Syniti Replicate

Setup Notes for Replicating from SAP® HANA®

Version 10.3

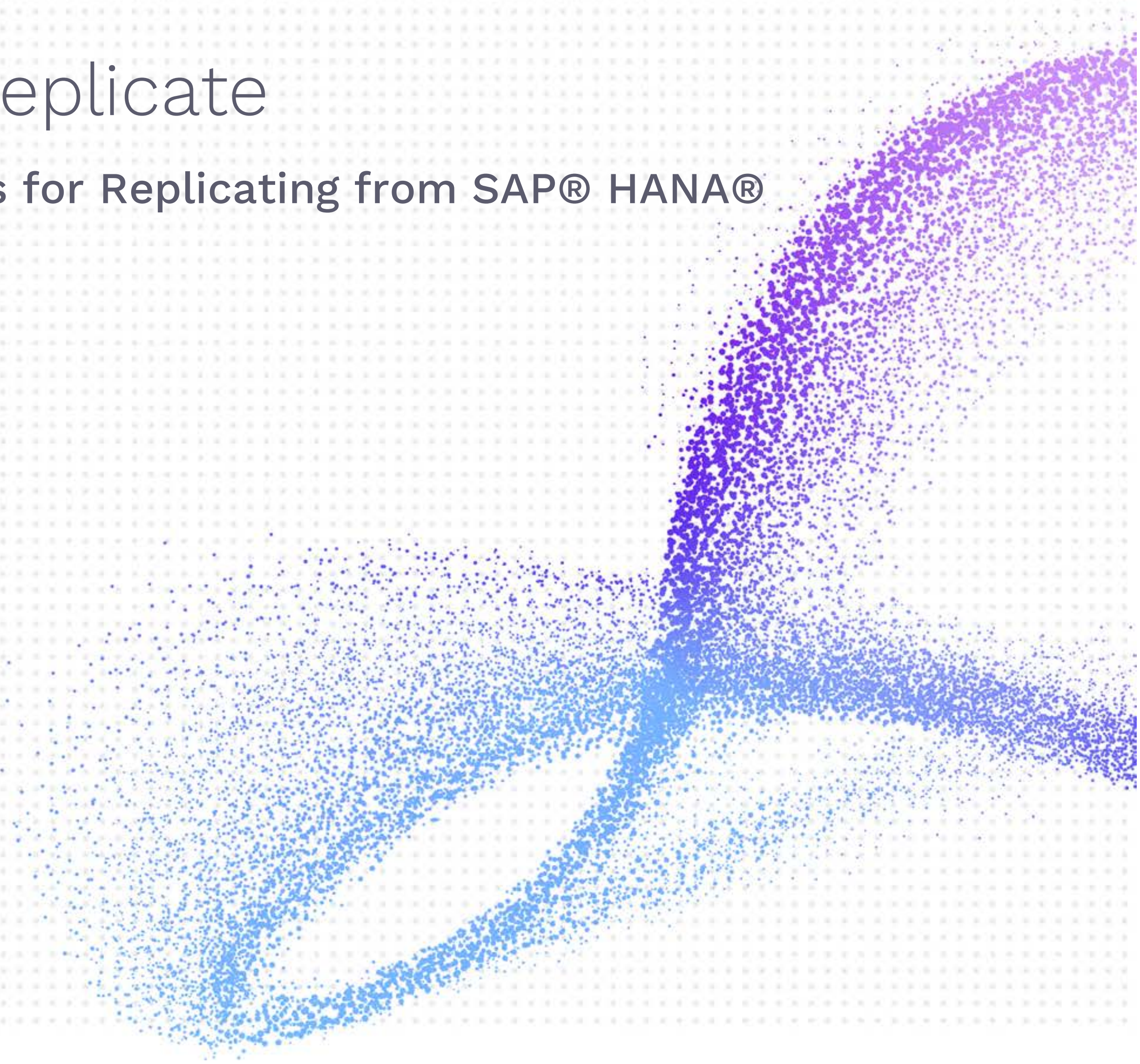


Table of Contents

Syniti Replicate

These notes provide essential information for setting up replications using **SAP HANA** 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 logs. For complete details on the setup process, check the *Syniti Replicate User Guide* available from the Management Center **Help** menu or the *Setup Guide*, available for download in the Help Center.

This guide describes the setup process using the Triggers option for one-way mirroring and synchronization when replicating data from an SAP HANA database. For mirroring replications using SAP HANA as a source, Syniti Replicate offers the following approach:

- **Triggers**: Uses custom triggers installed on the SAP HANA database to log changes.

For mirroring replications using SAP HANA as a target, see *Setup Notes for SAP® HANA® Transactional Replications (Target)* in the Help Center.

Connection Type

One of the following:

- SAP HANA .NET Data Provider version 1.0.84.0 or above (Supported in DBMoto 8.5 and above or Syniti Replicate 9.6 and above)
    1. Log in and register your account: https://community.sap.com/topics/hana-data-management-suite?sourceid=PPCDevelopers2&url_id=text-crm-xh12-ppc-ppcdbdg4-developeredition
    2. Search for “Software Downloads”
       SAP HANA CLIENT Winx86 (32 or 64 bit)

       Assembly: Usually located in C:\Program Files\sap\hdbclient\ado.net\v4.5\Sap.Data.Hana.v4.5.dll
- SAP HANA ODBC Driver

  Available as part of the SAP HANA CLIENT Winx86 (32 or 64 bit)

  SAP HANA CLIENT Winx86 (32 or 64 bit)

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_OBJS.

Note that SAP HANA does not use tablespaces, but you can define a schema for the master and log tables. If you create a new user, the database will define automatically a schema with the same name as the user; alternatively, you can create a new schema using the command: CREATE SCHEMA 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 (Data Replicator 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. 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 by Syniti Replicate 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 BIGINT GENERATED BY DEFAULT AS IDENTITY:
  Transaction ID number associated with each record data change (transaction)

- SNAME VARCHAR(64):
  Name of the schema the transaction was applied to.

- TNAME VARCHAR(64):
  Name of the table the transaction was applied to.

- TTS TIMESTAMP:
  Transaction timestamp indicating when the transaction was submitted to the system

- TUSER VARCHAR(81):
  Name of the user who executed the transaction
- FLAG TINYINT
  Numeric flag used to mark a transaction as processed by the log reader
- PRIMARY KEY (TID)

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 BIGINT:
  Transaction ID, link to the corresponding record in the master table

- __OP VARCHAR(1):
  Indicates the type of operation:

  'I' INSERT
  'D' DELETE
  'B' UPDATE: values before the update
  'A' UPDATE: values after the update

- __FLAG TINYINT
  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:

```
CREATE TABLE _DBM__SOURCET
(__TID BIGINT,
 __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.

SAP HANA User Permissions

When setting up replications that use SAP HANA as a source 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 Replications

1. AUTHORITY TO CONNECT TO DATABASE
   This should already be granted when the user is created.

2. 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. The command is not needed if the user is the owner of the table (i.e., created the table.)

Transactional Replications/Initial Refresh

This section includes information on the minimum permissions needed to set up and run mirroring where triggers are used to identify changed data. For precise details on syntax for SAP HANA, check the following resources:

https://help.sap.com/doc/4fe29514fd584807ac9f2a04f6754767/2.0.02/en-US/20f674e1751910148a8b990d33efbdc5.html

1. CREATE USER FOR SYNITI REPLICATE OPERATIONS
   The DBA should create a user for Syniti Replicate operations on the machine. The 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.

2. 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. The command is not needed if the user is the owner of the table (i.e., created the table.)

3. AUTHORITY TO CREATE AND DROP TRIGGERS
   To set up transactional replications, you need permission to create triggers in the schema where the source data table resides.

4. AUTHORITY TO CREATE AND DROP SEQUENCES
   You need permission to create sequences as part of the setup of the Syniti Replicate master table.

5. AUTHORITY FOR AN EXTERNAL USER TO WRITE TO THE TRIGGER SCHEMA
   When triggers are executed (following an INSERT, UPDATE or DELETE by an external user), the external user ID requires write access to the trigger schema (which contains the master table and the log tables.)

   ```
   GRANT INSERT ON SCHEMA <trigger schema> TO <external user>
   GRANT SELECT ON <sequence_name> TO <external user>
   ```

   where:

   <trigger schema> is the schema used for the master table

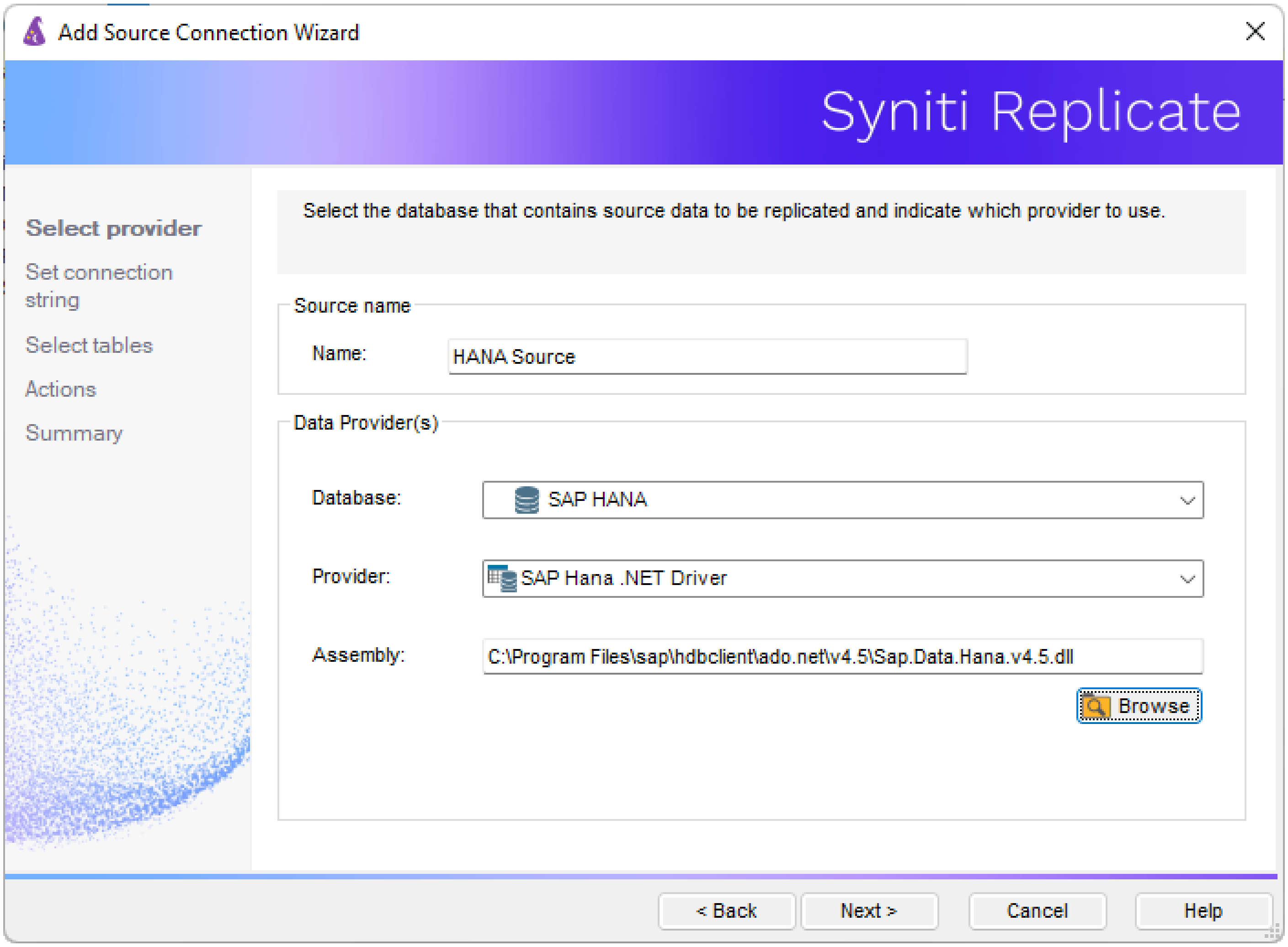
   <external user> is any possible user that will run DML commands to the replicated tables.

   <sequence_name> is the sequence defined with the master table. The sequence name is generated from the master table name using the suffix "_SEQ": <master name>_SEQ.

Add Source Connection Wizard

The following field(s) require specific information for SAP HANA.

Select Provider Screen



Add Source Connection Wizard

Syniti Replicate

Select the database that contains source data to be replicated and indicate which provider to use.

Source name

Name: HANA Source

Data Provider(s)

Database: SAP HANA

Provider: SAP Hana .NET Driver

Assembly: C:\Program Files\sap\hdbclient\ado.net\v4.5\Sap.Data.Hana.v4.5.dll

Database

Choose SAP HANA from the drop-down list.

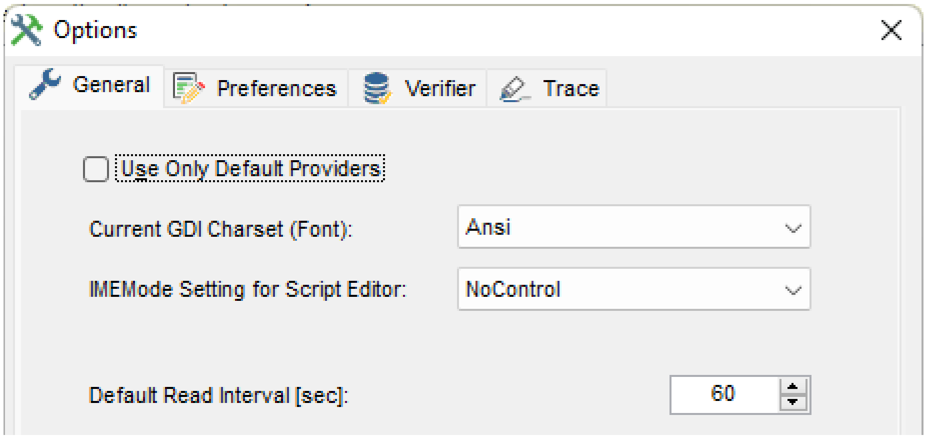
Provider

The default value is SAP Hana .NET Driver. However, an ODBC driver is also available.

To access the ODBC driver for SAP HANA:

1. Exit the Add Source Connection wizard.
2. On the **Tools** menu in the Management Center, click **Options.**
3. In the **Options** dialog, uncheck the **Use Only Default Providers** option.

4. Click **OK** to save the change.
5. Restart the Add Source Connection wizard.



□ Use Only Default Providers

Current GDI Charset (Font): Ansi

IMEMode Setting for Script Editor: NoControl

Default Read Interval [sec]: 60

Assembly

(For .NET Data Provider connections only)

If no value is provided in the Assembly field, use the **Browse** button to locate the Sap.Data.Hana DLL. It is typically located in:
C:\Program Files\sap\hdbclient\ado.net\v4.5\Sap.Data.Hana.v4.5.dll

NOTE: The version number of your SAP HANA .NET Data Provider may be different. Syniti Replicate supports SAP HANA .NET Data Provider version 1.0.84.0 or above.

Set Connection String Screen

Data Source

The IP address of your database server followed by a colon and the server port number.

IMPORTANT: The server name and port number should follow the format <server>:<port>. If you do not specify the port number, the default is used and the default is often not appropriate for standard SQL communications via client access. The format of the port for **standard SQL communication via client access is 3<instance-number>15** where <instance number> is the instance number of the SAP HANA database engine. For example, for instance 00, the port number would be 30015.

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 SAP HANA.

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 Replicate tables with names _DBM__LOG_#. You should create a new schema to use specifically for the Syniti Replicate Master and Log tables.

Tablespace

SAP HANA does not use tablespaces.

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 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 SAP HANA sources.

Uncommitted Transactions Recovery Option

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

None
No gap conditions handled. Uncommitted transactions may cause data to be skipped in replication.

Wait
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.

Continue and Process when Committed
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.