

—
인터페이스/연동
—

Tibero

DB Link 가이드

TMAXTibero

Copyright © 2025 TmaxTibero. All Rights Reserved

Copyright Notice

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

대한민국 경기도 성남시 분당구 정자일로 45, 티맥스소프트타워 (13613)

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 TIBERO 의 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 DB Link 가이드

발행일: 2025-12-4

소프트웨어 버전: Tibero7 FS02

안내서 버전: 1.1

제, 개정 이력

안내서 버전	개정일자	개정 사유 및 내용	비고
1.0	2024.02.14	최초 제정	작성자: 민정기, 유하은
1.1	2025.12.04	문서 검토 및 수정	확인자: 박기환

목록

제1장 DB Link 소개.....	4
1.1. 개요	4
1.2. DB Link 장, 단점	4
제2장 Tibero to Tibero	5
2.1. Tibero to Tibero DB Link	5
2.2. DB Link 생성 및 사용	6
제3장 Tibero to Oracle	9
3.1. 개요	9
3.2. Gateway for Oracle 설정	11
3.3. Oracle 클라이언트	25
3.4. 문제 해결	28
3.5. Tibero7 의 gw_install.sh 수행	30
제 4장 Oracle to Tibero	32
4.1. 개요	32
4.2. DB Link 설정	34
4.3. ODBC Manager 설치	46
4.4. 문제 해결	52

제1장 DB Link 소개

본 장에서는 DB Link의 기본적인 개념과 장,단점에 대해서 설명한다.

1.1. 개요

DB Link는 현재의 데이터베이스에서 네트워크상의 다른 데이터베이스에 접속하기 위해 접속 설정을 정의하는 객체이다.

1.2. DB Link 장, 단점

- 장점
 - 분산된 데이터를 다루기 편리하다.
 - 데이터베이스에 한번만 접속해도 Link를 통해 다른 데이터베이스에 간편히 접속할 수 있다.
- 단점
 - Link를 통해 접속된 데이터베이스의 세션과 Lock이 증가할 수 있으므로 실시간 트랜잭션이 많은 시스템에서 장애가 발생할 수 있다.
 - 배치작업과 같은 큰 트랜잭션을 DB Link를 통해 직접 작업할 경우 장애 발생 가능성이 증가한다.
 - DB 운영자의 경우 데이터베이스간 DB Link에 대해 사용현황 관리가 필요하다.
 - 장애가 발생할 경우 개발, 시스템 운영, 네트워크, DB 업무 담당자간 확인이 필요하다.
 - Long Type, CLOB, BLOB 타입을 Link를 통해 DML 작업할 경우 오류가 발생할 가능성이 있다.

참고

Link 사용을 지양하며, AP에서 각각 별도로 처리하는 것을 권장한다.

제2장 Tiberio to Tiberio

본 장에서는 Tiberio와 Tiberio 간의 DB Link 생성 방법을 설명한다.

2.1. Tiberio to Tiberio DB Link

2.1.1. 제약사항

다음은 Tiberio 간의 호환성에 대한 설명이다.

- Source에 해당하는 Tiberio에서 Target에 해당하는 Tiberio에 연결할 수 없다면 Link 구성을 할 수 없다.

2.1.2. Tiberio 클라이언트 설정

접속하려는 Tiberio(Tiberio_A라고 함)의 IP 주소, 포트 번호, DB NAME 내용을 확인하고, DB Link를 생성할 Tiberio(Tiberio_B라고 함)의 네트워크 설정 파일(tbdsn.tbr)에 설정한다.

Tiberio_A는 설정이 필요없고, Tiberio_B가 설치되어 있는 서버의 tbdsn.tbr 파일을 아래와 같이 설정한다.

```
# Tiberio_B 정보
tiberio=(
  (INSTANCE=(HOST=localhost)
    (PORT=5555)
    (DB_NAME=tiberio)
  )
)

# 접속할 Tiberio_A의 정보 추가 설정
tiberio_a=(
  (INSTANCE=(HOST=192.168.1.121)
    (PORT=4444)
    (DB_NAME=tiberio)
  )
)
```

2.1.3. 대상 서버 연결 확인

다음과 같이 실행해서 Tiberio_B Instance에 tbdsn.tbr 파일의 설정을 확인한다.

- 접속하려는 Tiberio(Tiberio_A)로 접속한 것을 볼 수 있다.

```
$ tbsql sys/tibero@tibero_a
```

```
tbsQL 7
```

```
TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.
```

```
Connected to Tibero using tibero_a.
```

```
SQL> select * from v$instance;
```

```
INSTANCE_NUMBER  INSTANCE_NAME      DB_NAME      HOST_NAME      PARALLEL  THREAD#  VERSION
STARTUP_TIME  STATUS              SHUTDOWN_PENDING  TIP_FILE
```

```
-----
-----
-----
                0 tibero          tibero      dqa-optane1    NO          0 7
2023/06/12  NORMAL              NO
/hdd/home/sampler22/tibero7/config/tibero.tip
```

```
1 row selected.
```

2.2. DB Link 생성 및 사용

Tibero to Tibero DB Link를 생성할 계정(Tibero_B)으로 접속하여 DB Link 생성 작업을 수행한다.

2.2.1. tbsql에 접속하여 DB Link Object 생성

아래와 같은 문법을 사용하여 DB Link Object를 생성한다.

```
SQL> create database link < DB LINK명 > connect to < 접속 사용자 ID > identified by < 접속
패스워드 >
      2 using < 접속에 사용할 alias >;
```

항목	설명
< DB LINK명 >	생성할 DB Link Object 이름이다.
< 접속 사용자 ID >	대상 DB 서버에 접속할 사용자 이름이다.
< 접속 패스워드 >	대상 DB 서버에 접속할 패스워드이다.
< 접속에 사용할 alias >	Source DB의 tbsn.tbr에 설정된 Alias 이름이다.

다음은 tbsql에 접속하여 DB Link Object 생성에 대한 예이다.

```

$ tbsql tibero/tmax

tbsQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to Tibero.

SQL> create database link T7LINK connect to tibero identified by 'tmax'
      2 using 'tibero_a';

Database Link 'T7LINK' created.

```

참고

DB Link를 생성하기 위해서는 CREATE DATABASE LINK 또는 CREATE PUBLIC DATABASE LINK 권한이 필요하다.

2.2.2. SQL 실행

생성이 완료되면 아래와 같이 < @Link명 >을 붙여 사용한다.

```

SQL> CONN tibero/tmax
Connected to Tibero.

SQL> ls

NAME                                SUBNAME                                TYPE
-----
T7LINK                                DATABASE LINK

SQL> select * from dual@T7LINK;

DUMMY
-----
X

1 row selected.

```

사용 중인 Link의 이름으로 알 수 없는 경우 다음과 같이 View를 조회한다.

```
SQL> select * from user_db_links;
```

OWNER	DB_LINK	USERNAME
HOST	CREATED	
TIBERO	T7LINK	TIBERO
tibero_a	2023/06/14	

```
1 row selected.
```

제3장 Tibero to Oracle

본 장에서는 Tibero에서 DB Link Object를 만들어 Oracle의 데이터를 가져오는 방법을 설명한다.

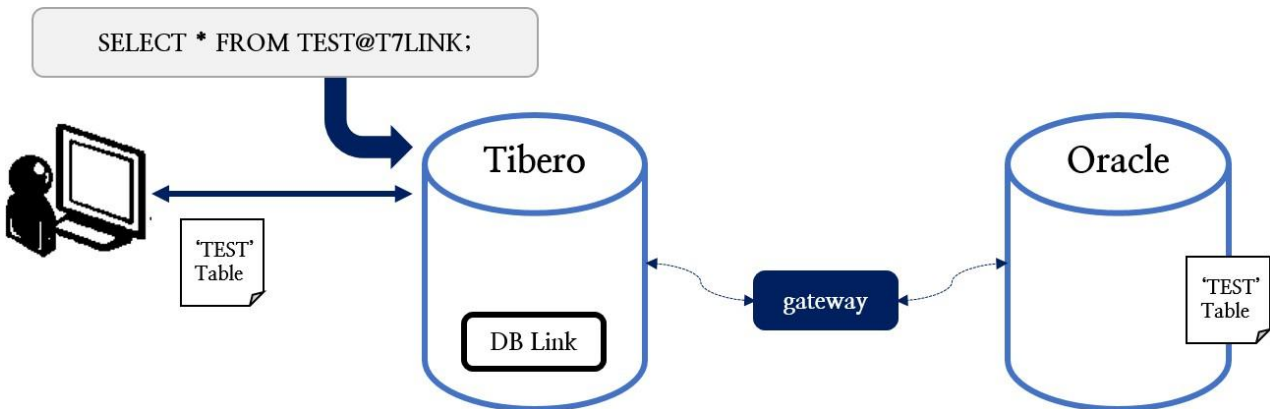
3.1. 개요

DB Link를 통해 질의를 수행할 때 DB Link 대상이 Tibero가 아닌 다른 DBMS라면 각각의 DBMS를 위한 Gateway를 통해 DB Link를 생성하거나 수행할 수 있다.

3.1.1. Gateway

Tibero 서버는 다른 DBMS에 필요한 질의를 해당 Gateway에 전달하고, Gateway는 다른 DBMS에 접속하여 Tibero 서버로부터 전달 받은 질의를 수행한 후 결과를 Tibero 서버로 전송한다. 다른 DBMS의 DB Link 기능을 사용하려는 경우 해당 DBMS에 대한 Gateway 바이너리와 환경설정 파일이 필요하다.

[그림 3.1] Tibero to Oracle 구조도



3.1.2. Gateway 디렉터리 구조

Gateway는 기본적으로 TBGW_HOME 환경변수를 통해 설정 파일을 읽고 로그 파일을 기록한다.

TBGW_HOME 환경변수가 설정되어 있지 않은 경우 디폴트 값은 '\$TB_HOME/client/gateway'이다.

Gateway가 사용하는 설정 파일과 로그 파일이 존재하는 디렉터리 구조는 다음과 같다.

```
$TBGW_HOME
+-- < DBMS Vender명 >
  |-- config
    |-- tbgw.cfg
  |-- log
    |-- Log File
```

```

[/yuha/t7/gateway]$ cd $TBGW_HOME
[/yuha/t7/gateway]$ pwd
/yuha/t7/gateway
[/yuha/t7/gateway]$ tree -F
.
├── gw4orc1*
├── oracle/
│   ├── config/
│   └── log/

```

3 directories, 1 file

다음은 디렉터리와 파일의 설명이다.

- < DBMS Vender명 >/config
Gateway 설정 파일이 생성되는 위치이다.
다음은 해당 디렉터리에 있는 파일의 설명이다.

파일	설명
tbgw.cfg	Gateway 설정 파일로, 사용자가 Gateway와 관련된 설정값을 변경하는 경우 생성한다.

- < DBMS Vender명 >/log
Gateway와 관련된 로그 파일이 생성되는 위치이다.

3.1.3. 확인사항

Tibero to Oracle DB Link를 구성하는 경우 다음의 사항을 확인한다.

- 기본 바이너리 위치
 - Gateway for Oracle 바이너리 파일은 Tibero 버전 및 OS에 따라 위치 및 파일명이 다르다.
 - Tibero 6 이하 버전에서는 oracle과의 dblink를 위한 gw4orc1 바이너리가 oracle 버전에 따라 나누어져 있었는데,
Tibero 7 부터는 gw4orc1 바이너리로 통합되었다.

Tibero 버전	OS	위치 및 파일명
Tibero 7	Unix 계열	\$TB_HOME/client/bin/gw4orc1

- 내부적으로 Oracle 클라이언트 라이브러리를 사용한다.
- Oracle 클라이언트 라이브러리
 - Gateway for Oracle인 경우 내부적으로 Oracle 클라이언트 라이브러리를 사용하여 Oracle에 접속한다.
(Oracle 서버 바이너리 안에 있는 라이브러리, Instant 클라이언트 라이브러리도 가능)
 - Gateway가 설치된 곳에 Oracle 클라이언트 라이브러리가 존재해야 한다.
 - Oracle 클라이언트 라이브러리가 Oracle 서버에 연결되지 않는다면 DB Link 역시 연결될 수 없다.

3.1.4. 제약사항

- LONG RAW, NCLOB을 지원하지 않는다.
- LOB, LONG의 경우 아래와 같은 형태를 일부 지원한다.
 - insert into (local) ~ select from (dblink)
 - insert into (dblink) ~ select from (local)
- Multi Threaded Agent를 지원하지 않는다.
- DB Link를 통한 UDF(User Defined Function) 사용은 PSM(=PL/SQL) 내부에서 가능하다.
 - Select 절에 직접적으로 DB Link를 통한 UDF 사용은 지원하지 않는다.
 - PSM(=PL/SQL) 내부에서 DB Link UDF를 사용하고 해당 PSM을 Select 절에서 간접 사용하는 구조는 가능하다.

3.2. Gateway for Oracle 설정

본 절에서는 Gateway for Oracle의 구성 방식과 설정 방법에 대해서 설명한다.

Gateway for Oracle은 다음의 방식으로 구성한다.

- Local 방식
Gateway를 미리 기동하지 않고 DB Link를 사용할 경우 Gateway 프로세스가 기동되는 방식으로, Tibero 서버와 Gateway for Oracle이 동일 서버에 존재할 경우에 Local 방식으로 구성이 가능하다.
- Listener 방식
Gateway를 미리 기동해서 Listening하는 방식으로 모든 경우에 설정이 가능하다.

3.2.1. Gateway 설정(UNIX 계열)

Local 또는 Listener 방식 중 하나의 방식을 선택하여 설정한다.

설정이 다른 경우 개별적으로 설명하며 동일할 경우는 구분하지 않고 제목에 '공통'이라는 문구를 사용한다.

다음은 UNIX 계열에서 Gateway를 설정하는 과정이다.

1. profile 설정(공통)
2. Oracle 라이브러리 관련 권한 변경(공통)
3. Gateway 바이너리 복사(공통)
4. Network Alias 설정
5. Gateway 환경설정(공통)
6. DB Link 생성 및 확인

각 과정에 대한 상세한 설명은 해당 절의 내용을 참고한다.

3.2.1.1. Local 방식

1. profile 설정(공통)

Gateway for Oracle이 기동되는 사용자의 OS 환경 파일(.profile, .bash_profile 등)에 다음과 같이 환경변수를 설정한다.

- 설정 방법

```
export TBGW_HOME=$TB_HOME/client/gateway
export ORACLE_HOME=<Oracle Home>
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export ORACLE_SID=<Oracle SID>
export PATH=$ORACLE_HOME/bin:$PATH
```

- 설정 예

```
# Gateway for Oracle 19c
export TBGW_HOME=/yuha/t7/gateway
export ORACLE_HOME=/yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export ORACLE_SID=oracle19
export PATH=$ORACLE_HOME/bin:$PATH
```

주의

1. Local 방식의 경우, Tibero 서버가 기동하기 전에 위 환경변수 설정을 가지고 기동이 되어야 한다. 만약, 환경변수 설정 및 Local 방식을 새롭게 구성한다면 Tibero 서버 재기동이 필요하며 재기동이 힘들다면 Listener 방식을 고려한다.
2. OS에 맞게 환경변수 LD_LIBRARY_PATH(Linux), LIBPATH(AIX)를 설정한다.
3. TBGW_HOME은 Tibero 바이너리 폴더 밖으로 설정할 것을 권장한다.

```
$ echo $TB_HOME
/yuha/t7/tibero7

$ echo $TBGW_HOME
/yuha/t7/gateway
```

2. Oracle 라이브러리 관련 권한 변경(공통)

Oracle 라이브러리를 설치한 사용자 또는 root 계정에서 다음과 같이 권한을 부여한다.

```
chmod o+rx $HOME
chmod o+rx $ORACLE_HOME
chmod o+rx $ORACLE_HOME/lib
chmod o+r $ORACLE_HOME/lib/*
chmod o+rx $ORACLE_HOME/bin
chmod o+rx $ORACLE_HOME/bin/sqlplus
chmod o+x $ORACLE_HOME/network
chmod o+x $ORACLE_HOME/network/admin
chmod o+r $ORACLE_HOME/network/admin/tnsnames.ora
```

3. Gateway 바이너리 복사(공통)

지정된 경로에 바이너리 복사를 수행한다.

- 설정 방법

```
$ mkdir $TBGW_HOME
$ cp <Gateway Binary명> $TBGW_HOME/gw4orc1
$ chmod u+x $TBGW_HOME/gw4orc1
```

- 설정 예

```
$ mkdir $TBGW_HOME
$ cp $TB_HOME/client/bin/gw4orc1 $TBGW_HOME/gw4orc1
$ chmod u+x $TBGW_HOME/gw4orc1

$ cd $TBGW_HOME

$ ls
gw4orc1
```

주의

profile 설정과 라이브러리 권한을 설정한 후 ldd gw4orc1를 수행했을 때 Oracle 라이브러리에 대해서 not found 부분이 없어야 한다.

```
$ ldd gw4orc1
linux-vdso.so.1 => (0x00007fff1e1ee000)
libc1ntsh.so.19.1 =>
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libc1ntsh.so.19.1
(0x00007f3c88537000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f3c88333000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f3c88117000)
libm.so.6 => /lib64/libm.so.6 (0x00007f3c87e15000)
libc.so.6 => /lib64/libc.so.6 (0x00007f3c87a47000)
libnnz19.so =>
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libnnz19.so
(0x00007f3c872ff000)
libnsl.so.1 => /lib64/libnsl.so.1 (0x00007f3c870e5000)
librt.so.1 => /lib64/librt.so.1 (0x00007f3c86edd000)
libaio.so.1 => /lib64/libaio.so.1 (0x00007f3c86cdb000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f3c86ac1000)
/lib64/ld-linux-x86-64.so.2 (0x00007f3c8c545000)
libc1ntshcore.so.19.1 =>
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libc1ntshcore.so.19.1
(0x00007f3c86521000)
```

4. Network Alias 설정

Tibero 클라이언트의 Network Alias 설정 파일에 Gateway 정보를 설정한다.

Network Alias 설정 파일명은 Tibero 버전에 따라 각각 다음과 같다.

Tibero 7 : \$TB_HOME/client/config/tbdsn.tbr

- o tbdsn.tbr 설정 방법

```
<Gateway
  Alias명>=( (GATE
  WAY=
    (PROGRAM=<Gateway Binary 경로 및 파일명>)
    (TARGET=<tnsnames.ora에 설정된 Alias>)
    (TX_MODE={GLOBAL | LOCAL})
  )
)
```

항목	설명
PROGRAM	Gateway 바이너리(gw4orcl)의 경로를 반드시 절대 경로로 입력한다.
TARGET	Oracle 클라이언트의 tnsnames.ora에 설정한 Alias 이름으로 대소문자 수준까지 일치해야 하며, 단위 테스트로 sqlplus를 이용해 해당 Alias로 접속이 되는지 확인한다.
TX_MODE	처리 방식을 글로벌 트랜잭션(Global Transaction)과 로컬 트랜잭션(Local Transaction) 중에서 설정한다. Commit을 요청할 경우 글로벌 트랜잭션만 Two-phaseCommit으로 동작한다. TX_MODE의 값은 처리 여부에 따라 다음과 같이 설정할 수 있다. - GLOBAL : 글로벌 트랜잭션인 경우 설정값이다. - LOCAL : 로컬 트랜잭션인 경우 설정값이다.

- TARGET 항목에 대한 접속 확인 방법

TARGET에 입력 값을 sqlplus의 Alias로 작성하여 접속이 되어야 한다.

```
[/yuha/t7/tibero7/client/config]$ sqlplus tibero/tmax@ORCL

SQL*Plus: Release 19.0.0.0.0 - Production on wed Jun 14 14:56:07 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Last successful login time: wed Jun 14 2023 14:55:11 +09:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL>
```

- o tbdsn.tbr 설정 예

```
# gateway for oracle
# local
gw_local=(
    (GATEWAY=
        (PROGRAM=/yuha/t7/gateway/gw4orc1)
        (TARGET=ORCL)
        (TX_MODE=GLOBAL)
    )
)
)
```

5. Gateway 환경설정(공통)

tbgw.cfg 파일에 초기화 파라미터의 설정값을 명시함으로써 Gateway와 관련된 설정을 변경할 수 있다.

Gateway를 설치한 서버에서 \$TBGW_HOME/oracle/config/tbgw.cfg 파일을 설정한다.

o 설정 방법

```
LOG_DIR = <log-dir>
LOG_LVL = <log_lvl>
MAX_LOG_SIZE = <max-log-size>
MAX_LOG_BAKCUP_SIZE = <max-log-backup-size>
FETCH_SIZE = <fetch-size>
SKIP_CHAR_CONV = <skip-char-conv>
```

항목	설명
LOG_DIR	로그가 생성될 디렉터리이다. (기본값: \$TBGW_HOME의 log 폴더)
LOG_LVL	로그레벨을 설정한다. (기본값 : 2) 0-6 까지 설정 가능하며 높을수록 많은 로그를 생성한다. - 운영인 경우 : 1 또는 2를 설정한다. - 문제 대응인 경우 : 5 또는 6을 설정한다.
MAX_LOG_SIZE	로그 파일의 최대 크기를 설정한다. (단위 : Byte)
MAX_LOG_BAKCUP_SIZE	백업 폴더의 최대 크기를 설정한다. 백업 폴더의 크기가 MAX_LOG_BACKUP_SIZE보다 커지면 오래된 파일 순으로 약 1/3 가량의 백업 파일이 삭제된다. 백업이 이루어지는 시점은 2가지 경우가 있다. - 로그를 쓰려 할 때 쓰고 있던 로그 파일 크기가 MAX_LOG_SIZE보다 크다는 것이 감지될 때 - 로그 파일을 쓰던 세션이 종료될 때
FETCH_SIZE	fetch일 경우 사용하는 사이즈로 최대 64KB이다. (기본값 : 32K) 대용량 fetch일 경우 설정값을 높이면 네트워크 부하를 줄일 수 있다.

항목	설명
SKIP_CHAR_CONV	<p>일반적으로 해당 설정은 필요하지 않으며 특수한 상황(US7ASCII 캐릭터 셋에 한글이 들어있는 경우)에만 적용한다.</p> <p>- Y : Gateway가 Oracle로부터 데이터를 받아올 때 환경변수 NLS_LANG을 사용하고 전달받은 데이터를 그대로 Tiberio 서버에 보낸다.</p> <p>- N : Tiberio에서 사용하는 캐릭터 셋에 맞추어 Oracle 서버에 데이터 요청을 한다. 이럴 경우 Oracle에서 해당 캐릭터 셋에 맞게 처리하여 Gateway로 데이터를 보낸다. (기본값)</p>

○ 설정 예

```
$ cd $TBGW_HOME/oracle/config
$ vi tbgw.cfg
LOG_DIR=/yuha/t7/gateway/oracle/log
LOG_LVL=2
MAX_LOG_SIZE=502400000
```

6. DB Link 생성 및 확인

DB Link 생성 방법은 Local과 Listener 방식이 동일하다.

Local 방식에서는 Gateway 프로세스의 기동이 필요하지 않다.

다음은 DB Link 생성과 확인 방법에 대한 설명이다.

○ DB Link 생성 방법

```
create database link <DB Link명> connect to <user명> identified by '<password>'
using '<Network Alias명>';
```

Link를 생성할 경우 권한 에러가 발생하면 아래와 같은 권한을 부여한다.

```
GRANT CREATE DATABASE LINK TO USER_NAME;
GRANT CREATE PUBLIC DATABASE LINK TO USER_NAME;
```

○ DB Link 생성 예

```

$ tbsql tiber0/tmax

tbSQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to Tibero.

SQL> create database link T7LINK connect to tiber0 identified by 'tmax'
      2 using 'gw_local';

Database Link 'T7LINK' created.

```

○ DB Link 생성 확인 예

DB Link 생성을 확인할 때 select문을 'Table명@< DB Link명 >'으로 설정하면 DB Link를 사용할 수 있다.

```

$ tbsql tiber0/tmax

tbSQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to Tibero.

SQL> select * from dual@T7LINK;

DUMMY
-----
X

1 row selected.

```

○ DB Link 이용하여 Oracle의 Table 조회 예

Oracle 에 Table 생성 후 생성한 Database Link 를 이용하여 조회하는 예시이다.
Oracle 에 test 테이블을 생성한다.

```

$ sqlplus tiber0/tmax

SQL*Plus: Release 19.0.0.0.0 - Production on wed Jun 14 16:01:10 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Last Successful login time: wed Jun 14 2023 15:58:11 +09:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> create table test(c1 number);

```

```

Table created.

SQL> insert into test values(100);

1 row created.

SQL> commit;

Commit complete.

SQL> select * from test;

      c1
-----
      100

```

Tibero 에서 생성한 Database Link (T7LINK) 이용하여 데이터를 조회, 삽입할 수 있다.

```

$ tbsql tibero/tmax

tbSQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to Tibero.

SQL> select * from test@T7LINK;

      c1
-----
      100

1 row selected.

SQL> insert into test@T7LINK values(200);

1 row inserted.

SQL> commit;

Commit completed.

SQL> select * from test@T7LINK;

      c1
-----
      100
      200

2 rows selected.

```

3.2.1.2. Listener 방식

1. profile 설정(공통)

Gateway for Oracle이 기동되는 사용자의 OS 환경 파일(.profile, .bash_profile 등)에 다음과 같이 환경변수를 설정한다.

- 설정 방법

```
export TBGW_HOME=$TB_HOME/client/gateway
export ORACLE_HOME=<Oracle Home>
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export ORACLE_SID=<Oracle SID>
export PATH=$ORACLE_HOME/bin:$PATH
```

- 설정 예

```
# Gateway for Oracle 19c
export TBGW_HOME=/yuha/t7/gateway
export ORACLE_HOME=/yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export ORACLE_SID=oracle19
export PATH=$ORACLE_HOME/bin:$PATH
```

주의

1. Local 방식의 경우, Tibero 서버가 기동하기 전에 위 환경변수 설정을 가지고 기동이 되어야 한다. 만약, 환경변수 설정 및 Local 방식을 새롭게 구성한다면 Tibero 서버 재기동이 필요하며 재기동이 힘들다면 Listener 방식을 고려한다.
2. OS에 맞게 환경변수 LD_LIBRARY_PATH(Linux), LIBPATH(AIX)를 설정한다.
3. TBGW_HOME은 Tibero 바이너리 폴더 밖으로 설정할 것을 권장한다.

```
$ echo $TB_HOME
/yuha/t7/tibero7

$ echo $TBGW_HOME
/yuha/t7/gateway
```

2. Oracle 라이브러리 관련 권한 변경(공통)

Oracle 라이브러리를 설치한 사용자 또는 root 계정에서 다음과 같이 권한을 부여한다.

```

chmod o+rx $HOME
chmod o+rx $ORACLE_HOME
chmod o+rx $ORACLE_HOME/lib
chmod o+r $ORACLE_HOME/lib/*
chmod o+rx $ORACLE_HOME/bin
chmod o+rx $ORACLE_HOME/bin/sqlplus
chmod o+x $ORACLE_HOME/network
chmod o+x $ORACLE_HOME/network/admin
chmod o+r $ORACLE_HOME/network/admin/tnsnames.ora

```

3. Gateway 바이너리 복사(공통)

지정된 경로에 바이너리 복사를 수행한다.

- 설정 방법

```

$ mkdir $TBGW_HOME
$ cp <Gateway Binary명> $TBGW_HOME/gw4orc1
$ chmod u+x $TBGW_HOME/gw4orc1

```

- 설정 예

```

$ mkdir $TBGW_HOME
$ cp $TB_HOME/client/bin/gw4orc1 $TBGW_HOME/gw4orc1
$ chmod u+x $TBGW_HOME/gw4orc1

$ cd $TBGW_HOME

$ ls
gw4orc1

```

주의

profile 설정과 라이브러리 권한을 설정한 후 ldd gw4orc1를 수행했을 때 Oracle 라이브러리에 대해서 not found 부분이 없어야 한다.

```

$ ldd gw4orc1
linux-vdso.so.1 => (0x00007fff1e1ee000)
libc1ntsh.so.19.1 =>
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libc1ntsh.so.19.1
(0x00007f3c88537000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f3c88333000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f3c88117000)
libm.so.6 => /lib64/libm.so.6 (0x00007f3c87e15000)
libc.so.6 => /lib64/libc.so.6 (0x00007f3c87a47000)
libnnz19.so =>
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libnnz19.so
(0x00007f3c872ff000)
libnsl.so.1 => /lib64/libnsl.so.1 (0x00007f3c870e5000)
librt.so.1 => /lib64/librt.so.1 (0x00007f3c86edd000)
libaio.so.1 => /lib64/libaio.so.1 (0x00007f3c86cdb000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f3c86ac1000)
/lib64/ld-linux-x86-64.so.2 (0x00007f3c8c545000)

```

```
libclntshcore.so.19.1 =>
/yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libclntshcore.so.19.1
(0x00007f3c86521000)
```

4. Network Alias 설정

Tibero 클라이언트의 Network Alias 설정 파일에 Gateway 정보를 설정한다.

Network Alias 설정 파일명은 Tibero 버전에 따라 각각 다음과 같다.

Tibero 7 : \$TB_HOME/client/config/tbdsn.tbr

- o tbsn.tbr 설정 방법

```
<Gateway
  Alias명>=( (GATE
WAY=
  (LISTENER=
    (HOST=<Gateway IP>)
    (PORT=<Gateway PORT>)
  )
  (TARGET=<tnsnames.ora에 설정된 Alias>)
  (TX_MODE= {GLOBAL | LOCAL })
)
```

항목	설명
LISTENER	Listener가 기동되어 있는 서버에 대한 정보를 입력한다. - HOST : Gateway가 기동되어 있는 서버 IP 주소이다. - PORT : Gateway가 Listening 하는 포트 번호이다. (기본값 : 9999)
TARGET	Oracle 클라이언트의 tnsnames.ora에 설정한 Alias 이름으로 대소문자 수준까지 일치해야 하며, 단위 테스트로 sqlplus를 이용해 해당 Alias로 접속이 되는지 확인한다.
TX_MODE	처리 방식을 글로벌 트랜잭션(Global Transaction)과 로컬 트랜잭션(Local Transaction) 중에서 설정한다. Commit을 요청할 경우 글로벌 트랜잭션만 Two-phaseCommit으로 동작한다. TX_MODE의 값은 처리 여부에 따라 다음과 같이 설정할 수 있다. - GLOBAL : 글로벌 트랜잭션인 경우 설정값이다. - LOCAL : 로컬 트랜잭션인 경우 설정값이다.

- TARGET 항목에 대한 접속 확인 방법

TARGET에 입력 값을 sqlplus의 Alias로 작성하여 접속이 되어야 합니다.

```
[/yuha/t7/tibero7/client/config]$ sqlplus tibero/tmax@ORCL

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Jun 14 14:56:07 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.
```

```
Last Successful login time: Wed Jun 14 2023 14:55:11 +09:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL>
```

o tbdns.tbr 설정 예

```
# gateway for oracle
# listener
gw_listen=(
    (GATEWAY=
        (LISTENER=
            (HOST=192.168.17.24)
            (PORT=9999)
        )
        (TARGET=ORCL)
        (TX_MODE=GLOBAL)
    )
)
```

5. Gateway 환경설정(공통)

tbgw.cfg 파일에 초기화 파라미터의 설정값을 명시함으로써 Gateway와 관련된 설정을 변경할 수 있다.

Gateway를 설치한 서버에서 \$TBGW_HOME/oracle/config/tbgw.cfg 파일을 설정한다.

o 설정 방법

```
LISTENER_PORT = <port-no>
LOG_DIR = <log-dir>
LOG_LVL = <log_lvl>
MAX_LOG_SIZE = <max-log-size>
MAX_LOG_BAKCUP_SIZE = <max-log-backup-size>
FETCH_SIZE = <fetch-size>
SKIP_CHAR_CONV = <skip-char-conv>
```

항목	설명
LISTENER_PORT	Listening 포트 번호로 로컬방식의 경우 필요하지 않다. (기본값 : 9999)
LOG_DIR	로그가 생성될 디렉터리이다. (기본값: \$TBGW_HOME의 log 폴더)
LOG_LVL	로그레벨을 설정한다. (기본값 : 2) 0-6 까지 설정 가능하며 높을수록 많은 로그를 생성한다. - 운영인 경우 : 1 또는 2를 설정한다. - 문제 대응인 경우 : 5 또는 6을 설정한다.
MAX_LOG_SIZE	로그 파일의 최대 크기를 설정한다. (단위 : Byte)

항목	설명
MAX_LOG_BAKCUP_SIZE	백업 폴더의 최대 크기를 설정한다. 백업 폴더의 크기가 MAX_LOG_BACKUP_SIZE보다 커지면 오래된 파일 순으로 약 1/3 가량의 백업 파일이 삭제된다. 백업이 이루어지는 시점은 2가지 경우가 있다. - 로그를 쓰려 할 때 쓰고 있던 로그 파일 크기가 MAX_LOG_SIZE보다 크다는 것이 감지될 때 - 로그 파일을 쓰던 세션이 종료될 때
FETCH_SIZE	fetch일 경우 사용하는 사이즈로 최대 64KB이다. (기본값 : 32K) 대용량 fetch일 경우 설정값을 높이면 네트워크 부하를 줄일 수 있다.
SKIP_CHAR_CONV	일반적으로 해당 설정은 필요하지 않으며 특수한 상황(US7ASCII 캐릭터 셋에 한글이 들어있는 경우)에만 적용한다. - Y : Gateway가 Oracle로부터 데이터를 받아올 때 환경변수 NLS_LANG을 사용하고 전달받은 데이터를 그대로 Tiberio 서버에 보낸다. - N : Tiberio에서 사용하는 캐릭터 셋에 맞추어 Oracle 서버에 데이터 요청을 한다. 이럴 경우 Oracle에서 해당 캐릭터 셋에 맞게 처리하여 Gateway로 데이터를 보낸다. (기본값)

○ 설정 예

```
$ cd $TBGW_HOME/oracle/config

$ vi tbgw.cfg
LISTENER_PORT=9999
LOG_DIR=/yuha/t7/gateway/oracle/log
LOG_LVL=2
MAX_LOG_SIZE=502400000
```

6. DB Link 생성 및 확인

DB Link 생성 방법은 Local과 Listener 방식이 동일하다.

단, Listener 방식은 먼저 Gateway 프로세스의 기동이 필요하다.

Listener 방식에서 Gateway 프로세스의 기동 방법은 다음과 같다.

```
$ ./gw4orc1
```

다음과 같이 Gateway 프로세스 기동 후, gw4orc1 프로세스가 기동된 것을 확인할 수 있다.

```

$ ps -ef |grep gw4orc1
t7      15840 12233  0 15:35 pts/2    00:00:00 grep --color=auto gw4orc1

$ ./gw4orc1

$ ps -ef |grep gw4orc1
t7      15844      1  1 15:35 ?        00:00:00 ./gw4orc1
t7      15857 12233  0 15:35 pts/2    00:00:00 grep --color=auto gw4orc1

```

다음은 지정 포트 번호가 9999일 경우, Listening 포트 번호를 확인하는 예제이다.

```

$ netstat -na |grep 9999
tcp      0      0 0.0.0.0:9999      0.0.0.0:*        LISTEN

```

다음은 DB Link 생성과 확인 방법에 대한 설명이다.

- DB Link 생성 예

```

$ tbsql tibero/tmax

tbsQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to Tibero.

SQL> create database link T7LINK connect to tibero identified by 'tmax'
      2 using 'gw_listen';

Database Link 'T7LINK' created.

```

- DB Link 생성 확인 예

DB Link 생성을 확인할 때 select 문을 'Table명@< DB Link명 >'으로 설정하면 DB Link를 사용할 수 있다.

```

$ tbsql tibero/tmax

tbsQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to Tibero.

SQL> select * from dual@T7LINK;

DUMMY
-----
X

1 row selected.

```

3.3. Oracle 클라이언트

본 절에서는 Oracle 서버와 클라이언트 호환성 확인과 Instant 클라이언트 설정 방법에 대해서 설명한다.

3.3.1. 호환성 확인

Tibero to Oracle DB Link의 경우 Gateway가 내부적으로 Oracle 클라이언트를 사용하므로 Oracle 클라이언트와 서버가 호환되지 않으면 DB Link를 연결할 수 없다.

아래의 호환성 도표를 참고하여 연결 가능 여부를 확인한다.

[그림 3.2] Oracle 서버와 클라이언트 호환성 도표

Client Version	Server Version								
	19c	18c	12.2.0	12.1.0	11.2.0	11.1.0	10.2.0	10.1.0	9.2.0
19c	Yes	Yes	Yes	Yes	Yes ^{#9}	No	No	No ^{#3}	No ^{#3}
18c	Yes	Yes	Yes	Yes	Yes ^{#9}	No	No	No ^{#3}	No ^{#3}
12.2.0	Yes	Yes	Yes	Yes	Yes ^{#9}	No	No	No ^{#3}	No ^{#3}
12.1.0	Yes	Yes	Yes	Yes	Yes	Was	Was ^{#7}	No ^{#3}	No ^{#3}
11.2.0	Yes ^{#9}	Yes ^{#9}	Yes ^{#9}	Yes	Yes	Was	Was ^{#7}	No	Was ^{#5}
11.1.0	No	No	No	Was	Was	Was	Was ^{#7}	Was ^{#6}	Was ^{#5}
10.2.0	No ^{#10}	No ^{#10}	No ^{#10}	Was ^{#7}	Was ^{#7}	Was ^{#7}	Was	Was	Was ^{#5}
10.1.0 ^{#4}	No	No	No	No	Was ^{#6}	Was ^{#6}	Was	Was	Was
9.2.0	No	No	No	No ^{#8}	Was ^{#5}	Was ^{#5}	Was ^{#5}	Was	Was

3.3.2. Instant 클라이언트 설정

Gateway for Oracle의 경우 내부적으로 Oracle 클라이언트 라이브러리 또는 서버 라이브러리를 이용하여 Oracle 에 연결한다.

Oracle 클라이언트 또는 서버를 설치하기가 어려운 경우 Instant 클라이언트를 이용한 DB Link 설정이 가능하다.

다음은 Oracle Instant 클라이언트를 설정하는 과정에 대한 설명이다.

1. Instant 클라이언트 다운로드 및 압축 해제
2. profile 설정
3. tnsnames.ora 생성 및 설정
4. Network Alias 설정(UNIX 계열 참고)
5. Gateway 환경설정(UNIX 계열 참고)
6. DB Link 생성 및 확인(UNIX 계열 참고)

각 과정에 대한 상세한 설명은 해당 절의 내용을 참고한다.

4~6 과정은 3.2.1.1 과 3.2.1.2의 과정과 동일하여 참고하여 진행한다.

1. Instant 클라이언트 다운로드 및 압축 해제

Oracle 홈페이지 접속한 후 Gateway가 설치된 서버의 OS와 bit에 맞는 Instant 클라이언트를 다운로드하여 압축을 해제한다.

다음은 Linux 버전의 실행 결과이다.

```
$ unzip instantclient-basic-linux.x64-19.16.0.0.odbu.zip
Archive:  instantclient-basic-linux.x64-19.16.0.0.odbu.zip
  inflating: instantclient_19_16/adrci
  inflating: instantclient_19_16/BASIC_LICENSE
  inflating: instantclient_19_16/BASIC_README
  inflating: instantclient_19_16/genezi
  inflating: instantclient_19_16/libclntshcore.so.19.1
    linking: instantclient_19_16/libclntsh.so -> libclntsh.so.19.1
    linking: instantclient_19_16/libclntsh.so.10.1 -> libclntsh.so.19.1
    linking: instantclient_19_16/libclntsh.so.11.1 -> libclntsh.so.19.1
    linking: instantclient_19_16/libclntsh.so.12.1 -> libclntsh.so.19.1
    linking: instantclient_19_16/libclntsh.so.18.1 -> libclntsh.so.19.1
  inflating: instantclient_19_16/libclntsh.so.19.1
  inflating: instantclient_19_16/libipc1.so
  inflating: instantclient_19_16/libmq11.so
  inflating: instantclient_19_16/libnnz19.so
    linking: instantclient_19_16/libocci.so -> libocci.so.19.1
    linking: instantclient_19_16/libocci.so.10.1 -> libocci.so.19.1
    linking: instantclient_19_16/libocci.so.11.1 -> libocci.so.19.1
    linking: instantclient_19_16/libocci.so.12.1 -> libocci.so.19.1
    linking: instantclient_19_16/libocci.so.18.1 -> libocci.so.19.1
  inflating: instantclient_19_16/libocci.so.19.1
  inflating: instantclient_19_16/libociei.so
  inflating: instantclient_19_16/libocijdbc19.so
  inflating: instantclient_19_16/liboramysql19.so
    creating: instantclient_19_16/network/
  inflating: instantclient_19_16/ojdbc8.jar
  inflating: instantclient_19_16/ucp.jar
  inflating: instantclient_19_16/uidrvci
  inflating: instantclient_19_16/xstreams.jar
    creating: instantclient_19_16/network/admin/
  inflating: instantclient_19_16/network/admin/README
finishing deferred symbolic links:
  instantclient_19_16/libclntsh.so -> libclntsh.so.19.1
  instantclient_19_16/libclntsh.so.10.1 -> libclntsh.so.19.1
  instantclient_19_16/libclntsh.so.11.1 -> libclntsh.so.19.1
  instantclient_19_16/libclntsh.so.12.1 -> libclntsh.so.19.1
  instantclient_19_16/libclntsh.so.18.1 -> libclntsh.so.19.1
  instantclient_19_16/libocci.so -> libocci.so.19.1
  instantclient_19_16/libocci.so.10.1 -> libocci.so.19.1
  instantclient_19_16/libocci.so.11.1 -> libocci.so.19.1
  instantclient_19_16/libocci.so.12.1 -> libocci.so.19.1
  instantclient_19_16/libocci.so.18.1 -> libocci.so.19.1
```

2. Profile 설정

다음은 profile 설정 예이다. TNS_ADMIN의 경우 tnsnames.ora 파일의 위치를 설정한다.

```
# Instant client
export TBGW_HOME=/yuha/t7/gateway
export ORACLE_HOME=/yuha/t7/instantclient_19_16
export LD_LIBRARY_PATH=$ORACLE_HOME:$LD_LIBRARY_PATH
export PATH=.:$ORACLE_HOME:$PATH
export TNS_ADMIN=$ORACLE_HOME
```

주의

1. LD_LIBRARY_PATH 설정에 해당 경로의 모든 라이브러리가 존재하므로 \$ORACLE_HOME의 설정에 주의한다.
2. OS에 맞게 환경변수 LD_LIBRARY_PATH(Linux), LIBPATH(AIX)를 설정한다.
3. profile 설정 후 ldd gw4orcl를 수행했을 때 Oracle 라이브러리에 대해서 not found 부분이 없어야 한다.

```
$ ldd gw4orcl
```

```
$ ldd gw4orcl
linux-vdso.so.1 => (0x00007ffd9d59b000)
libc1ntsh.so.19.1 => /yuha/t7/instantclient_19_16/libc1ntsh.so.19.1
(0x00007f4b5b99a000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f4b5b796000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f4b5b57a000)
libm.so.6 => /lib64/libm.so.6 (0x00007f4b5b278000)
libc.so.6 => /lib64/libc.so.6 (0x00007f4b5aea000)
libnnz19.so => /yuha/t7/instantclient_19_16/libnnz19.so
(0x00007f4b5a836000)
libnsl.so.1 => /lib64/libnsl.so.1 (0x00007f4b5a61c000)
librt.so.1 => /lib64/librt.so.1 (0x00007f4b5a414000)
libaio.so.1 => /lib64/libaio.so.1 (0x00007f4b5a212000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f4b59ff8000)
/lib64/ld-linux-x86-64.so.2 (0x00007f4b5fab5000)
libc1ntshcore.so.19.1 =>
/yuha/t7/instantclient_19_16/libc1ntshcore.so.19.1 (0x00007f4b59a54000)
```

3. tnsnames.ora 생성 및 설정

\$TNS_ADMIN으로 설정된 경로에 tnsnames.ora 파일을 생성한다.

- 설정 방법

```

$ cd $TNS_ADMIN
$ vi tnsnames.ora
<Alias명> =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <Oracle 서버 접속 IP>)
      (PORT = <Oracle 서버 접속 Port>))
    (CONNECT_DATA = (SERVER = DEDICATED)(SERVICE_NAME = <Oracle 서버의 SID>))
  )
)

```

항목	설명
< Alias명 >	tbdsn.tbr 설정의 TARGET 부분과 대소문자 수준까지 일치해야한다.
ADDRESS	대상 Oracle에 대한 접속 정보를 설정한다. - PROTOCOL : 프로토콜을 설정한다. - HOST : 대상 ORACLE 서버의 접속 IP 주소 정보이다. - PORT : 대상 ORACLE 서버의 접속 포트 번호 정보이다.
CONNECT_DATA	대상 Oracle 서버의 연결 방식 및 SID를 설정한다. - SERVER : 서버의 연결 방식이다. - SERVICE_NAME : 대상 ORACLE 서버의 SID 이다.

○ 설정 예

다음은 Oracle 19c에 해당하는 설정으로 각 Oracle 버전마다 문법이 일부 달라질 수 있다.

만약 \$TNS_ADMIN으로 경로를 설정하지 않았다면, 기본 경로는 "ORACLE_HOME/network/admin" 이다.

```

$ cd $TNS_ADMIN

$ cat tnsnames.ora
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.17.24)(PORT = 1521))
    (CONNECT_DATA = (SERVER = DEDICATED)(SERVICE_NAME = oracle19))
  )
)

```

3.4. 문제 해결

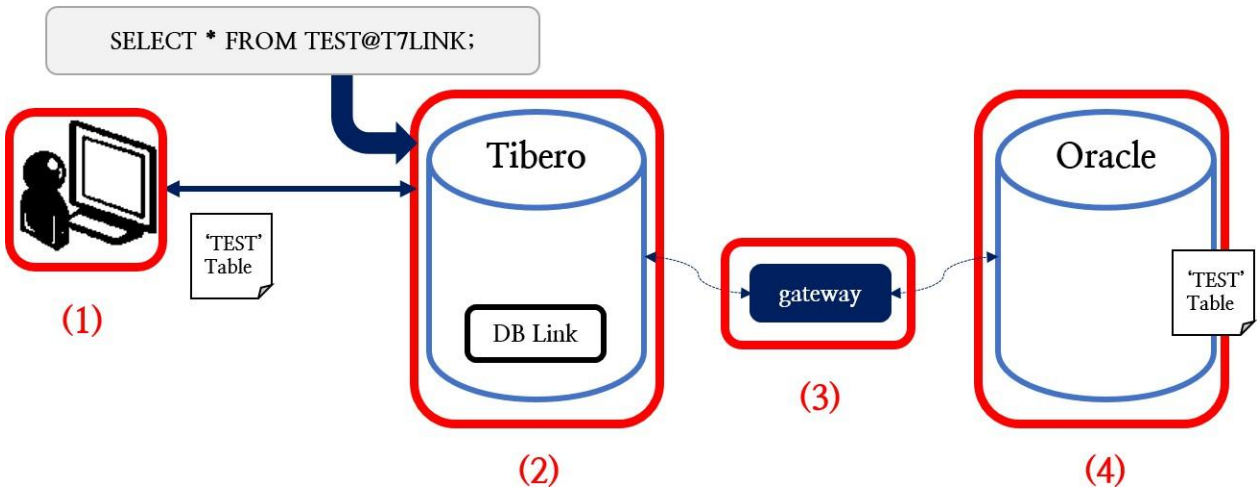
3.4.1. 데이터 조회

글자 깨짐 현상

DB Link를 통하여 Oracle의 데이터를 조회할 때 글자가 깨지는 현상이 발생한다.

다음은 글자 깨짐 현상이 발생할 때 확인해야 할 항목을 나타내는 그림으로 상황에 맞게 각 항목의 적절한 설정이 필요하다.

[그림 3.3] 글자 깨짐 현상이 발생할 경우 확인 요소



- 클라이언트 환경 (1)

- 터미널창으로 접속하여 tsql을 통해 확인할 경우

OS 언어 설정(LANG, LC_ALL 등), 접속하는 SSH 또는 텔넷의 언어 설정, TB_NLS_LANG 환경변수 값을 확인한다.

- 클라이언트 Tool을 사용하여 확인할 경우

구분	설명
Tibero Studio 를 사용하는 경우	접속할 때 지정하는 캐릭터 셋 설정 확인이 필요하다. 만약, 미지정의 경우 접속하는 Tibero 서버의 캐릭터 셋으로 설정된다.
ODBC를 사용할 경우 (Orange for Tibero 등)	TB_NLS_LANG 환경변수 값을 확인한다.

- Tibero 캐릭터 셋 (2)

Oracle 캐릭터 셋이 Tibero에서 호환이 되는 캐릭터 셋인지 확인한다.

예를 들어 Tibero의 캐릭터 셋이 MSWIN949(한글), Oracle 캐릭터 셋이 JA16SJIS(일본어)일 경우 호환되지 않는다.

- Gateway (3)

Gateway는 Oracle에서 Tibero로 데이터를 가지고 올 때 아래와 같은 2개의 환경변수의 영향을 받는다.

환경변수	설명
TB-NLS_LANG	Gateway에서 Tiber로 접속할 때 영향을 받는다.
NLS_LANG	Gateway에서 Oracle로 접속할 때 영향을 받는다.

참고

Tibero6 이하의 TB-NLS_LANG의 기본값은 MSWIN949(한글)이므로 Tiber의 환경이 한글이 아닌 다른 환경일 경우 TB-NLS_LANG 설정을 고려한다.

Tibero7 의 TB-NLS_LANG의 기본값은 UTF8이다.

- Oracle (4)

Oracle 자체에서 깨진 글자가 아닌지 확인한다.

3.5. Tibero7 의 gw_install.sh 수행

Tibero7 에서 gw4orcl 로 Gateway 바이너리가 통합되며 \$TB_HOME/client/bin 에 gw_install.sh 스크립트가 추가되었다.

gw_install.sh 수행 시, \$TBGW_HOME와 하위 폴더가 자동으로 생성되어 설정된다.

```
$ sh gw_install.sh
Configuring gateway environment for tiber to oracle...
TBGW_HOME is /yuha/t7/gateway.
Gateway configuration file is created in /yuha/t7/gateway/oracle/config. Check the file,
and add or rewrite options if you want.
Gateway install is done.
```

```
$ cd $TBGW_HOME

$ tree -f
.
├── ./oracle
│   ├── ./oracle/config
│   │   └── ./oracle/config/tbgw.cfg
│   ├── ./oracle/lib
│   │   └── ./oracle/lib/libclntsh.so.19.1 ->
│   │       /yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libclntsh.so
│   └── ./oracle/log
4 directories, 2 files
```

ldd gw4orcl 확인할 수 있다.

```
$ pwd
/yuha/t7/tibero7/client/bin

$ ldd gw4orcl
linux-vdso.so.1 => (0x00007ffffd65df000)
```

```

libclntsh.so.19.1 =>
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libclntsh.so.19.1
(0x00007f055e271000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f055e06d000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f055de51000)
libm.so.6 => /lib64/libm.so.6 (0x00007f055db4f000)
libc.so.6 => /lib64/libc.so.6 (0x00007f055d781000)
libnnz19.so => /youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libnnz19.so
(0x00007f055d039000)
libnsl.so.1 => /lib64/libnsl.so.1 (0x00007f055ce1f000)
librt.so.1 => /lib64/librt.so.1 (0x00007f055cc17000)
libaio.so.1 => /lib64/libaio.so.1 (0x00007f055ca15000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f055c7fb000)
/lib64/ld-linux-x86-64.so.2 (0x00007f056227f000)
libclntshcore.so.19.1 =>
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/lib/libclntshcore.so.19.1
(0x00007f055c25b000)

```

아래와 같이 접속을 확인할 수 있다.

```

$ tbsql tibero/tmax@gw_listen

tbsQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to using gw_listen.

SQL> select * from dual;

DUMMY
-----
X

1 row selected.

```

제 4장 Oracle to Tibero

본 장에서는 Oracle에서 DB Link Object를 만들어 Tibero의 데이터를 가져오는 DB Link 생성을 설명한다.

4.1. 개요

Oracle에서는 이기종 분산 데이터베이스 환경에서 Non-Oracle 시스템 데이터에 접근하기 위해 Heterogeneous Services(HS)와 Database Gateway(DG) 모듈을 지원하고 있으며, 해당 모듈과 타 DB의 Generic ODBC Driver 연동작업을 통해 DB Link를 생성할 수 있다.

참고

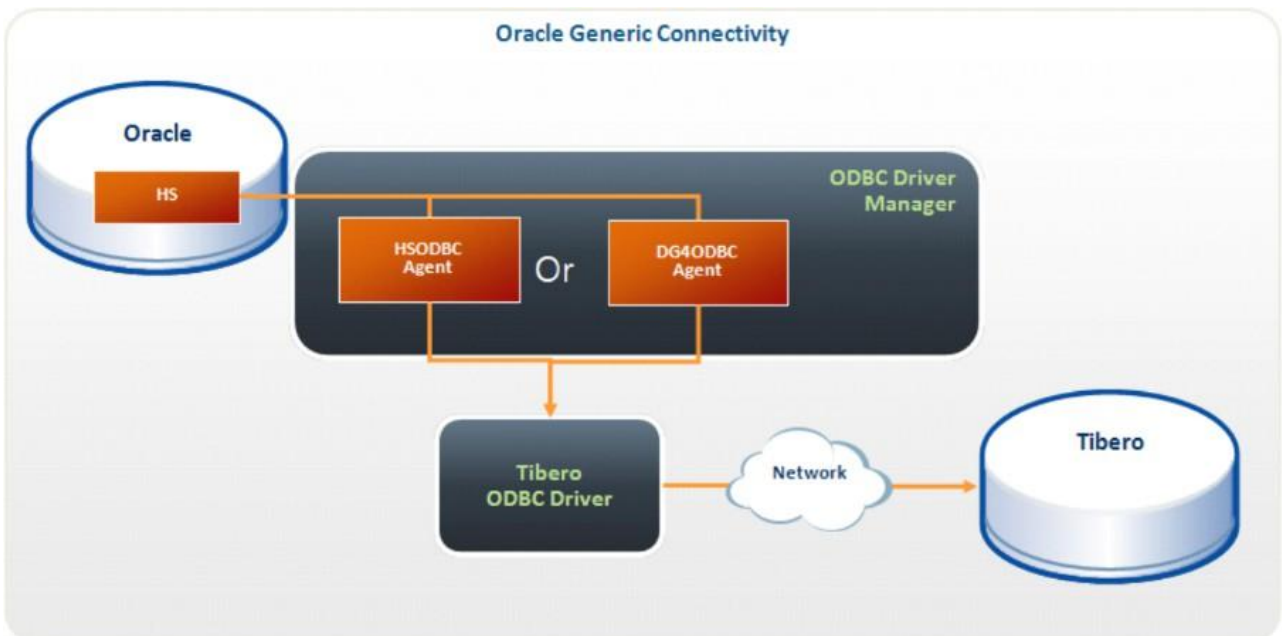
Oracle Gateway 는 이기종간 서비스(Heterogeneous Connectivity)에 의해 Oracle SQL 을 Non-Oracle 시스템의 적절한 언어로 변환하는 기능과 Non-Oracle 시스템의 메타데이터를 로컬 서식에서 나타내는 데이터 디렉터리 변환의 2 가지 종류의 변환이 제공된다.

변환을 사용할 수 없는 경우에는 Pass-Through 기능을 사용해 Oracle 이외의 시스템에 시스템 고유의 SQL 문을 실행할 수 있다.

Pass-Through 이란, Oracle Database 에서 해석 없이 non-oracle 시스템으로 직접 SQL 을 보낼 수 있는 기능이다.

DBMS_HS_PASSTHROUGH 패키지를 사용해서 이기종 DB 고유의 커멘드나 문장을 어플리케이션 또는 Oracle 서버에서 이기종 데이터베이스로 전달 가능하다.

[그림 4.1] Oracle to Tibero 구조도



4.1.1. Oracle Gateway

Oracle 버전에 따라 사용하는 Gateway가 다르며 Oracle to Tiberio의 경우 Oracle에서 제공하는 Gateway를 이용한다.

Oracle 10g 이하 버전에서는 HSODBC를 이용하며, Oracle 11g 이상 버전부터는 DG4ODBC를 사용한다.

분류	설명
HSODBC	Oracle 10g에서 사용한다. Oracle 10g 클라이언트 바이너리에는 존재하지 않으며 Oracle 10g 서버 바이너리에만 존재한다.
DG4ODBC	Oracle 11g 이상에서 사용한다. Oracle 11g 이상 클라이언트 바이너리에는 존재하지 않으며 Oracle 11g 이상 Gateway 바이너리 또는 Oracle 11g 이상 서버 바이너리에 존재한다.

참고

여러 개의 Oracle Database Gateway를 설치할 경우, 제품 구성요소를 Oracle Database Gateway for ODBC로 선택한다.

제약사항

다음은 Oracle에서 제공하는 Gateway인 HSODBC와 DG4ODBC의 제약사항이다.

Database Gateway for ODBC Limitations

- BLOB/CLOB data cannot be read through pass-through queries
- Updates or deletes that include unsupported functions within a WHERE clause are not allowed
- Stored procedures are not supported
- Cannot participate in distributed transactions
- they support single-site transactions only. Gateways are an integral component of the overall heterogeneous connectivity solution. Specifically, they provide: SQL and Data Dictionary Translation information
- Datatype Translation
- Callback link is not supported

참고

위 내용은 Oracle에 대한 제약 사항으로 Tiberio 제약사항은 아니다. 내용의 자세한 설명은 Oracle 홈페이지(<http://www.oracle.com>)를 참고한다.

4.1.2. DB Link 연결방식

연결방식은 OS에 따라 다르다.

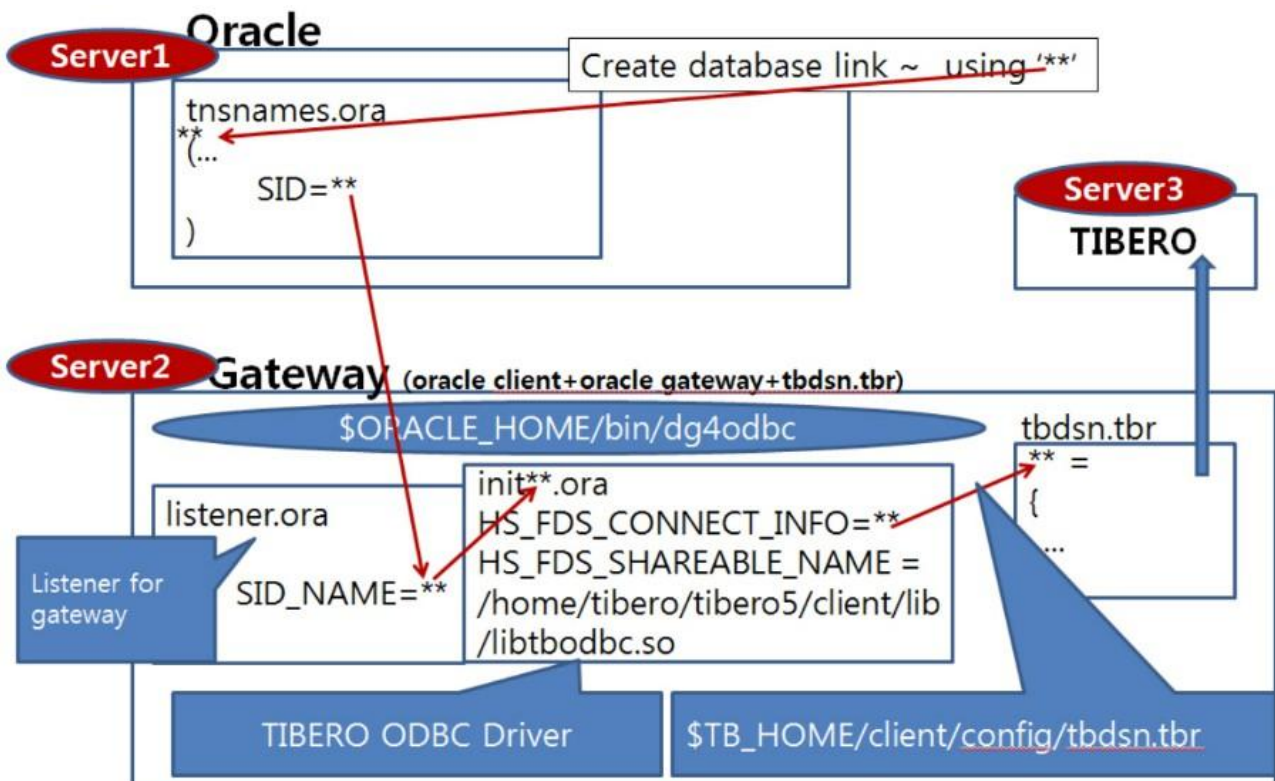
분류	설명
UNIX 계열	<p>아래 2가지 방법이 가능하다.</p> <ul style="list-style-type: none"> - HSODBC(10g) 또는 DG4ODBC(11g 이상)에서 Tiberio ODBC Driver 직접 연결 방식의 경우, 별도의 Manager 설치가 필요하지 않다. - Driver Manager 사용 연결방식 또는 iodbc, unixodbc 사용의 경우, 별도의 Manager 설치가 필요하다.

4.2. DB Link 설정

Tiberio ODBC Driver 직접 연결방식의 경우 Oracle to Tiberio 호출 흐름은 다음과 같다.

Oracle 서버와 Gateway 서버가 분리될 경우 다음의 설정으로 수행되지만 분리되지 않는다면 모든 설정을 Oracle 서버에서 한다.

[그림 4.2] Oracle to Tiberio 호출 흐름



1. create database ~ using 문구를 수행한다.
2. tnsnames.ora 파일의 Alias명의 SID를 찾는다.
3. listener.ora 파일에서 SID를 찾아서 다음 단계를 수행한다.
4. init{sid}.ora 파일의 설정 정보를 읽어온다.
5. tbdsn.tbr로 Tibero 서버를 찾는다.

4.2.1. UNIX 계열 (Linux 포함)

다음은 UNIX 계열에서 DB Link 를 설정하는 과정이다.

1. 사전확인 및 준비사항
2. Tibero 클라이언트 설치
3. tnsnames.ora 설정
4. Listener 설정 및 기동
5. Oracle Gateway 환경설정 (init<DSN>.ora)
6. DB Link 생성 및 확인

각 과정에 대한 상세한 설명은 해당 절의 내용을 참고한다.

1. 사전확인 및 준비사항

다음과 같은 사항들을 확인 및 준비한다.

- Oracle Gateway 설치 여부 확인

다음과 같이 조회하여 파일 사이즈가 0보다 크면 설치된 것이고, 0으로 나온다면 설치가 되지 않은 것이다.

```
$ ls -al $ORACLE_HOME/bin/dg4odbc
-rwxr-x--x 1 oracle dba 738168 4월 10 15:49
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/bin/dg4odbc
```

- Oracle Gateway 파일의 bit 확인

OS 명령어인 file 명령어를 사용하여 32bit인지 64bit인지 확인한다.

다음은 Gateway 파일의 bit를 확인하는 예제이다. 실행 결과를 보면 해당 파일은 Linux 64bit인 것을 확인할 수 있다.

```
$ file $ORACLE_HOME/bin/dg4odbc
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/bin/dg4odbc: ELF 64-bit LSB
executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for
GNU/Linux 2.6.32, BuildID[sha1]=05c8c6e1646c7f78af76fec4fc034f83383843f1, not
stripped
```

참고

1. Oracle 10g의 경우 file ORACLE_HOME/bin/hsodbc로 조회한다.
2. Gateway가 32bit라면 내부적으로 사용하는 Tibero ODBC Driver 역시 32bit를 사용해야 한다. 하지만 64bit로 설치되어 있어도 해당 모듈이 32bit인 경우도 있다.

2. Tibero 클라이언트 설치

Oracle Gateway가 설치된 서버에 설치한다.

실제 연동을 위해서는 '\$TB_HOME/client/lib/libtodb.so(또는 libtodb.sl)'만 있어도 가능하지만 단위 테스트 및 설치의 편의성을 위하여 Tibero 클라이언트 설치를 권장한다.

다른 사용자에게 Tibero 서버가 설치되어 있다면 권한 설정을 한 후 해당 라이브러리를 이용할 수 있다.

별도로 Tibero 클라이언트 바이너리가 없을 때 Tibero 서버 바이너리를 가져다가 다음과 같이 설정을 한다.

Alias 항목은 편의성을 위하여 설정한 부분으로 필요하지 않다면 설정하지 않아도 된다.

- o profile 설정

```
# Oracle To Tibero DB Link Setting
export TB_HOME=/yuha/oracle/tibero7
export LD_LIBRARY_PATH=$TB_HOME/client/lib:$LD_LIBRARY_PATH
export PATH=$TB_HOME/client/bin:$PATH
alias nadm='cd $ORACLE_HOME/network/admin'
alias hadm='cd $ORACLE_HOME/hs/admin'
```

주의

OS에 맞게 환경변수 LD_LIBRARY_PATH(Linux), LIBPATH(AIX)를 설정한다.

- o tbdns.tbr 설정

\$TB_HOME/client/config/tbdns.tbr 파일을 설정한다.

- 설정 방법

```
<Alias명>=( (INSTANCE=(HOST=<Tibero 서버 IP>)
(PORT=<Tibero 서버 Port>)
(DB_NAME=<Tibero 서버 DB명>)
)
)
```

항목	설명
< Alias 명 >	임의로 설정한다. 단, Tibero ODBC Driver 직접 연결방식의 경우 Oracle Gateway 환경설정의 HS_FDS_CONNECT_INFO 파라미터에 설정된 값과 일치해야 한다.
HOST	DB Link의 경우 사용하는 Tibero 서버의 접속 IP 주소 정보이다.
PORT	DB Link의 경우 사용하는 Tibero 서버의 접속 포트 번호 정보이다.
DB_NAME	DB Link의 경우 사용하는 Tibero 서버의 접속 DB_NAME 정보이다.

■ 설정 예

```
o2t=(
  (INSTANCE=(HOST=192.168.17.24)
    (PORT=5555)
    (DB_NAME=tibero)
  )
)
```

○ Tibero 접속 테스트

각 설정이 완료된 후 Alias명으로 접속해서 설정 정보를 확인한다. 접속에 문제가 발생하는 경우 tbdns.tbr 파일의 설정 정보를 확인한다.

```
$ tbsql sys/tibero@o2t

tbsQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to vGlizXJv using o2t.
```

3. tnsnames.ora 설정

DB Link Object를 생성하는 Oracle 서버에서 \$ORACLE_HOME/network/admin/tnsnames.ora 파일을 설정한다.

○ 설정 방법

```
<Alias명>=
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = <Gateway 서버 IP>)
      (PORT = <Gateway 서버 Port>))
    (CONNECT_DATA = (SID = <Gateway 서버 SID명>))
    (HS = OK)
  )
```

항목	설명
< Alias명 >	임의로 설정한다. 단, DB Link를 생성할 때 using 절에 해당 Alias를 주어야 한다.
ADDRESS	대상 Oracle에 대한 접속 정보를 설정한다. - PROTOCOL : 프로토콜을 설정한다. - HOST : Gateway 서버의 접속 IP 주소 정보이다. - PORT : Gateway 서버의 접속 포트 번호 정보이다.
CONNECT_DATA	대상 Oracle 서버의 SID를 설정한다. - SID : 임의로 설정한다. 단, Gateway 서버의 listener.ora의 SID_NAME 설정과 일치해야 한다.

항목	설명
HS	OK로 설정한다.

○ 설정 예

```

tb7=
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.17.24)(PORT = 1522))
    (CONNECT_DATA = (SID = dg4_tibero))
    (HS = OK)
  )

```

주의

괄호가 누락되는 경우, 문제를 찾을 수 없으므로 위 문법에 주의한다.

4. Listener 설정 및 기동

해당 설정 및 기동은 Gateway 서버에 설정한다.

만약 별도로 Oracle 서버와 분리되지 않았다면 Oracle 서버에 설정한다.

- listener.ora 설정

\$ORACLE_HOME/network/admin/listener.ora 파일을 설정한다.

- 설정 방법

```

<LISTENER_NAME> =
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=<Listening할 IP>)(PORT=<Listening할 Port>))
  )
  SID_LIST_<LISTENER_NAME> =
    (SID_LIST=
      (SID_DESC=
        (SID_NAME=<Oracle 서버 SID명>)
        (ORACLE_HOME=<Oracle 홈주소>)
        (PROGRAM= {dg4odbc | hsodbc})
      )
    )
  )

```

항목	설명
< LISTENER_NAME >	임의로 설정한다. Listener를 시작하거나 종료할 때 해당 이름을 사용한다.

항목	설명
ADDRESS	Listening 할 정보를 설정한다. - PROTOCOL : 프로토콜을 설정한다. - HOST : Listening할 IP 주소 정보이다. - PORT: Listening할 포트 번호 정보이다.
SID_NAME	임의로 설정한다. 단, Oracle 서버에서 설정된 tnsnames.ora의 SID 설정과 일치해야 한다. Oracle Gateway 설정 파일을 찾을 때 해당 이름을 사용한다.
ORACLE_HOME	Gateway 서버에 설치된 Oracle 홈주소이다.
PROGRAM	- dg4odbc : Gateway 서버가 Oracle 11g이상일 때 설정한다. - hsdobc : Gateway 서버가 Oracle 10g일 때 설정한다.

○ 설정 예

```

LTIBERO =
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.17.24)(PORT=1522))
  )

SID_LIST_LTIBERO =
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=dg4_tibero)

(ORACLE_HOME=/yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1)
      (PROGRAM=dg4odbc)
    )
  )

```

• Listener 기동

listener.ora에서 설정한 LISTENER_NAME으로 기동, 종료, 상태 정보 등을 확인할 수 있다.

○ 기동

Oracle to Tibero DB Link를 사용하기 위해서 Listener를 기동한다.

```
$ lsnrctl start <listener_name>
```

다음은 LISTENER_NAME이 LTIBERO일 경우 Listener 기동 예제이다.

```

]$ lsnrctl start ltibero

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 15-JUN-2023 17:17:49

Copyright (c) 1991, 2019, Oracle. All rights reserved.

```

```

Starting /yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1/bin/tnslsnr: please
wait...

TNSLSNR for Linux: Version 19.0.0.0.0 - Production
System parameter file is
/yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1/network/admin/listener.ora
Log messages written to
/yuha/oracle/ORA19/app/oracle/diag/tnslsnr/haeun/ltibero/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.17.24)
(PORT=1522)))

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.17.24)(PORT=1522))
STATUS of the LISTENER
-----
Alias                ltibero
Version              TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date           15-JUN-2023 17:17:49
Uptime               0 days 0 hr. 0 min. 0 sec
Trace Level          off
Security             ON: Local OS Authentication
SNMP                 OFF
Listener Parameter File
/yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1/network/admin/listener.ora
Listener Log File
/yuha/oracle/ORA19/app/oracle/diag/tnslsnr/haeun/ltibero/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.17.24)(PORT=1522)))
Services Summary...
Service "dg4_tibero" has 1 instance(s).
  Instance "dg4_tibero", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

```

다음은 Listening 포트 번호를 확인하는 예제이다. 실행 결과를 보면 포트 번호가 1522인 것을 확인할 수 있다.

```

$ netstat -na |grep 1522
tcp        0      0 192.168.17.24:1522    0.0.0.0:*          LISTEN

```

○ 종료

```

$ lsnrctl stop <listener_name>

```

```

$ lsnrctl stop ltibero

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 15-JUN-2023 16:13:05

Copyright (c) 1991, 2019, Oracle. All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.17.24)(PORT=1522))
The command completed successfully

```

○ 상태 정보

```
$ lsnrctl status <listener_name>
```

```
$ lsnrctl status ltibero

LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 15-JUN-2023 17:18:20

Copyright (c) 1991, 2019, Oracle. All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=192.168.17.24)(PORT=1522))
STATUS of the LISTENER
-----
Alias                ltibero
Version              TNSLSNR for Linux: Version 19.0.0.0.0 - Production
Start Date           15-JUN-2023 17:18:15
Uptime                0 days 0 hr. 0 min. 4 sec
Trace Level          off
Security              ON: Local OS Authentication
SNMP                  OFF
Listener Parameter File
/yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1/network/admin/listener.ora
Listener Log File
/yuha/oracle/ORA19/app/oracle/diag/tnslsnr/haeun/ltibero/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=192.168.17.24)(PORT=1522)))
Services Summary...
Service "dg4_tibero" has 1 instance(s).
  Instance "dg4_tibero", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```

● tns ping 확인

Listener까지 기동을 했다면 tns ping을 이용하여 단위 테스트를 한다.

tnsnames.ora를 설정한 Oracle서버에서 확인하며 해당 파일의 Alias 이름을 이용하여 테스트를 수행한다.

다음은 tns ping으로 확인하는 방법이다.

```
$ tns ping <alias_name>
```

다음은 tns ping을 실행한 예로 결과를 보면 tnsnames.ora에서 설정한 Alias 이름이 tibero인 것을 확인할 수 있다.

```
$ tns ping tb7

TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 15-JUN-2023 17:18:34

Copyright (c) 1997, 2019, Oracle. All rights reserved.

Used parameter files:
/yuha/oracle/ORA19/app/oracle/product/19.0.0/db_1/network/admin/sqlnet.ora
```

Used TNSNAMES adapter to resolve the alias

Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.17.24)(PORT = 1522)) (CONNECT_DATA = (SID = dg4_tibero)) (HS = OK))

OK (0 msec)

5. Oracle Gateway 환경설정 (init{DSN}.ora)

Gateway 서버에 설정하고 '\$ORACLE_HOME/hs/admin' 디렉터리 안의 inithsodbc.ora 또는 initdg4odbc.ora 파일을 복사하여 init{DSN}.ora 파일을 생성한다.

```
$ pwd
/youha/oracle/ORA19/app/oracle/product/19.0.0/db_1/hs/admin
```

```
$ ls -al |grep init
-rw-r--r-- 1 oracle dba 489 4월 17 2019 initdg4odbc.ora
```

만약 SID_NAME이 'dg4_tibero'일 경우 initdg4_tibero.ora 파일을 생성한다.

```
```shell
$ cp initdg4odbc.ora initdg4_tibero.ora
```

```
HS_FDS_CONNECT_INFO = dg4_tibero
HS_FDS_SHAREABLE_NAME = /youha/oracle/tibero7/client/lib/libtobdbc.so
```

```
HS_FDS_CONNECT_INFO=dg4_tibero
#HS_FDS_TRACE_LEVEL=DEBUG
Tibero ODBC Driver
HS_FDS_SHAREABLE_NAME=/home/ora11/tibero7/client/lib/libtobdbc.so
iODBC Driver Manager
#HS_FDS_SHAREABLE_NAME=/home/ora11/iodbc/lib/libiodbc.so
#HS_LANGUAGE=AMERICAN_AMERICA.KO16MSWIN949
#
ODBC specific environment variables
#
#set ODBCINI=<full path name of the odbc initialization file>
#set ODBCINI=/home/ora11/odbc.ini
#
Environment variables required for the non-Oracle system
#
#set <envvar>=<value>
#set TB_NLS_LANG=UTF8
#set TBCLI_LOG_LVL=TRACE
```

다음은 설정 항목에 대한 설명이다.

- Oracle Gateway 환경설정 항목

항목	설명
HS_FDS_CONNECT_INFO	ODBC Datasource 이름을 기술하며 연결방식에 따라 다르다. (필수) - Tibero ODBC Driver 직접 연결방식의 경우 : tbdns.tbr에 설정된 Alias명을 설정한다. 만약, .odbc.ini 설정 파일이 존재하거나 tbdns.tbr과 동일한 Alias가 있다면 .odbc.ini 설정 파일을 먼저읽어들이므로 주의한다. - Driver Manager 방식의 경우 : Datasource 이름을 설정한다.
HS_FDS_SHAREABLE_NAME	라이브러리 경로 및 이름 지정, 연결방식에 따라 다르다. (필수) - Tibero ODBC Driver 직접 연결방식의 경우 : Tibero 라이브러리 경로 및 이름을 설정한다. - Driver Manager 방식의 경우 : 해당 Manager Driver 라이브러리 경로 및 이름을 설정한다.
HS_FDS_TRACE_LEVEL	Oracle Gateway의 로그 레벨을 설정한다. '\$ORACLE_HOME/hs/log' 경로에 로그가 남는다. (선택)
HS_LANGUAGE	Gateway의 캐릭터 셋을 설정한다. 미설정일 때 Listener를 기동하면 적용된 NLS_LANG의 환경변수를 따른다. (선택)

## 참고

위에서 언급한 항목들은 가장 많이 사용하는 부분이고 이외에 자세한 설명은 Oracle 홈페이지(<http://www.oracle.com>)를 참고한다.

### ○ Oracle Gateway 환경변수

환경변수	설명
ODBCINI	ODBC 초기 설정 파일 위치를 지정하는 변수이다. 연결방식에 따라 다르다. - Tibero ODBC Driver 직접 연결방식의 경우 : 설정이 필요하지 않다. - Driver Manager 방식의 경우 : 초기 설정 파일 위치를 지정한다.
Tibero 환경 변수	Tibero ODBC Driver 로드일 때 전달하는 환경변수이다. 미설정일 때 Listener를 기동하면 적용된 Tibero의 환경변수를 따른다. 가장 많이 사용하는 환경변수는 TB_NLS_LANG, TBCLI_LOG_LVL이다.

## 6. DB Link 생성 및 확인

- DB Link 생성 방법

```
create database link <DB Link명> connect to <user명> identified by "<password>" using '<Network Alias명>';
```

Link를 생성할 경우 권한 에러가 발생하면 아래와 같은 권한을 부여한다.

```
GRANT CREATE DATABASE LINK TO USER_NAME;
GRANT CREATE PUBLIC DATABASE LINK TO USER_NAME;
```

## 참고

작은따옴표(' '), 큰따옴표(" ") 사용에 주의한다.

단, < user 명>을 큰따옴표(" ")로 감싸는 것은 가능하다.

- DB Link 생성 예

```
$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Jun 15 17:26:24 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> drop database link o2t_link;

Database link dropped.

SQL> create database link o2t_link connect to tiberio identified by tmax using 'tb7';

Database link created.
```

- DB Link 확인 예

DB Link 생성 후 확인을 하는 경우 select 문에 'Table명@< DB Link명 >'으로 설정하면 DB Link를 사용할 수 있다.

```
SQL> select * from dual@o2t_link;
```

```
DUM

X
```

- sqlplus에 접속하여 Tiberio 테이블 조회 및 데이터 삽입 예

Tiberio 에서 test 테이블 생성

```

$ tbsql tibero/tmax@o2t

tbSQL 7

TmaxTibero Corporation Copyright (c) 2020-. All rights reserved.

Connected to vGlizXJv using o2t.

SQL> create table test(c1 number);

Table 'TEST' created.

SQL> insert into test values(10);

1 row inserted.

SQL> commit;

Commit completed.

```

sqlplus 접속하여 Tibero의 test 테이블 조회 및 데이터 삽입

```

$ sqlplus / as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Thu Jun 15 17:34:02 2023
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

```

```

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

```

```
SQL> select * from test@o2t_link;
```

```

 C1

 10

```

```
SQL> insert into test@o2t_link values(20);
```

```
1 row created.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL> select * from test@o2t_link;
```

```

 C1

```

## 4.3. ODBC Manager 설치

본 절에서는 UNIX 계열(Linux 포함)에서 iodbc 설치 및 연동하는 과정을 설명한다.

UNIX 계열의 경우, ODBC Manager가 존재하지 않으므로 iodbc 또는 UNIX ODBC를 별도로 설치해야 한다.

다음은 ODBC Manager를 설치하는 과정에 대한 설명이다.

1. Driver Manager 설치
2. profile 설정
3. tnsnames.ora 설정 (UNIX 계열 참고)
4. ODBC 환경 파일 설정 및 확인
5. Oracle Gateway 환경설정 (init<DSN>.ora)
6. DB Link 생성 및 확인 (UNIX 계열 참고)

각 과정에 대한 상세한 설명은 해당 절의 내용을 참고한다.

### 1. Driver Manager 설치

다음은 Driver Manager를 설치하는 과정에 대한 설명이다.

#### 1. 다운로드

<http://iodbc.org>에서 다운로드 후 설치를 원하는 서버에 업로드한다.

다음은 libiodbc-3.52.7.tar.gz 소스파일을 다운받아 서버에 설치한 결과이다.

만약, tar 옵션의 xvzf가 적용되지 않을 경우 먼저 gunzip으로 압축을 해제한 후 tar -xvf 옵션으로 해제한다.

```
$ tar -xvzf libiodbc-3.52.7.tar.gz

$ ls -al libiodbc-3.52.7
합계 868
drwxr-xr-x 14 oracle dba 4096 9월 10 2009 .
drwxrwxrwx 16 oracle dba 4096 6월 15 17:53 ..
-rw-r--r-- 1 oracle dba 1038 7월 27 2005 AUTHORS
-rw-r--r-- 1 oracle dba 735 8월 30 2009 COPYING
-rw-r--r-- 1 oracle dba 66851 9월 10 2009 ChangeLog
-rw-r--r-- 1 oracle dba 1480 9월 10 2009 IAFA-PACKAGE
-rw-r--r-- 1 oracle dba 9236 2월 21 2004 INSTALL
-rw-r--r-- 1 oracle dba 735 8월 30 2009 LICENSE
-rw-r--r-- 1 oracle dba 1520 8월 30 2009 LICENSE.BSD
-rw-r--r-- 1 oracle dba 25277 1월 21 2006 LICENSE.LGPL
-rw-r--r-- 1 oracle dba 10099 9월 10 2009 Makefile.am
-rw-r--r-- 1 oracle dba 32869 9월 10 2009 Makefile.in
-rw-r--r-- 1 oracle dba 12592 9월 10 2009 NEWS
-rwxr-xr-x 1 oracle dba 8828 9월 9 2009 PORT.OpenLink
-rw-r--r-- 1 oracle dba 11328 8월 30 2009 README
-rw-r--r-- 1 oracle dba 3193 9월 1 2009 README.CVS
-rw-r--r-- 1 oracle dba 2653 9월 10 2009 README.MACOSX
-rw-r--r-- 1 oracle dba 3949 8월 30 2009 acinclude.m4
```

```

-rw-r--r-- 1 oracle dba 37029 9월 10 2009 aclocal.m4
drwxr-xr-x 2 oracle dba 4096 9월 10 2009 admin
drwxr-xr-x 2 oracle dba 4096 9월 10 2009 bin
-rwxr-xr-x 1 oracle dba 5963 8월 30 2009 bootstrap.sh
-rwxr-xr-x 1 oracle dba 512594 9월 10 2009 configure
-rw-r--r-- 1 oracle dba 22978 9월 10 2009 configure.in
drwxr-xr-x 2 oracle dba 4096 9월 10 2009 debian
drwxr-xr-x 4 oracle dba 4096 9월 10 2009 drvproxy
drwxr-xr-x 2 oracle dba 4096 9월 10 2009 etc
drwxr-xr-x 2 oracle dba 4096 9월 10 2009 include
drwxr-xr-x 3 oracle dba 4096 9월 10 2009 iodbc
drwxr-xr-x 4 oracle dba 4096 9월 10 2009 iodbcadm
drwxr-xr-x 2 oracle dba 4096 9월 10 2009 iodbcinst
drwxr-xr-x 11 oracle dba 4096 9월 10 2009 mac
drwxr-xr-x 2 oracle dba 4096 9월 10 2009 man
drwxr-xr-x 2 oracle dba 4096 9월 10 2009 samples

```

## 2. iodbc 설치

tar.gz 파일을 가지고 설치를 진행하면 아래와 같은 기본적인 순서를 따른다.

설치 전 사전환경 점검에서 prefix를 설정하지 않으면 기본적으로 /usr/local 아래에 설치되기 때문에 특정 디렉터리를 지정하여 설치하고, 컴파일 및 설치 작업을 다시 수행할 경우 make clean 이후에 진행한다.

a) '\$HOME/iodbc' 디렉터리에 설치 전 사전환경 점검을 한다.

```

$ mkdir iodbc
$ cd libiodbc-3.52.7/

./configure --prefix=$HOME/iodbc --disable-gui

```

b) 점검한 결과 문제가 없으면 컴파일을 한다.

```
make
```

c) 컴파일된 파일을 설치한다.

```
make install
```

## 3. 설치 확인

설치된 서버에 원하는 bit로 설치가 되었는지 확인한다. 특정 OS에 따라 64bit 서버에 32bit로 설치되는 경우가 있으므로 file 명령어를 이용하여 확인이 필요하다.

만약, ODBC Manager가 64bit로 설치됐다면 내부적으로 사용하는 Tiberio ODBC 역시 64bit여야 하며 재설치가 필요할 경우 make clean 이후에 진행한다.

```
$ cd $HOME/iodbc/bin

$ file iodbctest
iodbctest: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.32,
BuildID[sha1]=9074c37e82e67ad88dfcf2534021d0b3d45edda4, not stripped
```

## 참고

다음은 OS별로 iodbc 64bit 설치와 설치 확인 방법에 대한 설명이다.

### ■ Linux

#### 1. 사전 환경 점검

```
export CFLAGS=-m64
./configure --prefix=$HOME/iodbc --disable-gui
```

#### 2. 컴파일

```
./make
```

#### 3. 설치

```
./make install
```

#### 4. 설치 확인

```
$ file iodbctest
iodbctest: ELF 64-bit LSB executable, AMD x86-64, version 1(SYSV),
for GNU/Linux 2.6.9, dynamically linked(uses shared libs), not stripped
```

### \* AIX

#### 1. 사전 환경 점검

```
...
export CFLAGS=-maix64
export LDFLAGS="-maix64 -brtl"
./configure --prefix=$HOME/iodbc --disable-gui
...
```

#### 2. 컴파일

```
...
```

```

export OBJECT_MODE=64
./make
...

3. 설치
...
./make install
...

4. 설치 확인
...
$ file iodbctest
iodbctest: 64-bit XCOFF executable or object module not stripped
...

```

## 2. profile 설정

profile 내에 아래와 같은 내용을 추가한다. IODBC\_HOME의 경우 iodbc를 설치한 위치로 설정한다.

```

iodbc setting
export IODBC_HOME=$HOME/iodbc
export LD_LIBRARY_PATH=$IODBC_HOME/lib:$LD_LIBRARY_PATH
export PATH=$IODBC_HOME/bin:$PATH

```

### 주의

OS에 맞게 환경변수 LD\_LIBRARY\_PATH(Linux), LIBPATH(AIX)를 설정한다.

## 3. ODBC 환경 파일 설정 및 확인

ODBC Driver Manager의 환경 파일에 Tiberio ODBC Driver를 등록하는 방법이다. 연결 테스트 전에 Tiberio 클라이언트 또는 서버의 설치 및 관련 환경설정이 되어야 한다.

### 1. 환경 파일 위치 및 이름 설정

\$HOME/.odbc.ini(개인 설정) 또는 /etc/odbc.ini(공통 설정)으로 설정 가능하다.

우선순위는 \$HOME/.odbc.ini(개인 설정) 이 높다.

- ODBC 환경 파일 설정 방법

```

[ODBC Data Sources]
<ODBC Data Sources> = Tibero7 ODBC driver
[ODBC]
Trace = 1
TraceFile = /yuha/oracle/iodbc/tmp/odbc.trace
[<ODBC Data Sources 세부설정>]
Driver = <Tibero ODBC Driver 파일>
Description = Tibero5 ODBC Datasource
SID = <tbdns.tbr 파일에 설정한 alias 정보>
User = dbtech
Password = dbtech

```

항목	설명
< ODBC Data Sources >	Datasource 이름으로 Oracle Gateway 설정 파일에 해당 내용이 들어간다.
< ODBC Data Sources 세부설정>	ODBC Data Sources에서 설정한 이름으로 대소문자까지 일치해야 한다.
Driver	ODBC Manager에서 로드하는 Tibero ODBC Driver 파일이다. 해당 파일 존재 여부 및 권한에 대해서 확인이 필요하다.
SID	Tibero 클라이언트 또는 서버의 tbdns.tbr 파일에 설정한 Alias 정보이다.
User	사용자를 의미한다. 테스트 및 링크 생성의 경우 별도로 사용자를 가져가므로 크게 의미는 없다.
Password	사용자 패스워드를 의미한다. 테스트 및 링크 생성의 경우 별도로 사용자를 가져가므로 크게 의미는 없다.

■ ODBC 환경 파일 설정 예

```

$ cat .odbc.ini
[ODBC Data Sources]
tibero7 = Tibero7 ODBC driver

[ODBC]
Trace = 1
TraceFile = /yuha/oracle/iodbc/tmp/odbc.trace

[Tibero7]
Server = 192.168.17.24
Port = 5555
Database = tibero
Driver = /yuha/oracle/tibero7/client/lib/libtbodbc.so
Description = Tibero7 ODBC Datasource
SID = tibero
User = tibero
Password = tmax

```

## 2. 연결 테스트

'\$IODBC\_HOME/bin' 폴더에 있는 iodbctest를 이용하여 연결 테스트를 수행한다.  
테스트에 문제가 발생하는 경우 ODBC 환경 파일 이름 및 위치와 설정을 확인한다.  
다음은 iodbctest를 이용한 연결 테스트 설정 방법이다.

```
iodbctest "DSN=<dsn>;UID=<user>;PWD=<pwd>"
```

항목	설명
DSN	ODBC 환경 파일에서 설정한 ODBC Datasources 이름이다.
UID, PWD	테스트 할 접속 계정 및 패스워드이다.

다음은 iodbctest를 이용한 연결 테스트 사용 예이다.

```
$ iodbctest "DSN=Tibero7;UID=tibero;PWD=tmax"
iODBC Demonstration program
This program shows an interactive SQL processor
Driver Manager: 03.52.0709.0909
Driver: 07.00.0218 (libtbodbc.so)

SQL>select * from test;

C1

10
20

result set 1 returned 2 rows.
```

## 4. Oracle Gateway 환경설정 (init{DSN}.ora)

HS\_FDS\_CONNECT\_INFO, HS\_FDS\_SHAREABLE\_NAME에 ODBC Driver 관련 설정을 한다.

```
HS_FDS_CONNECT_INFO=tibero
#HS_FDS_TRACE_LEVEL=DEBUG
Tibero ODBC Driver
HS_FDS_SHAREABLE_NAME=/home/ora11/tibero5/client/lib/libtbodbc.so

iODBC Driver Manager
#HS_FDS_SHAREABLE_NAME=/home/ora11/iodbc/lib/libiodbc.so
#HS_LANGUAGE=AMERICAN_AMERICA.KO16MSWIN949
#
ODBC specific environment variables
#
#set ODBCINI=<full path name of the odbc initialization file>
#set ODBCINI=/home/ora11/.odbc.ini
#
Environment variables required for the non-Oracle system
#
#set <envvar>=<value>
```

```
#set TB_NLS_LANG=UTF8
#set TBCLI_LOG_LVL=TRACE
```

```
HS_FDS_CONNECT_INFO=Tibero7

iODBC Driver Manager
HS_FDS_SHAREABLE_NAME=/yuha/oracle/iodbc/lib/libiodbc.so
```

## 4.4. 문제 해결

Oracle to Tibero DB Link 설정 및 사용할 때 자주 발생하는 문제점에 대해서 설명한다.

### 4.4.1. DB Link 설정 문제

#### Oracle 캐릭터 셋이 AL32UTF8일 경우

DB Link Object를 생성하는 Oracle 서버의 캐릭터 셋이 AL32UTF8일 때 발생하는 문제이다. Gateway 서버가 UNIX 계열일 때 발생하며 Windows 계열은 해당하지 않는다.

다음은 해당 문제의 현상과 해결방안에 대한 설명이다.

- 현상

Oracle 서버에서 아래와 같이 에러가 발생한다.

```
SQL> select name, value$ from sys.props$ where name like 'NLS_CHAR%';
```

```
NAME
```

```
VALUE$
```

```
NLS_CHARACTERSET
```

```
AL32UTF8
```

```
SQL> conn scott/tiger
```

```
Connected.
```

```
SQL> select * from dual@tlink;
```

```
select * from dual@tlink
```

```
*
```

```
ERROR at line 1:
```

```
ORA-28511: lost RPC connection to heterogeneous remote agent using
```

```
SID=ORA-28511: lost RPC connection to heterogeneous
```

```
remote agent using
```

```
SID=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.70.185)(PORT=1522))
```

```
(CONNECT_DATA=(SID=tibero)))
```

```
ORA-02063: preceding line from TLINK
```

```
Process ID: 14824
```

```
Session ID: 8 Serial number: 14464
```

- 해결방안

Oracle Gateway 파일인 init{DSN}.ora 파일에 HS\_LANGUAGE 설정을 통해 해결한다.

다음 설정값은 사이트 환경에 따라 다를 수 있다.

```
HS_LANGUAGE=AMERICAN_AMERICA.KO16MSWIN949
```

위와 같이 설정한 후에 재접속 및 테스트를 수행하면 정상적으로 수행된다.

```
$ sqlplus scott/tiger
SQL*Plus: Release 11.2.0.1.0 Production on Tue Jun 4 14:23:11 2013
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> select * from dual@tlink;
DUM

X
```

---

## 참고

Oracle을 AL32UTF8로 생성한 상태에서 Oracle generic ODBC Gateway를 이용해 다른 데이터베이스와 DB Link 구성을 하여 접속할 경우 SQLDriverConnectW로 주는 인자가 wchar\_t 형태(UTF-16형태)가 아니라 UTF-8 형태로 들어온다.

---