

Institutional Data Integration Using ODBC with Oracle Heterogeneous Services

Mingguang Xu

email: mxu@uga.edu

Data Integration Challenges

- Information integration is a challenge that affects many organizations
- At the Office of Institutional Research, UGA, we need several data sources for web applications and reporting
 - 1. IBM IMS
 - 2. IBM DB2
 - 3. Flat files
 - 4. Oracle databases
- Our goal is to build a centralized data repository, either in traditional ER model or dimensional model

Oracle's Solution

- **Oracle Transparent Gateways**

- Oracle Transparent Gateway works in conjunction with the Heterogeneous Services component of the Oracle Database server to access a particular, commercially available, non-Oracle system. For example, you use the Oracle Transparent Gateway for Sybase on Solaris to access a Sybase database operating on a Sun Solaris platform

- **Generic Connectivity**

- Oracle provides a set of agents, containing only generic code, that interface with the Heterogeneous Services component and comprise Generic Connectivity. These agents require drivers to provide access to the non-Oracle systems. Oracle provides Generic Connectivity agents for ODBC and OLE DB that enable you to use ODBC and OLE DB drivers to access non-Oracle data sources

What Can HS Do for You

- Both Oracle Generic Connectivity and Oracle Transparent Gateways provide the ability to transparently access data in non-Oracle systems from an Oracle environment
- 1. Remote data can be accessed transparently
- 2. There is no unnecessary data duplication
- 3. SQL statements can query several different databases
- 4. Oracle's application development and end user tools can be used
- 5. Users can talk to a remote database in its own language

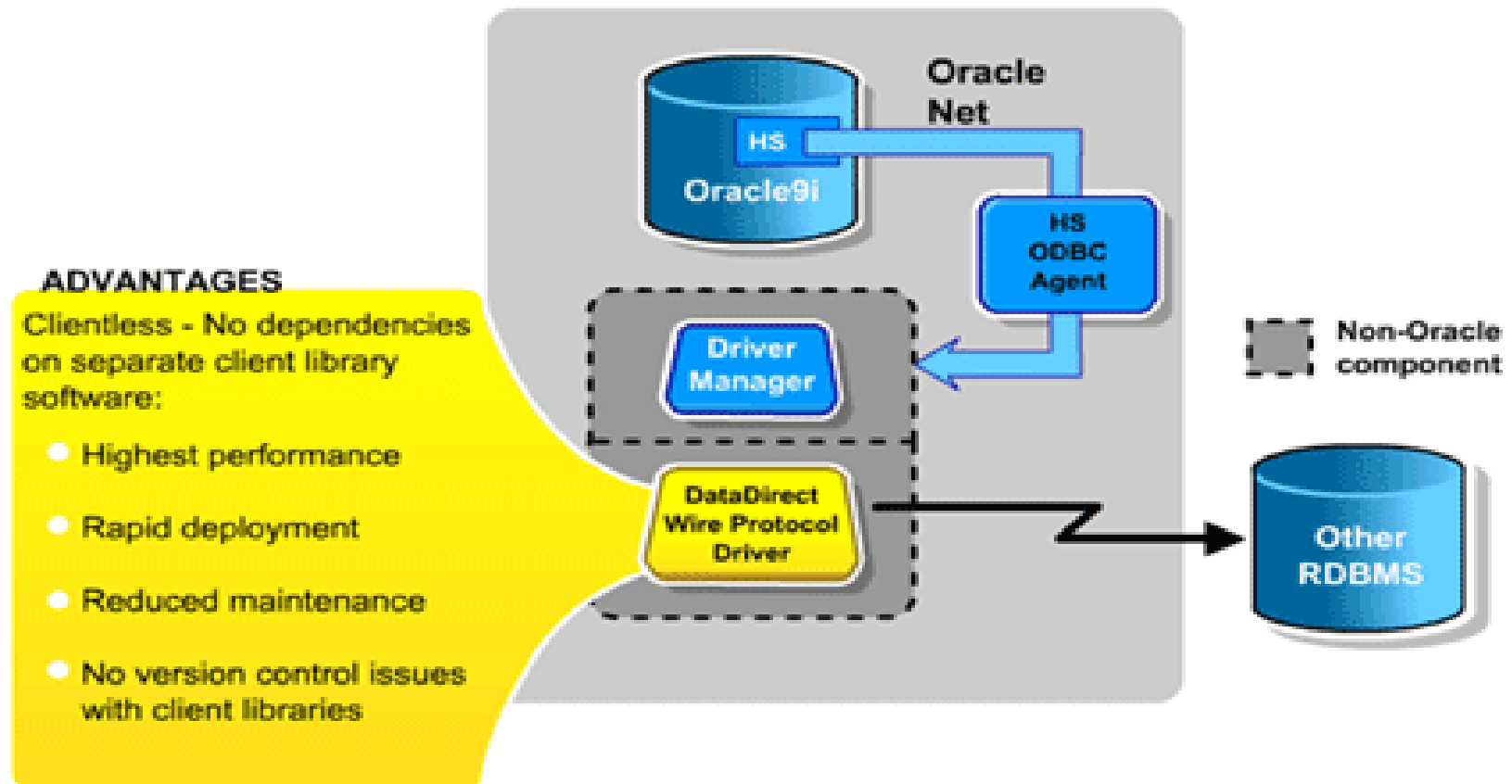
Advantages of Generic Connectivity

- No installation on data source machine
- No requirement to modify Oracle Client Library
- Easy to configure
- Easy to maintain
- Low cost
- Reliable

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. When Oracle DBMS installed, the agents are installed. From Oracle 10g, HS agents are available to Linux OS
- To access the non-Oracle data source 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.

Architecture of Generic Services



Architecture of Generic Services - *continued*

- 1. The client contacts the Oracle listener after resolving the service name
- 2. The listener redirect the connection request to HS agent
- 3. HS agent contacts ODBC driver by resolving Data Source Name (DSN)
- 4. Connect to non-Oracle data source based on DSN

Use Generic Connectivity with ODBC

- Set up ODBC driver
- Configure HSODBC

Set Up ODBC Driver

- We use the ODBC driver from Data-Direct (<http://www.datadirect-technologies.com>)
- Built-in mechanism to test ODBC connectivity
- Good support
- Reasonably priced

Set Up ODBC Driver _ *Continued*

- 1. Create a directory to contain the ODBC driver and related files, i.e. create an ODBC_HOME
- E.g.: /u01/app/odbc
- 2. Configure the odbc.ini file in \$ODBC_HOME: this file is similar to an address book for the odbc drivers

Configure odbc.ini file

- # define all odbc data source name
- # (DSN) here. The DSN can be anything you prefer

- *[ODBC Data Sources]*
- *peach=DataDirect 5.1 DB2 Wire Protocol*

- *dBase=DataDirect 5.1 dBaseFile(*.dbf)*

- *#Text=DataDirect 5.1 TextFile(*.*)*

- # DS specification
- *[peach]*
- *Driver=\$odbc_home/lib/ivdb221.so*
- *Description=DataDirect 5.1 DB2 Wire Protocol*
- *ApplicationUsingThreads=1*
- *Collection=D21*
- *IpAddress=host_name_of_DS*
- *Location=D21*
- *TcpPort=1234*
- *UseCurrentSchema=0*

- *[dBase]*
- *Driver=/opt/odbc32v51/lib/ivdbf21.so*
- *Description=DataDirect 5.1 dBaseFile(*.dbf)*
- *ApplicationUsingThreads=1*
- *CacheSize=4*
-

Configure HSODBC

- 1. tnsnames
- 2. listener
- 3. init<SID_NAME> of the HS subsystem
- 4. Oracle database

Modify tnsnames

- File location: `$oracle_home/network/admin`
- Add a new service name. Suppose the `service_name=juice`
- *Juice = # you may need to add a domain name suffix - ask your DBA*
(DESCRIPTION=
- *(ADDRESS_LIST=*
- *(ADDRESS= (PROTOCOL=TCP) # edit to point to your LISTENER*
(HOST=host_of_oracle server) # edit to point to your LISTENER
- *(PORT=8888) # edit to point to your LISTENER))*
- *(CONNECT_DATA=(SID=apple))*
- *(HS=OK))*

Modify tnsnames _ continued

- 1. (HS=OK) key word must be added manually. If opened with Oracle NCA, the entry will be erased.
- 2. (HS=OK) must be outside of SID section
- 3. After adding the new service, test it using
- > tnsping juice

Modify listener.ora

- File location: *\$oracle_home/newwork/admin*
- Add a new SID entry in this file. Suppose SID=apple

```
(SID_DESC=  
  (SID_NAME=apple)  
  (ORACLE_HOME=$oracle_home) # change this to your oracle home  
  (PROGRAM=hsodbc) #critical information  
  (ENVS=LD_LIBRARY_PATH=$oracle_home/lib:$odbc_home/lib  
)
```

This instructs the listener to service this sid "apple". You'll need to stop and start the listener to get it to pick up the changes

Configure HS Initiation File

- File location: `$oracle_home/hs/admin`. File name= `init<SID>.ora`

`initApple.ora`

- # 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 = peach #match DSN in $odbc_home/odbc.ini`
- `HS_FDS_TRACE_LEVEL = off`
- `HS_FDS_SHAREABLE_NAME = /opt/odbc32v51/lib/libodbc.so`

- # ODBC specific environment variables

- `set ODBCINI= $odbc_home/odbc.ini`

Configure Oracle Database

- Create a database link to point to the new service defined in `$oracle_home/network/admin/tnsnames.ora`
- Example:
- *create database link magic_link*
- *connect to "UID_at_YourDS" -- double quote is essential for some system*
- *Identified by "PW_at_YourDS" -- double quote is essential for some system*
- *using 'juice'*

Configuration Summary

- 1. Define 3-names: DSN (peach), SID_NAME (apple), SERVICE_NAME (juice)
- 2. Modify 4 files: \$odbc_home/odbc.ini,
\$oracle_home/hs/admin/initSID.ora,
\$oracle_home/network/admin/listener.ora
\$oracle_home/network/admin/tnsnames.ora
- 3. Define environment variables for odbc driver
- 4. Create database link to point to the outside data source

Common Errors

- ORA-28500: connection from ORACLE to a non-Oracle system returned this message: [Generic Connectivity Using ODBC][Microsoft][ODBC Driver Manager] Data source name not found and no default driver specified (SQL State: 00000; SQL Code: 0) ORA-02063: preceding 2 lines from CUSTARD The DSN name specified by "HS_FDS_CONNECT_INFO = SPONGE" in your %ORACLE_HOME%\hs\admin\initSID.ora could not be found. Check your iniSID.ora file and the ODBC manager.
- ORA-28545: Failed to make RSLV connection
- listener is not running, or has not been restarted.
- Could be that the PROGRAM in listener.ora is not 'hsodbc'
- Could be that the SID in tnsnames.ora is incorrect
- ORA-28500: connection from ORACLE to a non-Oracle system returned this message: [Generic Connectivity Using ODBC][H006] The init parameter <HS_FDS_CONNECT_INFO> is not set. Please set it in init<orasid>.ora file. ORA-02063: preceding 2 lines from JELLY Could be that the initSID.ora file is not named correctly. Match the SID to that in the listener.ora and tnsnames.ora files. -----
- ORA-12154: TNS:could not resolve service name The TNS Service name in your tnsnames.ora file does not match that specified in the 'using' clause of your database link.