

—  
백업/복구  
—

# Tibero

## ShutDown 절차 및 설명

**TMAXTibero**

Copyright © 2025 TmaxTibero. All Rights Reserved

TBAP008

## Copyright Notice

Copyright © 2025 TIBERO Co., Ltd. All Rights Reserved.  
대한민국 경기도 성남시 분당구 정자일로 45 티맥스소프트타워

## Website

www.tmaxtibero.com

## Restricted Rights Legend

All TIBERO Software (Tibero®) and documents are protected by copyright laws and international convention. TIBERO software and documents are made available under the terms of the TIBERO License Agreement and may only be used 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 TIBERO Co., Ltd.

이 소프트웨어(Tibero®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TIBERO Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용권 계약을 준수하는 경우에만 사용 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TIBERO의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물 작성 등의 행위를 하여서는 안 됩니다.

## Trademarks

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

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

## 안내서 정보

안내서 제목: Tibero ShutDown 절차 및 설명

발행일: 2025-11-21

안내서 버전: 1.2

## 제, 개정 이력

안내서 버전	개정일자	개정 사유 및 내용	비고
1.0	2016.04.25	최초 제정	
1.1	2024.02.21	문서 양식 변경, 온라인 매뉴얼 기반 적용	
1.2	2025.11.21	Tibero7 기반 용어 및 로그 변경	

## 1. 개요

- 본 문서는 Tiberio shutdown 후 boot시 진행되는 recovery에 대해 설명한 문서입니다.
- 비정상 종료(abort, abnormal) 시 cache recovery 시간을 단축할 수 있는 방법이 포함되어 있습니다.

## 2. Tiberio down 종류 및 부팅 시 recovery 종류

### 2.1 Tiberio down mode

#### 1) NORMAL

- 일반적인 종료 모드이다.
- Tiberio에 SYS 사용자로 접속한 다음 다른 모든 세션의 접속이 끊어질 때까지 기다린 후 그 다음 서버를 종료시킨다.
- tbdownd를 실행하고 나면 어떤 사용자도 더 이상 데이터베이스에 접속할 수 없게 된다.
- 하지만 tbdownd가 실행되기 전에 이미 데이터베이스에 접속한 사용자는 스스로 접속을 끊을 때까지 제한 없이 데이터베이스를 계속 사용할 수 있다.
- 다음 Tiberio boot시 recovery를 필요로 하지 않는다.

#### 2) POST\_TX

- 모든 트랜잭션이 끝날 때까지 기다리고 나서 Tiberio를 종료하는 모드이다.
- tbdownd POST\_TX 실행이 시작되면 더는 데이터베이스에 접속할 수 없고 이미 열려 있던 세션에서도 새로운 트랜잭션을 시작할 수 없게 된다.
- 다만, 현재 수행 중인 트랜잭션은 커밋 또는 롤백할 때까지 제한 없이 수행할 수 있으며, 커밋이나 롤백을 하는 순간 자동으로 데이터베이스 접속을 종료하게 된다.
- 또한 tbdownd 실행이 시작되면 데이터베이스에 접속한 클라이언트에게 서버 종료를 알리는 메시지를 보내지 않는다. tbSQL 유틸리티 등에서는 클라이언트가 서버 종료를 즉시 알지 못하고 그 다음 명령을 실행할 때 비로소 Tiberio가 종료되었음을 알게 된다.
- 다음 Tiberio boot시 recovery를 필요로 하지 않는다.

#### 3) IMMEDIATE

- 현재 수행 중인 모든 작업을 강제로 중단시키며, 진행 중인 모든 트랜잭션을 롤백하고 Tiberio를 종료하는 모드이다.
- 중단된 모든 트랜잭션을 rollback후에 Tiberio가 종료된다.
- 클라이언트에서 Tiberio 종료를 알지 못하는 것은 POST\_TX 모드와 같다. 트랜잭션이 오래 걸리는 작업 중에 있었다면, 이를 모두 롤백하기 위해서 다소 시간이 걸릴 수 있다.
- 다음 Tiberio boot시 recovery를 필요로 하지 않는다.

#### 4) SWITCHOVER

- Standby DB와 Primary DB를 동기화시킨 후 Primary DB를 NORMAL 모드처럼 종료하는 모드이다.
- Switchover 모드로 종료할 경우 NORMAL 모드와 동일하게 작동을 하며, 추가적으로 Primary에서 생성된 모든 Redo 로그를 Standby에 전송을 한뒤 종료하게 된다.
- Standby 중 하나를 종료한 후 Failover 모드로 기동하면 그 DB가 새로운 Primary가 된다.

#### 5) ABORT

- Tiberio의 프로세스를 강제로 종료하는 모드이다.

- SYS 사용자로 접속한 다음 Tiberio의 MONP 프로세스가 모든 프로세스를 OS의 강제 종료 시그널을 전달하여 강제로 종료시키는 모드이다.
- 서버가 종료된 다음에도 공유 메모리(Shared Memory), 세마포어(Semaphore) 등의 시스템 리소스가 남아있을 수 있으며, 로그 파일이나 데이터 파일도 마찬가지이다. 또한 다음 번에 Tiberio를 기동할 때 파손 복구에 많은 시간이 걸릴 수도 있다.
- 처리중인 SQL 모두 abort 시키고 rollback도 안시킨다.
- 다음 Tiberio boot 시 recovery 가 진행이 된다.
- ABORT는 다음과 같은 경우에만 제한적으로 사용할 것을 권장한다.
  - Tiberio의 내부 에러로 인한 정상적인 종료가 불가능한 경우
  - H/W에 문제가 발생하여 Tiberio를 즉시 종료해야 하는 경우
  - 해킹 등의 비상 상태가 발생하여 Tiberio를 즉시 종료해야 하는 경우

## 6) ABNORMAL

- Tiberio 서버에 접속하지 않고 현재 Tiberio 서버 상태와 상관없이 OS의 강제 종료 시그널을 사용하여 무조건 서버 프로세스를 강제로 종료시키는 모드이다. 따라서 이 모드는 비상 시에 사용하며, 다음번에 Tiberio를 기동할 때 파손 복구 과정이 필요하다.
  - Tiberio가 강제로 종료시키므로 사용하던 시스템 리소스를 해제 못 할 수 있다.
- 따라서 서버가 종료된 다음에도 공유 메모리(Shared Memory), 세마포어(Semaphore) 등의 시스템 리소스가 남아있을 수 있으며, 로그 파일이나 데이터 파일도 마찬가지이다. 또한 다음 번에 Tiberio를 기동할 때 파손 복구에 많은 시간이 걸릴 수도 있다.
- 다음 Tiberio boot 시 recovery 가 진행이 된다.

## 2.2 Tiberio recovery 종류

Tiberio의 recovery는 Crash Recovery / Instance Recovery / Media Recovery 로 나누어진다.

### 1) Crash Recovery

모든 노드(instance)가 다운되어 있는 상태에서, 첫 노드가 boot 될 때의 recovery 이다.

Cache Recovery 와 Transaction Recovery 의 두 단계로 나뉘지며 boot 시 자동 수행된다.

1-1) Cache recovery (Rolling Forward) : 종료 직전의 상태로 DB buffer cache 복원한다. redo로그를 읽어 마지막 체크포인트 이후부터 수행된 트랜잭션을 재현한다.

- 1st Pass : redo를 읽고 복구해야 될 block 을 파악한다.
- 2nd pass : 해당 data block에 대하여 복구를 진행한다.

1-2) Transaction Recovery ( Rolling Backward) : undo를 이용하여 종료 당시 commit 되지 않은 트랜잭션들을 모두 rollback 한다.

### 2) Instance Recovery

TAC에서 일부 노드(instance) 가 종료되었을 때, 나머지 노드 중 하나가 recovering instance 가 되고 죽은 instance 들의 redo 들을 읽어서 복구하는 recovery 과정이다. Crash recovery와 마찬가지로 cache recovery, transaction recovery 순서로 이루어지는데, 현재 살아있는 노드의 redo는 제외하고 적용한다는 점이 crash recovery와 다르고 나머지는 거의 유사하다. TAC에서 어떤 노드가 종료되면 자동으로 수행된다.

### 3) Media recovery

디스크 오류, 파일 삭제 등의 이유로 현재의 데이터 파일을 사용할 수 없는 경우, 특정 시점의 백업 파일을 가져와서 복원하고, 이후의 archive log와 redo log를 이용해서 복구하는 recovery 과정이다. 다른 recovery와는 달리 자동으로 진행되지 않고, 사용자가 alter database recover automatic; 등의 SQL로 실행시킨다.

### 3. WAS 강제 종료 시, DB의 rollback 과정 및 recovery 양 감소 방안

#### 3.1 DB 트랜잭션 rollback

DB종료를 위해 WAS 등을 먼저 종료했을 경우, 트랜잭션이 commit 되지 않은 상태이면, rollback 이 진행이 된다. (rollback 이 진행 중일 때 정상 DB down 명령(normal, post\_tx, immediate, switchover)을 받을 경우, rollback 이 완료될 때까지 기다리게 된다.)

rollback 중인 트랜잭션은 V\$TRANSACTION의 EXP\_RB\_TIME 컬럼을 통해(Tibero5sp1 FS02 이후) 완료되는 시간을 알 수 있다. (단위 : 초)

이 항목은 짧은 트랜잭션에 대해서는 유의미한 값을 얻기 어렵고, 수분에서 몇 시간 이상의 긴 rollback에 대하여 대략적으로 걸리는 시간을 확인할 수 있다.

단, DB내의 부하에 따라 예상시간이 바뀔 수 있다.

```
SQL> select sess_id, start_time, used_blk, EXP_RB_TIME from v$transaction;
```

SESS_ID	START_TIME	USED_BLK	EXP_RB_TIME
33	2025/11/20 19:35:32	142690	970

1 row selected.

버전이 낮아 V\$TRANSACTION 에 EXP\_RB\_TIME 항목이 없다면, USED\_BLK 항목이 줄어드는 속도와 남은 양을 계산하여, 대략적인 완료시간을 유추할 수 있다.

V\$TRANSACTION 의 USED\_BLK 값은 해당 트랜잭션에 사용된 undo block 의 수 이다.

```
SQL> !date
Thu Nov 20 17:27:22 KST 2025
SQL> select sess_id, start_time, used_blk from v$transaction;
```

SESS_ID	START_TIME	USED_BLK
30	2025/11/20 17:07:18	7537

1 row selected.

```
SQL> !date
Thu Nov 20 17:29:14 KST 2025
SQL> select sess_id, start_time, used_blk from v$transaction;
```

SESS_ID	START_TIME	USED_BLK
30	2025/11/20 17:07:18	2448

1 row selected.

### 3.2 DB강제 종료시, recovery 양 감소 방안

V\$TRANSACTION 으로 확인 시, rollback 할 양이 많아 기다리기 어려워 비정상 종료를 해야 하는 상황이라면, 다음 boot시 진행되는 cache recovery 시간을 줄여주기 위해 DB 종료전, 변경된 버퍼캐시의 내용을 디스크에 쓰도록 체크포인트를 수행 하여 복구해야 할 양을 줄일 수 있다.

```
SQL> ALTER SYSTEM CHECKPOINT;
```

체크포인트 진행 시에, 수행되는 시점을 기준으로 DB 버퍼캐시 내에 존재하는 dirty block을 디스크에 flush 하게 된다.

이는 그 시점 이전에 data block에 대해서는 recovery를 할 필요가 없다고 판단하여 recovery의 양이 줄어들게 된다. (checkpoint 진행시에 부하가 들어오지 않는다는 가정이 존재되어야 한다.)

[# dirty block : 버퍼캐시에 존재하는 block 중 내용이 바뀌었지만 아직 디스크와 동기화되지 않은 block.]

## 4. Tiber boot시, 진행되는 recovery

Tibero를 정상적으로 down(normal, post\_tx, immediate, switchover) 시에는 체크포인트가 수행되어, 다음 boot 시 recovery가 진행되지 않는다.

비정상 down(abort, abnormal) 시에는 다음 boot 시에 recovery 가 수행(cache, tx recovery) 되는데, 복구해야 할 양에 따라 오랜 시간이 소요될 수 있다.

Cache recovery의 대부분의 시간 소요는 I/O 수행시 소요되며, 읽어야 될 redolog 크기 + 복구해야 될 data block의 개수에 비례하게 된다.

Cache recovery 가 완료되면, tx recovery를 시작하고, DB는 open 된다. DB open시 tx recovery 완료를 기다리지 않는다. 이후 Tx recovery를 하는 세션은 다른 세션과 동일하게 취급된다.

## 5. 비정상 종료 필요 시, down 절차 및 기동로그

- 1) WAS down 등으로 DB 접속세션 모두 끊음 (jdown, kill -9 [jeus pid])
- 2) 세션 rollback 예상시간 확인

```
SQL> select sess_id, start_time, used_blk, EXP_RB_TIME from v$transaction;
```

SESS_ID	START_TIME	USED_BLK	EXP_RB_TIME
33	2025/11/20 19:35:32	142690	970

1 row selected.

3) 예상시간이 오래 걸려 rollback 을 기다릴 수 없을 경우, checkpoint 수행 후 강제 종료.

```
SQL> ALTER SYSTEM CHECKPOINT;
```

4) tbdwn abort

5) 다음 tboot 시 recovery 진행.

```
[cache recovery 진행]
[RV-39] [I] [0] Preparing cache recovery
[RV-39] [I] DF #0 (ts 0) resetlogs tsn=0, CF resetlogs tsn=0
...
[RV-39] [I] THREAD#0 recovery start tsn progress. (0000.004197a7 => 0000.00419a5d)
[RV-39] [I] THREAD#0 recovery point update (672.40276 => 674.5003)
[RV-39] [I] THREAD#0 logfile scanned. flushed=[674.97487/0000.00419bd9].
[RV-39] [I] THREAD#0 needs replay from 0000.00419a5d (92485 LOG_BLKs)
[RV-39] [I] DF average IO speed: 284 us/block
[RV-39] [I] [0] Starting cache recovery
...
[RV-39] [I] [0] can perform Offline Instance Recovery
...
[cache recovery 1st pass Done.]
[RV-39] [I] [0] Recovery 1st Pass Done
...
[cache recovery 2nd pass Done.]
[RV-39] [I] [0] Recovery 2nd Pass Done
...
[cache recovery 완료]
[RV-39] [I] [0] Cache recovery done: flushed tsn 0000.00419bd9
...
[Tx recovery 진행]
[TXR-37] [I] TX RECOVERY...PREPARE
[TXR-37] [I] TX RECOVERY...START (dead 34)
...
[DB는 open 되고 Tx recovery는 특정세션에서 계속 진행.]
[FRM-38] [I] Server is running at NORMAL mode (bootmode : NORMAL, bootflag : 0)
...
[Tx recovery 완료]
[TXR-37] [I] [0] TX RECOVERY...DONE
```

## 6. 참고사항

- 1) 세션이 종료되었을 때 진행 중이던 트랜잭션이 있다면, SESS\_CLEANUP 이 수행되는데, 이때 tbdwn immediate 명령으로 종료했을 경우 SESS\_CLEANUP 이 완료될 때까지 기다렸다가 DB가 down 된다. 이때 sys.log에서는 30 초 간격으로 나타나는 메시지를 통해 SESS\_CLEANUP 중임을 확인할 수 있다.

```
THROW. ec=ERROR_SERVER_IN_SHUTDOWN(-12039) [ Server instance is shutting down.]  
(csr_id:4294967295) [tbsvr_wthr.c:3078:wthr_get_new_cli_con]
```

**[30초 간격으로 출력]**

```
[FRM-563] [!] shutdown: waiting for releasing SESS lock (sessid=4282, status=SESS_CLEANUP, tid=4282,  
status=WAITING)
```

```
[FRM-563] [!] shutdown: waiting for releasing SESS lock (sessid=4282, status=SESS_CLEANUP, tid=4282,  
status=WAITING)
```

...

- 2) cache recovery 진행 도중 instance 종료 후 재 시작을 하게 된다면 처음부터 다시 시작하게 된다. **cache recovery는 SKIP 할 수 없다.**