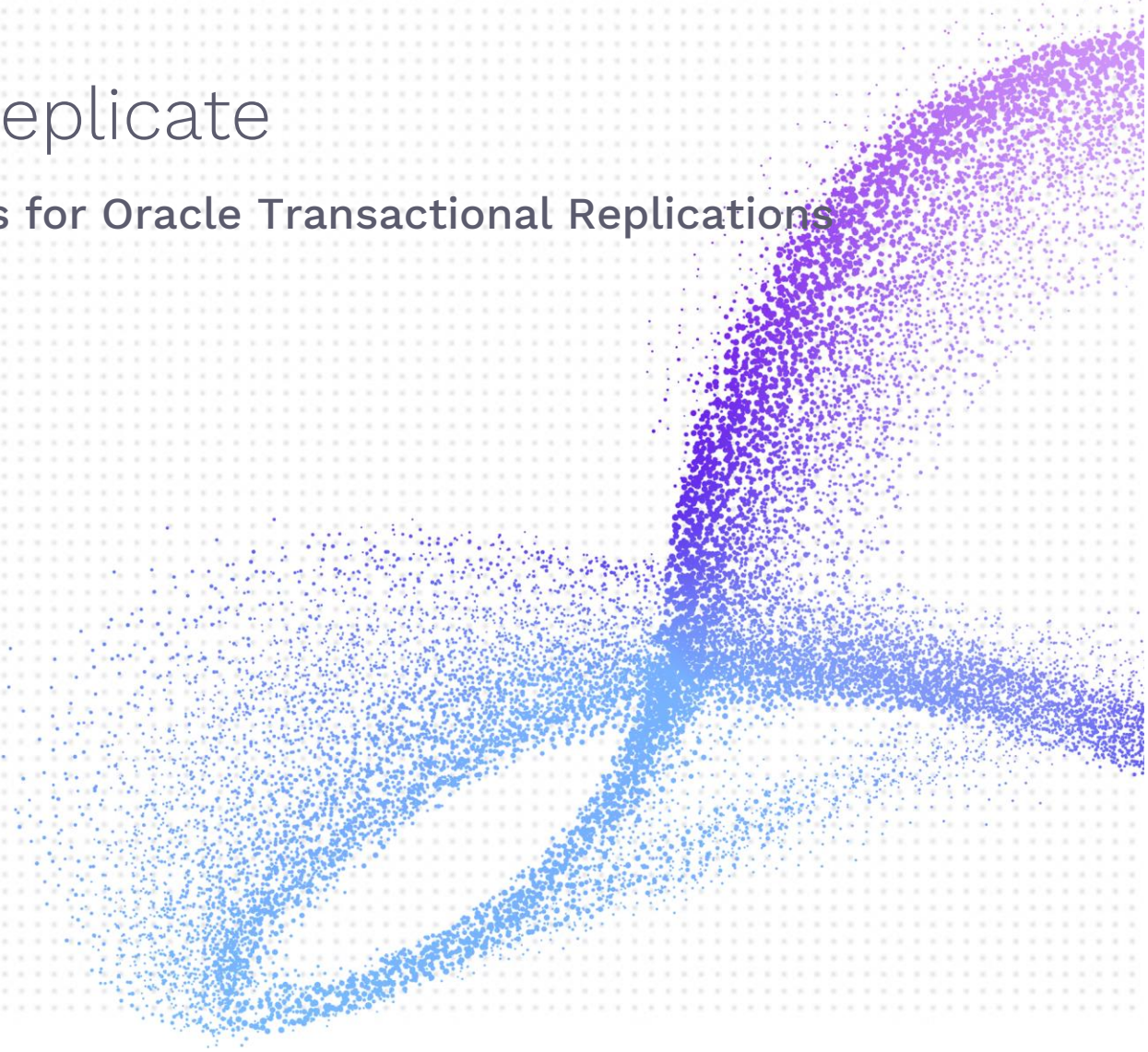
Syniti

Syniti Replicate

Setup Notes for Oracle Transactional Replications

Version 10.3



Syniti

Syniti Replicate

Document History

Version	Author	Date	Reviewer	Date	Comments
1	JHLorenzin	8/27/18	ALaspertini		
2	JHLorenzin	1/15/20	GDeBeni		
3	JHLorenzin	4/17/20	SUpadhaya		
4	VFarruggio	5/28/20	JHLorenzin	5/28/20	Fixed permissions
5	JHLorenzin	9/4/20	VFarruggio	9/4/20	Added Oracle Log Retention Time
6	JHLorenzin	1/21/21			Added Ref to RLSA
7	JHLorenzin	4/23/21	VFarruggio	4/23/21	Fix P.5
8	JHLorenzin		VFarruggio		SESSION info for synchronization

Syniti Replicate

Table of Contents

Syniti Replicate

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 Syniti Replicate wizards without additional documentation because it does not involve access to the Oracle logs.

In addition to on-premise versions of Oracle, Syniti Replicate also supports Autonomous Database, both Transaction Processing and Data Warehouse modules. However, transactional replications with Autonomous Database are only supported using triggers, so refer to the *Setup Notes for Oracle Using Triggers*.

This guide describes the setup process using either the Log Reader or Log Server Agent options for one-way mirroring and synchronization when replicating data from an Oracle database. For mirroring and synchronization replications using Oracle as a source, Syniti Replicate offers several approaches:

- **Log Reader**: Queries the Oracle Log Miner directly
- **Log Server Agent**: Uses a Windows service and a 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. Syniti Replicate also offers a Remote Log Server Agent configuration on request from the technical support team at https://support.syniti.com/hc/en-us/requests/new.
- **Remote Log Server Agent**: Uses a Java application installed on the Oracle server to log changes. Ask the technical support team via the Help Center to see if this approach would work in your application. Download the separate document *Setup For Oracle Replications with Remote Log Server Agent*.
- **Triggers**: Uses Syniti Replicate triggers installed on the Oracle database to log changes. Ask the technical support team via the Help Center to see if this approach would work in your application. Download the separate document *Setup Notes for Oracle Using Triggers*.

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

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.

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 Syniti Replicate 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 sdruser is the user ID:

```
grant create session to sdruser;
```

OR

```
grant connect to sdruser;
```

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), Syniti Replicate runs a SELECT_CATALOG command. If the user ID has insufficient privileges, an error is generated on the Oracle server.

```
grant select on SYS.ALL_USERS to <uid>;
grant select on SYS.ALL_TABLES to <uid>;
grant select on SYS.ALL_TAB_COMMENTS to <uid>;
grant select on SYS.ALL_OBJECTS to <uid>;
grant select on SYS.ALL_VIEWS to <uid>;
grant select on SYS.ALL_TAB_COLUMNS to <uid>;
grant select on SYS.ALL_COL_COMMENTS to <uid>;
grant select on SYS.ALL_CONSTRAINTS to <uid>;
grant select on SYS.ALL_CONS_COLUMNS to <uid>;
grant select on SYS.USER_CONSTRAINTS to <uid>;
grant select on SYS.USER_CONS_COLUMNS to <uid>;
grant select on SYS.ALL_IND_COLUMNS to <uid>;
```

   Example where sdruser is the user ID:

```
grant select on SYS.ALL_USERS to sdruser;
```

   etc.

3. AUTHORITY TO SELECT TABLES

   Syniti Replicate 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.

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

   Example where sdruser is the user ID:

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

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

   (**Note: Needed only when Oracle is a target**)

   To create a target table in the Management Center (as part of the Create Table Wizard), Syniti Replicate requires the permissions below. You first need to grant a quota on the tablespace in which you want to create a table or index. You can either grant unlimited tablespace to the user:

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

   Or, just define a quota limit on a specific tablespace:

```
alter user <uid> QUOTA 100M on <tablespace name>;
```

   Then you can grant create permissions and update permissions (insert, update and delete.)

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

Example where sdruser is the user ID:

```
grant unlimited tablespace to sdruser;
grant create on SAMPLE.EMPLOYEES to sdruser;
grant insert, update, delete on SAMPLE.EMPLOYEES to sdruser;
```

The insert, update and delete commands can be as broad as needed. They can also be granted to entire schemas or to all database tables.

5. AUTHORITY TO DROP TABLES, ALTER TABLES (**Optional**)
   (**Note: Needed only when Oracle is a target**)
   The use of these commands from within Syniti Replicate 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 can be as broad as needed. They can also be granted to entire schemas or to all database tables.

```
grant alter on <table> to <uid>;
grant drop on <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. 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 sdruser is the user ID:

```
grant create session to sdruser;
```

   OR

```
grant connect to sdruser;
```

2. 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), Syniti Replicate runs a SELECT_CATALOG command. If the user ID has insufficient privileges, an error is generated on the Oracle server.

```
grant select on SYS.ALL_USERS to <uid>;
grant select on SYS.ALL_TABLES to <uid>;
grant select on SYS.ALL_TAB_COMMENTS to <uid>;
grant select on SYS.ALL_OBJECTS to <uid>;
grant select on SYS.ALL_VIEWS to <uid>;
grant select on SYS.ALL_TAB_COLUMNS to <uid>;
grant select on SYS.ALL_COL_COMMENTS to <uid>;
grant select on SYS.ALL_CONSTRAINTS to <uid>;
```

```
grant select on SYS.ALL_CONS_COLUMNS to <uid>;
grant select on SYS.USER_CONSTRAINTS to <uid>;
grant select on SYS.USER_CONS_COLUMNS to <uid>;
grant select on SYS.ALL_IND_COLUMNS to <uid>;
```

Example where sdruser is the user ID:

```
grant select on SYS.ALL_USERS to sdruser;
```

etc.

3. AUTHORITY TO SELECT TABLES
   Syniti Replicate 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.

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

   Example where sdruser is the user ID:

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

4. AUTHORITY TO UPDATE TABLES, CREATE TABLES (**Optional**)
   (**Note: Needed only when Oracle is a target**)
   To create a target table in the Management Center (as part of the Create Table Wizard), Syniti Replicate requires the permissions below.
   You first need to grant a quota on the tablespace in which you want to create a table or index. You can either grant unlimited tablespace to the user:

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

   Alternatively, just define a quota limit on a specific tablespace:

   ```
   alter user <uid> QUOTA 100M on <tablespace name>;
   ```

   Then you can grant create permissions and update permissions (insert, update and delete.)

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

   Example where sdruser is the user ID:

   ```
   grant unlimited tablespace to sdruser;
   grant create on SAMPLE.EMPLOYEES to sdruser;
   grant insert, update, delete on SAMPLE.EMPLOYEES to sdruser;
   ```

   The insert, update and delete commands can be as broad as needed. They can also be granted to entire schemas or to all database tables.

5. AUTHORITY TO DROP TABLES, ALTER TABLES (**Optional**)
   (**Note: Needed only when Oracle is a target**)
   The use of these commands from within Syniti Replicate 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 can be as broad as needed. They can also be granted to entire schemas or to all database tables.

   ```
   grant alter on <table> to <uid>;
   grant drop on <table> to <uid>;
   ```

6. AUTHORITY TO SET UP A CONNECTION FOR TRANSACTIONAL REPLICATIONS (logminer)
   (**Note: Needed only when Oracle is a source in mirroring replications**)
   When setting up the connection for transactional replications (where Oracle is serving as the source of data, either for mirroring or for synchronization), you may need additional privileges that are not required for replication operations. You need to alter the database to enable SUPPLEMENTAL LOG.

   In Syniti Replicate (DBMoto) versions 7.0.4 and above, the supplemental log is enabled at the table level by default rather than for the entire database. In the default case, you need an ALTER DATABASE privilege to set up supplemental logging and an ALTER TABLE privilege on each table to be replicated when setting up a replication. If the table already has a supplemental log set, then Syniti Replicate does not need this privilege. You can control whether supplemental logging is enabled at the table or database level using the Oracle Setup Info dialog available from the Connection Properties dialog (under Transaction Log Type.)

   To set up supplemental logging independent of Syniti Replicate (i.e. not through the Oracle Setup Info dialog), use the following commands:

   For database-level supplemental logging:

   ```
   ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE INDEX) COLUMNS
   ```

   For minimal/table-level supplemental logging:

   ```
   ALTER DATABASE ADD SUPPLEMENTAL LOG DATA
   ```

   In this last case, also an ALTER TABLE statement will be used on each replicated table:

   ```
   ALTER TABLE MYSCHEMA.MYTABLE ADD SUPPLEMENTAL LOG GROUP <groupname> (<pkList>) ALWAYS
   ```

   Where *pkList* is a comma-separated list of the primary keys of the table, while *groupname* is an internally generated name of the supplemental log group.

   If using a dictionary file (required for Oracle 8.1, optional for later versions) you need execute permission to build the dictionary file.

   ```
   grant alter on <table> to <uid>;
   grant alter database to <uid>;
   grant execute on sys.dbms_logmnr_d to <uid>;
   ```

   Example:

   ```
   grant alter on SAMPLE.EMPLOYEES to sdruser
   grant alter database to sdruser
   grant execute on sys.dbms_logmnr_d to sdruser;
   ```

7. AUTHORITY TO SET UP TRANSACTIONAL REPLICATIONS and ACCESS REDO/ARCHIVED LOGS (logminer)
   (**Note: Needed only when Oracle is a source in mirroring replications**)
   When setting up the connection for transactional replications (where Oracle is serving as the source of data, either for mirroring or for synchronization), the following permissions are also required: SELECT_TABLE permissions on the source table (see above); execute permission to run the logminer; and access all system

tables to retrieve log information and the last transaction ID.

```
grant execute on sys.dbms_logmnr to <uid>;
grant select on sys.v_$parameter to <uid>;
grant select on sys.v_$log to <uid>;
grant select on sys.v_$logfile to <uid>;
grant select on sys.V_$logmnr_contents to <uid>;
grant select on sys.V_$thread to <uid>;
grant select on sys.V_$archive_dest to <uid>;
grant select on SYS.AUD$ to <uid>;
```

To use archived logs to identify database changes for mirroring, you must execute the following:

```
grant select on sys.v_$archived_log to <uid>;
```

For Oracle versions 10g and above, execute the following:

```
grant select on sys.v_$database to <uid>;
```

If supplemental logging is minimal, the following additional command is required

```
grant select on DBA_LOG_GROUP_COLUMNS to <uid>;
```

For Oracle versions 12c and above, execute the following:

```
grant select on cdb_pdbs to <uid>;
grant logmining to <uid>;
grant select on sys.v_$database to <uid>;
grant select on sys.v_$containers to <uid>;
grant select on DBA_LOG_GROUPS to <uid>;
grant select on DBA_LOG_GROUP_COLUMNS to <uid>;
grant EXECUTE_CATALOG_ROLE to <uid>;
```

Example:

```
grant select any transaction to sdruser;
grant logmining to sdruser;
grant select on sys.v_$archived_log to sdruser;
grant execute on sys.dbms_logmnr to sdruser;
grant select on sys.v_$parameter to sdruser;
grant select on sys.v_$log to sdruser;
grant select on sys.v_$logfile to sdruser;
grant select on sys.V_$logmnr_contents to sdruser;
grant select on sys.V_$thread to sdruser;
grant select on sys.V_$archive_dest to sdruser;
grant select on SYS.AUD$ to sdruser;
grant select on sys.v_$database to sdruser;
grant select on sys.v_$containers to sdruser;
grant select on DBA_LOG_GROUPS to sdruser;
grant select on DBA_LOG_GROUP_COLUMNS to sdruser;
grant EXECUTE_CATALOG_ROLE to sdruser;
```

For Oracle versions 12c and above using Pluggable Database (PDB) all the GRANT statements should be run on the container:

```
grant execute on sys.dbms_logmnr_d to <uid> container=ALL;
grant select on sys.v_$parameter to <uid> container=ALL;
grant select on sys.v_$log to <uid> container=ALL;
grant select on sys.v_$logfile to <uid> container=ALL;
grant select on sys.V_$logmnr_contents to <uid> container=ALL;
grant select on sys.V_$thread to <uid> container=ALL;
grant select on sys.V_$archive_dest to <uid> container=ALL;
grant select on sys.v_$archived_log to <uid> container=ALL;
grant select any transaction to <uid> container=ALL;
grant select on cdb_pdbs to <uid> container=ALL;
grant logmining to <uid> container=ALL;
grant select on sys.v_$database to <uid> container=ALL;
grant select on sys.v_$containers to <uid> container=ALL;
grant select on DBA_LOG_GROUPS to <uid> container=ALL;
grant select on DBA_LOG_GROUP_COLUMNS to <uid> container=ALL;
grant EXECUTE_CATALOG_ROLE to <uid> container=ALL;
ALTER USER <uid> SET CONTAINER_DATA=ALL container=CURRENT;
```

Oracle System Settings

Install Oracle Client with .NET Provider

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

If you install Syniti Replicate 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 Syniti Replicate should be able to find the provider automatically.

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

Oracle Log Retention Time Setting

In the Oracle database, log retention time determines how long the transaction information contained in the logs is retained, and not deleted or purged. If the need arises for Syniti Replicate to read earlier transactions from the database log, the retention time is a crucial parameter in determining the success or failure of this operation. The longer the retention time, the better the chances are of finding the required transactions in the database. If transactions are not found, a replication will fail and subsequently a full refresh is required to resynch the data.

Possible reasons why Syniti Replicate would need to read earlier transactions are:

- Prolonged stop of the Replication Agent.

- Prolonged disabled time for a replication, or any other condition that puts the replication in recovery mode.
- Prolonged maintenance activity of the servers involved (which may cause downtime of one or more services).
- Long refresh time (during refresh, the backlog of transactions gets larger and the effort to read them will be higher).
- Massive number of transactions executed on the source database for a continued period of time (a condition that impacts performance and, as a consequence, may require the log reader process to fall behind in order to read all transactions).

To manage conditions like those above, a log retention time of 72 hours is recommended. If 72 hours is not possible, a reasonable heuristic to use is: take the longest refresh time among all replications, and multiply that by two, with a minimum log retention time of 4 hours. However, note that if an extended period of downtime occurs on the system running Syniti Replicate or Oracle, these suggestions will not be sufficient and a full refresh will be needed when the system restarts.

Using Oracle Autonomous Database

If using Oracle Autonomous Database (either as a target or as a source with triggers), the following settings are needed:

- tnsnames.ora and sqlnet.ora should be placed in the ORACLE_HOME folder. Sample of ORACLE_HOME path: C:\app\client\supadhaya\product\12.2.0\client_1\network\admin
- Sqlnet.ora should NOT have: “SSL_SERVER_DN_MATCH=yes”.
  Sample of sqlnet.ora text where “DIRECTORY” is the path where the cwallet.sso file is present
  WALLET_LOCATION = (SOURCE = (METHOD = file) (METHOD_DATA = (DIRECTORY="C:\app\client\supadhaya\product\12.2.0\client_1\network\admin")))
- The client IP should be given access to the DB instance by adding it to the list in the Database’s “Access Control List”.

Setting Up for Synchronization

When replicating from Oracle using log-based synchronization mode, it is necessary to identify a user by retrieving session user information. This is achieved by enabling auditing: use the auditing `SESSION` statement option to generate a single audit record for each session created by connections to the instance. An audit record is inserted into the audit trail at connect time and updated at disconnect time.

Set the audit trail to ‘db’, to direct audit records to the database audit trail (the SYS.AUD$ table). Allowed values are:

db

db, extended

1. Connect to the database and run the following SQL statement:

```
ALTER SYSTEM SET audit_trail=DB SCOPE=SPFILE;
```

2. Restart the database instance after issuing this command.

To audit all successful and unsuccessful connections to and disconnections from the database, regardless of user, `by SESSION` (the default and only value for this option), connect to the database and run the following SQL statement:

```
AUDIT SESSION;

```

You can also set this option selectively for individual users:

```
AUDIT SESSION
BY dbmoto;
```

The effect of this command is persistent after database reboot.

In case you need to turn auditing off, you can do so by using the NOAUDIT statement, one for each AUDIT statement that you want to turn off. If, for example, you have run both the commands shown above, to turn them off, run:

```
NOAUDIT SESSION;

NOAUDIT SESSION
BY dbmoto;
```

Once auditing is enabled, querying DBA_STMT_AUDIT_OPTS returns the following record:

	USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE
▶	DBMOTO	(null)	CREATE SESSION	BY ACCESS	BY ACCESS

Querying DBA_PRIV_AUDIT_OPTS returns:

	USER_NAME	PROXY_NAME	PRIVILEGE	SUCCESS	FAILURE
▶	DBMOTO	(null)	CREATE SESS	BY ACCESS	BY ACCESS

There is no need to add any options to these views because the AUDIT SESSION command was used.

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 Syniti Replicate. (The dll is registered during installation of the Oracle .NET Data Provider.) If the value is not available, Syniti Replicate 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 Syniti Replicate 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 Syniti Replicate. It should not be used for any transactions occurring in either database involved in the synchronization. Syniti Replicate does not replicate transactions by the user you specify in this connection. This user ID is used by Syniti Replicate 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 whether you plan to perform replications using the Log Reader (default) or Log Server Agent.

Log Settings Screen

Log Settings

The Log Settings field displays the Oracle data source where the log is stored. This field is read-only and by default matches the data source specified in the source connection. If you are using the multitenant database architecture introduced in Oracle 12, and connecting to a pluggable database, you may need to modify the log location to match the Oracle root, or container, database. To modify the log location, click the [...] button.

Click [...] to open the Change Log Settings dialog to:

- Change the Oracle server where the log is stored (for Oracle 12 multitenant database installations).
- Set a path to an Oracle dictionary file (primarily for Oracle versions up to 9)
- Change the archived log settings

Supplemental Log Minimal Level/Supplemental Log Database Level

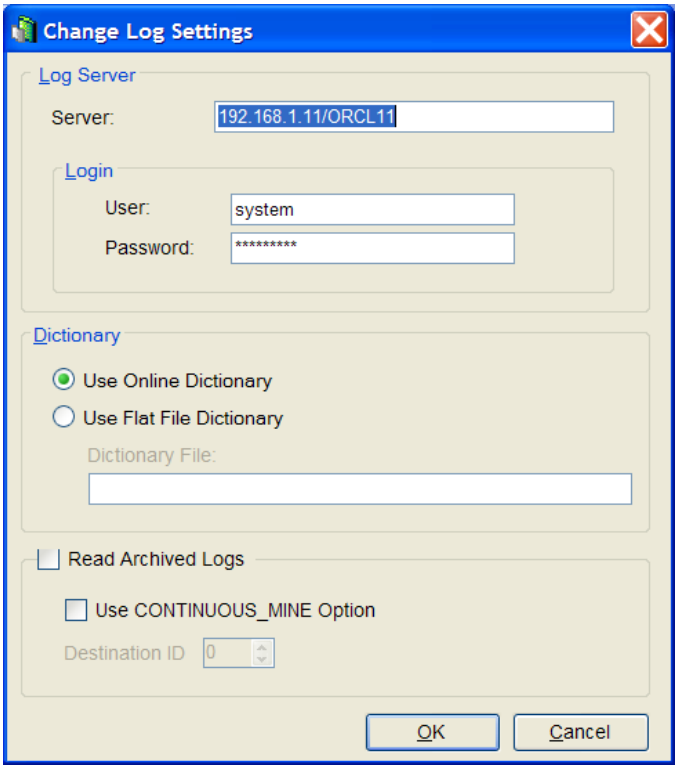
These fields are set automatically depending on what type of supplemental logging is currently set for your Oracle source connection. For transactional replications where Oracle is a source database, Syniti Replicate requires supplemental logging to be enabled at least at the minimal level (and later at the table level) to record sufficient information about changes to the data in the table.

Minimal Level establishes minimal supplemental logging at the database level, then, during replication setup, you can add logging for the specific table(s) involved in the replication.

Database Level establishes supplemental logging for the entire database, including primary key and index information for all tables. If you have a large number of tables in your database, and they are not involved in Syniti

Replicate replications, you should be aware that supplemental logging can add unnecessary processing time/log information.

Change Log Settings Dialog



Server

Specifies the Oracle data source where the log is stored. If you are using the multitenant database architecture introduced in Oracle 12, and connecting to a pluggable database, you need to modify the log location to match the Oracle root, or container, database.

User

User ID for the Oracle data source where the log is stored.

Password

Password for the Oracle data source where the log is stored.

Dictionary File

If the log miner dictionary is defined as a flat file (typically only for Oracle versions up to 9), type the path and file name for an Oracle dictionary file. The Oracle dictionary file should already be defined in your Oracle environment. See Setup for Different Oracle Versions for more information.

Read Archived Logs

When checked, Syniti Replicate accesses Oracle archived redo logs for transactions in addition to the online redo logs. Check with your Oracle system administrator to see if your database is configured to use ARCHIVELOG mode. If it is, you can check the **Read Archived Logs** option to ensure that all transactions between mirroring intervals

are identified by Syniti Replicate. For example, if your mirroring interval is set for 90 seconds, but within that time the online logs fill with transactions, the online logs are archived as needed to accommodate additional transactions. If you check the **Read Archived Logs** option, Syniti Replicate will be able to locate the last transaction ID from the archived log. Note that the **Read Archived Logs** option is not useful when using the default setting on Oracle databases, NOARCHIVELOG mode.

Use CONTINUOUS_MINE Option

When **ReadArchivedLogs** is checked, the **Use CONTINUOUS_MINE Option** is checked by default. The CONTINUOUS_MINE option instructs the Oracle log reader to load archived log files continuously as they get created, instead of loading a batch of files statically when the log reader is initially instantiated. Syniti Replicate can therefore replicate transactions as they are backed up. However, Oracle recommends that the CONTINUOUS_MINE option should not be used in a RAC (Real Application Cluster**)** environment where RAC nodes are continuously enabled/disabled.

Destination ID

This option allows you to set the location from which Syniti Replicate will pick up Oracle archived logs for replication. When the value is set to the default, 0, Syniti Replicate searches for Oracle archived logs in the standard destination, ('FLASH RECOVERY AREA (FRA)'). A value between 1 and 31 specifies an alternative location to retrieve archived logs. Oracle users can retrieve a list of all archive locations by running the query: SELECT * FROM V$ARCHIVE_DEST. This feature is useful when your environment includes a customized archive setup and you want to instruct Syniti Replicate use a specific location for archived logs.

Agent Settings Screen

Use Log Container

If your data source is an Oracle version 12 pluggable non-root database, you also need to complete the root database connection information:

Server, User, Password

The connection string for the root database. This is required because Oracle 12 does not keep redo log records for the entire database instance separately.

Use Remote LSA

This option is available for environments where it is preferable to read the Oracle log directly and involves the installation and configuration of an additional component on the Oracle server. Contact the technical support team via the Help Center for additional information.

Create Replication Wizard

After creating a connection, and setting up for transactional replications, create the replication. The following fields require specific information for Oracle.

Source Log Info

Service Name

This is a unique identifier for your Oracle Server. The value is obtained automatically from your Oracle database server and cannot be modified.

Dictionary File

If using Oracle 9 or later, you can leave this field blank to use the online dictionary. If you prefer to supply a dictionary file, provide a path and dictionary file name on your Oracle database server. The online dictionary/dictionary file contains table information that is used in replication.

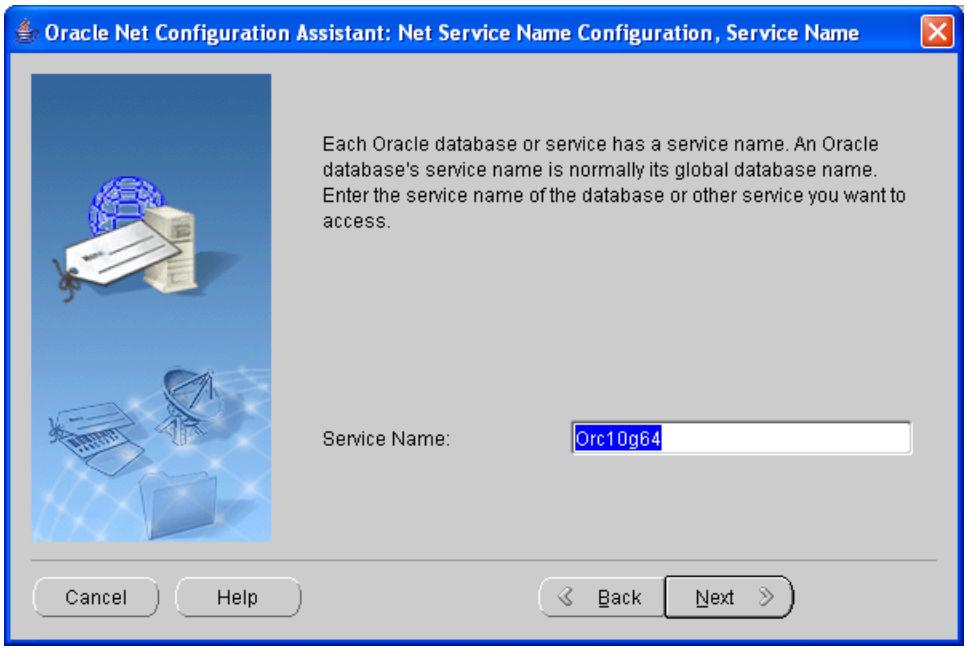
Configure the Oracle Client

Oracle 11 and above Client installation does not require extra steps to set up the Net Service Name for Oracle. For this reason, using an Oracle 11 or above client is recommended. 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.

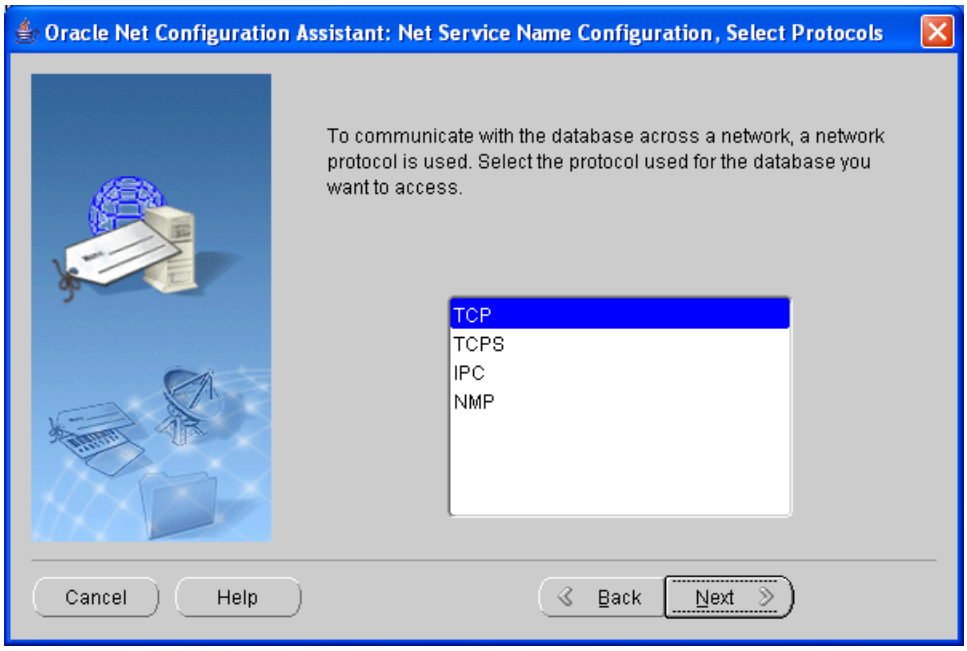
1. Run the installation process for the Oracle client that contains the Oracle .NET Provider.

   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.

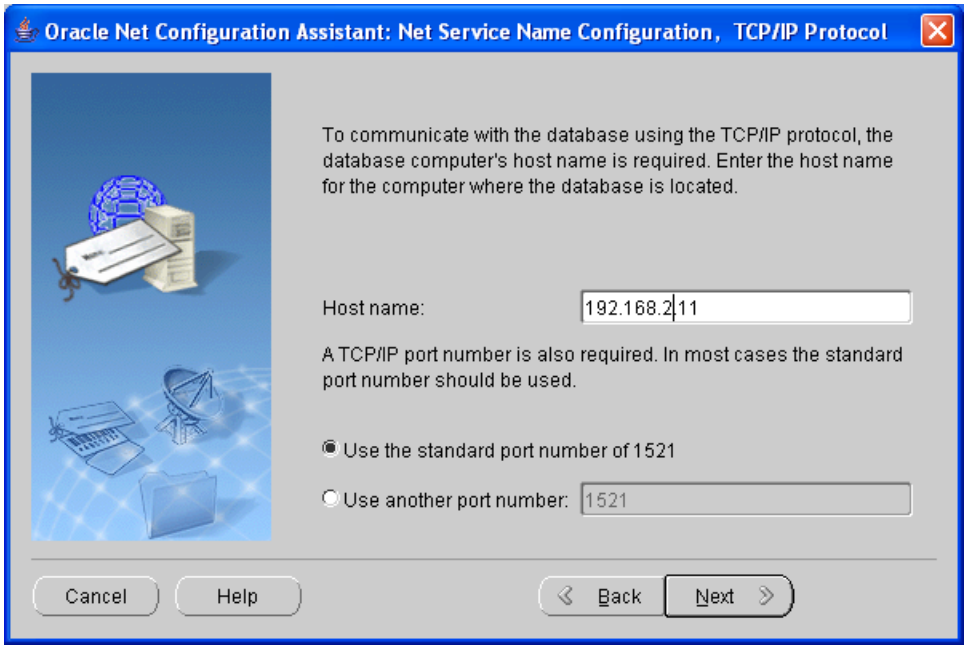
2. Enter the service name and click **Next**.



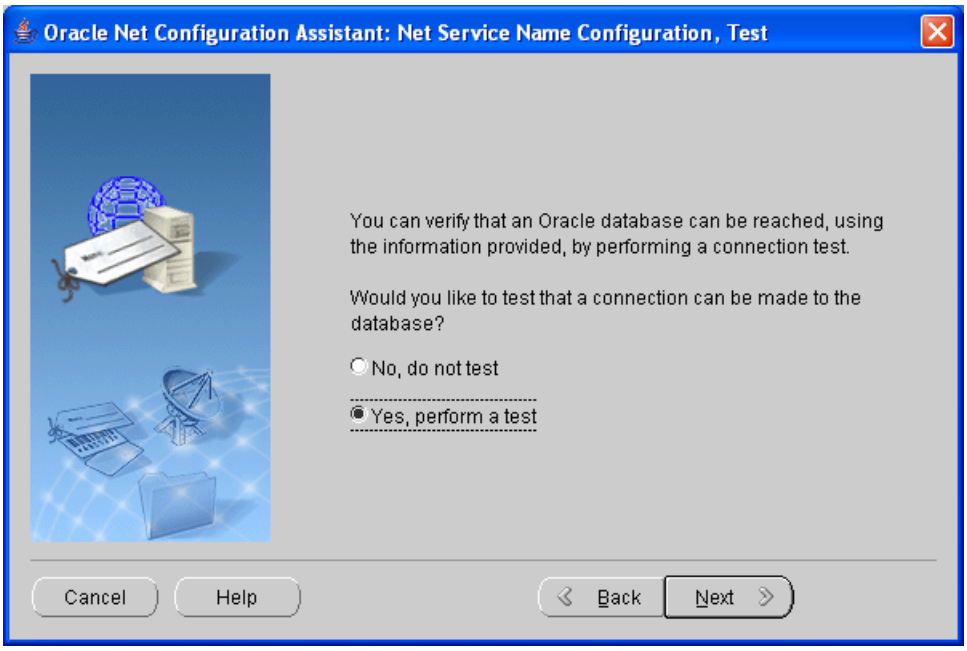
3. For the ACCESS_LOG network protocol, select **TCP**.



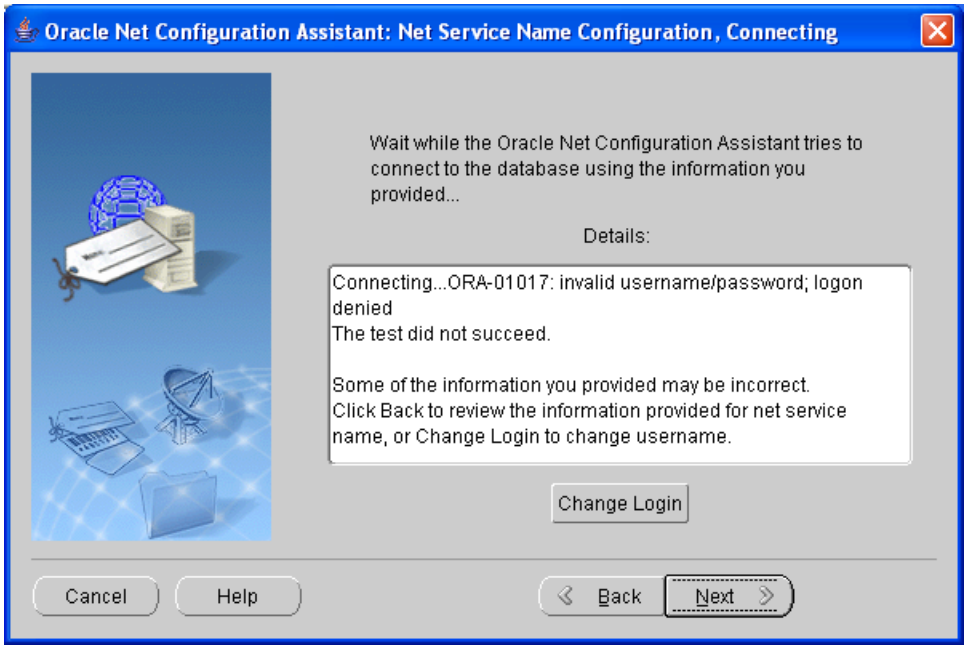
4. Enter the IP address of your Oracle system and click **Next**.



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

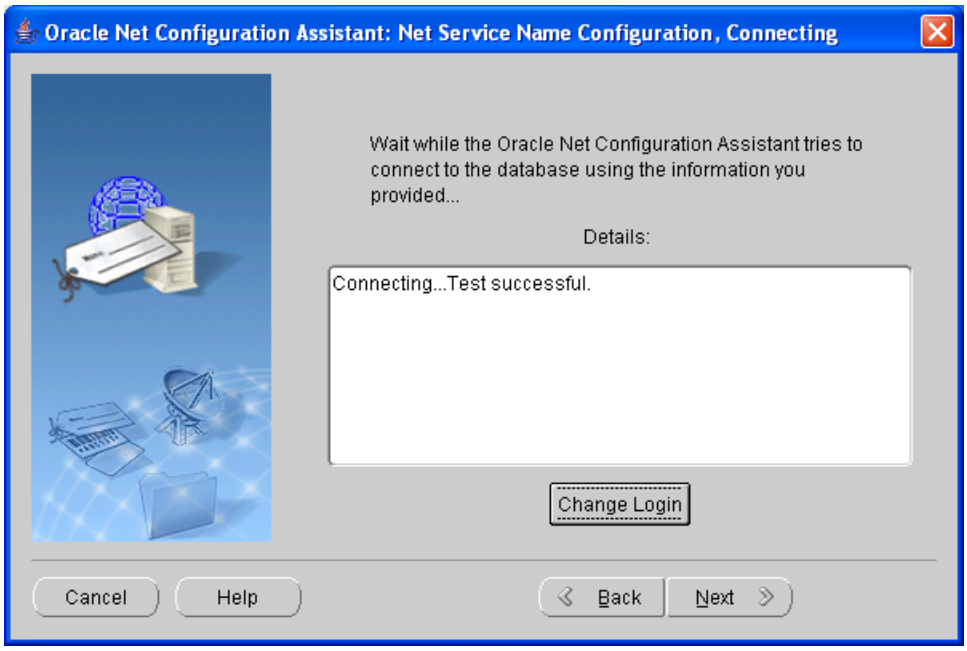


6. The test may fail because you do not have the correct login and password information.

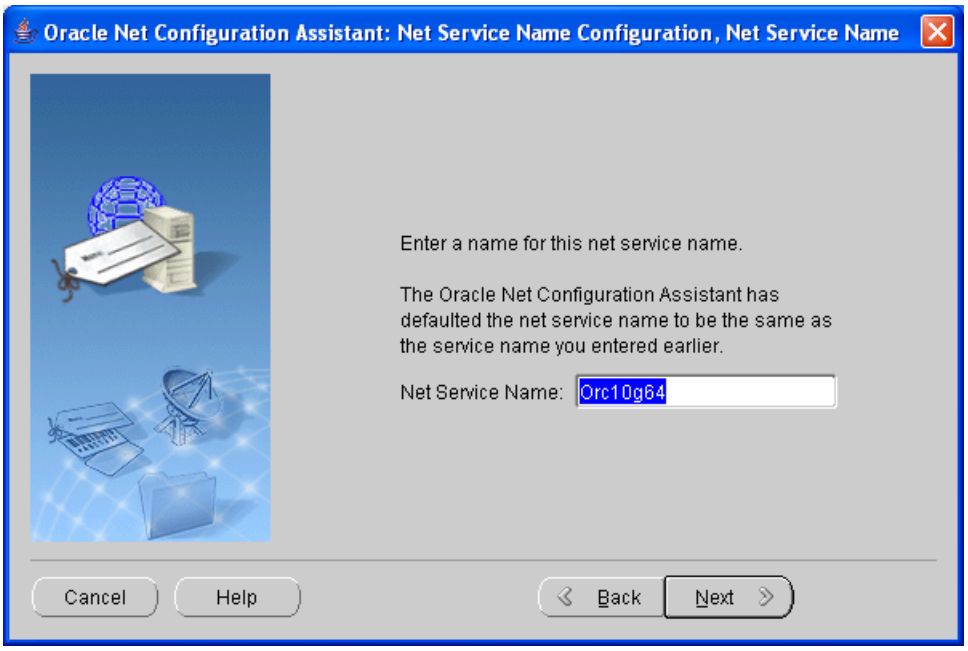


7. Click **Change Login**.

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



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



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