

—  
마이그레이션  
—

# Tibero

## 전환 유틸리티 가이드



Copyright © 2024 TmaxTibero. All Rights Reserved

## Copyright Notice

Copyright © 2024 TIBERO Co., Ltd. All Rights Reserved.

대한민국 경기도 성남시 분당구 황새울로 258 번길 29, 티맥스수내타워 우)13595

## 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, 2 차적 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 전환 유틸리티 가이드

발행일: 2024-02-14

소프트웨어 버전: Tibero7 FS02

안내서 버전: 1.0

## 제, 개정 이력

안내서 버전	개정일자	개정 사유 및 내용	비고
1.0	2024.02.14	최초 제정	작성자:정승현, 김유석

## 목록

제1장 Table Migrator.....	4
1. 개요.....	4
2. 수행 방법.....	6
3. 파라미터 지정 .....	7
4. 수행 예제.....	16

# 제1장 Table Migrator

본 장에서는 기존 데이터베이스를 Tibero로 전환하는 Table Migrator 유틸리티에 대해서 설명한다.

## 1. 개요

Table Migrator는 Tibero에서 제공하는 전환 유틸리티로 다른 DBMS에서 구성한 Table 데이터를 Tibero로 옮기는 기능을 수행한다.

### 참고

해당 바이너리는 TechNet에 접속하여 다운로드 > 데이터베이스 > Tibero > T-UP 에서 다운 받을 수 있는 T-UP 파일 안에 포함되어 있다.

### 1.1. 특징

Table Migrator 유틸리티의 특징은 다음과 같다.

- Command 기반이며, 내부적으로 JAVA의 JDBC 드라이버를 이용하여 DBMS에 접속한다.
- 별도 컴퓨터(클라이언트)가 필요하지 않으며 Source 또는 Target 서버에서 수행 가능하다.
- 전체, 사용자별 이관은 불가능하며 Table 단위로 데이터 이관만 가능하다. (스키마 객체 및 Table에 정의된 각종 제약조건은 전환되지 않음)
- 옵션을 사용하여 다양한 방법으로 전환이 가능하다. (SELECT 옵션, DPL, P-DPL 지원)

### 1.2. 실행 전 준비사항

다음은 Table Migrator 유틸리티 실행 전 준비사항이다.

- JRE 1.6 이상을 설치한다.
- Table Migrator 설치 전 아래의 파일이 준비되었는지 확인한다.
  - table\_migrator.zip : Table Migrator 바이너리
  - Source DB의 JDBC 드라이버

예)

DBMS	JDBC 드라이버
Oracle	ojdbc6.jar (Source DB 버전에 맞는 드라이버 사용)
MSSQL	sqljdbc4.jar (Source DB 버전에 맞는 드라이버 사용)

DBMS	JDBC 드라이버
DB2	db2jcc.jar (Source DB 버전에 맞는 드라이버 사용)
MYSQL	mysql-connector-java-5.1.4.jar (Source DB 버전에 맞는 드라이버 사용)
PostgreSQL	postgresql-42.2.27.jre6.jar (Source DB 버전에 맞는 드라이버 사용)
Sybase ASE	jconn3.jar (Source DB 버전에 맞는 드라이버 사용)
Informix	ifxjdbc.jar (Source DB 버전에 맞는 드라이버 사용)

- Table Migrator 바이너리를 Source DB 또는 Target DB(Tibero)에 위치한다.
- 이관 대상 Table을 생성 DDL을 이용하여 Target DB(Tibero)에 미리 생성한다.

### 1.3. 유의사항

Table Migrator 유틸리티를 실행할 때 다음과 같은 사항에 유의한다.

- Source DB의 Table을 이관할 경우 Target DB(Tibero)에 Tablespace, User, Table을 미리 생성 후 TableMigrator를 수행한다.
- DBMS 접속 계정에 DBA 권한이 필요하다. 만약, 고객사의 정책상 DBA 권한을 부여받지 못할 경우 이관 작업을 위한 사용자별 최소한의 권한 예시는 아래와 같다.

계정	권한
Oracle (SOURCE DB)	CONNECT, SELECT ANY TABLE, SELECT ANY DICTIONARY, ALTER SESSION
Tibero (TARGET DB)	CONNECT, SELECT ANY TABLE, SELECT ANY DICTIONARY, RESOURCE, ALTER SESSION

- long, longraw 컬럼을 지원하지 않는다.
- index가 존재하는 Table일 경우 DPL 옵션을 사용하지 못한다.

## 2. 수행 방법

### 2.1. 압축 파일 해제

table\_migrator.zip 바이너리 압축 파일을 해제하면 다음과 같은 파일이 생성된다.

파일	설명
migrator_cli.jar	Table Migrator 클래스이다.
toolcom.jar	Tibero 유틸리티 공통 라이브러리이다.
internal-jdbc-16.jar	Tibero JDBC 드라이버이다.
migrator.bat	Windows 계열 실행 파일이다.
migrator.sh	POSIX 계열 실행 파일이다.
migrator.properties.eg	Table Migrator 컨트롤파일이다.
msllogger-14.jar	로그 관련 인터페이스 클래스이다.
antlr-4.3-complete.jar	문자열 파싱 관련 인터페이스 클래스이다.
postgresql-42.2.27.jre6.jar	PostgreSQL JDBC 드라이버이다.

#### 참고

실행에 필요한 클래스 라이브러리 toolcom.jar, internal-jdbc-16.jar는 Target DB(Tibero)의 버전에 맞는 라이브러리로 교체하여 실행한다.

(라이브러리 기본 위치 : \$TB\_HOME/client/lib/jar)

### 2.2. migrator.sh(또는 .bat) 수정

UNIX 계열인 경우 migrator.sh, Windows 계열인 경우 migrator.bat 파일을 수정한다.

classpath 설정에 Source DB의 JDBC 드라이버가 설정되어 있는지 확인하고 없을 경우 추가한다.

```
java -classpath migrator_cli.jar:ojdbc6.jar:postgresql-42.2.27.jre6.jar:msllogger-14.jar:toolcom.jar:internal-jdbc-16.jar:antlr-4.3-complete.jar com.m.migrator.Main "$@"
```

### 2.3. Table Migrator 수행

다음은 Table Migrator를 실행하는 2가지 예로 UNIX 계열인 경우 migrator.sh, Windows 계열인 경우 migrator.bat 명령어를 입력한다.

- 명령 프롬프트에서 파라미터 지정 실행

```
sh migrator.sh SOURCE_TYPE=ORACLE \
SOURCE_URL=jdbc:oracle:thin:@127.0.0.1:1521:orcl \
SOURCE_USER=system SOURCE_PASSWORD=manager SOURCE_LOGIN_AS=NORMAL \
SOURCE_SCHEMA=scott SOURCE_TABLE=emp \
TARGET_URL=jdbc:internal:thin:@127.0.0.2:8629:t7 \
TARGET_USER=dbtech TARGET_PASSWORD=dbtech
```

- 컨트롤 파일 지정 실행

파라미터의 값을 컨트롤 파일에 설정한 후 실행한다. 만약, 파라미터의 값이 동시에 컨트롤 파일과 명령 프롬프트에 설정된 경우 우선순위는 명령 프롬프트에서 지정한 파라미터 값을 우선한다.

```
sh migrator.sh PROPERTY_FILE=migrator.properties
```

### 3. 파라미터 지정

파라미터를 지정하는 컨트롤 파일은 기본으로 migrator.properties 파일로 지정된다. 다음은 migrator.properties 파일에 설정된 파라미터의 예이다.

```
SOURCE_TYPE=ORACLE
SOURCE_DRIVER=oracle.jdbc.OracleDriver
SOURCE_URL=jdbc:oracle:thin:@localhost:1539:orcl
SOURCE_USER=sys
SOURCE_PASSWORD=oracle
SOURCE_LOGIN_AS=sysdba
SOURCE_SCHEMA=SCOTT
SOURCE_TABLE=t1
SOURCE_TABLE_DEDUPLICATE=N
TARGET_TYPE=DEFAULT
TARGET_DRIVER=com.m.internal.jdbc.MInternalDriver
TARGET_URL=jdbc:internal:thin:@localhost:59166:tibero
TARGET_USER=tibero
TARGET_PASSWORD=tmax
TARGET_SCHEMA=TIBERO
TARGET_TABLE=t1
EXTRACT_PARALLEL=N
EXTRACT_PARALLEL_RANGE_BY_ROWID=N
EXTRACT_THREAD_COUNT=1
SELECT_TSN=17322052
SELECT_CONDITION= c1>1
SELECT_FETCH_SIZE=1024
SELECT_PARTITION=part1
SELECT_SUBPARTITION=subpar1
SELECT_AS_BYTE=N
```

```

DB_FILE_MULTIBLOCK_READ_COUNT=-1
OPTIMIZER_HINT=/*+ FULL(t) PARALLEL(t, 4) */
USE_PARALLEL_HINT=N
USE_OBJECT_PARALLEL_HINT=N
PARALLEL_HINT_CNT=-1
INSERT_METHOD=dpl
COMMIT_SIZE=-1
INDEX_DISABLE=N
INSERT_BATCH=Y
BATCH_THRESHOLD=128
DPL_LOG_UNIT=500
INSERT_PARALLEL=N
INSERT_THREAD_COUNT=1
INSERT_PARTITION=
INSERT_ZERO_LENGTH_STRING_AS_NULL=Y
TRIM=
BYTEA_TO_BLOB=N
TEXT_TO_CLOB=N
MONITORING_LOG=N
MONITORING_LOG_UNIT=500
MONITORING_SESS_STAT=CPU used by this session;parse time cpu;parse count(hard)
CHECK_COLUMN_METADATA=N
ROW_QUEUE_TYPE=ARRAY(10240)
VERIFY_TABLE=N
VERIFY_TABLE_NAME=VERIFY_TBL
LOG_LEVEL=INFO
LOG_FILE_NAME=tablemigrator.log
LOG_DIR=

```

각 파라미터의 설명은 다음과 같다.

- Source 접속 정보 관련

파라미터	설명
SOURCE_TYPE	<p>Source DB의 타입을 설정하는 파라미터이다.</p> <ul style="list-style-type: none"> <li>- ORACLE(기본값) : Source DB가 Oracle일 때 설정하며 SOURCE_LOGIN_AS 파라미터의 설정이 필요하다.</li> <li>- GENERIC : Source DB가 일반 DB일 때 설정한다. 특정 벤더별 API를 사용하지 않고 표준 JDBC API만을 이용하여 동작하며 SOURCE_DRIVER 파라미터의 설정이 필요하다.</li> <li>- DEFAULT : Source DB가 Tiberio일 때 설정한다.</li> </ul>



파라미터	설명
SOURCE_DRIVER	<p>Source DB에 접속할 때 사용하는 JDBC Driver 클래스를 설정하는 파라미터로 SOURCE_TYPE=GENERIC일 경우 설정이 필요하다.</p> <ul style="list-style-type: none"> <li>- Oracle(기본값) : oracle.jdbc.OracleDriver</li> <li>- MSSQL : com.microsoft.sqlserver.jdbc.SQLServerDriver</li> <li>- DB2 : com.ibm.db2.jcc.DB2Driver</li> <li>- MYSQL : com.mysql.jdbc.Driver</li> <li>- PostgreSQL : org.postgresql.Driver</li> <li>- Sybase ASE : com.sybase.jdbc3.jdbc.SybDriver</li> <li>- Informix : com.informix.jdbc.IfxDriver</li> </ul>
SOURCE_URL	<p>Source DB의 JDBC 연결 URL을 설정하는 파라미터이다.</p> <ul style="list-style-type: none"> <li>- Oracle(기본값) : jdbc:oracle:thin:@localhost:1521:orcl</li> <li>- MSSQL : jdbc:sqlserver://127.0.0.1:1433;databaseName=master</li> <li>- DB2 : jdbc:db2://127.0.0.1:50000/logdb</li> <li>- MYSQL : jdbc:mysql://127.0.0.1:3306/mysql</li> <li>- PostgreSQL : jdbc:postgresql://127.0.0.1:5432/postgres</li> <li>- Sybase ASE : jdbc:sybase:Tds:127.0.0.1:7100?ServiceName=sybase</li> <li>- Informix : jdbc:informix-sqli://127.0.0.1:18220/informix:INFORMIXSERVER=informix</li> </ul>
SOURCE_USER	<p>Source DB에 로그인하는 사용자 이름을 설정하는 파라미터이다. 기본적으로 대문자로 인식되며 큰따옴표를 통해 대소문자 구분이 가능하다.</p> <p>(기본값 : sys)</p>
SOURCE_PASSWORD	<p>Source DB에 로그인 하는 사용자 패스워드를 설정하는 파라미터이다.</p> <p>(기본값 : oracle)</p>
SOURCE_LOGIN_AS	<p>Oracle에 접속할 때 권한을 설정하는 파라미터로 SOURCE_TYPE=ORACLE인 경우에만 설정한다.</p> <ul style="list-style-type: none"> <li>- SYSDBA(기본값)</li> <li>- NORMAL</li> </ul>
SOURCE_SCHEMA	<p>Source DB에서 이관하는 Table의 실제 사용자 이름을 설정하는 파라미터이다. 기본적으로 대문자로 인식되며 큰따옴표를 통해 대소문자 구분이 가능하다.</p> <p>(기본값 : SOURCE_USER에서 설정한 사용자 이름)</p>

파라미터	설명
SOURCE_TABLE	Source DB에서 이관하는 Table 이름을 설정하는 파라미터이다. 기본적으로 대문자로 인식되며 큰따옴표를 통해 대소문자 구분이 가능하다. (필수 입력)
SOURCE_TABLE_DEDUPLICATE	dedup 옵션이 사용된 테이블의 이관 동안 발생하는 deadlock 에러를 무시하는 파라미터이다. BATCH_THRESHOLD=2 이상일 경우 데이터 무결성이 보장되지 않으므로 BATCH_THRESHOLD=1인 경우에만 설정한다.  - Y  - N

- Target 접속 정보 관련

파라미터	설명
TARGET_TYPE	Target DB의 타입을 설정하는 파라미터이다.  - DEFAULT : Target DB가 Tibero일 때 설정한다.
TARGET_DRIVER	Target DB에 접속할 때 사용하는 JDBC Driver 클래스를 설정하는 파라미터이다.  (기본값 : com.m.internal.jdbc.MInternalDriver)
TARGET_URL	Target DB의 JDBC 연결 URL을 설정하는 파라미터이다.  (기본값 : jdbc:internal:thin:@localhost:8629:tibero)
TARGET_USER	Target DB에 로그인하는 사용자 이름을 설정하는 파라미터이다. 기본적으로 대문자로 인식되며 큰따옴표를 통해 대소문자 구분이 가능하다.  (기본값 : sys)
TARGET_PASSWORD	Target DB에 로그인하는 사용자 패스워드를 설정하는 파라미터이다.  (기본값 : tibero)
TARGET_SCHEMA	Target DB에 이관 받는 Table의 실제 사용자 이름이다. 기본적으로 대문자로 인식되고 큰따옴표를 통해 대소문자 구분이 가능하다.  (기본값 : TARGET_USER에서 설정한 사용자 이름)
TARGET_TABLE	Target DB에 이관 받는 Table 이름을 설정하는 파라미터이다. 기본적으로 대문자로 인식되고 큰따옴표를 통해 대소문자 구분이 가능하다.  (기본값 : SOURCE_TABLE에서 설정한 Table 이름)

- 데이터 Extractor 관련

파라미터	설명
EXTRACT_PARALLEL	<p>SOURCE DB에서 데이터를 추출하는 Thread의 parallel 동작 여부를 설정하는 파라미터이다. SOURCE_TYPE이 Oracle, default, postgresql인 경우에만 사용 가능하다. row가 rowid Hashing을 통해 나뉘지기 때문에 이관 동안 rowid 변경이 발생하지 않도록 해야한다.</p> <p>- Y</p> <p>- N (기본값)</p>
EXTRACT_PARALLEL_RANGE_BY_ROWID	<p>rowid의 범위를 지정하여 row를 나누도록 설정하는 파라미터이다. SOURCE_TYPE=Oracle, EXTRACT_PARALLEL=Y인 경우에만 설정 가능하며 ora_hash function을 지원하지 않는 Oracle 9버전 이하에서 성능적 이점을 가진다. IOT 테이블은 해당 옵션을 사용할 수 없다.</p> <p>- Y</p> <p>- N</p>
EXTRACT_THREAD_COUNT	<p>데이터를 추출하는 Thread의 수를 설정하는 파라미터이다. EXTRACT_PARALLEL_RANGE_BY_ROWID 옵션을 사용하는 경우 테이블의 extent 수에 따라 적절한 값으로 설정된다.</p> <p>(기본값 : 1)</p>
SELECT_TSN	<p>특정 TSN (Oracle에서는 SCN) 시점 이후의 데이터를 추출하도록 설정하는 파라미터이다.</p> <p>(기본값 : 미사용)</p>
SELECT_CONDITION	<p>Table 데이터 전체가 아닌 조건절에 해당하는 데이터만 추출하도록 설정하는 파라미터이다.</p> <p>(기본값 : 미사용)</p>
SELECT_FETCH_SIZE	<p>Source DB에서 추출해온 결과 set을 fetch 해오는 단위 (row 수)를 설정하는 파라미터이다.</p> <p>(기본값 : 1024)</p>
SELECT_PARTITION	<p>파티션 Table에서 지정한 파티션에서만 데이터를 추출하도록 설정하는 파라미터이다.</p> <p>(기본값 : 미사용)</p>
SELECT_SUBPARTITION	<p>파티션 Table에서 지정한 서브 파티션에서만 데이터를 추출하도록 설정하는 파라미터이다. SELECT_PARTITION 옵션과 같이 사용할 경우 SELECT_PARTITION 옵션이 적용된다.</p> <p>(기본값 : 미사용)</p>

파라미터	설명
SELECT_AS_BYTE	<p>Table의 char, varchar와 같은 문자열을 저장하기 위한 열에 데이터베이스 설정과 다른 캐릭터셋을 사용하여 실제 문자열이 저장되면 문자열 형태로 데이터를 가져올 때 문자열이 깨질 수 있다. 이를 방지하기 위하여 문자열이 아닌 바이너리 형태로 데이터를 가져오고, 바이너리 형태로 Tiberio에 옮기도록 설정하는 파라미터이다.</p> <p>- Y</p> <p>- N (기본값)</p>
DB_FILE_MULTIBLOCK_READ_COUNT	<p>이관 전 해당 세션에 대한 DB_FILE_MULTIBLOCK_READ_COUNT 값을 설정하는 파라미터이다. SOURCE_TYPE=Oracle인 경우에만 설정 가능하며 값을 -1로 설정 시 동작하지 않는다.</p> <p>(기본값 : -1)</p>
OPTIMIZER_HINT	<p>SOURCE DB로 부터 데이터를 SELECT 할 때 사용할 힌트를 설정하는 파라미터이다. 테이블 이름은 't'를 사용하며 SOURCE_TYPE=Oracle, SOURCE_TYPE=Default인 경우에만 사용 가능하다. 만약 USE_PARALLEL_HINT 옵션과 동시에 사용된다면, OPTIMIZER_HINT 옵션이 우선적으로 적용된다.</p> <p>- 예 : OPTIMIZER_HINT=/*+ FULL(t) PARALLEL(t, 4)</p>
USE_PARALLEL_HINT	<p>SOURCE DB에서 데이터를 SELECT 할 때 PARALLEL 힌트 사용 여부를 설정하는 파라미터이다. SOURCE_TYPE=Oracle, SOURCE_TYPE=Default인 경우에만 사용 가능하다.</p> <p>- Y</p> <p>- N (기본값)</p>
USE_OBJECT_PARALLEL_HINT	<p>PARALLEL 힌트 사용 시 Table 이름을 명시할 수 있도록 설정하는 파라미터이다. 해당 옵션을 활성화하기 위해서는 USE_PARALLEL_HINT=Y 설정이 필요하며, SOURCE_DB가 Default, Oracle인 경우에만 사용 가능하다.</p> <p>- Y</p> <p>- N (기본값)</p>
PARALLEL_HINT_CNT	<p>PARALLEL 힌트 사용 시 DOP를 설정하는 파라미터이다. 해당 옵션을 활성화하기 위해서는 USE_PARALLEL_HINT=Y 설정이 필요하다</p> <p>(기본값 : -1)</p>

- 데이터 Loader 관련

파라미터	설명
INSERT_METHOD	<p>데이터의 INSERT 방식을 지정하는 파라미터이다.</p> <p>- DPL(기본값) : DIRECT PATH LOADING</p> <p>- CPL : CONVENTIONAL PATH LOADING</p>
COMMIT_SIZE	<p>TARGET_DB에 데이터를 commit하는 사이즈를 지정하는 파라미터이다. INSERT_METHOD=cpl인 경우에만 동작하며 COMMIT_SIZE=-1로 설정하는 경우 기본값인 commit only once로 동작한다.</p> <p>(기본값 : commit once after loading)</p>
INDEX_DISABLE	<p>TARGET_TABLE의 INDEX를 자동으로 DISABLE 하도록 설정하는 파라미터이다.</p> <p>- Y</p> <p>- N (기본값)</p>
INSERT_BATCH	<p>Table의 데이터를 Batch Insert 방식으로 load 하도록 설정하는 파라미터이다. INSERT_METHOD=CPL인 경우에만 동작한다.</p> <p>- Y</p> <p>- N</p>
BATCH_THRESHOLD	<p>TARGET_DB에 Batch를 실행할 때 단위(row 수)를 설정하는 파라미터이다. INSERT_METHOD=CPL, INSERT_BATCH=Y 인 경우에만 동작하며 -1 값을 입력하면 동작하지 않는다.</p> <p>(기본값 : 32)</p>
DPL_LOG_UNIT	<p>로그를 출력하는 row의 로딩 단위를 설정하는 파라미터이다. INSERT_METHOD=DPL인 경우에만 사용 가능하며 0으로 설정하는 경우 로그가 출력되지 않는다.</p> <p>(기본값 : 500)</p>
INSERT_PARALLEL	<p>데이터를 load하는 Thread의 parallel 동작 여부를 설정하는 파라미터이다.</p> <p>- Y(기본값)</p> <p>- N</p>
INSERT_THREAD_COUNT	<p>데이터 loader thread 개수를 설정하는 파라미터이다. INSERT_PARALLEL=Y인 경우에만 동작한다.</p> <p>(기본값 : 4)</p>
INSERT_PARTITION	<p>지정한 파티션에 이관할 경우 설정하는 파라미터이다.</p> <p>(기본값 : 미사용)</p>

파라미터	설명
INSERT_ZERO_LENGTH_STRING_AS_NULL	<p>길이가 0인 String 타입 데이터가 이관 시 NULL로 들어가도록 설정하는 파라미터이다. INSERT_ZERO_LENGTH_STRING_AS_NULL=N으로 설정 시 길이가 0인 데이터는 공백("")으로 들어가며 SOURCE_TYPE=Oracle인 경우 설정이 무시된다.</p> <p>- Y(기본값)</p> <p>- N</p>
TRIM	<p>CHAR 타입 데이터의 선행, 후행에 있는 pad 문자를 없앤다.</p> <p>- Y</p> <p>- N(기본값)</p>

- POSTGRESQL 관련

파라미터	설명
BYTEA_TO_BLOB	<p>bytea 타입을 blob 타입으로 변환하도록 설정하는 파라미터이다.</p> <p>- Y</p> <p>- N(기본값)</p>
TEXT_TO_CLOB	<p>text 타입을 clob 타입으로 전화하도록 설정하는 파라미터이다.</p> <p>- Y</p> <p>- N(기본값)</p>

- 일반 설정

파라미터	설명
MONITORING_LOG	<p>모니터링 로그 출력 여부를 설정하는 파라미터이다. 이관에 사용된 동작에 대한 평균 시간 로그를 출력한다.</p> <p>- Y</p> <p>- N (기본값)</p>
MONITORING_LOG_UNIT	<p>모니터링 로그에서 시간을 평균 낼 때 사용할 로우의 수를 설정하는 파라미터이다.</p> <p>(기본값 : 500)</p>

파라미터	설명
MONITORING_SESS_STAT	<p>모니터링할 stat을 설정하는 파라미터이다. 여러개의 stat을 설정할 경우 ;로 구분해주며 해당 옵션은 SOURCE_TYPE=ORACLE인 경우에 만 설정 가능하다.</p> <p>(기본값 : 미사용)</p> <p>- 예 : MONITORING_SESS_STAT=CPU used by this session;parse time cpu;parse count (hard)</p>
CHECK_COLUMN_METADATA	<p>INSERT_METHOD=dpl인 경우 이관하기 전에 SOURCE DB table과 TARGET DB table의 메타데이터를 확인하는 옵션이다.</p> <p>- Y(기본값)</p> <p>- N</p>
ROW_QUEUE_TYPE	<p>추출한 데이터를 저장하는 queue 타입을 설정하는 파라미터이다. EXTRACT_PARALLEL=Y, INSERT_PARALLEL=Y 인 경우 SYNCHRO 타입을 사용하는 것이 성능상 좋으며 ARRAY 타입은 사이즈를 특정할 수 있다.</p> <p>- ARRAY (기본값 : 10240)</p> <p>- SYNCHRO</p>
VERIFY_TABLE	<p>이관 검증 테이블 사용 여부를 설정하는 파라미터이다.</p> <p>- Y</p> <p>- N (기본값)</p>
VERIFY_TABLE_NAME	<p>검증 테이블 이름을 설정하는 파라미터이다.</p> <p>(기본값 : VERIFY_TBL)</p>
LOG_LEVEL	<p>로그 레벨을 설정하는 파라미터이다.</p> <p>(기본값 : INFO)</p>
LOG_FILE_NAME	<p>로그 파일 이름을 설정하는 파라미터이다.</p> <p>(기본값 : tableMigrator.log)</p>
LOG_DIR	<p>로그 파일이 생성되는 디렉토리 경로를 설정하는 파라미터이다.</p> <p>(기본값 : Migrator 실행 경로)</p>

## 4. 수행 예제

다음은 Table Migrator 유틸리티로 Oracle(Source DB)에서 Tiberio(Target DB)로 전환을 수행하는 예제이다.

### 4.1. 테스트 환경 구성

Oracle에 테스트 Table, 데이터, 접속할 계정을 생성한다.

```
--DBTECH 계정에 Tablespace, User, Table, Table Data 생성
SQL> create tablespace DBTECH_TBS
2 datafile 'dbtech_tbs01.dtf' size 50M
3 AUTOEXTEND ON NEXT 20M MAXSIZE 5G
4 EXTENT MANAGEMENT LOCAL AUTOALLOCATE;

Tablespace created.

SQL> create user dbtech identified by dbtech
2 default tablespace DBTECH_TBS;

User created.

SQL> grant connect, resource to dbtech;

Grant succeeded.

CONNECT dbtech/dbtech;

SQL> create table board (no number,
2 empno number(4),
3 title varchar(20),
4 contents clob,
5 regdate date default sysdate,
6 hit number default 0)
7 PCTFREE 10
8 INITRANS 2;

Table created.

SQL> INSERT INTO BOARD VALUES
(1, 7369, 'introduction', 'Hello, my name is SMITH.', sysdate, 0);

1 row created.

SQL> INSERT INTO BOARD VALUES
(2, 7499, 'introduction', 'Hello, my name is ALLEN.', sysdate, 0);

1 row created.

SQL> INSERT INTO BOARD VALUES
```



```

(3, 7521, 'introduction', 'Hello, my name is WARD.', sysdate, 0);

1 row created.

SQL> INSERT INTO BOARD VALUES
(4, 7566, 'introduction', 'Hello, my name is JONES.', sysdate, 0);

1 row created.

SQL> INSERT INTO BOARD VALUES
(5, 7654, 'introduction', 'Hello, my name is MARTIN.', sysdate, 0);

1 row created.

SQL> commit;

Commit complete.

--Migration할 때 접속할 계정(DBTECH_TEST) 생성
SQL> CONNECT /as sysdba
Connected.

SQL> create user dbtech_test identified by dbtech_test
2 default tablespace DBTECH_TBS;

User created.

--DBTECH_TEST 계정에 권한 부여(connect)
SQL> grant connect to dbtech_test;

Grant succeeded.

--DBTECH_TEST 계정에 Table 조회 권한 부여(select on dbtech.board)
SQL> grant select on dbtech.board to dbtech_test;

Grant succeeded.

--DBTECH_TEST 계정에 권한 부여(select any dictionary)
SQL> grant select any dictionary to dbtech_test;

Grant succeeded.

```

## 4.2. 전환 수행

Oracle에 테스트 Table, 데이터, 접속할 계정을 생성한다.

```

SQL> create tablespace DBTECH_TBS
2 datafile 'dbtech_tbs01.dtf' size 50M
3 AUTOEXTEND ON NEXT 20M MAXSIZE 5G
4 EXTENT MANAGEMENT LOCAL AUTOALLOCATE;

```

Tablespace 'DBTECH\_TBS' created.

```
SQL> create user dbtech identified by dbtech
      2 default tablespace DBTECH_TBS;
```

User 'DBTECH' created.

```
SQL> grant connect, resource, select any dictionary to dbtech;
```

Granted.

```
SQL> conn dbtech/dbtech
Connected to Tibero.
```

```
SQL> create table board (no number,
      2                      empno number(4),
      3                      title varchar(20),
      4                      contents clob,
      5                      regdate date default sysdate,
      6                      hit number default 0)
      7 PCTFREE 10
      8 INITRANS 2;
```

Table 'BOARD' created.

```
SQL> exit
Disconnected.
```

```
$ ls -lrt
```

```
-rw-r--r--. 1 tiberio dba 1340431 Nov 15 15:29 antlr-4.3-complete.jar
-rw-r--r--. 1 tiberio dba    1376 Nov 15 15:29 ANTLR4_License.txt
-rw-r--r--. 1 tiberio dba 1604783 Nov 15 15:29 internal-jdbc-16.jar
-rw-r--r--. 1 tiberio dba    167 Nov 15 15:29 migrator.bat
-rw-r--r--. 1 tiberio dba    8313 Nov 15 15:29 migrator.properties.eg
-rw-r--r--. 1 tiberio dba    170 Nov 15 15:29 migrator.sh
-rw-r--r--. 1 tiberio dba 167137 Nov 15 15:29 migrator_cli.jar
-rw-r--r--. 1 tiberio dba 352917 Nov 15 15:29 mslllogger-14.jar
-rw-r--r--. 1 tiberio dba    1320 Nov 15 15:29 pgjdbc_License.txt
-rw-r--r--. 1 tiberio dba 850563 Nov 15 15:29 postgresql-42.2.27.jre6.jar
-rw-r--r--. 1 tiberio dba    4054 Nov 15 15:29 readme.txt
-rw-r--r--. 1 tiberio dba 123556 Nov 15 15:29 toolcom.jar
-rwxr-xr-x. 1 tiberio dba 4210517 Nov 16 13:54 ojdbc6.jar
```

```
$ssh migrator.sh SOURCE_TYPE=ORACLE SOURCE_URL=jdbc:oracle:thin:@localhost:1525:oraclesb
SOURCE_USER=dbtech_test SOURCE_PASSWORD=dbtech_test SOURCE_LOGIN_AS=NORMAL
SOURCE_SCHEMA=dbtech SOURCE_TABLE=board
TARGET_URL=jdbc:internal:thin:@localhost:6213:JSH_T7_DB TARGET_USER=dbtech
TARGET_PASSWORD=dbtech TARGET_SCHEMA=dbtech TARGET_TABLE=board
INFO[16:51:45,676][main]TableMigrator.initAndDoMigrate()(1429)
```

```
*****
```

```
*
```

```
* Start Log for table migrator.
```

```
* Vendor: tiberio
```

```

*      Product: table migrator
*      Version: 263750
*
* Trace is started on 16:51:45.675
*      Log level: INFO
*
*****

INFO[16:51:45,679][main]TableMigrator.initAndDoMigrate()(1431)
=== Parameters ===
SOURCE_TYPE=ORACLE
SOURCE_DRIVER=oracle.jdbc.OracleDriver
SOURCE_URL=jdbc:oracle:thin:@localhost:1525:oraclesb
SOURCE_USER=DBTECH_TEST
SOURCE_PASSWORD=*****
SOURCE_LOGIN_AS=NORMAL
SOURCE_SCHEMA=dbtech
SOURCE_SCHEMAORIGN=dbtech
SOURCE_TABLE=board
SELECT_SUBPARTITION=
BATCH_THRESHOLD=128
EXTRACT_PARALLEL=N
EXTRACT_THREAD_COUNT=1
SOURCE_TABLE_DEDUPLICATE=N
USE_PARALLEL_HINT=N
PARALLEL_HINT_CNT=-1
USE_OBJECT_PARALLEL_HINT=N
TARGET_TYPE=DEFAULT
TARGET_DRIVER=com.m.internal.jdbc.MInternalDriver
TARGET_URL=jdbc:internal:thin:@localhost:6213:JSH_T7_DB
TARGET_USER=DBTECH
TARGET_PASSWORD=*****
TARGET_SCHEMA=dbtech
TARGET_TABLE=board
COMMIT_SIZE=-1
INDEX_DISABLE=N
SELECT_FETCH_SIZE=1024
SELECT_AS_BYTE=N
DB_FILE_MULTIBLOCK_READ_COUNT=-1
INSERT_METHOD=dpl
INSERT_BATCH=Y
INSERT_PARALLEL=N
INSERT_THREAD_COUNT=1
INSERT_ZERO_LENGTH_STRING_AS_NULL=Y
DPL_LOG_UNIT=500
LOG_FILE_NAME=tablemigrator.log
LOG_LEVEL=INFO
CHECK_COLUMN_METADATA=N
=====
INFO[16:51:45,784][TabMig[16:51:45.680]]TbCmdConnection.connect()(58) Connect To The
Source- URL : jdbc:oracle:thin:@localhost:1525:oraclesb ID : DBTECH_TEST
INFO[16:51:45,983][TabMig[16:51:45.680]]TbCmdConnection.connect()(58) Connect To The
Target- URL : jdbc:internal:thin:@localhost:6213:JSH_T7_DB ID : DBTECH

```

```

INFO[16:51:46,228][TabMig[16:51:45.680]]TableMigrator.migrate()(256) SOURCE_TYPE :
ORACLE
INFO[16:51:46,228][TabMig[16:51:45.680]]TableMigrator.migrate()(257) SOURCE_SCHEMA :
dbtech
INFO[16:51:46,228][TabMig[16:51:45.680]]TableMigrator.migrate()(275) SOURCE_TABLE :
board
INFO[16:51:46,229][TabMig[16:51:45.680]]TableMigrator.migrate()(277) TARGET_TYPE :
DEFAULT
INFO[16:51:46,229][TabMig[16:51:45.680]]TableMigrator.migrate()(278) TARGET_SCHEMA :
dbtech
INFO[16:51:46,229][TabMig[16:51:45.680]]TableMigrator.migrate()(293) TARGET_TABLE :
board
INFO[16:51:46,236][TabMig[16:51:45.680]]OracleTableInfo.getTableInfo()(52) SELECT
column_name, data_type, virtual_column, hidden_column FROM DBA_TAB_COLS WHERE OWNER=? AND
TABLE_NAME=? order by column_id
INFO[16:51:46,292][TabMig[16:51:45.680]]OracleTableInfo.getTableInfo()(55) setString(1)
: DBTECH
INFO[16:51:46,292][TabMig[16:51:45.680]]OracleTableInfo.getTableInfo()(58) setString(2)
: BOARD
INFO[16:51:46,308][TabMig[16:51:45.680]]OracleTableInfo.getTableInfo()(125) SELECT
iot_type FROM dba_tables WHERE OWNER = ? and TABLE_NAME = ?
INFO[16:51:46,309][TabMig[16:51:45.680]]OracleTableInfo.getTableInfo()(128) setString(1)
: DBTECH
INFO[16:51:46,309][TabMig[16:51:45.680]]OracleTableInfo.getTableInfo()(131) setString(2)
: BOARD
INFO[16:51:46,311][TabMig[16:51:45.680]]TableMigrator.migrate()(375) start extracting
thread...
INFO[16:51:46,315][TabMig[16:51:45.680].E0]OracleDataExtractor.run()(86) run
INFO[16:51:46,315][TabMig[16:51:45.680].E0]DataExtractor.loggingTabAndColInfo()(531)
Extract Information
INFO[16:51:46,316][TabMig[16:51:45.680].E0]DataExtractor.loggingTabAndColInfo()(532)
table name : DBTECH.BOARD
INFO[16:51:46,316][TabMig[16:51:45.680].E0]DataExtractor.loggingTabAndColInfo()(538)
columns [6] - "NO" "EMPNO" "TITLE" "REGDATE" "HIT" "CONTENTS"
INFO[16:51:46,318][TabMig[16:51:45.680].E0]OracleDataExtractor.run()(93)
TabMig[16:51:45.680].E0 - sql : SELECT '0', t."NO", t."EMPNO", t."TITLE", t."REGDATE",
t."HIT", t."CONTENTS" FROM DBTECH.BOARD t
INFO[16:51:46,327][TabMig[16:51:45.680]]TableMigrator.migrate()(536) start loading
thread...
INFO[16:51:46,328][TabMig[16:51:45.680]]DataLoader.setRemainThr()(85) Set Remain Threads
[0], to [1]
INFO[16:51:46,332][L0]TbDPLDataLoader.setDirPathStream()(525) DPL Loading Information
INFO[16:51:46,332][L0]TbDPLDataLoader.setDirPathStream()(526) table name : dbtech.board
INFO[16:51:46,332][L0]TbDPLDataLoader.setDirPathStream()(532) columns [6] - "NO" "EMPNO"
"TITLE" "REGDATE" "HIT" "CONTENTS"
INFO[16:51:46,337][TabMig[16:51:45.680].E0]OracleDataExtractor.run()(282)
Total Extract Time : 20ms
Time For Putting Row on Queue : 0ms
Time For Executing Query Time : 12ms
Time For Executing Fetch Time : 4ms
Percentage of Query and Fetch => 80.00%
INFO[16:51:46,337][TabMig[16:51:45.680].E0]OracleDataExtractor.run()(283) TOTAL
Extracted ROWS: 5 (source schema = DBTECH, source table = BOARD)

```

```

INFO[16:51:46,338][TabMig[16:51:45.680]]TableMigrator.migrate()(580) Extractor termFlag
: [true], isError : [false]
INFO[16:51:46,338][TabMig[16:51:45.680].E0]DataExtractor.setTermFlag()(543) Set termFlag
and isError : [false]
INFO[16:51:46,338][TabMig[16:51:45.680].E0]DataExtractor.setTermFlag()(544) complete
INFO[16:51:46,354][L0.TbDPLDataLoader0]DataLoaderManager.run()(47) run
INFO[16:51:46,361][L0.TbDPLDataLoader0]DataLoaderManager.run()(240) Total Insert Time :
29ms
Time For Taking Row From Queue: 0ms
Time For Executing Batch : 4ms
Time For Executing Commit : 2ms
Percentage of Batch and Commit Time => 20.69%
INFO[16:51:46,361][L0.TbDPLDataLoader0]DataLoaderManager.run()(242)
[L0.TbDPLDataLoader0]1700121106361 TOTAL Loaded ROWS: 5 (target schema = dbtech, target
table = board)
INFO[16:51:46,374][TabMig[16:51:45.680]]TableMigrator.migrate()(613) Loader remainThr :
[0], isError : [false]
INFO[16:51:46,374][L0.TbDPLDataLoader0]DataLoader.finish()(80) Decrement remainThr : [0]
and isError : [false]
INFO[16:51:46,375][TabMig[16:51:45.680]]TableMigrator.resetThrs()(1359) terminate
extract thread...1
INFO[16:51:46,375][TabMig[16:51:45.680]]TableMigrator.resetThrs()(1373) terminate loader
thread...1
INFO[16:51:46,375][TabMig[16:51:45.680]]TableMigrator.migrate()(627) [VERIFICATION]
Source Info : dbtech.board : 5 Rows, Target Info : dbtech.board : 5 Rows [OK]
INFO[16:51:46,375][TabMig[16:51:45.680]]TableMigrator.migrate()(686) extract conneciton
close
INFO[16:51:46,377][TabMig[16:51:45.680]]TableMigrator.migrate()(695) loader connection
close
INFO[16:51:46,377][TabMig[16:51:45.680]]TableMigrator.migrate()(705) Elapsed Time
(milliseconds) : 697
INFO[16:51:46,377][TabMig[16:51:45.680]]TableMigrator.migrate()(707) End Migration

```