

Tibero

설치 안내서

Tibero 7

TMAXTibero

Copyright © 2025 TmaxTibero Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2025 TmaxTibero Co., Ltd. All Rights Reserved.

대한민국 경기도 성남시 분당구 정자일로 45, 티맥스소프트타워 우)13613

Website

<http://www.tmaxtibero.com>

기술서비스센터

Tel : +82-1544-8629

E-Mail : info@tmax.co.kr

Restricted Rights Legend

All TmaxTibero Software (Tibero®) and documents are protected by copyright laws and international convention. TmaxTibero software and documents are made available under the terms of the TmaxTibero License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxTibero Co., Ltd. Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxTibero trademarks, logos, or any other brand features.

This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

이 소프트웨어(Tibero®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TmaxTibero Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TmaxTibero의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 아니하며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보의 제공만을 목적으로 하고, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 아니하며, 사용설명서 상의 내용은 법적 또는 상업적인 특정한 조건을 만족시키는 것을 보장하지는 않습니다. 사용설명서의 내용은 제품의 업그레이드나 수정에 따라 그 내용이 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 아니합니다.

Trademarks

Tibero® is a registered trademark of TmaxTibero Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

Tibero®는 TmaxTibero Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

Detailed Information related to the license can be found in the following directory : \${INSTALL_PATH}/license/oss_licenses

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

관련 상세한 정보는 제품의 다음의 디렉터리에 기재된 사항을 참고해 주십시오. : \${INSTALL_PATH}/license/oss_licenses

안내서 정보

안내서 제목: Tibero 설치 안내서

발행일: 2025-09-30

소프트웨어 버전: Tibero 7.2.4

안내서 버전: v7.2.4

내용 목차

안내서에 대하여	xiii
제1장 설치 소개	1
1.1. 개요	1
1.2. 설치 구성 요소	2
1.2.1. 패키지 구성	2
1.2.2. 소프트웨어 배포	3
1.3. 시스템 요구 사항	3
1.3.1. 지원 플랫폼 및 운영체제	3
1.3.2. 하드웨어 요구사항	5
1.3.3. 소프트웨어 요구사항	6
제2장 설치 전 준비사항	7
2.1. 개요	7
2.2. 디스크 용량 확인	7
2.3. JDK 설치	7
2.4. 운영체제별 패키지 설치	8
2.5. 운영체제별 파라미터 설정	9
2.5.1. Linux	10
2.5.2. Solaris	12
2.5.3. AIX	13
2.6. NTP 서버 설정	15
2.7. 기타 확인사항	15
제3장 데이터베이스 설치	17
3.1. 개요	17
3.2. 수동 설치	17
3.2.1. Windows 환경	17
3.2.2. UNIX 환경	21
3.3. 설치 확인	25
3.3.1. 디렉터리 구조	25
3.3.2. 사용자 환경설정 파일의 적용 및 확인	34
3.3.3. 기동 및 종료	35
3.3.4. 사용자 계정	37
제4장 데이터베이스 제거	39
4.1. 개요	39
4.2. 수동 제거	39
4.2.1. Windows 환경	39
4.2.2. UNIX 환경	40
제5장 클라이언트 설치와 제거	41
5.1. 설치	41
5.2. 제거	45

제6장 멀티 인스턴스 설치와 제거	47
6.1. 설치	47
6.1.1. Windows 환경	47
6.1.2. Unix 환경	49
6.2. 제거	50
6.2.1. Windows 환경	50
6.2.2. Unix 환경	51
제7장 TAC 설치와 제거	53
7.1. 설치 전 준비사항	53
7.1.1. 인터커넥트 구성	53
7.1.2. IP 주소 및 포트 정보 확인	53
7.1.3. 소켓 버퍼 설정	54
7.1.4. 공유 디스크 타입 확인	55
7.2. 수동 설치	56
7.3. 설치 확인	56
7.4. 제거	56
Appendix A. 설치 후 문제 해결	59
A.1. 사용자 환경설정 파일	59
A.1.1. TB_HOME	59
A.1.2. LD_LIBRARY_PATH	59
A.1.3. TB_SID	59
A.2. TAC 설치	60
A.2.1. 노드 추가	60
A.2.2. TAC raw device 환경에서 TPR 관련 주의사항	61
A.3. 기타	62
A.3.1. 포트 번호	62
A.3.2. dbtimezone	62
Appendix B. 마운트	63
B.1. AIX	63
B.2. Linux	63
B.3. Solaris	64
Appendix C. Tibero 지원 문자 집합	65
Appendix D. system.sh	67
Appendix E. HugePage 설정 방법	69
E.1. Linux	69
E.1.1. 활성화	69
E.1.2. 비활성화	71
E.1.3. THP(Transparent Huge Pages) 비활성화	71
E.1.4. Pagetable Size 계산	72
E.2. AIX	73
E.2.1. 활성화	73

E.2.2. 비활성화	74
E.3. Solaris	74
E.3.1. 활성화	75
E.3.2. 비활성화	75
Appendix F. Undo 테이블 스페이스 크기 산정	77
F.1. 개요	77
F.2. 고려사항	77
F.3. 용량 계산 방법	78
Appendix G. Shared Memory 크기 산정	81
G.1. 개요	81
G.2. 고려사항	81
G.3. 용량 산정	82
Appendix H. system_install.sh	85
H.1. 목적	85
H.2. 옵션	85
Appendix I. system_patch.sh	87
I.1. 목적	87
I.2. 옵션	87
Appendix J. sample user 설치	89
J.1. 소개	89
J.2. 스키마	89
J.3. 설치 및 삭제 방법	90
색인	91

그림 목차

[그림 5.1]	Tibero Client Installer - Introduction	41
[그림 5.2]	Tibero Client Installer - Pre-Install settings	42
[그림 5.3]	Tibero Client Installer - Select Modules	42
[그림 5.4]	Tibero Client Installer - Pre-Installation Summary	43
[그림 5.5]	Tibero Client Installer - Install Progress	44
[그림 5.6]	Tibero Client Installer - Install Complete	44
[그림 5.7]	Tibero Client Uninstaller - Introduction	45
[그림 5.8]	Tibero Client Uninstaller - Uninstall Progress	46
[그림 5.9]	Tibero Client Uninstaller - Uninstall Complete	46
[그림 F.1]	Tibero TPR Report	79

표 목차

[표 E.1] 페이지 크기별 page table size (SHM 50GB 기준)	72
---	----

안내서에 대하여

안내서의 대상

본 안내서는 Tibero[®](이하 Tibero)를 설치 및 제거하고자 하는 모든 데이터베이스 사용자를 대상으로 기술한다.

안내서의 전제 조건

본 안내서는 Tibero를 시스템에 설치하는 과정을 설명한 안내서이다. 따라서 본 안내서를 원활히 이해하기 위해서는 다음과 같은 사항을 미리 알고 있어야 한다.

- 데이터베이스의 이해
- RDBMS의 이해
- 운영체제 및 시스템 환경의 이해
- UNIX 계열(LINUX 포함)의 기본 지식

안내서의 제한 조건

본 안내서는 Tibero를 실무에 적용하거나 운용하는 데 필요한 모든 사항을 포함하지 않는다.

안내서 구성

Tibero 설치 안내서는 총 7개의 장과 Appendix로 이루어져 있다.

각 장의 주요 내용은 다음과 같다.

- 제1장: 설치 소개

Tibero를 간략히 소개하고 설치에 필요한 시스템 요구사항을 기술한다.

- 제2장: 설치 전 준비사항

Tibero를 설치하기에 앞서 확인하고 설정해야 할 준비 사항을 기술한다.

- 제3장: 데이터베이스 설치

Tibero의 설치하는 방법을 기술한다.

- 제4장: 데이터베이스 제거

Tibero의 제거하는 방법을 기술한다.

- 제5장: 클라이언트 설치와 제거

Windows 환경에 대하여 인스톨러를 이용한 클라이언트 설치 방법에 대하여 설명한다.

- 제6장: 멀티 인스턴스 설치

멀티 인스턴스를 설치하는 과정에 대해서 설명한다.

- 제7장: TAC(Tibero Active Cluster) 설치

Tibero Active Cluster(이하 TAC) 환경을 구축하기 위한 Tibero의 설치 방법을 설명한다.

- Appendix A: 설치 후 문제 해결

Tibero를 설치한 후 발생할 수 있는 문제의 해결 방법을 기술한다.

- Appendix B: 마운트

플랫폼별로 CD를 마운트하는 방법을 기술한다.

- Appendix C: Tibero 지원 문자 집합

Tibero에서 지원하는 문자 집합을 설명한다.

- Appendix D: system.sh

system.sh(vbs)를 수행하는 경우 각 옵션에 대하여 설명한다.

- Appendix E: HugePage 설정 방법

각 운영체제별 HugePage 설정 방법에 대하여 설명한다.

- Appendix F: Undo 테이블 스페이스 크기 산정

Undo 테이블 스페이스 크기 산정 방법에 대하여 설명한다.

- Appendix G: Shared Memory 크기 산정

Tibero의 SQL Execution Work Area 크기 산정방법에 대해 설명한다.

안내서 규약

표기	의미
<<AaBbCc123>>	프로그램 소스 코드의 파일명
<Ctrl>+C	Ctrl과 C를 동시에 누름
[Button]	GUI의 버튼 또는 메뉴 이름
진하게	강조
""(따옴표)	다른 관련 안내서 또는 안내서 내의 다른 장 및 절 언급
'입력항목'	화면 UI에서 입력 항목에 대한 설명
하이퍼링크	메일 계정, 웹 사이트
>	메뉴의 진행 순서
+----	하위 디렉터리 또는 파일 있음
----	하위 디렉터리 또는 파일 없음
참고	참고 또는 주의사항
주의	주의할 사항
[그림 1.1]	그림 이름
[예 1.1]	예제 이름
AaBbCc123	Java 코드, XML 문서
[command argument]	옵션 파라미터
< xyz >	'<'와 '>' 사이의 내용이 실제 값으로 변경됨
	선택 사항. 예) A B: A나 B 중 하나
...	파라미터 등이 반복되어서 나옴
\${ }	환경변수

관련 안내서

안내서	설명
Tibero tbCLI 안내서	Call Level Interface인 tbCLI의 개념과 구성요소, 프로그램 구조를 소개하고 tbCLI 프로그램을 작성하는 데 필요한 데이터 타입, 함수, 에러 메시지를 기술한 안내서이다.
Tibero 애플리케이션 개발자 안내서	각종 애플리케이션 라이브러리를 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한 안내서이다.
Tibero External Procedure 안내서	External Procedure를 소개하고 이를 생성하고 사용하는 방법을 기술한 안내서이다.
Tibero JDBC 개발자 안내서	Tibero에서 제공하는 JDBC 기능을 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한 안내서이다.
Tibero tbESQL/C 안내서	C 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한 안내서이다.
Tibero tbESQL/COBOL 안내서	COBOL 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한 안내서이다.
Tibero tbPSM 안내서	저장 프러시저 모듈인 tbPSM의 개념과 문법, 구성요소를 소개하고, tbPSM 프로그램을 작성하는 데 필요한 제어 구조, 복합 타입, 서브 프로그램, 패키지와 SQL 문장을 실행하고 에러를 처리하는 방법을 기술한 안내서이다.
Tibero tbPSM 참조 안내서	저장 프러시저 모듈인 tbPSM의 패키지를 소개하고, 이러한 패키지에 포함된 각 프러시저와 함수의 프로토타입, 파라미터, 예제 등을 기술한 참조 안내서이다.
Tibero 관리자 안내서	Tibero의 동작과 주요 기능의 원활한 수행을 보장하기 위해 DBA가 알아야 할 관리 방법을 논리적 또는 물리적 측면에서 설명하고, 관리를 지원하는 각종 도구를 기술한 안내서이다.
Tibero 유틸리티 안내서	데이터베이스와 관련된 작업을 수행하기 위해 필요한 유틸리티의 설치 및 환경설정, 사용 방법을 기술한 안내서이다.
Tibero 에러 참조 안내서	Tibero를 사용하는 도중에 발생할 수 있는 각종 에러의 원인과 해결 방법을 기술한 안내서이다.
Tibero	Tibero의 동작과 사용에 필요한 초기화 파라미터와 데이터 사전, 정적 뷰, 동적 뷰를 기술한 참조 안내서이다.

안내서	설명
참조 안내서	
Tibero SQL 참조 안내서	데이터베이스 작업을 수행하거나 애플리케이션 프로그램을 작성할 때 필요한 SQL 문장을 기술한 참조 안내서이다.
Tibero Spatial 참조 안내서	Tibero에서 Geometry 타입에 대한 설명과 Spatial 기능 관련 프러시저 함수 목록 및 사용 방법 등을 기술한 안내서이다.
Tibero TEXT 참조 안내서	Tibero의 제공하는 Text Index를 소개하고, Text Index를 생성 하고 사용하는 방법을 기술하는 안내서이다.
Tibero TDP.NET 안내서	Tibero Data Provider for .NET 기능을 기술하는 안내서이다.
Tibero IMCS 안내서	Tibero에서 제공하는 In-Memory Column Store(이하 IMCS) 기능을 기술하는 안내서이다.

제1장 설치 소개

본 장에서는 Tibero를 간략히 소개하고 설치에 필요한 시스템 요구사항을 설명한다.

1.1. 개요

현재 기업의 비즈니스는 폭발적인 데이터의 증가와 다양한 환경 및 플랫폼의 등장으로 빠르게 확장되고 있다. 새로운 비즈니스 환경이 도래함에 따라 보다 더 효율적이고 유연한 데이터 서비스와 정보의 처리, 데이터 관리 기능이 필요하게 되었다.

Tibero는 이러한 변화에 맞춰 기업 비즈니스 구현의 기반이 되는 데이터베이스 인프라 구성을 지원하며 고성능, 고가용성 및 확장성의 문제를 해결하는 엔터프라이즈 데이터베이스 관리 시스템이다.

기존 DB의 단점을 보완하기 위해 Tibero는 독자적인 Tibero Thread Architecture를 채택, 구현하였다. 한정된 서버 프로세스의 CPU 및 메모리 등의 시스템 리소스를 효율적으로 사용하면서 뛰어난 성능과 안정성 및 확장성을 보장하고 편리한 개발 환경과 관리 기능을 제공한다.

Tibero는 초기 설계부터 대규모 사용자, 대용량 데이터, 강화된 안정성, 향상된 호환성 측면 등에서 다른 DBMS와 차별화를 고려하여 개발되었다.

Tibero는 대용량의 데이터를 관리하고 안정적으로 비즈니스의 연속성을 보장하는 데이터 관리 솔루션이다. Tibero는 이를 위해 RDBMS 환경에서 요구되는 주요 기능인 분산 데이터베이스 링크(Distributed Database Link), 데이터 이중화(Data Replication), 데이터베이스 클러스터(Data Cluster), 병렬 질의 처리(Parallel Query Processing) 등의 기능을 갖추고 있다. Tibero는 이처럼 기업이 원하는 최적의 데이터베이스 환경을 제공하는 대표적인 DB이다.

1.2. 설치 구성 요소

1.2.1. 패키지 구성

Tibero의 패키지 구성은 다음과 같다.

- Tibero 제품 CD 1매
- Tibero 제품 라이선스 및 보증
- Tibero 안내서
 - Tibero 릴리즈 노트
 - Tibero 소개
 - Tibero 설치 안내서
 - Tibero 관리자 안내서
 - Tibero 애플리케이션 개발자 안내서
 - Tibero External Procedure 안내서
 - Tibero Hadoop Connector 안내서
 - Tibero JDBC 개발자 안내서
 - Tibero tbCLI 안내서
 - Tibero tbESQL/C 안내서
 - Tibero tbESQL/COBOL 안내서
 - Tibero tbPSM 안내서
 - Tibero tbPSM 참조 안내서
 - Tibero TEXT 참조 안내서
 - Tibero Spatial 참조 안내서
 - Tibero Active Storage 관리자 안내서
 - Tibero TDP.NET 안내서
 - Tibero IMCS 안내서
 - Tibero 유틸리티 안내서
 - Tibero 에러 참조 안내서
 - Tibero 참조 안내서

1.2.2. 소프트웨어 배포

Tibero의 소프트웨어 배포 방식은 다음과 같다.

- 정식 버전

CPU 개수와 기능별 라이선스이다.

- 평가판 버전

날짜와 사용자의 수를 제한하는 방식의 라이선스이다.

데모 라이선스 파일은 테크넷을 통하여 발급받을 수 있다.

<http://technet.tmaxsoft.com/ko/front/main/main.do>

1.3. 시스템 요구 사항

본 절에서는 지원 플랫폼과 운영체제와 하드웨어, 소프트웨어 요구사항에 대해서 설명한다.

1.3.1. 지원 플랫폼 및 운영체제

Tibero의 지원 플랫폼 및 운영체제는 다음과 같다.

H/W, S/W	CPU	OS	Binary Bits
SUN	SPARC	Solaris 11	64bits
IBM	PPC	AIX 7.1 AIX 7.2 AIX 7.3	64bits
GNU	X86	Red Hat Enterprise Linux 7 Red Hat Enterprise Linux 8.1 Red Hat Enterprise Linux 8.2 Red Hat Enterprise Linux 8.3 Red Hat Enterprise Linux 8.4 Red Hat Enterprise Linux 8.5	64bits

H/W, S/W	CPU	OS	Binary Bits
		Red Hat Enterprise Linux 8.6	
		Red Hat Enterprise Linux 8.7	
		Red Hat Enterprise Linux 8.8	
		Red Hat Enterprise Linux 8.9	
		Red Hat Enterprise Linux 8.10	
		Red Hat Enterprise Linux 9.3	
		Red Hat Enterprise Linux 9.4	
		Red Hat Enterprise Linux 9.5	
		Red Hat Enterprise Linux 9.6	
		Oracle Linux 8.6	
		Oracle Linux 8.7	
		Oracle Linux 8.8	
		Oracle Linux 8.9	
		Oracle Linux 8.10	
		Oracle Linux 9.3	
		Oracle Linux 9.4	
		Oracle Linux 9.5	
		Oracle Linux 9.6	
		Rocky Linux 8.6	
		Rocky Linux 8.8	
		Rocky Linux 8.9	
		Rocky Linux 8.10	
		Rocky Linux 9.3	
		Rocky Linux 9.4	
		Rocky Linux 9.5	
		Rocky Linux 9.6	
		ProLinux 7.5	

H/W, S/W	CPU	OS	Binary Bits
		ProLinux 8.5 CentOS 7 Ubuntu 22.04 SUSE Linux Enterprise Server 15 SP6	
Microsoft	Intel(x86), AMD64	Windows Server 2016 Windows Server 2019 Windows Server 2022 Windows Server 2025	64bits

참고

현재 Windows 플랫폼의 경우 TAC를 지원하지 않는다.

1.3.2. 하드웨어 요구사항

Tibero를 설치하기 위해 필요한 하드웨어 요구사항은 다음과 같다.

참고

단, 파라미터 기본값을 기준으로 한 요구사항이기 때문에 요구사항을 충족하더라도 DB 파라미터 설정에 따라 설치가 정상적으로 되지 않을 수 있다.

Platform	RAM	Swap Space	/tmp Directory Space	HDD Space (Full / Client Only)
LINUX/x86	2GB	4GB	500MB	2.5GB / 400MB
Solaris	2GB	4GB	500MB	2.5GB / 400MB
AIX	2GB	4GB	500MB	2.5GB / 400MB
Windows	2GB	-	-	2GB / 400MB

Solaris 플랫폼의 경우 Swap Space를 설정하는 경우 아래 사항을 참고한다.

- 권장 설정 : 물리 메모리의 두 배
- 사용자 설정

(백그라운드 프로세스 개수 + 워커 프로세스 개수) * 프로세스 당 Swap 사용량(300MB)

1.3.3. 소프트웨어 요구사항

Tibero를 설치하기 위해 필요한 소프트웨어 요구사항은 다음과 같다.

Platform	OS	Compiler	JDK Version
LINUX/x86	Red Hat Enterprise Linux 7 kernel 3.10.0 이상	C99 지원 컴파일러, gcc version 4.8.5 이상	JDK 1.8 이상
Solaris	Solaris 11 64Bit 커널	C99 지원 컴파일러, Sun C 5.8 2005/10/13	JDK 1.8 이상
AIX	AIX 7.1 64Bit 커널 AIX 7.2 64Bit 커널 AIX 7.3 64Bit 커널	C99 지원 컴파일러, IBM XL C/C++ Enterprise Edition V7.0	JDK 1.8 이상
Windows	Windows Server 2016(64bit Only) Windows Server 2019(64bit Only) Windows Server 2022(64bit Only)	C99 지원 컴파일러	JDK 1.8 이상

AIX 7.1에서 컴파일된 바이너리의 경우 AIX 7.1 Technology Level 4 이상의 사양에서 정상적으로 설치 및 운영이 가능하다.

```
$ oslevel -r  
  
7100-04
```

참고

Windows AMD64(x64)의 경우 **Microsoft Visual C++ 2008 Redistributable Package(x64)**나 **Microsoft.NET Framework 3.5 SP1**이 미리 설치되어 있어야 정상적으로 설치 및 운영이 가능하다.

제2장 설치 전 준비사항

본 장에서는 Tibero를 설치하기에 앞서 필요한 하드디스크의 용량 확인 및 JDK 설치, 운영체제별 커널 파라미터(Kernel parameters)의 설정을 설명한다.

2.1. 개요

Tibero를 설치하기 전에 확인하고 설정해야 할 준비사항은 다음과 같다.

- [디스크 용량 확인](#)
- [JDK 설치](#)
- [운영체제별 패키지 설치](#)
- [운영체제별 커널 파라미터\(Shell Limits 파라미터\) 설정](#)
- [NTP 서버 설정](#)
- [호스트명과 포트 번호, 시스템 계정, localhost 확인](#)

2.2. 디스크 용량 확인

Tibero를 설치하기 위해서는 각 플랫폼별로 약간의 차이가 있지만 최소 2GB 이상의 하드디스크 여유공간이 필요하다. 이는 Tibero를 설치하고 나서 데이터베이스를 생성할 때 최소로 필요한 하드디스크 공간을 의미하기도 한다.

참고

하드웨어 요구사항에 대한 자세한 내용은 “[1.3. 시스템 요구 사항](#)”을 참고한다.

하드디스크 용량을 확인하는 방법은 여러 가지가 있다. UNIX 계열 시스템에서는 **df** 명령어로 확인할 수 있고, Windows 계열 시스템에서는 설치할 하드디스크의 **속성** 항목을 통해 확인이 가능하다.

2.3. JDK 설치

Tibero를 설치하기 전에 **JDK 1.8** 이상이 반드시 설치되어 있어야 한다.

다음 위치에서 JDK를 다운로드할 수 있다.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

만약 시스템이 Oracle사의 JDK를 사용하지 않는다면 각각의 시스템에 적합한 JDK를 찾아 설치한다. 예를 들어 AIX는 IBM에서 JDK를 다운로드받아 설치한다.

각 시스템별 JDK 설치 방법은 다음 위치에서 확인할 수 있다.

<http://www.oracle.com/technetwork/java/index.html>

참고

벤더별 JDK 설치 방법은 각 벤더에서 제공하는 설치 안내서를 참고한다.

2.4. 운영체제별 패키지 설치

본 절에서는 운영체제별로 요구하는 패키지에 대해 설명한다.

각 패키지들은 반드시 해당 버전 이상의 패키지가 설치되어 있어야 하며, 각 OS 및 버전별로 패키지명이나 버전이 상이할 수 있다.

Platform	Packages
Linux	gcc-3.4.6-11
	gcc-c++-3.4.6-11
	libgcc-3.4.6-11
	libstdc++-3.4.6-11
	libstdc++-devel-3.4.6-11
	libaio-0.3.105-2
	libaio-devel-0.3.105-2
	pstack
	glibc
Windows	WSH 5.6
Solaris	libCrun.so.1 (SUNW_1.6)
	pstack
AIX	procstack

[참고]

RHEL 8.1 이상 버전부터는 아래 패키지가 설치되어 있어야 한다.

Platform	Packages
Linux	libnsl.so.1
	libncurses.so.5

Ubuntu 22.04 이상 버전부터는 아래 패키지가 설치되어 있어야 한다.

Platform	Packages
Linux	libaio1
	libaio-dev
	libncurses5
	libtinfo5
	pstack

SUSE 15.6 이상 버전부터는 아래 패키지가 설치되어 있어야 한다.

Platform	Packages
Linux	libgcc_s1
	libstdc++6
	libnsl1
	libnsl2
	glibc
	libaio1
	pstack

2.5. 운영체제별 파라미터 설정

본 절에서는 운영체제별로 설정해야 할 커널 파라미터와 Shell Limits 파라미터를 설명한다. 운영체제별로 커널 파라미터의 값을 변경한 경우에는 반드시 시스템을 다시 기동시켜야 한다.

주의

세마포어는 Tiberio만 사용하는 것이 아니기 때문에, 추가로 각 항목별로 여유분(OS의 다른 프로세스용)도 추가해 주어야 한다. (예. 각각 +1000)

참고

Tibero 전체 Thread 수는 대략 아래와 같다.

$$\text{MAX_SESSION_COUNT} + (\text{MAX_SESSION_COUNT} / \text{WTHR_PER_PROC}) + 500$$

2.5.1. Linux

Linux 환경에서 설정할 커널 파라미터와 Shell Limits 파라미터는 다음과 같다.

- 커널 파라미터

- 설정 파일

```
/etc/sysctl.conf
```

- 설정값

커널 파라미터	값
kernel.sem	순서대로 SEMMSL, SEMMNS, SEMOPM, SEMMNI 최소 설정값은 아래와 같다. – SEMMSL : 2 / 권장값 : (Tibero 전체 Thread 수) x 2 – SEMMNS : (Tibero 전체 Thread 수) x 2 – SEMOPM : 2(=SEMMSL) / 권장값 : (Tibero 전체 Thread 수) x 2 – SEMMNI : (Tibero 전체 Thread 수) max값을 높게 설정해도 큰 문제가 없기에 여유롭게 설정하는 것을 권장한다.
kernel.shmall	ceil(shmmax/PAGE_SIZE)
kernel.shmmax	TOTAL_SHM_SIZE 이상 (byte)
kernel.shmmni	4096
fs.nr_open	nofile 파라미터 이상으로 설정해야만 한다.
fs.file-max	(nofile 파라미터) x (WTHR_PROC_CNT + PEP_PROC_CNT) 또는 67108864 으로 설정한다.
fs.aio-max-nr	1048576
net.ipv4.ip_lo cal_port_range	1024 65000
net.core.rmem_de fault	262144, OS 기존 설정값 중 최댓값

커널 파라미터	값
net.core.wmem_defaul	262144, OS 기존 설정값 중 최댓값
net.core.rmem_max	67108864, OS 기존 설정값, TCP_RCVBUF_SIZE, _INC_TCP_RCVBUF_SIZE 중 최댓값
net.core.wmem_max	67108864, OS 기존 설정값, TCP_SNDBUF_SIZE, _INC_TCP_SNDBUF_SIZE 중 최댓값

- Shell Limits 파라미터

- 설정 파일

```
/etc/security/limits.conf
```

- 설정값

파라미터	설명
nofile	(WTHR_PER_PROC * ((total data files in db) + 15)) + (tbsvr process count + 5) + 100 또는 크게 3,000,000 으로 설정한다.
nproc	MAX_SESSION_COUNT+10000 이상으로 설정한다. – Soft Limit : 65536 – Hard Limit : 65536

파일 디스크립터 한도 설정 주의

Linux 시스템에서 애플리케이션이 동시에 열 수 있는 파일 디스크립터 수는 여러 계층의 설정에 의해 제한된다. 일반적으로 *nofile*, *fs.nr_open*, *fs.file-max* 세 가지 파라미터에 의해 제약을 받는다. 계층 관계에 따라 반드시 아래 관계를 충족해야 한다.

```
nofile <= fs.nr_open <= fs.file-max
```

[참고]

RHEL 7.2 이상 버전부터는 아래 커널 파라미터 설정이 필요하다.

- 설정 파일

```
/etc/systemd/logind.conf
```

- 설정값

커널 파라미터	값
RemoveIPC	No

참고

SELinux를 enforce mode로 사용할 경우 프로세스가 비정상적인 동작을 할 수 있어 Tibero를 사용하는 경우 enforce mode로 설정하는 것을 권장하지 않는다.

2.5.2. Solaris

Solaris 환경에서 설정할 커널 파라미터와 Shell Limits 파라미터는 다음과 같다.

- 커널 파라미터

- 설정 파일

```
/etc/system
```

- 설정값

커널 파라미터	값
project.max-sem-ids	(Tibero 전체 Thread 수) x 2
process.max-sem-nsems	10000
process.max-sem-ops	200
project.max-shm-memory	4294967295 (물리적인 메모리의 절반)
project.max-shm-ids	100
project.max-lwps	MAX_SESSION_COUNT+10000 이상

- Shell Limits 파라미터

- 설정 파일

```
/etc/system
```

- 설정값

커널 파라미터	값
nofile	(WTHR_PER_PROC * ((total data files in db) + 15)) + (tbsvr process count + 5) + 100 또는 크게 3,000,000 으로 설정한다.
nproc	– Soft Limit : 65536

커널 파라미터	값
	– Hard Limit : 65536

[참고]

Solaris 10부터는 프로젝트를 이용하여 각 사용자 단위로 파라미터를 관리하는 것을 권장한다.

다음은 사용자 단위로 파라미터를 적용하는 예제이다.

1. 프로젝트를 생성한다.

```
# projadd -c "tibero" 'user.tibero'
```

2. 파라미터를 설정한다.

```
# projmod -sK "project.max-shm-memory=(privileged,4294967295,deny)" user.tibero
```

3. 해당 유저로 재접속한다.

아래와 같은 방법으로 변경된 파라미터 내역에 대해 확인이 가능하다.

```
# prctl -n project.max-shm-memory -i project user.tibero
project: 100: user.tibero
NAME      PRIVILEGE      VALUE      FLAG      ACTION      RECIPIENT
project.max-shm-memory
          privileged    4.00GB     -         deny        -
          system      16.0EB     max        deny        -
```

2.5.3. AIX

AIX 환경에서 설정할 커널 파라미터와 Shell Limits 파라미터는 다음과 같다.

- 커널 파라미터

AIX 환경에서 변경이 필요한 유일한 커널 파라미터는 **maxuproc**이다. 해당 파라미터는 **smitty** 또는 **chdev** 명령어를 통해 아래와 같이 변경한다. AIX 커널은 필요에 따라 자원을 사전 정의된 한계까지 동적으로 할당 및 재할당하기 때문에 그 외 별도의 커널 파라미터 설정은 필요하지 않다.

```
# 현재 설정된 값 확인
# lsattr -E -l sys0 -a maxuproc
maxuproc 1024 Maximum number of PROCESSES allowed per user True

# maxuproc 값 변경
# chdev -l sys0 -a maxuproc=16384
```

파라미터	설명
maxuproc	Tibero를 부팅할 때 기동하는 프로세스 개수와 연관된다. 따라서 DB 설치 유저에서 기동할 프로세스 개수 보다 큰 값으로 설정해야 한다. (권장 설정: 16384) [참고] 운영 환경에 따라 적절한 값을 계산할 때는 아래 내용을 참고한다. 사용자 설정 : 워커 프로세스 개수 + 백그라운드 프로세스 개수 + 시스템 프로세스 개수 + 여유분(100)

● Shell Limits 파라미터

– 설정 파일

/etc/security/limits

– 설정값

파라미터	권장 값
Soft FILE size	-1 (Unlimited)
Soft CPU time	-1 (Unlimited)
Soft DATA segment	-1 (Unlimited)
Soft STACK size	-1 (Unlimited)
Soft Real Memory size	-1 (Unlimited)

2.6. NTP 서버 설정

데이터베이스 운영 중에 xntpd 데몬에 의해 시스템 시간이 거꾸로 가게 되어 데이터가 잘못 출력되는 현상을 방지하기 위하여 시스템 시간을 동기화한 후에 "-x" 옵션을 주어 데몬을 재시작한다.

```
# stopsrc -s xntpd
# startsrc -s xntpd -a "-x"
```

시스템을 재시작할 때에도 해당 옵션을 적용하고자 한다면 "/etc/rc.tcpip"에서 아래의 내용에 대한 주석을 해제한다.

```
start /usr/sbin/xntpd "$src_running" "-x"
```

RHEL 8부터 xntpd 대신 chronyd가 기본 시간 동기화 데몬으로 사용된다.

```
# systemctl enable chronyd
# systemctl start chronyd
```

chronyd는 systemd 서비스로 관리되므로, 시스템 재시작 시 자동으로 실행된다. 추가적인 옵션이 필요하다면 /etc/chrony.conf 파일을 수정하여 서버 주소, 동기화 정책 등을 설정한다.

2.7. 기타 확인사항

Tibero의 설치를 시작하기 전에 호스트명과 포트 번호, 시스템 계정, localhost를 확인한다.

- 호스트명

라이선스를 요청할 때 필요한 호스트명을 확인한다. /etc/hosts 파일을 확인하거나 콘솔 화면에서 다음의 명령어를 입력하여 확인한다.

```
uname -n
```

- 포트 번호

Tibero가 기동할 때 부여될 포트 번호를 확인한다. (기본값: 8629)

- 시스템 계정

Tibero를 설치하고 운영할 시스템 계정을 확인한다.

- localhost

localhost가 제대로 설정되어 있는지 ping 명령을 통해 확인한다. 만약 제대로 설정되어 있지 않다면 /etc/hosts 파일을 수정한다.

```
ping localhost
```


제3장 데이터베이스 설치

본 장에서는 Tibero의 설치 방법에 대하여 설명한다.

3.1. 개요

Tibero에서는 제품 설치를 위한 방법을 다음의 형태로 제공하고 있다.

- 수동 설치

모든 플랫폼에 대하여 `tibero7-bin-<fixset>_<platform>.tar.gz` 형태로 제공된다.

3.2. 수동 설치

Tibero가 지원하는 모든 플랫폼에 대하여 데이터베이스를 수동으로 설치할 수 있다. 이를 수동 설치라 정의한다.

3.2.1. Windows 환경

다음은 Windows 환경에서 수동으로 Tibero를 설치하는 과정에 대한 설명이다. Windows 환경에서의 설치 과정은 반드시 관리자 권한으로 진행되어야 한다.

1. 다음의 두 가지 파일이 준비되었는지 확인한다.

- 바이너리 실행 파일(`tar.gz`)
- 라이선스 파일(`license.xml`)

2. **[시작] > [설정] > [제어판] > [시스템] > [고급]** 탭에서 **[환경변수]** 버튼을 클릭하여 환경변수 중 시스템 변수를 설정한다. 본 예제에서는 다음과 같이 시스템 변수를 설정한다.

환경변수	설정값
TB_HOME	C:\Tibero\tibero7
TB_SID	tibero
PATH	C:\Tibero\tibero7\bin;C:\Tibero\tibero7\client\bin

3. C:\Tibero 디렉터리에서 압축된 바이너리 실행 파일(`tar.gz`)을 해제한다. 그 후에 **%TB_HOME%**의 **license** 라는 하위 디렉터리에 발급받은 라이선스 파일(`license.xml`)을 복사한다.

4. %TB_HOME%\bin 디렉터리에서 다음의 명령어를 입력한다. 이때 서비스명은 생략할 수 있다.

```
tbinstall %TB_HOME% %TB_SID% [-s 서비스명]
```

이 명령이 실행되면 Tibero가 Windows 서비스에 등록된다. 등록된 Windows 서비스는 [시작] > [설정] > [제어판] > [관리 도구] > [서비스]에서 확인할 수 있다. 오른쪽 마우스를 클릭해서 [관리자 권한으로 실행]으로 명령 프롬프트 창을 실행한다.

```
C:\Tibero\tibero7\bin>tbinstall %TB_HOME% %TB_SID%
Microsoft (R) Windows Script Host 버전 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

TB_HOME = C:\Tibero\tibero7
TB_SID = tibero
service account = LocalSystem
Tibero_tibero installed successfully.
```

5. %TB_HOME%\config 디렉터리에서 다음의 명령어를 입력한다.

```
gen_tip.bat
```

이 명령이 실행되면 환경 파일(.tip)과 tbdsn.tbr, psm_commands.bat 파일이 생성된다.

```
C:\Tibero\tibero7\config>gen_tip.bat
Using TB_SID [tibero]
C:\Tibero\tibero7\config\tibero.tip generated
C:\Tibero\tibero7\client\config\tbdsn.tbr generated.
C:\Tibero\tibero7\config\psm_commands.bat generated
```

참고

1. tbdsn.tbr 파일 설정에 대해 자세한 내용은 "Tibero 관리자 안내서"의 "Appendix A. tbdsn.tbr"을 참고한다.
2. %TB_SID%.tip 파일 설정에 대해 자세한 내용은 "Tibero 참조 안내서"의 "Appendix A. 초기화 파라미터의 설정 예제"를 참고한다.

-
6. Tibero 서버를 'NOMOUNT 모드'로 기동한다.

```
C:\Tibero\tibero7\bin>tboot nomount

Tibero instance started up (NOMOUNT mode).
```

7. tbSQL 유틸리티를 이용하여 데이터베이스에 접속한다. 본 예제에서는 'sys' 사용자로 접속한다.

```
C:\Tibero\tibero7\bin>tbsql sys/tibero

tbSQL 7
```

Copyright (c) 2008, 2009, 2011, 2012 TmaxData Corporation. All rights reserved.

Connected to Tiberio.

8. **CREATE DATABASE** 문을 이용하여 원하는 데이터베이스를 생성한다. 본 예제에서는 'tiberio'로 생성한다.

```
SQL> create database "tiberio"
      user sys identified by tiberio
      maxinstances 8
      maxdatafiles 100
      character set MSWIN949
      national character set UTF16
      logfile
        group 1 'log001.log' size 50M,
        group 2 'log002.log' size 50M,
        group 3 'log003.log' size 50M
      maxloggroups 255
      maxlogmembers 8
      noarchivelog
        datafile 'system001.dtf' size 100M autoextend on next 10M maxsize unlimited
      default temporary tablespace TEMP
        tempfile 'temp001.dtf' size 100M autoextend on next 10M maxsize unlimited
      extent management local autoallocate
      undo tablespace UNDO
        datafile 'undo001.dtf' size 200M autoextend on next 10M maxsize unlimited
      extent management local autoallocate
      SYSSUB
        datafile 'syssub001.dtf' size 10M autoextend on next 10M maxsize unlimited
      default tablespace USR
        datafile 'usr001.dtf' size 100M autoextend on next 10M maxsize unlimited
      extent management local autoallocate;
Database created.
SQL> quit;
```

해당 과정에서 설정하는 주요 항목에 대한 설명은 다음과 같다.

항목	설명
maxdatafiles	데이터베이스에서 사용할 데이터 파일의 최댓값을 설정한다. (기본값: 100)
character set	데이터베이스의 문자 집합을 설정한다.
maxloggroups	로그 그룹의 최댓값을 설정한다. (기본값: 255)
maxlogmembers	로그 그룹 내의 로그 파일의 최댓값을 설정한다. (기본값: 8)

참고

CREATE DATABASE 문에 대한 자세한 내용은 "Tibero SQL 참조 안내서"를 참고한다.

9. 데이터베이스 생성이 완료되면 **tbboot** 명령어로 Tibero를 다시 기동한다.

```
C:\Tibero\tibero7\bin>tbboot

Tibero instance started up (NORMAL mode)
```

10. 기본 스크립트 호스트를 cscript로 변경한다. cscript로 설정되어 있지 않은 경우 vbs 파일을 실행할 때 사용자 입력 부분에서 경고창으로 출력되어 설치를 진행할 수 없는 현상이 발생한다.

```
C:\Tibero\tibero7\scripts> cscript //H:CScript
Microsoft (R) Windows Script Host 버전 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

기본 스크립트 호스트가 이제 "cscript.exe"로 설정되었습니다.

11. %TB_HOME%\scripts 디렉터리에서 **system.vbs** 명령어를 입력하면 현재 디렉터리에 있는 SQL 문이 저장된 파일(*.sql, 이하 SQL 파일)이 수행된다.

SQL 파일을 수행하면 role, system user, view, package 등이 생성된다. 사용되는 sys 및 syscat 계정에 대한 기본 암호는 각각 tibero, syscat이다.

```
C:\Tibero\tibero7\scripts>system.vbs
Microsoft (R) Windows Script Host 버전 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

Creating the role DBA...
    Running C:\Tibero\tibero7\scripts\create_dba_gen.sql...
Creating system users & roles...
    Running C:\Tibero\tibero7\scripts\system_users.sql...
Creating virtual tables(1)...
    Running C:\Tibero\tibero7\scripts\vt_drop_gen.sql...
Creating virtual tables(2)...
    Running C:\Tibero\tibero7\scripts\vt_create_gen.sql...
Granting public access to _VT_DUAL...
    Running C:\Tibero\tibero7\scripts\vt_dual.sql...
Creating the system generated sequences...
    Running C:\Tibero\tibero7\scripts\create_seq.sql...
Creating internal dynamic performance views...
    Running C:\Tibero\tibero7\scripts\dpvx.sql...
Creating system packages:
    Running C:\Tibero\tibero7\scripts\pkg\pkg_standard.sql...
    Running C:\Tibero\tibero7\scripts\pkg\pkg_dbms_output.sql...
    Running C:\Tibero\tibero7\scripts\pkg\pkg_dbms_lob.sql...
```

```
Running C:\Tibero\tibero7\scripts\pkg\pkg_dbms_utility.sql...
Running C:\Tibero\tibero7\scripts\pkg\pkg_dbms_obfuscation.sql...

.....중간 생략.....

Creating dynamic performance view descriptions...
Running C:\Tibero\tibero7\scripts\dpv_dict.sql...
Creating spatial meta tables and views ...
Running C:\Tibero\tibero7\scripts\create_gis.sql...
Creating internal system tables...
Running C:\Tibero\tibero7\scripts\sys_tbl.sql...
```

참고

system.vbs 실행할 때 환경변수에 설정한 "%TB_SID%"를 가지고 DB에 접속하기 때문에**[제어판] > [관리도구] > [데이터 원본(ODBC)]**에 %TB_SID%와 같은 이름의 DSN이 존재하지 않는지 주의해야 한다. 이는 데이터 원본(ODBC)에 "%TB_SID%"와 같은 이름의 DSN이 존재할 경우 데이터 원본(ODBC)에 있는 DSN에 설정된 DB로 접속을 시도하여 system.vbs가 수행되기 때문이다. system.sh(.vbs)에서 제공되는 각 옵션에 대한 설명은 ["Appendix D. system.sh"](#)를 참고한다.

12 Tibero의 설치가 정상적으로 완료되면 tblastener, tbsvr 프로세스가 실행된다. 해당 프로세스는 Windows 작업 관리자의 **[프로세스]** 탭을 통해 확인할 수 있다.

이미지 이름	사용자 이름	CPU	메모리 사용
...			
tblastener.exe	SYSTEM	00	2,804 KB
tbsvr.exe	SYSTEM	00	216,056 KB
...			

3.2.2. UNIX 환경

다음은 UNIX 환경에서 수동으로 Tibero를 설치하는 순서에 대한 설명이다.

- 1. 다음의 두 가지 파일이 준비되었는지 확인한다.
 - 바이너리 실행 파일(tar.gz)
 - 라이선스 파일(license.xml)
- 2. 해당 운영체제의 사용자 계정별로 존재하는 환경설정 파일(.bashrc, .bash_profile, .profile 등)에 환경변수를 설정한다. 본 예제에서는 다음과 같이 환경변수를 설정한다.

```
export TB_HOME=/home/tibero/Tibero/tibero7
export TB_SID=tibero
```

```
export LD_LIBRARY_PATH=$TB_HOME/lib:$TB_HOME/client/lib
export PATH=$PATH:$TB_HOME/bin:$TB_HOME/client/bin
```

참고

1. 환경변수를 변경하는 방법은 OS에 따라 다르므로, 자세한 내용은 각 OS 안내서를 참고한다.
2. 각 환경변수 값에 대한 설명은 “3.3.2. 사용자 환경설정 파일의 적용 및 확인”을 참고한다.

3. **/home/tibero/Tibero** 디렉터리에서 압축된 바이너리 실행 파일(tar.gz)을 해제한다. 그 후에 발급받은 라이선스 파일(license.xml)을 **\$TB_HOME/license** 디렉터리에 복사한다.

4. **\$TB_HOME/config** 디렉터리에서 다음의 명령어를 입력한다.

```
gen_tip.sh
```

이 명령이 실행되면 환경 파일(.tip)과 tbdns.tbr, psm_commands 파일이 생성된다.

```
tibero@Tibero:~/Tibero/tibero7/config$ gen_tip.sh
Using TB_SID "tibero"
/home/tibero/Tibero/tibero7/config/tibero.tip generated
/home/tibero/Tibero/tibero7/config/psm_commands generated
/home/tibero/Tibero/tibero7/client/config/tbdns.tbr generated.
Running client/config/gen_esql_cfg.sh
Done.
```

5. Tibero 서버를 '**NOMOUNT 모드**'로 기동한다.

```
tibero@Tibero:~/Tibero/tibero7/bin$ tbboot nomount
Change core dump dir to /home/tibero7/tibero7/bin/prof.
Listener port = 8629

Tibero7

TmaxData Corporation Copyright (c) 2008-. All rights reserved.
Tibero instance started up (NOMOUNT mode).
```

6. tbSQL 유틸리티를 이용하여 데이터베이스에 접속한다. 본 예제에서는 '**sys**' 사용자로 접속한다.

```
tibero@Tibero:~/Tibero/tibero7/client/bin$ tbsql sys/tibero

tbSQL 7

TmaxData Corporation Copyright (c) 2008-. All rights reserved.

Connected to Tibero.
```

```
SQL>
```

7. **CREATE DATABASE** 문을 이용하여 원하는 데이터베이스를 생성한다. 본 예제에서는 **'tibero'**로 생성한다.

```
SQL> create database "tibero"
      user sys identified by tibero
      maxinstances 8
      maxdatafiles 100
      character set MSWIN949
      national character set UTF16
      logfile
        group 1 'log001.log' size 50M,
        group 2 'log002.log' size 50M,
        group 3 'log003.log' size 50M
      maxloggroups 255
      maxlogmembers 8
      noarchivelog
      datafile 'system001.dtf' size 100M autoextend on next 10M maxsize unlimited
      default temporary tablespace TEMP
        tempfile 'temp001.dtf' size 100M autoextend on next 10M maxsize unlimited
        extent management local autoallocate
      undo tablespace UNDO
        datafile 'undo001.dtf' size 200M autoextend on next 10M maxsize unlimited
        extent management local autoallocate
      SYSSUB
        datafile 'syssub001.dtf' size 10M autoextend on next 10M maxsize unlimited
      default tablespace USR
        datafile 'usr001.dtf' size 100M autoextend on next 10M maxsize unlimited
        extent management local autoallocate;
Database created.

SQL> quit
Disconnected.
```

해당 과정에서 설정하는 주요 항목에 대한 설명은 다음과 같다.

항목	설명
maxdatafiles	데이터베이스에서 사용할 데이터 파일의 최대값을 설정한다. (기본값: 100)
character set	데이터베이스의 문자 집합을 설정한다.
maxloggroups	로그 그룹의 최대값을 설정한다. (기본값: 255)
maxlogmembers	로그 그룹 내의 로그 파일의 최대값을 설정한다. (기본값: 8)

참고

CREATE DATABASE 문에 대한 자세한 내용은 "Tibero SQL 참조 안내서"를 참고한다.

8. 데이터베이스 생성이 완료되면 **tbboot** 명령어로 Tibero를 다시 기동한다.

```
tibero@Tibero:~/Tibero/tibero7/bin$ tbboot
Change core dump dir to /home/tibero7/tibero7/bin/prof.
Listener port = 25010

Tibero 7

TmaxData Corporation Copyright (c) 2008-. All rights reserved.
Tibero instance started up (NORMAL mode).
```

9. **\$TB_HOME/scripts** 디렉터리에서 **system_install.sh** 셸을 실행한다. 이 셸을 실행하면 현재 디렉터리에 있는 SQL 파일이 수행된다. SQL 파일을 수행하면 role, system user, view, package 등이 생성된다. 사용되는 sys 및 syscat 계정에 대한 기본 암호는 각각 tibero, syscat이다.

```
tibero@Tibero:~/Tibero/tibero7/scripts$ system_install.sh
Enter SYS password:

Enter SYSCAT password:

Creating the role DBA...
create default system users & roles?(Y/N):

Creating prerequisite objects...
Creating internal system tables...
Creating additional system index...
Creating client policy table ...
Setting case insensitive password...
Creating text packages table ...
Creating the role DBA...
Creating system users & roles...
Creating example users...
Creating virtual tables...
Granting public access to _VT_DUAL...
Creating the system generated sequences...
Creating internal dynamic performance views...
Creating outline table...
Creating system tables related to dbms_job...
Creating system tables related to dbms_lock...
Creating system tables related to dbms_parallel_execute...
Creating system tables related to scheduler...
Creating system tables related to server_alert...
Creating system tables related to tpm...
```

```

Creating system tables related to tsn and timestamp...
Creating system tables related to rsrc...

.....중간 생략.....

Running /tibero/74/scripts/pkg/_pkg_sa_user_admin.tbw...
Running /tibero/74/scripts/label_security_packages.sql...
Running /tibero/74/scripts/label_security_synonym.sql...
Setting system.sh parameter...
Done.
For details, check /tibero/74/instance/tibero/log/system_init.log.
For details, check /home/tibero7/tibero7/instance/tibero/log/system_init.log.

```

참고

자세한 system_install.sh에 관한 설명은 “[Appendix H. system_install.sh](#)”를 참고한다.

system.sh(.vbs)는 v7.2.4에서 deprecated 될 예정이다.

- 10 Tibero의 설치가 정상적으로 완료되면 Tibero 프로세스가 실행된다. 이 프로세스는 다음의 명령어를 실행하면 확인할 수 있다.

```

tibero@Tibero:~/Tibero/tibero7/scripts$ ps -ef | grep tbsvr
tibero  19981      1  0 21:12 pts/2    00:00:00 tbsvr      ...
tibero  19983 19981  0 21:12 pts/2    00:00:00 tbsvr_TBMP  ...
tibero  19984 19981  0 21:12 pts/2    00:00:00 tbsvr_WP000 ...
tibero  19985 19981  3 21:12 pts/2    00:00:00 tbsvr_WP001 ...
tibero  19986 19981  1 21:12 pts/2    00:00:12 tbsvr_WP002 ...
tibero  19987 19981  2 21:12 pts/2    00:00:12 tbsvr_PEP000 ...
tibero  19988 19981  0 21:12 pts/2    00:00:00 tbsvr_AGNT  ...
tibero  19989 19981  1 21:12 pts/2    00:00:00 tbsvr_DBWR  ...
tibero  19999 19981  0 21:12 pts/2    00:00:00 tbsvr_RECO  ...

```

3.3. 설치 확인

본 절에서는 Tibero가 설치된 후 정상적인 설치 여부를 확인하는 방법에 대해서 설명한다.

3.3.1. 디렉터리 구조

Tibero가 설치되면 다음과 같은 디렉터리가 생성된다.

```

$TB_HOME
+- bin
|   |
|   +- update

```

```

|
+- client
|   |
|   +- bin
|   +- config
|   +- include
|   +- lib
|   |   |
|   |   +- jar
|   |   +- php
|   +- ssl
|   |   |
|   |   +- misc
|   +- epa
|       |
|       +- java
|           |
|           +- config
|           +- lib
|
+- config
|
+- database
|   +- $TB_SID
|       |
|       +- java
|
+- instance
|   |
|   +- $TB_SID
|       |
|       +- out
|           |   |
|           |   +- svr
|           |   +- lsnr
|           |
|           +- dump
|               |   |
|               |   +- act
|               |   +- callstack
|               |   +- diag
|               |   +- report
|               |   +- tracedump
|               |
|               +- log
|                   |   |
|                   |   +- act

```

```

|      | +- audit
|      | +- diag
|      | +- dlog
|      | +- ilog
|      | +- lsnr
|      | +- slog
|      | +- sqltrace
|      | +- memlog
|      |
|      +- path
|
+- lib
|
+- license
| |
| +- oss_licenses
|
+- nls
| |
| +- zoneinfo
|
+- scripts
  |
  +- pkg

```

위의 디렉터리 구조에서 \$TB_SID라고 보이는 부분은 각각의 시스템 환경에 맞는 서버의 SID로 바뀌서 읽어야 한다.

Tibero에서 사용하는 기본 디렉터리는 다음과 같다.

bin

Tibero의 실행 파일과 서버 관리를 위한 유틸리티가 위치한 디렉터리이다. 이 디렉터리에 속한 파일 중에서 tbsvr과 tblastener는 Tibero를 구성하는 실행 파일이며, tbboot와 tbdwn은 각각 Tibero를 기동하고 종료하는 역할을 담당한다. tbsvr과 tblastener 실행 파일은 반드시 tbboot 명령어를 이용하여 실행되어야 하며, 절대로 직접 실행해서는 안 된다.

client

다음은 하위 디렉터리에 대한 설명이다.

하위 디렉터리	설명
bin	Tibero의 클라이언트 실행 파일이 있는 디렉터리이다. 이 디렉터리에는 다음과 같은 유틸리티가 있다. - tbSQL : 기본적인 클라이언트 프로그램으로 사용자가 직접 SQL 질의를 하고 그 결과를 확인할 수 있다.

하위 디렉터리	설명
	- T-Up : 다른 데이터베이스에서 Tibero로의 호환성 평가와 마이그레이션을 지원한다. - tbExport : 논리적 백업이나 데이터베이스 간에 데이터 이동을 위해 데이터 베이스의 내용을 외부 파일로 저장한다. - tbImport : 외부 파일에 저장된 내용을 데이터베이스로 가져온다. - tbLoader : 대량의 데이터를 데이터베이스로 한꺼번에 읽어 들인다. - tbpc : C 언어로 작성된 프로그램 안에서 내장 SQL(Embedded SQL)을 사용하는 프로그램을 개발할 때 이를 C 프로그램으로 변환한다. 이렇게 변환된 프로그램을 C 컴파일러를 통해 컴파일할 수 있도록 도와주는 역할도 담당한다. 유틸리티에 대한 내용은 "Tibero 유틸리티 안내서"를 참고한다. 단, tbpc 유틸리티는 "Tibero tbESQL/C 안내서"를 참고한다.
config	Tibero의 클라이언트 프로그램을 실행하기 위한 설정 파일이 위치하는 디렉터리이다.
include	Tibero의 클라이언트 프로그램을 작성할 때 필요한 헤더 파일이 위치하는 디렉터리이다.
lib	Tibero의 클라이언트 프로그램을 작성할 때 필요한 라이브러리 파일이 위치하는 디렉터리이다. 자세한 내용은 "Tibero 애플리케이션 개발자 안내서"와 "TiberotbESQL/C 안내서"를 참고한다.
ssl	서버 보안을 위한 인증서와 개인 키를 저장하는 디렉터리이다.
epa	External Procedure와 관련된 설정 파일과 로그 파일이 있는 디렉터리이다. 이에 대한 자세한 내용은 "Tibero External Procedure 안내서"를 참고한다.

config

Tibero의 환경설정 파일이 위치하는 디렉터리이다. 이 위치에 존재하는 \$TB_SID.tip 파일이 Tibero의 환경설정을 결정한다.

database

다음은 하위 디렉터리에 대한 설명이다.

- \$TB_SID

Tibero의 데이터베이스 정보를 별도로 설정하지 않는 한 모든 데이터베이스 정보가 이 디렉터리와 그 하위 디렉터리에 저장된다. 이 디렉터리에는 데이터 자체에 대한 메타데이터(metadata)뿐만 아니라 다음과 같은 종류의 파일이 있다.

파일	설명
컨트롤 파일	다른 모든 파일의 위치를 담고 있는 파일이다.
데이터 파일	실제 데이터를 저장하고 있는 파일이다.
로그 파일	데이터 복구를 위해 데이터에 대한 모든 변경 사항을 저장하는 파일이다.

– \$TB_SID/java

JAVA_CLASS_PATH가 정의되지 않은 경우 Java EPA Class File이 저장되는 디렉터리이다.

instance

다음은 하위 디렉터리에 대한 설명이다.

– \$TB_SID/out

Tibero의 프로세스에서 로그를 남길 수 없는 특정 상황에 현재 상황에 대한 정보를 남기는 파일들이 남는 디렉터리이다. 해당 파일에 남는 내용은 매우 중요한 정보이기에 Tibero가 운영 중일 때 이 위치에 존재하는 파일을 읽거나 수정하지 않는 것을 권장한다. 리스너 프로세스의 것은 하위 디렉터리 "/lsnr"에, 일반 프로세스의 것은 "/svr"에 남는다.

– \$TB_SID/log

Tibero의 시스템 로그 파일(slog)과 DBMS 로그(dlog), Internal 로그(ilog), Listener 로그(lsnr), 메모리 로그 등의 로그 파일이 저장되는 디렉터리이다.

파일	설명
Activity 로그 파일(act)	서버 내 액티비티 모니터에서 타 스레드들의 동작을 점검하던 도중, 임의의 스레드가 장기간 멈춰 있음을 감지하였을 때 남기는 로그 파일이다. 장기간 동작하지 않는 스레드의 원인 분석에 사용될 수 있다.
AUDIT 로그 파일(audit)	데이터베이스 사용자가 시스템 특권 또는 스키마 객체 특권을 사용하는 것을 감시(AUDIT)한 내용을 기록한 파일이다.
Diagnostic 로그 파일(diag)	TAC의 diag 기능을 사용하는 경우, 글로벌 노드 간 지연 현상이 감지되었을 때 문제 현상에 대한 대략적인 정보를 남기는 로그 파일이다. 글로벌 노드 상에서 발생한 지연 현상의 원인을 분석하는데에 사용될 수 있다.
DBMS 로그 파일(dlog)	시스템 로그 파일에 기록되는 정보보다 좀 더 중요한 정보가 기록되는 파일이며, 서버 기동 및 종료, DDL 문장의 수행 등이 기록되는 파일이다.
Internal 로그 파일(ilog)	스레드별로 설정된 이벤트에 대한 시스템 로그가 기록되는 파일이며, Internal 로그를 보려면 tbiv를 이용해야 한다.
Listener 로그 파일(lsnr)	Listener의 디버깅을 위한 파일이다. 리스너에서 일어난 중요한 일이 기록되는 파일이며, 리스너의 버그를 해결하는 데 사용될 수 있다.
시스템 로그 파일(slog)	디버깅을 위한 파일이다. 서버가 하는 중요한 일이 기록되는 파일이며, 서버 성능이 저하되는 원인을 찾거나 Tibero 자체의 버그를 해결하는 데 사용될 수 있다.

파일	설명
SQLTRACE 로그 파일(sql trace)	세션에서 수행된 SQL들에 대한 정보를 남기는 로그 파일이다. SQL의 성능을 분석하거나 버그의 원인을 파악하는데에 사용될 수 있다.
Memory 로그 파일(memlog)	임의의 세션이 부득이한 문제로 강제 종료되는 경우, 해당 세션에서 남길 예정이었던 시스템 로그들을 남기는 파일이다. 문제가 발생한 세션을 분석하는데에 사용될 수 있다.

시스템 로그 파일, DBMS 로그 파일, Internal 로그 파일, Listener 로그 파일을 비롯한 로그 파일들은 데이터베이스를 사용할수록 계속 누적되어 저장된다. 지속되는 누적으로 인한 디스크 포화 상태를 방지하고, 성능 향상 및 최적화된 사용자 경험을 제공하기 위해 각 파일의 최대 크기와 전체 디렉터리의 최대 크기를 파라미터로 조정할 수 있도록 되어있다.

예외적으로, SQLTRACE 로그 파일과 Memory 로그 파일들의 경우는 해당 로그들의 취지에 맞게 각 파일의 크기를 제한하는 것이 아니라 디렉터리 내 존재하는 파일들의 크기의 합을 파라미터를 통해 제한할 수 있다. 해당 값을 넘어가는 경우 파일들을 별도의 하위 디렉터리에 보관하고, 이 하위 디렉터리의 최대 개수를 파라미터로 제한하여 최댓값을 넘는 경우 오래된 디렉터리를 삭제하도록 설정할 수 있다.

로그 파일을 설정하는 초기화 파라미터는 다음과 같다.

초기화 파라미터	설명
AUDIT_LOG_FILE_SIZE	Audit 로그 파일 하나의 최대 크기를 설정한다.
AUDIT_LOG_TOTAL_SIZE_LIMIT	Audit 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
ACT_LOG_FILE_SIZE	Activity 로그 파일 하나의 최대 크기를 설정한다.
ACT_LOG_TOTAL_SIZE_LIMIT	Activity 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
DIAG_LOG_FILE_SIZE	Diagnostic 로그 파일 하나의 최대 크기를 설정한다.
DIAG_LOG_TOTAL_SIZE_LIMIT	Diagnostic 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
DBMS_LOG_FILE_SIZE	DBMS 로그 파일 하나의 최대 크기를 설정한다.
DBMS_LOG_TOTAL_SIZE_LIMIT	DBMS 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
ILOG_FILE_SIZE	Internal 로그 파일 하나의 최대 크기를 설정한다.
ILOG_TOTAL_SIZE_LIMIT	Internal 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
LSNR_LOG_FILE_SIZE	Listener 로그 파일 하나의 최대 크기를 설정한다.
LSNR_LOG_TOTAL_SIZE_LIMIT	Listener 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
SLOG_FILE_SIZE	시스템 로그 파일 하나의 최대 크기를 설정한다.
SLOG_TOTAL_SIZE_LIMIT	시스템 로그 파일이 저장된 디렉터리의 최대 크기를 설정한다.
SQL_TRACE_TOTAL_SIZE_LIMIT	SQL_TRACE_DEST 내에 존재하는 파일들 크기의 합의 상한을 설정한다. (하위 디렉터리의 크기는 제외)

초기화 파라미터	설명
SQL_TRACE_BACKUP_LIMIT_CNT	SQLTRACE 로그 파일들이 보관되어있는 하위 디렉터리의 최대 개수를 설정한다. 값을 넘는 경우, 가장 오래된 하위 디렉터리부터 디렉터리 짝로 삭제된다.
MEM_LOG_TOTAL_SIZE_LIMIT	MEM_LOG_DEST 내에 존재하는 파일들 크기의 합을 상한을 설정한다. (하위 디렉터리의 크기는 제외)
MEM_LOG_BACKUP_LIMIT_CNT	Memory 로그 파일들이 보관되어있는 하위 디렉터리의 최대 개수를 설정한다. 값을 넘는 경우, 가장 오래된 하위 디렉터리부터 디렉터리 짝로 삭제된다.

– \$TB_SID/dump

Tibero의 DDL 또는 실행 중 에러에 의해 발생하는 덤프 파일이 저장되는 디렉터리이다.

하위 디렉터리	설명
act	서버 내 액티비티 모니터에서 타 스레드들의 동작을 점검하던 도중, 임의의 스레드가 장기간 멈춰 있음을 감지하였을 때, 남기는 덤프 파일들의 디렉터리이다. 덤프 파일 내에는 대상 스레드의 함수 호출 스택 정보가 남는다.
callstack	DDL dump 명령 혹은 서버 내 요청으로 인한 콜스택 덤프 시, tstack 스레드에서 대상 스레드들의 함수 호출 스택 정보를 파일 형태로 남기는 디렉터리이다.
diag	TAC의 diag 기능을 사용하는 경우, 글로벌 노드 간 지연 현상이 감지되었을 때 문제 현상에 대한 자세한 정보를 남기는 덤프 파일들의 디렉터리이다. 덤프 파일들에는 관련 스레드들의 함수 호출 스택 정보, 관련 스레드들의 자원 획득 및 경쟁에 대한 정보들이 남는다.
report	DB 성능 측정을 위해 DDL 혹은 TPR 패키지를 이용하여 TPR report를 생성하였을 때 만들어 지는 파일들의 디렉터리이다.
tracedump	DB 내 문제 상황에 대한 다양한 디버깅 정보들을 가진 덤프 파일들의 디렉터리이다. 문제가 발생하는 동작에서 저절로 덤프가 남는 경우도 있고, DDL dump 명령을 통해 남는 경우도 있다.

Activity 덤프, Diagnostic 덤프, Callstack 덤프, Tracedump 덤프 파일의 경우 데이터베이스를 사용할수록 계속 누적되어 저장될 수 있다. Tibero는 성능 향상 및 최적화된 사용자 경험을 제공하고, 지속되는 덤프 파일 누적으로 인한 디스크 포화 상태를 예방하기 위해 디렉터리 내 존재하는 파일들의 크기의 합을 파라미터를 통해 제한할 수 있다. 해당 값을 넘어가는 경우 파일들을 별도의 하위 디렉터리에 보관하며, 이와 더불어 별도의 파라미터를 통해 하위 디렉터리들의 최대 개수를 제한하여 최댓값을 넘는 경우 오래된 디렉터리를 삭제하도록 설정할 수 있다. (TPR의 경우는 사용자 요청으로만 파일이 생성되기 때문에 자동으로 관리하지 않음)

제어 파라미터	설명
ACT_DUMP_TOTAL_SIZE_LIMIT	ACT_DUMP_DEST 내에 존재하는 파일들 크기의 합을 상한을 설정한다. (하위 디렉터리의 크기는 제외)

제어 파라미터	설명
ACT_DUMP_BACKUP_LIMIT_CNT	Activity 덤프 파일들이 보관되어있는 하위 디렉터리의 최대 개수를 설정한다. 값을 넘는 경우, 가장 오래된 하위 디렉터리부터 디렉터리 째로 삭제된다.
CALLSTACK_DUMP_TOTAL_SIZE_LIMIT	CALLSTACK_DUMP_DEST 내에 존재하는 파일들 크기의 합을 상한을 설정한다. (하위 디렉터리의 크기는 제외)
CALLSTACK_DUMP_BACKUP_LIMIT_CNT	Callstack 덤프 파일들이 보관되어있는 하위 디렉터리의 최대 개수를 설정한다. 값을 넘는 경우, 가장 오래된 하위 디렉터리부터 디렉터리 째로 삭제된다.
DIAG_DUMP_TOTAL_SIZE_LIMIT	DIAG_DUMP_DEST 내에 존재하는 파일들 크기의 합을 상한을 설정한다. (하위 디렉터리의 크기는 제외)
DIAG_DUMP_BACKUP_LIMIT_CNT	Diagnostic 덤프 파일들이 보관되어있는 하위 디렉터리의 최대 개수를 설정한다. 값을 넘는 경우, 가장 오래된 하위 디렉터리부터 디렉터리 째로 삭제된다.
TRACE_DUMP_TOTAL_SIZE_LIMIT	TRACE_DUMP_DEST 내에 존재하는 파일들 크기의 합을 상한을 설정한다. (하위 디렉터리의 크기는 제외)
TRACE_DUMP_BACKUP_LIMIT_CNT	TRACEDUMP 덤프 파일들이 보관되어있는 하위 디렉터리의 최대 개수를 설정한다. 값을 넘는 경우, 가장 오래된 하위 디렉터리부터 디렉터리 째로 삭제된다.

– \$TB_SID/path

Tibero의 프로세스 간에 통신을 위한 소켓 파일이 있는 디렉터리이다. Tibero가 운영 중일 때 이 위치에 존재하는 파일을 읽거나 수정해서는 안 된다.

lib

Tibero 서버에서 Spatial과 관련된 함수를 사용하기 위한 라이브러리 파일이 있는 디렉터리이다.

license

Tibero의 라이선스 파일(license.xml)이 있는 디렉터리이다. XML 형식이므로 일반 텍스트 편집기로도 라이선스의 내용을 확인할 수 있다.

다음은 하위 디렉터리에 대한 설명이다.

하위 디렉터리	설명
oss_licenses	반드시 준수해야 하는 오픈소스 라이선스의 대한 정보를 확인할 수 있는 디렉터리이다.

nls

다음은 하위 디렉터리에 대한 설명이다.

하위 디렉터리	설명
zoneinfo	Tibero에서 사용하는 시간대 파일이 있는 디렉터리이다.

scripts

Tibero의 데이터베이스를 생성할 때 사용하는 각종 SQL 문장이 있는 디렉터리이다. 또한 Tibero의 현재 상태를 보여주는 각종 뷰의 정의도 이 디렉터리에 있다.

다음은 하위 디렉터리에 대한 설명이다.

하위 디렉터리	설명
pkg	Tibero에서 사용하는 패키지의 생성문이 저장되는 디렉터리이다.

3.3.2. 사용자 환경설정 파일의 적용 및 확인

Tibero가 설치된 후에 사용자의 환경설정 파일에는 다음의 내용이 추가된다.

환경변수	설명
TB_HOME	Tibero가 설치된 디렉터리이다. 만약, Tibero를 설치하는 과정에서 설치 디렉터리를 입력하지 않았다면 기본적으로 {사용자의 홈 디렉터리}/Tibero/tibero7가 TB_HOME으로 설정된다.
TB_SID	설치 과정 중에 입력한 SID가 여기에 해당한다. 설치 과정 중에 입력하지 않았다면 기본적으로 tibero 가 SID로 설정된다.
LD_LIBRARY_PATH	Tibero를 사용할 때 필요한 공유 라이브러리가 위치한 경로이다. 필요한 라이브러리는 모두 \$TB_HOME/lib와 \$TB_HOME/client/lib 안에 있고, OS별로 환경변수가 다르게 지정된다. – SunOS, Linux : LD_LIBRARY_PATH – AIX : LIBPATH
PATH	Tibero를 사용하기 위한 디렉터리 경로이다. 기본적으로 \$TB_HOME/bin과 \$TB_HOME/client/bin이 설정된다.

UNIX에서는 비정상적으로 프로세스를 종료할 경우 가상 메모리 정보를 디스크로 남긴다. (Core Dump)

이 과정에서 많은 컴퓨팅 자원을 요구하여 시스템 성능을 순간적으로 저하시키며, 이때 생성되는 Core 파일은 많은 디스크 용량을 필요로 한다. 따라서 Tibero에서는 이 기능을 사용하지 않음을 권장하지 않는다.

Core Dump 기능을 사용하지 않으려면 사용자 계정의 환경설정 파일(.profile)에 다음의 명령어를 추가한다.

```
ulimit -c 0
```

위와 같이 사용자의 환경설정 파일이 생성되어도 실제로 적용된 것은 아니다. 환경설정 파일을 적용하기 위해서는 다음과 같은 과정을 거쳐야 한다.

1. 다음의 명령어를 입력하여 사용자 환경설정 파일을 적용한다. UNIX 계열(LINUX 포함)에서만 아래 명령어를 입력한다.

```
sh .profile
```

2. 환경설정 파일이 제대로 적용되었는지 확인하려면 다음의 명령어를 입력한다.

– Linux 계열의 경우

```
echo $TB_HOME
```

– Windows 계열의 경우

```
echo %TB_HOME%
```

3.3.3. 기동 및 종료

Tibero의 기동과 종료는 데이터베이스 관리자(DBA: Database Administrator, 이하 DBA)만 할 수 있다.

기동

Tibero의 기동은 **tbboot** 명령어를 사용한다. 옵션을 포함한 **tbboot**의 명령어는 다음과 같다.

```
tbboot [-h] [-v] [-l] [-C] [-c] [-t BOOTMODE]
```

옵션	설명
-h	tbboot 명령어의 도움말을 보여주는 옵션이다.
-v	Tibero의 버전 정보를 보여주는 옵션이다.
-l	Tibero의 라이선스 정보를 보여주는 옵션이다.
-C	Tibero가 지원하는 문자 집합의 정보를 보여주는 옵션이다.
-c	Tibero가 replication mode로 설정되어 있을 경우 replication mode를 사용하지 않는 옵션이다.
-t BOOTMODE	Tibero 서버를 기동할 수 있는 옵션이다. 이 옵션은 생략이 가능하다. Tibero에서는 tbboot 명령어에서 부트 모드(bootmode)를 제공한다. – NORMAL : 정상적으로 데이터베이스의 모든 기능을 사용할 수 있는 모드이다. – NOMOUNT : Tibero의 프로세스만 기동시키는 모드이다. – MOUNT : 미디어 복구 등의 데이터베이스 관리를 위해 사용하는 모드이다. – RECOVERY : Tibero Standby Cluster를 구축할 때 Standby 쪽의 데이터베이스를 운영하는 모드이다. – RESETLOGS : Tibero 서버를 기동하는 과정에서 로그 파일을 초기화하며, 불완전 미디어 복구 이후에 사용하는 모드이다. – FAILOVER : Tibero Standby Cluster 환경에서 Standby를 Primary로 사용하기 위한 모드이다. – READONLY : 데이터베이스를 읽는 작업만 허용하고, 변경 작업을 허용하지 않는 모드이다.

종료

Tibero의 종료는 **tbdwn** 명령어를 사용한다. 옵션을 포함한 **tbdwn**의 명령어는 다음과 같다.

```
tbdwn [-h] [-t DOWNMODE]
```

옵션	설명
-h	tbdwn 명령어의 도움말을 보여주는 옵션이다.
-t DOWNMODE	Tibero 서버를 종료할 수 있는 옵션이다. 이 옵션은 생략이 가능하다. Tibero에서는 tbdwn 명령어에서 다운 모드(downmode)를 제공한다. - NORMAL : 일반적인 종료 모드이다. - POST_TX : 모든 트랜잭션이 끝날 때까지 대기한 후 Tibero를 종료하는 모드이다. - IMMEDIATE : 현재 수행 중인 모든 작업을 강제로 중단시키며 진행 중인 모든 트랜잭션을 롤백하고 Tibero를 종료하는 모드이다. - ABORT : Tibero의 프로세스를 강제로 종료하는 모드이다. - SWITCHOVER : Standby DB와 Primary DB를 동기화시킨 후 Primary DB를 NORMAL 모드처럼 종료하는 모드이다. - ABNORMAL : Tibero 서버에 접속하지 않고 서버 프로세스를 무조건 강제로 종료시키는 모드이다.

tbSQL 유틸리티를 이용한 데이터베이스 접속

tbsql 명령으로 데이터베이스에 접속한다. 다음과 같이 데이터베이스 사용자의 ID와 패스워드를 함께 입력하여 접속할 수 있다.

```
tbsql sys/tibero
```

리스너를 통한 접속을 위해서는 다음과 같이 데이터베이스 사용자 ID와 패스워드 뒤에 @을 입력하고 @ 뒤에는 **tbdsn.tbr** 파일에 지정된 별칭을 입력한다.

```
tbsql sys/tibero@tibero
```

3.3.4. 사용자 계정

Tibero가 설치되면 아래와 같은 사용자 계정이 자동으로 생성된다.

계정	설명
SYS	데이터베이스 관리를 위한 계정으로 시스템 패키지, 동의어, 사용자, 역할, 가상 테이블, 시퀀스, 동적 뷰 등을 생성하고 관리한다.
SYSCAT	데이터베이스 관리를 위한 정적 카탈로그 뷰를 생성하고 관리하는 계정이다.
SYSGIS	GIS(Geographic Information System)와 관련된 테이블 생성 및 관리를 하는 계정이다.
OUTLN	동일한 SQL을 수행할 때 항상 같은 질의 플랜(plan)으로 수행될 수 있게 관련 힌트(hint)를 저장하는 등의 일을 하는 계정이다.
TIBERO, TIBERO1	CONNECT, RESOURCE, DBA 역할이 부여된 샘플 사용자 계정이다.

제4장 데이터베이스 제거

본 장에서는 설치된 Tibero를 제거하는 방법에 대해서 설명한다.

4.1. 개요

설치된 Tibero를 제거하는 방법은 다음과 같다.

- 수동 제거

모든 플랫폼에서 \$TB_HOME/.installation/Tibero_Uninstaller를 실행한다.

4.2. 수동 제거

본 절에서는 Windows 환경과 UNIX 환경에서 수동 모드로 Tibero를 제거하는 순서를 설명한다.

4.2.1. Windows 환경

다음은 Windows 환경에서 수동으로 Tibero를 제거하는 과정에 대한 설명이다. Windows 환경에서의 제거 과정은 반드시 관리자 권한으로 진행되어야 한다.

1. Tibero를 종료한다.

```
tbdown
```

2. %TB_HOME%\bin 디렉터리에서 다음의 명령어를 입력한다.

```
tbuninstall %TB_SID%
```

이 명령이 실행되면 Windows 서비스에 등록된 Tibero가 제거된다.

```
C:\Tibero\tibero7\bin>tbuninstall %TB_SID%
Microsoft (R) Windows Script Host 버전 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

TB_HOME = C:\Tibero\tibero7
TB_SID = tibero
service name = Tibero_tibero
Tibero_tibero uninstalled successfully.
```

3. Tiberο가 설치된 디렉터리를 포함한 모든 하위 디렉터리를 해당 운영체제의 명령어로 삭제한다.

```
rmdir %TB_HOME%
```

4.2.2. UNIX 환경

다음은 UNIX 환경에서 수동으로 Tiberο를 제거하는 순서에 대한 설명이다.

1. Tiberο를 종료한다.

```
tbdownd
```

2. Tiberο가 설치된 디렉터리를 포함한 모든 하위 디렉터리를 해당 운영체제의 명령어로 삭제한다.

```
rm -rf $TB_HOME
```

제5장 클라이언트 설치와 제거

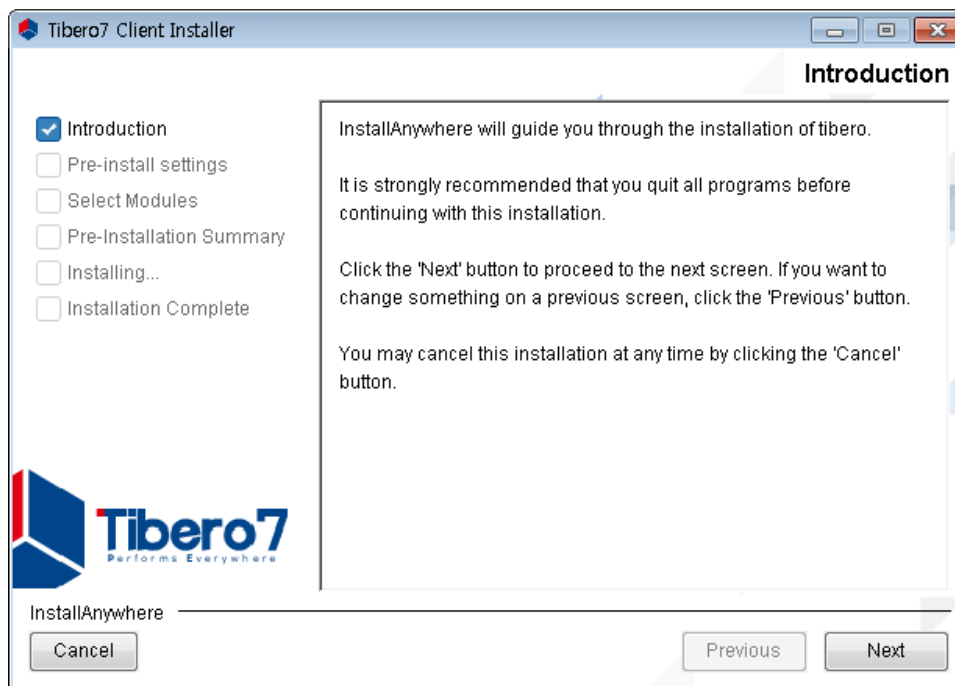
본 절에서는 Windows 환경에 대하여 인스톨러를 이용한 클라이언트 설치 방법에 대하여 설명한다.

5.1. 설치

Windows 환경에서 Tiberio 클라이언트를 설치하는 순서는 다음과 같다.

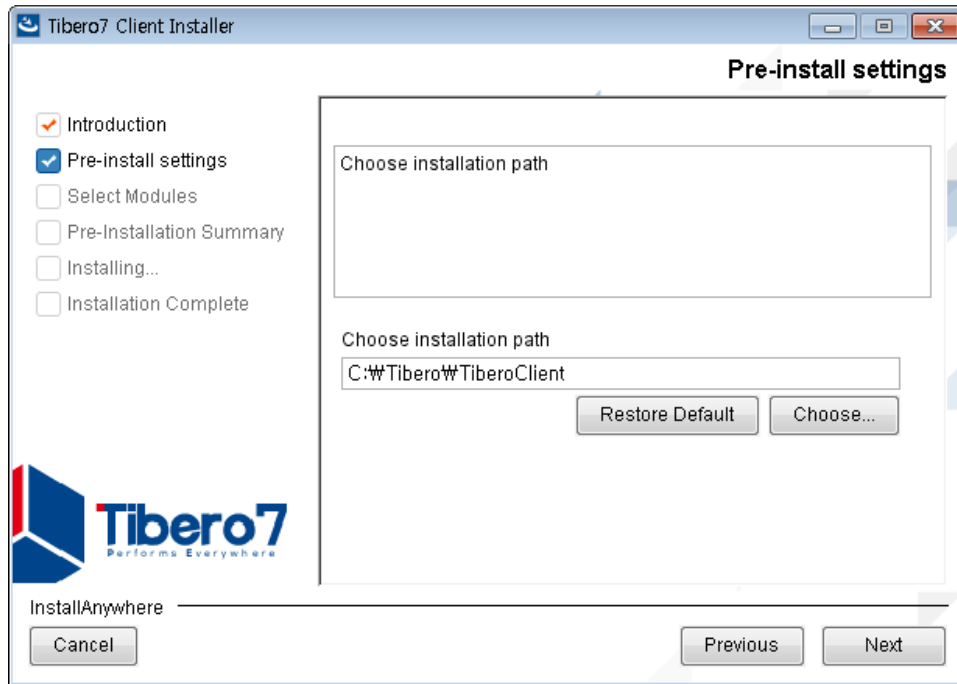
1. tiberio7_<fixset>_windows_<bit>_client.exe 파일을 더블 클릭하면 클라이언트 인스톨러가 아래와 같이 실행된다. **[Next]** 버튼을 클릭하여 클라이언트 설치를 진행한다. 만약 설치 과정을 취소하려면 **[Cancel]** 버튼을 클릭한다.

[그림 5.1] Tiberio Client Installer - Introduction



2. Tiberio 클라이언트를 설치할 디렉터리를 선택하는 화면이 나타난다. 기본 디렉터리로 'C:\Tiberio\TiberioClient'가 설정되어 있다. 설치 경로를 변경하려면 **[Choose]** 버튼을 클릭하여 Tiberio 클라이언트가 설치될 경로를 선택한다. 만약, 기본 디렉터리로 설정하려면 **[Restore Default]** 버튼을 클릭한다. 설치할 경로가 정해지면 **[Next]** 버튼을 클릭한다.

[그림 5.2] Tiberio Client Installer - Pre-Install settings

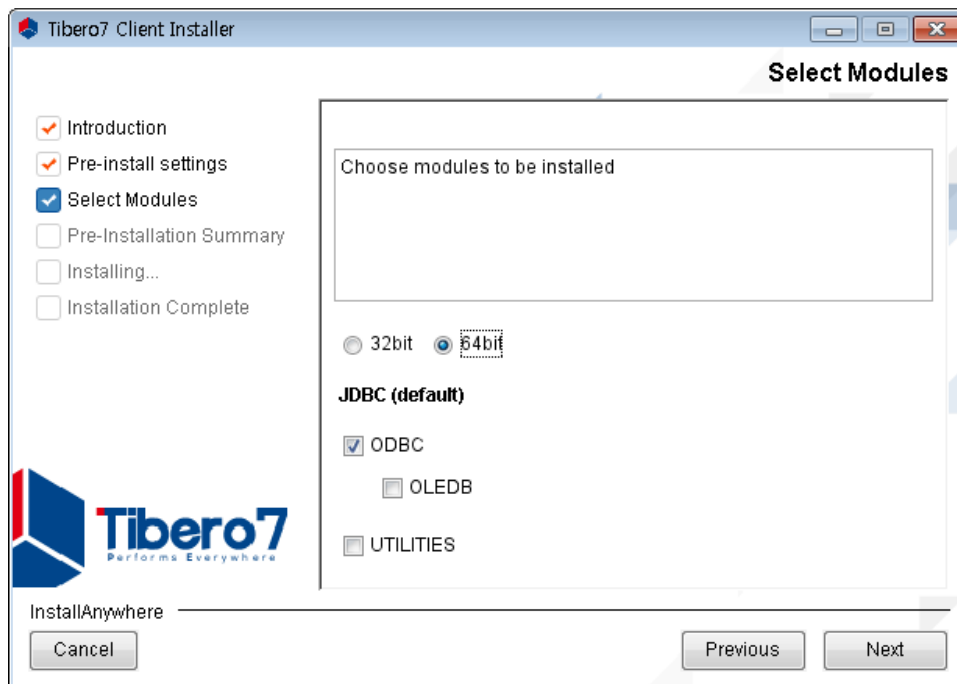


참고

기존의 파일 경로를 선택해서 덮어쓰게 되면 기존 설치된 버전과의 차이로 인하여 예상치 못한 에러가 발생할 수 있으므로 주의해야 한다.

3. Tiberio 클라이언트 인스톨러를 통하여 설치할 구성요소를 선택한다.

[그림 5.3] Tiberio Client Installer - Select Modules



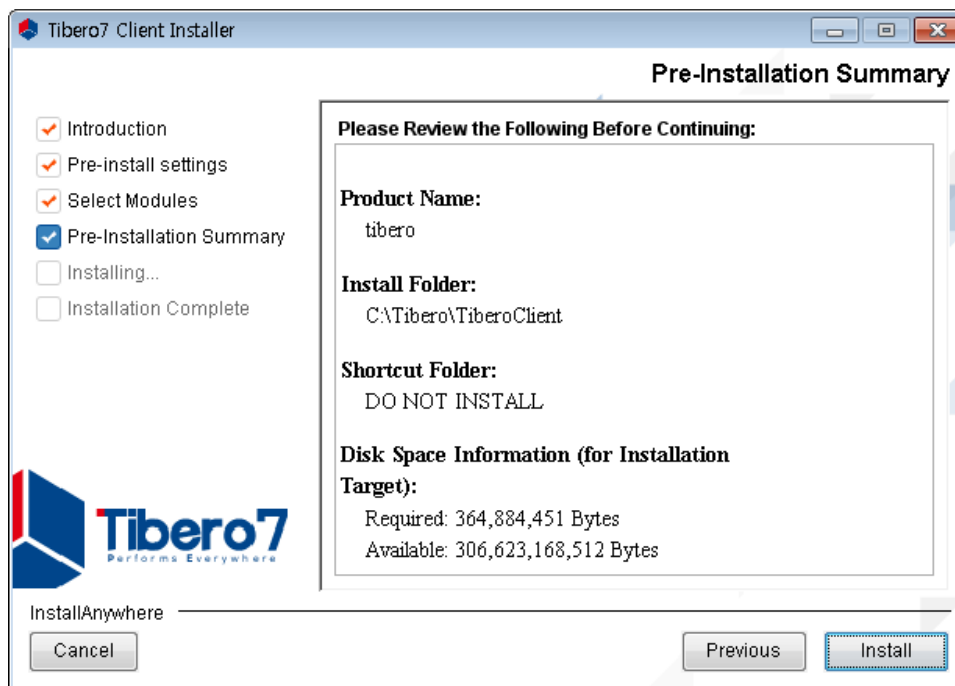
JDBC Driver의 경우 기본으로 설치가 진행이 되며, 각 옵션에서 선택 가능한 항목은 다음과 같다.

옵션	설명
Bit	설치할 클라이언트 라이브러리의 비트 수를 선택한다.
ODBC	ODBC 데이터 원본 관리자에 Tibero ODBC Driver 공급자를 등록하기 위한 비트를 선택하여 설치한다.
OLE DB	OLE DB 모듈을 설치한다. OLE DB 모듈의 경우 반드시 ODBC 모듈과 별개로 설치할 수 없고, ODBC도 같이 선택하여 설치를 진행해야 한다.
UTILITIES	tbSQL, tbLoader 등과 같은 클라이언트 유틸리티를 설치한다.

4. 설치 시작부터 지금까지 설정한 정보를 확인할 수 있도록 요약 정보를 보여준다.

설치 경로 등 설정이 올바른지 확인하고 Tibero 클라이언트 설치를 진행하기 위해 **[Install]** 버튼을 클릭한다. 만약 설정한 설치 정보를 수정하려면 **[Previous]** 버튼을 클릭하여 이전 화면으로 이동한 후 설정된 내용을 수정한다.

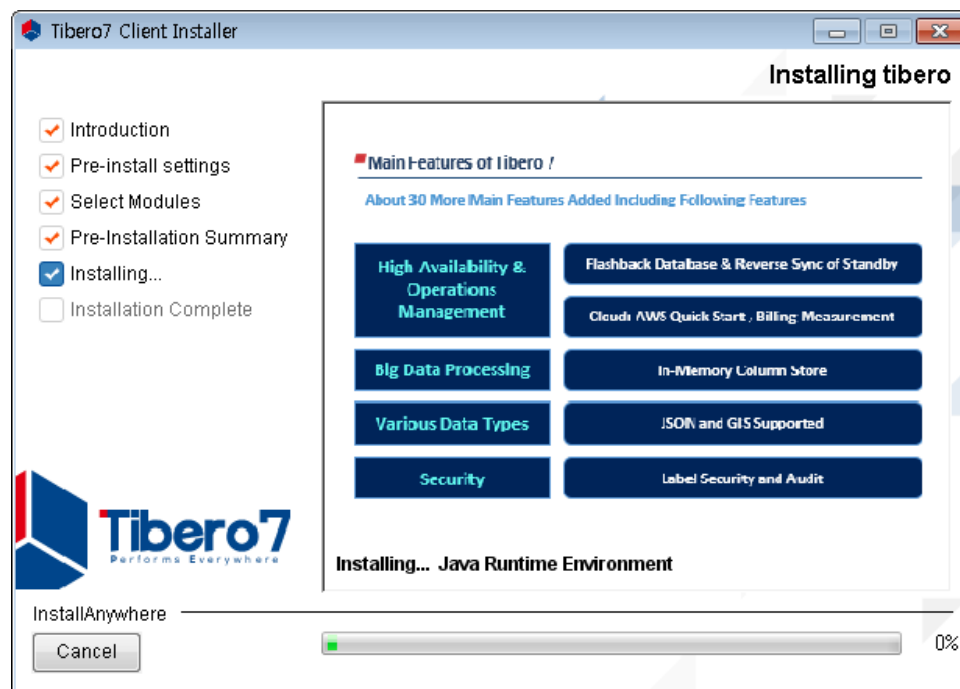
[그림 5.4] Tibero Client Installer - Pre-Installation Summary



5. Tibero클라이언트의 설치를 진행한다.

다음과 같이 진행 상태 바를 통해 설치가 진행되는 상황을 확인할 수 있다.

[그림 5.5] Tiberio Client Installer - Install Progress



6. Tiberio 클라이언트가 정상적으로 설치되면 아래와 같은 화면이 나타난다. **[Done]** 버튼을 클릭하여 설치를 종료한다.

[그림 5.6] Tiberio Client Installer - Install Complete



5.2. 제거

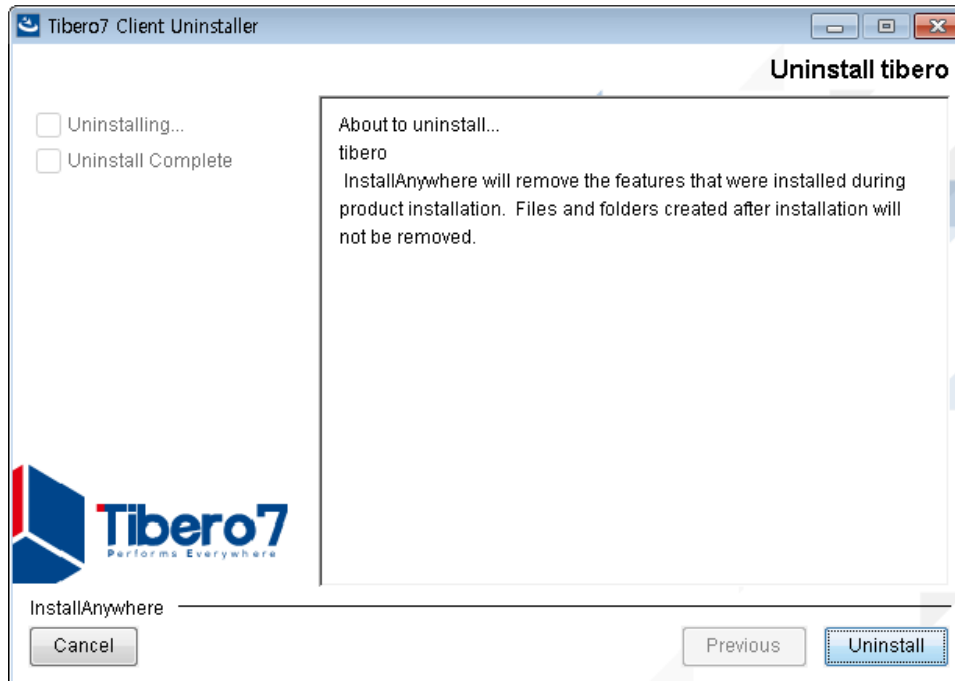
설치된 클라이언트를 제거하는 순서는 다음과 같다.

참고

Tibero 클라이언트를 제거하는 과정에서 기록되는 로그는 자동으로 삭제가 되지 않는다. 따라서 제거 과정이 완료된 후에 사용자가 별도로 삭제해 주어야 한다.

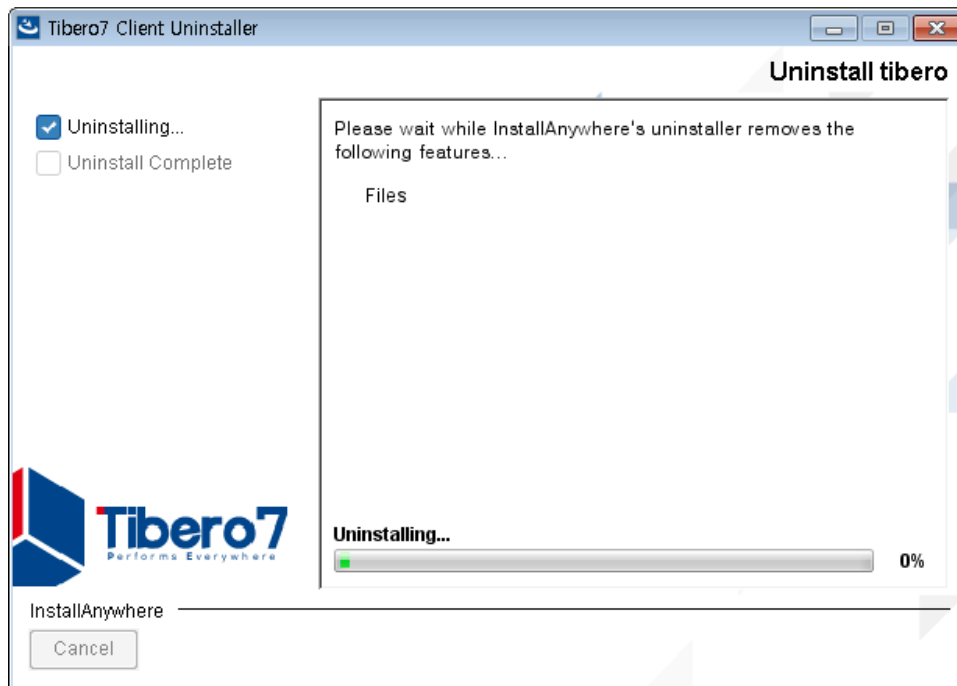
1. "{Tibero Client 설치 경로}\installation\Tibero Client Uninstall.exe" 파일을 더블 클릭하면 설치된 클라이언트를 제거하기 위해 아래와 같은 화면이 실행된다. **[Uninstall]** 버튼을 클릭하여 클라이언트 제거를 진행한다. 만약 제거 과정을 취소하려면 **[Cancel]** 버튼을 클릭한다.

[그림 5.7] Tibero Client Uninstaller - Introduction



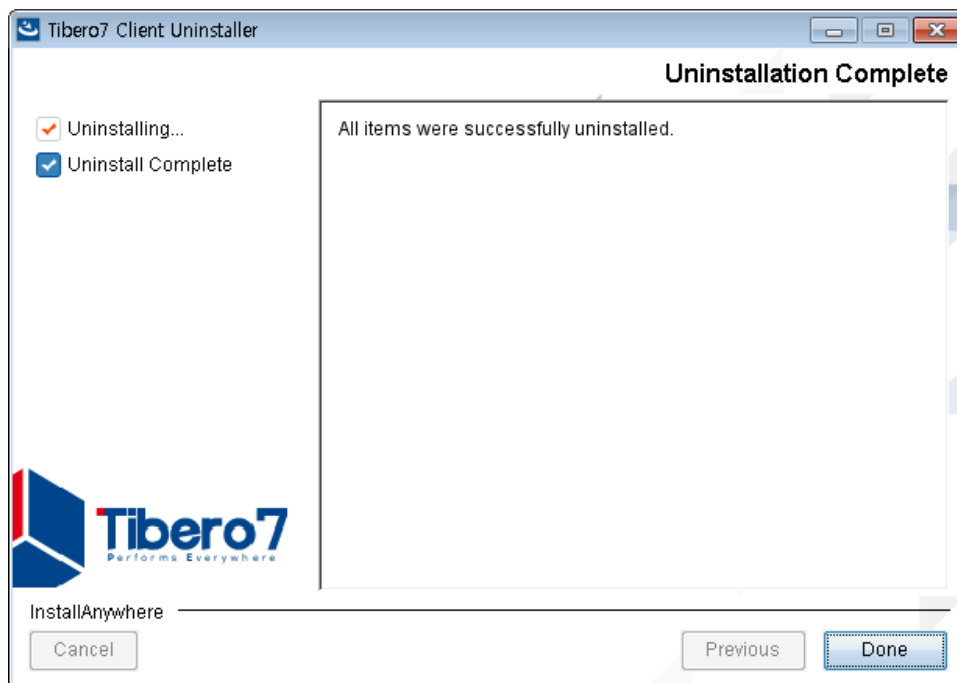
2. 다음은 설치된 Tibero 클라이언트의 제거가 진행되는 화면이다.

[그림 5.8] Tiberio Client Uninstaller - Uninstall Progress



3. 제거가 완료되면 다음과 같은 화면이 나타난다. **[Done]** 버튼을 클릭하여 제거 과정을 종료한다.

[그림 5.9] Tiberio Client Uninstaller - Uninstall Complete



제6장 멀티 인스턴스 설치와 제거

본 절에서는 멀티 인스턴스를 설치하고 제거하는 과정에 대해서 설명한다.

6.1. 설치

Windows 계열과 Unix(Linux) 계열에서 하나의 Tibero 바이너리 실행 파일과 라이선스 파일을 공유하면서 TB_SID와 환경 파일(.tip)을 따로 설정하여 두 개 이상의 서로 다른 데이터베이스 인스턴스를 동시에 설치할 수 있다.

6.1.1. Windows 환경

Windows 환경에서 수동 모드로 Tibero의 인스턴스를 두 개 이상 설치하는 순서는 다음과 같다.

1. 먼저 “3.2.1. Windows 환경”을 참조하여 Tibero 인스턴스를 하나 설치한다.

본 예제에서는 먼저 설치된 인스턴스의 환경변수가 아래와 같다고 가정하고 설명을 진행한다.

환경변수	설정값
TB_HOME	C:\Tibero\tibero7
TB_SID	tibero1
PATH	%TB_HOME%\bin;%TB_HOME%\client\bin

2. 추가로 설치할 인스턴스를 위한 TB_SID 환경변수를 설정한다. 반드시 이전에 설치된 인스턴스의 TB_SID와 값이 달라야 한다.

```
set TB_SID=tibero2
```

3. %TB_HOME%\bin 디렉터리에서 다음의 명령어를 입력한다. 이때 서비스명은 생략할 수 있다.

```
tbinstall %TB_HOME% %TB_SID% [-s 서비스명]
```

이 명령이 실행되면 Tibero가 Windows 서비스에 등록된다. 등록된 Windows 서비스는 [시작] > [설정] > [제어판] > [관리 도구] > [서비스]에서 확인할 수 있다. 오른쪽 마우스를 클릭해서 [관리자 권한으로 실행]으로 명령 프롬프트 창을 실행한다.

```
C:\Tibero\tibero7\bin>tbinstall %TB_HOME% %TB_SID%
Microsoft (R) Windows Script Host 버전 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
```

```
TB_HOME = C:\Tibero\tibero7
TB_SID = tibero2
service account = LocalSystem
Tibero_tibero installed successfully.
```

4. **%TB_HOME%\config** 디렉터리에서 다음의 명령어를 입력한다.

```
gen_tip.bat
```

이 명령이 실행되면 환경 파일(.tip)과 tbdns.tbr, psm_commands.bat 파일이 생성된다.

```
C:\Tibero\tibero7\config> gen_tip.bat
Using TB_SID [tibero2]
C:\Tibero\tibero7\config\tibero2.tip generated
C:\Tibero\tibero7\client\config\tbdns.tbr generated.
C:\Tibero\tibero7\config\psm_commands.bat generated
```

5. **%TB_HOME%\config** 디렉터리에 만들어진 환경 파일(.tip)을 열어서 LISTENER_PORT 값을 이전에 설치된 인스턴스의 LISTENER_PORT 값과 다르게 설정한다.

```
.....중간 생략.....

DB_NAME=tibero2
LISTENER_PORT=9629

.....중간 생략.....
```

참고

_LSNR_SPECIAL_PORT, _LSNR_SSL_PORT, CM_PORT 초기화 파라미터의 기본값이 각각 LISTENER_PORT+1, LISTENER_PORT+2, LISTENER_PORT+3로 설정된다.

따라서 LISTENER_PORT 값을 설정할 때 이전에 설치된 인스턴스와 새로 설치할 인스턴스의 LISTENER_PORT, _LSNR_SPECIAL_PORT, _LSNR_SSL_PORT, CM_PORT 초기화 파라미터 값이 서로 겹치지 않도록 주의한다.

-
6. 이 후 과정은 [“3.2.1. Windows 환경”](#)의 6번 이후의 절차와 동일하다. 단, 8번 절차에서 CREATE DATABASE 문을 이용하여 원하는 데이터베이스를 생성할 때 로그 파일이나 데이터 파일의 경로를 새로운 인스턴스의 TB_SID와 맞게 수정해야 한다.

6.1.2. Unix 환경

Unix 환경에서 수동 모드로 Tibero를 설치하는 순서는 기본적으로 싱글 인스턴스 설치와 동일하며, 순서는 다음과 같다.

1. 먼저 “3.2.2. UNIX 환경”을 참조하여 Tibero 인스턴스를 설치한다.

본 예제에서는 먼저 설치된 인스턴스의 환경변수가 아래와 같다고 가정하고 설명을 진행한다.

환경변수	설정값
TB_HOME	/home/tibero/Tibero/tibero7
TB_SID	tibero1
LD_LIBRARY_PATH	\$TB_HOME/lib:\$TB_HOME/client/lib
PATH	\$PATH:\$TB_HOME/bin:\$TB_HOME/client/bin

2. 추가로 설치할 인스턴스를 위한 TB_SID 환경변수를 설정한다. 반드시 이전에 설치된 인스턴스의 TB_SID와 값이 달라야 한다.

```
export TB_SID=tibero2
```

3. \$TB_HOME/config 디렉터리에서 다음의 명령어를 입력한다.

```
gen_tip.sh
```

이 명령이 실행되면 환경 파일(.tip)과 tbdns.tbr, psm_commands 파일이 생성된다.

```
tibero@Tibero:~/Tibero/tibero7/config$ gen_tip.sh
Using TB_SID "tibero2"
/home/tibero/Tibero/tibero7/config/tibero2.tip generated
/home/tibero/Tibero/tibero7/config/psm_commands generated
/home/tibero/Tibero/tibero7/client/config/tbdns.tbr generated.
Running client/config/gen_esql_cfg.sh
Done.
```

4. \$TB_HOME/config 디렉터리에 만들어진 환경 파일(.tip)을 열어서 LISTENER_PORT 값을 이전에 설치된 인스턴스의 LISTENER_PORT 값과 다르게 설정한다.

```
tibero@Tibero:~/Tibero/tibero7/config$ cat tibero2.tip
.....중간 생략.....

DB_NAME=tibero2
LISTENER_PORT=9629
.....중간 생략.....
```

참고

_LSNR_SPECIAL_PORT, _LSNR_SSL_PORT, CM_PORT 초기화 파라미터의 기본값이 각각 LISTENER_PORT+1, LISTENER_PORT+2, LISTENER_PORT+3로 설정된다.

따라서 LISTENER_PORT 값을 설정할 때 이전에 설치된 인스턴스와 새로 설치할 인스턴스의 LISTENER_PORT, _LSNR_SPECIAL_PORT, _LSNR_SSL_PORT, CM_PORT 초기화 파라미터 값이 서로 겹치지 않도록 주의한다.

5. 이 후 과정은 “3.2.2. UNIX 환경”의 5번 이후의 절차와 동일하다. 단, 7번 절차에서 CREATE DATABASE 문을 이용하여 원하는 데이터베이스를 생성할 때 로그 파일이나 데이터 파일의 경로를 새로운 인스턴스의 TB_SID와 맞게 수정해야 한다.

6.2. 제거

설치된 멀티 인스턴스를 제거하는 순서를 설명한다.

6.2.1. Windows 환경

Windows 환경에서 수동 모드로 Tiberο의 멀티 인스턴스를 제거하는 순서는 다음과 같다. 순서는 기본적으로 싱글 인스턴스 제거와 동일하다.

1. 본 예제에서는 설치된 멀티 인스턴스의 환경변수가 아래와 같다고 가정하고 설명을 진행한다.

환경변수	설정값
TB_HOME	/home/tibero/Tibero/tibero7
TB_SID	tibero1
	tibero2
PATH	\$PATH:\$TB_HOME/bin:\$TB_HOME/client/bin

2. 먼저 Tiberο 인스턴스를 종료한다. 제거할 인스턴스의 환경변수를 설정한 후 인스턴스를 종료한다.

```
set TB_SID=tibero1
tbdwn
```

3. %TB_HOME%\bin 디렉터리에서 다음의 명령어를 입력한다.

```
tbuninstall %TB_SID%
```

이 명령이 실행되면 Windows 서비스에 등록된 Tiberο가 제거된다. 오른쪽 마우스를 클릭해서 **[관리자 권한으로 실행]**으로 명령 프롬프트 창을 실행해야 한다.

```
C:\Tibero\tibero7\bin>tbuninstall %TB_SID%
Microsoft (R) Windows Script Host 버전 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

TB_HOME = C:\Tibero\tibero7
TB_SID = tibero
service name = Tibero_tibero
Tibero_tibero uninstalled successfully.
```

4. 추가로 제거할 인스턴스의 TB_SID 환경변수를 설정한 후 인스턴스 종료 및 서비스 제거 절차를 동일하게 수행한다.

```
set TB_SID=tibero2
tbdowndown
```

```
C:\Tibero\tibero7\bin>tbuninstall %TB_SID%
Microsoft (R) Windows Script Host 버전 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

TB_HOME = C:\Tibero\tibero7
TB_SID = tibero2
service name = Tibero_tibero2
Tibero_tibero2 uninstalled successfully.
```

5. Tibero가 설치된 디렉터리를 포함한 모든 하위 디렉터리를 해당 운영체제의 명령어로 삭제한다.

```
rmdir %TB_HOME%
```

6.2.2. Unix 환경

Unix 환경에서 수동 모드로 Tibero를 제거하는 순서는 기본적으로 싱글 인스턴스 제거와 동일하며, 순서는 다음과 같다.

1. 본 예제에서는 먼저 설치된 인스턴스의 환경변수가 아래와 같다고 가정하고 설명을 진행한다.

환경변수	설정값
TB_HOME	/home/tibero/Tibero/tibero7
TB_SID	tibero1
	tibero2
LD_LIBRARY_PATH	\$TB_HOME/lib:\$TB_HOME/client/lib
PATH	\$PATH:\$TB_HOME/bin:\$TB_HOME/client/bin

2. 먼저 Tibero 인스턴스를 종료한다. 제거할 인스턴스의 환경변수를 설정한 후 인스턴스를 종료한다.

```
export TB_SID=tibero1  
tbdown
```

3. 추가로 제거할 인스턴스의 TB_SID 환경변수를 설정한 후 인스턴스를 종료한다.

```
export TB_SID=tibero2  
tbdown
```

4. Tibero가 설치된 디렉터리를 포함한 모든 하위 디렉터리를 해당 운영체제의 명령어로 삭제한다.

```
rm -rf $TB_HOME
```

제7장 TAC 설치와 제거

본 절에서는 TAC(Tibero Active Cluster) 환경을 구축하기 위한 Tibero의 설치 방법을 설명한다.

7.1. 설치 전 준비사항

TAC를 구축하기 전에 다음의 사항을 확인한다.

- 시스템 요구사항

TAC 설치를 위한 시스템 요구사항의 내용은 “1.3. 시스템 요구 사항”을 참고한다.

- 설치 전 준비사항

TAC를 구성 전 “제2장 설치 전 준비사항”을 참고하여 기본적인 사항을 점검한다.

- 인터커넥트 구성

- IP 주소 및 포트 정보 확인

- 소켓 버퍼 설정

- 공유 디스크 타입 확인

7.1.1. 인터커넥트 구성

인터커넥트(Interconnect)란 TAC 노드간의 통신에 필요한 별도의 네트워크이다.

인터커넥트를 구성할 때 노드간 직접 연결보다는 별도의 인터커넥트 스위치(Switch)를 사용하는 것을 권고한다. 노드간 직접 연결시 인터커넥트가 단일 장애점(Single point of failure)이 될 수 있고 3노드 이상 구성에 어려움이 있기 때문이다. 별도의 인터커넥트 스위치 사용시 노드간 직접 연결보다 서비스 안정성 및 확장성 등의 측면에서 장점이 있다.

7.1.2. IP 주소 및 포트 정보 확인

설치를 시작하기 전에 IP 주소와 포트 번호를 확인한다. 기존 Tibero 단일 인스턴스에서 필요로 하는 IP 주소(외부 IP 주소)와 Listener 포트 번호 외에 추가로 Interconnect IP 주소와 포트 번호, CM 포트 번호가 필요하다. VIP, IP 필터링 기능을 사용하는 경우 동일 서브넷에 IP가 부여되어 있는지 확인이 필요하다.

- IP 주소

구분	설명
외부 IP 주소	외부 통신을 위해 부여될 IP 주소를 확인한다.
Interconnect IP 주소	각 노드간 통신을 위해 부여될 내부 IP 주소를 확인한다.

- 포트 번호

구분	설명
Listener 포트 번호	각 노드의 Tiberio 인스턴스를 기동할 때 부여될 포트 번호를 확인한다.
CM 포트 번호	각 노드의 통신을 위한 CM에 부여될 포트 번호를 확인한다.
Interconnect 포트 번호	각 노드의 Tiberio 인스턴스간에 통신할 포트 번호를 확인한다.

7.1.3. 소켓 버퍼 설정

운영체제의 소켓 버퍼 값을 설정한다.

- AIX

파라미터	권장 값
SB_MAX	4194304
TCP_RECVSPACE	4194304
TCP_SENDSIZE	1048576

- Linux

파라미터	권장 값
net.core.rmem_max	67108864, OS 기존 설정값, TCP_RCVBUF_SIZE, _INC_TCP_RCVBUF_SIZE 중 최댓값
net.core.wmem_max	67108864, OS 기존 설정값, TCP_SNDBUF_SIZE, _INC_TCP_SNDBUF_SIZE 중 최댓값

- Solaris

파라미터	권장 값
max_buf	4194304
recv_buf	4194304
send_buf	1048576

참고

sb_max 파라미터는 AIX 운영체제에만 해당한다.

7.1.4. 공유 디스크 타입 확인

TAC는 모든 인스턴스가 같이 사용할 수 있는 공유 디스크의 공간을 필요로 한다.

TAC의 실행과 운영을 위해서는 최소 7개의 공유 파일이 필요하다. 파일의 위치는 설치할 때 사용자에게 입력받는 {SHARED_DISK_HOME}을 기준으로 생성되며 노드마다 필요로 하는 파일은 {TAC_INSTANCE_ID}를 붙여서 생성한다. 예를 들어 {SHARED_DISK_HOME}이 dev/tac이고 {TAC_INSTANCE_ID}가 0일 경우 Undo 로그 파일의 경로는 'dev/tac/UNDO0.dtf'이다.

다음은 처음 노드를 설치할 때 필요한 공유 파일의 목록과 파일 생성 경로이다.

파일	경로
컨트롤 파일	{SHARED_DISK_HOME}/c1.ctf(변경 가능)
CM용 클러스터 파일	{SHARED_DISK_HOME}/cfile/cls1_cfile(변경 가능)
Redo 로그 파일	{SHARED_DISK_HOME}/log{TAC_INSTANCE_ID}1.log {SHARED_DISK_HOME}/log{TAC_INSTANCE_ID}2.log {SHARED_DISK_HOME}/log{TAC_INSTANCE_ID}3.log
Undo 로그 파일	{SHARED_DISK_HOME}/UNDO{TAC_INSTANCE_ID}.dtf
사용자 테이블 스페이스 파일	{SHARED_DISK_HOME}/usr001.dtf
시스템 테이블 스페이스 파일	{SHARED_DISK_HOME}/system001.dtf
임시 테이블 스페이스 파일	{SHARED_DISK_HOME}/temp001.dtf

참고

하드웨어 특성상 위의 설정 대로 파일명을 사용할 수 없는 경우는 Symbolic Link를 활용한다.

처음 노드 설치 이후 추가적으로 노드를 설치하기 위해 각 노드당 4개의 공유 파일이 요구된다. 각 파일의 경로는 위의 표를 참고한다.

- Redo 로그 파일 3개
- Undo 로그 파일

참고

1. 공유 디스크로 Raw Device를 사용할 경우 위의 각 공유 파일들은 하나의 Raw Device 파일에 해당하며 설치하기 전에 미리 Raw Device 관리자에게 해당 파일을 생성해 줄 것을 요청해야 한다. 단, 각 파일의 크기는 충분해야 하고 Block Size는 512로 맞추어져 있어야 하며 Charater Type으로 설정해야 한다.
 2. 컨트롤 파일, CM용 클러스터 파일과 리소스 파일의 경우 사용자가 원하는 경로와 파일 이름으로 설정할 수 있다.
-

7.2. 수동 설치

수동으로 TAC 환경을 구축하고 Tibero를 설치하는 방법은 "Tibero 관리자 안내서"의 "14장 Tibero Cluster Manager"와 "15장 Tibero Active Cluster"를 참조한다.

7.3. 설치 확인

설치 후 정상 작동 여부 확인은 **tbcm** 명령어를 이용하여 수행할 수 있다.

```
tbcm -s
```

실행 후 나타나는 화면을 통하여 CM의 설정상태를 확인할 수 있다.

```
CM information
=====
CM NAME           : cm0
CM UI PORT        : 8635
RESOURCE FILE PATH : /home/tibero7/cm0_res.crf
CM MODE           : GUARD ON, FENCE OFF
LOG LEVEL         : 2
=====
```

참고

tbcm 명령의 자세한 사용법은 "Tibero 관리자 안내서"의 "14장 Tibero Cluster Manager"를 참고한다.

7.4. 제거

TAC 환경에서 한 노드를 제거하기 위한 방법에는 **콘솔 모드**와 **수동 모드**가 있다.

콘솔 모드

콘솔 모드로 하나의 노드를 제거하는 순서는 다음과 같다.

1. Tiberero 인스턴스를 종료한다.

```
tbdown
```

2. TBCM을 종료한다.

```
tbcm -d
```

3. Tiberero가 설치된 디렉터리를 포함한 모든 하위 디렉터리를 해당 운영체제의 명령어로 삭제한다.

```
$ rm -rf $TB_HOME
```

수동 모드

수동 모드로 하나의 노드를 제거하는 순서는 다음과 같다.

1. Tiberero 인스턴스를 종료한다.

```
tbdown
```

2. TBCM을 종료한다.

```
tbcm -d
```

3. Tiberero 인스턴스가 설치된 디렉터리를 포함한 모든 하위 디렉터리를 삭제한다.

```
rm -rf $TB_HOME
```


Appendix A. 설치 후 문제 해결

본 부록에서는 Tibero를 설치한 후 발생할 수 있는 문제를 해결하는 방법을 설명한다.

A.1. 사용자 환경설정 파일

A.1.1. TB_HOME

- 문제

```
TBR-2048 : Data source was not found.
```

tbSQL 유틸리티에 접속할 때 이러한 메시지가 나타나면 **TB_HOME**이 설정되지 않았거나 잘못된 경우 또는 Tibero를 설치한 후 환경설정 파일을 적용하지 않은 경우에 발생하는 문제이다.

- 해결 방법

사용자 환경설정 파일의 **TB_HOME** 환경변수를 확인하고 변경한 뒤 시스템에 적용시킨다.

A.1.2. LD_LIBRARY_PATH

- 문제

```
tbsql: error while loading shared libraries: libtbcli.so:  
cannot open shared object file: No such file or directory
```

tbSQL 유틸리티에 접속할 때 이러한 메시지가 나타나면 **LD_LIBRARY_PATH**가 설정되지 않았거나 잘못된 경우 또는 Tibero를 설치한 후 환경설정 파일을 적용하지 않은 경우에 발생하는 문제이다.

- 해결 방법

사용자 환경설정 파일의 **LD_LIBRARY_PATH**를 확인하고 변경한 뒤 시스템에 적용시킨다.

A.1.3. TB_SID

- 문제

```
ERROR: environment variable $TB_SID is not set
```

```
tbdownd: environment variable TB_HOME or TB_SID is not set.
```

```
Tip file open failure.: No such file or directory
```

```
tbdownd failed. proc info file is deleted.
```

tbboot 또는 **tbdownd** 명령어를 실행할 때 위와 같은 메시지 중 하나가 나타나면 **TB_SID**가 설정되지 않은 경우에 발생하는 문제이다.

- 해결 방법

사용자 환경설정 파일의 **TB_SID**를 확인하고 변경한 뒤 시스템에 적용시킨다.

A.2. TAC 설치

본 절에서는 TAC 환경에서 Tibero를 설치한 후 발생할 수 있는 문제와 해결방법을 설명한다.

A.2.1. 노드 추가

tip 파일 접근 문제

- 문제

```
tip file does not exist / reading tip file failed / malformed tip file
```

노드를 추가할 때 설정 파일(.tip)을 찾지 못하여 발생하는 문제이다.

- 해결 방법

TB_HOME, **TB_SID**를 참고하여 해당 경로에 기존 Tibero 인스턴스의 **\$TB_SID.tip**이 위치해 있는지 확인한다.

TAC 접근 문제

- 문제

```
extracting information from the existing cluster failed
```

노드를 자동으로 추가할 때 설정값들을 읽어오기 위하여 기존 노드에 접속하는데 이를 실패하여 발생하는 문제이다.

- 해결 방법

- **TB_HOME**와 **TB_SID**가 제대로 설정되어 있는지 확인한다.
- **tbsql**로 기존 노드에 접속 가능한지 확인한다.
- 기존 노드가 작동하고 있지 않은 경우 **tbcm**, **tbboot** 명령어로 기동한 후 노드 추가를 재시도한다.

scp 파일 전송 문제

- 문제

```
Remote file transfer failed! / Remote configuration export failed!
```

새로운 노드에 Tiberio 설치 파일을 전송하는 도중 실패하여 발생하는 문제이다.

- 해결 방법

- 추가하려는 노드에 현재 노드에서 접근할 수 있는지 ping 또는 ssh 등으로 확인한다.
- 원활하게 이루어지지 않을 경우 계정 동기화 설정과 네트워크 설정, 방화벽 설정 등을 확인한다.

A.2.2. TAC raw device 환경에서 TPR 관련 주의사항

- 문제

Tiberio 5.0 r61295 이후부터는 TPR 관련 테이블을 담는 TPR 전용 테이블 스페이스(SYSSUB)가 추가되었다. 설치 또는 패치할 때 시스템 스키마를 구성하는 단계에서 자동으로 \$TB_HOME/database/\$TB_SID 아래에 파일(syssub001.dtf)을 만든다. 단, DB_CREATE_FILE_DEST가 명시되어 있는 경우에는 해당 위치에 만든다.

하지만 TAC raw device 사용 환경이라면 그 시스템 스키마를 수행하는 노드의 로컬 디스크에 해당 파일이 생성되어 다른 노드에서는 TPR 테이블 스페이스에 접근하지 못하는 오류가 발생한다. 공유 파일 시스템이면서 DB_CREATE_FILE_DEST가 적절한 경로로 지정되지 않은 환경에서도 마찬가지이다.

- 해결 방법

아래와 같은 방법으로 SYSSUB 테이블 스페이스를 재생성한다.

1. 테이블 스페이스를 삭제한다.

```
SQL> DROP TABLESPACE SYSSUB INCLUDING CONTENTS AND DATAFILES;
```

2. 테이블 스페이스를 생성한다.

```
SQL> CREATE TABLESPACE SYSSUB DATAFILE '<SYSSUB 위치>/syssub001.dtf' ...;
```

3. 다음의 스크립트를 실행한다.

```
$TB_HOME/scripts/tpr.sql  
$TB_HOME/scripts/pkg/_pkg_dbms_tpr.tbw
```

A.3. 기타

A.3.1. 포트 번호

- 문제

```
Listener port = 8629
bind() failed: Address already in use.
Error: Timedout while trying to open port 8629
Check if there are any Tibero instances running.
Tibero instance startup failed!
```

tbboot 명령어를 실행할 때 사용하려는 포트 번호가 이미 사용 중인 경우에 발생하는 문제이다.

- 해결 방법

해당 인스턴스가 이미 실행 중인지 확인하거나 `$TB_HOME/client/config` 디렉터리에 있는 `$TB_SID.tip` 파일에 설정된 포트 번호를 수정한다.

A.3.2. dbtimezone

- 권고 사항

`dbtimezone` 값은 `timestamp with local timezone` 데이터 타입을 DB에 저장할 때 사용되는 기준 `timezone` 값이다. DB 내부에서 처리되는 값이므로 불필요한 연산이 추가되어 성능상에도 안좋은 영향을 미치게 되어 임의 변경을 권고하지 않는다.

Appendix B. 마운트

본 부록에서는 플랫폼별로 CD를 마운트하는 방법을 설명한다.

B.1. AIX

AIX에서 CD를 마운트하는 방법은 다음과 같다.

1. 루트 계정으로 접속한다.
2. CD-ROM을 마운트시킬 마운트 포인트(Mount Point)를 생성한다.

```
# mkdir /cdrom
```

3. 마운트 명령을 실행한다.

```
mount <option><device_name><mount_point>
```

B.2. Linux

Linux에서 CD를 마운트하는 방법은 다음과 같다.

1. 오토 마운팅(AutoMounting)이 되는지 확인한다.

```
$ ps -aux | grep automount
```

2. 루트 계정의 비밀번호를 이용하여 루트 계정으로 이동한다.

```
$ su -  
Password : *****
```

3. 마운트 명령을 실행한다.

```
# mount <option><device_name><mount_point>
```

4. 언마운트(Unmount) 명령을 실행한다.

```
# umount <mount_point>
```

B.3. Solaris

Solaris에서 CD를 마운트하는 방법은 다음과 같다.

1. 볼륨 매니저(Volume Manager)를 검사한다. 즉, 자동으로 인식하는지를 확인하는 과정이다.

```
$ ps -e | grep vold
```

2. 기존의 CD가 있다면 이를 꺼낸다.

```
$ eject
```

3. CD를 넣고 다음의 명령을 입력하여 마운트한다.

```
# mkdir <mount_point>  
# mount <options> <device_name><mount_point>
```

Appendix C. Tiberio 지원 문자 집합

본 부록에서는 Tiberio에서 지원하는 문자 집합(character set)을 설명한다.

Tiberio에서는 다음과 같은 문자 집합을 제공한다.

언어	문자 집합(character set)	설명
다국어	UTF8	8-bit 국제 표준 다국어(기본값)
	UTF16	16-bit 국제 표준 다국어
한국어	EUCKR	EUC 16-bit 한국어
	MSWIN949	MS Windows 코드 페이지 949 한국어
영어	ASCII	ASCII 7-bit 영어
일본어	SJIS	SHIFT-JIS 16-bit 일본어
	SJISTILDE	전각물결문자를 포함하는 SHIFT-JIS 16-bit 일본어
	JA16SJIS	MS Windows 코드 페이지 932 일본어
	JA16SJISTILDE	전각물결문자를 포함하는 MS Windows 코드 페이지 932 일본어
	JA16EUC	EUC 24-bit 일본어
	JA16EUCTILDE	전각물결문자를 포함하는 EUC 24-bit 일본어
중국어	GBK	MS Windows 코드 페이지 936 중국어
	GB18030	Chinese National Standard GB18030-2000
중국어, 홍콩어, 대만어	ZHT16HKSCS	HKSCS2001 홍콩어
	ZHT16BIG5	BIG5 코드 페이지 중국어
	ZHT16MSWIN950	MS Windows 코드 페이지 950 중국어
	EUCTW	EUC 32-bit Traditional Chinese
베트남어	VN8VN3	VN3 8-bit 베트남어
태국어	TH8TISASCII	Thai Industrial Standard 620-2533 - ASCII 8-bit
동유럽어	EE8ISO8859P2	ISO8859-2 동유럽어
서유럽어	WE8MSWIN1252	MS Windows 코드 페이지 1252 서유럽어
	WE8ISO8859P1	ISO8859-1 서유럽어
	WE8ISO8859P9	ISO8859-9 서유럽어(터키어)
	WE8ISO8859P15	ISO8859-15 서유럽어

언어	문자 집합(character set)	설명
러시아어, 불가리아어	CL8MSWIN1251	MS Windows 코드 페이지 1251 키릴문자
	CL8KOI8R	KOI8-R 키릴문자
	CL8ISO8859P5	ISO8859-5 키릴문자
	RU8PC866	IBM-PC 코드 페이지 866 8-bit 키릴문자
그리스어	EL8ISO8859P7	ISO8859-7 그리스어
	EL8MSWIN1253	MS Windows 코드 페이지 1253 8-bit 그리스어
아랍어	AR8ISO8859P6	ISO8859-6 아랍어
	AR8MSWIN1256	MS Windows 코드 페이지 1256 8-bit 아랍어
히브리어	IW8ISO8859P8	ISO8859-8 히브리어

참고로 Tiberio 바이너리를 설치할 때에 다음과 같은 방법으로 설치 가능한 문자 집합에 대해 확인할 수 있다.

```
$ tbboot -C
```

```
Available character set list
```

```
ASCII
CL8ISO8859P5
CL8KOI8R
CL8MSWIN1251
EE8ISO8859P2
EUCKR
.....중간 생략.....
```

```
Available nls_date_lang set list
```

```
AMERICAN
BRAZILIAN PORTUGUESE
JAPANESE
KOREAN
.....중간 생략.....
```

Appendix D. system.sh

본 부록에서는 system.sh(vbs)를 수행하는 경우 각 옵션에 대하여 설명한다.

system.sh(vbs)에서 사용 가능한 옵션은 다음과 같다.

```
$ system.sh -h
Usage: system.sh [option] [arg]
-h : dispaly usage
-p1 password : sys password
-p2 password : syscat password
-a1 Y/N : create default system users & roles
-a2 Y/N : create system tables related to profile
-a3 Y/N : register dbms stat job to Job Scheduler
-a4 Y/N : create TPR tables
pkgonly : create psm built-in packages only
-sod Y/N : separation of duties
error : print error whenever sql encounters error
```

다음은 각 옵션에 대한 설명이다.

옵션	설명
-h	도움말 화면을 출력한다.
-p1 password	sys 계정의 패스워드를 입력한다. (기본값: tiber0)
-p2 password	syscat 계정의 패스워드를 입력한다. (기본값: syscat)
-a1	시스템 유저 생성 및 권한부여 여부를 설정한다. 최초 설치일 경우 반드시 'Y'로 설정한다. – Y : 시스템 유저 생성 및 권한을 부여한다. – N : 데이터를 보존한다.
-a2	데이터베이스 사용자의 패스워드 관리 정책을 지정할 수 있는 프로파일을 위한 테이블의 생성 여부를 설정한다. 최초 설치일 경우 반드시 'Y'로 설정한다. – Y : 기존 테이블을 삭제한 후 생성한다. – N : 데이터를 보존한다. 프로파일에 대한 자세한 내용은 "Tibero 관리자 안내서"의 "5장 사용자 관리와 데이터베이스 보안"을 참고한다.

옵션	설명
-a3	데이터베이스 객체의 통계 정보 수집을 위한 Job 스케줄링 등록 여부를 설정한다. 최초 설치일 경우 'Y'로 설정하는 것을 권장한다. – Y : 통계정보 수집을 위한 Job을 삭제한 후 등록한다. – N : 등록하지 않는다.
-a4	통계정보를 주기적으로 자동 수집하기 위한 TPR 관련 테이블의 생성 여부를 설정한다. 최초 설치일 경우 반드시 'Y'로 설정한다. – Y : 기존 테이블을 삭제한 후 생성한다. – N : 데이터를 보존한다. APM에 대한 자세한 내용은 "Tibero 관리자 안내서"의 "제17장 Tibero Performance Repository"를 참고한다.
pkgonly	PSM built-in package 스크립트들만 수행한다.
-sod	보안을 위하여 기존 하나의 관리자 권한을 아래와 같이 3개의 관리자로 나뉘어 설치할 것인지에 대한 여부를 설정한다. – SYSSEC : 보안 관리자 – SYSAUD : 감사 관리자 – SYSADM : 시스템 관리자 (기본값: N, Windows 계열에서는 미지원)
error	system.sh 수행 중 특정 스크립트에서 에러가 발생할 경우 에러 내용을 터미널에 출력한다. (Windows 계열에서는 미지원)

Appendix E. HugePage 설정 방법

본 부록에서는 각 운영체제별 HugePage 설정 방법에 대하여 설명한다.

E.1. Linux

본 절에서는 Linux 환경에서 HugePage를 설정하기 위한 방법에 대하여 설명한다.

HugePage로 설정된 공간은 공유 메모리 용도로 사용되며, 이 때 특징은 다음과 같다.

- 메모리 주소 변환에 사용되는 TLB(Translation Lookaside Buffer) hit 비율 상승으로 메모리 접근 성능이 향상될 수 있다.
- Page 관리를 위한 구조인 Page Table Size가 줄어든다. 공유 메모리는 DB process 전체가 접근하는 공간으로서 같은 메모리에 대한 Page Table이 process마다 생성되는데, HugePage를 사용함으로써 메모리 사용량을 크게 줄일 수 있다. session 수가 증가 (process 수가 증가) 할수록 메모리 절약 효과는 증대된다.
- 확보된 HugePage 공간은 DB 인스턴스 부팅 시점에 공유 메모리로 할당되므로 사용/해제를 위해 DB 인스턴스 재기동이 필요하다. 이 때 할당된 메모리는 공유 메모리를 위한 공간으로 고정되어 시스템 내에서 다른 용도(kernel 메모리, process 메모리 등)로 사용되지 않는다.
- HugePage로 설정된 Page는 자동으로 locked 상태가 되며, 공유 메모리 영역이 디스크로 swap out되지 않도록 보장한다.

참고

Page Table Size에 대한 자세한 계산식은 “[E.1.4. Pagetable Size 계산](#)”을 참고한다.

E.1.1. 활성화

HugePage를 활성화하기 위해 root 권한이 필요하며, 설정 방법은 다음과 같다.

1. 현재 운영체제에서 지원하는 HugePage의 크기를 다음과 같이 확인한다.

```
$ grep Hugepagesize /proc/meminfo
Hugepagesize:      2048 KB
```

2. Tiberio를 운영하는 유저 그룹의 아이디를 다음과 같이 확인한다.

```
$ id -a
uid=579(tibero) gid=579(tibero) groups=579(tibero)
```

3. HugePage를 할당할 그룹과 개수를 "/etc/sysctl.conf"에 적용한다.

커널 파라미터	설명	공식
vm.nr_hugepages	HugePage 개수	최소 $[TOTAL_SHM_SIZE / \text{HugePage 크기}]$ 이상
vm.hugetlb_shm_group	HugePage를 할당할 그룹 아이디	Tibero를 운영하는 유저 그룹의 아이디

참고

DB 인스턴스만을 부팅하기 위한 vm.nr_hugepages 의 최소값이 $[TOTAL_SHM_SIZE / \text{HugePage 크기}]$ 이기 때문에, OS 상에 타 프로그램들이 존재하는 경우 이를 고려하여 OS 담당자와 충분한 상의 후 값을 조정 및 반영해야한다.

다음은 TOTAL_SHM_SIZE의 설정값이 1024MB 이고, 현재 운영체제에서 지원하는 HugePage 크기가 2MB(2048 KB)인 경우에 커널파라미터를 적용한 예제이다.

```
$ cat /etc/sysctl.conf
.....중간 생략.....
vm.nr_hugepages=520
vm.hugetlb_shm_group=579
```

4. Max locked memory 제한값을 "/etc/security/limits.conf"에 적용한다.

설정값	공식
memlock	HugePage 크기 * HugePage 개수

다음은 앞에서 구한 값으로 memlock 값을 설정한 예제이다.

```
$ cat /etc/security/limits.conf
.....중간 생략.....
tibero          soft      memlock         1048576
tibero          hard      memlock         1048576
```

5. 운영체제를 재시작한다.

아래와 같이 변경된 HugePage 값을 확인할 수 있다.

```
$ egrep -e HugePages_Total /proc/meminfo
HugePages_Total:    512 KB
```

6. 환경설정 파일(.tip)에 다음과 같은 Tibero 초기화 파라미터를 설정한다.

초기화 파라미터	설정값
TOTAL_SHM_SIZE	HugePage 크기 * HugePage 개수
USE_HUGE_PAGE	Y

다음은 앞에서 구한 값으로 초기화 파라미터를 설정한 예제이다.

```
$ cat tibero.tip
.....중간 생략.....
TOTAL_SHM_SIZE=1G
USE_HUGE_PAGE=Y
```

7. Tibero 서버를 재기동한다.

```
$ tbdwn

Tibero instance terminated (NORMAL mode).

$ tbboot
Listener port = 8629
Change core dump dir to /home/tibero/tibero7/instance/tibero.

Tibero 7

TmaxData Corporation Copyright (c) 2008-. All rights reserved.
Tibero instance started up (NORMAL mode).
```

E.1.2. 비활성화

HugePage 활성화를 위해 수행한 과정과 동일한 순서로 수정하였던 값을 되돌리는 과정을 진행한다.

E.1.3. THP(Transparent Huge Pages) 비활성화

Linux 환경의 경우, OS 상의 메모리 사용 패턴에 따라 자동으로 메모리 페이지 단위를 높여주는 THP 라는 기능을 지원하고 있다.

그러나 해당 기능을 활성화 하는 경우, 공유 메모리 뿐만 아니라 프로세스 메모리까지 HugePage 로 동작 하게 되면서 성능이 일관되지 않게 동작할 수 있다. (성능 저하를 야기할 수 있음)

때문에 해당 커널 파라미터를 never로 설정하는 것을 강력하게 권장한다.

E.1.4. Pagedtable Size 계산

x86_64 아키텍처에서 Linux는 4단계 계층적 페이지 테이블 구조(PGD→PUD→PMD→PTE)를 사용하며 페이지 크기가 커질수록(4KB→2MB→1GB) 테이블 계층이 한 단계씩 줄어들고 전체 page table 크기가 작아진다.

- 4KB page: 4단계 사용 (PGD→PUD→PMD→PTE)
- 2MB page: 3단계 사용 (PGD→PUD→PMD)
- 1GB page: 2단계 사용 (PGD→PUD)

다음은 TOTAL_SHM_SIZE의 설정값이 50GB 인 경우 증가할 수 있는 page table의 최대 크기를 계산한 예제이다.

- 전체 shared memory 크기 / 페이지 크기 = 필요한 페이지 개수
- 각 엔트리 크기 = 8 bytes (x86_64 기준)
- 마지막 레벨 테이블 크기 = 페이지 개수 × 8 bytes
- 전체 page table size = 모든 레벨의 테이블 크기 합

[표 E.1] 페이지 크기별 page table size (SHM 50GB 기준)

Page Size	필요 page 개수	마지막 레벨 테이블 크기	프로세스 page table size
4KB	13,107,200	102,400KB	(1GB page table size) + (2MB page table size) + 102,400KB
2MB	25,600	200KB	(1GB page table size) + 200KB
1GB	50	400B	400B

전체 page table size는 상위 레벨 테이블 크기가 누적되는 방식이므로 페이지 크기가 커질수록 필요한 테이블 레벨이 줄고, 전체 page table size가 감소한다.

- 4KB: 1GB page table size + 2MB page table size + 마지막 레벨(pte) 크기
- 2MB: 1GB page table size + 마지막 레벨(pmd) 크기
- 1GB: 마지막 레벨(pud) 크기만 필요

최종적으로, <DB 프로세스 개수 * 프로세스 page table size> 의 결과값이 DB 인스턴스 전체 page table 최대 size가 된다.

참고

Process Memory를 위한 Page Table Size는 별도로 계산된다.

E.2. AIX

AIX에서는 HugePage 대신 Large page라는 용어를 대신 사용하며 기본적인 장점은 HugePage와 동일하다.

E.2.1. 활성화

HugePage를 활성화하기 위한 설정 방법은 다음과 같다.

1. 운영체제 Large page 설정값을 변경한다.

AIX는 내부적으로 4KB 및 16MB 크기의 물리 Memory Pool을 유지한다. 이 Pool의 크기는 vmo 명령을 통해 16MB Memory Pool의 크기를 변경할 수 있다. 나머지 공간은 자연스럽게 4KB로 할당된다.

AIX 5.3 이후부터는 Large page Pool이 동적으로 유지되기 때문에 변경 후 시스템을 재부팅할 필요가 없다.

먼저 공유 메모리 할당된 공간이 디스크로 스왑되지 않도록 v_pinshm 설정이 별도로 필요하다. percent_of_real_memory는 전체 메모리 중에서 TSM이 차지하는 비율을 지정한다.

```
$ vmo r o v_pinshm=1
$ vmo r o maxpin%=percent_of_real_memory
```

Large page Pool 크기를 설정한다. num_of_lage_pages는 TSM을 16MB로 나눈 값이 되며, 정수 단위로 입력한다.

```
$ vmo p o lgpg_regions=num_of_lage_pages o lgpg_size=16 MB
```

2. 사용자 권한을 설정한다.

보안 정책 상 root 이외의 사용자는 Large page를 사용하기 위해서 아래와 같은 권한을 Tiberio 사용자에게 부여해 주어야 한다. chuser 명령을 통해 해당 권한을 설정할 수 있다.

```
$ chuser capabilities=CAP_NUMA_ATTACH,CAP_BYPASS_RAC_VMM,CAP_PROPAGATE <user id>
```

3. 환경설정 파일(.tip)에 다음과 같은 Tiberio 초기화 파라미터를 설정한다.

초기화 파라미터	설정값
TOTAL_SHM_SIZE	1GB(기본값)
USE_HUGE_PAGE	Y

다음은 앞에서 구한 값으로 초기화 파라미터를 설정한 예제이다.

```
$ cat tiber0.tip
.....중간 생략.....
TOTAL_SHM_SIZE=1G
USE_HUGE_PAGE=Y
```

4. Tiber0 서버를 재기동한다.

```
$ tbd0wn

Tiber0 instance terminated (NORMAL mode).

$ tbboot
Listener port = 8629
Change core dump dir to /home/tiber0/tiber07/instance/tiber0.

Tiber0 7

TmaxData Corporation Copyright (c) 2008-. All rights reserved.
Tiber0 instance started up (NORMAL mode).
```

E.2.2. 비활성화

Large page를 활성화면서 변경했던 값을 다시 원래대로 복구한다.

E.3. Solaris

ISM(Intimate Shared Memory) 기능을 통해서 HugePage 기능을 적용할 수 있다. Solaris에서는 HugePage 대신 Large page라는 동일한 의미의 용어를 대신 사용한다. ISM을 사용할 때의 이점은 아래와 같다.

- ISM 공유 메모리는 생성될 때 자동으로 운영체제에 의해서 locked 상태가 된다. 이를 통해 해당 메모리 영역이 디스크로 스왑되지 않도록 보장하며, 운영체제 커널 내부에서 I/O를 위해 더 빠르게 동작하는 locking 매커니즘을 사용할 수 있게 된다.
- 커널의 가상 메모리를 주소와 물리 메모리 주소 간 변환을 위한 메모리 구조를 여러 프로세스 사이에서 공유하게 되므로, CPU 소모 시간을 줄이고 커널 메모리 공간을 절약할 수 있다.
- 시스템의 MMU(Memory Management Unit)에서 지원하는 Large page가 자동으로 ISM 메모리 세그먼트에 할당된다. 이를 통해 page 관리를 위한 메모리 공간이 절약되고, 가상 메모리 주소 변환이 간단해 지므로 시스템 성능을 향상시킬 수 있다.

E.3.1. 활성화

아래와 같은 서버 초기화 파라미터 설정을 통해 활성화 할 수 있다. 해당 파라미터는 Solaris에서만 유효하며 기본값은 Y이다.

```
_USE_ISM=Y
```

이 기능을 켜면 `shmget`으로 공유 메모리를 만들고, `shmat`로 attach할 때 `SHM_SHARED_MMU`라는 옵션이 추가된다. 서버 프로세스와 리스너 프로세스에만 한정되어 적용되며, 서버의 TSM을 attach하는 클라이언트 모듈은 이 옵션을 적용하지 않는다. 만약, 해당 파라미터의 설정값이 Y가 아닌 경우에는 Y로 수정하고 Tiberio 서버를 재기동한다.

E.3.2. 비활성화

서버 초기화 파라미터 `_USE_ISM` 값을 N으로 수정하고 Tiberio 서버를 재기동한다.

```
_USE_ISM=N
```


Appendix F. Undo 테이블 스페이스 크기 산정

본 부록에서는 Tibero 설치 또는 운용 시, 필요한 Undo 테이블 스페이스 용량을 계산하고 조정하는 방법에 대해 설명한다.

F.1. 개요

데이터 갱신이 많은 시스템인 경우, Undo 테이블 스페이스에 대해 I/O가 많이 발생하므로 Undo 테이블 스페이스를 여러 개의 데이터 파일로 구성하여 I/O를 분산시킬 수 있다.

Undo 자동 관리는 다음의 장점을 가진다.

- 설계 및 관리 용이(rollback segment)
- Undo Data의 overwrite 방지
- 영역확장 용이
- 플래시백 쿼리 이용가능

참고

Tibero는 수동 관리는 지원하지 않는다.

F.2. 고려사항

다음은 Undo 테이블 스페이스 크기를 산정할 때 고려할 사항이다.

- 데이터 파일 용량, UNDO_RETENTION, Undo Segment 개수
- 각 Undo Segment 공간은 자동 관리되므로 사용자는 전체 Undo Segment의 최소/최대 개수와 데이터 파일 용량만 고려한다.
 - 최소 Undo Segment 개수(USGMT_ONLINE_MIN)의 기본값은 11이다. 단, Undo Segment 개수 보다 동시 수행 트랜잭션 개수가 많아 부하가 발생하는 경우, 추가 Undo Segment를 생성하는 오버헤드가 생긴다. 이와 같은 경우, 예상되는 동시 수행 트랜잭션 개수 만큼 USGMT_ONLINE_MIN 를 설정하여, 미리 Undo Segment를 생성해놓으면 오버헤드를 줄일 수 있다.

다음은 인스턴스 별 Undo Segment 개수를 확인하는 방법 이다.

```
SELECT COUNT(*) FROM dba_rollback_segs
      WHERE tablespace_name = (SELECT value FROM _vt_parameter WHERE name =
      'UNDO_TABLESPACE')
```

USGMT_ONLINE_MIN 가 이미 생성된 Undo Segment 개수 보다 크게 설정됐다면, (USGMT_ONLINE_MIN - 이미 생성된 Undo Segment 개수) 만큼 부팅 과정에서 추가 생성한다. TAC의 경우, 해당 파라미터를 각 인스턴스 별로 적용할 수 있으며, 적용 후 부팅 시 각 인스턴스 별 Undo Segment 개수가 다른 것을 확인할 수 있다.

- 최대 Undo Segment 개수(USGMT_ONLINE_MAX)는 각 인스턴스 별로 생성할 수 있는 Undo Segment의 상한을 의미한다.

Tibero는 하나의 Undo Segment를 여러 트랜잭션이 공유하여 사용하지 않도록, 동시 수행 트랜잭션 개수가 많을 때, 가능한 추가 Undo Segment를 생성한다.

USGMT_ONLINE_MIN, USGMT_ONLINE_MAX 파라미터는 각 인스턴스 별로 따로 적용되기 때문에 적합한 값을 설정하는 것을 권장한다.

F.3. 용량 계산 방법

아래 기준에 해당하는 Undo 테이블 스페이스 크기를 계산할 수 있다.

- Undo 테이블 스페이스를 생성하는데 필요한 최소 크기
- 예상 동시 수행 트랜잭션수를 기준으로 한 예상 크기
- TPR 정보를 이용한 실제 사용 크기

최초로 DB를 설치하는 경우 최소 크기 및 예상 크기 중 큰 값을 기준으로 Undo 테이블 스페이스 크기를 산정한다.

이후 TPR에서 트랜잭션 최대 부하가 발생했을 때, '초당 Undo block 소모량'을 기준으로 예상 크기와 실제 사용 크기가 맞는지 검증 할 수 있다.

[그림 F.1] Tiberio TPR Report

```

=====
7.17 Undo Statistics
=====

Undo Segment Summary
=====

```

TS#	Undoblks	TX count	Max Con-currency	Snap too old cnt	Nospace errcnt	Active blks	Unexpired blks	Expired blks
1	109,139	94,341	3	0	0	3,655	68,877	25,095,102

```

Undo Segment Stats
=====

```

TS#	End Time	Undoblks	TX count	Max Con-currency	Snap too old cnt	Nospace errcnt	Active blks	Unexpired	Expired blks
1	12/09 21:22:06	2,816	33,466	3	0	0	1,535	33,264	25,130,715
1	12/09 21:18:08	7,573	94,602	3	0	0	1,536	80,355	25,083,624
1	12/09 21:06:57	18,723	108,263	3	0	0	3,581	91,110	25,072,869
1	12/09 20:55:48	6,977	92,336	3	0	0	512	32,245	25,131,734
1	12/09 20:44:39	7,052	92,588	3	0	0	511	32,755	25,131,224
1	12/09 20:33:29	35,741	106,298	3	0	0	12,283	119,266	25,044,713
1	12/09 20:22:20	30,257	132,834	3	0	0	5,630	93,144	25,070,835

다음은 계산 공식에서 사용되는 항목과 설명이다.

항목	설명
USGMT_ONLINE_MIN	인스턴스 당 Undo Segment 최소 개수.
USGMT_ONLINE_MAX	인스턴스 당 Undo Segment 최대 개수(상한)
MAX_SESSION_COUNT	최대 트랜잭션 개수
USGMT_UNIFORM_EXTSIZE * 2	Undo Segment 1개 생성 시 필요한 용량이다. 각 Undo Segment는 최소 2개의 Undo extent가 필요하다.
초당 Undo block 소모량	DB 운영 중 TPR 리포트 Undo Statistics 섹터에서 최대 부하 구간에서의 Undo blocks / 트랜잭션 수행 시간(초) 으로 계산한다
UNDO_RETENTION	undo extent를 재사용하거나 반환하지 않도록 유지 시켜주는 시간, 기본 900초
여유값	필요하다면 충분한 여유값을 두는 것이 좋다.

다음은 위 항목을 토대로 Undo 테이블 스페이스 용량을 계산하는 공식이다.

Undo Segment 1개 생성 시 필요한 용량 = DB_BLOCK_SIZE(블록 사이즈, 기본 8KB) * _USGMT_UNIFORM_EXTSIZE(기본 512) * 2 = 8MB

- 최소 크기 = USGMT_ONLINE_MIN * Undo Segment 1개 생성 시 필요한 용량
- 예상 크기 = MAX_SESSION_COUNT * Undo Segment 1개 생성 시 필요한 용량 + 여유값
- 실제 사용 크기 = 초당 Undo block 소모량 * UNDO_RETENTION * DB_BLOCK_SIZE

Appendix G. Shared Memory 크기 산정

본 부록에서는 Tibero의 SQL Execution Work Area 크기 산정 방법에 대해 설명한다.

G.1. 개요

Tibero Shared Memory는 부팅할 때 고정된 용도로 사용되는 Fixed 영역과 운영 중에 자동으로 할당되어 사용되는 Shared Pool 영역으로 나뉜다.

- Fixed Memory

DB가 부팅할 때 파라미터 값을 바탕으로 크기가 고정되는 영역이다.

- Buffer Cache(Database Buffer)
- Log Buffer
- 그 외 항목 : 전역 변수를 위한 공간, 워킹 스레드 정보와 세션 정보를 관리하는 영역

- Shared Pool Memory

운영 중에 동적으로 할당되어 사용되는 영역이다.

- DD Cache
- PP Cache(Library Cache)
- 그 외 항목

G.2. 고려사항

다음은 Shared Memory 크기를 산정할 때 고려해야 할 사항이다.

- Shared Memory 크기를 운영 중에 동적으로 늘릴 수 없다.
- Buffer Cache와 Shared Pool 사용 패턴을 분석하여 전체 Shared Memory 크기를 알맞게 정해야 한다.
 - Buffer Cache
 - 주요 Workload를 돌려본 후 TPR 리포트의 Buffer Cache Hit율을 보고 판단한다.
 - Hit율이 90% 이하 : Buffer Cache 크기 늘려야 한다.
 - Shared Pool Memory
 - Workload를 돌려본 후 v\$sga의 'SHARED POOL MEMORY' 항목의 사용률 보고 판단한다.

- Shared Pool 사용률이 너무 높으면(Free 영역이 적은 경우) Shared Pool 크기를 늘려야 한다.
- 세션당 최소 1M 이상의 Shared Pool 영역 확보가 필요하다.

G.3. 용량 산정

Shared Memory 각 영역의 크기는 v\$sga에서 확인할 수 있다.

```
SQL> select * from v$sga;
NAME TOTAL USED
-----
SHARED MEMORY 536870912 536870912
FIXED MEMORY 430875880 430875880
SHARED POOL MEMORY 105992872 40974968
SHARED POOL ALLOCATORS 1 1
Database Buffers 357892096 357892096
Redo Buffers 10485760 10485760
```

Shared Memory의 크기는 초기화 파라미터로 설정해서 조회할 수 있다.

- Shared Memory

다음은 TOTAL_SHM_SIZE 파라미터를 설정해서 조회하는 예이다.

```
SQL> show param total_shm_size
NAME TYPE VALUE
-----
TOTAL_SHM_SIZE INT64 536870912
```

- Buffer Cache

다음은 DB_CACHE_SIZE 파라미터를 설정해서 조회하는 예이다.

```
SQL> show param db_cache_size
NAME TYPE VALUE
-----
DB_CACHE_SIZE UINT64 357892096
```

– 싱글 모드 : TOTAL_SHM_SIZE의 2/3

– TAC 모드 : TOTAL_SHM_SIZE의 1/2

- Log Buffer

다음은 LOG_BUFFER 파라미터를 설정해서 조회하는 예이다. (LOG_BUFFER 파라미터의 기본값 : 100MB)

```
SQL> show param log_buffer
NAME TYPE VALUE
```

LOG_BUFFER UINT32 104857600

- Shared Pool Memory

전체 Shared Memory 크기에서 Fixed 영역을 뺀 나머지 영역이다.

TOTAL_SHM_SIZE - [Fixed Memory]

- 싱글 모드

전체 Shared Pool 크기가 _MIN_SHARED_POOL_SIZE보다 커야 한다.

_MIN_SHARED_POOL_SIZE 기본값: 1M * MAX_SESSION_COUNT

- TAC 모드

TAC를 위한 CCC와 CWS 공간이 충분히 확보되어야 한다. CCC와 CWS를 위해 전체 Buffer Cache 크기의 25% 가량을 Shared Pool에서 사용한다.

CCC와 CWS를 위한 공간을 제외한 Shared Pool 영역이 _MIN_SHARED_POOL_SIZE보다 커야 한다.

[전체 Shared Pool 크기] = _MIN_SHARED_POOL_SIZE + [CCC 공간] + [CWS 공간]
+ [Free Shared Pool 크기]

PP Cache와 DD Cache 등을 위한 Free Shared Pool 크기는 CCC와 CWS 공간보다 커야 한다.

[Free Shared Pool 크기] > ([전체 Shared Pool 크기] - _MIN_SHARED_POOL_SIZE) / 2

Appendix H. system_install.sh

본 부록에서는 system_install.sh의 목적과 수행하는 경우 각 옵션에 대하여 설명한다.

H.1. 목적

DB 최초 설치 시 SQL 파일을 수행하기 위해 만들어졌다. 이 셸을 실행하면 현재 디렉터리에 있는 SQL 파일이 수행된다. SQL 파일을 수행하면 role, system user, view, package 등이 생성된다. 기존 system.sh에서 발생하던 불필요한 에러로그들을 제거하였다. system.sh의 옵션 a1, a2, a3, a4는 Y로 설치된다.

H.2. 옵션

```
$ system_install.sh -h
Usage: system_install.sh [option] [arg]
-h : display usage
-p1 password : sys password
-p2 password : syscat password
--except-example-users : do not create example users & roles
--install-xmltype : Install XMLTYPE as object type
```

다음은 각 옵션에 대한 설명이다.

옵션	설명
-h	도움말 화면을 출력한다.
-p1 password	sys 계정의 패스워드를 입력한다. (기본값: tiber0)
-p2 password	syscat 계정의 패스워드를 입력한다. (기본값: syscat)
--except-example-user	Example users 생성 여부를 설정한다. (기본값: N) – Y : Example user를 생성하지 않는다. – N : Example user를 생성한다.
--install-xmltype	XMLTYPE의 내부저장 타입을 지정한다. (기본값:N) – Y : object type으로 생성한다. – N : clob type으로 생성한다.

Appendix I. system_patch.sh

본 부록에서는 system_patch.sh의 목적과 수행하는 경우 각 옵션에 대하여 설명한다.

I.1. 목적

최초 설치 이후 patch 작업을 위해 만들어졌다. 이 셸을 실행하면 현재 디렉터리에 있는 SQL 파일이 수행된다. 기존 system.sh에서 발생하던 불필요한 에러로그들을 제거하였다.

I.2. 옵션

```
$ system_patch.sh -h
Usage: system_patch.sh [option] [arg]
-h : display usage
-p1 password : sys password
-p2 password : syscat password
--create-system-users : create default system users & roles
--except-example-user : do not create example users & roles
--create-sys-profile : create system tables related to profile
--create-job-scheduler-package : register dbms stat job to Job Scheduler
--create-tpv-tables : drop & create TPR tables
--install-xmltype : Install XMLTYPE as object type
```

다음은 각 옵션에 대한 설명이다.

옵션	설명
-h	도움말 화면을 출력한다.
-p1 password	sys 계정의 패스워드를 입력한다. (기본값: tiber0)
-p2 password	syscat 계정의 패스워드를 입력한다. (기본값: syscat)
--create-system-users	시스템 유저 생성 및 권한부여 여부를 설정한다. (기본값: N) - Y : 시스템 유저 생성 및 권한을 부여한다. - N : 데이터를 보존한다.
-create-sys-profile	데이터베이스 사용자의 패스워드 관리 정책을 지정할 수 있는 프로파일을 위한 테이블의 생성 여부를 설정한다. (기본값: N) - Y : 기존 테이블을 삭제한 후 생성한다.

옵션	설명
	- N : 데이터를 보존한다. 프로파일에 대한 자세한 내용은 "Tibero 관리자 안내서"의 "5장 사용자 관리와 데이터베이스 보안"을 참고한다.
--create-job-scheduler-package	데이터베이스 객체의 통계 정보 수집을 위한 Job 스케줄링 등록 여부를 설정한다. (기본값: N) - Y : 통계정보 수집을 위한 Job을 삭제한 후 등록한다. - N : 등록하지 않는다.
--create-tpv-tables	통계정보를 주기적으로 자동 수집하기 위한 TPR 관련 테이블의 생성여부를 설정한다. (기본값: N) - Y : 기존 테이블을 삭제한 후 생성한다. - N : 데이터를 보존한다. APM에 대한 자세한 내용은 "Tibero 관리자 안내서"의 "제17장 Tibero Performance Repository"를 참고한다.
--install-xmltype	XMLTYPE의 내부저장 타입을 지정한다. (기본값:N) - Y : object type으로 생성한다. - N : clob type으로 생성한다.

Appendix J. sample user 설치

본 부록에서는 sample user 스키마와 설치 방법에 대해 설명한다.

J.1. 소개

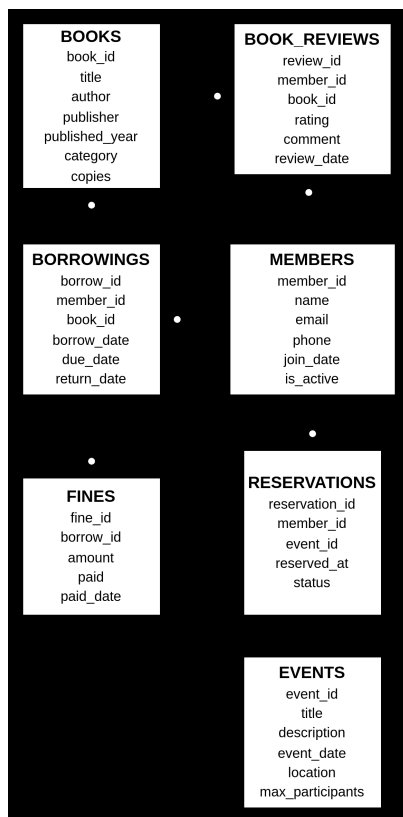
티베로 설치 후 테스트 편의를 위해 두 개의 스키마 더미 데이터를 제공한다.

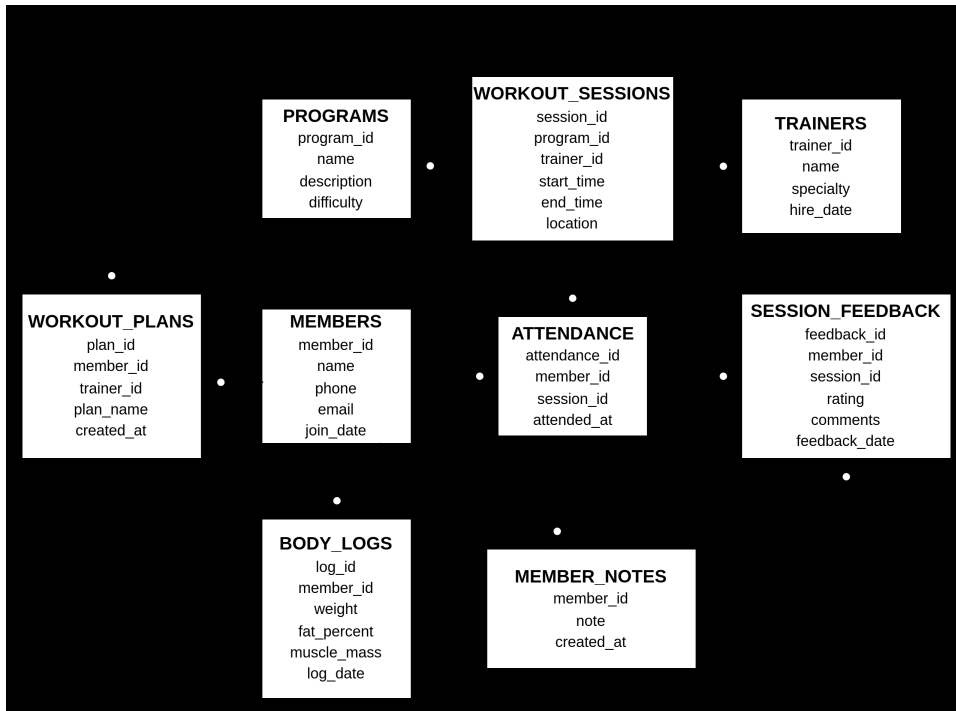
– LIBRARY(LB)

– Fitness(FC)

J.2. 스키마

다음은 각 스키마에 대한 ERD 다이어그램과.





J.3. 설치 및 삭제 방법

다음은 스키마 설치 방법이다.

```
-- tbsql 접속
tbsql sys/tibero

@[$TB_HOME/scripts/sample_user/LB/]lb_install.sql
@[$TB_HOME/scripts/sample_user/FC/]fc_install.sql
-- 비밀번호, 테이블스페이스 입력
```

설치시 LIBRARY에는 2550개, FITNESS에는 63개의 더미데이터가 같이 설치된다.

다음은 삭제 방법이다.

```
-- tbsql 접속
tbsql sys/tibero

@[$TB_HOME/scripts/sample_user/LB/]lb_uninstall.sql
@[$TB_HOME/scripts/sample_user/FC/]fc_uninstall.sql

-- 혹은 더 간단하게 직접 drop해도 된다.
drop user LIBRARY cascade;
drop user FITNESS cascade;
```

색인

A

Activity 로그 파일(act), 29
AIX Shell Limits 파라미터, 14
AIX 커널 파라미터, 13
AUDIT 로그 파일(audit), 29
AutoMounting, 63

C

CREATE DATABASE 문, 19

D

DBMS 로그 파일(dlog), 29
Diagnostic 로그 파일(diag), 29

H

H/W 요구 사항, 5

I

Internal 로그 파일(ilog), 29

J

JDK 설치, 7

L

Linux Shell Limits 파라미터, 11
Linux 커널 파라미터, 10
Listener 로그 파일(lsnr), 29

M

Memory 로그 파일(memlog), 30
Mount Point, 63

N

NTP 서버 설정, 15

S

S/W 요구 사항, 6
sample user, 89
Shared Memory 크기 산정, 81
Solaris Shell Limits 파라미터, 12
Solaris 커널 파라미터, 12
SQLTRACE 로그 파일(sqltrace), 30
system_install.sh, 85
system_patch.sh, 87
system.sh, 67
system.vbs 명령어, 20

T

T-Up, 28
TAC, 53
TAC설치 확인, 56
TAC수동 설치, 56
TAC제거, 56
tbboot, 27
tbboot 명령어, 35
tbcm 명령어, 56
tbdown, 27
tbdown 명령어, 36
tbExport, 28
tblImport, 28
tblistener, 27
tbLoader, 28
tbpc, 28
tbSQL, 27
tbsvr, 27

U

Unmount, 63

V

Volume Manager, 64

ㄷ

덤프 파일, 31
데이터 파일, 29
디스크 용량 확인, 7

ㄹ

로그 파일, 29

ㅁ

마운트, 63

마운트 포인트, 63

멀티 인스턴스 설치, 47

 Unix 환경, 49

 Windows 환경, 47

멀티 인스턴스 제거, 50

 Unix 환경, 51

 Windows 환경, 50

문자 집합(character set), 65

ㅂ

볼륨 매니저, 64

ㅅ

사용자 계정, 37

 SYS, 37

 SYSCAT, 37

 SYSGIS, 37

 TIBERO, 37

 TIBERO1, 37

사용자 환경설정 파일, 59

설치

 수동 설치, 17

수동 설치

 UNIX 환경에서의 설치, 21

 Windows 환경에서의 설치, 17

수동 제거

 UNIX 환경에서의 제거, 40

 Windows 환경에서의 제거, 39

시스템 로그 파일(slog), 29

ㅇ

언마운트, 63

오토 마운팅, 63

운영체제별 커널 파라미터, 9

운영체제별 패키지 설치, 8

ㅈ

제거

 수동 모드, 39

지원 플랫폼 및 운영체제, 3

ㅋ

컨트롤 파일, 29

클라이언트 설치, 41

클라이언트 제거, 45

ㅎ

환경변수, 34

 LD_LIBRARY_PATH, 34

 PATH, 34

 TB_HOME, 34

 TB_SID, 34