanID Int IDENTITY (1, 1) NOT NULL,
SalesmanName var char(10) NOT NULL,
CityID Int NOT NULL,
)
GO
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('George', 1)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Mark', 2)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Greg', 3)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Susie', 4)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Kevin', 3)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Bobby', 1)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Terry', 1)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Betty', 2)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Carl', 2)
INSERT INTO tableSalesman (SalesmanName, CityID) VALUES ('Gary', 4)
```

Step 4

Create a procedure to query all the cities from a comma-delimited list of SalesmanIDs. Within the procedure, you should create a temp table `#1` to hold the IDs that are passed to the procedure.

Remember that users want to know the city of their each selected salesman.

```
CREATE PROCEDURE storedProcedure_CityBySalesman
(
@IDs as var char(100) --SalesmanIDs
)
AS
CREATE TABLE #1
(
IDList Int
)
-- Optional index on the temp table
CREATE INDEX idx1 ON #1 (IDList)
INSERT INTO #1
```



```
SELECT  Convert(Int, NullIf(SubString(',' + @IDs + ',' , ID ,
CharIndex(',' , ',' + @IDs + ',' , ID) - ID) , '')) AS IDList
FROM    tableToolsStringParserCounter
WHERE   ID <= Len(',' + @IDs + ',') AND SubString(',' +
@IDs + ',' , ID - 1, 1) = ','
AND     CharIndex(',' , ',' + @IDs + ',' , ID) - ID > 0
```

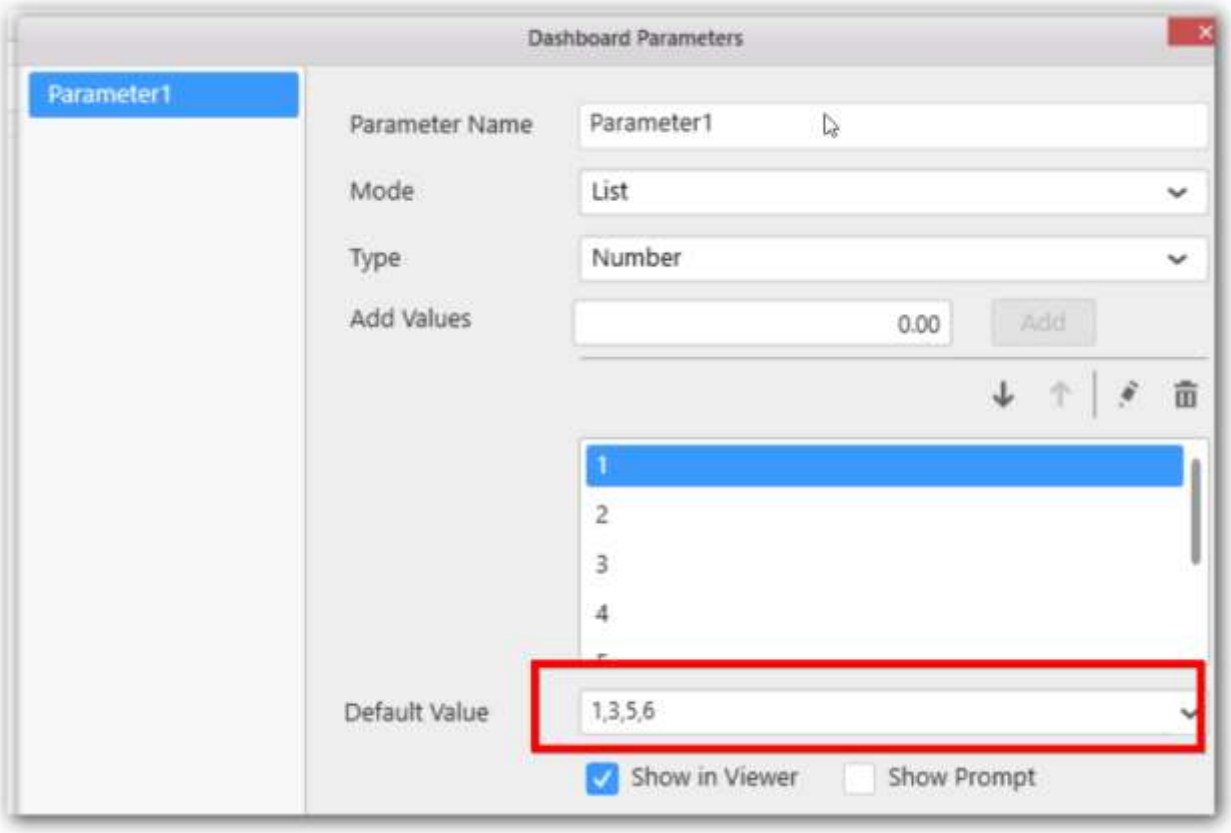
Note: Now, use the temp table #1 containing the list of SalesmanIDs that are passed by the parameter @IDs to obtain all Salesman's cities respectively in the final select statement of the procedure.

```
SELECT  SalesmanName, City
FROM  tableSalesman s,
1 t,
tableCity c
WHERE  s.CityID = c.CityID
AND  t.IDList = s.SalesmanID
GO
```

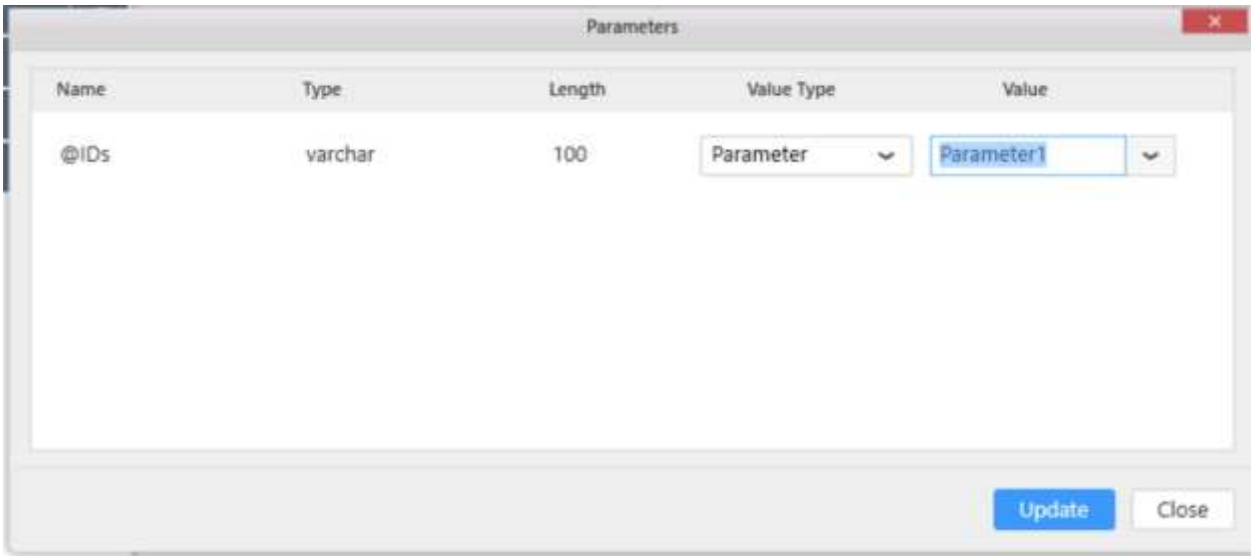
Step 5

Execute the stored procedure in the Dashboard Designer and view the results.

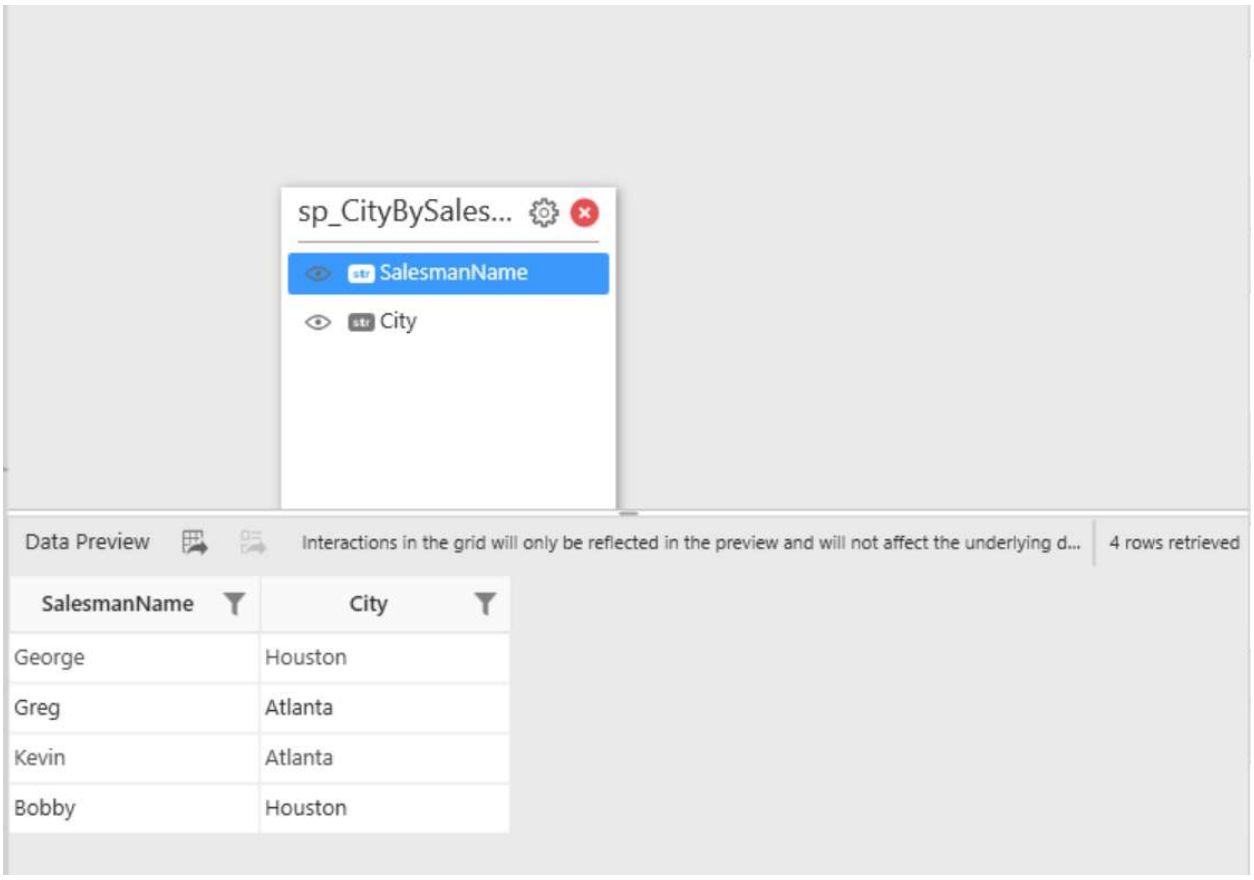
Add the multiple values in the list.



Pass this parameter to the stored procedure as follows.



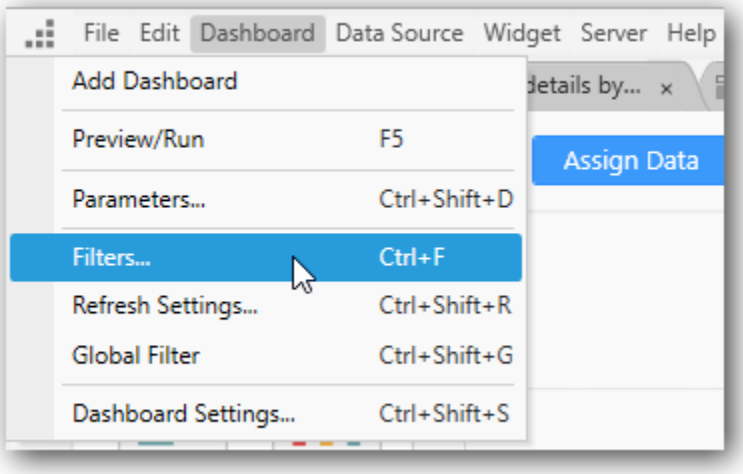
Results:

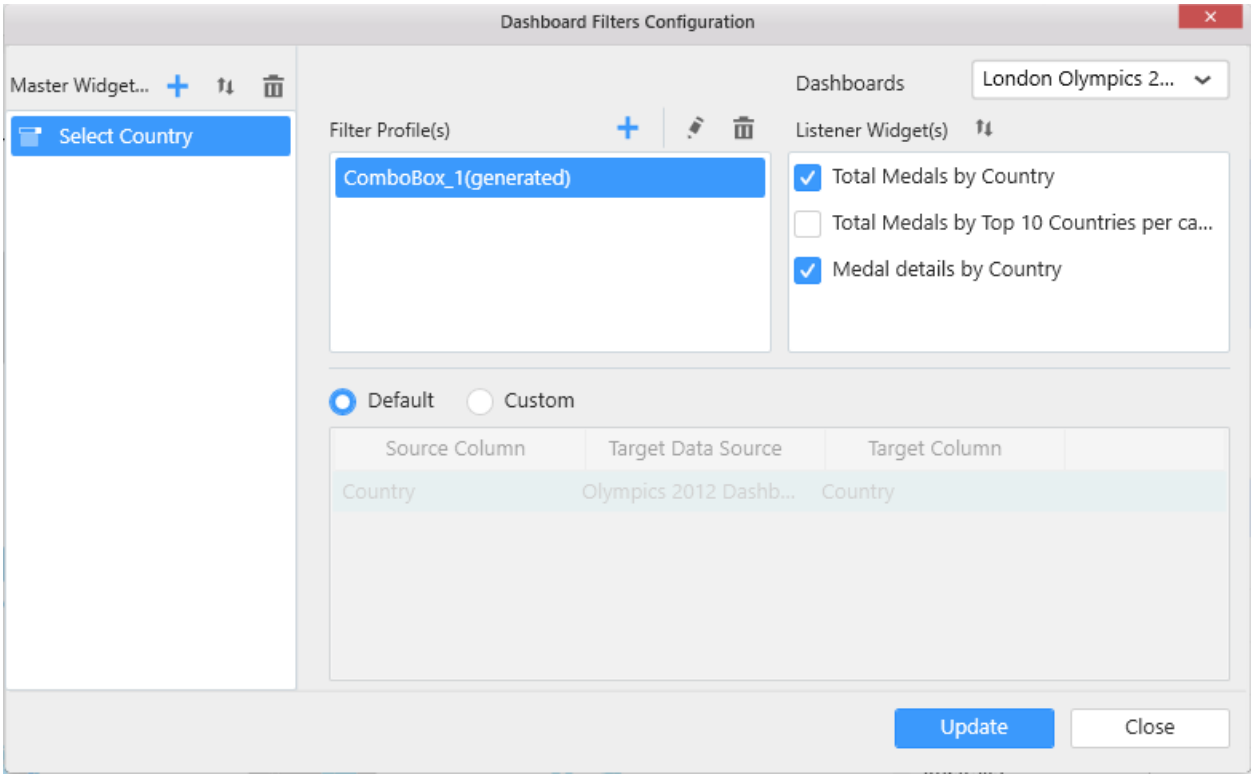


Configuring Dashboard Filters

Dashboard Filters allows you to control the interdependency of widgets in a dashboard w.r.t dynamic user interactions.

You can configure the dashboard filters through the Dashboard Filters Configuration window launched by selecting the Filters... menu item in Dashboard main menu.



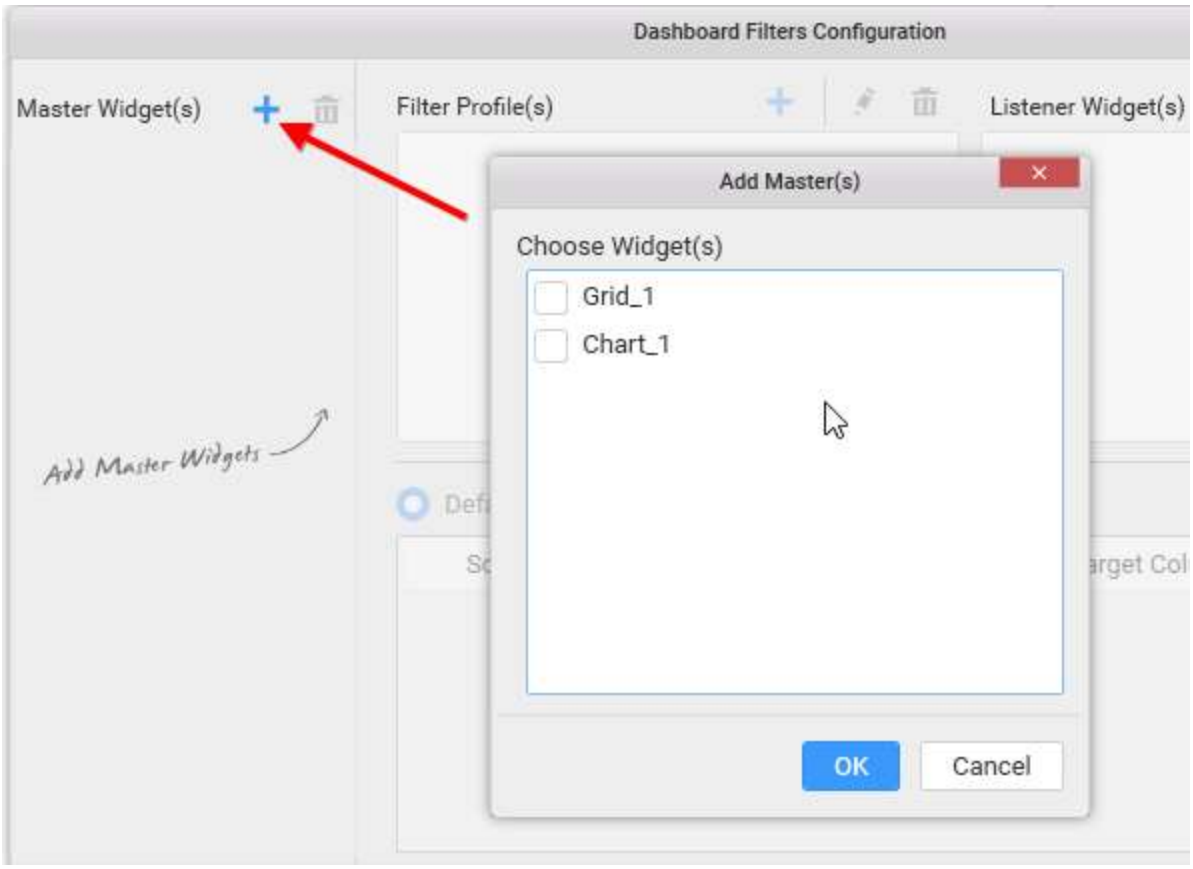


Master Widget(s)

In this window, the Master Widget(s) section holds the names of widgets whose Act as Master Widget property setting was enabled. If the widget is added under this section, it is subjected to have its filter effect on user interaction with it, reacted with the pending data updates over the widgets that were marked as listeners to it through the Listener Widget(s) section.

The master widget(s) data will not be refreshed with the pending data updates unless the Auto Refresh is configured. To configure the Auto Refresh in the Dashboard Designer, refer here. To enable the Auto Refresh in the Dashboard Platform SDK, refer here.

You can also add a widget into this section explicitly through Add option in the header of this section.



Selecting a widget in this section, will show its associated filter profiles and listener widgets in respective sections, which are customizable.

You can remove a selected widget from the Master Widget(s) section through the Remove option in its header.

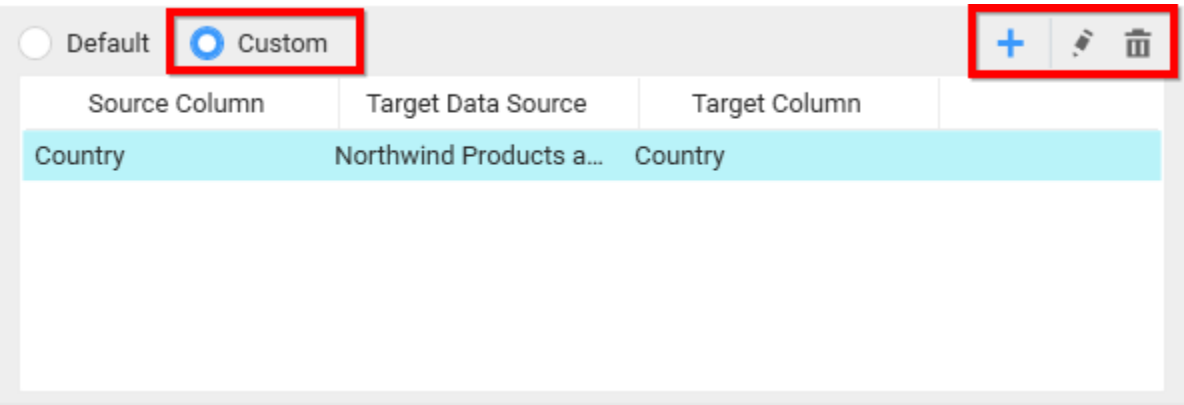
Note: Filter type widgets will get added automatically in this section for user convenience, by default. You can remove it, if not required.

Filter Profile(s)

Filter Profile(s) section holds a default profile generated automatically for one widget added in Master Widget(s) section. This profile holds the detail about the filter criteria and listener widget(s) to be affected based on that criteria.

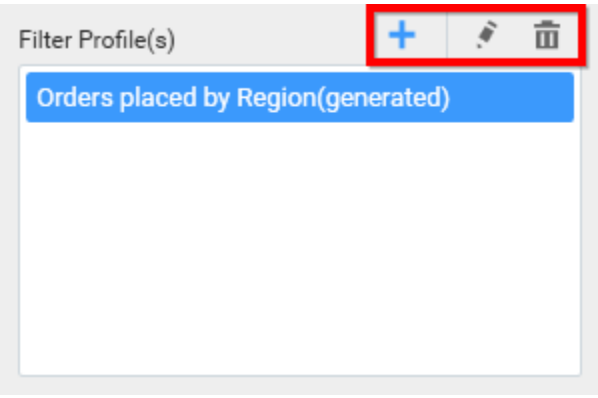
Filter criteria can be set through the bottom pane configuration.

This pane holds the detail about the mapping of a column in current data source with the one in target data source, which is same by default, based on which the user interaction filtering works. You can also customize this default filter criteria through switching to Custom option, like Add, Edit and Remove.



Note: Refer the detailed steps to achieve the cross data source filtering from this Knowledge base article.

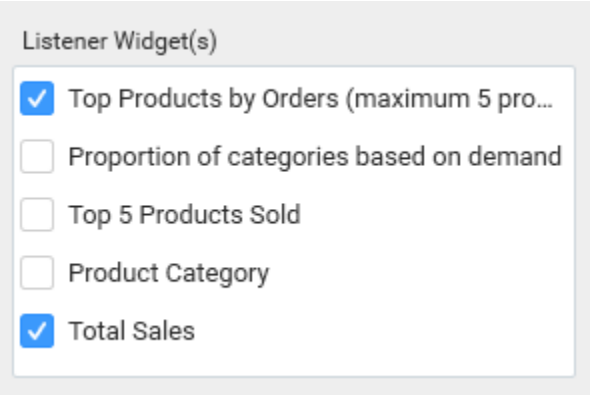
You can modify the default profile setting or remove the same or add a new profile through respective options in its header.



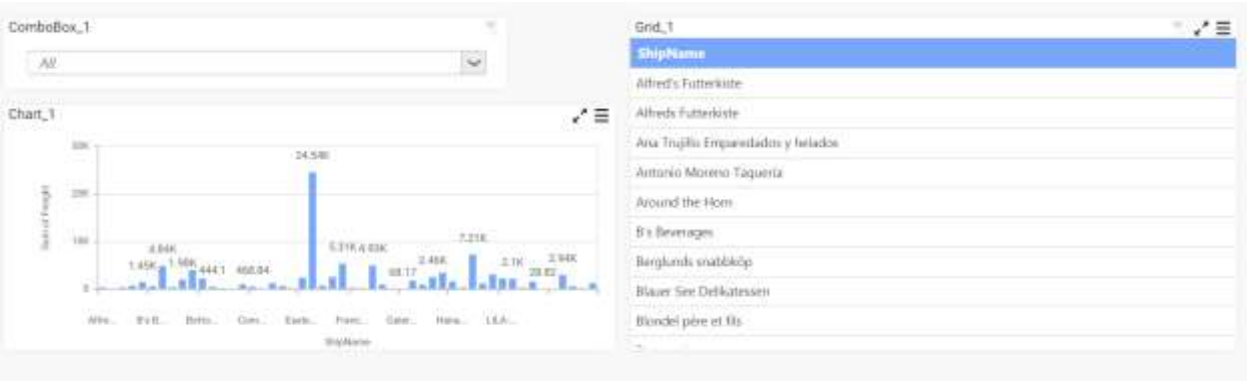
Note: Having more than one filter profile defined for a master widget will have the interaction effect reflected in such a way that, only records that satisfies all those filter criteria defined under those filter profiles will get filtered.

Listener Widget(s)

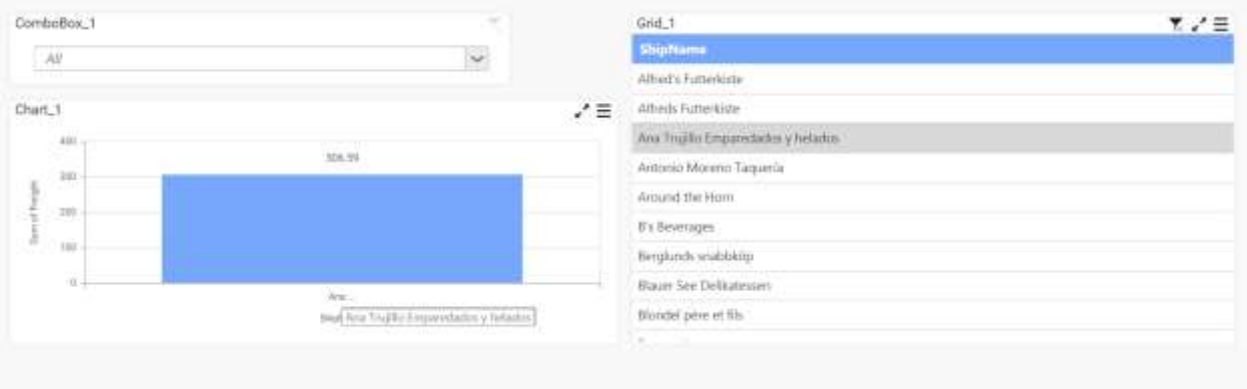
Listener Widget(s) section holds the name list of widgets other than the master widget it was associated with. Select the checkbox near to the respective widget name so as to map it to the respective master widget under the specified filter profile to respond to user interaction in master widget and unselect the one which you don't want to respond to.



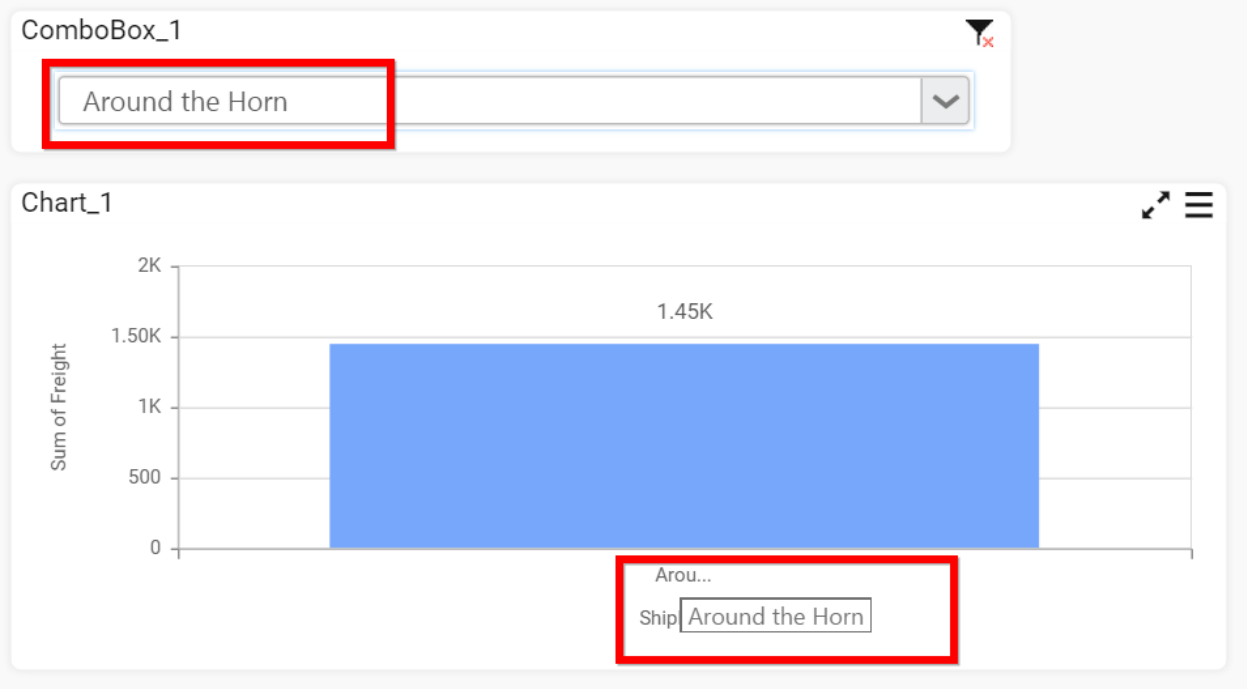
For example, Consider Chart*1 widget is marked as a listener widget to the Grid*1 and ComboBox*1 widgets, Grid*1 widget is marked as listener widget to the ComboBox_1 widget.



While applying filter in the grid widget, the chart widget get filtered based on the filter applied in the grid widget. Now, the chart widget contain the information about the selected ship name in the grid.



Now, on selecting the data in the combo box, the chart widget shows the particular detail.



Dashboard Filter Panel

Filter panel allows you to view the filter elements in a dedicated area of dashboard viewer instead of showing in the visualization region.Refer here for more details on filter panel.

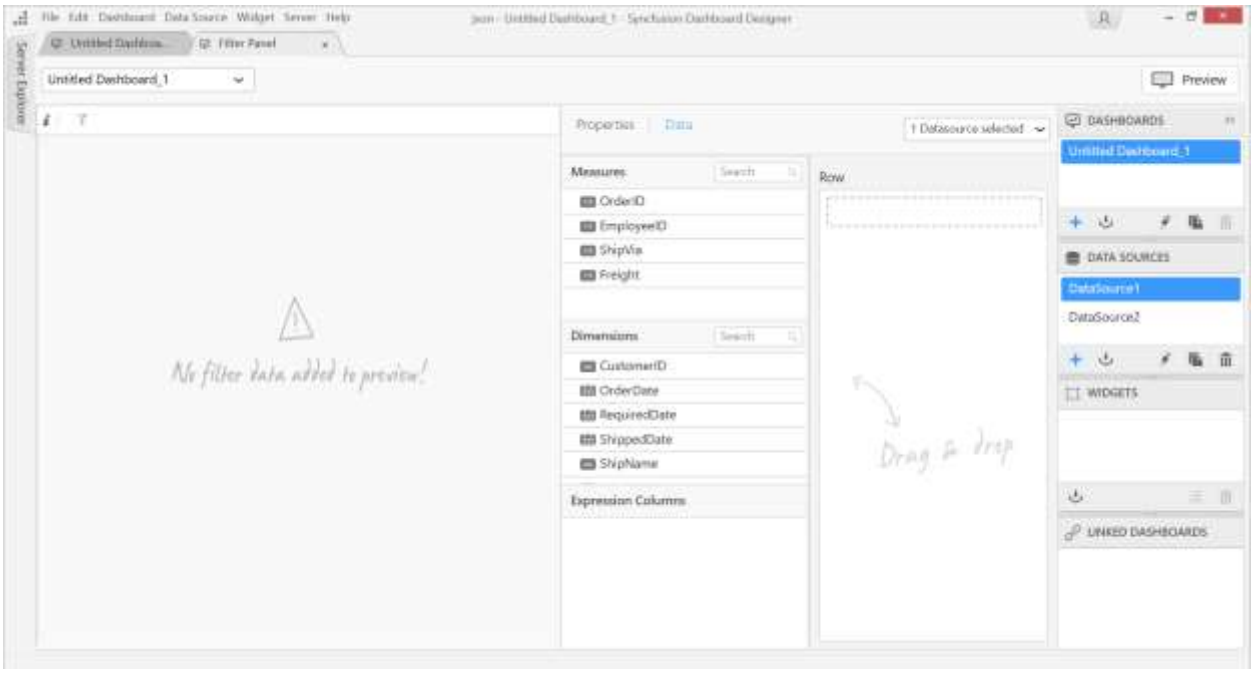
Dashboard Filter Panel

Filter panel allows you to view the filter elements in a dedicated area of Dashboard Viewer instead of showing in the visualization region.

How to open the filter panel tab

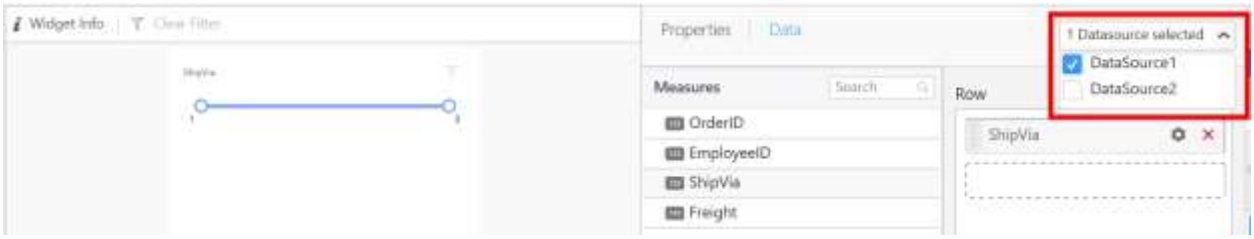
Each dashboard tab has a single filter configuration panel, you can open it by clicking Configure Filter Panel in the designer toolbar.

Filter panel tab



Configuring and formatting filter widgets in the filter panel

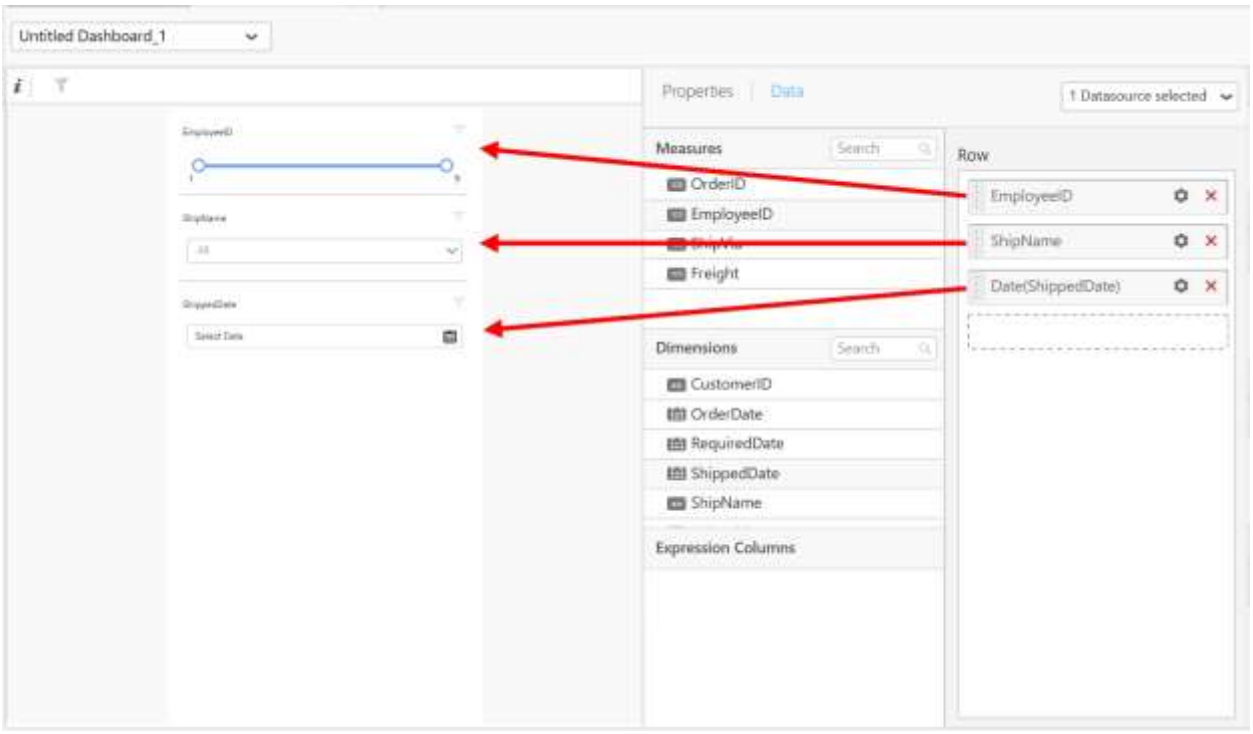
You can use multiple data sources by using the Multi selection data source option in the filter panel.



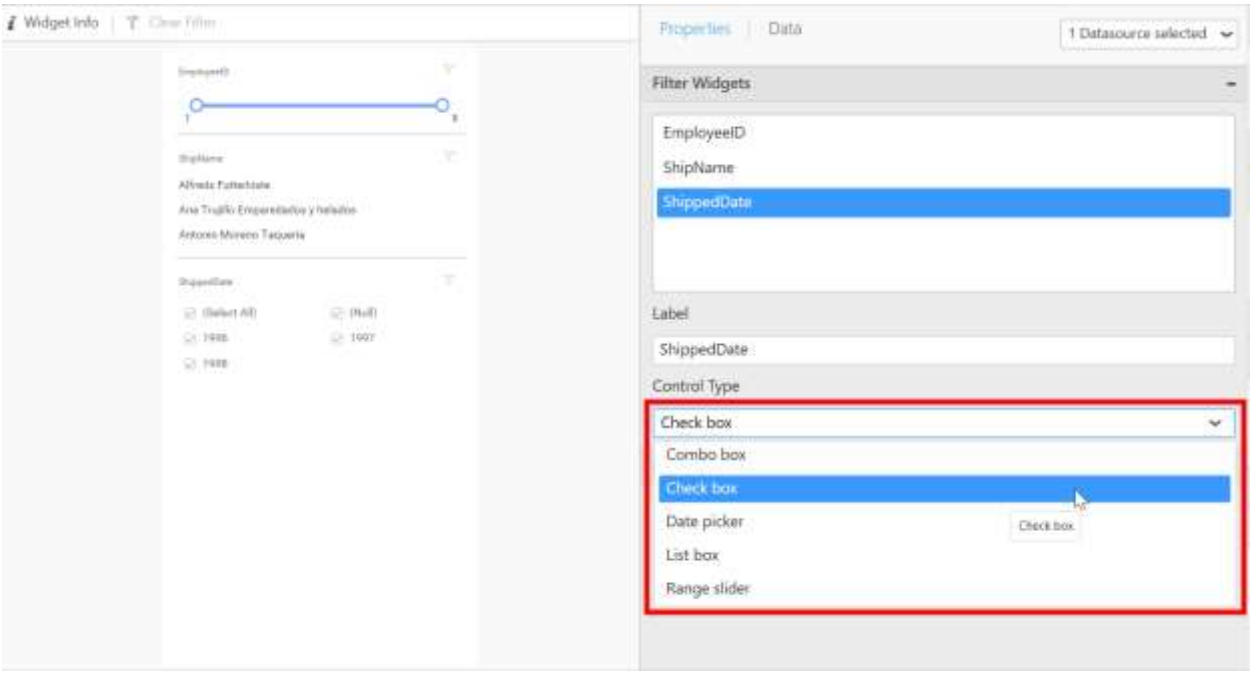
You can add countless filter widgets in the filter panel, and the filter widget will be predicated by the field dropped in the row section, by default.

For example,

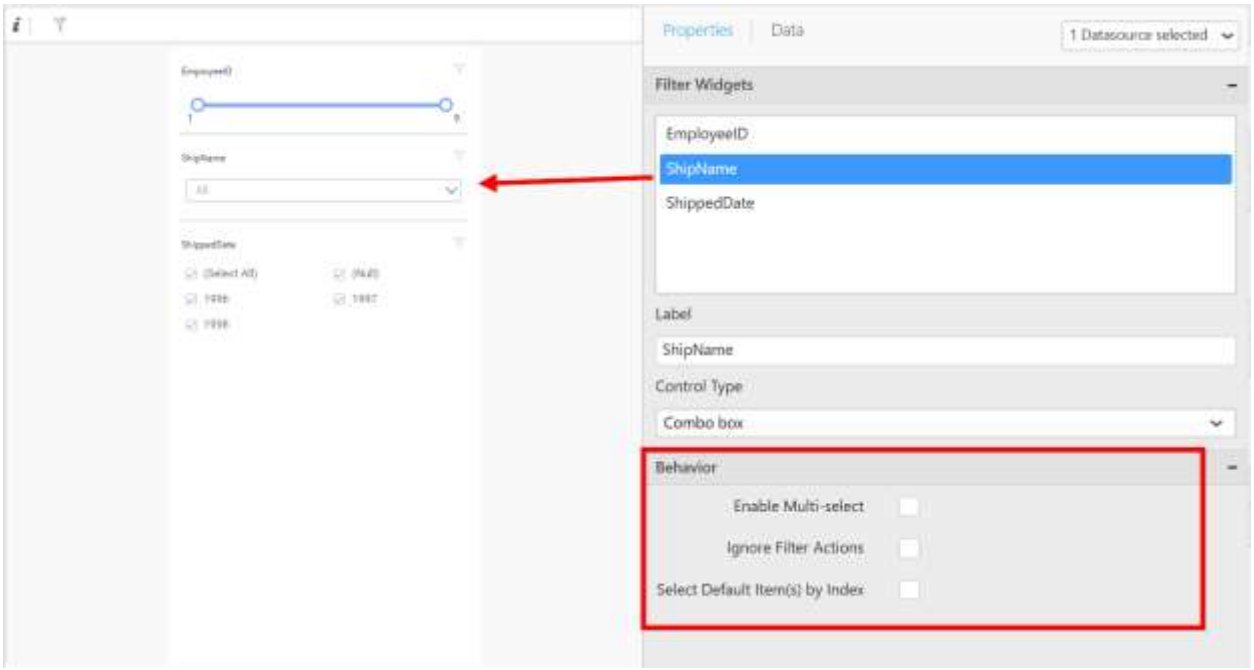
Numeric – Range slider String – Combo box *Date – Date picker* Boolean – Check box



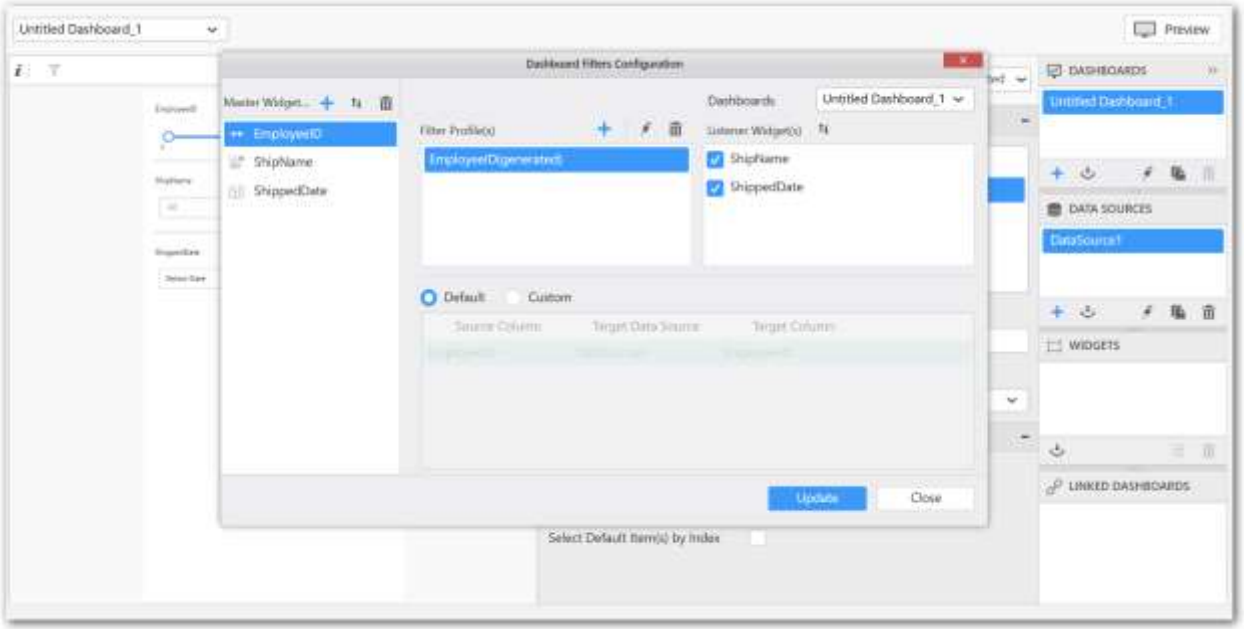
You can change the default type by using control type (possible type applicable for the selected field) option provided in the Properties tab.



The behavior section of the properties tab list out the individual properties of selected filter widget. Here, the list box item is selected, so the behavior section shows the properties of List box.

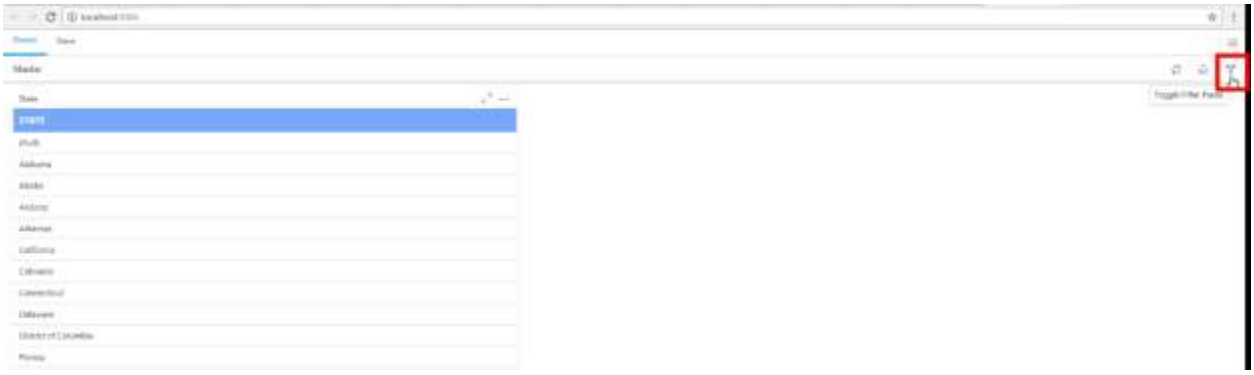


These filter widgets will be listed in Filter Action window setting to set the master slave relation.

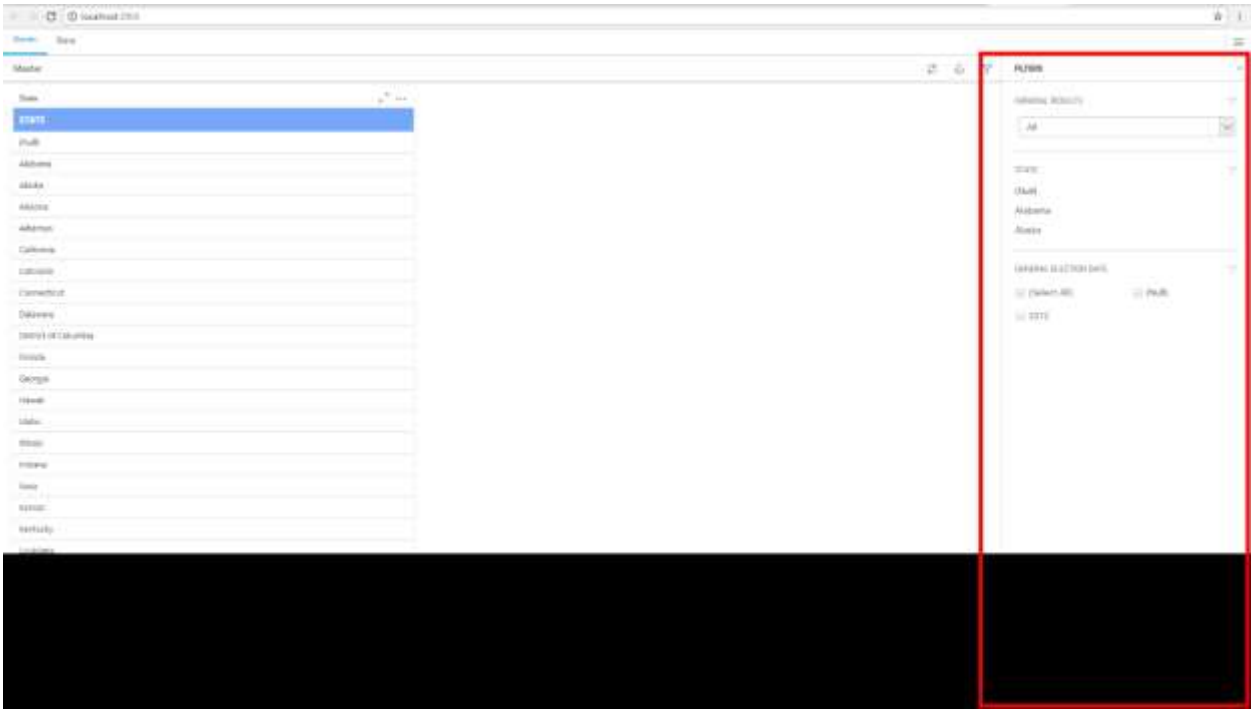


How to open the filter panel in viewer

Preview the dashboard report. After loading the report, click the icon at the right-top corner of the Dashboard Viewer.



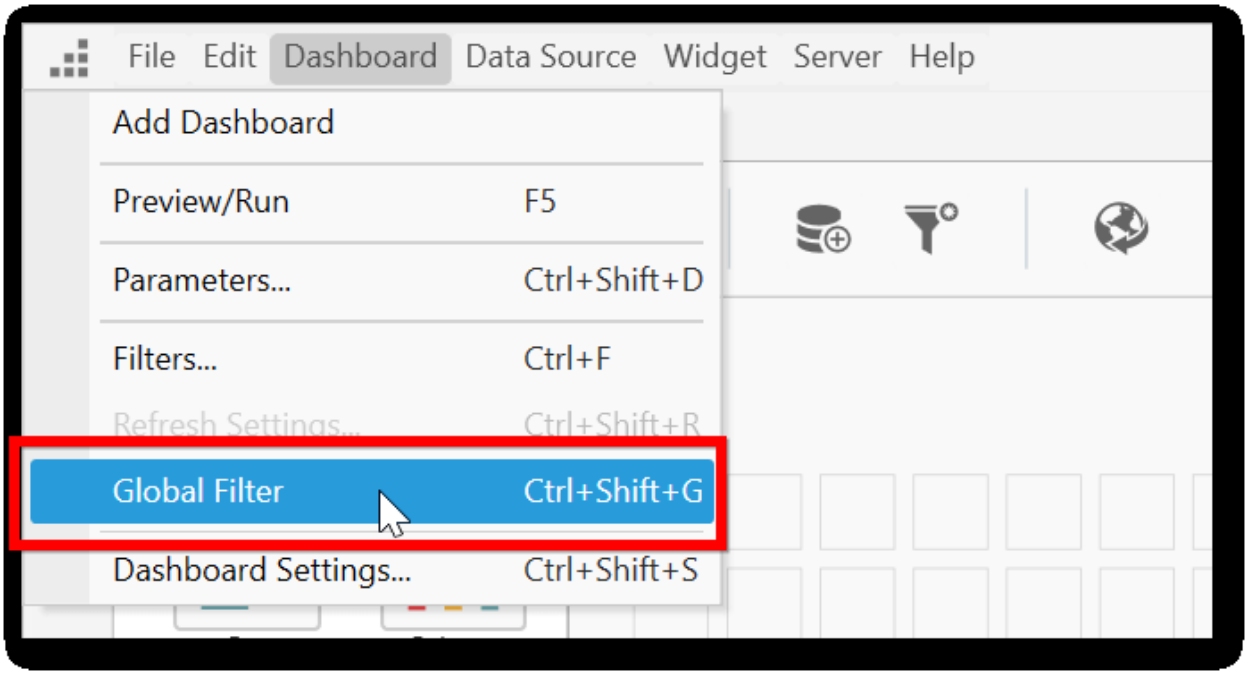
It will show the filter panel at the Dashboard Viewer. The dashboard will reflect based on the action on filter widgets.



Global filter panel

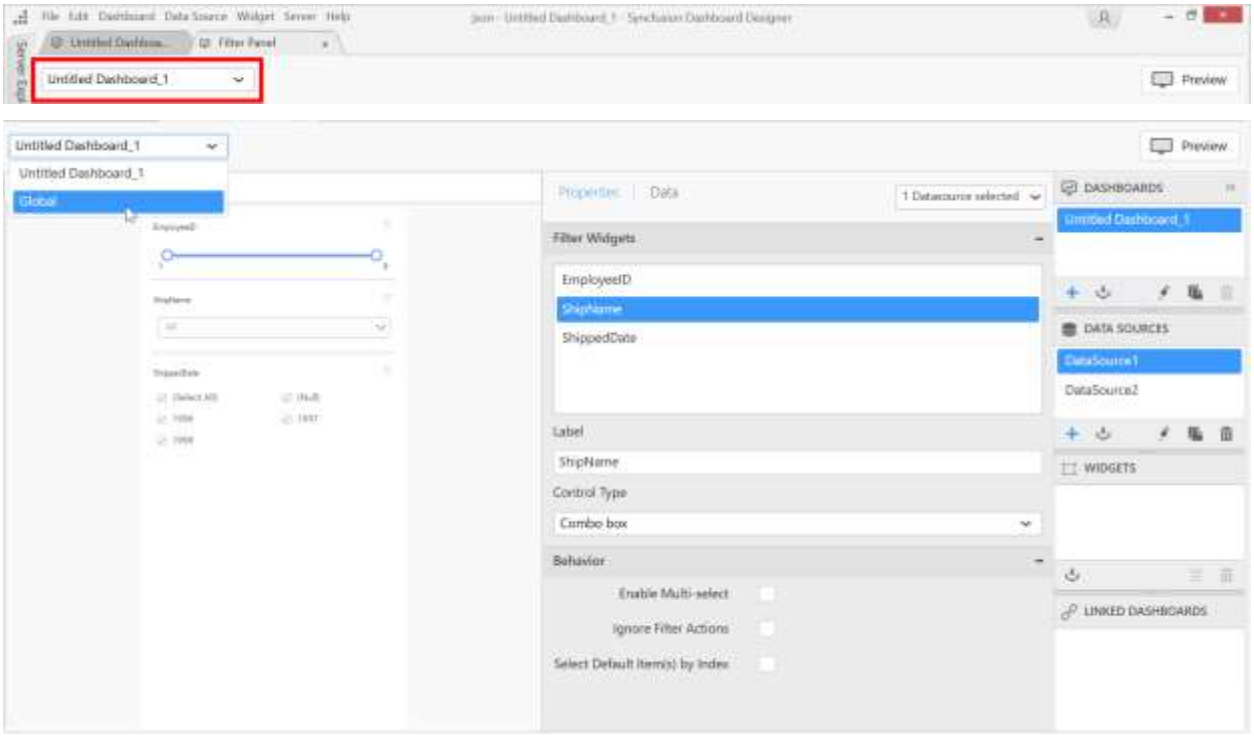
Global filter panel has functionalities like normal dashboard tab filter panels, but it will be visible in all the dashboard tabs and filters all the widgets in all dashboard tabs.

To open the global filter panel, click the Dashboard menu and select the Global Filter. You can also use the keyboard shortcut Ctrl + Shift + G.



How to configure the global filters

Click the combo box shown in the following screenshot and select Global.

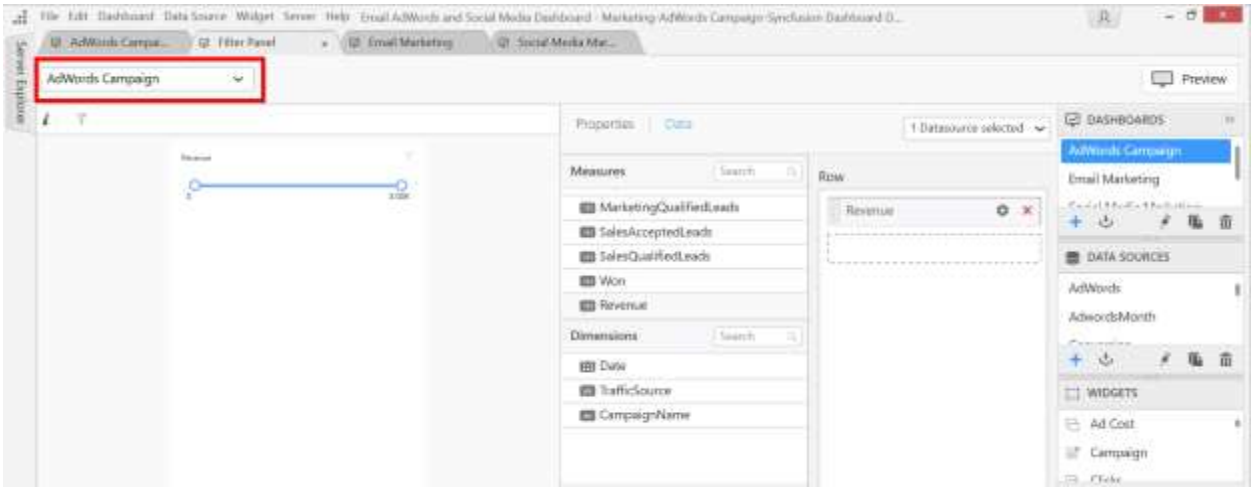


Now, you can add and format the filters widgets as you do in the normal dashboard tab filter widgets.

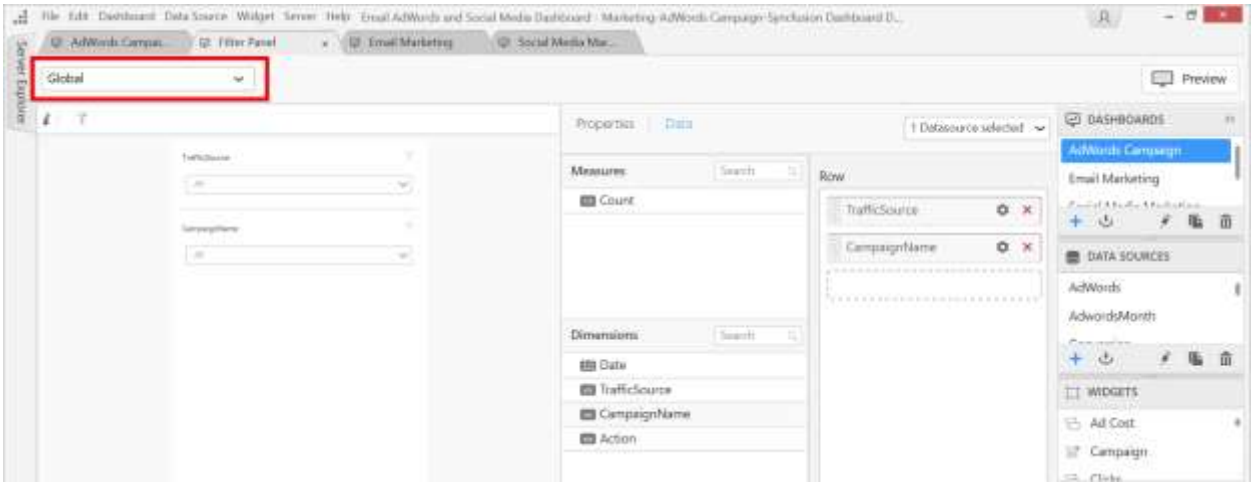
How to open global filter panel in the Dashboard Viewer

Consider the “Email AdWords and Social Media Dashboard – Marketing sample dashboard”.

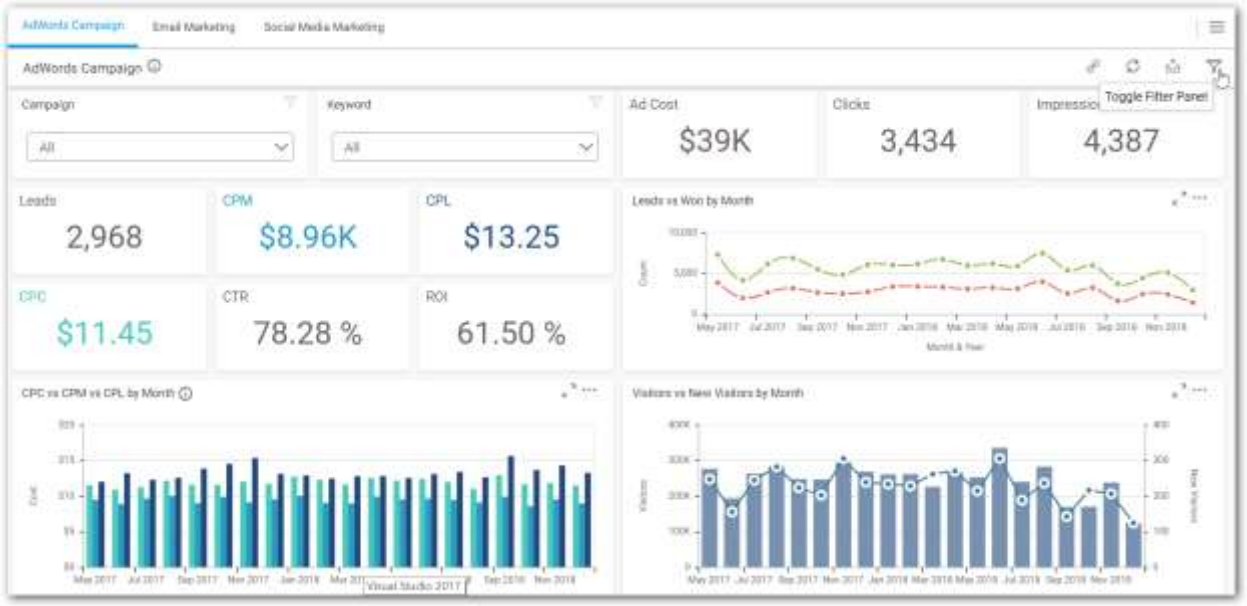
The AdWords Campaign tab filter panel is configured as shown in the following image.



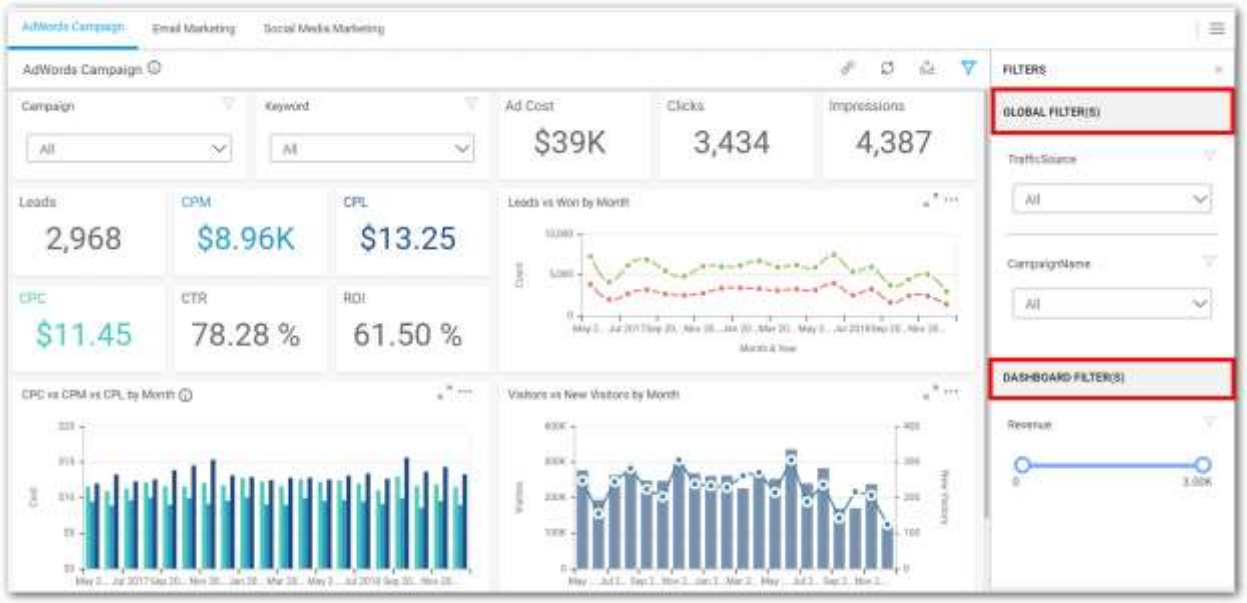
The global filter is configured with more filters as shown in the following screenshot.



Click Preview in dashboard designer and once the preview is launched click the filter panel button to open the filter panel.



The global filters and dashboard tab filters will be displayed in the filter area with the configured widgets. Refer to the following image for visualization.



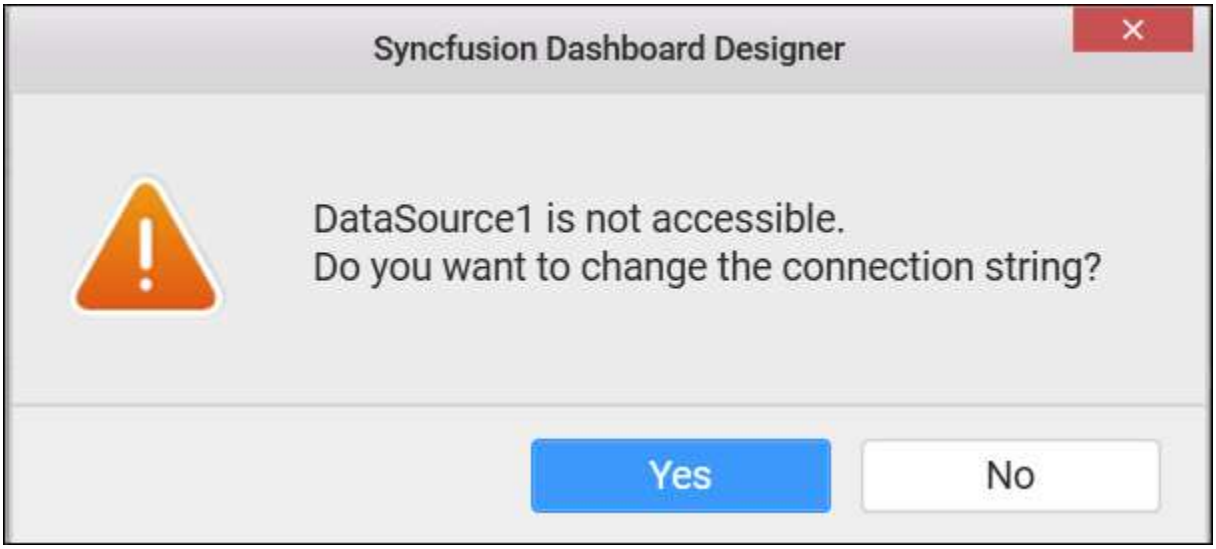
Importing the dashboard reports with global filters

If your current dashboard is configured with global filter panel and you are importing a dashboard with global filters, the following alert message will be shown to choose the merge behavior.

- **Use Existing**: Contains the existing widgets in the current global filter panel and ignores the imported global filter widgets.
- **Use Imported**: Replaces the current global filter panel by the newly imported global filter widgets.
- **Use Both**: Combines the current global filter panel and imported dashboard global filter panel.
- **Ignore Both**: Resets the widgets configured in the global filter panel.

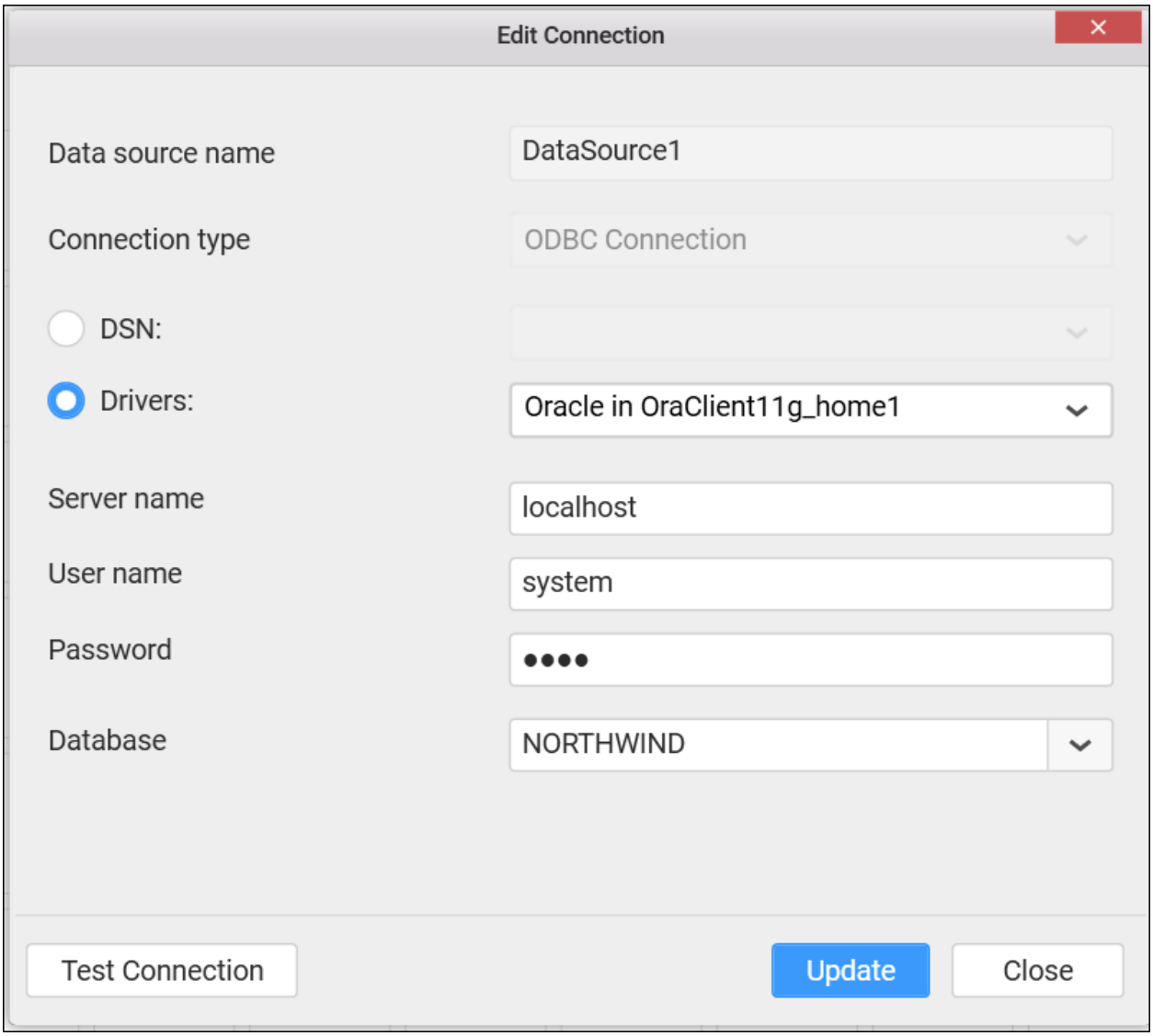
Opening or Adding an Existing Dashboard

Syncfusion Dashboard Designer now offers an ability to change the connection string and to retain the configured dashboard while opening a dashboard with invalid connection string. For example, Dashboard which is created with local SQL server credentials, now user opens that dashboard below message box gets displayed.

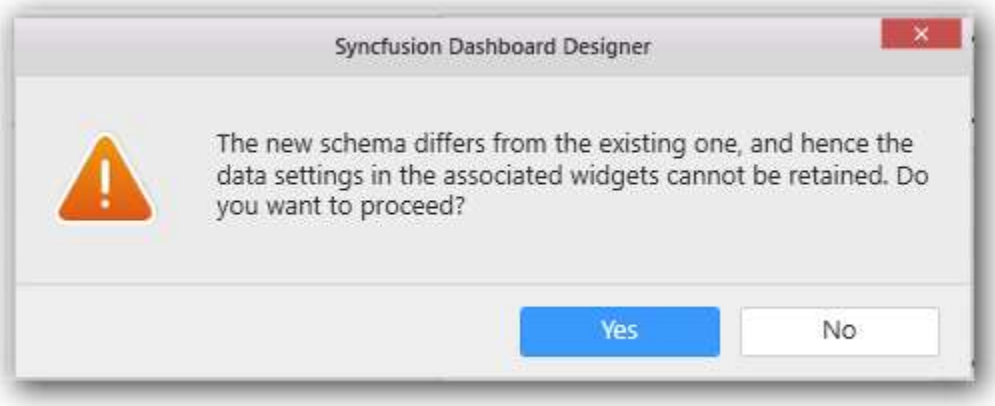


Click yes in the prompt window. Now message box will pop up connection dialog. Enter the proper connection details along with suitable credentials for data source and establish connection to the respective data source.

Click no in the prompt window. Now only the dashboard layout will appears and data will not maintained.



While click update button, if the new connection string has entirely different fields compared to older ones, it will show the below alert window.



If you click "Yes", it will do the below operations:

Removed the fields which are used in our application.

1)Expressions 2)User filters 3)Dashboard parameters 4)Linking 5)Widgets 6)Initial filters

If you click "No", can able to choose different connections.

Information: Connection will persist the dropped table(s), table relationships, data filters, data configuration to widgets, unless the schema is different from the previous data connection. i.e., the connected database should have similar schema like the database connected in sydx, which may exist in same or different location. If the connected one don't have a column that is available in previous one, connection will just ignore that column and its related settings alone and persist others. Beyond that level, connection will drop previous settings entirely.

Note: On-demand connection string feature is provided for all server types.

Configuring User Based Filter

Configuring User Filter using User Filter Window

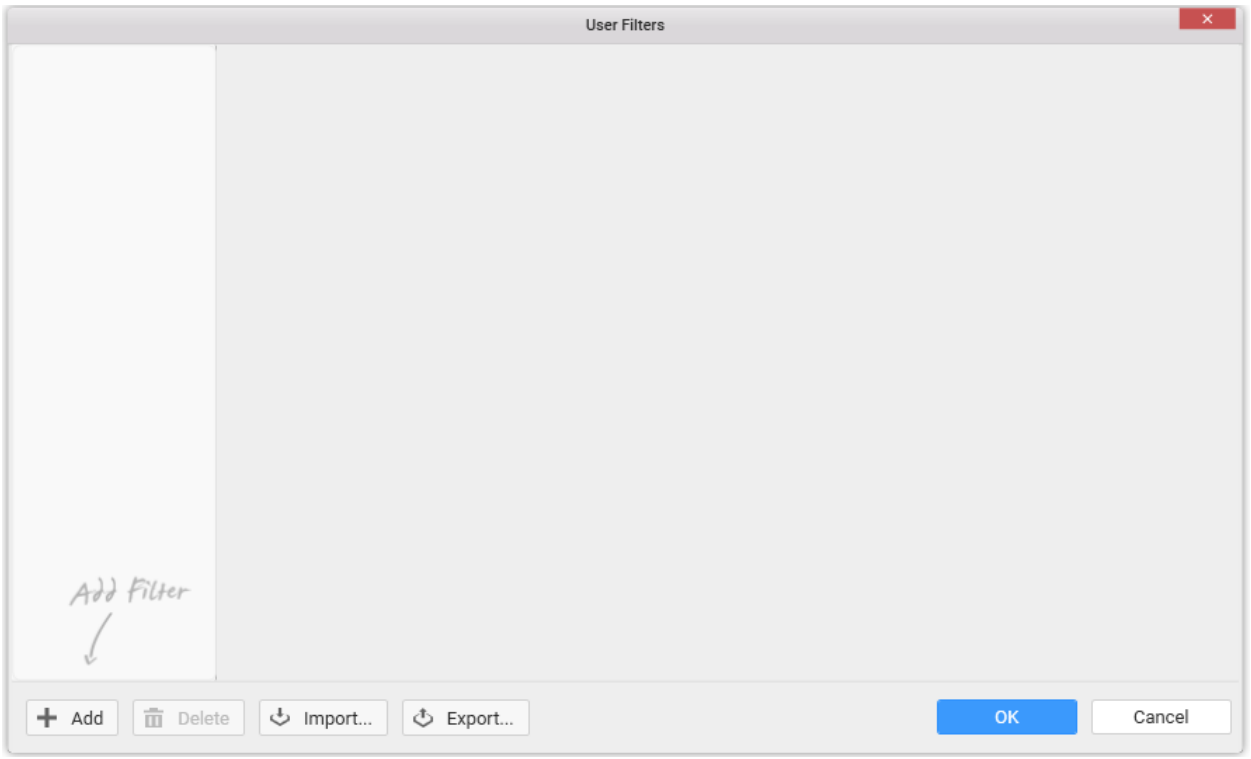
User Filter allows you to limit the data view of the published dashboard based on the logged in user. For example, in a sales dashboard that gets shared among Managers in different countries, you may want to show only the sales information of US to US Manager, and that of Europe to Europe Manager. In this case, instead of creating separate dashboards for each manager, you can make use of User based filter and define the data that will be available to each of them.

The user information will be referred from Dashboard Server user accounts. Once the dashboard is published to Dashboard Server, the view details in dashboard gets adjusted based on the logged in user.

How to Create a User Based Filter

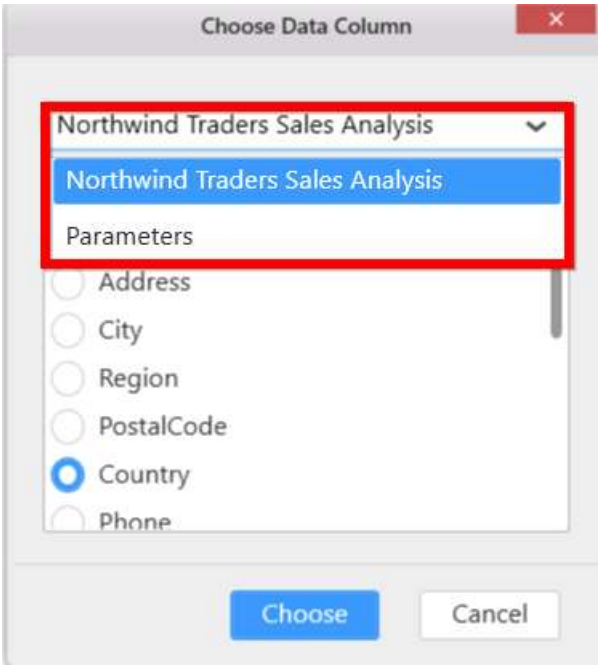
Login to Syncfusion Dashboard Server from your Syncfusion Dashboard Designer Application.

Navigate to the Server menu and select User Filter… menu item. Now, the User Filters popup window opens like below.



Click Add button at the bottom left corner to launch the Choose Data Column dialog.

Select a bounded data source or parameters in the combo box.

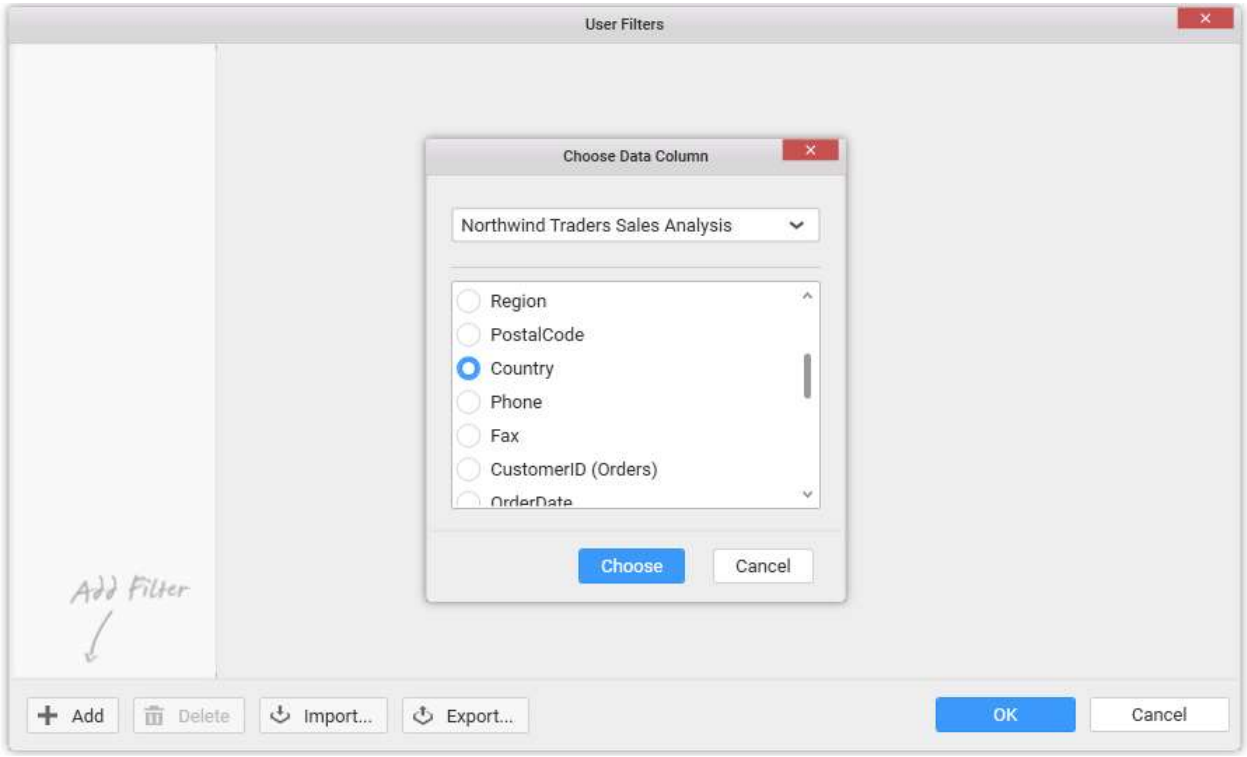


Note: The Parameters option will be visible, if parameter mode types are list and data column.

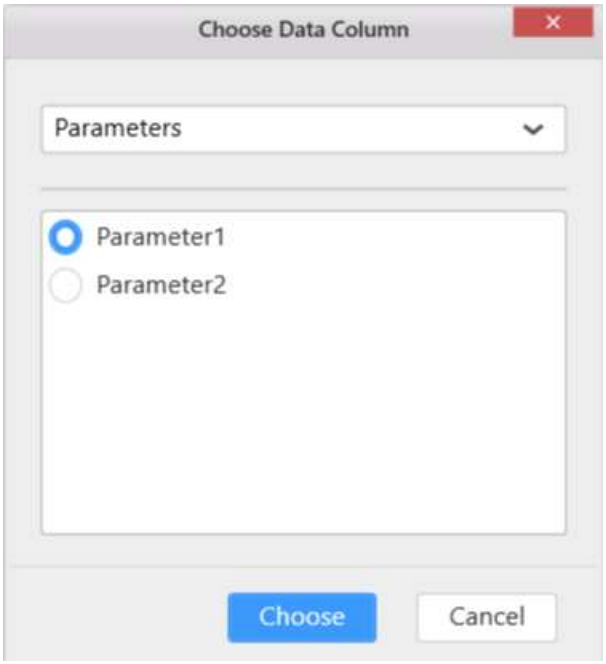
Note: When user based filter is configured with parameter, only parameter values will be shown in preview.

Information: User Filter will be applied based on the values in this field.

If data source is selected, choose a column from the list below shown.



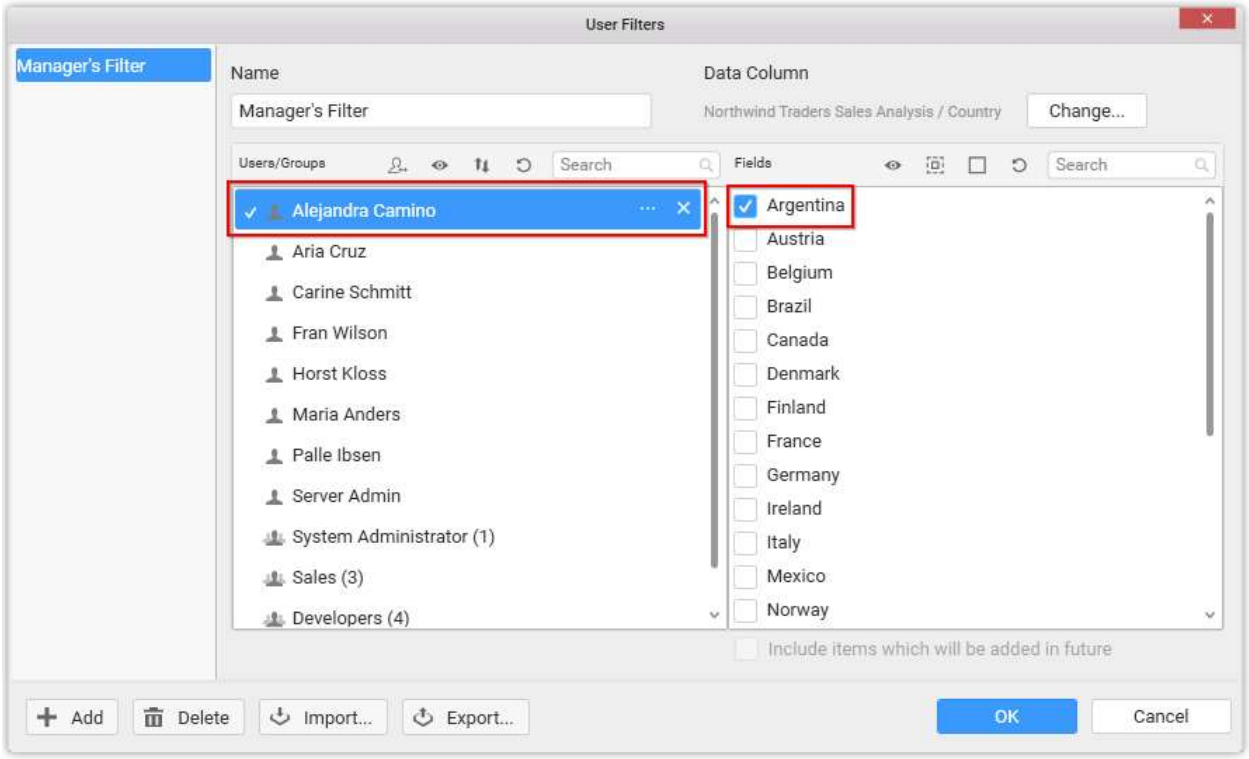
If parameters is selected, choose a parameter from the list below shown.



Now, a new user filter gets created with the users/groups registered in the Dashboard Server and the chosen field values get listed out for further configuration.

In the User Filters dialog, type in the Name field for naming the filter you created.

Select a user or group in the Users/Groups pane. Then on the right pane, select the value(s) from the fields that the selected user or group is allowed to see. Repeat the process till all users are assigned with the correct set of data.

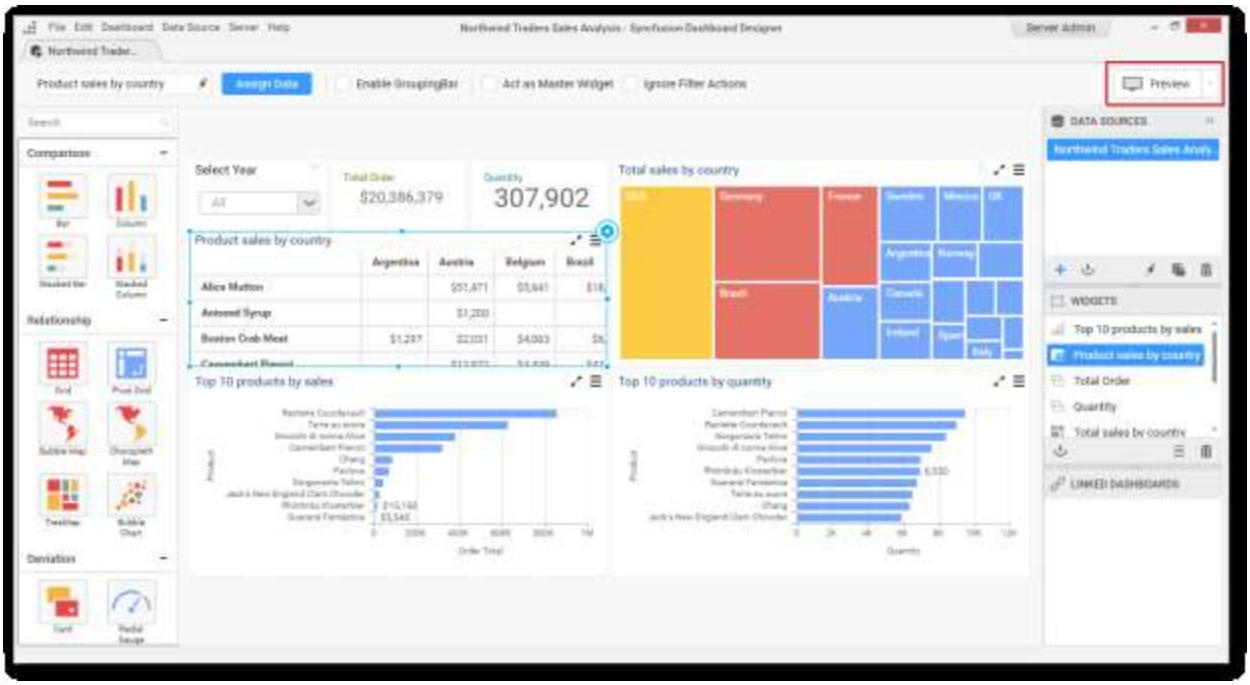


A search box is provided at top of both Users/Groups and Column values panes, so that you can narrow down the data in the dialog view if there is large number of users and groups or values in the fields, when you are looking for specific ones to configure.

Once completed, click OK.

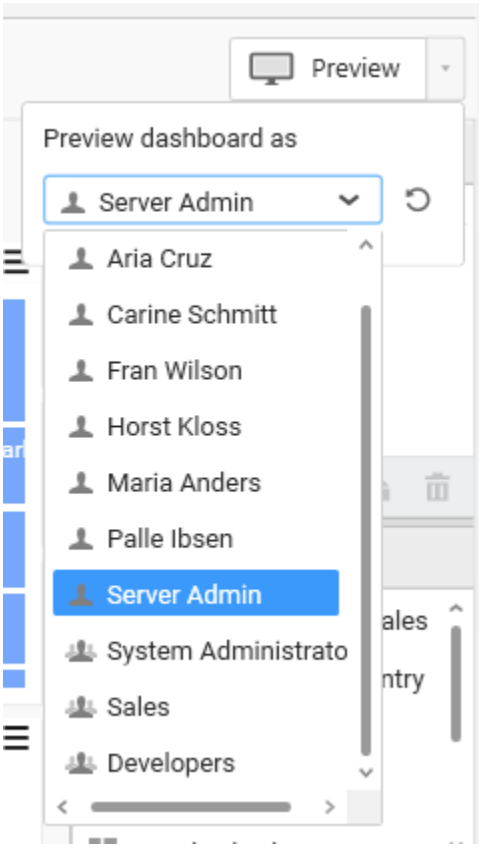
Previewing User Based Filter Configured Dashboards

When a dashboard opened in Dashboard Designer has user based filter set and is currently logged into Dashboard Server, the preview button in the toolbar will show a split button at right of it. Clicking the split button will drop down Preview dashboard as user menu with a drop down list populated with users/groups of the Dashboard Server. This menu will allow the user designing the dashboard to preview which user or group see which view once this dashboard published to respective Dashboard Server, and validate the user filters configured.



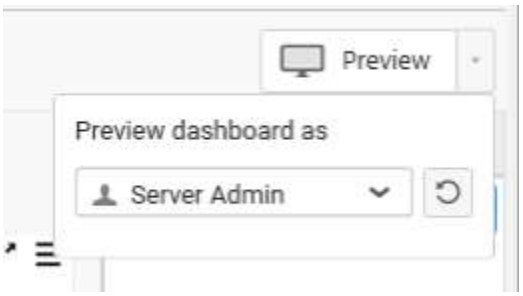
Here is the procedure to validate the configured user filters through different user/group preview.

Click the arrow next to Preview button.



Select the user or group from the drop down based on with you need to preview the dashboard.

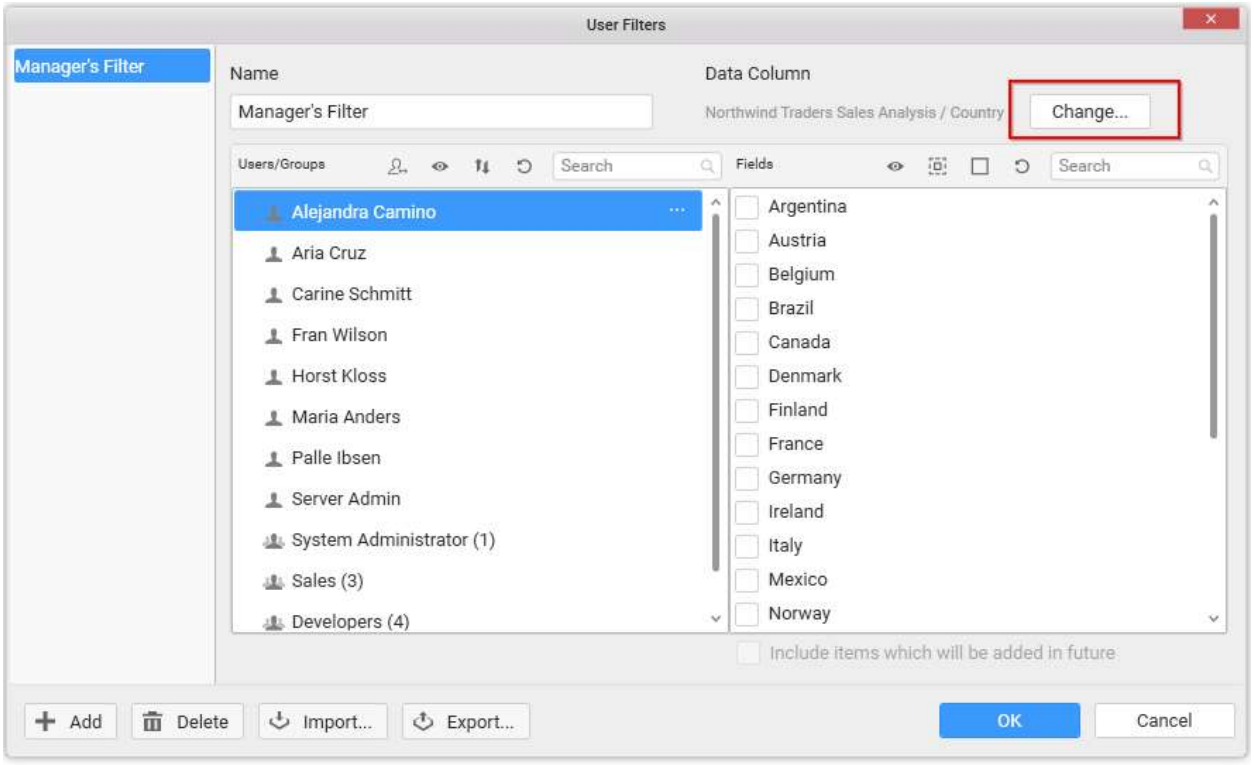
You can reload the users from the server by clicking the refresh button next to the drop down.



Information: In Public dashboards, the user based filters are not applicable and if any user filter is configured it will be ignored.

Changing the Filter Column

You can change the filter column selected while creating a user based filter, by clicking the Change… button at the top right corner which is highlighted below.



Note: Changing the column will reset the mapping already made.

Information: You can add more than one user filter through following the same procedure as discussed above, if you would like to set filter criteria by more than one column.

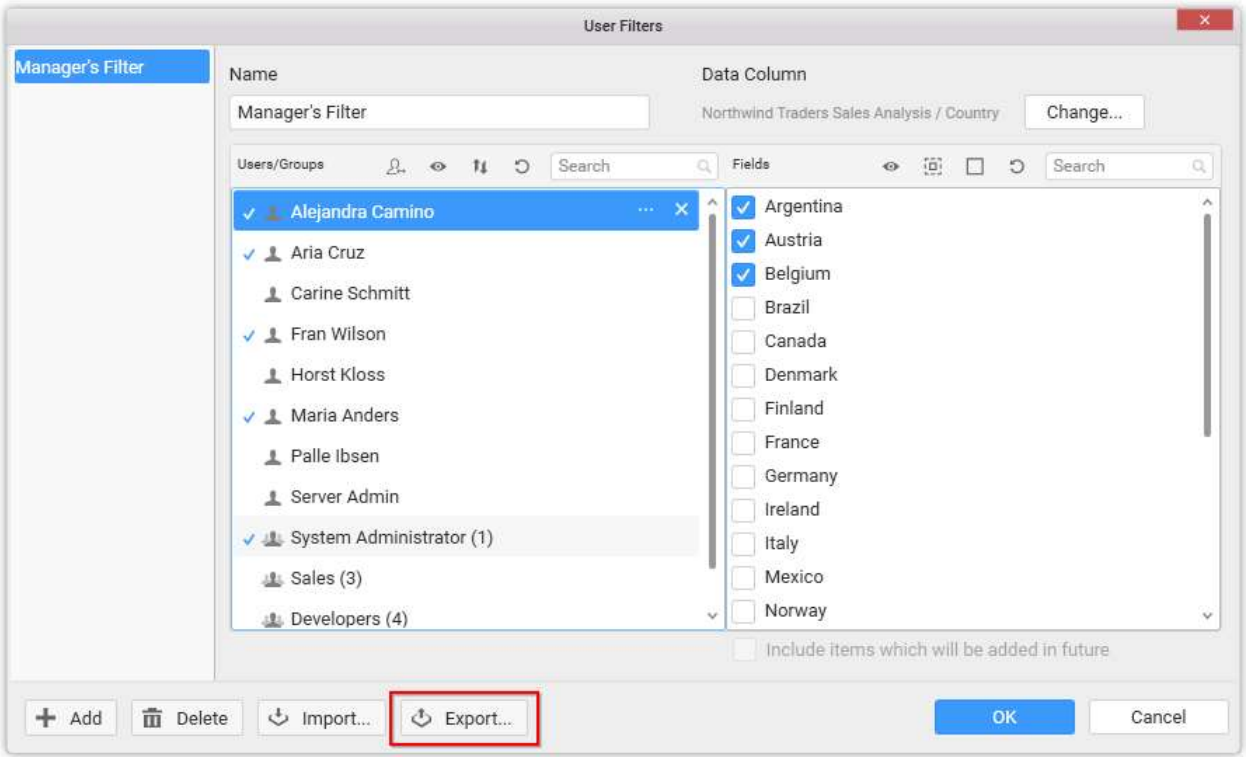
Sharing User Filters

Adding same user filter configuration for multiple dashboards provided, they all have identical data sources configured, can be achieved through the export/import options of user filters. Hence, you can reuse the user filter created for one dashboard to other dashboards.

Exporting User Filters

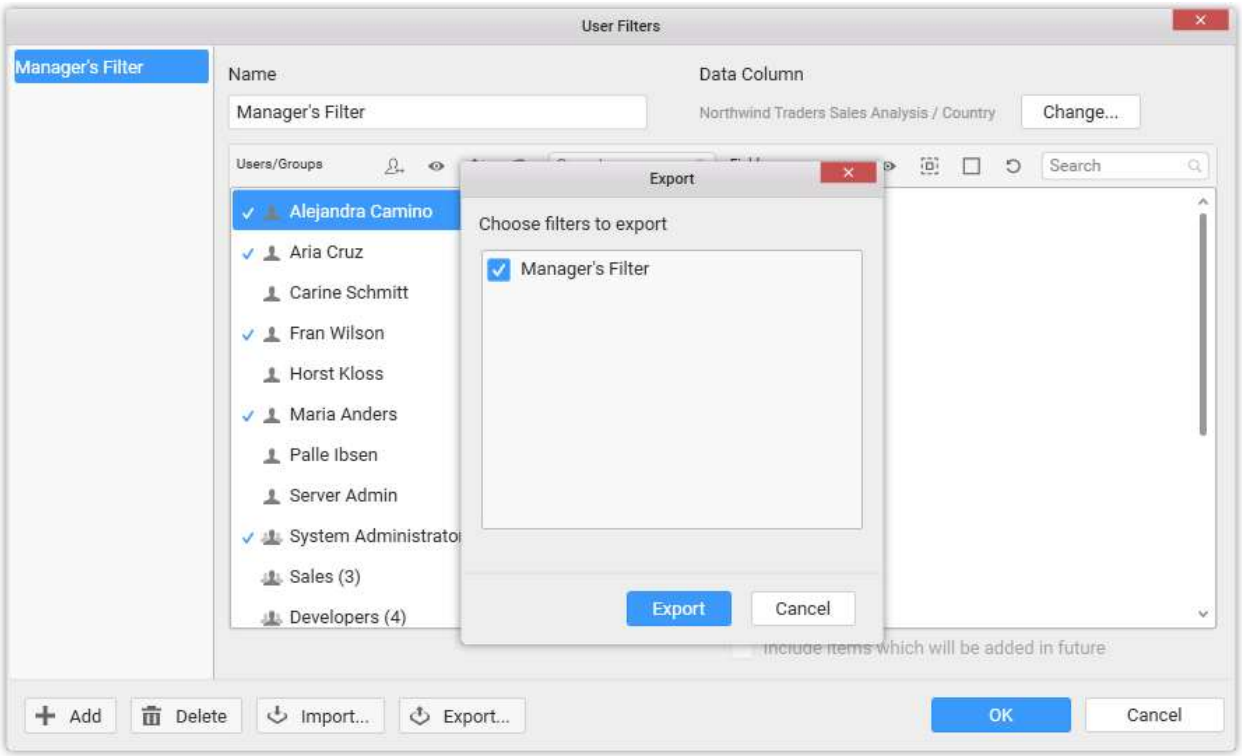
You can export the user filters created for a dashboard to an XML file and reuse it in another dashboard that has identical data source. Here is the procedure to achieve the same.

Once a user filter was created, click the Export button (highlighted below) at the bottom of the dialog.



Now the Export dialog opens.

From the dialog, choose the user filters which you need to export.

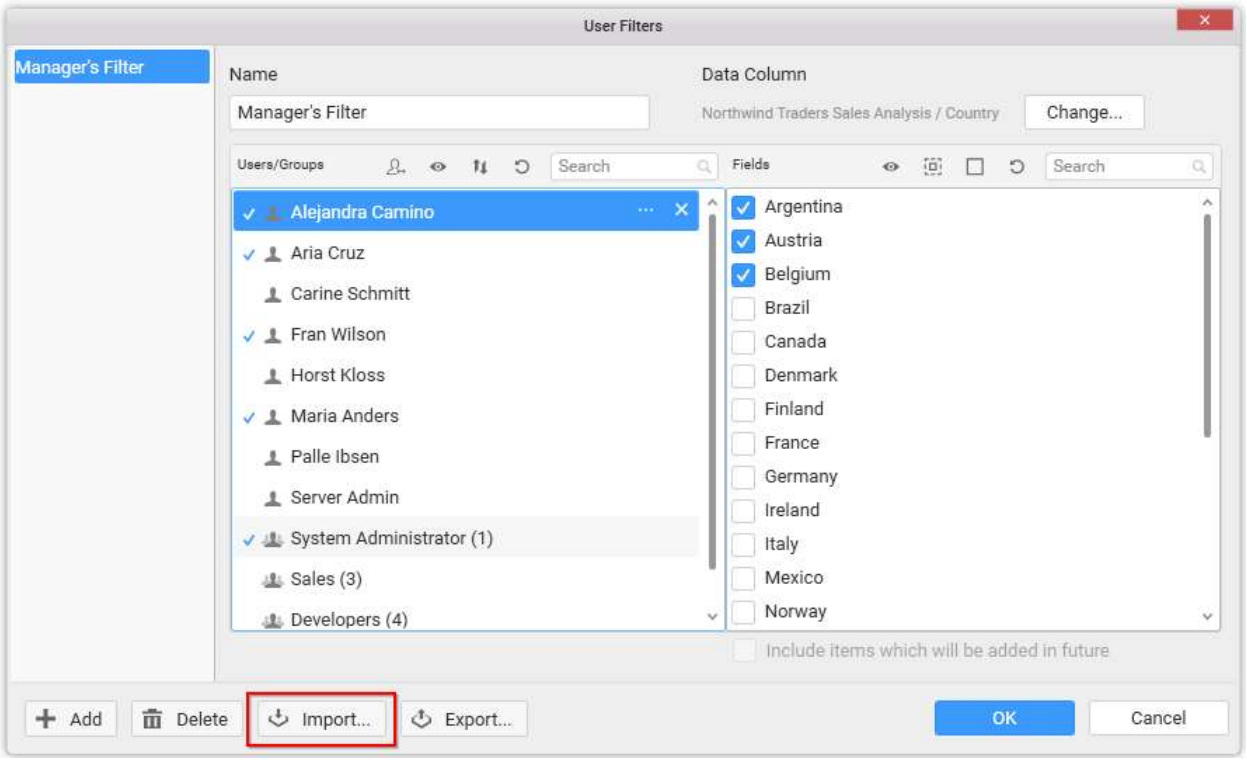


Click Export. A save dialog will open up, where you can specify the location and name of the file.

Importing User Filters

You can import already saved user filters in a file through the Import dialog. Here is the procedure to achieve the same.

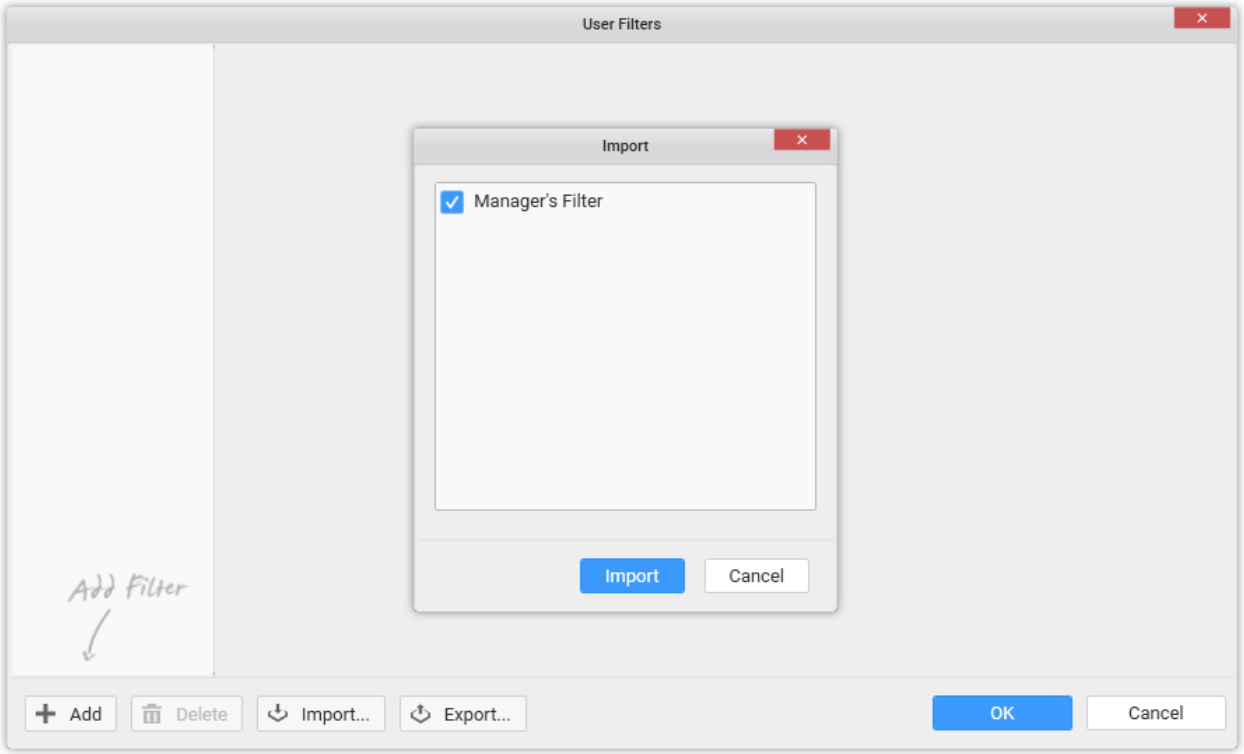
In the User Filters dialog, click Import button (highlighted below) at the bottom.



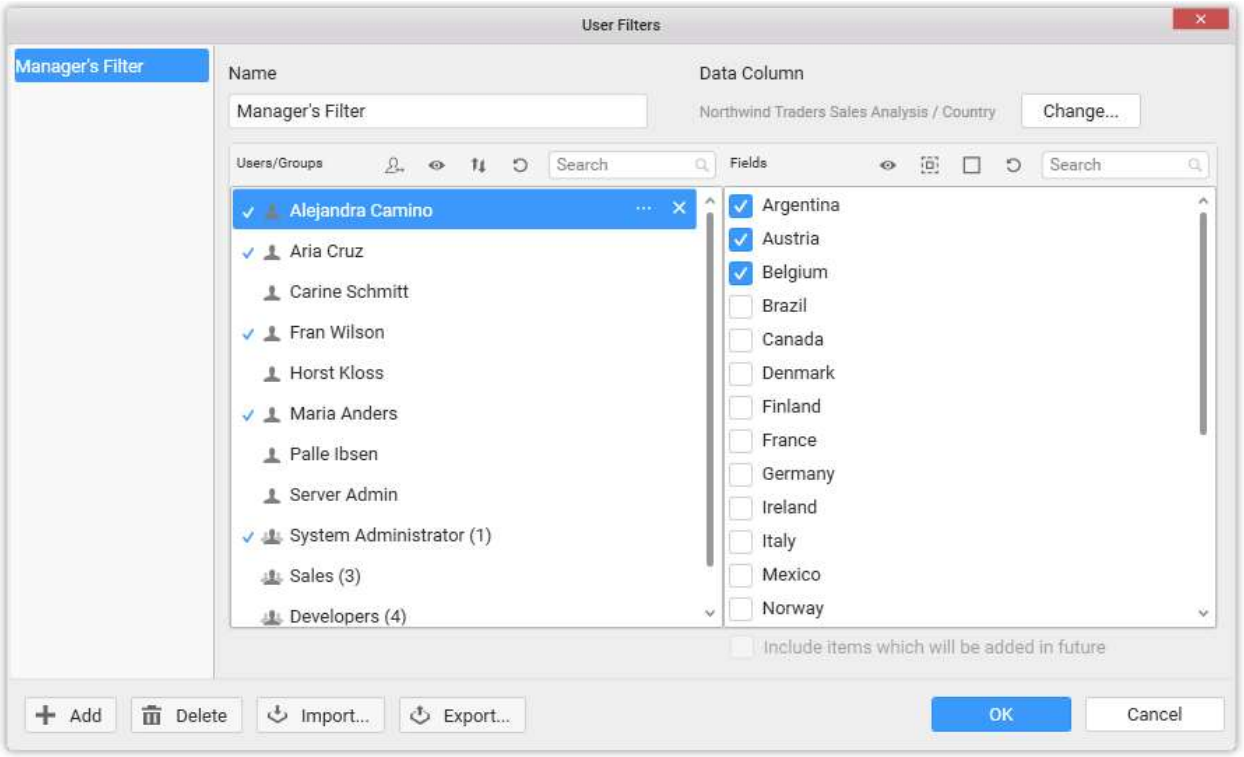
Now the Import dialog opens.

An open file dialog will pop up. Select the XML file (that holds the user filters previously saved) you need to import and click Open.

Import dialog opens with the user filters list. Choose the user filters which you need to import and click Import.

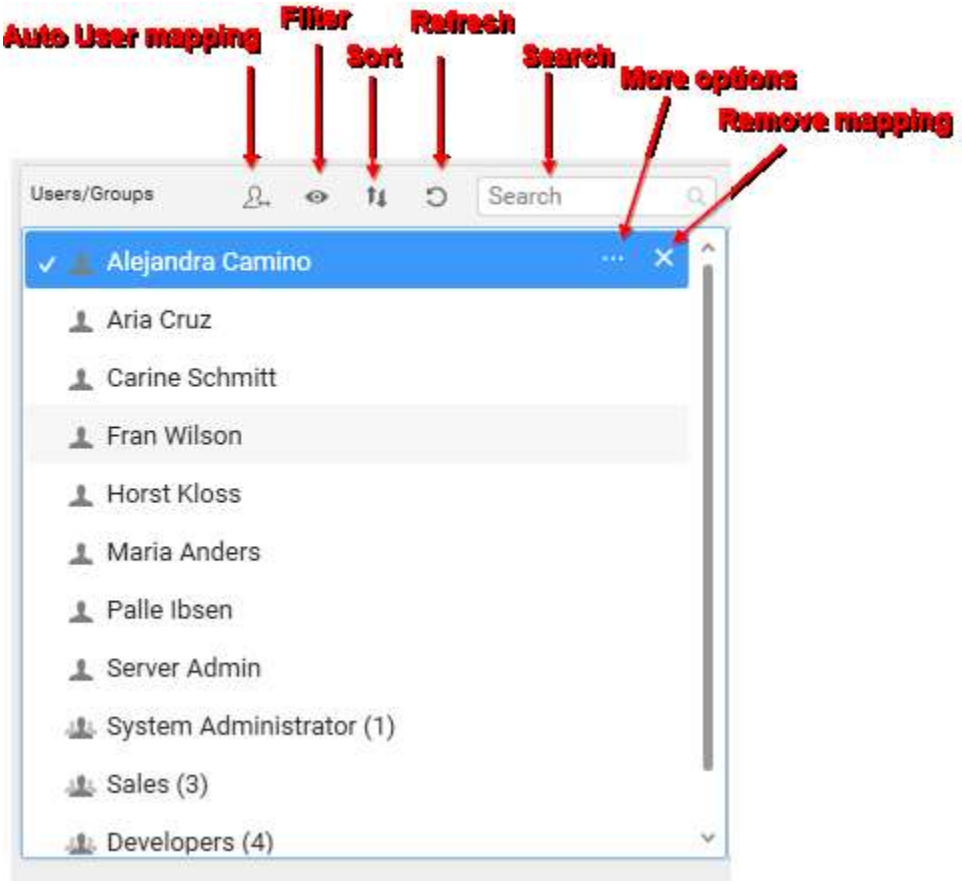


The selected user filters will get added to the current dashboard.



Note: There should be a column with the same name as that of the column used in the user filter being imported. Otherwise, the import will fail.

Users/Groups Functionalities

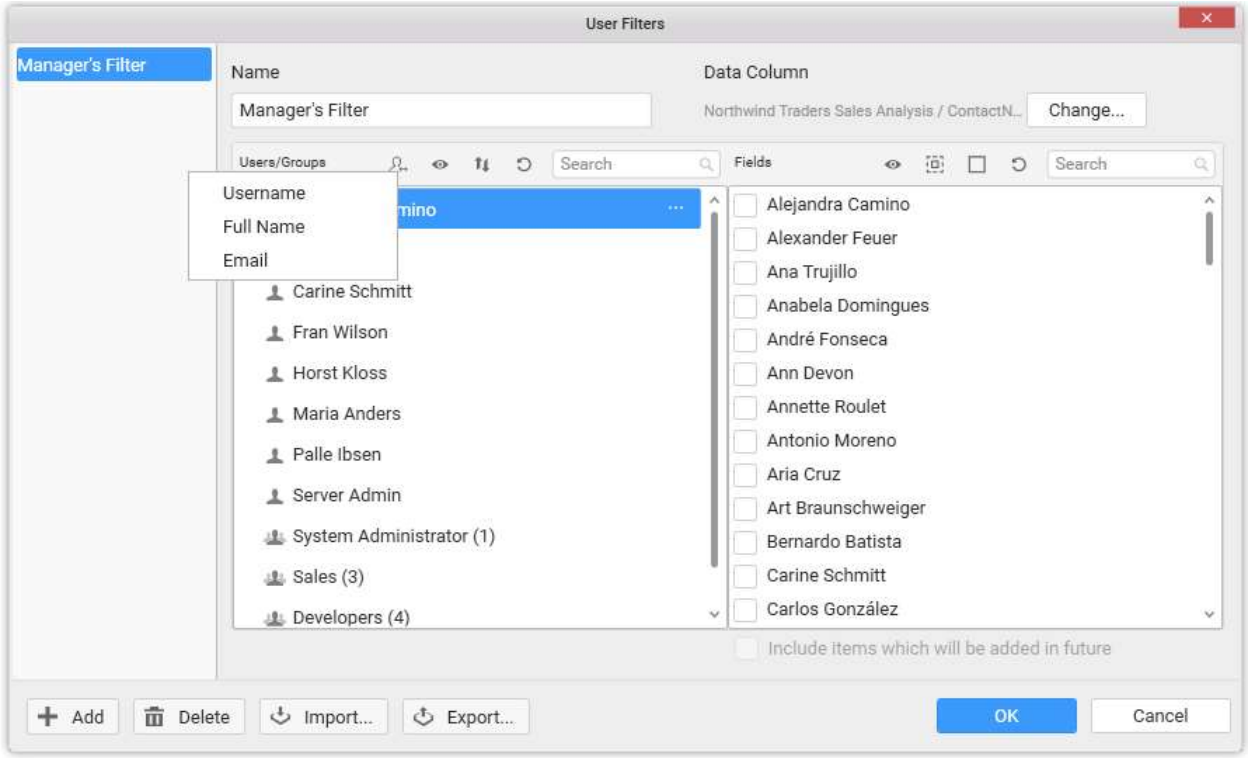


Auto Mapping of Users

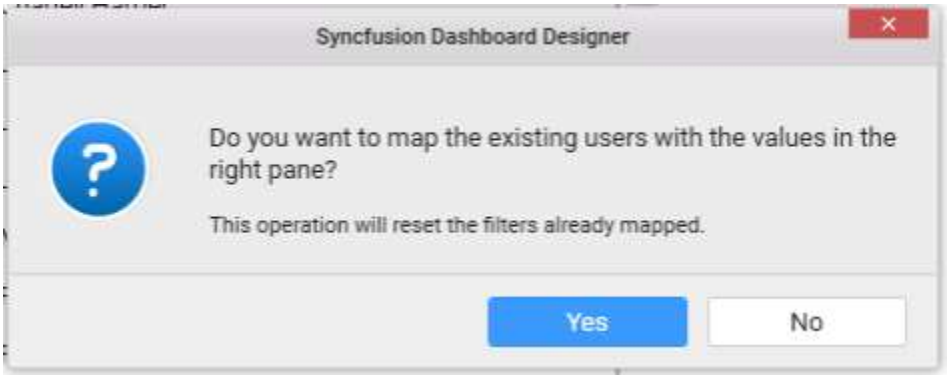
When the column chosen for applying user filter contains User Name, Full Name, or Email that matches one in Dashboard Server User Account, you can automatically map corresponding values in both left and right panes in the User Filters dialog without much effort. Here is the procedure to achieve the same.

Click Auto User mapping button at the top of the Users/Groups pane in User Filters dialog.

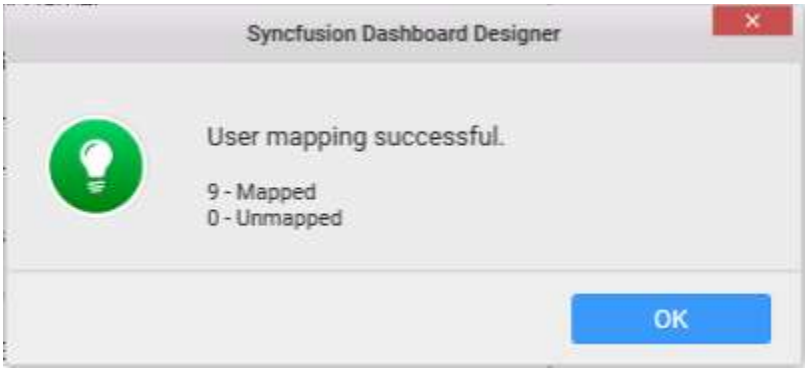
Select the item from the drop down menu based on the data in the column connected (In this example, we are mapping based on user's Full Name).



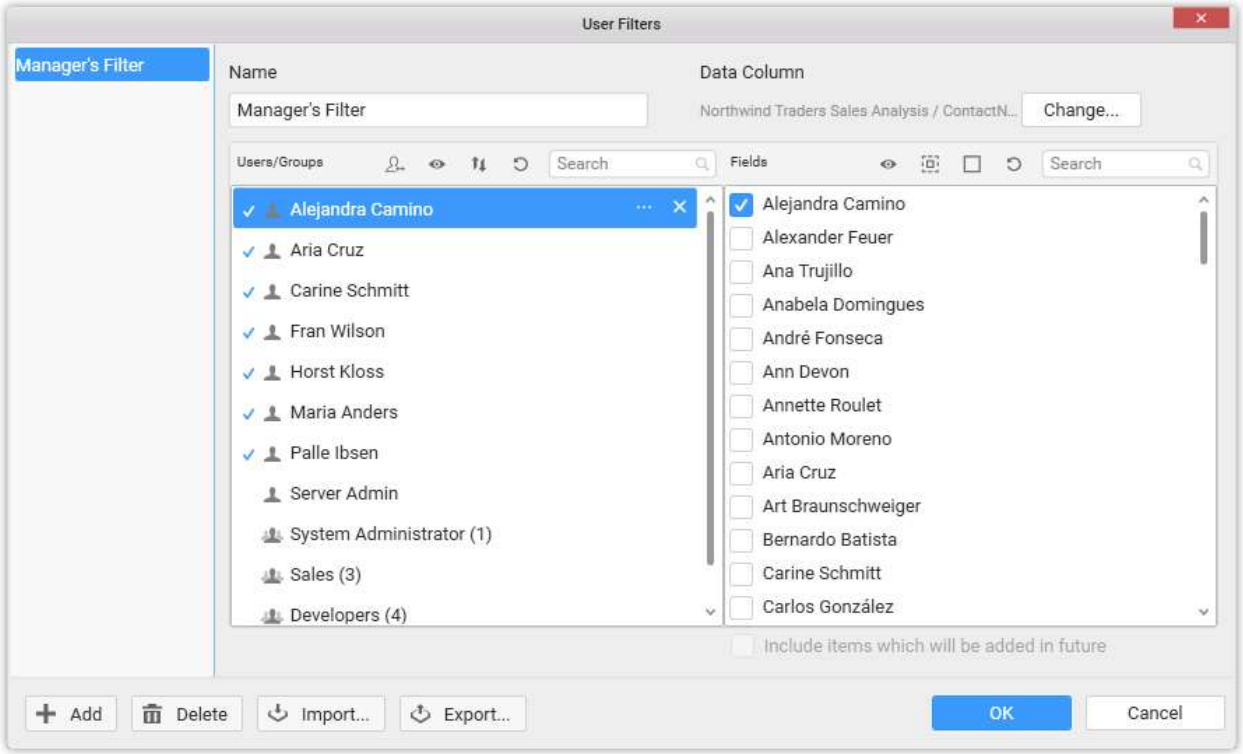
Now, a message confirmation dialog will get displayed.



Click Yes in the prompt window. Now a message box will pop up that showcasing the status of the user mapping.



All the users whose name matches with the value will map automatically.



Filter

You can filter the users/groups pane based on whether a value is mapped or not using the Filter button at top of it. Click the Filter icon and select any of the following values in the drop down menu to filter the users/groups list.

All Mapped * Unmapped

Sort

You can toggle the users/groups sorting in the pane by clicking the Sort icon. The list can be sorted in ascending or descending order, whose sort order is ascending, by default.

Refresh

You can reload the user/groups list from server anytime by clicking the Refresh icon.

Information: The mapping set for users will not be affected when the list is refreshed.

Search

You can filter the user/groups list based on the user's name by entering the first few characters of user's name in the search box at top.

Remove Mapping

You can remove the mapping made for a user by clicking the Remove button next to the user name. This will remove the values bind for the user, and the user will not be able to view any data in the dashboard.

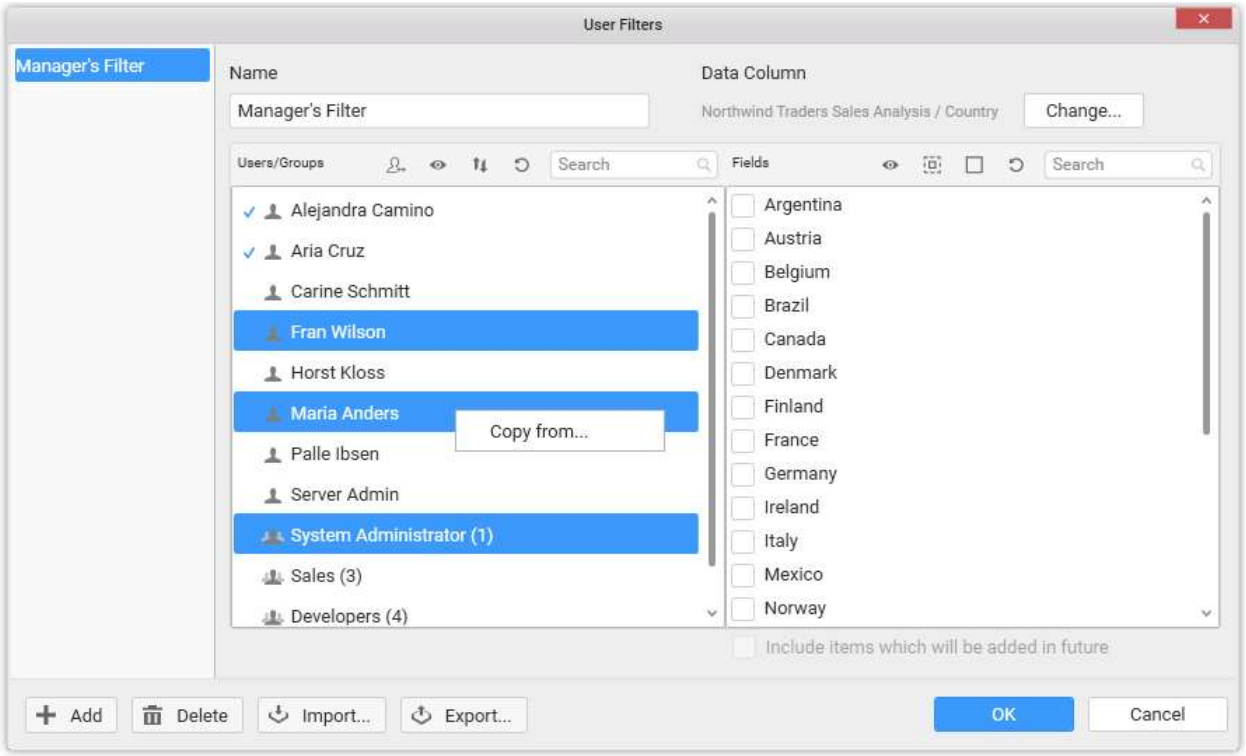
More Options

You can navigate to additional options available such as copying or merging the mapping already set for one user to another or remove the applied mapping or view the users present in a group, by clicking the more options button. The same options can be accessed via context menu by clicking the right mouse button.

How to copy the mapping set for one user to another

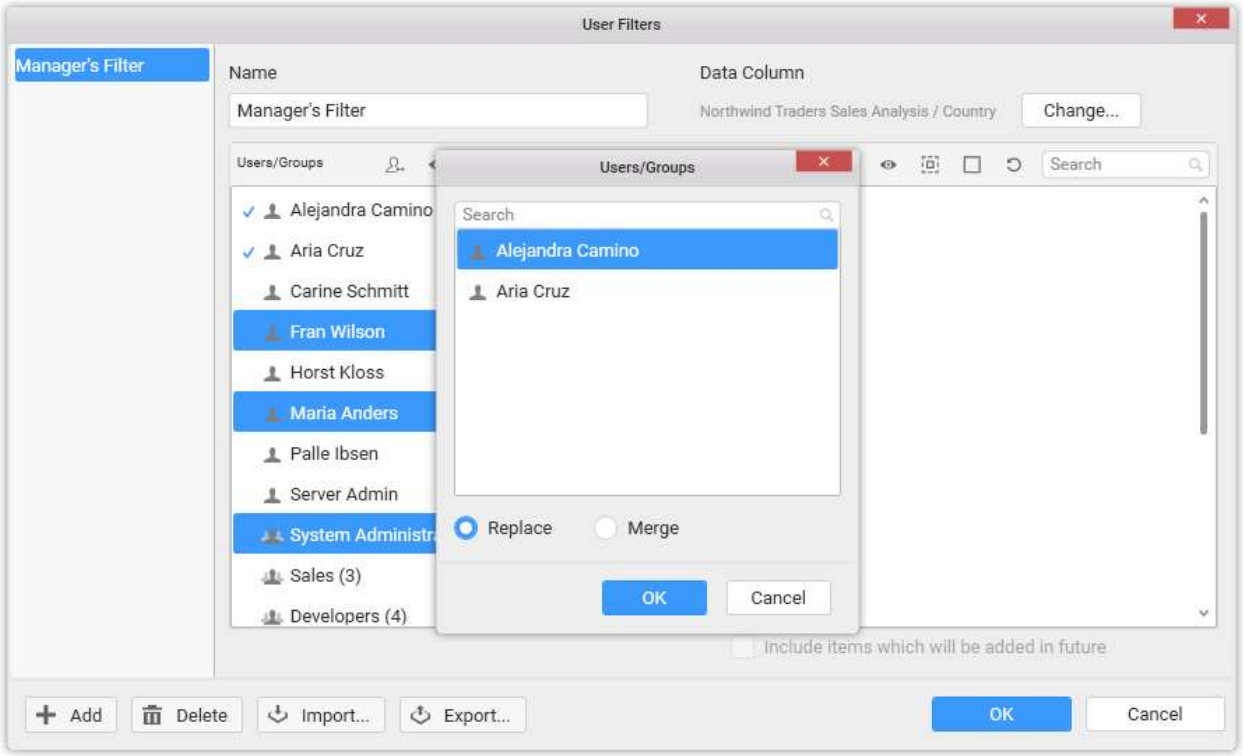
You can copy the mapping done for one user to another user(s) or group(s) through the following procedure.

Select the user(s) or group(s) for whom you need to set mapping. Open context menu by clicking the right mouse button.

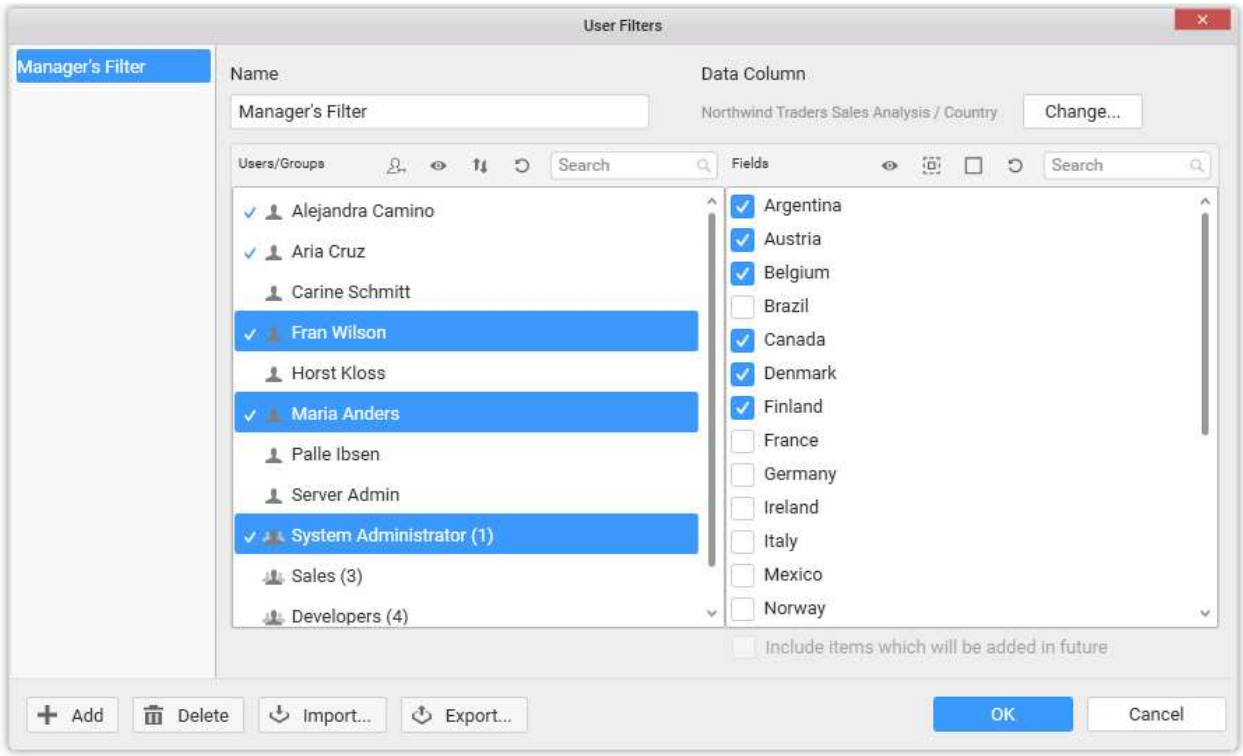


Click Copy from… menu item. Now, Users/Groups dialog will open.

Select the user/ group from which the mapping data need to be copied from.



Click OK to copy the mapping details.

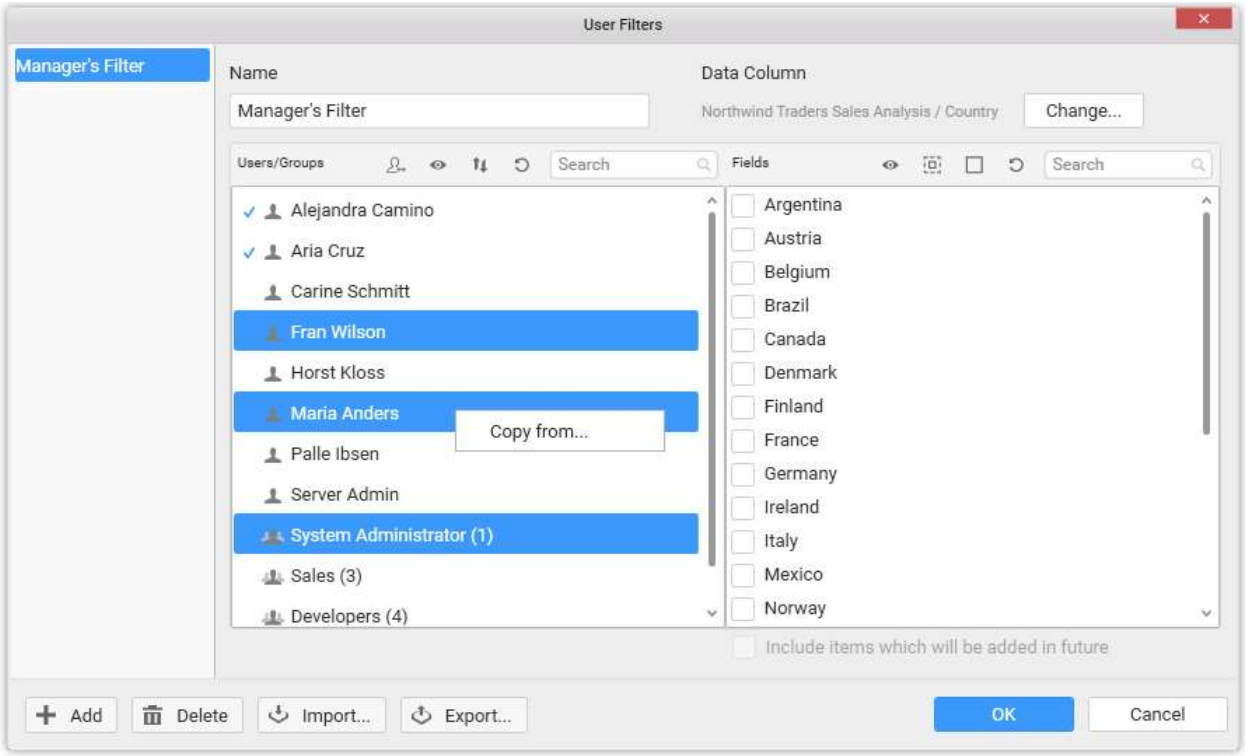


Note: Copying mapped data from another user/group will reset the already mapped data for the selected user. If you need to retain both information you can merge the data using Merge dialog.

How to merge multiple user(s) or group(s) mapping data to another user(s) or group(s)

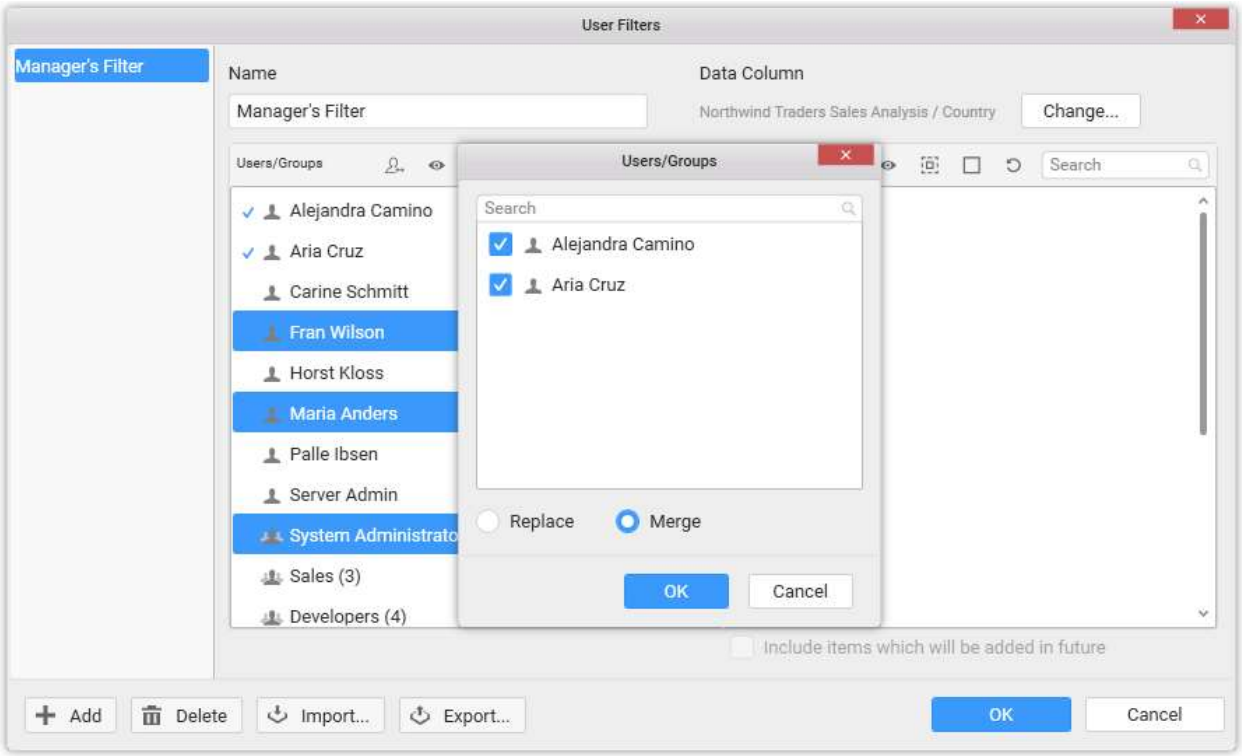
You can merge the mapped data of multiple users or groups to another user(s) or group(s) by using the merge option through the following procedure.

Select the user(s) or group(s) for whom you need to set mapping. Open context menu by clicking the right mouse button.



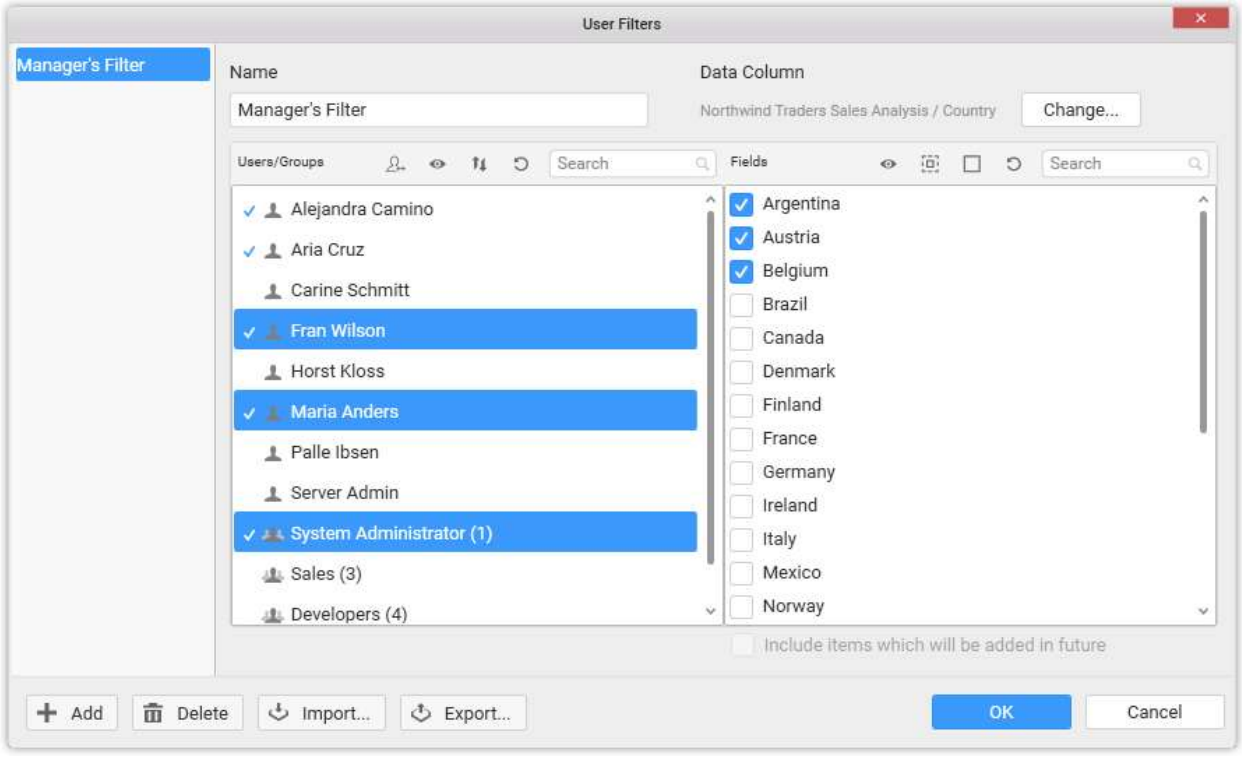
Select Copy from… menu item. Now, Users/Groups dialog opens.

Select Merge option.



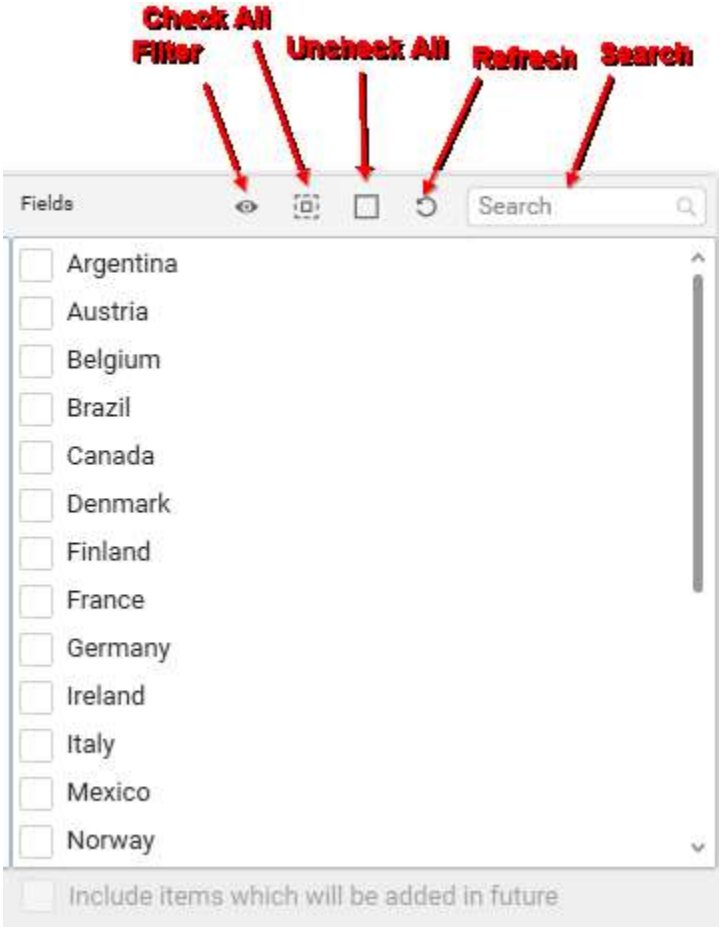
Select the user(s) or group(s) whose mapped data need to be merged to the selected user(s) or group(s).

Click OK to complete the merging process.



Information: The previously mapped data for the selected user will be maintained when you perform merge operation.

Column Values Functionalities



Filter

You can filter the column values based on whether it is checked or not using the Filter button at top of Fields pane. Click the Filter icon and select any of the following values in the drop down menu to filter the field values list.

- All
- Checked
- Unchecked

Check All

Click the Check All icon to check all the values in the list. This will allow the mapped user to access all data in the dashboard.

Uncheck All

Click the Uncheck All icon to uncheck all the values in the list. This will restrict the mapped user to access any data in the dashboard. By default,

when a user filter is created for a dashboard, all users will be restricted to view the data in the dashboard unless a value is bind for that user.

Refresh

You can reload the field values list from database anytime by clicking the Refresh icon.

Search

You can filter the column values list based on the value text by entering the first few characters of value in the search box at top.

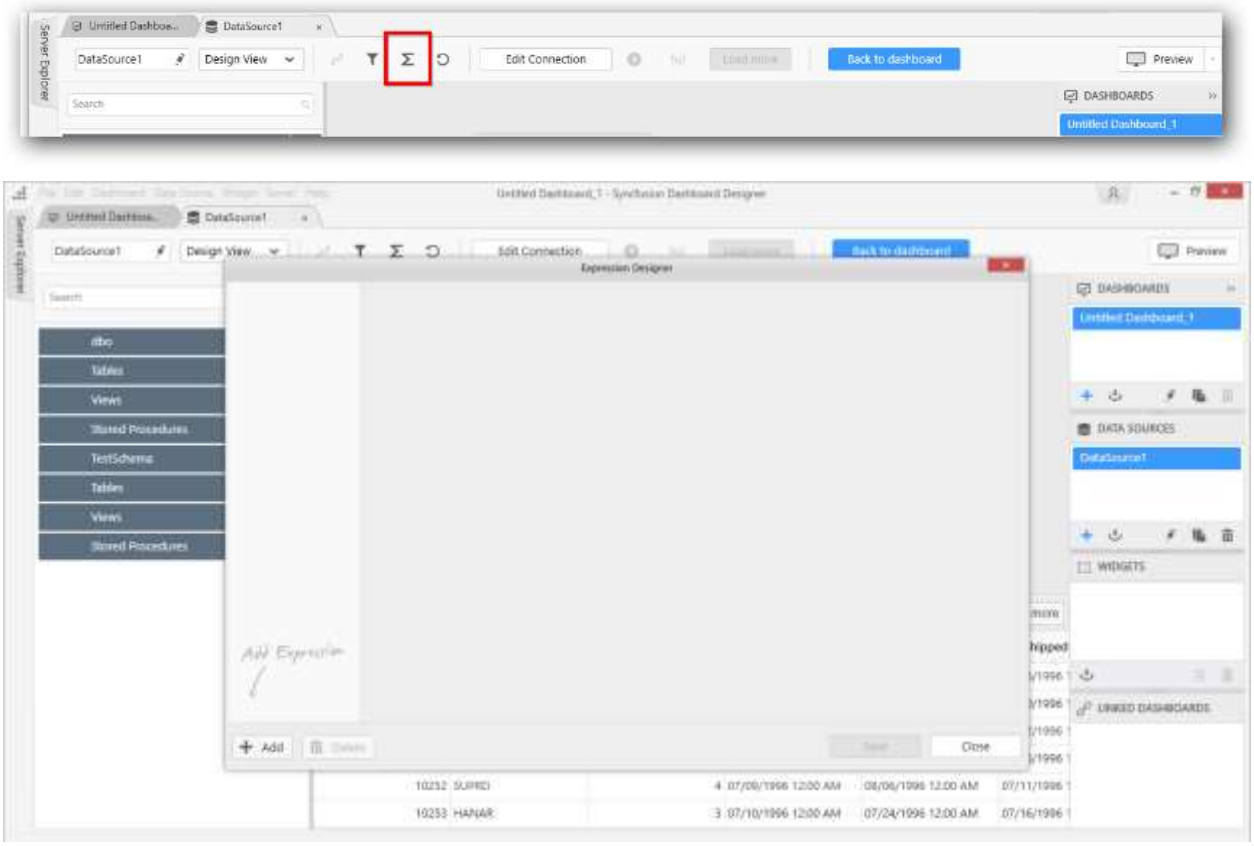
Configuring User Filter Using Expression Columns

The user based filter can be applied directly to the data source using the expression column thereby restricting the data showcased to the user. As this filter will be applied at Data Source itself, based on the logged in user in the Dashboard Designer, the data will be restricted.

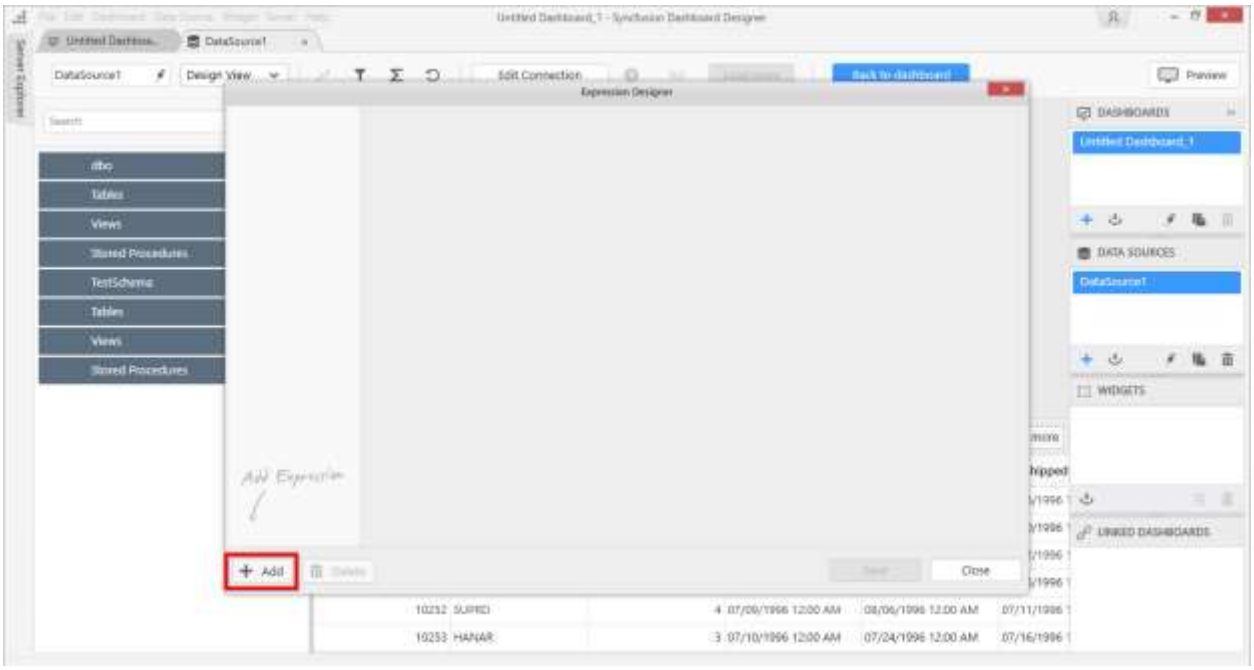
Creating User filter

Open the Data Source from DATA SOURCES window whose data need to be applied with user filter.

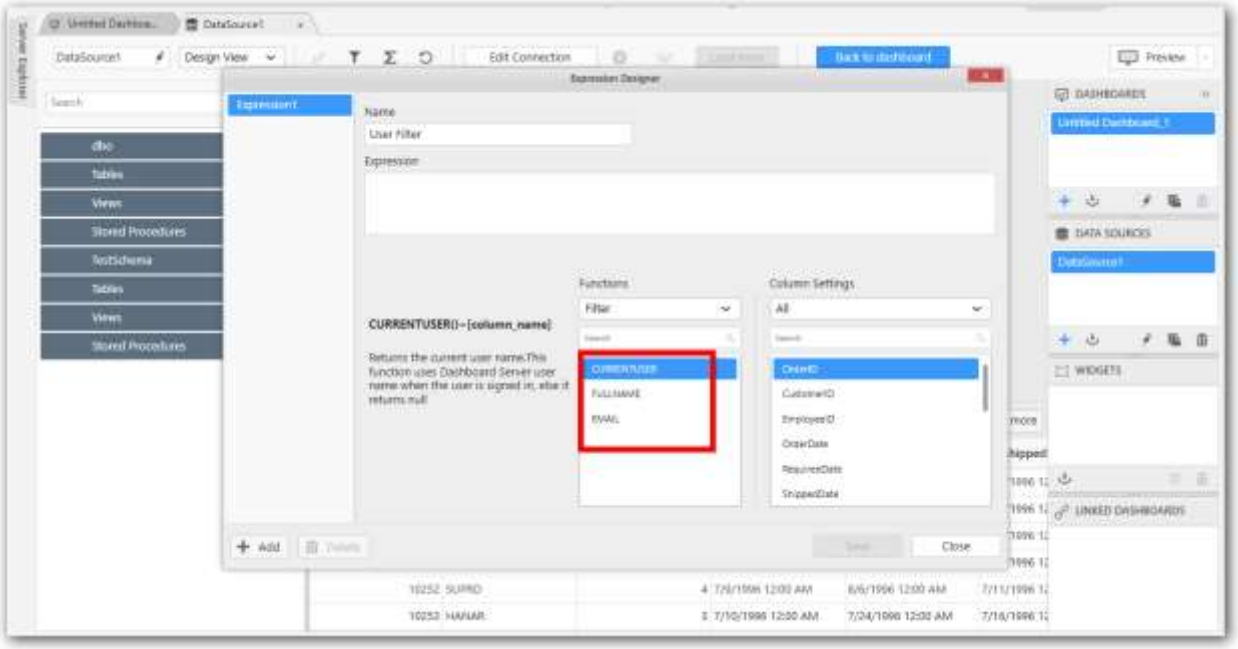
In the opened Data Source tab, open the Expression Designer window by clicking on the Expression Editor button from the toolbar.



Add a new filter by clicking on the Add button at the bottom left corner of the Expression Designer window.



In the new expression view, generate a user filter expression using any of the three predefined functions available and save the expression by specifying a name for the expression column.



The Predefined User Filter functions are,

CURRENTUSER - Returns the user name of the currently logged in user whose user account exists in the connected Dashboard Server. Otherwise, null.

Syntax: CURRENTUSER()=[Column Name]

For Example, CURRENTUSER()=[Car Inventors]

FULLNAME - Returns the full name of the currently logged in user whose user account exists in the connected Dashboard Server. Otherwise, null.

Syntax: FULLNAME()=[Column Name]

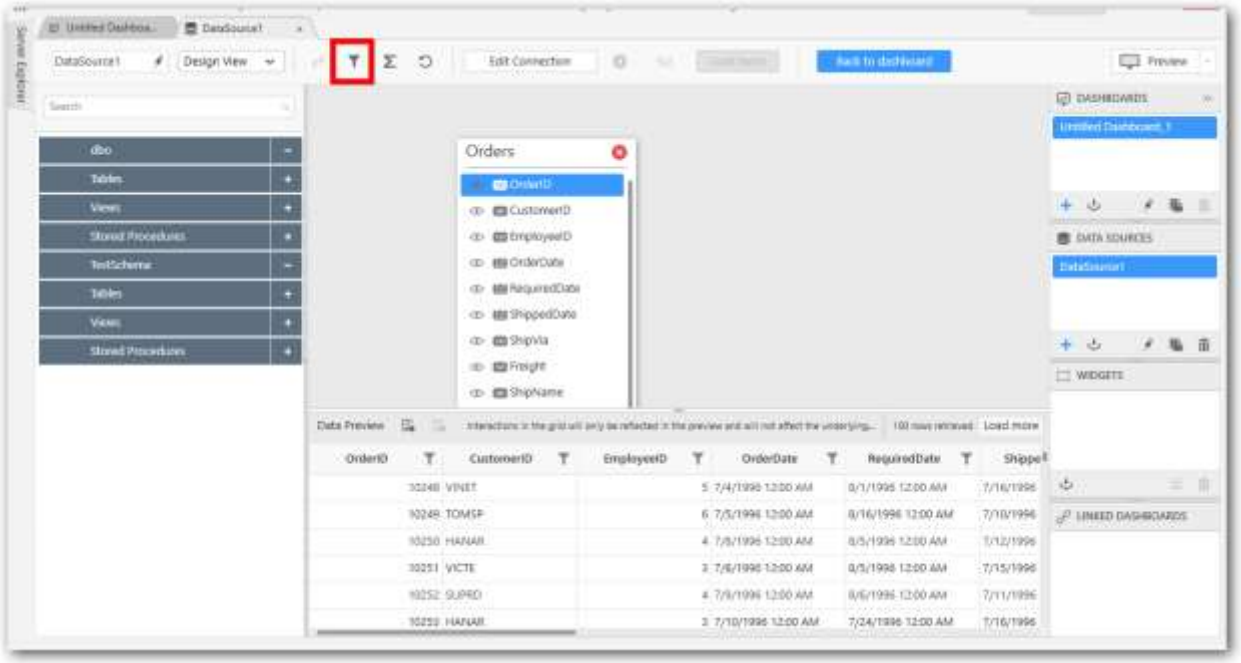
For Example, FULLNAME ()=[Car Inventors]

EMAIL - Returns the email of the currently logged in user whose user account exists in the connected Dashboard Server. Otherwise, null.

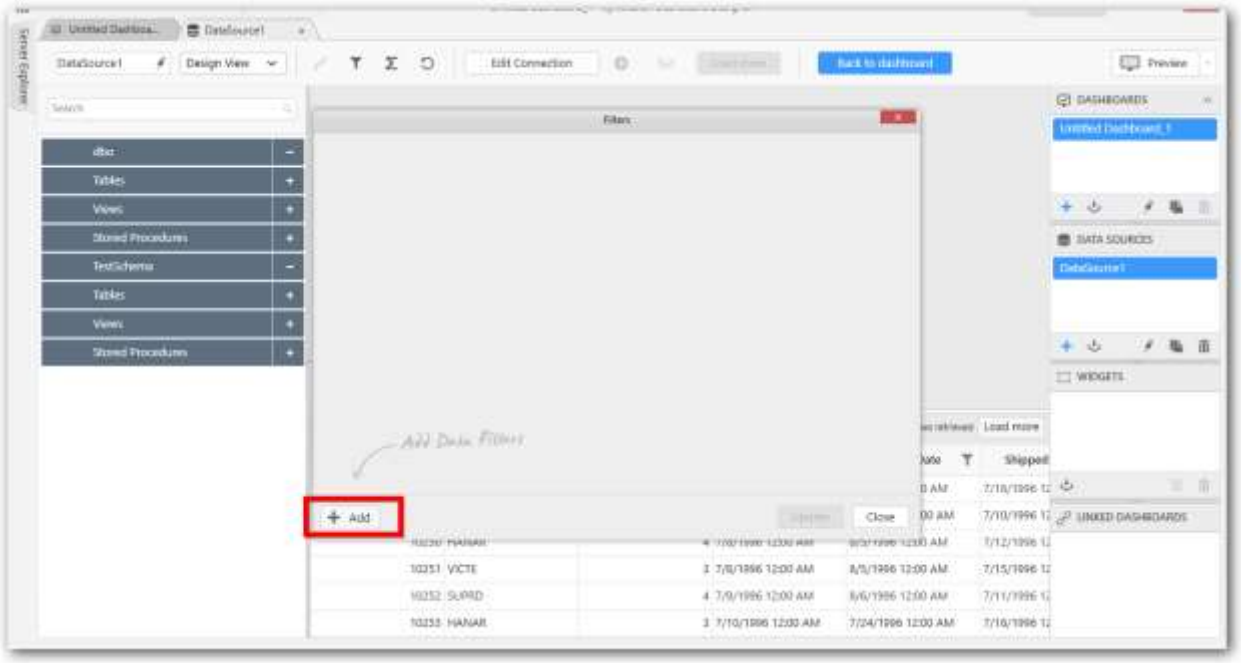
Syntax: EMAIL()=[email id]

For Example, EMAIL ()=[email id]

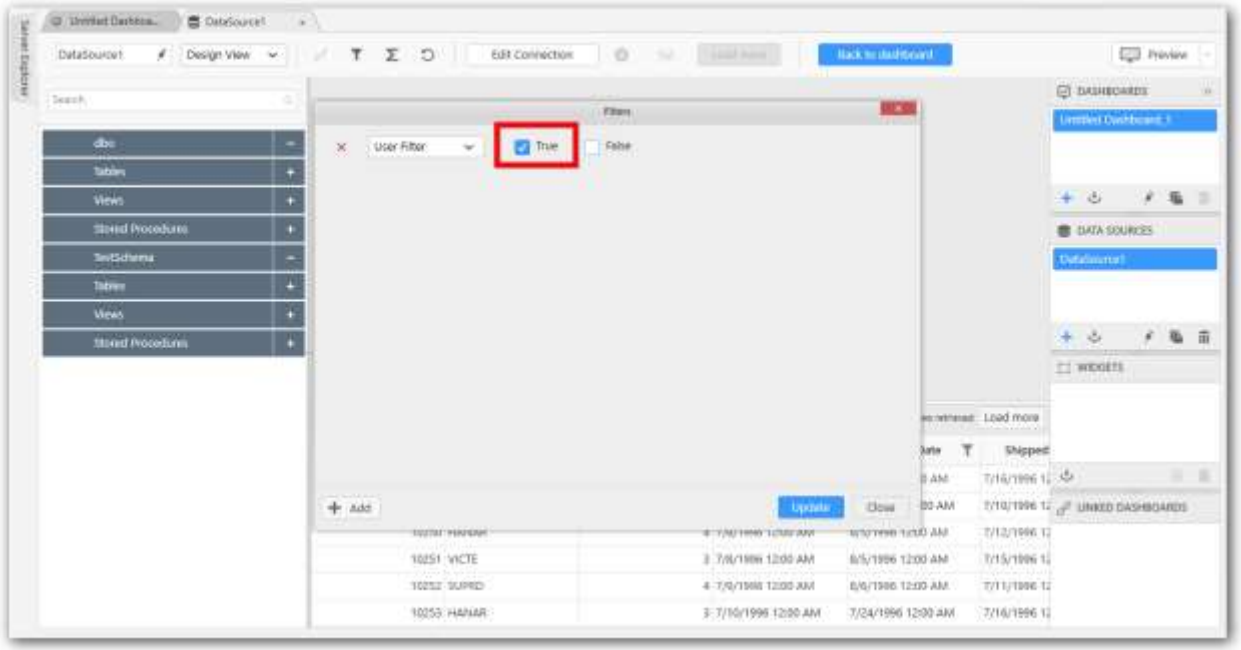
Once the Expression column is created, open the Filters window by clicking on the Add Filters icon from the toolbar to add this column as filter to the respective data source.



Click on the Add button to add a new filter.



Select the newly created expression column name from the dropdown list, enable the True check box and disable the False check box and save the filter.



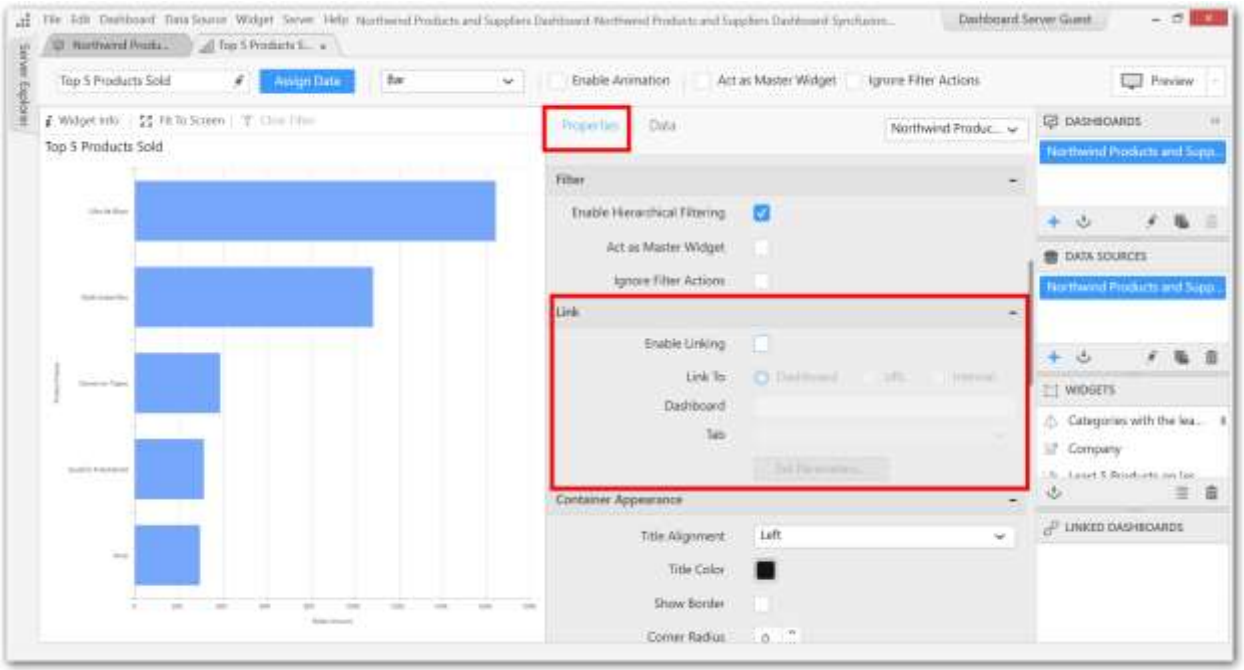
Now the dashboard will be filtered based on the logged in user such that only those records matching the respective column value with the one returned by expression column, will be displayed. Upon logging out the from the Dashboard Server, the dashboard will be showing no records.

Linking URLs and Dashboards

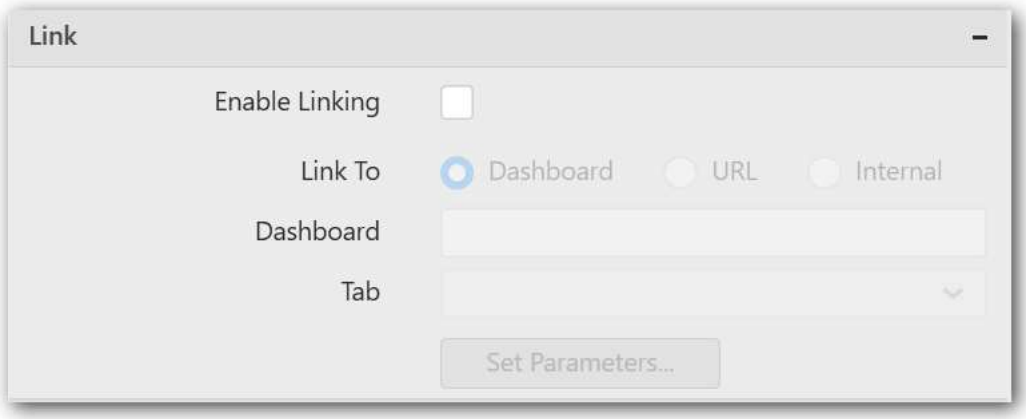
You can link any URLs or Dashboards (external/internal) to a visualization widget by enabling the Enable Linking property. It provides three types of linking as follows:

1.Dashboard 2.URL 3.Internal

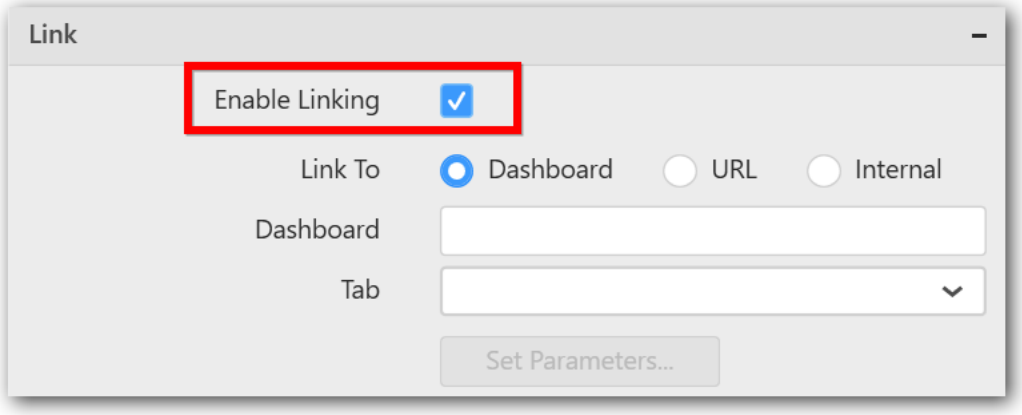
The Enable Linking option is available in the Properties tab of the widgets.



By default Enable Linking property will remain unchecked.



To enable linking select the Enable Linking checkbox and Dashboard linking option will be chosen as the default linking option.

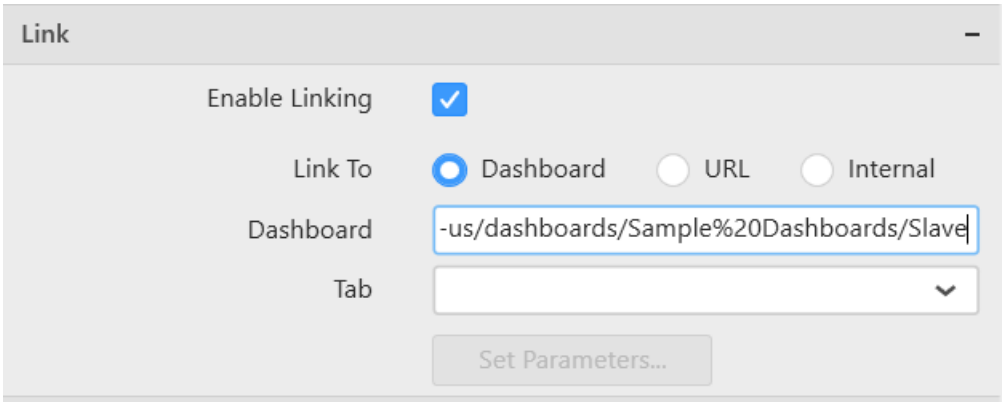


Dashboard

Dashboard linking option allows you to link the dashboards published in the dashboard server to your widget.

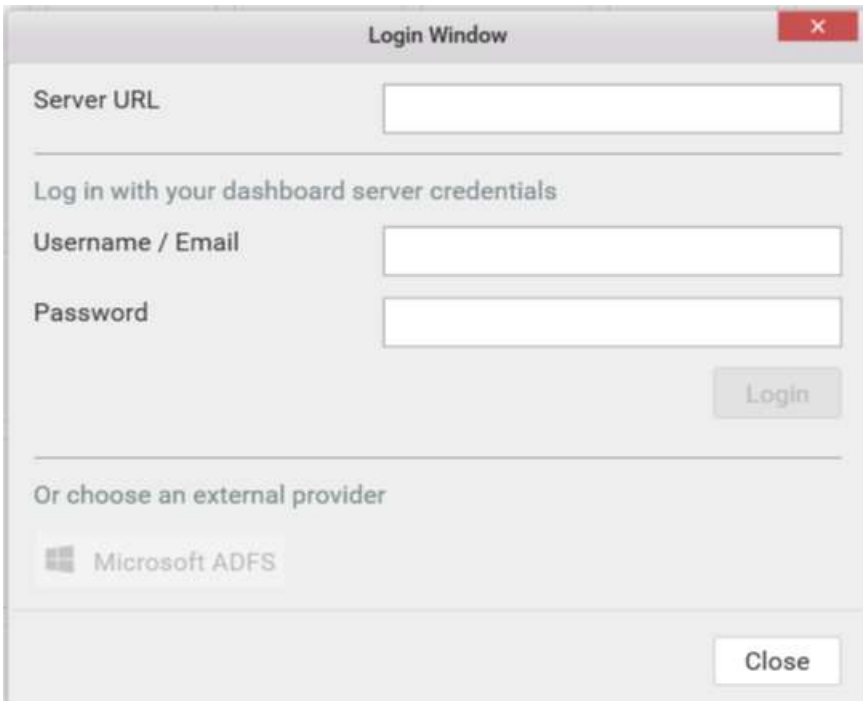
Select the dashboard option and paste the published dashboard HTTP/HTTPS URL.

Note: Refer here for getting the published dashboard link from the dashboard server.



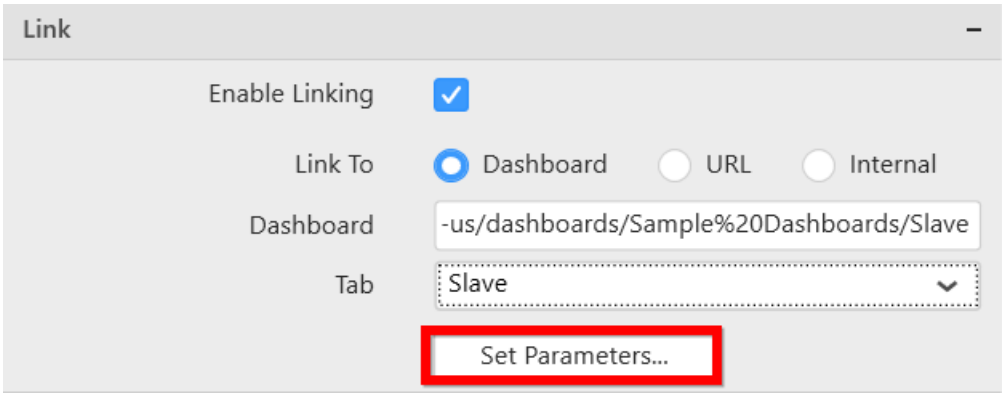
Information: For dashboard linking option you need to login to your dashboard server account in the dashboard designer. If you are not logged in already, the Login Window will be displayed once you entered the dashboard link in the text box.

You can provide the server credentials and click the Login.



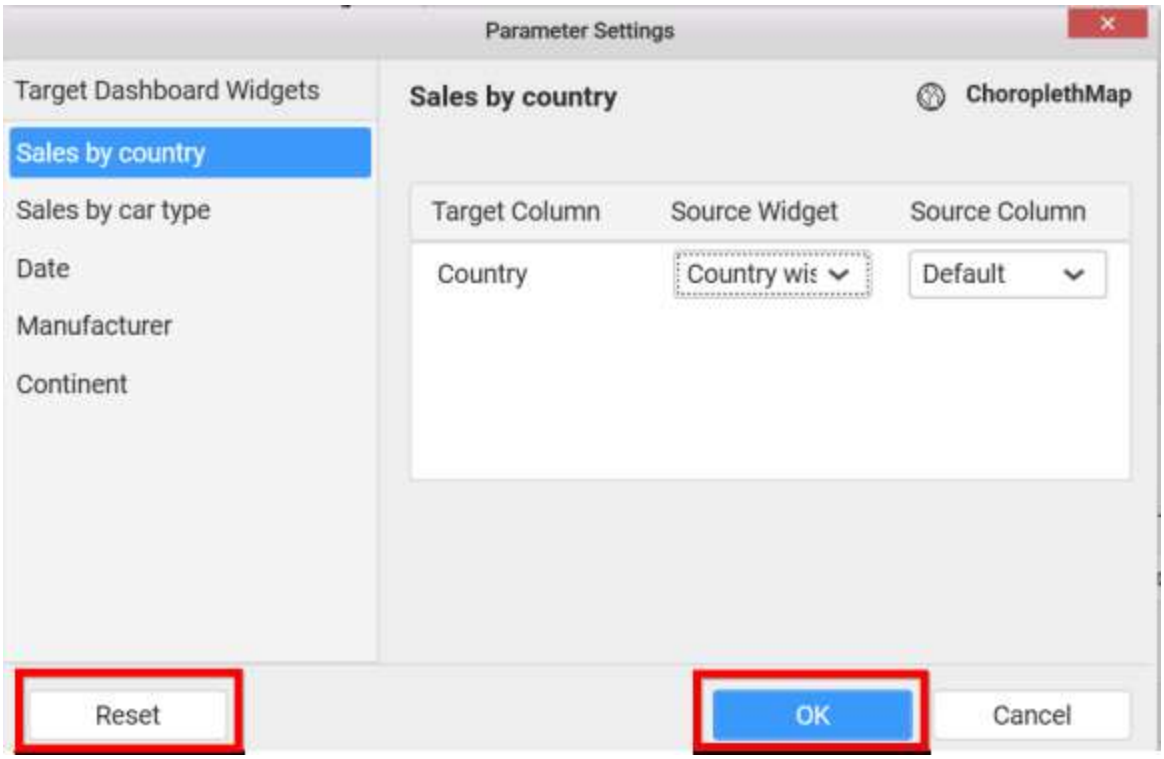
The list of tabs available in the linked dashboard will be listed in the Tab drop-down box. If the linked dashboard contains only a single tab it will be automatically chosen as the selected tab.

Now, the Set Parameters... option will get enable. You can click the Set Parameters....



The Parameter Settings window will be displayed. You can change the source widget and source column as required and click OK.

If you need to reset changes to the source widget and source column selection, click the reset and select ok.



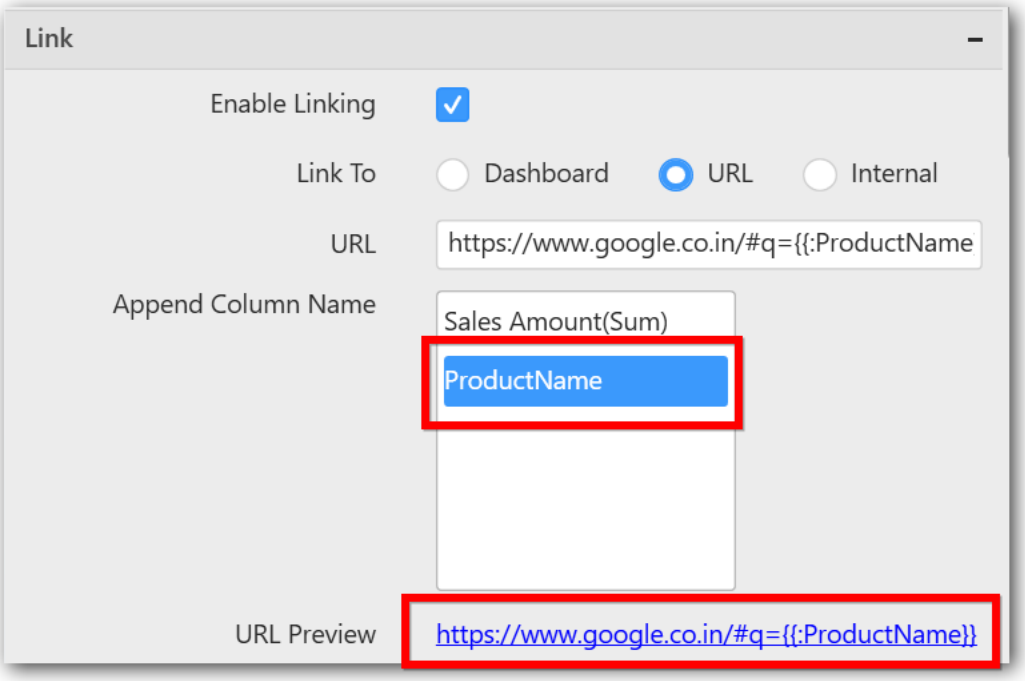
Note: To find the examples of dashboard linking option, refer to this Knowledge base article.

URL

URL linking allows you to link to the valid web URLs.

Select the URL option and enter the Web URL in the URL text box. If you click on the column names listed in the

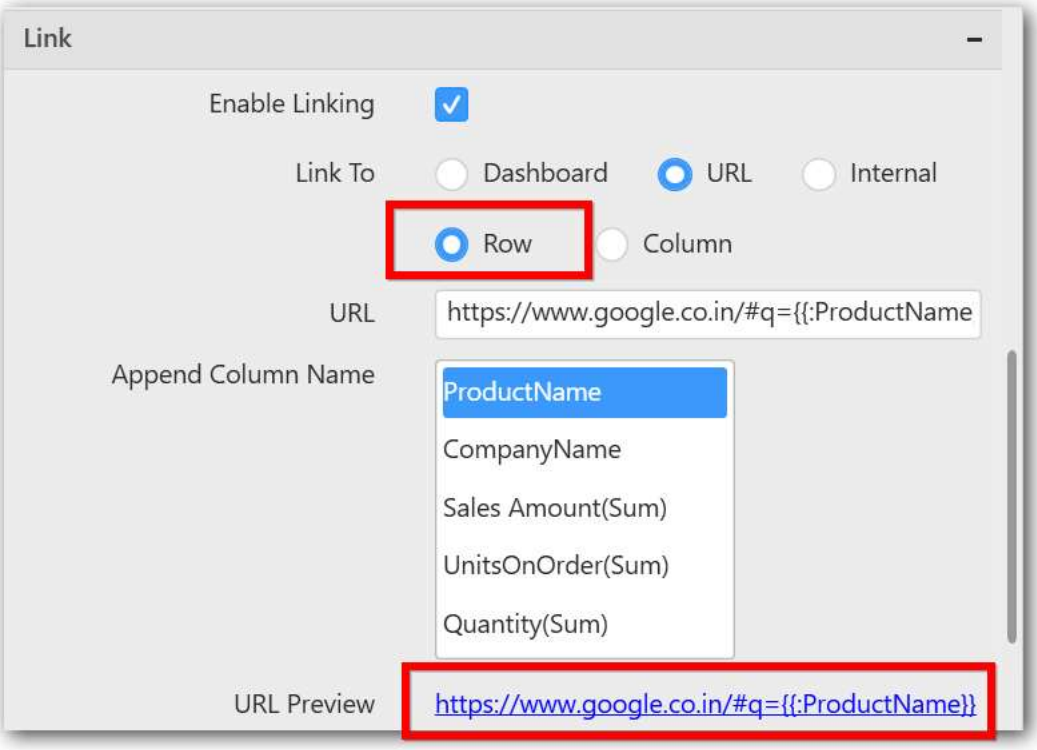
Append Column Name list , it will be appended to the URL entered in the URL text box.



For grid widget, you can get URL based on Row and Column. The dashboard linking property will be same as explained above.

Row

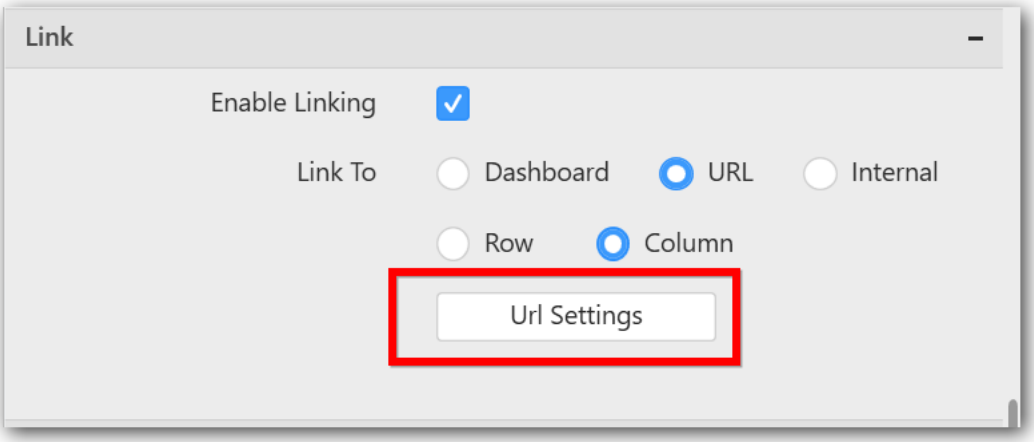
Enter the Web URL in the URL text box. If you click on the column names listed in the Append Column Name list , it will be appended to the URL entered in the URL text box.



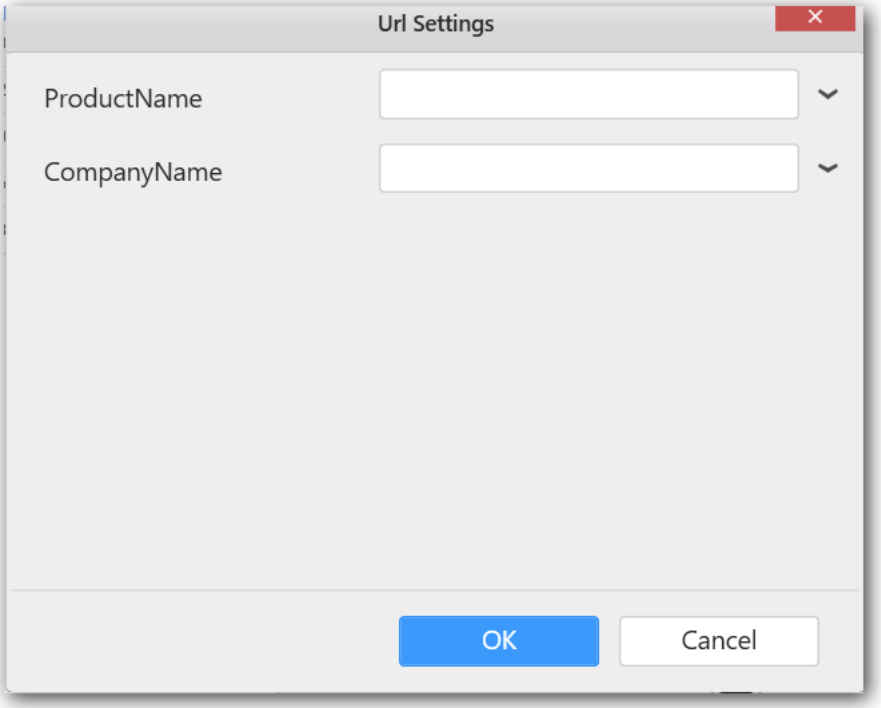
You can preview the linked URL using the URL preview option. If you click the preview URL link,it will be opened in a browser.

Column

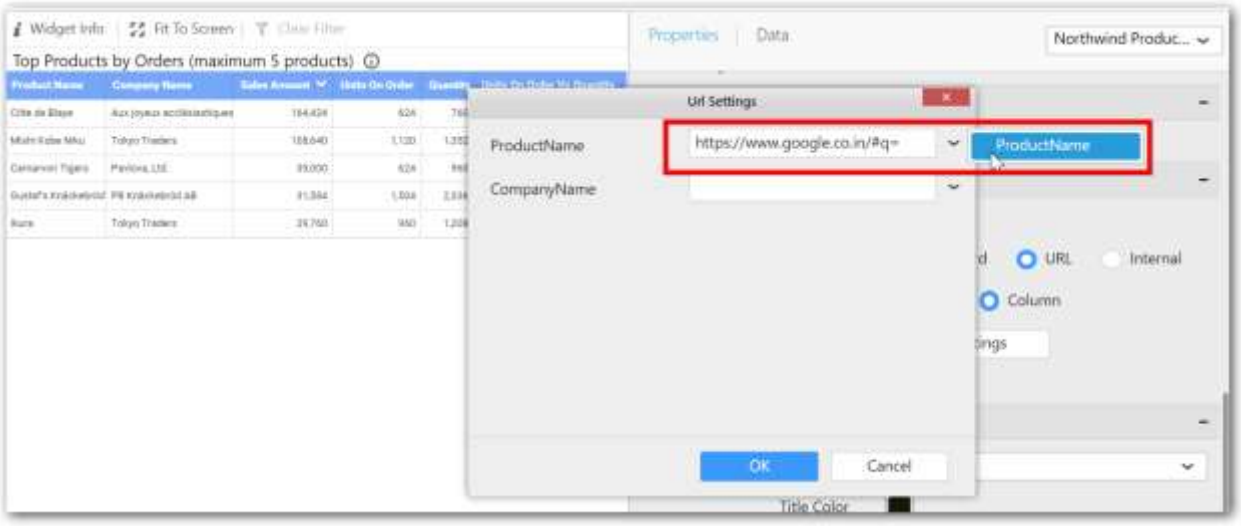
You can select the column and click the Url Settings.



The Url Settings wizard will be displayed.



Add the web Url to ShipName field and select the column name to append the column name to the link.



Now, the link will be added to the grid widget. You can select the URL link that you want to refer.

Current domain linking

Current domain linking option allows to link the reports within the same domain without mentioning the full domain name.

If you hosted a sub application in the server, you can provide the domain name in the URL by using the keyword **~currentdomain~**.

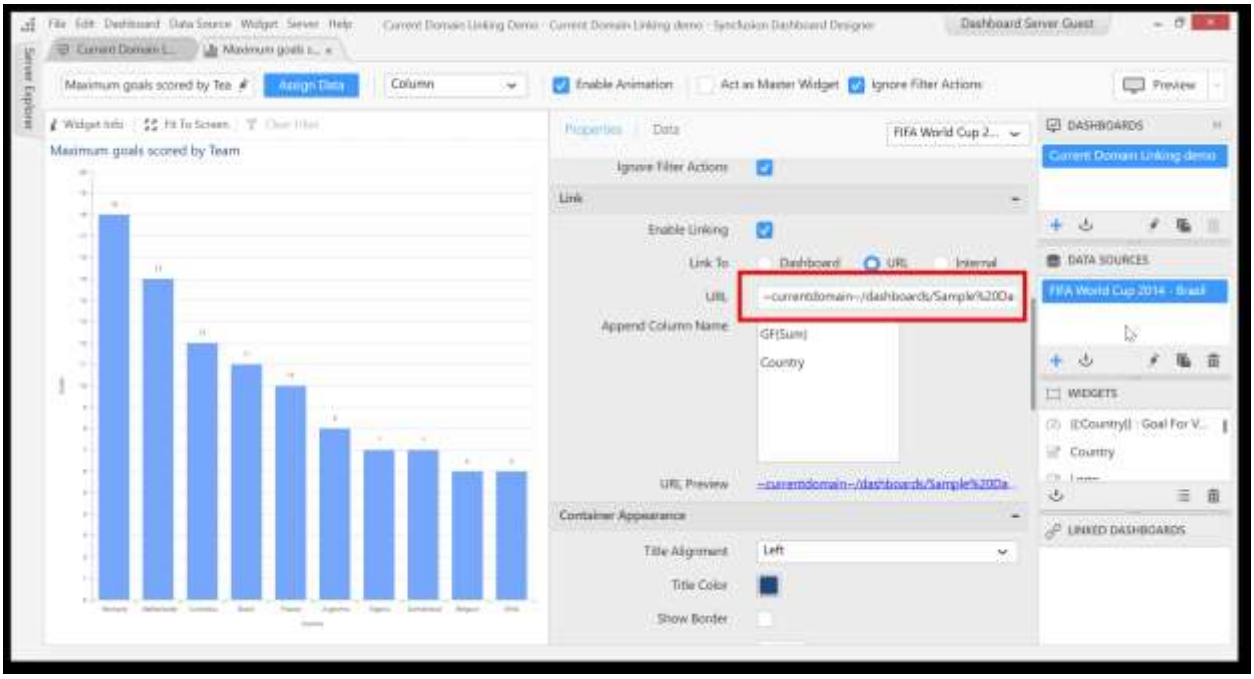
For an example,

Actual URL: http://domain.com/reports

Linking URL: ~currentdomain~/reports

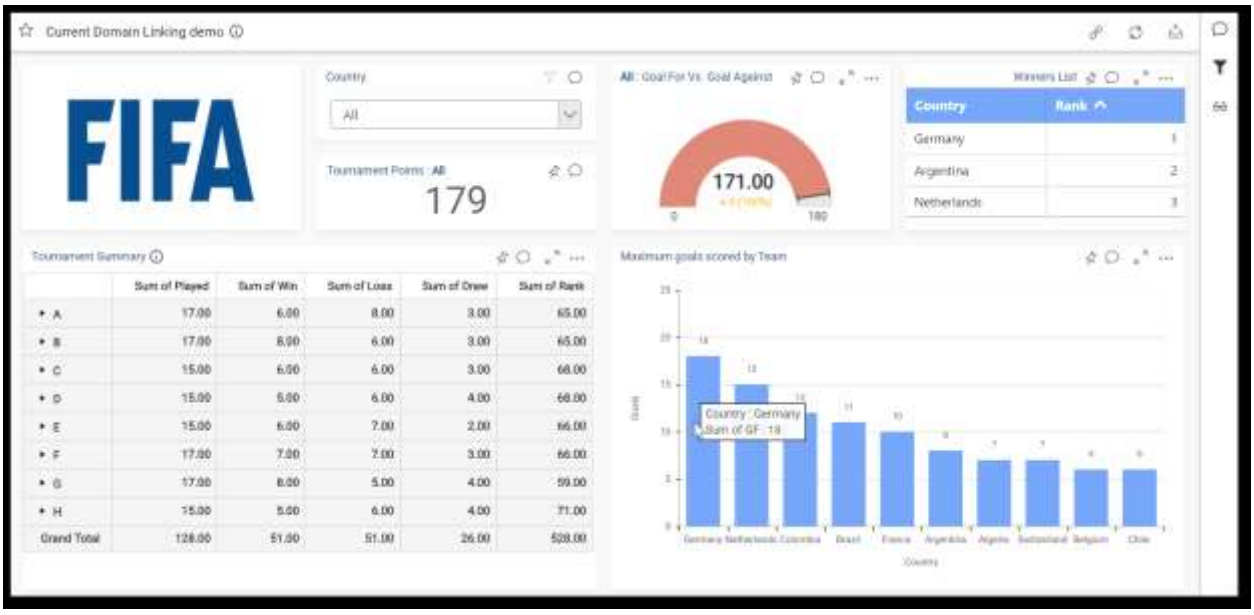
Dashboard designer view:

In the FIFA world cup 2014 dashboard ,another dashboard URL is linked in the below shown widget using the current domain linking.

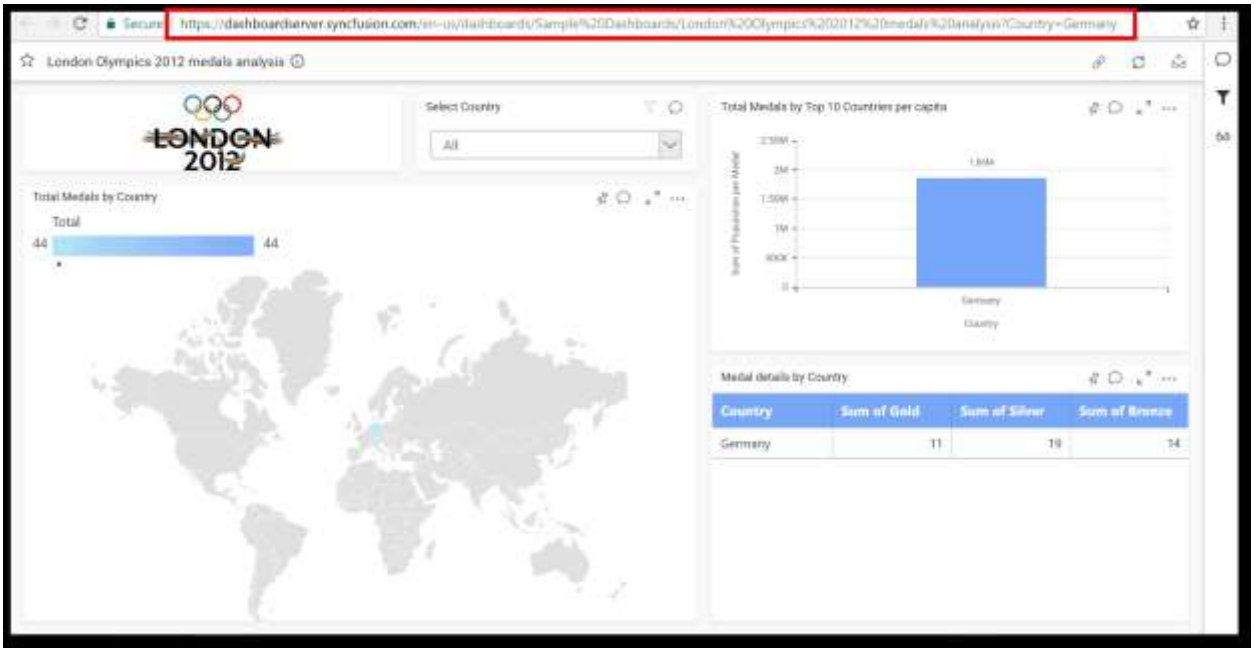


Dashboard server view:

To verify the dashboard linking, preview the dashboard.



If the Country name “Germany” is selected in the linked widget, it will navigate to the Target dashboard with the filter applied for the selected Country name.

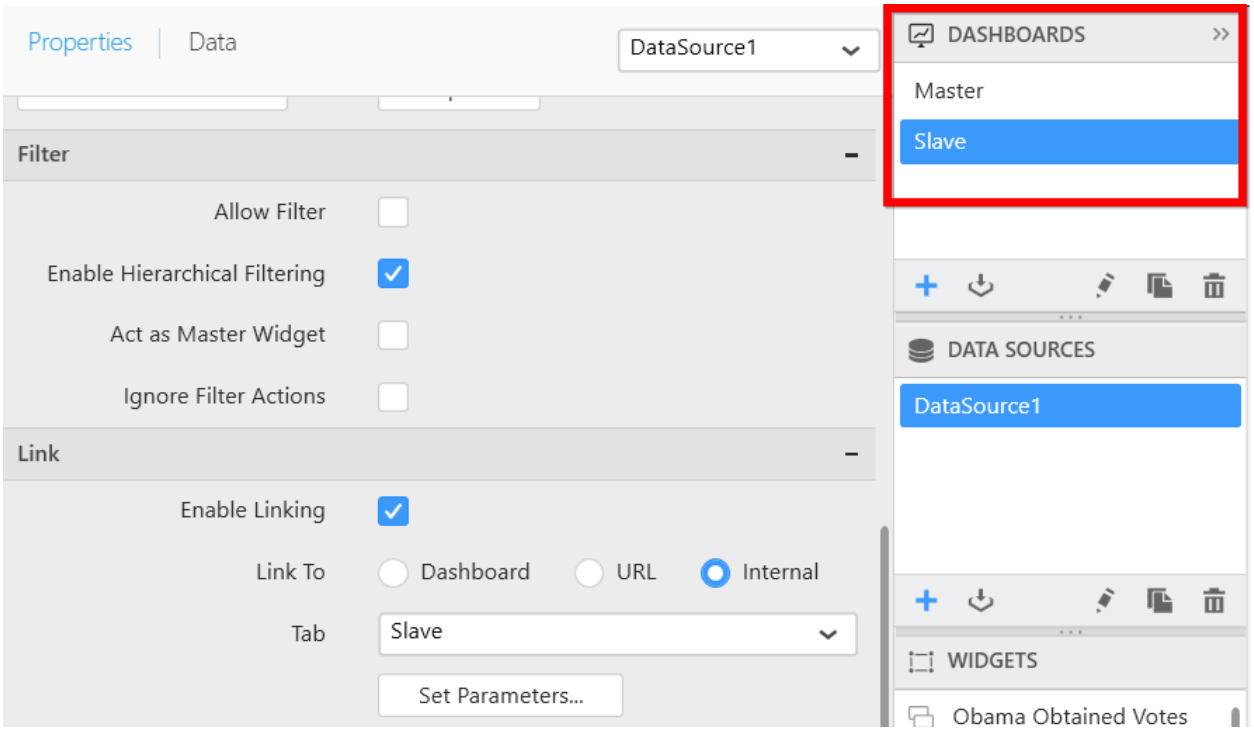


Note: The "**~currentdomain~**" keyword is case insensitive. You can use like ~CurrentDomain~, ~CURRENTDOMAIN~, etc.,

Information: If the Dashboard server user is not logged in the designer application, the current domain URL link will be considered as an invalid URL.

Internal

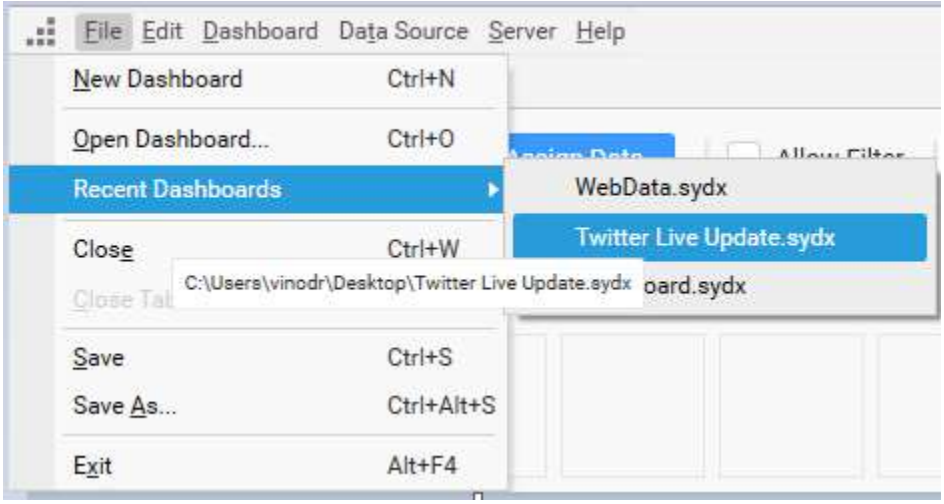
You can link to another tab of current dashboard using the Internal linking option. It will list out the dashboards other than the current active dashboard in the Tab drop-down list.



Now the set parameters option will be enabled. The Set Parameters... option will work as explained in 'Dashboard(External)' linking type.

Recently used Dashboard

You can open the recently opened/created dashboard using Recent Dashboards menu option in File menu. This list can show up to 10 recently used dashboards. The physical path will be shown in the tooltip for each dashboard.

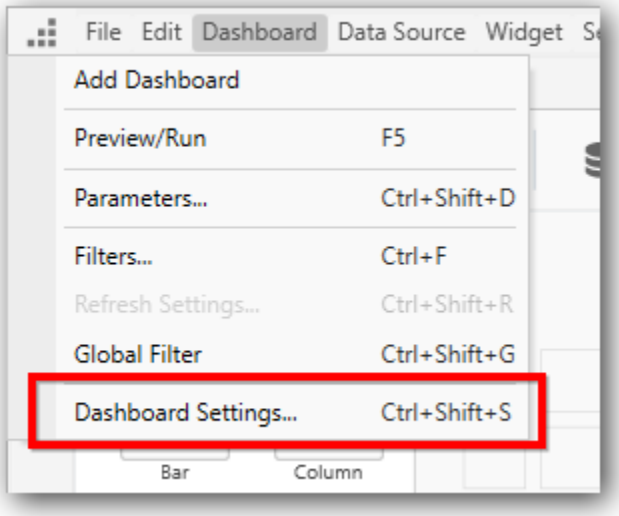


Commenting Dashboard and Widget

Syncfusion Dashboard Designer allows you to enable commenting over a Dashboard and/or its individual widgets by users when published to the Dashboard Server. You can toggle this setting through the Dashboard Settings option exposed in the application menu or Enable Commenting option in the toolbar for dashboard and in widget properties window for individual widgets.

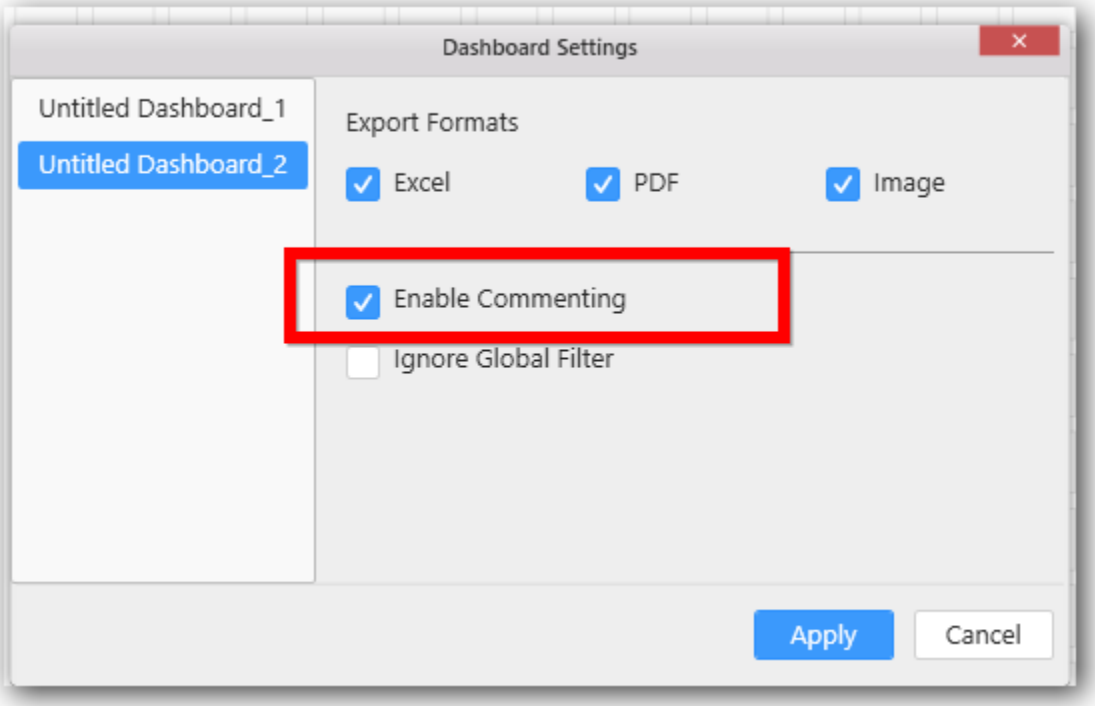
Commenting a Dashboard

To enable commenting a dashboard, navigate to the Dashboard menu and select Dashboard Setting

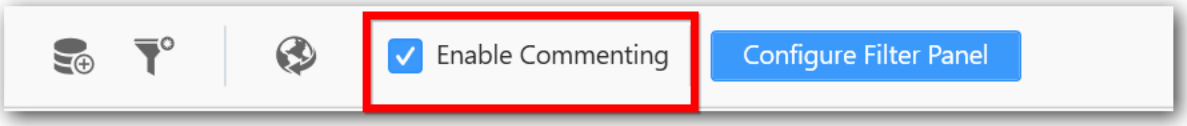


Click on the Dashboard Settings... menu item to launch the Dashboard Settings window. This can be also achieved by using the keyboard shortcut Ctrl+Shift+S.

Here, you can enable the commenting option for each dashboard and click Apply to save the settings for your dashboard.

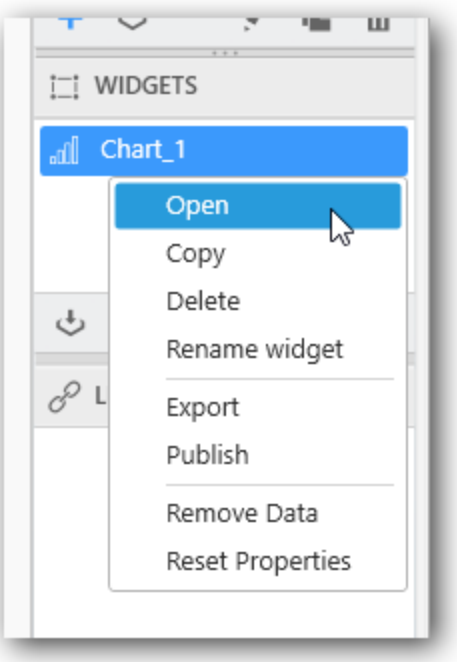


You can also enable commenting a dashboard by navigating to the dashboard tab and select Enable Commenting option in the toolbar.

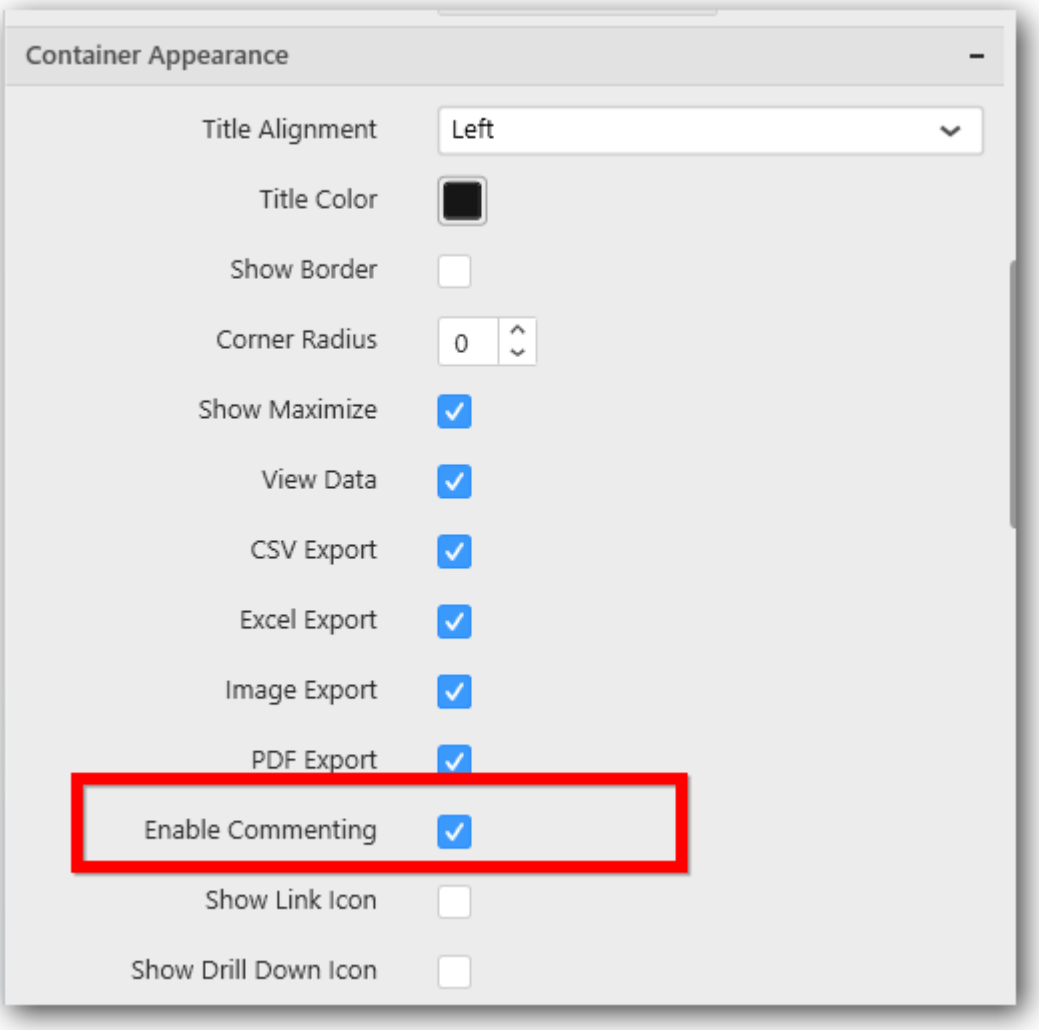


Commenting a Widget

To enable commenting for a widget, select that widget in the WIDGETS container window and either double click it or right click to open the context menu and select Open to open the selected widget in a separate tab.



After the widget tab has been opened, navigate to the properties pane and then select the Enable Commenting option under Container Appearance section.



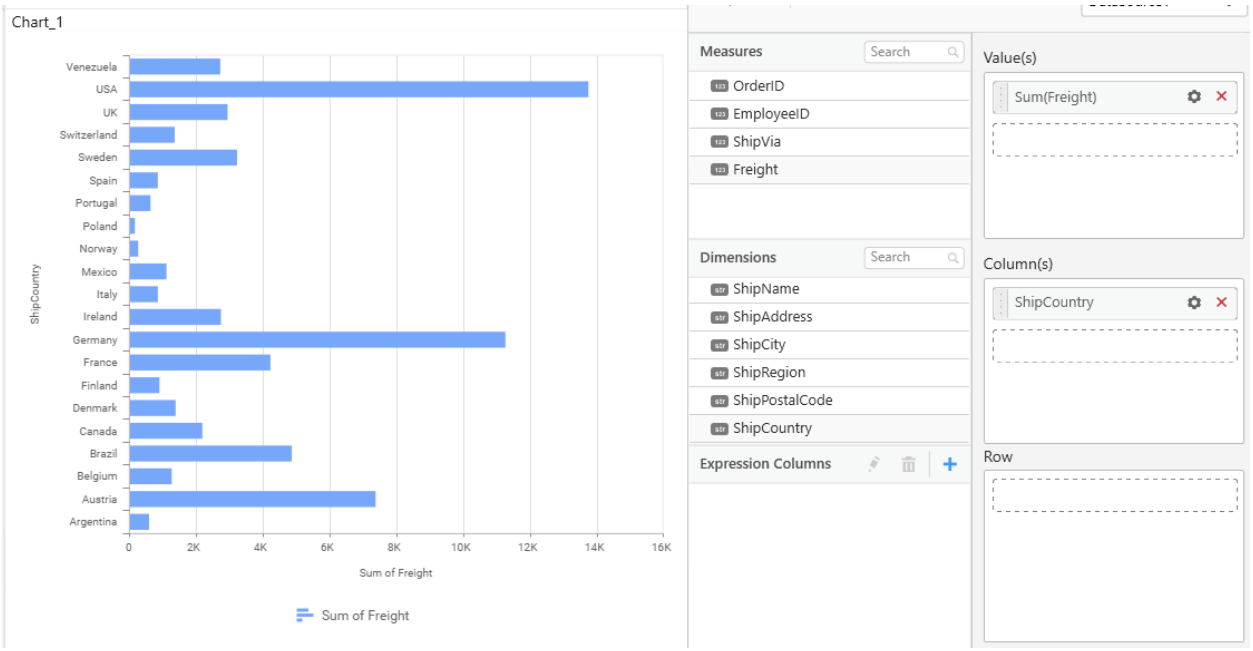
Note: Image, Label and RangeNavigator widgets do not have Commenting support.

You may find the commenting procedure here.

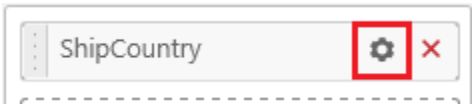
Advanced Sorting

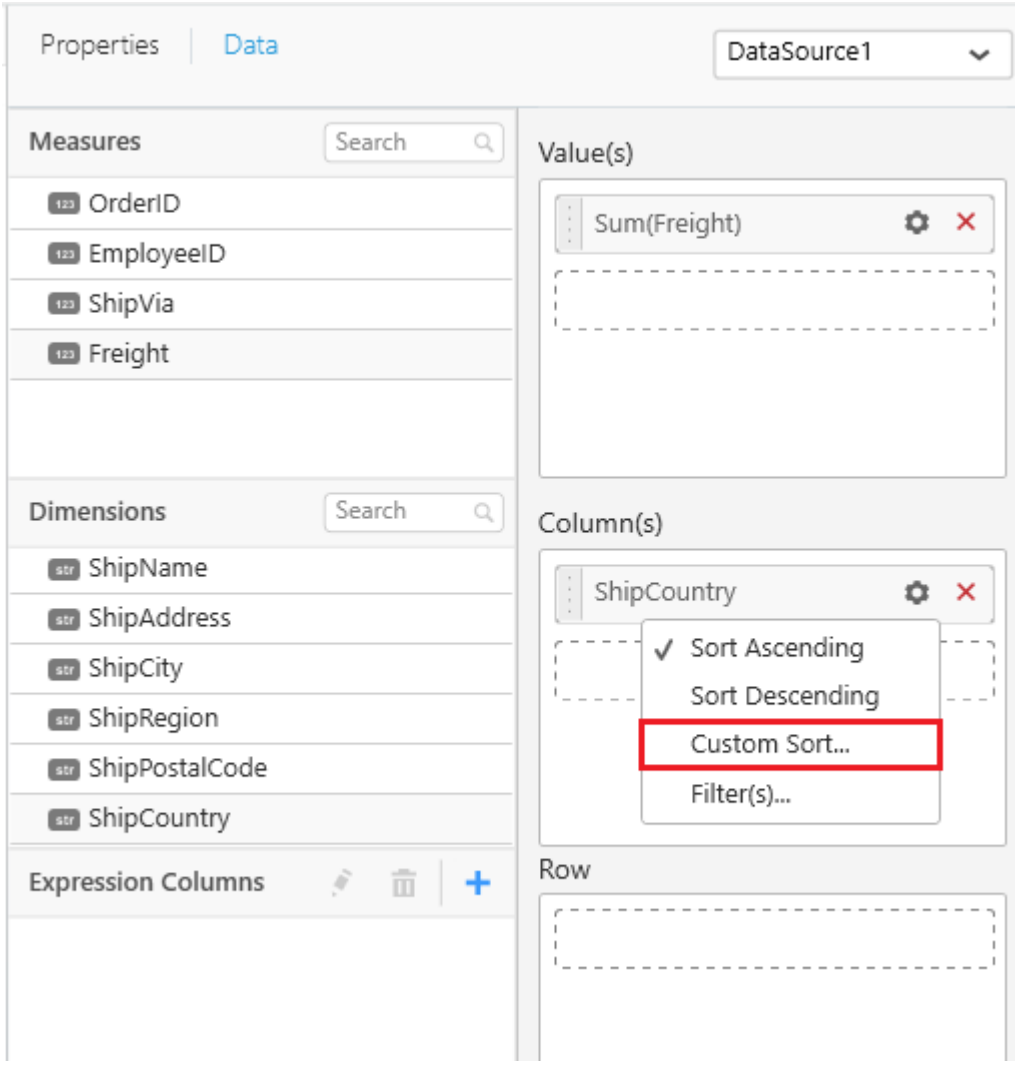
You can customize the sorting behavior of dimension fields in each widget based on your needs. You can order them based on Alphabet, Data Source (Default), Field or Manually.

To do so, first drag and drop the dimension field into control designer.

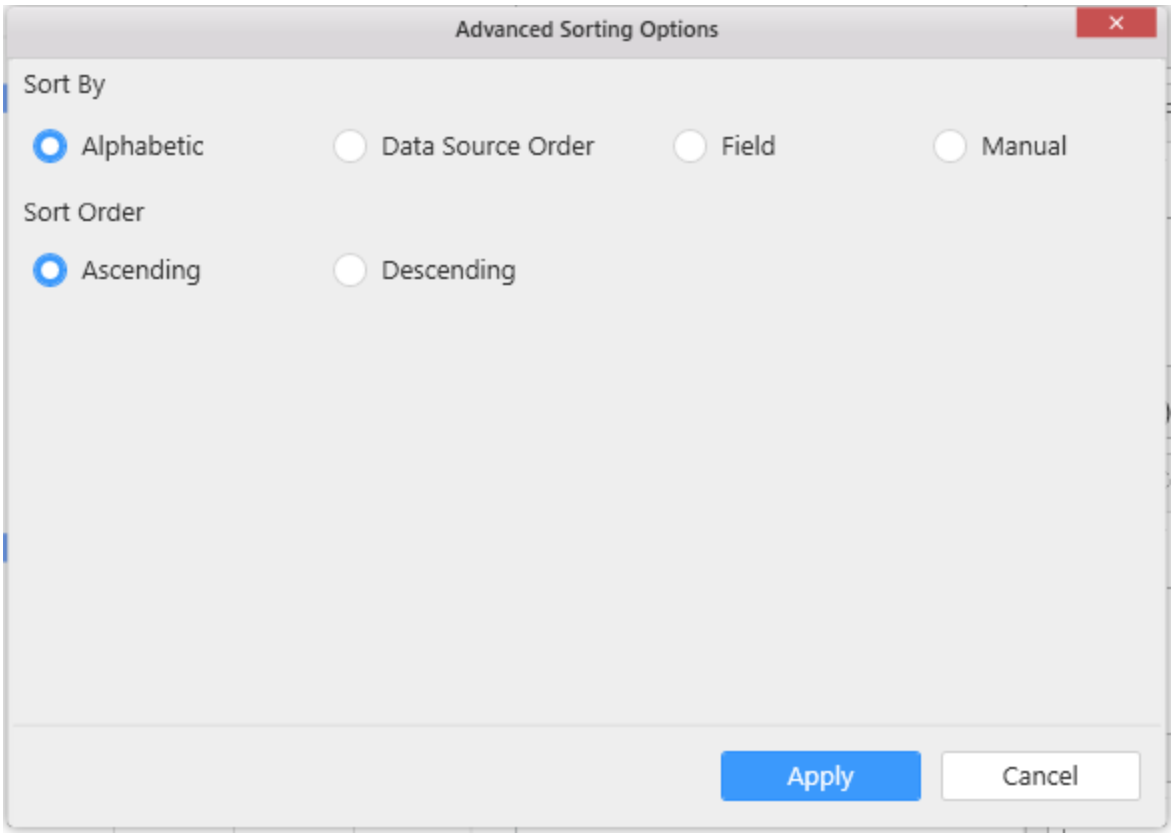


Click on the Settings icon available in the button and select Custom Sort option from the context menu.





Now you can see the Advanced Sorting Options dialog opened and look like in the below image.



Let's now see the options available in Advanced Sorting Options dialog briefly.

Sort Order

- Ascending - Displays the sorted results in ascending order.
- Descending - Displays the sorted results in descending order.

Sort By

- Alphabetic – Orders the data based on initial alphabet, either in ascending or descending. You can apply this sorting for more than one string field and during this scenario, we will order the data in hierarchical pattern as shown below.

Grid_1

Sum of Freight	ShipCountry	ShipCity
598.58	Argentina	Buenos Aires
1,186.11	Austria	Salzburg
6,205.39	Austria	Graz
821.23	Belgium	Charleroi
458.91	Belgium	Bruxelles
2,677.83	Brazil	Sao Paulo
1,685.27	Brazil	Rio de Janeiro
194.71	Brazil	Resende
322.38	Brazil	Campinas
9.92	Canada	Vancouver
793.95	Canada	Tsawassen
1,394.22	Canada	Montréal
448.85	Denmark	Kobenhavn
947.34	Denmark	Århus

- Data source - Places the resultant data from the data source, on query execution, as such without performing any additional operations like ascending or descending as shown below.

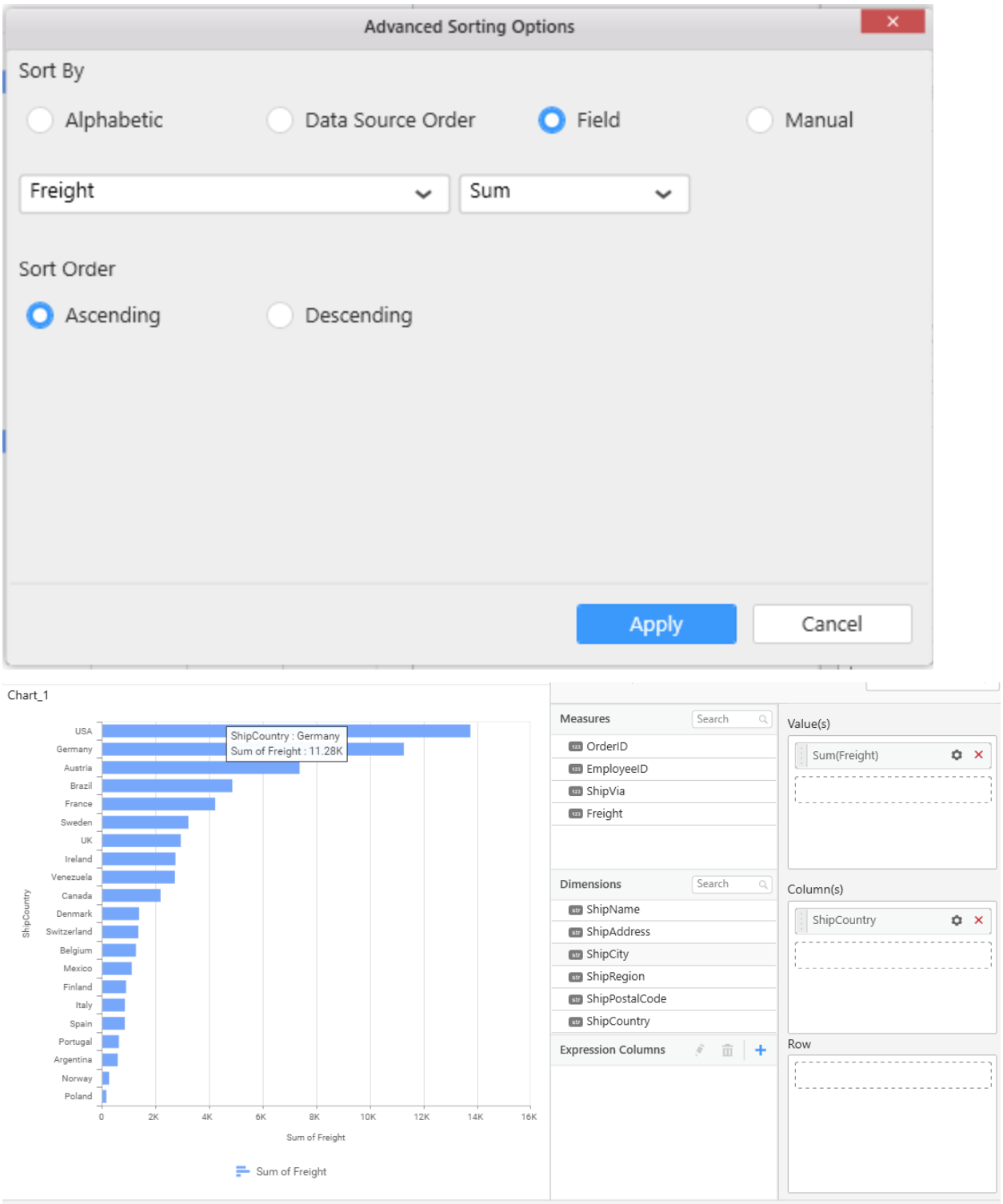
Grid_1

ShipCountry	ShipCity
France	Reims
Germany	Münster
Brazil	Rio de Janeiro
France	Lyon
Belgium	Charleroi
Switzerland	Bern
Switzerland	Genève
Brazil	Resende
Venezuela	San Cristóbal
Austria	Graz
Mexico	México D.F.
Germany	Köln
USA	Albuquerque

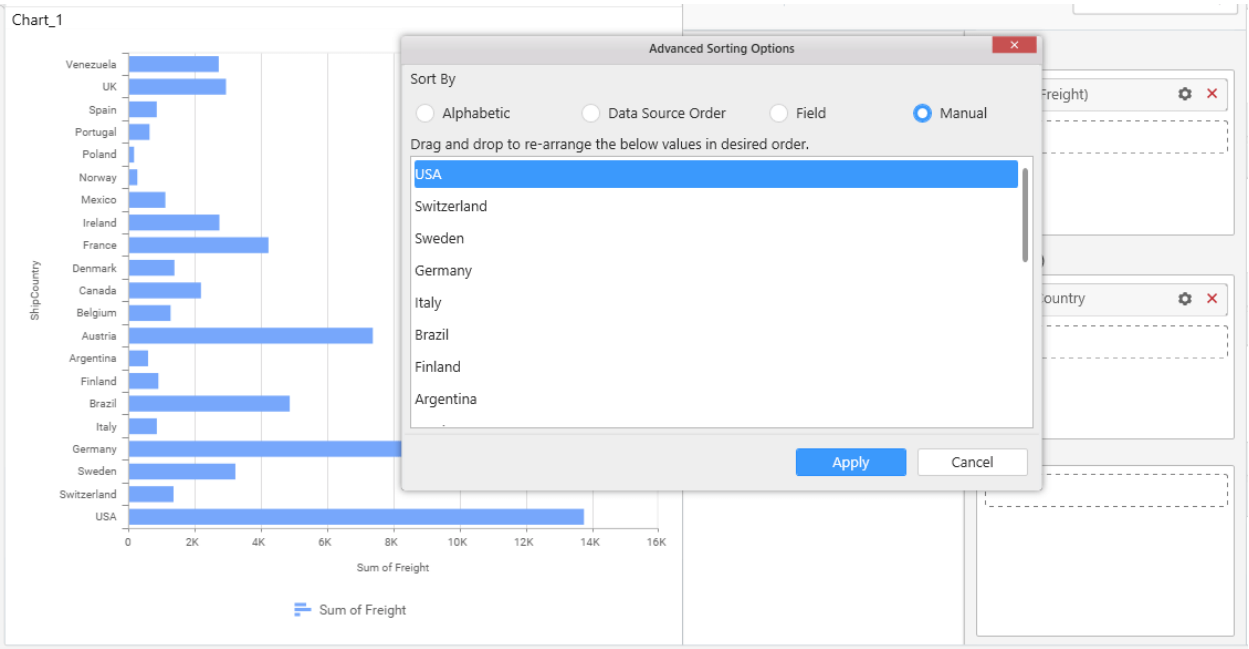
- Field - Orders the data based on the associated values of another measure or dimension field. For example, you could order several countries based on their their freight values.

When sorting with another field option, you must also specify the aggregation function to use. This option is not applicable for SSAS data sources because aggregations are defined when the cube is created and cannot be modified.

For example, the sort dialog box shown below is configured to sort the "Ship Country" field in ascending order and by the sum of the "Freight" measure. The results will be displayed in such a way that the "Ship Country" with lowest "Freight" value is displayed first and the "Ship Country" with the second lowest "Freight" value is displayed second, and so on.



- Manual - Manual sorting allows you to re-arrange the values extracted from a table column through simple drag and drop operation as shown below. You can apply this sorting for more than one columns. This sorting is not applicable for SSAS data sources.

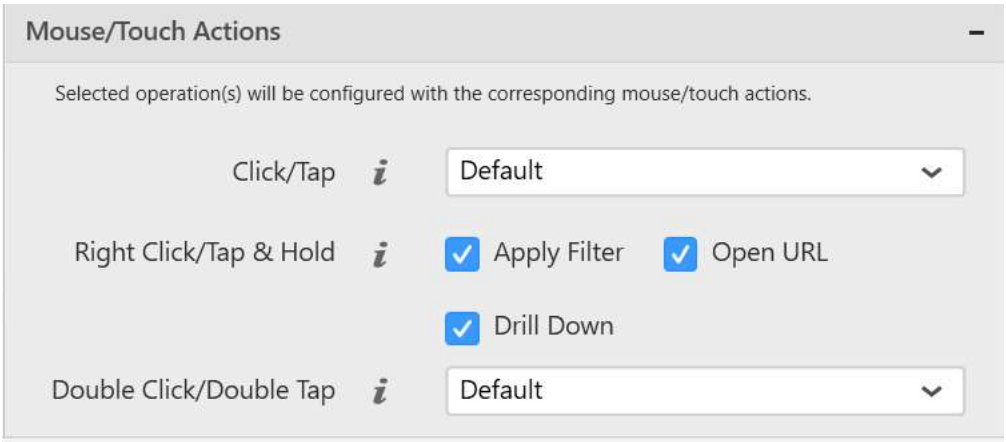


Note: As of now, Manual sorting is not available for "Date" type, "Date Time" type. Custom Sorting Option is not available for Raw Data, and proportional charts like Pie, Doughnut, Pyramid and Funnel.

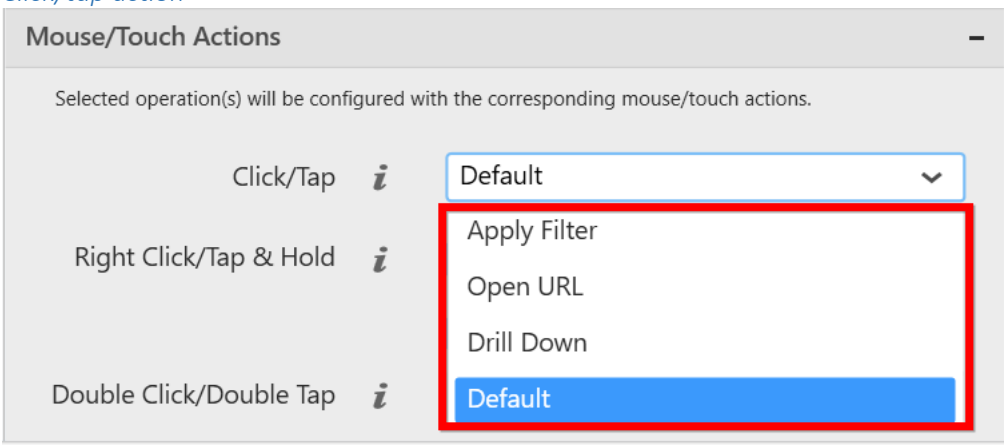
Mouse/Touch Actions

The Mouse/Touch Actions option allows choosing actions that are needed to occur when interacting with the dashboard in the dashboard viewer.

This option is available under the Widgets Properties Tab.



Click/tap action



You can choose an operation to be occurred while clicking the widget data in dashboard viewer when previewing the dashboard.

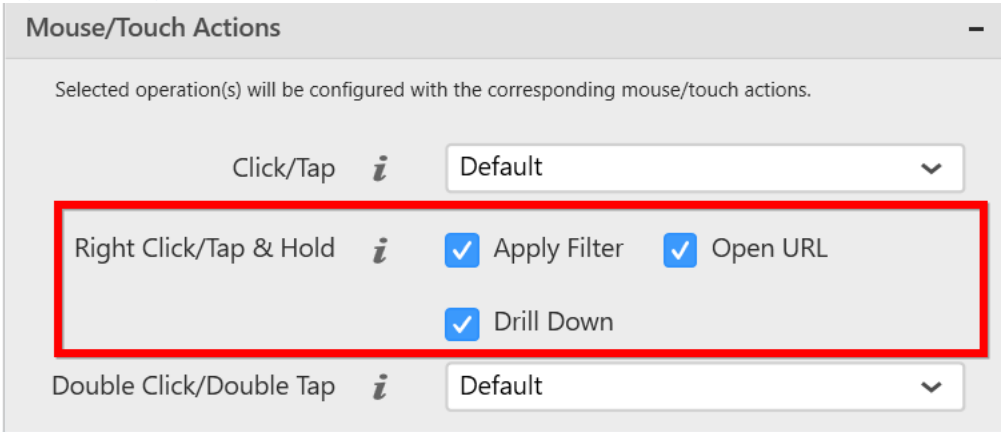
Master filter: The corresponding slave widgets can be filtered based on the selection made in the master widget.

Hyperlink: The mentioned URL or dashboard opens.

Drill down: If the widget supports drill down, the drill down will be configured and performed.

Default: Default action occurs for the user interactions.

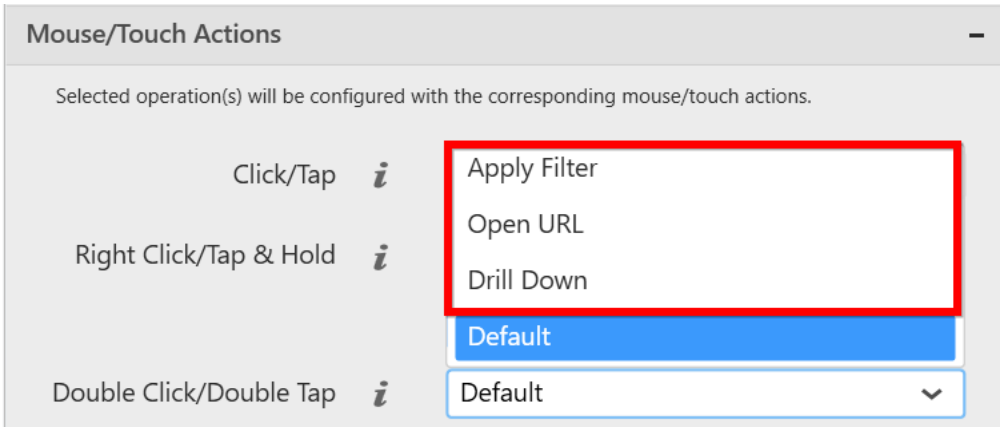
Right click/tap and hold



You can select more than one option in the Right click/Tap & Hold option. The selected items will be shown in the right-click context menu of the dashboard viewer using which you can perform the required action while interacting with dashboard in the dashboard viewer.

Double click/double tap

You can choose the actions need to be occurred for **Double click** or **Double Tap** user interaction. The available options will be same in the Single Click/Tap option. So, you can choose one among the four options such as master filter, hyperlink, drill down, and none.



Default behavior

The Default behavior priorities order for Click and DoubleClick actions in Widget Action Settings are updated in the table.

Actions	Default Click action	Default Double click action
Apply Filter && Open URL && Drill Down	Apply Filter	Drill Down
Open URL && Drill Down	Open URL	Drill Down
Apply Filter && Open URL	Apply Filter	None
Apply Filter && Drill Down	Apply Filter	Drill Down
Apply Filter	Apply Filter	None
Open URL	Open URL	None
Drill Down	Drill Down	Drill Down

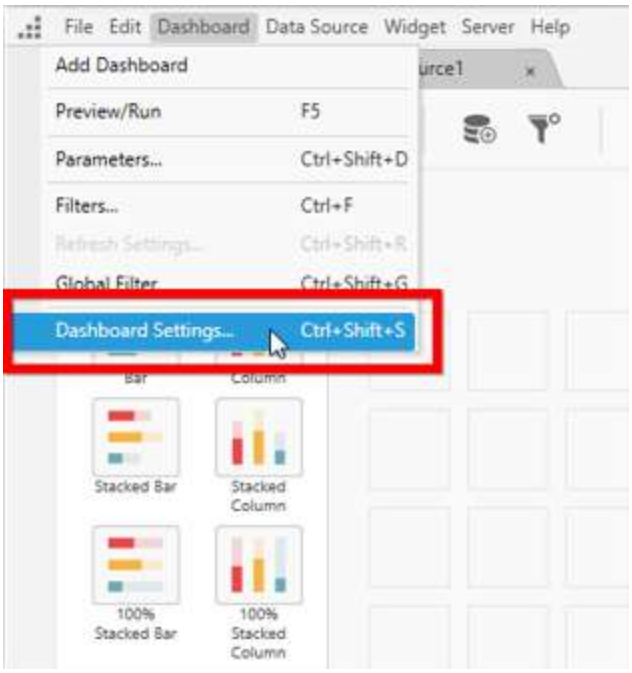
Note: The Mouse/Touch Actions option is introduced in the Dashboard Designer version 3.1.0.113.

Dashboard Settings

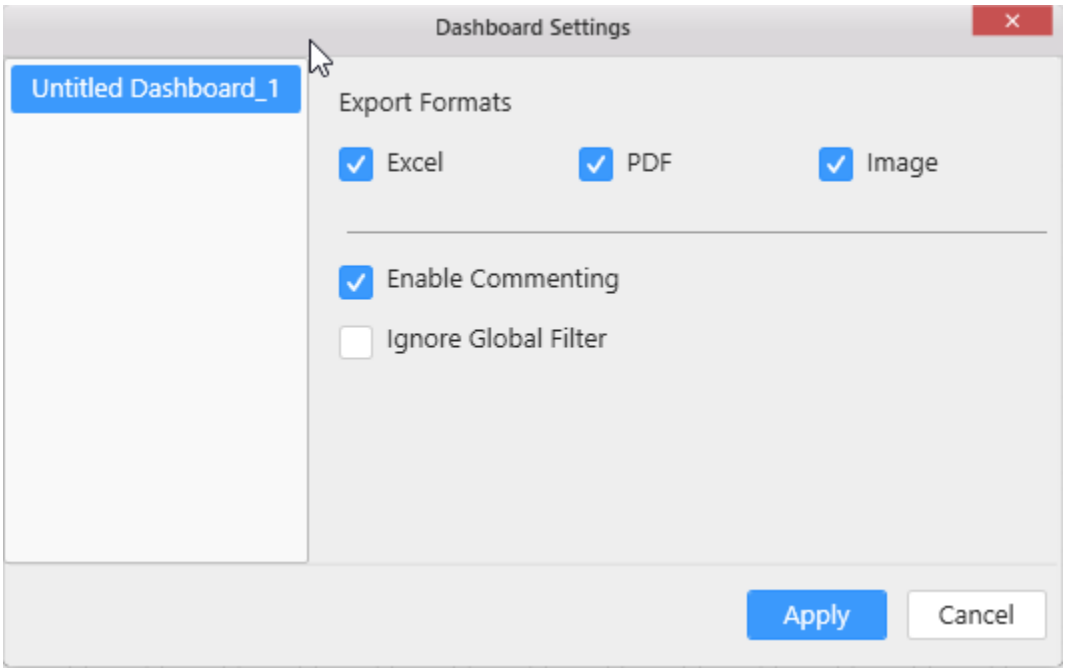
You can configure the Dashboard specific settings for each dashboard using the Dashboard Settings option. The following options are available in the Dashboard settings:

- Export Dashboards
- Enable commenting
- Ignore Global Filter

To open the Dashboard Settings, click the **Dashboard** menu and choose the **Dashboard Settings...** option. This can also be achieved by using the keyboard shortcut **Ctrl+Shift+S.**



By default, the Dashboard Settings window opens as follows.



Export formats

You can export the selected Dashboard to Excel, PDF, or image format in the Dashboard Preview or Dashboard Server.

Enable commenting

You can allow or restrict the commenting over the selected Dashboard after publishing in the Dashboard Server by using the Enable Commenting option.

Click here to learn about Commenting in the Dashboard Server.

Ignore global filter

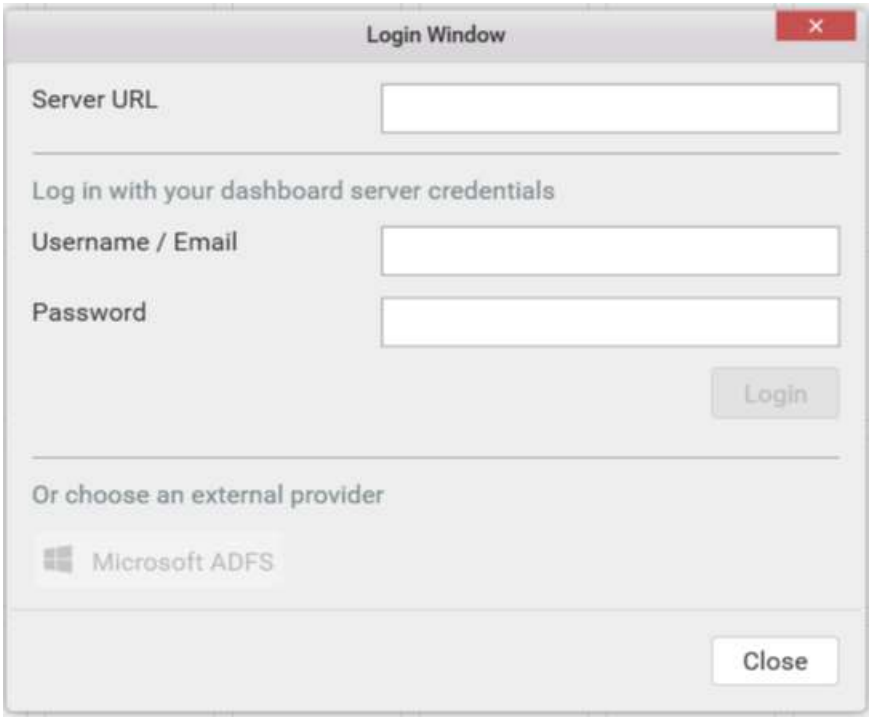
You can allow or restrict the Global filter action for the selected Dashboard using the Ignore Global Filter option.

Sharing Dashboard

Connecting to a Server

Logging into Server

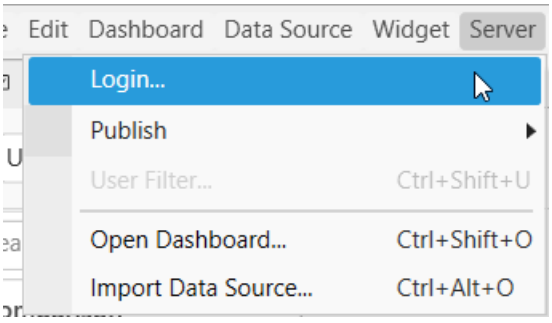
Log into the Dashboard Server through the login window.



Login window can be launched through any of the following ways.

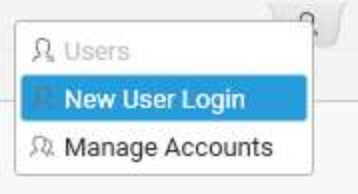
Through Application Menu

Click the Server menu and select the Login... menu item to launch the login window.



Through Title Bar

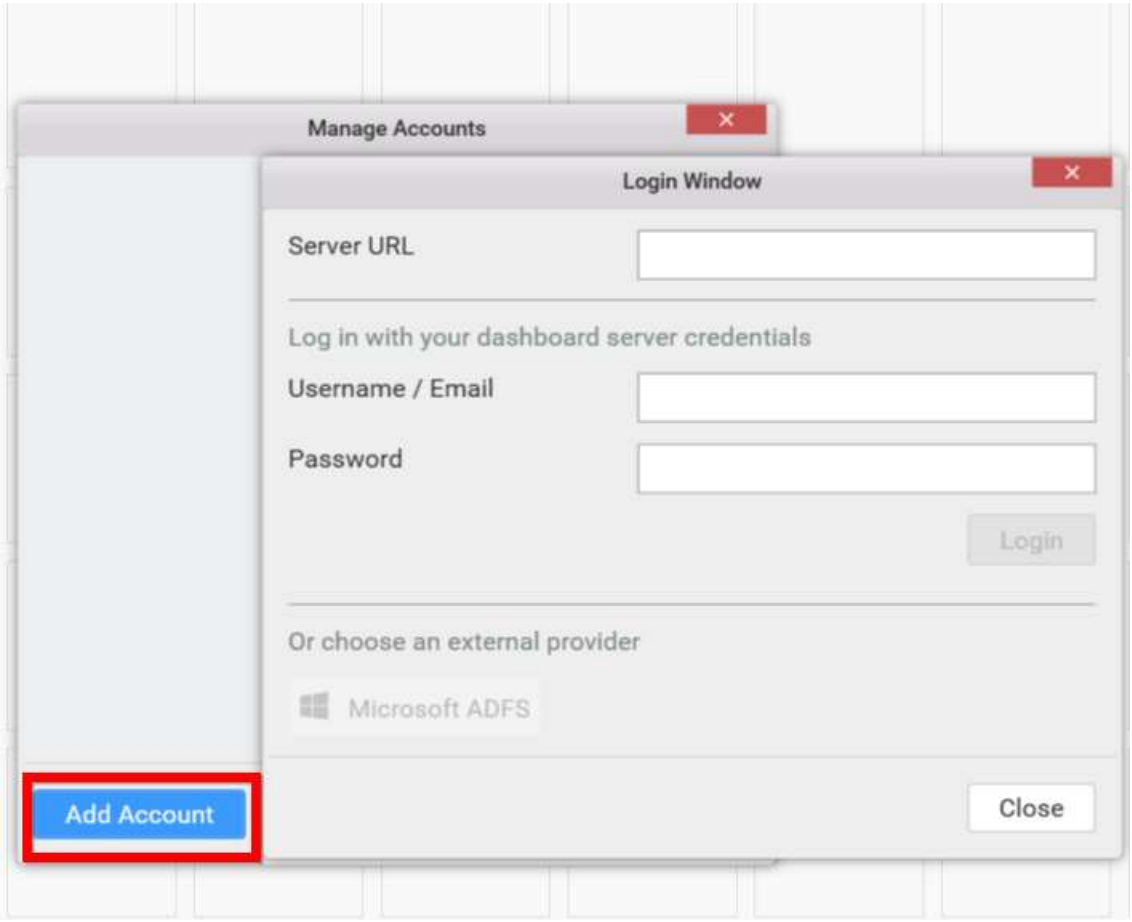
Click the Login option in the title bar and choose the New User Login menu item to launch the login window.



Through Manage Accounts

Click the Login option in the title bar and chose Manage Accounts menu item in the drop down list.

In the Manage Accounts dialog window, click Add Account to launch the login window.

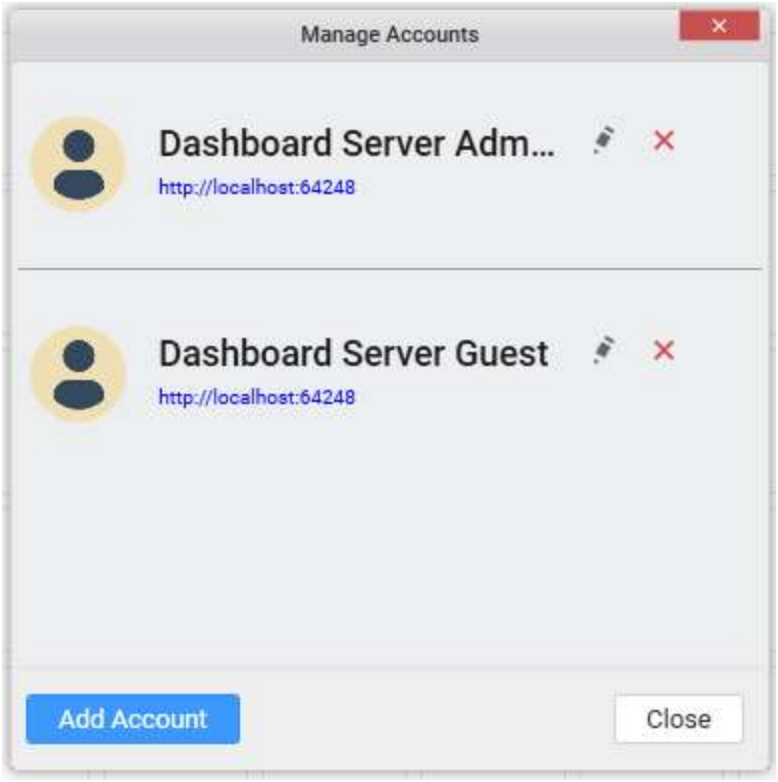


Managing Multiple User Accounts

Syncfusion Dashboard Designer provides the ability to configure multiple user accounts of same or different dashboard servers. Through this, current user login can be switched in between these accounts without requiring to feed the user account credentials every time.

Manage Accounts

Manage Accounts dialog maintain all the added user information. Each user account added can be edited and/or removed.

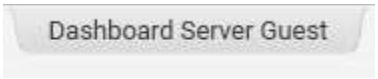


Click the Login option in the title bar and select Manage Accounts menu item in the drop down list. Now, Manage Accounts dialog window will be opened.

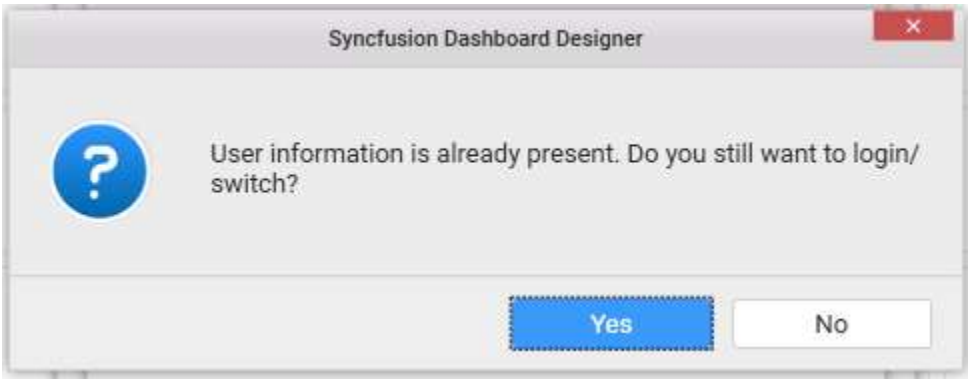
Note: Added user accounts and current login information will be maintained even Dashboard Designer upgraded to a newer version later, until removed explicitly.

Add Account

In the Manage Accounts dialog window, click Add Account to launch the login window for adding user information. If newly added user information is not present already, added user account will act as current user and Login icon changed with the current user name

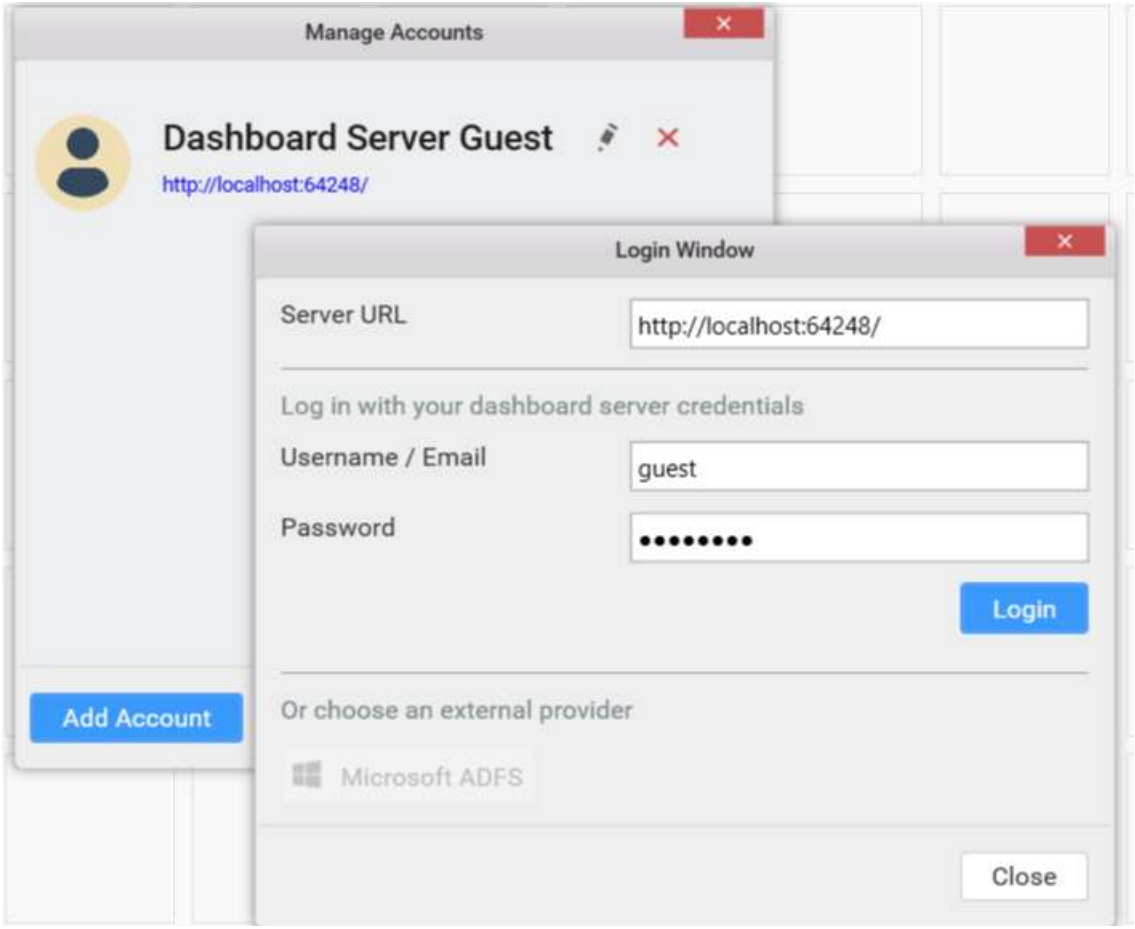


If the added user is already present in the list, show dialog box to prevent the duplicate user information are added again.



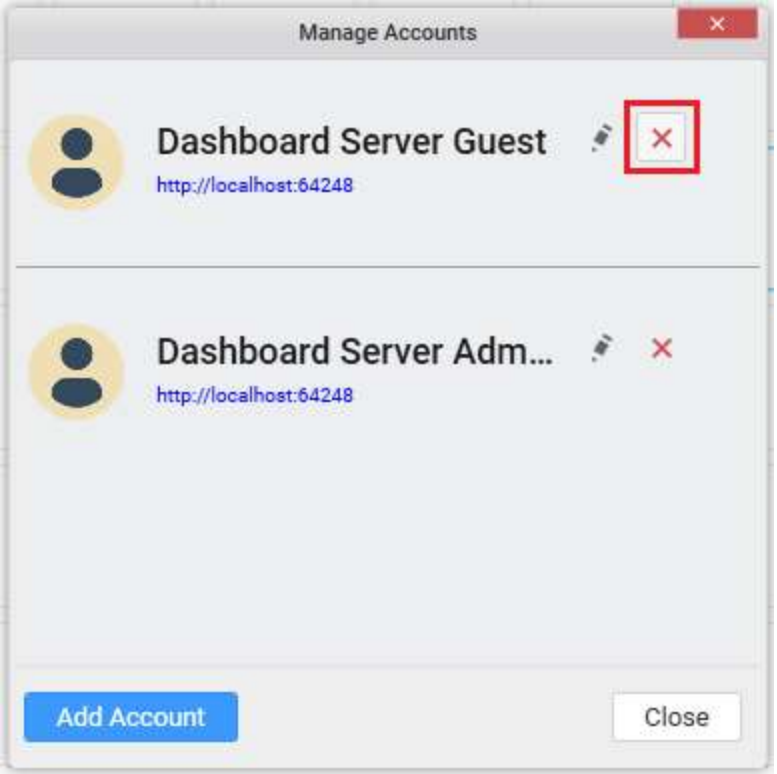
Edit Account

To edit an existing user account, navigate to the respective account in Manage Accounts dialog window and click Edit icon highlighted below. This will launch the Login Window to update the details. If provided user information is valid, login will succeed. Otherwise, fails.

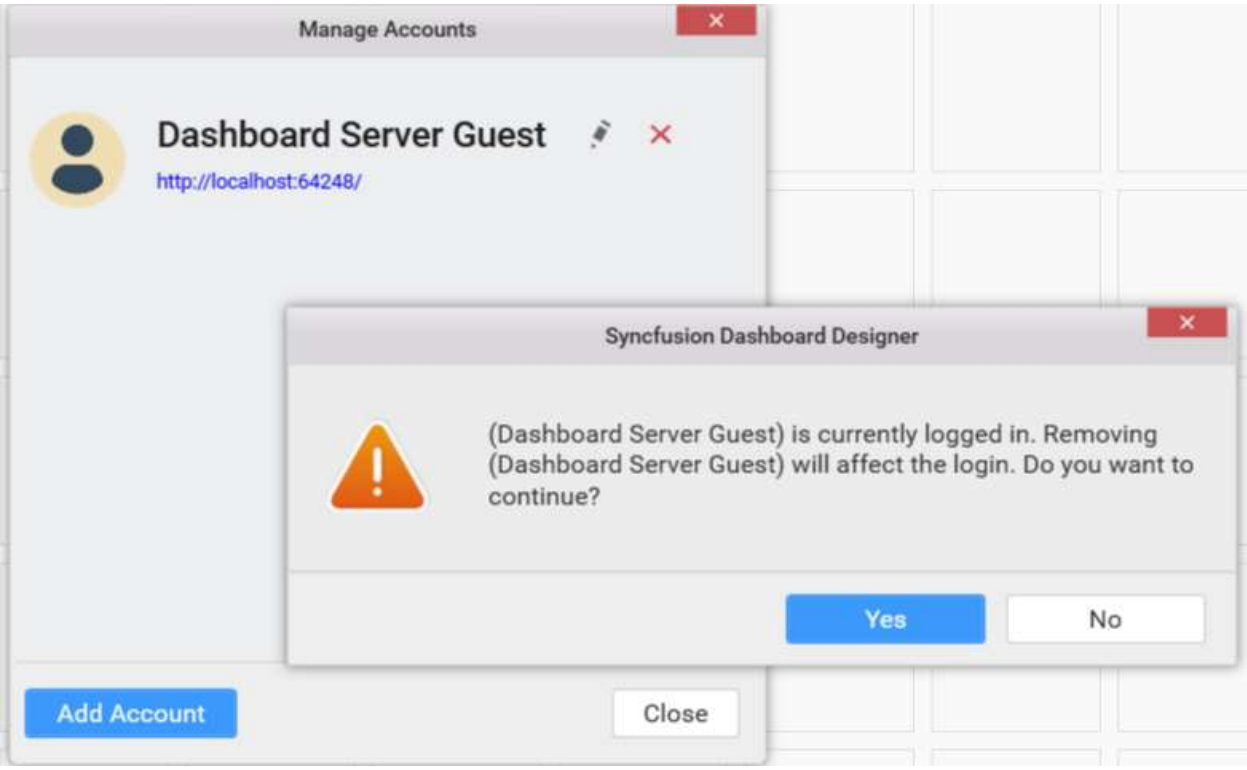


Remove Account

To remove a user account, navigate to the respective account in Manage Accounts dialog window and click Remove icon highlighted below.

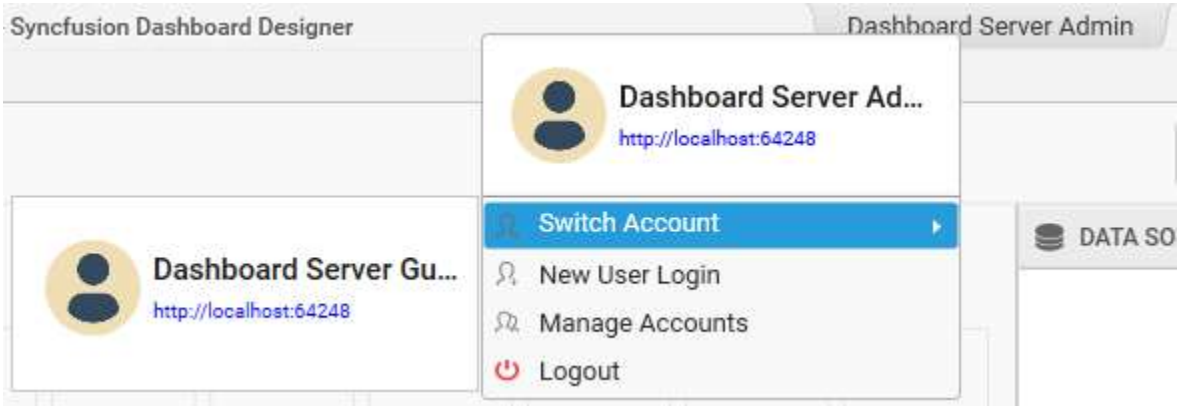


If selected user account is the one currently logged in, alert message will be displayed before removing the account from the list for confirmation.

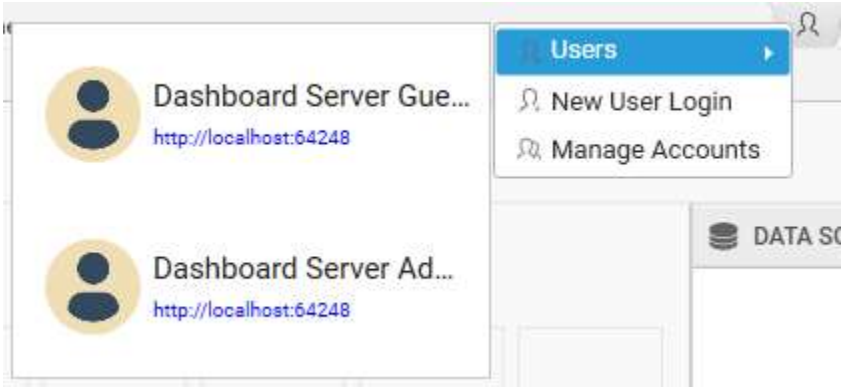


Switch Account

To switch the current login between the accounts you added already, click the Login option in the title bar and choose Switch Account option. Navigate to the added user accounts that are displayed in sub menu and select the respective account to switch to. Selected account will then be considered as Current account.



Note: When the network disconnected between designer and server, selected user sign-in will not be showed until connection is re-established. Meanwhile, message Unable to connect to the remote server will be displayed on attempting to connect When you are not logged in currently, you will get Users menu item displayed instead with the configured user accounts in sub menu.

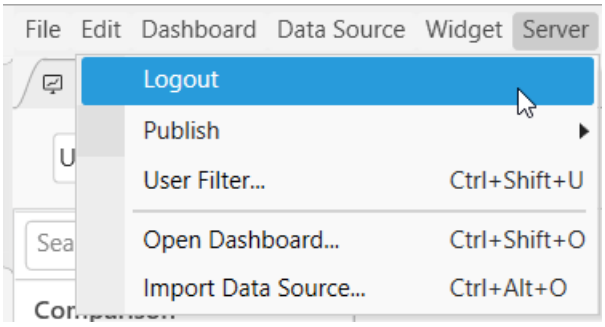


Logging out from Server

You may logout from the dashboard server through any of the following ways.

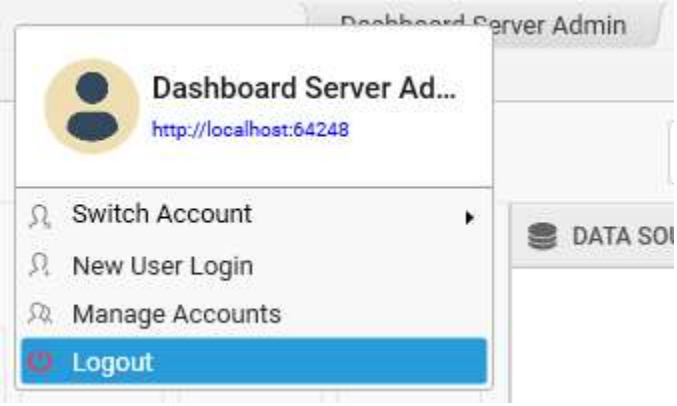
Through Application Menu

Click the Server menu and select the Logout menu item to sign-out the current user.



Through Title Bar

Click the Login option in the title bar and choose Logout option to sign-out the current user.

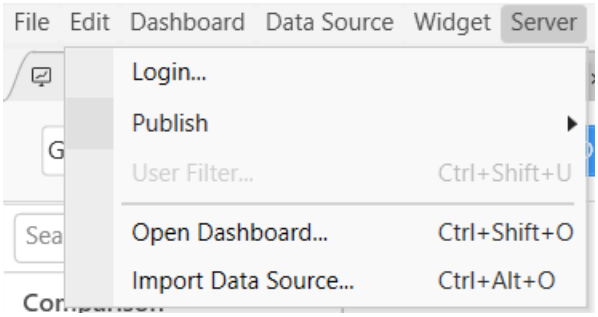


Note: After the current user logged out, the added user accounts will still be maintained under Users sub menu for re-login later.

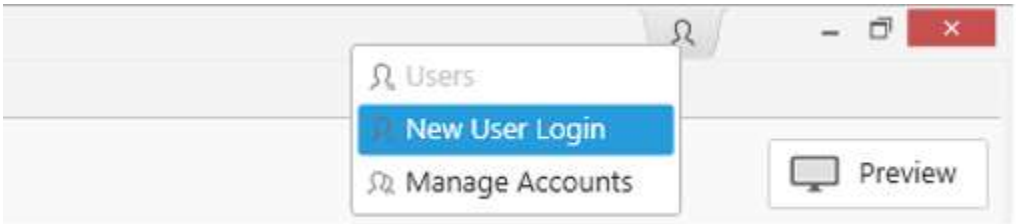
Logging into Dashboard Server from Designer using External providers

You can login to Syncfusion Dashboard Server from Dashboard Designer application using Microsoft Azure Active Directory Services (ADFS). To enable this, you need to connect to a Dashboard Server that has Microsoft ADFS login enabled. To know more about configuring Dashboard Server with Microsoft ADFS refer here.

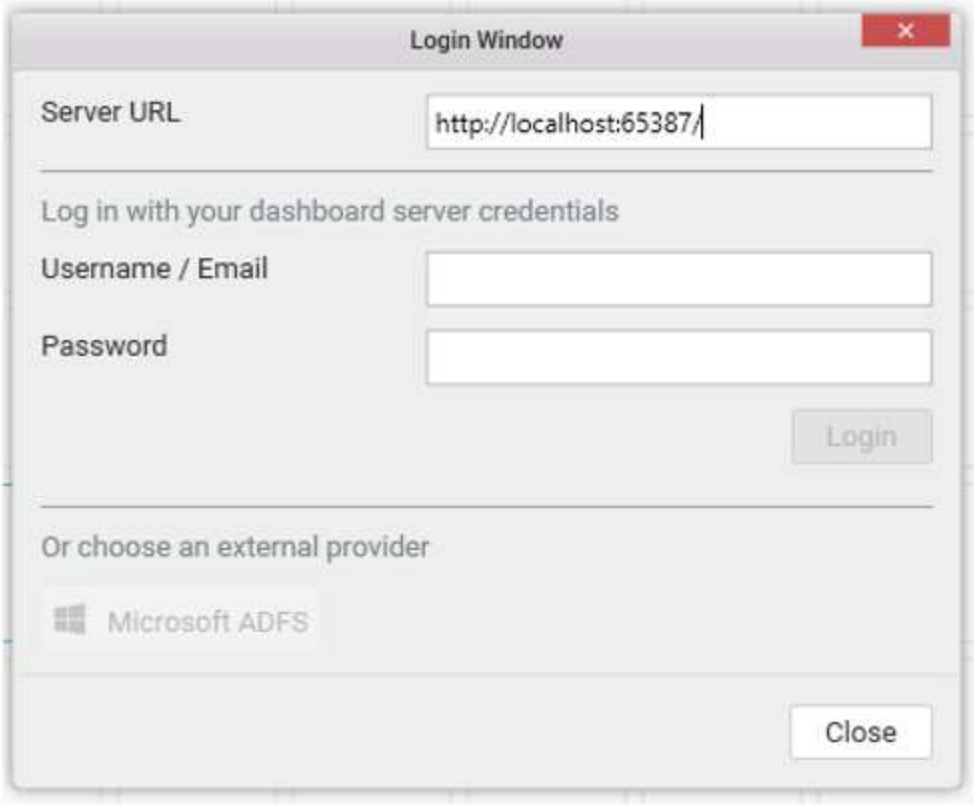
From **Server** menu click on **Login**. The Server login window will be popping up.



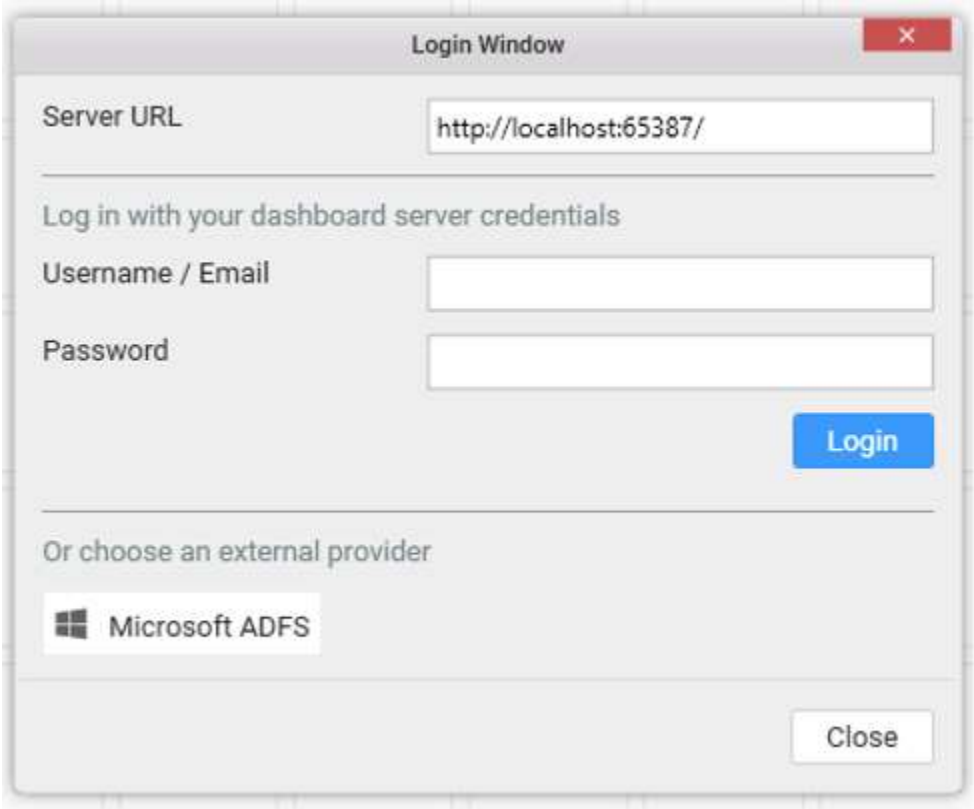
You can also navigate to login window by clicking on the User icon and from the dropped menu, click on "New User Login" menu item.



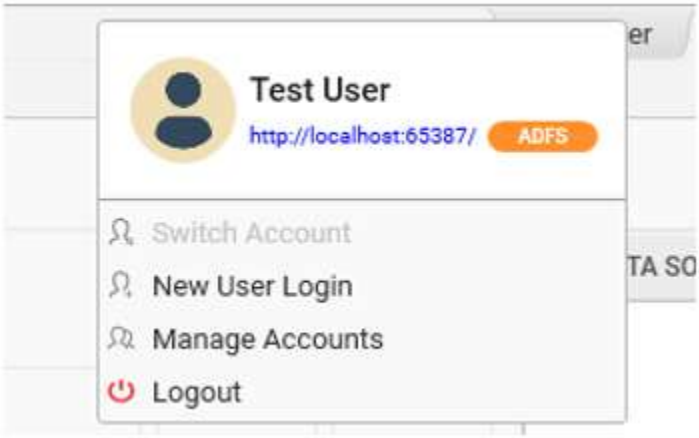
Type in the Dashboard Server URL in the Server URL field



If the given server is configured with ADFS then, the external provider login option will be enabled

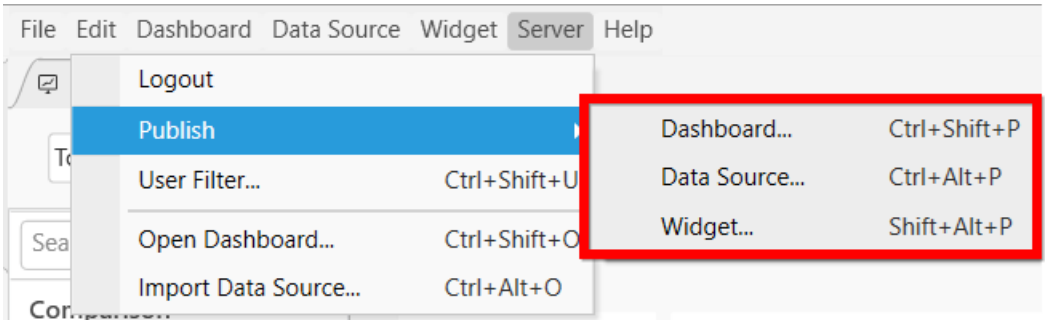


Clicking on Microsoft ADFS button will open the Azure login page and you can login with your Microsoft credentials. If the log in is successful, you will see the current user added in the available user's list.



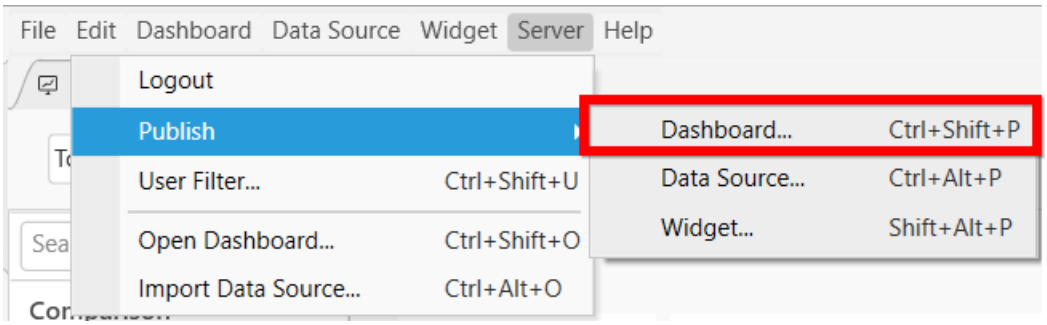
Publishing to Server

Dashboards, Data sources and Widgets can be exported to server from Dashboard Designer through the Publish option in the Application Menu.

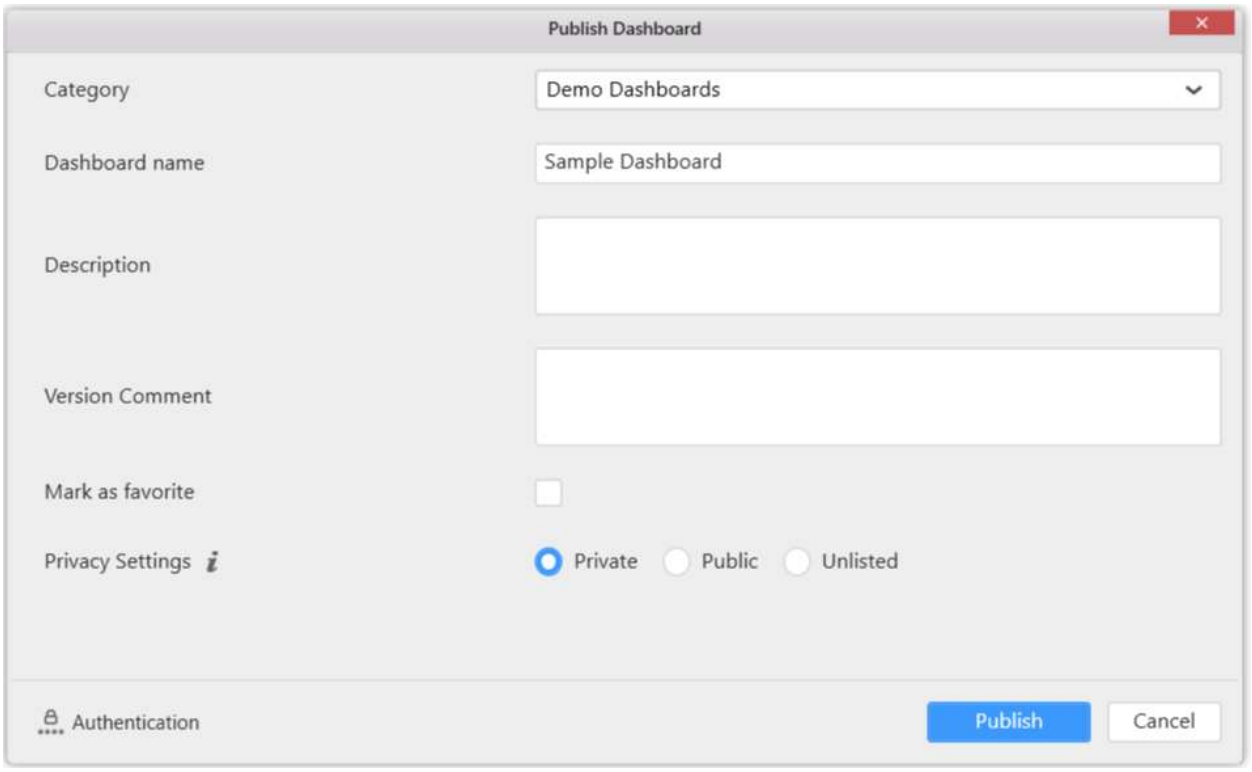


Publishing Dashboard to Server

Click Server Menu and navigate to Publish menu item. Select Dashboard... menu item.



If you logged in to Dashboard Server already, you will be prompted with the publish dialog like below.



In this dialog, Category drop down list shows the list of categories added in the connected dashboard server. This represents the category under which the dashboard need to be published. Set the dashboard name near the Dashboard name label.

Set the description, if you prefer, commenting over the dashboard, near Description area. Version Comment is for commenting over each version of publish. Mark as favorite is for marking dashboard as favorite in the dashboard server. However, these are optional.

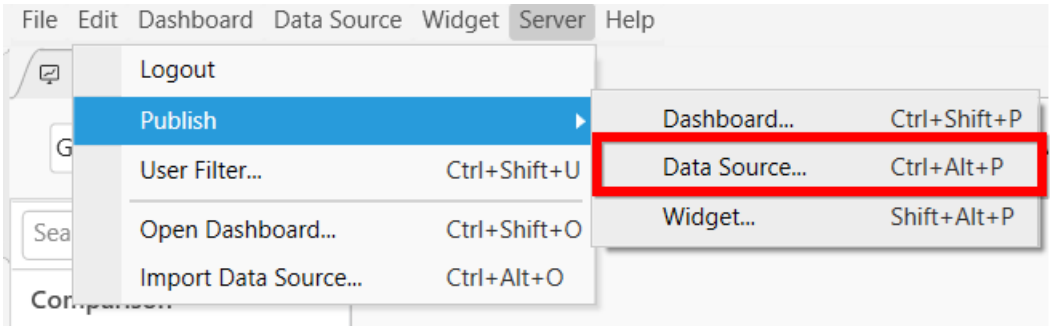
Privacy Settings is for setting accessibility for dashboard in the dashboard server. By default, Private option is selected.

Click Publish to export dashboard to dashboard server.This dashboard can now be viewed under Dashboards in the published dashboard server like below.

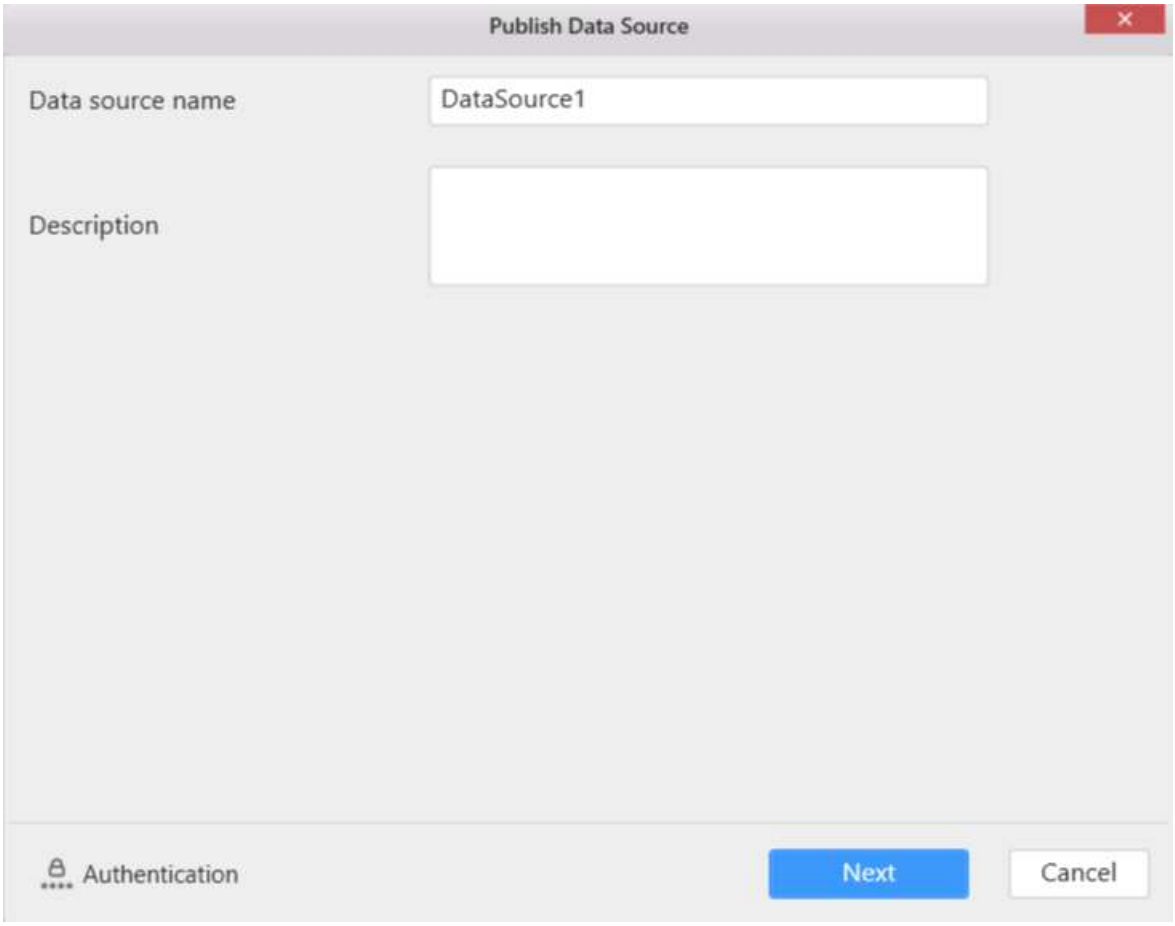
Click Cancel to cancel the export operation and close the dialog.

Publishing Data source to Server

Click Server Menu and navigate to Publish menu item. Select Data Source… menu item.



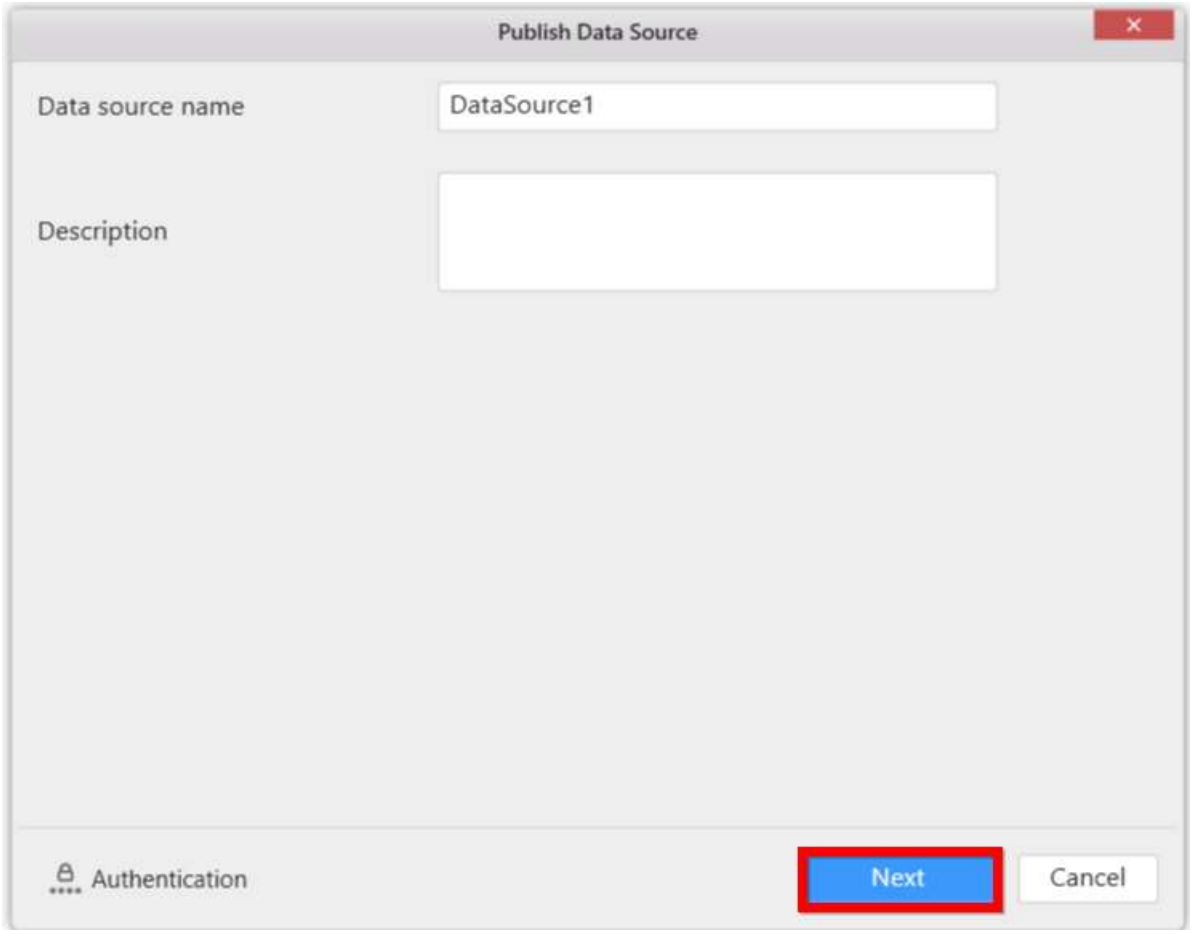
If you logged into Dashboard Server already, you will be prompted with the publish dialog like below.



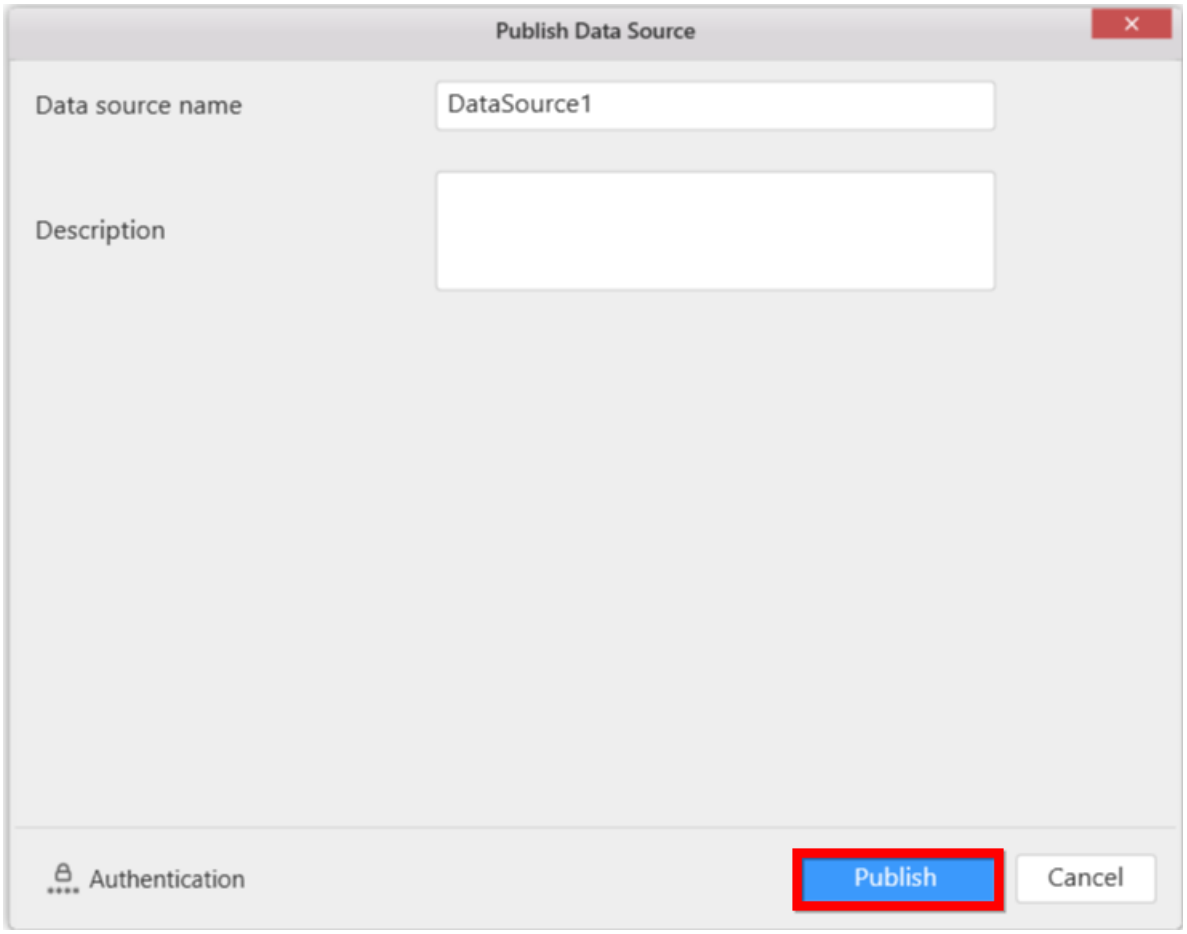
In this dialog, enter the data source name with which you require the data source to be published, in Data Source name text area.

Enter the description in the Description text area illustrating the data source, if required. This is optional.

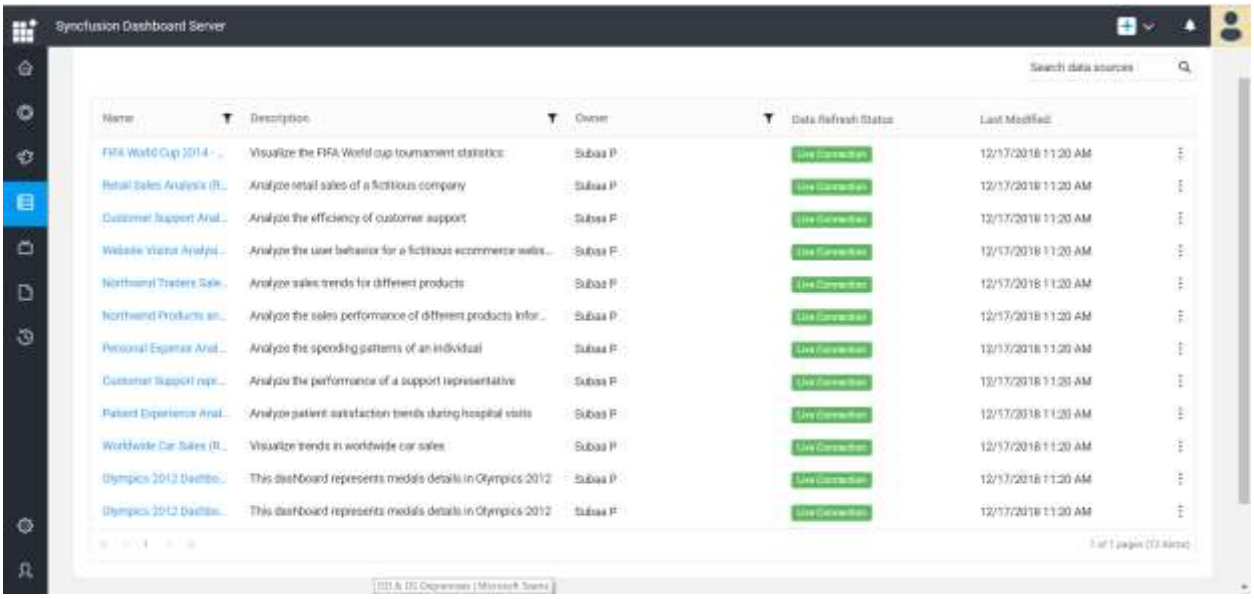
Click Next button, the publishing options dialog will be shown for exporting file type data source in the server.



Click Publish to export server type data source to dashboard server.



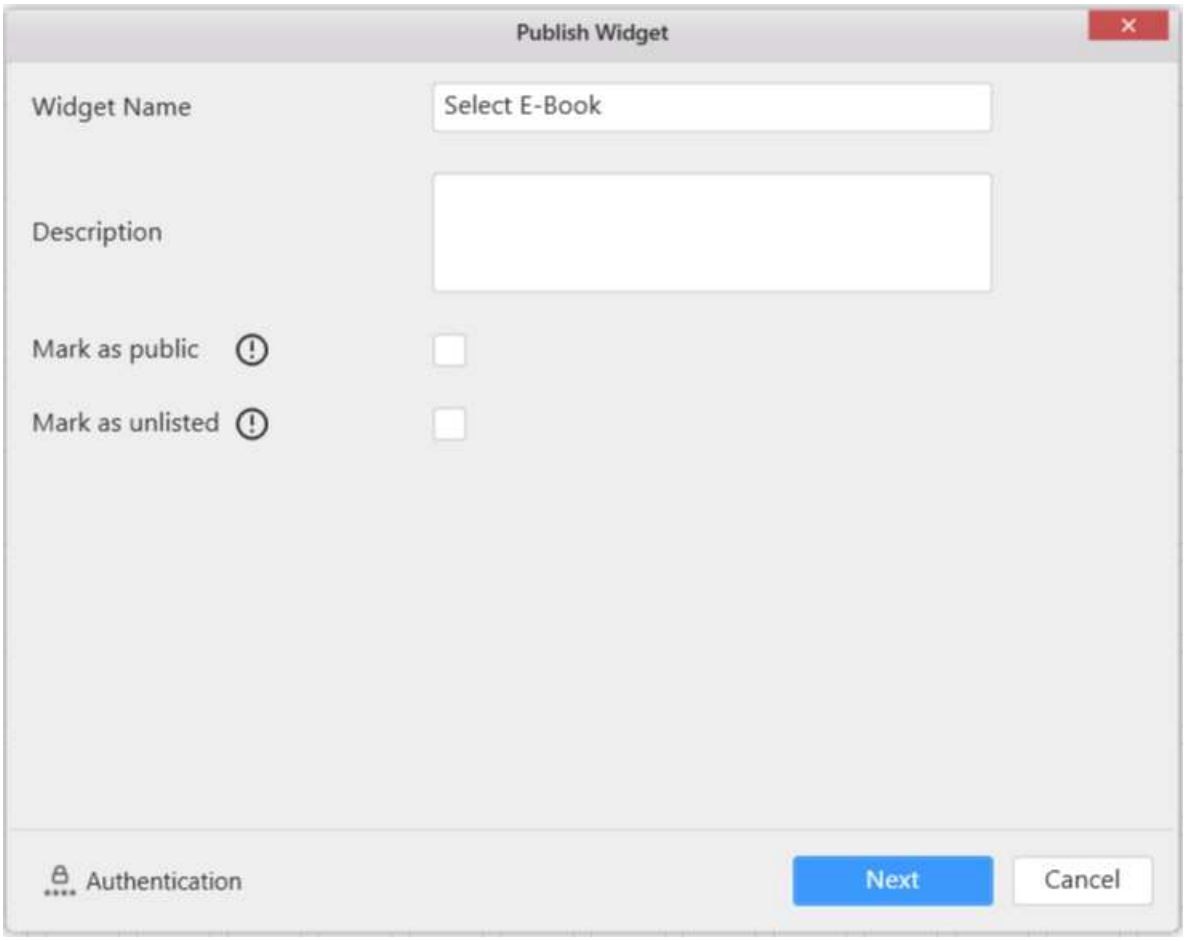
This data source can now be viewed under Data Sources in the published dashboard server like below.



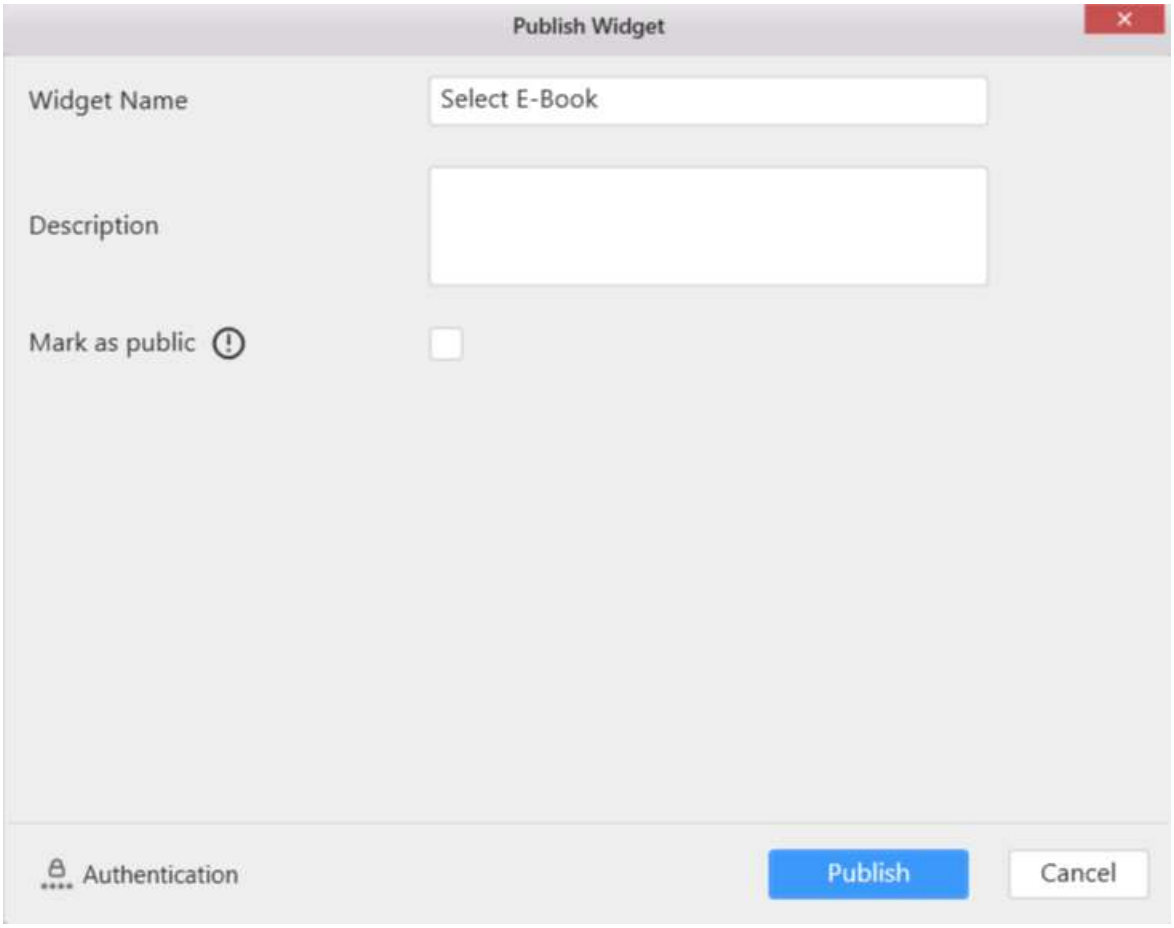
Click Cancel to cancel the export operation and close the dialog.

Publishing Widget to Server

Click Server menu and navigate to Publish menu item. Select Widget… menu item.



If you logged into Dashboard Server already, you will be prompted with the publish dialog like below.



In this dialog, enter the widget name with which you require the widget to be published/exported to server, in Widget Name text area.

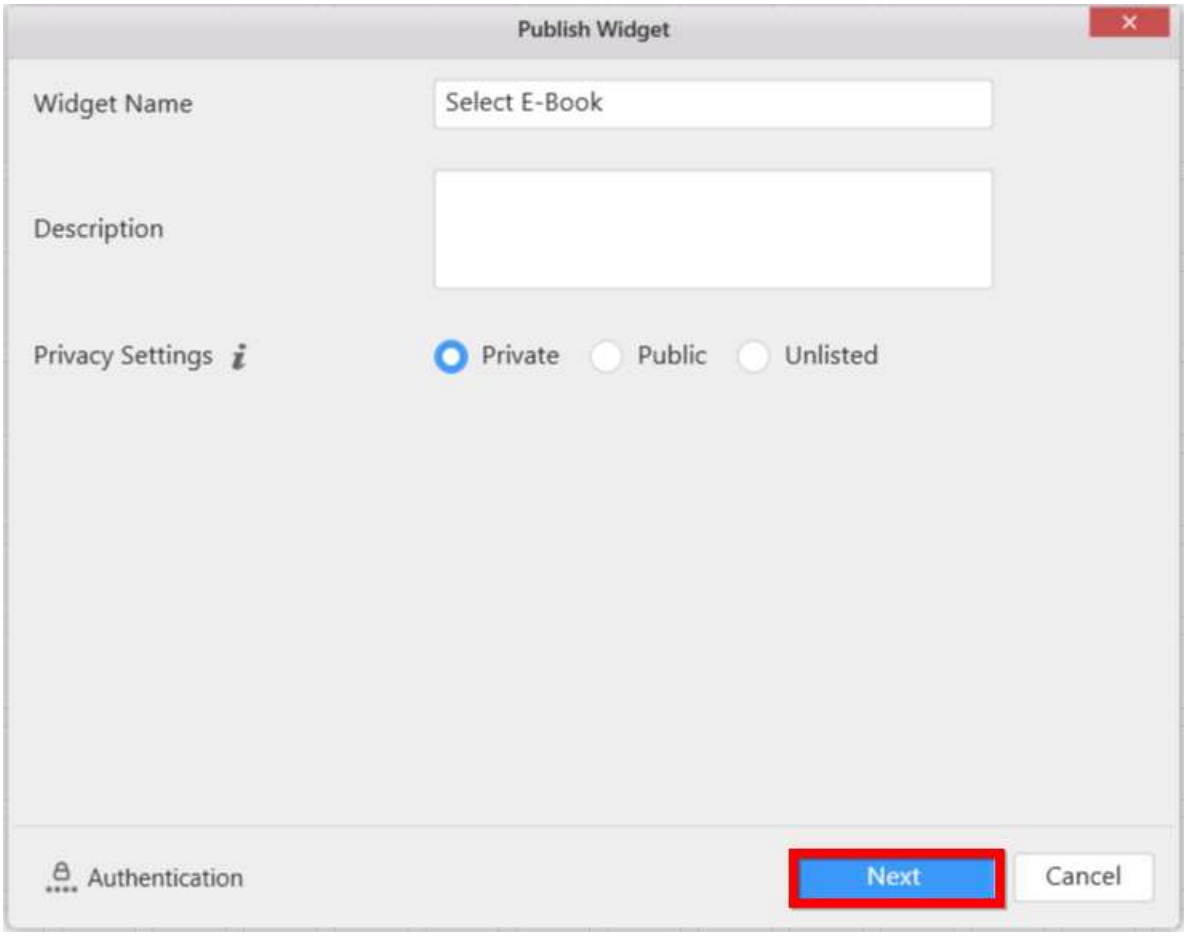
Enter the description in the Description text area illustrating the widget,if required. This is optional.

Privacy Settings is for setting accessibility for widget in the

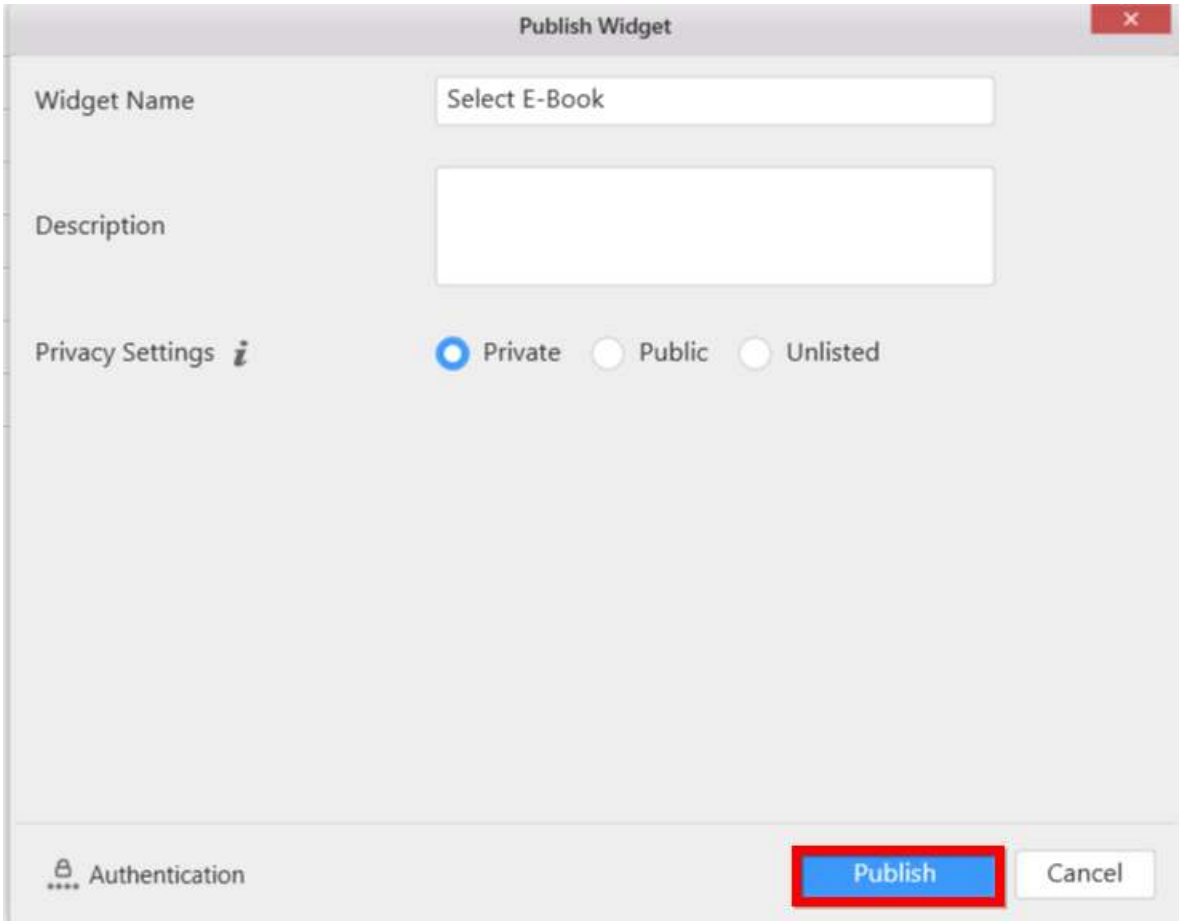
dashboard server. By default, Private option is selected.

Next button will be shown, if widget is configured with file type

datasource. Click Next button, the publishing options dialog will be shown for exporting file type data source in the server.



Publish button will be shown, if widget is configured with server type data source. Click Publish to export widget to dashboard server.



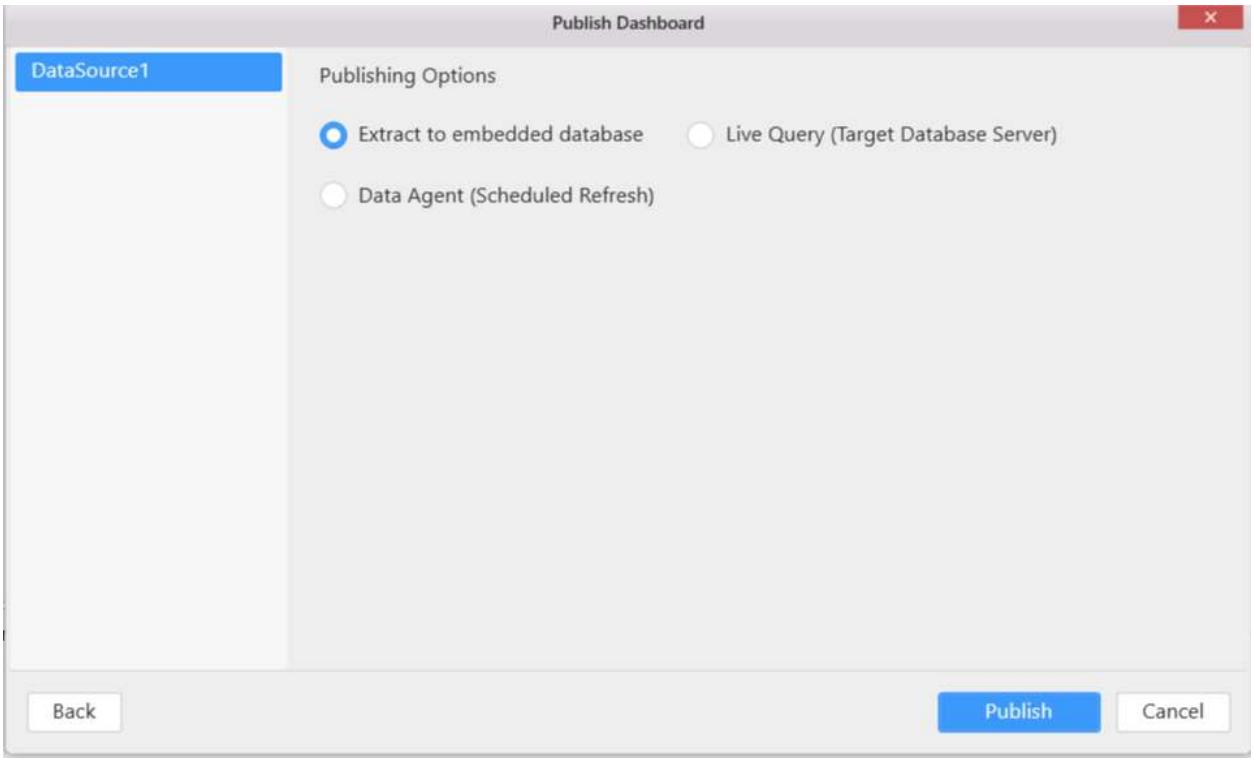
This widget can now be viewed under Widgets in the published dashboard server like below.

Click Cancel to cancel the export operation and close the dialog.

Publishing Data to Server

Data publish is required for publishing dashboards into dashboard server whose data was created through connection types other than server-based data connection types.

With this, proceeding after Publish Dashboard dialog, the following dialog will be shown which provides three publishing options to publish the data for corresponding data source.

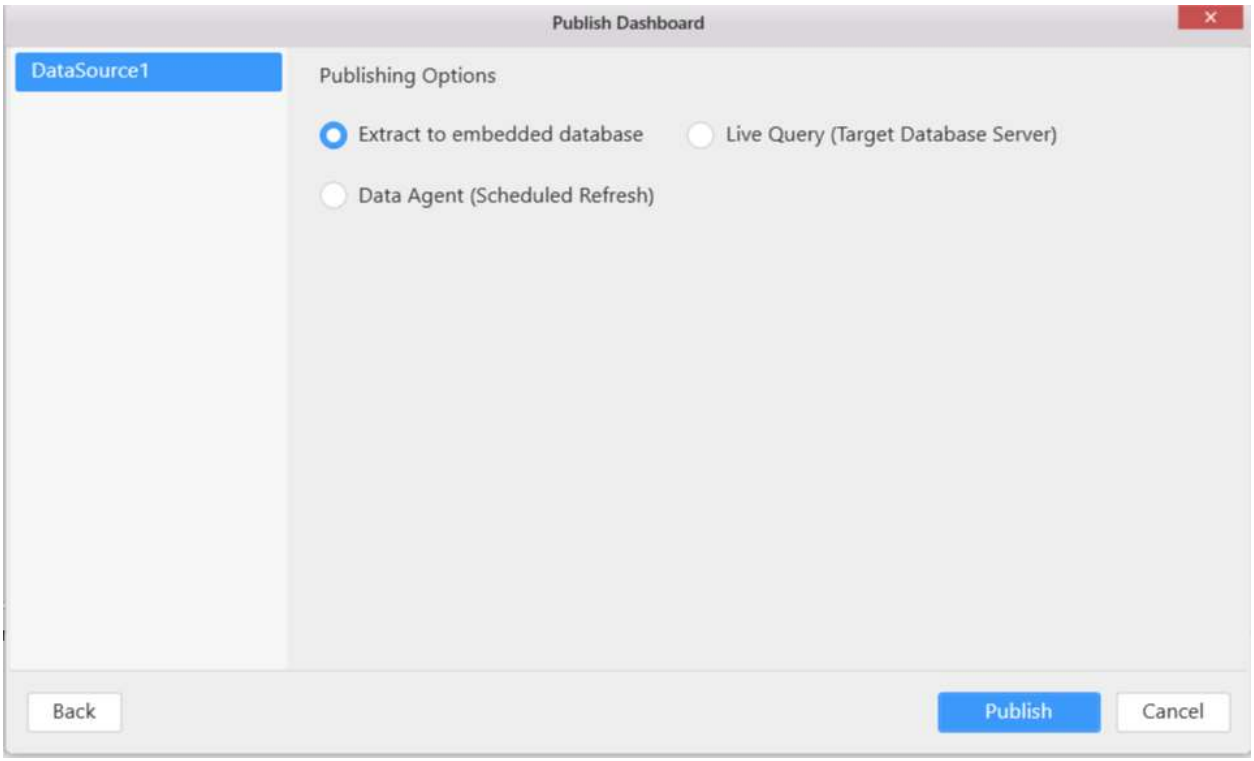


The below are the publishing option,

Extract to Embedded database Live Query * Data Agent

Publishing Data Extract to Embedded Database

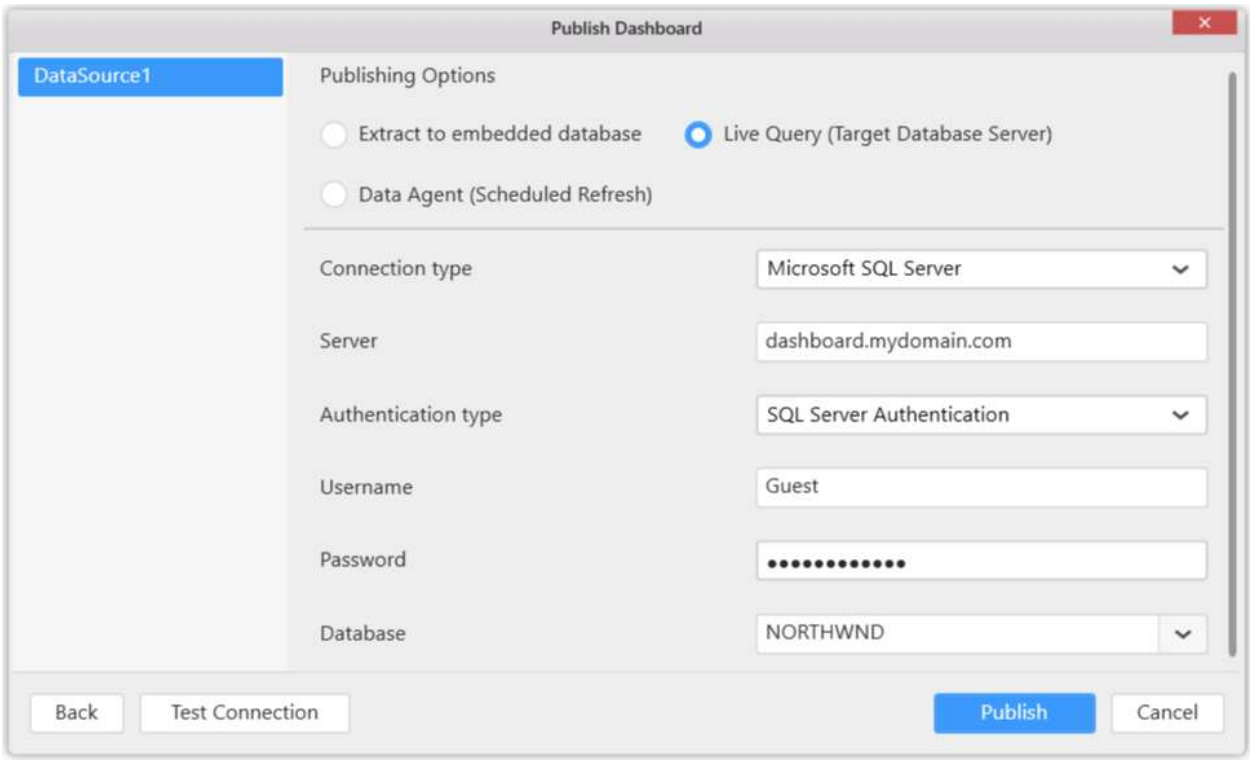
This option allows you to publish the data of the corresponding dashboard with its embedded database directly, without specifying the target server.



Publishing Data to Live Query (Target Database Server)

This option allows you to publish the data of the corresponding dashboard to a targeted database server whose system requirements has been discussed here. The database server should either have Microsoft SQL Server installed or ODBC-enabled databases like SQL, MySQL or Oracle, or Spark SQL configured.

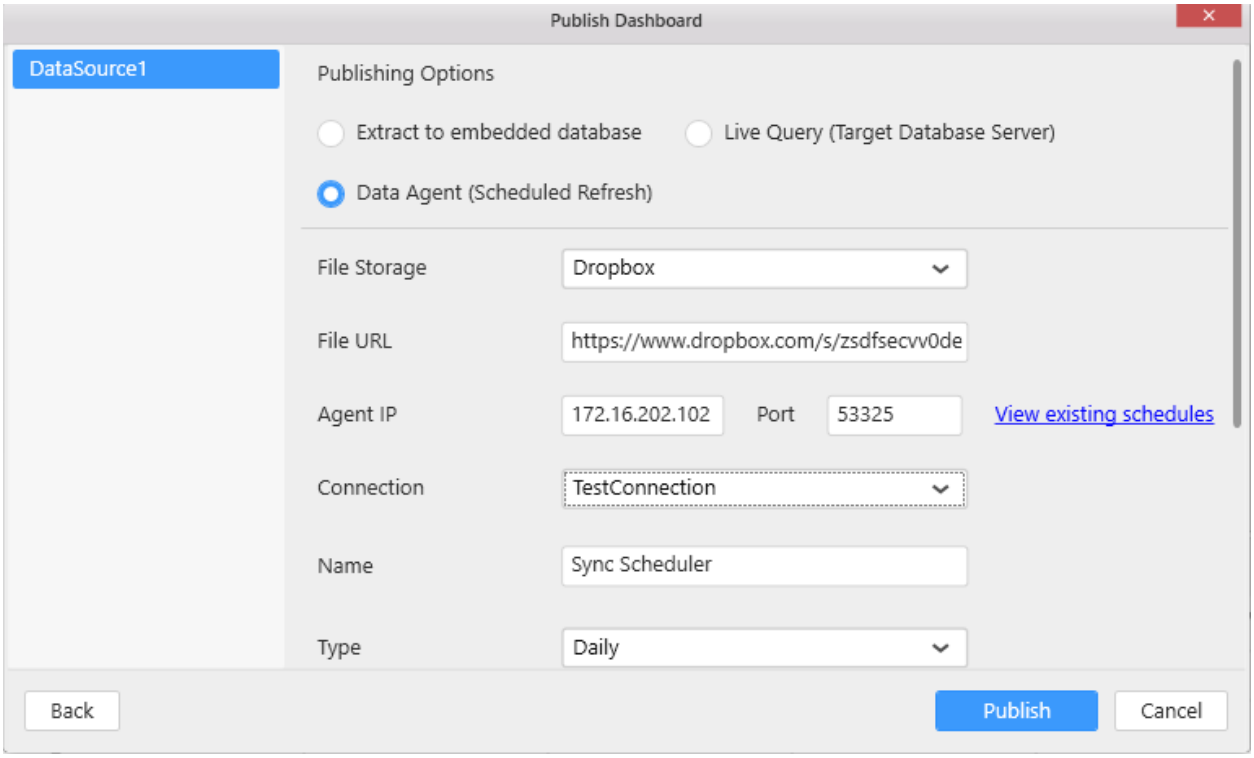
You can test connection details using Test Connection button in the Publish Dashboard dialog.



For data connections, other than server type, such as Microsoft Excel, JSON, CSV, Salesforce, Web Data Source and Microsoft Azure Table Storage, the data will be moved into the servers optionally as SQL Server, Spark SQL or ODBC-enabled databases (MSSQL, MySQL or Oracle) from the in-memory database which has been created while designing the data source.

Publishing Data to Data Agent

This option allows you to publish the data of the corresponding dashboard to SQL server. Also, it enables you to extract the updated data periodically from source connection type based on schedules.



Click here to know more about Data Agent.

You can configure the target server for every data source listed in the left-side panel of publishing window independently.

Note: If you are unable to view published dashboard while using two different machines or servers, user needs to ensure the access permission to the server which is configured with dashboard server.

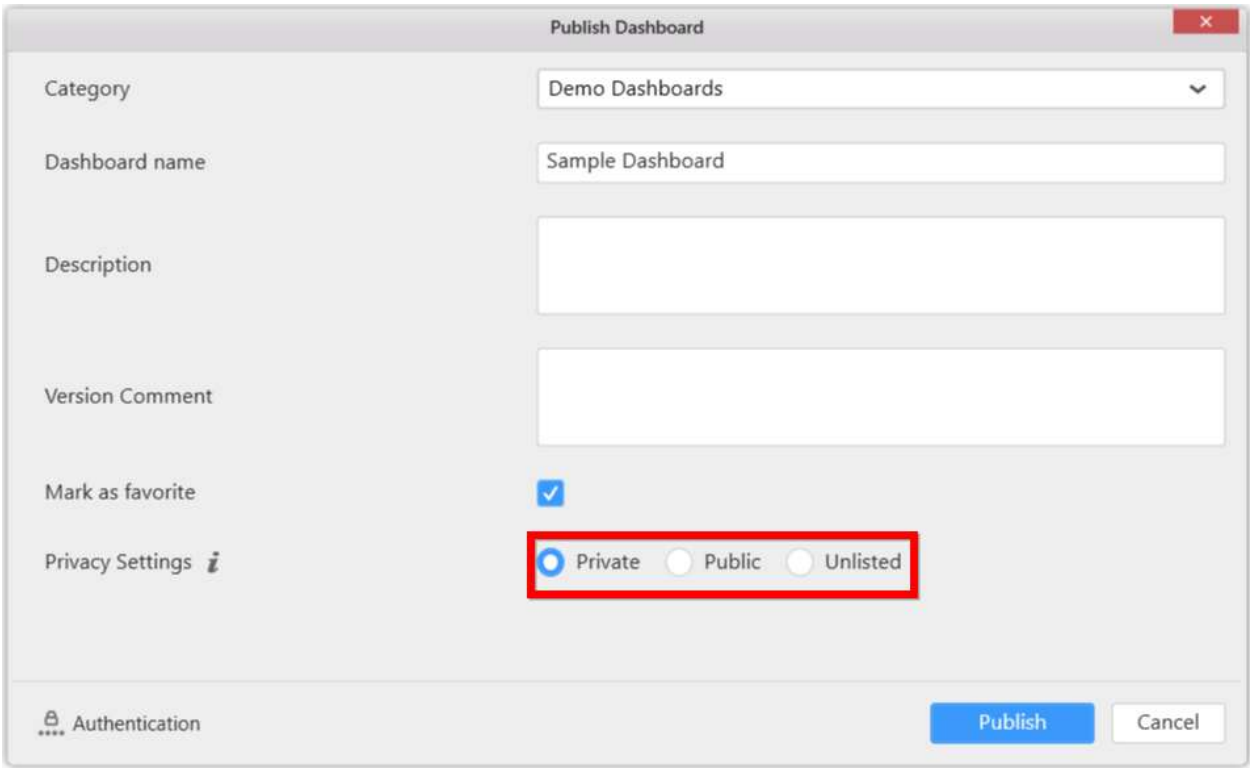
User Filter Configuration

You may configure user filters to restrict data view based on user logged in. Please refer here for more detail.

Privacy Settings

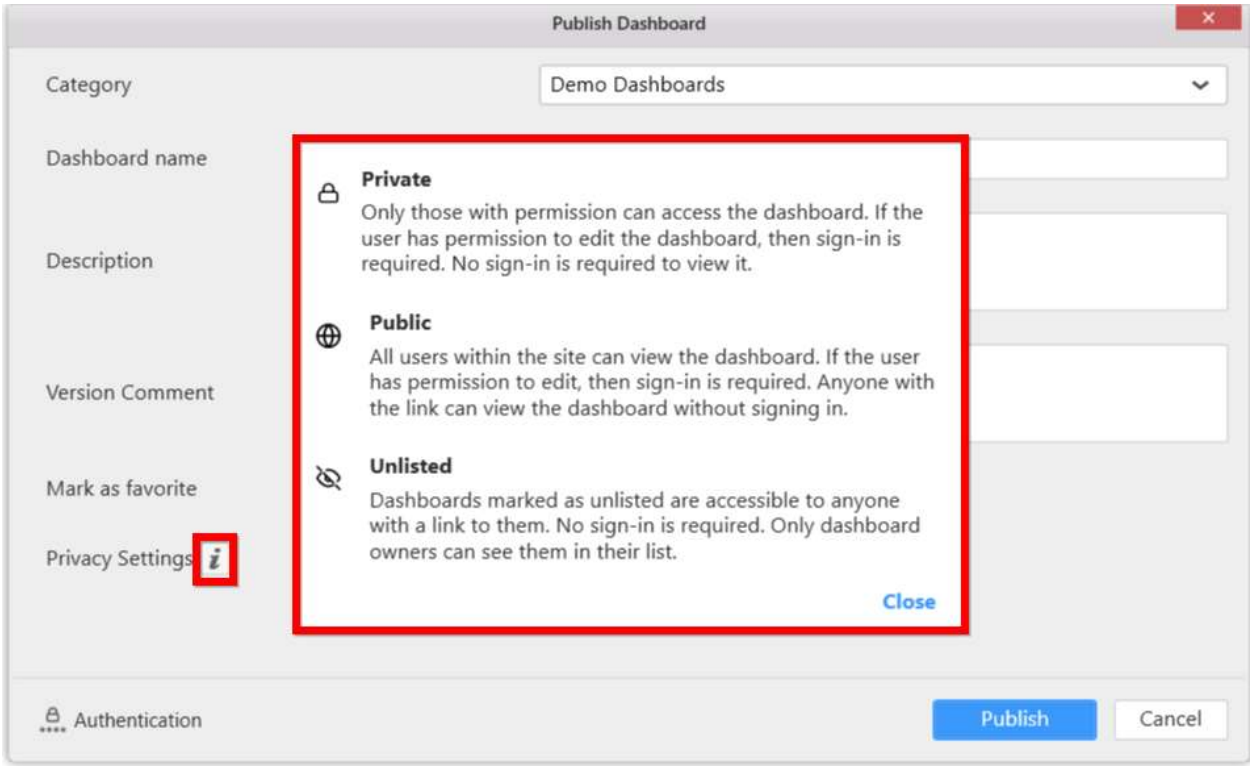
You can set dashboard accessibility for particular dashboard while publishing it to a Dashboard Server using Privacy Settings option in the Publish Dashboard dialog.

Note: Setting dashboard as public will disable the user based filter and other user specific functionalities applied to the dashboard.



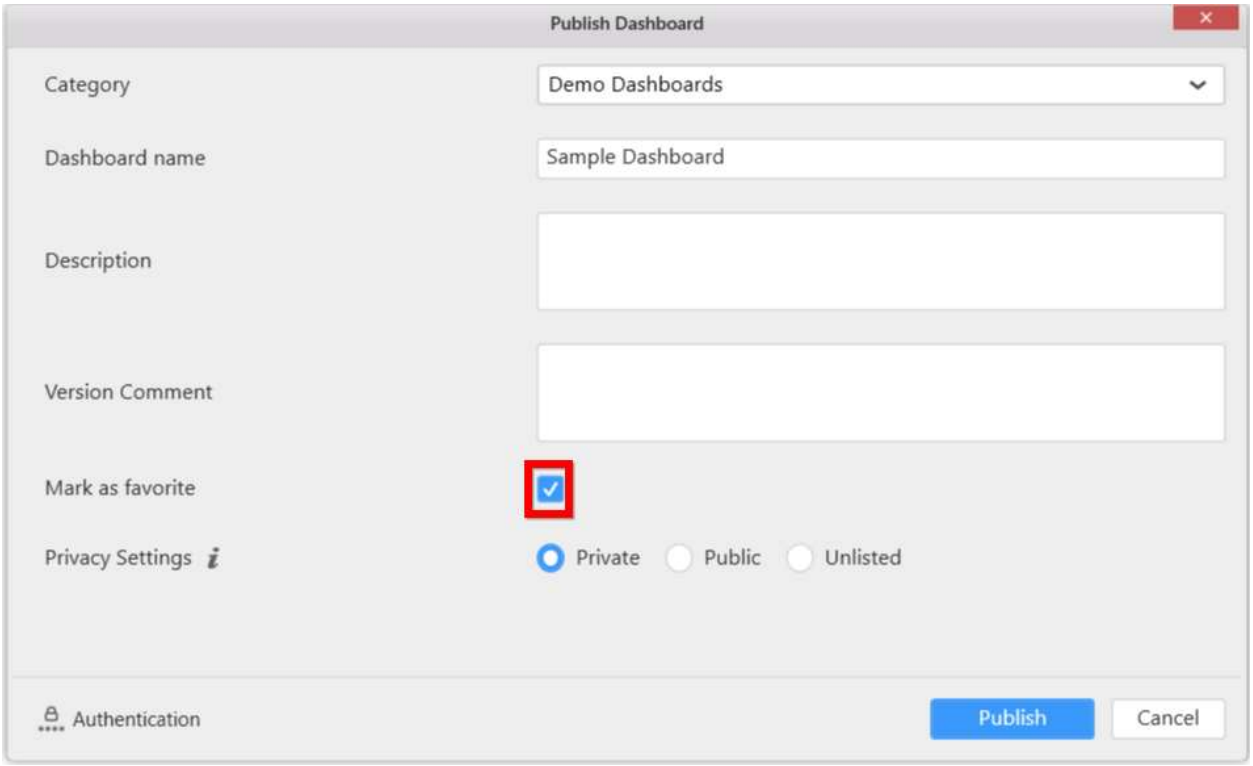
You can mark a dashboard as public even after publishing to dashboard server as discussed here.

Note: You can click information icon to know about Privacy Settings options as shown like below.



Marking as Favorite

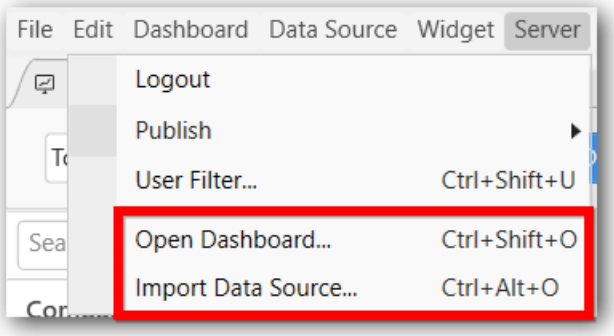
You can mark a particular dashboard as favorite while publishing to a Dashboard Server, so that dashboard can be listed under Favorite Dashboards category in the Dashboard Server. To set a dashboard as favorite, enable the check box Mark as favorite in the Publish Dashboard dialog.



You can mark a dashboard as favorite even after publishing to dashboard server as discussed here.

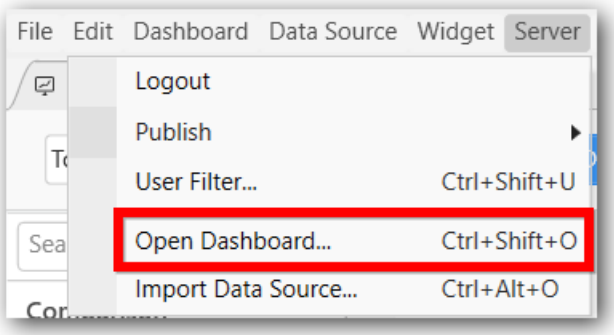
Importing from Server

Dashboards and Data sources can be imported from Dashboard Server through the Application Menu.

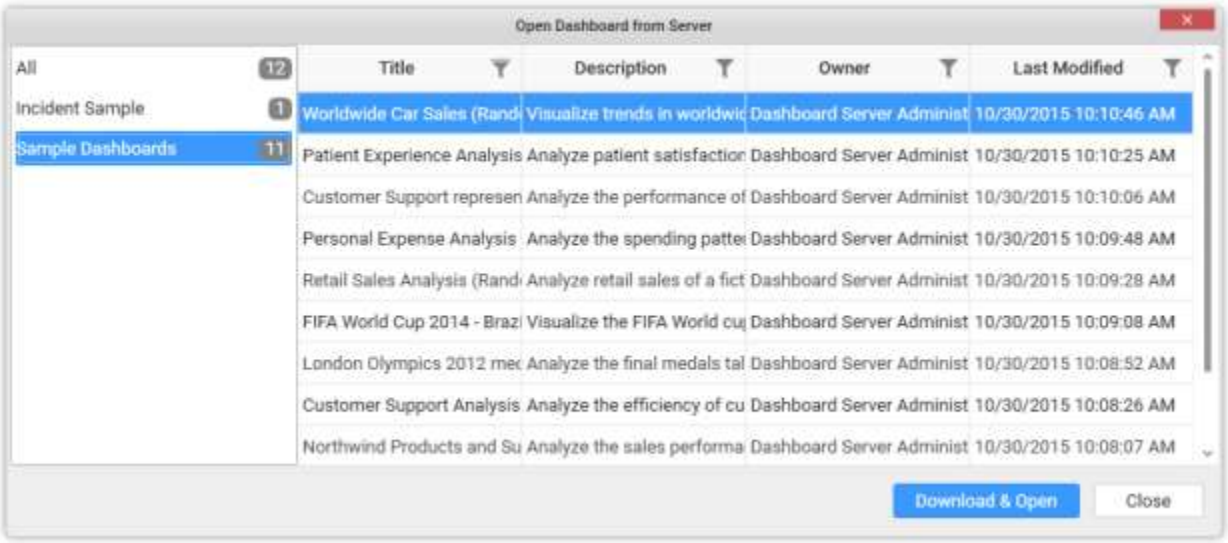


Importing Dashboard from Server

Click Server Menu and select Open Dashboard... menu item.

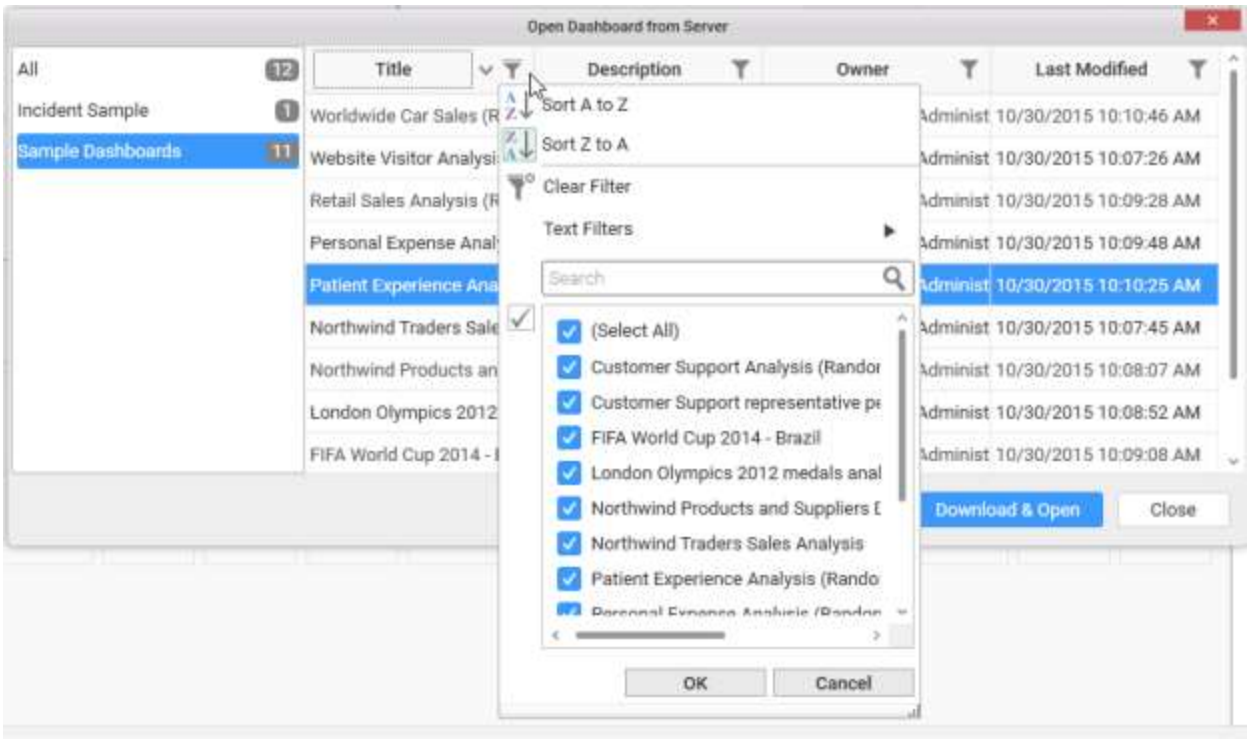


If you logged in to Dashboard Server already, you will be prompted with the Open Dashboard from Server dialog like below:



Note: Dashboards created using Web designer will not be displayed in the desktop designer.

Select the category in left pane and select the required dashboard to import at right pane. You may also filter and sort the display view for convenience.



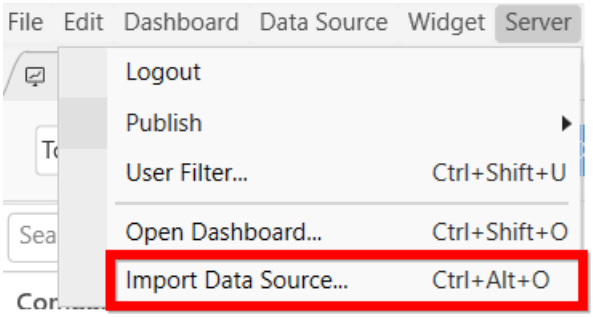
Click Download & Open to download the selected dashboard and open in the dashboard designer from which this operation was handled.

Click Close to close the dialog.

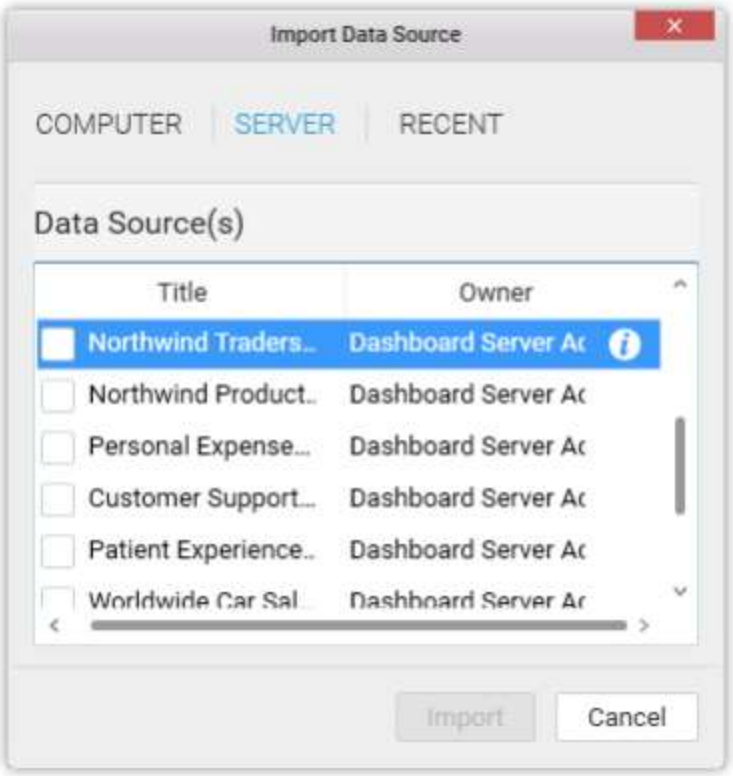
Save this dashboard in your local machine at your preferred location with a suitable name in SYDX format.

Importing Data source from Server

Click Server Menu and select Import Data Source… menu item.



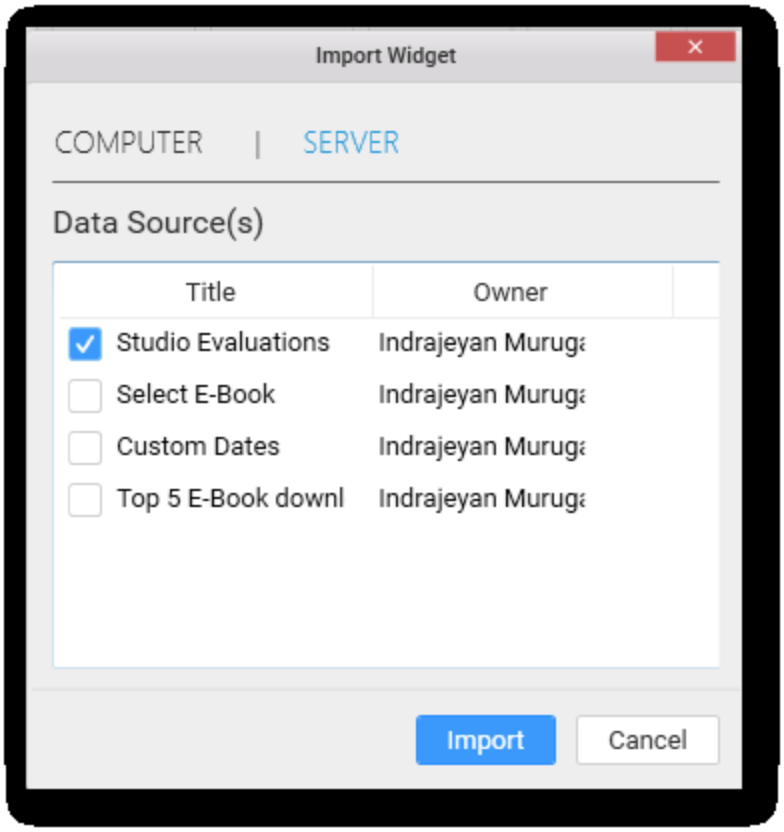
If you logged in to Dashboard Server already, you will be prompted with the Import Data Source dialog like below:



Importing Widget from Server

Click Import icon from WIDGETS container.

If you logged in to Dashboard Server already, you will be prompted with the Import Widget dialog like below:

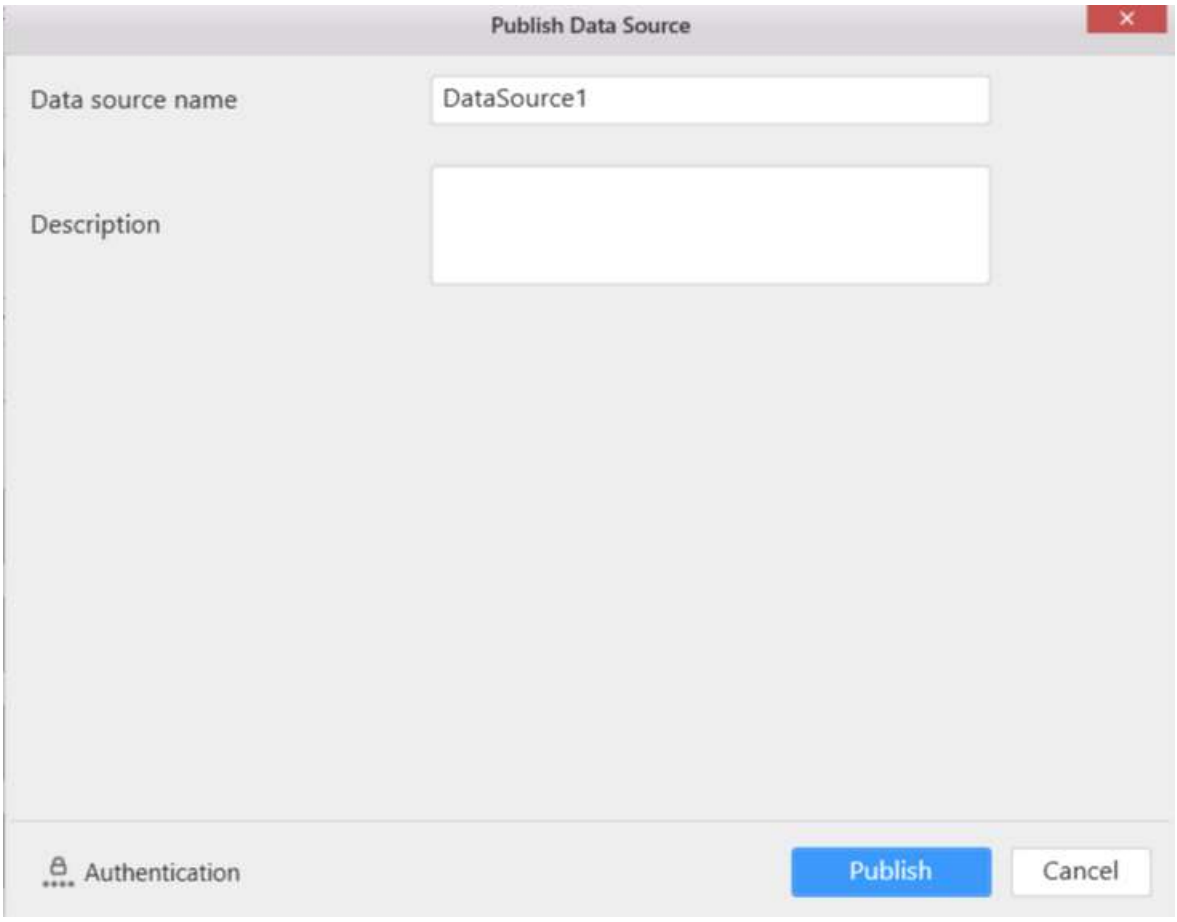


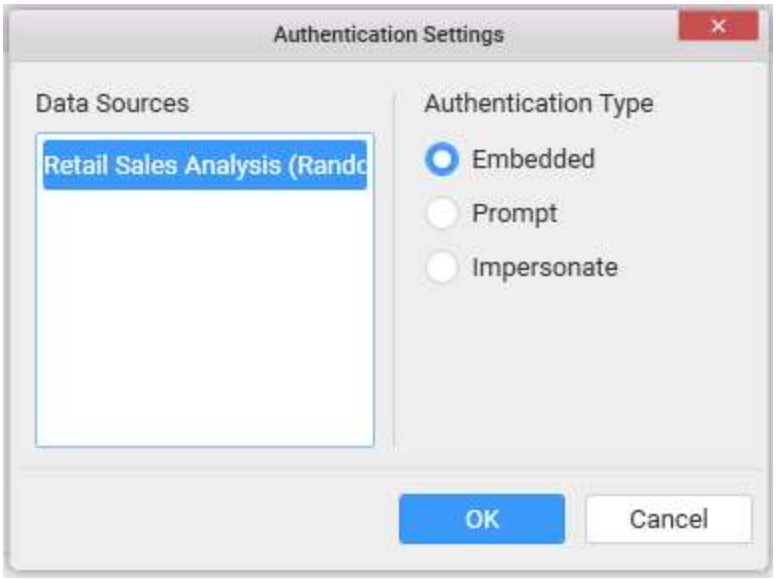
Data Source Authentication Modes

When a dashboard is published to a Dashboard Server, you can restrict access to the dashboards based on the following Authentication modes

1. SQL Server Impersonation within a domain
2. Prompt for User Credentials
3. Embedded User Credentials

Configure these modes through the Authentication option in the Publish windows.





SQL Server Impersonation within a domain

Under Windows authentication mode, user accounts imported from Active Directory to Dashboard Server will impersonate as Dashboard Server user to communicate with SQL Server instance. For querying, both SQL Server and Active Directory account names should be the same and their

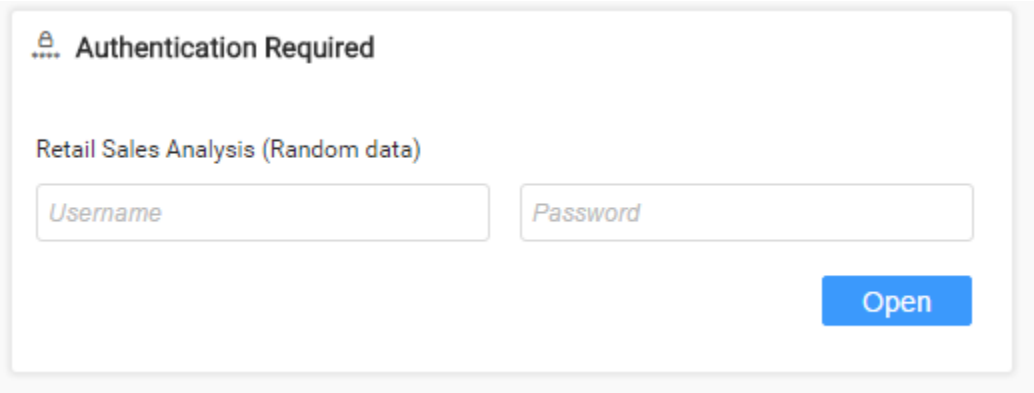
authentication type should be Windows Authentication. With this setup, SQL Server administrator can define permissions for specific users.

Thus, the queries submitted is executed based on the permissions defined for that user in the SQL Server. The response will be sent accordingly back to the Dashboard. In Dashboard Designer, when you are connecting to an SQL Server to enable impersonation, you should connect using an account that has Impersonate permission enabled.

Note: The Impersonate option will get displayed only when the signed in dashboard server user account is the one imported from Active Directory.

Prompt for User Credentials

Setting Prompt option will prompt to enter the SQL Server credentials upon launching the respective dashboard(s). Using this mode, you can enforce row-level security in non-Active Directory environment for users with different set of SQL Server credentials.



Embedded User Credentials

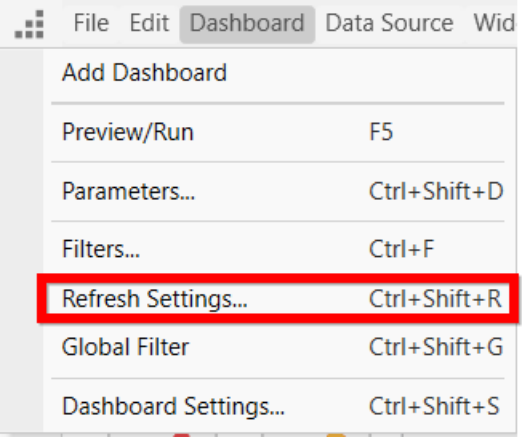
No authentication will be done in this mode. The connection parameters embedded in Dashboard will be used for querying up data.

Refresh Dashboard

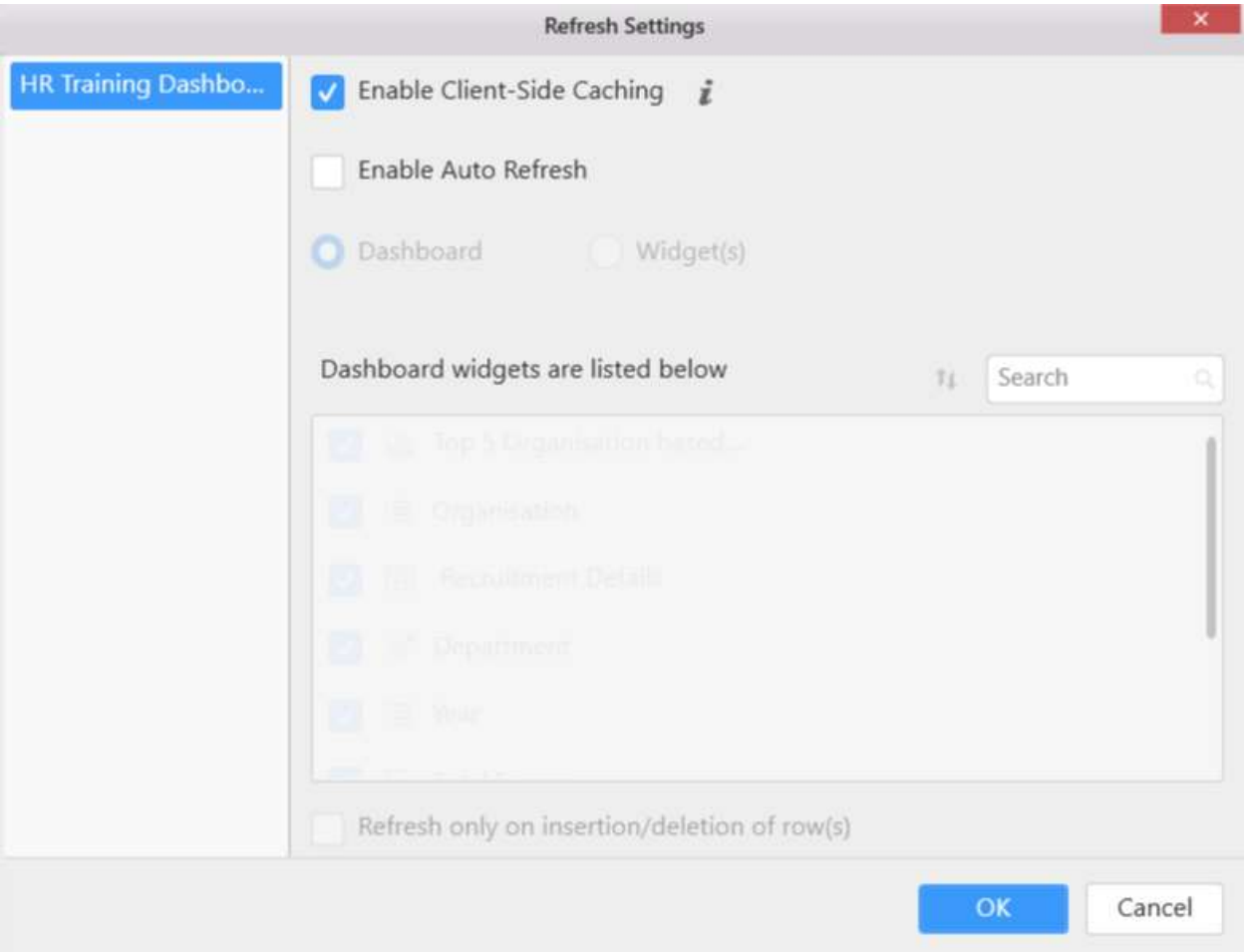
Syncfusion Dashboard Designer allows you to configure the scheduled refresh of dashboard. Either the whole dashboard or specific widgets in a dashboard can be refreshed automatically based on a timer.

How to enable automatic refresh for a Dashboard?

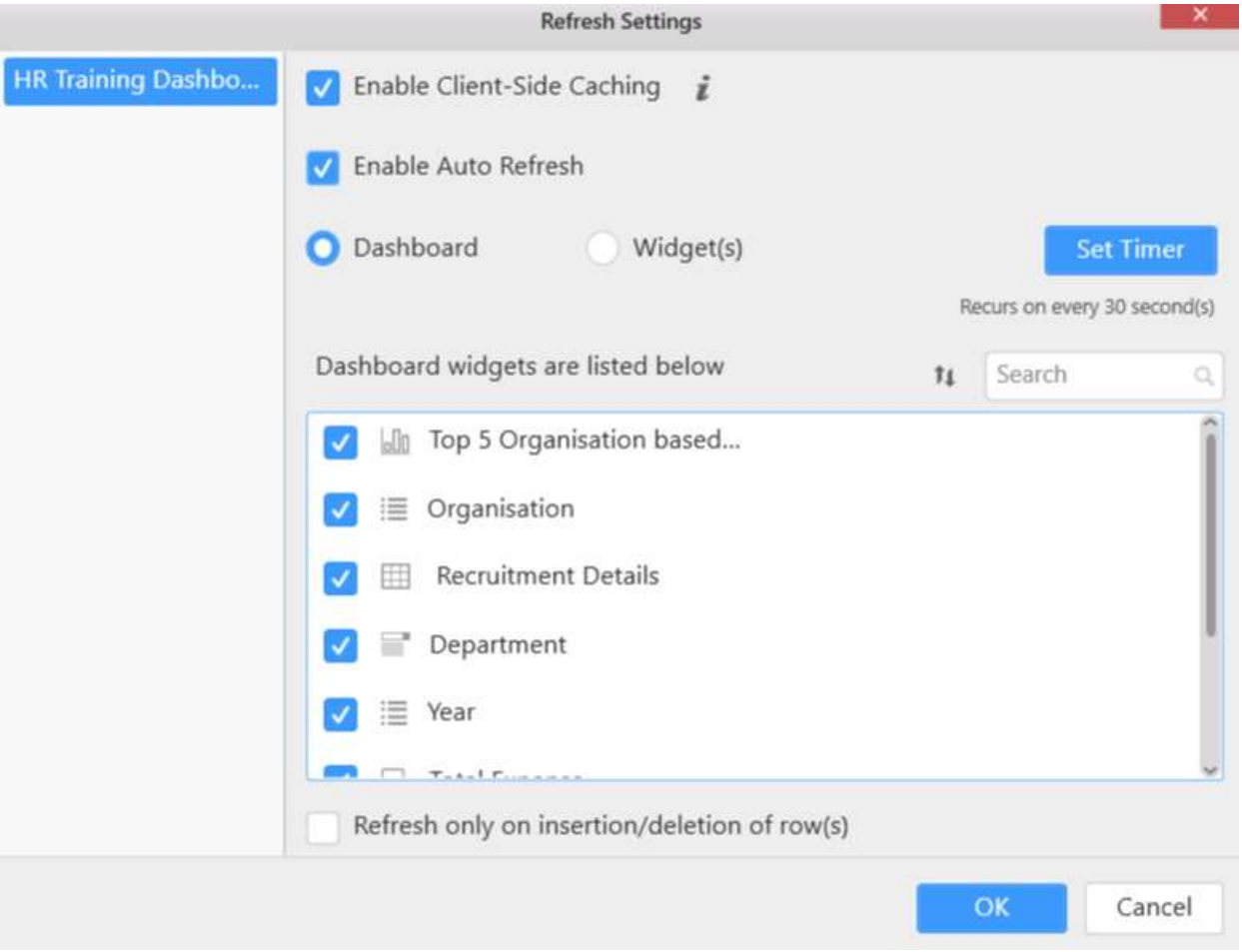
To enable auto refresh, navigate to the Dashboard menu and click the Refresh Settings… to open the Refresh Settings dialog. The refresh settings window can also be opened by using the keyboard shortcut **Ctrl+ Shift+ R**.



Select the dashboard for which you need to set the refresh settings.



Choose Enable Auto Refresh.

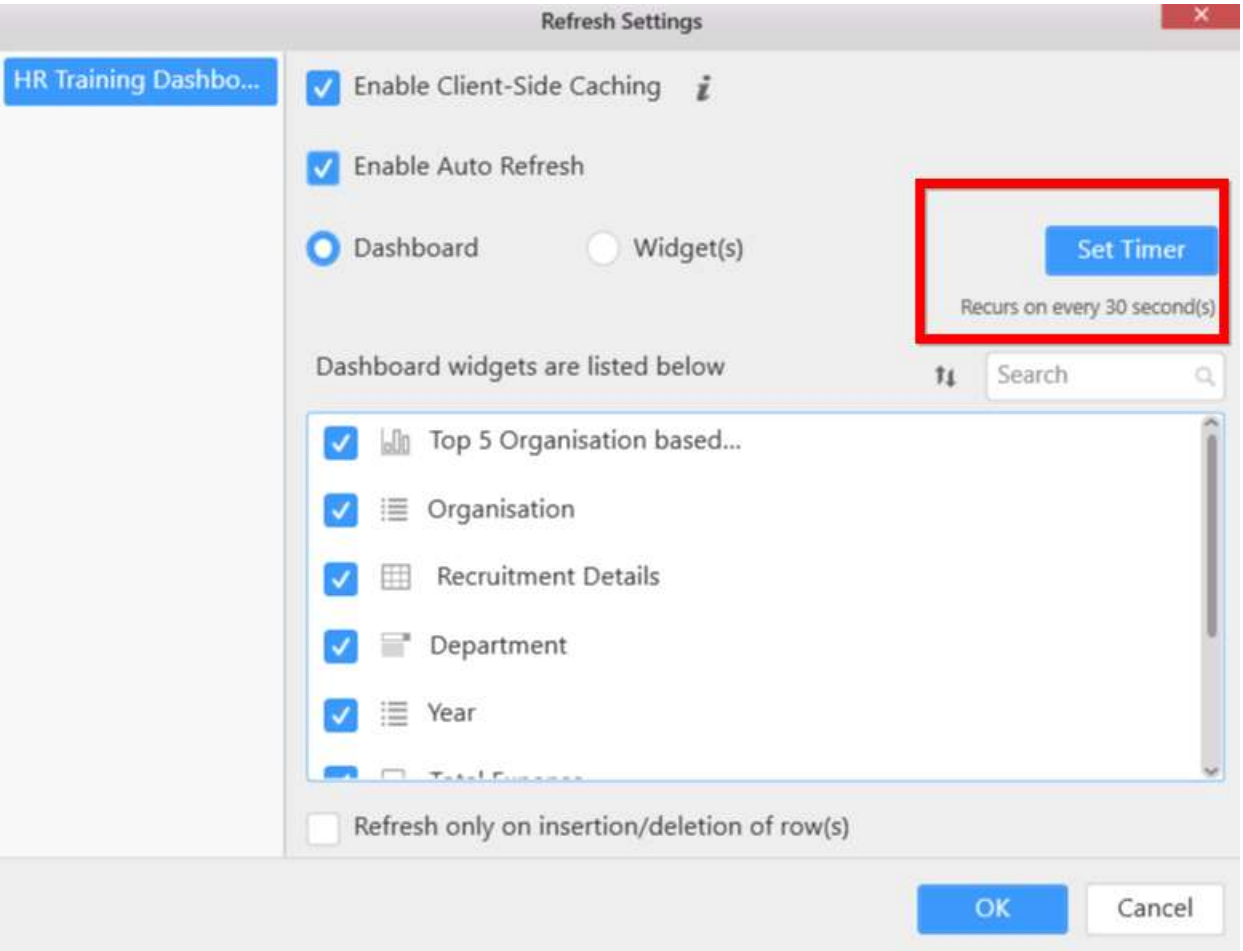


By default, the auto refresh will be applied to all widgets present in the dashboard report.You can select/unselect the widgets based on your requirement. The selected widgets will alone be refreshed in the viewer.

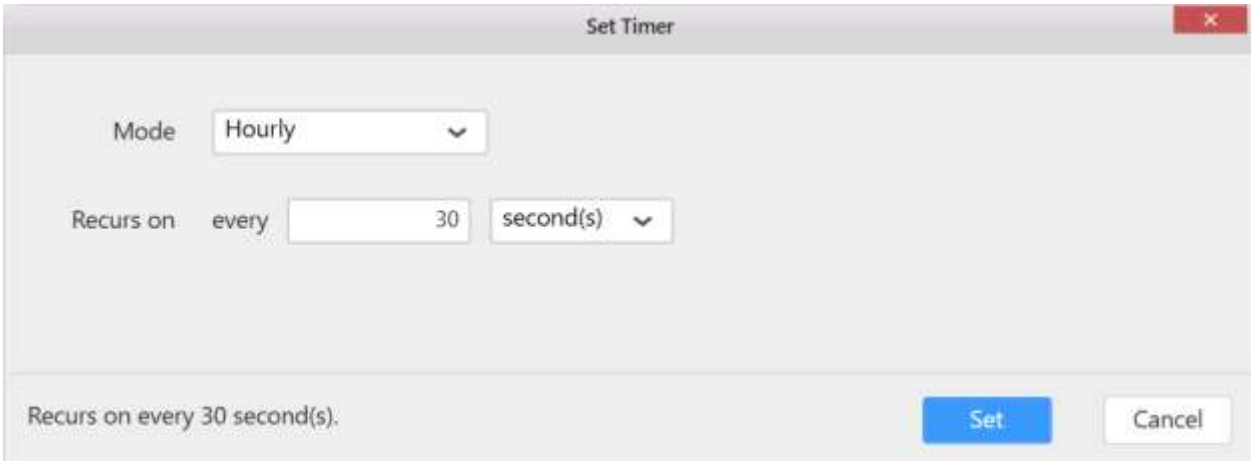
Click OK to save your changes.

How to set the timer for automatic refresh?

The time Interval for automatic refresh is **30 seconds**, by default. To set a different time interval, click the Set Timer.



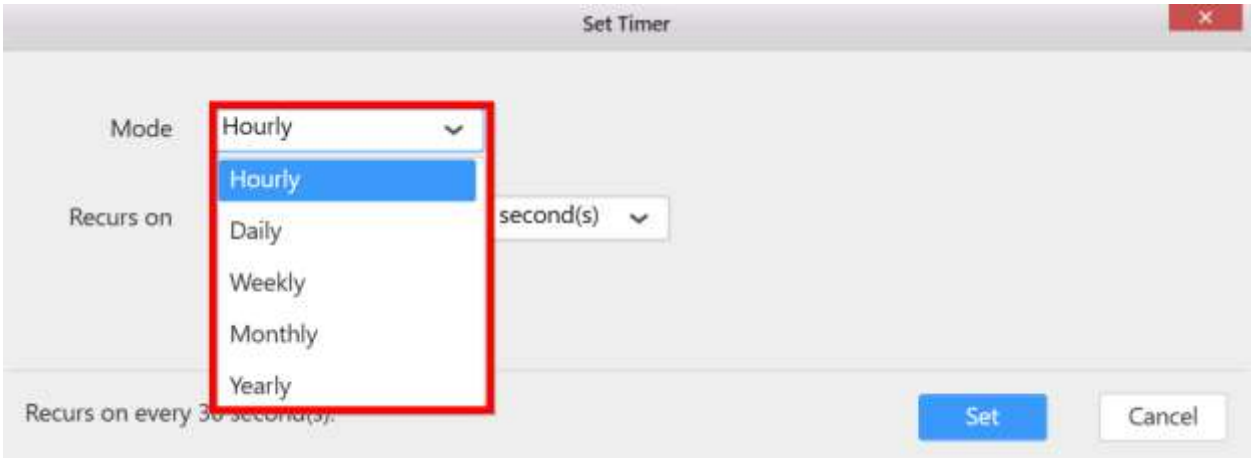
Now the Set Timer window will be opened.



You can select the required Mode and set the time settings based on your requirement.

Mode available in set timer window

The available modes are **Hourly**,**Daily**,**Weekly**, **Monthly** and **Yearly**.

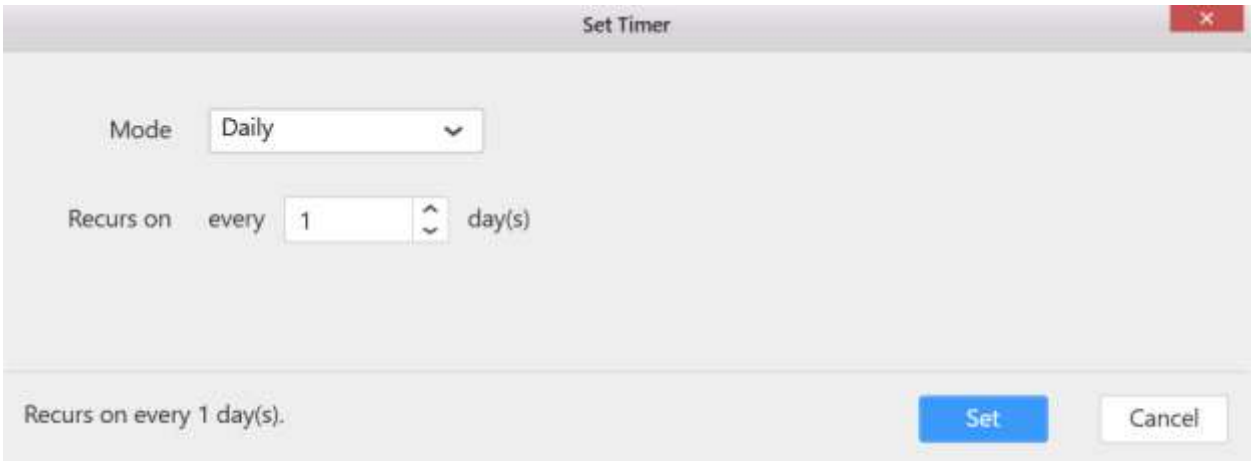


Hourly mode

In hourly mode, you can set the refresh interval time from seconds to hours.

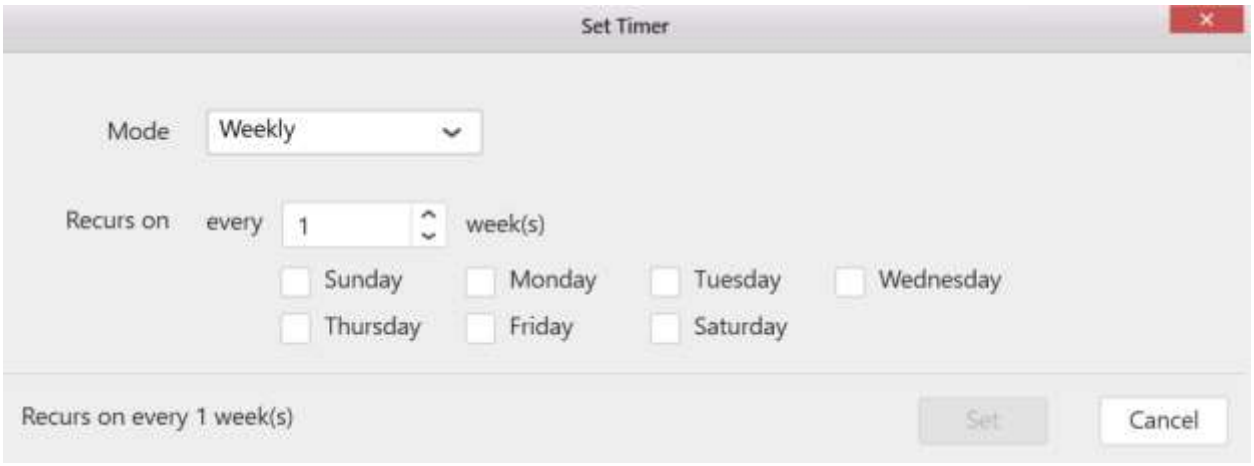
Daily mode

The dashboards will be refreshed on every selected day.



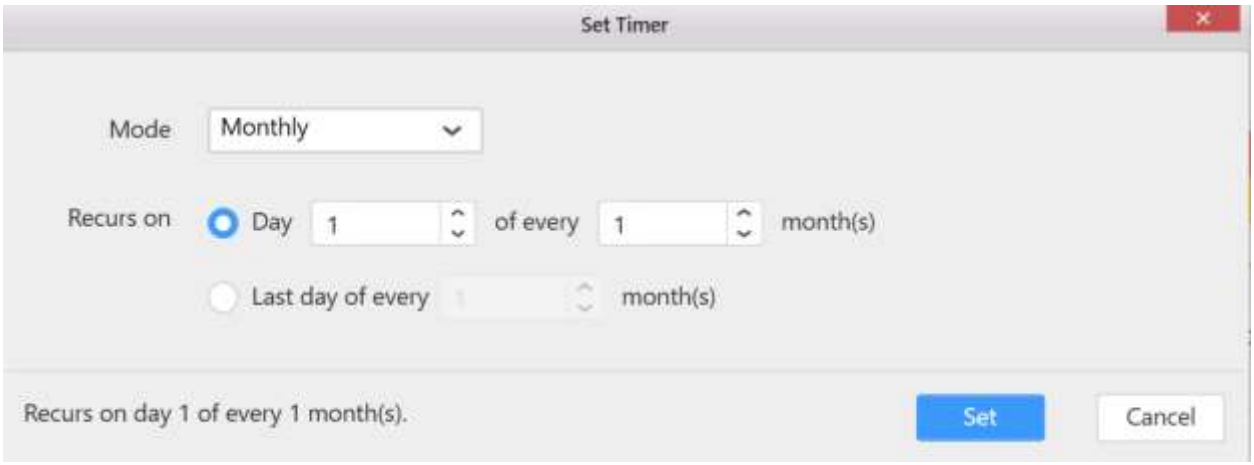
Weekly mode

You can select the week interval and days of the week by using the weekly mode.



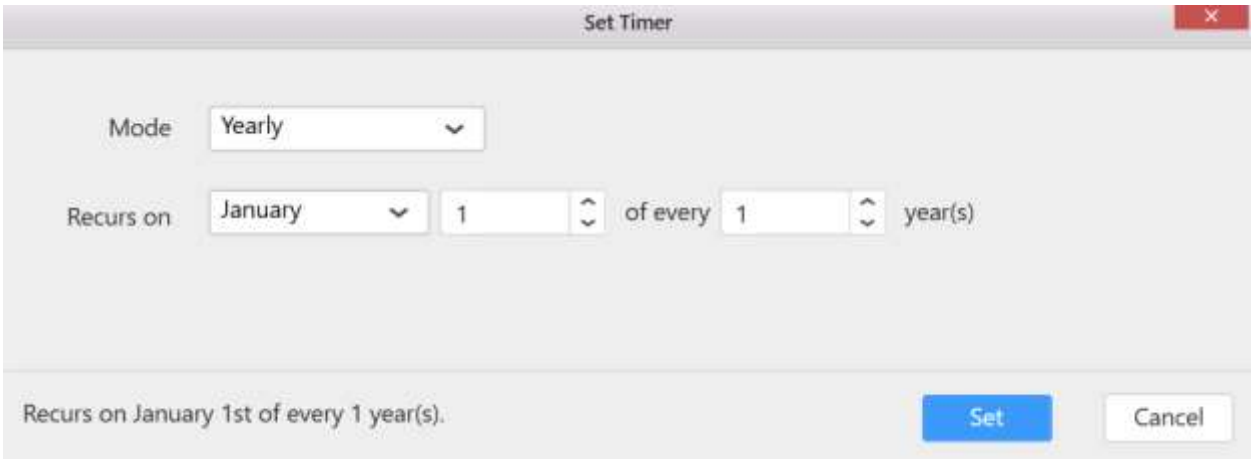
Monthly mode

You can select the specific date and month interval for refresh interval or you can select the last day of the month as the timer value.



Yearly mode

The Yearly mode allows you to set the month, date and year interval.

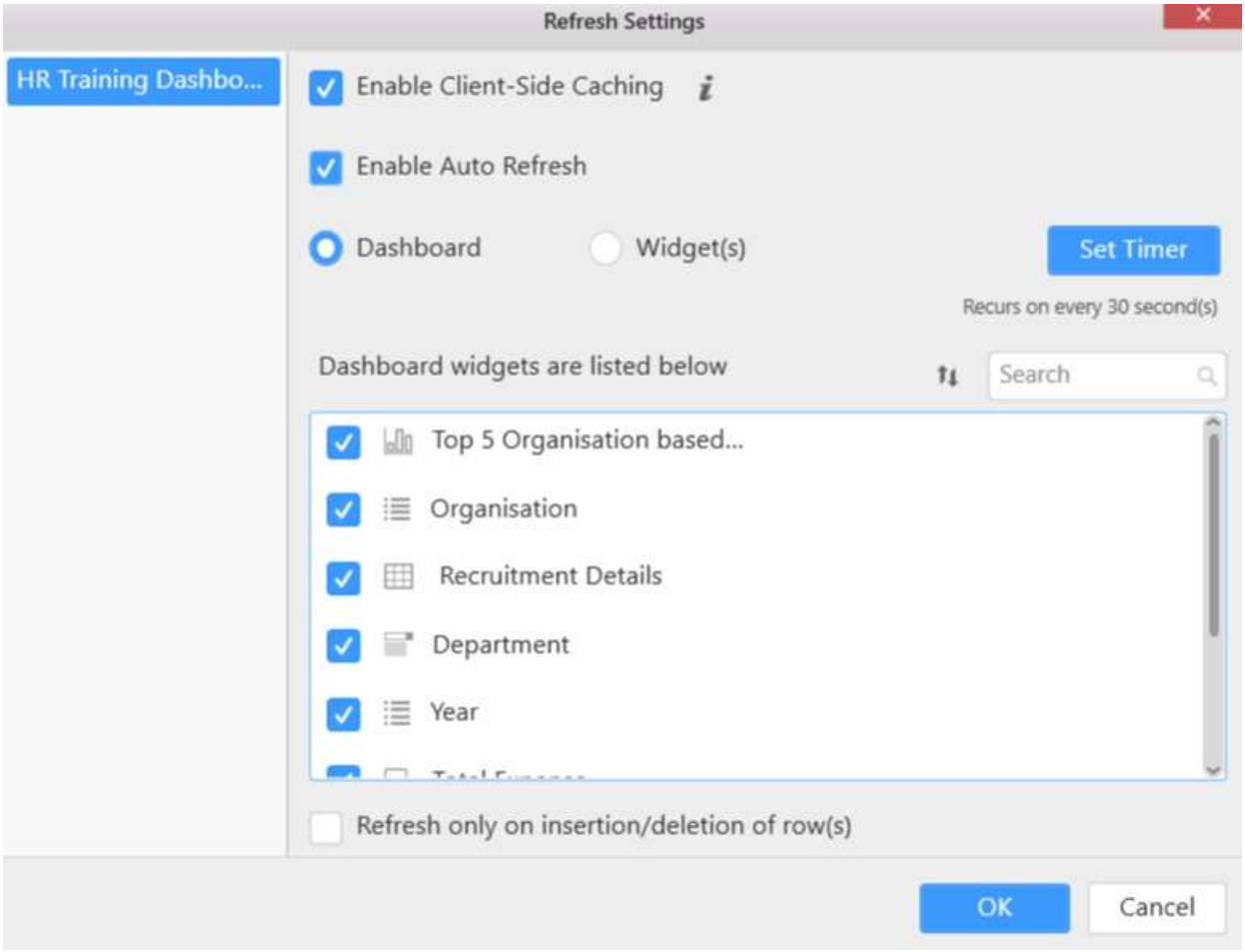


After setting the timer values click the Set to apply the changes.

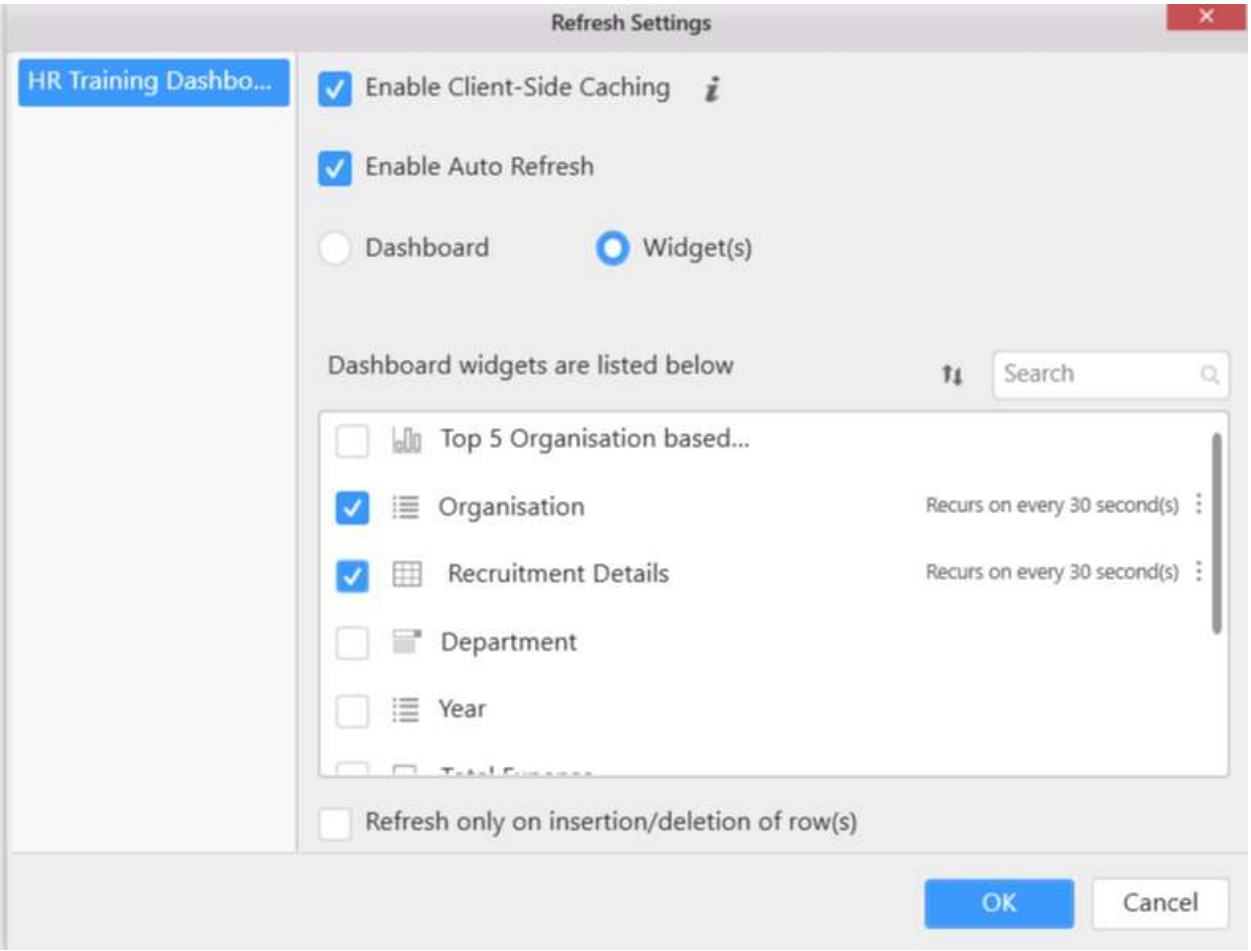
How to enable automatic refresh for particular widgets in a dashboard?

The automatic refresh for particular widgets alone can be configured through the Refresh Settings dialog.

Select Enable Auto Refresh in the dialog and then select Widget(s).

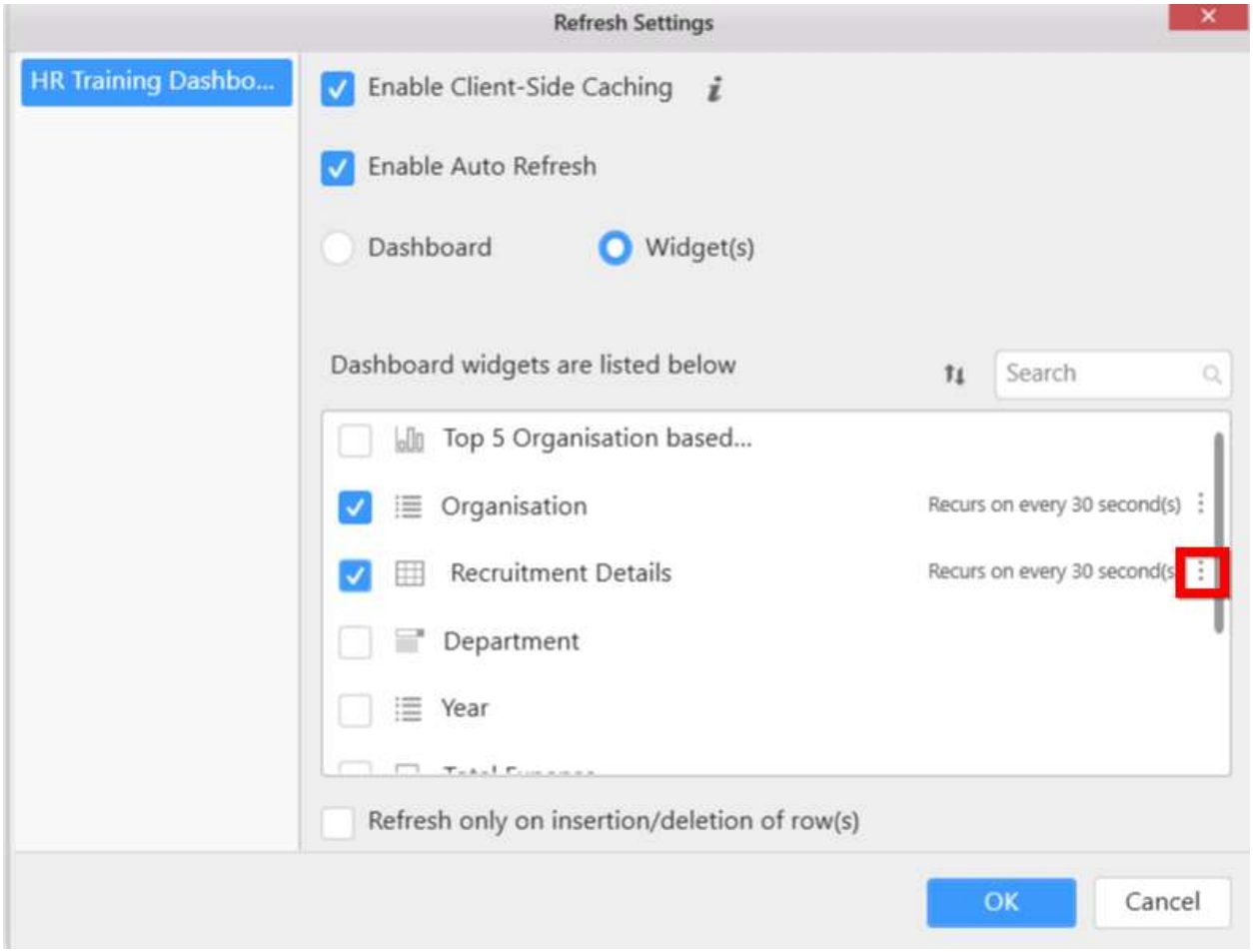


Select the widgets that you needed to refresh automatically...

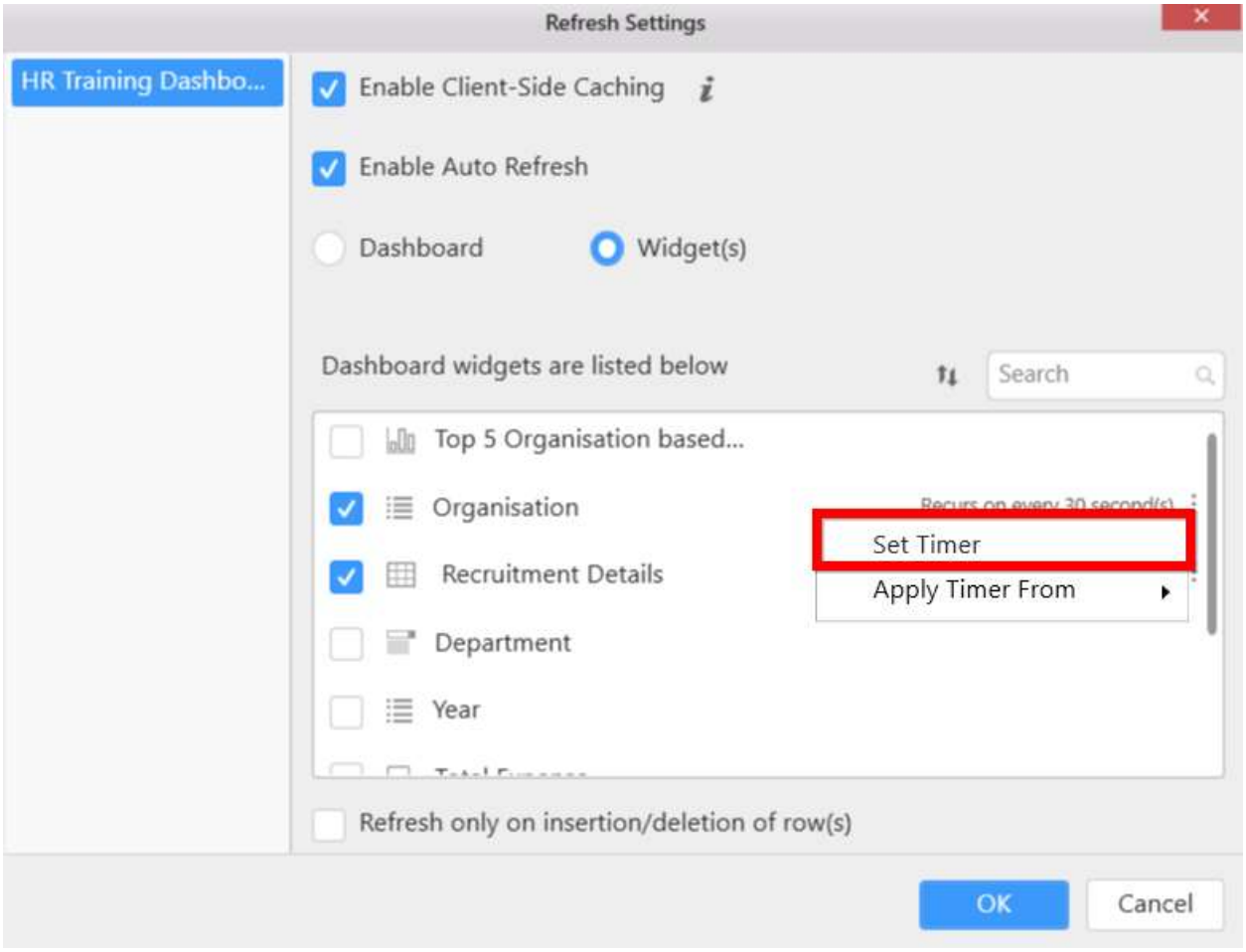


Set timer option for widgets

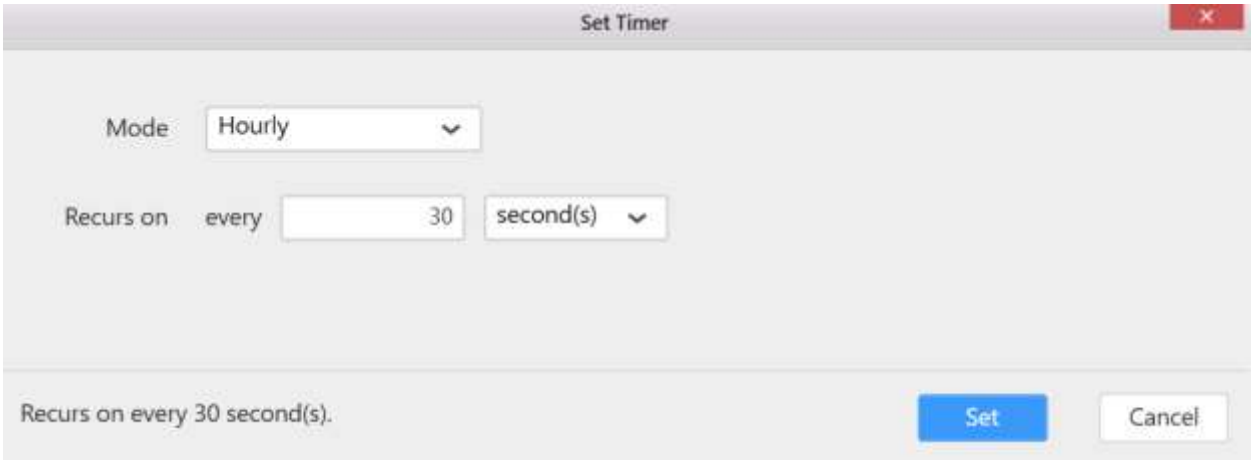
Now, click the **settings** to set the timer for the selected widget.



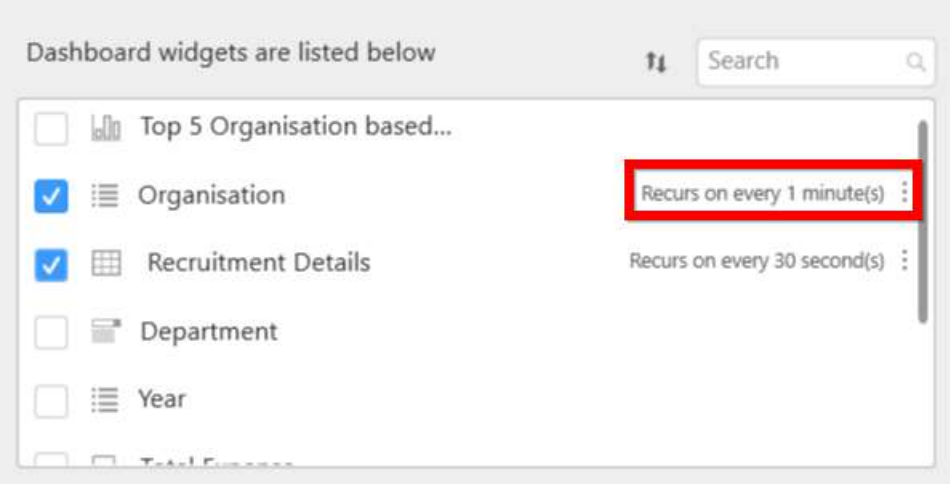
Click the Set Timer shown in the menu.



Now, the Set timer window will be shown and you can set the required time for refresh.

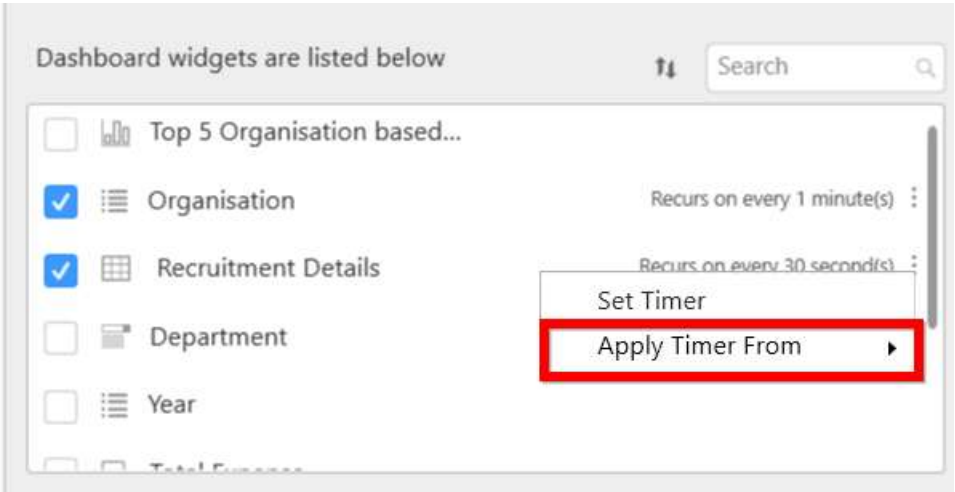


Now ,the the time interval will be shown next to the widget as follows.

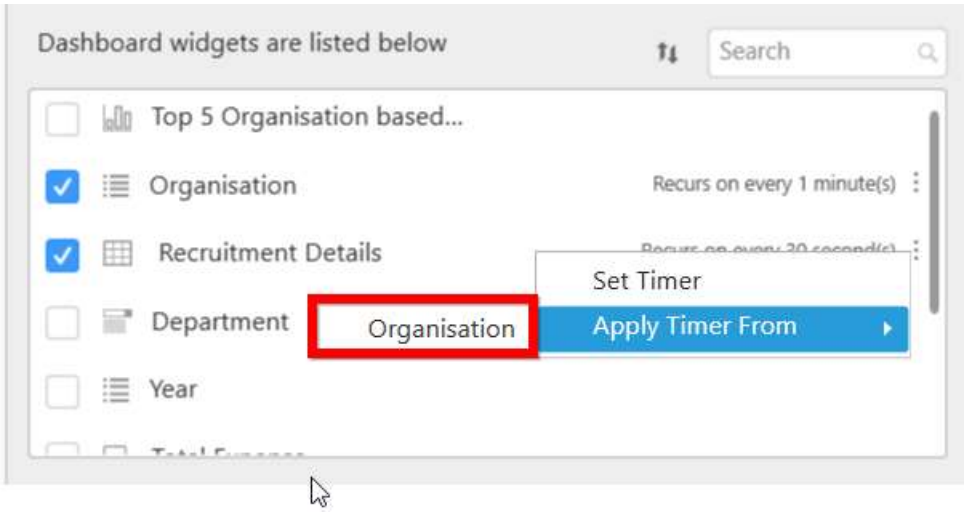


Apply timer from option for setting the timer value in widgets

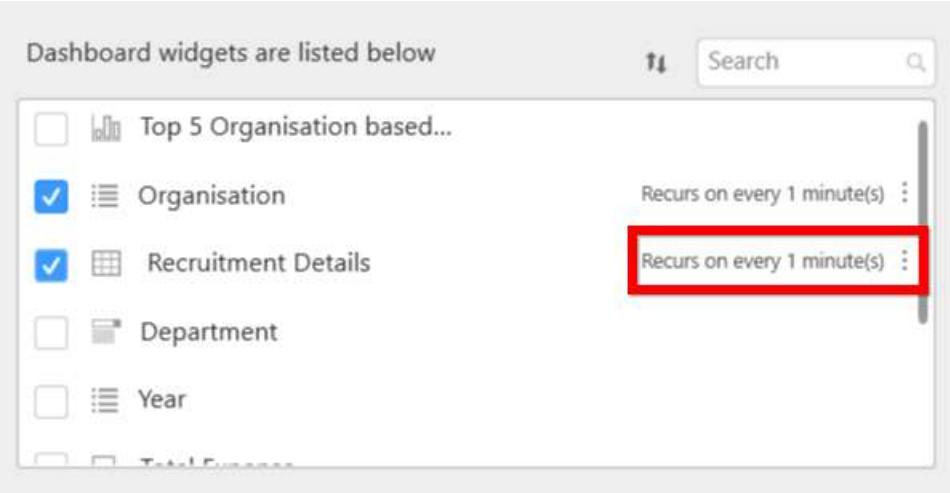
This option allows to use the same timer value applied to other widgets. Click the settings to get the menu option.



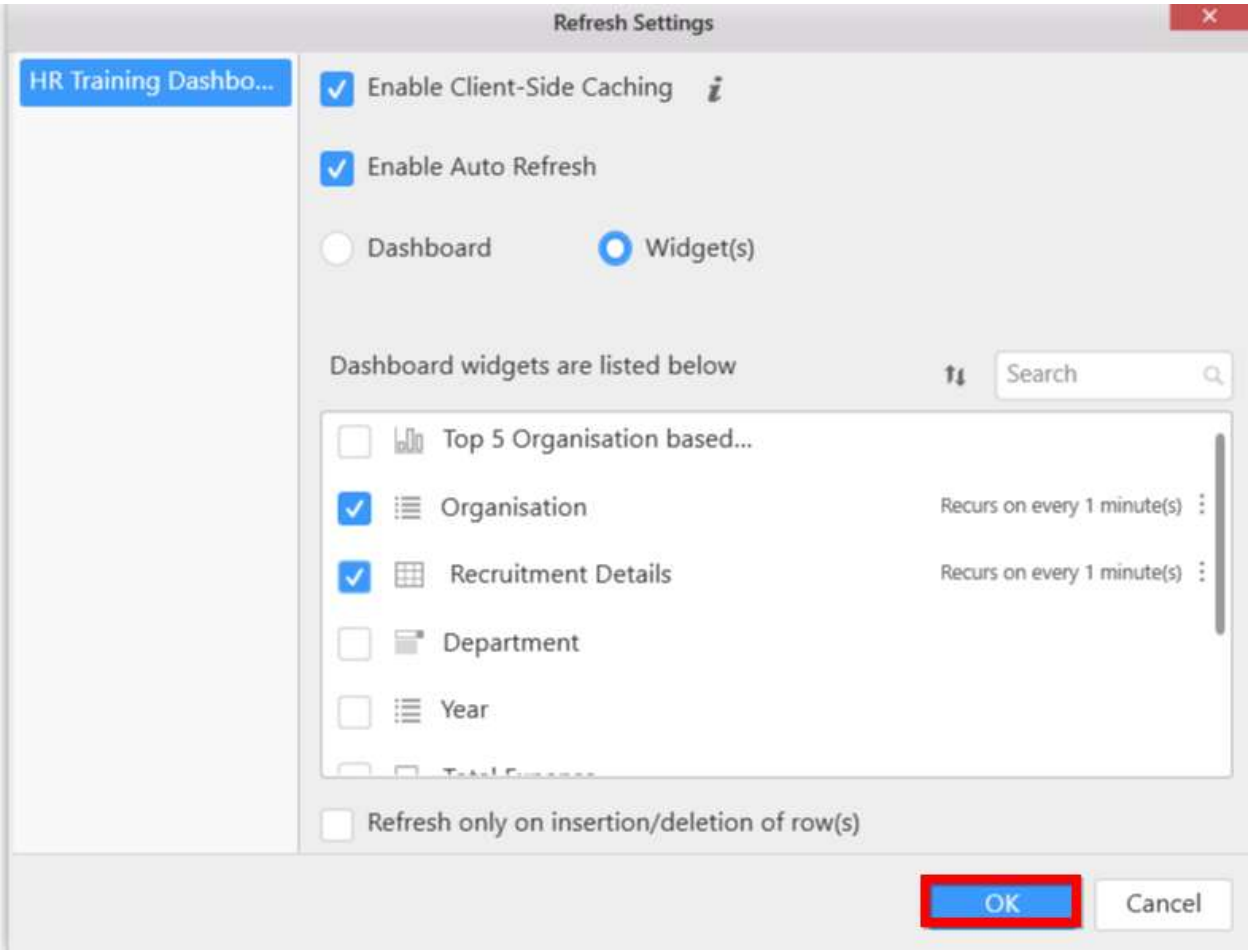
Select the widget from which you want to apply the timer value.



Now,the timer value will applied to Organization Category widget from Categories with the least demand.



Click OK to apply the refresh settings and save your dashboard report.

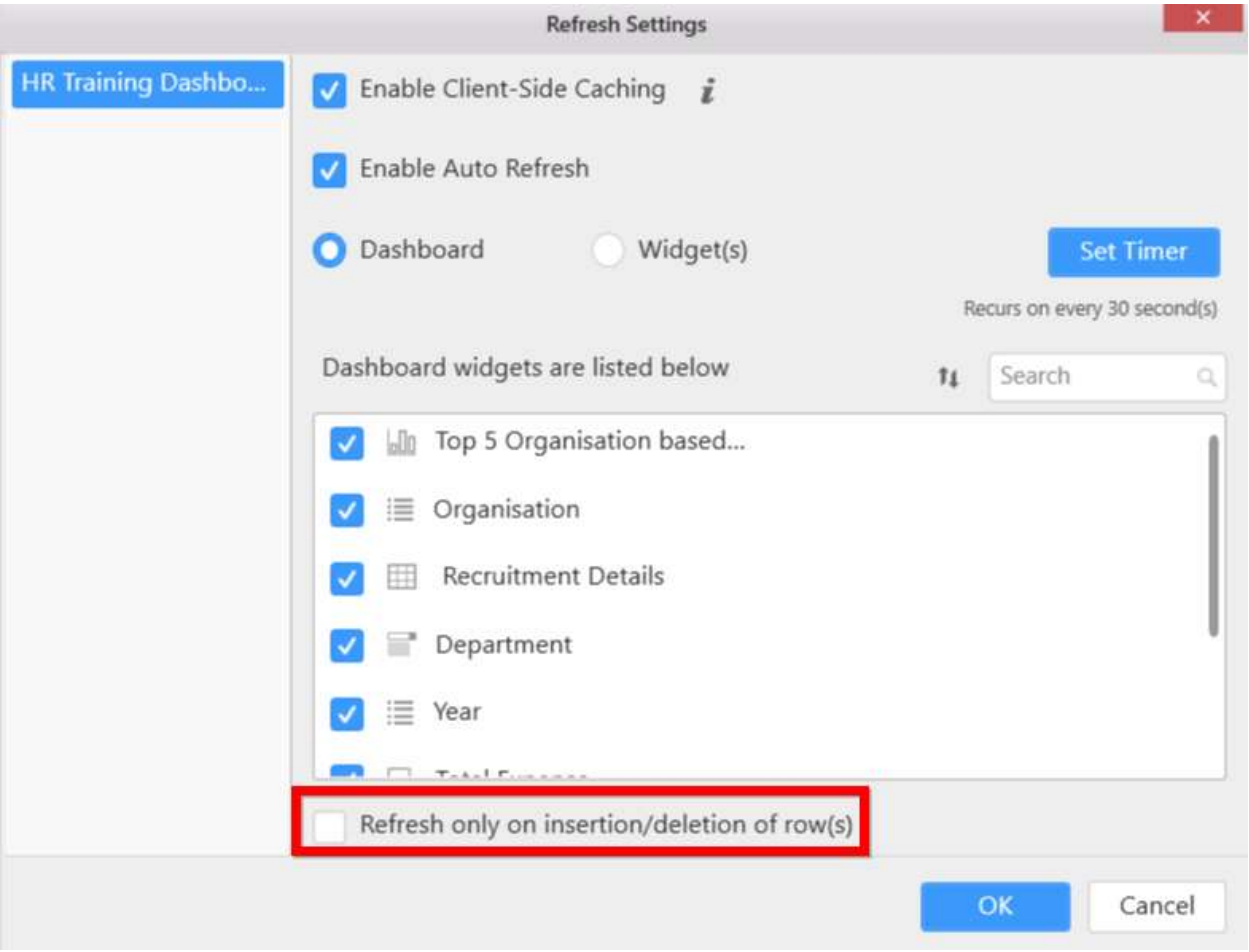


How to enable automatic refresh only for any record insertion or deletion in the associated database?

Automatic refresh will keep track of changes in the data even it is an update to existing record. To refresh only on new insertion or deletion of records,

Enable the auto refresh settings for the dashboard or widget and set the required timer value.

Click the **Refresh only on for insertion/deletion of row(s)** to apply the changes.



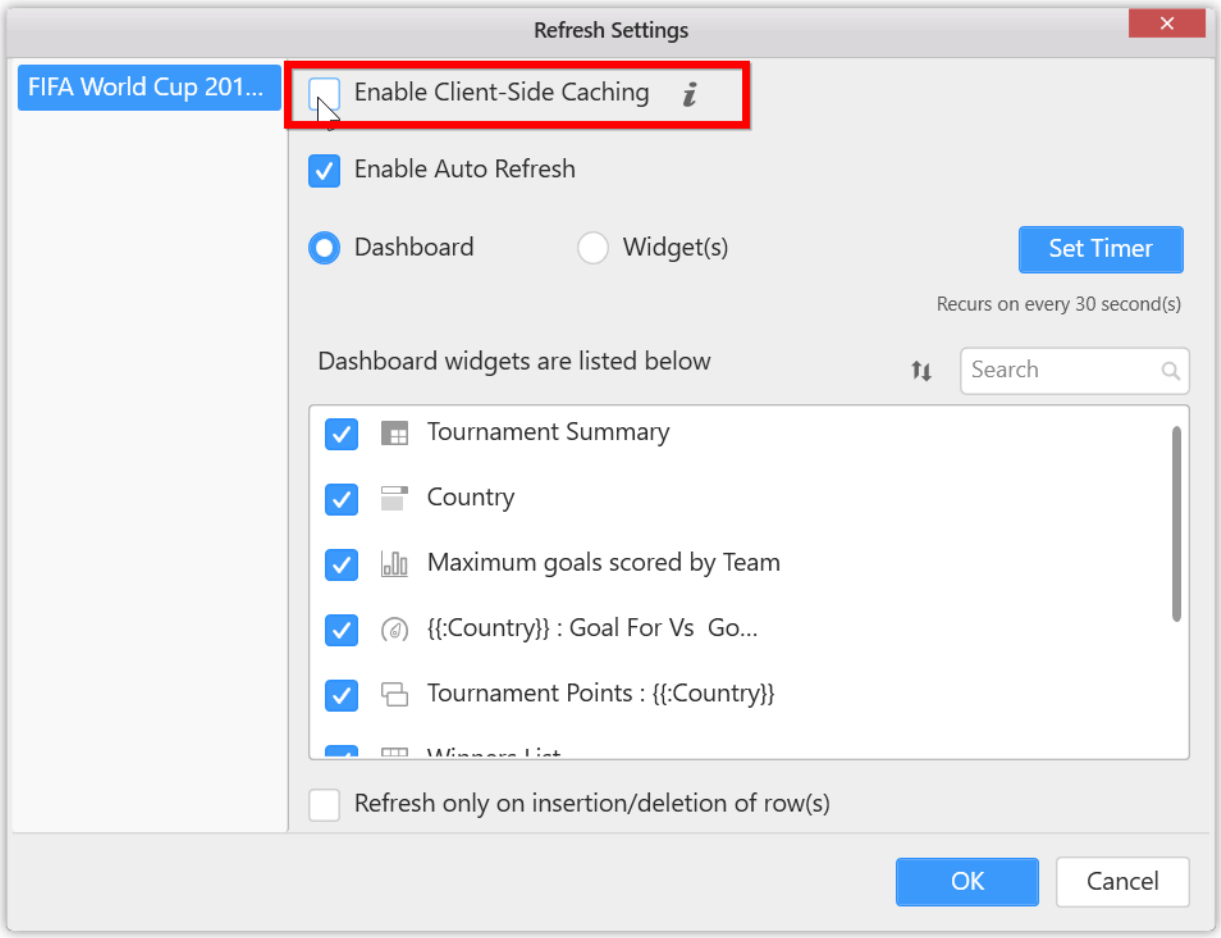
Now click OK to apply the settings and **save** your dashboard report.

Note: The User Interface shown in this document is applicable for the 3.1 Dashboard Designer version.If, you are using a lower version of the dashboard designer application the UI and options may differ.

How to enable client-side caching

The Enable Client-Side caching option enables or disables the client-side cache. If the option is enabled, it maintains the client-side cache for the master - slave widget interactions made in the dashboard viewer/server. So, when there is an interaction in the dashboard, and if the same filters were already used in the same dashboard, data will not be fetched from the server again. It will be used from the client-side cache memory, so the dashboard loads faster. If the option is disabled, it does not maintain the client-side cache for the interaction data.

To enable the Client-Side Caching option check the option like shown in the following screenshot. Un-checking the option will disable the caching.



Server Explorer

Server explorer option is provided to log on Dashboard Servers and Data Integration Platform servers through the UMS. Using this server, you can explore their databases and connections that are available in DIP servers of the Dashboard Designer.

Dashboard Server

The **Dashboard Server** is used to organize and share Dashboards through a web interface. Dashboards and data sources can be shared to other authorized users from the Dashboard Designer through the Dashboard Server in which they have access.

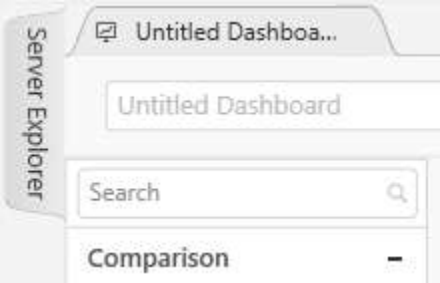
Data Integration platform

Data Integration Platform has been built to automate the flow of data between systems. So that, you can import dashboards, data sources (also create data sources), and widgets using the Rest API from the server.

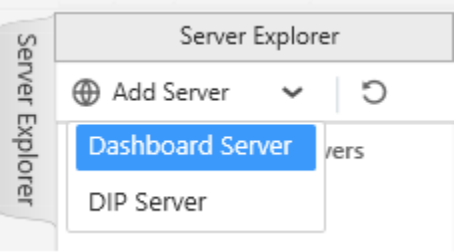
The Dashboard Designer application provides multi-user access to login to the server. Dashboards and data sources can be shared to other authorized users from the Dashboard Designer through the Dashboard Server in which they have access.

Server Explorer

To open the server explorer, Click Server Explorer that is present in the left side of the Dashboard Designer.

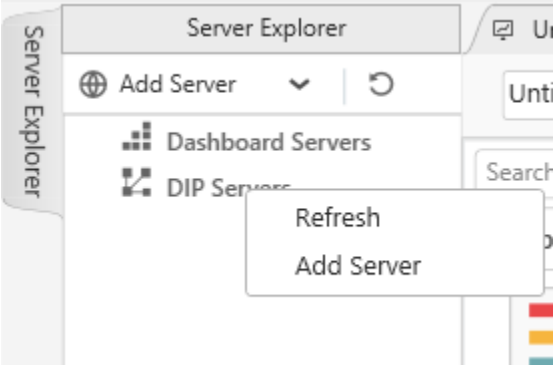


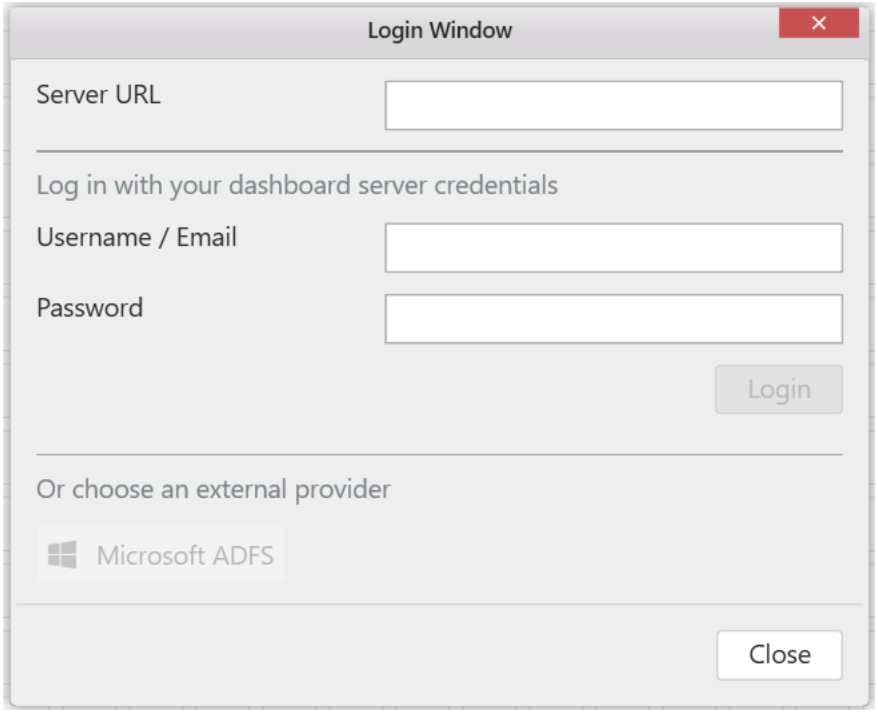
Automatically, the server explorer will be expanded. In this, you can find the option for adding the Dashboard Server and the DIP server.



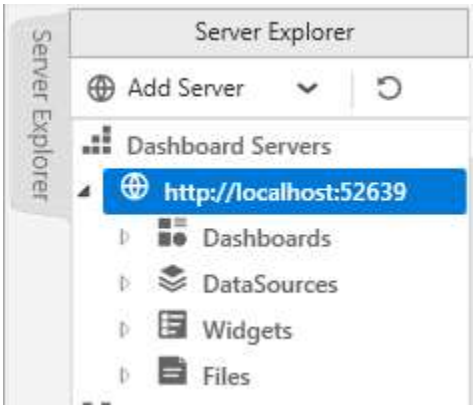
Logging into Dashboard Server

Choose Dashboard server menu from the Add server drop-down or choose Add Server by right-clicking the Dashboard servers option, the Dashboard Server Login Window opens.





Now, the server is added under the Dashboard Servers category in tree view structural format.



Note: The added Dashboard Server will not affect the current user's login and their information through server explorer that are not maintained in Manage Accounts. This information will be shown in the server explorer alone.

Dashboard Server login through application menu/title bar/manage accounts

Add the Dashboard Server through any one of the following mentioned options:

- Application Menu
- Title Bar
- Manage Accounts

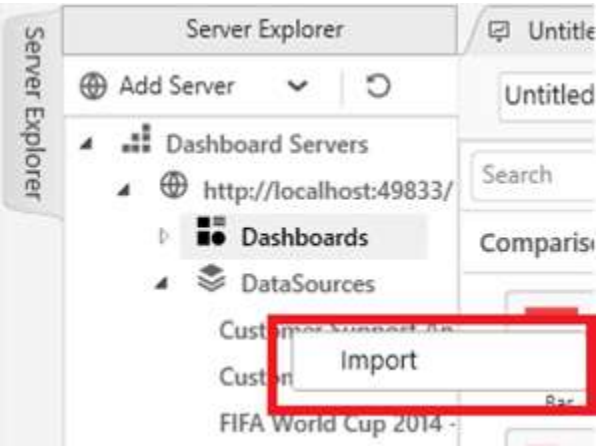
The logged user will be automatically displayed in the server explorer under Dashboard Server category as a default user. This will also be applicable for switching users. If you switch the user account from one to another, the old user will be removed from the server explorer and the newly switched user will be displayed as a first item.

Note: If added user is deleted or modified from the above-mentioned place, the newly switched user will also be affected in the server explorer.

Import and refresh Dashboard Server

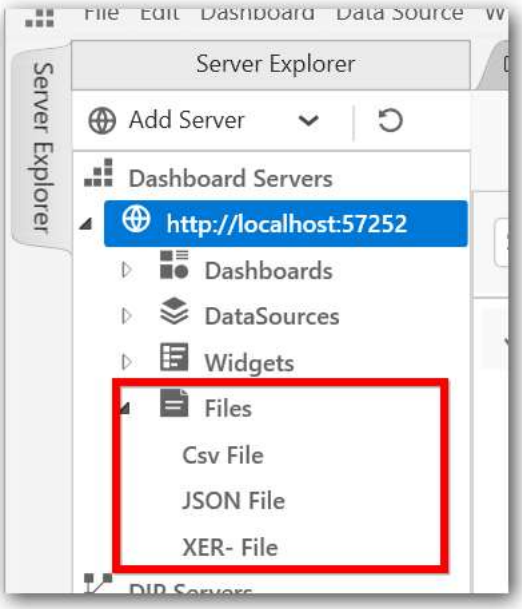
Import dashboard, data source, and widgets from server explorer

Right-click the dashboard/data source/widget from the tree view. The selected item will be imported from the server.

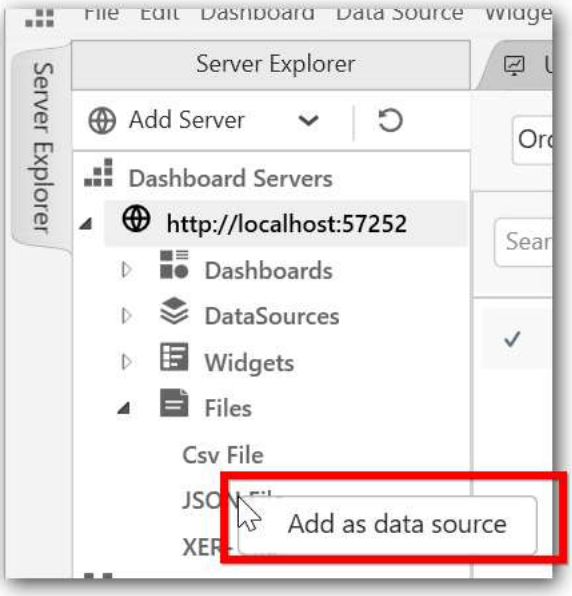


Import data source from the `Files` category

You should add the file type such as Excel, CSV, JSON, Text, or Xer document in the Dashboard Server. So that, the data source is added under files category either as server login or refresh.

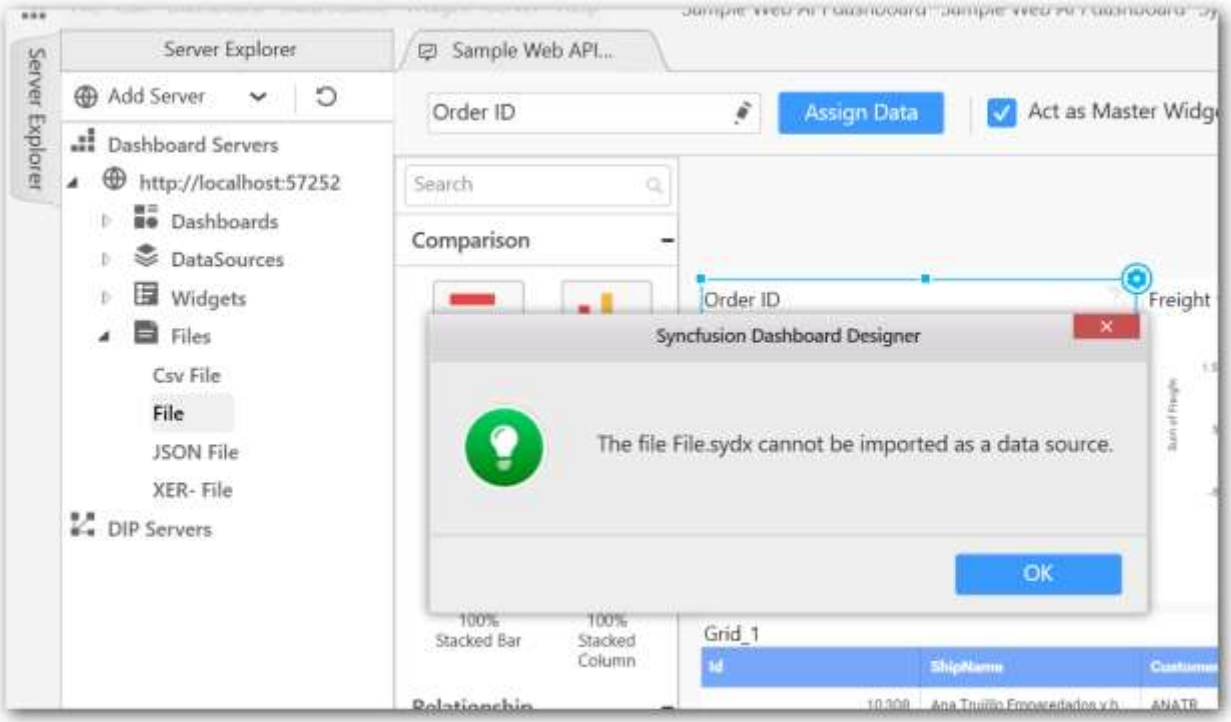


To import data source, right-click any one of the data sources under files category from the tree view. By clicking the Add as data source, the selected data source item will be imported from the server.



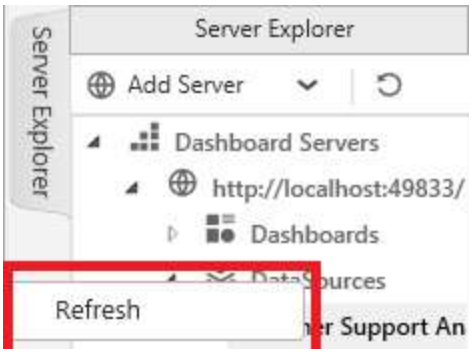
After adding the data source from server explorer, the data source will act as online data source.

If the file is invalid, the following information message will be displayed.



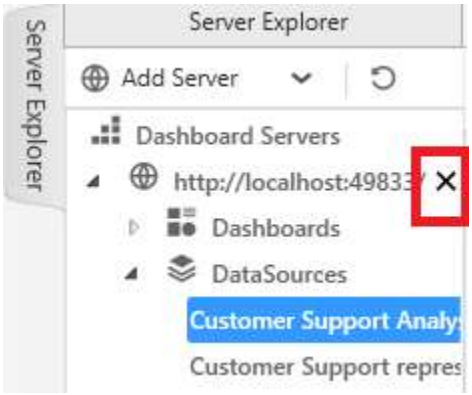
Refresh the dashboard, data source, widgets, and files

Right-click any one of the Dashboard Server URL or dashboard/data source/widget/files from the tree view. The selected item will be refreshed from the server.



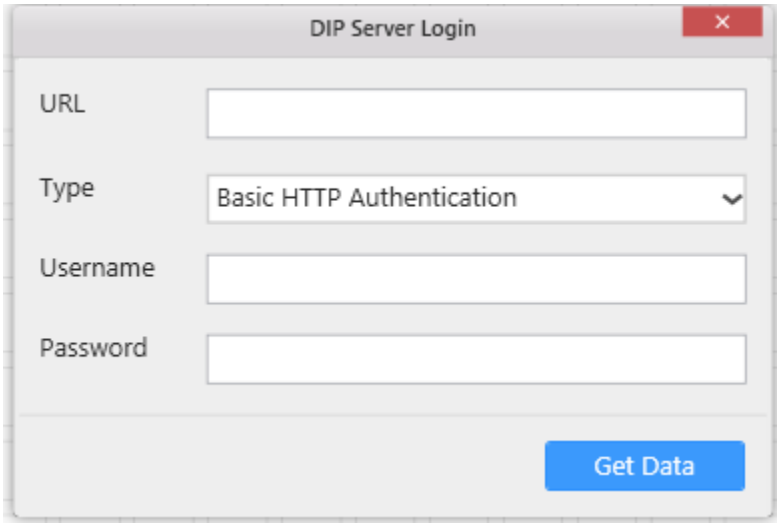
Remove option

Remove button will be displayed only if the Dashboard Server is logged through the server explorer option. Otherwise, the remove button will not be displayed in each server URL. The selected dashboard server is removed from the tree view after clicking the remove button.

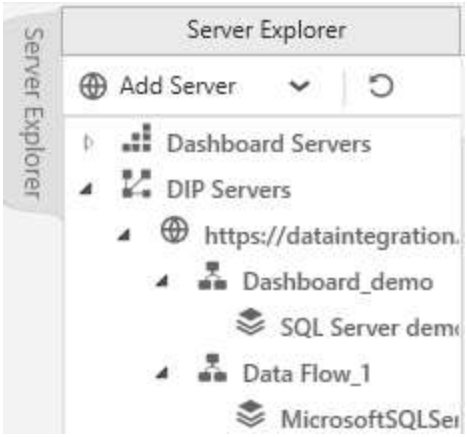


Logging into DIP server

Click the Add server drop-down and choose the DIP server menu. The DIP Server login window will open. Or else, right-click the DIP server item and click the add server button.



Automatically, the added server will be displayed in the DIP Server tree view item along with the data flow. The data flow is added to the specified server with database connections in the designer application.



Note: Need to give the server URL in the following mentioned format:
Normal DIP installation without UMS: Syntax: http://{hostname}:{port number}

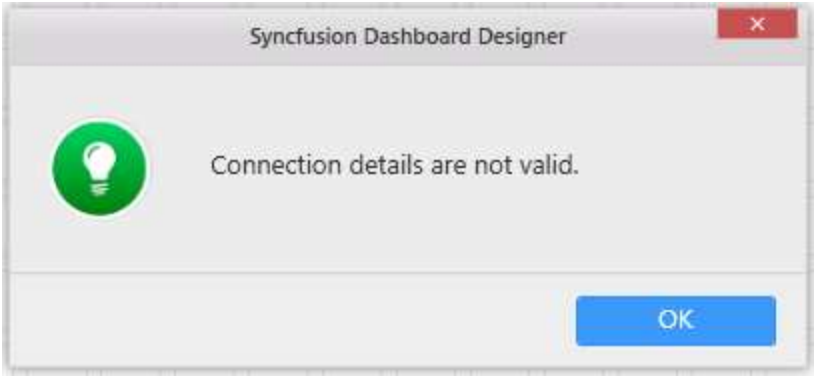
Normal DIP installation with UMS: Syntax: https://{hostname}:{port number}

Create Data source and Refresh DIP server

Create data source option

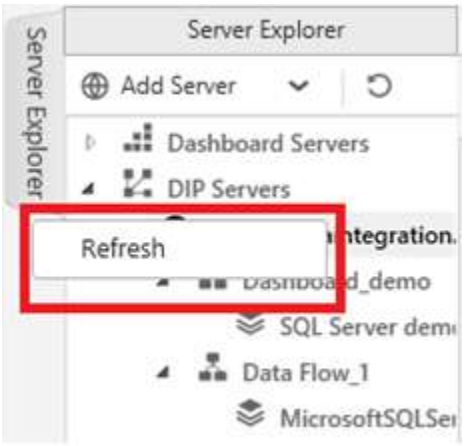
Right-click any one of the connections under the data flow from the tree view. The selected data source will be created based on the connection details provided in the API. Automatically, the data source tab will be opened and tables are added to the canvas if tables are available in the connection.

Note: If the invalid connection is provided in the data flow while creating the data source from the DIP server, the following alert message will be displayed.



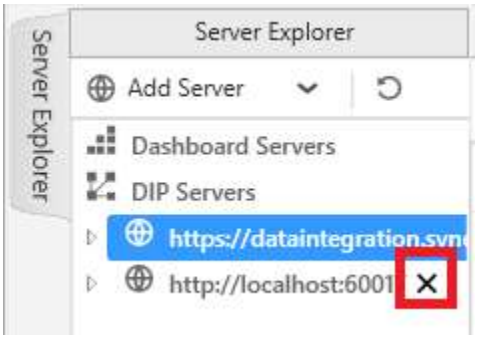
Refresh option

Right-click any one of the DIP server URL from the tree view. The selected item present in the selected URL will be refreshed from the server.



Remove option

The selected DIP server is removed from the tree view after clicking the remove button.



Note: If the LAN is disconnected, the selected user's sign-in should not be shown until the connection is established, and you can see the message as "Unable to connect to the remote server".

User details persistence

All added details of Dashboard Server and DIP server credentials will be maintained to display all saved credentials in the server explorer tree view whenever the application starts. The user account information will be retained, even if you uninstall the application.

Keyboard Shortcuts

Syncfusion Dashboard Designer allows you to create dashboard using Keyboard (without mouse). You can use the following shortcut keys to work with dashboard.

Menu Shortcut keys

Action	Shortcut Key
Create New Dashboard	`Ctrl + N`
Open Dashboard	`Ctrl + O`
Close current Dashboard	`Ctrl + W`
Close Current Tab	`Ctrl + T`
Save Dashboard Report	`Ctrl + S`
Save Dashboard Report as/td>	`Ctrl + Alt + S`
Exit Application	`Alt + F4`
Undo	`Ctrl + Z`
Redo	`Ctrl + Y`
Copy	`Ctrl+ C`
Paste	`Ctrl+ V`
Preferences	`Ctrl+ Shift + A`
Delete selected control	`Del`
Preview	`F5`
Open Dashboard Parameters Window	`Ctrl + Shift + D`
Open Refresh Settings Window	`Ctrl + Shift + R`
Open Global Filter panel	`Ctrl + Shift + G`
Open Dashboard Settings Window	`Ctrl + Shift + S`
Open Filter Action Window	`Ctrl+ F`
Create New Data source	`Ctrl + Shift + N`
Import Data Source	`Ctrl + Shift + I`
Move Next Tab	`Ctrl + Tab`
Open User Filters Window	`Ctrl+ Shift + U`
Open Dashboard from Dashboard server	`Ctrl + Shift + O`
Open Dashboard from Dashboard server	`Ctrl + Shift + O`

Publish Dashboard	`Ctrl + Shift + P`
Publish Data source	`Ctrl + Alt + P`
Publish Widget	`Shift + Alt + P`
Import Data source from Dashboard server	`Ctrl + Alt + O`
Help	`F1`
Control Selection Change	`Tab`
Open control Properties tab	`F4`
Cancel drag operation	`Esc`
To navigate	`Tab`
Reverse navigation	`Shift + Tab`
Close the current opened window	`Esc`
Update current window	`Enter`
To rename the report	`Ctrl +R`
Open sub menu of File	`Alt + F`
Open sub menu of Edit	`Alt + E`
Open sub menu of Data sources	`Alt + T`
Open sub menu item of Widget	`Alt + W`
Open sub menu item of Dashboard	`Alt + D`
Open sub menu of Server	`Alt + S`
Open sub menu item of Help	`Alt + H`

Dashboard Design Page shortcut keys

Action	Shortcut Key
To move control use	`Arrow keys`
To resize control use	`Shift + Arrow`
To move next or previous control use	`Ctrl + Arrow`

Data Design Page shortcut keys

Action	Shortcut Key
To open Join editor	`Ctrl + J`

To open expression editor	`Ctrl + E`
To open Initial filter	`Ctrl + I`

Expression editor shortcut keys

Action	Shortcut Key
Add new expression	`Ctrl + I`
Delete selected expression	`Ctrl+ D`
To add selected function or column	`Space`

Control Designer shortcut keys

Action	Shortcut Key
To navigate properties tab	`Ctrl + P`
To navigate data tab	`Ctrl + D`
To update the widget when defer update is enabled	`Ctrl + L`

Label Settings Window shortcut keys

Action	Shortcut Key
Change font style	`Ctrl + Shift + F`
Increase font size	`Ctrl + Shift + F3`
Decrease font size	`Ctrl + Shift + F2`
Label vertical alignment	`Ctrl + Shift + W`
Add hyper link	`Ctrl + Shift + H`
Remove hyper link	`Ctrl + Shift + Q`
Clear format	`Ctrl + Space`

Default Keys Used for Navigation

Action	Shortcut Key
Default Navigation	`Tab` and `Shift + Tab`
Tree view	`Ctrl + Tab` and `Shift + Ctrl + Tab`
To open drop down list	`F4`
To check and uncheck the check box and radio button	`Space`

Open context Menu	`Apps`
Select and click	`Enter`

Localization

Localization is the process of translating an application's user interface based on specific cultures.

Localizing Dashboard Designer

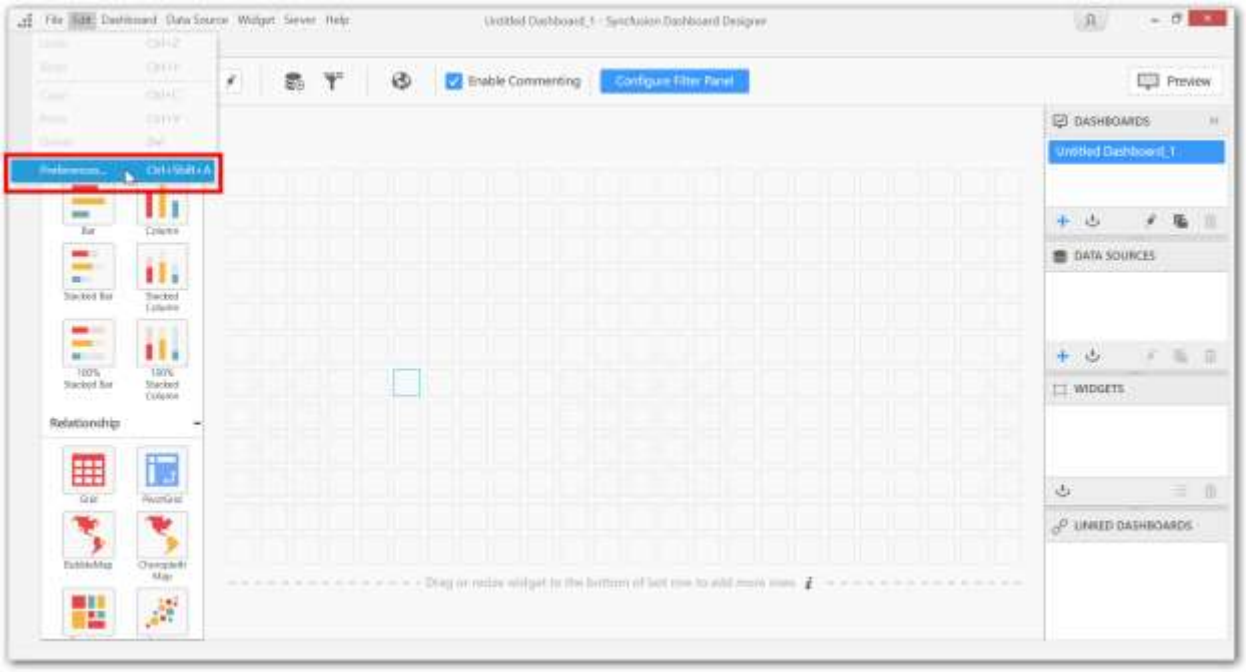
You can localize the Syncfusion Dashboard Designer by placing the resource file (.resx). This file is modified as per your preferred culture in the Localization folder within the installed location of Dashboard Designer application.

To localize the Syncfusion Dashboard Designer using resource file, follow the below steps

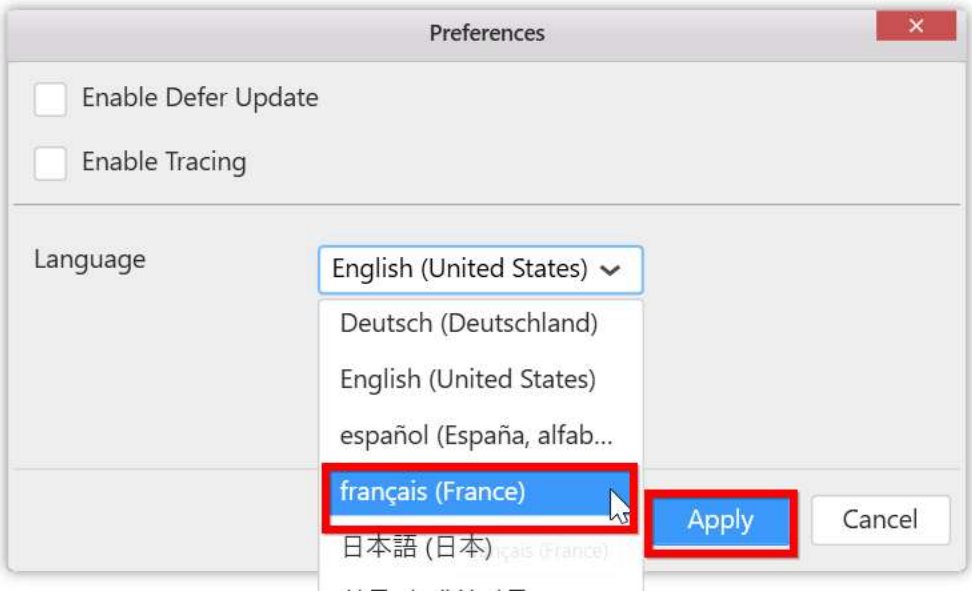
- Create a copy of the default resource file (**Resources.en-US.resx**) of Syncfusion Dashboard Designer. This designer is available in the Localization folder at the installed location of the Dashboard Designer application. Place the created file in the same location.

C:\Program Files (x86)\Syncfusion\Dashboard Designer\DashboardDesigner\Localization

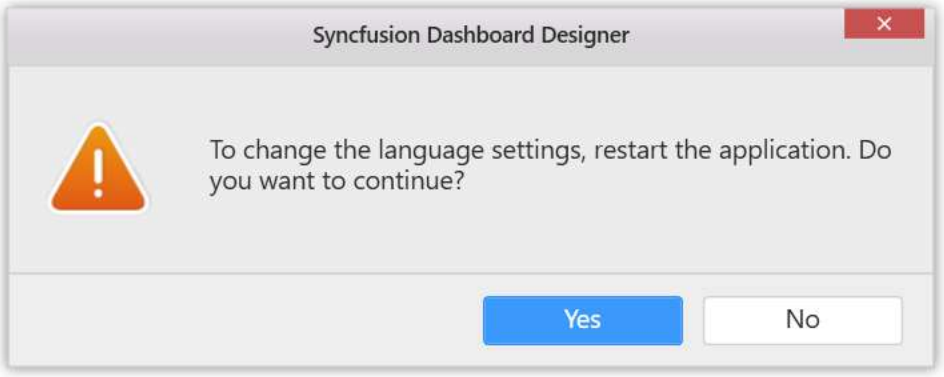
- Open the copied file using the Text Editor such as Notepad++. Each data row of the resource file contains a name, and value.
- Edit the value corresponding to each name based on the specific culture.
- Rename the file as Resources..resx. Here, the culture name illustrates the codes of language and country. For example, you have to specify file name as **Resources.fr-FR.resx** for **French** culture.
- Close any instance of Syncfusion Dashboard Designer application kept open, so as to get the above changes into effect.
- To change the language, open the Preferences window by clicking on the Preferences... menu item under the Edit menu.



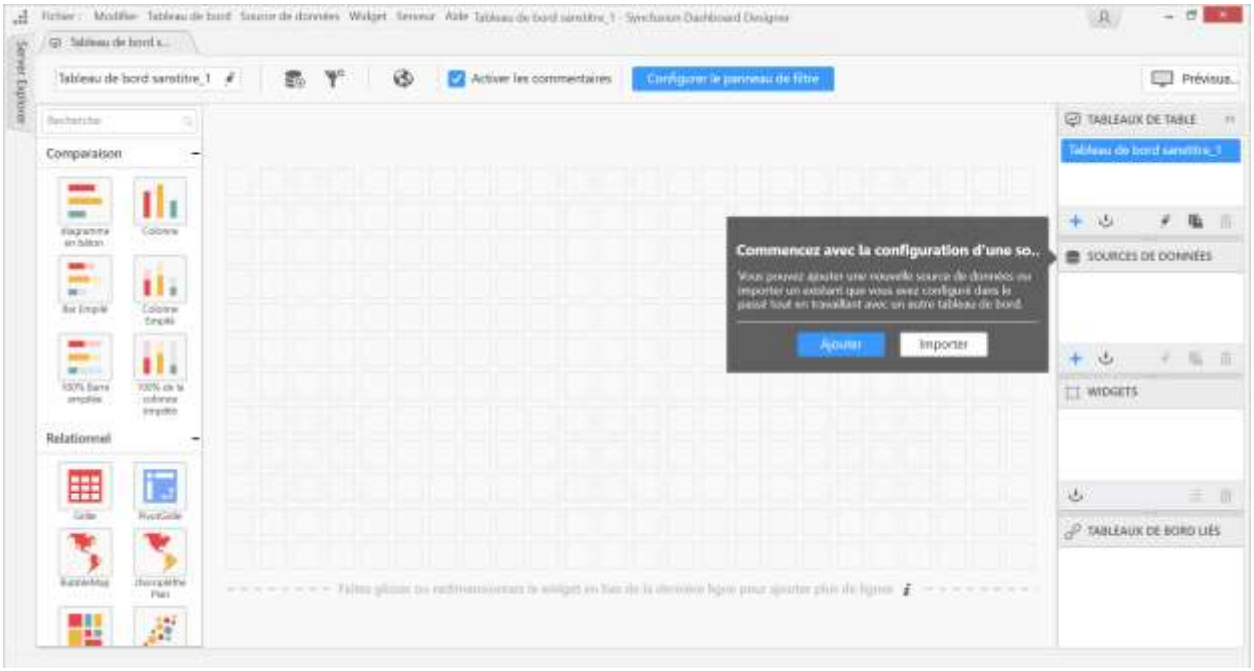
- The newly added language will be listed in languages list in the Language drop down list and select the language from that list and click on Apply button.



- This process requires closure of Syncfusion Dashboard Designer to get the selected language take into effect during next startup of the Dashboard Designer.



- After restarting the application the application texts will be shown in the selected culture.

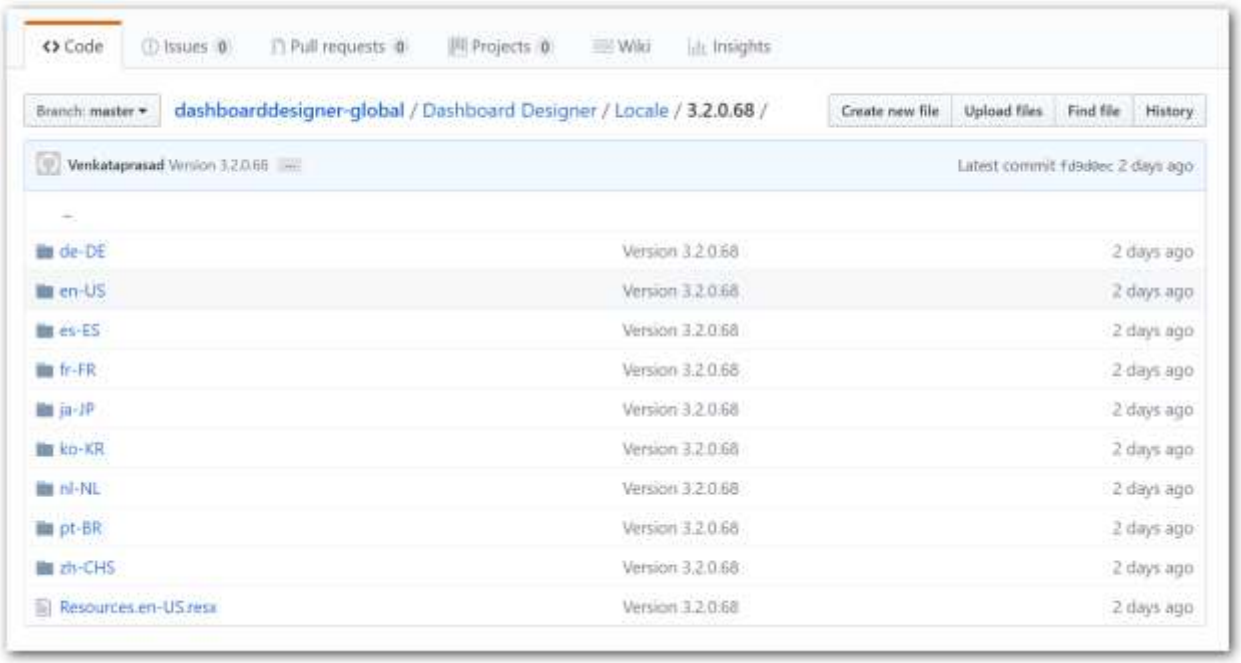


How to add new localizations for a new language resource file?

Read the documentation about how to add new localizations in the Syncfusion Dashboard Designer from the Syncfusion Dashboard Designer GitHub repository.

Sample localization files in GitHub repository

Syncfusion dashboard team generates and updates the localization files in the Github Syncfusion Dashboard Designer repository after every public release.



You can download and use the published resource files in the repository. The following are the languages available in the GitHub repository:

- Germany
- Spanish
- French
- Japanese
- Korean
- Dutch
- Portuguese - Brazil
- Chinese (Simplified)

How to edit localization messages for an existing language resource files?

Read the documentation about how to edit existing localizations in the Syncfusion Dashboard Designer from the Syncfusion Dashboard Designer GitHub repository.

Previewing Localized Dashboard from Syncfusion Dashboard Designer

Read the following documentation about how to add new localizations and how to edit existing localizations in the Syncfusion Dashboard Viewer.

How to add new localization

How to edit existing localizations

Custom Rebranding

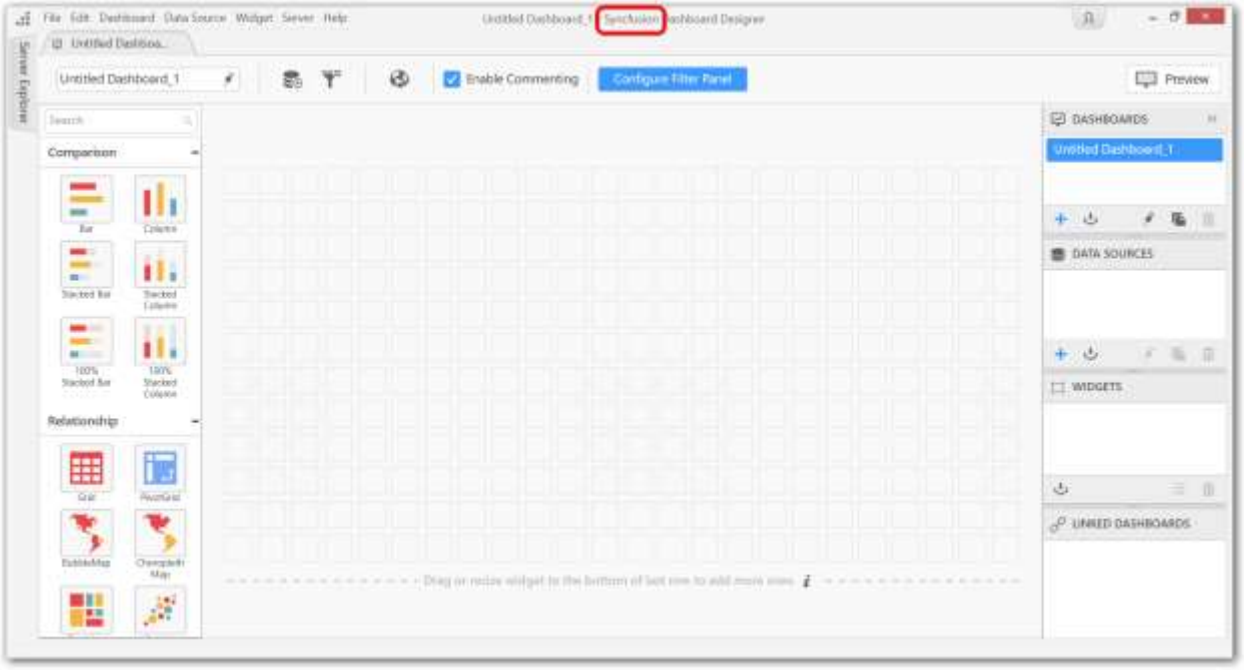
Syncfusion Dashboard Designer can be rebranded by changing the following items in the custom branding folder available in the Syncfusion Dashboard Designer installed location.

- Organization name
- Build version

- Product name
- Copyright information
- Product overview
- Company URL
- Help Document URL
- App icon
- Company logo
- Title icon
- URL image

Organization name

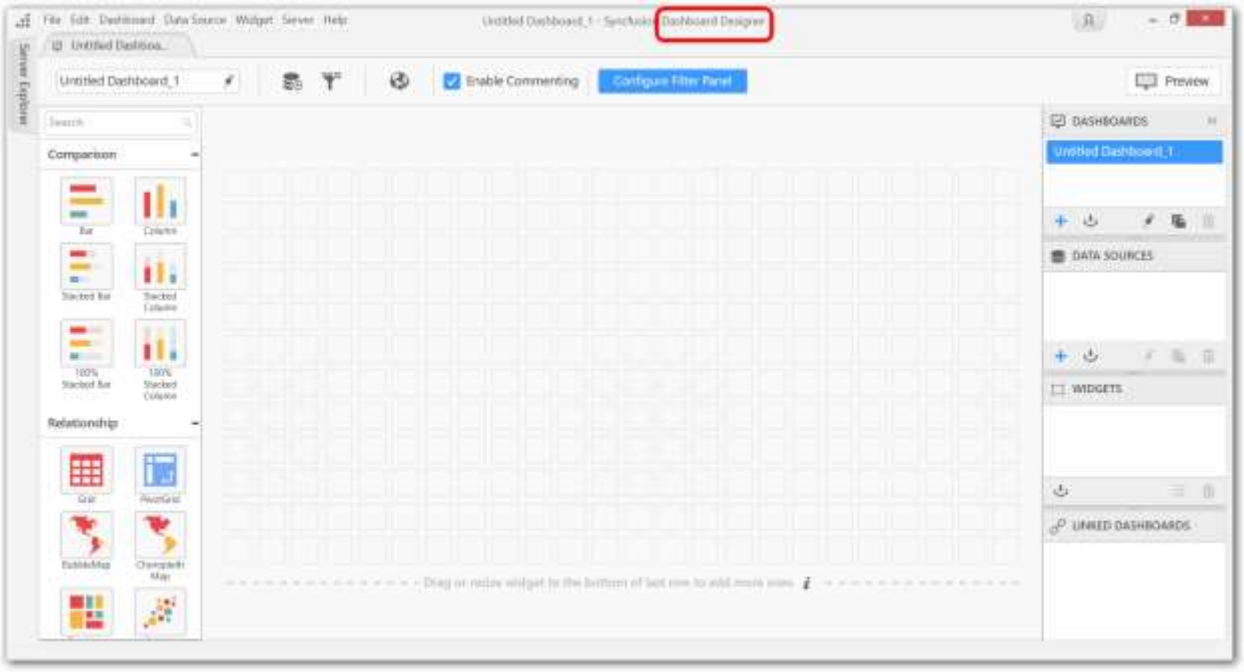
This setting applies change in the organization name defined in the title bar of the Main window and the About window.



Information: Special characters are not allowed in the organization name.

Product name

This setting applies change in the product name defined in the title bar of the Main window and the About window.



Information: Special characters are not allowed in the product name.

Build version

This setting applies change in the build version of Dashboard Designer in the About window.

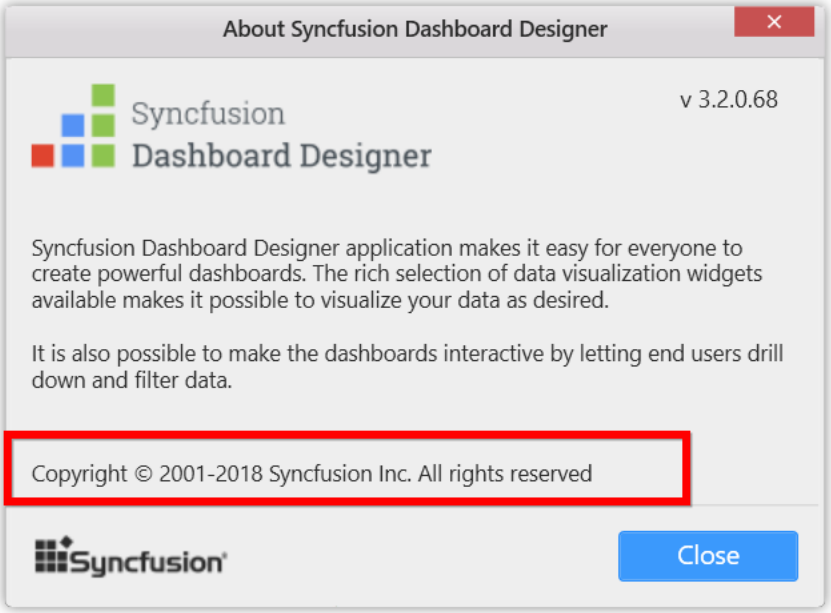
Information: The build version related changes will be reflected only when the organization name is changed to any name other than Syncfusion.

Product overview

This setting applies change in the overview of Dashboard Designer showcased in the About window.

Copyright information

This setting applies change in the copyright information of Dashboard Designer in the About window.

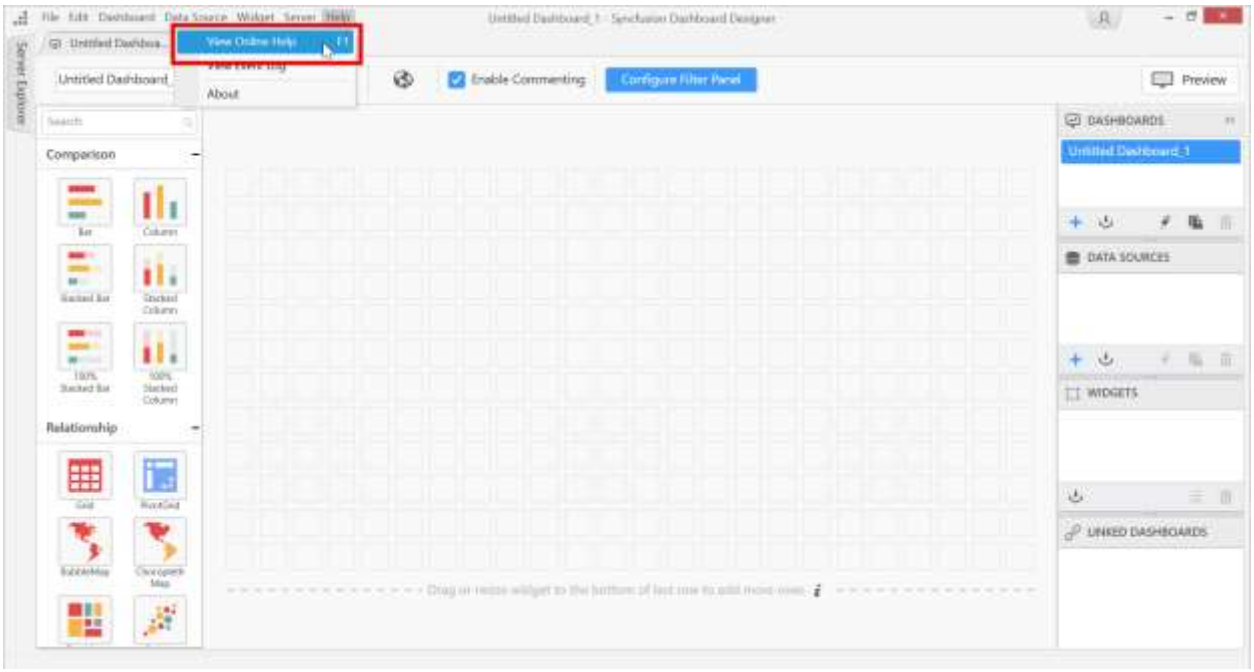


Company URL

This setting applies change in the Company URL showcased in the About window.

Help document URL

This setting applies change in the help documentation link while clicking the View Online Help.



App icon, company logo, title icon, and URL image

Setting these will apply changes in respective areas of the Main window and the About window.

Handling custom rebranding in Dashboard Designer

After installation of Dashboard designer, the **BrandingText.xml** file will be generated in the **CustomBranding** folder of the Dashboard Designer installed location.

C:\Program Files (x86)\Syncfusion\Dashboard Designer\DashboardDesigner\CustomBranding\BrandingText.xml

```
<DashboardDesigner>
<OrganizationName>Syncfusion</OrganizationName>
<ProductName>Dashboard Designer</ProductName>
<Version>11.0</Version>
<ProductOverview>
Syncfusion Dashboard Designer application makes it easy for everyone to create powerful dashboards. The rich selection of data visualization widgets
available makes it possible to visualize your data as desired.

It is also possible to make the dashboards interactive by letting end users drill down and filter data.
</ProductOverview>
<CopyRightInformation>Copyright © 2001-2018 Syncfusion Inc. All rights reserved</CopyRightInformation>
<CompanyUrl>http://www.syncfusion.com</CompanyUrl>
<HelpDocumentUrl>http://help.syncfusion.com/dashboard-platform</HelpDocumentUrl>
</DashboardDesigner>
```

Open this file and edit the respective attribute values such as, OrganizationName, ProductName, Version, ProductOverview, CopyrightInformation,Company URL, and Help Document URL that are to be rebranded.

Replace images that exists in the same folder such as, App icon, Company logo, Title icon, and URL image with required ones but with same name and dimensions.

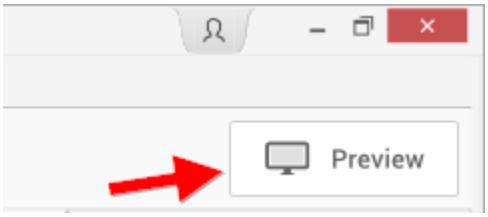
Image	File Name	Dimensions(pixels)
App Icon	Windows-AppLogo-16x16.ico	16x16
Company Logo	Windows-CompanyLogo-220x49.png	220x49
Title Icon	Windows-TitleLogo-16x16.png	16x16
URL Image	Windows-UrlLogo-100x25.png	100x25

Restart the Dashboard Designer application to get these changes reflected in the Dashboard Designer.

Previewing Dashboard

Previewing Dashboard

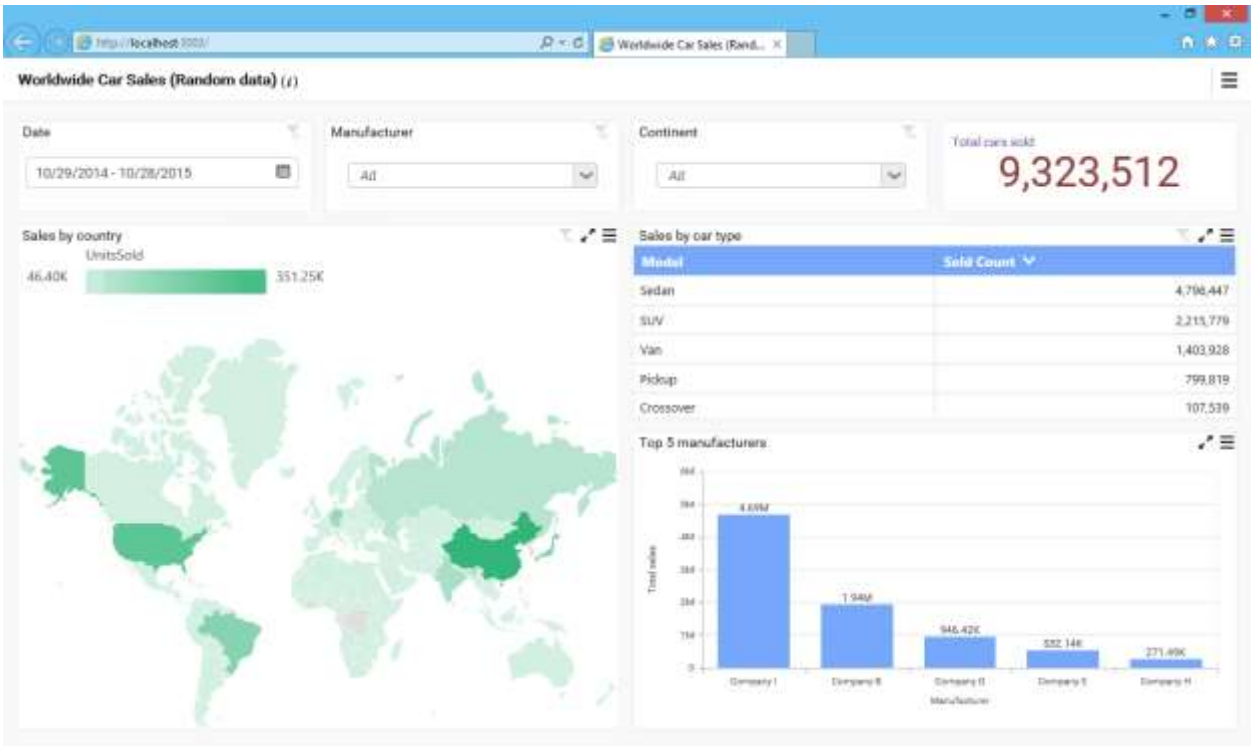
You can preview the currently opened dashboard through clicking the Preview at right corner in tools pane.



This preview button will be visible across all the window tab views in Syncfusion Dashboard Designer, thereby allowing you to preview the dashboard from anywhere within application.

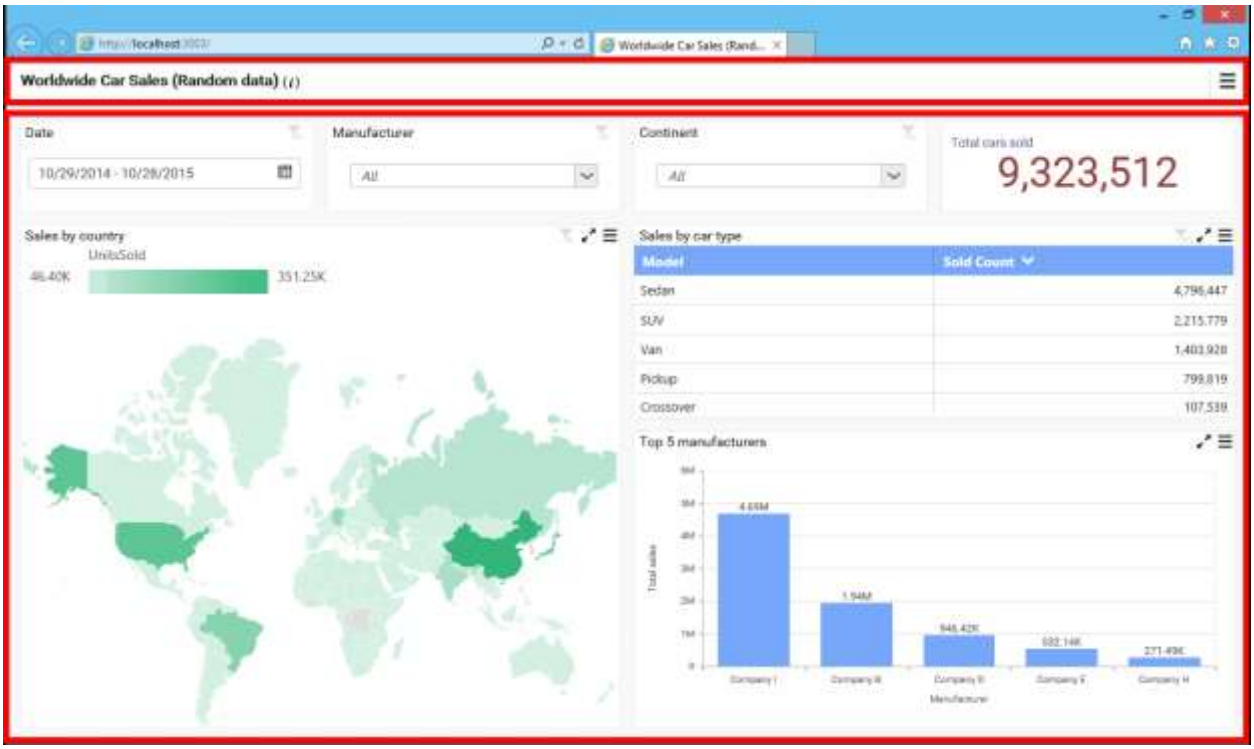
Previewing Dashboard using Dashboard Viewer

Dashboard was previewed using built-in Dashboard Viewer opened in your default web browser.

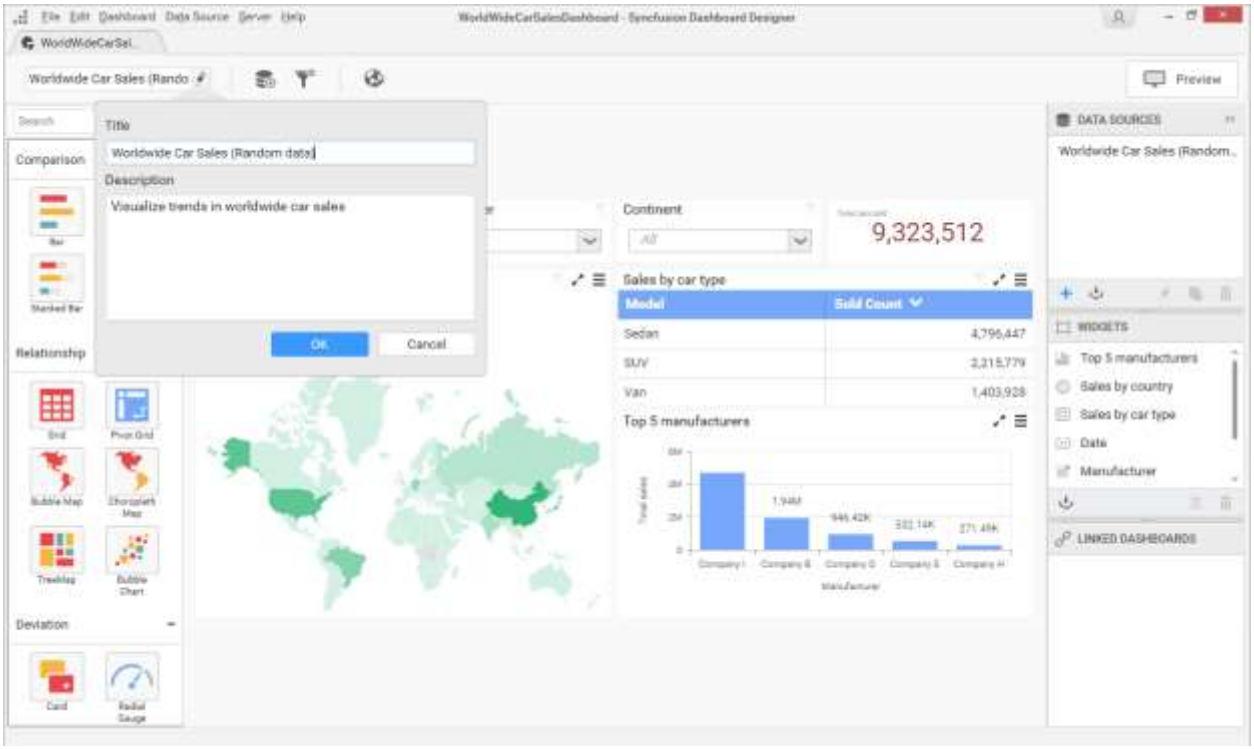


You can preview and interact the dashboard only if dashboard service is running in web development server in parallel. This service get started as discussed here.

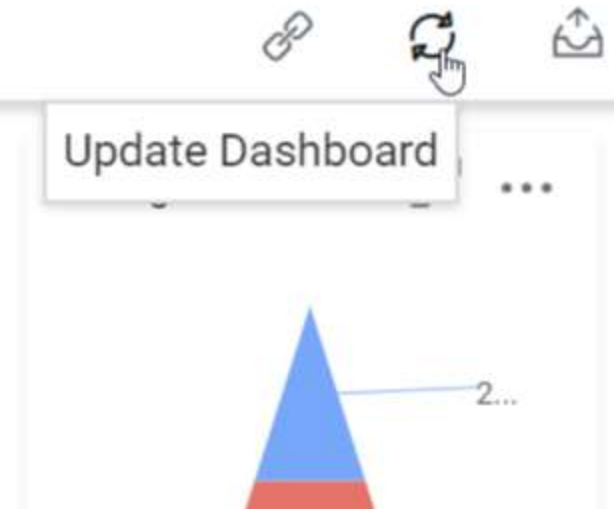
Dashboard Viewer can be split into two sections: Title and Content, like below.



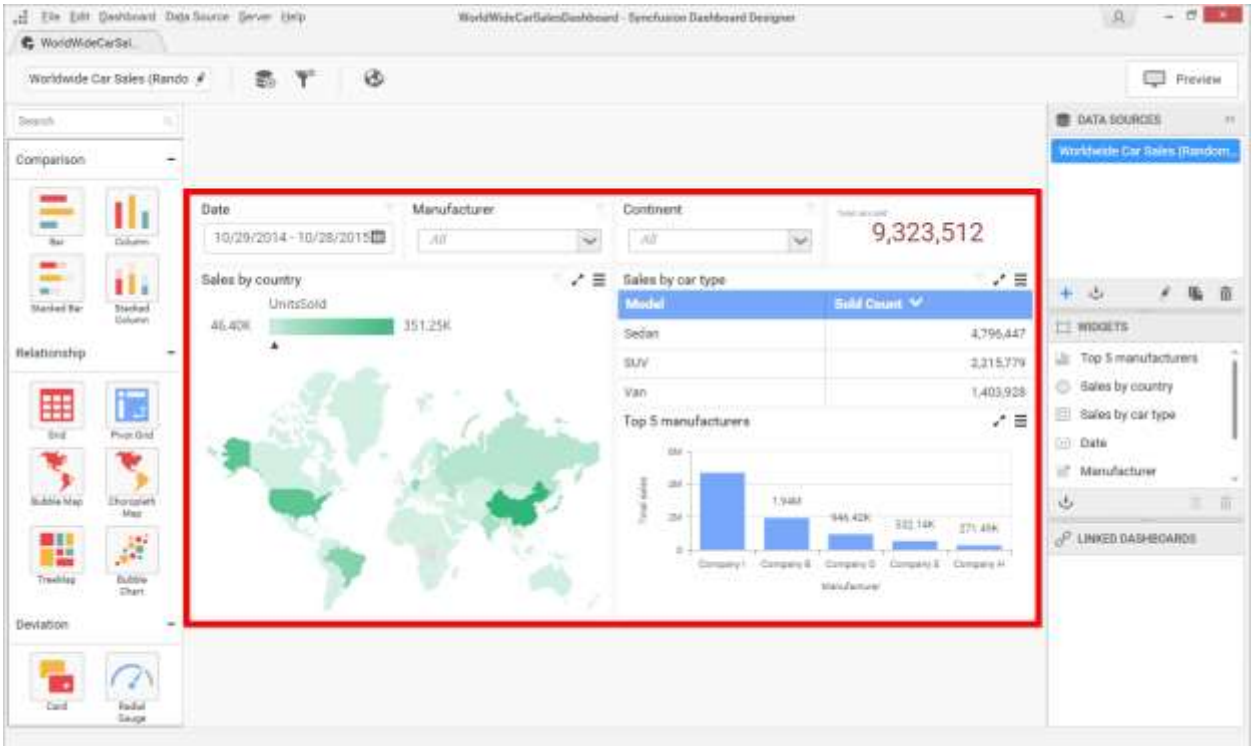
The Title section holds the dashboard title, description (can be viewed through clicking the i icon near to the title) that were set in Dashboard Designer before publish like below.



And, a menu to handle the dashboard update and export operations at runtime like below.

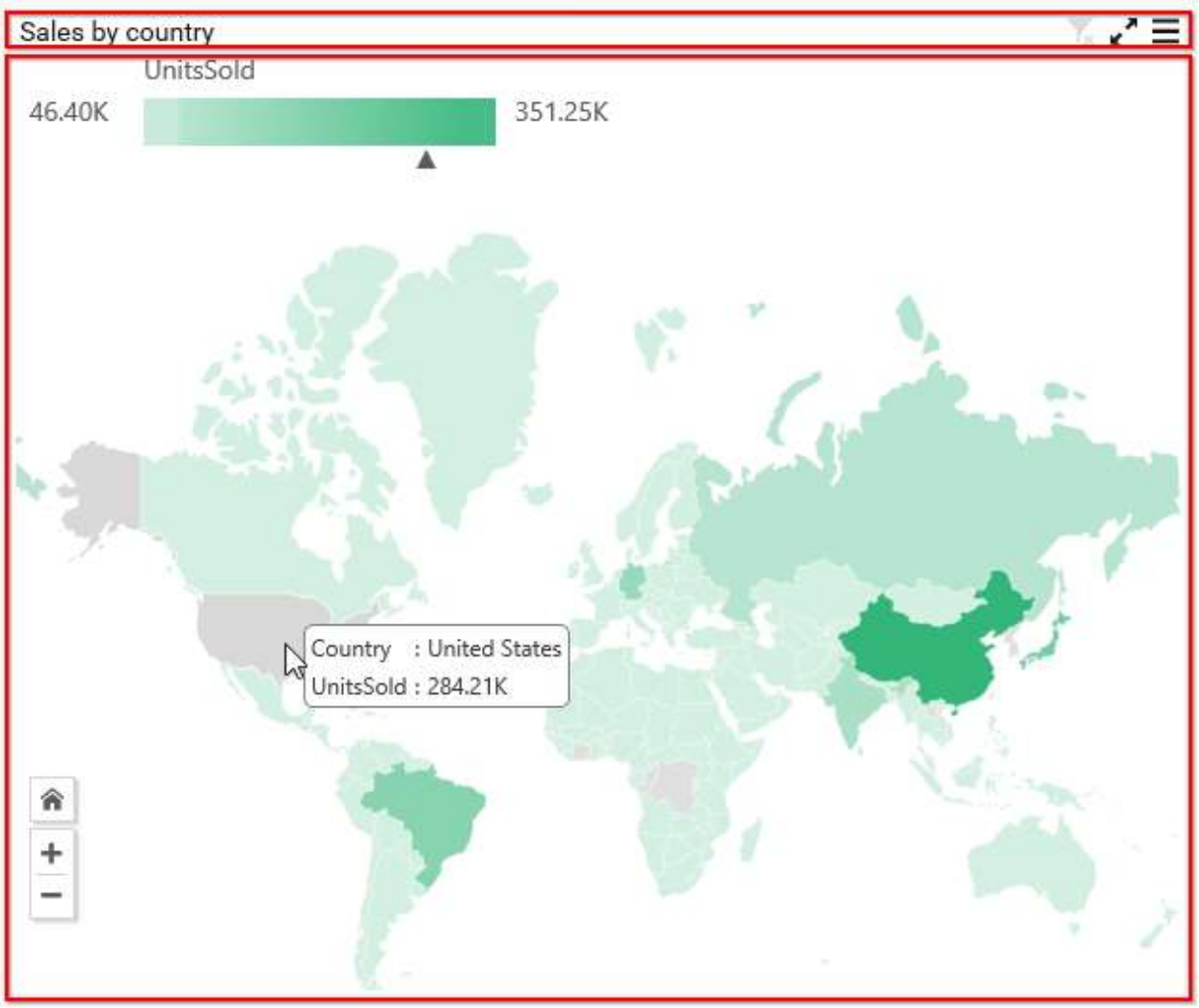


The Content section holds the complete content of the dashboard design area in Dashboard Designer.



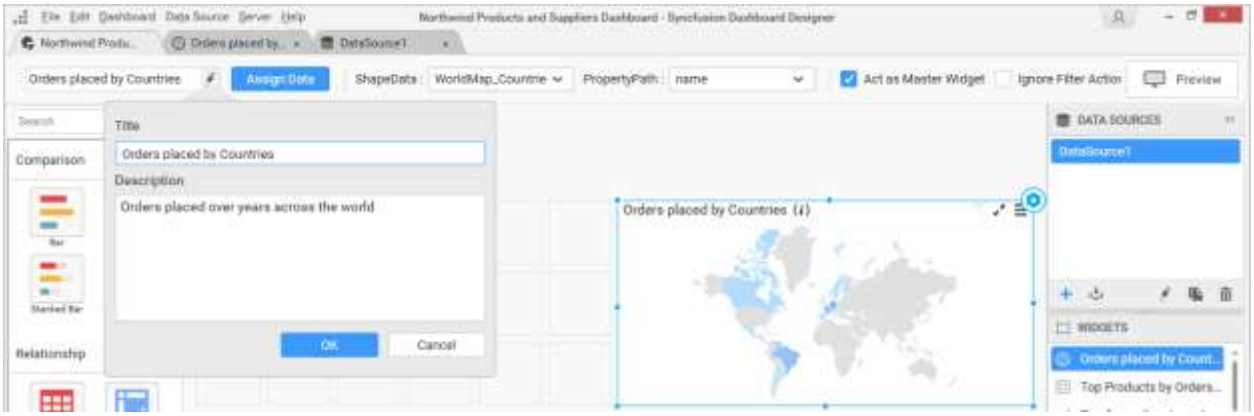
You can interact with the dashboard widgets in the viewer and apply filter on top of other widgets based on the dashboard filters configuration made in that dashboard through Dashboard Designer.

Each widget in dashboard viewer itself can split into two sections: Title and Content, like below.



The Title section holds the widget title, description (can be viewed through mouse hovering the i icon near to the title).

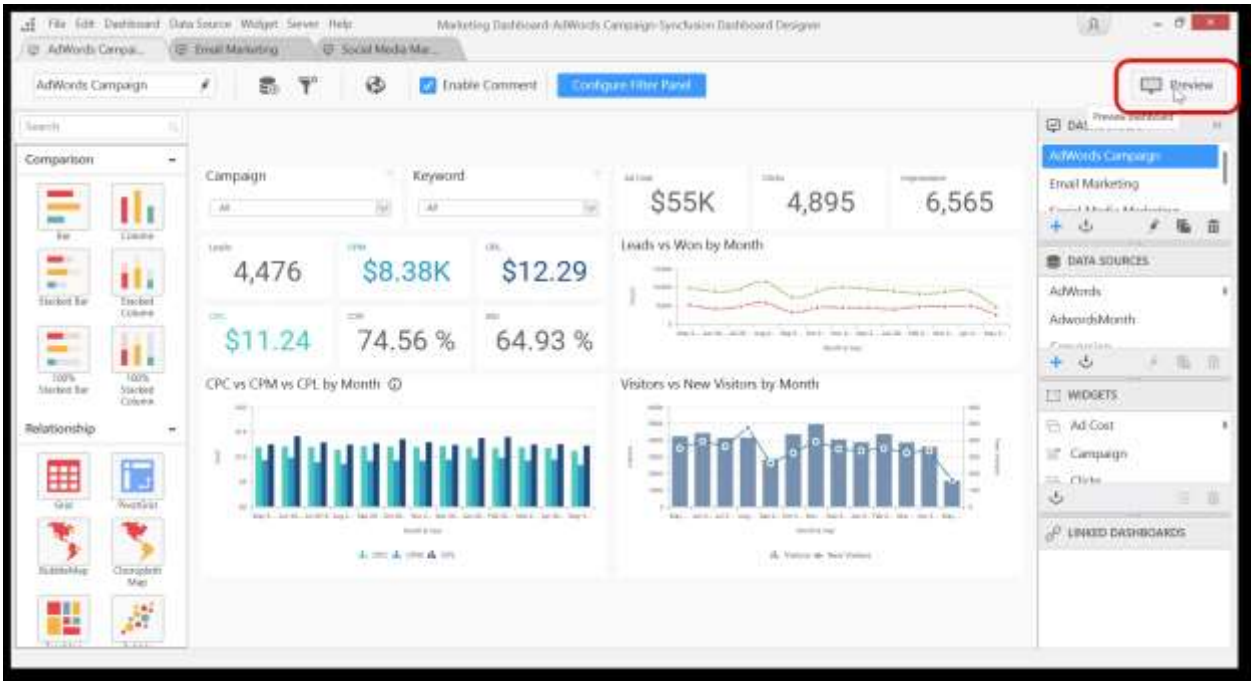
These were one that can be set through Dashboard Designer before publish like below.



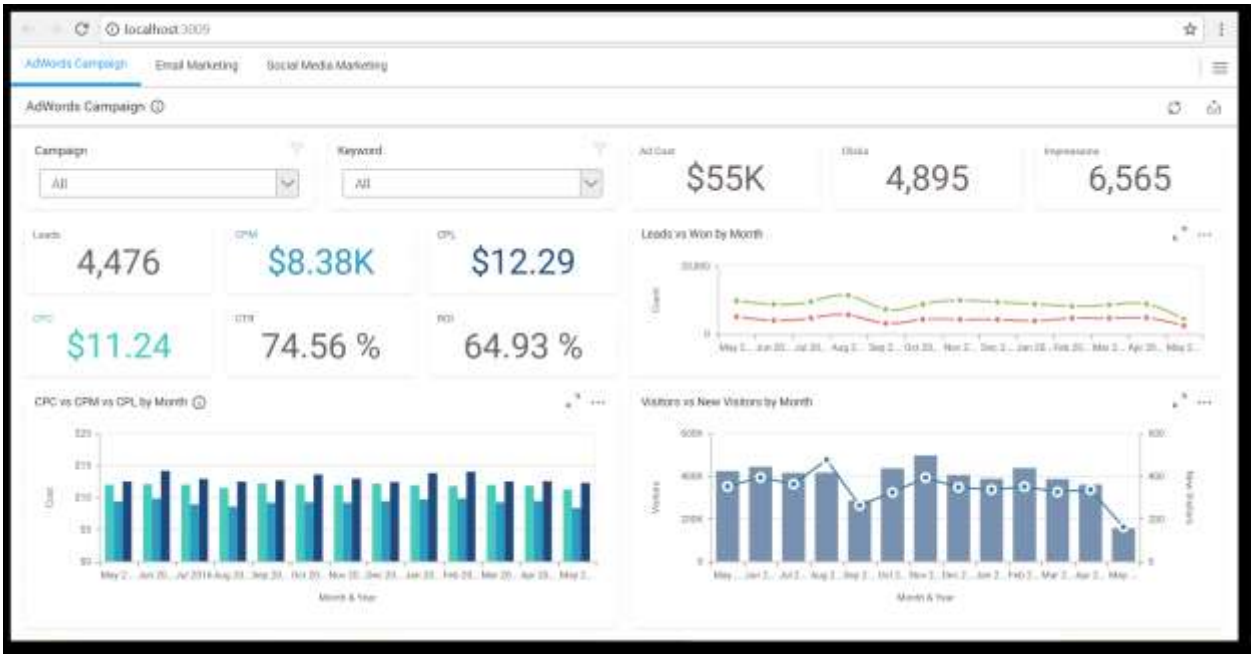
It also holds maximize, drop down menu, and filter status options at right end like below.

Previewing Multi-tabbed dashboards in Dashboard viewer

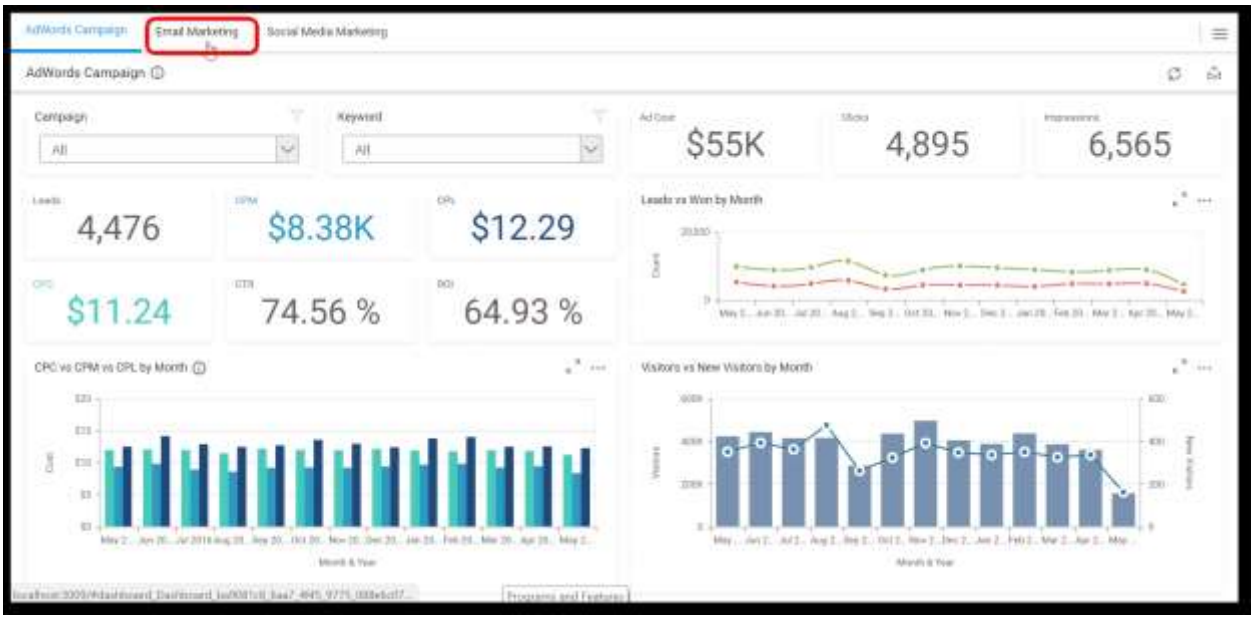
To preview the dashboards in the dashboard viewer, click on the preview option provided in top of the dashboard designer as shown in the below image,Also you can use the keyboard shortcut key F5.



Dashboard will be launched in your browser.



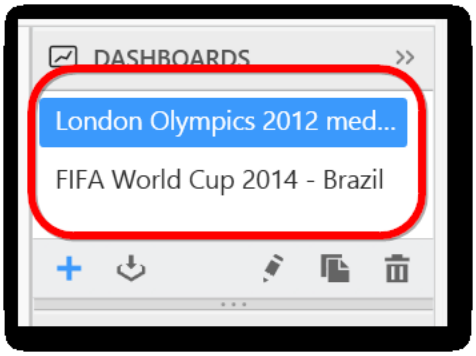
You can select the tab you want to preview.



Now the selected dashboard will be rendered.

Rearranging the Dashboard tabs

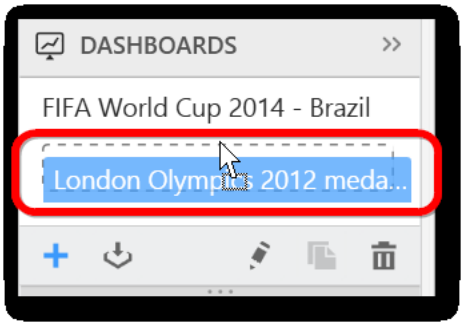
The Dashboard tabs will be previewed in the order as shown in the dashboard's container.



You can change the order of Dashboard by repositioning the dashboard list in the dashboard container using either drag and drop option or context menu.

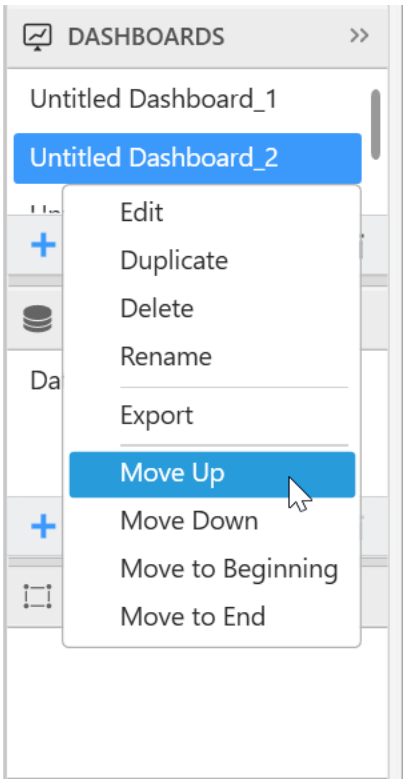
Rearrangement of dashboard tabs using the drag and drop option

Select the dashboard you want reposition and drag it to the place you want as shown in the below image.



Rearrangement of dashboard tabs using the context menu option

Right-click the dashboard tab name in the dashboard container to display the list of available options for rearranging the dashboard tabs.



Move Up: Moves the selected item to one position higher in the list. For example, the selected item is moved from 3rd tab to 2nd tab.

Move Down: Moves the selected item to one position lower in the list. For example, the selected item is moved from 2rd tab to 3nd tab.

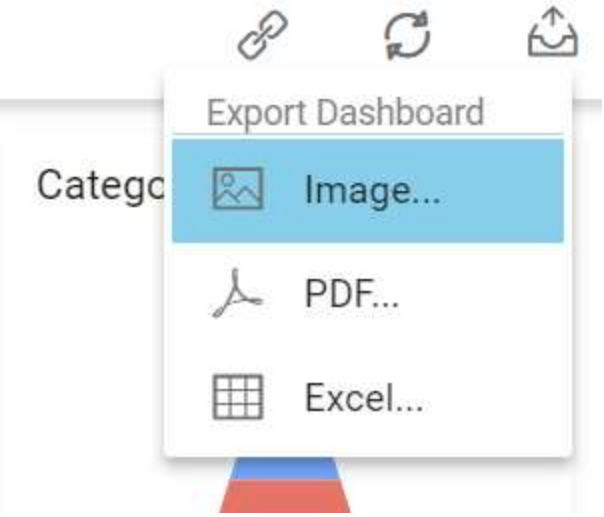
Move to Beginning: Moves the selected item from any position to first in the list.

Move to End: Moves the selected item from any position to last in the list.

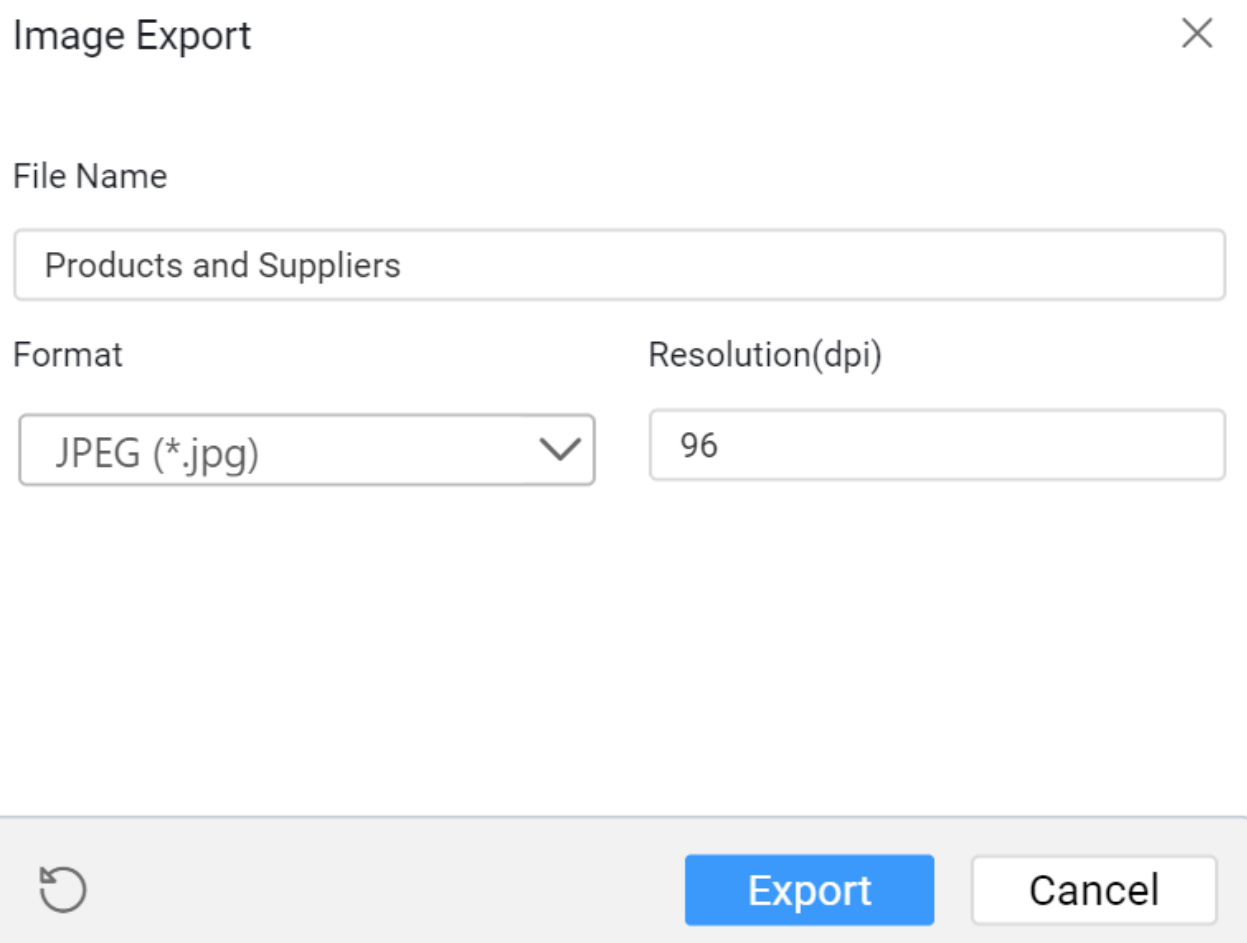
Dashboard Export Settings

Exporting Dashboard to Image

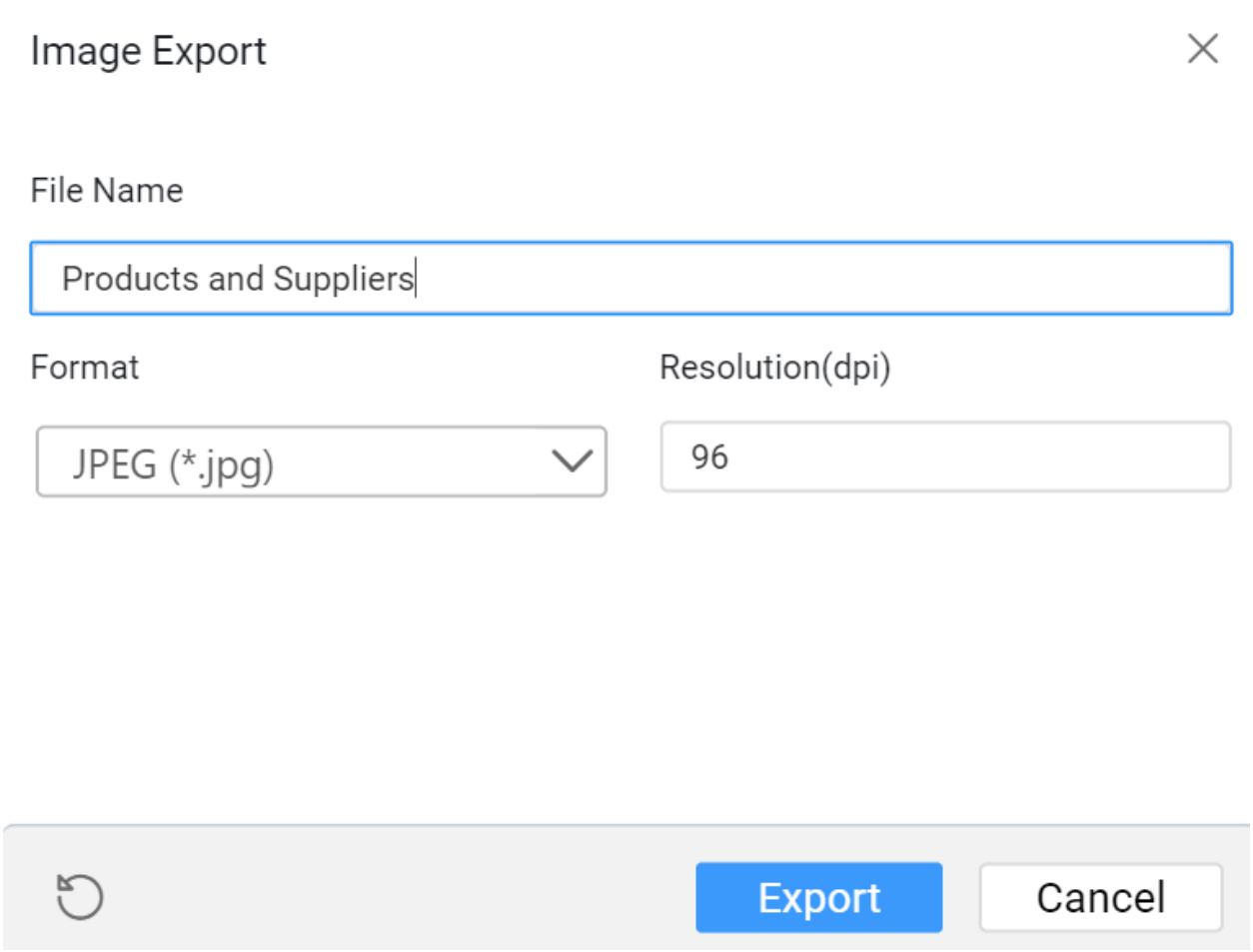
Export the current view of the dashboard in the form of image by clicking the Export icon at right corner of the title section.



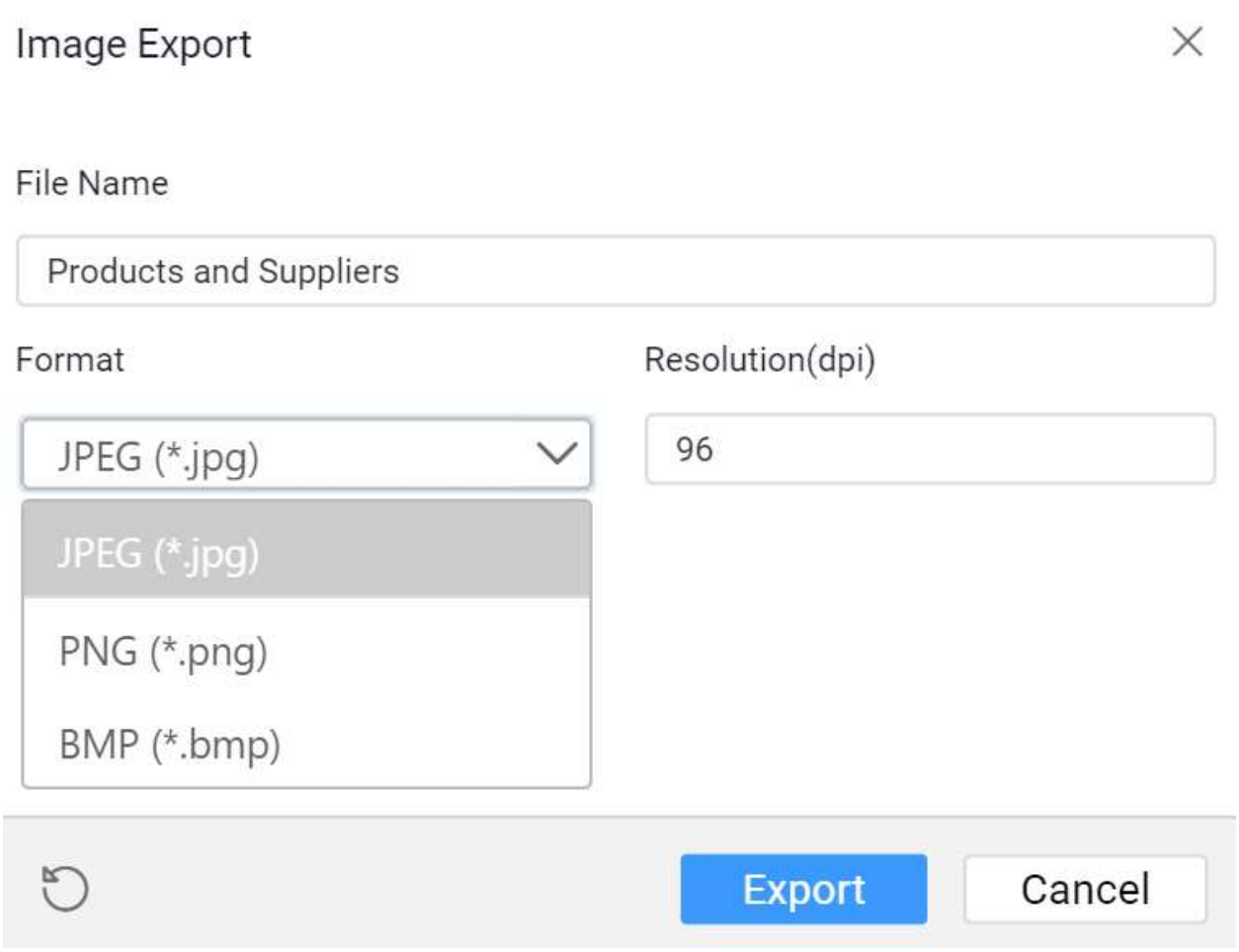
By clicking the Image option, the pop-up will be shown as follows.



Set the File Name fields with preferred values by replacing the default ones.

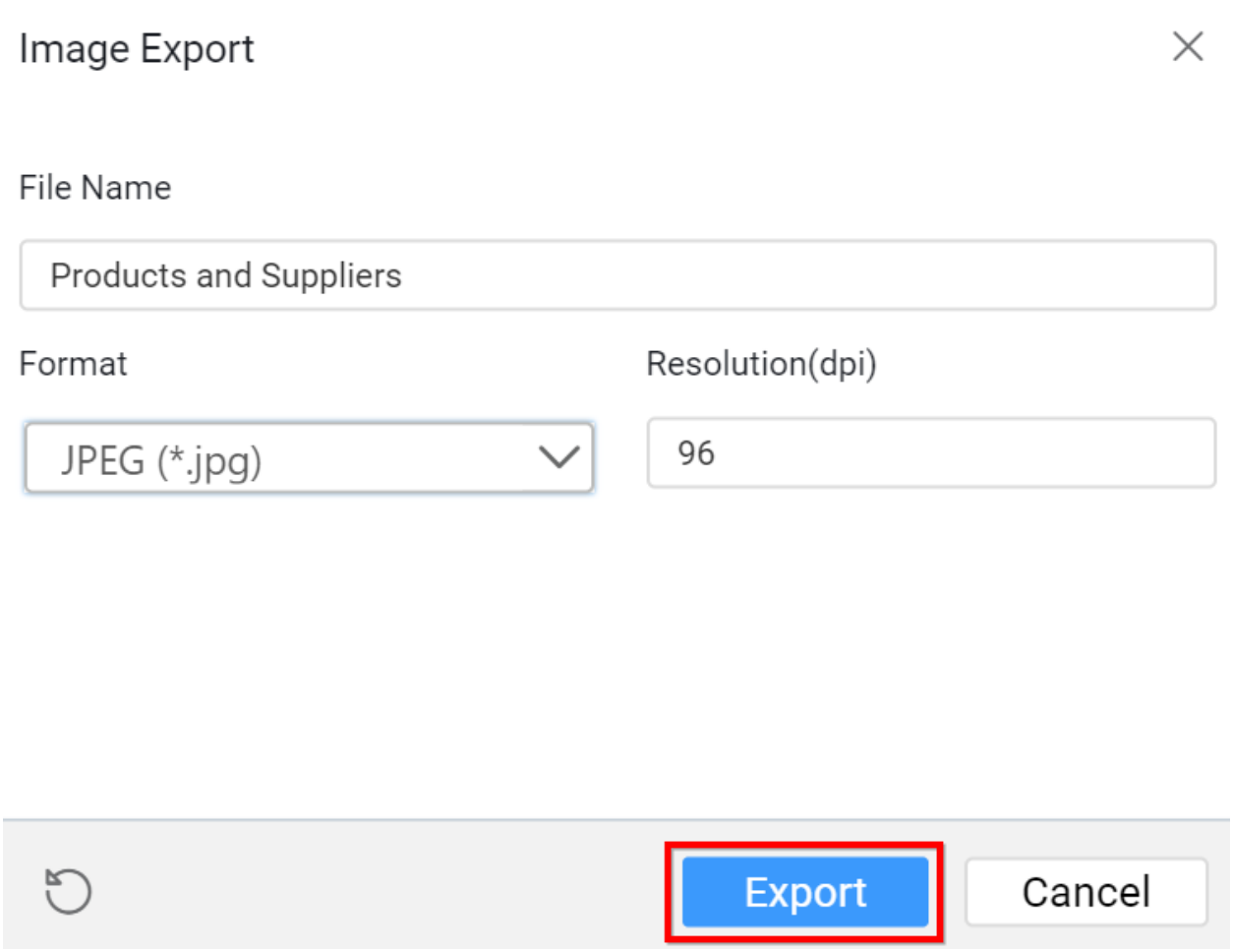


The default and minimum value for Resolution is 96 dpi (dots per inch). Maximum value allowed to set is 1790 dpi.

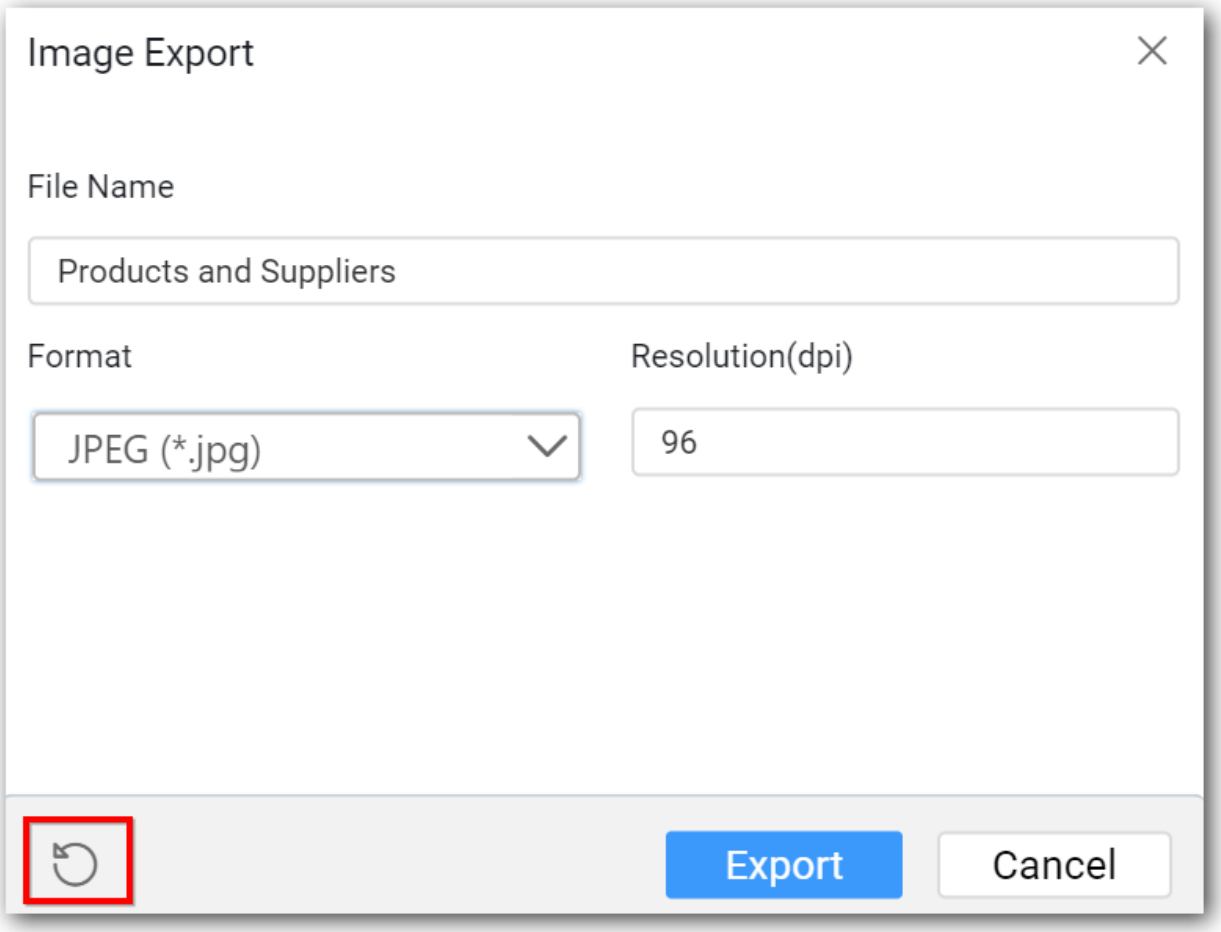


You can choose the image Format as JPEG (*.jpg), PNG (*.png), or BMP (*.bmp) file format.

Click the Export button, the current view of the dashboard will display in the chosen image format with applied settings.



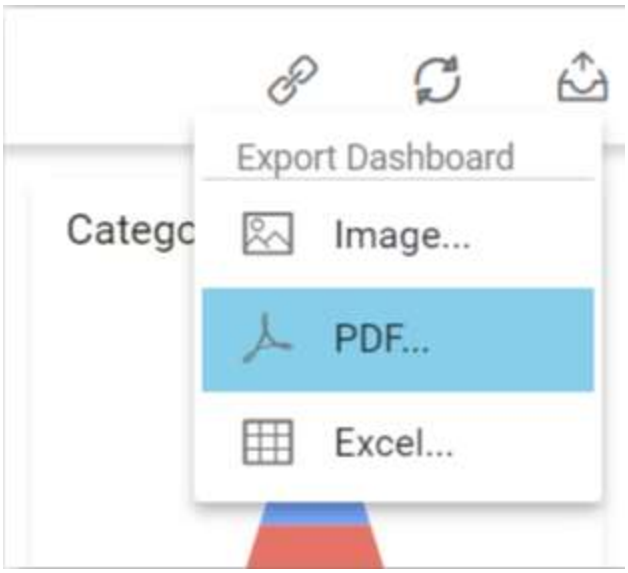
Click the Reset button, the default values get restored in the pop-up.



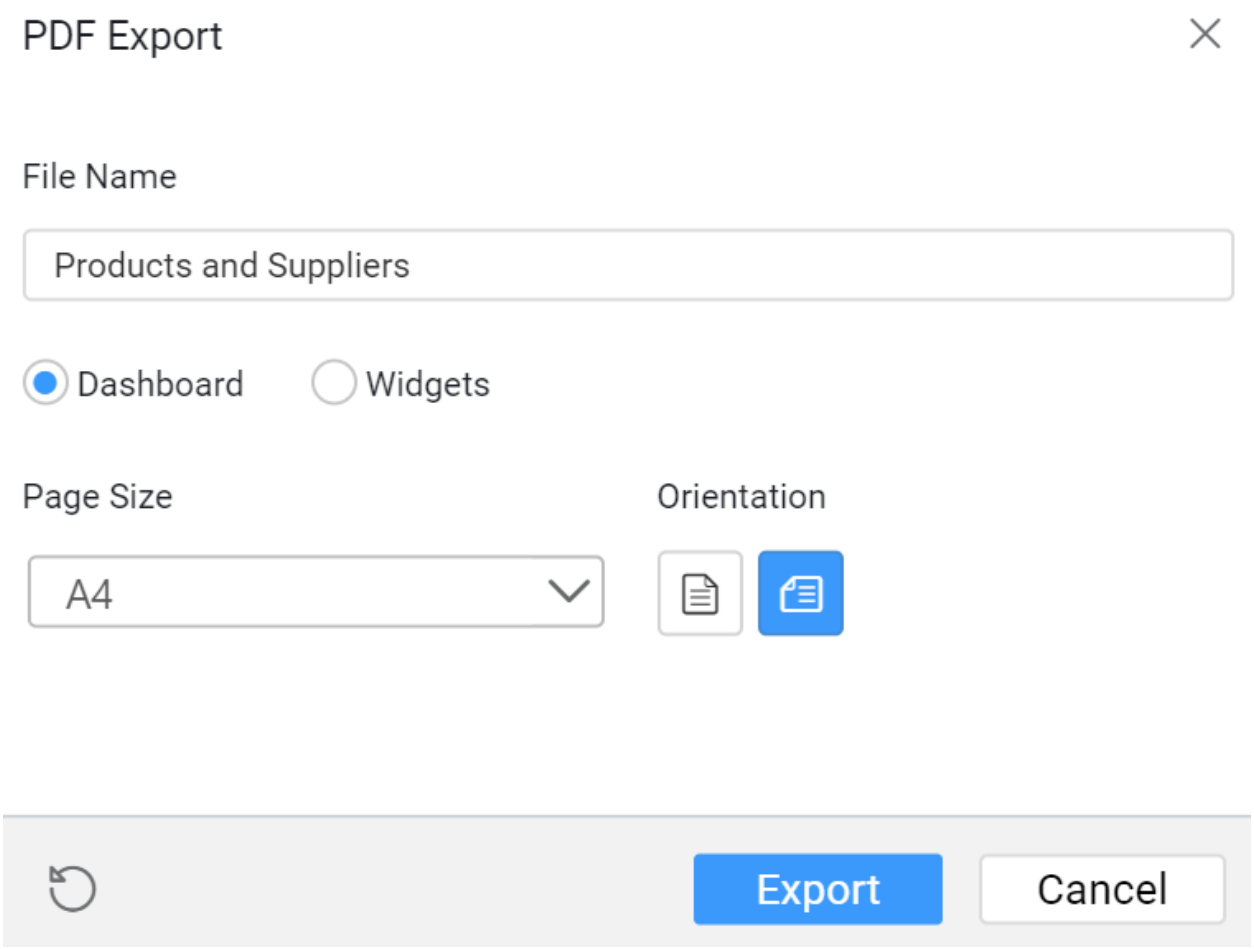
Note: In Windows Operating System, Safari browser does not support the Export to Image functionality. Whereas the Mac OS or iOS that have Safari browser version 6+ support the Export to Image functionality.

Exporting Dashboard to PDF

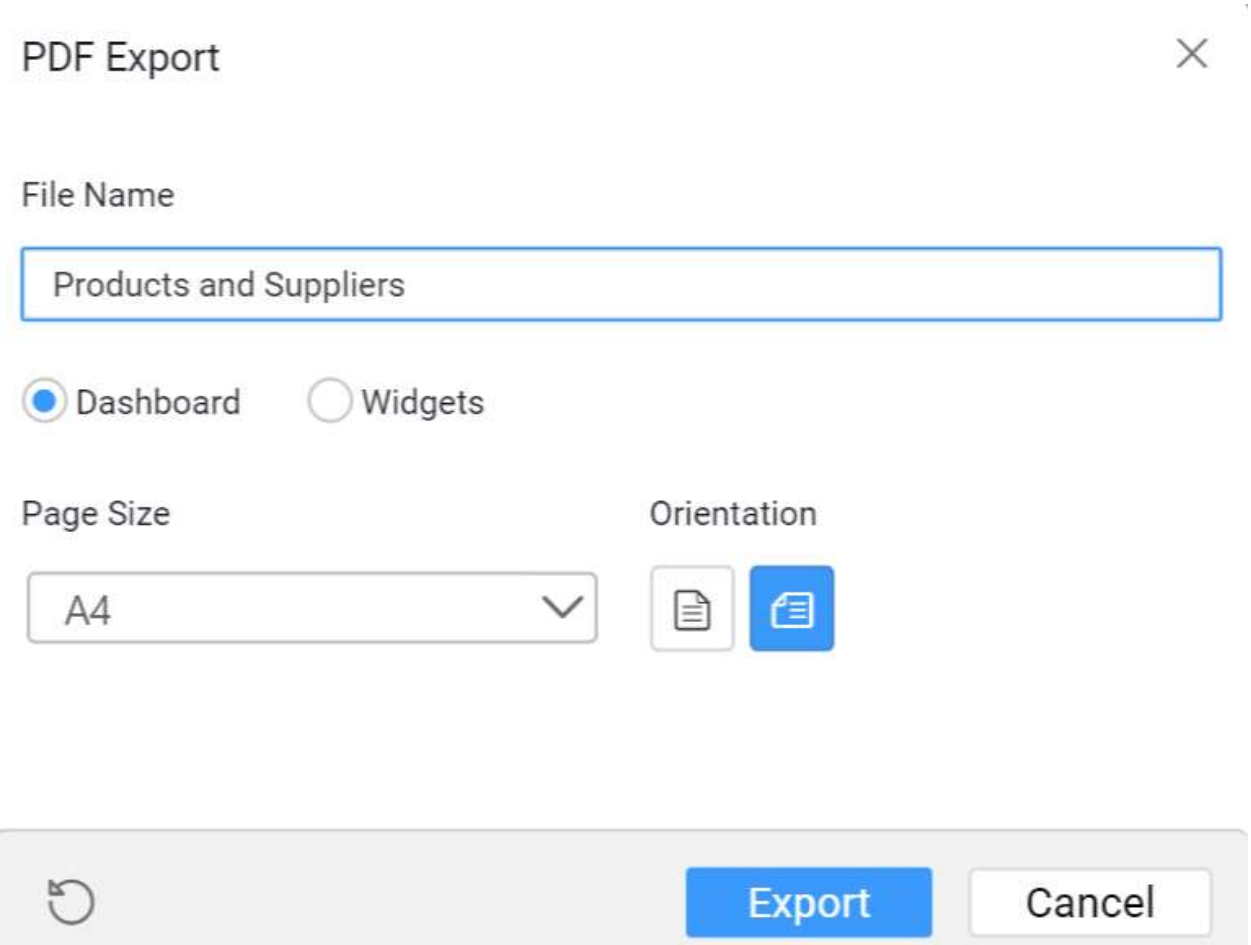
You can obtain the data showcased in the dashboard through exporting it to PDF format by clicking the Export icon at right corner of the title section.



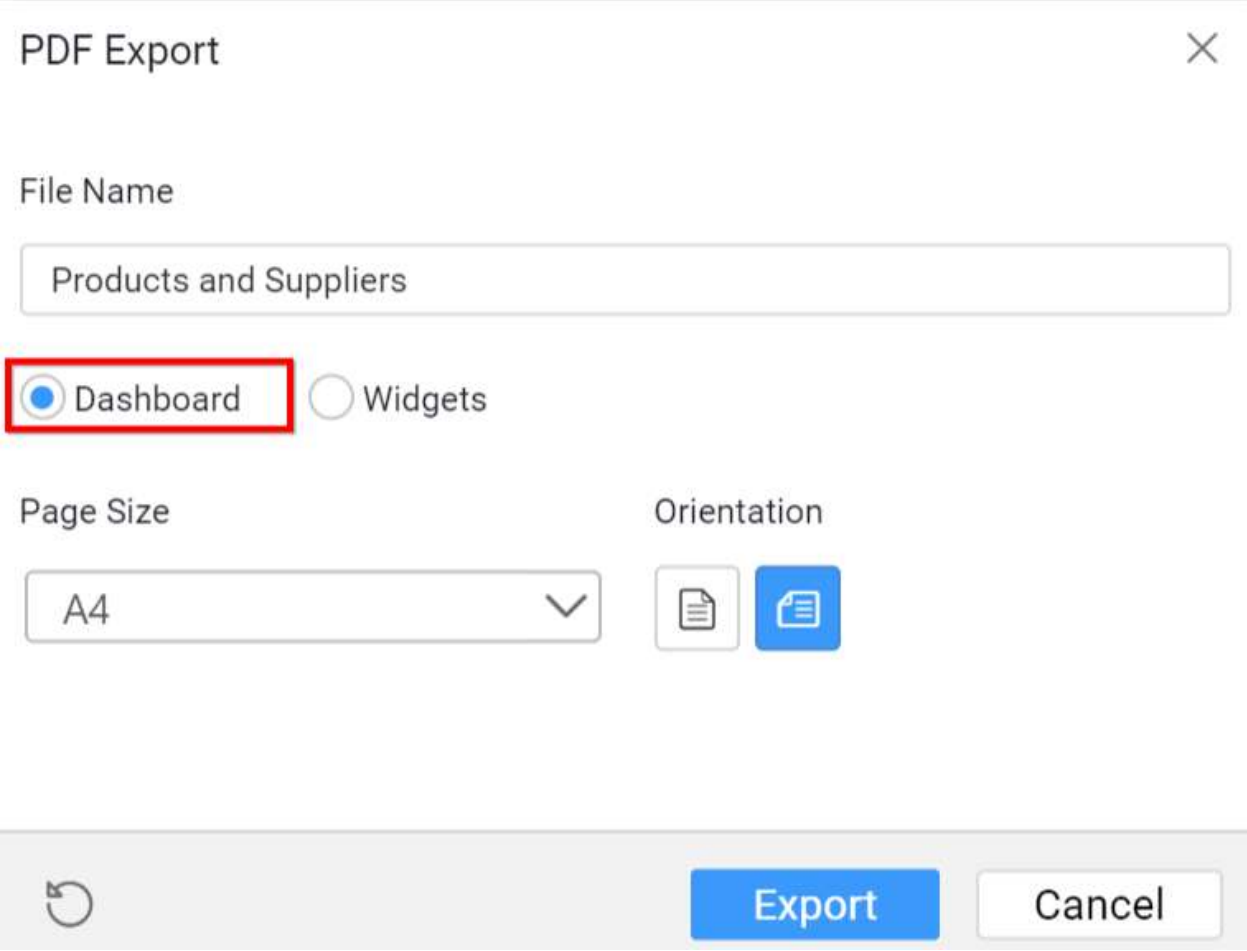
Click the PDF option, the pop-up will be shown as follows.



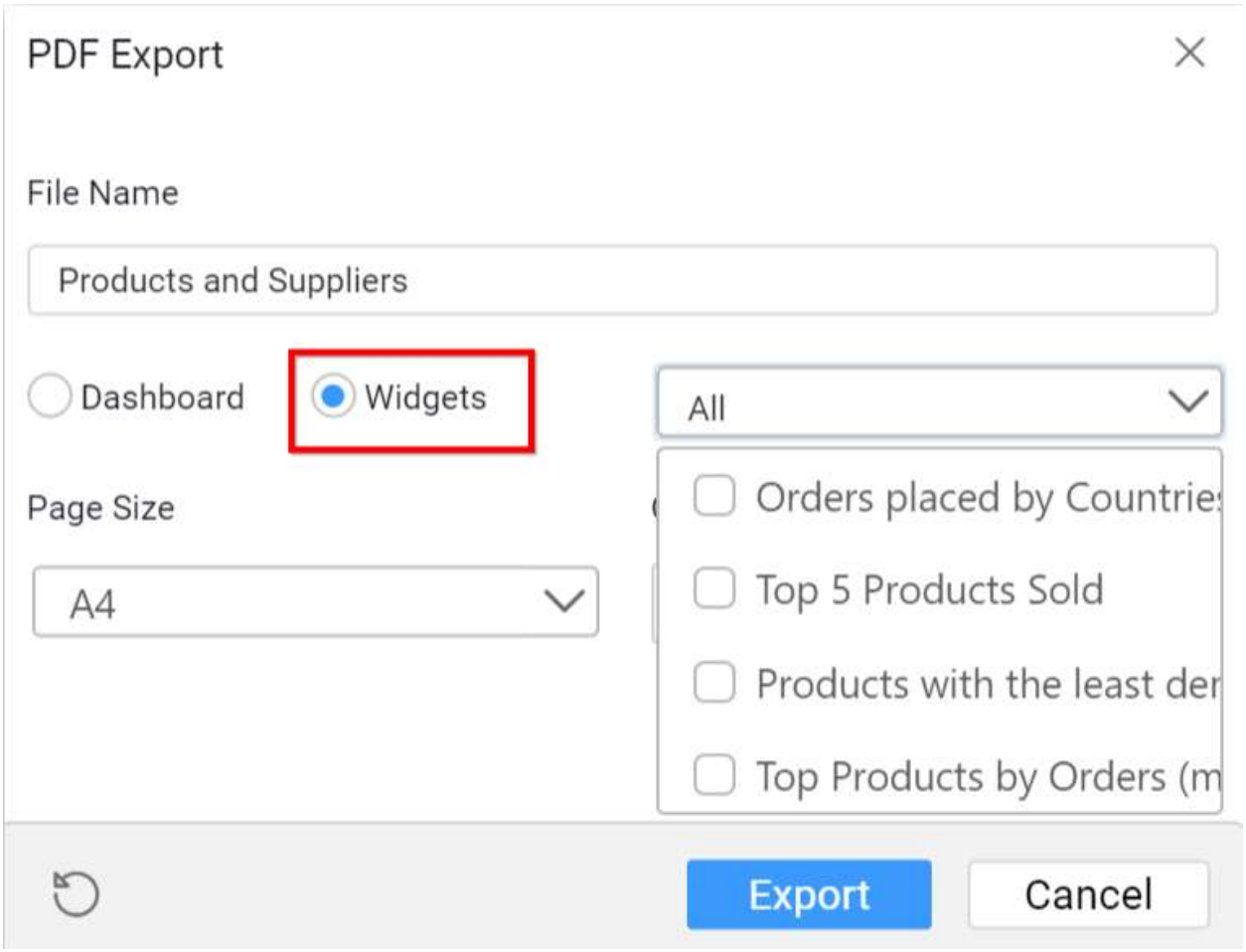
Set the File Name fields with preferred values by replacing the default ones.



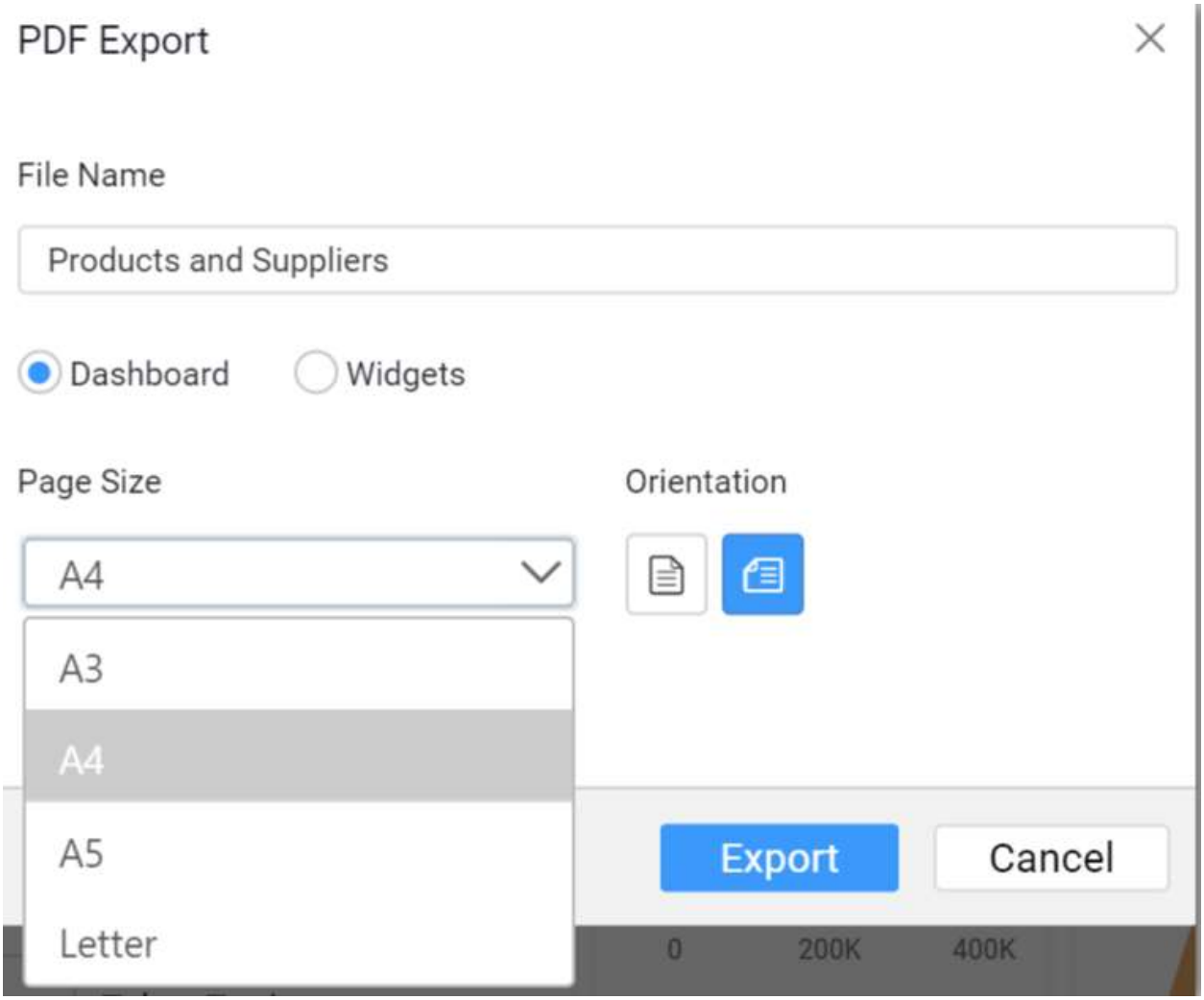
Set the Dashboard radio button, the full dashboard has been exported to the PDF file.



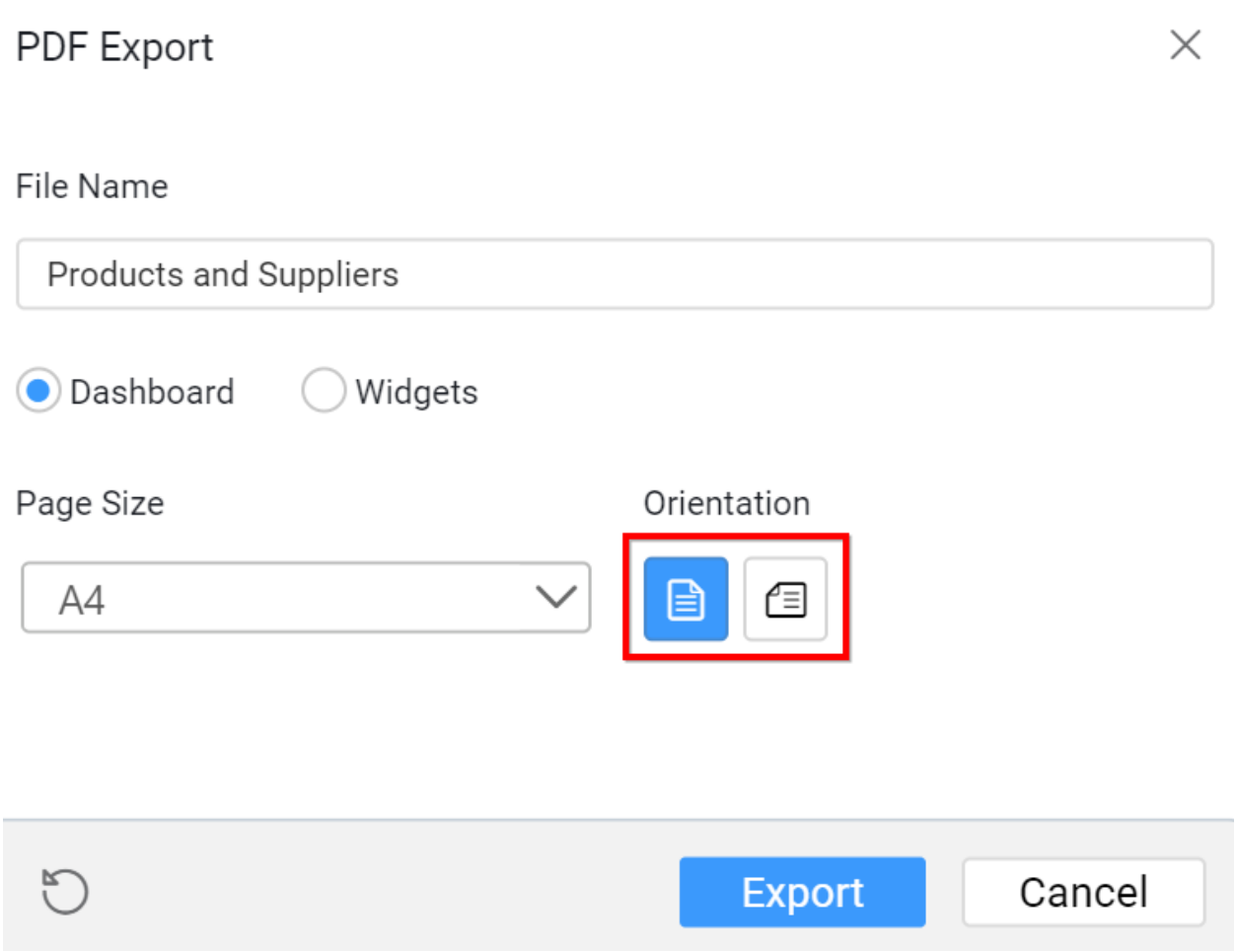
Set the Widgets radio button, each widget except filtering widgets has been exported to an individual page.



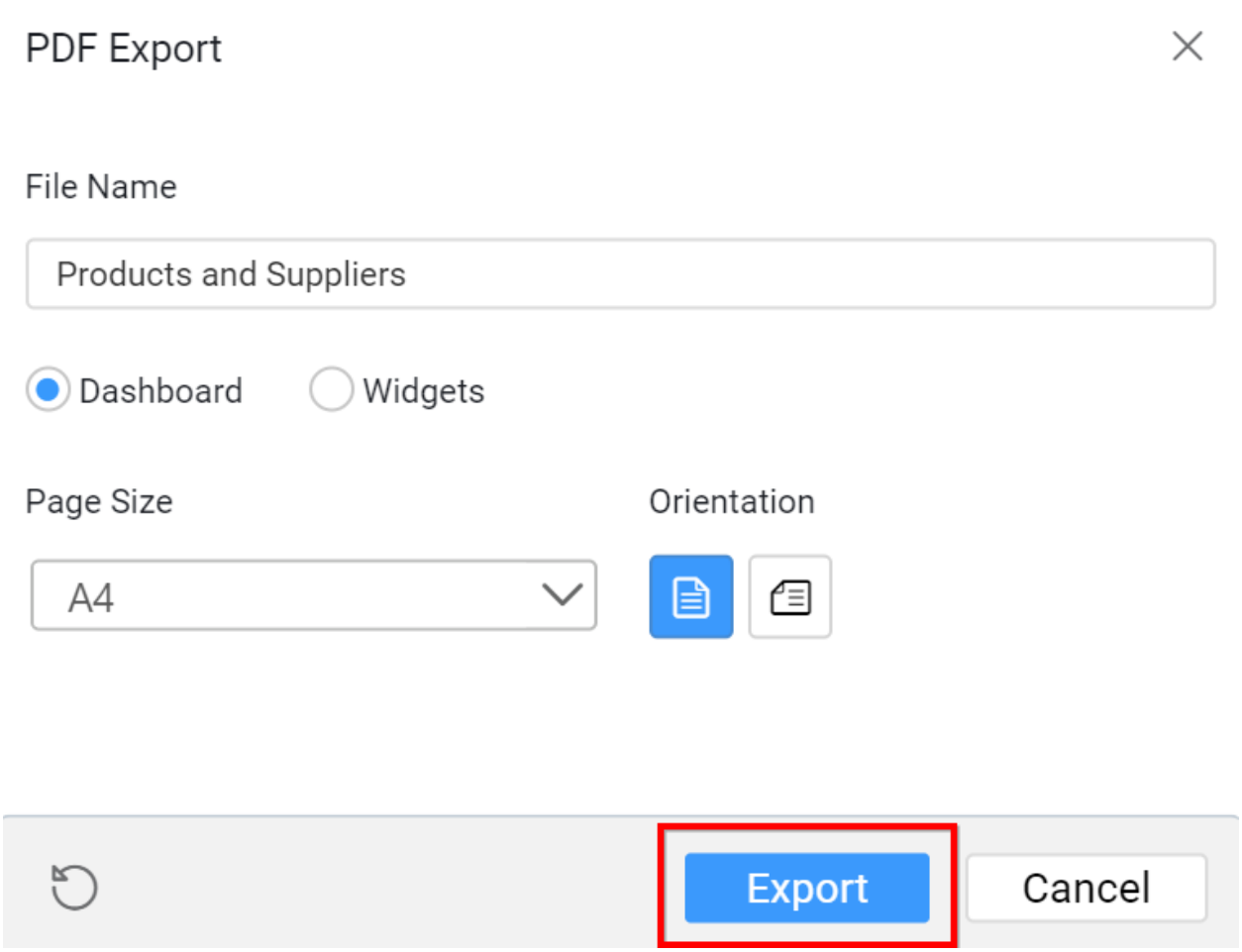
Set the preferred Page Size of the PDF File.



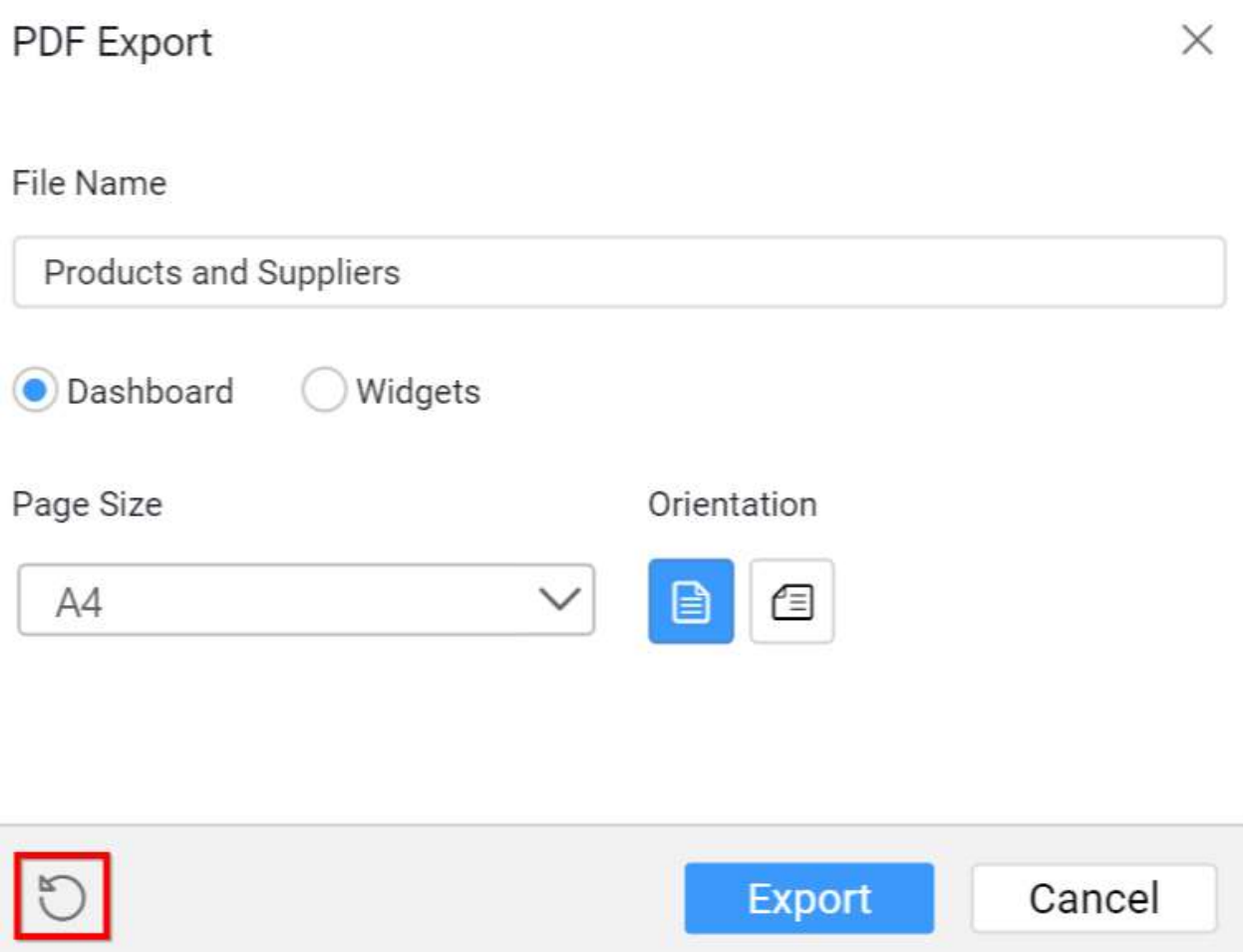
Choose the Orientation of the page as either portrait or landscape mode.



Click the Export button, the data in the dashboard will be displayed in the PDF file format.

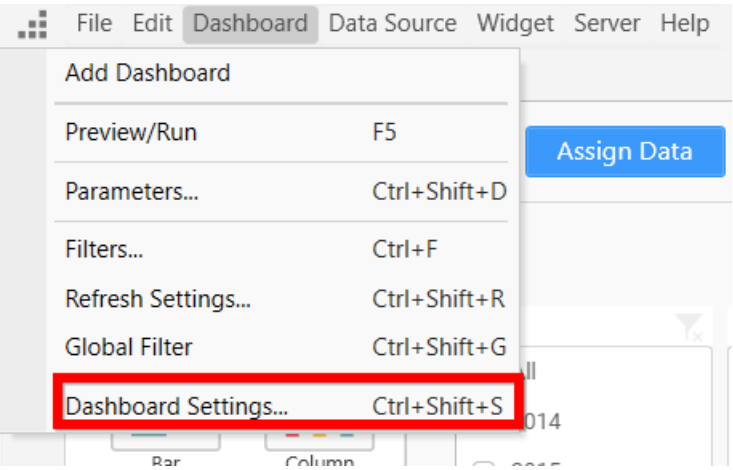


Click the Reset button, the default values get restored in the pop-up.

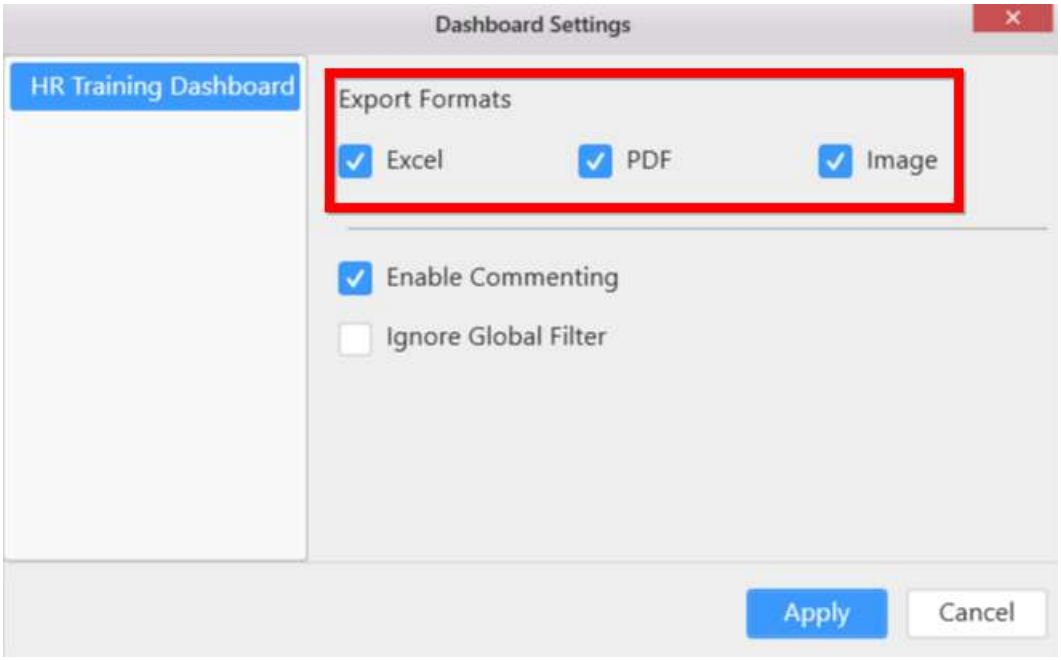


How to Set Dashboard Export Settings from Designer

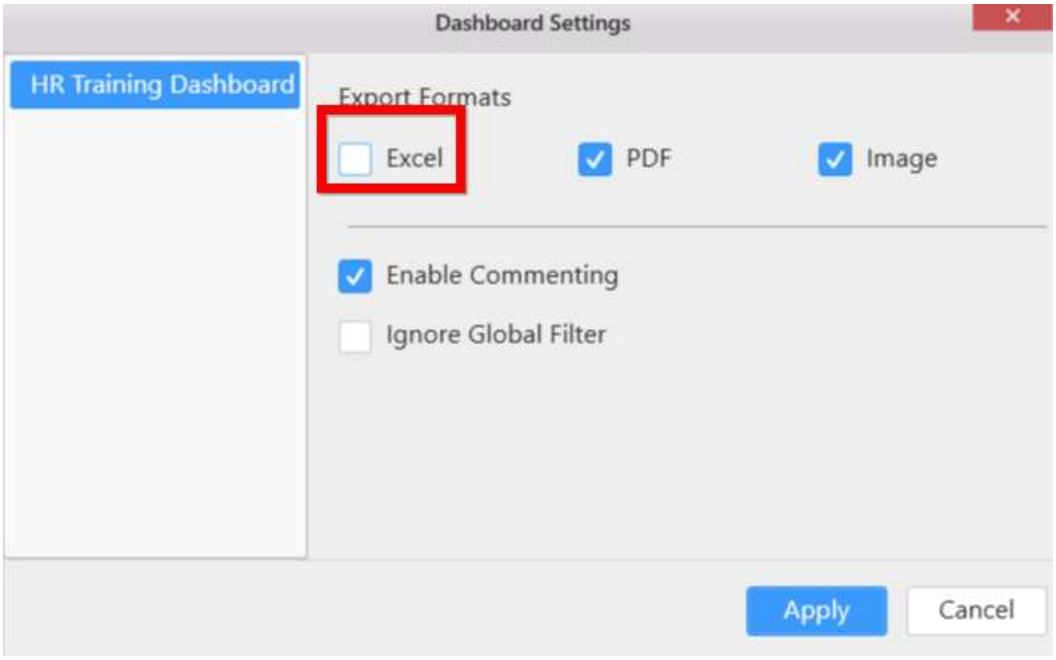
Click Dashboard menu and Select Dashboard Settings option as shown below.



Dashboard Settings window will be opened.



By default, Export Formats options will be enabled, if you want to disable any option you can uncheck the options.



Click Apply button to update the changes in the window.

Now, disabled option will not be shown while previewing the dashboard.

Troubleshooting

PhantomJS exports the widgets/dashboards to an image/PDF. PhantomJS (phantomjs.org) is a headless WebKit scriptable with JavaScript and the latest stable release is version 2.1. If the PhantomJS is not downloaded while installing the dashboard setup, you can do it manually to enable the exporting operation. The following steps illustrate how to install the PhantomJS.exe to the dashboard service.

1. Download the PhantomJS from the following link.

https://bitbucket.org/ariya/phantomjs/downloads/phantomjs-2.1.1-windows.zip

2. Extract the PhantomJS-2.1.1-windows.zip and find the PhantomJS.exe file in the bin folder. Copy the PhantomJS.exe file and paste inside the dashboard service folder in the following directories:

Dashboard Designer : C:\ProgramData\Syncfusion\DashboardDesigner\{Version}\IISExpress_DashboardService.

Dashboard Platform SDK : %localappdata%\Syncfusion\Dashboard Platform SDK\Service.

Dashboard Server:

Version 3.2 and higher	{Dashboard Server Installed Location}\Dashboard Server\DashboardServer.web\API
Version 3.1 and lower	{Dashboard Server Installed Location}\DashboardService

3. For the Dashboard Azure Package, download the PhantomJS from the link and extract PhantomJS-2.0.0-windows.zip, and then find the PhantomJS.exe file in the bin folder.
4. Copy the PhantomJS.exe file and paste it in the following location.

Version 3.2 and higher	/site/wwwroot/ds/Dashboard Server/DashboardServer.Web/API
Version 3.1 and lower	/site/wwwroot/ds/DashboardService

5. Now, the exporting operation can be performed for dashboards.

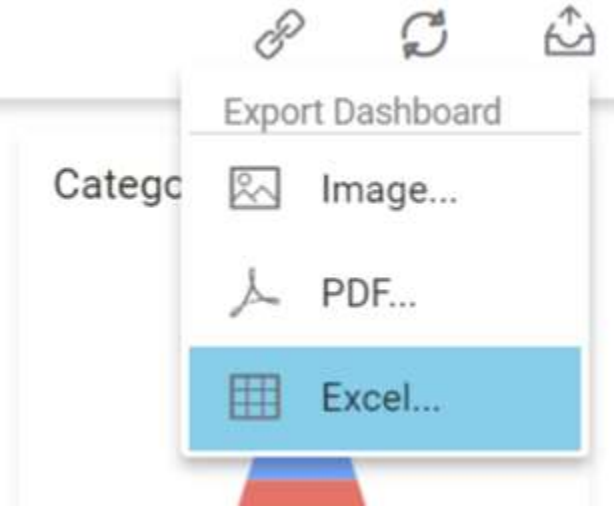
Limitation in PDF export

When the grid widget is exported to PDF, the maximum rows of data exported to PDF will be 1000. The time taken for exporting the data to PDF depends on the number of columns added in the grid widget.

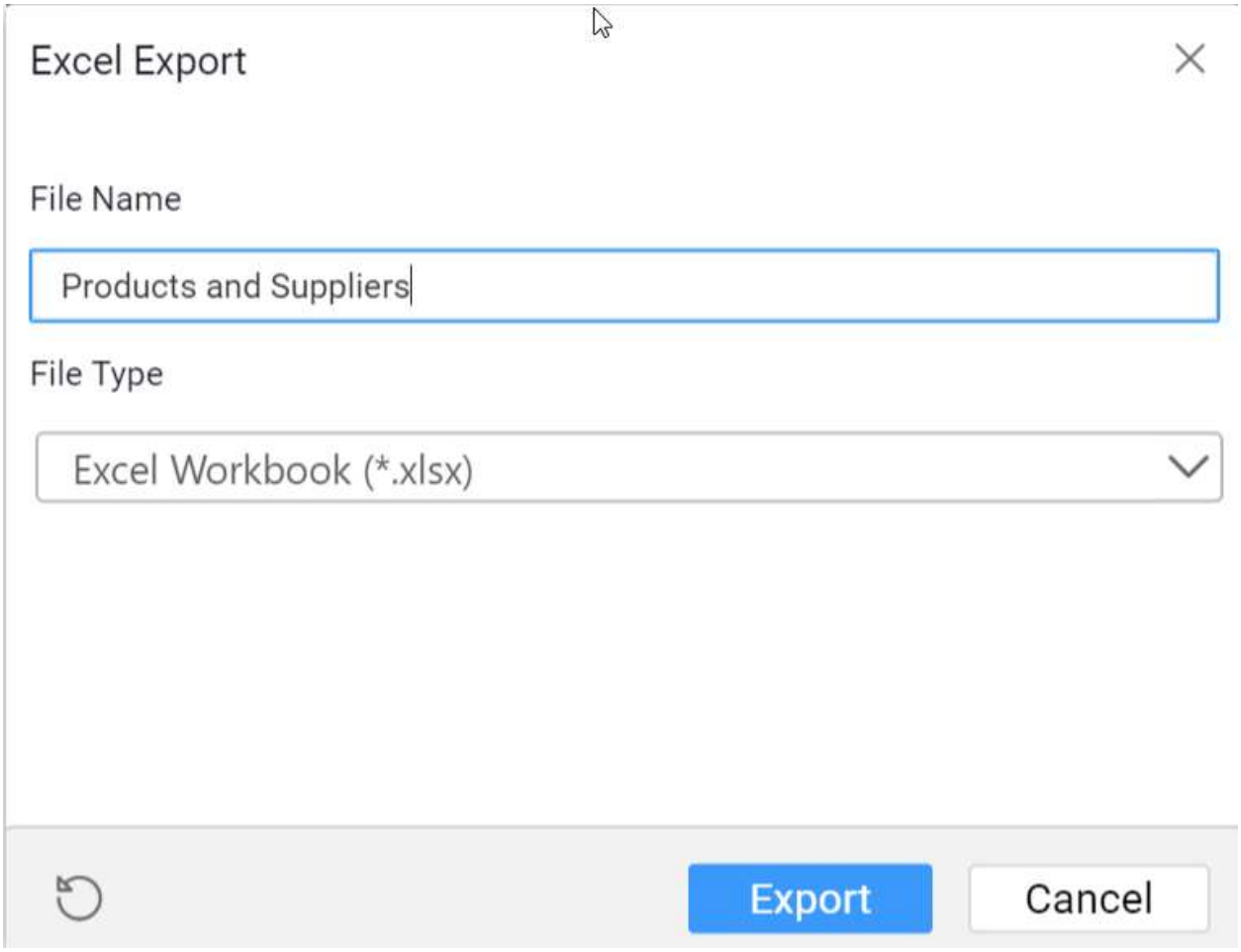
To avoid the performance issue, the maximum rows exported to PDF is limited to 1000. To export more than thousand rows of data, use the excel export.

Exporting Dashboard to Excel

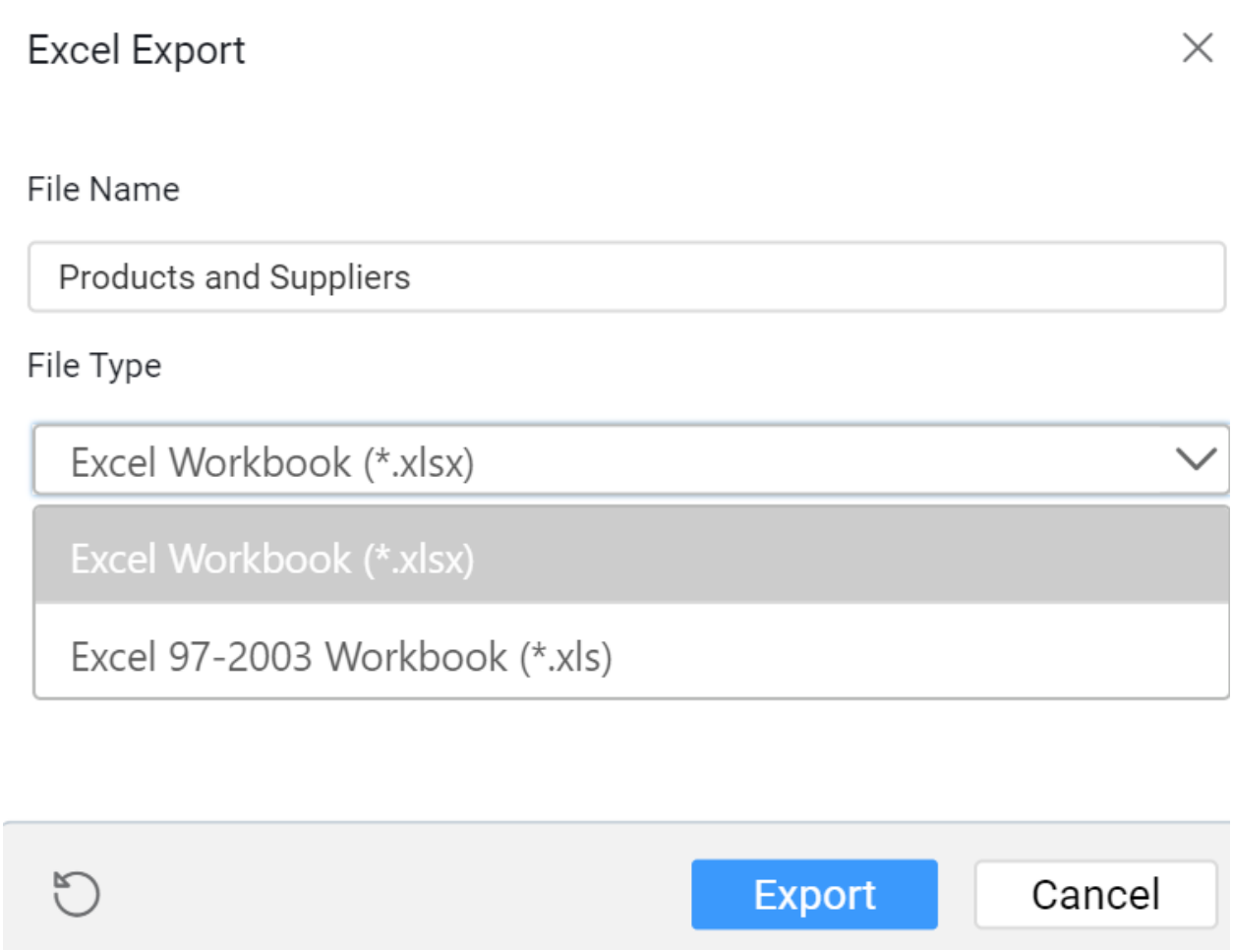
You can obtain the aggregated data showcased in the dashboard by exporting it to Excel format using the Export icon at right corner of the title section.



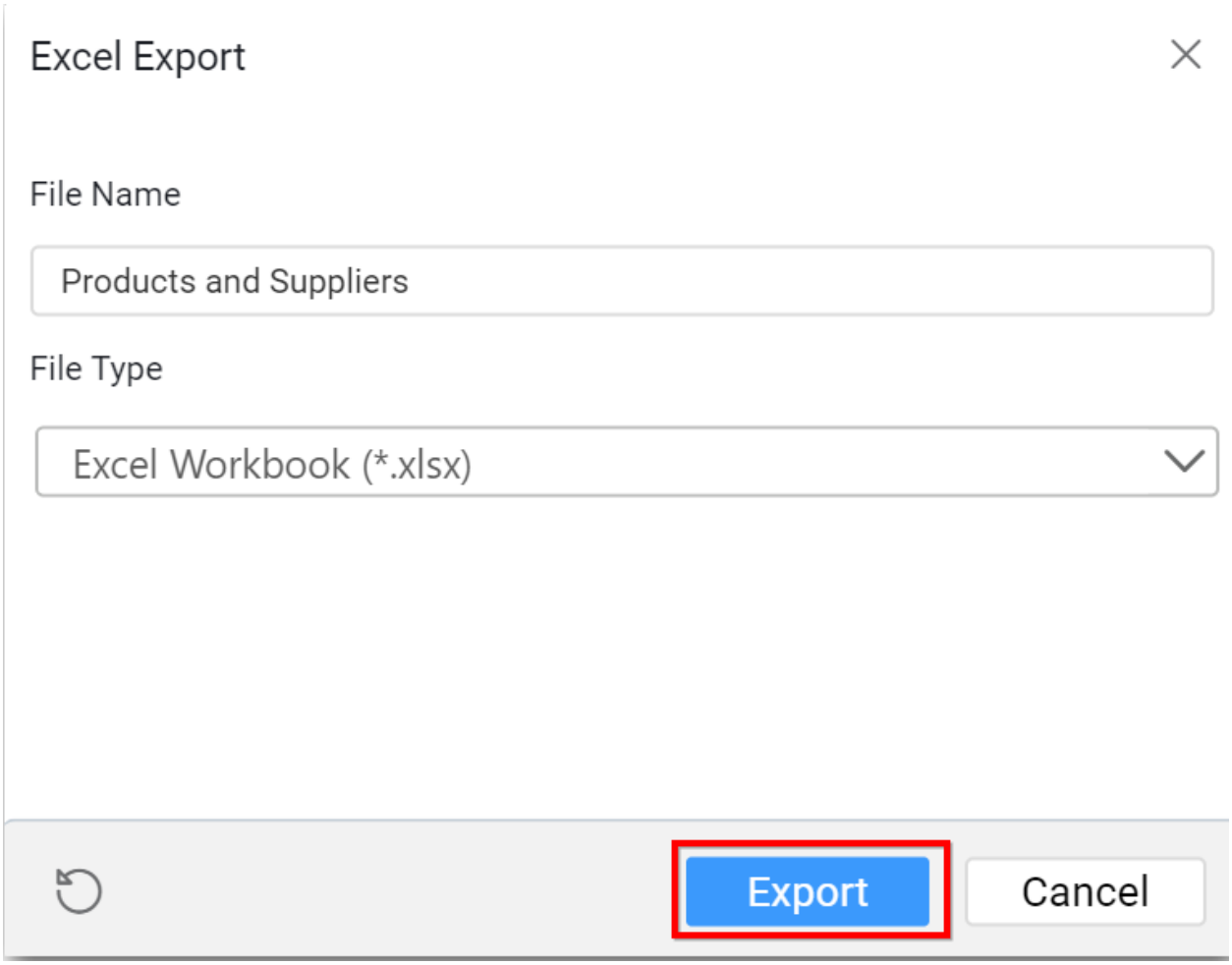
Click the Excel option, the pop-up will be shown as follows.



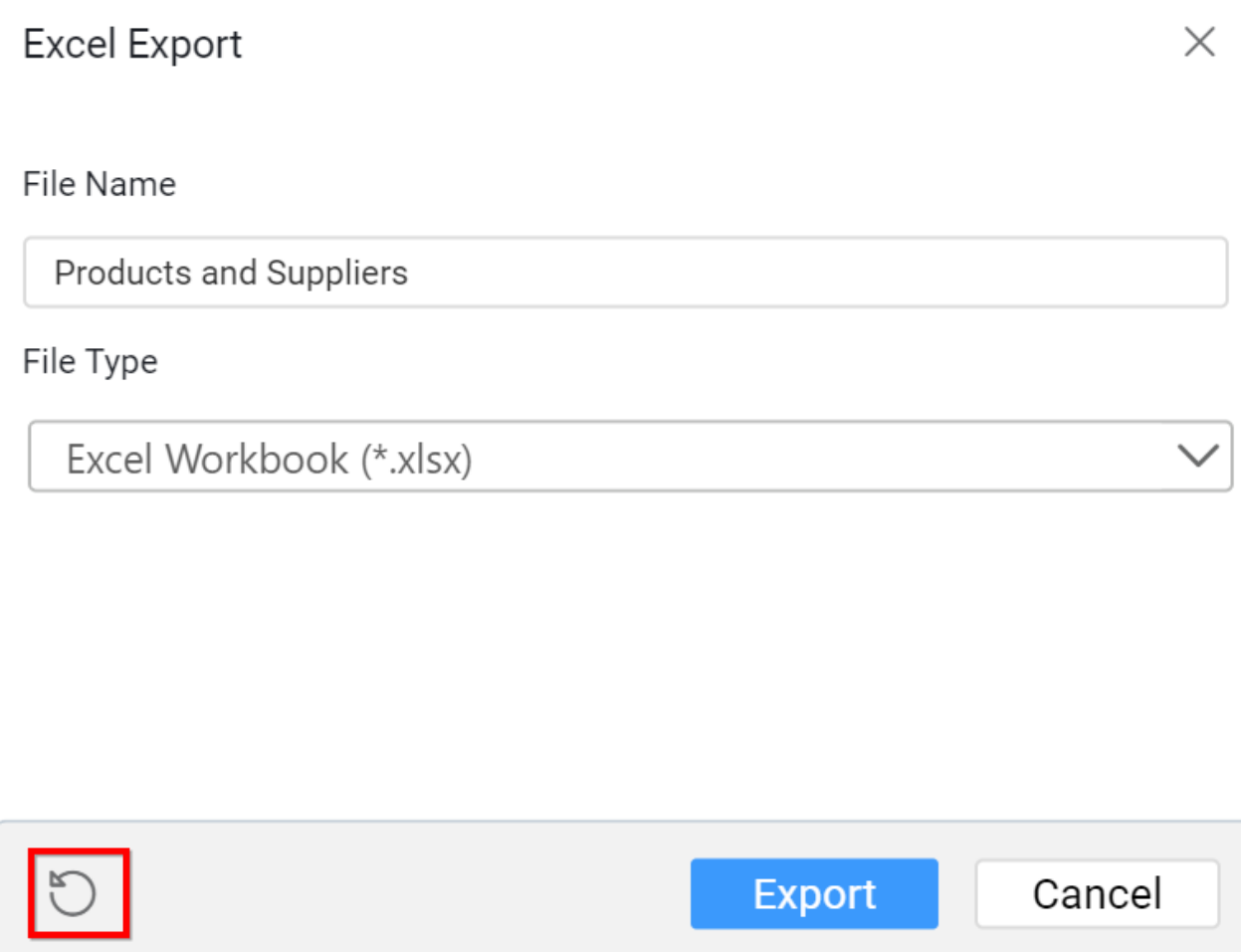
Set the File Name field with preferred value by replacing the default one.



Choose the Format as either Excel Workbook (*.xlsx) or Excel 97-2003 Workbook (*.xls).



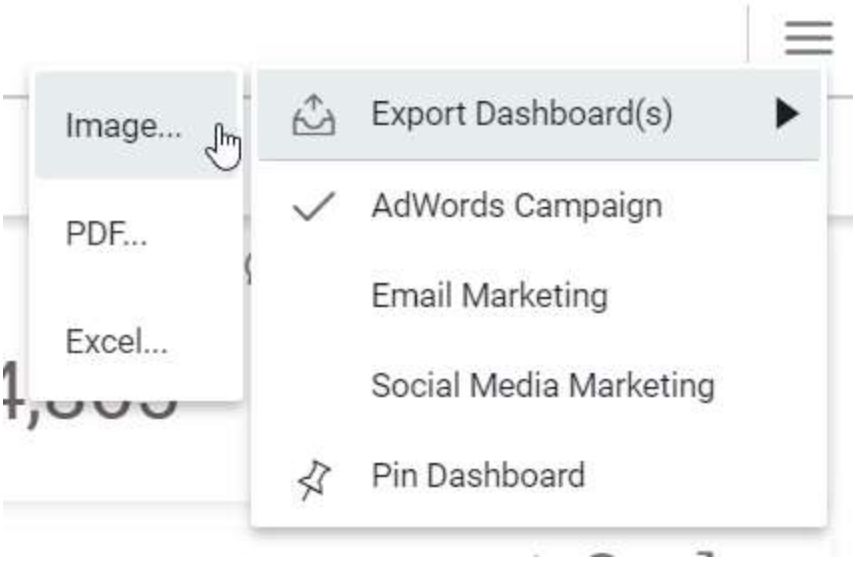
Click the Export button, the data in the dashboard will be displayed in the chosen Excel format.



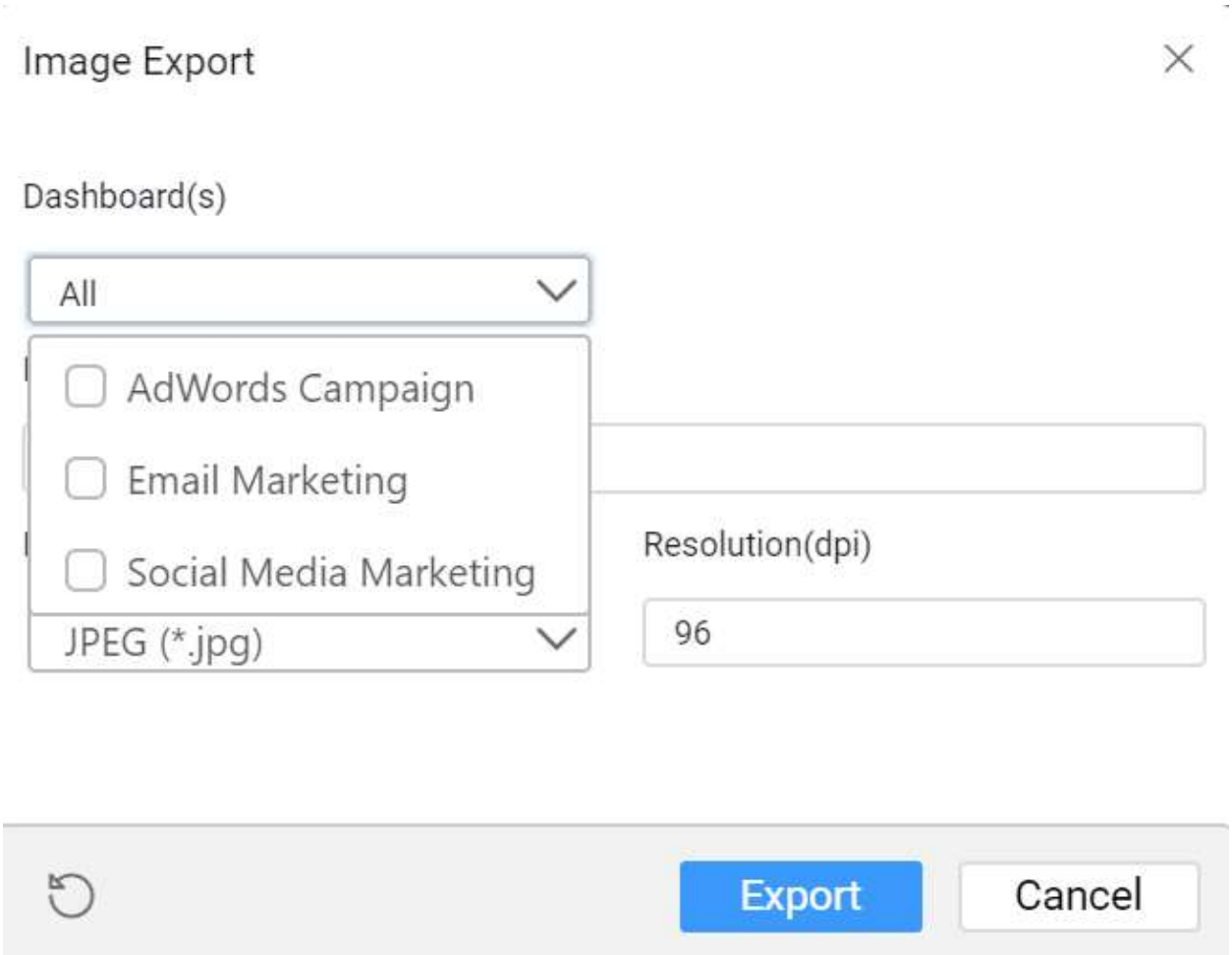
Click the Reset button, the default values get restored in the pop-up.

Exporting multi-tabbed Dashboard to image

Export the current view of the multi-tabbed dashboard in the form of image by clicking the Image in the Exports Dashboard option in the drop-down menu at right corner of the tab title section.



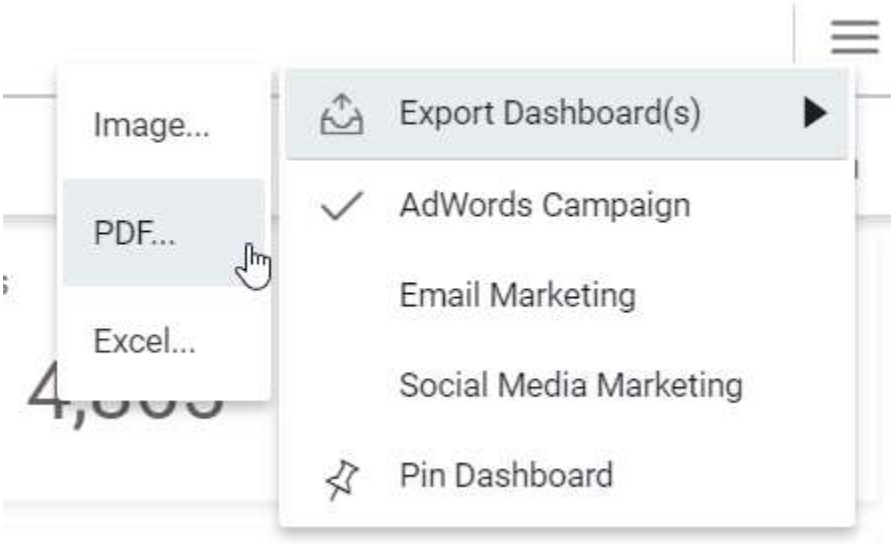
Clicking the Image option, the pop-up will be shown as follows.



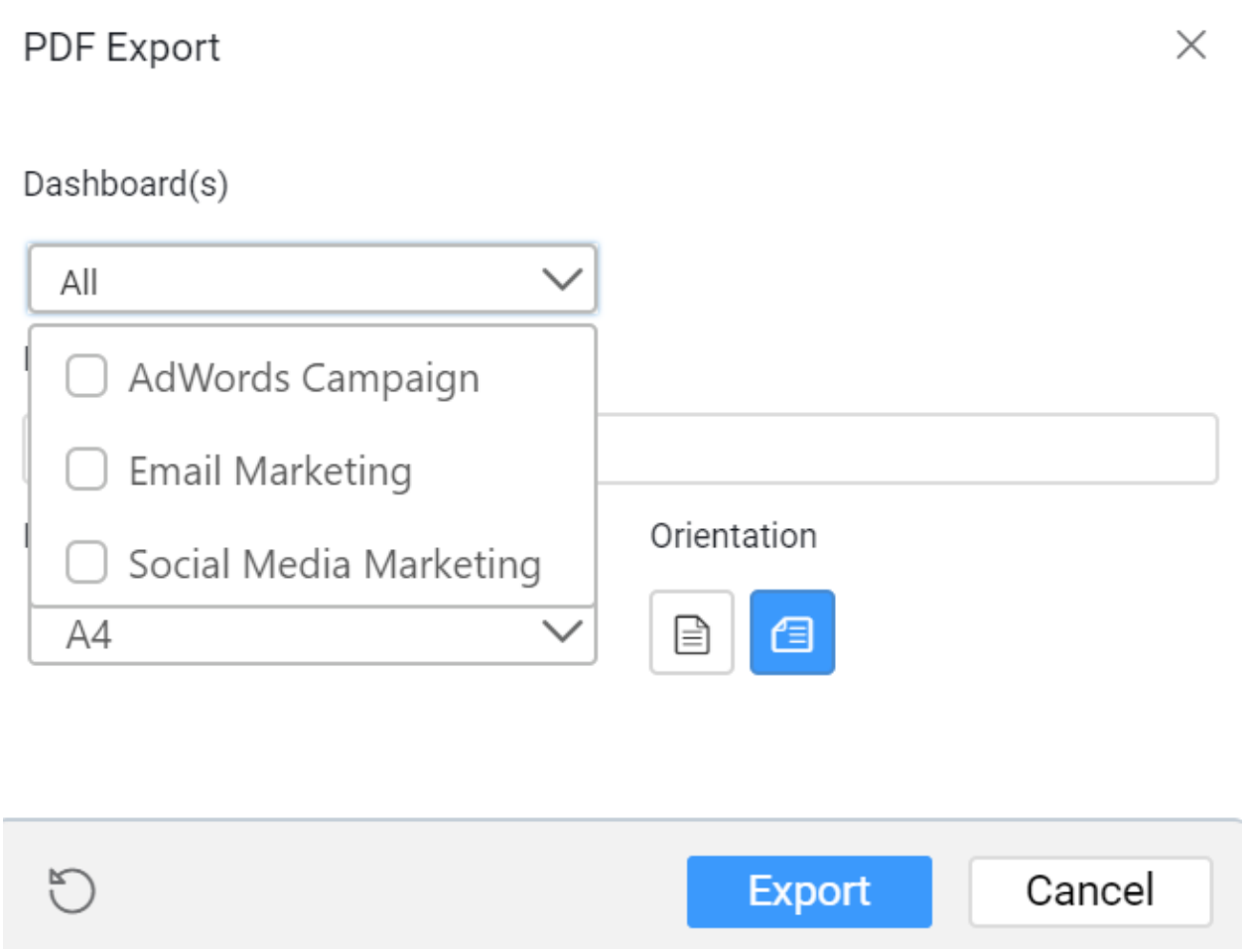
Note: In Windows Operating System, the Safari browser does not support the Dashboard Export to Image functionality. Whereas the Mac OS or iOS that have Safari browser version 6+ support the Dashboard Export to Image functionality.

Exporting multi-tabbed Dashboard to PDF

You can obtain the data showcased in the dashboard through exporting it to PDF format by clicking the PDF in the Exports Dashboard option in the drop-down menu at right corner of the title section.

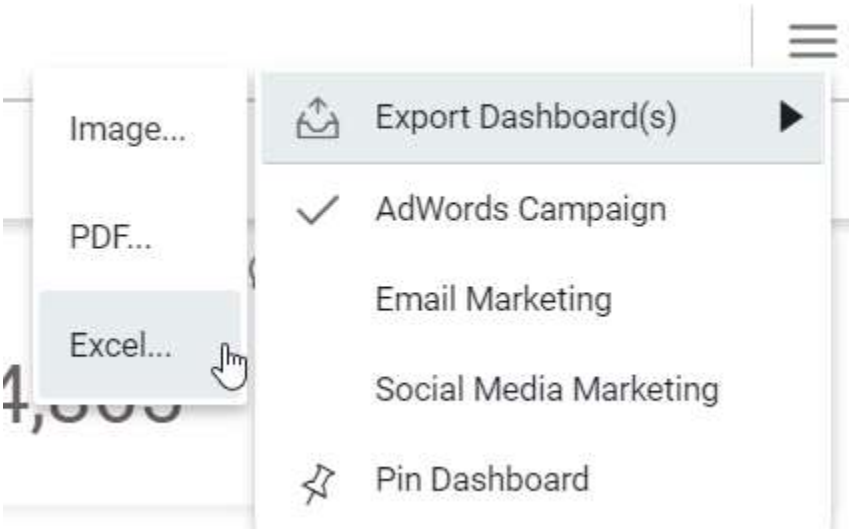


Click the PDF option, the pop-up will be shown as follows.

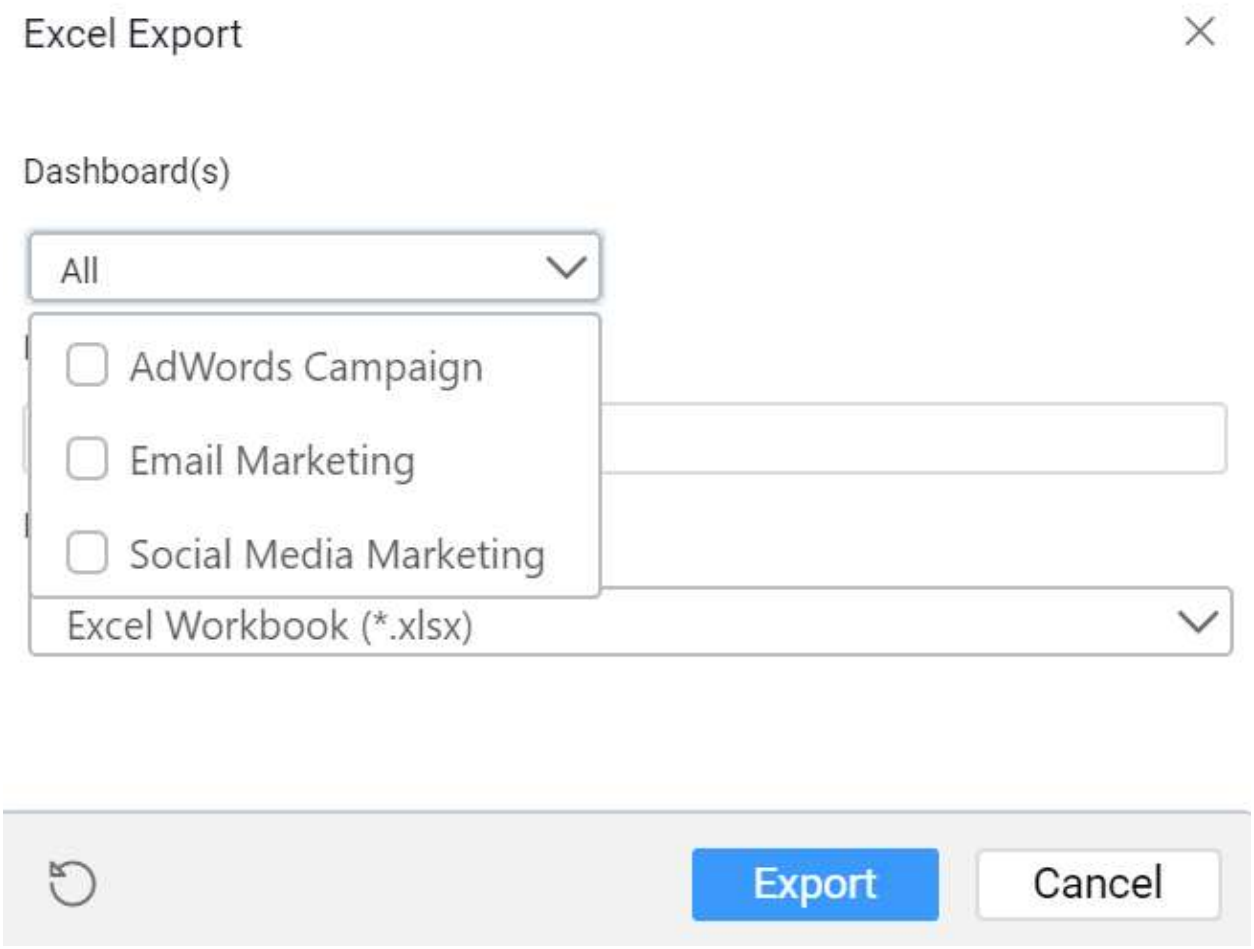


Exporting multi-tabbed Dashboard to Excel

You can obtain the aggregated data showcased in the dashboard through exporting it to Excel format by clicking the Excel in the Exports Dashboard option in the drop-down menu at right corner of the title section.

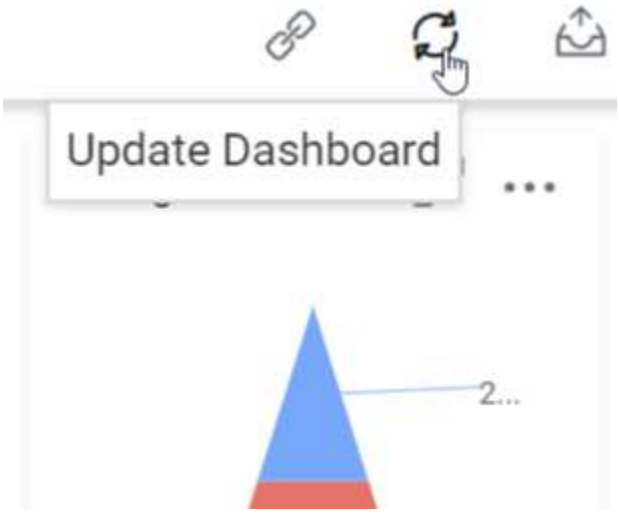


Click the Excel option, the pop-up will be shown as follows.



Updating Dashboard

You can update the dashboard manually by clicking the Update Dashboard icon at right corner of the title section.



This will update the dashboard with the browser cached data view which will remain for 10 minutes. After this time, the updated data get fetched from the data server and store in the cache.

Troubleshooting

PhantomJS WebKit is used to support the image and PDF export operations in widget, dashboard, and schedule. Without this, the image and PDF export options in dashboard, widgets, and schedules no longer available. PhantomJS (phantomjs.org) is a headless WebKit scriptable with JavaScript and the latest stable version is 2.1. If this WebKit is not downloaded and installed during the installation process of dashboard installer, you can do it manually.

Following steps illustrate how to download and extract the PhantomJS.exe and add to the dashboard service.

1. Download the PhantomJS WebKit from here. 2. Extract the PhantomJS-2.2.1-windows.zip and find the PhantomJS.exe file in the bin folder. 3. Copy the PhantomJS.exe file and paste it into the dashboard service directory in respective installers.

a. Dashboard Designer – C:\ProgramData\Syncfusion\DashboardDesigner\{Version}\IISExpress_DashboardService.

b. Dashboard Server – {Dashboard Server Installed Location}\Dashboard Service.

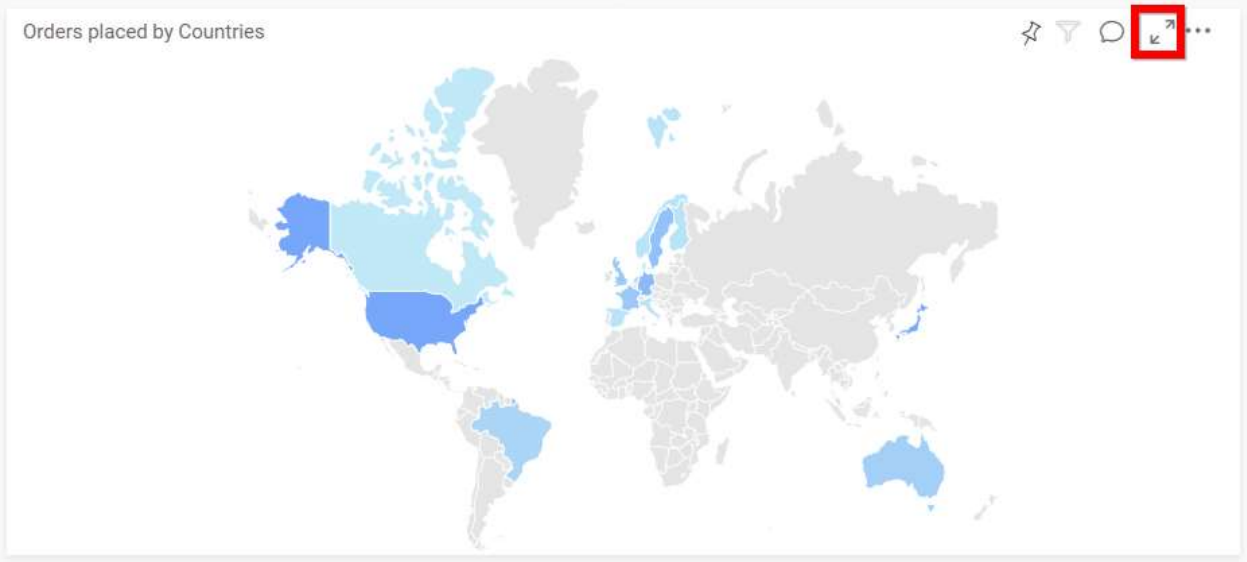
c. Dashboard Platform SDK - %localappdata%\Syncfusion\Dashboard Platform SDK\Service

Now, the exporting operation can be performed in the dashboard, widget, and schedule.

Widget Settings

Maximizing Widget View

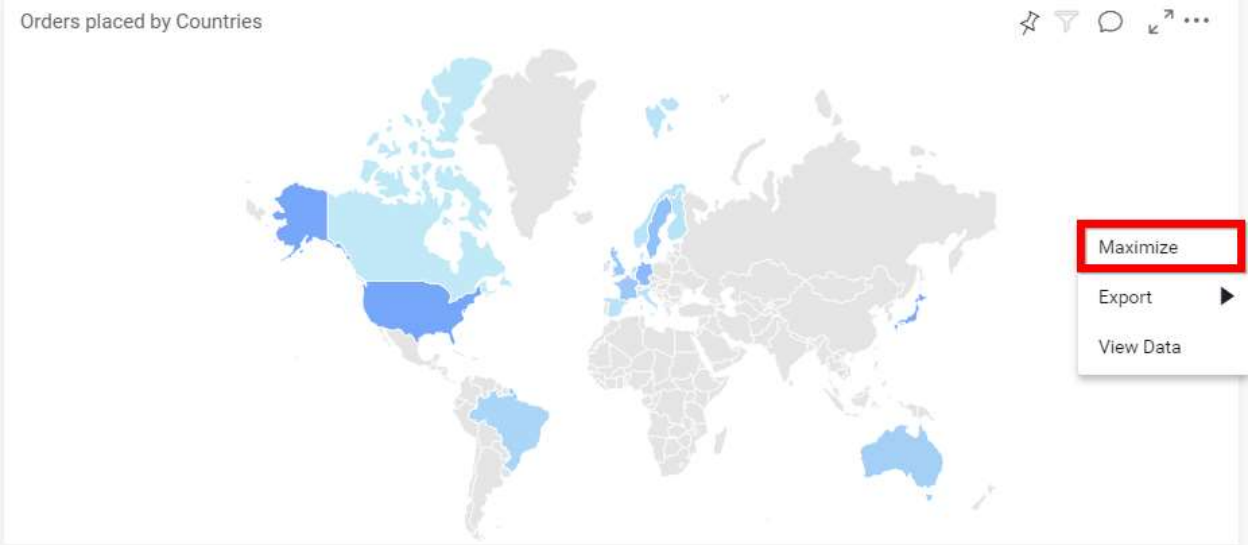
You can get the maximized view of widget by clicking the Maximize icon at right corner of the widget title section.



You can also get the maximized view of widget by following below steps.

Right click on the widget that you want to get maximized view.

Click the Maximize option shown in the list like below.



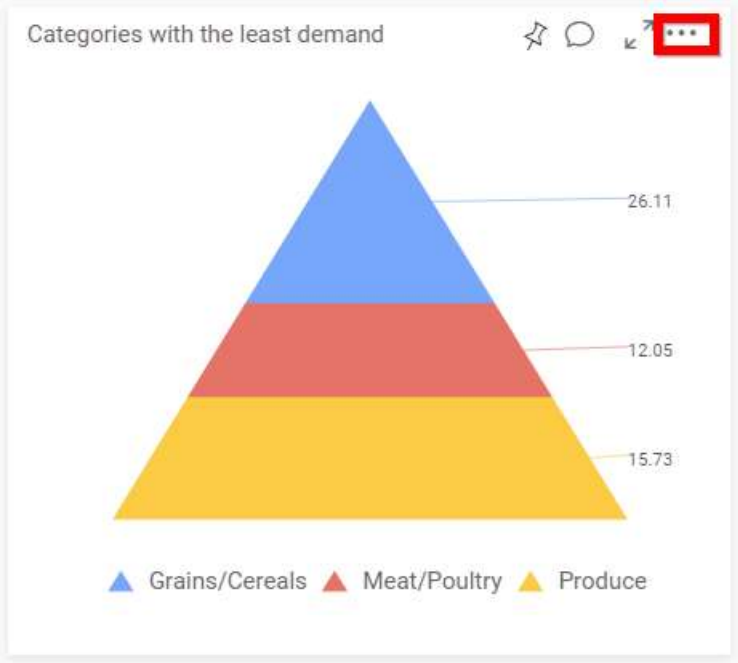
The Maximized view will be like below.

Exporting Widget to Image

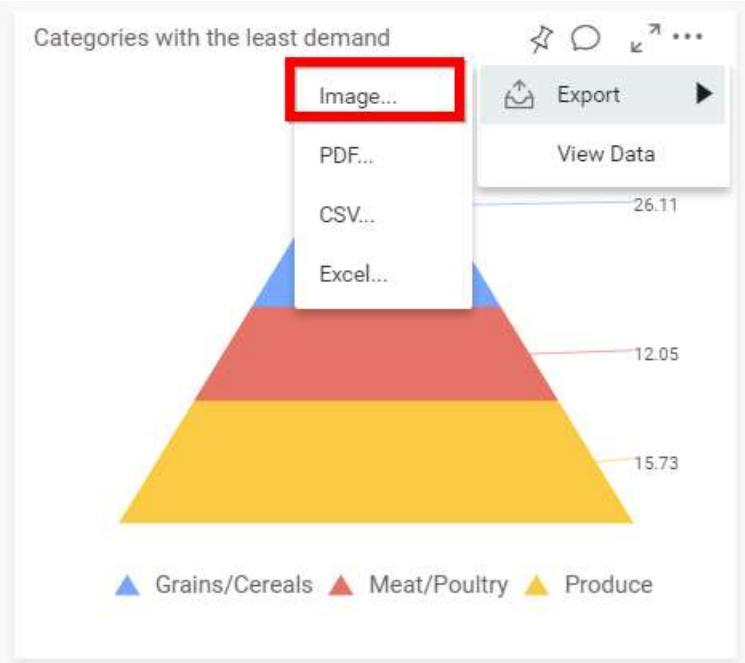
You can export widget as Image in following ways.

By using More Options icon at the right corner of the title section.

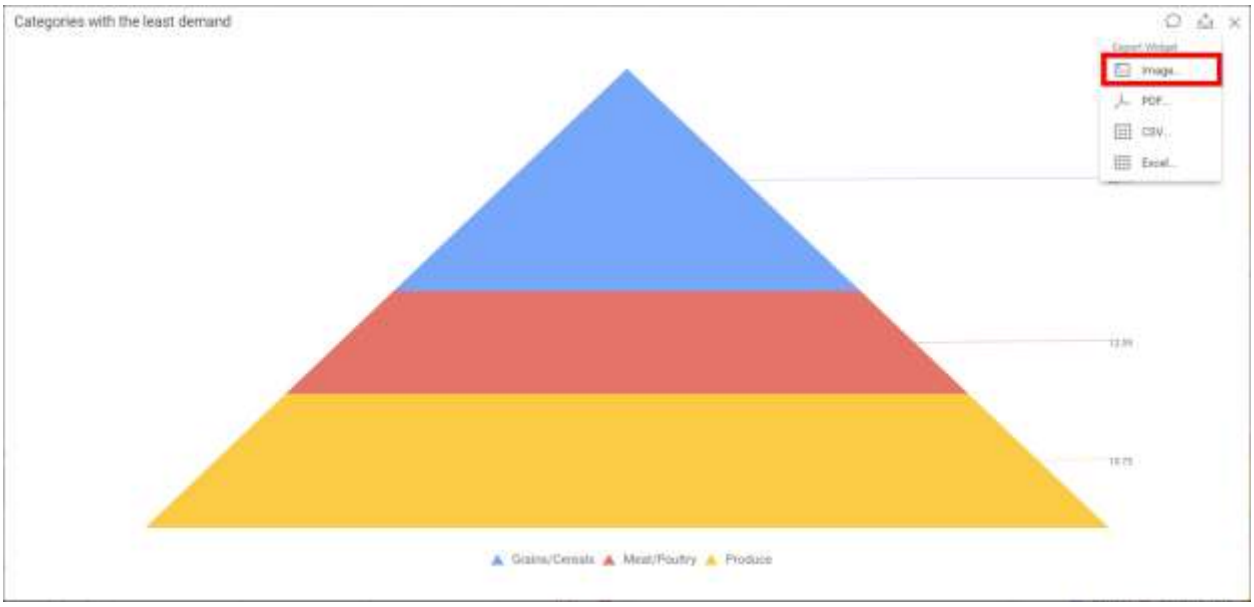
Click More Options icon in the widget at the right corner of the title section.



Select the Export option and choose Image option from list shown.

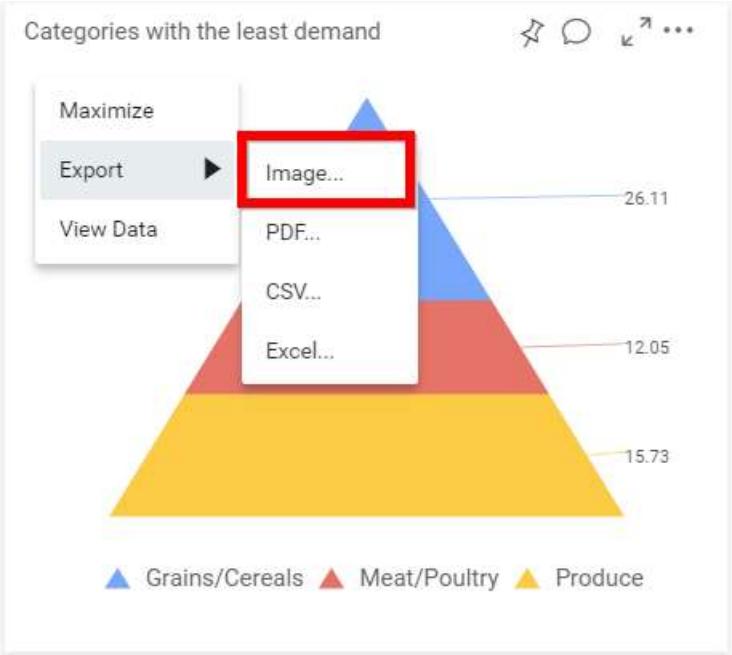


By clicking the Export icon at the right corner of the title section in maximized view.

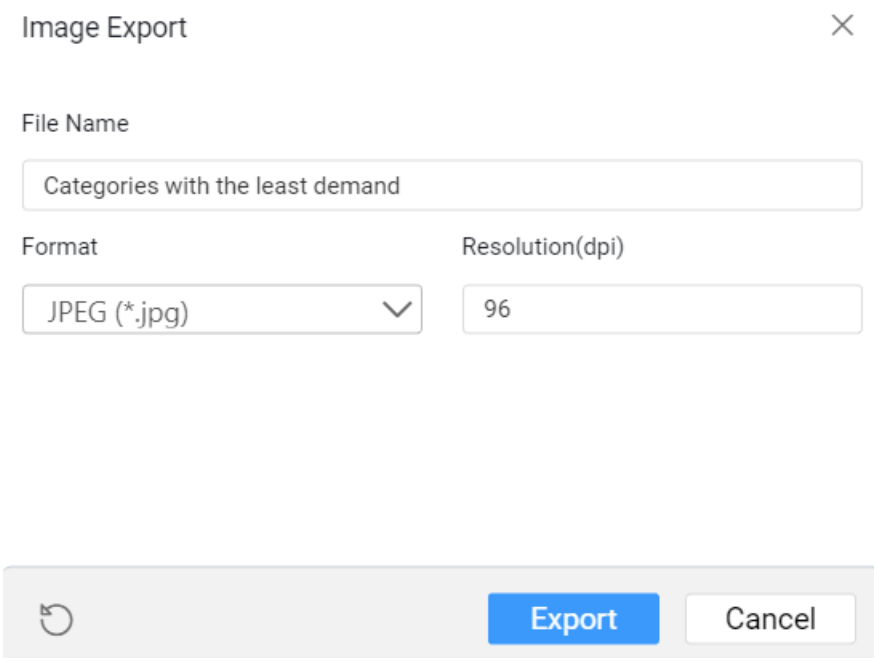


Right click on the widget that you want to export.

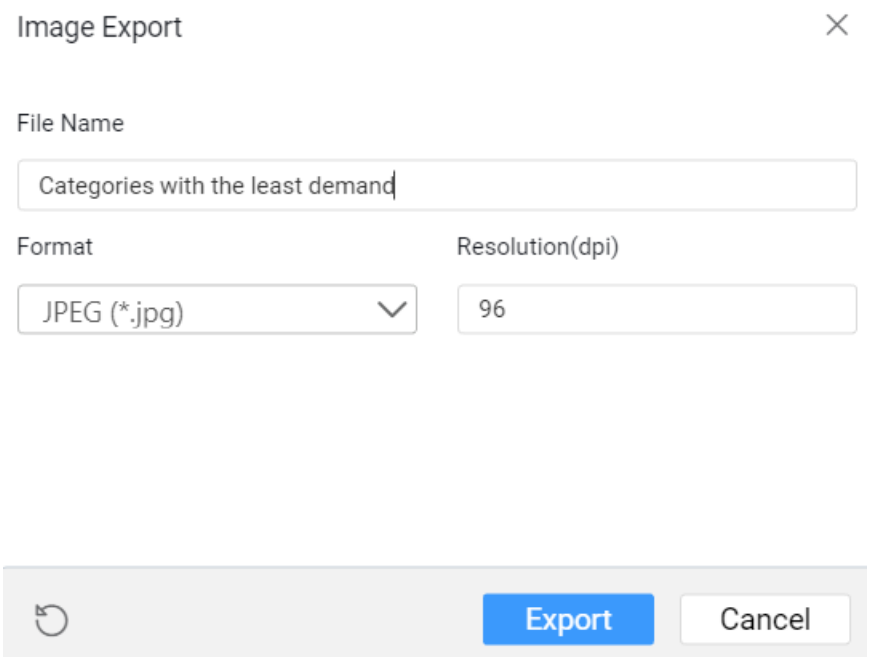
Select the Export option and choose Image option from list shown.



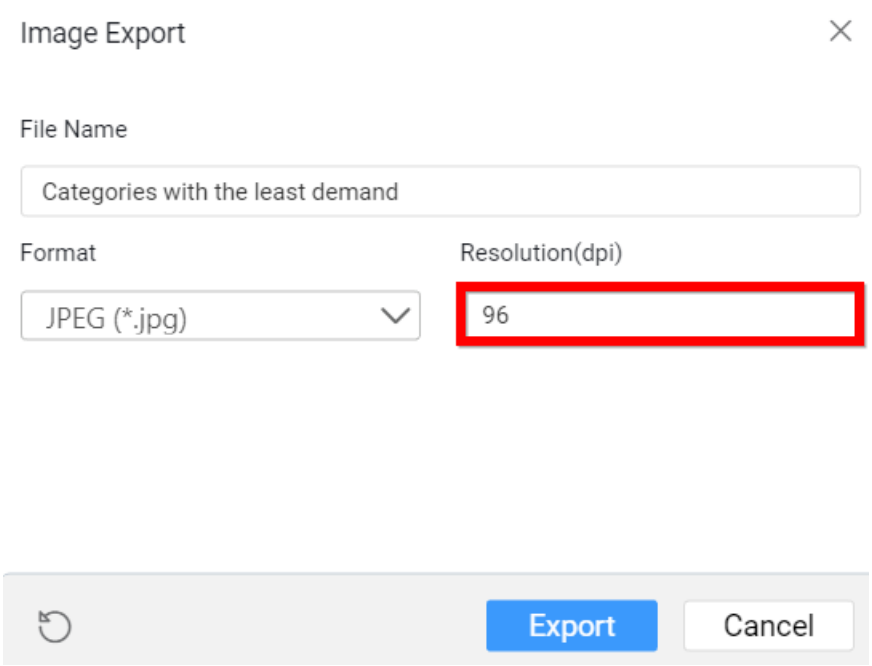
Click the Image option, the pop-up will be shown as follows.



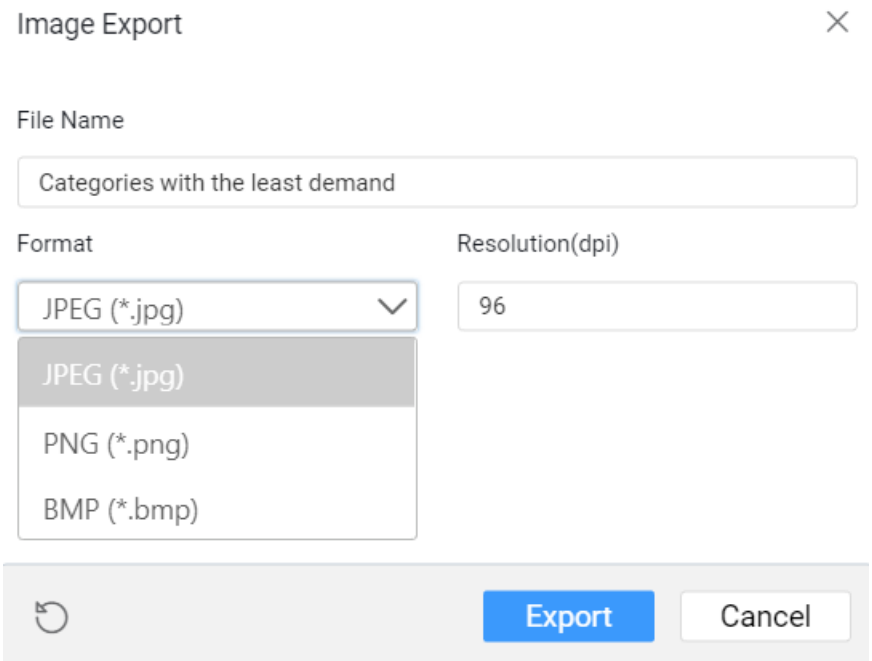
Enter the File Name fields with preferred values by replacing the default ones.



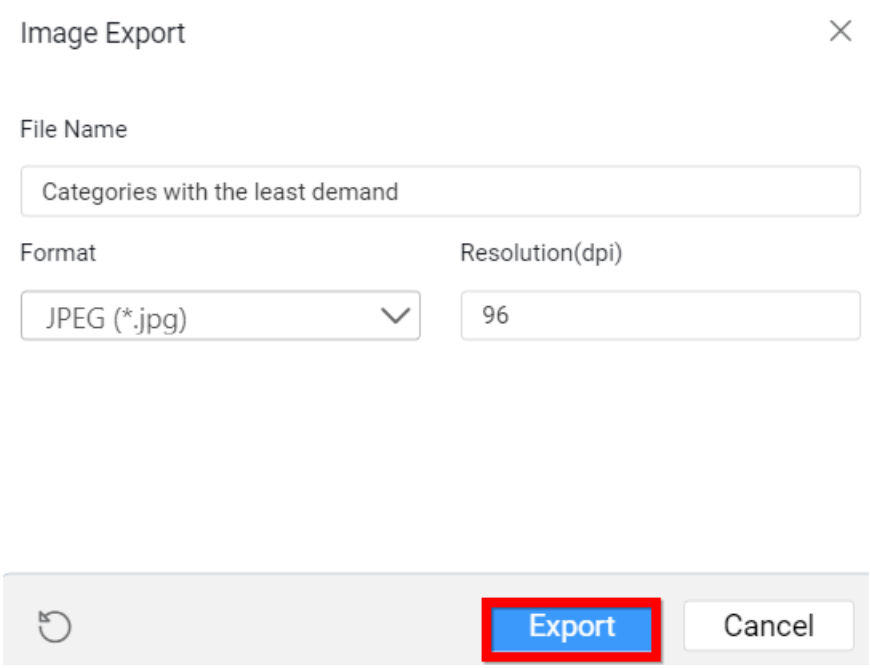
The default and the minimum value for Resolution is 96 (dpi). Maximum value allowed to set is 1790 (dpi).



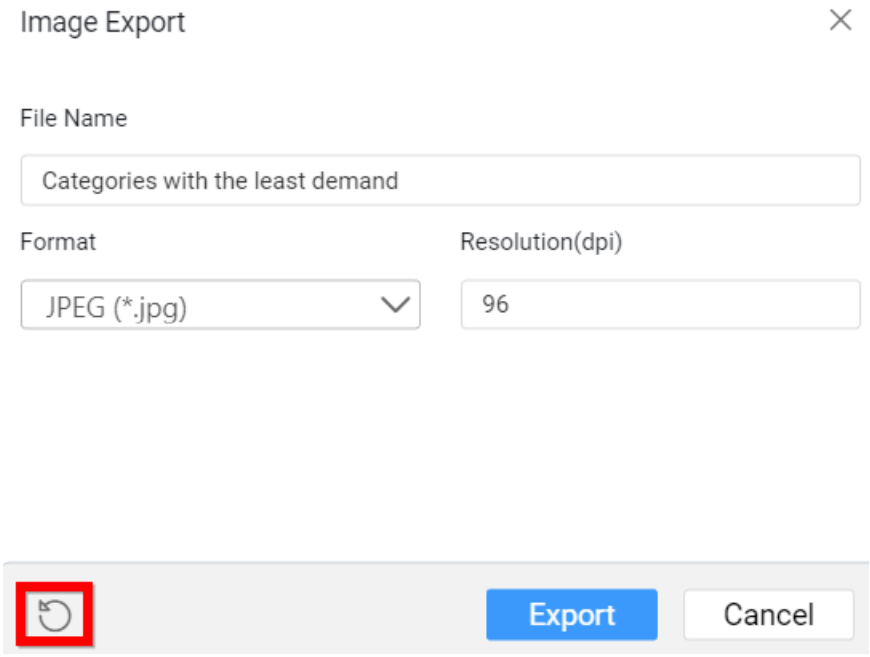
You can choose the image Format as JPEG (*.jpg), PNG (*.png), or BMP (*.bmp) file format.



Click the Export button, the current view of the widget will be displayed in the chosen image format with applied settings.



Click the Reset button, the default values get restored in the pop-up.



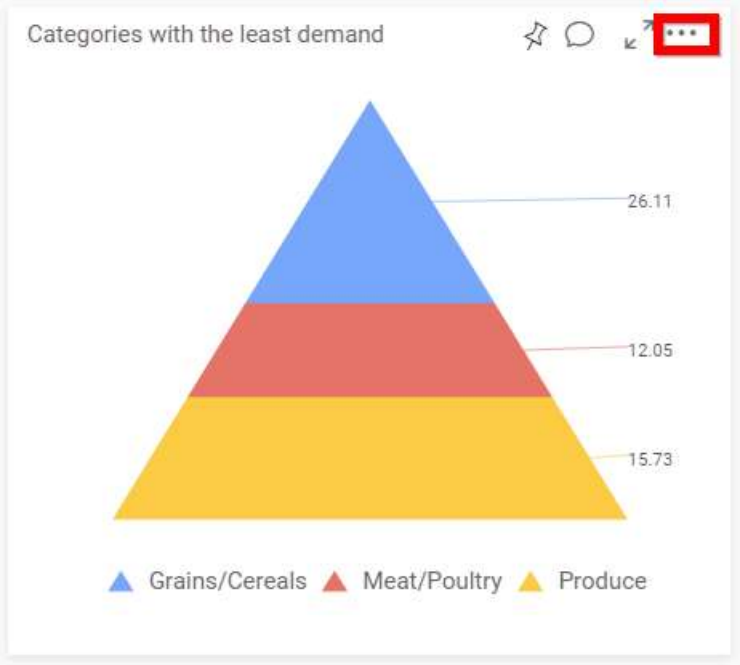
Note: In Windows Operating System, Safari browser does not support the Export to Image functionality. Whereas the Mac OS or iOS that have Safari browser version 6+ support the Export to Image functionality.

Exporting Widget to PDF

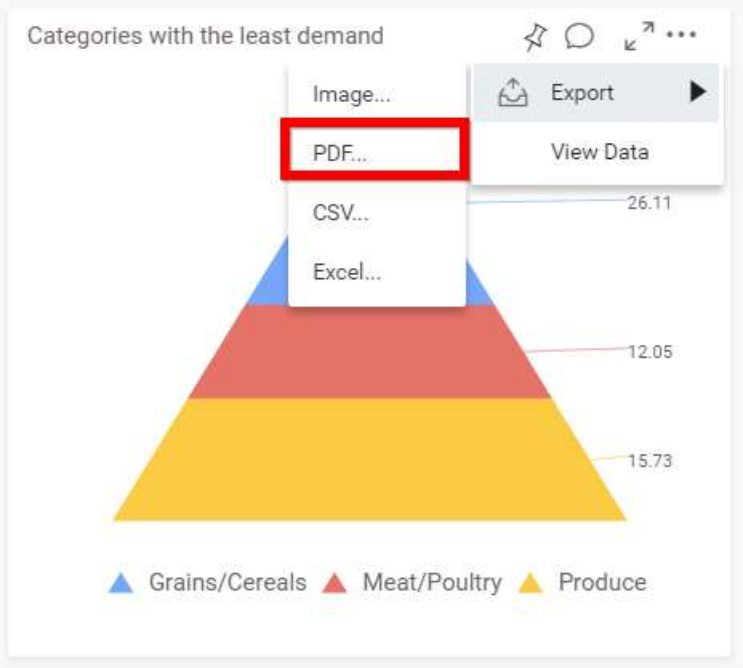
You can export widget as PDF in following ways.

By using More Options icon at right corner of the title section.

Click More Options icon in the widget at the right corner of the title section.



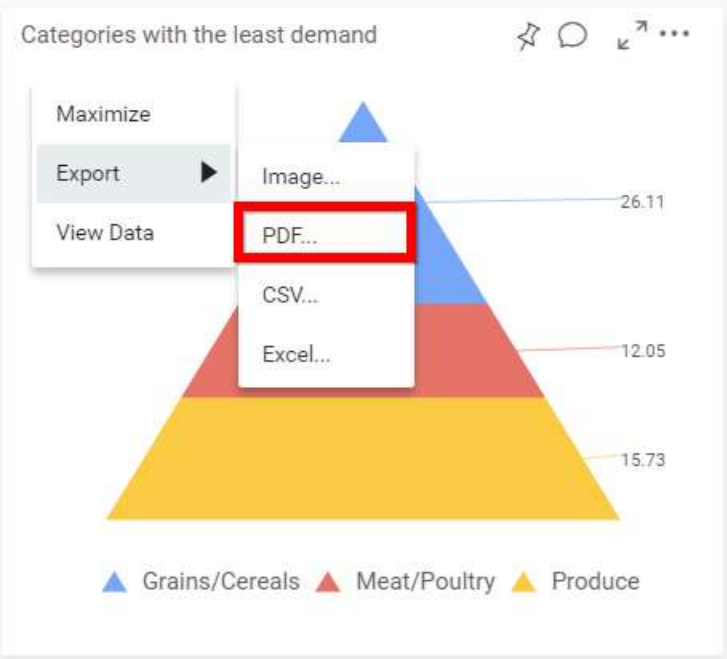
Select the Export option and choose PDF option from list shown.



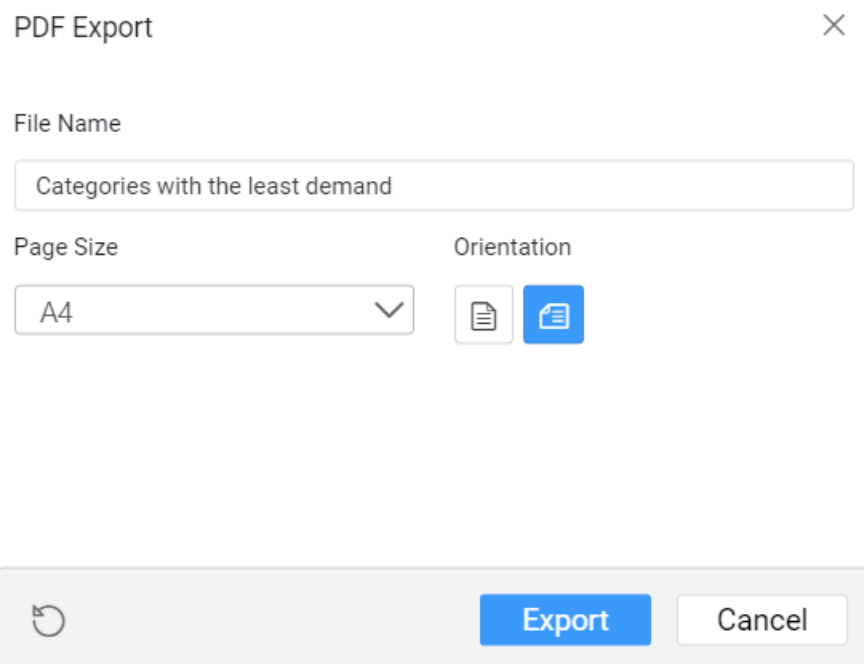
By clicking the Export icon at the right corner of the title section in maximized view.

Right click on the widget that you want to export.

Select the Export option and choose PDF option from list shown.

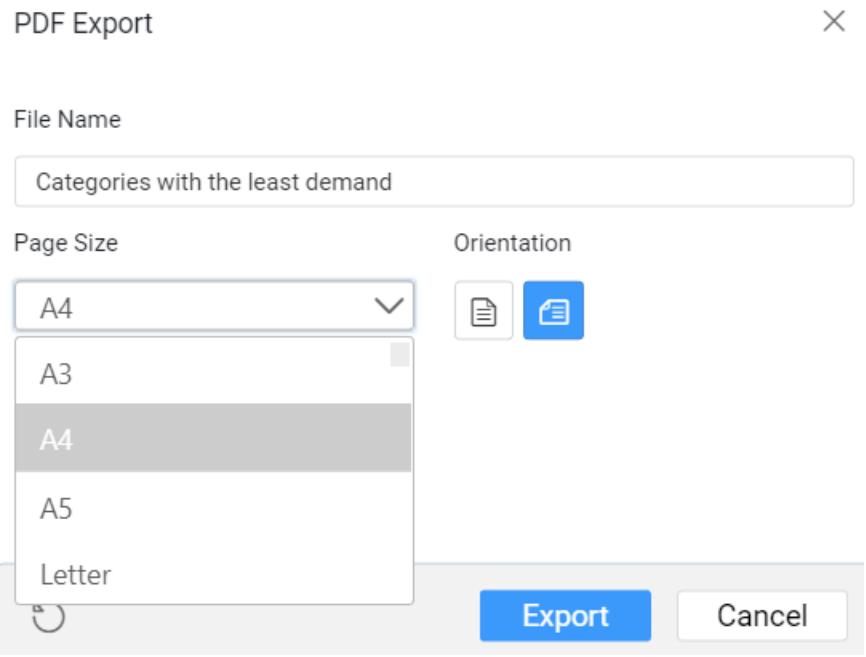


Click the PDF option, the pop-up will be shown as follows.

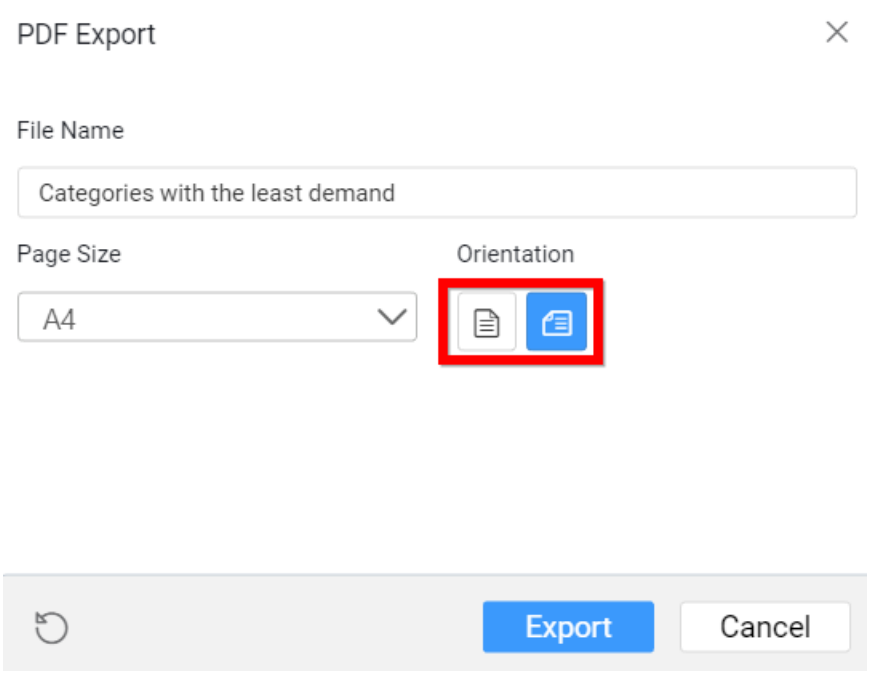


Set the File Name fields with preferred values by replacing the default ones.

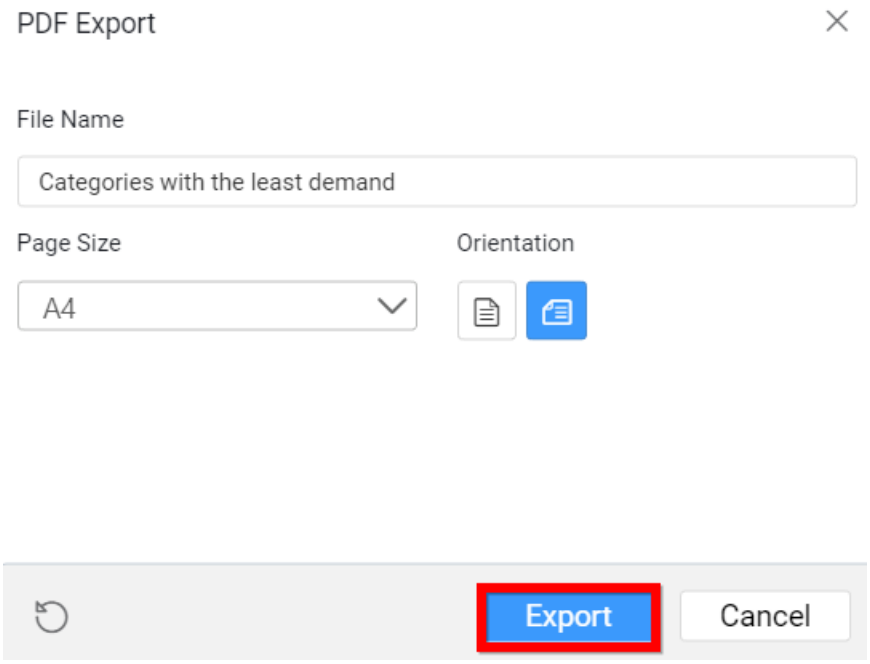
Set the preferred Page Size of the PDF File.



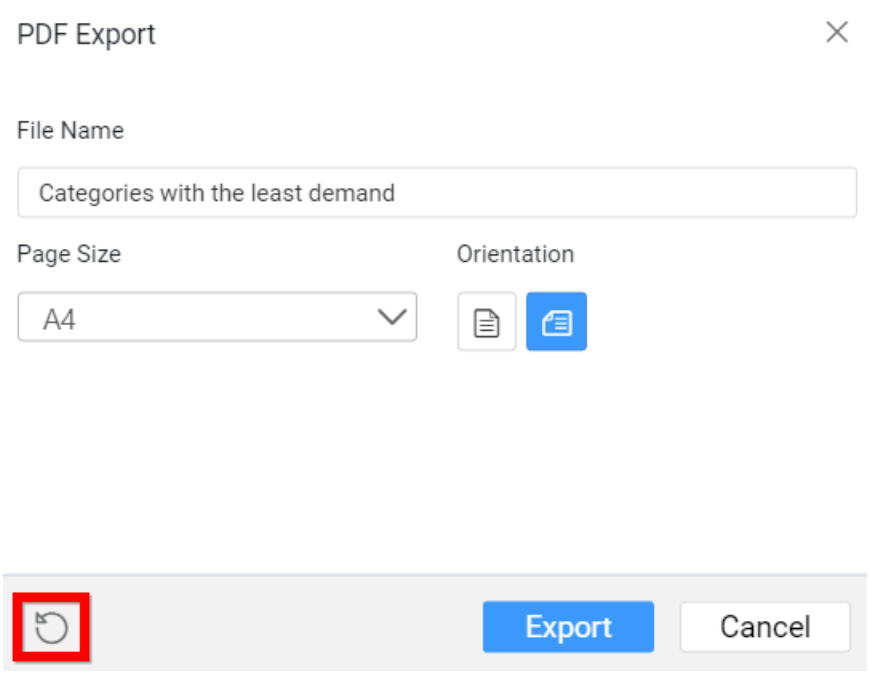
Choose the Orientation of the page as either portrait or landscape mode.



Click the Export button, the data in the widget will be displayed in the PDF file format.



Click the Reset button, the default values get restored in the pop-up.

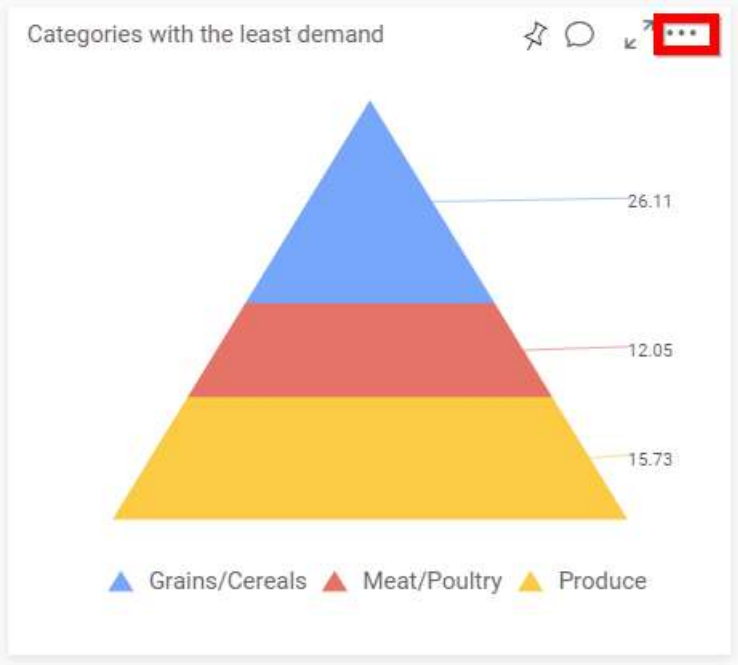


Exporting Widget to CSV

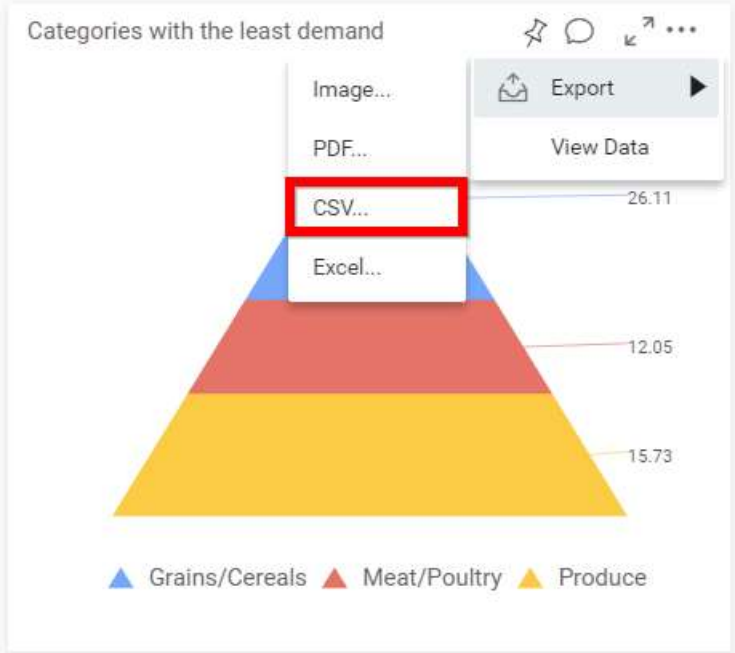
You can export widget data as CSV in following ways.

By using More Options icon at the right corner of the title section.

Click More Options icon in the widget at the right corner of the title section.



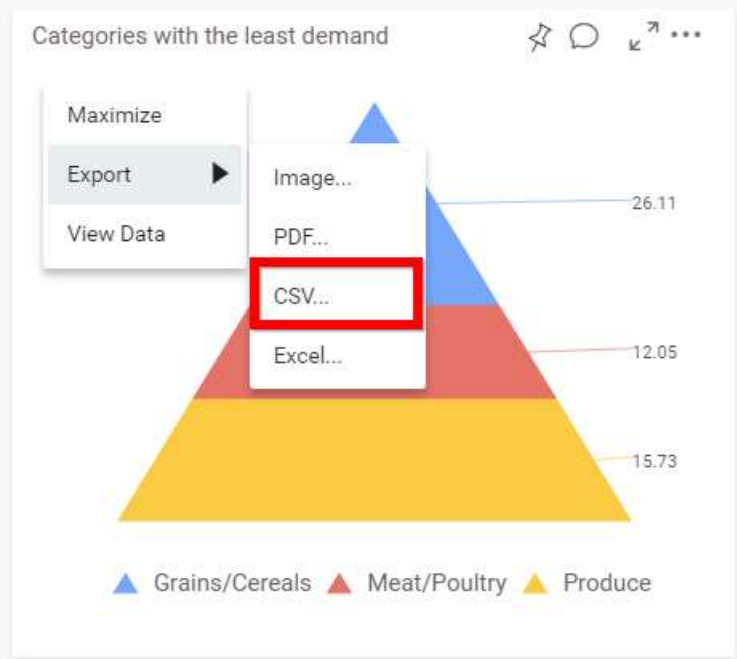
Select the Export option and choose CSV option from list shown.



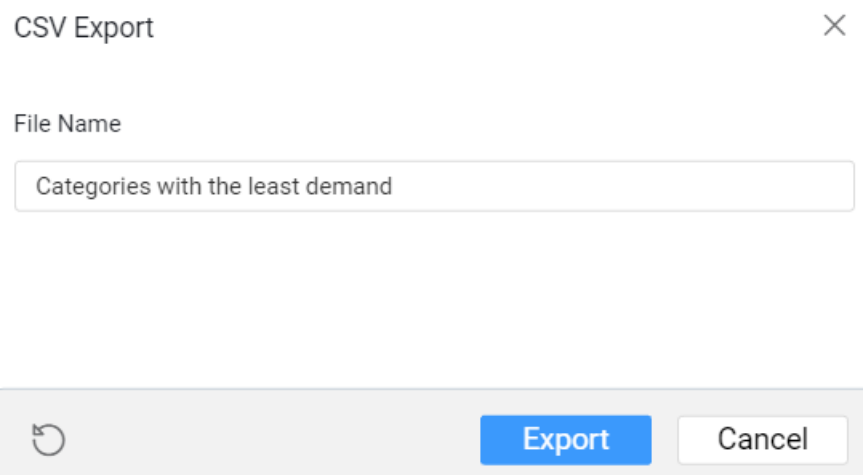
By clicking the CSV icon at right corner of the title section in maximized view.

Right click on the widget that you want to export.

Select the Export option and choose CSV option from list shown.

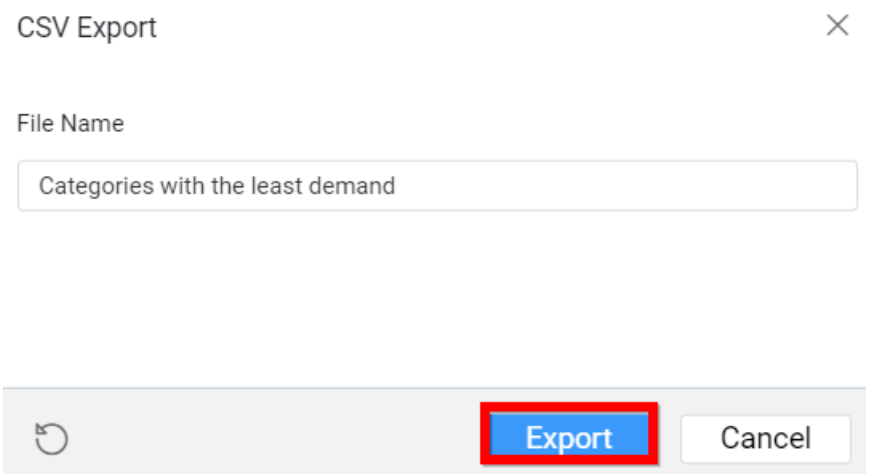


Click the CSV option, the pop-up will be shown as follows.

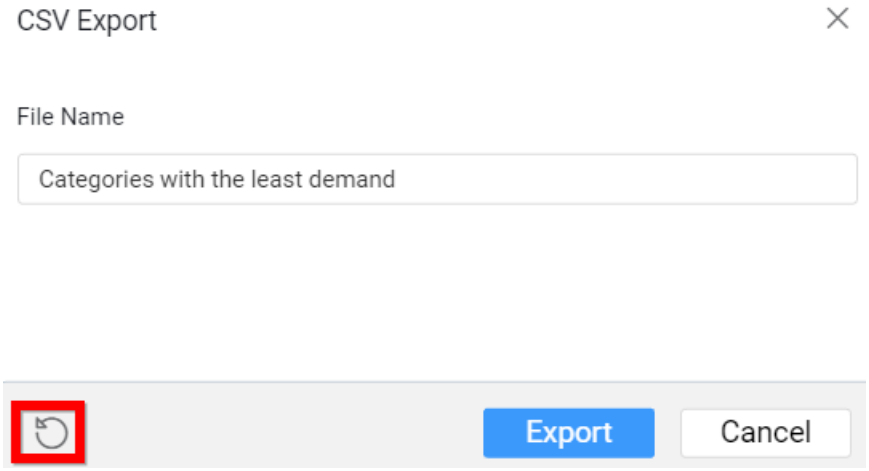


Set the File Name field with preferred value by replacing the default one.

Click the Export button, the data in the widget will be displayed in the CSV format.



Click the Reset button, the default values get restored in the pop-up.

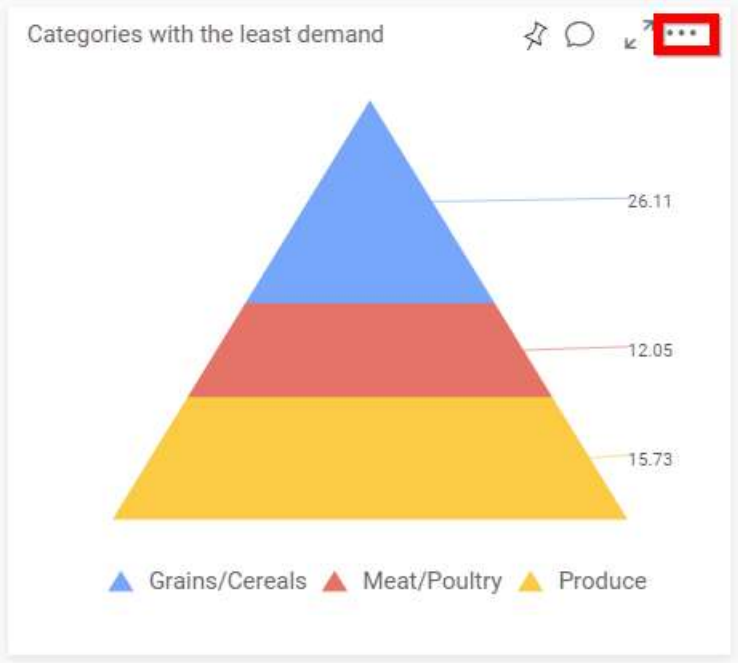


Exporting Widget to Excel

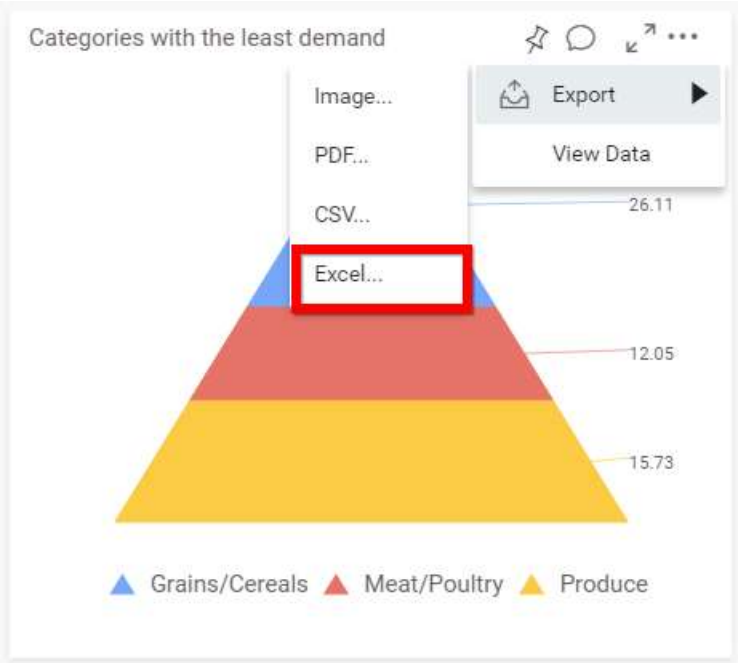
You can export widget data as Excel in following ways.

By using More Options icon at the right corner of the title section.

Click More Options icon in the widget at the right corner of the title section.



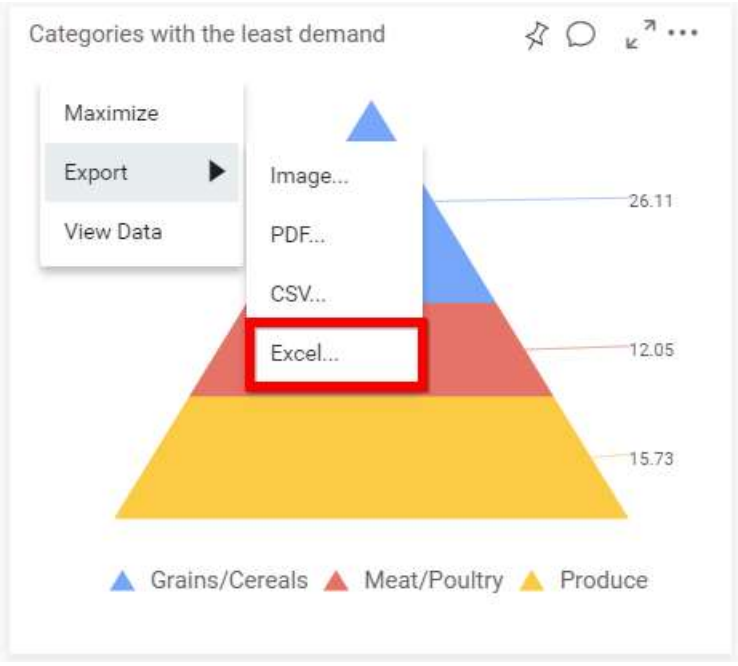
Select the Export option and choose Excel option from list shown.



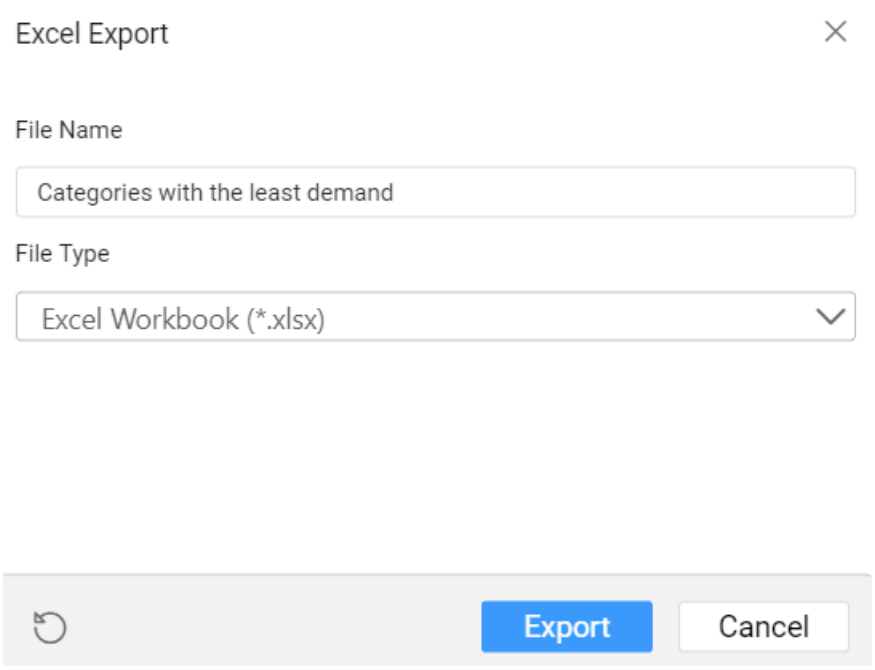
By clicking the Export icon at right corner of the title section in maximized view.

Right click on the widget that you want to export.

Select the Export option and choose Excel option from list shown.

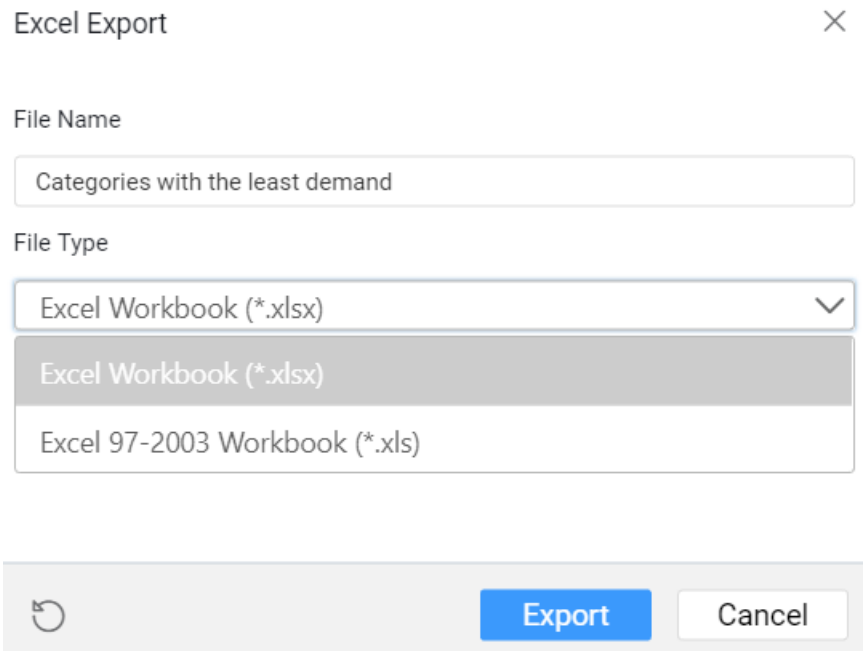


Click the Excel option, the pop-up will be shown as follows.

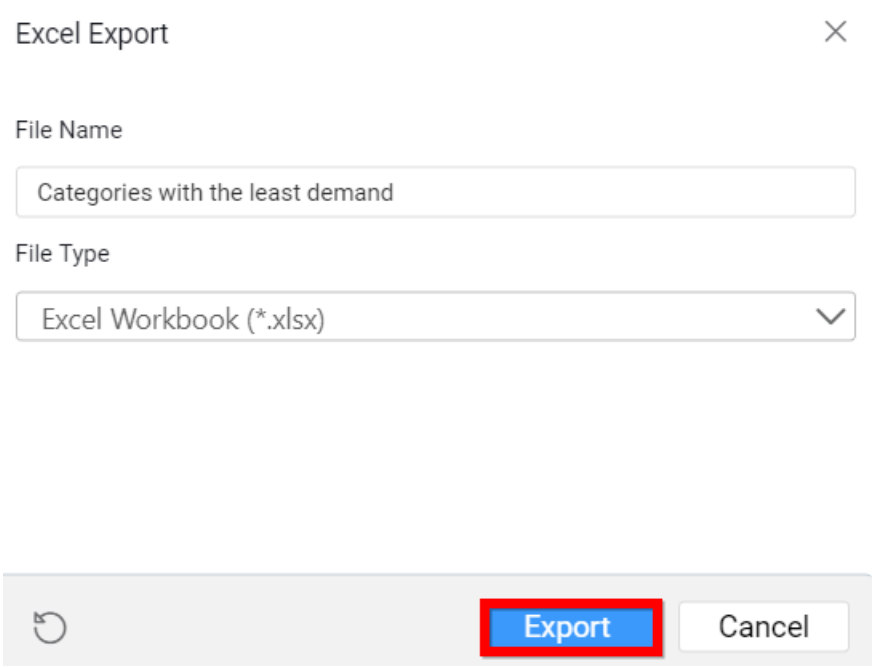


Set the File Name field with preferred value by replacing the default one.

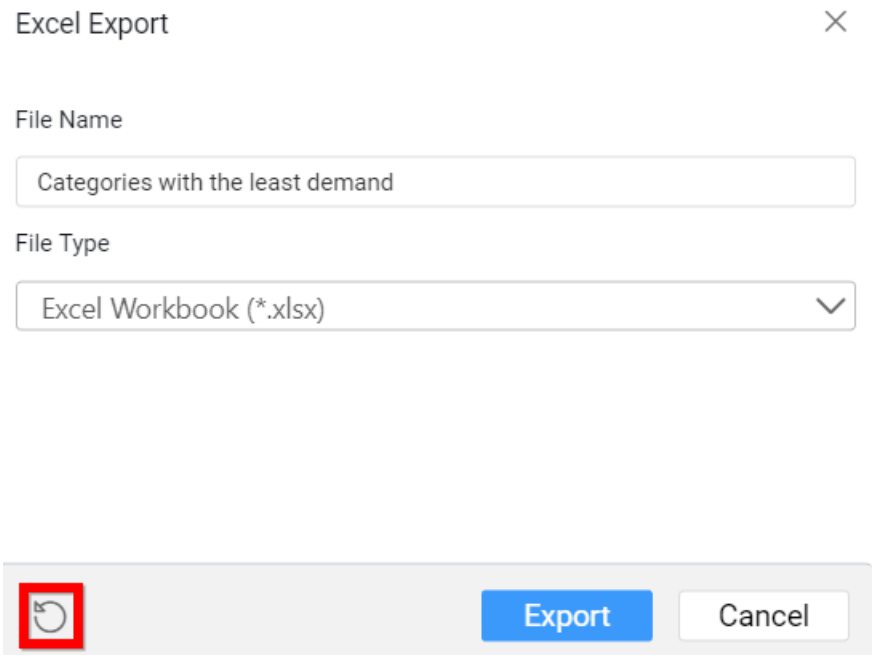
Choose the Format as either Excel 97-2003 Workbook (*.xls) or Excel Workbook (*.xlsx).



Click the Export button, the data in the widget will be displayed in the Excel format.



Click the Reset button, the default values get restored in the pop-up.



Filtering Views through URL Parameters

Passing Parameter With URL

You can pass parameters to a dashboard by including them in a dashboard URL. Passing parameter values within URL will apply filter in the dashboard on initial load itself.

To set a dashboard parameter within a URL, use the following syntax:

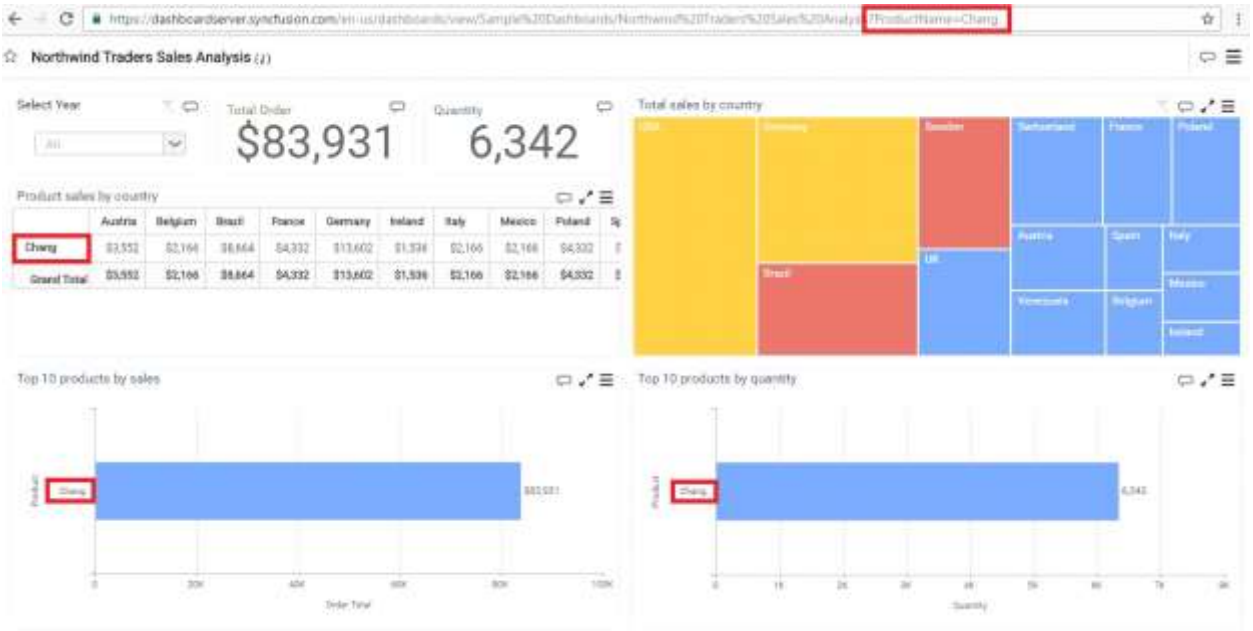
`parameter=value1, value2,..., valueN`

where parameter represents the column name.

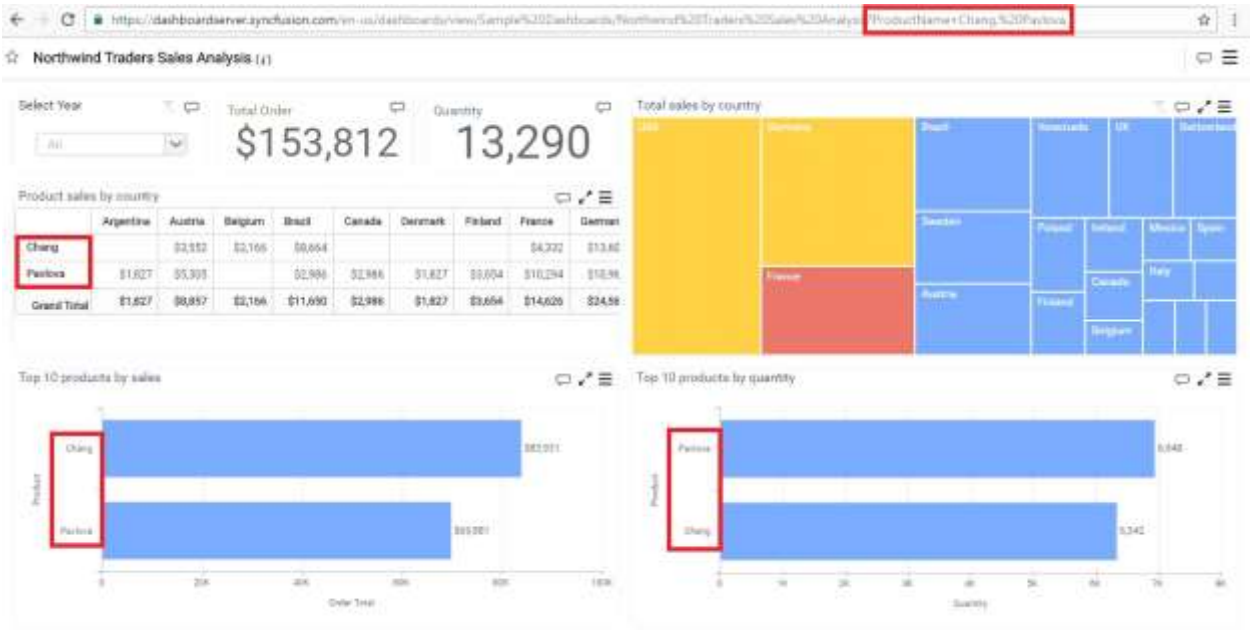
To append your query string made with parameters and values, to URL, add a prefix (?) to the query string like below.

http://<servername>/<culturename>/dashboards/view/<category>/<dashboardname>?ProductName=Chang

Here is a dashboard view illustrating the same with single-valued parameter.



Here is a dashboard view illustrating the same with multi-valued parameter.



Supported Operators and Date & Time Functions

You can also define parameters with operators to search for one or more values like below.

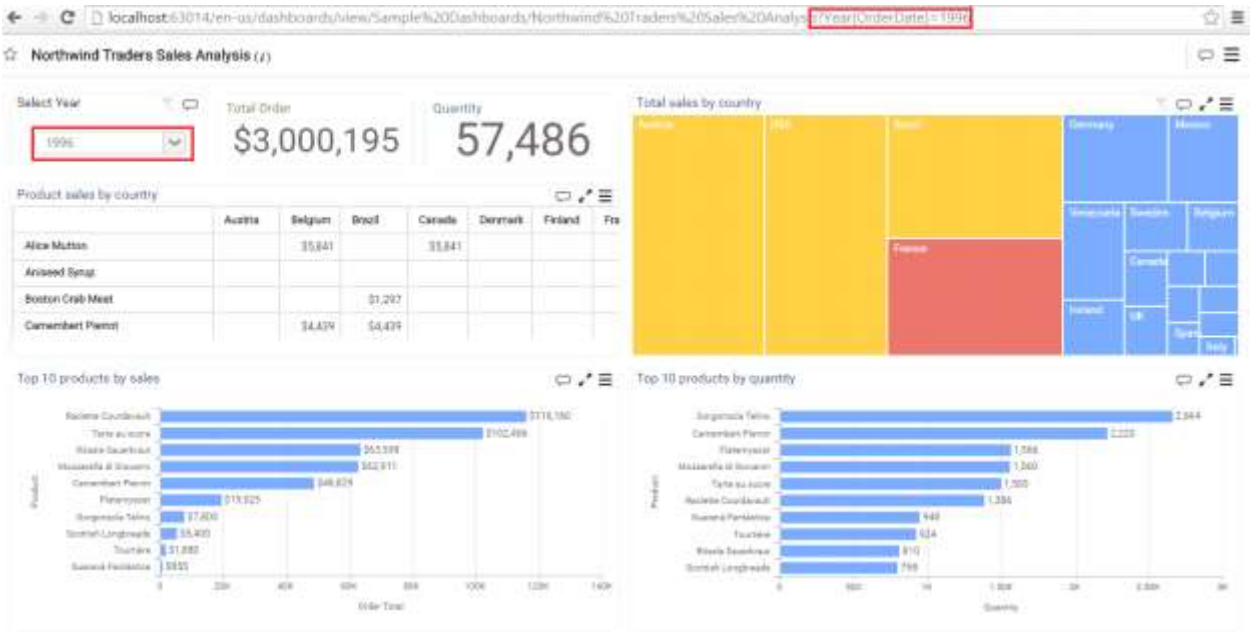
Operator	Syntax
IN	parameter=IN(value1, value2,..., valueN)
NOTIN	parameter=NOTIN(value1, value2, â€¦, valueN)
BETWEEN	parameter=BETWEEN(value1, value2)
INBETWEEN	parameter=INBETWEEN(value1, value2)
STARTSWITH	parameter=STARTSWITH(value)
ENDSWITH	parameter=ENDSWITH(value)
CONTAINS	parameter=CONTAINS(value1, value2)

You can define parameters (date and time typed columns) with date & time functions applied to search for formatted date values like below.

http://<servername>/<culturename>/dashboards/view/<category>/<dashboardname>?Year(OrderDate)=1996

Function	Syntax
YEAR	YEAR(parameter)=value1, value2, â€¦, valueN
MONTHNAME	MONTHNAME(parameter)=value1, value2, â€¦, valueN
QUARTER	QUARTER(parameter)=value1, value2, â€¦, valueN
QUARTERYEAR	QUARTERYEAR(parameter)=value1, value2, â€¦, valueN
MONTHYEAR	MONTHYEAR(parameter)=value1, value2, â€¦, valueN
DAYMONTHYEAR	DAYMONTHYEAR(parameter)=value1, value2, â€¦, valueN
MONTHDAYYEAR	MONTHDAYYEAR(parameter)=value1, value2, â€¦, valueN
HOURS	HOURS(parameter)=value1, value2, â€¦, valueN
MINUTES	MINUTES(parameter)=value1, value2, â€¦, valueN
DAY	DAY(parameter)=value1, value2, â€¦, valueN
SECONDS	SECONDS(parameter)=value1, value2, â€¦, valueN
DATEHOUR	DATEHOUR(parameter)=value1, value2, â€¦, valueN
DAYOFWEEK	DAYOFWEEK(parameter)=value1, value2, â€¦, valueN
DAYOFYEAR	DAYOFYEAR(parameter)=value1, value2, â€¦, valueN
WEEKOFYEAR	WEEKOFYEAR(parameter)=value1, value2, â€¦, valueN

Here is a dashboard view illustrating the use of parameter with date & time function.



Note: You can also define parameters with operators (except STARTSWITH and ENDSWITH) and date time functions combination.

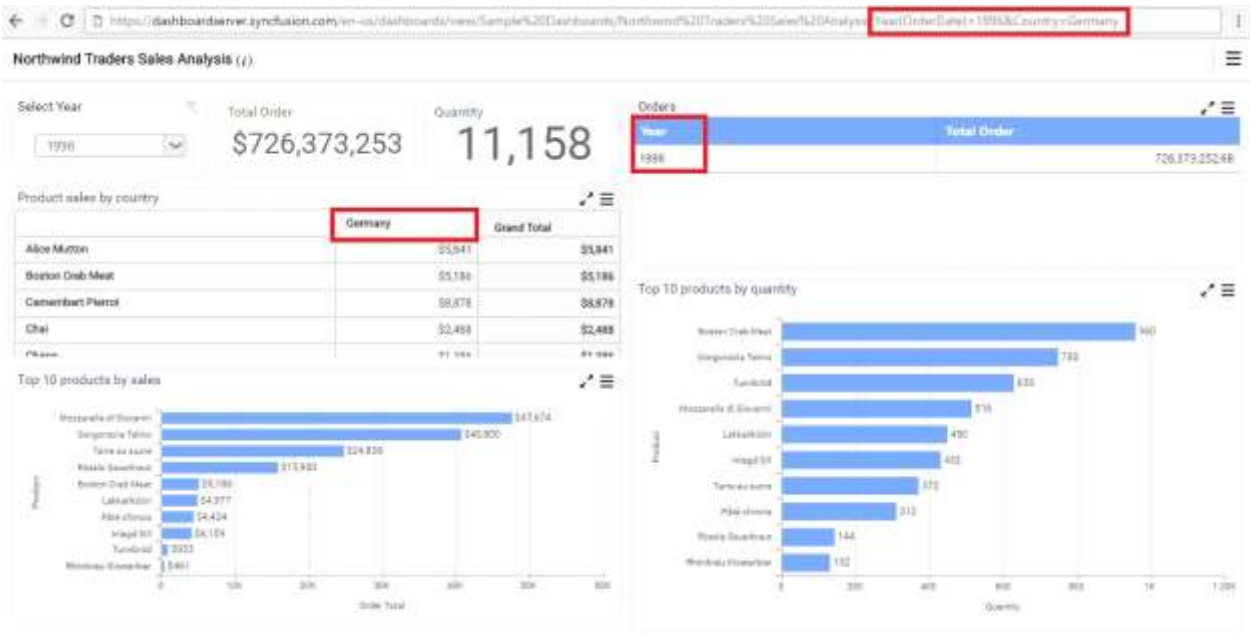
For example, YEAR(OrderDate)=IN(1996,1997)

Passing Multiple Parameters With URL

You can pass more than one parameter within a URL introducing an ampersand (&) symbol in between them to differentiate like below.

```
http://<servername>/<culturename>/dashboards/view/<category>/<dashboardname>?Year(OrderDate)=1996&Country=Germany
```

Here is a dashboard view illustrating the same.



Information: The parameter names and values are case-sensitive.

Information:

Information: The operators and date & time function names are case-insensitive.

Information:

Information: Characters like comma (,) and ampersand (&) in value should be prefixed and suffixed with tilde (~) symbol to differentiate itself from syntax elements.For example, CompanyName=Syncfusion Inc~,~

Information:

Information: Invalid parameter name will get ignored from filter consideration.

Information:

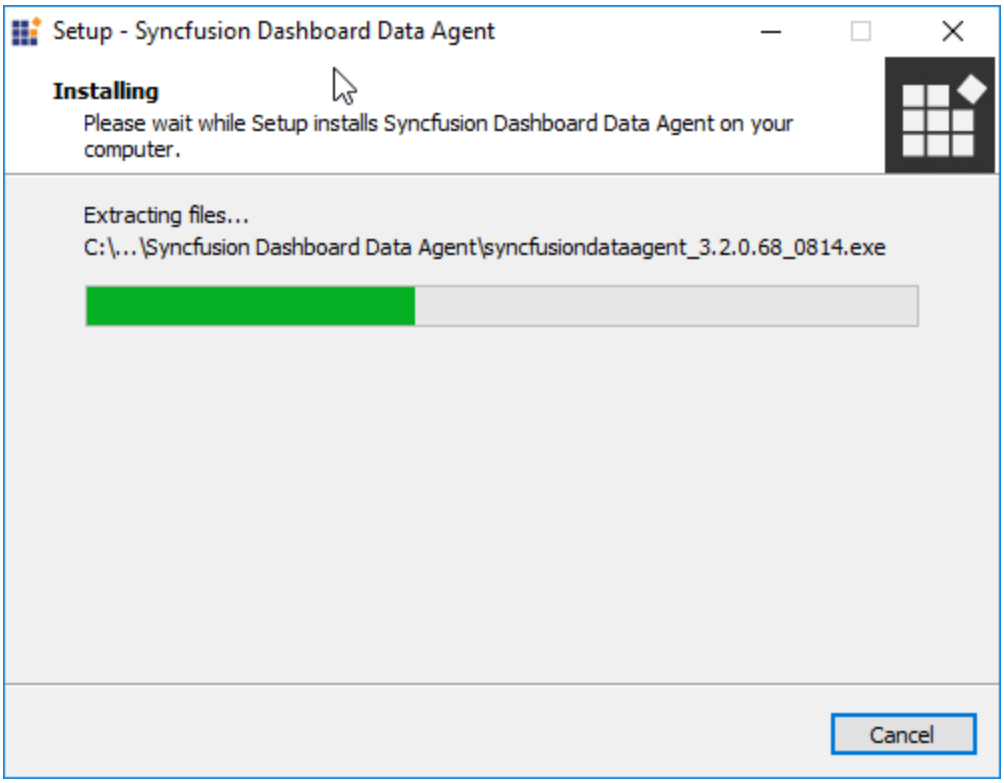
Information: Invalid parameter value will result in "No data available to display" in widgets.

Extracting Live Data

To import data from Salesforce Objects, RESTful Web Services, Microsoft Azure Table Storage, Microsoft Excel, CSV, Text Document, JSON, SQLite, Microsoft Access you need to install Syncfusion Dashboard Data Agent in the data server. This agent extracts the web accessible data to the data server and refresh the data to keep it alive based on provided time schedules.

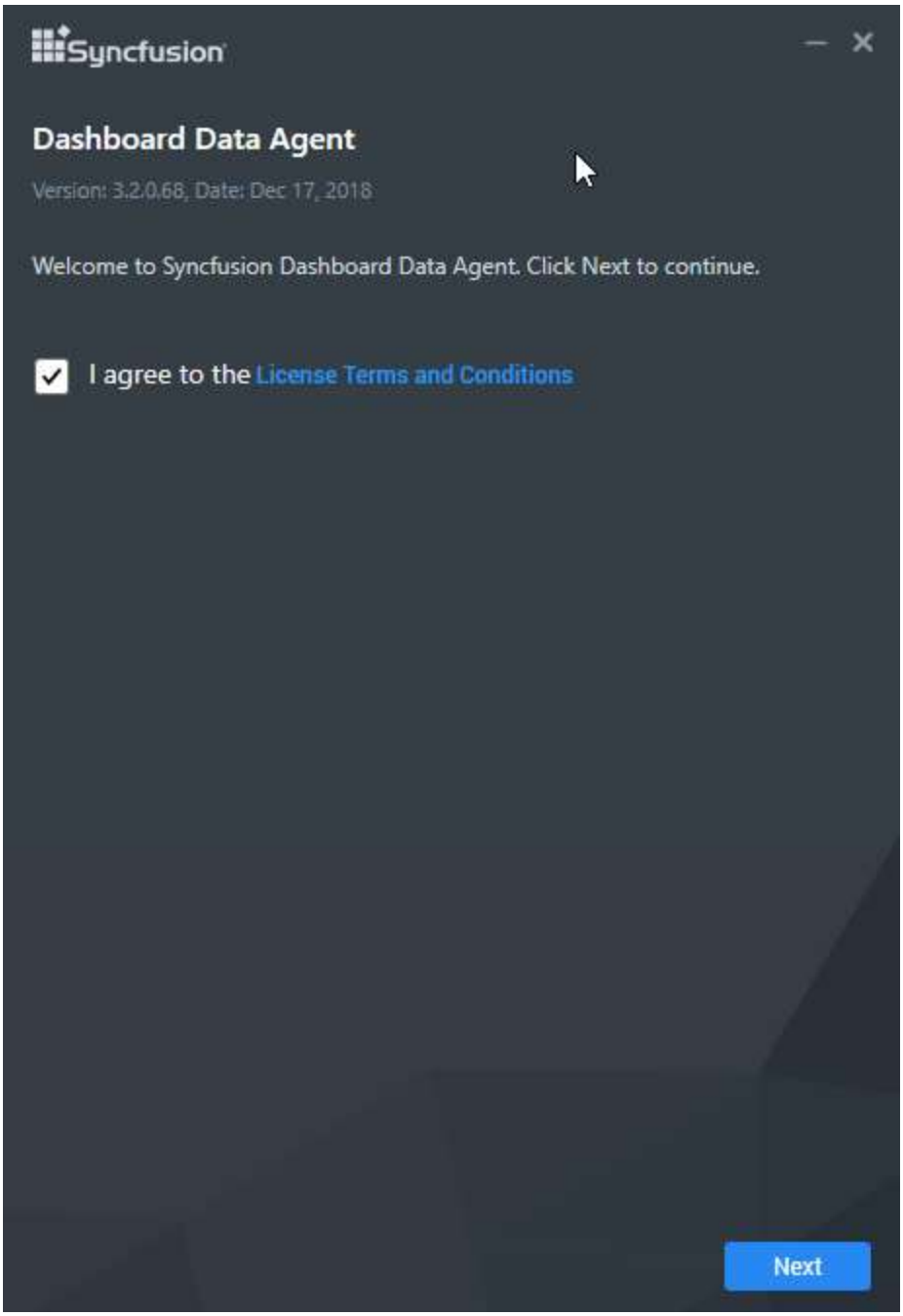
Installing Syncfusion Dashboard Data Agent

Run the installer either through clicking the Run button or by double-clicking the executable file from the saved location.



The installer extracts the files and launch the installation wizard.

Read and accept the license terms and conditions through checking the option I agree to the License Terms and Conditions and click Next.

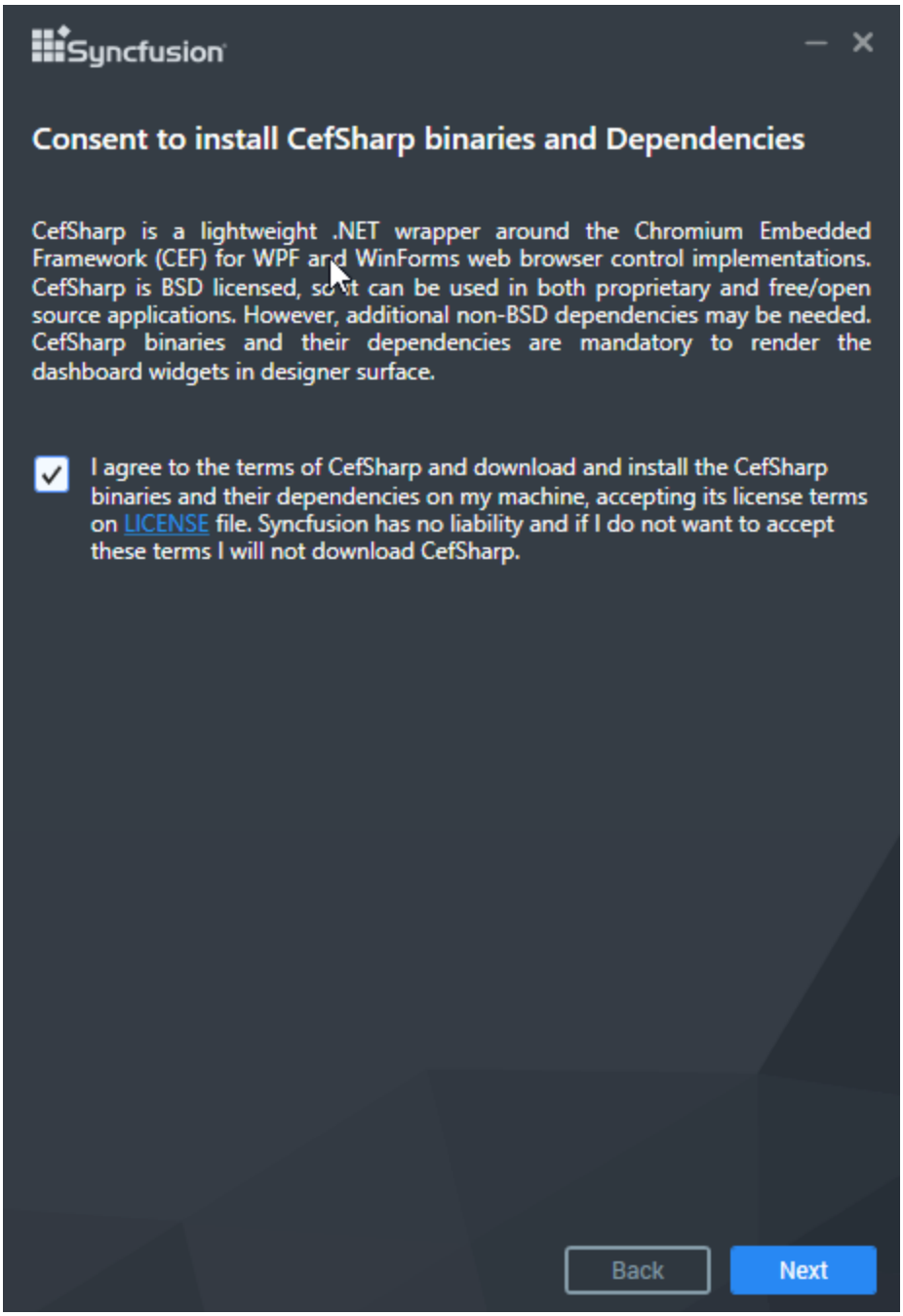


Browse to the location where you would like to install the Syncfusion Dashboard Data Agent and click Next.

Browse to the location where you want to install the Dashboard Designer application, and then click Next.

Read and accept the Consent the install CefSharp binaries and Dependencies and click NEXT.

Information: cefSharp is a lightweight .NET wrapper around the Chromium Embedded Framework (CEF) used for WPF and WinForms web browser control implementations. CefSharp is BSD licensed, so it can be used in both proprietary and free/open source applications. However, additional non-BSD dependencies may be needed. CefSharp binaries and their dependencies are mandatory to render dashboard widgets in the designer surface.

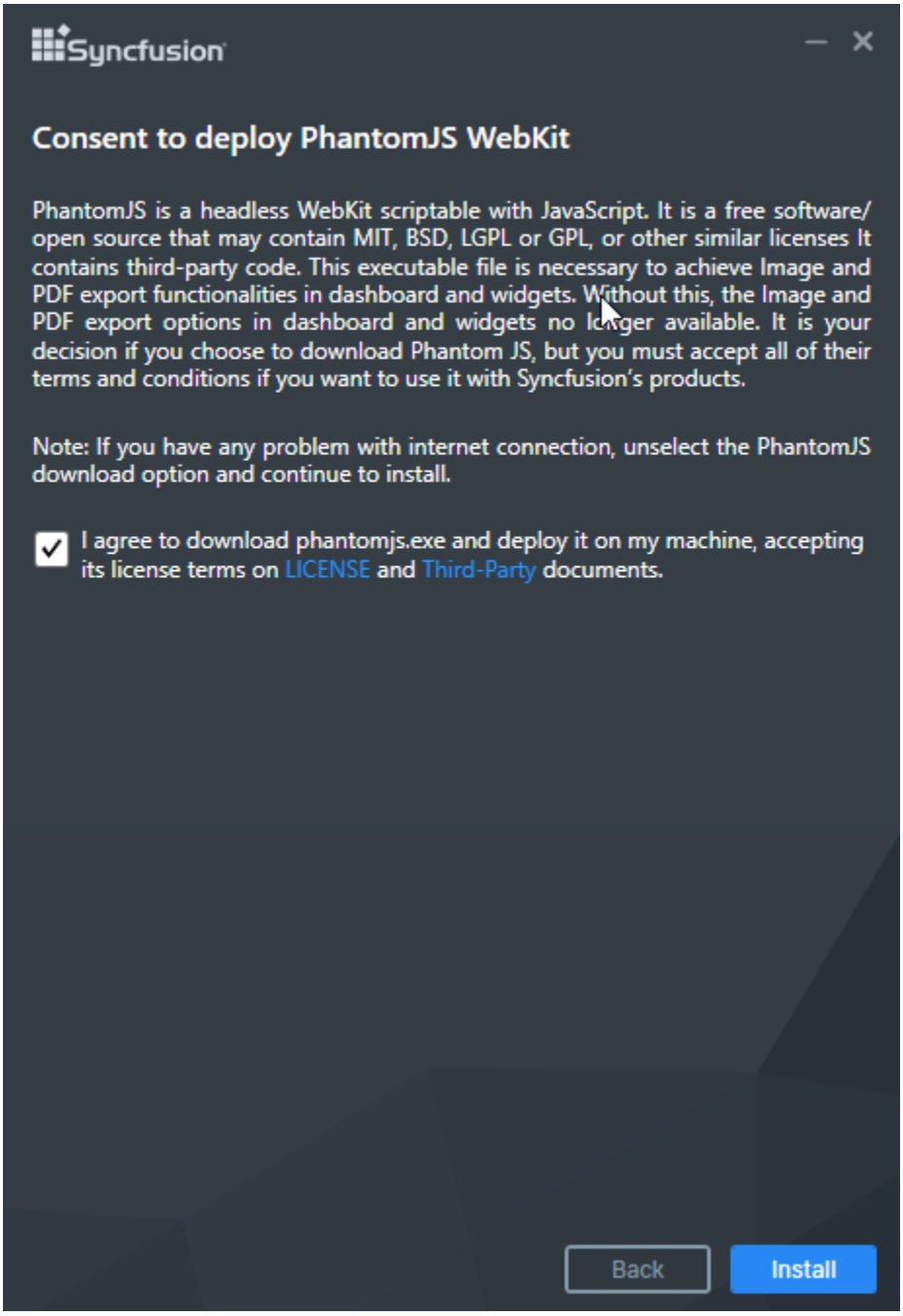


Read and accept the Consent to deploy PhantomJS Webkit and click Install.

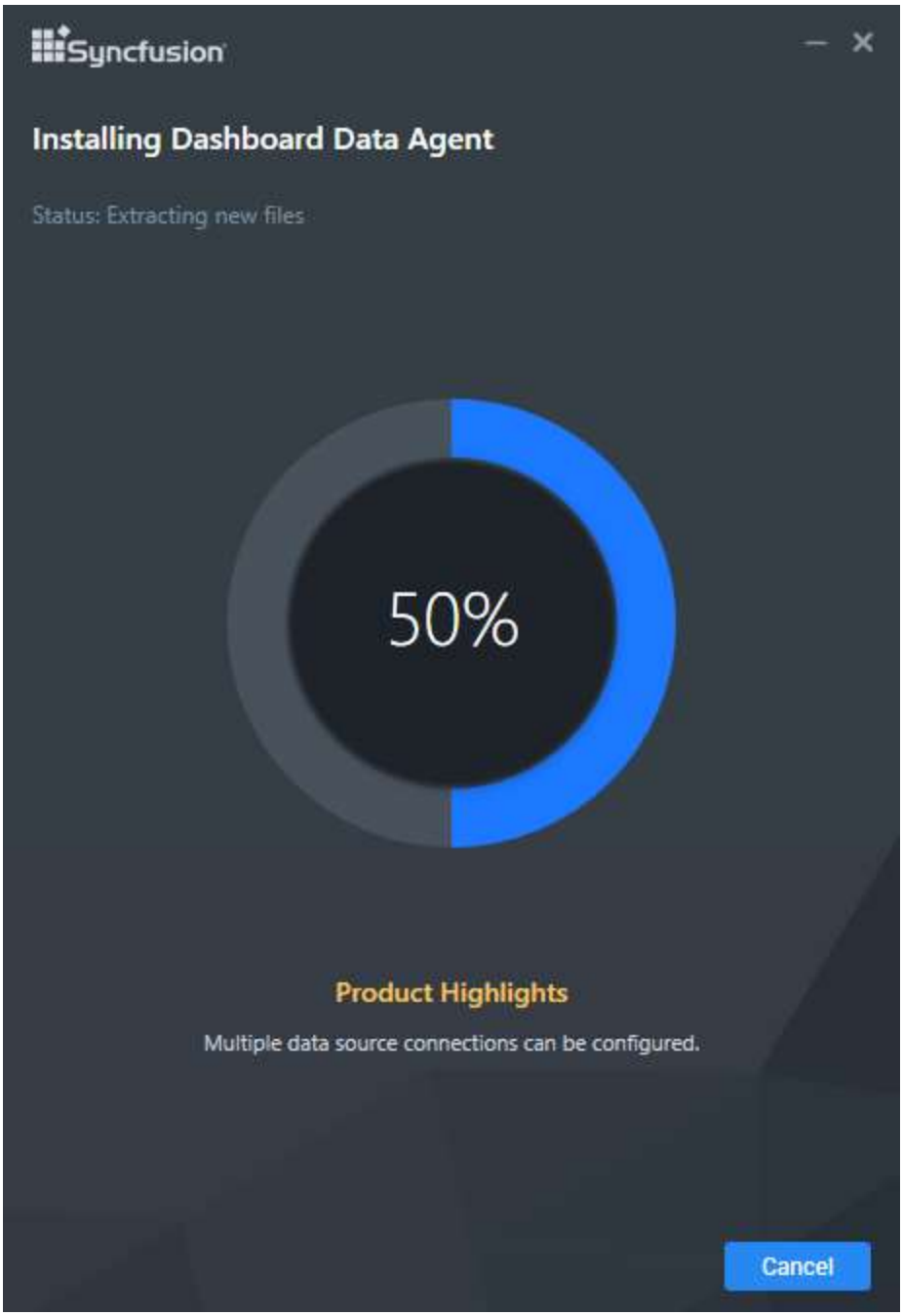
Information: PhantomJS is a headless WebKit scriptable with JavaScript. This is a free software/open source, and it may contain MIT, BSD, LGPL, or GPL, or other similar licenses that contain third-party code. This executable file is necessary to achieve Image and PDF export functionalities in the Dashboard and widgets. Without this file, the image and PDF export options in the Dashboard and widgets will no longer be available. If you choose to download PhantomJS, must accept all terms and conditions to use it with Syncfusion's products.

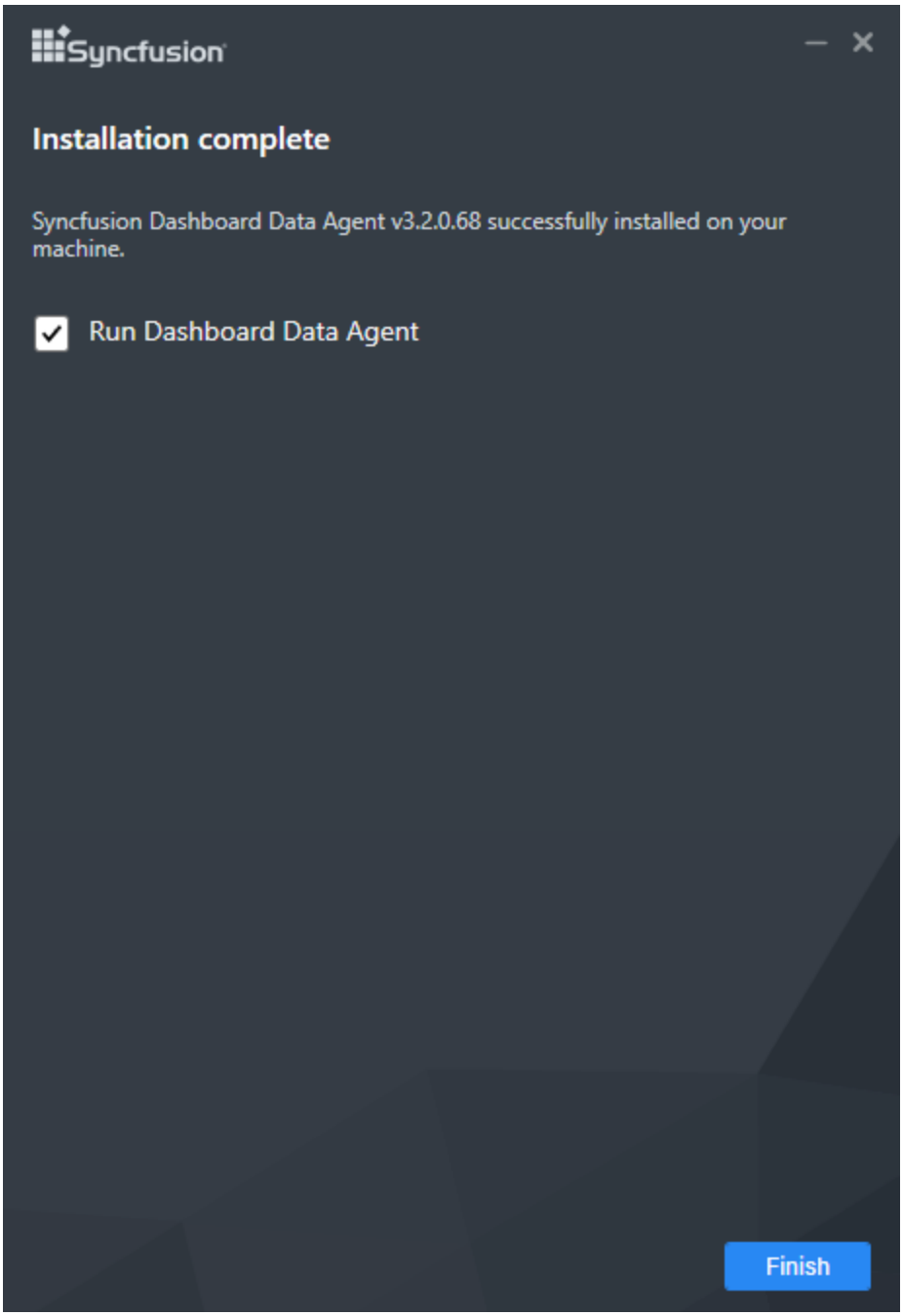
Note: If you have any problem with internet connection or do not have internet connection, unselect the PhantomJS download option and continue to install. To manually install the PhantomJS, please refer

this.



Now the installation begins. You can cancel the installation anytime through pressing Cancel, if needed.

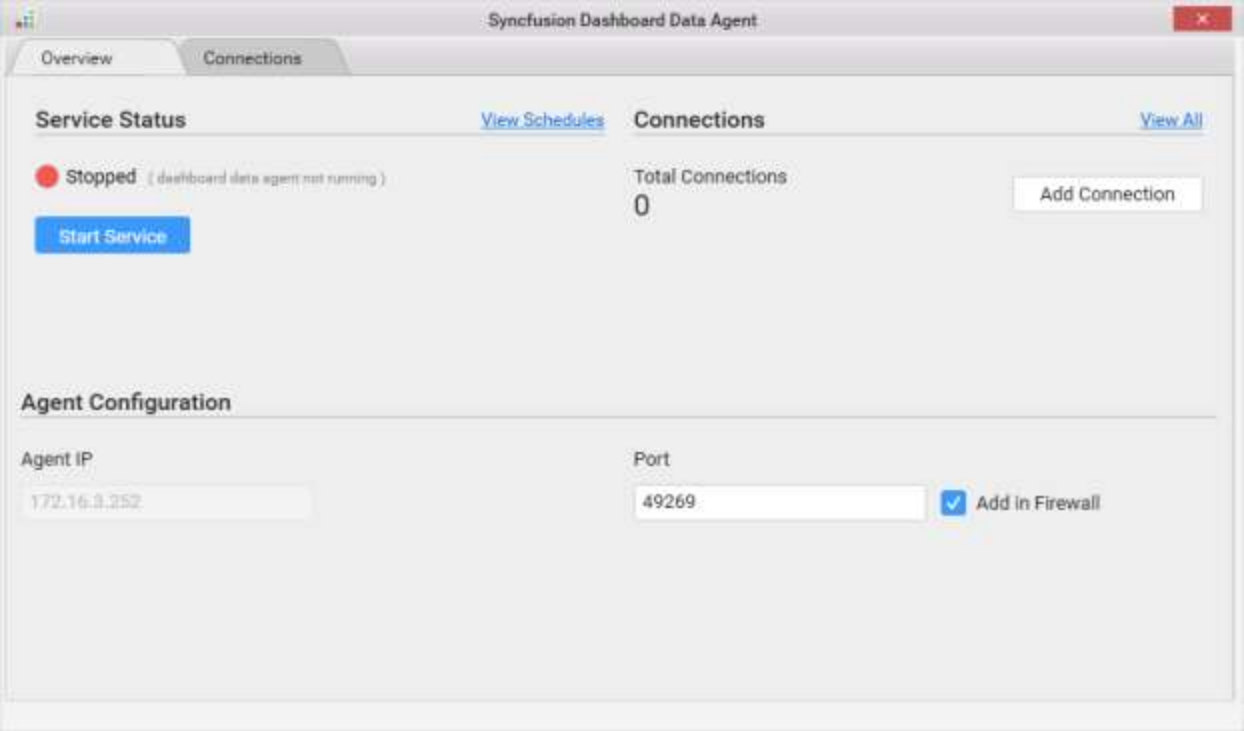




On successful installation, the above screen appears. Click Finish to close the installation wizard and run the newly installed Syncfusion Dashboard Data Agent. You can also run the application later by unchecking the option Run Data Agent.

Configuring Syncfusion Dashboard Data Agent

Syncfusion Dashboard Data Agent is a self-hosting service which is prepared to be installed only in data servers. It is responsible for extracting data from Salesforce objects, RESTful web services, Azure Table Storage, Excel, CSV, Text and full data update for web accessible resources on the time based on schedulers.

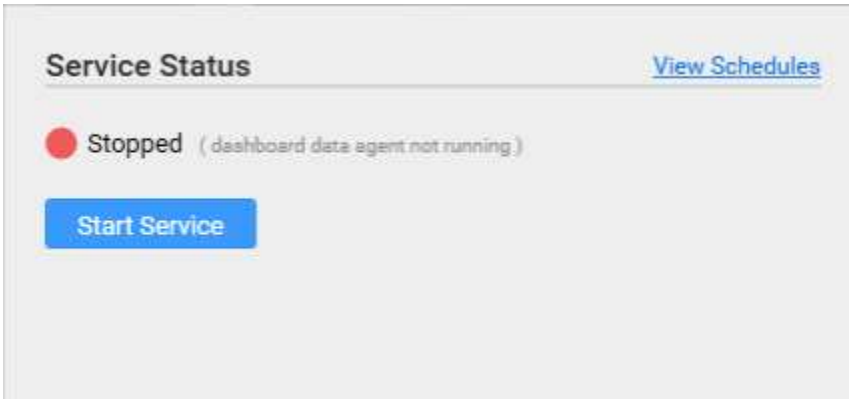


The Overview tab of the Syncfusion Dashboard Data Agent has following segments:

1. Service Status
2. Connections
3. Agent Configuration

Service Status

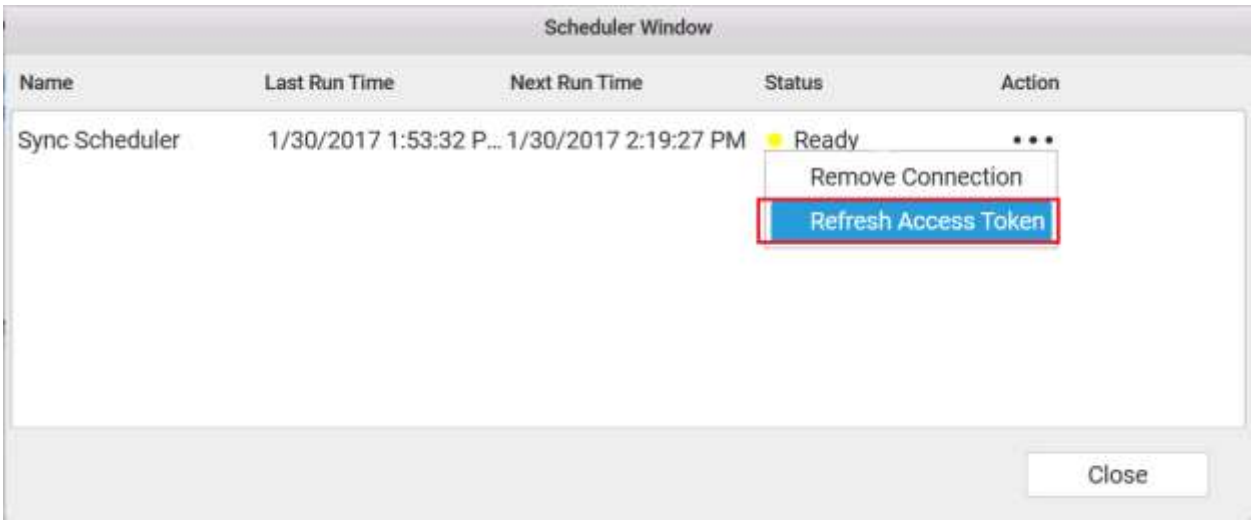
We can start the Dashboard Data Agent Service on a specific port by click on Start Service button.



Once you clicking on start service button the Syncfusion Dashboard Data Agent start the service on the specified port which was mentioned in Agent Configuration Section.

We can stop the Dashboard Data Agent Service on a specific port by click on Stop Service button at any time.

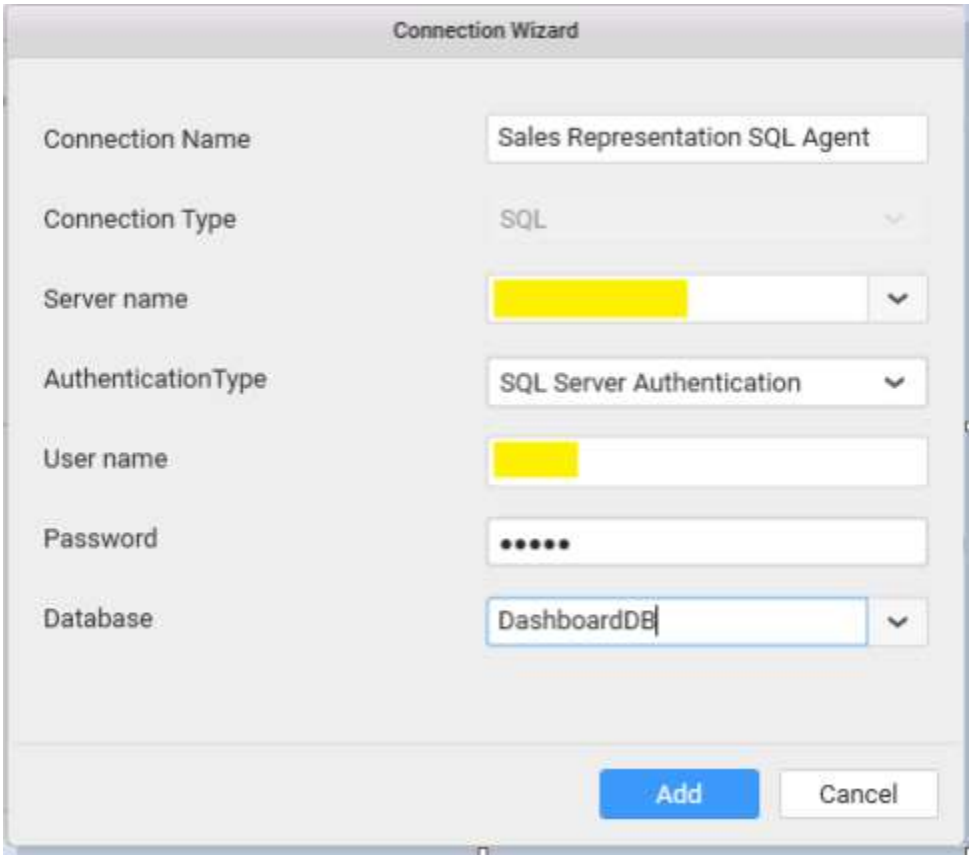
View Schedules lists the scheduled dashboard report in Scheduler Window.You can remove the particular schedule from the list by clicking Remove Connection option in Action menu of the scheduler list.



Note: Here the Refresh Access Token button enabled only when OAuth access token get expired.

Connections

We can add the Target Connection details (where the published data should be extracted) by clicking Add Connection button.



By Clicking Add Connection button the connection wizard open and fill the connection details in the required area to save the connection in Agent.

Once the connection was added in agent you can publish/extract the data on specific connection by using its connection name.

View All allows us to navigate to the connection tab.

Agent Configuration

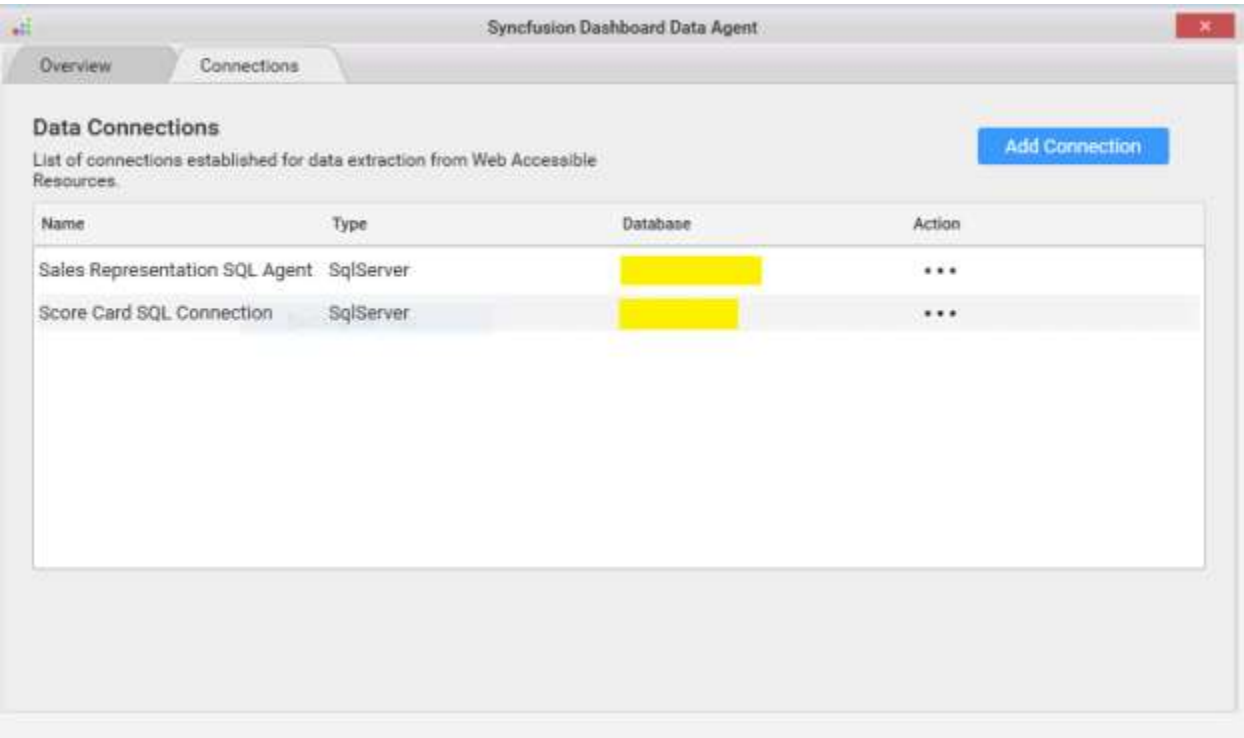
Agent IP text box shows you the IP of the machine where the Syncfusion Dashboard Data Agent was installed.

Port text box allows us to specify the port where the Dashboard Agent service need to be run.

Add in Firewall check box allows us to add a new inbound rule for the mentioned port in firewall.

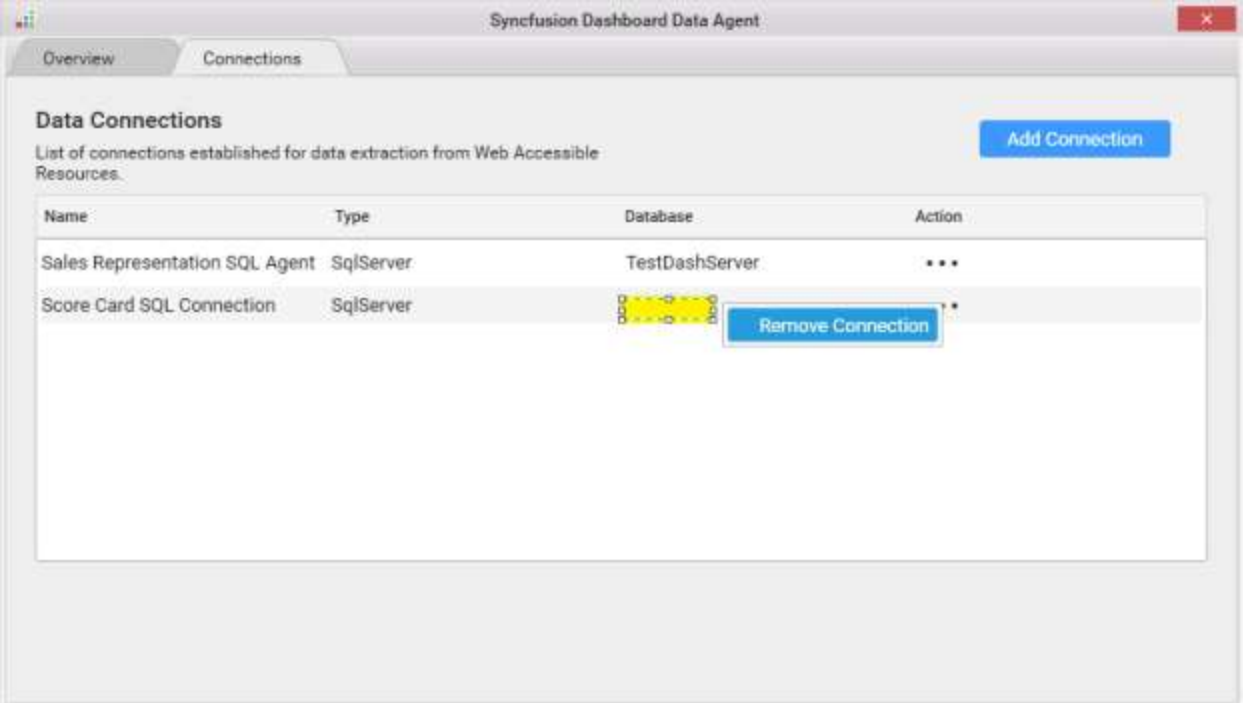
Note: If you want to access the Agent which runs on a specific port from the client machine then you need to add inbound rule for the specific port in firewall.

The Connections tab of the Syncfusion Dashboard Data Agent allows us to handle connections configuration in agent.



The Add Connection button in Connections tab is same as the Add Connection button in Overview tab which is used to add and configure connections in agent.

This tab lists the SQL connections that are configured in Syncfusion Dashboard Data Agent. You can remove the particular connection from the list by clicking Remove Connection option in Action menu of the connection from Data Connections list

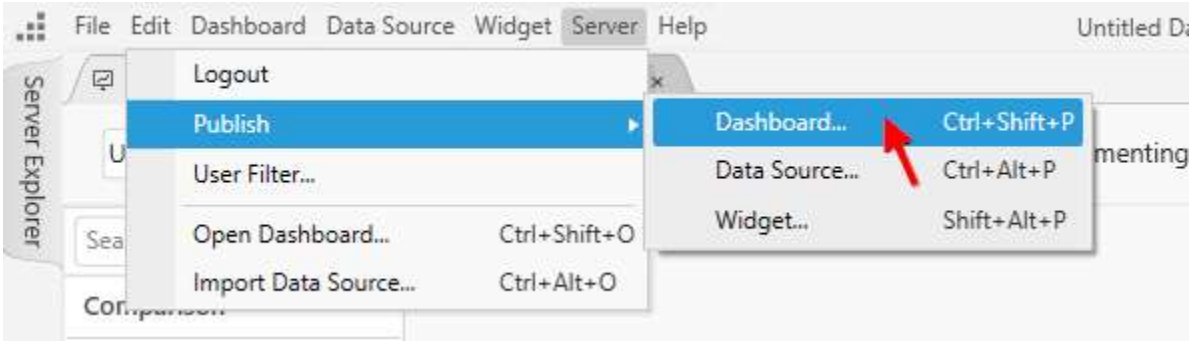


Note: Connection modification in agent is irrelevant to the Service state i.e. you can add/remove connection at any time, even when the service is in stop state.

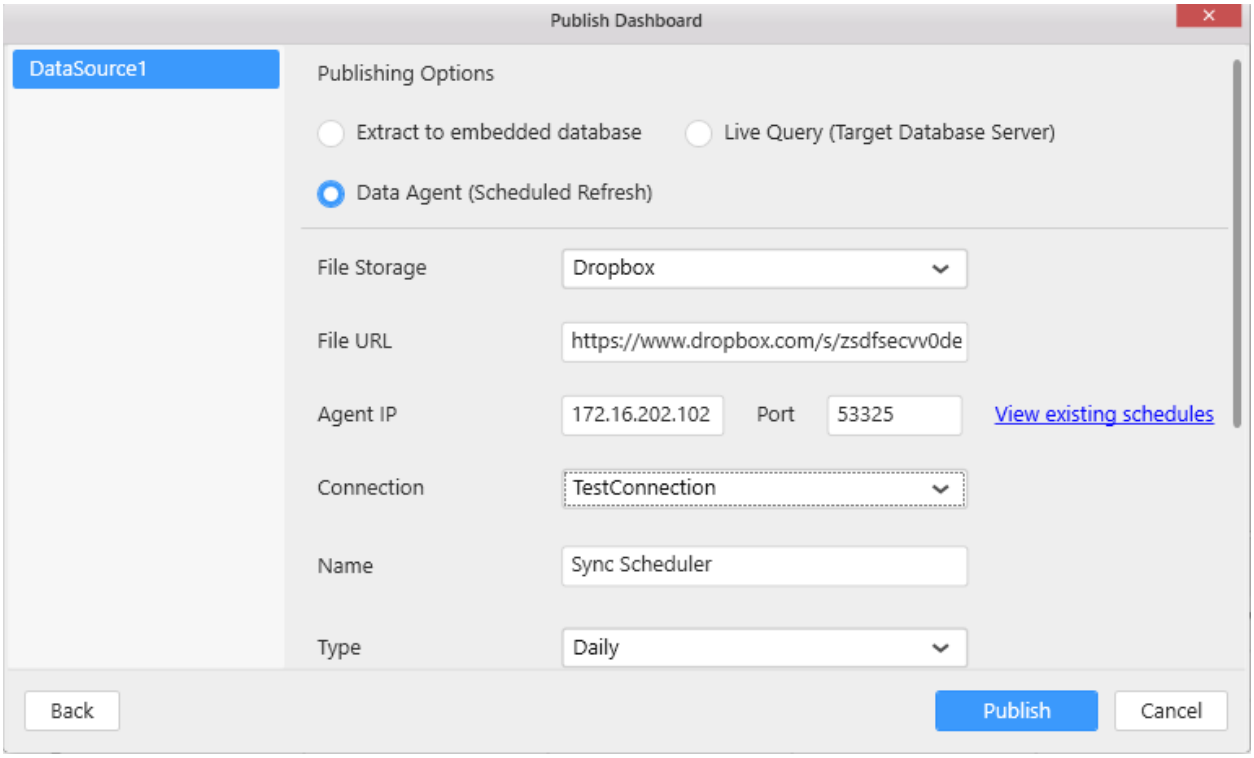
Extracting Data from Web Accessible Resources through Agent

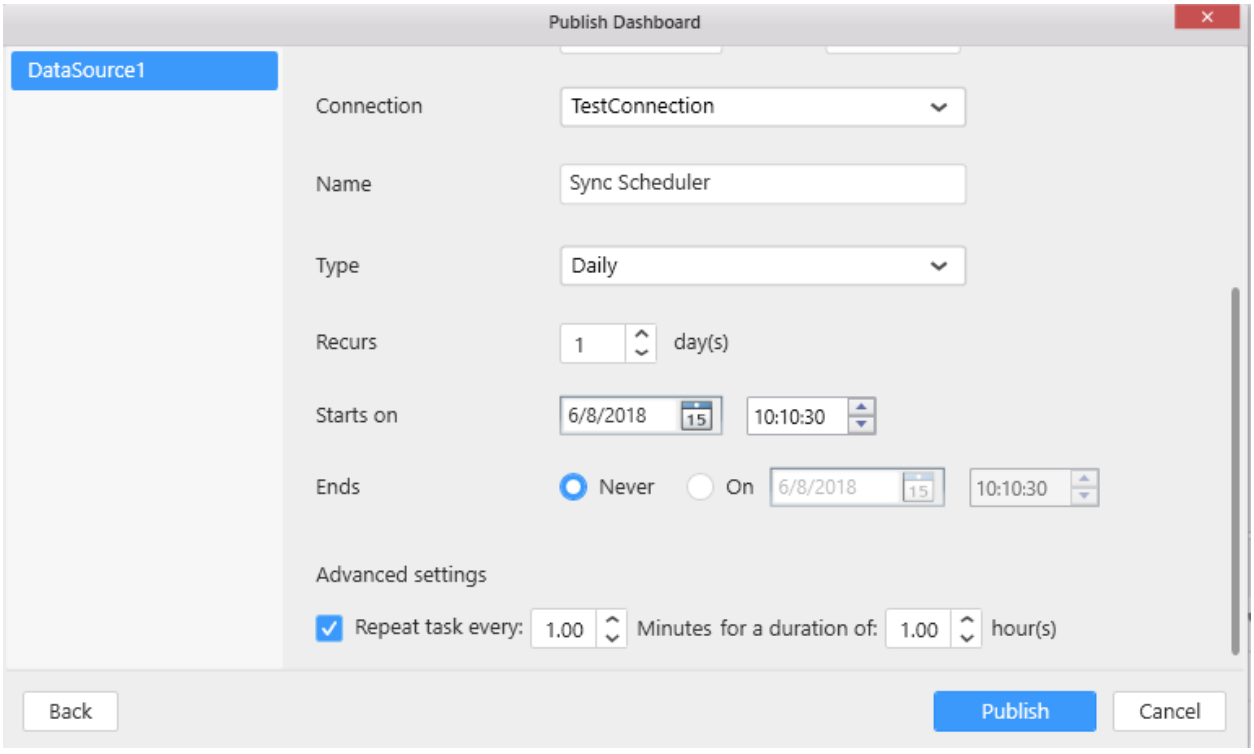
After Configuring the Syncfusion Dashboard Data Agent in data server we can publish the reports which was created using Salesforce Objects, RESTful Web Services, Microsoft Azure Table Storage, Microsoft Excel, CSV, Text Document, JSON, SQLite, and Microsoft Access in the data server by following way.

Once you have completed the dashboard click on Server from menu and Publish and then Dashboard.



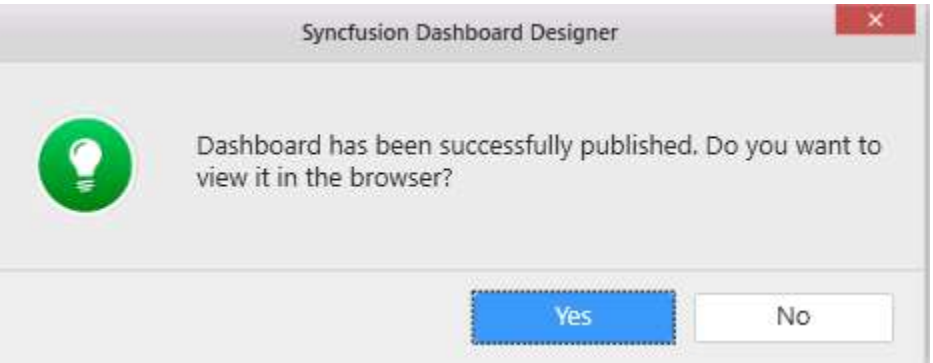
In Publish Dashboard window enter the IP of the data server where the agent was installed and enter the port number (Where the agent was running) in Port text box. Once you entered the Agent details when you click on the connection combo box it will show the list of connections configured in Agent. Enter the name of the scheduler and configure the scheduler by the list of options available in Data Agent window like daily update, weekly update, monthly update, when to start and end the scheduler etc...





Note: Here the File Storage and File URL options are available only for Microsoft Excel, CSV, Text Document, JSON, SQLite and Microsoft Access data sources. If you need to configure the schedule refresh for these type of data sources, you must upload the file into Google Drive or Dropbox or Dashboard Server cloud service.

Click on Publish button once you have configured the scheduler.



Defer Update

In Dashboard Designer application, while configuring and formatting widgets, the query is generated and executed every time to refresh the widget with the changes.

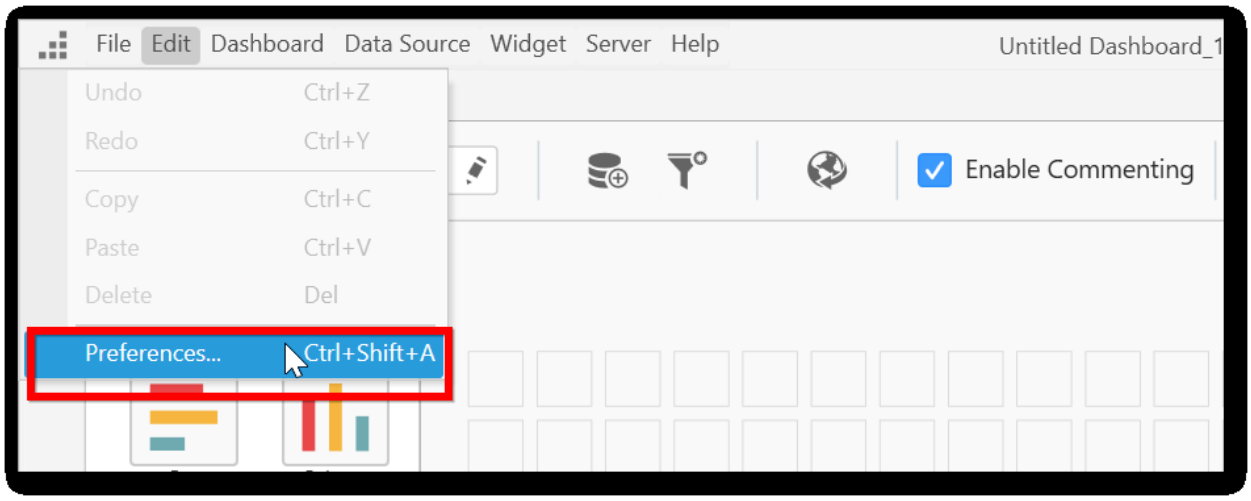
For example, if you drop the Order ID field in the widget, change the summary type from Sum to Count, and apply the measure formatting. Now, three queries will be generated and executed in your server for dropping the Order ID field, changing the summary type, and applying the measure formatting. The widget will also be refreshed three times.

If you enable the **defer update**, you can avoid the multiple calls to the database server and you can manually update the widget with single call to the database server. It also helps you reduce the overall dashboard designing time.

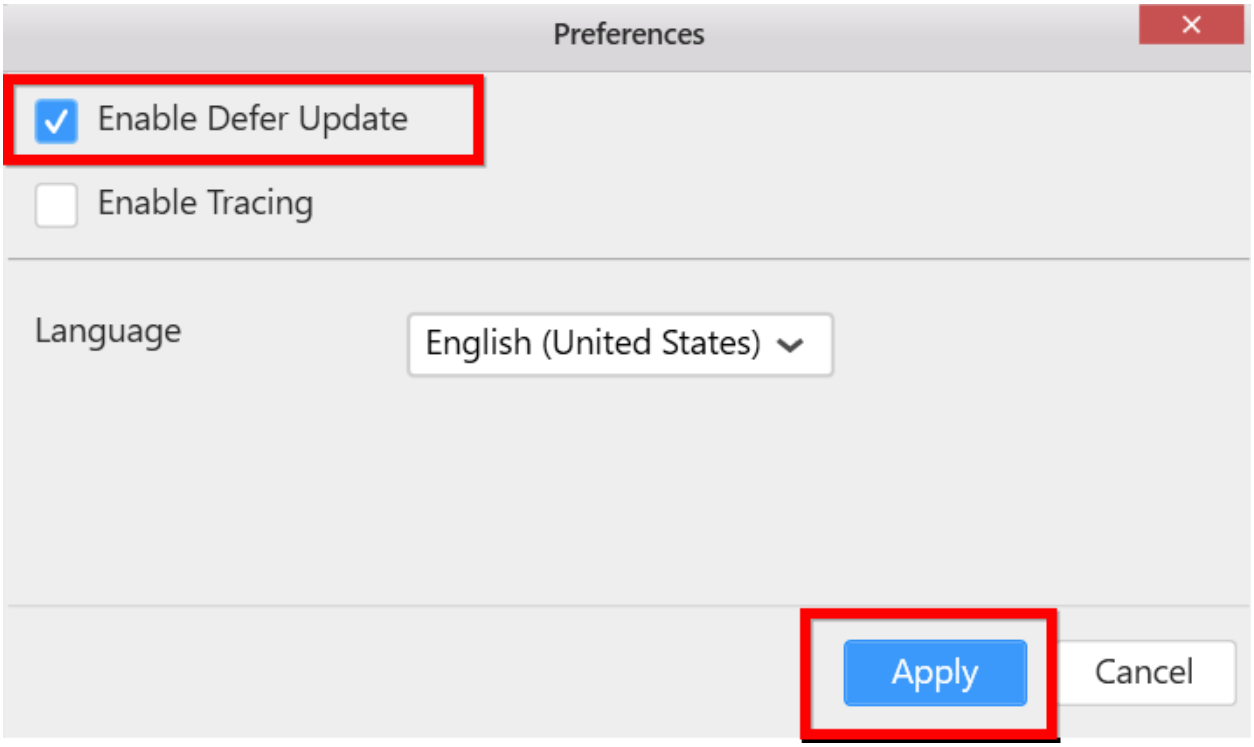
How to enable/disable the defer update

By default, the defer update option is disabled in the Dashboard Designer application. You can turn on/off the defer update using the Preferences option provided in the Edit menu.

1. To enable the defer update, click Preferences.... The Preferences window opens.



2. Select the Enable Defer Update check box. Clearing this check box disables the defer update and makes the designing of dashboard widgets normal (i.e., widget will be refreshed for every change).



Note: The defer update state will be preserved while opening the Dashboard Designer application next time.

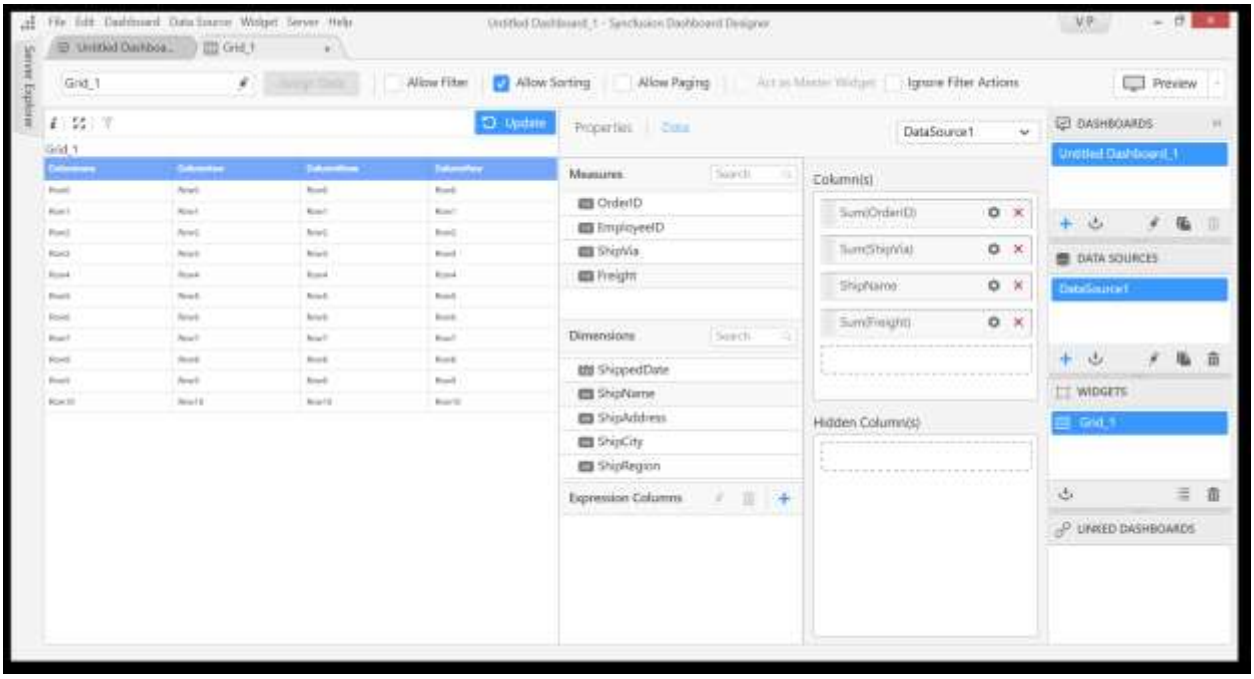
Information: You can disable/enable the defer update anytime during the dashboard report design.

Configuring widgets with defer update

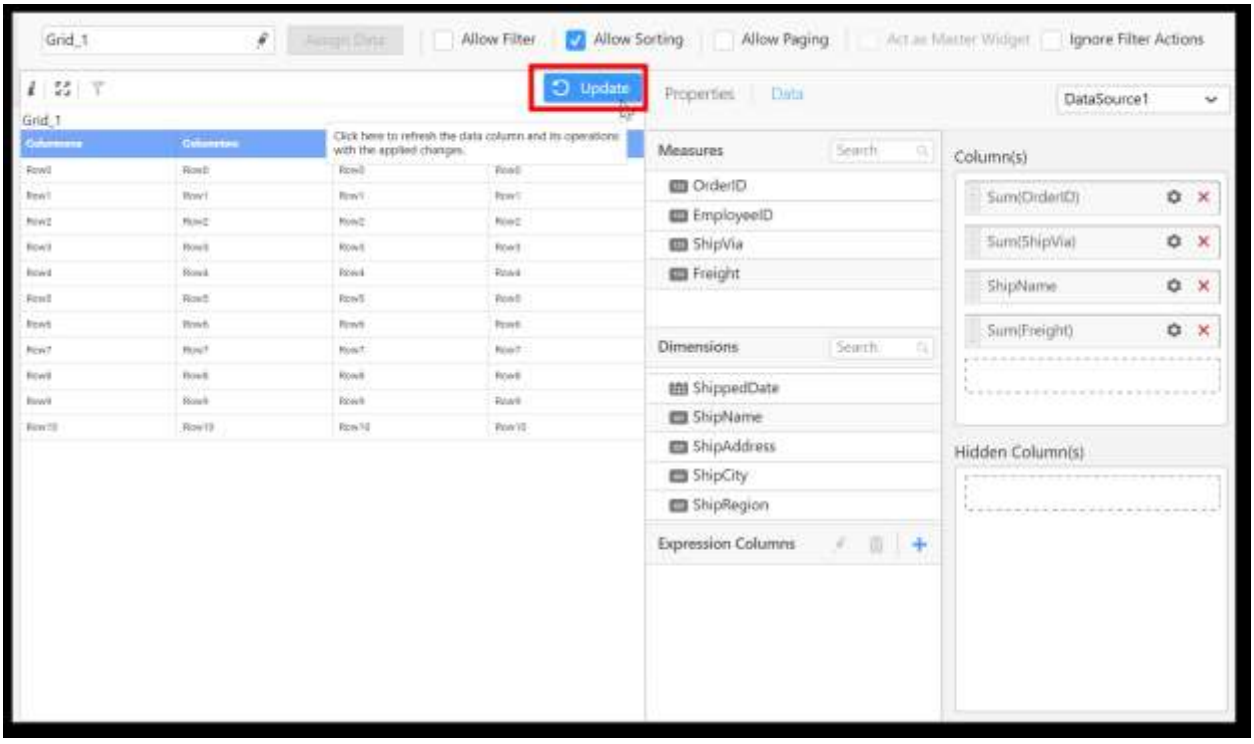
While configuring the dashboard widgets/filter panel, the widget should be manually updated by users. The update button is provided in the control preview area in the widgets tab.

Click Update to fetch the data and refresh the widgets.

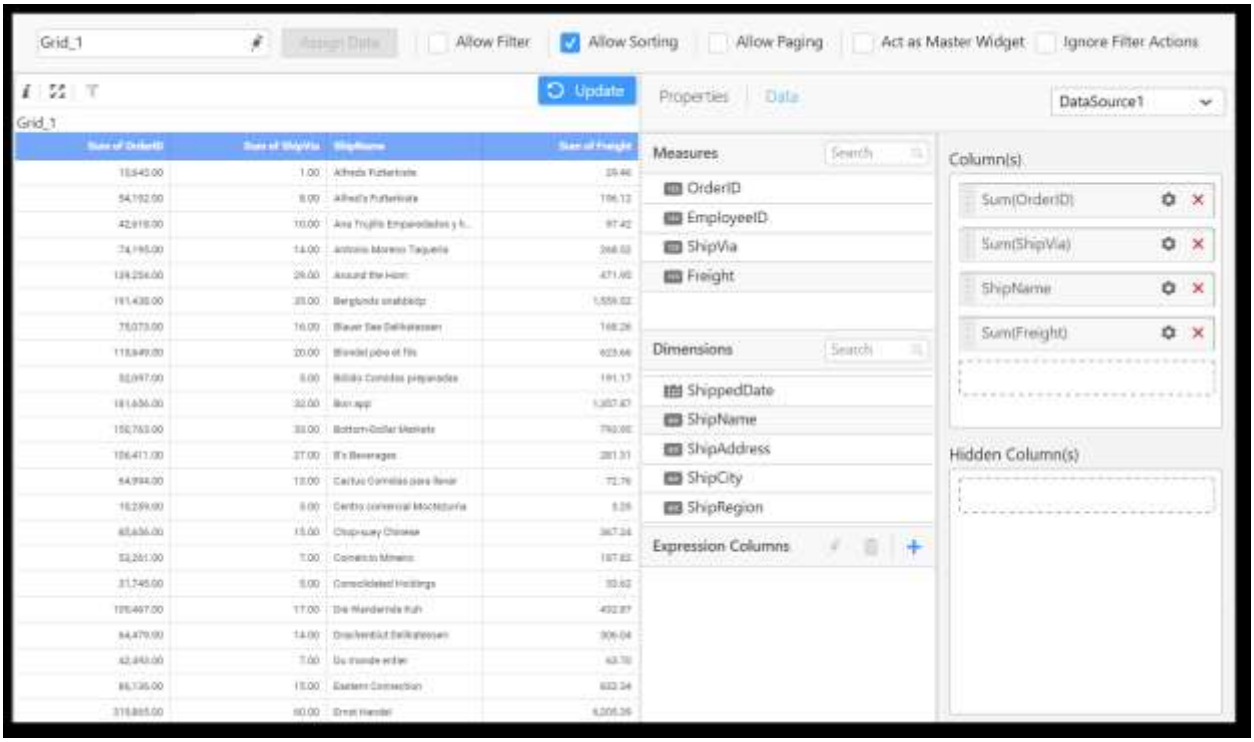
The following screenshot shows configuring the widget with defer update in enabled state.



The following screenshot illustrates manually refreshing the widget using the update button.



The following screenshot shows the refreshed view of the widget.



Note: You can also use the keyboard shortcut Ctrl+ L to refresh the widget.

Automatic widget refresh criteria

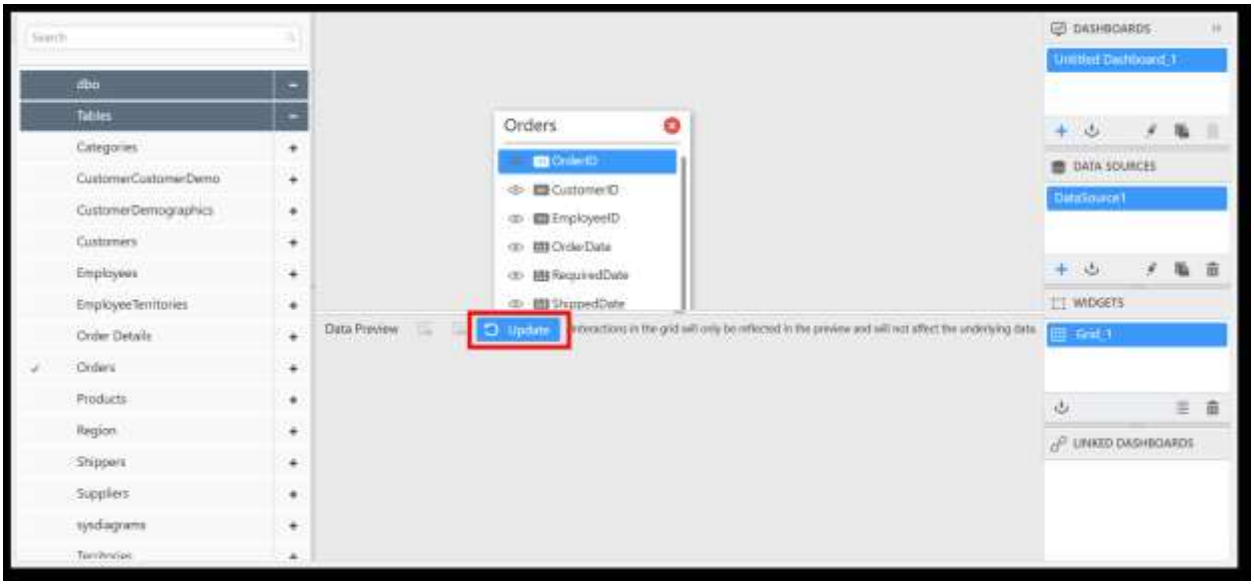
Data fetching is mandatory in the following cases, so the widget will be automatically refreshed, if it is not updated manually.

- Switching the internal tabs in the widgets/filter panel tab.
- Closing the widgets/filter panel.
- Previewing the dashboard.
- Turn off the defer update from enabled state.

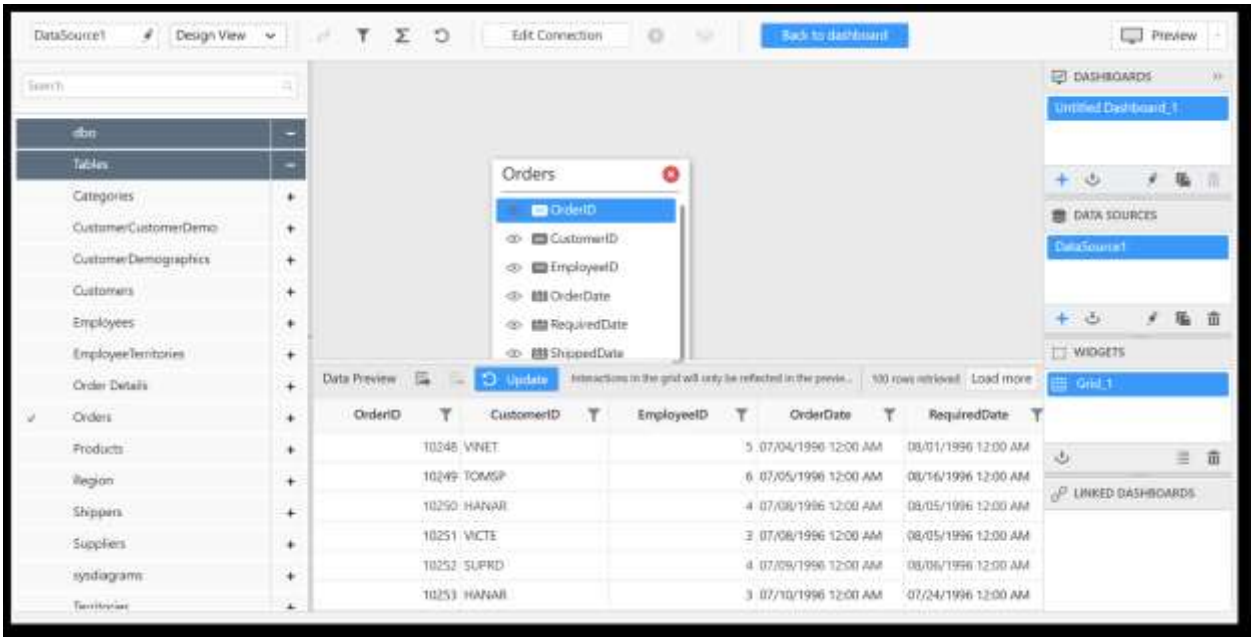
Information: You should update the widgets manually using the update button before switching the tabs or saving the report to ensure the changes applied properly and to avoid any unexpected errors in automatic update.

Defer update in preview data grid

The data preview grid in the data source tab fetches and displays the raw data by clicking the Update button.



After fetching the data, it will show the raw data as shown in the following screenshot.



While performing any changes in the data source like applying initial filter, creating expression, changing the data type, etc., the data in the preview grid will not be automatically updated. To preview the updated data for the applied changes, click Update in the preview grid.

Dashboard connection string switcher utility

The **dashboard connection string switcher utility** is a UI tool that helps you to change the connection string details in one or more dashboard (*.Sydx) files at a time without using the Dashboard Designer application.

Generally, the dashboards are prepared with local server database details, and you may have to update the connection details to a production database server before publishing it to our Dashboard Server application.

System requirements

The system requirements to run this utility is same as Dashboard Designer application. So, refer to the details in the following link.

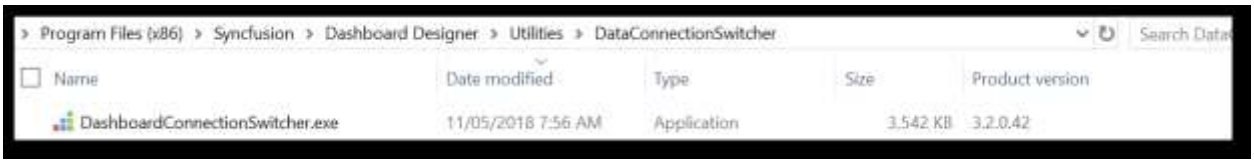
https://help.syncfusion.com/dashboard-platform/dashboard-designer/system-requirements

How to run the utility

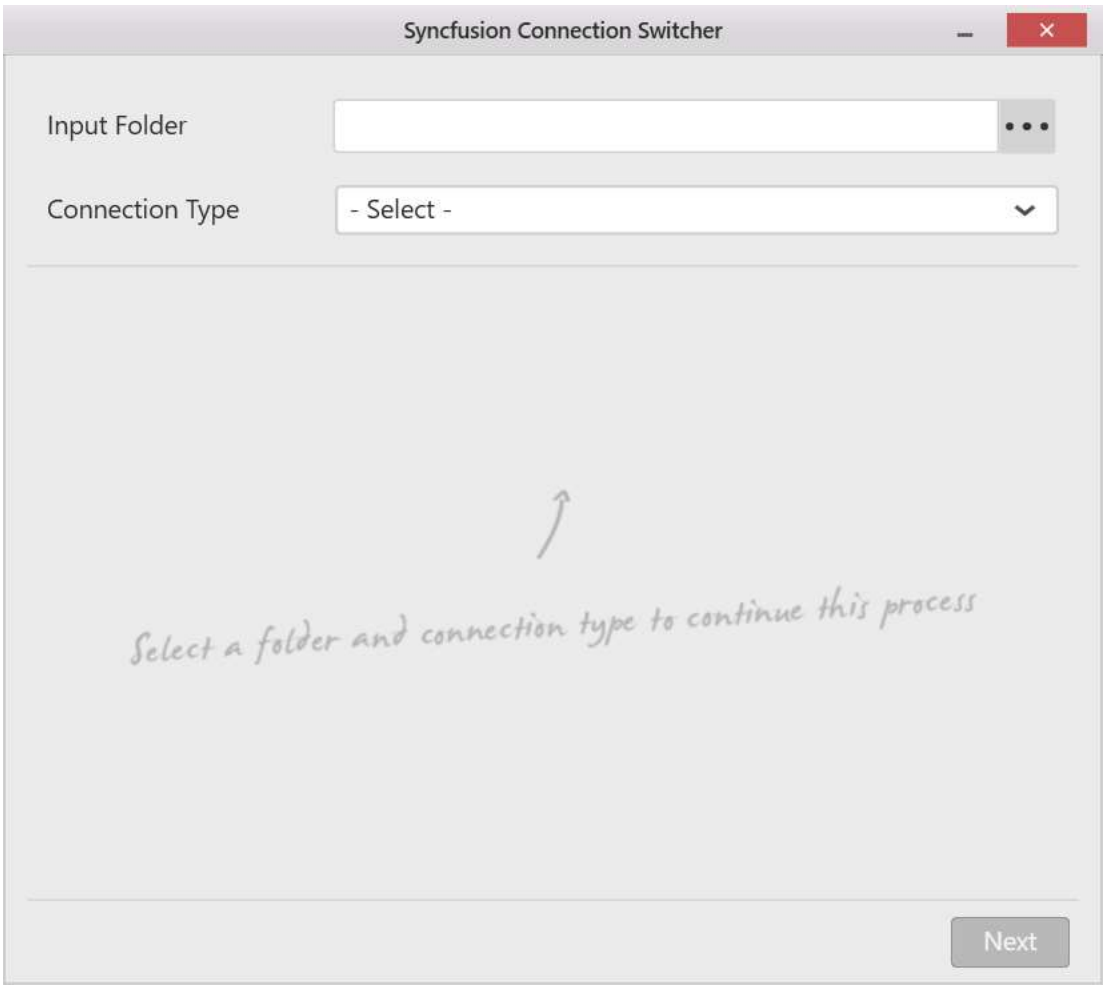
1. Open the location where the utility is shipped. The utility is shipped in the following locations of our dashboard platform builds.

Build	Location
Dashboard Designer	C:\Program Files (x86)\Syncfusion\Dashboard Designer\Utilities\DataConnectionSwitcher
Dashboard Server	C:\Program Files (x86)\Syncfusion\Dashboard Server\Utilities\DataConnectionSwitcher
Dashboard Designer	C:\Program Files (x86)\Syncfusion\Dashboard Platform SDK\Utilities\ DataConnectionSwitcher

2. Select the DashboardConnectionSwitcher.exe file and run the utility by double-clicking it.

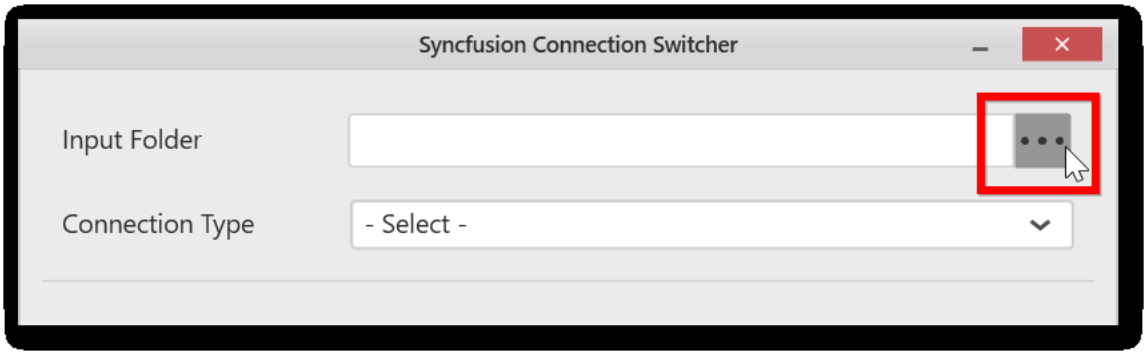


The application will be launched with the following screen.

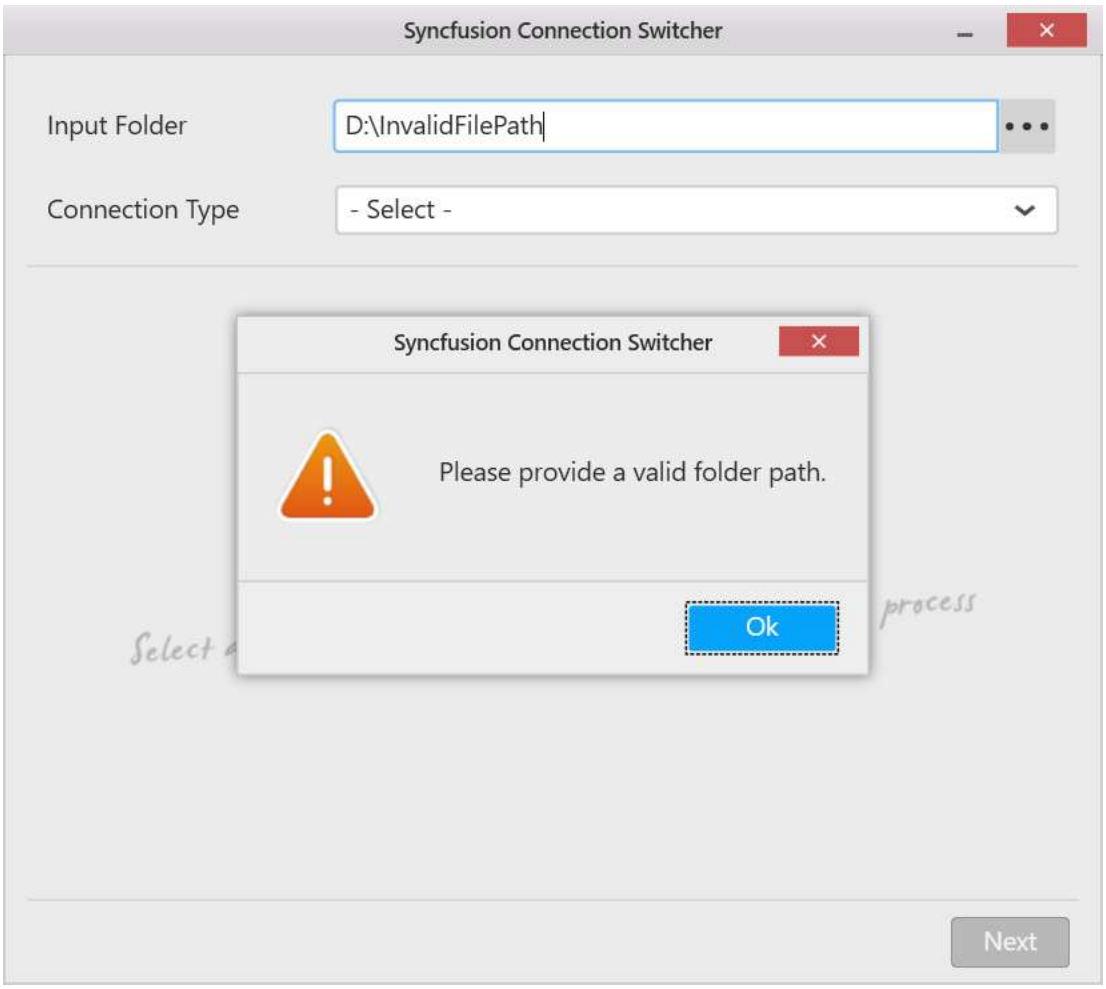


How to select the connection type and input data sources

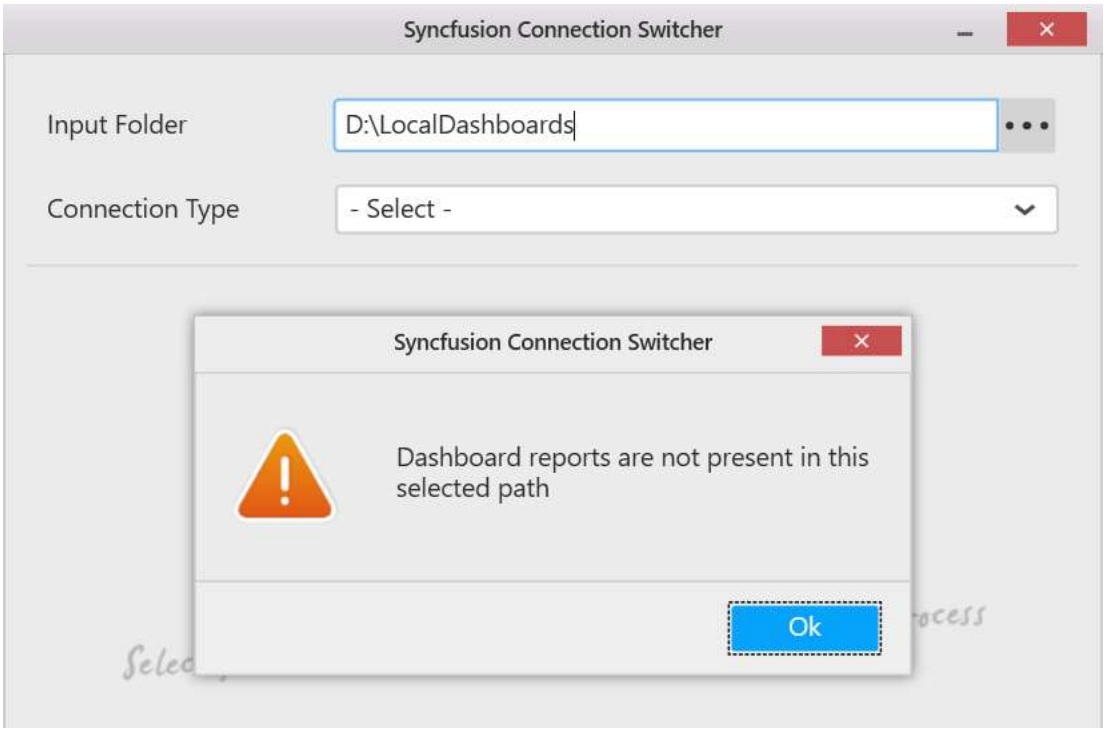
1. Place all the dashboard files in which you need to change the connection string in an individual folder.
2. Browse the file folder or paste the folder path as an input.



Note: If the file path is invalid, you will get the following alert message.

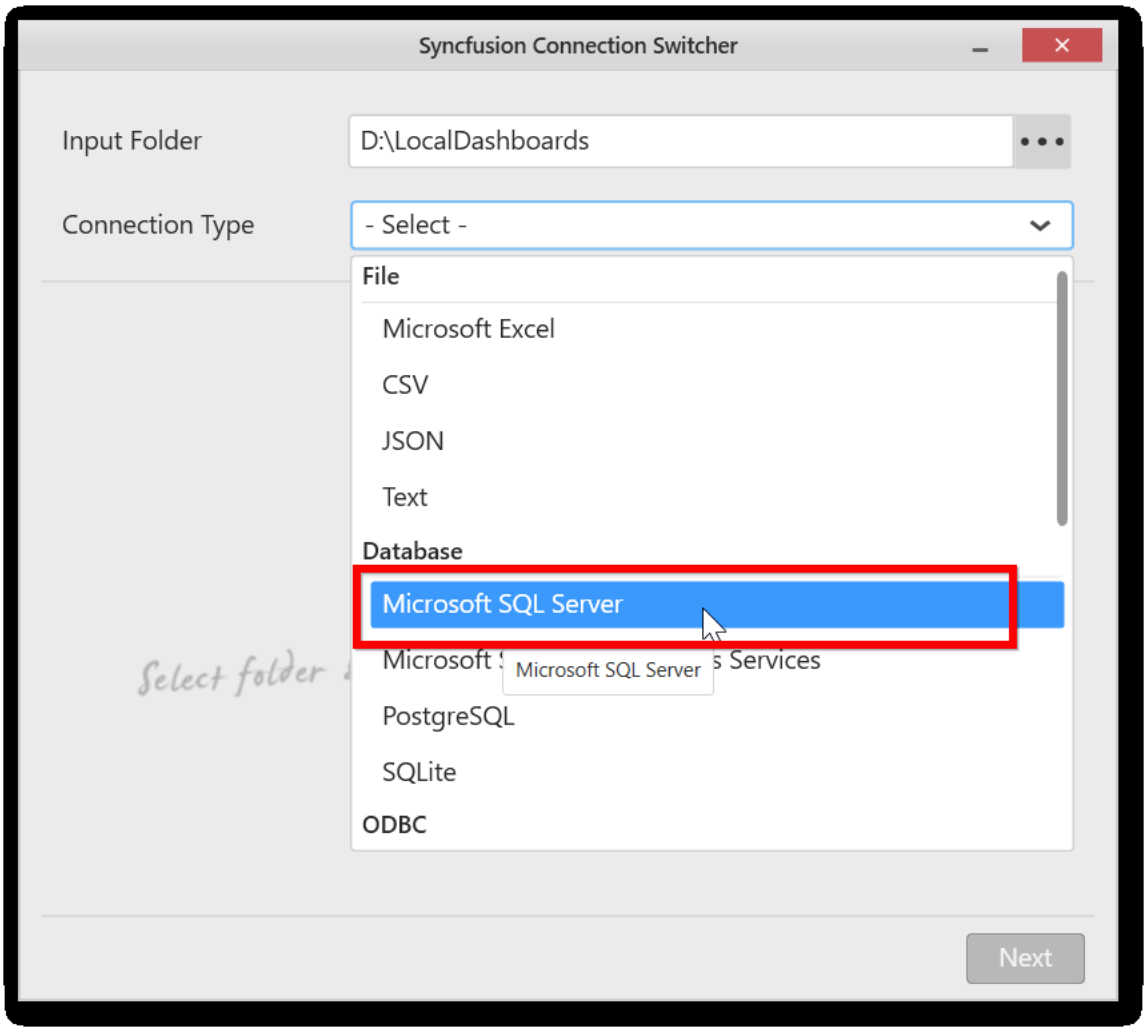


Note: If no valid .Sydx files are present in the folder, the following alert message will be displayed.

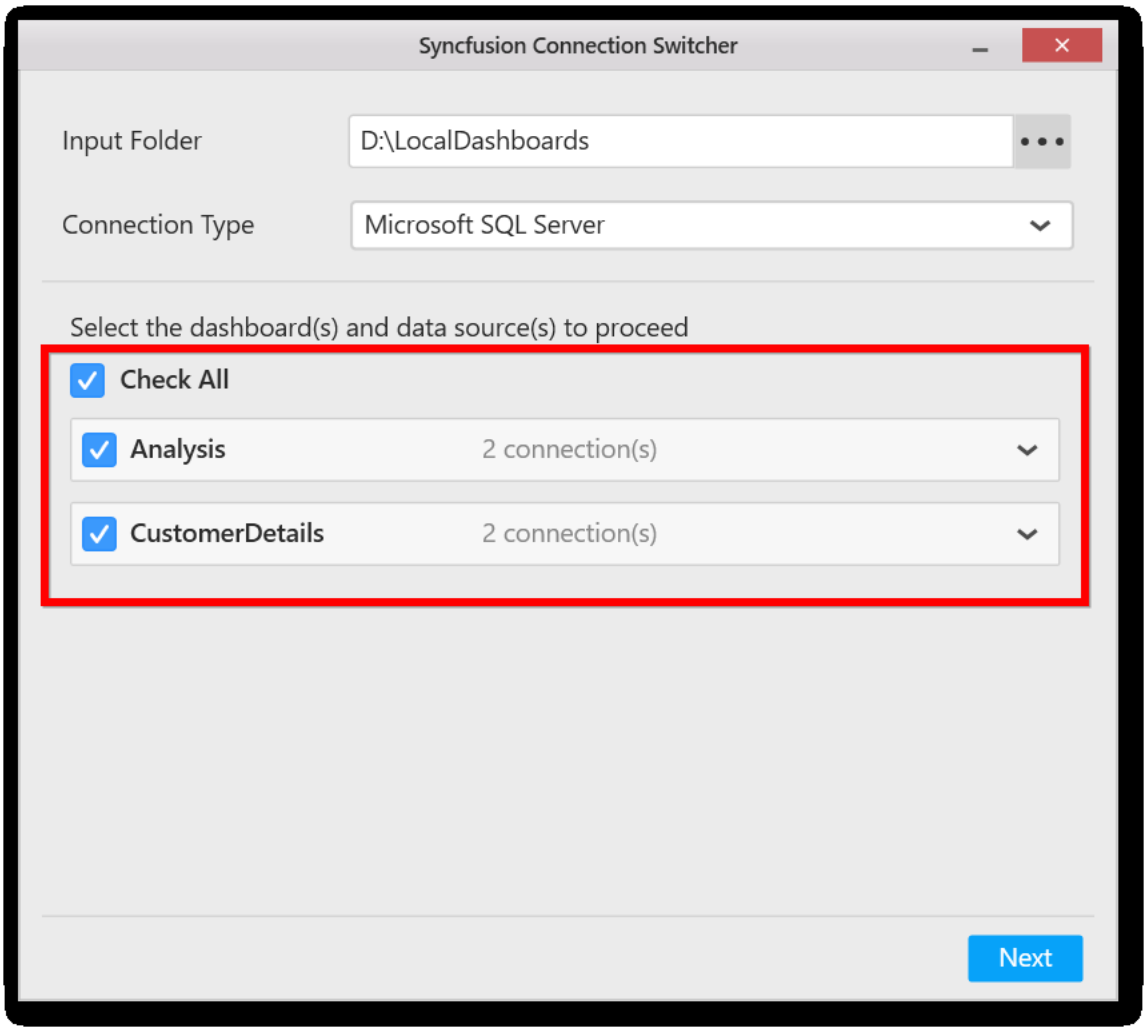


Information: Limitation: Only dashboards created with **version 3.2 or more** can be used as the input to this tool.

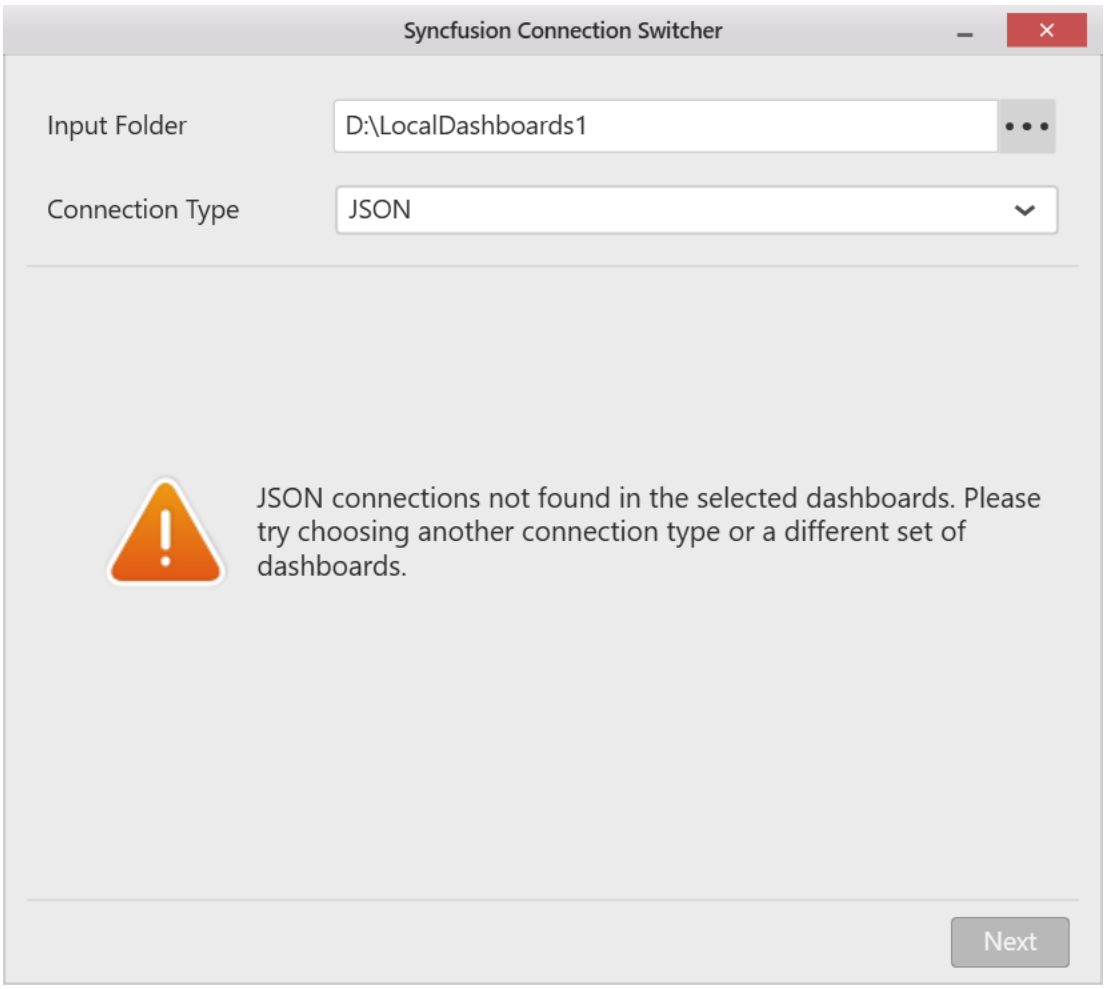
3. Select the connection type using the Connection Type drop-down option.



It will display the data sources configured with the selected connection type. By default, all the dashboards and data sources configured with the selected connection type will be displayed and checked.



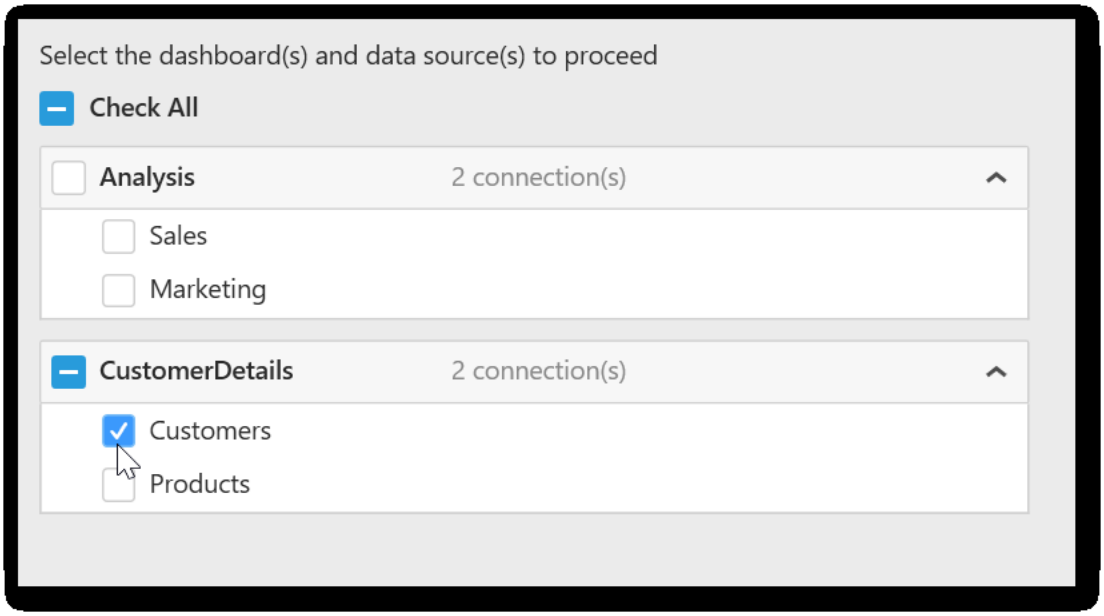
Note: If there is no data source matched with the selected connection type, the following alert message will be displayed.



4. Click the dashboard node to see the available data sources.

The list of data sources will be shown and selected by default.

5. You can select the required data source by checking the checkbox near the data source name; the unchecked data sources will not be processed and will remain same till next time you open the dashboard.

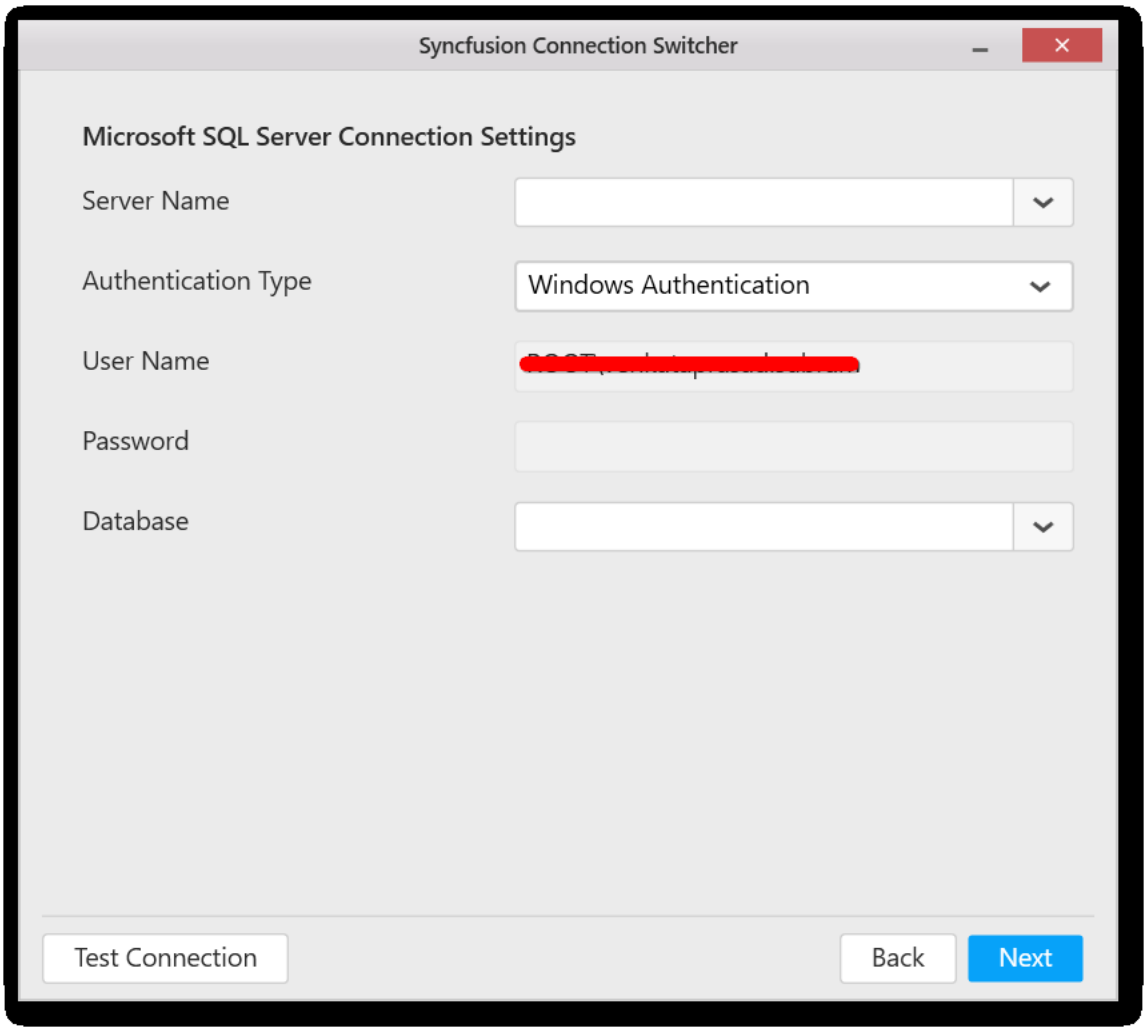


5. Click Next to proceed.

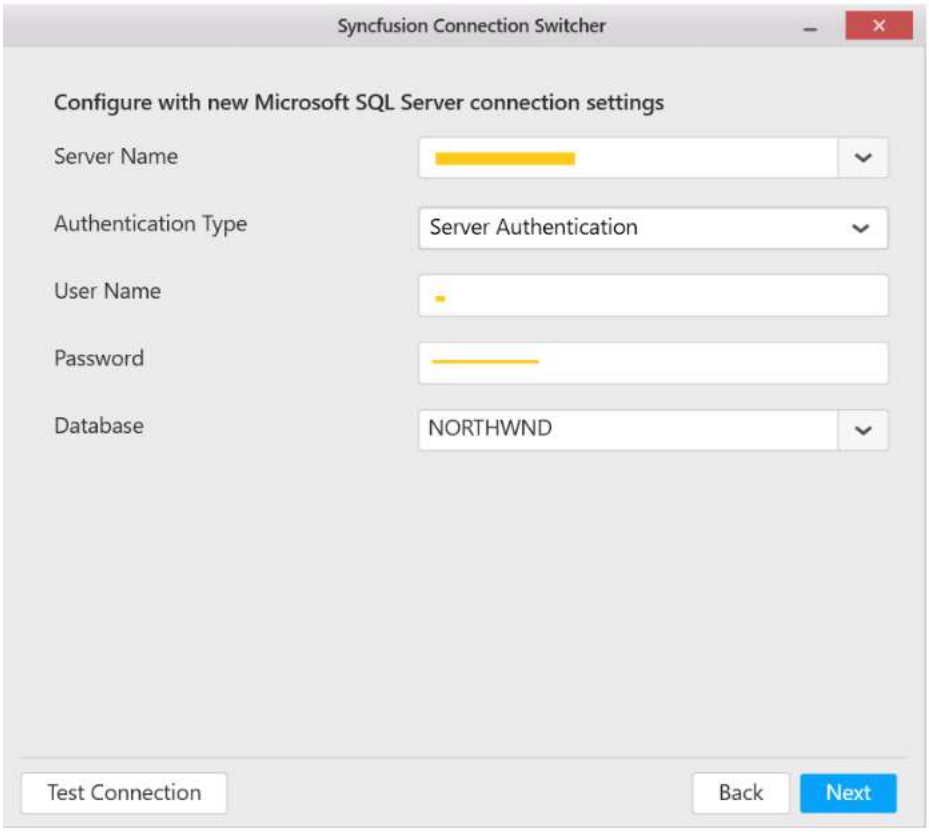
How to enter the new connection details

After clicking the Next button, the dialog will display options to choose or enter a new connection information of the same selected data source type.

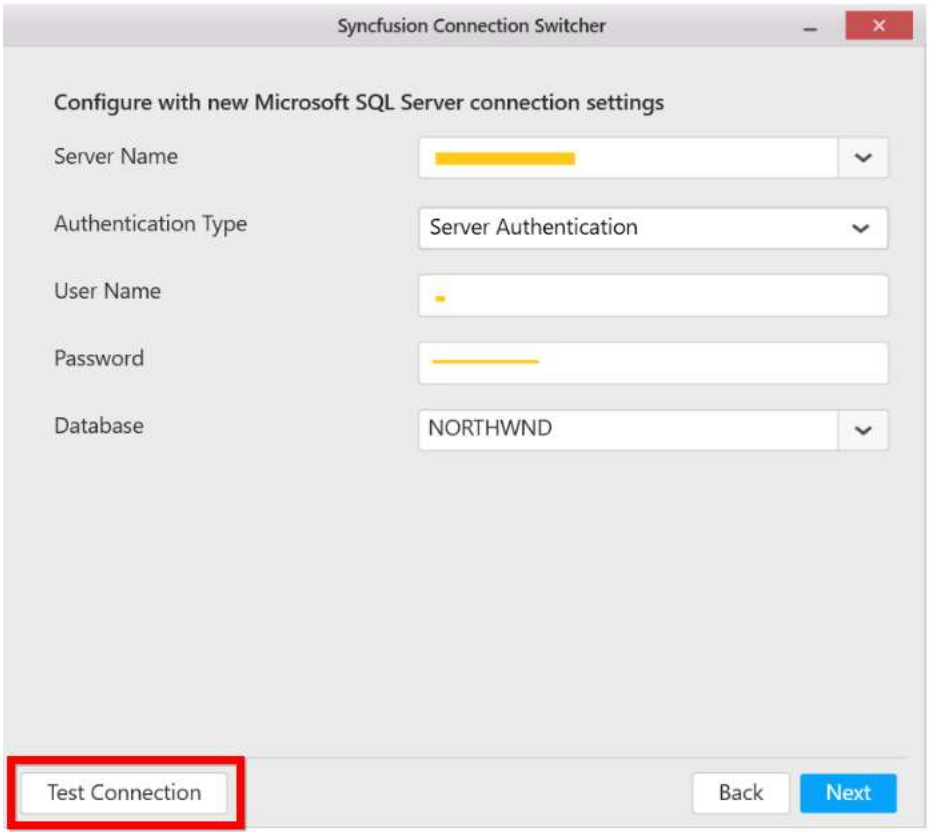
Information: Limitation: You can only connect to a different server of the same data source type and it is not possible to change the connection string of two or more different data source types in a single conversion.



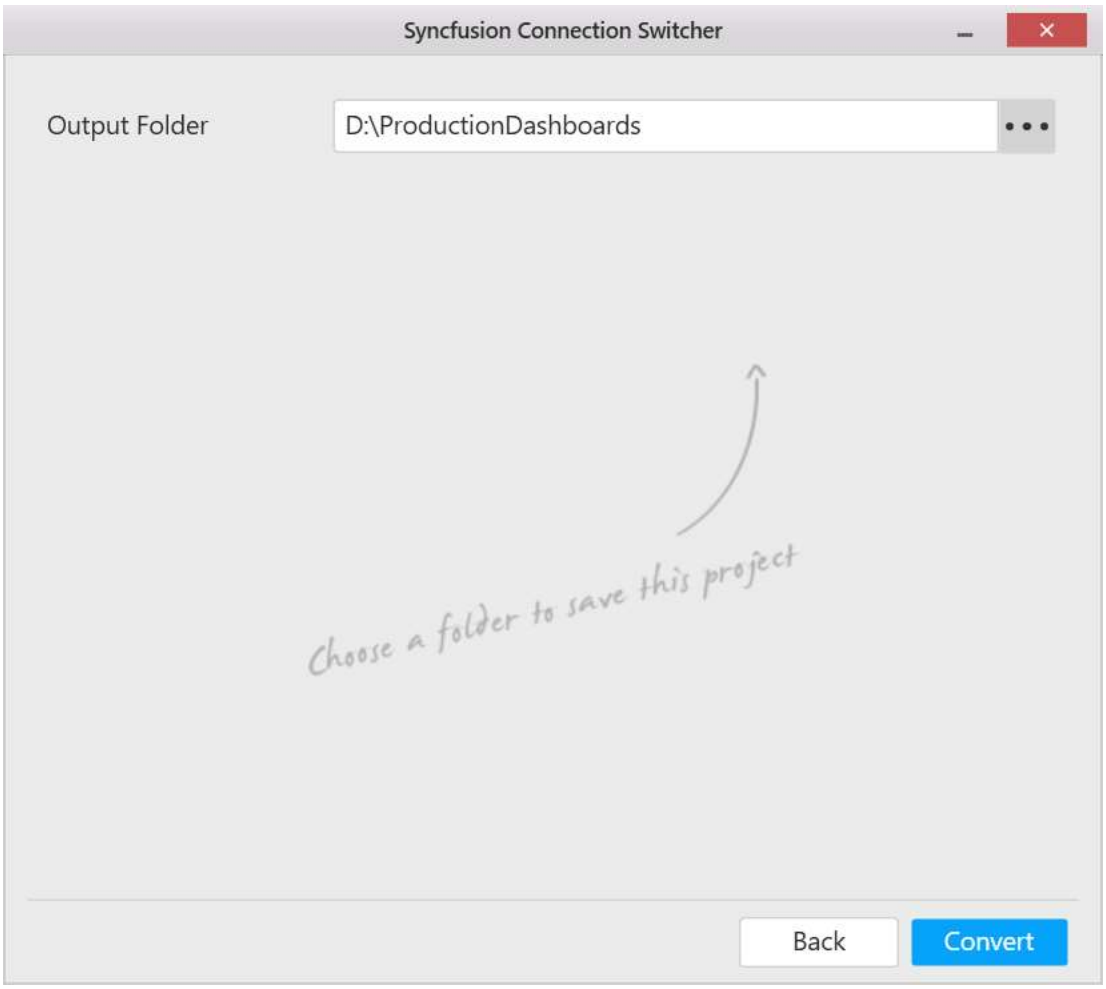
1. Enter new server credentials.



2. The **new server must be accessible** to proceed the data source information in the files, so make sure that the connection is accessible using the Test Connection button.



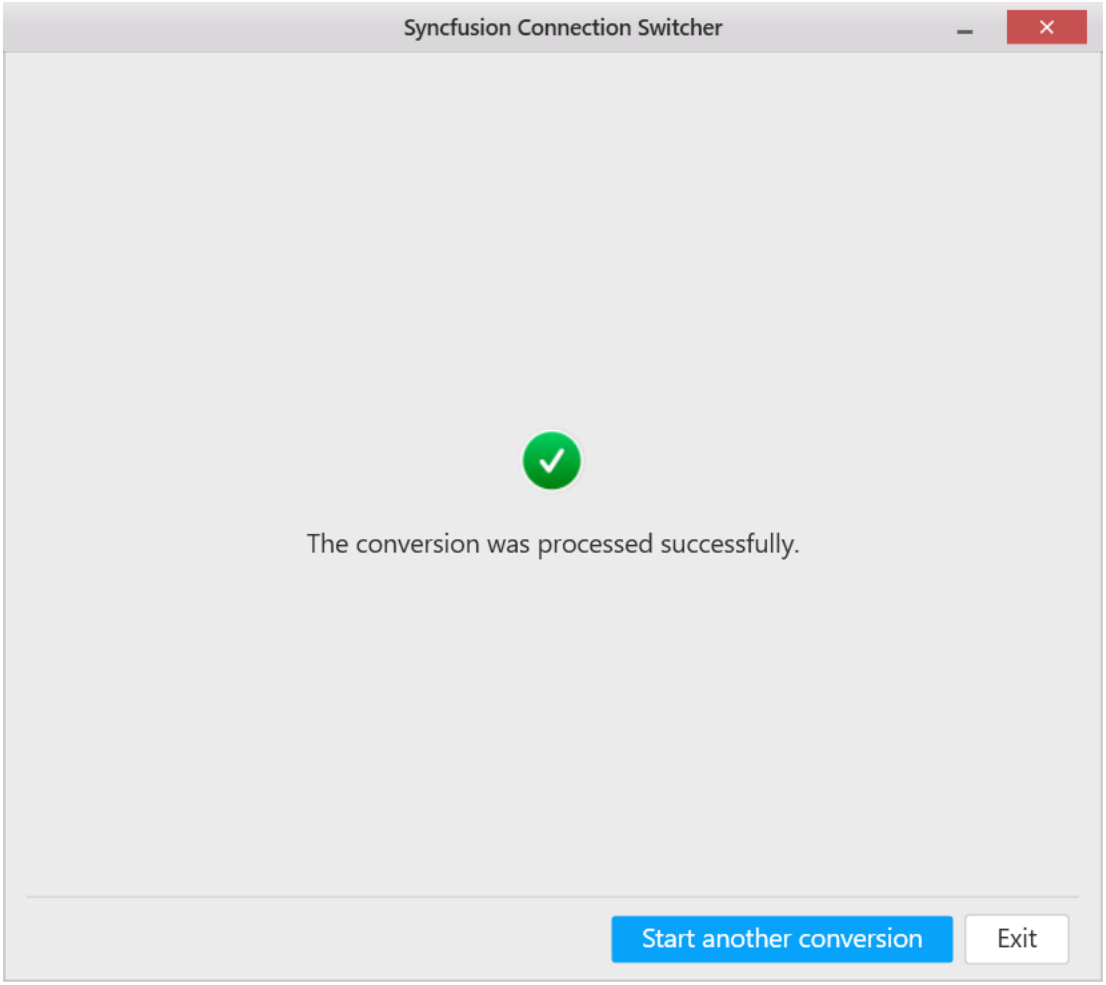
3. Click Next to proceed further. 4. Enter the output location where you need to save the modified dashboard.



Note: Make sure that the output folder exists in your local drive and have permission to save files. In case, the output folder contains the dashboards with same files, the existing files will be replaced with processed files.

Information: The input and output folder paths can be same, but it is strongly suggested to save the folders at different locations from the original dashboards location.

After the files are successfully done, you will get the following information.



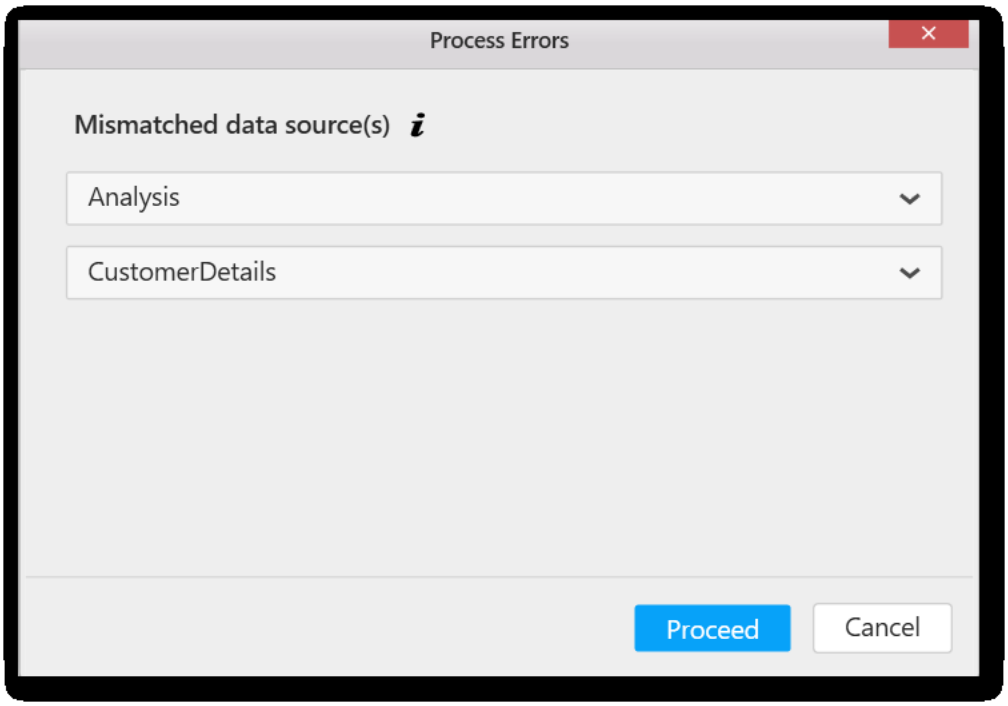
5. You can initiate another connection string changing process by clicking the Start another conversion.

How to handle the schema mismatch

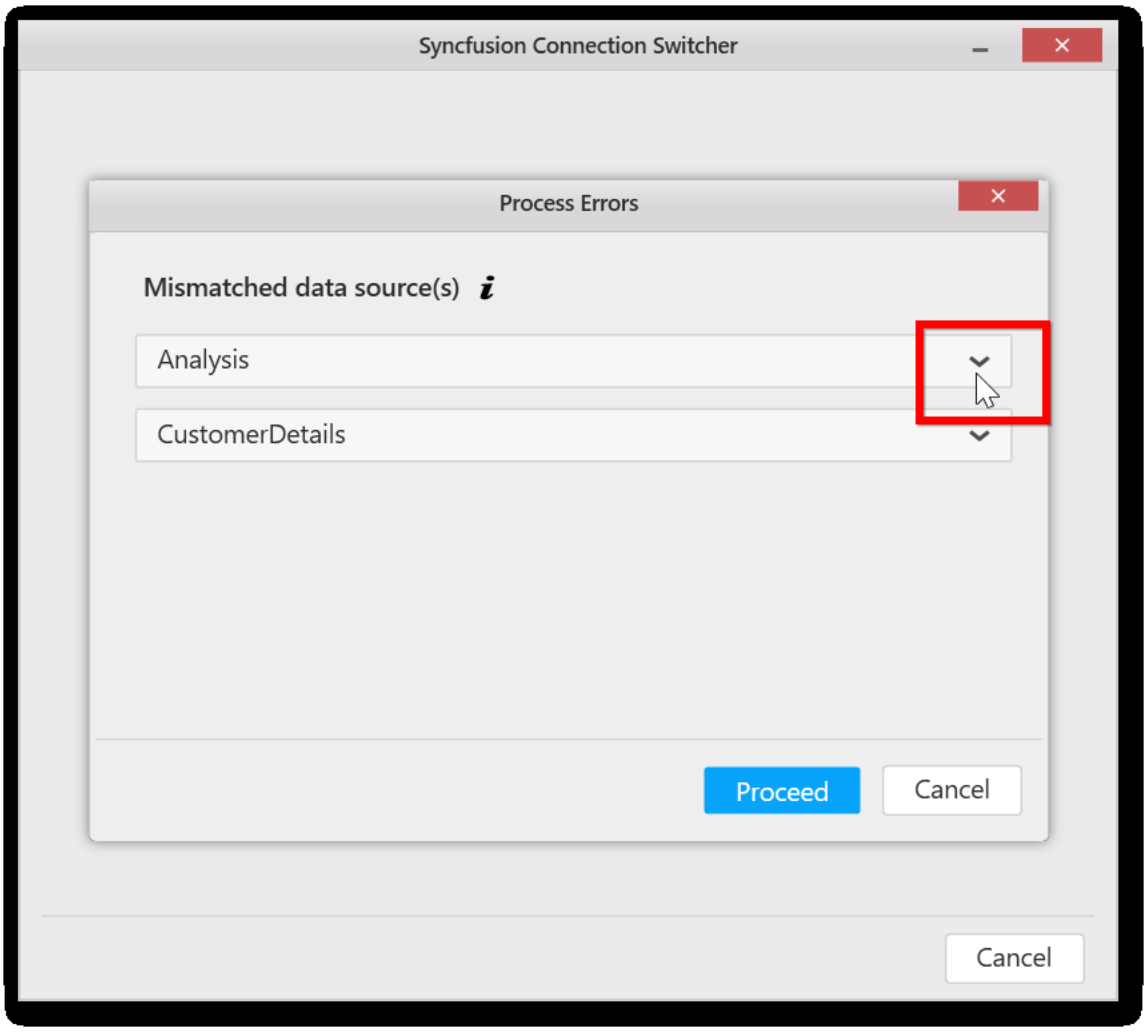
Note: Refer to the following Knowledge base article to learn the limitations for retaining schema information.

https://www.syncfusion.com/kb/8663/edit-data-source-connection-and-its-limitations-for-retained-schema

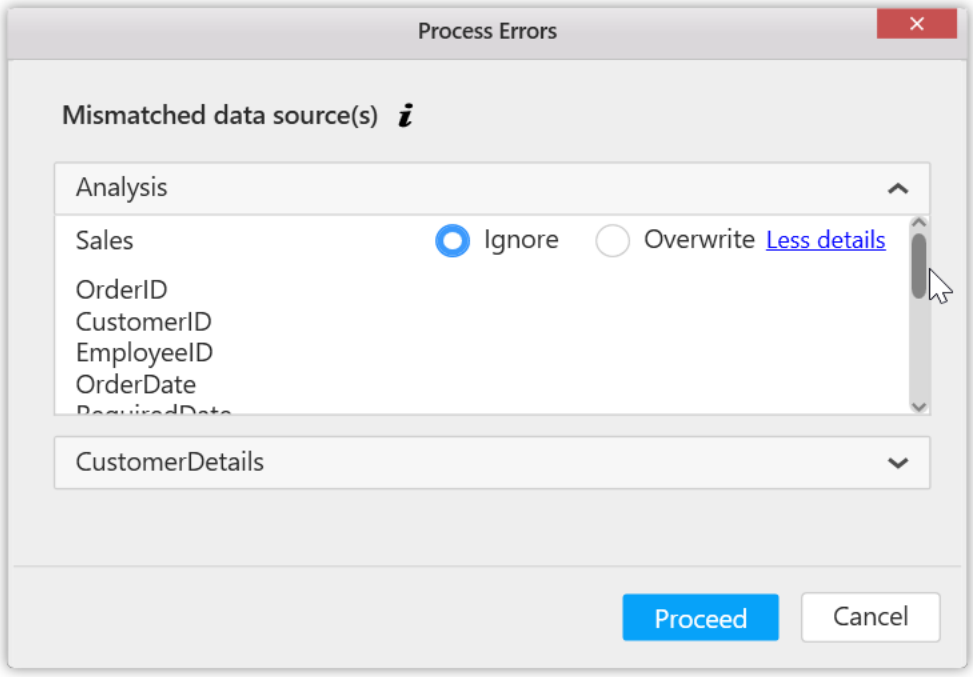
1. Click Convert to display the details in the process errors dialog, if the new connection string details differ from the existing connection string information.



2. Expand the data source to see the mismatched column details.



3. You can see two options: Ignore and Overwrite with the mismatch column details.



Ignore: It is the default option. If the data source is ignored, the connection string will not be changed, and the existing data source will be used.

Overwrite: If the data source information is overwritten, the new connection string will be used in the output files.

Information: If the **overwrite** option is used or selected, the file will be saved with the entered connection string. But, when you open the modified SYDX file in the Dashboard Designer application next time, the widgets expression will be cleared due to schema mismatch.

Changing the web data source connection information

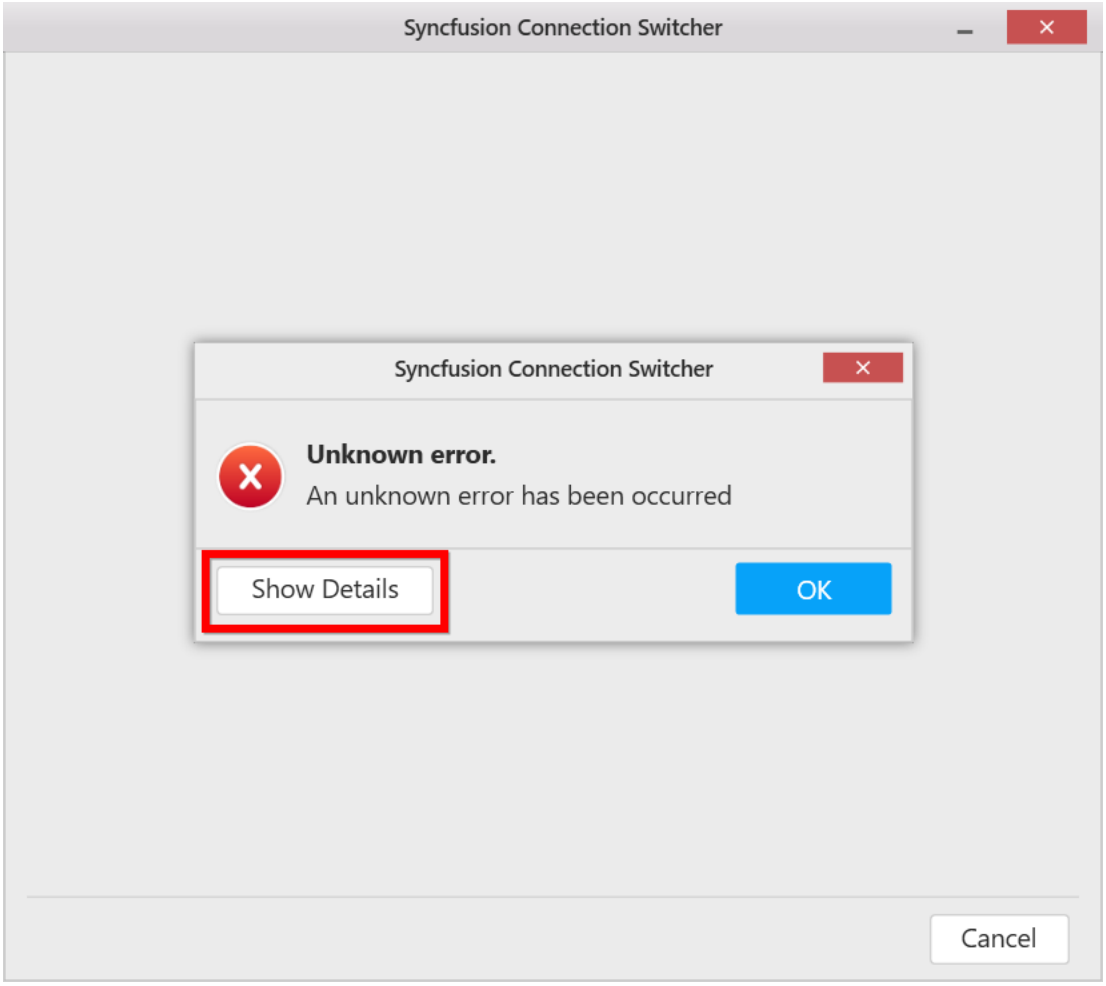
CefSharp and its dependency assemblies are required to change the web data source connection information, so make sure that you have checked the option to ship the CefSharp files while installing the Dashboard designer/Dashboard platform SDK builds.

The Cef assemblies are not shipped with the Syncfusion Dashboard Server build, so you can download the assemblies from this link location and place it in your utility folder location by accepting the following license terms.

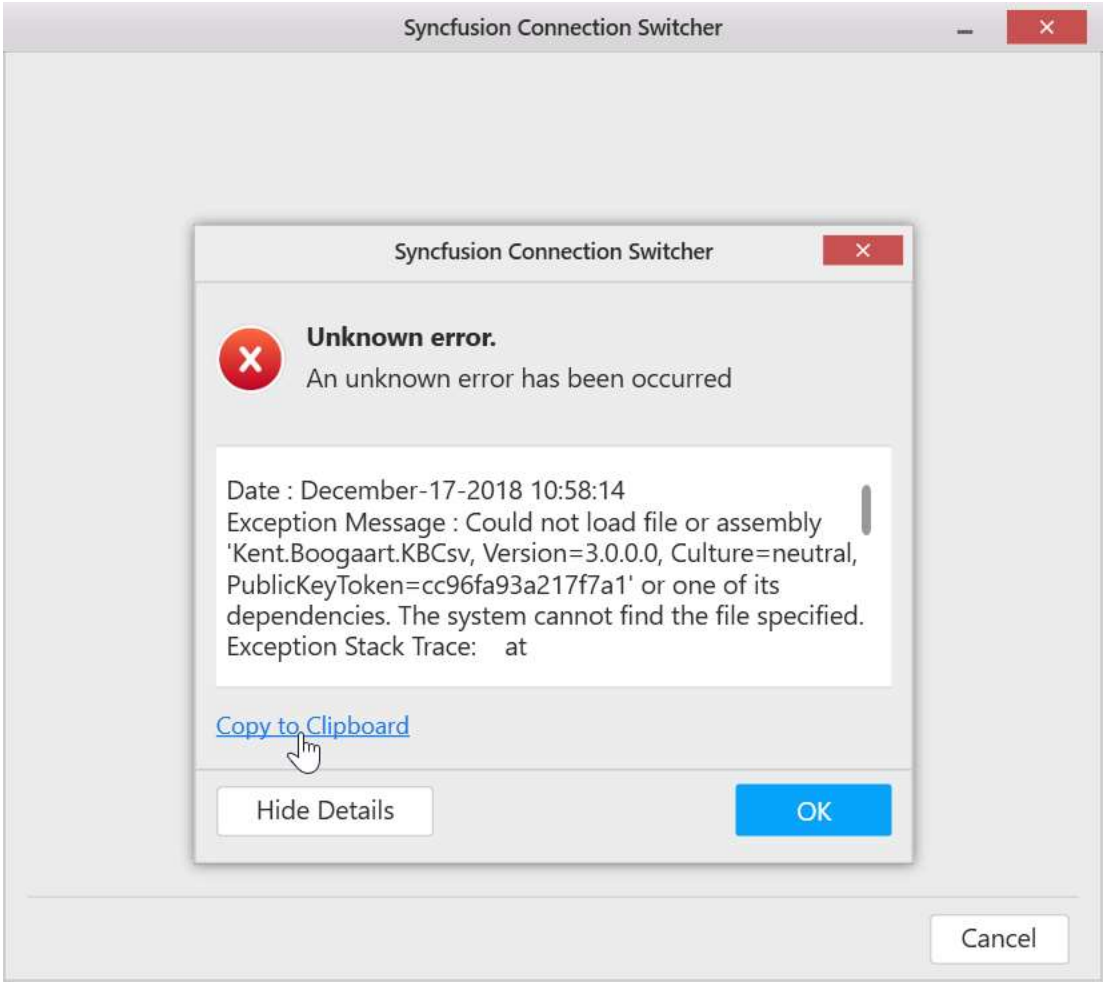
Acting on errors

If any unexpected error occurs, try to copy the exception information and contact our support by creating a new incident and attaching the error information in the incident.

1. Click Show Details to know the occurred error.



2. Click Copy to Clipboard to copy the exception details and upload the same information with our support team with the details to replicate the issue.



Acting on Errors

Error logs

Dashboard Designer error log files

Errors that occur when working with Syncfusion Dashboard Designer are logged under the following location in a text file.

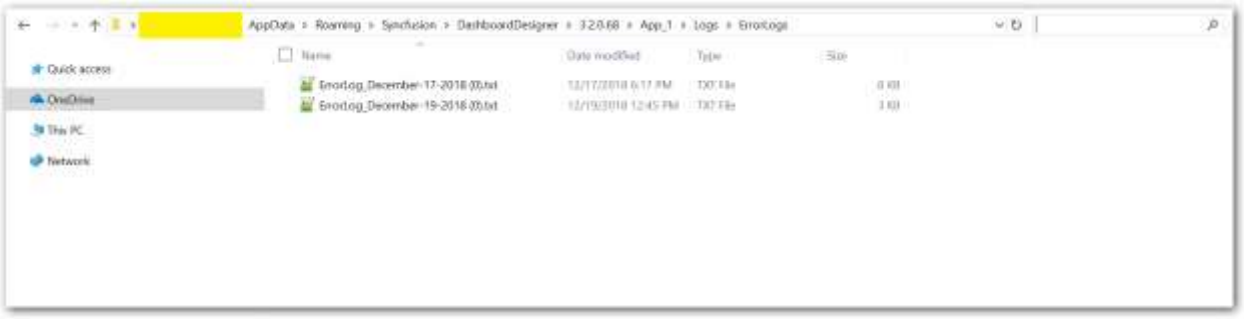
```
\%appdata%\Syncfusion\DashboardDesigner\[Version]\[Application_InstanceID]\Logs\ErrorLogs
```

Dashboard Viewer error log files

Errors that occur when previewing the dashboard through Dashboard Designer are logged under the following location in a text file.

```
\%appdata%\Syncfusion\DashboardDesigner\[Version]\IISExpress_DashboardService\ErrorLog
```

The errors occurred in a day will be logged in the same file one after another. The next day log will be added in a new file as follows.

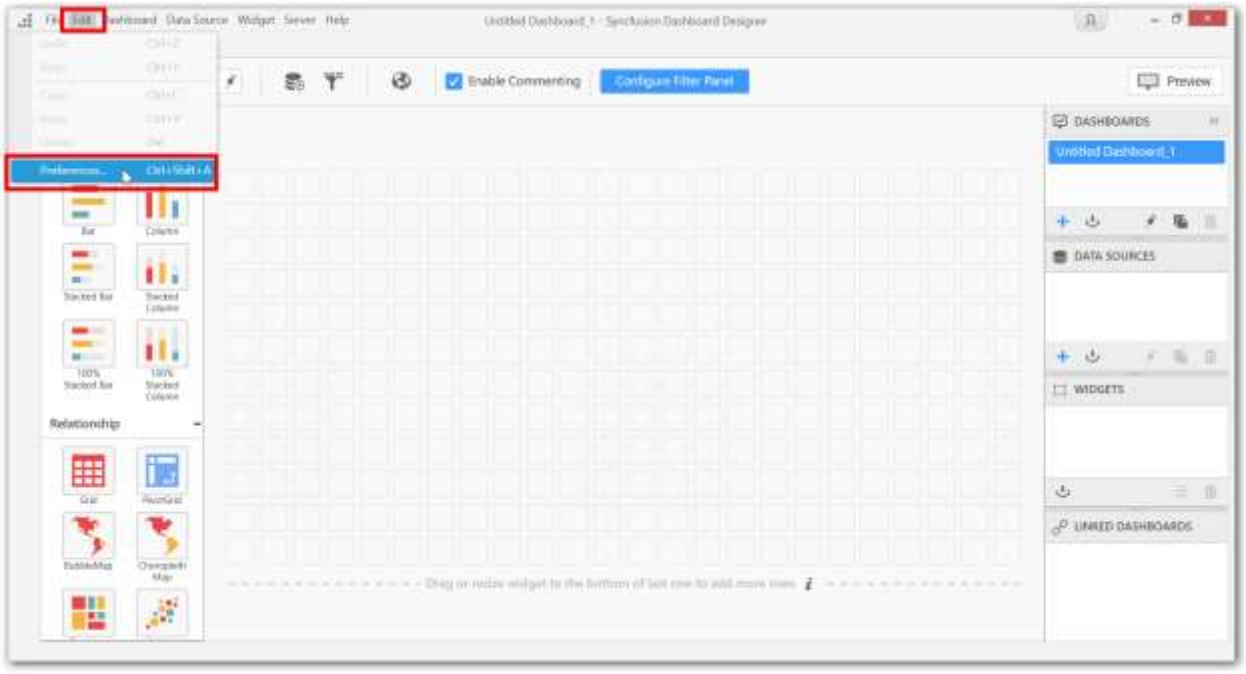


If you face any error while working, collect the log files and share with us by creating a support ticket in Direct-Trac.

Not only the errors from the application are logged here, but also some generic errors like, data connection errors are logged due to network failure.

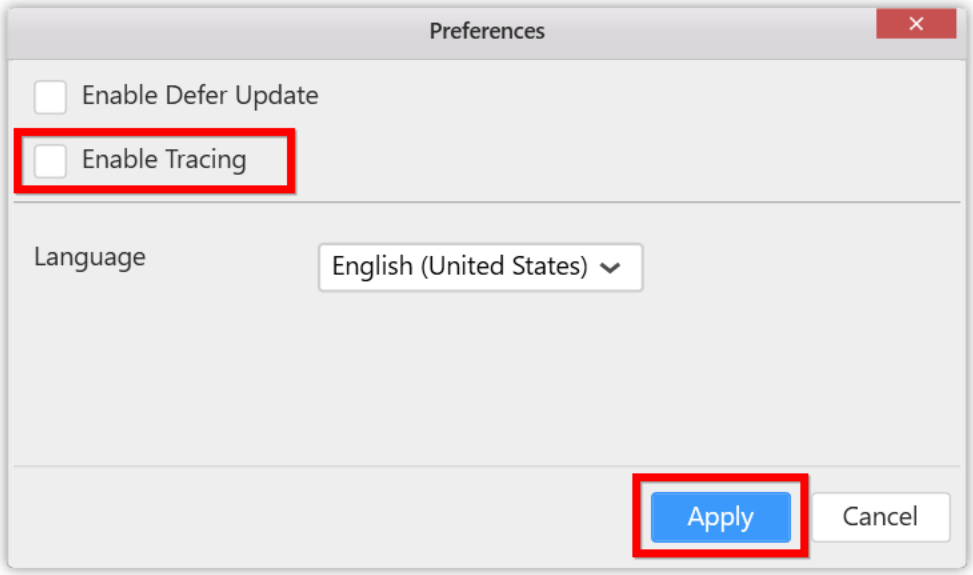
Event logs

The user interactions and some debug information can be logged in a .log file when the Enable Tracing feature is enabled, which is available in the Preferences window. The Preferences window can be opened by clicking **Preferences...** in the Edit menu.



Enable tracing

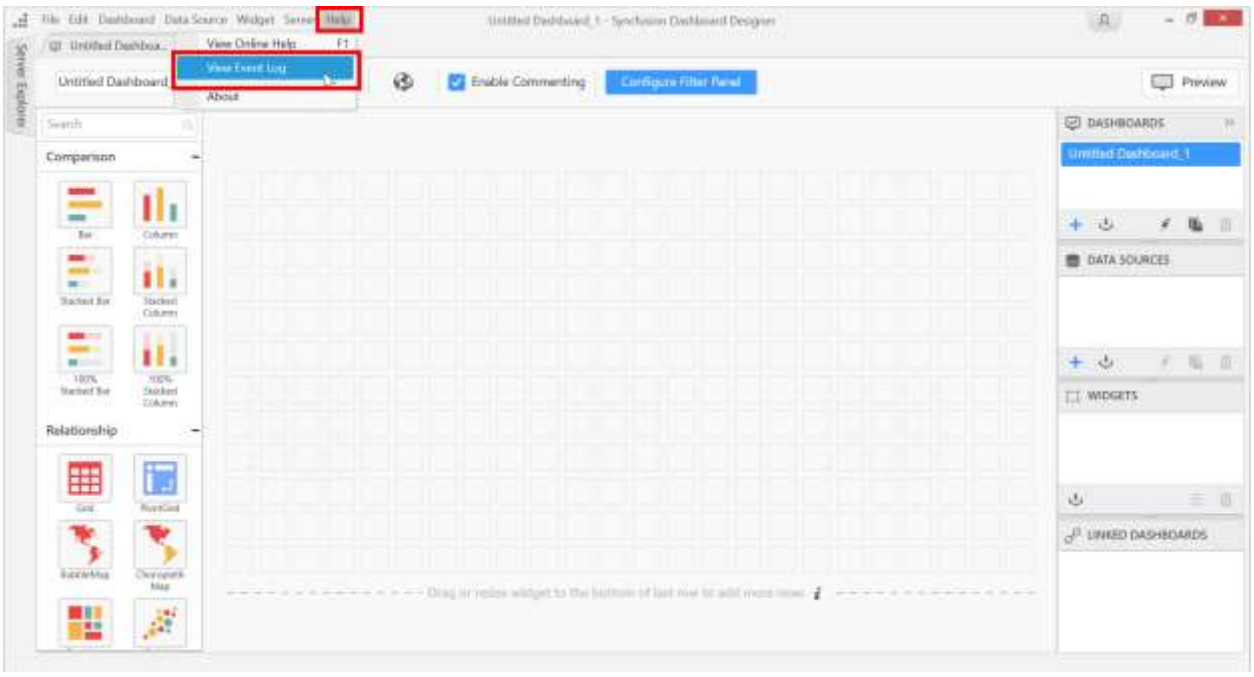
By default, logging the events will be disabled. Click **Enable Tracing** in the Preferences window to enable the logging process.



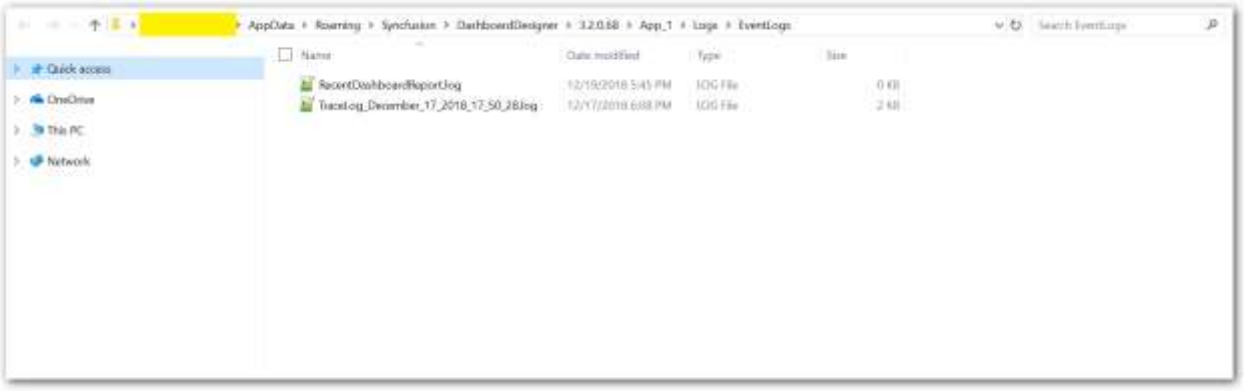
Note: The settings will be remembered when you launch the application next time, and the events will be logged automatically if the tracing is enabled.

View log files

You can view the log files by using the **View Event Log** option under the Enable Tracing menu item.



The log file folder will be shown in the Files Explorer as shown in the following screenshot.



A new .log file will be created every time when you launch the application.

When errors occur while working, the log file having the name as **RecentDashboardLog.log** and the latest trace log file having the name as **TraceLog*Month*date*year*mm_ss.log** are collected and shared with Syncfusion support through the Direct-Trac support system. The user interactions and some debug related details will be logged in this file.

Troubleshooting Errors

The following troubleshooting steps help you resolve the known issues in Dashboard Designer.

DD001

CODE	DD001
TEXT	Dashboard cannot be opened since it was created using a newer version of the Dashboard Designer application.
DESCRIPTION	Newer version dashboard cannot be opened in older version of Dashboard Designer.
SOLUTION	Upgrade the designer to the latest version.

DD002

CODE	DD002
TEXT	You cannot connect to a server with different version.
DESCRIPTION	Version conflicts between Dashboard Designer and Dashboard Server.
SOLUTION	Upgrade the server or designer to the latest version.

DD003

CODE	DD003
TEXT	The file that you are trying to open is not a valid dashboard file.
DESCRIPTION	The loaded file may either be corrupted or invalid.
SOLUTION	The opened file may be altered or corrupted. Please relaunch the application and then open the dashboard.

DD004

CODE	DD004
TEXT	An error occurred while sending request to the server.
DESCRIPTION	There might be a problem with the provided server URL.
SOLUTION	1. Make sure that your device has been connected to the internet. 2. Check whether the server URL is valid. 3. Check whether the server is running or not. If not, start the Dashboard Server from the desktop shortcut or installed location.

DD005

CODE	DD005
TEXT	Could not publish the request item as public.
DESCRIPTION	Mark as public has been restricted in Dashboard Server.
SOLUTION	Enable the â€˜Mark dashboard and widget as publicâ€™ option in the Dashboard Serverâ€™s settings.

DD006

CODE	DD006
TEXT	The Dashboard Service could not be contacted.
DESCRIPTION	Dashboard windows service is not installed in the system. Either install the service or change the preview preference to use IIS.
SOLUTION	1. Make sure that dashboard windows service is configured or installed. 2. IIS Express is required to start the dashboard service. If it is not installed on the machine, please install (IIS Express) before starting the Dashboard Service.

DD007

CODE	DD007
TEXT	The loaded file is not a valid widget file.
DESCRIPTION	The loaded dashboard file may either be corrupted or invalid.
SOLUTION	The opened file may be altered or corrupted. Please relaunch the application and then open the widget.

DD008

CODE	DD008
TEXT	The required assembly was not found.
DESCRIPTION	One or more required assemblies missing at installation location. It may be either moved or renamed.
SOLUTION	Please reinstall the Dashboard Designer application or contact Syncfusion support.

DD009

CODE	DD009
TEXT	Unable to launch the dashboard.
DESCRIPTION	Login information is not valid or invalid dashboard.
SOLUTION	The logged-in server is not available. Dashboard might have been moved or deleted.

DD010

CODE	DD010
TEXT	Could not import the requested Excel file.
DESCRIPTION	This Excel file is saved in the older Excel 95 format.
SOLUTION	This work book has been saved in the older Excel 95 format so import it after upgrading to a newer version.

DD011

CODE	DD011
TEXT	Access denied to add item.
DESCRIPTION	Access denied to the logged-in user.
SOLUTION	Please enable write access to user in the Dashboard Server's settings.

DD012

CODE	DD012
TEXT	An error occurred while opening the web page.
DESCRIPTION	Cannot navigate to the requested URL.
SOLUTION	Make sure that your device has been connected to the internet.

DD013

CODE	DD013
TEXT	Could not link the dashboard.
DESCRIPTION	Unable to add link to the requested dashboard.
SOLUTION	1. Please check the URL 2. Check whether the requested dashboard is present in server or not. 3. Make sure that the device has been connected to the internet

DD014

CODE	DD014
TEXT	Invalid user filter file.
DESCRIPTION	The loaded user filter file may either be corrupted or invalid.
SOLUTION	Check whether the file is in a valid format or not.

DD015

CODE	DD015
TEXT	Could not import the selected user filter.
DESCRIPTION	The user filter file that you are trying to import is not supported with the present data source.
SOLUTION	Check whether the data source is available or not.

DD016

CODE	DD016
TEXT	Could not import the requested item as public.
DESCRIPTION	The file type data source cannot be imported as public.
SOLUTION	1. Check whether the device has been connected to the internet, if the file is connected from the server. 2. Check whether the server URL is valid, if the file is connected from the server. 3. Check whether read permission is restricted for the current user, if the file is connected from the server. 4. Check whether data file exists in the connected location (while added as new data source).

DD017

CODE	DD017
TEXT	Invalid Column name
DESCRIPTION	There might be a mismatch between the column names or the column does not exist in the connected data source.
SOLUTION	The mentioned column may be renamed or deleted from the data source. Please reconfigure the widget or data source.

DD018

CODE	DD018
TEXT	The connection returned JSON data is invalid.
DESCRIPTION	The JSON file that you are trying to connect may be either corrupted or invalid.
SOLUTION	Check whether the JSON data is in valid format or not.

DD019

CODE	DD019
TEXT	Error occurred while converting JSON data into table(s).
DESCRIPTION	Improper structural format is designed in JSON file.
SOLUTION	Check whether the JSON file is in a valid format or not.

DD020

CODE	DD020
TEXT	An error occurred while downloading the file from online.
DESCRIPTION	Unable to download the file from the given online link due to invalid link or dashboard server running status.
SOLUTION	1. Check whether your device has been connected to the internet. 2. Check whether the shareable link is valid. 3. Check whether the server is running or not. If not, start the Dashboard Server from the desktop shortcut or installed location.

DD021

CODE	DD021
TEXT	An error occurred while migrating data to the target server.

DESCRIPTION	There might be a problem with internet access or the dashboard data agent service has been stopped.
SOLUTION	1. Check whether the device has been connected to the internet. 2. Check whether the Dashboard Data Agent is running or not. If not, start the Dashboard Data Agent from the desktop shortcut or installed location.

DD022

CODE	DD022
TEXT	An error occurred while sending request to the DIP server.
DESCRIPTION	There might be a problem with either provided DIP server URL or credentials or the DIP server has been stopped.
SOLUTION	1. Check whether your device has been connected to the internet. 2. Check if the DIP server URL is valid. 3. Check whether the entered username and password are valid. 4. Check whether the DIP server is running or not. If not, start the DIP Server from the desktop shortcut or installed location.

Unknown Errors

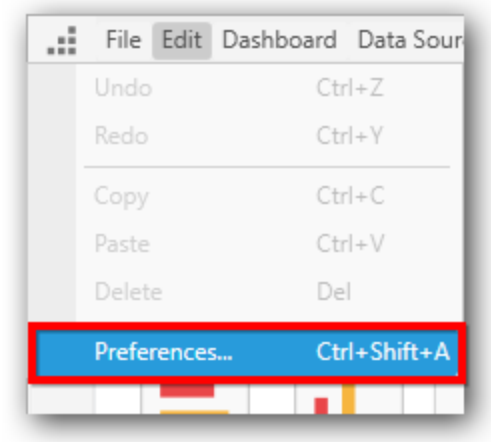
If you have faced any other unknown errors while using the Syncfusion Dashboard Designer application, contact us by creating a support incident with the necessary log files.

Preferences

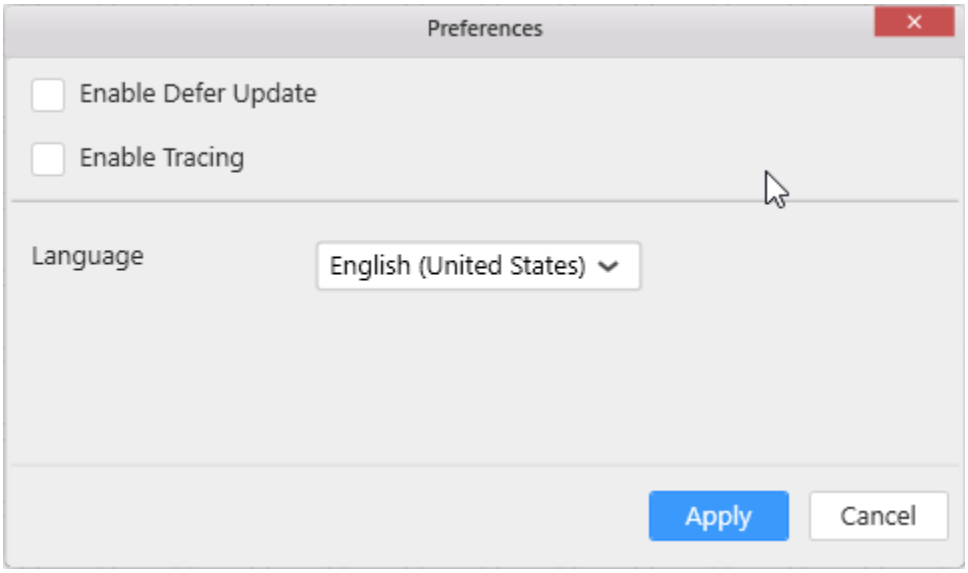
You can configure the Designer properties using Preferences option. The following properties are available in Preferences settings.

- Enable Defer Update
- Enable Tracing
- Language

To open the Preferences, click Edit menu and choose the Preferences... option. This can be also achieved by using the keyboard shortcut **Ctrl+Shift+A**



By Default Preferences... window shows like below,



Enable Defer Update

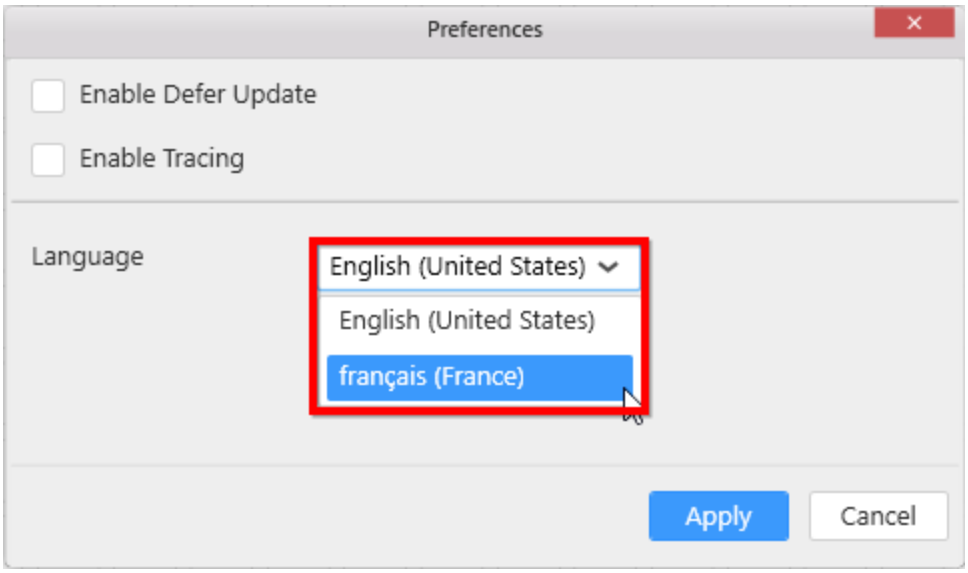
You can enable/ disable the Defer Update settings by Enable Defer Update option.

Enable Tracing

You can enable/ disable the Tracing settings by Enable Tracing option.

Language

You can choose the languages for the Dashboard Designer by using Language option.



Click here to know more about language changes in Dashboard Designer.

Once you have enabled these properties, it will maintain in Dashboard Designer until you change these properties.

