



Data Replication

Setup Notes Progress OpenEdge Replications

Version 9.7.2.26

Software Release Date: 1/15/2021



<https://support.hitsw.com>

Document History

Version	Author	Date	Reviewer	Date	Comments
1	SUpadhaya	12/29/20	VFarruggio	01/07/21	

Syniti Data Replication

Table of Contents

Connection Type	1
Trigger-Based Replication Overview	1
Syniti DR Log Tables	2
Master table structure	2
Log Table Structure.....	3
Reading from/Managing Log Table	3
User Permissions for Replications with OpenEdge.....	4
Refresh with OpenEdge as Either Source or Target Database.....	4
Transactional Replications with OpenEdge as Source	4
Additional Requirements/Limitations for OpenEdge.....	5
Add Target/Source Connection Wizard.....	5
Select Provider Screen	6
Set Connection String Screen	6
Connection Properties Dialog.....	7

Syniti Data Replication

These notes provide essential information for setting up replications using **Progress OpenEdge**.

This guide describes the setup process using the Triggers option for one-way mirroring when replicating data from an OpenEdge database. Triggers are installed on the OpenEdge database to log changes. Syniti Data Replication also supports replicating from a relational database (via Refresh or Mirroring) to an OpenEdge target.

For complete details on the setup process, check the *Syniti Data Replication User Guide* available from the Management Center **Help** menu or the *Syniti Data Replication Setup Guide*, available for download in the [Help Center](#).

Connection Type

Progress DataDirect 7.1 Wire Protocol ODBC driver for OpenEdge should be installed.

Download the Windows-64 bit or 32-bit version from:

<https://www.progress.com/datadirect-connectors>

Assembly: DDOE27 (file name: DDOE27.DLL)

Sample Path: C:\Program Files\Progress\DataDirect\Connect64_for_ODBC_71\Drivers\ DDOE27.DLL

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 Connection wizards so that triggers can be created to log table changes for replication.

For each table involved in the replication, Syniti DR creates three java-based triggers on 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 Data Replication. 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 connection. The master log table can be an existing table or one created specifically to hold Syniti DR 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`.

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

Copyright © 2021 BackOffice 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 Data Replication

if there are many transactions using the same master log and log tables. Syniti DR may report errors about locked tables during replication. Although Syniti DR 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 DR 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 DR 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 BIGINT`:
Transaction ID number associated with each record data change (transaction)
- `SNAME VARCHAR(256)` :
Name of the schema the transaction was applied to.
- `TNAME VARCHAR(256)` :
Name of the table the transaction was applied to.
- `TTS TIMESTAMP` :
Transaction timestamp indicating when the transaction was submitted to the system
- `TUSER VARCHAR(256)` :
Name of the user who executed the transaction

Syniti Data Replication

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 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 structure as:

```
CREATE TABLE __DBM__LOG_x
(__TID BIGINT),
__OP CHAR(1),
ID INTEGER,
FNAME VARCHAR(30),
LNAME VARCHAR(50))
```

Reading from/Managing Log Table

Syniti DR 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 DR 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 DR 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 DR 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.

User Permissions for Replications with OpenEdge

When setting up replications that use OpenEdge 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 DR 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 OpenEdge as Either Source or Target Database

1. AUTHORITY TO SELECT TABLES

Syniti DR runs a SELECT statement to count records in target table(when used as target) or to identify records to replicate(when used as source). 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 PUB.EMPLOYEES to sdruser
```

2. AUTHORITY TO UPDATE TABLES

(Note: Needed only when OpenEdge is a target)

When OpenEdge is used as target in a refresh replication, the user should have INSERT, DELETE permissions on the target table involved in the replication.

```
grant insert on <table> to <UID>
```

```
grant delete on <table> to <UID>
```

Example where sdruser is the user ID:

```
grant insert on PUB.EMPLOYEES to sdruser
```

```
grant delete on PUB.EMPLOYEES to sdruser
```

3. AUTHORITY TO CREATE TABLES (Optional)

(Note: Needed only when OpenEdge is a target)

To be able to create a target table in the Management Center (as part of the Create Table Wizard), the user ID should have the RESOURCE privilege assigned to it.

Example where sdruser is the user ID:

```
grant RESOURCE to sdruser
```

Transactional Replications with OpenEdge as Source

This section includes information for mirroring where OpenEdge is the data source.

1. AUTHORITY TO SELECT TABLES

Syniti DR 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 PUB.EMPLOYEES to sdruser
```

In addition, the user should also have access to trigger config table, masterlog table and the log table (created for replication).

```
grant select on <SCHEMA>."_DBM__TRG_OBJS" to <UID>
```

Syniti Data Replication

```
grant select on <SCHEMA>."_DBM__MASTERLOG " to <UID>
```

Example where sdruser is the user ID:

```
grant select on PUB."_DBM__TRG_OBJS" to sdruser
grant select on PUB."_DBM__MASTERLOG" to sdruser
grant select on PUB."_DBM__LOG_x" to sdruser
```

2. AUTHORITY TO CREATE TABLES, UPDATE TABLES

To set up transactional replications, you need permissions to create and alter log tables; create and drop triggers; create,select and drop sequence. For that, dba or resource privilege must be granted to the user.

If resource privilege is granted, then the user should also have INSERT, UPDATE, DELETE permissions on the config table, masterlog table and the log table and SELECT privilege on the sequence created during transactional setup.

```
grant insert on <table> to <UID>
grant delete on <table> to <UID>
grant select on sequence <SCHEMA>."_DBM__MASTERLOG_SEQ" to <UID>
```

Example where sdruser is the user ID:

```
grant insert on PUB.EMPLOYEES to sdruser
grant delete on PUB.EMPLOYEES to sdruser
```

Additional Requirements/Limitations for OpenEdge

The OpenEdge database should be managed by the Admin Server to allow for remote connections.

- SQL clients should be able to connect to the OpenEdge database. When adding the OpenEdge database to the Admin Server's Managed Databases, a Port Number should be provided. This will be used in configuring a DSN for connecting to the database. Also, "Both" or "SQL" option should be selected for the "Database Broker Type".
- As OpenEdge triggers are Java code snippets, JAVA must be installed on the OpenEdge server and the server should be configured properly to point to the JDK installation. It is important to read the release notes for the Progress version being used as there may be specific requirements for a particular platform when creating and executing triggers like JAVA version etc.
- Refresh replications using BulkInsert mode for an OpenEdge target table do not support CLOB and BLOB datatypes.

Add Target/Source Connection Wizard

The following field(s) require specific information for OpenEdge.

Syniti Data Replication

Select Provider Screen

Add Target Connection Wizard

Syniti Data Replication

Select the database target where to replicate data and indicate which provider to use.

Target name

Name:

Data Provider(s)

Database:

Provider:

< Back Next > Cancel Help

Database

Choose Progress OpenEdge from the drop-down list.

Provider

The default value is Open Edge ODBC Driver.

Set Connection String Screen

Click in the **Connection String** value field to open the **Open Edge ODBC Connection Dialog**. Values entered in this dialog should reflect those needed to make a successful connection to an OpenEdge database via an ODBC connection string, with or without Syniti DR. There are no fields specific to Syniti DR.

Host Name

Specify the OpenEdge server's IP address or machine name.

Port Number

Type the port number that was provided when adding the database to the Admin Server's Managed Databases.

Database Name

Type the name of the OpenEdge database.

User Id

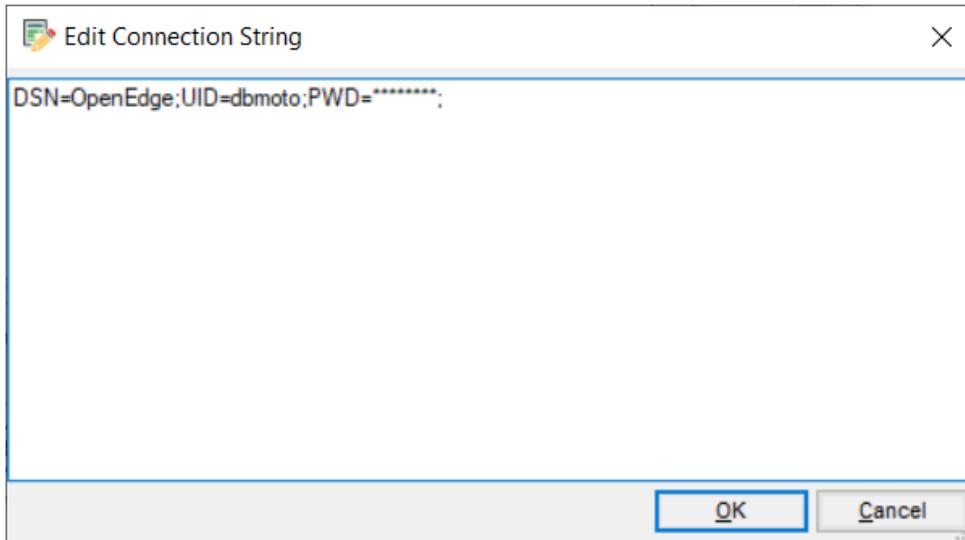
The user Id that will be used by Syniti DR to connect to the database.

Syniti Data Replication

Password

User's password.

If you have already configured a System DSN to connect to the OpenEdge database, click **Edit** at the bottom of the "Set Connection String" screen and provide the connection string that uses the DSN as shown below:



Connection Properties Dialog

The default insert mode for replicated data has been set as follows:

- Refresh replications – Single Insert
- Mirroring replications – Single Insert

To change these settings, and to complete setup for OpenEdge, you need to edit the connection properties for the connection created using the Create Target Connection wizard (above).

1. In the Metadata Explorer, locate the OpenEdge target connection you created.
2. From the right mouse button menu, choose **Connection Properties**.
3. Scroll to the **Refresh Options** section to change the Bulk Insert settings as needed. Single Insert is the default. Bulk Insert (with Array Binding) offers slight performance improvement over Single Insert and should be selected if better performance is required.
4. Scroll to the **Mirroring Options** section to change the Bulk Insert settings as needed. Bulk Insert is NOT set as the default but it offers better performance, so you may want to change the settings.

Syniti Data Replication

Target Connection Properties - OpenEdge

Connection properties for the target database.

General	
Connection Name	OpenEdge
Connection	Open Edge ODBC Driver

Dynamic Properties	
Max Number of Concurrent Connections	50
Command Timeout	60
Disconnected Mode	False
LOBs Cache Size (Kb)	1024
CREATE TABLE Prefix	
CREATE TABLE Postfix	
CREATE INDEX Postfix	
CREATE TABLE Custom Rule	Automatic

Refresh Options	
Default Insert Mode	SingleInsert
Default Bulk Type	ArrayBinding
Default Block Size	1000
Default Isolation Level	ReadCommitted

Mirroring Options	
Default Mirroring Insert Mode	SingleInsert
Default Mirroring Block Size	1000
Default Commit Isolation Level	ReadCommitted
Default Mapping Rule	MapByName

Transactional Support	
Transactional Setup	Disabled

Connection Name
Name assigned to the connection.

OK Apply Cancel

5. Click **OK** to save your changes.