

**Connecting  
Oracle10g  
on  
Linux x86  
to  
SQL Server 2K/2K5**

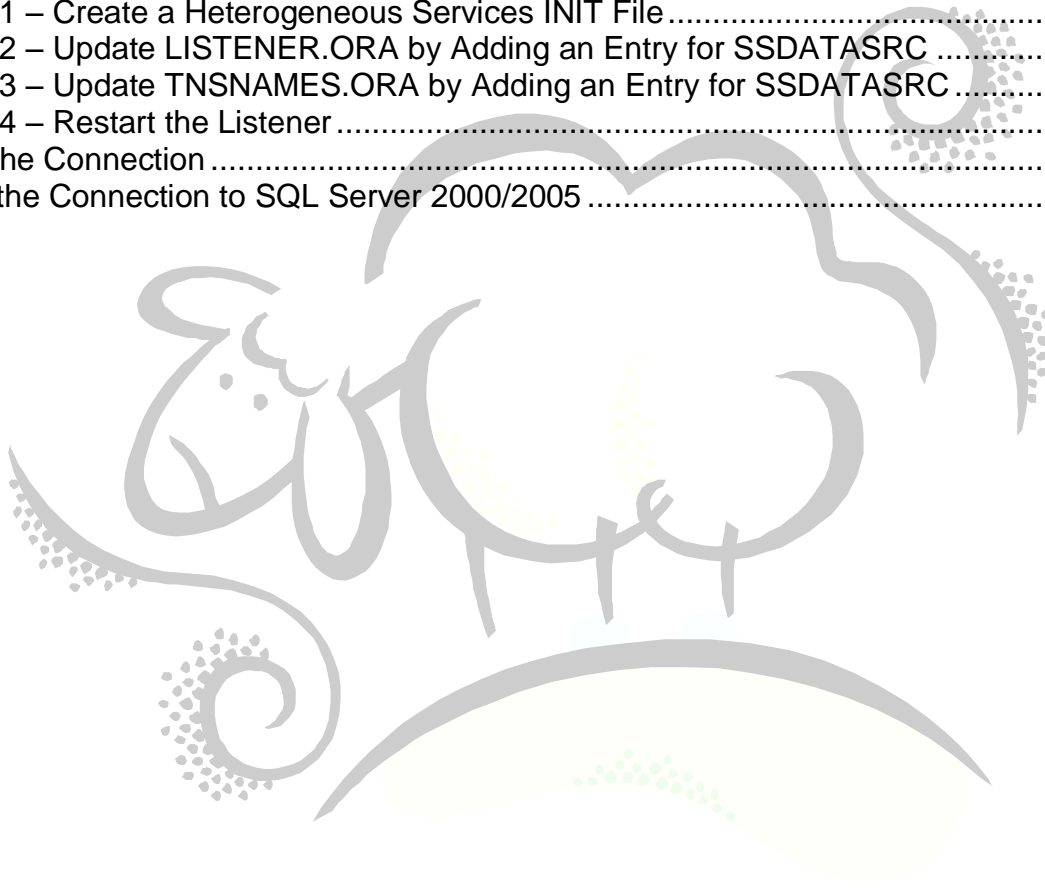
This work may be reproduced and redistributed, in whole or in part, without alteration and without prior written permission, provided all copies contain the following statement:

Copyright ©2011 sheepsqueezers.com. This work is reproduced and distributed with the permission of the copyright holder.

This presentation as well as other presentations and documents found on the sheepsqueezers.com website may contain quoted material from outside sources such as books, articles and websites. It is our intention to diligently reference all outside sources. Occasionally, though, a reference may be missed. No copyright infringement whatsoever is intended, and all outside source materials are copyright of their respective author(s).

## Table of Contents

Introduction .....	4
Pre-Requisites .....	5
Pre-Step 1 – Ensure that unixODBC is Installed .....	5
Pre-Step 2 – Ensure that FreeTDS is Installed .....	5
Pre-Step 3 – Register FreeTDS with the unixODBC Driver Manager .....	6
Pre-Step 4 – Create an ODBC Data Source Name .....	6
Pre-Step 5 – Chmod of odbc.ini and odbcinst.ini .....	7
Pre-Step 6 – Create a SQL Server 2000/2005 User ID/Password to Desired Database .....	7
Pre-Step 7 – Test Your ODBC Data Source .....	7
Pre-Step 8 – Oracle Net Configuration Assistant run TCP and IPC Protocols .....	8
Configuring Oracle10g for Generic Connectivity .....	9
Step 1 – Create a Heterogeneous Services INIT File .....	9
Step 2 – Update LISTENER.ORA by Adding an Entry for SSDATASRC .....	9
Step 3 – Update TNSNAMES.ORA by Adding an Entry for SSDATASRC .....	10
Step 4 – Restart the Listener .....	11
Testing the Connection .....	12
Test the Connection to SQL Server 2000/2005 .....	12



## Introduction

This document details how to set up your Oracle10g database installation on a 32-bit Intel x86 machine running Linux in order to pull data from and push data to a Microsoft SQL Server 2000/2005 database.

While this document focuses on a server running a Linux distribution, the same procedure outlined below can most likely be performed on a server running any flavor of Unix.

While Microsoft Windows users have the choice between ODBC and OLE DB to communicate with their databases, Linux/Unix users currently can only use ODBC, although this may change in the future. This document focuses on ODBC database connections only.

Oracle10g Heterogeneous Services agent is the process through which an Oracle server connects to a non-Oracle system such as SQL Server. There are two types of Heterogeneous Services agents: *Oracle Transparent Gateways* and *Generic Connectivity Agents*. With Oracle Transparent Gateways, Oracle provides gateways to access several commercially available non-Oracle systems, but not all operating systems and databases are represented. On the other hand, Oracle's Generic Connectivity agents use ODBC and OLEDB to communicate with the desired database. As long as there is an available driver for your operating system, then you can communicate with the desired database. In this paper, we will concentrate on Oracle's Generic Connectivity via ODBC.

Note that Oracle10g is the first release of the Oracle database to allow Heterogeneous Services to a non-Oracle database on a Linux operating system.

**Note that the connection method described in this document is not supported on x86-64/64 bit Linux platforms at this time. The file HSODBC, while it exists, contains zero bytes.**

**NOTE: MAKE SURE TO TEST THIS ON A TEST MACHINE BEFORE ATTEMPTING ON PRODUCTION SERVERS!!! YOU HAVE BEEN WARNED!!**

## Pre-Requisites

This section outlines what software needs to be installed on your Linux server before you begin any modifications to your Oracle10g configuration.

### Pre-Step 1 – Ensure that unixODBC is Installed

You must have the ODBC Driver Manager software, called unixODBC, is installed on your server as well as its associated development software. Download the following RPMs from the web:

```
unixODBC-devel-2.2.11-19.i586.rpm
unixODBC-2.2.11-19.i586.rpm
```

You can install these two packages by using the following commands at the root command prompt:

```
rpm -ivh unixODBC-devel-2.2.11-19.i586.rpm
rpm -ivh unixODBC-2.2.11-19.i586.rpm
```

### Pre-Step 2 – Ensure that FreeTDS is Installed

You must also have an ODBC Driver available to facilitate the communication between your Oracle10g database and the SQL Server 2000/2005 database. Download the following file to the /tmp directory from the website [www.freetds.org](http://www.freetds.org):

```
freetds-stable.tgz
```

and uncompress and untar it using the following command:

```
tar -xvzf freetds-stable.tgz
```

Create the following environment variables as root:

```
export SYBASE=/usr/local/freetds
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/$SYBASE/lib
```

Once completed, you'll have a subdirectory called freetds-0.64 installed in /tmp. Change to that directory and issue the following commands as root to compile FreeTDS and install it into your system:

```
./configure --with-tdsver=8.0 --with-unixodbc=/usr/lib
make
make install
```

### Pre-Step 3 – Register FreeTDS with the unixODBC Driver Manager

This step allows the ODBC Driver Manager software to register the existence of the ODBC Driver FreeTDS. As root, open up the text file `/etc/tds.driver.template` and enter in the following text:

```
[FreeTDS]
Description = v0.64 with protocol v8.0
Driver = /usr/local/lib/libtdsodbc.so
```

Next, at the root command prompt, type the following command and hit the Enter key:

```
odbcinst -i -d -f /etc/tds.driver.template
```

Next, check the file `/etc/unixODBC/odbcinst.ini` to ensure that it looks something like this:

```
[FreeTDS]
Description = v0.64 with protocol v8.0
Driver      = /usr/local/lib/libtdsodbc.so
UsageCount  = 1
```

### Pre-Step 4 – Create an ODBC Data Source Name

Next, we will create the ODBC Data Source Name which will be associated with our SQL Server 2000/2005 database. Open up the file `/etc/unixODBC/odbc.ini` and enter in the following text. Note that you will have to fill in your own values for the italicized items:

```
[DATA_SOURCE_NAME]
Driver = FreeTDS
Description = Enter in a description here
Trace = No
Server = SQL Server IP Address
Port = SQL Server Port Number
Database = Database Name
```

Here is an example:

```
[SSDATASRC]
Driver = FreeTDS
Description = SQL Server 2000 on SSDEV001 to Database SSDATABASE
Trace = No
Server = 198.168.23.1
Port = 1433
Database = SSDATABASE
```

The text [SSDATASRC] is the ODBC Data Source Name and can be changed to whatever you like. The Driver must remain FreeTDS. The Description can be anything you'd like. Here, Trace is set to No. The Server entry is the IP Address of the SQL Server 2000/2005 machine

you want to connect to. The Port entry is the SQL Server 2000/2005 port and is usually 1433. The Database entry is the name of the database you want to interact with.

### Pre-Step 5 – Chmod of `odbc.ini` and `odbcinst.ini`

Before you continue on, as root, please change both `odbc.ini` and `odbcinst.ini` to read/write for user, group and world:

```
chmod 666 /etc/unixODBC/odbc.ini
chmod 666 /etc/unixODBC/odbcinst.ini
```

You probably want to throttle back on these when you have completed this document.

### Pre-Step 6 – Create a SQL Server 2000/2005 User ID/Password to Desired Database

Since there is no Windows authentication from a Linux/Unix machine, you'll have to create a SQL Server 2000/2005 database username and password to the desired database with SQL Server Authentication. On SQL Server 2000, you use the SQL Server Enterprise Manager to add a new user to the desired database. Ensure that you select "new" from the Login Name drop-down box so that you can specify SQL Server Authentication and a password. Ensure that you have selected the desired database from the drop-down box labeled Database. For SQL Server 2005, you must create a new user via the operating system and then add that user to the desired database.

For our test database SSDATABASE on SQL Server 2000, we created a user named ssuser1 with password ssuser1 who will be authenticated by SQL Server and not the operating system.

### Pre-Step 7 – Test Your ODBC Data Source

UnixODBC provides a tool called `isql` which allows you to connect to your SQL Server 2000/2005 database. At the root command prompt, type in the following and hit the enter key:

```
isql DATA-SOURCE-NAME USERNAME PASSWORD
```

Using our example,

```
isql SSDATASRC ssuser1 ssuser1
```

If all went well, you should receive the following prompt:

```
excelsior:~ # isql SSDATASRC ssuser1 ssuser1
+-----+
| Connected!                                     |
|                                               |
| sql-statement                                |
| help [tablename]                             |
| quit                                         |
+-----+
```

```

|
+-----+
SQL>

```

You can issue a SQL command at the SQL> prompt as shown below with the results of the query:

```

SQL> SELECT COUNT(*) FROM MY_DATABASE_TABLE;
+-----+
|          |
+-----+
| 1276     |
+-----+
SQLRowCount returns 1
1 rows fetched
SQL>

```

If all went well in this step, it indicates that your Linux/Unix machine can communicate successfully with your SQL Server 2000/2005 database.

### Pre-Step 8 – Oracle Net Configuration Assistant run TCP and IPC Protocols

When setting up the listener from within the `netca` GUI (logged on as the oracle user), ensure that you have requested not only the TCP protocol, but the IPC protocol as well. When asked for a KEY, enter in the value `EXTPROC0`. Your listener.ora file should look something like this:

```

# listener.ora Network Configuration File:
# /usr/local/app/oracle/product/10.2.0/network/admin/listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /usr/local/app/oracle/product/10.2.0)
      (PROGRAM = extproc)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC0))
      (ADDRESS = (PROTOCOL = TCP) (HOST = excelsior) (PORT = 1521))
    )
  )

```



## Configuring Oracle10g for Generic Connectivity

In this section, we outline what steps are needed from an Oracle10g database point-of-view.

### Step 1 – Create a Heterogeneous Services INIT File

Recall that in Pre-Step 4 above, we created a unixODBC Data Source Name called SSDATASRC. In order for Oracle10g to find this ODBC Data Source and use it properly, you have to create an Oracle10g Heterogeneous Services parameter file located in \$ORACLE\_HOME/hs/admin called `initSSDATASRC.ora`. Note that the unixODBC Data Source Name is embedded in this file name. This file looks like this:

```
# This is a sample agent init file that contains the HS parameters
# that are needed for an ODBC Agent.

#
# HS init parameters
#
HS_FDS_CONNECT_INFO = SSDATASRC
HS_FDS_TRACE_LEVEL = 4
HS_FDS_SHAREABLE_NAME = /usr/lib/libodbc.so

#
# ODBC specific environment variables
#
set ODBCINI=/etc/unixODBC/odbc.ini
```

Note the parameter `HS_FDS_CONNECT_INFO` is set to `SSDATASRC`. Note also that `HS_FDS_SHAREABLE_NAME` points to the location and filename of the `libodbc.so` shared object.

If you wanted to communicate with several SQL Server 2000/2005 databases in your organization, you would create an Heterogeneous Services parameter file for each database/table combination. Here, we are interacting with the table `SSDATABASE` on the SQL Server 2000 server `SSDEV001` on `198.168.23.1` (all of this is defined in the `odbc.ini` file).

### Step 2 – Update LISTENER.ORA by Adding an Entry for SSDATASRC

Recall above in Pre-Step 8, we showed an example `listener.ora` file. First, stop the listener now by using the `lsnrctl stop LISTENER` command. You now have to update this file by hand by adding the following section to the `SID_LIST` section

```
(SID_DESC=
  (PROGRAM=/usr/local/app/oracle/product/10.2.0/bin/hsodbc)
  (SID_NAME=SSDATASRC)
  (ORACLE_HOME=/usr/local/app/oracle/product/10.2.0)
  (ENV=LD_LIBRARY_PATH=/usr/local/app/oracle/product/10.2.0/lib:/usr/local/lib)
)
```

so that the entire file looks like this now:

```

# listener.ora Network Configuration File:
#   /usr/local/app/oracle/product/10.2.0/network/admin/listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /usr/local/app/oracle/product/10.2.0)
      (PROGRAM = extproc)
    )
    (SID_DESC=
      (PROGRAM=/usr/local/app/oracle/product/10.2.0/bin/hsodbc)
      (SID_NAME=SSDATASRC)
      (ORACLE_HOME=/usr/local/app/oracle/product/10.2.0)
      (ENV=LD_LIBRARY_PATH=/usr/local/app/oracle/product/10.2.0/lib:/usr/local/lib)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC0))
      (ADDRESS = (PROTOCOL = TCP) (HOST = excelsior) (PORT = 1521))
    )
  )

```

### Step 3 – Update TNSNAMES.ORA by Adding an Entry for SSDATASRC

Similar the Step 2 above, we add an entry to `tnsnames.ora` like this:

```

SSDATASRC =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = tcp) (HOST = 198.168.23.4) (PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = SSDATASRC)
    )
    (HS = OK)
  )

```

Note that the HOST IP Address is still the IP address of the Oracle10g database as well as its PORT number. The only place where the SQL Server 2000/2005 database is mentioned is in the `odbc.ini` file!

Also, note the entry `HS=OK` in the `tnsnames.ora` entry above. This indicates that `SSDATASRC` is a Heterogeneous Service.

## Step 4 – Restart the Listener

Using the `lsnrctl start LISTENER` command, start the listener up. You should see something similar to this output to your screen:

```
oracle@excelsior:/usr/local/app/oracle/product/10.2.0/network/admin> lsnrctl
start LISTENER
```

```
LSNRCTL for Linux: Version 10.2.0.1.0 - Production on 04-NOV-2006 20:51:21
```

```
Copyright (c) 1991, 2005, Oracle. All rights reserved.
```

```
Starting /usr/local/app/oracle/product/10.2.0/bin/tnslsnr: please wait...
```

```
TNSLSNR for Linux: Version 10.2.0.1.0 - Production
```

```
System parameter file is
```

```
/usr/local/app/oracle/product/10.2.0/network/admin/listener.ora
```

```
Log messages written to
```

```
/usr/local/app/oracle/product/10.2.0/network/log/listener.log
```

```
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC)))
```

```
Listening on:
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=excelsior) (PORT=1521)))
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=EXTPROC)))
```

```
STATUS of the LISTENER
```

```
-----
```

Alias	LISTENER
Version	TNSLSNR for Linux: Version 10.2.0.1.0 - Production
Start Date	04-NOV-2006 20:51:21
Uptime	0 days 0 hr. 0 min. 0 sec
Trace Level	off
Security	ON: Local OS Authentication
SNMP	OFF

```
Listener Parameter File
```

```
/usr/local/app/oracle/product/10.2.0/network/admin/listener.ora
```

```
Listener Log File
```

```
/usr/local/app/oracle/product/10.2.0/network/log/listener.log
```

```
Listening Endpoints Summary...
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC)))
```

```
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=excelsior) (PORT=1521)))
```

```
Services Summary...
```

```
Service "PLSExtProc" has 1 instance(s).
```

```
Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this
service...
```

```
Service "SSDATASRC" has 1 instance(s).
```

```
Instance "SSDATASRC", status UNKNOWN, has 1 handler(s) for this service...
```

```
The command completed successfully
```

## Testing the Connection

### Test the Connection to SQL Server 2000/2005

Next, log into an Oracle10g database schema that has been given CREATE DATABASE LINK privileges. Create a database link to the ODBC Data Source Name SSDATASRC like this:

```
SQL> create database link ssdb connect to "ssuser1" identified by
"ssuser1" using 'SSDATASRC';
```

Database link created.

Note that double-quotes around the username and password. Note that we are providing the TNSNAMES.ORA entry SSDATASRC as the value for the USING statement in the CREATE DATABASE LINK command above.

Next, let's try a query:

```
SQL> select count(*) from MY_DATABASE_TABLE@ssdb;
```

```
  COUNT(*)
-----
      1276
```

Take note of the following example:

```
SQL> select count(distinct COL1) from MY_DATABASE_TABLE@ssdb;
select count(distinct COL1) from MY_DATABASE_TABLE@ssdb
*
ERROR at line 1:
ORA-00904: "COL1": invalid identifier
```

In this case, we have to surround the column name COL1 with double-quotes in order for this query to work properly:

```
SQL> select count(distinct "COL1") as CNT from MY_DATABASE_TABLE@ssdb;
```

```
  CNT
----
  1276
```

Assuming you have been given the appropriate rights, you can also insert into this table:

```
INSERT INTO MY_DATABASE_TABLE@ssdb VALUES('FLUFF','FLUFFY FLUFFY');
```

If you want to select for your new value, you would have to query the database like this:

```
SQL> select * from MY_DATABASE_TABLE@ssdb where "COL1"='FLUFF';
```

```
COL1      COL2
-----
FLUFF      FLUFFY FLUFFY
```

**Support sheepsqueezers.com**

If you found this information helpful, please consider supporting [sheepsqueezers.com](http://sheepsqueezers.com). There are several ways to support our site:

- ☐ Buy me a cup of coffee by clicking on the following link and donate to my PayPal account: [Buy Me A Cup Of Coffee?](#).
- ☐ Visit my Amazon.com Wish list at the following link and purchase an item:  
<http://amzn.com/w/3OBK1K4EIWIR6>

Please let me know if this document was useful by e-mailing me at [comments@sheepsqueezers.com](mailto:comments@sheepsqueezers.com).

**Support sheepsqueezers.com**

If you found this information helpful, please consider supporting [sheepsqueezers.com](http://sheepsqueezers.com). There are several ways to support our site:

- ☐ Buy me a cup of coffee by clicking on the following link and donate to my PayPal account: [Buy Me A Cup Of Coffee?](#).
- ☐ Visit my Amazon.com Wish list at the following link and purchase an item:  
<http://amzn.com/w/3OBK1K4EIWIR6>

Please let me know if this document was useful by e-mailing me at [comments@sheepsqueezers.com](mailto:comments@sheepsqueezers.com).

- Support sheepsqueezers.com**
- If you found this information helpful, please consider supporting [sheepsqueezers.com](http://sheepsqueezers.com). There are several ways to support our site:
- ☐ Buy me a cup of coffee by clicking on the following link and donate to my PayPal account: [Buy Me A Cup Of Coffee?](#).
  - ☐ Visit my Amazon.com Wish list at the following link and purchase an item:  
<http://amzn.com/w/3OBK1K4EIWIR6>
- Please let me know if this document was useful by e-mailing me at [comments@sheepsqueezers.com](mailto:comments@sheepsqueezers.com).

**Support sheepsqueezers.com**

If you found this information helpful, please consider supporting [sheepsqueezers.com](http://sheepsqueezers.com). There are several ways to support our site:

- ☐ Buy me a cup of coffee by clicking on the following link and donate to my PayPal account: [Buy Me A Cup Of Coffee?](#).
- ☐ Visit my Amazon.com Wish list at the following link and purchase an item:  
<http://amzn.com/w/3OBK1K4EIWIR6>

Please let me know if this document was useful by e-mailing me at [comments@sheepsqueezers.com](mailto:comments@sheepsqueezers.com).