



Tibero RDBMS 4 Utility Guide

Tibero RDBMS 4 데이터베이스 시스템의 관리자가 작업을 수행하기 위해 필요한 관련 유틸리티 사용법을 알아보도록 한다.



Contents

- 1. Tibero 4 Utility 소개 2**
- 2. tbSQL..... 3**
 - 2.1. Tibero 데이터베이스 접속을 위한 환경설정4
 - 2.2. tbSQL 접속 및 종료 5
 - 2.3. 간단한 SQL Test 6
- 3. tbMigrator 11**
- 4. tbExport 15**
 - 4.1. 전체 데이터베이스 모드로 tbExport 16
 - 4.2. 사용자 모드로 tbExport 17
 - 4.3. 테이블 모드로 tbExport 18
- 5. tbImport 19**
 - 5.1. 전체 데이터베이스 모드로 tbImport 20
 - 5.2. 사용자 모드로 tbImport 20
 - 5.3. 테이블 모드로 tbImport 20
- 6. tbLoader 21**
- 7. tbPROF 27**

Update History

Date	Worker	Comments
2011.03.11	박근용	문서서식업데이트
2009.10.20	남유진	문서서식업데이트 및 내용보완•교정
2009.06.20	백서현	최초작성

1. Tibero 4 Utility 소개

Tibero4에서 제공하고 있는 Utility의 종류와 기능은 다음과 같다.

- **tbSQL** Tibero에서 제공하는 텍스트 기반 대화형 SQL명령어 처리기
- **tbMigrator** 다른 DBMS가 구성한 데이터베이스 전체 또는 일부를 Tibero 데이터베이스로 옮기는 것을 도와주는 유틸리티
- **tbExport** 논리적인 백업을 위해 제공하는 Tibero export 유틸리티
- **tbImport** tbExport에 의하여 생성된 Export file을 Tibero데이터베이스로 import하는 유틸리티
- **tbLoader** 대량의 데이터를 한꺼번에 적재하는 유틸리티
- **tbProf** SQL Trace 파일 생성 유틸리티

***Tibero utility 위치 : \$TB_HOME/client/bin

***필수 환경 : java Hot Spot 1.5.0_17 이상

2. tbSQL

Tibero 4에서 제공하고 있는 텍스트 기반 대화형 SQL명령어 처리기이다.

Tibero 서버가 제공하는 대부분의 기능은 tbSQL을 이용하여 접근할 수 있다. 여기서는 tbSQL접속 및 종료방법과 tbSQL을 이용하여 수행할 수 있는 기본 작업을 테스트해본다.

[tbSQL 도움말 확인]

```
[tibero:/home/tibero]$tbsql --help
Usage: tbsql [options] [logon] [script]

options
-----
-h,--help      Displays this information
-v,--version   Displays version information
-s,--silent    Sets silent mode. Does not display the start-up message,
               prompts and commands
-i,--ignore    Ignore the logon script (eg, tbsql.login)

logon
-----
[username[/password[@datasource]]]

script
-----
@filename[.ext] [parameters]
```

*Option:

- h : tbSQL정보표시
- v : 버전정보표시
- s : 프롬프트 및 명령 표시를 숨기는 자동 모드 설정
- i : tbSQL 실행 시 세션 환경 파일(tbsql.login)을 무시

2.1. Tibero 데이터베이스 접속을 위한 환경설정

tbSQL을 통해 Tibero 데이터베이스 접속을 하기 위해서는 클라이언트 접속 설정파일 `tbnet_alias.tbr`의 정보가 Tibero 데이터베이스의 정보(`$TB_SID.tip`)와 일치하는지 확인한다.
(Tibero 4 SP1 부터는 `tbnet_alias.tbr`파일명이 `tbdsn.tbr`로 변경됨.)

[확인사항]

- IP
- PORT
- DB_NAME

[`$TB_HOME/config/$TB_SID.tip`]

```
[tibero:/home/tibero/tibero4/config]$vi tibero.tip
#-----
# tibero 초기화 parameter
#-----
DB_NAME=tibero
LISTENER_PORT=8629
CONTROL_FILES=/home/tibero/tbdata/c1.ctl
DB_CREATE_FILE_DEST=/home/tibero/tbdata
DBWR_CNT=1
.....이하 생략....
```

[`$TB_HOME/client/config/tbnet_alias.tbr`]

```
#-----
# /home/tibero/tibero4/client/config/tbnet_alias.tbr
# Network Configuration File.
tibero=(
  IP=localhost
  PORT=8629
  DB_NAME=tibero
)
```

접속정보가 올바르게 설정되어야 tbSQL을 통해 Tibero데이터베이스에 접속할 수 있다.

2.2. tbSQL 접속 및 종료

데이터베이스를 NORMAL모드로 기동시킨 후, tbsql SYS계정으로 접속한 후 종료해본다.

[접속방법]

```
$ tbsql SYS계정
```

[tbSQL 접속]

```
[tibero:/home/tibero]$tbsql sys/tibero

tbSQL 4

TmaxSoft, Co. Copyright(C) 2001-2009. All rights reserved.

Connected to Tibero.

SQL>
```

[종료방법]

```
SQL>exit( or quit)
```

[tbSQL 종료]

```
SQL> exit
Disconnected.
```

참고로, Tibero 설치 시 기본적으로 제공하는 유저는 다음과 같다.

[Tibero default user]

- sys : DBMS의 데이터딕셔너리소유자, 데이터베이스 관리자(super user)
DB생성 가능
Default password - tibero
- syscat : 모든 권한이 SYS와 같으나 DB생성 권한은 없음.
Default password - syscat
- tibero : sample 사용자 계정
Default password - tmax
- tibero1 : sample 사용자 계정
Default password - tmax

2.3. 간단한 SQL Test

tbsql SYS계정으로 접속 후, 수행할 수 있는 기본 작업을 테스트한다.

[TEST 내용]

- 접속한 계정 스키마의 모든 객체 보기
- datafile경로 확인
- tablespace 생성/생성정보확인/삭제
- user생성/Database접속 및 리소스 사용권한 생성 및 삭제
- 자주 사용하는 간단한 SQL명령

2.3.1. 접속한 계정 스키마에 속한 모든 객체 보기

tbsql SYS계정으로 서버에 접속하여 SYS소유의 모든 객체를 ls명령어를 통해 확인한다. SYS 사용자는 Tibero가 스스로를 관리하기 위해 내부적으로 사용하는 수많은 객체를 가지고 있으므로, 다음처럼 매우 긴 목록이 나오게 된다.

```
SQL> ls

NAME                                SUBNAME                                OBJECT_TYPE
-----
_GET_PART_BOUND_STR                 FUNCTION
HS$_BASE_CAPS_PK                    INDEX
HS$_BASE_DD_PK                      INDEX
... 중간 생략 ...
V$WAITUSE_STAT                      VIEW
V$WAITUSE_THR_STAT                  VIEW
X$USER_ROLE                          VIEW

SQL>
```

ls명령어 뒤에는 옵션을 넣어 수행시킬 수도 있다. 옵션을 통해 user가 원하는 객체만을 선택적으로 확인해 볼 수 있다. 옵션으로 위치할 수 있는 내용은 다음과 같다.

```
LS [{FUNCTION|INDEX|PACKAGE|PROCEDURE|SEQUENCE|
    SYNONYM|TABLE|TRIGGER|TS|USER|VIEW}]
```

단, "LS"명령은 tbSQL이 사용자 편의를 위해 제공하는 명령어이며, Tibero SQL문장에서 지원하는 명령은 아니다. 즉, tbSQL이 아닌 JDBC, CLI등을 이용하여 접속하였을 경우 이 명령을 사용할 수 없다.

2.3.2. datafile경로 확인

Datafile의 경로 확인은 DBA_DATA_FILES뷰를 통해 확인할 수 있다.

```
SQL> SELECT file_name FROM DBA_DATA_FILES;
```

```
FILE_NAME
```

```
-----  
/home/tibero/tbdata/system001.tdf  
/home/tibero/tbdata/undo001.tdf  
/home/tibero/tbdata/usr001.tdf
```

```
3 rows selected.
```

2.3.3. tablespace 생성/생성정보확인/삭제

1) tablespace생성

```
SQL> CREATE TABLESPACE test_tb  
DATAFILE '/home/tibero/tbdata/test_tb.tdf' SIZE 10M  
AUTOEXTEND ON NEXT 1M MAXSIZE 512M  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE;  
Tablespace created.
```

2) 생성된 tablespace의 정보 확인

```
SQL> SELECT A.FILE_NAME, A.TABLESPACE_NAME, BYTES/1024/1024||'M' "SIZE"  
FROM DBA_DATA_FILES A, DBA_TABLESPACES B  
WHERE A.FILE_ID = B.TS_ID  
AND A.TABLESPACE_NAME = 'TEST_TB';
```

```
FILE_NAME                TABLESPACE_NAME  SIZE  
-----  
/home/tibero/tbdata/test_tb.tdf  TEST_TB          10M
```

```
1 row selected.
```

3) tablespace삭제

```
DROP TABLESPACE test_tb INCLUDING CONTENTS AND DATAFILES;
```


2.3.4. user 생성/Database접속 및 리소스 사용권한 생성 및 삭제

1) user생성

```
SQL> CREATE USER test IDENTIFIED BY test;  
User 'TEST' created.
```

2) Database접속 및 리소스 사용권한 생성

```
SQL> GRANT RESOURCE , CONNECT TO test;  
Granted.
```

3) user삭제

```
SQL> DROP USER test CASCADE;  
User 'TEST' dropped.
```

2.3.5. 자주 사용되는 간단한 SQL명령

1) set 환경 설정

현재 tbSQL 에 대한 환경설정값은 show all 명령어를 통해 볼 수 있다.

```
SQL> show all

SYSTEM PARAMETERS
-----
autocommit OFF
autotrace OFF
blockterminator "."
comment ON
ddlstats OFF
define ON
editfile ".tbedit.sql"
escape "₩"
feedback ON
fileext "sql"
filepath ""
history 50
linesize 80
long 80
numwidth 10
outputsize 1024
pagesize 24
profile OFF
prompt "SQL> "
sqlterminator ";"
serveroutput OFF
time OFF
timing OFF
timeout ON
trimsPOOL OFF
```

셋팅된 값을 변경하기 위해서는 다음과 같이 실행시킨다.

- ▶ 한 페이지당 표시되는 라인 수(기본값 24)

```
SQL> SET PAGESIZE 100
```

- ▶ 한 줄에 표시될 텍스트숫자(기본값 80)

```
SQL> SET LINESIZE 150
```

- ▶ sql쿼리 실행 시간을 확인(기본값 OFF)

```
SQL> SET TIMING ON
```

- ▶ 현재 시간을 표시(기본값 OFF)

```
SQL> SET TIME ON
```

- ▶ 수행중인 쿼리의 실행 계획이나 통계 정보 출력(기본값 OFF)

```
SQL> SET AUTOTRACE ON
```

- ▶ tbPSM 수행결과를 화면에 출력하기 위한 명령어 (기본값 OFF)

```
SQL> SET SERVEROUTPUT ON
```

- 2) SQL파일 실행

```
SQL> start 파일명(or @파일명)
```

- 3) 버퍼에 저장되어 있는 내용 편집

```
SQL> ed
```

- 4) OS명령어 실행

```
SQL> !OS명령어(or host OS명령어)
```

- 5) 이전 명령 실행

```
SQL> /(or run)
```

- 6) 화면 내용 저장(spool off까지의 내용저장)

```
SQL> spool 파일명  
...  
SQL> spool off
```

- 7) 명령어 도움말 보기

```
SQL> help 명령
```

- 8) 표시되는 컬럼 사이즈 조절

```
SQL> column 컬럼명 format a20
```

- 9) 숫자 출력포맷 조절

```
SQL> column 컬럼명 format 999,999
```

3. tbMigrator

tbMigrator는 다른 데이터베이스에 저장된 테이블, 인덱스 뷰 등의 스키마 객체와 내장 SQL(Embedded SQL), PSM 프로그램 등을 Tibero 데이터베이스로 옮겨 이전의 데이터베이스에서와 같은 기능을 수행하도록 한다.

[tbMigrator에서 제공하는 기능]

- 타 DBMS의 데이터 및 응용 프로그램을 Tibero로 마이그레이션
- 테이블, 인덱스, 뷰, 동의어 등의 데이터베이스 오브젝트와 테이블에 정의된 각종 제약조건을 마이그레이션
- 사용자 특권(privilege) 및 역할(role)을 마이그레이션
- 마이그레이션 대상 데이터베이스에 대한 정보 제공

*** 자세한 사항은 Tibero RDBMS Utility_tbMigrator 기술백서 참조

3.1. tbMigrator 설치 및 실행

3.1.1. 설치

별도의 라이선스 없이 \$TB_HOME/client/bin/tbmigrator.zip 파일의 압축을 풀면 바로 설치됨

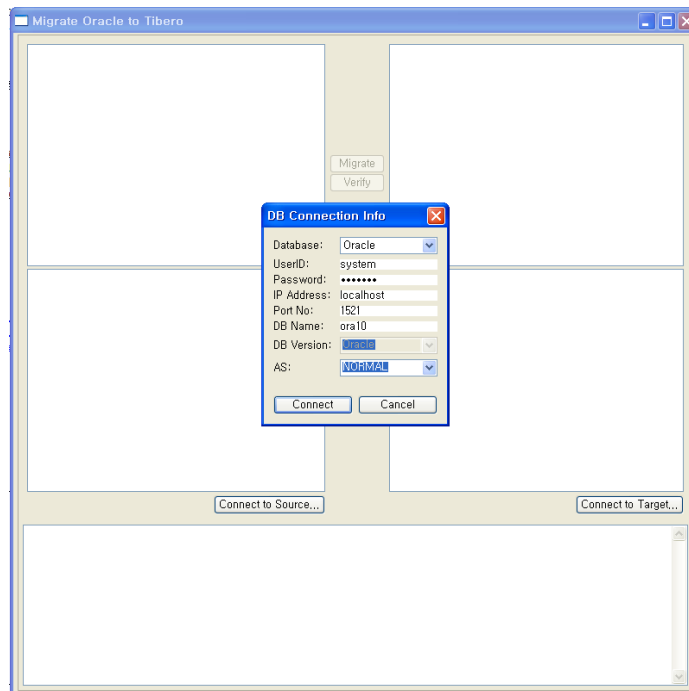
[tbMigrator 설치에 필요한 클래스 라이브러리]

- tbMigrator 클래스 파일: tbmigrator.jar
- Tibero 유틸리티 공통 라이브러리: tbtoolcom.jar
- Tibero JDBC 드라이버: tibero-jdbc.jar
- 마이그레이션 대상 데이터베이스의 JDBC

***마이그레이션 작업시 tbmigrator\홈\plugins\wtbmigrator_XXX\libs\₩밑에 필요한 .jar 파일을 가져다 놓아야 함.

3.1.2. 실행

- 1) 실행하려는 해당 OS에 맞는 디렉토리로 이동 후 tbMigrator 실행파일을 실행
- 2) tbMigrator는 JAVA로 구현되어 있으며, JVM(java Virtual Machine)이 설치되어 있는 어떤 플랫폼에서도 바로 운용 가능



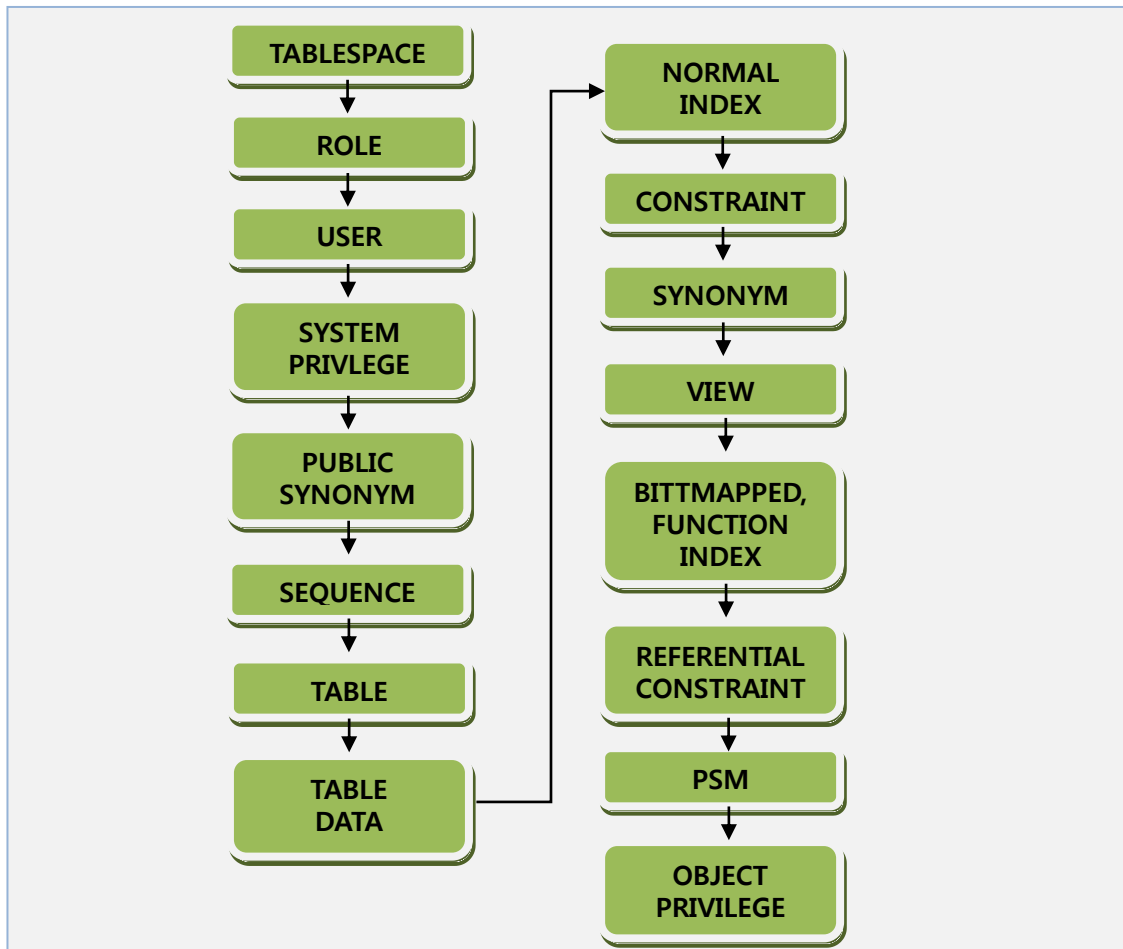
Source DB 접속정보와 Target접속정보를 입력하여 마이그레이션 작업할 스키마를 선택한 후 Migrate버튼을 누르면 작업이 실행된다.

***실행화면에 대한 자세한 사항은 Tibero RDBMS Utility_tbMigrator 기술백서를 참조한다.

3.1.3. tbMigrator 옵션

- 1) Execute DDL : 타 DBMS에서 추출한 DDL문을 Tibero에서 실행함.
- 2) Migrate Table Rows : Table 데이터 마이그레이션
- 3) Migrate Option
 - Multi Thread Count : 테이블 데이터를 병렬적으로 마이그레이션 하기 위해 사용하는 스레드의 개수
 - Table Rows : 지정한 로우 수 만큼만 마이그레이션 하기 위해 사용
 - Table Name Condition : 특정한 조건의 테이블을 마이그레이션 하기 위하여 사용하는 선별조건으로 SQL 질의조건의 형태로 지정
 - Insert Table Data by One Row : row by row로 single row insert 기능
 - Use Direct Path Load : 버퍼캐시를 통하지 않고 데이터 파일에 직접 기록
- 4) Data Type
 - Read as Byte : 마이그레이션 할 때 byte 단위로 읽어서 처리
 - LONG to CLOB : 오라클 LONG 타입을 테베로 CLOB 타입으로 마이그레이션
 - LONGRAW to BLOB : 오라클 LONG RAW 타입을 테베로 BLOB 타입으로 마이그레이션.

3.1.4. 수행순서



3.1.5. 출력로그

1) DDL 로그

- 마이그레이션 진행 중에 수행하는 모든 DDL 문장을 로깅한다.
- 스크립트 파일은 sql의 형태로 tbSQL에서 그대로 사용할 수 있다.
- 로그 파일의 형식 : mig_script_yy_MM_dd_HHmms.sql

```
CREATE TABLESPACE "USERS" DATAFILE
'users01.dbf' SIZE 6291456 AUTOEXTEND ON NEXT 1310720 MAXSIZE
34359721984
/
CREATE TABLESPACE USERS DATAFILE
'users01.dbf' SIZE 6291456 AUTOEXTEND ON NEXT 1310720 MAXSIZE
34359721984
/
CREATE ROLE CONNECT IDENTIFIED BY password
/
```

2) 진행상황 로그

- 마이그레이션을 진행하면서 발생하는 모든 정보를 기록한다.
- 기록하는 내용
 - 시작 / 종료 시간정보
 - 오브젝트 생성 정보
 - (오브젝트 생성이 실패하게 되면 *표시와 함께 에러 메시지를 표시한다.)
 - 테이블 데이터 마이그레이션 정보
- 마이그레이션 전후의 테이블 로우의 개수를 비교하여 해당 테이블의 마이그레이션 성공여부를 표시한다.
- 로그 파일의 형식 migration_ yy_MM_dd_HHmms.log

```
TiBero Version: 3.x - Tmax Data, Co. Copyright(C) 2001-2007.
About to migrate: Thu Jul 12 15:33:25 KST 2007
----- Migrate TABLESPACE -----
TABLESPACE."TS_TBS" is created successfully.
[TABLESPACE(S) GENERATED]It took 64515[ms]
----- Migrate ROLE -----
CREATE ROLE CONNECT IDENTIFIED BY password
CREATE ROLE RESOURCE IDENTIFIED BY password
.....
```

4. tbExport

Tibero 데이터베이스에 저장된 스키마 객체 전체 또는 일부를 SQL문장 형태로 추출하여 tbExport에서 정의한 고유형식을 갖는 파일에 저장한다.(논리적인 백업)

[tbExport 모드종류]

- 전체 데이터베이스모드[FULL]
- 사용자 모드[USER]
- 테이블 모드[TABLE]

[tbExport 파라미터 확인]

tbExport를 실행하기 위한 기본적인 명령어 라인의 파라미터에 대한 정보

```
[tibero:/home/tibero]$tbexport
tbexport
tbExport 4.0 - TmaxSoft, Co. Copyright(C) 2001-2009.
Usage: tbexport [PARAMETER1=VALUE] [PARAMETER2=VALUE] ...
Parameters:
  CFGFILE           Config file name
  USERNAME          Database user name
  PASSWORD          User password
  IP                IP address, default: localhost
  PORT              PORT number, default: 8629
  FILE              Export dump file name, default: default.dat
  NO_PACK_DIR       Export unpacked dump files to specified directory. If this option is specified, FILE parameter will be ignored.
  OVERWRITE         Overwrite datafile if same file name exists: Y/N, default: N
  LOG               Export log file name
  FULL              Full Mode: Y/N, default: N
  USER              User Mode: user name list
  TABLE            Table Mode: table name list
  QUERY             Where predicate: (Optional) to filter data to be exported(must be used with TABLE parameter)
  INDEX             Export Index: Y/N, default: Y
  GRANT             Export Grant: Y/N, default: Y
  CONSTRAINT        Export Constraint: Y/N, default: Y
  ROWS              Export Table Rows: Y/N, default: Y
  SCRIPT            LOG THE DDL SCRIPT: Y/N, default: N
  THREAD_CNT        Thread Count, default: 4
  SERVER_VER        Tibero Server Version, default: 4
  PARALLEL_DEGREE  Option of parallel hint, default: 0 (NOT PARALLEL)
```


4.1. 전체 데이터베이스 모드로 tbExport

SYS사용자를 제외한 모든 사용자의 오브젝트를 추출한다. DBA권한을 가진 사용자만 가능하다.

[Syntax]

```
tbexport USERNAME=sys PASSWORD=tibero IP=localhost PORT=8629 FILE=exp_data_full.dat  
LOG=exp_data_full.log FULL=y SCRIPT=y
```

위 tbExport에서 사용한 파라미터는 ip가 localhost이고 port가 8629인 Tibero server에 SYS계정으로 접속하여 SYS사용자를 제외한 모든 사용자의 오브젝트를 추출하여 exp_data_full.dat파일에 저장하고, 이에 대한 각 오브젝트에 대한 DDL문을 script로 log파일에 저장하겠다는 의미이다.

수행시키면 아래와 같이 export작업이 수행된다.

```
tbexport  
tbExport 4.0 - TmaxSoft, Co. Copyright(C) 2001-2009.  
the entire database: Wed Oct 07 13:43:21 KST 2009  
Export character set: MSWIN949  
exporting tablespaces  
...중간 생략...  
exporting schema: "TIBERO"  
exporting constraints  
exporting schema: "TIBERO1"  
exporting constraints  
exporting referential constraints  
Packing the file...  
Export completed successfully: Wed Oct 07 13:43:26 KST 2009
```

Export된 파일을 확인해 보면 다음과 같이 생성되었다.

```
[tibero:/home/tibero]$ls -lrt exp_data_full.*  
-rw-r--r-- 1 tibero4 tmax 45443 10월 7 13:43 exp_data_full.log  
-rw-r--r-- 1 tibero4 tmax 163840 10월 7 13:43 exp_data_full.dat
```

Export받은 exp_data_full.dat파일은 이진파일이며, exp_data_full.log파일은 텍스트 파일로 export log가 기록되어 있다. Export가 정상적으로 수행되었는지 확인하려면 log파일을 분석하면 된다.

4.2. 사용자 모드로 tbExport

Export대상 사용자를 지정하고 지정한 사용자가 소유한 모든 스키마 정보를 추출한다. DBA는 하나 이상의 사용자를 지정할 수 있다.

[Syntax]

```
tbexport USERNAME=sys PASSWORD=tibero IP=localhost PORT=8629 FILE=exp_data_user.dat  
LOG=exp_data_user.log USER=tibero SCRIPT=y
```

수행시키면 아래와 같이 Export작업이 수행된다.

```
tbexport  
tbExport 4.0 - TmaxSoft, Co. Copyright(C) 2001-2009.  
the USER: tibero: Wed Oct 07 13:48:53 KST 2009  
Export character set: MSWIN949  
  exporting schema: "TIBERO"  
  exporting constraints  
  exporting referential constraints  
Packing the file...  
Export completed successfully: Wed Oct 07 13:48:55 KST 2009  
[tibero:/home/tibero]$
```

Export된 파일을 확인해 보면 다음과 같이 생성되었다.

```
[tibero:/home/tibero]$ls -lrt exp_data_user.*  
-rw-r--r-- 1 tibero4 tmax 6063 10월 7 13:48 exp_data_user.log  
-rw-r--r-- 1 tibero4 tmax 40960 10월 7 13:48 exp_data_user.dat
```

4.3. 테이블 모드로 tbExport

하나 이상의 지정한 테이블과 연관된 인덱스 등의 데이터베이스 오브젝트를 함께 추출한다. 테이블지정 시 소유자도 명시해야 한다.

[Syntax]

```
tbexport USERNAME=sys PASSWORD=tibero IP=localhost PORT=8629 FILE=exp_data_table.dat
LOG=exp_data_table.log TABLE=tibero.employee SCRIPT=y
```

수행시키면 아래와 같이 export작업이 수행된다.

```
tbexport
tbExport 4.0 - TmaxSoft, Co. Copyright(C) 2001-2009.
the TABLE: tibero.employee: Wed Oct 07 13:52:10 KST 2009
Export character set: MSWIN949
  exporting table: "TIBERO"."EMPLOYEE"
    [0] EMPLOYEE                                20 rows exported.
CREATE TABLE "EMPLOYEE" (
  "EMP_NO" VARCHAR(8),
  "EMP_NAME" VARCHAR(20),
  "HIREDATE" DATE,
  "SALARY" NUMBER(8,3),
  "BONUS" NUMBER(8,3),
  "DEPT_CD" VARCHAR(4),
  "MANAGER" VARCHAR(8)
)
TABLESPACE "USR"
PCTFREE 10
INITRANS 2
/
  exporting indexes
CREATE UNIQUE INDEX "EMPLOYEE_PK" ON "EMPLOYEE" (
  "EMP_NO" ASC
)
LOGGING
TABLESPACE "USR"
PCTFREE 20
INITRANS 2
/
  exporting constraints
ALTER TABLE "EMPLOYEE" ADD CONSTRAINT "EMPLOYEE_PK"
PRIMARY KEY (
  "EMP_NO"
)
/
ALTER TABLE "EMPLOYEE" MODIFY  "EMP_NO" NOT NULL
/
Packing the file...
Export completed successfully: Wed Oct 07 13:52:12 KST 2009
```

Export된 파일을 확인해 보면 다음과 같이 생성되었다.

```
[tibero:/home/tibero]$ls -lrt exp_data_table.*
-rw-r--r-- 1 tibero4 tmax 1259 10월 7 13:52 exp_data_table.log
-rw-r--r-- 1 tibero4 tmax 20480 10월 7 13:52 exp_data_table.dat
```

5. tbImport

tbExport에 의하여 생성된 export file로부터 스키마 객체를 Tibero데이터베이스에 저장하는 유틸리티이다. 예제는 앞서 설명한 tbExport에서 생성한 export file을 이용하여 import해보도록 한다.

[tbImport 모드종류]

- 전체 데이터베이스모드[FULL]
- 사용자 모드[USER]
- 테이블 모드[TABLE]

[tbImport 파라미터 확인]

tbImport를 실행하기위한 기본적인 명령어 라인의 파라미터에 대한 정보

```

tbimport
tbImport 4.0 - TmaxSoft, Co. Copyright(C) 2001-2009.
Usage: tbimport [PARAMETER1=VALUE] [PARAMETER2=VALUE] ...
Parameters:
  CFGFILE          Config file name
  USERNAME         Database user name
  PASSWORD         User password
  IP               IP address, default: localhost
  PORT             PORT number, default: 8629
  FILE             Import dump file name, default: default.dat
  NO_PACK_DIR     Import unpacked dump files from specified directory. If this option is specified, FILE parameter will be ignored.
  LOG              Import log file name
  FULL             Full Mode: Y/N, default: N
  USER            User Mode: user name list
  FROMUSER        FromUser toUser Mode: user name list(must be used with TOUSER parameter)
  TOUSER          FromUser toUser Mode: user name list(must be used with FROMUSER parameter)
  TABLE          Table Mode: table name list
  INDEX           Import Index: Y/N, default: Y
  GRANT           Import Grant: Y/N, default: Y
  CONSTRAINT      Import Constraint: Y/N, default: Y
  ROWS            Import Table Rows: Y/N, default: Y
  DPL             Use Direct Path Load: Y/N, default: N
  PIPELINING      Use pipelined import(must be used with DPL parameter): Y/N, default: N
  SCRIPT          LOG THE DDL SCRIPT: Y/N, default: N
  THREAD_CNT      Thread Count, default: 4
  SERVER_VER      Tibero Server Version, default: 4
  IGNORE          Ignore create error due to object existence: Y/N, default: N
  IO_BUF_SIZE     Specify the buffer size of file I/O, default: 16M(16777216)
  BIND_BUF_SIZE   Specify the buffer size of DPL stream, default: 1M(1048576)
    
```

5.1. 전체 데이터베이스 모드로 tbImport

전체 데이터베이스 모드는 export한 파일로부터 Tibero데이터베이스 전체를 import하는 모드로, DBA만이 사용할 수 있는 모드이며 SYS 사용자를 제외한 모든 사용자의 객체를 import하기 위해 사용한다.

[Syntax]

```
tbimport USERNAME=sys PASSWORD=tibero IP=localhost PORT=8629 FILE=exp_data_full.dat  
LOG=imp_data_full.log FULL=y SCRIPT=y
```

5.2. 사용자 모드로 tbImport

사용자 모드는 export한 파일로부터 지정한 사용자에게 대하여 그 사용자가 소유하고있는 모든 스키마 객체를 import하는 모드로, DBA는 하나 이상의 사용자에게 대하여 이 모드를 사용 가능하다.

[Syntax]

1) export 한 user에서 동일 user로 import할 때

```
tbimport USERNAME=sys PASSWORD=tibero IP=localhost PORT=8629 FILE=exp_data_user.dat  
LOG=imp_data_user.log USER=tibero SCRIPT=y
```

2) export 한 user가 아닌 다른 user에게로 import할 때

```
tbimport USERNAME=sys PASSWORD=tibero IP=localhost PORT=8629 FILE=exp_data_user.dat  
LOG=imp_data_user.log FROMUSER=tibero TOUSER=tibero1 SCRIPT=y
```

5.3. 테이블 모드로 tbImport

테이블 모드는 export한 파일로부터 하나 이상의 테이블을 지정하여 그 테이블과 연관된 인덱스 등의 스키마 객체들을 함께 import하는 모드이다. 테이블 모드를 사용하기 위해서는 TABLE= tablelist 형태로 파라미터를 설정하며, 주의해야 할 점은 테이블을 소유한 사용자를 반드시 명시해야 한다. (예를 들어, TIBERO.EMPLOYEE)
만약 FROMUSER/TOUSER를 함께 사용할 경우에는 테이블 소유자를 명시하지 않고 작성한다.

[Syntax]

1) export 한 user에서 동일 user로 import할 때

```
tbimport USERNAME=sys PASSWORD=tibero IP=localhost PORT=8629 FILE=exp_data_table.dat  
LOG=imp_data_table.log TABLE=tibero.employee SCRIPT=y
```

2) export 한 user가 아닌 다른 user에게로 import할 때

```
tbimport USERNAME=sys PASSWORD=tibero IP=localhost PORT=8629 FILE=exp_data_table.dat  
LOG=imp_data_table.log FROMUSER=tibero TOUSER=tibero1 TABLE=employee SCRIPT=y
```

6. tbLoader

컬럼데이터만 일반 텍스트 파일로 만들어서 한꺼번에 적재하는 유틸리티이다.
(insert SQL 문을 일일이 만들어 작성할 필요가 없다.)

데이터를 로드하는 방법은 다음과 같다.

[데이터 로드 방법]

- Conventional Path Load

사용자가 지정한 데이터 파일을 읽어가며, 컬럼 데이터를 읽어 한 개의 레코드마다 직접 insert문을 통하여 데이터를 데이터베이스 서버에 로드시킴.

- Direct Path Load

지정한 데이터 파일을 읽어가며, 특정 컬럼의 데이터 타입에 맞게 데이터를 컬럼 배열 (column array)형태로 만든다. 컬럼 배열 형태의 데이터는 block formatter을 거쳐서 Tibero데이터베이스 block형태에 맞게 만들어지고, 이 블록을 직접 Tibero 데이터베이스에 쓰게 된다.

6.1. tbLoader

tbLoader 유틸리티 사용시 필요한 입출력파일 작성방법과 파라미터에 대해 설명한다.

[필요한 입출력파일]

- 컨트롤파일[Control File] -입력파일
- 데이터파일[Data File] -입력파일
- 로그파일[Log File] -출력파일
- 오류파일[BadFile] -출력파일

6.1.1. 컨트롤 파일(Control File)

tbLoader의 실행을 위한 파라미터를 지정한 파일이다.

[컨트롤 파일 형식과 파라미터 설명]

```
LOAD DATA
[CHARACTERSET charset_name]                --문자셋 지정
[INFILE data_file_name]                    --데이터 파일 지정
[LOGFILE log_file_name]                    --로그 파일 지정
[BADFILE bad_file_name]                    --오류 파일 지정
[APPEND|REPLACE|TRUNCATE]                 --테이블에 존재하는 기존 데이터 처리방법을 기술
INTO TABLE table_name                     --테이블 지정
[FIELDS [TERMINATED BY field_terminator]   --필드 종료자 지정
[OPTIONALLY ENCLOSED BY enclosed_by_start_string --필드의 값의 시작과 끝을 감쌀 문자열을 지정
[AND enclosed_by_end_string]]             --필드의 값의 시작과 끝을 감쌀 문자열을 지정
[ESCAPED BY escaped_by_string]             --특수문자 또는 문자열을 읽을 수 있도록 지정
[LINES [STARTED BY line_start_string]      --라인 시작 문자열 지정
[TERMINATED BY line_terminator_string]   --라인 종료 문자열 지정
]
[IGNORE number LINES]                      --데이터 파일에서 무시할 라인의 개수를 지정
(column_name [POSITION(from.to)] [OUTFILE] [CONSTANT constant_value], --컬럼 상수값 지정
.....)]
```

단, 컨트롤 파일에서 정한 위의 형식의 순서대로 사용해야 하고, '['와 ']'로 감싸진 부분은 생략가능한 선택적인 부분이다.

6.1.2. 데이터 파일(Data File)

데이터베이스의 테이블에 저장할 데이터가 들어있는 텍스트 파일이며, 다음 두 가지 형식이 있다.

1) 고정된 레코드 형태(Fixed Record Format)

: 컨트롤파일에서 모든 컬럼에 대해 Position정보를 명시하였을 경우 적용되는 형식

2) 분리된 레코드 형태(Separated Record Format)

: 컨트롤파일에서 모든 컬럼에 대해 Position정보를 명시하지 않았을 경우 적용되는 형식

6.1.3. tbLoader실행

[Syntax]

```
tbloader userid=계정@DB명 control=컨트롤 파일 위치
```

[tbLoader 파라미터 확인]

```
[tibero:/home/tibero]$tbloader
tbLoader Version: 3.0 - TmaxSoft, Co. Copyright(C) 2001-2009.
```

Usage: tbloader [OPTIONS]

OPTIONS:

```
userid      tibero userid/passwd@dbname
control     control file name
log         log file name
bad         bad file name
data        data file name
skip        skip lines in data file[default:0]
direct      direct path load[default:N]
dpl_log     enable direct path load logging[default:N]
message     loading progress message to stdout
readsize    read buffer size
bindsize    bind buffer size
errors      errors to allow(default:50)
rows        rows per commit
multithread use multithread for dpl(default:Y)
```

```
Example: tbloader userid=tibero/tmax@dbname control=sample.ctl bindsize=1000000
```

6.1.4. 로그파일과 오류파일 확인

tbLoader실행 후 생성되는 출력파일인 로그파일(.log)과 오류 파일(.bad)파일을 확인한다.

1) 로그파일 : tbLoader의 실행 과정을 기록한 파일

2) 오류파일 : tbLoader의 실행 시 로드에 실패한 레코드의 데이터를 기록한 파일

6.2. tbLoader TEST

[테스트 시나리오]

- 1) 테이블을 생성한다.
- 2) 컨트롤 파일을 작성한다.
- 3) 데이터 파일을 작성한다.
- 4) 명령어를 실행한다.
- 5) 실행결과 로그파일과 오류파일을 체크한다.

* 테이블 생성 - 공통

해당 데이터베이스명은 tibero이며, 테이블의 소유자는 tibero/tmax라고 가정한다.

```
CREATE TABLE MEMBER (  
  ID          NUMBER(4) NOT NULL,  
  NAME       VARCHAR(15),  
  JOB        VARCHAR(9),  
  BIRTHDATE  DATE,  
  CITY       VARCHAR(10),  
  PICTURE    BLOB,  
  AGE        NUMBER(3),  
  RESUME     CLOB);  
  
CREATE TABLE CLUB(  
  ID          NUMBER(6) NOT NULL,  
  NAME       VARCHAR(10),  
  MASTERID   NUMBER(4));
```

6.2.1. 분리된 레코드 데이터 파일 형태일 경우

1) 컨트롤 파일을 작성한다.(sample1.ctl)

```
LOAD DATA
INFILE './sample1.dat'
APPEND
INTO TABLE club
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 LINES
(id,name, masterid)
```

2) 데이터 파일을 작성한다.(sample1.dat)

```
id name masterid|
111111, FC-SNIFER, 2345|
Dkkkkkkkkk|
11112, "DOCTOR CLUBE ZZANG", 2222|
11113, "ARTLOVE", 3333|
11114, FINANCE, 1235|
11115, "DANCE MANIA", 2456|
```

3) 실행

```
$ tloader userid=tibero/tmax@tibero control=./sample1.ctl
```

4) 로그파일 및 오류 파일 확인

[로그파일] sample1.log

```
[tibero:/home/tibero]$cat sample1.log
tbLoader : Release 4.0
TmaxSoft, Co. Copyright(C) 2001-2009.

Data File : ./sample1.dat
Bad File : ./sample1.bad

Table CLUB was loaded from Data File.

      Column Name      Position      DataType
-----
ID                      1 NUMERIC EXTERNAL
NAME                    2 CHARACTER
MASTERID                3 NUMERIC EXTERNAL

Record 2 - rejected.
Cause - TBR-80053 : Some relatively positioned columns are not present in the record.

Record 3 - rejected. Table CLUB, Error in column NAME
Cause - TBR-80025 : Column data exceeds data buffer size.

Record 6 - rejected. Table CLUB, Error in column NAME
Cause - TBR-80025 : Column data exceeds data buffer size.

Table CLUB:
3 Rows successfully loaded.
3 Rows not loaded due to data errors

Elapsed time was: 00:00:00.086783
```

[오류파일] sample1.bad

```
[tibero:/home/tibero]$cat sample1.bad
Dkkkkkkkk
11112, "DOCTOR CLUBE ZZANG", 2222|
11115, "DANCE MANIA", 2456|
```

6.2.2 고정된 레코드 데이터 파일 형태일 경우

1) 컨트롤 파일을 작성한다.(sample2.ctl)

```
LOAD DATA
INFILE './sample2.dat'
APPEND
INTO TABLE MEMBER
(id position (01:04),
name position (06:12),
job position(13:20),
birthdate position(21:31),
city position(32:36),
age position(37:38))
```

2) 데이터 파일을 작성한다.(sample2.dat)

```
7777 KKS CHAIRMAN 1975-11-18 SEOUL 33
7839 BSH MANAGER 1982-11-18 BUSAN 28
7934 MILLER CLERK 1967-01-24 BUSAN 37
7566 JONES MANAGER
7499 ALLEN SALESMAN dddddddd KYUNG-JU
aaa7654 MARTIN SALESMAN
```

3) 실행

```
$ tloader userid=tibero/tmax@tibero control=./sample2.ctl
```

4) 로그파일 및 오류 파일 확인

tbLoader실행 후 생성된 sample2.log파일과 sample2.bad파일을 직접 확인해보도록 한다.

7. tbPROF

SQL 실행 정보를 profiling하기 위한 SQL trace 파일은 정적 혹은 동적으로 SQL_TRACE 파라미터를 적용하여 생성할 수 있다.

SQL_TRACE = Y|N (Default : N)

[tbPROF 파라미터 확인]

\$tbprof

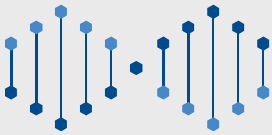
Usage: tbprof tracefile outputfile [print=] [sort=] [aggregate=]

print=integer List only the first 'integer' SQL statements.
 sys=yes|no Filter SQL statements that 'SYS' user executes.
 aggregate=yes|no Aggregate statistics of same SQL statements.
 sort=option Set of zero or more of the following sort options:

- prscnt number of times parse was called
- prscpu cpu time parsing
- prsela elapsed time parsing
- prsdsk number of disk reads during parse
- prsqry number of buffers for consistent read during parse
- prscu number of buffers for current read during parse
- execnt number of execute was called
- execpu cpu time spent executing
- exeela elapsed time executing
- exedsk number of disk reads during execute
- exeqry number of buffers for consistent read during execute
- execu number of buffers for current read during execute
- exerow number of rows processed during execute
- fchcnt number of times fetch was called
- fchcpu cpu time spent fetching
- fchela elapsed time fetching
- fchdsk number of disk reads during fetch
- fchqry number of buffers for consistent read during fetch
- fchcu number of buffers for current read during fetch
- fchrow number of rows fetched
- userid userid of user that parsed the cursor

- tracefile : SQL Trace의해 생성된 통계정보를 가진 파일명
- outputfile : tbPROF를 통해 읽기 가능한 텍스트 파일로 생성할 파일명
- print : 지정된 수의 SQL문에 대해서만 TRACE결과를 PRINT함. (Default : ALL)
- sys : SYS유저가 내부적인 작업을 위해 실행한 SQL문을 출력 시 포함여부(Default : yes)
- aggregate : 같은 SQL에 대한 정보를 합산할지 여부를 지정하는 파라미터 (Default : no)
- sort : 결과정렬방식을 지정하는 파라미터. (SORT_OPTION에 따라 결과들을 내림차순으로 정렬함. 여러 개 동시 지정가능하며 필드는 ','로 구분하고 공백이 있으면 안됨.)

***사용법에 관한 자세한 사항은 Tibero Utility_tbPROF기술백서 참조할 것.



Information Intelligence, **Tibero**