Using DataDirect Connect® *for* ODBC with Oracle Heterogeneous Services

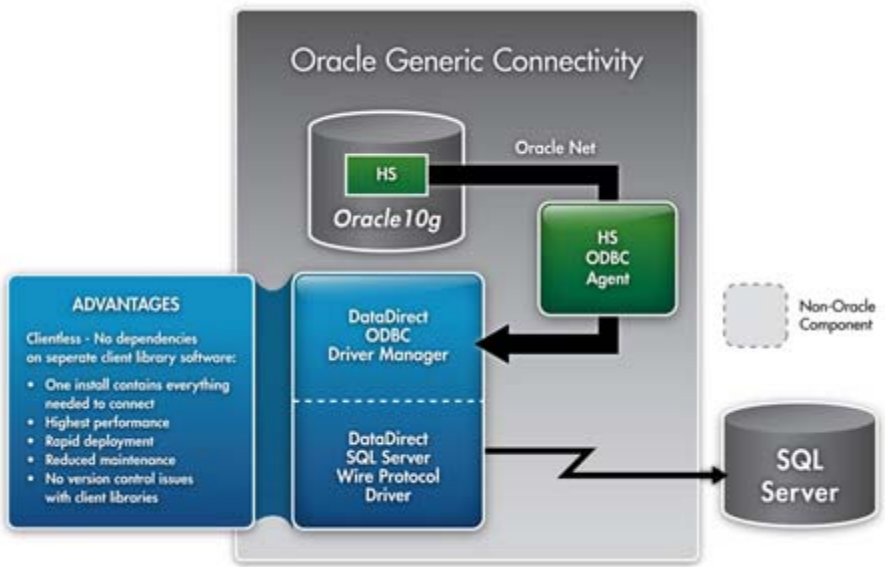
Introduction

Heterogeneous Services and Generic Connectivity provide Oracle customers the ability to access and integrate non-Oracle data sources, providing a wide degree of flexibility in a multi-database environment.

Companies who wish to use Generic Connectivity to consolidate and integrate data with Oracle require optimal connectivity to ensure the best performance. DataDirect Connect *for* ODBC delivers the most scalable and best performing connectivity available for Oracle Heterogeneous Services.

This article explains how to use DataDirect Connect *for* ODBC and the DataDirect OpenAccess™ ODBC driver with Oracle Heterogeneous Services.

Select from the topics below to learn more:



DataDirect Connect *for* ODBC in an Oracle Heterogeneous Services Environment

DataDirect TECHNOLOGIES

Generic Connectivity Architecture

Generic Connectivity is implemented by using a Heterogeneous Services ODBC agent. An ODBC agent is included as part of your Oracle system. Be sure to use the agent shipped with your particular Oracle system and installed in the same $ORACLE_HOME.

To access the non-Oracle data store using Generic Connectivity, the agent works with an ODBC driver. The ODBC driver that you use must be on the same platform as the ODBC agent. The non-Oracle data stores can reside on the same machine as the Oracle database or a different machine.

Installation Steps

This example shows the configuration of Generic Connectivity on a SUN Solaris system using Oracle 10g and hsodbc; DataDirect's 5.3 ODBC driver to connect to Microsoft SQL Server; and DataDirect's OpenAccess 6.0 ODBC driver to connect to a test database. Please check your Oracle documentation for specific version and platform support.

Note: Starting with Oracle 11g, the HS executable name is now called DG4ODBC. If you're using a 64-bit version of Oracle you must use a 64-bit ODBC driver. If you're using a 32-bit version of Oracle, you must use a 32-bit ODBC driver. Please refer to DataDirect KB doc#2466288PG for the supportability matrix.

1) Install the data dictionary tables and views for Heterogeneous Services.

Using the server manager or sqlplus logged on as sys, run caths.sql. For example using the server manager you can use the following example:

```
SQL> connect internal
SQL> @<ORACLE_HOME>/rdbms/admin/caths.sql;
```

This script is located in $ORACLE_HOME/rdbms/admin

2) Install the DataDirect Connect *for* ODBC Driver (or DataDirect OpenAccess SDK ODBC driver).

Some non-Oracle data stores will require that particular database's client library components to be installed. If the database is DB2, Sybase, SQL Server or Informix you should use the DataDirect Connect *for* ODBC Wire Protocol driver for the particular database you are trying to access. These drivers do not require any additional components to be installed to connect to the database.

3) Configure your odbc data source in the odbc.ini file. The example below is a data source to connect to Microsoft SQL Server 2000.

Note: Annotations to following examples file begin with the ◄ symbol and should not be included in the actual file.

```
[MS_SQLServer2000]
Driver=/opt/odbc32v53/lib/ivmsss23.so ◄ Configured during ODBC driver installation
Description=SQL Server
Database=dbname                       ◄ Name of target database
Address=120.2.200.176,1433            ◄ IP address and port of target database
Quoteld=No
AnsiNPW=No
```

The example below is a data source to connect to a Test DataDirect OpenAccess database.

```
[Test_MyDB]
Driver=/opt/oaodbc60/lib/ivoa22.so     ◄ Configured during ODBC driver installation
Description=DataDirect OpenAccess SDK 6.0
Host=<OA Server host>
Port=19996
ServerDataSource=memory
```

4) Make sure the following entries are in the tnsnames.ora and listener.ora.

```
TNSNAMES.ORA
HSALIAS=           ◄ This name can be anything you want it to be
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=hostname)(PORT=1521))
    (CONNECT_DATA=(sid=hsMydb))        ◄ Needs to match the SID in listener.ora.
    (HS=ok)                            ◄ HS clause goes in the description
    )
LISTENER.ORA
LISTENER =
  (DESCRIPTION_LIST=
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = tcp)(HOSTt = unixhost)(PORT = 1521))
      )
    )

SID_LIST_LISTENER=
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=hsMydb)                      ◄ Match the SID in tnsnames.ora.
      (ORACLE_HOME=/db/oracle/product/10g)   ◄ Appropriate $ORACLE_HOME
      (PROGRAM= hsodbc)                      ◄ Agent Executable
      )
    )
```

5) Before starting the listener, make sure the ODBC lib directory is specified in the shared library environment variable.

Sample for DataDirect Connect *for* ODBC:

```
LD_LIBRARY_PATH=/opt/odbc32v53/lib:/db/oracle/product/10g/bin ◄ odbc lib path
```

Sample for DataDirect OpenAccess SDK ODBC:

```
LD_LIBRARY_PATH=/opt/oaodbc60/lib:/db/oracle/product/10g/bin  ◄ odbc lib path
```

After the LD_LIBRARY_PATH has been modified, start the listener.

If you do not want to add the LD_LIBRARY_PATH as an environment variable, you will need to add it to the listener.ora file. For example:

Sample for DataDirect Connect for ODBC:

```
(SID_DESC =
      (ORACLE_HOME = /db/oracle/product/10g)
```

```
      (SID_NAME = hsMydb)
      (PROGRAM = hsodbc)
      (ENVS=LD_LIBRARY_PATH =/opt/odbc32v53/lib:/db/oracle/product/10g/bin)
                    )
```

Sample for DataDirect OpenAccess SDK ODBC:

```
SID_DESC =
      (ORACLE_HOME = /db/oracle/product/10g)
      (SID_NAME = hsMydb)
      (PROGRAM = hsodbc)
      (ENVS=LD_LIBRARY_PATH
/opt/oaodbc56/lib/ssunos5:/db/oracle/product/10g/bin)
    )
```

6) Run "lsnrctl services" to verify that you now have a service handler for the hsMydb sid.

```
LSNRCTL> services
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXPPROC)))
Services Summary...
Service "hsMydb" has 1 instance(s).
   Instance "hsMydb", status UNKNOWN, has 1 handler(s) for this service…
      Handler(s):
         "DEDICATED" established:1 refused:0
            LOCAL SERVER
   The command completed successfully
```

7) Create the Initialization file. You must create and customize an initialization file for your generic connectivity agent. Oracle supplies a sample initialization file named "inithsodbc.ora", which is stored in the $ORACLE_HOME/hs/admin directory.

To create an initialization file, copy the appropriate sample file and rename the file to use the actual SID name chosen in step 4. . In this example, the SID noted in the listener and tnsnames files is "hsMydb" so the new initialization file would be called inithsMydb.ora.

Note that the SID name and the initialization file name are case sensitive.

8) Make sure the following entries are in the inithsMydb.ora now located in $ORACLE_HOME/hs/admin

Sample initialization file for DataDirect Connect *for* ODBC:

```
INITHSMYDB.ORA
 # HS init parameters
 #
 HS_FDS_CONNECT_INFO = MS_SQLServer2000      ◄ odbc data_source_name
 HS_FDS_TRACE_LEVEL = 0                      ◄ trace levels 0 - 4  (4 is verbose)
 HS_FDS_TRACE_FILE_NAME = hsmsql.trc         ◄ trace file name
 HS_FDS_SHAREABLE_NAME = /opt/odbc32v53/lib/libodbc.so ◄ full path to odbc driver manager
 #
 #
 # ODBC specific environment variables
 #
 set ODBCINI=/opt/odbc32v53/odbc.ini         ◄ location of odbc.ini
 #
 # Environment variables required for the non-Oracle system
 #
```

Sample initialization file for DataDirect OpenAccess SDK ODBC:

```
INITHSMYDB.ORA
# HS init parameters

#
HS_FDS_CONNECT_INFO = Test_MyDB          ◄ odbc data_source_name
HS_FDS_TRACE_LEVEL = 0                   ◄ trace levels 0 - 4  (4 is verbose)
HS_FDS_TRACE_FILE_NAME = hsoa.trc        ◄ trace file name
HS_FDS_SHAREABLE_NAME = =/opt/oaodbc60/lib/libodbc.so ◄ full path to odbc driver manager
#
#
# ODBC specific environment variables
#
set ODBCINI=/opt/oaodbc60/odbc.ini    ◄ location of odbc.ini
OASDK_ODBC_HOME=/opt/oaodbc60/lib/      ◄ location of OpenAccess lib directory
#
# Environment variables required for the non-Oracle system
#
```

9) Create a database link to access target database. Be sure to use the appropriate quotes as shown in the following example:

```
SQL> create database link hsdb            ◄ Link name can be anything you want
SQL> connect to "user" identified by "password" ◄ Must be a valid user/pwd on target DB
SQL> using 'hsalias';                     ◄ Name from Tnsnames.ora
```

10) To test, run a simple query of a known table on the target data store.

```
SQL> select * from employee@hsdb;

     empid firstname       lastname        department job
---------- --------------- --------------- ---------- ---
     10000 Joseph          Johnston        Sales      CDW
     10001 John            Ladd            Sales      WNV
     10002 Ronald          Wall            Relations  NPI
     10003 Julie           Reynolds        Relations  NPO
     10004 Bill            Baird           Telemarket PHN
     10005 Jason           Linde           Sales      WND
     10006 Edward          Lufner          Telemarket CDG
     10007 Mike            Seibt           Networking IDW

8 rows selected.
```

Common Errors and Solutions Associated with Heterogeneous Services and Generic Connectivity

The following list contains some of the most common errors associated with setting up Heterogeneous Services and Generic Connectivity.

ORA-28509: unable to establish a connection to non-Oracle system
ORA-02063: preceding line from HS
***Cause*:** This indicates a problem with the Oracle configuration files.
***Action*:**

- Make sure the HOST parameter in the tnsnames.ora file is correct
- Make sure the PORT number is correct
- Make sure the SID name is correct in both tnsnames.ora and listener.ora

ORA-02068: following severe error from HS
ORA-03114: not connected to ORACLE
***Cause*:** This indicates the required syntax for the tnsnames.ora file is not present.
***Action*:** Add (HS=OK) in the description section of the tnsnames.ora file.

ORA-02068: following severe error from HS
ORA-28511: lost RPC connection to heterogeneous remote agent using %tns_address%
***Cause*:** The listener is unable to spawn the HS agent or the agent cannot find the ODBC lib directory.
***Action*:** The PROGRAM line in the LISTENER.ORA file is incorrect or not specified. Make sure LD_LIBRARY_PATH includes the $ODBC_HOME/lib directory. If not, set LD_LIBRARY_PATH and restart the listener.

ORA-28500: connection from ORACLE to a non-Oracle system returned this message: [Transparent gateway for ODBC][H001] The environment variable <HS_FDS_CONNECT_INFO> is not set.
ORA-02063: preceding 2 lines from HS
***Cause*:** Incorrect parameter settings in the HS init.ora file.
***Action*:** Set HS_FDS_CONNECT_INFO in the HS init.ora file to the data source name located in the odbc.ini file.
***Example*:** HS_FDS_CONNECT_INFO = MS_SQLServer Wire Protocol
Make sure the HS init.ora file exists in the $ORACLE_HOME/hs/admin directory and has the same name as the SID in the LISTENER.ORA.
***Example*:** If SID=hsodbc in the listener.ora file, then the HS init.ora file would be named $ORACLE_HOME/hs/admin/inithsodbc.ora

ORA-28500: connection from ORACLE to a non-Oracle system returned this message:
[Transparent gateway for ODBC][H001] The environment variable
<HS_FDS_SHAREABLE_NAME> is not set.
ORA-02063: preceding 2 lines from HS
***Cause*:** Incorrect parameter settings in the HS init.ora file.
***Action*:** Set HS_FDS_SHAREABLE_NAME to the full path plus filename to the libodbc.so file.
Example: HS_FDS_SHAREABLE_NAME=/opt/odbc32v53/lib/libodbc.so

ORA-28500: connection from ORACLE to a non-Oracle system returned this message:
[Transparent gateway for ODBC]DRV_InitTdp:
(SQL State: 01000; SQL Code: 0)
ORA-02063: preceding 2 lines from HS
***Cause*:** The HS agent cannot find the odbc.ini file.
***Action*:** Set the ODBCINI variable in the HS init.ora file.
***Example*:** set ODBCINI=/opt/odbc32v53/odbc.ini

ORA-00942: table or view does not exist [Transparent gateway for ODBC]DRV_OpenTable:
[DataDirect][ODBC SQL Server Driver][SQL Server]Invalid object name '%table%'.
SQL State: S0002; SQL Code: 208)
ORA-02063: preceding 2 lines from HS
***Cause*:** The data source in the odbc.ini file has incorrect database information.
***Action*:** Consult the *DataDirect Connect for ODBC Reference Guide* for information on setting parameters for your datasource.

ORA-28500: connection from ORACLE to a non-Oracle system returned this message:
[Transparent gateway for ODBC]DRV_InitTdp: [DataDirect][ODBC SQL Server
Driver][libssclient15]General network error. Check your network documentation.
(SQL State: 08001; SQL Code: 11)
ORA-02063: preceding 2 lines from HS
Cause: There is a problem at the network layer communicating with the foreign data source.
***Action*:** Make sure the destination host or IP address and port number are correct for the data source in the odbc.ini file.

ORA-28500: connection from ORACLE to a non-Oracle system returned this message:
[Transparent gateway for ODBC]DRV_InitTdp: [DataDirect][ODBC SQL Server Driver][SQL Server] Login failed
(SQL State: 28000; SQL Code: 4002)
ORA-02063: preceding 3 lines from HSTEST
***Cause*:** The Oracle database link created for the foreign datasource has either no credentials or incorrect credentials.
***Action*:** Recreate the Oracle database link with the proper username and password. Also, username and password must be in double quotes.
***Example*:**

```
create database link ODBC connect to "sa" identified by "pencil"
using 'hsodbc'.
```

FOR MORE INFORMATION

800-876-3101

Worldwide Sales

Belgium (French).....0800 12 045
Belgium (Dutch).....0800 12 046
France.....0800 911 454
Germany.....0800 181 78 76
Japan.....0120.20.9613
Netherlands.....0800 022 0524
United Kingdom.....0800 169 19 07
United States.....800 876 3101

DataDirect TECHNOLOGIES

DataDirect Technologies is the software industry's only comprehensive provider of software for connecting the world's most critical business applications to data and services, running on any platform, using proven and emerging standards. Developers worldwide depend on DataDirect® products to connect their applications to an unparalleled range of data sources using standards-based interfaces such as ODBC, JDBC™ and ADO.NET, XQuery and SOAP. More than 300 leading independent software vendors and thousands of enterprises rely on DataDirect Technologies to simplify and streamline data connectivity for distributed systems and to reduce the complexity of mainframe integration. DataDirect Technologies is an operating company of Progress Software Corporation (Nasdaq: PRGS).

www.datadirect.com