

InfoSphere DataStage

Setting up the distributed transaction stage



© 2012 IBM Corporation

The purpose of this presentation is to give an overview on how to setup the Distributed Transaction Stage, referred to as DTS, with an Oracle database and how to troubleshoot configuration issues that might be encountered. These steps can be used with InfoSphere™ DataStage® 8.1 and 8.5. All examples are from AIX® 6.1 operating system with Oracle 11gR2 client and MQ Series 7.0.1.

Table of contents

- Overview
- Creating Oracle switch load file
- Common issue with compiling Oracle load file
- Adding resource manager to MQ queue manager configuration
- Verifying and debugging resource manager setup issue
- Creating job with MQ Connector and DTS

This presentation provides an overview and specific details on how to configure Oracle and MQ so DTS can be used in DataStage jobs. You will go through the steps on how to create an Oracle switch load file and learn common issues when compiling it. This presentation also shows how to add resource manager to the MQ queue manager configuration file and verify this configuration before creating the actual DataStage job with DTS.

Overview

- DTS configuration has three main areas
 - Switch load library creation
 - MQ Queue Manager configuration (qm.ini)
 - Special requirements in DataStage job

Getting DTS configured and running requires three distinguished tasks - Oracle switch load library creation, MQ queue manager configuration and setting up the DataStage job with a special requirement for DTS. This education module approaches each step individually and provides examples and resolutions to common issues in each specific area. These tasks typically require assistance from the MQ administrator and the Oracle database administrator in addition to the DataStage developer building the actual job with DTS.

Oracle switch file creation (1 of 2)

- MQ Server must be running on same host as DataStage engine
- Login as mqm (Administrative account for MQ)
- Ensure C compiler (xlc) is in path
- Set ORACLE_HOME
- Go to switch file source code location

```

bash-3.2$ id
uid=213(mqm) gid=210(mqm)
bash-3.2$ export PATH=/usr/vacpp/bin:$PATH
bash-3.2$ export ORACLE_HOME=/home/oracle/app/oracle/product/11.2.0/client_3
bash-3.2$ cd /usr/mqm/samp/xatm/
bash-3.2$ ls
amq0xab0.sqb      amqsxab0.ec      amqsxag0.c      amqsxas0.pc      oraswit.c      xaswit.exp
amq0xaf0.sqb      amqsxab0.sqc      amqsxag0.c.inf  amqsxas0.sqc     oraswitd.c     xaswit.mak
amq0xag0.cbl      amqsxaf0.ec      amqsxas0.cp     db2swit.c        sybswit.c      xaswit.mak.orig
amq0xas0.sqb      amqsxaf0.sqc     amqsxas0.ec     infswit.c        xa.h

```

The primary requirement for DTS is to have MQ Server installed on the same server as the DataStage engine. Remote MQ queue managers are not supported.

You also need to build a switch load library for the database to be used with DTS. The switch load file is a shared library that is loaded by the code in your WebSphere® MQ application and the queue manager. Its purpose is to simplify the loading of the database's client shared library, and to return the pointers to the XA functions.

Complete these tasks as the MQ administrator – typically the UNIX® account is mqm. Once you are logged on to the machine, setup your path to the compiler. For example, on AIX it is xlc_r and typically it is located in /usr/vacpp/bin.

The example displayed on the slide is using an Oracle database with DTS, so you need to setup the ORACLE_HOME environment variable so the shared libraries for the Oracle client can be found when the switch load library is being built.

The C code and make file for the switch load libraries are found under the MQ example directory, which is typically under /usr/mqm/samp/xatm.

Oracle switch file creation (2 of 2)

- Queue Manager must be stopped
- Compile Oracle switch load library
- Switch load library bitness must match DataStage bitness
- Note: Modifications to xaswit.mak file might be required depending on specific environment and configuration

```
bash-3.2$ endmqm QMGR
AMQ8146: WebSphere MQ queue manager not available.
bash-3.2$ make -f xaswit.mak oraswit
      xlc_r -e MQStart -bE:xaswit.exp -bM:SRE -q64 -I /usr/mqm/inc -L /home/oracle/app/oracle/product/11.2.0/client_3/lib -lcintsh -lm -o /var/mqm/exits64/oraswit oraswit.c
bash-3.2$ ls -ltr /var/mqm/exits64/oraswit
-rwx----- 1 mqm    mqm    1395 Oct 17 13:59 /var/mqm/exits64/oraswit
```

5

Setting up the distributed transaction stage

© 2012 IBM Corporation

MQ Queue Manager should be in a stopped state before attempting to compile the Oracle switch load library. The default make file might not work for all systems, so some changes might be required.

To build the Oracle switch load library, “cd” to the directory that contains the C program. Typically it is located in /var/mqm/samp/xatm. The ORACLE_HOME environment variable must be set so the compiler can locate the required Oracle shared libraries. The C compiler, XL C for AIX in this example, must be located in the path before starting the process.

Once the user environment is setup correctly you can build the switch library by issuing the command “Make -f xawit.mak oraswit”. If there were no errors during the build, there should be a new oraswit file in /var/mqm/exits64 and /var/mqm/exits directories. Default make file will build both 32bit and 64bit versions of the switch load library. Realize only the one that matches the bitness of DataStage is required.

Adjust the permissions to this file so the DataStage users are able to load it. The DataStage users will need read permissions to the oraswit file.

Common issues with Oracle switch file creation (1 of 3)

Compiler not found in path

Solution: Add compiler location to path

```
bash-3.2$ make -f xaswit.mak oraswit
xlc_r -e MQStart -bE:xaswit.exp -bM:SRE -q64 -I /usr/mqm/inc -L /lib -lclntsh -lm -o /var/mqm/
exits64/oraswit oraswit.c
/usr/bin/sh: xlc_r: not found.
make: 1254-004 The error code from the last command is 127.

Stop.
```

ORACLE_HOME not set or points to invalid location

Solution: Set ORACLE_HOME correctly

```
bash-3.2$ make -f xaswit.mak oraswit
xlc_r -e MQStart -bE:xaswit.exp -bM:SRE -q64 -I /usr/mqm/inc -L /lib -lclntsh -lm -o /var/mqm/
exits64/oraswit oraswit.c
ld: 0706-006 Cannot find or open library file: -l clntsh
ld:open(): A file or directory in the path name does not exist.
make: 1254-004 The error code from the last command is 255.

Stop.
```

This slide displays two common issues when building the Oracle switch load library. The first one is showing the error when compiler is not found in the path. To resolve this, ensure `xlc_r` is in your path (typically found in `/usr/vacpp/bin`).

The second one is showing the error when `ORACLE_HOME` is not set correctly. Run the `export` command with `ORACLE_HOME` set to the correct base directory for the Oracle client.

Common issues with Oracle switch file creation (2 of 3)

Queue Manager is up and running

Solution: Stop Queue Manager and try again

```
bash-3.2$ make -f xaswit.mak oraswit
      xlc_r -e MQStart -bE:xaswit.exp -bM:SRE -q64 -I /usr/mqm/inc -L /home/oracle/app/oracle/product/11.2.0/client_3/lib -lcintsh -lm -o /var/mqm/exits64/oraswit oraswit.c
ld: 0711-851 SEVERE ERROR: Output file: /var/mqm/exits64/oraswit
      The file is in use and cannot be overwritten.
make: 1254-004 The error code from the last command is 12.

Stop.
bash-3.2$ endmqm QMGR
Quiesce request accepted. The queue manager will stop when all outstanding work
is complete.
bash-3.2$ make -f xaswit.mak oraswit
      xlc_r -e MQStart -bE:xaswit.exp -bM:SRE -q64 -I /usr/mqm/inc -L /home/oracle/app/oracle/product/11.2.0/client_3/lib -lcintsh -lm -o /var/mqm/exits64/oraswit oraswit.c
bash-3.2$
```

The third common issue is the Queue Manager is in running state and there is an existing oraswit file that is locked by the running process. The simple solution is to stop Queue Manager and try recompiling the Oracle switch load library again.

Common issues with Oracle switch file creation (3 of 3)

Make file is trying to build 32 bit switch load file on 64 bit only system

Solution: Comment out 32bit line from xaswit.mak file and try again

```
bash-3.2$ make -f xaswit.mak oraswit
xlc_r -e MQStart -bE:xaswit.exp -bM:SRE -I /usr/mqm/inc -L /home/oracle/app/oracle/product/11.
2.0/client_3/lib32 -lclntsh -lm -o /var/mqm/exits/oraswit oraswit.c
ld: 0706-006 Cannot find or open library file: -l clntsh
ld:open(): A file or directory in the path name does not exist.
make: 1254-004 The error code from the last command is 255.
```

```
oraswit:
# $(CC) $(CFLAGS32) -I $(MQINCDIR) $(ORALIBFLAGS32) -o $(MQEXITDIR32)/oraswit oraswit.c
$(CC) $(CFLAGS64) -I $(MQINCDIR) $(ORALIBFLAGS64) -o $(MQEXITDIR64)/oraswit oraswit.c
```

The fourth issue that may occur is lack of 32bit Oracle client libraries on the system. If you do not have \$ORACLE_HOME/lib32, and only have \$ORACLE_HOME/lib, this might indicate that the only 64bit Oracle client libraries are available and a small change in the make file is required. To avoid building the 32bit Oracle switch load library, comment the highlighted line from xaswit.mak file and try again.

Defining XA Resource Manager (1 of 2)

- MQ Server must be running on same server as DataStage
- DBA must run \$ORACLE_HOME/rdbms/admin/initxa.sql on Oracle server
- User specified in XAOpenString must be granted select permissions on DBA_PENDING_TRANSACTIONS table
- Add XA Resource Manager to /var/mqm/<Queue Manager>/qm.ini
- Example:

```
XAResourceManager:  
  Name=oral  
  SwitchFile=oraswit  
  XAOpenString=Oracle_XA+Acc=F/scott/tiger+Seq=20+SqlNet=ORACLE11+DbgFl=0x07+LogDir=/tmp/xalogs+Threads=true  
  ThreadOfControl=THREAD
```

The second part of the process is to setup Oracle as Resource Manager in qm.ini and setup the Oracle side as well.

The Oracle DBA should run the xainit.sql script that is typically located in \$ORACLE_HOME/rdbms/admin, and grant the user specified in the XAOpenString permissions to run select queries against the table called DBA_PENDING_TRANSACTIONS.

Once the Oracle side is setup, the next step is to add the XA Resource Manager to the qm.ini file for the Queue Manager as seen in the screen capture displayed on this slide. This particular XAOpenString sets the Username to scott, password to tiger, Oracle database name to ORACLE11 and the XA related log directory to /tmp/xalogs. Adjust these settings as needed, the rest can be left as they are.

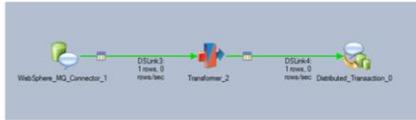
Defining XA Resource Manager (2 of 2)

- Starting Queue Manager
- Check logs for errors
 - XA related logs (/tmp/xalogs in example)
 - MQ logs (/var/mqm/errors)
 - Queue Manager logs (/var/mqm/qmgrs/<Queue Manager>/errors)
- Example
 - Database user does not have select permission on DBA_PENDING_TRANSACTIONS table
 - /tmp/xalogs/xa<uniqueid>.trc file will have error similar to ORA-00942: table or view does not exist

Queue Manager needs to be started after the XA Resource Manager has been added to the qm.ini file. Once the Queue Manager is up and running, check the MQ, Queue Manager and XA related logs for any possible errors before proceeding to the job design. One common error is the ORA-00942 error in the XA Resource Manager logs, indicating that the user does not have permission to run the required select statements against the DBA_PENDING_TRANSACTIONS table.

Once the logs are error free, its time to move to the job design.

Job design (1 of 3)



- Required settings in the MQ Connector Stage:
 - MQ work queue must be used
 - Transaction should be set to individual message (Record Count = 1)
 - End of Wave marked should be set to After

Properties Columns Advanced	
▼ Usage	
Queue name	QUEUE1
Access mode	As in queue definition
▶ Other queue settings	No
Wait time	-1
Message quantity	1
▼ End of data message type	
▶ Process end of data message	Yes
▶ Refresh	No
Message read mode	Move to work queue
▼ Work queue	
Name	WORKQ
Append node number	No
Context mode	Set all
▶ Monitor queue depth	No
▼ Transaction	
Record count	1
Time interval	0
▶ Message controlled	No
▶ Blocking transaction processing	No
▼ End of wave	After
End of data	Yes
▶ Message options	No
▶ Error queue	No
▶ Filter messages	No
▶ Publish/Subscribe	No

The next step is to build a DataStage job. The example displayed on this slide uses a simple job that consists of three stages: MQ Connector, Transformer and DTS.

There are some special requirements for DTS to function as intended and the next few slides will cover each requirement one by one. The first step will focus on the MQ Connector properties. This example is using a single node configuration and only one work queue called WORKQ for the messages that are being processed. Limiting transaction size to one is important as well to ensure each MQ message is a single XA unit of work and the end of the wave must come after each row.

Job design (2 of 3)

- Required settings in DTS Connector Stage
 - Work Queue name must match what was used with MQ Connector
 - Requires Reject Queue for failed transactions
 - Oracle connection settings (same as in the qm.ini)

The diagram illustrates the configuration of a DTS Connector Stage. It shows a central icon labeled 'DSLLink4' with an arrow pointing to it from the left. Below the icon are two property windows. The left window is titled 'Properties | Columns | Advanced | Partitioning' and shows the 'Connection' section with the following settings:

Connection	
XA database name	
Server	ORACLE11
Username *	scott
Password *	*****
Use external authentication	No
Usage	

The right window is titled 'Properties | Advanced' and shows the 'Advanced' section with the following settings:

Connection	
Queue manager *	QMGR
Work queue *	WORKQ
Append node number	No
Usage	
Use MQ messaging	Yes
Reject failing units	Yes
Reject queue *	REJECTQ
Prepend reject	No
Context mode *	None
Log successful units	No
Abort after this many failures	0
Order records	No

12

Setting up the distributed transaction stage

© 2012 IBM Corporation

DTS contains two sets of properties that can be accessed by clicking the link of the icon in the stage properties view.

Clicking the input link will let you set the properties for the Oracle database and these should match the settings you had defined for the XA resource manager in qm.ini file for the queue manager.

Selecting the icon lets you define the required MQ properties for the DTS. Work queue name should be set in the same way as it is defined in the MQ Connector stage. The reject queue, where all failed transactions are routed, must be specified as well.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, AIX, DataStage, InfoSphere, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.