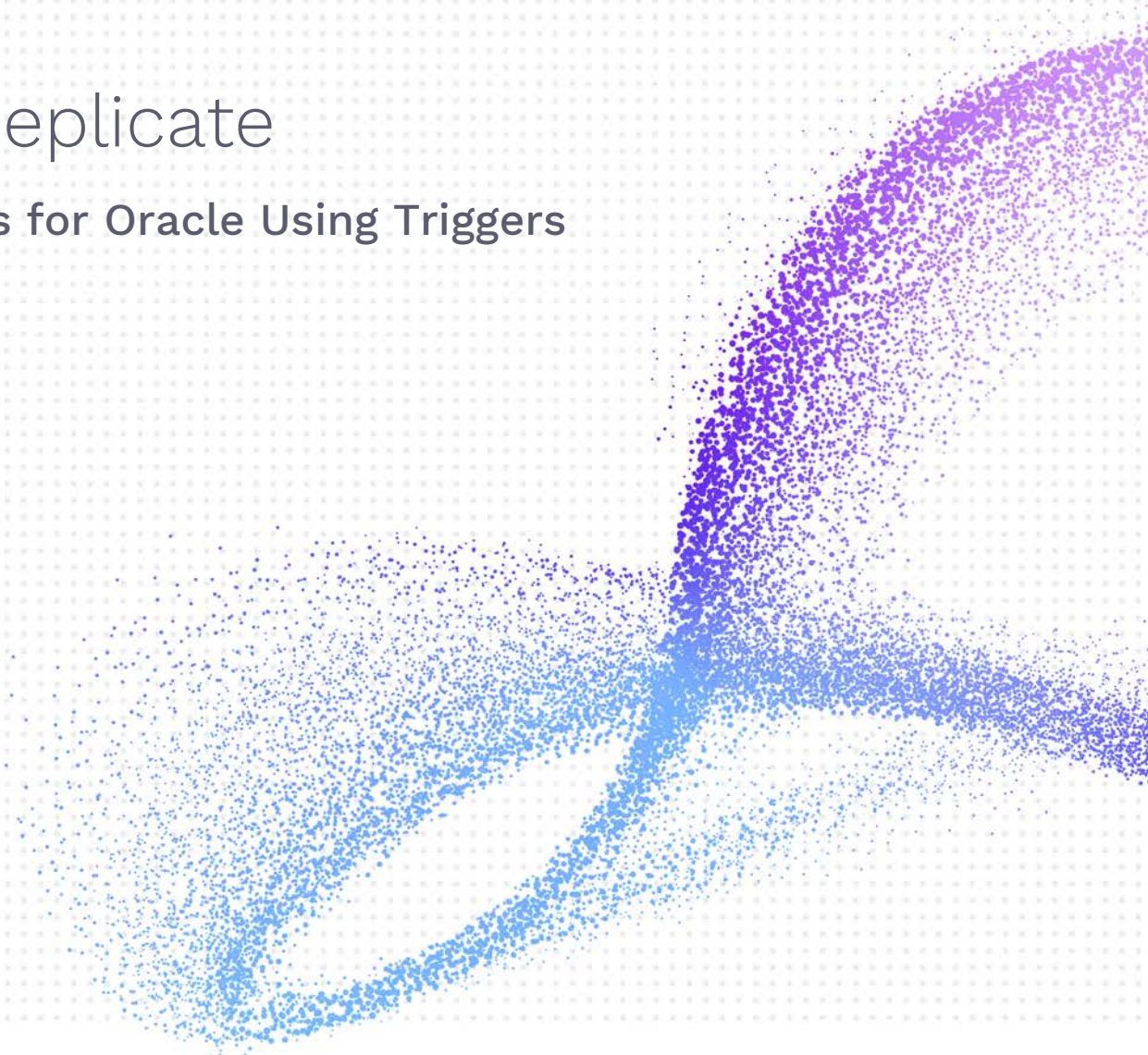




# Syniti Replicate

## Setup Notes for Oracle Using Triggers

Version 10.1



## Table of Contents

Connection Type.....	1
Trigger-Based Replication Overview.....	1
Syniti Replicate Log Tables.....	2
Master table structure.....	2
Log Table Structure.....	3
Reading From/Managing Log Table.....	4
Oracle User Permissions for Trigger-based Replications.....	4
Refresh with Oracle as Either Source or Target Database.....	4
Transactional Replications/Initial Refresh with Oracle as Either Source or Target Database.....	5
Oracle System Settings.....	7
Install Oracle Client with .NET Provider.....	7
Add a Source Connection Wizard.....	8
Select Provider Screen.....	8
Set Connection String Screen.....	8
Enable Transactional Replication Wizard.....	8
Log Type Screen.....	8
Trigger Settings Screen.....	8
Configure the Oracle Client.....	10

# 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.

**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, Syniti Replicate 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 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 triggers installed on the Oracle database to log changes. This approach may be preferred in your application.

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](#).

## 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, Syniti Replicate 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 Syniti Replicate. 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 Syniti Replicate 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_OBJ`.

# Syniti Replicate

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

```
CREATE TABLESPACE SYNITIDR
```

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 Replication Agent 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. Syniti Replicate may report errors about locked tables during replication. Although Syniti Replicate 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 “Syniti Replicate 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.

## Syniti Replicate 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 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 NUMBER  
GENERATED BY DEFAULT AS IDENTITY  
Transaction ID number associated with each record data change (transaction)

Copyright © 2022 Back Office Associates, LLC d/b/a Syniti. All rights reserved. This document contains confidential and proprietary information and reproduction is prohibited unless authorized by Syniti. Names appearing within the product manuals may be trademarks of their respective owners.

# Syniti Replicate

- SNAME VARCHAR  
Name of the schema the transaction was applied to.
- TNAME VARCHAR  
Name of the table the transaction was applied to.
- TRXID VARCHAR  
Transaction XID of the current Oracle transaction
- TTS TIMESTAMP  
Transaction timestamp indicating when the transaction was submitted to the system
- TUSER VARCHAR  
Name of the user who executed the transaction
- FLAG NUMBER  
Numeric flag used to mark a transaction as processed by the log reader

## 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 NUMBER  
Transaction ID, link to the corresponding record in the master table
- \_\_ID NUMBER  
Record id within the log table
- \_\_OP CHAR  
Indicates the type of operation:  
  
‘I’ INSERT  
‘D’ DELETE  
‘B’ UPDATE: values before the update  
‘A’ UPDATE: values after the update
- \_\_FLAG NUMBER  
Numeric flag used to mark a transaction as processed by the log reader
- <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:

Copyright © 2022 Back Office Associates, LLC d/b/a Syniti. All rights reserved. This document contains confidential and proprietary information and reproduction is prohibited unless authorized by Syniti. Names appearing within the product manuals may be trademarks of their respective owners.



# Syniti Replicate

```
CREATE TABLE __DBM__SOURCET
(__TID NUMBER(19),
 __OP CHAR,
 ID INTEGER,
 FNAME VARCHAR(30),
 LNAME VARCHAR(50))
```

## Reading From/Managing Log Table

Syniti Replicate 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, Syniti Replicate 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. Syniti Replicate also uses the unique \_\_TID column to keep track of the point where the last record was read and processed from the log tables.

Syniti Replicate 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 for Trigger-based Replications

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>;
```

# Syniti Replicate

```
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;
```

### 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;
```

## 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

The recommended approach is to assign a tablespace for the Master table and Log Tables so that it is easier to control log table sizes. Note that Syniti Replicate does not create a tablespace, but the user can specify access to a tablespace. If you want to have a table space named SYNITIDR, you must create it beforehand:

```
create tablespace <tablespace> datafile <data_file> size <size>
```

Example where SYNITIDR\_TBLSPACE is the name of the tablespace and synitidr\_tblspace.dat is the datafile with an initial allocation of 100 Mb:

```
create tablespace SYNITIDR_TBLSPACE datafile 'synitidr_tblspace.dat' size 100M online;
```

### 2. CREATE USER FOR SYNITI REPLICATE OPERATIONS AND TABLESPACE PERMISSIONS

The DBA should create a user for Syniti Replicate 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 Syniti Replicate: permissions to connect, select tables, insert/update/delete records and so on. Once a tablespace has been created, you can decide whether to grant unlimited quota to the user:

```
alter user <UID> quota unlimited on <TBLSPACE>;
```

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

# Syniti Replicate

```
alter user <UID> QUOTA 100M on <TBLSPACE>;
```

Example where SYNITIDR\_TBLSPACE is the name of the tablespace and SDRUSER is the name of the user.

```
alter user SDRUSER quota unlimited on SYNITIDR_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 sdruser is the user ID:

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

OR

```
grant connect to sdruser;
```

### 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), 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;
```

### 5. 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;
```

### 6. SET UP TRANSACTIONS (triggers)

To set up transactional replications, you need permission to create and drop triggers in the schema where the \_DBM\_\_TRG\_OBJ\$ table resides:



# Syniti Replicate

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

Example:

```
grant create any trigger to SDRUSER;
grant drop any trigger to SDRUSER;
grant select any trigger to SDRUSER;
```

## 7. ACCESS TRIGGER CONFIG TABLE

You need to access the trigger configuration 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 SDRUSER:

```
grant select on SCOTT."_DBM__TRG_OBJS" to SDRUSER;
```

## 8. MANAGE SEQUENCES

```
grant select on SYS.ALL_TABLES to <UID>;
grant select on SYS.ALL_TRIGGERS to <UID>;
grant select on SYS.ALL_SEQUENCES to <UID>;
grant create any index to <UID>;
grant create any sequence to <UID>;
grant drop any sequence to <UID>;
grant select any sequence to <UID>;
```

Example:

```
grant select on SYS.ALL_TABLES to SDRUSER;
grant select on SYS.ALL_TRIGGERS to SDRUSER;
grant select on SYS.ALL_SEQUENCES to SDRUSER;
grant create any index to SDRUSER;
grant create any sequence to SDRUSER;
grant drop any sequence to SDRUSER;
grant select any sequence to SDRUSER;
```

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 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.

Copyright © 2022 Back Office Associates, LLC d/b/a Syniti. All rights reserved. This document contains confidential and proprietary information and reproduction is prohibited unless authorized by Syniti. Names appearing within the product manuals may be trademarks of their respective owners.

# Syniti Replicate

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.

## 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 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 Syniti Replicate and the names are `_DBM__LOG_#`, where # is a number. The selected schema for the Master and Log tables must not contain other non-Syniti tables with names `_DBM__LOG_#`. The recommended approach is to create a new schema to use specifically for the Syniti Master and Log tables.

## Tablespace

The recommended approach is to 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 Syniti Replicate 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, Syniti Replicate deletes items from the log. This field specifies the maximum number of records to delete from the Syniti Replicate 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.

## Uncommitted Transactions Recovery Option

This section can be used to indicate how uncommitted transactions should be handled during replication.

<b>None</b>	No gap conditions handled. Uncommitted transactions may cause data to be skipped in replication.
<b>Wait</b>	Set a number of mirroring intervals to hold all replications on the connection to wait when a gap is found due to an uncommitted transaction. All replications in the connection will hold and wait for the number of cycles specified to see if the gap is filled. If after the number of cycles, a transaction is still not committed, it will be skipped.
<b>Continue and Process when Committed</b>	Instead of pausing all replications in case of a gap, replication proceeds with all currently committed transactions. During the next mirroring cycle, the trigger log table is checked for earlier

transactions that now have been committed, and any identified transactions are processed.

## Skip Uncommitted Transactions Older Than

If wait or continue are selected, this property sets a limit on the amount of time to wait for uncommitted transactions. For instance, a value of 15 minutes means that, no matter which option you choose, transactions opened and not committed for more than 15 minutes will be skipped.

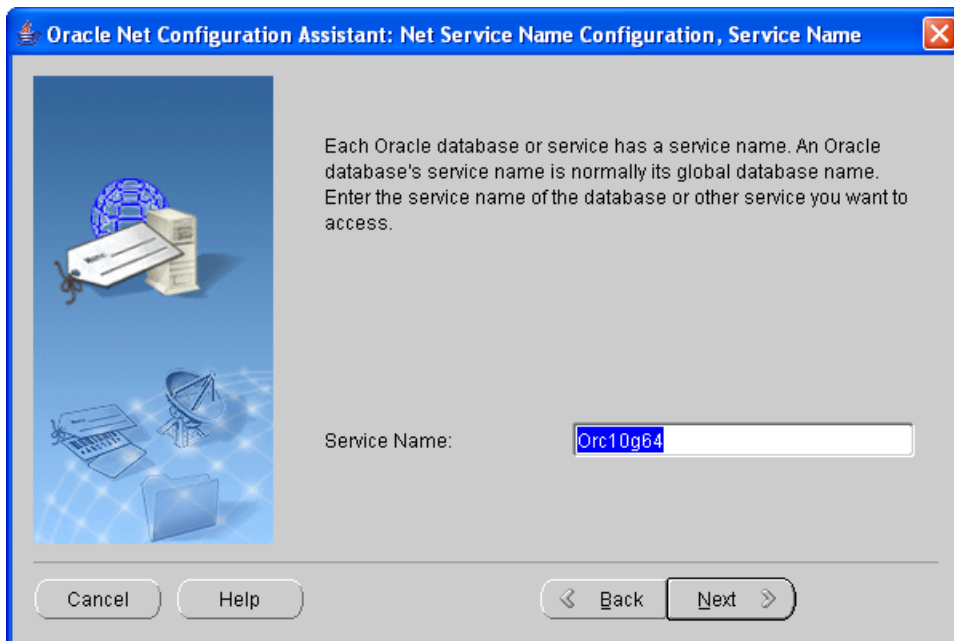
## Configure the Oracle Client

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.

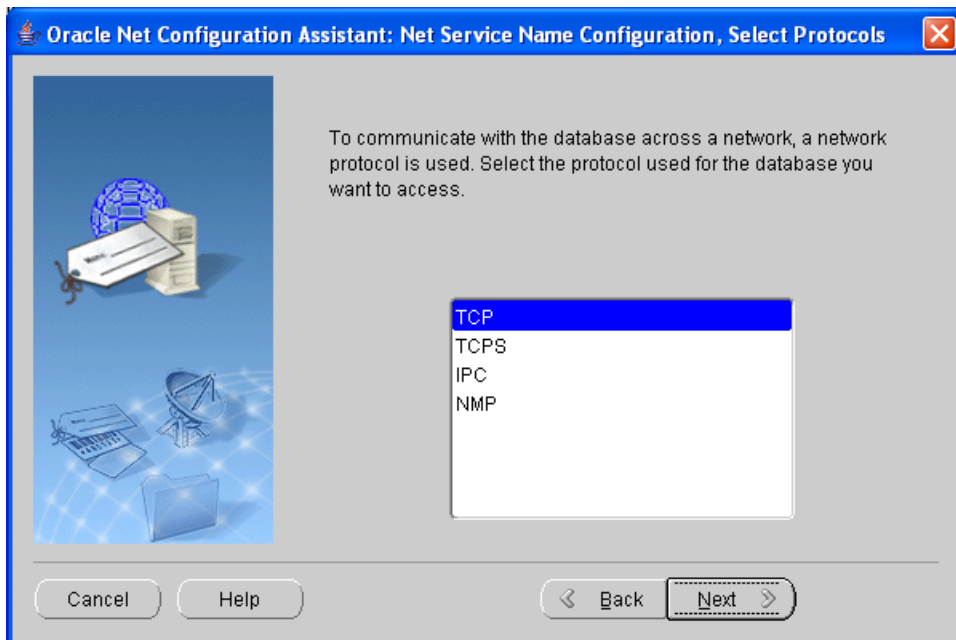
The Oracle Net Configuration Assistant starts up automatically after you install the Oracle client. You can also start it manually from the Windows Start menu: All Programs -> Oracle - OraClient10g\_homes1 -> Configuration and Migration Tools -> Net Configuration Assistant.

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

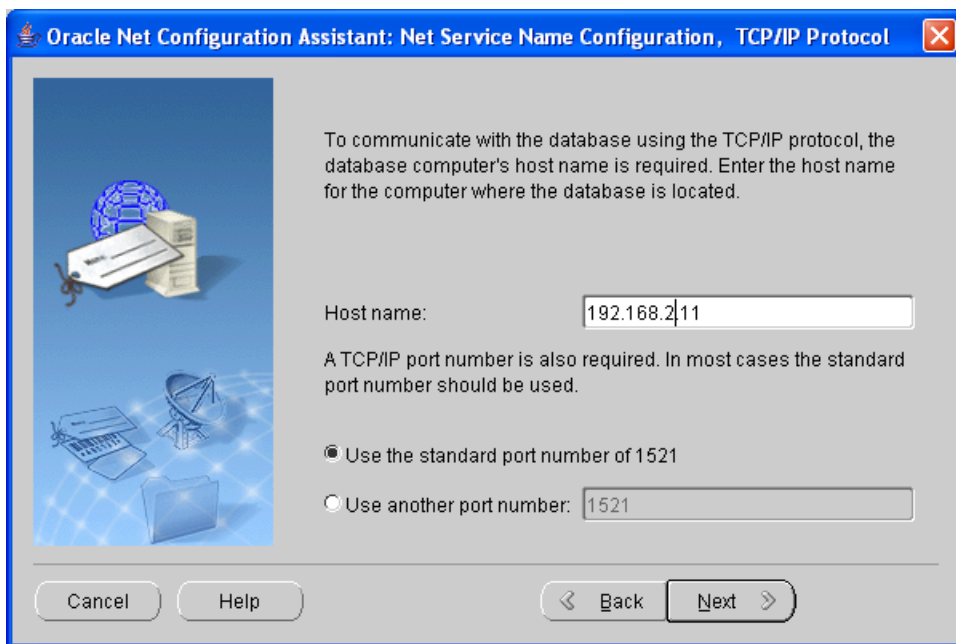


3. For the ACCESS\_LOG network protocol, select **TCP**.

# Syniti Replicate



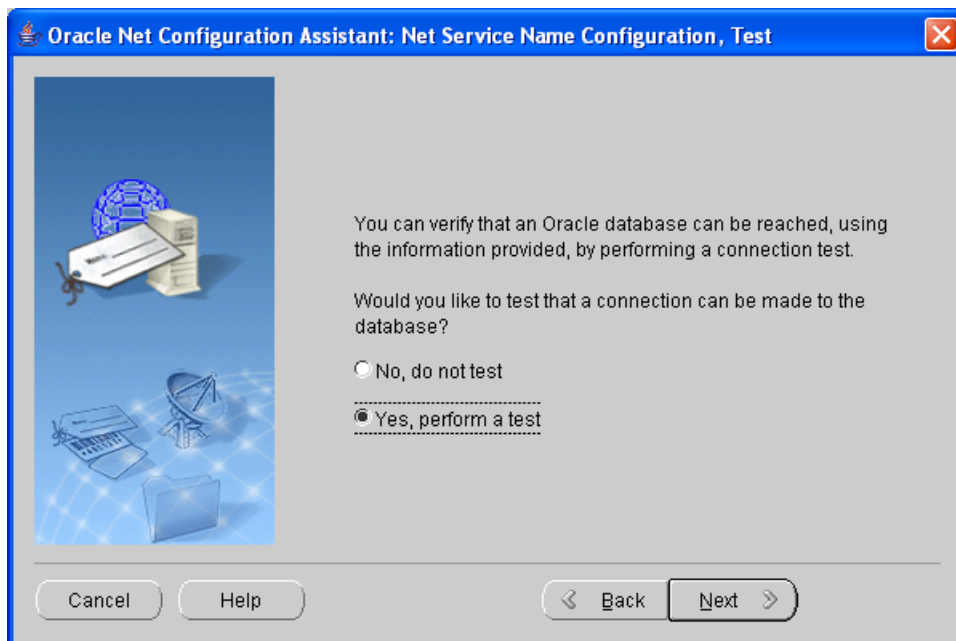
4. Enter the IP address of your Oracle system and click **Next**.



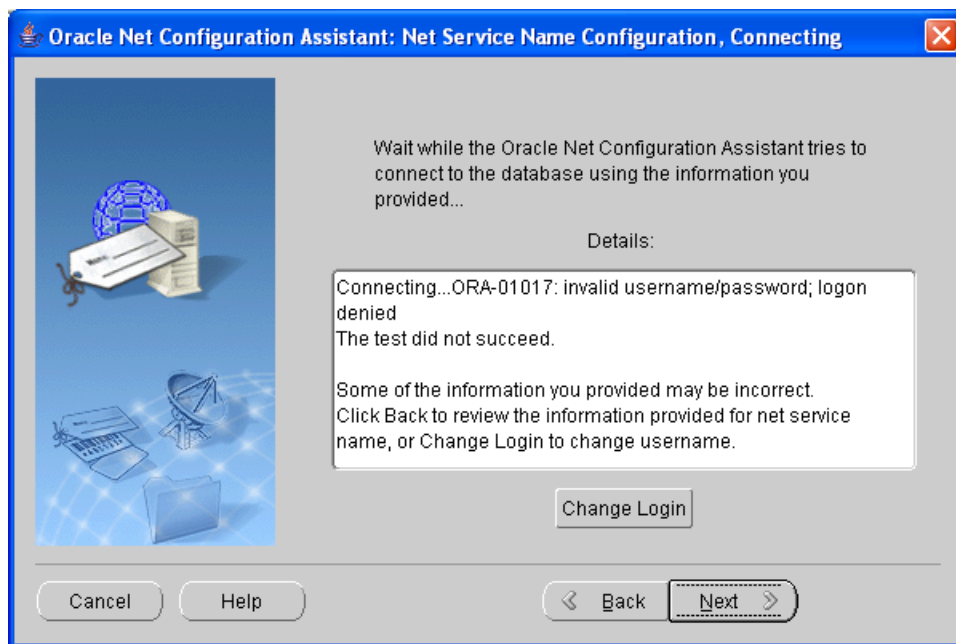


# Syniti Replicate

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

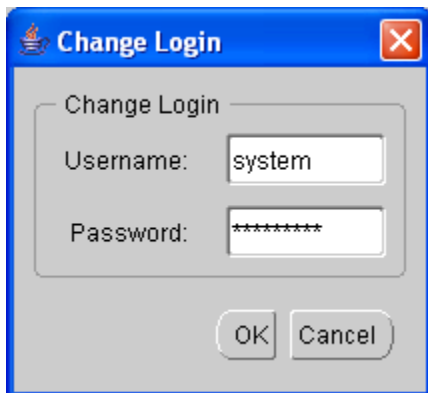


6. The test may fail because you do not have the correct login and password information.

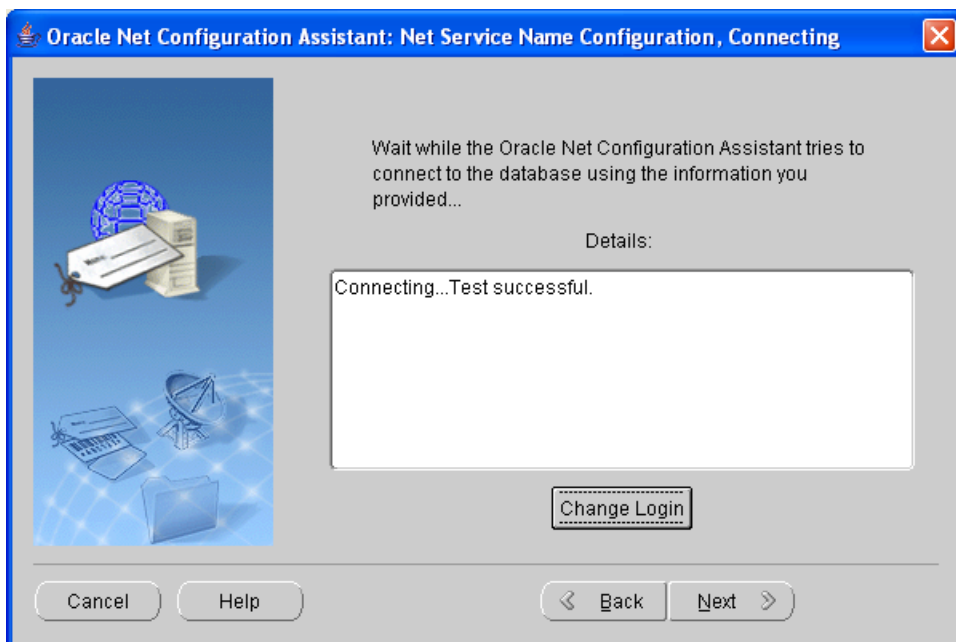


7. Click **Change Login**.

# Syniti Replicate



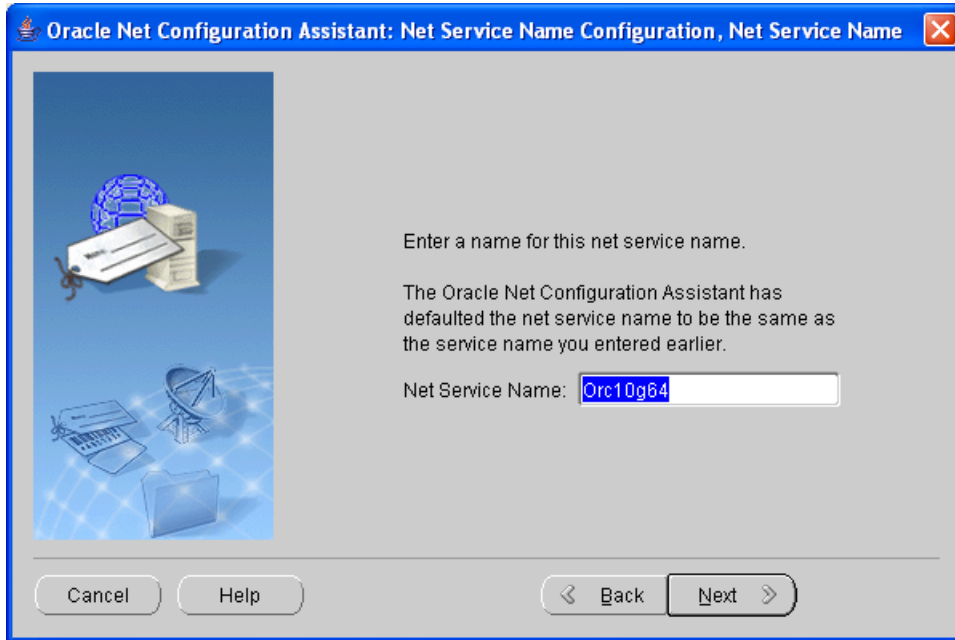
8. Enter the user ID and password of your Oracle system, then click **OK**.



9. When the test is successful, click **Next**.

10. Either accept the default for the NET service name, or enter a new service name.

# Syniti Replicate



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

