



Syniti Replicate

Setup Notes for Replicating with PostgreSQL

Version 10.1



Table of Contents

Setup Notes for Replicating with PostgreSQL.....	1
Connection Type.....	1
PostgreSQL System Settings	2
Notes and Limitations on Using PostgreSQL as a Source in Mirroring	2
Setup steps on PostgreSQL (Windows environment).....	2
Add Source Connection Wizard	3
Select Provider Screen	3
Enable Transactional Replication Wizard.....	4
Log Type Screen	4
Log Settings.....	4

Syniti Replicate

These notes provide essential information for setting up replications using **PostgreSQL**. 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 PostgreSQL log. 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](#).

This guide describes the setup process using the Log Server Agent option for one-way mirroring and synchronization when replicating data from a PostgreSQL database. For mirroring and synchronization replications using PostgreSQL as a source, Syniti Replicate offers:

- **Log Server Agent:** Uses a Windows service and a Log Server component to query the PostgreSQL log for increased performance when dealing with large amounts of data.
- **Triggers:** Uses Syniti Replicate triggers installed on the PostgreSQL database to log changes. This option is required for implementing synchronization with PostgreSQL. (Ask the technical support team via the [Help Center](#) to see if this approach would work in your application.) Download the separate setup guide [Setup Notes for PostgreSQL Using Triggers](#).

Connection Type

PostgreSQL .NET Data Provider [recommended by PostgreSQL](#)

Assembly: Npgsql (file name: Npgsql.dll)

Sample path: C:\Npgsql-2.2.3-net40\Npgsql.dll

To use the bulkinsert feature for increased performance:

1. The driver DLL has to be installed using the [Microsoft Global Assembly Cache Tool](#) (GAC). If the provider is not registered, you will receive the following error when attempting to use the provider:

```
Error creating the command for writing to the target (Replication: 'EMP1' - Target table: 'public.test1')
```

```
System.IO.FileNotFoundException: Could not load file or assembly 'Npgsql, Version=2.2.3.0, Culture=neutral, PublicKeyToken=5d8b90d52f46fda7' or one of its dependencies. The system cannot find the file specified.
```

```
File name: 'Npgsql, Version=2.2.3.0, Culture=neutral, PublicKeyToken=5d8b90d52f46fda7'
```

```
at ....
```

The commands for installing the PostgreSQL DLLs are as follows when running from the location where the GAC tool is installed:

```
gacutil.exe -i C:\Npgsql-2.2.3-net40\Npgsql.dll
```

```
gacutil.exe -i C:\Npgsql-2.2.3-net40\Mono.Security.dll
```

2. The provider version number stored in the DBReplicator.exe.config file (in the Syniti Replicate install folder) should match the provider version that you are using. To change the version of the provider, specify the "newVersion" in the DBReplicator.exe.config file as follows:

```
<dependentAssembly>
  <assemblyIdentity name="Npgsql" publicKeyToken="5d8b90d52f46fda7"
culture="neutral"></assemblyIdentity>
  <bindingRedirect oldVersion="2.2.3.0" newVersion="2.2.3.0"></bindingRedirect>
</dependentAssembly>
```

PostgreSQL System Settings

Syniti Replicate supports PostgreSQL versions 9.5 and above as a source database for mirroring. It takes advantage of the PostgreSQL replication slot feature by requiring that every connection to PostgreSQL from Syniti Replicate use a dedicated replication slot. The [PostgreSQL documentation](#) provides an in-depth explanation of replication slots, but briefly, a single slot represents an ordered stream of changes that occur on a specific database on the origin server. Note that if you enable a replication slot from Syniti Replicate without using it, the slot will consume resources and prevent the database cleanup functionality to purge old records from the transaction log, because they do not appear as consumed by an existing slot. Therefore, it is important to remove replication slots from the database by removing the Syniti Replicate connection if it is not used in replication.

Notes and Limitations on Using PostgreSQL as a Source in Mirroring

- Obtain the library files developed for PostgreSQL replications by requesting them in the [Help Center](#)
- PostgreSQL Versions 9.5 and above are supported as a source for mirroring. Synchronization is supported only with Triggers.
- No DDL (schema changes) operations are logged, only INSERT, UPDATE and DELETE.
- In updates, only the after record of the new changes is logged. No before image is available
- The functions used to create and retrieve changes from the replication slots can only be executed by a superuser or a user with replication role.
- Replicated tables need to have primary keys defined.
- Replication last ID by datetime is not supported.

Setup steps on PostgreSQL (Windows environment)

1. Syniti Replicate uses the test_decoding.dll (test_decoding.so) that is included with PostgreSQL. However, it is possible to obtain a specialized library for enhanced performance, dbm_decoding, that reduces the size of the stream sent through the network:
 - Contact the technical support team via the [Help Center](#) to request the library files developed for PostgreSQL replications.
 - Extract the files from the zip file to a folder that is accessible to the PostgreSQL lib folder.
 - If using Microsoft Windows, rename the appropriate DLL to dbm_decoding.dll.
If using LINUX, rename the .so file to dbm_decoding.so.
 - Copy the file to the lib folder in your PostgreSQL installation.
2. Edit the postgresql.conf file to set up the environment for replication using Syniti Replicate:

```
wal_level = logical
max_replication_slots = 3
track_commit_timestamp = on
```

Syniti Replicate

where:

wal_level sets the amount of information that is written to the WAL (Write-Ahead Logging, or Postgres transaction log). The value **Logical** represents the level of detail necessary to support logical decoding, required for using PostgreSQL as a source database for replicating data changes only in Syniti Replicate. This parameter can only be set at server start.

max_replication_slots sets the maximum number of replication slots available to all applications. Syniti Replicate requires a “replication_slot” for every connection that it uses to replicate from PostgreSQL. A maximum number of 3 is reasonable if you are defining one single connection to the database. If you require additional connections to the same PostgreSQL database, the maximum number of slots should be raised accordingly.

Details on settings in the configuration file can be found in the [PostgreSQL documentation](#).

3. Restart the PostgreSQL database service so that your changes take effect.
4. Make sure that the user setting up and running replications is a superuser or a user with replication role.

Add Source Connection Wizard

Select Provider Screen

Database

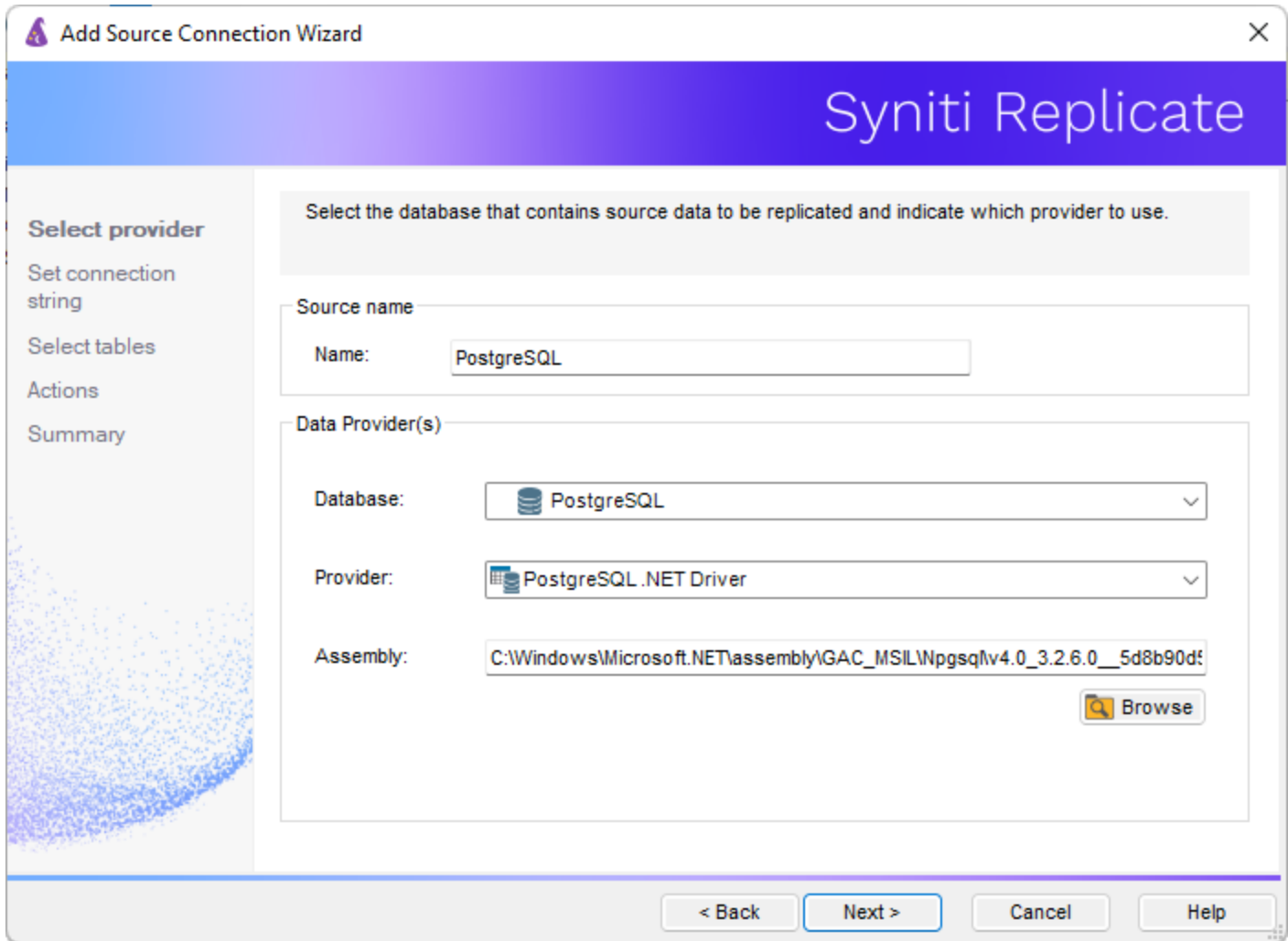
Select PostgreSQL from the drop-down list.

Provider

PostgreSQL .NET Driver

Assembly

Locate the file Npgsql.dll, typically in the installation folder for the PostgreSQL .NET Provider.



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 PostgreSQL.

Log Type Screen

Select the Log Server Agent option.

Log Settings

Decoding Plugin: Plugin Name

The default plugin is called 'test_decoding' and is normally installed with every PostgreSQL installation. To verify availability, check the location where the decoding plugins are normally installed:

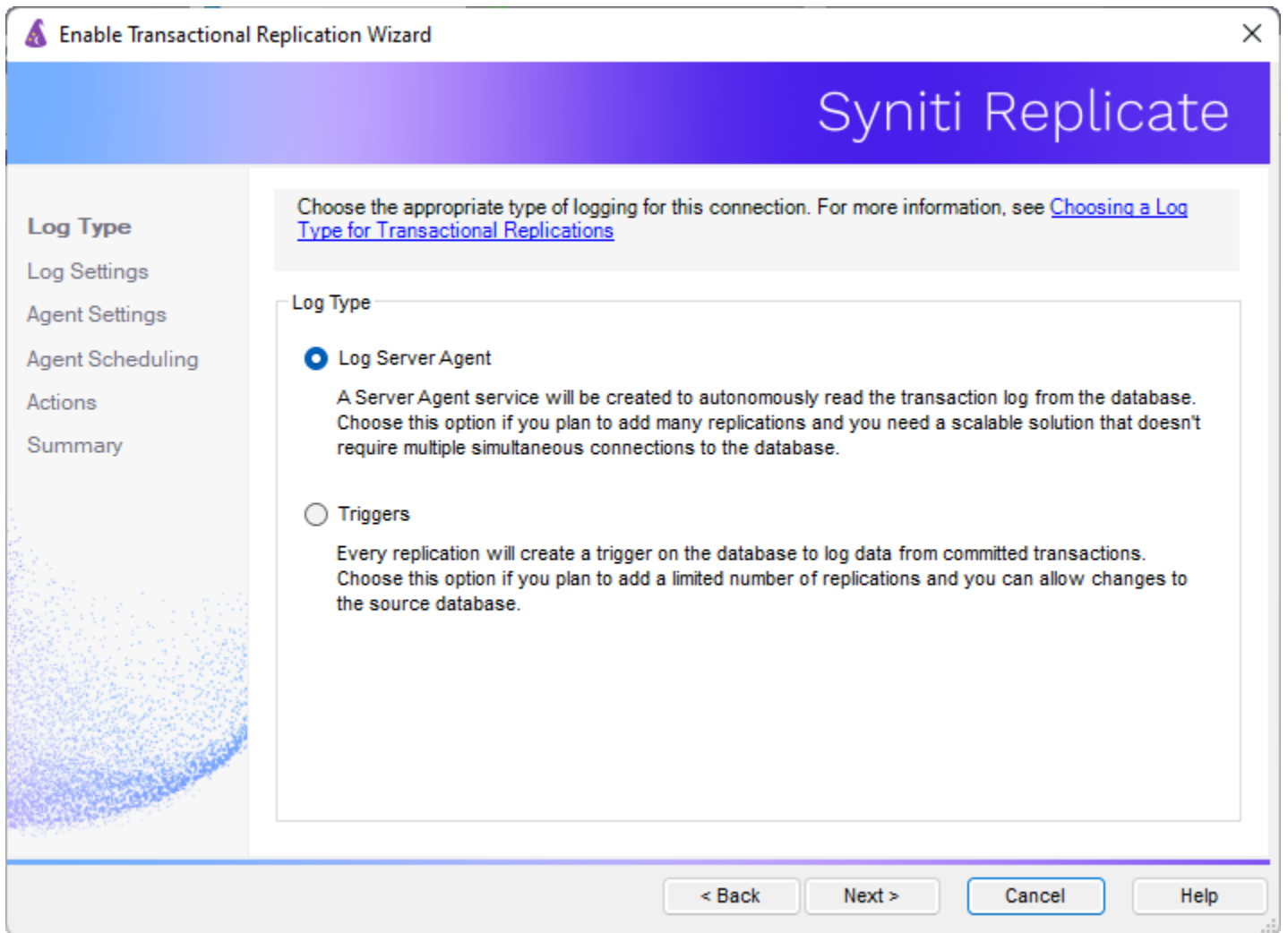
MS Windows <Postgres installation folder>/lib

Syniti Replicate

Ubuntu /usr/lib/postgres/9.5/lib

RedHat /usr/pgsql-9.5/lib

If the plugin is missing, you can either install it using the Postgres installation files, or request a file from the technical support team via the [Help Center](#) to install the dbm_decoding plugin. Note that if you are using PostgreSQL as a cloud database, you need to use the test_decoding plugin.



Replication Slot: Use Existing Slot

Replication slots are channels that you open on the database to allow Syniti Replicate to receive data changes from the log. While the default is to use a new slot specifically for a Syniti Replicate connection, it is possible to reuse an existing slot if it is not consumed by any other connection. A replication slot has to be uniquely assigned to a single Syniti Replicate connection.

Copyright© 2022 by BackOffice Associates, LLC d/b/a Syniti and/or affiliates. 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.

Replication Slot: Add New Slot

Syniti Replicate needs to define a “`replication_slot`” for every connection that it uses to replicate from PostgreSQL. This option allows you to set up a new slot from Syniti Replicate. However, the new slot will be added only if the maximum number of slots has not been exceeded. This value is set in the `postgresql.conf` file. Be aware that if slots are created then left unused, they still consume resources, so you should always manage replication slots carefully.