Adabas Open Systems Event Replicator

Table Of Contents

Chapter 1 - Preface 1
Introduction 1
The Business Case 1
The Options 1
The What, Why, and How of Event Replication 3
What is live data event replication? 3
Why should we want to perform live data event replication? 3
Is live data event replication always the best solution? 3
How does Open Systems Event Replication perform live data event replication? 3
Open Systems Event Replicator Architecture - Replicating to a Relational Database 6
Open Systems Event Replicator Architecture - Replicating Adabas to Adabas 8
Replication Overviews 10
Replication Overview from the "front end" perspective 10
Replication Overview from the "back end" perspective 10
Chapter 2 - Event Replicator Installation 11
System Requirements 11
Windows 11
Linux 11
Solaris 11
HPUX 11
AIX 12
Prerequisites 13
Installing the Event Replicator on a Windows Environment 14
Installing UNIX Event Replicator Components on the Replicator Administrator 17
Installing the Event Replicator on a UNIX Environment 19
Instalation considerations in an Adabas to Adabas environment 23
Installing the JMS Server 25
Default Port Numbers and Firewall Considerations 28
Replication Components 31
Chapter 3 - Event Replicator Removal 36
Uninstalling the Event Replicator 36
Chapter 4 - Using the Event Replicator - Adabas to Relational 37
Creating a CDD for Event Replication using the CONNX SQL Engine 37
Importing the Adabas Source Tables 39
Importing the Target Database 41
Set the CDD Security 43
Enabling a CDD for Event Replication 45
JMS Queues/Topics as a Replication Target 51
Starting the CONNX JMS Server 51

JMS Data Format (XML format) 55
Delete (Puts where Criteria in order to identify record) 55
Insert (Shows all fields that were inserted) 55
Update (Changed CITY field to Kirkland. It shows all the fields for the updated record and contains where Criteria in order to identify the record) 56
Create (This is for CREATE TABLE. It displays the meta data for the table) 57
Drop (This is for DROP TABLE.) 57
Adding a Replication to the Event Replicator 58
Mapping columns to a new Target Table 62
Adding New Column Mappings 64
Deleting Column Mappings from a New Target Table 69
Changing the Column Order in a New Target Table 72
Changing Source Column Mapping 74
Changing Target Column Mapping 77
Clearing Source Column Mapping 81
Clearing Target Column Mapping 83
Importing and Exporting Column Mappings 85
Deleting a Replication 89
Viewing the Source Table Schema 91
Configuring Replication and Source Database Servers 95
Viewing Server Status 98
Building New Target Database Tables 101
Dropping and Recreating Target Database Tables 103
Deploying the Event Replication 105
Performing an Initial State 107
Suppressing an Initial State 109
Un-Deploying all Replications 112
Un-Deploying selected Replications 115
Stopping Event Replication 118
Restarting Event Replication 121
Starting, Stopping and Getting Status from the Event Server 124
Event Replicator Tutorials 125
About Event Replicator Tutorials 125
Performing a Simple Replication - Adabas to Relational Database 126
Create a CDD containing the source and target databases 126
Importing the Adabas Source Tables 127
Importing the Target Database 131
Set the CDD Security 134
Create a Replication 135
Adding a new Replication and Redeploying 144
Open the Event Replicator CDD and add the source table to the target database 144
Redeploy the active Replications in the CDD 148

Validate the deployment 150
Removing a Replication from Deployment 152
Open the Event Replicator CDD and un-deploy a replication 152
Validate the un-deploy 154
Manipulating Replication Data 156
Filtering Replication Data 156
Concatenating Multiple Data Columns 165
Excluding Data Columns 168
Performing Math Operations on Replication Data 174
Using SQL Expressions to Split a Column 178
Renaming a Target Column 181
Chapter 5 - Using the Event Replicator - Adabas to Adabas 185
Preparing an Adabas database for Adabas to Adabas Replication 185
Creating a CDD for Adabas to Adabas Event Replication using the CONNX SQL Engine 186
Adding the Source and Target Database connections 188
Set the CDD Security 190
Enabling a CDD for Adabas to Adabas Event Replication 192
Adding an Adabas to Adabas File Copy Replication to the Event Replicator 197
Renumbering Target Files 201
Deleting a Replication 206
Configuring Replication and Source Database Servers 208
Viewing Server Status 211
Deploying the Event Replication 214
Performing an Initial State 216
Un-Deploying all Replications 218
Un-Deploying selected Replications 221
Stopping Event Replication 224
Restarting Event Replication 227
Starting, Stopping and Getting Status from the Event Server 230
Using adaopr command line Utility 231
Chapter 6 - Event Replicator Menus and Tabs 236
File Menu 236
Edit Menu 237
View Menu 241
Tables Menu 246
Servers Menu 248
Help Menu 250
Replication Design Tab 251
Deployed Replications Tab 253
Server Status Tab 255
Chapter 7 - Event Replicator Functions 259
Open the Event Replication CDD 259

Mapping SQL Expressions to Target Column....262
Adjusting Display Column Width....263
Event Replication Registry Settings....267
Registry Variables....267
Unix/Linux Environment Variables....267
A2A specific Unix/Linux Environment Variables....267
Registry Variables....268
Unix/Linux Environment Variables....274
A2A specific Unix/Linux Environment Variables....275
Data Filtering....277
Validating Active Replications....278
Switching from Test to Production....279
Changing Source Tables....283
Starting the UNIX Message Queue....287
Stopping the UNIX Message Queue....288
Starting the UNIX Replication Controller....289
Stopping the UNIX Replication Controller....290
Chapter 8 - Event Replicator Troubleshooting Guide....291
Event Replication Troubleshooting Suggestions....291
Server Status....291
Log Files....292
Event Replication Error Messages....293
General Messages....294
Admin....296
Controller....299
Event Consumer....309
Event Producer....316
Debug....323
Event Replicator User Exits....325
Event Replication Troubleshooting Do's and Don'ts....326
Do's....326
Don'ts....326
Appendix A - Technical Support....327
Support Contacts....327
Appendix B - Copyright Page....328
Trademarks and Copyright....328
Index....337

Chapter 1 - Preface

Introduction

The CONNX Open Systems Event Replicator is a real-time data replication tool that propagates changes from a source database to a target database. With the Open Systems Event Replicator, events that affect data in the source table, such as updates, inserts and deletes, will automatically be propagated to the target table in real time.

The Open Systems Event Replicator has an easy-to-use graphical interface that deploys installation and configuration information to the source and target database servers.

The Business Case

Because it can store and process multi-value fields, handle large volumes of data, and manage high-speed transactions, Adabas is ideally suited to house the critical business data needs of today's fast-paced companies. With Adabas, applications are able to save and retrieve complex data structures in a single physical record and in a single operation. As a multi-value database, Adabas has significant transactional performance advantages over relational databases, which would require several joins between many tables to accomplish the same task.

Although multi-value Adabas excels in transactional performance, relational databases excel in analytical capability. Relational databases also have a large selection of analytical and reporting tools available for companies to use. The Open Systems Event Replicator for Adabas give companies the best of both worlds: superior multi-value database performance and SQL-based tools for data reporting and analysis.

The Options

There are several ways to move Adabas data into a relational 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 Adabas system to the relational 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 relational 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 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 Adabas (Adabas ConnecX SQL Engine, it only takes a few clicks to move a snapshot of Adabas data into 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 changed 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). The Open Systems Event Replicator can replicate data from one Adabas database to another Adabas database for backup or disaster recovery purposes. It can also replicate Adabas data to relational 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 Adabas target or the target relational 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 in Adabas. This allows transactional applications to continue to benefit from Adabas high-speed performance and opens the door for using SQL-based data analysis and reporting tools against the relational 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 on-demand environment.
- Capture changes 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. We 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 Open Systems Event Replication 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. In the Open Systems Event Replicator, 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. In the Open Systems Event Replicator, 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.

  In the Open Systems Event Replicator, if the transaction in the target database does not succeed, it retain the transaction on the message queue and generate a problem notification message. 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. The Open Systems Event Replicator 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**

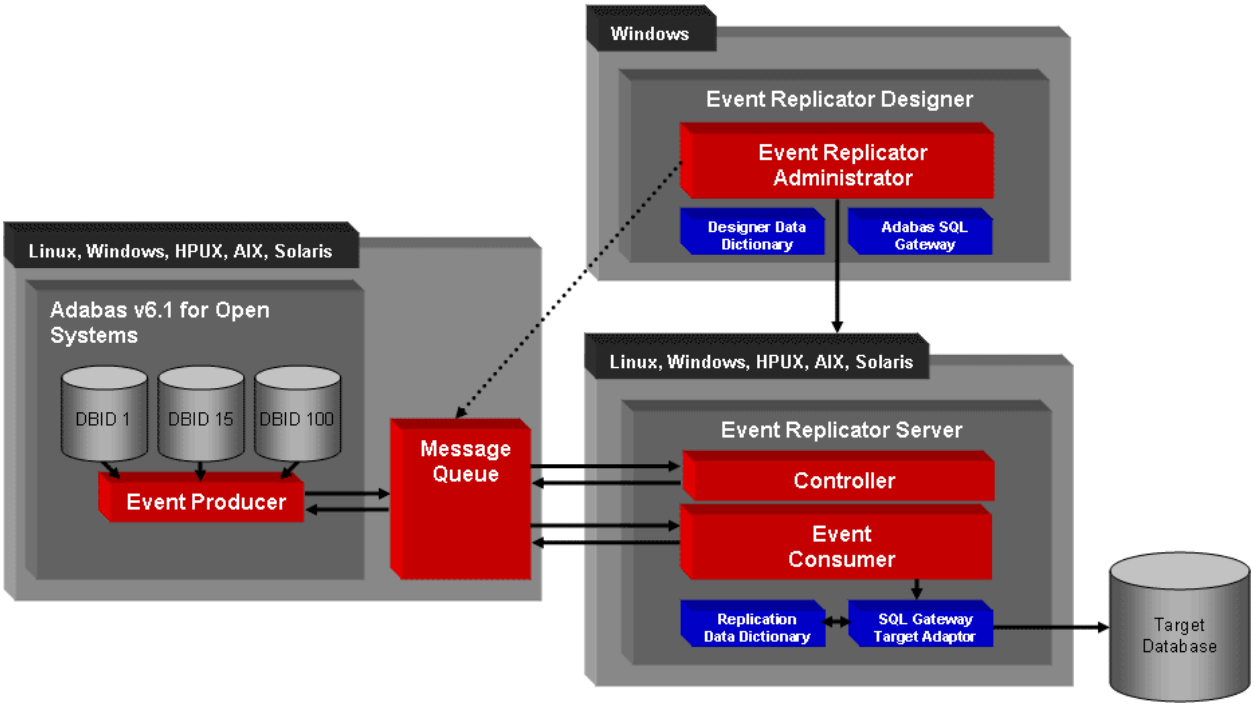
Because the changes have been stored once the transaction completes, we know that the changes are permanent and cannot be lost due to distressing events such as power outage. However complications can occur in the live data event replication process. Two possible problems are:

  - 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 damage can occur but the replication system won't be aware of the problem but the system users will be. For example, data may be accidentally deleted from a target table. Since the Open Systems Even Replicator only monitors changes on the source database, but not on the target database, it will not be aware of this damage. 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.

Open Systems Event Replicator Architecture - Replicating to a Relational Database

This diagram illustrates the Open Systems Event Replicator architecture.



In the diagram, the Event Replicator components are spread across three different servers, the recommended configuration. You can also install all components on a single Windows server (because the Replication Administrator must reside on Windows), or on two servers (the Replication Administrator must reside on Windows, the other components can reside on Windows, Linux or Unix servers).

The Open Systems Event Replicator consists of the following components:

1. Adabas v6.1 (or above) for Open Systems

   The Event Replicator real-time change data capture capability is only available in Adabas 6.1 and above for Open Systems and requires a special Adabas replication license. Contact Software AG for more information about obtaining the Open Systems Event Replicator license.

2. Replication Administrator

   The Event Replicator graphical user interface is written in C#.NET and can only run in a Windows environment. Using the Replication Administrator, you can define new replications in less than a minute.

3. Event Producer (EP)

   The Event Producer is a DLL (shared library) that is loaded by the Adabas nucleus. Every time a change is made to an Adabas database, the EP is notified of the change in real time. Transactions are grouped together and placed on the Message Queue, to be picked up by the Event Consumer.

4. 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. Until the components are available, the EP persistently stores Adabas transactions that affect data content in the message queue for future retrieval.

5. Controller

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

6. 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 ConnecX SQL Engine Adaptor, recreates the transaction on the target database.

7. ConnecX SQL Engine

   The EC uses the ConnecX SQL Engine to perform "initial states." An "initial state" moves all the data from a source Adabas file to a target table for the first time. Initial states may also be automatically initiated by the Event Replicator if an unrecoverable error occurs such as the Adabas database shutting down improperly.

8. Designer Data Dictionary

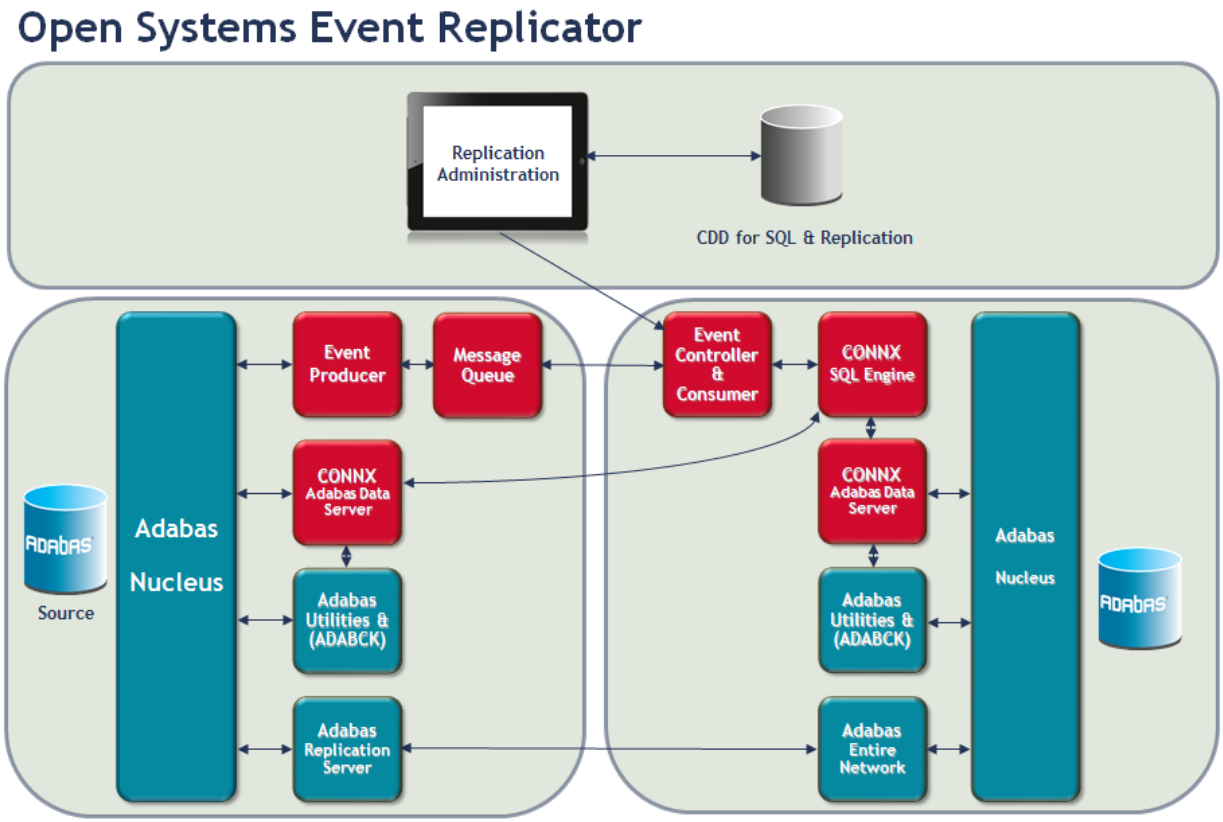
   The ConnecX SQL Engine Event Replication Designer requires a data dictionary containing the Adabas files' SQL based definitions.

9. ConnecX SQL Engine Target Adaptor

   In addition to the ConnecX SQL Engine , the Event Replicator requires adaptors for the target databases. The ConnecX SQL Engine has target adaptors for Oracle, SQL Server, DB2, Sybase, and Informix.

Open Systems Event Replicator Architecture - Replicating Adabas to Adabas

This diagram illustrates the Open Systems Event Replicator architecture when doing Adabas to Adabas replication..



This diagram shows the Open Systems Event Replicator in an Adabas to Adabas replication environment.

In this environment, the data is replicated directly from the source Adabas nucleus to the target Adabas nucleus. The following components are required:

1. Adabas v6.3.1 (or above) for Open Systems

   Adabas to Adabas replication requires Adabas version 6.3.1 and above. An A2A license is required for both the nucleus as well as the Open Systems Event Replicator to use this feature. Once replications are deployed and the initial state process has completed, data is replicated from the source nucleus directly to the target nucleus. The replication server is integrated into the nucleus and is only shown separately in the above diagram for illustrative purposes.

2. Entire Net-work

   The Adabas to Adabas Event Replicator uses standard Adabas calls to transfer data between the source and target nucleii, so if the target database is on a physically different machine, Entire Net-work is required.

3. Replication Administrator

The Event Replicator graphical user interface is written in C#.NET and can only run in a Windows environment. Using the Replication Administrator, you can define new replications in less than a minute.

3. Event Producer (EP)

   The Event Producer is a DLL (shared library) that is loaded by the Adabas nucleus. This is the component that communicates with the nucleus to tell it which files to replicate as well as to get status on current replications. Status information is sent to the Replication Administrator where it is displayed.

4. 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. In an Adabas to Adabas configuration, this component is used for passing communication messages between the EP, the controller and the Replication Administrator.

5. Controller

   The Controller manages communicating deploy requests as well as status between the EP and the Replication Administrator. It also handles calling the Adabas Utility ADABCK to perform the initial state process..

6. CONNX SQL Engine

   The Controller uses the CONNX SQL Engine to perform "initial states." An "initial state" moves all the data from a source Adabas file to a target table for the first time. The CONNX SQL Engine uses the Adabas utility ADABCK to extract the data from the source database as well as to load it in the target database. Data is extracted from the source database by ADABCK where it is passed to the CONNX Data Server running on that system. The Data Server then passes the data via TCP/IP to the CONNX SQL Engine which in turn sends it via another TCP/IP connection to a second CONNX Data Server running on the target system. From there the data is passed to ADABCK where it is loaded into the target database. When replicating from Adabas to Adabas, the CONNX SQL Engine is not used to convert the data to SQL as it is when replicating from Adabas to a Relational database. Instead, the data is transferred directly between the source Adabas nucleus and the target Adabas nucleus in the native format.

7. Designer Data Dictionary

   The Event Replication Designer requires a data dictionary containing the Adabas nucleus' connection information.

Note: When doing Adabas to Adabas replication, an Adabas to Adabas license must be present for the Adabas nucleus as well as an Adabas to Adabas license for the Open Systems Event Replicator. For further information on how to install and configure Adabas for Adabas to Adabas replication please refer tot he appropriate Adabas documentation.

Replication Overviews

Replication Overview from the "front end" perspective

To replicate data using the Open Systems Event Replicator:

1. Verify that you already have a CONNX Data Dictionary (CDD) containing the Adabas file definitions for the source files.
2. Add a link to the desired target database to the data dictionary.
3. Start the Event 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 (EC), and specify any port overrides for communicating to the message queue.
4. Select the source tables to replicate, and the target database.
   - The Event Replicator assumes a one-for-one column mapping between the source and the target tables. Use the Event 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 ConnecX SQL Engine. Using the Data Dictionary manager, a SQL based filter can be applied to any table in the "SQL View Clause" field.
6. Press Deploy, and real-time event replication begins.

Replication Overview from the "back end" perspective

When you deploy a replication:

1. The Replication Administrator uses the "front end" information to create a new replication CDD and sends it to the Controller.
2. The Controller quiesces 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 EC engines depending on the work to be done. An EC is started for every source DBID/target database combination in the new replication CDD.
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 the ConnecX SQL Engine, and inserting it into the target database using a standard INSERT/SELECT SQL statement. During initial state processing, changes to the source Adabas file are kept in the Message Queue for future playback after initial state processing is complete.
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 the ConnecX SQL Engine Target Adaptor to place changes in the target database. This continues in real time until replication is stopped.

Chapter 2 - Event Replicator Installation

System Requirements

Windows

Required System	Recommended System
Operating System	
64bit Windows Server class operating system	64bit Windows Server class operating system
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
IDE or SCSI disk drive controller	Raid controller or mirrored drive controller
10 MB/sec network card or WAN connection	1 GB network card or LAN connection
Software	
CONNX SQL Engine Adabas 6.1.4 or higher with Relational target, Adabas 6.3 SP1 or higher with Adabas to Adabas replication	CONNX SQL Engine Adabas 6.1.4 or higher with Relational target, Adabas 6.3 SP1 or higher with Adabas to Adabas replication

* The operating system requirements listed below are for the Open Systems Even Replicator. Adabas 6.3 SP1 may have requirements for version numbers higher than those listed below. Please see the Adabas documentation for information about Adabas system requirements.

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

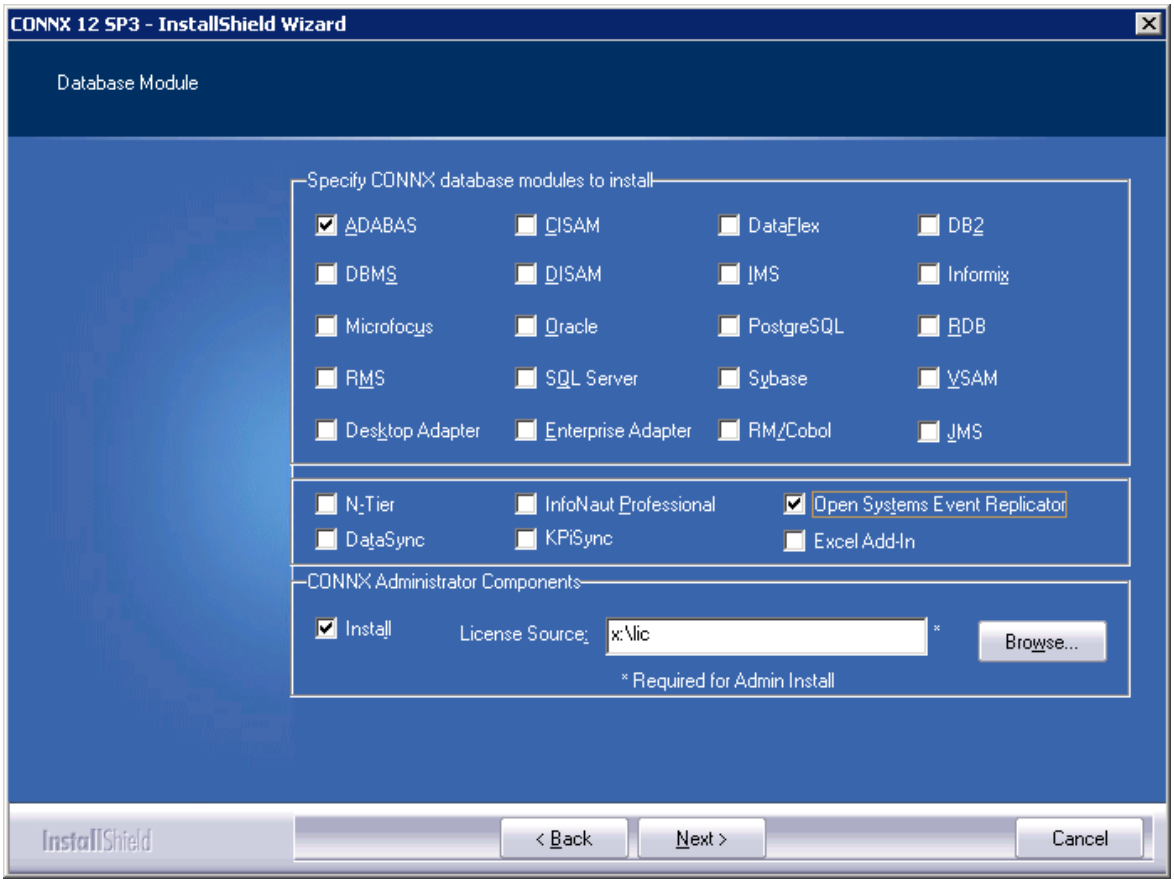
Prerequisites

Before you install the Open Systems Event Replicator:

- Upgrade to Adabas version 6.3 SP1 or later for Adabas to Adabas replication or 6.1.4 or later for Adabas to Relational replication.
- If replicating to a JMS target, version 1.7 or higher of the Java JRE must be installed on the machine where the CONNX JMS Server is installed.
- Install the server portion of CONNX SQL Engine on the Event Replicator source server.
- Install the client portion of CONNX SQL Engine on all Event Replication Controller and Event Replication Administrator servers.
- Create a single CONNX Data Dictionary username and password for all databases.
- Give all Event Replicator users administrator-level privileges for the target data bases.
- Importing prerequisites:
    - Import all the source and target databases into a single CONNX Data Dictionary (CDD).
        - For SQL Server, Oracle, Sybase or Informix target databases on UNIX, use OLEDB to import the target database into the CDD.
    - For Oracle target databases on UNIX, the system running the Event Replicator Controller must have the same entries in tnsnames.ora as the system used to import the Oracle database into the CDD.
- License prerequisites:
    - A CONNX license file for each database.
    - A CONNX Adabas to Relational license as well as a Software AG ADAREX license for doing replication with a relational database as the target
    - A CONNX Adabas to Adabas license as well as a Software AG ADAREX and A2A license for doing Adabas to Adabas replication.

Installing the Event Replicator on a Windows Environment

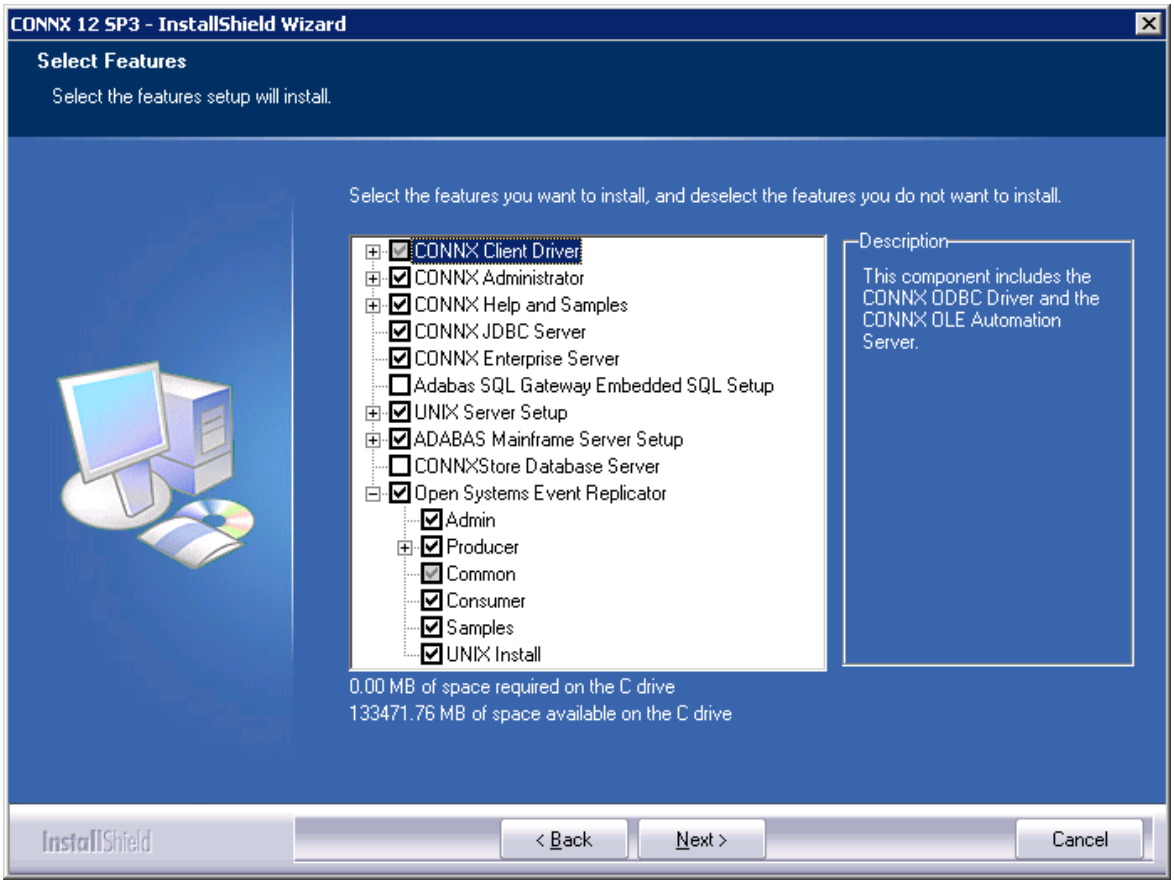
1. The Open Source Event Replicator is installed as part of the main CONNX installation package. On the Database Module screen, select Open Systems Event Replicator



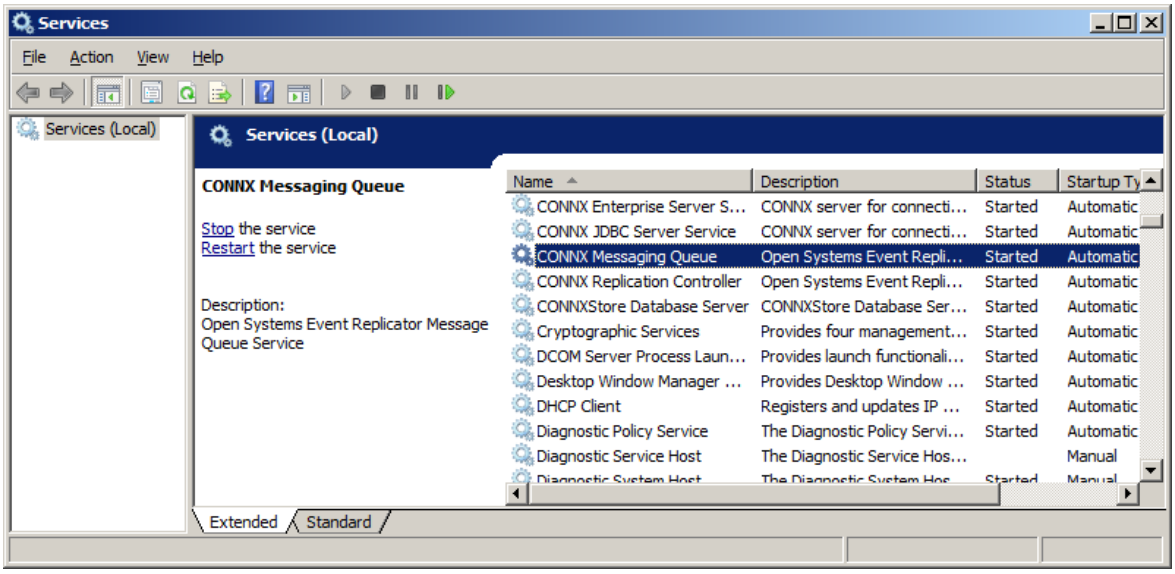
2. The **Select Features** window appears. You can select the Event Replicator components you wish to install.

Note: The Event Producer must be installed on the same server as Adabas; the other components can be installed on different systems.

Note: The Event Replicator Administrator (**Admin**) must be installed on a Windows environment even if the Event Producer or the Controller/Event Consumer are installed on UNIX environment.



3. After the installation process completes, restart your machine if asked to do so.
4. Click the **Start** button, point to **Settings**, and point to **Control Panel**. Select **Administrative Tools** and then **Services**. The CONNX Messaging Queue and CONNX Replication Controller services should be in the Local Services list and their Server Status should be **Started**.



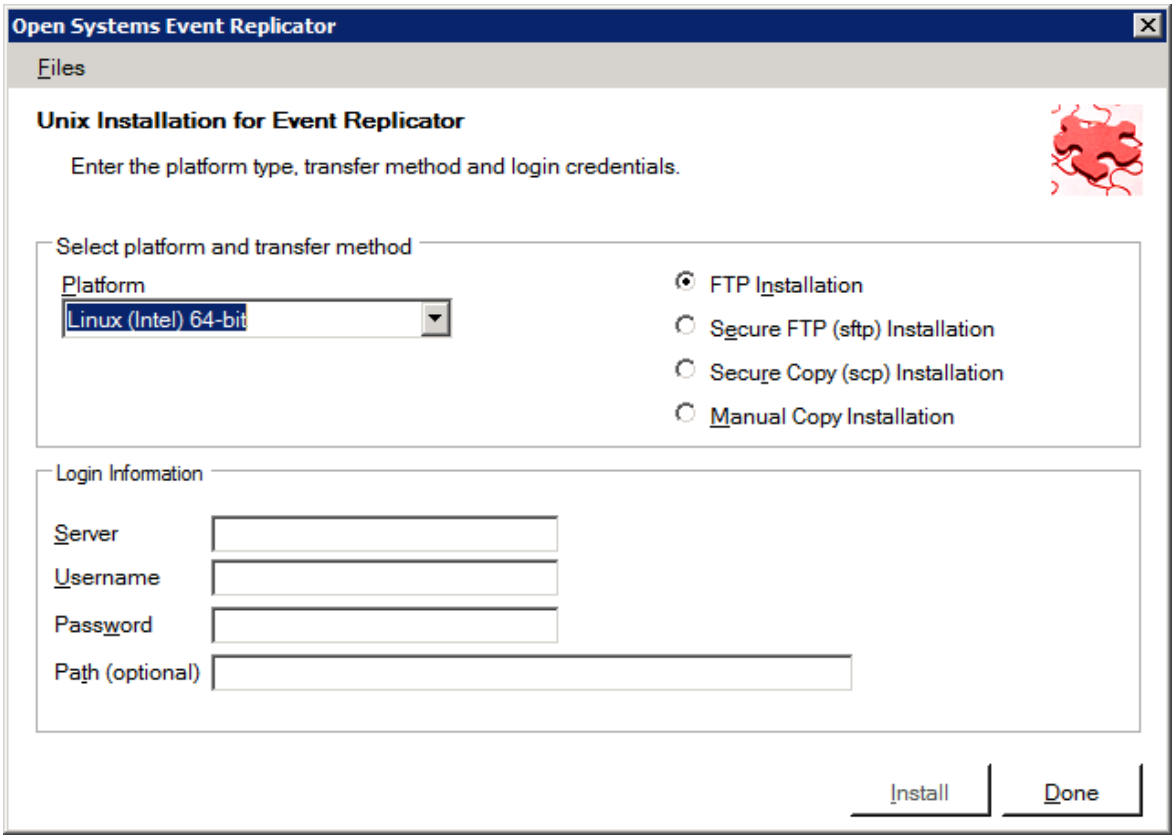
5. If the Event Producer or the Controller/Event Consumer are on UNIX environments, install those components.
6. If the event producer and controller were installed on different machines, start the message queue on both machines (Windows or UNIX) before deploying a replication. Note: If there is a firewall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

Installing UNIX Event Replicator Components on the Replicator Administrator

In order for the Event Replicator to run on a UNIX system, install components on both the Windows Event Replicator Administrator system and the UNIX system.

Note: Before installing the Event Replicator on UNIX, install the Event Replicator Administrator on a Windows machine and the CONNX Adabas SQL Gateway client component on the UNIX machine.

1. On the **Start** menu, click **Programs**, click **CONNX Open Systems Event Replicator** and then click **UNIX Installer**. The **Open Systems Event Replication for UNIX Setup** window appears.

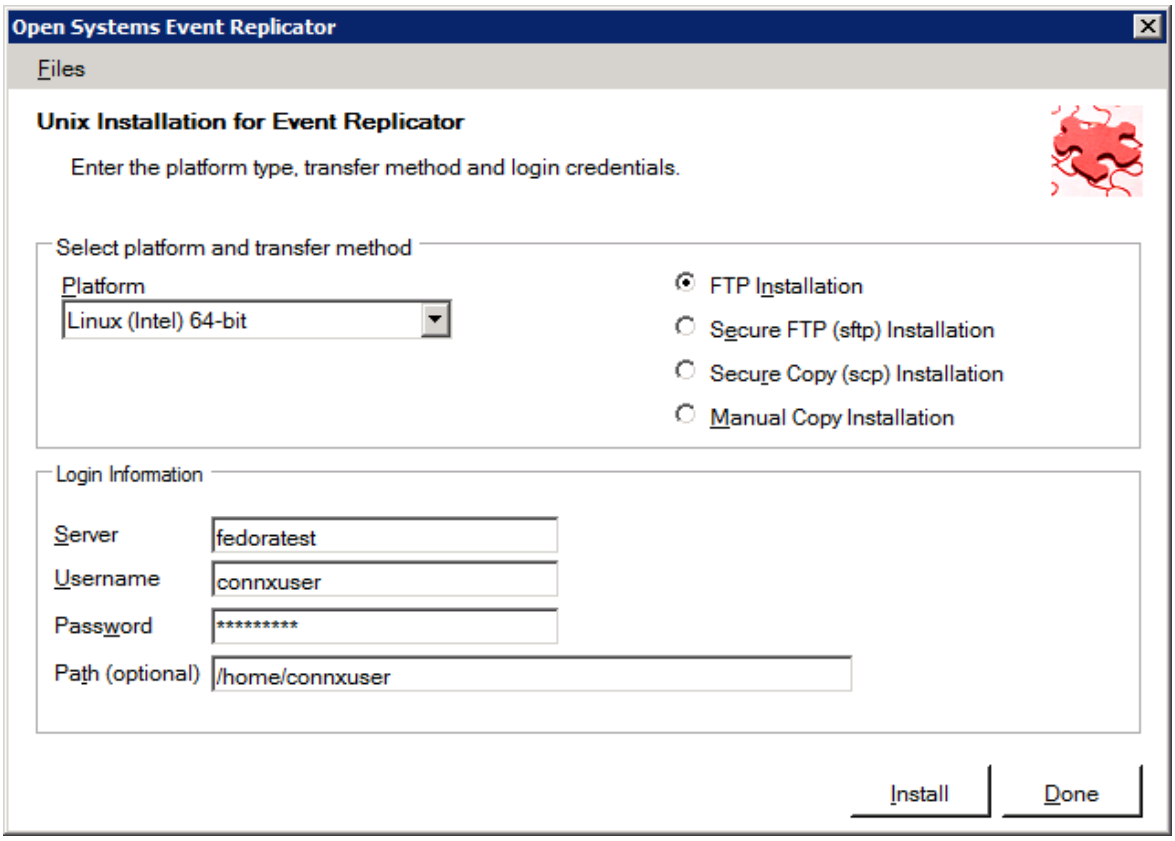


2. If the target system has an FTP, SFTP or SCP server enabled, skip to step 4.

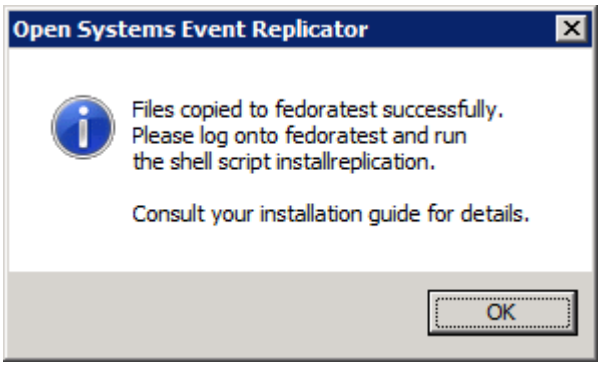
3. If the target system does not have an FTP server enabled, select the Manual Copy Installation option and click the Begin Client Installation button. This option will create the necessary install files in the REPLICATION\INSTALL\UNIX\TEMPINST subdirectory of the CONNX installation directory. Move these files via an alternate copy method to the Unix server and then proceed to the section Installing the Event Repicator on a Unix Environment.

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**.



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



Installing the Event Replicator on a UNIX Environment

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

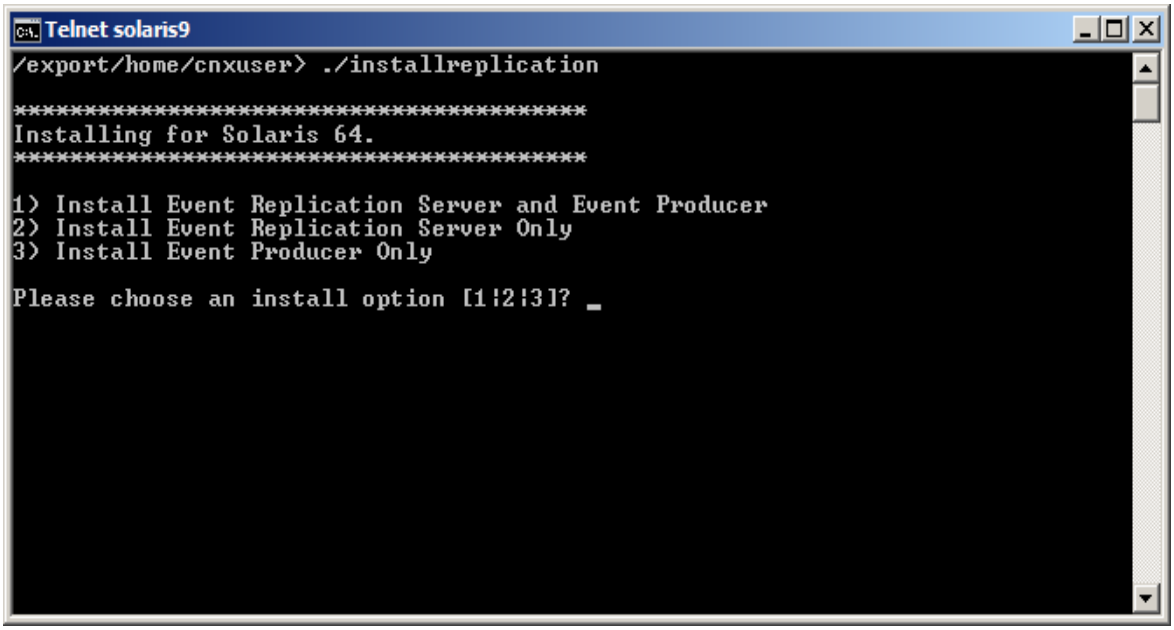
- install the Event Replicator Administrator on a Windows machine.
- install the Event Replicator components on both the Windows Event Replicator Administrator system and the UNIX system.
- install the ConnecX SQL Engine component on the UNIX machine.

We strongly recommend using the same account to install both Adabas and the Event Replicator.

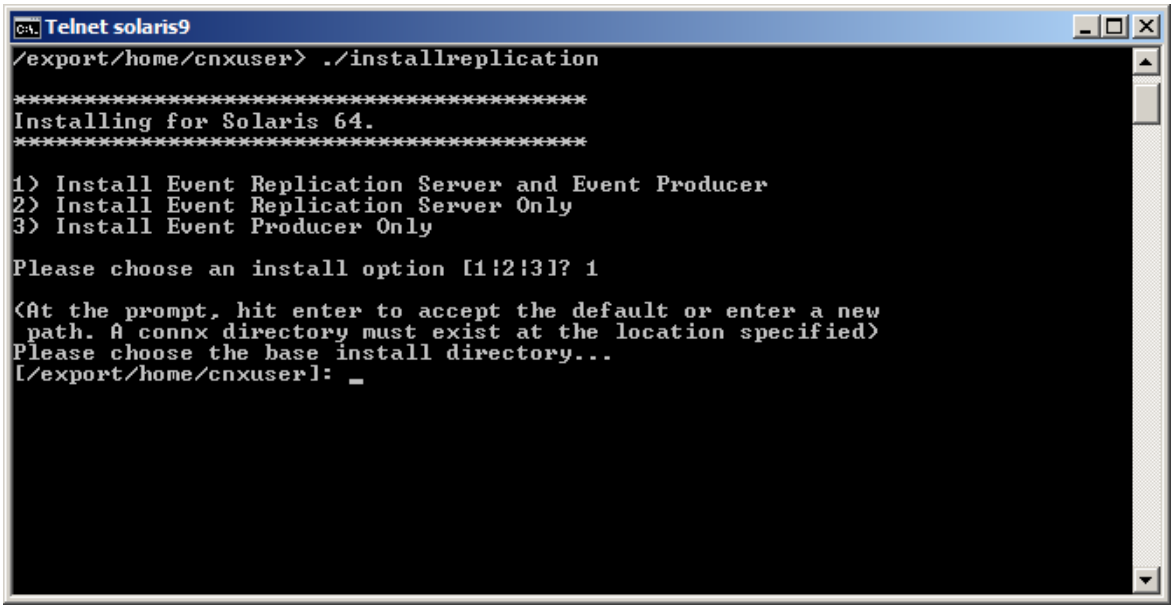
1. On the Windows Event Replicator Administrator system, run the Unix Installer from the CONNX Open Source Event Replicator Start menu.
2. Open a session in the UNIX installation directory on the UNIX Event Replication machine.
3. Run the install script ./installreplication.
4. Enter an install option:
    - If this is the Adabas source database server, install the Event Producer.
    - If this is the Event Consumer and the Controller server, install the Event Replication Server.

- Option 1 will install both the Event Replication Server and the Event Producer,
- Option 2 will only install the Event Replication Server, and
- Option 3 will only install the Event Producer.

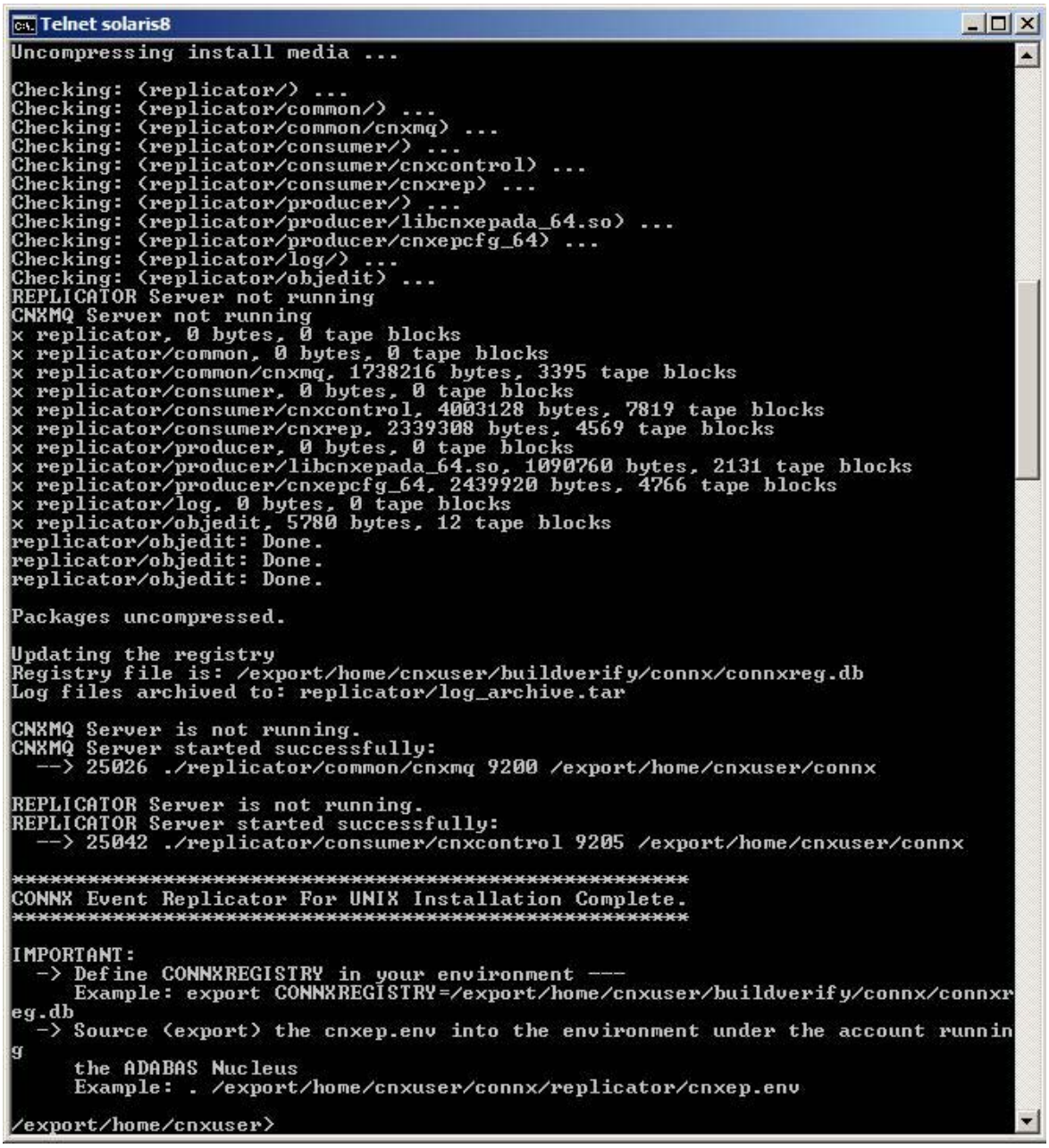
Note: The Event Replication Server contains the Event Consumer and the Controller.



5. Press **Enter**. You will be prompted for an installation directory.



6. Select the same install directory as the one containing the ConnecX SQL Engine installation. Press **Enter**. The install script will install the selected components and start the servers.



7. After the installation completes:
   - Define CONNX_EP_DATA_PATH in your UNIX environment.

     CONNX_EP_DATA_PATH is the location of the Event Producer log file.
     - CONNX_EP_DATA_PATH can be in the same directory as ConnecX SQL Engine.
     - The account used to run Adabas must have read/write access to CONNX_EP_DATA_PATH
   - Define CONNXREGISTRY in your UNIX environment.
   - Define ADAREX in your UNIX environment.
   - Using the account running the ADABAS Nucleus, source (export) cnxep.env into the environment (for example, /export/home/cnxuser/connx/replicator/cnxep.env).
   - If necessary, start the message queue and start the replication controller.

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 ConnecX SQL Engine administrative machine and the Open Systems Event Replicator Administrator machine, you will not be able to use the Open Systems Event Replicator to replicate to those target tables.

   Note: 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..

   You are now ready to enable the Event Replicator.

Note: If Adabas is upgraded after the Open Systems Event Replicator has been installed, both the Adabas SQL Gateway and the Open Systems Event Replicator MUST be re-installed.

Instalation considerations in an Adabas to Adabas environment

In order to use the new Adabas to Adabas replication feature, the following requirements must be met:

1. Both the source and target Adabas nucleus must be version 6.3.1 or above.
2. The source and target systems must be the same platform. (i.e. both systems must be Solaris, etc.)
3. An Adabas to Adabas license must be installed for the source and target nucleus as well as the Open Systems Event Replicator
4. A CDD is required and it must contain connections for at least two Adabas 6.3.1 databases. It is not necessary to import files, but the connections must be defined. If two 6.3.1 connections are not present in the CDD, Adabas to Adabas replication will not be available in the Replication Administrator.
5. Before doing Adabas the Adabas replication, the source database must be prepared for replication.
    1. ADADBM has a new parameter called REPLICATION_FILES that prepares the source nucleus for replication
    2. Syntax is ADADBM DBID=xx REPLICATION_FILES=(file1, file2, file3, file4) where xx is the database ID of the source and file1 - file4 are the file numbers reserved for replication. An example is:
    ADADBM DBID=1 REPLICATION_FILES=(15,16,17,18)
6. ADADBM can also be called with the REMOVE_REPLICATION parameter to remove the replication files. After making this call, the nucleus will no longer be able to replicate files until ADADBM is called again with REPLICATION_FILES.
7. The environment variable ADABINPATH must be set in the environment(s) where the Replication Server (controller) is running as well as where the CONNX SQL Gateway Server's listener is running. On Unix systems, this environment can be exported in the start up scripts. The startup scripts are in the connx directory and are named eventserver and connxserver respectively. On Windows, this needs to be set in the Environment Settings section of the Computer properties.
    Note: In Windows, the installer will set ADABINPATH automatically.
    Note2: In Unix/Linux the CONNX SQL Gateway Server (data server) must be running in a user account that has permission to execute the binaries in the ADABIN location. It is recommended that both the data server as well as the Event Server (controller) be run from the same user account that was used to start the Adabas nucleus.

The Open Systems Event Replicator is designed to be able to be distributed across multiple machines. Certain components must be installed on the source and target machines, however.

Components and their locations are:

- Source Adabas server:
    1. Adabas SQL Gateway client
    2. Adabas SQL Gateway Adabas data server

3. Event Producer (EP)

- Target Adabas server
    1. Adabas SQL Gateway Adabas data server
- Controller (Event Server):
    1. Adabas SQL Gateway client
    2. Open Systems Event Replicator Controller (Event Server)
- Replication Administrator machine (must be Windows XP SP2 or above):
    1. Adabas SQL Gateway with Open Systems Event Replicator selected

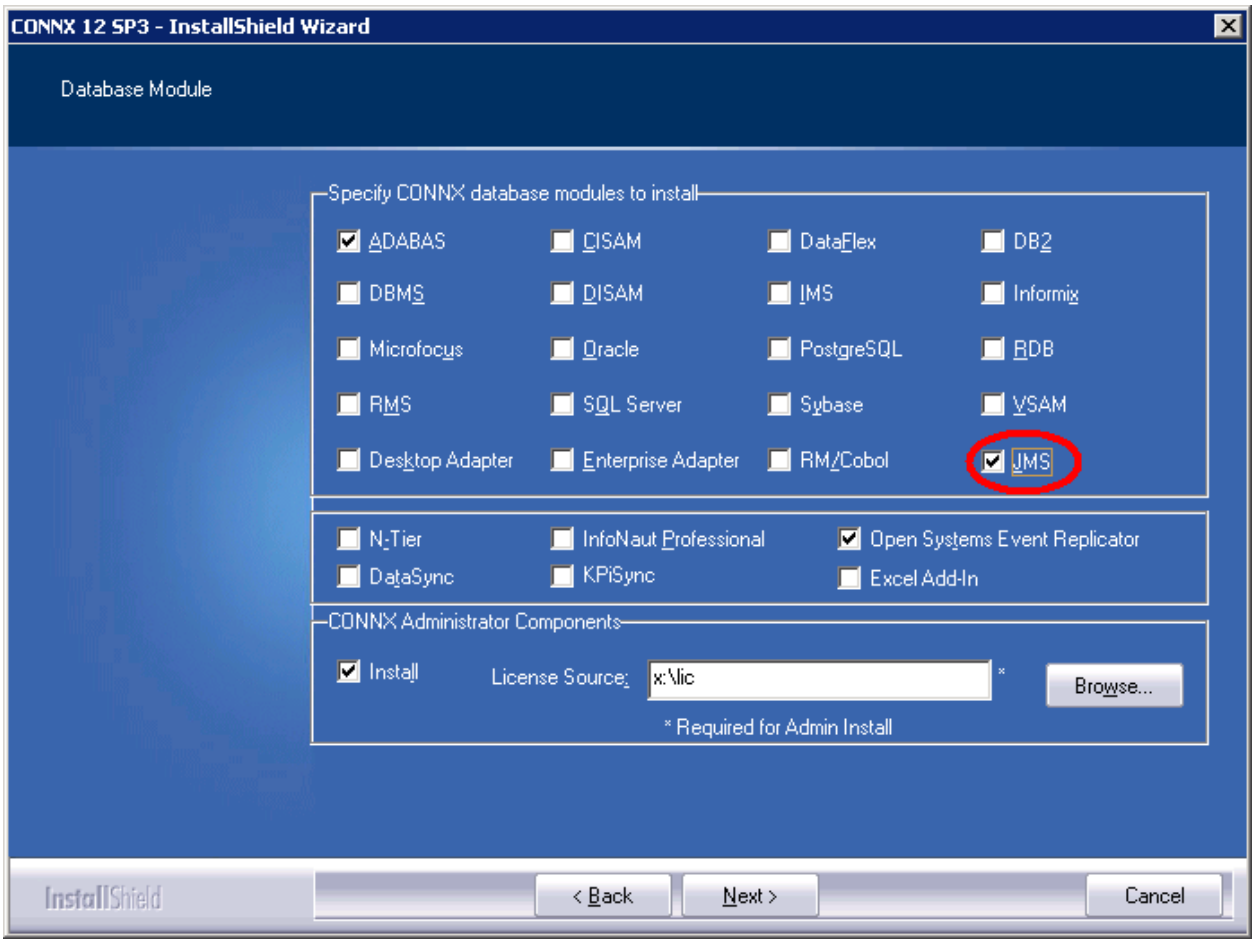
While the Open Systems Event Replicator is designed to be distributed across multiple machines, in an Adabas to Adabas environment, the following configuration is most likely optimal:

- Source Adabas server:
    1. Adabas SQL Gateway Client
    2. Adabas SQL Gateway Adabas data server
    3. Open Systems Event Replicator with both the controller and EP installed
- Target Adabas server:
    1. Adabas SQL Gateway Adabas data server
- Replication Administrator machine (must be Windows XP SP2 or above)
    1. Adabas SQL Gateway with Open Systems Event Replicator selected

Note: For information about installing the Adabas SQL Gateway, please see the CONNX Installation Guide. For information installing Event Replicator, please see the previous topics in this chapter for your specific platform.

Installing the JMS Server

On Windows, the JMS Server is automatically installed when JMS is selected on the Database Module screen during the Windows installation process.



.

For UNIX/Linux, the JMS Server must be installed by the UNIX JMS Installer. During the Windows installation process, select JMS (see screen above). In addition to installing the JMS components to Windows, it will also install the Unix JMS Installer.

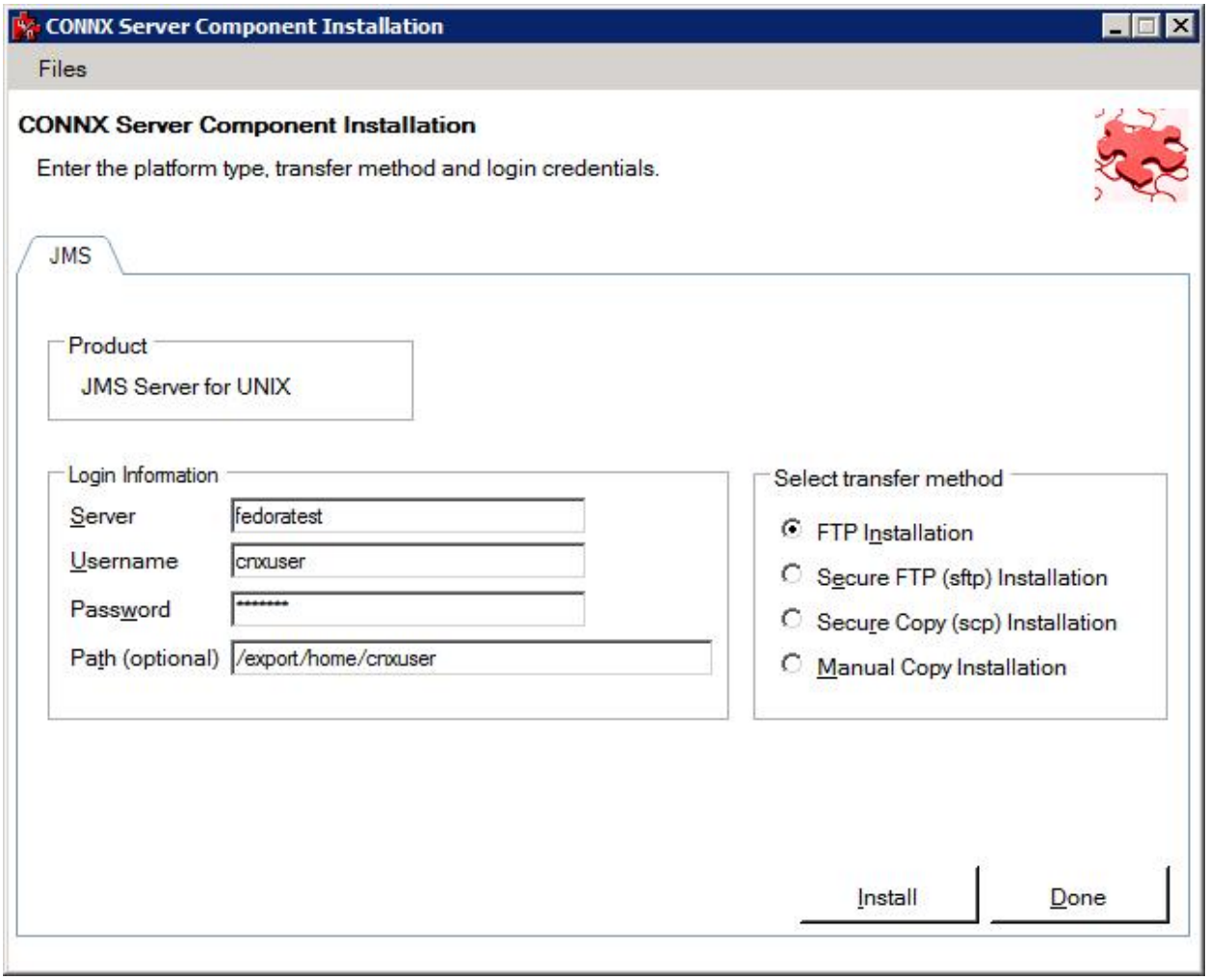
The CONNX JMS Server must be installed on the same machine as the JMS Messaging Server.

The CONNX JMS Server uses the Open Source Apache Logging Services for logging output. This means that the Apache log4j.jar file must be present on the machine where the CONNX JMS Server and the JMS Messaging Server are installed. The log4j.jar file can be downloaded from the Apache web site.

UNIX/Linux Installation

After the Windows installation has completed, go to the Start|All Programs menu and select Unix Installer from the "CONNX JMS Server" menu. This will run the Unix JMS installer which uses FTP, SFTP or SCP to transfer the contents to the selected UNIX/Linux machine. Please consult your system administrator for the transfer method appropriate for your environment.

After the files have been transferred, you will be prompted to complete the installation by invoking "installjms" on the Unix system. This Unix shell script will set up the JMS Server in a directory named connx which will be under the base install library that is specified when running the installjms script.



Note: If you leave the Install Location blank, the installer will copy the files to the home directory of the user specified in the User ID field.

After the installation has completed, the CONNX JMS Server is started on Unix/Linux by running the jmsserver shell script with the START option and on Windows by selecting "Start JMS Server" from the Start|Programs|CONNX JMS Server menu. ("Start JMS Server" is a Windows short cut to jmsserver.bat which is located in the \connx32\CONNXJMS folder) The first time this script is run, you will be prompted for the classpath to the log4j.jar file as well as the location of the JMS provider you wish to connect to. The classpath information will be validated and once the information if found to be correct, the CONNX JMS Server will be started. You will also be prompted for the port the JMS Server will listen on. The default is port 7600.

Changing JMS Server Configuration:

If any of the classpath information or the port number needs to be changed after the initial setup, the JMS Server can be re-configured on Windows by selecting "Configure JMS Server" from the

Start|Programs|CONNX JMS Server menu or by running the jmsserver shell script with the configure option (./jmsserver configure) on UNIX/Linux.

Note for Universal Messaging Server installations:

For the Universal Messaging Server, the default location for the jms jars is \softwareag\UniversalMessaging\lib on Windows and /opt/softwareag/UniversalMessaging/lib on UNIX/Linux. The default location for log4j.jar is \softwareag\common\lib\ext on Windows and /opt/softwareag/common/lib/ext on UNIX/Linux.

Note for webMethods installations:

webMethods does not use the javax.jms interface. When the configuration program prompts for the location of javax, the location of webMethods should be specified. The default location is /opt/softwareag/common on UNIX/Linux and c:\softwareag\common on Windows.

Note: The CONNX JMS Server requires Java JRE version 1.7 or higher.

Details about the CONNX JMS Server are covered more fully in Chapter 4, in the topic, "JMS Queues/Topics as a Replication Target".

Default Port Numbers and Firewall Considerations

The following is a list of Open Systems Event Replicator components and the default ports they use:

Component Name	Default Port
CONNX Message Queue	9200
CONNX Replication Controller	9205
cnxepcfg (or cnxepcfg_64)	9207
CONNX data server for Adabas	6500
ADABCK - Adabas to Adabas replication only. Uses the CONNX data server.	6500
License Server	7501

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

Because the Open Systems Event Replicator 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 an Open Systems Event Replicator component is installed. This includes the Replication Administration machine, the EP machine, and the Event Server (controller) machine.

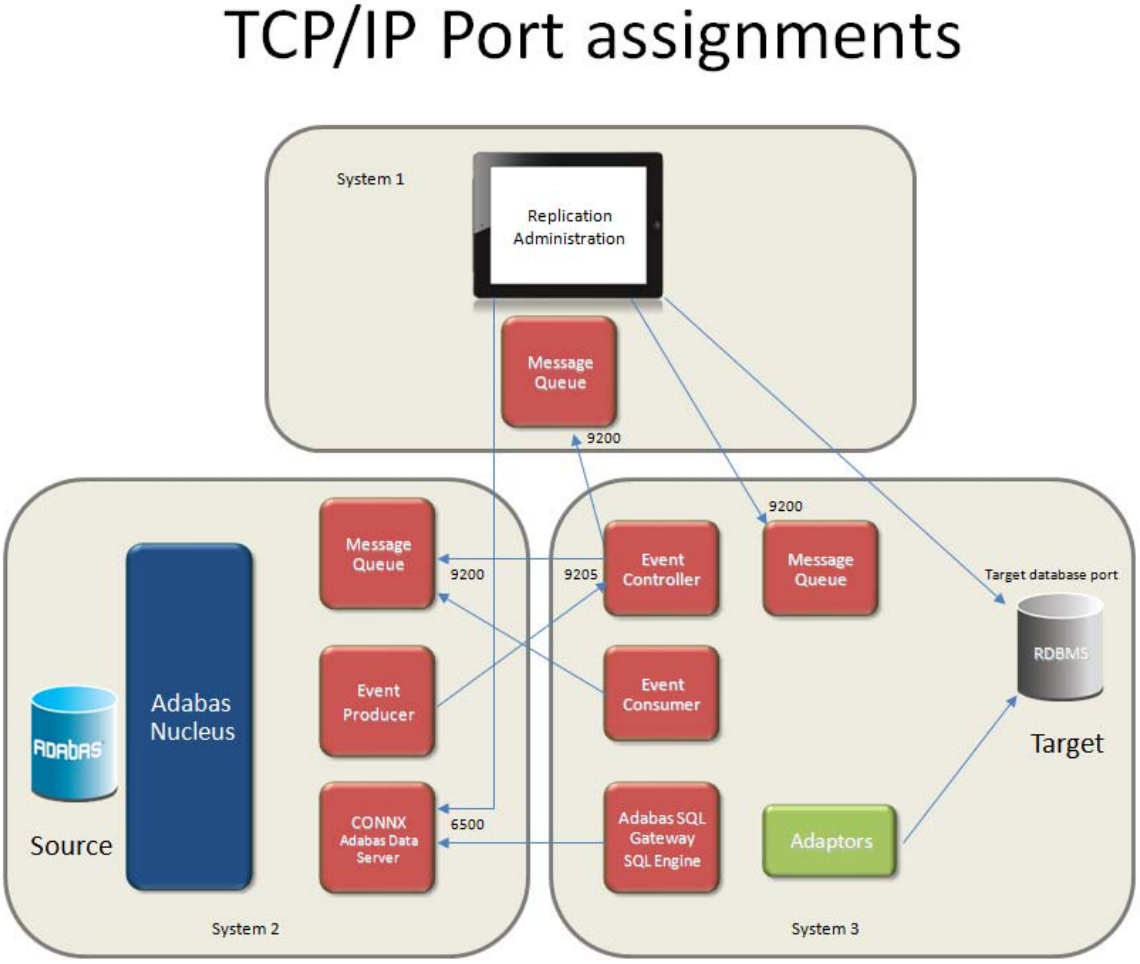
Port 9205: This port is used by the controller to accept connections from the EP. If the EP is on a different machine than the controller, port 9205 on the controller machine must be open and allowed to accept connections from the EP machine. Note, if there are multiple EP machines in the configuration,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

Port 9207: This port is used by an EP helper process and is always running on the same machine as the EP. This port does not need to be opened in the firewall.

Port 6500: This port is used by the CONNX listener (data server). While the data server is not a replication component, it is used by replication to access Adabas during the initial state process. The Replication Administrator also uses the dataserver to connect to the source Adabas database. The data server must be present on the EP machine and port 6500 must be open and allowed to accept connections from the machine where the controller is installed. In an Adabas to Adabas environment, the data server is responsible for moving data between the source and target instances of ADABCK during the initial state process.

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 Adabas database along with the EP installed on System 2 and the Controller along with the target database installed on System 3:



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 port 6500.
  - 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 port 6500 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 port 6500 and 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
- 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 Server (System 3). If the target databas is on a different system, the appropriate port will need to be open on that system and allow access from System 3 and System 1.

Note: This example depicts the Open Systems Event Replicator with a relational database as the target. If the target database is another Adabas database, the Adabas Entire Network is required. In addition to the ports listed above, additional ports may need to be opened in the firewall. See the Adabas documentation for more details.

Note: In an environment where a firewall is present between the data server and any other system, the Data Server parameter **CNXCONNECTBACK** should be set to 0.

Replication Components

The following is a description of the Open Systems Event Replicator 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 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.
EP Configuration Helper	cnxepcfg_64	connx/replicator/producer	the EP (libcnxepada_64)	connx/replicator/log/EP_CONFIG.log	Invoked by the first replication enabled nucleus to be started and stays up until the last replication enabled nucleus is shut down. This process is a helper of the EP and is not the actual EP.

					The EP communicates with this process via TCP/IP on port 9207.
CONNX Event Producer (EP)	libcnxepada_64.so*	connx/replicator/producer	shared library loaded into the Adabas nucleus process space	connx/replicator/log/EP_<dbid>.log	The EP is shared library and not a stand alone process. It is loaded by the Adabas nucleus when the nucleus has been enabled for replication. The EP is responsible for starting the cnxepcfg_64helper process.

***** on AIX systems, the EP has an extension of .a rather than .so

In addition to the above Replication Components, the CONNX Adabas Data Server components are also installed on the EP machine (and possibly the Target Databas machine). The Data Server components are:

Component Name	Process Name	Location	Invoked by	Log File	Comments
CONNX Listener	cnxrun	connx/	connxserver script in the connx directory	connx/cnxrun.log	The Adabas Data Server is made up of a Listener and Server component. cnxrun is the listener component
CONNX Server	cnxadaC0	connx/	cnxrun	connx/cnxrun.log	A separate cnxadaC0 process is started for each connection made to Adabas from a CONNX client. The number of running cnxadaC0 process will be dependant on the number of active connections to the 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.
EP Configuration Helper	cnxepcfg.exe	c:\connx32\replication\producer	the EP	c:\connx32\replication\log\ EP_CONFIG.log	Invoked by the first replication enabled nucleus to be started and stays up until the last replication enabled nucleus is shut down. This process is a helper of the EP and is not the actual EP. The EP communicates with this

					process via TCP/IP on port 9207.
CONNX Event Producer (EP)	cnxepada64.dll	c:\connx32\replication\producer	shared library loaded into the Adabas nucleus process space	c:\connx32\replication\log\ EP_<dbid>.log	The EP is shared library and not a stand alone process. It is loaded by the Adabas nucleus when the nucleus has been enabled for replication. The EP is responsible for starting the cnxepcfg helper process.

In addition to the above Replication Components, the CONNX Adabas Data Server components are also installed on the EP machine (and possibly the Target Databas machine). The Data Server components are:

Component Name	Process Name	Location	Invoked by	Log File	Comments
Enterprise Server Service (CONNX Listener)	cnxremote.exe	c:\connx32\cnxremote\bin	Windows Service	c:\connx32\cnxremote.log	The Adabas Data Server is made up of a Listener and Server component. cnxremote is the listener component on Windows
CONNX Server	cnxadabas.dll	c:\windows\system32	cnxremote or loaded directly into CONNX client depending on	c:\connx32\cnxremote.log	A separate server thread is started for each connection made to

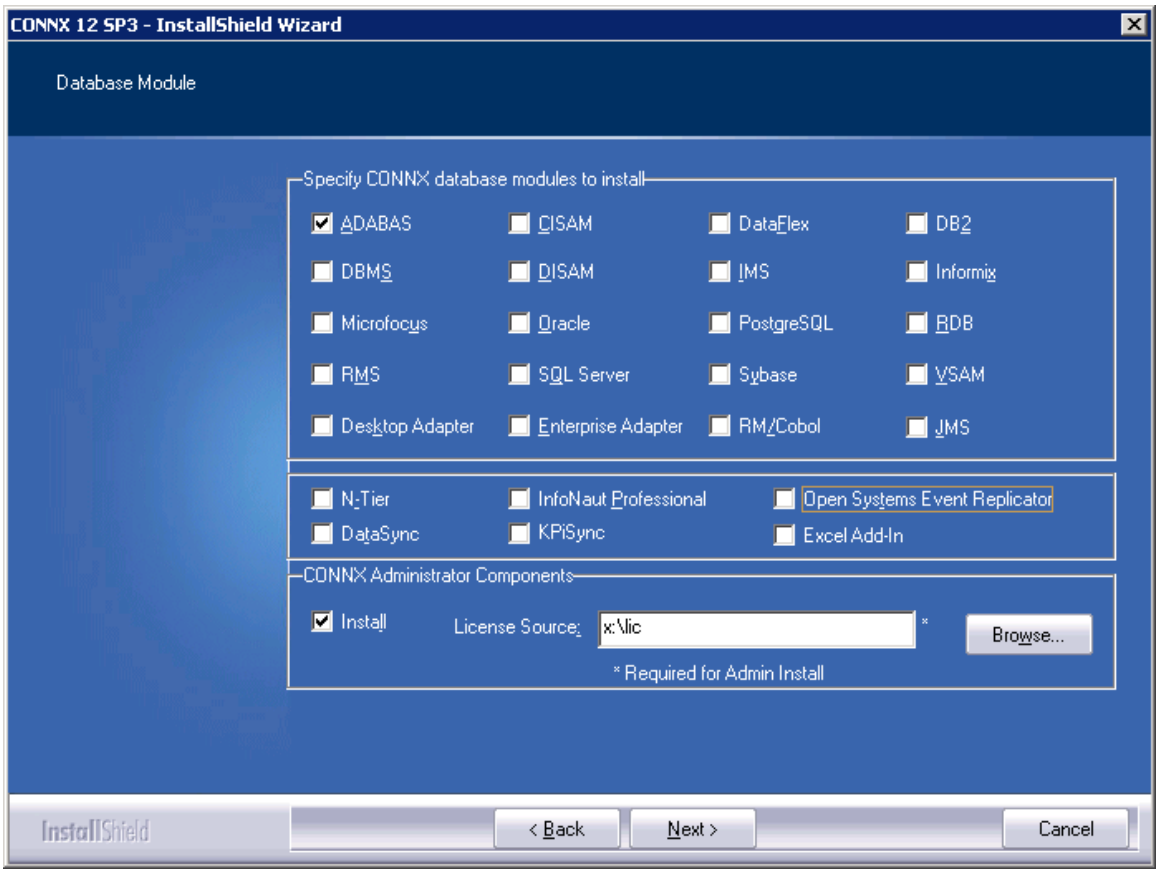
			configuration		Adabas from a CONNX client. The number of threads will be dependant on the number of active connections to the database.

Chapter 3 - Event Replicator Removal

Uninstalling the Event Replicator

To uninstall the Event Replicator, use the Installation Wizard.

1. On the Database Modules screen, uncheck Open Systems Event Replicator checkbox and press Next to continue.



Creating a CDD for Event Replication using the CONNX SQL Engine

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

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

Create a CDD for Event Replication using the CONNX SQL Engine if:

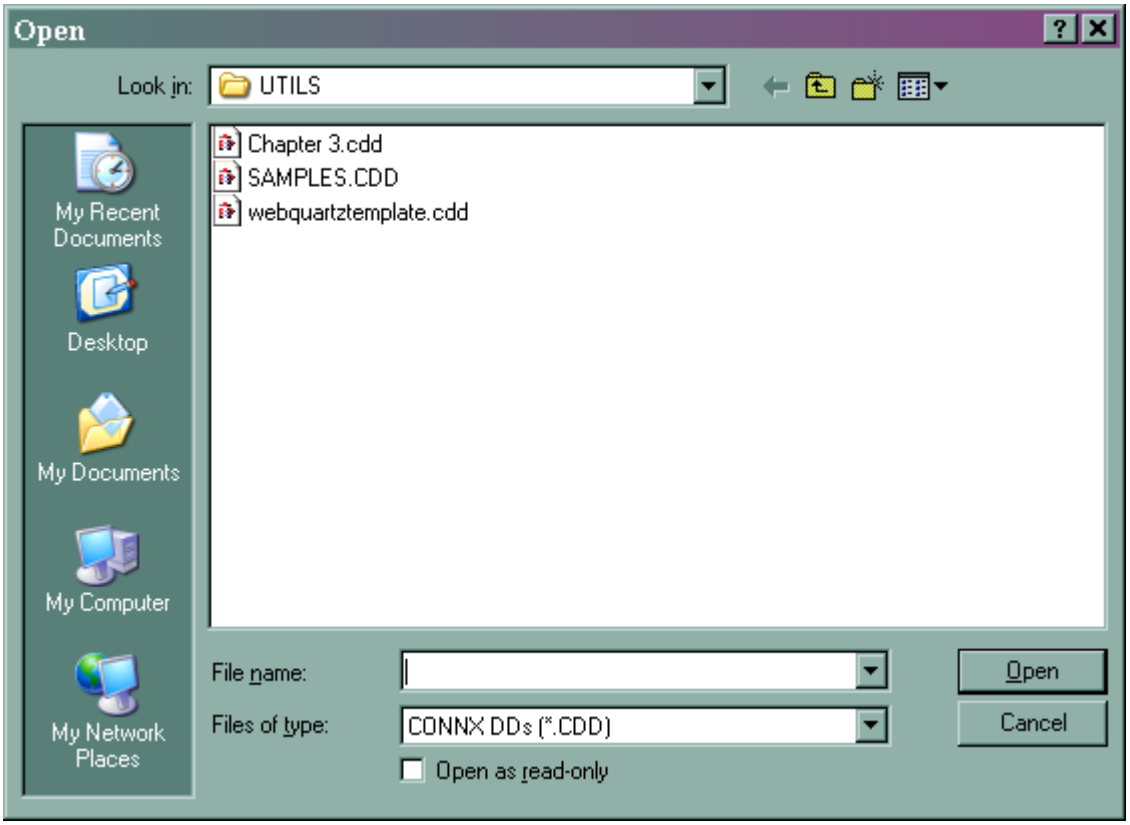
- You have never used the CONNX SQL Engine before
- You do not have an existing CDD containing the source tables and target database needed for event replication.

After you create a CDD for Event Replication using the CONNX SQL Engine you will be ready to:

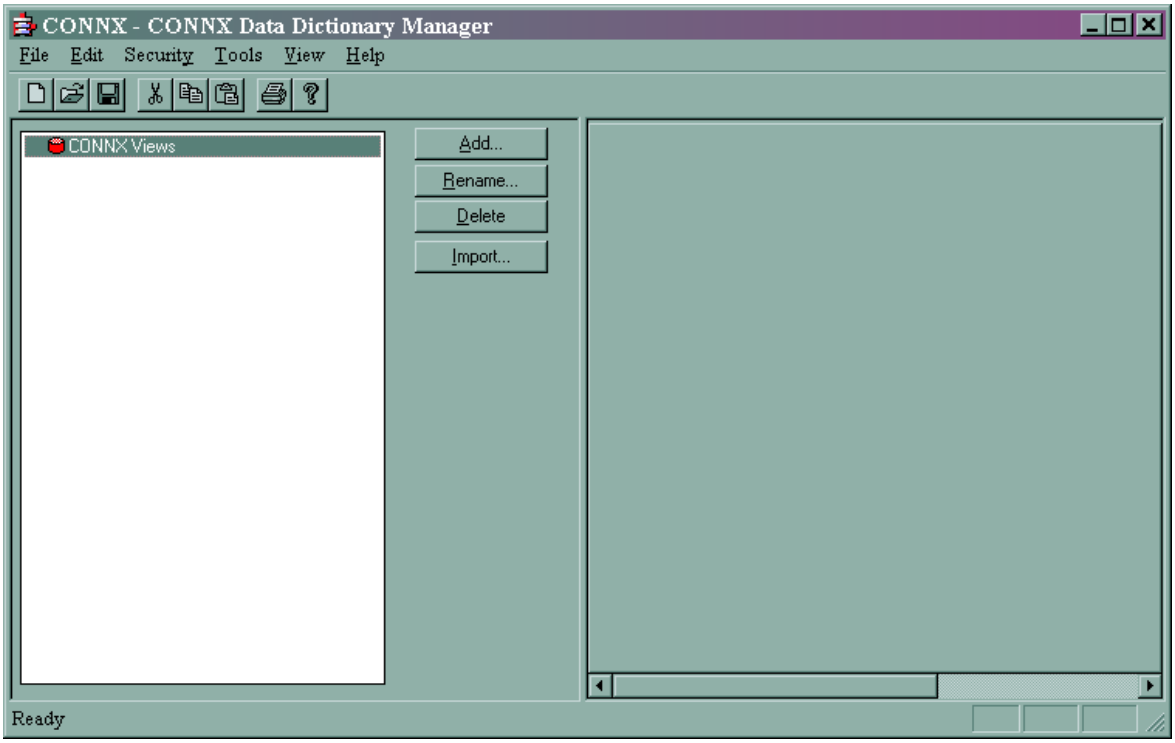
- Enable the Event Replicator

The Open Systems Event Replicator 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.

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

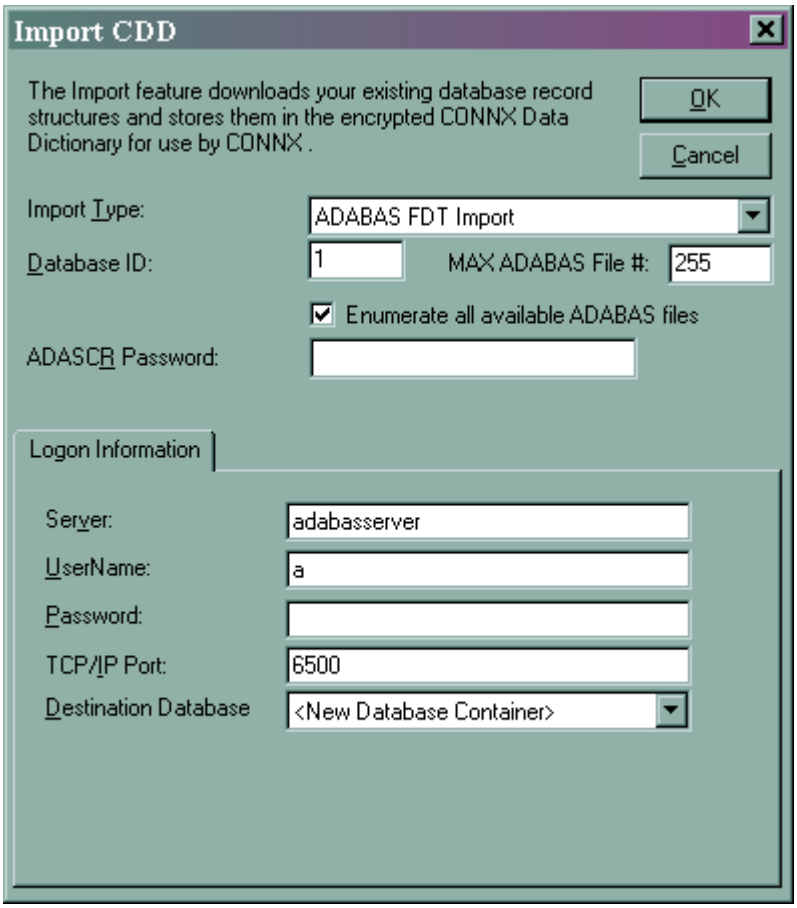


2. Click **Cancel**. The **CONNX Data Dictionary Manager** window appears.



Importing the Adabas Source Tables

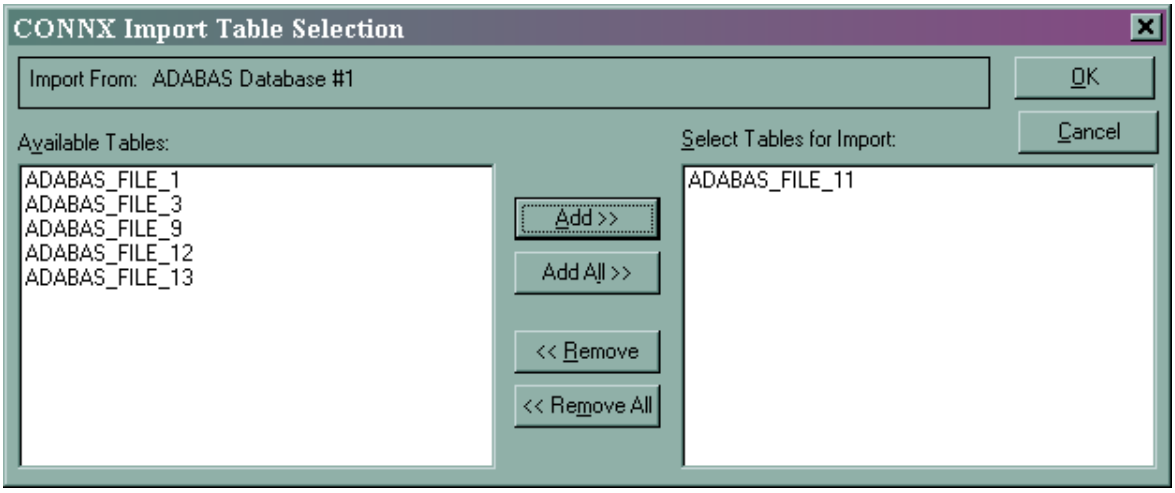
1. Click **Import**. The **Import CDD** window appears.



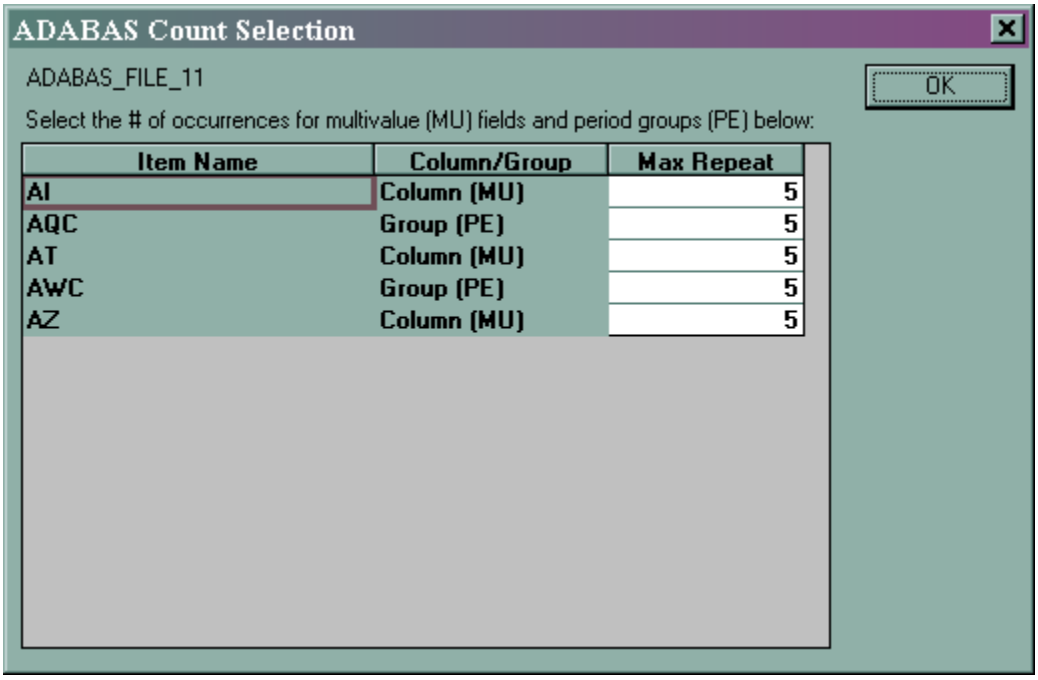
2. In **Import Type**, select the correct Adabas Import. Type (the example shows FDT Import). Type the Adabas Database ID for the Adabas source tables in **Database ID**. Type 255 in **MAX ADABAS File #**. Type in the logon information.

   Warning: Do not use localhost as the server name for the Adabas Source or Target tables. In certain instances using localhost as the server name may cause you to lose all your source table records during an initial state.

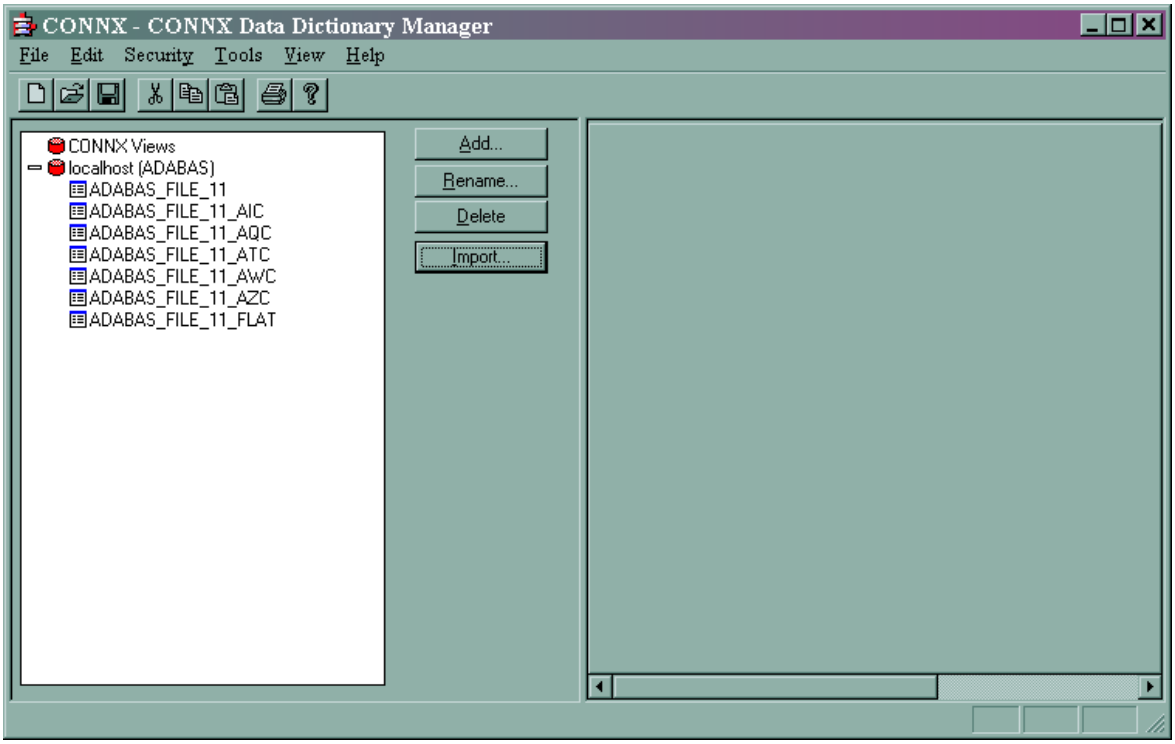
   Click **OK**. The **CONNX Import Table Selection** window appears.



3. Add the source tables to be replicated to the target database and click **OK**. The **ADABAS Count Selection** window for the source table appears.



4. If the default values are incorrect, change them. Click **OK**. The **CONNX Data Dictionary Manager** window containing the source table information appears.



You have imported the Adabas source tables into the CDD.

Importing the Target Database

If the target database is an Adabas database, follow the Importing the Adabas Source Tables instructions above and import the target database and its tables.

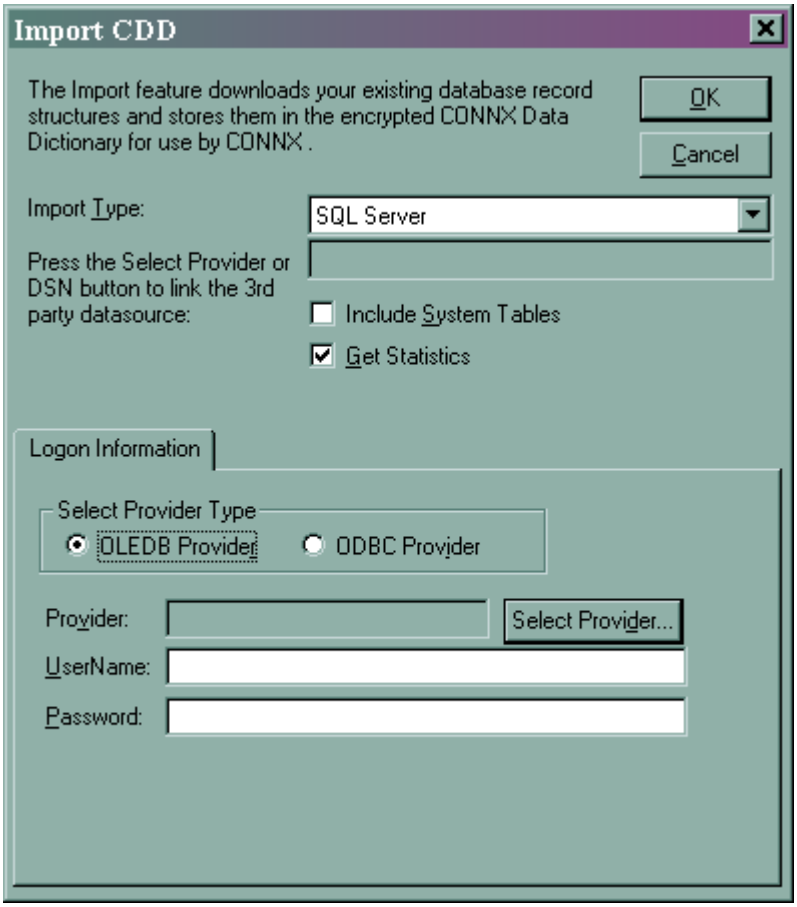
Warning: The Event Replicator cannot create Adabas target tables. Adabas target tables must be created using Adabas, not the Event Replicator.

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

Warning: If your UNIX target database is not correctly configured on both the ConnecX SQL Engine administrative machine and the Open Systems Event Replicator Administrator machine, you will not be able to use the Open Systems Event Replicator to replicate to those target tables.

For non-Adabas target databases on Windows:

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



2. In **Import Type**, select the target database type.

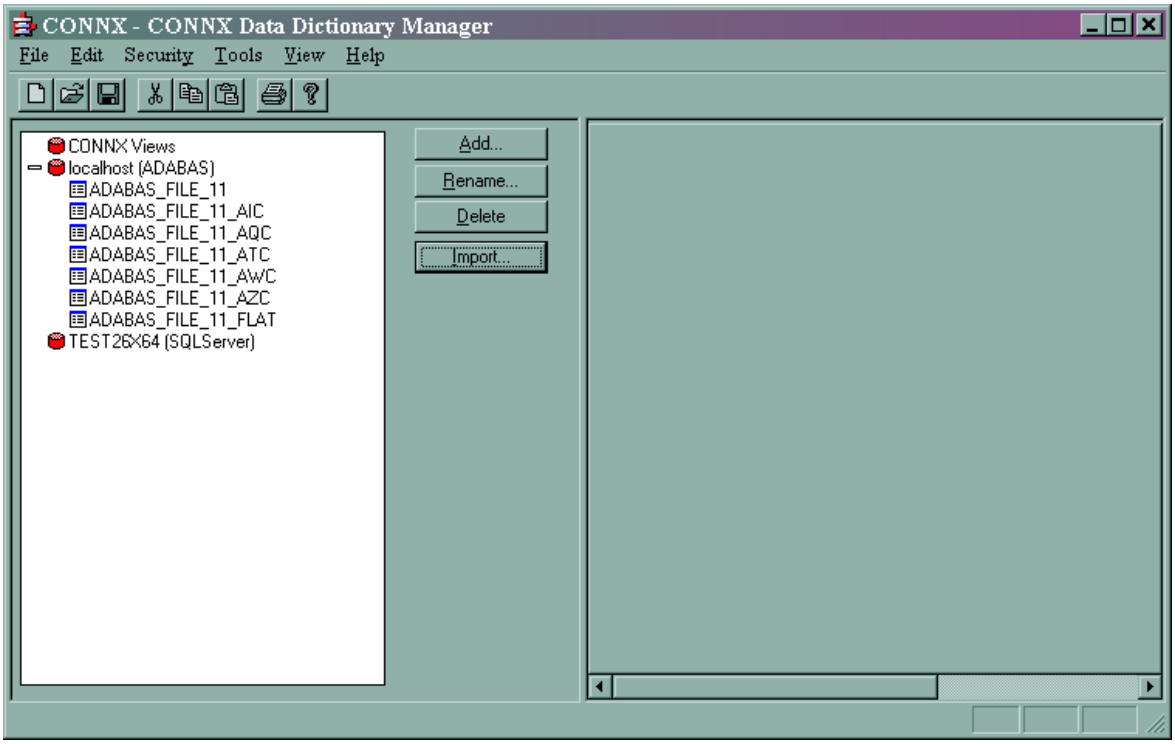
   Note: Different target databases will have Import CDD windows with different required information. The example shows SQL Server.

3. In **Select Provider Type**, click **OLEDBC Provider**. Click Select Provider and supply the required information. Click **OK**.

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

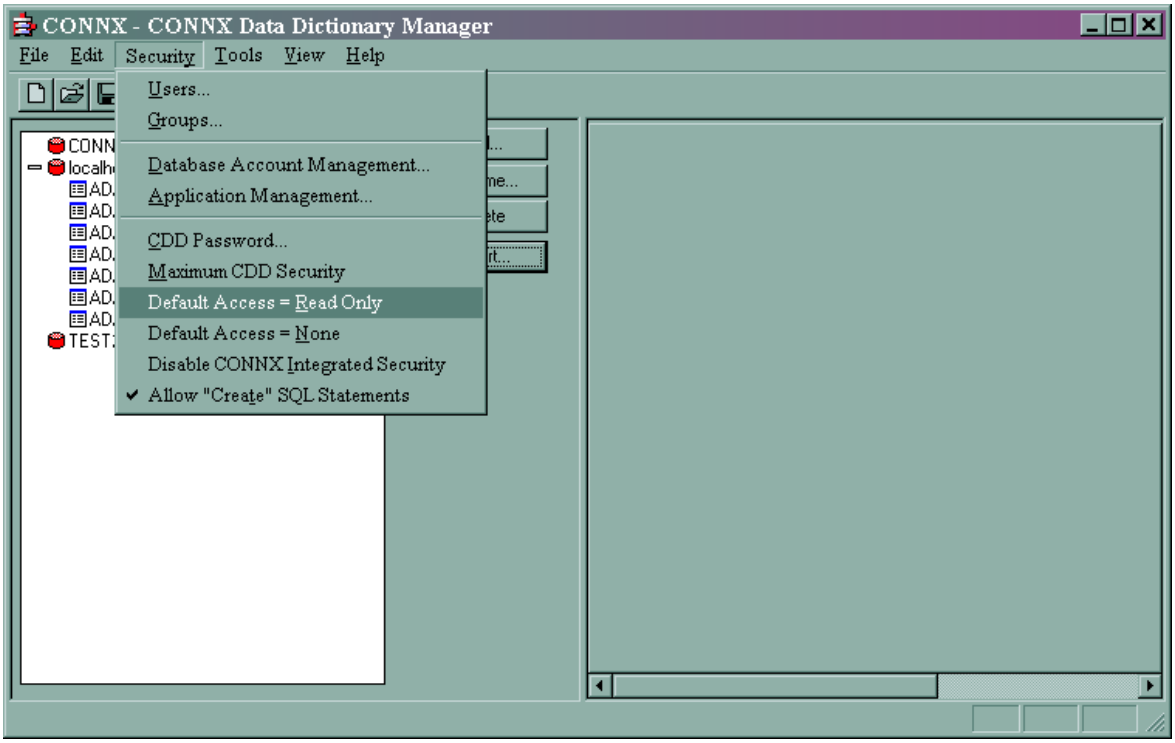
   - If any of the existing tables will be used as target tables, select them and click **Add**. After enabling a CDD for Event Replication, map the source columns to the target table.
   - If none of the existing tables will be target tables, do not select anything. You will create the new target table 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.

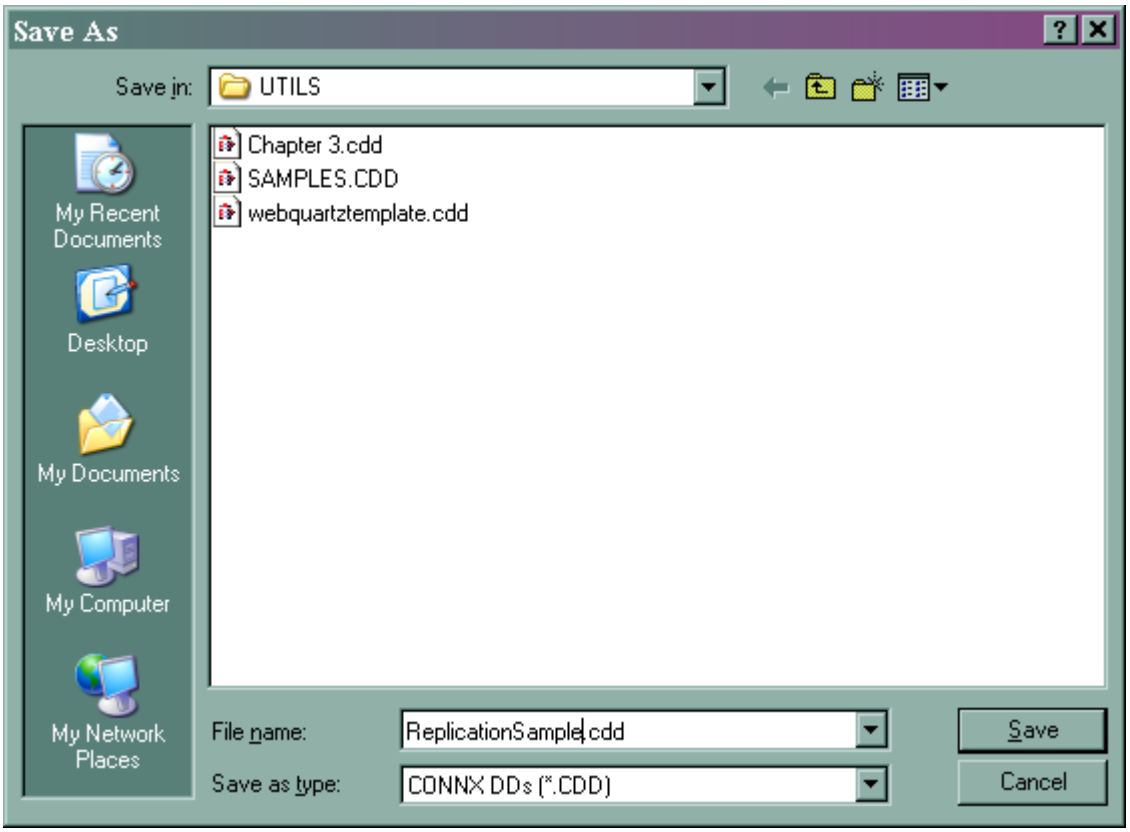


Set the CDD Security

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 use the Open Systems Event Replicator.
- 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:

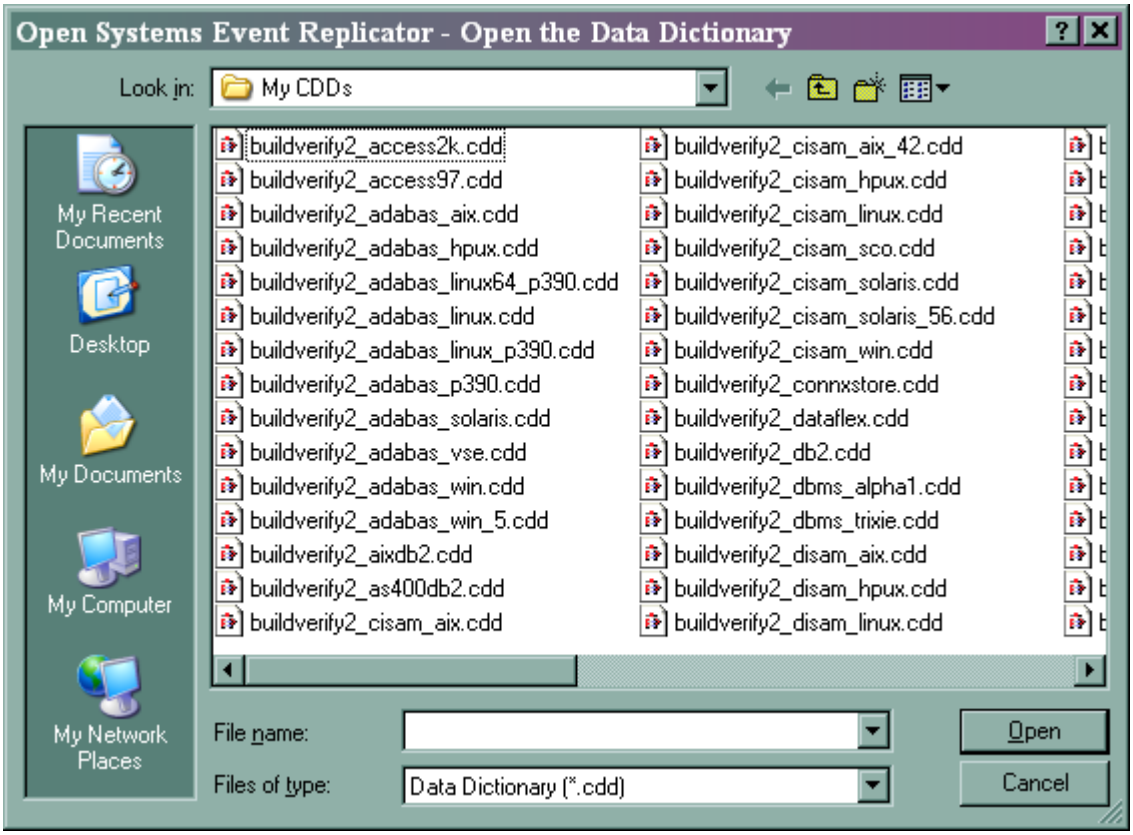
- Modified with column mapping
- Deployed (which starts replicating data from source database to target database)

The Event Replication 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.

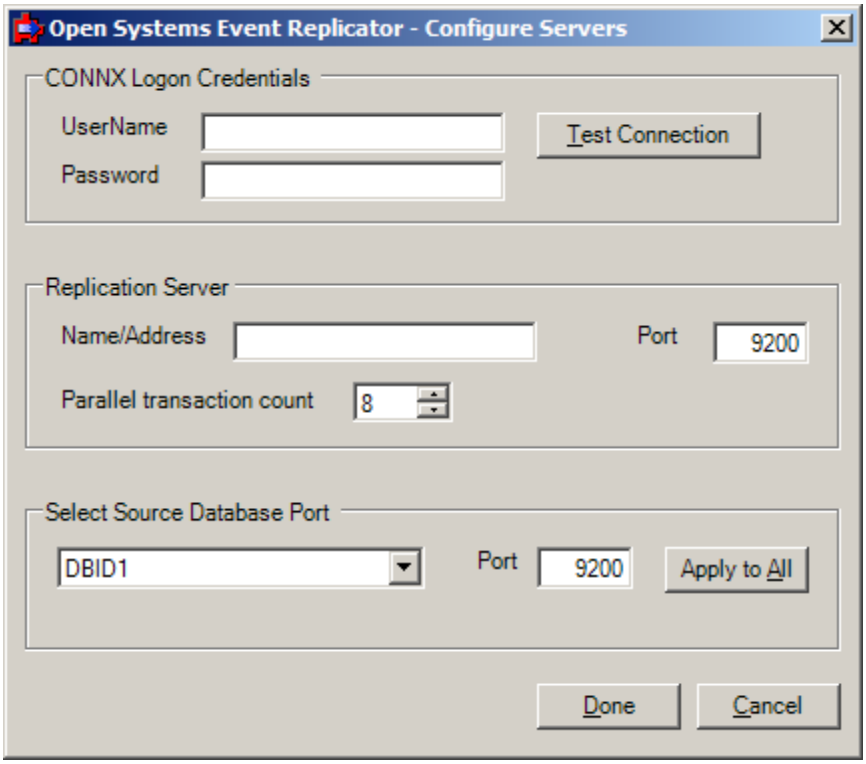
Caution: Two separate CDDs cannot replicate to the same targets from two different controllers. The Event Replicator is designed to maintain data integrity from the source to the target; any target modifications from more than one controller will result in replication errors, disabling the Event Replicator.

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

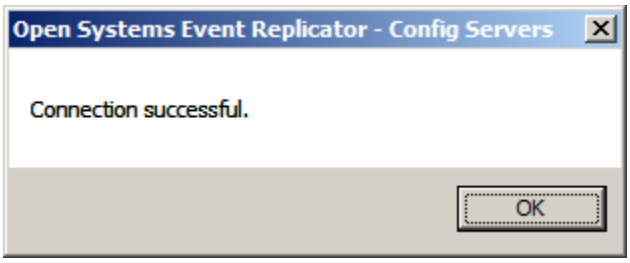


Note: Only CONNX Data Dictionaries (CDD) that contain both the source and target (destination) data bases can be used for replication.

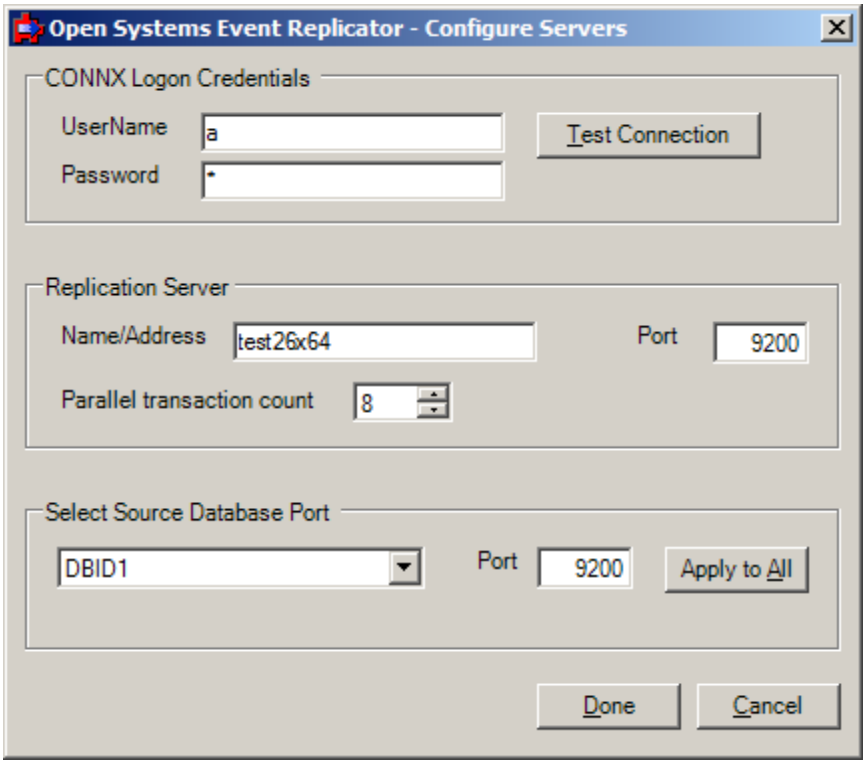
2. If the Replication Administrator has been used previously, the **Open Systems Event Replication** window appears. The Event Replicator will normally open the last opened CDD.
    - If this is the CDD you want, you do not need to re-enable the Event Replicator for this CDD.
    - If this is not the 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.



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:

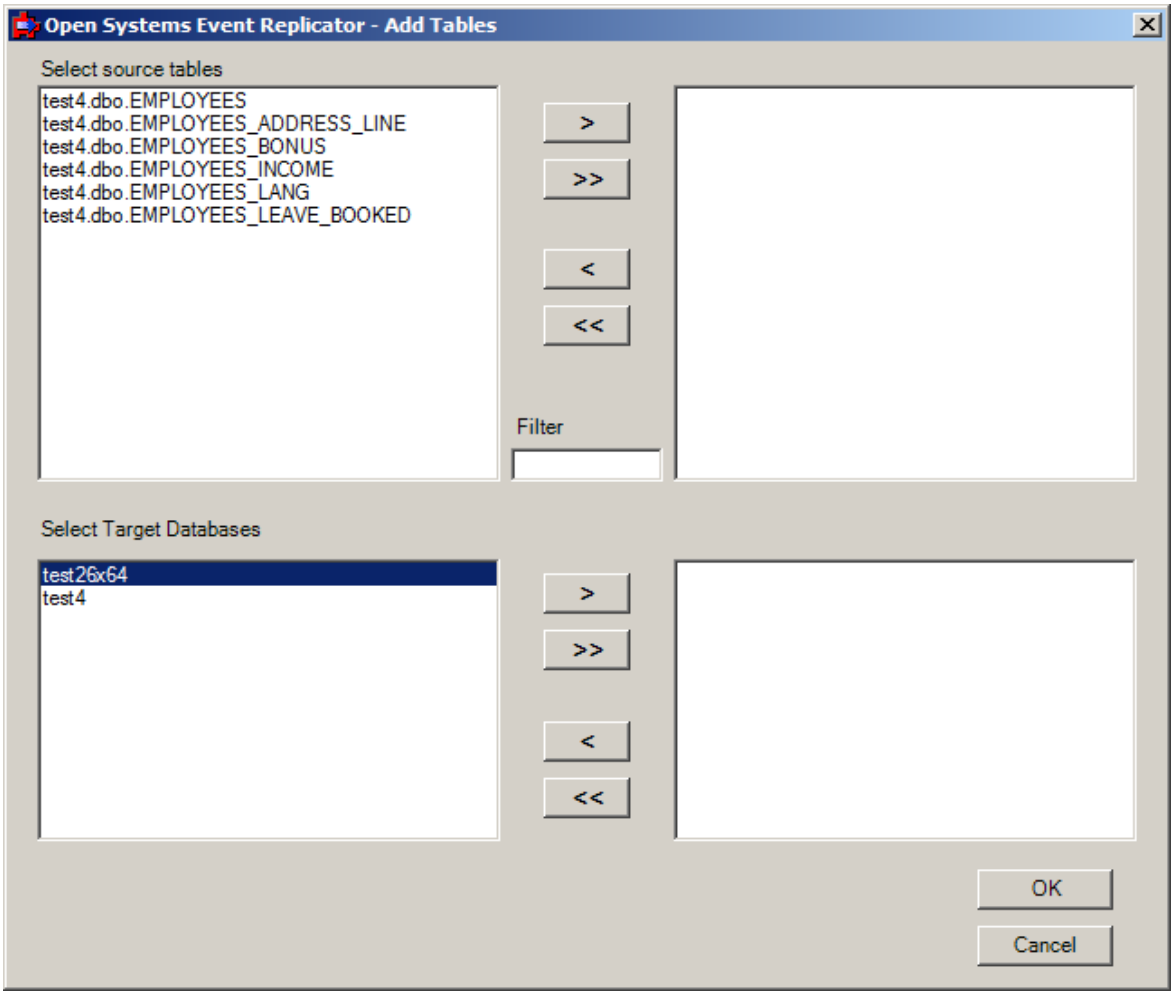


5. Click **OK**. The message window closes.
6. Enter the server name or IP address of the Replication Server in **Name/Address** and the server port number in **Port**.

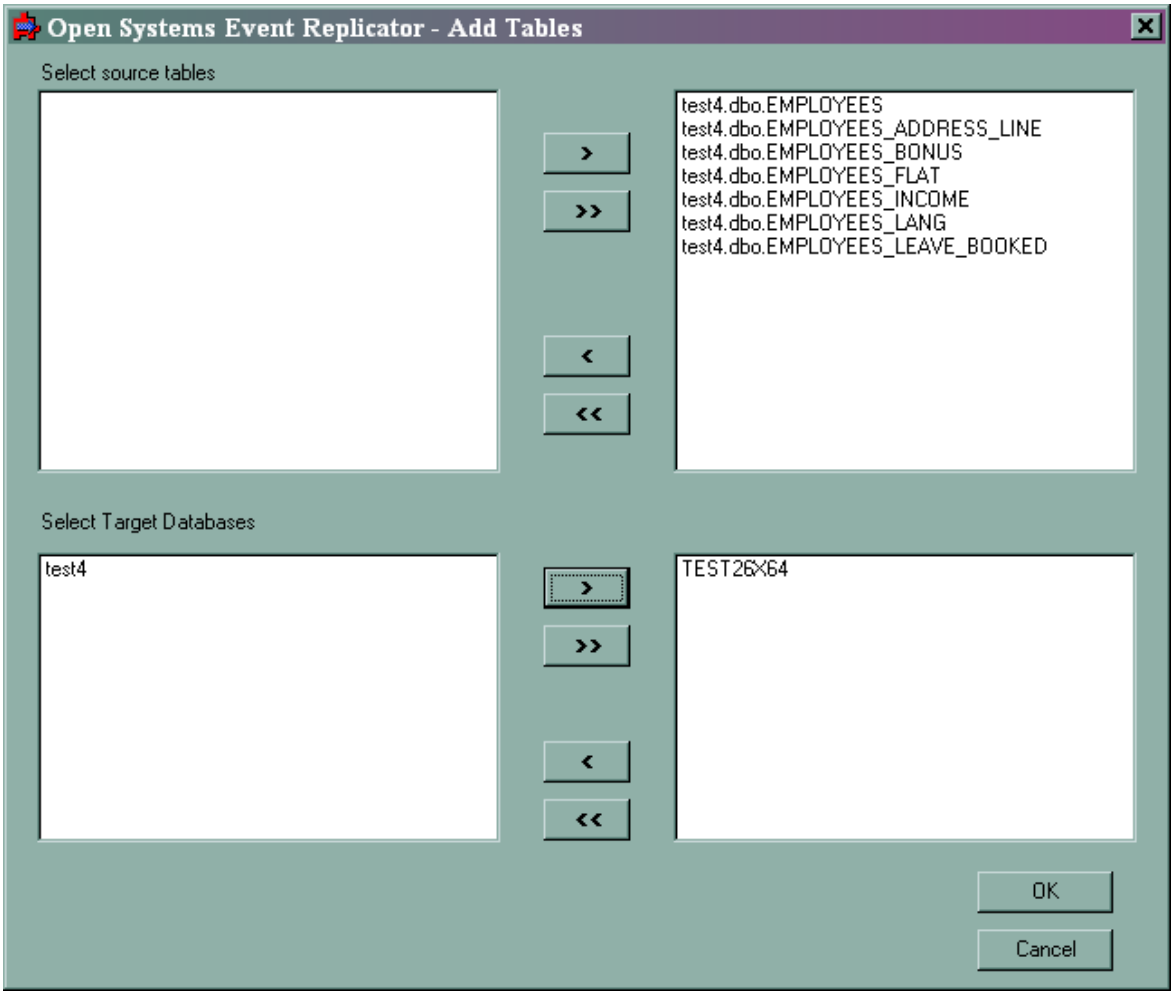


Note: The default port number is 9200.

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



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



9. Click **OK**. The **Open Systems Event Replication** window appears.

JMS Queues/Topics as a Replication Target

The Open Systems Event Replicator supports replicating to the JMS Queues and Topics. The process of replicating to a JMS Queue or Topic is the same as replicating to a relational target such as SQL Server or Oracle except that the data placed on the JMS Queue or Topic is in xml format rather than being put directly into a physical database.

Start the CONNX JMS Server component

Configure a JMS Queue or Topic

Creating a CONNX JMS CDD Entry This includes:

- Opening the CONNX Data Dictionary (CDD) Manager
- Importing the JMS connection
- Saving the CDD

Starting the CONNX JMS Server

The CONNX JMS server must be installed on the same platform in which the JMS Que/Topic provider resides. The CONNX JMS Server is a java program that communicates between CONNX and the JMS Queues/Topics. If the JMS provider resides on an UNIX Machine, additional installation steps must be done which are described below.

The CONNX JMS Server uses the Open Source Apache Logging Services for logging output. This means that the Apache log4j.jar file must be present on the machine where the CONNX JMS Server and the JMS Messaging Server are installed. The log4j.jar file can be downloaded from the Apache web site.

To install CONNX JMS Server on a Windows machine:

The JMS Server is automatically installed on Windows when JMS is selected during the Install. To start the JMS Server, select "Start JMS Server" from the Start|Programs|CONNX JMS Server menu. The first time the JMS Server is run, you will be prompted for the classpath to the log4j.jar file as well as the location of the JMS provider you wish to connect to. The classpath information will be validated and once the information if found to be correct, the CONNX JMS Server will be started.

To install CONNX JMS Server on an Unix/Linux machine:

1. On the Windows PC where CONNX is istalled, go to the Start|All Programs menu and select "UNIX Installer" from the Start|Programs|CONNX JMS Server menu.
2. Fill in the dialog box and press Install. (see "Installing JMS" in Chapter 2 - Event Replicator Installation for more information)
3. This install program will ftp the JMS Server files to the Unix/Linux machine.
4. Exit the install program and sign on to the Unix/Linux machine and run the script ./installjms.
5. This will unpack the JMS Server and you will now have a new script called jmsserver.
6. The file created is jmsserver. It is located in the connx subfolder under the installation directory. The first time this batch file is run, you will be prompted for the classpath to the log4j.jar file as well as the location of the JMS provider you wish

to connect to. The classpath information will be validated and once the information if found to be correct, the CONNX JMS Server will be started.

To start the JMS server component:

1. Ensure that version 1.7 or higher of the JAVA Runtime Environment has been installed on your computer.
2. Start the JMS Server
   - On Unix: execute the jmsserver script with the start paramater. e.g. ./jmsserver start. Use the stop paramater to stop the JMS Server. This script is located in the /connx folder.
   - On Windows: Go to the Start Menu and select "Start JMS Server" from the CONNX JMS Server menu. This will start the JMS Server in a java command window. To terminate the server, use CTRL-C within the JMS Server command window.

Note: The JMS Server listens on port 7600 by default. This can be changed on Windows by running "Configure JMS Server" from the CONNX JMS Server menu and on UNIX/Linux run the shell script jmsserver with the configure option (./jmsserver configure).

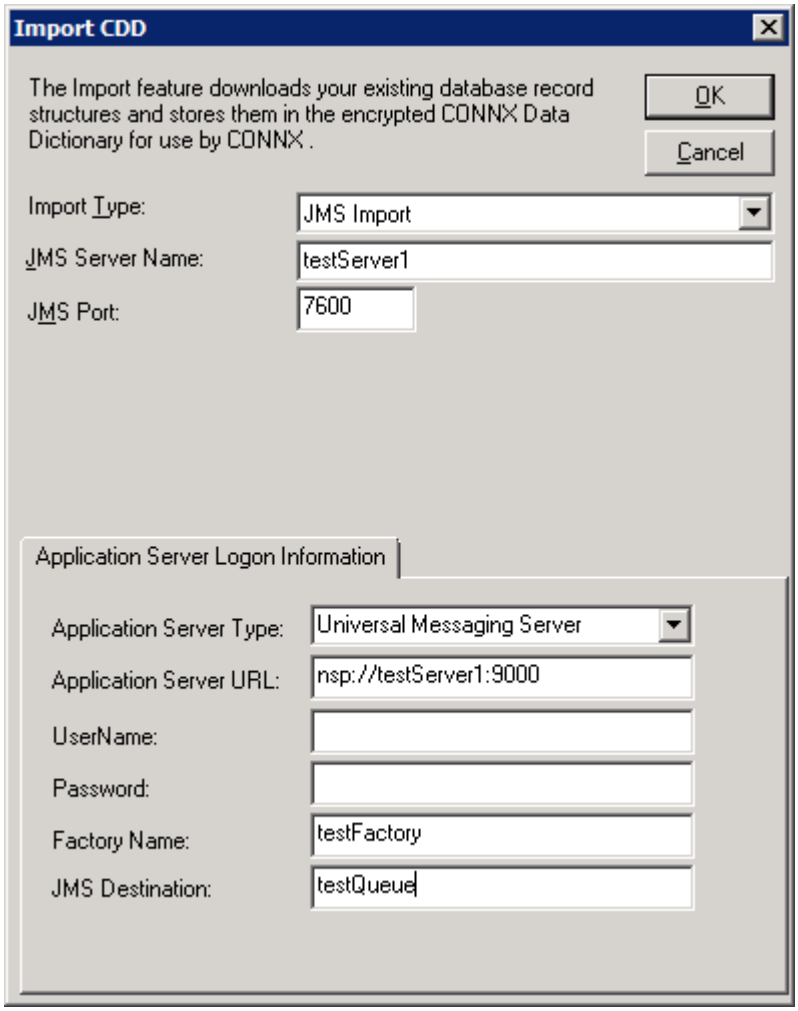
Note: The JMS Server Component must be started on the same platform in which the WebMethods Message Service Provider resides.

Create a JMS entry in the CDD:

To create the CONNX JMS CDD entry, bring up the "CONNX - CONNX Data Dictionary Manger and perform the following:

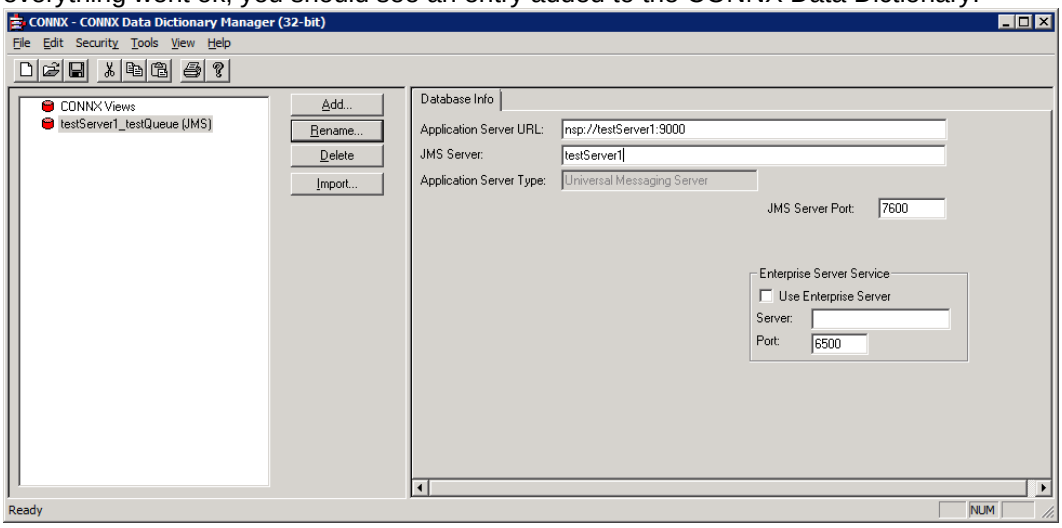
1. Select the "Import" button
2. In the "Import CDD" dialog, for "Import Type:", select "JMS Import"
3. In the "JMS Server Name:" item, enter the machine name where the JMS server is running. (for this example, it is assumed that the JMS server will be running on the local machine, so "localhost" is used).
4. Leave the "JMS Port:" as is (port 7600). If you changed the "listening port" when starting up the cnxJMSserver.jar, then this value must match the value you specified at that time.
5. Under the "Application Server Logon Information" tab, the "Application Server Type:" select the Application Server Type you wish to connect to. Universal Messaging Server is selected by default. Note, if JNDI Properties File is selected, the JNDI Properties file needs to be located in the same location as the CONNX JMS Server. There is a sample JNDI.properties file called JNDI.properties.sample in this location. You can copy it and name the copy JNDI.properties and then modify that file.
6. If Universal Messaging Server, Glassfish or webMethods is selected, "Application Server URL:" will be set to the defaults for the selected type. The name of the server will be auto filled from the JMS Server Name. While the JMS Server Name can be "localhost" or 127.0.0.1, the server name specified in the Application Server URL CANNOT be "localhost" or 127.0.0.1. A valid DNS host name or IP address (other than local loopback) must be used. It is recommended that the JMS Server Name also be the DNS name or IP address of the server.

7. If JNDI Properties File is selected, the values will be read from jndi.properties which should be located in the same location as the JMS Server. A sample file named jndi.properties.sample is provided and is located in \connx32\connxjms on Windows and in the connx directory on UNIX/Linux. This file can be modified and saved as jndi.properties.
8. Leave "UserName:" and "Password:" entries blank unless a user name and password has been associated with the JMS queue. In that case, use the appropriate values.
9. In the "Factory Name:" item, enter the Factory Name you wish to use.
10. In the "JMS Destination:" item, enter the Queue Name or Topic Name you wish to connect to.



11. Enter "OK".

12. An attempt will be made to connect to the webMethods Message Service Provider. If everything went ok, you should see an entry added to the CONNX Data Dictionary.



13. Select File from the menu and then Save.

JMS Data Format (XML format)

The Open Systems Event Replicator writes data to the JMS Queues and Topics in an XML format. Each event from the source database is written to an XML document that contains one or more of the following tags:

- **CONNXdocument** - This tag represents all the elements of a single event from the source database. It contains the property "committed" which provides the date and time it was placed on the message queue.
- **method** - This tag describes which of the three record formates are being represented. Possible values are insert, update and delete.
- **tableName** - This tag represents the table name as it is displayed in the CDD under the target JMS database icon.
- **fields** - This tag represents a collection of field names and values.
- **field name** - This tag represents the column name as it is displayed in the CDD for the table description it is part of. If the source Adabas file was imported into the CDD via an FDT import, the field names will be represented in the CDD as the Adabas short names and will therefore be represented in the XML as short names. If the CDD was imported using DDL or an SYSOBJH file, the CDD may contain long names and by extension, the XML will also contain long names.
- **length** - Used in the Create method and specifies the length of the field.
- **pre** - Used in the Create method and specifies the precision of the field.
- **scale** - Used in the Create method and specifies the scale of the field. For data types that don't use scale, the value will be zero.
- **value - This property of field name contains the value of the data for this record.**
- **whereCriteria** - This is the SQL Where criteria used to identify a specific record.

There are five record formats used. They are identified as:

Delete (Puts where Criteria in order to identify record)

```
<?xml version="1.0" encoding="UTF-8"?>
<CONNXdocument version="1"  committed="2012/05/04-12:39:25">
  <method>delete</method>
  <tableName>ADABAS_EMPLOYEES_JMS</tableName>
  <whereCriteria>"ISN_EMPLOYEES" = 1434  </whereCriteria>
</CONNXdocument>
```

Insert (Shows all fields that were inserted)

```
<?xml version="1.0" encoding="UTF-8"?>
<CONNXdocument version="1"  committed="2012/05/04-12:41:04">
  <method>insert</method>
  <tableName>ADABAS_EMPLOYEES_JMS</tableName>
  <fields>
    <field name="ISN_EMPLOYEES" value="1472" />
```

```
    <field name="PERSONNEL_ID" value="90001041" />
    <field name="FIRST_NAME" value="John" />
    <field name="LAST_NAME" value="Smith" />
    <field name="MIDDLE_NAME" value="Robert" />
    <field name="MAR_STAT" value="S" />
    <field name="SEX" value="M" />
    <field name="NBIRTH" value="711106" />
    <field name="CITY" value="Redmond" />
    <field name="POST_CODE" value="98052" />
    <field name="COUNTRY" value="USA" />
    <field name="AREA_CODE" value="425" />
    <field name="PHONE" value="5551234" />
    <field name="DEPT" value="SALES" />
    <field name="JOB_TITLE" value="SALES ENINGEER" />
    <field name="LEAVE_DUE" value="20" />
    <field name="LEAVE_TAKEN" value="1" />
  </fields>
</CONNXdocument>
```

Update (Changed CITY field to Kirkland. It shows all the fields for the updated record and contains where Criteria in order to identify the record)

```
<?xml version="1.0" encoding="UTF-8"?>
<CONNXdocument version="1"  committed="2012/05/04-12:41:09">
  <method>update</method>
  <tableName>ADABAS_EMPLOYEES_JMS</tableName>
  <whereCriteria>"ISN_EMPLOYEES" = 1472  </whereCriteria>
  <fields>
    <field name="ISN_EMPLOYEES" value="1472" />
    <field name="PERSONNEL_ID" value="90001041" />
    <field name="FIRST_NAME" value="John" />
    <field name="LAST_NAME" value="Smith" />
    <field name="MIDDLE_NAME" value="Robert" />
    <field name="MAR_STAT" value="S" />
    <field name="SEX" value="M" />
    <field name="NBIRTH" value="711106" />
    <field name="CITY" value="Kirkland" />
    <field name="POST_CODE" value="98052" />
    <field name="COUNTRY" value="USA" />
    <field name="AREA_CODE" value="425" />
    <field name="PHONE" value="5551234" />
    <field name="DEPT" value="SALES" />
    <field name="JOB_TITLE" value="SALES ENINGEER" />
    <field name="LEAVE_DUE" value="20" />
    <field name="LEAVE_TAKEN" value="1" />
  </fields>
</CONNXdocument>
```

Create (This is for CREATE TABLE. It displays the meta data for the table)

```
<?xml version="1.0" encoding="UTF-8"?>
<CONNXdocument version="1"  committed="2015/02/05-16:23:33">
  <method>create</method>
  <tableName>EMPLOYEES</tableName>
  <fields>
    <field name="ISN_EMPLOYEES" type="integer" length="4" pre="0"
   scale="0" />
    <field name="PERSONNEL_ID" type="string" length="8" pre="0"
   scale="0" />
    <field name="FIRST_NAME" type="string" length="20" pre="0" scale="0"
   />
    <field name="NAME" type="string" length="20" pre="0" scale="0" />
    <field name="MIDDLE_NAME" type="string" length="20" pre="0"
   scale="0" />
    <field name="MAR_STAT" type="string" length="1" pre="0" scale="0" />
    <field name="SEX" type="string" length="1" pre="0" scale="0" />
    <field name="NBIRTH" type="date" length="6" pre="0" scale="0" />
    <field name="CITY" type="string" length="20" pre="0" scale="0" />
    <field name="POST_CODE" type="string" length="10" pre="0" scale="0"
   />
    <field name="COUNTRY" type="string" length="3" pre="0" scale="0" />
    <field name="AREA_CODE" type="string" length="6" pre="0" scale="0"
   />
    <field name="PHONE" type="string" length="15" pre="0" scale="0" />
    <field name="DEPT" type="string" length="6" pre="0" scale="0" />
    <field name="JOB_TITLE" type="string" length="25" pre="0" scale="0"
   />
    <field name="LEAVE_DUE" type="integer" length="4" pre="0" scale="0"
   />
    <field name="LEAVE_TAKEN" type="integer" length="4" pre="0"
   scale="0" />
  </fields>
</CONNXdocument>
```

Drop (This is for DROP TABLE.)

```
<?xml version="1.0" encoding="UTF-8"?>
<CONNXdocument version="1"  committed="2015/01/26-13:59:48">
  <method>drop</method>
  <tableName>EMPLOYEES</tableName>
</CONNXdocument>
```

Adding a Replication to the Event Replicator

Adding a Replication to the Open Systems Event Replicator includes:

- Opening a CONNX Data Dictionary (CDD)
- Adding a source table from the Adabas tables in the CDD
- Adding 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

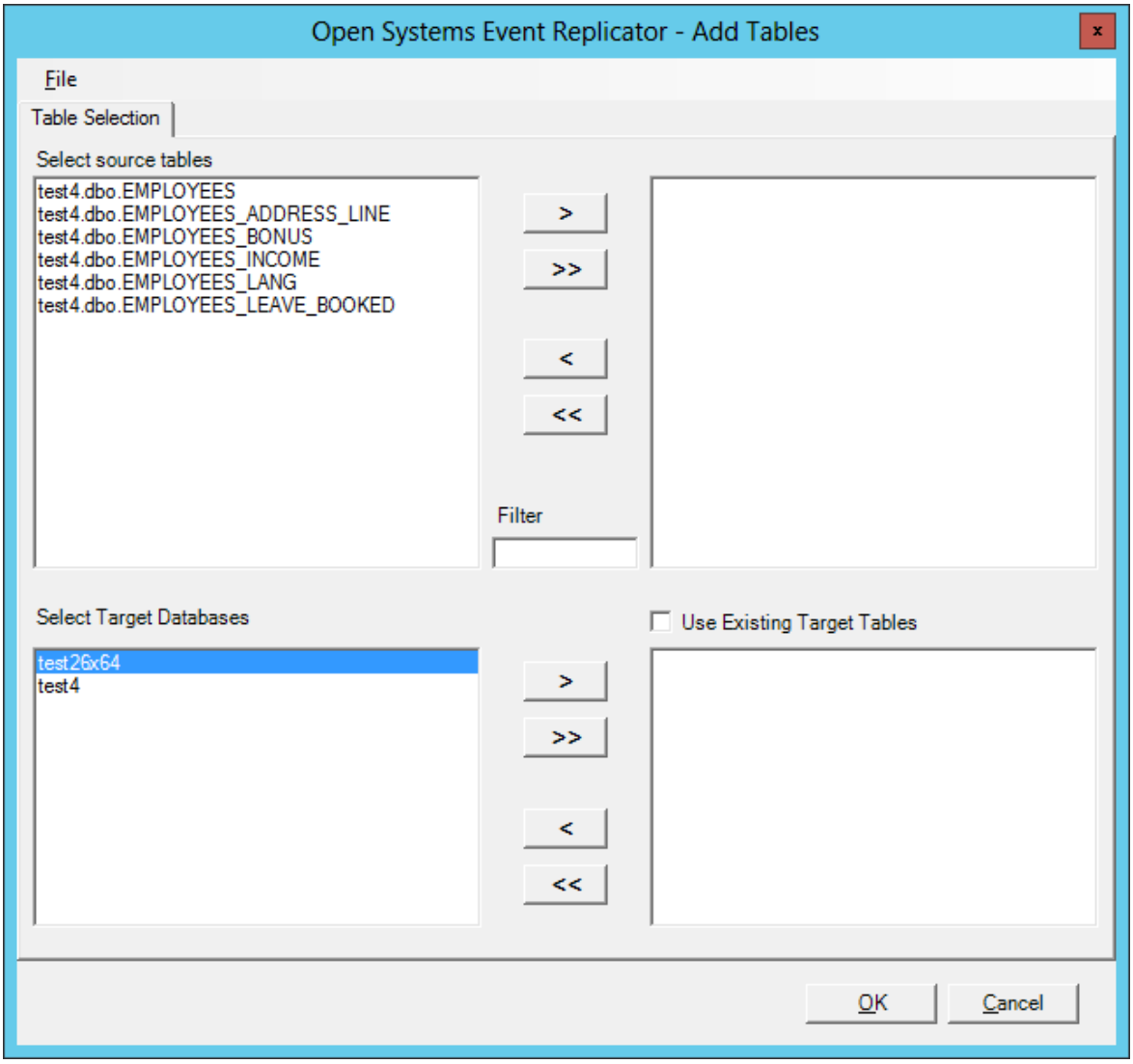
After you add a replication you will be able to:

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

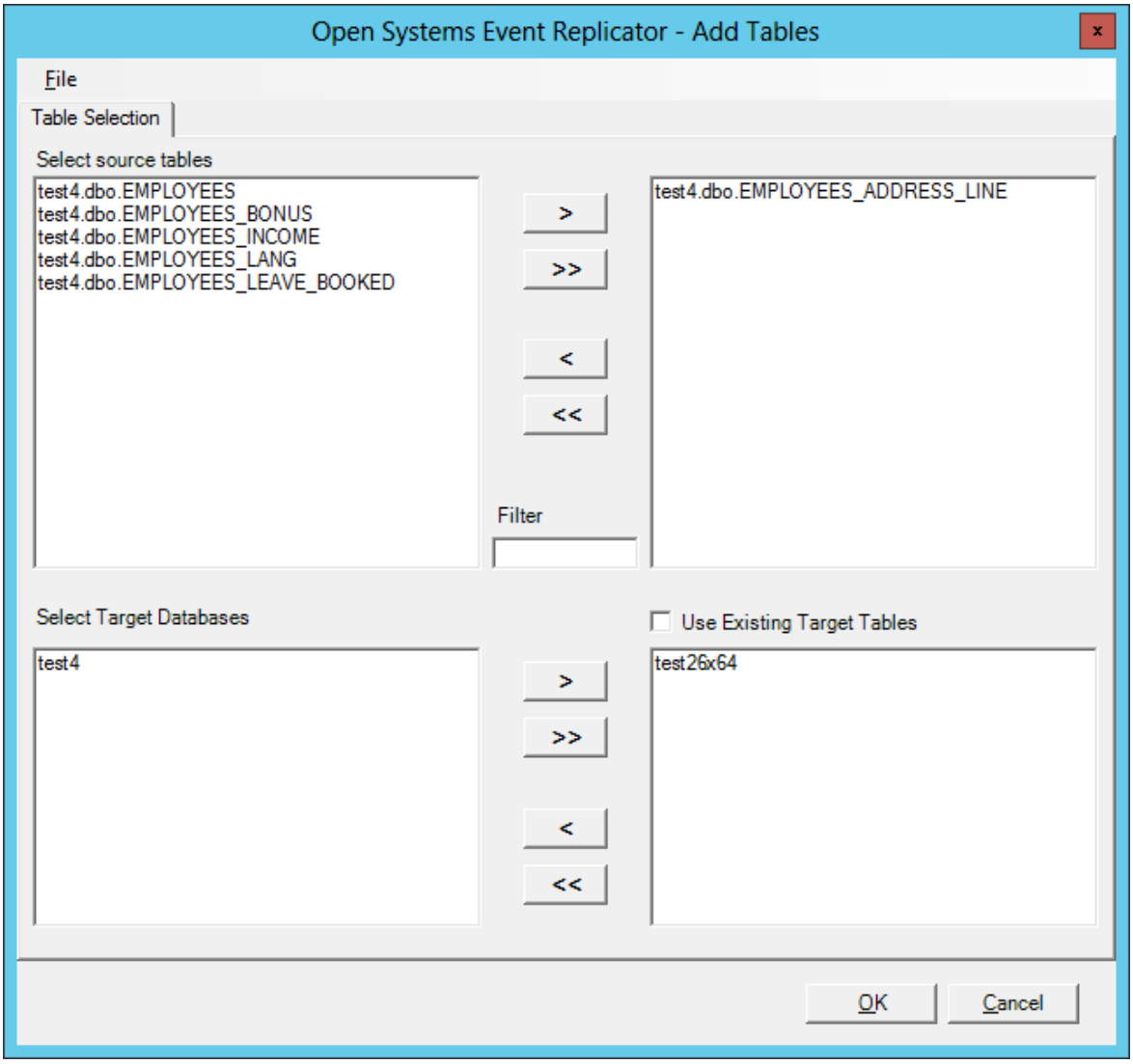
You can add additional source tables to the Event Replicator and associate them with a target database at the same time.

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 Open Systems Event Replicator CDD.
2. Click **Add Tables**. The **Add Tables** window appears.

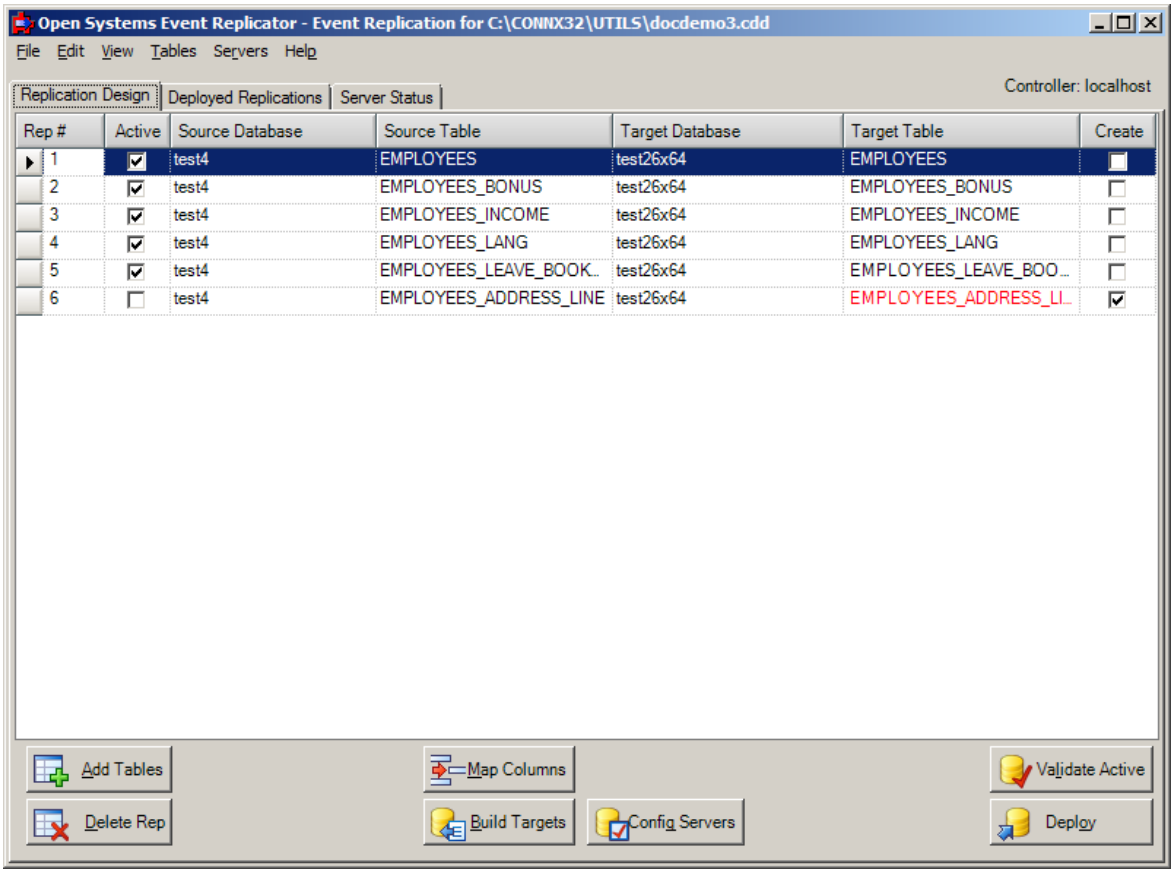


3. Select any additional source tables you wish to replicate and specify the target Database. By default the source tables will map to new target tables which exactly match the source tables to be created on deploy. Select the **Use Existing Target Tables** and the Replication Administrator will search 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.



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 **Open Systems Event Replication** window appears and the tables you added are in the list.



5. If you want to deploy the changes you've made, 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**.

For more information about **Validate Active**, see Validating Active Servers. For more information about **Deploy**, see Deploying the Event Replicator.

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:

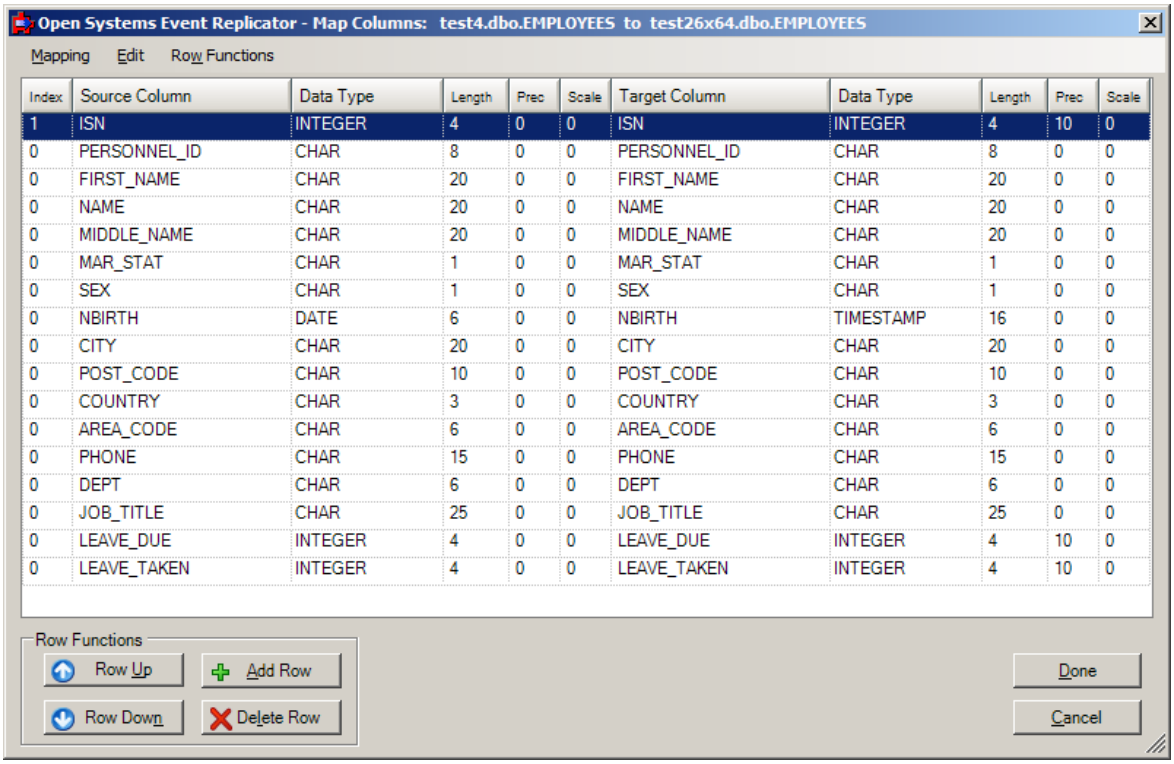
- Create the target table
- Deploy the replication (which will start replicating data from source database to target database)

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

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 the column source and the column attributes.

Warning: Since the Event Replicator cannot create Adabas Target Tables, you cannot map columns to a new Adabas Target Table. All Adabas Target Table changes must be done using Adabas, not the Event Replicator.

1. Open the Open Systems Event Replicator CDD.
2. Select a Target Table to view. Click **Map Columns**. The **Map Columns** window appears.



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

- 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 Open Systems Event Replicator 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)

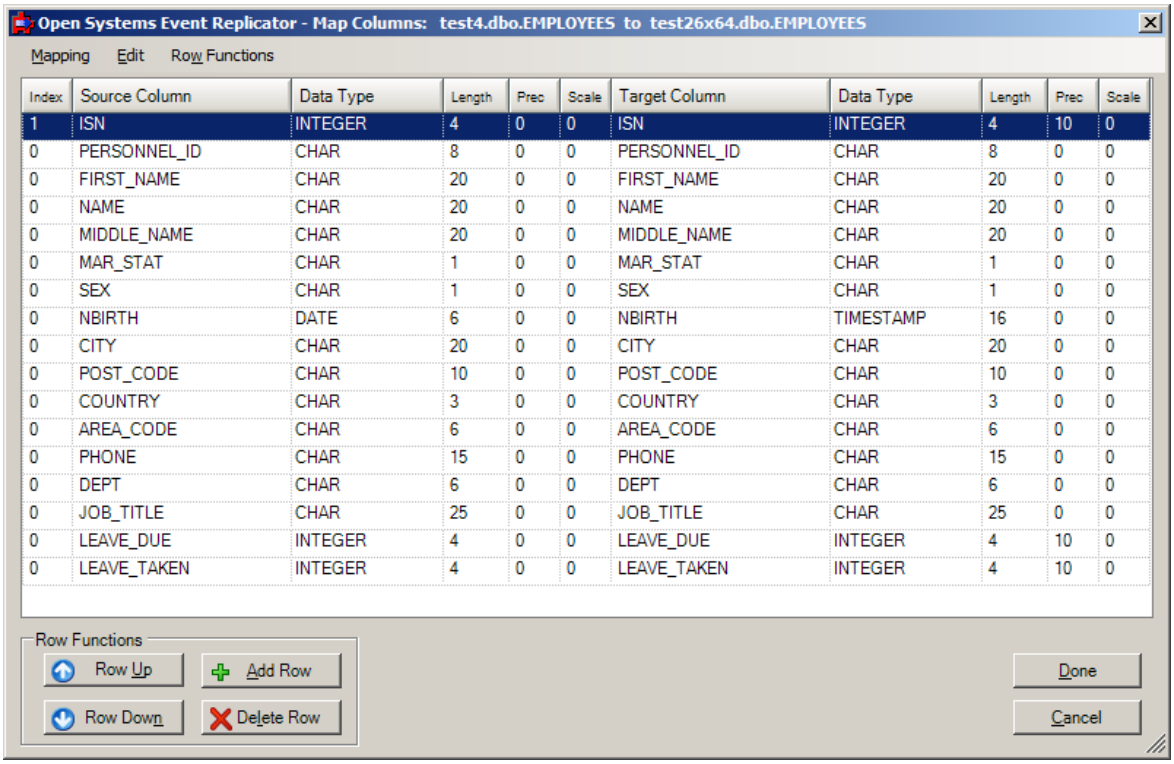
A target table is considered to be "new" if it does not exist on the target data base and it has not been created or deployed.

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.

Warning: Since the Event Replicator cannot create Adabas Target Tables, you cannot map columns to a new Adabas Target Table. All Adabas Target Table changes must be done using Adabas, not the Event Replicator.

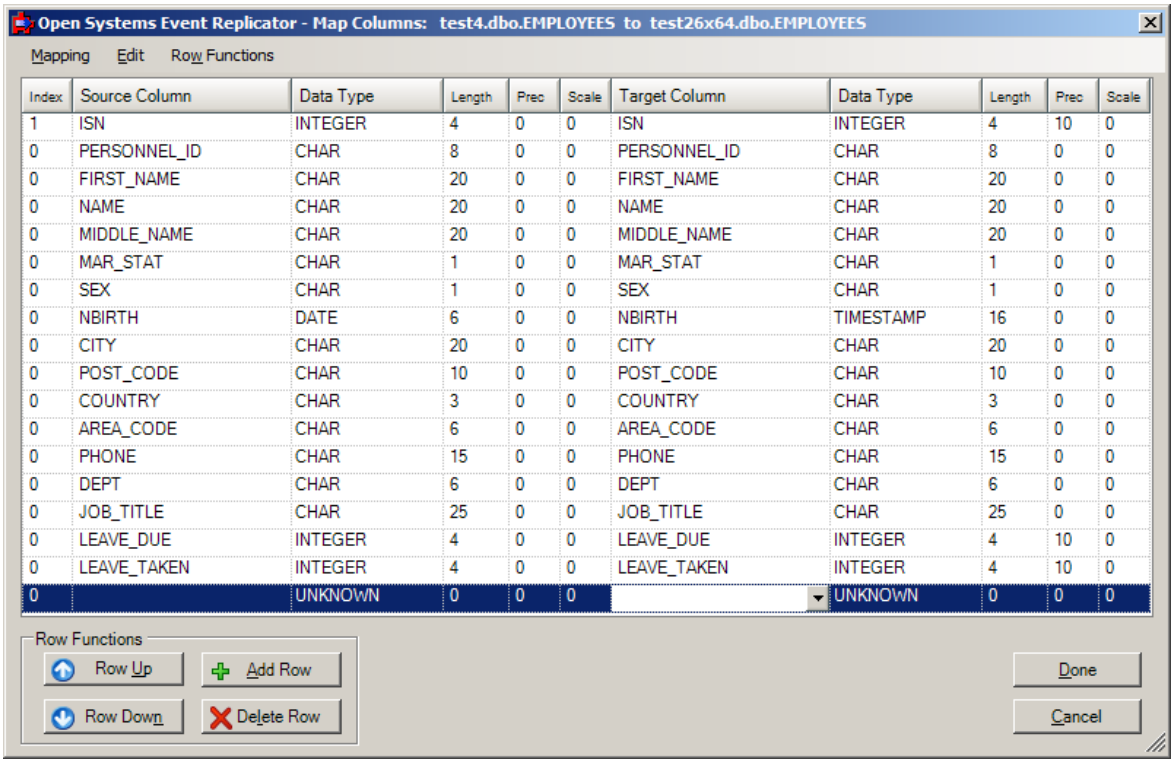
Note: Column mapping does not change the structure of the Source table or an existing Target table. The Add Row, Delete Row, Row Up and Row Down buttons are only available when mapping to a new target table. If you are mapping to an existing target table and wish to change the structure of that table, 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 Open Systems Event Replicator CDD.
2. Select a Target Table to add rows to. Click **Map Columns**. The **Map Columns** window appears.



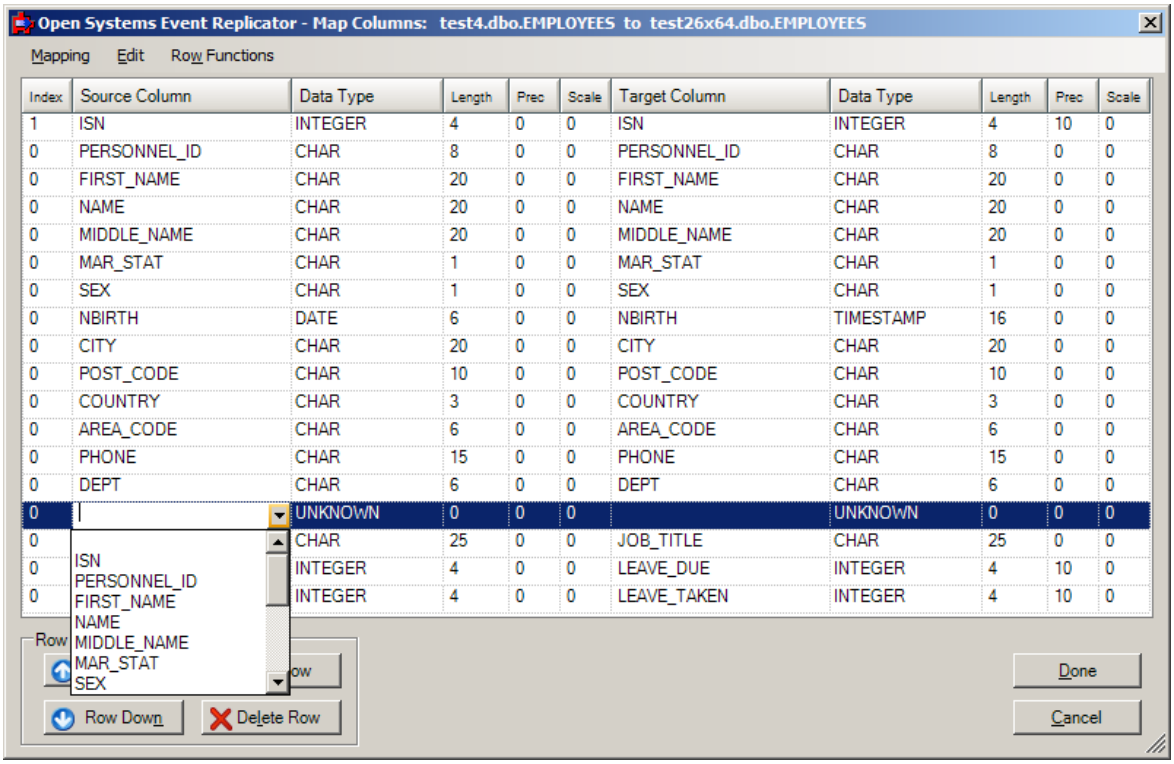
Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

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.

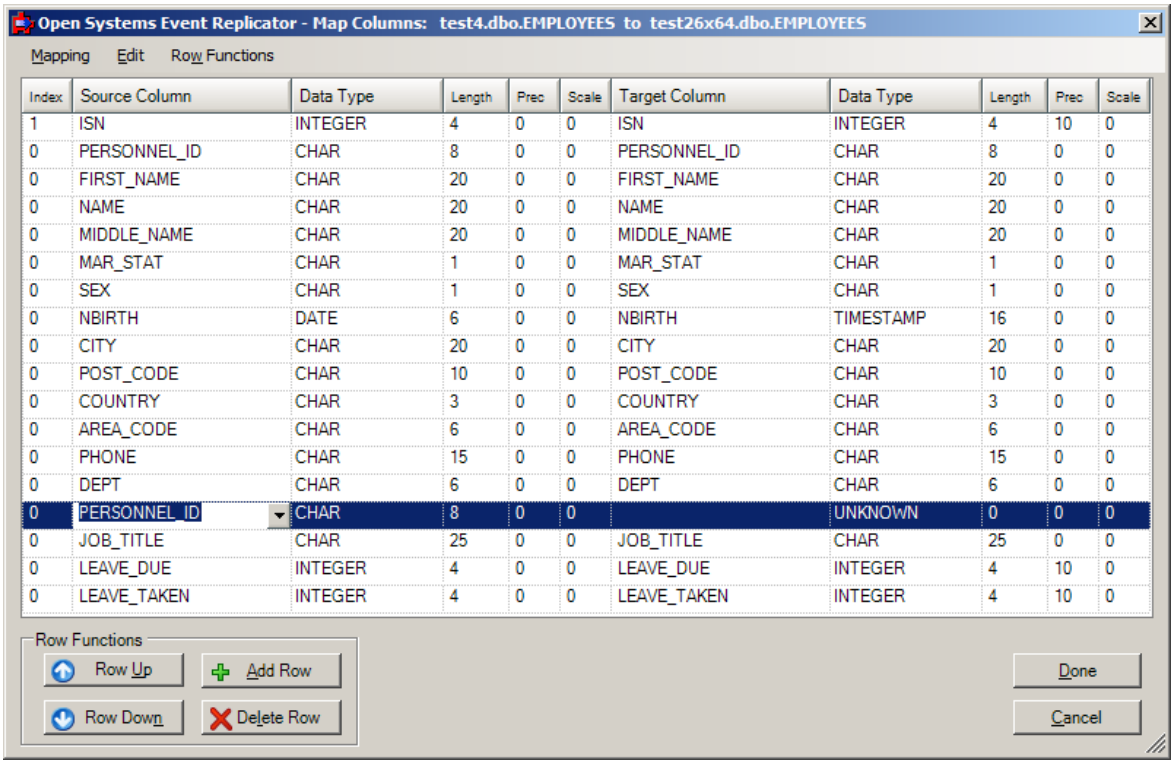


Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0
0		UNKNOWN	0	0	0		UNKNOWN	0	0	0

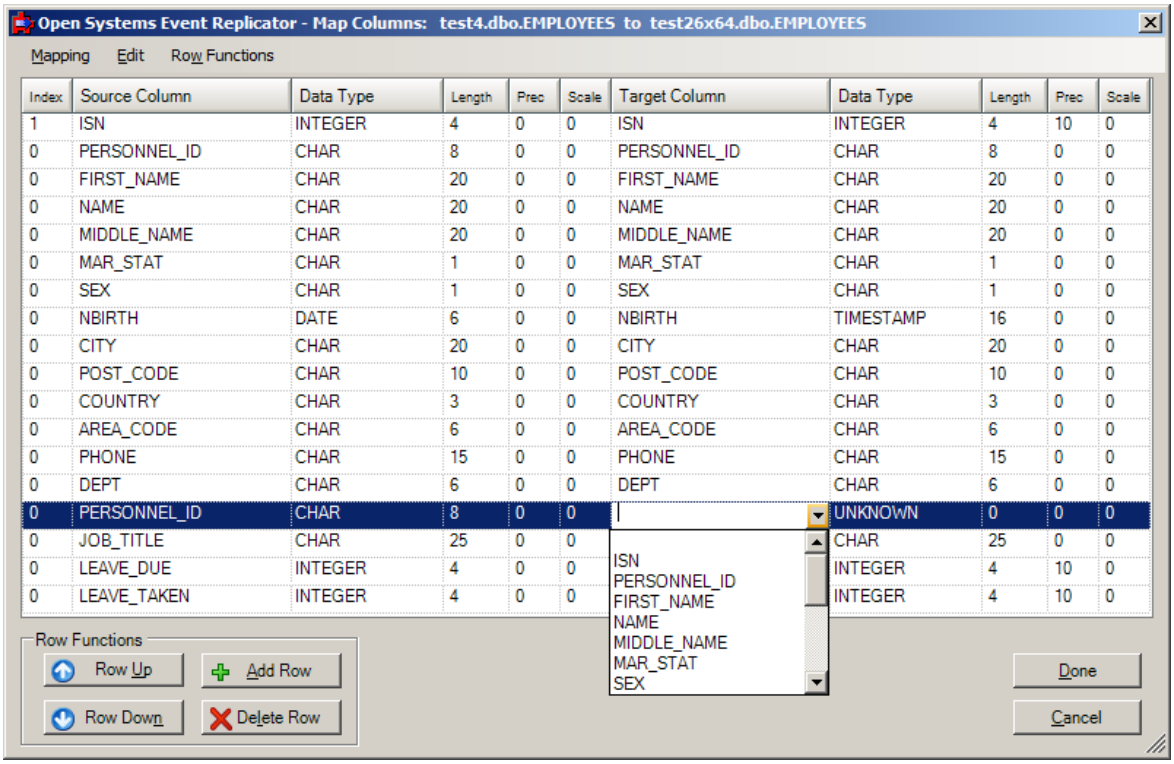
   b. Use **Row Up** or **Row Down** to place the row in the correct position.
   c. Click the **Source Column** box for the row you have added. A drop down arrow appears. Click on the arrow and a list of available source table columns appears.



d. Select the source table column to be added.

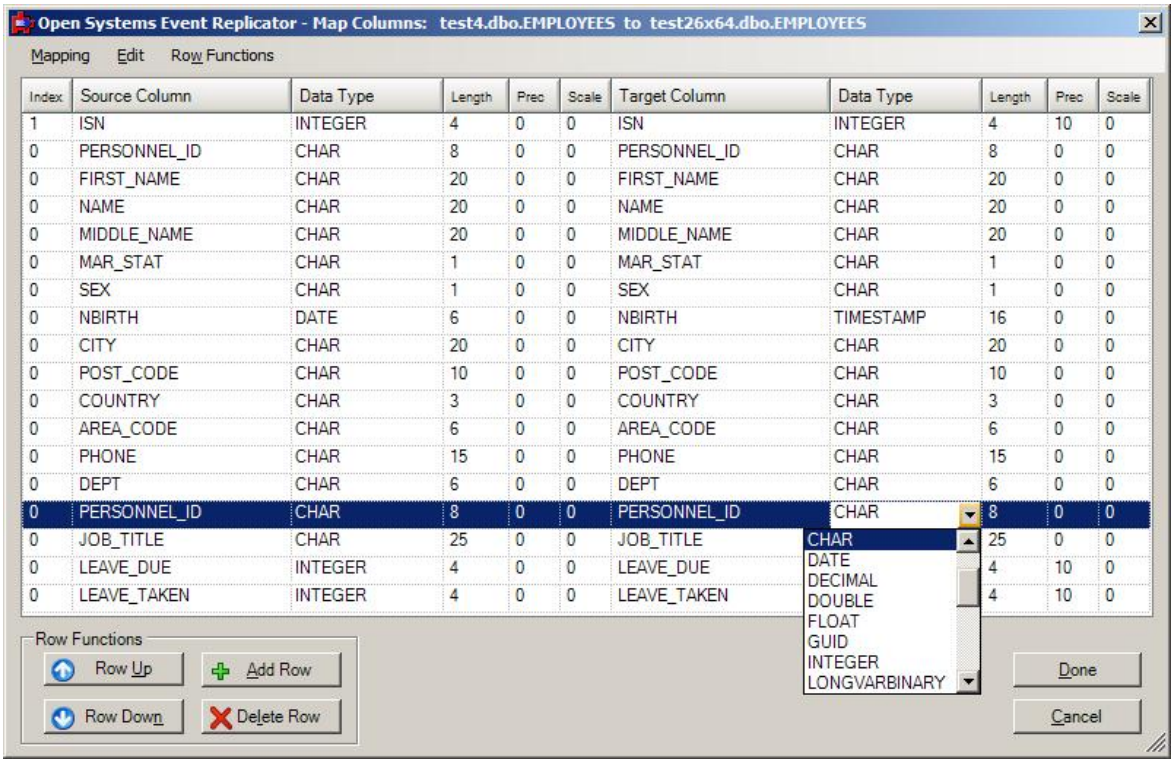


e. Click the **Target Column** box for the row you have added. A drop down arrow appears. Click on the arrow and a list of available target table columns appears



f. Select a target column name from this list or enter another target column name in **Target Column**.

g. Select the desired **Data Type** of the target column and fill in the **Length**, **Precision** and **Scale**.



Note: The ISN field is always shown in the column map because replication requires a unique key present to work and if there are no other unique keys on the table the ISN can then be used. The ISN is shown even if the table was built with a Create Table statement that did not specify an ISN.

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

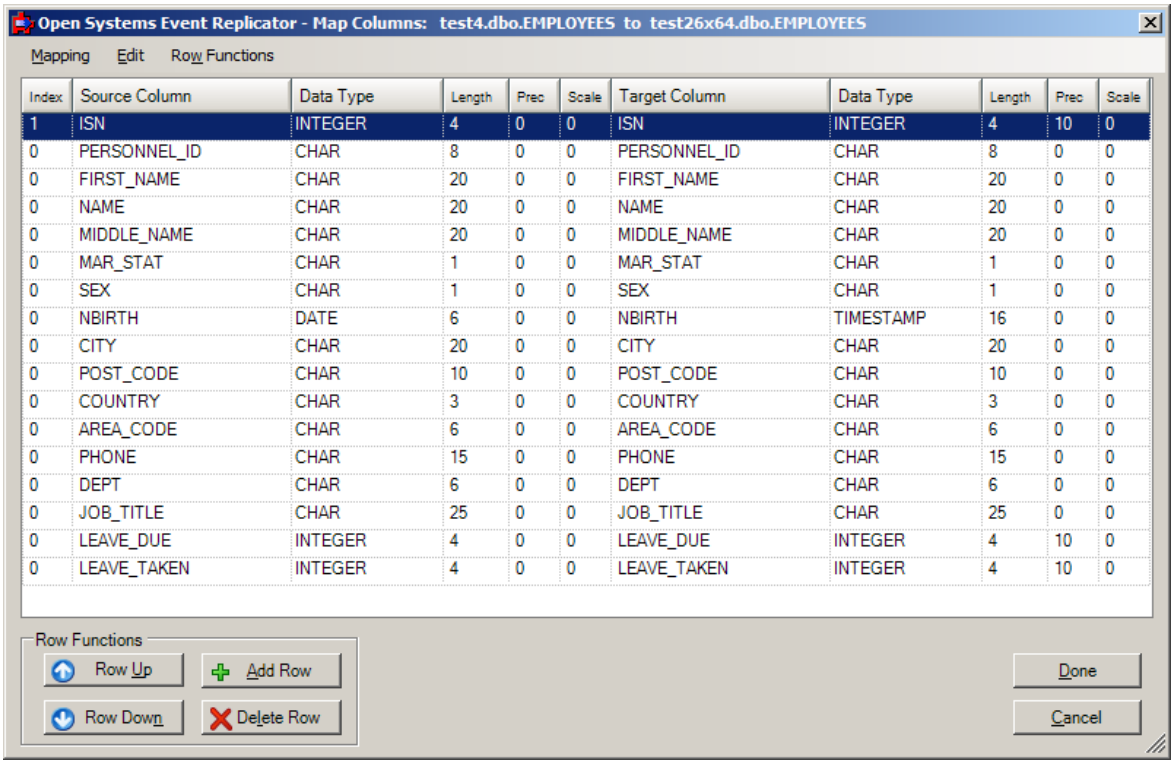
After you delete a new column mappings you will be able to:

- 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 data base and it has not been created or deployed.

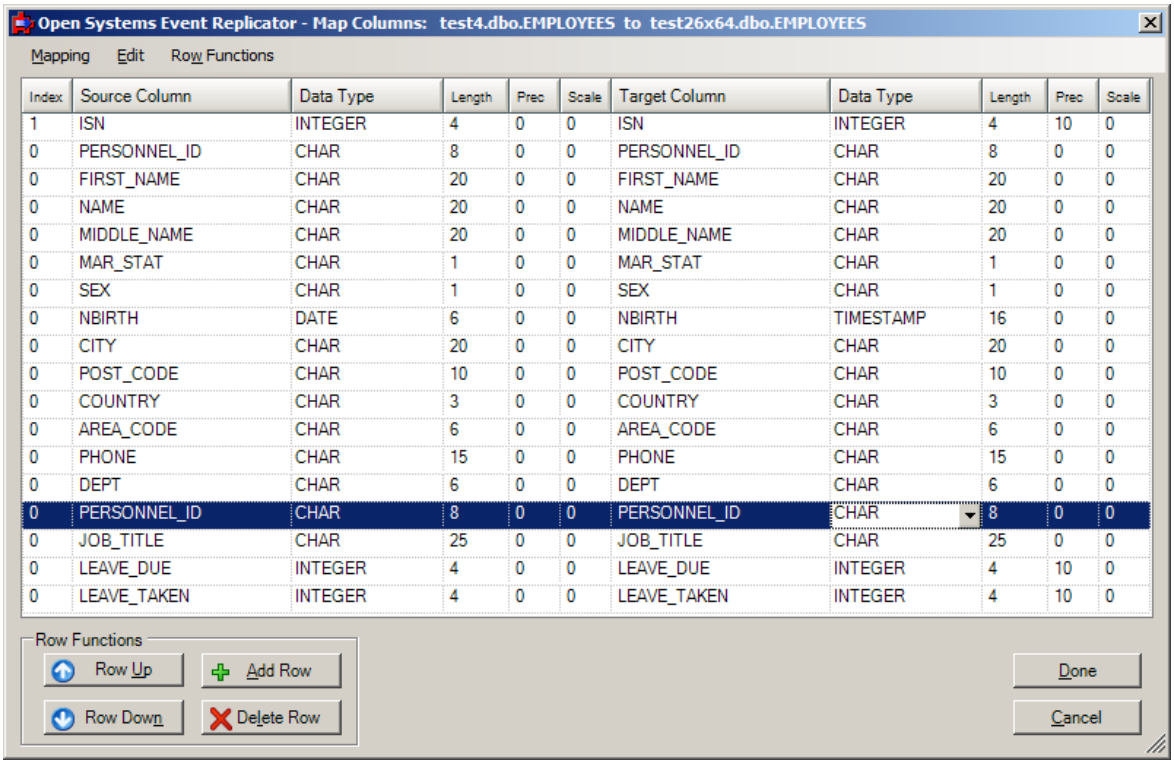
Note: Deleting a column map does not change the structure of the Source table or an existing Target table. The Add Row, Delete Row, Row Up and Row Down buttons are only available when mapping to a new target table. If you are mapping to an existing target table and wish to change the structure of that 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 Open Systems Event Replicator CDD.
2. Select a non-Adabas Target Table. Click **Map Columns**. The **Map Columns** window appears.



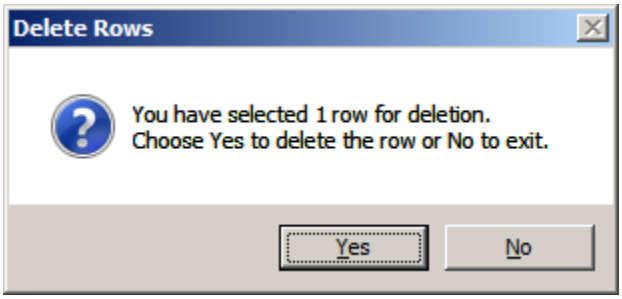
Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

3. To delete a row from the Target Table:
   a. Select the row to be deleted.

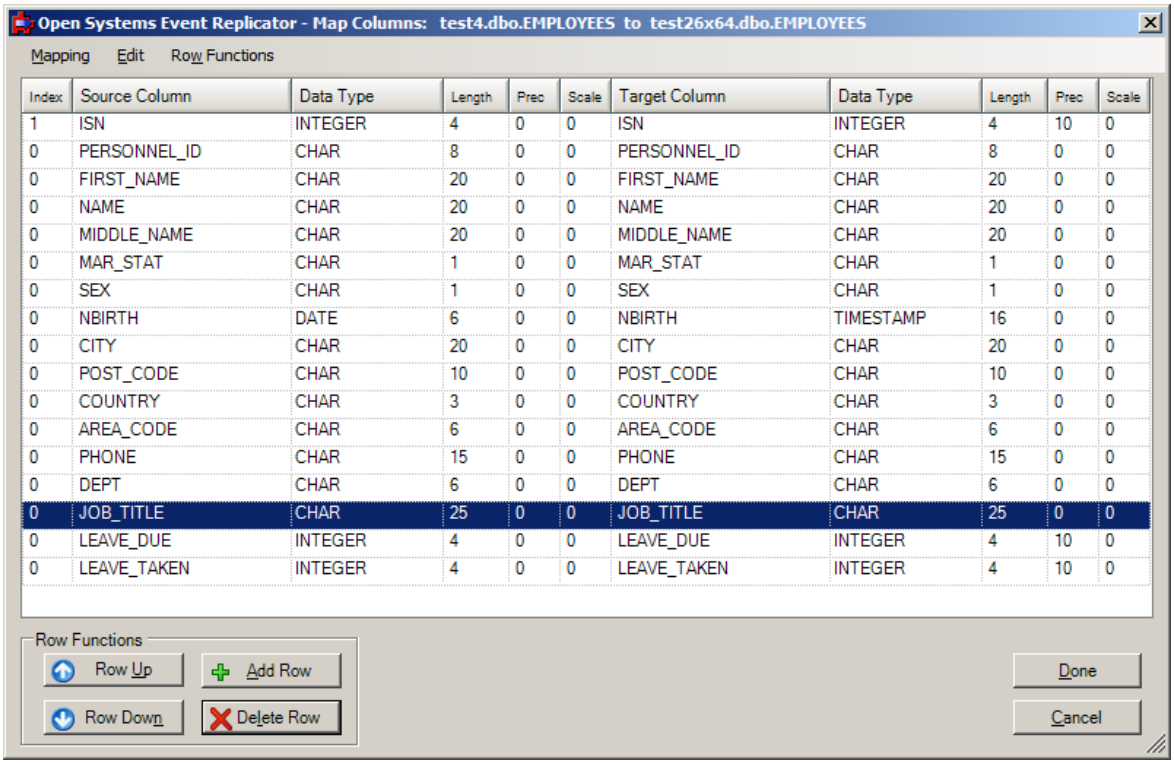


Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

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



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



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

Note: You can not use the Open Systems Event Replicator to delete a column map from the Source Table.

4. Click **Done**. The **Open Systems Event Replication** 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)

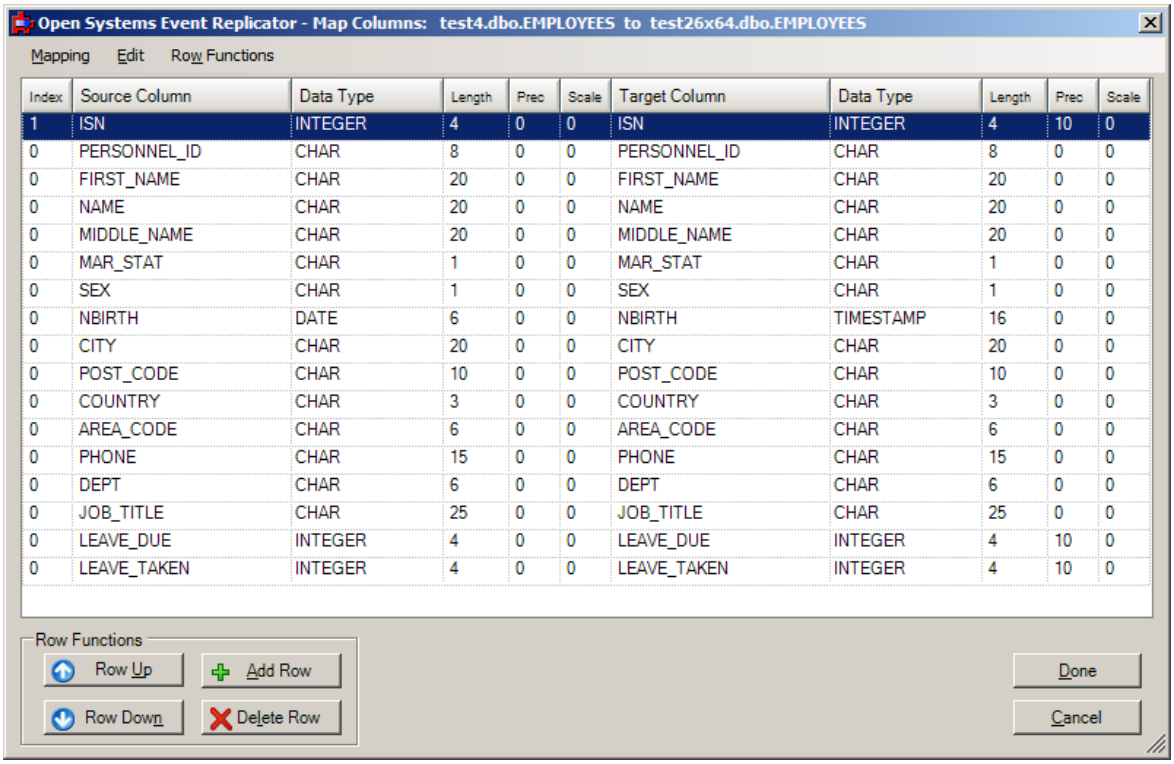
A target table is considered to be "new" if it does not exist on the target data base and it has not been created or deployed.

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 Open Systems Event Replicator 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.**

Warning: Since the Event Replicator cannot create Adabas Target Tables, you cannot change the row order in a Adabas Target Table. All Adabas Target Table changes must be done using Adabas, not the Event Replicator.

1. Open the Open Systems Event Replicator CDD.
2. Select a new Target Table to change. Click **Map Columns**. The **Map Columns** window appears.



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

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

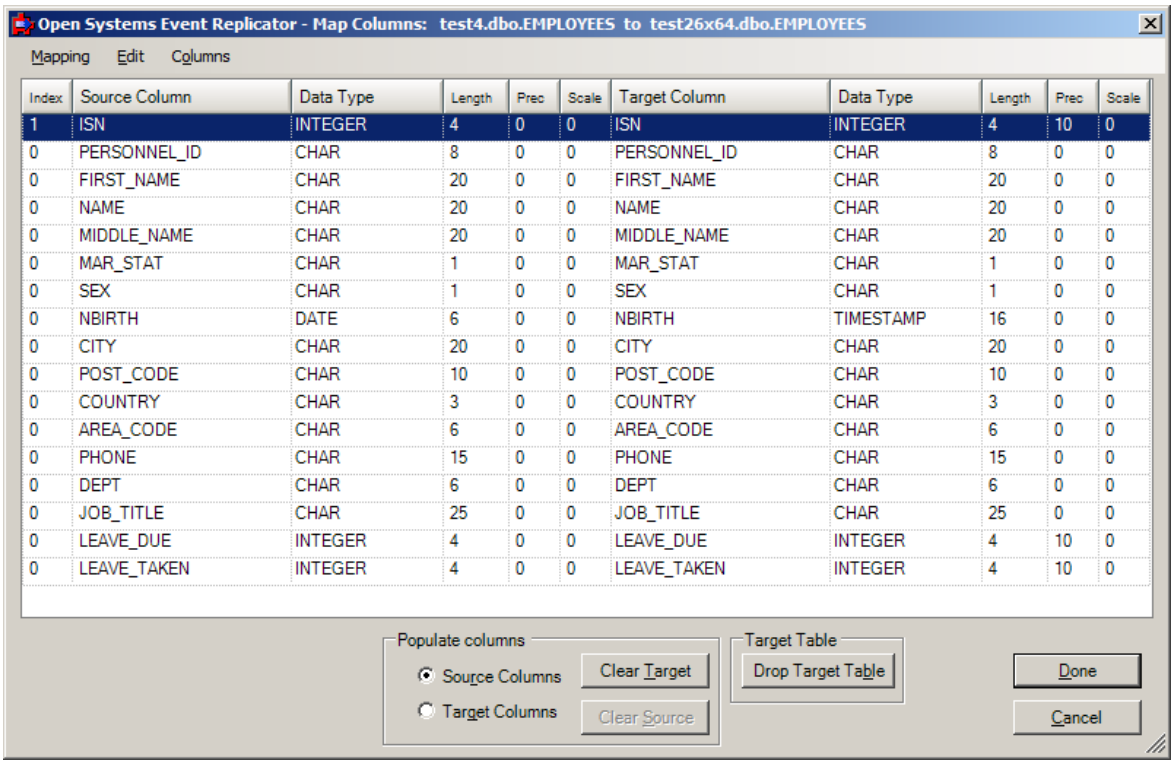
After you change the source mappings you will be able to:

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

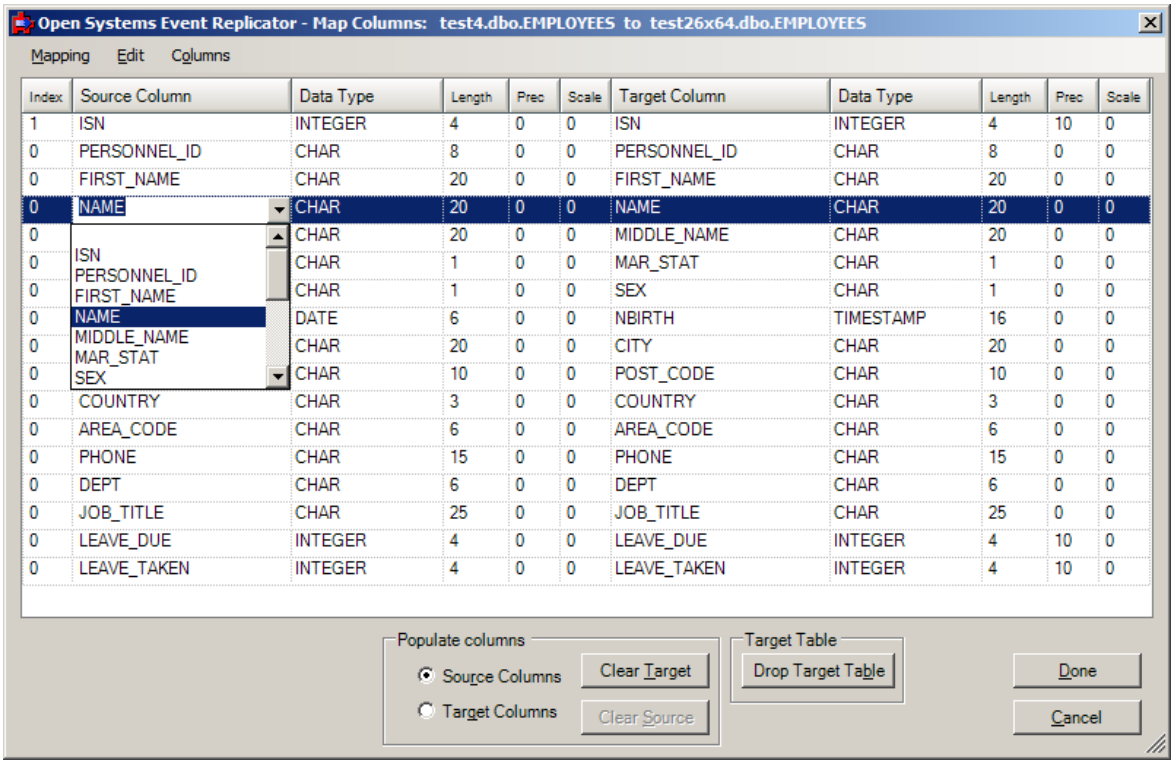
Map Columns allow you to change which source column an existing target column is mapped to.

Note: The Source and Target columns must have compatible data types, lengths, precisions and scales.

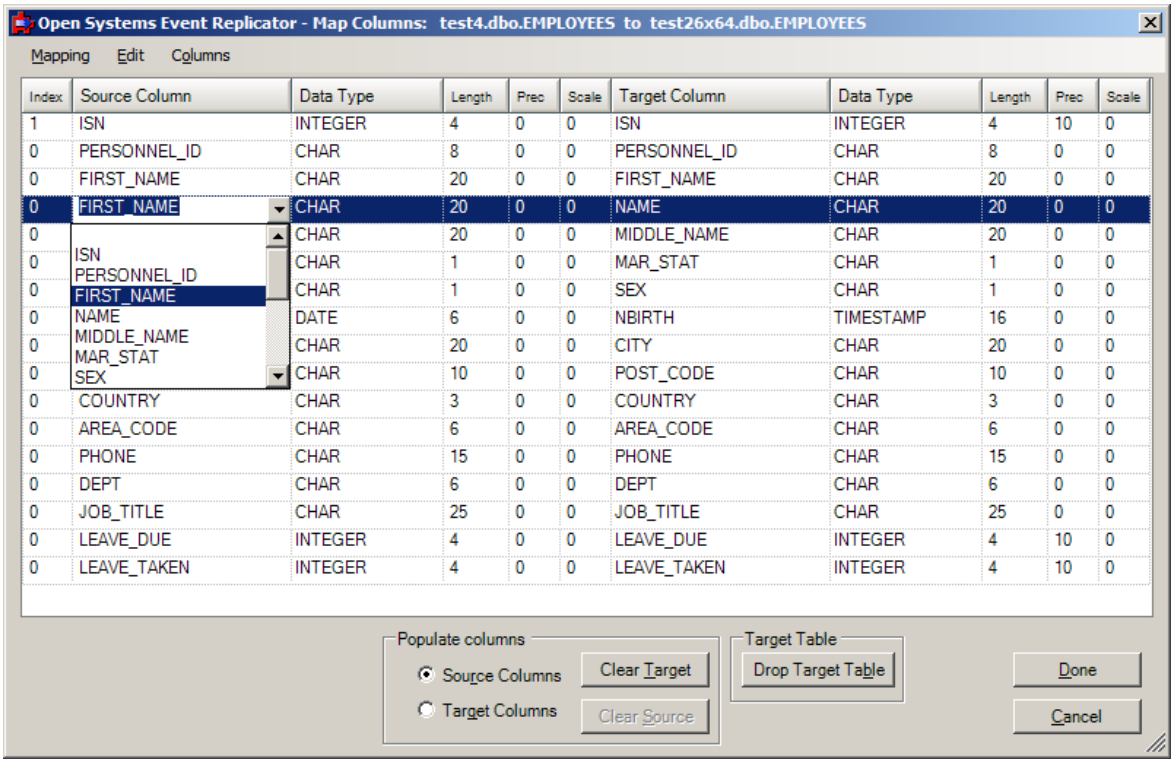
1. Open the Open Systems Event Replicator CDD.
2. Select a Target Table to modify. Click **Map Columns**. The **Map Columns** window for an existing target table appears.



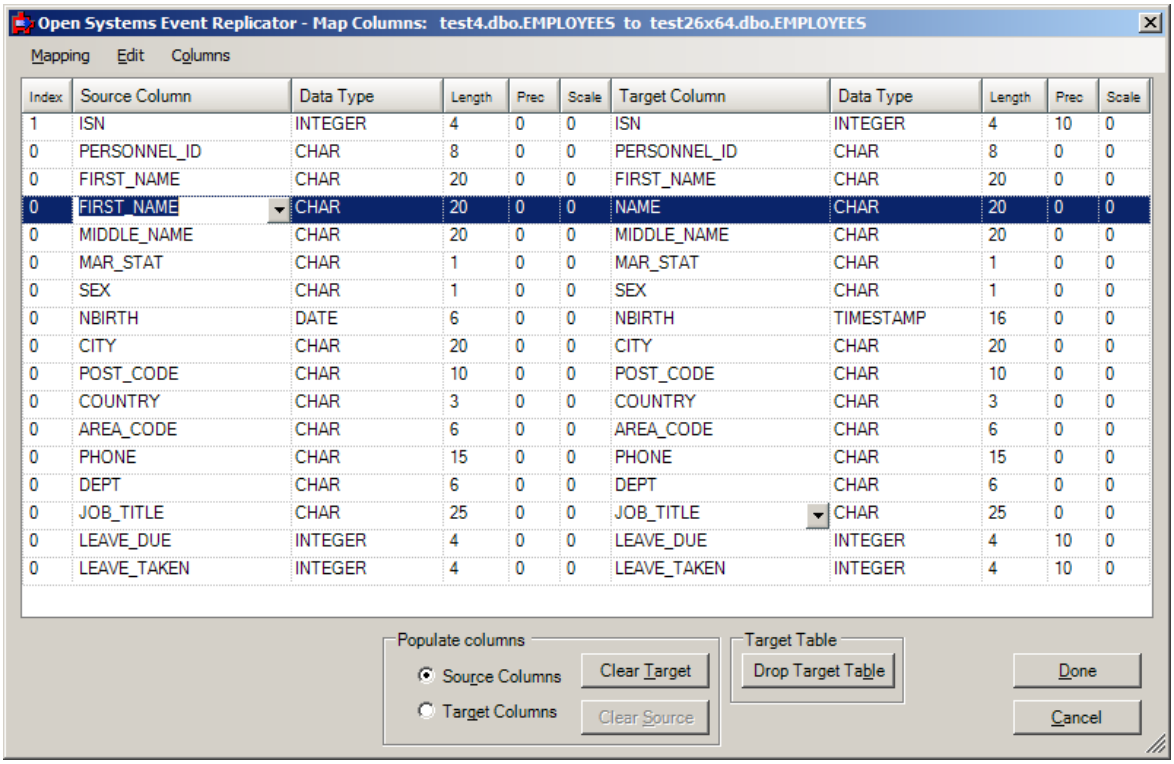
3. Under **Populate columns**, click **Source Column**. Click the **Source Column** box for the replication you wish to change. A drop down arrow appears. Click on the arrow and a list of available source column names appears.



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



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



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	FIRST_NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

6. Click **Done**. The **Open Systems Event Replication** window reappears.

Note: The ISN field is always shown in the column map because replication requires a unique key present to work and if there are no other unique keys on the table the ISN can then be used. The ISN is shown even if the table was built with a Create Table statement that did not specify an ISN.

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:

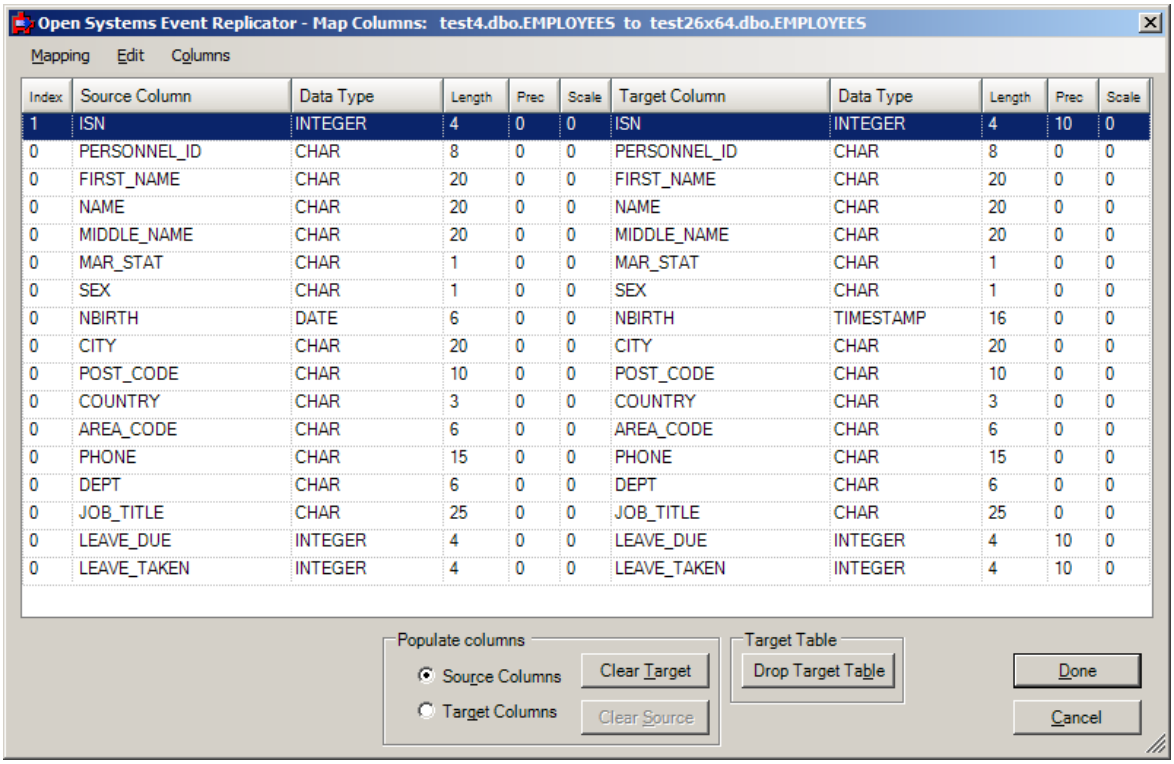
- Deploy the replication (which will start replicating data from source database to target database)

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, lengths, precisions and scales.

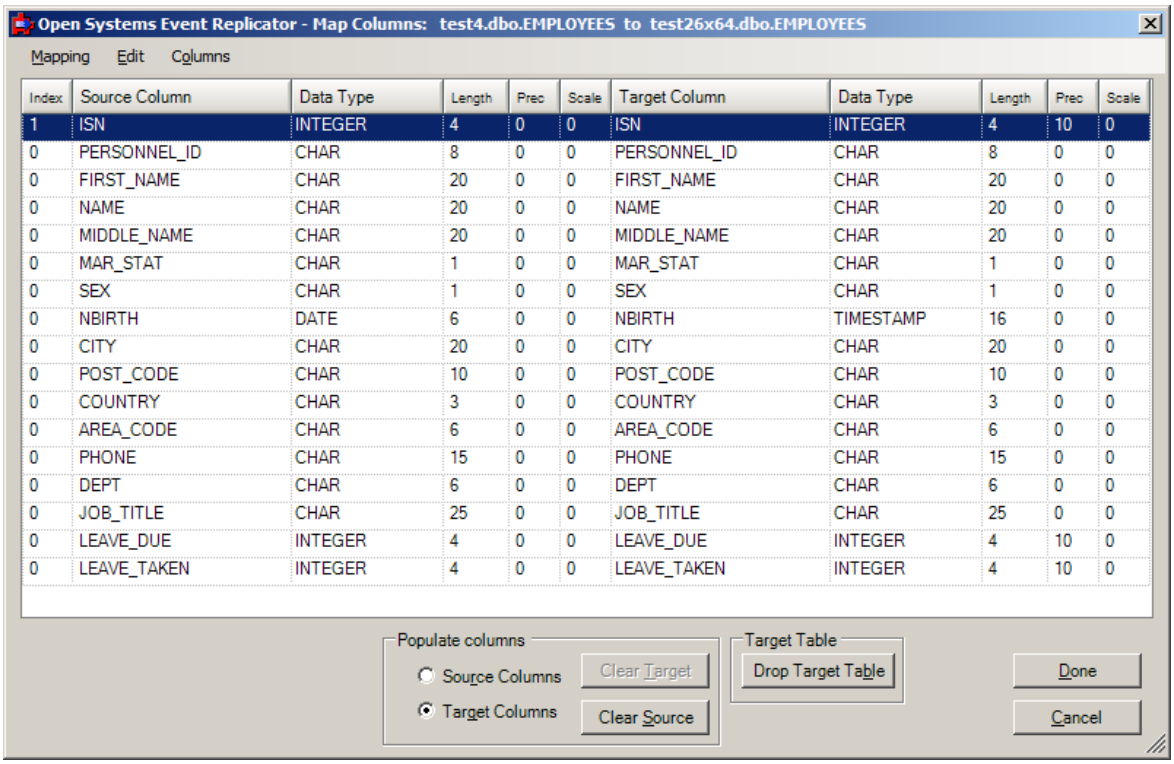
Note: You can not use the Open Systems Event Replicator to delete a row from the Source Table.

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.

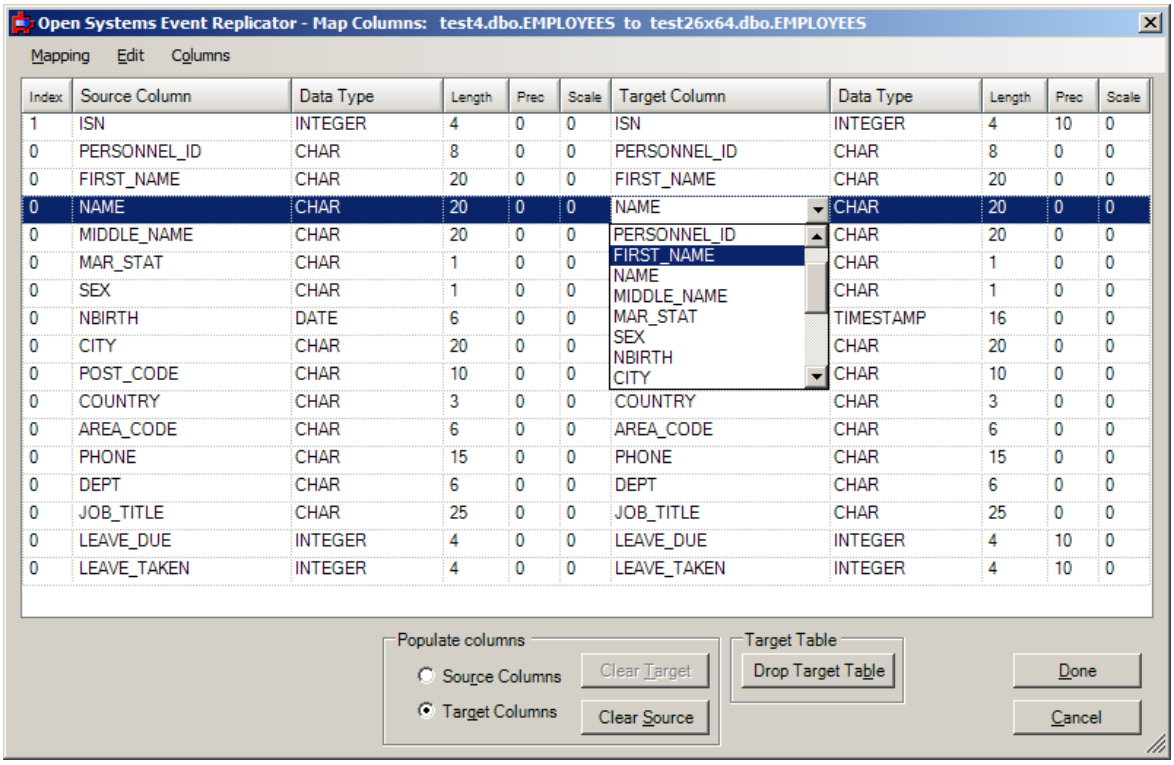


Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

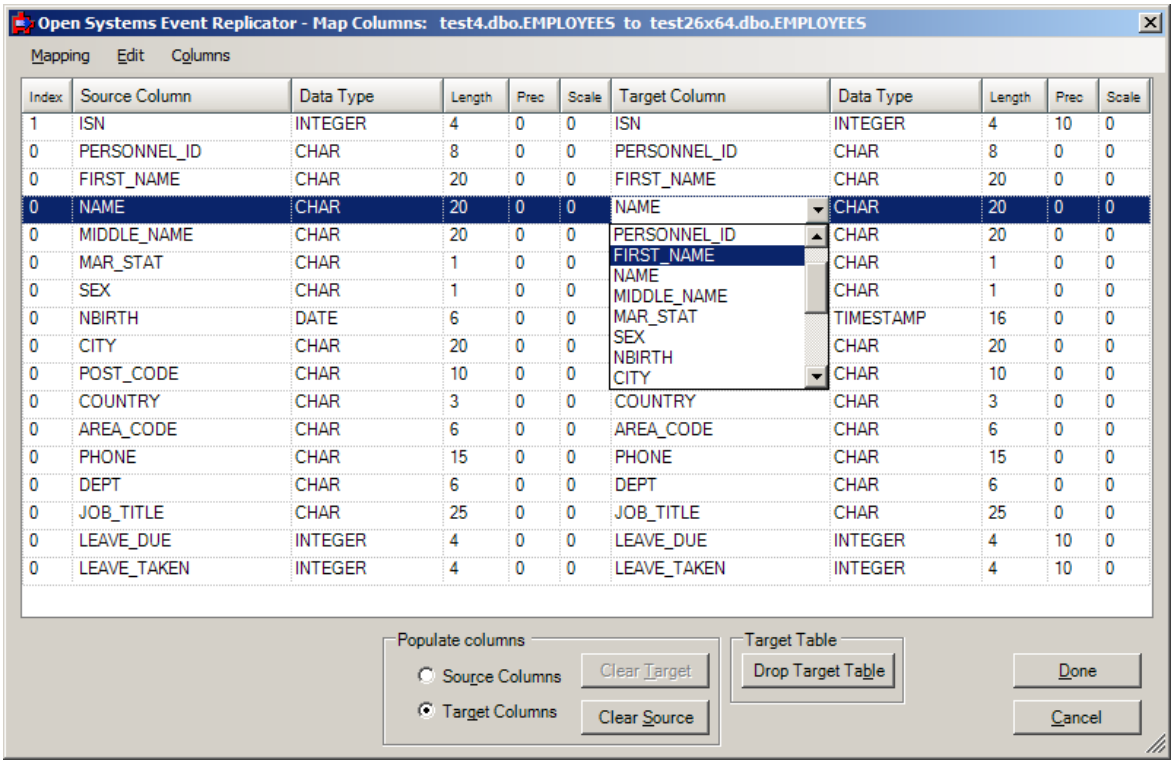
3. Under **Populate columns**, click **Target Columns**.



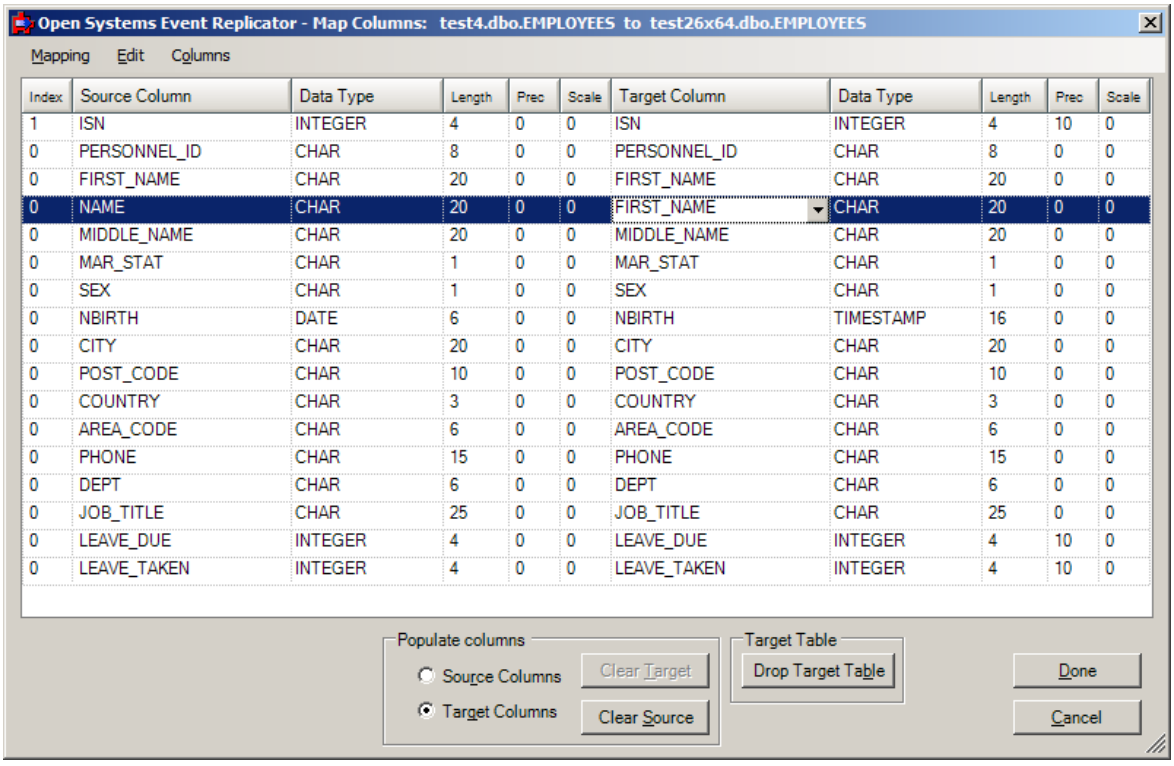
4. Click the **Target Column** box for the replication you wish to change. A drop down arrow appears. Click on the arrow and a list of available source column names appears.



5. Select the target column name to map the corresponding target column to.



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



7. Click **Done**. The **Open Systems Event Replication** window reappears.

   Note: The ISN field is always shown in the column map because replication requires a unique key present to work and if there are no other unique keys on the table the ISN can then be used. The ISN is shown even if the table was built with a Create Table statement that did not specify an ISN.

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

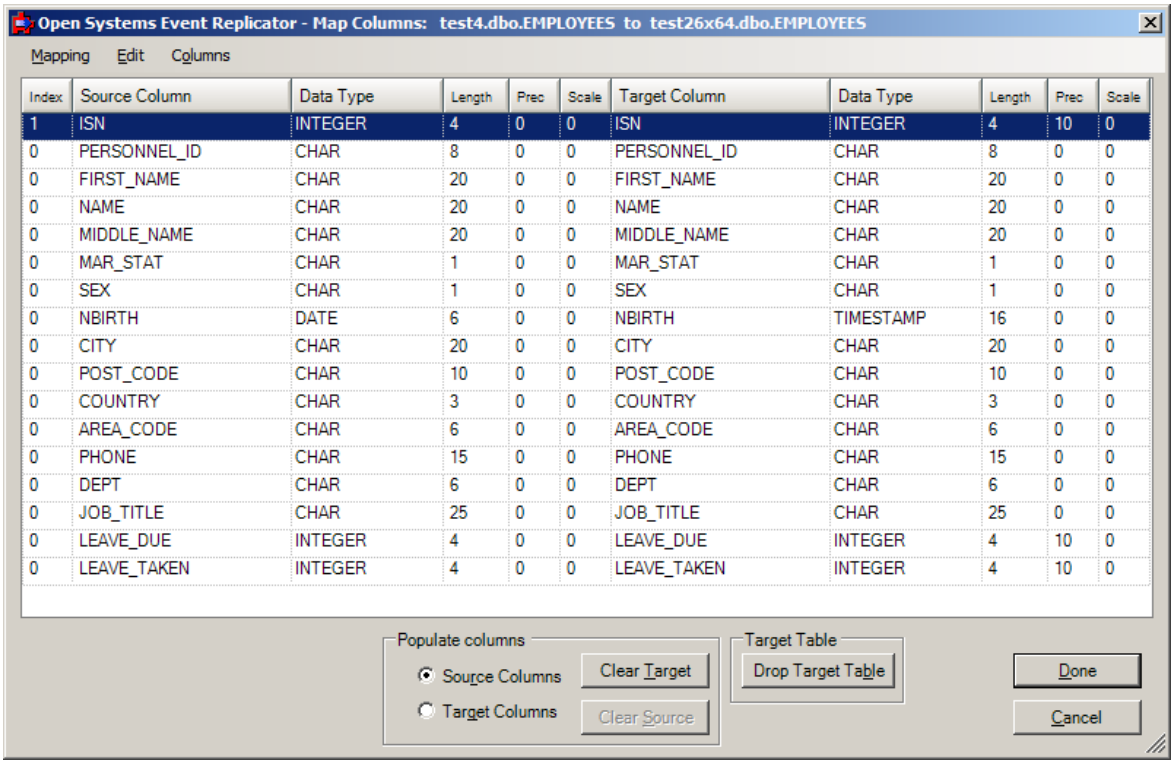
After you clear the source mappings you will be able to:

- 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.

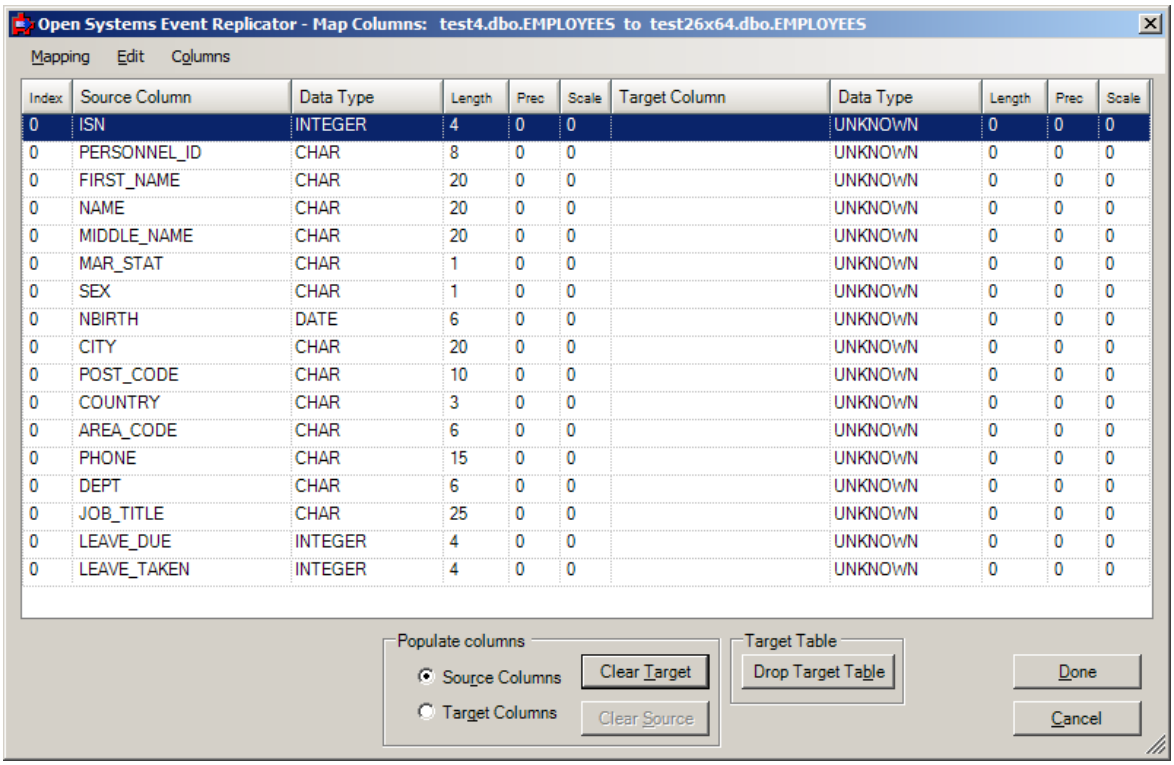
1. Open the Open Systems Event Replicator CDD.
2. Select a Target Table to modify. Click **Map Columns**. The **Map Columns** window for an existing target table appears.



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

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

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



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:

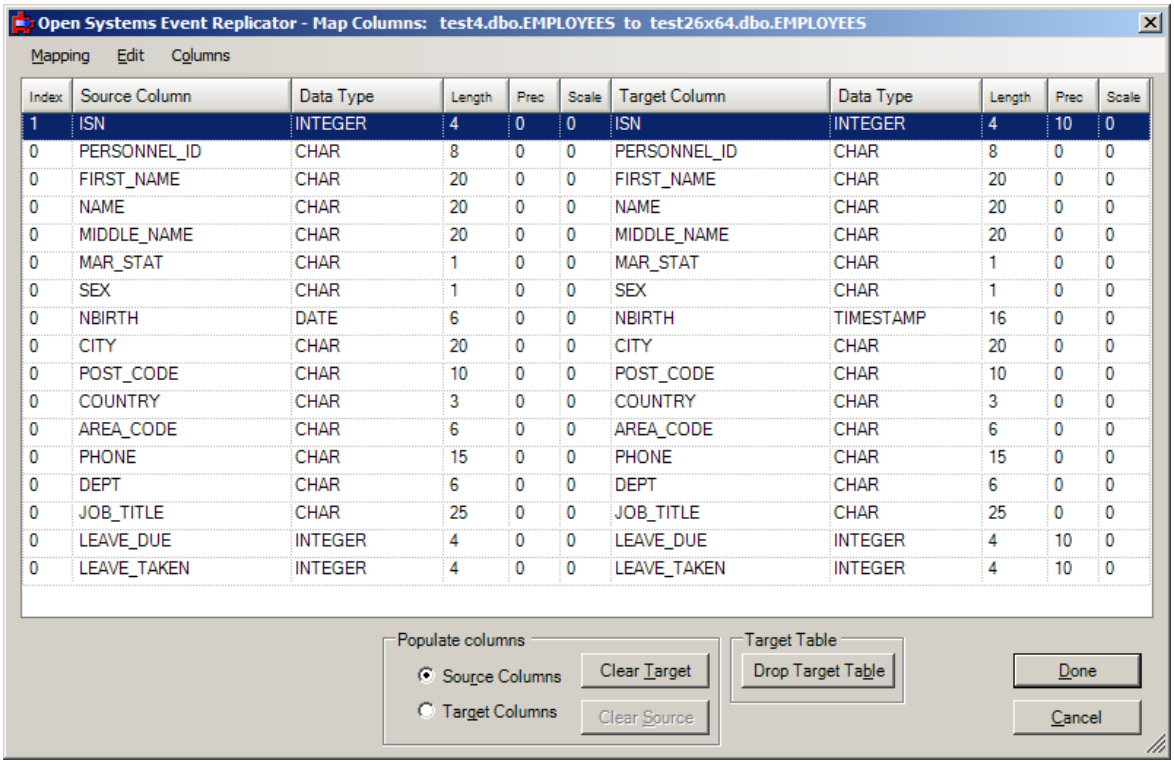
- The default source column mappings are incorrect

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)

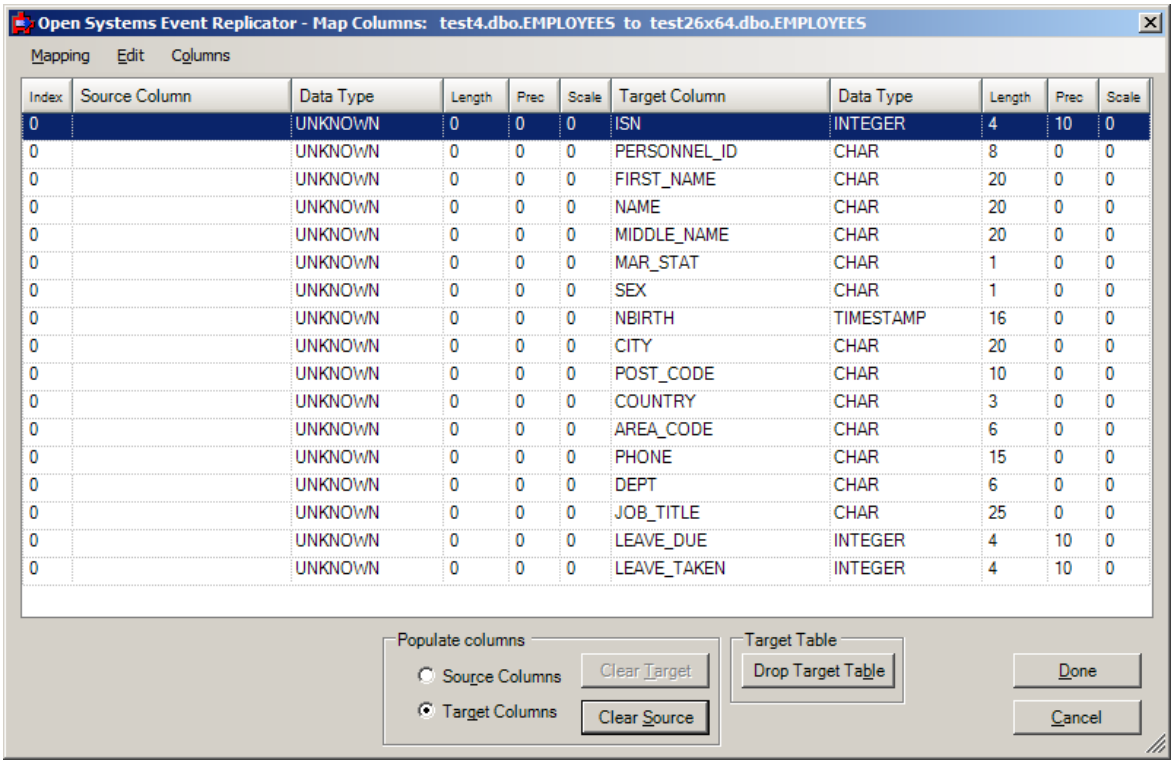
If you wish to change most of the target column mapping, you may find it easier to start with blank Source column names. The column names will still appear in the drop-down list.

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.



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN	INTEGER	4	0	0	ISN	INTEGER	4	10	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	CHAR	20	0	0	FIRST_NAME	CHAR	20	0	0
0	NAME	CHAR	20	0	0	NAME	CHAR	20	0	0
0	MIDDLE_NAME	CHAR	20	0	0	MIDDLE_NAME	CHAR	20	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	NBIRTH	DATE	6	0	0	NBIRTH	TIMESTAMP	16	0	0
0	CITY	CHAR	20	0	0	CITY	CHAR	20	0	0
0	POST_CODE	CHAR	10	0	0	POST_CODE	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	CHAR	25	0	0	JOB_TITLE	CHAR	25	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	10	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	10	0

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


Open Systems Event Replicator - Map Columns: test4.dbo.EMPLOYEES to test26x64.dbo.EMPLOYEES
Mapping
Edit
Columns
Index | Source Column | Data Type | Length | Prec | Scale | Target Column | Data Type | Length | Prec | Scale
0 | | UNKNOWN | 0 | 0 | 0 | ISN | INTEGER | 4 | 10 | 0
0 | | UNKNOWN | 0 | 0 | 0 | PERSONNEL_ID | CHAR | 8 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | FIRST_NAME | CHAR | 20 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | NAME | CHAR | 20 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | MIDDLE_NAME | CHAR | 20 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | MAR_STAT | CHAR | 1 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | SEX | CHAR | 1 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | NBIRTH | TIMESTAMP | 16 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | CITY | CHAR | 20 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | POST_CODE | CHAR | 10 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | COUNTRY | CHAR | 3 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | AREA_CODE | CHAR | 6 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | PHONE | CHAR | 15 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | DEPT | CHAR | 6 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | JOB_TITLE | CHAR | 25 | 0 | 0
0 | | UNKNOWN | 0 | 0 | 0 | LEAVE_DUE | INTEGER | 4 | 10 | 0
0 | | UNKNOWN | 0 | 0 | 0 | LEAVE_TAKEN | INTEGER | 4 | 10 | 0
Populate columns
Source Columns
Target Columns
Clear Target
Clear Source
Target Table
Drop Target Table
Done
Cancel

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:

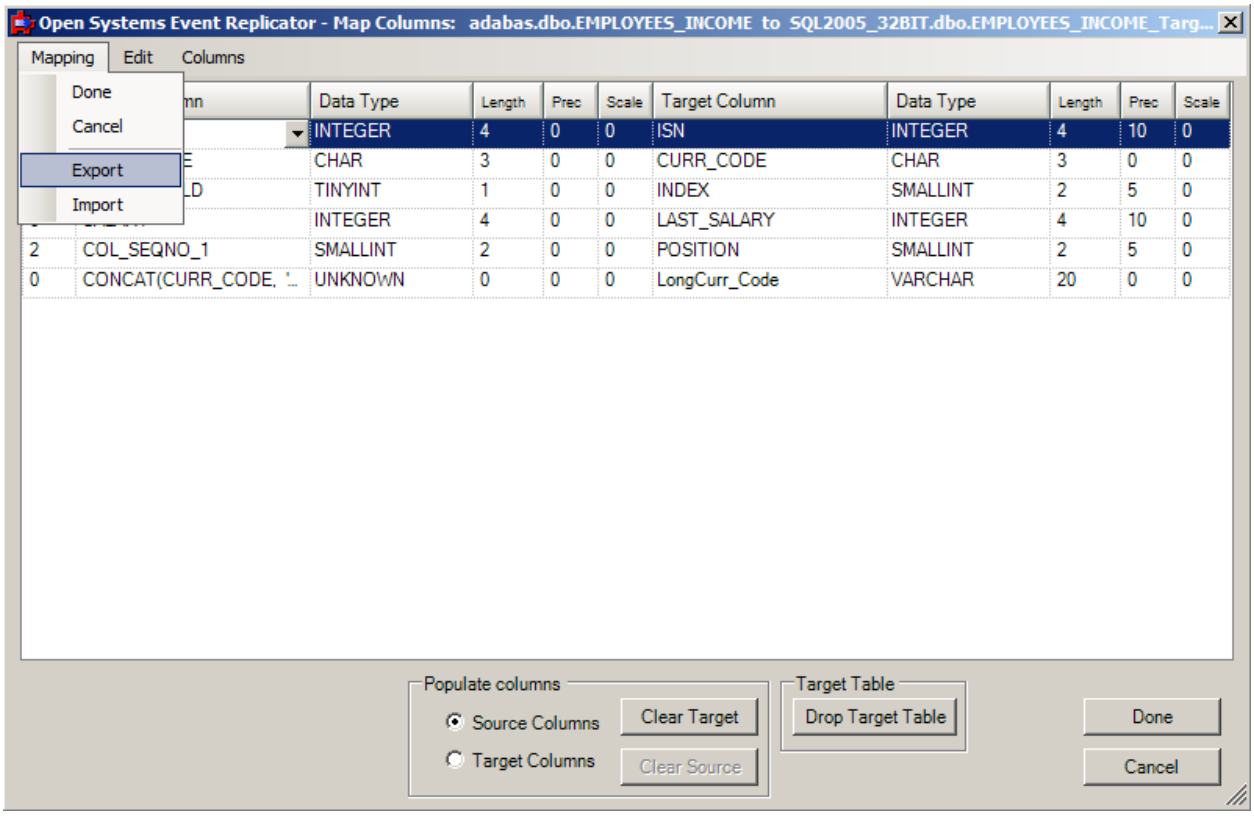
This is done while at the Column Mapping form. Select the "Mapping" menu item, then "Export". Then it will prompt for a name and location of a file to save the mappings to. The mappings will be saved in a comma separated format so they can be viewed and changed if desired with another file editor.

Importing a Column Mapping from a file:

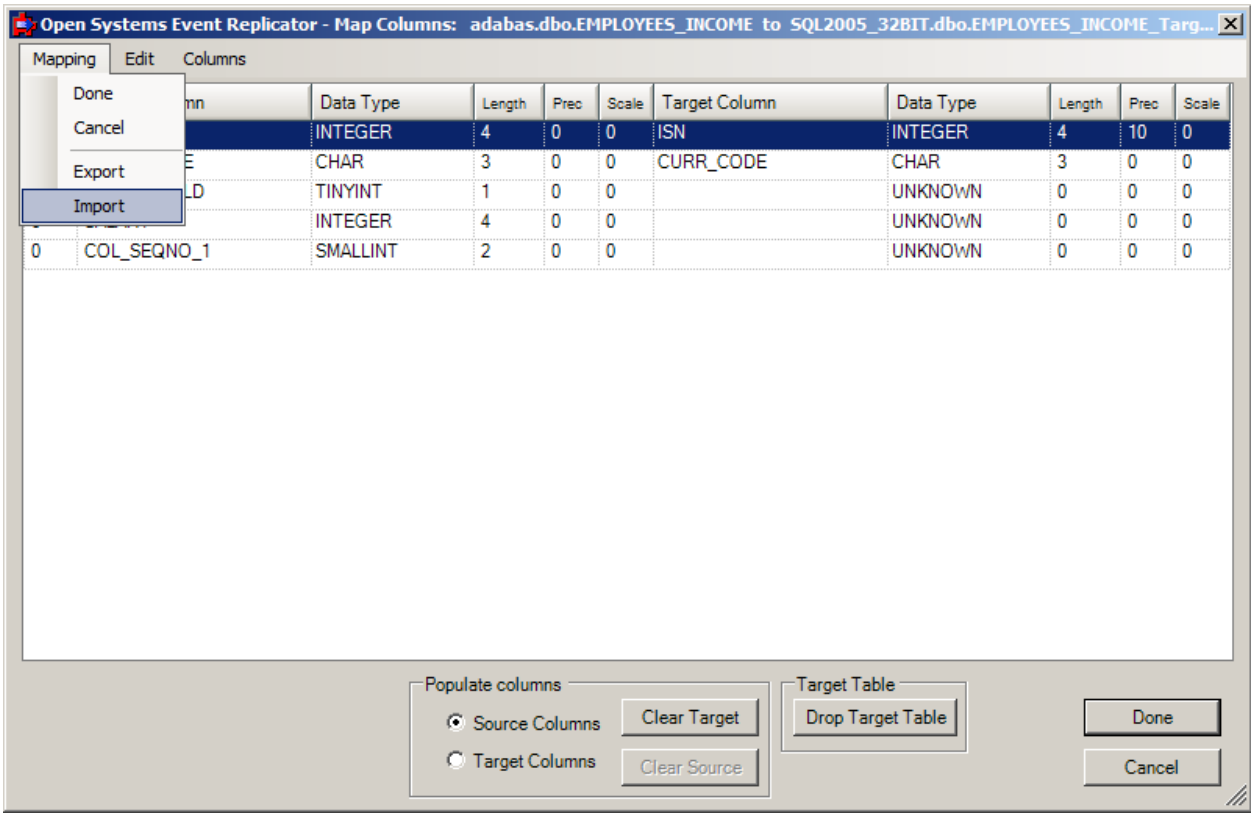
This is done while at the Column Mapping form. Select the "Mapping" menu item, then "Import". Locate the mapping file to import and press ok, then the column mapper will try to apply all the mappings saved in the file to the open column mapping form. When importing, the mappings the on screen will be 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:

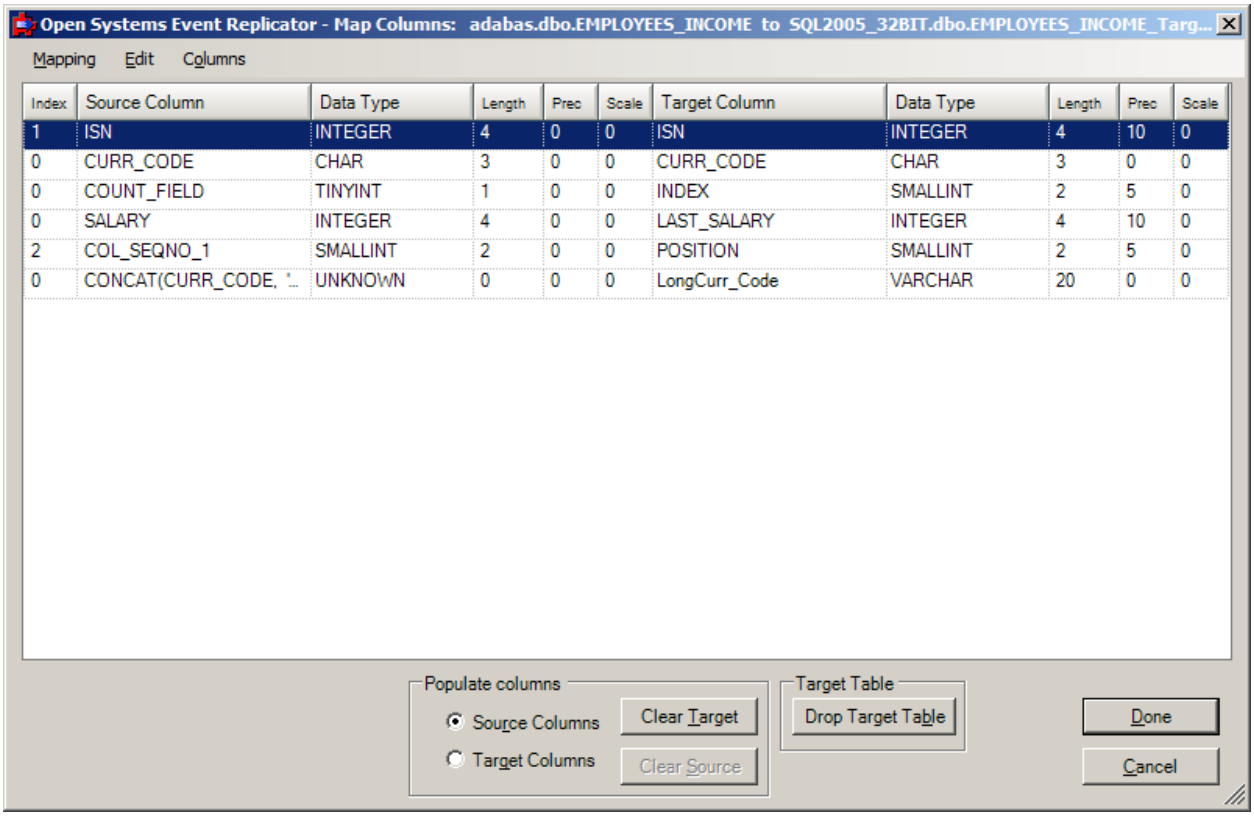
Step 1: A replication that is has a target table with column mappings that include target column renames and an expression on the source table. The user wishes to point the replication source to a different target table. Since that will drop the existing column mappings, before changing the target table 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 colum 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.


Open Systems Event Replicator - Map Columns: adabas.dbo.EMPLOYEES_INCOME to SQL2005_32BIT.dbo.EMPLOYEES_INCOME_Targ...
Mapping
Edit
Columns
Index
Source Column
Data Type
Length
Prec
Scale
Target Column
Data Type
Length
Prec
Scale
1 ISN INTEGER 4 0 0 ISN INTEGER 4 10 0
0 CURR_CODE CHAR 3 0 0 CURR_CODE CHAR 3 0 0
0 COUNT_FIELD TINYINT 1 0 0 INDEX SMALLINT 2 5 0
0 SALARY INTEGER 4 0 0 LAST_SALARY INTEGER 4 10 0
2 COL_SEQNO_1 SMALLINT 2 0 0 POSITION SMALLINT 2 5 0
0 CONCAT(CURR_CODE, '... UNKNOWN 0 0 0 LongCurr_Code VARCHAR 20 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.

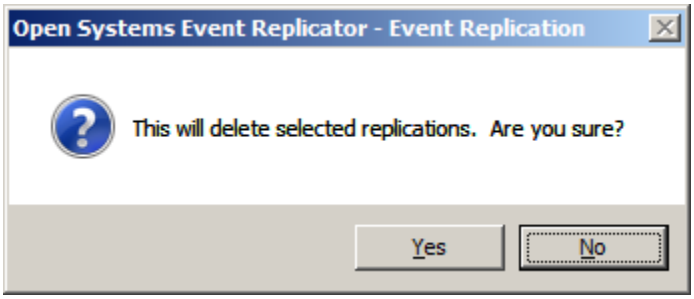
After you delete a replication you will be able to:

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

You can only delete one replication at a time.

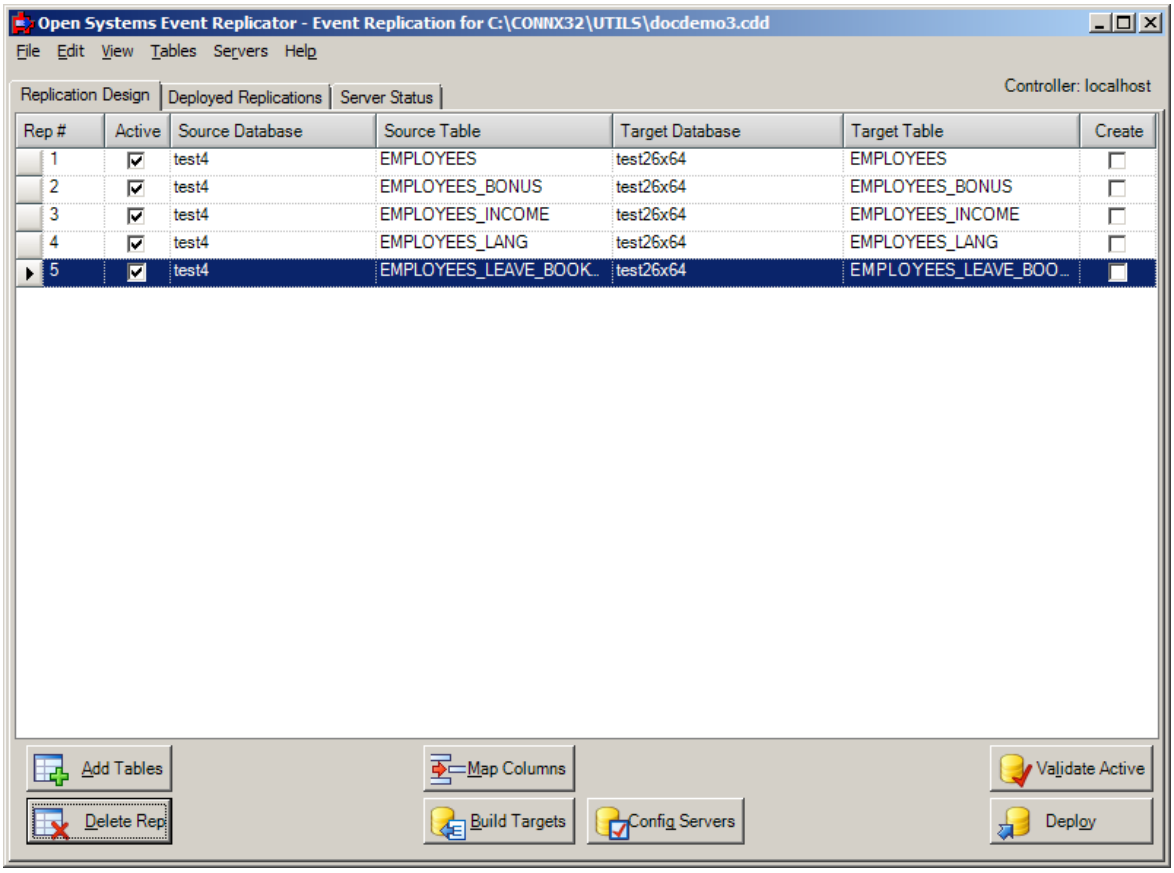
Deleting a replication does not remove the replication by itself. The replication will no longer appear in the Replication Design tab. Data replication will continue until the next time the CDD is deployed.

1. Open the Open Systems Event Replicator CDD.
2. Select the replication you wish to delete and click **Delete Rep**. The following message appears:



3. Click **Yes**. The selected replication will be deleted no matter whether the replication is marked **Active** or **Create**.

   The replication no longer appears in the list of those available to be deployed.



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 data base schemas

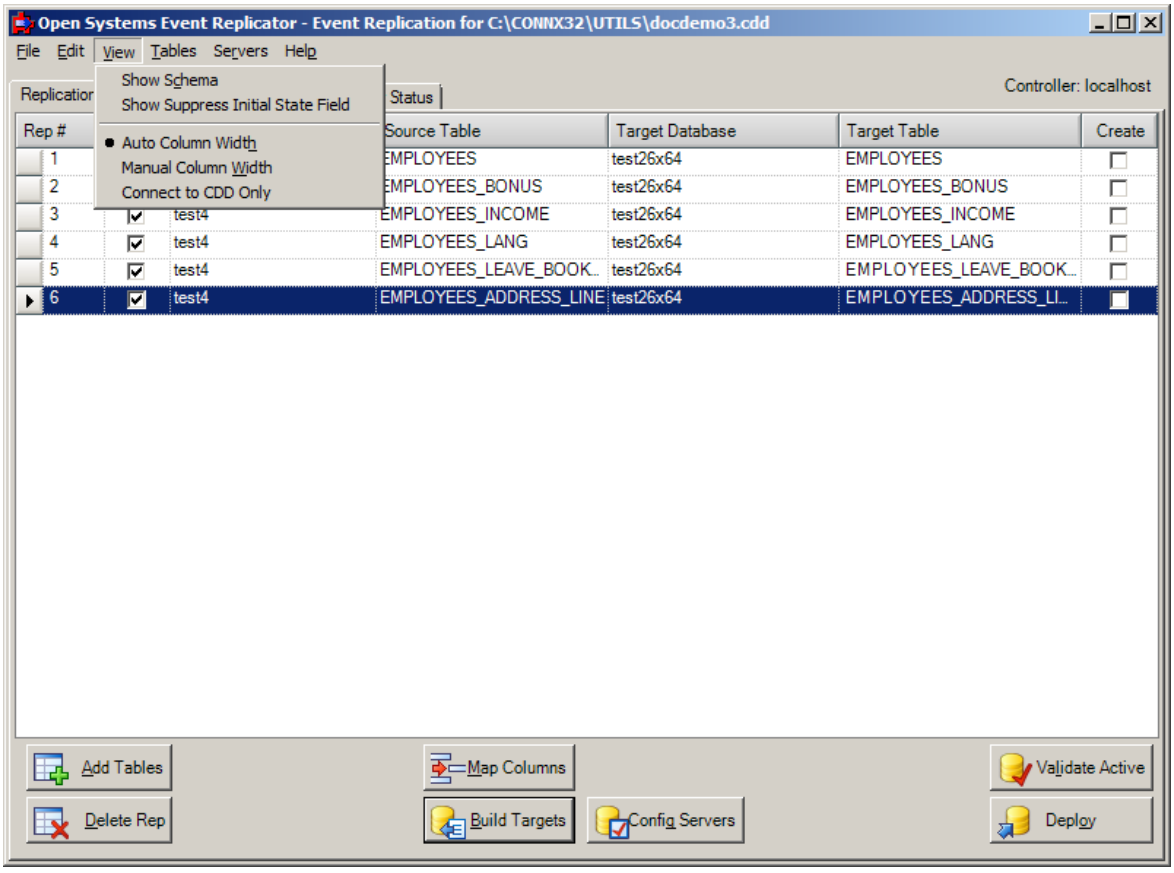
View the source table schema if:

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

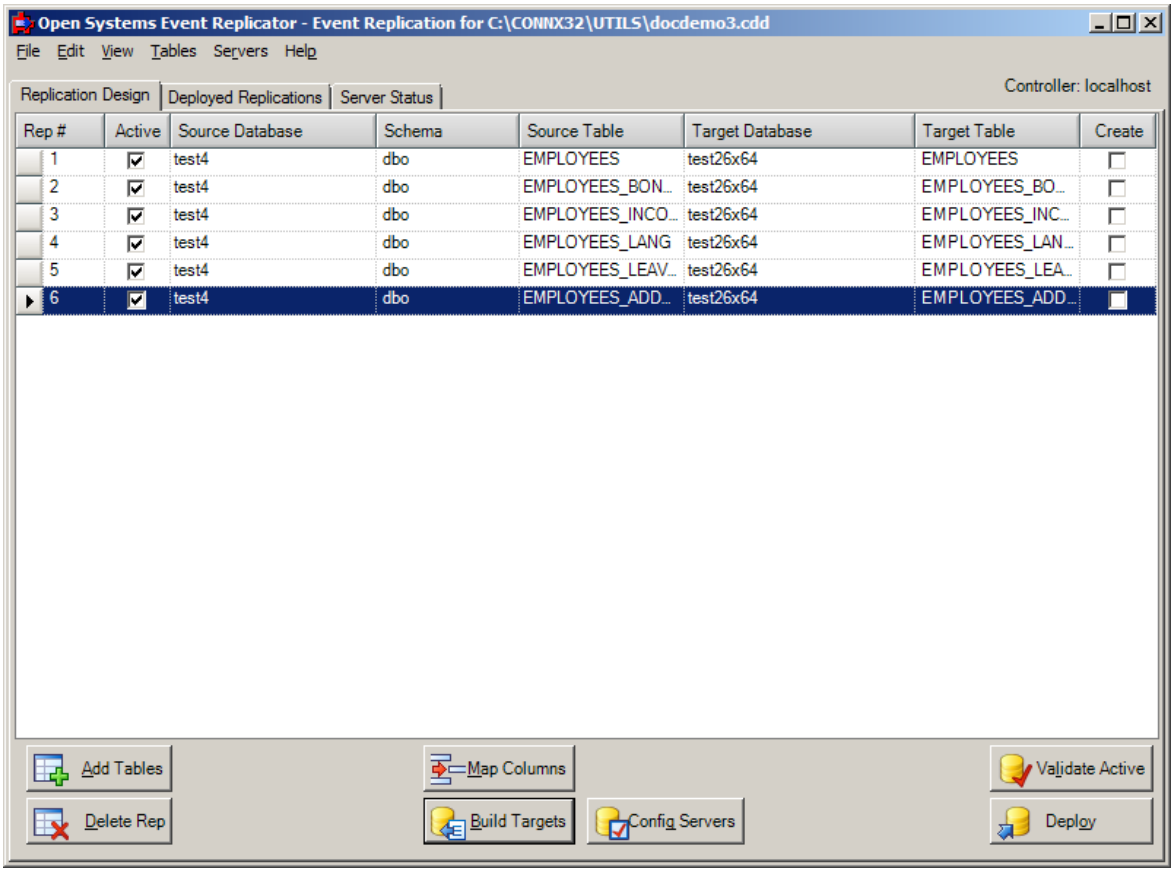
After you view the source table schema you will be able to:

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

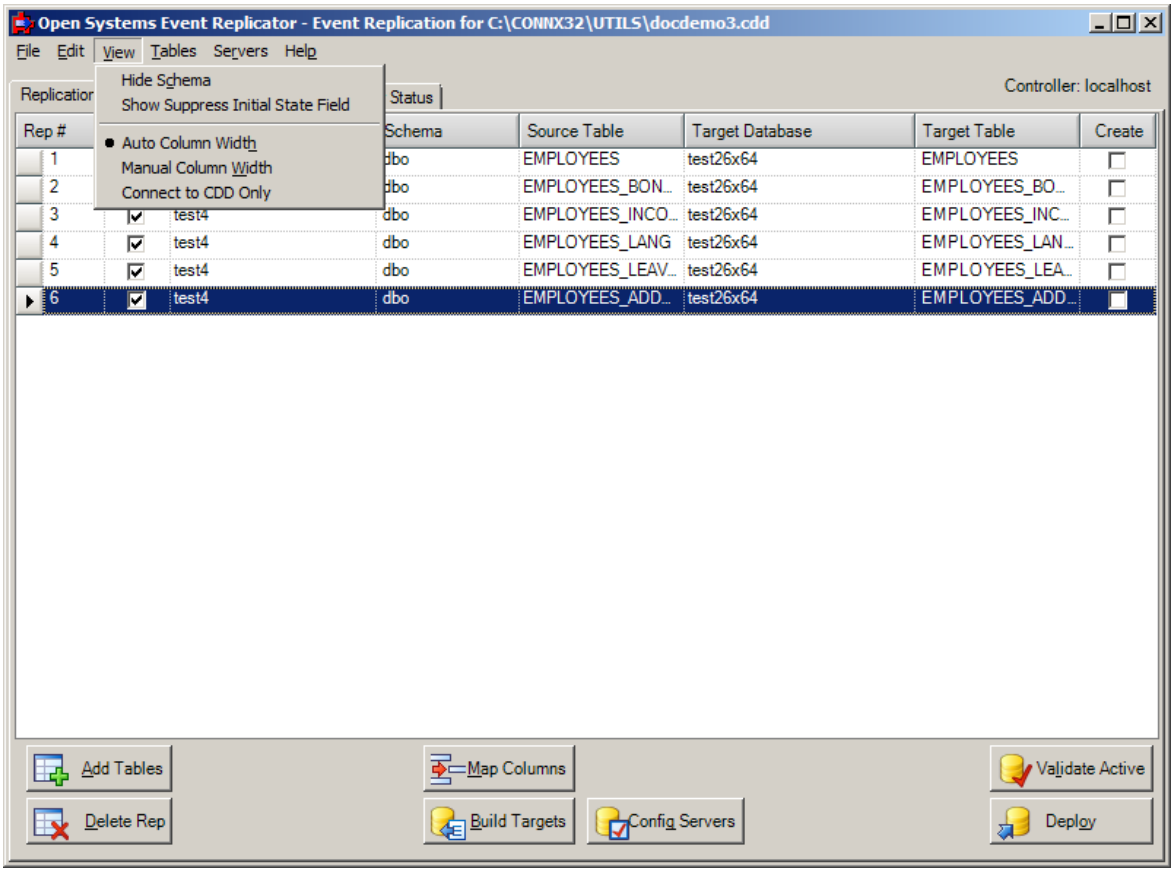
1. Open the Open Systems Event Replicator CDD.
2. From the **View** menu, click **Show Schema**.



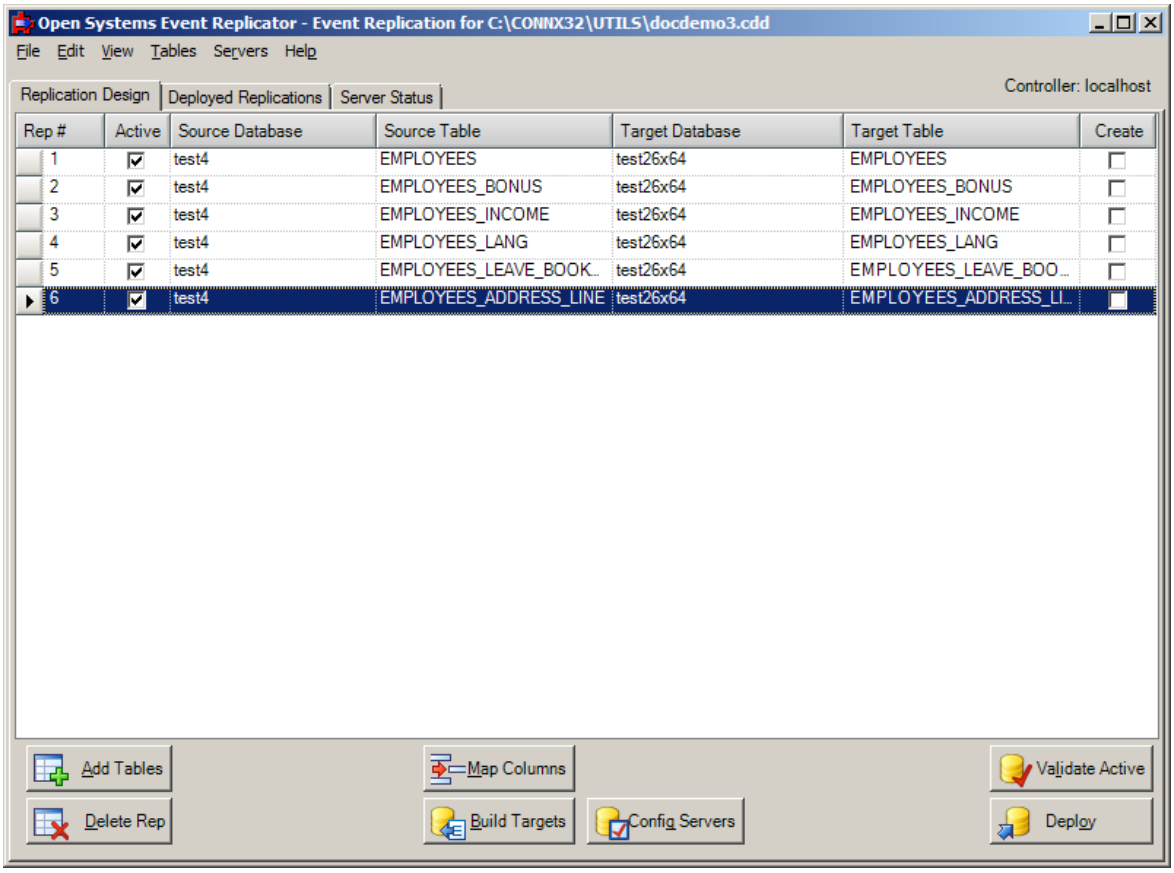
The **Open Systems Event Replication** window now contains an additional column.



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



This removes the schema column.


Open Systems Event Replicator - Event Replication for C:\CONNX32\UTILS\docdemo3.cdd
File Edit View Tables Servers Help
Replication Design
Deployed Replications
Server Status
Controller: localhost
Rep # Active Source Database Source Table Target Database Target Table Create
1 test4 EMPLOYEES test26x64 EMPLOYEES
2 test4 EMPLOYEES_BONUS test26x64 EMPLOYEES_BONUS
3 test4 EMPLOYEES_INCOME test26x64 EMPLOYEES_INCOME
4 test4 EMPLOYEES_LANG test26x64 EMPLOYEES_LANG
5 test4 EMPLOYEES_LEAVE_BOOK... test26x64 EMPLOYEES_LEAVE_BOO...
6 test4 EMPLOYEES_ADDRESS_LINE test26x64 EMPLOYEES_ADDRESS_LI...
Add Tables
Delete Rep
Map Columns
Build Targets
Config Servers
Validate Active
Deploy

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

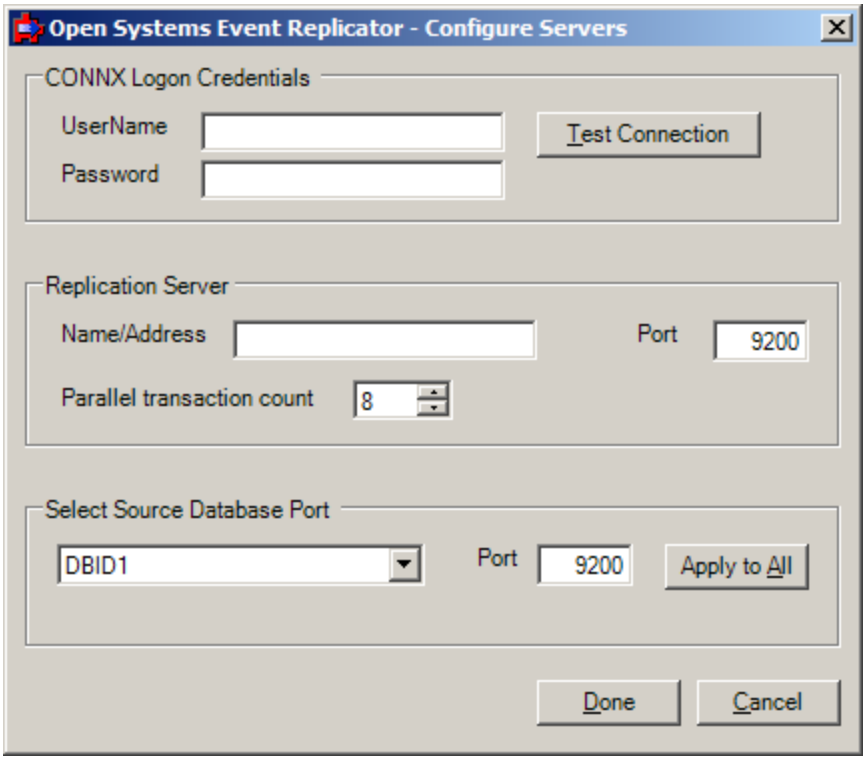
Configure the replication and source database servers if:

- The configuration information is missing, changed, or incorrect

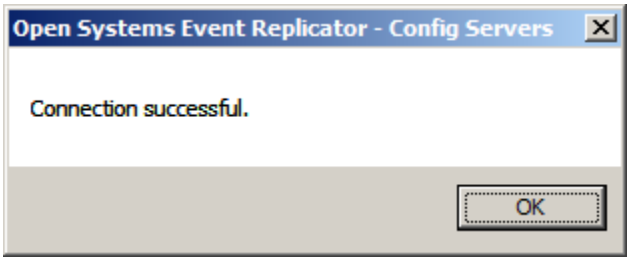
After you configure the replication and source database servers you will be able to:

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

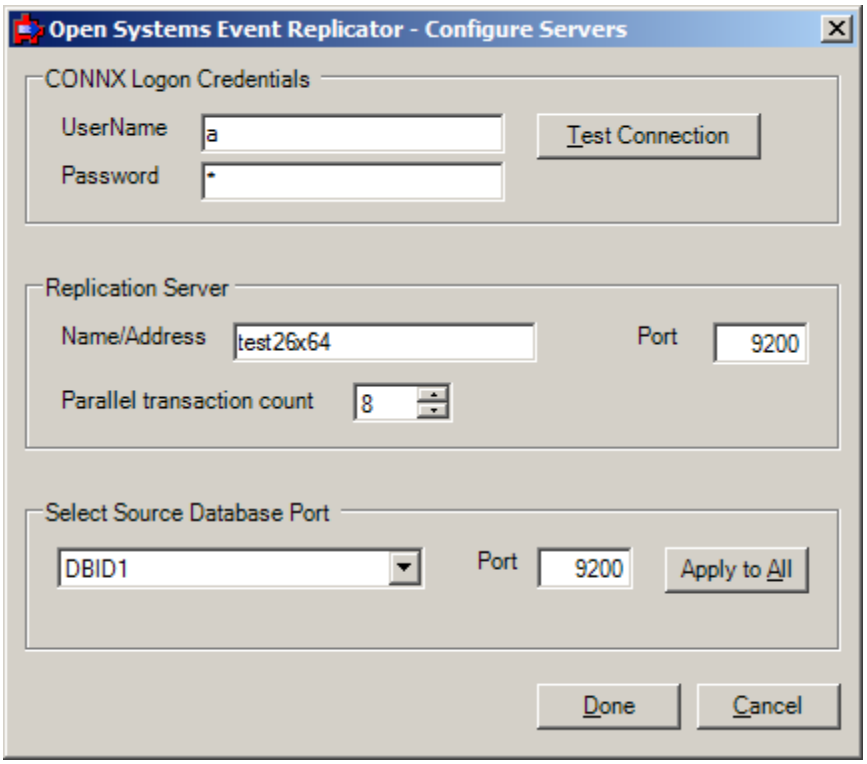
1. Open the Open Systems Event Replicator CDD.
2. Click **Config Servers**. The **Configure Servers** window appears.



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.



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:

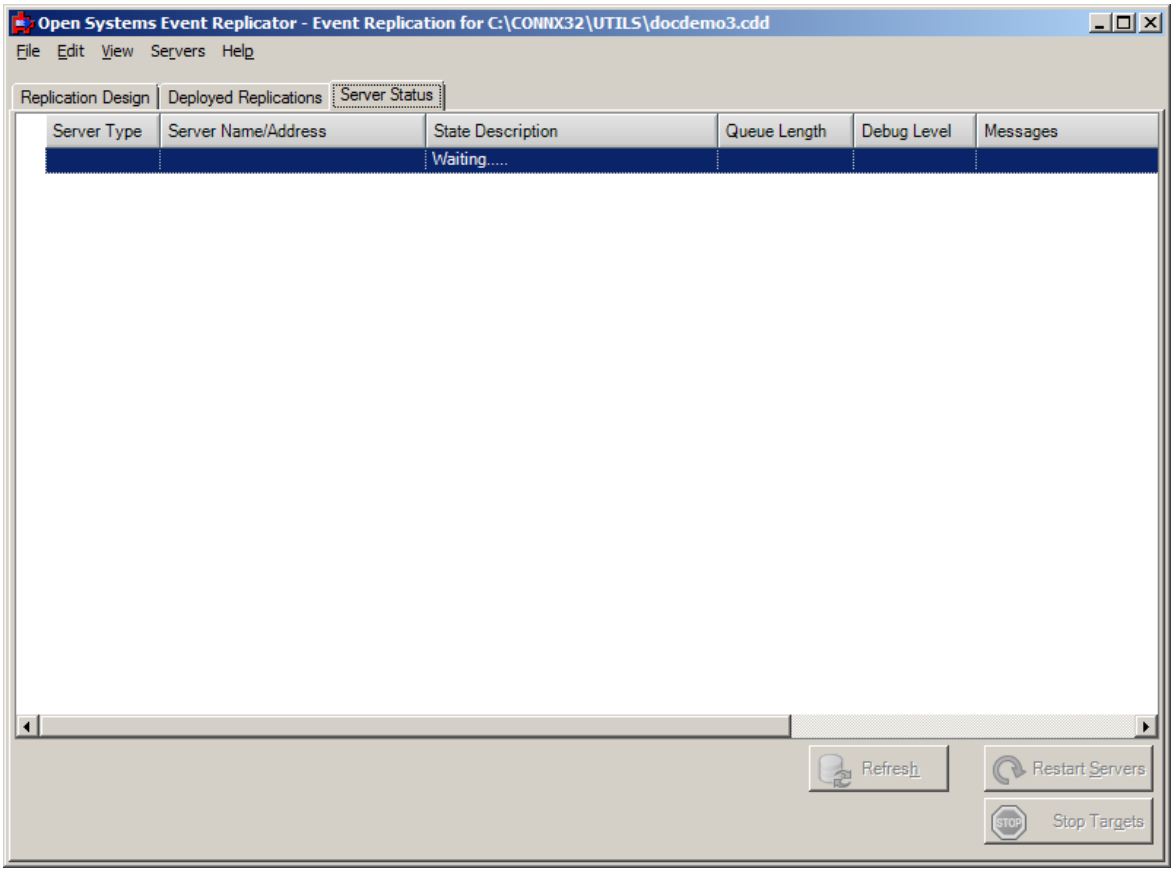
- The log file contains error messages about a server.

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

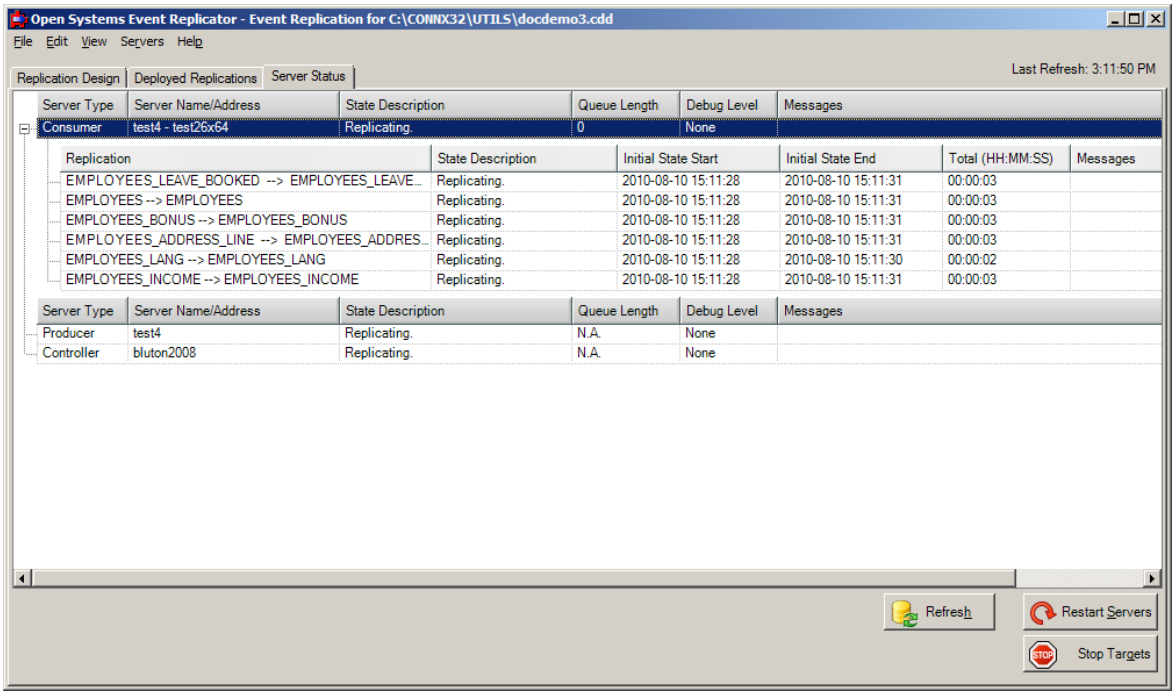
- Stop the replication servers
- Restart the replication servers

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

1. Open the Open Systems Event Replicator CDD.
2. Click the **Server Status** tab. The following appears:



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



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. Potential values and their meaning are as follows:

  - **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)

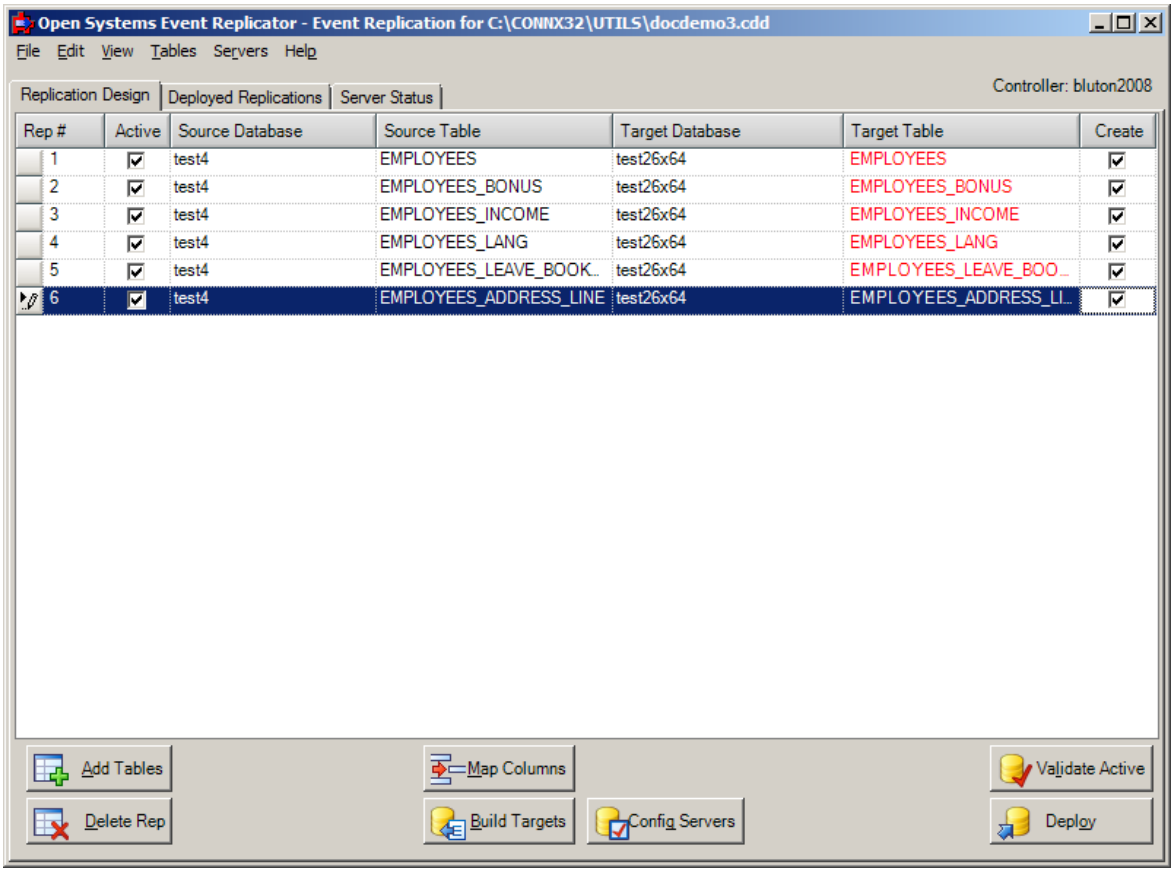
You can build new target tables without deploying the Event Replicator.

Note: The Event Replicator can not build tables for Adabas target databases. The table must exist on the target Adabas database and be imported into the CDD before it is enabled for replication.

To add an existing target database tablet 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 Open Systems Event Replicator CDD.
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

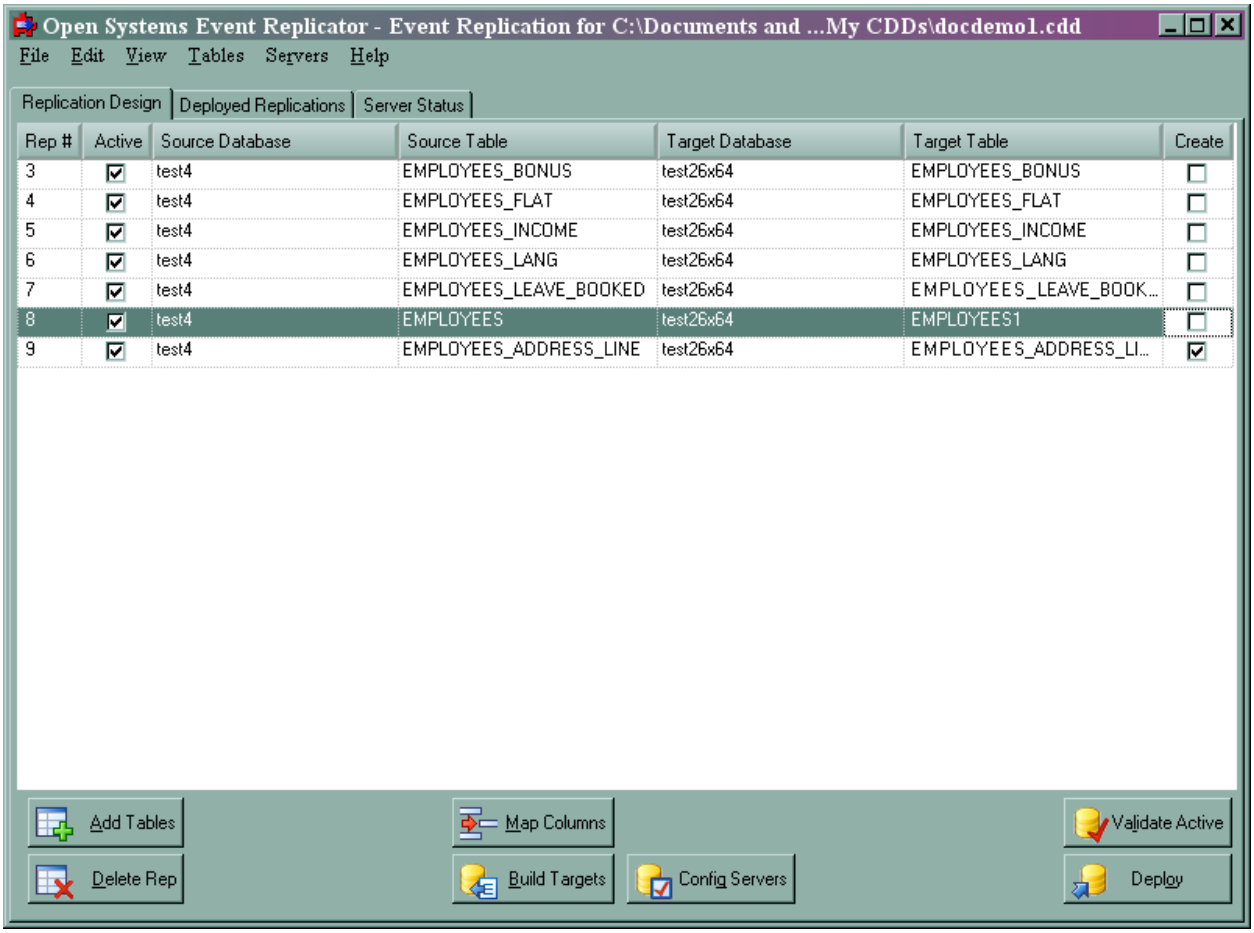
After dropping and recreating new target database tables you will be able to:

- 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 the Event Replicator.

If a target table exists on the target database, there is no check in the **Create Table** column.

1. Open the Open Systems Event Replicator CDD.
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 the Event Replication

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

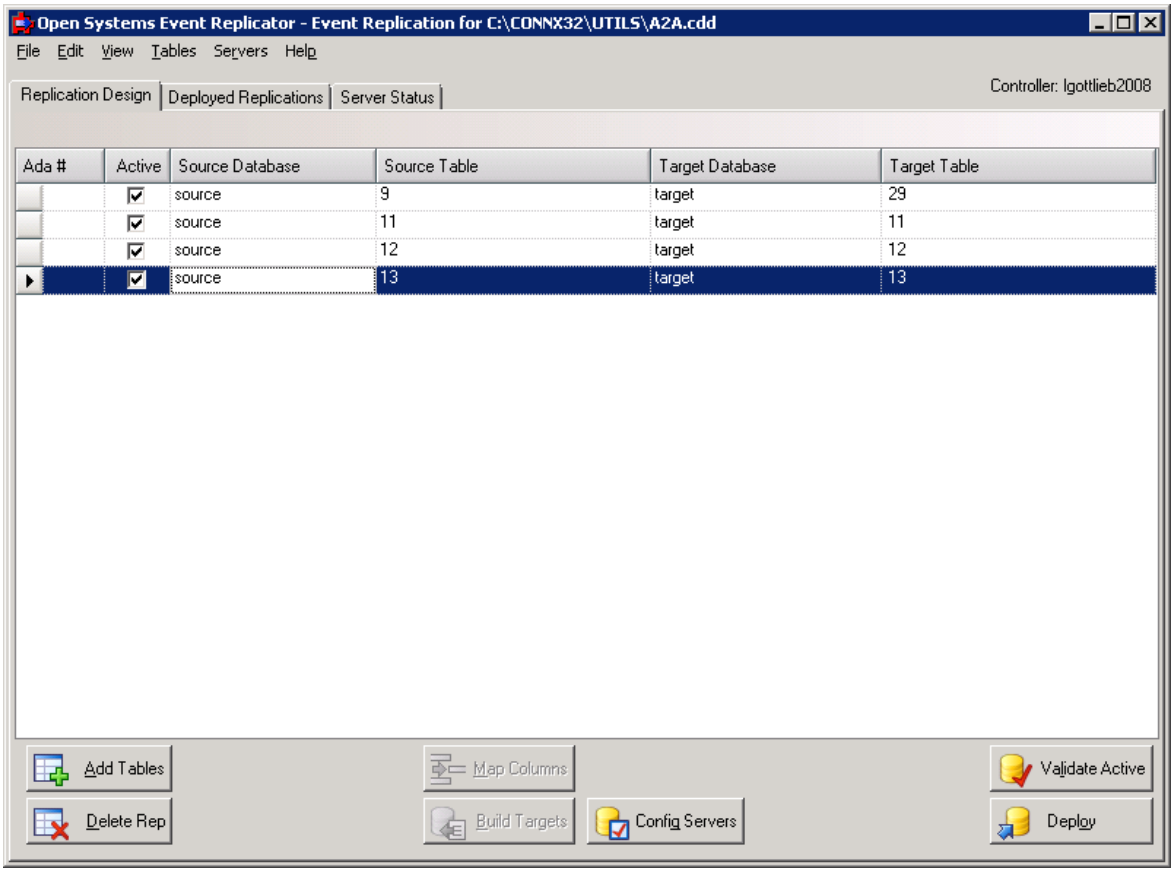
After deploying the event replication you will be able to:

- Check server status

All changes made using the Open System Event Replicator 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.



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.

The **Ada #** column will be blank until the deploy has completed. In an Adabas to Adabas replication environment, the Ada # relates to the ID column from the Adabas adaopr command when using the display=replication option. For more information about the adaopr command and the replication related options, please see the Adabas Utilities documentation on the Software AG Empower website.

Note: Starting with version 12 SP3, Adabas to Adabas replications can no longer be undeployed by unchecking the **Active** checkbox and redploying. To undeploy an individual Adabas to Adabas replication, use the UnDeploy Selected button on the **Deployed Replications** 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

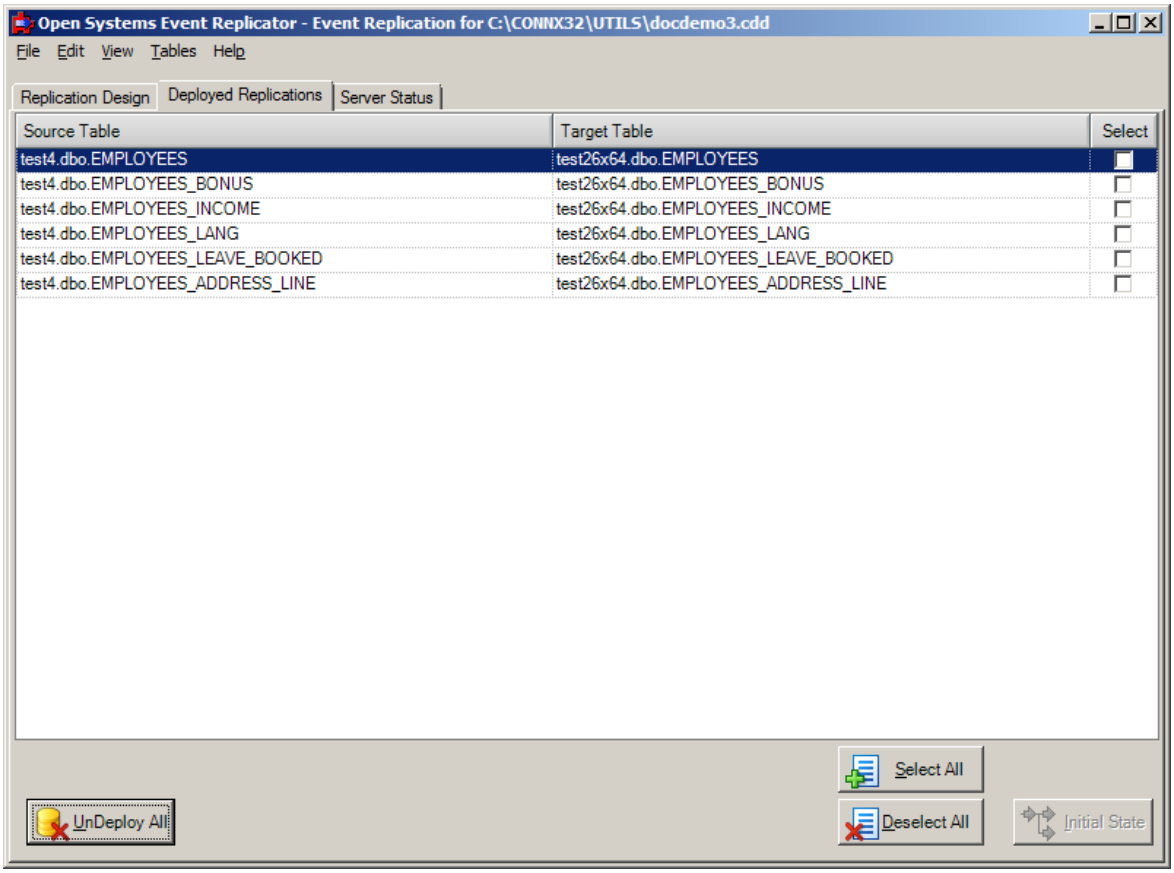
After performing an Initial State you will be able to:

- 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 Open Systems Event Replicator CDD.
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**. The following message appears:

If the initial state request failed, the following message appears:

Suppressing an Initial State

Suppressing an Initial State includes:

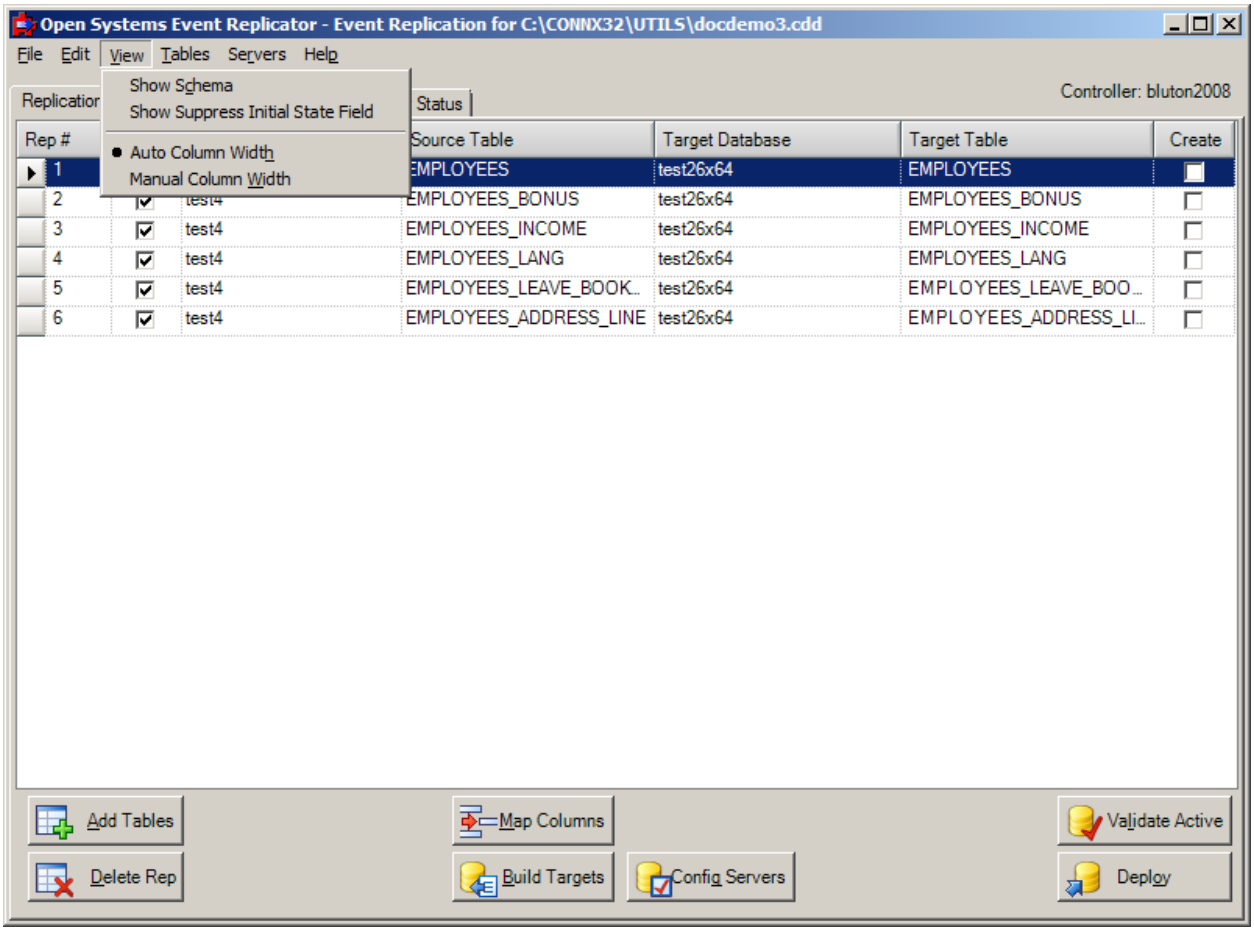
- Opening a CONNX Data Dictionary (CDD)
- Moving a copy of the source database into the target database

Suppress an Initial State if:

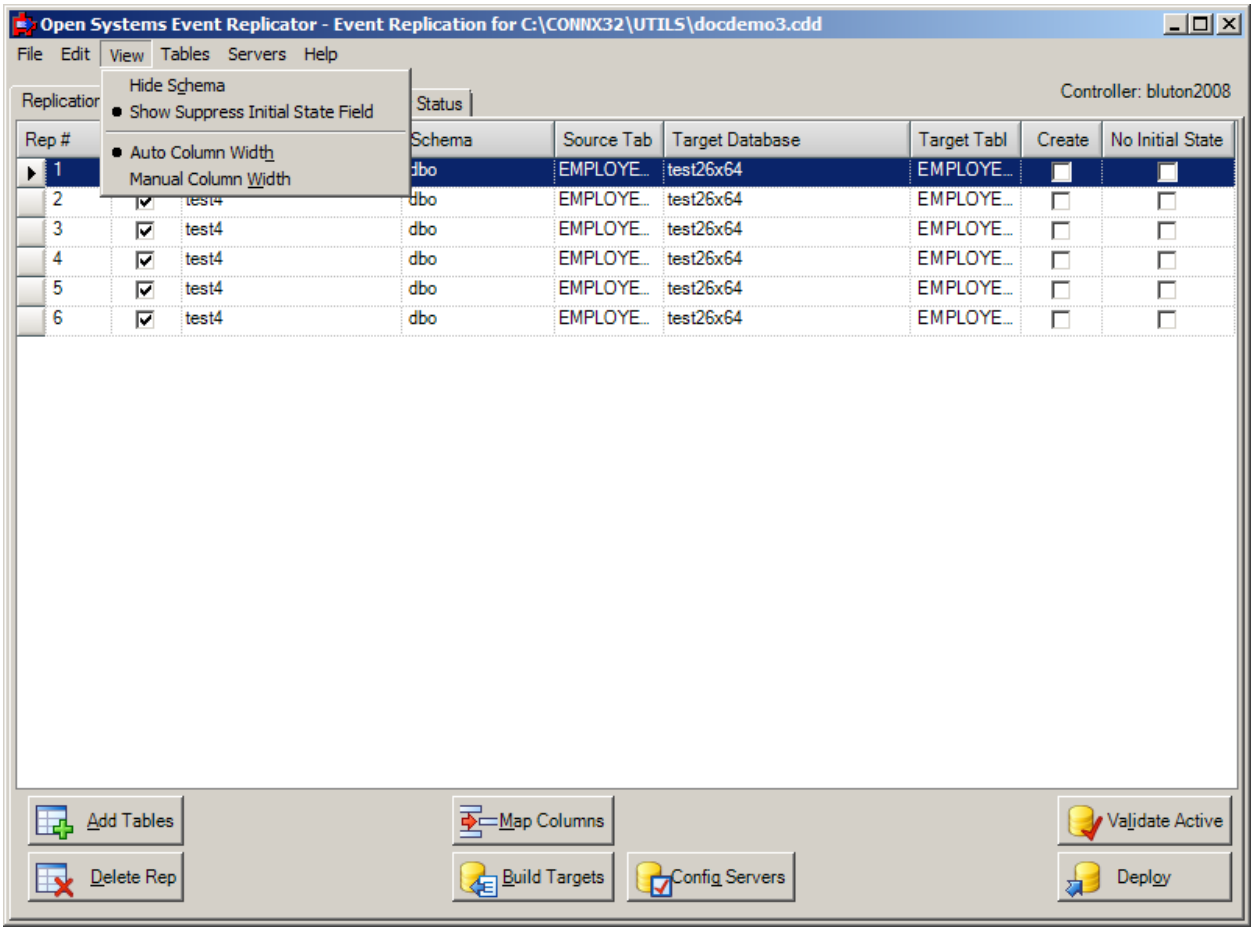
- An initial state has already been performed and the replication has been undeployed
- The target table has been synchronized with the source with an outside tool
- The system experienced a crash and you need to bring replication back on-line as soon as possible and you can tolerate some inconsistencies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

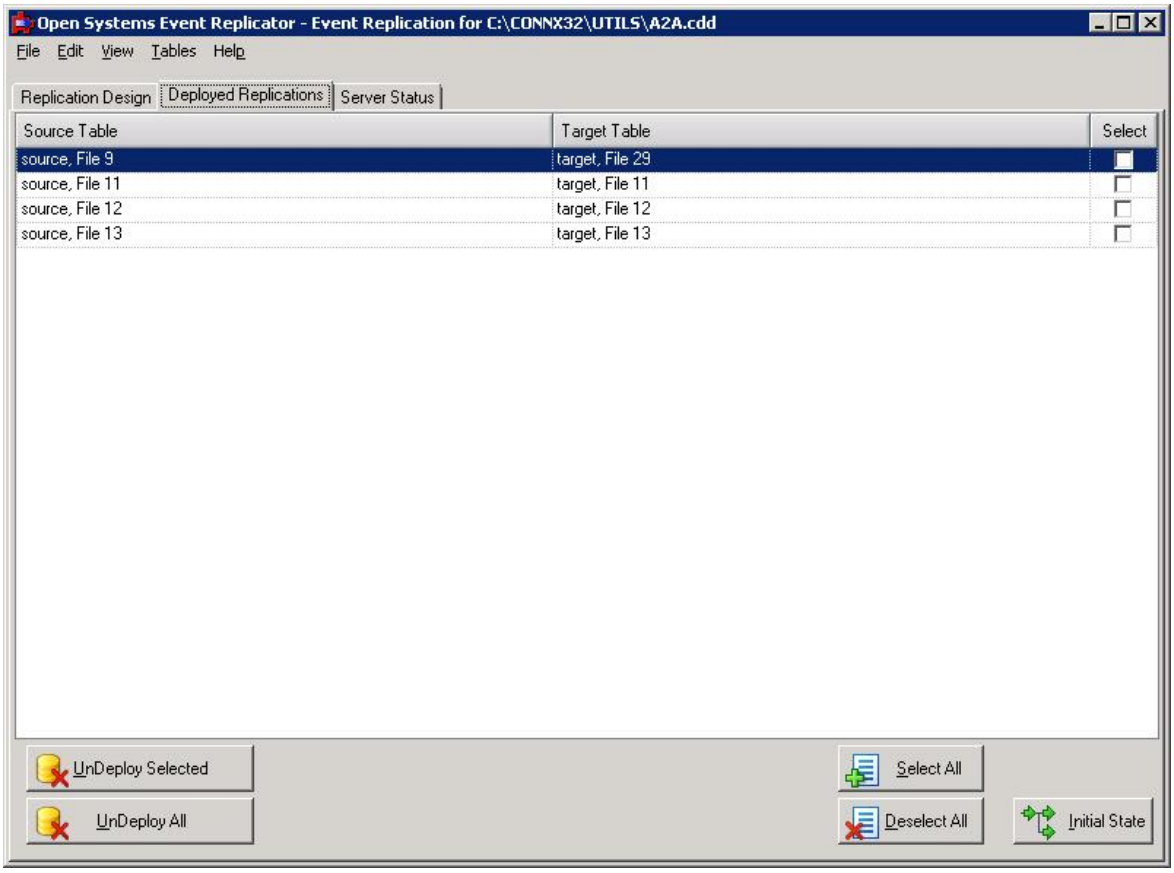
- Do maintenance on the target database
- Replicate to a new target database
- Use a different Controller
- Create a new CDD and use the new CDD for replication

Un-deploy All will un-deploy all active replications at once and put the controller and EP in an idle state.

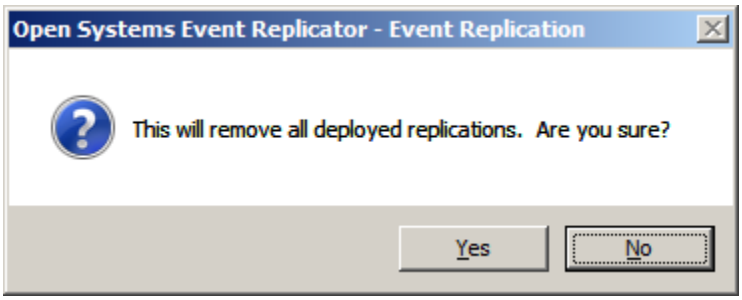
Un-deploy All un-deploys the entire CDD, not just the individual replications.

If you don't want to un-deploy all active replications, use the **UnDeploy Selected** button instead of the **UnDeploy All** button.

1. Open the Open Systems Event Replicator CDD.
2. Click the Deployed Replications tab.

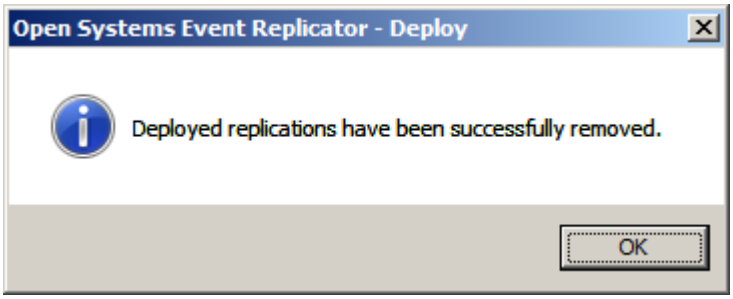


3. Click **Un-Deploy All**. The following message appears:



Warning: Clicking **Yes** will un-deploy **ALL** replications whether you do or don't click **Select.**

The following message appears:


Open Systems Event Replicator - Deploy
Deployed replications have been successfully removed.
OK

Un-Deploying selected Replications

Un-Deploying selected replications includes:

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

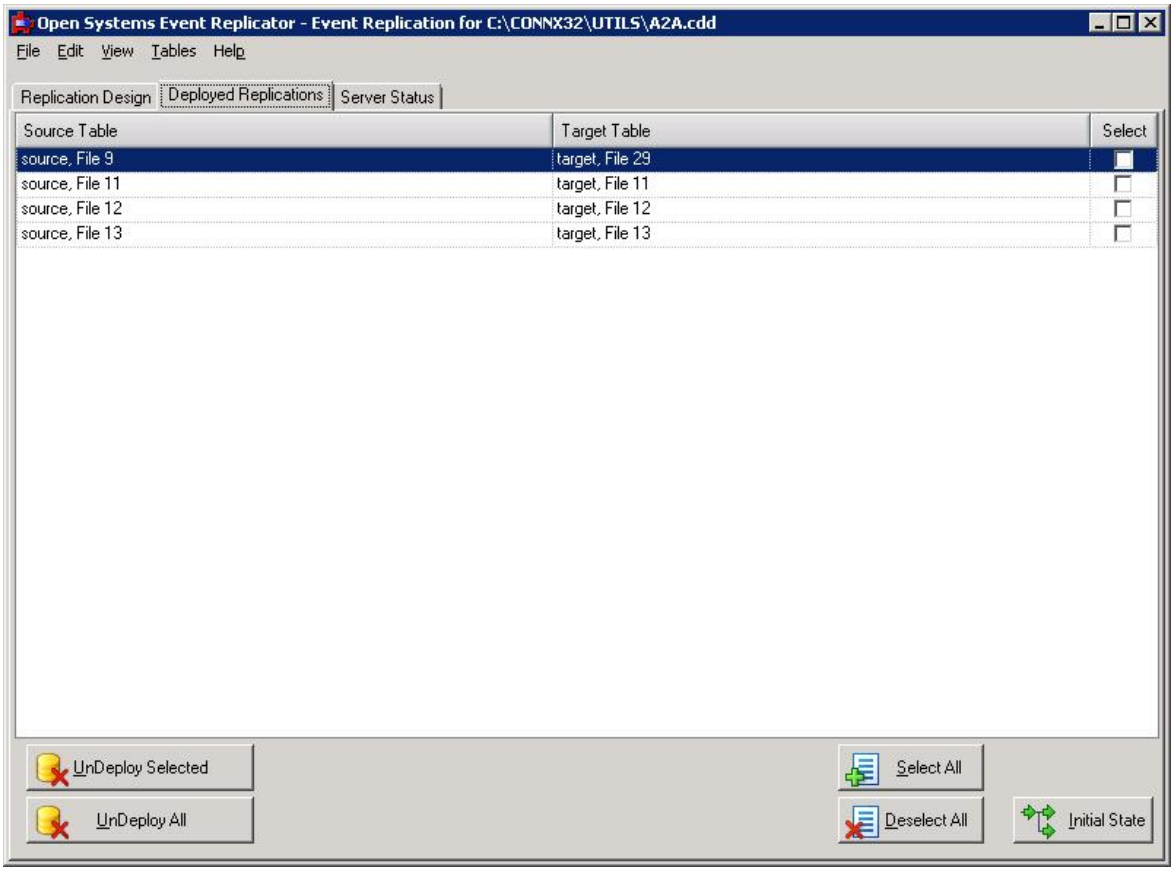
Un-Deploy selected event replications if:

- The target database will be out of service for an extended period
- You no longer want to use some but not all of the current replications

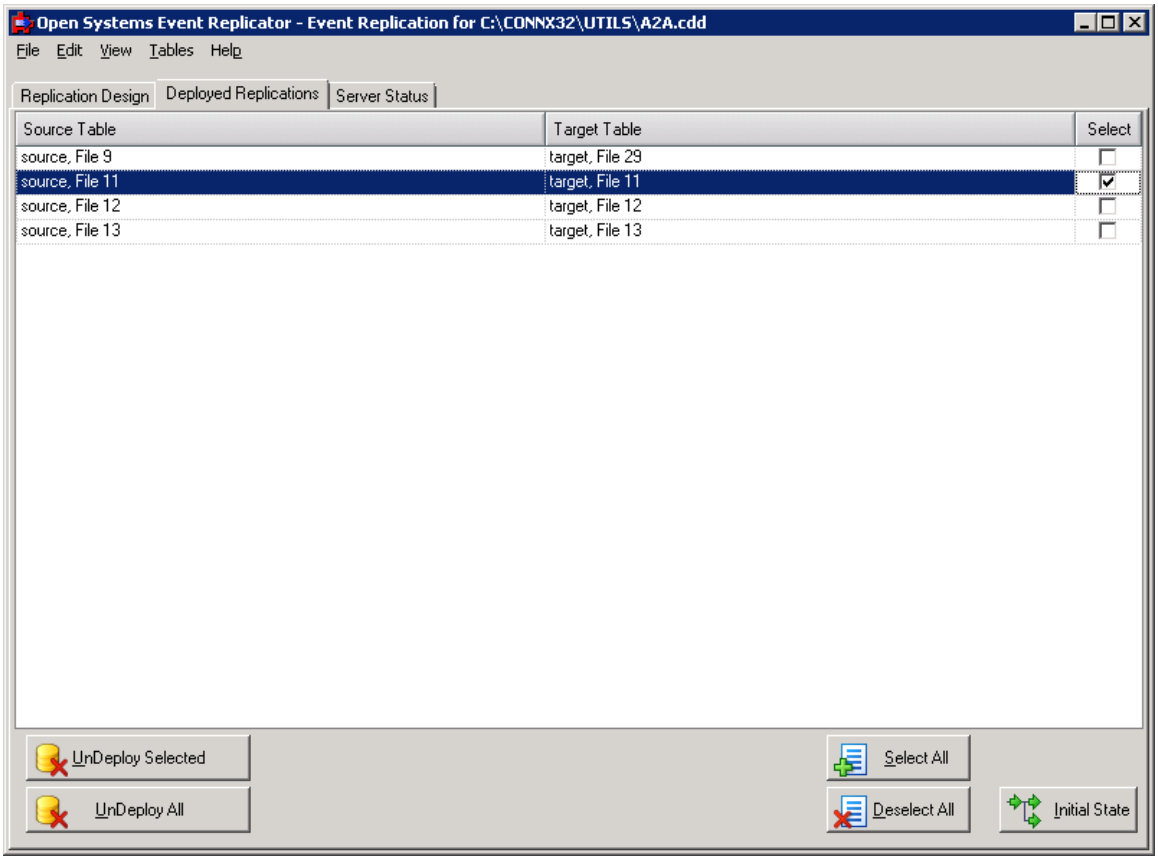
Un-deploy Selected will un-deploy only those replications with the **Select** checkbox checked.

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**.

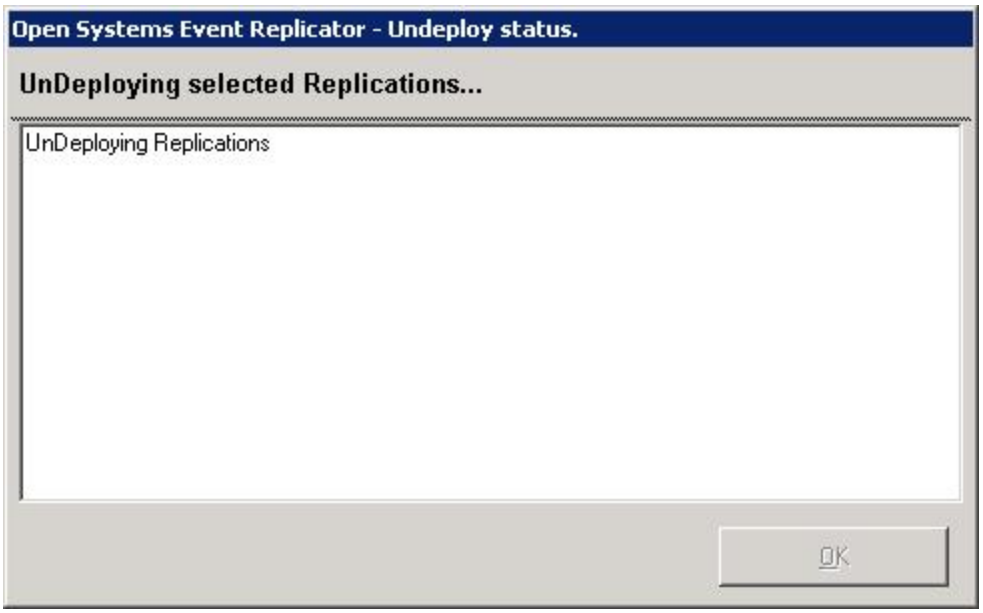
1. Open the Open Systems Event Replicator CDD.
2. Click the Deployed Replications tab.



3. Select a replication to Undeploy - in this case we will select file 11.



4. Click **Un-Deploy Selected**. The following message appears:



A message will be sent to the EP instructing it to remove the replication(s) specified from the list of deployed replications.

Note: If you use **UnDeploy Selected** to remove all the replications, the controller and EP will remain active with nothing to do. If yo wish them to become inactive, use the UnDeploy 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.

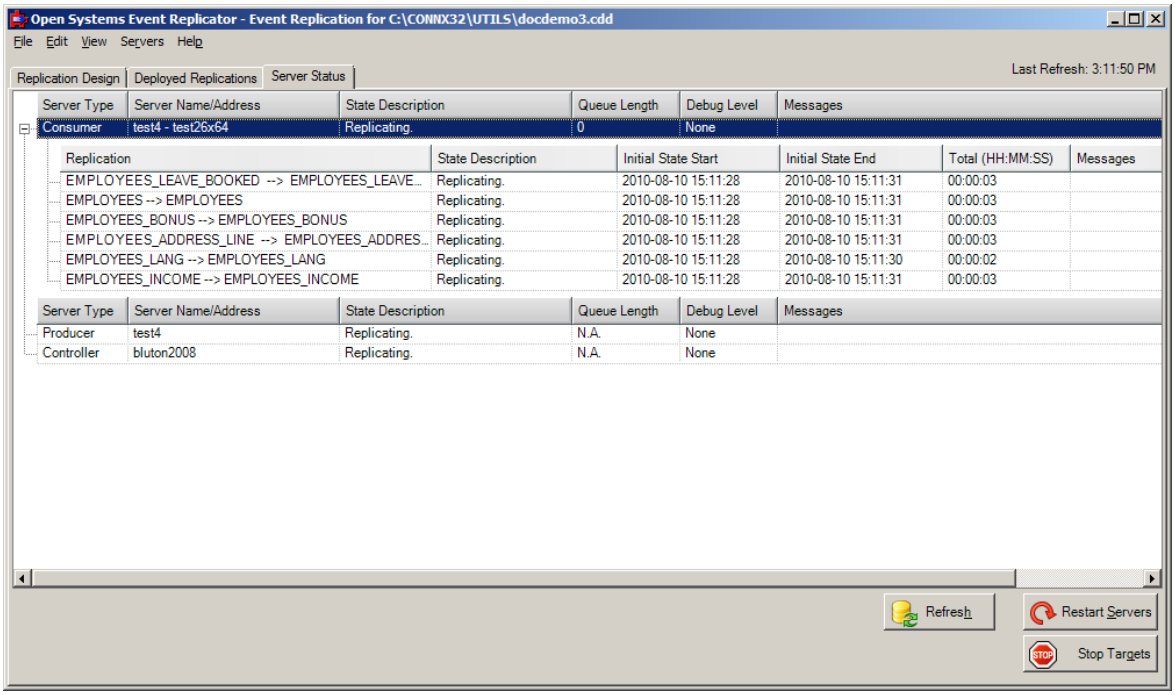
Stop the target servers before you do any target database or server maintenance.

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

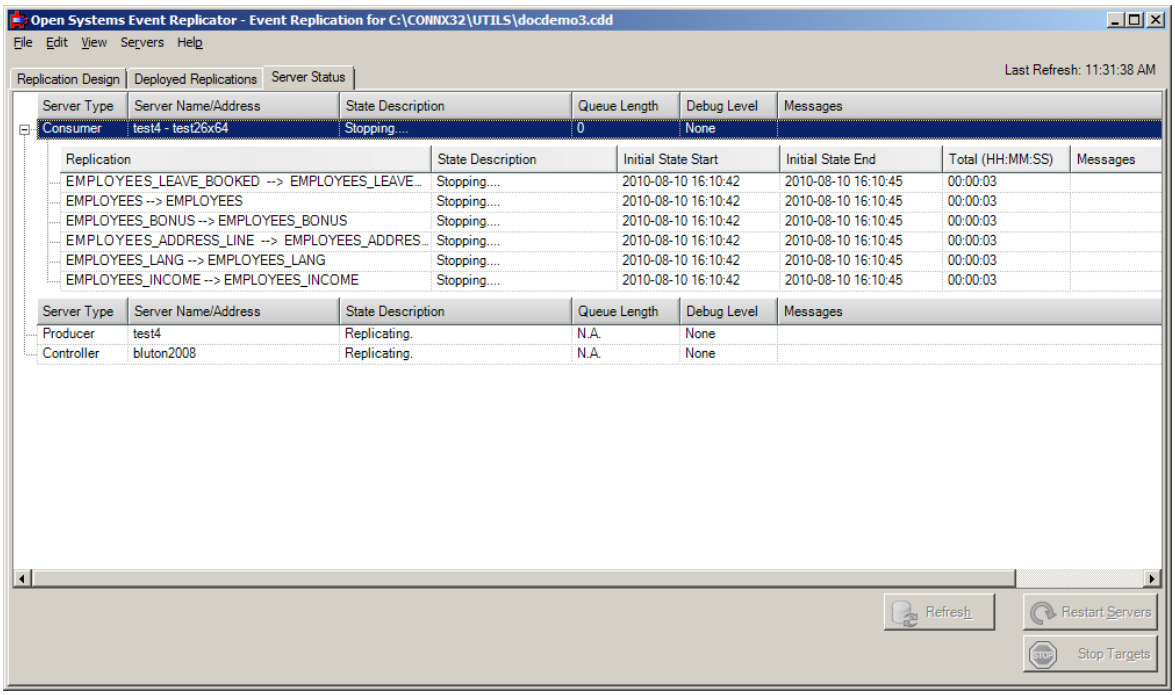
Note: **Stop Targets** does not stop source or replication servers.

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

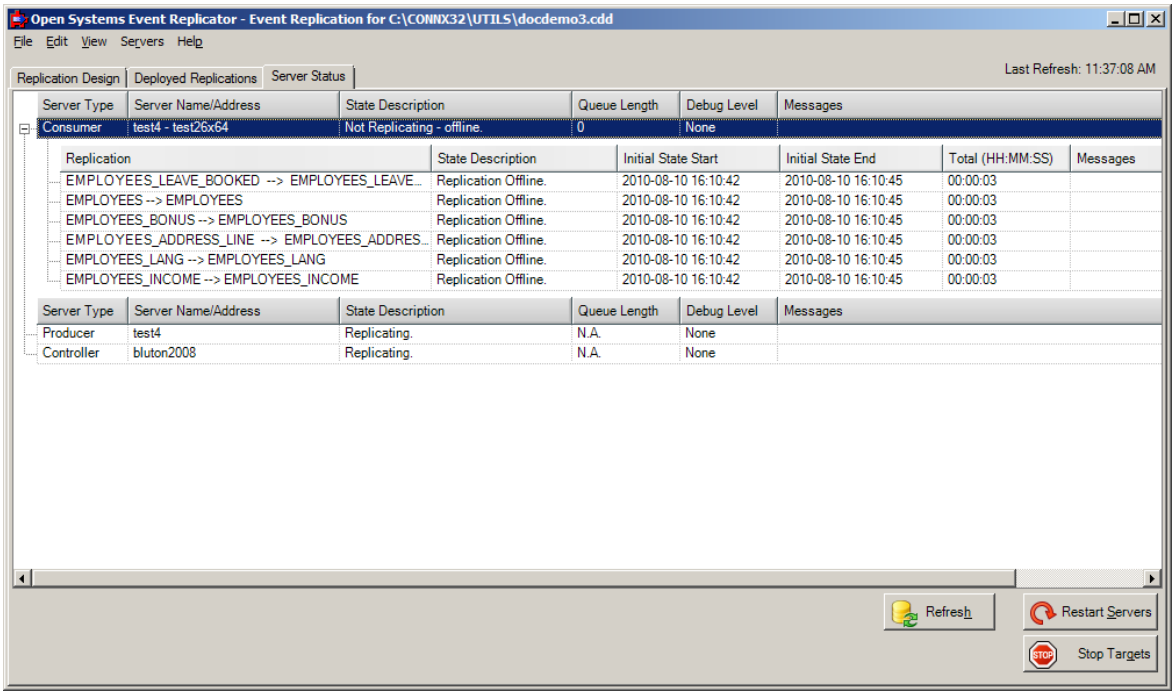
1. Open the Open Systems Event Replicator CDD.
2. Click the **Server Status** tab.



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:


Open Systems Event Replicator - Event Replication for C:\CONNX32\UTILS\docdemo3.cdd
File Edit View Servers Help
Replication Design
Deployed Replications
Server Status
Last Refresh: 11:37:08 AM
Server Type
Server Name/Address
State Description
Queue Length
Debug Level
Messages
Consumer
test4 - test26x64
Not Replicating - offline.
0
None
Replication
State Description
Initial State Start
Initial State End
Total (HH:MM:SS)
Messages
EMPLOYEES_LEAVE_BOOKED --> EMPLOYEES_LEAVE...
Replication Offline.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES --> EMPLOYEES
EMPLOYEES_BONUS --> EMPLOYEES_BONUS
EMPLOYEES_ADDRESS_LINE --> EMPLOYEES_ADDRES...
EMPLOYEES_LANG --> EMPLOYEES_LANG
EMPLOYEES_INCOME --> EMPLOYEES_INCOME
Producer
test4
Replicating.
N.A.
None
Controller
bluton2008
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

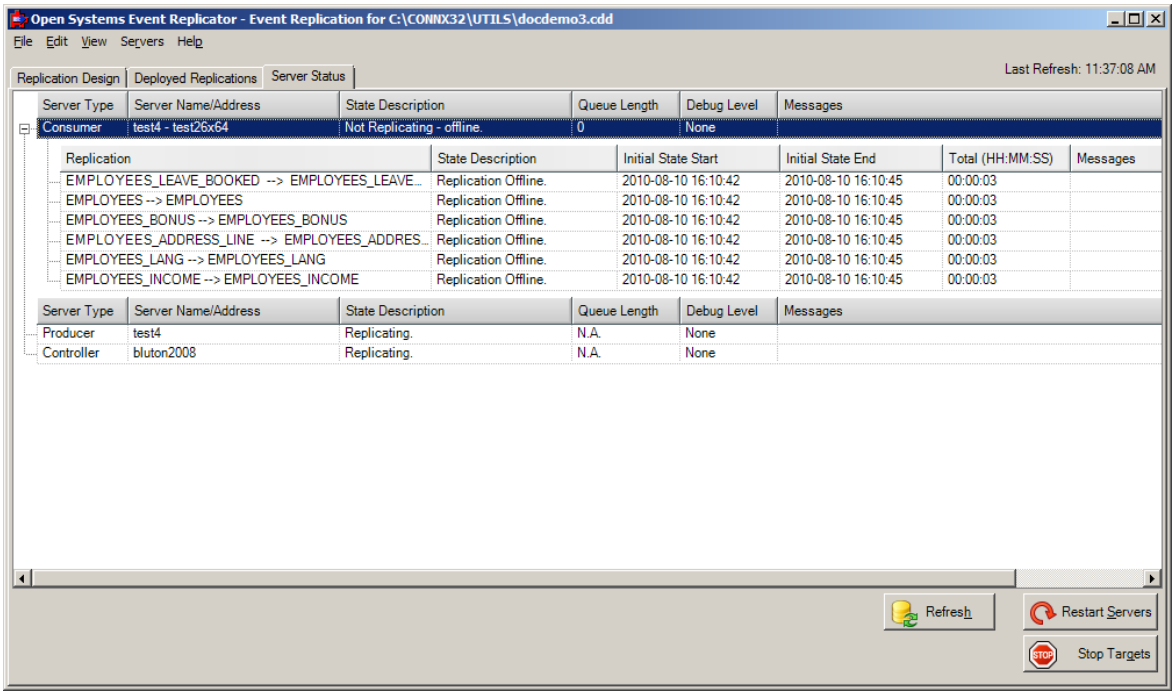
If the Event Replicator has been stopped, restart the target servers to enable replication.

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.

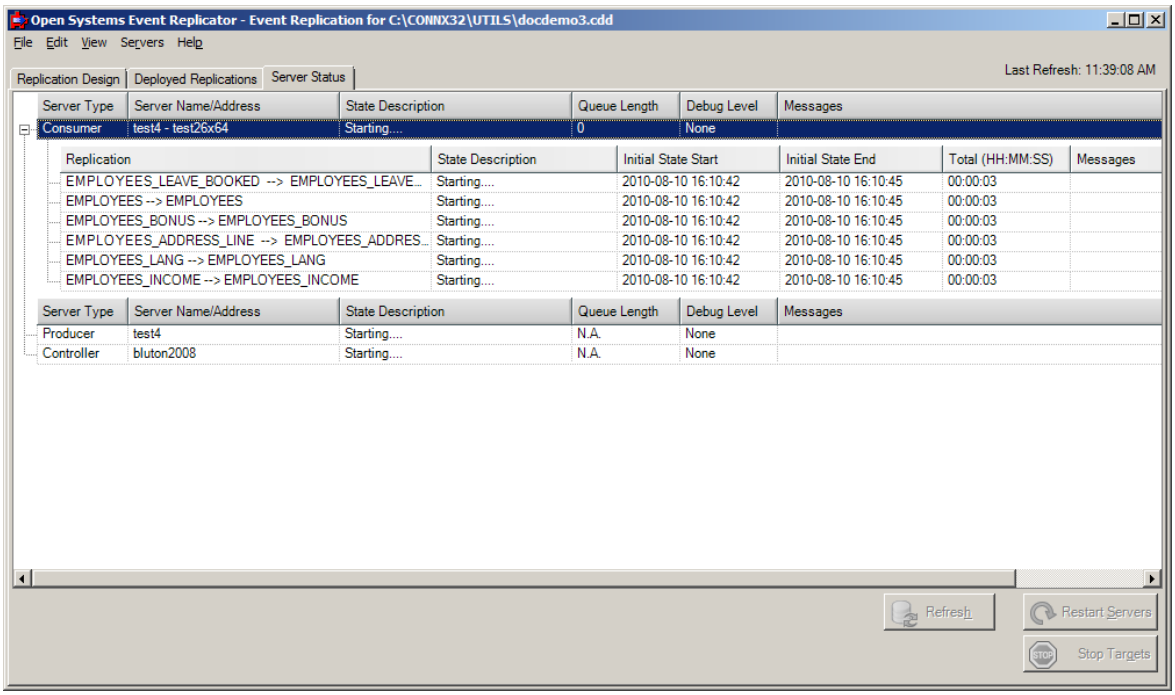
All replications that have been processed while the servers have been stopped remain in the queue in the order received. After the servers have been started, the Event Replicator processes the replications in the order they appear in the queue.

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

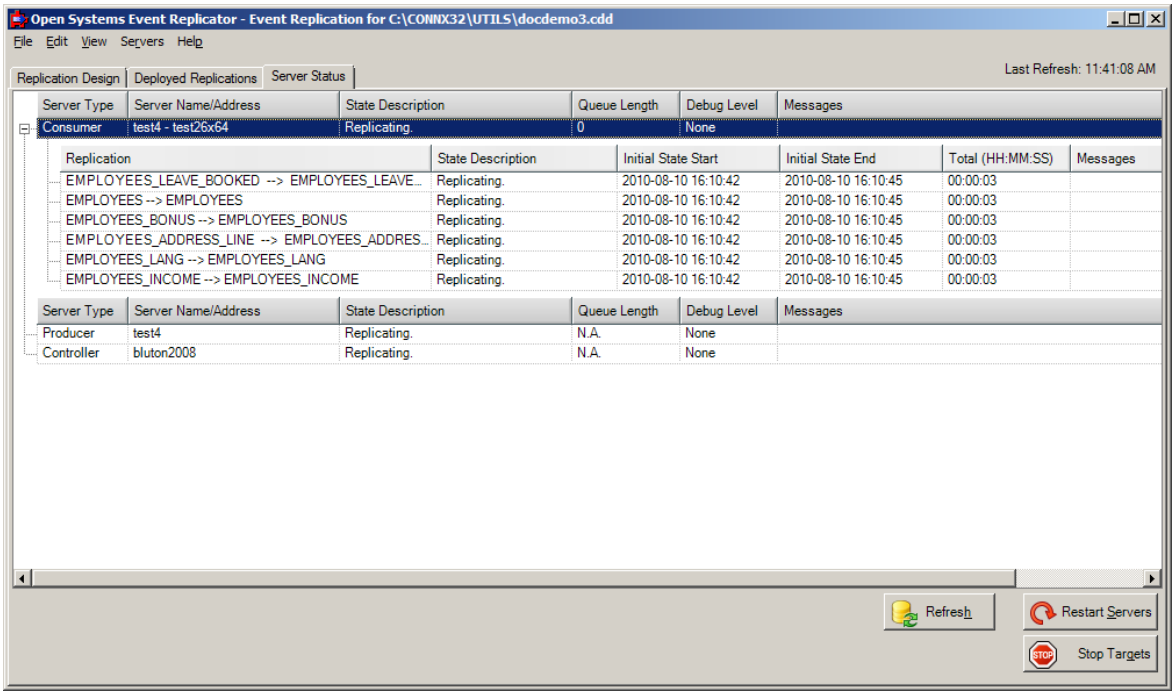
1. Open the Open Systems Event Replicator CDD.
2. Click the **Server Status** tab.



3. Click **Restart Servers**.



When the servers have been started, the following appears:


Open Systems Event Replicator - Event Replication for C:\CONNX32\UTILS\docdemo3.cdd
File Edit View Servers Help
Replication Design
Deployed Replications
Server Status
Last Refresh: 11:41:08 AM
Server Type
Server Name/Address
State Description
Queue Length
Debug Level
Messages
Consumer
test4 - test26x64
Replicating.
0
None
Replication
State Description
Initial State Start
Initial State End
Total (HH:MM:SS)
Messages
EMPLOYEES_LEAVE_BOOKED --> EMPLOYEES_LEAVE...
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES --> EMPLOYEES
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES_BONUS --> EMPLOYEES_BONUS
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES_ADDRESS_LINE --> EMPLOYEES_ADDRES...
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES_LANG --> EMPLOYEES_LANG
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES_INCOME --> EMPLOYEES_INCOME
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
Server Type
Server Name/Address
State Description
Queue Length
Debug Level
Messages
Producer
test4
Replicating.
N.A.
None
Controller
bluton2008
Replicating.
N.A.
None
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.

Note: The Message Queue should be started before the Event Server and the Adabas nucleus that is involved in replication. If the Message Queue is not started first, error messages will appear in the log files indicating that either the EP or the Controller could not contact the Message Queue. These messages will continue to appear until the Message Queue is started. When stopping services, the Message Queue should be stopped last.

Event Replicator Tutorials

About Event Replicator Tutorials

Each event replication tutorial contains a script listing the topics you need to follow to perform common replication activities. Click on a topic title in the script and the topic instructions appear.

You can find generic information about the topics in the event replication scripts in the **Using the Event Replicator** chapter.

Performing a Simple Replication - Adabas to Relational Database

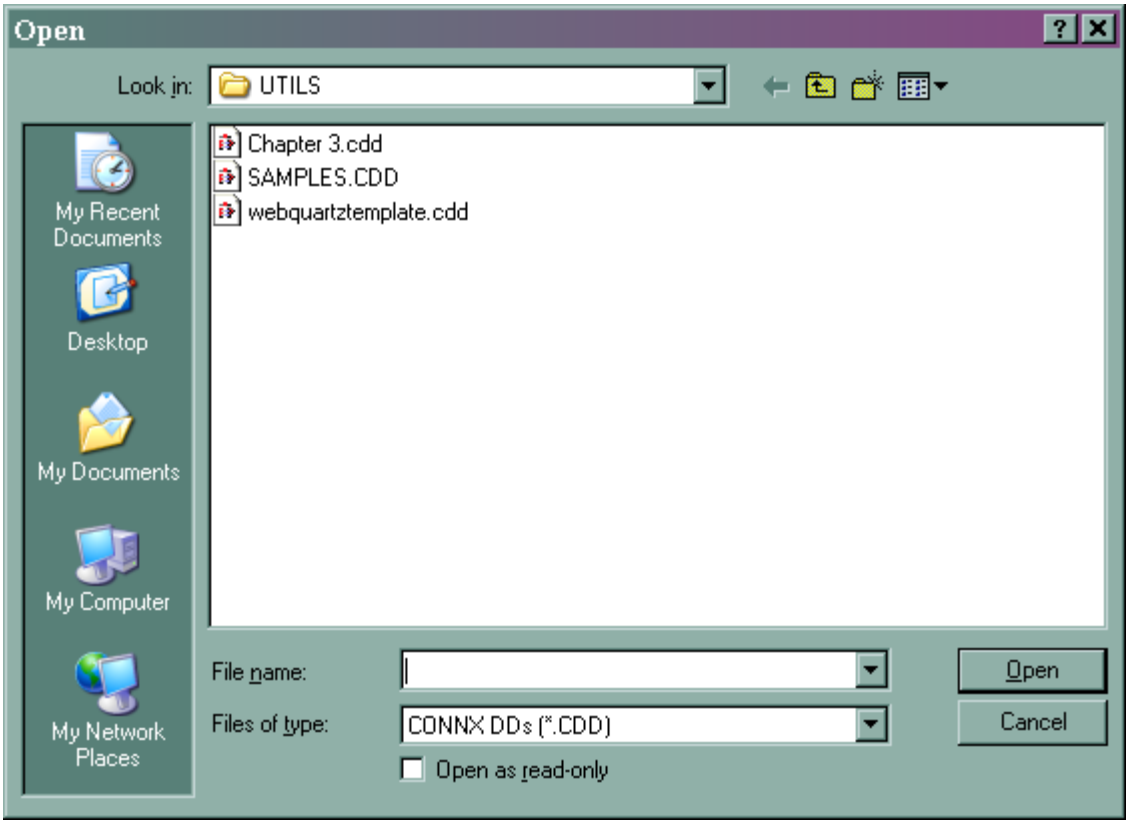
This tutorial replicates a table from an Adabas database to a SQL Server database. If you do not have access to SQL Server, Microsoft provides a free version called SQL Server Express Edition. For more information see http://msdn.microsoft.com

Prerequisites:

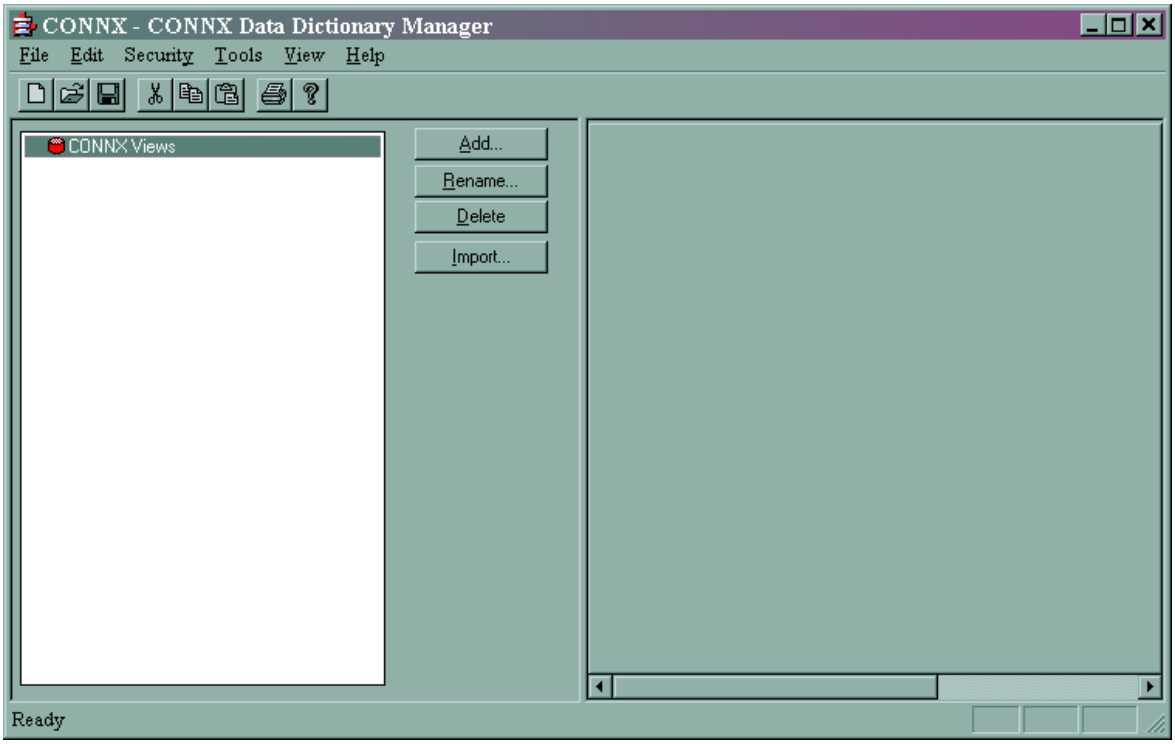
- Install Adabas version 6.1.4 or above.
- Install ConnecX SQL Engine 1.1.2 or above
- Install the Open Systems Event Replicator
- Create an Adabas database with a DBID of 1 using the Adabas sample tables. For this example, we will be using the Adabas sample table EMPLOYEES-NAT (File11)
- Start the Adabas database.
- An available SQL Server database. For this tutorial, we recommend that you create a database named Replication. Otherwise, you can use any existing SQL Server database.

Create a CDD containing the source and target databases

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

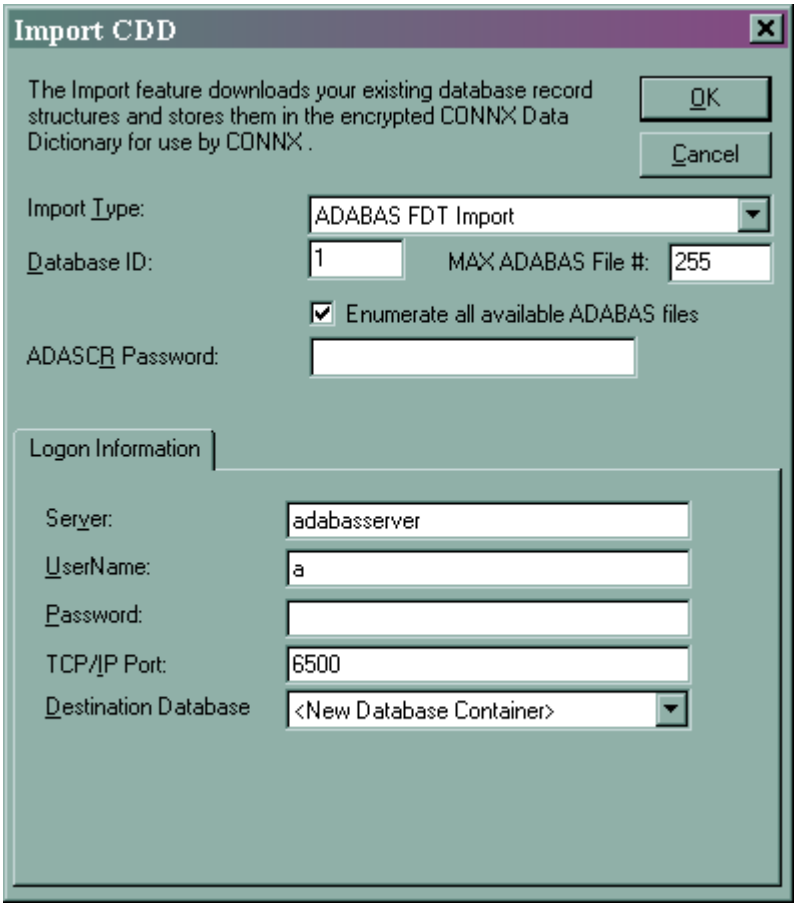


2. Click **Cancel**. The **CONNX Data Dictionary Manager** window appears.



Importing the Adabas Source Tables

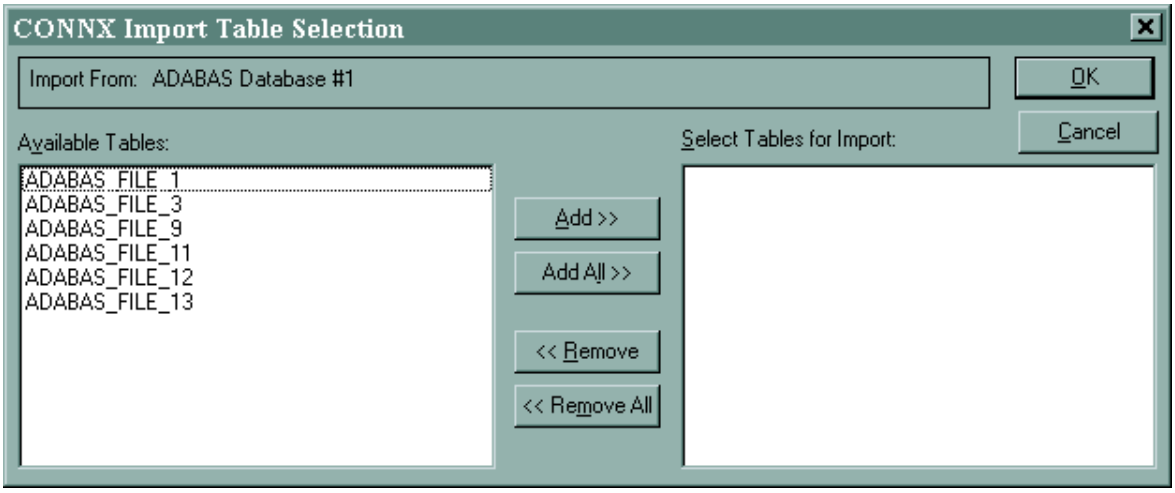
1. Click **Import**. The **Import CDD** window appears.



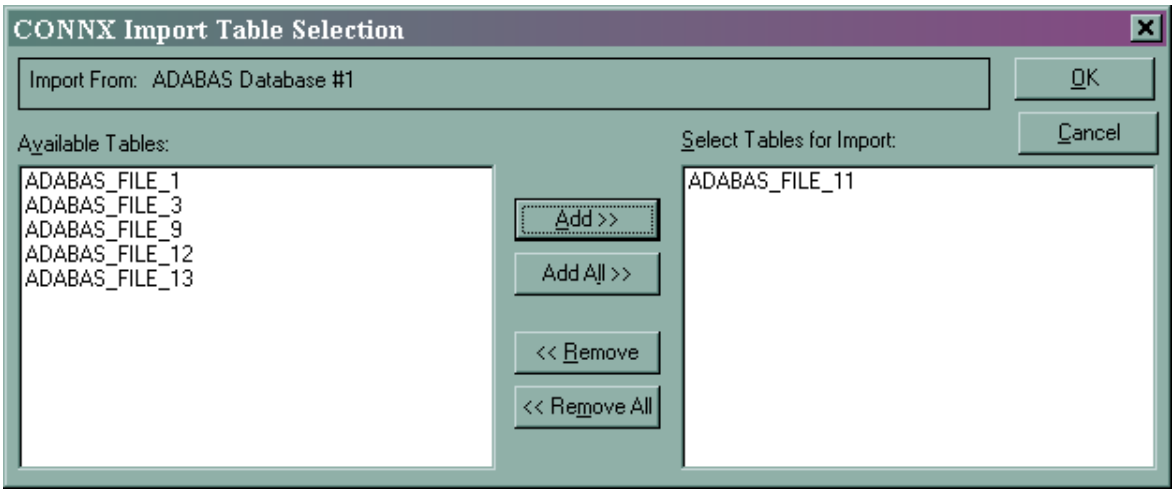
2. In **Import Type**, select FDT Import. Type 1 in **Database ID**. Type 255 in **MAX ADABAS File #**. Type in the logon information.

   Warning: Do not use localhost as the server name for the Adabas Source tables. In certain instances using localhost as the server name may cause you to lose all your source table records during an initial state.

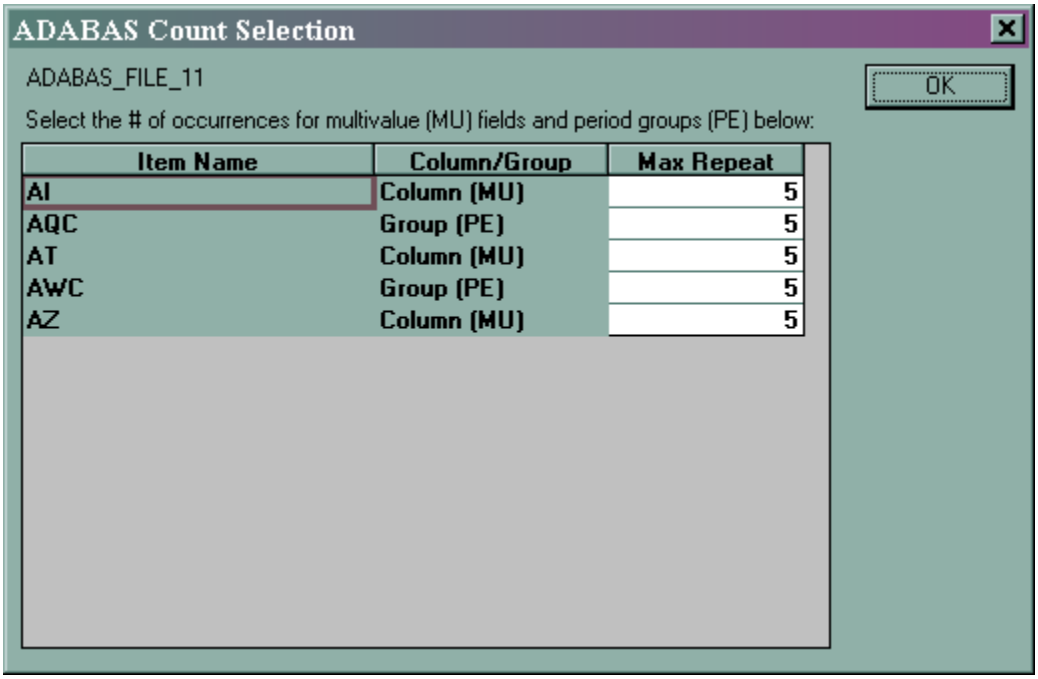
   Click **OK**. The **CONNX Import Table Selection** window appears.



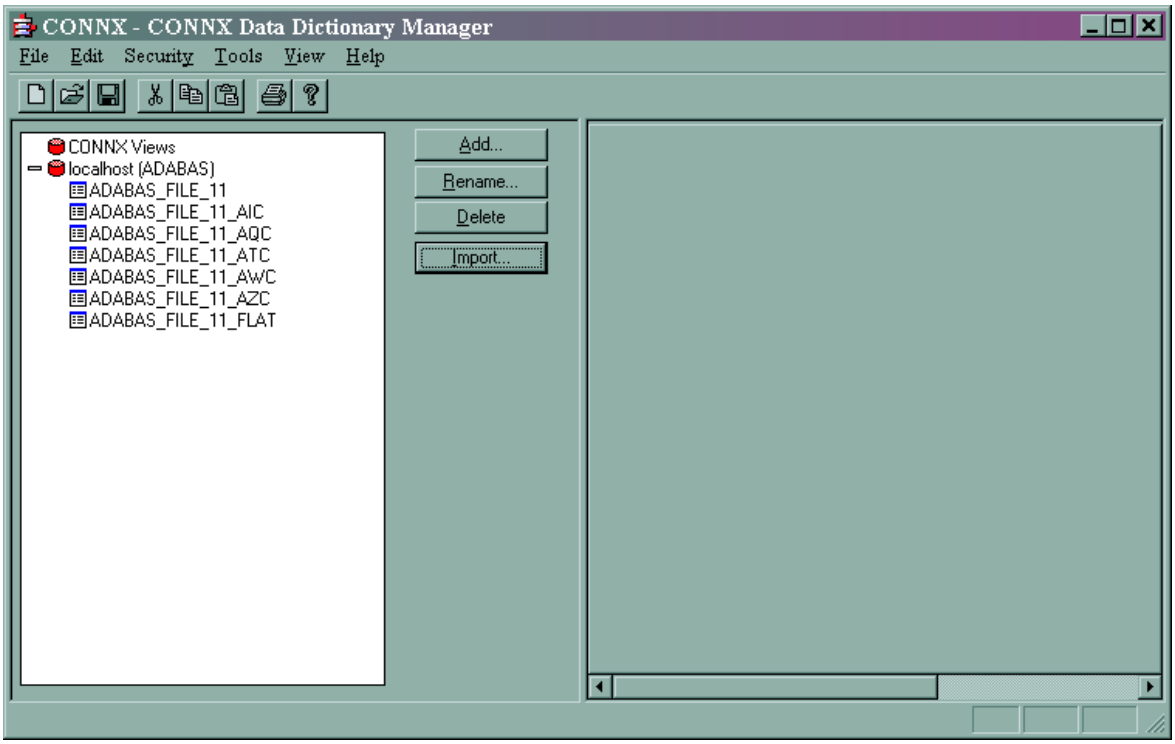
Select ADABAS_FILE_11 and click **Add**.



3. Click **OK**. The **ADABAS Count Selection** window for the source table appears.

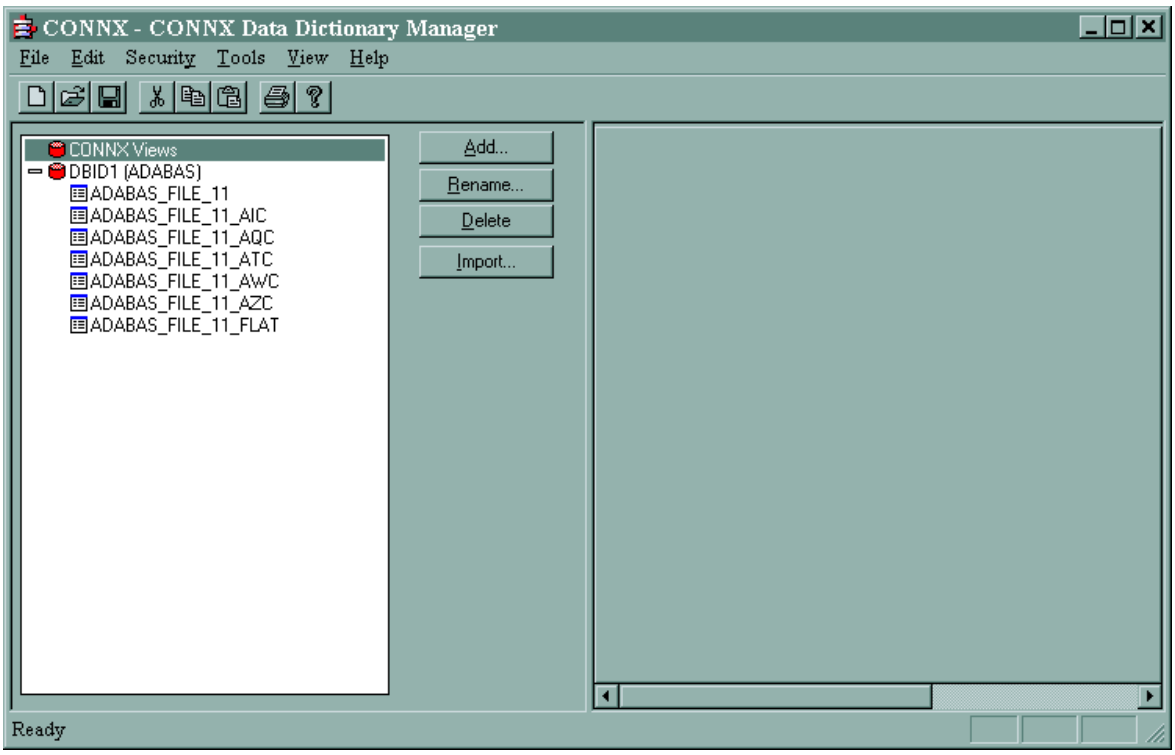


4. Accept the default values and click **OK**. The **CONNX Data Dictionary Manager** window containing the source table information appears.



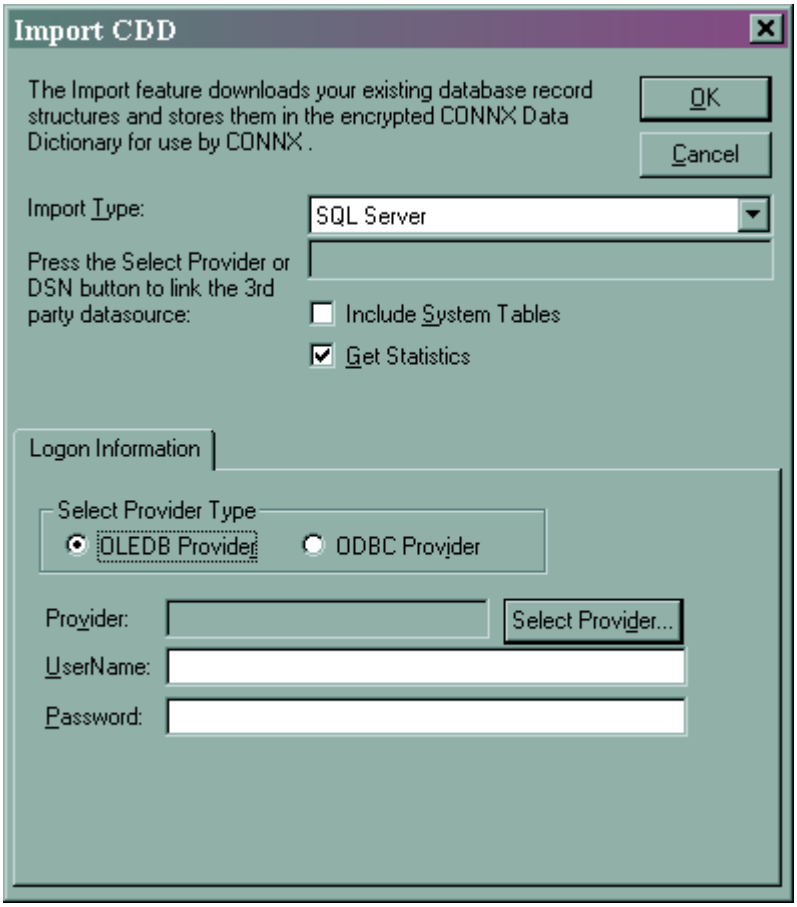
You have imported the Adabas source tables into the CDD.

5. Rename the database from localhost to DBID1. In the CONNX Data Dictionary window, click **localhost (ADABAS)**.Click **Rename. O**n th**e Rename Object** window type DBID1.



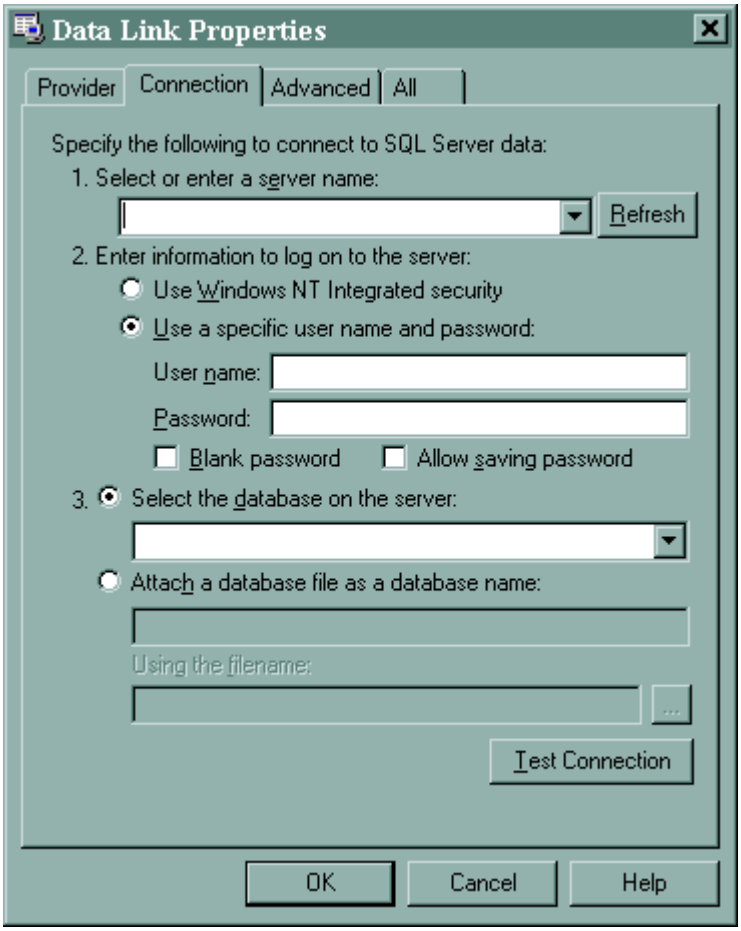
Importing the Target Database

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



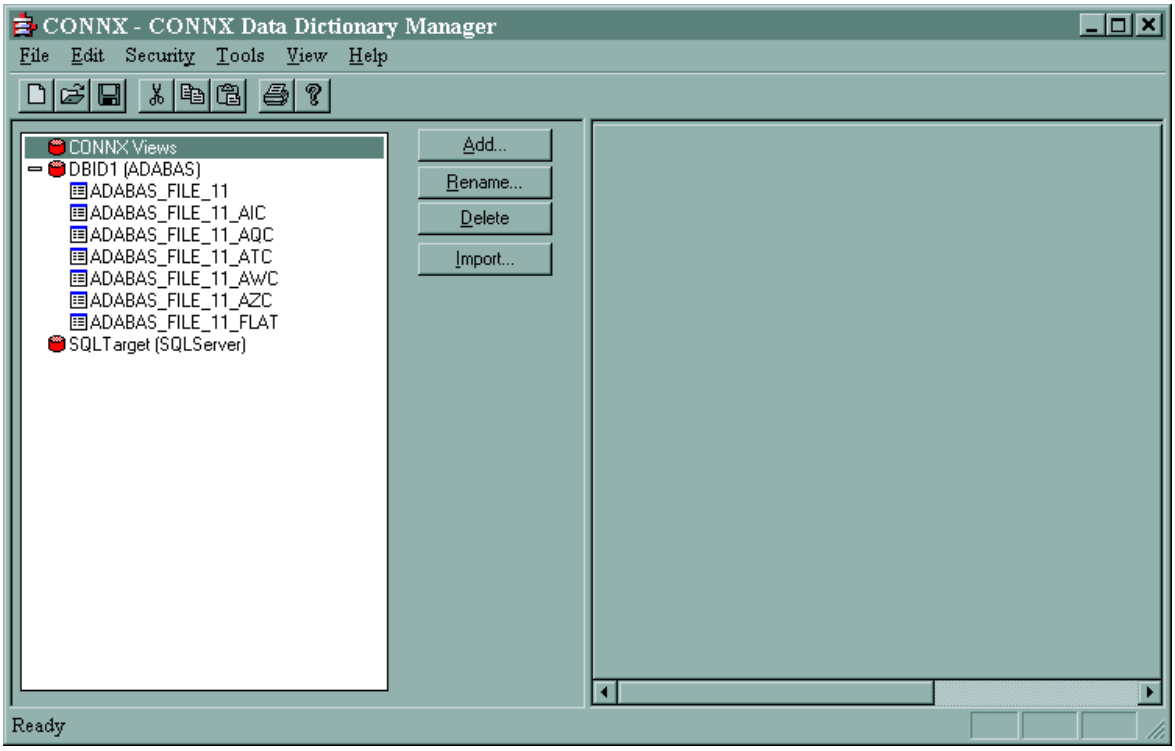
2. In **Import Type**, select SQL Server.

3. In **Select Provider Type**, click **OLEDBC Provider**. Click **Select Provider**. The **Data Link Properties** window appears. Select **Microsoft OLEDB Provider for SQL Server** and click **Next**. The **Data Link Provider** appears.



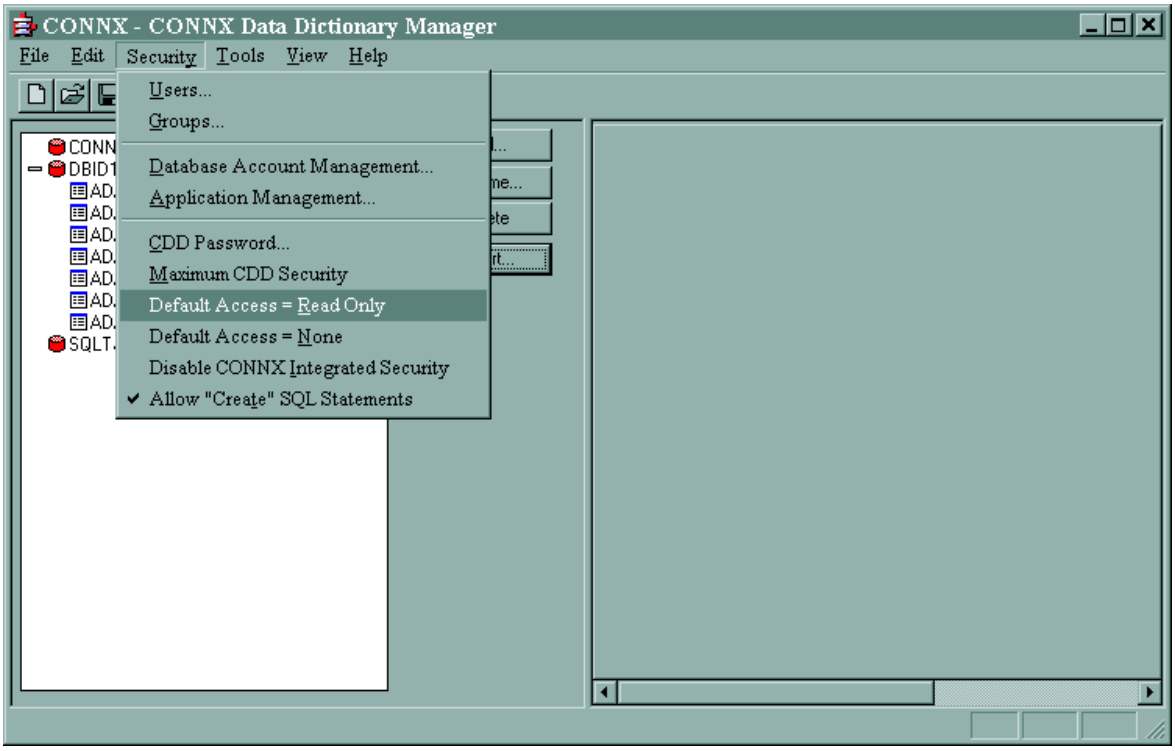
Fill in the window based on your SQL Server installation and click **OK**. The **CONNX Import Table Selection** window appears. If there are any tables in your SQL Server database, they will appear in **Available Tables** (in this example, we created a new SQL Server database so there are no tables).. Click **Add All** and then click **OK**. The **CONNX Data Dictionary Manager** window appears.

Rename the target database to SQLTarget.

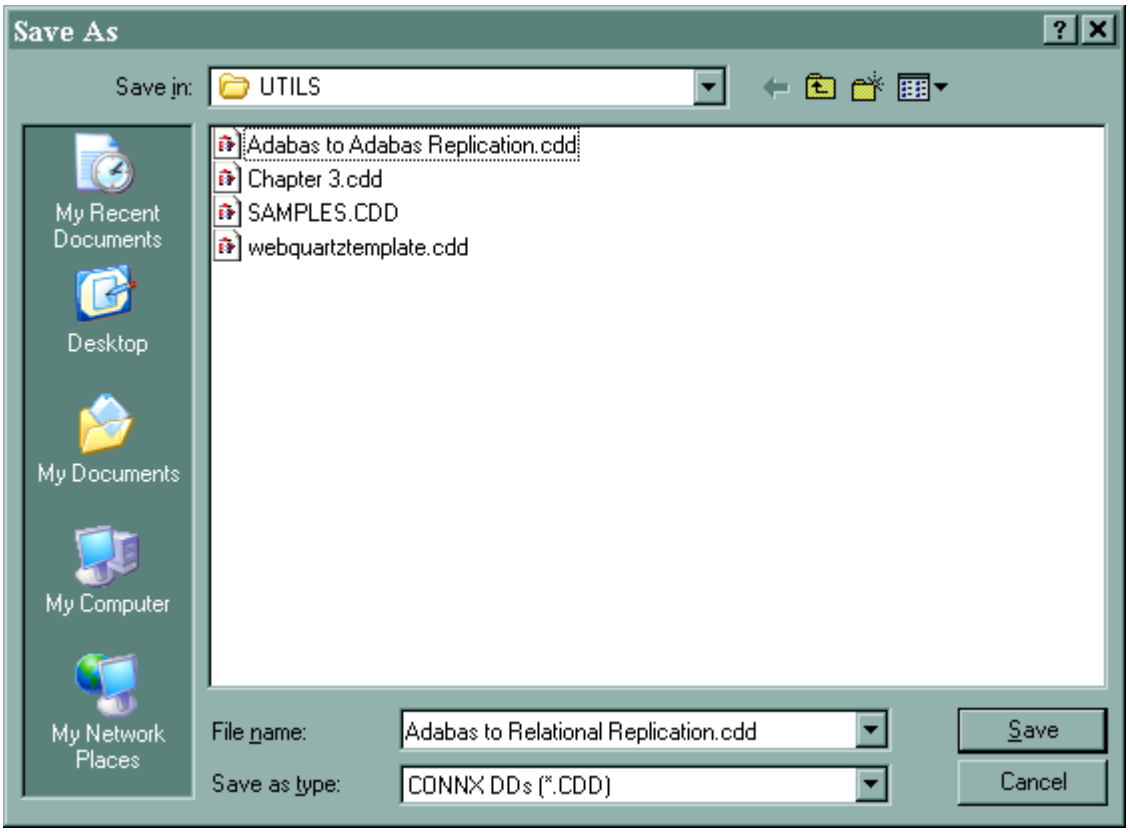


Set the CDD Security

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



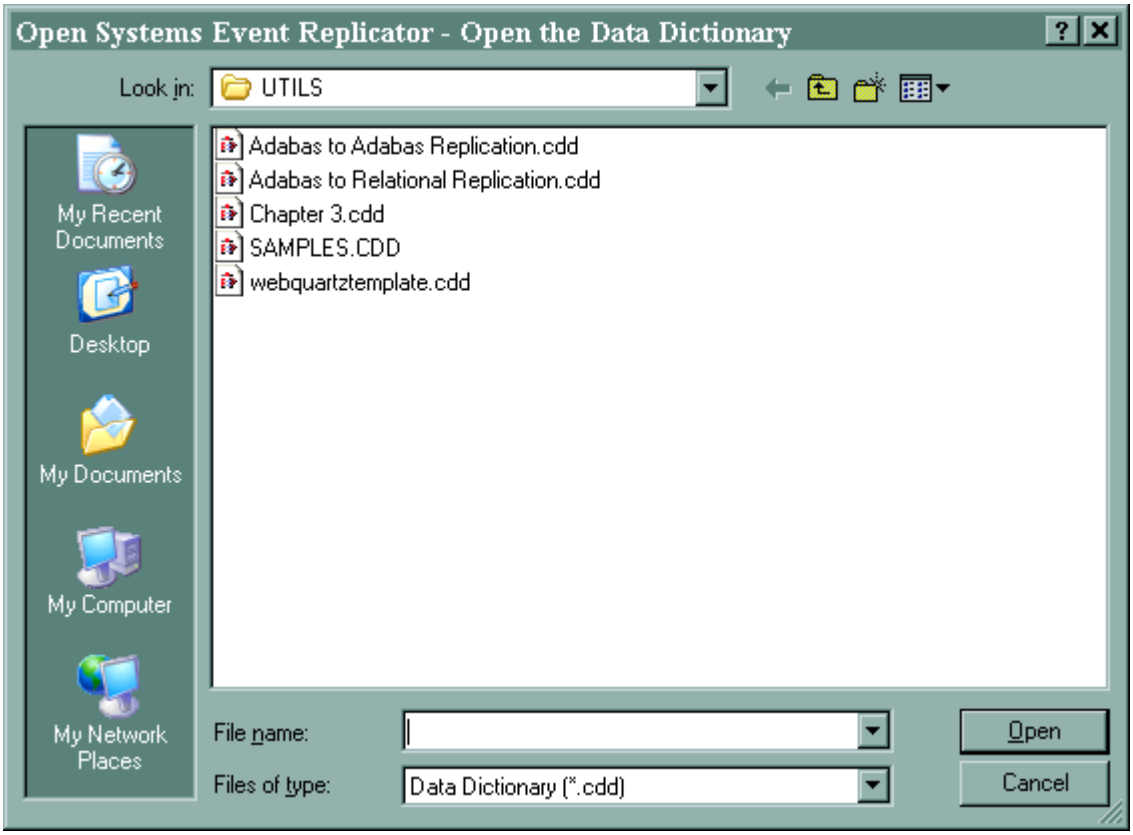
2. In the **File** menu, select **Save As**.



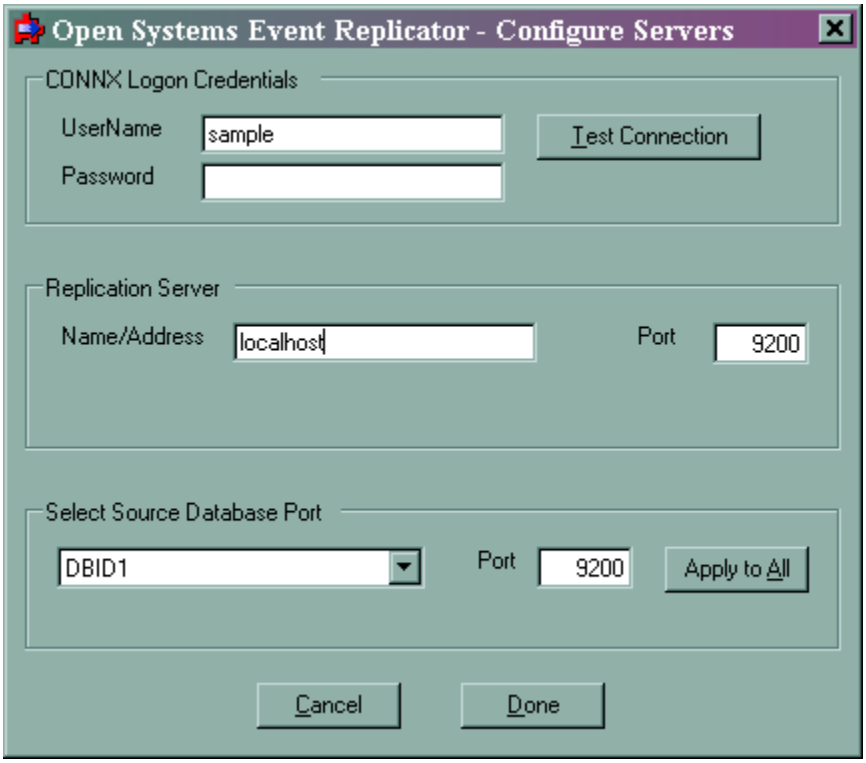
3. Name your **CDD Adabas to Relational Replication** and click **Save**.
4. Close the CONNX Data Dictionary Manager.

Create a Replication

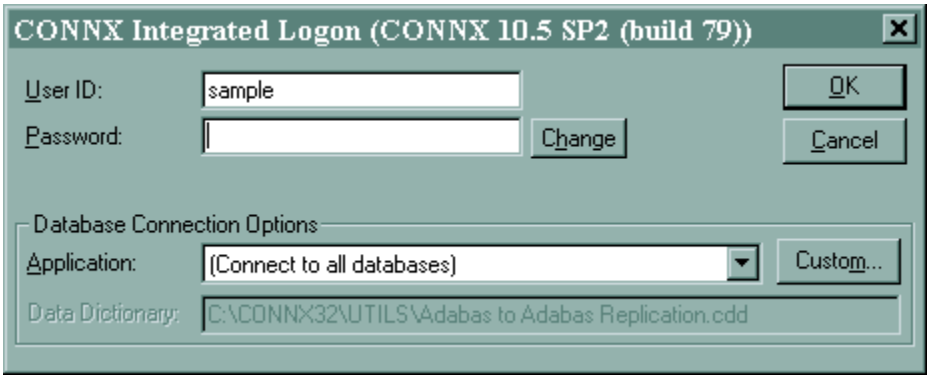
1. On the **Start** menu, click **Programs**, click **CONNX Solutions**, click **Open Systems Event Replicator** and then click **Replication Administrator.** The **Open the Data Dictionary** window appears.



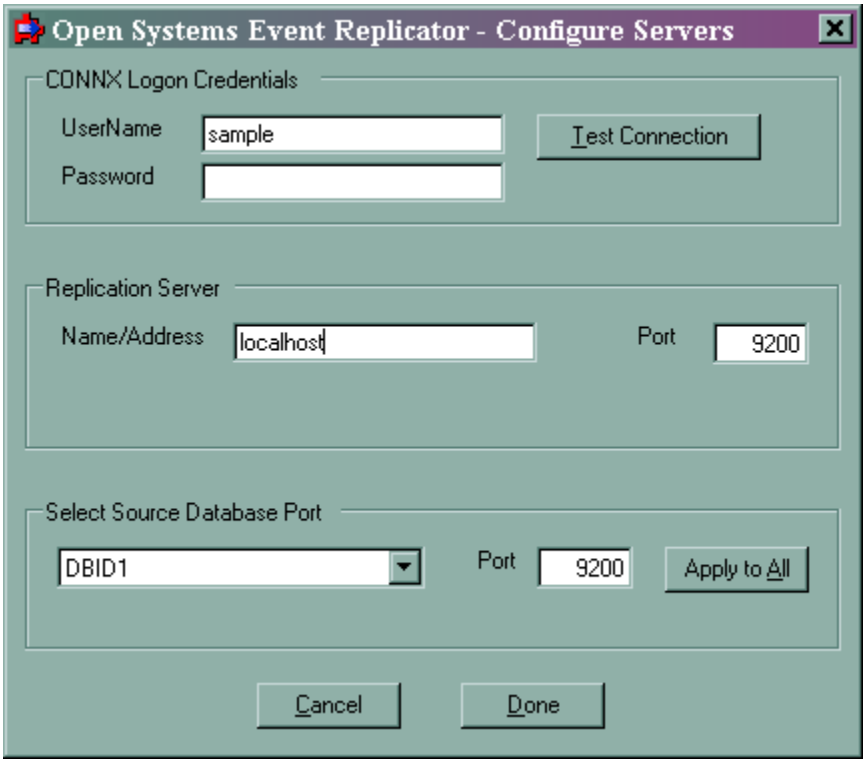
3. Select a **Adabas to Relational Replication.cdd** and click **Open**. The **Configure Servers** window appears.



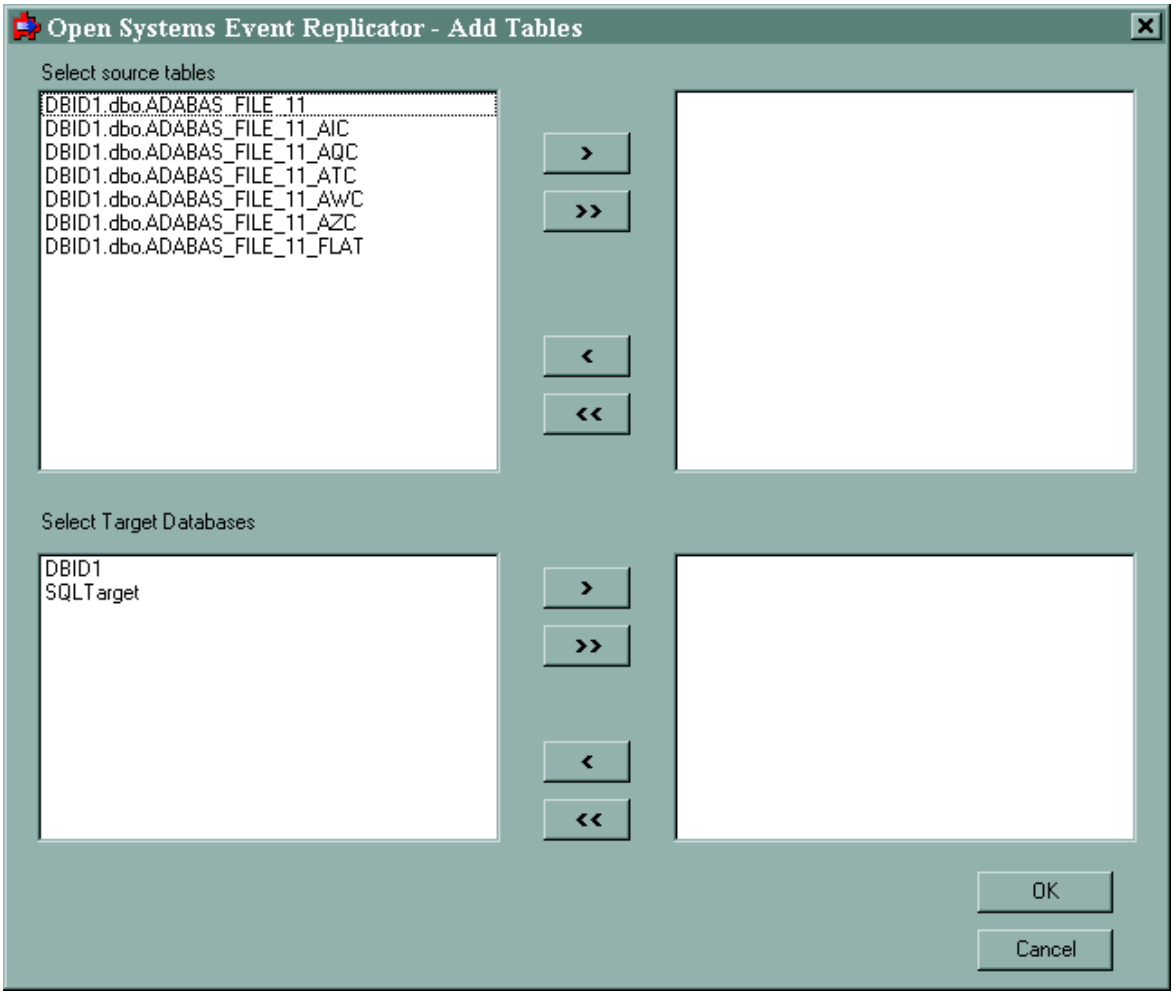
4. In **CONNX Logon Credentials** enter your **Sample** in **UserName;** leave the Password blank. Click **Test Connection**. The **CONNX Integrated Logon** window appears:



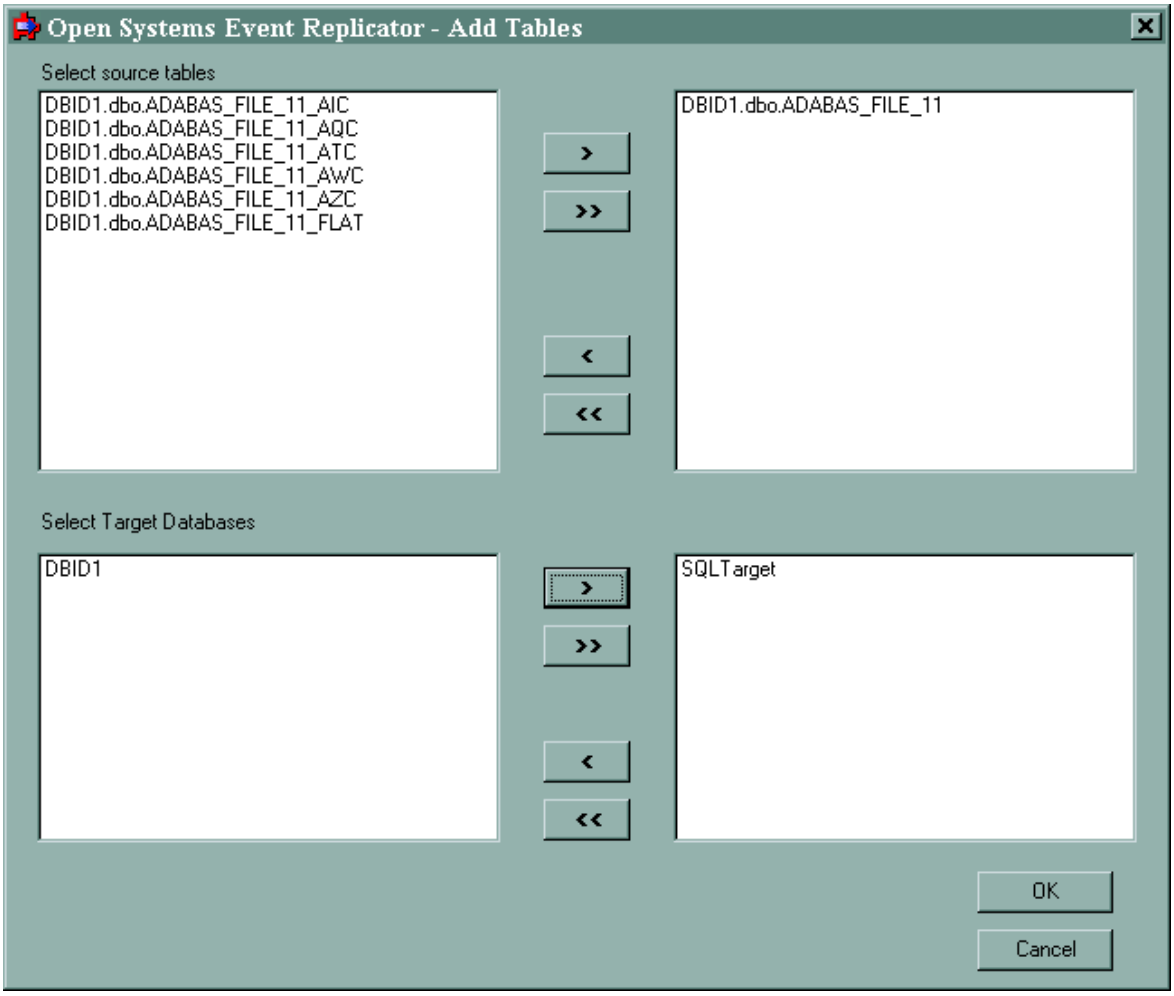
5. Leave the Password blank and click **OK**.
6. You will be prompted to create a user in the CDD. Press **OK.** The User ID **sample** will be created with a blank password.
7. When the connection succeeds, enter **localhost** in **Name/Address** and **9200** in **Port**.



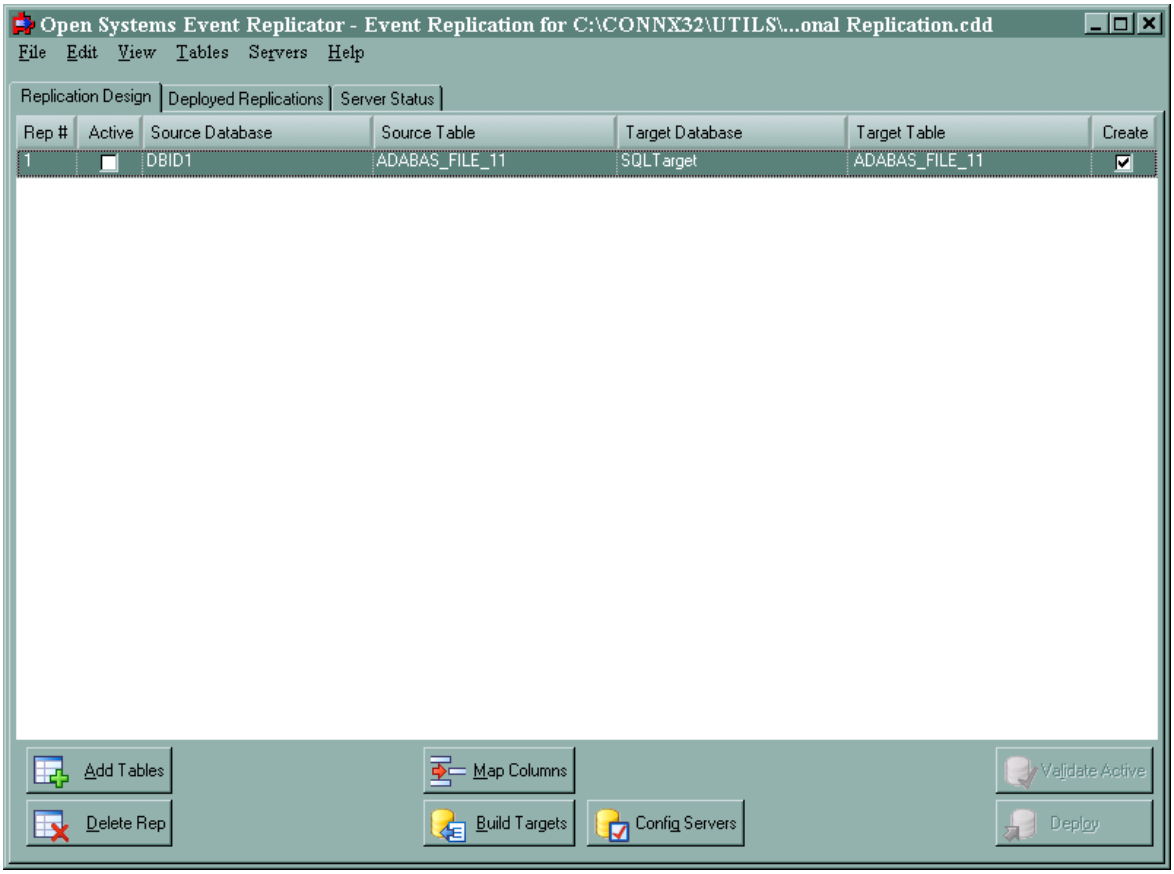
8. Click **Done**. The **Add Tables** window appears.



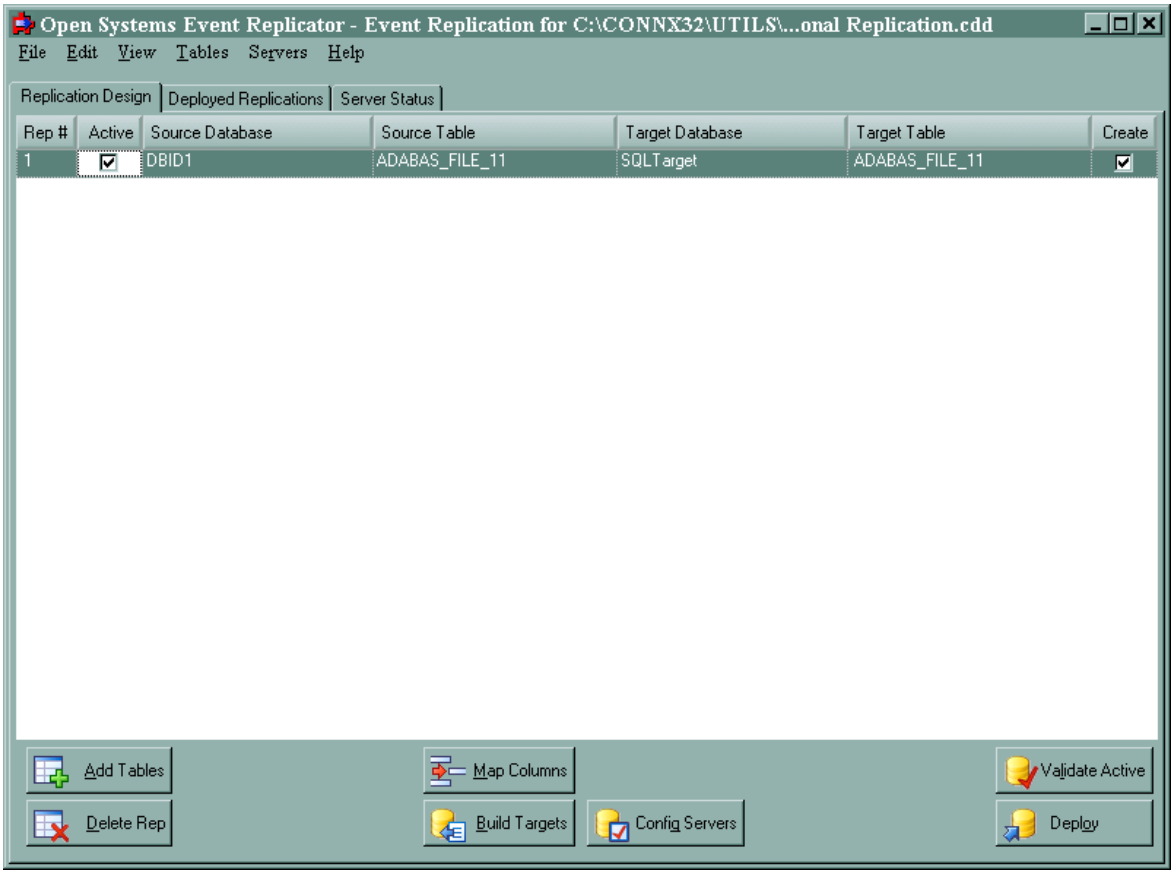
9. From **Select source tables,** select **DBID1.dbo.ADABAS_FILE_11.** From **Select Target Databases**, select **SQLTarget.**



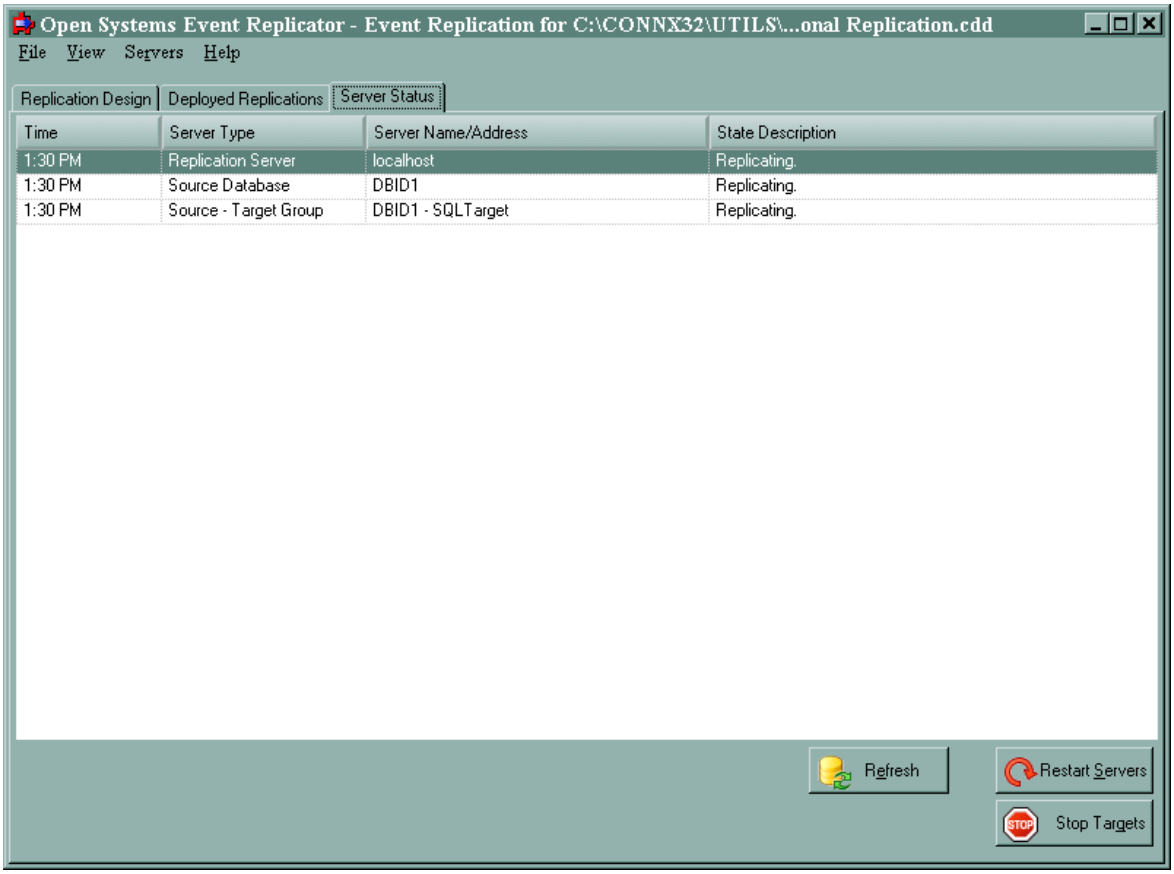
10. Click **OK**. The **Open Systems Event Replication** window appears.



Click **Active** and then **Deploy**.



11. Click the **Server Status** tab. Verify that the **Replication Server**, **Source Database** and **Source - Target Group** contain **Replicating** in **State Description**.



This completes a simple Adabas to relational database replication. You can do inserts, updates and deletes against the ADABAS_FILE_11 in database 1 and the changes will replicate to your SQL Server database.

Adding a new Replication and Redeploying

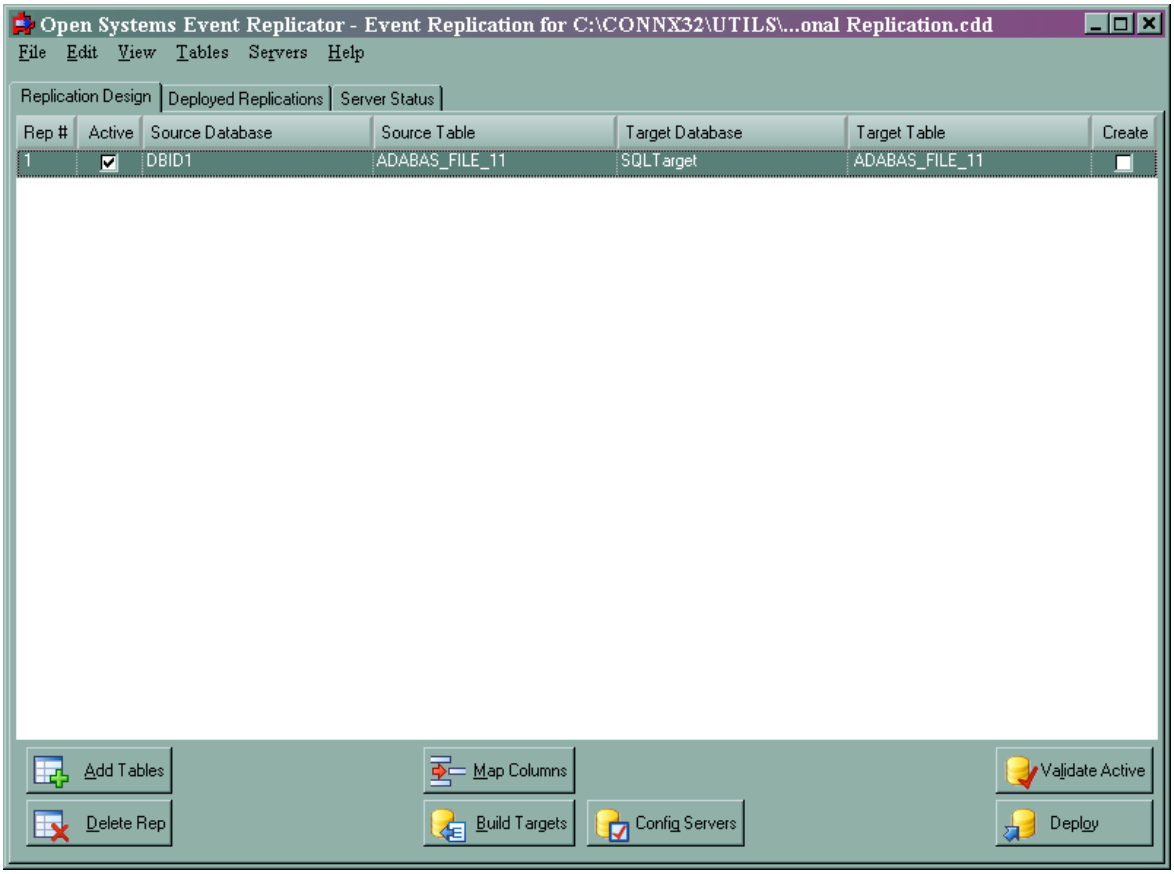
This tutorial adds a new replication from an Adabas database to a SQL Server database. If you do not have access to SQL Server, Microsoft provides a free version called SQL Server Express Edition. For more information see http://msdn.microsoft.com

Prerequisites:

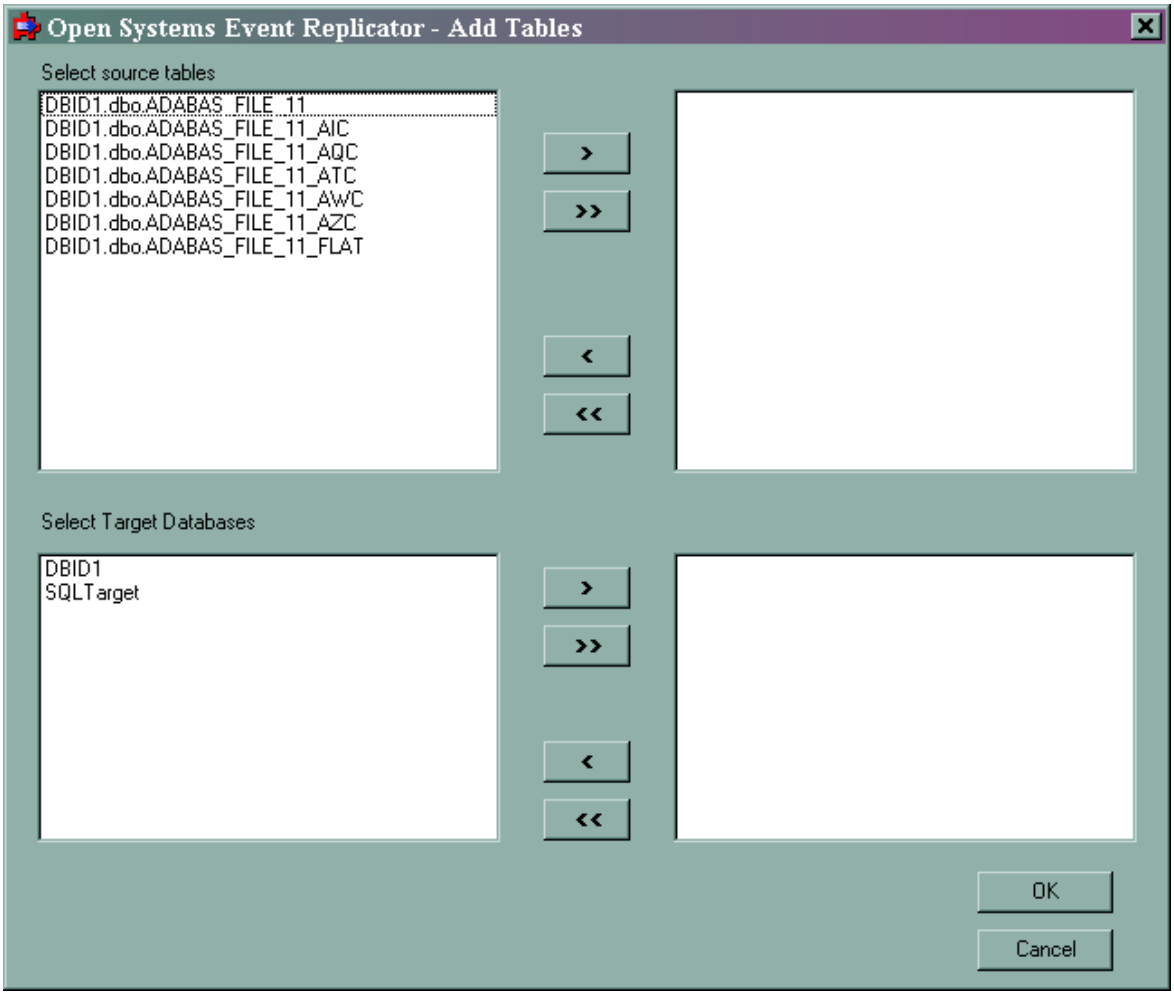
- Install Adabas version 6.1.4 or above.
- Install ConnecX SQL Engine 1.1.2 or above
- Install the Open Systems Event Replicator
- Create an Adabas database with a DBID of 1 using the Adabas sample tables. For this example, we will use the Adabas sample table **EMPLOYEES-NAT (File11)**
- Start the Adabas database.
- An available SQL Server database. For this tutorial, we recommend that you create a database named Replication. Otherwise, you can use any existing SQL Server database.
- Create a CDD containing the source and target databases. For this example, we will use the CDD **Adabas to Relational Replication** (our last opened CDD) that was created in the Performing a Simple Replication - Adabas to Relational Database tutorial.
- Replicate (create and deploy) one of the source tables to the target database. In this example, we will assume that table **DBID1.dbo.ADABAS_FILE_11** has been replicated to database **SQLTarget.**

Open the Event Replicator CDD and add the source table to the target database

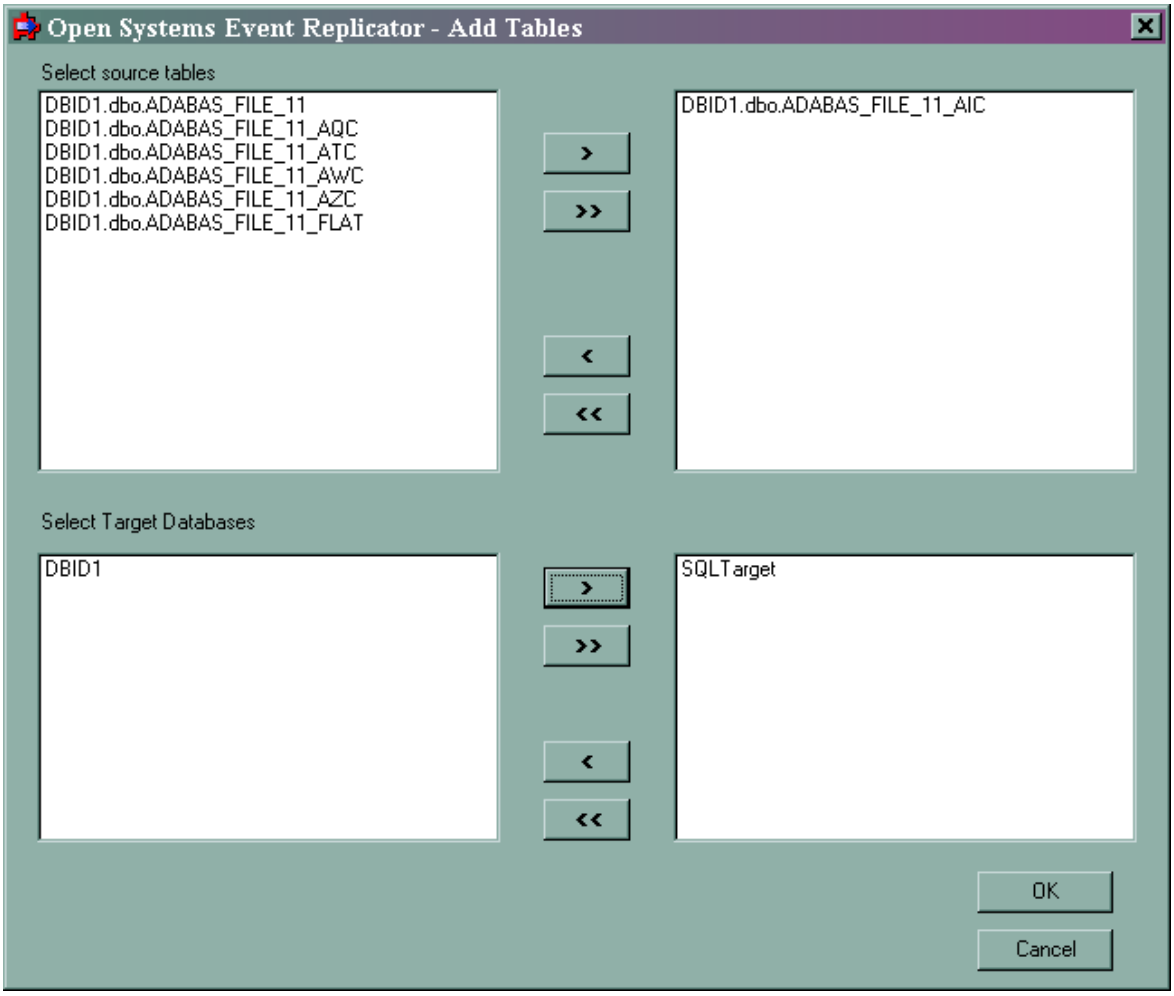
1. On the **Start** menu, click **Programs**, click **CONNX Open Systems Event Replicator** and then click **Replication Administrator**. The **Open Systems Event Replication** window with the information from the **Adabas to Relational Replication** CDD appears.



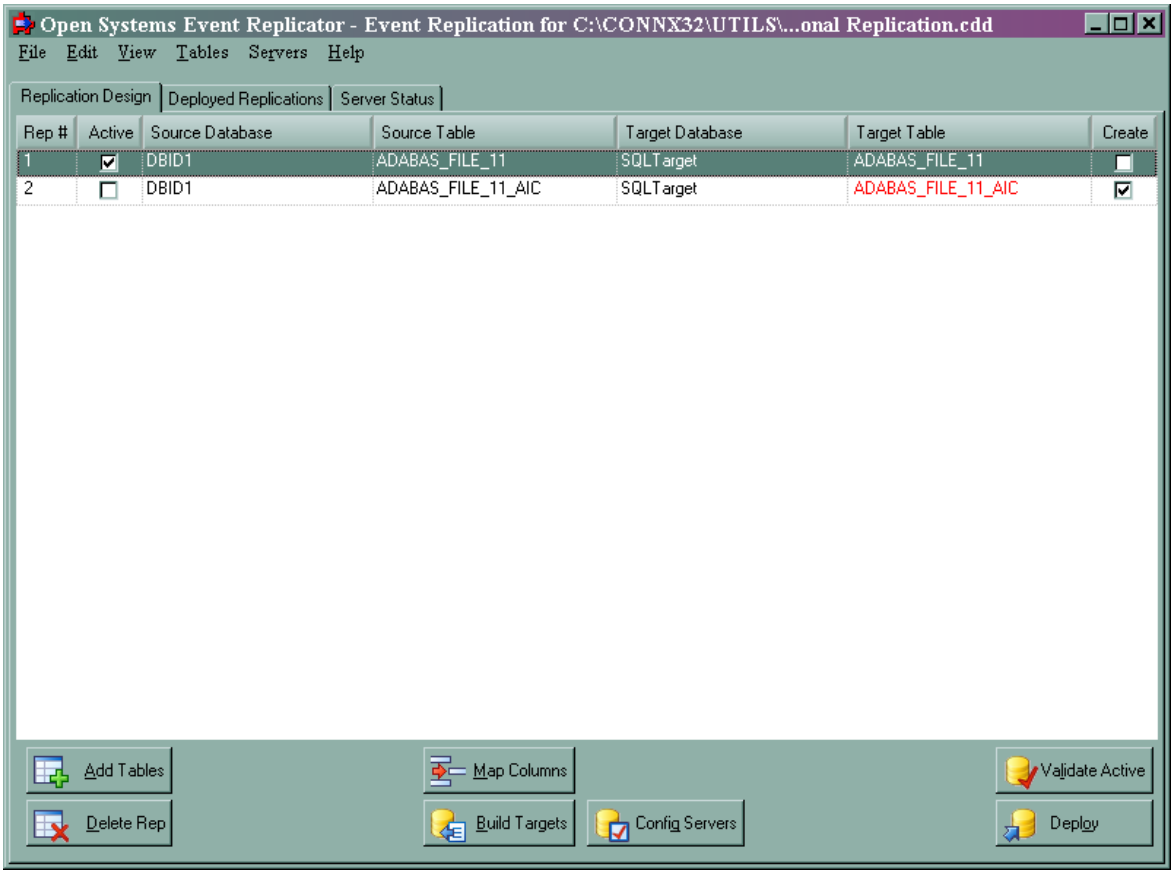
2. Click **Add Tables**. The **Add Tables** window appears.



3. Select **DBID1.dbo.ADABAS_FILE_11_AIC** from **Select source tables** and **SQLTarget** from **Select Target Databases.**



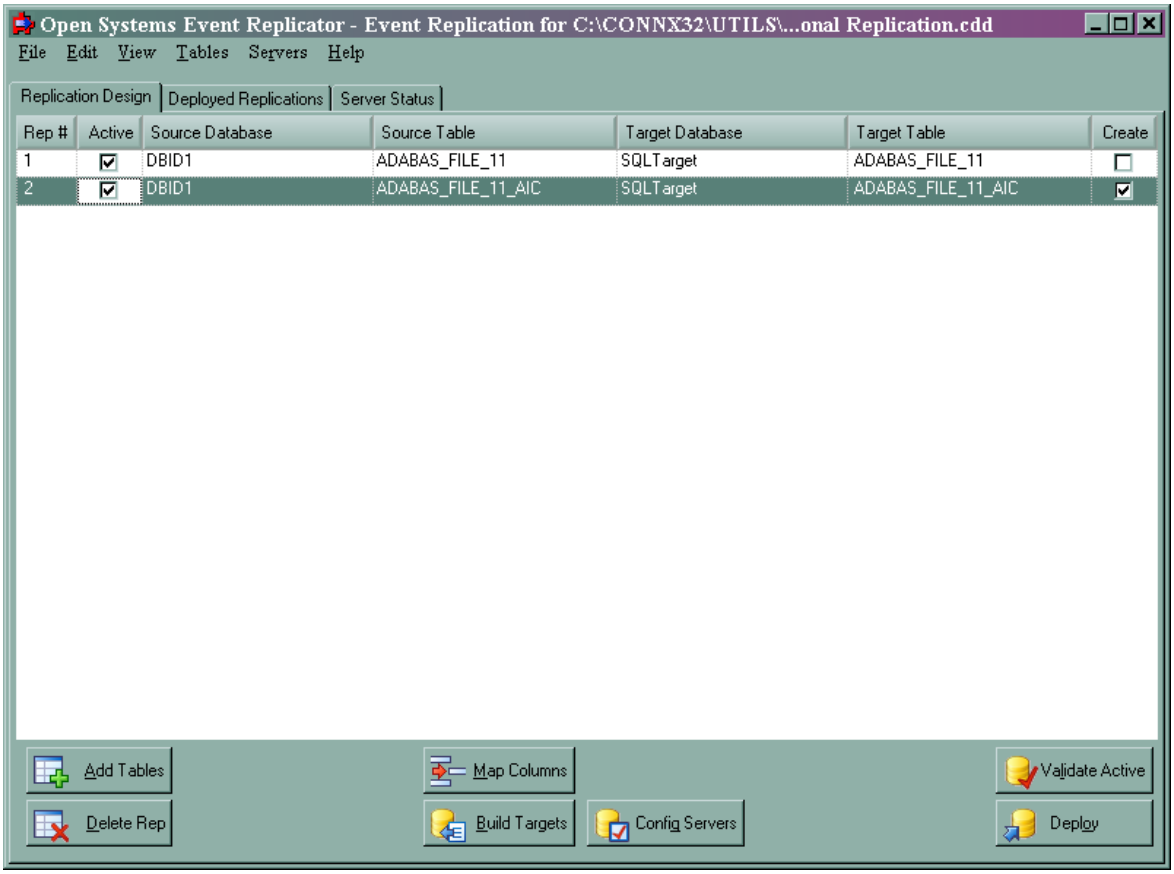
4. Click **OK**. The Open Systems Event Replication window appears and the **ADABAS_FILE_11_AIC** table you just added is in the list.



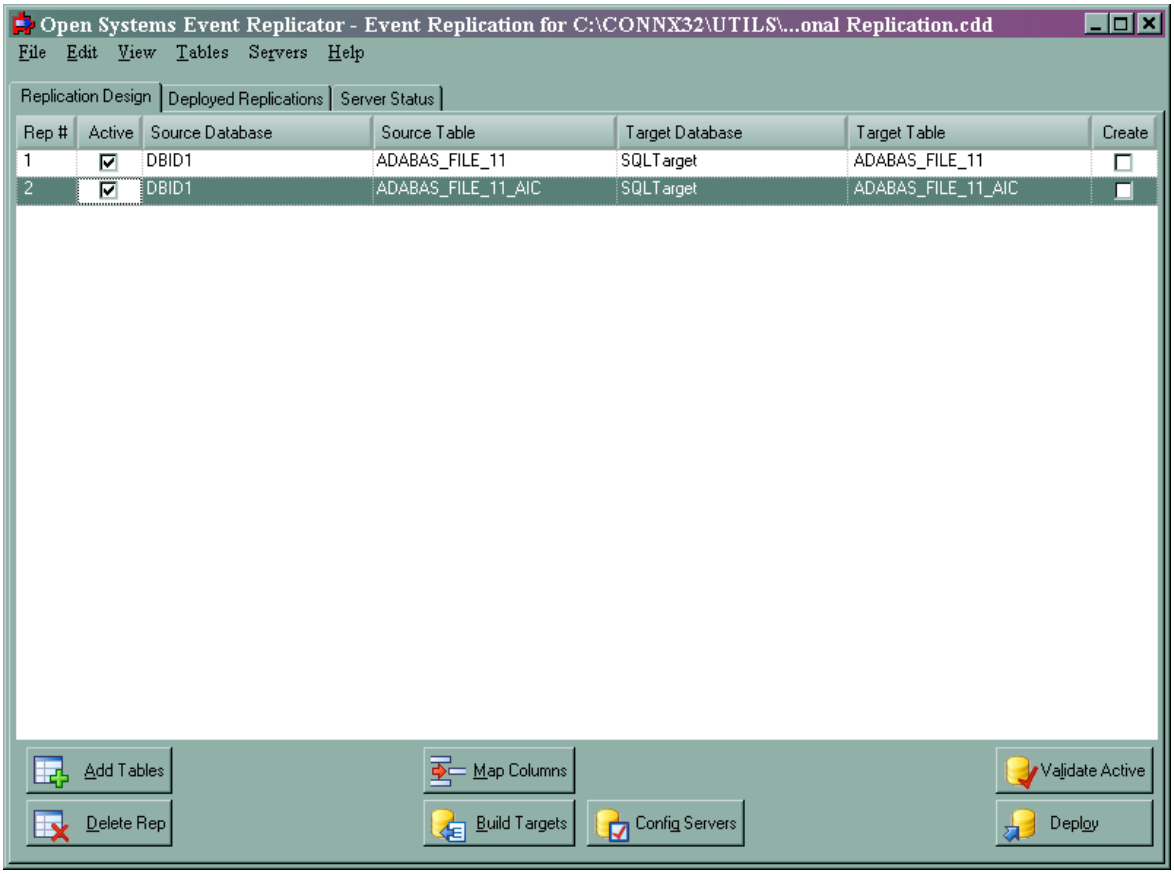
Because **ADABAS_FILE_11_AIC** has not been deployed, its replication line has the **Target Table** name in red, the **Active** check box cleared and the **Create** check box selected.

Redeploy the active Replications in the CDD

1. To deploy **ADABAS_FILE_11_AIC**, go to the **Rep #** 2 line and click **Active.**



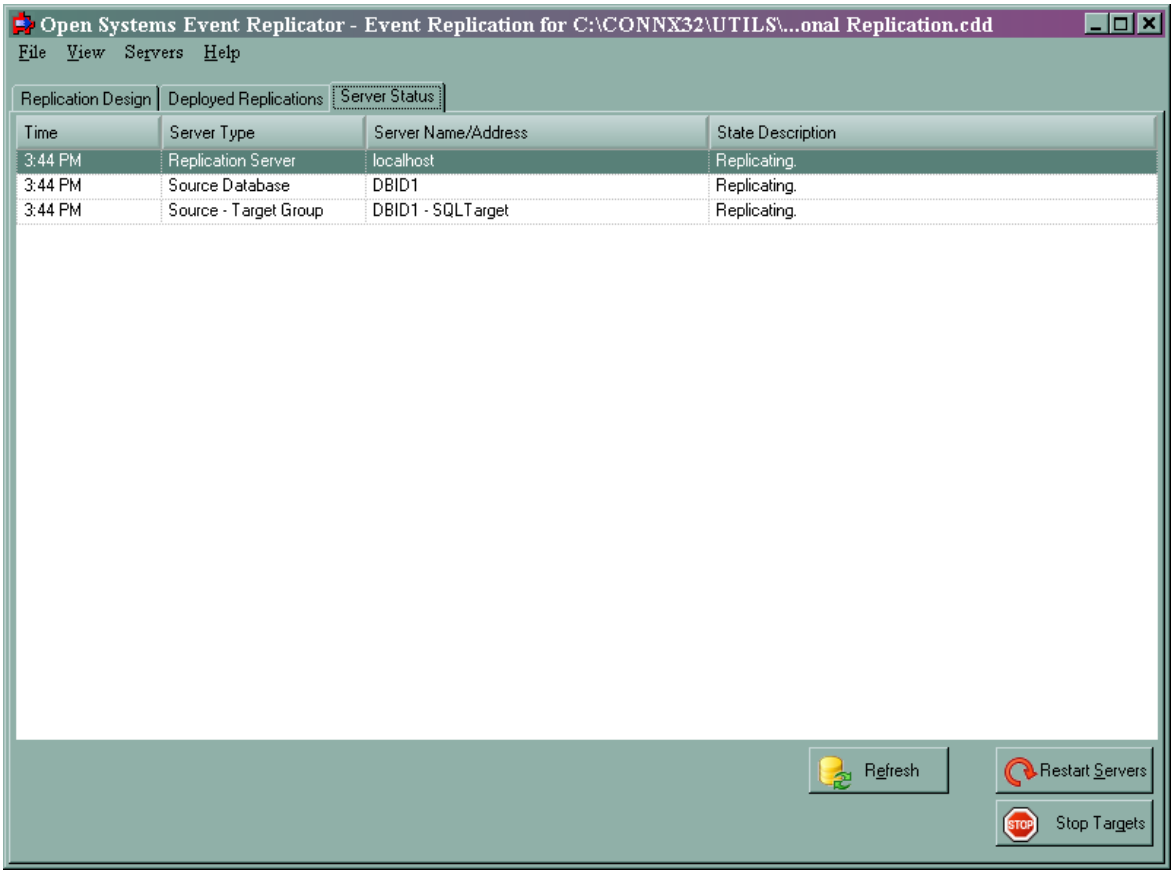
2. Click **Validate Active**. The **Passed Validation** message appears. Click **OK** and then click **Deploy**. When the **Active Replications Deployed to Controller** message appears, click **OK**.



Each deployed replication has the **Active** check box selected and the **Create** check box cleared.

Validate the deployment

Click the **Server Status** tab. Verify that the **Replication Server**, **Source Database** and **Source - Target Group** contain **Replicating** in **State Description**.



This completes adding a new replication and redeploying all the active replications. You can do inserts, updates and deletes against both the ADABAS_FILE_11 and ADABAS_FILE_11_AIC in database 1 and the changes will replicate to your SQL Server database.

Removing a Replication from Deployment

This tutorial removes a replication from the active replications.

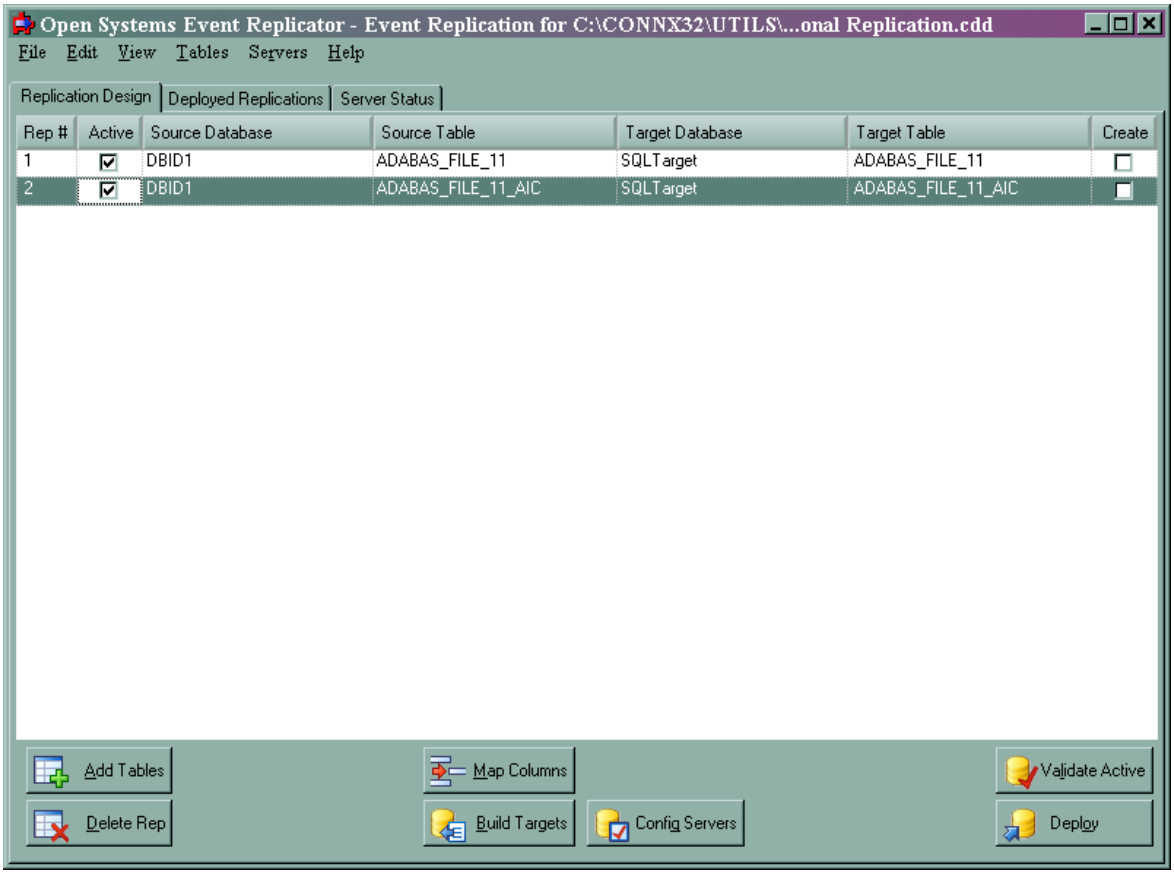
Prerequisites:

- Install Adabas version 6.1.4 or above.
- Install ConnecX SQL Engine 1.1.2or above
- Install the Open Systems Event Replicator
- Create an Adabas database with a DBID of 1 using the Adabas sample tables. For this example, we will use the Adabas sample table **EMPLOYEES-NAT (File11)**
- Start the Adabas database.
- An available SQL Server database. For this tutorial, we recommend that you create a database named Replication. Otherwise, you can use any existing SQL Server database.
- Create a CDD containing the source and target databases. For this example, we will use the CDD **Adabas to Relational Replication** (our last opened CDD) that was created in the Performing a Simple Replication - Adabas to Relational Database tutorial.
- Replicate (create and deploy) two source tables to the target database. In this example, we will assume that tables **DBID1.dbo.ADABAS_FILE_11** and **DBID1.dbo.ADABAS_FILE_11_AIC** have been replicated to database **SQLTarget.**

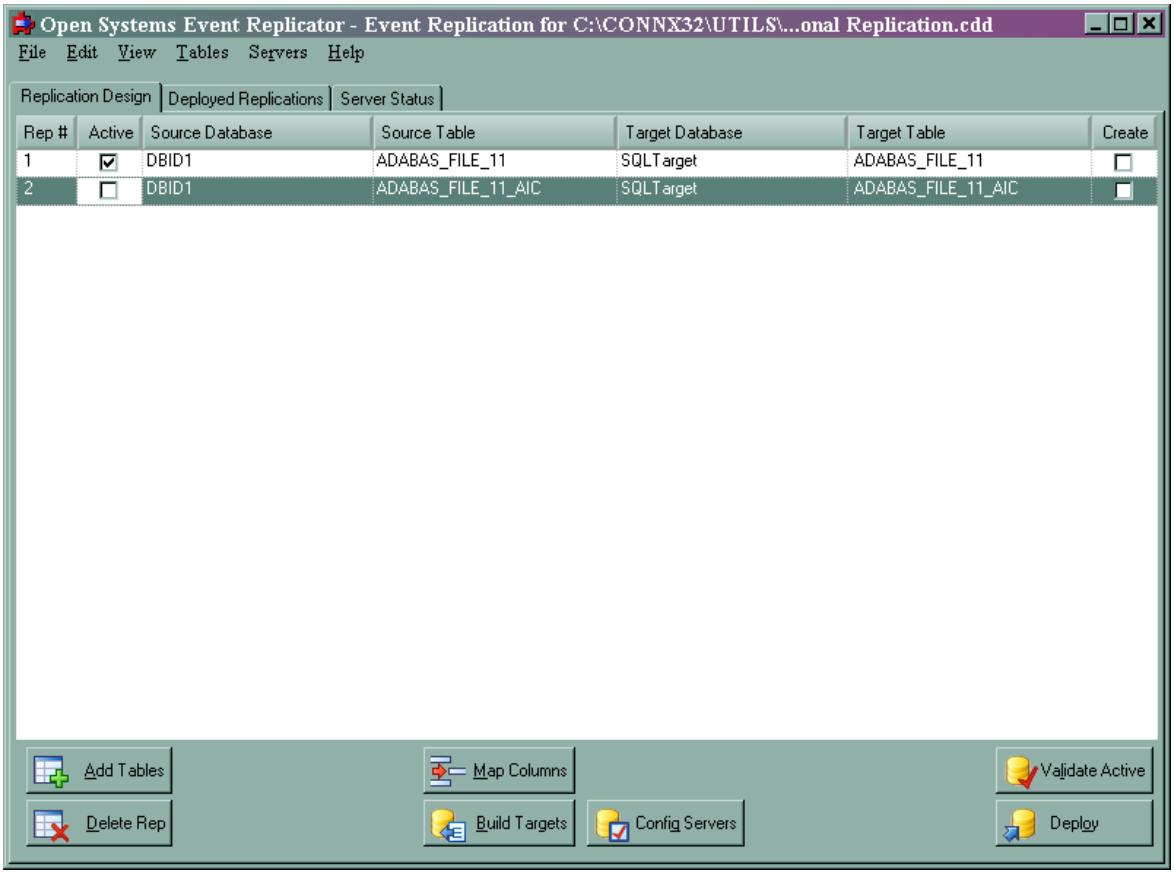
You cannot remove all the active replications at once using this method.

Open the Event Replicator CDD and un-deploy a replication

1. On the **Start** menu, click **Programs**, click **CONNX Open Systems Event Replicator** and then click **Replication Administrator**. The **Open Systems Event Replication** window with the information from the **Adabas to Relational Replication** CDD appears.



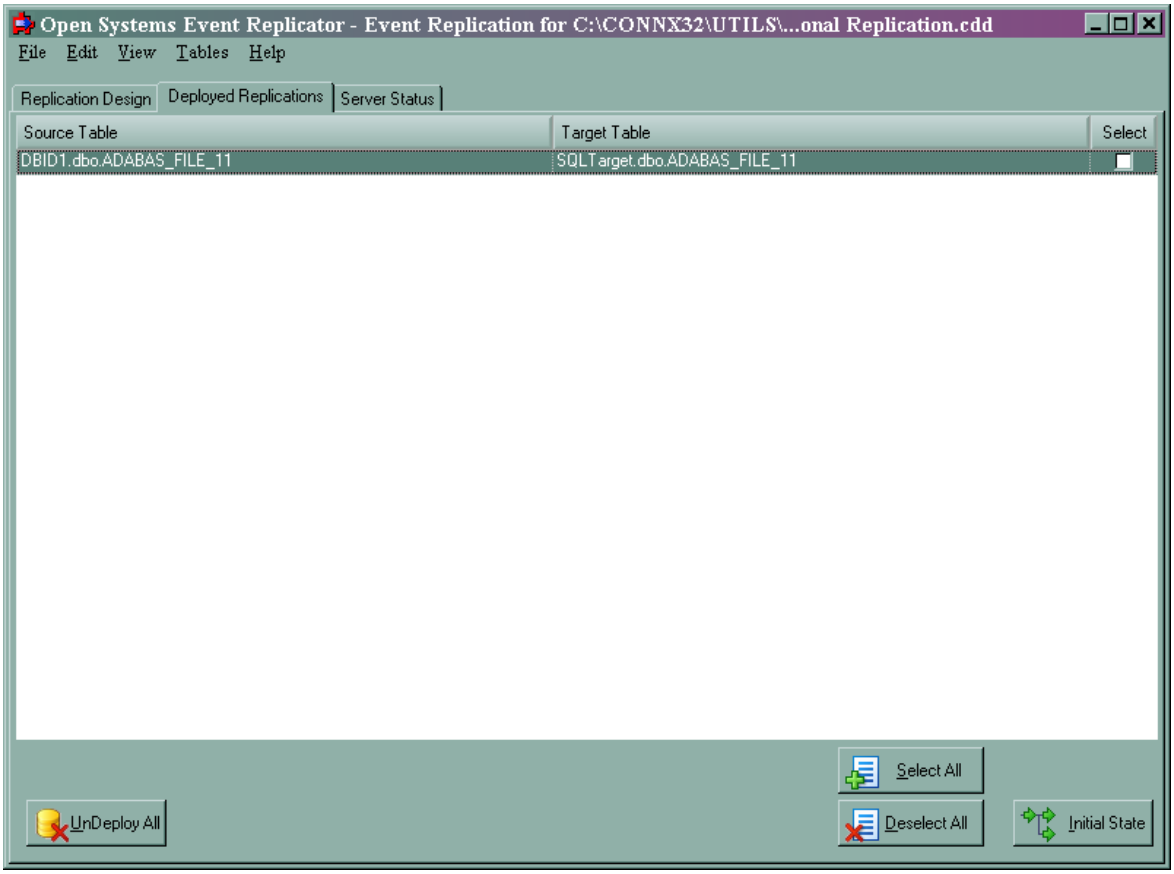
2. Select a replication to un-deploy. In this example, we will select **ADABAS_FILE_11_AIC**. On the **Rep #** 2 line, clear **Active** and click **Deploy**. The **Active Replications Deployed to Controller** message appears. Click **OK**. The updated **Open Systems Event Replication** window appears.



Although **ADABAS_FILE_11_AIC** still appears in the window, it will not be deployed since **Active** is not selected.

Validate the un-deploy

Click the **Deployed Replications** tab. Verify that the only entry is for **DBID1.dbo.ADABAS_FILE_11**.



This completes removing a replication from deployment. You can do inserts, updates and deletes against ADABAS_FILE_11 in database 1 and the changes will replicate to your SQL Server database.

Manipulating Replication Data

This tutorial manipulates replication data by:

- Filtering data
- Combining multiple fields
- Excluding data fields
- Applying mathematical formulas
- Extracting a portion of a field
- Renaming fields

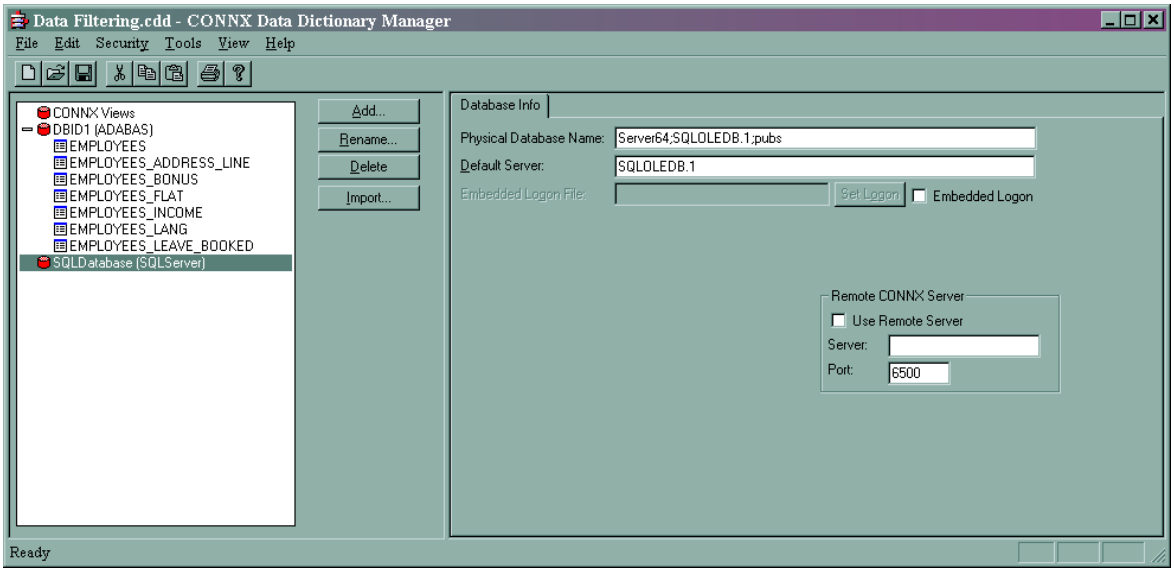
Prerequisites:

- Install Adabas version 6.1.4 or above.
- Install ConnecX SQL Engine 1.1.2 or above
- Install the Open Systems Event Replicator
- Create an Adabas database with a DBID of 1 using the Adabas sample tables. For this example, we will use the Adabas sample table **EMPLOYEES-NAT (File11)**
- Start the Adabas database.
- An available SQL Server database. For this tutorial, we recommend that you create a database named SQLDatabase. Otherwise, you can use any existing SQL Server database.
- Create a CDD containing the source and target databases. For this example, we will use the CDD **Adabas to Relational Replication** (our last opened CDD) that was created in the Performing a Simple Replication - Adabas to Relational Database tutorial.

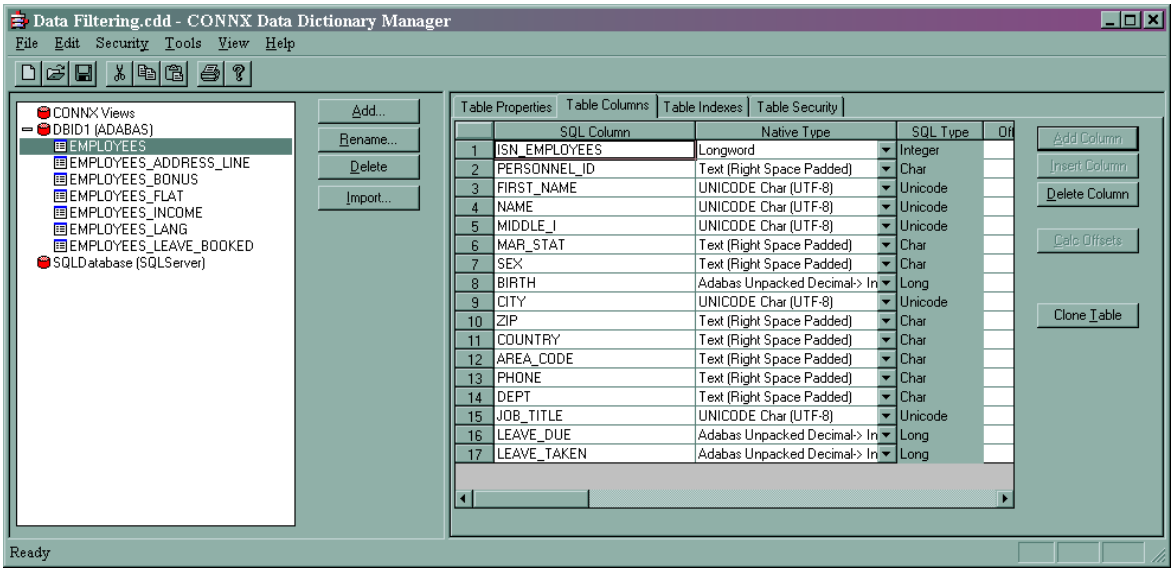
Filtering Replication Data

In this example, we will copy a table view, filtering out a portion of the original table during the copying process.

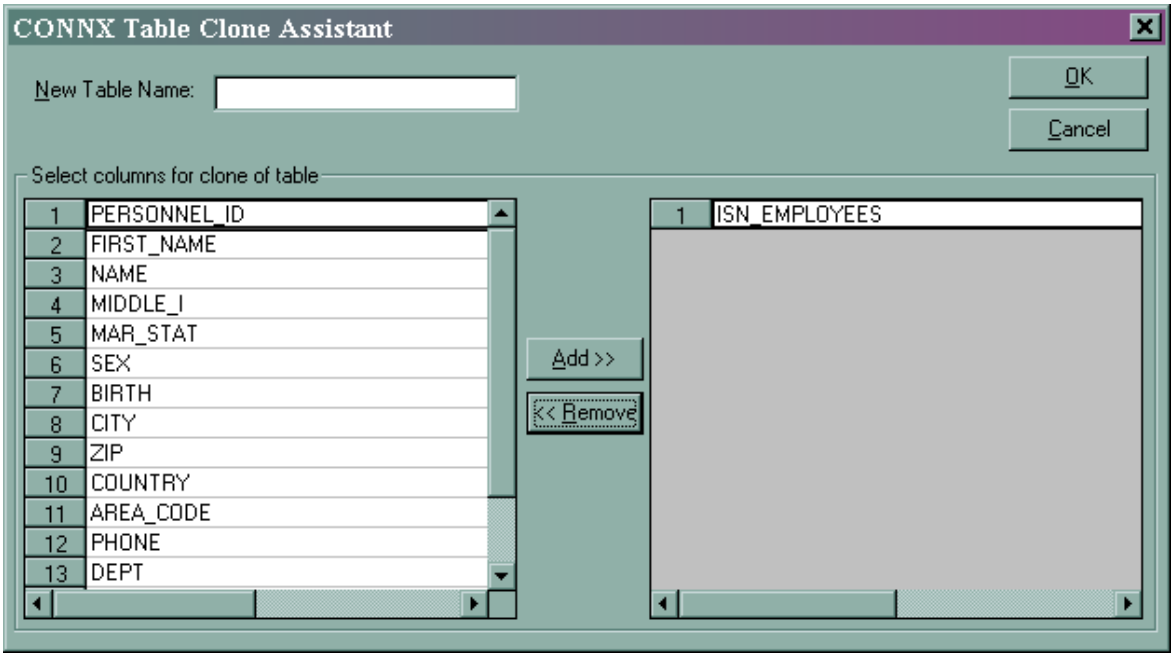
1. On the **Start** menu, click **Programs**, click **CONNX Driver** and then click **CONNX Data Dictionary**. The **CONNX Data Dictionary Manager** window appears.



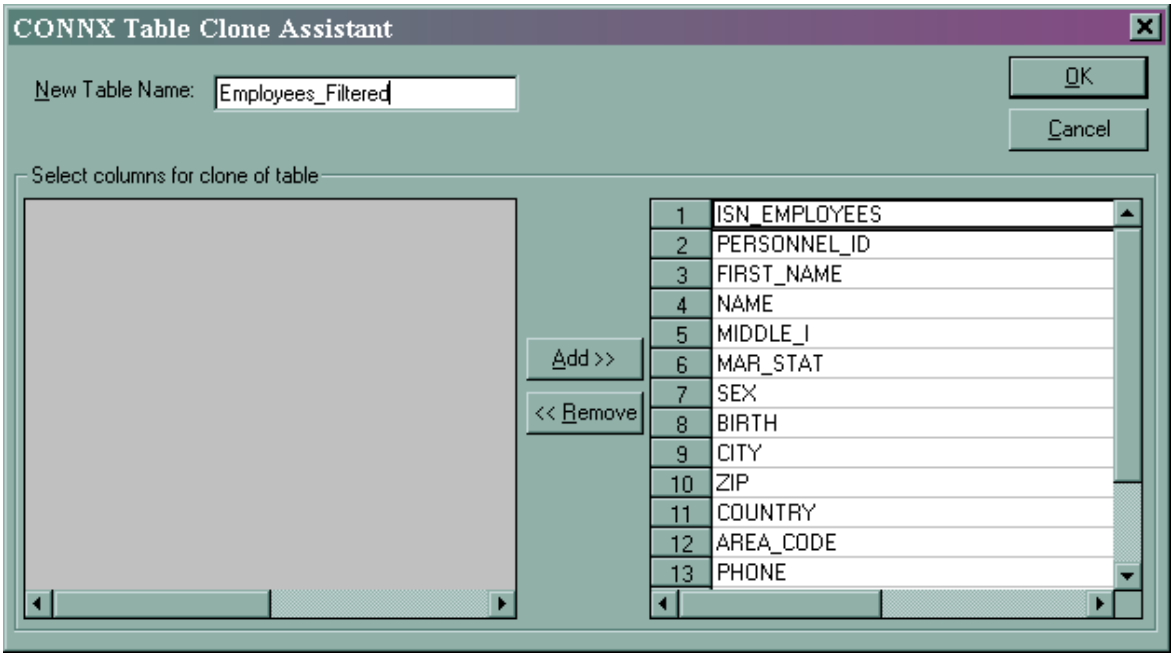
2. Select the **EMPLOYEES** table and click the **Table Columns** tab.



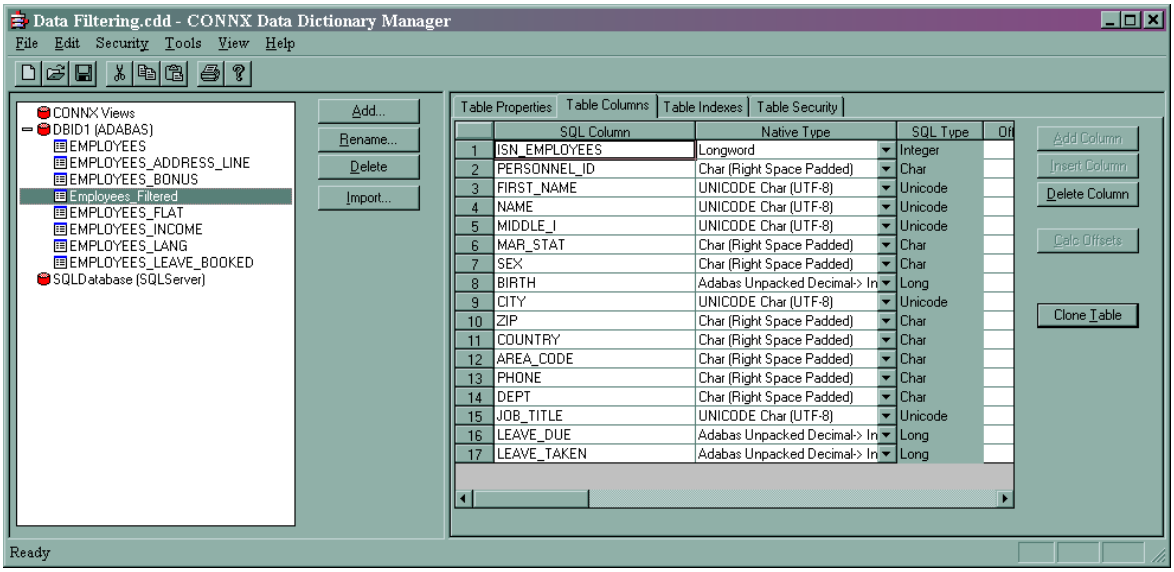
3. Click **Clone Table**. The **Clone Table Assistant** appears.



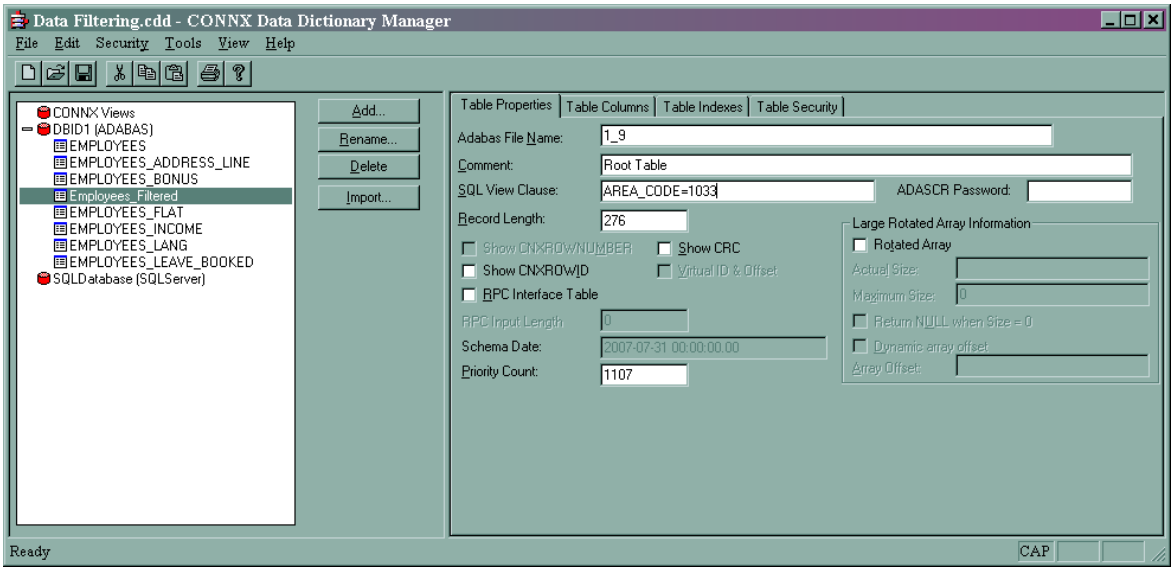
4. In **New Table Name**, enter Employees_Filtered. Select all the columns and click **Add>>**. All the columns move from the left side to the right.



5. Click **OK**. The **CONNX Data Dictionary Manager** window appears. Employees_Filtered has been added to the CDD and the **Table Columns** tab contains the SQL Table column information.



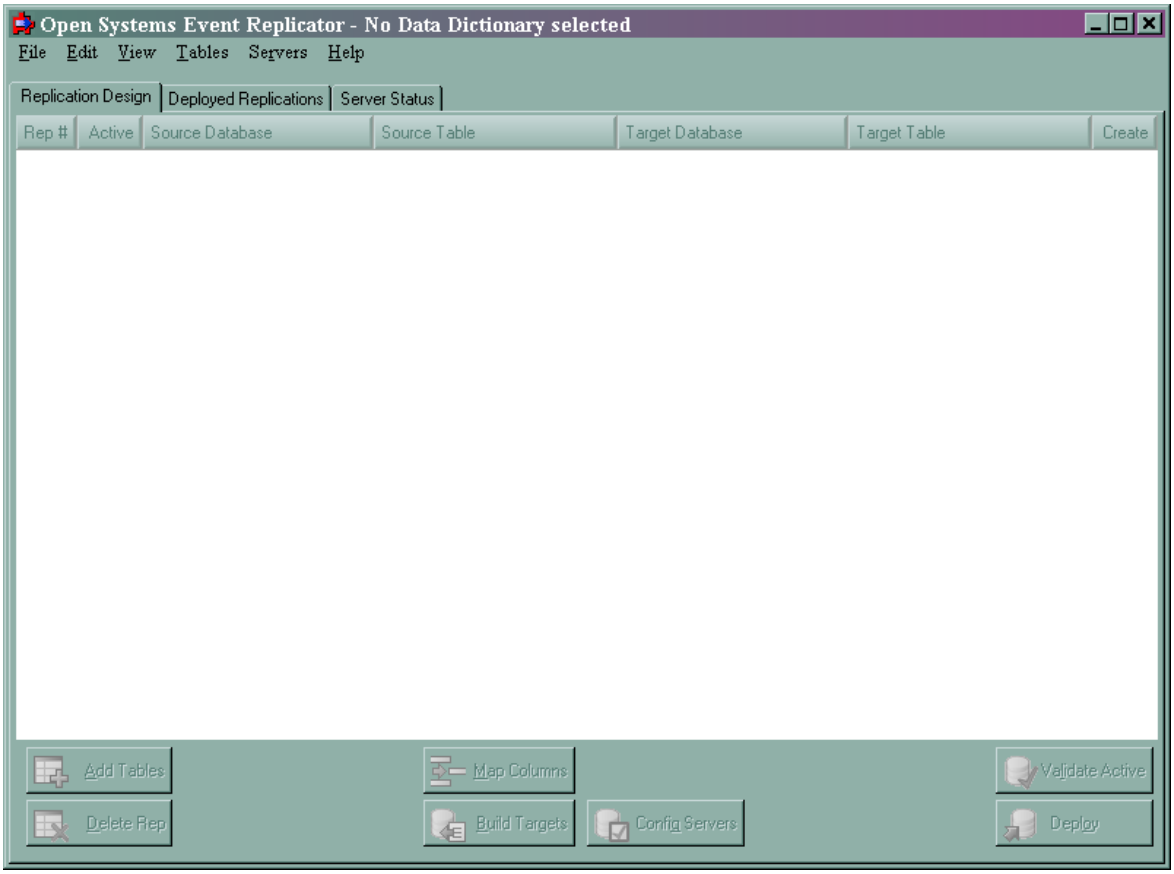
6. Click the **Table Properties** tab. We use the **SQL View Clause** field to filter data for replication. In this example, we only want the employees who have a phone number with area code 1033. Enter **AREA_CODE=1033** in **SQL View Clause**.



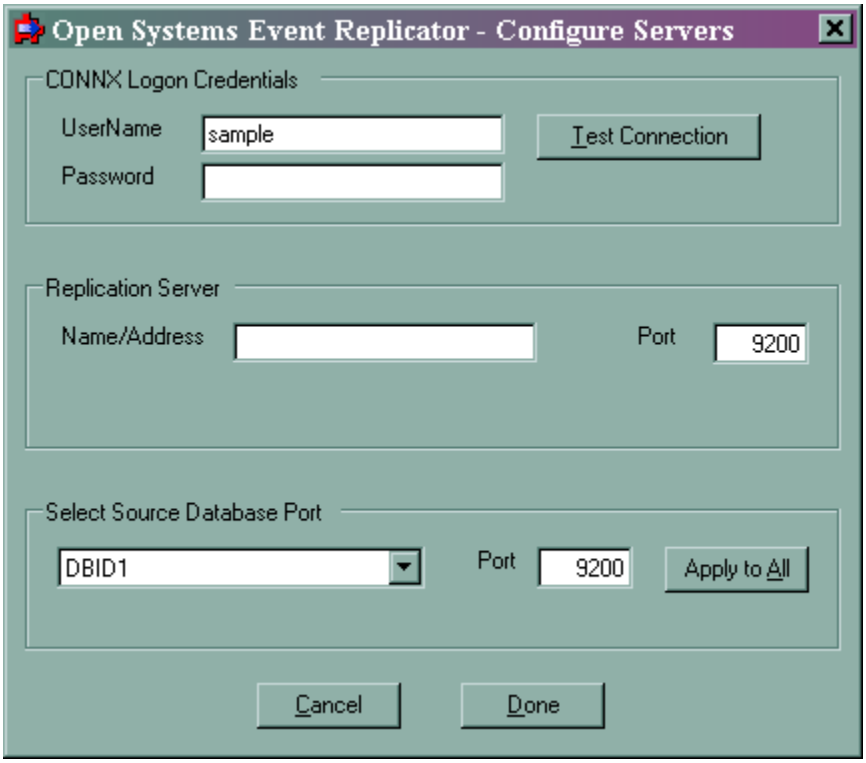
7. In the **File** menu, select **Save**.

   Employees_Filtered is now a clone of the view of the Employees table. Employees and Employees_Filtered are the same physical table. The Employees view shows the entire table's data; the Employees_Filtered view shows only the records for employees with AREA_CODE=1033. The filter will be applied when queries are applied to the Employees_Filtered table.

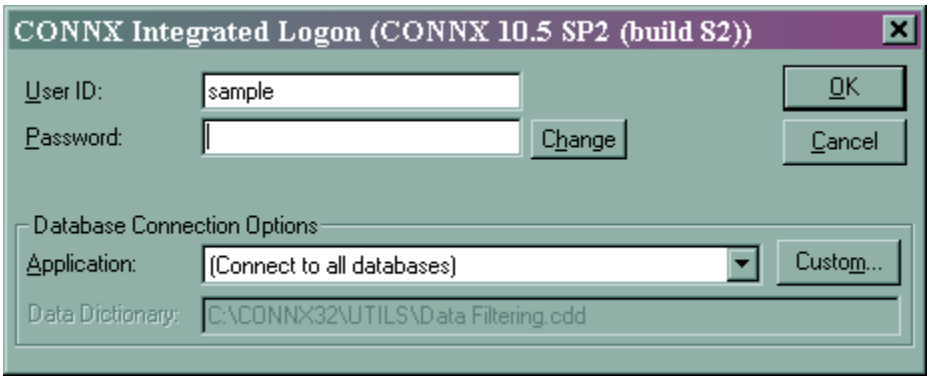
8. Close the **CONNX Data Dictionary Manager** window and on the **Start** menu, click **Programs**, click **CONNX Solutions**, click **Open Systems Event Replicator** and then click **Replication Administrator**. If the **Configure Servers** window appears, click **Cancel**. The **No Data Dictionary selected** window appears.



9. In the **File** menu, select **OpenCDD**. Select **Data Filtering.cdd** and click **Open**. The **Configure Servers** window appears.



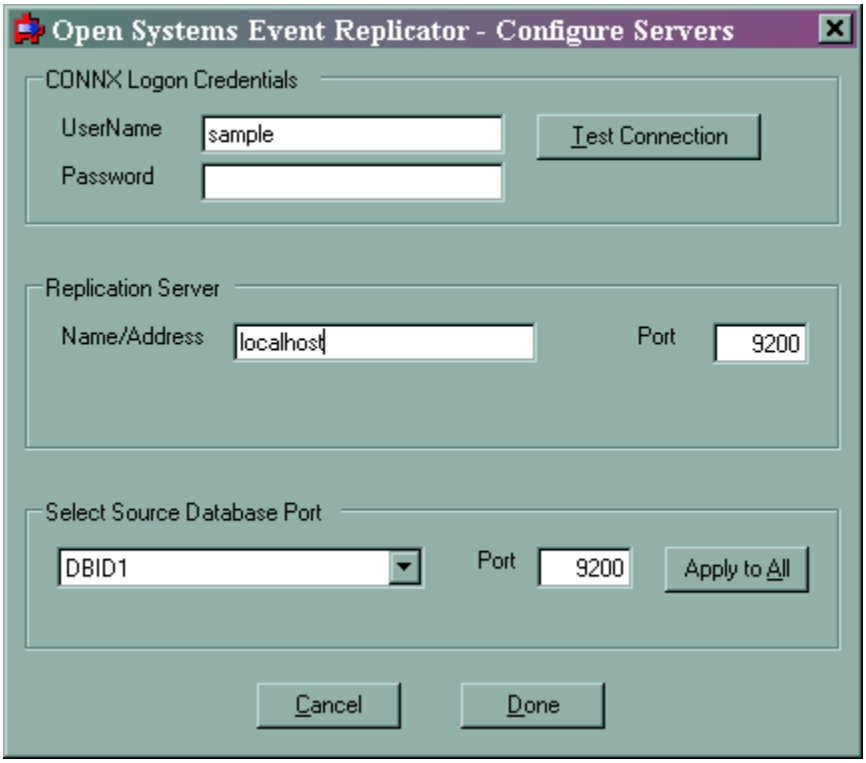
10. In **CONNX Logon Credentials** enter **Sample** in **UserName** and leave **Password** blank. Click **Test Connection**. The **CONNX Integrated Logon** window appears.



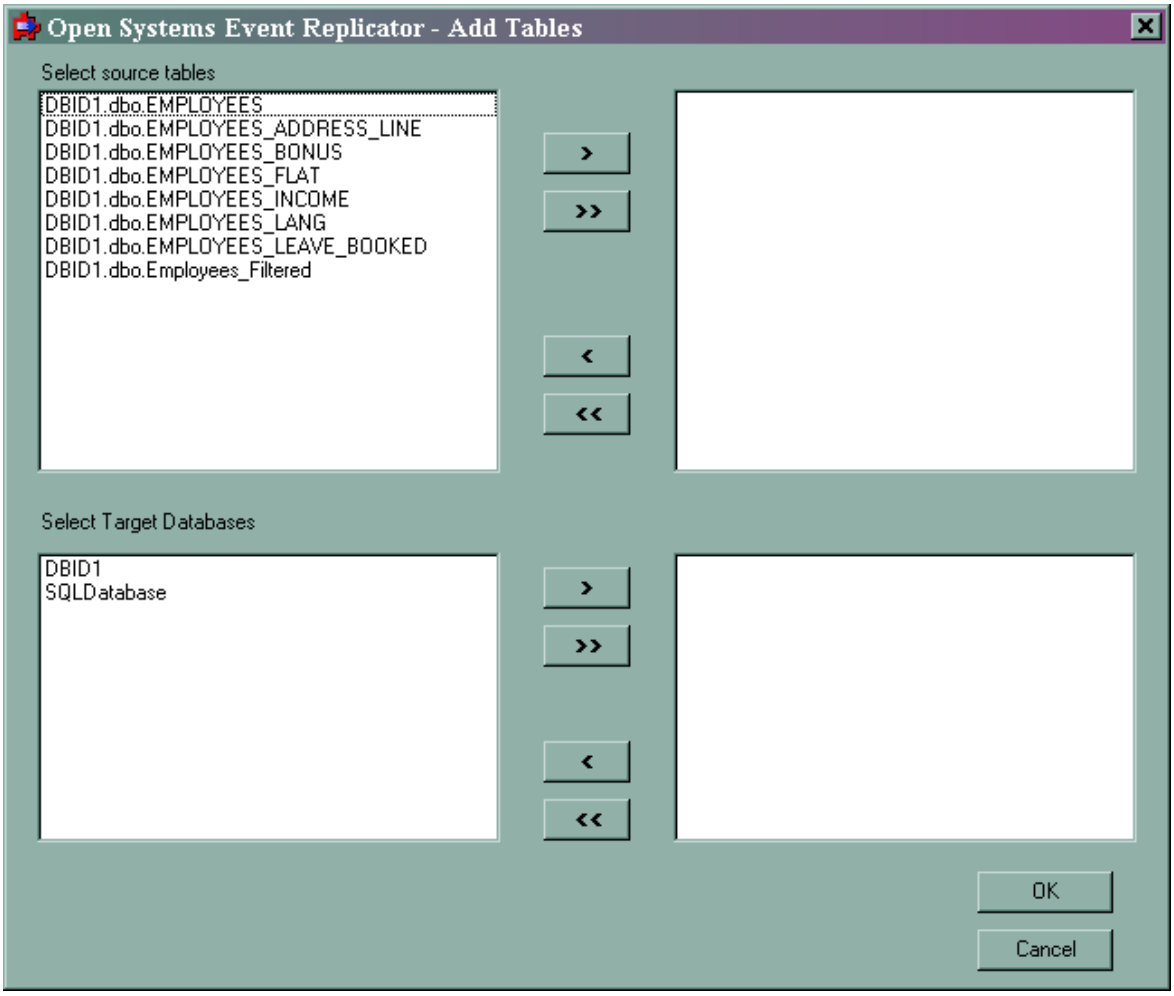
11. Leave **Password** blank and click **OK**.

    You will be prompted to create a user in the CDD. Press **OK.** The User ID **sample** will be created with a blank password.

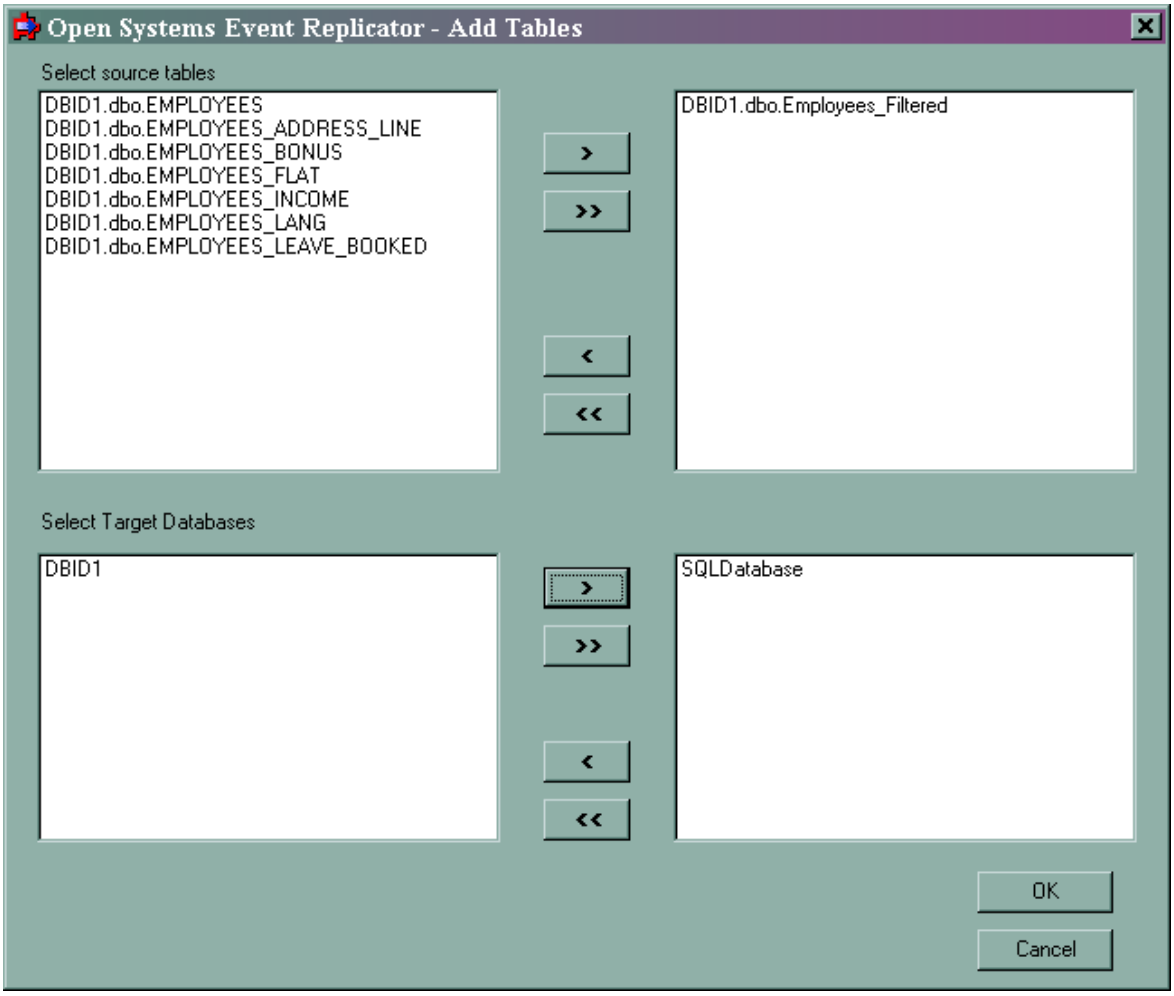
    When the connection succeeds, enter **localhost** in **Name/Address** and **9200** in **Port**.



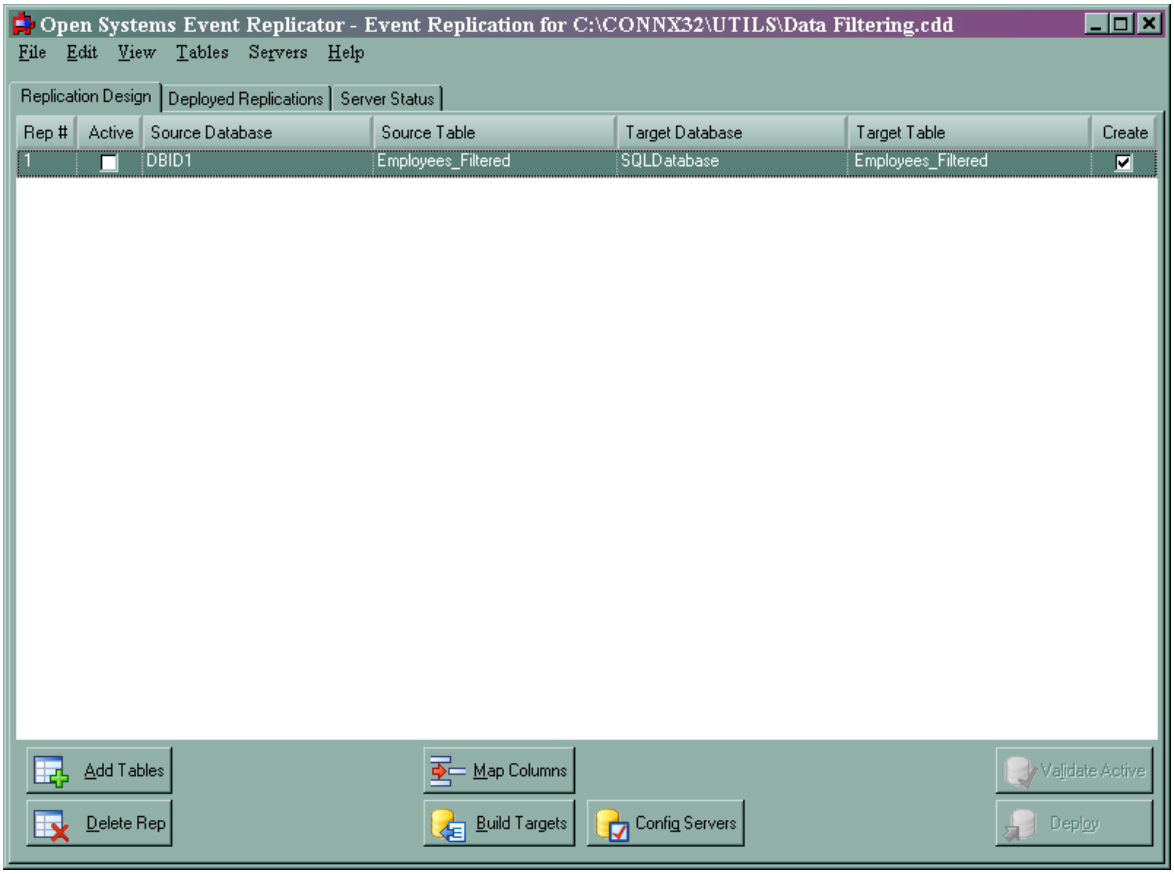
12. Click **Done**. The **Add Tables** window appears.



13. From **Select source tables**, select **DBID1.dbo.Employees_Filtered**. From **Select Target Databases**, select **SQLDatabase**.



14. Click **OK**. The **Open Systems Event Replication** window appears.

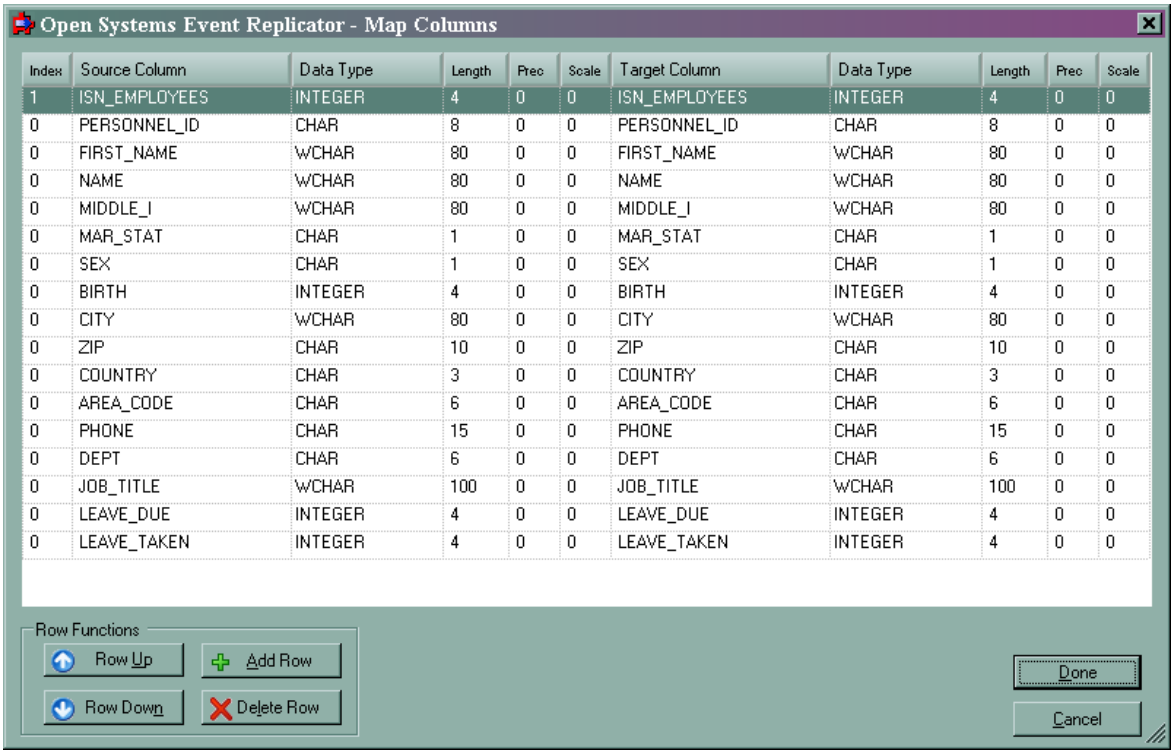


This completes filtering replication data.

Concatenating Multiple Data Columns

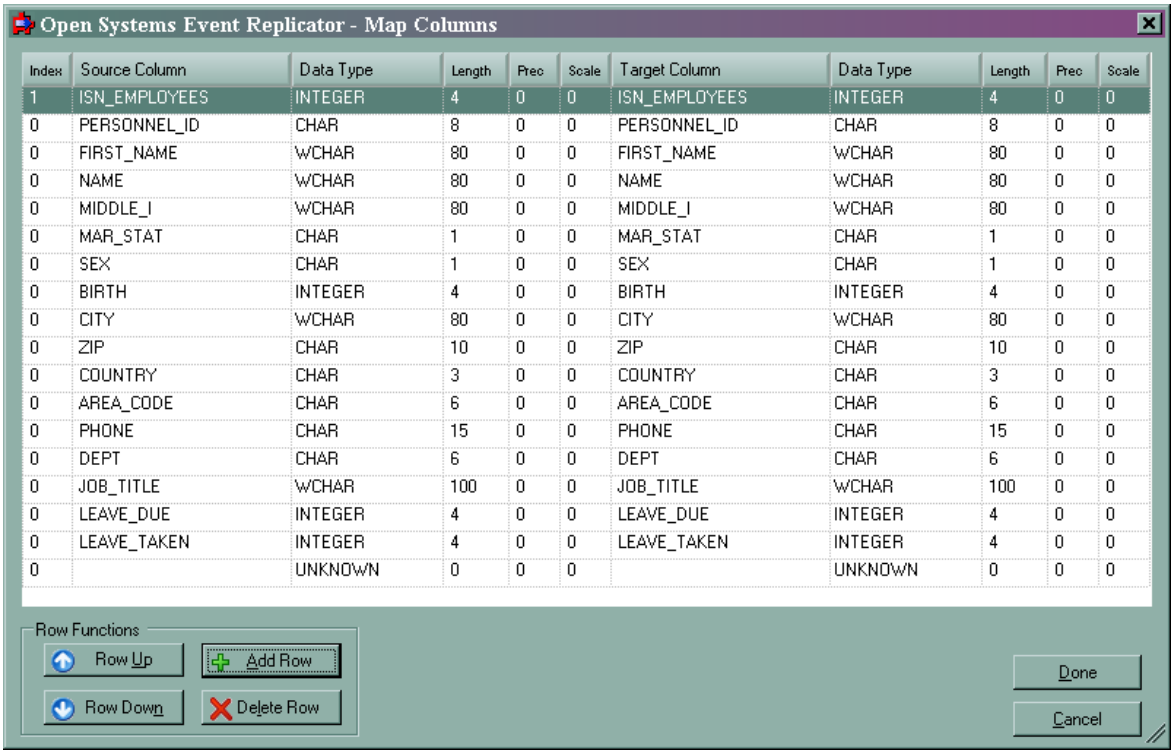
In this example, we will combine the first name, middle initial and last name fields in the copy of the Employees table into one field (FULL_NAME) so it's easier to read the contents.

1. Using the Employees_Filtered table created in **Filtering Replication Data**, click **Map Columns**. The **Map Columns** window appears.



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN_EMPLOYEES	INTEGER	4	0	0	ISN_EMPLOYEES	INTEGER	4	0	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	WCHAR	80	0	0	FIRST_NAME	WCHAR	80	0	0
0	NAME	WCHAR	80	0	0	NAME	WCHAR	80	0	0
0	MIDDLE_I	WCHAR	80	0	0	MIDDLE_I	WCHAR	80	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	BIRTH	INTEGER	4	0	0	BIRTH	INTEGER	4	0	0
0	CITY	WCHAR	80	0	0	CITY	WCHAR	80	0	0
0	ZIP	CHAR	10	0	0	ZIP	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	WCHAR	100	0	0	JOB_TITLE	WCHAR	100	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	0	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	0	0

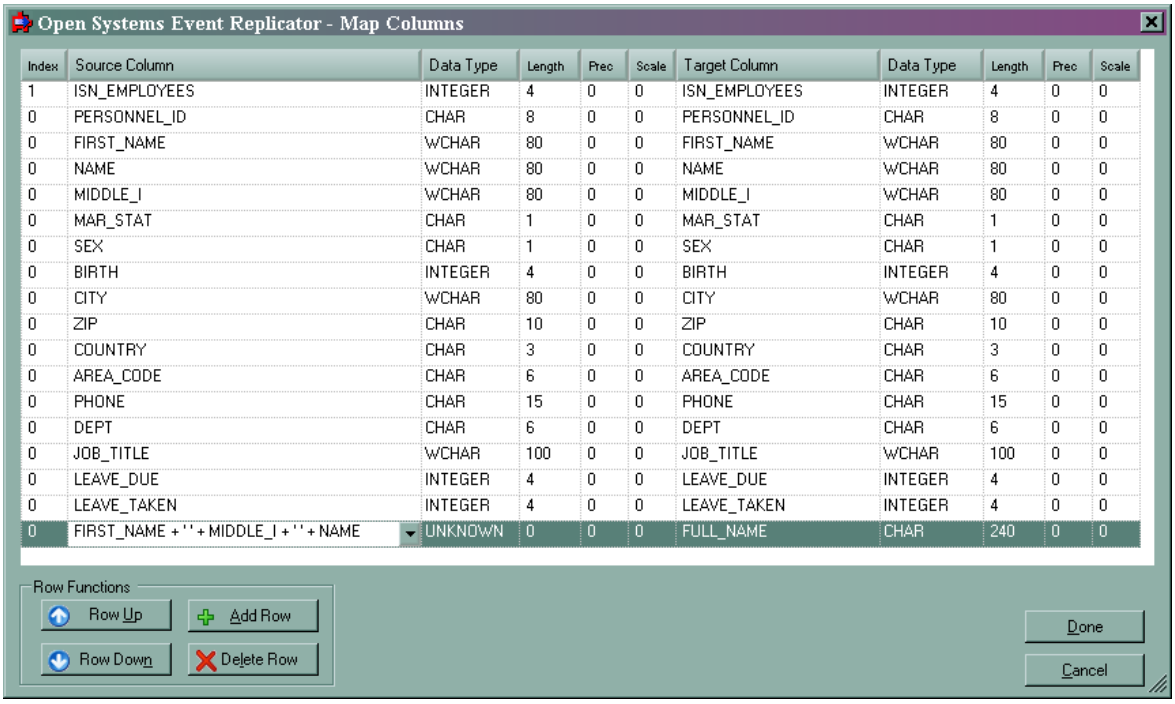
2. Click **Add Row**. A new row appears at the end of the table description. **Note:** This example illustrates column mapping on a table that does not already exist on the target database. The Add Row, Delete Row, Row Up and Row Down buttons are only available when mapping to a new target table. If 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.**



- In **Source Column** enter the SQL expression **FIRST_NAME + ' ' + MIDDLE_I + ' ' + NAME**

  This concatenates the first, last and middle initial into one field with a space between each name.
- In **Target Column** enter **FULL_NAME**.
- In the target column **Data Type** select **CHAR**.
- In the target column **Length** enter **240**.

  Since each of the three columns that are being concatenated into **FULL_NAME** has an 80 characters length, **FULL_NAME**'s length is 240.

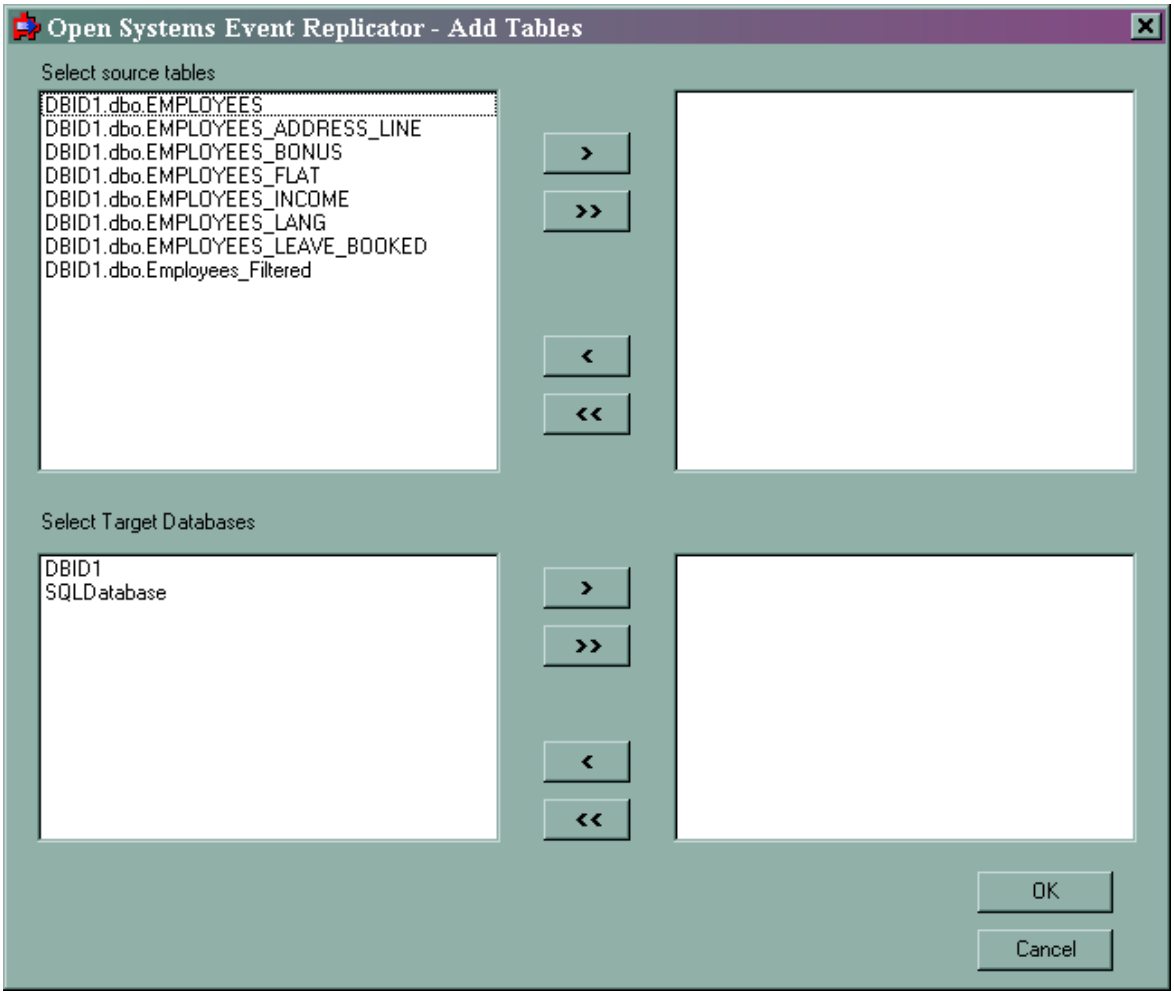


3. Click **Done**. This completes concatenating multiple data columns.

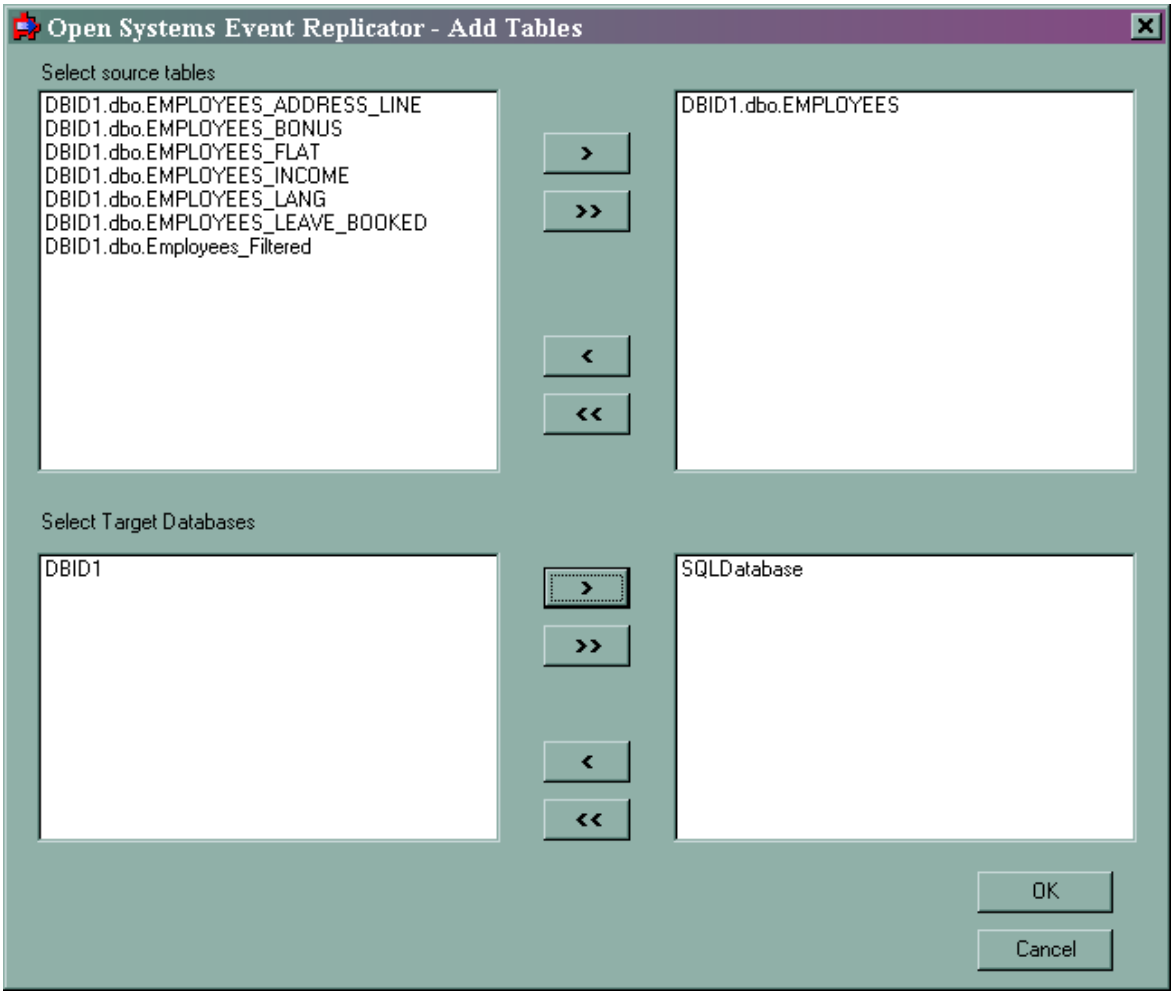
Excluding Data Columns

In this example, we will exclude all the name data from a replication table so the information can be anonymously analyzed by a demographic program.

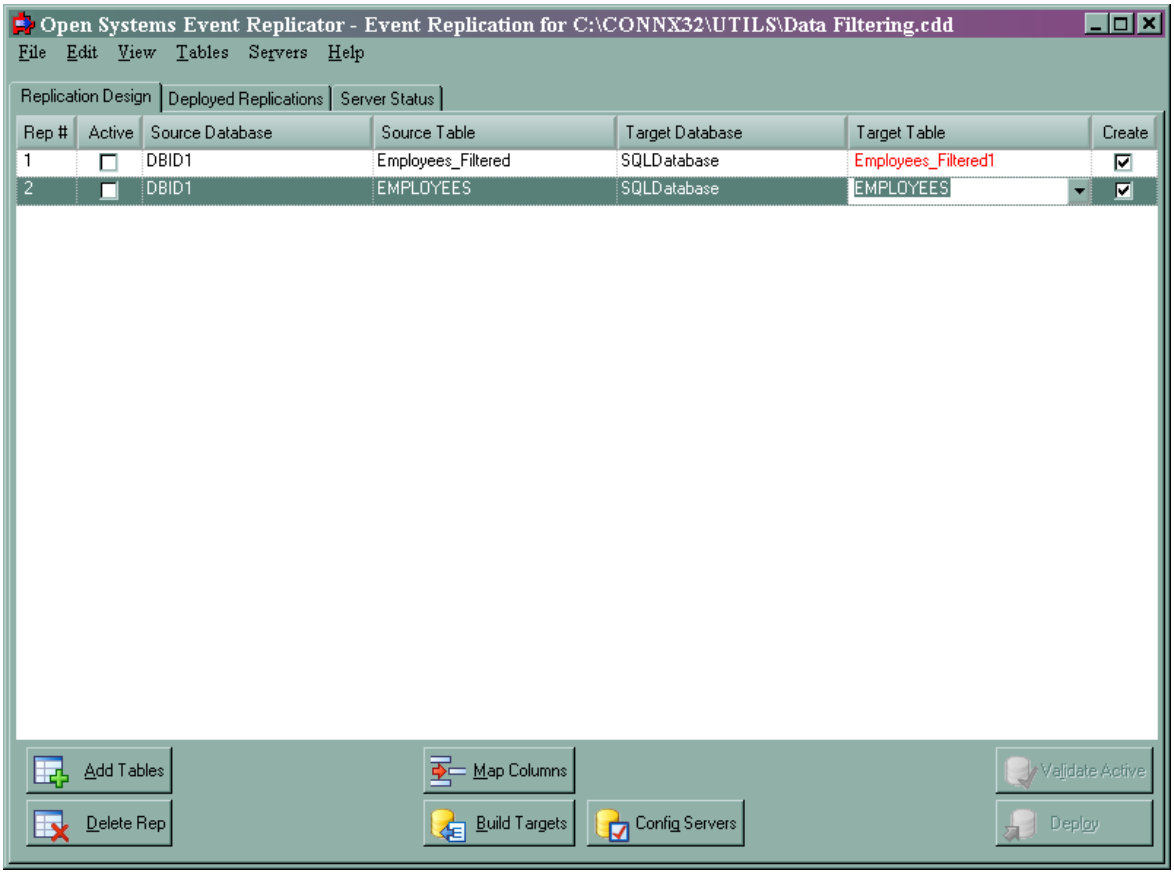
1. From the **CONNX Data Dictionary Manager** window, click the **Replication Design** tab. Click **Add Tables**. The **Add Tables** window appears.



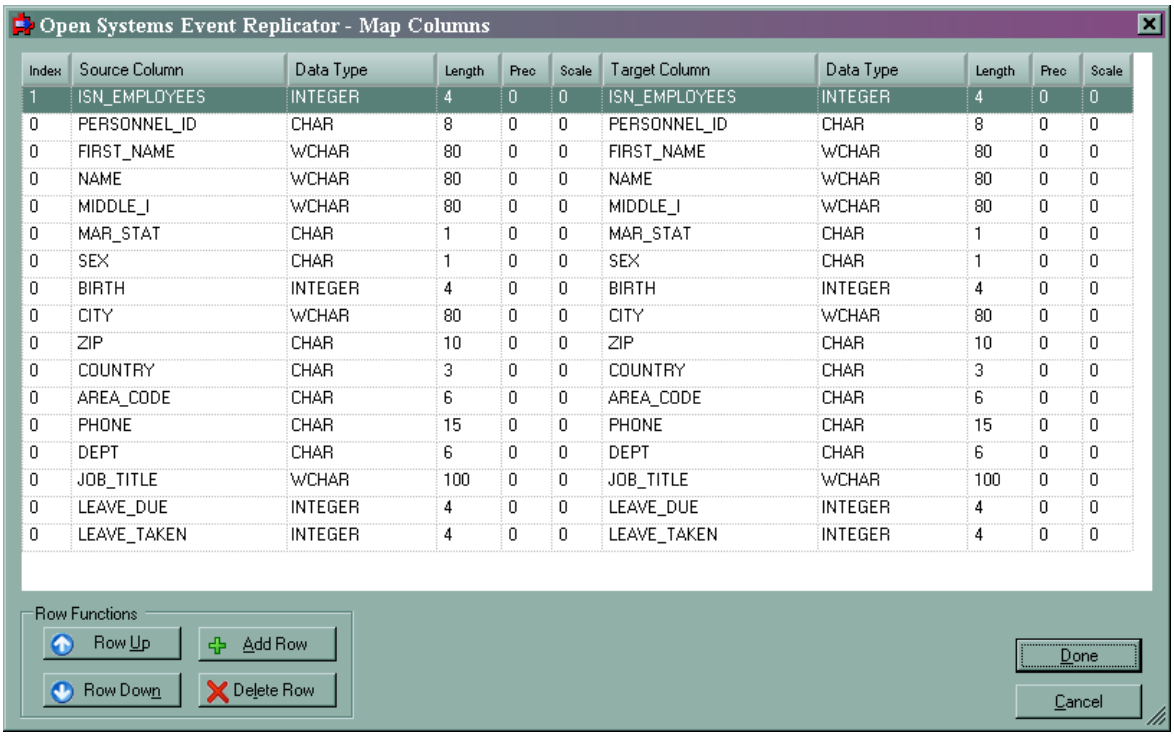
2. From **Select source tables**, select **DBID1.dbo.EMPLOYEES**. From **Select Target Databases**, select **SQLDatabase**.



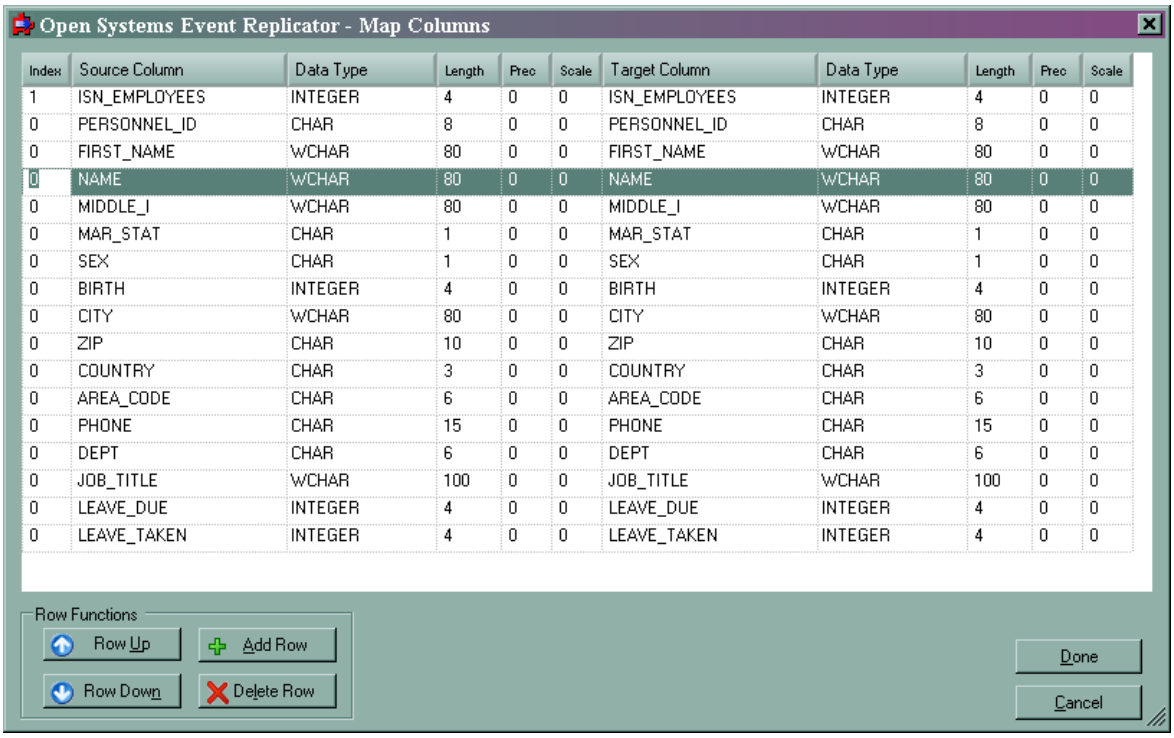
3. Click **OK**. The **Open Systems Event Replication** window appears. Select the Employees table.



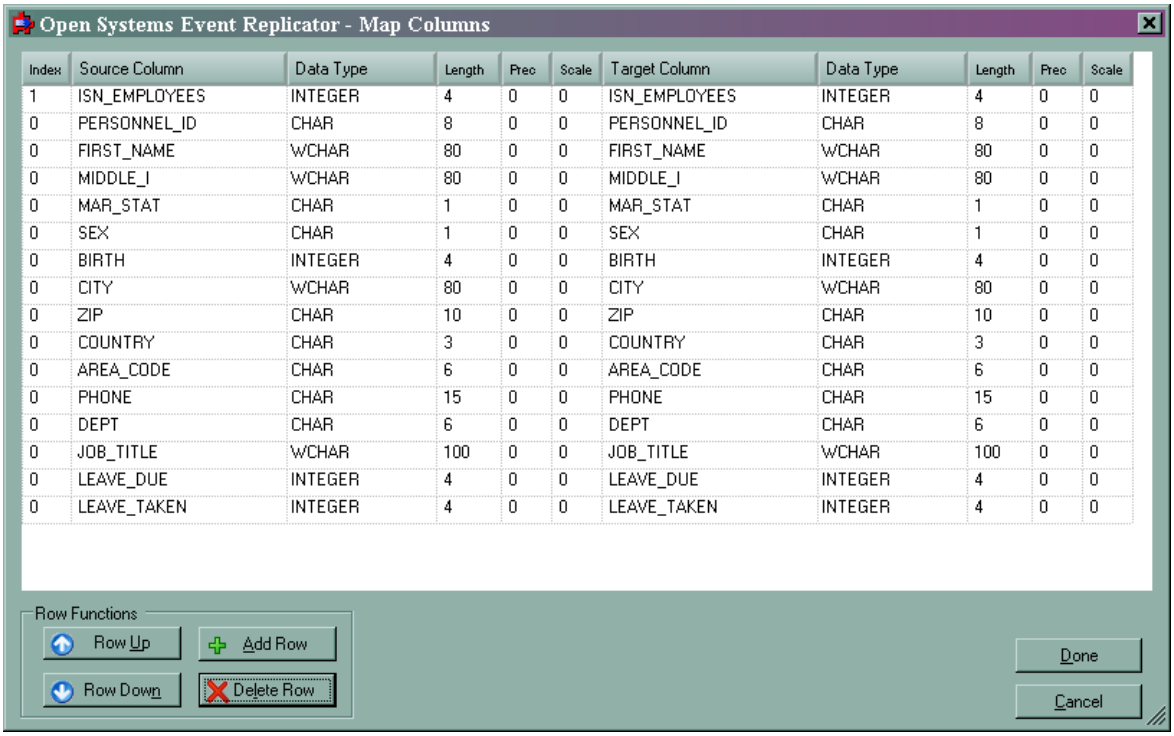
4. Click **Map Columns**. The **Map Columns** window appears.



5. Select the **Name** column

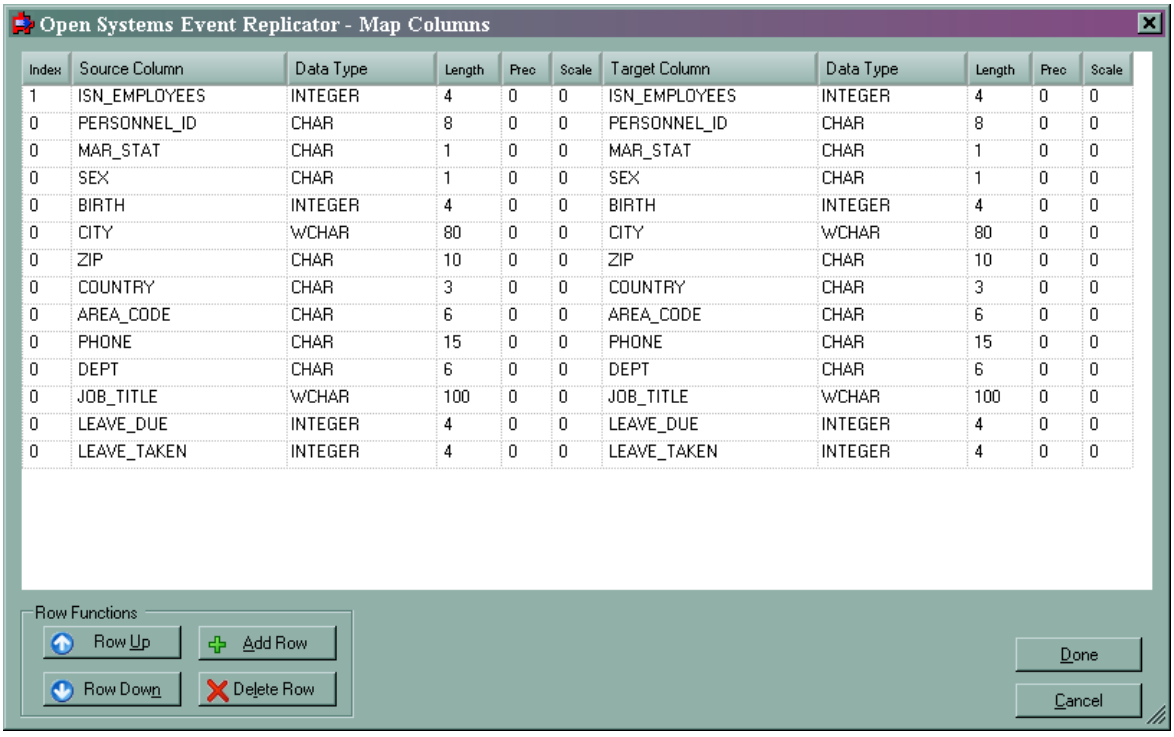


6. Click **Delete Row**. A message appears asking if you want to delete a single row. Click **Yes**. The Map Column window now looks like this:



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN_EMPLOYEES	INTEGER	4	0	0	ISN_EMPLOYEES	INTEGER	4	0	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	WCHAR	80	0	0	FIRST_NAME	WCHAR	80	0	0
0	MIDDLE_I	WCHAR	80	0	0	MIDDLE_I	WCHAR	80	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	BIRTH	INTEGER	4	0	0	BIRTH	INTEGER	4	0	0
0	CITY	WCHAR	80	0	0	CITY	WCHAR	80	0	0
0	ZIP	CHAR	10	0	0	ZIP	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	WCHAR	100	0	0	JOB_TITLE	WCHAR	100	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	0	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	0	0

7. Delete the **MIDDLE_I** and **FIRST_NAME** rows the same way you deleted the **NAME** row. The **Map Column** window will look like this:



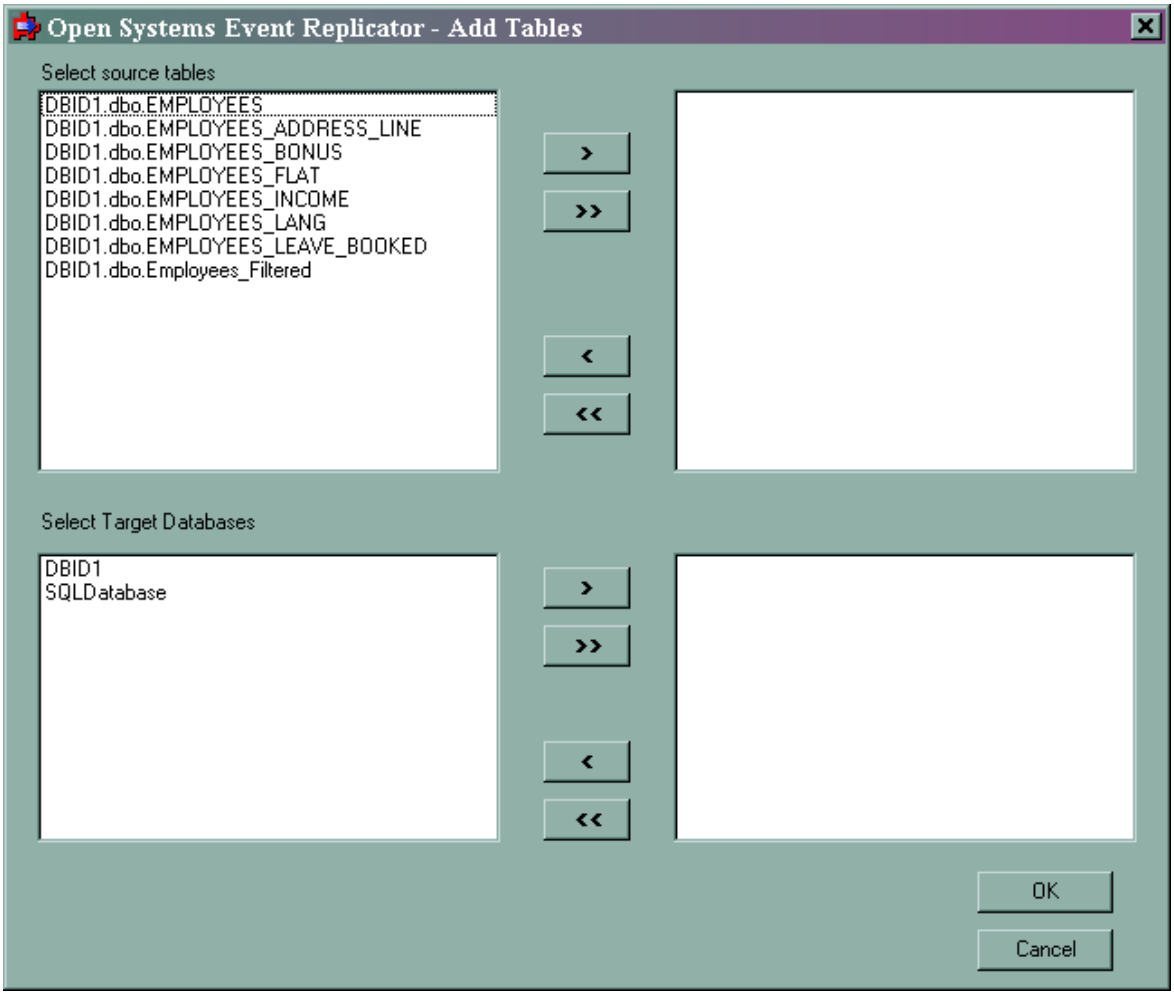
Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN_EMPLOYEES	INTEGER	4	0	0	ISN_EMPLOYEES	INTEGER	4	0	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	BIRTH	INTEGER	4	0	0	BIRTH	INTEGER	4	0	0
0	CITY	WCHAR	80	0	0	CITY	WCHAR	80	0	0
0	ZIP	CHAR	10	0	0	ZIP	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	WCHAR	100	0	0	JOB_TITLE	WCHAR	100	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	0	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	0	0

8. Click **Done**. This completes excluding data columns.

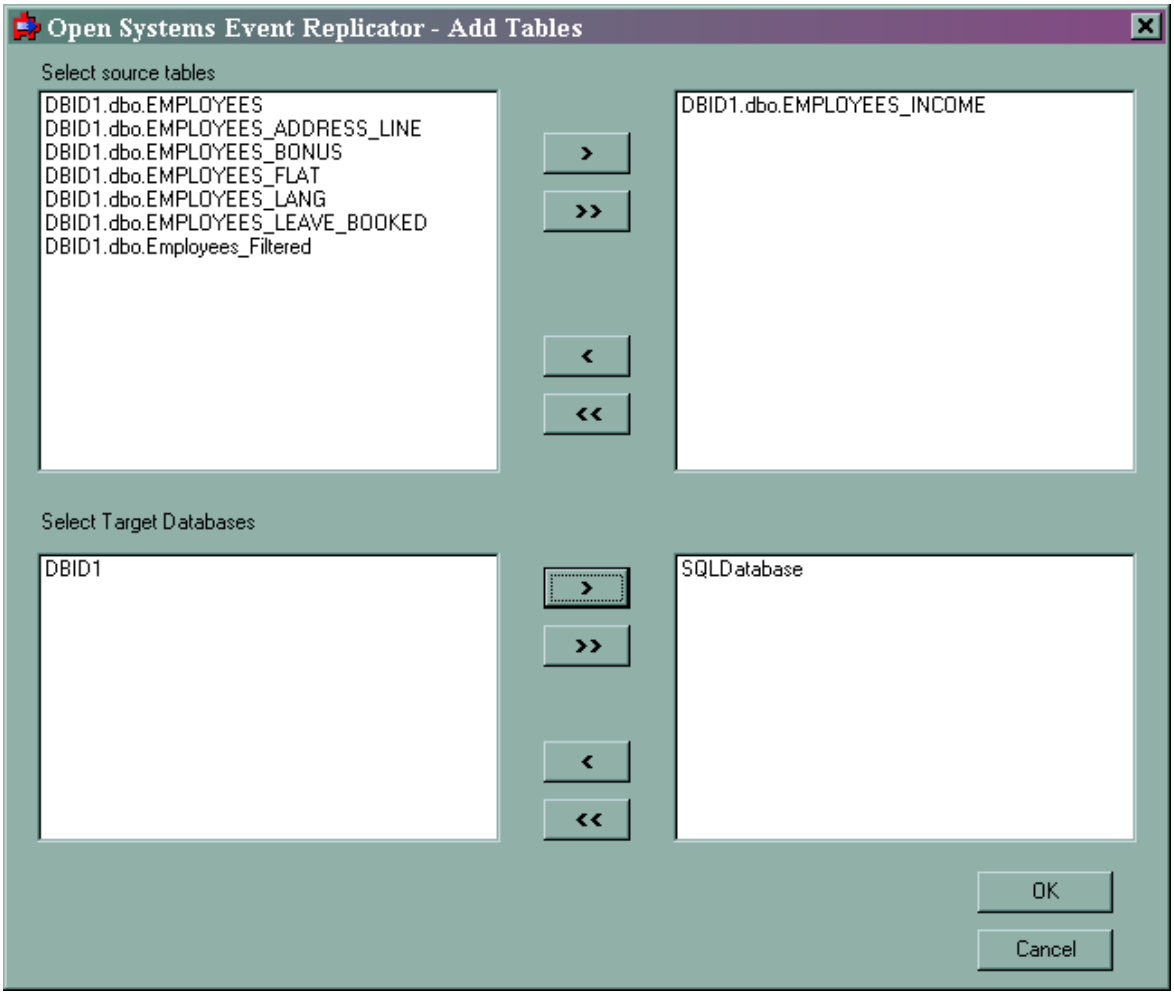
Performing Math Operations on Replication Data

In this example, we will calculate the maximum amount of money an employee can contribute to a special savings plan.

1. From the **CONNX Data Dictionary Manager** window, click the **Replication Design** tab. Click **Add Tables**. The **Add Tables** window appears.

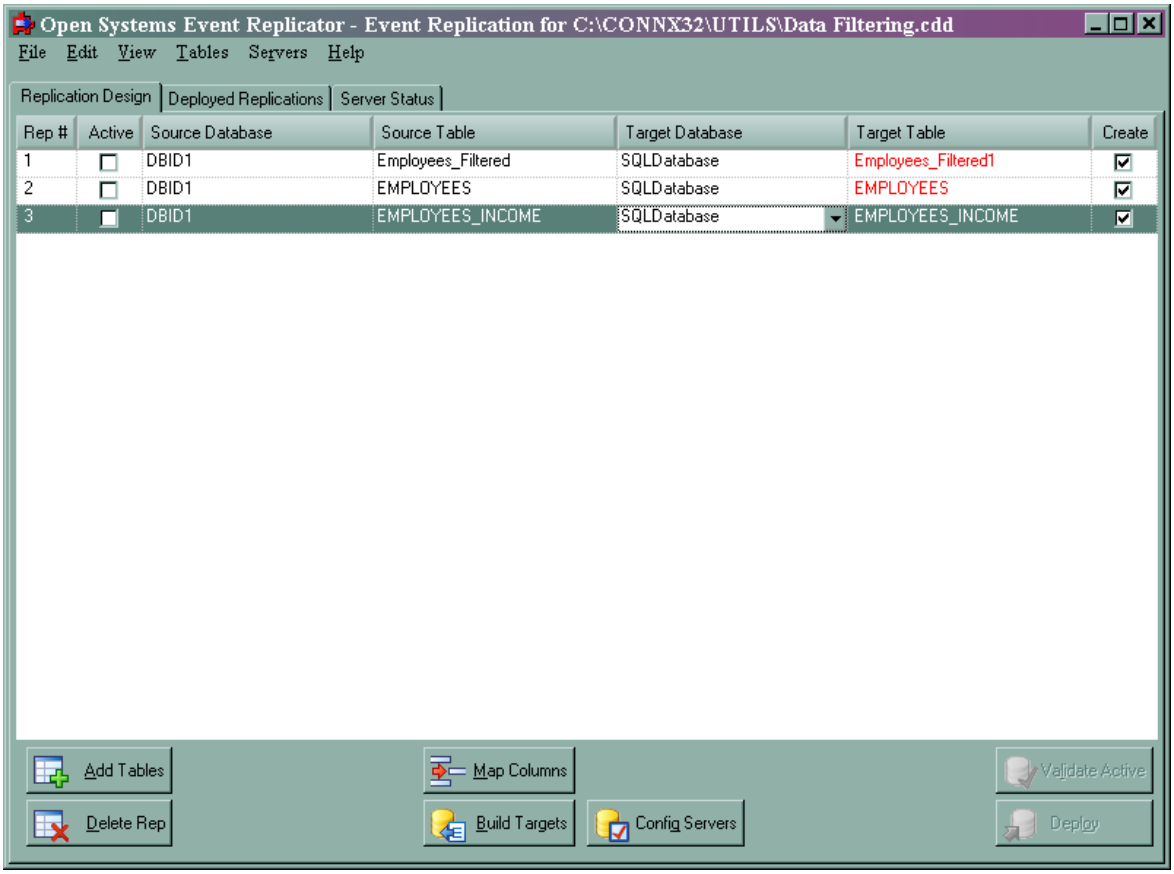


2. From **Select source tables**, select **DBID1.dbo.EMPLOYEES_INCOME**. From **Select Target Databases**, select **SQLDatabase**.

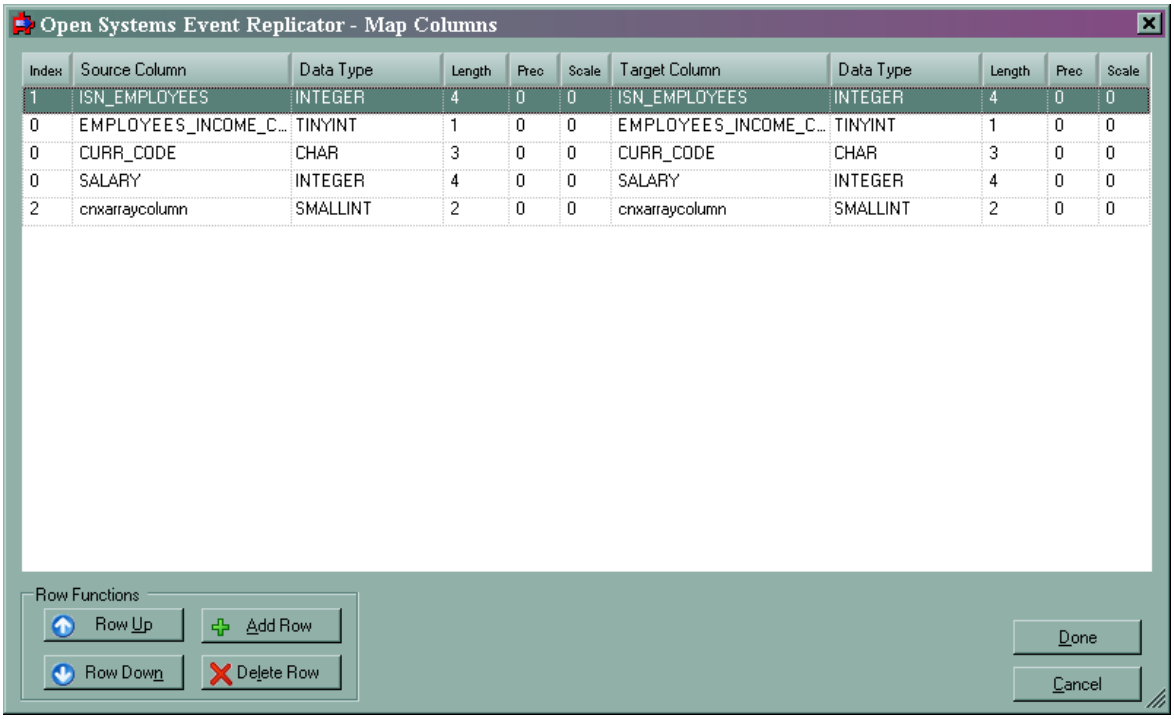


.

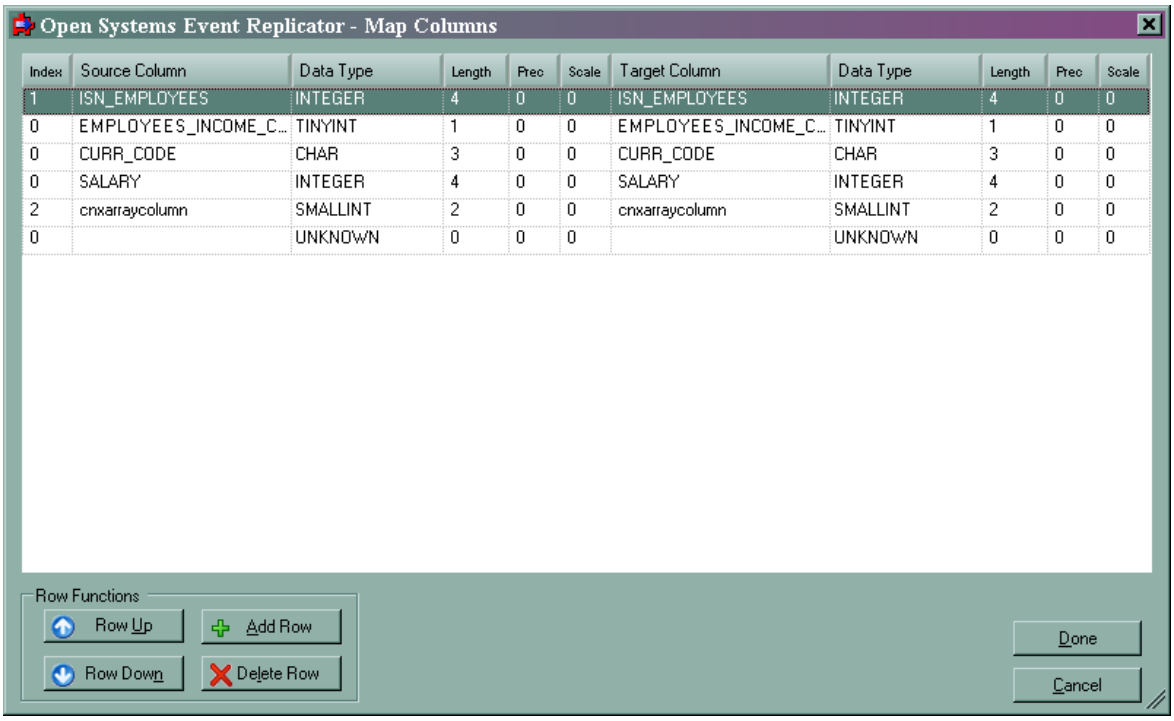
3. Click **OK**. The **Open Systems Event Replication** window appears. Select the **EMPLOYEES_INCOME** table.



4. Click **Map Columns**. The **Map Columns** window appears.

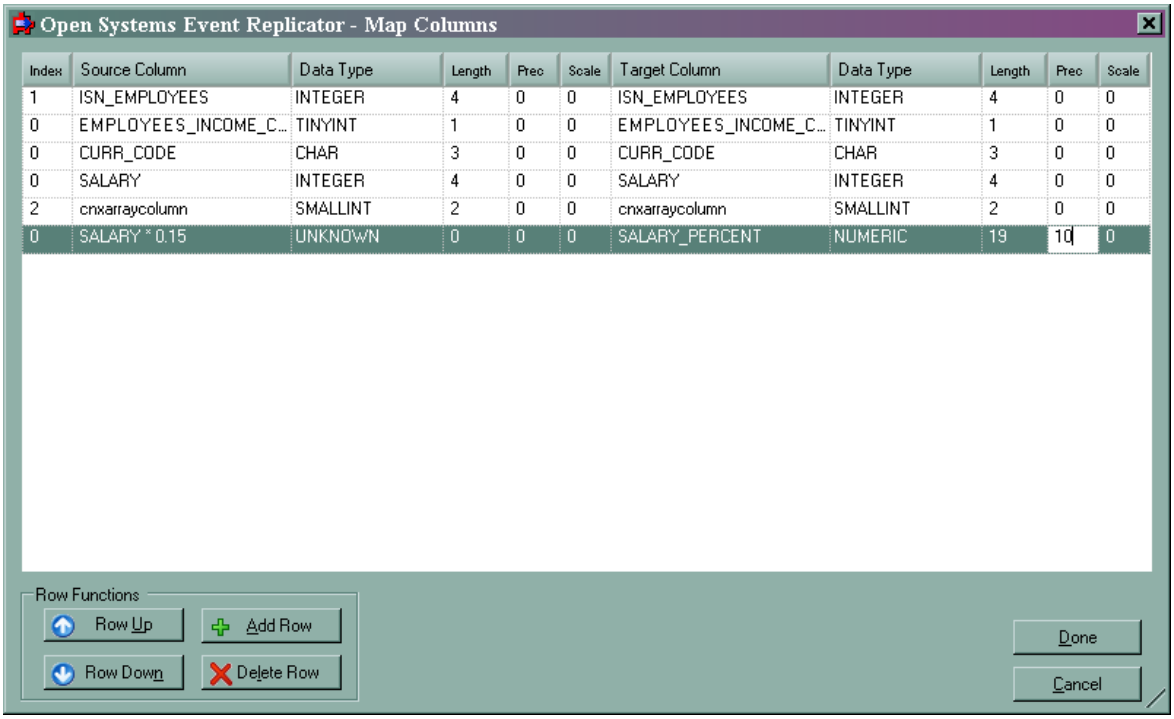


5. Click **Add Row** to create a new entry in the column map.



- In **Source Column** enter "**SALARY * 0.15**"; this will calculate15% of the value in **SALARY** and place it into a new column in the Target table
- In **Target Column** enter **SALARY_PERCENT**.
- In the target column **Data Type** select **NUMERIC**.
- In the target column **Length** enter **19**.
- In the target column **Prec** enter **10**.

.

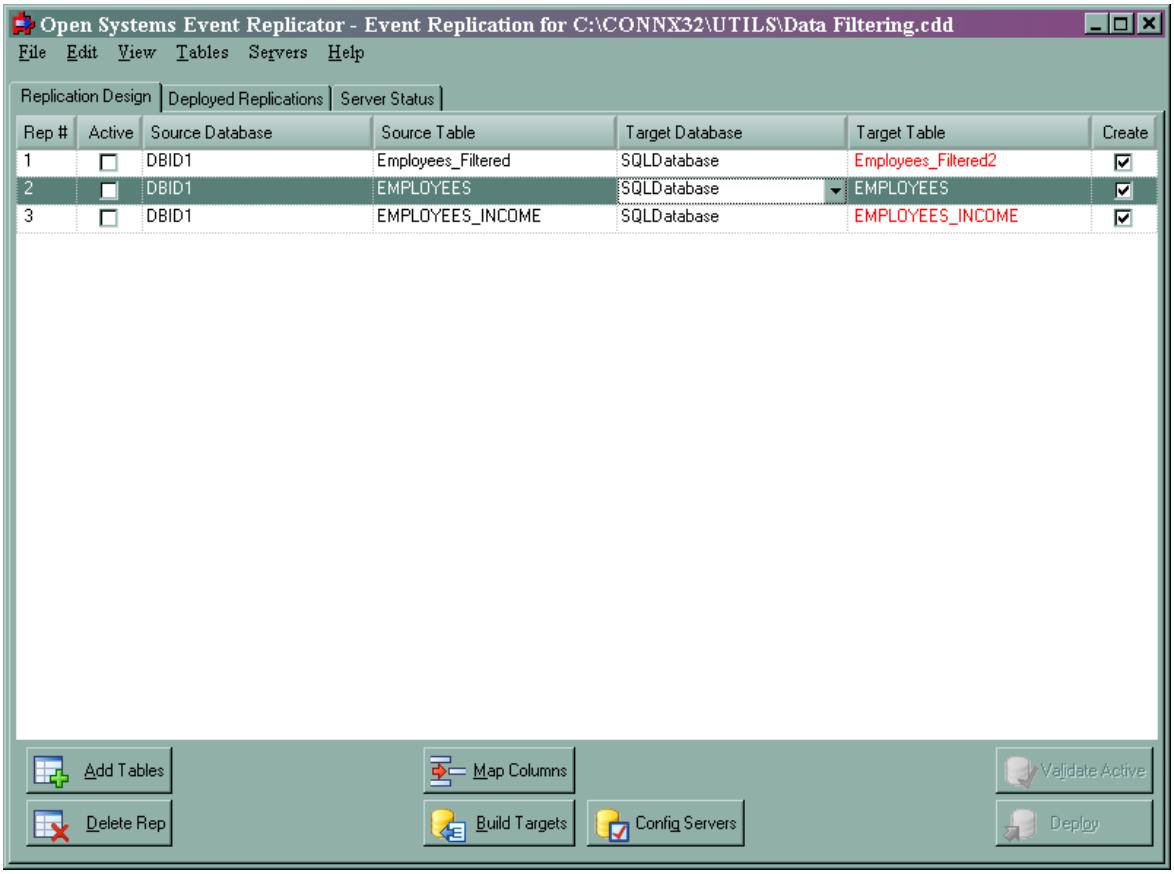


6. Click **Done**. This completes performing math operations on replication data.

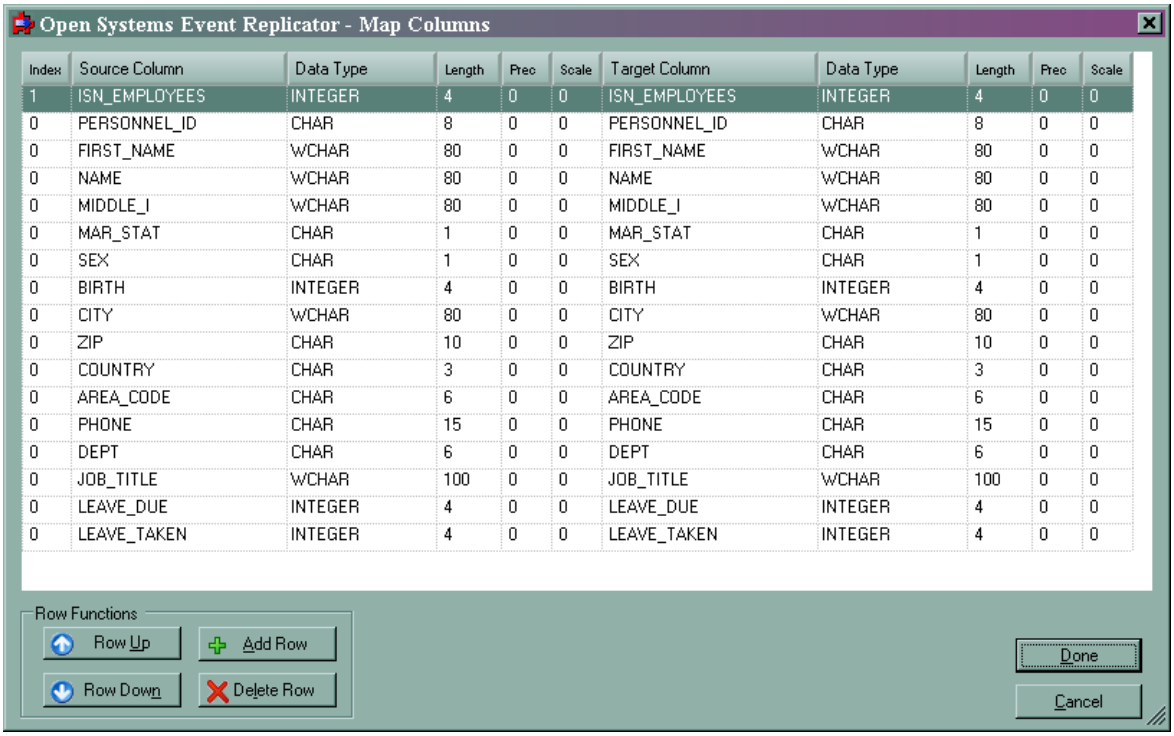
Using SQL Expressions to Split a Column

In this example, we will use **CITY** to create a two-letter City name abbreviation.

1. From the **Open Systems Event Replication** window select the **EMPLOYEES** table.

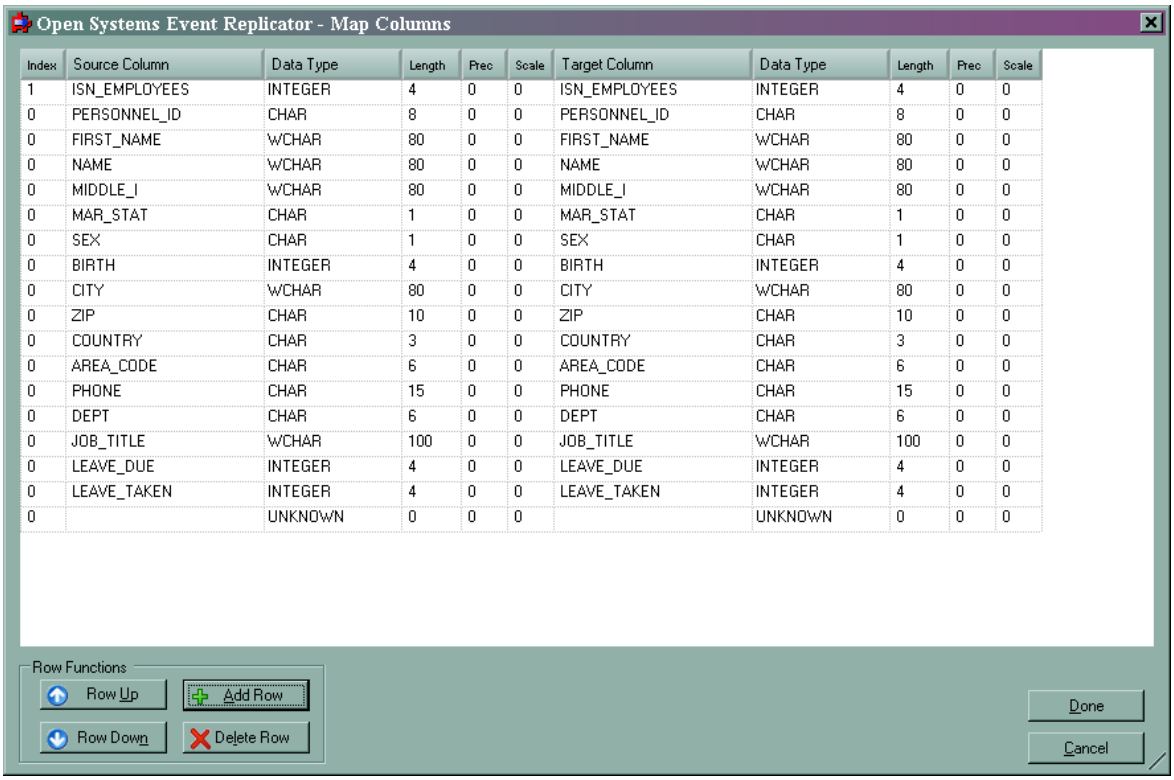


2. Click **Map Columns**. The **Map Columns** window appears.



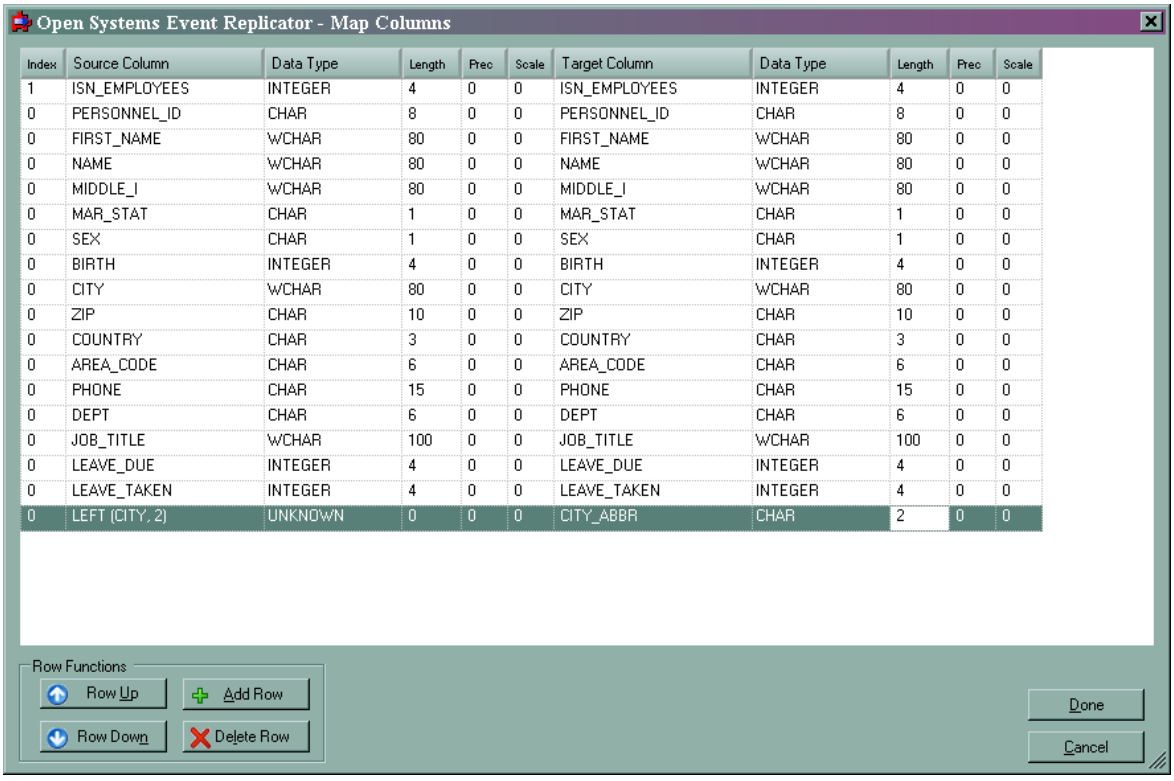
Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN_EMPLOYEES	INTEGER	4	0	0	ISN_EMPLOYEES	INTEGER	4	0	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	WCHAR	80	0	0	FIRST_NAME	WCHAR	80	0	0
0	NAME	WCHAR	80	0	0	NAME	WCHAR	80	0	0
0	MIDDLE_I	WCHAR	80	0	0	MIDDLE_I	WCHAR	80	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	BIRTH	INTEGER	4	0	0	BIRTH	INTEGER	4	0	0
0	CITY	WCHAR	80	0	0	CITY	WCHAR	80	0	0
0	ZIP	CHAR	10	0	0	ZIP	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	WCHAR	100	0	0	JOB_TITLE	WCHAR	100	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	0	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	0	0

3. Click **Add Row** to create a new entry in the column map.



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN_EMPLOYEES	INTEGER	4	0	0	ISN_EMPLOYEES	INTEGER	4	0	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	WCHAR	80	0	0	FIRST_NAME	WCHAR	80	0	0
0	NAME	WCHAR	80	0	0	NAME	WCHAR	80	0	0
0	MIDDLE_I	WCHAR	80	0	0	MIDDLE_I	WCHAR	80	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	BIRTH	INTEGER	4	0	0	BIRTH	INTEGER	4	0	0
0	CITY	WCHAR	80	0	0	CITY	WCHAR	80	0	0
0	ZIP	CHAR	10	0	0	ZIP	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	WCHAR	100	0	0	JOB_TITLE	WCHAR	100	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	0	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	0	0
0		UNKNOWN	0	0	0		UNKNOWN	0	0	0

- In **Source Column** enter "**LEFT (CITY, 2)"**; this will take the leftmost two characters in **CITY** and place them into a new column in the Target table
- In **Target Column** enter **CITY_ABBR**.
- In the target column **Data Type** select **CHAR** .
- In the target column **Length** enter **2**.



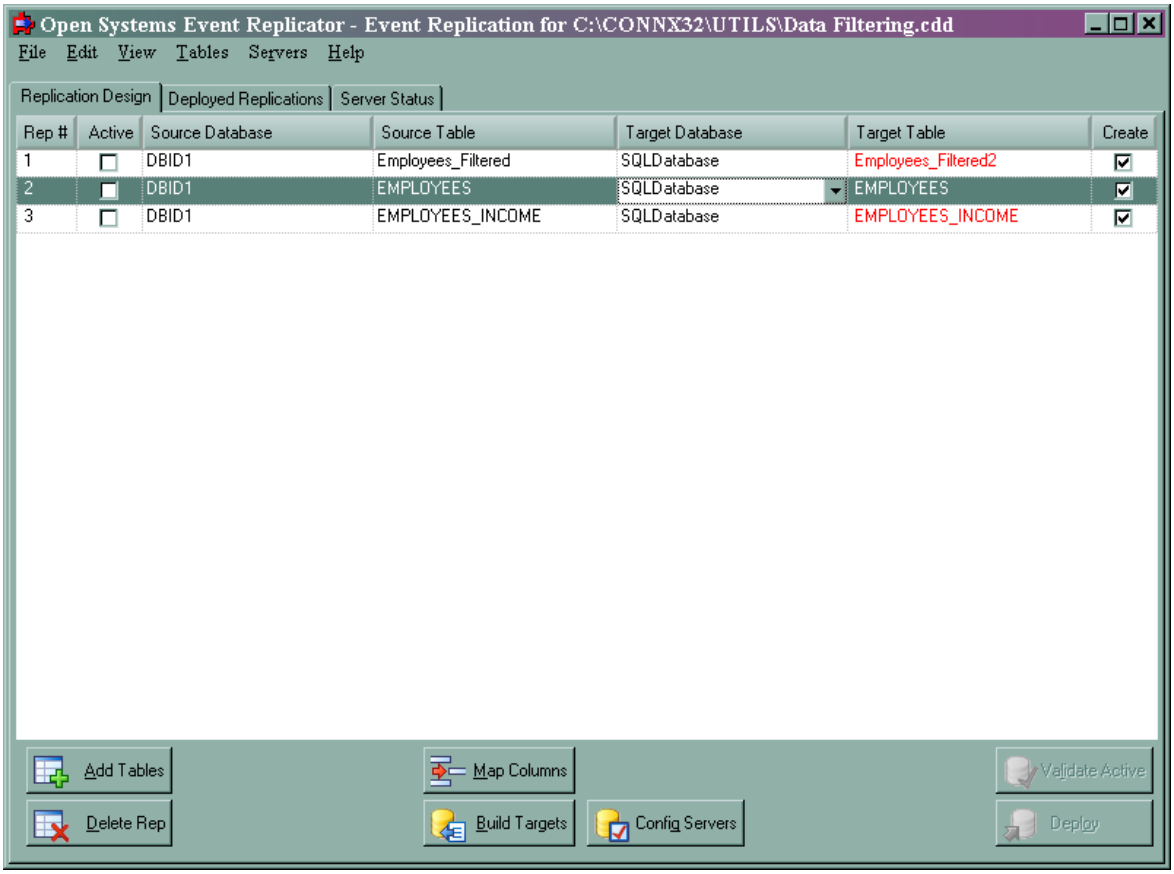
Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN_EMPLOYEES	INTEGER	4	0	0	ISN_EMPLOYEES	INTEGER	4	0	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	WCHAR	80	0	0	FIRST_NAME	WCHAR	80	0	0
0	NAME	WCHAR	80	0	0	NAME	WCHAR	80	0	0
0	MIDDLE_I	WCHAR	80	0	0	MIDDLE_I	WCHAR	80	0	0
0	MAR_STAT	CHAR	1	0	0	MAR_STAT	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	BIRTH	INTEGER	4	0	0	BIRTH	INTEGER	4	0	0
0	CITY	WCHAR	80	0	0	CITY	WCHAR	80	0	0
0	ZIP	CHAR	10	0	0	ZIP	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	WCHAR	100	0	0	JOB_TITLE	WCHAR	100	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	0	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	0	0
0	LEFT (CITY, 2)	UNKNOWN	0	0	0	CITY_ABBR	CHAR	2	0	0

4. Click **Done**. This completes using SQL expressions to split a column.

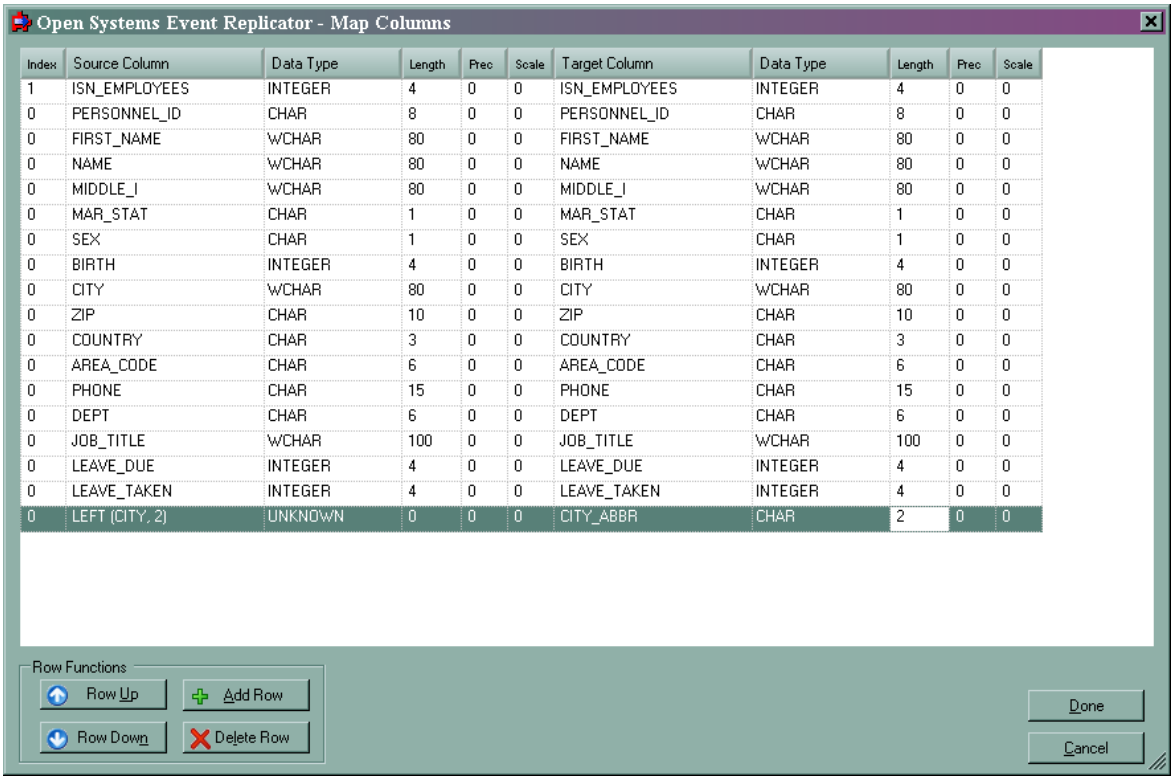
Renaming a Target Column

In this example, we will rename the **MAR_STAT** column to **Marital_Status** so the column name more clearly reflects its contents.

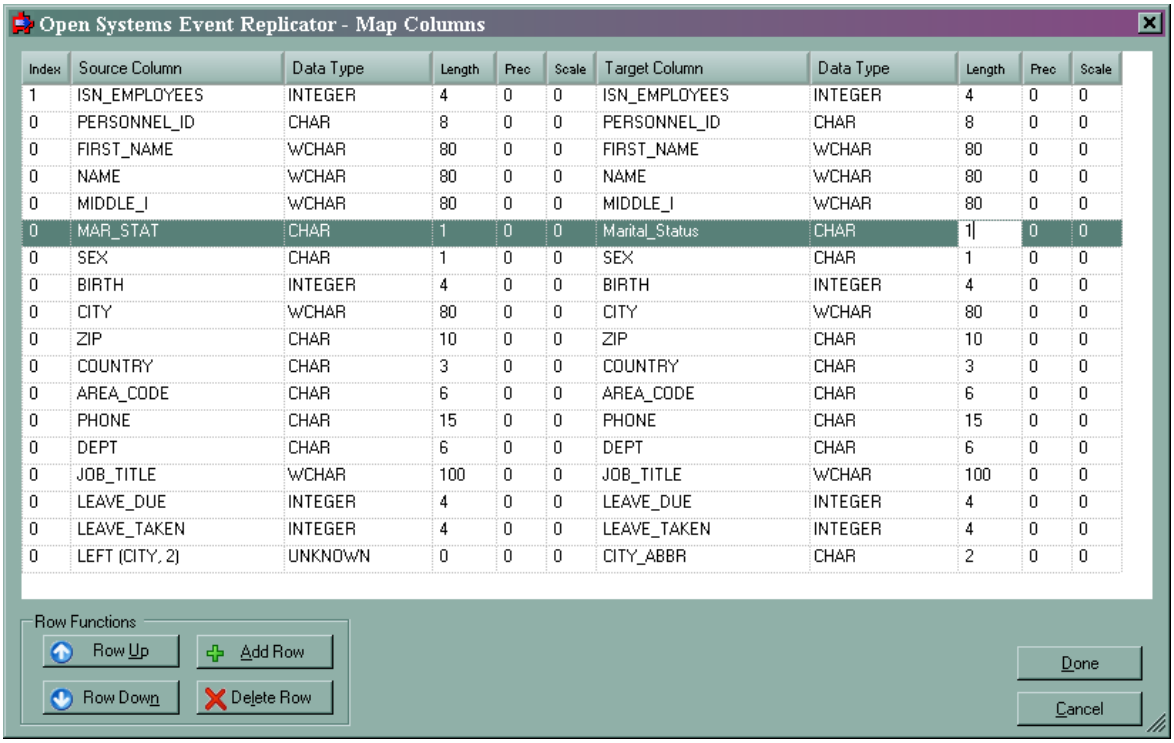
1. From the **Open Systems Event Replication** window select the **EMPLOYEES** table.



2. Click **Map Columns**. The **Map Columns** window appears.



- In the target column **MAR_STAT** field enter "**Marital_Status"**.
- In the target column **Data Type** select **CHAR**.
- In the target column **Length** enter **1**.



Index	Source Column	Data Type	Length	Prec	Scale	Target Column	Data Type	Length	Prec	Scale
1	ISN_EMPLOYEES	INTEGER	4	0	0	ISN_EMPLOYEES	INTEGER	4	0	0
0	PERSONNEL_ID	CHAR	8	0	0	PERSONNEL_ID	CHAR	8	0	0
0	FIRST_NAME	WCHAR	80	0	0	FIRST_NAME	WCHAR	80	0	0
0	NAME	WCHAR	80	0	0	NAME	WCHAR	80	0	0
0	MIDDLE_I	WCHAR	80	0	0	MIDDLE_I	WCHAR	80	0	0
0	MAR_STAT	CHAR	1	0	0	Marital_Status	CHAR	1	0	0
0	SEX	CHAR	1	0	0	SEX	CHAR	1	0	0
0	BIRTH	INTEGER	4	0	0	BIRTH	INTEGER	4	0	0
0	CITY	WCHAR	80	0	0	CITY	WCHAR	80	0	0
0	ZIP	CHAR	10	0	0	ZIP	CHAR	10	0	0
0	COUNTRY	CHAR	3	0	0	COUNTRY	CHAR	3	0	0
0	AREA_CODE	CHAR	6	0	0	AREA_CODE	CHAR	6	0	0
0	PHONE	CHAR	15	0	0	PHONE	CHAR	15	0	0
0	DEPT	CHAR	6	0	0	DEPT	CHAR	6	0	0
0	JOB_TITLE	WCHAR	100	0	0	JOB_TITLE	WCHAR	100	0	0
0	LEAVE_DUE	INTEGER	4	0	0	LEAVE_DUE	INTEGER	4	0	0
0	LEAVE_TAKEN	INTEGER	4	0	0	LEAVE_TAKEN	INTEGER	4	0	0
0	LEFT (CITY, 2)	UNKNOWN	0	0	0	CITY_ABBR	CHAR	2	0	0

3. Click **Done**. This completes renaming a target column

.

Preparing an Adabas database for Adabas to Adabas Replication

Before doing Adabas to Adabas replication, the source database must be initialized for replication. This includes:

- Running the Adabas tool ADADBM to initialize the source database

The ADADBM tool must be run with the parameter REPLICATION_FILES prior to doing Adabas to Adabas replication.

Note: You will not be able to run ADADBM with the REPLICATION_FILES parameter unless you have an A2A license for Adabas installed.

Command Format:

ADADBM REPLICATION_FILES=(<file1>, <file2>, <file3>, <file4>)

where:

- file1: metadata file
- file2: replication transaction file
- file3: replication command file
- file4: LOB file for replication command file

e.g: ADADBM DBID=1 REPLICATION_FILES=(15,16,17,18)

To uninitialize replication use the parameter **REMOVE_REPLICATION**

e.g: ADADBM DBID=1 REMOVE_REPLICATION

This will stop all replication and remove the replication system files. **Important:** Only do this if you no longer wish to replicate data

For more information about the ADADBM or other Adabas utilities please refer to the Software AG Adabas documentation.

Creating a CDD for Adabas to Adabas Event Replication using the CONNX SQL Engine

Creating a CDD for Adabas to Adabas Event Replication using the CONNX SQL Engine is the first step in getting data replicated. This includes:

- Opening the CONNX Data Dictionary (CDD) Manager
- Setting up the source and target Adabas connections
- Setting the correct CDD security
- Saving the CDD

Create a CDD for Adabas to Adabas Event Replication using the CONNX SQL Engine if:

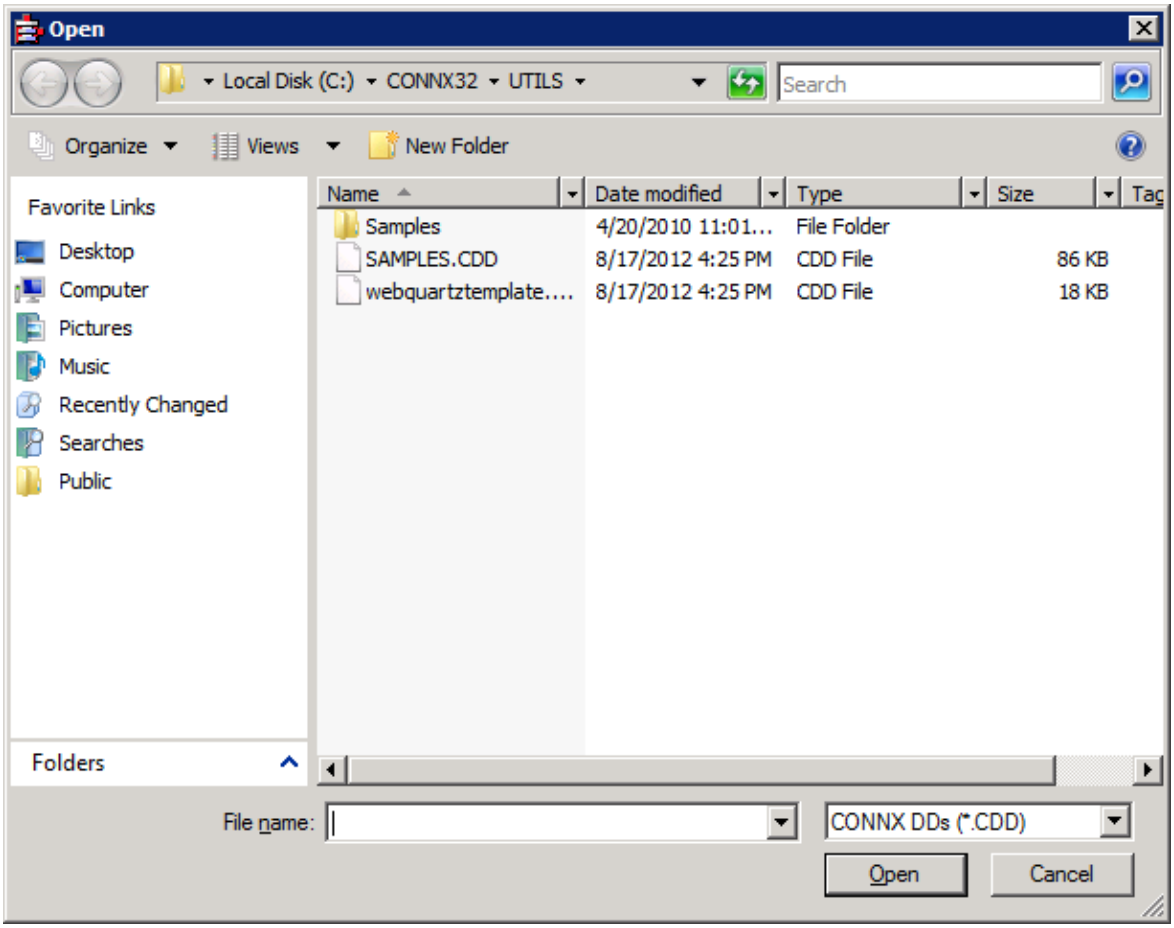
- You have never used the CONNX SQL Engine before
- You do not have an existing CDD containing the source and target Adabas connections needed for event replication.

After you create a CDD for Adabas to Adabas Event Replication using the CONNX SQL Engine you will be ready to:

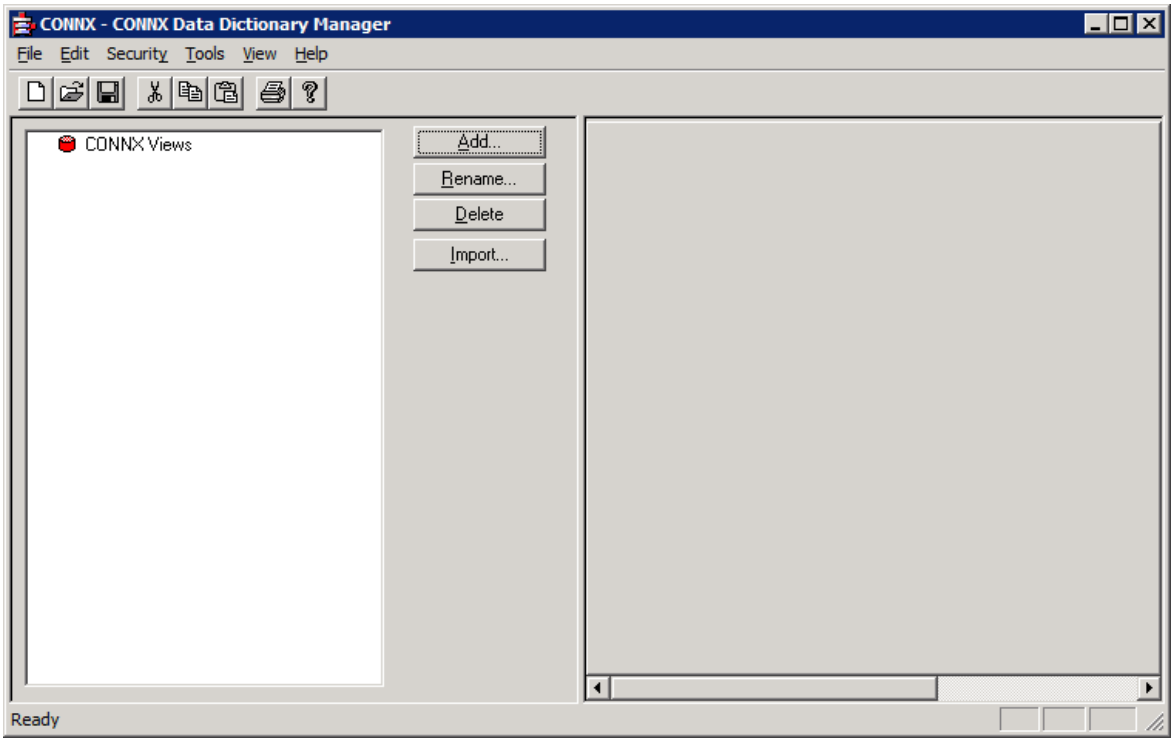
- Enable the Event Replicator

The Open Systems Event Replicator 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 and target databases.

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

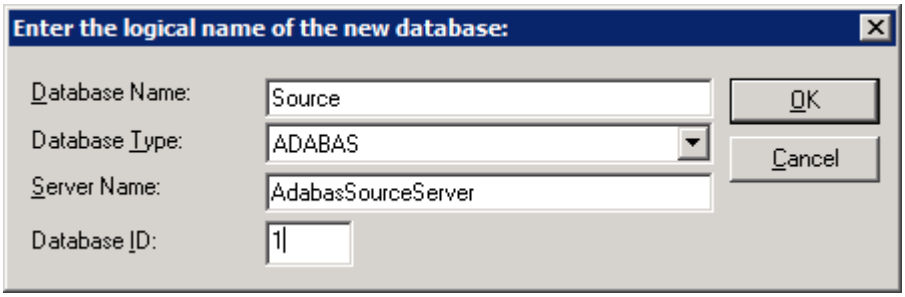


2. Click **Cancel**. The **CONNX Data Dictionary Manager** window appears.

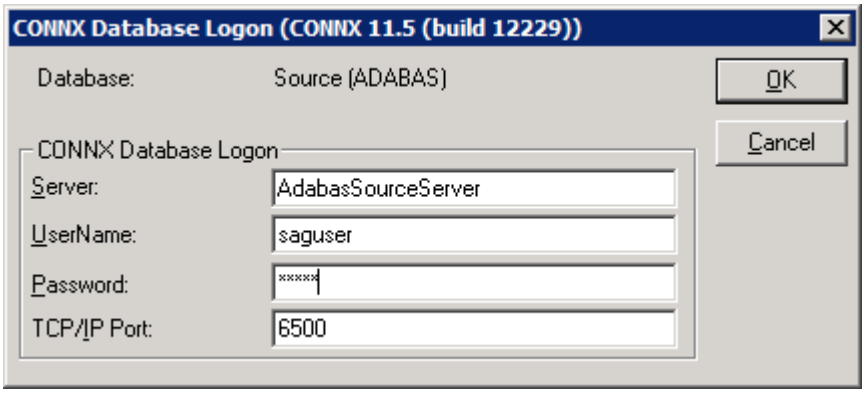


Adding the Source and Target Database connections

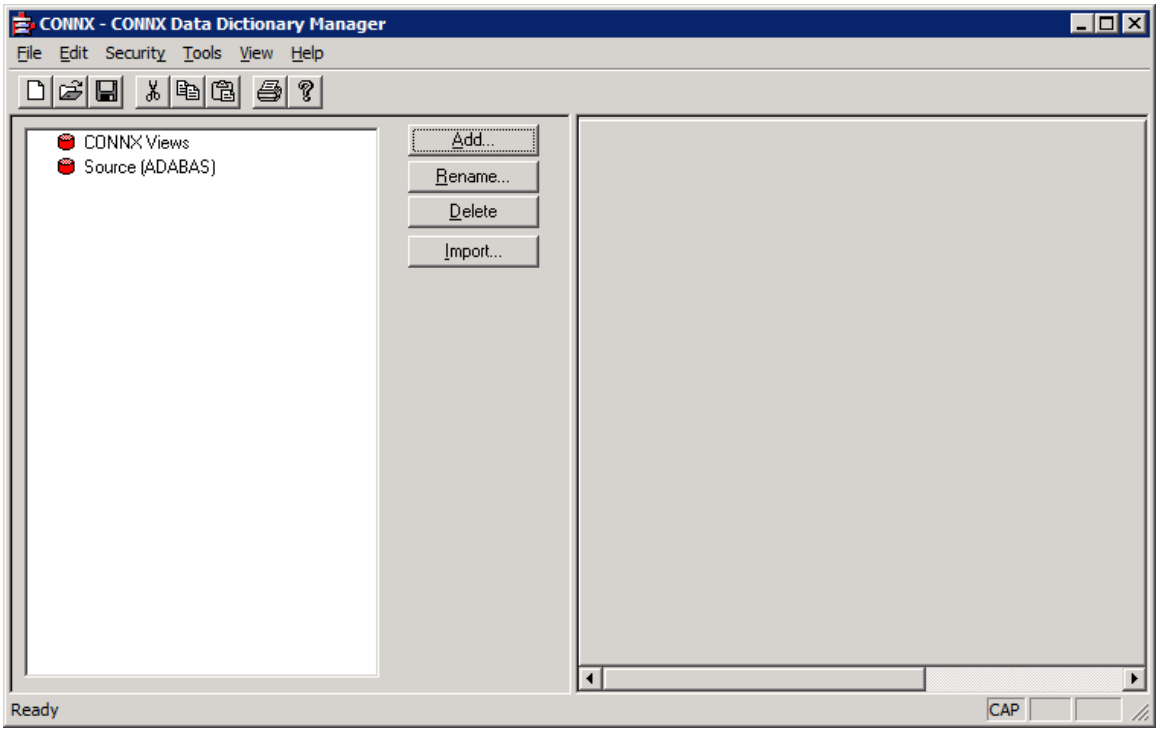
1. From the Tools menu, select Add Database Connection...



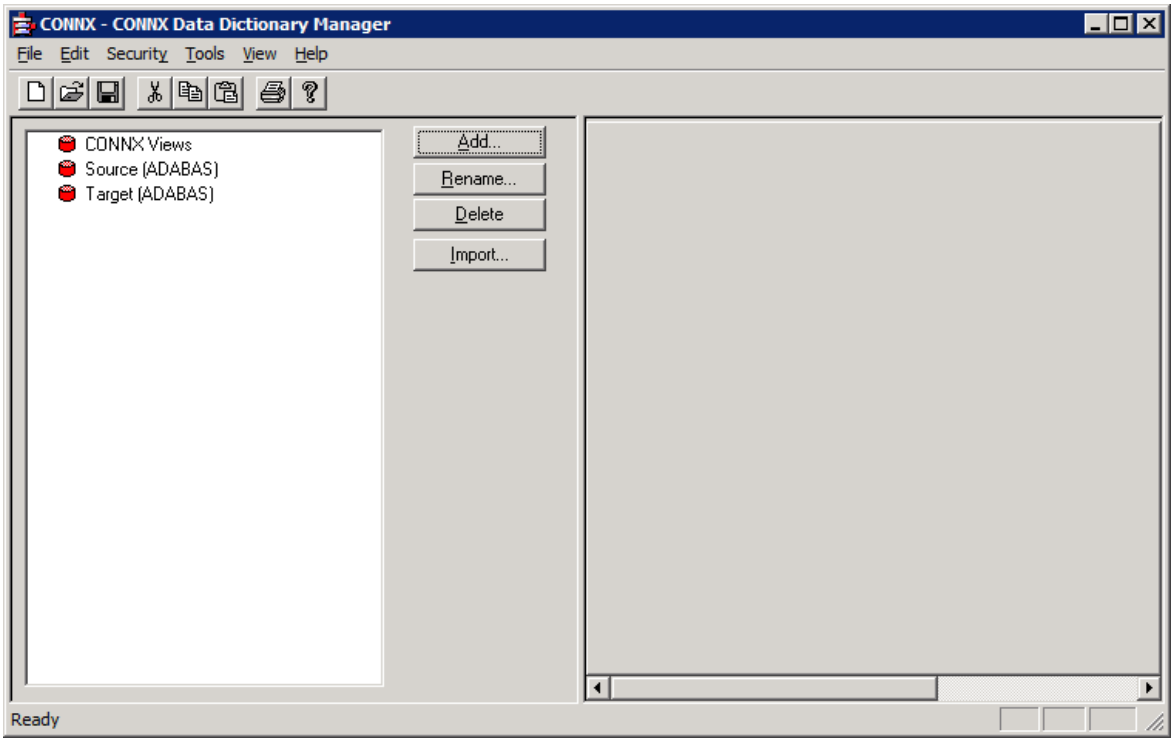
Enter logon information



2. You now have a connection to the source database defined in the CDD

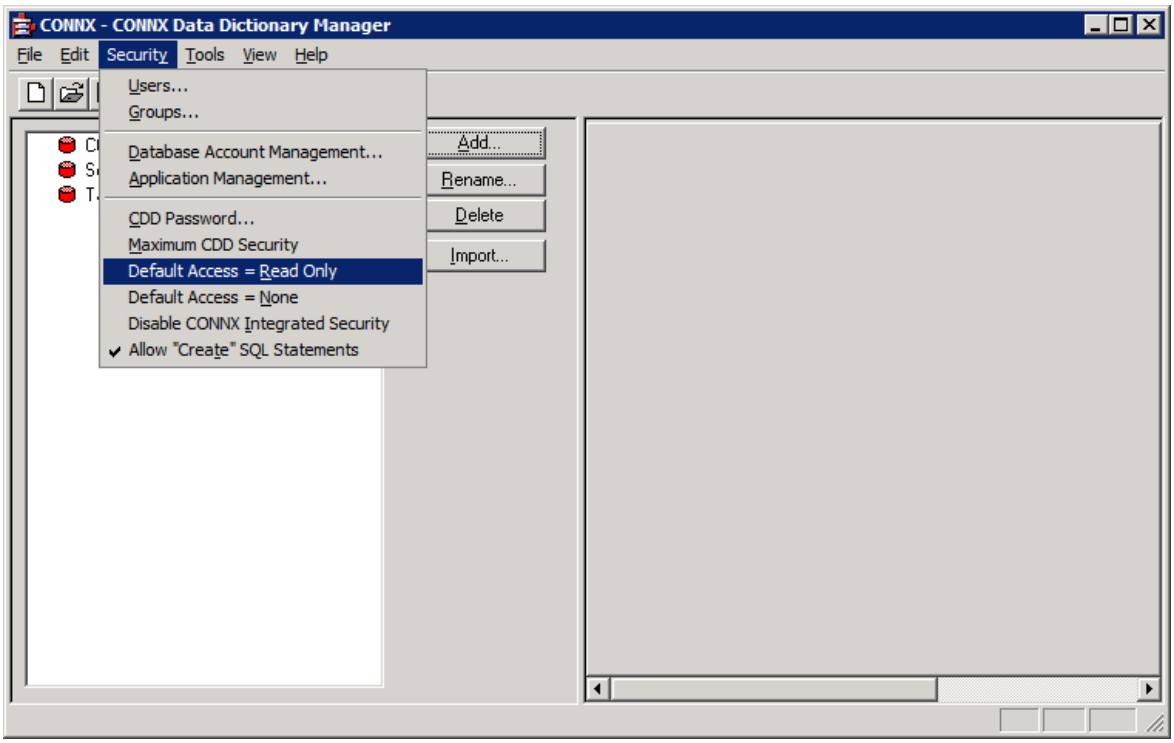


3. Repeat Step 1 for the target database. You will now have the source and target database connections defined in the CDD

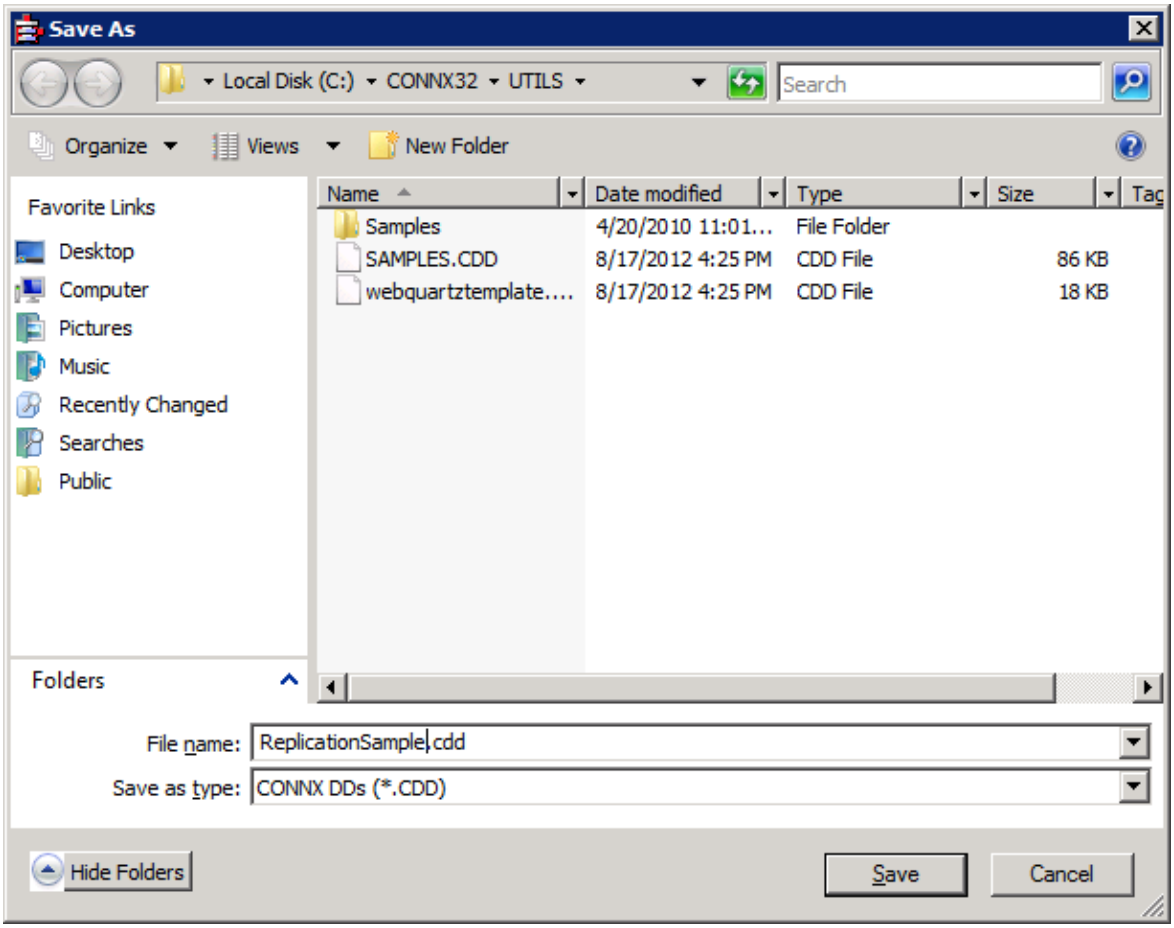


Set the CDD Security

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 Adabas to Adabas Event Replication

Enabling a CDD for Adabas to Adabas 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 files 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 use the Open Systems Event Replicator.
- 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:

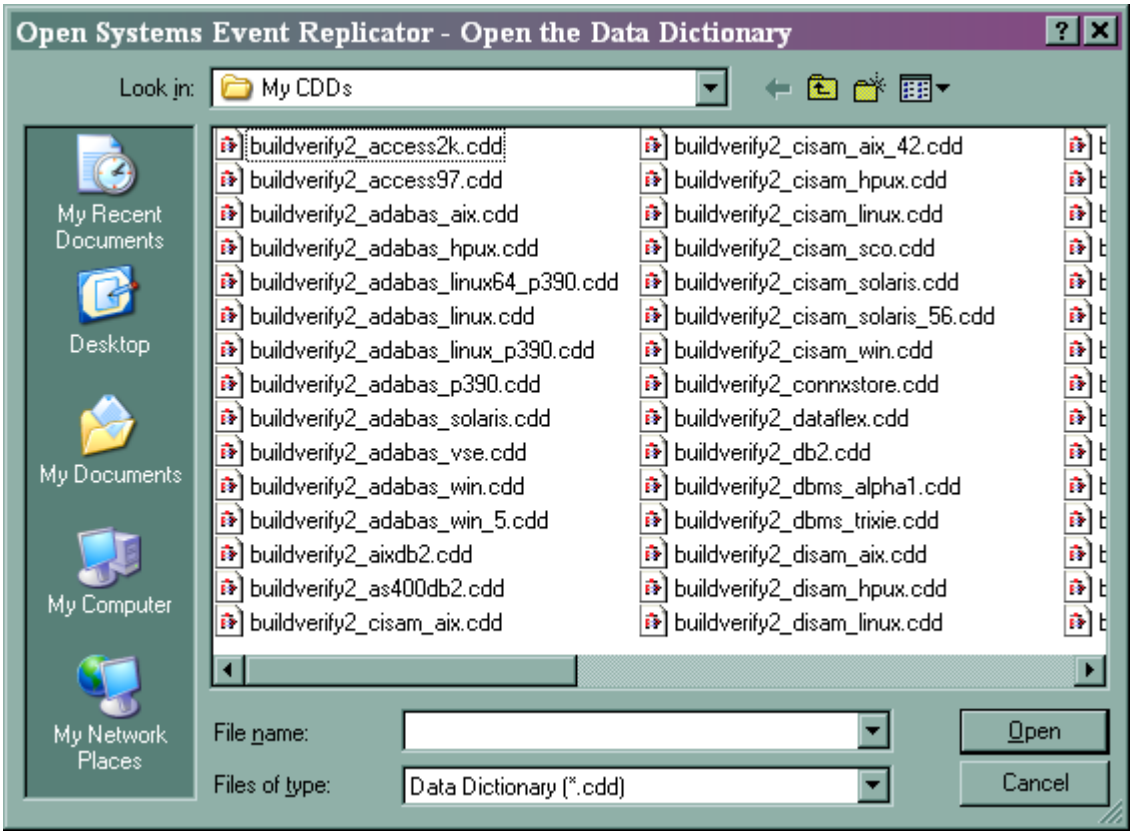
- Deployed (which starts replicating data from source database to target database)

The Event Replication 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.

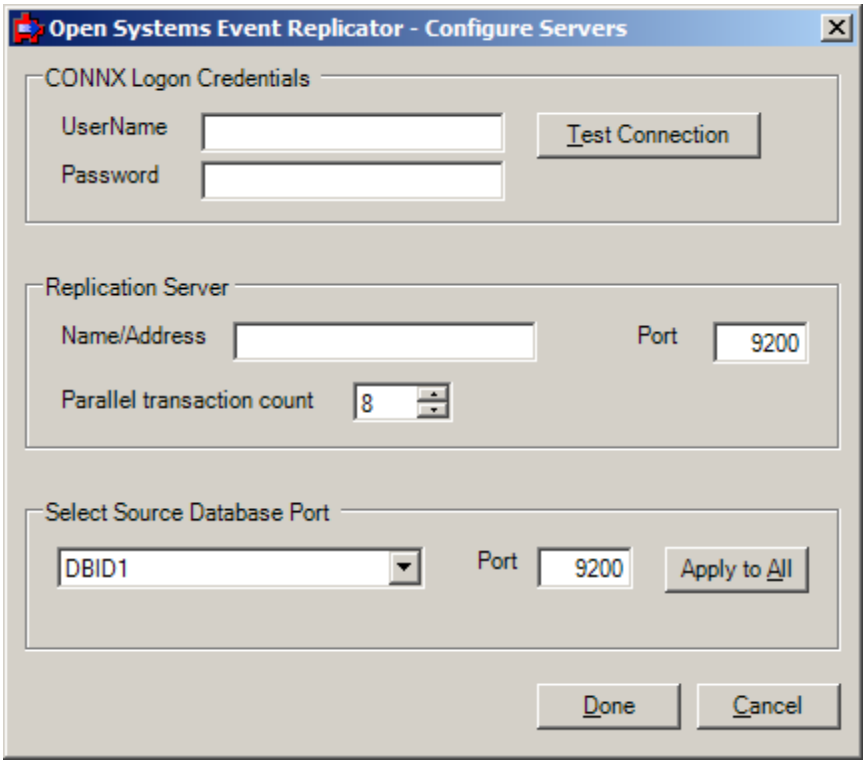
Caution: Two separate CDDs cannot replicate to the same targets from two different controllers. The Event Replicator is designed to maintain data integrity from the source to the target; any target modifications from more than one controller will result in replication errors, disabling the Event Replicator.

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

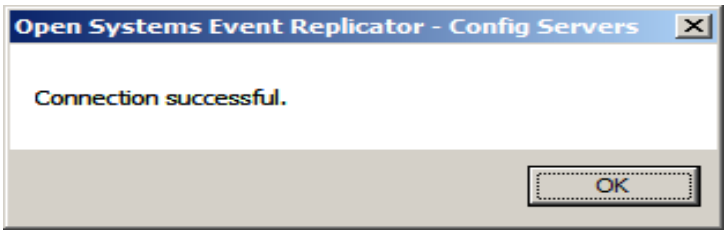


Note: Only CONNX Data Dictionaries (CDD) that contain both the source and target (destination) data bases can be used for replication. Both the source and target database must be for Adabas version 6.3 SP1 or later.

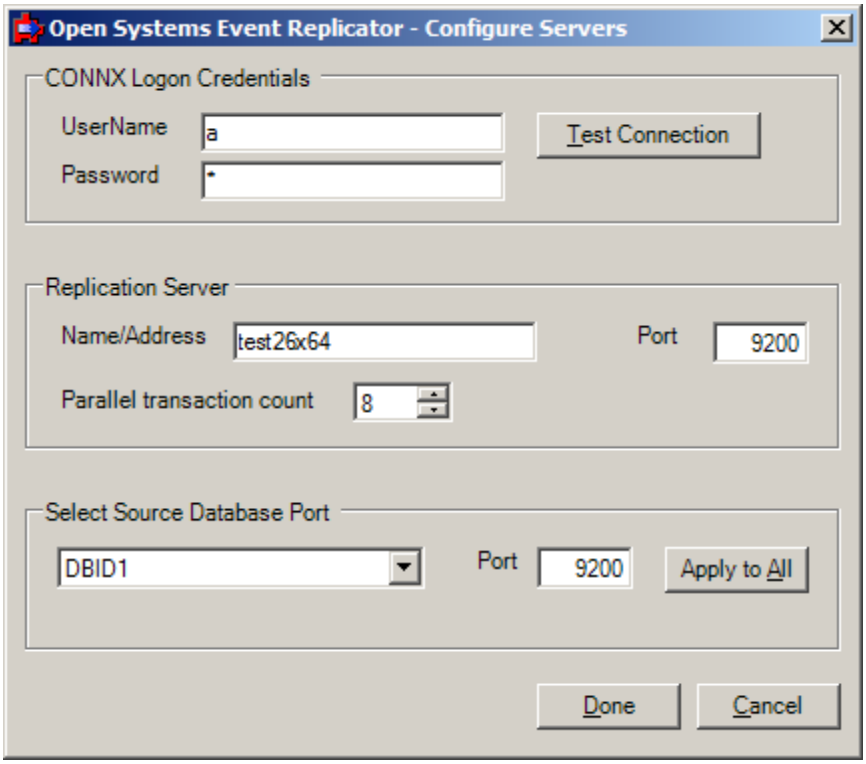
2. If the Replication Administrator has been used previously, the **Open Systems Event Replication** window appears. The Event Replicator will normally open the last opened CDD.
    - If this is the CDD you want, you do not need to re-enable the Event Replicator for this CDD.
    - If this is not the 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.



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:

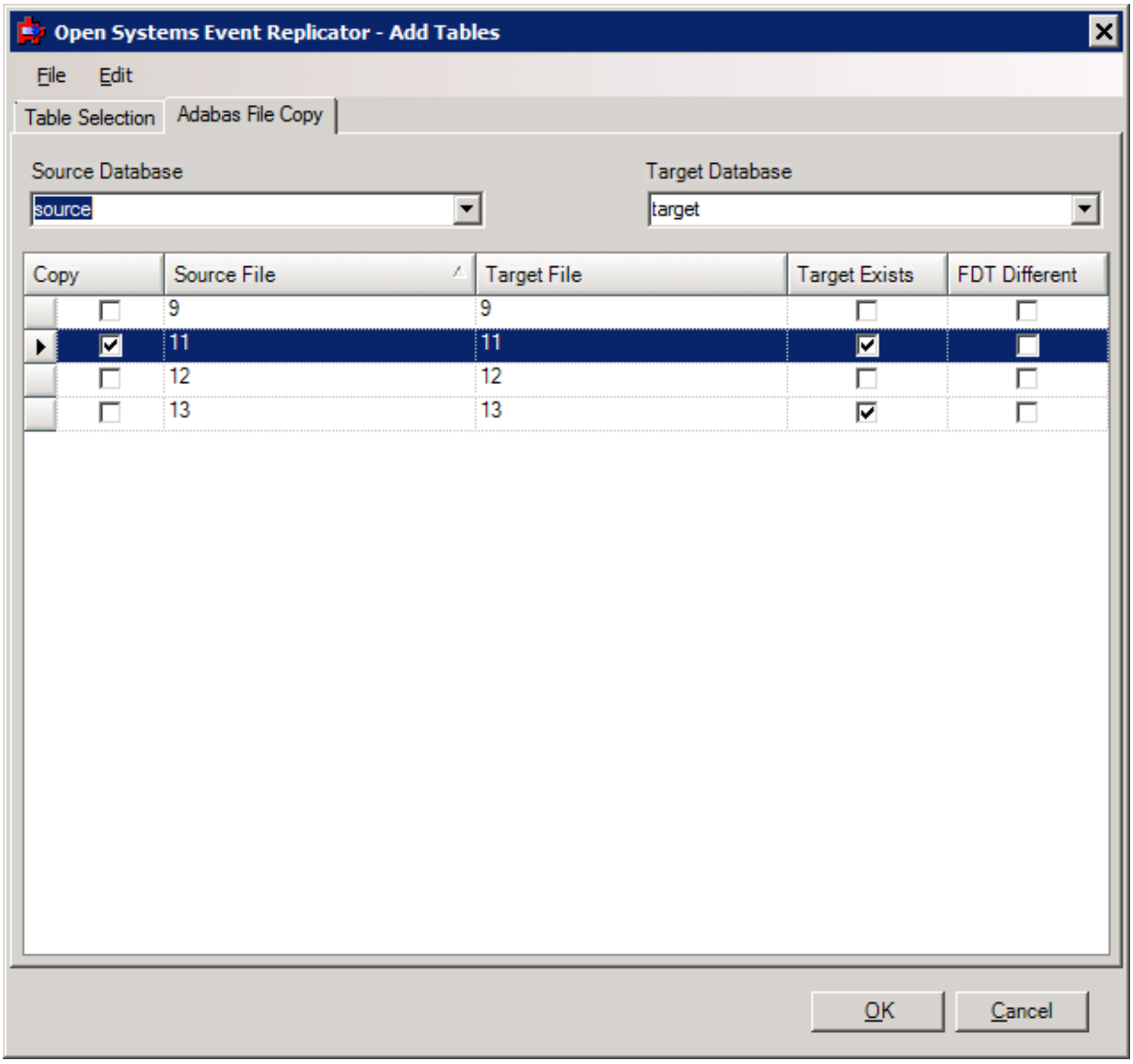


5. Click **OK**. The message window closes.
6. Enter the server name or IP address of the Replication Server in **Name/Address** and the server port number in **Port**.



Note: The default port number is 9200.

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



8. Select the source file number(s) you wish to replicate.

   Note: the target file does not need to exist, but if a file with the same file number does exist on the target, the fdt's will be compared and if they are different, a warning will appear. The target file must have the same file number as the source, so if there is a conflict it will need to be resolved prior to continuing.

9. Click **OK**. The **Open Systems Event Replication** window appears.

Adding an Adabas to Adabas File Copy Replication to the Event Replicator

Adding an Adabas to Adabas File Copy Replication to the Open Systems Event Replicator includes:

- Opening a CONNX Data Dictionary (CDD)
- Selecting a source nucleus from the Adabas nucleus in the CDD
- Selecting a target Adabas nucleus
- Select the file numbers that will be replicated

Add an Adabas to Adabas File Copy replication if:

- You wish to replicate a source file exactly to another Adabas nucleus file
- You do not wish to change or modify the data during replication, just transfer the data exactly as it is

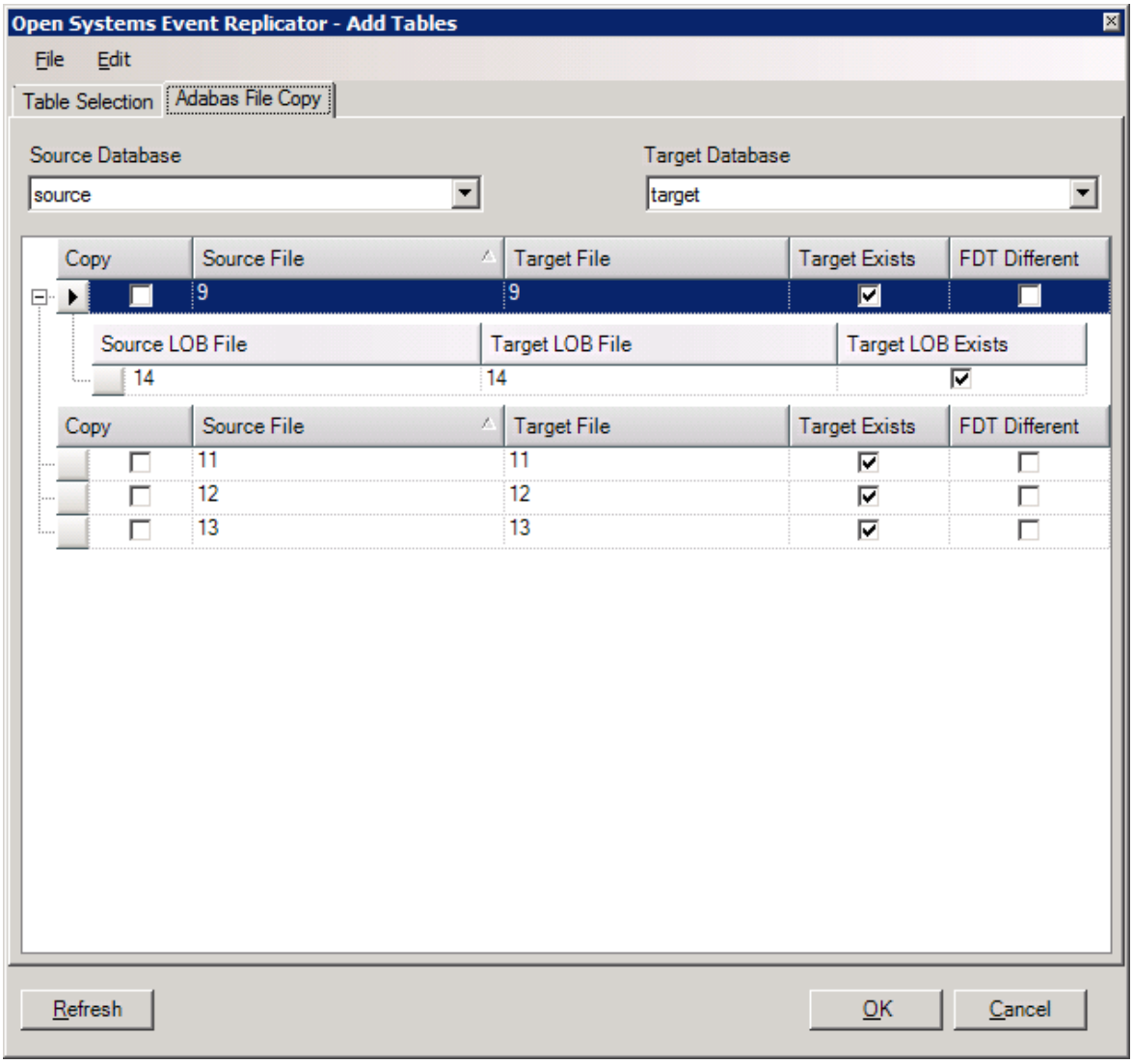
After you add a replication you will be able to:

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

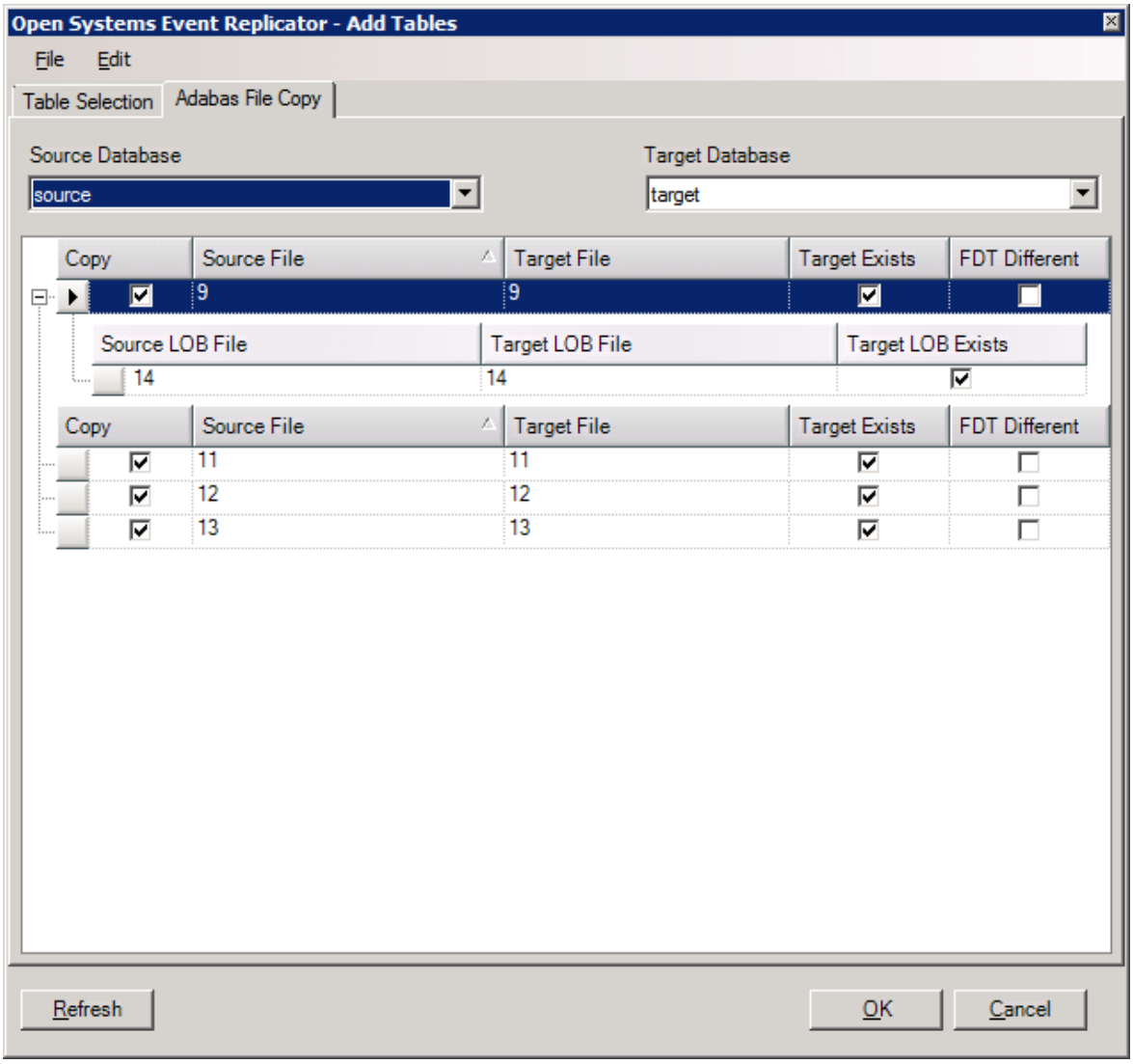
You can add many source files to the Event Replicator and associate them with a target database at the same time.

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

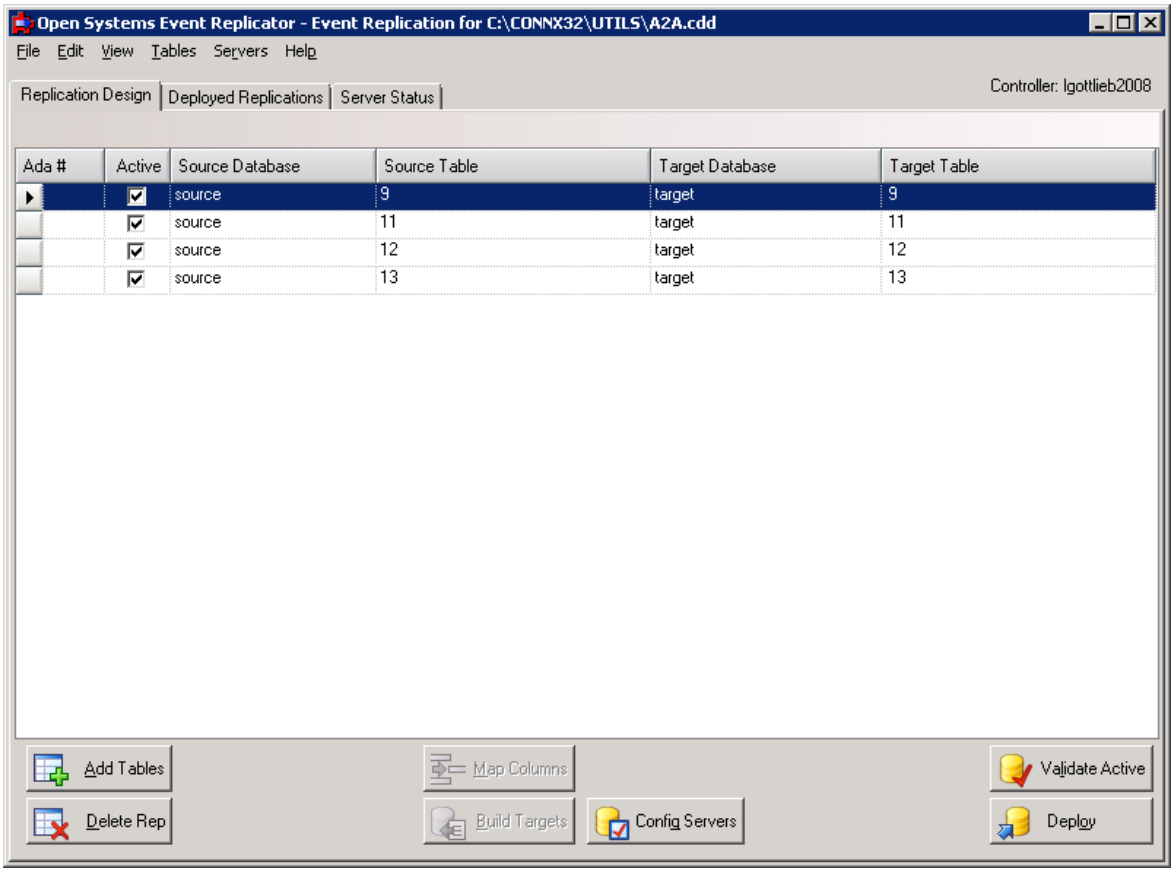
1. Open the Open Systems Event Replicator CDD.
2. Click Add Tables. The Add Tables window appears. If this is a new CDD then the Add Tables form will open automatically. Select the Adabas File Copy tab and you will see dropdowns to select the Source and Target Database. Once selected, the file numbers available to replicate will be listed. *If you do not see the "Adabas File Copy" tab check the following: Are there more than one Adabas nucleus with a version 6.3.1.03 or higher imported into the CDD? Do you have a license for Adabas to Adabas File Copy replication?



3. Select the "Copy" checkbox for the files you wish to replicate. On the right is information about the same file number on the target database. It will tell if the target file already exists and if the fdt of the target is different then the source before the replication is started. It should be noted that target tables are dropped and recreated on deploy, so check to make sure the target database is the correct one.



4. Click **OK**. The **Open Systems Event Replication** window appears and the files you added are in the list. The "Map Columns" and "Build Targets" buttons are disabled because the functionality does not apply to A to A file Copy replications.



5. If you want to deploy the changes you've made, 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**.

   For more information about **Validate Active**, see Validating Active Servers. For more information about **Deploy**, see Deploying the Event Replicator.

To assign a different file number to the target file, please see the section on Renumbering Target Files.

Note: The list of files displayed on the Add Tables dialog is cached in the Replication Administrator the first time this dialog is opened. If files are added or removed from Adabas while the Replication Administrator is running, you may need to use the Refresh button to refresh the list of files.

Renumbering Target Files

Renumbering files on the target system includes:

- Opening a CONNX Data Dictionary (CDD)
- Selecting a source nucleus from the Adabas nucleus in the CDD
- Selecting a target Adabas nucleus
- Select the file numbers that will be replicated from the source
- Specify a new file number for the target

Use the renumber feature if:

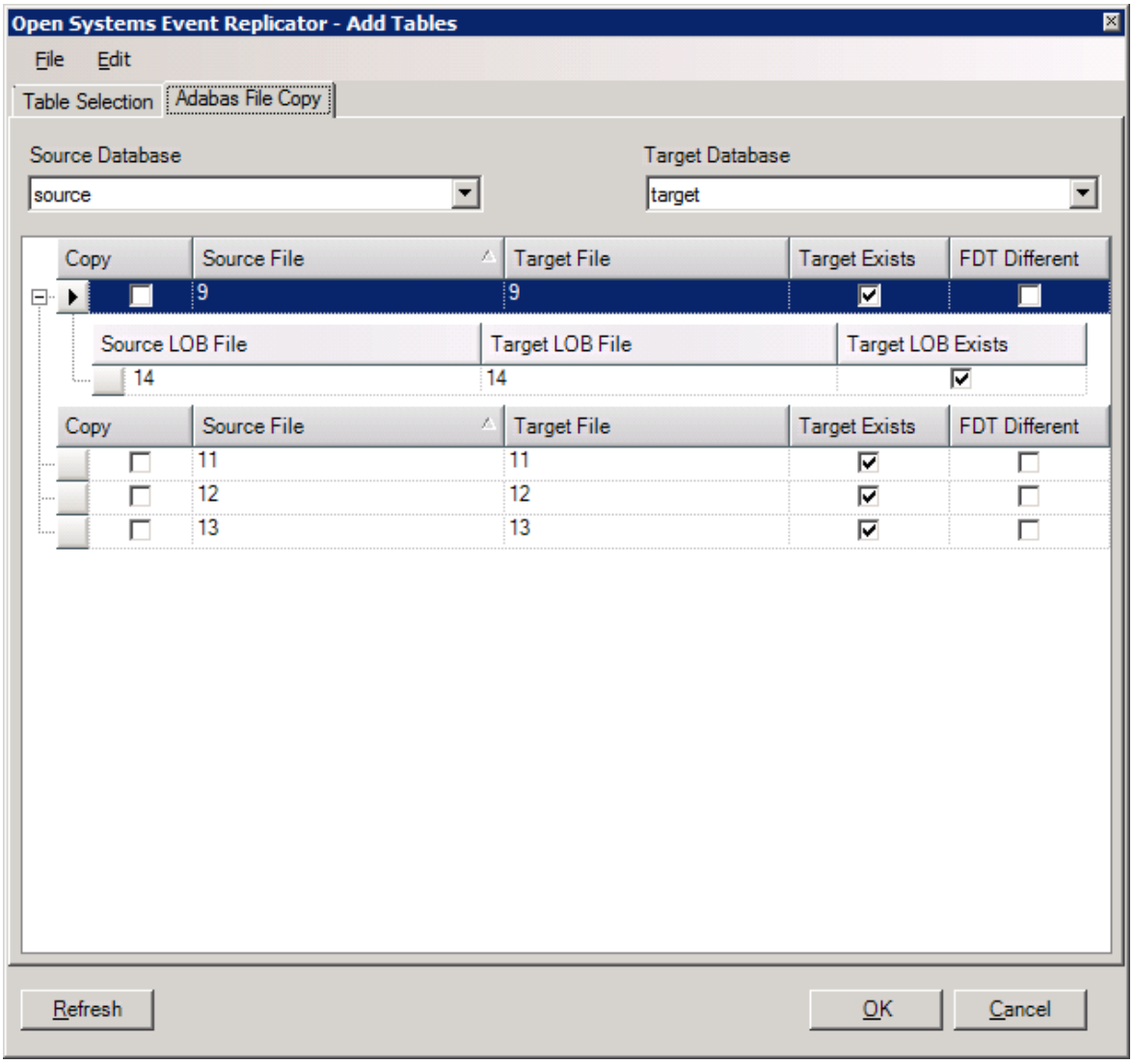
- You wish to replicate Adabas files but you want the target files to have different file numbers than the source

After you add a replication with renumbered target files you will be able to:

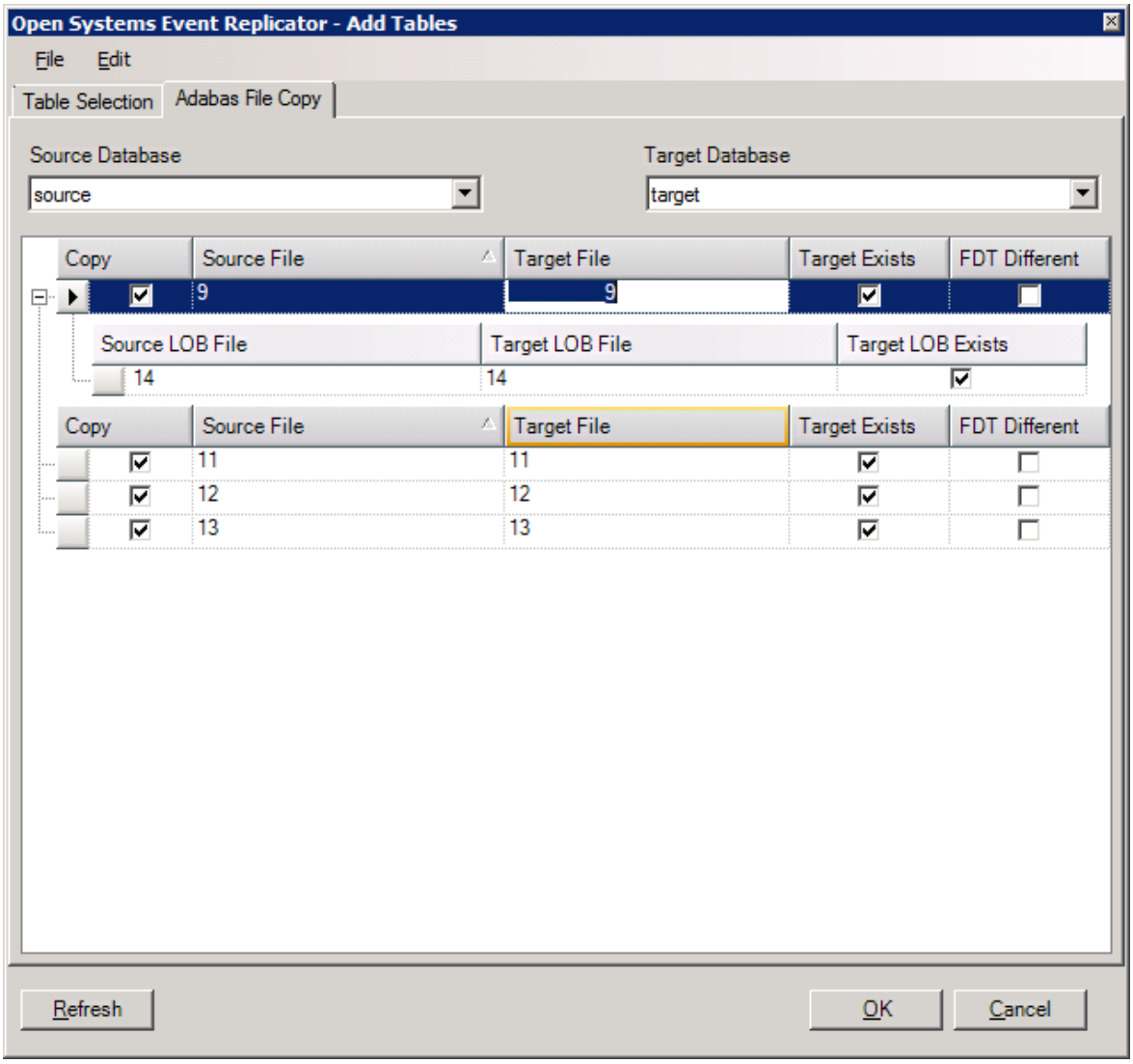
- Deploy it (which will start replicating data from source database to target database)

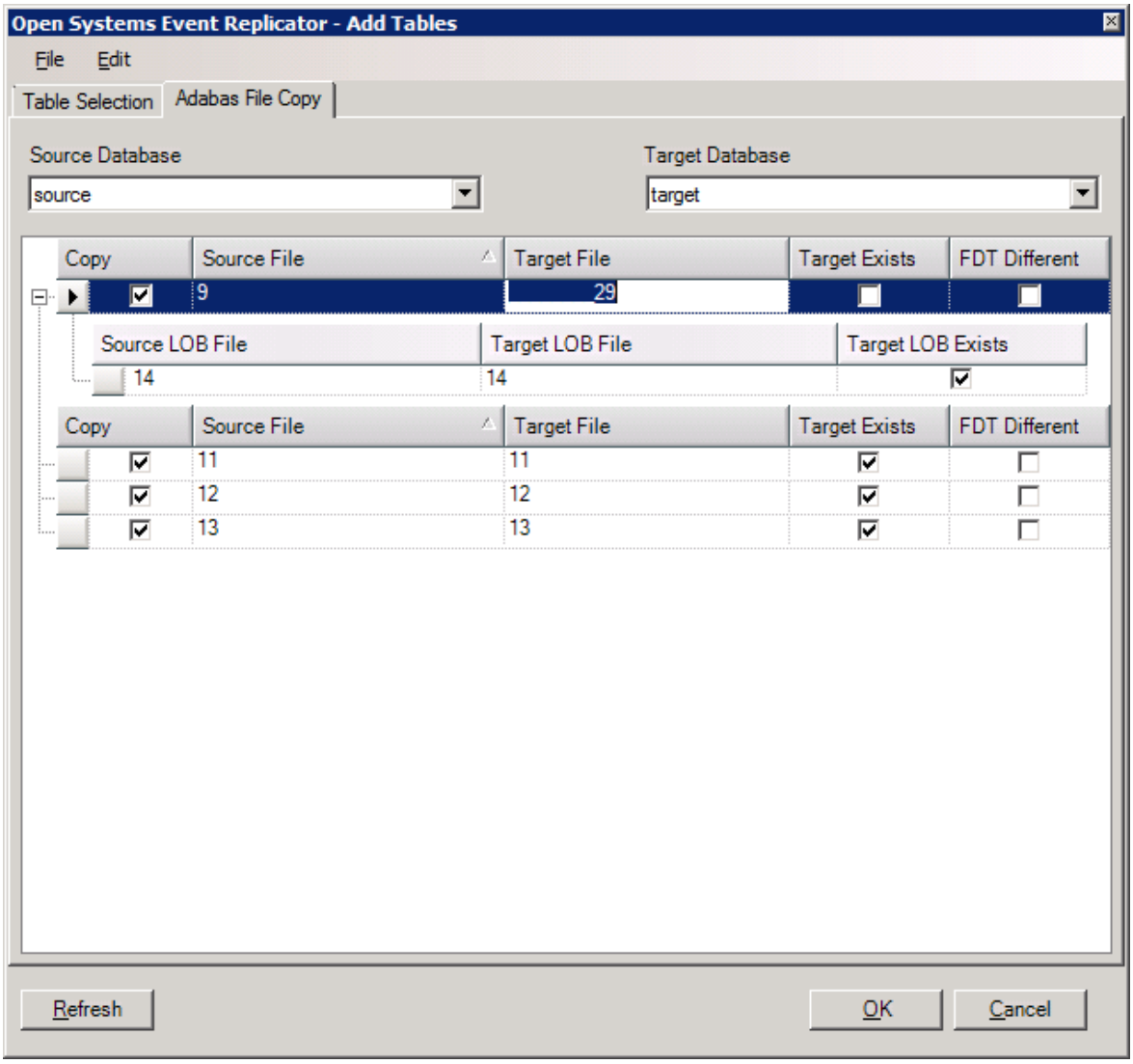
You can renumber the target file in the Add Tables dialog..

1. Open the Open Systems Event Replicator CDD.
2. Click Add Tables. The Add Tables window appears. If this is a new CDD then the Add Tables form will open automatically. Select the Adabas File Copy tab and you will see dropdowns to select the Source and Target Database. Once selected, the file numbers available to replicate will be listed. *If you do not see the "Adabas File Copy" tab check the following: Are there more than one Adabas nucleus with a version 6.3.1.03 or higher imported into the CDD? Do you have a license for Adabas to Adabas File Copy replication?



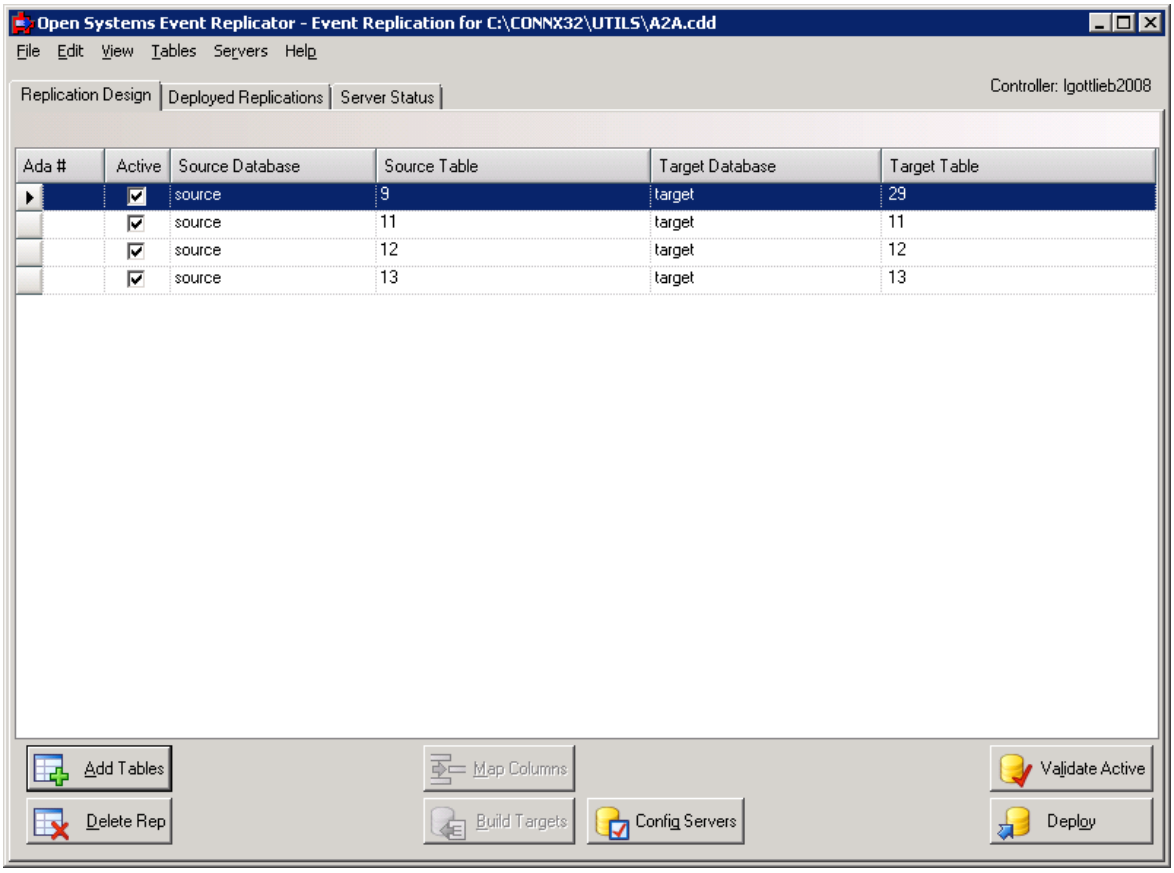
3. Select the "Target File" field for the file to be renamed and enter the new file number. Below is an example of changing file 9 to file 29


Open Systems Event Replicator - Add Tables
File
Edit
Table Selection
Adabas File Copy
Source Database
source
Target Database
target
Copy
Source File
Target File
Target Exists
FDT Different
9
9
Source LOB File
Target LOB File
Target LOB Exists
14
14
Copy
Source File
Target File
Target Exists
FDT Different
11
11
12
12
13
13
Refresh
OK
Cancel



If the source file has a LOB file associated with it, the LOB file will be displayed below the source file in the grid. In this case, the LOB file for file 9 is file 14. The LOB file can also be renumbered in the same manner. Click in the Target LOB File field and change the number.

4. Click **OK**. The **Open Systems Event Replication** window appears and the files you added are in the list. The "Map Columns" and "Build Targets" buttons are disabled because the functionality does not apply to A to A file Copy replications. Note that the Target Table number reflects the renumbered table. LOB files are not displayed on this screen.



5. If you want to deploy the changes you've made, 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**.

For more information about **Validate Active**, see Validating Active Servers. For more information about **Deploy**, see Deploying the Event Replicator.

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.

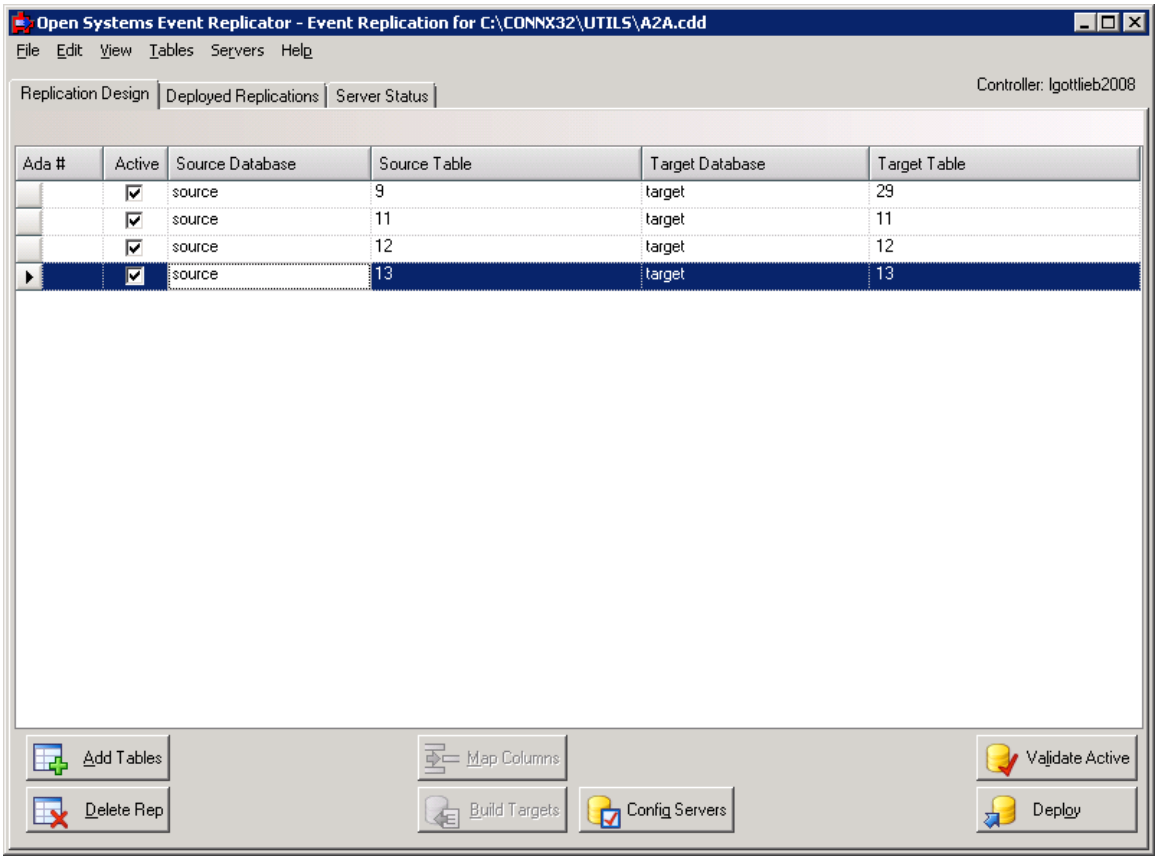
After you delete a replication you will be able to:

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

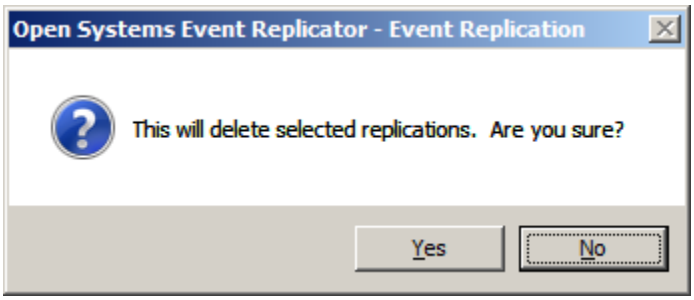
You can only delete one replication at a time.

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 Open Systems Event Replicator CDD.
2. Select the replication you wish to delete and click **Delete Rep**.

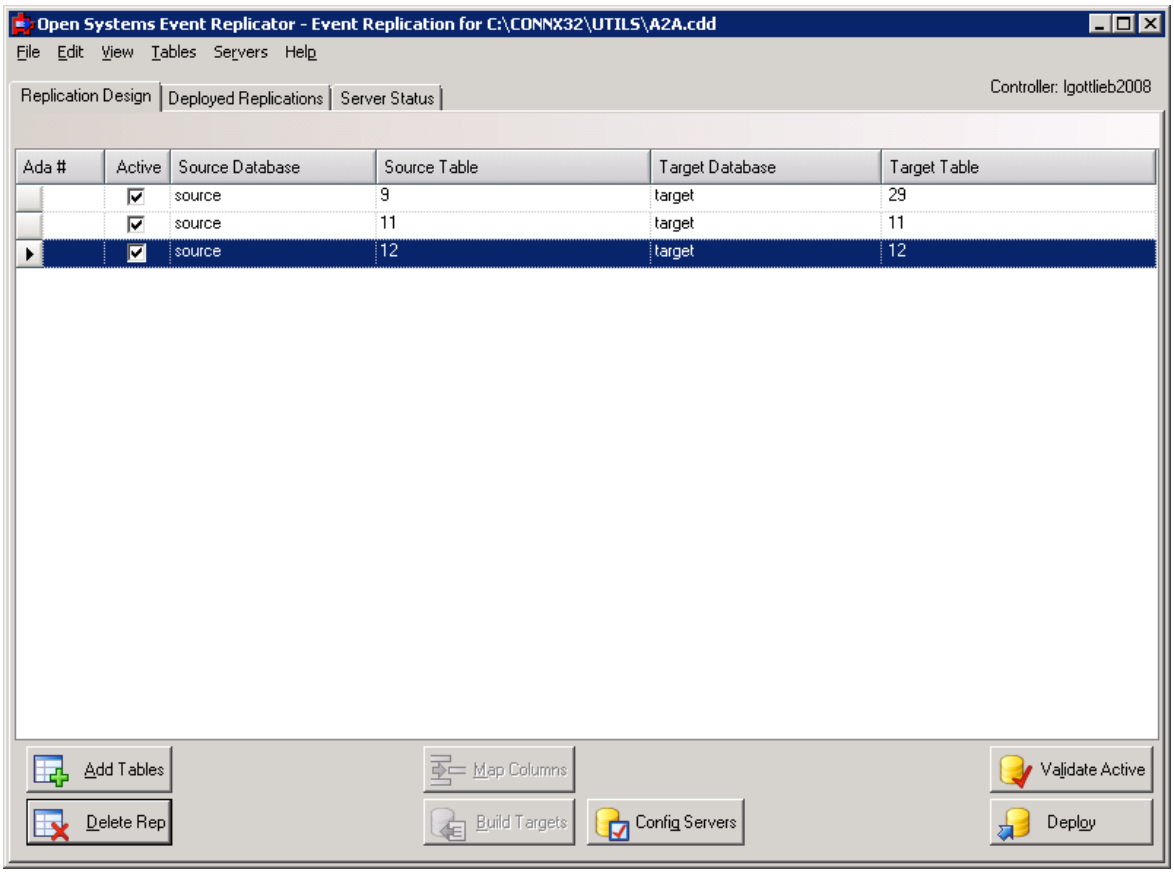


3. The following message appears:



3. Click **Yes**. The selected replication will be deleted no matter whether the replication is marked **Active** or **Create**.

The replication no longer appears in the list of those available to be deployed.



4. When you are done deleting replications, click the **File** menu and then click **Save**.

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

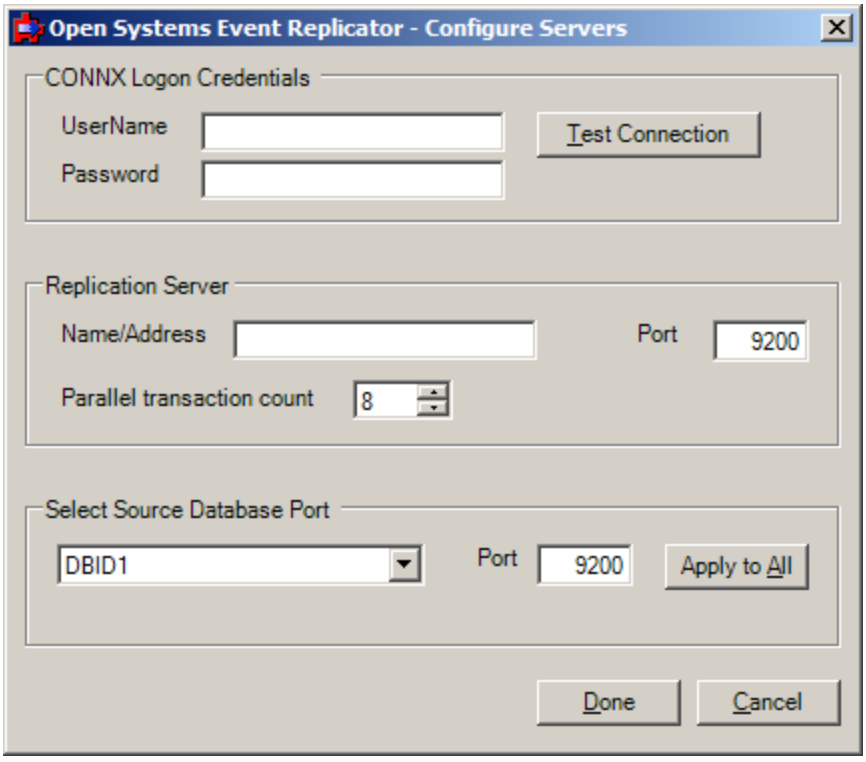
Configure the replication and source database servers if:

- The configuration information is missing, changed, or incorrect

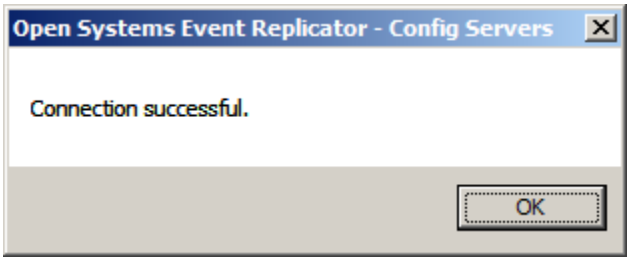
After you configure the replication and source database servers you will be able to:

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

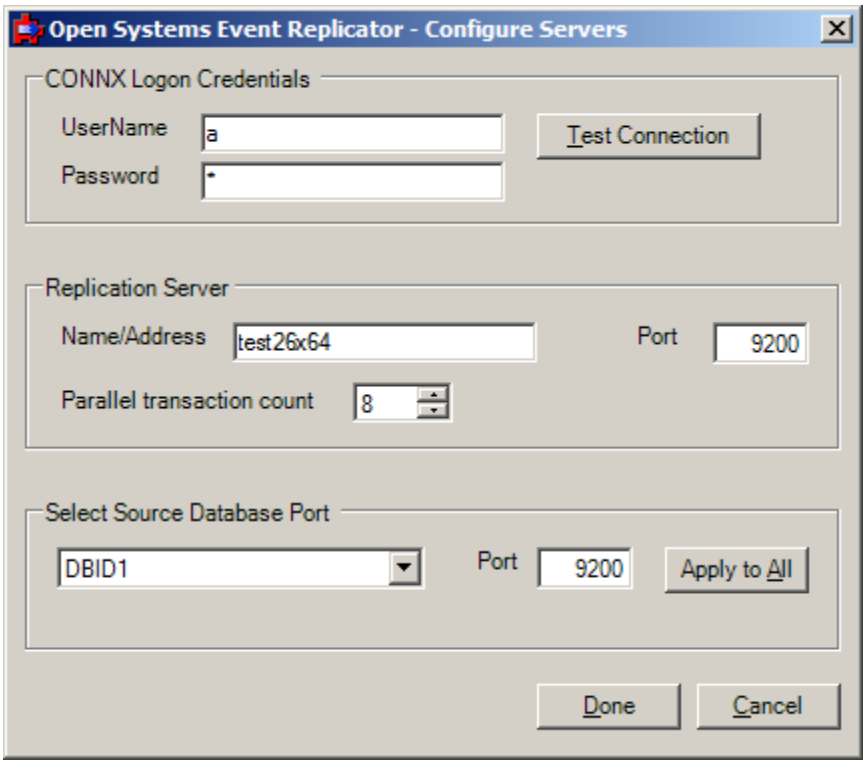
1. Open the Open Systems Event Replicator CDD.
2. Click **Config Servers**. The **Configure Servers** window appears.



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.



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:

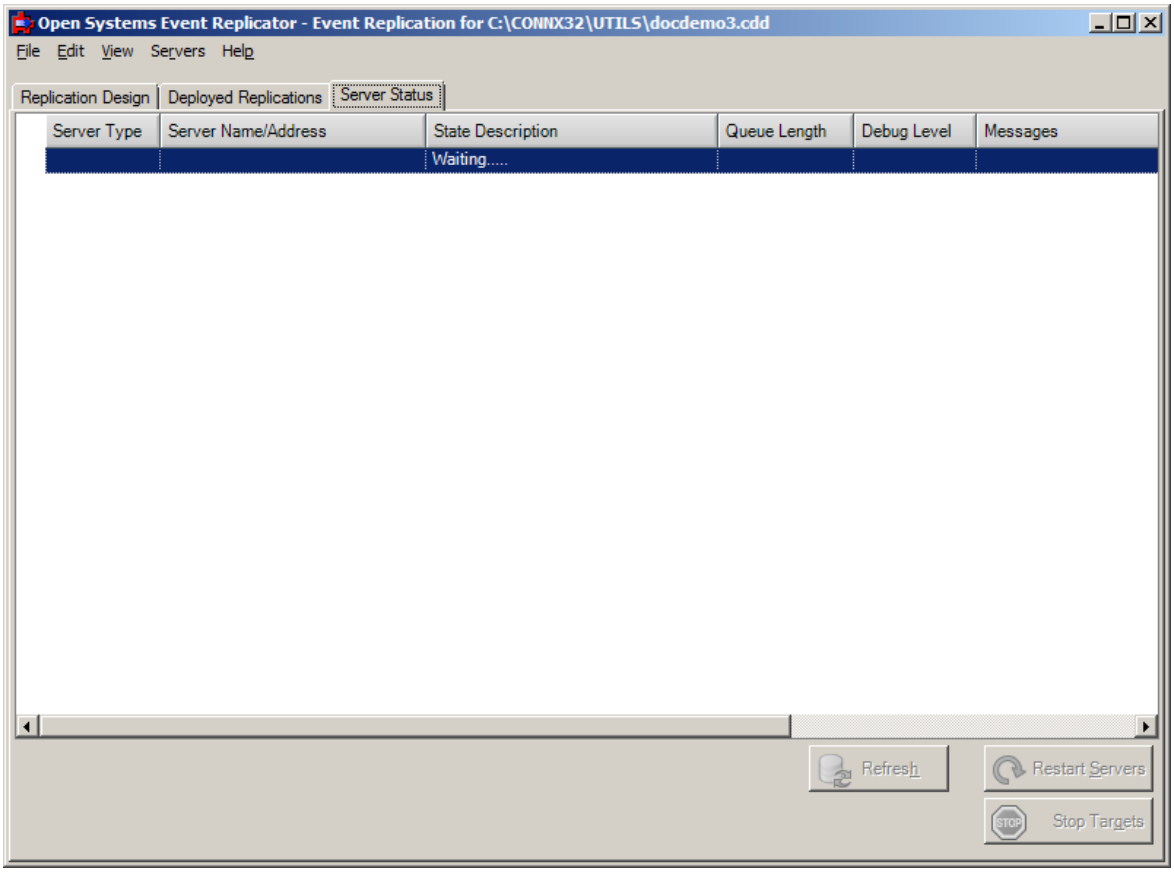
- The log file contains error messages about a server.

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

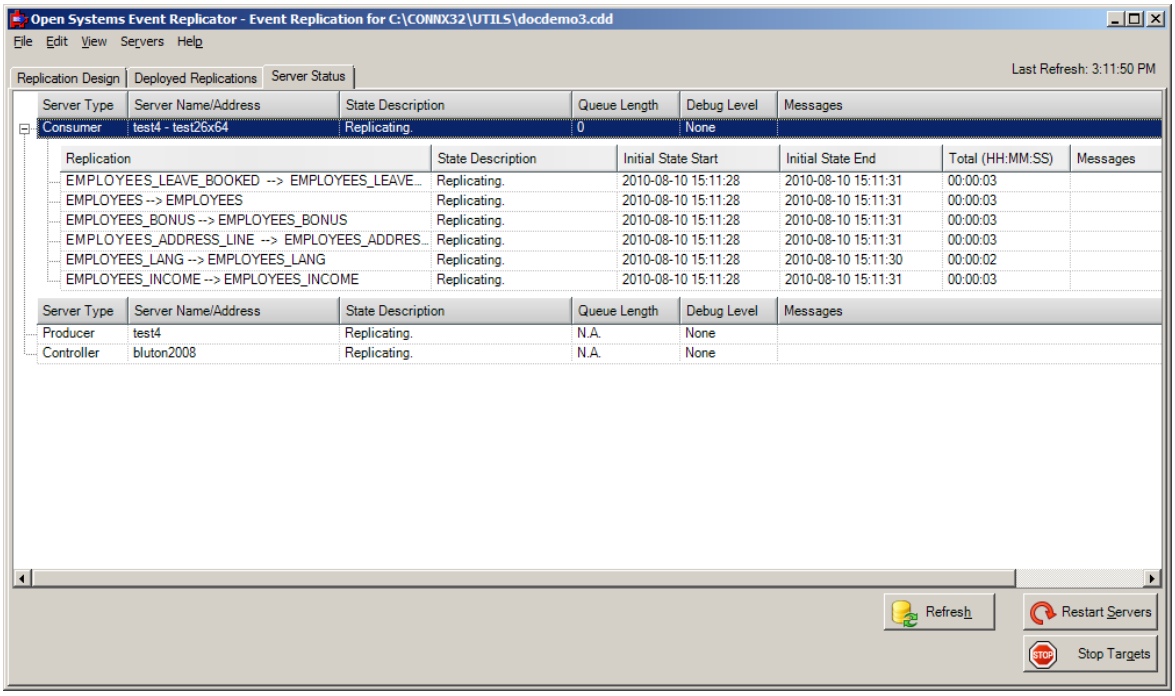
- Stop the replication servers
- Restart the replication servers

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

1. Open the Open Systems Event Replicator CDD.
2. Click the **Server Status** tab. The following appears:



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



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. Potential values and their meaning are as follows:

  - **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.

Deploying the Event Replication

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

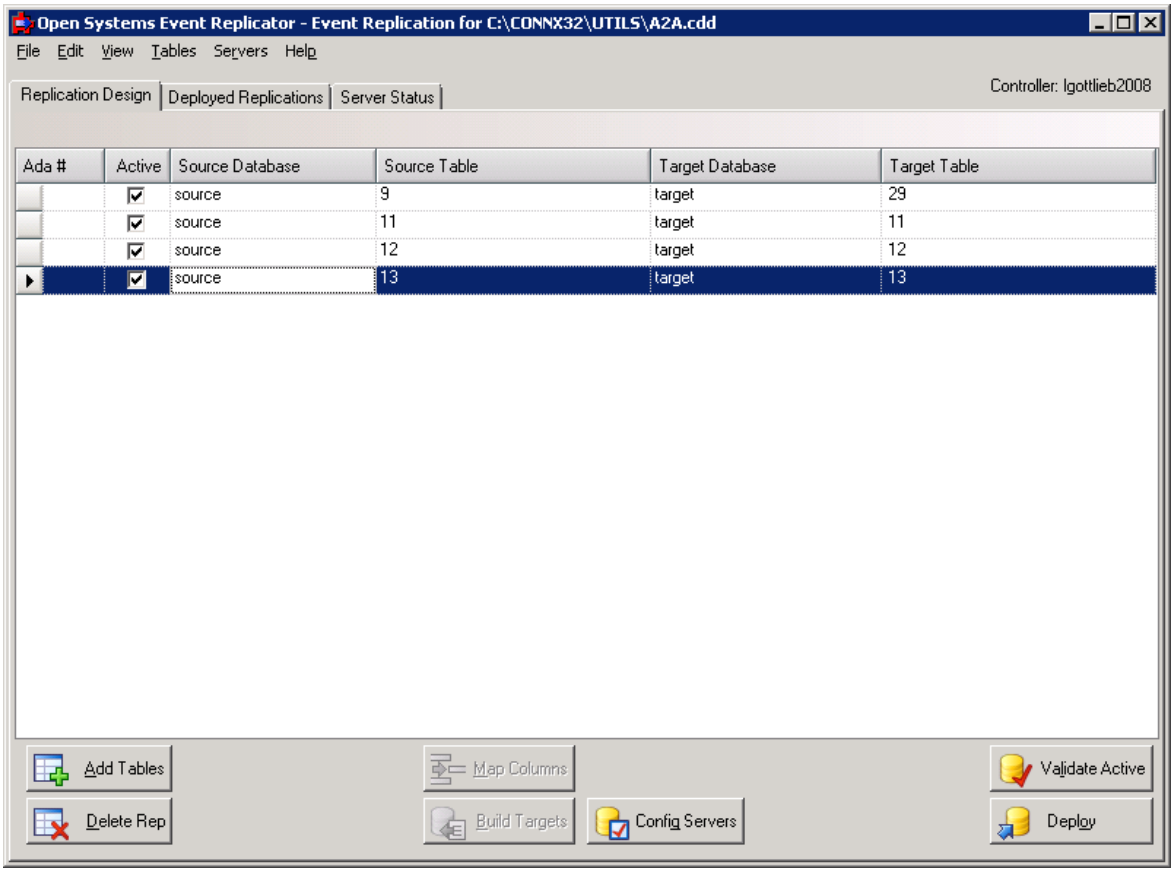
After deploying the event replication you will be able to:

- Check server status

All changes made using the Open System Event Replicator 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.



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.

The **Ada #** column will be blank until the deploy has completed. In an Adabas to Adabas replication environment, the Ada # relates to the ID column from the Adabas adaopr command when using the display=replication option. For more information about the adaopr command and the replication related options, please see the Adabas Utilities documentation on the Software AG Empower website.

Note: Starting with version 12 SP3, Adabas to Adabas replications can no longer be undeployed by unchecking the **Active** checkbox and redploying. To undeploy an individual Adabas to Adabas replication, use the UnDeploy Selected button on the **Deployed Replications** 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

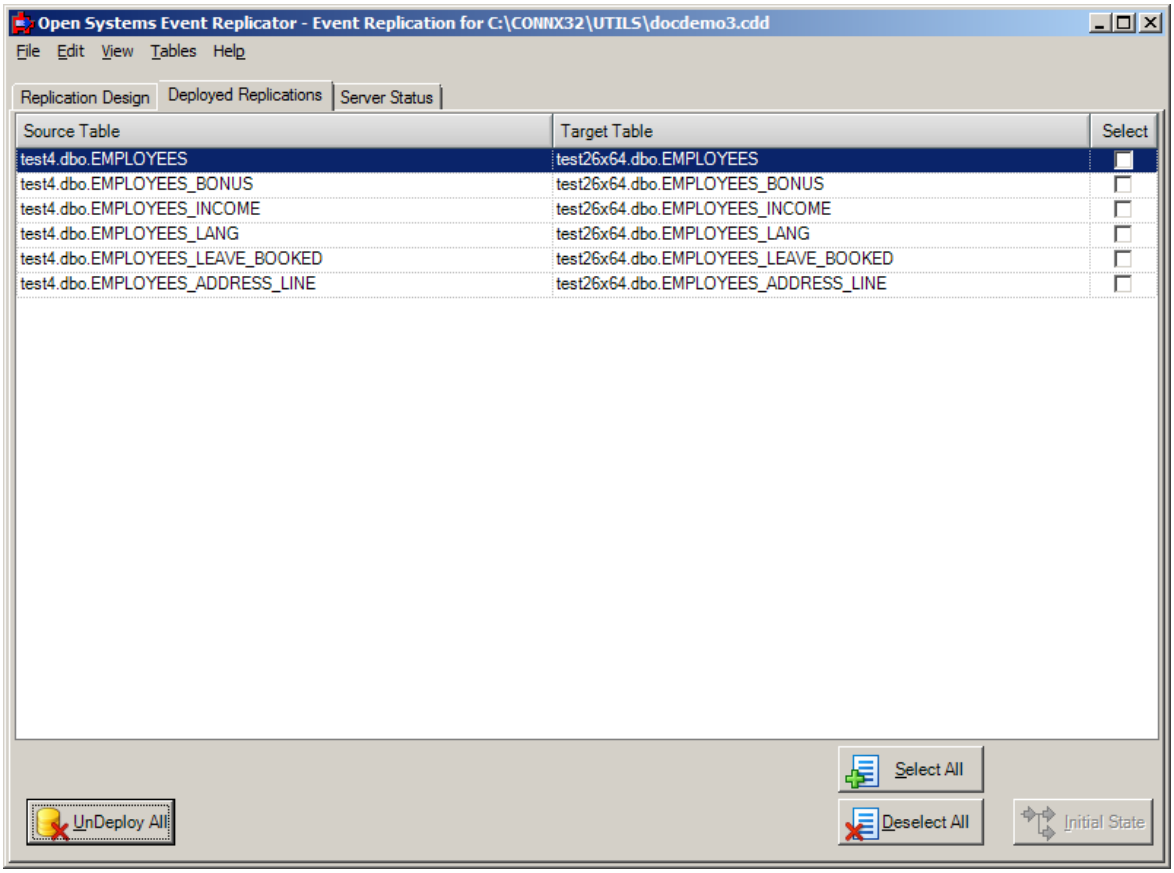
After performing an Initial State you will be able to:

- 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 Open Systems Event Replicator CDD.
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**. The following message appears:

If the initial state request failed, the following message appears:

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:

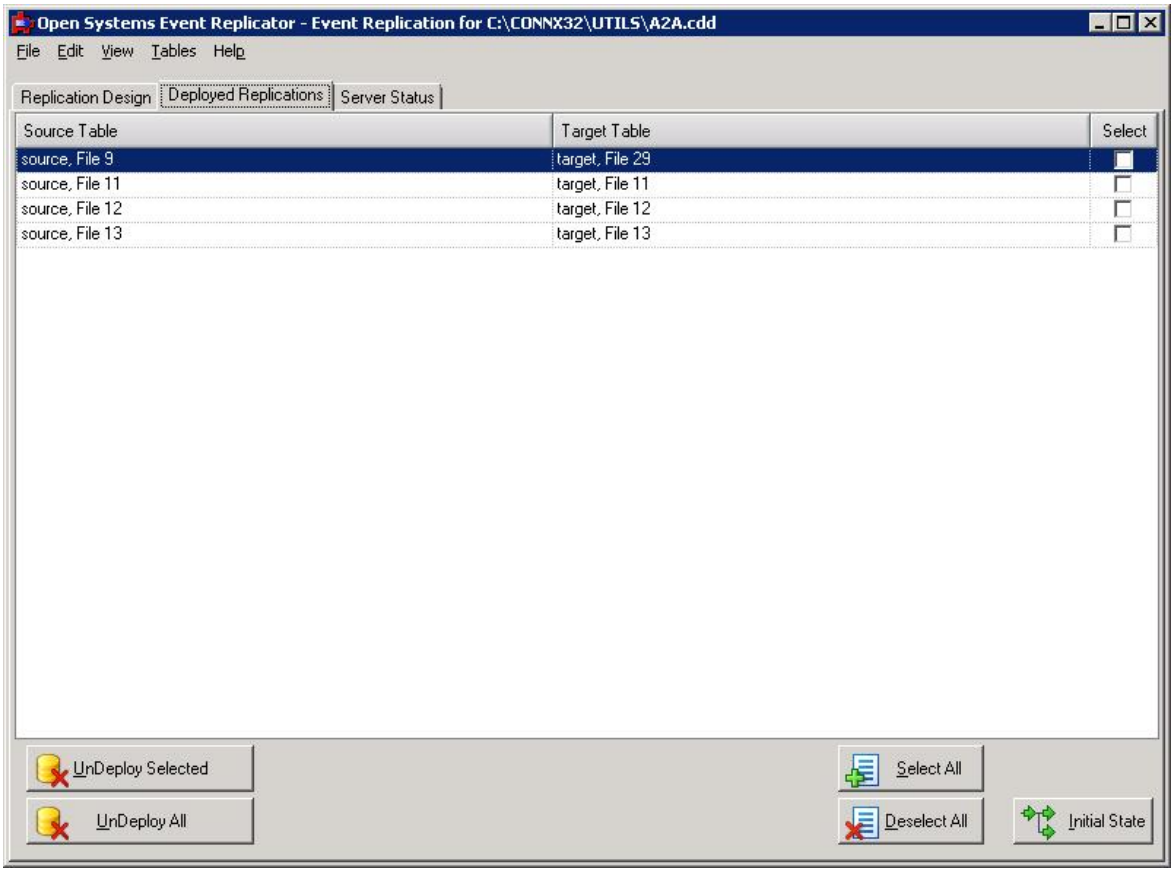
- Do maintenance on the target database
- Replicate to a new target database
- Use a different Controller
- Create a new CDD and use the new CDD for replication

Un-deploy All will un-deploy all active replications at once and put the controller and EP in an idle state.

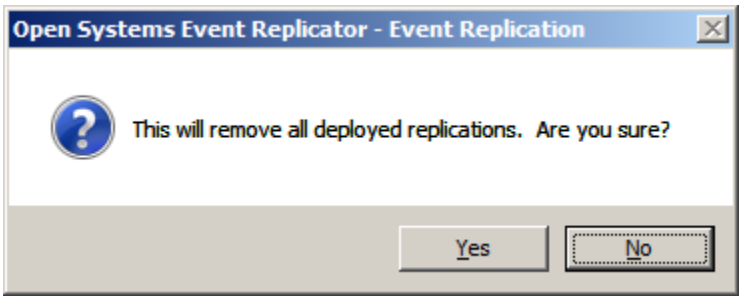
Un-deploy All un-deploys the entire CDD, not just the individual replications.

If you don't want to un-deploy all active replications, use the **UnDeploy Selected** button instead of the **UnDeploy All** button.

1. Open the Open Systems Event Replicator CDD.
2. Click the Deployed Replications tab.

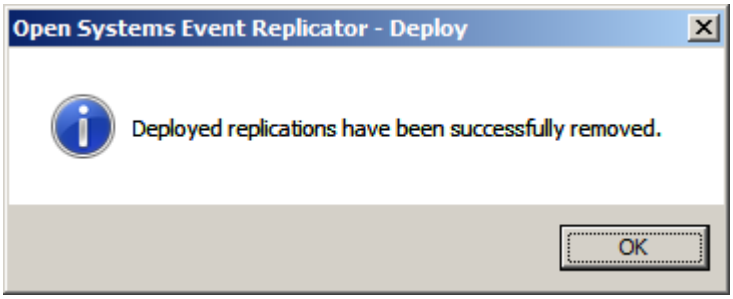


3. Click **Un-Deploy All**. The following message appears:



Warning: Clicking **Yes** will un-deploy **ALL** replications whether you do or don't click **Select.**

The following message appears:


Open Systems Event Replicator - Deploy
Deployed replications have been successfully removed.
OK

Un-Deploying selected Replications

Un-Deploying selected replications includes:

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

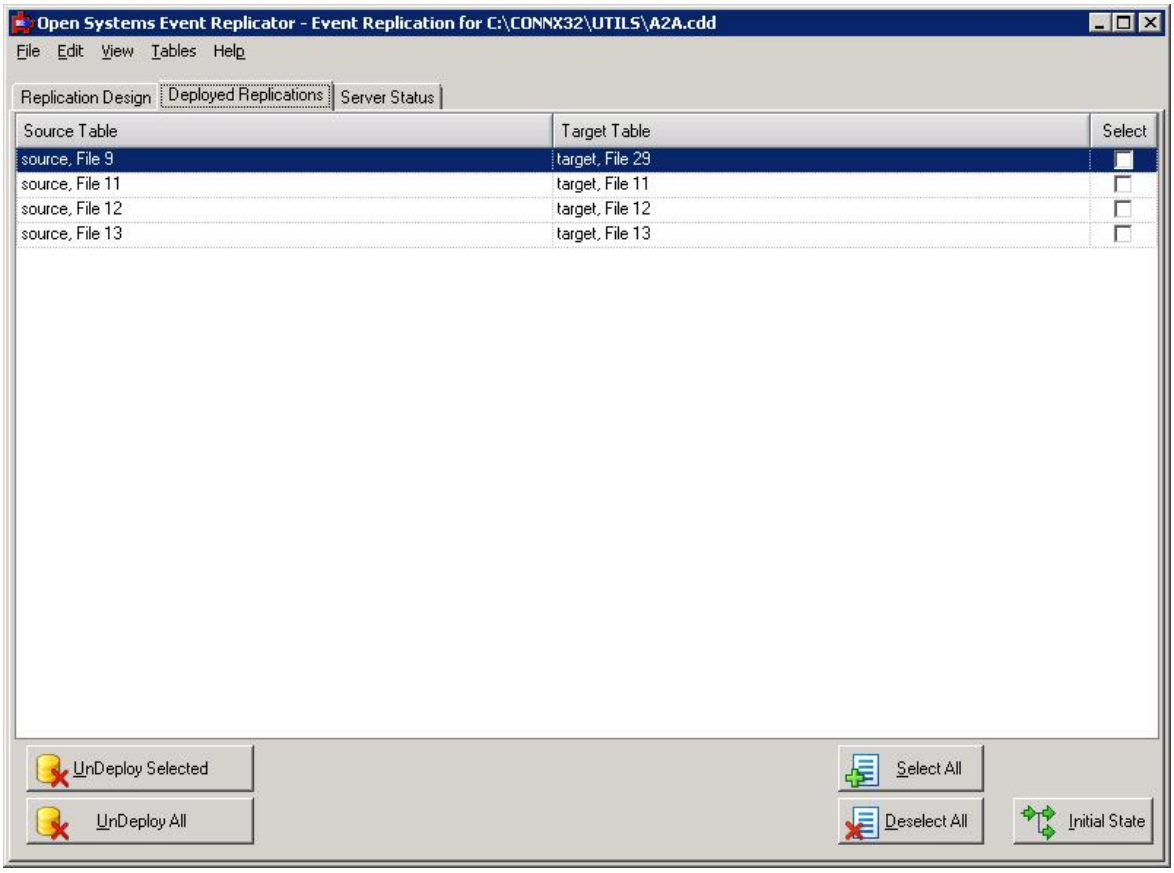
Un-Deploy selected event replications if:

- The target database will be out of service for an extended period
- You no longer want to use some but not all of the current replications

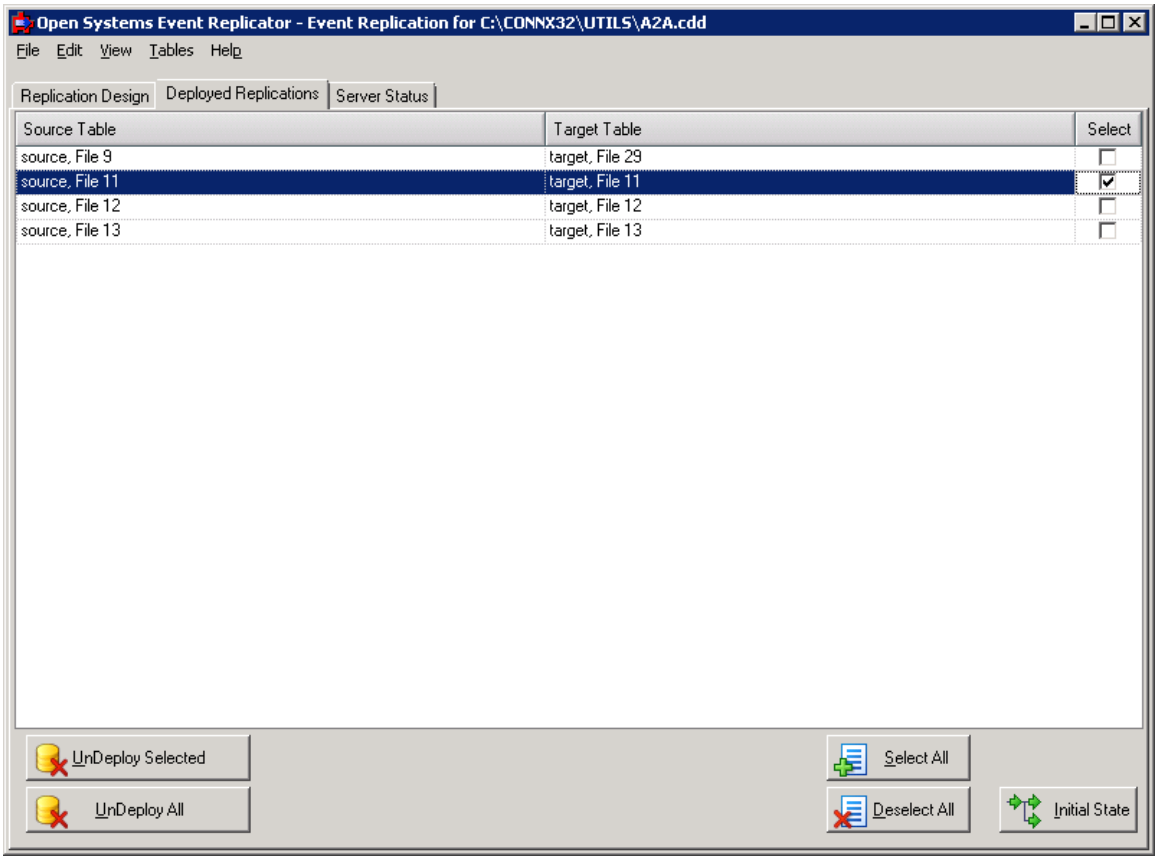
Un-deploy Selected will un-deploy only those replications with the **Select** checkbox checked.

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**.

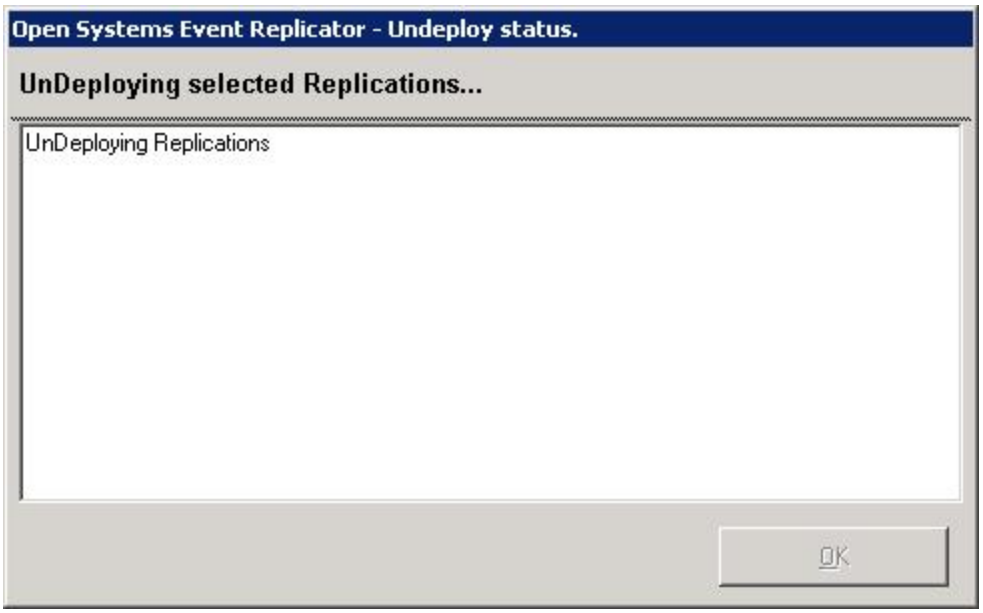
1. Open the Open Systems Event Replicator CDD.
2. Click the Deployed Replications tab.



3. Select a replication to Undeploy - in this case we will select file 11.



4. Click **Un-Deploy Selected**. The following message appears:



A message will be sent to the EP instructing it to remove the replication(s) specified from the list of deployed replications.

Note: If you use **UnDeploy Selected** to remove all the replications, the controller and EP will remain active with nothing to do. If yo wish them to become inactive, use the UnDeploy 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.

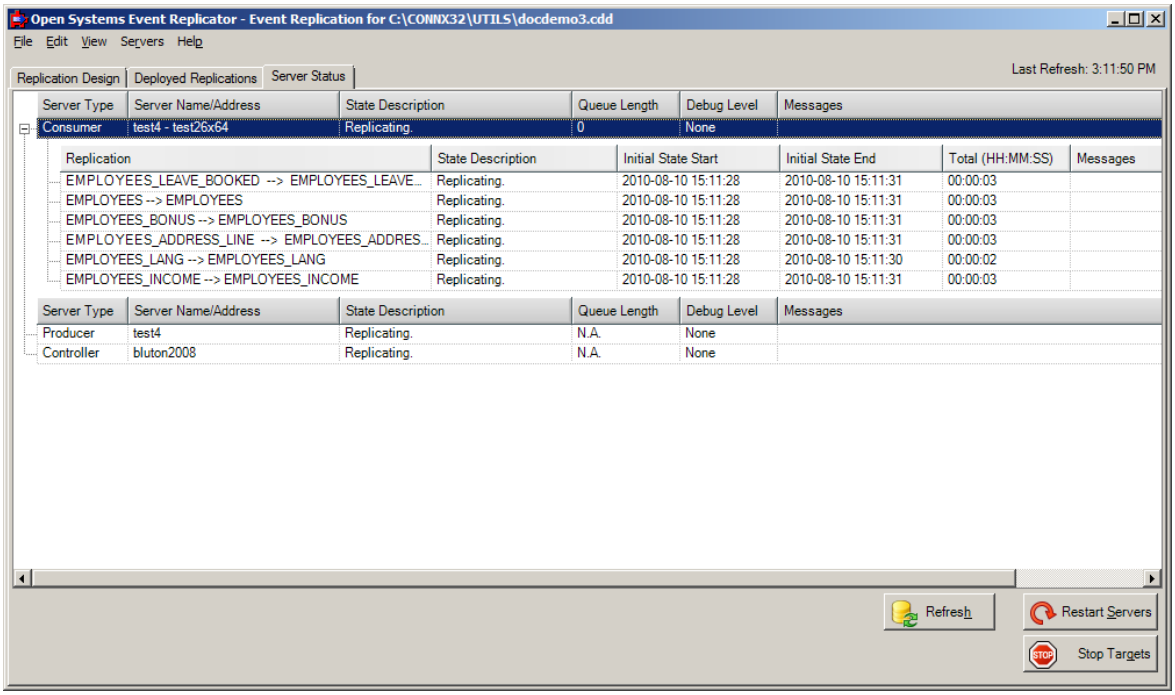
Stop the target servers before you do any target database or server maintenance.

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

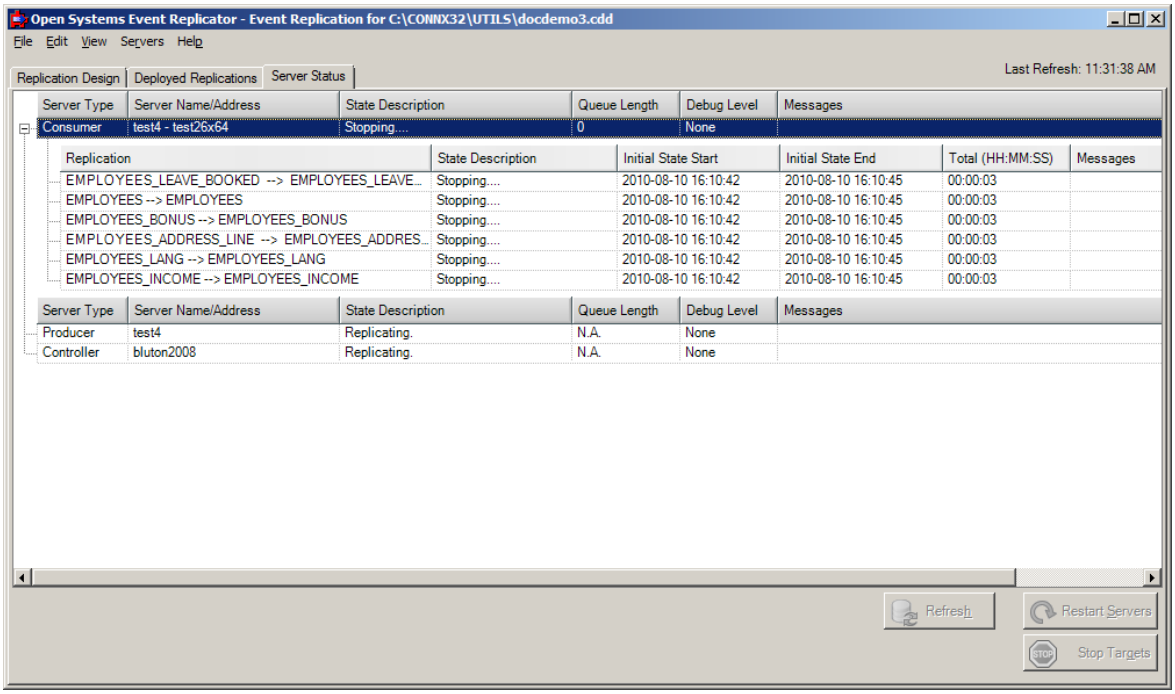
Note: **Stop Targets** does not stop source or replication servers.

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

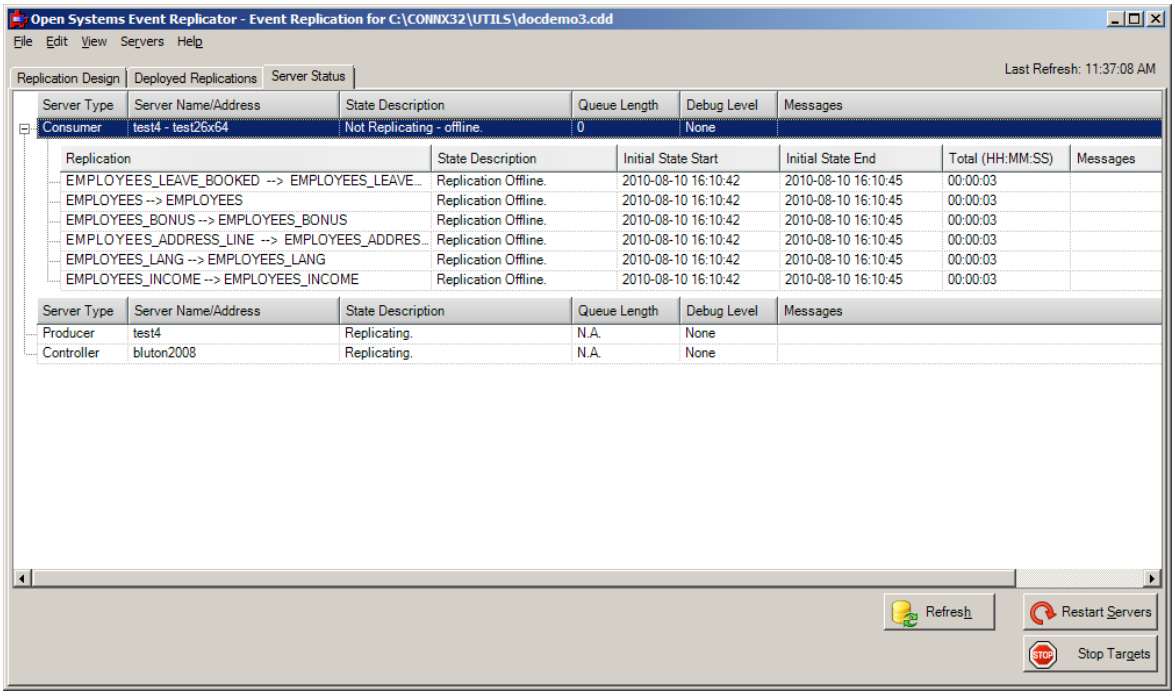
1. Open the Open Systems Event Replicator CDD.
2. Click the **Server Status** tab.



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:


Open Systems Event Replicator - Event Replication for C:\CONNX32\UTILS\docdemo3.cdd
File Edit View Servers Help
Replication Design
Deployed Replications
Server Status
Last Refresh: 11:37:08 AM
Server Type
Server Name/Address
State Description
Queue Length
Debug Level
Messages
Consumer
test4 - test26x64
Not Replicating - offline.
0
None
Replication
State Description
Initial State Start
Initial State End
Total (HH:MM:SS)
Messages
EMPLOYEES_LEAVE_BOOKED --> EMPLOYEES_LEAVE...
Replication Offline.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES --> EMPLOYEES
EMPLOYEES_BONUS --> EMPLOYEES_BONUS
EMPLOYEES_ADDRESS_LINE --> EMPLOYEES_ADDRES...
EMPLOYEES_LANG --> EMPLOYEES_LANG
EMPLOYEES_INCOME --> EMPLOYEES_INCOME
Producer
test4
Replicating.
N.A.
None
Controller
bluton2008
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

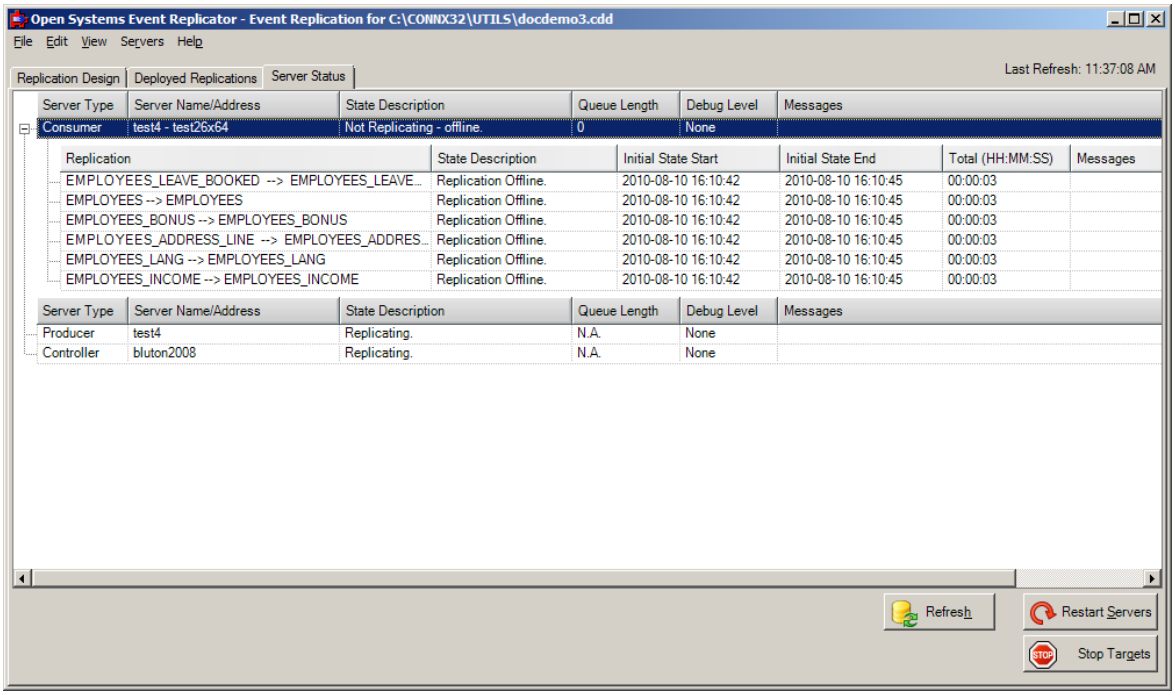
If the Event Replicator has been stopped, restart the target servers to enable replication.

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.

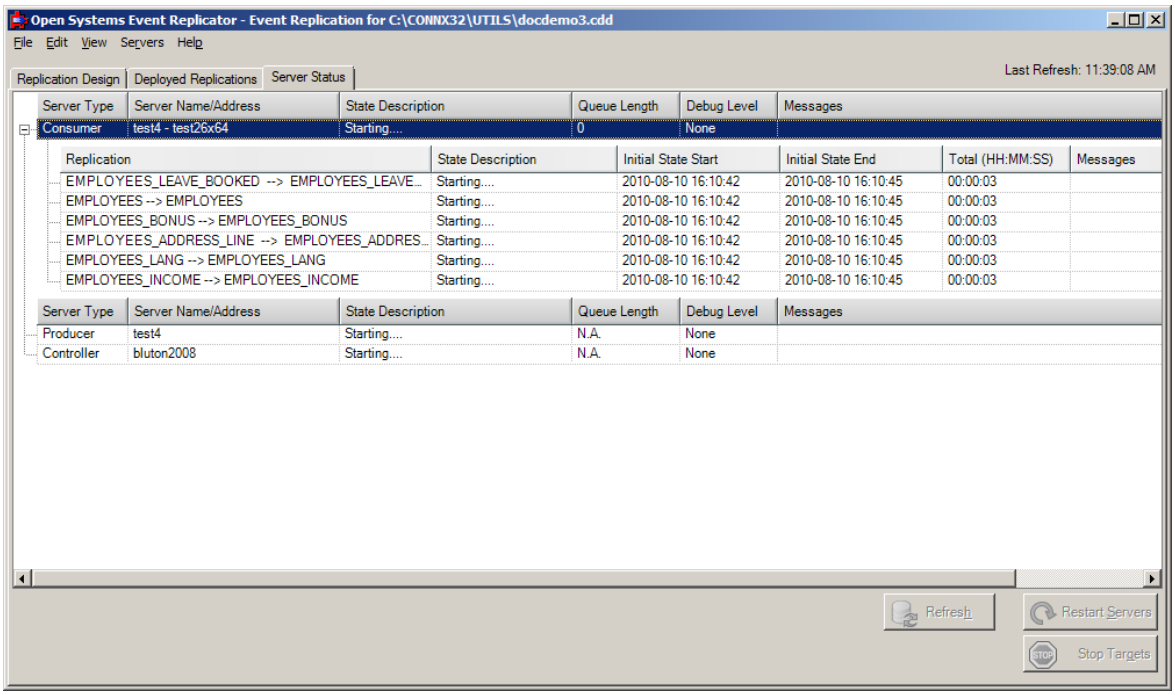
All replications that have been processed while the servers have been stopped remain in the queue in the order received. After the servers have been started, the Event Replicator processes the replications in the order they appear in the queue.

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

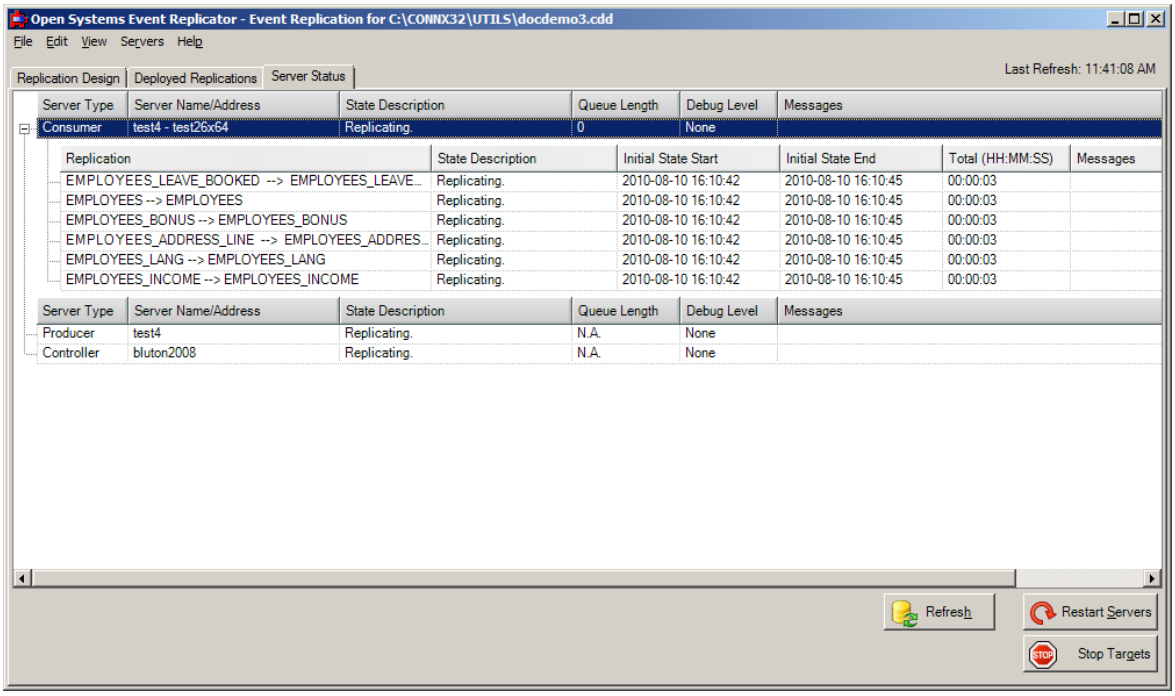
1. Open the Open Systems Event Replicator CDD.
2. Click the **Server Status** tab.



3. Click **Restart Servers**.



When the servers have been started, the following appears:


Open Systems Event Replicator - Event Replication for C:\CONNX32\UTILS\docdemo3.cdd
File Edit View Servers Help
Replication Design
Deployed Replications
Server Status
Last Refresh: 11:41:08 AM
Server Type
Server Name/Address
State Description
Queue Length
Debug Level
Messages
Consumer
test4 - test26x64
Replicating.
0
None
Replication
State Description
Initial State Start
Initial State End
Total (HH:MM:SS)
Messages
EMPLOYEES_LEAVE_BOOKED --> EMPLOYEES_LEAVE...
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES --> EMPLOYEES
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES_BONUS --> EMPLOYEES_BONUS
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES_ADDRESS_LINE --> EMPLOYEES_ADDRES...
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES_LANG --> EMPLOYEES_LANG
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
EMPLOYEES_INCOME --> EMPLOYEES_INCOME
Replicating.
2010-08-10 16:10:42
2010-08-10 16:10:45
00:00:03
Server Type
Server Name/Address
State Description
Queue Length
Debug Level
Messages
Producer
test4
Replicating.
N.A.
None
Controller
bluton2008
Replicating.
N.A.
None
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.

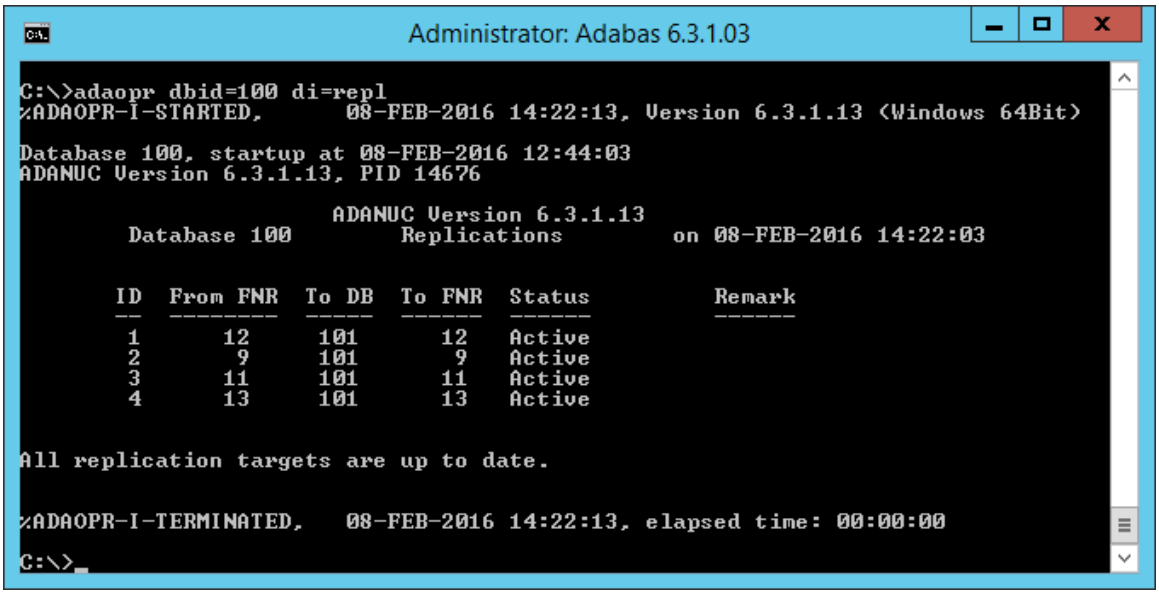
Note: The Message Queue should be started before the Event Server and the Adabas nucleus that is involved in replication. If the Message Queue is not started first, error messages will appear in the log files indicating that either the EP or the Controller could not contact the Message Queue. These messages will continue to appear until the Message Queue is started. When stopping services, the Message Queue should be stopped last.

Using adaopr command line Utility

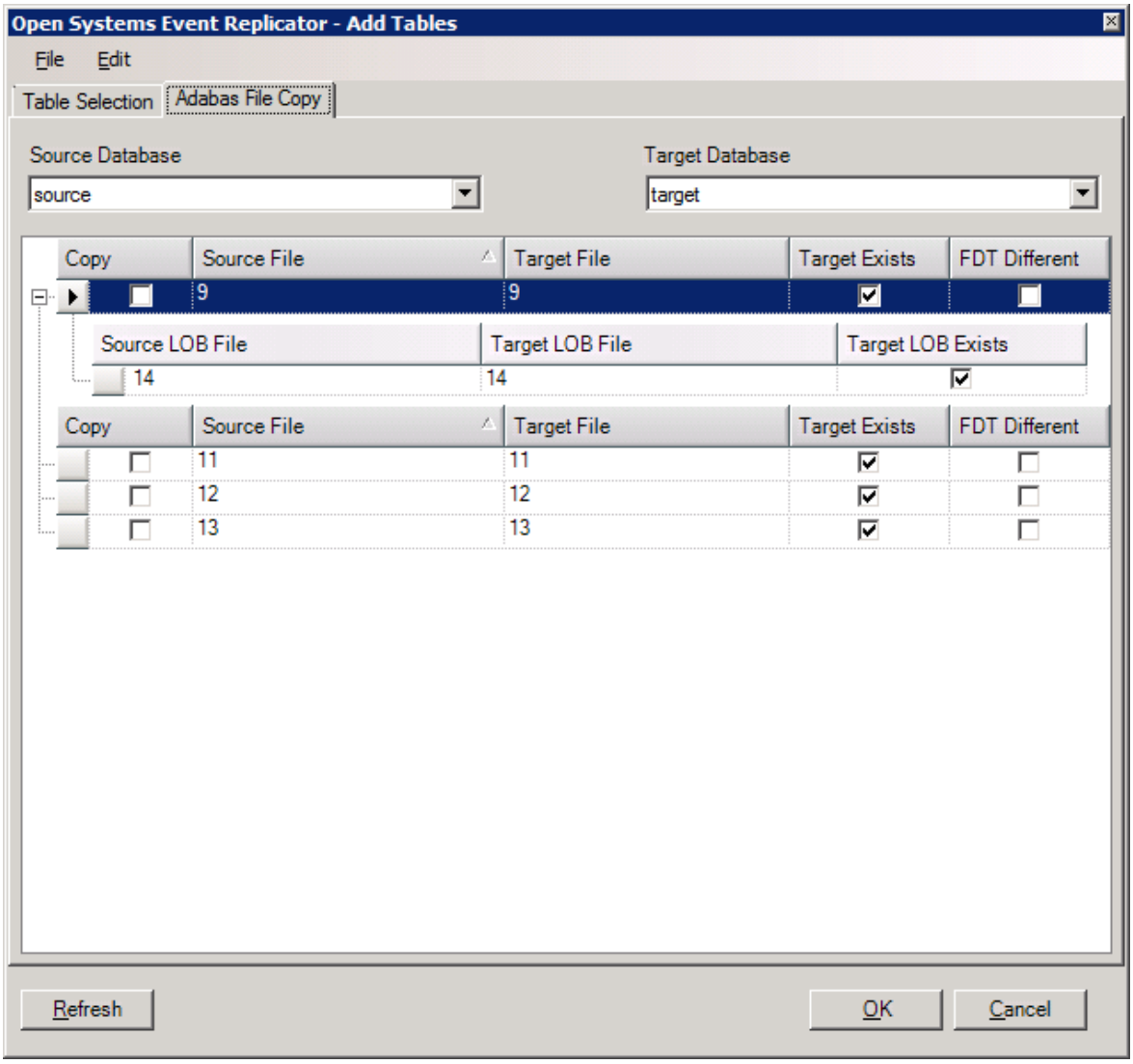
The Open Systems Event Replication Administration tool can recognize replications that have been added, removed or had status changed via the adaopr command line utility.

Before using the Event Replication Administrator with a nucleus that has had replications previously added to it via the adaopr utility, the Event Server must first be Synchronized with the EP running within that nucleus.

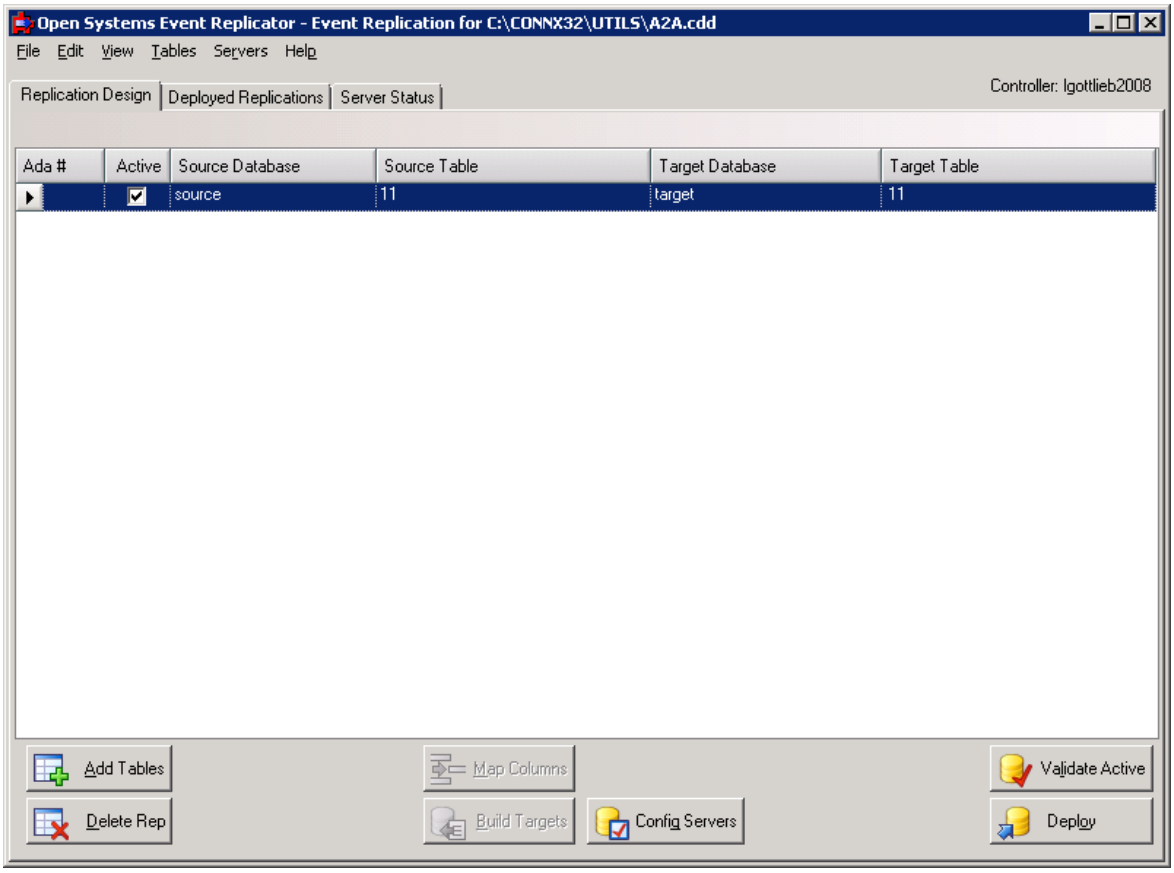
1. Select a nucleus that contains replications added via the adaopr utility



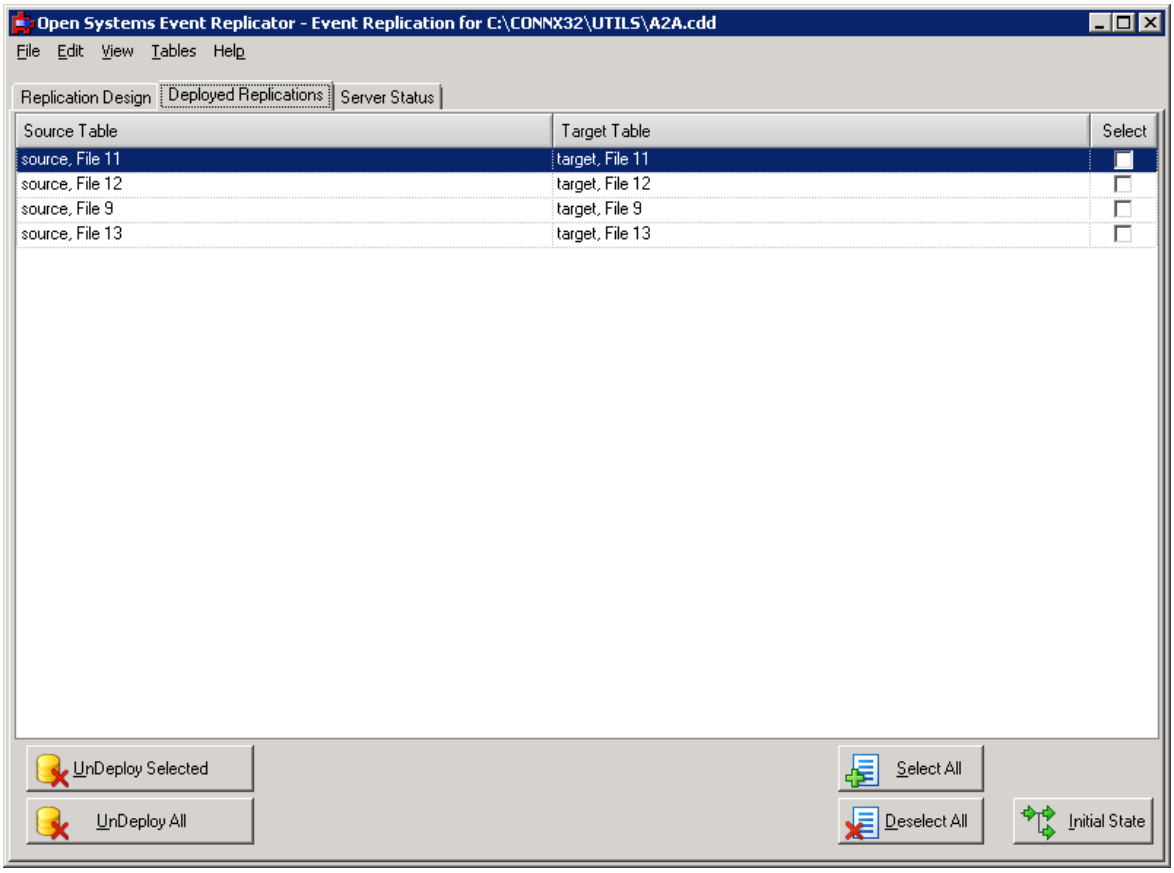
2. Open the Open Systems Event Replicator CDD that contains database connections for the source and target databases. If this is a brand new CDD, it will need to be Enabled for Replication.
3. On the Add Tables dialog, either select one of the files that is currently being replicated or choose a new file to replicate.

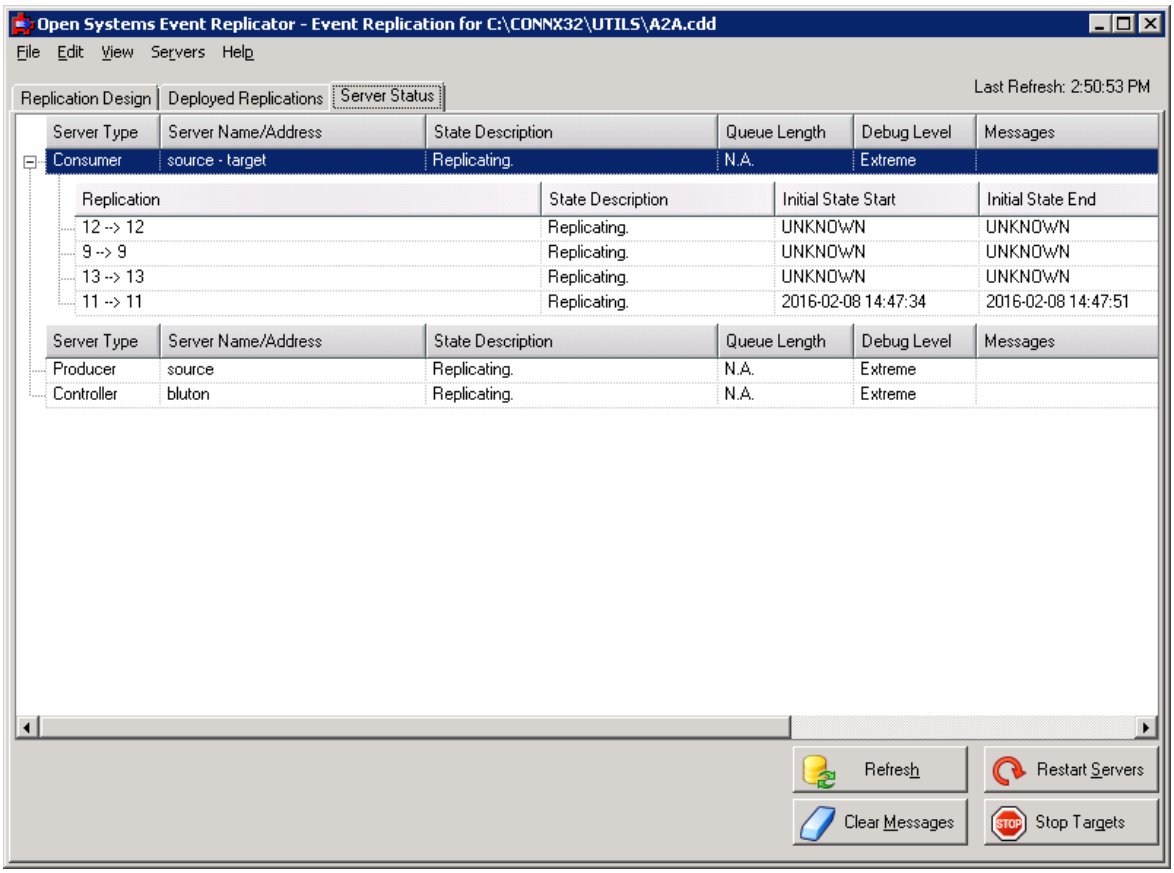


3. Click **OK** and then click Deploy from the Replication Design screen:



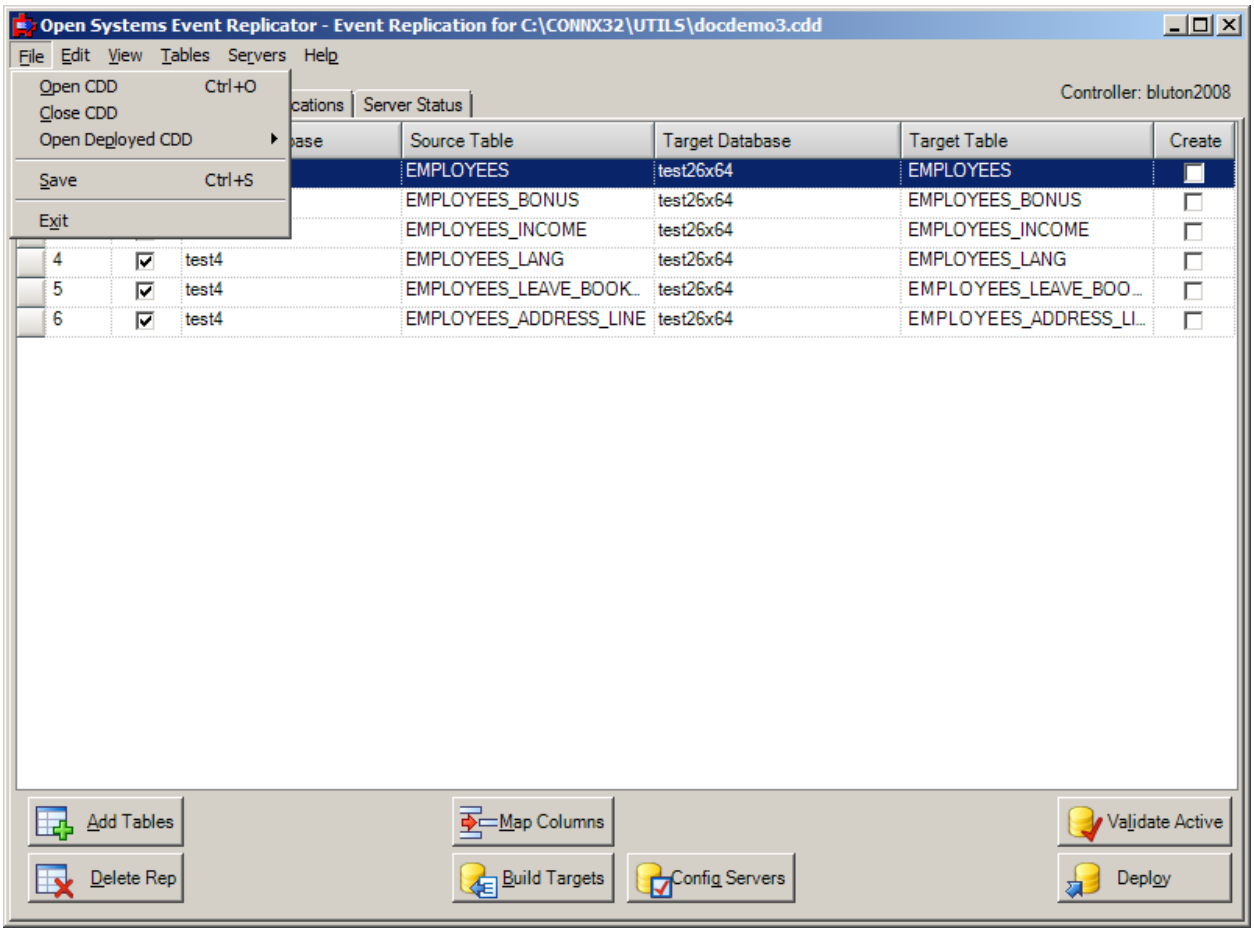
During the deploy process, the EP will become active and will send a current list of replications back to the controller. This list will in turn be sent to the Administrator which will display them in the Active Replications tab as well as the Server Status Tab:


Open Systems Event Replicator - Event Replication for C:\CONNX32\UTILS\A2A.cdd
File Edit View Tables Help
Replication Design
Deployed Replications
Server Status
Source Table
Target Table
Select
source, File 11
target, File 11
source, File 12
target, File 12
source, File 9
target, File 9
source, File 13
target, File 13
UnDeploy Selected
UnDeploy All
Select All
Deselect All
Initial State


Open Systems Event Replicator - Event Replication for C:\CONNX32\UTILS\A2A.cdd
File Edit View Servers Help
Replication Design
Deployed Replications
Server Status
Last Refresh: 2:50:53 PM
Server Type
Server Name/Address
State Description
Queue Length
Debug Level
Messages
Consumer
source - target
Replicating.
N.A.
Extreme
Replication
State Description
Initial State Start
Initial State End
12 --> 12
Replicating.
UNKNOWN
UNKNOWN
9 --> 9
Replicating.
UNKNOWN
UNKNOWN
13 --> 13
Replicating.
UNKNOWN
UNKNOWN
11 --> 11
Replicating.
2016-02-08 14:47:34
2016-02-08 14:47:51
Server Type
Server Name/Address
State Description
Queue Length
Debug Level
Messages
Producer
source
Replicating.
N.A.
Extreme
Controller
bluton
Replicating.
N.A.
Extreme
Refresh
Restart Servers
Clear Messages
Stop Targets

File Menu

Applies to all tabs.



The File Menu has four commands:

- Open CDD (or Ctrl+O)
- Close CDD
- Open Deployed CDD

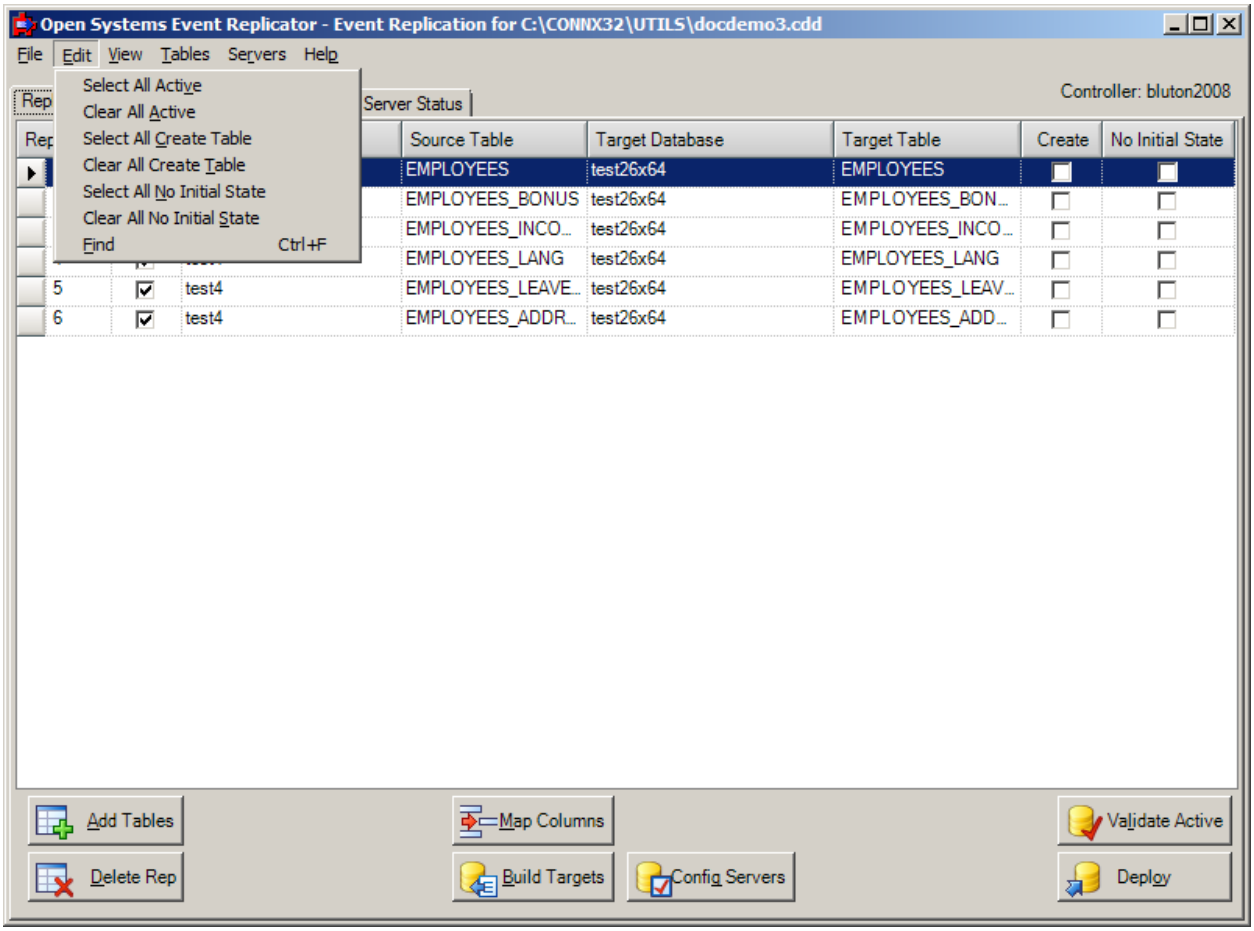
  Contains a list of the deployed CDDs
- Save (or Ctrl+S)

  Saves any replication changes generated since the last save or when replication began.
- 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.

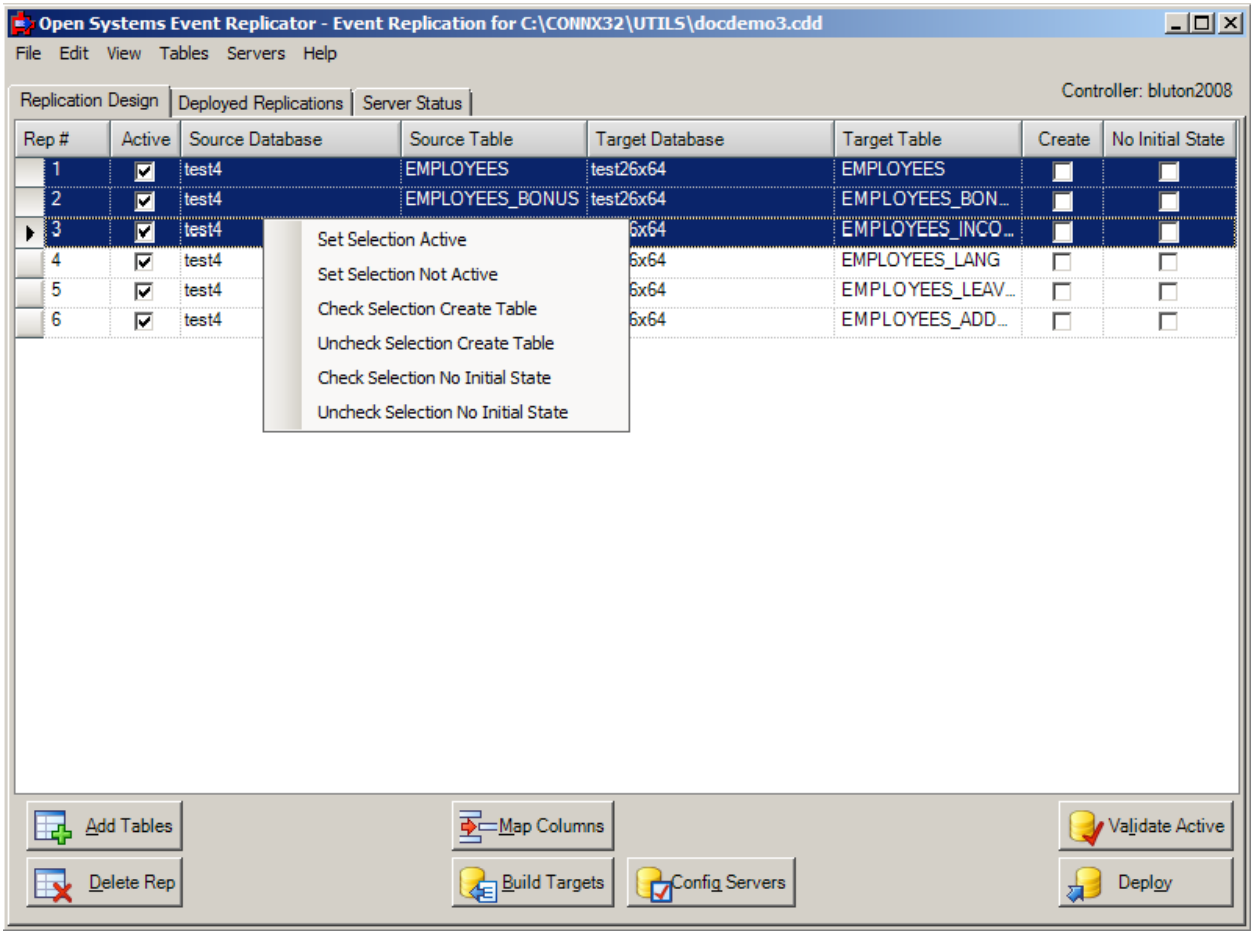
- **Clear All No Initial State**

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

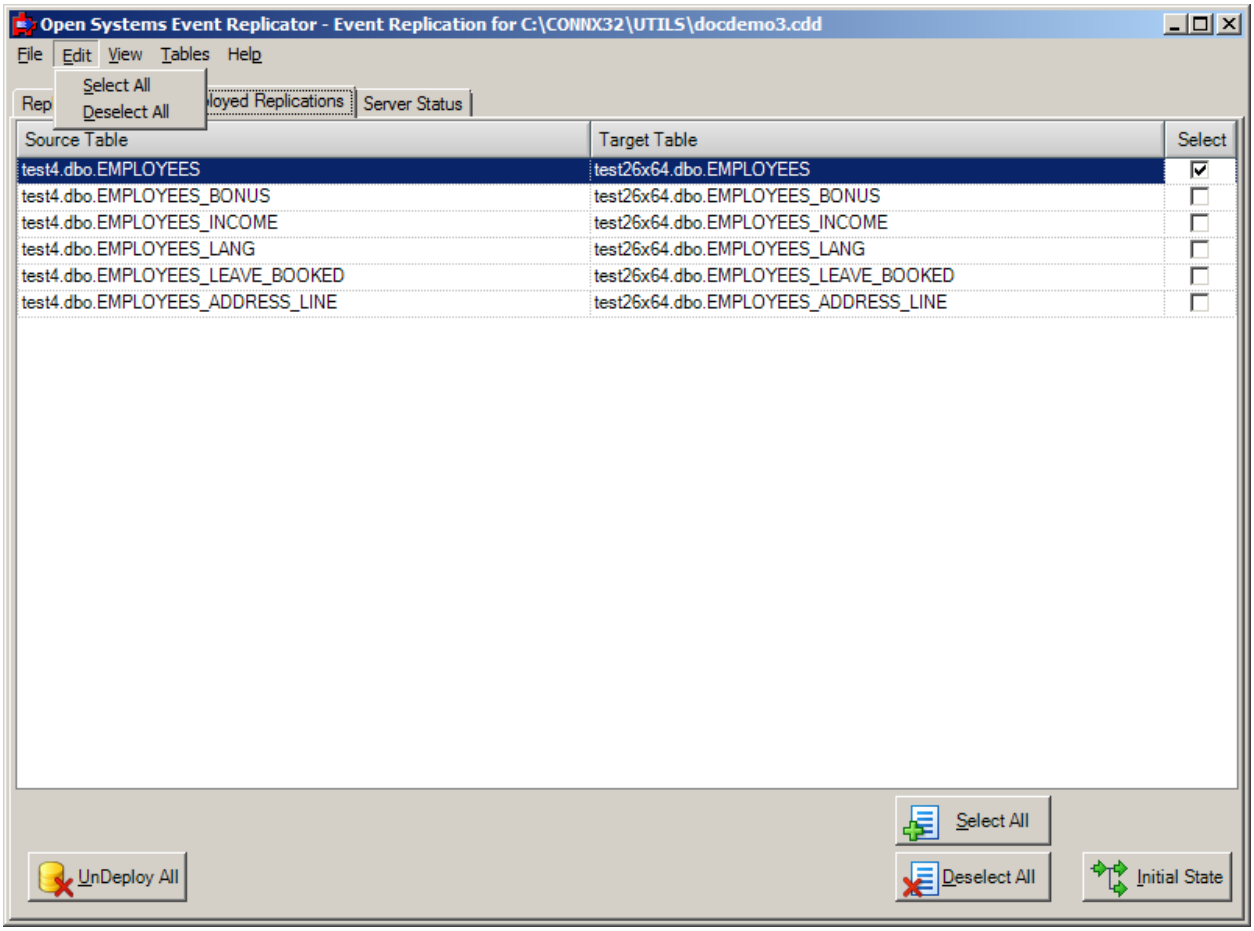
- **Find**

  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:

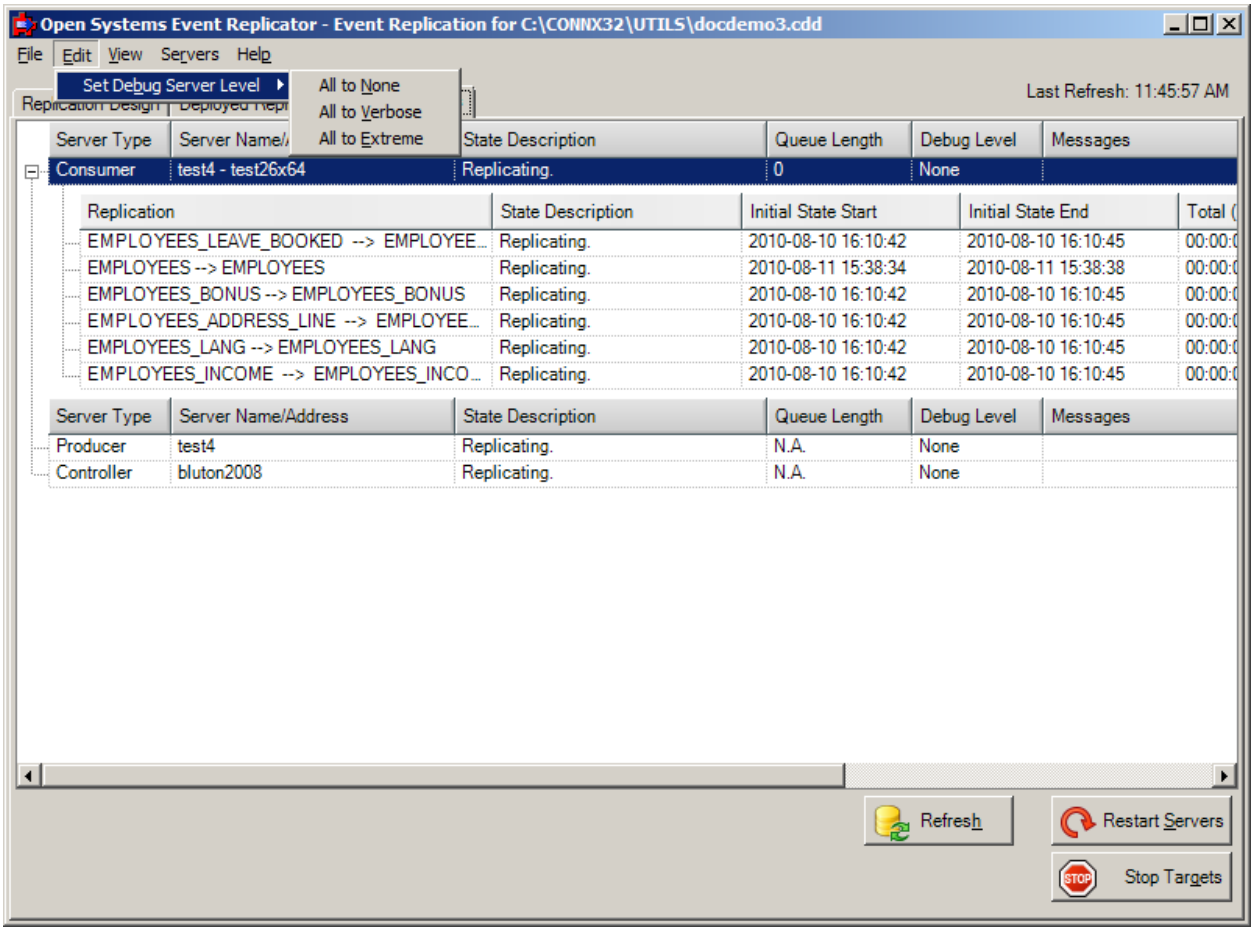


The Edit Menu has two commands:

- **Select All/Deselect All**

  Check or clear the Select checkbox 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. Three are three values for debug level:

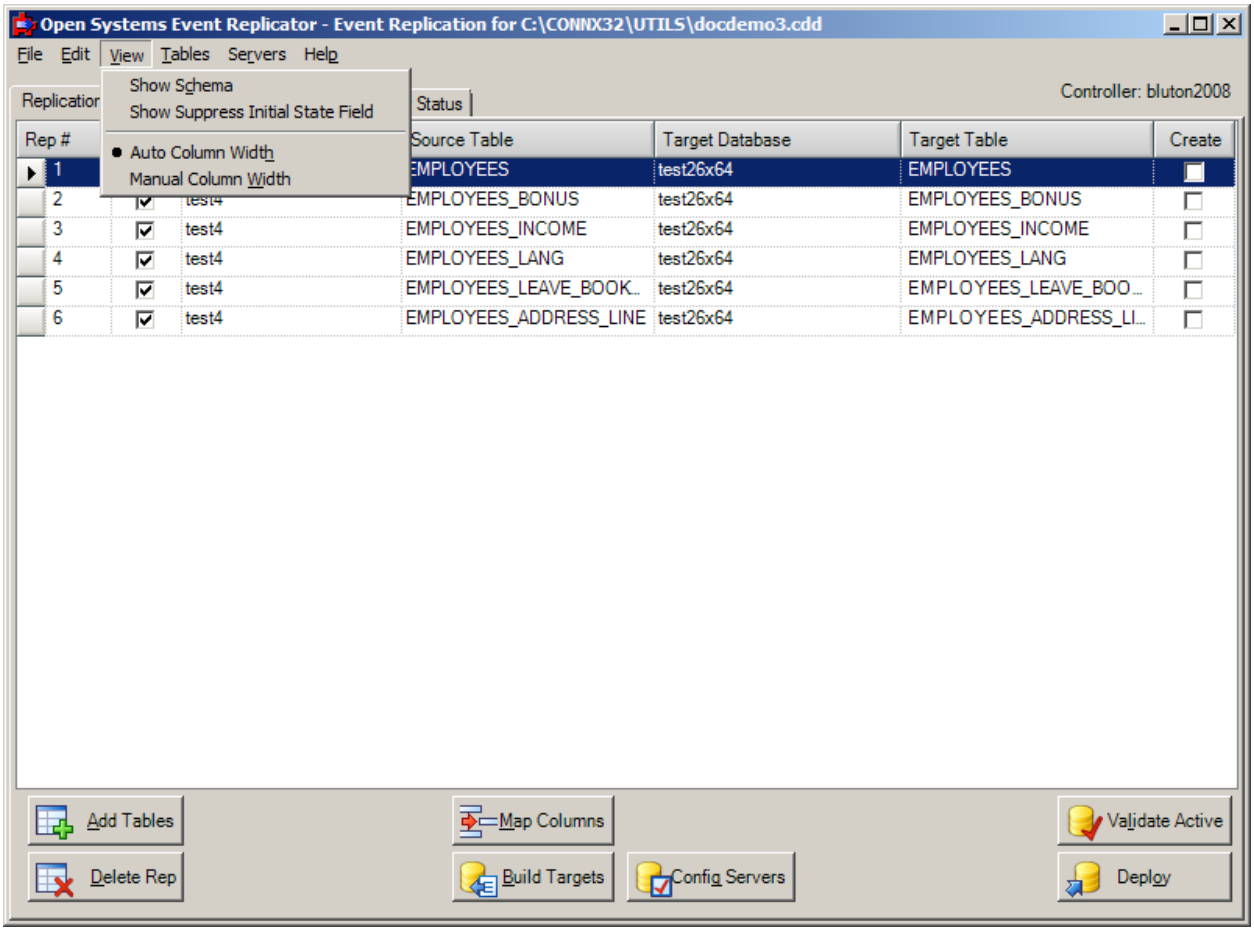
  - **None:** Minimal messages are written to the log. Errors are always written.
  - **Verbose:** Diagnostic information is written when errors occur along with transaction information.
  - **Extreme:** Complete diagnostics including buffer and transaction dumps are written to the log.
    Note: This 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

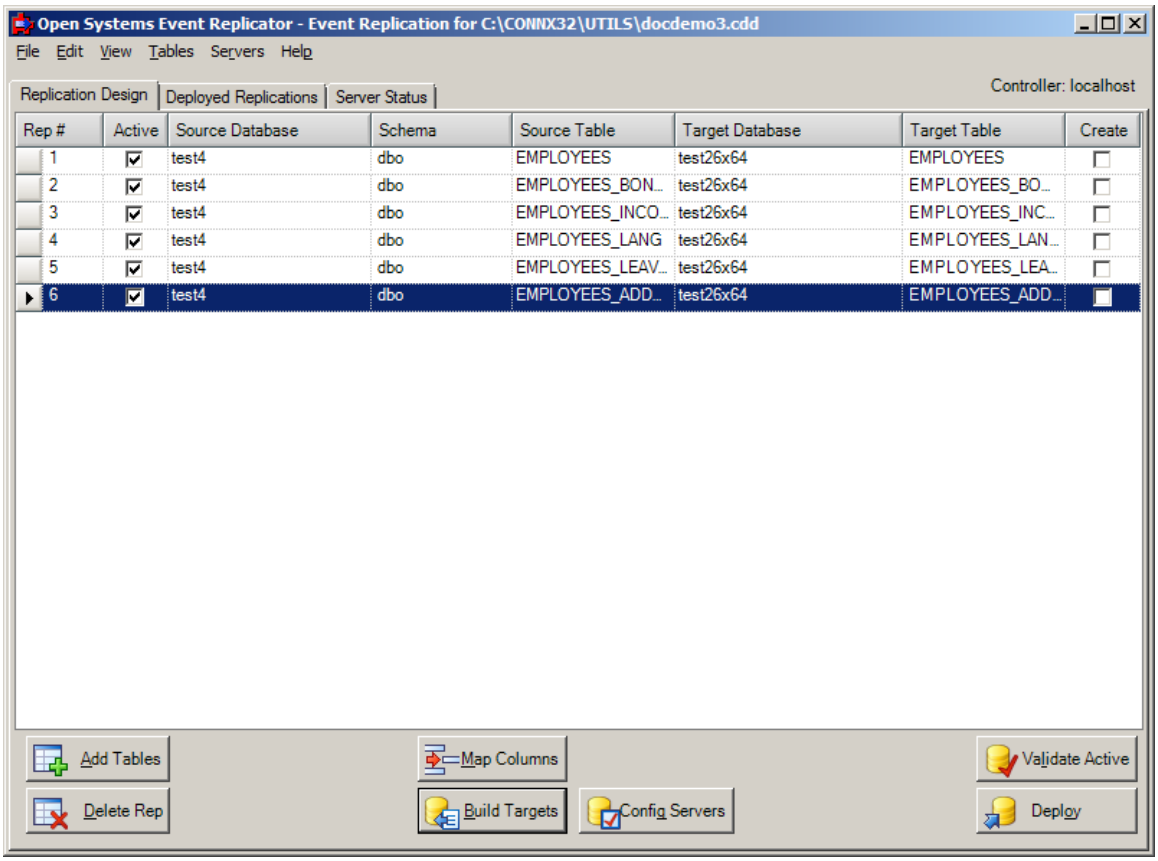
Replication Design tab:



The View Menu has four commands:

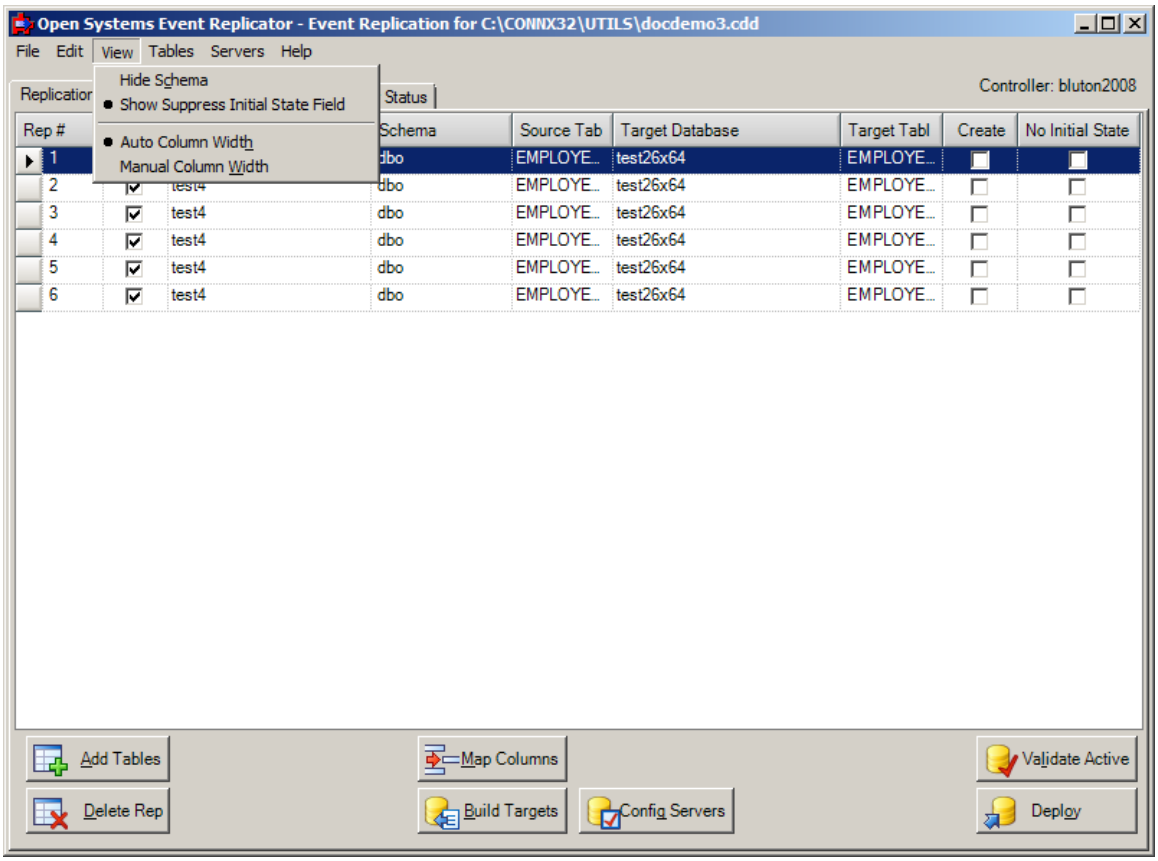
- **Show/Hide Schema**

  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.



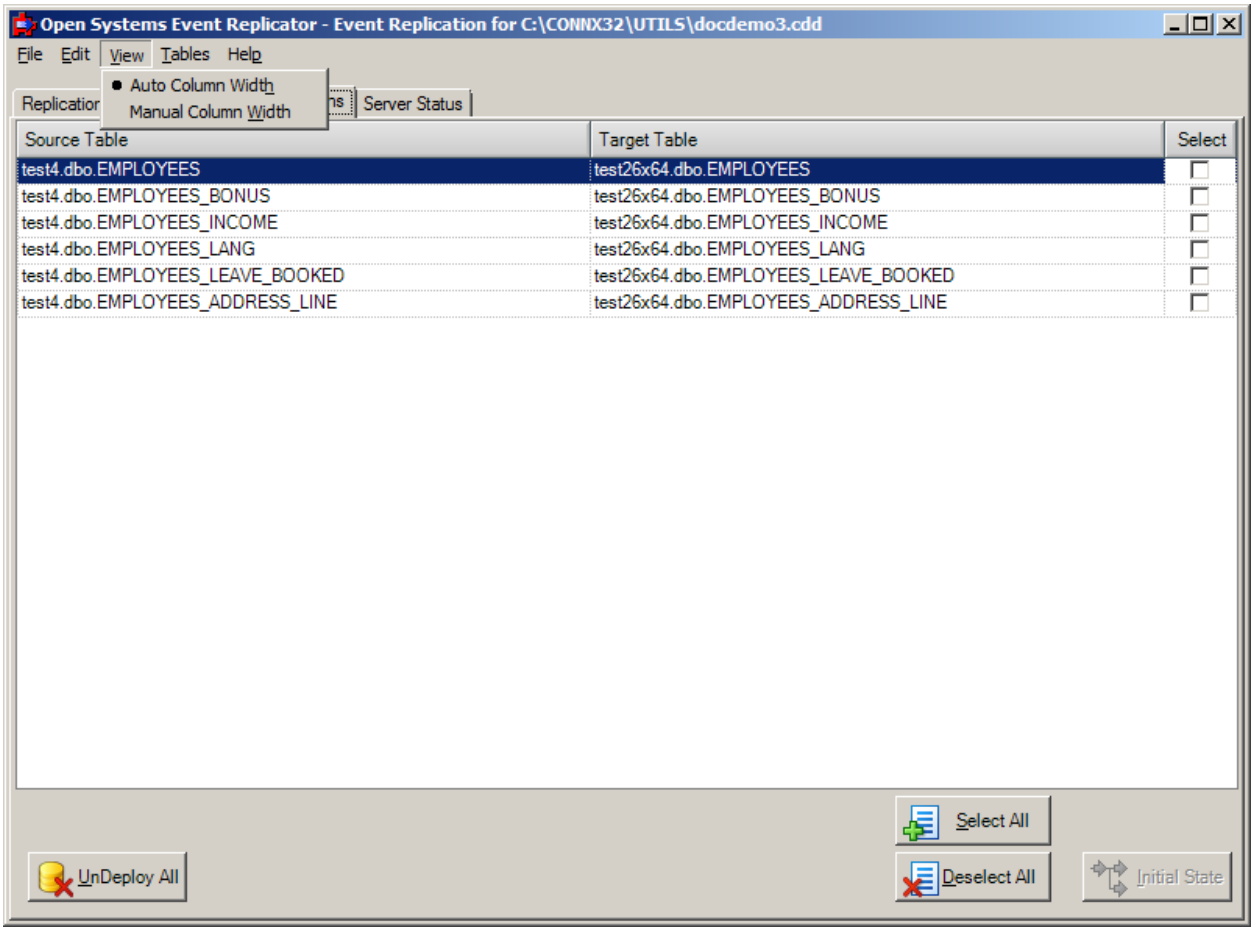
- **Auto Column Width**

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

- **Manual Column Width**

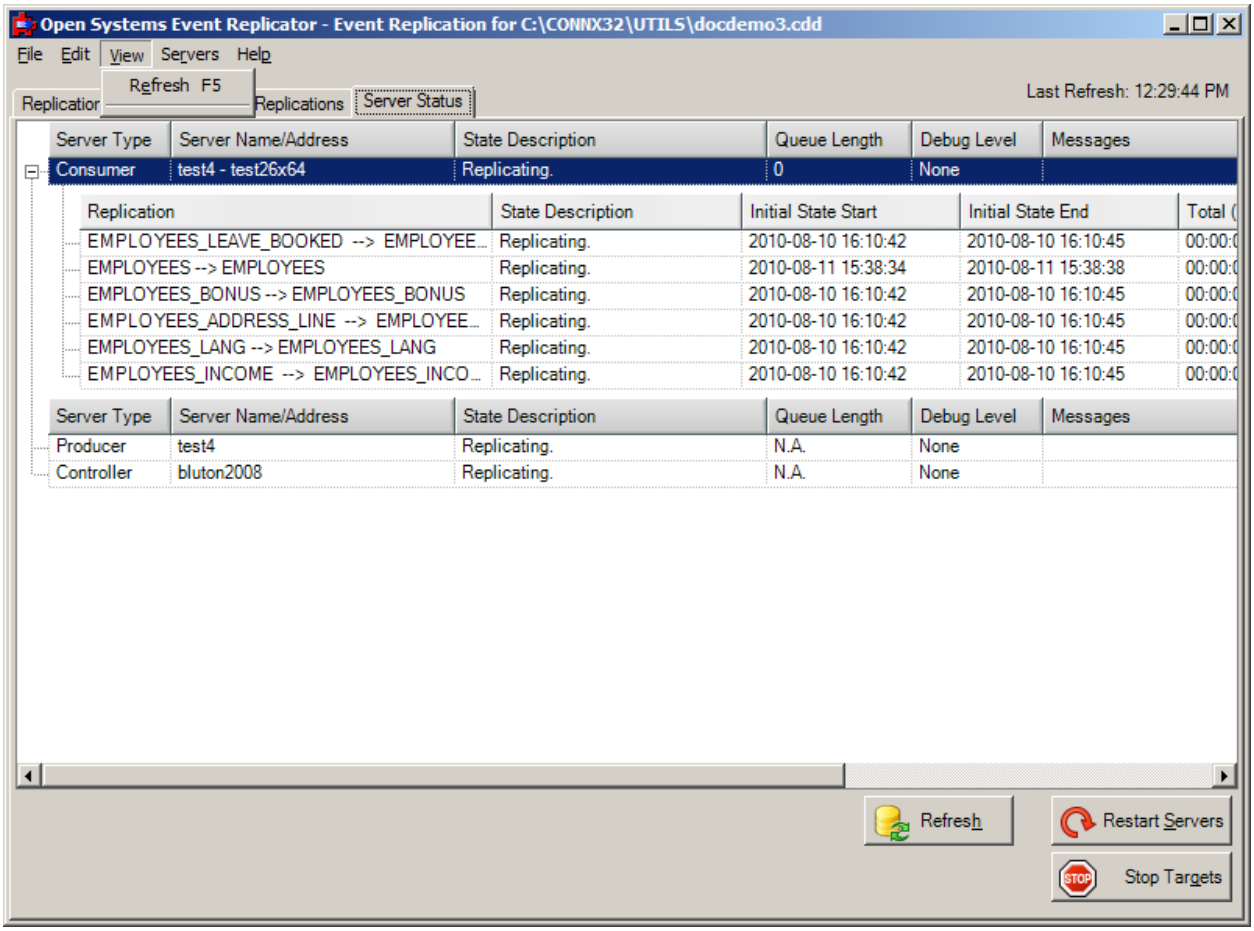
  When selected the width of the columns will not be updated when the size of the application window changes. This option will be automatically selected whenever a column width is manually changed. If you wish to reset the width of the columns to fill the window after manually sizing them, select Auto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

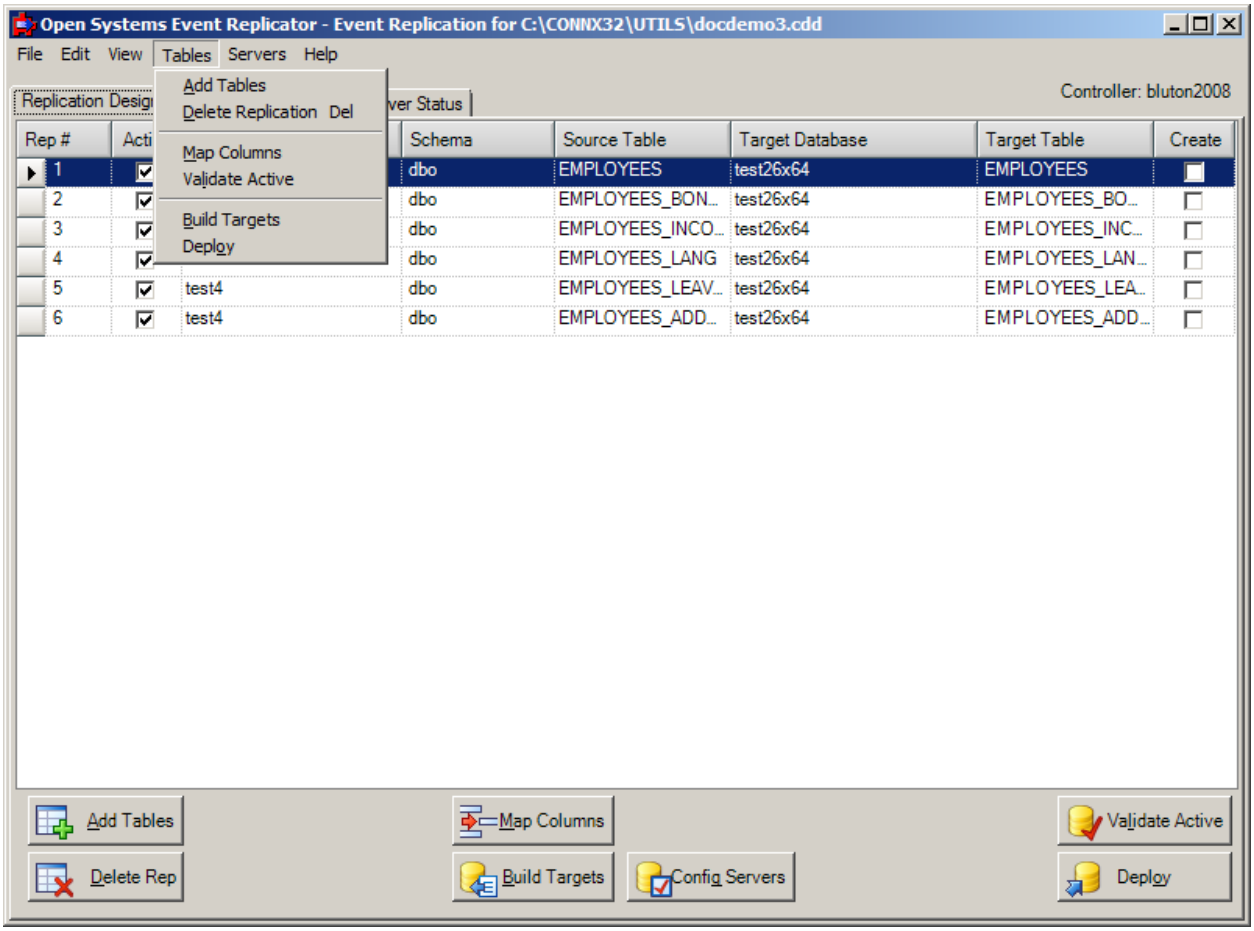
- **Refresh**

  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

Delete Replication

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

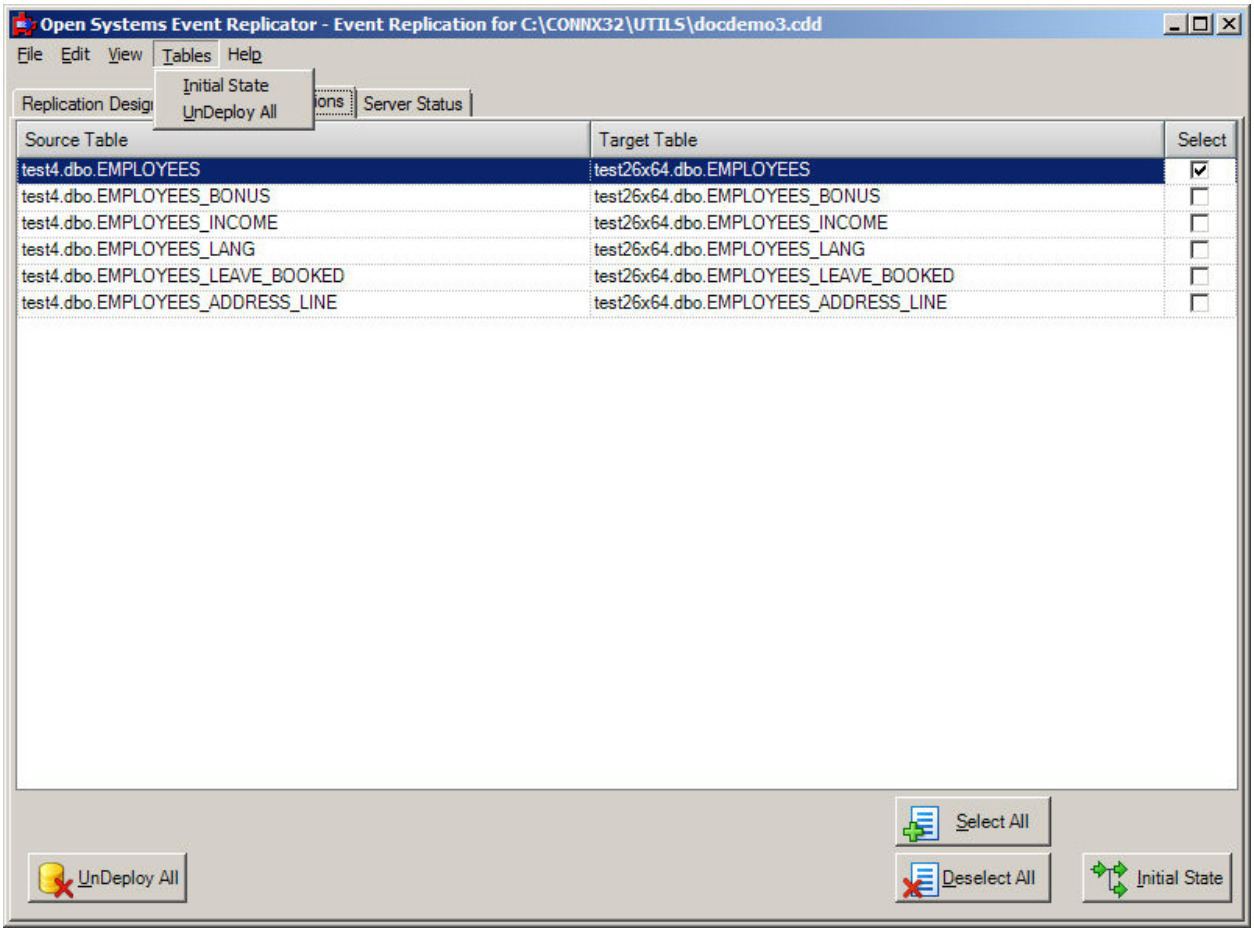
Validate Active

Build Targets

Deploy

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 two commands:

- **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.

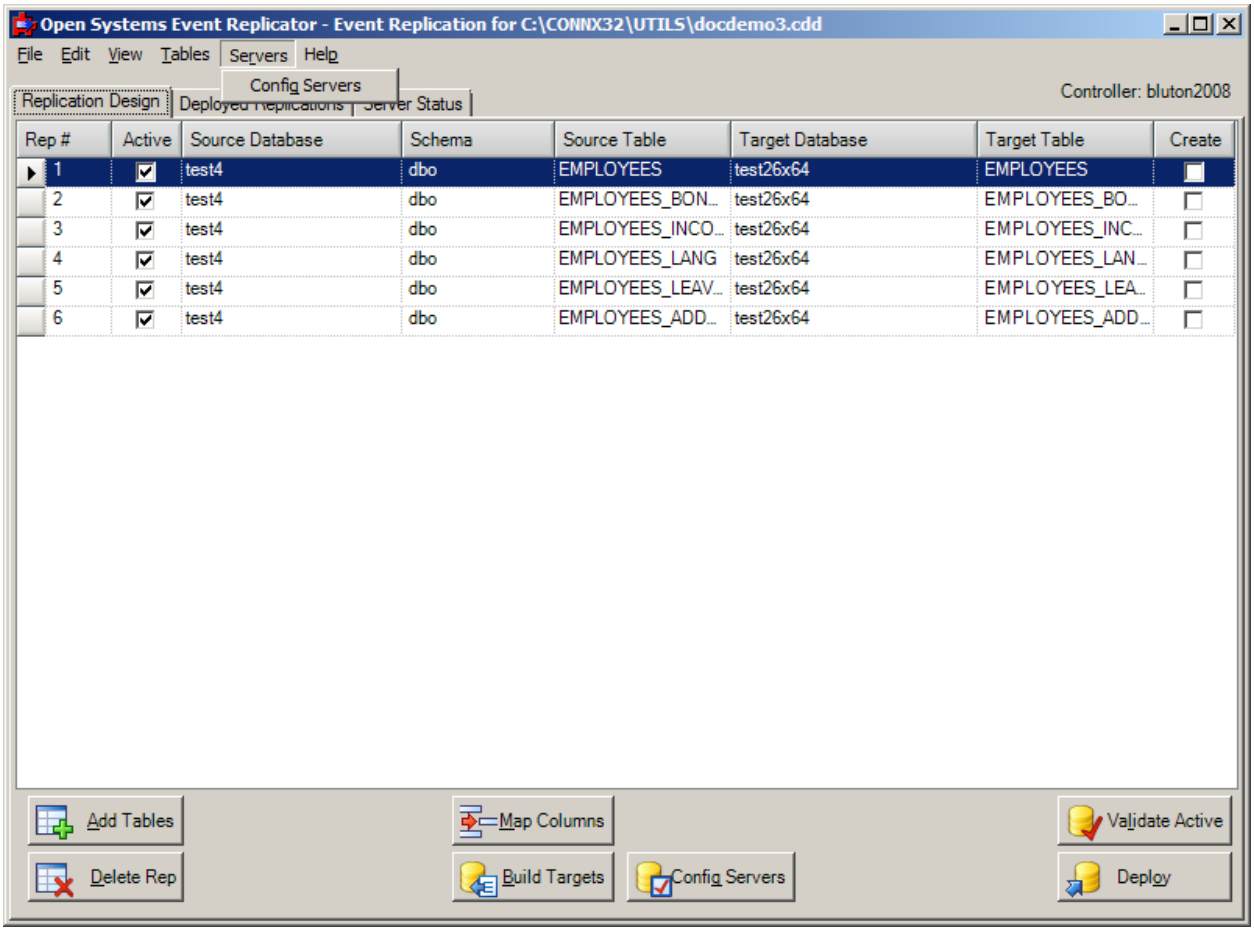
- **UnDeploy All**

  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:

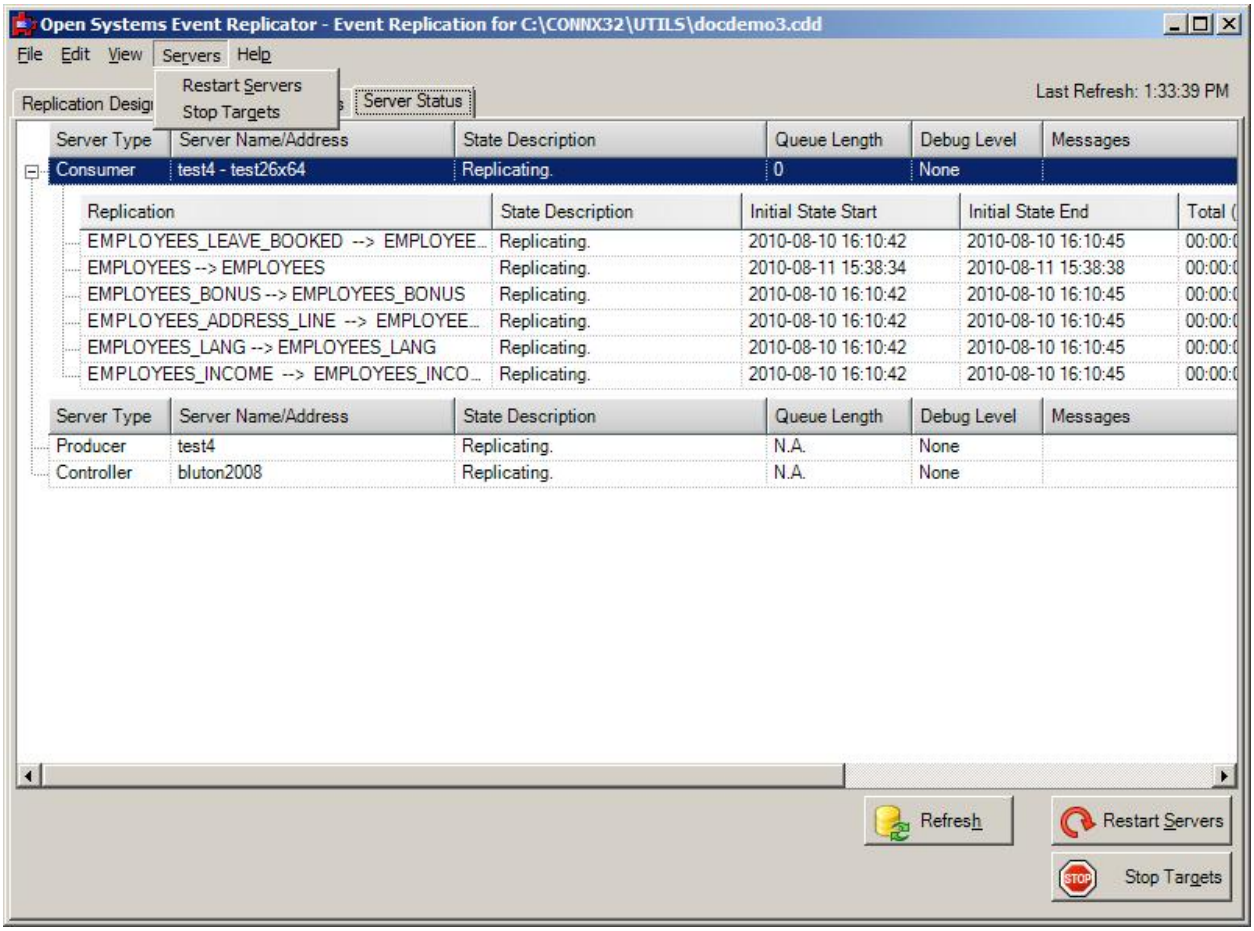


The Servers menu has one command:

- 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:

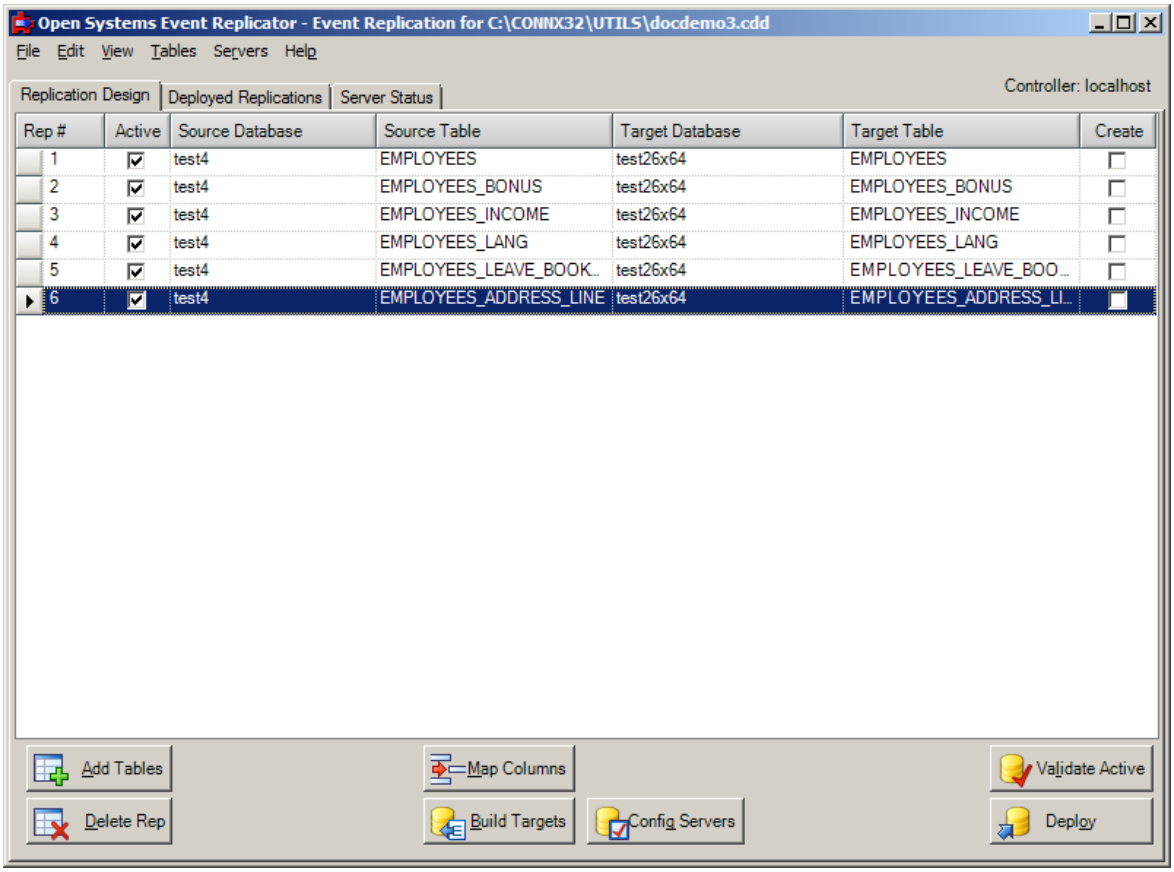
- User Guide

  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 data base, 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.

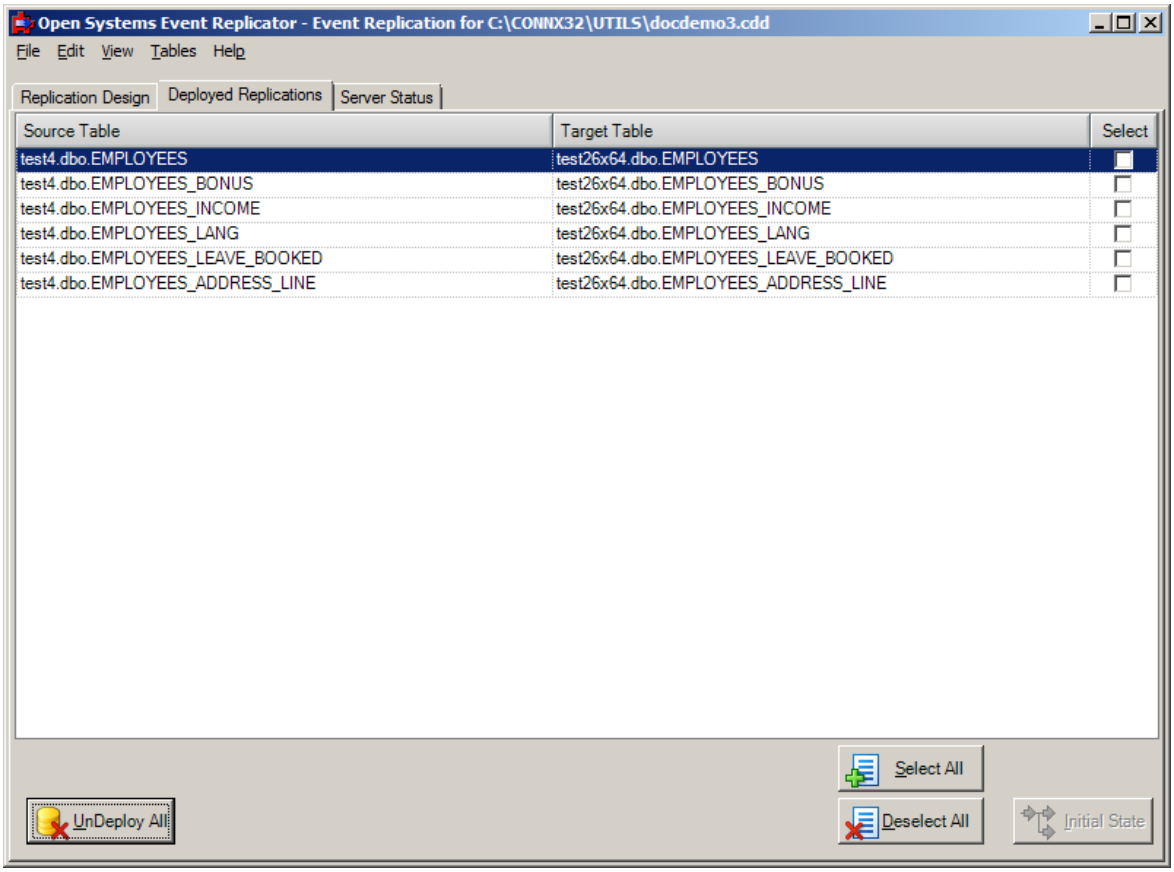
There are command buttons for the following:

- 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

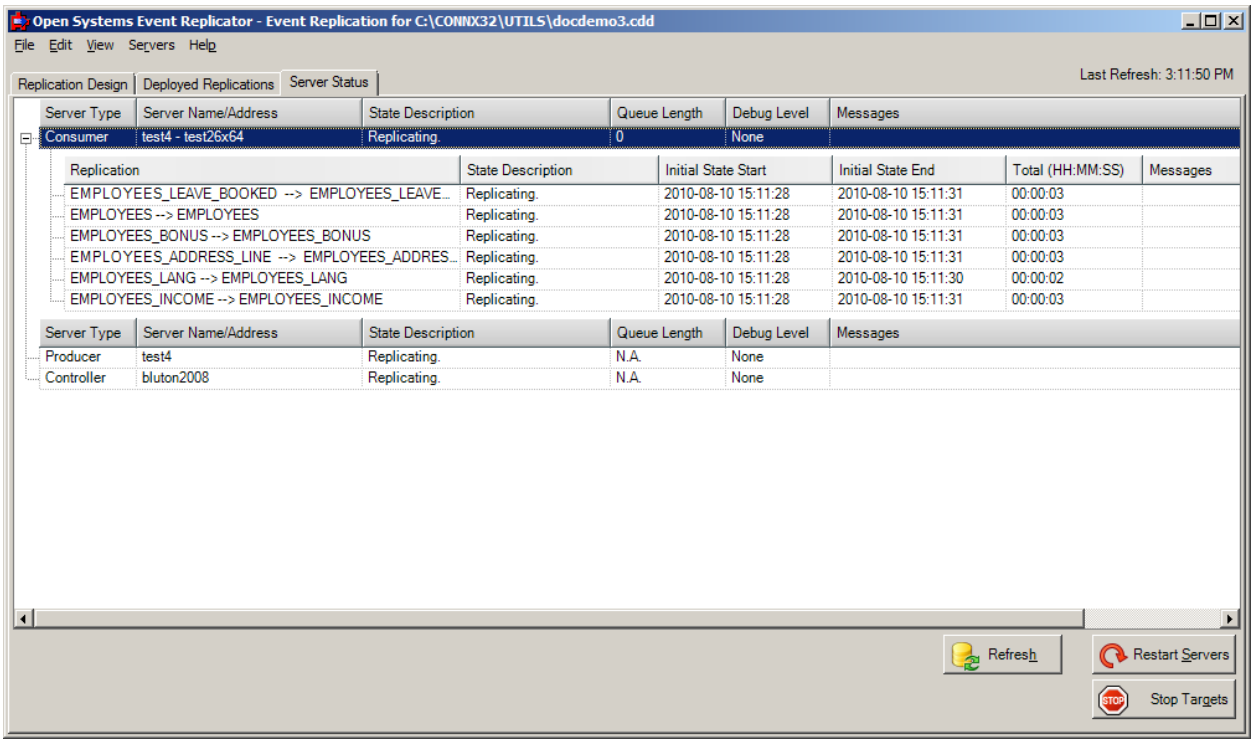
  Clears the **Select** check box for all replications

- Initial State

  Process an Initial State on all selected replications

Server Status Tab

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



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.

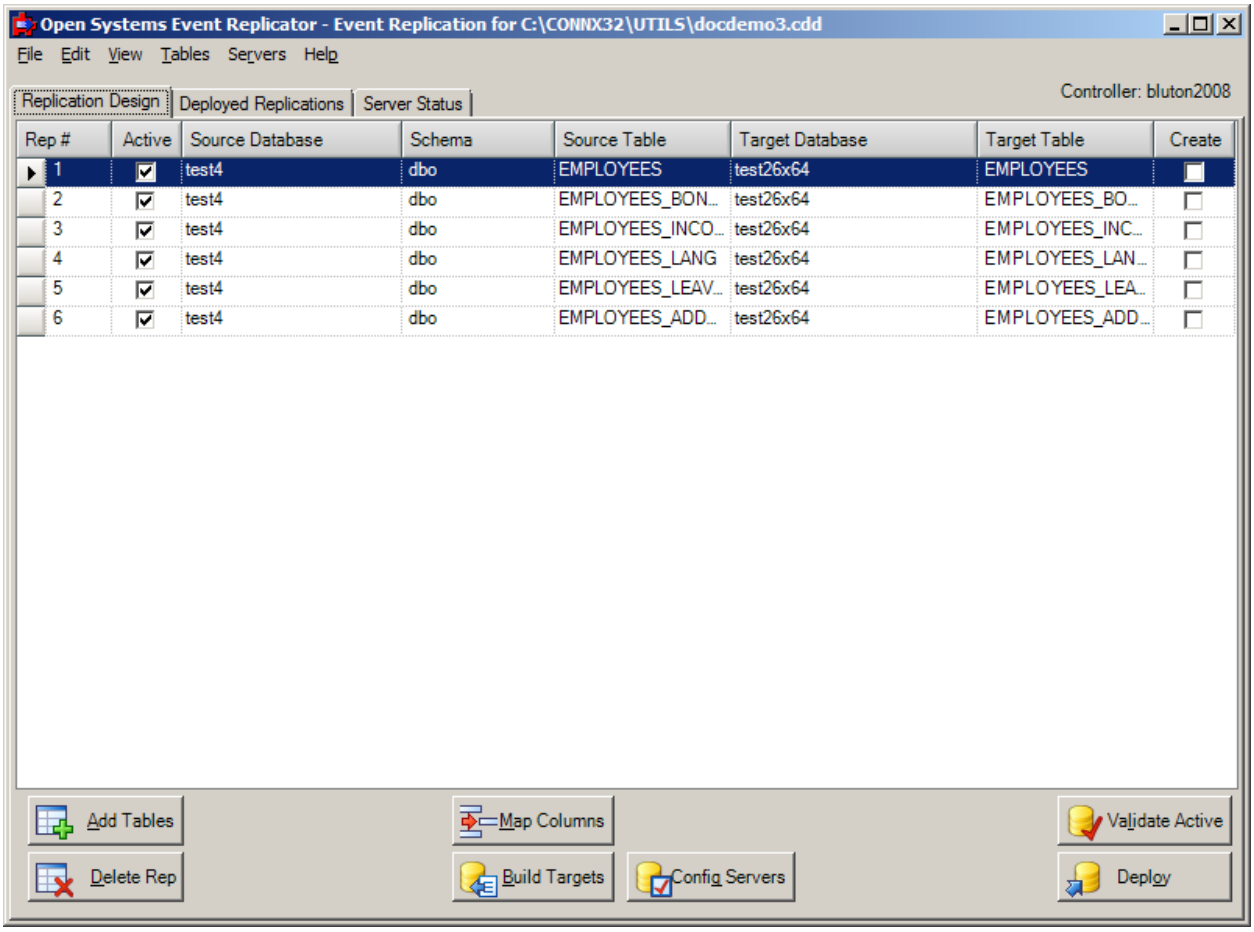
The server status information will automatically refresh every 30 seconds. To manually refresh the server status, click the **Refresh** button.

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

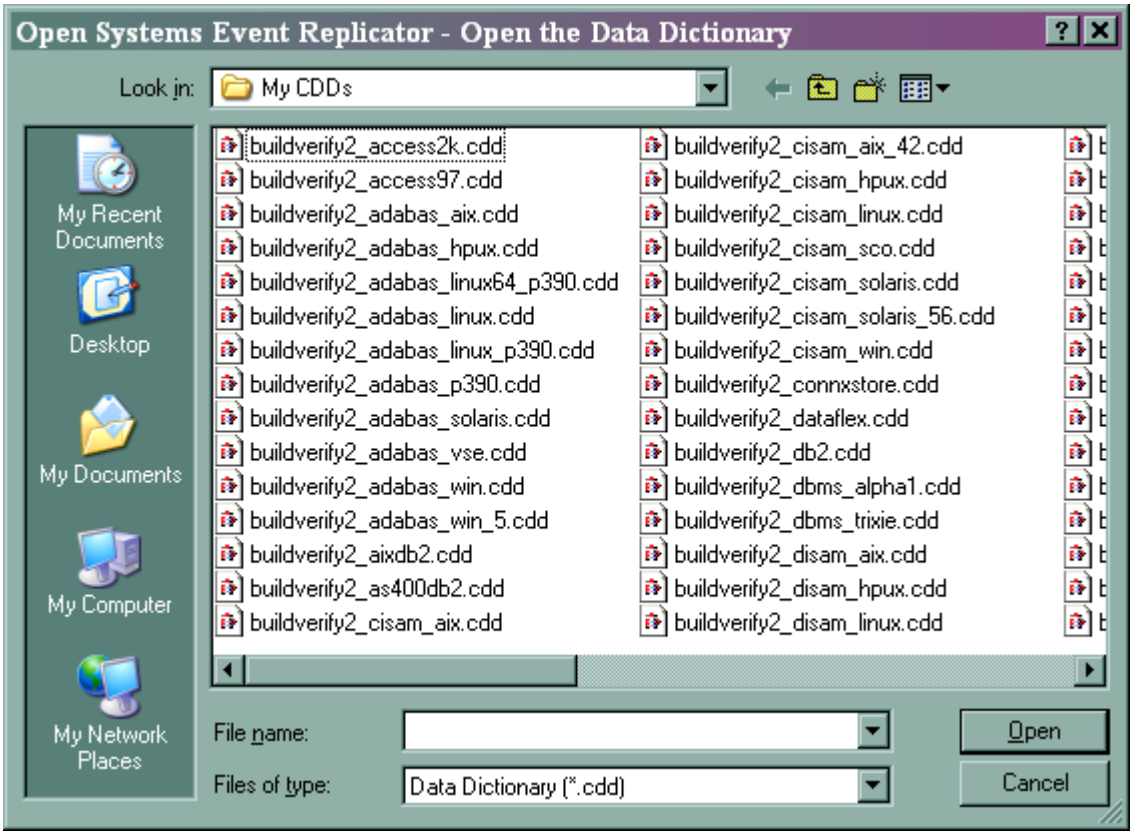
You can also adjust the display column width.

Open the Event Replication CDD

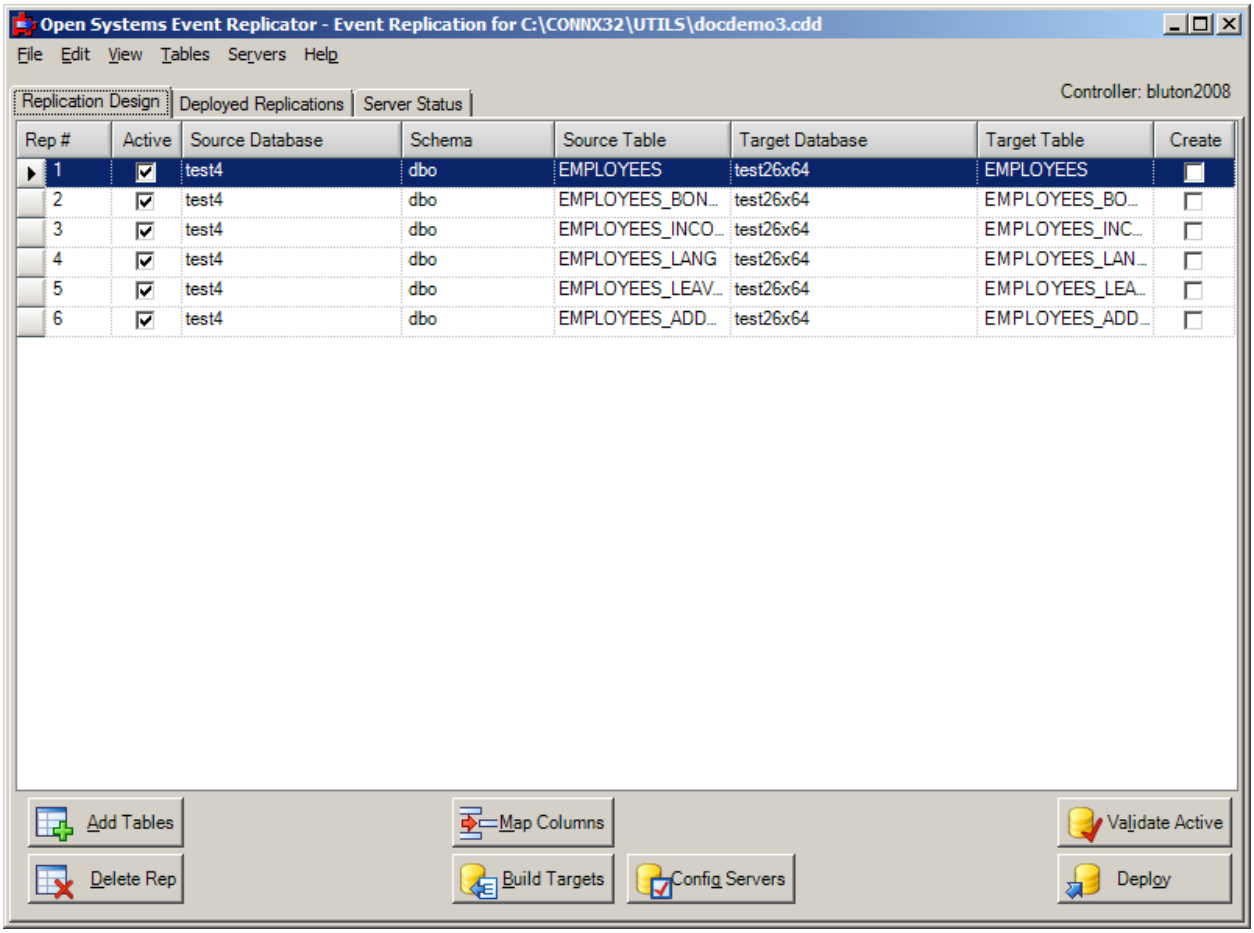
On the **Start** menu, click **Programs**, click **CONNX Open Systems Event Replicator** and then click **Replication Administrator.** The **Open Systems Event Replication** window with the information from the last opened CDD appears.



If there is no last opened CDD, or the last opened CDD has been moved or deleted, the **Open the Data Dictionary** window appears.



Select the CDD that you wish to add a table to and click **Open**. The **Open Systems Event Replication** window appears.



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

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.

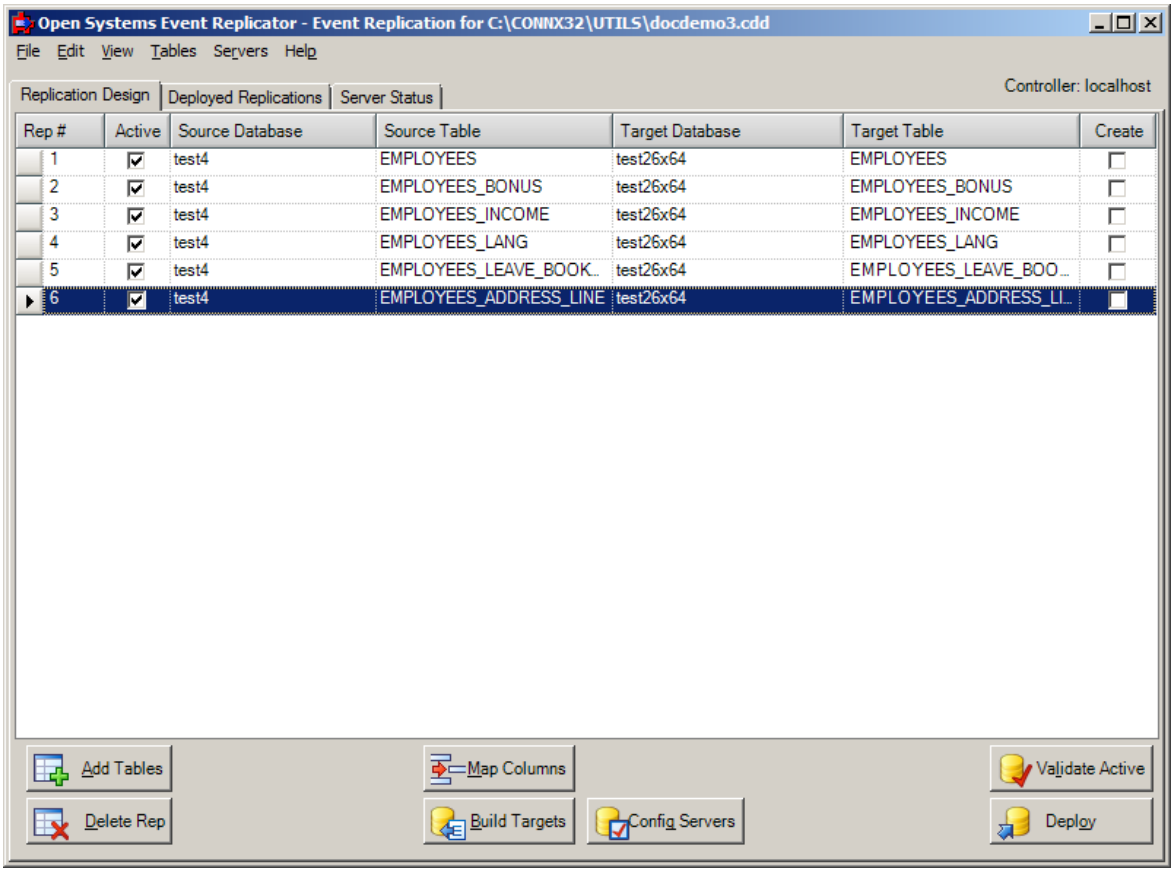
Note: Only non-aggregate SQL functions can be mapped to a **Target Column**.

Adjusting Display Column Width

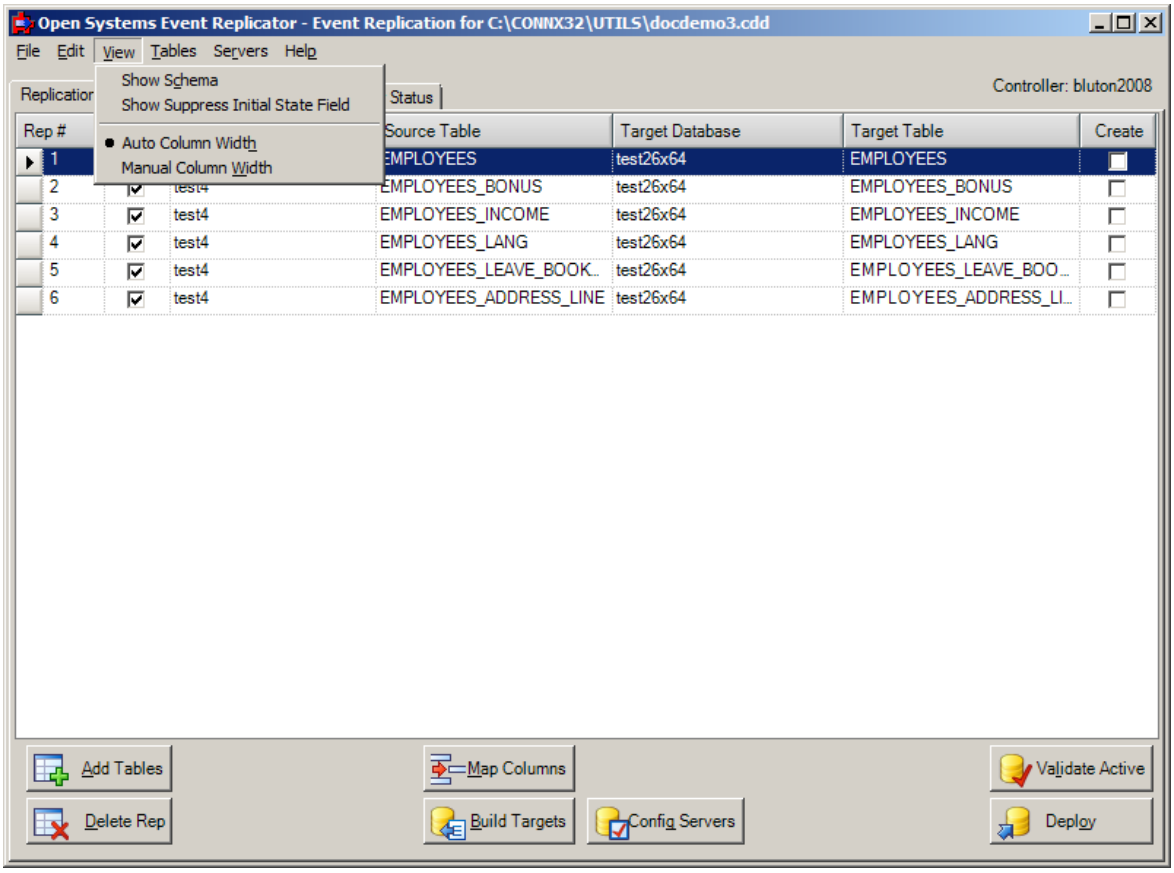
When replication information has been retrieved, by default, it fills the entire width of the display screen. You can adjust the width of any of the display columns.

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

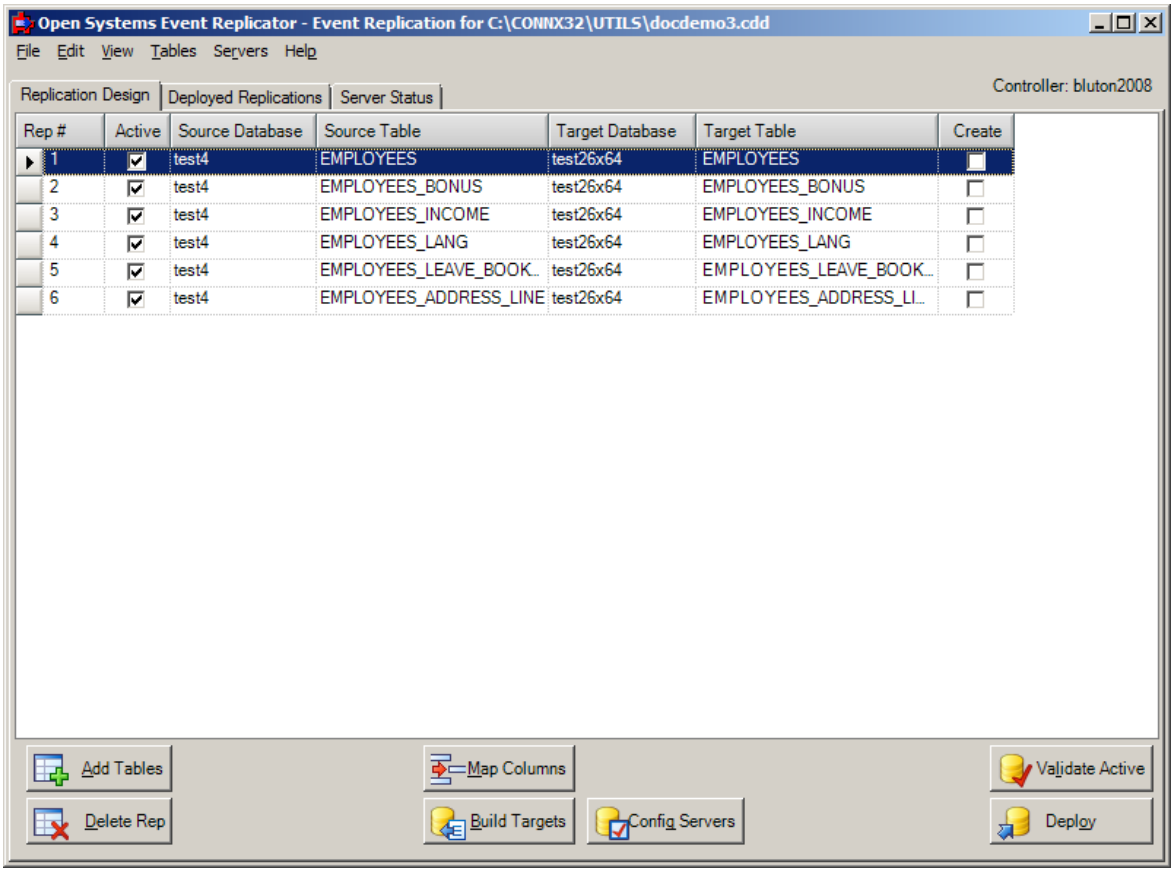
1. Open the Open Systems Event Replicator CDD.
2. Click the Replication Design tab. The following appears:



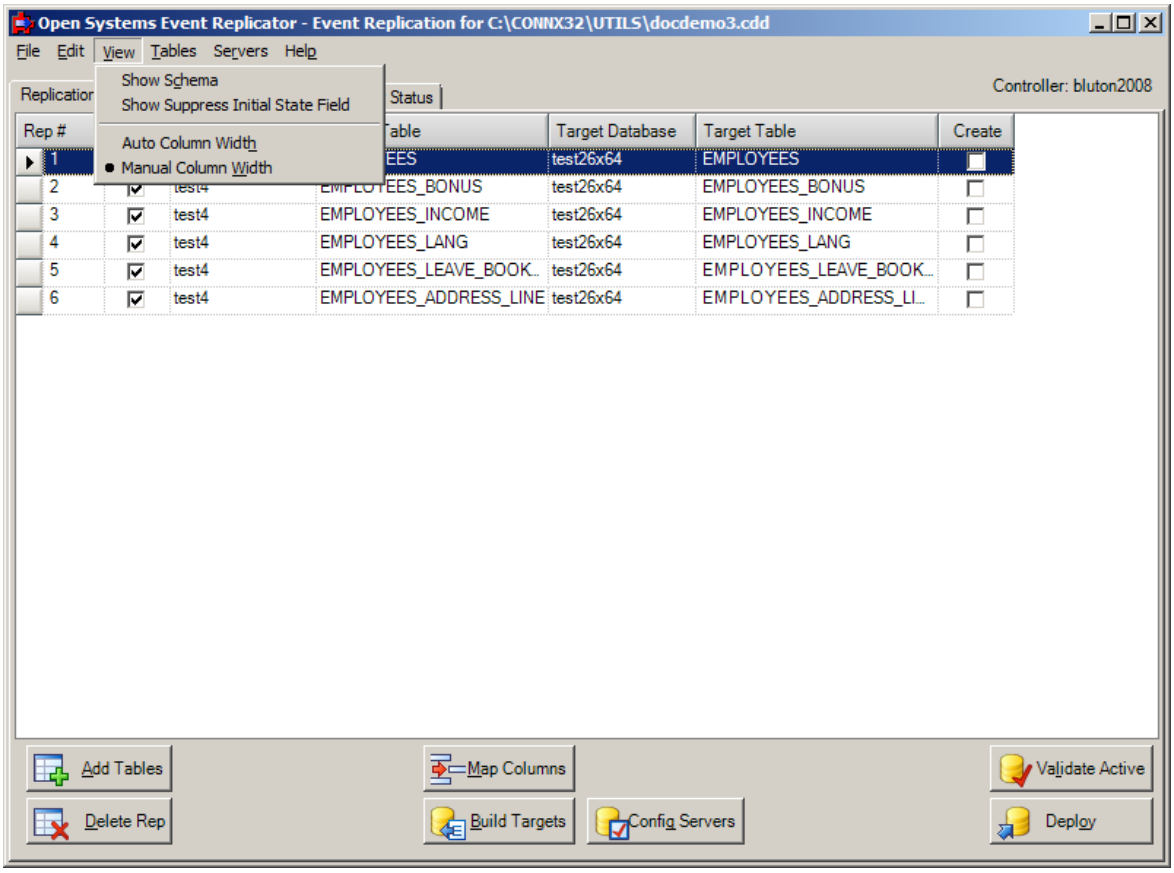
The **View** Menu displays the **Column Width** status. **Auto Column Width** is the default.



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.NSPAWNLAG
- 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.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.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.

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 ConnecX SQL Engine built-in features. 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 **ConnecX SQL Engine,** Chapter 21- Advanced Features of CONNX.

If two replications want the same tables but different data, in ConnecX SQL Engine, 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**.

Switching from Test to Production

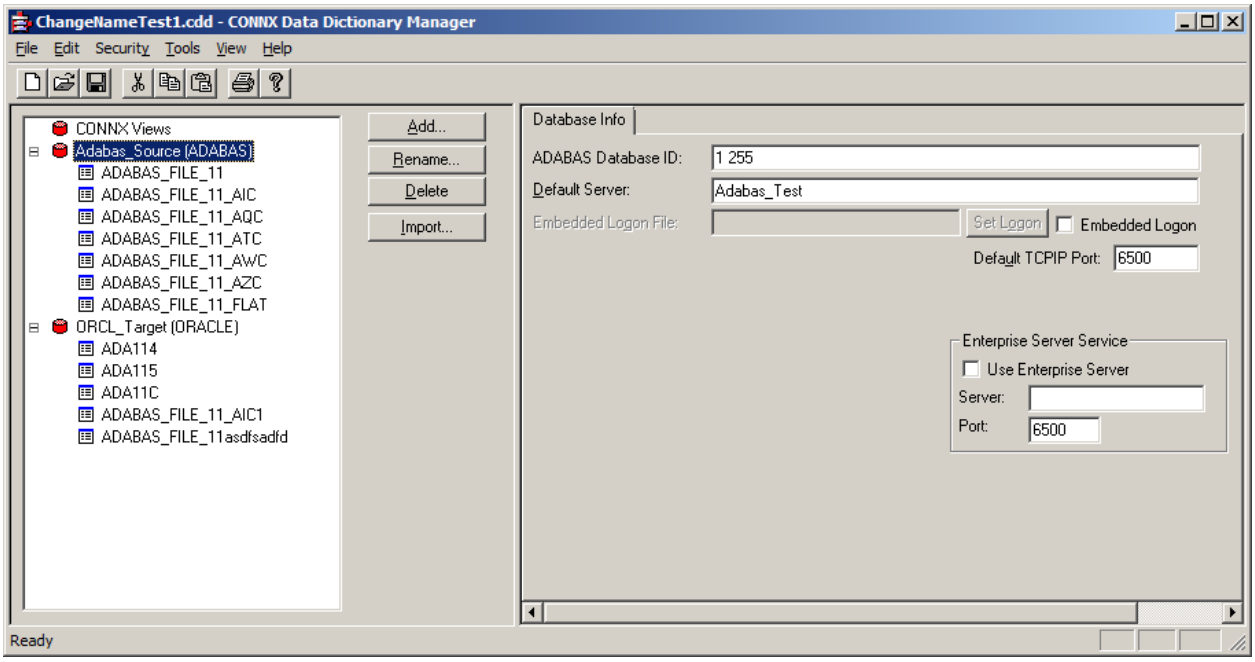
Typically, a CDD with replications will be created in a test or development environment. Once development and testing of the replications is done, the replications need to be rolled out to the production environment. This can be achieved by making a copy of the CDD and then modifying the server names and/or the database ID's in the CDD.

How to move a Event Replication Data Dictionary from a test to production environment.

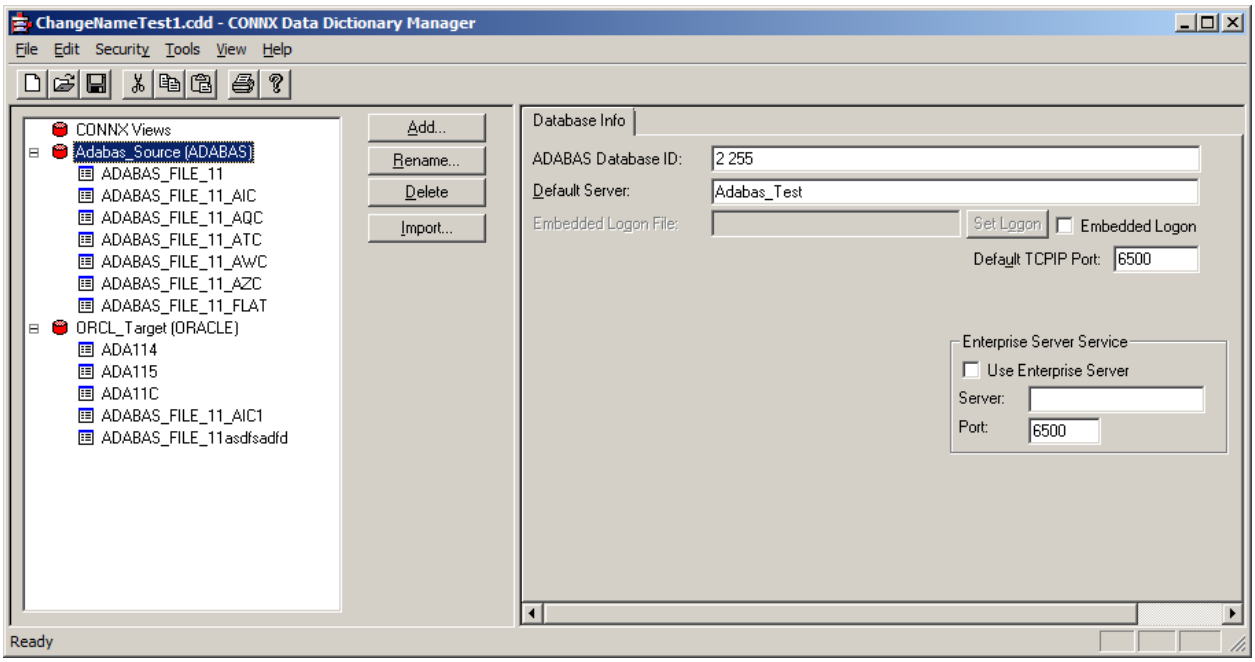
1. As a good practice with any modifications, make a back up copy of the CONNX Data Dictionary that is going to modified.

2. The following example is going to have a CONNX Data Dictionary that is replicating tables from a test Adabas Nucleus to a test Oracle database. In order to transition the data dictionary from a test to production environment, the two environments must have identical table names and structures to what was in the test databases.

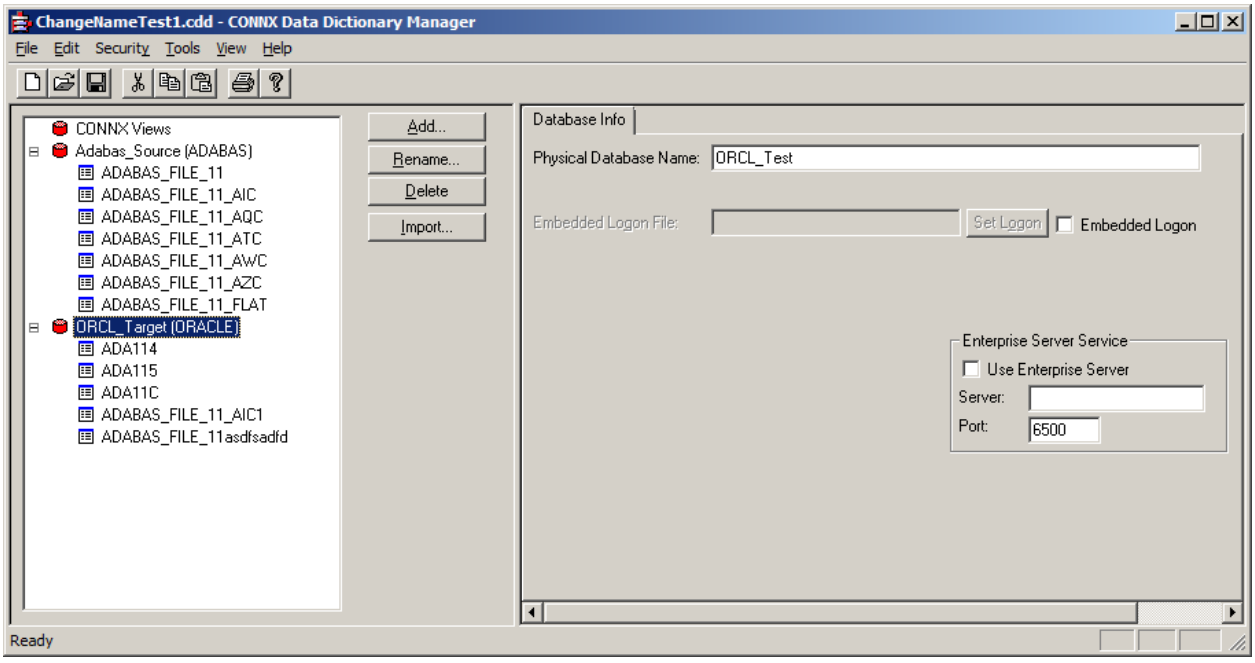
3. Close the Replication Administrator if open and then open the Data Dictionary in the Data Dictionary Manager. Select the Adabas Nucleus in the container list on the left side and it will show the Adabas Database ID:



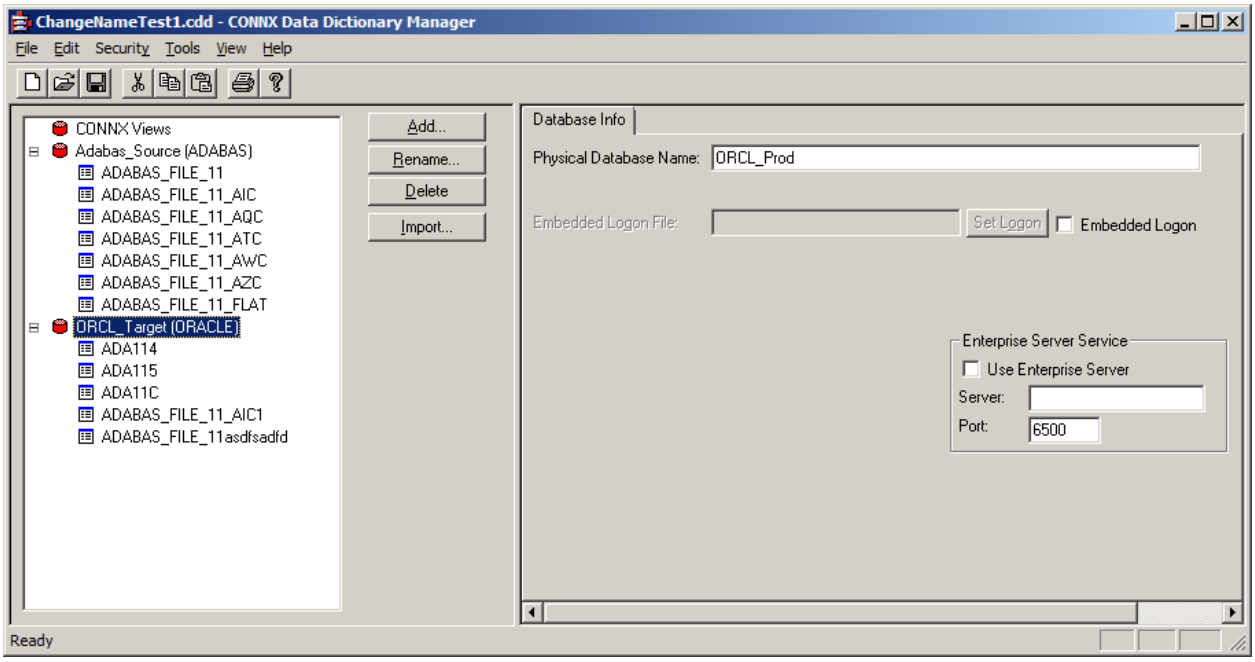
4. In the field for ADABAS Database ID, change the id number from the test nucleus to the production nucleus. In this example, DBID 1 is our test nucleus and it is changed to 2 the production nucleus



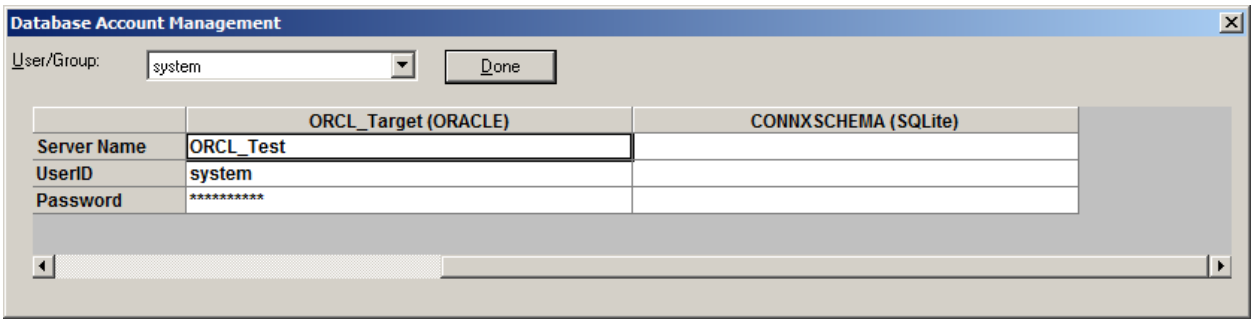
5. Next select the Oracle container in the list on the left and it will show the Physical Database Name of the Oracle server.



6. Change the Physical Database Name from the test server to the production server. In this example the test server is ORCL_Test and the production server is ORCL_Prod.

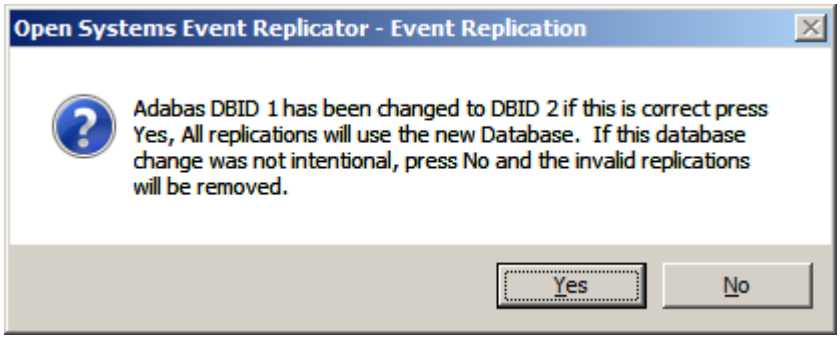


7. Next, for Oracle Servers the Database Account Management entries must also be changed. In the menu bar at the top of the CONNX Data Dictionary manager click "Security" -> "Database Account Management". That will bring up the Database Account Management window, using the User/Group dropdown list, select each of the accounts and then change the Server Name under the Oracle Database column. If necessary also change the UserID and Password to log into the new server.

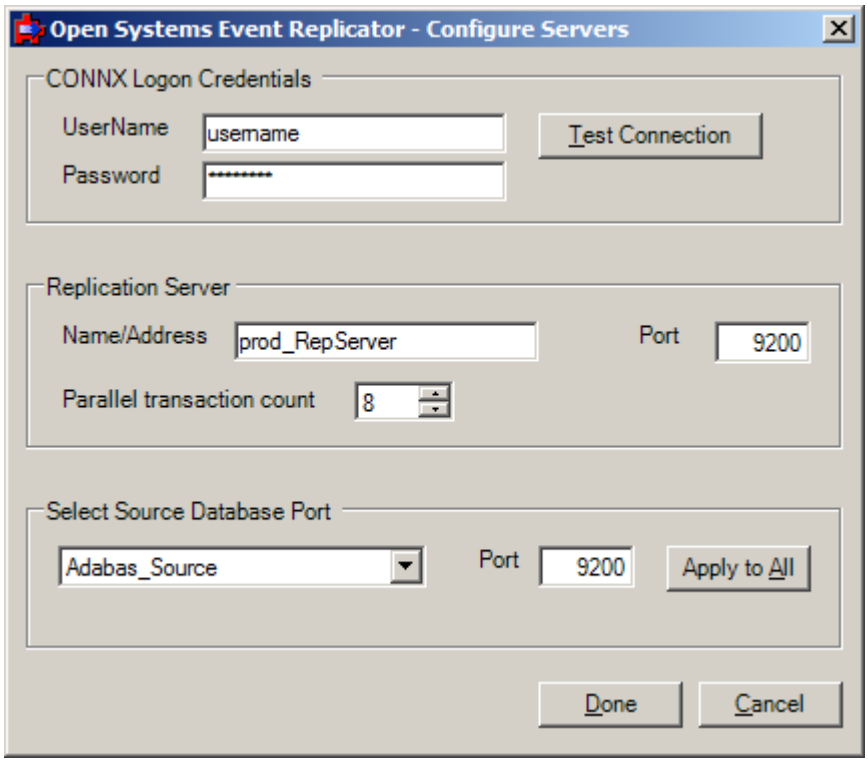


Once all of the entries in the User/Group dropdown have been changed to the new Oracle Server name, select the Done button.

8. Save the changes in the CONNX Data Dictionary Manger and close the data dictionary. Then open the Replication Administrator and the CONNX Data Dictionary that was just modified. The Replication Administrator will detect the changes to the servers and prompt the user if these changes were intentional. Select Yes to both change questions and the Replications will be updated to use the new servers specified.



9. Then select the menu item "Servers" -> "Config Servers" and now change the Replication Server to the production replication server.



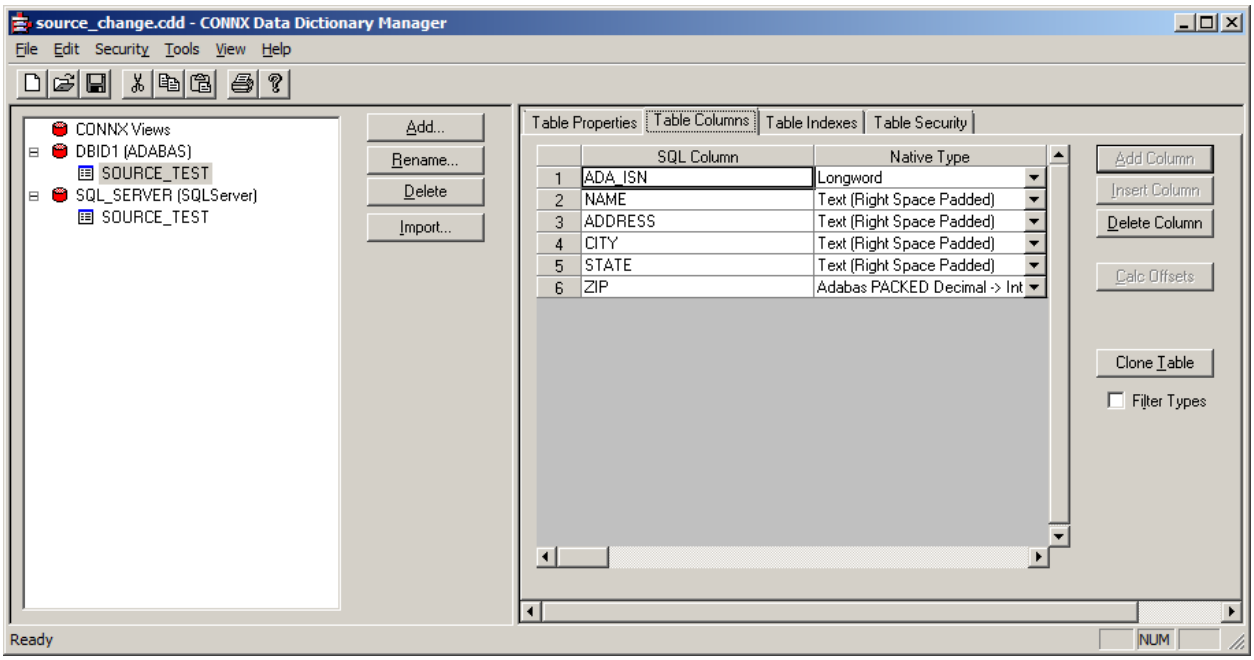
10. Select "Done" when finished and the Replications are ready to Deploy to the production environment.

Changing Source Tables

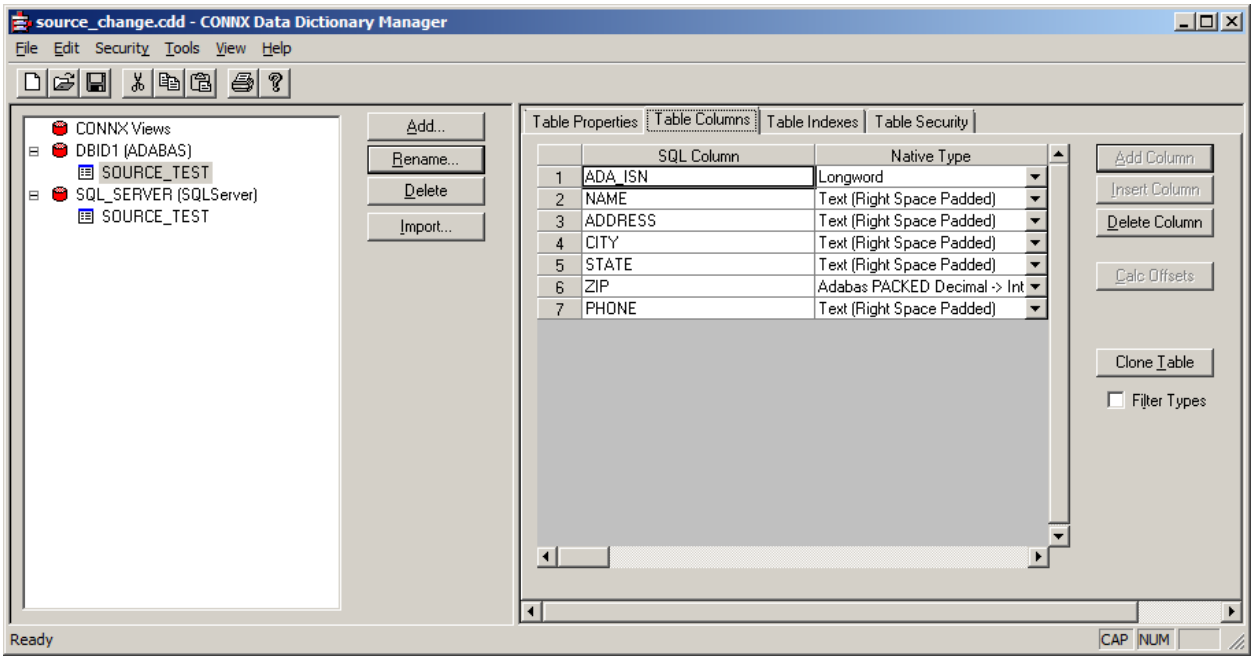
If the shape of a source table from a deployed replication changes (columns are added or deleted or a data type is changed), the source table will need to be re-imported into the CDD and the column mappings for the replication will need to be updated to reflect the new shape of the source table.

This is a sample showing the steps for adding a column to a source table and deploying that change to the replication server:

In this example, we have a source table with the following definition

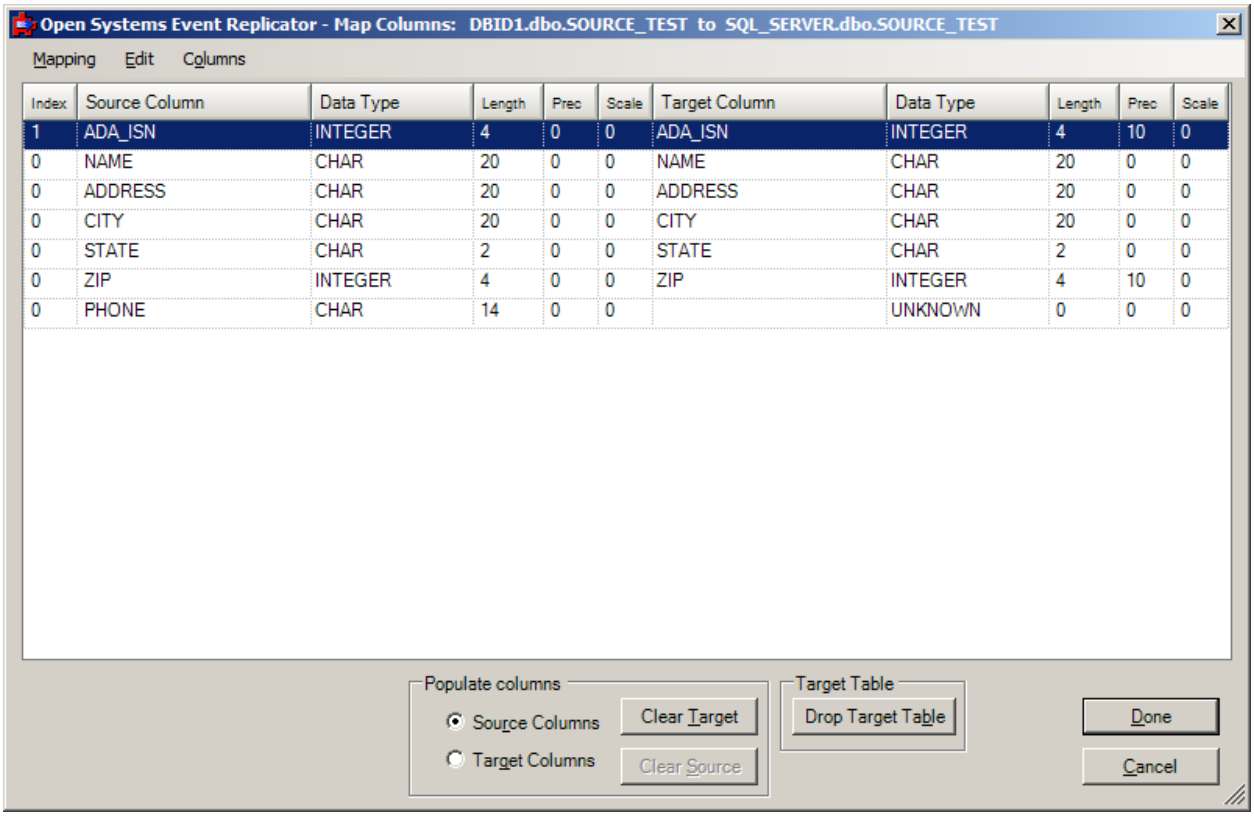


This table is deployed and replicating to a target table of the same name on SQL Server. If the column PHONE is added to the Adabas file, the table will need to be re-imported to reflect this new column.

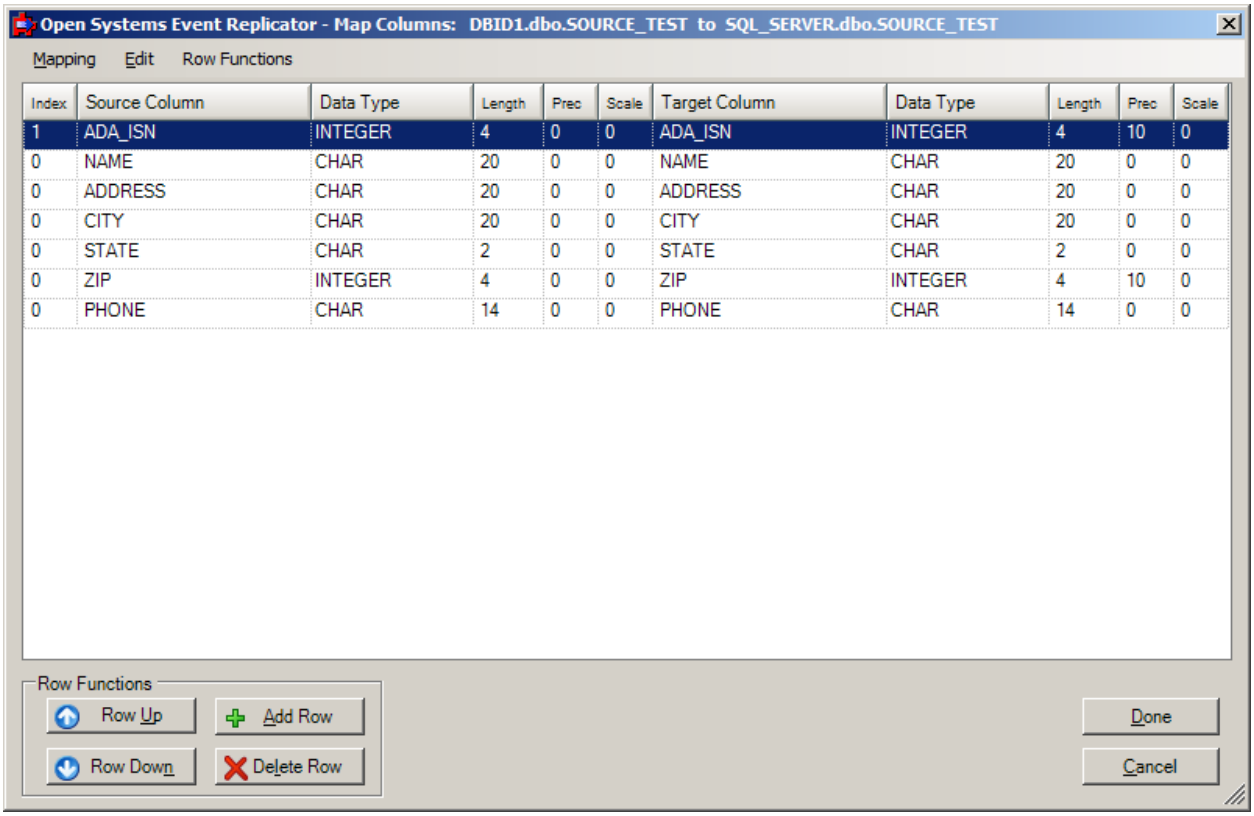


After re-importing the table, save the CDD and exit the CDD Manager. Open the CDD in the Replication Administrator. On the Replication Design tab, select the replication that contains the source table that was changed and press the Map Columns button.

On the Map Columns dialog, the new PHONE column appears on the source side but is not mapped to anything on the target.



Since the target table exists and does not contain a corresponding column, it needs to be dropped and re-created. To drop the target table, press the Drop Target Table button. This will create new default metadata for the new target table and automatically map the new column to the target.



Press Done to accept the new column mapping. On the Replication Design screen, the target table will have the Create check box checked.

You will now need to re-deploy the replications. The target table will be created as part of the deploy process. Alternatively, pressing the Build Targets button will create the new target table.

Note: If you are not using the Replication Administrator to create the target table, you will need to manually make the same change on the source table and re-import that table as well. In this case, skip the step of dropping the target table and map the new column manually. In this case, the target table will not be created at deploy time.

Note: Due to the internal record keeping mechanisms, it is possible for the Replication Administrator to not recognize an altered table as being the same as one that was already deployed. In this situation, it will recognize the replication or replications that contain the old form of the table or tables and will remove those replications. In this situation, a message will appear when the Replication Administrator is started informing you that there are replications that cannot be resolved to existing tables and will be removed. You will need to press Add Tables to re-add these replications. After adding the replication, you must adjust the target table to the existing table name and then follow the steps for mapping the new column.

Note: This sample illustrates how to update a replication when a new column has been added. The same concepts apply for removing columns or changing data types.

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

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}\CONNX32\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. I f 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.	Path 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.

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 mis-matched 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 my 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:	Contact technical support.

				(**<nucleus response code>**).	
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 \CONNX32\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.

Appendix A - Technical Support

Support Contacts

Product support and technical assistance for the ConnecX SQL Engine are available through your local Software AG Regional Support Center, Software AG's Empower website, or your Software AG Account Manager.

Access to Empower can be found at the following addresses:

http://empower.softwareag.com

http://www.softwareag.com

ConnecX SQL Engine support can also be reached via e-mail at support@softwareag.com.

Trial copies and pre-release versions are covered by separate contracts. Please contact your Software AG Account Manager for more information.

Appendix B - Copyright Page

Trademarks and Copyright

Legal Notices

This notice is valid for all products of the Adabas®, Bolero®, EntireX®, Natural® and Tamino® (the "Product") product line and associated Software AG products.

IMPORTANT: PLEASE READ BEFORE INSTALLING THE SOFTWARE

You are not allowed to install or use the Product without a corresponding license agreement.

If you have entered into a license agreement with one of Software AG's subsidiaries or distributors (the "License Agreement") the installation and use of the Product is subject to your acceptance of additional terms which are provided for you in the section Additional Terms.

If you have not entered into a License Agreement you must execute the license agreement with Software AG which is provided for you in the section Software AG License Agreement for Trial Versions of Software AG Software ONLY.

Copyright Notice

Adabas®, Bolero®, EntireX®, Natural® and Tamino® are Products of Software AG, a German corporation with its principal place of business at Uhlandstrasse 12, 64297 Darmstadt.

Software AG is the sole owner of all industrial property rights and copyright to the Products and accompanying user documentation or has the respective distribution rights. References made in or on the Products to the copyright or to other industrial property rights must not be altered, deleted or obliterated in any manner.

© Copyright 1998-2016 CONNX Solutions, Inc. All Rights Reserved.

© Copyright 2017 Software AG, Uhlandstrasse 12, 64297 Darmstadt, Germany 2016. All rights reserved.

Trademark Notice

The name Software AG and/or all Software AG Products are either trademarks or registered trademarks of Software AG. Other product and company names mentioned herein may be the trademarks of their respective owners.

CONNX, FTL, InfoNaut, InfoNaut Professional, and CONNX Data Synchronization are trademarks or registered trademarks of Software AG, in the USA and other countries.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at http://softwareag.com/licenses.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at http://softwareag.com/licenses and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

The following are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries:Microsoft, Microsoft Access, Microsoft Excel, Microsoft Internet Information Server,

Microsoft .NET, Microsoft SQL Server, Microsoft Visual Basic, Microsoft Visual C++, Microsoft Visual FoxPro, Microsoft Visual Studio, Microsoft Windows.

The following are either registered trademarks or trademarks of their respective companies or organizations in the United States and/or other countries: Adabas - Software AG; Extra!, Attachmate - Attachmate Corporation; Borland C++ - Inprise Corporation; Crystal Reports - Seagate Software; DataFlex - Data Access Corporation; DB2, AIX, AS/400, C-ISAM, OS/2, OS/390, OS/400, IBM Communications Server, IBM Personal Communications, IBM eNetwork Personal Communications, IBM eNetwork Communications Server - International Business Machines Corporation; Dharma ODBC Integrator - Dharma Systems, Inc.; FOCUS -- IBI, Inc.; HP-UX -- Hewlett-Packard Company; Impromptu - Cognos Incorporated; Itanium - Intel Corporation; JetForm - JetForm Corporation; Linux - Linus Torvalds; NetWare for SAA, Novell - Novell, Inc.; Micro Focus -- Micro Focus International Limited; Oracle, Oracle Rdb, Oracle CODASYL DBMS, Oracle Developer, Form Builder, and Report Builder - Oracle Corporation; Paradox - Corel Corporation; Pathworks, Network File Transfer, RMS, DECnet, VMS, OpenVMS, VAXServer, AlphaServer - Compaq Computer Corporation; PowerBuilder - Sybase, Inc.; POISE (People Oriented Information Systems for Education) -- Campus America/Jenzabar; POWER*flex* - POWER*flex* Corporation; RUMBA, Wall Data - NetManage, Inc.; SCO -- Caldera Systems, Inc.; SCT, Plus2000 - Systems and Computer Technology Corporation; SIS - University of California Berkeley; Solaris, Sun - Sun Microsystems, Inc.; Unix - The Open Group.

Portions of the **PostgreSQL Data Base Management System** are copyrighted © 1996-2002, PostgreSQL Global Development Group
Portions Copyright © 1994-7 Regents of the University of California
Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.
IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

SAS Institute Inc.
SAS/C OnlineDocTM, Release 7.00, Cary, NC: SAS Institute Inc., 2001.
SAS/C OnlineDocTM, Release 7.00, April 2001
SAS/C OnlineDocTM, Release 7.50, April 2004
Copyright © 2001 by SAS Institute Inc., Cary, NC, USA.
All rights reserved. Produced in the United States of America.
U.S. Government Restricted Rights Notice
Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987).
SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.
April 2001
SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

dbCrypto
Copyright (c) 2002, Dr Brian Gladman, Worcester, UK.
All rights reserved.

LICENSE TERMS
The free distribution and use of this software in both source and binary
form is allowed (with or without changes) provided that:
1. distributions of this source code include the above copyright
   notice, this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright
   notice, this list of conditions and the following disclaimer
   in the documentation and/or other associated materials;
3. the copyright holder's name is not used to endorse products
   built using this software without specific written permission.
ALTERNATIVELY, provided that this notice is retained in full, this product
may be distributed under the terms of the GNU General Public License (GPL),
in which case the provisions of the GPL apply INSTEAD OF those given above.
DISCLAIMER
This software is provided 'as is' with no explicit or implied warranties
in respect of its properties, including, but not limited to, correctness
and/or fitness for purpose.

SIS

Copyright (c) 1989-1994, 2017 The Regents of the University of California.

All rights reserved.

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

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

ICU Library

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2005 International Business Machines Corporation and others

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE

FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION

WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Hiredis

Copyright (c) 2009-2011, Salvatore Sanfilippo <antirez at gmail dot com>

Copyright (c) 2010-2011, Pieter Noordhuis <pcnoordhuis at gmail dot com>

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.
* Neither the name of Redis nor the names of its contributors may be used
  to endorse or promote products derived from this software without specific
  prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR

ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OCILib

Licensed under Apache 2.0

SAFMQ

Copyright 2004-2007 Matthew J. Battey, Omaha, NE

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

FreeTDS

Copyright (C) 1998-1999 Brian Bruns

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details. You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

ACE Library

Copyright and Licensing Information for ACE(TM), TAO(TM), CIAO(TM), DAnCE(TM), and CoSMIC(TM)

ACE(TM), TAO(TM), CIAO(TM), DAnCE>(TM), and CoSMIC(TM) (henceforth referred to as "DOC software") are copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (c) 1993-2009, all rights reserved. Since DOC software is open-source, freely available software, you are free to use, modify, copy, and distribute--perpetually and irrevocably--the DOC software source code and object code produced from the source, as well as copy and distribute modified versions of this software. You must, however, include this copyright statement along with any code built using DOC software that you release. No copyright statement needs to be provided if you just ship binary executables of your software products.

You can use DOC software in commercial and/or binary software releases and are under no obligation to redistribute any of your source code that is built using DOC software. Note, however, that you may not misappropriate the DOC software code, such as copyrighting it yourself or claiming authorship of the DOC software code, in a way that will prevent DOC software from being distributed freely using an open-source development model. You needn't inform anyone that you're using DOC software in your software, though we encourage you to let us know so we can promote your project in the DOC software success stories.

The ACE, TAO, CIAO, DAnCE, and CoSMIC web sites are maintained by the DOC Group at the Institute for Software Integrated Systems (ISIS) and the Center for Distributed Object Computing of Washington University, St. Louis for the development of open-source software as part of the open-source software community. Submissions are provided by the submitter ``as is'' with no warranties whatsoever, including any warranty of merchantability, noninfringement of third party intellectual property, or fitness for any particular purpose. In no event shall the submitter be liable for any direct, indirect, special, exemplary, punitive, or consequential damages, including without limitation, lost profits, even if advised of the possibility of such damages. Likewise, DOC software is provided as is with no warranties of any kind, including the warranties of design, merchantability, and fitness for a particular purpose, noninfringement, or arising from a course of dealing, usage or trade practice. Washington University, UC Irvine, Vanderbilt University, their employees, and students shall have no liability with respect to the infringement of copyrights, trade secrets or any patents by DOC software or any part thereof. Moreover, in no event will Washington University, UC Irvine, or Vanderbilt University, their employees, or students be liable for any lost revenue or profits or other special, indirect and consequential damages.

DOC software is provided with no support and without any obligation on the part of Washington University, UC Irvine, Vanderbilt University, their employees, or students to assist in its use, correction, modification, or enhancement. A number of companies around the world provide commercial support for DOC software, however. DOC software is Y2K-compliant, as long as the underlying OS platform is Y2K-compliant. Likewise, DOC software is compliant with the new US daylight savings rule passed by Congress as "The Energy Policy Act of 2005," which established new daylight savings times (DST) rules for the United States that expand DST as of March 2007. Since DOC software obtains time/date and calendaring information from operating systems users will not be affected by the new DST rules as long as they upgrade their operating systems accordingly.

The names ACE(TM), TAO(TM), CIAO(TM), DAnCE(TM), CoSMIC(TM), Washington University, UC Irvine, and Vanderbilt University, may not be used to endorse or promote products or services derived from this source without express written permission from Washington University, UC Irvine, or Vanderbilt University. This license grants no permission to call products or services derived from this source ACE(TM), TAO(TM), CIAO(TM), DAnCE(TM), or CoSMIC(TM), nor does it grant permission for the name Washington University, UC Irvine, or Vanderbilt University to appear in their names.

UMac

Copyright (c) 1999-2000 Ted Krovetz (tdk@acm.org)

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee, is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of the University of California and Ted Krovetz not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

The Regents of the University of California and Ted Krovetz disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the University of California or Ted Krovetz be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Gigabase

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHOR OF THIS SOFTWARE BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

System.Net.FtpClient

Copyright (c) 2013 J.P. Trosclair

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

SharpSSH

Copyright (c) 2002-2015 Atsuhiko Yamanaka, JCraft,Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JCRAFT, INC. OR ANY CONTRIBUTORS TO THIS SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Mentalis

Copyright © 2002-2007, The Mentalis.org Team

All rights reserved.

http://www.mentalis.org/

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Neither the name of the Mentalis.org Team, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NLog

Copyright (c) 2004-2016 Jaroslaw Kowalski <jaak@jkowalski.net>, Kim Christensen, Julian Verdurmen

All rights reserved.

Redistribution and use in source and binary forms, with or without

modification, are permitted provided that the following conditions

are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of Jaroslaw Kowalski nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MariaDB ODBC Driver

The MariaDB client libraries for C, Java and ODBC are distributed under the LGPL license, version 2.1 or later. The LGPL license allows you to distribute these MariaDB client libraries freely with any application.

About Software AG, Inc.

Based in Reston, Va., Software AG, Inc. is a wholly owned subsidiary of Software AG headquartered in Darmstadt, Germany. Founded in 1969, Software AG continues to be Europe's largest and a leading global provider of system software and services enabling enterprise data integration and management. Software AG's products control the central IT processes of thousands of renowned companies worldwide including Lufthansa, Siemens, Citibank, Merck, DaimlerChrysler, Sony, BP and Telefonica. Software AG develops products and solutions that support the XML (Extensible Markup Language) standard. XML simplifies the exchange of documents and data as well as the integration of cutting-edge Web applications into traditional IT architectures. In 2003, the corporation achieved 422 million euros in total revenue. Software AG has representation in 59 countries worldwide, and currently employs a staff of approximately 2,600.

Software AG and/or all Software AG products are either trademarks or registered trademarks of Software AG.

All other product and company names mentioned herein are used for identification purposes only, and may be the trademarks or service marks of their respective owners.

Index

2

2039 152nd Avenue NE 369

5

50 Millstone Road 369

8

88CONNX 369

A

A2A 225

A2A specific Unix/Linux Environment Variables 306

About 3, 6, 61, 108, 111, 144, 237, 240, 279, 289, 294, 316, 332, 369

about cloning 316

about obtaining 6

About Open Systems 289

About Software AG 369

Accept 1, 15, 145, 306, 334

accept incoming 334

Access 3, 20, 145, 165, 334, 368, 369

ServLine24 368

ACID 3

preserve 3

ACID stands 3

Action 334, 366

Action type 366

Nx 366

Active 11, 101, 111, 115, 117, 119, 128, 145, 165, 173, 240, 244, 250, 273, 290, 292, 294, 317, 332, 334, 367

Active checkbox 317

active Event Consumers 11

Active Servers 317

Validating 317

Adabas ... 1, 6, 11, 12, 14, 15, 18, 20, 38, 61, 65, 67, 78, 108, 115, 135, 145, 165, 173, 177, 237, 257, 306, 316, 334, 368, 369

changes 1

CONNX 108, 237

data dictionary containing 6

OLEDB Driver 1

Adabas ConnecX 1

ADABAS Count Selection window 38, 145

Adabas Import 38

Adabas Nucleus 6, 20, 135, 257, 306, 334

running 20

Adabas SQL Gateway 1, 6, 11, 12, 14, 15, 18, 20, 38, 108, 145, 165, 173, 177, 237, 316, 368

Adabas SQL Gateway Installation Guide 15

Adabas SQL Gateway User Guide 20, 38

Adabas Target Tables 65, 67, 78

Adabas to Adabas 225

Adabas v6.1 6

ADABAS_FILE_11 ..... 145, 165, 173

ADABAS_FILE_11_AIC 165

ADABAS_FILE_11_AIC table 165

ADABCKSLEEPTIME 306

ADABCKTIMEOUT 306

adadbm 212

adaopr 266

add 3, 11, 38, 48, 61, 67, 108, 115, 145, 165, 177, 237, 290, 298, 316

Source Database 108, 237

table 61

Add Row 65, 67, 177

Add Tables 48

Adding Adabas to Adabas replications 225

Adding New Column Mappings 67

Adjusting 302

Display 302

ADM 334

Admin 15, 334

connect 334

writeable 334

Administrator 1, 334

administrator-level 14

administrator-level privileges 14

Advanced Features 316

CONNX 316

Agreement 15, 369

subject 369

AIX 12, 369

All . 3, 6, 11, 14, 15, 38, 48, 65, 67, 78, 108, 111, 117, 119, 128, 135, 139, 145, 165, 173, 177,

237, 240, 244, 250, 257, 261, 273, 316, 317, 332, 334, 367, 369
port number 108, 237
ALL replications....128, 250
alloc....334
Allows SQL....1
ALTERNATIVELY 369
Although ADABAS_FILE_11_AIC....173
America 369
United States 369
and/or 369
API....334
API Error....334
Apply 3, 11, 108, 177, 237, 316, 369
April 2001 369
AREA_CODE 177
AS/400....369
assistance 368
ConnecX 368
Atomicity....3
preserves 3
Attachmate 369
Attachmate Corporation 369
Available Tables....38, 145

B

Bad Reply....334
Bellevue....369
Borland 369
BP....369
Build Target....48, 117
Build Targets 115, 290
Building....115
New....115
Building 200 369
Business Case 1

C

C#.NET....6
cache....334
EP 334
calculate15 177
Caldera....369
Caldera Systems....369
California Berkeley....369
University....369
Call....1, 3, 145, 165, 306, 334, 366
called during....334
Campus America/Jenzabar 369
Cancel....37, 38, 145, 177
Cary 369
CDD 11, 14, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 111, 115, 117, 119, 122, 128, 135, 139, 145, 165, 173, 177, 213, 237, 240, 244, 247, 250, 257, 261, 271, 273, 298, 316, 317, 332, 334
Select 65, 101, 103, 317
CDD containing....145, 165, 173, 177
source....145, 165, 173, 177
CDD Initialized 334
CDD Object....334
CDD Security 38, 145
change 1, 3, 6, 11, 15, 38, 61, 65, 67, 74, 78, 80, 85, 91, 94, 108, 119, 145, 165, 173, 237, 244, 271, 302, 306, 332, 334, 369
Adabas 1
Event Replication Administrator 11
Source 65
change CONNX.REPLICATION.ENGINE_PORT 306
port 306
changes back....302
changes during 3
changing DEBUG_LEVEL 332
Chapter 21-Advanced Features....316
CONNX 316
Check....115, 117, 119, 122, 128, 139, 165, 244, 247, 250, 261, 317, 332, 334, 367
Controller....334
Engine 334
Event Consumer....334
Check API....334
Check Event Consumer....334
Check log....334
Choose Destination Location window....15
Choose Folder window 15
C-ISAM 369
Citibank....369

City 177
CITY_ABBR 177
Clear Active 48, 128, 173, 250, 290
Clear Source 91, 94
Clear Target 91, 94
Clone Table 177, 316
Clone Table Assistant 177, 316
Cntrl+O 271
Cntrl+S 271
cnxep.env 20
CNXEPCFG 334
CNXMQ 6
Co 369
COL 334
column mapping 11, 48, 61, 67, 74, 91, 94
Column Order 78
Column Width 279, 302
columns 1, 11, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 103, 111, 115, 117, 177, 240, 279, 284, 294, 298, 302, 316, 317, 334
  Mapping 65
command line 266
Commercial Computer Software 369
COMMIT 334
completes . 3, 11, 15, 20, 37, 111, 145, 165, 173, 177, 240, 334
  uninstall 37
completes concatenating 177
completes excluding 177
completes performing 177
Computer Technology Corporation 369
Concatenating 177
  Multiple Data 177
Config 108, 237, 334
  existing 334
config file 334
Config Servers 48, 108, 237
Configuration API 334
Configuration Path 334
Configuration Replication window 119, 244
CONFIGURE 48, 108, 177, 237, 334
Configure Replication Server 108, 237
Configure Servers dialog 334
Configure Servers window 48, 108, 145, 177, 237
Configuring 48, 108, 145, 237
  Replication 108, 237
Confirm Password 15
Connecting 20, 38, 334
  Admin 334
  Controller 334
  EP 334
  Replicator 334
  SQL Server 20, 38
  Target DB 334
ConnecX 1, 6, 11, 12, 14, 18, 20, 38, 316, 368
  assistance 368
  containing 20
  portion 14
  server portion 14
CONNX 1, 6, 11, 14, 15, 18, 20, 37, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 111, 115, 117, 119, 122, 128, 135, 139, 145, 165, 173, 177, 237, 240, 244, 247, 250, 257, 261, 298, 306, 316, 328, 329, 330, 331, 332, 334, 368, 369
  Advanced Features 316
  Chapter 21-Advanced Features 316
CONNX 10.5 ....65, 101, 103
CONNX Basics 108, 237
CONNX CD 15, 37
CONNX Configuration 20, 38, 334
CONNX Configuration Manager 334
CONNX Configuration Settings 20, 38
CONNX Data Dictionary 11, 14, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 111, 115, 117, 119, 122, 128, 135, 139, 145, 177, 237, 240, 244, 247, 250, 257, 261, 334
CONNX Data Dictionary Manager ..38, 108, 145, 177, 237, 334
CONNX Data Dictionary username 14
CONNX Data Synchronization 369
CONNX Import Table Selection window ..38, 145
CONNX Installation Guide 15
CONNX Integrated 145, 177
CONNX Integrated Logon window 145, 177

CONNX Logon Credentials 48, 108, 145, 177, 237
CONNX Message 6, 306, 334
port number 306
CONNX Message Queue 6, 15, 306, 334
CONNX Message Queue listening port 306
CONNX Solutions 18, 38, 48, 145, 165, 173, 177, 298, 369
CONNX User Reference Guide 108, 237, 316
CONNX.REPLICATION.CTRL_PORT 306
CONNX.REPLICATION.CTRL_PORT specifies 306
CONNX.REPLICATION.DEBUG_LEVEL 306
CONNX.REPLICATION.DEBUG_LEVEL controls 306
CONNX.REPLICATION.ENGINE_PORT 306
CONNX.REPLICATION.EP_ADA_CFG_PORT 306
CONNX.REPLICATION.EP_ADA_CFG_PORT specifies 306
port 306
CONNX.REPLICATION.INSTALLPATH 334
CONNX.REPLICATION.MQ_PORT 306
CONNX.REPLICATION.MQ_PORT specifies 306
CONNX.REPLICATION.NOTIFY 306
CONNX.REPLICATION.NOTIFY_CMD 306
CONNX.REPLICATION.NOTIFY_CMD specifies 306
CONNX.REPLICATION.STATUSREFRESHINTERVAL 306
CONNX.REPLICATION.STATUSREFRESHINTERVAL controls 306
CONNX/Replication 334
CONNX_EP_ADA_CFG_PORT 306
CONNX_EP_ADA_CFG_PORT specifies 306
port 306
CONNX_EP_DATA_PATH 20, 306
name 306
CONNX_EP_DATA_PATH specifies 306
CONNX_EP_DEBUG_LEVEL 306
CONNX_EP_DEBUG_LEVEL controls 306
CONNX_EP_NOTIFY 306
CONNX_EP_NOTIFY_CMD 306
CONNX_EP_NOTIFY_CMD specifies 306
CONNX_EP_QUEUE_PORT 306
CONNX_EP_QUEUE_PORT specifies 306
CONNX32/REPLICATION/COMMON/EVENTNOTIFY.BSH 306
CONNX32/REPLICATION/COMMON/EVENTNOTIFY.CMD 306
CONNX32/Replication/log 332, 367
Consistency 3
Contact Software AG 6
containing 20
ConnecX 20
Control Panel 15
point 15
Controller 1, 6, 11, 12, 15, 20, 48, 101, 111, 119, 128, 165, 173, 240, 244, 250, 294, 306, 330, 331, 332, 334, 367
Check 334
connect 334
EP 334
Replications Deployed 165, 173
use 306
Controller instructs 11
EC 11
Controller quiesces 11
Controller user/process 334
Controller/Event Consumer 15
controller's 334
copy 1, 3, 122, 135, 177, 247, 257, 368
Employees 177
table during 177
Copyright Infragistics 369
CPU 12
Create 1, 3, 11, 14, 38, 48, 65, 67, 74, 78, 85, 101, 115, 117, 119, 128, 145, 165, 173, 177, 244, 250, 290, 301, 316, 334, 366, 369
use CITY 177
Create CDD 213
create Controller 334
EP 334
Crystal Reports 369
CTRL 334
CTRL_OP_FLAGS 306

CTRL_PERF_ENABLED 306
CTRL_PERF_QUERY_INTERVAL.....306
CTRL_PERF_STAT_FILE 306
CTRL_PORT.....306
Custom.....15
customername,5,20 1
D
Daemon.....334
DaimlerChrysler 369
Darmstadt.....369
Data.... 1, 3, 6, 11, 14, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 115, 117, 119, 122, 135, 145, 177, 237, 244, 247, 257, 290, 298, 316, 334, 367, 369
  enterprise-wide view.....369
  Excluding 177
  Initial State duplicates.....122, 247
Data Access Corporation 369
Data Dictionary..... 6, 11, 48, 101, 145, 177, 298, 316, 367
data dictionary containing 6
  Adabas.....6
data dictionary ex1.cdd 367
data dictionary ex2.cdd 367
Data Dictionary window... 48, 101, 103, 145, 298
  Open.....101, 103
data during replication.....1
Data Filtering.....316
Data Link Properties window.....145
Data Link Provider.....145
data reporting 1
Data Type.....65, 67
Database.. 1, 3, 6, 11, 14, 38, 48, 108, 115, 119, 145, 165, 173, 177, 237, 244, 290, 334, 367
  open license 334
Database ID 38, 145, 334
database shutting.....6
database SQLTarget.....165, 173
database's.....3
databases containing 3
database–usually 3
DataFlex.....369
DB 334
DB ID 334
DB Name 334
DB2 .....1, 6, 20, 38, 369
DBG_OPTIONS.....306
DBID 145, 165, 173, 177
DBID1 145
  localhost 145
DBID1.dbo.ADABAS_FILE_11 173
DBID1.dbo.ADABAS_FILE_11_AIC.....173
DEBUG_LEVEL.....306
debugging/status.....306
Default Access 38, 145
default port number.....29
Define ADAREX.....20
Define CONNX_EP_DATA_PATH 20
Define CONNXREGISTRY 20
Delete.....3, 101, 165, 177, 234
  MIDDLE_I.....177
Delete config.....334
Delete Rep 48, 101
delete replication.....234
Delete Row 65, 74, 177
Deleting.....1, 3, 74, 85, 101, 117, 145, 165, 173, 298, 334
  Replication.....101
Deploy.....61, 65
Deploy button.....128, 250
  disable 128, 250
Deployed Replications .1, 48, 165, 173, 292, 302
Deployed Replications Tab.....292
Deploying 1, 6, 11, 15, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 115, 117, 119, 122, 128, 135, 145, 165, 173, 237, 244, 247, 250, 257, 271, 290, 292, 294, 302, 317, 332, 334, 367
  Event Replication 119, 244
Deployment.....117, 119, 165, 173, 244, 334
dequeue.....334
  Initial State.....334
dequeue EP 334
dequeue on Controller 334
dequeue Server Status 334
dereferenced.....334

Dharma....369
Dharma ODBC Integrator....369
Dharma Systems....369
Different Controller....128, 250
DIR....334
Disable Process....334
  Request....334
DISABLE_ROW_CT_CHECK....306
DISABLED....334
Disabling.... 128, 139, 250, 261
  Deploy button....128, 250
  Event Replicator....139, 261
DISCLAIMER....369
  SQLcrypt....369
Display.. 111, 135, 240, 257, 290, 292, 302, 332, 367
  Adjusting....302
DLL....6, 334
DNS....334
DNS_TO_DOT....306
Domanin/UserName....15
Done. 3, 11, 48, 65, 67, 74, 78, 80, 85, 101, 103, 108, 145, 177, 237, 332
done deleting replications....101
done troubleshooting....332
done viewing....103
don't want....3, 128, 250
Don'ts 3, 65, 78, 128, 250, 367
Dr Brian Gladman....369
Durability....3

E

East Windsor....369
EC....6, 11, 334
  Controller instructs....11
EC Configuration....334
EC reads....6, 11
  MQ....11
EC_EIS_WAIT....306
EC_EP_PORT....306
EC_EP_SERVER....306
EC_IGNORE_ZERO....306
EC_IS_TRUNCATE_ONLY....306
EC_SLEEP_ON_STARTUP....306
EC_SQL_ERR_MODE....306
EC_SQL_MAX_HANDLES....306
Edit Menu....273
Education....369
  People Oriented Information Systems....369
email....366
e-mail....368
Employees....177
  copy....177
EMPLOYEES table....177
Employees_Filtered....177
Employees_Filtered table....177
EMPLOYEES_INCOME table....177
Enabling
  Event Replicator....48
encounters....306, 334
  Nx....306
End Transaction....334
Engine.....1, 6, 11, 12, 14, 18, 20, 38, 145, 165, 173, 177, 316, 334, 368
  Check....334
Engine 1.1.2or....173
Engine Adaptor....6
engines depending....11
enqueue....334
  Initial State....334
enqueue Engine....334
enqueue EP....334
enqueue init....334
Enter .11, 15, 18, 20, 48, 67, 108, 145, 177, 237, 301, 334
  TCP/IP....18
Enter AREA_CODE....177
enterprise-wide....369
enterprise-wide view....369
  data....369
EP....6, 11, 334
  cache....334
  connect....334
  Controller....334
  create Controller....334

open Controller 334
Replication Administrator....6
EP_ADA_CFG_PORT 306
ep_config....306, 334
EP_EVENT_ERR_MODE 306
EP_QUEUE_HOST....306
ERR....334
Error Action 334
Error Action Code....334
Error Class 334
Error Code....334
Error State....334
Error State Code 334
ERRORS....334
ET_SYNC_WAIT....306
ETL....1
Europe's 369
euros 369
Event Consumer 1, 6, 20, 306, 334, 367
Check....334
stopping 6
Event Producer 1, 6, 11, 12, 15, 20, 119, 244, 306, 334, 367
Event Producer user/process....334
Event Replication . 1, 3, 6, 11, 15, 18, 20, 38, 48, 61, 74, 80, 85, 103, 119, 135, 139, 144, 145, 165, 173, 177, 244, 257, 261, 289, 298, 306, 332, 334, 367
Deploying....119, 244
Event Replication Administrator....1, 11, 334
change....11
Event Replication Controller....1, 48
Event Replication Designer....6
Event Replication Designer requires....6
Event Replication Engine....1
Event Replication Error Messages....334
Event Replication Registry Settings....306
Event Replication Server....20
Event Replication Server contains....20
Event Replication Troubleshooting Do's....367
Event Replication Troubleshooting Suggestions 332
Event Replication using....38
Event Replicator....1, 3, 6, 11, 14, 15, 18, 20, 37, 38, 48, 61, 65, 67, 74, 78, 85, 115, 117, 122, 135, 139, 144, 145, 165, 173, 177, 247, 257, 261, 289, 290, 298, 328, 329, 330, 331, 334, 366, 367, 369
Disabling....139, 261
Enabling 48
Installing 15
Uninstalling....37
Event Replicator Administrator .1, 11, 15, 18, 20, 38
Windows....18, 20
Event Replicator Architecture 6
Event Replicator assumes 11
Event Replicator calls 366
NOTIFY_CMD 366
Event Replicator Components 18
Replicator Administrator....18
Event Replicator Controller....14
Event Replicator Error Message....334
Event Replicator Installation 15, 37
Event Replicator license 6, 15
License Server 15
Event Replicator needs....38, 367
Event Replicator Tool 15, 37
Event Replicator Tutorials....144
Event Replicator User Exits....366
eventnotify....306, 366
eventnotify.cmd file 306, 366
eventserver 330, 331
ex1.cdd 367
ex2.cdd 367
Excluding 177
Data....177
Exec....334
initialize....334
Existing 1, 284, 334
config....334
Server Status....334
SQL Gateway....1
Target....284
Exit Setup window....37

export/home/cnxuser/connx/replicator/cnxep.env 20
Expressions....11, 177, 301
Extensible Markup Language....369
Extra....369
F
FAR 52.227-19....369
FATAL ERROR....334
FDT Import....38
File menu.... 38, 48, 61, 101, 145, 177, 271
File11.... 145, 165, 173, 177
Filtering Replication Data....177
Finish....3, 15, 37
Firewall....29
FIRST_NAME....177
FIRST_NAME rows....177
FIRST_NAME+MIDDLE_I+LAST_NAME....301
FOCUS....369
forget. 1
FTL....369
FULL_NAME....177, 301
G
GB....12
GB Free Space....12
GB RAM....12
General Features....316
General Messages....334
Germany....369
GHz....12
GNU General Public License....369
terms....369
Government Restricted Rights Notice....369
GPL....369
GPL apply INSTEAD OF....369
GUI....1
GUI-based....1
H
healthcare....369
Help Menu....289
Hewlett-Packard Company....369
Hide Schema....103
higher/Suse 10/Linux 390....12
HPUX....12
HP-UX....12
HP-UX....369
I
IBI....369
IBM AIX5L Version 5.2....12
IBM Communications Server....369
IBM eNetwork Communications Server....369
IBM eNetwork Personal Communications....369
IBM Personal Communications....369
ID....334
ID>_<engine ID....334
IDE....12
Iinstallation Wizard....37
Import....1, 14, 38, 115, 145
Import CDD....38, 145
Import CDD window....38, 145
Import Type....38, 145
Impromptu....369
Inc....369
including....369
Lufthansa....369
Incompatible API....334
Index....65, 334
InfoNaut....369
InfoNaut Professional....369
Informational....334
Informix....1, 6, 14, 20, 38
Infragistics....369
Initial State....3, 11, 38, 122, 125, 135, 145, 247, 257, 292, 334
dequeue....334
enqueue....334
Performing....122, 247
Initial State duplicates....122, 247
data....122, 247
initial state processing....11, 334
Initial State replaces....122, 247
Initial State. 3
initialize....334
Exec....334
initialize ep_config....334

Initiates 1
Inprise Corporation 369
INSERT/SELECT SQL 11
Install 18, 119, 177, 244, 328, 332
Install ConnecX 145, 165, 173, 177
Install window 15
  Ready 15
Installation Directory. 18, 20, 328, 329, 330, 331, 334, 367
Installation Location 18
installeventrep 20
Installing ... 6, 14, 15, 18, 20, 145, 165, 173, 330, 332, 334
  Event Replicator 15
  UNIX 18
Installing JMS Server 26
INSTALLPATH 334
InstallShield Wizard 15, 37
  start 15, 37
int 334
Internal API 334
International Business Machines Corporation 369
Internet 369
Introduction 1
Invalid Server Status 334
Invalid user/pass 334
IO Error 334
IP 48, 108, 237
IP address 48, 108, 237
Isolation 3
IT 369
It's 1

J

JMS 26, 54, 58
July 369
June 1987 369

L

L/N 334
L/N1 334
L/N2 334
L/N3 334
L/N4 334
L/Nx 334
LAN 12
leave 145, 177, 302
  Server Status window 302
LEFT 177
Legal Information 369
Length 67, 80, 85, 177, 334
length exceeds 334
License Administrator 15
License Agreement window 15
License Server 15, 334
  Event Replicator license 15
LICENSE TERMS 369
  SQLcrypt 369
Linux 6, 12
LN/3 334
Local Services list 15
localhost 38, 145, 177
  DBID1 145
Log Files ..20, 111, 139, 240, 261, 306, 332, 334
  See 332
logfile 334
Logical DB 334
login 108, 237
logon 38, 48, 108, 145, 177, 237
Logon Credentials 48, 108, 177, 237
Logon Credentials enter 145
Lufthansa 369
  including 369

M

machine/PC 12
Manual 1, 111, 240, 279, 302, 369
Manual Column Width 279, 302
Map .1, 11, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 103, 177, 284, 301, 317, 334
Map Columns .65, 67, 74, 78, 80, 85, 91, 94, 177
Map Columns window .65, 67, 74, 78, 80, 85, 91, 94, 177
map existing 334
map existing config 334
Mapping 65
  columns 65

Mappings 67, 74, 80, 85, 91, 94
MAR_STAT 177
Marital_Status 177
Mark 101, 119, 244, 317, 369
Math Operations 177
  Performing 177
MAX ADABAS File 38, 145
MB RAM 12
MB/sec 12
Merck 369
Message ID 334
Message Queue ... 1, 3, 6, 11, 15, 108, 119, 237, 244, 328, 334
Message Type 334
metadata 369
MHz CPU 12
Microsoft 135, 145, 165, 257, 369
Microsoft Access 369
Microsoft Corporation 369
Microsoft Excel 369
Microsoft Internet Information Server 369
Microsoft Management Console 135, 257
Microsoft provides 145, 165
Microsoft SQL Server 369
Microsoft Visual 369
Microsoft Visual Basic 369
Microsoft Visual FoxPro 369
Microsoft Visual Studio 369
Microsoft Windows 369
MIDDLE_I 177
  Delete 177
mis-matched 334
mode 1
Modify .. 37, 48, 61, 80, 85, 91, 94, 306, 334, 367
MQ 11, 48, 108, 145, 177, 237
  EC reads 11
MQ_CONFIG_DIR 306
MQ_PORT 306
MQ_QUEUE_DIR 306
mqserver 328, 329
msg 334
MU/PE column 334
Multiple Data 177, 367
  Concatenating 177
MySQL 1

N

N4 334
NAME 177, 306
  CONNX_EP_DATA_PATH 306
Name column 177
Name/Address 48, 108, 145, 177, 237
NC 369
needs 1, 3, 38, 48, 65, 144, 332, 367
  today's 1
NET 366, 369
NET 2003 366
NetAdvantage 369
NetManage 369
New ... 3, 6, 11, 38, 65, 67, 74, 78, 115, 117, 128, 145, 165, 177, 250, 334
  Building 115
new Replication 11, 165, 334
New Table Name 177
new Target Table 38, 65, 67, 74, 78
Next 15, 37, 101, 145
NJ 08520 USA 369
non-Adabas 38, 67, 74, 301
Non-MU/PE column 334
non-Windows 20, 38
NORMAL 334
North Carolina 27513 369
Not Replicating 332
NOTIFY 306
NOTIFY_CMD 306, 366
  Event Replicator calls 366
Nucleus Response 334
NULL 334
NULL source 334
Number 177, 306, 332, 334
Nx 306, 334, 366
  Action type 366
  encounters 306

O

ODBC 1

OK . 15, 38, 48, 61, 108, 145, 165, 173, 177, 237
OLEDB 1
OLEDB Driver 1
Adabas....1
OLEDBC Provider 38, 145
online....3, 6, 369
Online Help 369
Open... 1, 3, 6, 11, 14, 15, 18, 20, 37, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 111, 115, 117, 119, 122, 128, 135, 139, 145, 165, 173, 177, 237, 240, 244, 247, 250, 257, 261, 271, 289, 290, 298, 306, 332, 334, 366, 369
Data Dictionary window....65, 103
open Admin 334
open Controller....334
EP....334
Open Deployed CDD 271
open EC 334
open EP....334
open EP Identity....334
open existing....334
open existing config 334
open license....334
database....334
open Replicator....334
Open Systems 1, 3, 6, 11, 15, 18, 20, 37, 38, 48, 61, 65, 74, 78, 80, 85, 103, 119, 145, 165, 173, 177, 244, 289, 290, 298, 306, 334, 366, 369
Open Systems Even Replicator....3
Open Systems Event Replication window65, 103
Open Systems Event Replicator....119, 244
use....65
Operating....12
System....12
Operating System list....18
Operating Systems....12, 18
Options....1, 15, 20, 369
Oracle.... 1, 6, 14, 20, 38, 108, 237
Oracle, SQL Server....1, 6
OS 334
OS Command 334
OS/2 369
OS/400....369
OS-specific....334
P
PA RISC....12
Passed Validation 165
Password 14, 15, 18, 48, 108, 145, 177, 237
PC 367
People Oriented Information....369
People Oriented Information Systems....369
Education 369
performance statistics....306
Performing 122, 145, 177, 247
Initial State....122, 247
Math Operations....177
Simple 145
Performs 1
PID 334
platform-specific....14
Plus2000 369
point 3, 15
Control Panel....15
timethe....3
Port 11, 29, 48, 108, 145, 177, 237, 306, 334
change CONNX.REPLICATION.ENGINE_PORT 306
CONNX.REPLICATION.EP_ADA_CFG_PORT specifies....306
CONNX_EP_ADA_CFG_PORT specifies . 306
port number....29, 48, 108, 237, 306, 334
all....108, 237
CONNX Message....306
portion 14, 177
ConnecX....14
Postgres....1
POWERflex....369
POWERflex Corporation....369
Precision 65, 67, 80, 85
Preparing Setup window....15, 37
Prerequisites....14, 145, 165, 173, 177
preserve 3
ACID....3

atomicity 3
Press Deploy....11
Press Enter 20
Primary Key....334
Problem Resolution....3
Programs 1, 18, 38, 48, 145, 165, 173, 177, 298, 366

Q

QUEUE....334

R

Read Only 38, 145
read/write....20, 369
Ready 3, 15, 20, 38, 48, 317, 332
Install window 15
Recommended System....12
Recreating....117
Red Hat 12
Redeploying 165
Redmond....369
Refer....334
User Guide 334
Refresh.... 111, 240, 306, 334
Refresh Server Status....334
Registry 306, 334
Registry Variables 306
Relational 1, 3, 145, 165, 173, 177
Relational Database....1, 145
Relational Replication 145, 165, 173, 177
Relational Replication.cdd....145
Release 7.00 369
Remove 3, 37, 74, 101, 103, 128, 135, 173, 250, 257, 334, 367
Rename....145, 177
Rename Object window 145
Renumbering Target Files....229
Rep....165, 173, 290
Replicating 111, 145, 165, 240, 294
Replication 1, 3, 6, 11, 14, 15, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 111, 115, 117, 119, 128, 135, 139, 144, 145, 165, 173, 177, 237, 240, 244, 250, 257, 261, 271, 273, 284, 290, 292, 294, 298, 302, 306, 316, 317, 330, 331, 332, 334, 367
forget. 1
Tables depending....284
Replication Administrator..1, 6, 11, 48, 145, 165, 173, 177, 292, 294, 298, 317, 332, 334
EP....6
writeable 334
Replication Administrator matches 334
Replication Controller15, 48, 108, 119, 135, 145, 177, 237, 244, 257, 306, 330, 331, 332, 367
Replication Data....177
Replication Design 101, 177, 290, 292, 294
Replication Design Tab..101, 177, 290, 292, 294
Replication Designer....1
Replication Display 302
Replication Engine 1, 306, 334
Replication Log Directory....139, 261
Replication Overviews 11
replication server ID 334
Replication Server Map 334
Replication Servers.....1, 11, 48, 108, 119, 135, 139, 145, 165, 237, 244, 257, 261, 332, 334
Replication Work Queue 334
Replications Deployed 165, 173
Controller....165, 173
Replications Tab 173, 292, 302
replications want 316
Replicator....3, 18, 334, 367
connect....334
Replicator Administrator 18, 367
Event Replicator Components 18
request....3, 119, 122, 244, 247, 306, 334
DISABLE PROCESS 334
respawn....334
Required System 12
Reserved....306, 369
respawn 334
request 334
respawned 334
Restart Servers....139, 261, 287
Reston....369
Restricted Rights....369
return....37, 292, 294, 334

REX 14
RL 334
RL Pool 334
Row Down 67, 78
Row Up 78
RUMBA 369
running 20
ADABAS Nucleus 20
S
SALARY 177
SALARY_PERCENT 177
Sample 145, 165, 173, 177, 306, 366
SAS Campus Drive 369
SAS Institute 369
SAS Institute Inc 369
SAS/C OnlineDocTM 369
SAS® 369
Save 1, 38, 61, 65, 101, 135, 145, 177, 257, 271, 334
Save As 38, 145
Scale 65, 67, 80, 85
SCO 369
SCSI 12
SCT 369
Seagate Software 369
Security menu 38, 145
See 3, 15, 61, 108, 115, 139, 145, 165, 237, 261, 279, 294, 316, 332, 334, 366, 367
Log Files 332
User Guide 334
See OS 334
Select . 11, 15, 20, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 117, 122, 128, 145, 165, 173, 177, 237, 247, 250, 290, 292, 294, 298, 301, 302
CDD 65, 103
Target Table 65
Select ADABAS_FILE_11 145
select ADABAS_FILE_11_AIC 173
Select Administrative Tools 15
select Auto Column Width 302
Select Data Filtering.cdd 177
select DBID1.dbo.ADABAS_FILE_11 145
Select DBID1.dbo.ADABAS_FILE_11_AIC ... 165
select DBID1.dbo.EMPLOYEES 177
select DBID1.dbo.Employees_Filtered 177
select DBID1.dbo.EMPLOYEES_INCOME ... 177
select FDT Import 145
Select Features window 15
Select Microsoft OLEDB Provider 145
SQL Server 145
select OpenCDD 177
Select Provider 38, 145
Select Provider Type 38, 145
select SQLDatabase 177
select SQLTarget 145
Select Target 117, 122, 145, 165, 177, 247
Select Target Databases 165
Selective un-deploy 131, 253
sendmail program 366
Server 1, 3, 6, 12, 14, 15, 18, 20, 38, 48, 108, 111, 119, 135, 139, 145, 165, 177, 237, 240, 244, 257, 261, 287, 294, 302, 306, 332, 334, 369
Server database 145, 165, 173, 177
server name 38, 48, 108, 145, 237, 334
server portion 14
ConnecX 14
Server Status .. 15, 111, 135, 139, 145, 165, 240, 257, 261, 294, 302, 306, 332, 334
existing 334
show 302
Viewing 111, 240
Server Status Tab 294
Server Status window 302
leave 302
Servers Menu 287
Service Account Selection window 15
Services 1, 15, 128, 135, 250, 257, 332, 334, 367, 369
services enabling 369
ServLine24 368
Access 368
set 1, 38, 48, 108, 145, 237, 273, 302, 334, 366, 369
Set Port 108, 237

Setting up webMethods Queues 54
Settings 15, 20, 38, 306
Setup 15
Setup Status window 15
Setup Type 15
show 3, 38, 103, 177, 294, 302, 332
  Server Status 302
Show Schema 103
Show/Hide Schema 279
Showing 103
  Source Table Schema 103
Shutdown 334
Siemens 369
Simple 145, 369
  Performing 145
SIS 369
Software AG 368, 369
Software AG Account Manager 368
Software AG and/or all Software AG 369
Software AG continues 369
Software AG headquartered 369
Software AG Regional Support Center 368
Software AG´s ServLine24 368
Software AG's 369
Solaris 12, 369
Solutions. 3, 18, 38, 48, 145, 165, 173, 177, 298, 369
Solutions offers 369
SolutionsIQ 369
Sony 369
source 1
Source 1, 3, 6, 11, 14, 15, 20, 37, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 115, 117, 119, 122, 135, 139, 145, 165, 173, 177, 237, 244, 247, 257, 261, 290, 292, 301, 334, 367, 369
  CDD containing 145, 165, 173, 177
  change 65
Source Column .. 11, 65, 67, 80, 85, 91, 94, 177, 301, 334
Source Column enter 177
  SQL 177
Source Column Mapping 80, 91
source column name 80, 301
Source Database 1, 3, 20, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 103, 108, 115, 117, 119, 122, 135, 139, 145, 165, 237, 244, 247, 257, 261, 290, 334
  add 108, 237
  set 119, 244
Source Database Servers 20, 108, 237
source database starts 3
source DBID/target database 11
Source Table 1, 38, 48, 61, 65, 67, 74, 78, 85, 103, 122, 145, 165, 247, 290, 334
source table metadata 65, 67
Source Table Schema 103
  Showing 103
Spawned ep_config 334
Split 177
  SQL Expressions 177
SQL 1, 6, 11, 12, 14, 18, 20, 38, 65, 67, 145, 165, 173, 177, 301, 316, 334, 368, 369
  Source Column enter 177
SQL Expressions 67, 177
  Split 177
SQL Gateway 1, 6, 11
SQL Gateway Adaptor 6
SQL Gateway Target Adaptor 6, 11
SQL Server 1, 14, 20, 38, 145, 165, 173, 177
  Connecting 20, 38
  Select Microsoft OLEDB Provider 145
SQL Server database 145, 165, 173, 177
SQL Server Express Edition 145, 165
SQL Server Integration Services 1
SQL View Clause 11, 177, 316
SQL View Clause Text Box 316
SQL-based 1
SQLcrypt 369
  DISCLAIMER 369
  LICENSE TERMS 369
SQLDatabase 177
SqlRegistry tool 334
SQLTarget 145, 165
SSIS 1
standards-based 369

start .. 3, 6, 11, 15, 18, 20, 37, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 103, 108, 115, 117, 119, 122, 139, 145, 165, 173, 177, 237, 244, 247, 261, 298, 328, 330, 332, 334
InstallShield Wizard 15, 37
UNIX 328, 330
UNIX Message Queue.....328
Start button.....15
Start menu.. 18, 38, 48, 103, 145, 165, 173, 177, 298
start replicating data... 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 103, 108, 115, 117, 122, 237, 247
state.....6, 11
State Description..... 111, 145, 165, 240
StateDescription column 111, 240, 294
states.....6
statistics.....306
Status ... 3, 15, 37, 111, 119, 135, 139, 145, 165, 240, 244, 257, 261, 294, 302, 306, 332, 334
Status Message 334
Status Received 111, 240, 294
Status Tab.... 135, 139, 145, 165, 257, 261, 294, 302, 334
STATUSREFRESHINTERVAL 306
Stop Server 334
Stop Targets..... 135, 257, 287, 334
stopping.....6, 329, 331
Event Consumer.....6
UNIX 329, 331
UNIX Message Queue.....329
subject.....369
Agreement.....369
substring.....1
Suite 150 369
Sun 369
Sun Microsystems.....369
SUN Solaris.....12
support 334, 368, 369
XML 369
Support Contacts 368
Support DLL 334
Suppressing an Initial State 125
Sybase 1, 6, 14, 20, 38
synchronization.....1
System Requirements 12
system running.....14
System Unknown 111, 240, 294, 332
Systems 1, 3, 6, 11, 12, 14, 15, 18, 20, 37, 38, 48, 61, 65, 74, 78, 80, 85, 103, 111, 145, 165, 173, 177, 240, 289, 290, 294, 298, 306, 332, 334, 366, 369
Operating.....12
Systems Even Replicator.....3

T

table ..1, 11, 20, 38, 48, 61, 65, 67, 78, 103, 108, 115, 117, 122, 135, 145, 165, 173, 177, 237, 247, 257, 284, 290, 292, 298, 316, 334, 367
Adding 61
table DBID1.dbo.ADABAS_FILE_11 .....165, 173
table during 177
copying 177
table EMPLOYEES-NAT 145, 165, 173, 177
table ID.....334
Table Properties.....177
Tables depending 284
replication 284
Tables Menu 284
target.....1
Target..1, 3, 6, 11, 14, 20, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 101, 103, 108, 115, 117, 119, 122, 128, 135, 139, 145, 165, 173, 177, 237, 244, 247, 250, 257, 261, 284, 290, 292, 301, 317, 334, 367
existing 284
target account 3
Target Adaptor 6, 11
Target Column 11, 65, 67, 80, 85, 91, 94, 177, 301
target column Data Type select CHAR.....177
target column Data Type select NUMERIC ...177
target column Length 177
target column mapping 11, 85, 94
target column MAR_STAT.....177
target column name 67, 85
target column Prec.....177
Target Data Type 65
target database needed.....38

Target Databases. 1, 3, 6, 11, 14, 20, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 103, 108, 115, 117, 122, 128, 135, 139, 145, 165, 173, 237, 247, 250, 257, 261, 290, 367
Target DB 334
connect 334
Target Group 145, 165
Target Length 65
Target Table ... 1, 3, 6, 48, 65, 67, 74, 78, 80, 85, 91, 94, 101, 115, 117, 165, 290, 317, 334, 367
create 48
Target Table contain 290
target table ID 334
target table metadata 65, 67
Target tables 38, 61, 65, 177, 317
TBL 334
TCP/IP 18, 334, 367
Enter 18
tcpip port number 29
Technical Support 332, 334
telecommunications 369
Telefonica 369
terminates 37, 334
uninstall 37
terms 369
GNU General Public License 369
Test Connection 48, 108, 145, 177, 237
The Net Memetic Pte Ltd 369
The Open Group 369
Time column 111, 240
Timeout 334
Timeout sending 334
Timeout waiting 334
timethe 3
point 3
tnsnames.ora 14
today's 1
needs 1
Toll Free 369
transaction-oriented 3
TX 334
TXID 334

U

U.S. 369
UK 369
UnDeploy 292
Un-Deploy 131, 253
Un-Deploy all 128, 250
UnDeploy All 292
Un-Deploying 128, 250
Under Populate columns 80, 85, 91, 94
Uninitialized 334
uninstall 37
completes 37
terminates 37
Uninstall Complete window 37
Uninstall Setup 37
Uninstall Setup Status window 37
Uninstall Verification window 37
Uninstall Verification window opens 37
Uninstalling 37
Event Replicator 37
United States 369
America 369
United States and/or 369
University 369
California Berkeley 369
UNIX ...6, 14, 15, 18, 20, 38, 328, 329, 330, 331, 332, 369
Installing 18
start 328, 330
stop 329
Stopping 331
UNIX Installer 18
UNIX Machine 18, 20, 328, 330
UNIX Message Queue 328, 329
Starting 328
Stopping 329
Unix servers 6
UNIX Setup window 18
Unix/Linux 306, 334
Unix/Linux Environment Variables 306
Unknown 332, 334
Unknown EP 334

Unrecognized ADABAS table....334
Unsupported backend....334
USA....369
use 1, 3, 6, 11, 12, 14, 20, 37, 38, 48, 65, 67, 74, 78, 85, 108, 128, 145, 165, 173, 177, 237, 250, 306, 316, 334, 369
  Controller....306
  Open Systems Event Replicator....65
use CITY....177
  create....177
use localhost....38, 145
use OLEDB....14
Use Row Up....67
User Guide....20, 38, 289, 334
  Refer....334
  See....334
User ID....18, 145, 177, 334
User Name....48, 108, 237
UserName....145, 177
Uses ConnecX....1
Uses SQL Gateway....1
Using....1, 6, 11, 20, 38, 48, 65, 67, 78, 119, 144, 145, 165, 173, 177, 244, 290, 316, 367, 369

V

Va....369
Valid CONNX.REPLICATION.ENGINE_PORT....306
Valid CONNX.REPLICATION.STATUSREFRESHINTERVAL....306
Validate....61, 165, 173, 317, 334
  Active Servers....317
Validate Active....61, 165, 317
Validation....317
Validation fails....317
validation succeeds....317
Values L/N1....334
Values N1....334
VB....366
View Clause Text Box....316
View Menu....103, 279, 302
Viewing....111, 240
  Server Status....111, 240

W

WA 98052....369
Wall Data....369
WAN....12
WARNINGS....334
Washington....369
Web....369
webMethods....54
Welcome window....15, 37
Why....3
Width....302
window containing....38, 145
Windows6, 12, 14, 15, 18, 20, 37, 38, 48, 61, 65, 67, 74, 78, 80, 85, 91, 94, 103, 108, 119, 145, 165, 173, 177, 237, 244, 289, 298, 302, 334, 369
  Event Replicator Administrator....18, 20
Windows 2000/2003/XP....12
Windows 2000/2003/XP/Vista....12
Windows Environment....6, 15
Windows NT....12
Windows NT Advanced Server....12
Windsor Corporate Park....369
Wizard Complete window....15, 37
Worcester....369
Working....108, 237
writeable....334
  Admin....334
  Replication Administrator....334

X

xf inst_eventrep_sol_64....20
XML....58, 369
  support....369
XML simplifies....369

Y

Yes....14, 37, 74, 101, 128, 177, 250

Z

z/OS....369