

CONNX Data Synchronization 12.5

Table Of Contents

DataSync.....	1
Introduction	1
System Requirements.....	2
Migrating from an Earlier Version of DataSync.....	3
Summary of Product Functions.....	4
The CONNX Data Synchronization Tool Window.....	5
Installing DataSync	6
Synchronization Type Options.....	7
Getting Started	8
Settings Tab	19
Setting Tab sub-tabs	19
Sync Behavior sub-tab.....	24
To create index before table population.....	30
To recreate indexes on target table	33
To disable logging.....	36
To set transaction batch size	37
To set days to keep history	38
Data Types sub-tab.....	39
To treat CHAR data types as CLOB	40
To select the maximum number/decimal precision for target columns.....	42
Error Handling sub-tab.....	43
Metadata sub-tab	46
SMTP Mail sub-tab	47
Tables Tab.....	49
Transformations Tab	53
Designing a Transform.....	58
Using the Expression Builder	109
Groups Tab.....	117
Schedule Tab	124
Per Schedule Synchronization Tasks.....	140
Per Schedule Synchronization Tasks	140
Before Synch Task.....	141
On Success Task	142
On Failure Task.....	143
Global Tasks Tab	144
Available Types of Scheduled Tasks.....	145
Scheduled Tasks.....	145
Global Scheduled Synchronization Tasks	146
Global Scheduled Synchronization Task - Before Synch.....	147

Global Scheduled Synchronization Task - On Success Task 148

Global Scheduled Synchronization Task - On Failure Task 149

Global On-Demand Synchronization Tasks..... 150

Global On-Demand Synchronization Task - Before Synch Task..... 151

Global On-Demand Synchronization Task - On Success Task..... 152

Global On-Demand Synchronization Task - On Failure Task 153

Troubleshooting..... 154

Purging History feature 154

Compact Database feature 155

Duplicate keys 156

Error messages 157

 Command-line Feature..... 159

Starting a synchronization from a command line 159

 Trademarks and Copyrights 160

CONNX Data Synchronization Tool..... 160

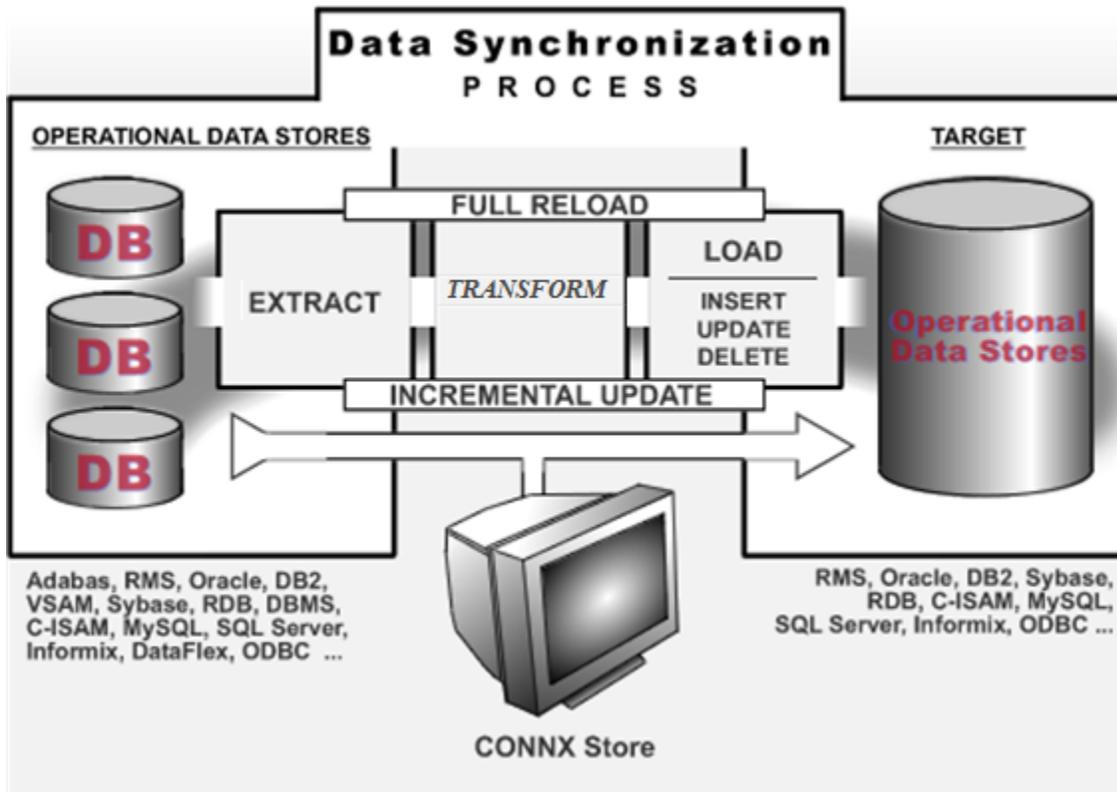
Index..... 163

DataSync

Introduction

The **CONNX Data Synchronization Tool** pumps data to a data mart or a data warehouse. With CONNX Data Synchronization, users can move a complete snapshot of a pre-defined list of tables, views or complex transformations into the Data Warehouse. The tool can also incrementally update the current snapshot of data through a simple-to-use scheduling interface. This accomplishes the goal of "near real-time" data synchronization.

Note: This is a Windows-based tool.



System Requirements

The system requirements for the CONNX Data Synchronization Tool are listed in the table below:

Required	Recommended
Windows XP SP2 or above/2003/Vista/2008 Server /Windows 7	Windows XP SP2 or above/2003/Vista/2008 Server /Windows 7
Server	
200 GB Free Space	100+ GB Free Space
800 MHz or higher CPU	Dual processor - 2 GHz or higher CPU
256 MB RAM	2 GB RAM
IDE or SCSI disk drive controller	Raid controller or mirrored drive controller
10 MB/sec network card or WAN connection	1 GB network card or LAN connection

Migrating from an Earlier Version of DataSync

DataSync uses a migration utility to help upgrade the CONNXSTORE database to include necessary changes to the database. A program called "DatapumpUpgrade" is run during the install to perform this task. In certain cases (when upgrading from CONNX 8.8, 8.9, and 9.0 SP1), the upgrade cannot work because the CDD must first be saved in the CONNX 11 (and later) format. The upgrade will succeed when the necessary actions listed in the table below are completed.

The following matrix displays actions that should be taken when upgrading to CONNX 12:

CONNX VERSION	ACTION
8.8, 8.9, and 9.0 SP1	Save the CDD in the new CONNX 11 (and later) format by opening the CONNX Data Dictionary and re-saving it Re-create DataSync schedules (necessary because of third-party control changes)
9.0 SP2 to 10.5 build 8331	Re-create DataSync schedules (necessary because of third-party control changes)

Summary of Product Functions

DataSync provides several unique product functions.

- **Full reloads on demand**

Purges all prior snapshot data in the data warehouse, replacing it in the data warehouse with a completely new snapshot.
- **Incremental updates on demand**

Detects which records have been added, deleted, or changed and inserts, updates, or deletes the corresponding data in the data warehouse snapshot tables at a fraction of the usual time. The incremental update runs up to 10 times faster than the full update, a significant decrease in the time it takes to synchronize.
- **Scheduled incremental updates**

Detects which records have been added, deleted, or changed and inserts, updates, or deletes the corresponding data in the data warehouse snapshot tables. Each file in the synchronization process may be scheduled either independently or as a group.
- **Scheduled full reloads**

Purges all prior snapshot data in the data warehouse. A completely new snapshot of the data is placed in the data warehouse. Each file in the synchronization process may be scheduled either independently or as a group.
- **Datamart model schema creation**

Automatically performs a complete analysis of all source database schemas and performs complete, applicable transformations for the target datamart schema.
- **Error detection and correction**

Provides options for automatically detecting error conditions and resyncing until a successful sync has been performed.
- **Task triggering**

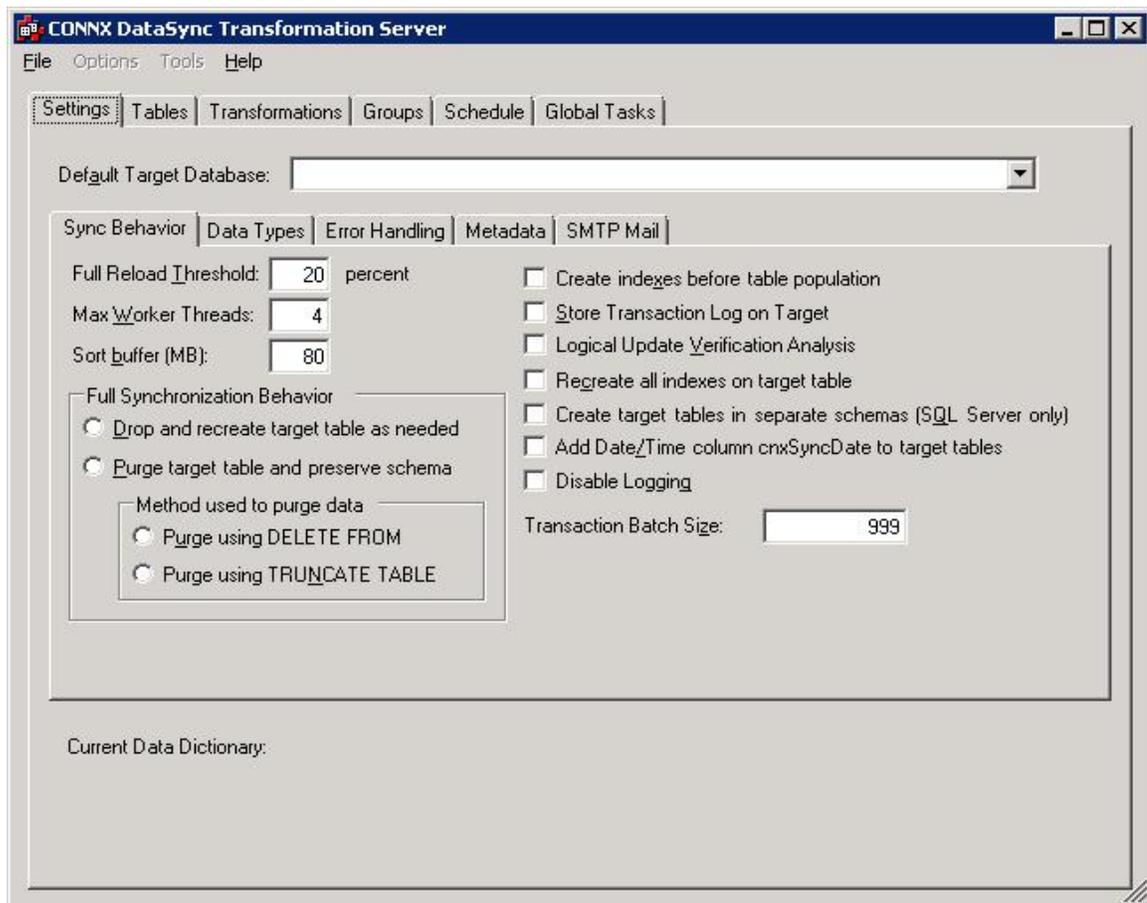
Additional tasks can be triggered based on the status of a synchronization. A task can be as simple as sending an e-mail to notify that a sync was successful or as complex as calling an ETL (Extract, Transform, Load) tool to perform additional transformations against the synchronized data.
- **Transformations**

Transforms can be created to change or manipulate data that is being synced from the source to the target. Transformations, such as column expressions, column mapping, aggregation, lookups, unions, etc. can be developed. This enables DataSync to have ETL (Extract, Transform, Load) functionality.

The CONNX Data Synchronization Tool Window

There are six primary tabs in the DataSync window: **Settings, Tables, Transformations, Groups, Schedule, and Global Tasks.**

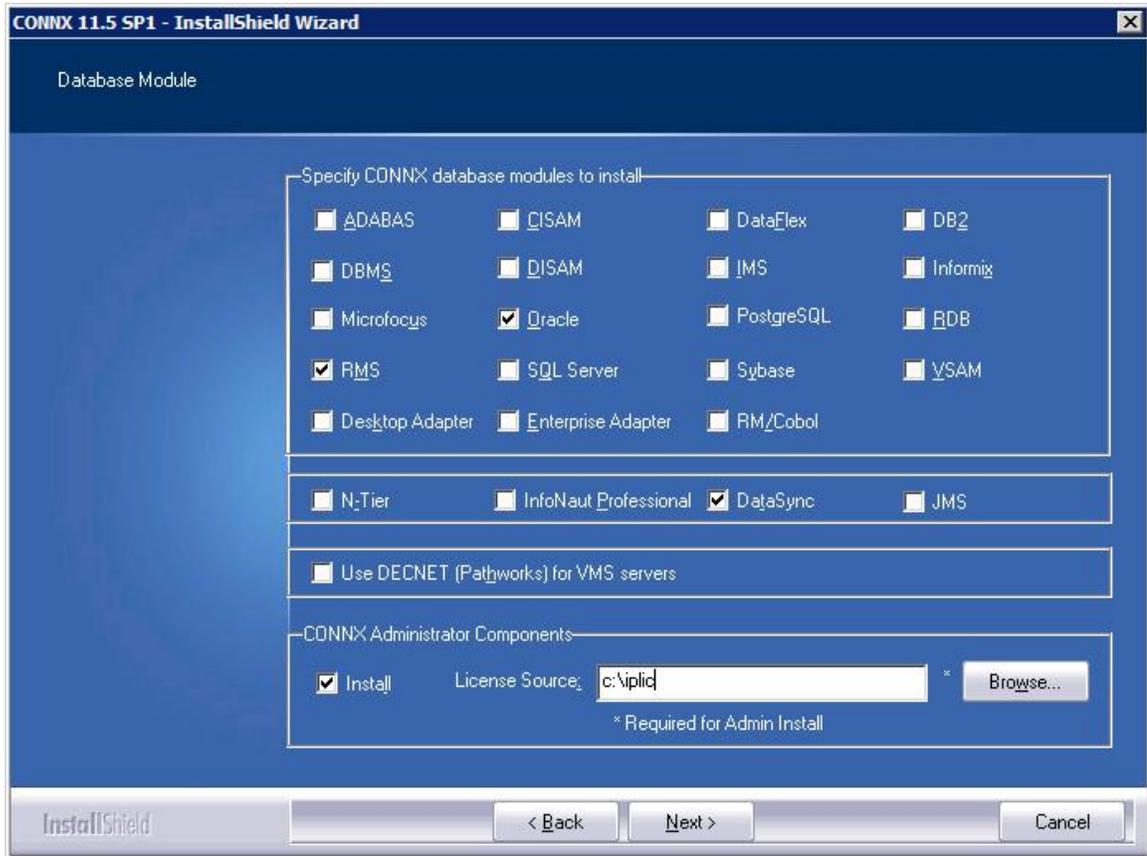
- The **Settings** tab enables the user to select an available database in the CONNX Data Dictionary (CDD).
- The **Tables** tab enables the user to select tables from the CDD for which to start the synchronization process. The Tables tab can be used for testing purposes, for on-demand synchronizations of single tables, and for examination of the status of the synchronizations for each individual table.
- The **Transformations** tab enables source feeds to be transformed into a Target table. The source feeds can include multiple tables, derived columns, aggregates, tables subsets, and expressions. The transformation tab includes a Transformation Wizard, Query Builder, and Expression Editor. (Requires Transformation Server license)
- The **Groups** tab enables the user to place multiple tables into a single group for scheduling purposes. New Groups can be created with the Add button. Existing properties of a group can be modified with the Properties button.
- The **Schedule** tab facilitates the scheduling of individual tables or views, or entire groups. Schedules can be created to occur hourly, daily, monthly, weekly, or yearly. Pre- and post-synchronization tasks can be assigned at the schedule level.
- The **Global Tasks** tab enables users to schedule global and on-demand synchronization tasks and to enter a description of the tasks that are to be performed.



Installing DataSync

DataSync is installed during a CONNX installation. When the Database Module screen is shown, DataSync should be checked. It is necessary to have the appropriate DataSync license. See the **CONNX Installation Guide** for more information.

1. Install CONNX (Version 11 and above). The **DataSync** module must be licensed and selected before continuing with installation. Click the **Next** button.
2. **Important:** *If you already have one or more CONNX licenses, you must select the check box for each database for which you are licensed. Failure to do so will cause the license for that database to uninstall.*



3. Once CONNX and DataSync are installed, you may need to **restart** your computer. Please do so if prompted.
4. Confirm that CONNXStore Database Server Service and the CONNX DataSync Schedule Service are running by going to the **Start** menu. Point to **Settings**, and then click **Control Panel**. Click **Administrative Tools**, and then click **Services**. The **Status** column for both services should read **Started**. Close **Control Panel**.

Synchronization Type Options

DataSync offers four options to synchronize the source data:

Automatic	This is the default option. Except for the first synchronization, when a Full Reload is performed, subsequent synchronizations will attempt to use a CRC algorithm to detect and synchronize only changed, inserted, or deleted records. This can cause dramatic improvement in synchronization times.
Incremental Update	Performs an Incremental Update, if possible. If not, a Full Reload synchronization will be performed.
Full Reload	This will do a Full Reload synchronization from the source to the target.
Incremental Only	In some cases, an Automatic synchronization will revert to a Full Reload; for example, when the number of records changed have exceeded a threshold. When Incremental Only is selected, under NO circumstances will the data be dropped and reloaded to the target. This is advantageous when source data is operational 24x7.

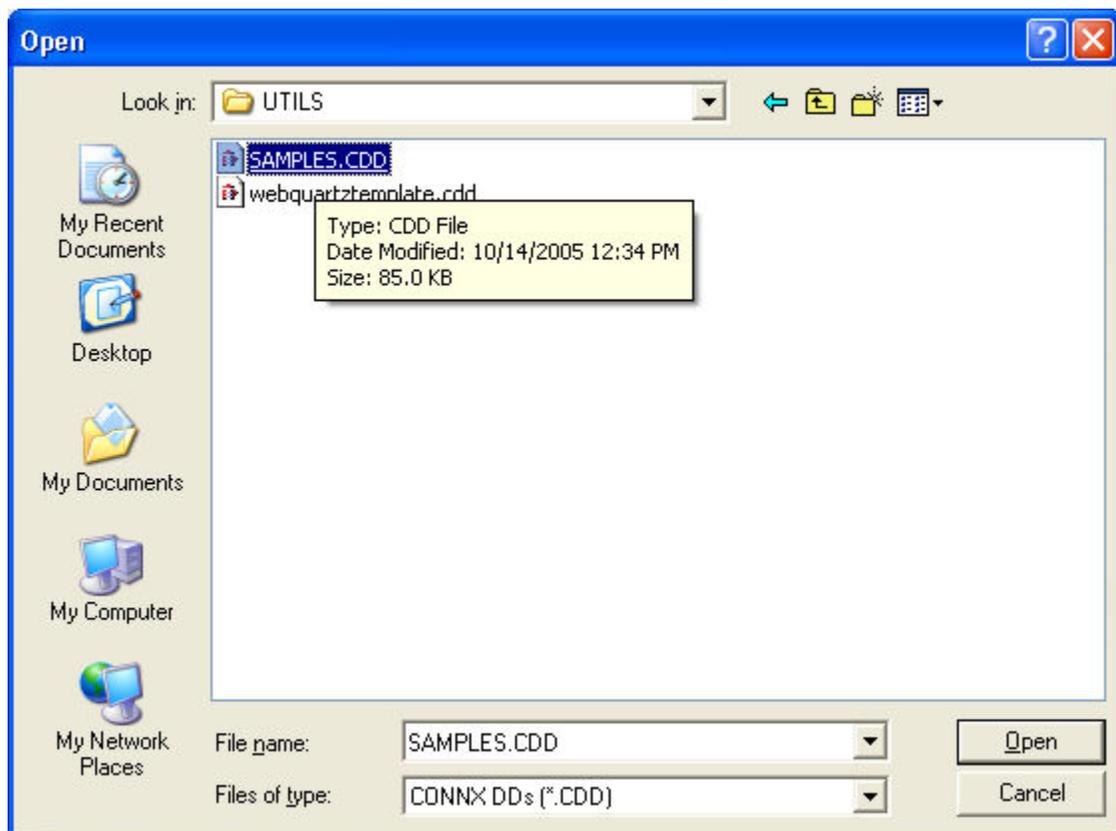
Getting Started

The following example goes step-by-step through an example of synchronizing a source feed.

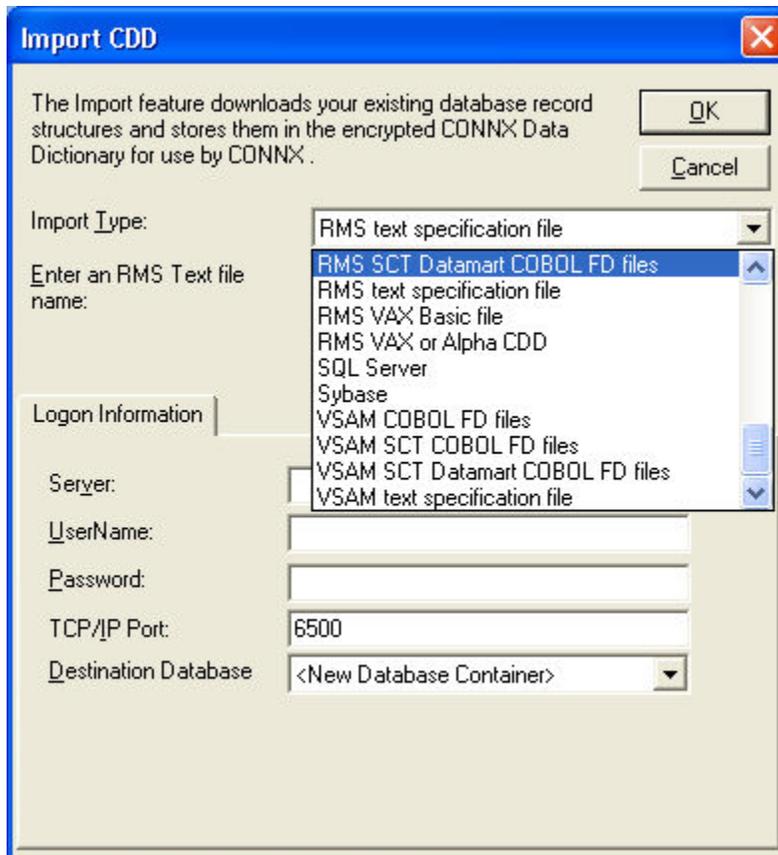
1. DataSync requires a CONNX CDD File. Click the **Start** button and then point to **All Programs**. Point to **CONNX Driver** and then click **CONNX Data Dictionary**. The **Open** dialog box appears.
2. Build a .cdd as described in the **CONNX User Reference Online Help** or **CONNX User Guide**, or select a .cdd file from the list shown. Click the **Browse** button to locate a .cdd file on the network, or click the **Cancel** button if you intend to create a new .cdd in the **CONNX Data Dictionary Manager**.

There must be one or more source database in the .cdd file and one or more target database, although the selected target database does not need to be populated initially.

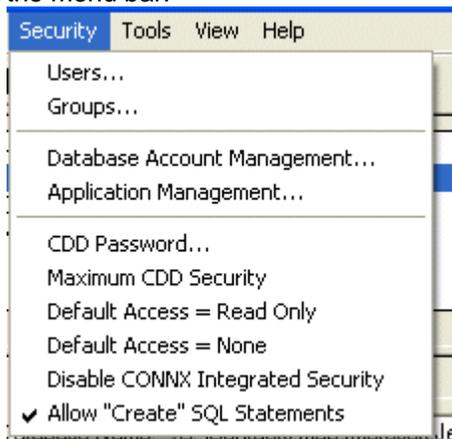
Important: You will be using the same .cdd file as both synchronization data source and target so it must contain more than one database type.



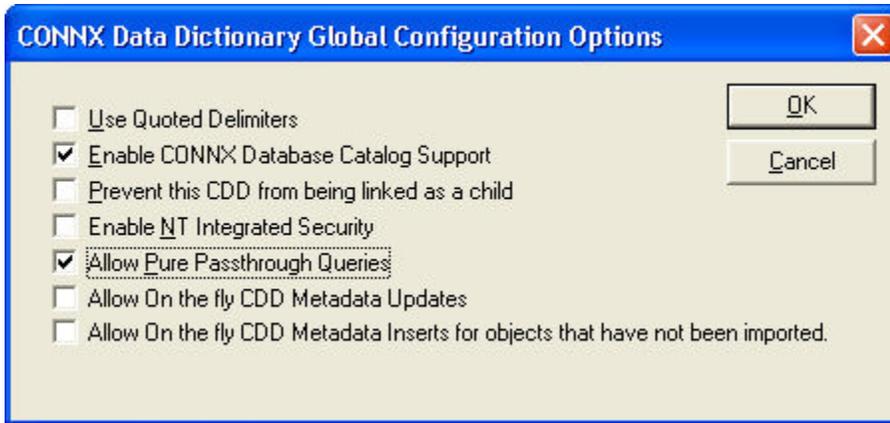
3. If you are creating a new .cdd and are importing a target database type, click the **Include System Tables** check box in the Import CDD dialog box and then select at least one of the tables for the targeted data warehouse or datamart.



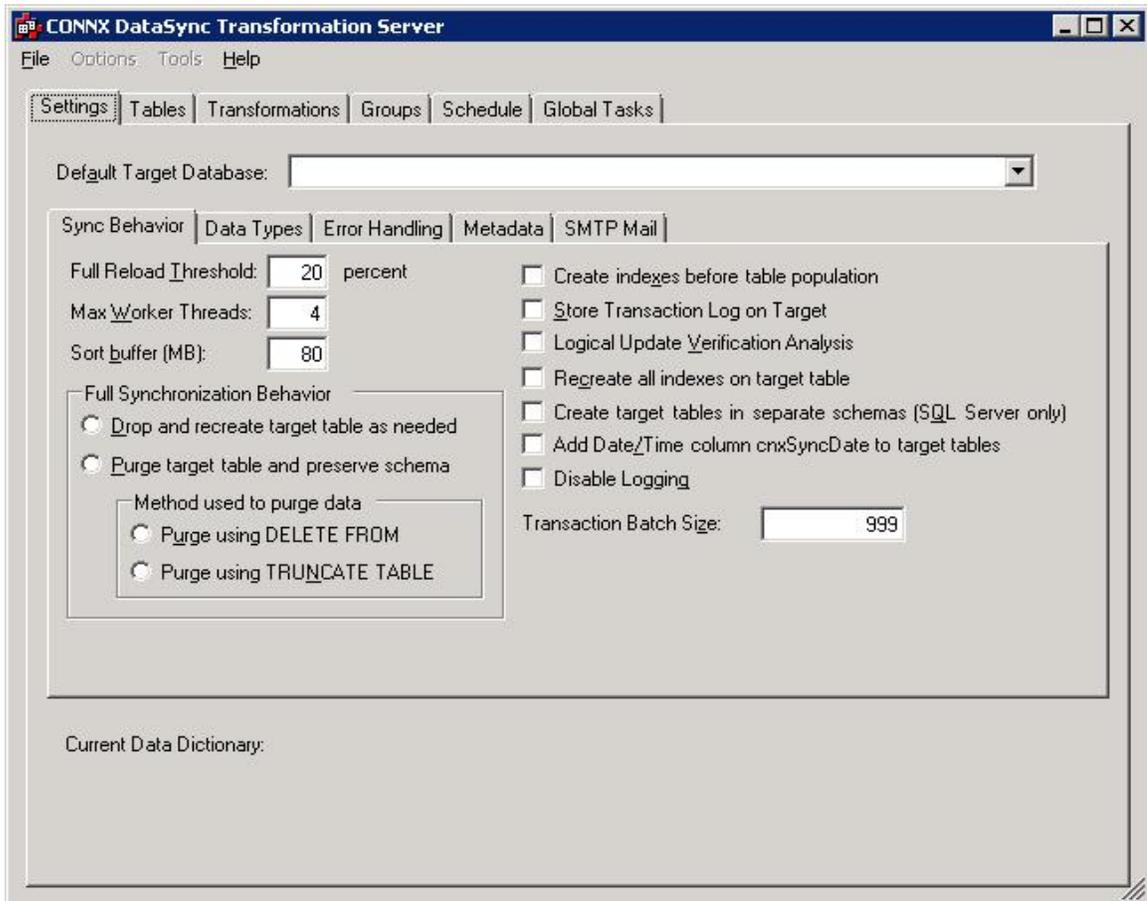
4. Verify the CONNX settings required by the CONNX Data Synchronization Tool. Click **Security** on the menu bar.



5. Verify that **Default Access = Read Only** and **Default Access = None** are unchecked.
6. Select **Tools** on the menu bar, and then select **Options**. The **CONNX Data Dictionary Global Configuration Options** dialog box appears. Verify that **Enable CONNX Database Catalog Support** is checked and that **Allow Pure Passthrough Queries** is checked.



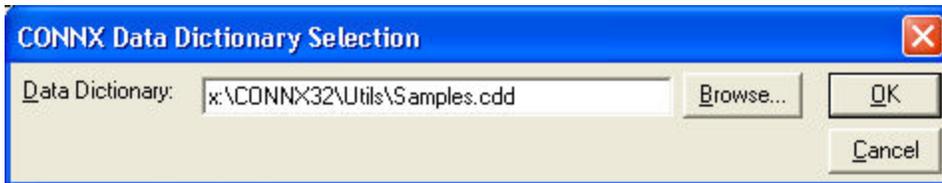
7. Click the **OK** button to return to the CONNX Data Dictionary Manager. On the **File** menu, click **Save**, and then click **Exit**.
8. Click the **Start** button, and then point to **All Programs**. Point to **CONNX DataSync**, and click **CONNX DataSync Administrator**. The CONNX Data Synchronization tool window appears.



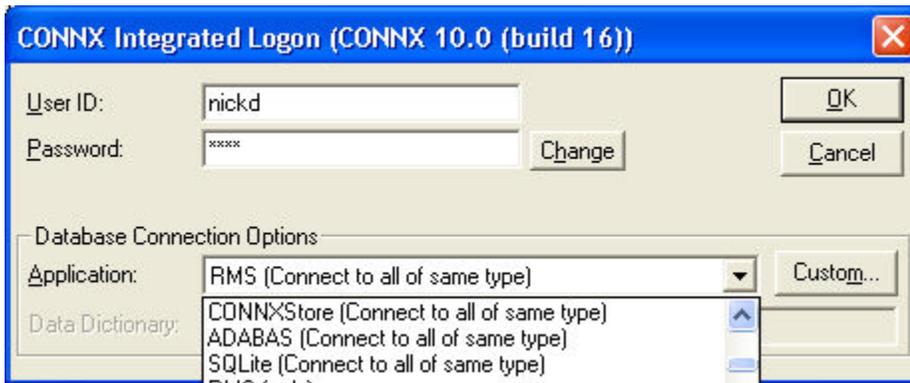
9. In the CONNX Data Synchronization Tool, under **File** on the menu bar, select **Select Data Source**. The **CONNX Data Dictionary Selection** dialog box appears.



10. Enter the name of a CONNX Data Dictionary (CDD) in the Data Dictionary text box, or click the **Browse** button to locate a CDD, and then click the **OK** button.

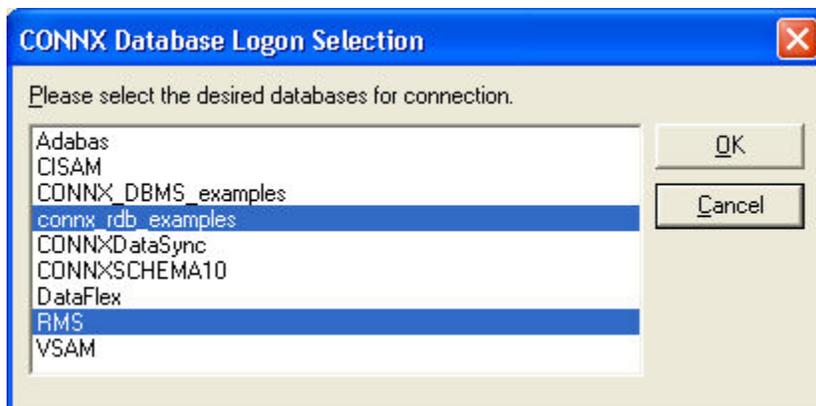


11. The **CONNX Integrated Logon** dialog box appears. Enter a user id used with the .cdd you selected, along with a password. Click the **Change** button to change the password.



12. Click the **Custom** button to customize the databases you wish to use with the CONNX Data Synchronization tool.
13. The **CONNX Database Logon Selection** dialog box appears. Select the databases you wish to use for synchronization, and then click the **OK** button.

Note: CONNXDataSync may appear in the list of database selections depending on your initial setup of CONNX. Select it along with your source and target databases. If CONNXDataSync does not appear, it is automatically created.



14. Click the **OK** button in the CONNX integrated logon dialog box to continue. If you add new databases to a .cdd during synchronization tasks, you must log on to the database server.
15. The **CONNX Data Synchronization Tool** appears. In the **Settings** tab, click the **Sync Behavior** sub-tab. Select a target in **Default Target Database**. The Target Database is the location in which you intend to create a data warehouse or repository that contains copies of data tables from the other database systems in your CONNX CDD. For instance, if you wanted to duplicate your VSAM files in an Oracle database, the Oracle database is your Target Database.
16. Enter a number in **Full Reload Threshold** text box. If more than this percentage of rows has changed in a table, you must perform a full reload instead of an incremental update on your data. Normally, this number should be fairly small, in the range of 3 to 20 percent. The actual value that is best for your system depends on many variables and offers an opportunity for tuning your system to its maximum performance levels.



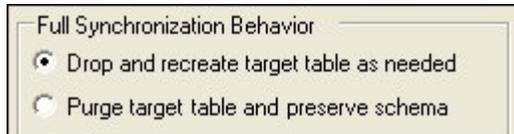
17. Enter a number in the **Max Worker Threads** text box. (Recommended = 4 threads per CPU on the Data Sync machine) The maximum number is the number of threads on the PC (and also the number of servers on the host database server) that can be used when processing the synchronization. For example, if there are 1000 tables of the same size, 1 out of 3 threads would process about 333 tables each. Increasing the number of threads uses more resources on the PC and on the host database server, but can also increase performance.

On the other hand, increasing the thread count to a level that is too high can severely impact performance and possibly cause failures. It is strongly recommended that you do not exceed ten threads per CPU available on the Data Synchronization machine and also to consider the system's total memory, since each thread can allocate large blocks of memory when processing large tables (see Step 33 for more information).

In order to test the effects of multiple threads, create a group and then synchronize the group.

Tip: A few tests with different threading values can determine a good level for Max Worker Threads for your system through examination of the group synchronization time.

18. In the **Sort buffer** text box, enter the maximum amount of memory used to sort files per thread. The number should reflect the amount of memory available on the machine used as the sorting buffer and the number of threads used. The actual amount used may be much smaller if the tables being synchronized are small. The default sort buffer of 80 megabytes may be compatible with most systems that follow our recommended guidelines for minimal machine configuration.
19. In the **Full Synchronization Behavior** combo box, select the **Drop and recreate target table as needed** radio button to perform that function. If, for example, the schema has been altered on the source table, the target table reflects the changes. This feature also makes it faster to drop and recreate data than to delete rows and refill them. This setting is the default.



20. Select the **Purge table and preserve schema** radio button to delete all the data in the target table and repopulate the data by insertion. This method works to the advantage of people who make custom modifications to the target schema. Preserving the schema allows the modifications to be saved. However, the system is no longer "self-healing" to schema changes on the source table; the only way a new copy of the table is created on the target service is if the table does not already exist on the target.

There are two options for Purge target table and preserve schema:

- **Purge using DELETE FROM**

Choose the Delete feature when all deletes are logged and can be rolled back in case something goes wrong. If there is an error on a Truncate command, the database is left in the state of the error, as the command is permanent. If a Delete command has problems, the database rolls back to the previous state on transaction failure.

- **Purge using TRUNCATE TABLE** (for target servers where truncate is available).

The CONNX Data Synchronization tool may perform a Truncate more quickly than a Delete. Choose Truncate if:

- You do not want to delete indexes in the target table that are not in the source table. This could happen if the target table is used for specialized reporting.
- Truncate does not activate triggers, referential integrity constraints, or other database restrictions. Since database integrity is maintained on the source database for the synchronization, choose Truncate Table when performance is a concern

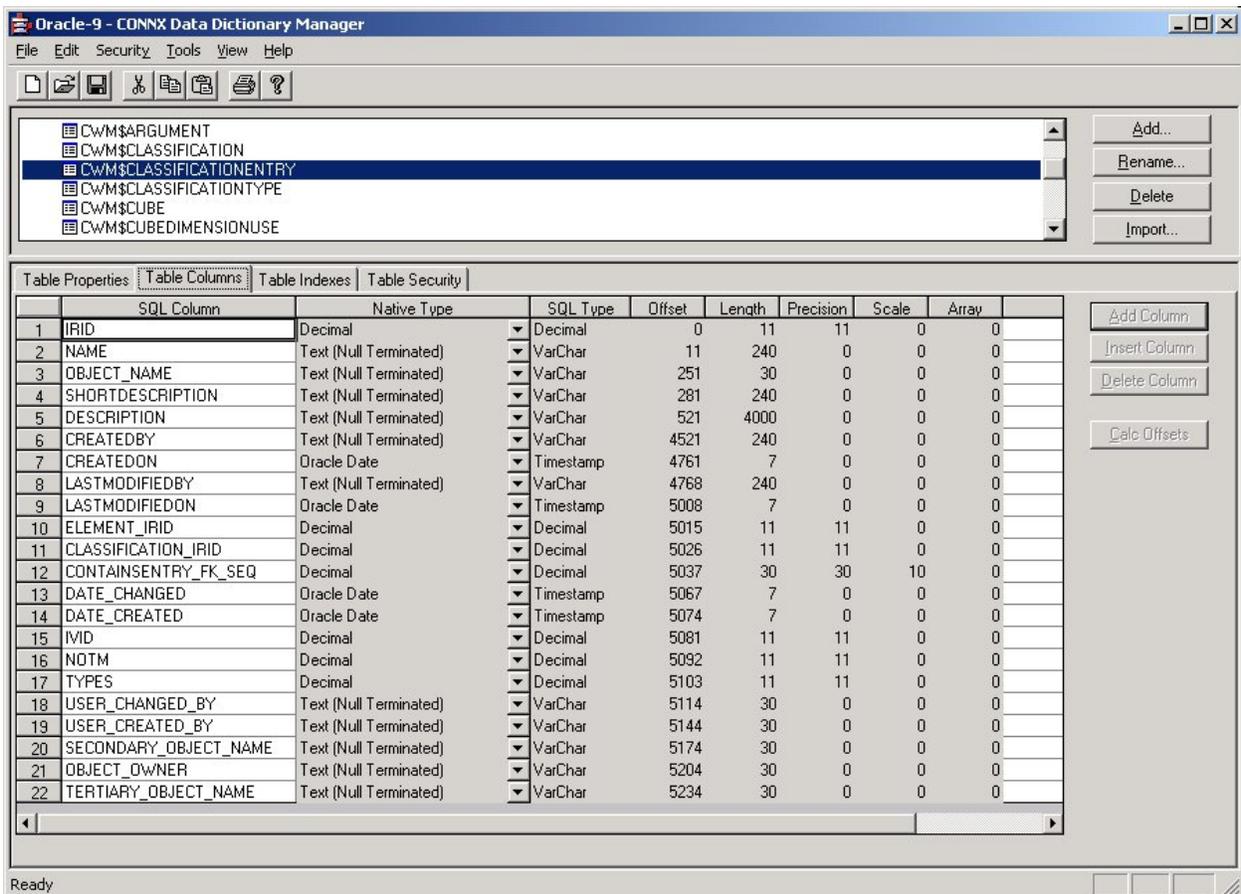
21. To reorder the synchronization process so that indexes are created first, click **Create indexes before table population**. This can have either a negative or positive effect on performance and space requirements of a full synchronization, depending upon the database system. For instance, if the target is Oracle, it will be faster to create the indexes after the load of the data. Inserting rows with an existing index updates the indexes on every insert, whereas creation of an index at the end of the synchronization is somewhat more efficient.

On the other hand, with RMS as the target system, if indexes are created as the synchronization completes, then two entire copies of the file will exist on the target system, as a sequential file is converted into an indexed file with a CONVERT/FDL command. The best choice depends strongly on the type of database system you are using and on the amount of resources available for performing the synchronizations.

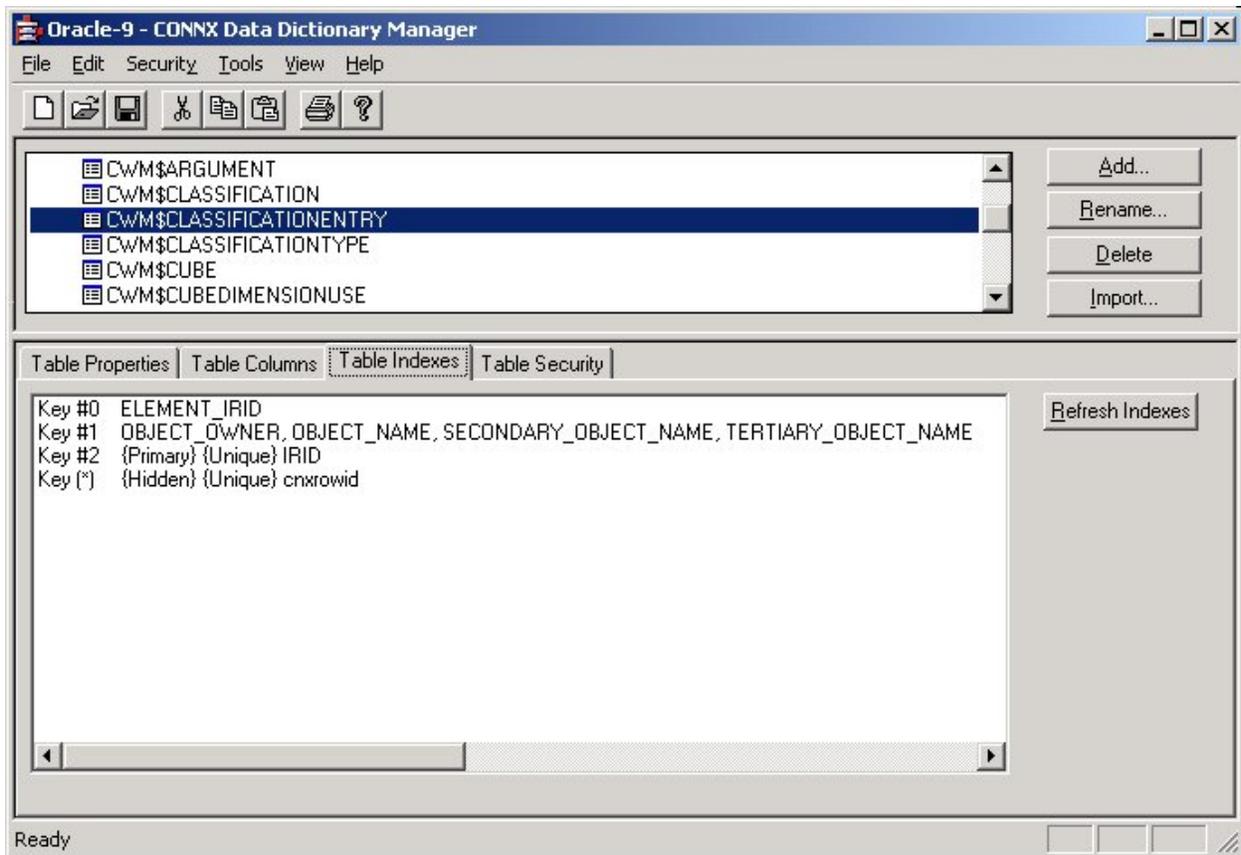
22. To create a transaction log which is stored in the target database, click the **Store Transaction Log on Target** check box. If this option is not selected, the transaction log is not created. A transaction log is a table that contains the unique index information for each row that was inserted, updated, or deleted along with a flag that tells what sort of action was performed on each row. By using a transaction log, systems that need an audit process can tell exactly what was changed and what sort of operation was performed on the data. It should be noted, however, that the creation of a transaction log may slow synchronizations.

Create indexes before table population
 Store Transaction Log on Target
 Logical Update Verification Analysis
 Recreate all indexes on target table
 Create target tables in separate schemas (SQL Server only)
 Treat CHAR as VARCHAR on Target Table at or more
 Synchronize Blobs
 Add Date/Time column cnxSyncDate to target tables

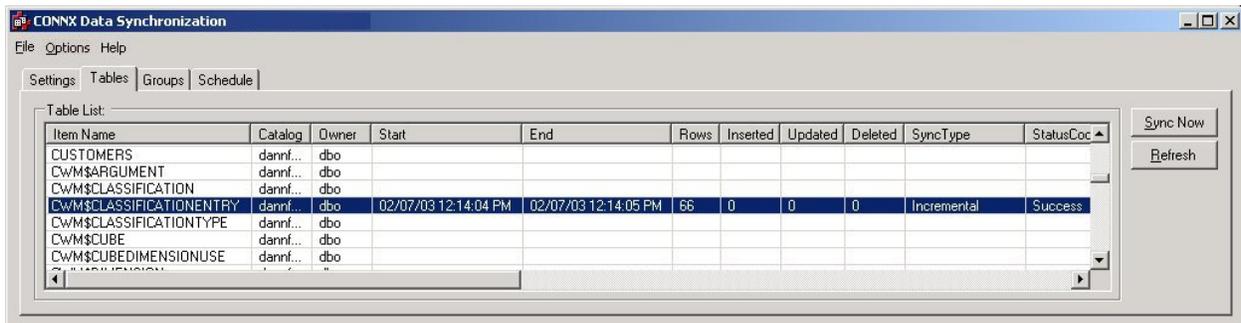
In the following example, an Oracle table "CWM\$CLASSIFICATIONENTRY" was synchronized.



A unique ID, IRID, appears on the Table Indexes tab.

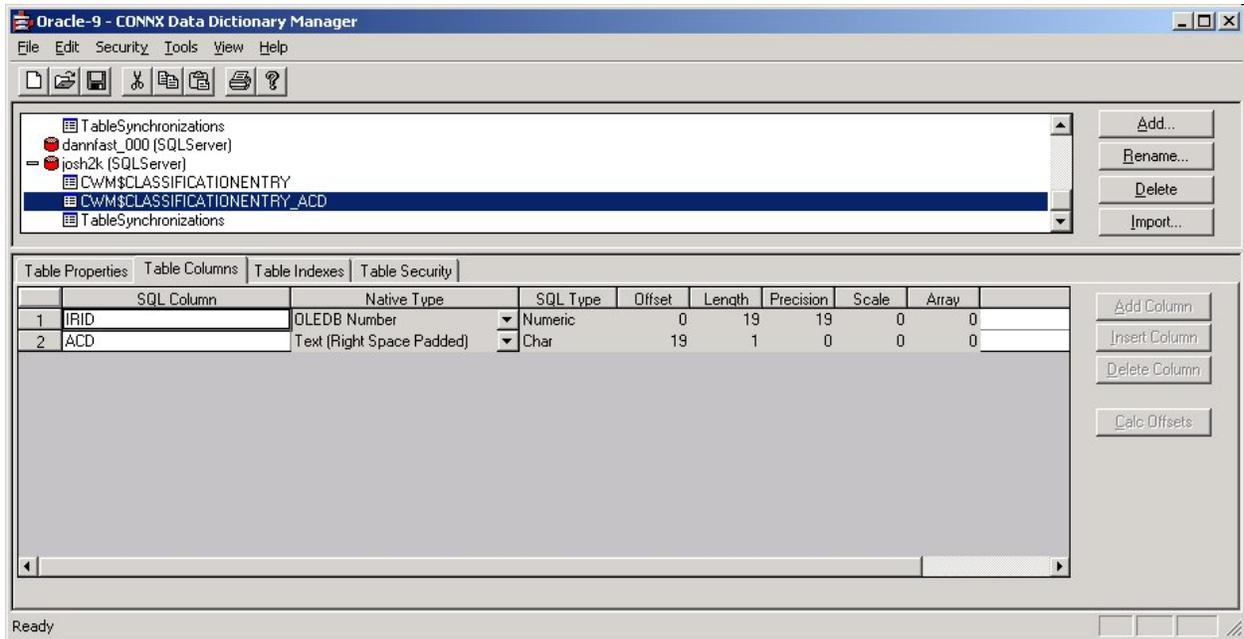


The table is successfully synchronized and appears in the CONNX Data Synchronization tool on the Tables tab.



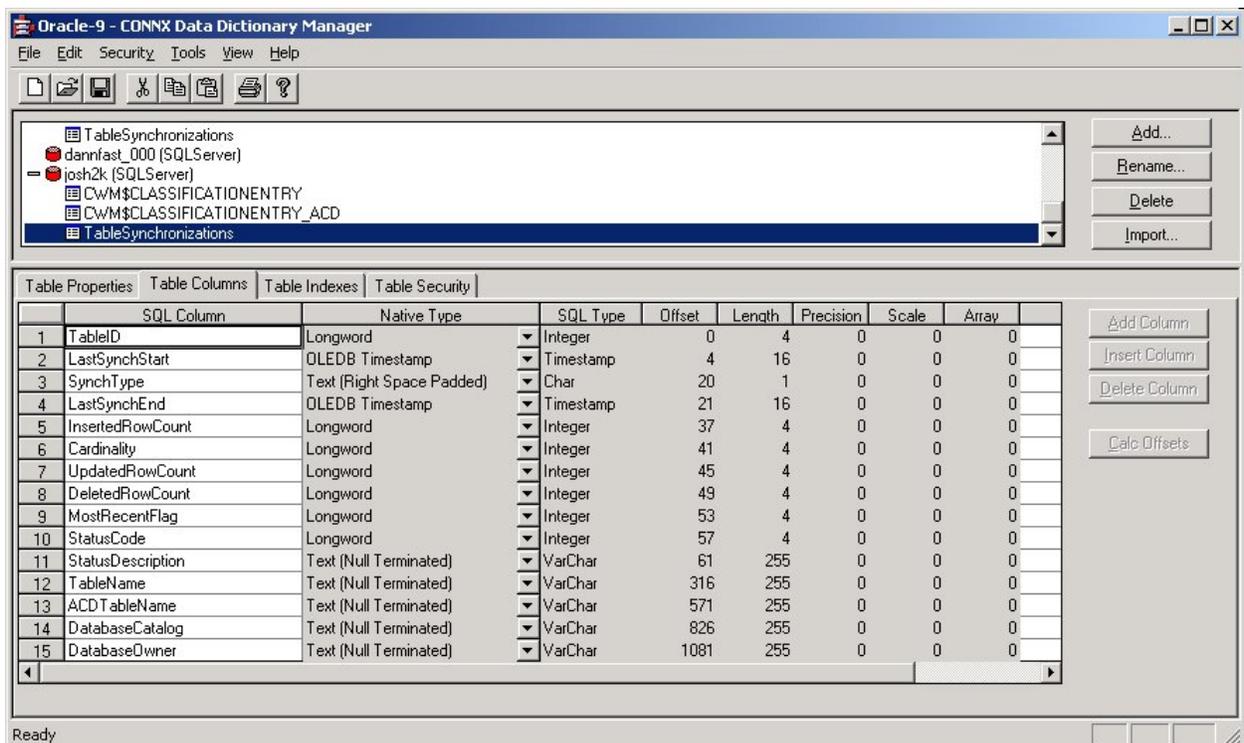
In fact, two tables have been created on the target system, CWM\$CLASSIFICATIONENTRY and CWM\$CLASSIFICATIONENTRY_ACD, which contains the index information and operation type of all of the adds, changes, and deletes performed on the target table.

CONNX Data Synchronization 12.5



If information has been inserted, then column ACD will be equal to 'I'. If information has been changed by an update of an existing row, then column ACD will be equal to 'C'. If information has been deleted from the source table, then column ACD will be equal to 'D'.

In addition to the transaction table, a comprehensive table called TableSynchronizations in the target database tracks what has been happening for all of the tables being synchronized.



The following definitions apply to the above example:

- TableID is an INTEGER that uniquely identifies each table.
- LastSynchStart is a DATE that tells when the synchronization started.
- SynchType is a single CHAR that tells if the synchronization was full or incremental.
- LastSynchEnd is a DATE that tells when the synchronization ended.
- InsertedRowCount is an INTEGER that tells how many rows were inserted for the synchronization. It will be zero for a full synchronization. For a full sync, the number of rows processed is stored in the cardinality field.
- Cardinality is an INTEGER that tells how many rows are in the synchronized target table.
- UpdatedRowCount is an INTEGER that tells how many rows were updated for this synchronization. It is zero for a full synchronization.
- DeletedRowCount is an INTEGER that tells how many rows were deleted in the target table for this synchronization. It is zero for a full synchronization.
- MostRecentFlag is an INTEGER that is reserved for future usage. Currently, it is set to 1.
- StatusCode is an INTEGER that indicates success or failure. A value of zero indicates failure. A value of -1 indicates success.
- StatusDescription is a VARCHAR(255) field that describes any errors that occurred. A more detailed message is found in the data synchronization log.
- TableName is a VARCHAR(255) field that contains the name of the table synchronized.
- ACDTableName is a VARCHAR(255) field that holds the name of the key data transaction log. In the above example, for TableName CWM\$CLASSIFICATIONENTRY, this field contains the value "CWM\$CLASSIFICATIONENTRY_ACD".
- DatabaseCatalog is a VARCHAR(255) field that contains the name of the database catalog for the synchronized table.
- DatabaseOwner is a VARCHAR(255) field that contains the name of the database owner for the synchronized table.

23. [Logical Update Verification Analysis](#) is used in conjunction with the [Store Transaction Log on Target](#) option. For each change verified, the target table transaction log will change the 'C' (for changed) in the ACD column of the transaction log to 'V' (for verified). If Logical Update Verification Analysis is selected, for each row that has had a change detected via CRC differences, a column by column search is performed to confirm a difference.

This option should be selected in rare cases where the physical bits in the data have changed, but the actual meaning remains the same. Some examples might include the following:

- A. Multiple physical values with different internal representations but the same physical meaning, such as:
 - a. Positive and negative zero on one's complement machines
 - b. Positive signed and unsigned decimal numbers (e.g. signed + 1 and unsigned + 1 have the same meaning but different internal representation)
- B. Insignificant changes. If, for example, the very last bit of a floating point number changed in the mantissa, the actual meaning of what is stored may not really have changed. For instance, someone might have edited a field that contained the value "2.1" as a floating point number. Though no changes were made, the value was saved again into the database. It is possible that the number will differ slightly due to the nature of floating point. A comparison shows that both the source and target numbers are intended as 2.1, but a CRC comparison shows that some bit has changed and consider them as completely different.
- C. Masked values. If, for example, a null-terminated string is stored in the source database, anything after the null string terminator is a physical difference that causes a different CRC,

but the data stored may be the same. For instance, the string of bytes 'D', 'o', 'g', 0, 'f', 'o', 'o', 'd', 0 will be interpreted simply as "Dog" since a null terminator (zero) follows the 'g'. The string of bytes 'D', 'o', 'g', 0, 'f', 'i', 'g', 'h', 't', 0 will also be interpreted simply as "Dog" since a null terminator (zero) follows the first 'g'. Anything beyond the null terminator has no meaning as far as the database is concerned. But CRC values will indicate that there has been a change.

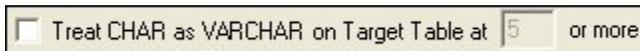
24. **Logical Update Verification Analysis** should only be used in special cases, where a careful analysis has shown that more data is being processed than is necessary. Selecting this option makes the synchronizations run more slowly, since for every change detected, every single column in the record is compared from source to target - one a column at a time. If a post-process step that is very slow must be performed on the changes (for example some expensive ETL operations) then logical update verification analysis might be worthwhile.

25. By default, only the unique index used to perform incremental updates is created on the target table after synchronization. check the **Recreate all indexes on target table** check box to recreate all indexes from the source table on the target table.

26. If you are using SQL Server as a target database, click the [Create target tables in separate schemas \(SQL Server only\)](#) check box to create target tables under different SQL Server owners. DataSync creates a SQL Server user for the purpose of creating tables with identical table names from different source catalogs. For example, if you want to synchronize both the test and production version of a table into a single SQL Server database, CONNX fully qualifies the target name (product.table or test.table) so that there is no name collision.



27. On the **Settings** tab, click the **Data Types** sub-tab. To reduce the amount of space required by the synchronization of source files using CHAR () fixed-width data types, click the [Treat CHAR as VARCHAR on Target Table](#) check box. You can change CHAR () fixed-width data types to VARCHAR data types by entering the length in the text box. In this example, CHAR () fixed-width data types of five or more characters are automatically converted to VARCHAR data types. Determining this threshold is very important, since very small VARCHAR fields waste more space than they save. They are also less efficient on some database systems when performing certain tasks, such as indexing.



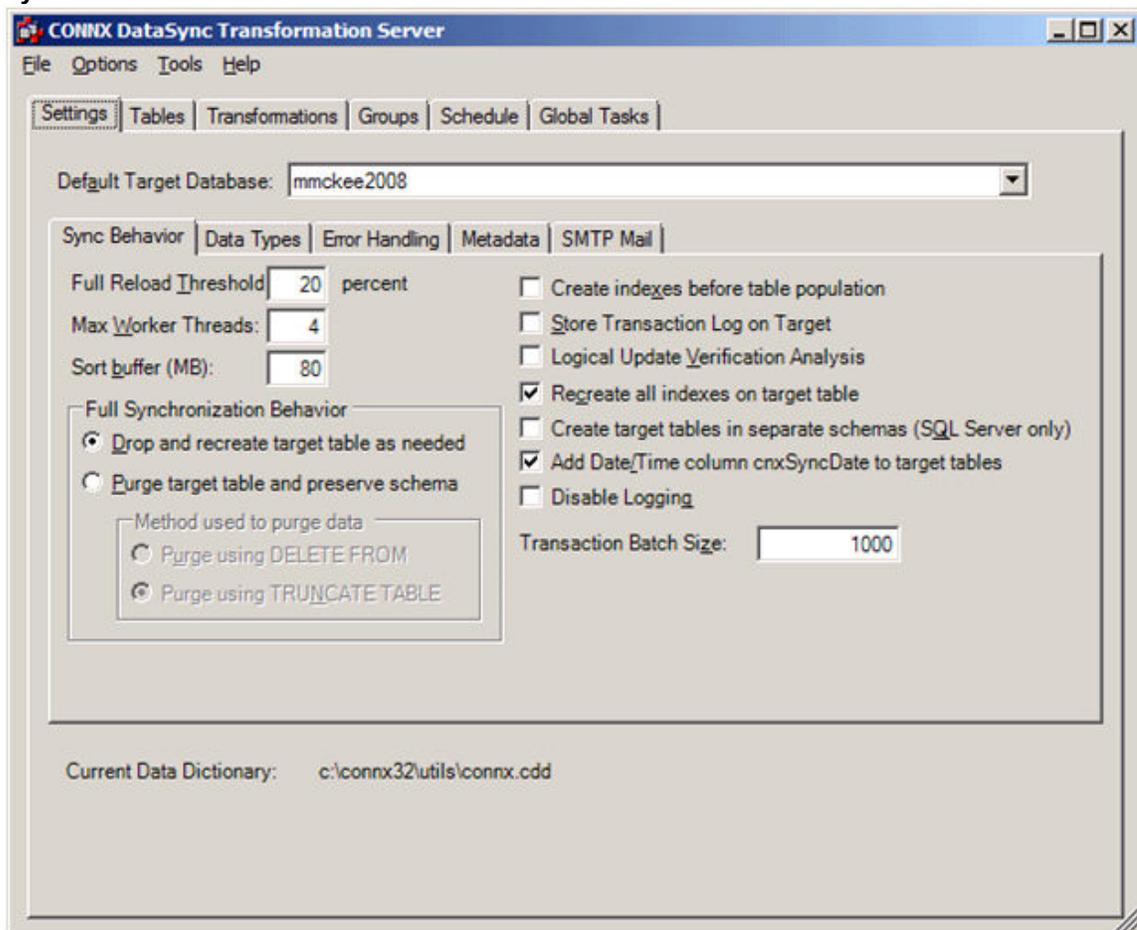
28. Continue with the synchronization processes as described in [To select a CONNX Data Dictionary for the data synchronization process](#).

Settings Tab

Setting Tab sub-tabs

The Setting Tab has five sub-tabs:

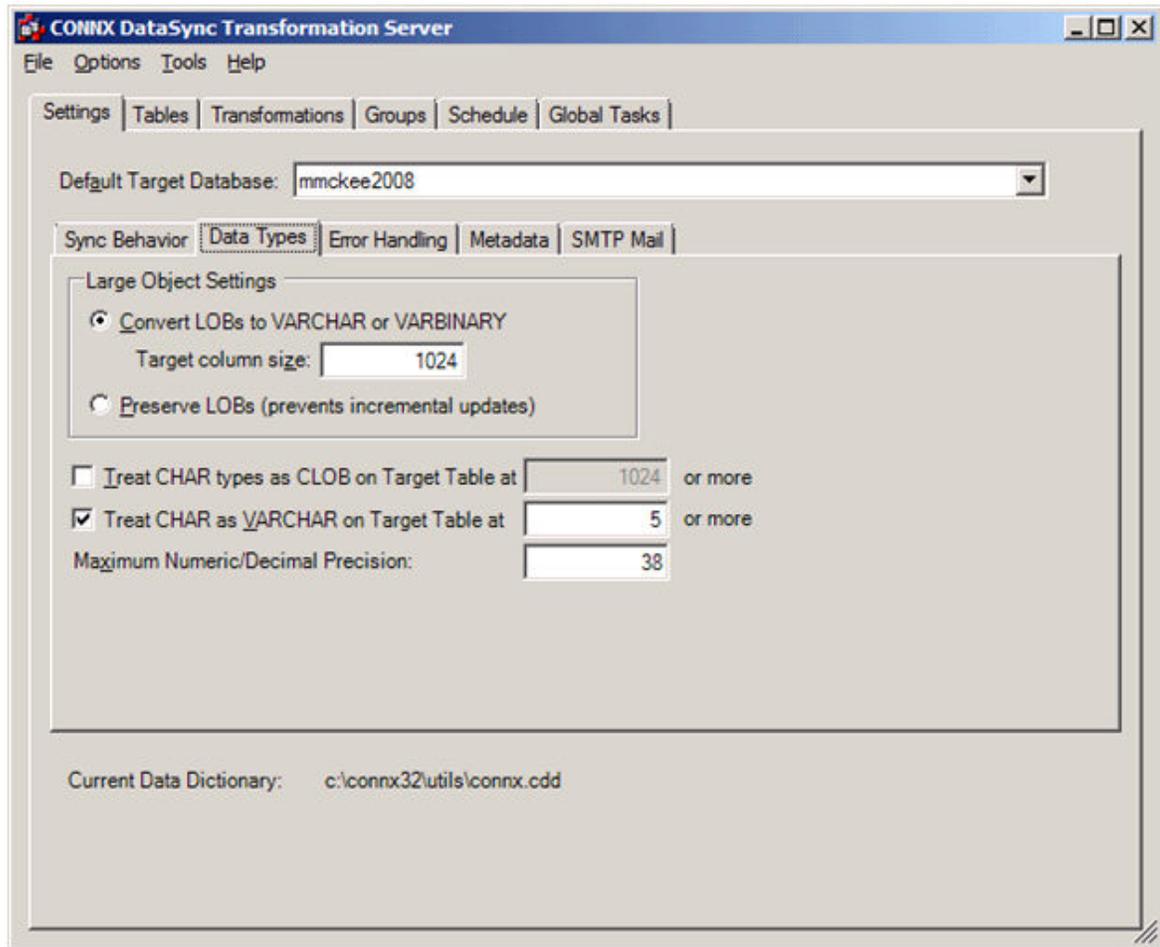
- **Sync Behavior**



Functions in the Sync Behavior sub-tab include:

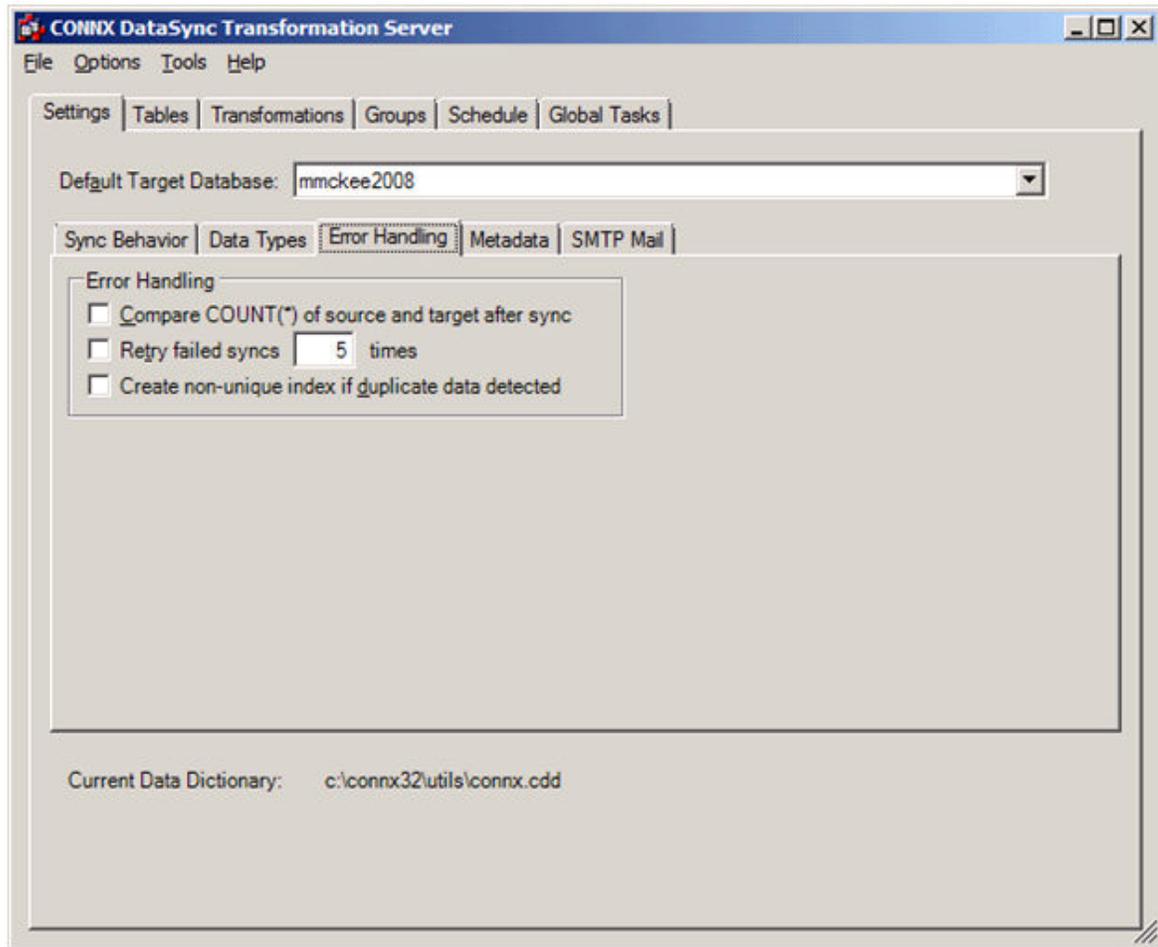
- Setting the target database to synchronize
- Purging target tables and preserving the schema
- Creating indexes before loading the data
- Storing transaction logs on the target database
- Verifying and analyzing updates
- Recreating indexes on the target table
- Creating a non-unique index for duplicate data (SQL Server)
- Adding date/time column to target tables
- Disabling logging

- **Data Types**



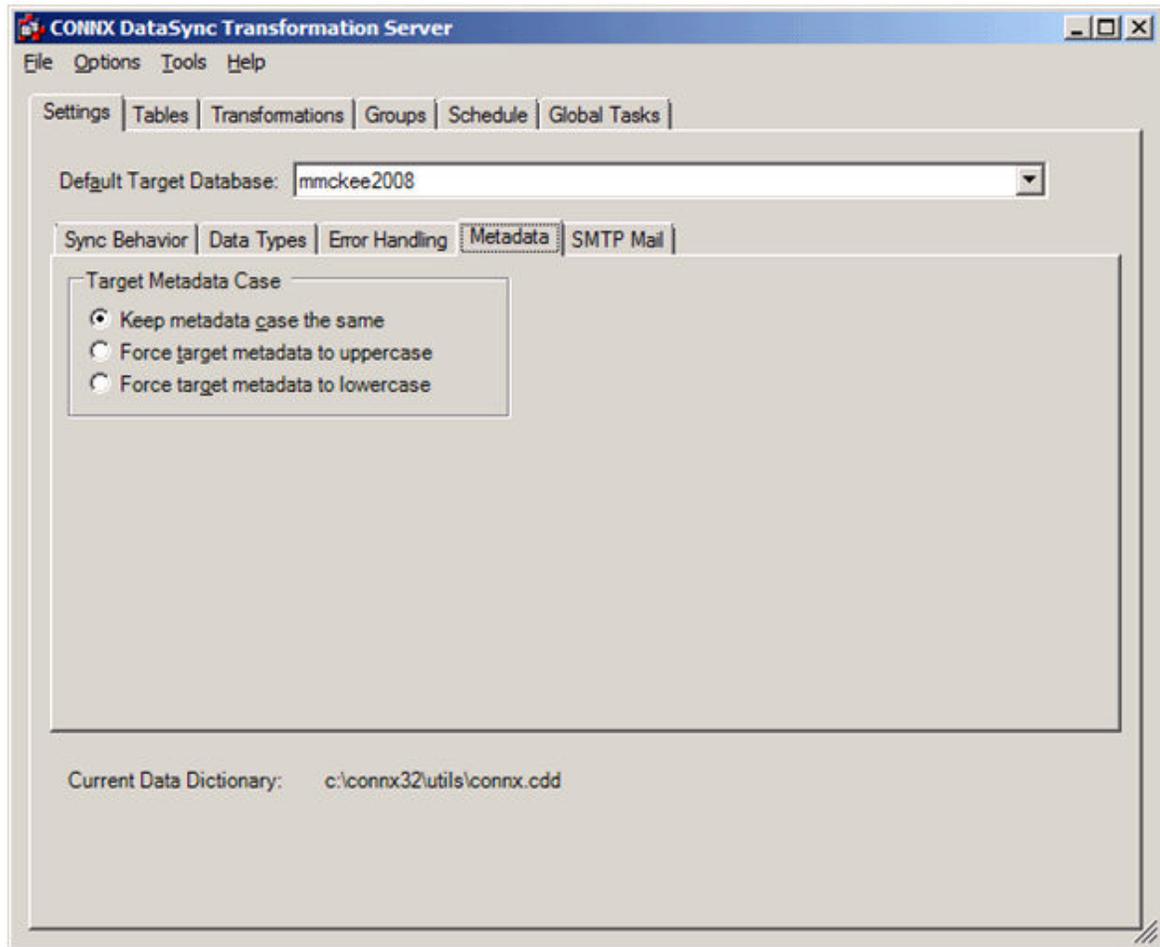
Functions in the Data Types -sub-tab include:

- Synchronizing blobs
 - Treating CHAR data types as CLOB
 - Treating CHAR data types as VARCHAR
 - Selecting the maximum numeric or decimal precision
- Error Handling



Functions in the Error Handling sub-tab include:

- Comparing source and target COUNT(*)
 - Retrying failed synchronizations
 - Creating non-unique indices for duplicate data
- Metadata

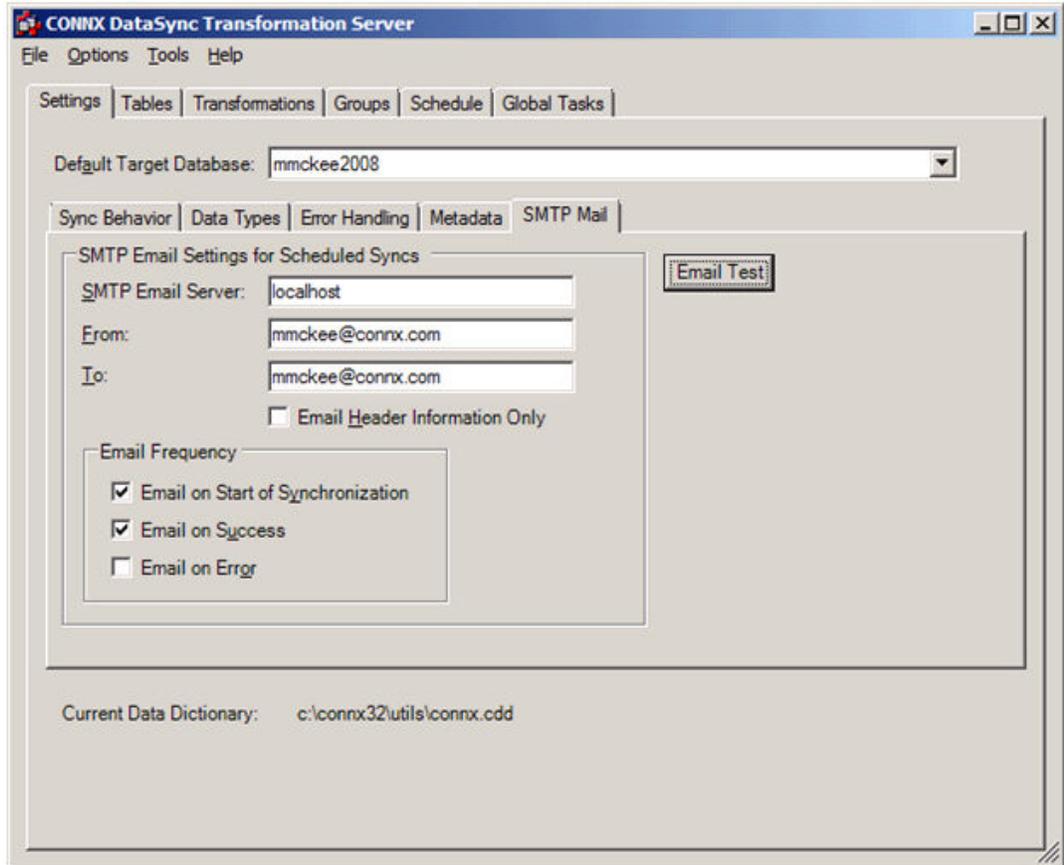


Functions in the Metadata sub-tab include:

- Specifying the metadata case
- SMTP Mail

Functions in the Metadata sub-tab include:

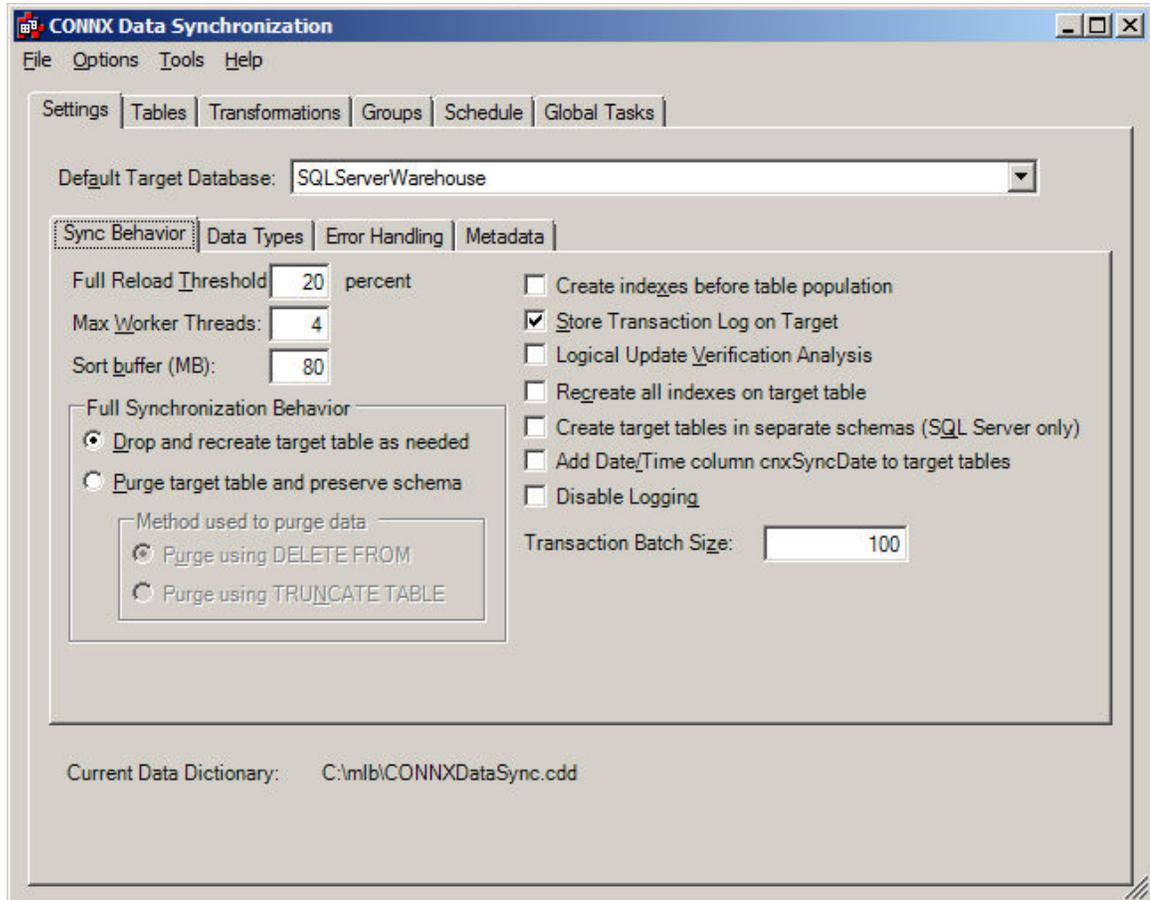
- Sending status on synchronizations from a SMTP Mail Server to a client or group.



Sync Behavior sub-tab

To select a target database for synchronization

1. In the **Settings** tab, click the **Sync Behavior** sub-tab.

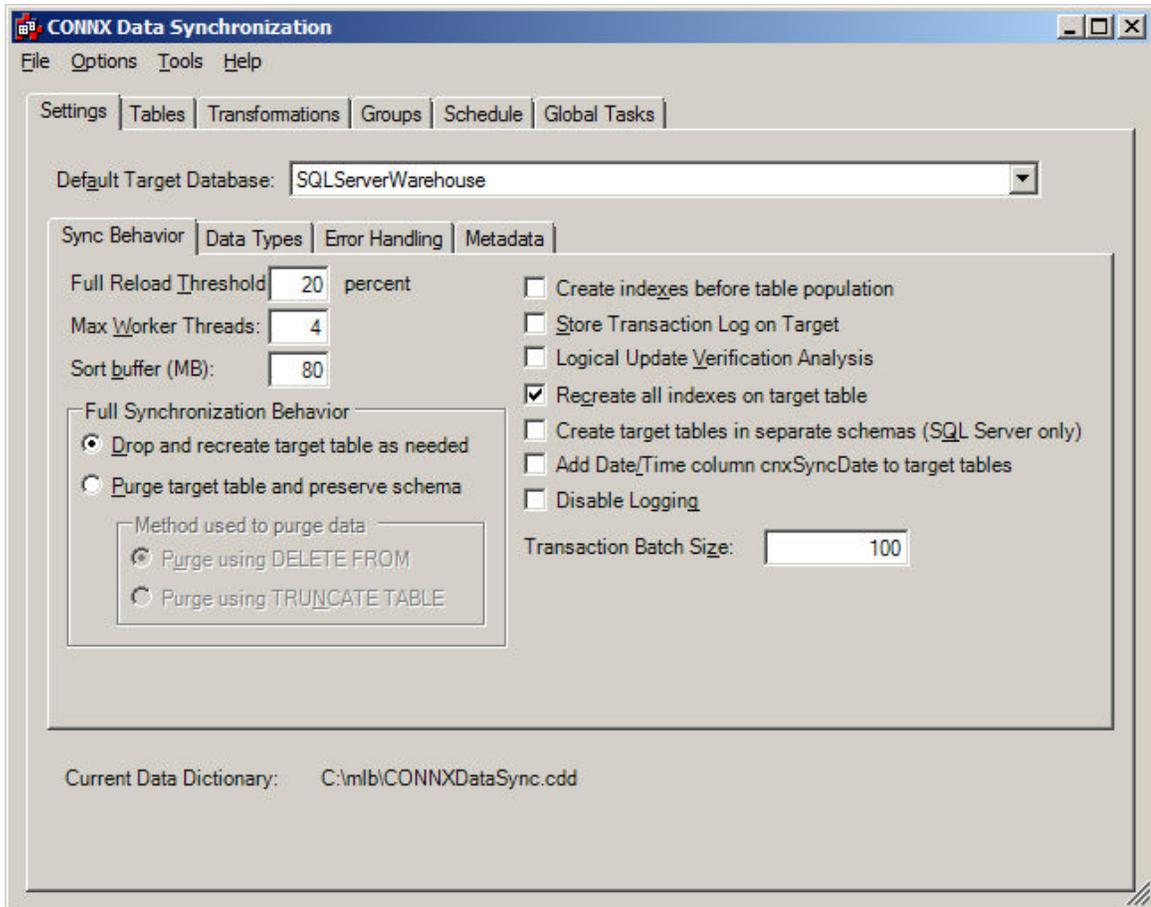


2. Select the target database for the synchronization task from **Default Target Database**. This should have already been established in the CONNX Data Dictionary Manager window. The target database should be one of the following database types: Rdb, RMS, Oracle, SQL Server, IBM DB2, MySQL, C-ISAM, and Informix and Sybase through the CONNX ODBC Adapter.
3. The percentage amount that appears in **Full Reload Threshold** is the number that will determine if it is more efficient to perform a full reload or an incremental update based on the percentage of rows changed in the source table. This number should be very small.
4. Enter a number in **Max Worker Threads**. The maximum number is the number of threads on the PC (and also the number of servers on the host database server) that can be used when processing the synchronization. For example, if there are 1000 tables of the same size, 1 out of 3 threads would process about 333 tables each. Increasing the number of threads uses more resources on the PC and on the host database server, but can also increase performance.
5. In **Sort buffer (MB)**, enter the maximum amount of memory used to sort files per thread. The number should reflect the amount of memory available on the machine used as the sorting buffer and the number of threads used. The actual amount used may be much smaller if the tables being synchronized are small. The default sort buffer of 80 megabytes may be compatible with most systems that follow our recommended guidelines for minimal machine configuration.

6. In **Transaction Batch Size**, enter the number of rows that will be bulk copied (if the target database supports it) into to the target database in a single operation during a full sync.

To control Full Synchronization Behavior

1. In the **Settings** tab, click the **Sync Behavior** sub-tab. Select the target database for the synchronization task from **Default Target Database** list.



2. There are two kinds of ways to process a full synchronization. One is to drop and recreate the schema on the target table every time. The other is to delete the data in the target table. When deleting data, we can either truncate the data or delete the data. **Truncation** is typically much faster, but not all systems support truncate. If you select truncation, and the target does not support truncation, you will receive a warning and **Delete** will be used instead. An advantage to dropping and recreating the schema is that changes in the source table will automatically be reflected in the target and so the system is self-healing. However, there may be customizations on the target server. In this case, we want to purge the data instead. **Truncation**, while faster, is not typically logged and therefore the user will have to redo a failed sync. **Deletion** works on every system, but is much slower than dropping and adding the schema or truncation.

If the target database does not support truncation, a warning appears for every table selected for deletion even though the synchronization continues and succeeds.

There are two options for **Full Synchronization Behavior**:

Drop and recreate target table as needed is the default behavior.

Select **Purge target table and preserve schema** if:

- You do not want to delete indices in the target table that are not in the source table. This could happen if the target table is used for specialized reporting
- or
- Your security policy does not allow you to drop tables

There are two options for **Purge target table and preserve schema**:

- **Purge using DELETE FROM**

Choose the Delete feature when all deletes are logged, and can be rolled back in case something goes wrong.

If there is an error on a Truncate command, the database is left in the state of the error, as the command is permanent. If a Delete command has problems, the database rolls back to the previous state on transaction failure.

- **Purge using TRUNCATE TABLE** (for target servers where truncate is available)

The CONNX Data Synchronization tool may perform a Truncate more quickly than a Delete. Choose Truncate if:

You do not want to delete indexes in the target table that are not in the source table. This could happen if the target table is used for specialized reporting.

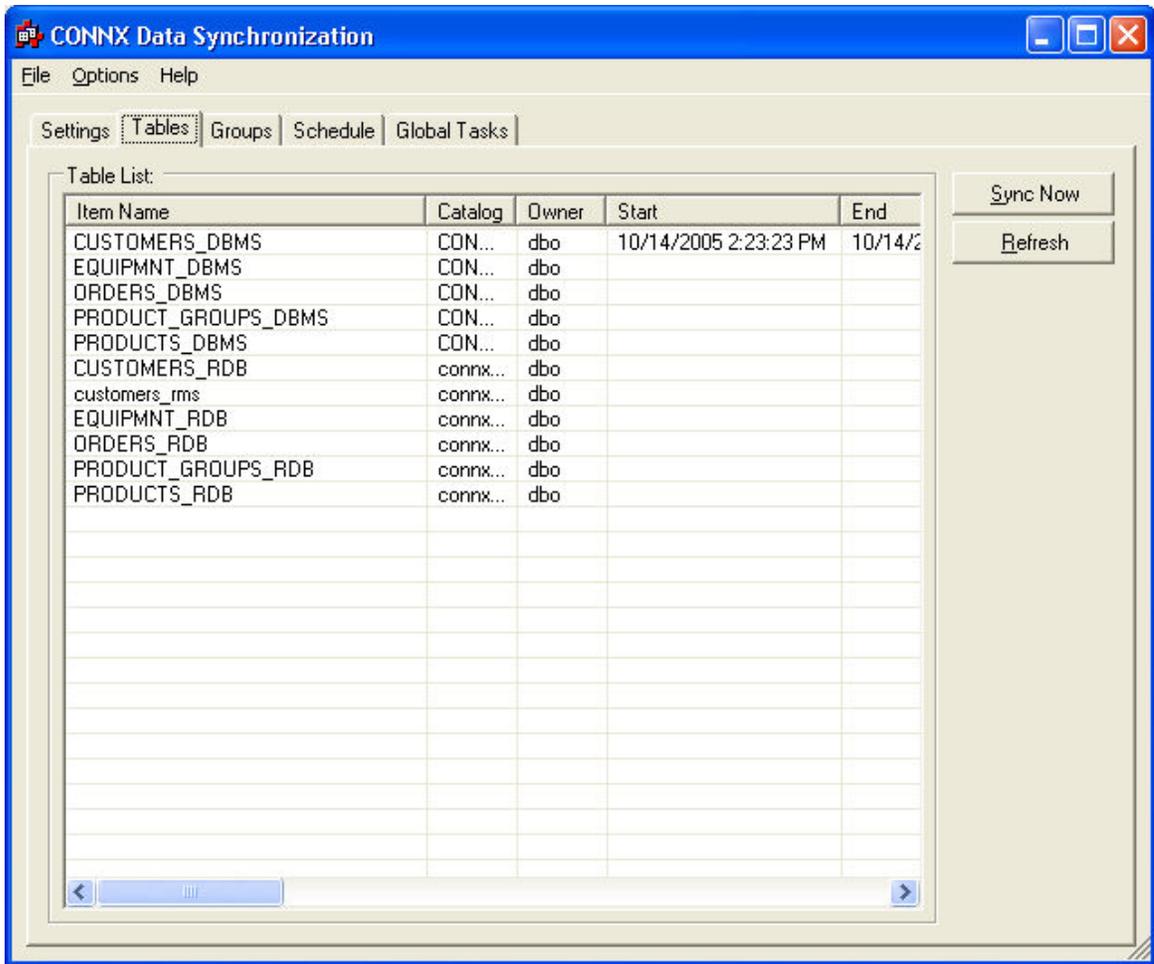
or

Your security policy does not allow you to drop tables.

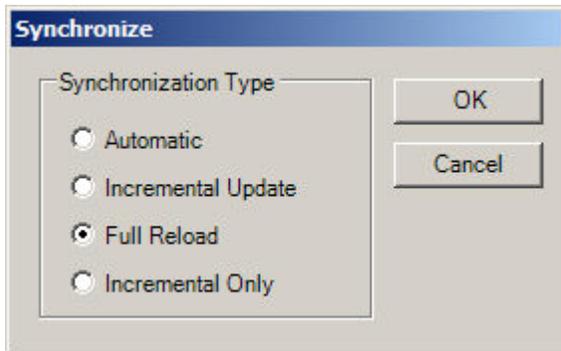
Truncate does not activate triggers, referential integrity constraints, or other database restrictions. Since database integrity is maintained on the source database for the synchronization, choose Truncate if performance is a concern:

If an unsupported database is chosen, the sync succeeds, but a warning message is issued.

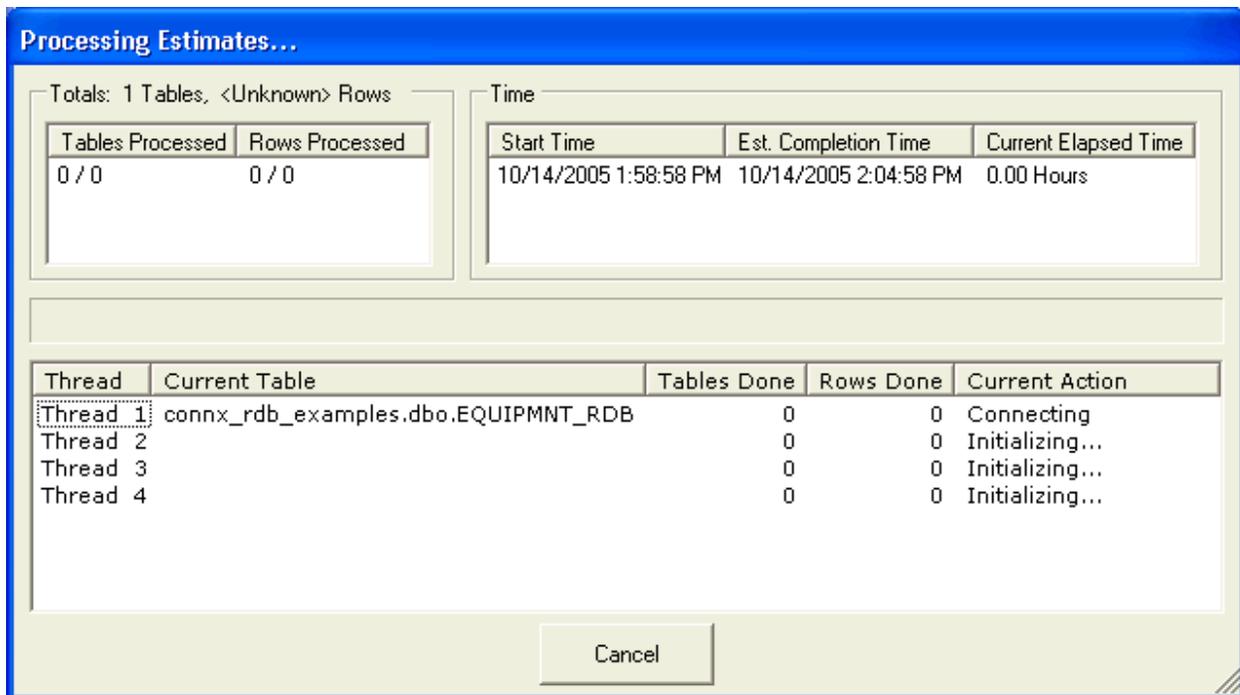
3. On the **Tables** tab, select a source table to synchronize, and then click the **Sync Now** button.



4. The **Synchronize** dialog box appears. Select the **Full Reload** radio button, and then click the **OK** button.



5. The Processing Estimates dialog box appears.



There is a short pause as the CONNX Data Synchronization tool creates a target location.

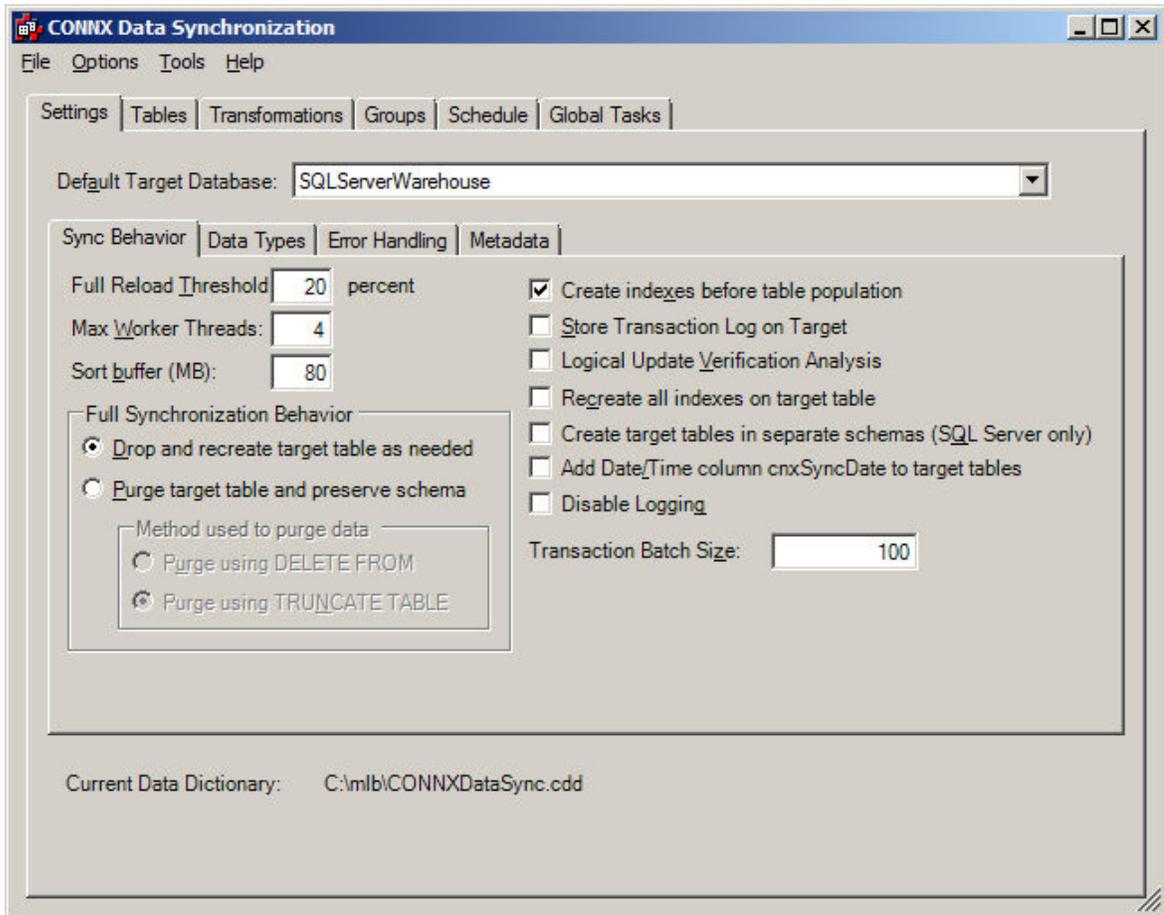
If the target database does not support truncation, a warning appears for every table selected for deletion even though the synchronization continues and succeeds. Click the **OK** button.



To create index before table population

You can reorder the synchronization process so that indexes are created first.

1. In the **Settings** tab, click the **Sync Behavior** sub-tab.



2. Click **Create indexes before table population**.

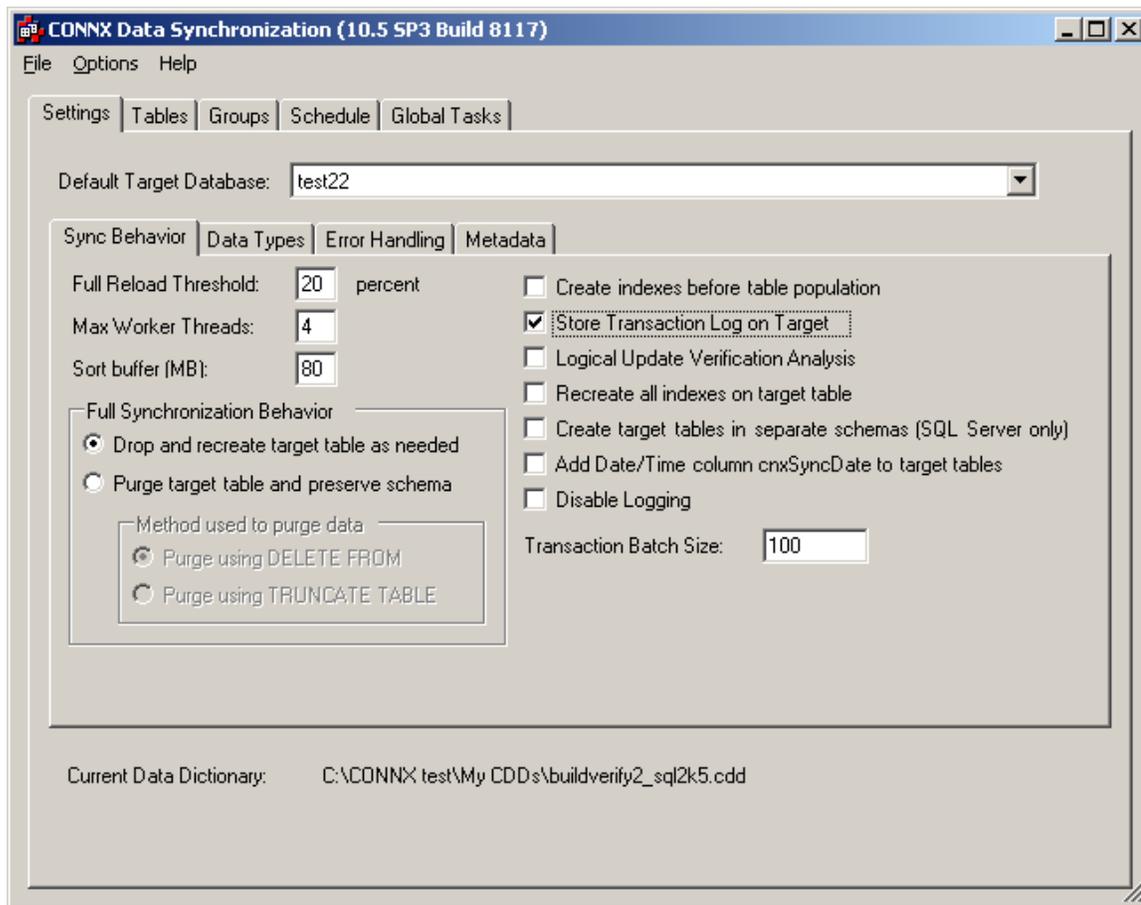
This can have either a negative or positive effect on performance and space requirements of a full synchronization, depending upon the database system.

If the target is an Oracle database, it will be faster to create the indexes after the load of the data. Inserting rows with an existing index updates the indexes on every insert, whereas creation of an index at the end of the synchronization is somewhat more efficient.

To store transaction logs on target databases

This setting stores copies of the transaction log on the target server. A second copy of the transaction log is stored on the CONNXStore server by default.

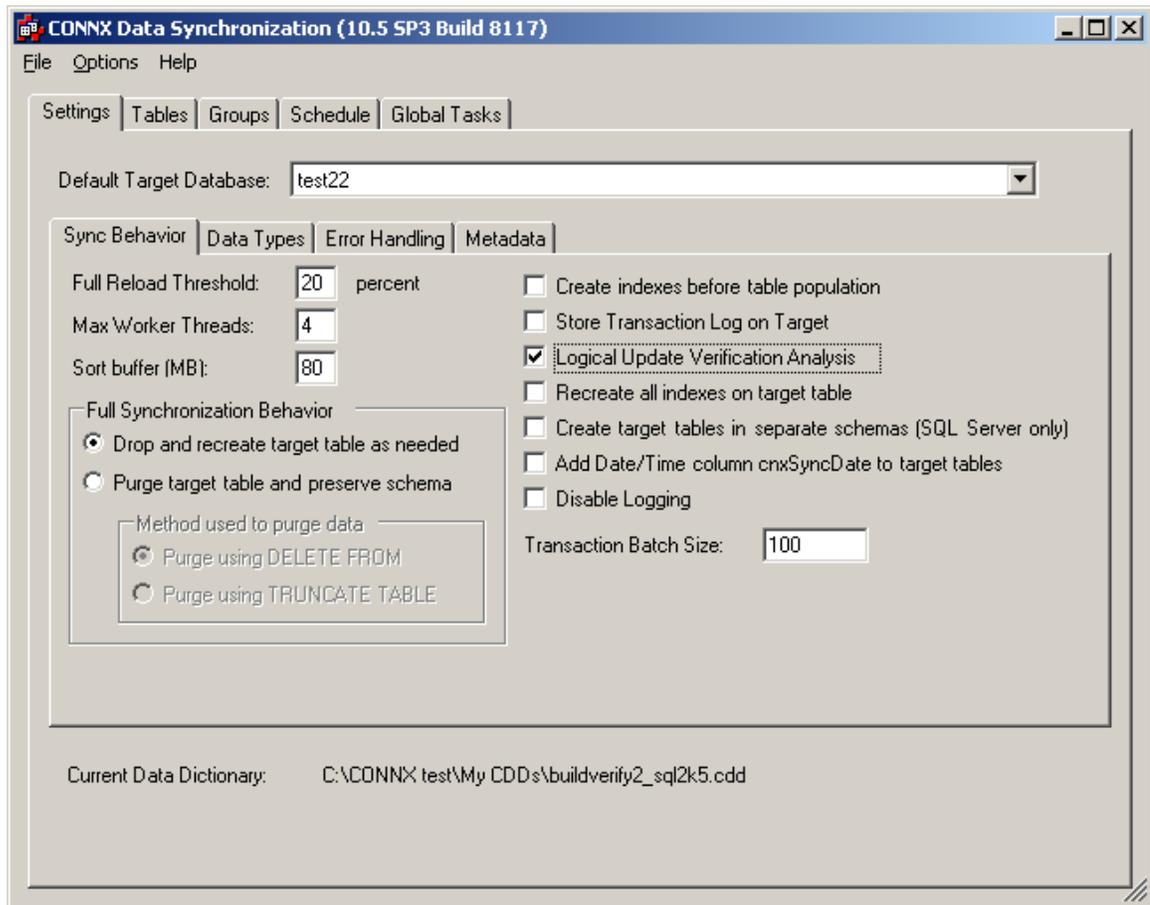
1. In the **Settings** tab, click the **Sync Behavior** sub-tab, Select the **Store Transaction Log on Target** check box.



2. A copy of the transaction log appears on the target server.

To perform verification analysis on updates

1. In the **Settings** tab, click the **Sync Behavior** sub-tab.



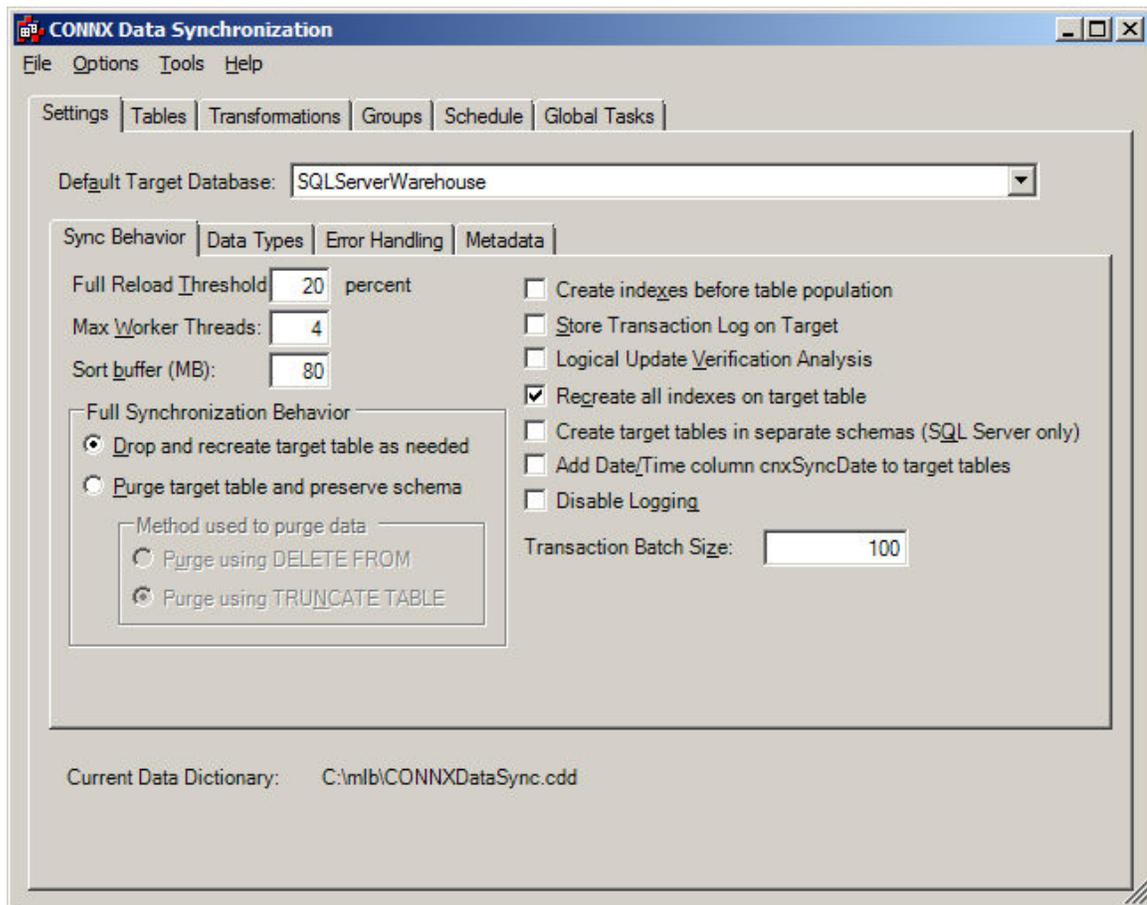
2. Selecting the **Logical Update Verification Analysis** check box enables the CONNX Data Synchronization tool to perform logical comparisons on every field in every record.

Note: This check box should remain cleared unless used for its distinct purpose as it can hinder performance.

To recreate indexes on target table

By default, only the unique index used to perform incremental updates is created on the target table after synchronization.

1. In the **Settings** tab, click the **Sync Behavior** sub-tab.



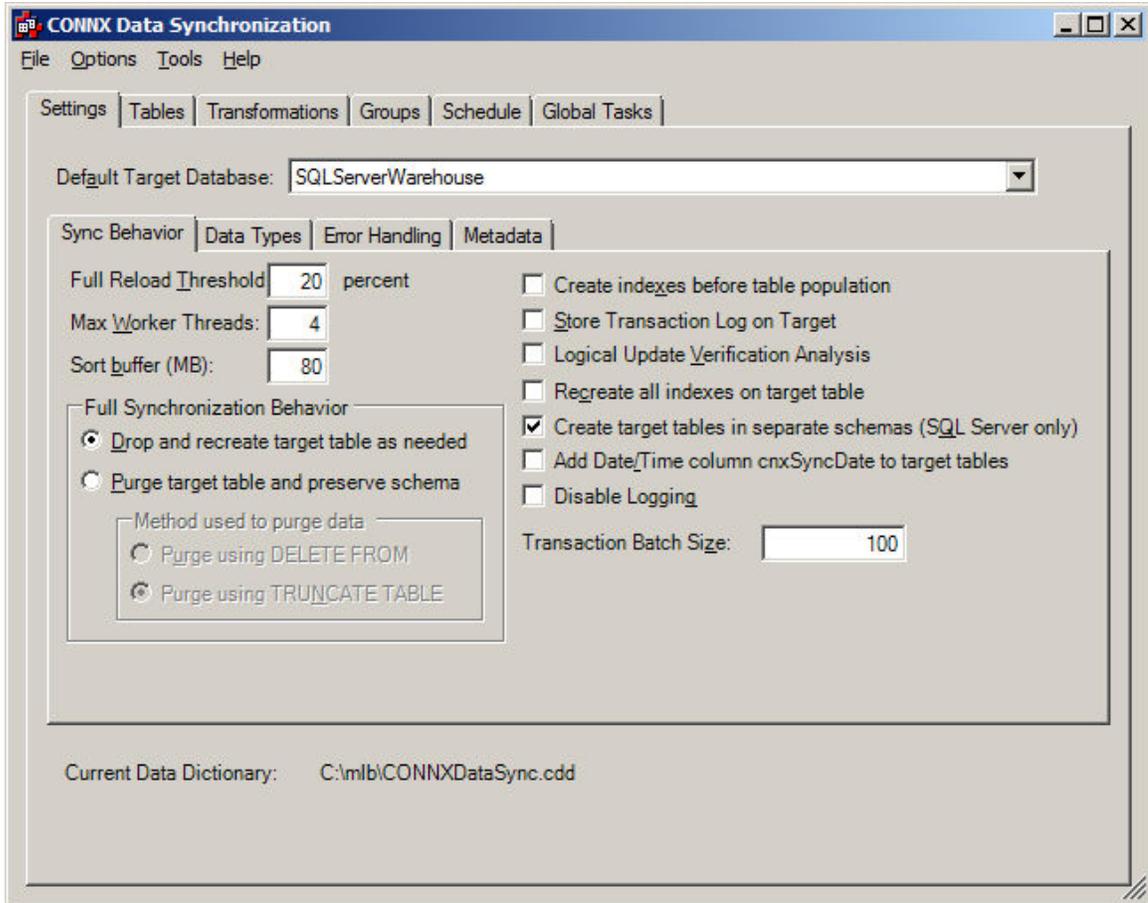
2. Check **Recreate all indexes on target table** to recreate all indexes from the source table on the target table.

To create target tables in separate schemas (SQL Server only)

You can only create target tables in separate schemas if you are using SQL Server as a target database. CONNX Data Synchronization creates a SQL Server user for the purpose of creating tables with identical table names from different source catalogs.

If you want to synchronize both the test and production version of a table into a single SQL Server database, CONNX fully qualifies the target name (product.table or test.table) so that there is no name collision.

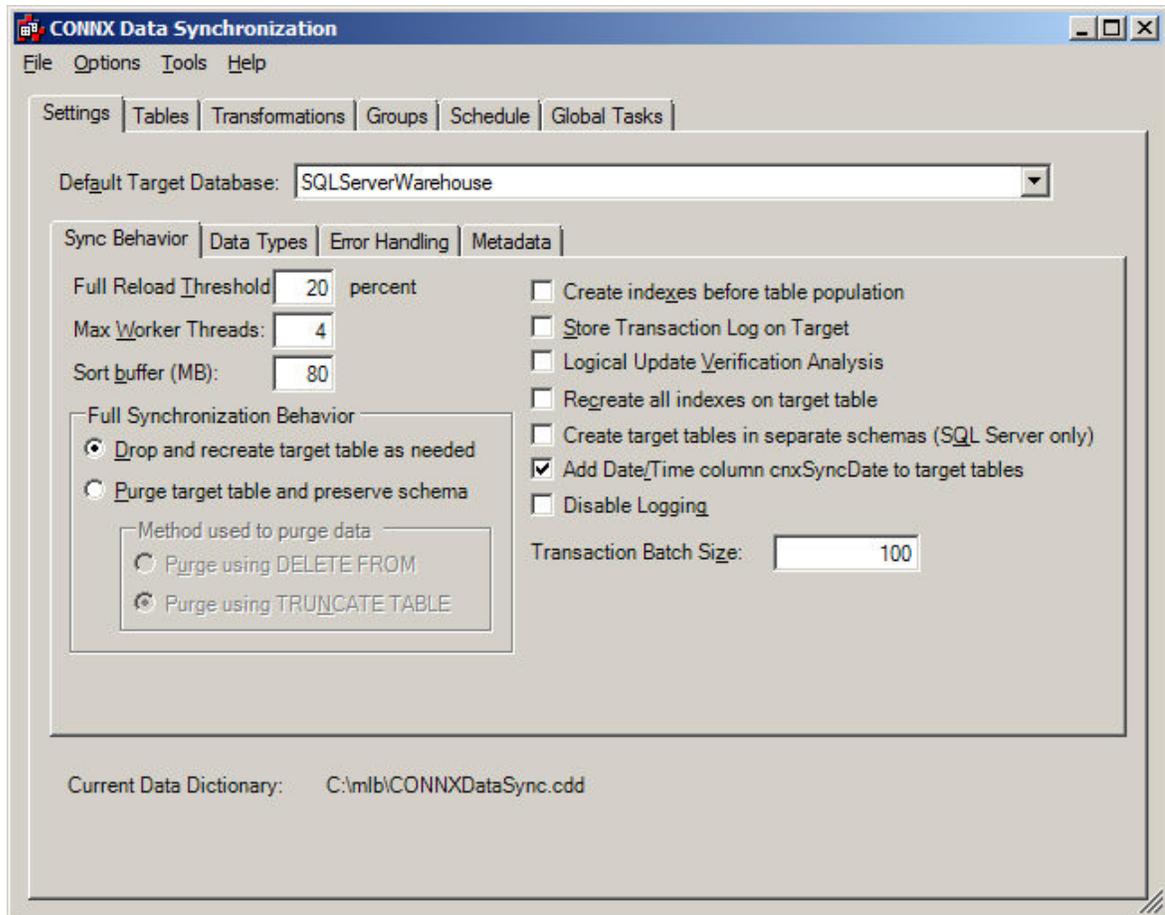
1. In the **Settings** tab, click the **Sync Behavior** sub-tab.



2. Click **Create target tables in separate schemas (SQL Server only)**. This will create target tables under different SQL Server owners.

To add a date/time column to target tables

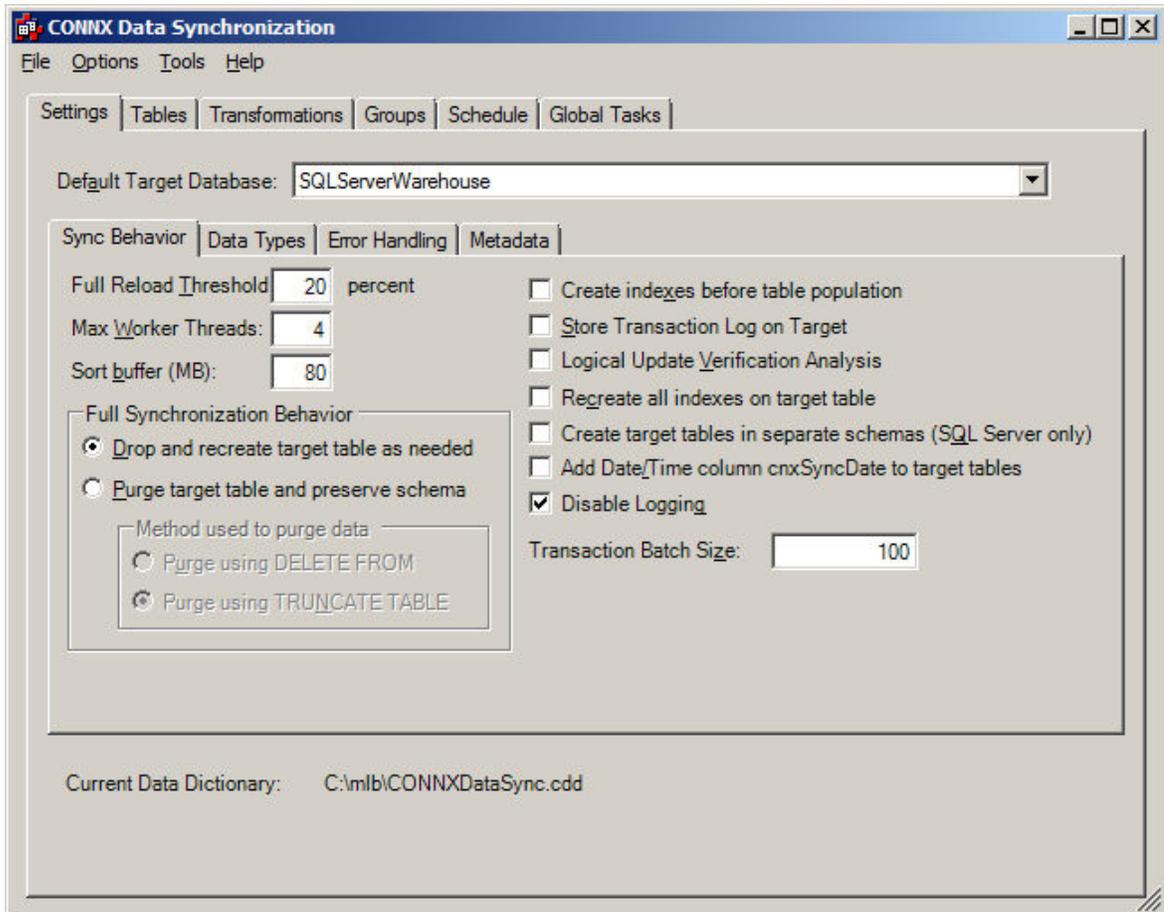
1. In the **Settings** tab, click the **Sync Behavior** sub-tab.



2. Click **Add Date/Time column cnxSyncDate to target tables**. This will append a date and time column to the end of each row in the synchronized target tables. The date and time of the last synchronization operation performed against the target row appears in the new column.

To disable logging

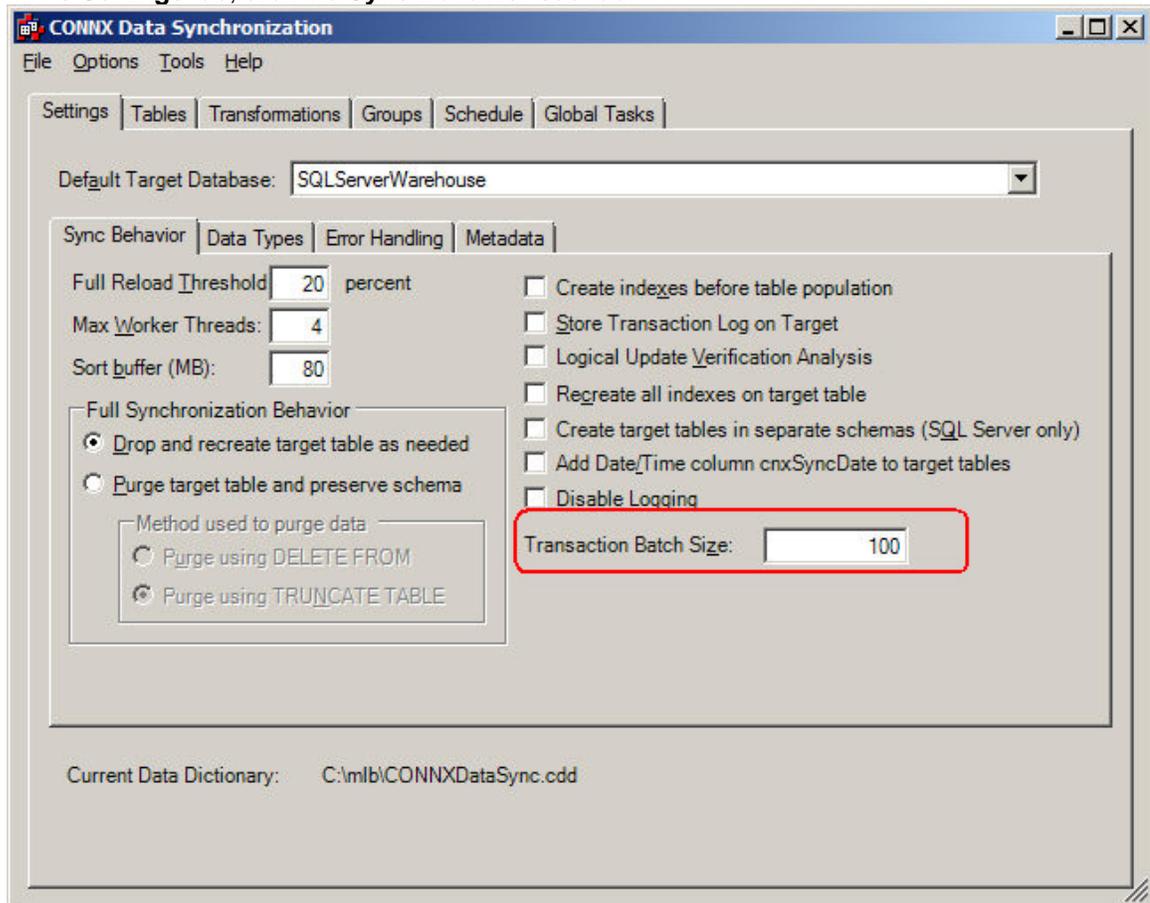
1. In the **Settings** tab, click the **Sync Behavior** sub-tab.



2. Click **Disable Logging**. This will prevent log information from being written to the datasync.log file. The datasync.log file is stored in the datasync folder.

To set transaction batch size

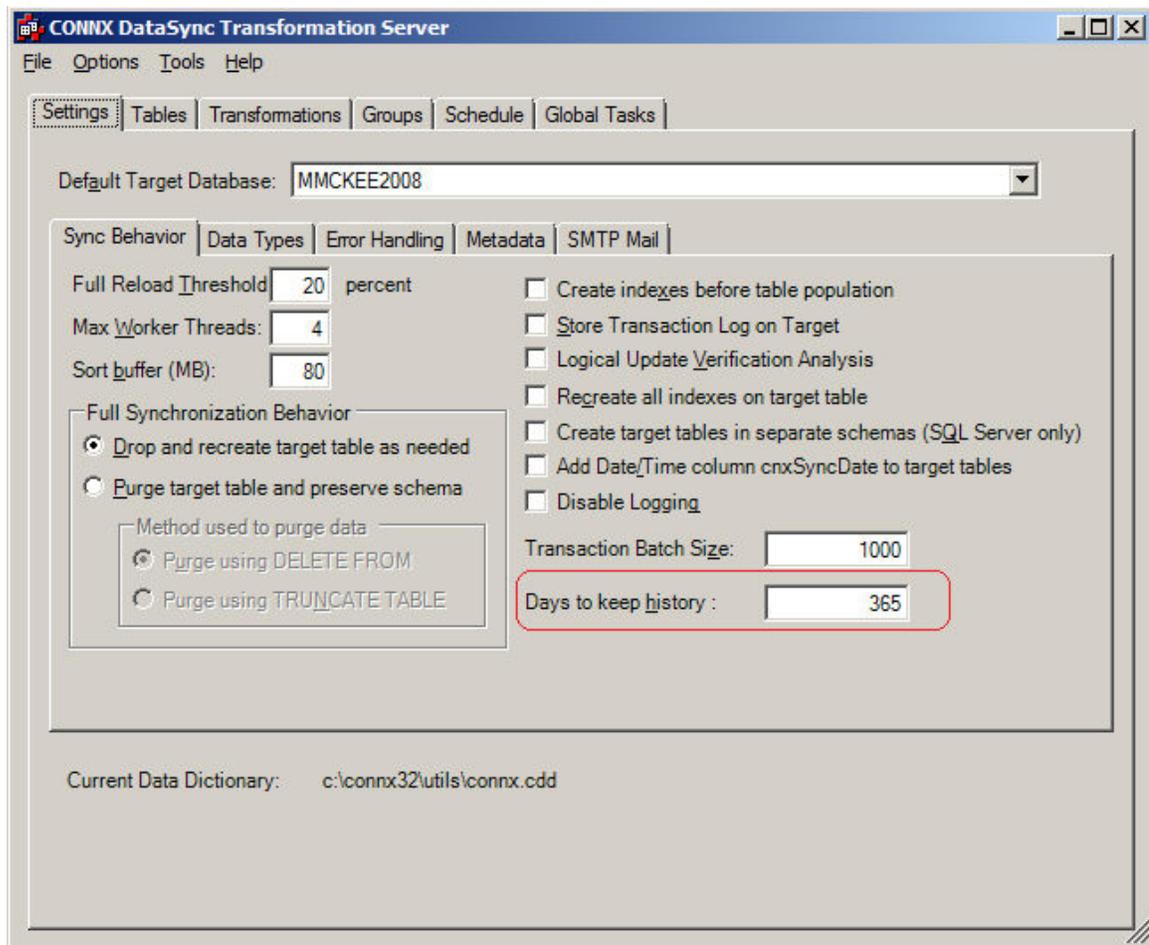
1. In the **Settings** tab, click the **Sync Behavior** sub-tab.



2. Transaction Batch Size refers to how many records are inserted into the target database in a single transaction. Some database systems have problems if too many records are involved in a transaction, and Datasync offers the ability to set this field manually. 100 is the default.

To set days to keep history

1. In the **Settings** tab, click the **Sync Behavior** sub-tab.



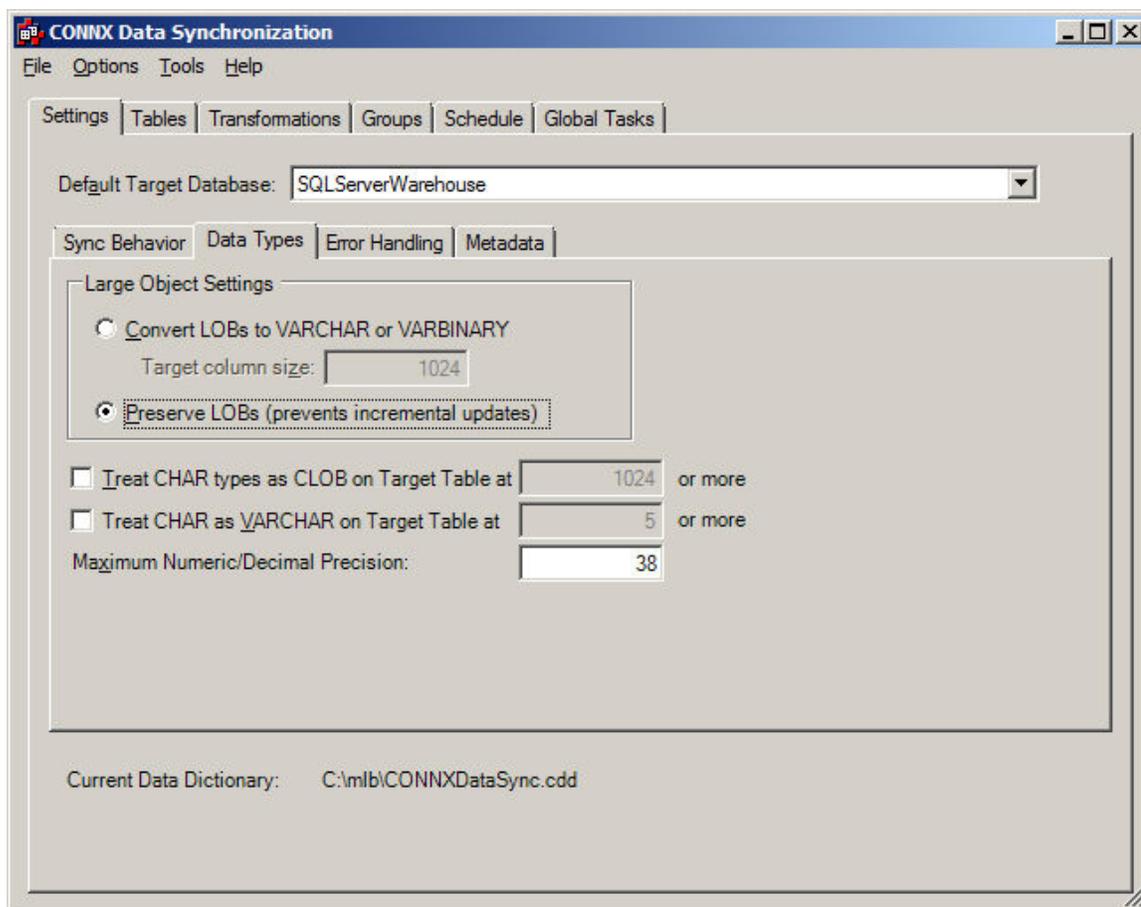
2. You may edit the Days to keep history, or use the default 365. This field means that Synchronization records that are older than the above circled value will be purged from the database. This helps keeps the database size smaller.

Data Types sub-tab

To synchronize blobs

CONNX Data Synchronization offers users with data that is stored as blobs (binary large objects) the option to synchronize that data to both a target server and a main server. Users may attempt this at their own risk, as the unwieldy size of the data may slow down the synchronization process.

1. On the **Settings** tab, click the **Data Types** sub-tab.

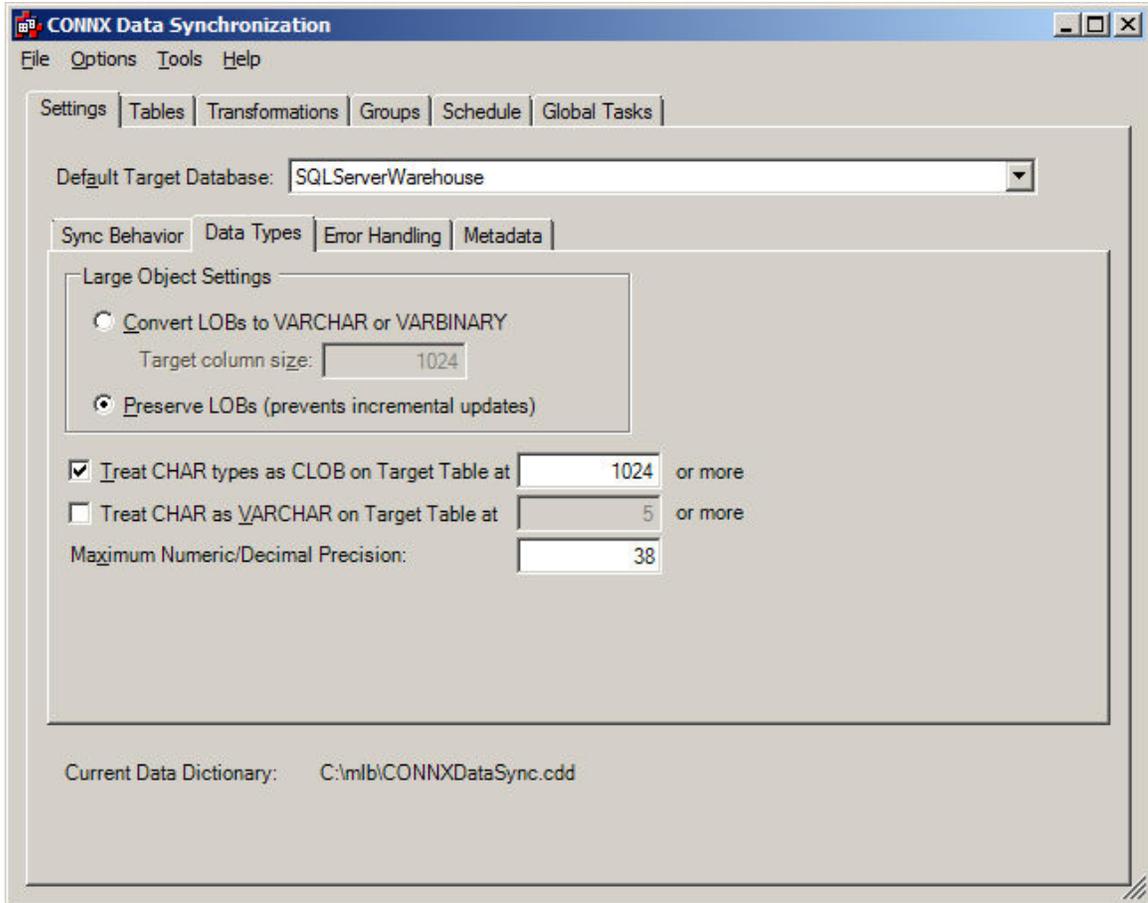


2. To synchronize source and target server blobs, click **Preserve LOBs**.
3. If you click **Convert LOBs to VARCHAR or VARBINARY** (default setting), CONNX will change blobs to either VARCHAR or VARBINARY, depending on the data. Enter the target column size (default is 1024).

To treat CHAR data types as CLOB

Use the **Treat CHAR as CLOB on Target Table at** options if your target data base does not support very large character fields (like SQL Server).

1. On the **Settings** tab, click the **Data Types** sub-tab.



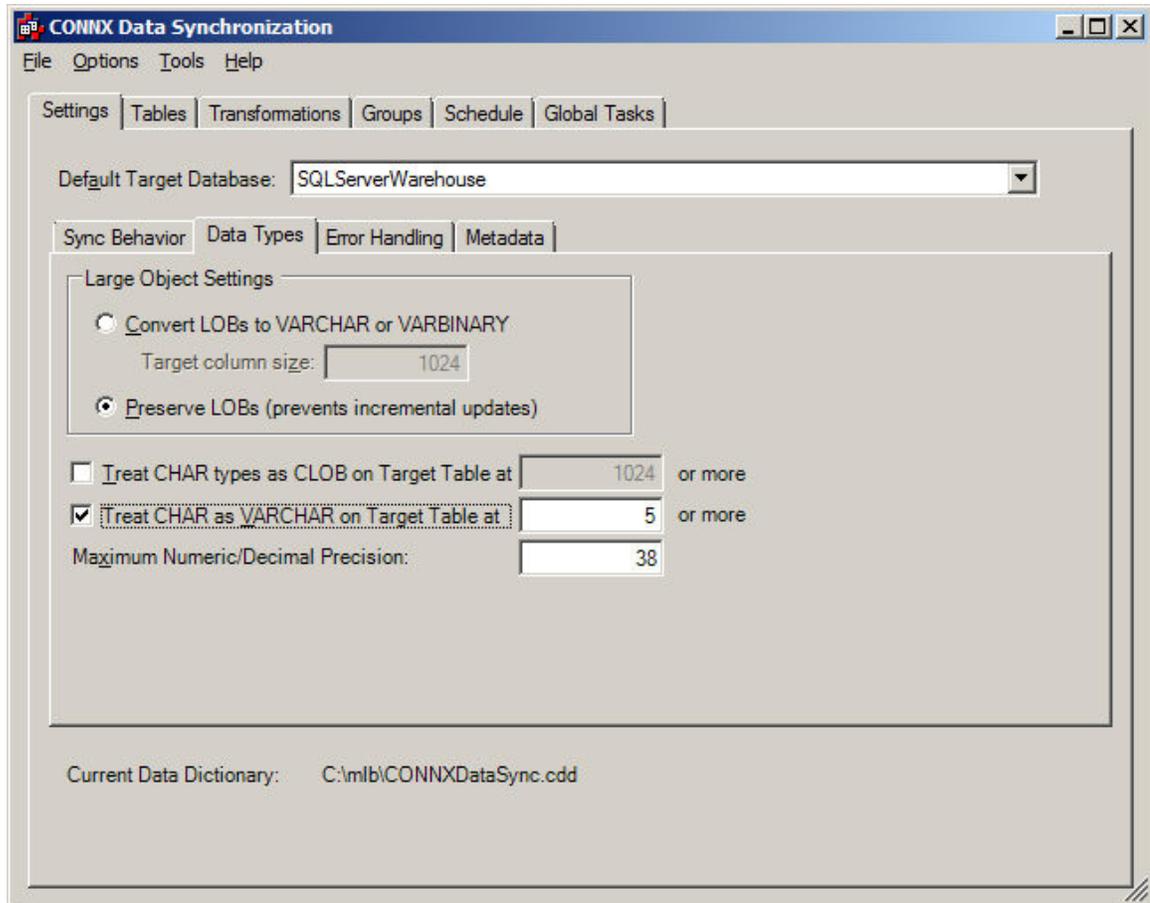
2. If your target data base does not support very large character fields, click **Treat CHAR as CLOB on Target Table at** and enter the field size.

The default value is 1024 but there is no minimum or maximum value.

To treat CHAR data types as VARCHAR

Use the **Treat CHAR as VARCHAR on Target Table at** options to save space in your database table.

1. On the **Settings** tab, click the **Data Types** sub-tab.



2. If character fields are Long and not fully populated, click **Treat CHAR as VARCHAR on Target Table at** and enter the field size.

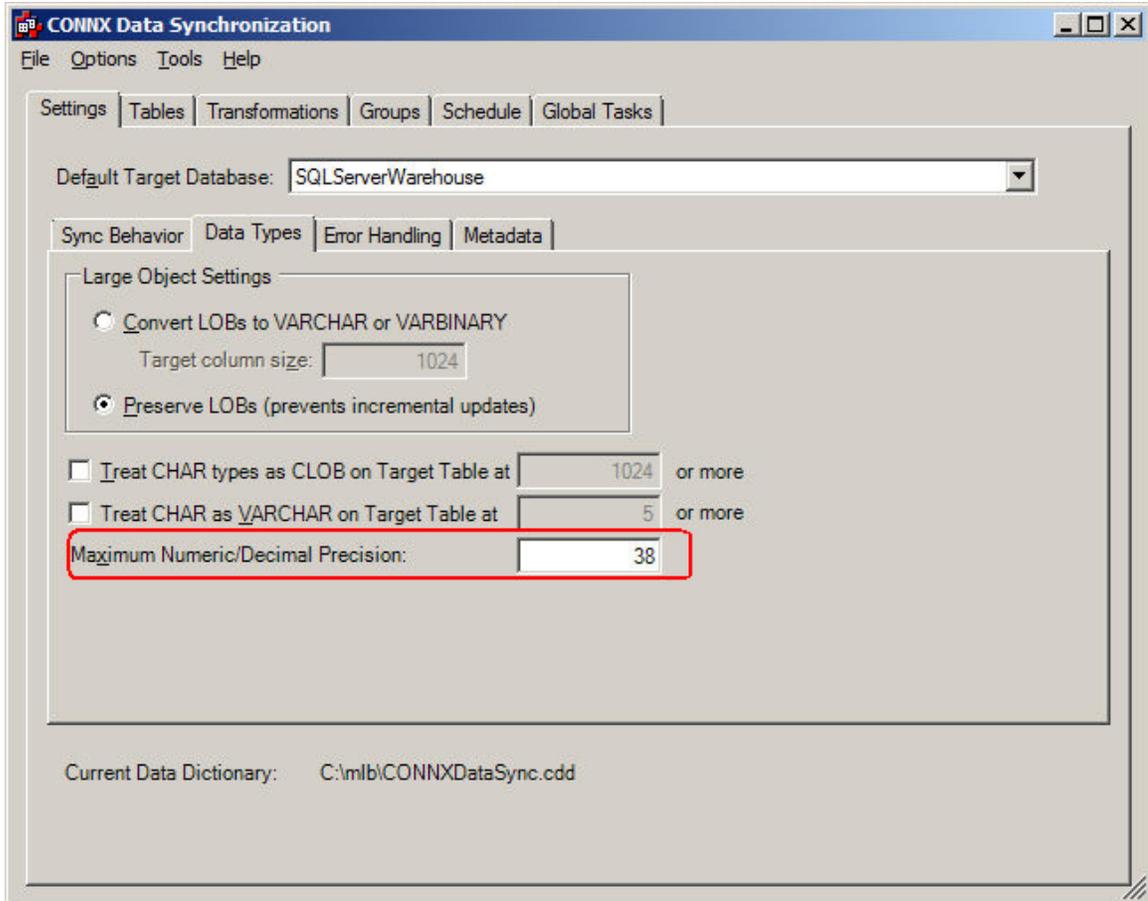
Note: The field should contain at least 5 characters as there is minimal space savings for short CHAR fields.

Example: If the source table has a CHAR(100) field that averages 10 characters in length, it will be transformed into a VARCHAR(100) field on the target table after the synchronization. This could save an average of 90 characters per row.

To select the maximum number/decimal precision for target columns

Specify the maximum number/decimal precision for target columns when your source data base has a higher default numeric precision than your target data base.

1. On the **Settings** tab, click the **Data Types** sub-tab.



2. Enter the maximum numeric or decimal precision value in **Maximum Numeric/Decimal Precision**.

This option is useful when the source is a database like PostgreSQL, which has a high default numeric precision (over 100), and the target is SQL Server which has a maximum numeric precision of 38.

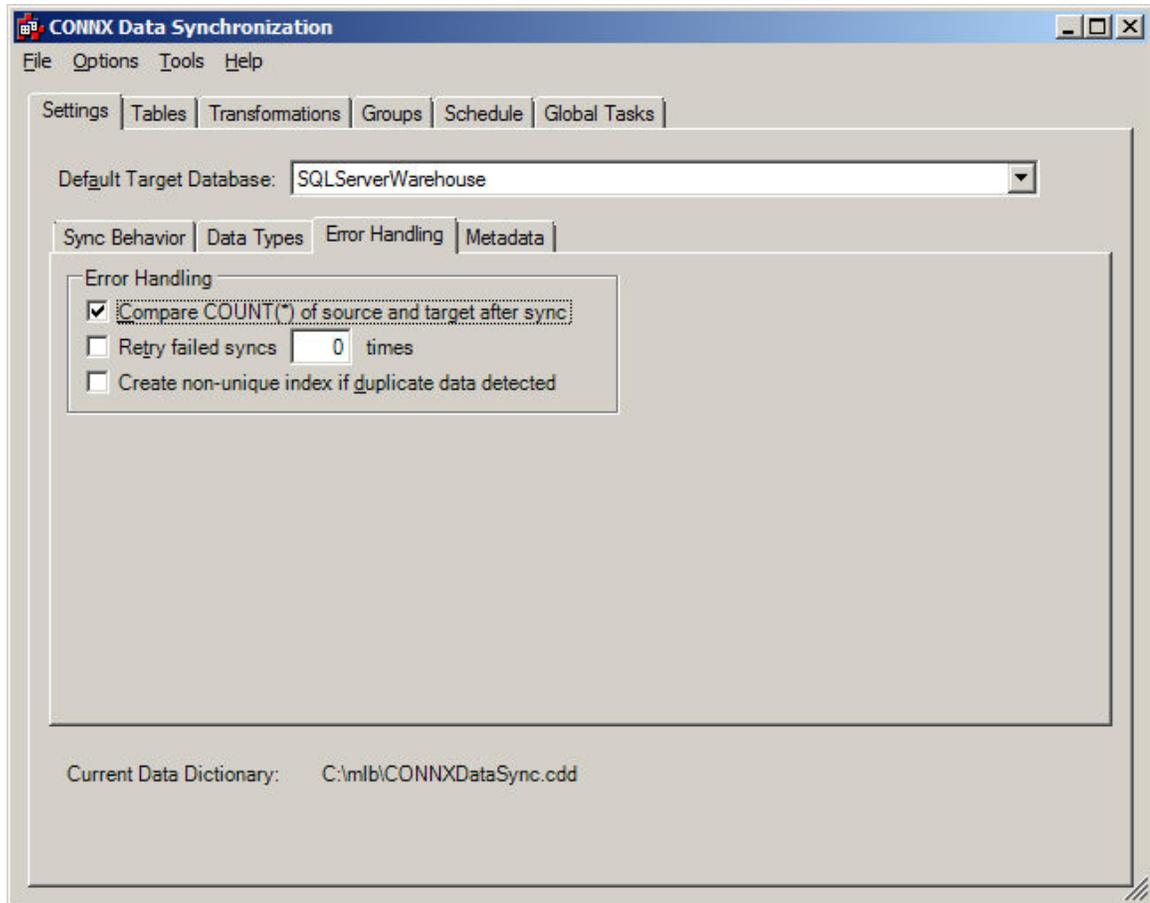
Error Handling sub-tab

To compare COUNT(*) of source and target

When this option is selected, CONNX performs a count(*) from the source table, and compares it to a count(*) from the target table. If the counts don't match, then the sync is considered a failure.

This option is most useful when used in conjunction with the **Retry failed syncs** option.

1. On the **Settings** tab, click the **Error Handling** sub-tab.



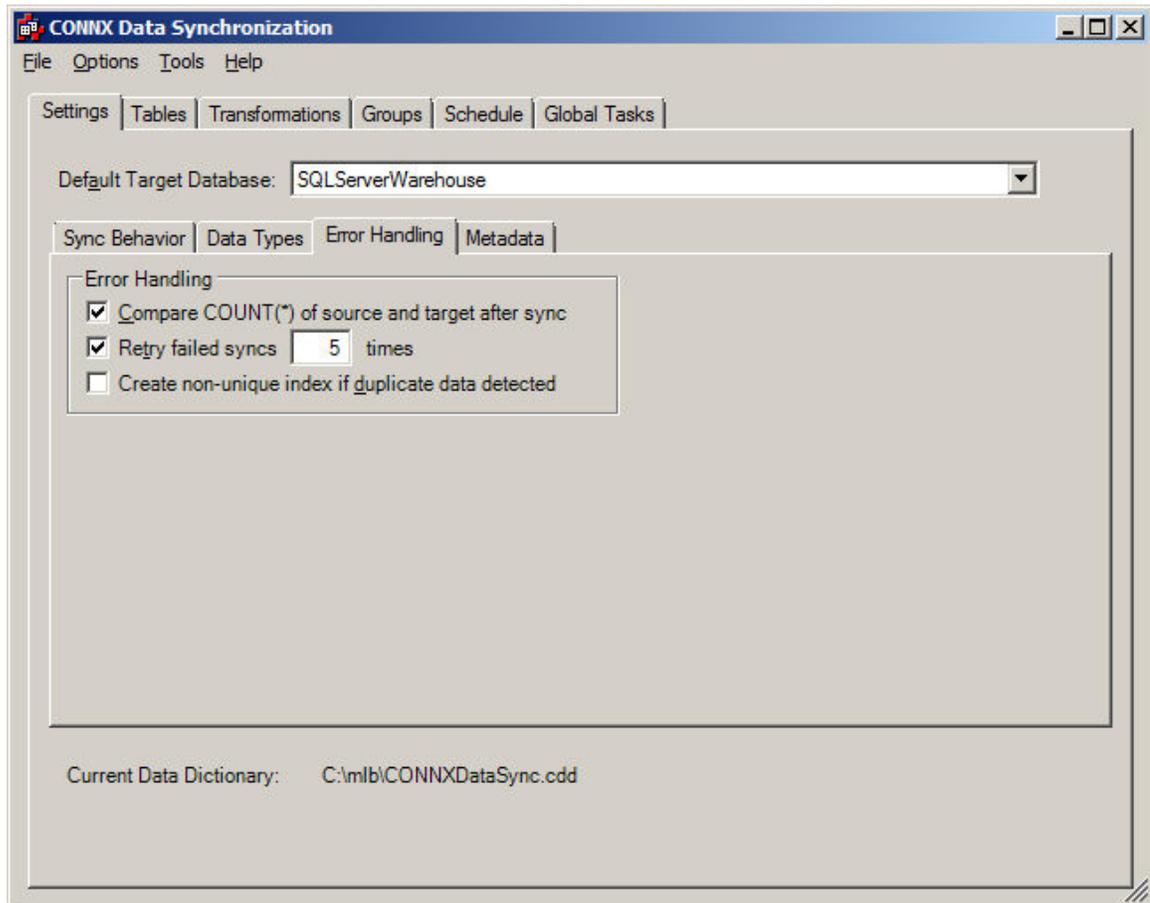
2. To compare source and target table counts, click **Compare COUNT(*) of source and target after sync**.

Note: This option can only be selected if the sync is performed during a "batch window" and the source database remains static during that time.

To retry failed synchronizations

When this option is selected, CONNX Data Synchronization automatically switches to a full synchronization and retries the synchronization if it fails.

1. On the **Settings** tab, click the **Error Handling** sub-tab.

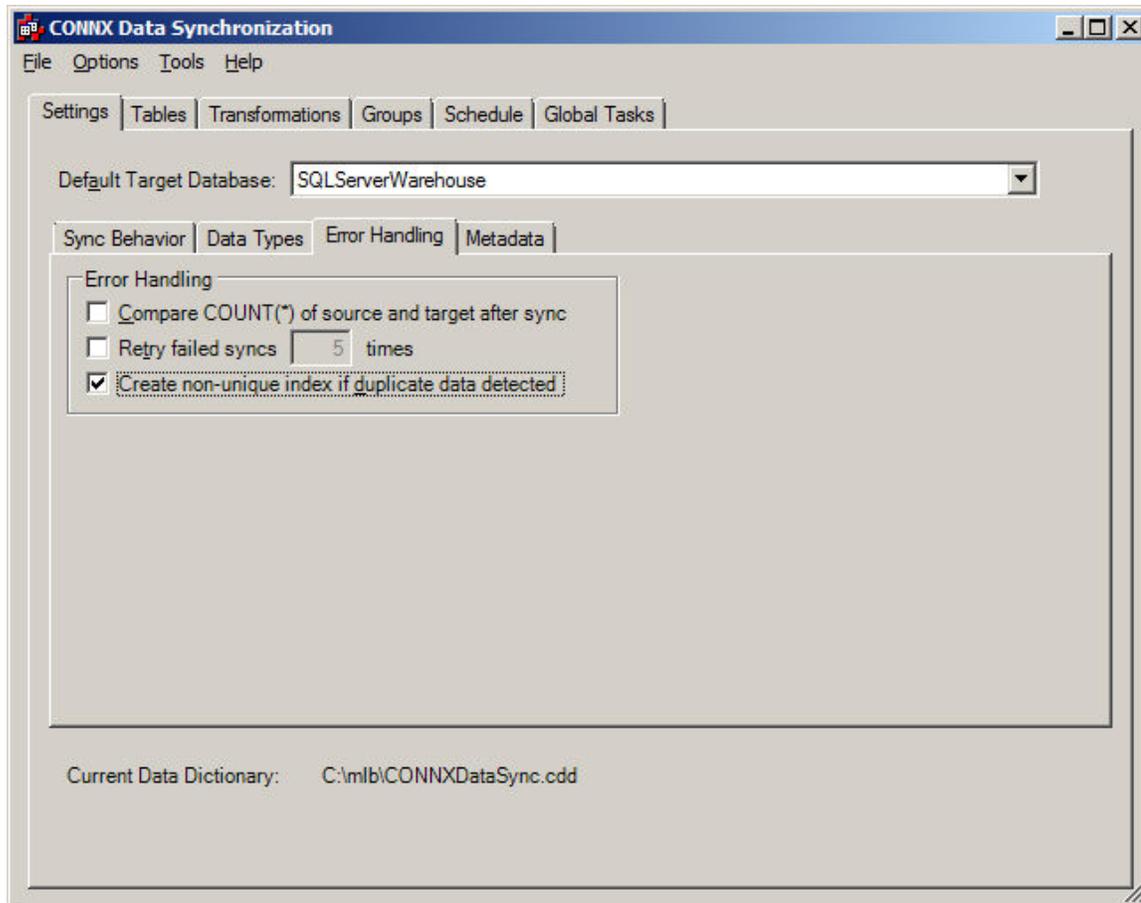


2. To retry failed synchronizations, click **Retry failed syncs** and enter the number of times to retry the synchronization.

To create a non-unique index if duplicate data is detected

Do not use this option unless absolutely necessary. If duplicate data is detected during the full sync, the data source table should be corrected.

1. On the **Settings** tab, click the **Error Handling** sub-tab.



2. Click **Create non-unique index if duplicate data detected**. CONNX creates a non-unique primary key instead of a unique primary key during a full update if duplicate data is detected.

Metadata sub-tab

To target metadata case

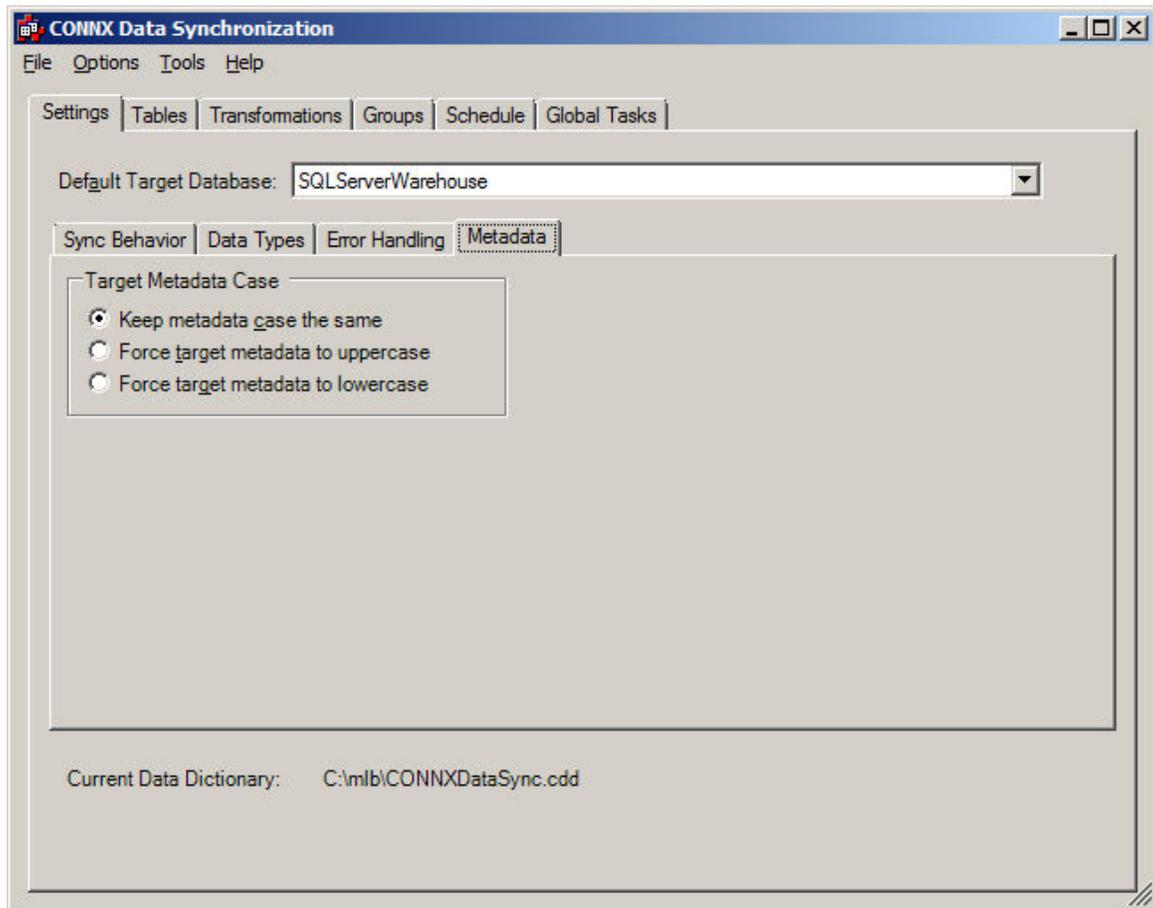
CONNX by default uses quoted identifiers when creating target tables. If column or table names are mixed-case, then double quotes must be used to refer to them in SQL statements.

However, it may make more sense for the target metadata to be in all uppercase, or all lowercase.

- If Oracle metadata is in all uppercase, then it is treated as a case-insensitive identifier and double quotes are not needed.
- If PostgreSQL metadata is in all lowercase, then it is treated as case insensitive.

CONNX Data Synchronization allows you to choose the case for the target metadata.

1. In the **Settings** tab, click the **Metadata** sub-tab.

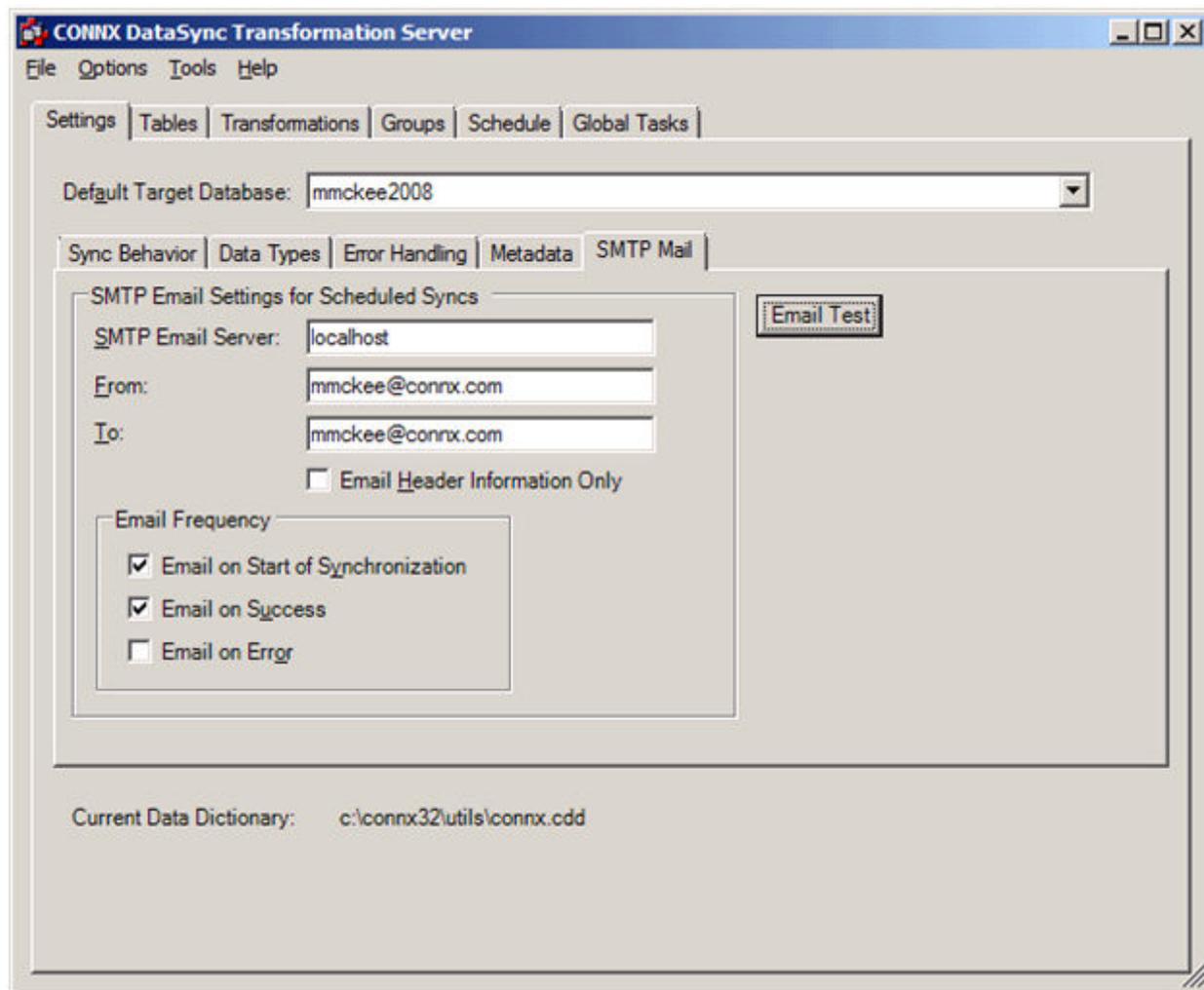


2. Select **Keep metadata case the same** to ensure that the metadata remains either all uppercase or all lowercase.
3. Select **Force target metadata to uppercase** to ensure that all metadata remains uppercase.
4. Select **Force target metadata to lowercase** to ensure that all metadata remains lowercase.

SMTP Mail sub-tab

SMTP Mail from Scheduled Syncs

CONNX Datasync has the ability to have email notification sent to a group or to individual(s), after a scheduled sync has been performed. This enables administrators to have instant access to information relating to sync's. In addition, it can also send email if problems result from a synchronization. To use this ability, the user should navigate to the SMTP Mail sub-tab. It appears as:



The user should fill in the following:

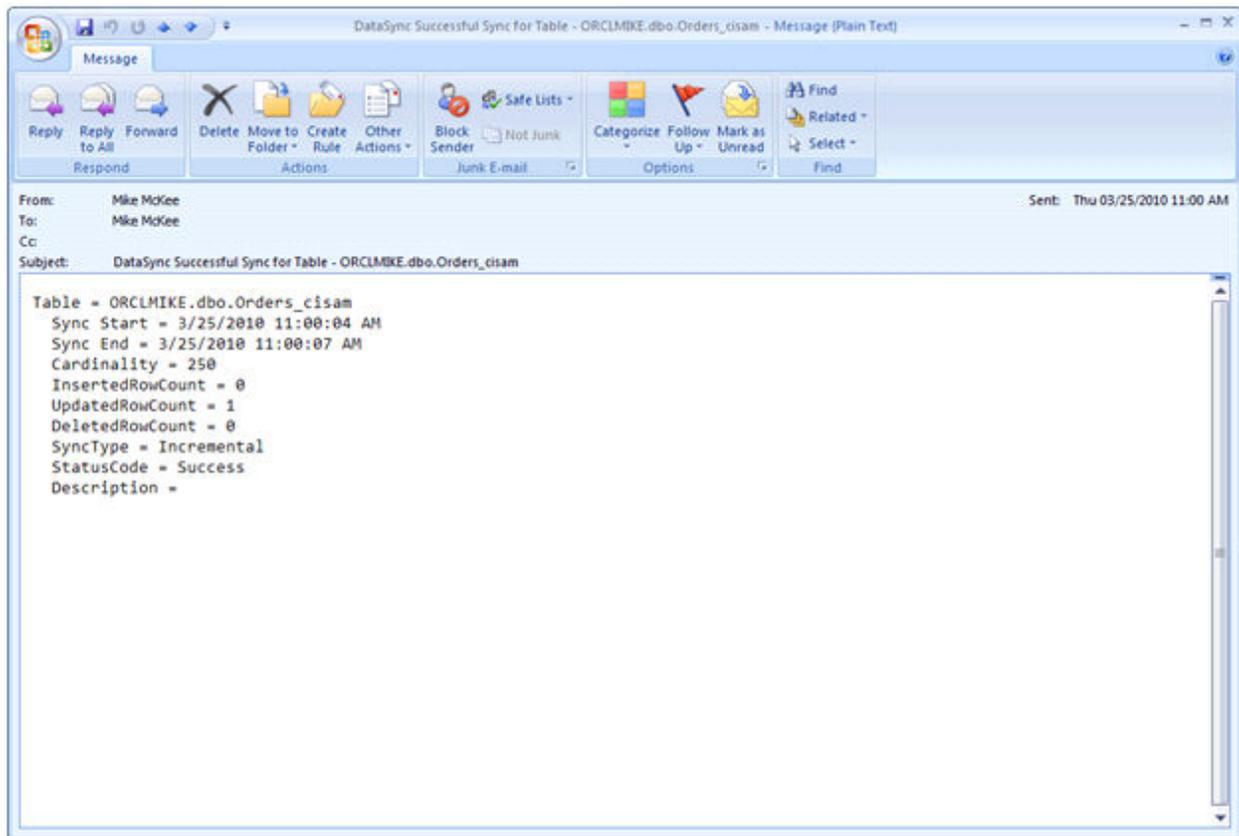
- SMTP Email Server
On Windows machines, this can be invoked by starting the SMTP Mail Service.
- From
This reflects the sender's email address. For this example, the sender and receiver are the same.

CONNX Data Synchronization 12.5

- To
This indicates the recipient of the email.
- Frequency
Emails can be sent on start, success, and error conditions.

An Email Test button is provided to verify that the SMTP settings are correct. After pressing this button, an immediate response will be provided indicating that the settings are correct or incorrect.

After a scheduled sync has been performed, Email messages should be sent to the designated location, and the content of the email should indicate detailed status relating to the synchronization that was performed.

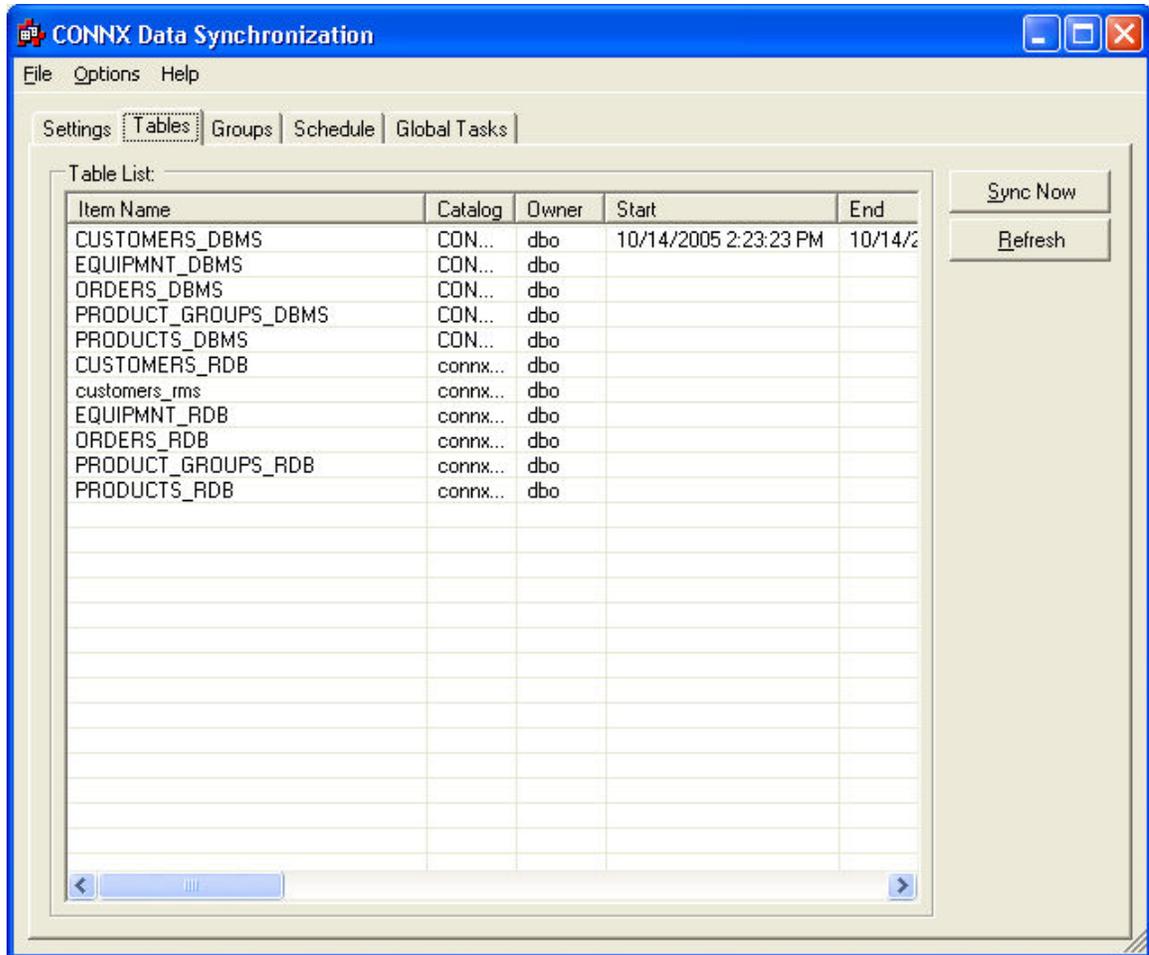


The subject will always start with DataSync. You could set up certain mail rules to have these messages placed to an appropriate folder.

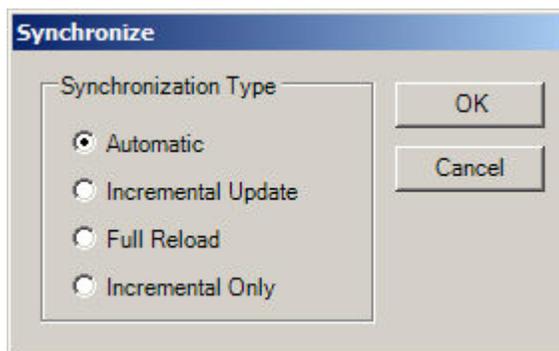
Tables Tab

Synchronizing Tables to Target Destination

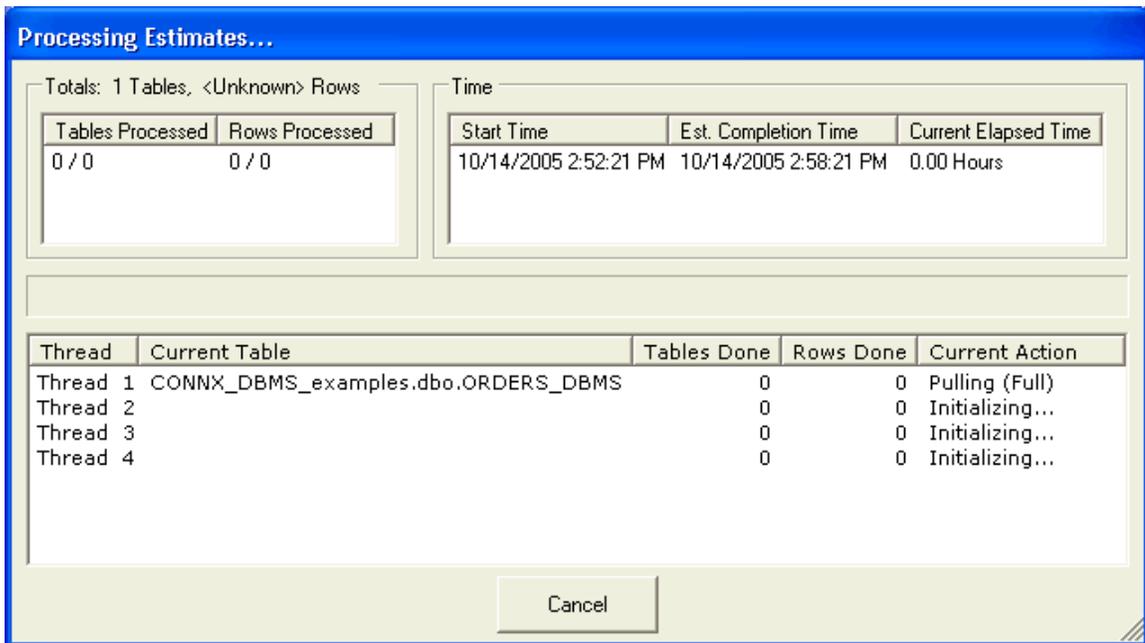
1. Select the **Tables** tab in the CONNX Data Synchronization tool window.



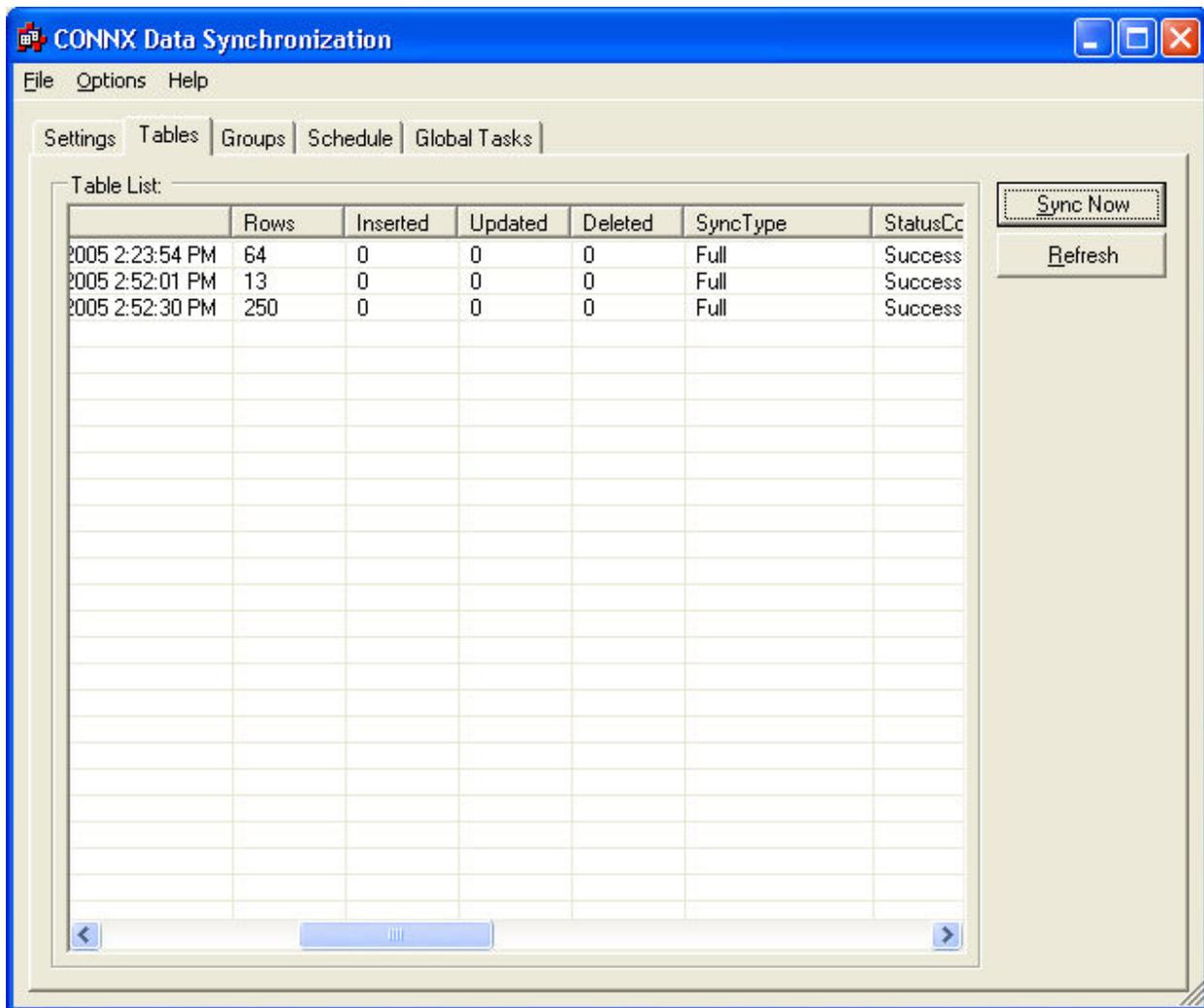
2. Select a table, and then click the **Sync Now** button to start the synchronization process.
3. The **Synchronize** dialog box appears.



4. Select the synchronization type, **Automatic**, **Incremental Update**, **Full Reload**, or **Incremental Only**, depending on what is required. The recommended synchronization for the select database is displayed. Note that an **Incremental Update** is only available if a full reload has already been performed. Automatic will perform an Incremental Update unless the amount of changed records has exceeded the threshold, or a unique key is not available for the table. If Incremental Only has been selected, it will always incrementally update the table (the threshold will be ignored).
5. Click the **OK** button to start the synchronization process.
6. The **Processing Estimates** screen appears.



- The **top left pane** in the Synchronizing screen displays the number of tables and rows that have been processed so far. The totals from a previous run (if any) appears as a text string below the pane.
 - The **upper right pane** displays the start time, estimated completion time, and time spent so far in synchronization. The estimated completion time and elapsed time change as synchronization progresses. The estimates improve over time as more and more synchronizations of the selected type are performed. The time estimate for the initial synchronization of any given type is a rough approximation.
 - The **bottom pane** displays the variety of thread tasks. The first column contains the number of threads being run, the second column contains the name of the table. The third column contains the number of tables completed. The fourth column contains the number of rows. The final column explains the active function of the currently running thread. When complete, the CONNX Data Synchronization window appears.
7. Once the synchronization is complete, the **Tables** tab reappears. Click the **Refresh** button to see the most recent changes.
 8. Scroll to the right to view the results of the synchronization process. Several columns appear with data describing the results of the process.



Column Name	Description
Start	The date and time the last synchronization process was performed.
End	The date and time the last synchronization process ended.
Count	The number of rows in the target database at the end of the last synchronization process.
Inserted	The number of rows inserted in the target database before the last synchronization process was run.
Updated	The number of rows updated before the last synchronization process was run.
Deleted	The number of rows deleted before the last synchronization process was run.

CONNX Data Synchronization 12.5

SyncType	The type of synchronization process run, either automatic update, incremental, or full reload.
StatusCode	States whether the synchronization succeeded or failed.
StatusDescription	Comments included for last synchronization run.

7. Click the **Refresh** button if a scheduled synchronization has occurred and you want to view the changes.

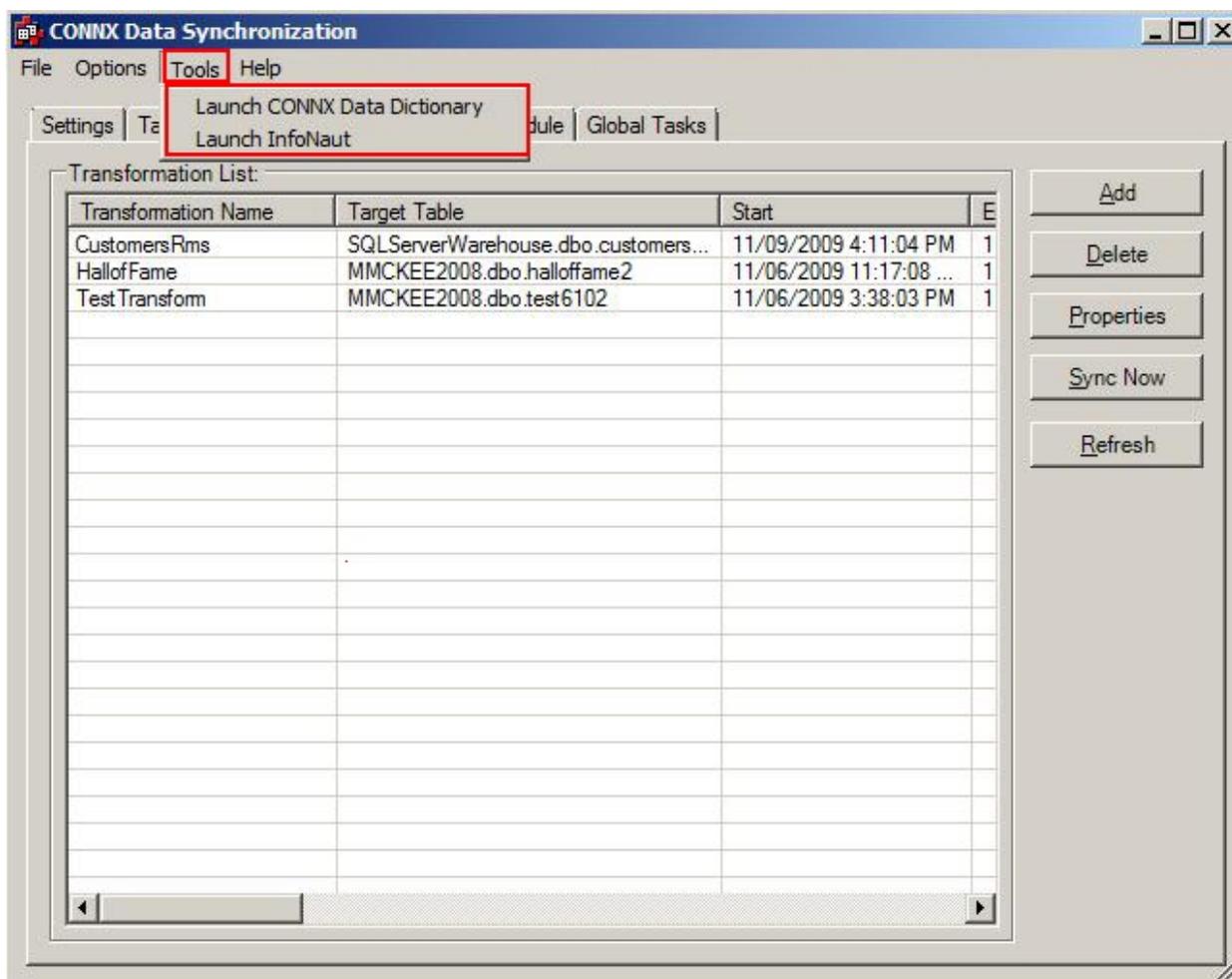
Transformations Tab

Before you start

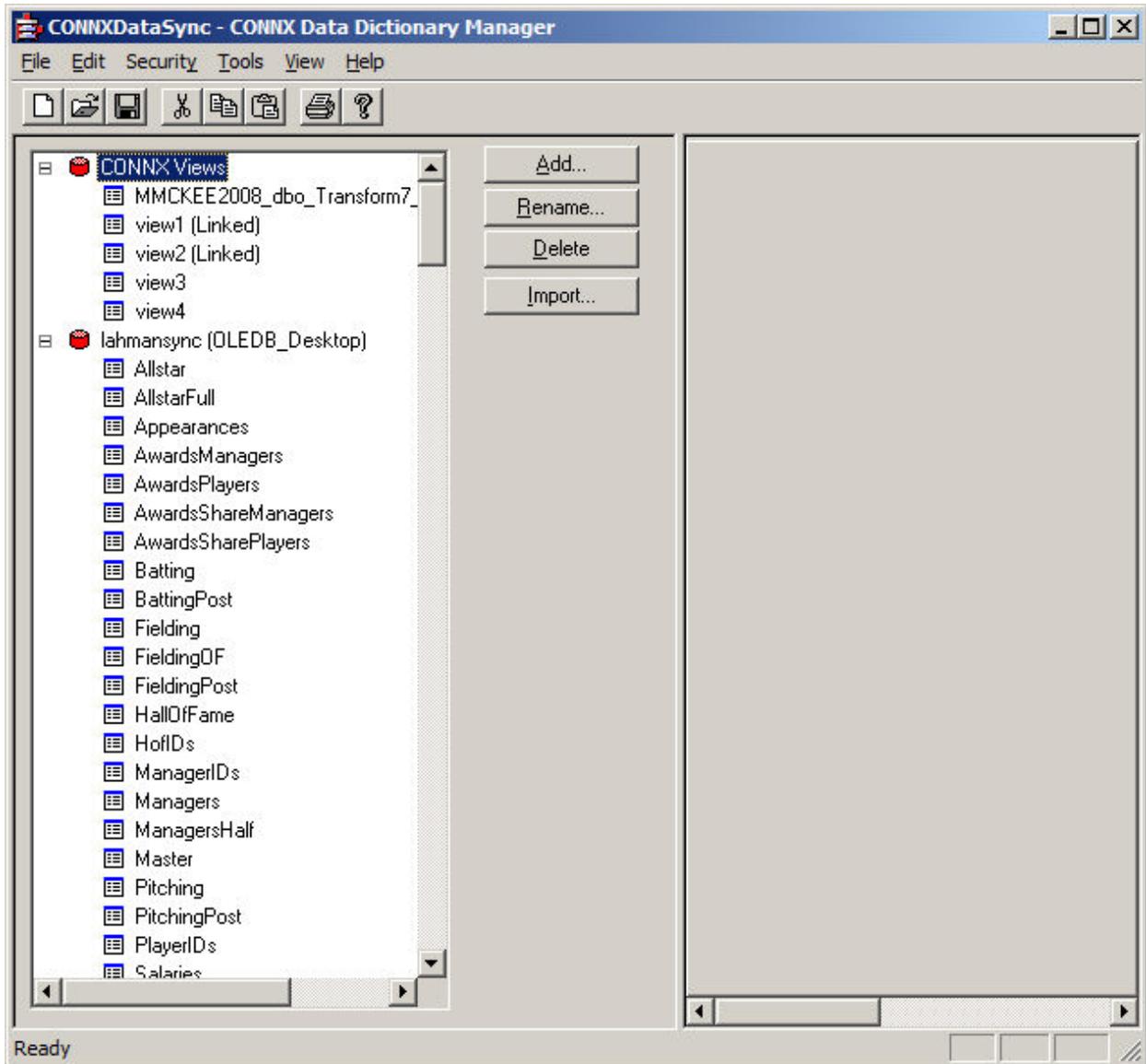
Transformations are at the "heart" of an ETL (Extract, Transform, Load) Project. Datasync now offers the ability to transform the source feeds, and produce a target table on a selected target database. Before a transformation is created, it is often a good idea to take advantage of CONNX Tools to help design the potential transformation. During the DataSync design process, the following CONNX Tools can offer valuable support:

- CONNX Data Dictionary Tool.
- CONNX Infonaut Tool.

Both of these tools are available by selecting Tools



The CONNX Data Dictionary Manager will display all of the tables/databases/views that CONNX has put under a virtual federated database. These tables and indexes can offer insight into selecting a unique index for a transformation that might include multiple tables.



CONNX Data Dictionary

CONNX Data Dictionary enables database, tables, and indexes to be displayed, enabling proper selection of a unique index for multiple table transformations.

Infonaut can be used to preview the transformation to see what the data could look like. Since all transformations are SQL Statements, Infonaut offers the ability to display the results. You simply can cut/paste your query into the Infonaut grid window, and execute the query.

InfoNaut Professional - C:\mlb\CONNDataSync.cdd

File View Connection Tools CONNX Help

Execute Show Tables Build Query Report Refresh CDD QDBC Trace Get Query Schema Max Rows 3

Data Execution Plan Uniform SQL Relational DB SQL

Transaction: Begin Commit Rollback

Recent Connections: DD=C:\mlb\CONNDataSync.cdd;Prompt=2;Persist Security Info=True;APPLICATION=ALL;

Recent Queries: SELECT CONNX_DBMS_examples.dbo.ORDERS_DBMS.*, SQLServerWarehouse.dbo.customers_rms.customerid, SQLServerWarehouse.dbo.customers_rms.customername FROM SQL

```

SELECT CONNX_DBMS_examples.dbo.ORDERS_DBMS.*,
SQLServerWarehouse.dbo.customers_rms.customerid,
SQLServerWarehouse.dbo.customers_rms.customername
FROM SQLServerWarehouse.dbo.customers_rms INNER JOIN
CONNX_DBMS_examples.dbo.ORDERS_DBMS ON
CONNX_DBMS_examples.dbo.ORDERS_DBMS.CUSTOMERID =
SQLServerWarehouse.dbo.customers_rms.customerid

```

Drag a column header here to group by that column.

CNX_ORDERS	CNX_CUSTOMERS	CNX_PRODUCTS	ORDERID	CUSTOMERID	PRODUCTID	ORDERDATE	PRODUCTQUANTITY	customerid1	customername
1 1:25.4	1:28.1	1:85.4	101	ALWAO	77	12/12/1997 12:00..	2	ALWAO	Always Open C
2 1:25.3	1:28.1	1:93.3	101	ALWAO	59	12/12/1997 12:00..	15	ALWAO	Always Open C
3 1:80.6	1:28.1	1:58.2	84	ALWAO	63	08/30/1997 12:00..	20	ALWAO	Always Open C
* <null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>

Record: 1 of 3 Records/Sec: 0 Time: 10.278081

Infonaut Query Results

At this time, you should select the Transformations Tab, and you will be directed to a display that shows all the existing transformations that have been saved. The following section in [adding a transform](#)

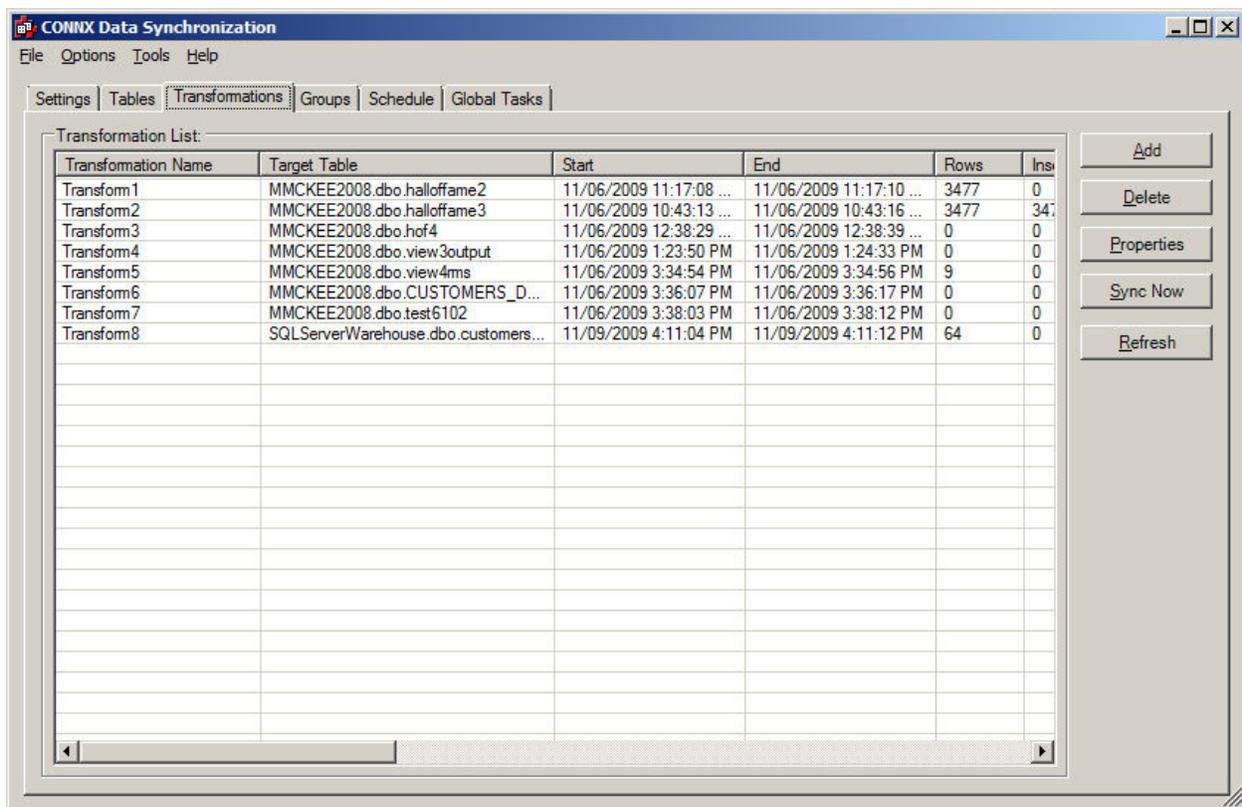
Designing a Transform

Source sub-tab

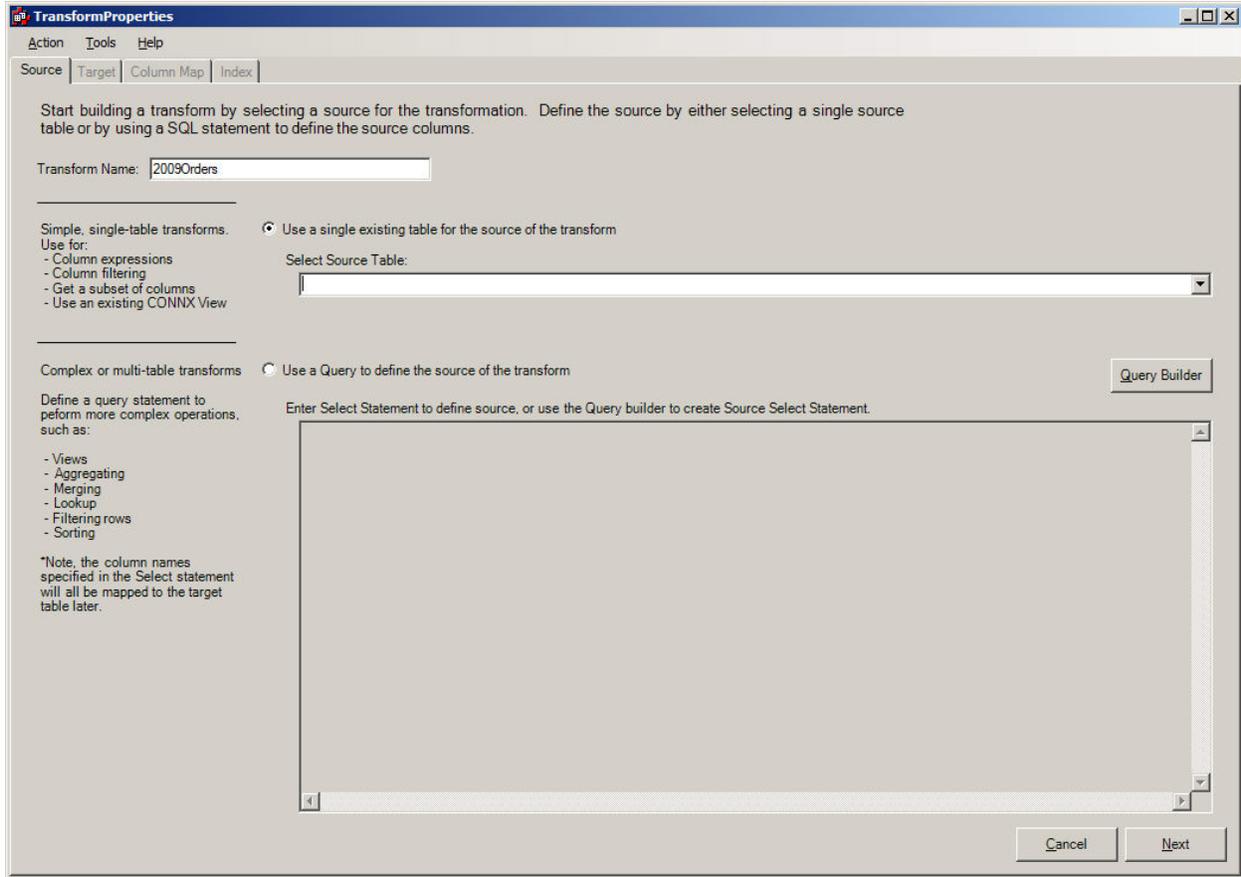
Creating a Transform

DataSync offers the ability to do transformations by selecting the transformation tab on the DataSync window. When Transformations is selected, all of the Transformations that have been currently been saved are listed, along with information relating to the last synchronization. The Transformation Name lists the name of the Transform. By default the Transformation Name will be *Transformxxx* where xxx = a number starting with 1. The Target Table will be a three part name consisting of:

- Database
- User
- Table name



Adding a Transform is accomplished by selecting the Add button. A new window will be created that invokes the Transformation Wizard.



The first part of the multiple tab wizard is the Source Tab. The Source Tab is responsible for the following:

- Transform Name (User is required to enter a descriptive name for the Transform of up to 32 characters).
- Transform Type option (User can select from Single Table Transform, Complex Transform using QueryBuilder, or Free-Form SQL)

Selecting Transform Name

The Transform Name represents the name that will be applied to the Transform. The ETL Designer should describe something the relates to the transform.

Selecting the type of Transform

Depending on the type of transform desired, there are three options (Simple Source Table, Free-From SQL, Complex Transform using QueryBuilder) to choose from. The table below helps identify which type is appropriate based on the transformation desired.

Transform Type	Simple Source Table Transform	Free From SQL	Complex Transform via QueryBuilder
----------------	-------------------------------	---------------	------------------------------------

Selecting Subset of Table or Tables	x	x	x
Translating coded values (e.g. Changing the Source column from a 1 to Male)	x	x	x
Calculated Values (e.g. OrderAmt * 1000)	x	x	x
Filtering		x	x
Sorting		x	x
Joining Multiple Tables (e.g. Lookups)		x	x
Aggregation		x	x
Unions		x	x

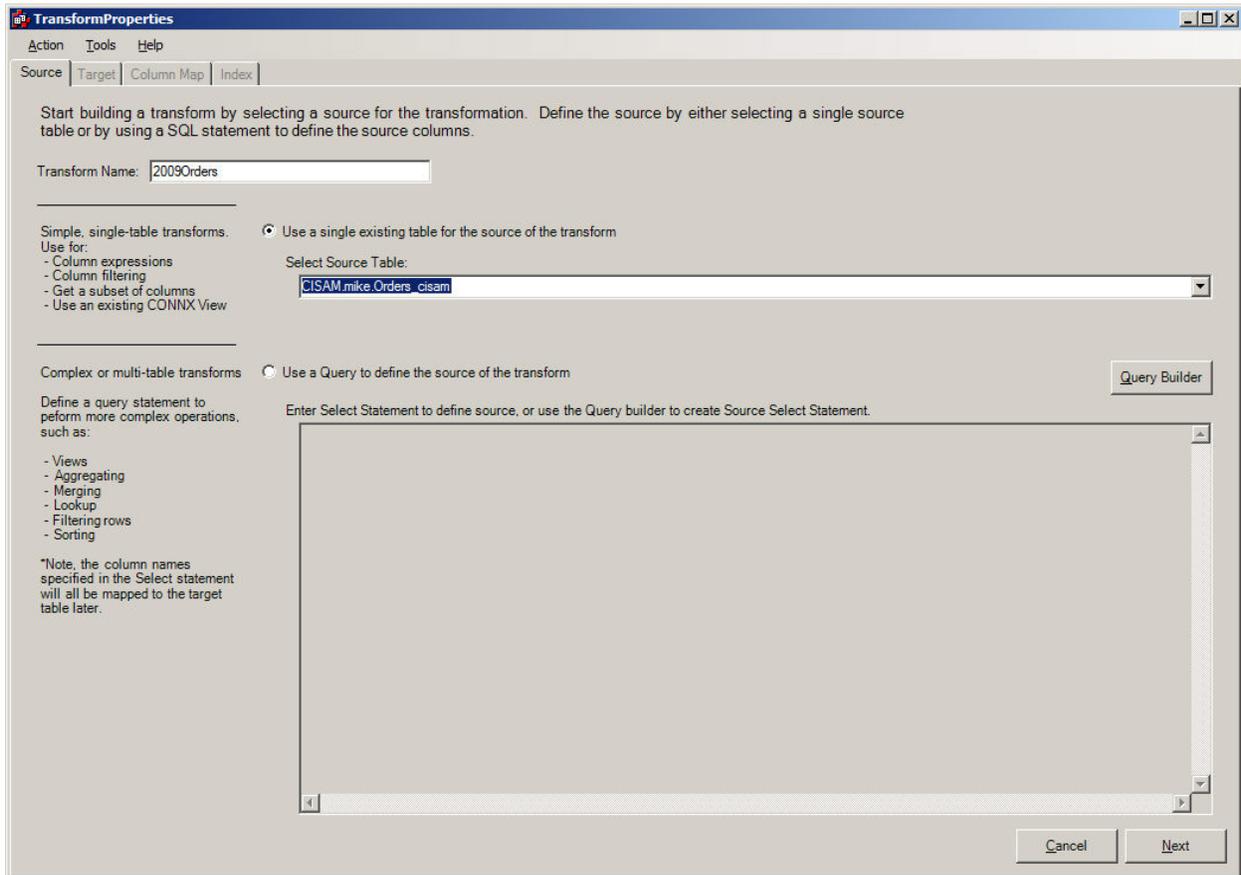
When the Transformation Type has been selected, the user should press the NEXT Button to continue creating the Transformation. The NEXT Button will advance the user to one of the three choices:

- [Free Form SQL Transform](#)
- [Designing a Single Table Transform](#)
- [Designing a complex Transform using QueryBuilder](#)

Designing a Single Table Transform

To choose a Single Table transform, the radio button must be selected that says, "Use single table for existing transform". A drop down list box will be displayed under the **Select Source Table** caption. This list will contain all of the potential tables that included in the CONNX Data Dictionary File. The user must select a single table which will serve as the input for the target destination file.

One valid reason to choose this transform is for transforms that might only include certain fields of an existing table. It can also be used to map derived columns (e.g. orderamt * 100). When choosing this transform, SQL is not required unless a derived field is created, and requires SQL to create a transformation from source column(s).

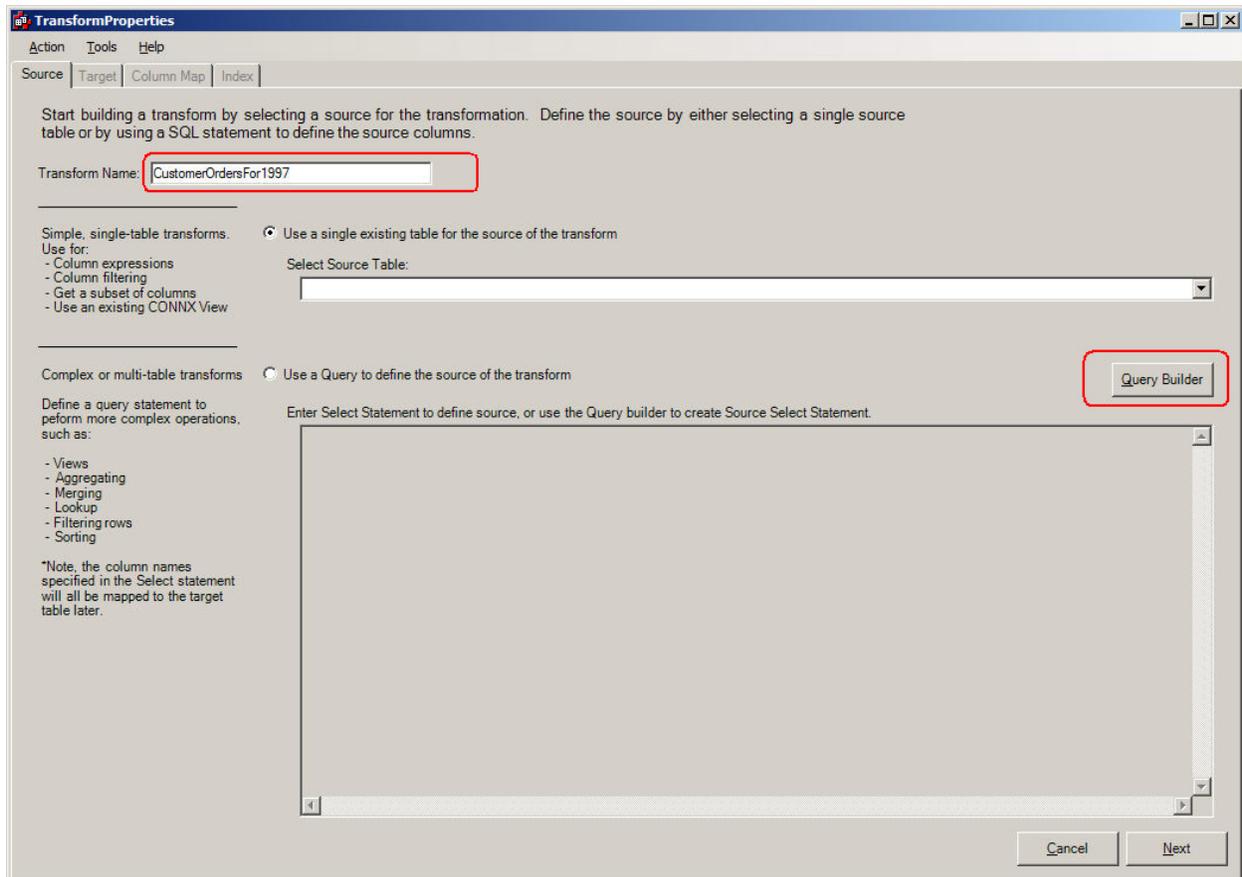


By pressing the next button, the next step in the wizard is [Assigning a Destination](#).

Designing a Transform in Query Builder

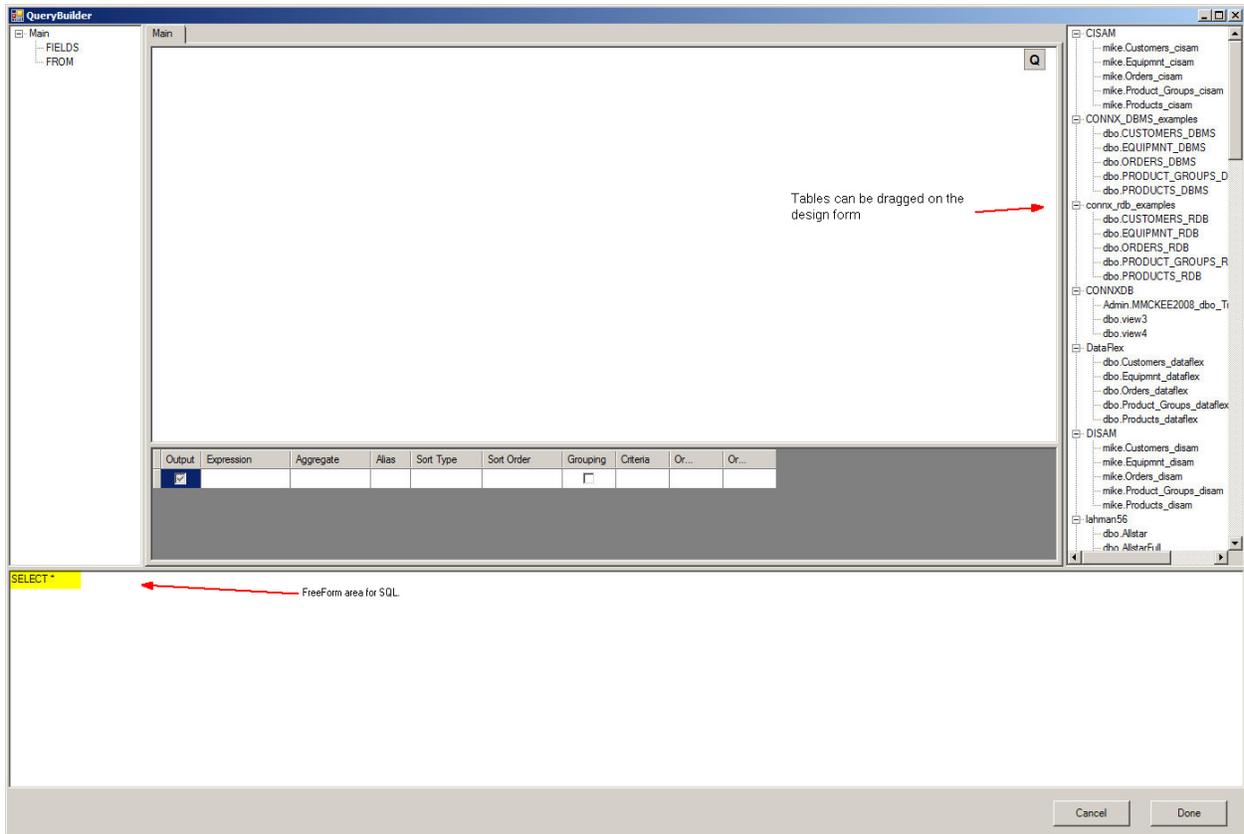
Certain Transformations are easier to be created when a graphical tool can enable the necessary tables/views/columns to be seen. DataSync has a graphical query tool that enables transformations to be easily created.

To navigate to the Query Builder, the user must select the Query Builder Button.



The QueryBuilder screen appears as follows:

CONNX Data Synchronization 12.5



In this example, the two tables (CUSTOMERS_RMS and ORDERS_RMS) will be used to generate a transform that produces a combined table that produces output only if the order occurred in 1997.

Only interested in the Customername, and CustomerPhone

Join arrow is produced by clicking left mouse button.

Select all the fields (including the possible unique key of OrderID, CustomerID, ProductID)

Fields will be displayed here. New ones can be added by adding text into the Expression below

Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria	Or...	Or...
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.orderid					<input type="checkbox"/>			
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.customerid					<input type="checkbox"/>			
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.productid					<input type="checkbox"/>			
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.orderdate					<input type="checkbox"/>			
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.productquantity					<input type="checkbox"/>			
<input checked="" type="checkbox"/>	RMS.dbo.customers_rms.customername					<input type="checkbox"/>			
<input checked="" type="checkbox"/>	RMS.dbo.customers_rms.customerphone					<input type="checkbox"/>			

```

SELECT RMS.dbo.orders_rms.orderid, RMS.dbo.orders_rms.customerid,
RMS.dbo.orders_rms.productid, RMS.dbo.orders_rms.orderdate,
RMS.dbo.orders_rms.productquantity, RMS.dbo.customers_rms.customername,
RMS.dbo.customers_rms.customerphone
FROM RMS.dbo.customers_rms INNER JOIN
RMS.dbo.orders_rms ON RMS.dbo.customers_rms.customerid =
RMS.dbo.orders_rms.customerid
    
```

The SQL Generated does not include the condition where the year must be 1997. The SQL to generate this is `YEAR(CISAM.mike.Orders_cisam.ORDERDATE) = '1997'`. This can be added in the grid below.

CONNX Data Synchronization 12.5

The screenshot shows the QueryBuilder interface with the following components:

- Left Panel (FIELDS):** Lists tables and their fields. Under 'FROM', 'RMS.dbo.orders_n' and 'RMS.dbo.customers_n' are visible.
- Main Panel:** Shows two tables: 'customers_rms (RMS.dbo.x)' and 'orders_rms (RMS.dbo.x)'. Fields are selected with checkboxes.

Table	Field	Type	Selected
customers_rms	customerid	string	<input checked="" type="checkbox"/>
customers_rms	customername	string	<input checked="" type="checkbox"/>
customers_rms	customeraddress	string	<input type="checkbox"/>
customers_rms	customercity	string	<input type="checkbox"/>
customers_rms	customerstate	string	<input type="checkbox"/>
customers_rms	customerzip	string	<input type="checkbox"/>
customers_rms	customercountry	string	<input type="checkbox"/>
customers_rms	customerphone	string	<input checked="" type="checkbox"/>
orders_rms	orderid	int:32	<input checked="" type="checkbox"/>
orders_rms	customerid	string	<input checked="" type="checkbox"/>
orders_rms	productid	int:32	<input checked="" type="checkbox"/>
orders_rms	orderdate	date	<input checked="" type="checkbox"/>
orders_rms	productquantity	int:32	<input checked="" type="checkbox"/>
- Table:** A table with columns: Output, Expression, Aggregate, Alias, Sort Type, Sort Order, Grouping, Criteria, Or...

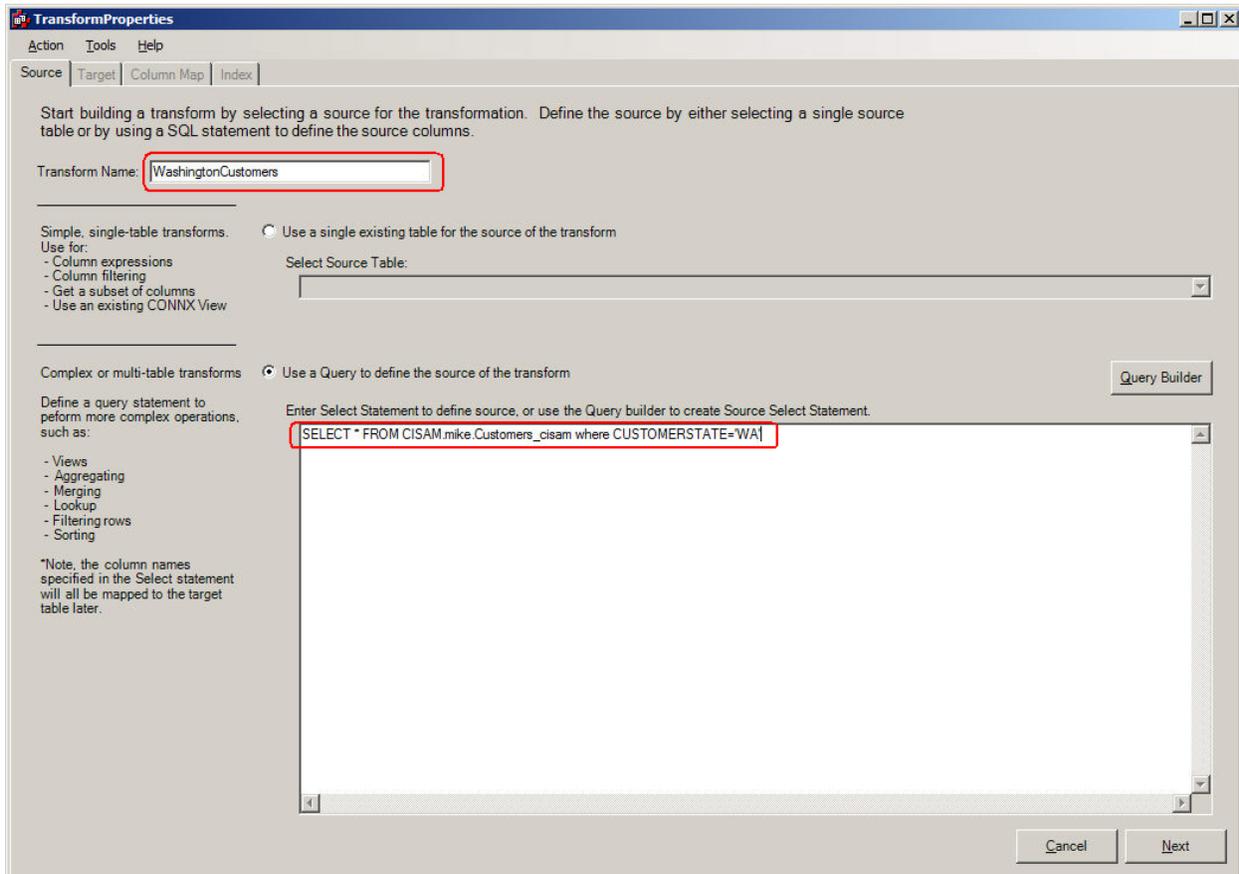
Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria	Or...
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.orderid					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.customerid					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.productid					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.orderdate					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	RMS.dbo.orders_rms.productquantity					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	RMS.dbo.customers_rms.customername					<input type="checkbox"/>		
<input checked="" type="checkbox"/>	RMS.dbo.customers_rms.customerphone					<input type="checkbox"/>		
<input type="checkbox"/>	YEAR(CISAM.mike.Orders_cisam.ORDERDATE)					<input type="checkbox"/>	= '1997'	
- Right Panel:** A tree view showing a database structure with folders like 'CISAM', 'CONNX_DBMS_examples', 'DataFlex', and 'DISAM'.
- Bottom Panel:** Contains the SQL query:


```
SELECT RMS.dbo.orders_rms.orderid, RMS.dbo.orders_rms.customerid,
RMS.dbo.orders_rms.productid, RMS.dbo.orders_rms.orderdate,
RMS.dbo.orders_rms.productquantity, RMS.dbo.customers_rms.customername,
RMS.dbo.customers_rms.customerphone
FROM RMS.dbo.customers_rms INNER JOIN
RMS.dbo.orders_rms ON RMS.dbo.orders_rms.customerid =
RMS.dbo.orders_rms.customerid
WHERE YEAR(CISAM.mike.Orders_cisam.ORDERDATE) = '1997'
```

By clicking the next button, the user will be directed to [Target Sub-Tab](#).

Designing a Free-From SQL Transform

If the user has a specific SQL Transform that they wish to create, it can be done easily with the Free-Form SQL Option. To do this option, the designer must select the Transform name, then either type or cut/paste from Infonaut the query that they wish to become a transform. If there is a syntax error, it will be caught when they select **Next**



Target sub-tab

Selecting a Target Destination

Selecting a Target Destination involves selecting a target file for an appropriate database. The designer has a few choices to make in this section.

1. Where will the output go?
2. Will it go to a new table?
3. Will it go to an existing table?
4. If it goes to an existing table, can the existing table be altered in case the transform has been changed?

All of these issues are addressed by the Target Window below. In many cases the default can be chosen, and the NEXT button can be selected, but the designer has the ability to select a different target database, owner, and table name, in addition to deciding to route the output to a new/existing table.

The screenshot shows the 'Transform Properties' dialog box for CONNX DataSync Transformation Server, with the 'Target' tab selected. The dialog is titled 'CONNX DataSync Transformation Server - Transform Properties' and has a menu bar with 'Action', 'Tools', and 'Help'. Below the menu bar are tabs for 'Source', 'Target', 'Column Map', and 'Index', with 'Target' being the active tab.

The main area is titled 'Select the target of the transform' and contains the following text: 'Select the destination table for the transform data. Selecting a new target table will allow you to design the structure of the destination table in the next screen, which will later be created during the first sync. Selecting an existing target table will allow data to be moved into a table that already exists on the destination database.'

There are three radio button options:

- Specify a new table for the target of the transform to be created on the first sync. This option includes three input fields: 'Select Database' (a dropdown menu), 'Select Owner' (a dropdown menu), and 'Specify the New Table Name' (a text box).
- Select an existing table to use for the Target in the transform. This option includes a dropdown menu showing 'SQL2005_32BIT_001.dbo.CustomerOrders'. Below this dropdown is a checkbox labeled 'Drop and Re-Create the existing target table.*' with a note: '* This allows the structure of a table that already exists in the destination database to be changed in the next screen'.
- Create a change data capture synchronization which tracks changes but does not move data. No target table is required.

At the bottom of the dialog, there are two sections:

- Data Change Detection Method:**
 - Standard CDC Method
 - Timestamp Field Method. This option includes a 'Timestamp Field:' dropdown menu and a 'Drift Seconds:' text box containing the value '5'.
- Change Default Full Synchronization Behavior:**
 - Use values from DataSync main screen. This section contains two radio button options:
 - Force drop and recreate of the target table
 - Force purging of the existing table
 - Method used to purge data:
 - Purge using DELETE FROM
 - Purge using TRUNCATE TABLE

At the bottom right of the dialog are three buttons: '< Prev', 'Next >', and 'Cancel'.

When the selection has been made, the third step in creating Transformation, [Column Mapping](#) will be performed.

Synchronization using Timestamp

Transform Properties

Action Tools Help

Source Target Column Map Index

Select the target of the transform

Select the destination table for the transform data. Selecting a new target table will allow you to design the structure of the destination table in the next screen, which will later be created during the first sync. Selecting an existing target table will allow data to be moved into a table that already exists on the destination database.

Specify a new table for the target of the transform to be created on the first sync

Select Database: Select Owner: Specify the New Table Name:

Select an existing table to use for the Target in the transform

SQL2005_64BIT.dbo.master2

Drop and Re-Create the existing target table.*

* This allows the structure of a table that already exists in the destination database to be changed in the next screen

Data Change Detection Method

Standard CDC Method

Timestamp Field Method

Timestamp Field: debut

Drift Seconds: 5

< Prev Next > Cancel

At the bottom of the transform Properties form is a section called "Data Change Detection Method".

Normally, the radio button for the option "Standard CDC Method" should be selected; however, under special circumstances, "Timestamp Field Method" can be chosen. This method requires two special conditions to operate correctly. First, there must be a column that contains timestamp information within the table that is updated with the current timestamp every time the record is updated. Second, there must not be any deletes performed against the data (unless you do not care if the deletes get propagated to the target). The "Timestamp Field Method" only operates against inserts and updates of the data, and deletes are never performed. You can think of this method as an archival of any inserts or updates to the data. When these two special conditions are present, the "Timestamp Field Method" is the preferred method of synchronization.

Put another way, the source table must contain a timestamp field that will show the date and time that each record is updated. Also, any records deleted from the source table will not be deleted from the target table.

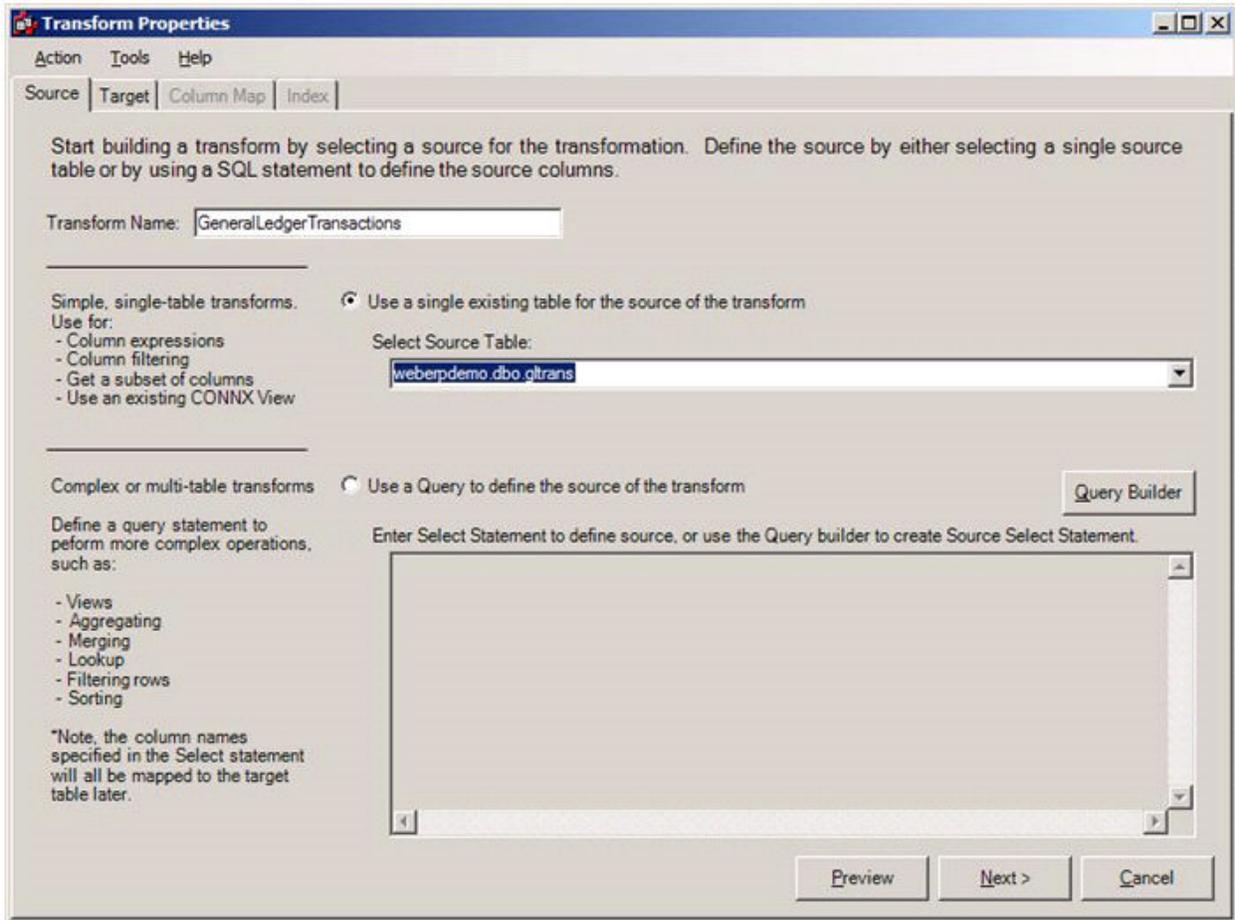
When the "Timestamp Field Method" radio button is chosen, a drop-down list will show all available timestamp fields. Choose from this list the field updated with a new timestamp every time the record is modified.

The edit field called "Drift Seconds" is used to handle time drift for the source data database machine.

For example, if the source database machine is synchronized on a daily basis using an NTP service, the apparent time of the machine may suddenly change by a small amount. To avoid the possibility of missing a crucial data update, a "drift seconds" field is supplied that allows for time corrections to the

machine's internal clock. Due to the existence of leap seconds, this field should always be set to at least one second, even if the clock on the source database machine is perfectly accurate.

So let's examine how we might use "Timestamp Field Method" on a table that contains a field we can use for this purpose. We will create a transform called GeneralLedgerTransactions against a MySQL table called gltrans:



Because this table has a field that contains date/time information called transdate, we will use this field to perform our synchronization.

Transform Properties

Action Tools Help

Source **Target** Column Map Index

Select the target of the transform

Select the destination table for the transform data. Selecting a new target table will allow you to design the structure of the destination table in the next screen, which will later be created during the first sync. Selecting an existing target table will allow data to be moved into a table that already exists on the destination database.

Specify a new table for the target of the transform to be created on the first sync

Select Database: Select Owner: Specify the New Table Name:

Select an existing table to use for the Target in the transform

Drop and Re-Create the existing target table.
* This allows the structure of a table that already exists in the destination database to be changed in the next screen

Data Change Detection Method

Standard CDC Method

Timestamp Field Method

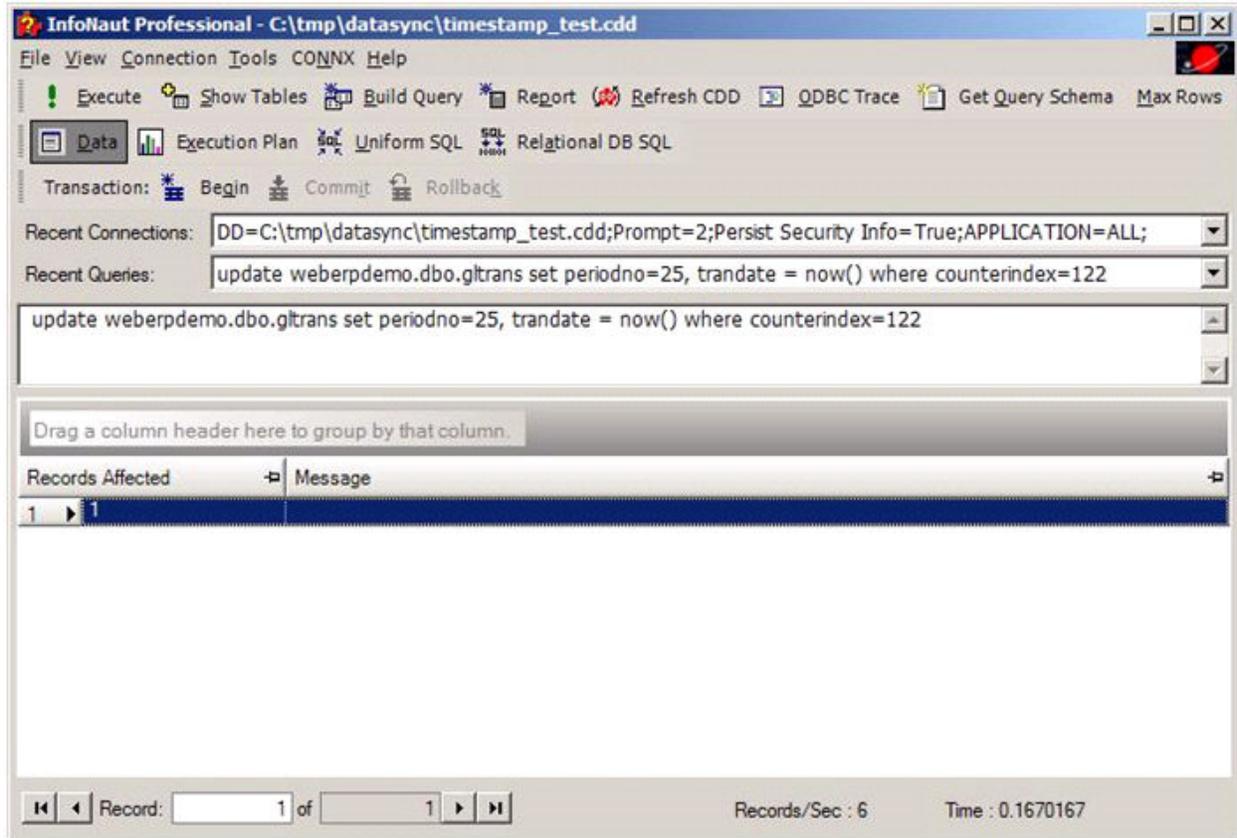
Timestamp Field:

Drift Seconds:

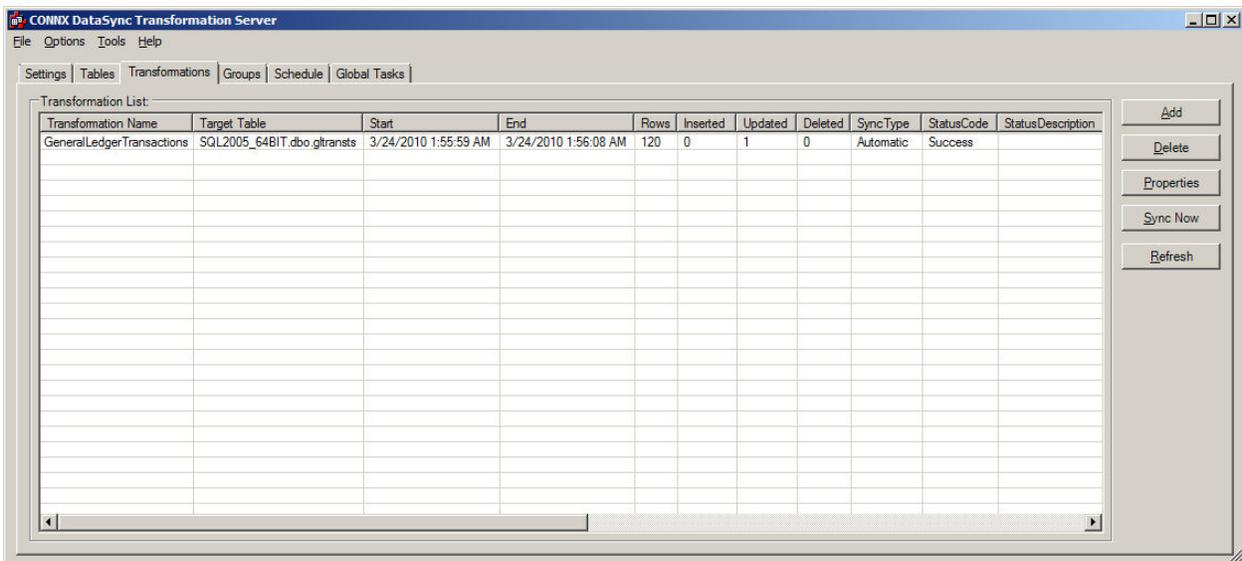
< Prev Next > Cancel

First we perform a full synchronization to get a perfect copy of the source data defined on the target table. Next we will change a record in the source table, updating the transaction date along with the period.

CONNX Data Synchronization 12.5



Having modified a record in the source table, we perform an automatic synchronization and get this result:



It may seem surprising that we see five records updated instead of one record. This oddity is due to the fact that we synchronize all changes up to and including the last moment in the data from our previous synchronization (with an error window of "drift seconds").

counterindex	type	typeno	chequeno	trandate	periodno	account	narrative	amount	posted	jobref	tag
103	105	10	2	0	05/18/2009	24	5000	DUMBLE - SLIC...	12.018	0	0
104	106	10	2	0	05/18/2009	24	1460	DUMBLE - SLIC...	-12.018	0	0
105	107	10	2	0	05/18/2009	24	4100	DUMBLE - SLIC...	-6.25	0	0
106	108	10	2	0	05/18/2009	24	1100	DUMBLE	6.25	0	0
107	109	11	3	0	05/16/2009	24	5000	DUMBLE - SLIC...	-12.02	0	0
108	110	11	3	0	05/16/2009	24	1460	DUMBLE - SLIC...	12.02	0	0
109	111	11	3	0	05/16/2009	24	4100	DUMBLE - SLIC...	6.25	0	0
110	112	11	3	0	05/16/2009	24	1100	DUMBLE	-6.25	0	0
111	113	35	18	0	05/17/2009	24	5700	Cost roll on rele...	0.000999999	0	0
112	114	35	18	0	05/17/2009	24	1460	Cost roll on rele...	-0.000999999	0	0
113	115	28	6	0	05/17/2009	24	1440	10 - SLICE Com...	9.01275	0	0
114	116	28	6	0	05/17/2009	24	1460	10 - SLICE -> B...	-9.01275	0	0
115	117	35	19	0	05/17/2009	24	5700	Cost roll on rele...	0.00174999	0	0
116	118	35	19	0	05/17/2009	24	1460	Cost roll on rele...	-0.00174999	0	0
117	119	28	7	0	05/17/2009	24	1440	10 - SLICE Com...	1.80255	0	0
118	120	28	7	0	05/17/2009	24	1460	10 - SLICE -> B...	-1.80255	0	0
119	121	26	7	0	05/17/2009	24	1460	10 SLICE - Slice...	1.80255	0	0
120	122	26	7	0	03/24/2010	25	1440	10 SLICE - Slice...	-1.80255	0	0
*	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>

Since our oldest transaction date from the previous cycle was 5/17/2009, we see that the records with counterindex values of 105, 106, 107, and 108 have also been transferred to the target table. (These are known as "time window" updates.)

CONNX Data Synchronization 12.5

Recent Connections: DD=C:\tmp\datasync\timestamp_test.cdd;Prompt=2;Persist Security Info=True;APPLICATION=ALL;

Recent Queries: SELECT * FROM SQL2005_64BIT.dbo.gltrans

SELECT * FROM SQL2005_64BIT.dbo.gltrans

counterindex	type	typeno	chequeno	trandate	periodno	account	narrative	amount	posted	jobref	tag
104	110	11	3	0	5/16/2009..	24	1460	DUMBLE - SLIC...	12.02	0	0
105	111	11	3	0	5/16/2009..	24	4100	DUMBLE - SLIC...	6.25	0	0
106	112	11	3	0	5/16/2009..	24	1100	DUMBLE	-6.25	0	0
107	113	35	18	0	5/17/2009..	24	5700	Cost roll on rele...	0.00099999...	0	0
108	114	35	18	0	5/17/2009..	24	1460	Cost roll on rele...	-0.00099999...	0	0
109	115	28	6	0	5/17/2009..	24	1440	10 - SLICE Com...	9.01275	0	0
110	116	28	6	0	5/17/2009..	24	1460	10 - SLICE -> B...	-9.01275	0	0
111	117	35	19	0	5/17/2009..	24	5700	Cost roll on rele...	0.00174999...	0	0
112	118	35	19	0	5/17/2009..	24	1460	Cost roll on rele...	-0.00174999...	0	0
113	119	28	7	0	5/17/2009..	24	1440	10 - SLICE Com...	1.80255	0	0
114	120	28	7	0	5/17/2009..	24	1460	10 - SLICE -> B...	-1.80255	0	0
115	121	26	7	0	5/17/2009..	24	1460	10 SLICE - Slice...	1.80255	0	0
116	105	10	2	0	5/18/2009..	24	5000	DUMBLE - SLIC...	12.018	0	0
117	106	10	2	0	5/18/2009..	24	1460	DUMBLE - SLIC...	-12.018	0	0
118	107	10	2	0	5/18/2009..	24	4100	DUMBLE - SLIC...	-6.25	0	0
119	108	10	2	0	5/18/2009..	24	1100	DUMBLE	6.25	0	0
120	122	26	7	0	3/24/2010..	25	1440	10 SLICE - Slice...	-1.80255	0	0
*	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>	<null>

Record: 120 of 120 Records/Sec : 265 Time : 0.4520452

The result is that our target table now contains the same information as the original source data.

Remember that deletes performed against the source data using this technique will not be reflected on the target table. Hence, if five records had been removed from the source table, we would still see them on the target table.

Also, if we synchronize the table again without making any changes, we will (in this case) get one record updated. The reason that we see an apparent update is that the most recent date from the data in our last synchronization is within the window of: (newest timestamp transferred +/- drift seconds). Thus, we will always see at least one record updated every time we synchronize using this method.

Settings | Tables | Transformations | Groups | Schedule | Global Tasks

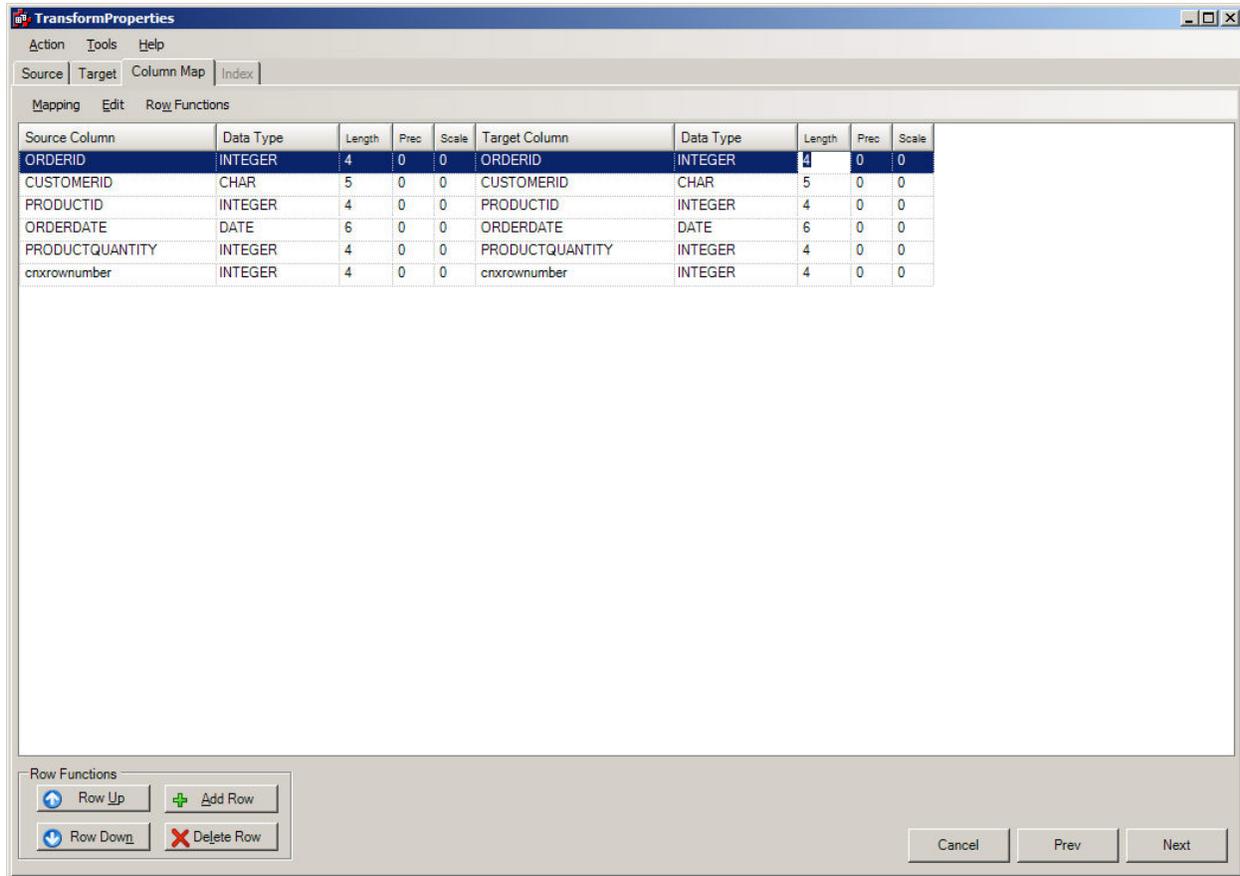
Transformation Name	Target Table	Start	End	Rows	Inserted	Updated	Deleted	SyncType	StatusCode
GeneralLedgerTransactions	SQL2005_64BIT.dbo.gltrans	3/24/2010 1:55:59 AM	3/24/2010 1:56:08 AM	120	0	1	0	Automatic	Success

Buttons: Add, Delete, Properties, Sync Now, Refresh

Column Map sub-tab

Column Mapping to new Target Table

Column mapping involves the proper mapping of Source to Target Columns. In some cases where Transforms have been designed using Free-Form SQL, or designed in the Query Builder, the defaults that have been provided might enable the user to simply hit the NEXT button, which will move them to the Index Mapper screen.



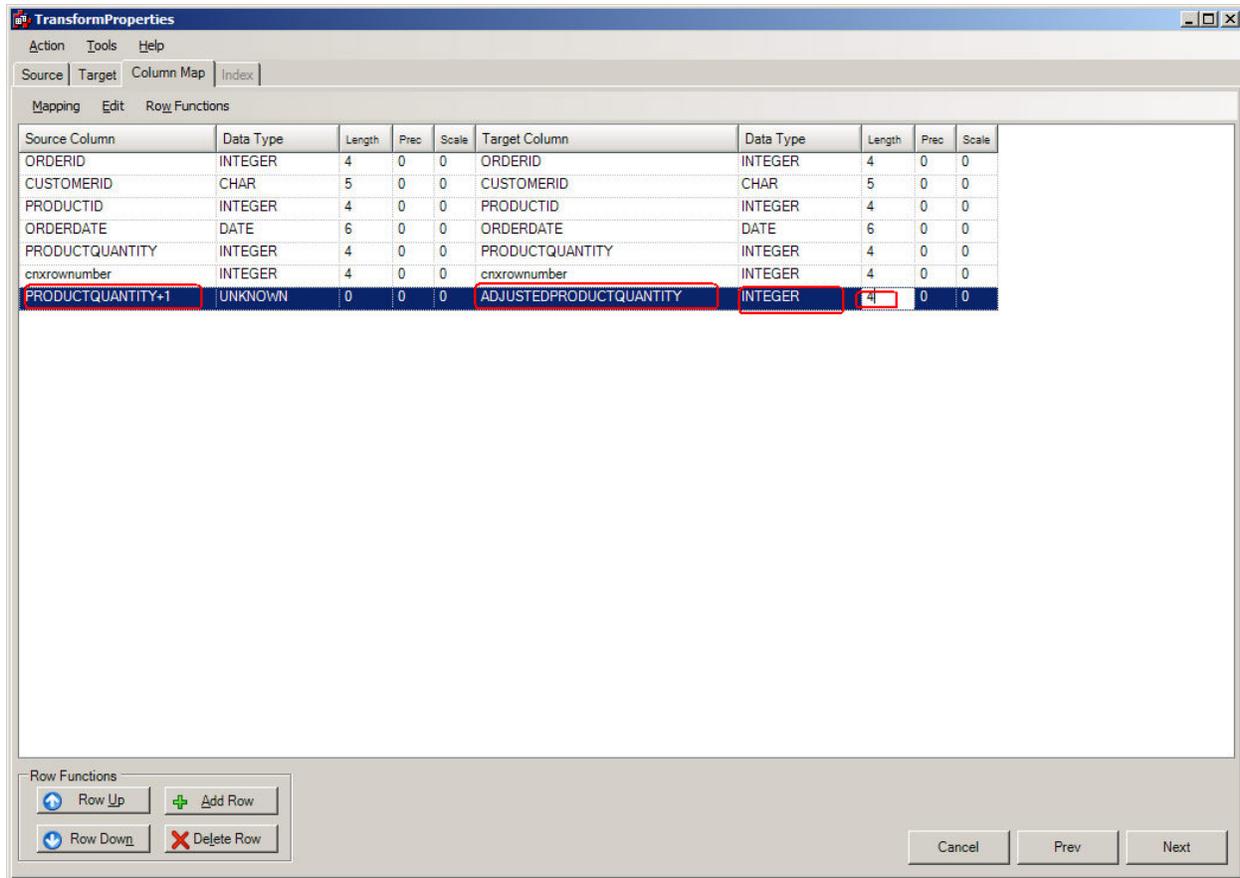
However, the following scenarios will involve using the GUI controls to select the proper Column Mapping.

New Derived Column	New Columns can be added when the Designer has selected Single Table Transform. The Designer will select the "Add Row" button. This will produce an empty row in which the user can create a Derived field using SQL Language. An example could be:	Single Table Transforms
--------------------	---	-------------------------

	<i>Order Quantity +1 as Undercapitalized</i>	
Deleting a Column	If Single Table Transform is selected, there might be situations where certain columns aren't needed in the target table. The designer can highlight the specific row, and choose to delete it. The "Delete Row" button will be pressed.	Single Table Transforms
Moving Row up and down	This has the net effect of moving the specified column so that it appears in a different location with respect to the Created table. For instance if order id was listed as the first column, but the designer wanted it as the third column, the designer could press the Row Up/Row Down button to change its location.	Single Table Transforms Query Builder Transforms Free-Form SQL Transforms
Adjustments to Target Table Attributes	Target Table names, datatypes, lengths, precisions, and scales can be changed. Generally, this should be done with caution, as the default values might be appropriate.	Single Table Transforms Query Builder Transforms Free-Form SQL Transforms

Examples:

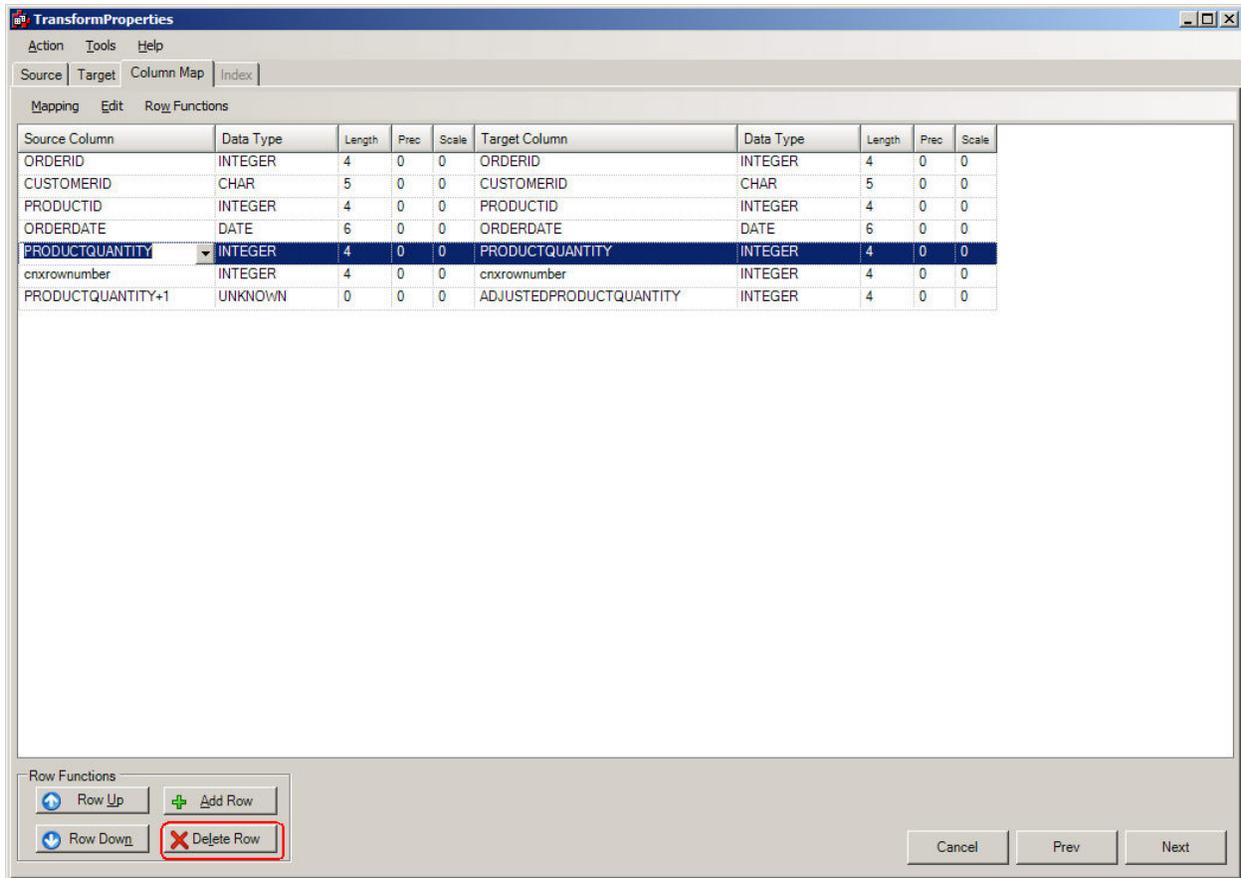
Adding a Derived Column



After Add Row was selected, the designer needs to enter the fields above that are circled. In the case of Data Type, a drop-down list is provided. In the case of the Column Names, The user should type the appropriate Target Column Name, and SQL Transformation.

Delete Row

CONNX Data Synchronization 12.5

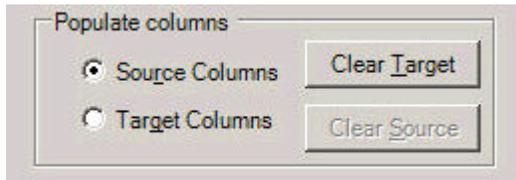


In this scenario, the highlighted row will be deleted. It is important to avoid deleting columns that might be a unique key, as Incremental synchronization depends on having a unique key.

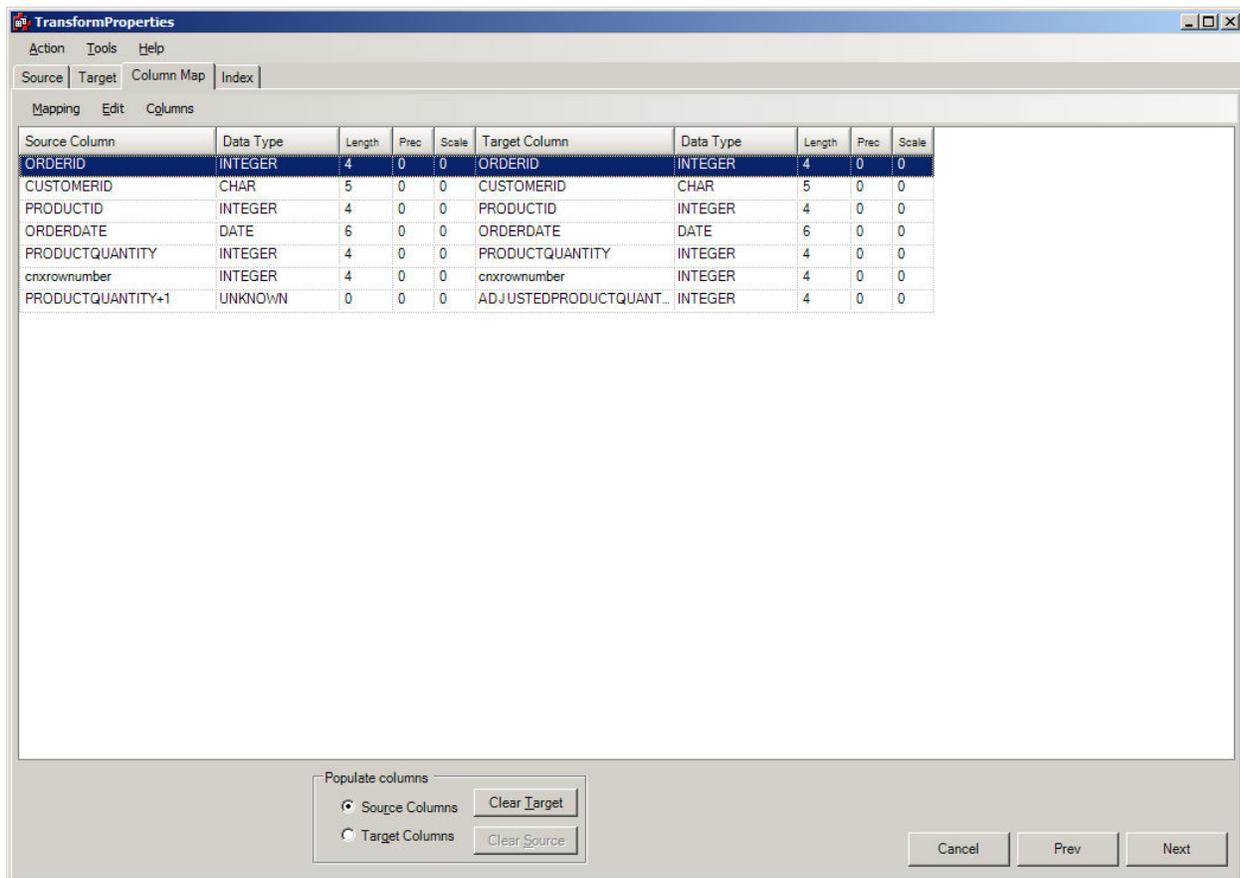
After the designer is done with column mapping, the last part of the Transformation tasks deals with [Index Mapping](#).

Column Mapping to Existing Target Table

In the situations where a target table exists, column mapping must not alter the physical target table. New target columns cannot be added, but adjustments can be made to alter the source feed. The screen appears slightly different, as there is a new Populate Columns Panel.



The Column Mapping to Existing Target Table screen appears as:



Let suppose that ProductQuantity was no longer desired. The user must navigate to the ProductQuantity column on the Source Column, and blank it out. This will prevent ProductQuantity from being used. Derived Fields can be changed, but the Target Table Column that the Derived Column feeds to CANNOT be altered.

CONNX Data Synchronization 12.5

For cases where the Designer wishes to have the flexibility of mapping each column individually, the "CLEAR TARGET/CLEAR SOURCE buttons were created. This has the net effect of clearing either the Source Mappings or Target Mappings. This is generally done in the case where there is a large discrepancy between the Source Feed, and the existing Target Table.

Example 1:

Target Table A had 1000 columns, but the source feed only represented 10 Columns.

The designer might do the following:

1. Select Source Columns radio button, and press the Clear Target. This will remove the Target Mappings from the screen.
2. Individually map each Target Column to the Source Column. This is accomplished by using the Drop-Down List box for the Target Column names. You will need to do this for the 10 columns desired.

Example 2:

Target Table A has 10 Columns, but the Source Feed has 10000 columns.

1. Select Target Columns radio column, and press the Clear Source. This will remove the Source mappings from the screen.
2. Individually map each Source Columns by using the Drop-Down list box for the Solumn Column Names. You will need to do this for the 10 columns desired.

In certain scenarios, the Designer might want to Clear the Source or Target Columns. This might be done in the case where the Target Table has hundreds of columns, and the designer wants to map a source feed that is much smaller.

Index sub-tab

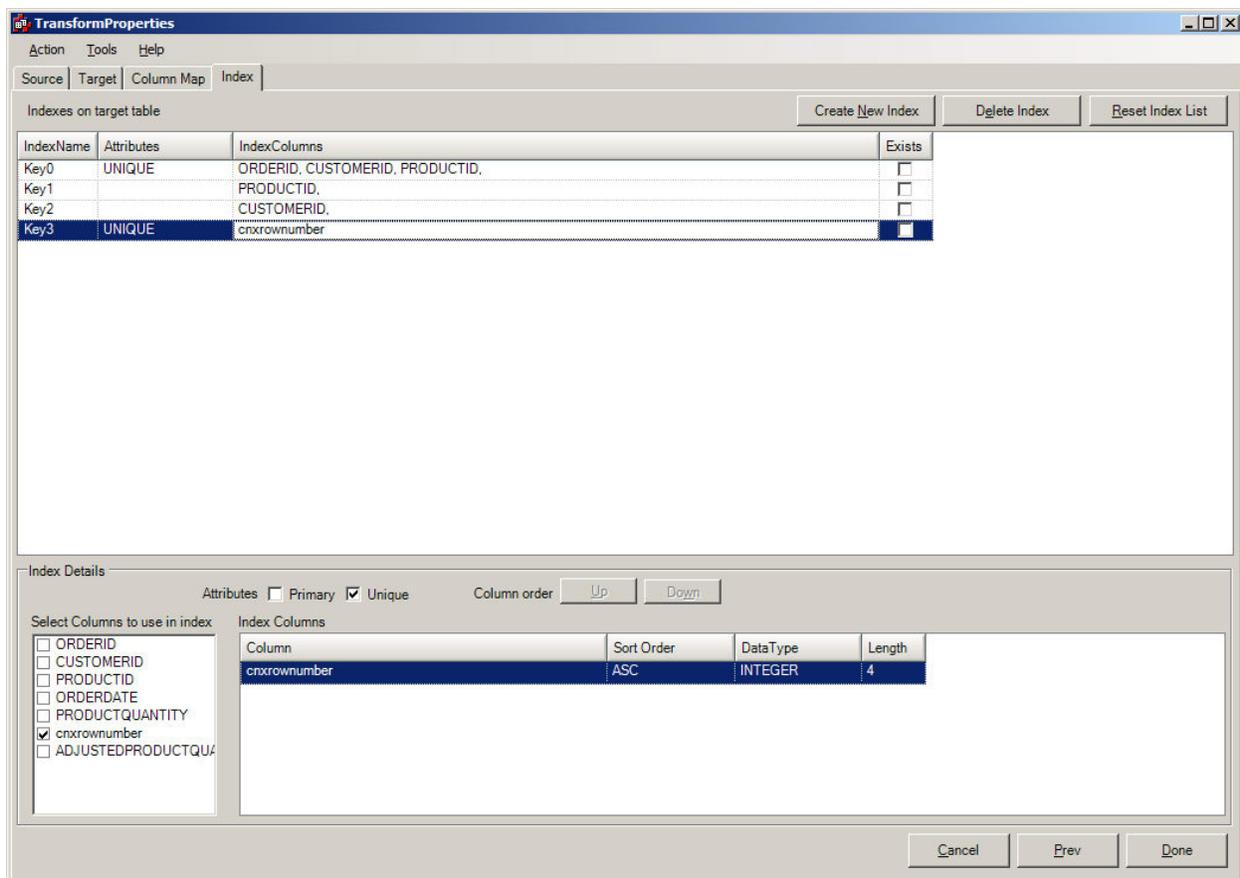
Index Mapping

Datasync offers a way to provide Index Mapping to the designated Transformation. The Index Mapping tab offers the following functionality:

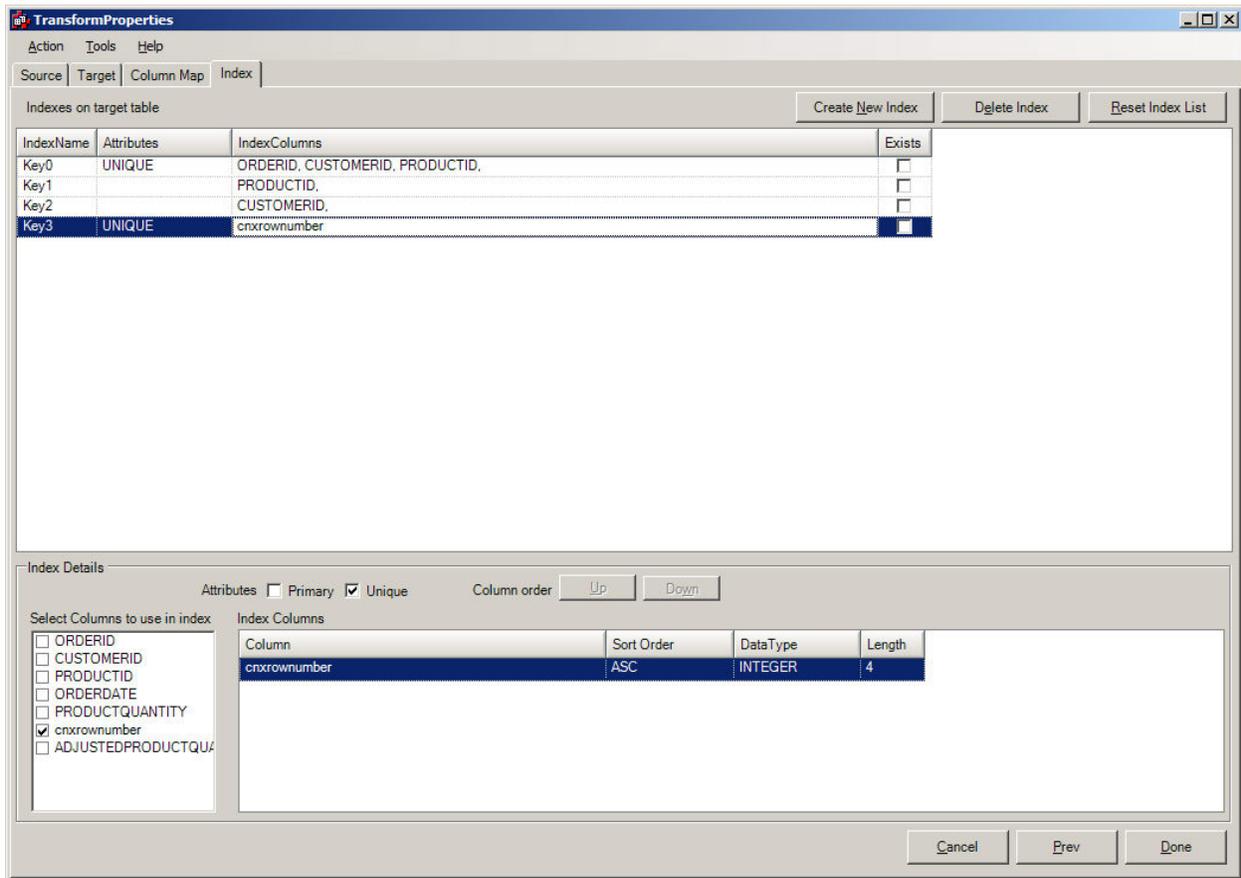
- Ability to propagate "suggested" indexes that it has harvested from the source tables to the target table
- Ability to create a new index
- Ability to delete an index

For Single Table Transforms, and Aggregate Transforms, the defaults given might not require any user intervention. The goal of index mapping is to select which indexes will be added to the target table. **It is essential that a unique key be created in order for Incremental Synchronization to work.** A unique key is not required in cases where Incremental Synchronization is not desired. When these transforms are synchronized, a full synchronization will be performed.

The screen may appear as:



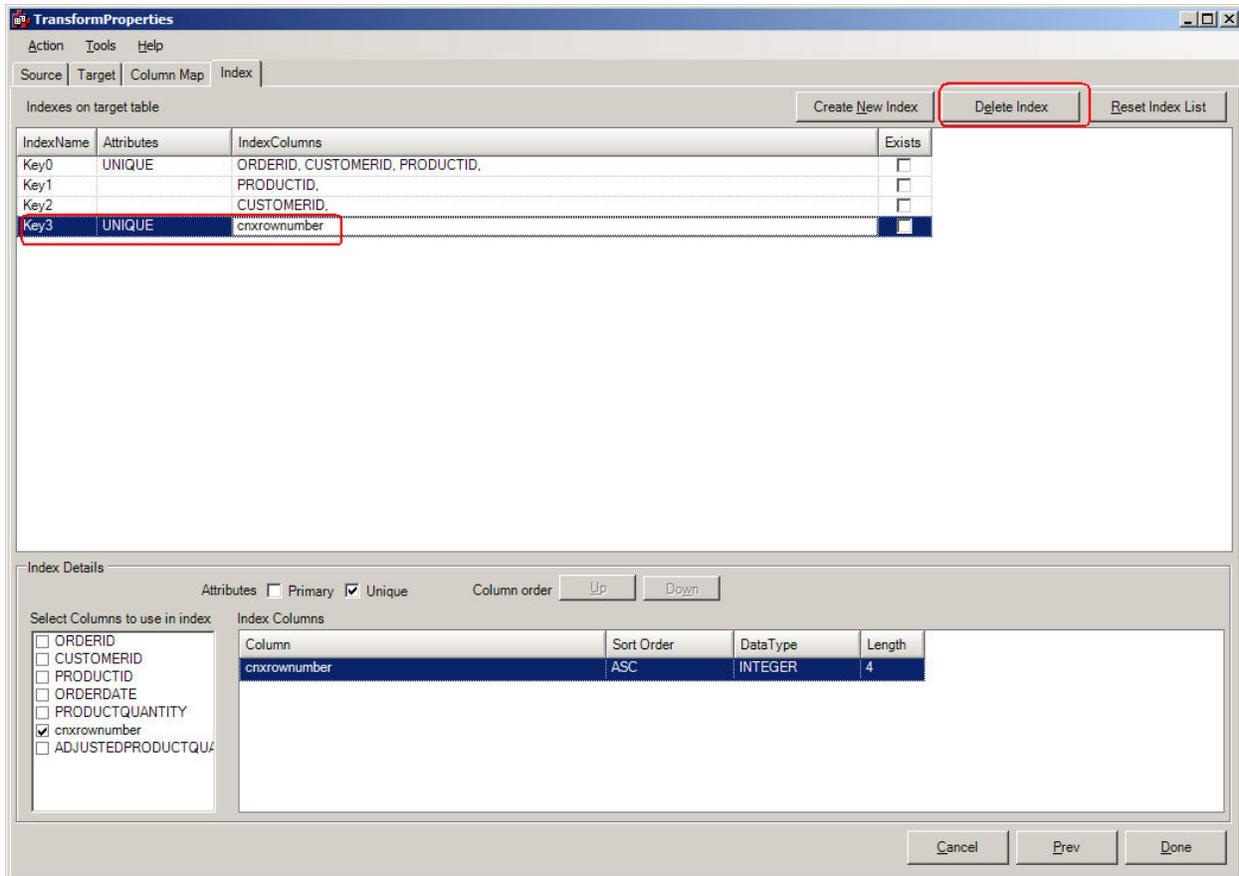
The actions that are available to the user are:



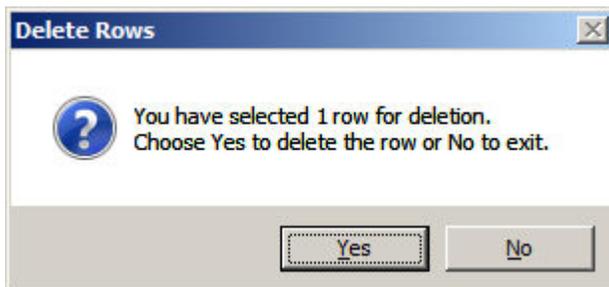
Create an Index	By doing this, the user will be prompted to select the appropriate columns that make up the index. They will also determine whether it is unique, and the order of the columns (in case it is a compound key). This is caused by selecting the "CREATE INDEX" button.
Delete an Index	This will delete the index from the target table. This is caused by selecting the "DELETE INDEX" button.
Reset Index List	This will reset the indexes so that they appear as they were before entering this screen.

Examples:

Deleting an Index



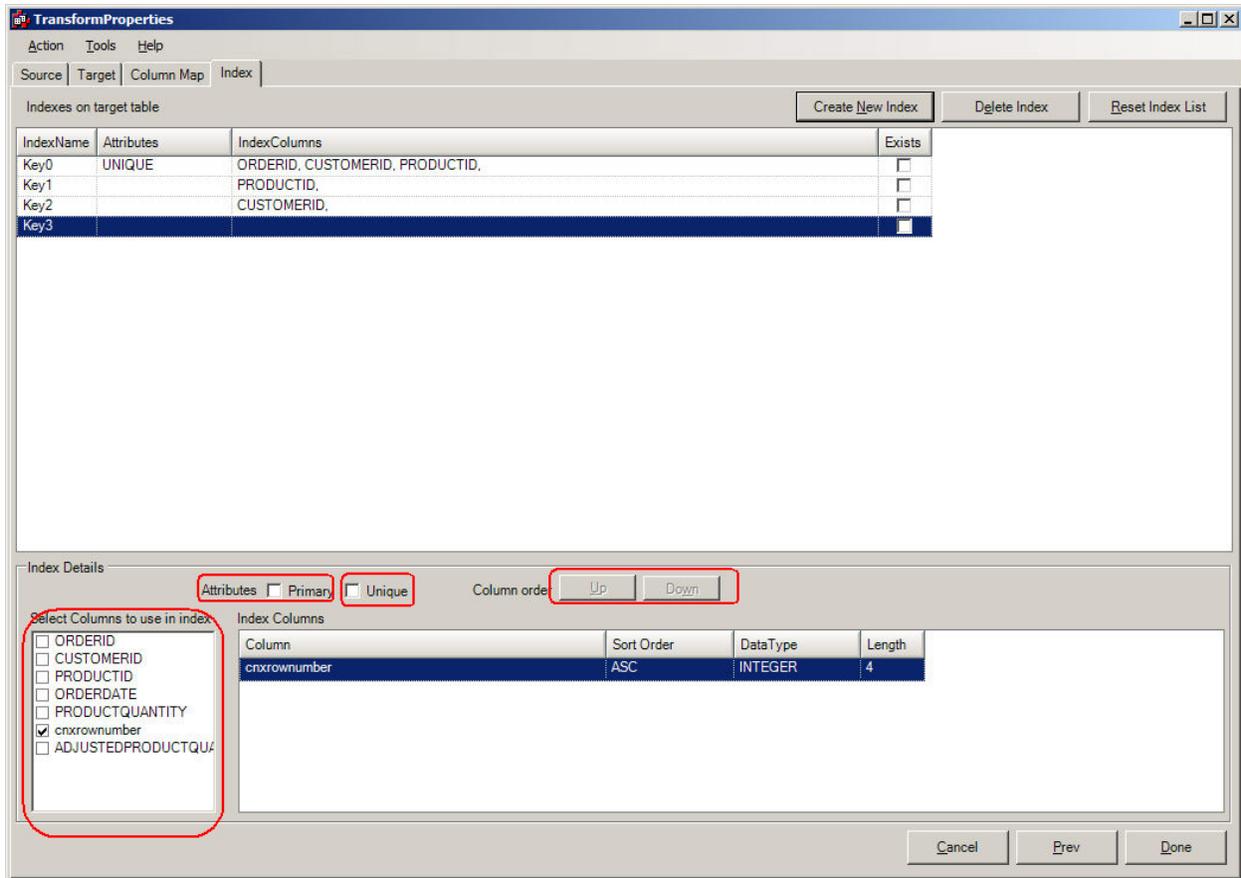
After this has been done, there will be a prompt:



If Yes is the selection, the Index will be deleted.

Adding an Index

By pressing the Create Index Button, an empty row will appear.



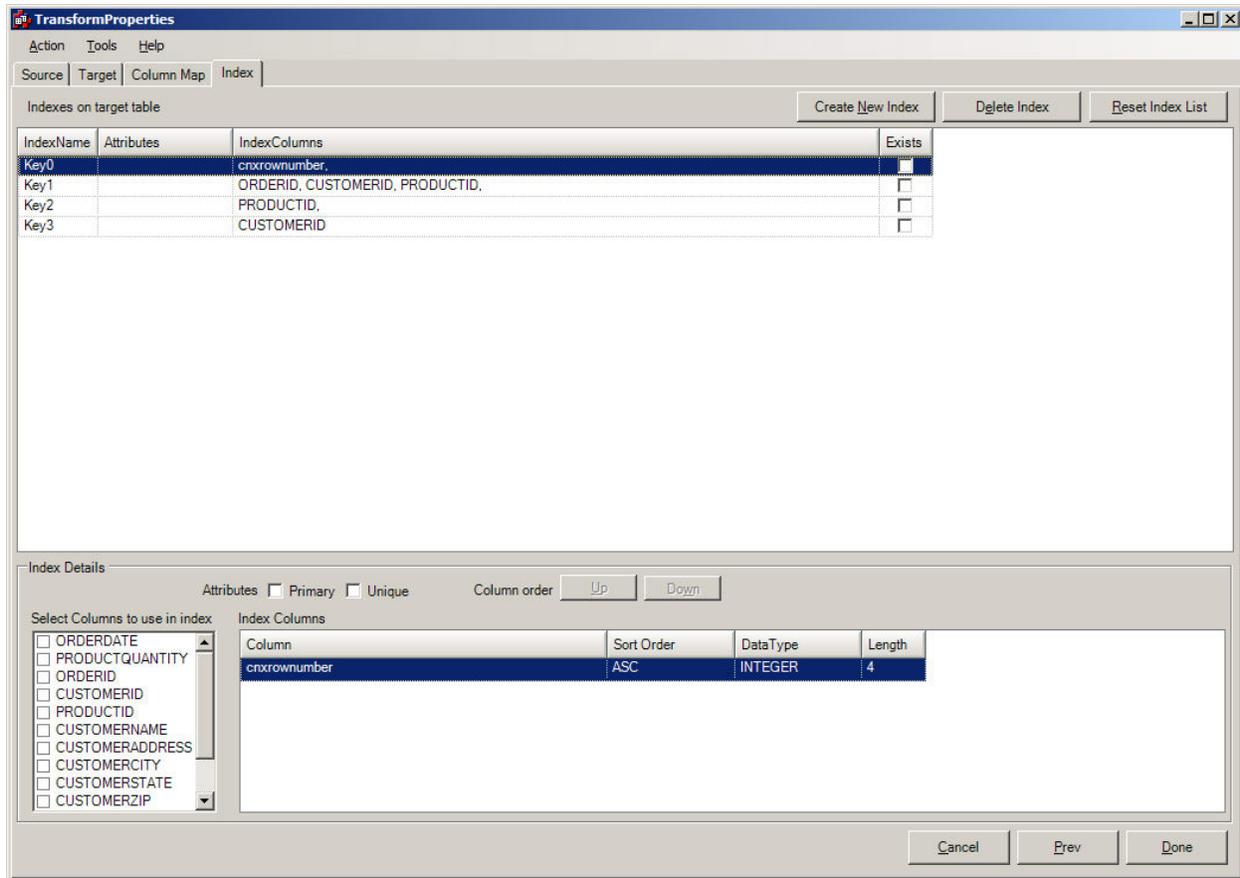
The items that are circled are controls in which the user can influence the column(s) that make up the Index in the Index Columns Pane.

Creating a Unique Index for a Transform that has multiple tables

DataSync attempts to determine all of the possible keys that a transform might have. To do this, it looks at the source feed that makes up the transform, and checks to see whether these column(s) are indexed. If they are, they are possible candidates for target table indexes. It is up to the designer to decide whether the default indexes that are shown are valid for the target feed.

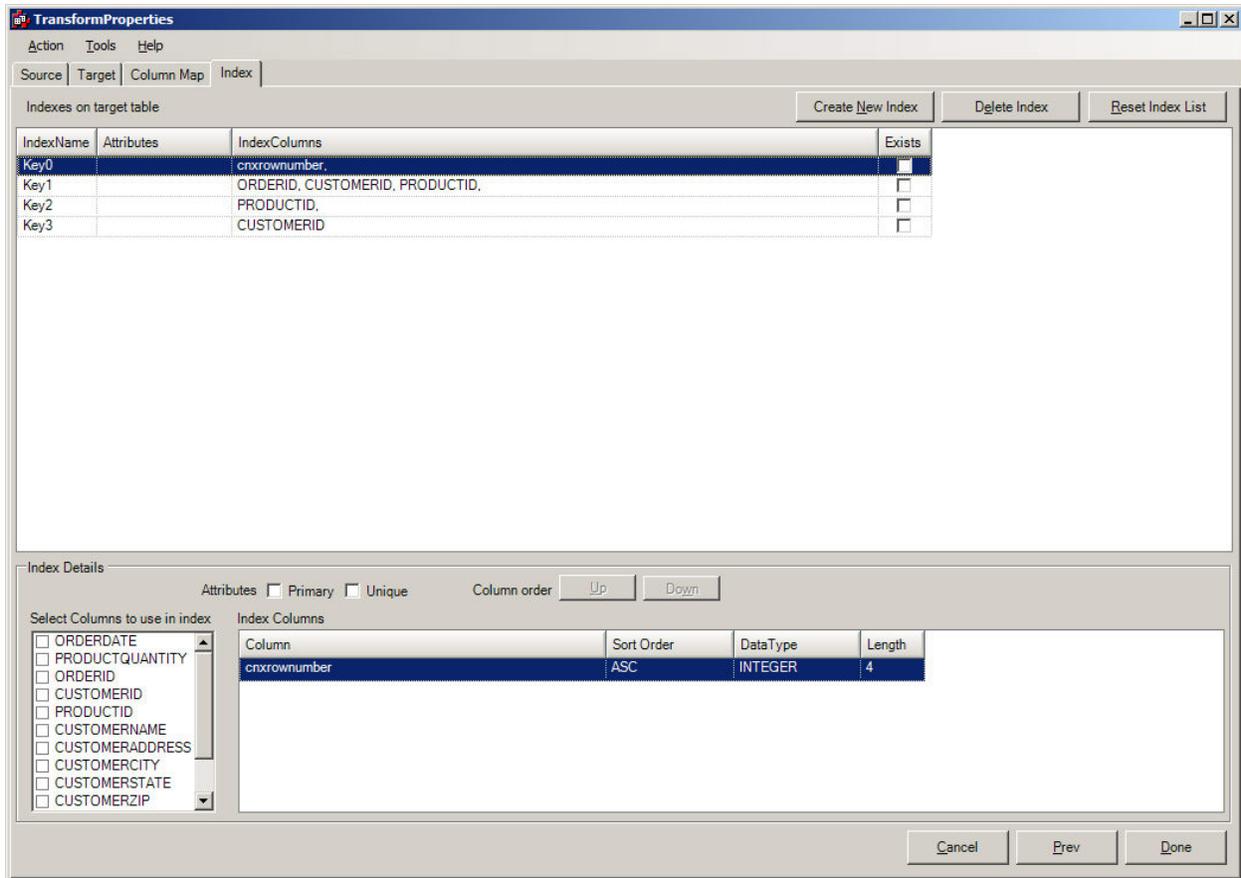
When a transform is composed of more than a single table, all potential keys are extracted from the source table inputs and are listed as defaults. Datasync cannot determine whether the key is unique, due to the fact that it is a "joined" table, and a unique key for a single table, might not be a unique key for the joined transform.

Consider this example:



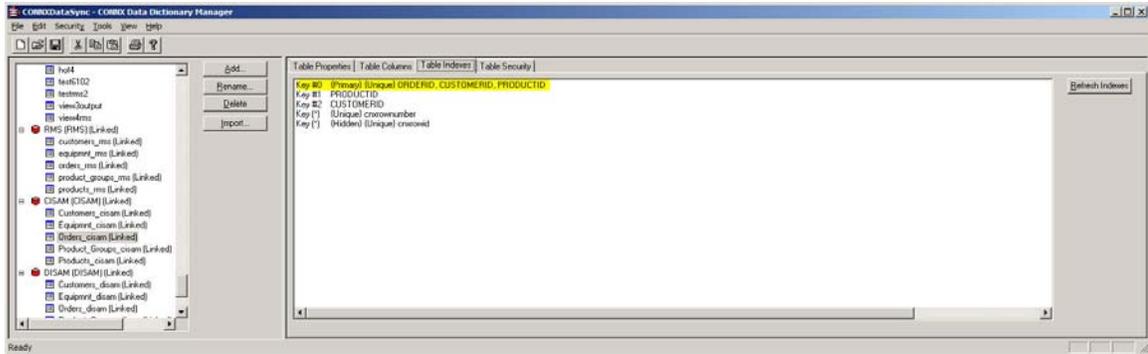
When the Index Mapper screen is reached, it displays all of the indexes that make up this transformation. It has retrieved this index information by displaying all the indexes for:

- Orders_cisam
- Customers_cisam

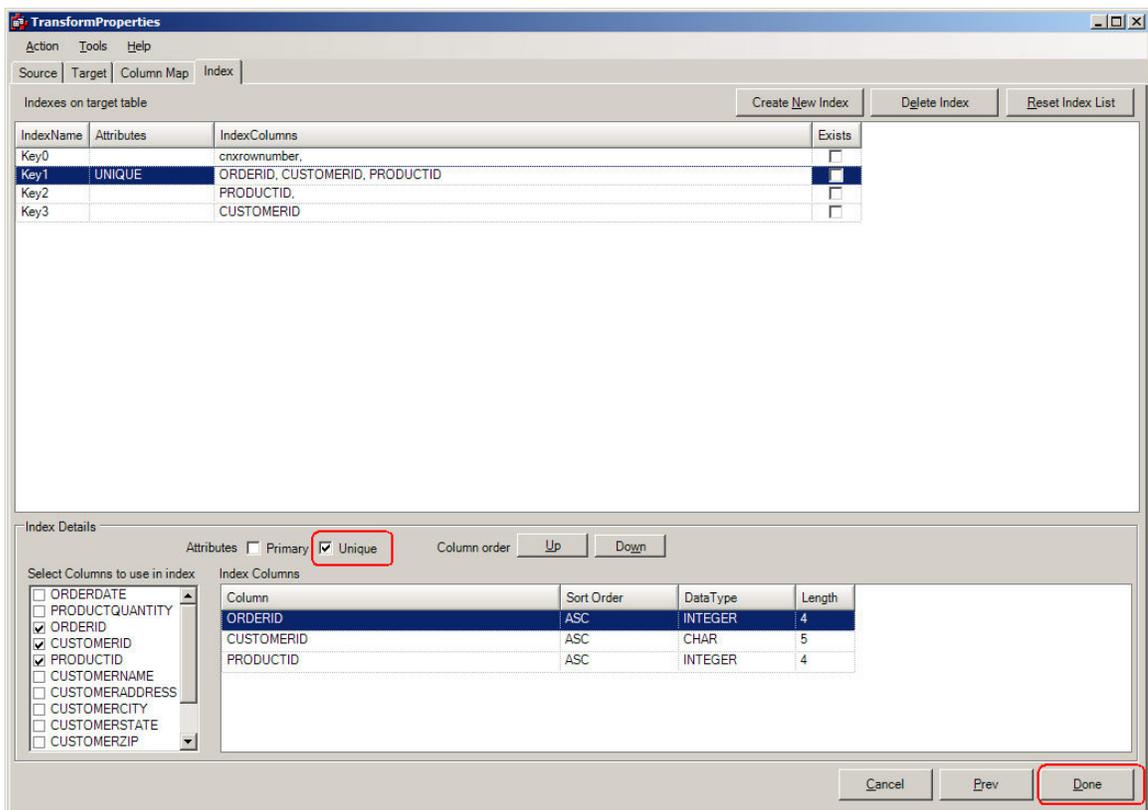


None of the Indexes show an unique key (If one existed, it would show up in the Attributes column for the specified key). Therefore, this transformation CANNOT be incrementally synchronized. The designer might intuitively know that a unique key could be composed of ORDERID, CUSTOMERID, PRODUCTID. In fact, the ORDERS_CISAM table recognizes that combination as an unique key. If the designer needs assistance on determining how to make an unique key, the following steps should be followed:

1. In the two table join, which table is considered the "child" table. This means when they are connected to each other, which table will produce more than 1 occurrence if there are matches. A good way to determine this is to examine the column(s) that join the tables. Which table represents the joined column(s) as an unique key. This is generally the Parent Table (Customers_cisam in this example). Therefore, the other table will be the Child Table (Orders_cisam).
2. Go to TOOLS/Launch CONNX Data Dictionary Tool from the Menu bar, and look for the Orders_cisam table. When you have found this table, look at the indexes for this table.



3. The highlighted Key #0 is unique, which represents the ORDERID, CUSTOMERID, and PRODUCTID. This could be considered a valid unique key.
4. Go back to the Index Mapper screen, and complete the Index Transformation.

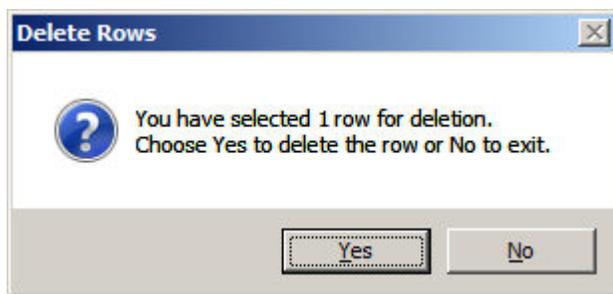


5. Click Unique checkbox, and then Click the Done button. This action will result in the Target Table having a unique key of ORDERID, CUSTOMERID, and PRODUCTID. The only way to determine whether the key is actually unique is to run a manual synchronization of this transformation. If there is a duplicate, an error will result, and the Transformation should be edited to identify a proper unique key.

Creating a Unique Index for a Transform that does not show a potential unique or non-unique key in the default list

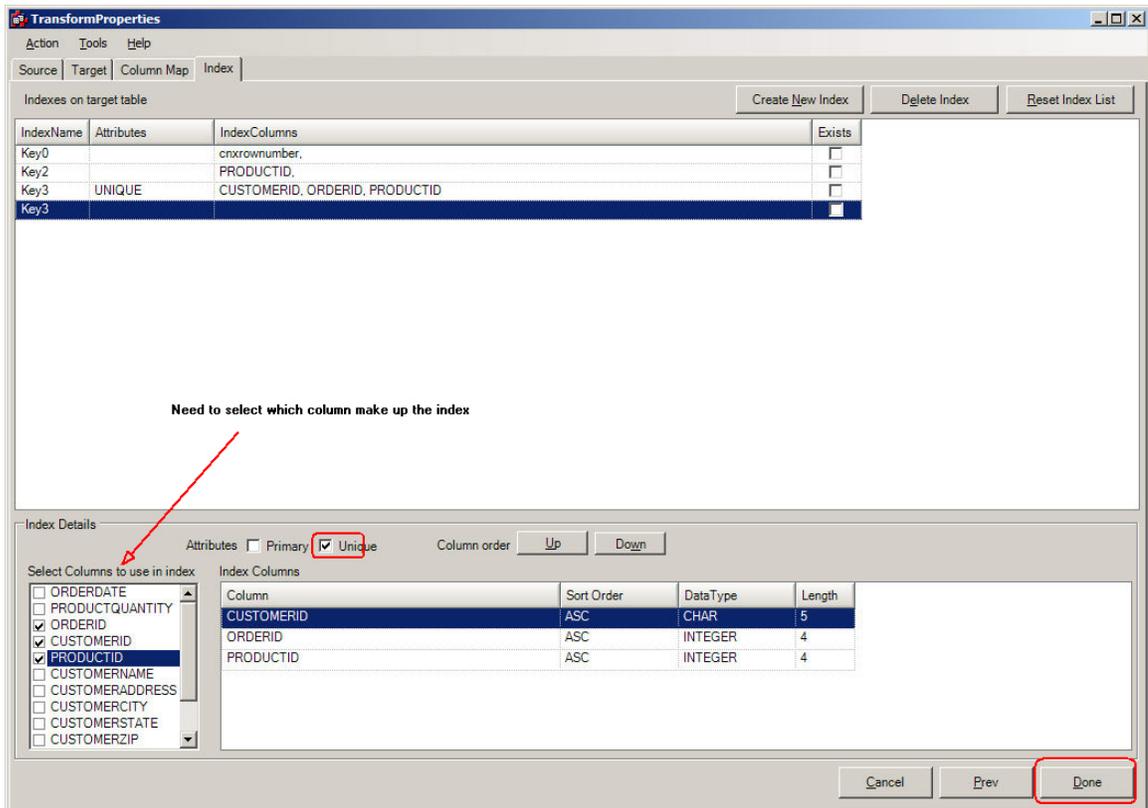
When the potential indexes are shown, the user might not see a potential key that could be used as an unique key. The user has two choices:

1. If the Index is not shown, this could be due to the fact that a column that potentially identifies the unique key was not part of the transformation. Consider the last example. If PRODUCTID was not part of the select list, the key (ORDERID, CUSTOMERID, PRODUCTID) would not be displayed. The designer might have to revise the transformation to include this column.
2. If the index is not shown, the user might have to create a unique key by selecting the CREATE INDEX button from above. This button can be used in creating all types of keys (unique, composite, non-unique). To demonstrate this feature, the DELETE INDEX functionality will be demonstrated. The example will delete the ORDERID, CUSTOMERID, PRODUCTID composite key, and then add it back. The first step is to click the DELETE INDEX button. This will produce the screen:



After clicking yes, then the Index Mapper will not see that Index.

3. Click the CREATE INDEX button.
4. This will produce a new display that shows an empty row, which will be filled in automatically when the designer tells which columns to use.



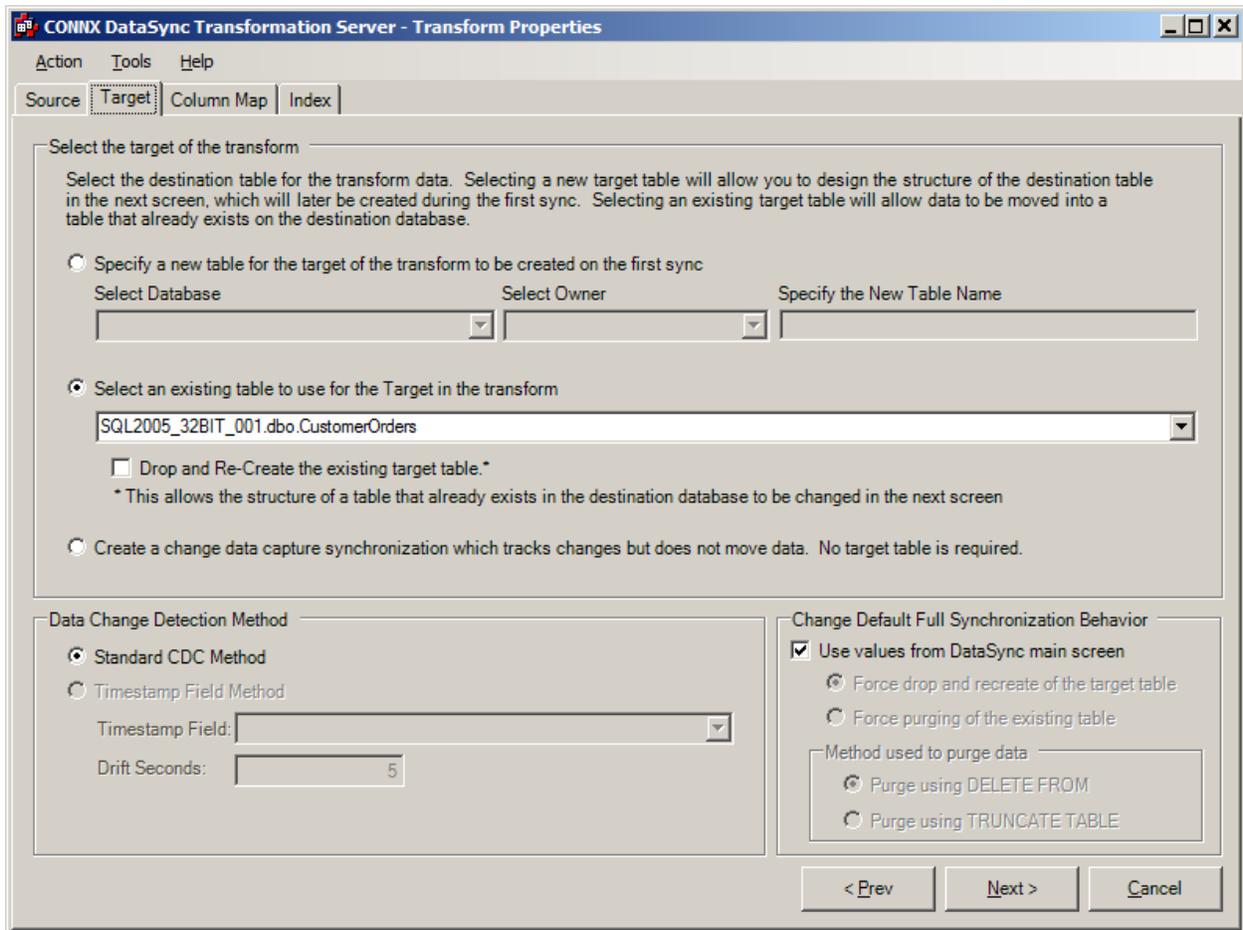
- After filling in the appropriate fields: UNIQUE must be checked in order to do incremental sync's, the DONE button should be pressed.

After the Index Mapping has been done, the main window will be displayed, and the transformation can be checked by running a [manual synchronization](#) on it.

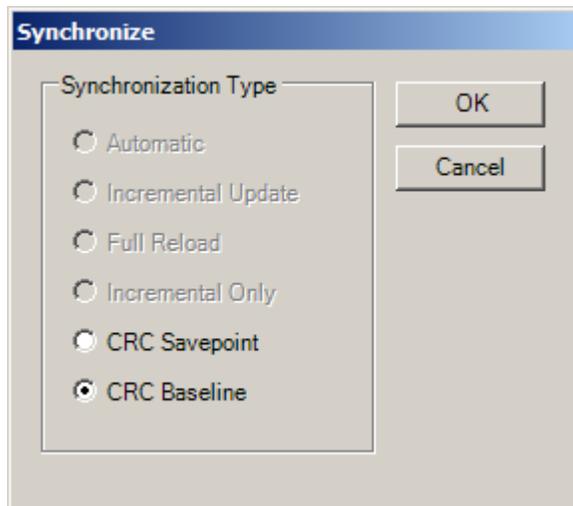
Change Data Capture transform

Creating and Syncing the Change Data Capture Transform

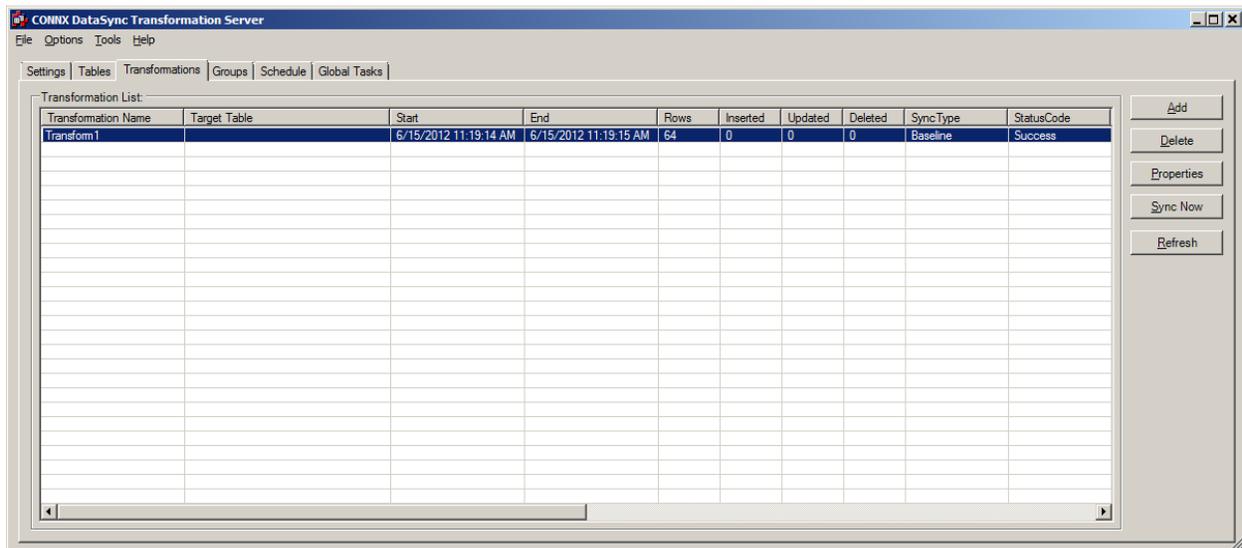
During the creation of the change data capture transform, select the source as before. Then on the Target Tab, check the "Create a change data capture..." button, and press "Next". The column mapping tab will be skipped because there is no target table, the transform is just tracking the changes on the source table. On the index tab, select or specify a unique index if there is one and press "Done".



After pressing done you are returned to datasync on the Transformations tab. Select the change data capture transform that was just created and press the "Sync Now" button. The Synchronize window will appear the first time a change data capture transformation is synced, it must create a baseline that future changes will be compared to. To do this select the "CRC Baseline" option, if there is no baseline and "CRC Savepoint" is checked the baseline will still be created. Press OK, the transform will be synced.



As we can see looking at the statistics for the baseline sync, 64 rows were synced, and since this is a baseline, there were no Inserts updates or deletes. In a separate application, in this case I used Infonaut to add, update and delete some rows in the source table. To see those changes, in DataSync with the same cdc transform selected, press the "Sync Now" button again.



This time in the Synchronize window, leave CRC Savepoint checked. That will tell DataSync to only get the changes from the last time a baseline was run. Press OK and the sync will run and return to the DataSync Transformations tab.

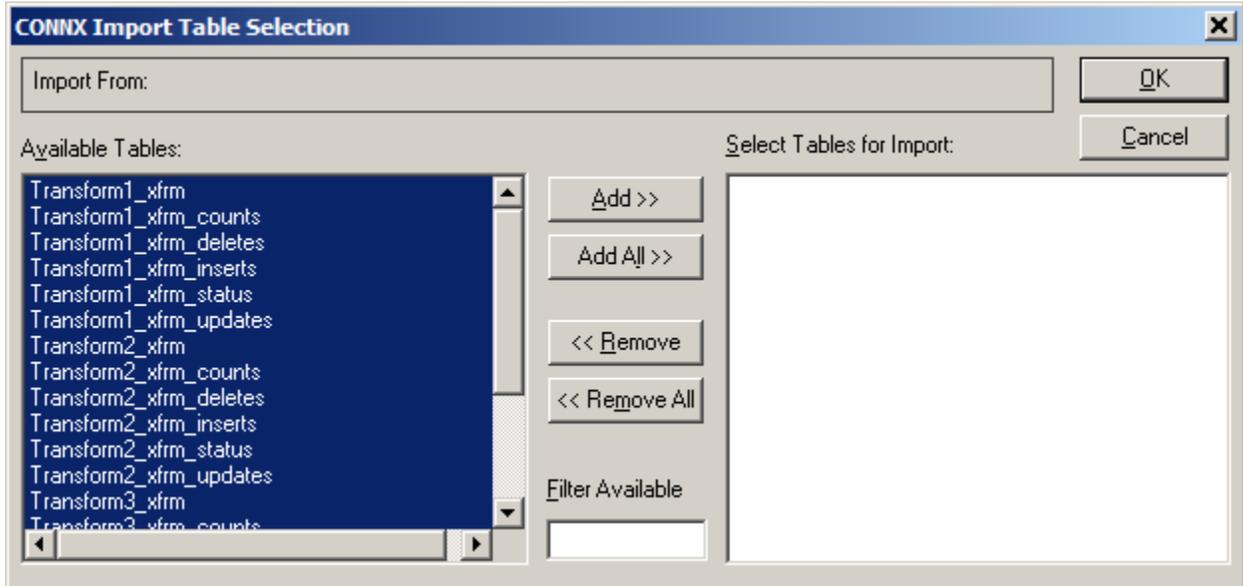
Using the change data capture synchronization in SQL Server Integration Services

1) Create a new Data Dictionary, different than the one used for DataSync, to use for the transformation data tables. Click the Start button and then point to All Programs. Point to CONNX Driver and then click CONNX Data Dictionary. The Open dialog box appears, click cancel to get to a new empty Data Dictionary.

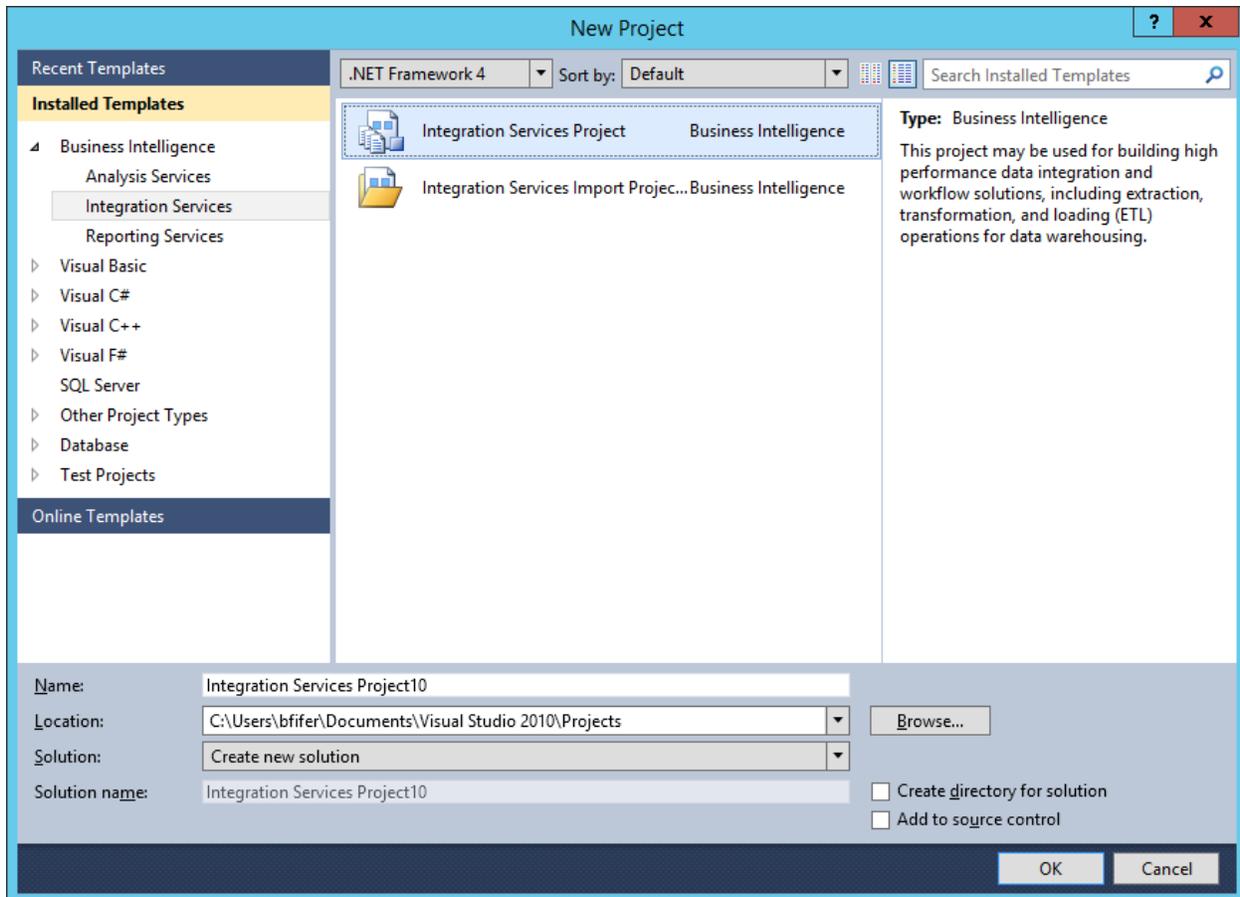
2) Import the new change data capture tables into the new Data Dictionary. Press the import button, then in the "Import Type" dropdown list, select "Change Data Capture Target". Then in the "Server" field put the name of the server that DataSync is running its synchronizations on. For the "UserName" and "Password" fields, enter the username and password that was used to connect to the DataSync Data Dictionary. For example if user1 and pwd1 were used to connect to the Data Dictionary opened by DataSync, transformations were created, then user1 and pwd1 would be entered on this screen.

3) In the Import Table Selection screen, press the "Add All" button to get all the change data capture tables. These will be used later by the Sql Server Integration Services Work packages to get the rows that have been inserted, updated or deleted. Press "OK" and the tables are added to the Data Dictionary.

Save the Data Dictionary and close the Data Dictionary Manager tool. It can be in the same folder as the DataSync Data Dictionary, but it needs to have a different name.

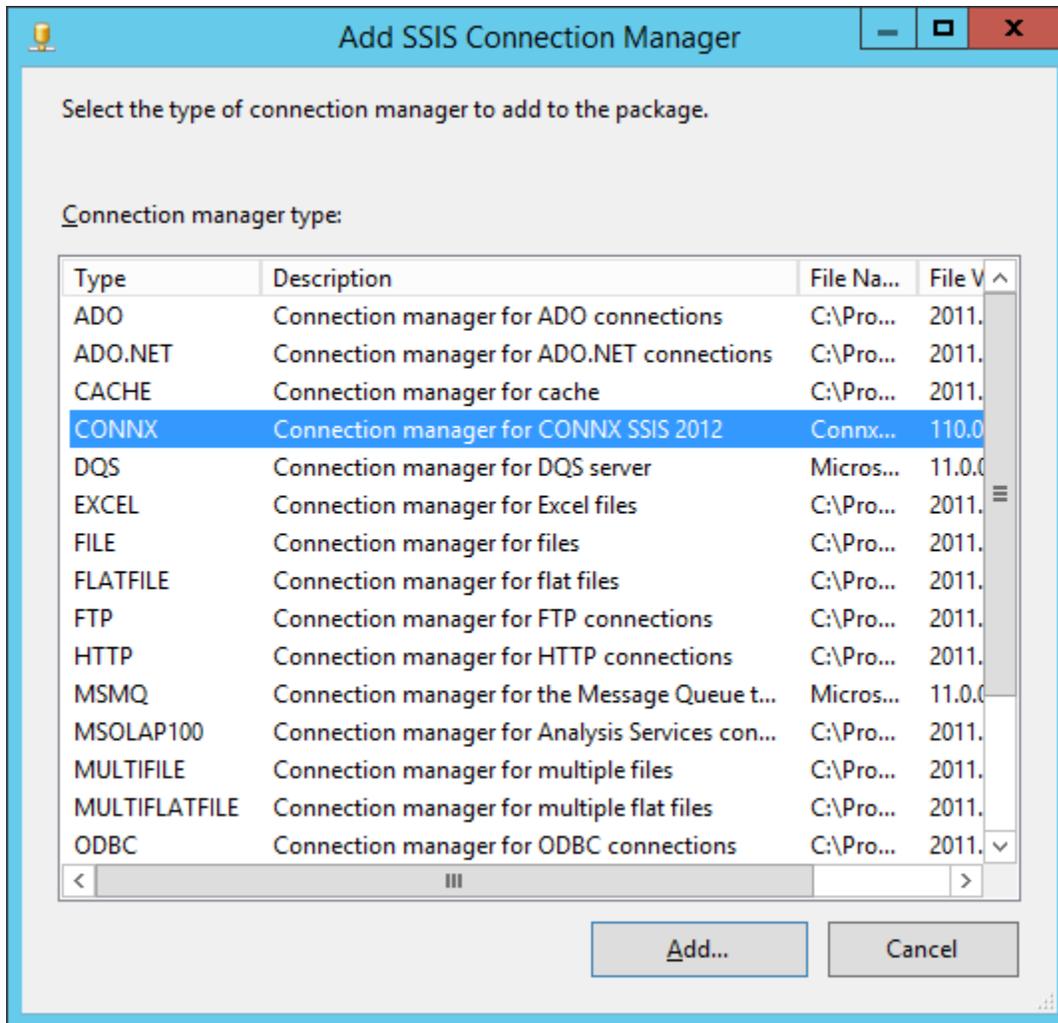


4) Now the SQL Server Integration Services package can be built. This walkthrough will be with SQL Server 2012 Integration Services, however SQL Server 2005, 2008 and 2014 Integration Services works the same way. Click the Start button and then point to All Programs. Point to Microsoft SQL Server 2012, then click SQL Server Business Intelligence Development Studio. Click File --> New --> Project to open the New Project form. Select the "Business Intelligence Project" project type and then "Integration Services Project" enter a name and press the OK button.

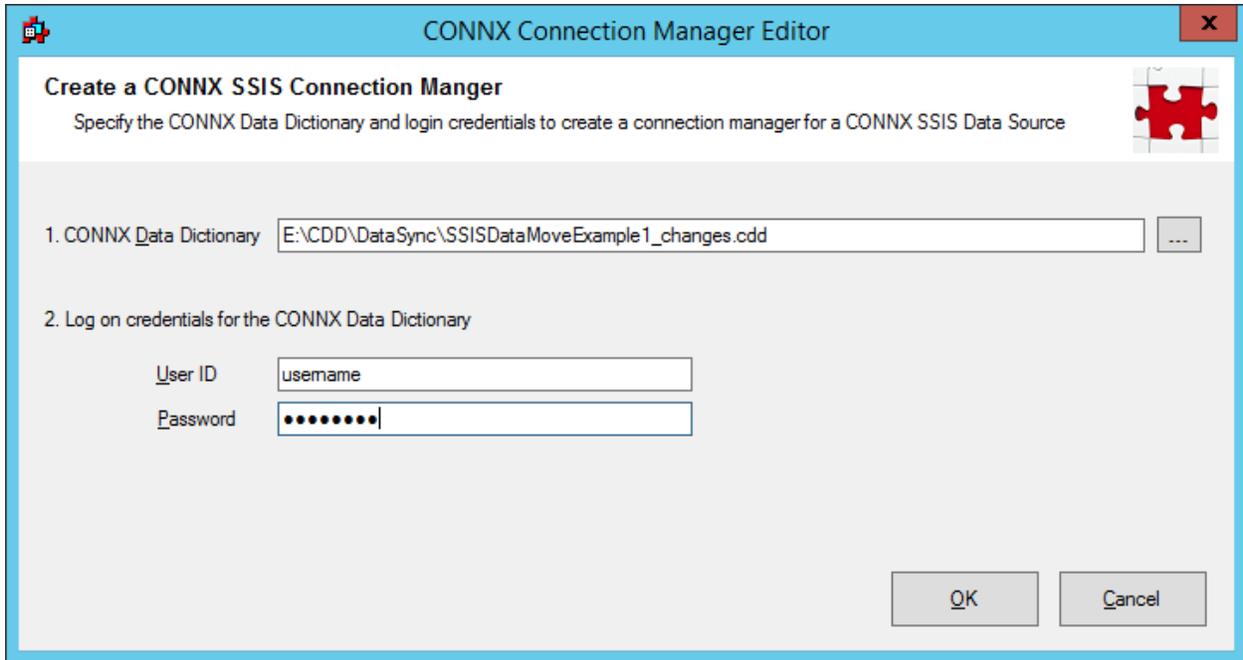


5) This should open an empty Package.dtsx environment, and the selected tab should be "Control Flow". The first thing that must be done is to add a connection to the transform table CONNX Data Dictionary. To add a connection, right click in the "Connection Managers" tab in the lower part of the "Control Flow" tab then select "New Connection.." from the pop up menu.

6) Select "CONNX" from the list of Connection manager Types and press the "Add..." button.



7) Enter the name and location of the CONNX Data Dictionary that contains the transform tables. Also enter the username and password used to connect to that transform Data Dictionary. Press the OK button and it will connect to the data dictionary and verify the connection information.



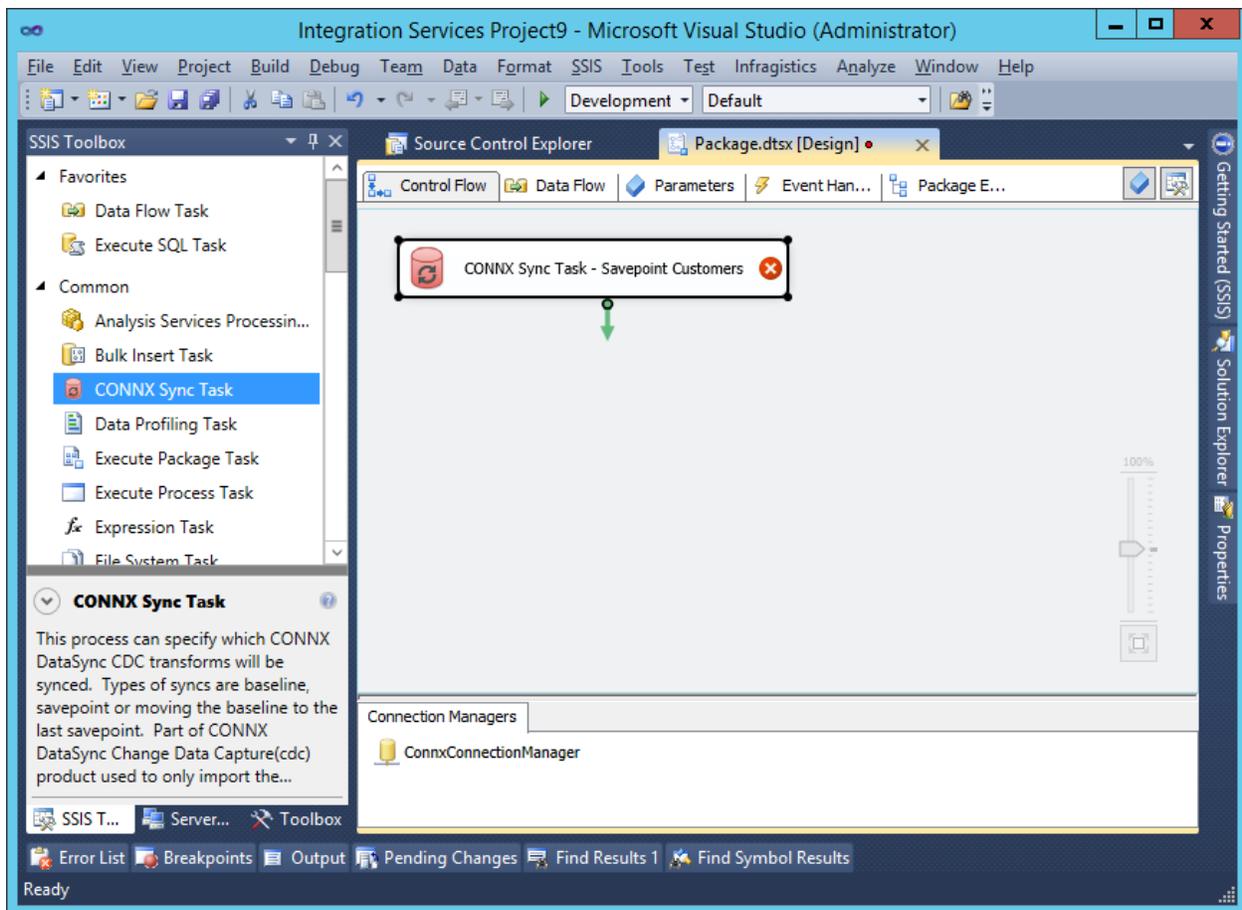
The screenshot shows a dialog box titled "CONNX Connection Manager Editor" with a close button (X) in the top right corner. The main heading is "Create a CONNX SSIS Connection Manger" (note the typo in the original image). Below the heading is the instruction: "Specify the CONNX Data Dictionary and login credentials to create a connection manager for a CONNX SSIS Data Source".

Step 1: "1. CONNX Data Dictionary" is followed by a text input field containing the path "E:\CDD\DataSync\SSISDataMoveExample1_changes.cdd" and a browse button ("...").

Step 2: "2. Log on credentials for the CONNX Data Dictionary" is followed by two input fields: "User ID" containing "username" and "Password" containing a masked password of ten dots.

At the bottom right, there are two buttons: "OK" and "Cancel".

8) Now that the connection is added in the "Connection Managers" tab, in the SSIS toolbox select drag the CONNX Sync Task to the design space. This will run the Savepoint Sync in the DataSync tool from the SSIS package. Rename it to - Savepoint Customers.



9) Double click on the CONNX Sync Task to configure it. If there is only one CONNX Connection manager, it will connect to that and list the Change Data Capture Transforms available. Select the transform to sync and then on the right select the type of Sync to run. Descriptions are:

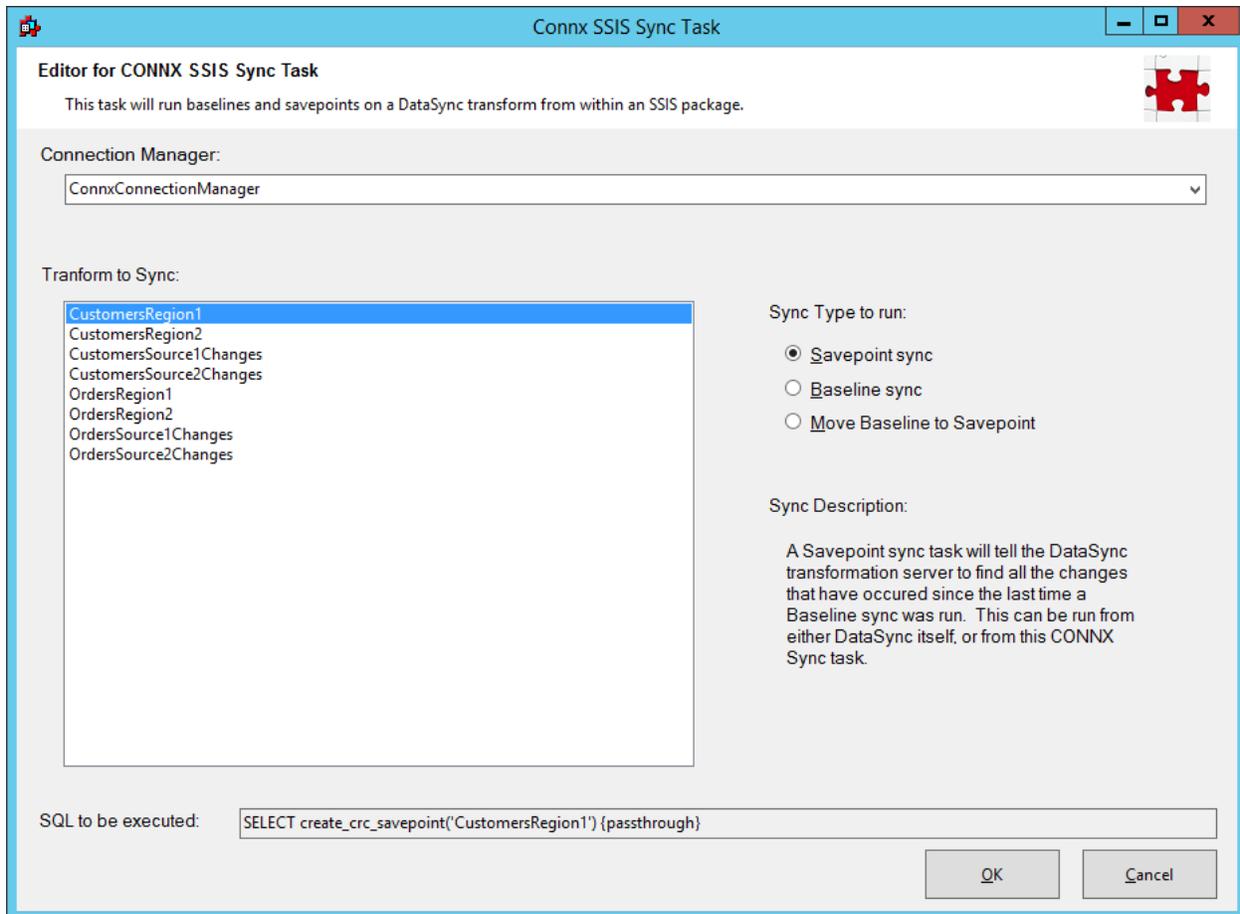
Savepoint sync: A Savepoint sync task will tell the DataSync transformation server to find all the changes that have occurred since the last time a Baseline sync was run. This can be run from either DataSync itself, or from this CONNX Sync task.

Baseline sync: A Baseline sync task when run will tell the DataSync transformation server to read and save the data as it is. That way all future Savepoint syncs will use this as starting point to find changes in the data. Can be run from either DataSync, or this CONNX Sync task.

Move Baseline to Savepoint sync: Making the last Savepoint sync the new change baseline is the perfect way to reset the starting point for tracking changes after the previous changes have been migrated to the targets. It doesn't actually run a Baseline sync, it simply sets the last Savepoint sync to be the new baseline.

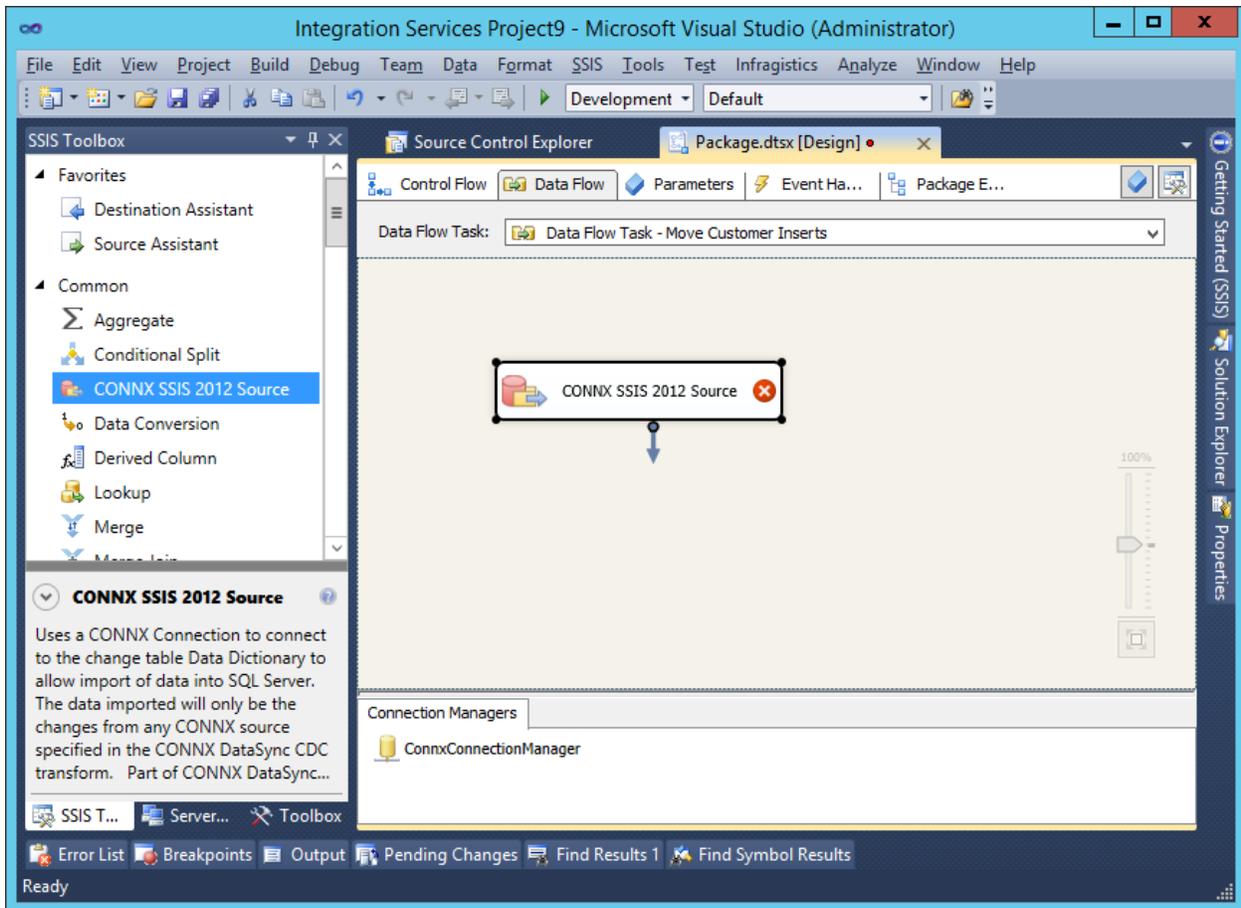
Press OK to close the window and save the changes.

* The SQL to be executed is for reference only, The sql string is created by selecting the transform and type of sync, it is displayed so the admin knows what SQL is actually being executed if they wish to. No user editing of this line directly is necessary.

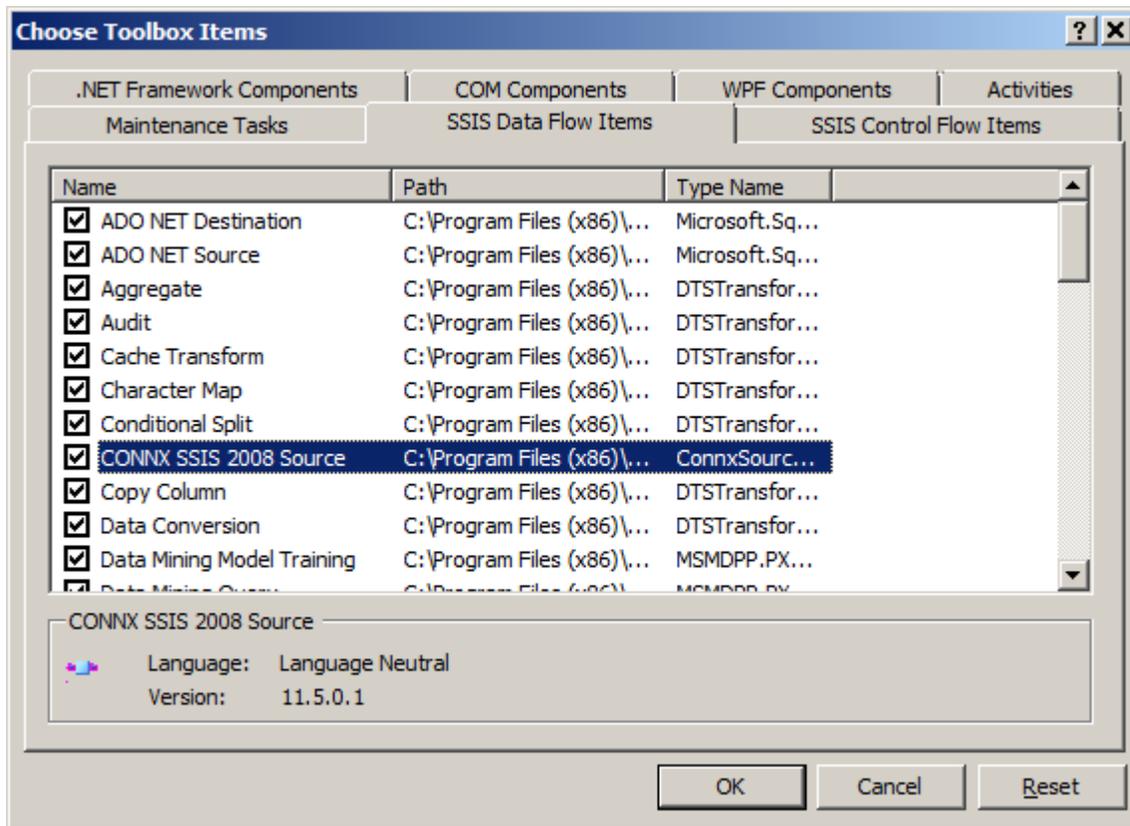


10) Now that the savepoint sync is run, we can get the data that has changed and move it into the target table. From the SSIS toolbox, drag a Data Flow Task to the designer and double click on it to bring up the Data Flow designer. Notice the SSIS toolbox has changed to only tools that work in a Data Flow Task. Drag a CONNX SSIS 2012 Source object to the design pane.

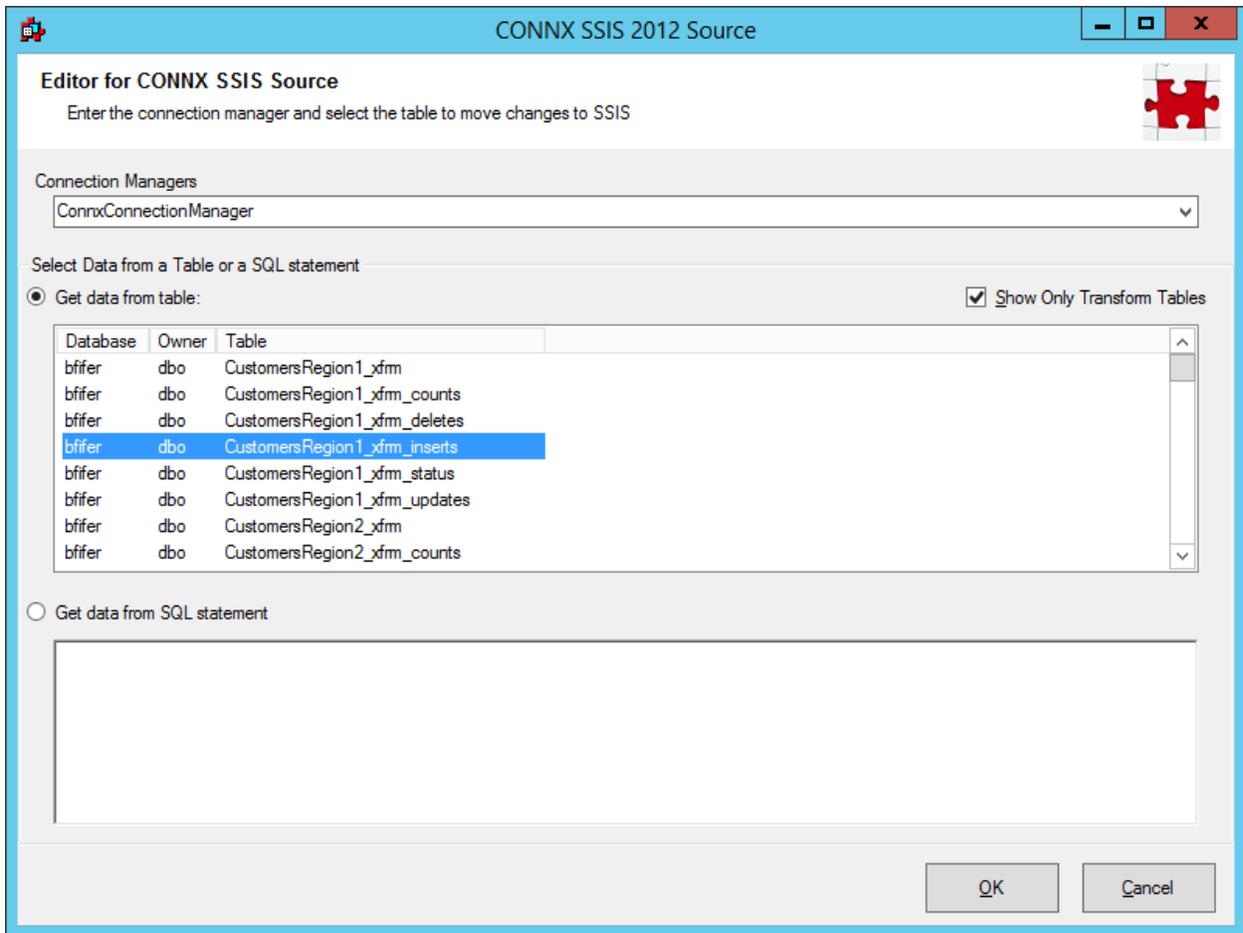
CONNX Data Synchronization 12.5



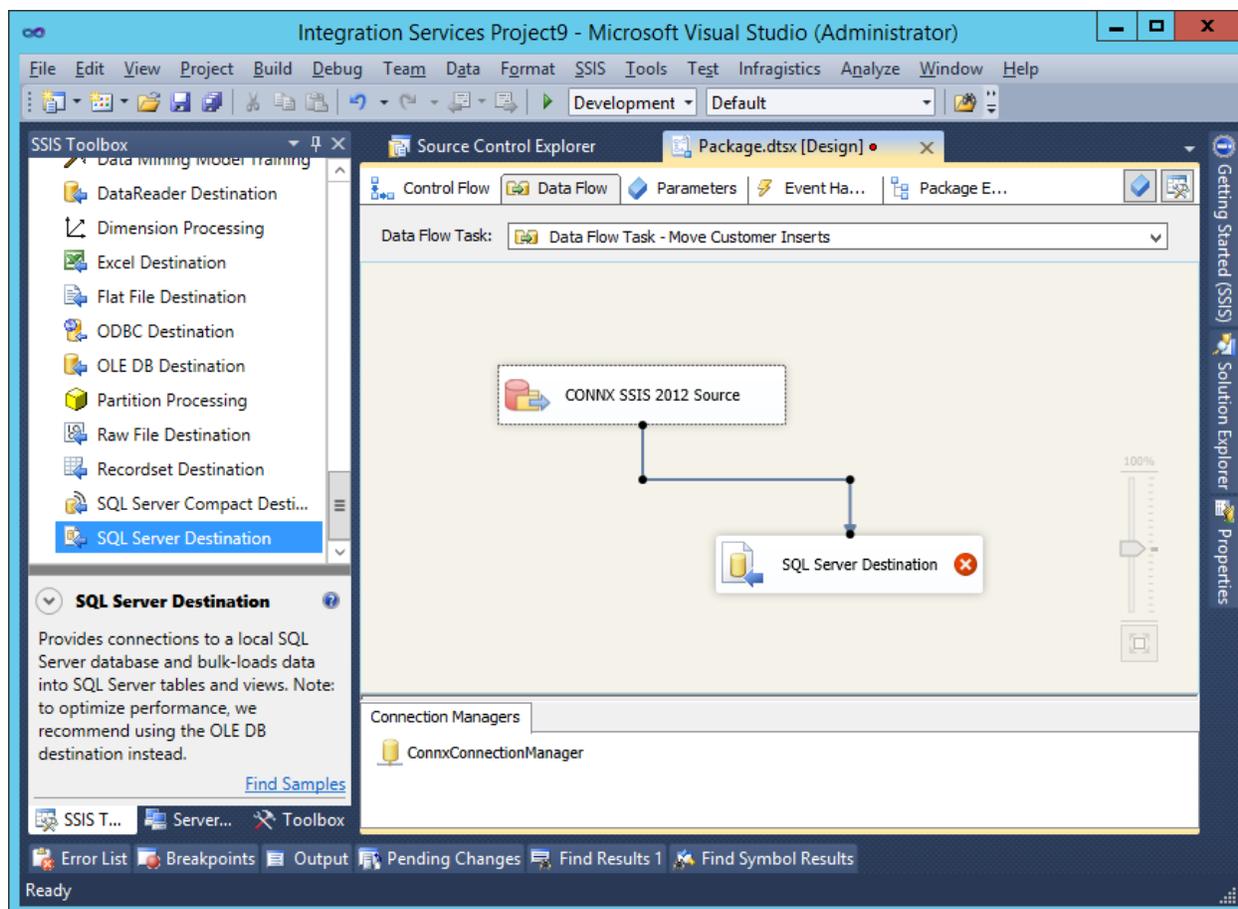
* Note: In the Business Development Studios for SQL Server 2012 and 2014, the toolbox will automatically updated with the CONNX SSIS components. However in Business Development Studio for SQL Server 2005 and 2008 you will need to add them to the toolbox. If there is no "CONNX SSIS 2008 Source" object in the toolbox, right click on the toolbox and select "Choose Items..." from the popup menu. The list of available tools will take several seconds to appear, when it does, choose "SSIS Data Flow Items" and put a check in the box next to "CONNX SSIS 2008 Source".



11) Double click on the CONNX SSIS 2012 Source box in the design window to configure the source data. If there is only one CONNX Connection manager then it will automatically connect to it and display the change data capture tables. The list of tables will have the name of the transform first, then be appended with what data is in the table. For this one we are going to use the `_xfrm_inserts` table to get all new rows that have been inserted into the table. Press the OK button to Save the new configuration and return to the Data Flow designer. Another option to specifying a table is to use a select SQL Statement using the change tables. This is there for customized data retrieval and is an option, however likely all the change data is needed and selecting a table will suffice.



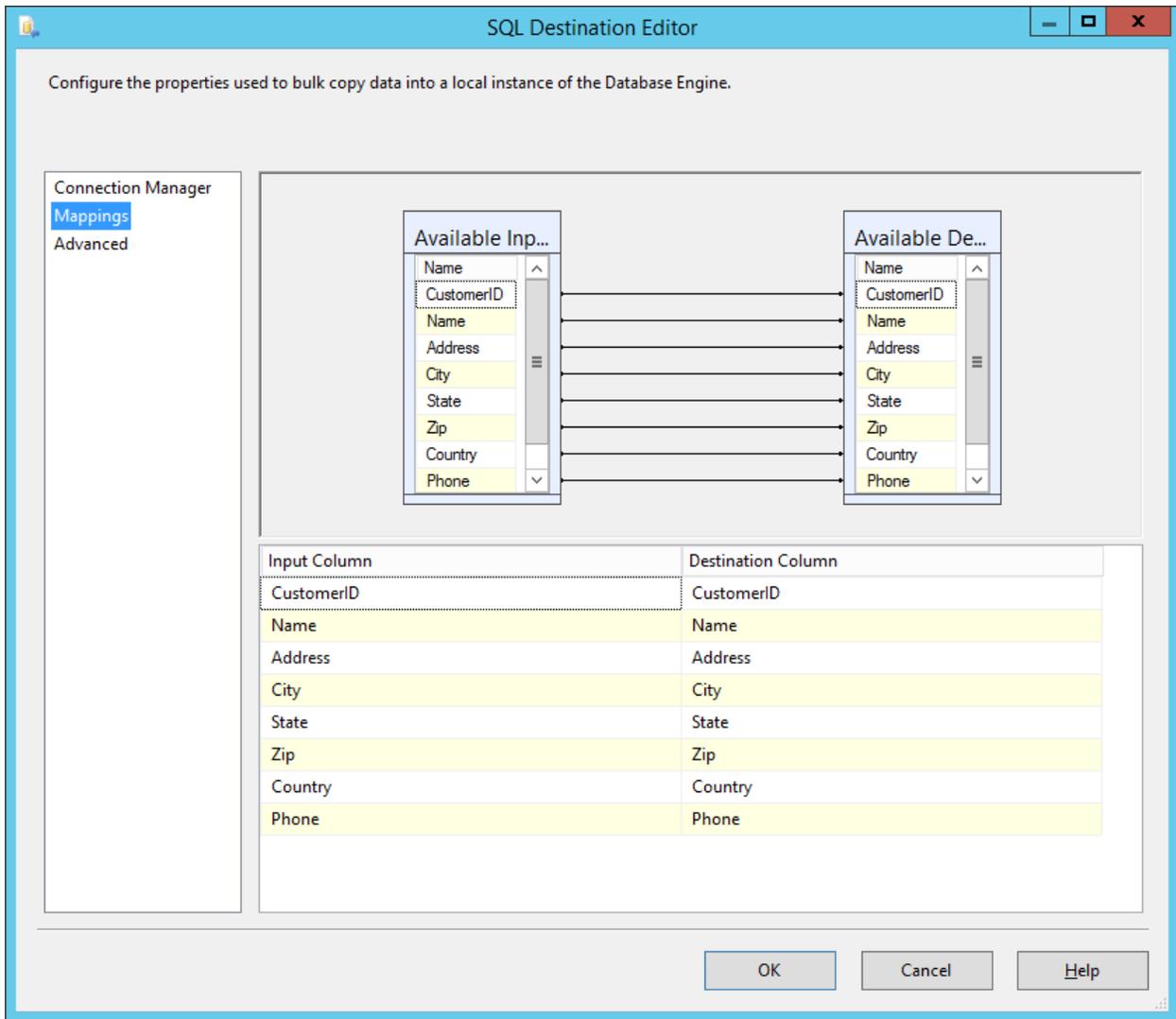
12) The Data Flow is now ready for a target to move the change data to. Go to the toolbox, in the "Data Flow Destinations" section and drag "SQL Server Destination" to the design pane and drop it. Then drag the green arrow from the "CONNX SSIS 2012 Source" object to the "SQL Server Destination" object and the designer will create a green line between the two objects.



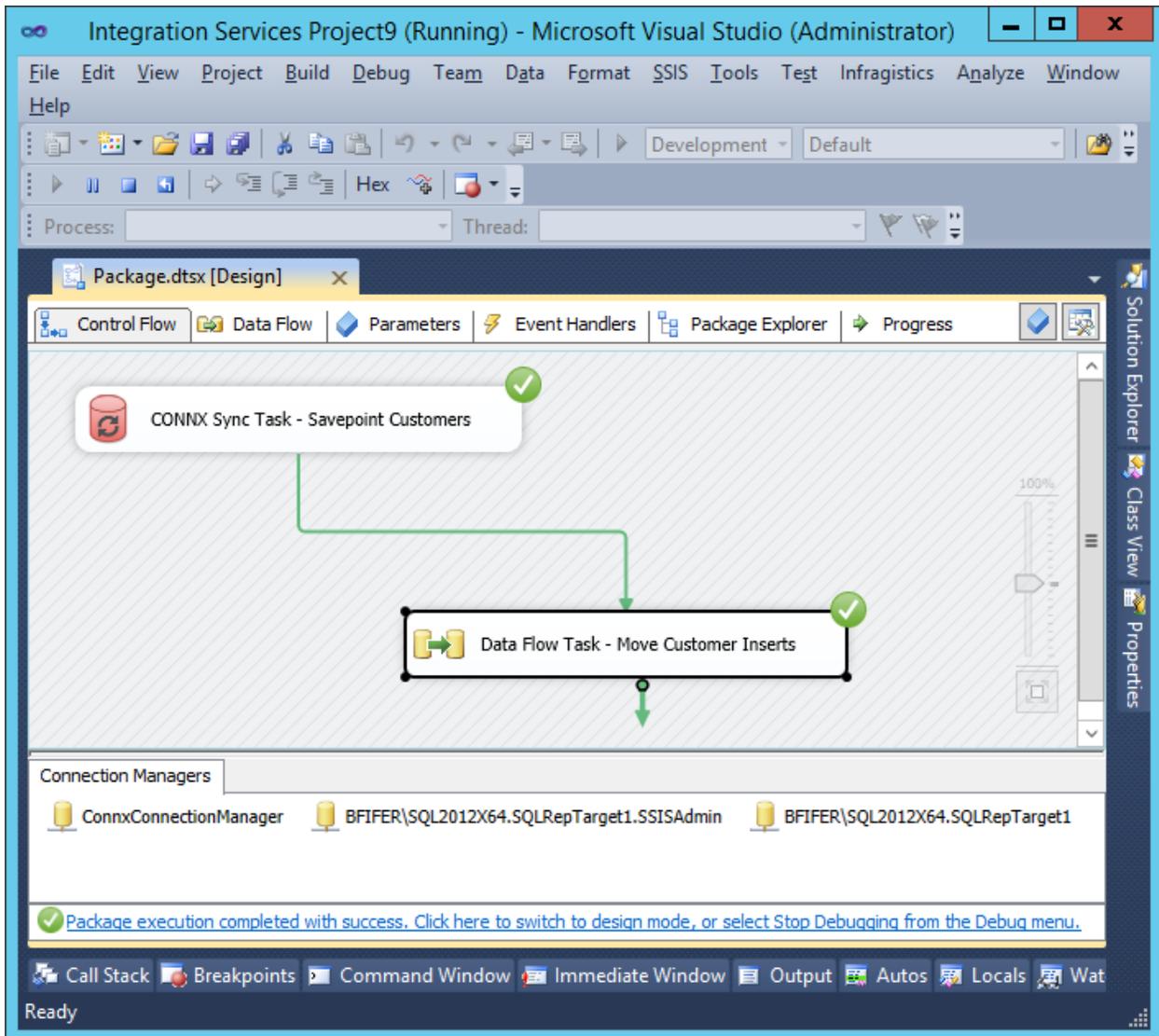
13) Now we need to edit where the change data is moved to, right click on the SQL Server Destination object and select "Edit". That will open the SQL Destination Editor on the Connection manager screen. Next to the "Connection Manager" dropdown list select the "New..." button. That will bring up the Connection Configure screen, if you have a SQL Server connection it will be displayed in the list. If not, Select the "New" button and input the connection information to the SQL Database you wish to be the target. Select "OK" to finish and the connection specified will now be in the "Connection Manager" dropdown.

14) Select a table to move the data to from the "Use a table or view:" dropdown menu. If this is going to a new table, press the "New..." button and specify the name in the create table statement then hit "OK" and it will be selected in the dropdown menu.

15) Select "Mappings" in the list on the left, the default mappings will be displayed. Change if necessary and when finished press "OK" to return to the Data Flow designer.

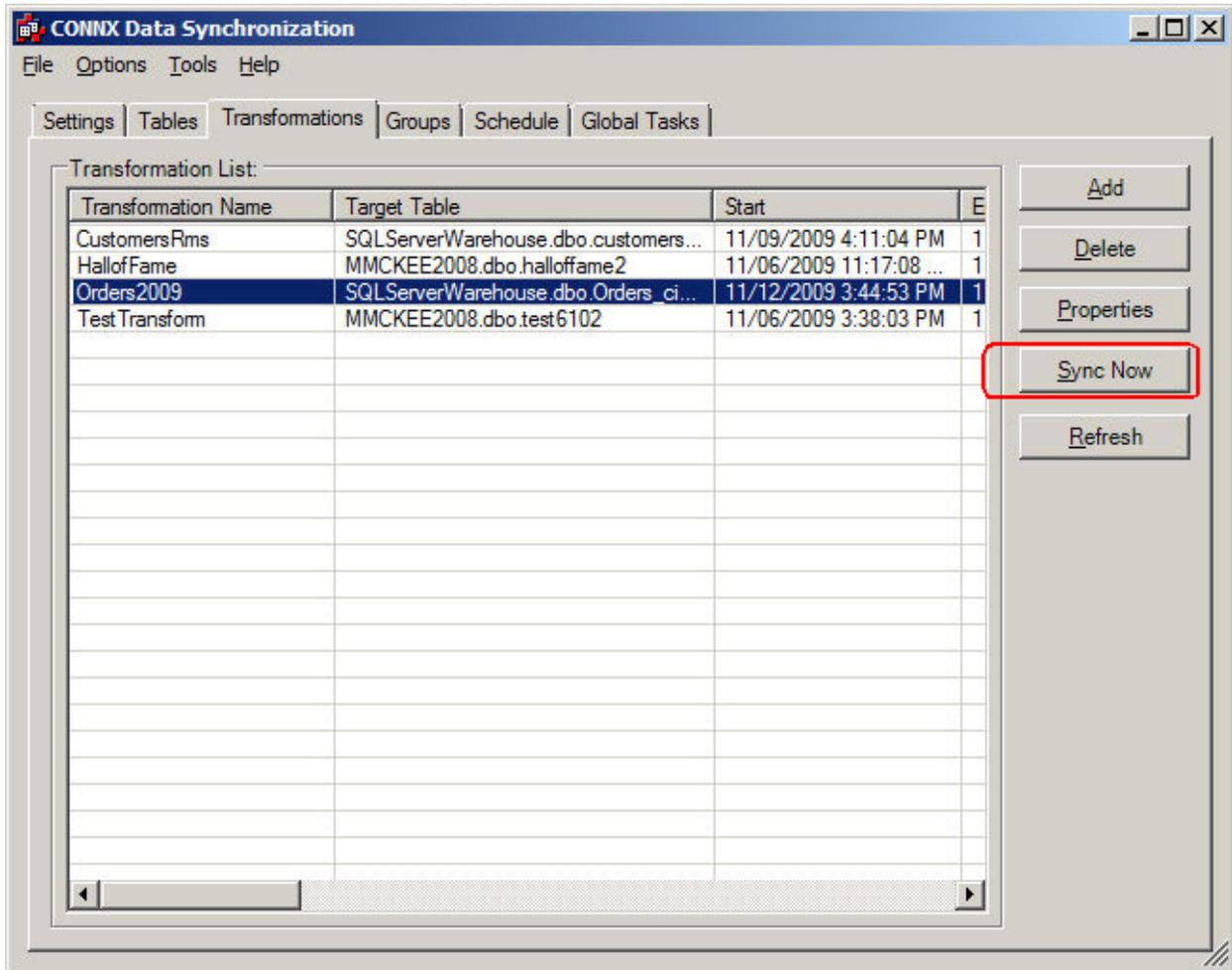


16) The package is now ready to move change data. The package can be run by pressing F5, green indicates a successful run, red indicates an error. Note, if there has not been any changes on the source data the sync will be green for success, however no entries will be moved because there were no changes.



Performing a Manual Synchronization

When the Transform has been saved, it will show up in the list of Transformations as a saved Transform. It can then be manually synchronized. Select the Sync Now button, and the synchronization should occur.



To validate the transformation, you can use Infonaut to display the contents of the new transformed table. The name of the table is SQLServerWarehouse.dbo.Orders_cisam.

Invoke Infonaut by selecting TOOLS/Launch Infonaut. After connecting to the CDD, you can list the tables in the CDD, and select SQLServerWarehouse.dbo.Orders_cisam. The query appears as:

InfoNaut Professional - C:\mlb\CONNXDataSync.cdd

File View Connection Tools CONNX Help

Execute Show Tables Build Query Report Refresh CDD QDBC Trace Get Query Schema Max Rows

Data Execution Plan Uniform SQL Relational DB SQL

Transaction: Begin Commit Rollback

Recent Connections: DD=C:\mlb\CONNXDataSync.cdd;Prompt=2;Persist Security Info=True;APPLICATION=ONDEMAND;

Recent Queries: SELECT * FROM SQLServerWarehouse.dbo.Orders_cisam

SELECT * FROM SQLServerWarehouse.dbo.Orders_cisam

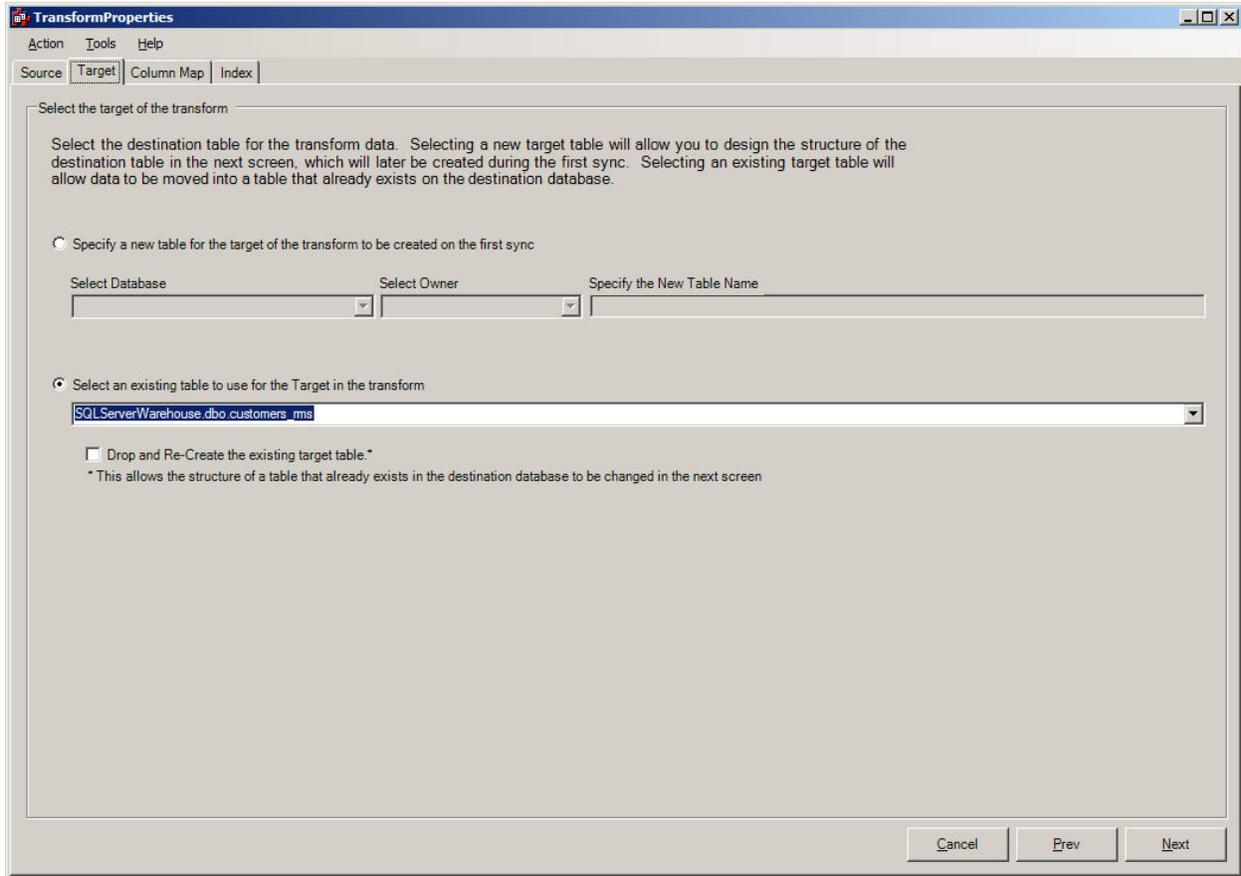
Drag a column header here to group by that column.

ORDERID	CUSTOMERID	PRODUCTID	ORDERDATE	PRODUCTQUANTITY	cnxrownumber	ADJUSTEDPRODUCTQUANTITY	
1	1	MERRG	25	05/15/1995 12:00...	30	1	31
2	1	MERRG	40	05/15/1995 12:00...	40	2	41
3	1	MERRG	59	05/15/1995 12:00...	8	3	9
4	1	MERRG	64	05/15/1995 12:00...	44	4	45
5	2	FOODI	31	05/16/1995 12:00...	35	5	36
6	3	TOMMI	41	06/20/1995 12:00...	10	6	11
7	3	TOMMI	71	06/20/1995 12:00...	3	7	4
8	4	RATTC	16	06/23/1995 12:00...	10	8	11
9	4	RATTC	43	06/21/1995 12:00...	10	9	11
10	4	RATTC	56	06/21/1995 12:00...	60	10	61
11	5	SAWYH	2	07/28/1995 12:00...	12	11	13
12	5	SAWYH	27	07/28/1995 12:00...	2	12	3
13	6	MORNS	11	07/31/1995 12:00...	4	13	5
14	6	MORNS	19	07/31/1995 12:00...	20	14	21
15	6	MORNS	52	07/31/1995 12:00...	5	15	6

Record: 1 of 250 Records/Sec : 2770 Time : 0.090261

Editing a Transformation

When you are at the main screen for Transforms, a Transform can be edited by selecting Properties. The steps to edit a transform are similar to creating a transform with one notable exception. If the Transform has been successfully synced, a physical file exists, and the default is to not permit major alterations to the Target Table. This means that any Transformation that has the potential for physical alteration cannot be done, unless "Drop and Recreate existing Target Table" is checked. The screen shot below demonstrates this:

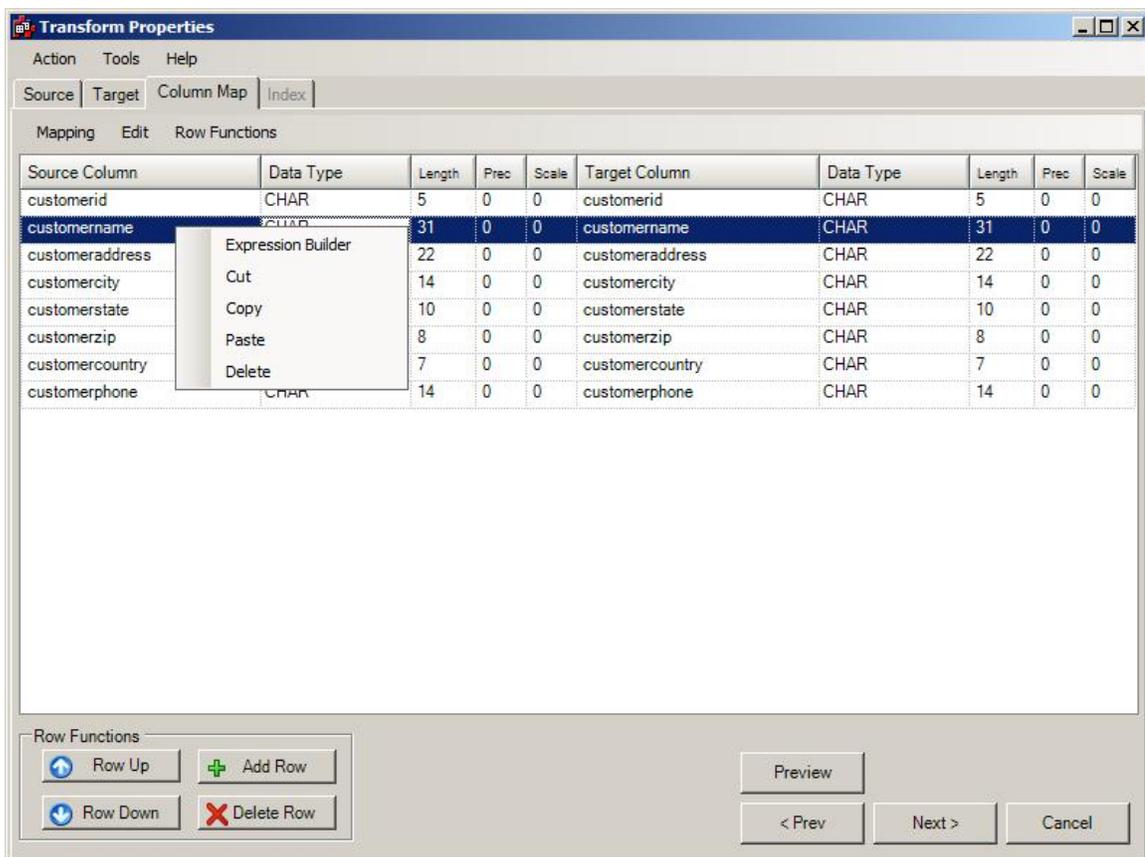


Using the Expression Builder

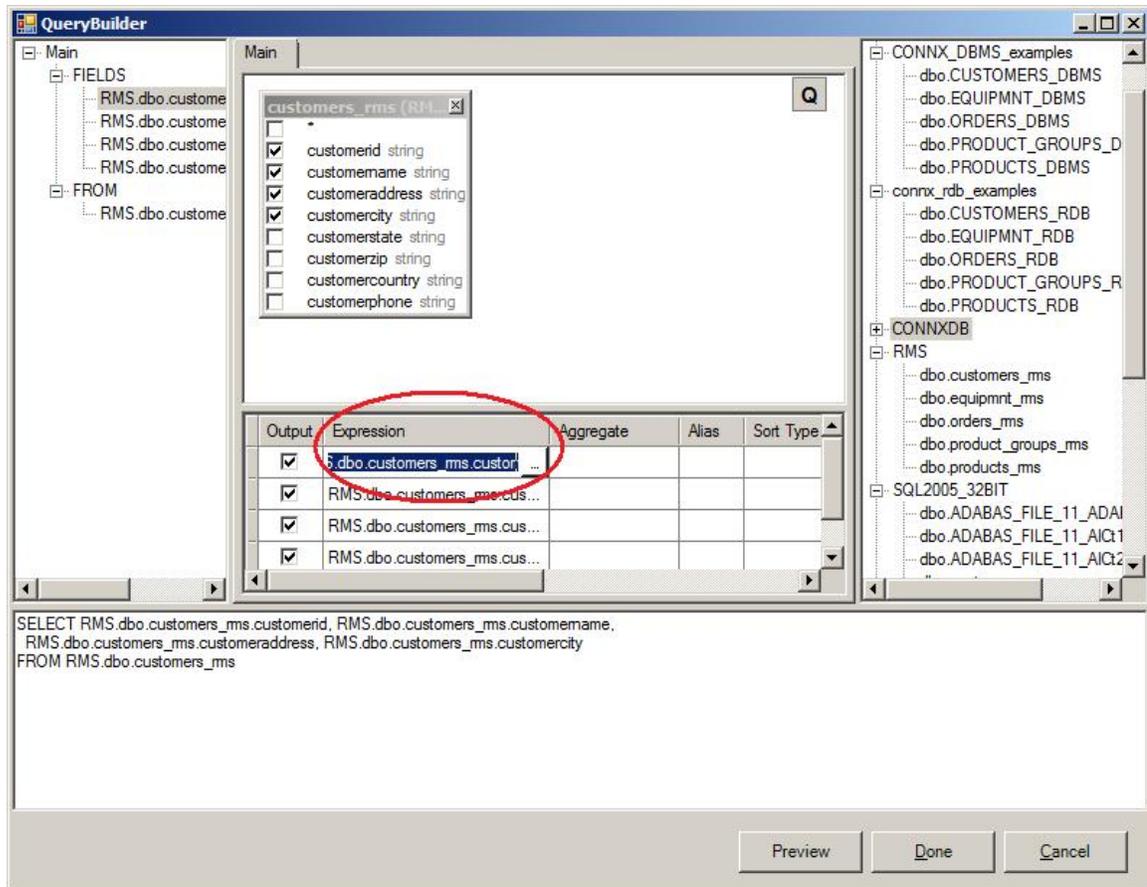
How to enter:

The Expression builder can be opened from either the **Query builder** or the **Column Mapping tab** in the **Transform Properties** wizard. From the **Column Mapping** tab, double click on the **Source column** that you wish to build into an expression. Another way to enter the Expression builder is to right click on the **Column map table** and select **Expression Builder** from the context menu. This will open the **Expression builder** and put the selected column name into the Expression field. From the **Query builder**, either right click on the **Expression field** and select **Expression Builder** from the menu. Another way is to select the box with the "..." at the right side of the **expression cell** in the **query builder**.

1. Enter the Expression builder from the Column Map



2. Enter the Expression builder from the Query Builder.



Why to use:

The Expression builder is designed to help the user create complex expressions easily. It lists functions supported by **CONNX**, gives information about the functions and the arguments it contains. This can be a helpful reminder of standard SQL functions and a good way to learn about **CONNX** specific functions.

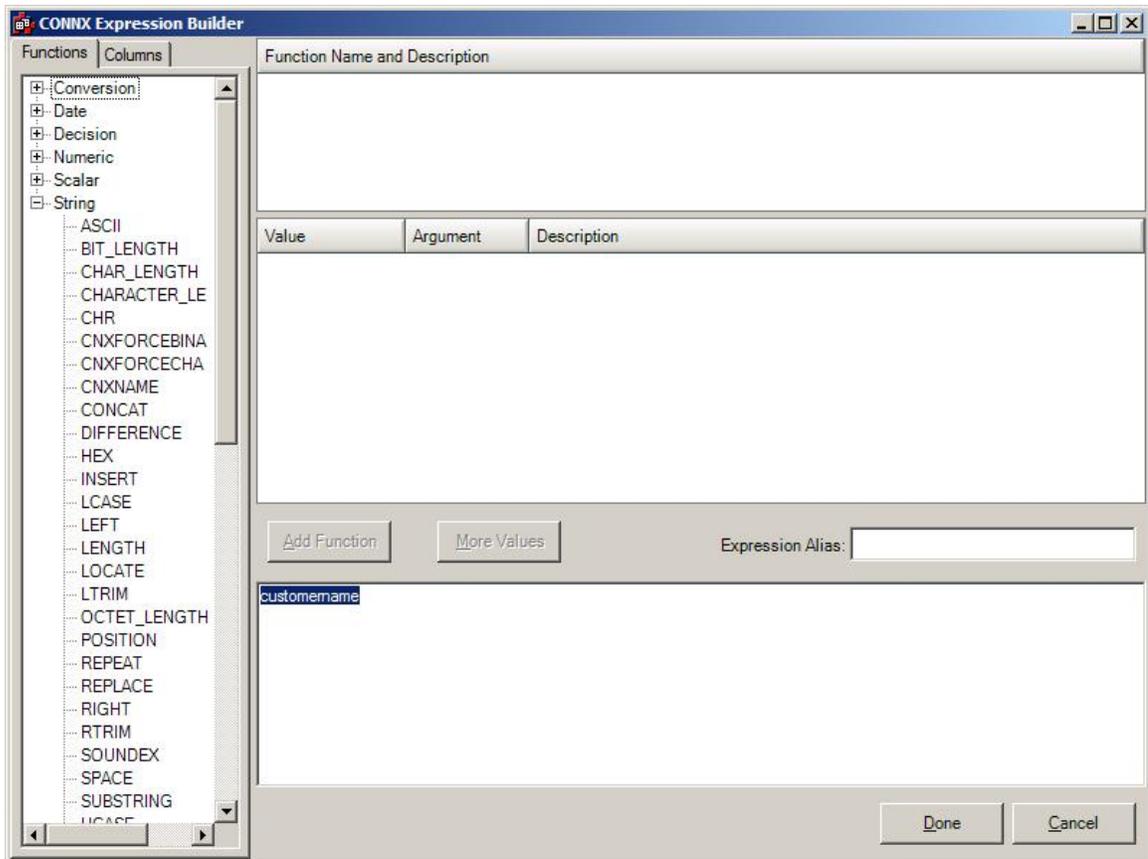
The Layout:

The **Expression builder** is composed of the following parts: **Functions available** list, **Columns available** list, **Function Name and Description**, **Function Argument Description** table, **Expression Alias** input field, and **Expression SQL** field. All functions supported by CONNX are listed in groups in the left panel, the groups can be expanded or collapsed to see all functions available. When a function is selected the name and the description of the function are shown in the table on the top right. Also after a function is selected the list of function arguments and their descriptions can be found in the **Argument Description** table. The **Expression field** at the lower right can either be edited directly or it will show the built Expression. The **Expression Alias** entry field is for creating a simple name for the expression that can be used for the Target column name when mapping.

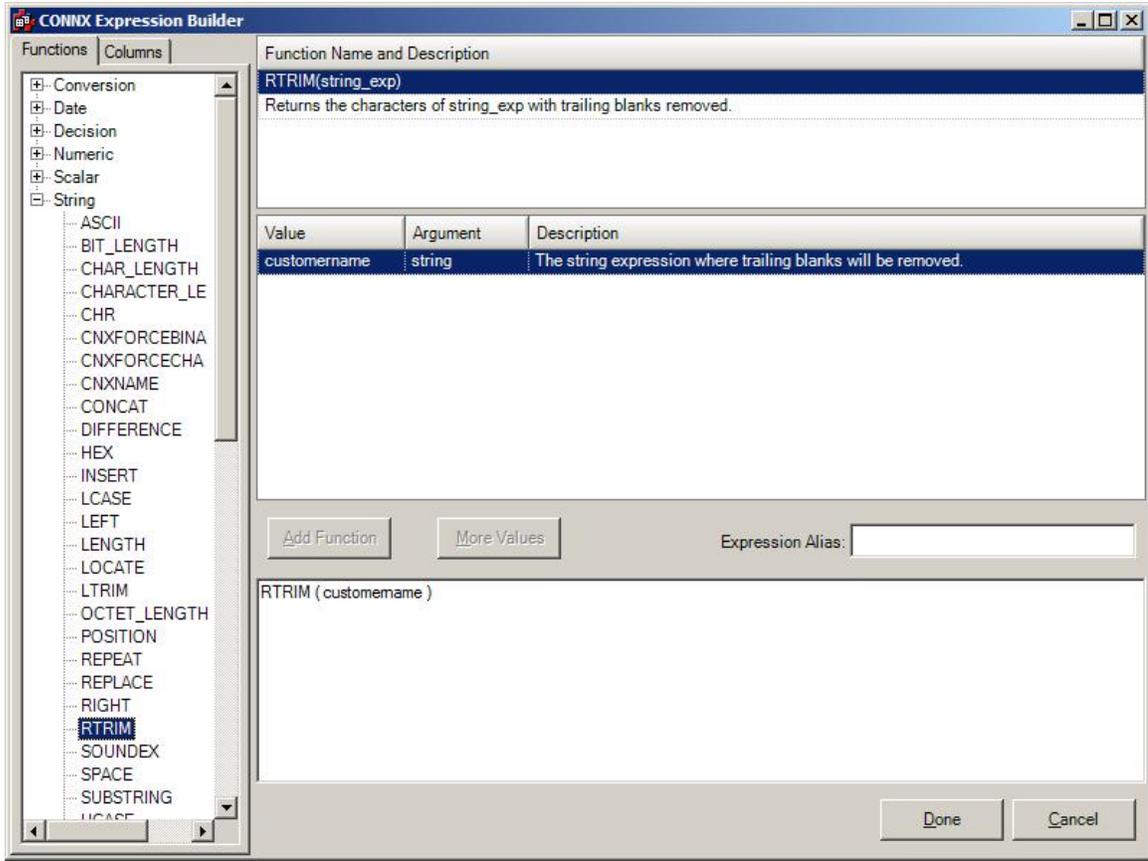
Example:

The following example goes step-by-step through an example of building a transform

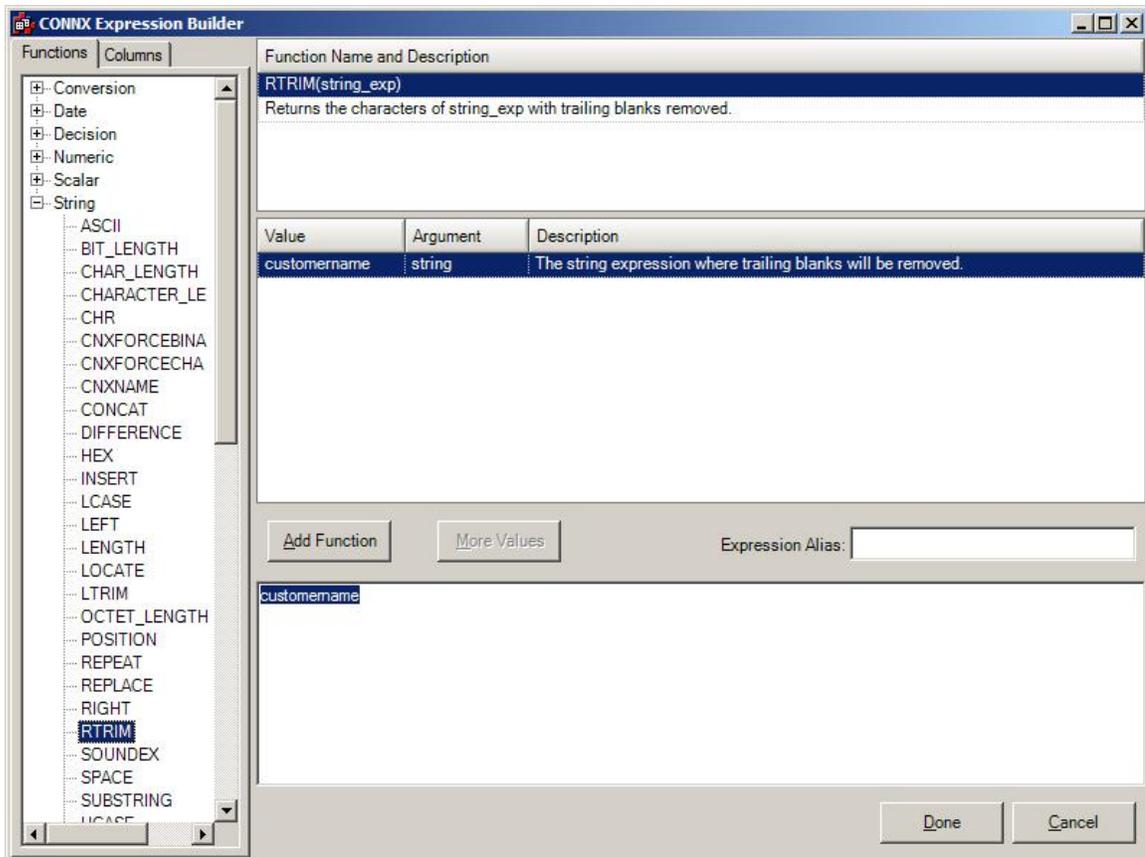
1. Upon entry to the **Expression Builder** the contents of the column selected will appear in the **Expression SQL** field and be highlighted.



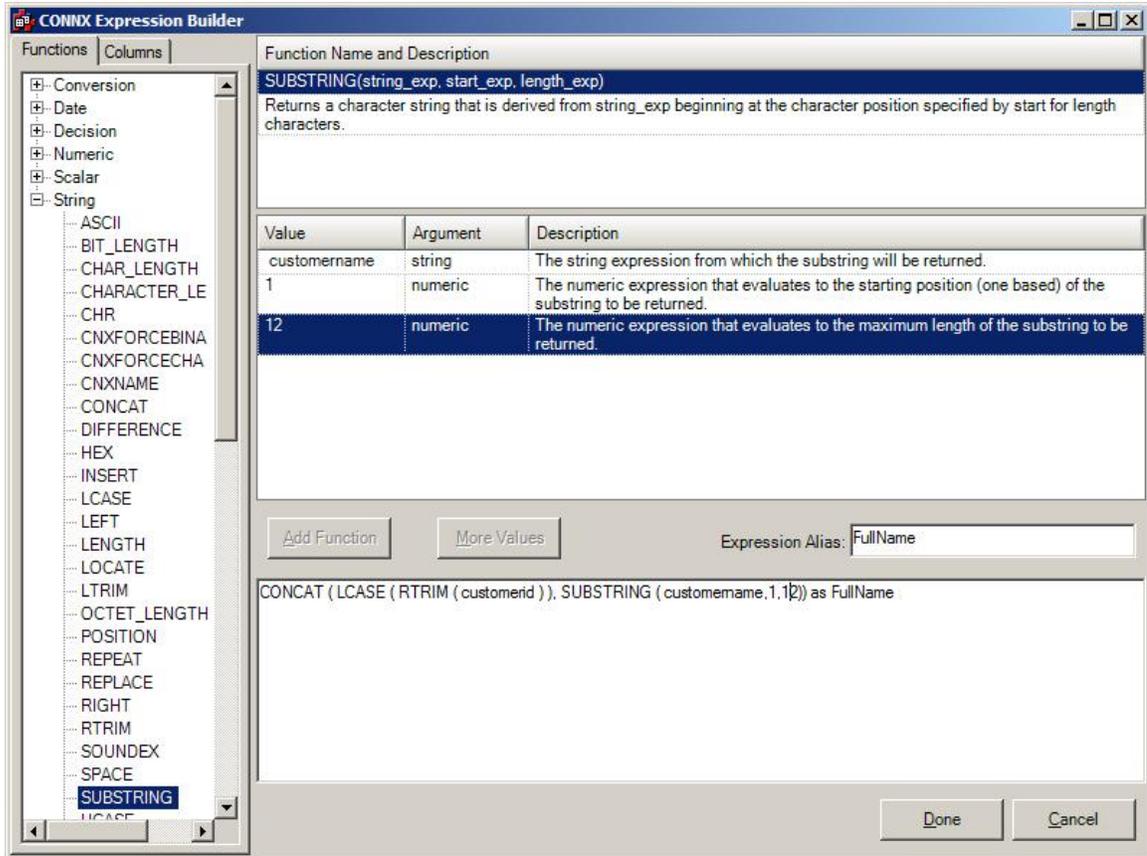
2. Browse through the **function list** expanding or collapsing groups to find the appropriate function. Select functions that may be of interest to see their description to aid selection. Once the appropriate function is found double click or select the **Add Function** button and it will appear in the **Expression SQL** field.



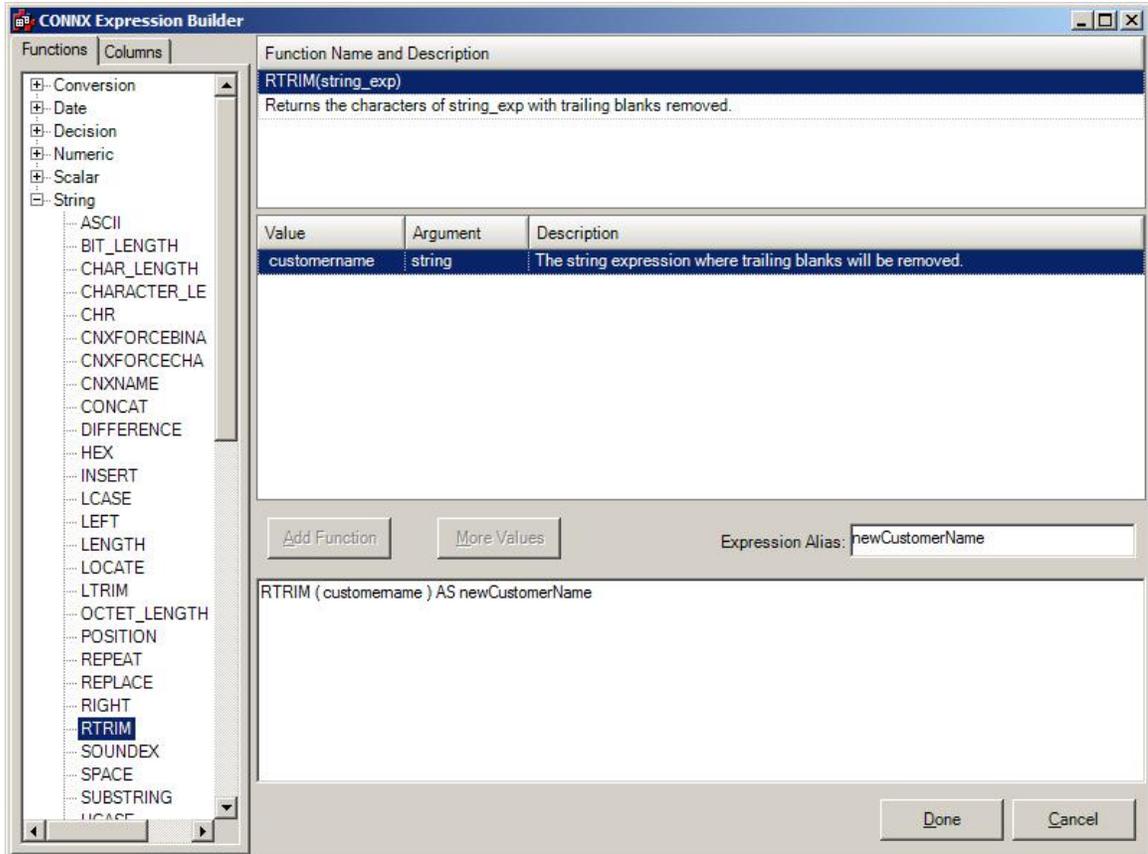
3. When the function appears in the **Expression SQL** field the highlighted text will then appear as the first argument of the expression. Other expression arguments can be filled out by either direct editing in the **Expression SQL** field, or by editing the value cell in the **Expression Arguments** table.
4. The **Expression Arguments** table can be used to enter the values for the expression. The first field can be edited and used to put values into the function. The second column in the table tells the datatype of argument the function is expecting. The third column has a description of what the purpose of the argument is in the function. In the for each argument in the table edit the Value field and enter the appropriate column name, characters or numbers for the function argument. Note, as values are entered here they also appear in the **Expression SQL** field.



5. Another way to enter the values for the function arguments is to edit them directly in the **Expression SQL** field. Any changes here will be reflected in the **Expression Arguments** table.
6. Add another function to the expression by placing the cursor to the desired location of the new function in the **Expression SQL** field and double clicking the function in the list. Dragging the cursor to make a selection in the **Expression SQL** field before double clicking the function will cause the selected text to appear in the first argument of the function.

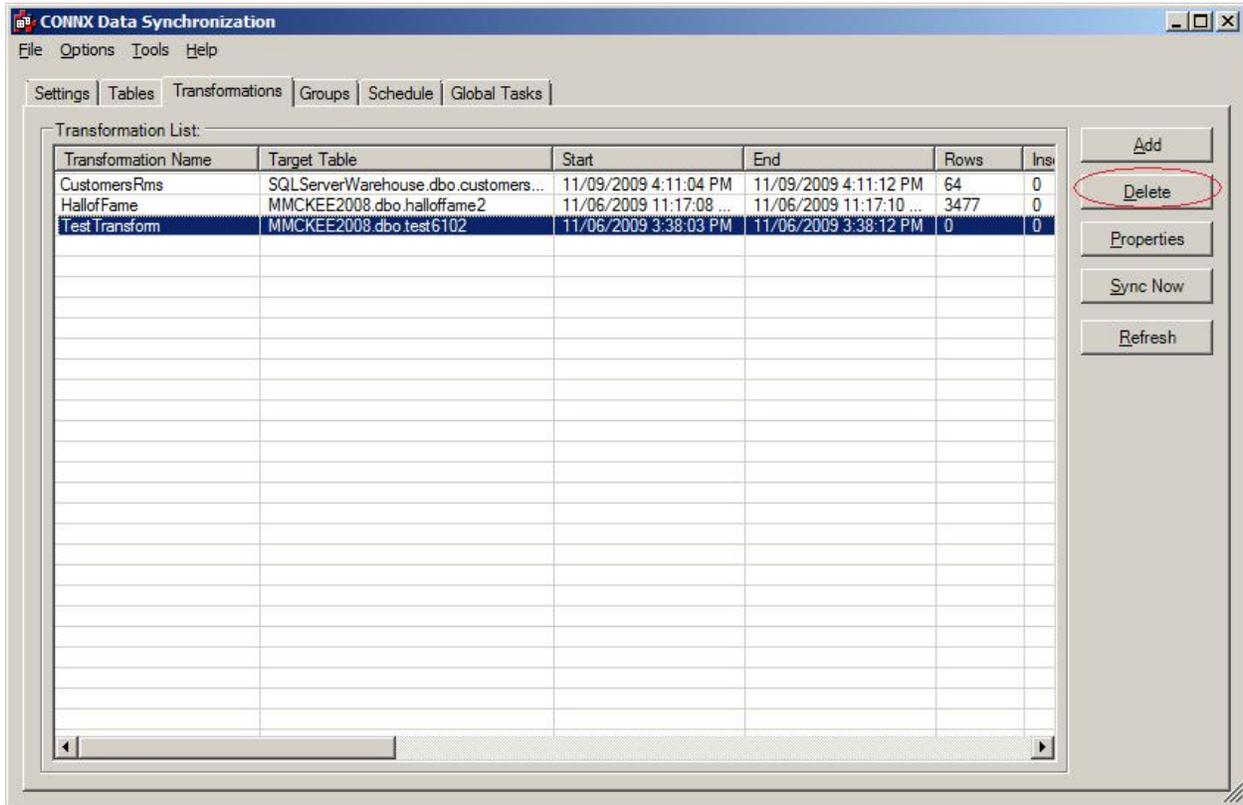


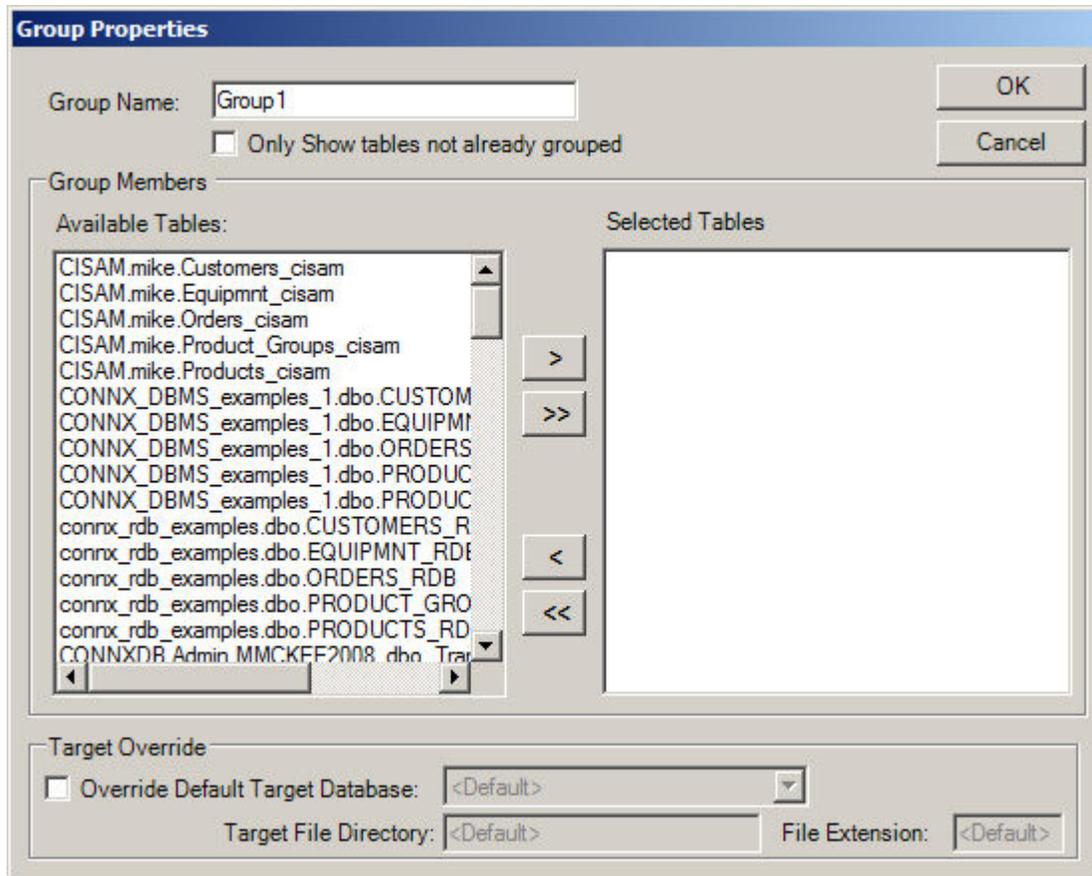
7. Add an alias to be used for the corresponding target table column name. In the Expression alias field type a simple and descriptive name for the expression. It will automatically be added to the end of the expression.



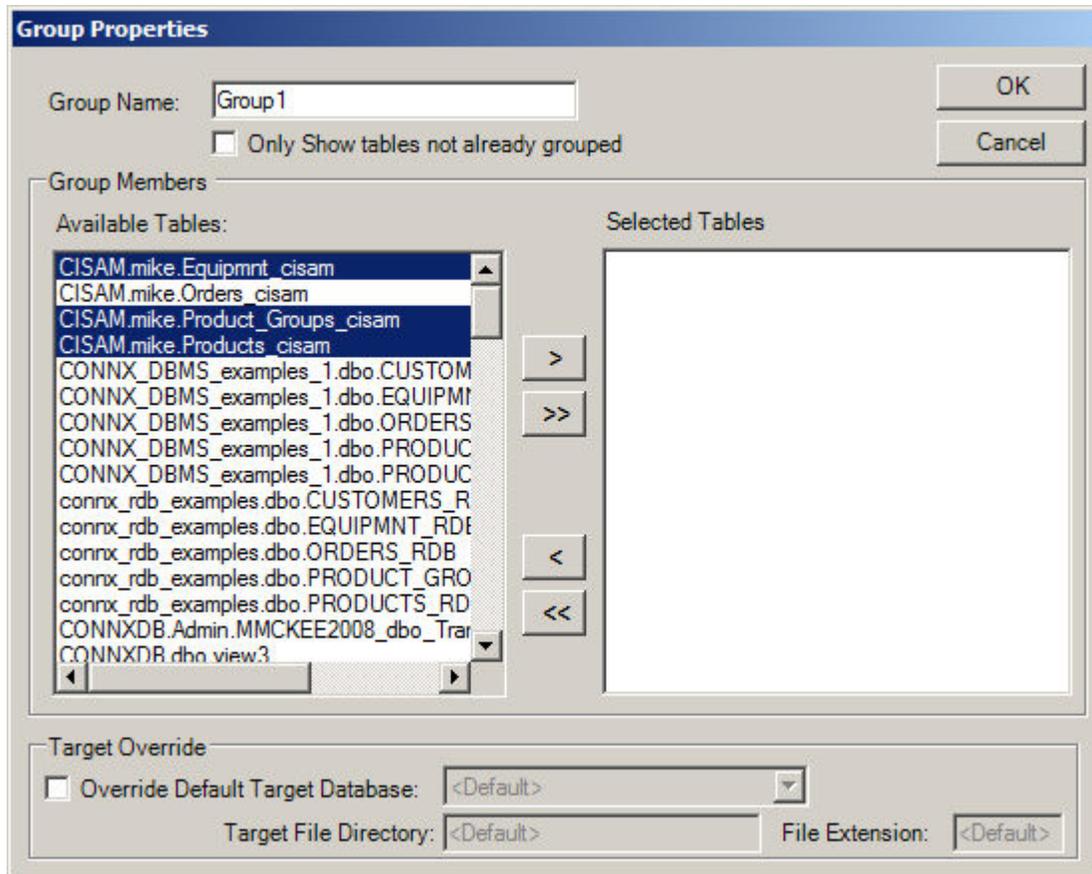
Deleting a Transform

Deleting a Transform simply deletes the Transform from the list of available transforms to sync. **It will not delete the Destination Target Table** (if it exists). The procedure to delete a transform requires navigating to the **Transformations** tab and pressing the **Delete** button.

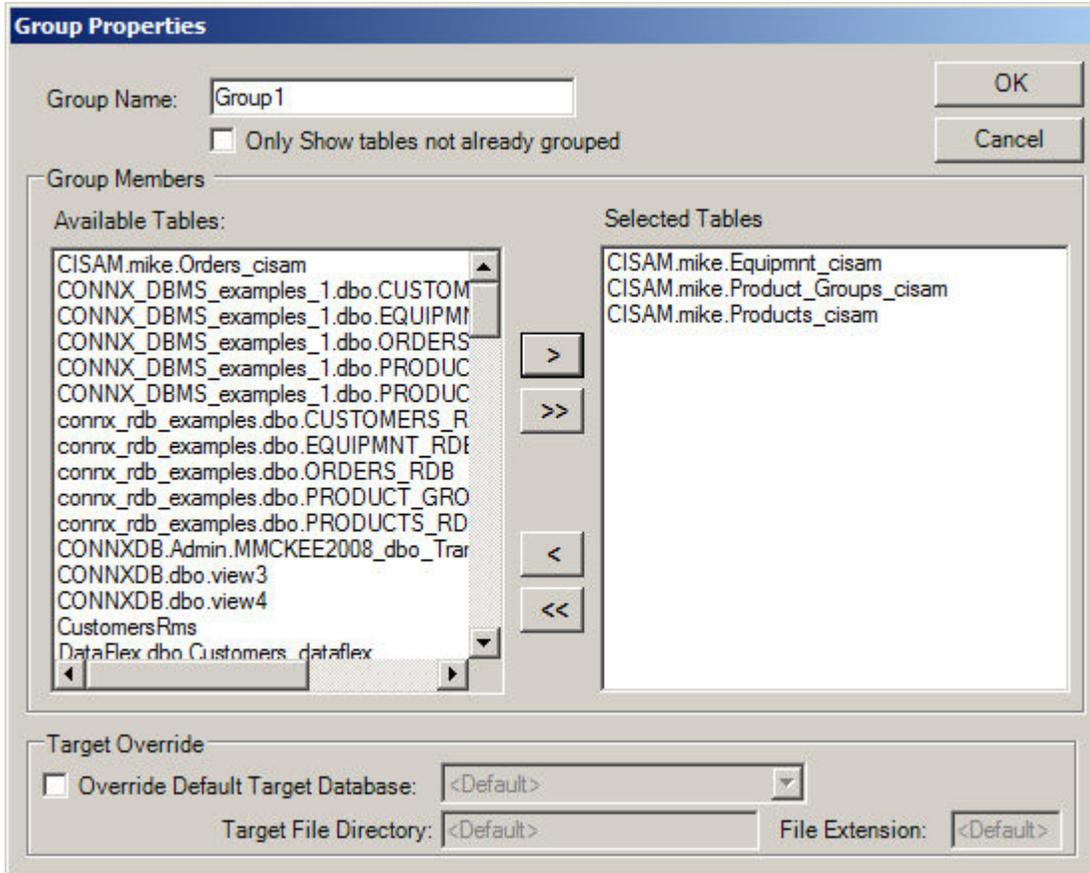




4. Click the **Only show tables not already grouped** check box to show ungrouped tables. Displaying and viewing ungrouped tables may help to avoid accidentally scheduling resynchronization of identical table(s). (Optional step.)
5. Optionally, you may override the default target database by clicking the **Override Default Target Database** check box and selecting a new target. Enter the target file name directory, if applicable. (This function is only available for VSAM and RMS databases, which use target file names.) To change the file name extension, for example, from .dat to .bak, enter the new file name extension in the **File Extension** text box.
6. Enter a name for the new group, and then select the tables to be included in the group.



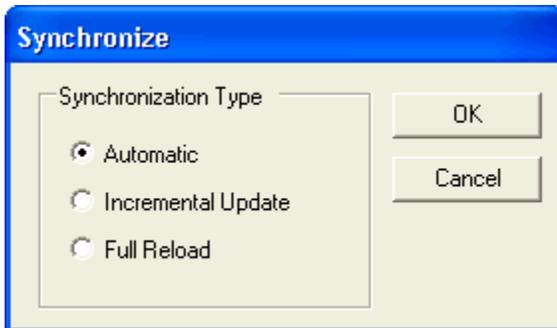
7. Click the single arrow button that points to the right to move the selected tables.



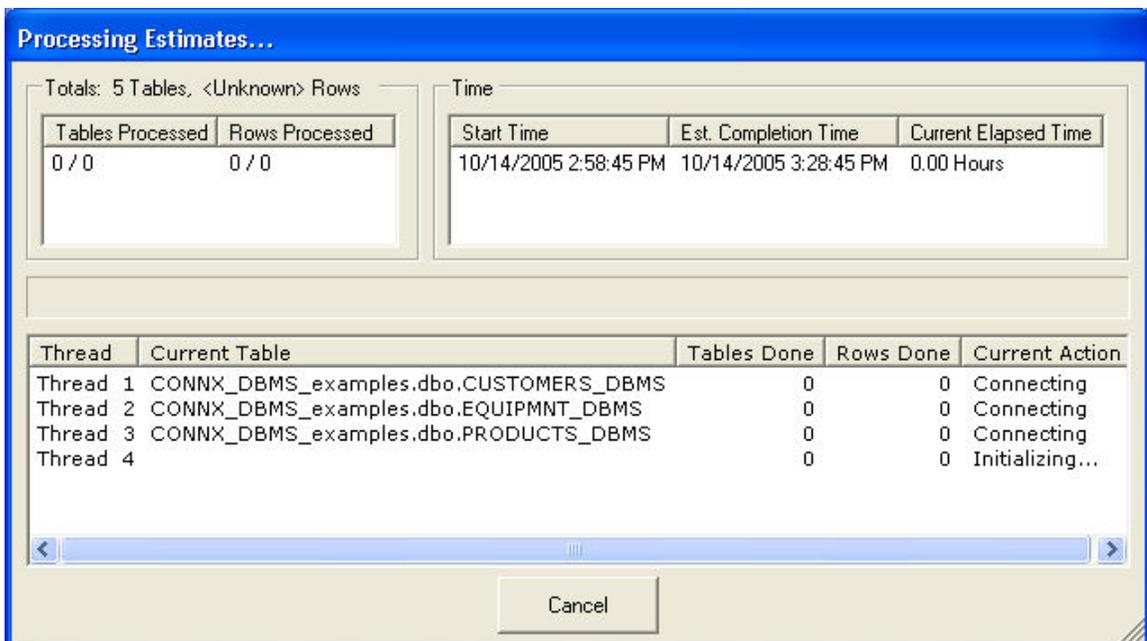
8. Click the **OK** button to create the group.

To synchronize a group on demand

1. Click the **Sync Now** button to start the synchronization process.
2. The **Synchronize** dialog box appears.



3. Select the synchronization type, depending on whether an incremental update or a full reload is preferred. The recommended synchronization for the select database is displayed.
4. Click the **OK** button to start the synchronization process.
5. The **Processing Estimates** screen appears.



6. The **top left pane** in the Synchronizing screen displays the number of tables and rows that have been processed so far. The totals from a previous run (if any) appears as a text string below the pane.
 - The **upper right pane** displays the start time, estimated completion time, and time spent so far in synchronization. The estimated completion time and elapsed time change as synchronization progresses.
 - The **slider bar** below the left pane gives a visual estimate of time until completion.

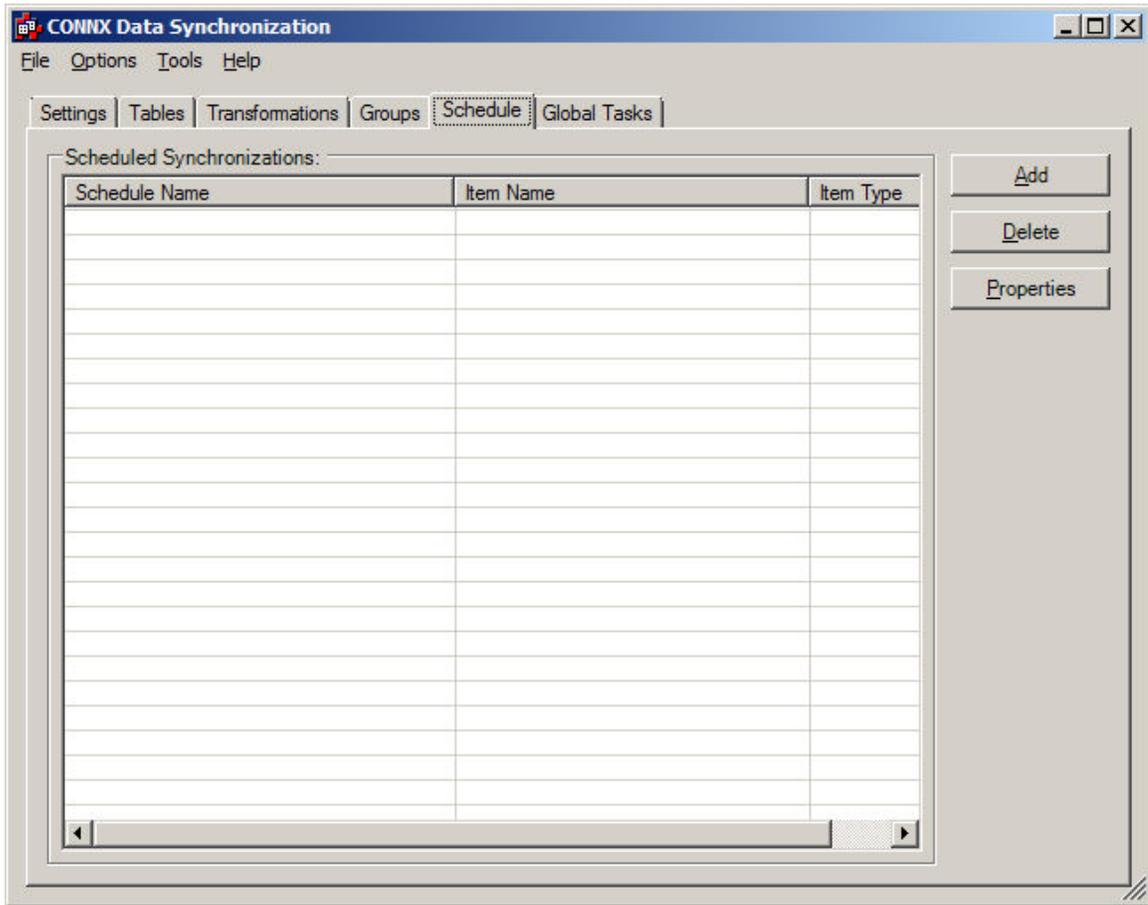
- The **bottom pane** displays the variety of thread tasks. The first column contains the number of threads being run, the second column contains the name of the table. The third column contains the number of tables completed. The fourth column contains the number of rows. The final column explains the active function of the currently running thread. When complete, the CONNX Data Synchronization window appears.
6. If you wish to perform this synchronization on a regular basis, you can create a schedule. To create a schedule, click the **Schedule** tab to establish the timing of the synchronization. See [To schedule a synchronization task](#) for more information.

Schedule Tab

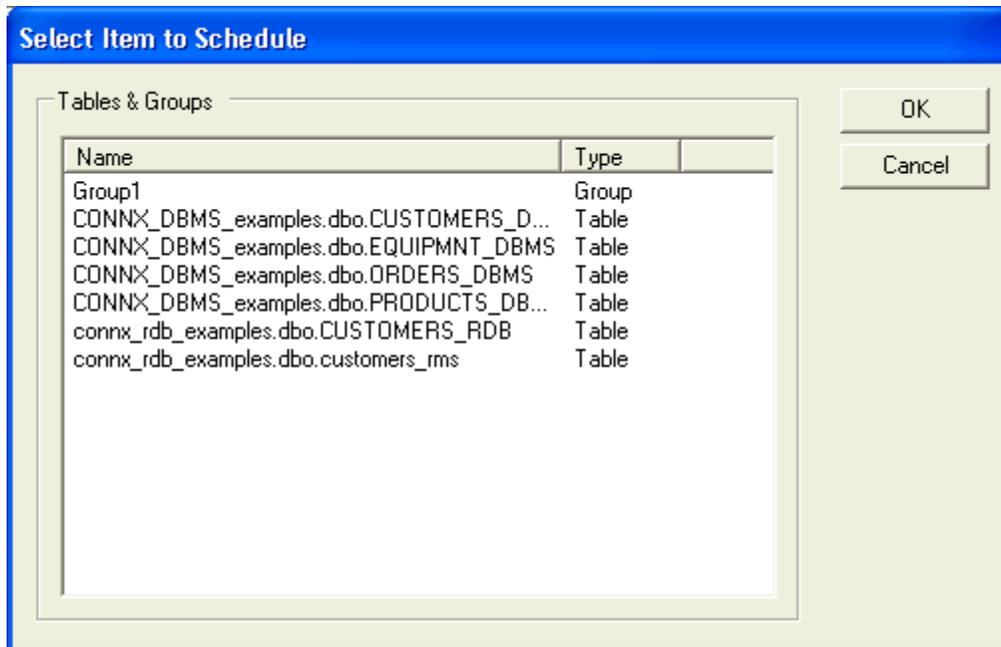
To schedule a synchronization task

Note: Do not attempt to schedule a new task while a Data Sync scheduling task is currently running. See [To select a target database for synchronization](#) for more information.

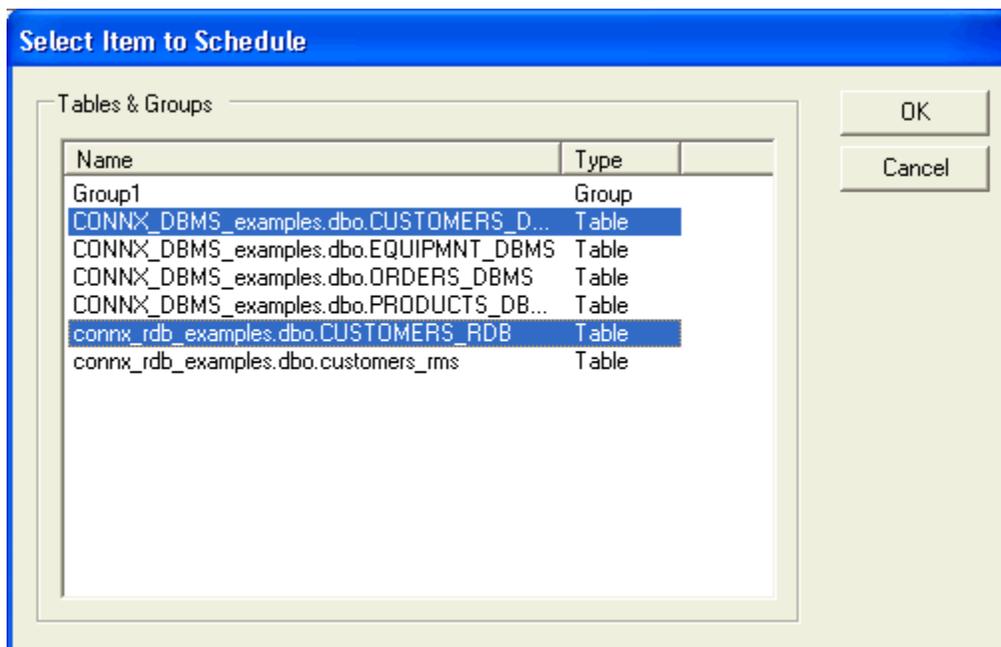
1. Select the **Schedule** tab in the CONNX Data Synchronization tool window.



2. Click the **Add** button to create a new scheduling task.
3. The **Select Item to Schedule** dialog box appears.



4. Select the tables or groups to be included in the scheduling task.



5. Click the **OK** button.
6. The **Synchronization Recurrence Frequency** dialog box appears.

Synchronization Recurrence Frequency

Schedule Name

Table CONNX_DBMS_examples.dbo.CUSTOMERS_DBMS

Synchronization Type

Automatic
 Incremental Update
 Full Reload

Trigger File

Run only when trigger file is present at scheduled time

Trigger File:

Synchronization Tasks

Before Synch Task:

On Success Task:

On Failure Task:

7. Enter a name for the schedule in the **Schedule Name** text box, and then click the **Set Recurrence Pattern** button.



8. The **Date and Time Selection** dialog box appears.

Date and Time Selection

Date Selection

Trigger every days

Trigger on specific days:

Time Selection

Trigger every Hours between and

Trigger at these times:

Active Range

This item is active beginning

This item is only active until

Holidays

No special treatment

Do not trigger on holidays

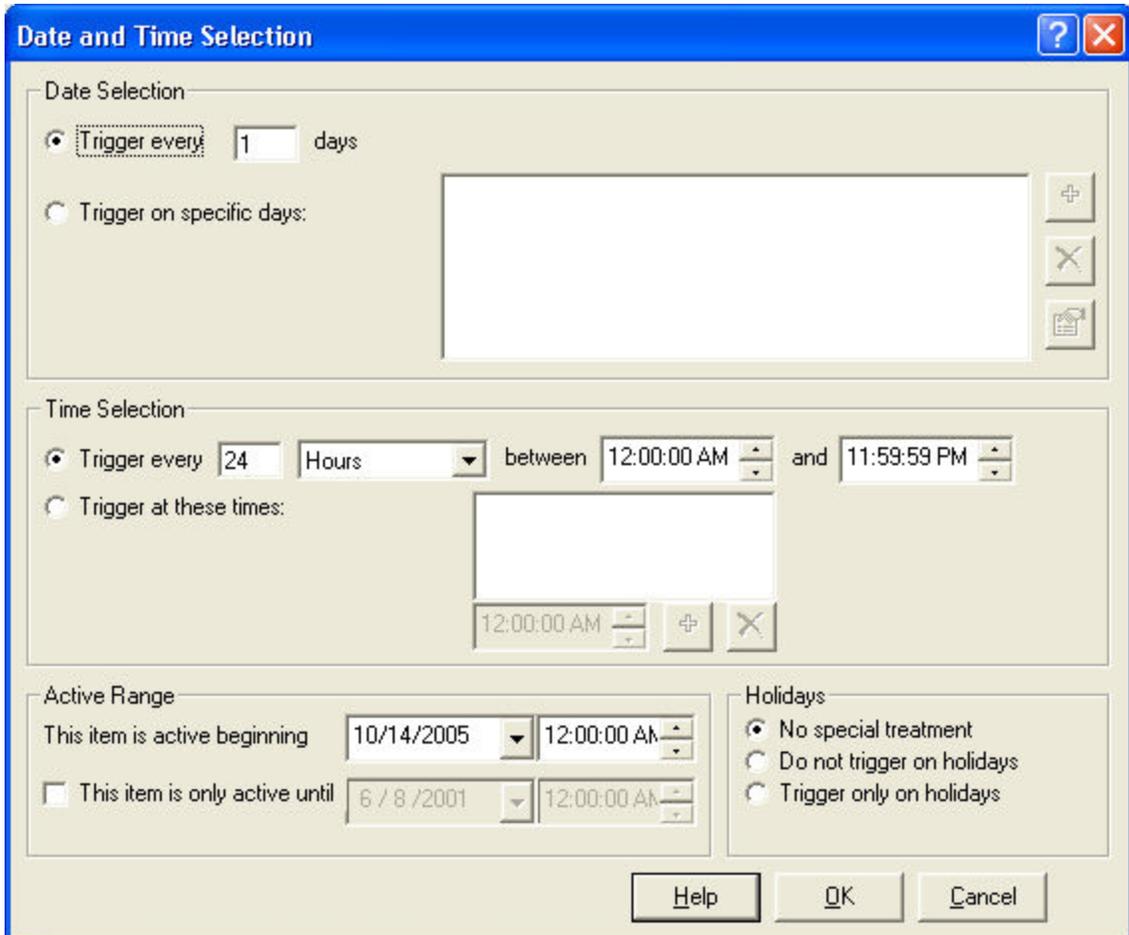
Trigger only on holidays

Help OK Cancel

9. The options available in this dialog box enable you to set a synchronization schedule that can occur on a specific day, or at a specific minute or hour. You can also establish a specific range of days or hours in which the synchronization is active, and then specify the day and time at which the synchronization ceases.
10. Click the **OK** button to return to the **Synchronization Recurrence Frequency** dialog box. Click the **OK** button. If the CONNXSchedule service has not been started, a dialog box appears asking if you would like to start the service. Click the **Yes** button.
11. If the service has been started once, then stopped, a dialog box appears asking if you would like to restart. Click the **Yes** button.
12. The **Synchronization Recurrence Frequency** dialog box appears. Click the **OK** button to continue.

To select a synchronization date

1. In the **Date and Time Selection** dialog box, under **Date Selection**, click the **Trigger every ___ days** radio button, and then enter a number in the text box.



2. If you want to change the trigger date, click the **Trigger on specific days** radio button, and then click the **Plus** sign button.



3. The **Select Days** dialog box appears.

Select Days

Months

January May September
 February June October
 March July November
 April August December

Days

Trigger on these days of the month: 1 2 3

Trigger on these days of the week: Sunday Monday Tuesday

Trigger 0 days After the Nearest relative to the beginning of the month

Trigger 0 days After the last day of the month

Trigger 0 days After Easter

Help OK Cancel

4. Under **Months**, select a month check box. You may select more than one month.

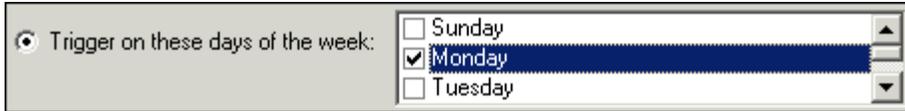
Months

January May September
 February June October
 March July November
 April August December

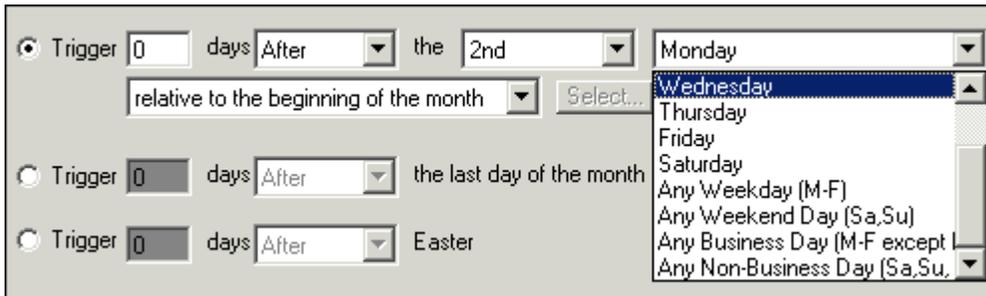
5. Under **Days**, there is a range of daily triggers to choose from. If you select **Trigger on these days of the month**, you must click the radio button, then click the check boxes for the days. You can choose one or more days.

Trigger on these days of the month: 1 2 3

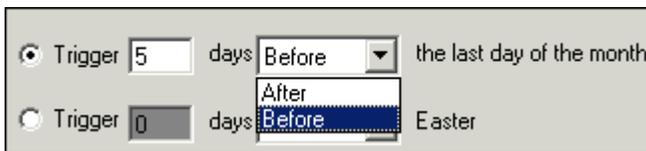
6. If you select **Trigger on these days of the week**, you must click the radio button, then click the check boxes for the days of the week. You can choose one or more days of the week.



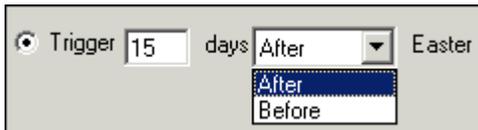
7. If you select **Trigger** ___ **days** ___ **the** _____, you must click the radio button, then choose items from the list boxes.



8. If you select **Trigger** ___ **days** ___ **the last day of the month**, you must click the radio button, enter a day, then choose either **After** or **Before** from the list box.



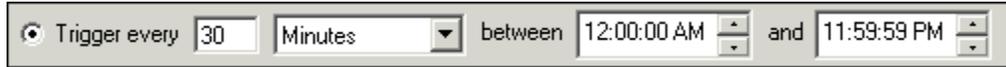
9. If you select **Trigger** ___ **days** ___ **Easter**, you must click the radio button, enter a day, then choose either **After** or **Before** from the list box.



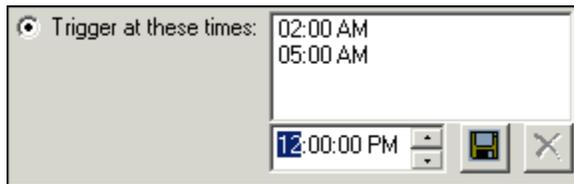
10. Click the **OK** button to return to the **Date and Time Selection** dialog box.

To select a synchronization time

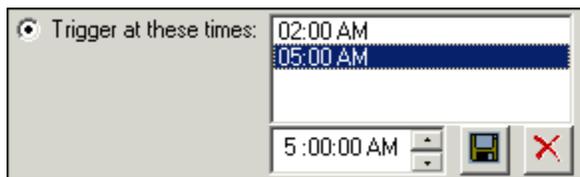
1. In the **Date and Time Selection** dialog box, under **Time Selection**, click the **Trigger every** ____ **between** ____ **and** ____ radio button, and then enter a number in the first text box. Select either **Minutes** or **Hours** from the first list box, then select the hour from the two clocks during which you want the synchronization task to run.



2. If the scheduling requirement is irregular, you may use the second option. Select the radio button, and then select the hours you need from the list box. When you have completed your scheduling input, click the **Save** icon.



3. To remove any of the information from this schedule, select the time you want to remove, then click the **Delete** button. To return to adding more scheduling time, click the **Plus** button.



4. Click the **OK** button to return to the **Synchronization Recurrency Frequency** dialog box.

To select an active range

1. In the **Date and Time Selection** dialog box, under **Active Range**, select a date from the **This item is active beginning** text box, and then select a time from the list box.

Active Range

This item is active beginning 10/18/2002 12:00:00 AM

This item is only active until 6 / 8 /2001 12:00:00 AM

2. To establish an end date for a synchronization task, select the **This item is only active until** check box.

This item is only active until 3 / 8 /2003 12:00:00 AM

3. Click the **OK** button to return to the **Synchronization Recurrence Frequency** dialog box.

To run a trigger file

1. In the **Synchronization Recurrence Frequency** dialog box, under **Trigger Files**, select the **Run only when trigger file is present at scheduled time** check box. Trigger files are created by external database processes and are designed to signal the appropriateness and timing of a synchronization. For example, a user might not want to run a scheduled synchronization until an important database backup is complete. The job that performs the database backup can delete the trigger file, perform the backup, and then create the trigger file. If the backup is still being run, the synchronization is further delayed until the trigger file is found. Once the backup is complete, the trigger file appears, and the synchronization proceeds as expected.
2. Enter the name of the trigger file in the **Trigger File** text box, and then click **OK**.

Synchronization Recurrence Frequency

Schedule Name:

Table: CONNX_DBMS_examples.dbo.CUSTOMERS_DBMS

Synchronization Type: Automatic Incremental Update Full Reload

Every 1 day(s) Every 24 Hours

Trigger File: Run only when trigger file is present at scheduled time

Trigger File:

Synchronization Tasks:

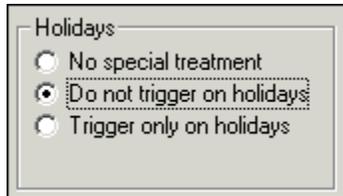
Before Synch Task:

On Success Task:

On Failure Task:

To establish holiday scheduling

1. In the **Date and Time Selection** dialog box, under **Holidays**, select from the choices by clicking either the **No special treatment**, **Do not trigger on holidays**, or **Trigger only on holidays** radio button.



The CONNX Data Synchronization Tool is currently programmed to include New Years Day (January 1), Memorial Day (May 30), Independence Day (July 4), Thanksgiving Day (fourth Thursday in November), Labor Day (first Monday in September), and Christmas (December 25) in its holiday schedule.

2. Click the **OK** button to return to the **Synchronization Recurrence Frequency** dialog box.

3. The **Synchronization Recurrence Frequency** dialog box appears.

The screenshot shows a dialog box titled "Synchronization Recurrence Frequency". At the top, there is a "Schedule Name" field containing "Schedule1" and a "Set Recurrence Pattern" button. To the right are "OK" and "Cancel" buttons. Below this, the "Group" is set to "Group1". The "Synchronization Type" section has three radio buttons: "Automatic" (selected), "Incremental Update", and "Full Reload". The text "On selected days" is visible to the right of these options. At the bottom, the "Synchronization Tasks" section contains three text input fields labeled "Before Synch Task:", "On Success Task:", and "On Failure Task:", all of which are currently empty.

4. Click the **Set Recurrence Pattern** button.



5. The **Date and Time Selection** dialog box appears. Check or change the properties of the synchronization triggers.

Date and Time Selection [?] [X]

Date Selection

Trigger every days

Trigger on specific days: [+] [X] [📅]

Time Selection

Trigger every between and

Trigger at these times:

 [+] [X]

Active Range

This item is active beginning

This item is only active until

Holidays

No special treatment

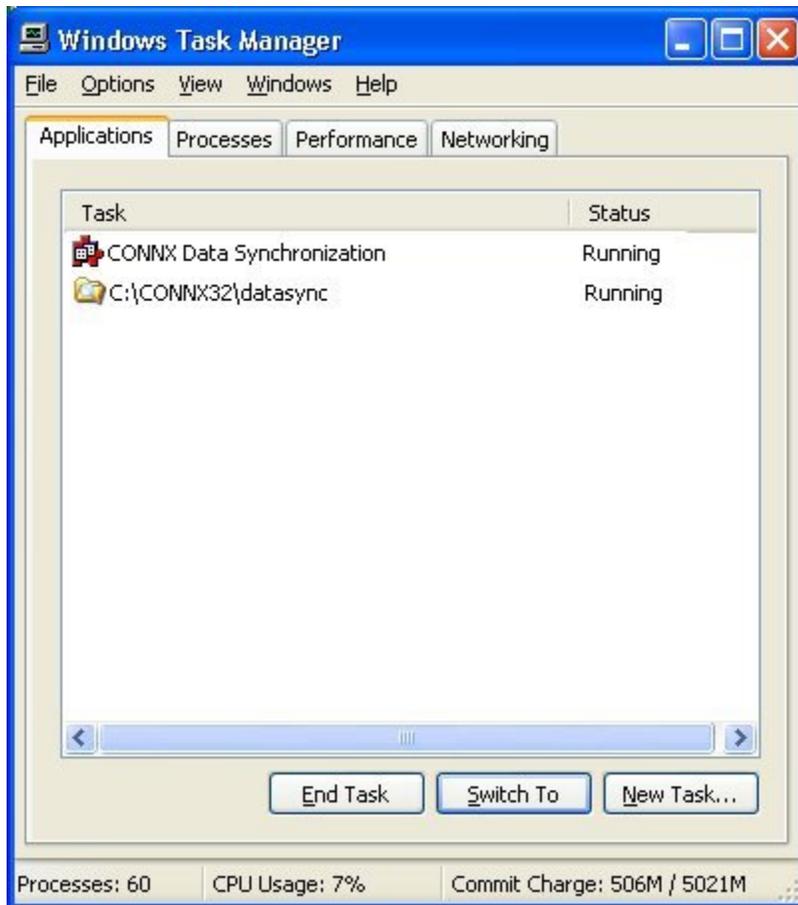
Do not trigger on holidays

Trigger only on holidays

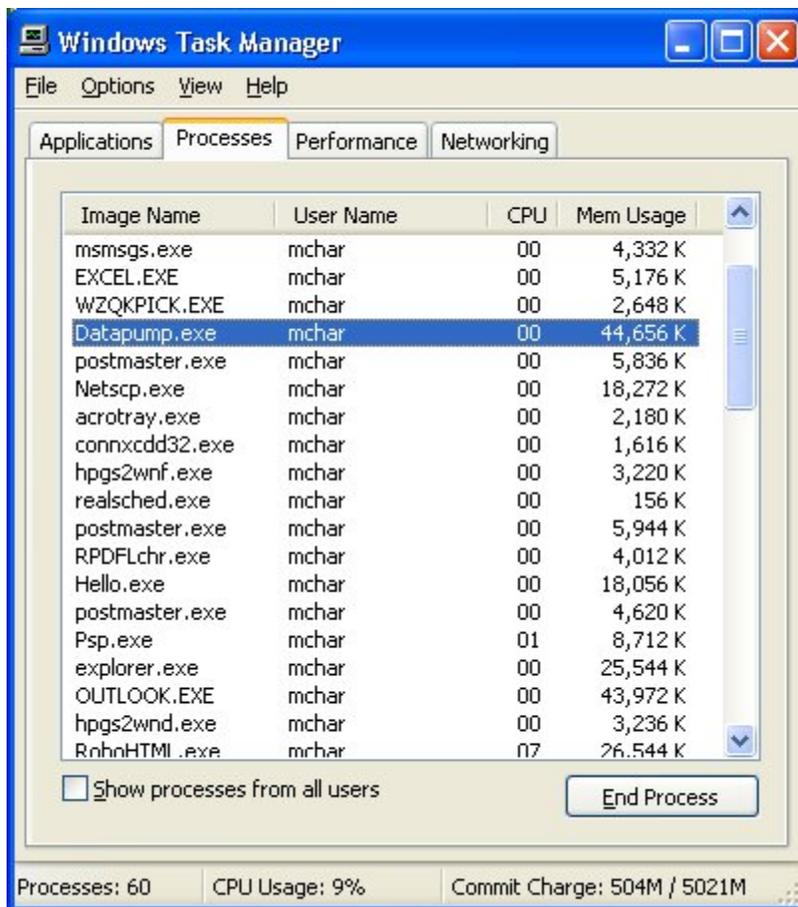
[Help] [OK] [Cancel]

To verify or view a synchronization in process

1. To view or verify a synchronization in process, click **Control, Alt, Delete** to open the Windows Task Manager window.



2. On the **Processes** tab, locate postmaster.exe, ScheduleService.exe, and Datapump.exe.



3. If the processes are running, the CPU time changes accordingly.

Per Schedule Synchronization Tasks

Per Schedule Synchronization Tasks

To associate tasks with individual synchronizations on specific tables or groups of tables, use the text boxes below Synchronization Tasks in the Synchronization Recurrence Frequency window. The tasks defined in this window run on distinct schedules, unlike Global Scheduled or Global On-Demand synchronization tasks which run for every scheduled synchronization. See also [To schedule a synchronization task](#).

Synchronization Recurrence Frequency

Schedule Name:

Table: CONNX_DBMS_examples.dbo.CUSTOMERS_DBMS

Synchronization Type:
 Automatic
 Incremental Update
 Full Reload

Trigger File: Run only when trigger file is present at scheduled time
 Trigger File:

Synchronization Tasks:
 Before Synch Task:
 On Success Task:
 On Failure Task:

Before Synch Task

To define a task to run before a synchronization performed on a specific table or group of tables, use the Before Synch text box.

The screenshot shows the 'Synchronization Recurrence Frequency' dialog box. The 'Schedule Name' is 'Schedule1'. The 'Table' is 'CONNX_DBMS_examples.dbo.CUSTOMERS_DBMS'. The 'Synchronization Type' is 'Automatic'. The 'Trigger File' section is checked and set to '\\pwwshare\operations\rms_ready.txt'. The 'Synchronization Tasks' section has three text boxes: 'Before Synch Task:', 'On Success Task:', and 'On Failure Task:'. A red arrow points to the 'Before Synch Task:' text box.

For example, you may wish to synchronize a particular set of database files, and then inform the administrator of that particular database of the scheduled synchronization.

1. Compose a command line that resembles the following:

```
c:\utils\epsendmail nickd@connx.com, databaseadmin-ops@connx.com, "DB Sync", "DB Sync Task started", 255.255.255.255
```

2. Test your work with the [Global On-Demand Synchronization Before Sync Task](#) and then cut-and-paste it into the Before Synch Task text box under Synchronization Tasks in the Synchronization Recurrence Frequency window. Substitute the TCP/IP address of your SMTP mail server for the TCP/IP address shown above.

On Success Task

To define a task to run if a synchronization performed on a specific table or group of tables was successful, use the On Success Task text box.

The screenshot shows the 'Synchronization Recurrence Frequency' dialog box. It has a blue title bar and a light beige background. At the top, there's a 'Schedule Name' field with 'Schedule1' and an 'OK' button. Below that is a 'Set Recurrence Pattern' button and a 'Cancel' button. The 'Table' field contains 'CONNX_DBMS_examples.dbo.CUSTOMERS_DBMS'. Under 'Synchronization Type', there are three radio buttons: 'Automatic' (selected), 'Incremental Update', and 'Full Reload'. The frequency is set to 'Every 1 day(s) Every 24 Hours'. The 'Trigger File' section has a checked box for 'Run only when trigger file is present at scheduled time' and a text box containing '\\pwwshare\operations\rms_ready.txt'. At the bottom, there's a 'Synchronization Tasks' section with three text boxes: 'Before Synch Task:', 'On Success Task:', and 'On Failure Task:'. A red arrow points to the 'On Success Task:' text box.

For example, you may wish to synchronize a particular set of database files, and then inform the administrator of the success of the scheduled synchronization.

1. Compose a command line that resembles the following:

```
c:\utils\epsendmail nickd@connx.com, databaseadmin-ops@connx.com, "DB Sync", "DB Sync Task started", 255.255.255.255
```

2. Test your work with the [Global On-Demand Synchronization Before Sync Task](#) and then cut-and-paste it into the Before Synch Task text box under Synchronization Tasks in the Synchronization Recurrence Frequency window. Substitute the TCP/IP address of your SMTP mail server for the TCP/IP address shown above.

On Failure Task

To define a task to run if a synchronization performed on a specific table or group of tables was unsuccessful, use the On Success Task text box.

Synchronization Recurrence Frequency

Schedule Name:

Table:

Synchronization Type: Automatic Incremental Update Full Reload

Every 1 day(s) Every 24 Hours

Trigger File: Run only when trigger file is present at scheduled time

Trigger File:

Synchronization Tasks:

Before Synch Task:

On Success Task:

On Failure Task:

To inform the administrator of the failure of a scheduled synchronization, for example:

1. Compose a command line that resembles the following:

```
c:\utils\epsendmail nickd@connx.com, databaseadmin-ops@connx.com, "DB Sync Failure", "DB Sync Task failed", 255.255.255.255
```

2. Test your work with the [Global On-Demand Synchronization On Failure Task](#) and then cut-and-paste it into the On Failure Task text box under Synchronization Tasks in the Synchronization Recurrence Frequency window. Substitute the TCP/IP address of your SMTP mail server for the TCP/IP address shown above.
3. To test the message failure behavior, you will have to cause a synchronization to fail. In lieu of that task, test the message in the [Global On-Demand Synchronization On Success](#) task text box and then paste the identical command into the On Failure text box under Synchronization Tasks in the Synchronization Recurrence Frequency window described above.

Global Tasks Tab

Available Types of Scheduled Tasks

Scheduled Tasks

CONNX Data Synchronization enables the user to execute globally scheduled tasks that occur simultaneously with synchronization of data. These tasks can be any sort of operation, including batch files, executable files, backups, or data dumps. Execution of the command is automatic.

Tasks that should not be included in globally scheduled synchronization tasks include any task that requires input from a user. Such tasks cause the synchronization to stop abruptly as it waits for user input.

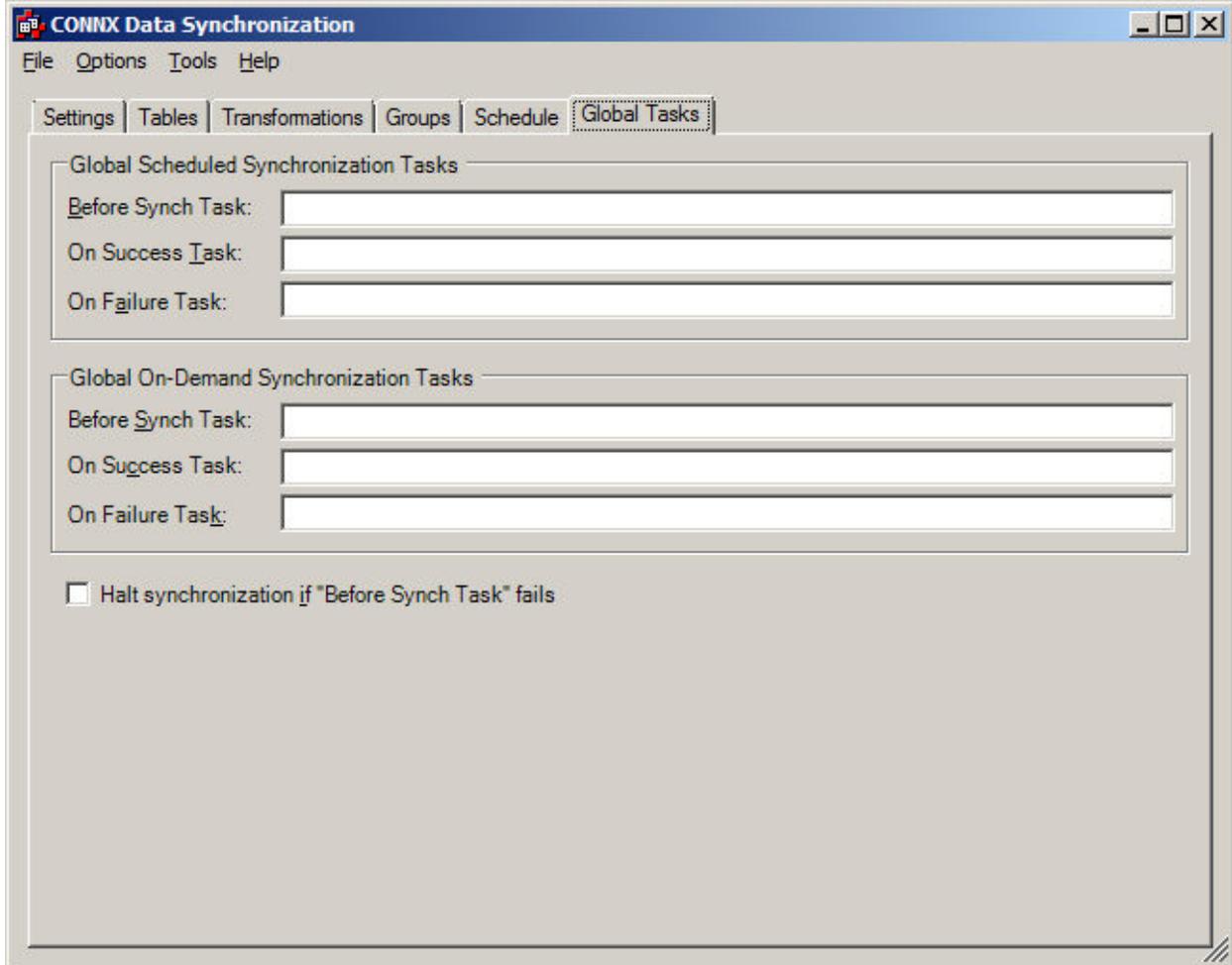
There are three types of tasks available:

- [Global Scheduled Synchronization Tasks](#)
- [Global On Demand Synchronization Tasks](#)
- [Per Schedule Synchronization Tasks](#) (Located on the Schedule tab)

Global Scheduled Synchronization Tasks

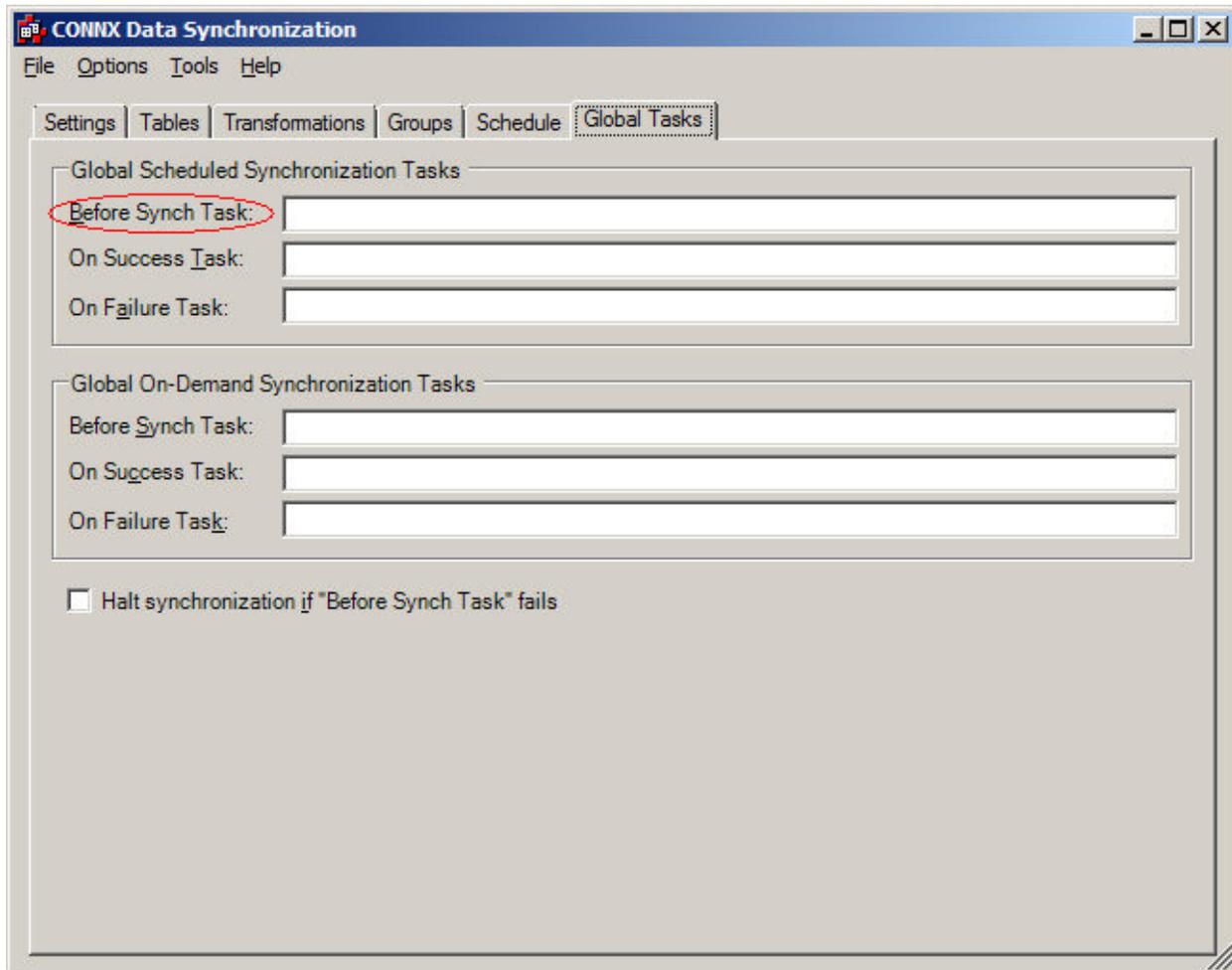
Global Scheduled Synchronization Tasks

Global Scheduled Synchronization Tasks can be performed for every scheduled synchronization.



Global Scheduled Synchronization Task - Before Synch

The Global Scheduled Synchronization Task entered into the **Before Synch Task** text box runs before every scheduled synchronization. For example, you could choose to send an e-mail to yourself to show that a task was started.



1. Open and download EpSendMail from <http://www.codeproject.com/vb/net/epsendmail.asp>.
2. Since the following is the expected format for using the tool...

Usage: EPSendMail from@email.com, to@email.com, subject, message, [smtp Server], [cc1@email.com;cc2@email.com;...], [attachment1;attachment2 ;...]

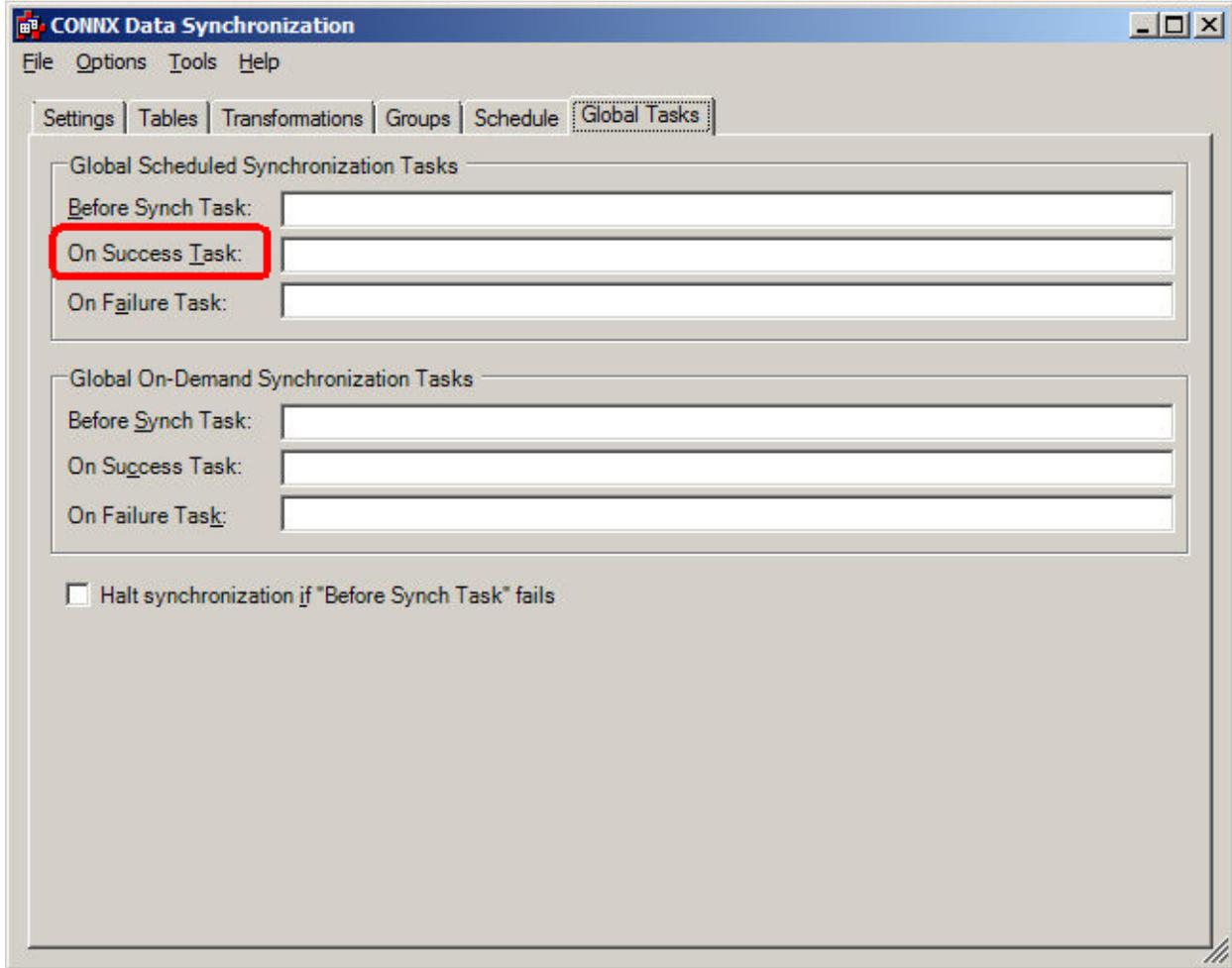
...a usable command line would look like this:

```
c:\utils\epsendmail dogbert@connx.com, dogbert@connx.com, "Task
Started", "Global Synchronization Task started", 255.255.255.255
```

3. Test the command through the **Global On Demand Synchronization Before Synch Task** text box, and then cut-and-paste it into the **Before Synch Task** text box under **Global Scheduled Synchronization** tasks. Instead of 255.255.255.255, enter the TCP/IP address of your SMTP mail server.

Global Scheduled Synchronization Task - On Success Task

The Global Scheduled Synchronization Task entered into the **On Success Task** text box runs after every scheduled synchronization. For example, you could choose to send an e-mail to yourself to show that a task succeeded, so that you can check that your scheduled tasks are performing correctly.



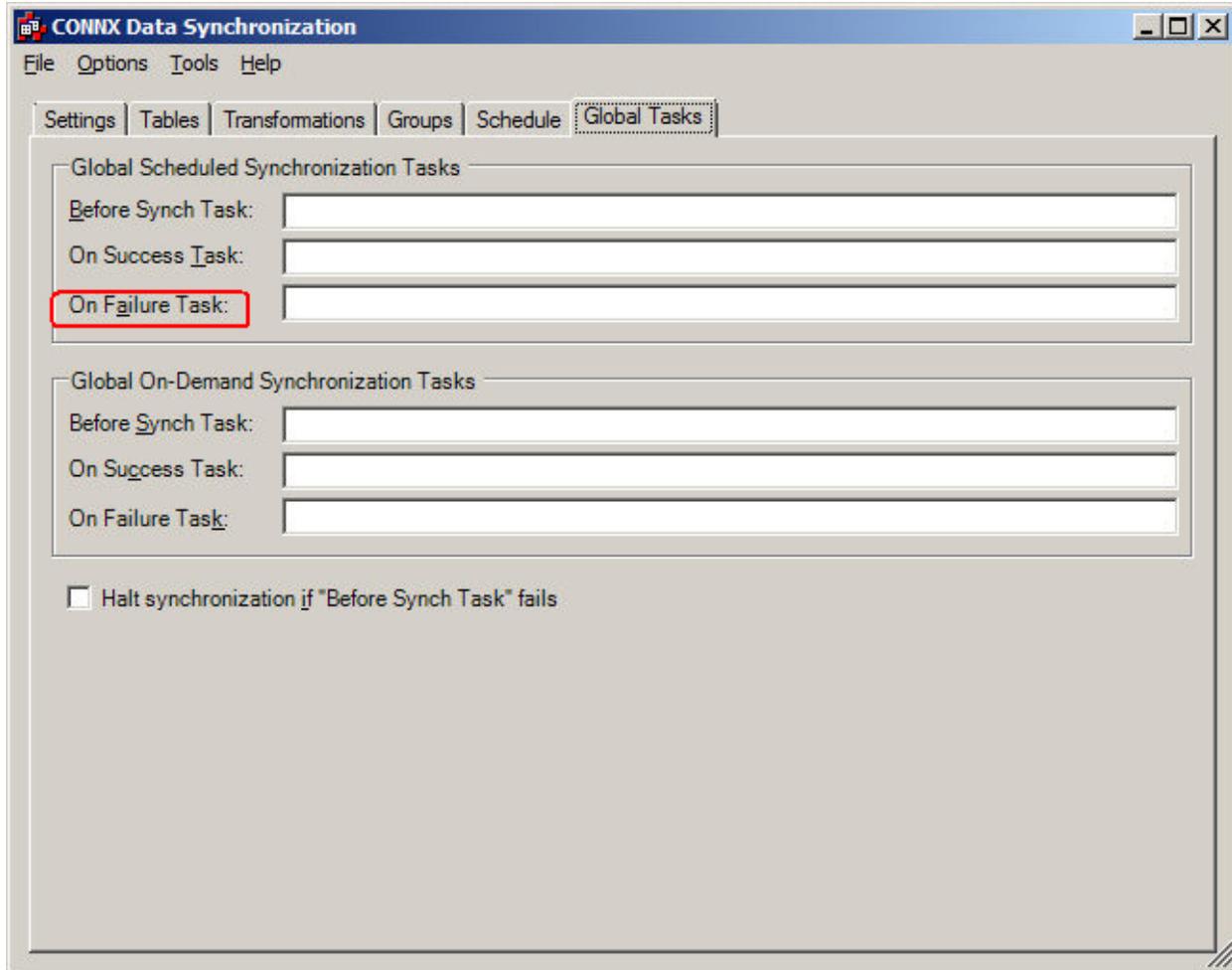
For example:

1. Compose a command line that resembles the following:

```
c:\utils\epsendmail nickd@connx.com, nickd@connx.com, "Task Succeeded",
"Global Synchronization Task succeeded", 255.255.255.255
```
2. Test the command through the **Global On Demand Synchronization On Success Task** text box, and then cut-and-paste it into the **On Success Task** text box under **Global Scheduled Synchronization** tasks. Instead of 255.255.255.255, enter the TCP/IP address of your SMTP mail server.

Global Scheduled Synchronization Task - On Failure Task

The Global Scheduled Synchronization Task entered into the **On Failure Task** text box runs after every failed synchronization. For example, you could choose to send an e-mail to Operations to show that a task failed, so that you can receive instructions on how to correct the problem.



For example:

1. Compose a command line that resembles the following:

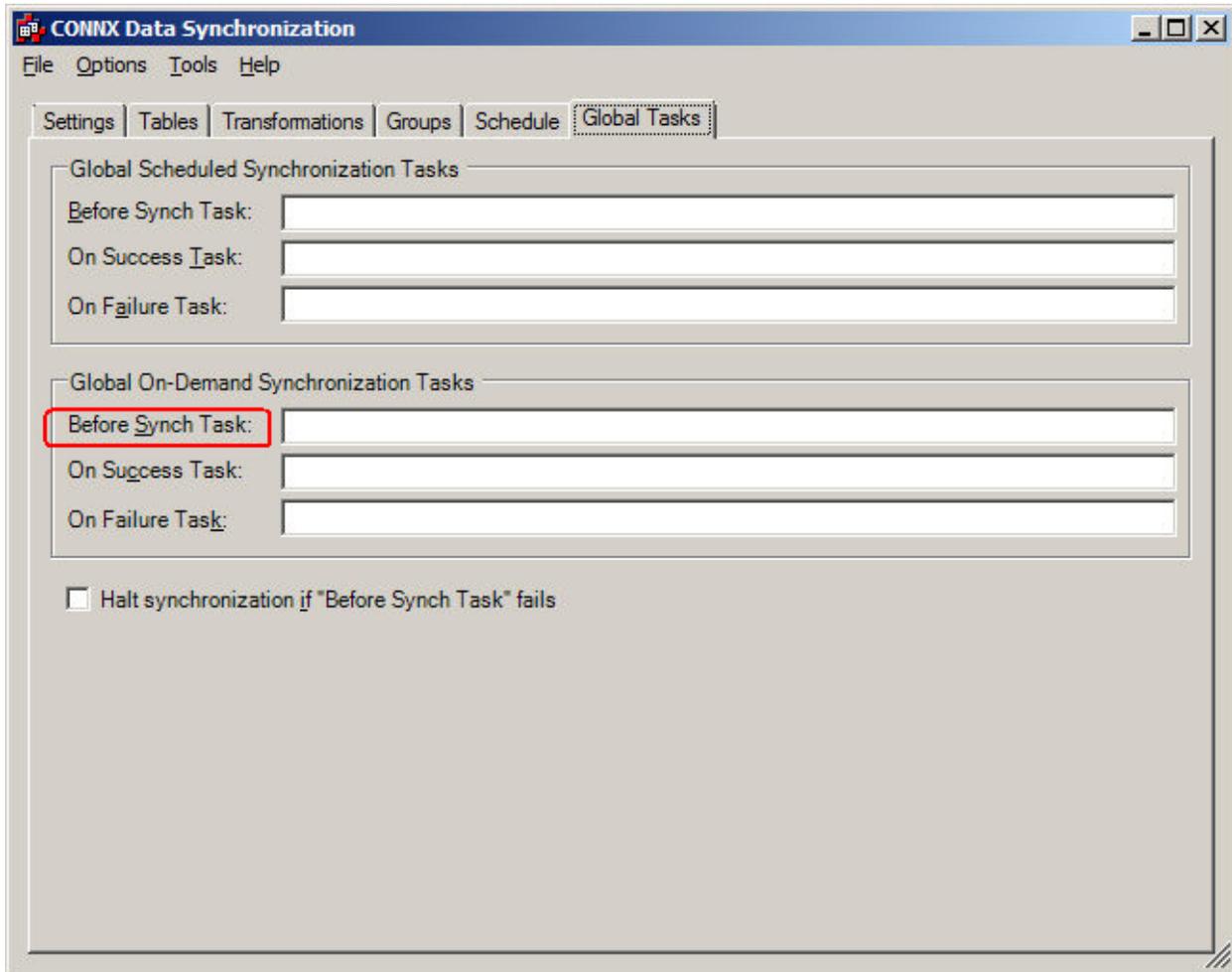
```
c:\utils\epsendmail nickd@connx.com, operations@connx.com, "Task
Succeeded", "Global Synchronization Task succeeded", 255.255.255.255
```

2. Test the command through the **Global On Demand Synchronization On Failure Task** text box, and then cut-and-paste it into the **On Failure Task** text box under **Global Scheduled Synchronization** tasks. Instead of 255.255.255.255, enter the TCP/IP address of your SMTP mail server.

Global On-Demand Synchronization Tasks

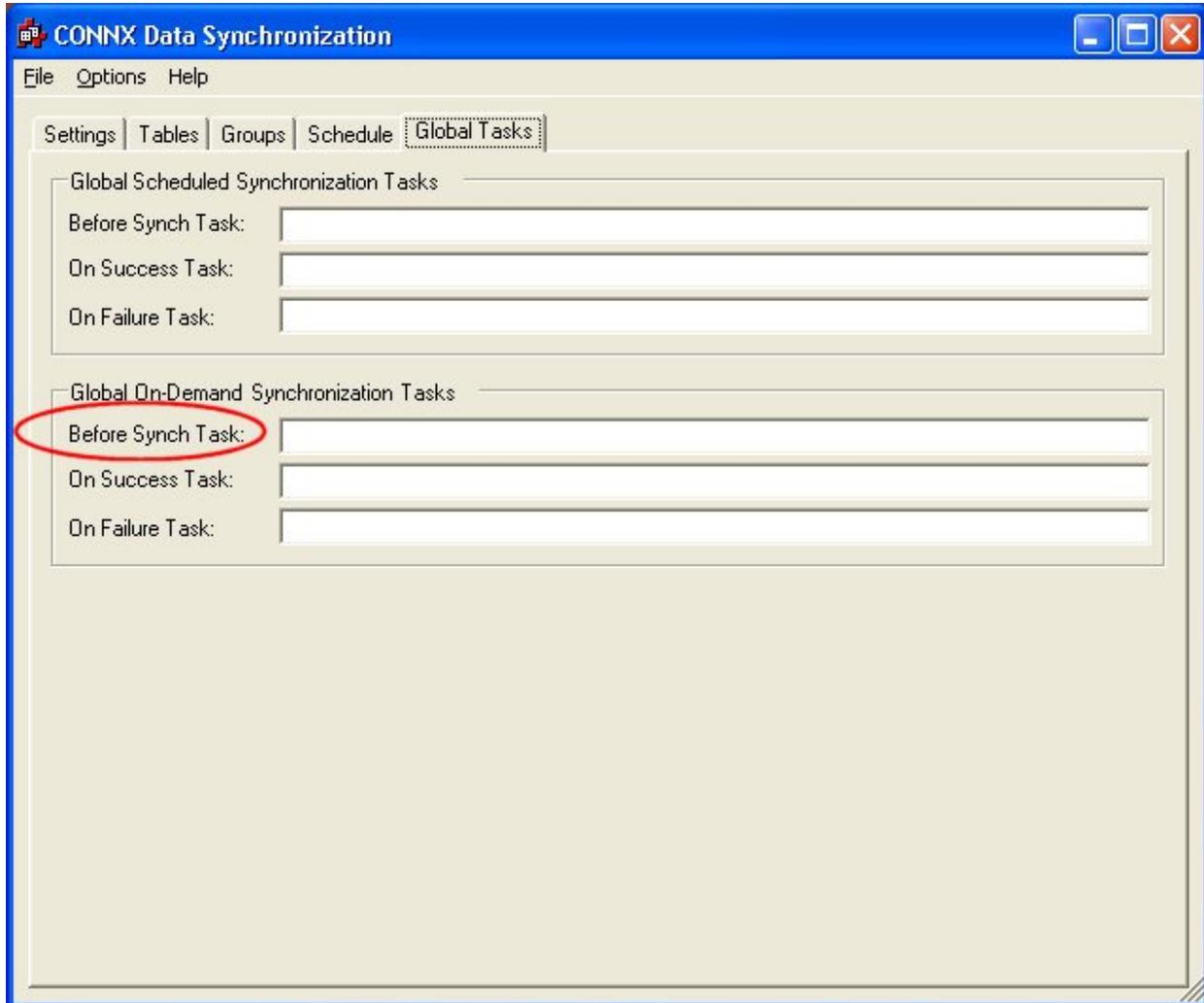
Global On-Demand Synchronization Tasks

Global On-Demand Synchronization tasks can be performed for every interactive synchronization that is manually started.



Global On-Demand Synchronization Task - Before Synch Task

The Global On-Demand Synchronization Task entered into the **Before Synch Task** text box runs before every on-demand synchronization.



For example, you could choose to dump your target database before performing synchronizations:

1. Create the following task under Global On-Demand Synchronizations Tasks in the **Before Synch Task** text box:

```
isql -S servername -Usa -P username -Q"backup database corporate to db_store"
```
2. When the synchronization is run, an entry resembling the following appears in the log file if the synchronization was successful:

```
Log Entry: Thursday, September 9, 2004 3:18:39 PM
Executed BeforeSync Task. isql -S servername -Usa -P username -Q"backup database corporate to db_store" ReturnStatus = 0
```

Global On-Demand Synchronization Task - On Success Task

The Global On-Demand Synchronization Task entered into the **On Success Task** text box runs after every successful on-demand synchronization.

For example, you could choose to execute a stored procedure after successfully performing synchronizations:

1. Create the following task under Global On-Demand Synchronizations Tasks in the **On Success Task** text box:


```
isql -S servername -Usa -P username -d workflow -Q"exec populate_views"
```
2. When the synchronization is run, an entry resembling the following appears in the log file if the synchronization was successful:


```
Log Entry: Thursday, September 9, 2004 3:18:39 PM
Executed AfterSyncSuccess Task. isql -S servername -Usa -P username -d
workflow -Q"exec populate_views" ReturnStatus=0
```

Global On-Demand Synchronization Task - On Failure Task

The Global On-Demand Synchronization Task entered into the **On Failure Task** text box runs after every failed on-demand synchronization.

The screenshot shows the 'CONNIX Data Synchronization' application window. The 'Global Tasks' tab is selected. It contains two sections: 'Global Scheduled Synchronization Tasks' and 'Global On-Demand Synchronization Tasks'. Each section has three text input fields: 'Before Synch Task:', 'On Success Task:', and 'On Failure Task:'. The 'On Failure Task:' field in the 'Global On-Demand Synchronization Tasks' section is highlighted with a red rectangular box. At the bottom, there is a checkbox labeled 'Halt synchronization if "Before Synch Task" fails'.

For example, you could choose to send an e-mail and restore the database in which the synchronization failed by creating a batch job that contains the commands to perform both operations:

1. Create the following task under Global On-Demand Synchronization Tasks in the **On Failure Task** text box:
`C:\utils\global-failure.cmd`
2. When the file global-failure contains an e-mail command and also an ISQL command to restore the database, both commands execute whenever an executed on-demand synchronization fails.

Troubleshooting

Purging History feature

If the CONNX Data Synchronization tool runs for a lengthy period of time, historical information that accumulates may decrease its high performance levels. Use the following procedure to purge the historical information.

- On the **Options** menu, select **Purge History**. The historical information is removed from the log.



Compact Database feature

To compress the CONNXStore database and restore lost performance levels, use the Compact Database feature.

- On the **Options** menu, select **Compact Database**. The contents of the CONNXStore database are compressed.



Duplicate keys

If a target index does not build, a message about duplicate keys may appear. This may be due to case sensitivity of the source and target servers.

For example, an RMS server may read "JOHN DOE" and "John Doe" and "john doe" differently, but SQL Server may read them equally.

Although a unique index may exist on a source database, indicating that the data is unique, the server may not accept the data and declare that it is not unique.

To analyze this problem, first perform a query consisting of concatenating the key columns together and checking for counts greater than one. Such a query might show 4 duplicate rows for a table with unique key columns Name and Address, for example.

To solve the problem, correct the data on the source server. If (for instance) "John Doe", "JOHN DOE" and "john doe" are really the same person, merge the information in the three records together.

In cases where part number "aaa-2756" is, in fact, distinct from part number "AAA-2756", build SQL Server (or whatever other target server we are using) in case-sensitive mode.

If an incremental synchronization gives an error about duplicate keys, the error message provides a sample query that should return the duplicate data to aid in resolving the issue. Insert the query directly into the InfoNaut query tool to locate the duplicate data.

Error messages

Error message	Resolution
<p>The CONNX Listener process (CNXRUN##_MAIN) is not running on the system.</p>	<p>The listener is not running or otherwise unavailable on the database server.</p>
<p>Colliding Schedules. This schedule will not be run.</p>	<p>You have overlapping schedules. Perhaps you have run the synchronization task on a basis that is too frequent for your Win32 synchronization server to keep up. You can tell how long (in seconds) table synchronizations take to perform by performing the following query:</p> <pre> SELECT b.TableName, timestampdiff(SQL_TSI_SECOND, a.LastSynchStart, a.LastSynchEnd) AS "seconds", a.SynchType, a.InsertedRowCount, a.UpdatedRowCount, a.DeletedRowCount , a.Cardinality FROM CONNXDataSync..TableSynchronizations a, CONNXDataSync..TableList b WHERE a.TableID = b.TableID ORDER BY timestampdiff(SQL_TSI_SECOND, a.LastSynchStart, a.LastSynchEnd) desc </pre> <p>You can tell how long (in seconds) groups take to synchronize with the following query:</p> <pre> SELECT b.GroupName, timestampdiff(SQL_TSI_SECOND, a.LastSynchStart, a.LastSynchEnd) AS "seconds", a.SynchType, a.InsertedRowCount, a.UpdatedRowCount, a.DeletedRowCount , a.Cardinality FROM CONNXDataSync..GroupSynchronizations a, CONNXDataSync..Groups b WHERE a.GroupID = b.GroupID ORDER BY timestampdiff(SQL_TSI_SECOND, a.LastSynchStart, a.LastSynchEnd) desc </pre> <p>If you do not require a complete history, you can simply look at the Tables and Groups tabs to see about how long the last synchronization took. By taking a look at how long the tasks take, you can decide how frequently you might schedule them. If a file takes 300 seconds to synchronize, then you would not want to schedule synchronizations more frequently than perhaps 10 minutes (twice the sync rate). If a group takes 4000 seconds to synchronize then one hour is too frequent, since that is only 3600 seconds.</p>

CONNX Data Synchronization 12.5

Probably out of disk space.	Perform some disk housekeeping or purchase a larger disk drive.
Unable to start task (the paging file is too small for this operation to complete).	Create a larger page file on your system.

Command-line Feature**Starting a synchronization from a command line**

It's possible to start a CONNX data synchronization task from a command line through the executable file located at ..\..\datasync\CONNXsyncitem.exe. Once the file is installed, a synchronization can be performed on demand.

The command line interface is as follows:

```
CONNXSyncitem.exe <table, transform, or group name>@<Group|Table>@<F|A|I>
```

where

F = Full Sync

A = Automatic

I = Incremental

Example:

```
C:\CONNX32\datasync\CONNXSyncitem.exe Transform1 @Table@A
```

The command-line parameters are as follows:

Parameter	Definition
<table or group name>	The name given to the item or collection to be synchronized by the CONNX Data Synchronization tool.
Group Table (one or the other, but not both) - This is case sensitive	Tells the CONNX Data Synchronization tool the type of object to be synchronized, for example, whether it is a group of tables & transforms or a single table/transform.
F A I (must be F or A or I)	Defines the synchronization type - F=Full, A=Automatic, I=Incremental.

Trademarks and Copyrights

CONNX Data Synchronization Tool

This guide describes the CONNX Data Synchronization tool. Consult your sales representative for licensing policies. The term CONNX used throughout the manual applies to all CONNX products, unless otherwise specified. No responsibility is assumed for third-party products that may be used in conjunction with the software described in this document.

Content is subject to change without notice. Data used in examples is fictitious unless noted. No part of this document may be reproduced or transmitted in any form or by any means, for any purpose, without the express written consent of CONNX Solutions.

CONNX Data Synchronization, version 12.5

© CONNX Solutions, Inc., 2016

All rights reserved.

About CONNX Solutions and CONNX

Established in 1989, CONNX Solutions was originally founded as a division of SolutionsIQ, a leading software consulting and technical staffing company based in Bellevue, Washington. The division became a separate company in 2001, focusing on products and services that help companies maximize the value of their information assets. Our motto is: "Data access and integration made simple." With our flagship product, CONNX, a single metadata model can be created that spans all enterprise data sources and applications requiring data access. The result is an enterprise-wide view of data that provides a reusable standards-based framework for secure read/write information access. CONNX Solutions offers a variety of consulting, training, and support options for CONNX customers. With over 3000 customers worldwide, CONNX is relied upon by government and business entities in a wide range of industries, including manufacturing, healthcare, financial services, telecommunications, aerospace, and information technology.

CONNX Solutions, Inc., 2039 152nd Avenue NE, Redmond, WA, USA

Telephone: (425) 519-6600

Toll Free: (888)-88-CONNX

Facsimile: (425) 519-6601

Information: info@connx.com

Internet: www.connx.com

Technical Support (Maintenance Contract): (888) 930-2727

Technical Support E-mail: support@connx.com

CONNX, InfoNaut, Simplified Data Access, and CONNX Data Synchronization are registered trademarks or trademarks of CONNX Solutions in the USA and other countries.

Portions of the PostgreSQL Data Base Management System are copyrighted © 1996-2002, PostgreSQL Global Development Group

Portions Copyright © 1994-1996 Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A

PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

Other product and company names used within this document may be trademarks or registered trademarks of their respective owners or organizations in the United States and/or other countries.

Index

A

Active range 157
 select 157
Add 38, 60
 Select 38
Add button 5, 142, 149
Address 182
Administration Guide 7
Administrative Tools 7
Administrators 7, 166, 167, 168
After 153
Allow Pure Passthrough Queries 7
Alt 163
AS 183
Automatic 52, 170

B

Bak 142
BASIS 186
Before 33, 56, 153, 166, 172, 177
Before Sync Task 27
Before Synch 166, 172, 177
 use 166
Before Synch Task 166, 167, 172, 177
Bellevue 186
blobs 42
Browse button 7
buffer 7

C

Calling 4
 ETL 4
Cancel button 7
Case-insensitive 49
CDD 5, 7
cdd file 7
Change button 7
CHAR 7, 44
Check 7
 Recreate 7
Christmas 159
C-ISAM 27

CLOB 43
Close button 7
Close Control Panel 7
CNXRUN##_MAIN 183
CnxSyncDate 38
Colliding 183
 Schedules 183
Column 38, 45, 87
 target tables 38
Column Mapping 87, 93
Column Name 52, 182
Column Tab 87
command line 185
Compact Database feature 181
Compare COUNT 46
compression 181
CONNX ... 1, 2, 4, 5, 7, 27, 29, 35, 37, 42, 46, 47, 48, 49, 52, 142, 147, 149, 159, 170, 180, 183, 186
CONNX CD 7
 Return 7
CONNX CDD 7
 modifying 7
CONNX CDD Administrator 7
CONNX Data 1, 2, 4, 5, 7, 27, 29, 35, 42, 47, 52, 142, 147, 149, 159, 170, 180, 186
 Installing 7
CONNX Data Dictionary 5, 7, 27, 52
CONNX Data Dictionary Global Configuration
 Options dialog 7
CONNX Data Dictionary Manager 7, 27
CONNX Data Dictionary Manager window 27
CONNX Data Dictionary Selection dialog 7
CONNX Data Synchronization ... 1, 2, 4, 5, 7, 27, 29, 35, 47, 52, 142, 147, 149, 159, 170, 180, 186
CONNX Data Synchronization Administrator ... 7
CONNX Data Synchronization Tool.. 1, 2, 4, 5, 7, 27, 29, 35, 52, 142, 149, 159, 180, 186
CONNX Data Synchronize 42
CONNX Database Logon Selection dialog 7
CONNX Installation dialog 7

CONNX Installation Guide	7	DataSync	3, 7, 37
CONNX Integrated	7	Earlier version	3
CONNX Integrated Logon dialog	7	Date	7, 38, 52, 149, 153, 156, 157, 159, 160
CONNX ODBC Adapter	27	Date Selection.....	153
CONNX Security	7	date/time	38
CONNX Solutions	186	Day.....	159
CONNX Store.....	4	Thanksgiving	159
CONNX User Reference Online Help	7	DB Sync.....	166, 167
CONNX32	7	DB Sync Failure	168
CONNXDataSync.....	7	DB Sync Task	168
CONNXDataSync.Groups.....	183	DB Sync Task started	166, 167
CONNXDataSync.GroupSynchronizations	183	Db_store	177
CONNXDataSync.TableList.....	183	DBA.....	27
CONNXDataSync.TableSynchronizations	183	Decimal precision.....	45
CONNXSchedule	7, 149	target columns.....	45
CONNXStore.....	7, 34, 181	Default Access	7
CONNXStore Database Server	7	Delete.....	29, 141, 163
Control.....	163	Delete button.....	156
Control Panel	7	Delete From	29
CPU.....	2, 7, 163	DeletedRowCount.....	7
Data Sync	7	Demand	147
CRCs.....	4, 7	Desc.....	183
Create.....	7, 33, 37, 48, 62	Designing	67, 75
non-unique index.....	48	Free	75
Create index.....	33	Transform	67
Create target tables.....	37	Detected.....	48
Creates datamart	4	Disable	39
Custom button.....	7	Disk Cost button.....	7
D		Drop	7, 29
Dat.....	142	select.....	7
Data Dictionary.....	7	Drop/recreate	29
Data Sync.....	7, 149	Duplicate data	48
CPU	7	Duplicate keys.....	182
Data Sync scheduling	149	E	
Data Synchronization	7, 52	Editing	132
Data Synchronization process	52	Transforms	132
Data Warehouse	1, 4, 7	e-mail	4, 27, 172, 174, 175, 179, 186
Database Modules window	7	Enterprise-wide	186
DatabaseCatalog	7	EpSendMail.....	172
DatabaseOwner	7	EPSendMail from@email.com.....	172
Datamart.....	4, 7	Error message provides.....	182
Datapump.exe.....	163	Error messages.....	182, 183

- Establish..... 159
- ETL.....4, 7
 - calling.....4
- Everyone7
- Executed AfterSyncSuccess Task 178
- Executed BeforeSync Task..... 177
- Exit7
- Exit button7
- Expression Builder 134
- Extract, Transform.....4
- F**
- File Extension..... 142
- File menu.....7
- Force49
 - Select.....49
- Free75
 - Designing.....75
- FROM..... 183
- From SQL Transform75
- FTP.....4
- Full Reload 4, 7, 27, 29, 52, 147
- Full Reload Threshold text box7, 27
- Full Synchronization Behavior7
- G**
- GB Free Space2
- GB Hard Drive2
- GB RAM2
- GHz2
- Global... 165, 171, 172, 174, 175, 176, 177, 178, 179
- Global On Demand 174, 175, 176, 177, 178, 179
- Global On Demand Synchronization Before Synch Task..... 172
- Global On-Demand Synchronization Task.... 176, 177, 178, 179
- Global Scheduled..... 165, 171, 172, 174, 175
- Global Scheduled Synchronization Task 170, 171, 172, 174, 175
- Global tasks5, 27
- Groups..... 4, 5, 7, 142, 147, 149, 165, 166, 167, 168, 183, 186
 - synchronize 147
- Groups tab5, 142
- Groups tab enables 5
- Groups take 183
- H**
- Healthcare..... 186
- history 180, 183
- Holidays 159
- Hours 149, 156, 183
- I**
- IBM DB2.....27
- ID 7
- IDE 2
- Import CDD dialog 7
- Inc 186
- Include System Tables 7
- Incremental update 4, 7, 27, 52, 147, 149
- Independence Day..... 159
- Index Mapping 96
- INDIRECT 186
- InfoNaut 182, 186
- Informix 27
- InsertedRowCount 7
- Installing..... 7
 - CONNX Data..... 7
- INTEGER 7
- Internet..... 186
- Introduction 1
- J**
- JOHN DOE 182
- July..... 159
- K**
- Key during..... 48
- keys..... 7, 48, 182
- L**
- Labor Day 159
- LAN 2
- LastDeletedCount 52
- LastInsertedCount..... 52
- LastRowCount 52
- LastStatusCode 52
- LastStatusDescription 52
- LastSyncEnd..... 52

LastSynchEnd	7	Next button.....	7
LastSynchStart.....	7	Non	48
LastSyncStart.....	52	November	159
LastSyncType	52	NT Service	7
LastUpdatedCount	52	Null-terminated.....	7
Legal.....	186	O	
Level.....	7	On Failure	168, 175, 179
Max Worker Threads	7	On Failure Task	27, 168, 175, 179
Listener process	183	On Success Task.....	27, 167, 168, 174, 178
Load	4	On-Demand	5, 165, 176, 177, 178, 179
Log Entry	177, 178	Only.....	37, 142
logical	7, 35, 163	Open dialog.....	7
Logical Update Verification Analysis....	7, 35, 163	Operations	175
Logon dialog.....	7	Oracle	7, 27, 49
Long	44	Oracle table.....	7
M		ORDER BY timestampdiff.....	183
Machine1/username.....	7	Override Default.....	142
Maintenance Contract.....	186	P	
Manual.....	130	passthrough	7
Performing.....	130	PC	7, 27
Max Worker Threads.....	7, 27	Per	165
level	7	Perform verification analysis on updates.....	35
Maximum number	45	performance.....	7, 27, 29, 35, 180, 181
select	45	Performing	130
MB RAM.....	2	Manual.....	130
MB/sec	2	Place multiple tables into	142
Memorial Day	159	Plus	153
Menu bar	7	Plus button	156
File.....	7	PM.....	177, 178
Security.....	7	Point.....	7
Select Tools.....	7	Settings	7
metadata	49, 186	PostgreSQL	49
MHz.....	2	post-synchronization	5
Microsoft.....	7	pre-installation.....	7
Microsoft Web site.....	7	Preserve schema.....	29
MostRecentFlag	7	Proceed.....	7
MySQL	27	Step	7
N		Process	163
NET	7	Processing Estimates	52, 147
NET Framework button	7	Processing Estimates dialog.....	29
New Years Day	159	Product Functions.....	4
Next.....	7	Summary.....	4

- Programs.....7, 27
- Programs group7
- Properties.....5, 142, 160
- Purge Method dialog.....29
- Purge target tables.....29
- Purging.....7, 29, 180
 - Select.....7, 29
- purging history.....180
- Purging History feature180
- Q**
- QueryBuilder67
- R**
- Rdb.....27
- Read Only7
- read/write.....186
- Recreate.....7, 36
 - Check.....7
- Recreate indexes36
- Recurrence Pattern149, 160
- Redmond.....186
- Refresh button.....52
- Resynchronization.....142
 - scheduling142
- Resyncing.....4
- Retry failed synchronizations47
- Retry option46
- ReturnStatus177, 178
- RMS7, 27, 142, 182
- Run.....158
- S**
- Save7
- Save icon.....156
- Schedule dialog.....149
 - Select Item.....149
- Schedule Name.....149
- Scheduled Synchronization Tasks..27, 165, 171, 172, 174, 175
- Scheduled Synchronizations.... 27, 52, 158, 160, 165, 166, 167, 168, 170, 171, 172, 174, 175
- Scheduled Tasks.....170
- Schedules. 5, 142, 147, 149, 159, 160, 165, 170, 183
 - Colliding.....183
 - resynchronization142
 - Select149
- ScheduleService.exe163
- Schronization Recurrence Frequency dialog. 158
- SCSI.....2
- Security7
 - menu bar7
- SELECT7, 27, 29, 38, 42, 49, 149, 157, 183
 - active range.....157
 - Add.....38
 - Drop.....7
 - Force49
 - Keep49
 - Purge.....7, 29
 - Schedule149
 - Settings27
 - Synchronize.....42
 - This item.....157
- Select Components window7
- Select Data Source7
- Select Days dialog153
- Select Item149
 - Schedule dialog.....149
- Select Tools7
 - menu bar7
- Selecting27, 45, 52, 65, 77, 153, 156, 157
 - maximum number45
 - Single65
 - Target Destination77
- Separate schemas.....37
- Servename177, 178
- Service7, 149, 186
- Service Pack.....186
- Services7, 149, 186
- Set.....7, 22, 34, 40, 149, 160, 166, 167
- Set Service Login dialog7
- Set transaction batch size.....40
- Setting Tab sub.....22
- Settings5, 7, 27, 34, 38, 42
 - Point7
 - Select27

Signal	158	Synchronization Recurrence Frequency window	165, 166, 167, 168
appropriateness	158	Synchronization Recurrence Frequency dialog	156, 165, 167, 168
Simplified Data Access	186	Synchronization succeeded.....	52
Single	65	Synchronization task.....	27, 149, 156, 157, 165, 166, 167, 168, 172, 174, 175, 176, 177, 178, 179, 183
Selecting.....	65	Synchronization Type Options.....	8
Single group	142	Synchronization using Timestamp.....	79
place multiple tables into	142	Synchronizations take.....	183
SMTP	166, 167, 168, 172, 174, 175	Synchronize	147
SMTP Mail.....	50	group	147
Smtip Server	172	Synchronize blobs.....	42
Solutions.....	186	Synchronize dialog.....	29, 52, 147
SolutionsIQ.....	186	Synchronizing	42, 52, 147
Sort buffer.....	7	select	42
SQL	37, 49, 75	SynchType	7
SQL Server	4, 7, 27, 37, 182	SyncType	52
SQL_TSI_SECOND	183	System Requirements	2
Standards-based.....	186	T	
Start..	5, 7, 52, 56, 147, 149, 166, 167, 172, 176, 183, 185	Table Indexes	7
OK button	52, 147	Table population	33
Status	4, 5, 7, 29	TableID	7
Status Description	29	TableName	7
StatusCode	7, 52	Tables	1, 2, 4, 5, 7, 27, 29, 33, 37, 38, 44, 46, 48, 49, 52, 65, 142, 147, 149, 165, 166, 167, 168, 182, 183
StatusDescription	7, 52	TableSynchronizations	7
Store.....	7, 27, 34, 42, 178	Tabs	22
Store Transaction Log on Target	7, 34	Target columns	45
Store transaction logs on target databases	34	decimal precision	45
Stores copies	34	Target database.....	7, 27, 29, 37, 52, 142, 177
Substitute	166, 167, 168	Target Destination.....	77
TCP/IP	166, 167, 168	Selecting.....	77
Summary	4	Target metadata case.....	49
SUPPORT.....	186	Target Tabe	44
Support@connx.com.....	186	Target tables	36, 38, 44
Sybase	27	column.....	38
Sync Now button	29, 52, 147	Task Started.....	172
Synchronization..	1, 2, 4, 5, 7, 27, 29, 38, 42, 44, 47, 52, 130, 147, 149, 153, 156, 157, 158, 159, 160, 163, 165, 166, 167, 168, 170, 172, 174, 175, 176, 177, 178, 179, 182, 183, 185, 186	Task Succeeded	174, 175
view.....	163	TCP/IP	166, 167, 168, 172, 174, 175
Synchronization Recurrence Frequency dialog	149, 157, 159, 160	enter	172, 174, 175

Substitute.....	166, 167, 168	Unique index.....	48
Technical Support	186	University	186
Technical Support E-mail.....	186	1994-6 Regents.....	186
Telecommunications	186	UpdatedRowCount	7
Thanksgiving	159	UPDATES	186
Day	159	USA.....	177, 178, 186
Time ... 4, 7, 38, 46, 52, 147, 149, 153, 156, 157, 158, 159, 160, 163, 180		Usage.....	172
Time estimate.....	52	Use.....	166
Time Selection	149, 153, 156, 157, 159, 160	Before Synch.....	166
Timestampdiff.....	183	Username	177, 178
Toll Free	186	V	
Trademark.....	186	VARCHAR	7, 44
Transform.....	60, 62, 65, 67, 132, 141	View	160, 163
Designing.....	67	synchronization	163
Editing.....	132	VSAM.....	7, 142
Transformations	4	W	
Treat CHAR.....	7, 43, 44	WA	186
Treat CHAR data types	43, 44	WAN.....	2
Trigger	4, 29, 153, 156, 158, 159, 160	Washington	186
Trigger Files	158	WHERE.....	183
Troubleshooting	180, 181, 182, 183	Win32 synchronization.....	183
Truncate	29	Windows 2000	2
U		Windows NT.....	2
Under Days	153	Windows NT Advanced Server.....	2
Under Full Synchronization Behavior.....	29	Windows Task Manager window	163
Under Months.....	153	Workflow	178
Uninstall.....	7	Y	
		Yes button.....	7, 149
