Syniti Replicate

Volume & Tuning Guide for Refresh

Version 10.2

Table of Contents

Syniti Replicate

Refresh

Introduction

A refresh is one-time complete replication from source to target table, according to replication settings and scripts. You can control the timing of the replication, identify the columns to be replicated and add scripts to transform data during replication. The source and target databases can be on the same or different database servers and platforms. For example, you can replicate an Oracle or a HANA table to SQL Server.

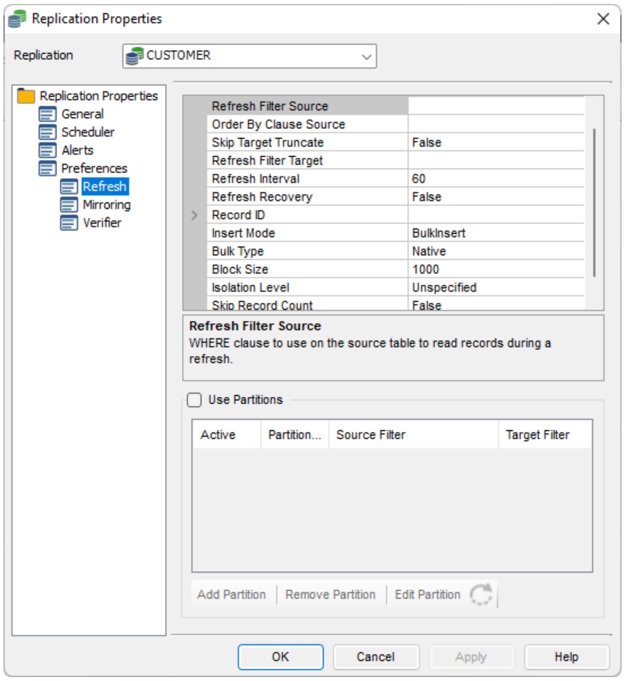
Replication Properties

You have two options to access the replication properties:

In the Metadata Explorer, select the replication. From the right mouse button menu, choose **Replication Properties**.

In the Replication Browser or Replication Monitor, select the replication. From the right mouse button menu, choose **Replication properties**.

Under Preferences, select **Refresh**



Replication: CUSTOMER

Refresh Filter Source	
Order By Clause Source	
Skip Target Truncate	False
Refresh Filter Target	
Refresh Interval	60
Refresh Recovery	False
Record ID	
Insert Mode	BulkInsert
Bulk Type	Native
Block Size	1000
Isolation Level	Unspecified
Skip Record Count	False

Refresh Filter Source
WHERE clause to use on the source table to read records during a refresh.

☐ Use Partitions

Active	Partition...	Source Filter	Target Filter

Refresh Filter Source: Allows you to specify a WHERE condition to apply during refresh applications to filter the records that are replicated. Click to open the Expression Generator and write a SQL expression to apply during the refresh replication.

Order By Clause Source: Provides access to the Expression Generator to create an expression for additional control over processing records.

Skip Target Truncate: The default setting is False and enables the truncation of a target table data before starting a refresh. Set to True to skip the truncation prior to starting the refresh.

Refresh Filter Target: Allows you to specify a WHERE condition to apply during the deletion of records on the target table prior to a refresh operation. Click to open the Expression Generator and write a SQL expression to apply during the refresh replication.

Refresh Interval: Default value is 60 seconds. Allows you to configure how long to wait before retrying a refresh in cases where the refresh replication was unable to start. The value must be expressed in seconds and greater than 0.

Refresh Recovery: When checked, and a refresh replication is interrupted while running, the Replication Agent is able to pick up the replication from the point at which it was interrupted. However, for this option to work, the source table involved in the replication must have one or more primary keys defined. If there are no primary keys, and the option is checked, a warning is printed in the log and recovery from the interruption is not possible.

Refresh will begin at the first record. If you are using a replication group, refresh recovery will start at the replication where the error occurred rather than starting at the first replication in the group. If you check this option, you may find that the replication runs more slowly. If performance is adversely affected, uncheck the option.

Record ID: Only valid when Refresh Recovery is set to True. Indicates the current record index to which the replication is set. The drop-down menu allows you to reset the index.

Refresh Staging: Not supported for all databases. Default value is False. Set to True to first populate a staging table with records from the source, then merge the content to the target table. This is useful for very large tables when an initial refresh truncate/delete is performed, but reading and writing the target data could leave the target database temporarily without data before the insert is performed. When Refresh Staging is set to true, all data is stored in the staging table and the data is merged to the actual target table after processing all source records: the target table is not emptied out and target database users are able to access data during the refresh process. The staging table is truncated after the refresh is complete. Here is a list of supported databases:

- IBM Db2 for iSeries
- IBM Db2 for z/System
- IBM Db2 LUW
- IBM Informix
- MariaDB
- MySQL
- Oracle
- MS SQL Server
- MS SQL Server CE
- MS Access
- MS SQLAzure
- PostgreSQL
- Sybase ASE

Insert Mode: The options are:

- SingleInsert--inserts record by record using SQL INSERT statements
- BulkInsert--inserts blocks of records. Block size is determined by the Block Size option. Choose this option for increased performance if using one of the data access providers which currently support bulk inserts:

  Syniti Ritmo/i (IBM Db2 for i) .NET Provider

  SAP HANA ODBC driver

  SAP HANA .NET Provider

  SAP Sybase ASE .NET Provider

  SAP Sybase IQ .NET Data Provider

  Microsoft SQL Server .NET Provider (using SQL Server's SqlBulkCopy)

  MySQL .NET Data Provider

  PostgreSQL .NET Data Provider

  Oracle .NET provider.

  For SQL Server, note that BulkInsert is the default option. Record details are not reported in the log when BulkInsert is set.
- SimulatedBulk--For databases that do not support bulk insertion, Syniti Replicate simulates the effect by grouping records into blocks before performing an insert. This is a way to optimize performance.

Bulk Type: Depending on the database type and provider type, some of the following options may not be available.

- ArrayBinding--inserts multiple rows at a time using an array of parameters
- Native--uses the native functionality of the provider, if available
- FTP--bulk insert implemented through FTP
- S3—bulk insert available for Amazon Redshift

Block size: Set the value for the number of rows to insert in a single operation using the Bulk option above.

Isolation level: This option allows you to choose a specific isolation level on the Refresh operation.

Skip Record Count: Default is False. Set to true to skip the SELECT COUNT(*) operation during refresh replications. This command is used only to show replication progress in the Replication Monitor, and omitting the operation may improve performance.

Fire Triggers: When the target database is SQL Server and the Insert Mode is set to BulkInsert, this option can be set to True to fire any triggers defined on the target database. The default value is False to provide optimal performance during replication.

Use Partitions: Check this option to define multiple partitions or subsets within a replication during the refresh operation, so that multiple parallel threads move a specific subset of the data. This is useful to improve performance during refresh operations. Click Add Partition or select a partition and click Edit Partition to open the Partition Properties dialog.

Performance and Tuning

This section contains topics that describe:

- Performance and Efficiency
- Reviewing Replication Performance Over Time
- Using Partitions for Refresh Replications
- Managing Performance with Thread Settings

Performance and Efficiency

Processing performance for Syniti Replicate varies widely from one configuration to another. Processing time depends on the resources available:

- Network bandwidth in connection between the DBMS servers and the server where Syniti Replicate is running
- Number of processors for the machines involved
- Types of processors
- Memory available
- CPU usage

Replication design needs to be carefully balanced to avoid bottlenecks during the replication process. Any measures you take depend on:

- Number of source and target connections
- Number of tables in replication
- Record size and record numbers for each table replicated
- Grouping of replications to optimize open database connections and query processing
- Presence of long size data types, such as LOB/LONG, Varchar/LONG or varbinary fields.

For instance, the size of the table to be replicated and the number of transactions are important to evaluate whether it is more convenient to run continuous refreshes during the specified time, instead of a single full refresh and a continuous mirroring afterwards.

The Management Center provides access to the Dashboard, a tool to analyze replication data and performance over a period of time based on the contents of a specified Metadata and history file or database. It can provide information for each table in a selected set of tables including:

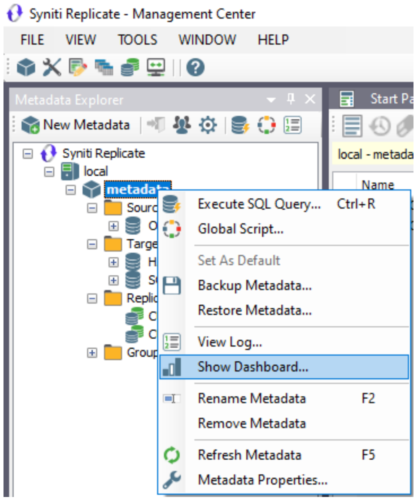
- Total number of processed records
- Total number of failed records
- Total number of records
- Total number of replication sessions
- Total number of processed records by all replications per unit of time

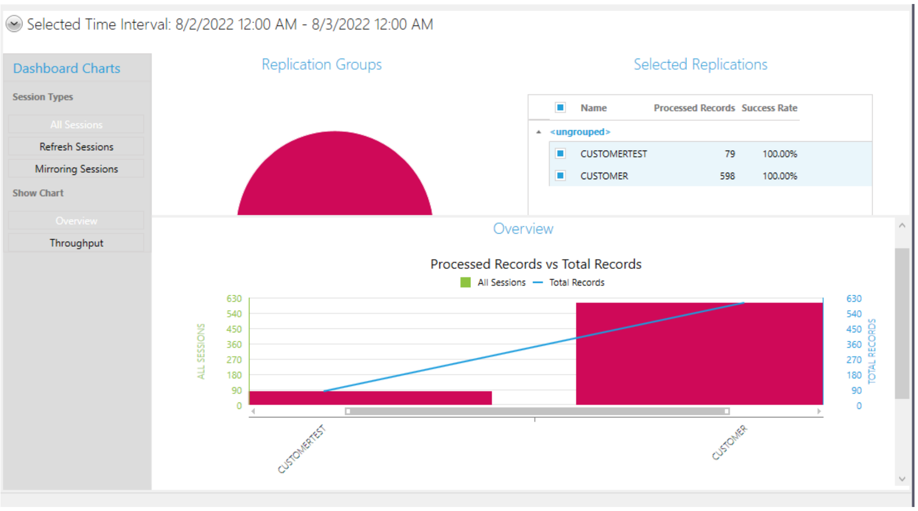
- Average time it took to replicate specified records

Reviewing Replication Performance over Time

From the Management Center, you can access the Dashboard (Statistics Tab) to analyze performance of replications associated with a specific metadata.

The Dashboard provides tools for you to select replications to analyze and charts to review data on replication performance.





Name	Processed Records	Success Rate
<ungrouped>		
CUSTOMERTEST	79	100.00%
CUSTOMER	598	100.00%

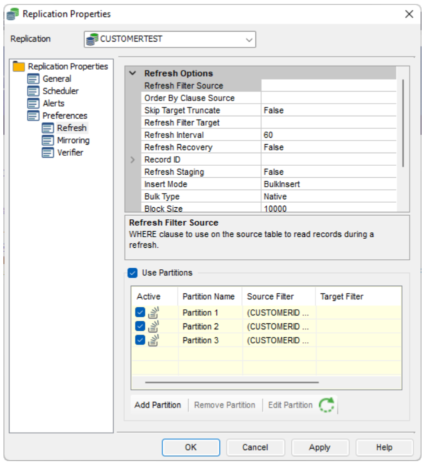
Using Partitions for Refresh Replications

Refresh Partitions allow you to define multiple partitions or subsets within a replication during the refresh operation, so that multiple parallel threads move a specific subset of the data. This approach improves the overall performance of the refresh.

Create a Partition Manually

To set up partitions for a refresh replication:

1. In the Metadata Explorer, select the replication.
2. Choose Replication Properties from the right mouse button menu.
3. In the Preferences, click the Refresh option to display Refresh properties.



Replication: CUSTOMERTEST

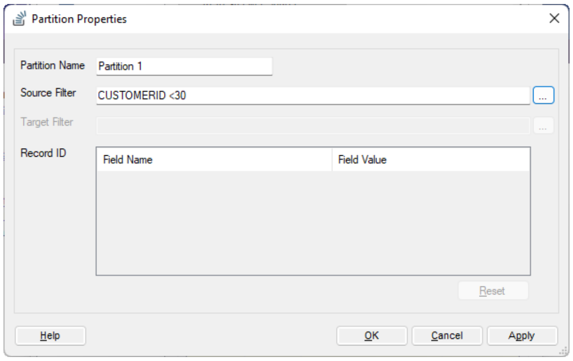
Refresh Options	
Refresh Filter Source	
Order By Clause Source	
Skip Target Truncate	False
Refresh Filter Target	
Refresh Interval	60
Refresh Recovery	False
Record ID	
Refresh Staging	False
Insert Mode	BulkInsert
Bulk Type	Native
Block Size	10000

Refresh Filter Source
WHERE clause to use on the source table to read records during a refresh.

☑ Use Partitions

Active	Partition Name	Source Filter	Target Filter
☑	Partition 1	(CUSTOMERID ...	
☑	Partition 2	(CUSTOMERID ...	
☑	Partition 3	(CUSTOMERID ...	

4. In the Partitions area, check Use Partitions.
5. Click **Add Partition** to open the Partition Properties dialog.



Partition Name: Partition 1
Source Filter: CUSTOMERID <30
Target Filter:
Record ID:

Field Name	Field Value

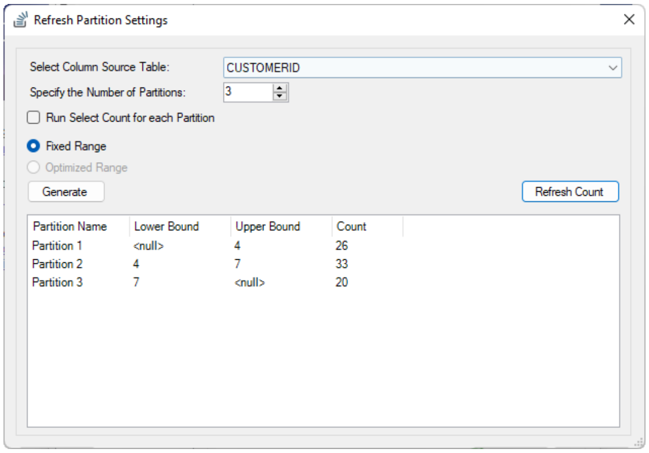
6. Edit the auto-generated name for the partition as needed.
7. Set a Source and Target Filter to determine the data included in the partition.

8. Click OK to create the partition.

Create a Partition using Partition Filter Wizard

To set up partitions for a refresh replication:

1. In the Metadata Explorer, select the replication.
2. Choose Replication Properties from the right mouse button menu.
3. In the Preferences, click the Refresh option to display Refresh properties.
4. In the Partitions area, check Use Partitions.
5. Click on the green icon ⟳ to open the wizard.
6. Select Column source table, specify the number of partitions, then click on generate.



Refresh Partition Settings

Select Column Source Table: CUSTOMERID

Specify the Number of Partitions: 3

□ Run Select Count for each Partition

● Fixed Range

○ Optimized Range

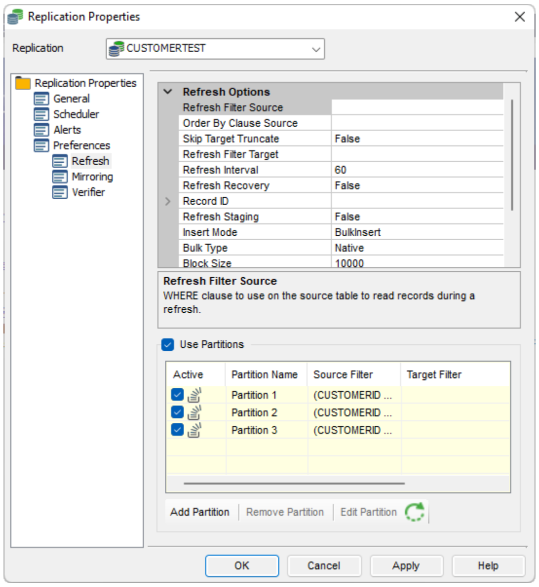
Partition Name	Lower Bound	Upper Bound	Count
Partition 1	<null>	4	26
Partition 2	4	7	33
Partition 3	7	<null>	20

7. Optionally you can run a select count for each partition, to validate if the number of partitions is acceptable according to your needs.
8. Click **OK** to create the partition.

Manage Partitions

To manage partitions for a replication:

1. In the Metadata Explorer, select the replication.
2. Choose **Replication Properties** from the right mouse button menu.
3. In the **Preferences**, click the **Refresh** option to display Refresh properties.



Replication: CUSTOMERTEST

Refresh Options	
Refresh Filter Source	
Order By Clause Source	
Skip Target Truncate	False
Refresh Filter Target	
Refresh Interval	60
Refresh Recovery	False
Record ID	
Refresh Staging	False
Insert Mode	BulkInsert
Bulk Type	Native
Block Size	10000

Refresh Filter Source
WHERE clause to use on the source table to read records during a refresh.

☑ Use Partitions

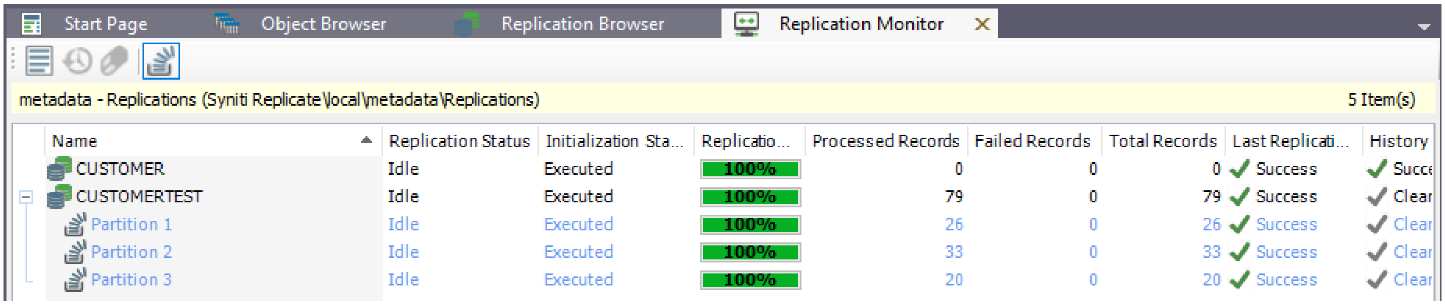
Active	Partition Name	Source Filter	Target Filter
☑	Partition 1	(CUSTOMERID ...	
☑	Partition 2	(CUSTOMERID ...	
☑	Partition 3	(CUSTOMERID ...	

4. In the Partitions area:

- Check or uncheck Use Partitions to switch on/switch off use of partitions.
- Select one or more partitions by clicking in the checkbox to the left of each partition name, then edit or remove the partition(s).

View Replication Progress

In the Replication Monitor, partitions are by default not visible. Click Show Partition button on the top toolbar to view partitions as sub rows of each replication:



metadata - Replications (Syniti Replicate\local\metadata\Replications) 5 Item(s)

Name	Replication Status	Initialization Sta...	Replicatio...	Processed Records	Failed Records	Total Records	Last Replicati...	History
CUSTOMER	Idle	Executed	100%	0	0	0	✓ Success	✓ Succe
CUSTOMERTEST	Idle	Executed	100%	79	0	79	✓ Success	✓ Clear
Partition 1	Idle	Executed	100%	26	0	26	✓ Success	✓ Clear
Partition 2	Idle	Executed	100%	33	0	33	✓ Success	✓ Clear
Partition 3	Idle	Executed	100%	20	0	20	✓ Success	✓ Clear

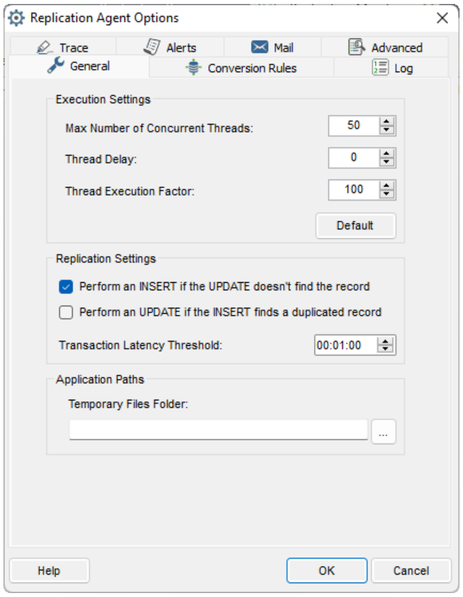
Right-click on a partition to access actions to perform on the partition.

When the refresh is being executed, a specific number of records and replication progress information for each partition is displayed. The top replication row will show cumulative values from each of its partitions.

Managing Performance Using Thread Settings

Here are some hints on how to manage performance using settings found in the Replication Agent Options dialog and Replication Properties dialog.

Replication Agent Options Dialog (General Tab)



Execution Settings

Max Number of Concurrent Threads: 50

Thread Delay: 0

Thread Execution Factor: 100

Replication Settings

☑ Perform an INSERT if the UPDATE doesn't find the record

☐ Perform an UPDATE if the INSERT finds a duplicated record

Transaction Latency Threshold: 00:01:00

Application Paths

Temporary Files Folder:

Max number of concurrent threads

Increasing this number will boost the performance but the CPU and memory usage will also increase and other applications running on the same PC might be affected.

Thread Delay

When this value is greater than 0, it represents the number of time slices the thread executes before going into a sleep state. The purpose of this parameter is to release some CPU resources if the CPU usage is too high. Setting this value to 1 instructs Syniti Replicate to suspend each thread for a fraction of time (Sleep(0)) at every time slice. 1 is the value that releases the maximum of the CPU usage but it's also the value that penalizes performance the most. Setting it to n indicates that a thread will sleep every n slices of time. If the thread delay is 0, the thread is never suspended if there are replications running.

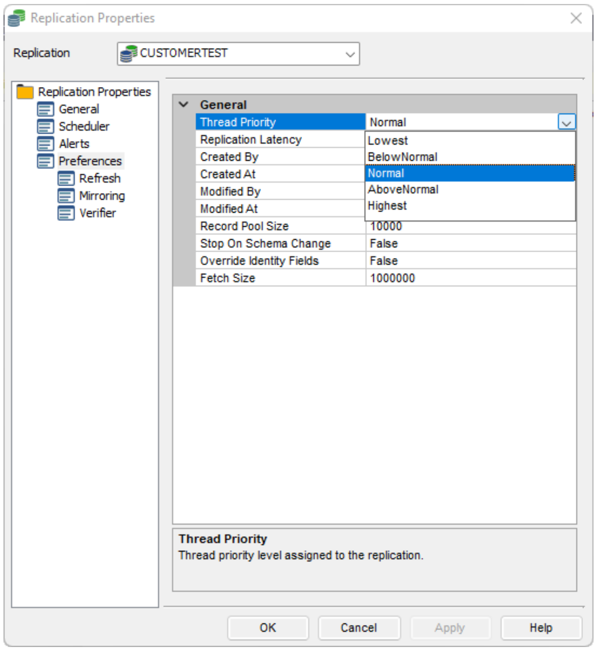
Thread execution factor

This value is a factor indicating how much time a thread spends processing a task (time slice) before switching to the following task in the execution list. It is used to increase the time slice assigned to each task in the following way. The number specified multiplies the slice of time assigned to each task (reader or writer). For example, if you double the value, the time assigned for each task is doubled.

Replication Properties Dialog

Thread Priority

This value affects the time slice assigned to a thread when the thread is running the current replication. It's similar to the "Thread execution factor", but relative to a specific replication. Increasing the thread priority of the replication might increase the performance because it increases the slice of time dedicated for processing the replication. However, the performance is also affected by the total number of replications running and the "Max number of concurrent threads" value set in the Replication Agent Options dialog.



Replication: CUSTOMERTEST

Thread Priority	Normal
Replication Latency	
Created By	
Created At	
Modified By	
Modified At	
Record Pool Size	10000
Stop On Schema Change	False
Override Identity Fields	False
Fetch Size	1000000

Thread Priority
Thread priority level assigned to the replication.

Improving Performance Using Refresh with SQL Server and MySQL

If you are replicating to Microsoft SQL Server or MySQL using Refresh mode, you can improve performance by using the bulk insert option during replication. This option is selected by default in the Preferences tab of the Replication Properties dialog.

1. In the Target Connection wizard, when creating a target connection, select the Microsoft SQL Server .NET Data Provider or the MySQL .NET Data Provider.
2. In the Replication wizard, when configuring the replication, choose Refresh as the Replication mode.
3. Select the replication in the Metadata Explorer.

4. From the right mouse button menu, choose Replication Properties.
5. In the Preferences tab, check that the Insert Mode property is set to **BulkInsert**.
6. Block Size, default value is 10,000. You can adjust according to your needs.

Command Timeout in SQL Server

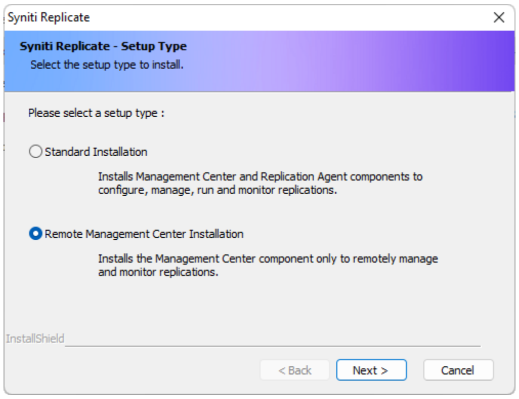
Default value is -1, which means that no timeout is set for Syniti Replicate and the default timeout determined by the .NET data provider is used. For example, the default value for Microsoft SQL Server is 15 seconds. To set a different timeout value, enter a value in seconds, you can set this value in **300**. This property determines the amount of time before any command from Syniti Replicate times out while waiting for a response from the database server.

Managing and Monitoring Replications on Remote Servers

To monitor or manage replications remotely (for example, if you have Syniti Replicate installed on several systems and want to manage the installations from a single location), it is possible to:

Add remote servers to any standard installation of Syniti Replicate.

Install the Management Center only, then add remote servers to the Management Center.



Syniti Replicate

Syniti Replicate - Setup Type

Select the setup type to install.

Please select a setup type :

○ Standard Installation

Installs Management Center and Replication Agent components to configure, manage, run and monitor replications.

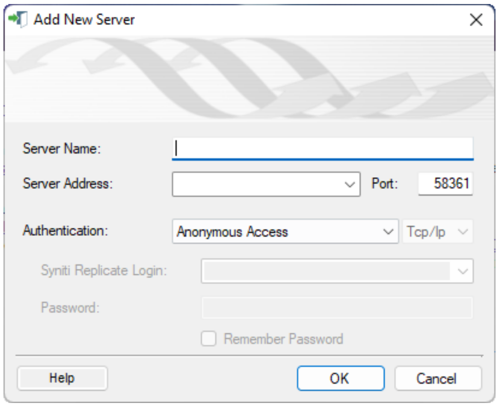
● Remote Management Center Installation

Installs the Management Center component only to remotely manage and monitor replications.

If you choose to manage replications remotely, there are a few issues to consider:

When you install Syniti Replicate to be run from a remote Management Center, the metadata connection has to be accessible from the remote system and point to a unique server name identifiable in the local network. Local-only connection strings such as those that might be used with Microsoft SQL Server CE or Microsoft Access do not work unless remote access is specifically established (for example, if pointing to a local database file, make sure that the file is shared and accessible from the remote machine).

Each Management Center component installed remotely connects to one (or more) server agent(s) to retrieve authorization/security information and metadata information for the server. The Management Center application will be able to monitor and partially modify remote Syniti Replicate configurations. When the Management Center is used for the local system, additional permissions are available, such as the ability to modify server security and server configuration settings, set the default metadata, add or remove metadata, change license, and so on.



Port: 58361

Authentication: Anonymous Access

Tcp/Ip

☐ Remember Password

User control over local and remote installations of the server depends on the established user ID and permissions. In a default Syniti Replicate installation, there are no user restrictions. Users can be defined in the User Settings dialog, available in the Metadata Explorer.

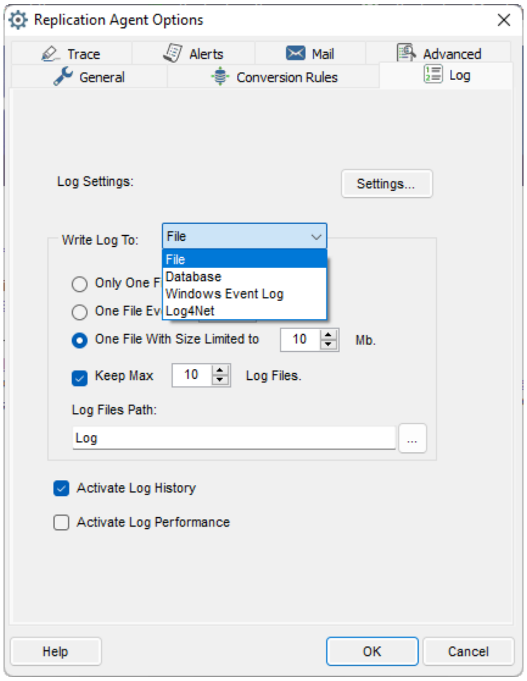
Discovering Syniti Replicate Log Capabilities

The settings on this tab apply to the Syniti Replicate log (.log extension, viewable in the Log Viewer), the Syniti Replicate history files (.his extension, viewable in the History Viewer), and the Server Log (_SA naming convention.) This tab allows you to specify log type, location (if a database), size and generation frequency.

Write Log To

You can save the Syniti Replicate log and history to one of the following. Your selection will be applied to both the log and history files.

- A text file
- A database
- The Microsoft Windows Event Log
- Apache log4net™



Log Settings: Settings...

Write Log To: File

File
Database
Windows Event Log
Log4Net

○ Only One F
○ One File Ev
● One File With Size Limited to 10 Mb.

☑ Keep Max 10 Log Files.

Log Files Path: Log

☑ Activate Log History

☐ Activate Log Performance

File Settings

Only one file

If selected, a single file with no size limit is used for traces. This file can quickly become very large and should be monitored on a regular basis.

One file every X day

If selected, and a number entered, generates a new trace file after the number of days specified. All old trace files are kept in the log directory. The convention used to name the Management Center trace files is DBMotoEM_ xxxx.trc, where xxxx is a number starting at 0001. If you do not set the "Keep Max X Log Files" option, remember to clean up your log directory regularly.

One file with size limited to X Mb

If selected, and a number entered, creates a new trace file when size X is reached. All old trace files are kept in the log directory. The convention used to name the Management Center trace files is DBMotoEM_xxxx.trc, where xxxx is a number starting at 0001. If you do not set the "Keep Max X Log Files" option, remember to clean up your log directory regularly.

Keep Max X Log Files

This value is checked by default. **If unchecked, all old trace files are kept in the log directory**. If checked, keeps only the specified number of old log files in the log directory. Use file dates (not file names) to determine the most recent files.

Keep all logs is highly recommended when you want to replicate tables with millions of records in a Refresh mode or when you have a huge number of transactions in Mirroring. You have to consider, keeping the files in the disk, the hard disk space used by Syniti Replicate will be incremented.

Activate Log History

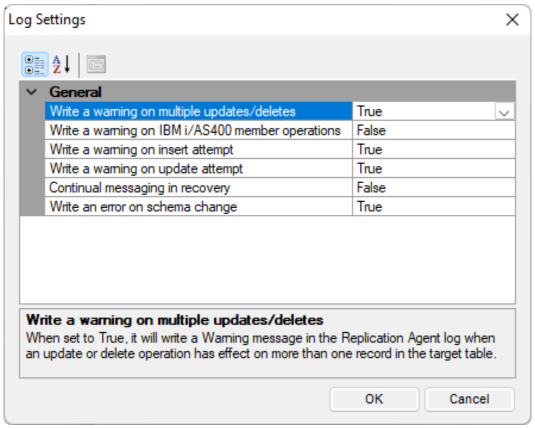
Check this option to start logging replication status information for each replication session. All information is recorded in a file with a .his extension in the Syniti Replicate Logs folder. This information can then be displayed and reviewed in the History Viewer.

You may want to leave this option unchecked if you are concerned about disk space and processing load for the Replication Agent. The process stores one record for each replication during each refresh/mirroring session. In the case of mirroring, this typically means that the .his file is updated every 60 seconds. In order to deploy this information in the dashboard, you need to enable the log history.

Activate Log Performance

Unchecked by default. When checked, it generates a log file ('TaskTimer_<current_day>.txt) containing information for each replication or replication group regarding execution time, number of records processed and thread priority.

Log Settings Dialog



Log Settings

General	
Write a warning on multiple updates/deletes	True
Write a warning on IBM i/AS400 member operations	False
Write a warning on insert attempt	True
Write a warning on update attempt	True
Continual messaging in recovery	False
Write an error on schema change	True

Write a warning on multiple updates/deletes
When set to True, it will write a Warning message in the Replication Agent log when an update or delete operation has effect on more than one record in the target table.

Write a warning on multiple updates/deletes

True by default so that warnings are written to the log for update or delete operations that update more than one field. The warning is useful because it notifies you that there may be issues with primary key field settings which do not uniquely identify records. Disable this warning in cases where a trigger is attached to the field being updated or deleted, and that trigger also updates other fields. In this case, the log could quickly fill with warning messages for a behavior that is intentional.

Write a warning on IBM i/AS400 member operations

Relevant only when using IBM Db2 for i (AS/400) as a source database. False by default. When set to true, any operation performed on members will generate a message in the Replication Agent log.

NOTE: This flag was previously Write a warning on member reorganize. The earlier behavior is included in the new functionality.

Write a warning on insert attempt

True by default. Adds a warning to the Syniti Replicate log when an attempt is made to insert a record not found while trying to do an update.

Write a warning on update attempt

False by default. Adds a warning to the Syniti Replicate log when an attempt is made to update a record following a duplicate key error

Continual messaging in recovery

True by default. When set to true, error messages are written to the log even during recovery.

Write an error on schema change

True by default. When set to true, an error message is written to the log when an ALTER TABLE has been executed on a replicated table. It is highly recommended that you keep this option enabled so that you can detect table changes and thereby avoid or quickly repair replication errors.