

CONNX InstantdbSync

Table Of Contents

Preface	1
Introduction	1
The What, Why, and How of Event Replication	3
InstantdbSync Architecture	6
Replication Overviews.....	8
Chapter 1 - Installation	10
System Requirements.....	10
Windows	10
Linux	10
Solaris.....	10
HPUX.....	10
AIX.....	11
Preparing a SQL Server database as a source	12
Preparing a MySQL database as a source	13
Preparing Oracle as a source	14
Installing InstantdbSync on a Windows Server.....	16
Transferring InstantdbSync components to Unix.....	19
Installing InstantdbSync on a UNIX	21
Default Port Numbers and Firewall Considerations	24
Replication Components	27
Chapter 2 - Using the Event Replicator	29
Creating a CDD for Event Replication	29
Importing Source Tables from SQL Server	30
Importing Source Tables from MySQL	33
Set the CDD Security	36
Enabling a CDD for Event Replication	39
Adding a Replication	45
Mapping columns to a new Target Table.....	49
Adding New Column Mappings.....	51
Deleting Column Mappings from a New Target Table.....	55
Changing the Column Order in a New Target Table	58
Changing Source Column Mapping	60
Changing Target Column Mapping	63
Clearing Source Column Mapping	67
Clearing Target Column Mapping	69
Importing and Exporting Column Mappings.....	71
Deleting a Replication	75
Viewing the Source Table Schema.....	76
Configuring Replication and Source Database Servers	80

Viewing Server Status.....	83
Building New Target Database Tables	86
Dropping and Recreating Target Database Tables	88
Deploying Replications.....	89
Performing an Initial State	91
Suppressing an Initial State	93
Un-Deploying all Replications	96
Stopping Event Replication	98
Restarting Event Replication.....	101
Starting, Stopping and Getting Status from the Event Server	104
Chapter 3 - Event Replicator Menus and Tabs.....	105
File Menu.....	105
Edit Menu	106
View Menu.....	110
Tables Menu	115
Servers Menu	118
Help Menu	120
Replication Design Tab	121
Deployed Replications Tab	123
Server Status Tab	125
Chapter 4 - Event Replicator Functions	129
Mapping SQL Expressions to Target Column.....	129
Adjusting Display Column Width	130
Event Replication Registry Settings.....	134
Registry Variables	134
Unix/Linux Environment Variables	134
A2A specific Unix/Linux Environment Variables	135
Registry Variables.....	135
Unix/Linux Environment Variables.....	141
A2A specific Unix/Linux Environment Variables.....	143
Data Filtering.....	144
Validating Active Replications.....	145
Starting the UNIX Message Queue.....	146
Stopping the UNIX Message Queue.....	147
Starting the UNIX Replication Controller.....	148
Stopping the UNIX Replication Controller.....	149
Using the repcmd command line utility	150
Chapter 5 - ACD Replications	153
Introduction to ACD Replications	153
ACD Target Table Metadata	153
Creating ACD Replications	155
ACD Output.....	157

Appendix A - Troubleshooting Guide	160
Event Replication Troubleshooting Suggestions	160
Server Status	160
Log Files	161
Event Replication Error Messages.....	162
General Messages	163
Admin.....	165
Controller	169
Event Consumer.....	179
Event Producer.....	186
Debug	194
Event Replicator User Exits	196
Event Replication Troubleshooting Do's and Don'ts.....	197
Do's.....	197
Don'ts.....	197
Index.....	199

Preface

Introduction

InstantdbSync is a real-time data replication/synchronization and change data capture solution that instantly propagates transactional changes from a source database to a target database. InstantdbSync monitors the source database in real-time for events that affect data in the source table, such as updates, inserts and deletes, and instantly propagates the changes to the target table in real-time, as the events occur.

InstantdbSync is a best of breed, enterprise application, which provides an easy-to-use graphical interface to deploy transactional replication with sub second speed. The GUI interface make creation and configuration of the replication jobs from the source to target database servers simple and easy.

The Business Case

In today's world of increasing data volumes and Big Data, we are seeing databases that are designed to store and process hundreds or millions of fields, handle large volumes of data, and manage high-speed transactions, all while they house the critical business data needs of today's fast-paced companies. With the databases of today, applications are able to save and retrieve complex data structures in a single physical record and in a single operation. Today's databases have significant transactional performance advantages over traditional databases, which would require several joins between many tables to accomplish the same task.

While some databases excel in transactional performance, other databases excel in analytical capability. Relational databases have a large selection of analytical and reporting tools available for companies to use. The InstantdbSync solution gives companies the ability to seamlessly move data from a source database to any number of relational, non-relational, Cloud, desktop, or in memory databases for easier access, data reporting, and analysis.

The Options

There are several ways to move relational data to another database:

1. The extract and load approach

Most relational databases have "bulk mode" import tools that will read data from a text or binary file and place it into the database. A program is written to extract the data into a text file (using the same formats as the relational schema) and transfer the text files from the source database system to the target database system. The relational database table schemas are manually created and a database utility is run to load the files into the table(s).

This was the common approach used in the past simply because there was no other way to accomplish the task. But there are several problems with this option:

- a. There are many manual steps involved in the process.
- b. Only a snapshot of the data is moved into the relational database. Each time current data is required in the target database, the process must be repeated.

2. Using the relational database vendors ETL tools

Many relational databases, such as Oracle, SQL Server, and DB2, have GUI-based tools that make it easy to move data from other relational sources into their database using an OLEDB or ODBC driver. SQL Server is an example of a relational database that comes with a set of powerful data management tools. Using SQL Server Integration Services (SSIS) combined with an OLEDB Driver for a given data source, it only takes a few clicks to move a snapshot of the source data into the target SQL Server.

These tools make the extraction and load of the data very easy in comparison to bulk loading data, but there are still problems:

- a. It's a manual process.
- b. It's a data snapshot. The process must be repeated every time current data is needed.

3. The real-time change data capture approach

Change data capture detects changes to data on a source database and replicates those changes to a target database in real-time (sub-second replication). InstantdbSync can replicate data from one a source database to another similar database for backup or disaster recovery purposes. It can also replicate source data to over 100 supported target databases.

There are two advantages to this technology:

- a. "Fire and forget." Once the replication has been established, no further effort or manual steps are required. The replication will continue to run in the background until otherwise directed.
- b. The data in the replicated database or the target database is not a one-time snapshot, but rather a living copy of the data that is kept in real-time synchronization with the source data. This allows transactional applications to continue to benefit from a source database with high-speed performance and opens the door for using SQL-based data analysis and reporting tools against the relational target copy.

The What, Why, and How of Event Replication

What is live data event replication?

There are two mainstream ways to replicate data between databases.

- Take a snapshot of the source database and duplicate the data into the target database. This is usually done in a batch or in an on-demand environment.
- Capture changes Real Time as they happen in the source database and perform the exact same changes on the target database. This is usually done in a live data or transaction-oriented environment.

Live data updates are an evolution of the snapshot idea. We first capture an image of the source data and create target data. Then we watch the source data to see if there are any changes. As changes occur, we capture these changes in real time and perform the same changes we found on the source database against the target database.

Why should we want to perform live data event replication?

- **Live data event replication is the most efficient way to move information as it changes from the source database to the target database.** When we first capture an image of the source database and create the target database the system must read and write every row of data in the entire database. From that point, live data event replication only accesses information in the original database that has been altered. If an organization has tens of millions of rows of data, only a very small fraction of the data typically changes during daily operations. Even the most efficient snapshot operations must still examine every row of data from the source every time they synchronize the source and target databases. Live data event replication consumes less computing resources than the other database replication alternatives.
- **Live data event replication data does not differ from reality in the same way snapshots of data differ.** If we take weekly snapshots of our database system to create our target database, we will have an excellent image of our current business situation in both the target and source databases right after the snapshot is taken. But as time passes, the source database starts to diverge from the last snapshot. Transactions applied to the source database will not show up in our aging target database until we take another snapshot. Eventually, the duplicated data loses any value, so we are forced to update the target database; usually by creating a new snapshot of the source data.

Live data event replication does not suffer from this defect. As transactions are performed on the source database, the information is immediately moved into the target database and the source and target databases stay current.

- **Companies may need multiple database formats for business reasons.** A particular group may need to analyze some facet of the financial picture such as current inventory levels or the status of accounts receivable but the group may not know how to use the source database system. A live image of the source database system in a format that the group knows how to access solves this problem.

Since badly formed queries can negatively impact critical databases, users may not be authorized to perform ad-hoc queries against the source database. Target databases containing current images of all or part of the source database can allow ad-hoc queries without any danger of impacting the performance of the source database.

Is live data event replication always the best solution?

Live data event replication is not appropriate in every situation. The method of choice (to make a snapshot or to use live data event replication) depends on the eventual use of the data.

There are times when it makes sense to use a snapshot of the source database. For month-end reporting we do not want the financial picture to give us new answers every time we query it.

How does InstantdbSync perform live data event replication?

- **Initial State**

To start live data event replication, we must take the source database and create a snapshot that duplicates all of the data at a single point in time. With InstantdbSync, this step is called the "Initial State".

Performing an initial state operation moves a copy of the source database into the target database.

- **Queuing**

After we have created the initial state, we will have an exact image of the database as it looked at a single point of time - the beginning of the snapshot creation step. But between the time when the initial state operation started and the time when it finished, it is possible that some updates have occurred against the source database. We don't want to lose these changes. The updates are stored in a message queue as they occur and at the end of the initial state, we apply the changes to the target database so it is current again. As new changes come in, we add them to the bottom of the message queue. We continue to use the message queue to update the target database. After a change has been applied to the target database it is removed from the queue.

The message queue allows us to preserve the state of the target database, even if the target database is taken off line. We simply add new data changes to the message queue and when the target database is ready to be brought online, we restart the target server and update the target database with the stored changes until the queue is exhausted.

- **ACID**

ACID stands for Atomicity, Consistency, Isolation, and Durability. For a database to be reliable, the system must preserve the ACID of the data transactions.

- Atomicity means that a group of operations is performed as a unit. With InstantdbSync, we process the transactions in the exact time order in which they occurred and have designed the target database process so it preserves the atomicity of the updates.
- Consistency means that our database must be in a fully functional state when the transaction begins and when it ends. The target database system's transactions ensure the system retains database integrity.
- Isolation means that the database changes happen as a group so that outside users of the system do not have to worry about the individual updates. Although moving money from one account to another account requires the money to be removed from the source account before it is added to the target account, an outside observer will see the accounts either in the original state or with both the addition and the subtraction. They will not see just the addition or just the subtraction.

With InstantdbSync, if the transaction in the target database does not succeed, the transaction is retained on the message queue and a problem notification message is generated. Once the problem has been corrected, then the transaction will be replayed. Partial transactions are not committed.

- Durability guarantees (not just promises) that a transaction has been fully and correctly completed and the state of the changed records will not be partly done or somehow undone. InstantdbSync transactions are durable because of the target database's relational properties; uncommitted transactions are retained in a durable format in the message queue until they have been successfully completed.

- **Problem Resolution**

Because the changes have been stored once the transaction completes, we know that the changes are permanent and cannot be lost due to distressing events such as power outage. However complications can occur in the live data event replication process. Two possible problems are:

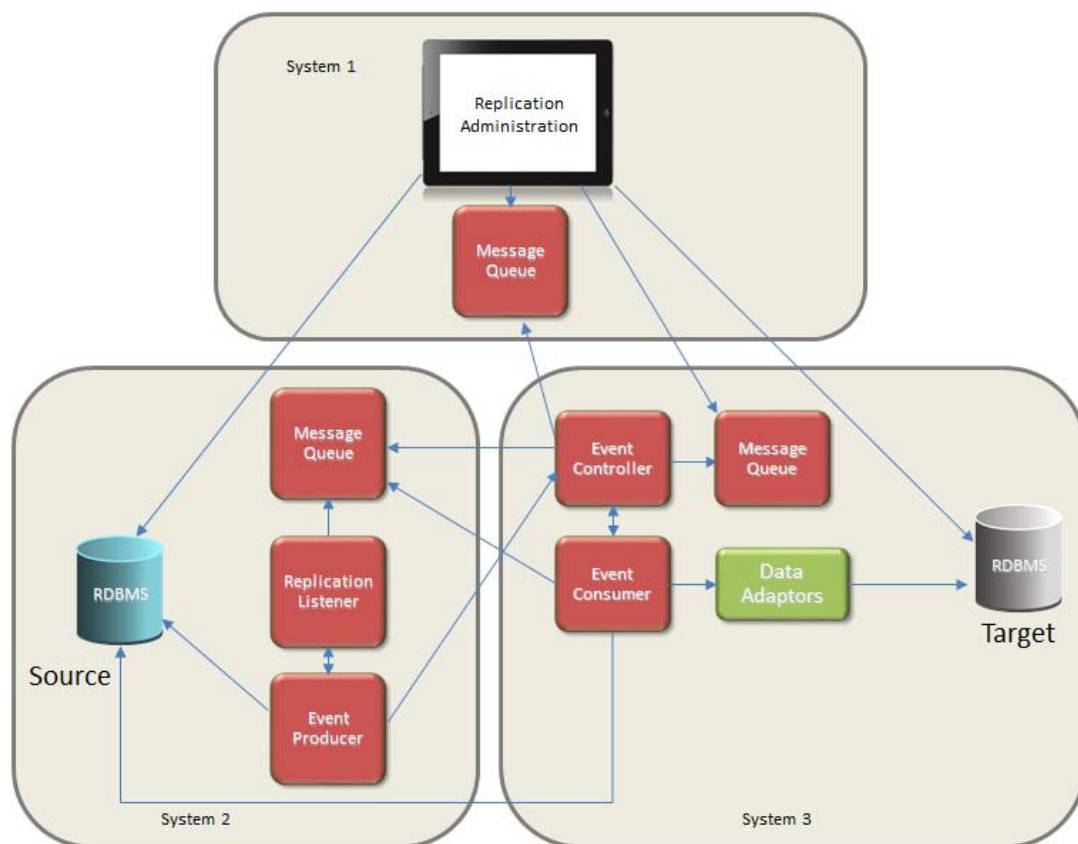
- The source database may be damaged by hardware failure and have to be restored from backup.
- Resources needed to store and forward a transaction may not be available. The system detects this and similar problems and automatically performs an initial state (if required) to create a new snapshot at a feasible time.

Certain kinds of problems can occur that the replication system won't be aware of but the system users will be. For example, data may be accidentally deleted from a target table. Since InstantdbSync only monitors

changes on the source database, but not on the target database, it will not be aware of this problem. However, the system users can simply request a new initial state, and the target system will once again be synchronized with a true picture of the current data.

InstantdbSync Architecture

InstantdbSync Architecture



In the diagram, the replication components are spread across three different servers. In larger environments where there is a heavy transaction load or the source database is consuming all of its server's resources, it is recommended that the replication components be installed on a separate server.

InstantdbSync consists of the following replication components:

1. Replication Administrator

The InstantdbSync Replication Administrator is a graphical user interface that runs in a Windows environment. Using the InstantdbSync Replication Administrator, you can define new replications in less than a minute.

2. Event Producer (EP)

The Event Producer consists of the Replication Listener that spawns Event Producer processes for every Source database/Target database combination of replications present in the current set of deployed

replications. The EP monitors the transaction logs of the source database and every time a change is made to a table that is represented in a deployed replication, the EP tracks that change within the active transaction. Once a transaction is committed, the elements of that transaction are grouped together and placed on the Message Queue, to be picked up by the Event Consumer.

3. Message Queue (CNXMQ)

The CONNX Message Queue stores messages so the system can recover if communication is lost between components. If the Event Consumer or Controller components shut down for any reason (such as power outages or system reboots), the Message Queue ensures full recoverability when the components are brought back online again.

4. Controller

The Controller manages starting and stopping the Event Consumer. It also distributes the replication plan deployed by the InstantdbSync Replication Administrator to the EP and the Event Consumer.

5. Event Consumer (EC)

The Event Consumer is the heart of event replication. The EC reads the transactions placed on the Message Queue by the EP, and using the appropriate CONNX data adaptor, recreates the transaction on the target database.

6. CONNX Client

The EC uses the CONNX Client to perform initial states and communications with the target database. An initial state moves all the data from a source table to a target table for the first time. The CONNX Client is also used for communications between the EP and the source database for purposes of reading the transaction logs.

Replication Overviews

InstantdbSync replicates transactions or Events from a source database to a target database. Source tables can be replicated with the same schema as the source or transformations can be performed using column mapping capabilities found in the Replication Administrator. This process of updating the target database in real-time is referred to as Event Replication. The components that make up InstantdbSync are referred to as the Replication Components.

There are three major stages of replication. The deploy process, the initial state process and the replicating process.

1. The deploy process:
Once the desired source and target tables have been identified and any column mapping done, the replications need to be deployed to the replication controller. During this process, the Controller analyzes the replications and notifies the Event Producer (EP) of which tables are being replicated from the source. It also notifies the Event Consumer (EC) of which target tables the data is destined for. Once the EP and EC have processed the information that was deployed, they will send a status message back to the Controller to indicate success or failure. If the deploy was successful, the Controller will initiate the Initial State process.
2. The Initial State process:
Before replication can begin, the target table must first be in sync with the source table. To achieve this, any previously existing data in the target must be deleted and then a copy of all the rows from the source table must be moved to the target table. This is known as the Initial State process. While an Initial State is running, any changes made to the source database will be held in the message queue and will be processed after a successful completion of the Initial State. Note, if deploy is adding an additional replication or replications to a list of previously deployed replications, only the newly added replications will go through the initial state process.
3. Replication:
After the Initial State is finished, the system enters a state where it is monitoring the source database transaction logs and replicating transactions as they occur.

Replication Overview from the "front end" perspective

To replicate data using the InstantdbSync:

1. Verify that you already have a CONNX Data Dictionary (CDD) containing the file definitions for the source files.
2. Add a link to the desired target database to the data dictionary.
3. Start the InstantdbSync Replicator Administrator and select the CDD.
The first time you open the CDD you will be prompted to enter the location of the replication server components (Controller), and specify any port overrides for communicating to the message queue.
4. Select the source tables to replicate, and the target database.
 - InstantdbSync assumes a one-for-one column mapping between the source and the target tables. Use the InstantdbSync Replication Administrator to change the source column to target column mapping.
 - You can map SQL expressions to target columns. This provides a powerful transformation capability.
5. Add replication filters by using built-in features of the CONNX Data Dictionary Manager. Using the Data Dictionary Manager, a SQL based filter can be applied to any table in the "SQL View Clause" field. For more information on using the SQL View Clause, please see the CONNX User Reference Guide.

6. Press Deploy, and real-time event replication begins.

Replication Overview from the "back end" perspective

When you deploy a replication:

1. The InstantdbSync Replication Administrator uses the "front end" information to create a new replication CDD and sends it to the Controller.
2. The Controller pauses any active Event Consumers (EC).
3. The Controller deploys the new replication CDD in the proper location.
4. After reading the instructions from the new replication CDD, new event filters are sent to the Event Producer (EP) through the Message Queue (MQ). The EP captures only events for the replicated tables, and ignores all other events.
5. The Controller starts up one or more EP and EC engines depending on the work to be done. An EP and EC is started for every source DBID/target database combination in the new replication CDD. If there were any previously active ECs that were paused in step 2, they will be started again at this point.
6. The Controller determines whether an "initial state" is required based on persistent state information stored in a memory mapped file. If an initial state is required, the Controller instructs the EC to start the initial state processing.

Initial states are performed by reading the source data using CONNX, and inserting it into the target database using a standard INSERT/SELECT SQL statement. During initial state processing, changes to the source tables are kept in the Message Queue for future playback after initial state processing is complete.

Note: If a new table was added to an existing EP/EC combination of replications, the new table will have an Initial state performed but the existing tables will not. The replications for the existing tables will be paused during the initial state process. As with the new table(s), any transactions that occur against the existing replications will be queued until the initial state process finishes.

7. After any initial state processing, the EP captures changes to the selected tables and places them on the MQ. The EC reads the MQ and uses CONNX to place changes in the target database. This continues in real time until replication is stopped.

Chapter 1 - Installation

System Requirements

Windows

Required System	Recommended System
Operating System	
Windows 2012 Server/Windows 10 or Higher	Windows 2012 Server/Windows 10 or Higher
Server	
20 GB Free Space	100 GB Free Space
Dual core CPU	Quad core processor or higher
8 GB RAM	16 GB RAM or higher *
Standard hard drive controller	Raid controller or mirrored drive controller
100 MB/sec network card or WAN connection	1 GB network card or LAN connection
Software	
CONNX Version 12 or higher	CONNX Version 12 or higher
When SQL Server is the source database: SQL Server 2008 or above. When MySQL is the source database: MySQL 5.6 or above on Windows or Linux on Intel When Oracle is the source database: Oracle 11g to 18c	When SQL Server is the source database: SQL Server 2008 or above. When MySQL is the source database: MySQL 5.6 or above on Windows or Linux on Intel When Oracle is the source database: Oracle 11g to 18c

* The amount of RAM necessary is dependant on the number of source databases and target databases that are configured for replication. A system with 16GB RAM will allow a configuration of approximately 50 source databases and 50 target databases with approximately 100 tables each. Larger environments may need more RAM or hard drive capacity or may need to have components distributed across more than one server.

Linux

Required System
Operating System
Linux kernel 2.6.18 and above on both Intel and zLinux platforms (examples include RHEL 5 and above or openSUSE 10.2 and above)

Solaris

Required System
Operating System
Oracle Solaris 9 or higher

HPUX

Required System
Operating System
HP-UX 11.0 and above

AIX

Required System
Operating System
IBM AIX Version 6.1 and above

Preparing a SQL Server database as a source

When replicating with SQL Server as the source database, the following prerequisites must be met:

1. The version of SQL Server must be 2008 or later.
2. The recovery model for the database must be Full.
3. At least one full backup must have been performed prior to starting replication. If a full backup has never been performed, the EP will stop and there will be errors in the log explaining what is wrong.
Note: The backup must be done AFTER the recovery model has been set to Full or Bulk-logged. If the recovery model is changed to Simple, it will need to be set back to Full or Bulk-logged and a new backup will need to be done prior to using the database as a replication source.
4. The SQL Server User ID used to connect to the source SQL Server database must be a member of the sysadmin server role.

The following restrictions apply:

1. The sum of all the fields in a table must be less than 8,000 bytes.
2. BLOB/CLOB data types are not supported. This includes TEXT, NTEXT, IMAGE, VARCHAR(MAX), etc.
3. Computed columns have limited support.
 - The index used for replication cannot be a computed column.
 - The source table can have computed columns but the target table cannot.
 - If a computed column is part of the replication, the data in the computed column will not be replicated. In this case, the target column that the computed column is mapped to will receive nulls.

Preparing a MySQL database as a source

When replicating with MySQL as the source database, the following prerequisites must be met:

1. The version of MySQL must be 5.6 or later.
2. Transaction logging must be enabled and the following MySQL values must be set in the my.ini file:
 - a. binlog_format=ROW.
 - b. binlog_row_image=FULL
 - c. log-bin="<basename>-bin" **Note:** basename is typically the name of the server MySQL is installed on.
3. When importing a MySQL data source into a CDD for use with replication, do not use "localhost" as the server name. Instead, use the actual server name or IP address.

The following restrictions apply:

1. BLOB/CLOB data types, JSON, GEOMETRY and ENUM are not supported.
2. MySQL must reside on Windows or Linux.

Preparing Oracle as a source

When replicating with Oracle as the source database, the following prerequisites must be met:

- 1) The version of Oracle must be 11g to 18c. The Oracle logminer is not available in Oracle versions past 18c.
- 2) The Oracle Database-Level Supplemental Logging must be at least Minimal Supplemental Logging. Also, the connection to the logminer must be the same as the Oracle source database, logmining from a remote Oracle Database is not allowed. Supplemental Logging can be activated with the following command:

```
alter database add supplemental log data;
```

To specify the location of the log data, enter the following command:

```
alter system set log_archive_dest_1='location-<path to log file>' scope=both;
```

To see the current location of the logs, enter the following command:

```
archive log list;
```

Once supplemental log data has been enabled, the database must be restarted in archive mode:

```
shutdown immediate;
startup mount;
alter database archivelog;
alter database open;
```

- 3) The Oracle User used to connect to the source Oracle database must have permissions to access the logminer and read all the entries in the V\$TRANSACTION view. The following is an example of a user with the required permissions:
 - a) create role ep_logmnr;
 - b) grant create session, logmining, execute, alter system, execute_catalog_role, select any dictionary, select any transaction, select any table, create tablespace, drop tablespace to ep_logmnr;
 - c) grant select on v_\$logmnr_contents to ep_logmnr;
 - d) grant select on v_\$logmnr_dictionary to ep_logmnr;
 - e) grant select on v_\$logmnr_parameters to ep_logmnr;
 - f) grant select on v_\$logmnr_logs to ep_logmnr;
 - g) grant select on v_\$archived_log to ep_logmnr;
 - h) grant select on V_\$TRANSACTION to ep_logmnr;
 - i) create user epminer identified by epminer;
 - j) grant ep_logmnr to epminer;
 - k) alter user epminer quota unlimited on users;
- 4) If using a multitenant container database (CDB) a common user must be used to access the logminer. This is an example of a common user with the required permissions:
 - a) create role c##cnx_ep_logmnr CONTAINER=ALL;

- b) grant create session, logmining, alter system, execute_catalog_role, select any dictionary, select any transaction, lock any table, select any table, create tablespace, drop tablespace to c##cnx_ep_logmnr CONTAINER=ALL;
- c) grant select on v_\$logmnr_contents to c##cnx_ep_logmnr CONTAINER=ALL;
- d) grant select on v_\$logmnr_dictionary to c##cnx_ep_logmnr CONTAINER=ALL;
- e) grant select on v_\$logmnr_parameters to c##cnx_ep_logmnr CONTAINER=ALL;
- f) grant select on v_\$logmnr_logs to c##cnx_ep_logmnr CONTAINER=ALL;
- g) grant select on v_\$archived_log to c##cnx_ep_logmnr CONTAINER=ALL;
- h) grant select on V_\$TRANSACTION to c##cnx_ep_logmnr CONTAINER=ALL;
- i) create user c##oracleuser4 identified by oracleuser CONTAINER=ALL;
- j) grant c##cnx_ep_logmnr to c##oracleuser4 CONTAINER=ALL;
- k) alter user c##oracleuser4 quota unlimited on users CONTAINER=ALL;

CONTAINER=ALL is syntax for Oracle 12 and above, and is not valid for oracle 11 and below.

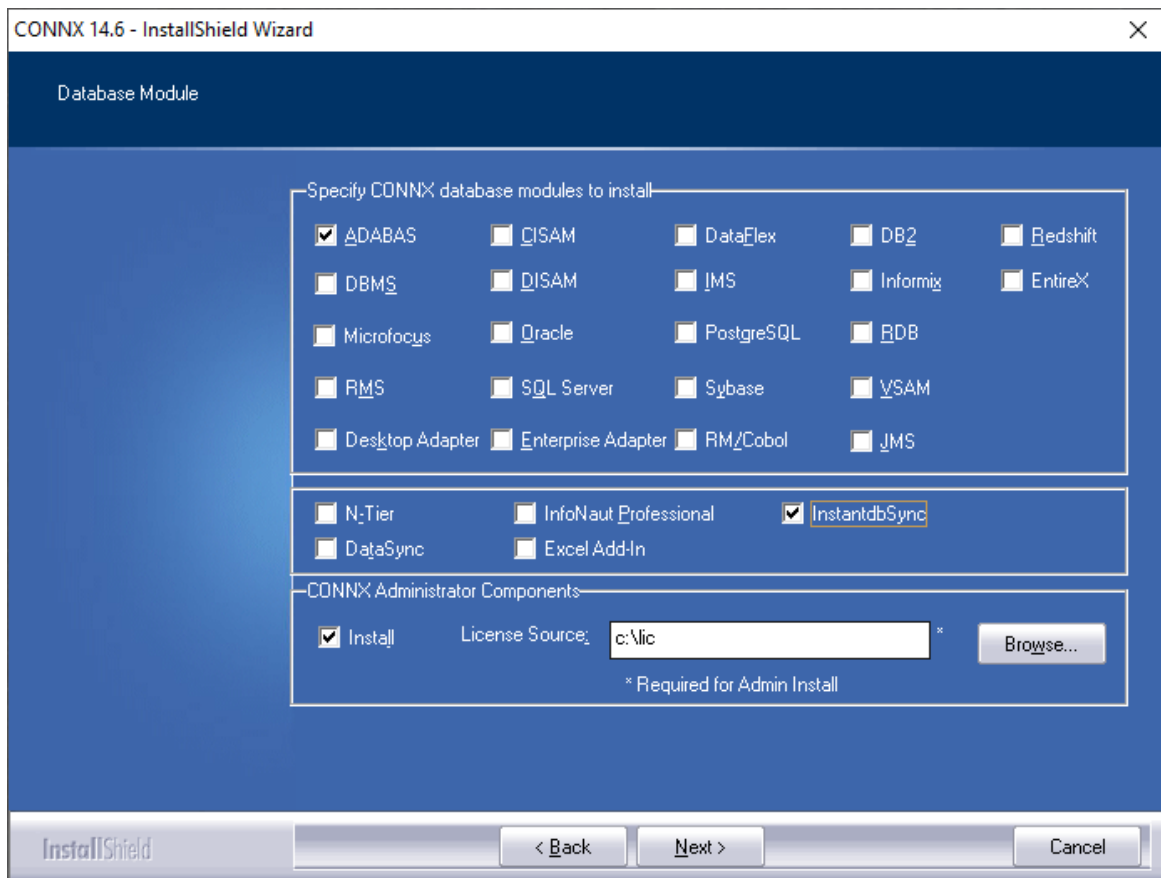
Notice that the role and user name all start with c## - this prefix is required by oracle for common user and role names.

The following restrictions apply:

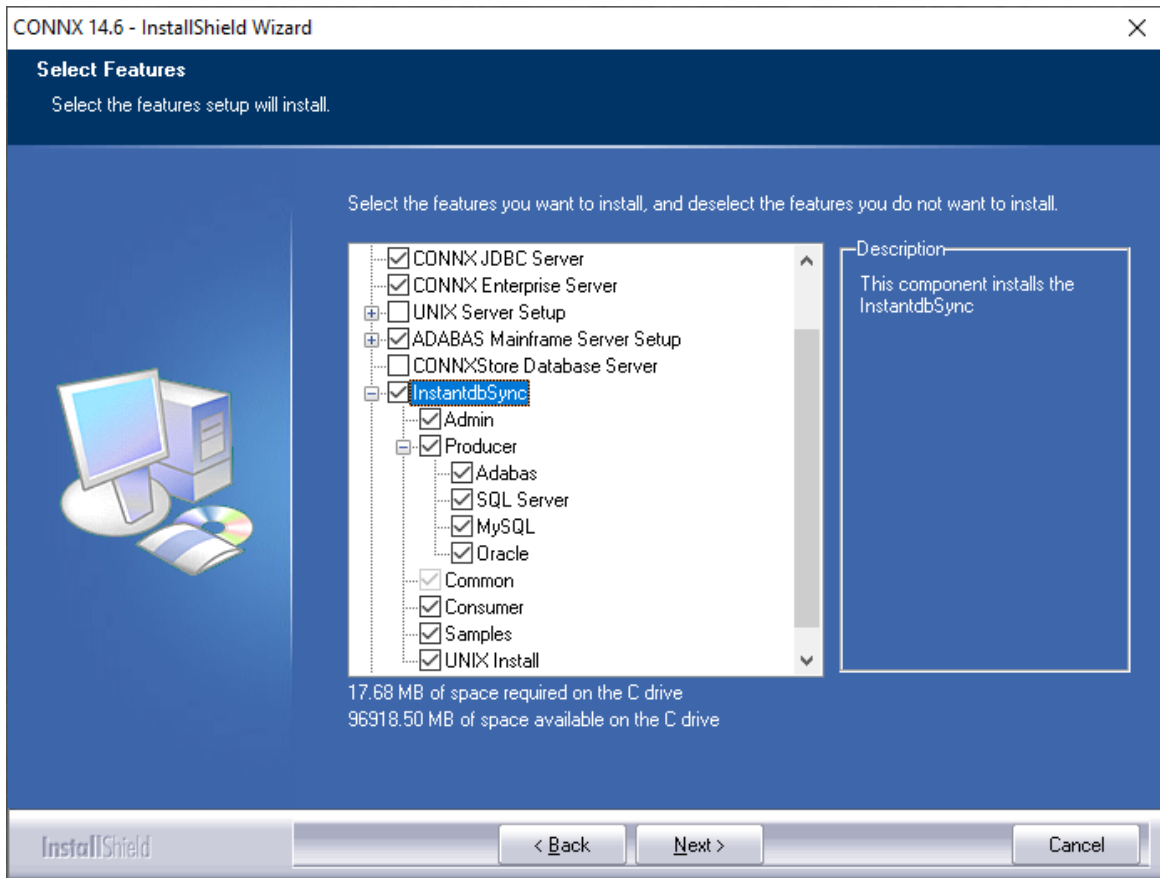
1. BLOB/CLOB data types are not supported. This includes CLOB, NCLOB, BLOB, BFILE, LONG, LONG RAW, etc.
2. Computed columns have limited support.
 - The index used for replication cannot be a computed column.
 - The source table can have computed columns but the target table cannot.
 - If a computed column is part of the replication, the data in the computed column will not be replicated. In this case, the target column that the computed column is mapped to will receive nulls.

Installing InstantdbSync on a Windows Server

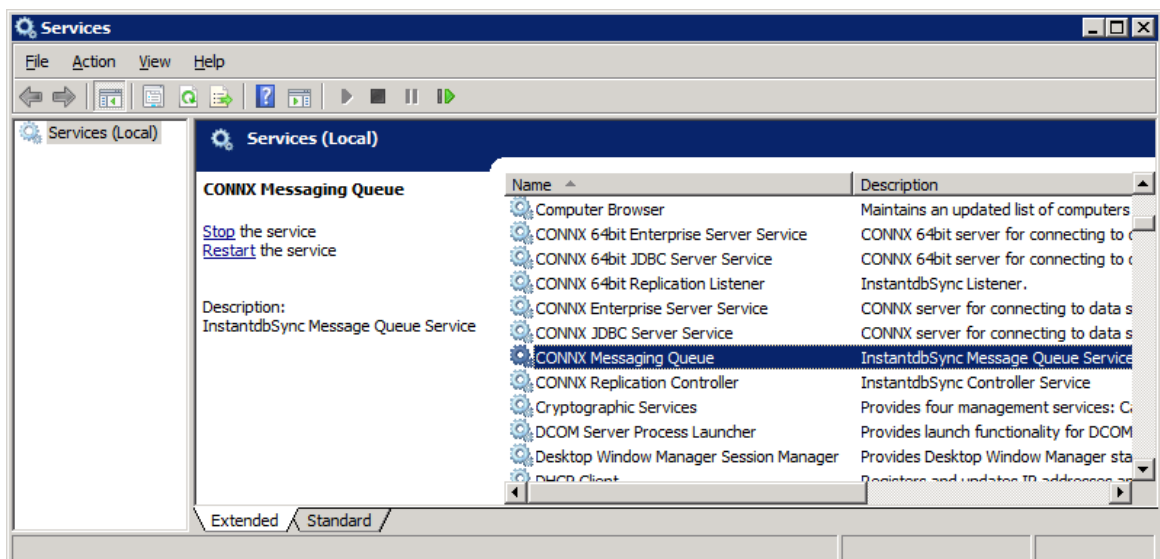
1. InstantdbSync is installed as part of the main CONNX installation package. On the Database Module screen, select InstantdbSync



2. The **Select Features** window appears. You can select the InstantdbSync components you wish to install.
Note: The InstantdbSync Administrator (**Admin**) must be installed on a Windows environment even if the Event Producer or the Controller/Event Consumer are installed on UNIX environment.



- After the installation process completes, restart your machine if asked to do so.
- Click the **Start** button, point to **Settings**, and point to **Control Panel**. Select **Administrative Tools** and then **Services**. The CONNX Messaging Queue, CONNX Replication Controller and CONNX Replication Listener (if the EP was selected for installation on this machine) services should be in the Local Services list and their Server Status should be **Started**.



5. If the Event Producer or the Controller/Event Consumer are on UNIX environments, [install those components](#).
6. If the event producer and controller were installed on different machines, start the message queue on both machines (Windows or UNIX) before deploying a replication. Note: If there is a firewall between the machines, make sure the message queue port (default is 9200) is open.
7. You are now ready to [enable the Event Replicator](#).

Note: The Administration component is a .NET 4.0 application and requires .NET 4.0 to be installed on the system. If the installer detects that .NET 4.0 is not installed, it will prompt you to allow it to run the Microsoft .NET installer before proceeding. If you decline this installation, the CONNX installation will not be allowed to continue..

Transferring InstantdbSync components to Unix

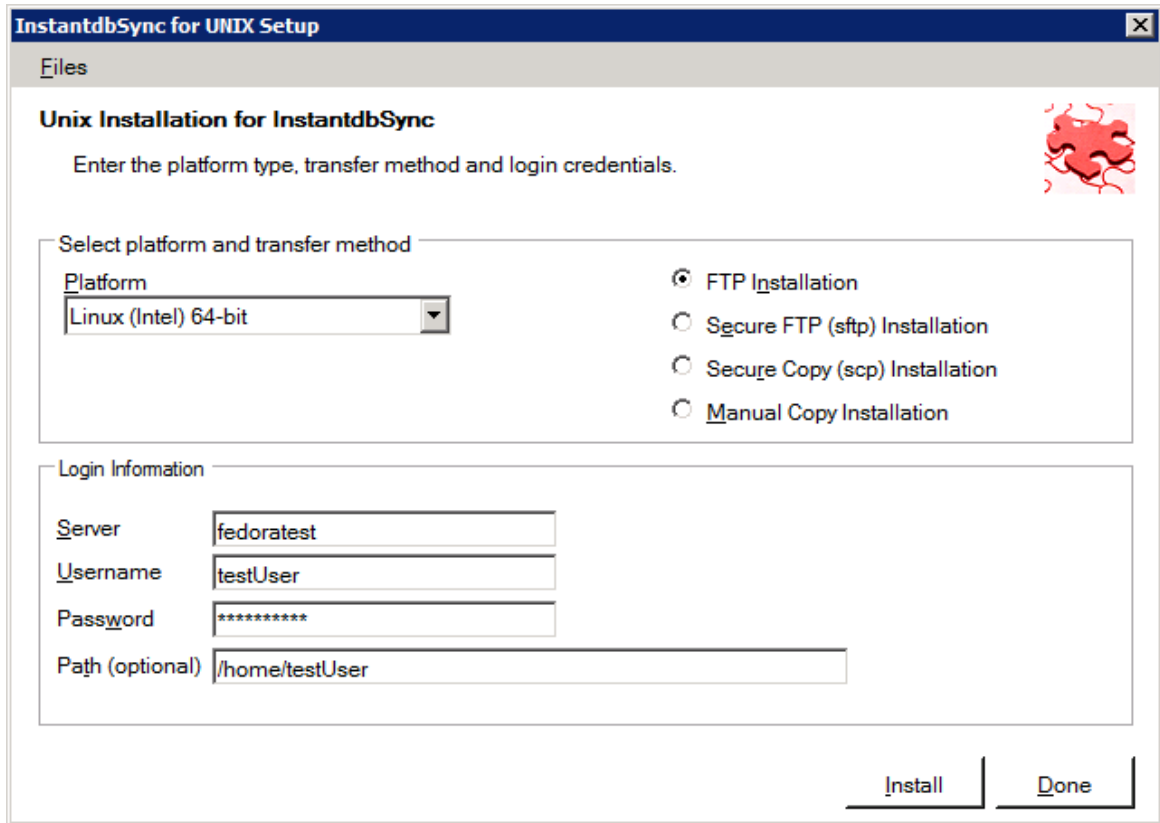
The InstantdbSync Controller and EC can be installed on a Linux or Unix system. The EP must be installed on a Windows system.

Before installing InstantdbSync on UNIX, a [Windows Installation](#) must be completed.

1. On the **Start** menu, click **Programs**, click **CONNX InstantdbSync** and then click **UNIX Installer**. The **InstantdbSync for UNIX Setup** window appears.

2. If the target system has an FTP, SFTP or SCP server enabled, skip to step 4.
3. If the target system does not have an FTP server enabled, select the Manual Copy Installation option and click the Begin Client Installation button. This option will create the necessary install files in the InstantdbSync\INSTALL\UNIX\TEMPINST subdirectory of the CONNX installation directory. Move these files via an alternate copy method to the Unix server and then proceed to the section [Installing InstantdbSync on Unix](#).
4. Select an operating system from the **Platform** list box and the desired transfer method (FTP, SFTP or SCP).
5. In the **Login** area, enter the TCP/IP host name or address for your system platform in **Server**, the user account name in **Username**, and the account password in **Password**. Enter the installation directory in **Path**.

Note: Specifying a Path is optional. If you do not specify a path, the files will be transferred to the default home directory for the user specified in **Username**.



InstantdbSync for UNIX Setup

Files

Unix Installation for InstantdbSync

Enter the platform type, transfer method and login credentials.

Select platform and transfer method

Platform
Linux (Intel) 64-bit

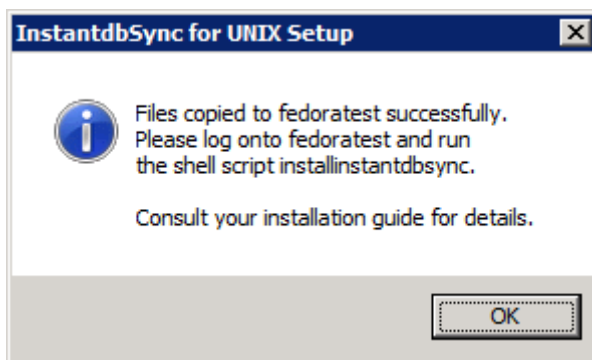
☒ FTP Installation
☐ Secure FTP (sftp) Installation
☐ Secure Copy (scp) Installation
☐ Manual Copy Installation

Login Information

Server: fedoratest
 Username: testUser
 Password: *****
 Path (optional): /home/testUser

Install Done

- Click **Install**. The following message appears after a successful install:



Installing InstantdbSync on a UNIX

In order for the Event Replicator to run on a UNIX system:

- install the InstantdbSync Replicator Administrator on a Windows machine.
 - install the InstantdbSync replication components on both the Windows InstantdbSync Replicator Administrator system and the UNIX system.
 - install the CONNX client component on the UNIX machine.
1. On the Windows InstantdbSync Replication Administrator system, run the Unix Installer from the **Start | Programs | CONNX InstantdbSync** menu.
 2. Open a session in the UNIX installation directory on the UNIX replication machine.
 3. Run the install script `./installreplication`.
 4. You will be prompted for an installation directory.
 5. Select the same install directory as the one containing the CONNX client installation. Press **Enter**. The install script will install the components and start the servers.

```

Telnet solaris8
Uncompressing install media ...

Checking: (replicator/) ...
Checking: (replicator/common/) ...
Checking: (replicator/common/cnxmq) ...
Checking: (replicator/consumer/) ...
Checking: (replicator/consumer/cnxcontrol) ...
Checking: (replicator/consumer/cnxrep) ...
Checking: (replicator/producer/) ...
Checking: (replicator/producer/libcnxepada_64.so) ...
Checking: (replicator/producer/cnxepcfg_64) ...
Checking: (replicator/log/) ...
Checking: (replicator/objedit) ...
REPLICATOR Server not running
CNXMQ Server not running
x replicator, 0 bytes, 0 tape blocks
x replicator/common, 0 bytes, 0 tape blocks
x replicator/common/cnxmq, 1738216 bytes, 3395 tape blocks
x replicator/consumer, 0 bytes, 0 tape blocks
x replicator/consumer/cnxcontrol, 4003128 bytes, 7819 tape blocks
x replicator/consumer/cnxrep, 2339308 bytes, 4569 tape blocks
x replicator/producer, 0 bytes, 0 tape blocks
x replicator/producer/libcnxepada_64.so, 1090760 bytes, 2131 tape blocks
x replicator/producer/cnxepcfg_64, 2439920 bytes, 4766 tape blocks
x replicator/log, 0 bytes, 0 tape blocks
x replicator/objedit, 5780 bytes, 12 tape blocks
replicator/objedit: Done.
replicator/objedit: Done.
replicator/objedit: Done.

Packages uncompressed.

Updating the registry
Registry file is: /export/home/cnxuser/buildverify/connx/connxreg.db
Log files archived to: replicator/log_archive.tar

CNXMQ Server is not running.
CNXMQ Server started successfully:
--> 25026 ./replicator/common/cnxmq 9200 /export/home/cnxuser/connx

REPLICATOR Server is not running.
REPLICATOR Server started successfully:
--> 25042 ./replicator/consumer/cnxcontrol 9205 /export/home/cnxuser/connx

*****
CONNX Event Replicator For UNIX Installation Complete.
*****

IMPORTANT:
-> Define CONNXREGISTRY in your environment ---
Example: export CONNXREGISTRY=/export/home/cnxuser/buildverify/connx/connxr
eg.db
-> Source (export) the cnxep.env into the environment under the account runnin
g
the ADABAS Nucleus
Example: . /export/home/cnxuser/connx/replicator/cnxep.env
/export/home/cnxuser>

```

7. After the installation completes:
 - Define CONNXREGISTRY in your UNIX environment.
 - The message queue and controller are started automatically. If errors occurred, they will need to be corrected and then these components will need to be started. See the sections start the message queue and start the replication controller for information about starting these components.
8. If you are replicating to a SQL Server, Sybase, Oracle, DB2 or Informix target database, follow the special configuration instructions found in **Connecting to SQL Server, Sybase, Oracle, DB2 and Informix from UNIX** in **CONNX Configuration Settings - non-Windows** in the CONNX User Guide.

Warning: If your UNIX target database is not correctly configured on both the InstantdbSync administrative machine and the machine where the controller is installed, you will not be able to replicate to those target tables.

Note: If there is a firewall between the machines involved in replication, make sure the message queue port (default is 9200) is open. This includes the Windows machine where the Replication Administrator is located. For a list of ports used in replication, please see the section titled [Default Port Numbers and Firewall Considerations](#).

You are now ready to [enable the Event Replicator](#).

Default Port Numbers and Firewall Considerations

The following is a list of replication components that use TCP/IP communications and the default ports they use:

Component Name	Default Port
CONNX Message Queue	9200
CONNX Replication Controller	9205
CONNX Data Server (optional depending on target database)	6500
License Server	7501

If CONNX is installed in an environment where a firewall is present, these ports need to be opened.

Because replication is designed to work in a distributed environment, it is likely that components will be installed on different machines. If there is a firewall in between the machines where the different components are installed, the ports for the components need to be open in the firewall.

The following describes each port and when it needs to be open:

Port 9200: This port is used by the CONNX Message Queue and must be open and allowed to accept connections from all machines where replication components are installed. This includes the Replication Administration machine, the EP machine, and the Event Server (controller) machine.

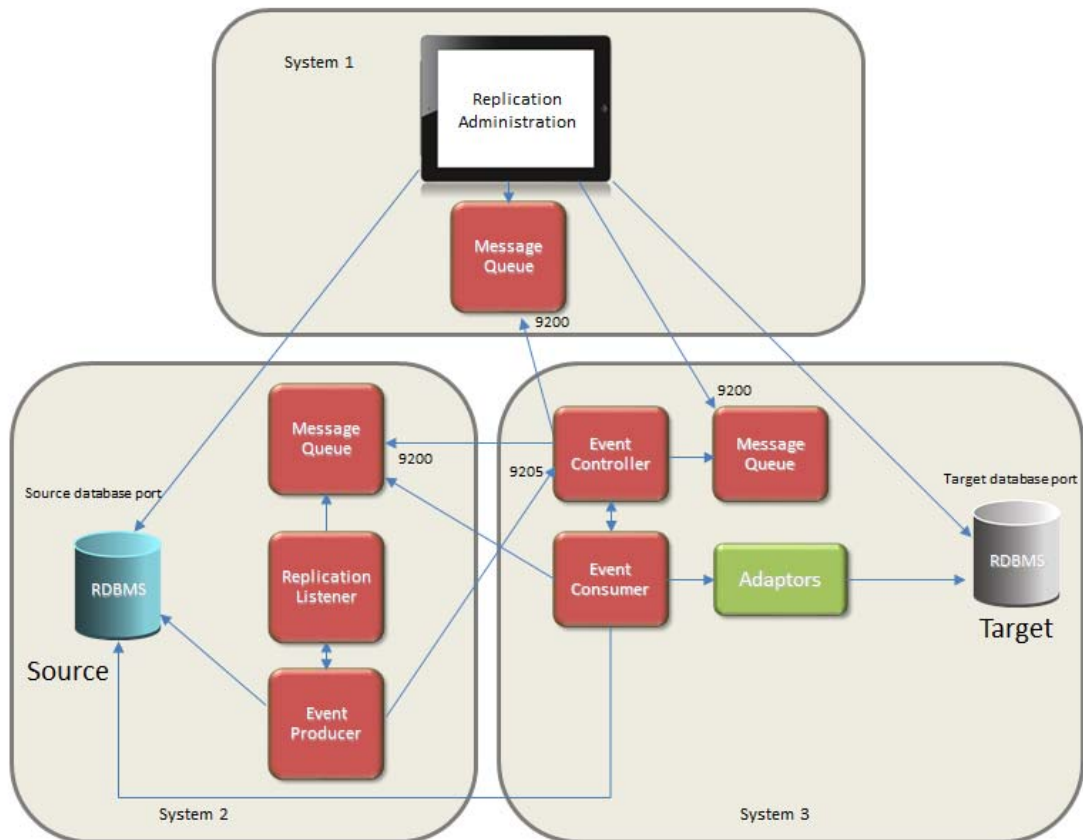
Port 9205: This port is used by the controller to accept connections from the EP. If the EP is on a different machine than the controller, port 9205 on the controller machine must be open and allowed to accept connections from the EP machine. Note, if there are multiple EP machines in the configuration, then port 9205 needs to be able to accept connections from each EP machine.

Port 6500: This port is used by the CONNX listener (data server). While the data server is not a replication component, it may be used by replication depending on the particular configuration.

Port 7501: This port is used by the license server and must be open and allowed to accept connections from any machine where CONNX components are installed. This includes the EP, Controller, Replication Administrator, and any other CONNX clients running in the environment.

The diagram below illustrates a typical configuration with the Replication Administrator installed on System 1, the source database along with the EP installed on System 2 and the Controller along with the target database installed on System 3:

TCP/IP Port assignments



In this example:

- **System 1**
 - must open port 9200 and allow connections from System 3
 - from any source port, it must be able to connect to System 2 on the source database port.
 - from any source port, it must be able to connect to System 3 on port 9200 and the target database port
- **System 2**
 - must open port 9200 and allow connections from System 3
 - must open the source database port and allow connections from System 1 and System 3
 - from any source port, it must be able to connect to System 3 on port 9205
- **System 3**
 - must open port 9200 and allow connections from System 1
 - must open the target database port and allow connections from System 1
 - must open port 9205 and allow connections from System 2
 - from any source port, it must be able to connect to System 1 on port 9200
 - from any source port, it must be able to connect to System 2 on the source database port and port 9200
- **All Systems**
 - from any source port, they must be able to connect to the License Server (not depicted in the above diagram) on port 7501

CONNX InstantdbSync

- License Server (not depicted in the above diagram)
 - must open port 7501 and accept connections from any system where CONNX components are installed.

Note: This example depicts the target database on the same system as the Event Controller (System 3). If the target database is on a different system, the appropriate ports will need to be open on that system.

Note: This example depicts the Replication Listener and Event Producer on the same system as the source database. If the source database is another system, the appropriate ports will need to be open on that system.

Replication Components

The following is a description of the InstantdbSync components:

UNIX/Linux

Component Name	Process Name	Location	Invoked by	Log File	Comments
CONNX Message Queue	cnxmq	connx/replicator/common	mqserver script in the connx directory	connx/replicator/log/cnxmq_run.log	A message queue needs to be running on each system that contains replication components.
CONNX Replication Controller	cnxcontrol	connx/replicator/consumer	eventserver script in the connx directory	connx/replicator/log/CTRL.log	
CONNX Event Consumer (EC)	cnxrep	connx/replicator/consumer	Controller (cnxcontrol)	connx/replicator/log/EC_<source>_<target>.log	Used for moving transactions from the message queue to the target database.

Windows

Component Name	Process Name	Location	Invoked by	Log File	Comments
CONNX Message Queue	cnxmq.exe	c:\connx32\replication\common	Windows Service	c:\connx32\replication\log\cnxmq_run.log	A message queue needs to be running on each system that contains replication components.
CONNX Replication Controller	cnxcontrol.exe	c:\connx32\replication\consumer	Windows Service	c:\connx32\replication\log\CTRL.log	

CONNX Event Consumer (EC)	cnxrep.exe	c:\connx32\replication\consumer	cnxcontrol.exe	c:\connx32\replication\log\EC_<source>_<target>.log	Used only with replications that have a relational target. Is not used with Adabas to Adabas replication. There will be one cnxrep process running for each source database/target database combination.
CONNX Event Producer (EP)	cnxepsqlserver.exe	c:\connx32\replication\producer	cnxeprun.exe	c:\connx32\replication\log\EP_<dbid>.log	The EP is started by the EP listener. There is one EP process for every source/target database combination.
CONNX Event Producer Listener	cnxeprun.exe	c:\connx32\replication\producer	Windows Service	c:\connx32\replication\log\CNxEPLISTENER.log	..

Chapter 2 - Using the Event Replicator

Creating a CDD for Event Replication

Creating a CDD for Event Replication is the first step in getting data replicated. This includes:

- Opening the CONNX Data Dictionary (CDD) Manager
- Importing tables from the source database
- Importing the target database
- Setting the correct CDD security
- Saving the CDD

Create a CDD for Event Replication if:

- You do not have an existing CDD containing the source tables and target database needed for event replication.

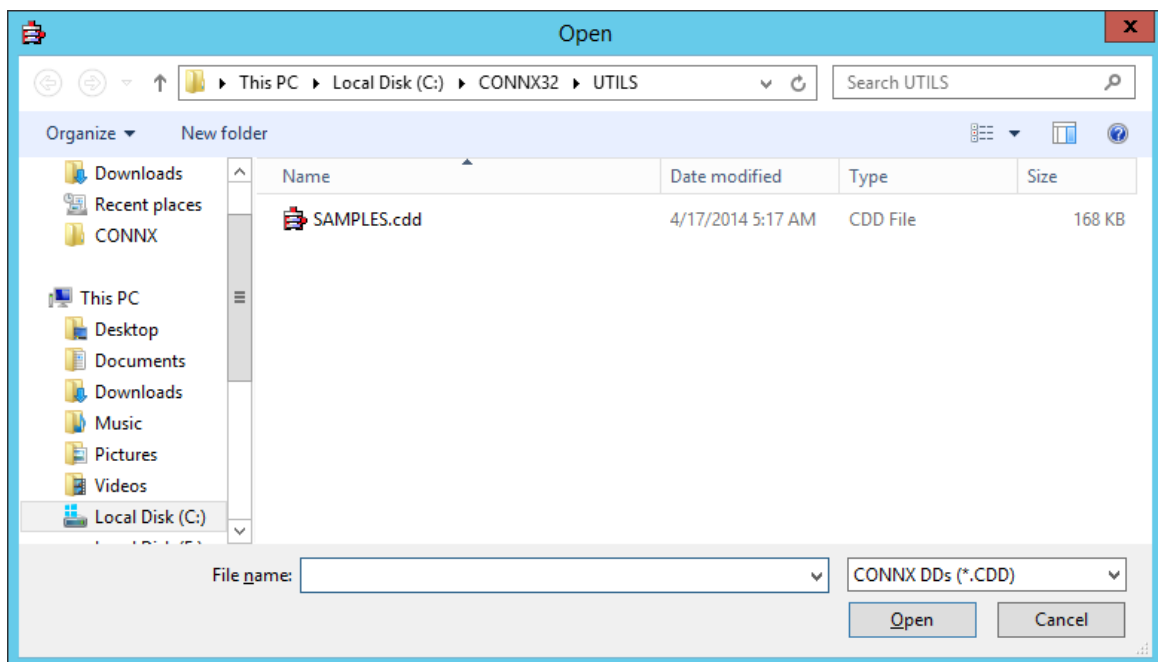
After you create a CDD for Event Replication you will be ready to:

- Enable the Event Replicator

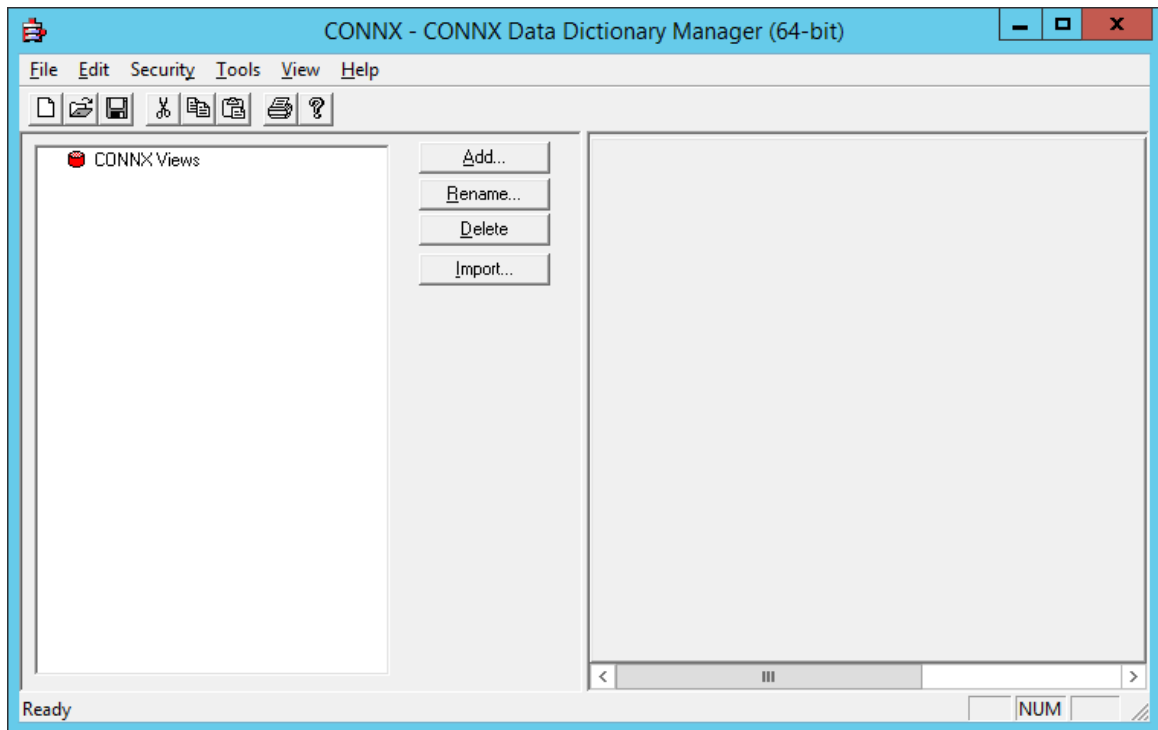
InstantdbSync needs a CONNX Data Dictionary (CDD) to replicate. Create a new CDD if you do not have an existing CDD that contains both the source tables to be replicated and the target database.

Note: For this example, the source database will be SQL Server and the target database will be Oracle

1. On the **Start** menu, click **Programs**, click **CONNX Driver** and then click **CONNX Data Dictionary**. The **Open** window appears.



2. Click **Cancel**. The **CONNX Data Dictionary Manager** window appears.



Importing Source Tables from SQL Server

1. Click **Import**. The **Import CDD** window appears.

Import CDD

The Import feature downloads your existing database record structures and stores them in the encrypted CONNX Data Dictionary for use by CONNX.

OK **Cancel**

Import Type: **SQL Server**

Press the Select Provider or DSN button to link the 3rd party datasource:

☐ Include System Tables

☒ Get Statistics

Logon Information

Select Provider Type

☒ OLEDB Provider ☐ ODBC Provider

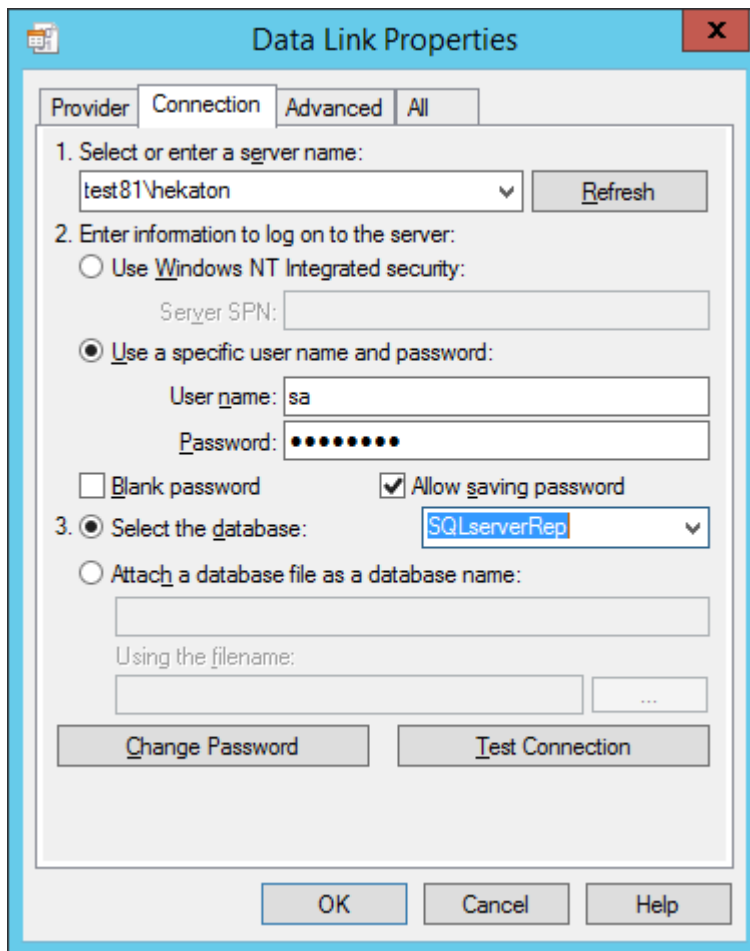
Provider: **Select Provider...**

UserName:

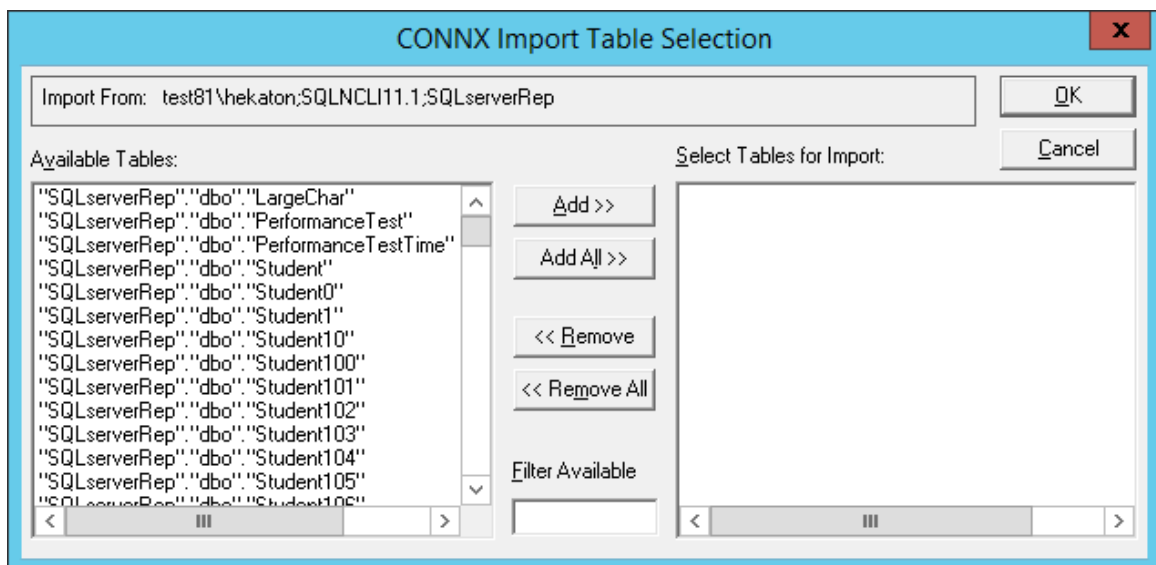
Password:

2. Select OLEDB Provider and then click the Select Provider button. Fill in the SQL Server Data Link Properties..

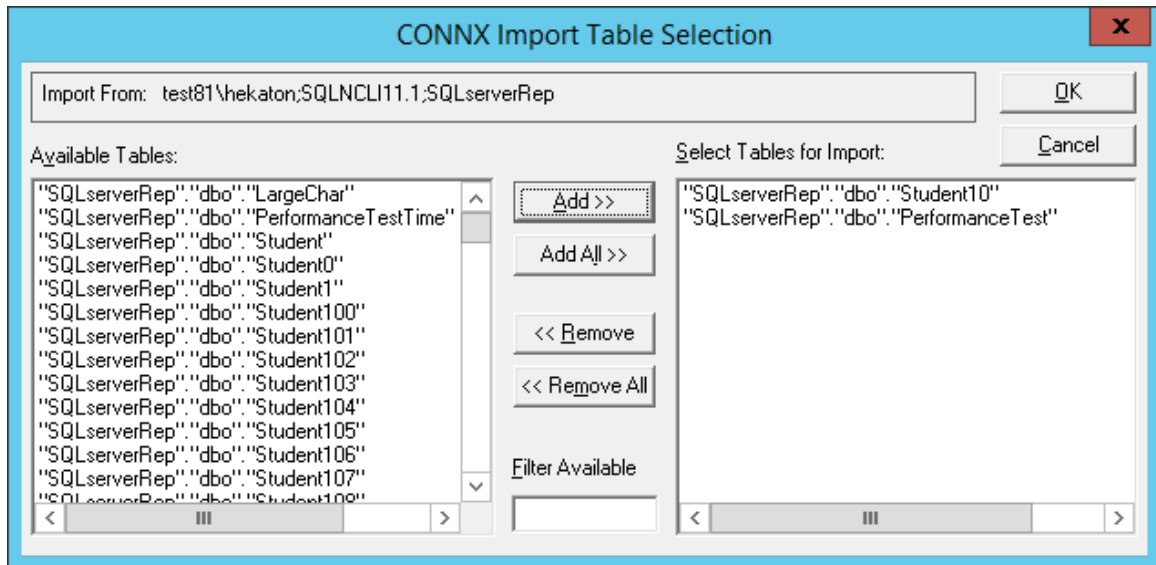
Note: If "Use a specific user name and password" is selected, be sure to also check "Allow saving password". If this is not checked, the password will not be passed back to the CDD Manager and it will have to be re-entered prior to completing the import.



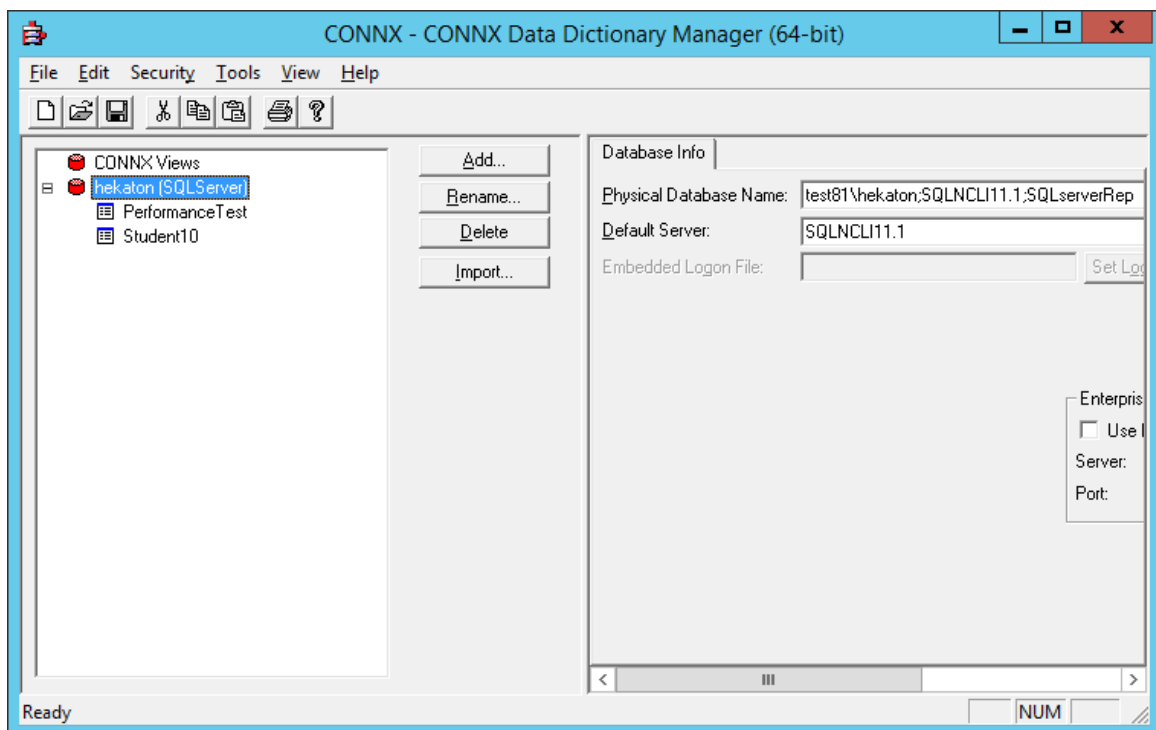
Click **OK**. The **CONNX Import Table Selection** window appears.



3. Add the source tables to be replicated to the target database and click **OK**.



4. The **CONNX Data Dictionary Manager** window containing the source table information appears.



You have imported the source tables into the CDD.

Importing Source Tables from MySQL

1. Click **Import**. The **Import CDD** window appears.

The Import feature downloads your existing database record structures and stores them in the encrypted CONNX Data Dictionary for use by CONNX .

OK Cancel

Import Type: MySQL

Enter Parameters below or Click Advanced for additional options

☐ Include System Tables

☒ Get Statistics

Logon Information

Server: [] Advanced

UserName: []

Password: []

Port: 3306

Catalog: [] Select Catalog...

2. Select **MySQL** in the Import Type drop down box. Do not use the Enterprise Adapter for the import method.
3. Fill in values for Server, UserName and Password. The Port is set to the default value of 3306. If MySQL is configured to use a different port, this value will need to be changed to match the port MySQL is configured to use.
4. Press the Select Catalog button. This will bring up a list of the available catalogs. Select one and press OK. Alternatively, pressing the Advanced button will allow the specification of all MySQL driver settings including the catalog. If Select Catalog is used, the rest of the MySQL driver settings will be the default settings.
5. Press OK on the Import CDD dialog. The **CONNX Import Table Selection** window appears.
6. Follow steps 3 and 4 in the above section on importing from SQL Server to select the desired tables.

Importing the Target Database

This example illustrates importing from an Oracle database as the replication target. If the target database is a SQL Server database, follow the [Importing Source Tables from SQL Server](#) instructions above and import the target database and its tables.

Warning: If the target database does not allow table creation via SQL, InstantdbSync cannot create target tables. Target tables must be created in that database's native environment and not with the Replication Administrator.

If your target database is on UNIX, follow the special configuration instructions found in **CONNX Configuration Settings** in the **CONNX User Guide**.

1. Click **Import**. The **Import CDD** window appears. In **Import Type**, select the target database type.

Note: Different target databases will have Import CDD windows with different required information. The example shows Oracle.

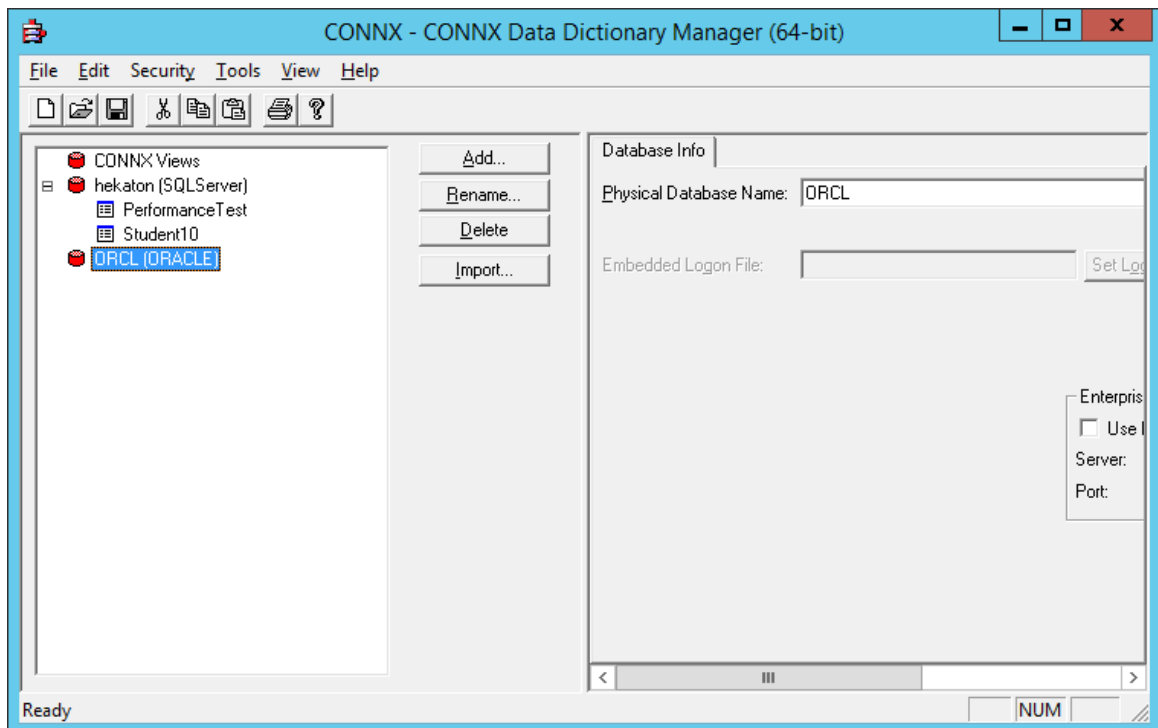
The screenshot shows the 'Import CDD' window. The title bar is 'Import CDD'. The main text says: 'The Import feature downloads your existing database record structures and stores them in the encrypted CONNX Data Dictionary for use by CONNX.' Below this are 'OK' and 'Cancel' buttons. The 'Import Type' dropdown is set to 'Oracle database'. The 'Enter the SQLNet connect name' text box contains 'ORCL'. There are two checkboxes: 'Include System Tables' (unchecked) and 'Get Statistics' (checked). At the bottom, there is a 'Logon Information' section with a 'UserName' text box containing 'oracleuser' and a 'Password' text box containing 'xxxxxxx'.

2. Fill in the SQLNet connect name and the Oracle logon credentials. Click **OK**.

The CONNX Import Table Selection window appears. If there are any tables in the target database, they will appear in Available Tables.

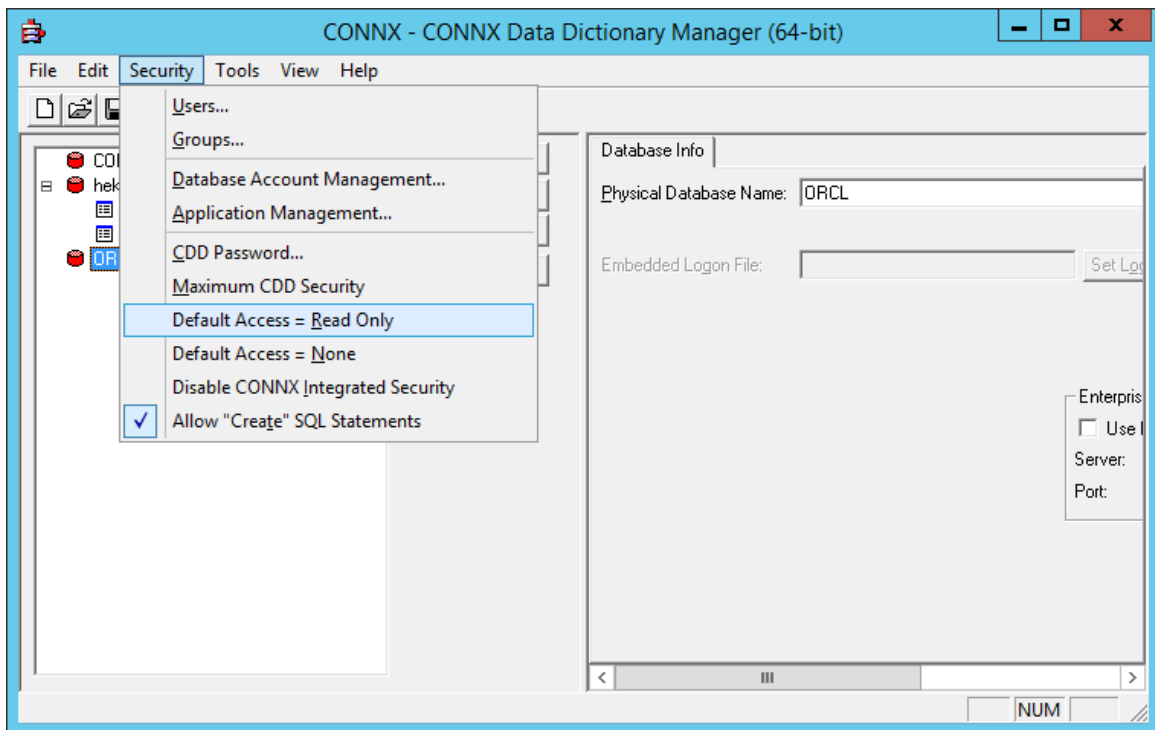
- If any of the existing tables will be used as target tables, select them and click **Add**. After enabling a CDD for Event Replication, map the source columns to the target table.
- If none of the existing tables will be target tables, do not select anything. You will create the new target table in the Event Replicator.
- To avoid future table name conflicts, click **Add All** to import all available tables.

Click **OK**. The CONNX Data Dictionary Manager window appears.

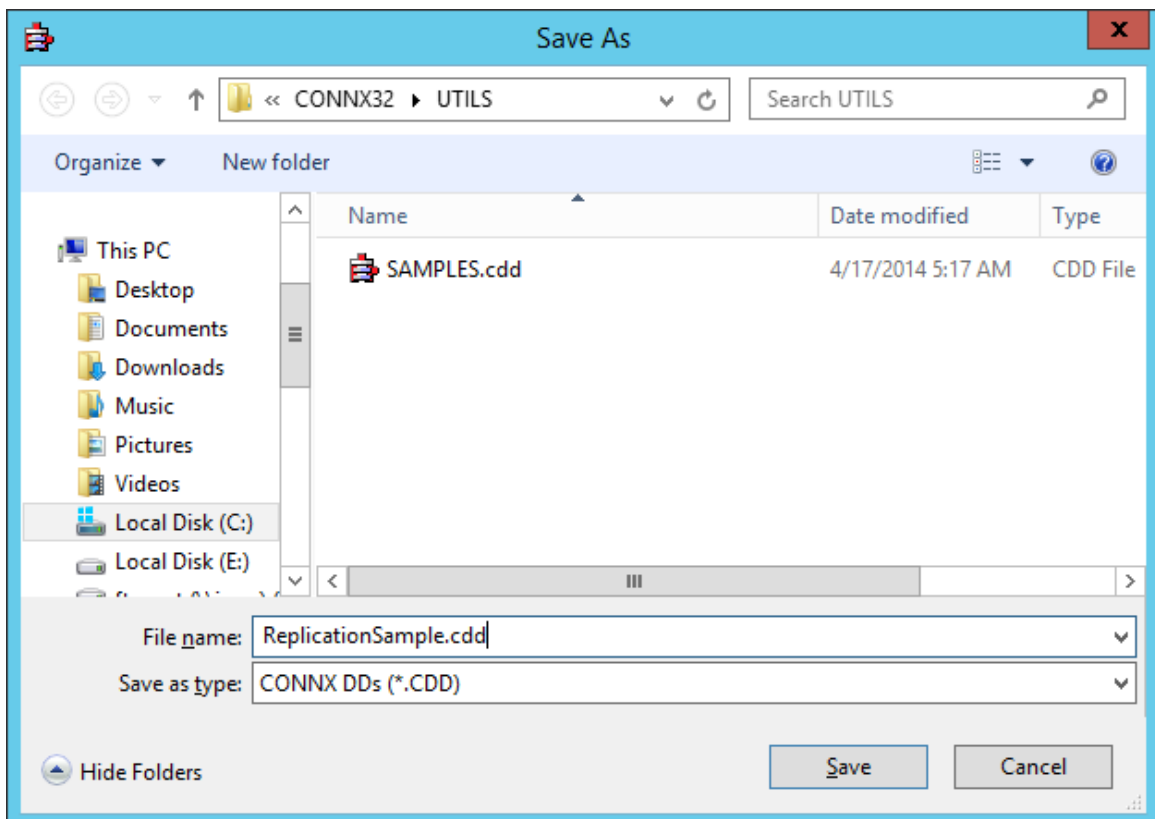


Set the CDD Security

1. In the **Security** menu, clear **Default Access = Read Only**.



2. In the **File** menu, select **Save As**.



3. Name your CDD and select the appropriate folder to save it in. Click **Save**.

4. Close the CONNX Data Dictionary Manager.

Enabling a CDD for Event Replication

Enabling a CDD for Event Replication is the first step in getting data replicated. This includes:

- Opening a CONNX Data Dictionary (CDD)
- Entering connection information to connect to the source and target databases through CONNX
- Selecting tables from a source database and a target database to be used in replication

Enable a CDD for Event Replication if:

- This is the first time you have used InstantdbSync.
- This is the first time you have used this CDD for event replication.
- You are using a different CDD than the one used during the last event replication session.

After you enable the Event Replicator you will have a CDD with replications that are ready to be:

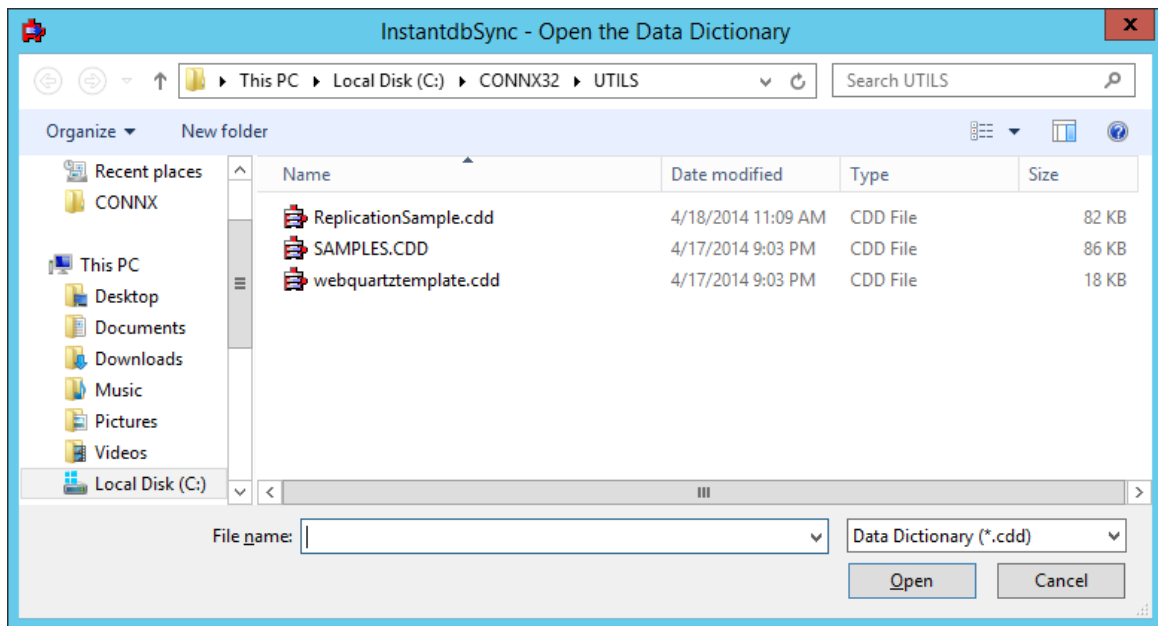
- Modified with column mapping
- Deployed (which starts replicating data from source database to target database)

The InstantdbSync Controller is designed to use a single CDD. When designing a set of replications create one CDD that includes all the desired source tables and target databases.

Caution: Replications from two separate CDDs cannot be deployed to the same Controller; if you attempt to do so, the replications from the second Data Dictionary will replace the replications from the first.

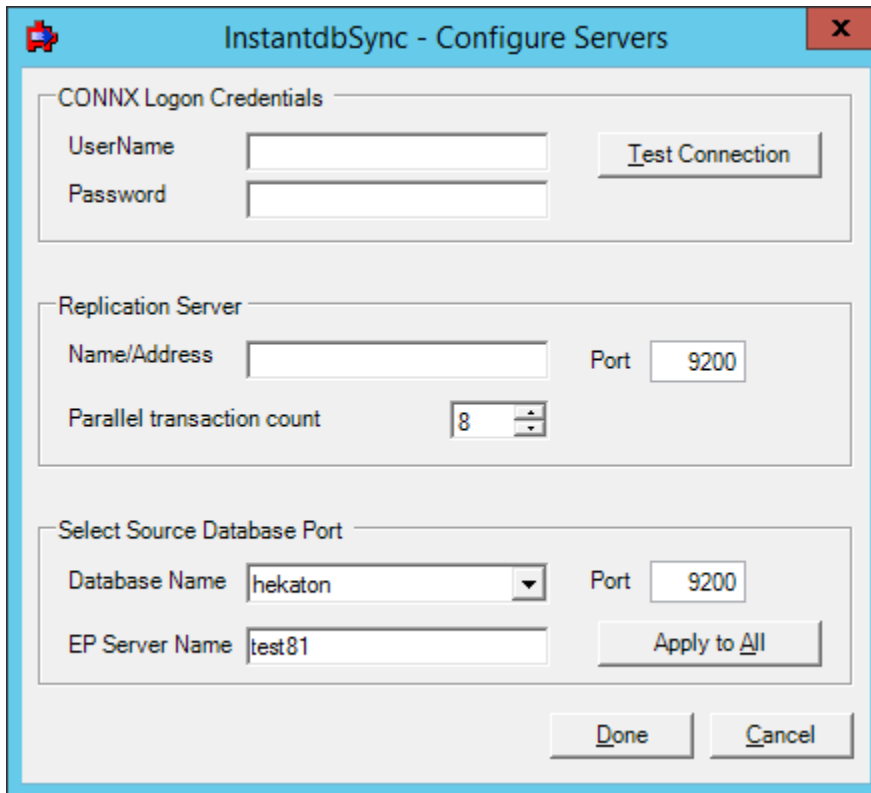
Caution: Two separate CDDs cannot replicate to the same targets from two different controllers. InstantdbSync is designed to maintain data integrity from the source to the target; any target modifications from more than one controller will result in replication errors, disabling the replication.

1. If you do not have a replication CDD created, create one.
2. On the **Start** menu, click **Programs**, click **CONNX InstantdbSync** and then click **InstantdbSync Administrator**. If this is the first time the InstantdbSync Administrator has been used, the **Open the Data Dictionary** window appears.



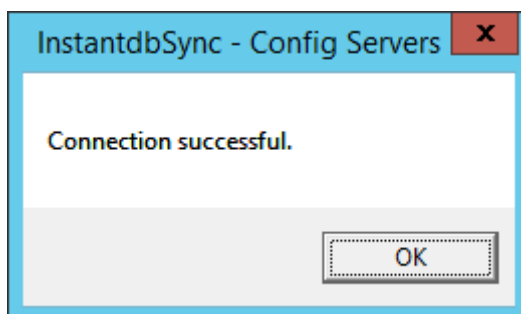
Note: Only CONNX Data Dictionaries (CDD) that contain both the source and target (destination) databases can be used for replication.

2. If the InstantdbSync Administrator has been used previously, the main InstantdbSync window appears. The application will normally open the last opened CDD.
 - If this is the CDD you want, you do not need to re-enable event replication for this CDD.
 - If this is not the CDD you want, from the **File** Menu, click **Open CDD**. Select a CDD that contains the source tables to be replicated and the target database, and click **Open**. The **Configure Servers** window appears
3. If the Replication Administrator has not been used before, select a CDD that contains the source tables to be replicated and the target database, and click **Open**. The **Configure Servers** window appears.



The image shows a Windows-style dialog box titled "InstantdbSync - Configure Servers". It has a blue title bar with a red close button (X) in the top right corner. The dialog is divided into three main sections. The first section, "CONNX Logon Credentials", contains two text input fields for "UserName" and "Password", and a "Test Connection" button to the right. The second section, "Replication Server", contains a "Name/Address" text input field, a "Port" text input field with the value "9200", and a "Parallel transaction count" spinner box set to "8". The third section, "Select Source Database Port", contains a "Database Name" dropdown menu with "hekaton" selected, a "Port" text input field with "9200", and an "EP Server Name" text input field with "test81". To the right of the "EP Server Name" field is an "Apply to All" button. At the bottom of the dialog are "Done" and "Cancel" buttons.

4. Enter your User Name and Password in **CONNX Logon Credentials** and click **Test Connection**. If the data is valid, the following message window appears:



5. Click **OK**. The message window closes.
6. Enter the server name or IP address of the Replication Server in **Name/Address** and the server port number in **Port**.

InstantdbSync - Configure Servers

CONNX Logon Credentials

UserName: testuser

Password: *****

Test Connection

Replication Server

Name/Address: test26

Port: 9200

Parallel transaction count: 8

Select Source Database Port

Database Name: hekaton

Port: 9200

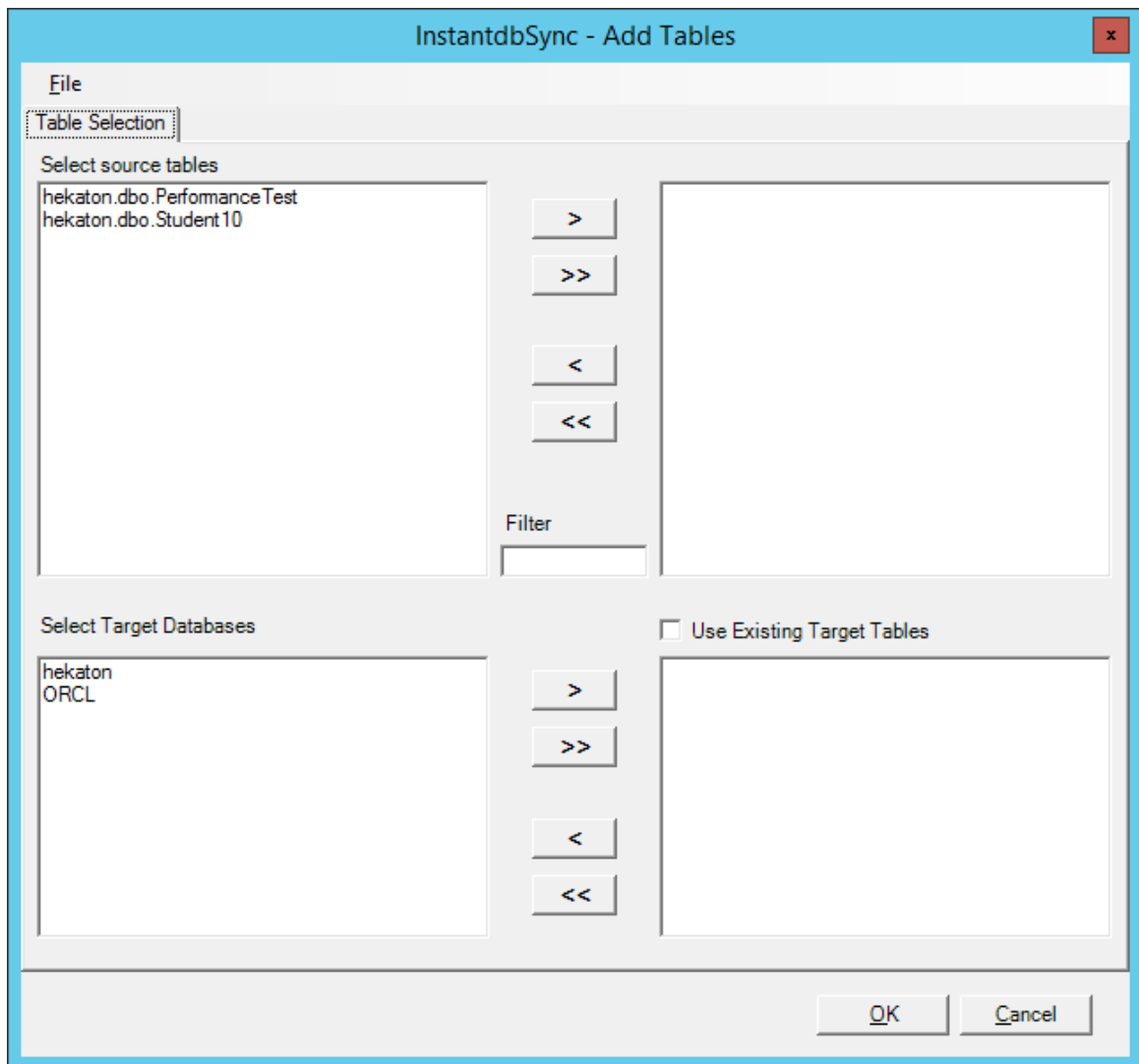
EP Server Name: test81

Apply to All

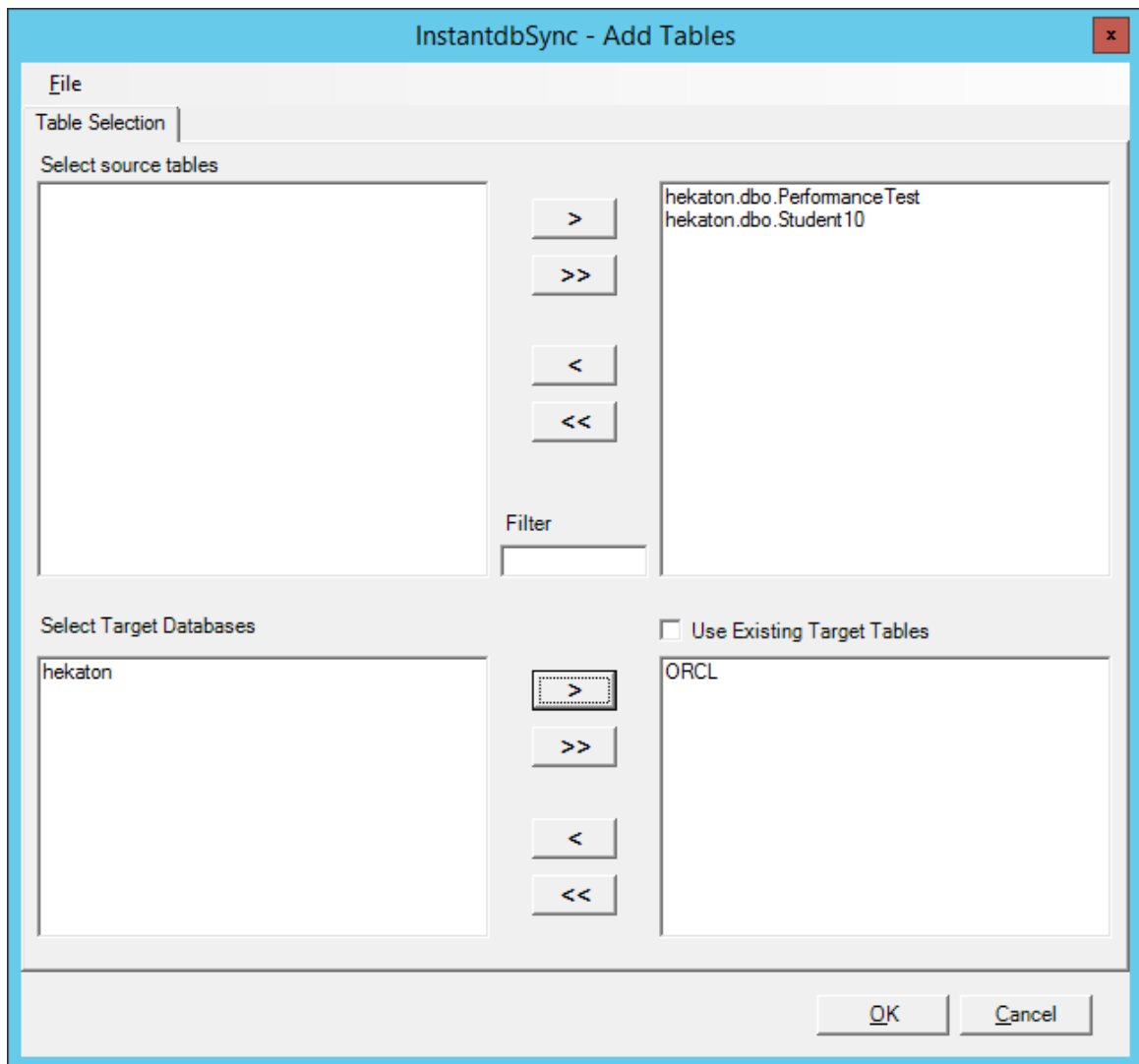
Done Cancel

Note: The port refers to the message queue server that is running on the machine where the replication controller is running. The controller uses this message queue server for its communications with the other components.. The default port number is 9200.

7. Click **Done**. The **Add Tables** window appears.



8. Select the source tables you wish to replicate and the target database.



9. Click **OK**. The **InstantdbSync** window appears.

Adding a Replication

Adding a Replication to InstantdbSync includes:

- Opening a CONNX Data Dictionary (CDD)
- Selecting a source table or tables from the CDD
- Selecting a target database

Add a replication if:

- You wish to replicate a source table that is not in the list of Event Replication source tables
- You wish to replicate a source table to an additional target table

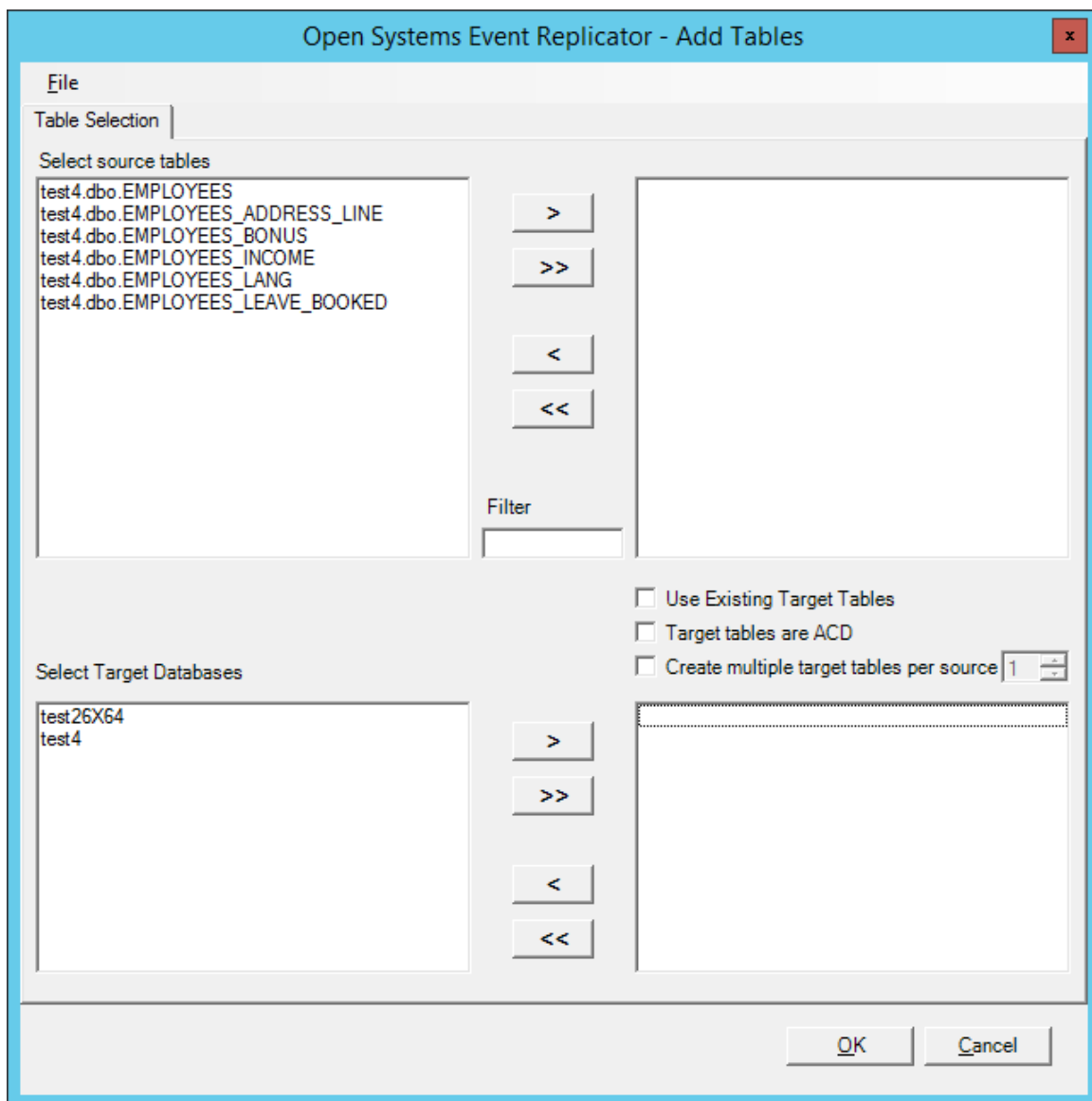
After you add a replication you will be able to:

- Modify it with column mapping
- Deploy it (which will start replicating data from source database to target database)

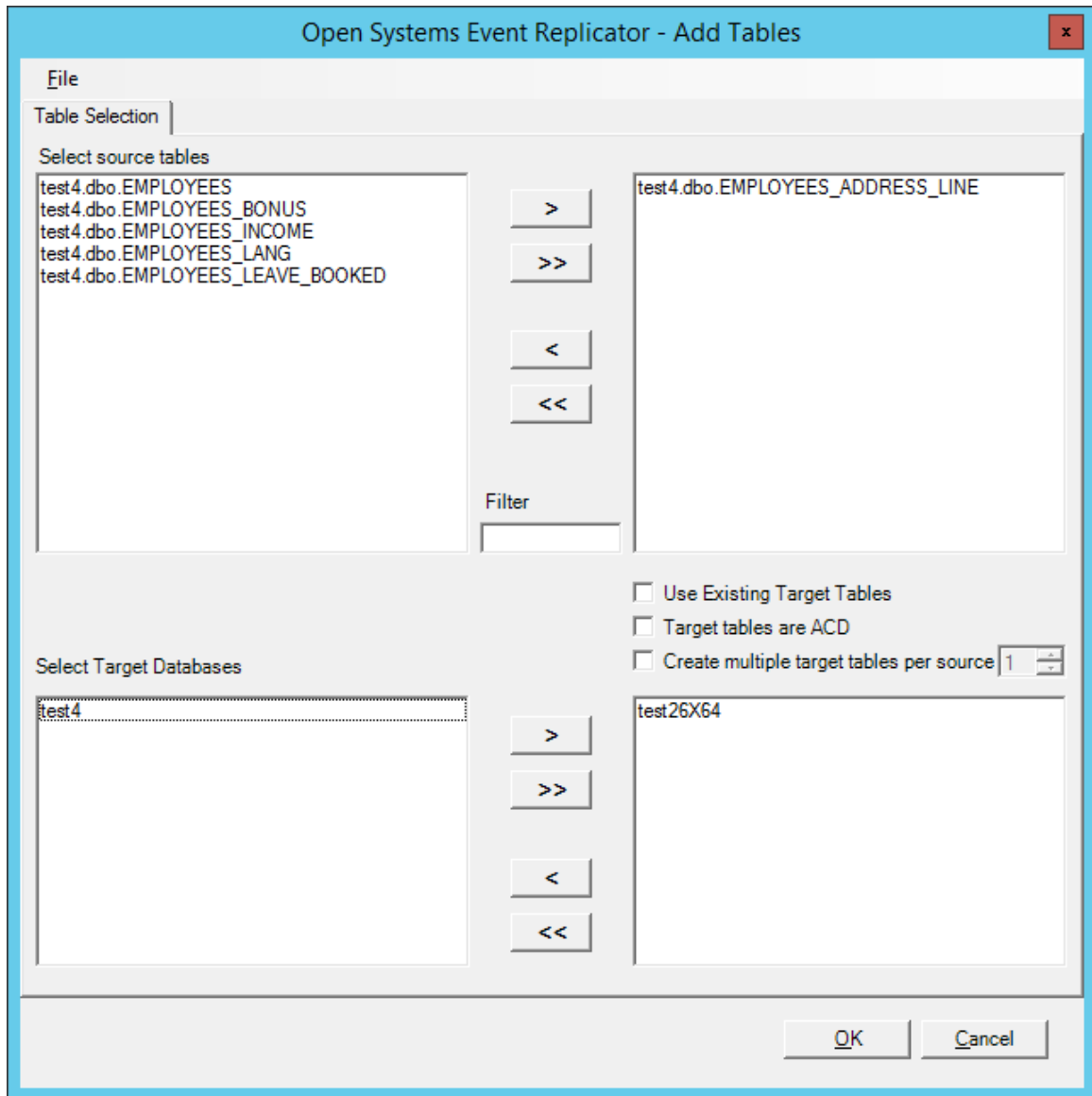
You can add additional source tables to InstantdbSync and associate them with a target database at the same time.

Warning: To maintain data integrity, a table cannot be both the source table in one replication and a target table in another replication.

1. Open the InstantdbSync Replication Administrator.
2. Click **Add Tables**. The **Add Tables** window appears.

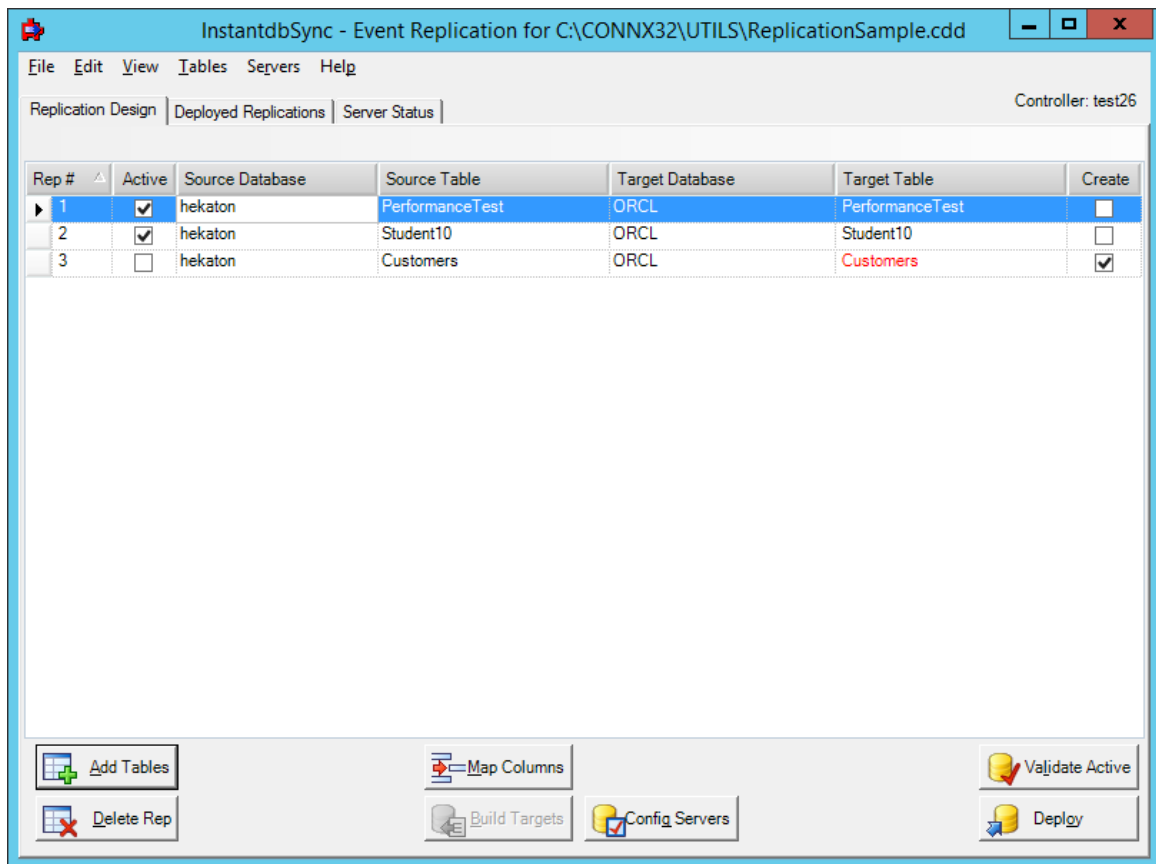


3. Select any additional source tables you wish to replicate and specify the target Database. By default the source tables will map to new target tables which exactly match the source tables to be created on deploy. Select the **Use Existing Target Tables** and the Replication Administrator will search for a table with the same name as the source table in the target database. If a table with the same name is not found in the target database, it will just create a new table with the same name. In this process it will also map columns with the same name from the source table to the target table. If there are columns in the existing tables with names that do not match the source they must be mapped manually in the **Column Map** window. If you are creating replications that are for ACD changes, then select the **Target tables are ACD**. If you would like to create several replications for each source table selected use the **Create multiple target tables per source** checkbox. Once checked, use the dropdown to specify how many replications to create. Each replication will have a different new target table, the name will be automatically generated.



Note: If there are a large number of source tables, the **Filter** window can be used to reduce the number of tables displayed. For example, if you know the table you are interested in has the word "Account" in it, entering Account in the Filter window will display only those tables that contain that word in their name.

4. Click **OK**. The **InstantdbSync Administrator** window appears and the tables you added are in the list.



- If you want to deploy the changes you've made, check the **Active** check box to in the second column and then click **Validate Active** and then **Deploy**. If you wish to save the changes but not deploy them until a later time, click the **File** menu and then click **Save**.

Note: Clicking **Validate Active** is an optional step. All active replications will be automatically validated during the deploy process. Use the **Validate Active** button when you want to validate newly added replications prior to deploy.

For more information about **Validate Active**, see [Validating Active Replications](#). For more information about **Deploy**, see [Deploying Replications](#).

Mapping columns to a new Target Table

Mapping columns to a new Target Table includes:

- Opening a CONNX Data Dictionary (CDD)
- Viewing, and if necessary, changing some of the target table metadata

Map columns to a new Target Table if:

- You wish to change the target table metadata so it is different than the source table metadata
- You don't need all of the source columns replicated to the target table

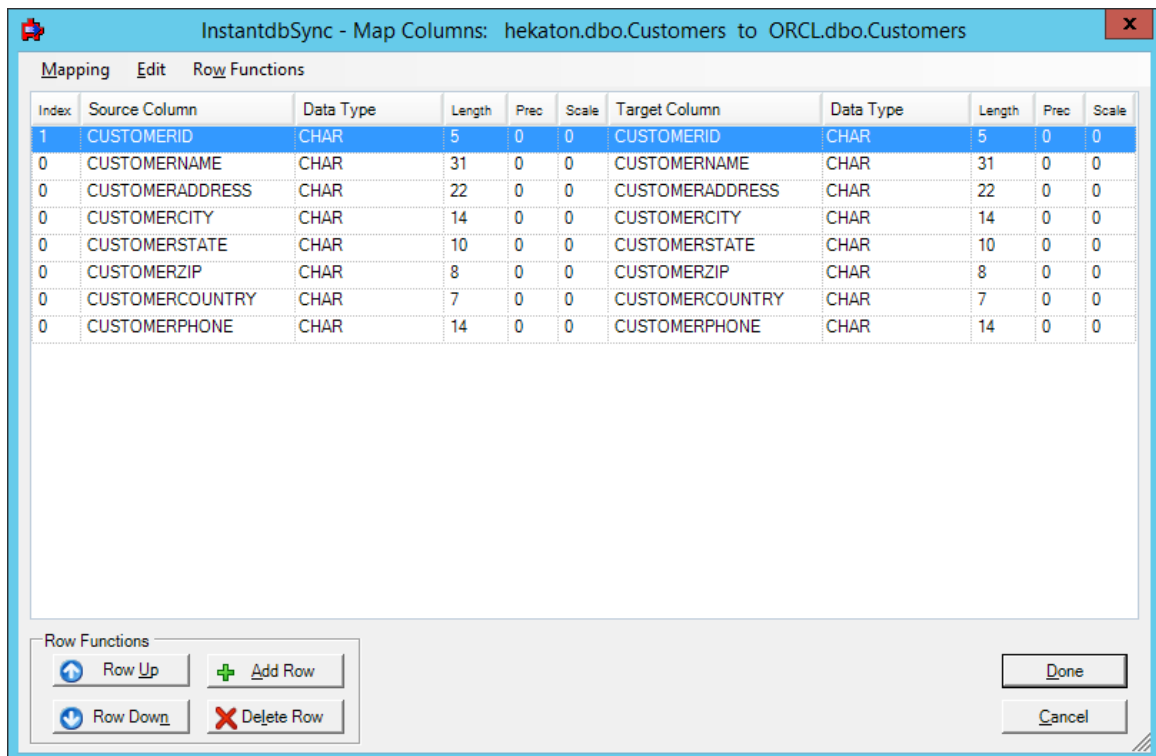
After you map columns to a new Target Table you will be able to:

- Create the target table
- Deploy the replication (which will start replicating data from source database to target database)

A target table is considered to be "new" if it does not exist on the target database and it has not been created or deployed.

When the InstantdbSync Replication Administrator creates a Target Table, it assigns column source and attributes based on the associated Source Table. **Map Columns** allow you to change the column source and the column attributes.

1. Open the InstantdbSync Replication Administrator.
2. Select a Target Table to view. Click **Map Columns**. The **Map Columns** window appears.



- The **Index** entry is used to indicate if a field will be part of the target table index. Even if your source table contains multiple index fields, you can only use a single unique index for replication. You can indicate the index order by designating "1" as the first part of the index, "2" as the second part of the index (if necessary), and so on. A zero means this field is not part of the index.
- The **Source Column** names are pre-populated and cannot be changed.
- Select the correct SQL data type in Source and Target **Data Type**.
- You can change the Source and Target **Length**, **Precision** and **Scale** values if they are incorrect.
- **Target Column** is pre-populated with the same name as the **Source Column**. This name can be changed.

Note: You cannot use the column mapper to change the order of the rows in the Source Table.

Adding New Column Mappings

Adding new column mappings includes:

- Opening a CONNX Data Dictionary (CDD)
- Adding a row to the column map grid
- Specifying the source and target column schemas for the new row

Add new column mappings if:

- You wish to replicate source data that is not in the current column mapping
- Your column map will include a source column SQL expression

After you add a new column mappings you will be able to:

- Create the target table
- Deploy the replication (which will start replicating data from source database to target database)

A target table is considered to be "new" if it does not exist on the target database and it has not been created or deployed.

By default, when a new target table is defined, all source table metadata is duplicated in the target table metadata. You can add column maps to a new non-Adabas Target Table. The additional target maps can map to any Source Columns, including SQL expressions.

Note: Column mapping does not change the structure of the Source table or an existing Target table. The Add Row, Delete Row, Row Up and Row Down buttons are only available when mapping to a new target table. If you are mapping to an existing target table and wish to change the structure of that table, you will need to press the "Drop Target Table" button (available when mapping to an existing table). This will drop the target table and allow you to re-create it with the new structure. **Warning: Only do this if you intend to physically drop and re-create the target table. All data in the existing table will be lost.**

1. Open the InstantdbSync Replication Administrator.
2. Select a Target Table to add rows to. Click **Map Columns**. The **Map Columns** window appears.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Row Functions

3. To add a row to the Target Table
 - a. Click somewhere in the data and click **Add Row**. A new row appears at the end.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0
0	UNKNOWN	UNKNOWN	0	0	0	UNKNOWN	UNKNOWN	0	0	0

Row Functions

- b. Use **Row Up** or **Row Down** to place the row in the correct position.
 - c. Click the **Source Column** box for the row you have added. A drop down arrow appears. Click on the arrow and a list of available source table columns appears.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0		UNKNOWN	0	0	0		UNKNOWN	0	0	0
0		CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERADDRESS	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Row Functions

Row Up Add Row

Row Down Delete Row

Done Cancel

d. Select the source table column to be added.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERNAME	CHAR	31	0	0		UNKNOWN	0	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Row Functions

Row Up Add Row

Row Down Delete Row

Done Cancel

e. Click the **Target Column** box for the row you have added. A drop down arrow appears. Click on the arrow and a list of available target table columns appears

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERNAME	CHAR	31	0	0		UNKNOWN	0	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0		CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0		CHAR	14	0	0

Row Functions

Row Up Add Row

Row Down Delete Row

Done

Cancel

- Select a target column name from this list or enter another target column name in **Target Column**.
- Select the desired **Data Type** of the target column and fill in the **Length**, **Precision** and **Scale**.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	DATE	14	0	0

Row Functions

Row Up Add Row

Row Down Delete Row

Done

Cancel

Deleting Column Mappings from a New Target Table

Deleting new column mappings includes:

- Opening a CONNX Data Dictionary (CDD)
- Deleting a row from the column map grid

Delete a new column mappings if:

- You wish to remove source data from the current column mapping

After you delete a new column mappings you will be able to:

- Create the target table
- Deploy the replication (which will start replicating data from source database to target database)

A target table is considered to be "new" if it does not exist on the target database and it has not been created or deployed.

Note: Deleting a column map does not change the structure of the Source table or an existing Target table. The Add Row, Delete Row, Row Up and Row Down buttons are only available when mapping to a new target table. If you are mapping to an existing target table and wish to change the structure of that table, you will need to press the "Drop Target Table" button (available when mapping to an existing table). This will drop the target table and allow you to re-create it with the new structure. **Warning: Only do this if you intend to physically drop and re-create the target table. All data in the existing table will be lost.**

1. Open the InstantdbSync Replication Administrator.
2. Select a replication and click **Map Columns**. The **Map Columns** window appears.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Row Functions

3. To delete a column from the Target Table:
 - a. Select the row that represents the column on the target to be deleted.

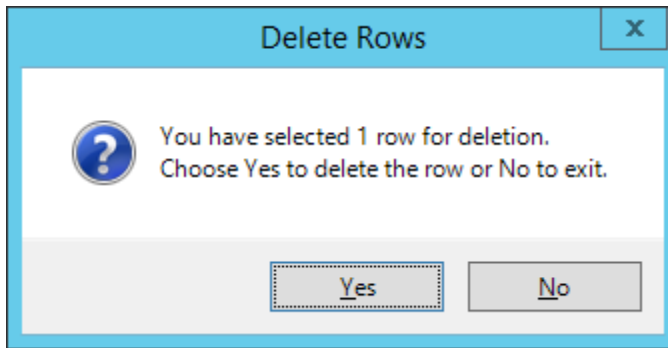
InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

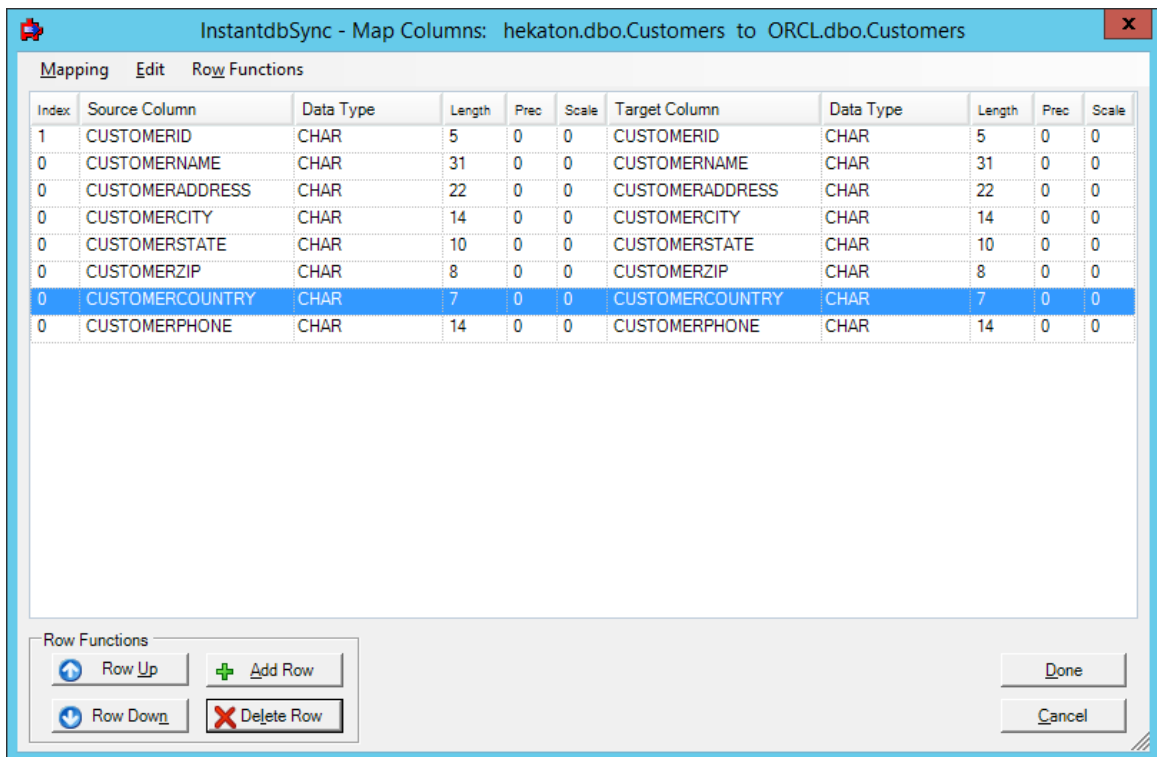
Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Row Functions

- b. Click **Delete Row**. The following message appears:



c. Click **Yes**. The row is deleted.



Note: You can not use the column mapper to delete a column from the Source Table.

4. Click **Done**. The **InstantdbSync** main window reappears.

Changing the Column Order in a New Target Table

Changing the column order in a new target table includes:

- Opening a CONNX Data Dictionary (CDD)
- Moving the rows in a column map so the target table columns are in the correct order

Change the column order in a new target table if:

- You want the source and target table columns to be in different order

After you change the column order in a new target table you will be able to:

- Create the target table
- Deploy the replication (which will start replicating data from source database to target database)

A target table is considered to be "new" if it does not exist on the target database and it has not been created or deployed.

You can change the column order in a new Target Table. Columns can be grouped together even if they don't appear in that order in the Source Table.

Note: You can not use the column mapper to change the order of the columns in the Source Table. The Add Row, Delete Row, Row Up and Row Down buttons are only available when mapping to a new target table. If you are mapping to an existing target table and wish to change the structure of that table, you will need to press the "Drop Target Table" button (available when mapping to an existing table). This will drop the target table and allow you to re-create it with the new structure. **Warning: Only do this if you intend to physically drop and re-create the target table. All data in the existing table will be lost.**

1. Open the InstantdbSync Replication Administrator.
2. Select a replication that contains a new target table to change. Click **Map Columns**. The **Map Columns** window appears.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.Customers

Mapping Edit Row Functions

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Row Functions

3. To change the order of the rows in the target table:
 - a. Select the row to be moved.
 - b. Click **Row Up** or **Row Down**. The selected row moves in the specified direction.

Changing Source Column Mapping

Changing source column mappings includes:

- Opening a CONNX Data Dictionary (CDD)
- Changing which source column a target column is mapped to

Change the source column mappings if:

- You wish to change which source column a target column is mapped to

After you change the source mappings you will be able to:

- Deploy the replication (which will start replicating data from source database to target database)

Map Columns allow you to change which source column an existing target column is mapped to.

Note: The Source and Target columns must have compatible data type, length, precision and scale.

1. Open the InstantdbSync Replication Administrator.
2. Select a replication to modify. Click **Map Columns**. The **Map Columns** window for an existing target table appears.

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	FIRSTNAME	CHAR	10	0	0	FIRSTNAME	CHAR	10	0	0
0	LASTNAME	CHAR	21	0	0	LASTNAME	CHAR	21	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

3. Under **Populate columns**, click **Source Column**. Click the **Source Column** box for the replication you wish to change. A drop down arrow appears. Click on the arrow and a list of available source column names appears.

InstantdbSync - Map Columns: hekaton.dbo.CustomerInfo to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERCITY	CHAR	10	0	0	FIRSTNAME	CHAR	10	0	0
0	CUSTOMERCOUNTRY	CHAR	21	0	0	LASTNAME	CHAR	21	0	0
0	CUSTOMERID	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	FIRSTNAME	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERCOUNTRY	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Populate columns

☒ Source Columns

☐ Target Columns

Target Table

4. Select the source column name to map the corresponding target column to.

InstantdbSync - Map Columns: hekaton.dbo.CustomerInfo to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	FIRSTNAME	CHAR	10	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERCOUNTRY	CHAR	10	0	0	FIRSTNAME	CHAR	10	0	0
0	CUSTOMERID	CHAR	21	0	0	LASTNAME	CHAR	21	0	0
0	CUSTOMERNAME	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	FIRSTNAME	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	LASTNAME	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Populate columns

☒ Source Columns

☐ Target Columns

Target Table

5. The selected Source Column will map to the corresponding Target Column.

InstantdbSync - Map Columns: hekaton.dbo.CustomerInfo to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	FIRSTNAME	CHAR	10	0	0	CUSTOMERNAME	CHAR	31	0	0
0	FIRSTNAME	CHAR	10	0	0	FIRSTNAME	CHAR	10	0	0
0	LASTNAME	CHAR	21	0	0	LASTNAME	CHAR	21	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Populate columns

☒ Source Columns
 ☐ Target Columns

Clear Target
 Clear Source

Target Table

Drop Target Table

Done

Cancel

- Click **Done**. The **InstantdbSync** main window reappears.

Changing Target Column Mapping

Changing target column mappings includes:

- Opening a CONNX Data Dictionary (CDD)
- Changing which target column a source column is mapped to

Change the target column mappings if:

- You wish to change which target column a source column is mapped to

After you change the column mappings you will be able to:

- Deploy the replication (which will start replicating data from source database to target database)

When the Event Replicator creates a Target Table, it assigns column source and attributes based on the associated Source Table. **Map Columns** allow you to change which target column a source column is mapped to

Note: The Source and Target columns must have compatible data types, length, precision and scale.

Note: You can not use the column mapper to delete a column from the Source Table.

1. Open the InstantdbSync Replication Administrator
2. Select a replication to modify. Click **Map Columns**. The **Map Columns** window for an existing target table appears.

The screenshot shows the 'InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.CustomerInfo' window. It has a 'Mapping' tab selected. The window contains a table with columns: Index, Source Column, Data Type, Length, Prec, Scale, Target Column, Data Type, Length, Prec, Scale. The table lists 8 columns: CUSTOMERID, CUSTOMERNAME, CUSTOMERADDRESS, CUSTOMERCITY, CUSTOMERSTATE, CUSTOMERZIP, CUSTOMERCOUNTRY, and CUSTOMERPHONE. All columns are mapped to their corresponding target columns with the same data type (CHAR) and attributes (Length, Prec, Scale). At the bottom, there are buttons for 'Populate columns' (Source Columns, Target Columns), 'Clear Target', 'Clear Source', 'Target Table', 'Drop Target Table', 'Done', and 'Cancel'.

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

- Under **Populate columns**, click **Target Columns**.

In this example, the target table has two columns that do not exist in the source table (FIRSTNAME and LASTNAME). By populating the target columns, these two columns become visible and do not have any default mapping to corresponding source columns.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0
0		UNKNOWN	0	0	0	FIRSTNAME	CHAR	10	0	0
0		UNKNOWN	0	0	0	LASTNAME	CHAR	21	0	0

Populate columns

☐ Source Columns

☒ Target Columns

Target Table

- Click the **Target Column** box for the column you wish to change. A drop down arrow appears. Click on the arrow and a list of available source column names appears. Select the target column name to map the corresponding target column to.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERCOUNTRY	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERID	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERNAME	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERPHONE	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERSTATE	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERZIP	CHAR	14	0	0
0		UNKNOWN	0	0	0	FIRSTNAME	CHAR	10	0	0
0		UNKNOWN	0	0	0	LASTNAME	CHAR	21	0	0

Populate columns

☐ Source Columns

☒ Target Columns

Target Table

6. The selected Source Column will map to the corresponding Target Column..

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	FIRSTNAME	CHAR	10	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0
0		UNKNOWN	0	0	0	FIRSTNAME	CHAR	10	0	0
0		UNKNOWN	0	0	0	LASTNAME	CHAR	21	0	0

Populate columns

☐ Source Columns

☒ Target Columns

Target Table

7. You can also map either existing source columns or an SQL expression to the two columns that exist in the target table but not the source. In this example, we have use the SUBSTRING expression to fill

CONNX InstantdbSync

FIRSTNAME and LASTNAME.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5
0	CUSTOMERNAME	CHAR	31	0	0	FIRSTNAME	CHAR	10
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14
0	SUBSTRING (CUSTOMERNAME, 1, 10)	UNKNOWN	0	0	0	FIRSTNAME	CHAR	10
0	SUBSTRING (CUSTOMERNAME, 11, 21)	UNKNOWN	0	0	0	LASTNAME	CHAR	21

Populate columns

☐ Source Columns

☒ Target Columns

Target Table

8. Click **Done**. The **InstantdbSync** main window reappears.

Clearing Source Column Mapping

Clearing source column mappings includes:

- Opening a CONNX Data Dictionary (CDD)
- Clearing the target columns

Clear the source column mappings if:

- The default target mappings are incorrect

After you clear the source mappings you will be able to:

- Select the correct target columns to map to the source columns
- Deploy the replication (which will start replicating data from source database to target database)

Map Columns allow you to change which source column a target column is mapped to.

If you wish to change most of the source column mapping in a Target Table, you may find it easier to start with blank Target column names. The column names will still appear in the drop-down list.

1. Open the InstantdbSync Replication Administrator.
2. Select a Target Table to modify. Click **Map Columns**. The **Map Columns** window for an existing target table appears.

The screenshot shows the 'InstantdbSync - Map Columns: hekaton.dbo.CustomerInfo to ORCL.dbo.CustomerInfo' window. It features a table with columns for Index, Source Column, Data Type, Length, Prec, Scale, Target Column, Data Type, Length, Prec, and Scale. The first row is highlighted in blue, showing a mapping from 'CUSTOMERID' to 'CUSTOMERID'. Below the table, there are options to 'Populate columns' (Source Columns or Target Columns) and buttons for 'Clear Target', 'Clear Source', 'Drop Target Table', 'Done', and 'Cancel'.

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	FIRSTNAME	CHAR	10	0	0	FIRSTNAME	CHAR	10	0	0
0	LASTNAME	CHAR	21	0	0	LASTNAME	CHAR	21	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Populate columns
☒ Source Columns ☐ Target Columns
 Clear Target Clear Source
 Target Table
 Drop Target Table
 Done Cancel

3. Under **Populate columns**, click **Source Column**.

4. Click **Clear Target**. The **Target Column** names are blanked out.

The screenshot shows the 'InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.CustomerInfo' dialog box. The 'Mapping' tab is active, displaying a table with columns: Index, Source Column, Data Type, Length, Prec, Scale, Target Column, Data Type, Length, Prec, and Scale. The table lists eight source columns: CUSTOMERID, CUSTOMERNAME, CUSTOMERADDRESS, CUSTOMERCITY, CUSTOMERSTATE, CUSTOMERZIP, CUSTOMERCOUNTRY, and CUSTOMERPHONE. All target columns are currently blank. Below the table, there are two sections: 'Populate columns' with radio buttons for 'Source Columns' (selected) and 'Target Columns', and 'Target Table' with a 'Drop Target Table' button. At the bottom right are 'Done' and 'Cancel' buttons. The 'Clear Target' button is highlighted in the 'Populate columns' section.

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
0	CUSTOMERID	CHAR	5	0	0		UNKNOWN	0	0	0
0	CUSTOMERNAME	CHAR	31	0	0		UNKNOWN	0	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0		UNKNOWN	0	0	0
0	CUSTOMERCITY	CHAR	14	0	0		UNKNOWN	0	0	0
0	CUSTOMERSTATE	CHAR	10	0	0		UNKNOWN	0	0	0
0	CUSTOMERZIP	CHAR	8	0	0		UNKNOWN	0	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0		UNKNOWN	0	0	0
0	CUSTOMERPHONE	CHAR	14	0	0		UNKNOWN	0	0	0

Populate columns
☒ Source Columns ☐ Target Columns
Clear Target Clear Source

Target Table
Drop Target Table

Done Cancel

Clearing Target Column Mapping

Clearing target column mappings includes:

- Opening a CONNX Data Dictionary (CDD)
- Clearing the source columns

Clear the target column mappings if:

- The default source column mappings are incorrect

After you clear the column mappings you will be able to:

- Select the correct source columns to map to the target columns
- Deploy the replication (which will start replicating data from source database to target database)

If you wish to change most of the target column mapping, you may find it easier to start with blank Source column names. The column names will still appear in the drop-down list.

1. Open the Open Systems Event Replicator CDD.
2. Select a Target Table to modify. Click **Map Columns**. The **Map Columns** window for an existing target table appears.

The screenshot shows the 'InstantdbSync - Map Columns: hekaton.dbo.CustomerInfo to ORCL.dbo.CustomerInfo' window. It features a table with columns for Index, Source Column, Data Type, Length, Prec, Scale, Target Column, Data Type, Length, Prec, and Scale. The table lists 10 columns: CUSTOMERID, CUSTOMERNAME, FIRSTNAME, LASTNAME, CUSTOMERADDRESS, CUSTOMERCITY, CUSTOMERSTATE, CUSTOMERZIP, CUSTOMERCOUNTRY, and CUSTOMERPHONE. All source and target columns are currently mapped to each other. Below the table, there are controls for 'Populate columns' (Source Columns and Target Columns) and 'Clear Target'/'Clear Source' buttons. There is also a 'Target Table' section with a 'Drop Target Table' button and 'Done'/'Cancel' buttons at the bottom right.

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	FIRSTNAME	CHAR	10	0	0	FIRSTNAME	CHAR	10	0	0
0	LASTNAME	CHAR	21	0	0	LASTNAME	CHAR	21	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

3. Under **Populate columns**, click **Target Column**.
4. Click **Clear Source**. The **Source Column** names are blanked out.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
0		UNKNOWN	0	0	0	CUSTOMERID	CHAR	5	0	0
0		UNKNOWN	0	0	0	CUSTOMERNAME	CHAR	31	0	0
0		UNKNOWN	0	0	0	FIRSTNAME	CHAR	10	0	0
0		UNKNOWN	0	0	0	LASTNAME	CHAR	21	0	0
0		UNKNOWN	0	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0		UNKNOWN	0	0	0	CUSTOMERCITY	CHAR	14	0	0
0		UNKNOWN	0	0	0	CUSTOMERSTATE	CHAR	10	0	0
0		UNKNOWN	0	0	0	CUSTOMERZIP	CHAR	8	0	0
0		UNKNOWN	0	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0		UNKNOWN	0	0	0	CUSTOMERPHONE	CHAR	14	0	0

Populate columns

☐ Source Columns

☒ Target Columns

Target Table

Importing and Exporting Column Mappings

Summary:

The Column Mapping Form has a feature to help save mappings once they have been created. In a replication after finishing mapping the source columns to target columns, those mappings can be saved to a file by selecting "Mapping --> Export". If something changes in the replication and you wish to retrieve these saved mappings, you can select "Mapping --> Import" to bring them back into the replication.

Reasons to use:

After making changes to the default mappings in a replication, if the source or target table needs to be changed outside of replication, the built mappings can be saved so the columns can be easily remapped after the change. A few cases where it can save time to import column mappings:

- When the target table is changed to a different table that has a similar schema. When pointed to the new target table the replication will reset to the default column mappings.
- Changing the source table results in a new replication.

Exporting Column Mapping to a file:

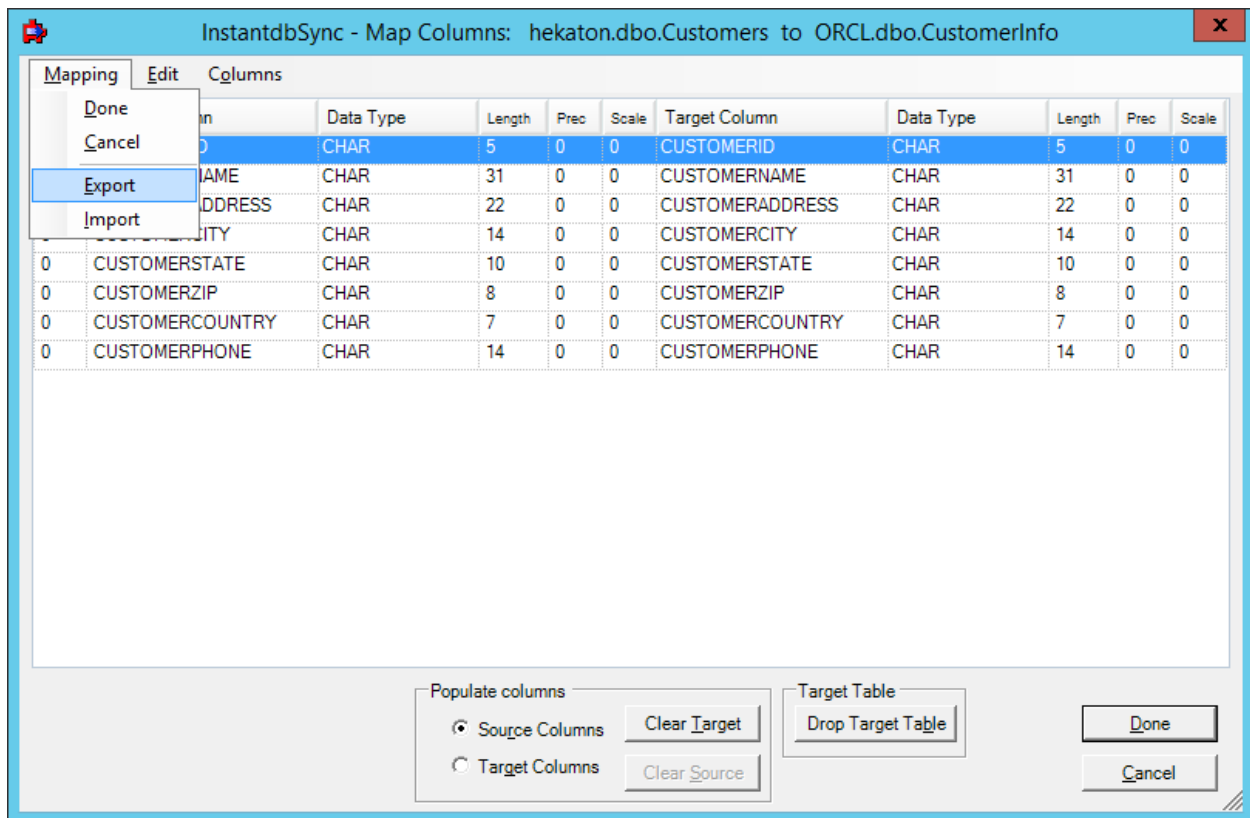
This is done while at the Column Mapping form. Select the "Mapping" menu item, then "Export". Then it will prompt for a name and location of a file to save the mappings to. The mappings will be saved in a comma separated format so they can be viewed and changed if desired with another file editor.

Importing a Column Mapping from a file:

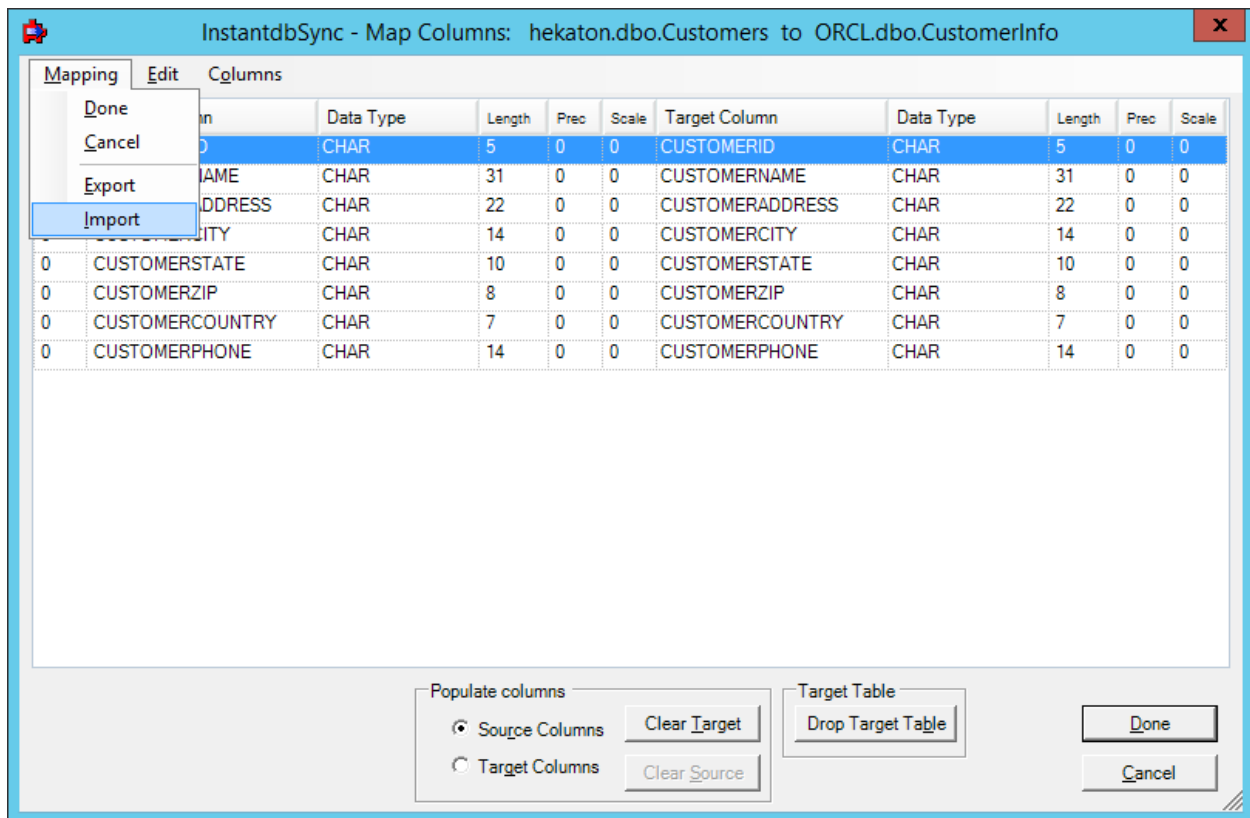
This is done while at the Column Mapping form. Select the "Mapping" menu item, then "Import". Locate the mapping file to import and press ok, then the column mapper will try to apply all the mappings saved in the file to the open column mapping form. When importing, the mappings the on screen will be overwritten by the mappings that are being imported. Columns mapped in the import file that are not on the screen will be added. Some column mappings from the import file may not be resolved to the source and target table columns and will not be added. Note, the validation of the mappings is performed when the "Done" button is pressed on the column mapper, not on import. This gives the user the ability to change a mapping that was imported but has data that is out of date with the new table.

Example:

Step 1: A replication that is has a target table with column mappings that include target column renames and an expression on the source table. The user wishes to point the replication source to a different target table. Since that will drop the existing column mappings, before changing the target table in the replication, go into the Column Map form and Export the mappings with "Mapping" --> "Export": Press done to go back to the main replication screen.



Step 2: After changing the target table on the replication, open the Column map for that replication, it will show the default column mappings. Import the column mappings using "Mapping" --> "Import" and selecting the file that was previously exported.



Step 3: Verify the new mappings from the import are what was desired and select Done to validate the mappings.

InstantdbSync - Map Columns: hekaton.dbo.Customers to ORCL.dbo.CustomerInfo

Mapping Edit Columns

Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	CUSTOMERID	CHAR	5	0	0	CUSTOMERID	CHAR	5	0	0
0	CUSTOMERNAME	CHAR	31	0	0	CUSTOMERNAME	CHAR	31	0	0
0	CUSTOMERADDRESS	CHAR	22	0	0	CUSTOMERADDRESS	CHAR	22	0	0
0	CUSTOMERCITY	CHAR	14	0	0	CUSTOMERCITY	CHAR	14	0	0
0	CUSTOMERSTATE	CHAR	10	0	0	CUSTOMERSTATE	CHAR	10	0	0
0	CUSTOMERZIP	CHAR	8	0	0	CUSTOMERZIP	CHAR	8	0	0
0	CUSTOMERCOUNTRY	CHAR	7	0	0	CUSTOMERCOUNTRY	CHAR	7	0	0
0	CUSTOMERPHONE	CHAR	14	0	0	CUSTOMERPHONE	CHAR	14	0	0

Populate columns

☒ Source Columns ☐ Target Columns

Clear Target Clear Source

Target Table

Drop Target Table

Done Cancel

Deleting a Replication

Deleting a replication includes:

- Opening a CONNX Data Dictionary (CDD)
- Removing the replication from the Replication Design tab

Delete a replication if:

- You no longer want to replicate this source data to this target table.

After you delete a replication you will be able to:

- Deploy the data dictionary (the deleted replication will be removed from the Controller)

Deleting a replication does not remove the replication by itself. The replication will no longer appear in the [Replication Design tab](#). Data replication will continue until the next time the CDD is deployed.

1. Open the InstantdbSync Replication Administrator.
2. Select the replication or replications you wish to delete and click **Delete Rep**. A confirmation dialog will appear. Click Yes to proceed with the delete operation.
3. The selected replication or replications will be deleted no matter whether the replication is marked **Active** or **Create**.
4. When you are done deleting replications, click the **File** menu and then click **Save**.

Viewing the Source Table Schema

Viewing the source table schema includes:

- Opening a CONNX Data Dictionary (CDD)
- Showing or hiding the source database schemas

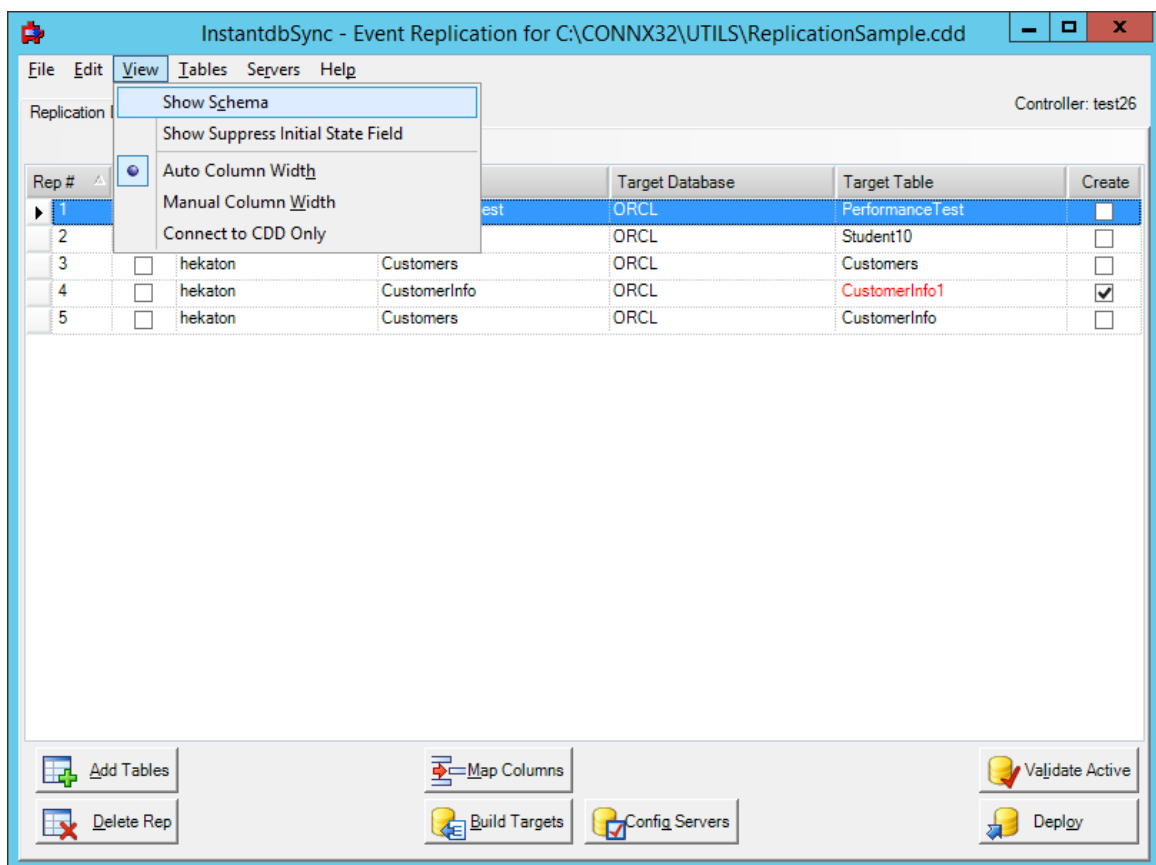
View the source table schema if:

- You may have more than one table with the same name but different schemas

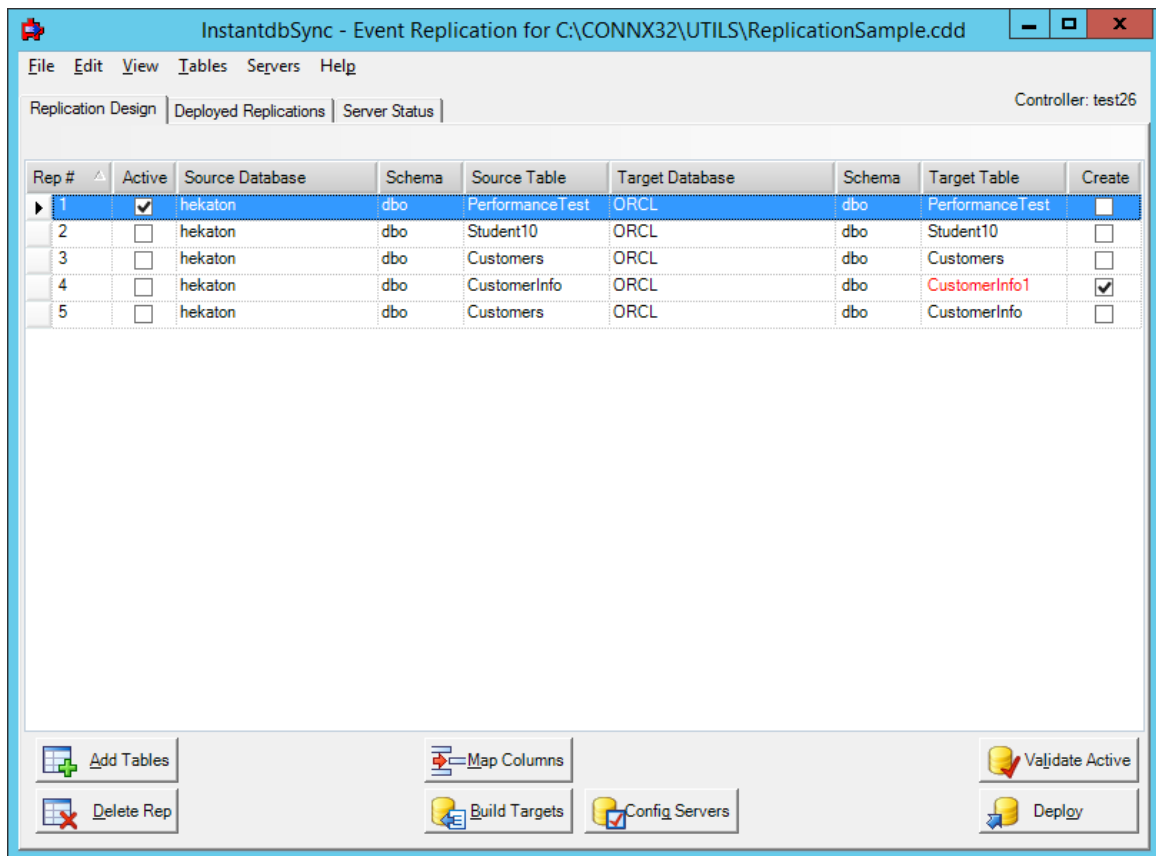
After you view the source table schema you will be able to:

- Map Columns
- Deploy the replication (which will start replicating data from source database to target database)

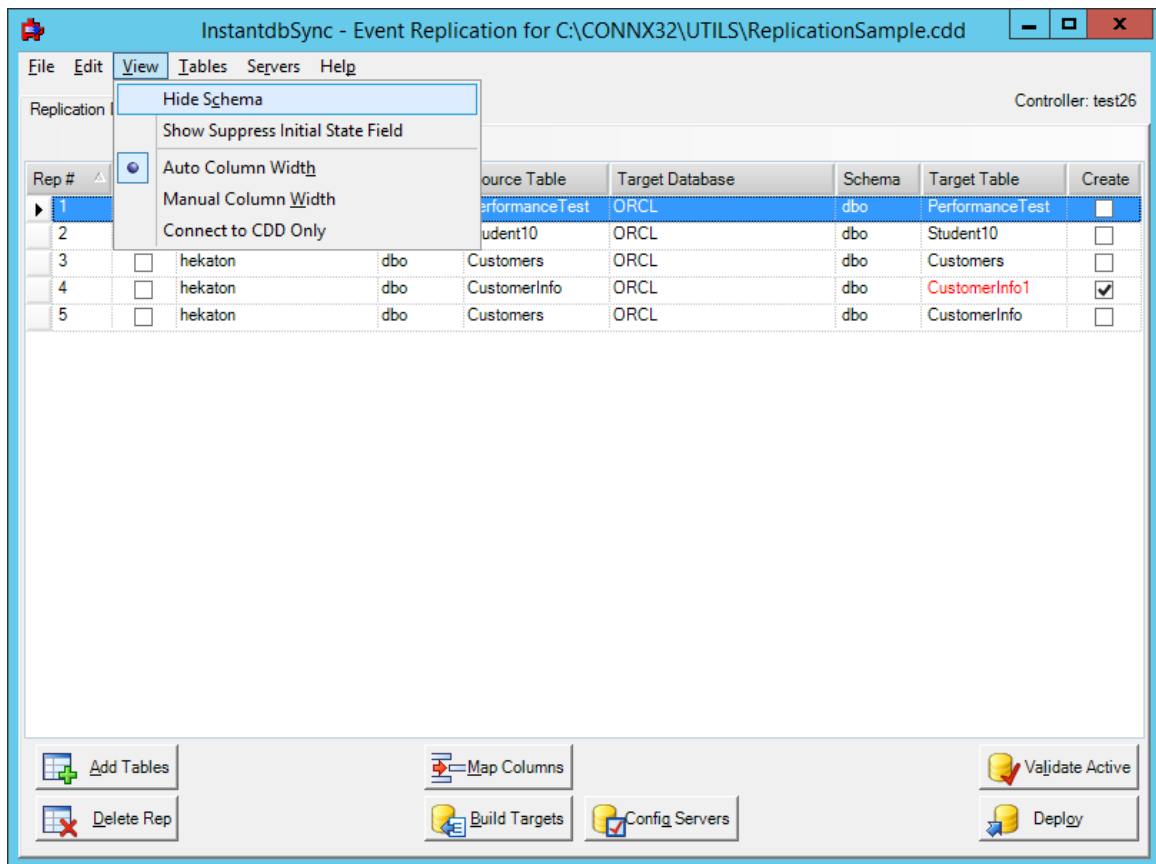
1. Open the InstantdbSync Replication Administrator.
2. From the **View** menu, click **Show Schema**.



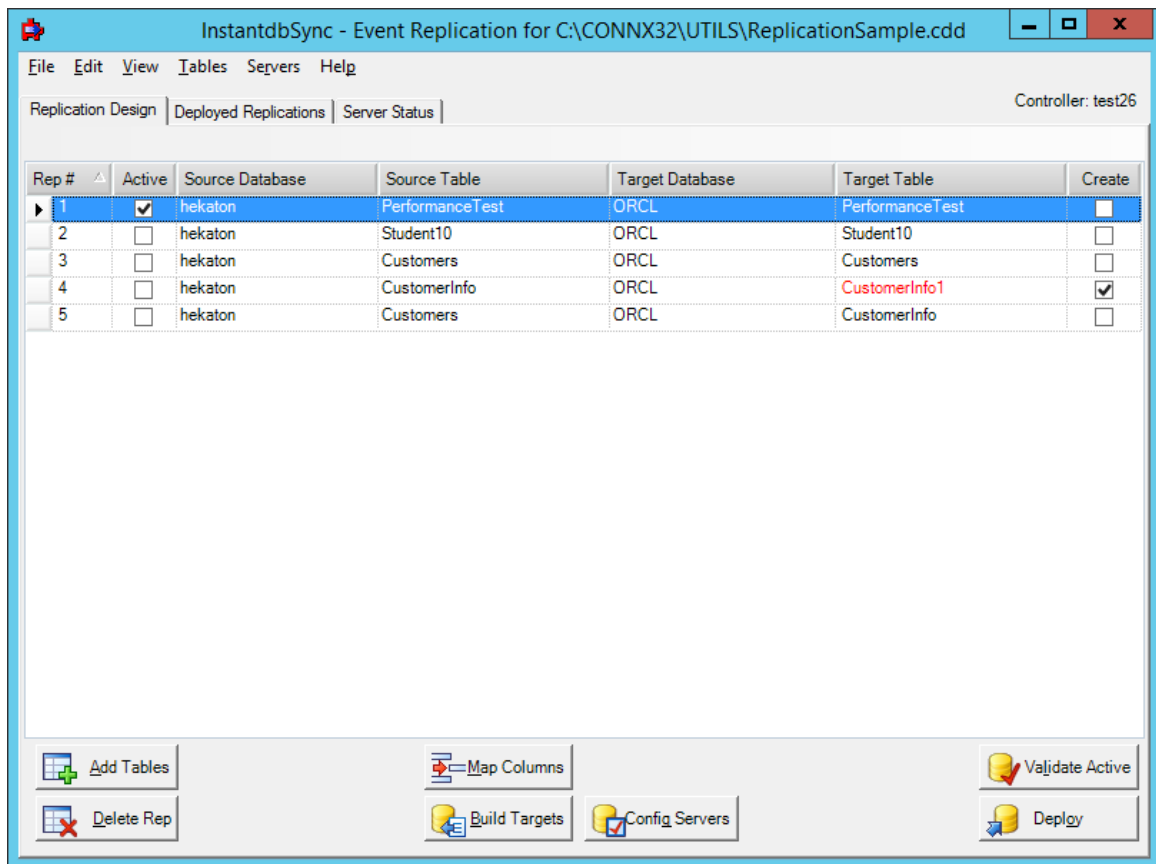
The **InstantdbSync** main window now contains an additional column.



- When you are done viewing the schemas, you can click **Hide Schema**.



This removes the schema column.



Configuring Replication and Source Database Servers

Configuring replication and source database servers includes:

- Opening a CONNX Data Dictionary (CDD)
- Entering the CONNX login credentials
- Specifying the replication server and its communication port
- Assigning a source database communication port

Configure the replication and source database servers if:

- The configuration information is missing, changed, or incorrect

After you configure the replication and source database servers you will be able to:

- Add source tables to be replicated
- Deploy the replication (which will start replicating data from source database to target database)

1. Open the InstantdbSync Administrator.
2. Click **Config Servers**. The **Configure Servers** window appears.

InstantdbSync - Configure Servers

CONNX Logon Credentials

UserName

Password

Replication Server

Name/Address Port

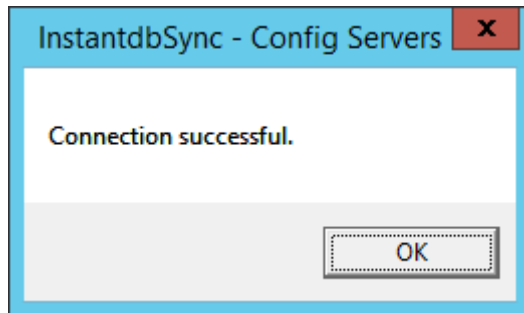
Parallel transaction count

Select Source Database Port

Database Name Port

EP Server Name

3. Enter your User Name and Password in **CONNX Logon Credentials** and click **Test Connection**. If the data is valid, the following message window appears:



4. Click **OK**. The message window closes.
5. To configure a replication server, enter the server name or IP address of the Replication Server in **Name/Address** and the server port number in **Port**.
Parallel transaction count specifies the number of transactions that will be processed in parallel. For initial state processing, this will be the number of simultaneous initial states that are running. Although the optimum value for this field is dependent on the available memory and speed of the server, a general guideline is to set this value to 2x the number of processor cores.

 A screenshot of a Windows-style dialog box titled "InstantdbSync - Configure Servers". The dialog has a blue header bar with a red close button (X) on the right. The main area is light gray and contains three sections:

- CONNX Logon Credentials**: Contains a "UserName" field with "testuser", a "Password" field with masked characters, and a "Test Connection" button.
- Replication Server**: Contains a "Name/Address" field with "test26", a "Port" field with "9200", and a "Parallel transaction count" spinner box set to "8".
- Select Source Database Port**: Contains a "Database Name" dropdown menu with "hekaton", a "Port" field with "9200", and an "EP Server Name" field with "test81". There is also an "Apply to All" button.

 At the bottom of the dialog are "Done" and "Cancel" buttons.

Note: The default port number is 9200.

6. To set the server port number on a single Source Database, select the Source Database from the list, enter the server port number in **Port**, and click **Set Port**.

Note: Do not click **Apply to All** if you only want to set the port number on a single Source Database.

7. To set the same server port number for all the databases, enter the server port number in **Port**, and click **Apply to All**. You do not have to select any Source Databases.

Note: You cannot add a Source Database here. To add a Source Database to the CONNX Data Dictionary, use the CONNX Data Dictionary Manager. For more information about the CONNX Data Dictionary Manager, see **CONNX Basics - Working with CDDs** in the **CONNX User Reference Guide**.

8. Click **Done**.

Viewing Server Status

Viewing server status includes:

- Opening a CONNX Data Dictionary (CDD)
- Viewing the current state of all the replication servers.

View the server status if:

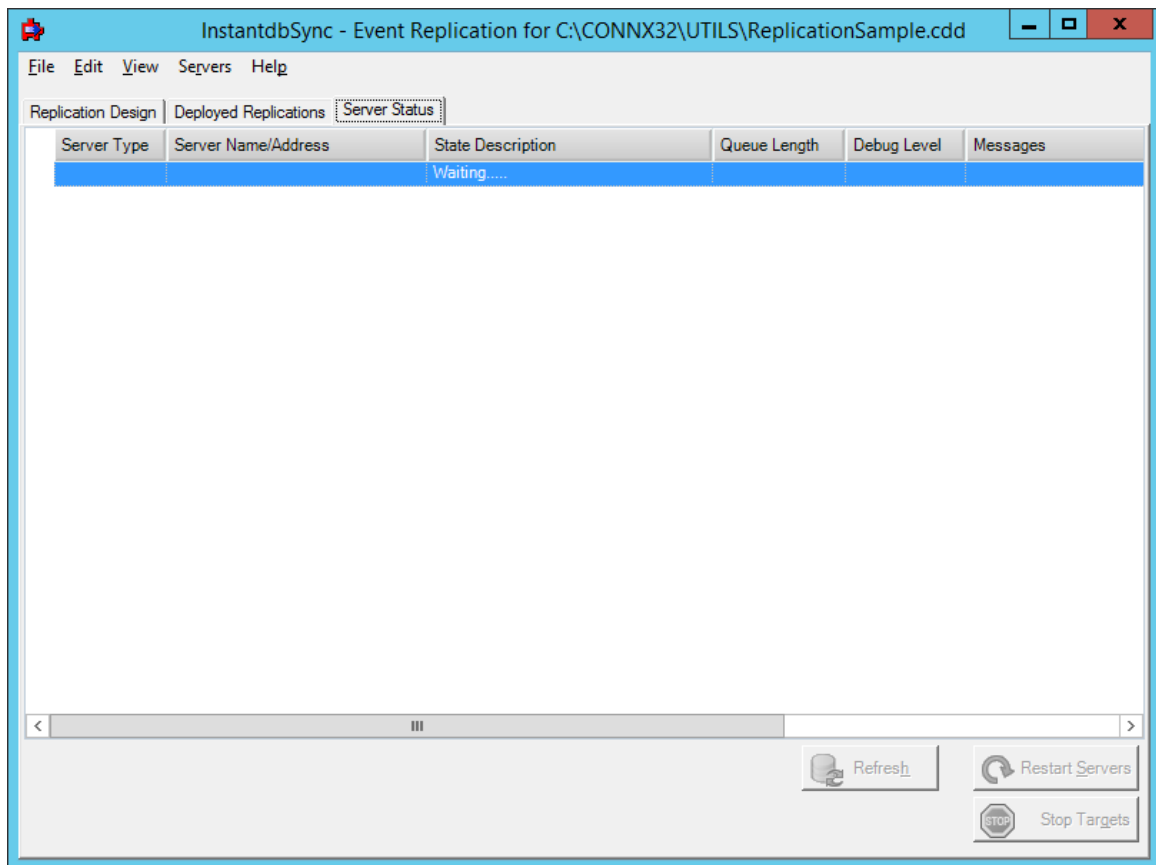
- The log file contains error messages about a server.

After you view the server status you will be able to:

- Stop the replication servers
- Restart the replication servers

You may want to view the server status if you receive an error message about a server.

1. Open the InstantdbSync Replication Administrator.
2. Click the **Server Status** tab. The following appears:



When the server information has been retrieved, the following displays:

Server Type	Server Name/Address	State Description	Queue Length	Debug Level	Messages
Consumer	hekaton - ORCL	Replicating	0	None	
Replication		State Description	Initial State Start	Initial State End	Total (HH:MM:SS)
Customers --> CustomerInfo		Replicating	2014-04-21 14:33:15	2014-04-21 14:33:15	00:00:01
Student10 --> Student10		Replicating	2014-04-21 14:33:36	2014-04-21 14:33:36	00:00:01
Customers --> Customers		Replicating	2014-04-21 14:33:25	2014-04-21 14:33:26	00:00:01
PerformanceTest --> PerformanceTest		Replicating	2014-04-21 14:33:04	2014-04-21 14:33:05	00:00:01
Producer	hekaton	Replicating	N.A.	None	
Controller	test26	Replicating	N.A.	None	

This grid contains the status for all the Replication components. The columns and their meaning are as follows:

- **Server Type**
- **Server Name/Address**

This is the name of the server that the corresponding component is installed on.

- **State Description**

This tells the current state of the component or replication.

- **Queue Length**

This indicates the number of elements in the queue. In normal operation, this number will grow and shrink depending on load. A rapid increase in this number that never goes down is an indication that the target database is offline and transactions are queuing but not being processed on the target.

- **Debug Level**

This indicates the debug level of each component

- **Messages**

If there are any error messages associated with this component, they will be displayed here. **Hint:** Double click on this field to display the entire message in an edit box for easier reading.

The server status information will automatically refresh every 30 seconds. To manually refresh the server status, click the **Refresh** button.

The **Time** field in the upper right corner will contain the latest refresh time.

You can also [adjust the display column width](#).

For more information about server status and a full description of the potential State Description values, see [Server Status Tab](#).

Building New Target Database Tables

Building new target database tables includes:

- Opening a CONNX Data Dictionary (CDD)
- Creating target database tables

Build new target database tables if:

- You wish to create target database tables without deploying replications.

After building new target database tables you will be able to:

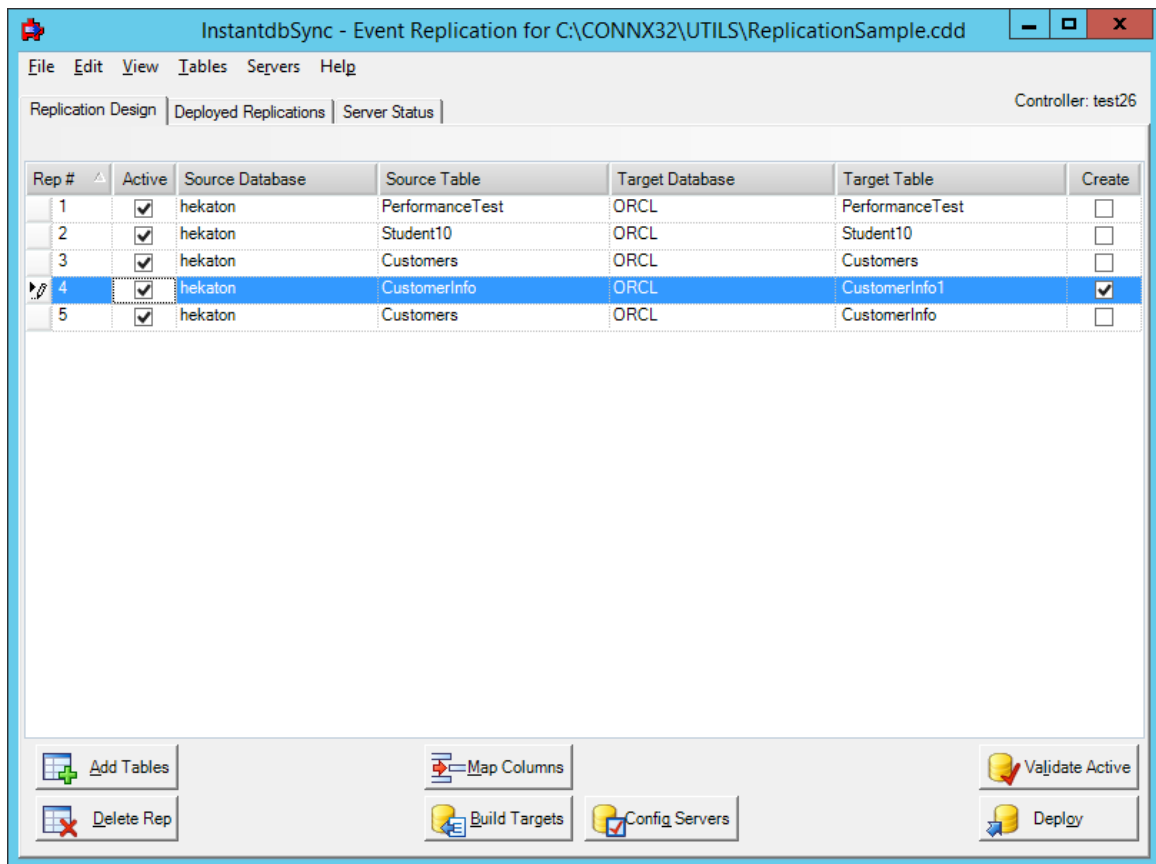
- Deploy the replication (which will start replicating data from source database to target database)

You can build new target tables without deploying a replication.

To add an existing target database tables to the Event Replicator, see [Adding a Table to the Event Replicator](#).

If a target table is new and does not exist on the target database, the Target Table name will be **red** and its row will have a check in the **Create Table** column.

1. Open the InstantdbSync Replication Administrator.
2. Go to the target table row, click **Active** and click **Build Targets**. This will create the table on the target database without deploying the replication.



A dialog box showing status will be displayed. If there are any errors, those errors will be displayed in this dialog box.

Dropping and Recreating Target Database Tables

Dropping and recreating new target database tables includes:

- Opening a CONNX Data Dictionary (CDD)
- Dropping selected target tables
- Recreating selected target tables

Drop and recreate new target database tables if:

- The target data is not valid
- The target data is out-of-date
- It's easier to re-create the target data than try to fix it before deployment

After dropping and recreating new target database tables you will be able to:

- Deploy the replication (which will start replicating data from source database to target database)

There are two ways to drop and recreate tables on the target database without deploying a replication.

If a target table exists on the target database, there is no check in the **Create Table** column. To drop and recreate a target table:

1. Open the InstantdbSync Replication Administrator
2. Go to the target table row, click **Active**, click **Create**, and click **Build Targets**.

This will drop and recreate the table on the target database without deploying the replication.

If the purpose of dropping the target table is to change the column mapping, the target table can also be dropped from within the Map Columns Dialog by pressing the Drop Target Table button. (this button is only visible when mapping to an existing target table) After dropping the table and re-mapping the columns, press the Done button to return to this screen. The Create check box will be checked. Press Build Targets to rebuild the target table.

Caution: Drop and recreate will delete all data in the target table.

Deploying Replications

Deploying the event replication includes:

- Opening a CONNX Data Dictionary (CDD)
- Marking replications to be deployed as active
- Deploying active replications to the replication controller

Deploy the event replication if:

- You want to begin actively replicating your source data to the target databases

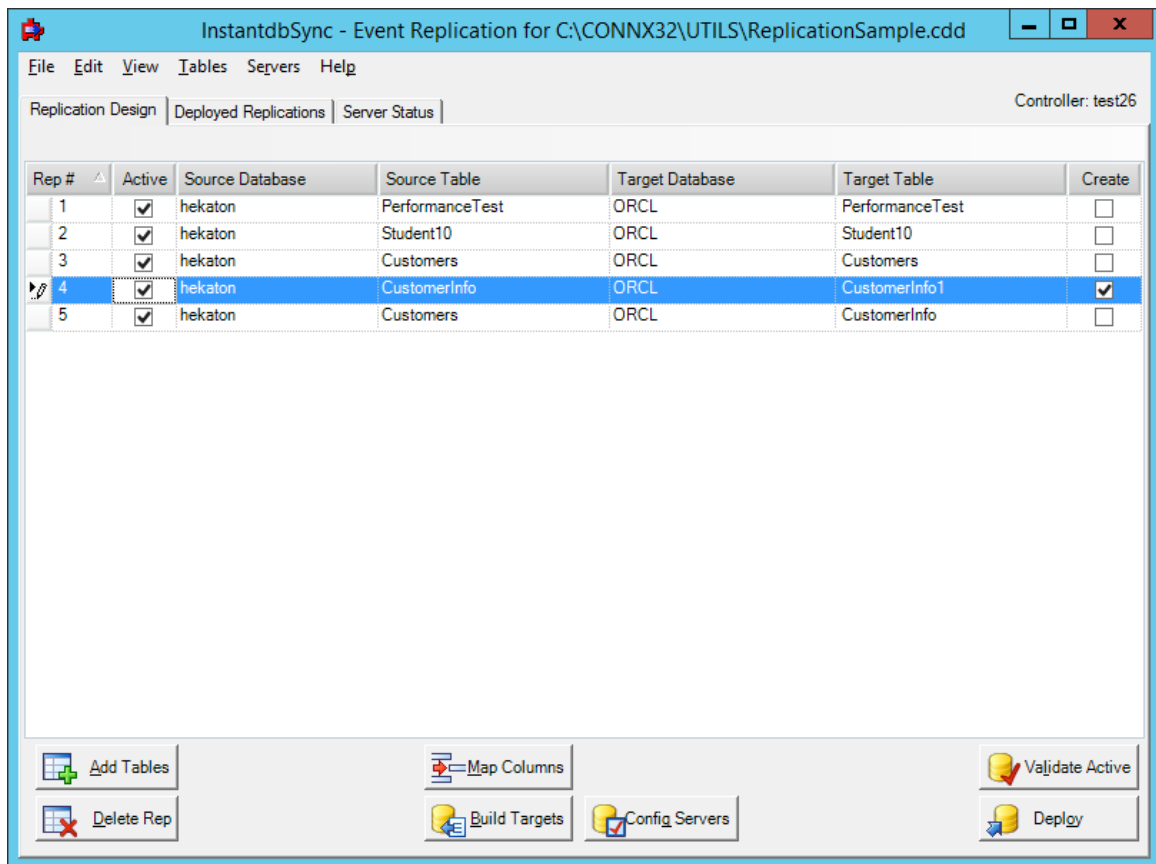
After deploying the event replication you will be able to:

- Check server status

All changes made using the InstantdbSync Replication Administrator must be deployed to the appropriate replication server for them to take effect.

Note: If the event producer and controller were installed on different machines, start the message queue on both machines (Windows or UNIX) before deploying a replication.

1. Open the Open Systems Event Replicator CDD.
2. To deploy a replication, click **Active** in that row. If the target table does not exist or you want to drop and recreate it, check **Create** in that row as well.



3. Validate that the active replications are ready to be deployed.
4. Click **Deploy**. A status dialog will appear displaying each step and status of the deployment process. When the deploy is complete there will either be message showing that the system is replicating or processing initial states. If there are any errors during the deploy process, they will be displayed in this message box. For mor information on status after a deploy, go to the Server Status tab.

Performing an Initial State

Performing an Initial State includes:

- Opening a CONNX Data Dictionary (CDD)
- Moving a copy of the source database into the target database

Perform an Initial State if:

- The target database is corrupt
- An unrecoverable error occurred

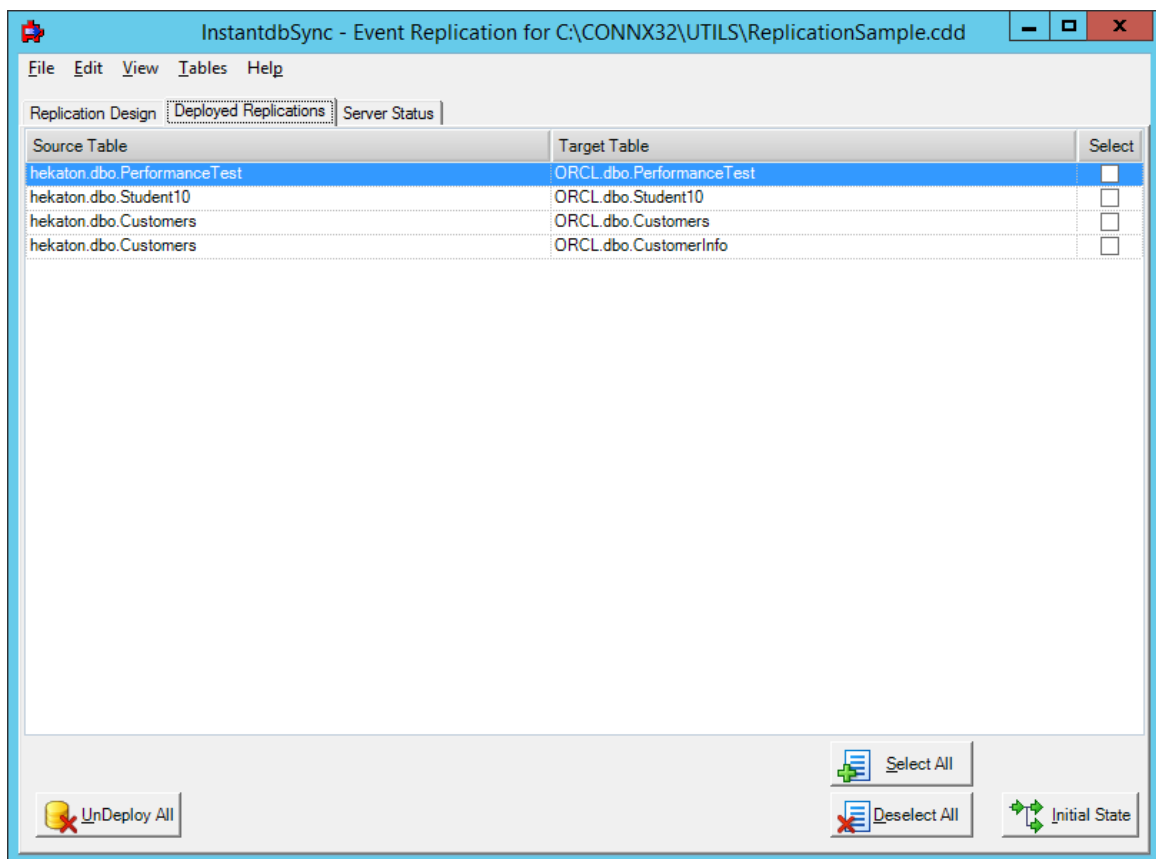
After performing an Initial State you will be able to:

- Deploy the Event Replicator (which starts replicating data from source database to target database)

Initial State duplicates the data in the source database and inserts it into the target database.

Initial State replaces the entire contents of the selected target tables with the current version of the source tables.

1. Open the InstantdbSync Replication Administrator.
2. Click the Deployed Replications tab.



3. Click the **Select** check box for the target tables you wish to overlay with the current source table contents and click **Initial State**. If the initial state request fails, a dialog box will displayed saying the request failed.

Note: This process describes a manual Initial State request. During the deploy process, an initial state will be done automatically if:

1. This replication was not deployed previously.
2. The initial state failed due to an error on a previous deploy.

If either of these two situations is true, the replication or replications that meet these criteria will have an initial state performed. If a replication was previously deployed and you are doing another deploy for the purpose of adding new replications, the previously deployed replications will not have the initial state process repeated.

Suppressing an Initial State

Suppressing an Initial State includes:

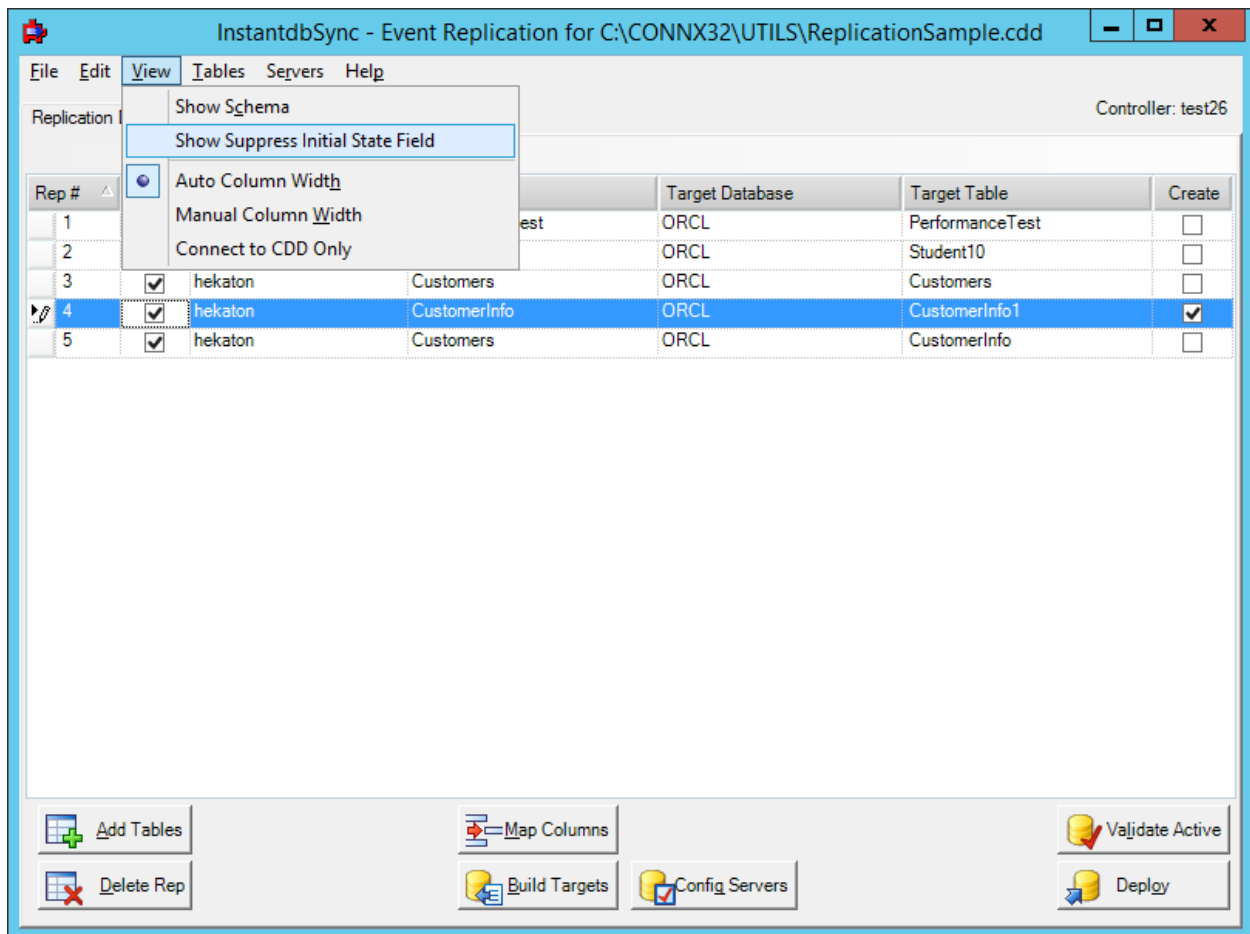
- Opening a CONNX Data Dictionary (CDD)
- Moving a copy of the source database into the target database

Suppress an Initial State if:

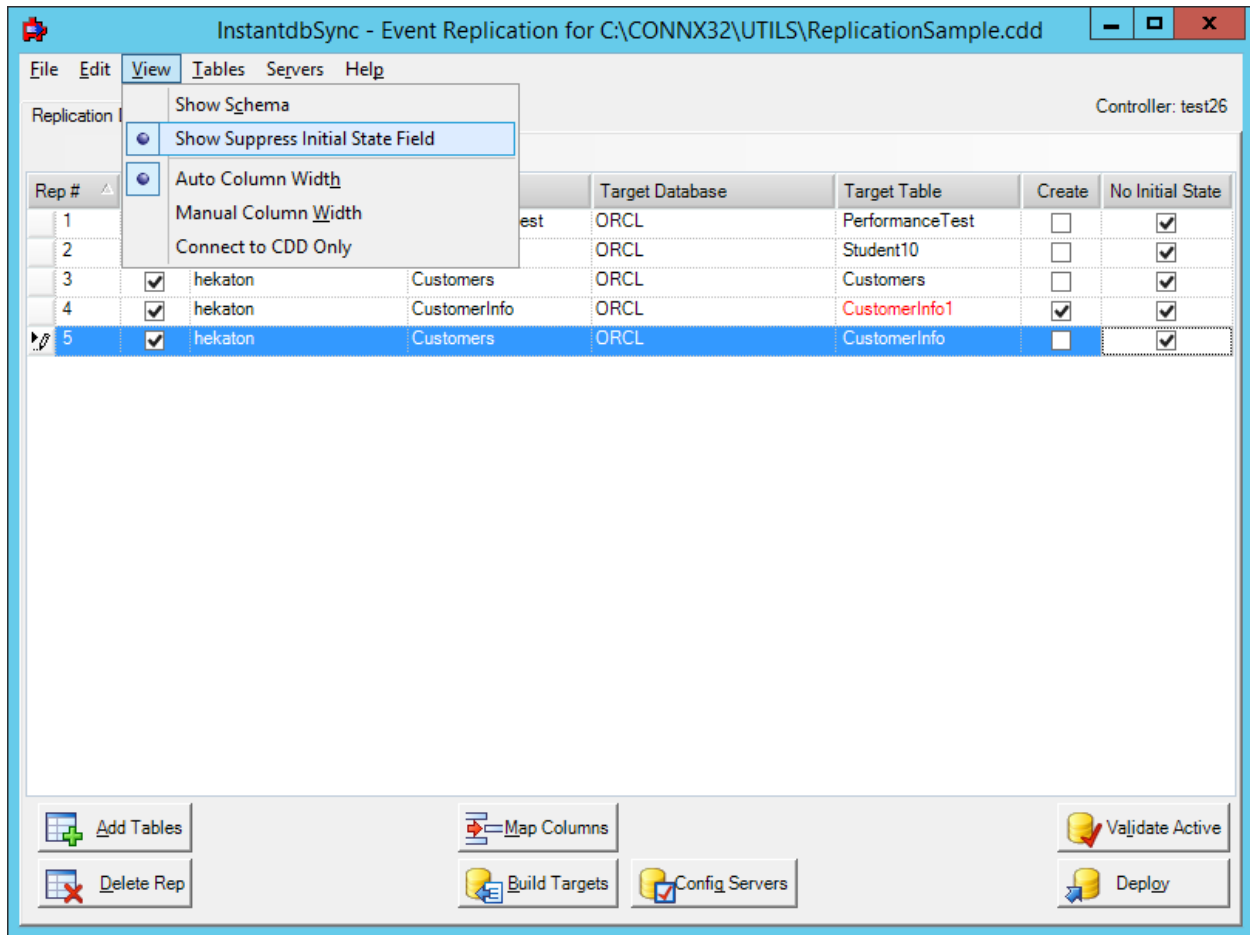
- An initial state has already been performed and the replication has been undeployed
- The target table has been synchronized with the source with an outside tool
- The system experienced a crash and you need to bring replication back on-line as soon as possible and you can tolerate some inconsistencies between the source and target tables.

Important note: Suppressing an initial state is an **Advanced** function and should only be done if you can guarantee that the target table is an **exact** copy of the source table. If the initial state process is not done, it is possible that the source and target tables will not contain the same records and errors may occur during the normal replication process. If this occurs, it is strongly recommended that an initial state be performed. In the case of recovering after a system wide failure, it is possible that the Replication Server will be unable to recover its internal configuration files. In this case a redeploy is required. If the initial state is suppressed at this time, it is strongly recommended that a [manual initial state](#) be performed as soon as possible.

To prevent an Initial State from occurring when a replication is deployed, an advanced feature called No Initial State must be activated. To activate this feature, go to the View Menu on the Replication Design tab and select Show Suppress Initial State Field.



A new column named No Initial State will now be displayed on the right side of the screen.



Check this box for each replication you wish to suppress the initial state for and press the Deploy button. After the Deploy, no initial states will occur for the selected replications. After the deploy, any transactions that occur on the source table will be replicated to the target. If the target table is not in sync with the source table, errors will occur.

Un-Deploying all Replications

Un-Deploying all replications includes:

- Opening a CONNX Data Dictionary (CDD)
- Removing the CDD from the Controller
- Removing all active replications

Un-Deploy all event replications if:

- The target database will be out of service for an extended period
- You no longer want to use this target database

After un-deploying all replication you will be able to:

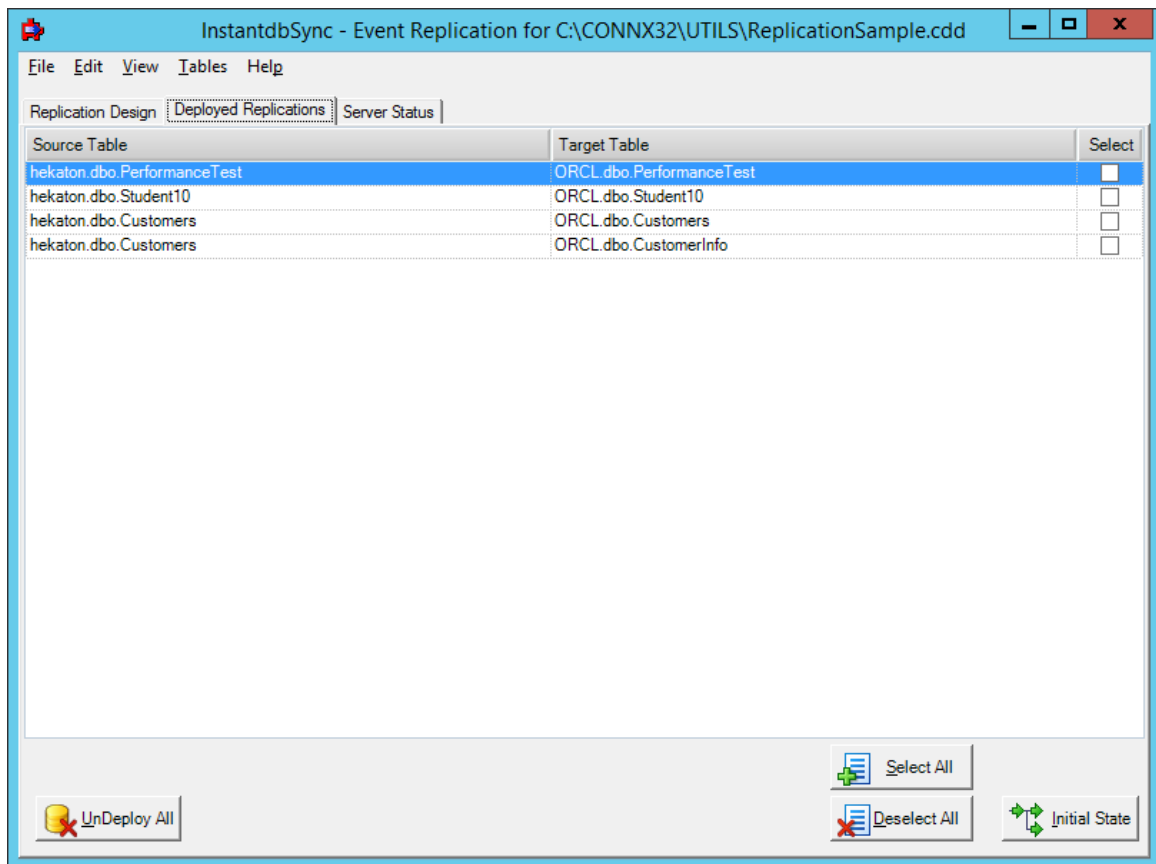
- Do maintenance on the target database
- Replicate to a new target database
- Use a different Controller
- Create a new CDD and use the new CDD for replication

Un-deploy All is the only way to un-deploy all active replications at once. You cannot un-deploy all active replications by clearing all the **Active** check boxes on the Replication Design tab.

Un-deploy All un-deploys the entire CDD, not just the individual replications.

If you don't want to un-deploy all active replications, in the Replication Design tab, you can select those replications you wish to un-deploy, clear **Active**, and click **Deploy**.

1. Open the InstantdbSync Replication Administrator.
2. Click the Deployed Replications tab.



- Click **Un-Deploy All**. A confirmation dialog will appear. click **Yes** to un-deploy.

Warning: Clicking **Yes** will un-deploy **ALL** replications whether you do or do not click the **Select All** button.

Stopping Event Replication

Stopping event replication includes:

- Opening a CONNX Data Dictionary (CDD)
- Stopping the target database from receiving replications

Stop event replication if:

- You wish to bring the server down to do any target database or server maintenance and bring the server right back up.
- You wish to save all the replications generated by the source database without performing an initial state.

After stopping event replication you will be able to:

- Perform maintenance on the target database
- Restarting event replication

To temporarily stop the Event Replicator without removing the deployed replications or losing any data that will be copied to the target tables, stop the target servers.

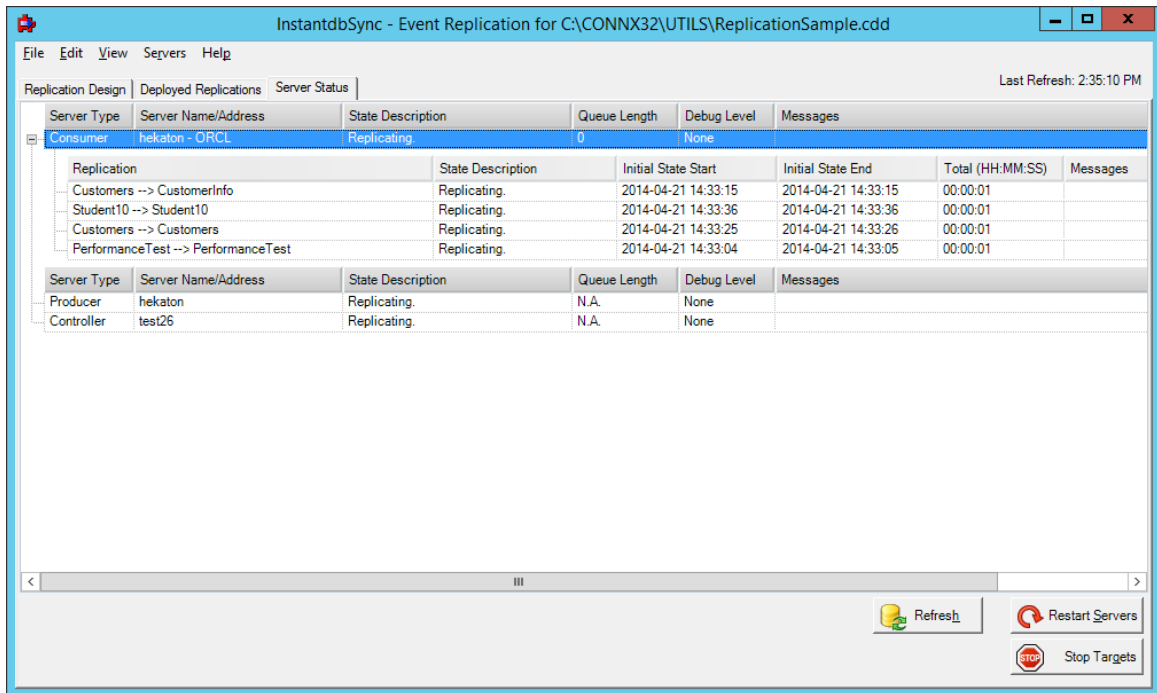
Stop the target servers before you do any target database or server maintenance.

Note: **Stop Targets** does not stop individual servers. All target servers listed in the **Server Status** tab will be stopped.

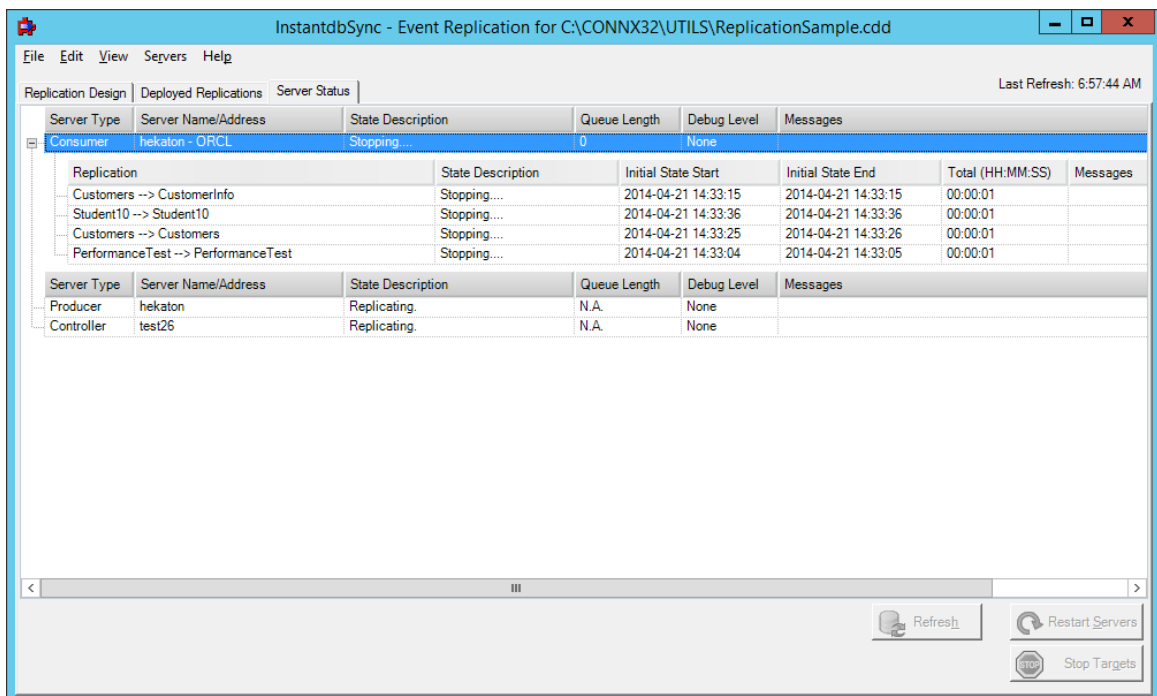
Note: **Stop Targets** does not stop source or replication servers.

- To stop source servers, go to the **Microsoft Management Console** and stop the **CONNX Replication Listener** service.
- To stop the replication server, go to the **Microsoft Management Console** and stop the **CONNX Replication Controller** service.

1. Open the InstantdbSync Replication Administrator.
2. Click the **Server Status** tab.



3. Click **Stop Targets**. While the target servers are being stopped, the following displays:



4. When the target servers have been stopped, the Consumer(s) and Replications will be in an Offline status:

CONNX InstantdbSync

InstantdbSync - Event Replication for C:\CONNX32\UTILS\ReplicationSample.cdd

File Edit View Servers Help

Replication Design Deployed Replications Server Status Last Refresh: 6:58:44 AM

Server Type	Server Name/Address	State Description	Queue Length	Debug Level	Messages																														
Consumer	hekaton - ORCL	Not Replicating - offline.	0	None																															
<table border="1"> <thead> <tr> <th>Replication</th> <th>State Description</th> <th>Initial State Start</th> <th>Initial State End</th> <th>Total (HH:MM:SS)</th> <th>Messages</th> </tr> </thead> <tbody> <tr> <td>Customers --> CustomerInfo</td> <td>Replication Offline.</td> <td>2014-04-21 14:33:15</td> <td>2014-04-21 14:33:15</td> <td>00:00:01</td> <td></td> </tr> <tr> <td>Student10 --> Student10</td> <td>Replication Offline.</td> <td>2014-04-21 14:33:36</td> <td>2014-04-21 14:33:36</td> <td>00:00:01</td> <td></td> </tr> <tr> <td>Customers --> Customers</td> <td>Replication Offline.</td> <td>2014-04-21 14:33:25</td> <td>2014-04-21 14:33:26</td> <td>00:00:01</td> <td></td> </tr> <tr> <td>PerformanceTest --> PerformanceTest</td> <td>Replication Offline.</td> <td>2014-04-21 14:33:04</td> <td>2014-04-21 14:33:05</td> <td>00:00:01</td> <td></td> </tr> </tbody> </table>						Replication	State Description	Initial State Start	Initial State End	Total (HH:MM:SS)	Messages	Customers --> CustomerInfo	Replication Offline.	2014-04-21 14:33:15	2014-04-21 14:33:15	00:00:01		Student10 --> Student10	Replication Offline.	2014-04-21 14:33:36	2014-04-21 14:33:36	00:00:01		Customers --> Customers	Replication Offline.	2014-04-21 14:33:25	2014-04-21 14:33:26	00:00:01		PerformanceTest --> PerformanceTest	Replication Offline.	2014-04-21 14:33:04	2014-04-21 14:33:05	00:00:01	
Replication	State Description	Initial State Start	Initial State End	Total (HH:MM:SS)	Messages																														
Customers --> CustomerInfo	Replication Offline.	2014-04-21 14:33:15	2014-04-21 14:33:15	00:00:01																															
Student10 --> Student10	Replication Offline.	2014-04-21 14:33:36	2014-04-21 14:33:36	00:00:01																															
Customers --> Customers	Replication Offline.	2014-04-21 14:33:25	2014-04-21 14:33:26	00:00:01																															
PerformanceTest --> PerformanceTest	Replication Offline.	2014-04-21 14:33:04	2014-04-21 14:33:05	00:00:01																															
Producer	hekaton	Replicating.	N.A.	None																															
Controller	test26	Replicating.	N.A.	None																															

< III >

Refresh Restart Servers Stop Targets

Restarting Event Replication

Restarting event replication includes:

- Opening a CONNX Data Dictionary (CDD)
- Start the replication server, the source database and the target database

Restart event replication if:

- A error that caused replication to stop has been corrected

After restarting event replication you will be able to:

- Check the server status

If the Event Replicator has been stopped, restart the target servers to enable replication.

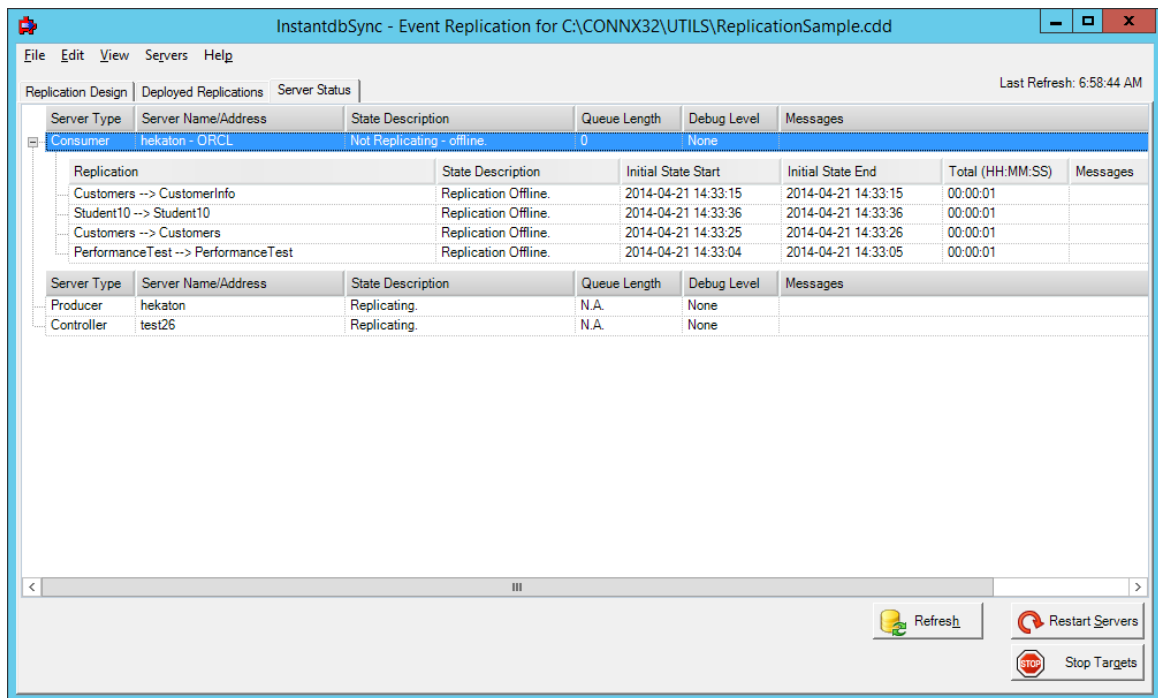
If a replication server or source database was stopped due to an error, **Restart Servers** can restart the replication server or source database once the error condition has been corrected. See the log file in the Replication Log Directory for more information.

All replications that have been processed while the servers have been stopped remain in the queue in the order received. After the servers have been started, the Event Replicator processes the replications in the order they appear in the queue.

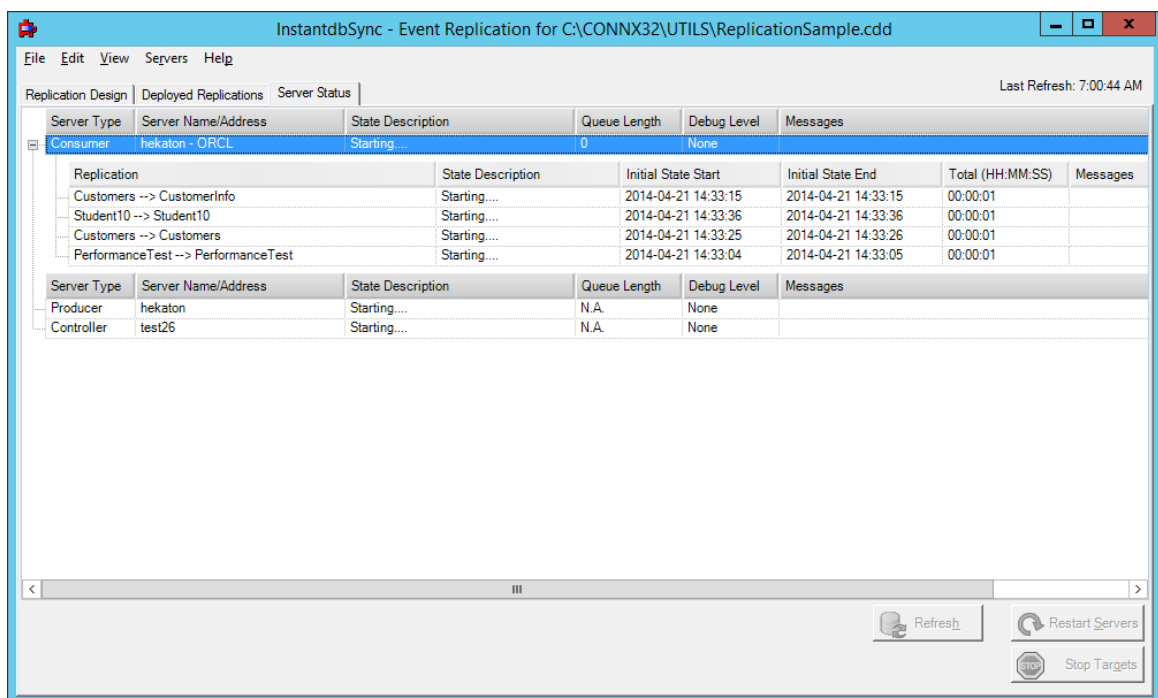
Note: **Restart Servers** does not start individual servers. All servers listed in the **Server Status** tab will be started.

1. Open the InstantdbSync Replication Administrator.
2. Click the **Server Status** tab.

CONNX InstantdbSync



3. Click **Restart Servers**.



When the servers have been started, the following appears:

InstantdbSync - Event Replication for C:\CONN32\UTILS\ReplicationSample.cdd

File Edit View Servers Help

Replication Design Deployed Replications Server Status Last Refresh: 7:01:44 AM

Server Type	Server Name/Address	State Description	Queue Length	Debug Level	Messages	
Consumer	hekaton - ORCL	Replicating.	0	None		
Replication		State Description	Initial State Start	Initial State End	Total (HH:MM:SS)	Messages
Customers --> CustomerInfo		Replicating.	2014-04-21 14:33:15	2014-04-21 14:33:15	00:00:01	
Student10 --> Student10		Replicating.	2014-04-21 14:33:36	2014-04-21 14:33:36	00:00:01	
Customers --> Customers		Replicating.	2014-04-21 14:33:25	2014-04-21 14:33:26	00:00:01	
PerformanceTest --> PerformanceTest		Replicating.	2014-04-21 14:33:04	2014-04-21 14:33:05	00:00:01	
Producer	hekaton	Replicating.	N.A.	None		
Controller	test26	Replicating.	N.A.	None		

III

Refresh Restart Servers Stop Targets

Starting, Stopping and Getting Status from the Event Server

On Windows:

On a Windows system, the Event Server and Message Queue are installed as Windows Services and will start automatically when Windows starts. If you wish to stop the Event Server or the Message Queue, go to the Windows Services manager, select the service you wish to stop and select Stop from the Action menu item.

On Unix:

On a Linux/Unix system, the Event Server and Message Queue can be started and stopped with start and stop parameter.

The syntax for the Message Queue is:

```
./mqserver [start|stop|status]
```

The syntax for the Event Server is:

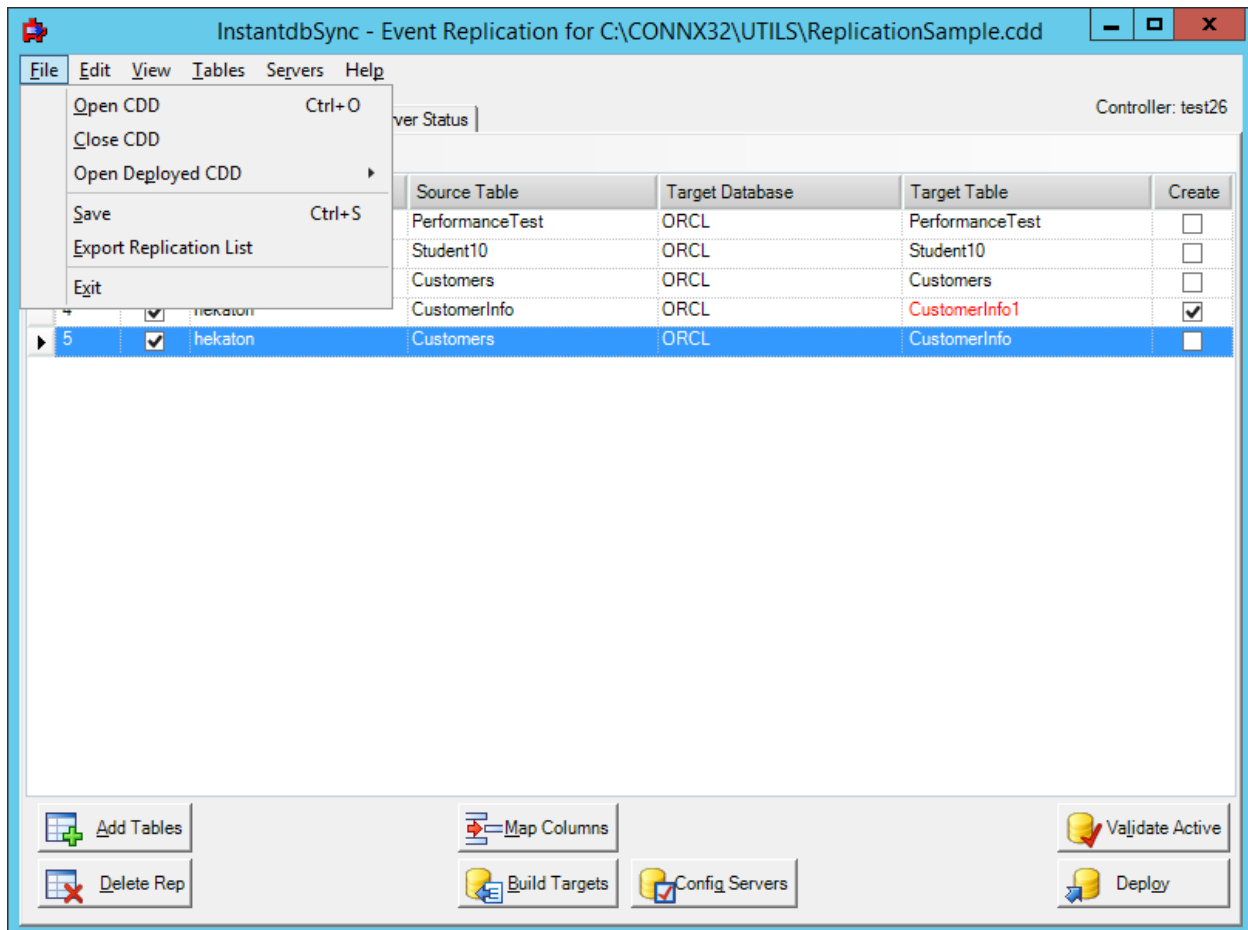
```
./eventserver [start|stop|status]
```

The status parameter displays whether or not the server is running. If the server is already running when you pass in the start parameter, the server will first be stopped and then restarted.

Chapter 3 - Event Replicator Menus and Tabs

File Menu

Applies to all tabs.

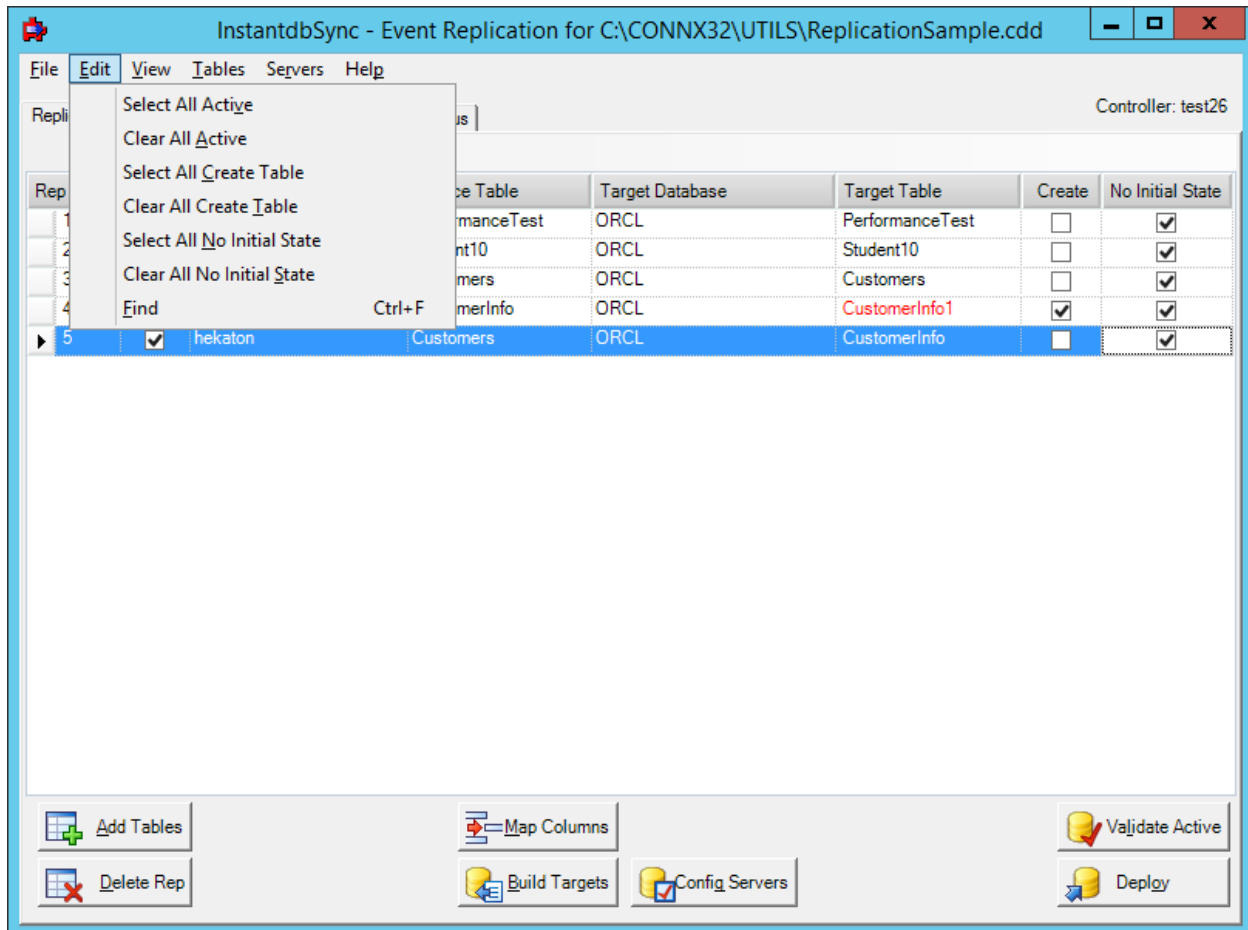


The File Menu has four commands:

- Open CDD (or Ctrl+O)
- Close CDD
- Open Deployed CDD
Contains a list of the deployed CDDs
- Save (or Ctrl+S)
Saves any replication changes generated since the last save or when replication began.
- Export Replication List
Exports the information in each row to a comma separated variable (.csv) file.
- Exit

Edit Menu

Applies to all tabs

Replication Design tab:

The Edit Menu has seven commands:

- **Select All Active**
Clicks the Active setting in all replications in this CDD
- **Clear All Active**
Clears the Active setting in all the replications in this CDD
- **Select All Create Table**
Clicks the Create Table setting in all replications in this CDD
- **Clear All Create Table**
Clears the Create Table setting in all the replications in this CDD
- **Select All No Initial State**
Sets all replications to suppress Initial State processing during a deploy. This is an advanced function and

will only be displayed when the option "Show Suppress Initial State Field" is selected from the View menu.

If No Initial State is selected, you must ensure the source and target tables are identical prior to starting replication. If they are not, errors will occur. This option is not displayed by default and must be turned on in the View menu.

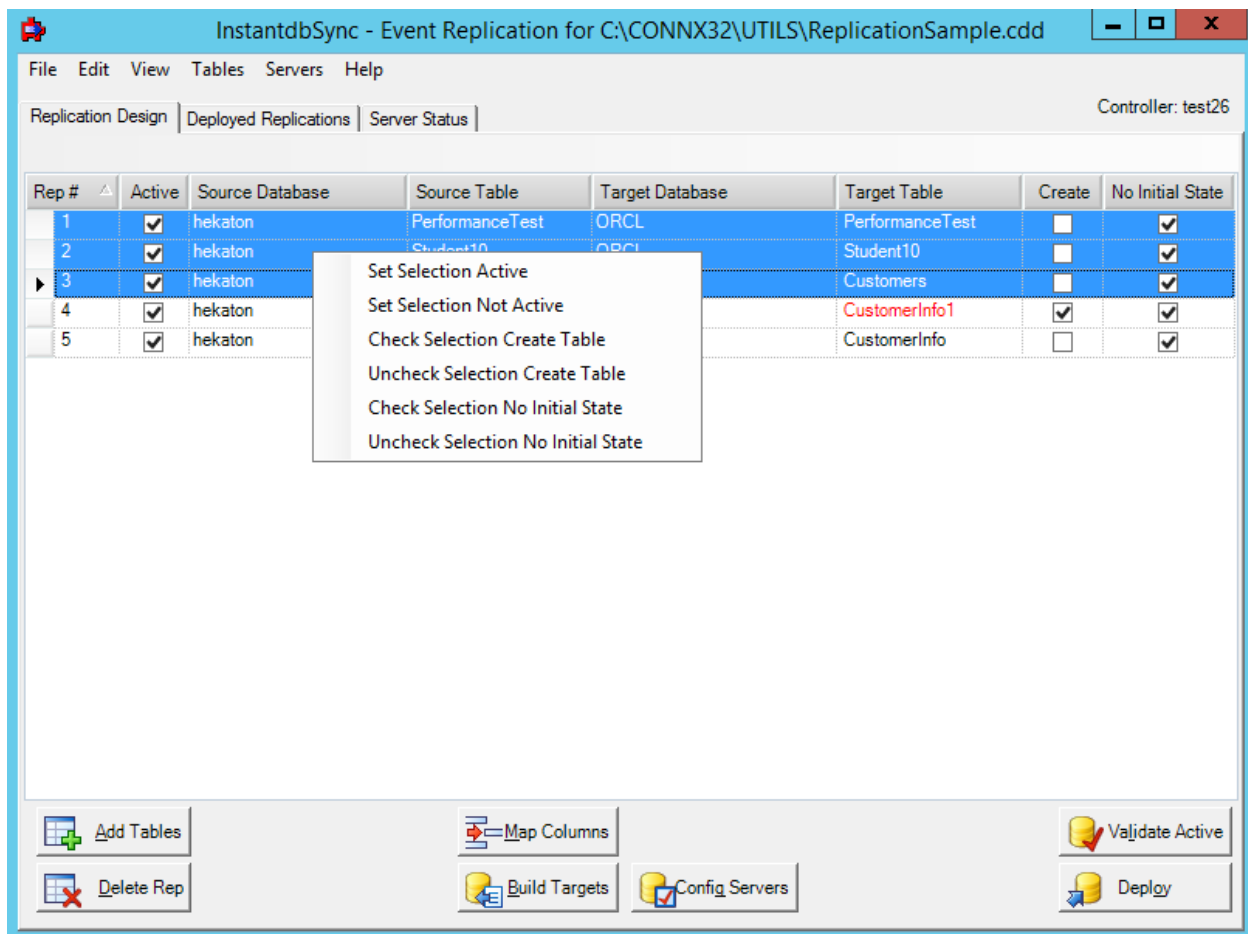
- **Clear All No Initial State**

Clears the No Initial State setting in all the replications in this CDD

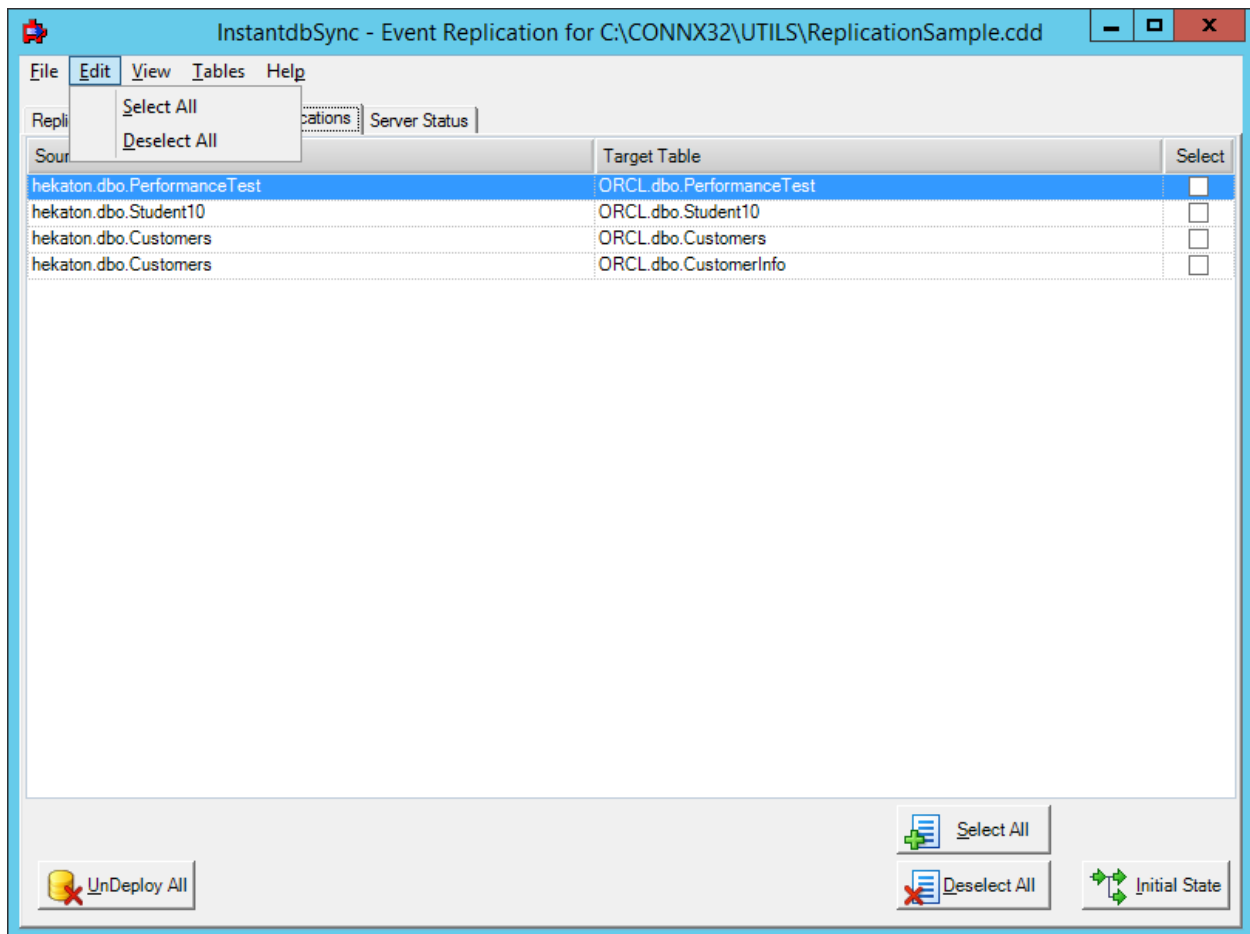
- **Find**

Brings up a Find dialog to allow searches for replications within this screen

There is also a Right Click context menu that contains the same options for Active, Create Table and No Initial State that allows the same functions to be performed on selected Replications



Deployed Replications tab:

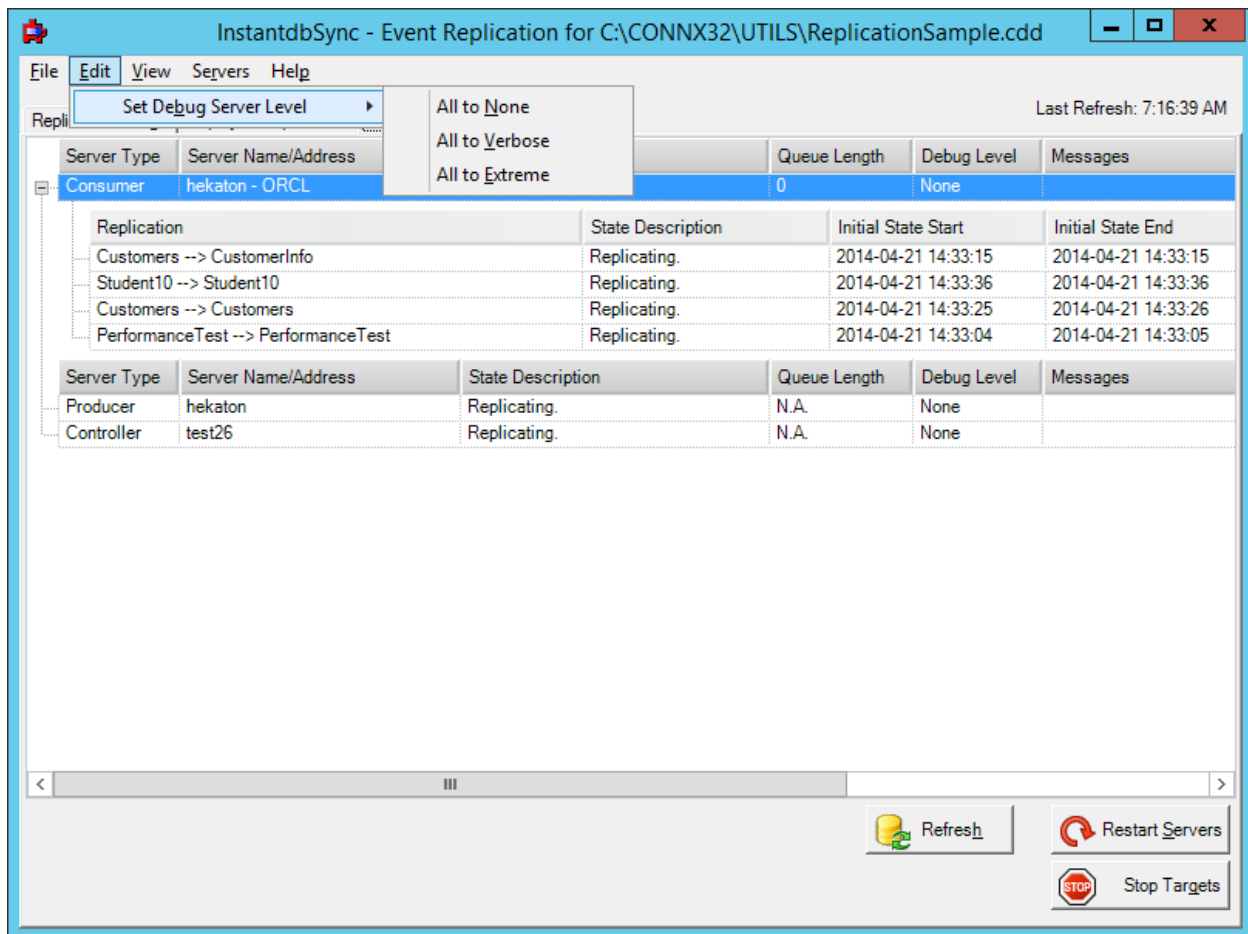


The Edit Menu has two commands:

- **Select All/Deselect All**

Check or clear the Select check box for all replications. The Initial State button acts on selected replications.

Server Status tab:



The Edit Menu has one command with three sub commands:

- **Set Debug Server Level**

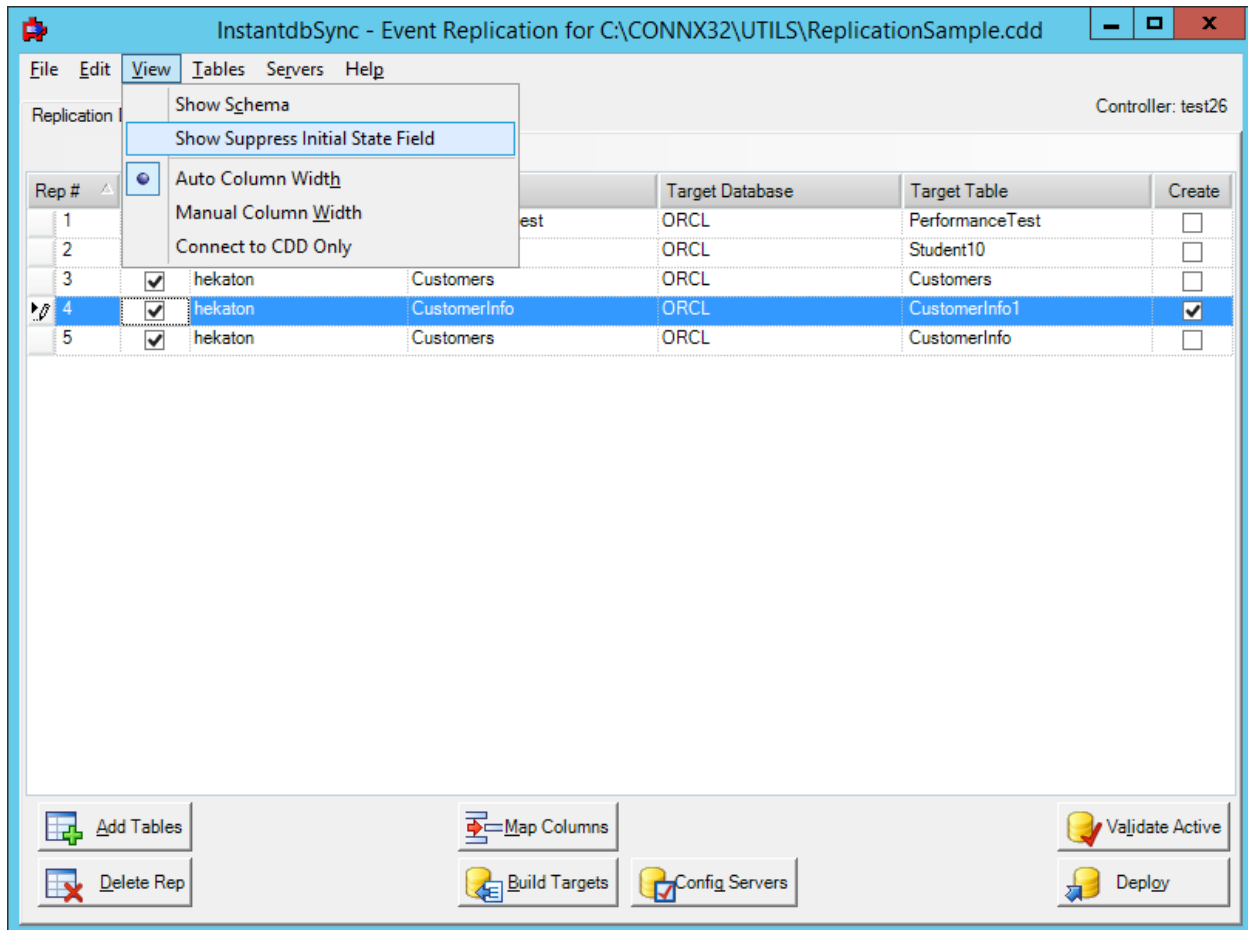
This sets the debug level for all the components without restarting the component. To change the debug level a component uses when it starts, change its DEBUG_LEVEL in the registry. See [Event Replication Registry Settings](#). There are three values for debug level:

- **None:** Minimal messages are written to the log. Errors are always written.
- **Verbose:** Diagnostic information is written when errors occur along with transaction information.
- **Extreme:** Complete diagnostics including buffer and transaction dumps are written to the log.
Note: This mode will cause a degradation in performance. Only use this mode when debugging an error situation.

Note: After changing the debug level, the status in the grid may not change right away. This is due to the fact that the system may not have made the change and reported the new status in time to be reflected in the current request for status. Pressing the Refresh button or waiting 30 seconds for the next automatic refresh will result in the new value being displayed.

View Menu

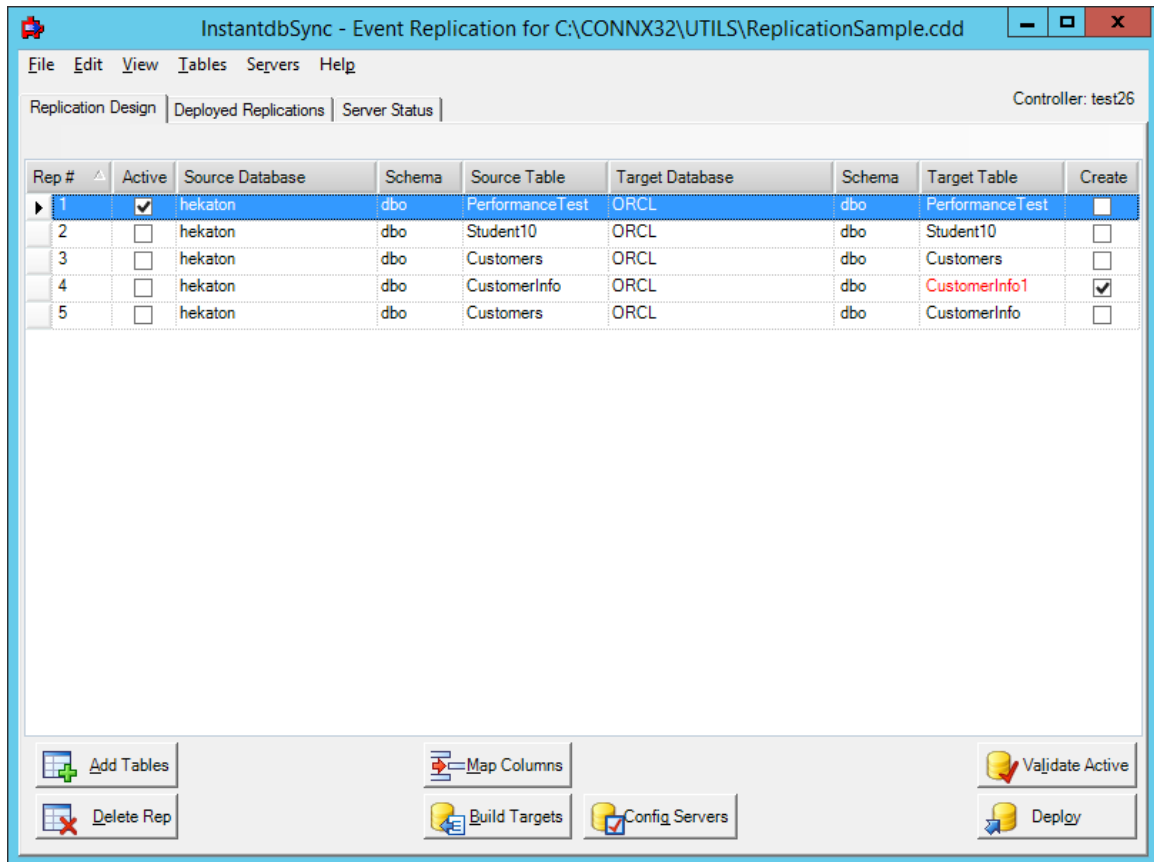
Applies to all tabs

Replication Design tab:

The View Menu has four commands:

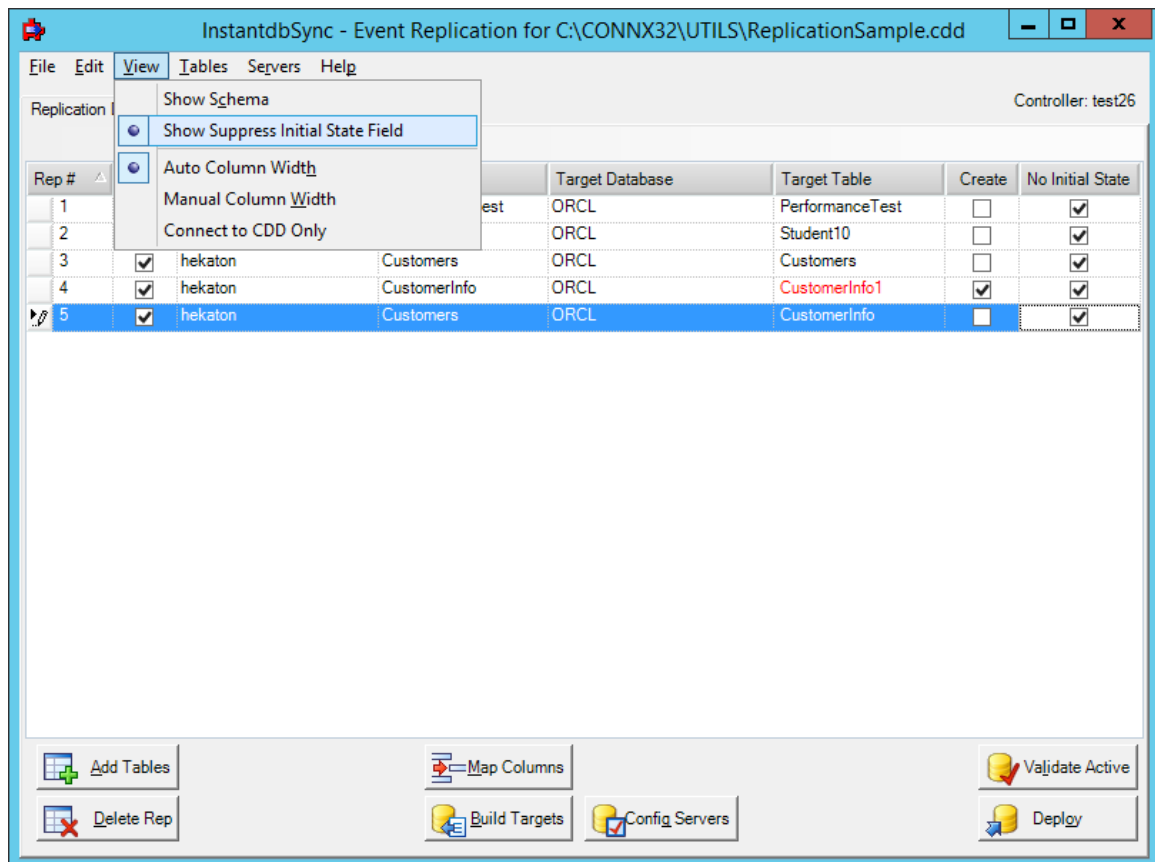
- **Show/Hide Schema**

Selecting Show Schema will add the column Schema to the grid. This column displays the Schema name for the each source table



- **Show Suppress Initial State Field**

This option will display a column in the grid which will allow the Initial State process to be suppressed when the replication is deployed. **Note:** This is an advanced function. If No Initial State is selected, you must ensure the source and target tables are identical prior to starting replication. If they are not, errors will occur. This option is not displayed by default.



- **Auto Column Width**

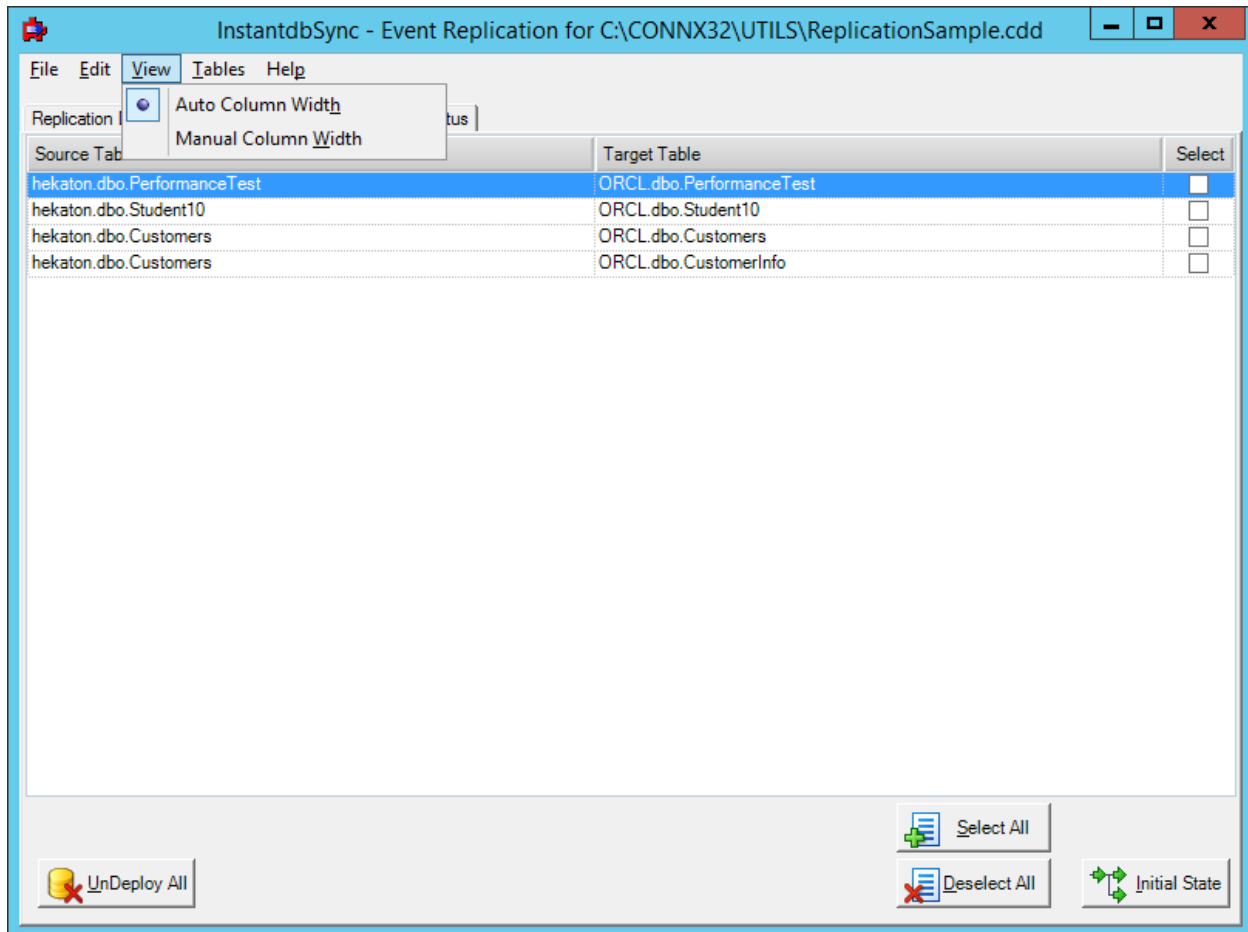
When selected the width of each column will be automatically calculated based on the width of the application window

- **Manual Column Width**

When selected the width of the columns will not be updated when the size of the application window changes. This option will be automatically selected whenever a column width is manually changed. If you wish to reset the width of the columns to fill the window after manually sizing them, select Auto Column Width from the menu.

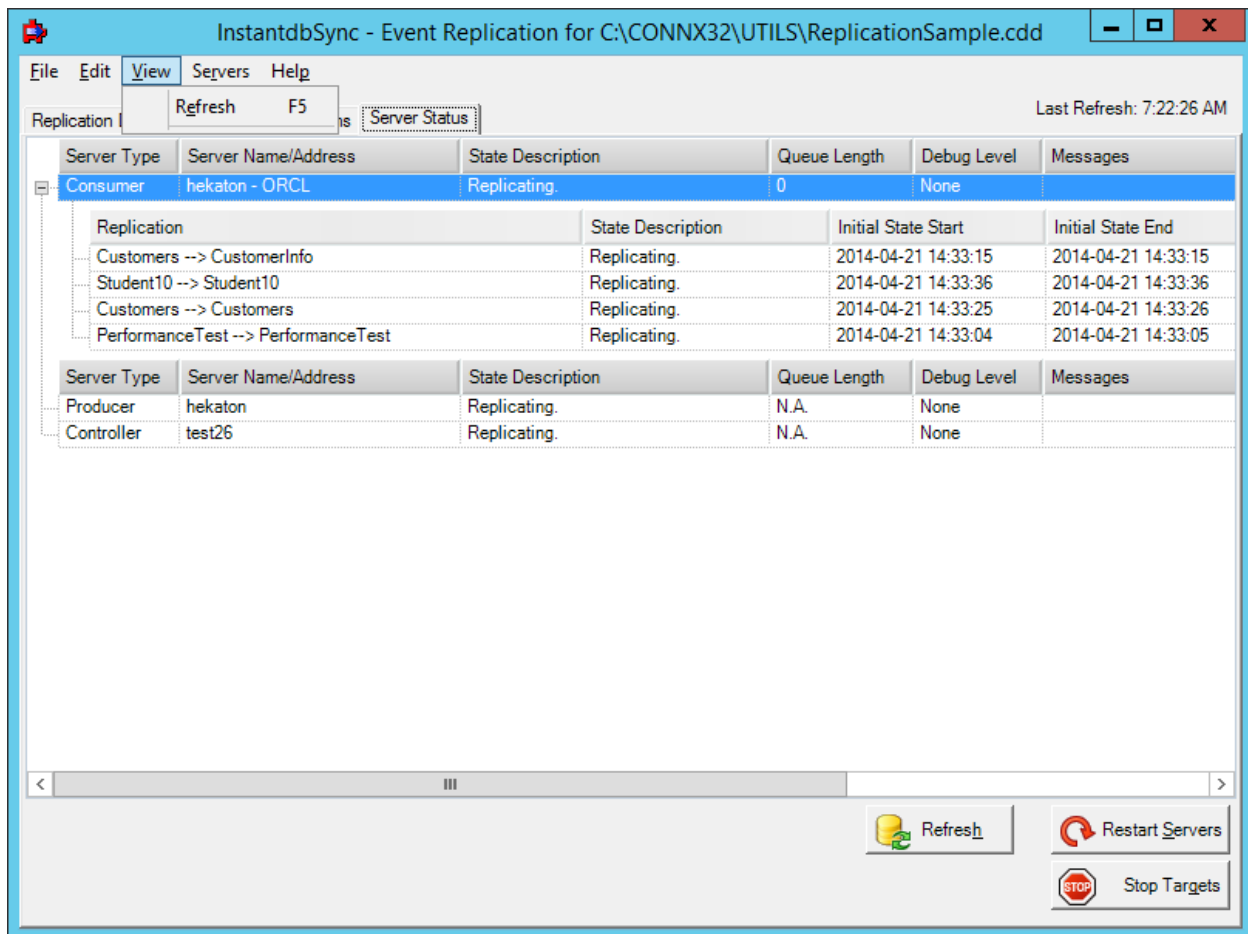
For more information see [Adjusting Display Column Width](#).

Deployed Replications tab:



The View Menu has two commands, Auto Column Width and Manual Column Width. These two commands work the same as on the Replication Design tab.

Server Status tab:



The View Menu has one command:

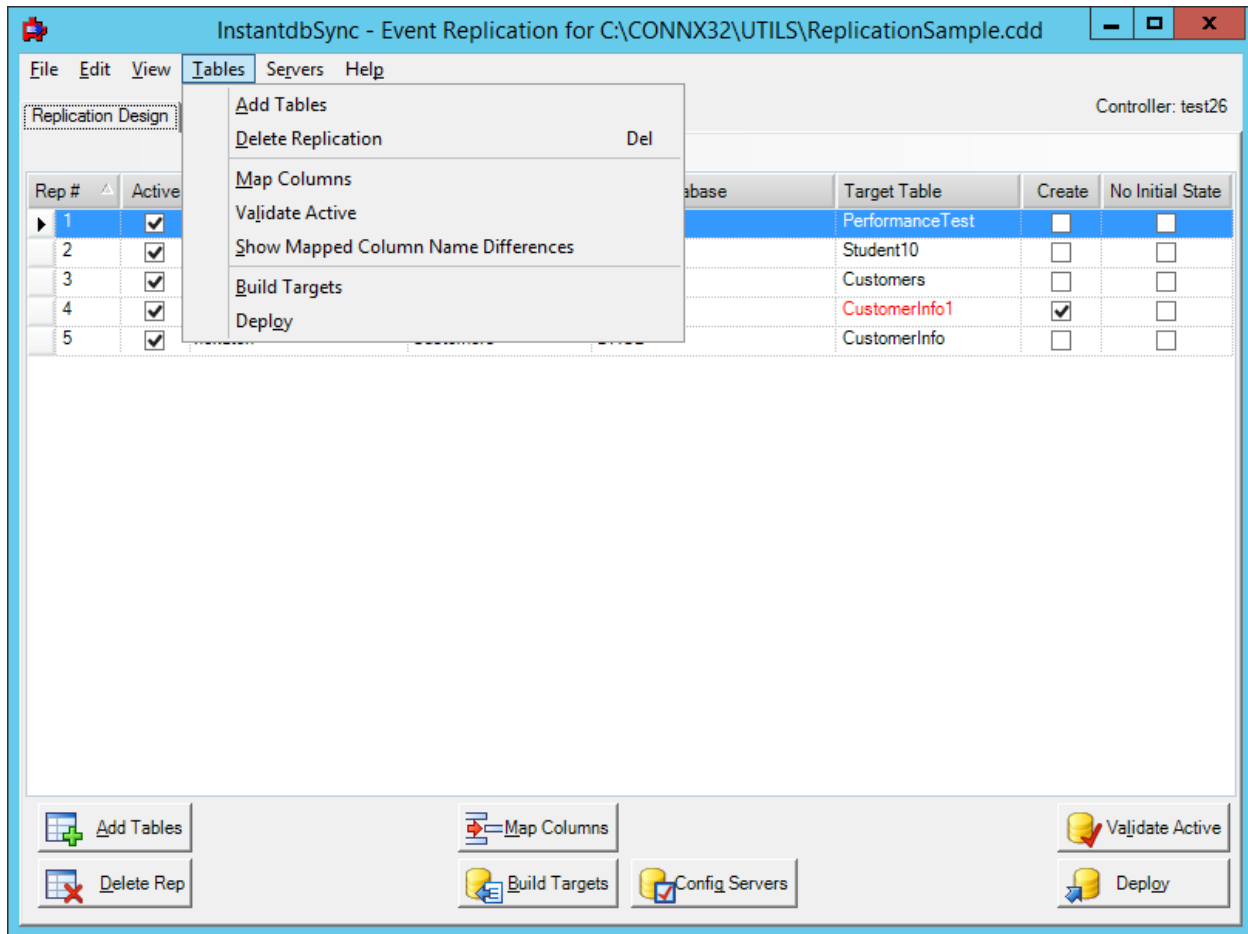
- **Refresh**

This option will refresh the screen with the current status. By default, the status will be refreshed automatically every 30 seconds. To get status prior to the automatic refresh, select this command from the menu or press the Refresh button at the bottom of the screen.

Tables Menu

Applies to Replication Design and Deployed Replications tabs.

Replication Design tab:



The Tables Menu on the Replication Design tab has six commands:

[Add Tables](#)

- Brings up the Add Tables dialog

[Delete Replication](#)

- Deletes the selected replication or replications

[Map Columns](#) (maps to either new or existing Target Tables depending on the replication)

- Brings up the Map Columns dialog

[Validate Active](#)

- Validates the replications that are marked active

[Show Mapped Column Name Differences](#)

- Shows a list of all columns in a replication where the source and target columns are not the same.

[Build Targets](#)

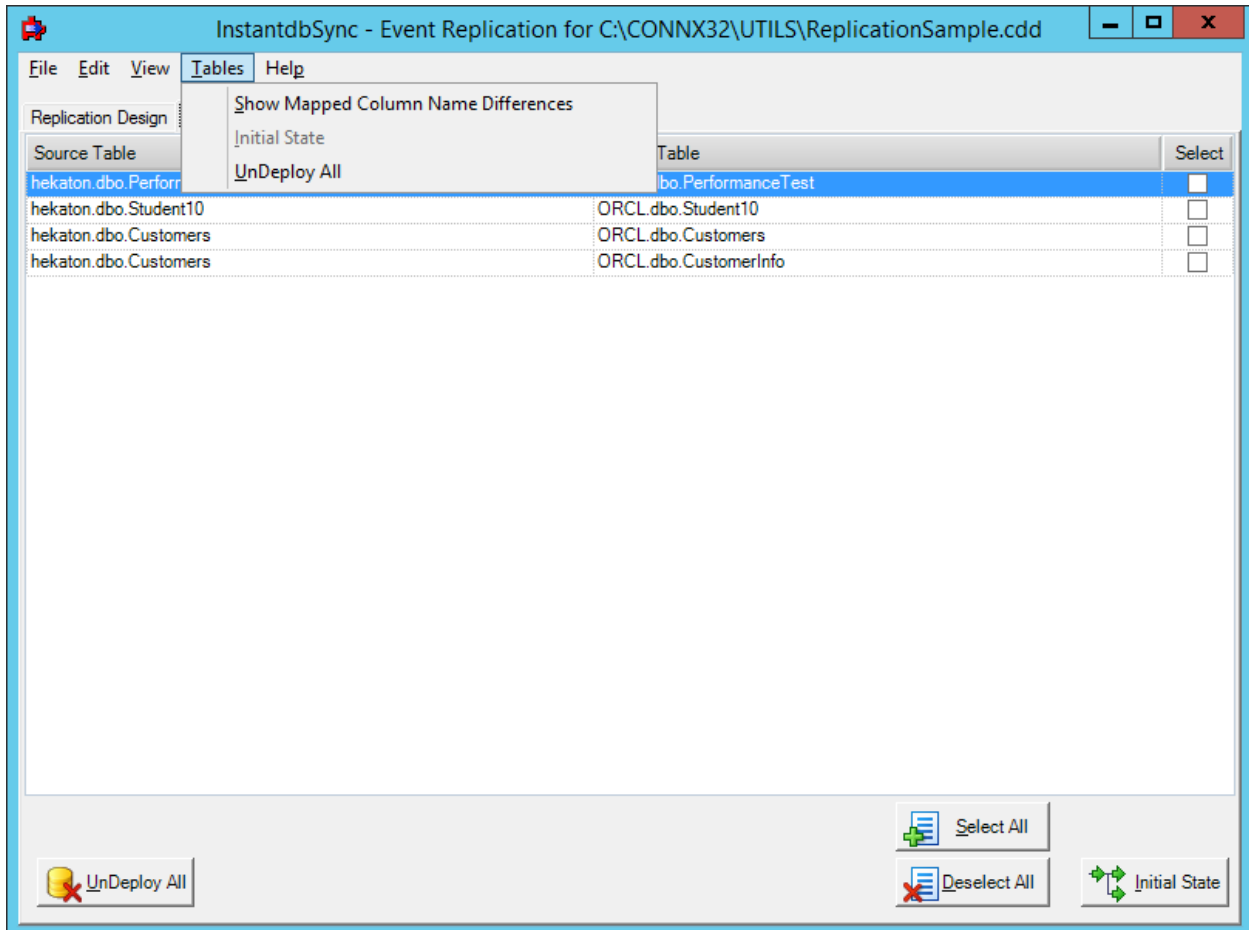
- Builds the target tables for all replications that are marked **Active** and have the **Create** check box checked.

Deploy

- Deploys all replications that are marked **Active**

Each menu item on the Tables menu has a corresponding button at the bottom of the screen

Deployed Replications tab:



The Tables Menu on the Deployed Replications tab has three commands:

- **Show Mapped Column Name Differences**

Shows a list of all columns in a replication where the source and target columns are not the same.

- **Initial State**

Selecting this option will send an Initial State request to the controller for all replications that have the Select box checked. Use this option if you need to refresh a target table on a deployed replication. The corresponding Initial State button at the bottom of the screen performs the same function.

- **UnDeploy All**

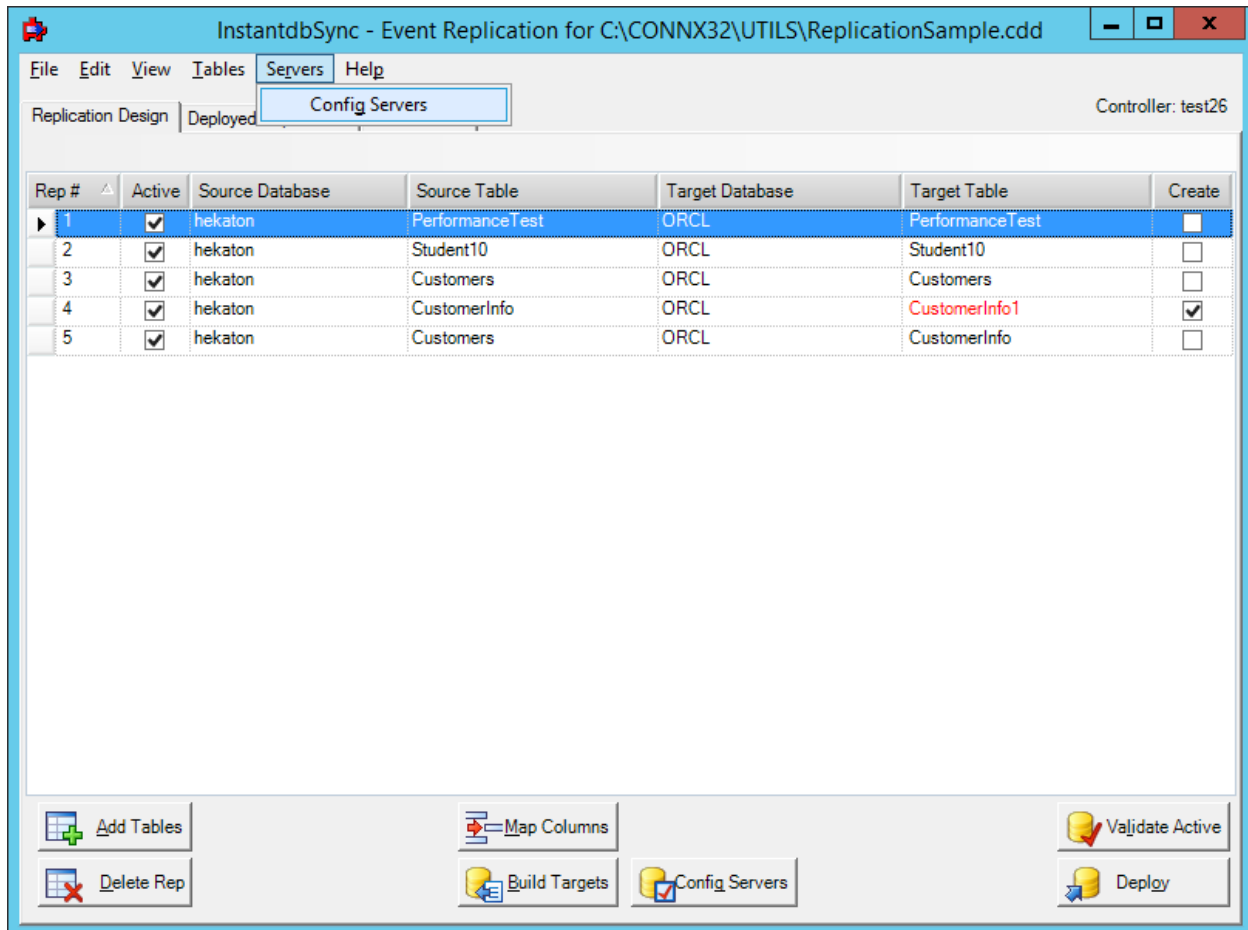
Selecting this option will undeploy all deployed replications. This stops the entire replication process. No transactions on the source will be queued or replicated. If you wish to undeploy individual replications, go

to the Replication Design tab, uncheck the active check box for those replications and then press the Deploy button. This will not cause previously deployed replications to go through the initial state process.

Servers Menu

Applies to Replication Design and Server Status tabs.

Replication Design tab:

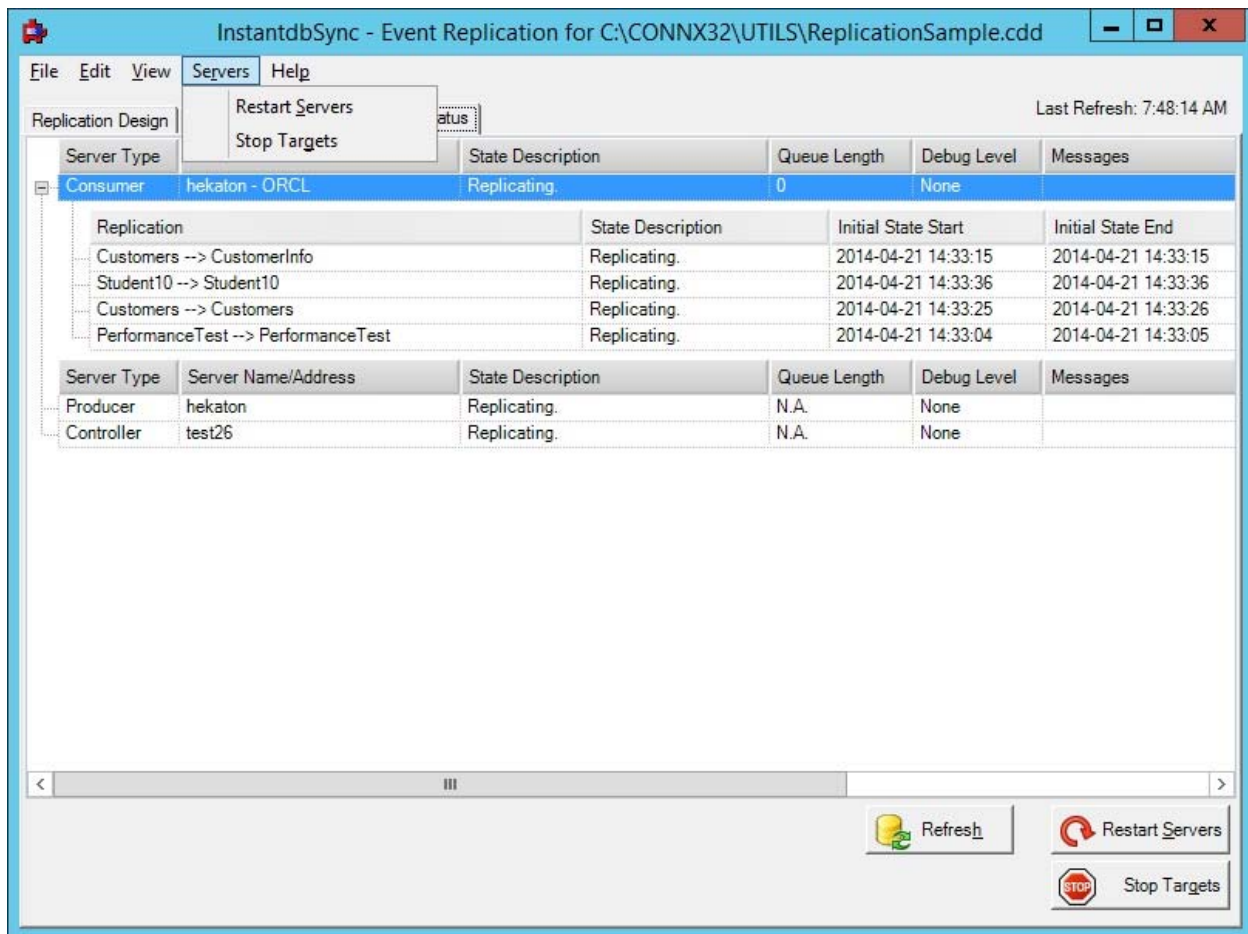


The Servers menu has one command:

- [Config Servers](#)

Invokes the Config Servers dialog which specifies server names and ports to be used during the replication process.

Server Status tab:



The Servers Menu has two commands:

- [Stop Targets](#)

The Stop Targets command will tell the replication server to stop sending transactions to the target database. After stopping the targets, the State Description for the Consumers and Replications will show Offline. Transactions that occur on the source database will be queued in the Message Queue until a Restart Servers command is sent to the controller or the controller itself is restarted. At this point transaction process will continue normally.

- [Restart Servers](#)

The Restart Servers command will tell the replication server to resume processing transactions to the target database.

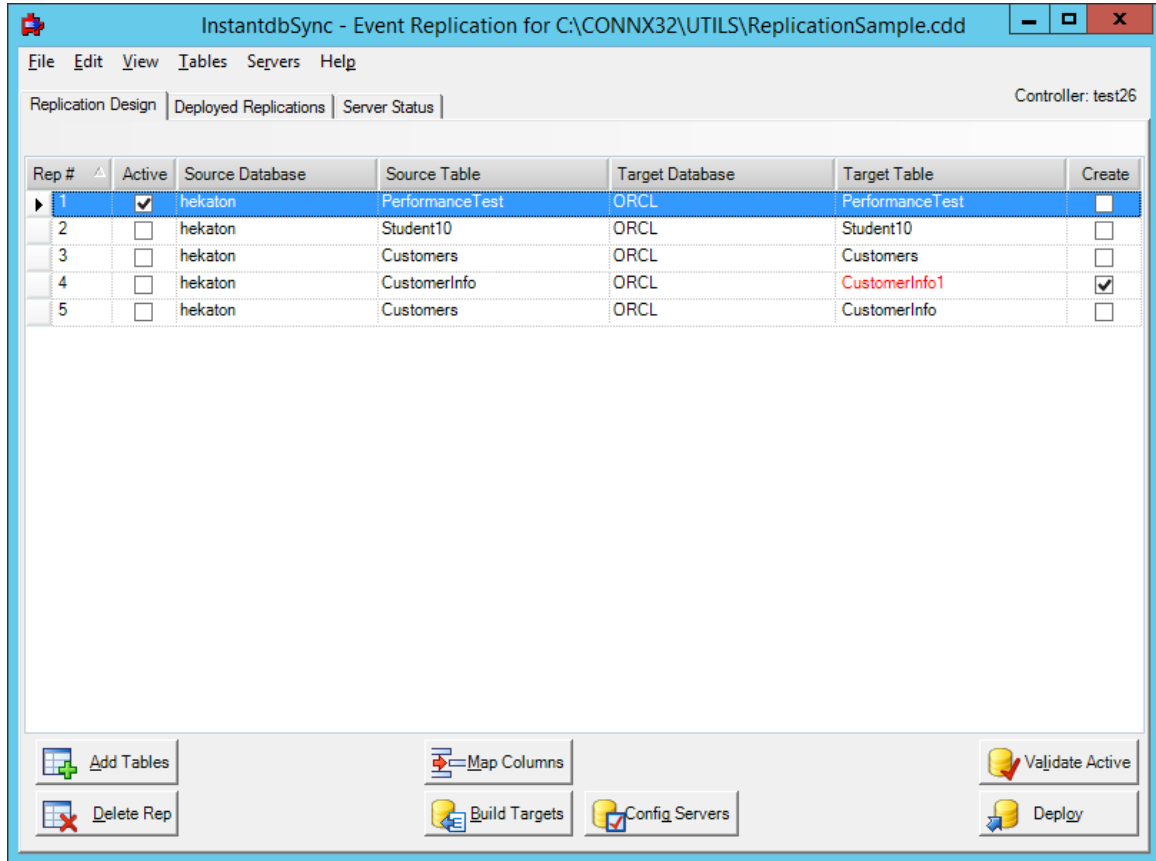
Help Menu

The Help Menu has two commands:

- User Guide
Opens this User Guide
- About Open Systems Event Replication
Opens the **About Open Systems Event Replicator** window

Replication Design Tab

The **Replication Design** tab is used for creating replications that will later be deployed to the replication controller. It displays the selected source tables within the source database, and the target tables in the target database that have been selected for replication.



- The **Rep #** is a unique identifier permanently assigned to that replication. **Note:** This number is not reused. If a replication is deleted, subsequent replications will not be assigned that number.
- The **Active** check box is used to determine which replications will be deployed. Click **Active** if you want this replication to be deployed. If you wish to undeploy a specific replication, uncheck the **Active** check box and press deploy. this will remove that replication from the Replication Controller. You cannot undeploy all replications from this tab. This task must be done from the [Deployed Replications Tab](#).
 - The Target Database must exist and must have been added to the Open Systems Event Replicator using **Add Tables**.
 - If the Target Table does not exist, the Open Systems Event Replicator will create the Target Table when **Active** and **Create** are turned and you click either **Build Targets** or **Deploy**.
- The **Create** check box is used if the target table does not exist on the target database. To create the target table, check both the **Active** and the **Create** check boxes and press the [Build Targets](#) button. If you wish to deploy at the same time, pressing the [Deploy](#) button will first build the target table and then deploy the replication.
- Clear **Active** if you do not want to deploy this replication.
- **Source Database**, **Source Table**, **Target Database** and **Target Table** contain the names of the databases and tables that have been made available for replication.

- When a new replication is deployed an initial state will be performed on target table. This process involves copying all the records from the source table to the target table. During this process any transactions that occur on the source table will be held in the message queue until the initial state process has completed. At that time normal transaction processing from the message queue to the target table will commence. If the target database is Microsoft SQL Server or Oracle a bulk load will be used during the initial state process. It is possible to [suppress the initial state](#).

You can adjust the display column width if part of the information is not visible.

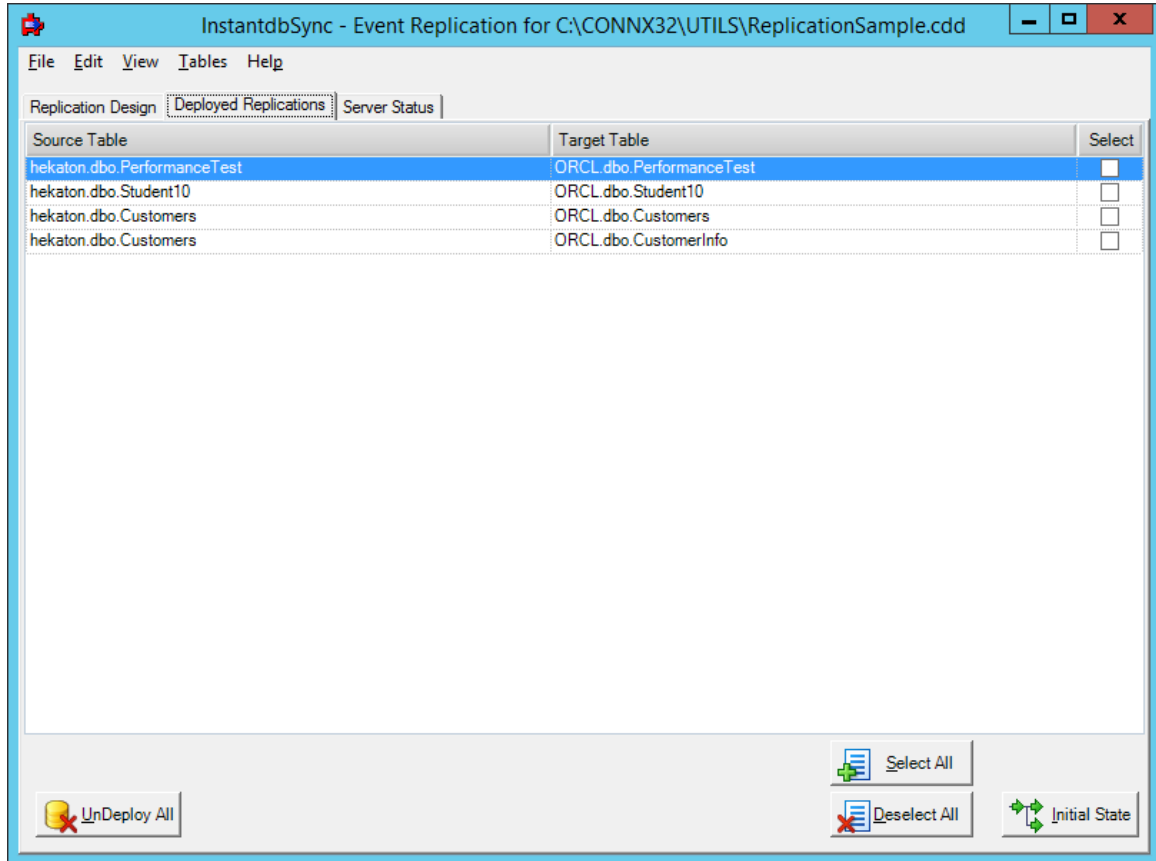
There are command buttons for the following:

- [Add Tables](#)
- [Delete Rep](#)
- [Map Columns](#)
- [Build Targets](#)
- [Config Servers](#)
- [Validate Active](#)
- [Deploy](#)

Deployed Replications Tab

The **Deployed Replications** tab displays read-only information. It lists the deployed source tables and their associated target tables.

Note: Replications must be deployed before this tab will be active. If this tab is selected when no replications have been deployed, the screen will be blank and all buttons will be disabled.



You can adjust the display column width if part of the information is not visible.

There are command buttons for the following:

- [UnDeploy All](#)

UnDeploys all currently deployed replications

- Select All

Selects all replicatons on this screen. Selections are used for Initial State requests

- Deselect All

Clears the **Select** check box for all replications

- [Initial State](#)

Process an Initial State on all selected replications

Server Status Tab

The **Server Status** tab shows a system-wide server view.

Server Type	Server Name/Address	State Description	Queue Length	Debug Level	Messages																														
Consumer	hekaton - ORCL	Replicating.	0	None																															
<table border="1"> <thead> <tr> <th>Replication</th><th>State Description</th><th>Initial State Start</th><th>Initial State End</th><th>Total (HH:MM:SS)</th><th>Messages</th></tr> </thead> <tbody> <tr> <td>Customers --> CustomerInfo</td><td>Replicating.</td><td>2014-04-21 14:33:15</td><td>2014-04-21 14:33:15</td><td>00:00:01</td><td></td></tr> <tr> <td>Student10 --> Student10</td><td>Replicating.</td><td>2014-04-21 14:33:36</td><td>2014-04-21 14:33:36</td><td>00:00:01</td><td></td></tr> <tr> <td>Customers --> Customers</td><td>Replicating.</td><td>2014-04-21 14:33:25</td><td>2014-04-21 14:33:26</td><td>00:00:01</td><td></td></tr> <tr> <td>PerformanceTest --> PerformanceTest</td><td>Replicating.</td><td>2014-04-21 14:33:04</td><td>2014-04-21 14:33:05</td><td>00:00:01</td><td></td></tr> </tbody> </table>						Replication	State Description	Initial State Start	Initial State End	Total (HH:MM:SS)	Messages	Customers --> CustomerInfo	Replicating.	2014-04-21 14:33:15	2014-04-21 14:33:15	00:00:01		Student10 --> Student10	Replicating.	2014-04-21 14:33:36	2014-04-21 14:33:36	00:00:01		Customers --> Customers	Replicating.	2014-04-21 14:33:25	2014-04-21 14:33:26	00:00:01		PerformanceTest --> PerformanceTest	Replicating.	2014-04-21 14:33:04	2014-04-21 14:33:05	00:00:01	
Replication	State Description	Initial State Start	Initial State End	Total (HH:MM:SS)	Messages																														
Customers --> CustomerInfo	Replicating.	2014-04-21 14:33:15	2014-04-21 14:33:15	00:00:01																															
Student10 --> Student10	Replicating.	2014-04-21 14:33:36	2014-04-21 14:33:36	00:00:01																															
Customers --> Customers	Replicating.	2014-04-21 14:33:25	2014-04-21 14:33:26	00:00:01																															
PerformanceTest --> PerformanceTest	Replicating.	2014-04-21 14:33:04	2014-04-21 14:33:05	00:00:01																															
Producer	hekaton	Replicating.	N.A.	None																															
Controller	test26	Replicating.	N.A.	None																															

This grid contains the status for all the Replication components. The columns and their meaning are as follows:

- **Server Type**
 - **Controller**

The Controller is responsible for accepting work messages from the Administrator and distributing those work orders to the Event Producer and Event Consumer. The Controller manages error messages as well as work messages and is responsible for starting and stopping the Event Consumer. It resides on the same system as the Event Consumer but does not need to be on the same system as the Event Producer or the target database.

- **Producer**

The Event Producer resides on the same system as Adabas and is loaded into the process space of the Adabas nucleus. When changes are made in Adabas files that have been identified for replication, the Event Producer stores those changes in a persistent store on the hard drive. When it receives an ET for a transaction, the Event Producer assembles all the elements of the transaction

into a message and puts it on the message queue.

- **Consumer**

The Event Consumer is responsible for taking transactions off the message queue and executing them on the target database. Once a transaction has been successfully committed, it will be removed from the message queue. There is one Event Consumer for each source DBID/target database pair. If two Adabas DBID's are being replicated to a single target database, there will be two Event Consumers; one for each source/target combination. In addition to transaction processing, the Event Consumer is also responsible for Initial State processing. The number of initial states or transactions it can process in parallel is determined by the Parallel transaction count setting on the Config Servers dialog. This dialog can be access from the Servers menu on the Replication Design Tab. Under each Consumer entry on the grid is a list of replications currently being processed for that Event Consumer. Each replication entry has the following columns:

1. **Replication:** This is the name of the source table followed by the name of the target table.
2. **State Description:** This tells the current state of the replication. For a complete list of State Descriptions, see below.
3. **Initial State Start:** This is the date and time the last initial state was started for this replication.
4. **Initial State End:** This is the date and time the last initial state ended.
5. **Total (HH:MM:SS):** The total time the last initial state took for this replication.
6. **Messages:** If there are any error messages associated with this replication, they will be displayed here. **Hint:** Double click on this field to display the entire message in an edit box for easier reading.

The Event Consumer can show performance statistics about the replications if the controller has the **CTRL_PERF_ENABLED** registry setting set to 1. The amount of time each statistics period is collected before it is recorded is 30 seconds by default or can be set with the **CTRL_PERF_QUERY_INTERVAL** registry setting. Once the controller is recording and sending back statistics, they can be hidden on the Replication Administrator for clarity by unchecking View --> Show Statistics. The statistics recorded are:

- a. **Consumer start date** - the day and time the consumer first started replicating after a deploy.
- b. **Status Interval length** - the length of time in seconds statistics are gathered before recording them. (set in the registry on the controller).
- c. **Statistics Totals**
 - i. **Consumer Since Start** - these are the total values collected since the engine was started with the first deploy.
 - ii. **Last Interval** - these are the total values collected during the last statistics recording interval (as set by the **CTRL_PERF_QUERY_INTERVAL** registry setting).
- d. **Statistics Rates**
 - i. **Consumer Total** - these are the rates for this consumer. Rates are determined amount per second.
 - ii. **Record Loader(i)** - these are the rates for each Record loader the consumer is running. The number of Record loaders is set in the Replication administrator Config Servers screen with the "Parallel transaction count".
- e. **Initial States** - the number of initial states processed by this consumer.

- f. **Transactions** - number of transactions the consumer has processed and committed to the target.
- g. **Rows Processed** - number of individual rows processed by the consumer and sent to the target.
- h. **Events** - number of messages from the source processed and sent to the target by the consumer.
- i. **Inserts** - number of rows inserted into the target by the consumer.
- j. **Updates** - number of rows updated in the target by the consumer.
- k. **Deletes** - number of rows delete in the target by the consumer.
- l. **Errors** - number of errors recorded during replication by the consumer.

- **Server Name/Address**

This is the name of the server that the corresponding component is installed on.

- **State Description**

This tells the current state of the component or replication. Potential values and their meaning are as follows:

- **Replicating:** Active and replicating.
- **Not Replicating - offline:** This indicates that either the target database as gone down or replication has been stopped by pressing the Stop Targets button. (see below) This is a Consumer status message.
- **Replication Offline:** This indicates that an individual replication is offline either due to an unrecoverable error on the target or by pressing the Stop Targets button. (see below) This is a Replication status message.
- **Processing Initial State...:** On the Consumer, this indicates that it is processing initial states. Each replication that is actively being processed will also indicate this status.
- **Initial State Pending:** This indicates that an initial state has been requested for a replication, but it has not started yet.
- **Initial State Complete:** This indicates that an initial state has finished for this replication but the overall process has not yet been completed by the Event Consumer. Once all pending initial states have completed, the consumer and all replications with a successful initial state will go to a Replicating status.
- **Initial State Failed:** This indicates that an error occurred during initial state processing. Check the Messages field for more information about the error.
- **Deploy Failed:** This indicates that an error occurred during the deploy process. Check the Messages field for all components for mor information about the error.
- **No Active Configuration:** This indicates that nothing has been deployed to the controller.
- **Status of the System Unknown/Unable to contact controller:** This indicates that the Replication Administrator was not able to contact the controller to get status. Common causes of this message are the controller is not running or the message queues on either the Administrator machine, the EP machine or the controller machine are not running.
- **Queue Length**

This indicates the number of elements in the queue. In normal operation, this number will grow and shrink depending on load. A rapid increase in this number that never goes down is an indication that the target database is offline and transactions are queuing but not being processed

on the target.

- **Debug Level**

This indicates the debug level of each component. There are 3 possible values:

1. **None:** Minimal messages are written to the log. Errors are always written.
2. **Verbose:** Diagnostic information is written when errors occur along with transaction information.
3. **Extreme:** Complete diagnostics including buffer and transaction dumps are written to the log. **Note:** This mode will cause a degradation in performance. Only use this mode when debugging an error situation.

- **Messages**

If there are any error messages associated with this component, they will be displayed here. **Hint:** Double click on this field to display the entire message in an edit box for easier reading.

The server status information will automatically refresh every 30 seconds. To manually refresh the server status, click the **Refresh** button.

The **Time** field in the upper right corner will contain the latest refresh time.

You can also [adjust the display column width](#).

Chapter 4 - Event Replicator Functions

Mapping SQL Expressions to Target Column

Instead of selecting a **Source Column** name to map to a non-Adabas **Target Column**, you can create a SQL function and place it in **Source Column** to map to a **Target Column**.

Example: Enter (FIRST_NAME+MIDDLE_I+LAST_NAME) in the blank Source Column drop-down value to create FULL_NAME in the Target Column.

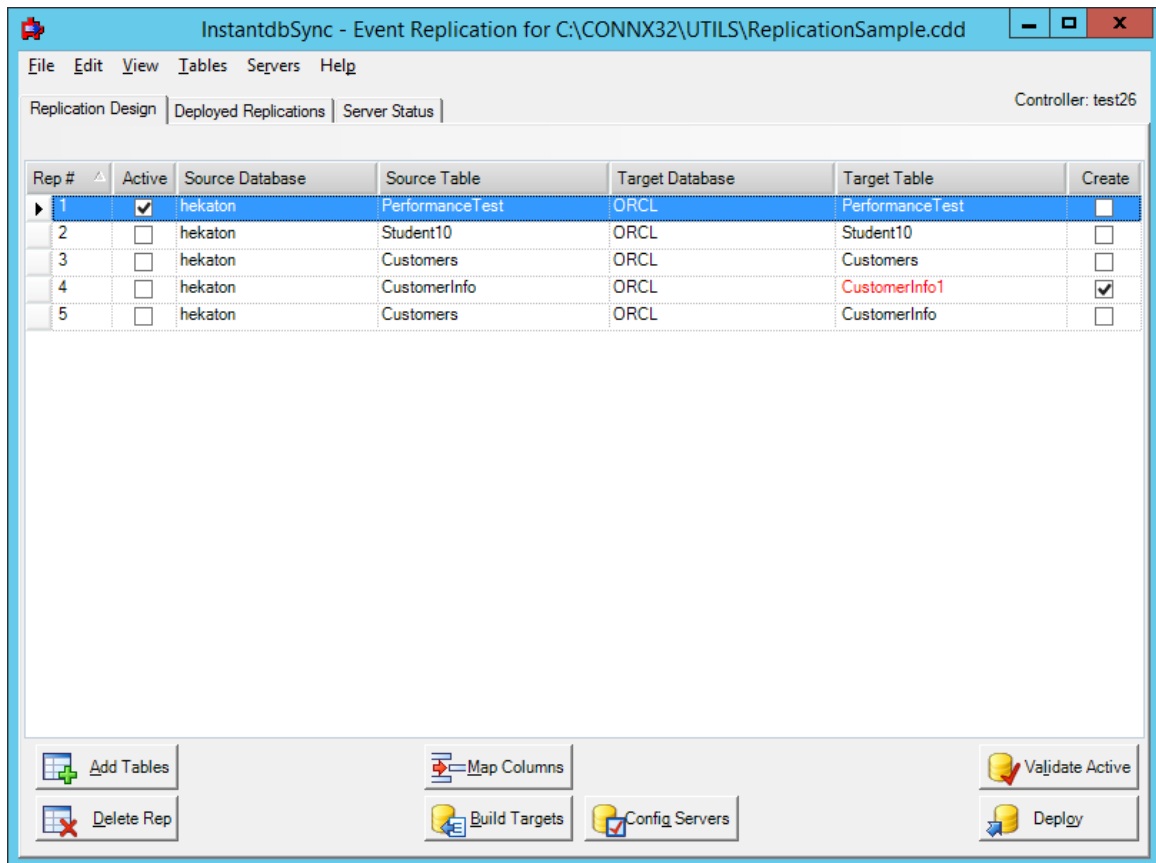
Note: Only non-aggregate SQL functions can be mapped to a **Target Column**.

Adjusting Display Column Width

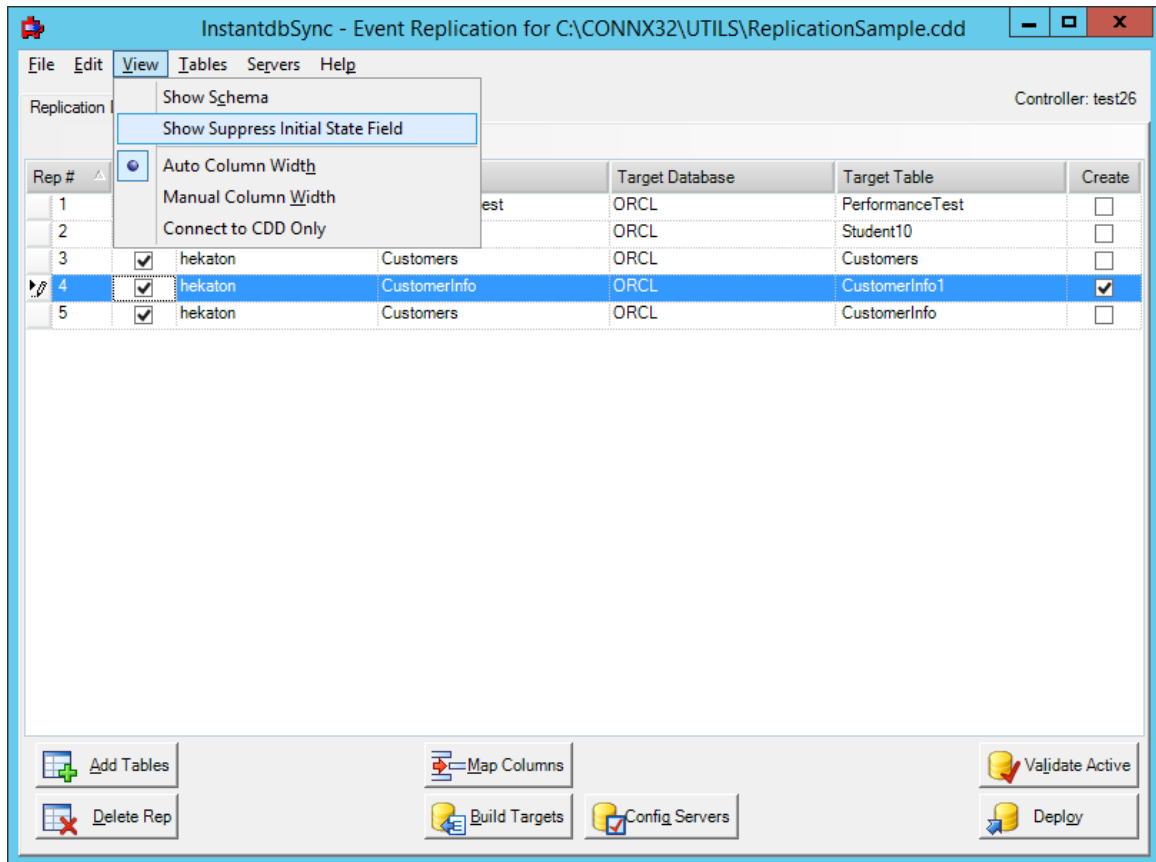
When replication information has been retrieved, by default, it fills the entire width of the display screen. You can adjust the width of any of the display columns.

The instructions show the Replication Design tab but you can also adjust the width of the Deployed Replications tab.

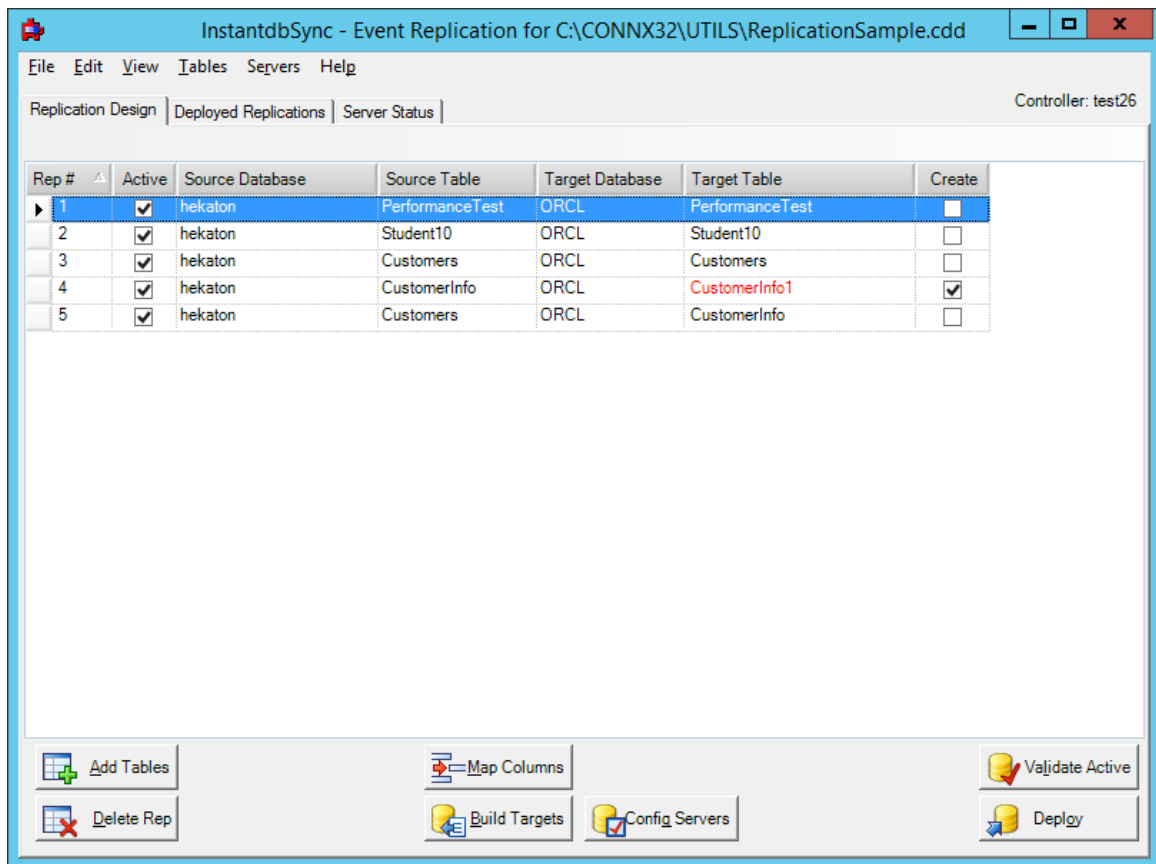
1. Open the Open Systems Event Replicator CDD.
2. Click the Replication Design tab. The following appears:



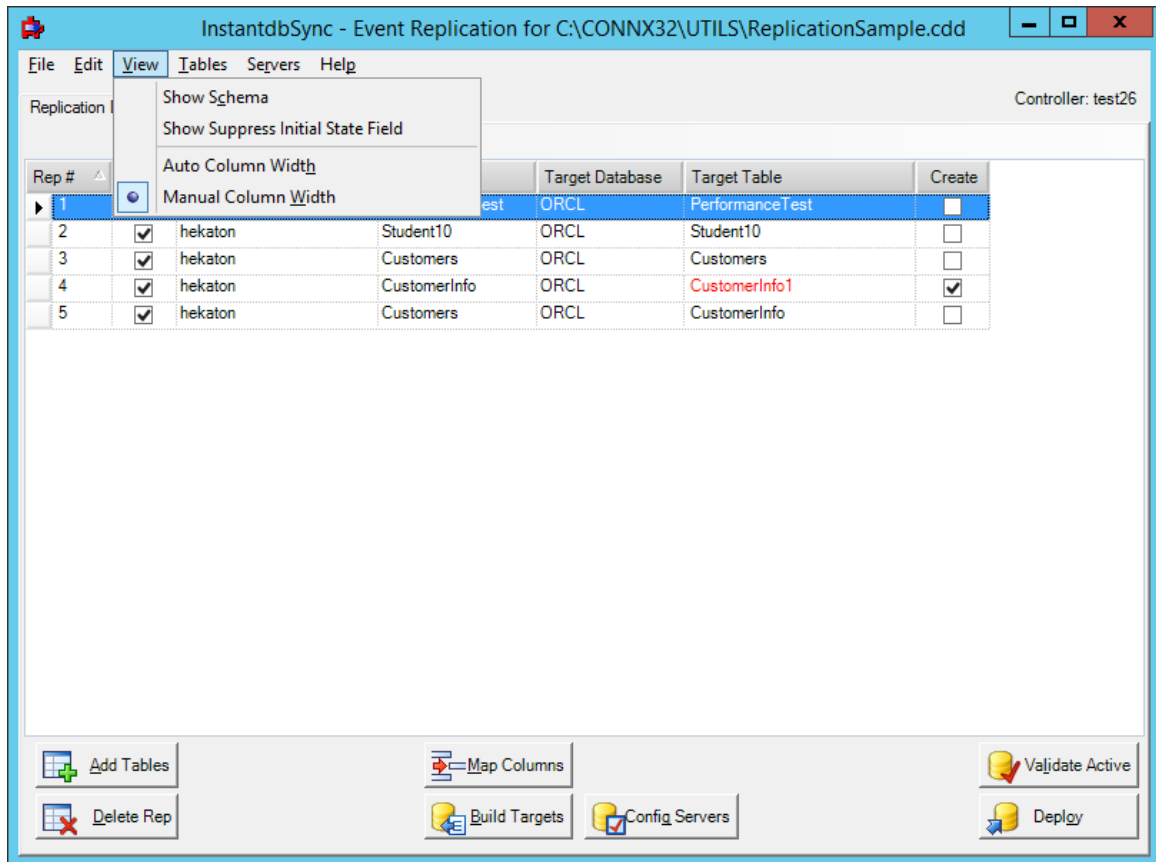
The **View** Menu displays the **Column Width** status. **Auto Column Width** is the default.



To adjust the column width, adjust the size of the cells by placing the mouse arrow on the horizontal line between column headings and performing a click + drag operation to the left or right..



When the cell width has been changed manually, the **Column Width** value changes to **Manual**.



Even if you change the size of the screen, the **Manual Column Width** size and setting stay until you select **Auto Column Width**.

Event Replication Registry Settings

You can modify several Open Systems Event Replication registry settings:

Registry Variables

- [CONNX.REPLICATION.CTRL_PORT](#)
- [CONNX.REPLICATION.DEBUG_LEVEL](#)
- [CONNX.REPLICATION.EP_ADA_CFG_PORT](#)
- [CONNX.REPLICATION.MQ_PORT](#)
- [CONNX.REPLICATION.NOTIFY](#)
- [CONNX.REPLICATION.NOTIFY_CMD](#)
- [CONNX.REPLICATION.STATUSREFRESHINTERVAL](#)
- [CONNX.REPLICATION.CTRL_PERF_ENABLED](#)
- [CONNX.REPLICATION.CTRL_PERF_QUERY_INTERVAL](#)
- [CONNX.REPLICATION.CTRL_PERF_STAT_FILE](#)
- [CONNX.REPLICATION.ADABCK_ET_SYNC_WAIT](#)
- [CONNX.REPLICATION.FIXED_TO_BINARY](#)
- [CONNX.REPLICATION.CONTINUEMODE](#)
- [CONNX.REPLICATION.DEBUG_STARTINGLSN](#)
- [CONNX.REPLICATION.NSPAUNLAG](#)
- [CONNX.REPLICATION.CTRL_OP_FLAGS](#)
- [CONNX.REPLICATION.DBG_OPTIONS](#)
- [CONNX.REPLICATION.DBG_EC_EP_PORT](#)
- [CONNX.REPLICATION.DBG_EC_EP_SERVER](#)
- [CONNX.REPLICATION.DBG_EC_IGNORE_ZERO](#)
- [CONNX.REPLICATION.DISABLE_ROW_CT_CHECK](#)
- [CONNX.REPLICATION.DNS_TO_DOT](#)
- [CONNX.REPLICATION.EC_EIS_WAIT](#)
- [CONNX.REPLICATION.EC_IS_TRUNCATE_ONLY](#)
- [CONNX.REPLICATION.EC_SLEEP_ON_STARTUP](#)
- [CONNX.REPLICATION.EC_SQL_ERR_MODE](#)
- [CONNX.REPLICATION.EC_SQL_MAX_HANDLES](#)
- [CONNX.REPLICATION.EP_EVENT_ERR_MODE](#)
- [CONNX.REPLICATION.MQ_CONFIG_DIR](#)
- [CONNX.REPLICATION.EP_QUEUE_HOST](#)
- [CONNX.REPLICATION.MQ_PORT](#)
- [CONNX.REPLICATION.MQ_QUEUE_DIR](#)

Unix/Linux Environment Variables

- [CONNX_EP_ADA_CFG_PORT](#)
- [CONNX_EP_DATA_PATH](#)
- [CONNX_EP_DEBUG_LEVEL](#)

- [CONNX_EP_NOTIFY](#)
- [CONNX_EP_NOTIFY_CMD](#)
- [CONNX_EP_QUEUE_PORT](#)

A2A specific Unix/Linux Environment Variables

- [ADABCKTIMEOUT](#)
- [ADABCKSLEEPTIME](#)

Registry Variables

CONNX.REPLICATION.CTRL_PORT

CONNX.REPLICATION.CTRL_PORT specifies the CONNX Message Queue controller port..

If CONNX.REPLICATION.CTRL_PORT is greater than 1024, the Replication Controller will accept Replication Engine communications on that port value.

If the port specified in CONNX.REPLICATION.CTRL_PORT is unavailable, then either change CONNX.REPLICATION.CTRL_PORT to 0 to allow the Replication Controller to use the first available port, or change CONNX.REPLICATION.CTRL_PORT to a port value that is available.

Valid CONNX.REPLICATION.CTRL_PORT values are:

- 0 - Use first available port
- 1 through 1024 - Reserved for internal use
- 1025 through 65535 - Use the port specified. The specified port must be available.

The default CONNX.REPLICATION.CTRL_PORT value is 9205.

CONNX.REPLICATION.DEBUG_LEVEL

CONNX.REPLICATION.DEBUG_LEVEL controls the debug output that goes to the log file.

The CONNX.REPLICATION.DEBUG_LEVEL values are:

- 0 - debug off.
- 1 - Normal debug output. Messages provide some debugging/status information but do not severely affect performance.
- 2 - Extreme debug output. The transaction level diagnostics generated will negatively impact performance.

The default CONNX.REPLICATION.DEBUG_LEVEL is 0 (off).

CONNX.REPLICATION.EP_ADA_CFG_PORT

CONNX.REPLICATION.EP_ADA_CFG_PORT specifies the port that the ep_config process will listen on for requests from the event producer.

The default CONNX.REPLICATION.EP_ADA_CFG_PORT value is 9207.

CONNX.REPLICATION.MQ_PORT

CONNX.REPLICATION.MQ_PORT specifies the CONNX Message Queue listening port..

The default CONNX.REPLICATION.MQ_PORT value is 9200.

CONNX.REPLICATION.NOTIFY

CONNX.REPLICATION.NOTIFY specifies whether or not to call the notification command job specified by CONNX.REPLICATION.NOTIFY_CMD if the system encounters an Nx error.

The CONNX.REPLICATION.NOTIFY values are:

- 0 - Do not call the notification command. CONNX.REPLICATION.NOTIFY_CMD is ignored.
- 1 - Call the notification command specified by CONNX.REPLICATION.NOTIFY_CMD.

The default CONNX.REPLICATION.NOTIFY value is 0 (do not call).

CONNX.REPLICATION.NOTIFY_CMD

CONNX.REPLICATION.NOTIFY_CMD specifies the location and name of the notification job to call when NOTIFY is 1 and the system encounters an Nx error.

If CONNX.REPLICATION.NOTIFY is 0, CONNX.REPLICATION.NOTIFY_CMD has no effect.

Example:

CONNX.REPLICATION.NOTIFY_CMD="C:\CONNX32\REPLICATION\COMMON\EVENTNOTIFY.CMD"

The default value is "eventnotify" and there is a sample eventnotify.cmd file in the Samples directory.

CONNX.REPLICATION.STATUSREFRESHINTERVAL

CONNX.REPLICATION.STATUSREFRESHINTERVAL controls the Server Status refresh interval. This setting applies to the Replication Administrator.

When you [view the Server Status](#), the status is normally refreshed every 30 seconds. CONNX.REPLICATION.STATUSREFRESHINTERVAL changes the default refresh interval.

Valid CONNX.REPLICATION.STATUSREFRESHINTERVAL values (in seconds) are:

- 0 - disable background refresh.
- Between 1 and 30 - 30 seconds (minimum value).

Greater than 30 - Number of seconds specified up to 86,400 seconds (maximum value).

Note: 86,400 seconds is 24 hours.

CONNX.REPLICATION.CTRL_PERF_ENABLED

CONNX.REPLICATION.CTRL_PERF_ENABLED enables the gathering and reporting of performance statistics by the Replication Controller.

Performance statistics will be written to the CTRL.log file. Optionally, the current statistics output will be directed to a file specified by CTRL_PERF_STAT_FILE.

Valid CONNX.REPLICATION.CTRL_PERF_ENABLED values are:

- 0 - disable performance statistics
- 1 - enable performance statistics. .

The default CONNX.REPLICATION.CTRL_PERF_ENABLED value is 0 (do not gather and report statistics).

CONNX.REPLICATION.CTRL_PERF_QUERY_INTERVAL

CONNX.REPLICATION.CTRL_PERF_QUERY_INTERVAL determines the number of seconds between statistics calls the controller makes to the EC.

Valid CONNX.REPLICATION.CTRL_PERF_QUERY_INTERVAL values are:

Number of seconds specified up to 86,400 seconds (maximum value).

The default CONNX.REPLICATION.CTRL_PERF_QUERY_INTERVAL value is 3600 seconds.

CONNX.REPLICATION.CTRL_PERF_STAT_FILE

CONNX.REPLICATION.CTRL_PERF_STAT_FILE causes the current statistics output to be saved to a designated file.

The output will still appear in the CTRL.log file. This file can be used as an easy way to access the most recent statistics without having to parse the CTRL.log file. Only the most current statistics will be in this file. For statistics history, the CTRL.log file must be used..

Valid CONNX.REPLICATION.CTRL_PERF_STAT_FILE values are:

The name (including the path) of the statistics file you wish to use.

The default CONNX.REPLICATION.CTRL_PERF_STAT_FILE value is blank which means do not use an additional statistics output file.

CONNX.REPLICATION.ADABCK_ET_SYNC_WAIT

CONNX InstantdbSync

CONNX.REPLICATION.ADABCK_ET_SYNC_WAIT causes the ADABCK DUMP step in the initial state process to use the ET_SYNC_WAIT parameter.

The Adabas utility ADABCK is used during the initial state process for Adabas to Adabas replication. When set, the ET_SYNC_WAIT parameter defines the time (in seconds) that ADABCK waits for ET-logic users to come to ET status at the end of the DUMP function. Setting the CONNX.REPLICATION parameter ADABCK_ET_SYNC_WAIT allows the user to control the value of this parameter for replication.

This parameter is set in the registry (sqlregistry on Unix/Linux machines and CONNX Configuration Manager on Windows machines) on the machine where the Replication Controller is running.

Valid CONNX.REPLICATION.ADABCK_ET_SYNC_WAIT values are:

0 - Do not set ET_SYNC_WAIT on the ADABCK DUMP command. ADABCK will use the value set in the nucleus' TT parameter.

1 through 32767 - The number of seconds to set ET_SYNC_WAIT on the ADABCK DUMP command.

The default CONNX.REPLICATION.ADABCK_ET_SYNC_WAIT is 60 seconds.

For more information on the ADABCK ET_SYNC_WAIT parameter, please see the Adabas Utilities guide.

Note: This setting does not apply when replicating to a relational target.

CONNX.REPLICATION.FIXED_TO_BINARY

CONNX.REPLICATION.FIXED_TO_BINARY causes the replication to use the binary format specifier in format buffers for fixed one and two byte fields.

This setting enables compatibility with format buffer handling in Natural for fixed one and two byte fields.

Note: This setting applies to Adabas replication only, when replicating to a relational target.

CONNX.REPLICATION.CONTINUEMODE

CONNX.REPLICATION.CONTINUEMODE will determine if replication continues when an error is encountered..

This should only be used for debugging purposes at the direction of a support engineer.

The default value of this setting is true. The error is logged and replication continues.

CONNX.REPLICATION.DEBUG_STARTINGLSN

CONNX.REPLICATION.DEBUG_STARTINGLSN will set lsn of the fist SQL server transaction log record for processing when replication starts.

This should only be used for debugging purposes at the direction of a support engineer.

CONNX.REPLICATION.NSPAWNLAG

CONNX.REPLICATION.NSPAWNLAG will delay the start of the relational event producers by the specified number of seconds

This should only be used for debugging purposes at the direction of a support engineer.

The default value is zero.

CONNX.REPLICATION.CTRL_OP_FLAGS

CONNX.REPLICATION.CTRL_OP_FLAG will enable enhanced debugging of the controller. Possible values are 1, and 2.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is zero.

CONNX.REPLICATION.DBG_OPTIONS

CONNX.REPLICATION.DBG_OPTIONS will enable enhanced debugging of the EP logger. Possible values are 0 and 32.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is zero.

CONNX.REPLICATION.DBG_EC_EP_PORT

CONNX.REPLICATION.DBG_EC_EP_PORT will enable isolated debugging of the event consumer by overriding the port.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is zero.

CONNX.REPLICATION.DBG_EC_EP_SERVER

CONNX.REPLICATION.DBG_EC_EP_SERVER will enable isolated debugging of the event consumer by overriding the server name of the controller.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is "localhost".

CONNX.REPLICATION.DBG_EC_IGNORE_ZERO

CONNX.REPLICATION.DBG_EC_IGNORE_ZERO will cause SQL delete and update operations that affect zero rows (when they should have affected 1 or more) to be treated as success.

No retry of the SQL operation will occur.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is false.

CONNX.REPLICATION.DISABLE_ROW_CT_CHECK

CONNX.REPLICATION.DISABLE_ROW_CT_CHECK will ignore errors conditions where an update or delete was performed, but zero rows were affected.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is false.

CONNX.REPLICATION.DNS_TO_DOT

CONNX.REPLICATION.DNS_TO_DOT will cause replication to convert host names to IP addresses before sending the information to other remote components.

This will potentially bypass DNS resolution problems where DNS names are not properly configured on all systems where components reside.

This should only be used at the direction of a support engineer.

The default value is false.

CONNX.REPLICATION.EC_EIS_WAIT

CONNX.REPLICATION.EC_EIS_WAIT will cause replication wait for the specified number of seconds after an initial state has completed..

This will potentially bypass DNS resolution problems where DNS names are not properly configured on all systems where components reside.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is 0.

CONNX.REPLICATION.EC_IS_TRUNCATE_ONLY

CONNX.REPLICATION.EC_IS_TRUNCATE_ONLY will cause replication only use truncate to clear target tables. Normally if truncate fails, then a delete will be issued.

When this setting is enabled, if the truncate fails, then the entire initial state operation fails.

The default value is false.

CONNX.REPLICATION.EC_SLEEP_ON_STARTUP

CONNX.REPLICATION.EC_SLEEP_ON_STARTUP will cause replication to sleep for the specified number of seconds at startup, providing enough time to attach a debugger..

This should only be used for debugging purposes at the direction of a support engineer.

The default value is 0.

CONNX.REPLICATION.EC_SQL_ERR_MODE

CONNX.REPLICATION.EC_SQL_ERR_MODE will determine what action is taken when an error occurs.

The following three values are available:

0 = Permissive. The error is logged and replication continues if the error is not fatal.

1 = Suspend. The error is logged, and replication is suspended for the specific table.

2 = Stop. The error is logged, and replication is stopped for all tables.

The default value is 0 (Permissive).

CONNX.REPLICATION.EC_SQL_MAX_HANDLES

CONNX.REPLICATION.EC_SQL_MAX_HANDLES will determine the maximum number of SQL statement handles to keep cached for improved performance.

The default value is 800.

CONNX.REPLICATION.EP_EVENT_ERR_MODE

CONNX.REPLICATION.EP_EVENT_ERR_MODE will determine the behavior of replication in the event of an error in the event producer itself.

The following three values are available:

0 = Permissive. The error is logged and replication continues if the error is not fatal.

1 = Suspend. The error is logged, and replication is suspended for the specific table.

2 = Stop. The error is logged, and replication is stopped for all tables.

The default value is 1 (Suspend).

CONNX.REPLICATION.MQ_CONFIG_DIR

CONNX.REPLICATION.MQ_CONFIG_DIR will overwrite the default location use for the configuration files of the message queue.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is an empty string.

CONNX.REPLICATION.EP_QUEUE_HOST

CONNX.REPLICATION.EP_QUEUE_HOST will overwrite the default name of the local TCP/IP connection for the message queue.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is "127.0.0.1".

CONNX.REPLICATION.MQ_PORT

CONNX.REPLICATION.MQ_PORT will overwrite the default port used to communicate to the local message queue.

The default value is 9200.

CONNX.REPLICATION.MQ_QUEUE_DIR

CONNX.REPLICATION.MQ_QUEUE_DIR will overwrite the default location use for the files of the message queue.

This should only be used for debugging purposes at the direction of a support engineer.

The default value is an empty string.

Unix/Linux Environment Variables

CONNX_EP_ADA_CFG_PORT

CONNX_EP_ADA_CFG_PORT specifies the port that the ep_config process will listen on for requests from the event producer.

The default CONNX_EP_ADA_CFG_PORT value is 9207.

CONNX_EP_DATA_PATH

CONNX_EP_DATA_PATH specifies the directory path used by the Event Producer to store configuration files. CONNX_EP_DATA_PATH must be able to read from and write to the Adabas nucleus.

CONNX InstantdbSync

There is no CONNX_EP_DATA_PATH default value. You must assign a path name to CONNX_EP_DATA_PATH.

CONNX_EP_DEBUG_LEVEL

CONNX_EP_DEBUG_LEVEL controls the debug output that goes to the log file.

The CONNX_EP_DEBUG_LEVEL values are:

- 0 - debug off.
- 1 - Normal debug output. Messages provide some debugging/status information but do not severely affect performance.
- 2 - Extreme debug output. The transaction level diagnostics generated will negatively impact performance.

The default CONNX_EP_DEBUG_LEVEL is 0 (off).

CONNX_EP_NOTIFY

CONNX_EP_NOTIFY specifies whether or not to call the notification command job specified by CONNX_EP_NOTIFY_CMD.

The CONNX_EP_NOTIFY values are:

- 0 - Do not call the notification command. CONNX_EP_NOTIFY_CMD is ignored.
- 1 - Call the notification command specified by CONNX_EP_NOTIFY_CMD.

The default CONNX_EP_NOTIFY value is 0 (do not call).

CONNX_EP_NOTIFY_CMD

CONNX_EP_NOTIFY_CMD specifies the location and name of the notification job to call when CONNX_EP_NOTIFY is 1 and an Nx error is encountered.

If CONNX_EP_NOTIFY is 0, CONNX_EP_NOTIFY_CMD has no effect.

Example: CONNX_EP_NOTIFY_CMD="../../../REPLICATOR/ETC/EVENTNOTIFY.BSH"

The default value is "eventnotify" and there is a sample eventnotify.bsh file in the Samples directory.

CONNX_EP_QUEUE_PORT

CONNX_EP_QUEUE_PORT specifies the port number of the CONNX Message queue on the local Adabas Nucleus machine.

The CONNX_EP_QUEUE_PORT may be the same as the CONNX.REPLICATION.MQ_PORT if both the Event Consumer and Event Producer reside on the same machine.

The default CONNX_EP_QUEUE_PORT is 9200.

A2A specific Unix/Linux Environment Variables

ADABCKTIMEOUT

ADABCKTIMEOUT specifies the amount of time the CONNX Server will wait for the ADABCK utility on the **target** database to start accepting data before it times out and returns an error. In some circumstances, ADABCK on the target may need to wait for some other process to complete before it can start accepting data from the source. An example of this would be if a an initial state is being done on a very large source file and the target database needs to be expanded in order to accommodate the file. In this case, ADABCK must wait for the nucleus to finish expanding the database before it can start accepting data from the source.

The value for ADABCKTIMEOUT is specified in minutes and the default is 10.

ADABCKTIMEOUT is a CONNX Server environment variable and must be set in the connxserver script on the target system.

This setting is only valid on Unix and Linux systems.

IMPORTANT NOTE: Changing this setting will not improve performance and setting it incorrectly can result in unexpected errors. Do not change the value of this setting unless directed to do so by CONNX Technical Support.

ADABCKSLEEPTIME

ADABCKSLEEPTIME specifies the amount of time the CONNX Server will wait between retrying writes when it receives an error while writing data to the target pipe the ADABCK utility is reading from.

While writing data to the target system, the CONNX Server may receive a response that the target is not yet ready to receive data. In this case, the server will wait for the time interval specified by ADABCKSLEEPTIME and then it will retry the write operation. It will continue to retry write operations until the timeout period specified by ADABCKTIMEOUT has been exceeded.

The value for ADABCKSLEEPTIME is specified in milliseconds and the default is 10.

ADABCKSLEEPTIME is a CONNX Server environment variable and must be set in the connxserver script on the target system.

This setting is only valid on Unix and Linux systems.

IMPORTANT NOTE: Changing this setting can have a severe impact on the performance of the initial state process. Do not change the value of this setting unless directed to do so by CONNX Technical Support.

Data Filtering

You can add replication filters by using the built-in features of CONNX. Using the Data Dictionary manager, apply a SQL based filter to any table in the **SQL View Clause** field.

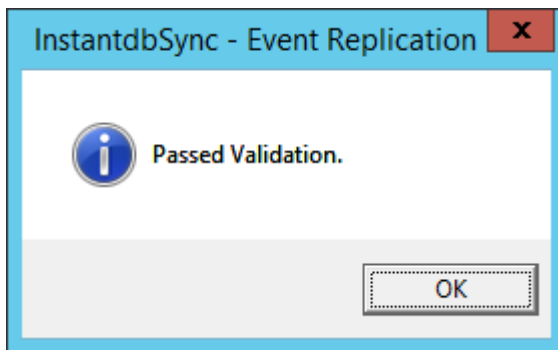
To filter replication by data content, use a SQL View Clause when creating the CDD entry. For more information, see the **SQL View Clause Text Box** topic in the **General Features** section of the **CONNX User Reference Guide**, Chapter 21-Advanced Features of CONNX.

If two replications want the same tables but different data, in the CONNX CDD Manager, clone the original table, add all the columns to the cloned table, and use a SQL View Clause to filter the data. For more information about cloning tables, see the **Clone Table Assistant** section in the **Advanced Features of CONNX** chapter in the **CONNX User Reference Guide**.

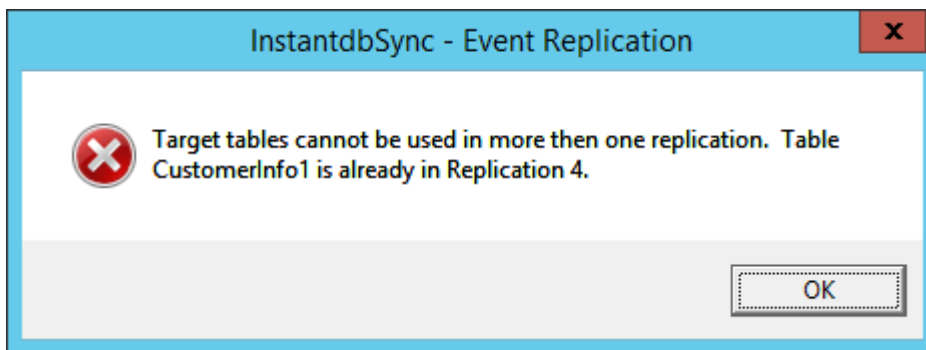
Validating Active Replications

1. Open the Open Systems Event Replicator CDD.
2. Click the **Active** check box for each replication you wish to validate and then click **Validate Active**.

The Replication Administrator checks all the Replications marked Active to make sure they are ready to be deployed. If the validation succeeds, the following message appears:



If the Validation fails, a message similar to the one below appears. It contains useful information to fix the problem.



Common errors include two replications with the same target table, or if a replication does not have current mapped columns.

Note: A replication must be marked Active for it to be validated; if a replication is not marked Active then the Replication Administrator ignores it when you click **Validate Active**.

Starting the UNIX Message Queue

Start the UNIX Message Queue if:

- The message queue does not automatically start after installing the Event Replicator on a UNIX environment
- The UNIX machine reboots and the message queue does not start automatically.

To start the UNIX message queue go to the CONNX installation directory and execute `./mqserver start`

Stopping the UNIX Message Queue

Stop the UNIX Message Queue if you will be reinstalling or updating the Event Replicator software

To stop the UNIX message queue, go to the CONNX installation directory and execute `./mqserver stop`

Starting the UNIX Replication Controller

Start the UNIX Replication Controller if:

- The replication controller does not automatically start after installing the Event Replicator on a UNIX environment
- The UNIX machine reboots and the replication controller does not start automatically.

To start the UNIX replication controller go to the CONNX installation directory and execute `./eventserver start`

Stopping the UNIX Replication Controller

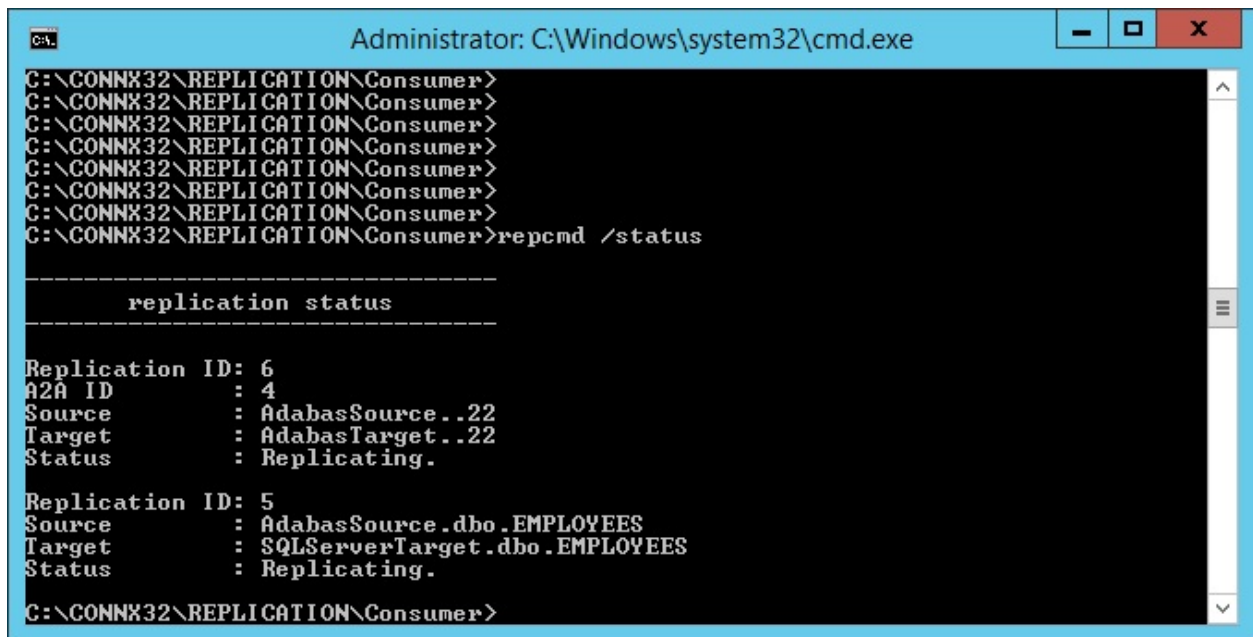
Stop the UNIX Replication Controller if you will be reinstalling or updating the Event Replicator software

To stop the UNIX replication controller, go to the CONNX installation directory and execute `./eventserver stop`

Using the repcmd command line utility

The repcmd command line utility is a program that can be used to pause and resume the replications, get replication status and start initial states on replications. The repcmd utility is in the replication\consumer folder and must be run from machine where the replication controller is installed. The repcmd program is tied to the local replication controller it is installed with, to use the commands with a different controller, the repcmd program local to the other controller must be used.

To get the status of the replications deployed to the local controller: repcmd /status This will list all the replications deployed and their current state.



```

Administrator: C:\Windows\system32\cmd.exe

C:\CONNX32\REPLICATION\Consumer>
C:\CONNX32\REPLICATION\Consumer>
C:\CONNX32\REPLICATION\Consumer>
C:\CONNX32\REPLICATION\Consumer>
C:\CONNX32\REPLICATION\Consumer>
C:\CONNX32\REPLICATION\Consumer>
C:\CONNX32\REPLICATION\Consumer>
C:\CONNX32\REPLICATION\Consumer>repcmd /status

-----
replication status
-----

Replication ID: 6
A2A ID       : 4
Source       : AdabasSource..22
Target      : AdabasTarget..22
Status      : Replicating.

Replication ID: 5
Source      : AdabasSource.dbo.EMPLOYEES
Target     : SQLServerTarget.dbo.EMPLOYEES
Status     : Replicating.

C:\CONNX32\REPLICATION\Consumer>

```

To pause replications use: repcmd /pause After executing, the success return means the message has been sent to the controller. After the command is sent, a status call can be used to see the state of the controller.

```

Administrator: C:\Windows\system32\cmd.exe

C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>repcmd /pause
serverState STAT_ACT_DISABLE_ENGINES success

C:\CONN32\REPLICATION\Consumer>repcmd /status

-----
replication status
-----

Replication ID: 6
A2A ID       : 4
Source       : AdabasSource..22
Target       : AdabasTarget..22
Status       : Replication Paused.

Replication ID: 5
Source       : AdabasSource.dbo.EMPLOYEES
Target       : SQLServerTarget.dbo.EMPLOYEES
Status       : Replication Offline.

C:\CONN32\REPLICATION\Consumer>

```

To resume replications use: `repcmd /resume` After executing, the success return means the message has been sent to the controller. After the command is sent, a status call can be used to see the state of the controller.

```

Administrator: C:\Windows\system32\cmd.exe

C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>repcmd /resume
serverState STAT_ACT_RESTART_ALL success

C:\CONN32\REPLICATION\Consumer>repcmd /status

-----
replication status
-----

Replication ID: 6
A2A ID       : 4
Source       : AdabasSource..22
Target       : AdabasTarget..22
Status       : Replicating.

Replication ID: 5
Source       : AdabasSource.dbo.EMPLOYEES
Target       : SQLServerTarget.dbo.EMPLOYEES
Status       : Replicating.

C:\CONN32\REPLICATION\Consumer>_

```

To start an initial state on one or many replications use: `repcmd /initialstate id=replicationid,replicationid` example to start an initial state on one replication use: `repcmd /initialstate id=4` To start an initial state on three replications use: `repcmd /initialstate id=4,5,6` After the initial state request has been sent, the status can be returned to see the progress of the initial states with `repcmd /status`

```

Administrator: C:\Windows\system32\cmd.exe
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>
C:\CONN32\REPLICATION\Consumer>repcmd /initialstate id=5,6
type:1, id:5
type:1, id:6
initial State request sent.
C:\CONN32\REPLICATION\Consumer>repcmd /status

-----
replication status
-----

Replication ID: 6
A2A ID       : 4
Source       : AdabasSource..22
Target       : AdabasTarget..22
Status       : Initial State Pending.

Replication ID: 5
Source       : AdabasSource.dbo.EMPLOYEES
Target       : SQLServerTarget.dbo.EMPLOYEES
Status       : Initial State Complete.

C:\CONN32\REPLICATION\Consumer>_

```

* Note: When using the windows powershell to run initial state requests from the repcmd it may be necessary to use quotes around the command statement. For example from the powershell run: `cmd.exe /c "repcmd.exe /initialstate id=4,5,6"` or `cmd.exe /c 'repcmd.exe /initialstate id=4,5,6'`

* Note: The command to start an initial state uses the Replication ID found in the CONNX Replication Administrator. If an Adabas to Adabas replication is added through the Adabas commands and not with the Replication Administrator, but it will not have a Replication ID to perform an initial state. Status can still be returned on the new Adabas to Adabas replication, but not starting initial states. To get a replication ID for the new Adabas to Adabas replication, open the Replication Administrator, go to the Server Status tab to sync up the Replications. Then perform a deploy to send the new Replication ID to the controller. Then a repcmd /status command will show the Replication ID for the Adabas to Adabas replication and the initial state command can be used.

Chapter 5 - ACD Replications

Introduction to ACD Replications

ACD (Add, Change, Delete) Replications are a special type of replication that track the operations done against a database rather than replicating the data itself. When an ACD Replication is defined, a source table is defined and a special ACD target table is created in the target database. The Event Producer will now monitor the source database for any transactions against the table specified in the ACD Replication and a record will be inserted into the ACD target table specifying each operation (Add, Change or Delete) that occurs for the transaction. The ACD target table is a transaction log for the specified source table.

ACD Target Table Metadata

The ACD target table consists of all the columns from the source table preceded by five pre-defined columns that provide transaction log information. The five columns are:

Column Name	Data Type	Description
CNX_ACD_TXID	VarChar(64)	Transaction ID from source database
CNX_ACD_SequenceNumber	Integer	Sequence number within transaction
CNX_ACD_TimeCreated	TimeStamp	Timestamp from source database
CNX_ACD_ChangeUser	WVarChar(256)	User ID that made the change in the source database
CNX_ACD_ChangeType	Char(1)	Type of change. Possible values are A,B,C or D

CNX_ACD_TXID

- This is the transaction ID from the source database. All operations within a transaction will have the same CNX_ACD_TXID

CNX_ACD_SequenceNumber

- Each operation within a transaction will have a sequence number denoting its position within a transaction. For example, if a transaction contains an insert, an update and a delete in that order, the sequence number of the insert will be 1, update will be 2 and the delete will be 3.

CNX_ACD_TimeCreated

- This is the timestamp from the source database denoting the time the operation occurred.

CNX_ACD_ChangeUser

- User ID that committed the transaction

CNX_ACD_ChangeType

- This field indicates the type of operation that occurred (Add, Change or Delete). There are 4 possible values:
 1. **A**
The value A indicates Add. When a record is inserted into the source table, a record with a change type of A will be inserted into the target ACD table.
 2. **B**
The value B indicates the before image key field of an update. If an update on the source table changes the value of the primary key, the key value in this record will reflect the value prior to the update. If the key value did not change on the update, a B record will still be created but the value in the key column(s) will be the same as the value(s) in the C record.
 3. **C**
The value C indicates the post image record of an update. When an update occurs on the source table, a C record will be inserted into the target ACD table that contains all the data from the source after the update was completed.
 4. **D**
The value D indicates Delete. When a record is deleted from the source table, a D record is added to the target ACD table that contains the key of the deleted record.

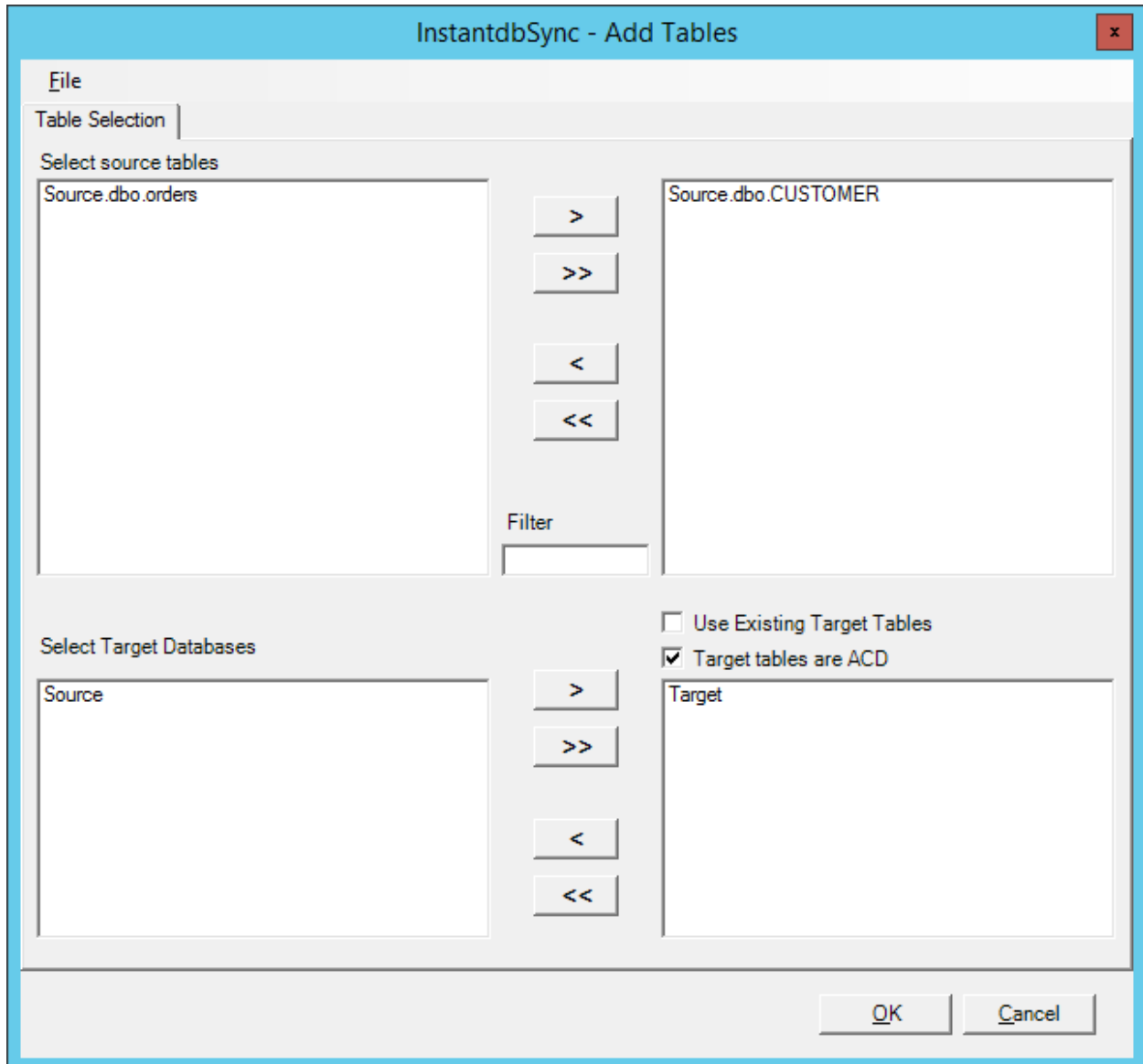
The ACD target table will be created with a non-unique index on the same column(s) as the primary key or unique index from the source table. Correct processing of the ACD target table relies on the metadata being known to InstantdbSync. While indexes may be added to this table to aide in faster retrieval of data, columns should never be added or removed nor should the index that was created with the table be removed.

Creating ACD Replications

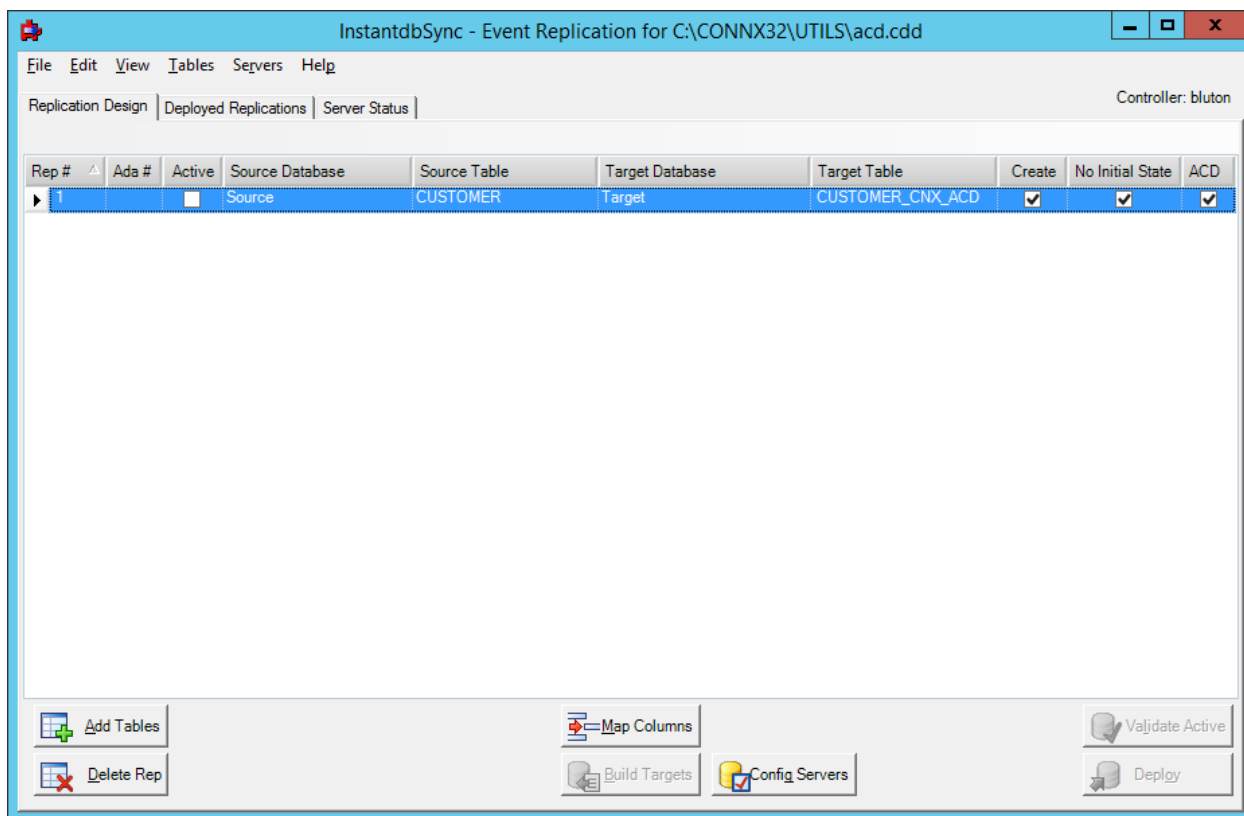
ACD replications require a CDD the same as data replications. Please see the topic titled "[Creating a CDD for Event Replication](#)" in Chapter 2 for information about creating a CDD.

Note: If you already have a replication CDD, you do not need to create a separate CDD for ACD replications. A single CDD can support all types of replications at the same time.

To create an ACD replication, open the InstantdbSync Replication Administrator and select Add. The Add Tables dialog will appear. Select the source table and target database. Check the "Target tables are ACD" checkbox. This will instruct the InstantdbSync Admin to create ACD replications for the tables that have been selected. If you wish to also create data (non-ACD) replications, they need to be created separately from the ACD replications.



Press OK to return to the Replication Design screen.



When ACD replications are present, the ACD and No Initial State columns will be displayed. The ACD column is checked if the replication is an ACD replication and un-checked if the replication is a data replication. This column is read-only and cannot be changed by the user. The No Initial State column is always checked for ACD replications and cannot be changed by the user. For data replications, the No Initial State column defaults to un-checked and can be changed by the user if desired. Please see the topic ["Suppressing an Initial State"](#) in Chapter 2 for more information on suppressing the initial state process for non-ACD replications. The ACD target table is a log of changes made from the point that the replication is deployed and as such, an initial state has no meaning and is therefore not allowed.

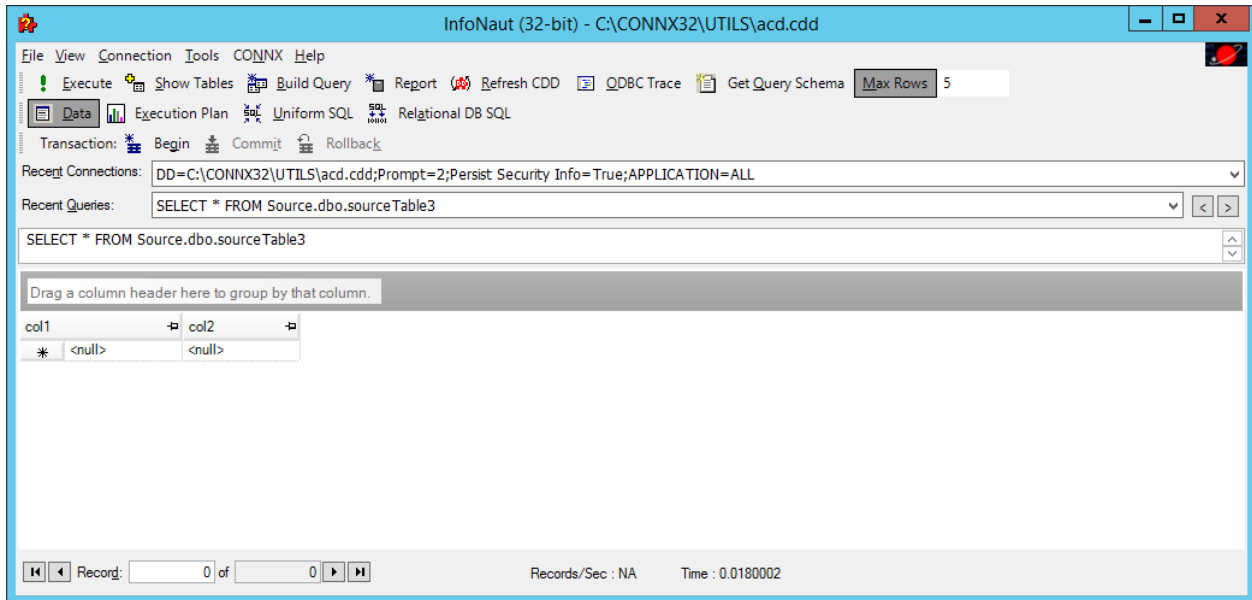
By default, the ACD target table will have the same name as the source table with "_CNX_ACD" appended to the name. The columns in the ACD target table will be the same as the source table with the ACD specific columns at the beginning. See the topic ["Introduction to ACD Replications"](#) in this chapter for a list of the ACD specific columns. To see the metadata for the ACD target, press the Map Columns button. Note that the metadata displayed is read-only and cannot be modified.

Once the ACD replication has been created, check the Active check box and press the Deploy button. Please see the topic ["Deploying Replications"](#) in Chapter 2 for information on the Deploy function.

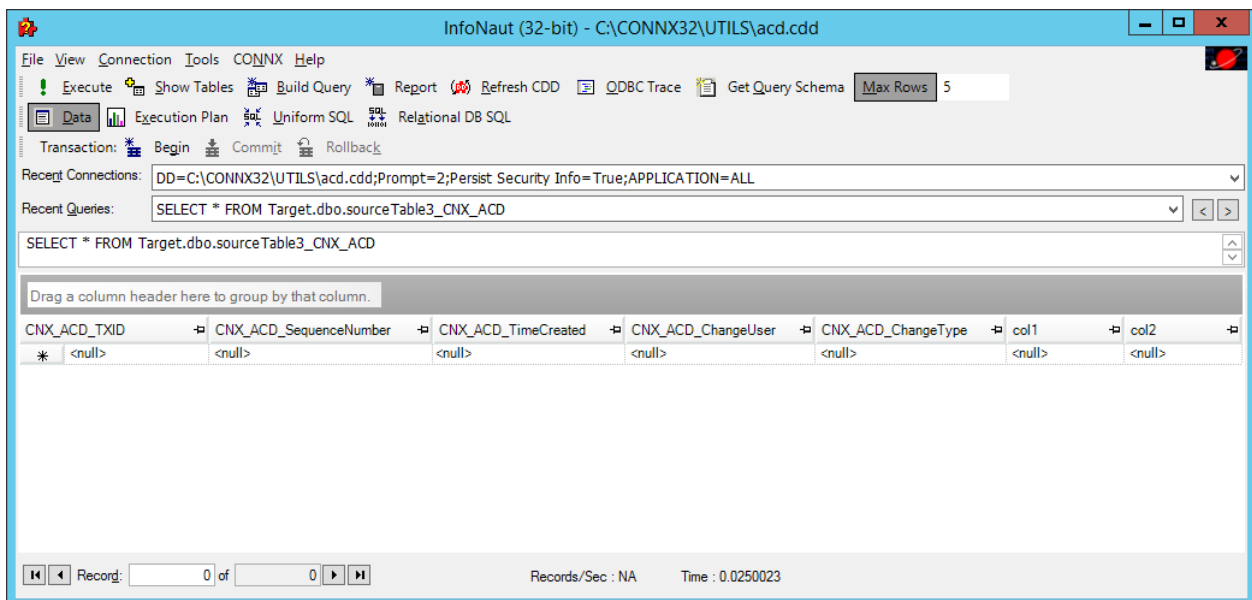
ACD Output

The ACD target table contains 5 columns (CNX_ACD_TXID, CNX_ACD_SequenceNumber, CNX_ACD_TimeCreated, CNX_ACD_ChangeUser and CNX_ACD_ChangeType) followed by the columns from the source table. The ACD target table is a transaction log specific to the source table specified in the ACD replication.

The following examples will use a source table with two character columns, col1 and col2. Col1 is the Primary Key:



The target ACD table has the CNX_ACD columns followed by col1 and col2:



If the following SQL is executed against the source table:

Begin Transaction;

INSERT INTO sourceTable3 (col1, col2) VALUES('abc', 'def');

INSERT INTO sourceTable3 (col1, col2) VALUES('ghi', 'jkl');

Commit;

The ACD target table will contain:

CNX_ACD_TXID	CNX_ACD_SequenceNumber	CNX_ACD_TimeCreated	CNX_ACD_ChangeUser	CNX_ACD_ChangeType	col1	col2
1	0000:00000592	11/23/2016 12:24:25.0000000 PM	A	A	ab	def
2	0000:00000592	11/23/2016 12:24:25.0000000 PM	A	A	gh	jkl
*	<null>	<null>	<null>	<null>	<null>	<null>

Each row represents a log entry for each operation. Since both inserts were done within the same transaction, they have the same TXID. The Sequence number represents the order within the transaction the operations were executed; i.e. the row with the value 'ab' in col1 was inserted first and has a SequenceNumber of 1.

The TimeCreated column is the timestamp that represents the time the SQL statement was executed.

The ChangeUser column represents the user who executed the query.

The ChangeType column indicates the type of change. A type value of A represents Insert.

Issuing the update statement:

UPDATE sourceTable3 set col2 = 'xxx' where col1 = 'ab';

will result in the following entries being added to the ACD target:

CNX_ACD_TXID	CNX_ACD_SequenceNumber	CNX_ACD_TimeCreated	CNX_ACD_ChangeUser	CNX_ACD_ChangeType	col1	col2
1	0000:00000592	11/23/2016 12:24:25.0000000 PM	A	A	ab	def
2	0000:00000592	11/23/2016 12:24:25.0000000 PM	A	A	gh	jkl
3	0000:00000593	11/23/2016 12:36:13.0000000 PM	B	C	ab	<null>
4	0000:00000593	11/23/2016 12:36:13.0000000 PM	C	C	ab	xxx
*	<null>	<null>	<null>	<null>	<null>	<null>

Lines 3 and 4 represent the update statement. Notice that an update results in two rows being added to the ACD table; a B record and a C record. The B record contains data for the Primary Key (or Unique Index) before the update occurred and <null> for the rest of the fields in the record. The C record contains the entire post update record. In this example, col2 was updated so the key information is the same for both the B and C record.

If the key field is updated, the output depends on the way the source database manages key field updates. For example, SQL Server 2012 issues a delete followed by an insert in its transaction log when a key field is updated. This will result in a D record followed by an A record being added to the ACD table. SQL Server 2014 issues an update in this situation. This would result in a B field that contains the original value for the key followed by a C record that contains the entire post update record.

Deleting a record results in a D record being generated. A D record contains the key value for the record that was deleted and <null> in the rest of the columns.

DELETE from sourceTable3 where col1 = 'ab';

Will result in:

	CNX_ACD_TXID	CNX_ACD_SequenceNumber	CNX_ACD_TimeCreated	CNX_ACD_ChangeUser	CNX_ACD_ChangeType	col1	col2
1	0000.00000592	1	11/23/2016 12:24:25.000000...	A		ab	def
2	0000.00000592	2	11/23/2016 12:24:25.000000...	A		gh	jkl
3	0000.00000593	1	11/23/2016 12:36:13.000000...	B		ab	<null>
4	0000.00000593	1	11/23/2016 12:36:13.000000...	C		ab	xxx
5	0000.00000594	1	11/23/2016 12:45:39.000000...	D		ab	<null>
*	<null>	<null>	<null>	<null>		<null>	<null>

Records in the ACD table are only inserted. InstantdbSync will not delete them. Once a set of records has been processed by the application consuming this information, it should delete the processed records if desired.

Appendix A - Troubleshooting Guide

Event Replication Troubleshooting Suggestions

Server Status

- I deployed the CDD, how do I know if the replication system is active?
Open the deployed CDD (if it is not open) and [view the Server Status](#). After a few seconds the grid will show the replication servers and their status.
- I cannot open the Server Status screen.
The current open CDD is not the one that has been deployed. Open the deployed CDD and [view the Server Status](#) to see the deployed server.
- On the Server Status screen it says **Status of the System Unknown**.
If the status is **Unknown**, the replication server is either down or cannot be contacted. Check the services to make sure the CONNX Replication Controller is **Started**. If the Controller is on a different machine then the replication administrator, check to see if it is accessible over the network.
- One of the entries in the Server Status screen is **Not Replicating**.
Check the log files at {install directory}\CONN32\Replication\log\, for an error message that explains the reason the server is not replicating. See the Log Files troubleshooting suggestions below.
- I receive the message "**Queue service is unavailable, check log at: <path to log>ADM.log**" when I attempt to deploy
This message indicates that the message queue service is not running either on the Windows machine where the Replication Administrator is running or on the server where the Replication Controller is running. The entry in the ADM.log file will specify the name of the system it was trying to contact. If the message queue services are running, please ensure that there are no firewalls between the Replication Administrator machine and the Replication Controller machine. If a firewall is required, port 9200 will need to be opened. Please note that Microsoft Windows has a firewall built into it and it is on by default.

If the message queue services are running and there is either no firewall present or there is but port 9200 is open, please contact Technical Support.

- I received the message "**Cannot contact controller, check log at: <path to log>ADM.log**" when I attempt to deploy
When the deploy button is pressed, the Replication Administrator sends a request for current status to the controller. The purpose of this message is to verify that all the components are running and will be able to respond to the deploy message. If you receive the "**Cannot contact controller**" message, it indicates that the controller did not respond to the status request. The three most likely causes of this message are:
 1. The controller (Event Server) is not running
 2. The message queue on the Replication Administrator machine is not running
 3. A firewall is preventing the controller from contacting the message queue on the Replication Administrator machine

When the Replication Administrator sends a message to the controller, it places that message on the controller's message queue. When the controller responds, it places the response message on the Replication Administrator's message queue. If the controller is not able to contact this message queue, it will not be able to send the response and the Replication Administrator will time out and issue the message stating it was not able to contact the controller. If you receive this message and have verified the controller is running, check the CTRL.log file. There will be a message preceded with (E) that will describe the problem that caused the controller to not respond. The most likely cause is that the controller was not able to contact the message queue on the Replication Administrator machine. If the message queue service is running on the Replication Administrator machine, check to see if the Windows firewall is running. If it is,

port 9200 will need to be opened to allow communications. You can also authorize the program cnxmq.exe in the firewall exceptions.

If the controller and message queue services are running and there are no firewall issues, please contact Technical Support.

Log Files

- For more information about the codes and messages displayed in the log files see the [Event Replication Error Messages](#).
- If you are not seeing enough information in the log files you may need to increase the number of messages by changing DEBUG_LEVEL. To change DEBUG_LEVEL go to [Event Replication Registry Settings](#).
- If a problem arises that cannot be corrected by looking at the errors in the log files, you may have to contact Technical Support. Have all the log files ready to be sent. Technical Support needs the information in all the log files to determine what is causing the problem.

Event Replication Error Messages

The Open Systems Event Replicator Error Message format is:

Error Code (Error Class): Explanation

There are six types of error messages:

Error Code	Message Type
1001 through 1999	General Messages
2001 through 2999	Admin
3001 through 4999	Controller
5001 through 5999	Event Consumer
6001 through 8999	Event Producer
9001 through 9999	Debug

There are two error classes:

Error Class	Description
W	Warning/ Informational
E	Error

There are four kinds of error actions:

Error Action Code	Error Action	Values
X	No action performed	
L	Message logged	
Nx	Message will notify	Values N1 (infrequent notification) through N4 (always notify). Notification must be enabled.
L/Nx	Message will log and notify	Values L/N1 (infrequent notification and always log) through L/N4 (always notify and log). Message always logs. Notification must be enabled to occur.

There are five error states:

Error State Code	Error State	Description
U	Uninitialized	No CDD deployed

N	Normal	System is running normally
C	Configure	System is processing deployment information
D	Disable Process	System is disabled. Usually due to an error or Stop Server message. - Note, if this code is issued due to an error rather than a shutdown command, the default setting is for replication to continue. The error should be investigated however, because there is a high probability that the transaction was not replicated.
S	Shutdown	System has received a Shutdown message

The Message ID below is composed of the Error Code, Error Class, Error Action and Error State.

General Messages

WARNINGS

Message ID				Explanation	Resolution
1001	W	L	N	Application initialized.	Informational.
1002	W	L	S	Application shutting down.	Informational.
1300	W	L	N	Application state has changed to: (ACTIVE).	Server has entered an active state.
1301	W	L	C	Application state has changed to: (CONFIGURE).	Server is processing a CDD for deployment.
1302	W	L	D	Application state has changed to: (DISABLED).	Server has entered an inactive state. May indicate an error has occurred. Check log for more information.
1303	W	L	S	Application state has changed to: (SHUTDOWN).	Server has entered a shutdown state due to normal process control.
1304	W	L	N	Application is attempting to switch identity: (<old identity>)-->(<new identity>).	No action required.

ERRORS

Message ID				Explanation	Resolution
1501	E	X	U	Unable to initialize application state.	Error occurred while reading internal configuration files. Please contact technical support.

1502	E	X	D	Unable to write to system log: (<path to log file>).	Error occurred while writing to log file. Verify log file directory is writable and user ID has appropriate permissions.
1503	E	X	D	Unable to execute user notification. Return code from OS Command (<error code>)	Verify that the notification command can be executed. Refer to User Guide.
1504	E	L/N3	D	Unable to allocate dynamic memory.	Check system resources.
1505	E	L/N3	D	System call failure (<system call>): (<OS specific error>).	Contact technical support.
1506	E	L/N3	S	The process configuration path length exceeds the maximum allowed.	Reinstall to allocation with a shorter install path length. See your OS documentation for path length limits.
1507	E	L/N3	S	Unable to create a data directory (<directory name>): (<OS specific error msg>).	See OS error message and correct as appropriate.
1508	E	X	S	Unable to initialize the Exec sub-system.	Contact technical support.
1509	E	X	S	The CONNX registry cannot be read or does not contain a value for: (CONNX.REPLICATION.INSTALLPATH).	Windows: Verify that the INSTALLPATH value is correct in the CONNX Configuration Manager (CONNX\Replication). Unix/Linux: Use the SqlRegistry tool to verify the INSTALLPATH value is correct.
1510	E	L/N3	S	Event Replicator message API version mismatch.	Verify all Event Replicator components are upgraded to a compatible version.
1511	E	L/N3	D	Unable to resolve host (<host>).	DNS name did not resolve. Specify dot notation addresses.
1512	E	L	N	Bad multi-packet message. Fragment sequence out of order.	Message queue internal error. Contact technical support.
1800	E	X	U	The process configuration path is either blank or the length exceeds the maximum allowed.	Reinstall to allocation with a shorter install path length. See your OS documentation for path length limits.
1801	E	X	U	The process name is either blank or the length exceeds the maximum allowed.	Internal error. Contact technical support.

1802	E	X	U	The specified configuration directory does not exist.	Windows: Verify that the INSTALLPATH value is correct in the CONNX Configuration Manager (CONNX\Replication) Unix/Linux: Use the SqlRegistry tool to verify the INSTALLPATH value is correct
1803	E	X	U	Unable to create a log file directory.	Verify installation directory is writable and user ID has appropriate permissions.
1804	E	X	U	Unable to initialize process log (<logfile>)	Verify installation directory is writable and user ID has appropriate permissions.
1805	E	X	U	Unable to install signal handler: (<OS-specific error message>).	See OS error message and correct as appropriate.
1806	E	X	U	Application is attempting to switch identity: (<Engine ID>-->(<Engine ID>).	Error notification is enabled and the notification command is either blank or the length exceeds the maximum allowed.
1807	E	X	U	The process version is either blank or the length exceeds the maximum allowed.	Contact technical support.

Admin

WARNINGS

Message ID				Explanation	Resolution
2001	W	L	N	Deployment message queued.	Deployment message was sent to the controller's message queue. No action required.
2002	W	L	N	Preparing to queue Deployment message.	No action required.
2003	W	L	N	Deployment CDD Initialized.	No action required.

ERRORS

Message ID	Explanation			Resolution
------------	-------------	--	--	------------

2501	E	L	N	Invalid configuration path.	<p>Path does not exist.</p> <p>Windows: Verify that the INSTALLPATH value is correct in the CONNX Configuration Manager (CONNX\Replication).</p> <p>Unix/Linux: Use the SqlRegistry tool to verify the INSTALLPATH value is correct.</p>
2502	E	L	N	Unable to open master configuration file: API Error (<OS-specific error>).	Verify installation directory is writable to the Admin process.
2503	E	L	N	Unable to initialize replications.	Contact technical support.
2504	E	L	N	Unsupported source database type.	An internal error has occurred in the deployed CDD. Contact technical support.
2505	E	L	N	Replication configuration contains no active replications.	Deploy replications.
2506	E	L	N	System call failure (<system call>): (<OS specific error>).	See OS error message and correct as appropriate.
2507	E	L	N	Unable to open master configuration file: IO Error (<OS specific error>).	Ensure the install directory is writable and user ID has appropriate permissions. See OS error message and correct as appropriate.
2508	E	L	N	Unable to connect to Replicator queue service on (<queue server host>):(<server port>): (<queue sub-system error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
2509	E	L	N	Unable to open Replicator work queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.

2510	E	L	N	Unable to enqueue work message: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
2511	E	L	N	Unable to retrieve Support DLL instance.	Contact technical support.
2512	E	L	N	Unable to initialize Support DLL.	Contact technical support.
2513	E	L	N	Unable to find CDD Object (<CDD object>).	Contact technical support.
2514	E	L	N	Unable to update CDD Object (<CDD object>).	Contact technical support.
2515	E	L	N	Unable to find Replication server information.	Contact technical support.
2516	E	L	N	Unable to open Replication Work Queue (<queue subsystem error>). Queue does not exist.	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
2517	E	L	N	Unable to initialize Replication Server Map.	Contact technical support.
2518	E	L	N	Bad Reply from Status Message.	Contact technical support.
2519	E	L	N	Replication Server Map not found.	CDD not deployed properly. Re-deploy CDD.
2520	E	L	N	Failed to retrieve Server Status.	Refresh Server Status. If the problem persists, contact technical support.
2521	E	L	N	Failed to write Server Status.	Ensure that install directory is writable to the current process.
2522	E	L	N	Invalid Server Status path: (<server status database path>).	Ensure that INSTALLPATH is set correctly.
2523	E	L	N	Invalid replication server ID: (<replication server ID>).	Contact technical support.

2524	E	L	N	Invalid replication server name: (< replication server name >).	Validate the server name on the Configure Servers dialog in the Replication Administrator. If the server name is correct, ensure that the server is connected to the network and is operating correctly.
2525	E	L	N	Invalid server state request: (< server state >).	Contact technical support.
2526	E	L	N	Unable to delete status queue (< queue name >): (< queue subsystem error >).	Non-fatal error; if problem persists, start and stop the CONNX message queue service (Windows) or Daemon (Unix/Linux).
2527	E	L	N	Invalid target table ID: (< table ID >).	Contact technical support.
2528	E	L	N	Configuration API error: (< API error >).	Contact technical support.
2529	E	L	N	Failed to purge existing Server Status.	Ensure that the install directory is writable to the Replication Administrator and retry Server Status. If problem persists, contact technical support.
2530	E	L	N	Failed to dequeue Server Status: (< queue subsystem error >).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
2531	E	L	N	Incompatible API version in ADM message reply.	One or more components are at a different version level than the Administrator. Ensure that all components of the Replication Server have the same build number and version.
2532	E	L	N	Unable to allocate dynamic memory.	Check system resources for available memory.
2533	E	L	N	Unable to initialize status queue. Unable to report status.	Ensure that the CONNX message queue service is running.
2534	E	L	N	Unable to gain exclusive access to deployment CDD.	Ensure that the CDD is not open by another Event Replication Administrator or the CONNX Data Dictionary Manager.
2535	E	L	N	Unable to purge the Controller work queue (< queue name >): (< queue subsystem error >).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication

				Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
--	--	--	--	---

Controller

WARNINGS

Message ID				Explanation	Resolution
3001	W	L	N	De-initializing driver support failed.	Non-fatal warning. No action necessary.
3002	W	L	N	Recovering from an unclean shutdown.	Previous shutdown was not executed via normal process mechanism. No action necessary.
3003	W	L	N	Engine unreachable: PID:(<process ID>). Attempting to re-spawn: (<engine ID>).	Controller lost TCP/IP connection to Replication Engine and is attempting to restart Engine.
3004	W	L	S	Stop service request received.	No action necessary.
3005	W	L	N	Monitor accept for unknown engine: (<engine ID>).	No action necessary.
3006	W	L	N	Master configuration contains invalid engine count. Defaulting to 1.	No action necessary.
3007	W	L	N	Engine spawned: PID:(<process ID>). ID:(<engine ID>).	Engine has been started successfully. No action necessary.
3008	W	L	N	Engine initialization complete: PID:(<process ID>) ID:(<engine ID>).	No action necessary.
3009	W	L	N	Event producer initialization complete. ID:(<engine ID>).	No action necessary.
3010	W	L	N	Deploy succeeded.	No action necessary.
3011	W	L	N	Deploy failed.	Check the Controller log file for the preceding messages containing additional error information and take action based on those messages.

ERRORS

Message ID				Explanation	Resolution
4001	E	L/N1	N	Unable to connect to local queue service: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4002	E	L/N1	N	Unable to open controller work queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4003	E	L/N2	N	Unable to dequeue work message: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4004	E	L/N3	N	Unable to open master configuration file: IO Error: (<OS specific error>).	See OS error message and correct as appropriate.
4005	E	L/N1	N	Unable to open EP Identity queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4006	E	L/N3	N	Unable to spawn engine: (<OS specific error>).	See OS error message and correct as appropriate.
4007	E	L/N1	N	Unable to connect to EP queue service: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.

4008	E	L/N1	N	Unable to open EP work queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4009	E	L/N1	N	Unable to open EP target queue (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4010	E	L/N2	N	Unable to enqueue EP work message: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4011	E	L/N2	N	Unable to enqueue Engine work message: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4012	E	L/N3	N	Unable to open master configuration file: (<OS specific error>).	See OS error message and correct as appropriate.
4013	E	L/N3	N	Unable to initialize replications: (<API error message>).	Contact technical support.
4014	E	L/N3	N	Unsupported backend for replication source.	Configuration files may have been modified outside Event Replication. Re-deploy replications.
4015	E	L/N1	N	Invalid target table id for replication	Configuration files may have been modified outside Event Replication. Re-deploy replications.
4016	E	L/N1	N	Replication configuration contains no active replications.	Configuration files may have been modified outside Event Replication. Re-deploy replications.

4017	E	L/N3	N	Unable to write master configuration file: IO Error: (<OS specific error>).	Verify install directory is writable and user ID has appropriate permissions.
4018	E	L/N3	S	Unable to open replication license file: (<API error message>).	Ensure that a valid Event Replication license is installed and registered.
4019	E	L/N3	S	Unable to lock replication license file: (<API error message>).	User license count has been exceeded. Ensure that a valid Event Replication license is installed and registered.
4020	E	L/N3	S	Driver support initialization failed: (<API error message>).	Windows: DLL was not registered. Ensure that the system was rebooted after installation. Unix/Linux: Shared library could not be located or loaded. Check dependencies.
4021	E	L	N	Unable to delete failed configuration: (<config file name>): (<OS specific error>).	Ensure that the install directory is writable to the Controller user/process. See OS error message and correct as appropriate.
4022	E	L	N	Bad work message. Unexpected message fragment type: (<embedded message type>).	Contact technical support.
4023	E	L	N	Unable to flush configuration to disk: (<OS specific error>).	See OS error message and correct as appropriate.
4024	E	L	N	Unable to open existing config (<config file name>): (<OS specific error>) Starting over.	See OS error message and correct as appropriate.
4025	E	L	N	Unable to map existing config (<config file name>): (<OS specific error>). Starting over.	See OS error message and correct as appropriate.
4026	E	L	N	Configuration mapped to NULL. Starting over.	Out of resources. Check disk space and available memory.
4027	E	L	N	Unable to create new configuration (<config file name>): (<OS specific error>).	Out of resources. Check disk space and available memory.

4028	E	L	N	Unable to map new configuration (<config file name>): (<OS specific error>).	Out of resources. Check disk space and available memory.
4029	E	L	N	New configuration mapped to NULL.	Out of resources. Check disk space and available memory.
4030	E	L/N1	N	Unable to initialize target queues. Exceeded maximum attempts to connect to queue service.	Ensure that the CONNX message queue service is running.
4031	E	L/N1	N	Invalid replication server ID. Unable to process configuration.	Configuration files may have been modified outside Event Replication. Re-deploy replications.
4032	E	L/N1	N	Cannot retrieve object from master configuration: (<configuration object>).	Contact technical support.
4033	E	L/N1	N	Invalid source table id for replication: (<table ID>).	Contact technical support.
4035	E	L/N1	C	Socket error sending status change to producer	This error indicates a problem with the communication between the controller and the producer. Ensure that the source nucleus is running and the nucleus log indicates that replication was successfully initialized. Also ensure that there are no firewalls present between the controller and the producer that would prevent communication. If the producer is running and there are no network issues, contact technical support.
4036	E	L/N1	N	Unable to create control reply queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4037	E	L/N1	S	Configuration reset. Stopping all servers and shutting down.	Invalid license or all active replications removed from this CDD. Either correct the license file,
4038	E	L/N3	S	Unable to accept connections on port (<port number>).	Attempted to accept connections on the first available port. No ports were available. Ensure that there is a range of ports available. If configured to a specific port, ensure that port is

					available.
4039	E	L	N	Monitor request timed out.	Attempted to communicate with the Replication Engine but did not receive a timely reply. Check the Event Consumer log files.
4040	E	L	N	Unable to send engine monitor request: (<socket error>).	TCP/IP error. Check the Event Consumer log files.
4041	E	L	N	Monitor accept for active or deactivated engine.	Sequence error detected and corrected. No action necessary.
4042	E	L	N	Monitor unable to retrieve configuration for re-spawned engine.	Controller is in an invalid state. Contact technical support.
4043	E	L	N	Monitor timed out sending configuration to re-spawned engine.	Communication with engine lost; system will attempt to reconnect. No action necessary.
4044	E	L	N	Monitor error sending configuration to re-spawned engine: (<socket error>).	TCP/IP error. Check the Event Consumer log files.
4045	E	L	N	Monitor request reply timed out.	Engine failed to respond in a timely manner. Check the Event Consumer log files.
4046	E	L	N	Unable to receive engine monitor request reply: (<socket error>).	TCP/IP error. Check the Event Consumer log files.
4047	E	L/N3	S	Unable to restart acceptor for connections on port (<port number>).	Controller was configured to accept on a specific port and port is no longer available. Specify an available port. or Controller was configured to accept on a range of ports and no ports in the range were available. Ensure that there are available ports.
4048	E	L/N3	S	Monitor accept for an engine with a mismatched process ID (<engine ID>_<engine ID>): (<current process ID>) != (<incoming process ID>).	Contact technical support.
4049	E	L	N	Database specific configuration missing table descriptors.	Contact technical support.

4050	E	L	N	Invalid user/pass length in master configuration. Please re-configure and re-deploy.	Contact technical support.
4051	E	L/N1	N	Invalid source database id for replication: (< database ID >).	Contact technical support.
4052	E	L	N	Connection timed out sending configuration to engine.	Non-fatal error; engine will re-attempt connection. No action necessary.
4053	E	L	N	Connection error sending configuration to engine: (< socket error >).	TCP/IP error. Check the Event Consumer log files.
4054	E	L	N	Unknown engine dereferenced: (< engine ID >).	Contact technical support.
4055	E	L	N	Unable to connect to Admin queue service: (< queue subsystem error >).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4056	E	L	N	Unable to open Admin status queue: (< queue subsystem error >).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4057	E	L	N	Unable to enqueue message on Admin status queue: (< queue subsystem error >).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
4058	E	L/N1	N	Master configuration file does not exist: (< config file name >). Unable to process.	Configuration files may have been deleted outside Event Replication. Redeploy replication CDD.
4059	E	L/N1	N	Unable to load replication columns from master configuration for ID: (< table ID >). Error: (< API error >).	Contact technical support.

4060	E	L	N	Unable to remove old configuration: (<config file name>) Error: (<OS specific error>).	See OS error message and correct as appropriate.
4061	E	L	N	Unable to initialize engine: PID: (<process ID>) ID: (<engine ID>).	Contact technical support.
4062	E	L	N	Unable to initialize event producer.	Contact technical support.
4063	E	L	N	Monitor accept for disabled engine.	Sequence error detected and corrected. No action necessary.
4064	E	L/N3	N	Engine request for respawn denied. Engine disabled: PID: (<process ID>) ID: (<engine ID>). Please contact technical support.	Contact technical support.
4065	E	L	N	Version mismatch for engine: ID: (<engine ID>).	One or more components are at a different version level than the Controller. Ensure that all components of the Replication Server have the same build number and version.
4066	E	L	N	Engine configuration status check send timed out. ID: (<engine ID>).	Ensure that the system has sufficient resources and that the install directory is readable and writable. Check the Engine logs.
4067	E	L	N	Engine configuration status check send error: ID: (<engine ID>) Error: (<socket error>).	Ensure that the system has sufficient resources and that the install directory is readable and writable. Check the Engine logs.
4068	E	L	N	Engine configuration status check reply error: ID:(<engine ID>) Error: (Status not NORMAL)	Check the Engine logs.
4069	E	L	N	Engine configuration status check reply timed out. ID: (<engine ID>).	Ensure that the system has sufficient resources and that the install directory is readable and writable. Check the Engine logs.
4070	E	L	N	Engine configuration status check reply error: ID: (<engine ID>) Error: (<socket error>).	Ensure that the system has sufficient resources and that the install directory is readable and writable. Check the Engine logs.

4071	E	L	N	Unable to create Controller to EP status queue: (<API error>).	Ensure that the system has sufficient resources and that the install directory is readable and writable.
4072	E	L	N	The Controller to EP status queue has been deleted: (<queue name>).	Contact technical support.
4073	E	L	N	Unable to open Controller to EP status queue (<queue name>): (<queue API error>).	Ensure that the system has sufficient resources and that the install directory is readable and writable.
4074	E	L	N	Controller to EP status queue message version mismatch detected.	Ensure that the latest version of the Event Replicator is installed on all machines.
4075	E	L	N	Unknown message type encountered on Controller to EP status queue: (<internal message type>).	Contact technical support.
4076	E	L	N	The EP has reported a failed deploy.	Consult the appropriate EP log and correct environment. Re-deploy.
4077	E	L	N	Unknown EP encountered in post-deploy enquiry: (<EP id>).	Contact technical support.
4078	E	L	N	Error in message dequeue on Controller to EP status queue (<queue name>): (<queue API error>).	Ensure that the CONNX message queue is active on all machines involved in replication, the queue directory is readable and writable, and that the system has sufficient resources.
4079	E	L	N	Duplicate source column detected: TBL: (<table name>) COL: (<column name>).	Contact technical support.
4080	E	L	N	Unable to resolve target table column: TBL: (<table name>) COL: (<column name>).	Contact technical support.

4081	E	L	N	Non-unique table column ordinal detected: (<table name>).	Contact technical support.
4082	E	L	N	Fatal error. Unable to configure producer identities.	Contact technical support.
4083	E	L	N	Post-deploy EP status check exceeded maximum time limit.	Ensure that the Adabas nucleus has been started and has been correctly configured for replication.
4084	E	L	N	Source table has no index defined: (<table name>).	Replications must be configured with at least one primary index.
4085	E	L	N	Target table has no index defined: (<table name>).	Replications must be configured with at least one primary index.
4086	E	L/N1	N	Unable to open license for database involved in replication (<DB name>): (<API error message>)	Each database involved in replication must have a corresponding license file. Ensure that your license server is properly configured and running.
4087	E	L/N1	N	Unable to lock license for database involved in replication (<DB name>): (<API error message>)	Each database involved in replication must have a corresponding license file. Ensure that your license server is properly configured and running.
4200	E	L	N	Unrecognized ADABAS table name format: (<Adabas table name>). Table: (<table name>).	Contact technical support.
4201	E	L	N	Invalid column name exceeds maximum length. Column: (<column name>). Table: (<table name>).	Contact technical support.
4202	E	L	N	Invalid virtual column type in configuration. Column: (<column name>). Table: (<table name>).	Contact technical support.
4203	E	L	N	Invalid index column type in configuration. Column: (<column name>). Table: (<table name>).	Contact technical support.

4204	E	L	N	Duplicate table encountered in ADABAS cluster definition: Table: (<table name>): (<int>).	Contact technical support.
4205	E	L	N	Non-MU/PE column encountered in rotated table descriptors (<column name>): (<table name>).	Contact technical support.
4206	E	L	N	MU/PE column encountered in table index descriptors (<column name>): (<table name>).	Contact technical support.
4207	E	L	N	Unrecognized physical DB Name format for ADABAS: (<internal DB name>).	Contact technical support.
4208	E	L	N	Unable to create queue name. Max queue name exceeded (<max value>): (<supplied value>).	Contact technical support.
4209	E	L	N	No ADABAS cluster definitions match the requested source information.	Contact technical support.
4303	E	L/N1	N	Unable to perform Adabas to Adabas Initial State	The Adabas to Adabas initial state process uses the Adabas utility ADABCK to do a DUMP of the source files and an OVERLAY on the target. Error 4303 indicates that there was an error in either the source or target ADABCK process. Please check the ADABCK log files for information about the error. These log files are located in the replicator's log directory (connx32\replication\log on Windows and connx/replicator/log on Unix/Linux) in a subdirectory named DBxxx where xxx is the 3 digit number of the database ID. Please check the logs for both the source and target. Note, if an error on the source prevented any processing, there may not be an ADABCK log on the target.
Event Consumer					

WARNINGS

Message ID				Explanation	Resolution
5001	W	L	C	Loading new EC configuration.	Informational.
5002	W	L	D	Request to DISABLE PROCESS received.	Informational. A Stop Target was issued from the Administrator.
5003	W	L	N	Request to initial state source.	Informational.
5004	W	L	U	EC Configuration locked by another process: (<Configuration Path>). Exiting.	Duplicate engine respawned due to temporary communication failure. No action required.
5005	W	L	N	Configuration is dirty: recovering from bad shutdown.	Informational
5006	W	L	N	Engine active: Source: (<DB name>):(<internal DB ID>) --> Target: (<DB name>):(<internal DB ID>).	Informational
5201, 5202	W	L	N	Record information pertaining to a 5203 informational message	Informational
5203	W	L	N	Engine encountered an SQL error. The operation will be retried. The SQL error can be found in the body of the error message. This message is issued for SQL errors that are transient. They normally succeed on a retry. If they do not succeed after 3 retries, error 5702 will be issued.	Informational
5301	W	L	S	Shut down occurring before enqueue of Initial State end card. Initial state will be replayed.	Informational.
5302	W	L	S	Shut down occurring before dequeue of Initial State. Initial State will be replayed.	Informational.
5303	W	L	S	Shut down occurring before dequeue of transaction.	Informational.

ERRORS

Message ID				Explanation	Resolution
------------	--	--	--	-------------	------------

5501	E	L/N2	N	Unable to connect to EP queue service: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5502	E	L/N2	N	Unable to open cursor for EP transaction queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5503	E	L/N2	N	Unable to get message from EP transaction queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5504	E	L/N1	N	Unable to dequeue message from EC work queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information...
5505	E	L/N1	N	Unable to connect to local queue service: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5506	E	L/N1	N	Unable to open EC work queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5507	E	L/N3	N	Unable to create new replication.	Check system resources for available memory.

5508	E	X	U	Unable to initialize the process name.	Non-fatal error. No action necessary.
5509	E	L	N	Unable to open connection to status queue service on (<queue service host>): (<port>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5510	E	L	N	Unable to open status queue (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5511	E	L/N3	D	Unable to set the process name. Unable to initialize logfile. Deploy failed.	Contact technical support.
5512	E	L/N3	D	Null record loader returned. Unable to continue processing.	Check system resources for available memory.
5513	E	L/N3	D	Duplicate initial state end received.	Contact technical support.
5514	E	L/N3	D	Unexpected message in queue.	Contact technical support.
5515	E	L	N	Unable to open existing config (<config file name>): (<OS specific error>). Starting over.	Non-fatal error. System will recover automatically.
5516	E	L	N	Unable to map existing config (<config file name>): (<OS specific error>). Starting over.	Non-fatal error. System will recover automatically.
5517	E	L	N	Configuration mapped to NULL. Starting over.	Out of resources. Check disk space and available memory.
5518	E	L	N	Configuration file does not belong to this engine: Expected: (<engine ID>) Mapped: (<engine ID>). Starting over.	Non-fatal error. System will recover automatically.
5519	E	L/N3	S	Unable to create new configuration (<config file name>): (<OS specific error>).	Out of resources. Check disk space and available memory. See OS error message and correct as appropriate.

5520	E	L/N3	S	Unable to map new configuration (<config file name>): (<OS specific error>).	Out of resources. Check disk space and available memory. See OS error message and correct as appropriate.
5521	E	L/N3	S	New configuration mapped to NULL.	Out of resources. Check disk space and available memory.
5522	E	L	N	Configuration is dirty: recovering from bad shutdown.	Non-fatal error. No action necessary.
5523	E	L/N3	S	Unable to flush configuration to disk: (<OS specific error>).	See OS error message and correct as appropriate.
5524	E	L/N3	S	Unexpected message state initializing queue.	Contact technical support.
5525	E	L/N3	S	Duplicate initial state found while initializing queue.	Contact technical support.
5526	E	L	N	Unable to enqueue status message on queue (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5527	E	L	N	Unable to set SQL statement for replicating: Length is zero.	Contact technical support.
5528	E	L	N	Unable to enqueue init marker: (<queue interface error>).	Ensure that the CONNX message queue service is running.
5531	E	L/N3	D	Unable to restart engine. Queue initialization failure.	Contact technical support.
5532	E	L/N3	S	Control host address has an invalid length: (<host address>).	Contact technical support.
5533	E	L/N3	S	Unable to connect to Controller service: (<socket error>).	Check controller log.
5534	E	L	S	Timeout waiting for initial configuration. Exiting.	System unable to process all replications. Check for resource availability.

5535	E	L	S	Connection to Controller service lost: (<socket error>). Exiting.	Check controller log.
5536	E	L/N3	D	Corrupt queue. Partial message detected.	Contact technical support.
5537	E	L/N3	D	Corrupt queue. Message sequence out of order.	Contact technical support.
5538	E	L/N3	D	Corrupt queue. Message sequence out of order.	Contact technical support.
5539	E	L/N3	D	Corrupt queue. Message sequence out of order.	Contact technical support.
5540	E	L/N3	D	Corrupt queue. Message sequence out of order.	Contact technical support.
5541	E	L/N3	D	Corrupt queue. Message sequence out of order.	Contact technical support.
5700	E	L/N2	D	Unable to connect to Target DB: (<SQL error>).	Check API error and correct problem. When problem has been corrected, restart servers from the Status Tab in the Replication Administrator.
5701	E	L/N2	D	End Transaction error: (<SQL error>).	Check SQL error and correct problem. When problem has been corrected, restart servers from the Status Tab in the Replication Administrator.
5702	E	L/N2	N	SQL error (<SQL error type>): (<SQL error>). Final try, giving up. Process will disable.	Target table may have been modified externally. Check SQL error and correct problem. When problem has been corrected, restart servers from the Status Tab in the Replication Administrator. An initial state may be required.
5703	E	L/N2	N	SQL action affected 0 records: (<SQL error>).	Target table may have been modified externally. An initial state may be required.

5704	E	L	N	Unable to set connection information: Logical DB name length exceeds maximum: (<logical DB name>).	Contact technical support.
5705	E	L	N	Unable to set connection information: Connection string length exceeds maximum: (<connection string>).	Contact technical support.
5706	E	L/N3	D	Unable to set connection attributes: (<SQL error>).	Contact technical support.
5707	E	L/N2	D	Current transaction rolled back.	Application state is disabled. Check Event Consumer logs, fix problems and re-deploy.
5708	E	L/N2	D	Begin transaction failed: (<SQL error>).	Application state is disabled. Check Event Consumer logs, fix problems and re-deploy.
5709	E	L/N2	D	Invalid transaction. NULL source map.	Contact technical support.
5710	E	L	N	SQL error: (<SQL error>).	Transaction event has failed. Event Consumer will re-connect and retry the transaction.
5711	E	L	N	Current transaction rolled back.	No action required.
5712	E	L/N2	D	Application has entered an exception state. Current transaction rolled back	No action required.
5713	E	L/N3	D	An error occurred during initial state processing.	Compound error state within the engine is preventing clean error handling. Prior error messages contain additional information. Contact technical support.
5800	E	L/N2	N	Unable to connect to EP queue service: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.

5801	E	L/N2	N	Unable to open EP transaction queue (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5802	E	L/N2	N	Unable to enqueue message on EP transaction queue (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5803	E	L/N2	N	Unable to dequeue message from EP transaction queue (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
5804	E	L/N2	S	Error in COMMIT. Transaction not committable due to application exception state.	Compound error state within the engine is preventing clean error handling. Prior error messages contain additional information. Contact technical support.
5805	E	L/N2	S	Unexpected end of queue. Transaction manager out of sync with event consumer. QUEUE:(<queue name>).	Contact technical support.

Event Producer

WARNINGS

Message ID				Explanation	Resolution
6001	W	L	U	No existing configuration found. Creating new configuration.	Informational.
6002	W	L	U	Event producer: Identity configured.	Informational.

6003	W	L	N	The transaction cache is not empty at shutdown.	Informational.
6004	W	L	N	The transaction queue is not empty at shutdown.	Informational.
6005	W	L	N	Processing empty transaction.	Informational.
6006	W	L	D	Resetting the Event Producer. Deleting current configuration.	Informational. Response to a request to remove all replications.
6007	W	L	U	Recovering from dirty shutdown.	Informational.
6008	W	L	C	Processing existing configuration.	Informational.
6009	W	L	N	Existing configuration loaded.	Informational.
6010	W	L	C	Processing new configuration.	Informational.
6011	W	L	N	Dynamic configuration loaded.	Informational.
6012	W	L	N	Process configuration unloaded successfully	Informational.
6200	W	L	N	Spawned ep_config: PORT:(<port number>).	Informational.
6250	W	L	S	Listener exiting main control loop.	Informational.

ERRORS

Message ID				Explanation	Resolution
7001	E	L/N2	N	Unable to connect to local queue service: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7002	E	L/N2	N	Unable to open target queue (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.

7003	E	L/N2	N	Unable to enqueue on target queue (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7004	E	L/N2	N	Unable to open EP work queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7005	E	L	N	Unable to delete unused record caches: DIR:(<queue directory>) ERR:(<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7006	E	L/N2	N	Unable to dequeue EP work queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7007	E	L	S	Queue unreachable and shutdown called during transaction queue processing: TXID:(<internal transaction ID>).	Ensure that the CONNX message queue is the last process to stop when stopping all Event Replicator services.
7008	E	L	S	Queue unreachable and shutdown called during initial state queue processing: TXID: (<queue subsystem error>).	Ensure that the CONNX message queue is the last process to stop when stopping all Event Replicator services.
7009	E	L	N	Unable to purge target queue (<queue name>): (<queue subsystem error>).	Ensure that the CONNX message queue is the last process to stop when stopping all Event Replicator services.
7010	E	L/N2	N	EP configuration error: duplicate source.	Contact technical support.

7011	E	L/N2	N	Unable to open EP control queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7012	E	L/N2	N	Unable to peek EP control queue: (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7013	E	L	U	Unable to open configuration cache for EP (<config file name>): IO Error: (<OS specific error>).	Verify installation directory is readable and user ID has appropriate permissions. See OS error message and correct as appropriate.
7014	E	L	U	Unable to read cached configuration for EP (<config file name>): IO Error: (<OS specific error>)	Verify installation directory is readable and user ID has appropriate permissions. See OS error message and correct as appropriate.
7015	E	L	U	Unable to create configuration cache for EP (<config file name>): IO Error: (<OS specific error>)	Verify installation directory is writable and user ID has appropriate permissions. See OS error message and correct as appropriate.
7016	E	L	U	Unable to cache configuration for EP (<config file name>): IO Error: (<OS specific error>).	Verify installation directory is writable and user ID has appropriate permissions. See OS error message and correct as appropriate.
7017	E	L/N1	U	Unable to open existing configuration (<config file name>): (<OS specific error>). Initial state required.	Verify installation directory is writable and user ID has appropriate permissions. See OS error message and correct as appropriate.
7018	E	L/N1	U	Unable to map existing configuration (<config file name>): (<OS specific error>).	Out of resources. Check disk space and available memory.
7019	E	L/N1	U	Pointer to existing configuration is NULL: (<OS specific error>).	Out of resources. Check disk space and available memory.
7020	E	L/N3	U	Unable to initialize mapped allocator: (<API error>).	Contact technical support.

7021	E	L/N3	U	Unable to initialize transaction data.	Contact technical support.
7022	E	L/N3	U	Unable to initialize record data.	Contact technical support.
7023	E	X	S	The CONNX registry cannot be read or does not contain a value for: (CONNX.REPLICATION.INSTALLPATH).	Adabas nucleus was terminated unexpectedly. Initial state required to regain data integrity.
7024	E	L	U	Unable to delete dirty configuration (<config file name>): (<OS specific error>).	Verify installation directory is writable and user ID has appropriate permissions. See OS error message and correct as appropriate.
7025	E	L/N3	D	Unable to create new configuration (<config file name>): (<OS specific error>).	Verify installation directory is writable and user ID has appropriate permissions. See OS error message and correct as appropriate.
7026	E	L/N3	D	Unable to map new configuration (<config file name>): (<OS specific error>).	Out of resources. Check disk space and available memory.
7027	E	L/N3	D	Pointer to new configuration is NULL: (<OS specific error>).	Out of resources. Check disk space and available memory.
7028	E	L/N3	D	Unable to construct new mapped allocator: (<API error>).	Contact technical support.
7029	E	L/N3	D	Unable to construct new transaction data.	Contact technical support.
7030	E	L/N3	D	Unable to construct new record data.	Contact technical support.
7031	E	L/N3	D	Unable to flush the configuration to disk: (<OS specific error>).	Verify installation directory is writable and user ID has appropriate permissions. See OS error message and correct as appropriate.
7032	E	L	N	Unable to connect to status queue.	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7033	E	L	N	Unable to open status queue.	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable

					to the current process. See the User Guide for more information.
7034	E	L	N	Unable to enqueue status message.	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7035	E	L	N	Unable to delete cache map (<queue name>): (<queue subsystem error>).	Ensure that the message queue has been started and initialized correctly, the port number in the Replication Administrator matches the port number in the Registry, and the message queue directory is writable to the current process. See the User Guide for more information.
7036	E	LN/3	U	Unable to load cached configuration. Deployment required.	Ensure that the install directory is writable to the Event Producer user/process and re-deploy.
7037	E	L	S	Unable to reset the Event Producer. Forcing internal replication processes to shutdown.	Contact technical support.
7038	E	L/N3	D	Duplicate transaction requested.	Contact technical support.
7039	E	L/N3	D	Unable to retrieve pointer to mapped memory: (<API error>).	Contact technical support.
7040	E	L/N3	D	Unable to free pointer to mapped memory: (<API error>).	Contact technical support.
7041	E	L/N3	D	Unable to cache record. Invalid transaction ID.	Contact technical support.
7042	E	L/N1	D	Unable to retrieve transaction data alloc info.	Contact technical support.
7043	E	L/N1	D	Unable to retrieve record data alloc info.	Contact technical support.
7044	E	L	N	Initial state request received when no configuration exists.	Non-fatal error. No action necessary.

7045	E	L	N	Delete config request received when no configuration exists.	Non-fatal error. No action necessary.
7046	E	L/N2	N	EP configuration error: Duplicate source.	Contact technical support.
7047	E	L/N3	S	Unable to initialize the Event Producer. Forcing internal replication processes to shutdown.	Contact technical support.
7048	E	LN/3	D	Unable to cache configuration.	Ensure that the install directory is writable to the Event Producer user/process and then re-deploy.
7049	E	LN/3	S	Number of cached transactions is non-zero and the configuration is new.	Contact technical support.
7050	E	LN/3	S	Number of cached transactions does not match the saved configuration.	Contact technical support.
7051	E	LN/3	S	Number of cached transactions does not match the number of restored transactions.	Contact technical support.
7052	E	LN/3	S	Unable to reconstitute record data: (<API error>)	Contact technical support.
7053	E	LN/3	S	Number of cached transactions is zero and the number of cached records is non-zero.	Contact technical support.
7071	E	LN/3	N	Adabas to Adabas Deploy Failed.	Check previous error messages in the EP log. Check Nucleus log.
7072	E	LN/3	N	Call to Adabas nucleus to change replication status failed.	Check previous messages in the EP log for more detail. Check Nucleus log. Possible causes of this error are adding a file to the replication list that has a referential integrity dependency on another file that was not added or attempting to do an initial state on a file that has a referential integrity dependency on another file that was not part of the initial state process - most likely because it was not added to the replication list.
7200	E	L/N3	D	Internal API call failed. Nucleus Response: (<nucleus response code>).	Contact technical support.
7201	E	L/N3	D	Unable to retrieve Primary Key data.	Contact technical support.
7202	E	L/N3	D	Internal call failed. Nucleus Response: (<nucleus response code>).	Contact technical support.

7203	E	L/N	D	Unable to resolve sub tables: (<API error>).	Contact technical support.
7204	E	L/N3		Event handler termination called with non-zero reference count.	Adabas has terminated abnormally. Contact technical support.
7205	E	L/N3		FATAL ERROR: Unable to initialize producer.	Contact technical support.
7206	E	L		FATAL ERROR: Unable to initialize producer global status framework.	Contact technical support.
7207	E	L/N3		FATAL ERROR: Unable to initialize ep_config.	Contact technical support.
7208	E	L/N3		FATAL ERROR: Unable to create handler.	Contact technical support.
7210	E	L		Unable to send message to ep_config: (<socket error>).	System will try to restart CNXEPCFG automatically. If it doesn't, contact technical support.
7211	E	L/N3		Unable to send message to ep_config: (<socket error>).	System will try to restart CNXEPCFG automatically. If it doesn't, contact technical support.
7212	E	L/N3		Failed to retrieve reply message to ep_config: (<socket error>).	System will try to restart CNXEPCFG automatically. If it doesn't, contact technical support.
7213	E	L/N3		Unable to spawn ep_config: (<OS specific error>).	See OS error message and correct as appropriate.
7215	E	L/N3		Unable to open connection to ep_config: PORT:(<port number>) ERR:(<socket error>).	Contact technical support.
7250	E	L		Unable to initialize socket acceptor on port (<port number>) (<socket error>).	See OS error message and correct as appropriate.
7251	E	L		Unable to accept incoming socket: (<socket error>). Will attempt to re-initialize acceptor.	Non-fatal error.

7252	E	L		Unable to re-initialize socket acceptor on port (<port number>) (<socket error>).	See OS error message and correct as appropriate.
7254	E	L		Receive failed for incoming request: (<host>)(<port number>): (<socket error>).	Ensure that the Adabas nucleus has been started and has been correctly configured for replication.
7255	E	L		Invalid message type received: (<host>)(<port number>): (<socket error>).	Contact technical support.
7256	E	L		Timeout sending response to client: (<host>)(<port number>).	Ensure that the Adabas nucleus has been started and has been correctly configured for replication.
7257	E	L		Unable to send response to client: (<host>)(<port number>): (<socket error>).	Ensure that the Adabas nucleus has been started and has been correctly configured for replication.

Debug

WARNINGS

Message ID				Explanation	Resolution
99801	W	L	U	EXEC: { {string} }.	Internal use only. For debug level 2.
99802	W	L	U	ADM: { {string} }.	Internal use only. For debug level 2.
99803	W	L	U	CTRL: { {string} }.	Internal use only. For debug level 2.
99804	W	L	U	EC: { {string} }.	Internal use only. For debug level 2.
99805	W	L	U	RL: { {string} }.	Internal use only. For debug level 2.
99806	W	L	U	RL Pool: { {string} }.	Internal use only. For debug level 2.
99807	W	L	U	TX: { {string} }.	Internal use only. For debug level 2.
99808	W	L	U	EP: { {string} }.	Internal use only. For debug level 2.
99809	W	L	U	CTRL (ADABAS): { {string} }.	Internal use only. For debug level 2.
99810	W	L	U	EP (ADABAS): { {string} }.	Internal use only. For debug level 2.

Event Replicator User Exits

The Open Systems Event Replicator has a user exit which allows a you to run a job if an error with an Action type of Nx (see Event Replication Error Messages) occurs.

To run a user-specified job when Nx errors occur, set NOTIFY to 1 and NOTIFY_CMD to the name of the user-specified job.

NOTIFY_CMD should contain the fully qualified path and file name of the job to be executed. The job can be a program, shell script, batch file or command file. It can contain multiple tasks.

If an Nx error occurs, the Event Replicator calls the NOTIFY_CMD command and passes a single parameter string containing the job name. The system will execute the job specified by NOTIFY_CMD parameter string. If no job is specified, the NOTIFY_CMD default job is "eventnotify."

In the samples directory, there is a VB .NET 2003 job. It creates a sendmail program that sends email notifications to a list of people specified in the sample eventnotify.cmd file.

For more information, see the Event Replication Registry Settings.

Event Replication Troubleshooting Do's and Don'ts

Do's

- Do put all replications in one data dictionary; do not spread replications across multiple data dictionaries that will be sent to one controller. A controller can only run replications from one data dictionary at a time.
Example: If the data dictionary ex1.cdd is deployed to controller 1 and then data dictionary ex2.cdd is deployed to controller 1, the replications in ex1.cdd are removed and the replications in ex2.cdd replace them.
- Do check the log files when data is not replicating. Since the replication controller runs as a background service errors there is no user interface to display errors, so they are written to log files. The log files can be found in the folder at the Installation Directory \CONN32\Replication\log.

Don'ts

- Don't modify target data outside of the replication tool. If data on the target database is modified outside of the replication tool, it can cause integrity issues and stop the replication.
- Don't deploy the same Data Dictionary to different active controllers. If there are two different controllers modifying data in the same target table, you lose data integrity from the source to the target tables. This will disable the Event Replicator.

Index

A

About... 3, 6, 47, 93, 96, 129, 140, 145, 164, 182
about cloning 164
about obtaining..... 6
About Open Systems 140
Accept 1, 154, 184
accept incoming 184
Access 3, 184
ACID 3
 preserve 3
ACID stands 3
Action 184, 218
Action type 218
 Nx 218
Active.... 8, 87, 96, 100, 102, 103, 110, 123, 141,
 143, 145, 165, 182, 184, 219
Active checkbox 165
active Event Consumers 8
Active Servers 165
 Validating 165
Adabas 1, 6, 8, 10, 19, 29, 47, 51, 53, 64, 93,
 100, 112, 154, 164, 184
 changes 1
 CONNX..... 93
 data dictionary containing..... 6
 OLEDB Driver 1
Adabas ConnecX 1
ADABAS Count Selection window 29
Adabas Import..... 29
Adabas Nucleus 6, 112, 154, 184
Adabas SQL Gateway1, 6, 8, 10, 19, 29, 93, 164
Adabas SQL Gateway User Guide 29
Adabas Target Tables..... 51, 53, 64
Adabas v6.1 6
add 3, 8, 29, 41, 47, 53, 93, 100, 141, 164
 Source Database..... 93
 table 47
Add Row 51, 53
Add Tables 41
Adding New Column Mappings..... 53
Adjusting..... 150

Display..... 150

ADM 184
Admin 184
 connect..... 184
 writeable 184
Administrator 1, 184
Advanced Features..... 164
 CONNX 164
AIX 10
All 3, 6, 8, 29, 41, 51, 53, 64, 93, 96, 102, 103,
 110, 112, 116, 123, 164, 165, 182, 184, 219
 port number 93
ALL replications 110
alloc..... 184
Allows SQL 1
API 184
API Error 184
Apply 3, 8, 93, 164
Atomicity 3
 preserves..... 3
Available Tables..... 29
B
Bad Reply 184
Build Target..... 41, 102
Build Targets..... 100, 141
Building 100
 New 100
Business Case 1
C
C#.NET 6
cache 184
 EP..... 184
Call 1, 3, 154, 184, 218
called during..... 184
Cancel..... 29
CDD 8, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80,
 87, 88, 93, 96, 100, 102, 103, 105, 110, 112,
 116, 121, 123, 164, 165, 182, 184
 Select 51, 87, 88, 165
CDD Initialized 184
CDD Object..... 184

CDD Security	29	Config.....	93, 184
change.. 1, 3, 6, 8, 29, 47, 51, 53, 60, 64, 66, 71, 77, 80, 93, 103, 121, 150, 154, 182, 184		existing	184
Adabas.....	1	config file.....	184
Event Replication Administrator	8	Config Servers	41, 93
Source	51	Configuration API.....	184
change CONNX.REPLICATION.ENGINE_PORT	154	Configuration Path	184
port.....	154	Configuration Replication window.....	103
changes back	150	CONFIGURE	41, 93, 184
changes during.....	3	Configure Replication Server.....	93
changing DEBUG_LEVEL.....	182	Configure Servers dialog	184
Chapter 21-Advanced Features	164	Configure Servers window.....	41, 93
CONNX.....	164	Configuring.....	41, 93
Check ... 100, 102, 103, 105, 110, 116, 165, 182, 184, 219		Replication.....	93
Controller	184	Connecting.....	29, 184
Engine.....	184	Admin	184
Event Consumer.....	184	Controller.....	184
Check API	184	EP.....	184
Check Event Consumer	184	Replicator	184
Check log	184	SQL Server.....	29
Clear Active	41, 110, 141	Target DB	184
Clear Source	77, 80	ConnecX	1, 6, 8, 10, 19, 29, 164
Clear Target	77, 80	CONNX	1, 6, 8, 19, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 87, 88, 93, 96, 100, 102, 103, 105, 110, 112, 116, 154, 164, 166, 167, 168, 169, 182, 184
Clone Table	164	Advanced Features	164
Clone Table Assistant	164	Chapter 21-Advanced Features	164
Cntrl+O	121	CONNX 10.5	51, 87, 88
Cntrl+S	121	CONNX Basics	93
CNXEPCFG	184	CONNX Configuration	29, 184
CNXMQ	6	CONNX Configuration Manager	184
COL	184	CONNX Configuration Settings	29
column mapping.....	8, 41, 47, 53, 60, 77, 80	CONNX Data Dictionary.....	8, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 87, 88, 93, 96, 100, 102, 103, 105, 110, 112, 116, 184
Column Order.....	64	CONNX Data Dictionary Manager	29, 93, 184
Column Width.....	129, 150	CONNX Import Table Selection window.....	29
columns.. 1, 8, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 88, 96, 100, 102, 129, 134, 145, 150, 164, 165, 184		CONNX Logon Credentials.....	41, 93
Mapping	51	CONNX Message	6, 154, 184
command line.....	170	port number	154
commandline	170	CONNX Message Queue	6, 154, 184
COMMIT	184	CONNX Message Queue listening port.....	154
completes	3, 8, 96, 184	CONNX Solutions	19, 29, 41

- CONNX User Reference Guide93, 164
- CONNX.REPLICATION.CTRL_PORT 154
- CONNX.REPLICATION.CTRL_PORT specifies 154
- CONNX.REPLICATION.DEBUG_LEVEL 154
- CONNX.REPLICATION.DEBUG_LEVEL controls 154
- CONNX.REPLICATION.ENGINE_PORT 154
- CONNX.REPLICATION.EP_ADA_CFG_PORT 154
- CONNX.REPLICATION.EP_ADA_CFG_PORT specifies 154
- port..... 154
- CONNX.REPLICATION.INSTALLPATH 184
- CONNX.REPLICATION.MQ_PORT 154
- CONNX.REPLICATION.MQ_PORT specifies 154
- CONNX.REPLICATION.NOTIFY 154
- CONNX.REPLICATION.NOTIFY_CMD 154
- CONNX.REPLICATION.NOTIFY_CMD specifies 154
- CONNX.REPLICATION.STATUSREFRESHINT
ERVAL 154
- CONNX.REPLICATION.STATUSREFRESHINT
ERVAL controls 154
- CONNX/Replication 184
- CONNX_EP_ADA_CFG_PORT 154
- CONNX_EP_ADA_CFG_PORT specifies 154
- port..... 154
- CONNX_EP_DATA_PATH 154
- name..... 154
- CONNX_EP_DATA_PATH specifies 154
- CONNX_EP_DEBUG_LEVEL 154
- CONNX_EP_DEBUG_LEVEL controls..... 154
- CONNX_EP_NOTIFY 154
- CONNX_EP_NOTIFY_CMD 154
- CONNX_EP_NOTIFY_CMD specifies 154
- CONNX_EP_QUEUE_PORT..... 154
- CONNX_EP_QUEUE_PORT specifies 154
- CONN32/REPLICATION/COMMON/EVENTN
OTIFY.BSH..... 154
- CONN32/REPLICATION/COMMON/EVENTN
OTIFY.CMD 154
- CONN32/Replication/log..... 182, 219
- Consistency 3
- Contact Software AG 6
- Controller ..1, 6, 8, 10, 41, 87, 96, 103, 110, 145,
154, 168, 169, 182, 184, 219
- Check 184
- connect..... 184
- EP 184
- use..... 154
- Controller instructs 8
- EC 8
- Controller quiesces 8
- Controller user/process..... 184
- controller's..... 184
- copy 1, 3, 105, 112
- CPU 10
- Create1, 3, 8, 29, 41, 51, 53, 60, 64, 71, 87,
100, 102, 103, 110, 141, 149, 164, 184, 218
- create Controller 184
- EP..... 184
- CTRL..... 184
- customername,5,20 1
- D**
- Daemon 184
- Data 1, 3, 6, 8, 29, 41, 47, 51, 53, 60, 64, 66, 71,
77, 80, 87, 88, 93, 100, 102, 103, 105, 112,
141, 164, 184, 219
- Initial State duplicates 105
- Data Dictionary6, 8, 41, 87, 164, 219
- data dictionary containing 6
- Adabas 6
- data dictionary ex1.cdd 219
- data dictionary ex2.cdd 219
- Data Dictionary window41, 87, 88
- Open..... 87, 88
- data during replication 1
- Data Filtering..... 164
- data reporting..... 1
- Data Type51, 53
- Database.....1, 3, 6, 8, 29, 41, 93, 100, 103, 141,
184, 219
- open license 184
- Database ID29, 184
- database shutting..... 6

database's	3	Adjusting.....	150
databases containing	3	DLL	6, 184
database—usually	3	DNS	184
DB	184	Done .3, 8, 41, 51, 53, 60, 64, 66, 71, 87, 88, 93, 182	
DB ID.....	184	done deleting replications	87
DB Name.....	184	done troubleshooting	182
DB2	1, 6, 29	done viewing	88
debugging/status	154	don't want.....	3, 110
Default Access	29	Don'ts	3, 51, 64, 110, 219
default port number	24	Durability	3
Delete	3, 87	E	
Delete config	184	EC	6, 8, 184
Delete Rep	41, 87	Controller instructs	8
Delete Row.....	51, 60	EC Configuration.....	184
Deleting	1, 3, 60, 71, 87, 102, 184	EC reads	6, 8
Replication	87	MQ.....	8
Deploy	47, 51	Edit Menu.....	123
Deploy button	110	email	218
disable	110	Enabling	
Deployed Replications	1, 41, 143, 150	Event Replicator	41
Deployed Replications Tab	143	encounters	154, 184
Deploying . 1, 6, 8, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 87, 88, 93, 100, 102, 103, 105, 110, 112, 121, 141, 143, 145, 150, 165, 182, 184, 219		Nx	154
Event Replication.....	103	End Transaction.....	184
Deployment	102, 103, 184	Engine	1, 6, 8, 10, 19, 29, 164, 184
dequeue	184	Check	184
Initial State	184	Engine Adaptor	6
dequeue EP	184	engines depending	8
dequeue on Controller.....	184	enqueue	184
dequeue Server Status	184	Initial State.....	184
dereferenced	184	enqueue Engine.....	184
Different Controller	110	enqueue EP	184
DIR	184	enqueue init	184
Disable Process	184	Enter	8, 19, 41, 53, 93, 149, 184
Request	184	TCP/IP	19
DISABLED.....	184	EP	6, 8, 184
Disabling.....	110, 116	cache.....	184
Deploy button	110	connect.....	184
Event Replicator	116	Controller.....	184
Display.....	96, 112, 141, 143, 150, 182, 219	create Controller.....	184
		open Controller.....	184
		Replication Administrator	6

- ep_config.....154, 184
- ERR.....184
- Error Action184
- Error Action Code.....184
- Error Class184
- Error Code.....184
- Error State.....184
- Error State Code184
- ERRORS.....184
- ETL.....1
- Event Consumer 1, 6, 154, 184, 219
 - Check.....184
 - stopping6
- Event Producer ... 1, 6, 8, 10, 103, 154, 184, 219
- Event Producer user/process.....184
- Event Replication ... 1, 3, 6, 8, 19, 29, 41, 47, 60, 66, 71, 88, 103, 112, 116, 140, 154, 182, 184, 219
 - Deploying.....103
- Event Replication Administrator1, 8, 184
 - change8
- Event Replication Controller.....1, 41
- Event Replication Designer.....6
- Event Replication Designer requires.....6
- Event Replication Engine1
- Event Replication Error Messages.....184
- Event Replication Registry Settings.....154
- Event Replication Troubleshooting Do's219
- Event Replication Troubleshooting Suggestions182
- Event Replication using.....29
- Event Replicator1, 3, 6, 8, 19, 29, 41, 47, 51, 53, 60, 64, 71, 100, 102, 105, 112, 116, 140, 141, 166, 167, 168, 169, 184, 218, 219
 - Disabling.....116
 - Enabling.....41
- Event Replicator Administrator1, 8, 19, 29
 - Windows19
- Event Replicator Architecture6
- Event Replicator assumes8
- Event Replicator calls.....218
 - NOTIFY_CMD218
- Event Replicator Components19
 - Replicator Administrator.....19
- Event Replicator Error Message.....184
- Event Replicator license6
- Event Replicator needs.....29, 219
- Event Replicator User Exits218
- eventnotify.....154, 218
- eventnotify.cmd file154, 218
- eventserver168, 169
- ex1.cdd219
- ex2.cdd219
- Exec184
 - initialize.....184
- Existing1, 134, 184
 - config.....184
 - Server Status.....184
 - SQL Gateway1
 - Target.....134
- Expressions8, 149
- F**
- FATAL ERROR.....184
- FDT Import.....29
- File menu29, 41, 47, 87, 121
- Finish3
- Firewall.....24
- FIRST_NAME+MIDDLE_I+LAST_NAME.....149
- forget1
- FULL_NAME.....149
- G**
- GB.....10
- GB Free Space10
- GB RAM.....10
- General Features164
- General Messages.....184
- GHz.....10
- GUI.....1
- GUI-based.....1
- H**
- Help Menu.....140
- Hide Schema88
- higher/Suse 10/Linux 39010
- HPUX.....10
- HP-UX.....10

I

IBM AIX5L Version 5.2	10
ID	184
ID>_<engine ID	184
IDE	10
Import	1, 29, 100
Import CDD	29
Import CDD window	29
Import Type	29
Incompatible API	184
Index.....	51, 184
Informational	184
Informix.....	1, 6, 29
Initial State	3, 8, 29, 105, 107, 112, 143, 170, 184
dequeue.....	184
enqueue.....	184
Performing	105
Initial State duplicates	105
data.....	105
initial state processing.....	8, 184
Initial State replaces	105
Initial State.	3
initialize.....	184
Exec.....	184
initialize ep_config.....	184
Initiates	1
INSERT/SELECT SQL.....	8
Install.....	19, 103, 166, 182
Installation Directory.....	19, 166, 167, 168, 169, 184, 219
Installation Location	19
Installing	6, 19, 168, 182, 184
UNIX	19
INSTALLPATH	184
int.....	184
Internal API	184
Introduction	1
Invalid Server Status	184
Invalid user/pass	184
IO Error.....	184
IP.....	41, 93

IP address.....	41, 93
Isolation.....	3
It's	1
L	
L/N	184
L/N1	184
L/N2	184
L/N3	184
L/N4	184
L/Nx.....	184
LAN	10
leave	150
Server Status window	150
Length	53, 66, 71, 184
length exceeds.....	184
License Server	184
Linux	6, 10
LN/3	184
localhost.....	29
Log Files	96, 116, 154, 182, 184
See	182
logfile.....	184
Logical DB.....	184
login	93
logon	29, 41, 93
Logon Credentials.....	41, 93
M	
machine/PC	10
Manual	1, 96, 129, 150
Manual Column Width	129, 150
Map	1, 8, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 88, 134, 149, 165, 184
Map Columns.....	51, 53, 60, 64, 66, 71, 77, 80
Map Columns window.....	51, 53, 60, 64, 66, 71, 77, 80
map existing.....	184
map existing config	184
Mapping	51
columns	51
Mappings	53, 60, 66, 71, 77, 80
Mark	87, 103, 165
MAX ADABAS File.....	29

- MB RAM 10
- MB/sec 10
- Message ID 184
- Message Queue 1, 3, 6, 8, 93, 103, 166, 184
- Message Type 184
- MHz CPU 10
- Microsoft 112
- Microsoft Management Console 112
- mis-matched 184
- mode 1
- Modify 41, 47, 66, 71, 77, 80, 154, 184, 219
- MQ 8, 41, 93
 - EC reads 8
- mqserver 166, 167
- msg 184
- MU/PE column 184
- Multiple Data 219
- MySQL 1
- N
- N4 184
- NAME 154
 - CONNX_EP_DATA_PATH 154
- Name/Address 41, 93
- needs 1, 3, 29, 41, 51, 182, 219
 - today's 1
- NET 218
- NET 2003 218
- New 3, 6, 8, 29, 51, 53, 60, 64, 100, 102, 110, 184
 - Building 100
- new Replication 8, 184
- new Target Table 29, 51, 53, 60, 64
- Next 87
- non-Adabas 29, 53, 60, 149
- Non-MU/PE column 184
- non-Windows 29
- NORMAL 184
- Not Replicating 182
- NOTIFY 154
- NOTIFY_CMD 218
 - Event Replicator calls 218
- Nucleus Response 184
- NULL 184
- NULL source 184
- Number 154, 182, 184
- Nx 154, 184, 218
 - Action type 218
 - encounters 154
- O
- ODBC 1
- OK 29, 41, 47, 93
- OLEDB 1
- OLEDB Driver 1
 - Adabas 1
- OLEDB Provider 29
- online 3, 6
- Open 1, 3, 6, 8, 19, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 87, 88, 93, 96, 100, 102, 103, 105, 110, 112, 116, 121, 140, 141, 154, 182, 184, 218
 - Data Dictionary window 51, 88
- open Admin 184
- open Controller 184
 - EP 184
- Open Deployed CDD 121
- open EC 184
- open EP 184
- open EP Identity 184
- open existing 184
- open existing config 184
- open license 184
 - database 184
- open Replicator 184
- Open Systems .. 1, 3, 6, 8, 19, 29, 41, 47, 51, 60, 64, 66, 71, 88, 103, 140, 141, 154, 184, 218
- Open Systems Even Replicator 3
- Open Systems Event Replication window . 51, 88
- Open Systems Event Replicator 103
 - use 51
- Operating 10
 - System 10
- Operating System list 19
- Operating Systems 10, 19
- Options 1
- Oracle 1, 6, 29, 93

Oracle, SQL Server	1, 6	Red Hat	10
OS	184	Refer	184
OS Command	184	User Guide	184
OS-specific	184	Refresh	96, 154, 184
P		Refresh Server Status	184
PA RISC	10	Registry	154, 184
Password	19, 41, 93	Registry Variables	154
pause	170	Relational	1, 3
PC	219	Relational Database	1
performance statistics	154	Remove	3, 60, 87, 88, 110, 112, 184, 219
Performing	105	Rep	141
Initial State	105	repcmd	170
Performs	1	Replicating	96, 145
PID	184	Replication ..1, 3, 6, 8, 41, 47, 51, 53, 60, 64, 66,	
point	3	71, 77, 80, 87, 88, 93, 96, 100, 102, 103, 110,	
timethe	3	112, 116, 121, 123, 134, 141, 143, 145, 150,	
Port	8, 24, 41, 93, 154, 184	154, 164, 165, 168, 169, 182, 184, 219	
change		forget	1
CONNX.REPLICATION.ENGINE_PORT		Tables depending	134
.....	154	Replication Administrator1, 6, 8, 41, 143, 145,	
CONNX.REPLICATION.EP_ADA_CFG_POR		165, 182, 184	
T specifies	154	EP	6
CONNX_EP_ADA_CFG_PORT specifies ..	154	writeable	184
port number	24, 41, 93, 154, 184	Replication Administrator matches	184
all	93	Replication Controller41, 93, 103, 112, 154, 168,	
CONNX Message	154	169, 182, 219	
Postgres	1	Replication Design	87, 141, 143, 145
Precision	51, 53, 66, 71	Replication Design Tab	87, 141, 143, 145
preserve	3	Replication Designer	1
ACID	3	Replication Display	150
atomicity	3	Replication Engine	1, 154, 184
Press Deploy	8	Replication Log Directory	116
Primary Key	184	Replication Overviews	8
Problem Resolution	3	replication server ID	184
Programs	1, 19, 29, 41, 218	Replication Server Map	184
Q		Replication Servers...1, 8, 41, 93, 103, 112, 116,	
QUEUE	184	182, 184	
R		Replication Work Queue	184
Read Only	29	Replications Tab	143, 150
Ready	3, 29, 41, 165, 182	replications want	164
Recommended System	10	Replicator	3, 19, 184, 219
Recreating	102	connect	184
		Replicator Administrator	19, 219

- Event Replicator Components 19
- request 3, 103, 105, 154, 184
 - DISABLE PROCESS 184
 - respawn 184
- Required System 10
- Reserved 154
- respawn 184
 - request 184
- respawned 184
- Restart Servers 116, 138
- resume 170
- return 143, 145, 184
- RL 184
- RL Pool 184
- Row Down 53, 64
- Row Up 64
- S**
- Sample 154, 218
- Save 1, 29, 47, 51, 87, 112, 121, 184
- Save As 29
- Scale 51, 53, 66, 71
- SCSI 10
- Security menu 29
- See3, 47, 93, 100, 116, 129, 145, 164, 182, 184, 218, 219
 - Log Files 182
 - User Guide 184
- See OS 184
- Select ... 8, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 87, 88, 93, 102, 105, 110, 141, 143, 145, 149, 150
 - CDD 51, 88
 - Target Table 51
- select Auto Column Width 150
- Select Provider 29
- Select Provider Type 29
- Select Target 102, 105
- sendmail program 218
- Server... 1, 3, 6, 10, 19, 29, 41, 93, 96, 103, 112, 116, 138, 145, 150, 154, 182, 184
 - server name 29, 41, 93, 184
- Server Status . 96, 112, 116, 145, 150, 154, 182, 184
 - existing 184
 - show 150
 - Viewing 96
- Server Status Tab 145
- Server Status window 150
 - leave 150
- Servers Menu 138
- Services 1, 110, 112, 182, 184, 219
- set 1, 29, 41, 93, 123, 150, 184, 218
- Set Port 93
- Settings 29, 154
- show 3, 29, 88, 145, 150, 182
 - Server Status 150
- Show Schema 88
- Show/Hide Schema 129
- Showing 88
 - Source Table Schema 88
- Shutdown 184
- Solaris 10
- Solutions 3, 19, 29, 41
- source 1
 - Source... 1, 3, 6, 8, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 87, 88, 93, 100, 102, 103, 105, 112, 116, 141, 143, 149, 184, 219
 - change 51
 - Source Column 8, 51, 53, 66, 71, 77, 80, 149, 184
 - Source Column Mapping 66, 77
 - source column name 66, 149
 - Source Database 1, 3, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 88, 93, 100, 102, 103, 105, 112, 116, 141, 184
 - add 93
 - set 103
 - Source Database Servers 93
 - source database starts 3
 - source DBID/target database 8
 - Source Table 1, 29, 41, 47, 51, 53, 60, 64, 71, 88, 105, 141, 184
 - source table metadata 51, 53
 - Source Table Schema 88
 - Showing 88
- Spawned ep_config 184

SQL	1, 6, 8, 10, 19, 29, 51, 53, 149, 164, 184
SQL Expressions	53
SQL Gateway	1, 6, 8
SQL Gateway Adaptor	6
SQL Gateway Target Adaptor	6, 8
SQL Server	1, 29
Connecting	29
SQL Server Integration Services	1
SQL View Clause	8, 164
SQL View Clause Text Box	164
SQL-based	1
SqlRegistry tool	184
SSIS	1
start	3, 6, 8, 19, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 88, 93, 100, 102, 103, 105, 116, 166, 168, 182, 184
UNIX	166, 168
UNIX Message Queue	166
Start menu	19, 29, 41, 88
start replicating data ...	41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 88, 93, 100, 102, 105
state	6, 8
State Description	96
StateDescription column	96, 145
states	6
statistics	154
Status ..	3, 96, 103, 112, 116, 145, 150, 154, 170, 182, 184
Status Message	184
Status Received	96, 145
Status Tab	112, 116, 145, 150, 184
Stop Server	184
Stop Targets	112, 138, 184
stopping	6, 167, 169
Event Consumer	6
UNIX	167, 169
UNIX Message Queue	167
substring	1
SUN Solaris	10
support	184
Support DLL	184
Suppressing an Initial State	107
Sybase	1, 6, 29
synchronization	1
System Requirements	10
System Unknown	96, 145, 182
Systems 1, 3, 6, 8, 10, 19, 29, 41, 47, 51, 60, 64, 66, 71, 88, 96, 140, 141, 145, 154, 182, 184, 218	
Operating	10
Systems Even Replicator	3
T	
table	1, 8, 29, 41, 47, 51, 53, 64, 88, 93, 100, 102, 105, 112, 134, 141, 143, 164, 184, 219
Adding	47
table ID	184
Tables depending	134
replication	134
Tables Menu	134
target	1
Target	1, 3, 6, 8, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 87, 88, 93, 100, 102, 103, 105, 110, 112, 116, 134, 141, 143, 149, 165, 184, 219
existing	134
target account	3
Target Adaptor	6, 8
Target Column	8, 51, 53, 66, 71, 77, 80, 149
target column mapping	8, 71, 80
target column name	53, 71
Target Data Type	51
target database needed	29
Target Databases ...	1, 3, 6, 8, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 88, 93, 100, 102, 105, 110, 112, 116, 141, 219
Target DB	184
connect	184
Target Length	51
Target Table	1, 3, 6, 41, 51, 53, 60, 64, 66, 71, 77, 80, 87, 100, 102, 141, 165, 184, 219
create	41
Target Table contain	141
target table ID	184
target table metadata	51, 53
Target tables	29, 47, 51, 165
TBL	184
TCP/IP	19, 184, 219

- Enter 19
- tcpip port number 24
- Technical Support 182, 184
- terminates 184
- Test Connection 41, 93
- Time column 96
- Timeout 184
- Timeout sending 184
- Timeout waiting 184
- timethe 3
 - point 3
- today's 1
 - needs 1
- transaction-oriented 3
- TX 184
- TXID 184
- U**
- UnDeploy 143
- Un-Deploy all 110
- UnDeploy All 143
- Un-Deploying 110
- Under Populate columns 66, 71, 77, 80
- Uninitialized 184
- UNIX 6, 19, 29, 166, 167, 168, 169, 182
 - Installing 19
 - start 166, 168
 - stop 167
 - Stopping 169
- UNIX Installer 19
- UNIX Machine 19, 166, 168
- UNIX Message Queue 166, 167
 - Starting 166
 - Stopping 167
- Unix servers 6
- UNIX Setup window 19
- Unix/Linux 154, 184
- Unix/Linux Environment Variables 154
- Unknown 182, 184
- Unknown EP 184
- Unrecognized ADABAS table 184
- Unsupported backend 184
- use 1, 3, 6, 8, 10, 29, 41, 51, 53, 60, 64, 71, 93, 110, 154, 164, 184
 - Controller 154
 - Open Systems Event Replicator 51
- use localhost 29
- Use Row Up 53
- User Guide 29, 140, 184
 - Refer 184
 - See 184
- User ID 19, 184
- User Name 41, 93
- Uses ConnecX 1
- Uses SQL Gateway 1
- Using 1, 6, 8, 29, 41, 51, 53, 64, 103, 141, 164, 219
- V**
- Valid CONNX.REPLICATION.ENGINE_PORT 154
- Valid
 - CONNX.REPLICATION.STATUSREFRESHINTERVAL 154
- Validate 47, 165, 184
 - Active Servers 165
- Validate Active 47, 165
- Validation 165
- Validation fails 165
- validation succeeds 165
- Values L/N1 184
- Values N1 184
- VB 218
- View Clause Text Box 164
- View Menu 88, 129, 150
- Viewing 96
 - Server Status 96
- W**
- WAN 10
- WARNINGS 184
- Why 3
- Width 150
- window containing 29
- Windows 6, 10, 19, 29, 41, 47, 51, 53, 60, 64, 66, 71, 77, 80, 88, 93, 103, 140, 150, 184
 - Event Replicator Administrator 19

CONNX InstantdbSync

Windows 2000/2003/XP	10	writeable.....	184
Windows 2000/2003/XP/Vista.....	10	Admin	184
Windows Environment	6	Replication Administrator	184
Windows NT	10	Y	
Windows NT Advanced Server	10	Yes.....	60, 87, 110
Working	93		
