DBMoto®

Setup Notes for Oracle Transactional Replications (Triggers)

Version 9.5.0.20

Software Release Date: 4/23/18

HiT Software, Inc.
4040 Moorpark Ave
Suite 221
San Jose, CA 95117

T +1 408-345-4001
F +1 408-345-4899
info@hitsw.com
www.hitsw.com

Document History

Version	Author	Date	Reviewer	Date	Approver	Date	Comments
1	JHLorenzin	9/4/18	ALaspertini		VFarruggio		

Table of Contents

These notes provide essential information for setting up replications using **Oracle** as a source database for one-way mirroring and synchronization**.** The setup process for a refresh replication can usually be completed using the DBMoto wizards without additional documentation because it does not involve access to the Oracle logs.

For complete details on the setup process, check the *DBMoto User Guide* available from the Management Center **Help** menu or the *DBMoto Setup Guide*, available for download in the Help Center.

This guide describes the setup process using the Triggers options for one-way mirroring and synchronization when replicating data from an Oracle database. For mirroring and synchronization replications using Oracle as a source, DBMoto offers several approaches:

- **Log Reader**: Queries the Oracle Log Miner directly. This is the default option and is described in the document *Setup Notes for Oracle Transactional Replications* available in the Help Center.
- **Log Server Agent**: Uses a Windows service and a DBMoto Log Server component to query the Oracle Log Miner for increased performance when dealing with large amounts of data. This option is supported for Oracle databases version 11 and above. This approach is described in the document *Setup Notes for Oracle Transactional Replications* available in the Help Center.
- **Triggers**: Uses DBMoto triggers installed on the Oracle database to log changes. This approach may be preferred in your application.

Connection Type

Oracle .NET Provider available from Oracle with the Oracle Database Client.

Assembly: Oracle.DataAccess (file name: Oracle.DataAccess.dll)

Sample path: C:\oracle10_2\client\odp.net\bin\2.x\Oracle.DataAccess.dll.

For Oracle version 10 clients, you need to configure the Net Service Name for the client to access the Oracle server.

Trigger-Based Replication Overview

A database trigger is code that is automatically executed in response to certain events on a database table. To define a trigger-based replication (mirroring or synchronization), you need to provide information in the Source and/or Target Connection wizards so that triggers can be created to log table changes for replication.

For each table involved in the replication, DBMoto creates 3 triggers in the source table that fire when a specific event occurs on a record:

- INSERT trigger which fires when a new record is being inserted in the table
- UPDATE trigger which fires when a record is modified
- DELETE trigger which fires when a record is deleted

If the replication is later deleted, the triggers are removed by DBMoto. However, note that if you change a replication from mirroring to refresh, the triggers on the source table are not deleted. All transactions will continue to be recorded in the log tables. If you are not planning to reset the replication to mirroring, it is better to delete the replication, so that the triggers are removed, and create a new refresh replication.

Data retrieved using the triggers is stored in log tables that are specified in your Source/Target connection. The master log table can be an existing table or one created specifically to hold DBMoto information. It contains general information about the transactions, like user name, timestamp, table name. A log table (_DBM__LOG_x) is also created for each source table in the replication, and contains the data changes identified by the triggers, as well as trigger objects _DBM__TRG_OBJS.

Note that DBMoto does not create a tablespace. If you want to have a table space named DBMOTO, you must create it beforehand using a SQL tool. Run the appropriate SQL statement for your environment. For example:

CREATE TABLESPACE DBMOTO

When creating a connection, it is important to set the retention time to keep the log table size under control. The higher the value, the more data is kept in the log tables. Try to estimate the number of transactions occurring in all the source tables during a retention period and be sure that the database and table space have enough storage capacity for all those transactions. The DBReplicator (engine) cleans up the log tables periodically, based on the retention setting in the connection dialog. If the engine is not running, the log tables are not cleaned up. This might create space problems in the database as the logs grow in size. If you stop the engine and you are not planning to run it again, be sure to remove all the mirroring synchronization replications.

In addition, if you have many table replications in a single group, using a single connection, all the replications share a master log table. Access to the log table for each source table can become a bottleneck if there are many transactions using the same master log and log tables. DBMoto may report errors about locked tables during replication. Although DBMoto is able to recover from these errors and continue replicating, a better approach is to prevent the errors by splitting the replications into multiple groups with multiple connections and multiple master log tables. First, create multiple source connections to the database. Use the Transaction Log Type field in the Connection Properties of each connection to open the Setup Info dialog and create a new master log table for each connection.

During replication:

- When a record is inserted in the source table, the INSERT trigger fires and inserts one record in the master table and one record in the log table associated with the source table. The record inserted in the log table contains all the original values of the INSERT statement.
- When a record is deleted from a table, the DELETE trigger fires and inserts one record in the master table and one record in the log table associated with the source table. The record inserted in the log table contains the key values of the deleted record.
- When a record is updated, the UPDATE trigger fires and inserts one record in the master table and two records in the log table associated with the source table. The two records inserted in the log table contain all the record values before and all the records after the update.

A transaction ID is saved both on the master records and log record to maintain a link between the transaction and the data changes for that transaction. See "DBMOTO Log Tables" for more details on the structure of the Master and Log tables.

Your system administrator needs to create and define appropriate table spaces and databases to hold the log tables. They should be large enough to handle the expected amount of replication data.

DBMOTO Log Tables

Log tables are used to record all data changes made to the source tables. They are populated by triggers that are fired when source tables are modified. Currently, log tables must reside in the replication source database. Note that log tables are created by DBMoto only if they do not already exist in the database.

There are two log tables associated with each replication: a master table, common to all replications using that connection, and a log table, one for each replication. The master table keeps track of all the transactions affecting the source tables and it records general transactional information.

Master table structure

- `TID DECIMAL(31,0) GENERATED BY DEFAULT AS IDENTITY:`
  Transaction ID number associated with each record data change (transaction)
- `SNAME VARCHAR(128):`
  Name of the schema the transaction was applied to.
- `TNAME VARCHAR(128):`
  Name of the table the transaction was applied to.
- `TTS TIMESTAMP:`
  Transaction timestamp indicating when the transaction was submitted to the system
- `TUSER VARCHAR(128):`
  Name of the user who executed the transaction

Log Table Structure

The Log table contains the actual data changes for a specific source table. Its structure depends on the structure of the source table.

- `__TID DECIMAL(31,0):`

  Transaction ID, link to the corresponding record in the master table
- `__OP CHAR:`

  Indicates the type of operation:

  'I' INSERT
  'D' DELETE
  'B' UPDATE: values before the update
  'A' UPDATE: values after the update
- `<Field list>:`

All the columns of the source table with their original data type. For example if the source table was created as:

```
CREATE TABLE SOURCET
(ID INTEGER,
 FNAME VARCHAR(30),
 LNAME VARCHAR(50))
```

The log table will have the following structure:

```
CREATE TABLE _DBM__SOURCET
(__TID DECIMAL(31,0),
 __OP  CHAR,
 ID    INTEGER,
 FNAME VARCHAR(30),
 LNAME VARCHAR(50))
```

Reading From/Managing Log Table

DBMOTO reads the log tables using the SELECT SQL statement. First, it queries the master table to see if new transactions came in since the last processed __TID. If transactions are found, DBMOTO queries the corresponding log table to collect the data changes and apply them to the target table.

The SELECTS on the master and log tables are sorted by the unique column __TID which ensures that all records will be read in the order that they were written. DBMOTO also uses the unique __TID column to keep track of the point where the last record was read and processed from the log tables.

DBMOTO provides options to manage log records that have been read and replicated. They can be deleted from the log table as soon as they are processed or a retention time can be set to leave this record in the log tables for the specified number of hours.

Oracle User Permissions

When setting up replications that use Oracle as either a source or target database, you need to be sure that the user ID used for making connections to the database has sufficient privileges to complete all the operations required for DBMoto to perform a replication.

This section is organized by the type of replication you want to perform. It describes in detail all the user authorities that will be required during the setup and execution of replications.

Refresh with Oracle as Either Source or Target Database

1. AUTHORITY TO CONNECT TO DATABASE

   To open a connection to an Oracle database, you need specific authority for a user ID using either of the following two syntaxes:

   ```
   grant create session to <uid>;
   ```

   OR

   ```
   grant connect to <uid>;
   ```

Example where dbmoto is the user ID:

```
grant create session to dbmoto;
```

OR

```
grant connect to dbmoto;
```

2. AUTHORITY TO SELECT CATALOG

   To display a list of tables and show fields in the table in the Management Center (for selecting a source or target table and for setting which fields to replicate), DBMoto runs a SELECT_CATALOG command. If the user ID has insufficient privileges, an error is generated on the Oracle server.

   ```
   grant select any table to <uid>;
   ```

   Example where dbmoto is the user ID:

   ```
   grant select any table to dbmoto;
   ```

3. AUTHORITY TO SELECT TABLES

   DBMoto runs a SELECT statement to identify records to replicate. Therefore, the user ID used to make a connection must have adequate authority to run a SELECT statement for tables involved in replication.

   If you used `grant select any table` above, you do not need to grant select authority to specific tables involved in replications because the above command covers all tables. However, in case you need it, the command to grant access to a specific table is:

   To verify existing authorities for a user ID, you can run the EDTOBJAUT command, where <TYPE> can be *LIB, *FILE, etc.

   ```
   grant select on <table> to <UID>;
   ```

   Example where dbmoto is the user ID:

   ```
   grant select on SAMPLE.EMPLOYEES to dbmoto;
   ```

4. AUTHORITY TO UPDATE TABLES, CREATE TABLES (Optional)

   To create a target table in the Management Center (as part of the Create Table Wizard), DBMoto uses the following commands.

   ```
   grant unlimited tablespace to <uid>;
   ```

   You first need to grant a quota on the tablespace in which you want to create a table or index. Then you can grant create permissions and update permissions (insert, update and delete.)

   ```
   grant create any table to <uid>;
   grant insert any table, update any table, delete any table to <uid>;
   ```

   Example where dbmoto is the user ID:

   ```
   grant unlimited tablespace to dbmoto;
   grant create any table to dbmoto;
   grant insert any table, update any table, delete any table to dbmoto;
   ```

   The insert, update and delete commands are broader than needed. They can also be granted to specific tables.

5. AUTHORITY TO DROP TABLES, ALTER TABLES (Optional)
   The use of these commands from within DBMoto is entirely optional (i.e. not necessary for running a refresh replication.) They are used if you choose to remove a table from Oracle or change the table via the Management Center SQL Query tab. The following commands are broader than needed. Alter and drop can also be granted to specific tables.
   ```
   grant alter any table to <uid>;
   grant drop any table to <uid>;
   ```

Transactional Replications/Initial Refresh with Oracle as Either Source or Target Database

This section includes information for mirroring where Oracle is the data source, and synchronization where Oracle can be either the "source" or "target" data source.

1. (OPTIONAL) CREATION OF TABLESPACE
   HiT Software recommends that you assign a tablespace for the Master table and Log Tables so that it is easier to control log table sizes. Note that DBMoto does not create a tablespace, but the user can specify access to a tablespace. If you want to have a table space named DBMOTO, you must create it beforehand:
   ```
   create tablespace <tablespace> datafile <data_file> size <size>
   ```

   Example where DBMOTO_TBLSPACE is the name of the tablespace and dbmoto_tblspace.dat is the datafile with an initial allocation of 100 Mb:
   ```
   create tablespace DBMOTO_TBLSPACE

   datafile 'dbmoto_tblspace.dat'

    size 100M

   online;
   ```

2. CREATE USER FOR DBMOTO OPERATIONS AND TABLESPACE PERMISSIONS
   The DBA should create a user for DBMoto operations on the machine, then grant privileges to a specific tablespace with the command below. The Oracle user ID you are planning to use should have sufficient permissions to complete all operations in DBMoto: permissions to connect, select tables, insert/update/delete records and so on.
   ```
   alter user <UID> quota unlimited on <TBLSPACE>;
   ```

   Example where DBMOTO_TBLSPACE is the name of the tablespace and DBMOTO is the name of the user.
   ```
   alter user DBMOTO quota unlimited on DBMOTO_TBLSPACE;
   ```

3. AUTHORITY TO CONNECT TO DATABASE
   To open a connection to an Oracle database, you need specific authority for a user ID using either of the

following two syntaxes:
```
grant create session to <uid>;
```
OR
```
grant connect to <uid>;
```

Example where dbmoto is the user ID:
```
grant create session to dbmoto;
```
OR
```
grant connect to dbmoto;
```

4. AUTHORITY TO SELECT A CATALOG
   To display a list of tables and show fields in the table in the Management Center (for selecting a source or target table and for setting which fields to replicate), DBMoto runs a SELECT_CATALOG command. If the user ID has insufficient privileges, an error is generated on the Oracle server.
   ```
   grant select any table to <uid>;
   ```
   Example where dbmoto is the user ID:
   ```
   grant select any table to dbmoto;
   ```

5. AUTHORITY TO SELECT TABLES
   DBMoto runs a SELECT statement to identify records to replicate. Therefore, the user ID used to make a connection must have adequate authority to run a SELECT statement for tables involved in replication.
   If you used `grant select any table` above, you do not need to grant select authority to specific tables involved in replications because the above command covers all tables. However, in case you need it, the command to grant access to a specific table is:
   To verify existing authorities for a user ID, you can run the EDTOBJAUT command, where <TYPE> can be *LIB, *FILE, etc.
   ```
   grant select on <table> to <UID>;
   ```
   Example where dbmoto is the user ID:
   ```
   grant select on SAMPLE.EMPLOYEES to dbmoto;
   ```

6. AUTHORITY TO UPDATE TABLES, CREATE TABLES
   To create a target table in the Management Center (as part of the Create Table Wizard), DBMoto uses the following commands.
   You first need to grant a quota on the tablespace in which you want to create a table or index. Then you can grant create permissions and update permissions (insert, update and delete.)
   ```
   alter user <uid> quota unlimited on <tablespace>;
   grant create any table to <uid>;
   grant insert any table, update any table, delete any table to <uid>;
   ```
   Example where dbmoto is the user ID:
   ```
   alter user dbmoto quota unlimited on tblspace;
   ```

```
grant create any table to dbmoto;
grant insert any table, update any table, delete any table to dbmoto;
```

The insert, update and delete commands are broader than needed. They can also be granted to specific tables.

7. AUTHORITY TO DROP TABLES, ALTER TABLES (Optional)
   The use of these commands from within DBMoto is entirely optional (i.e. not necessary for running a refresh replication.) They are used if you choose to remove a table from Oracle or change the table via the Management Center SQL Query tab. The following commands are broader than needed. Alter and drop can also be granted to specific tables.
   ```
   grant alter any table to <uid>;
   grant drop any table to <uid>;
   ```

8. SET UP TRANSACTIONS
   To set up transactional replications, you need permission to create and drop triggers in the schema where the _DBM__TRG_OBJS table resides:
   ```
   grant create any trigger to <UID>;

   grant drop any trigger to <UID>;
   ```

   Example:
   ```
   grant create any trigger to DBMOTO;
   grant drop any trigger to DBMOTO;
   ```

9. ACCESS TRANSACTION LOG
   You need to access the transaction log information stored in the table _DBM__TRG_OBJS. Note that if you revoke read access, the replication works but it is unable to read the last ID.
   ```
   grant select on <DEFAULT_SCHEMA>."_DBM__TRG_OBJS" to <UID>;
   ```

   Example where the user ID is DBMOTO:
   ```
   grant select on SCOTT."_DBM__TRG_OBJS" to DBMOTO;
   ```

NOTE: If you are using Oracle versions 12c and above Pluggable Database (PDB) architecture, contact the support team via the Help Center for additional instructions.

Oracle System Settings

Install Oracle Client with .NET Provider

Before connecting to an Oracle database from DBMoto, make sure the Oracle .NET provider is installed and accessible from the system where DBMoto is running.

If you install DBMoto on the system where the Oracle server is installed, the Oracle .NET Data Provider should already be installed as part of the Oracle installation and DBMoto should be able to find the provider automatically.

If you install DBMoto on a different system, to connect to the Oracle server you need to install the Oracle client on the same system as DBMoto.

Add a Source Connection Wizard

Select Provider Screen

Assembly

The value for Oracle should be the pathname to the .NET Assembly Oracle.DataAccess (file name: Oracle.DataAccess.dll.) For later versions of Oracle, you can leave the Assembly field blank because the dll path should be available to DBMoto. (The dll is registered during installation of the Oracle .NET Data Provider.) If the value is not available, DBMoto displays a message when you continue in the Source Connection wizard, allowing you to go back and type in the path. Find out the location of the assembly in your environment by searching for the file name Oracle.DataAccess.dll, then enter the path and the assembly file name as in the example below.

Set Connection String Screen

Data Source

Type the IP address of the server, then a colon and the port number, followed by / and the Oracle service name (for Oracle versions 11 and later) as in the example below.

```
122.333.4.555:1521/ORADB1
```

User ID

Enter a user ID which will be exclusively used by DBMoto and has the authority to read the database transaction log (redo log.) See a detailed list of authorities needed.

For Synchronization Replications:
The login/user ID that you provide must be unique to DBMoto. It should not be used for any transactions occurring in either database involved in the synchronization. DBMoto does not replicate transactions by the user you specify in this connection. This user ID is used by DBMoto during synchronization to read the database logs and perform the synchronization operations. Therefore, any transactions found in the logs with this user ID are not replicated as part of the synchronization data.

Enable Transactional Replication Wizard

For transactional replications (mirroring and synchronization), use the Enable Transactional Replication wizard after setting up a source connection. The following field(s) require specific information for Oracle.

Log Type Screen

Select the Triggers option.

Trigger Settings Screen

Master Table

Either specify an existing qualified table name, or click Change to create a new table to hold general information about replication transactions including user name, timestamp, table name for each transaction.

There are two tables associated with each replication: a Master table, common to all replications using that connection, and a Log table for each replication source table. The Master table keeps track of all the transactions affecting the source tables and it records general transactional information.

Master and Log tables are created in the schema specified when you set the Master table name. You can choose a Master table name, or use the default _DBM__MASTERLOG. Log tables are automatically generated by DBMoto and the names are _DBM__LOG_#, where # is a number. The selected schema for the Master and Log tables must not contain other non-DBMoto tables with names _DBM__LOG_# . HiT Software recommends that you create a new schema to use specifically for the DBMoto Master and Log tables.

Tablespace

HiT Software recommends that you assign a tablespace for the Master table and Log Tables so that it is easier to control log table sizes. If you leave this field blank, the default tablespace value for your login ID will be used. Your system administrator should be able to provide you with the appropriate value for this field.

Retention Time

The amount of time in hours that a transaction is kept in the log tables. The default value is 72 hours. When the amount of time a transaction resides in the log exceeds the retention time, the transaction is permanently removed from the log tables. Tuning the retention time provides control over the size of the log tables. It is also possible to instruct DBMoto to remove all the processed transactions at the end of each mirroring interval. Tuning the retention time provides control over the size of the log tables.

Delete Block Size

Based on the retention time, DBMoto deletes items from the log. This field specifies the maximum number of records to delete from the DBMoto log tables with a single SQL statement. The default value is 10,000 records. You do not typically need to edit this value.

Lower-case Trigger Identifiers

Check this option if your database installation uses lower-case trigger identifiers.

Trigger Order

Always inactive for Oracle sources.

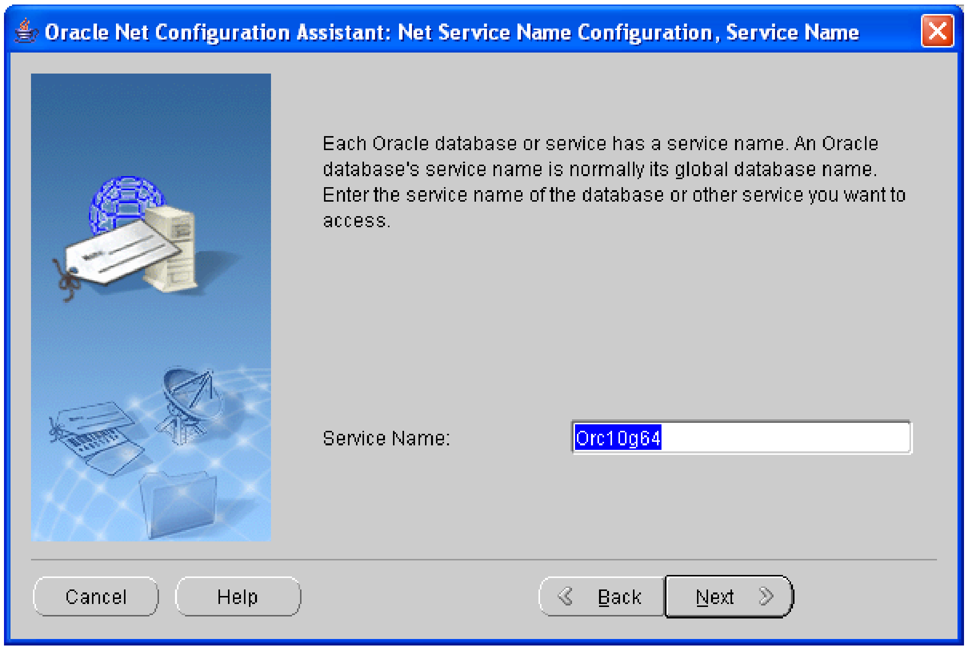
Configure the Oracle Client

Oracle 11 and 12 Client installation does not require extra steps to set up the Net Service Name for Oracle. For this reason, HiT Software recommends using the Oracle 11 or 12 Client. However, for those who are required to use Oracle 10, this section provides information on configuring the Oracle 10 Client and the Net Service Name for the Oracle client.

1. Run the installation process for the Oracle client that contains the Oracle ,NET Provider.

   The Oracle Net Configuration Assistant starts up automatically after you install the Oracle client. You can also start it manually from the Windows Start menu: All Programs -> Oracle – OraClient10g_homes1 -> Configuration and Migration Tools -> Net Configuration Assistant.

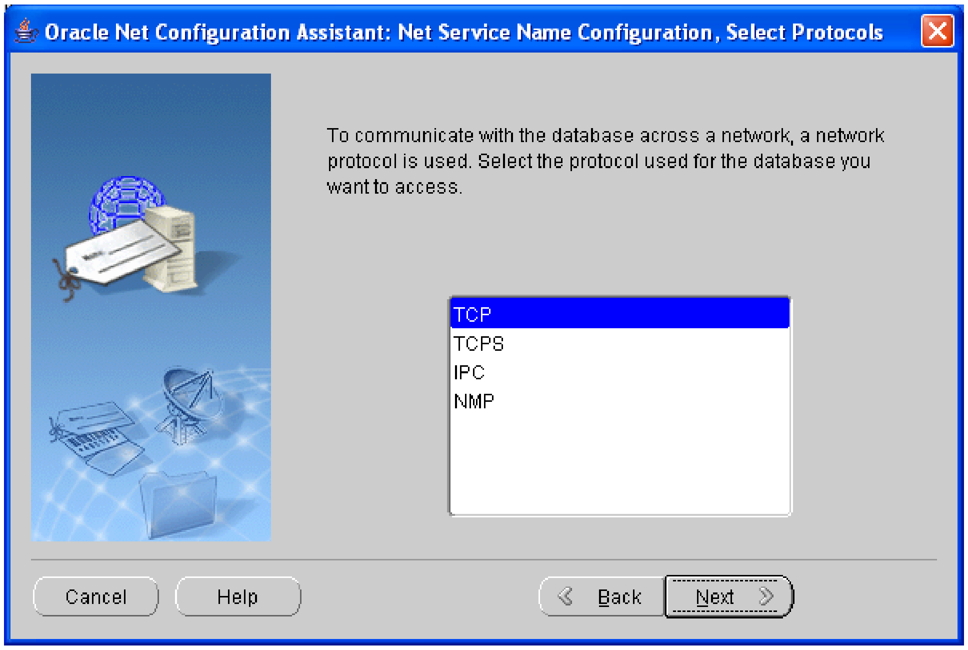
2. Enter the service name and click **Next**.


Oracle Net Configuration Assistant: Net Service Name Configuration, Service Name
Cancel
Help
Back
Next

Each Oracle database or service has a service name. An Oracle database's service name is normally its global database name. Enter the service name of the database or other service you want to access.

Service Name: Orc10g64

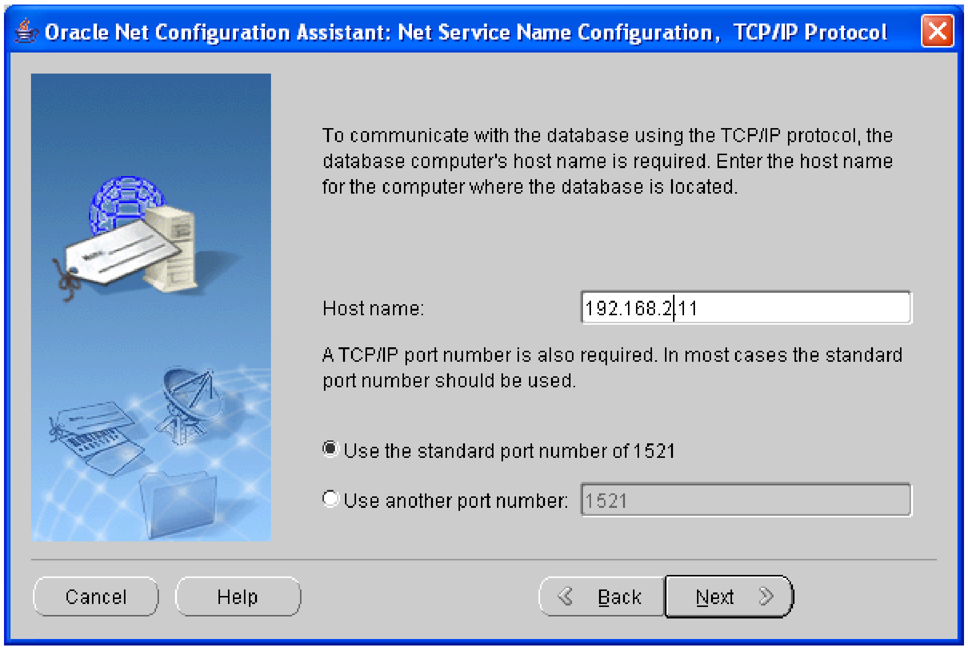
3. For the ACCESS_LOG network protocol, select **TCP**.


Oracle Net Configuration Assistant: Net Service Name Configuration, Select Protocols

To communicate with the database across a network, a network protocol is used. Select the protocol used for the database you want to access.

- TCP
- TCPS
- IPC
- NMP

4. Enter the IP address of your Oracle system and click **Next**.


Oracle Net Configuration Assistant: Net Service Name Configuration, TCP/IP Protocol

To communicate with the database using the TCP/IP protocol, the database computer's host name is required. Enter the host name for the computer where the database is located.

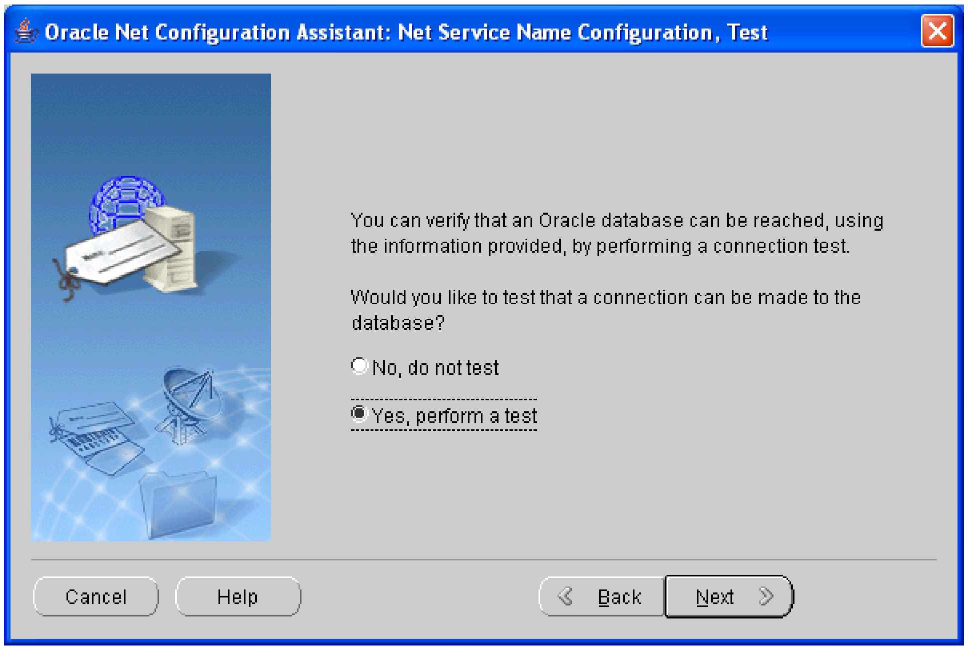
Host name: 192.168.2.11

A TCP/IP port number is also required. In most cases the standard port number should be used.

● Use the standard port number of 1521

○ Use another port number: 1521

5. Select the option **Yes, perform a test** and click **Next**.



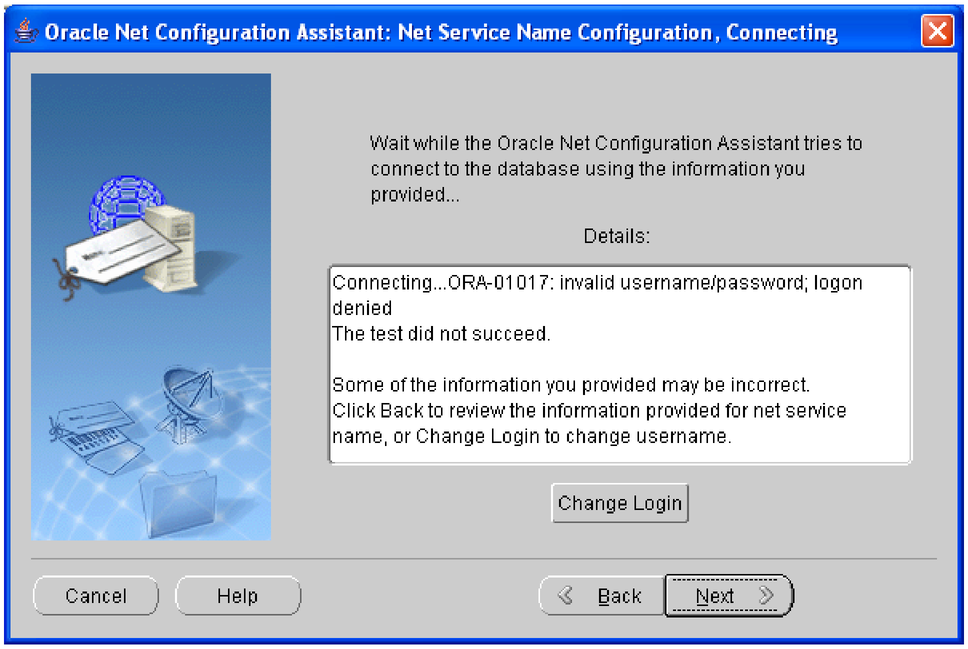
Oracle Net Configuration Assistant: Net Service Name Configuration, Test

You can verify that an Oracle database can be reached, using the information provided, by performing a connection test.

Would you like to test that a connection can be made to the database?

No, do not test [✓ Yes, perform a test]

6. The test may fail because you do not have the correct login and password information.



Oracle Net Configuration Assistant: Net Service Name Configuration, Connecting

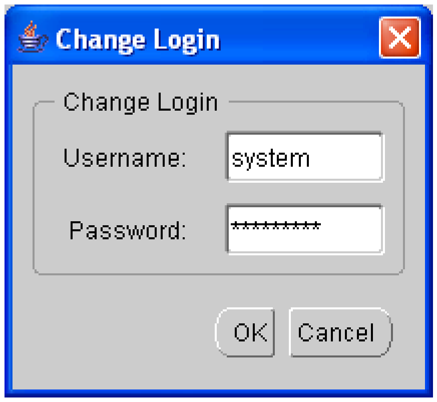
Wait while the Oracle Net Configuration Assistant tries to connect to the database using the information you provided...

Details:

```
Connecting...ORA-01017: invalid username/password; logon
denied
The test did not succeed.

Some of the information you provided may be incorrect.
Click Back to review the information provided for net service
name, or Change Login to change username.
```

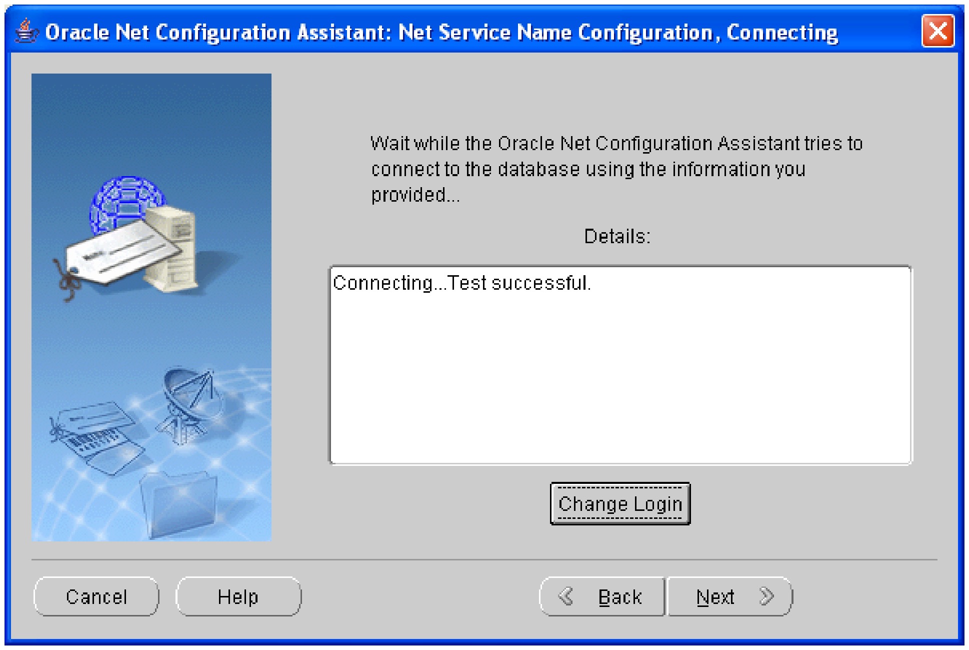
7. Click **Change Login**.



Username: system

Password: **********

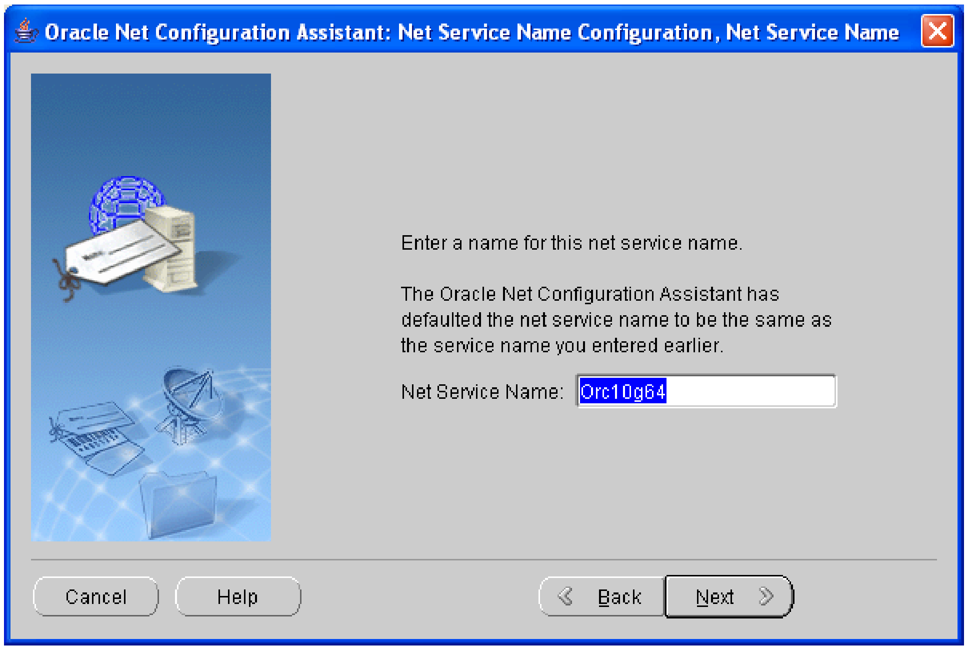
8. Enter the user ID and password of your Oracle system, then click **OK**.


Oracle Net Configuration Assistant: Net Service Name Configuration, Connecting
Details:
Change Login
Cancel
Help
Back
Next

Wait while the Oracle Net Configuration Assistant tries to connect to the database using the information you provided...

Connecting...Test successful.

9. When the test is successful, click **Next**.
10. Either accept the default for the NET service name, or enter a new service name.

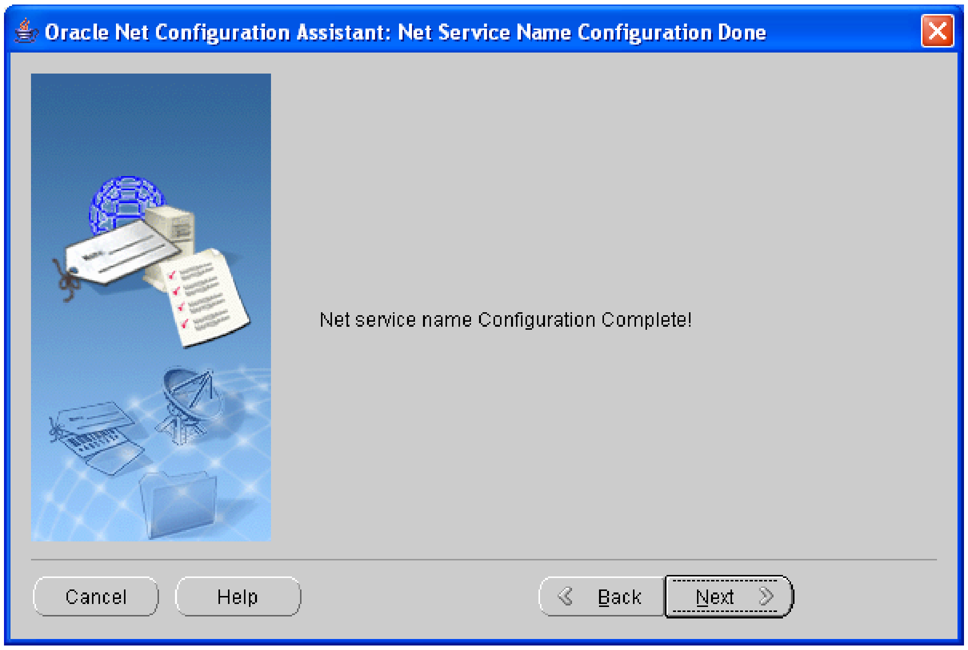

Oracle Net Configuration Assistant: Net Service Name Configuration, Net Service Name

Enter a name for this net service name.

The Oracle Net Configuration Assistant has defaulted the net service name to be the same as the service name you entered earlier.

Net Service Name: Orc10g64

11. Follow the on-screen instructions to complete the setup.


Oracle Net Configuration Assistant: Net Service Name Configuration Done

Net service name Configuration Complete!

Last Updated on 9/13/2018