



DBMoto[®]

Setup Notes for Microsoft SQL Server Transactional Replications

Version 9.5.0.20

Software Release Date: 4/23/18

HiT Software, Inc.
4040 Moorpark Ave
Suite 221
San Jose, CA 95117

T +1 408-345-4001
F +1 408-345-4899
info@hitsw.com
www.hitsw.com

Document History

Version	Author	Date	Reviewer	Date	Approver	Date	Comments
1	JHLorenzin	6/13/18	ALaspertini		VFarruggio		

Table of Contents

Connection Type	1
SQL Server User Permissions	1
Refresh with SQL Server as Either Source or Target Database	1
Transactional Replications/Initial Refresh with SQL Server as Either Source or Target Database	3
Add Source Connection Wizard	5
Select Provider Screen.....	5
Set Connection String Screen.....	5
Enable Transactional Replication Wizard	6
Log Type Screen.....	6
Log Settings Screen.....	6

These notes provide essential information for setting up replications using **Microsoft SQL Server** as a source database for one-way mirroring and synchronization. The setup process for a refresh replication can usually be completed using the DBMoto wizards without additional documentation because it does not involve access to the SQL Server logs.

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

This guide describes the setup process using either the Log Reader or Log Server Agent options for one-way mirroring and synchronization when replicating data from an Oracle database. For mirroring and synchronization replications using SQL Server as a source, DBMoto offers several approaches:

- **Log Reader:** Queries the SQL Server Log directly
- **Log Server Agent:** Uses a Windows service and a DBMoto Log Server component to query the SQL Server Log for increased performance when dealing with large amounts of data.
- **Triggers:** Uses DBMoto triggers installed on the SQL Server database to log changes (Ask the HiT Software Technical support team at <http://support.hitsw.com> to see if this approach would work in your application and to obtain the separate document *Setup Notes for Trigger-based Transactional Replications*).

Connection Type

Microsoft SQL Server .NET Data Provider (included with .NET Framework)

Assembly: System.Data (file name: System.Data.dll)

SQL Server User Permissions

When setting up replications that use SQL Server 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 DBMoto 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 SQL Server as Either Source or Target Database

1. AUTHORITY TO CONNECT TO DATABASE

This should already be granted when the user is created. However, here is the syntax, just in case:

```
grant connect to <uid>;
```

Example where dbmoto is the user ID:

```
grant connect to dbmoto;
```

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), DBMoto runs a SELECT command on the catalog. If the user ID has insufficient privileges, an error is generated on the SQL Server system.

The command below allows read access to any table in the database, including catalog information. It is actually broader than the necessary permission, since this step requires access only to the system tables.

```
use <database>;
exec sp_addrolemember 'db_datareader', '<uid>'
```

Example where dbmoto is the user ID:

```
use test;
exec sp_addrolemember 'db_datareader', 'dbmoto'
```

3. AUTHORITY TO SELECT TABLES

DBMoto 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. If you used the syntax in 2 above, you do not need to grant select authority to specific tables involved in replications because the above command covers all tables.

4. AUTHORITY TO CREATE TABLES

To create a target table in the Management Center (as part of the Create Table Wizard), DBMoto requires permissions to modify the database schema.

```
use <database>;
grant alter on schema::<schema> to <uid>;
grant create table to <uid>;
```

Example where dbmoto is the user ID:

```
use test;
grant alter on schema::dbo to dbmoto;
grant create table to dbmoto;
```

5. AUTHORITY TO UPDATE TABLES

To create a target table in the Management Center (as part of the Create Table Wizard), DBMoto requires write permissions to the database. The following command allows write access to any table in the database and is broader than the necessary permission. It is possible to grant more granular access to every single table in the database, if necessary, by changing permissions for a user.

```
use <database>;
exec sp_addrolemember 'db_datawriter', '<uid>'
```

Example where dbmoto is the user ID:

```
use test;
exec sp_addrolemember 'db_datawriter', 'dbmoto'
```

6. AUTHORITY TO DROP TABLES, ALTER TABLES (optional)

The use of these commands from within DBMoto is entirely optional (i.e. not necessary for running a refresh replication.) They are used if you choose to remove a table from SQL Server or change the table via the Management Center SQL Query tab. The following commands are broader than needed. Alter and drop can also

be granted to specific tables.

```
use <database>;
grant alter on schema::<schema> to <uid>;
grant create table to <uid>;
```

Example where dbmoto is the user ID:

```
use test;
grant alter on schema::dbo to dbmoto;
grant create table to dbmoto;
```

Transactional Replications/Initial Refresh with SQL Server as Either Source or Target Database

This section includes information for mirroring where SQL Server is the data source, and synchronization where SQL Server can be either the “source” or “target” data source.

7. AUTHORITY TO CONNECT TO DATABASE

This should already be granted when the user is created. However, here is the syntax, just in case:

```
grant connect to <uid>;
```

Example where dbmoto is the user ID:

```
grant connect to dbmoto;
```

8. 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), DBMoto runs a SELECT command on the catalog. If the user ID has insufficient privileges, an error is generated on the SQL Server system.

The command below allows read access to any table in the database, including catalog information. It is actually broader than the necessary permission, since this step requires access only to the system tables.

```
use <database>;
exec sp_addrolemember 'db_datareader', '<uid>'
```

Example where dbmoto is the user ID:

```
use test;
exec sp_addrolemember 'db_datareader', 'dbmoto'
```

9. AUTHORITY TO SELECT TABLES

DBMoto 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.

If you used the syntax in 2 above, you do not need to grant select authority to specific tables involved in replications because the above command covers all tables.

10. AUTHORITY TO CREATE TABLES (Optional)

To create a target table in the Management Center (as part of the Create Table Wizard), DBMoto requires

permissions to modify the database schema.

```
use <database>;  
grant alter on schema::<>schema> to <uid>;  
grant create table to <uid>;
```

Example where dbmoto is the user ID:

```
use test;  
grant alter on schema::dbo to dbmoto;  
grant create table to dbmoto;
```

11. AUTHORITY TO UPDATE TABLES (Optional)

To create a target table in the Management Center (as part of the Create Table Wizard), DBMoto requires write permissions to the database. The following command allows write access to any table in the database and is broader than the necessary permission. It is possible to grant more granular access to every single table in the database, if necessary, by changing permissions for a user.

```
use <database>;  
exec sp_addrolemember 'db_datawriter', '<uid>'
```

Example where dbmoto is the user ID:

```
use test;  
exec sp_addrolemember 'db_datawriter', 'dbmoto'
```

12. AUTHORITY TO DROP TABLES, ALTER TABLES (Optional)

The use of these commands from within DBMoto is entirely optional (i.e. not necessary for running a refresh replication.) They are used if you choose to remove a table from SQL Server or change the table via the Management Center SQL Query tab. The following commands are broader than needed. Alter and drop can also be granted to specific tables.

```
use <database>;  
grant alter on schema::<>schema> to <uid>;  
grant create table to <uid>;
```

Example where dbmoto is the user ID:

```
use test;  
grant alter on schema::dbo to dbmoto;  
grant create table to dbmoto;
```

13. AUTHORITY TO SET UP TRANSACTIONAL REPLICATIONS and ACCESS LOG

(Note: Not needed when SQL Server is a target in mirroring replications or when using triggers instead of the transaction log to identify data change.)

When setting up transactional replications (where SQL Server is serving as the source of data, either for mirroring or for synchronization), you need additional privileges that are not required for replication operations. You need a user with sysadmin privileges to set up and verify the distributor. For synchronization,

a separate user with sysadmin credentials is needed in order to handle conflicts based on the user.

```
exec sp_addsrvrolemember '<uid>', 'sysadmin';
```

Example:

```
exec sp_addsrvrolemember 'dbmoto', 'sysadmin';
```

14. AUTHORITIES TO EXECUTE REPLICATIONS WITHOUT USING A SYS ADMIN LOGIN (optional).

If you are using a login with system administrator privileges to run replications, you do not need to set the following privileges. However, in cases where you do not want to have replications running under a system administrator login, you can set the following privileges for an alternative login instead.

DBRS_distributor database: 'db_datareader' and 'replmonitor' roles

msdb database: 'db_datareader', 'SQLAgentReaderRole', 'SQLAgentUserRole' roles

publisher database: 'db_owner' OR 'db_datareader' role and PAL member

To add a user to the Publications Access List (PAL) from the SQL Server Management Studio, right click on Replication, then Local Publications, then the publication name, then Properties) and add the user. All remaining privileges can be set via the SQL Server Management Studio (Security, then 'User name', then Properties, then User Mapping).

Add Source Connection Wizard

Select Provider Screen

Database

Select Microsoft SQL Server from the drop-down list.

Provider

The value Microsoft .NET Driver (SQL Client) is provided automatically. You do not need to change this value.

Assembly

The pathname to the .NET Assembly for the SQL Server Data Provider is already available in the .NET environment so you do not need to provide a value for this field.

Set Connection String Screen

User ID

When replicating from SQL Server using mirroring or synchronization, for the user ID and password fields, provide a login that does not have system administrator privileges. This login ID should also be used exclusively for DBMoto (i.e., no other applications should use the login ID that you specify for DBMoto). You will need to provide a second login with system administrator (or the alternative specific privileges described) in the [Setup Info](#) screen. The second login is used to create the distributor and to access the transaction log.

For Synchronization Replications

The login/user ID that you provide must be unique to DBMoto. It should not be used for any transactions occurring in either database involved in the synchronization. DBMoto does not replicate transactions by the user

you specify in this connection. This user ID is used by DBMoto 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 whether you plan to perform replications using the Log Reader (default) or Log Server Agent.



Log Settings Screen

Publisher

The name of the database server that you are using for the connection.

Distributor

Initially, this field displays the name of the database server you are using for the connection. When you click Verify to see if a Distributor is installed for this database server, the value changes as follows:

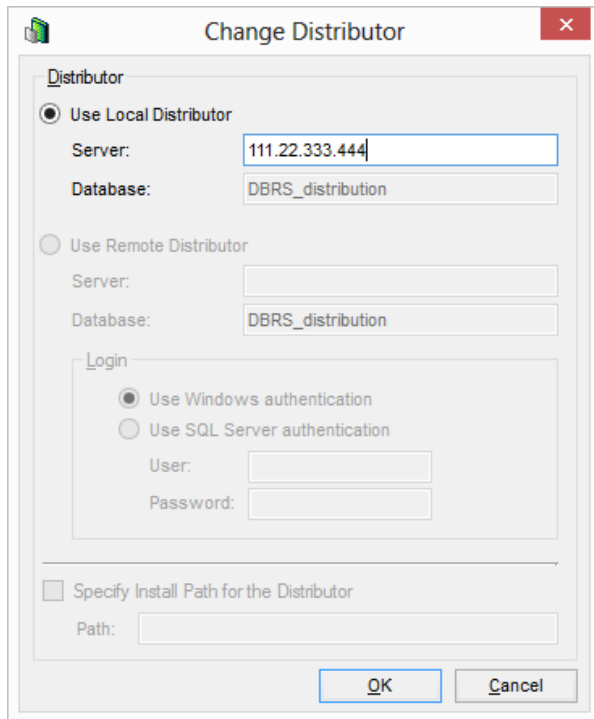
- If a distributor exists for the server, the server name is displayed and two buttons are active
 - The  Ellipsis button to the right of the Distributor field
Click here to open the Change Distributor dialog and view the distributor values. You cannot edit values in the dialog unless you first exit the dialog, remove the installed distributor, then re-open the dialog.
 - The **Remove** button is active so that you can remove the distributor in cases where you prefer to use a distributor defined for another accessible installation of SQL Server.
- If no distributor exists, the server name is displayed, and two buttons are active:
 - The  Ellipsis button to the right of the Distributor field
Click here to open the Change Distributor dialog and set or create a distributor on a different SQL Server installation from the one containing the table(s) to be replicated (remote distributor).
 - The **Install** button is active so that you can install a distributor. If you choose this option to install a distributor in the current SQL Server installation, DBMoto creates a distributor called DBRS_Distribution.

Be sure to install a distributor before closing the dialog.

Change Distributor Dialog

This dialog can be accessed by clicking  next to the Distributor field.

If you have a distributor installed in the SQL Server to which you are connecting for replication, the Use Local Distributor option is selected and values in the Server and Database fields reflect the SQL Server installation and distributor database. These values cannot be changed from the dialog. To change the distributor, you need to close the dialog, remove the local distributor and reopen the dialog.



If you do not have a distributor installed and want to set or create a distributor on a remote installation of SQL Server, select the **Use Remote Distributor** option and enter the SQL Server name (or an established alias for the SQL Server system), then the distributor database name. If the distributor does not exist, one will be created using the specified name. Provide a user name and password with appropriate privileges for creating the distributor in the remote installation of SQL Server.

If problems arise when DBMoto attempts to create the remote distributor, you may need to enter the full pathname to the location where the distributor should be created. This is usually in the SQL Server folder that contains the Data and Log folders, for example, "C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL." The path has to be the complete path up to the SQL Server folder that contains the Data and Log folders.

Use Windows Authentication

Select this option if you (or your system administrator) have set up your environment to use your Windows login ID to access SQL Server. You may also need to open the [Connection Properties dialog](#) and set the optional field Integrated Security to True.

Use SQL Server Authentication

Select this option to use a SQL Server login ID and password. Provide a system administrator login and password. HiT Software recommends that you use this type of authentication.

Last Updated on 6/13/2018