

TIME **X** TENDER

USER GUIDE



TIMEXTENDER

TimeXtender® User Guide

Version 2020-11-11

Find the latest version at

<https://support.timextender.com>

Copyright

© 2020 TimeXtender A/S. All Rights Reserved.

Trademarks

Microsoft®, Windows® and other names of Microsoft products are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other product names mentioned in this documentation may be trademarks or registered trademarks of their respective companies.

Contents

Contents	3
Introducing TimeXtender	6
The User Interface	7
Setting Up Your Project	10
Project Repositories	11
Projects	13
Data Warehouses	17
Business Units and Staging Databases	24
Team Development	26
Project Perspectives	32
Using the ODX	35
Installation	37
Data Storage	39
Data Sources	43
Data Selection and Filtering	52
Tasks	55
Security	58
Execution Queue, Logs and Statistics	60
Notifications on Critical Errors	63
Connecting to Data Sources	65
AnySource Data Source	75
CData Data Source	77
Custom Data Source	79
IBM DB2 Data Source	80
IBM Informix Data Source	81
Infor M3 (Movex) Adapter	82
Microsoft Dynamics AX Adapter	84
Microsoft Dynamics CRM Online Adapter	92

Microsoft Dynamics GP Adapter	94
Microsoft Dynamics Business Central (NAV) Adapter	96
Microsoft Excel Data Source	102
Microsoft SQL Server Data Source	103
ODBC Data Source	105
Oracle Database Data Source	107
Oracle MySQL Data Source	109
Salesforce Adapter	110
SAP Table Adapter	112
Sun System adapter	114
UNIT4 Business World (Agresso) Adapter	115
Text File Data Source	123
Designing the Data Warehouse	126
Selecting, Copying and Relating Tables	127
Selecting, Validating and Transforming Data	139
Previewing Data	146
Tables	150
Fields	165
Views	177
Indexes	182
History	184
Scripting	187
Database Schemas	200
Data Security	202
Documentation	207
Visualization	211
Performance Recommendations	213
Building Semantic Models	216
Tables	218
Fields and Measures	222

Relations	230
Security	232
Perspectives	236
Endpoints	237
Deployment and Execution	242
Building SSAS Multidimensional Cubes	244
Cubes	246
Dimensions	253
Measures	262
Handling Early Arriving Facts	266
Slowly Changing Dimensions	271
Building Qlik Models	277
Data Export	289
Deploying and Executing	292
Manual Deployment and Execution	294
Scheduled Execution	303
Incremental Loading	312
Multiple Environments	320

Introducing TimeXtender

The purpose of TimeXtender is to enable you to create and maintain a complete data warehouse solution with as little effort as possible. To achieve this, TimeXtender employs Data Warehouse Automation techniques to automate the tedious parts of the work.

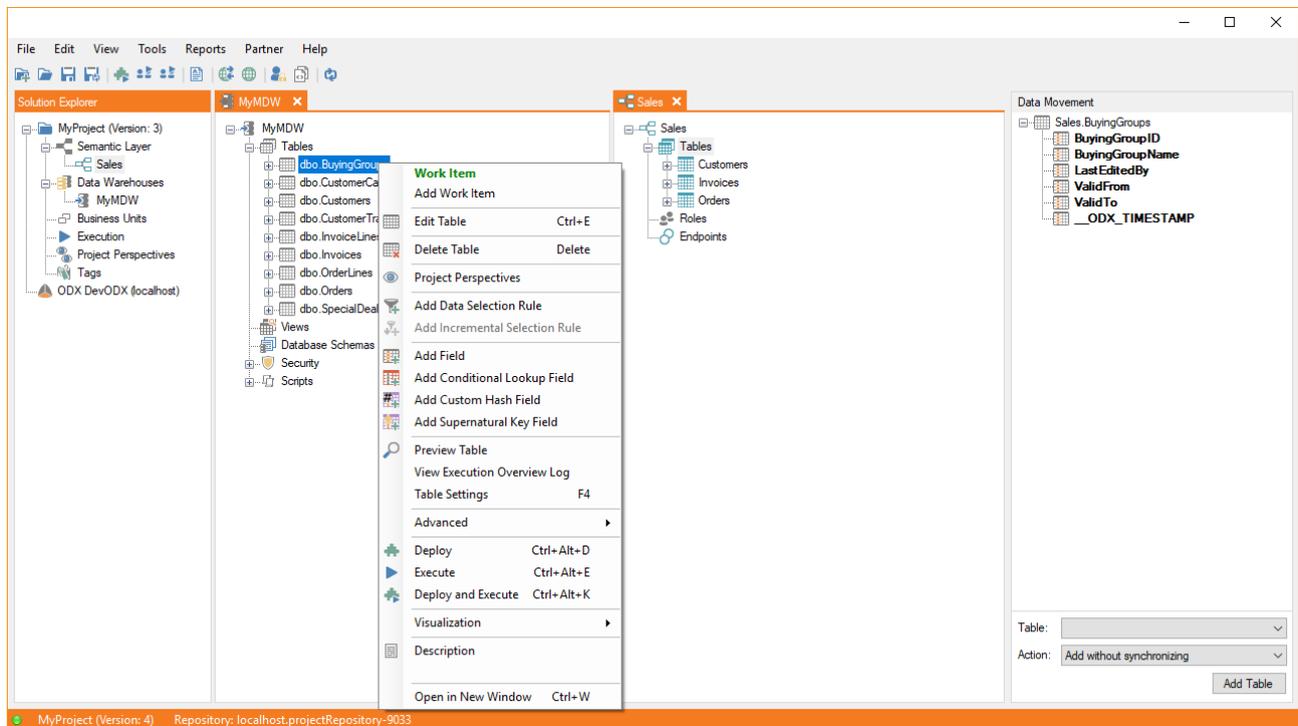
SQL code for the ETL (extract, transform, load) process, MDX code for SSAS Multidimensional cubes, indexes etc. are generated automatically. Most tasks can be accomplished using drag-and-drop in the graphical user interface, and the amount of code you need to write is minimized. However, if and when you need to customize the code, you can do it in your favorite development environment.

This user guide outlines the features and functionality of TimeXtender. Please note that this document does not cover detailed concepts of data modeling or the tables and fields of any particular relational database. If you need help with this, please explore our training options.

This section provides an overview of user interface you will be working with when you build a data warehouse in TimeXtender. Installation instructions for the software is provided on our support site at <https://support.timextender.com>

The User Interface

The TimeXtender user interface is a standard Windows application with a menu bar, a toolbar and a number of tabs that contain the different objects you will be working with. This section provides an overview of the interface.



The Menu

The commands, settings and tools available in TimeXtender are split between two menu locations. The general commands, such as new, open, find, refresh, options, help, can be found in the menu bar. Commands related to a specific object can be found in the shortcut menu that appears when you right click the object.

For easy access to the most used commands in the menu bar, a subset of the commands can also be found in the toolbar just under the menu bar.

A lot of the most used commands have keyboard shortcuts. On all menus, you will notice that the keyboard shortcut to a command is displayed to the right of the command in the menu.

The View menu

This menu contains all the commands related to how the project is displayed.

Almost any object in TimeXtender can be opened in a new window by right-clicking it and clicking **Open in New Window**. On the View menu, the open windows are listed under Open Windows to give you an easy overview if you find yourself with a lot of open windows.

On the View menu you can also customize the user interface by toggling the display of different elements:

- **Data types:** When checked, the data type of fields is postfixed to their name.
- **Deprecated features:** When checked, deprecated features are shown in the user interface. We recommend that you to keep this disabled for a cleaner interface.
- **Highlight descriptions:** When checked, objects that have a description are displayed with bold text in the tree. Descriptions are added from the shortcut menu on an object.

The Reports Menu

In the Reports menu, you can find items related to previous and future executions:

- Validation errors and warnings, i.e. rows of data that have violated a validation rule on a field or table.
- Execution queue log
- ODX execution and service log
- ODX task schedule

The Solution Explorer and tabs

In the Solution Explorer, all the top-level objects in your solution are listed, such as data warehouses and semantic models. Double-click an object, or click **Open** in the shortcut menu, to open the object in a new tab in a new tab group.

You can have as many tabs and tab groups as you like - or rather, as many as you can fit on your screen! When you save and close a project, the tabs you have open in are saved as a user setting. The next time you open the project, the same tabs will be available.

For rearranging tabs, you can use drag and drop or shortcut menu on tab header. You can move tabs between tab groups, and reorder tabs within tab groups. Tab groups can be moved from the shortcut menu on the tabs in the group.

You can drag and drop tables and fields between tabs in different tab groups, e.g. from a data warehouse to a semantic model.

Go to Source Table/Field

To ease navigation between tabs, a "shortcut" is available. The **Go to Source** command selects and displays the source table/field to make it easy to track where a specific table/field comes from.

Go to Source is available on the following objects:

- Tables on semantic models, data exports and Qlik models
- Mappings under tables on data warehouses
- 'Copy' mappings under fields on data warehouses

If the source table/field exists in an open tab in another tab group, the source table/field will be selected there. If not, a new tab in a new tab group will open to show the source table/field.

To go to a source table/field

- Right click the table, field or table mapping and click **Go to Source Table/Field**.

Note: Since tables on the ODX data storage are not listed in the UI, **Go to Source** is not available if the source is a table/field on the ODX data storage.

Color Theme

If you would like to easily tell different installations of TimeXtender apart, or if you are simply tired of the default colors, you can choose another color theme for the application. The color theme is saved in the user settings for the specific version of TimeXtender.

A color theme in TimeXtender consists of a theme - light or dark - in conjunction with an accent color. There are five predefined accent colors available in TimeXtender, giving you a total of ten color themes.

To change the color theme

- On the **Tools** menu, click **Options** and then click your preferred theme in the **Theme** list and your preferred accent color in the **Accent Color** list.

If you use [multiple environments](#), you can select a color theme for each environment to make them easier to tell apart. The color themes available for environments are different from the regular color themes to make sure they stand out when you apply them.

The Status Bar

The status bar in the bottom of the window contains information about the application and the currently open project, if any, and serves as a useful shortcut to some settings. The status bar can contain the following items, depending on configuration:

- The name of the currently open project with the version number in parenthesis. In front of the project name, an icon shows you if all changes have been saved (green) or there are changes waiting to be saved (red). Click this to open the **Open Project** window.
- The name of the repository server and database. Click this to open the **Options** window on the **Project repository** tab.
- The name of the current environment if any. Click this to open the **Environment properties** window.
- The license type and what company it is registered to. Click this to open the **License information** window.

Setting Up Your Project

In TimeXtender, your work is organized in projects stored in a repository on a SQL Server. Each project is basically a collection of data warehouses, tables, fields, business units, execution packages, semantic models, etc.

In this chapter, we will cover the objects found in most projects: the project itself, data warehouses, business units and staging databases. Each project can contain any number of data warehouse databases or business units. In turn, each business unit contains a staging database and a number of data sources.

Missing from the list above are data sources. They are covered in the chapter [Data Sources](#).

Note: If you use an [ODX server](#), data will be copied directly from the ODX server to a data warehouse. Consequently, you won't be using business units, including the staging database and the data sources that make up a business unit.

Project Repositories

Your projects are stored in a project repository on a SQL Server or Azure SQL Database. When you open TimeXtender for the first time, you can set up a repository in the Get Started wizard. If you do not, you will be prompted to set up a repository when you try to e.g. add a new project.

When you install a new version of TimeXtender, you will be prompted to run an upgrade script that automatically updates the repository to ensure compatibility with the new software version.

Note: The repository settings are saved for the current user and installation of TimeXtender. This means that if you open two instances of TimeXtender and change the repository settings in one of them, you will change the repository settings for both.

Setting up a Project Repository

To change the repository settings, follow the steps below:

1. On the **Tools** menu, click **Options** and then click the **Project repository** tab in the window that appears.
2. In the **Server name** box, enter the name of the database server on which you want to store the project. Click the ellipsis (...) to choose one of the available servers in your local Active Directory, if any.
3. In the **Authentication** list, click the mode of authentication you want to use. You have the following options:
 - **Windows Authentication:** Use the logged-in Windows user's credentials for authentication.
 - **SQL Server Authentication:** Use a login set up on the SQL Server. Enter the username and password in the corresponding fields.
 - **Azure AD Password Authentication:** Use Azure AD credentials from a domain that is not federated with Azure AD. Enter the username and password in the corresponding fields.
 - **Azure AD Integrated Authentication:** Use the logged-in Windows user's credentials for authentication, provided that he is logged in using Azure AD credentials from a domain that is federated with Azure AD.
4. In the **Database** box, enter a name for the database, and then click **Create** to create a new database to use as repository. Alternatively, you can select an existing database from the list. If you enter the name of an existing, but empty, database in the box and click **Create**, TimeXtender will offer to create the necessary tables and stored procedures in the database so it can be used as a repository.
5. (Optional) In the **Connection timeout** box, enter the number of seconds to wait before terminating the attempt to connect to the server. The default is 15 seconds. A value of zero will disable the timeout.

6. (Optional) In the **Command timeout** box, enter the number of seconds to wait before terminating the attempt to connect to the database. The default is 1800 seconds. A value of zero will disable the timeout.
7. (Optional) In the **Max. rows to copy** field, enter the batch size when using ADO.net transfer. '0' equals unlimited.
8. (Optional) In the **Encrypt connection** list, you can enable encryption of the connection, which is recommended when you are not in a private network (e.g. when your server is on Azure). You have the following options:
 - **No:** The communication is not encrypted (default).
 - **Yes:** The communication is encrypted. The server's certificate is verified by a certificate authority.
 - **Yes, trust server certificate:** The communication is encrypted. but the server's certificate is not verified. This setting is not recommended for use on public networks.
9. (Optional) Enter a SSIS server name in the **SSIS Server** box.
10. (Optional) Enter any addition connections settings in the **Additional connection properties** box.
11. Click **Test Connection** to verify that the connection is working.

Projects

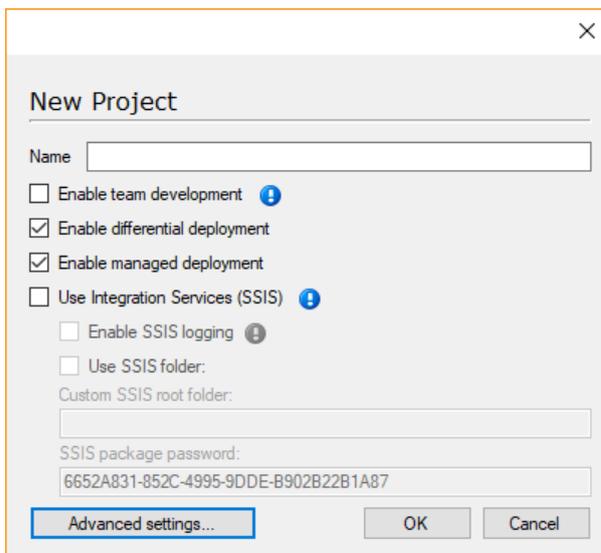
A TimeXtender project contains all other elements of the data warehouse.

You can only have one project open at a time. However, if you want to compare different versions of a project, you can open another instance of TimeXtender, and load another version of the project to view side-by-side.

Creating a Project

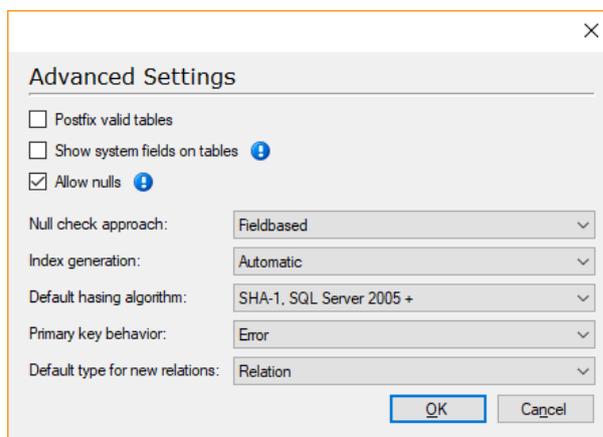
To create a projects follow the steps below:

1. On the **File** menu, click **New Project....** The **New Project** window appears.



2. In the **Name** box, type a name for the new project.
3. Select **Enable team development** to enable multiple developers to work on the project simultaneously. Note that this setting cannot be change once the project has been created.
4. Select **Enable differential deployment** to take advantage of TimeXtender's differential deployment feature that calculates what steps have changed and need to be deployed and selects only those steps for deployment. When differential deployment is disabled, all steps are deployed.
5. Select **Enable managed deployment** to have calculate dependencies and deploy the objects in the optimal order. There is no difference in performance or otherwise if you are only deploying one object. When managed deployment is disabled, you will have to make sure that objects are deployed in the correct order yourself.
6. Select **Use Integration Services (SSIS)** to use SQL Server Integration Services (SSIS) for data transfer. This SQL Server component Integration Services needs to be installed on the machine that deploys and executes the tables for this to work. TimeXtender will use ADO.net if SSIS is not enabled.

- Select **Enable SSIS Logging** to enable SSIS logging.
 - Clear **Use SSIS Folder** to not create a folder for the SSIS packages created. It is recommended to keep this setting enabled to prevent accidental overwriting of the packages by other projects that contain tables with the same names.
 - (Optional) Type the name of the SSIS folder in **Custom SSIS Root folder**.
 - (Optional) Type a new password for encrypting SSIS packages in **SSIS package password** or leave it as the default. SSIS packages that contain SQL Server logins are encrypted using this password. Please note that this only applies when you are working with sources that use SQL Server authentication. Sources that use Windows authentication do not require encrypted SSIS packages.
7. Click on **Advanced Settings...** to access additional settings for the project. The **Advanced Settings** window appears.



8. Select **Postfix valid tables** to postfix the valid table instance with "_V".
9. Select **Show system control fields** to show system control fields such as DW_ID, DW_Batch, DW_SourceCode and DW_Timestamp.
10. Select **Allow Nulls** to allow null field values instead of moving the row to the error table when encountering null values. In the **Null check approach** list, click **Field Based** to use a field based check or click **Row Based** to use row based check. A field based check will tell you exactly where the null value is, while a row based check will only tell you that the record has a null value.
11. In the **Index Generation** list, choose the default setting for index generation. Select **Automatic** to automatically create indexes as needed, **Manual** to enable automatic index generation on an a per-table basis or **Disabled** to not create any indexes automatically.
12. In the **Default hashing algorithm** list, click on the hashing algorithm you want to use for hash fields that are set to "use project default" hashing algorithm. You have the following options:
1. **SHA-2 512, SQL Server 2016 +**: The safest hashing algorithm in terms of the probability that two different data sets would create identical hashes. The size of the hash is 64 bytes. It is about 40% slower than "SHA-1, SQL Server 2016+" and you should only use this algorithm when extreme safety is required. This

- algorithm requires SQL Server 2016.
2. **SHA-1, SQL Server 2016 +:** The fastest hashing algorithm when the amount of data to be hashed is more than 8000 bytes. In those cases, it can be about 30% faster than "SHA-1, SQL Server 2005 +". Otherwise, the performance for the two algorithms are the same. The size of the hash is 20 bytes. This algorithm requires SQL Server 2016.
 3. **SHA-1, SQL Server 2005 +:** The default algorithm in TimeXtender. Slower than "SHA-1, SQL Server 2016 +" when the amount of data to be hashed is more than 8000 bytes. The size of the hash is 20 bytes and it is compatible with all SQL Server versions supported by TimeXtender.
 4. **Plain text (debug):** Used for debugging, it will concatenate the fields into a string. This way, you can see what data goes into creating the hash. Because of a limitation in SQL Server, the string is limited to a length of 4000 characters.
 5. **Legacy binary:** Provided for compatibility with earlier versions of TimeXtender and should not be used in new projects. It is not typesafe and is limited to a total of 4000 characters for all the fields the hash is calculated from.
 6. **Legacy plain text (debug):** Used for debugging legacy algorithms, it will concatenate the fields into a string with a length of up to 4000 characters.
13. Primary key behavior has to do with how TimeXtender should treat primary key violations. In the **Primary key behavior** list, Click **Error** to move the offending row to the error table, click **Warning** to move the offending row to the warning table or click **None** to ignore the violation.
 14. Under **Default type for new relations**, choose how TimeXtender should treat foreign key violations. Click **Error** to move the offending row to the error table, click **Warning** to move the offending row to the warning table or click **Relation only** to ignore the violation.
 15. Click **OK** to return to the previous window and **OK** again to create the project.

Since you can only have one project open at a time in TimeXtender, you will be asked to save the current project if you try to create a new project or load an existing project when you already have a project open.

Saving and opening Projects

Saving a project in the repository and opening a project from the repository works much the same as in other Windows programs.

Saving a Project

TimeXtender includes version control. On every save, a new version of the project is saved, meaning that you can always go back to an earlier version of your project.

To save the project, follow the steps below.

1. On the **File** menu, click **Save** or **Save As...**
2. If you clicked on **Save As...**, or you are saving the project for the first time, type a name for the project and click **OK**.

Note: The project is automatically saved after a successful deployment.

Opening a project

1. On the **File** menu, click **Open....**
2. (Optional) Click **Change Version**, click the version you want to open and click **OK** if you want to open a version of the project other than the latest version.
3. In the **Project** list, select the project that you want to open, and then click **OK**.

Exporting and Importing Projects

Being able to export a project to an XML document is useful when you want to save a copy of a running project for future reference, or if you want to reuse parts of a project in another project. You can export a project to an XML document, and you can import a project from an XML document.

Importing Projects from XML Documents

1. On the **File** menu, click **Import....**
2. In the **Import File** field, click the ellipsis (**..**), and then navigate to and select the file you want to import.
3. Click **Open**, and then click **OK**.

Exporting Projects to XML Documents

1. On the **File** menu, click **Export...**
2. In the **Export File** field, click the ellipsis (**..**), and then navigate to and select the file you want to export to.
3. Click **Open**, and then click **OK**.

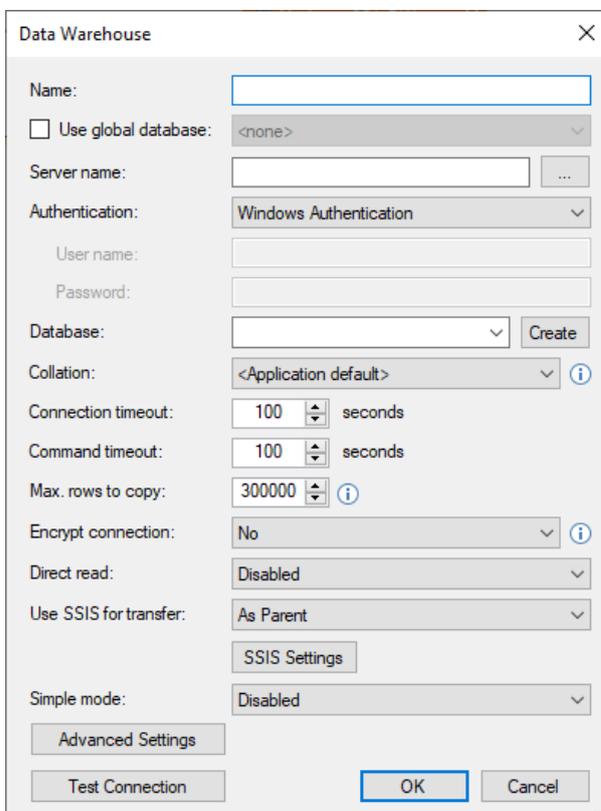
Data Warehouses

A data warehouse in TimeXtender is a SQL Server database on premise or in Azure where your data is stored for queries and analysis. Often, a TimeXtender project consists of one data warehouse where you consolidate data from one or more staging databases and a number of data sources.

During execution of a project, TimeXtender extracts data from the staging database or ODX data storage and transfers it to the data warehouse. Initially, the data resides in what is known as raw table instances in the data warehouse. TimeXtender applies data transformations and data cleansing rules and saves the resulting data in valid instances of the tables, ready for queries and analysis.

Adding A Data Warehouse

1. In the **Solution Explorer**, right-click **Data Warehouses**, and click **Add Data Warehouse**.



2. In the **Name** box, type a name for the data warehouse. The name cannot exceed 15 characters in length.
3. In the **Server name** box, type the name of the server that you want to store the database on. If it is a named instance, type the server name and the instance name. Click the ellipsis (...) to choose one of the available servers in you local Active Directory, if any.
4. In the **Authentication** list, click the mode of authentication you want to use. You have the following options:

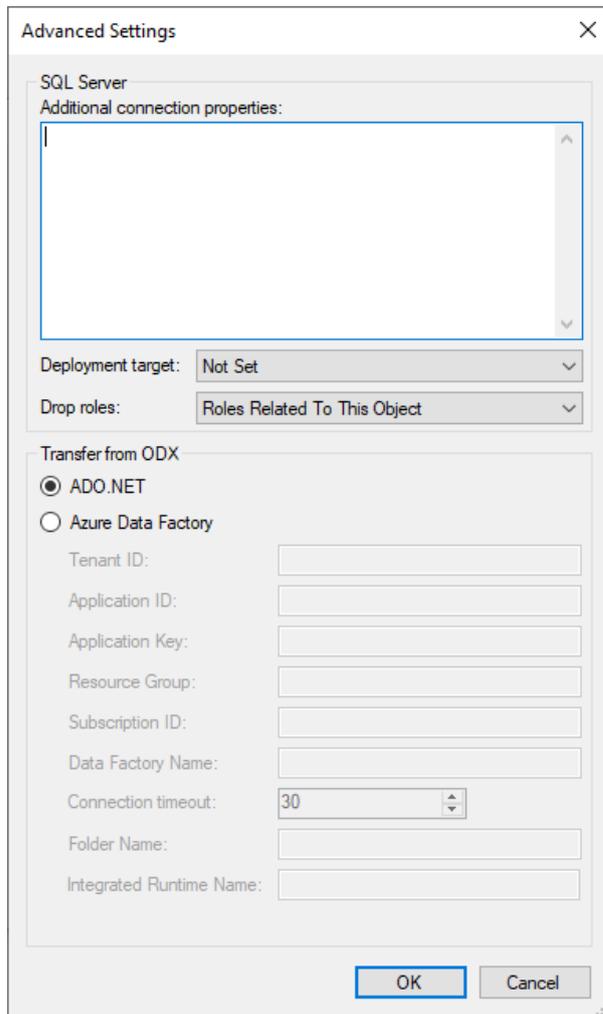
- **Windows Authentication:** Use the logged-in Windows user's credentials for authentication.
 - **SQL Server Authentication:** Use a login set up on the SQL Server. Enter the username and password in the corresponding fields.
 - **Azure AD Password Authentication:** Use Azure AD credentials from a domain that is not federated with Azure AD. Enter the username and password in the corresponding fields.
 - **Azure AD Integrated Authentication:** Use the logged-in Windows user's credentials for authentication, provided that he is logged in using Azure AD credentials from a domain that is federated with Azure AD.
5. In the **Database** box, select an existing database, or type the name of a new database, and then click **Create**.
 6. (Optional) In the **Collation** list, click on the collation you want to use. You can choose a specific collation or one of the following:
 - **<Application Default>:** Uses the application default, which is **Latin1_General_CI_AS**.
 - **<Server Default>:** Inherits the collation from the specified server.

It is best practice to use the same collation for the staging database, data warehouse database and any SSAS Multidimensional databases.
 7. (Optional) In the **Connection timeout** box, enter the number of seconds to wait before terminating the attempt to connect to the server. Set it to 0 to wait indefinitely.
 8. (Optional) In the **Command timeout** box, enter the number of seconds to wait before terminating a command.
 9. In the **Max. rows to copy** box, enter the batch size when using ADO.net transfer. '0' equals unlimited.
 10. (Optional) In the **Encrypt connection** list, you can enable encryption of the connection, which is recommended when you are not in a private network (e.g. when your server is on Azure). You have the following options:
 - **No:** The communication is not encrypted (default).
 - **Yes:** The communication is encrypted. The server's certificate is verified by a certificate authority.
 - **Yes, trust server certificate:** The communication is encrypted. but the server's certificate is not verified. This setting is not recommended for use on public networks.
 11. (Optional) In the **Direct read** list, you can enable direct read from the staging database(s) to the data warehouse database. With direct read enabled, data is transferred using a stored procedure containing a simple `SELECT` statement. This can, especially if TimeXtender is not on the same machine as the SQL Server, give better performance than SSIS or ADO.net since transfers using these technologies happen via TimeXtender . For direct read to work, some prerequisites must be met: On SQL Server, the databases need to be on the same server. On Azure SQL Database, the staging and data warehouse databases need to be in the same database. You have the following options for direct read:

- **Disabled**
 - **Matching Server:** Direct read is used if the data warehouse and staging data-base server names match.
 - **Matching Server and Database:** Direct read is used if the data warehouse and staging database server names and database names match.
12. In the **Use SSIS for transfer** list click on **Yes** to enable SQL Server Integration Services data transfer, **No** to disable it or leave it at **As Parent** to respect the project setting.
 13. (Optional) Click **SSIS Settings** to configure SSIS.

- In the **Local SSIS version** list, click on the SSIS version that are installed on the local machine. **Automatic** (default) will use the SSIS version corresponding to the data warehouse server and try to connect to the **Package SQL Server** through the data warehouse SQL Server. In this case, you do not need to enter authentication information.
 - Enter a server name in the **Package SQL Server** box. This server is used for storing the SSIS packages generated by TimeXtender. If you do not enter a value in this field, the packages are stored on the database server.
 - In the **Authentication** list, click on the type of authentication you want to use. Enter a **Username** and **Password** if required by the authentication type you have selected.
 - Select **Enable remote SSIS execution** to enable the execution of SSIS packages on a remote server and enter the details for the remote server below. Note that you will need to install the Remote SSIS Execution service on the remote server. It can be downloaded from the support site, [https:// support.timextender.com](https://support.timextender.com) .
14. In the **Simple Mode** list, click **Enabled** if you want to set all tables in the data warehouse to simple mode. See [Simple Mode](#) for more information.

15. Click **Test Connection** to verify that the connection is working.
16. Click **Advanced Settings** to access the advanced settings for the data warehouse.
These are all optional.



17. If you want to add additional connection strings, enter them in the **Connection String Properties** box.
18. If you are deploying your data warehouse on Azure Synapse Analytics (formerly Azure SQL Data Warehouse), click on **SQL Data Warehouse** in the **Deployment target** list. For other versions of SQL Server, the version and edition can usually be auto-detected, and the setting can be left at its default. However, you can use the option to set the deployment target explicitly if you experience issues.

Note: If you use Azure Synapse Analytics, some options and settings are not available due to the differences between this and other flavors of SQL Server. However, additional table settings are available to control Azure Synapse Analytics specific options. The differences are noted when applicable throughout the user guide.

19. See [Adding a Database Role](#) for an explanation of the **Drop role options** setting.
20. Under **Transfer from ODX**, select the method you want to use. The following settings are available:

- **ADO.net:** The application-native method.
- **Azure Data Factory:** Uses Azure Data Factory, which can be faster if the ODX data storage and data warehouse are both on Azure.

Cleaning up the Database

To prevent accidental data loss, deleting a table in TimeXtender does not delete the physical table in the data warehouse or staging database or, if you use SSIS transfer, the SSIS package that was created to copy data. The downside is that tables deleted in TimeXtender still takes up space in the database.

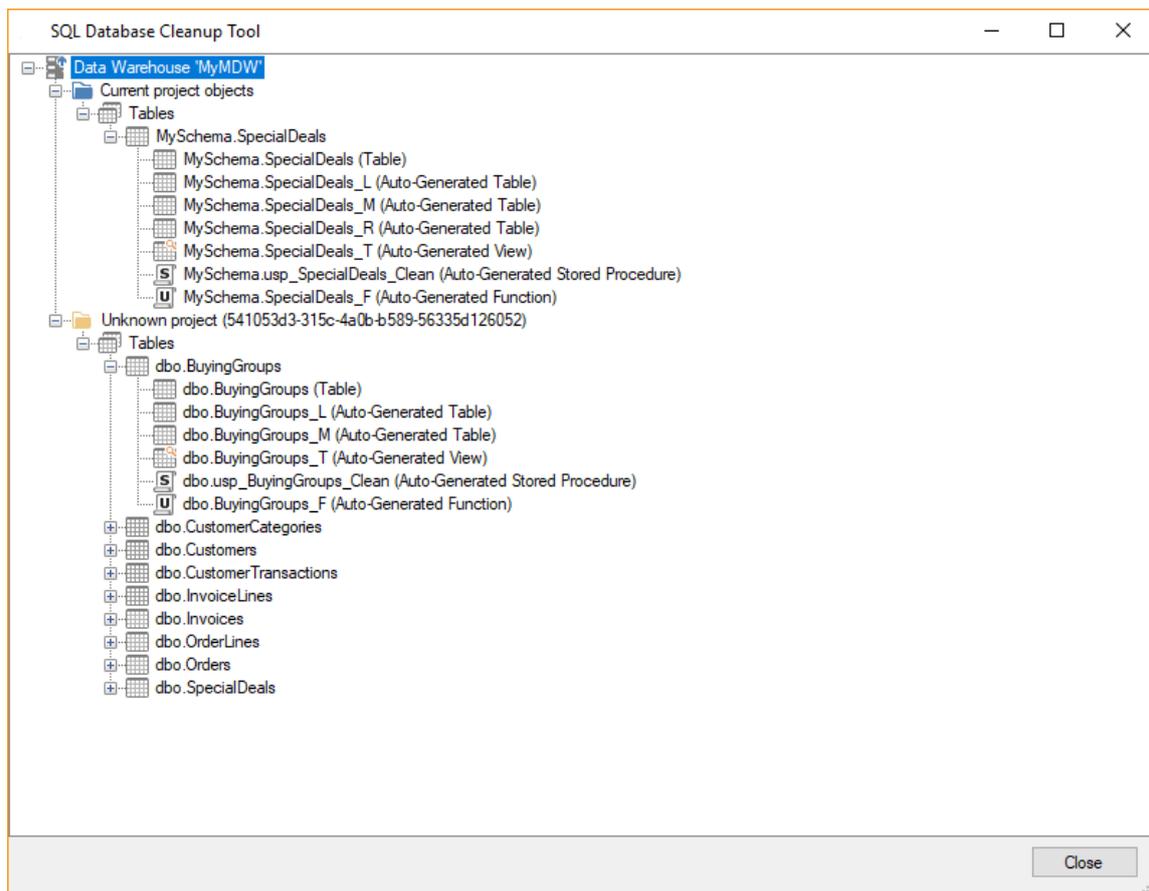
Identifying and Deleting Unused Tables

The **SQL Database Cleanup Tool** enables you to identify tables left behind by TimeXtender and delete - drop - them to free up space. Note that database schemas are not deleted from the database. You will need to drop those manually in SQL Server after deleting them in TimeXtender.

Warning: When you drop a table with the SQL Database Cleanup Tool, it is permanently deleted. Take care when you use the tool.

To clean up your data warehouse or staging database, follow the steps below.

1. Right click a data warehouse or staging database, click **Advanced** and click **SQL Database Cleanup Tool**. TimeXtender will read the objects from the database and open the **SQL Database Cleanup Tool** window.



2. The objects in the database that are no longer, or never was, part of the currently opened project are listed.
3. (Optional) Right click a table, view, procedure or function and click **Script** to display the SQL script behind the object.
4. Right click a table, view, procedure or table and click **Drop** to drop the object from the database.
 1. If the item does not have subordinate items, click **Yes** when TimeXtender asks you to confirm the drop.
 2. If the item has subordinate items, a window will open with a list of the objects that will be dropped. Clear the selection for any tables you want to keep and then click **Drop**.

Note: TimeXtender will automatically clear the selection for any incrementally loaded tables to prevent accidental data loss. TimeXtender will ask you to confirm if you want to drop an incrementally loaded table.

5. When you have dropped the all the objects you want to delete from the database, close the window.

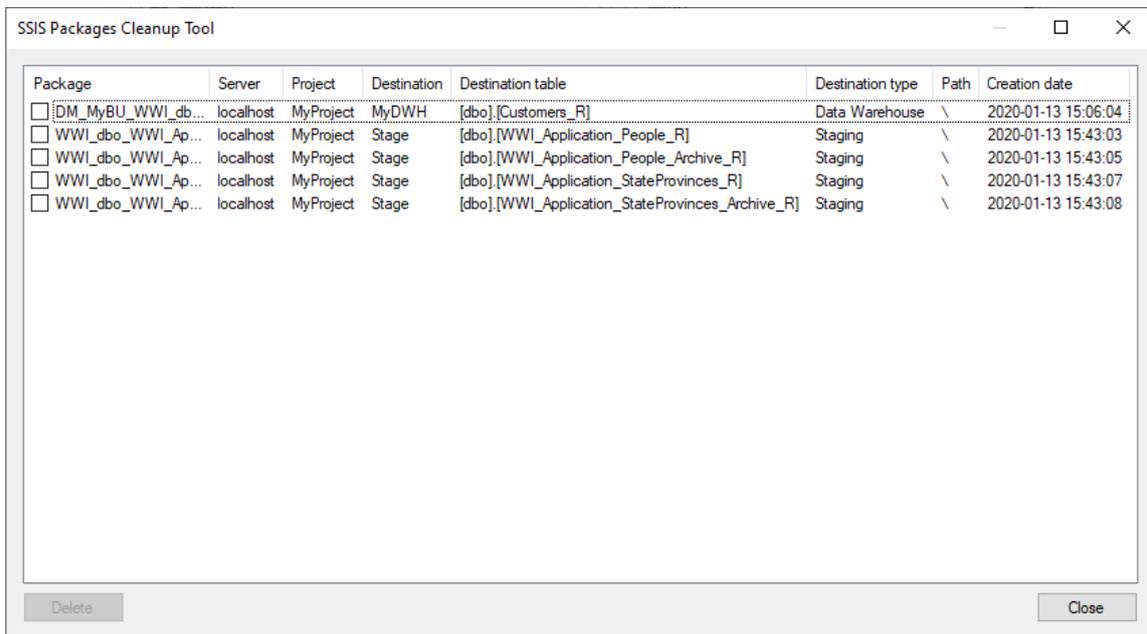
Identifying and Deleting Unused SSIS Packages

With the **SSIS Packages Cleanup Tool**, you can delete the SSIS packages that is left when you have deleted a table, stopped using SSIS for transfer or made other changes that

makes some packages unnecessary.

To clean up SSIS packages, follow the steps below.

1. In the Solution Explorer, right click your project, click **Advanced** and click **SSIS Packages Cleanup Tool**. A list of unused packages is displayed.



2. Select the packages you know are safe to delete, and click **Delete**.

Warning: When you delete a package with the SSIS Packages Cleanup Tool, it is permanently deleted from the server. Take care when you use the tool.

Business Units and Staging Databases

Note: If you use the [ODX Server](#), data will be copied directly from the ODX Server to a data warehouse. For this reason, you won't be using business units and you can safely skip this section.

In TimeXtender, a business unit is any part of your organization that you want to treat as a separate entity in your project. For example, you may want to treat a company headquarters and each of its subsidiaries as separate business units.

Each business unit in your project has a staging database and a number of data sources.

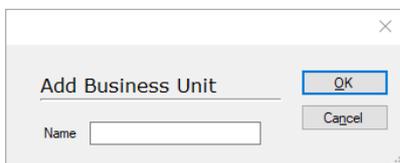
The staging database can be stored on SQL Server or on Azure SQL Database. It stores the data selected for extraction from the data sources. Additionally, many of the validation and transformation processes take place in the staging database. This ensures that the cleansing process has limited impact on the transaction database.

The difference between a staging database and a data warehouse is minimal. You can add custom tables, views, scripts, table relationships, security to a staging database just as if it was a data warehouse database.

Adding a Business Unit

To add a new business unit, follow the steps below:

1. In the **Solution Explorer**, right-click **Business Units**, and click **Add Business Unit**. The **Add Business Unit** window appears.

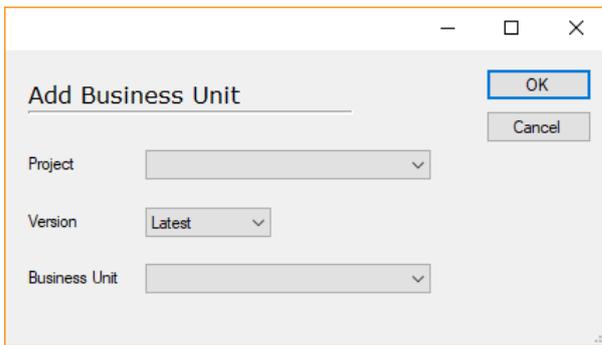


2. Type a name for the business unit and click **OK**. The **Staging Database** window appears. This window is similar to the window you see when you add a data warehouse. See [Adding a Data Warehouse](#) for more information on the different options.

Adding an External Business Unit

You can add business units from other projects to your project. This enables reuse of business units across different projects. To add an external business unit, follow the steps below:

1. In the **Solution Explorer**, right-click **Business Units**, and then click **Add External Business Unit**. The **Add Business Unit** window appears.



2. In the **Project** list, select the project that contains the business unit you want to add.
3. In the **Version** list, click on the version of the business unit you want to import. You have the following options:
 - **Latest:** Adds the last saved version of the project, which does not always correspond to the last deployed version
 - **Deployed:** Adds the last deployed version of the project
4. In the **Business Unit** list, click on the business unit you want to add, and then click **OK**.
5. Click either **By ID** to synchronize fields by ID or **By Name** to synchronize fields by name. The business unit is added and listed in the **Solution Explorer**.

If you make changes to the external business unit in its "own" project, you can synchronize the changes to your project.

To synchronize an external business unit

- Right click the external business unit click **Synchronize** and click either **By ID** to synchronize fields by ID or **By Name** to synchronize fields by name.

Enabling Simple Mode for a Business Unit

Simple mode is a setting on tables on business units aimed at maximizing performance when you need to copy large amounts of data into a staging database to create an exact copy. See [Simple Mode](#) for more information.

Simple mode can be enabled on the business unit level for all data source and tables on the business unit. It can be overridden on the data source level or on individual tables.

To enable simple mode for a business unit

- Right click on the business unit, click **Business Unit Settings** and select **Enable Simple Mode**.

Team Development

TimeXtender supports multiple developers working on the same project at the same time. Version notes enable developers to share details about their changes to the project and work items allows developers to see which objects other team members are working on to prevent them from modifying the same objects simultaneously.

The team development feature of TimeXtender depends on the developers to take care when working together as the feature itself does not prevent developers from making conflicting changes.

Warning: If two developers make changes to the same object, the repository can end up in an inconsistent state that prevents projects from being opened.

Follow the rules below to ensure that the project stays consistent:

1. Never work on the same object - table, dimension or cube - as someone else.
 - Always create work items before making any changes to an object.
 - If anyone else has the item marked for their use, wait for them to release the work item.
 - Once you are done working on an object, save the project and release the work item.
 - Keep the work items window open when working.
2. Do not rename tables, cubes or dimensions. The only exceptions are if you just created the table and have not saved yet or if nobody else has the project open. You can see who has a project open in the work items dialog.
3. Always save and reload (Ctrl+F5) before you start a new development task. This ensures that you have the latest version of all objects before you start taking over the work of somebody else.
4. Immediately before creating a new table, always save and reload.

Enabling the Team Development Environment

You need to enable Team Development for any project you want to use the feature in.

1. On the **Tools** menu, click **Repository Administration**.
2. On the **Projects** tab, right-click the relevant project and click **Enable Team Development**.
3. Close the window. Multiple users can now access the project concurrently.

Work Items

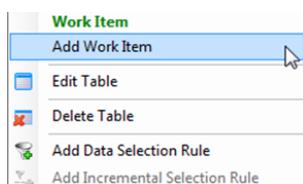
Work items allow the team to know what its users are working on. This will give a visual indication as to what areas of the project are currently under development. Work items are meant to be created immediately prior to starting work on a set of objects and can be either manually deleted or removed during deployment of the object. Work items can be added to the following objects:

- Project
- Data Warehouse
- Data Warehouse table
- Staging Database
- Staging Database table
- Data Source Table
- SSAS Multidimensional Database
- Cube

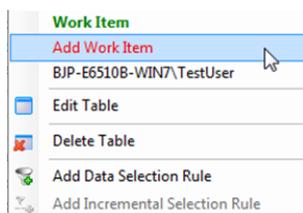
Adding Work Items

To add a work item, follow the steps below.

1. Right-click the relevant object and click **Add Work Item**.



If the object that the user is adding the work item to already has a work item created by another user, it will display **Add Work Item** in red and identify the other user below. This allows users to know if other users are modifying the same object in order to facilitate collaboration.



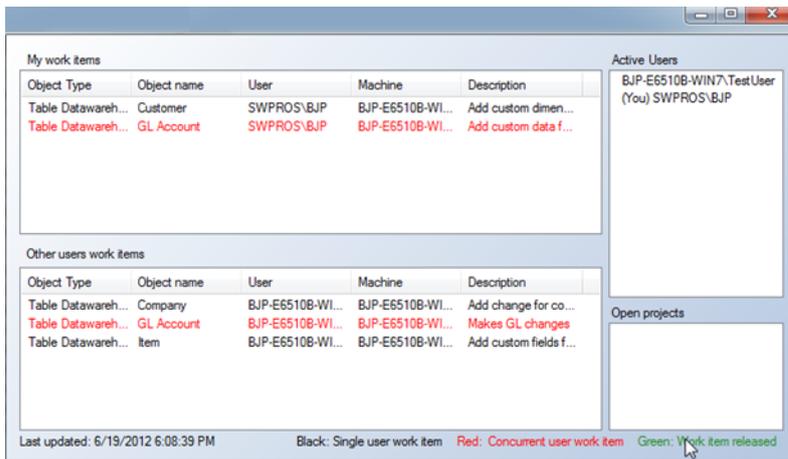
Otherwise, the Create Work Item window appears.

2. >In **Description** type a short description of the work item, e.g. what task the work item is a part of, and click **Create Work Item**.
3. >Click **Create Work Item**.

Viewing Existing Work Items

To view existing work items, follow the steps below.

1. On the **Tools** menu, click **Work Items**.
2. The **Work Item** window opens. It contains a list of your own work items, **My work items**, and a list of **Other users work items**.



Each list displays the following information:

Column	Description
--------	-------------

Object Type	The type of object the work item is associated with (project, database, table, or cube)
Object Name	The name of the project, database, table, or cube associated with the work item
User	The user name of the person who created the work item
Machine	The machine that the work item was created on
Description	Descriptive text defined by the user for the work item

Work items can have three colors depending on their status:

- Black: A work item that is only flagged by a single user.
- Red: Work items that share the same Object Name for different users. This tells you that some collaboration will be needed regarding which user is accessing the data. While Team Development allows multiple users to modify the project concurrently, these users should not modify the same object at the same time.
- Green: Work items that have been completed by another user who has saved or deployed and marked the work item as completed.

In the **Active Users** pane, all users currently accessing the project are shown.

Editing a Work Item

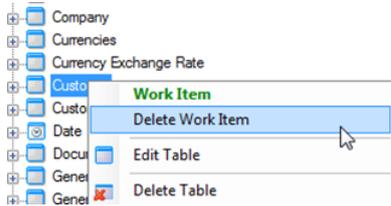
Work items can only be edited by the user that created the work item. To edit a work item, follow the steps below:

1. >On the **Tools** menu, click **Work Items**.
2. >In **My Work Items**, right-click the work item to edit and select **Edit Work Item**.
3. >Update the work item description and click **Update Work Item** to save the changes.

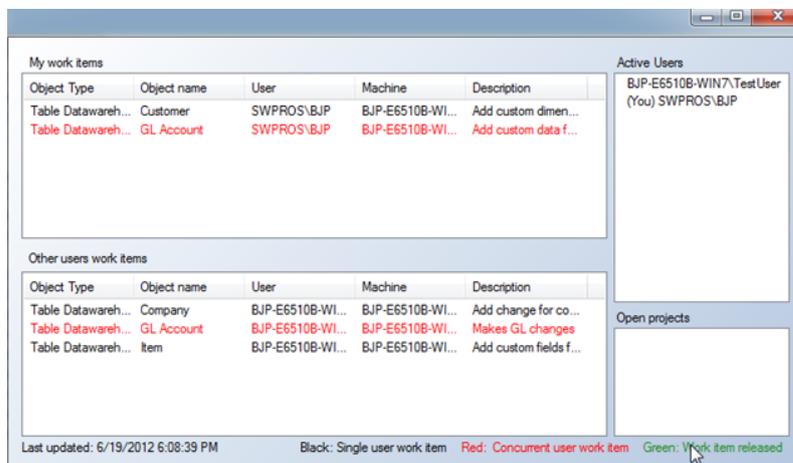
Deleting Work Items

Work items can only be edited by the user that created the Work item. You can delete a work item in three different ways:

- >Right click the relevant object and click **Delete Work Item**



- >On the **Tools** menu, click **Repository Administration**. On the **Work items** tab, click a work item and click **Delete**.
- >On the **Tools** menu, click **Work Items**.



In **My work items**, right-click the relevant work item to edit and click **Delete Work Item**. TimeXtender will ask you to confirm the deletion. Click **Yes**.

Work items can also be removed when adding version notes, which is discussed below.

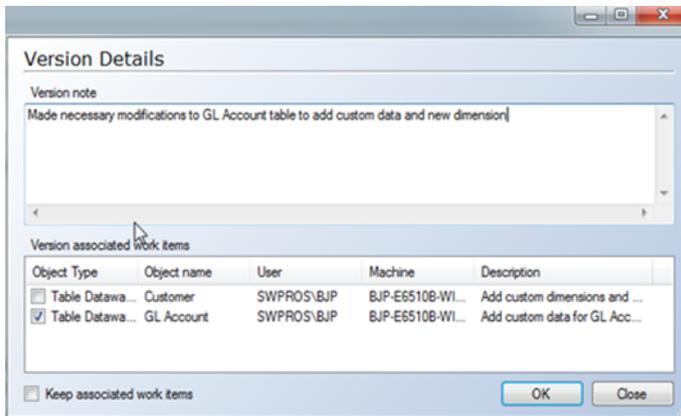
Version Notes

Version notes enables you to add comments about changes that you have made to a project. These comments can then be viewed in the future to reference changes that were made.

Adding Version Notes

1. >To ensure version notes are enabled, click **Options** on the **Tools** menu. The **Options** window appears.
2. >Under **Prompt for version details**, make sure one of the following options is selected:

- **>Every time the project is saved:** You will be prompted to enter a version note when you deploy an object or when you click the **Save** button.
 - **> Every time the project is saved during deployment:** You will only be prompted to enter a version note when you deploy an object.
3. **>When you deploy an object in the project, the **Version Details** window will appear just before the deployment begins.**



You can type in any details you want to include in the version note. If relevant, you can also select one of your existing work items to associate with the version note. Selecting a work item will also remove the work item from the list of existing work items. If you want to associate a work item with the version note, but not remove the work item, select **Keep Associated Work Items**. If you do not want to save a version note for this deployment, you can click **Close**. This will close the window without saving a version note.

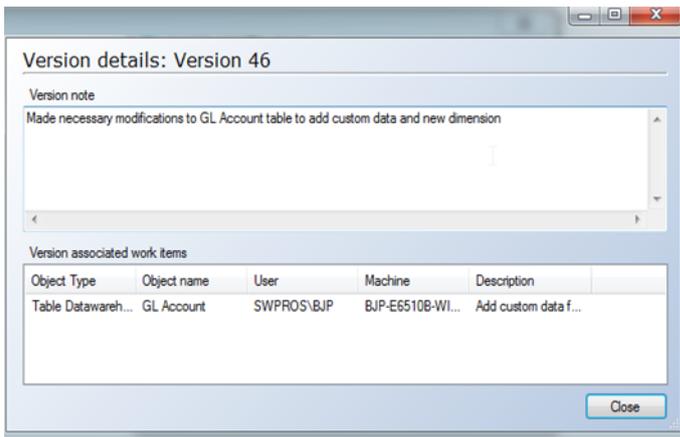
Viewing Version Notes

Viewing version notes can be useful when you would like to look back and see when particular changes were made. Follow the steps below to see the list of project versions and associated version notes.

1. **>On the **Tools** menu, click **Repository Administration**.** The Repository Administration window appears.
2. **>Right click the relevant project and click **Show Project Versions**.** The Project Versions window appears.
3. **>If a notepad icon is displayed in the **Note** column, a version note is available for the version.** Hover over the notepad icon to see a quick description of the note.



4. **>Click the notepad icon to see the full version note as well as any work items associated with the version note.**



Project Perspectives

The purpose of project perspectives is to make it easier to work with large projects. In big projects, it can be hard to maintain a good overview and find an individual object quickly.

The idea is that you can create different perspectives on a project. A perspective is a subset of the project objects that relate to a specific area or task. For example, you could create a “finance” perspective that contains all the tables, dimensions and cubes that are related to finance. When this perspective is active, anything else will be hidden.

A perspective can be static or dynamic. In dynamic project perspectives, any object that is related to an object already in the perspective will be included in the perspective. This is very useful if you have e.g. a SSAS Multidimensional cube, semantic model or Qlik model for sales and want to see the tables, fields etc. related to that. Creating a dynamic perspective based on the cube or model will give you that overview in seconds.

Adding a Project Perspective

To add a project perspective, follow the steps below.

1. In the **Solution Explorer**, right click Project Perspectives and click **Add/Edit Project Perspectives**.
2. Click **Add**. A new column is added to the grid with the name “Perspective” in the **Perspective Name** row. Double-click the name to edit it and click outside the field when you have finished typing. Select **Dynamic Perspective** if you want the new project perspective to be dynamic. You can change this setting at any time.
3. Each Qlik model, SSAS Multidimensional cube, data warehouse and business unit is listed in the first column of the grid. Click the **+** beside one of the objects to show the child objects. Select the Qlik models, cubes, dimensions and tables you want to include in your project perspective. If you add an object to a dynamic perspective, dependent objects will be added automatically. You can recognize dynamically added objects by the checkbox with an indeterminate state.
4. Click **OK**. The new project perspective now appears under **Project Perspectives** in the **Solution Explorer**.

Adding objects to and removing objects from a perspective

You can add most objects - tables, fields, dimensions, cubes - to a perspective. The same object can be added to as many perspectives as you want to.

In the Project Perspectives menu, the perspectives that the object is a part of are checked. If a dot is displayed next to the perspective name, the object is part of a dynamic perspective because it is related to an object that is explicitly part of the dynamic perspective.

To add an object to a perspective

- Right click the object, click on **Project Perspectives** and click on the name of perspective you want to add the object to.

To remove an object from a perspective

- Right click the object, click on **Project Perspectives** and click on the name of checked perspective you want to remove the object from.

Note: You cannot remove an object from a dynamic perspective if it is not explicitly selected to be part of the perspective. In other words, if there is a dot next to the perspective, it is not an explicit part of the perspective and cannot be removed.

Activating a perspective

To activate a perspective

- In the **Solution Explorer**, expand **Project Perspectives** and double-click the perspective you want to activate OR right-click the perspective and click **Use Project Perspective**.

Deactivating all perspectives

To deactivate the active perspective

- In the **Solution Explorer**, expand **Project Perspectives** and double-click the perspective with **(Active)** postfixed OR right-click the perspective and click **Use Project Perspective**.

Sorting objects in a perspective

Objects within a perspective can be sorted by the execution order or alphabetically. To change the sort order of the active perspective

- Right click the currently active perspective and select either **Sort by execution** order or **Sort alphabetically**. The chosen sort order will be saved for each perspective.

Note: It is not possible change the order of objects manually while a perspective is active.

Deploying and executing a perspective

Perspectives can be deployed and executed just as other objects. This enables you to easily work with a subset of you project from source to execution. You can deploy and/or execute a perspective in two ways:

- In the **Solution Explorer**, under **Project Perspectives**, right click the perspective you want to deploy and/or execute and click **Deploy**, **Execute** or **Deploy and Execute**.

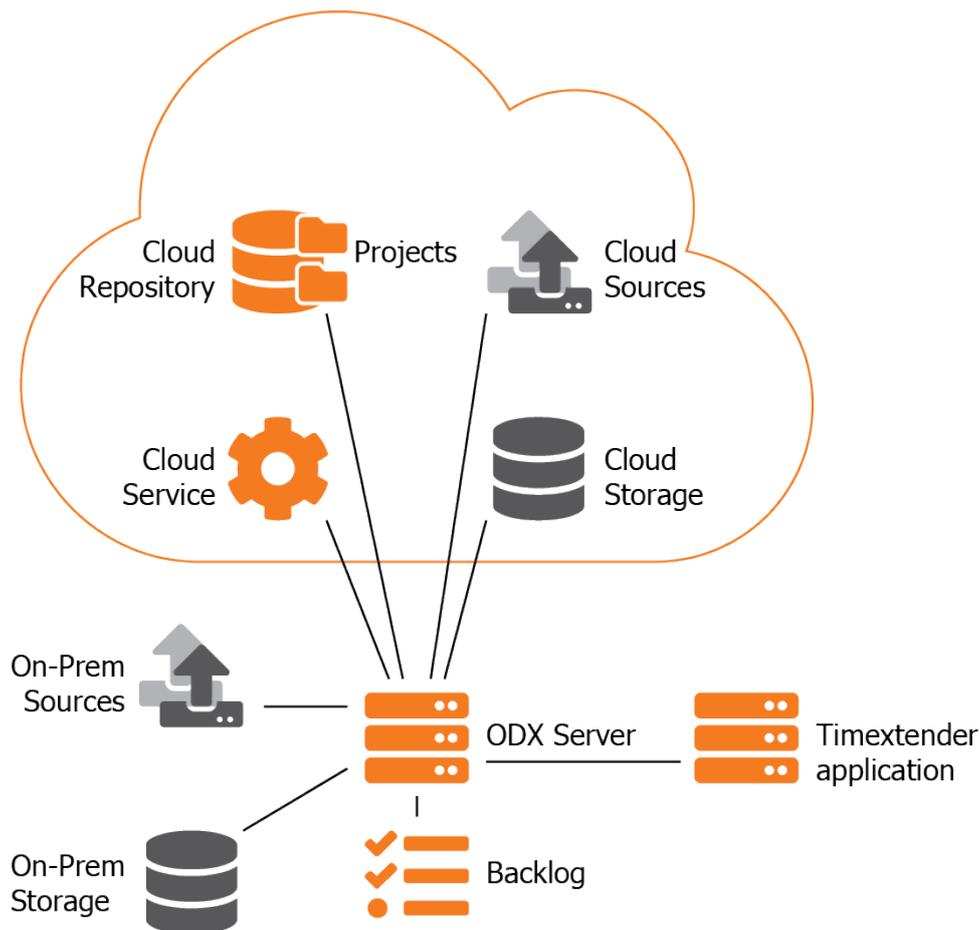
- You can also add a perspective to an execution package, for example if you want to execute the perspective on a schedule.

Using the ODX

In the Discovery Hub architecture, the ODX - Operational Data eXchange - is where data comes in from sources. In TimeXtender, the ODX layer is implemented as a stand-alone server that is optimized for handling very big amounts of data. In this chapter, you will learn how to set up and manage an ODX server.

The Structure of an ODX

The illustration below illustrates the different parts that make up the ODX part of the Discovery Hub architecture.



The **ODX server** sits in the middle of it all and copies data from **data sources** to a **data storage**, which can both be on-premise or in the cloud, and, when requested by TimeXtender, on to modern data warehouses.

For storing meta data about sources and storages as well as the tasks it needs to run, the server uses a local **backlog** that is continuously synced with the **cloud repository**. The cloud repository is divided into **projects**. Each ODX server has one project open at a time and two ODX servers should never use the same project at the same time.

The repository is not tied to a specific license. Instead, each company has a number of shared repository managed by TimeXtender. The ODX server gets the information it needs to connect to the cloud repository from the **cloud service**.

ODX is a client/ server solution. The server runs as a service with no GUI. Once initial setup has been completed, you manage the ODX server through the TimeXtender application that can be installed on the same or a different machine.

Installation of the ODX server is covered in the next section, followed by sections on adding data sources, tasks and security, execution and the administrative aspects of managing the ODX.

To learn more about using data from the ODX, see [Copying Data from an ODX to a Data Warehouse](#).

Installation

To install and configure the ODX Server, you will need the following:

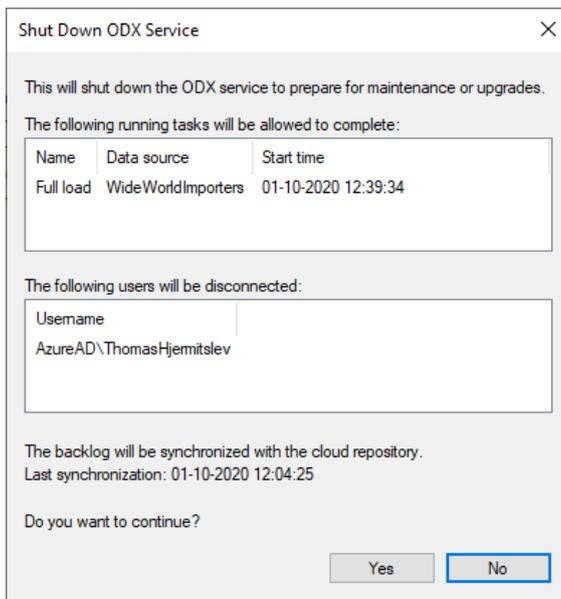
- The ODX Server installation package. Can be downloaded from the support site at support.timextender.com
- The client secret that is used to connect the ODX server to the cloud repository. Can be found in the TimeXtender User Portal at portal.timextender.com/odx
- A user account for running the ODX server service if you need to connect to SQL Servers using Active Directory authentication. The user account needs to have the appropriate access on these servers.

When these prerequisites are met, you are ready install the ODX server.

Installing and Configuring the ODX Server Service

To install and configure the ODX Server, follow the steps below.

1. If you are upgrading to a new version of the ODX, shut down the old server service.
 1. Open TimeXtender, right click the ODX and click **Shut Down ODX Service...**



2. Click **Yes** to confirm.
 3. Uninstall the ODX Server.
2. Run the installation package on the server. This will install the ODX Server Windows service.
3. When the installation package has completed, the **ODX Server Configuration** wizard will open to help you configure the server and set up and start the Windows service. Follow the steps of the wizard to complete the configuration.

Note: If you upgrading the ODX and connecting existing ODX project, you will be asked if you want to take over the lock. Information about the ODX that currently

has the lock will be displayed. Use this to confirm that it is safe to take over the lock.

If you need to reconfigure the ODX Server, you can start the wizard from the ODX Server's installation folder, which defaults to `C:\Program Files\TimeXtender\ODX Server [version]\`.

Connecting to the ODX Server

Once your ODX server is set up and running, you can connect to it from TimeXtender to manage it or use data from it.

To connect to an ODX server, follow the steps below.

1. From the **Tools** menu, click **Options**, then click the **ODX Server** tab in the window that appears.
2. In the **Server** box, enter the ODX server's address.
3. In the **Port** box, enter the port number if it is different from the default.
4. In the **Password** box, enter the password for managing the ODX.

Note: The ODX server accepts two different passwords, that are added during the initial configuration. The 'admin' password allows you to do management tasks such as adding data sources as well as copying data from the ODX. The 'user' password only allows you to copy data from the ODX.

5. (Optional) Set your preferred timeouts for opening and closing connection to the ODX as well as for sending and receiving data under **Timeouts**.
6. Click **OK**. The ODX server is now listed in the Solution Explorer.

To browse the tables in the ODX data storage

- double-click the ODX in the Solution Explorer

To make changes to the ODX

- right click the ODX in the Solution Explorer and click **Manage ODX**

Note: The ODX server you add by following the steps above is your primary ODX server. If you open a project that uses data from another ODX server, that server will be displayed in the Solution Explorer as a secondary ODX server.

Data Storage

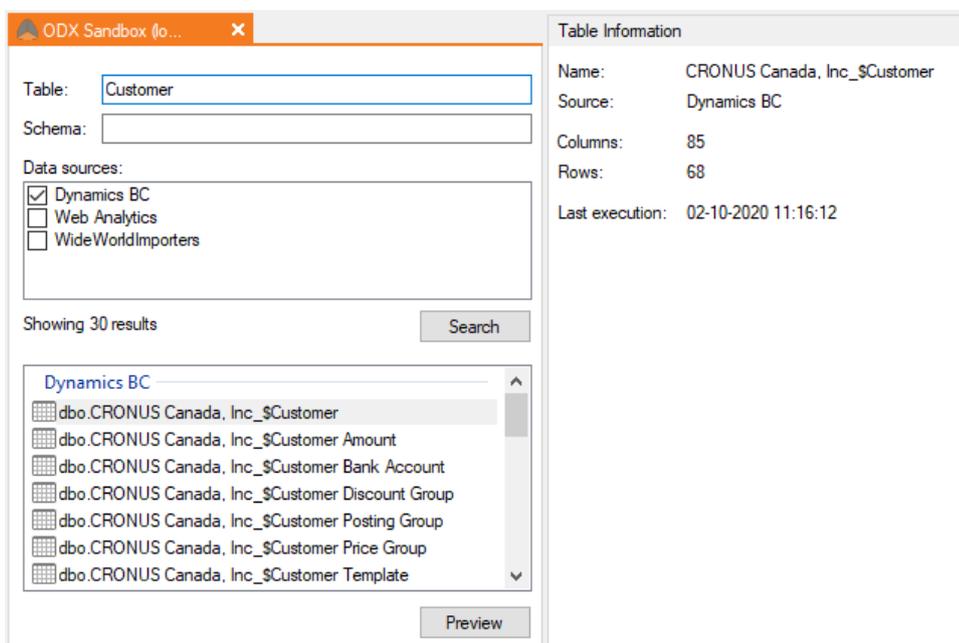
All data in the ODX is stored in the data storage. There is only one data storage per ODX, which can be on Azure Data Lake Storage Gen2 or a SQL Server database (on-prem or in Azure).

You will need to delete the existing storage to switch storage types. Deleting a data storage from the ODX Server does not delete the actual data in the data lake or on the SQL Server.

In general, the data storage "never forgets". No data is deleted from the storage unless you run storage management tasks (or delete it manually).

Browsing Tables in a Data Storage

Browsing the content of your data storage and selecting the tables you want to move into a data warehouse uses the same interface in TimeXtender.



To browse the tables in the ODX data storage

- double-click the ODX in the Solution Explorer

Since your data storage is likely to have many more tables that can be listed in the application in a useful way, no tables are listed initially. Use the search functionality to narrow the list of results to the tables you are interested in.

To search for tables across the storage, follow the steps below.

1. In the **Table** box, type the name, or part of a name, of the table. Leave the field blank to match all tables.
2. In the **Schema** box, type the name, or part of the name, of the schema you are interested in. Leave the field blank to match all schemas.

3. In the **Data Sources** list, select all the data sources you want to search.
4. Click **Search**

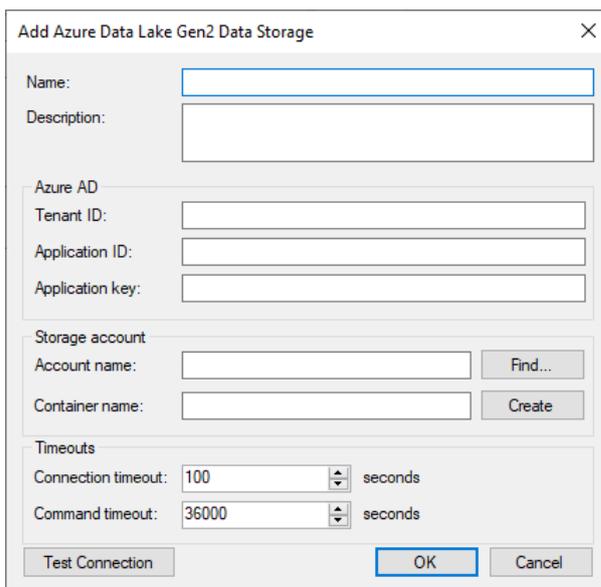
Warning: Large search results can make the UI unresponsive. For this reason, the list of results is limited to 10000 items. Since this issue is tied to machine performance, you may find that the UI becomes unresponsive with far fewer results.

When you select a table, the **Table Information** panel in the right-hand side of the UI will show you information about the table. If your storage is in Azure Data Lake, the information includes the number of columns, the number of files used for storing the table and the total file size. For SQL Server-based storages you can see the number of columns and rows. For both types of storage, the time of last execution is displayed.

Adding a Azure Data Lake Gen2 Data Storage

To add an Azure Data Lake Gen2 data storage, follow the steps below.

1. Open your ODX server in a tab and, right-click the aptly named **Right click to add Data Storage** node and click **Add Azure Data Lake Gen2 Data Storage...**



The screenshot shows a dialog box titled "Add Azure Data Lake Gen2 Data Storage". It contains the following fields and controls:

- Name:** A text input field.
- Description:** A text input field.
- Azure AD:** A section containing:
 - Tenant ID:** A text input field.
 - Application ID:** A text input field.
 - Application key:** A text input field.
- Storage account:** A section containing:
 - Account name:** A text input field with a "Find..." button to its right.
 - Container name:** A text input field with a "Create" button to its right.
- Timeouts:** A section containing:
 - Connection timeout:** A spinner box set to 100, followed by the text "seconds".
 - Command timeout:** A spinner box set to 36000, followed by the text "seconds".
- Buttons:** "Test Connection", "OK", and "Cancel" buttons are located at the bottom of the dialog.

2. In the **Name** box, type the name you want to use for the storage.
3. (Optional) In the **Description** box, type a description of the data storage.
4. In the **Tenant ID** box, enter the tenant ID GUID from Azure.
5. In the **Application ID** box, enter the application ID GUID from Azure.
6. In the **Application key** box, enter the application key from Azure.
7. In the **Account name** box, enter the name of the storage account you want to use. Click **Find...** to filter and browse the accounts on your Azure tenant to find the account.
8. In the **Container name** box, type the name of the container you want to use. Type the name and click **Create** to create a new container.

9. (Optional) In the **Connection timeout** box, enter a timeout for the commands you run on the storage.
10. (Optional) In the **Command timeout** box, enter a timeout for connecting to the storage.
11. If you want to use Azure Databricks, select **Use Azure Databricks** and enter the required options:
 - In the **Token** box, enter the token needed to authenticate with Azure Databricks.
 - In the **Cluster name** box, type the name of the cluster you want to use or leave it at the default.
 - In the **URL** box, enter the URL of the Azure Databricks service you want to use or leave it at the default.
12. Click **OK** to add the storage.

Adding A SQL Server Data Storage

To add a SQL Server data storage, follow the steps below.

1. Open your ODX server in a tab and, right click the aptly named **Right click to add Data Storage** node and click **Add SQL Server Data Storage...**
2. In the **Name** box, type the name you want to use for the storage.
3. (Optional) In the **Description** box, type a description of the data storage.
4. In the **Server** box, enter the address of the SQL Server you will be using.
5. In the **Authentication** list, click the mode of authentication you want to use. You have the following options:
 - **Windows Authentication:** Use the logged-in Windows user's credentials for authentication.
 - **SQL Server Authentication:** Use a login set up on the SQL Server. Enter the username and password in the corresponding fields.
 - **Azure AD Password Authentication:** Use Azure AD credentials from a domain that is not federated with Azure AD. Enter the username and password in the corresponding fields.
 - **Azure AD Integrated Authentication:** Use the logged-in Windows user's credentials for authentication, provided that he is logged in using Azure AD credentials from a domain that is federated with Azure AD.
6. In the **Database** box, type the name of the database.
 If you want to create a new database, make sure the authentication options below are correct and then type a name for the database and click **Create**. The **Create Database** window opens.
 1. In the **Collation** list, click on the collation you want to use.
 2. In the **Recovery model** list, click on the recovery model you want the database to use. Then click **OK**.
7. (Optional) In the **Command timeout** box, enter a timeout for the commands you run on the storage.

8. (Optional) In the **Connection timeout** box, enter a timeout for connecting to the storage.
9. If you plan to use Azure AD users in your security roles, select **Use Azure settings** and enter the required options:
 - In the **Tenant ID** box, enter the tenant ID GUID from Azure.
 - In the **Application ID** box, enter the application ID GUID from Azure.
 - In the **Application key** box, enter the application key from Azure.
10. (Optional) Enter any addition connections settings in the **Additional connection properties** box.
11. Click **OK** to add the storage.

Data Sources

The ODX Server can connect to a data source through one of four different types of providers:

- **CDS:** TimeXtender's own providers with tweaks and improvements that usually makes this the best choice if one is available for your data source.
- **CData:** Providers from CData, a third-party supplier of drivers and providers for connecting to a massive amount of different systems. For help with setting up CData providers, please refer to CData online documentation for ADO.net providers at <https://www.cdata.com/kb/help/>
- **ADF:** Providers from TimeXtender that use Azure Data Factory for transferring data.
- **ADO:** Providers installed on the machine that are built on ADO.NET.
- **OLE DB:** Providers installed on the machine that supports the OLE DB standard.

Adding and Synchronizing A Data Source

To add a new data source, follow the steps below.

1. Right click **Data Sources** and click **Add Data Source**. The Add Data Source wizard appears.
2. On the **Name and description** page, enter the following information and then click **Next**.
 1. In the **Name** box, type the name you want to use for the source.
 2. (Optional) The **Short Name** box contains a suggested value based on the name. You can customize this if you like. The short name is used by the ODX Server to identify the source and can be up to 10 characters long.
 3. (Optional) In the **Description** box, type a description of the data source.
3. On the **Select Provider** page, select the provider you want to use and click **Next**. If the provider you selected is not already installed on the ODX server, it will be downloaded from the online repository.
4. On the **Connection Info** page, enter the information needed to connect to the data source and click **Next**.

The content of the page depends on the provider you have chosen. The ODX server reads the list of options available from the provider and lists them in a property grid. In addition to that, different options for how the ODX server should read the source may also be available.

For the most popular providers, a template is available that exposes the information that will usually be required for a successful connection. When this is the case, click **More Options** to show all available options in the property grid.

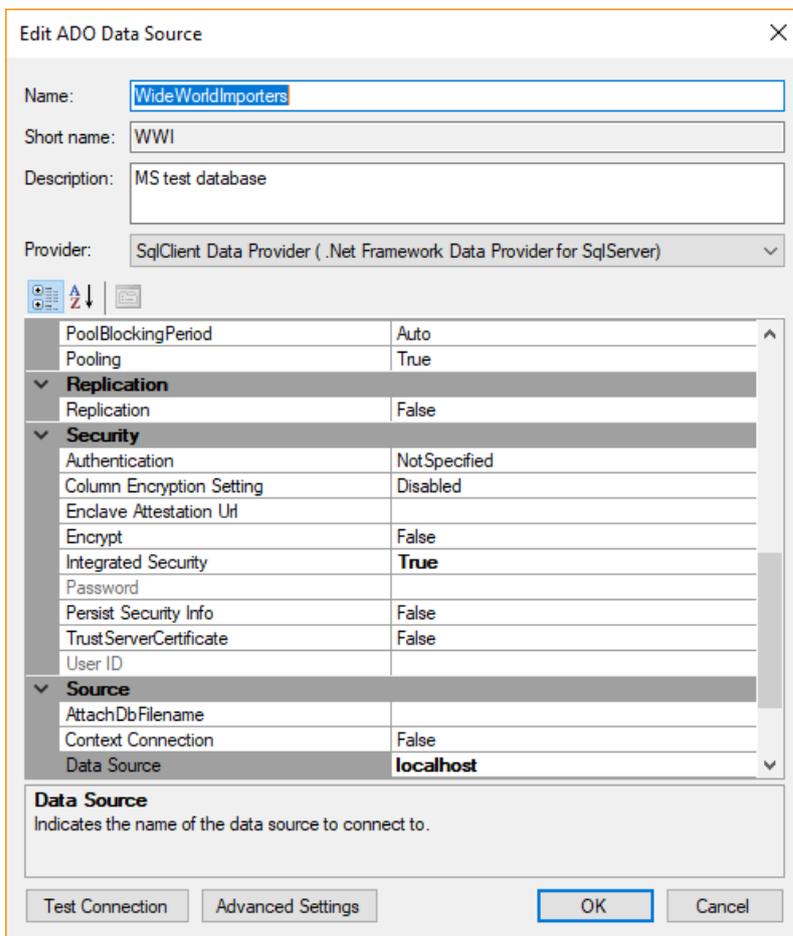
The following are some of the options that may be available:

- **Character Replacements:** Here, you can type a **Character** value to replace with the **Replace with** value in table and column names from the data source.
 - **Command Timeout:** The command timeout in seconds.
 - **Concurrent execution threads:** The maximum number of connections from the ODX server to the data source on execution.
 - **Force codepage conversion**
 - **Query Formatting:** **Prefix**, **Suffix** and **Separator** used in the source. Click **Read Value** to fill in the values automatically if possible.
 - **Schema Properties:** Type the **Schema Name**, **Table Name**, **Column Name** etc.
5. On the **Data** page, choose what tables will be available on the data source and then click **Next**.
 1. **All tables in the data source:** All tables are available. Choose this setting if all tables on the data source makes sense to have in the data storage.
 2. **Let me select the tables:** You select the tables that will be available. This is useful if there are temporary or system tables on the data source that it does not make sense to store on the data storage.
 6. A data selection page is displayed if you chose **Let me select the tables**. For more information, please see [Data Selection and Filtering](#).
 7. On the **Next Step** page, click **No, I'll do that later** if you do not want to add a transfer task after adding the data source and then click **Finish**.

Editing a Data Source's Connection Settings

To edit the connection settings of a data source, follow the steps below.

1. Right click the data source and click **Edit Data Source**. The **Edit Data Source** window appears.



2. In the **Name** and **Description** boxes, you can update the name and description of the data source.
3. Click on a new provider in the **Provider** list to change the provider used by the data source. The providers available depend on the type of provider you initially chose for the data source. You cannot change between, for instance, ADO.net and OLE DB providers.
4. In the property grid, you can change any settings you want to update.
5. Click **Advanced settings** to access additional settings. The **Advanced Data Source Properties** window opens.
 - In the **Query Formatting** list, type the **Prefix**, **Suffix** and **Separator** used in the source. Click **Read Value** to fill in the values automatically if possible.
 - In the **Character Replacements** list, you can type a **Character** value to replace with the **Replace with** value in table and column names from the data source.
 - In the **Schema Properties** list, type the **Schema Name**, **Table Name**, **Column Name** etc.

Synchronizing Objects on a Data Source

Synchronizing objects on a data source loads the meta data from the source into the cloud repository. No data can be copied from a source where objects have not been synchronized since the ODX server would not know what tables are available on the data source.

When you add a data source, a Synchronize task is added automatically. You can schedule it to synchronize the data source on regular intervals or execute the task manually to synchronize the data source immediately.

See [Tasks](#) for more information.

Setting Up Incremental Load

The ODX Server can load data incrementally to get the latest data quickly and with minimal load on the data source.

Since the ODX is built for handling sources with a lot of tables, you do not pick individual tables to load incrementally. Instead, you define the rules for when incremental load should or should not be applied. Each rule consists of a set of conditions and an action. When a condition matches a field, the action is applied to the table. For each field, the first set of conditions that matches is applied.

To add incremental load to a data source, follow the steps below.

1. Right click the data source and click **Set Up Incremental Load**.
2. Click **Add...** to add an incremental load setup rule.
3. Under **Conditions**, define the conditions a field needs to match to have an incremental load rule applied.
4. Under **Actions**, click on the action you want to apply to the fields that match the conditions. You have the following options:
 - **Load records where value is greater than the last maximum value:** Applies incremental load. Select **Handle primary key updates** to update existing records in the storage that have changed and select **Handle primary key deletes** to remove records deleted in the source from the storage.
5. Click **OK** to add the rule.
6. When you have set up all the rules you want to, click **OK**. To handle updates and deletes, the ODX Server needs to know the primary key of the tables that this is enabled on. If you are missing required primary keys, the **Incremental Load Primary Key Validation** window will open to let you select which fields to include in the primary key on the tables in question.

Setting Up Primary Keys

To handle updated and deleted records in the source on incremental load, the ODX server needs to know the primary key on each table. There is a rule-based interface available for defining primary keys.

To set up primary keys for a data source, follow the steps below.

1. Right click the data source and click **Primary Keys....**
2. Click **Add...** to add an incremental load setup rule.

3. Under **Conditions**, define the conditions a field needs to match to be included in or excluded from the primary key.
4. Under **Actions**, click on the action you want to apply to the fields that match the conditions. You have the following options:
 - **Include in primary key**
 - **Exclude from primary key**
5. Click **OK** to add the rule.

Overriding Data Types

To make fields from the data source "fit" the data type on an SQL Server storage, you can add data type overrides.

To add a data type override rule to a data source

- Right click the data source and click **Override Data Types**.

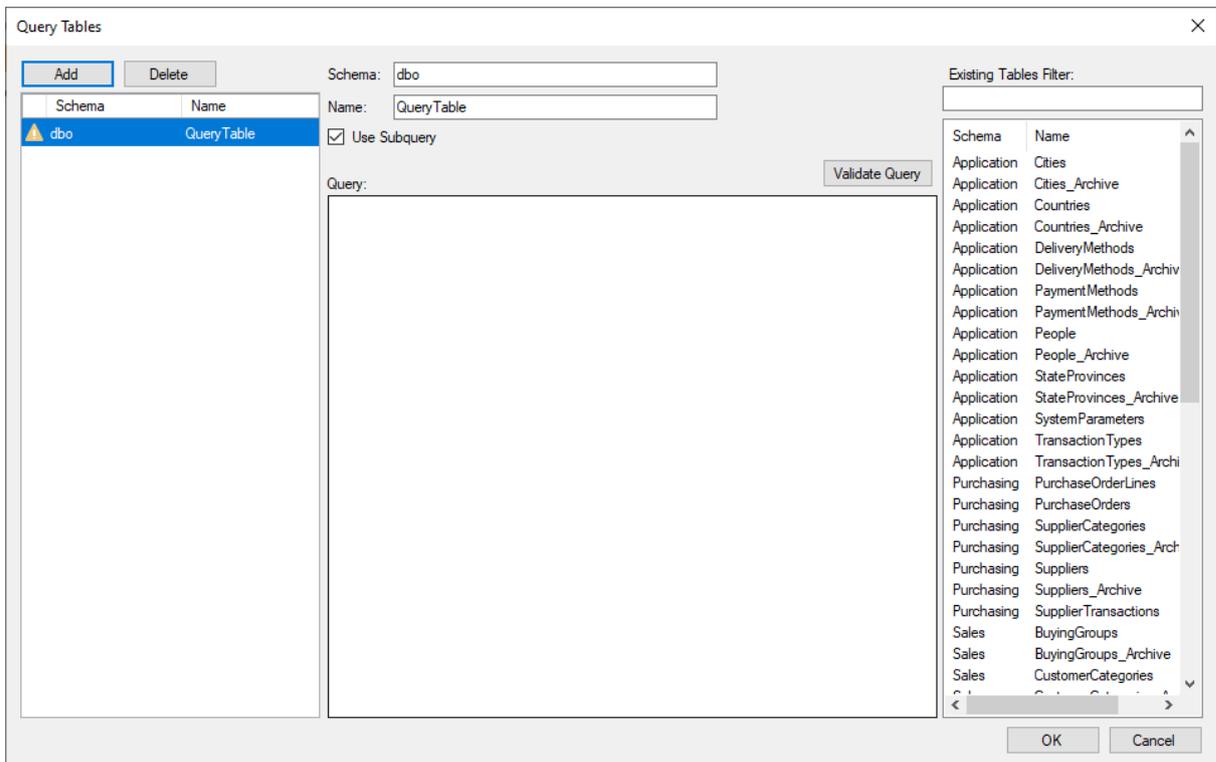
For more information, see [Adding a Data Type Override Rule](#).

Managing Query Tables

With the query tables feature, you can enter a SQL query that the ODX will turn into a table that works just like any other table in the ODX. This is useful, for instance, when a table cannot be extracted from the data source by the regular ODX logic.

To add a query table, follow the steps below.

1. Right click the data source containing the data you want to extract and click **Manage Query Tables**.



2. Click **Add** to add a new query table to the list.
3. (Optional) In the **Schema** box, type the Schema name you want the table to use.
4. In the **Name** box, type a name for the table.
5. In some cases, clearing **Use subquery** can lead to better performance. This includes query tables that will not be incrementally loaded or where the query does not contain an alias or a `WHERE` clause.
6. In **Query**, enter the query you want to use for creating the table. The query should contain a `SELECT` statement and follow the syntax required by the source. You can drag a table in from the list in the right-hand side of the window to add a `SELECT` statement that includes all fields in the table to use as a starting point.
7. (Optional) Click **Validate Query** to check if your query can be used for a query table.
8. If you need to add more query tables, repeat step 2-6.
9. Click **OK** to save the query tables.
10. [Synchronize the data source](#) to make the query table(s) available for transfer to the data warehouse.

Note: [Row filter rules](#) are not applied on query tables.

Browsing and Previewing Data Sources

To select the correct data to move into the data storage and on to the data warehouse, it is important to know what data is in the data source in the first place. In addition to that, it is useful to see how the ODX "sees" the data source. For these kinds of uses, the Data Source Explorer and the Query Tool is available.

The Data Source Explorer works on the meta data stored in the ODX repository. It allows you to browse the tables and columns in the data source to give you an overview of the structure of the data source. In addition to that, you can see how the different rules, you have set up in the ODX, affect the source.

To open the Data Source Explorer

- Right click the data source and click **Data Source Explorer**

Managing Data Source Providers

Custom data sources and CData providers can be downloaded and installed from an online repository when you add a new data source and with the Manage Data Source Providers tool. You can also use the tool to update a provider to the newest version or delete it from the ODX server.

To open Manage Data Source Providers

- Right click **Data Sources** and click **Manage Data Source Providers**.

To add one or more providers, follow the steps below.

1. Open **Manage Data Source Providers**
2. Click **Add...** to install the latest versions of the providers OR click **Add Specific Versions...** from the **Add** list if you want to add a version of a provider other than the latest
3. Select the providers you want to install and click **OK**. The ODX Server will download and install the required files.
4. Click **OK** to close the tool.

To update a provider

- Open **Manage Data Source Providers** , select a provider with Update Available in the Status column and click **Update**.

Note: You can have multiple major versions of the same provider installed, but only one minor version within each major version.

To delete a provider

- Open the tool, select a provider and click **Delete**.

The screenshot shows the 'Data Source Explorer' window. On the left, there is a search area with 'Schema:' and 'Table:' fields, and a list of tables under the 'dbo' schema. The table 'CRONUS Canada, Inc.\$Sales Header' is selected. Below the search area are several checkboxes for filtering: 'Included in transfer task', 'Incremental load rule applied', 'Primary key rule applied', 'Data selection rule applied', and 'Data type override applied'. A 'Showing 3.461 results' label and a 'Search' button are also present.

The main area displays a table with the following columns: 'Field source name', 'Field destination name', 'Primary key rule', 'Incremental load rule', and 'Data selection'. The table lists various fields and their mappings, with checkmarks in the 'Primary key rule' and 'Incremental load rule' columns for 'timestamp' and 'Document_Type'.

Field source name	Field destination name	Primary key rule	Incremental load rule	Data selection
timestamp	timestamp		✓	
Document_Type	Document_Type	✓		
No_	No_	✓		
Sell-to Customer No_	Sell-to_Customer_No_			
Bill-to Customer No_	Bill-to_Customer_No_			
Bill-to Name	Bill-to_Name			
Bill-to Name 2	Bill-to_Name_2			
Bill-to Address	Bill-to_Address			
Bill-to Address 2	Bill-to_Address_2			
Bill-to City	Bill-to_City			
Bill-to Contact	Bill-to_Contact			
Your Reference	Your_Reference			
Ship-to Code	Ship-to_Code			
Ship-to Name	Ship-to_Name			
Ship-to Name 2	Ship-to_Name_2			
Ship-to Address	Ship-to_Address			
Ship-to Address 2	Ship-to_Address_2			
Ship-to City	Ship-to_City			
Ship-to Contact	Ship-to_Contact			
Order Date	Order_Date			
Posting Date	Posting_Date			
Shipment Date	Shipment_Date			
Posting Description	Posting_Description			
Payment Terms Code	Payment_Terms_Code			

At the bottom of the window, there is a 'Preview' button and a status bar showing 'Last synchronization with source: 02-10-2020 12:45:07' and a 'Close' button.

You can search for specific tables and filter the search results to see only tables that e.g. are being transferred to data storage or support incremental load. On the field list, two names are listed for each field. The **Field source name** is the name of the field in the source, while the **Field destination name** is the name used in the ODX data storage and onwards to the data warehouse. Check marks indicate fields that are "hit" by a primary key,

incremental load or data selection rule. The data type of the field is also displayed, along with the new data type if a data type override rule is applied to the field.

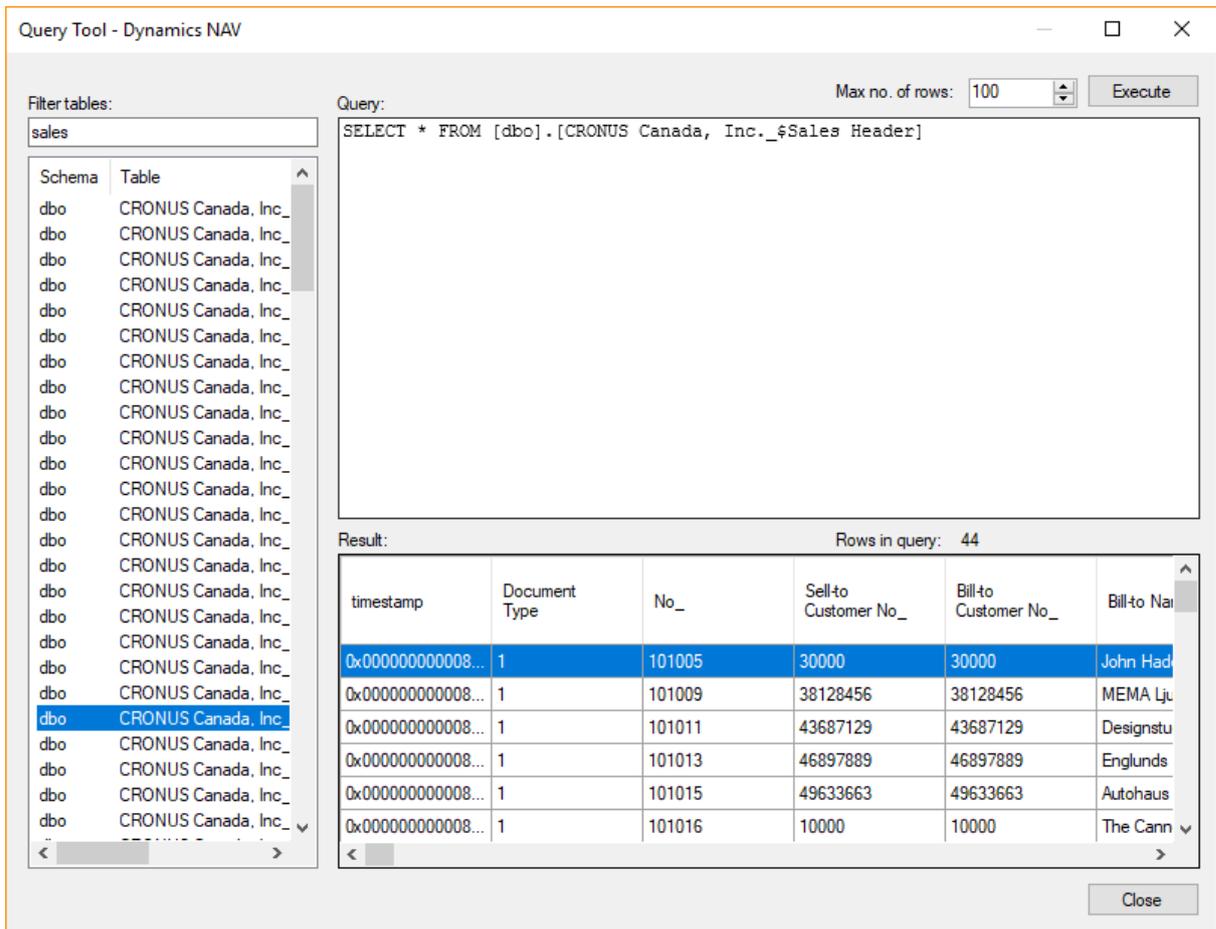
To open the Query Tool from the Data Source Explorer with data from the selected table listed

- Select a table and click **Preview**

The Query Tool works directly on the source and allows you to preview the data stored in the data source using various SQL queries.

To preview a table in the Query Tool, follow the steps below.

1. Right click the data source and click **Query Tool**.



2. Drag a table from the list in the left-hand side to the **Query** box to generate a `SELECT` statement that selects all rows from the table. You can edit this statement, for instance adding a `WHERE` clause.
3. (Optional) In **Max. no of rows**, enter the maximum number of rows you want the query to return.
4. When you are ready to run the query, click **Execute**. In the **Results** list, the returned rows will be listed.

The query tool has a few other features:

- As it is standard in other tools, the Query Tool will execute any selected text in the **Query** box instead of everything when you click **Execute**.
- If the **Query** box is not empty when you drag in a table, a complete `SELECT` statement will not be generated. Instead, the schema and table name, e.g. `[dbo].[Customers]` will be inserted.

Data Selection and Filtering

The ODX allows you to select data on two levels - the data source and the transfer task.

On the data source level, you select the data that should be available for tasks to transfer to data storage. This means filtering out the tables, columns and rows that would never be useful in the ODX storage and data warehouses. For instance, your data source might contain temporary or system tables that only make sense for the source system or very old data that are never accessed or relevant for reporting.

Data selection on the transfer task level gives you control over exactly what tables are transferred when and how. For instance, you could schedule bigger tables for both daily incremental loads and weekly full loads while full-loading smaller tables every day.

Note: The ODX philosophy differs from schools of data warehousing that put emphasis on only copying the tables and columns that are actually used in a report. When using an ODX, the recommendation is to copy all relevant data from data source to ODX data storage. 'Relevant data' is any data that is not obviously irrelevant. In other words, if you are in doubt, include it.

As hinted above, the data selection is more fine-grained on the data source level than the task level. See the table below for an overview.

	Data source	Transfer task
Table level	Selected tables are available for transfer from the data source.	Selected tables are transferred to the data source.
Column level	Selected columns are available for transfer from the data source.	Tables are transferred with the columns and rows selected on the data source level.
Row level	Rows of data can be filtered out for all or specific tables.	

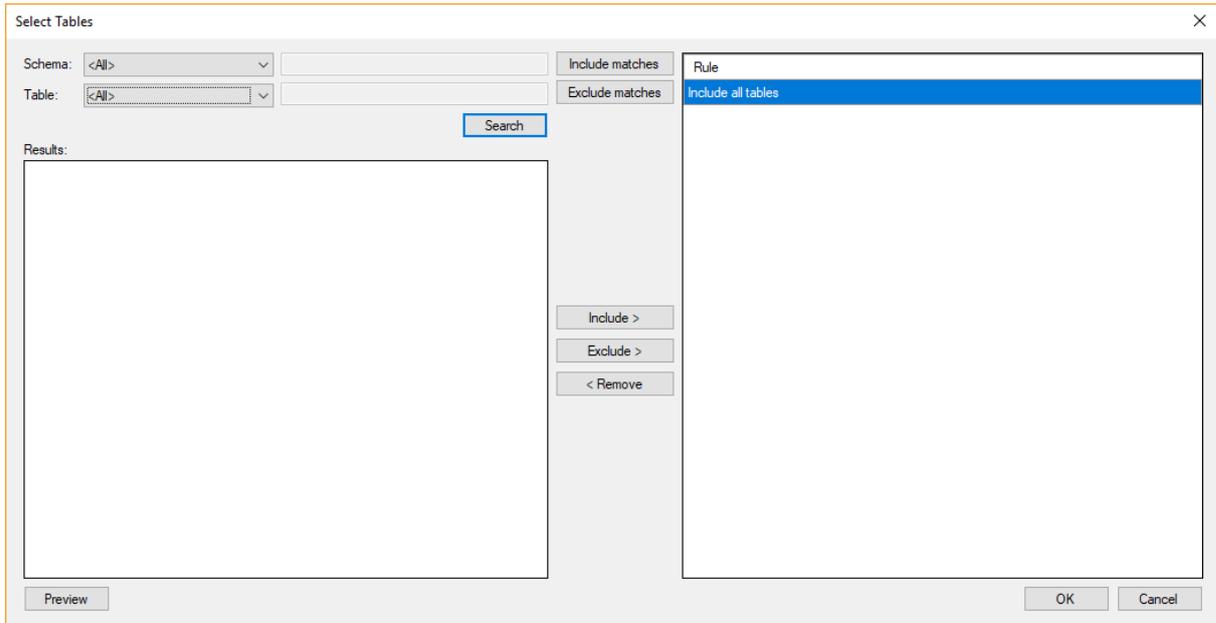
On all levels, data selection in the ODX is based on rules.

- On the table and column level, you can create "include" and "exclude" rules. Exclude rules always take priority over include rules. If the same table is matched by both an exclude and an include rule, the table will not be available or transferred. Per default, a "catch all" include is added to select everything, but if there are no include-rules, no data will be available on the data source or transferred by the task.
- On the row level, the rules include rows of data that match specific schema, table and column values. Per default, all rows from a table is included, i.e. if you don't add any rules, all rows from all selected tables are transferred.

Adding a Table Selection Rule

To add a rule for selecting data, follow the steps below.

1. Right click on the data source or task and click **Select Tables...**



2. Click **Search** to show tables matching the conditions.
3. Click **Include Matches** or **Exclude Matches** to include or exclude tables matching the search conditions. This kind of rule will "catch" any new tables in the data source that match the conditions, not just the tables in the **Results** list. Select one or more tables in the **Results** list and click **Include >** to add these specific tables to the list of available tables or **Exclude >** to exclude them from the list.

Adding a Column Selection Rule

To select the columns that should be available for transfer from the data source, follow the steps below.

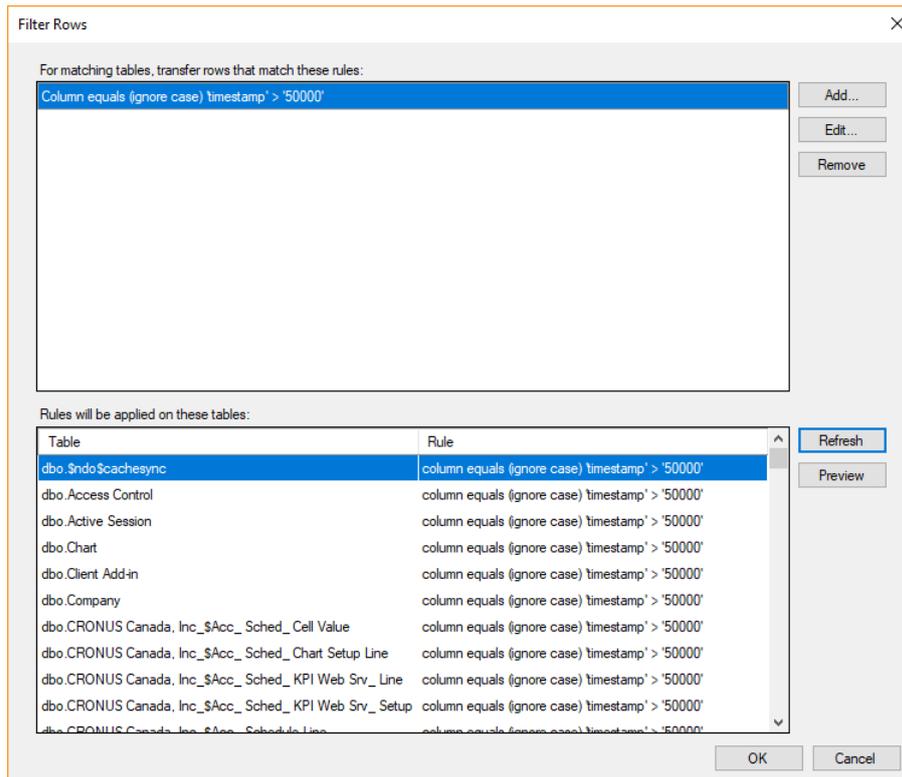
1. Right click the data source and click **Select Columns...**
2. Enter **Schema**, **Table** and **Column** conditions and click **Search** to show tables matching the conditions.
3. Click **Include Matches** or **Exclude Matches** to include or exclude tables matching the search conditions. This kind of rules will "catch" any new tables in the data source that match the conditions, not just the tables in the **Results** list. Select one or more tables in the **Results** list and click **Include >** to add these specific tables to the list of available tables or **Exclude >** to exclude them from the list.

Adding a Row Filter Rule

As mentioned above, all rows of data are transferred from a table unless you add a row filter rule that matches the table. The rule defines the criteria a row needs to have to be included, e.g. an ID over a specific number or a timestamp later than a specific date.

To filter out rows of data from transfers, follow the steps below.

1. Right click the data source and click **Filter Rows...**



2. Click **Add...** to add a new rule.
3. If you want the rule to apply to specific tables, enter the schema and/or table names that they need to match. In the **Schema/Table** list, click on the operator you want to use and type text to match in the box.
4. Click **Add** to add a new column and value combination. In the **Column** list, click on the operator you want to use and type the text to match in the box. In the **Value** list, click on the comparison you want to use and enter a value in the box.
5. Click **OK** to add the rule.
6. (Optional) In the **Filter Rows** window, click **Refresh** to see a list of affected tables. Click **Preview** to open the query tool with a preset query to see what rows will be transferred from a specific table.

Tasks

In the ODX, everything that needs to be able to be scheduled is organized in tasks on a data source. There are three types of tasks:

- **Transfer:** This task moves data from the data source to the data storage.
- **Synchronization:** This task synchronizes the structure of the source with the meta data stored in the ODX repository. A synchronization task is added automatically when you add a new data source.
- **Storage management:** This task performs clean-up and management tasks on the data storage. Execute a storage management task to delete old versions of data and move old versions of data to "cool" storage (applies Azure Data Lake Storage only).

Adding a Task to a Data Source

To add a task to a data source

- Right click the data source, click **Add Transfer Task** or **Add Synchronize Task** or **Add Storage Management Tasks** and follow the instructions in the wizard.

Editing a Task

To change settings for a task, follow the steps below.

1. Right click the task and click **Edit Storage Management Task** or **Edit Synchronize Task** or **Edit Storage Management Tasks**.
2. In the **Name** and **Description** boxes, you can update the name and description of the task.
3. The other settings available depend on the task type.
 - Transfer
 - Select **Use incremental load when available** to load data incrementally into the data storage when the task is executed. If no incremental load rules have been added to the tables copied by the tasks, this setting has no effect.

Note: This setting is not available in the **Add transfer task** wizard.

- Storage management
 - Select **Delete old versions to free up storage** and enter the number of versions to keep in storage in the **Versions to keep** box to delete all but the specified amount of versions from storage when you execute the task.
 - Select **Move old versions to cool storage to save costs** and enter the number of versions to keep in hot storage in the **Versions to keep in hot storage** box to move all but the specified amount of versions to cool storage when you execute the task.

Note: This setting applies to Azure Data Lake Storage only.

Executing a Task

Executing a task will transfer data from data source to data storage, respecting table selection on the data source and the task itself.

To execute a task

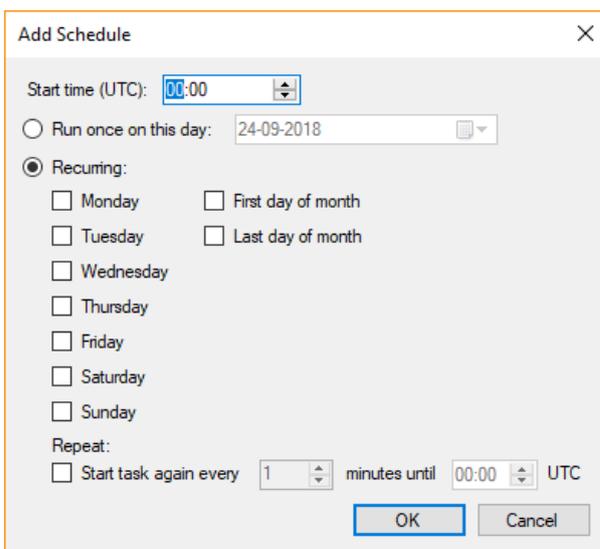
- Right click the task and click **Execute**. The status of the execution will displayed in parenthesis after the task name in the tree.

Scheduling a Task

You can schedule at task to have it execute at predefined times and intervals.

To schedule a n execution of a task, follow the steps below.

1. Right click the task and click **Manage Task Schedule**. The **Manage Task Schedule** window appears.
2. Click **Add** to add a new item to the schedule.



3. In the **Start time** box, enter the time where the execution should begin.
4. Click **Run once on this day** and enter a date if you want to schedule the task to run just once on a specific day.
5. Click **Recurring** to schedule the task to run a regular intervals. The options are Monday-Friday and the first and last day of a month.
6. Under repeat, click **Start task again** and enter a minute count and end time if you want the task to run repeatedly after first starting.

Selecting Tables

For transfer and storage management tasks, you can select which tables the task should process.

To select tables to process

- Right click the task and click **Select tables...**

See [Adding a Table Selection Rule](#) for more information.

Security

In the Discovery Hub architecture, the ODX storage is the central data repository for the entire organization. Employees are different, also in the data they should have access. For this reason, you can set up permissions to control access to data on the ODX storage in TimeXtender.

You can define roles, add users and groups to these roles, and grant/deny the role access to data. You create the permission setup with rules, just like when you select data, set up incremental load etc. Each rule contains criteria for which schema, table and column names it applies to and whether the role should be granted or denied access.

Note: Azure Data Lake data storages support granting access on the table level, while SQL Server data storages support the column level as well. If you grant/deny access to a column, TimeXtender will ignore the rule if the security setup is deployed to an Azure Data Lake data storage.

Adding a Role

To add a role

- Right click **Roles**, click **Add Role**. and follow the instructions in the wizard.

Managing Role Members

Naturally, you can add and remove members from a role after it has been created.

To add a member, follow the steps below.

1. Right click the role and click **Manage members...**
2. In the search box, enter the name of the user or group you want to add and click **Search**. If you click search without entering a search term, everything is listed.
3. Select the users and groups you want to add as members and click **Add >**.

To remove an existing member, follow the steps below.

1. Right click the role and click **Manage members...**
2. Select the member in the list to the right and click **< Remove**

Managing Permissions

As mentioned above, permissions are based on rules, that can also be added and removed from a role after it has been created.

To add a permission rule, follow the steps below.

1. Right click the role and click **Manage Permissions...**

2. Select a data source, enter criteria for **Schema**, **Table** and, if your data storage is an SQL Server, **Column** and then click **Search**. The tables/ columns are listed under **Results**.
3. Click **Grant Matches** or **Deny Matches** to grant/deny access to all tables/ columns that match the criteria
or
select individual tables/ columns and click **Grant >** or **Deny >** to grant/deny access to specific tables/ columns.

To remove a permission rule, follow the steps below.

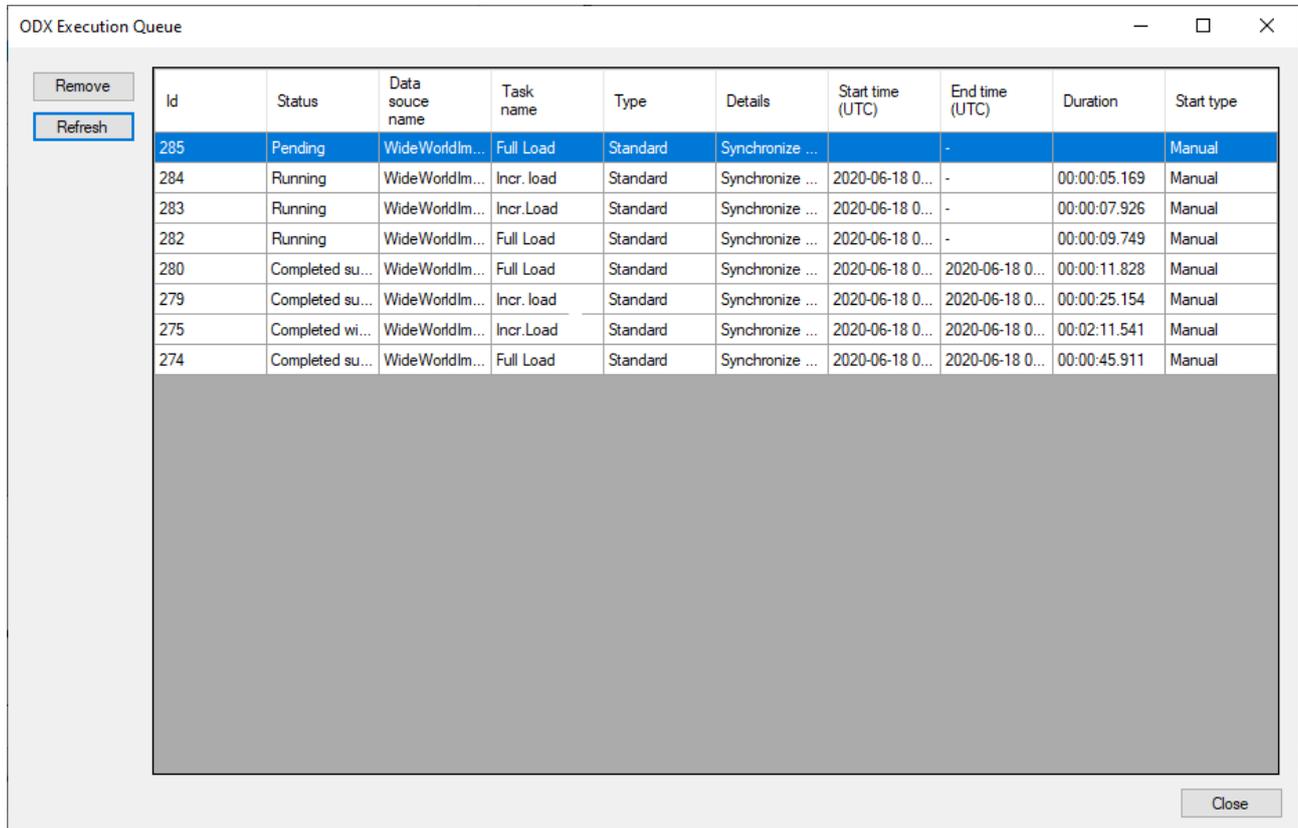
1. Right click the role and click **Manage Permissions...**
2. Select a rule in the list to the right and click **< Remove**.

Execution Queue, Logs and Statistics

You can keep track of what the ODX Server is doing and has done in a number of ways.

View Executing and Queued Tasks

In the Execution Queue, you can see the tasks that are currently executing, waiting to start or just finished executing.



Id	Status	Data source name	Task name	Type	Details	Start time (UTC)	End time (UTC)	Duration	Start type
285	Pending	WideWorldIm...	Full Load	Standard	Synchronize ...		-		Manual
284	Running	WideWorldIm...	Incr. load	Standard	Synchronize ...	2020-06-18 0...	-	00:00:05.169	Manual
283	Running	WideWorldIm...	Incr. Load	Standard	Synchronize ...	2020-06-18 0...	-	00:00:07.926	Manual
282	Running	WideWorldIm...	Full Load	Standard	Synchronize ...	2020-06-18 0...	-	00:00:09.749	Manual
280	Completed su...	WideWorldIm...	Full Load	Standard	Synchronize ...	2020-06-18 0...	2020-06-18 0...	00:00:11.828	Manual
279	Completed su...	WideWorldIm...	Incr. load	Standard	Synchronize ...	2020-06-18 0...	2020-06-18 0...	00:00:25.154	Manual
275	Completed wi...	WideWorldIm...	Incr. Load	Standard	Synchronize ...	2020-06-18 0...	2020-06-18 0...	00:02:11.541	Manual
274	Completed su...	WideWorldIm...	Full Load	Standard	Synchronize ...	2020-06-18 0...	2020-06-18 0...	00:00:45.911	Manual

To open the Execution Queue

- On the **Tools** menu, click **ODX Execution Queue**. The Execution Queue window appears.

To refresh the list of tasks in the Execution Queue

- Open the **Execution Queue** and click on **Refresh** or press **F5**.

To stop a task execution

- Open the **Execution Queue**, click on the running task you want to stop and click **Stop**.

View Scheduled Tasks

The Scheduled Tasks list shows you all the tasks scheduled to be executed.

To open Scheduled Tasks

- On the **Reports** menu, click **ODX Scheduled Tasks**.

View Previously Executed Tasks

The Execution Log shows you all the previous task executions.

The screenshot shows the 'Execution Log' window. On the left, there is a 'Filter' section with 'From' and 'To' date/time pickers, a 'Type' section with radio buttons for 'All', 'Standard', 'Outbound (transfer)', 'Outbound (preview)', 'Clean-up', and 'Security Setup', and a 'Start type' section with radio buttons for 'All', 'Manual', and 'Scheduled'. An 'Update' button is at the bottom of the filter section. The main area contains a table of task executions and a 'Task details' section below it.

Id	Status	Data source name	Task name	Type	Details	Start time (UTC)	End time (UTC)	Duration	Start type
20	Pending	WideWor...	Incremen...	Standard	Transfer ...		-		Manual
19	Running	WideWor...	Full load	Standard	Transfer ...	2020-09-...	-	00:00:47...	Manual
18	Complete...	WideWor...	Synchron...	Synchron...	Synchron...	2020-09-...	2020-09-...	00:00:02...	Manual
17	Complete...	SelfService	Synchron...	Synchron...	Synchron...	2020-09-...	2020-09-...	00:00:06...	Manual
16	Complete...	NAV demo	Synchron...	Synchron...	Synchron...	2020-09-...	2020-09-...	00:01:46...	Manual
15	Complete...	SelfService	Synchron...	Synchron...	Synchron...	2020-09-...	2020-09-...	00:00:08...	Manual
14	Complete...	NAV demo	Synchron...	Synchron...	Synchron...	2020-09-...	2020-09-...	00:03:16...	Manual
13	Complete...	NAV demo	Synchron...	Synchron...	Synchron...	2020-09-...	2020-09-...	00:00:42...	Manual

Task details:

ID	Severity	Timestamp (UTC)	Message
192	Warning	2020-09-24 08:51:09.673	The column [Website].[Suppliers].[CityID] is hidden. The column is ignored
191	Warning	2020-09-24 08:51:09.673	The column [Website].[Suppliers].[DeliveryMethodID] is hidden. The column is...
190	Warning	2020-09-24 08:51:09.673	The column [Website].[Suppliers].[PersonID] is hidden. The column is ignored
189	Warning	2020-09-24 08:51:09.673	The column [Website].[Suppliers].[PersonID] is hidden. The column is ignored
188	Warning	2020-09-24 08:51:09.673	The column [Website].[Suppliers].[SupplierCategoryID] is hidden. The column i...
187	Warning	2020-09-24 08:51:09.635	The column [Website].[Customers].[CityID] is hidden. The column is ignored
186	Warning	2020-09-24 08:51:09.635	The column [Website].[Customers].[DeliveryMethodID] is hidden. The column i...
185	Warning	2020-09-24 08:51:09.635	The column [Website].[Customers].[BuyingGroupID] is hidden. The column is i...
184	Warning	2020-09-24 08:51:09.635	The column [Website].[Customers].[PersonID] is hidden. The column is ignored
183	Warning	2020-09-24 08:51:09.635	The column [Website].[Customers].[PersonID] is hidden. The column is ignored

A 'Close' button is located at the bottom right of the window.

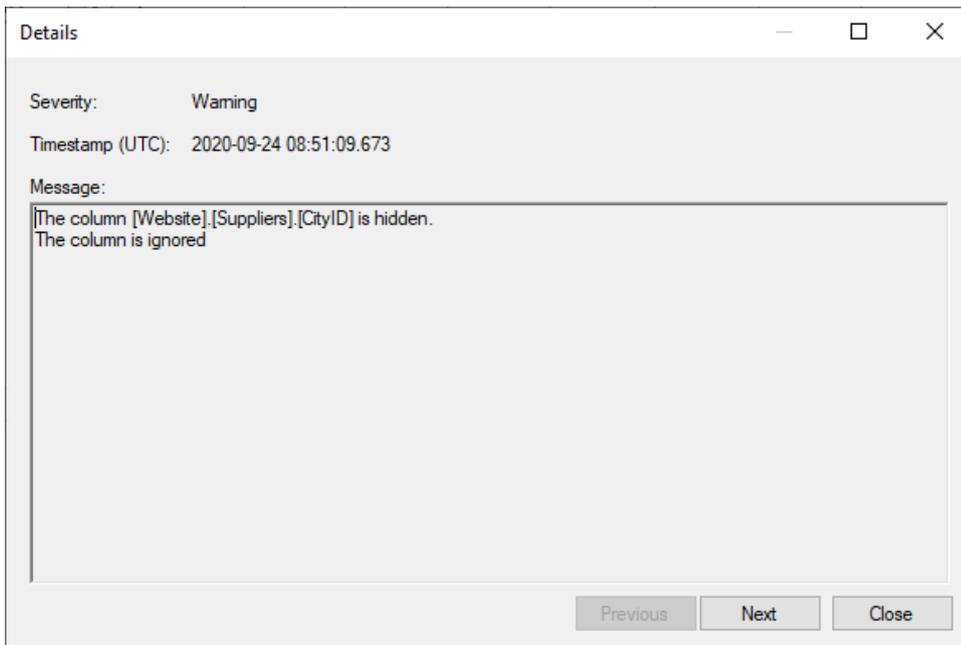
To open the Execution Log

- On the **Reports** menu, click **ODX Execution Log**.

In the left-hand side of the Execution Log window, some filter options are available:

- Time span (**From** and **To** dates and time)
- **Type** of task
- **Start type**

To see the full message for an item in the **Task details** list, double-click the item.



Click **Previous** or **Next** to navigate to other messages in the selected execution.

View Log of Service Calls

The ODX Server also logs the communication between the ODX Server Manager and itself.

To open the Service Log

- On the **Reports** menu, click **ODX Service Log**.

View Data Source and Data Storage Statistics

The ODX server collects some statistics on the data source and data storage. On the data source, you can see e.g. how many tables and columns are available. On the data storage the number of tables and columns and the total data consumption is listed.

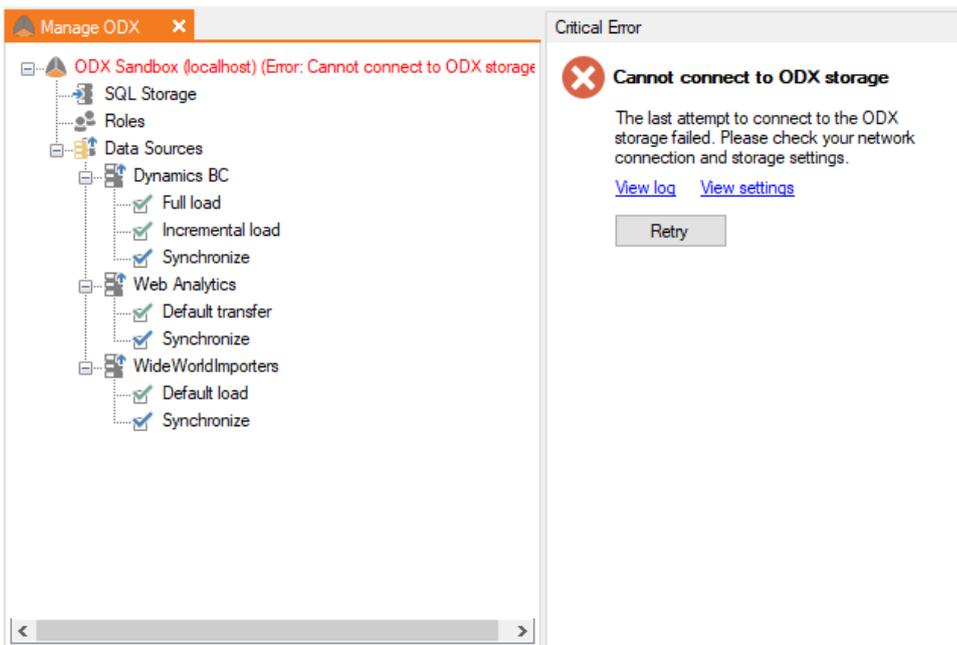
To view the statistics

- Right click a data source or the data storage and click **View Properties**.

Notifications on Critical Errors

The ODX is designed to run on a server without user interaction. Once your data sources have been set up and tasks scheduled, data flows from sources to data storage automatically. As long as your data sources stay the same and do not change, there is little reason to open the management interface.

However, this also means that if errors occur, it might take longer before you notice that something is wrong. For this reason, you can set up email notifications to be sent out when a critical error occurs. And should you open the management interface when a critical error has happened, it is very visible.



The following is treated as a critical error:

- The ODX cannot connect to the cloud repository
- The ODX cannot connect to the data storage
- The client secret the ODX tries to connect with is invalid

Adding an Email Notification For Critical Errors

To set up the ODX to send an email when a critical error occurs, follow the steps below.

1. In the **Solution Explorer**, right click the ODX and click **Notifications on Critical Errors**.

Notifications on Critical Errors

Notify by email when a critical error occurs

Recipients: ⓘ

Email Server

Server name:

Port:

From email:

User name:

Password:

Test Notification OK Cancel

2. Select **Notify by email when a critical error occurs** and enter the recipients in the **Recipients** box. Separate recipients with a semicolon.
3. Under **Email server**, enter the settings for your outgoing email server.
4. Click **Test Notification** to test if the email server settings are working.

Connecting to Data Sources

Note: This chapter does not apply if you use an ODX server. Instead of connecting to data sources through the business unit, you should [connect to data sources through the ODX server](#).

Data sources contain the data that you want to load into your data warehouse and use for analysis. TimeXtender supports two different approaches for connecting to source systems: simple data sources and intelligent application adapters.

A data source connector simply connects to the source and enables you to browse the content of the source. The current version of TimeXtender connects to the following data sources:

- [IBM DB2](#)
- [IBM Informix](#)
- [Microsoft Excel files](#)
- [Microsoft SQL Server](#)
- [Oracle](#)
- [Oracle MySQL](#)
- [Generic ODBC](#)
- [Text files](#)

In addition to that, TimeXtender can connect to other data sources using one of the following approaches:

- [AnySource OLE DB/ ADO.NET](#)
- [Custom Data Sources](#)
- [CData Data Sources](#)

An adapter is a component that enables you to easily extract and synchronize data from different source systems. The adapter knows how a given system organizes and stores data, which enables the adapter to simplify the table structure you see in TimeXtender.

For instance, data for each company in a Dynamics Business Central (NAV) system is stored in a separate set of tables. The Dynamics Business Central (NAV) adapter merges these tables together and lets you select companies on a global level.

TimeXtender includes Application Adapters for the following systems:

- [Infor Movex/ M3](#)
- [Microsoft Dynamics AX](#)
- [Microsoft Dynamics CRM](#)

- [Microsoft Dynamics GP](#)
- [Microsoft Dynamics Business Central \(NAV\)](#)
- [Salesforce](#)
- [SAP](#)
- [Sun Systems](#)
- [UNIT4 Agresso](#)

Data Extraction

Once you have added and configured a data source, the next step is to select the tables and fields you want to copy into the staging database. For more information, see [Moving Data from a Data Source to a Staging Database](#).

Selection templates offer another way of selecting what data to copy into the staging database. For more information, see [Selection Templates](#).

This section covers the settings for data extraction associated with data sources in general.

Integration services

On most data sources, you will find the setting **Use Integration Services For Transfer** that determines how TimeXtender transfers data between a source and a destination table. You have the following options:

- **As Parent:** The setting will be taken from the project setting 'Use Integration Services'.
- **Yes:** A SQL Server Integration Services (SSIS) package is used. This requires that the SQL Server component Integration Services is installed on the machine that deploys and executes the tables.
- **No:** ADO .NET is used. This does not require any SQL components except SQL Server Management Objects.

Using SSIS packages for transferring data is generally considered to be faster for transferring large amounts of data. When transferring data between tables with fewer records, using ADO .NET can sometimes be faster, as it takes time to load the SSIS packages from the SQL Server where they are stored before data transfer can begin.

It takes significantly longer to deploy SSIS packages than it does to deploy ADO .NET transfer. This setting can have a great impact on the overall deployment time of a project.

SSIS and ADO .NET use different technologies to transfer data. This means that if you get erroneous data through SSIS, you can sometimes get correct data using ADO .NET.

Simple Mode

Simple mode is a setting aimed at maximizing performance when you need to copy large amounts of data to create an exact copy. When a table is in simple mode, everything but the most basic functionality is disabled:

- Tables in simple mode do not support field transformations, field validations, conditional lookup fields.
- Tables in simple mode only have the valid instance of the table unless incremental load is enabled.

Per default, a data source inherits the simple model setting from the business unit, but you can override the setting on the data source and on individual tables on the data source.

To enable simple mode for a data source

- Right click the data source, click **Data Source Settings** and click **Enable** under **Simple mode**.

Guarding a Data Source

You can guard a data source in TimeXtender, which prevents tables that get their data from the data source from being deployed, executed or both. In general, the guard feature is useful for tables that contain data that never changes, e.g. from a legacy system. You can also guard a single table. See [Guarding a Table](#) for more information.

To guard a data source

- Right click the data source and click **Data Source Settings**. Select **Guard on deployment** to prevent TimeXtender from deploying the table and/or **Guard on execution** to prevent TimeXtender from executing the table.

Limiting Concurrent Transfers on Data Source

You can put a limit on the number of concurrent transfers from a specific data source. Some data sources can only handle a certain number of transfers before adding more transfers will actually slow down the overall performance of the transfer rather than speed it up.

To limit the amount of concurrent transfers from a data source

- Right click the data source, click **Data Source Settings** and enter the allowed number in the **Max concurrent transfers** box. "0" equals unlimited.

Allowing a Data Source to Fail

If you have some source systems in your solution that are less than critical for your reporting, you can configure your solution so that the entire execution does not stop just because TimeXtender cannot reach these noncritical systems. Instead, you can choose to keep the newest data from the system in question until fresh data can be fetched.

Follow the steps below to allow a data source to fail.

1. Right click the data source and click **Data Source Settings**.
2. Under **When transfer fails**, click the option you want to use if the transfer should fail. You have the following options:

- **Fail and stop the execution:** Stops execution and reports the execution as failed (default setting).
- **Continue execution without data:** Continues the execution, pretending that the source contains no data.
- **Continue with existing data:** Continues execution, retaining the existing data from the source.

Note: If you use additional connectors, you need to configure this setting on the additional data source as well as the template data source.

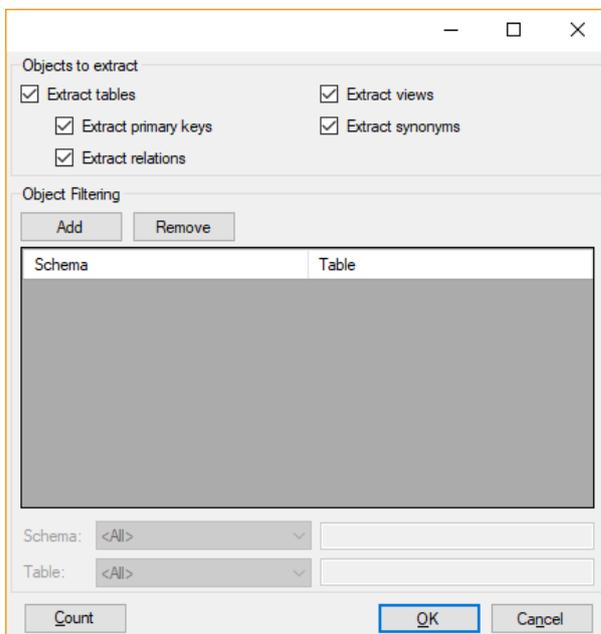
If a data source, that you have allowed to fail, fails during execution, the following execution message will be recorded: "Execution was successful, but one or more data sources failed".

Filtering What Objects to Extract

For large data sources, the amount of tables and fields can make it hard to maintain an overview when you select what data you want to use in your project. To enable you to limit the number of objects brought into your project before the data selection stage, TimeXtender contains a data extraction filter.

The data extraction filter is available on most data sources. To apply a data extraction filter, follow the steps below.

1. Right click the relevant data source, click **Edit [data source]** and click **Data Extraction Settings**



2. Under **Objects to extract** clear the checkboxes next to the object types you do not want to extract.

3. Under **Object filtering**, click **Add** to add a rule that tables and views must pass to be included in the extraction. For instance, a "like: sales" rule in the schema column means that only tables and views in the "sales" schema will be included.
4. Click **OK** to close the window and click **OK** in the **Edit [data source]** window to save the filter.
5. Right click the data source and click **Synchronize Data Source** to apply the filter.

Disabling the Column Cache

When you select data for extraction from a source, you can filter, sort and group the available tables and columns to find the exact objects you are looking for.

With large data sources, loading the meta data for this can be time and memory consuming. TimeXtender contains an option to toggle the caching of columns to cut the number of objects that are loaded and cached down to a manageable size. The downside is that you can then only filter on table names, not column names.

To disable the column cache

- Right click on the source, click **Data Source Settings** and clear **Enable full column caching** under **Data Extraction**.

Note: This option is not available on Dynamics Business Central (NAV) and Dynamics AX adapters and text file data sources.

Data Type Overrides

You can override the data type of fields on the data source level. While you can choose the data type for each individual field in the data warehouse it is easier to do on the data source level if you use the same field more than once in the data warehouse.

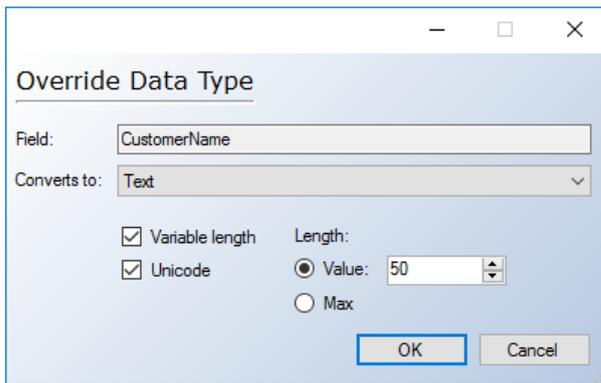
Data type overrides are implemented as rules that can be added and ordered to create a hierarchy. For each field, TimeXtender goes through the rules and applies the first rule that matches if there is any that do.

Under the data source in the tree, a field's original data type is displayed, while the overridden data type is displayed anywhere else.

Adding a Data Type Override to a single Field

To add a data type override to a single field on a data source, follow the steps below.

1. Right click the field and click **Override Data Type**. In the **Converts to** list, click on the data type you want to use and then adjust any settings for the data type you have selected.



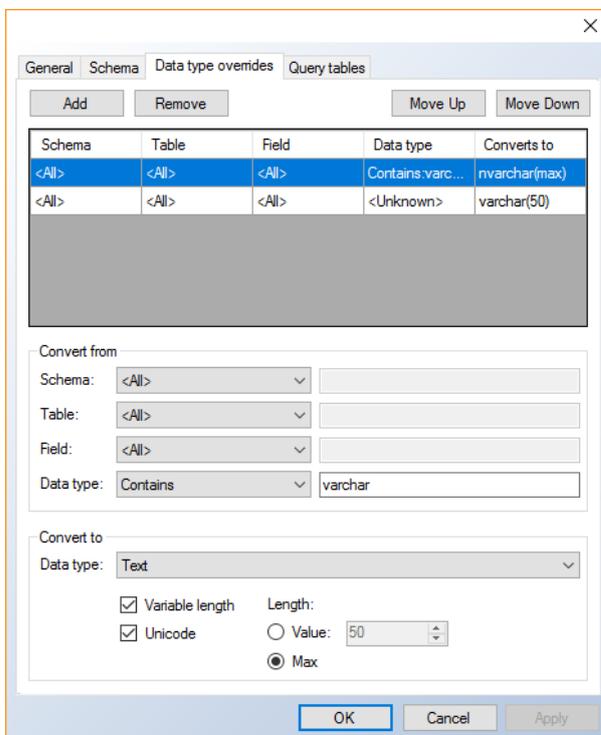
2. Click **OK**.
3. Right click on the data source and click **Synchronize Data Source** to apply the overrides.

Note: This creates a rule that matches the schema, table and field name of the field. If one of these names change, the rule will no longer match.

Adding a Data Type Override Rule

If you want to add a data type override that matches more than a single field, follow the steps below.

1. Right click a data source, click **Data Source Settings** and click the **Data type override** tab.



2. Click **Add**. A new rule is added to the list.

3. Under **Convert from** group, choose what criteria a field should fulfill to match the rule. You can add criteria on schema, table and field name as well as the data type. In the list of operators for the data type, you can click on "<Unknown>" to match data types that are not compatible with SQL Server.
4. Under **Convert to**, click on the data type you want to convert the matching fields to in the **Data type** list. Adjust any settings for the data type you have selected.
5. Click **OK**.
6. Right click on the data source and click **Synchronize Data Source** to apply the overrides.

Ordering data type overrides

Data type overrides are processed for each individual table. The first override from the top that matches the field is applied and any other matches are ignored. Follow the steps below to order the overrides.

1. Right click a data source, click **Data Source Settings** and click the **Data type override** tab.
2. Click on the override you want to reorder in the list and click on **Move Up** or **Move Down** to move the override up or down respectively.

Note: The default rule that matches any unknown data types cannot be reordered. However, you can edit the data type it converts to. The purpose of this rule is to ensure that all fields end up with a data type SQL Server can handle.

Query Tables

When you connect to a data source in TimeXtender, you can simply use the read objects feature to list the content of the source and pick the tables and fields you want to use in your solution.

However, some AnySource providers allows you to connect to a data source, but cannot list the contents of it. In other cases, it is simply useful to be able to create a table, that does not already exist on a data source, from a query. To enable you to use these data sources in TimeXtender, we have included the Query Tables feature. While the SQL behind ordinary tables is created by TimeXtender, you write the query that brings query tables to life.

The following data sources and adapters support the query tables feature:

- AnySource adapter
- Dynamics AX adapter
- Dynamics Business Central (NAV) adapter
- SQL data source

Adding Query Tables

To add a query table, follow the steps below.

1. Connect to a data source using one of the supported data sources or adapters.
2. Right click the data source, click **Data Source Settings** and then click the **Query Table** tab.
3. Click **Add**. A new table is added to the list.
4. In the **Name** box, type a name for the table.
5. (Optional) In the **Schema** box, type a schema to use.
6. In **Query**, enter the query you want to use for creating the table. The query should contain a `SELECT` statement and follow the syntax required by the source.
7. Select **Subquery needed** if you are using an alias in your query. Otherwise, selection rules will fail.
8. Repeat step 3-6 to add the tables you need and click **OK**.
9. Right click the data source and click **Read Objects from Data Source**. The tables are listed in the panel in the right-hand side of TimeXtender and can be included in the project like any other table.

Handling Accounts in Dynamics Business Central (NAV)

When you create query tables for Dynamics Business Central (NAV), you will have to consider how you handle accounts.

To get data from one account, remember the account in the FROM part of your statement:

```
SELECT * FROM [dbo].[MyCompany$MyTable]
```

To get data from multiple accounts, in the same way the Dynamics Business Central (NAV) Adapter does it, you can use placeholders:

```
SELECT * {0} FROM [dbo].[{1}$MyTable]
```

TimeXtender will replace the digits in curly brackets during execution to create the following statement for each account:

```
SELECT
* , CAST('MyCompany' AS nvarchar(30)) AS (DW_Account)
FROM
[dbo].[MyCompany$MyTable]
```

Template Data Sources

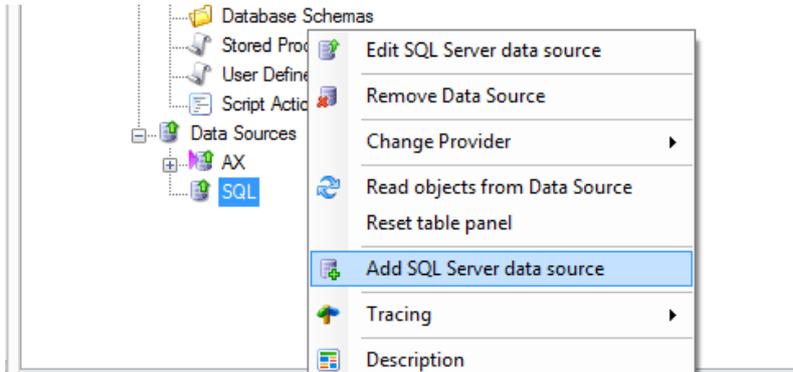
If you have more than one data source with an identical data structure, you can connect all of them together and use them as if they were one and the same. This feature is known as template data sources.

Adding an Additional Connection

Once you have added the first of the identical data sources as usual, you can add the additional data sources to the first data source. Since all of the data sources will have a similar data structure, it does not matter which one is added first.

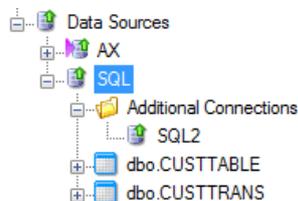
To add an additional connection, follow the steps below.

1. Right-click a data source and click **Add <data source> Data Source**.



An **Add <data source> Data Source** window appears.

2. Configure this second data source with the necessary parameters. This additional data source will then appear under the original data source in an **Additional Connections** folder.



When tables and fields are added to the data source, these changes will seamlessly be applied to all of the data sources that under **Additional Connections**.

The "DW_SourceCode" field on tables on the data source contains the name of the source from which the row was copied. If you have a setup like the one displayed above with two sources, "SQL" and "SQL2", a table on this source would look something like the list below.

Table: SQL_dbo_CUSTTABLE

	ACCOUNTNUM	NAME	CITY	COUNTY	COUNTRYREGION	DW_Id	DW_Batch	DW_SourceCode
▶	902301	Birch Company	Detroit	LAMAR	US	66	1	SQL
	902302	Dolphin Wholesal...	Biramwood	SHAWANO	US	67	1	SQL
	9100	Contoso Europe	Berlin		DE	68	1	SQL
	Contoso	Contoso Standar...				69	1	SQL
	1101	Forest Wholesales	Bothell	SNOHOMISH	US	70	1	SQL2
	1102	Sunset Wholesales	Artesia Wells	LA SALLE	US	71	1	SQL2
	1103	Cave Wholesales	Abbeville	WILCOX	US	72	1	SQL2
	1104	Desert Wholesales	Washington	DISTRICT O	US	73	1	SQL2
	1201	Snowy Wholes...	Avada	JEFFERSON	US	74	1	SQL2

Selection Templates

Once you have selected all the tables you need from a data source, added primary keys and set up incremental load, you can export all this information to a selection template. The template can then be applied to another data source in the same or another project. The data sources do not need to be identical.

Selection templates are stored in XML format.

Exporting a selection template

To export a selection template

- Right click a data source, click **Selection Template** and click **Export**. In the window that appears, chose a file name and folder and click **Save**.

The tables and fields selected, the primary keys chosen and the incremental load setup is saved in the exported file.

Importing a Selection Template

To import and apply a selection template to a data source, follow the steps below.

1. Right click the data source, click Selection Template and click **Import**. The **Import Selection Template** window opens.
2. Click **Browse** to find and select the template to import.
3. Under **Apply the following details from the template**, remove the check mark from any parts of the template you do not want to apply. If you remove the check mark from **Tables and fields**, but keep **Primary keys** and/or **Incremental Load Setup** selected, these settings will be applied to the tables and fields already selected from the data source.
4. Click **OK**.

Changing Data Source Providers

If you have moved your data to a new database, e.g. from Oracle Database to Microsoft SQL Server, you do not have to add a new data source to reflect this. Instead, you can change the type of your existing data source in TimeXtender with the change provider feature.

To change the provider for at data source

- Right click the data source, click on **Change Provider** and click on the database you want to change provider to. A connection settings window appears. Refer to the section on the relevant data source in this user guide to learn more about the settings.

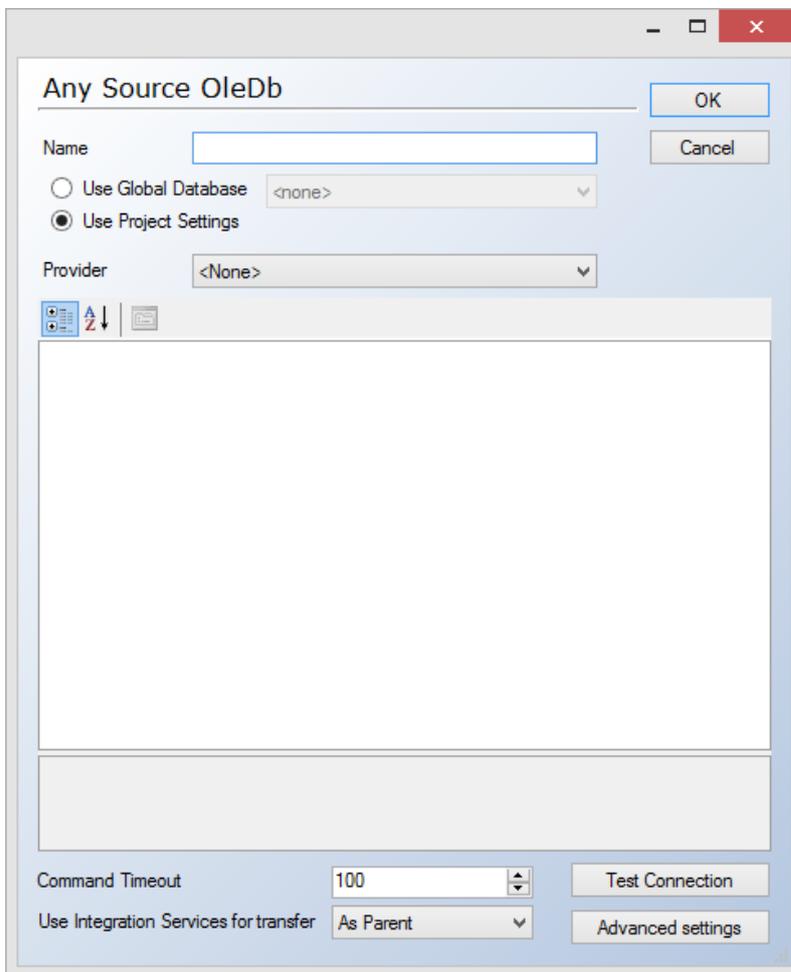
AnySource Data Source

With the AnySource Adapter, you can connect to any data source that you have an OLE DB or ADO provider installed for on your server. Instead of waiting on us to create an adapter that enables you to connect to a specific system, you can now acquire a provider from the system developer or a third party vendor.

Adding a AnySource OLE DB or ADO Data Source

The setup for the AnySource adapter is similar for the OLE DB and ADO based adapter. To add an AnySource OLE DB or ADO data source, follow the steps below.

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add AnySource OLE DB/ Add AnySource ADO**. The **AnySource OLE DB/ AnySource ADO** window opens.



2. In the **Name** box, type a name for the data source.
3. In the **Provider** list, click the provider you want to use.
4. In the property sheet, edit the settings for the connection. See the documentation for the provider you are using for more information.

5. (Optional) In **Command Timeout**, enter the number of seconds after which a command should time out.
6. (Optional) In the **Use Integration Services for Transfer** list, click **As Parent** to use the same setting as the parent business unit or **Yes** or **No** to use or not use Integration Services for transfer, respectively.
7. Click **Test Connection** to verify that the connection is working.
8. Click **Advanced settings** to access additional settings. The **Advanced Data Source Properties** window opens.
9. In the **Query Formatting** list, type the **Prefix**, **Suffix** and **Separator** used in the source. Click **Read Value** to fill in the values automatically if possible.
10. In the **Character Replacements** list, you can type a **Replace Character** to replace with the **Replace Value** in data from the data source.
11. In the **Schema Properties** list, type the **Schema Name**, **Table Name**, **Column Name** etc.
12. In the **Object Filtering** list, you can choose to filter the tables you receive from the provider using regular expressions or different string comparisons. Click **Table** or **Schema** in the **Object Type** list, click a filter type in the **Filter Type** list and type a value in the **Filter Value** List.
13. Click **OK** to close the **Advanced Data Source Properties** window.
14. Click **OK** to close the **AnySource OLE DB/ AnySource ADO** window and add the data source.

Note: On other data sources, you can use the preview feature in TimeXtender to view the content of a table on the source. This might not work on all AnySource data sources since the TimeXtender does not know exactly what the source is and what syntax to use. However, you can use the query tool to explore the content of a table. See [Query Tool](#) for more information.

CData Data Source

A CData Data Source uses a provider from CData to connect to a wide variety of data sources. In this section, you can learn how to add a CData data source. For help with configuring the individual providers, see our support site at <https://support.timextender.com> and CData's help site at cdata.com/kb/help/

Managing Data Source Providers

CData providers can be downloaded and installed from an online library using the **Manage CData Providers** tool. You can also use the tool to update a provider to the newest version or delete it.

To open the **Manage CData Providers** tool

- In the **Tools** menu, click **Manage Data Source Providers**.

To add a new provider, follow the steps below.

1. Open the **Manage Data Source Providers**
2. Click **Add...** OR click **Add Specific Versions...** from the **Add** list if you want to add a version of a provider other than the latest
3. Select the providers you want to install and click **OK**. The required files will be download and installed.
4. Click **OK** to close the tool.

To update a provider

- Open the tool, select a provider with Update Available in the Status column and click **Update**.

Note: You can have multiple major versions of the same provider installed, but only one minor version within each major version.

To delete a provider

- Open the tool, select a provider and click **Delete**.

Providers are downloaded to the folder listed in the **Component Path** box in the **Manage CData Providers** window.

To change the component path

- Click ... (browse), navigate to new folder and click **OK**.

Warning: If you change the component path while you have providers installed, you will have to reinstall these providers.

Adding and Synchronizing A Data Source

To add a new data source, follow the steps below.

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add CData Data Source**.
2. In the **Name** box, type the name you want to use for the source.
3. In the **Provider** list, click on the provider you want to use.
4. In the property grid, enter the information required to connect to the data source.
5. Click **OK** to add the data source.

Custom Data Source

The Custom Data Source works in conjunction with a separate provider - or driver - to enable access to data sources that are not supported by the core TimeXtender product.

Adding a Custom Data Source

The setup for a Custom Data Source depends on the provider you are using. While the general steps to add a Custom Data Source is explained below, you should consult the documentation for the provider to learn more about the specific settings.

To add a custom data source, follow the steps below.

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add Custom Data Source** The **Add Custom Data Source** window appears.
2. In the **Name** box, type a name for the data source.
3. In the **Provider** list, click the provider you want to use.
4. In the **Setup Property** list, click the property you want to edit. Go through all properties in the list and edit the settings for the connection in the property list below.
5. Click **Test Connection** to verify that the connection is working.
6. Click **OK** to close the window and add the data source.

IBM DB2 Data Source

TimeXtender can extract data from IBM DB2 databases.

Adding a DB2 Data Source

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add DB2 Data Source**. The **Add DB2 Data Source** window opens.

The screenshot shows the 'Add Data Source' dialog box for IBM DB2. The dialog is titled 'Add Data Source' and has a close button (X) in the top right corner. It contains several sections: 'Name' with a text input field; 'Use Global Database' with a radio button and a dropdown menu showing '<none>'; 'Use Project Settings' with a selected radio button; 'DB2 Server' section with 'Server name', 'Database', and 'Schema' text input fields, and radio buttons for 'iSeries (IBM)', 'iSeries (Managed)', 'z/OS', and 'UDB'; 'Login' section with 'User name' and 'Password' text input fields and a 'Test Connection' button; and 'Advanced' section with 'Command Timeout' (100), 'Connection Timeout' (15), 'Force character setting' (checkbox), 'Use Integration Services for transfer' (As Parent), and buttons for 'Additional Connection Properties' and 'Data Extraction Settings...'. There are 'OK' and 'Cancel' buttons in the top right area.

2. Enter the connection information.
3. (Optional) Click **Data Extraction Settings** if you want to limit the objects brought into TimeXtender before the data selection stage. For more information, see [Filtering What Objects to Extract](#).
4. Click **OK** to add the data source.

IBM Informix Data Source

TimeXtender can extract data from IBM Informix databases.

Adding a Informix Data Source

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add Informix Data Source**. The **Add Informix Data Source** window opens.

The screenshot shows the 'Add Informix Data Source' dialog box. It features a title bar with a close button (X). The main area is divided into several sections: 'Name' with a text input field; 'Use Global Database' (radio button) and 'Use Project Settings' (radio button, selected) with a dropdown menu showing '<none>'; 'Informix' section with 'Server name' text input and 'Database' dropdown; 'Login' section with 'User name' and 'Password' text inputs; 'Advanced' section with 'Command Timeout' (100), 'Connection Timeout' (15), and 'Use Integration Services for transfer' (As Parent) dropdown. At the bottom are 'Additional Connection Properties' and 'Data Extraction Settings...' buttons. 'OK' and 'Cancel' buttons are in the top right.

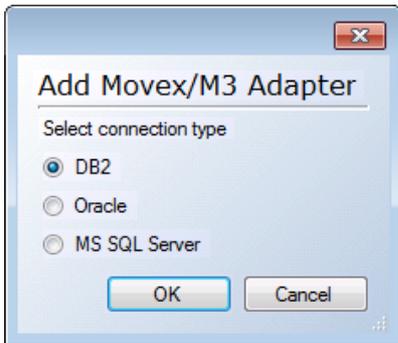
2. In the **Name** box, type a name for the data source.
3. In the **Server Name** box, enter the server name.
4. In the **Database** box, enter the database.
5. In the **Username** box, enter your username. Enter the corresponding password in the **Password** box.
6. (Optional) Click **Data Extraction Settings** if you want to limit the objects brought into TimeXtender before the data selection stage. For more information, see [Filtering What Objects to Extract](#).
7. Click **OK** to add the data source.

Infor M3 (Movex) Adapter

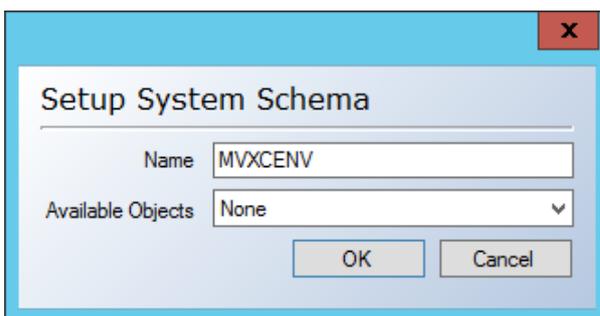
This adapter simplifies extraction of data from Infor M3, an ERP system previously known as Movex.

Adding a Movex/M3 adapter

1. Open a business unit, right click **Data Sources**, click **Adapter Data Sources** and then click **Add Movex/M3 Adapter**. The **Add Movex/M3 Adapter** window opens.

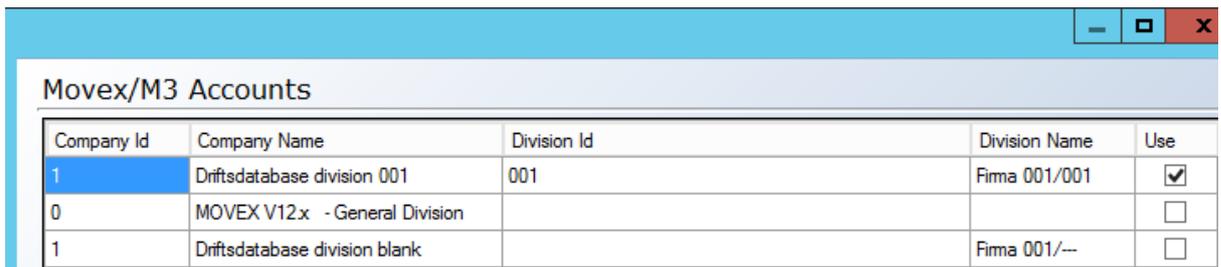


2. Click the database type of your M3 installation. You have the following options:
 - DB2
 - Oracle
 - Microsoft SQL Server
3. Click **OK**.
4. The standard window for adding a data source of the type you chose opens. Enter the connection details. See [Adding a DB2 Data Source](#), [Adding a Oracle Data Source](#) and [Adding a SQL Server Data Source](#) for more information.
5. Right click the M3 data source you just added and click **Change System Schema**. The **Setup System Schema** window opens.



6. Type the environment schema name in the **Name** box.
7. In the **Available Objects** list, click **None** to make no tables available for use in the project, **Company Tables Only** to make only tables from the chosen company available or **All Tables and Views** to make every table and view in the M3 database available. Click **OK**.

8. Right click the M3 data source and click **Setup Accounts**. The **M3/Movex Accounts** windows opens.



Company Id	Company Name	Division Id	Division Name	Use
1	Driftsdatabase division 001	001	Firma 001/001	<input checked="" type="checkbox"/>
0	MOVEX V12.x - General Division			<input type="checkbox"/>
1	Driftsdatabase division blank		Firma 001/--	<input type="checkbox"/>

9. Select the companies and divisions to include by checking the box in the **Use** column. Click **OK**.

Microsoft Dynamics AX Adapter

This adapter simplifies the extraction of data from Microsoft Dynamics AX.

If you connect to a Dynamics AX database as a regular data source, you will have to apply and maintain selection rules on all tables. With TimeXtender Dynamics AX adapter, you can select company accounts at a global level. You can, however, override this behavior on a table by table basis.

The adapter also extracts any virtual company accounts, including, table collections, and tables that are set up in the source database. The information can then be used in dimensions and cubes.

Furthermore, the adapter extracts all Base Enumerations and their associated labels and supports synchronization with the back-end application.

Importing XPO Files into Dynamics AX

The Dynamics AX adapter is only available if the .xpo file has been imported into Dynamics AX.

1. Import the .xpo file into Dynamics AX.
2. Compile the imported project within Dynamics AX.
3. Run all four classes in Dynamics AX to populate the tables.
4. Add a Dynamics AX adapter to your TimeXtender project. For more information, see [Add Dynamics AX Adapters](#).

Adding a Dynamics AX Adapter

Use the Dynamics AX Adapter to load data from separate Dynamics AX company accounts tables in a single table.

1. Open a business unit, right click **Data Sources**, click **Adapter Data Sources** and then click **Add Dynamics AX Adapter**.
2. Enter a name for the adapter, and then click **OK**.

You can now choose the provider which contains the data source you want to connect to.

Adding a Microsoft SQL Server Provider

1. Right-click the adapter, click **Source Providers** and then click **Microsoft SQL Provider**.
2. In the **Server name** box, enter the location of the server.
3. In the **Authentication** list, click the mode of authentication you want to use. You have the following options:
 - **Windows Authentication:** Use the logged-in Windows user's credentials for authentication.
 - **SQL Server Authentication:** Use a login set up on the SQL Server. Enter the username and password in the corresponding fields.

- **Azure AD Password Authentication:** Use Azure AD credentials from a domain that is not federated with Azure AD. Enter the username and password in the corresponding fields.
 - **Azure AD Integrated Authentication:** Use the logged-in Windows user's credentials for authentication, provided that he is logged in using Azure AD credentials from a domain that is federated with Azure AD.
4. In the **Database** field, enter the name of the database.
 5. In the **Connection timeout** box, enter the number of seconds to wait before terminating the attempt to connect to the server, and then click OK.
 6. In the **Command timeout** box, enter the number of seconds to wait before terminating the attempt to connect to the database.
 7. (Optional) In the **Encrypt connection** list, you can enable encryption of the connection, which is recommended when you are not in a private network (e.g. when your server is on Azure). You have the following options:
 - **No:** The communication is not encrypted (default).
 - **Yes:** The communication is encrypted. The server's certificate is verified by a certificate authority.
 - **Yes, trust server certificate:** The communication is encrypted. but the server's certificate is not verified. This setting is not recommended for use on public networks.
 8. In the **Use SSIS for transfer** list click on **Yes** to enable SQL Server Integration Services data transfer, **No** to disable it or leave it at **As Parent** to respect the project setting.
 9. Select **Force codepage conversion** to convert all fields to the collation of the data warehouse.
 10. Select **Force Unicode conversion** to declare all alphanumeric fields as **nvarchar**.
 11. Select **Allow dirty reads** to allow reading from the source without locking the table.
 12. If you want to add additional connection settings, click **Additional Connection Properties** . In the **Connection String Properties** window, enter the required connection strings, and then click **OK**.

Adding a Dynamics AX Odata Provider

The version of Dynamics AX released in 2016 - known as Dynamics AX 7 - introduced a new Azure-based data storage option. You can access this through the "AX7 Odata" provider.

The provider uses OAuth 2.0 authentication, which means you need to set up a client in Azure AD to connect - please see this article for more information:

<https://ax.help.dynamics.com/en/wiki/dynamics-ax-7-services-technical-concepts-guide/>

Note that this provider cannot use SSIS transfer.

To add a Dynamics AX OData provider, follow the steps below.

1. Right-click the adapter, click **Source Providers** and click **Add AX7 Odata Provider**. The **AX7 Odata** window appears.

AX7 Odata

Use Global Database: <none>

 Use Project Settings:

Setup Property: DataSource Properties

Authentication	
Password	
Username	
Client	
Client id	
Redirect URI	
Misc	
Authentication URL	https://login.windows.net/XXX.onmicrosoft.c
Command timeout	100
Connection timeout	200
Max. concurrent threads per table	1
Odata URL	https://XXX.cloudax.dynamics.com
String data type	
Max. character length	4000
Max. key character length	100

Authentication URL
Authentication URL.

2. In the **Password** row, enter the password you use to connect.
3. In the **Username** row, enter the username you use to connect.
4. Under client, enter the **Client Id** and **Return URI** you use to connect.
5. In the **Authentication URL** row, enter the URL used for authentication.
6. (Optional) In the **Connection Timeout** row, modify the number of seconds to wait before terminating an attempt to connect to the server.
7. (Optional) In the **Command Timeout** row, modify the number of seconds to wait before terminating an attempt to connect to the database.
8. (Optional) In the **Max. concurrent threads per table** row, modify the number of connections you will allow to a single table.
9. In the **Odata URL** row, enter the URL that you will get the data from.
10. (Optional) In the **Max. character length** row, set the max length in characters for fields that are strings.
11. (Optional) In the **Max. key character length** row, set the max length in characters for key fields that are strings.
12. If you need to connect through a proxy server, click **Proxy settings** in the **Setup property** list. In the **WebProxyApproach** list you have the following options:
 - **NoProxy**
 - **ApplicationProxy**: Use the proxy settings configured on the application level. To adjust the application settings, click **Options** on the **Tools** menu and then click

Proxy Settings. Note that using a proxy server for Internet connections can also be turned on and off from here.

- **SpecificProxy:** Enter the settings - server, port, username and password - to use for the adapter.

Adding an Oracle Provider

1. Right-click the adapter, and select **Source Providers**. Then select **Oracle Provider**.
2. In the **TNS alias** field, type the alias that identifies the database.
3. In the **Owner** list, select the owner of the database.
4. Specify the authentication mode. When you select **Oracle authentication**, you are prompted for a user name and a password.
5. Select **Convert Out of Range Dates to MS SQL min/max** if you want to convert all dates older than January 01, 1753 to 01-01-1753.
6. In the **Connection Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the server.
7. In the **Command Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the database.
8. If you want to set the character encoding to either Unicode or Non-Unicode, select **Force Character Setting**. Then select the preferred character encoding in the list.

Note: Forcing character encoding may affect performance.

If you want to add additional connection strings, click the **Additional Connection Properties** button. In the Connection String Properties window, type the preferred connection strings, and then click **OK**.

Setting the Account Table

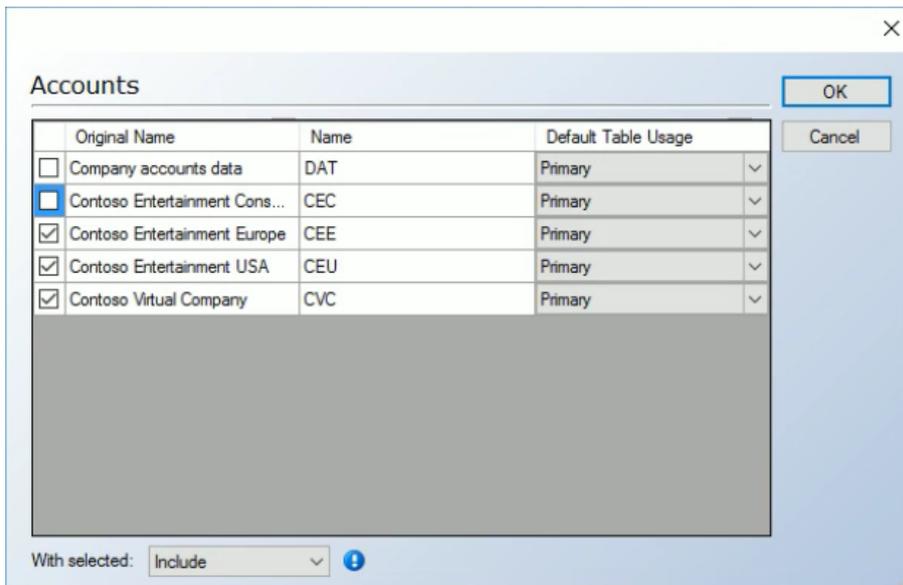
Before you can continue with setting up accounts, you have to verify that the company account table is correct.

1. Right-click the adapter, and then choose **Set Account Table**. The table DATAAREA and the field ID are selected by default.
2. Click **OK**.

Setting Up Dynamics AX Companies

When you have added a Dynamics AX adapter and specified a provider, you need to set up the accounts. To set up accounts:

1. Right-click the adapter, and select **Set Up Accounts**. An information message is displayed, which lists the accounts that have been added. Click **OK**. The **Accounts** window appears.



2. Select the accounts from which you want to retrieve data. In the **With selected** list, click **Include** to use the selected accounts in the data warehouse or click **Exclude** to exclude the selected accounts and include all other accounts. The last option is useful if you often add new accounts in Dynamics AX and want to make sure that all accounts are included in the data warehouse as soon as they exist in the ERP system. Note that the exclude option isn't available when you use the OData provider.
3. In the **Default Table Usage** list, specify the order in which data from the tables is retrieved and read. You have the following options:

Option	Definition
Primary	Data from this company account is read and retrieved first
Secondary	Data from this company account is read and retrieved after the primary account if they have not already been retrieved from the primary account
None	Tables from this company are not retrieved unless you specify at the table level that you want to retrieve data from a specific table. For more information, see Modifying Table Usage on Dynamics AX Tables .

4. Click **OK**.

Loading and Selecting Data from Dynamics AX Data Sources

1. Right-click the Dynamics AX adapter you want to select data from, and then select **Read objects**. The **Data Selection** pane displays all tables and fields.
2. In the **Data Selection** pane, select the tables and fields you want to extract to your staging database.

The tables and fields are displayed in the data source tree and in the staging database tree.

Adding Dynamics AX Virtual Table References

1. Expand the AX adapter that contains the table to which you want to add a virtual table reference, right-click the table, and then choose **Add Virtual Table Reference**.
2. In the Add Virtual Table Reference window, select the preferred virtual tables, and then click **OK**.

Viewing Dynamics AX Table Information

TimeXtender can retrieve table information directly from your Dynamics AX database.

1. Expand the Dynamics AX adapter that contains the table you want to view information about, right-click the table, and then select **View Table Information**. The three tabs in the View Table Information dialog contain the following information:

Fields Tab	Description
Name	Specifies the name of the field as it appears in the database
Label	Specifies the name of the field as it appears in the user interface
Help Text	Contains the help text for the field
EDT Name	Specifies the name of the extended Data Type if applicable
Enum Name	Specifies the name of the enumeration if applicable
System	Specifies whether the table is a system table or visible in the user interface

Relations Tab	Description
External Table	Specifies the name of the table the selected table is related to
Directions	Specifies whether the selected table is the child or the parent in the relation
Field	Specifies which field in the selected table that relates to a field in the related table
External Field	Specifies the field on the related table
Relation Type	Specifies the type of relation. Field specifies relation fields without conditions. ThisFixed specifies relation fields to restrict the records in the primary table. ExternFixed specifies relation fields that restrict the records in the related table

Virtual Company References

Description

Company	The name of the company account
Virtual Company	The name of the Virtual Company that contains tables shared by several company accounts

Viewing Dynamics AX Enum Table Information

All Enum values in Dynamics AX are represented as integers in the tables. However, you can see the corresponding literal values by viewing the enumeration table information.

- Expand the Dynamics AX adapter that contains the table you want to view information about, right-click the table, and then select **Preview Enum Table**.

Changing Dynamics AX Schemas

1. Right-click the AX adapter that contains the table whose priority you want to change, and then select **Change Schema**.
2. In the **Select Schema To Change list**, select the schema you want to change.
3. In the **New Schema Name** field, enter a name for the schema.

Modifying Table Usage on Dynamics AX Tables

When you set up accounts, you specify the default order in which data is retrieved from the individual accounts. However, it is possible to specify a different order of priority for individual tables.

1. Right-click the AX adapter that contains the table whose priority you want to change, and then select **Modify Table Usage**. The company accounts and the usage of all tables will be displayed.
2. Right-click the table and account field with the setting you want to change the order of priority on for data retrieval. You have the following options:

Priority	Definition
Default	Data from this table is read and retrieved first
Primary	Data from this table is read and retrieved first
Secondary	Data from this table are read and retrieved after the primary table if they have not already been retrieved from the primary table
None	Data from this table is not retrieved
1-9	Specify the order priority in the range from 1-9
Enter priority	If the order of priority exceeds the numbers 1-9, you can specify additional numbers

3. Click **OK**.

Modifying the Usage of a Single Dynamics AX Table

If you want to change the order of priority in which data is retrieved on a single table, you can do so from the individual table.

1. Right-click the AX adapter that contains the table whose priority you want to change, and then select the preferred table.
2. Right-click the table, and select **Modify Single Table Usage**. The company accounts and the usage specified in Setup Company Accounts will be displayed.
3. Right-click the field that contains the setting for the table, and then specify the table usage. You have the following options:

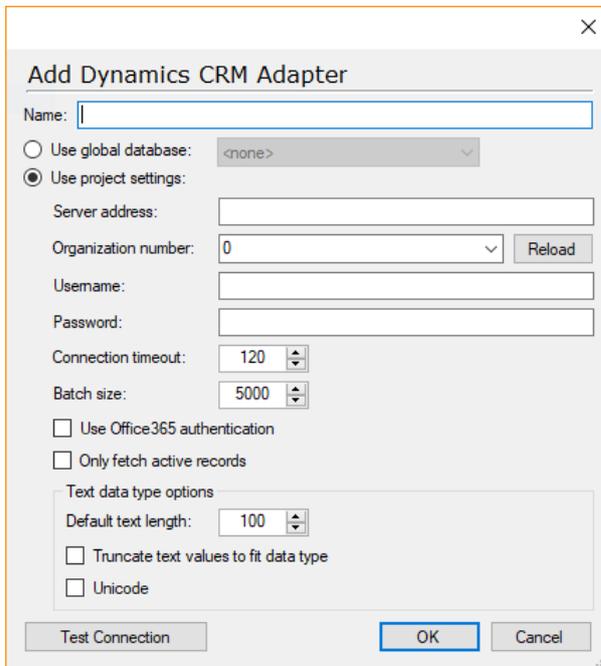
Priority	Definition
Default	Data is retrieved based on the settings specified when you set up the company accounts
Primary	Data from this table is read and retrieved first
Secondary	Data from this table is read and retrieved after the primary table if they have not already been retrieved from the primary table
None	Data from this table is not retrieved
1-9	Specify the order of priority in the range form 1-9
Enter pri- ority	If the order of priority exceeds the numbers 1-9, you can specify additional numbers here.

Microsoft Dynamics CRM Online Adapter

This adapter simplifies extraction of data from Microsoft Dynamics CRM Online.

Adding a Dynamics CRM Application Adapter

1. Open a business unit, right click **Data Sources**, click **Adapter Data Sources** and then click **Add Dynamics CRM Adapter**. The **Add CRM Adapter** window opens.



2. In the **Name** box, type a name for your data source.
3. In the **Server address** box, enter the URL of the Dynamics CRM server.
4. In the **Organization number** list, click on or enter the number of the organization you want to see data from. Click the **Reload** button next to the list to refresh the list of available organization numbers.
5. In the **Username** box, enter the user name you want use for authentication.
6. In the **Password** box, enter the password corresponding to your user name.
7. In the **Connection Timeout** box, type or select the number of seconds to wait for a connection to be established.
8. In the **Batch size** box, type or select the number of records to fetch in each request. Bigger batches uses more memory.
9. Select **Use Office365 authentication** to use Office365 for authentication instead of username and password.
10. Select **Only fetch active records** to fetch only active records.
11. In the **Default text length** box, enter the max length set for data types that does not have a maximum length defined.
12. Select **Truncate text values to fit data type** to make the data fit the data types by removing any extra data.
13. Select **Unicode** to set text fields to Unicode.

14. (Optional) Click **Test Connection** to verify that your settings are correct and then
15. Click **OK** to add the adapter data source.

Microsoft Dynamics GP Adapter

This adapter simplifies the extraction of data from Microsoft Dynamics GP.

If you connect to a Dynamics GP database as a regular data source, you will have to apply and maintain selection rules on all tables because different companies are stored in separate databases. With TimeXtender Dynamics GP Adapter, you can select company accounts at a global level and apply only one set of selection rules. It is, however, also possible to overrule this behavior on a table by table basis.

To Add a Dynamics GP Adapter

Use the Dynamics GP Adapter to load data from separate Dynamics company account databases in a single table.

1. Open a business unit, right click **Data Sources**, click **Adapter Data Sources** and then click **Add Dynamics GPAdapter**.
2. Enter a name for the adapter.
3. In the **Server name** field, enter the location of the server where Dynamics GP resides.
4. In the **Database** field, enter the name of the database (this should be the DYNAMICS database).
5. Specify the authentication mode. The default setting is **Windows authentication**. If you choose **SQL Server authentication**, you are prompted for a user name and a password.
6. In the **Command Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the database. The recommended value is 0 to disable the timeout. In the **Connection Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the server.
7. If you want to add additional connection strings, click the **Additional Connection Properties** button. In the **Connection String Properties** window, type the preferred connection strings, and then click **OK**.
8. (Optional) Click **Data Extraction Settings** if you want to limit the objects brought into TimeXtender before the data selection stage. For more information, see [Filtering What Objects to Extract](#).
9. The next window will be **GP Company Table Setup**. All settings should be left as default and click **OK**.

To Set Up Dynamics GP Companies

After you have added a Dynamics GP adapter and specified a provider, you will set up the accounts which represent the companies for which data will be extracted from the data source.

To set up accounts:

1. Right-click the adapter, and then choose **Read Dynamics GP Companies**. A dialog is displayed that shows all companies in the database.

2. In the **Template** list, select the company account you want to use as template for the table and column structure. If you are only selecting one company, then the template company must match the company that is selected.
3. Select **Use** to specify whether to retrieve data from the company.

To Load and Select Data from Dynamics GP Data Sources

1. Right-click the Dynamics GP Adapter from which you want to select data, and then select **ReadObjects from Data Source**. The Tables pane on the right displays all tables, fields, and views.
2. In the **Tables** pane, select the tables, fields, and views you want to extract to your staging database.
3. There are two ways of viewing the data: Alphabetical view, which displays all tables alphabetically, and Group view, where you specify how many tables each group must contain.
4. To view data in groups, enter the number of tables in each group in the **Group view** field, and then click **Group view**. You can then group the tables alphabetically or by specifying the number of tables you want in each group. To view data alphabetically, click **Alphabetical view**.

The tables and fields are displayed in the data source tree and in the staging database tree.

Microsoft Dynamics Business Central (NAV) Adapter

This adapter simplifies the extraction of data from Microsoft Dynamics Business Central (NAV).

If you connect to a Dynamics Business Central (NAV) database as a regular data source, you will have to apply and maintain selection rules on all tables because different companies are stored in separate tables. With Dynamics Business Central (NAV) adapter, you can select company accounts at a global level and apply only one set of selection rules. It is, however, also possible to overrule this behavior on a table by table basis.

Setup

Adding a Dynamics Business Central (NAV) Adapter

1. Open a business unit, right click **Data Sources**, click **Adapter Data Sources** and then click **Add Dynamics Business Central (NAV) Adapter**.
2. Click **Wizard Setup** and, in the **Choose Dynamics Business Central (NAV) Provider** window, select the provider you want to use:
 - **Microsoft SQL Provider:** Data is stored in SQL Server
 - **Navision Native:** Data is stored in the legacy native Navision format.
3. In the **Name** box, type a name for the adapter.
4. [Optional] Select **Read Aggregation Tables - SIFT** if you need to include Sum Index Flow Technology (SIFT) tables,
5. Click **OK**.
6. The flow continues with setting up the provider you selected. Please refer to the sections below:
 - [Setting Up an Microsoft SQL Server Provider](#)
 - [Setting Up a Navision Native Provider](#)
7. A window appears where you can input the table that contains the company accounts if it is different from the defaults:
 - **Schema:** Default: "dbo"
 - **Table:** Default: "Company"
 - **Field:** Default: "Name"
8. Click **OK**.
9. The **Adapter Settings** window opens.
 - Select **Enable Enhancements** to enable enhancements and enable the following settings.
 - **Display language**
 - **Option field text length**
 - **NAV 2013 or Later:** Select this if the version of Dynamics Business Central (NAV) is 2013 or later.
 - In the **Invalid Identifiers** box, enter any invalid identifiers.
 - Select **Consistent Read** to only transfer records added before an execution is started, i.e. records added or updated during an execution will not be

transferred. This setting only has effect if you use the SQL Server provider.

Warning: Take care when using this option when records are likely to be updated or deleted. If a record is updated during execution, the timestamp will change and subsequently, the updated record will not be copied to the staging database during execution.

Setting Up an Microsoft SQL Server Provider

1. Right-click the adapter, click **Source Providers** and then click **Microsoft SQL Provider**.
2. In the **Server name** box, enter the location of the server.
3. In the **Authentication** list, click the mode of authentication you want to use. You have the following options:
 - **Windows Authentication:** Use the logged-in Windows user's credentials for authentication.
 - **SQL Server Authentication:** Use a login set up on the SQL Server. Enter the username and password in the corresponding fields.
 - **Azure AD Password Authentication:** Use Azure AD credentials from a domain that is not federated with Azure AD. Enter the username and password in the corresponding fields.
 - **Azure AD Integrated Authentication:** Use the logged-in Windows user's credentials for authentication, provided that he is logged in using Azure AD credentials from a domain that is federated with Azure AD.
4. In the **Database** field, enter the name of the database.
5. In the **Connection timeout** box, enter the number of seconds to wait before terminating the attempt to connect to the server, and then click OK.
6. In the **Command timeout** box, enter the number of seconds to wait before terminating the attempt to connect to the database.
7. (Optional) In the **Encrypt connection** list, you can enable encryption of the connection, which is recommended when you are not in a private network (e.g. when your server is on Azure). You have the following options:
 - **No:** The communication is not encrypted (default).
 - **Yes:** The communication is encrypted. The server's certificate is verified by a certificate authority.
 - **Yes, trust server certificate:** The communication is encrypted. but the server's certificate is not verified. This setting is not recommended for use on public networks.
8. In the **Use SSIS for transfer** list click on **Yes** to enable SQL Server Integration Services data transfer, **No** to disable it or leave it at **As Parent** to respect the project setting.
9. Select **Force codepage conversion** to convert all fields to the collation of the data warehouse.
10. Select **Force Unicode conversion** to declare all alphanumeric fields as **nvarchar** .
11. Select **Allow dirty reads** to allow reading from the source without locking the table.

12. If you want to add additional connection settings, click **Additional Connection Properties** . In the **Connection String Properties** window, enter the required connection strings, and then click **OK** .

Setting Up a Navision Native Provider

When you want to retrieve data from Navision databases hosted in a Native Navision server environment, you will have to use ODBC. The Navision ODBC driver must be installed and configured prior to adding the Native NAV data source.

1. Expand the business unit, and then right-click **Data Sources** .
2. Point to **Adapter Data Sources** , and select **Add Dynamics Business Central (NAV) Adapter**.
3. Select Wizard Setup.
4. Select **Navision Native**.
5. In the **Name** field, type the name of the data source.
6. In the **DSN Name** , select the ODBC connection that you have configured for the data source. In the **Escape Character list**, select the escape character specific to your ODBC driver. The Text Type Behavior fields are used to control how the ODBC driver handles text. These fields are optional. You have the following options:

Option	Definition
Set Length	Specifies an exact text string length
Set Variable Length	True, if you want a variable text string length
Set Unicode	True, if you want to use Unicode

7. (Optional) In **Set Number of Decimals** , specify a fixed number of decimals.

Note: "Convert Out of Range Dates to MS SQL min/max" is not available for Navision native databases.

8. Select **Use low compatibility mode** if you have trouble retrieving data from the database.
9. In the **Command Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the database.
10. In the **Connection Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the server.
11. If you want to add additional connection strings, click the Additional Connection Properties button. In the Connection String Properties window, type the preferred connection strings, and then click **OK** .

Dynamics Business Central (NAV) Companies

Changing the Dynamics Business Central (NAV) Company Table

By default, when you add a Dynamics Business Central (NAV) adapter, the company account table is set to `dbo.Company`.

However, it is possible to change the account table. This is generally not recommended.

1. Right-click the Dynamics Business Central (NAV) adapter, and then click **Edit Account Table**.
2. In the **Table** list, select the account to table that you want to use.
3. In the **Name Field** list, select the field that contains the account name, and then click **OK**.

Setting Up Dynamics Business Central (NAV) Companies

When you have added a Dynamics Business Central (NAV) adapter and specified a provider, you need to set up accounts representing the companies requiring the extracted data.

To set up accounts:

1. Right-click the Dynamics Business Central (NAV) adapter, and then click **Set Up Accounts**. A window is displayed that shows all companies in the database.
2. In the **Template** list, select the company account you want to use as template for the table and column structure. If you are only selecting one company, then the template company must match the company that is selected.
3. Select **Use** to specify whether to retrieve data from the company.
4. In the **Default Table Usage** list, specify the order in which tables are retrieved and read. You have the following options:

Option	Definition
Primary	Data from this company account is read and retrieved first
Secondary	Data from this company account is read and retrieved after the primary account if they have not already been retrieved from the primary account
None	Tables from this company are not retrieved, unless you specify at table level that you want to retrieve data from a specific table

Modifying Table Usage on Dynamics Business Central (NAV) Tables

When you set up accounts, you specify the default order in which data is retrieved from the individual accounts. However, it is possible to specify a different order of priority for individual tables.

1. Expand **Data Sources**, right-click the NAV adapter that contains the tables whose priority you want to change, and then select **Modify Table Usage**. The company accounts and the usage of all tables will be displayed.

2. Right-click the field that contains the setting for the table and the account for which you want to change priority of data retrieval. You have the following options:

Priority	Definition
Default	Data from this table is read and retrieved first
Primary	Data from this table is read and retrieved first
Secondary	Data from this table is read and retrieved after the primary table if they have not already been retrieved from the primary table
None	Data from this table is not retrieved
1-9	Specify the order of priority in the range from 1-9.
Enter priority	If the order of priority exceeds the numbers 1-9, you can specify additional numbers.

3. Click **OK**.

Modifying the Usage of a Single Dynamics Business Central (NAV) Table

You can change the order in which data is retrieved from individual tables.

1. Right-click the Dynamics Business Central (NAV) adapter that contains the table whose priority you want to change, and then select the preferred table.
2. Right-click the table and select **Modify Single Table Usage**. The company accounts and the usage specified in **Setup Company Accounts** will be displayed.
3. Right-click the field containing the settings for the table, and then specify the table usage. You have the following options:

Priority	Definition
Default	Data is retrieved based on the settings specified when you set up the company accounts.
Primary	Data from this table is read and retrieved first
Secondary	Data from this table is read and retrieved after the primary table if they have not already been retrieved from the primary table
None	Specify the order of priority in the range from 1-9
Enter priority	If the order of priority exceeds the numbers 1-9 you can specify additional numbers here

4. Click **OK**.

Changing Dynamics Business Central (NAV) Schemas

You can change the schema for the entire Dynamics Business Central (NAV) adapter or for individual tables that belong to the adapter.

1. Right-click the Dynamics Business Central (NAV) adapter and then click **Change Schema** .
- OR -
Expand the Dynamics Business Central (NAV) adapter, and then select the table whose schema you want to change.
2. In the **Select Schema** to change list, select the schema you want to change, and then select **Change Schema** .
3. In the **New Schema Name** field, enter a name for the schema, and then click **OK**.

Microsoft Excel Data Source

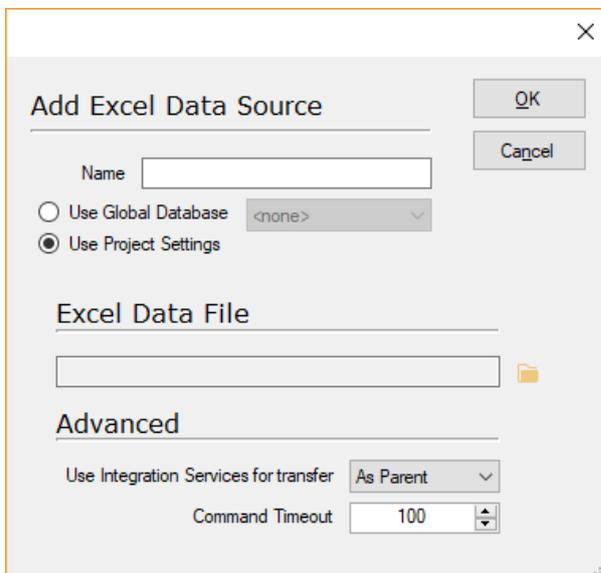
TimeXtender supports Microsoft Excel spreadsheets as a data source, both the older .xls format and the newer .xlsx format from Excel 2007 and beyond.

To use Excel files as a data source, you must ensure that the worksheet data is in list format. This means that the data must be set up in a database format consisting of one or more named columns. The first row in each column must have a label, and there can be no blank columns or rows within the list.

Adding an Excel Data Source

To add a new Excel data source, follow the steps below:

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add Excel Data Source**. The **Add Excel Data Source** window appears.



2. Click the folder icon under **Excel Data File** to display the **Find Files or Folders** window. Navigate to and select the Excel file you want to use as a data source, and then click **OK**.

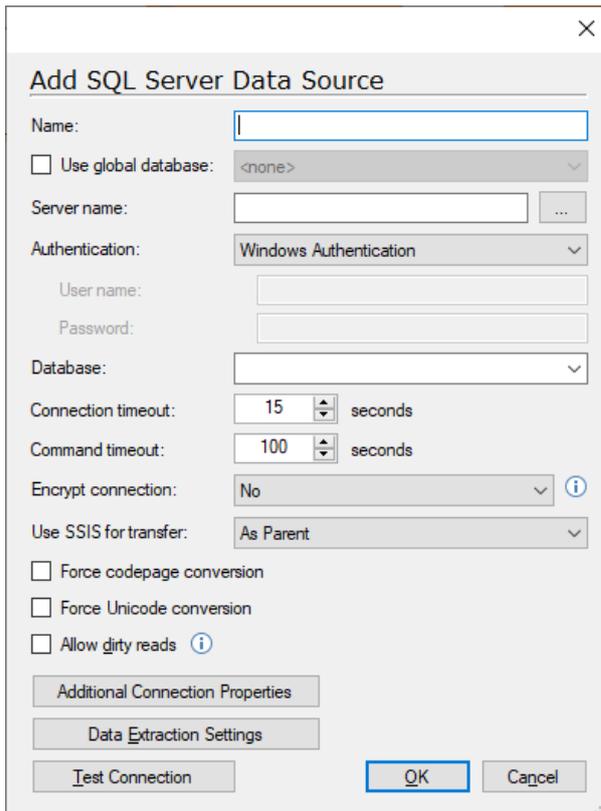
Microsoft SQL Server Data Source

TimeXtender supports all versions of Microsoft SQL Server as well as Azure SQL Database as a data source.

Adding a SQL Server Data Source

To add a new SQL Server data source, follow the steps below:

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add SQL Server Data Source**.



2. In the **Name** box, type a name for the data source. The name cannot exceed 15 characters in length.
3. In the **Server name** box, enter the location of the database server. Click the ellipsis (...) to choose one of the available servers in your local Active Directory, if any.
4. In the **Authentication** list, click the mode of authentication you want to use. You have the following options:
 - **Windows Authentication:** Use the logged-in Windows user's credentials for authentication.
 - **SQL Server Authentication:** Use a login set up on the SQL Server. Enter the username and password in the corresponding fields.
 - **Azure AD Password Authentication:** Use Azure AD credentials from a domain that is not federated with Azure AD. Enter the username and password in the corresponding fields.

- **Azure AD Integrated Authentication:** Use the logged-in Windows user's credentials for authentication, provided that he is logged in using Azure AD credentials from a domain that is federated with Azure AD.
5. In the **Database** box, enter the name of the database, or select it from the drop-down list.
 6. In the **Connection timeout** box, specify the number of seconds to wait before terminating the attempt to connect to the server.
 7. In the **Command timeout** box, specify the number of seconds to wait before terminating the attempt to connect to the database.
 8. (Optional) In the **Encrypt connection** list, you can enable encryption of the connection, which is recommended when you are not in a private network (e.g. when your server is on Azure). You have the following options:
 - **No:** The communication is not encrypted (default).
 - **Yes:** The communication is encrypted. The server's certificate is verified by a certificate authority.
 - **Yes, trust server certificate:** The communication is encrypted. but the server's certificate is not verified. This setting is not recommended for use on public networks.
 9. In the **Use SSIS for transfer** list click on **Yes** to enable SQL Server Integration Services data transfer, **No** to disable it or leave it at **As Parent** to respect the project setting.
 10. Select **Force codepage conversion** to convert all fields to the collation of the data warehouse.
 11. Select **Force Unicode conversion** to declare all alphanumeric fields as **nvarchar**.
 12. Select **Allow dirty reads** to allow reading from the source without locking the table.
 13. If you want to add additional connection settings, click **Additional Connection Properties** . In the **Connection String Properties** window, enter the required connection strings, and then click **OK**.
 14. (Optional) Click **Data Extraction Settings** if you want to limit the objects brought into TimeXtender before the data selection stage. For more information, see [Filtering What Objects to Extract](#).
 15. Click **Test Connection** to verify that the connection settings you have specified are working and then click **OK** to add the data source.

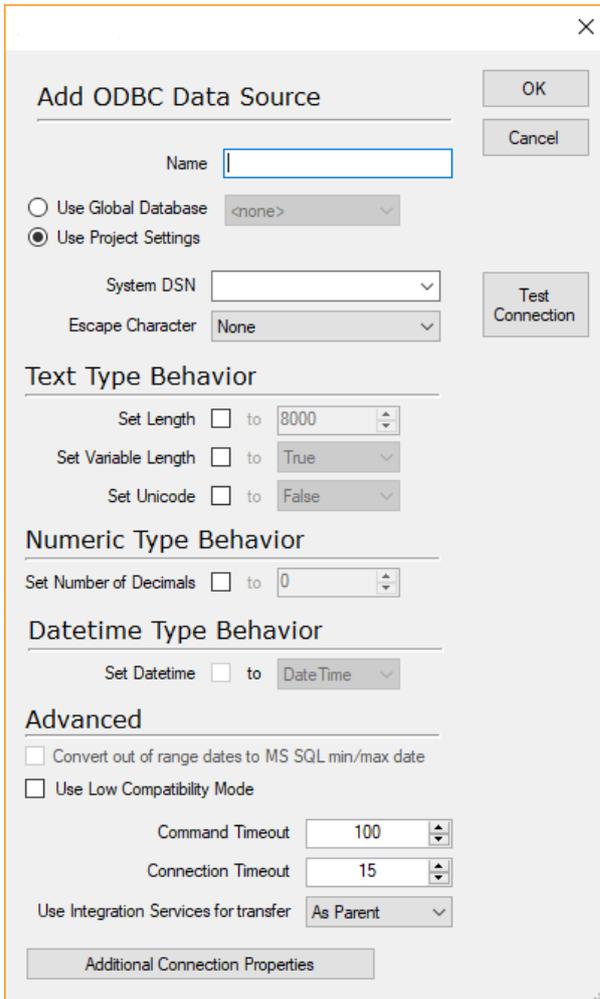
ODBC Data Source

TimeXtender supports ODBC for retrieving data from a wide range of systems. Both application specific and a generic option is available.

Adding an ODBC Data Source

To add a new ODBC data source, follow the steps below:

1. Open a business unit, right click **Data Sources**, click **Data Sources** and click **Add Generic ODBC Data Source** or **Application specific ODBC** followed by a click on the system you want to connect to.



2. In the **Name** field, type the name of the data source.
3. In the **System DSN** list, select the Data Source Name.
4. In the **Escape Character** list, select the escape character specific to your ODBC driver.
5. The **Text Type Behavior** fields are used to control how the ODBC driver handles text. These fields are optional. You have the following options:

Option	Definition
Set Length	Specifies an exact text string length

Set Variable Length	True, if you want a variable text string length
Set Unicode	True, if you want to use Unicode

6. In **Set Number of Decimals**, specify a fixed number of decimals. This field is optional.

Note: Convert out of range dates to MS SQL min/ max is not available for Navision native databases.

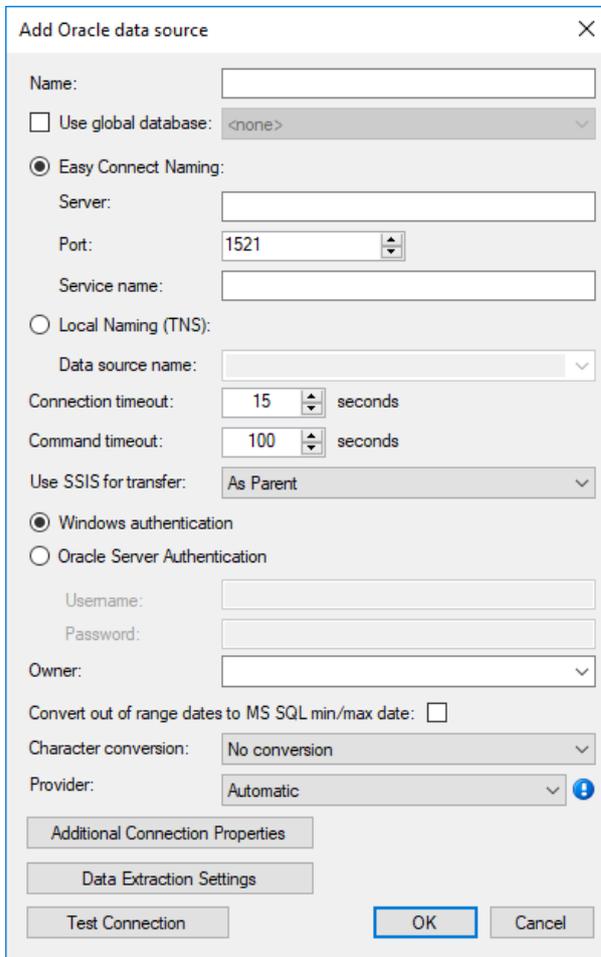
7. Select **Use low compatibility mode** if you have trouble retrieving data from the database.
8. In the **Command Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the database.
9. In the **Connection Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the server.
10. If you want to add additional connection strings, click the **Additional Connection Properties** button. In the **Connection String Properties** window, type the preferred connection strings, and then click **OK**.

Oracle Database Data Source

TimeXtender can extract data from Oracle databases.

Adding a Oracle Database Data Source

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add Oracle Data Source**. The **Add Oracle Data Source** window opens.



2. Type a **Name** used to identify the data source in TimeXtender.
3. If you want to use Easy Connect, also known as "EZCONNECT", leave **Easy Connect Naming** selected and enter the **Server**, **Port** and **Service name** of the data source.
4. If you want to use local naming, i.e. TNS, click **Local Naming (TNS)** and select the data source in the **Data Source Name** list, which is sourced from the local "tnsnames.ora" file.
5. (Optional) Enter the number of seconds to wait before terminating the attempt to connect to the database in **Command Timeout**.
6. (Optional) Enter the number of seconds to wait before terminating the attempt to connect to the server in **Connection Timeout**.
7. (Optional) In the **Use SSIS for transfer** list, you can change the default, **As Parent**, by clicking either **Yes** or **No**.
8. Type **TNS alias**, type the alias that identifies the database.

9. In the **Owner** list, click the owner of the database.
10. Under **Login**, click **Oracle Server authentication** if you want to use this login method and enter **User name** and **password**.
11. (Optional) Select **Convert out of range dates to MS SQL min/max** if you want to convert all dates older than January 01, 1753 to 01-01-1753.
12. (Optional) In the **Character conversion** list, select **Unicode** or **Non-Unicode** to convert all text to the selected encoding.
13. (Optional) In the **Provider** list, select the provider you want to use. If you leave the setting at **Automatic**, TimeXtender uses the best available provider.
14. (Optional) Add additional connection strings, click **Additional Connection Properties**. In the **Connection String Properties** window, type the connection strings and click **OK**.
15. (Optional) Click **Data Extraction Settings** if you want to limit the objects brought into TimeXtender before the data selection stage. For more information, see [Filtering What Objects to Extract](#).
16. Click **OK** to add the data source.

Oracle MySQL Data Source

TimeXtender supports MySQL data sources through ODBC.

Adding a MySQL Data Source

To add a MySQL data source, follow the steps below:

1. Open a business unit, right click **Data Sources**, click **Data Sources**, click **Application specific ODBC**, and then select the preferred MySQL native database.
2. In the **Name** field, type the name of the data source.
3. In the **System DSN** list, select the Data Source Name.
4. In the **Escape Character** list, select the escape character specific to your ODBC driver.
5. The **Text Type Behavior** fields are used to control how the ODBC driver handles text. These fields are optional. You have the following options:

Option	Definition
Set Length	Specifies an exact text string length
Set Variable Length	True, if you want a variable text string length
Set Unicode	True, if you want to use Unicode

6. In **Set Number of Decimals**, specify a fixed number of decimals. This field is optional.
7. Select **Convert out of range dates to MS SQL min/max** if you want to convert all dates older than January 01, 1753 to 01-01-1753.
8. Select **Use Low Compatibility Mode** if you have trouble retrieving data from the database.
9. In the **Command Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the database.
10. In the **Connection Timeout** field, specify the number of seconds to wait before terminating the attempt to connect to the server.
11. If you want to add additional connection strings, click the **Additional Connection Properties** button. In the **Connection String Properties** window, type the preferred connection strings, and then click **OK**.

Salesforce Adapter

The Salesforce application adapter enables you to extract data stored in your Salesforce Sales Cloud CRM. You will need an edition of Salesforce Sales Cloud that enables the use of the Salesforce AP. At the time of writing, this means at least the Enterprise edition.

The adapter uses the Salesforce REST API to extract data. The data types of the extracted data is converted from Salesforce data types to their SQL Server equivalents.

Adding a Salesforce Adapter

To connect to a Salesforce data source using the application adapter, follow the steps below.

1. Open a business unit, right click **Data Sources**, click **Adapter Data Sources** and then click **Add Salesforce Adapter**. The **Add Salesforce Adapter** window opens.

The screenshot shows the 'Add Salesforce Adapter' dialog box. It includes a title bar with a close button (X). The main area contains several fields and options: 'Name' (text box), 'Use Global Database:' (radio button, dropdown menu showing '<none>'), 'Use Project Settings:' (radio button, selected), 'Username:' (text box), 'Password:' (text box), 'Token:' (text box), 'API version:' (dropdown menu with an info icon), 'Custom URL:' (text box), 'Resolved URL:' (text box containing 'https://login.salesforce.com/services/Soap/c/'), 'Connection timeout:' (spin box with '100'), 'Batch size:' (spin box with '2000'), 'Table request limit:' (spin box with '1' and an info icon), 'Use label as name' (checkbox, unchecked), 'Unicode' (checkbox, checked), 'Proxy' section with 'Don't use proxy' (radio button, selected), 'Use application proxy settings' (radio button, unselected), 'Use these settings:' (radio button, unselected), 'Server:' (text box), 'Port:' (spin box with '8080'), 'Username:' (text box), 'Password:' (text box). At the bottom are 'Test Connection', 'OK', and 'Cancel' buttons.

2. Type the name you want to use for the adapter in the **Name** box.
3. Enter your **Username**, **Password** and **Token**. The token is provided by Salesforce and will change if the password is changed. Salesforce can be configured not to use security tokens. In that case, simply leave **Token** empty.

4. In the **API Version** list, click the Salesforce API version you want to use. Usually, you should use the newest version. However, if Salesforce makes changes in the API that breaks the adapter, you have the option of using an earlier version.
5. (Optional) Enter a **Custom URL** if you want to connect to your Salesforce sandbox instead of the default URL. The **Resolved URL** box shows you the entire URL the adapter will use to connect.
6. In the **Connection Timeout** box, type or select the number of seconds to wait for a connection to be established.
7. In the **Batch size** box, type or select the number of records to fetch in each request. Bigger batches uses more memory.
8. In the **Table request limit** box, type or select the maximum number of HTTP requests you want to allow per table.
9. Select **Use label as name** to use labels as names instead of the physical system names.
10. Select **Unicode** to make string based fields Unicode ready.
11. If you need to connect through a proxy server, you have the following options:
 1. **Don't use proxy**
 2. **Use application proxy settings:** Use the proxy settings configured on the application level. To adjust the application settings, click **Options** on the **Tools** menu and then click **Proxy Settings**. Note that using a proxy server for Internet connections can also be turned on and off from here.
 3. **Use these settings:** Enter the settings - server, port, username and password - to use for the adapter.
12. (Optional) Click **Test Connection** to verify that your settings are correct.
13. Click **OK** to add the adapter data source.

SAP Table Adapter

The SAP Table Adapter allows you to connect to SAP systems. The Adapter utilizes the **XTract IS** component from Theobald Software. Make sure that the latest version of the component is downloaded and installed on the server where TimeXtender is installed. The component can be downloaded from Theobald Software: <http://theobald-software.com/en/product-downloads.html>

Due to some issues in the standard SAP meta-data layer, you will need to install a function module in SAP. For more information, see this link: <https://help.theobald-software.com/Xtract-IS-EN/default.aspx?pageid=SAPCustomizing-EN:table-restrictions>

The function module will make it possible to:

- Extract tables/ table columns with an overall width greater than 512.
- Extract tables that contain at least one column of type F (floating point).
- Extract table TCURR which has some meta data problems in the Data Dictionary.

Adding a SAP Application Adapter

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add SAP Table Adapter**. The **Add SAP Table Adapter** window opens.

Add Sap Table Adapter [Close]

SAP Name:

String Translation:

Max Rows:

Package Size:

Table Name Filter: [Browse]

Use Custom Function

Custom Function:

Use Data Compression

Activate Background Extraction

Buffer Location: [Browse]

Use Table and Field Translations

Translation Id:

Use Global Database

Use Project Settings

Client:

User Name:

Password:

Language:

Host:

System Number:

Message Server:

Group:

SID:

Log Directory:

Encrypt Password

Use Single Server

[OK] [Cancel]

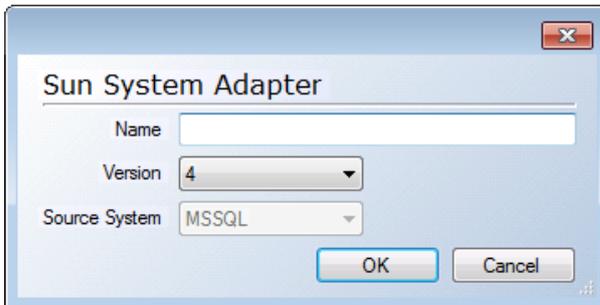
2. Fill in the settings that matches your setup and click **OK**.

Sun System adapter

The Sun System adapter simplifies the extraction of data from a Sun System using a Microsoft SQL Server.

Adding a Sun System adapter

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add Sun System Adapter**. The **Add Sun System Adapter** window opens.



2. Type a **Name** for the adapter.
3. In the **Version** list, click the version of Sun System you are connecting to.
4. Click **OK**.
5. The standard window for adding a Microsoft SQL Server data source opens. Enter the connection details for your data source - see [Adding a SQL Server Data Source](#) for more information.

UNIT4 Business World (Agresso) Adapter

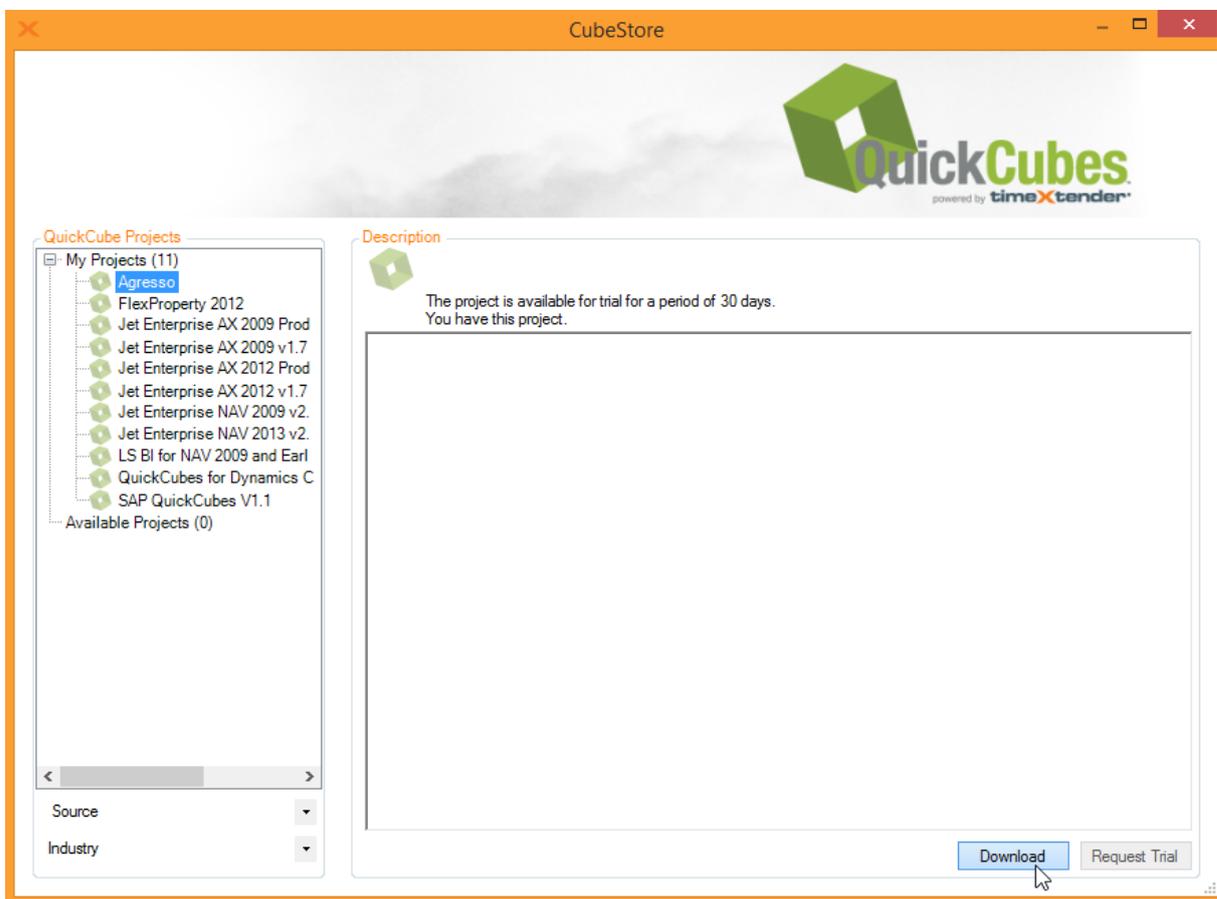
This adapter enables the extraction of data from the UNIT4 Business World ERP system, previously known as Agresso.

Adding an Agresso Adapter data source

Adding an Agresso Adapter consists of two major parts. Unlike other application adapters, the Agresso Adapter comes with a TimeXtender project based on Agresso's standard dimensional setup. Downloading and configuring this project is the first part, while the second part is configuring Agresso-specific settings for client, dimensions, translations and flexi-tables.

To download and configure the Agresso project, follow the steps below.

1. In the **File** menu, TimeXtender click **CubeStore**. The CubeStore window opens.

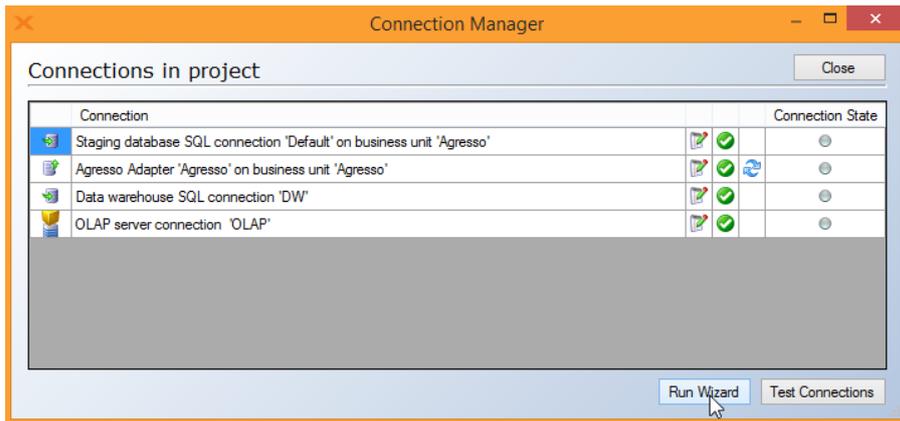


2. In the **QuickCube Projects** list, click **Agresso** and then click **Download**.

Note: The project is only available if you have a license for TimeXtender that includes the Agresso Adapter.

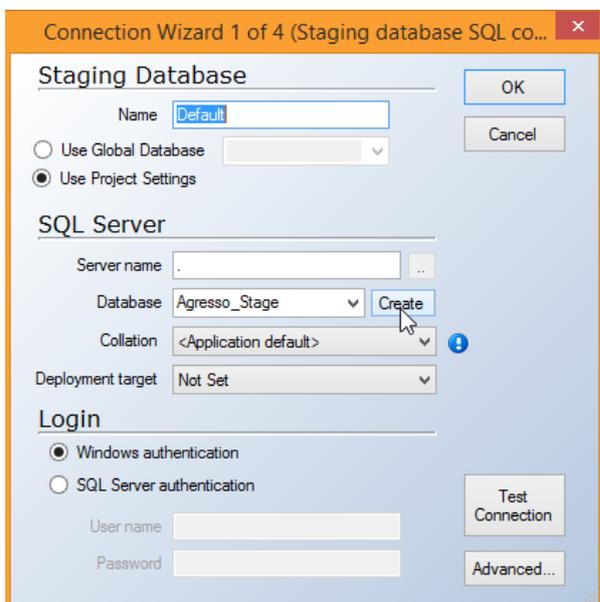
3. Once the project has been downloaded, TimeXtender will ask you if you want to run the Connection Manager. Click **Yes**.

4. The Connection Manager window opens.



Click **Run Wizard**. Four windows will now open in tour, allowing you to quickly configure the different connections.

5. Create a staging database for use with the project and click **OK**. For more information, see [Adding a Business Unit](#).



6. Enter the connection information for your Agresso database and click **OK**. For more information, see [Adding a SQL Server Data Source](#).

Connection Wizard 2 of 4 (Agresso Adapter 'Agresso...')

Agresso Adapter OK
Cancel

Name

Use Global Database <none>
 Use Project Settings

SQL Server

Server name ...

Database

Login

Windows authentication
 SQL Server authentication

User name
Password

Test Connection

Advanced

Command Timeout
Connection Timeout
Use Integration Services for transfer
Force Codepage Conversion
Force Unicode Conversion
Allow Dirty Reads ⓘ

Additional Connection Properties

7. Create a data warehouse database and click **OK**. For more information, see [Adding a Data Warehouse](#).

Connection Wizard 3 of 4 (Data warehouse SQL conn...)

Data Warehouse OK
Cancel

Name

Use Global Database
 Use Project Settings

SQL Server

Server name ..

Database Create

Collation ⓘ

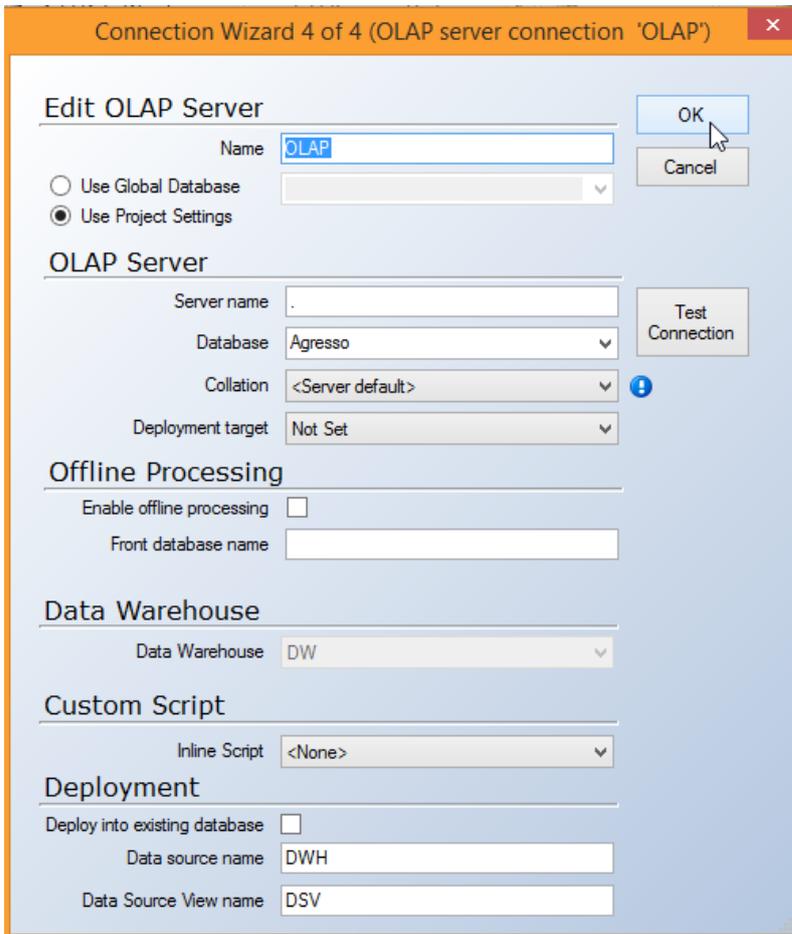
Deployment target

Windows authentication
 SQL Server authentication

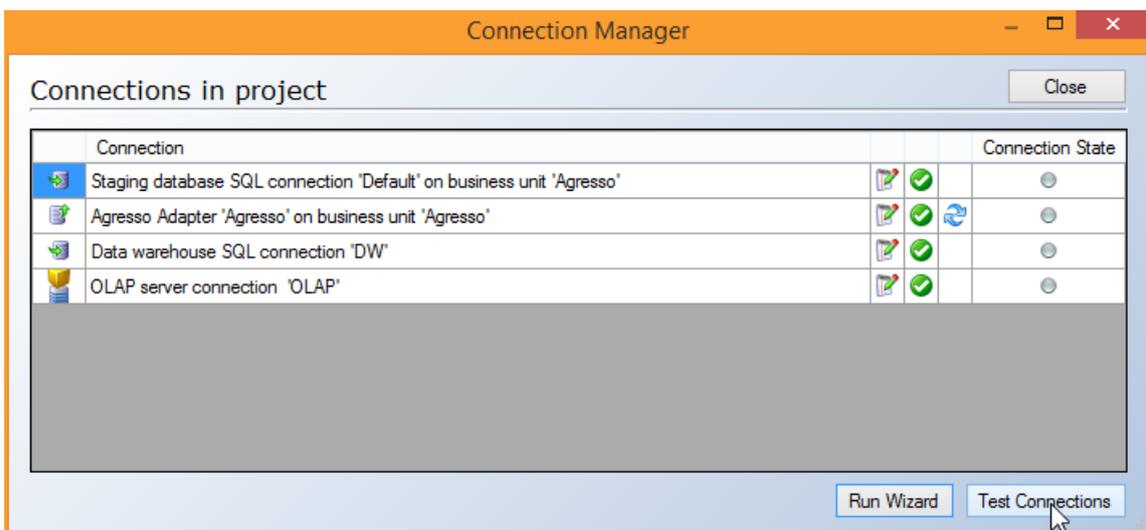
User name
Password

Test Connection
Advanced...

- Set up an SSAS Multidimensional server for the project. For more information, see [Adding an SSAS Multidimensional Server](#).



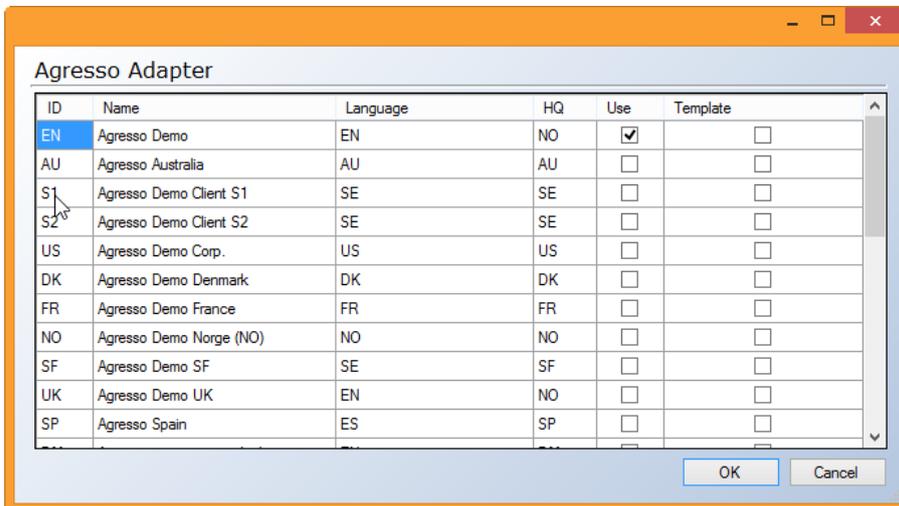
- Back in the **Connection Manager**, click **Test Connections** to ensure that all connections work and click **Close**.



As mentioned earlier, the second part of adding an Agresso Adapter involves setting up the clients, dimensions, flexi-tables and translations that are part of the Agresso system.

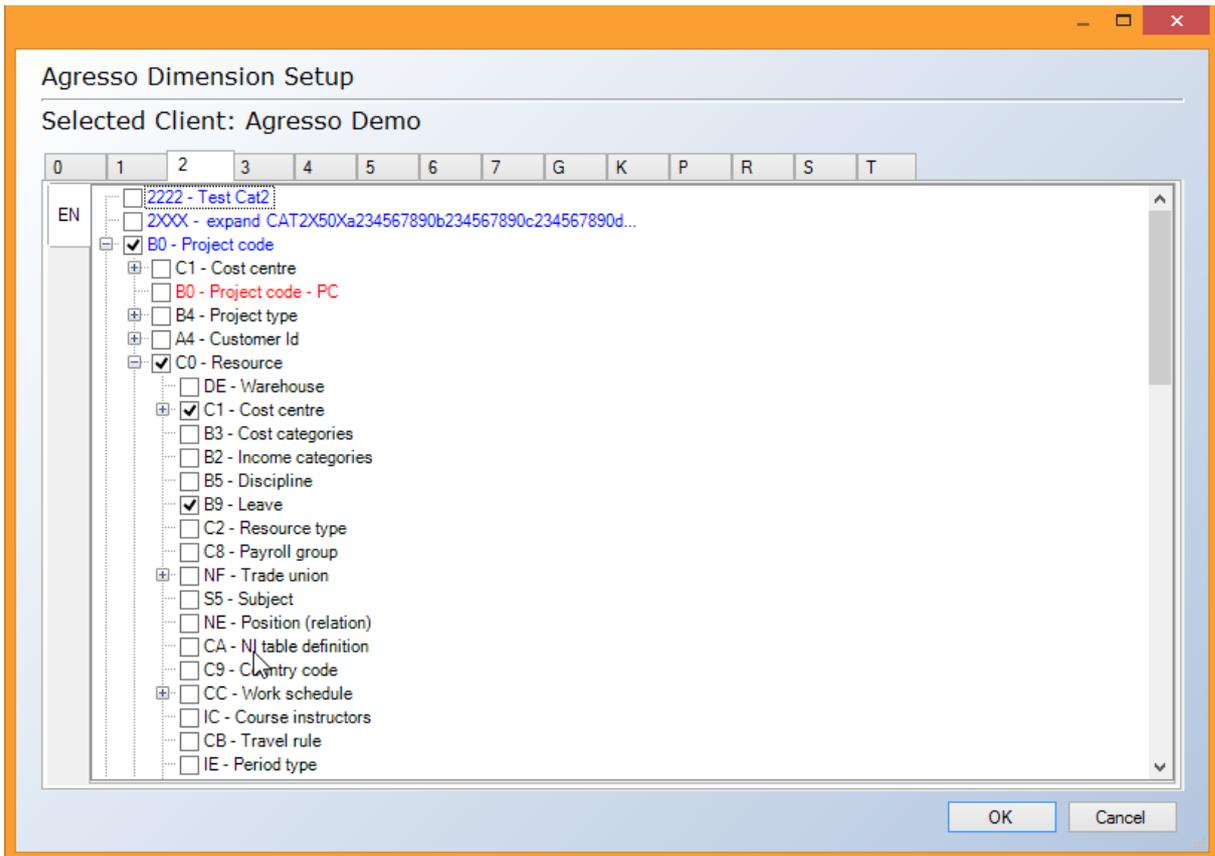
To configure the Agresso-specific settings, follow the steps below.

1. Right-click the Agresso Adapter you just added and click **Setup Clients**. A window containing a list of clients opens.

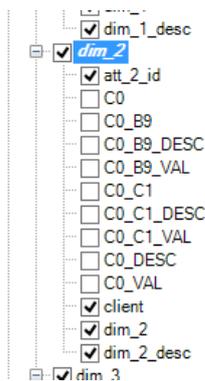


2. Choose the clients you want to extract data from by selecting the check boxes in the **Use** column. If you want a client to serve as a template for the dimensional setup (see below), select the check box in the **Template** column. Click **OK**.
3. Right-click the Agresso Adapter and click **Synchronize Data Source**. The synchronization window opens. Wait for the process to finish and click **Close**.
4. Before the next step, the project needs to be deployed and executed. On the **Tools** menu, click **Deploy and execute project**. The **Deploy and Execute** window opens. Click **Start**, wait for the process to finish, and click **Close**.

- Right-click the Agresso Adapter and click **Setup Dimensions**. The **Agresso Dimension Setup** window opens.

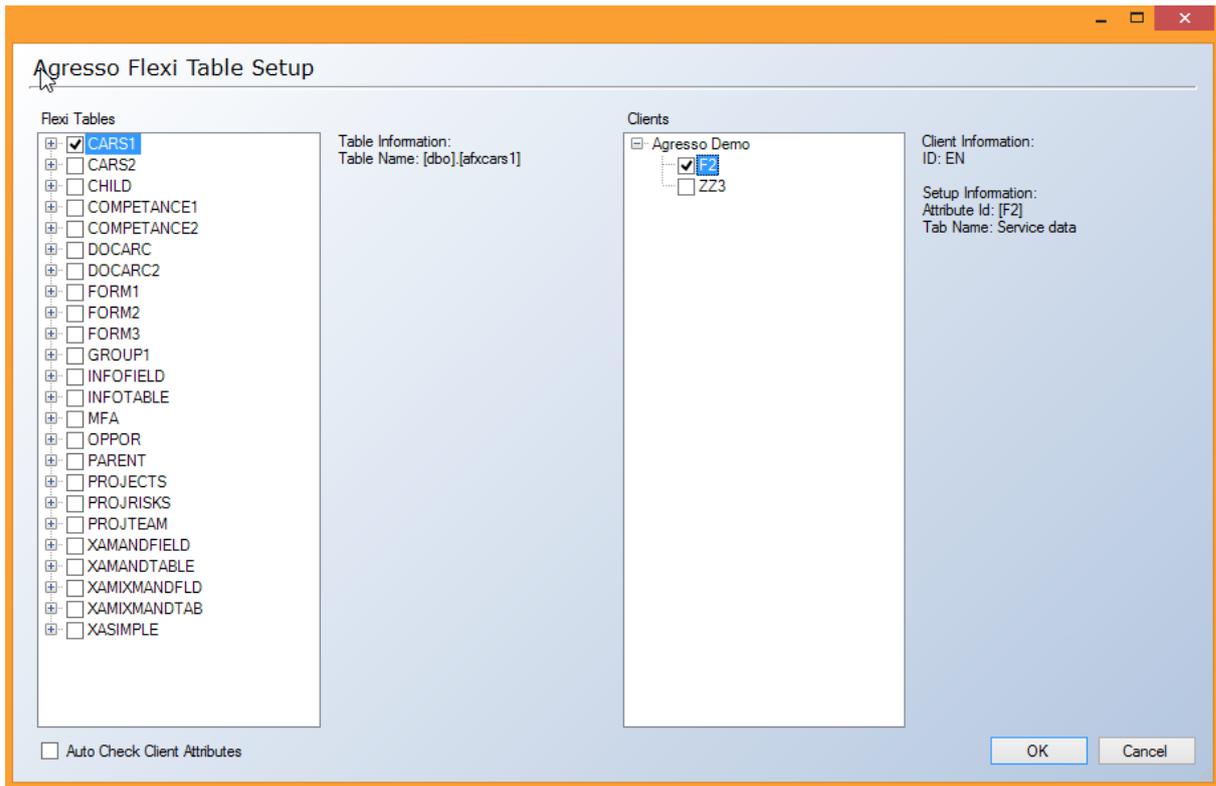


- Click **Load Dimension Values** (Default setting only shows previously selected dimension attributes)
- Select the dimension attributes you want to use in the project. If you did not select a template in the client setup, you need to do this for each client.
- Click **OK**. The selected attributes are automatically added to the dimension table and can then be used in the project.

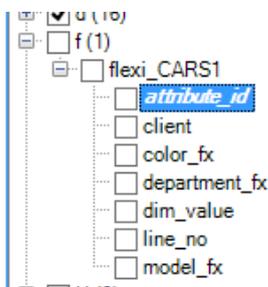


- Right-click the Agresso Adapter and click **Setup Flexi Tables**. The **Agresso Flexi Table Setup** window opens. The content of the **Flexi Tables** and **Clients** list depends

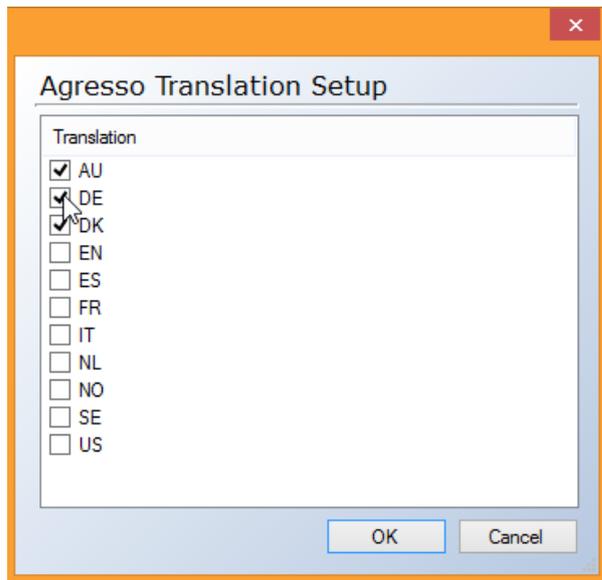
on the clients and dimensions you have chosen previously.



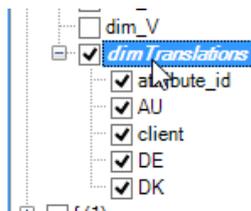
10. Select the combinations of flexi-tables and client attributes that you want to use in your project.
11. Click **OK**. The selected flexi-tables are automatically added to the source tables and must be integrated into the solution.



12. Right-click the Agresso Adapter and click **Setup Translations**. The **Agresso Translation Setup** window opens.



13. Select the translations you want to use in your project from the **Translation** list.
14. Click **OK**. The selected translations are automatically added to the source tables for use in the project.



Text File Data Source

One of the simplest ways to transfer data between systems is through the use of text files. For instance, some industrial equipment produce log files in text format that can be used as a data source in TimeXtender to include productivity data in the data warehouse.

One text file data source corresponds to one table in TimeXtender . You can load a single text file or multiple identically formatted text files.

Adding a Text File Data Source

While virtually identical, you have to choose between the **Single Text File** and **Multiple Text File** data sources when you set up a text file data source. The **Multiple Text File** data source is usually preferable even if you only have one text file since it enables you to add more text files as sources should you need it sometime in the future. To add a text file data source, follow the steps below:

1. Open a business unit, right click **Data Sources**, click **Data Sources** and then click **Add Multiple Text File data source/ Add Single Text File data source**. The **Add Multiple Text File** or **Add Single Text File** window appears.

The screenshot shows the 'Add Multiple Text File' dialog box. It includes fields for 'Name' and 'Table Name'. Under the 'Text file' section, 'Use Project Settings' is selected. The 'Setup' tab is active, showing options for 'Format' (Delimited), 'Header Row to skip' (0), 'Header Row delimiter', 'Row delimiter', 'Field delimiter', 'Field lengths', 'Field names in first data row', 'Text qualifier', 'File', 'Culture' (Danish (Denmark)), 'Unicode' (checkbox), 'Code page' (dropdown), 'Post processing' (Backup), 'Backup folder', and 'Use SSIS' (As Parent). There is also a 'Custom delimiters' section with 'Header row delimiter value', 'Row delimiter value', and 'Field delimiter value' inputs. 'OK' and 'Cancel' buttons are at the bottom right.

2. In the **Name** box, type a name for the data source.
3. In the **Table Name** box, type the name of the table that is created in the staging database. The table is prefixed with the data source name. If you want to set your own prefix, clear **Auto Prefix Tables** and type the prefix you want to use in **Manual Table Prefix** box.

4. (Optional) Click your preferred option in the **Transfer Failure Option** list to change the **Allow Failing Data Source** setting.
 5. In the **Format** list, click the format of the text file, e.g. how TimeXtender should make sense of the content of the file.
 - Select **Delimited** if rows and fields are separated by a character and click the relevant characters in the **Header Row delimiter**, **Row delimiter** and **Field delimiter** lists.
 - Select **FixedWidth** if the fields have a fixed length and type the lengths in **Field lengths** in a semicolon-separated format, e.g. "2;4;8;3".
 - Select **RaggedRight** if the last field is delimited by a character, while the previous fields are fixed width. Click the relevant characters in the **Header Row delimiter**, **Row delimiter** lists and type the lengths in **Field lengths** in a semicolon-separated format, e.g. "2;4;8;3".
 6. Select **Field names in first data row** if the first row of data contains field name, i.e. not data.
 7. Type a **Text Qualifier**, often a quotation mark, if you would like TimeXtender to strip from the fields before loading data into the staging data base.
 8. If you are adding a multiple text files data source, enter the path to the files you want to process separated by semicolon (;) in the **File** box. You can also use wildcards. Use "*" for any number of characters and "?" for a single character. You can also click the folder icon next to the **File** box to choose the file to process.
- OR -
- If you are adding a single text file data source, click the folder icon next to the **File** box to choose the file to process.
9. In the **Culture** list, click the language of the text file.
 10. In the **Code page** list, click the codepage of the text file.
 11. Select **Unicode** if TimeXtender should treat your file as Unicode.
 12. In the **Post processing** list, click the action you want TimeXtender to perform when the file has been processed.
 - Select **Backup** to move the file to a backup folder and click the folder icon next to the **Backup folder** field to select the folder.
 - Select **Delete** to delete the file.
 - Select **None** to leave the file as it is.
 13. In the **Use Integration Services for transfer** list, you can click Yes or No to change the setting from the default As Parent.
 14. Click the **Columns** tab and click **Get Fields** to load the fields, which will then be displayed in a list in the left-hand side of the window. You can select one or more fields in the list and adjust different settings for them:
 - **Column name**
 - **Data type**
 - **Text length**: Enter the maximum number of characters in the field.
 - **Variable length**: Select if you do not want the field to have a fixed length.
 - **Unicode**: Select to convert data to Unicode

- **Number of decimals:** Enter the maximum number of decimals allowed in the field.
15. (Optional) Select **Retain null values** to set the value of empty fields to the field's data type's default value instead of null.
 16. (Optional) Select **Enable file column** to add a column with the name of the source file to data.
 17. Click **Update** to show a preview of the data as TimeXtender understands it with the settings you have chosen. You have the option of adjusting the **Number of rows** to see more or less rows.
 18. Click **OK** to add the data source.

Designing the Data Warehouse

Once you have set up your data warehouses and business units and established connections to the sources you want to use, it is time to start designing your data warehouse. This involves transferring data from the source systems to the data warehouse(s) via a staging database or ODX and applying transformations and data cleansing rules along the way.

Selecting, Copying and Relating Tables

At its core, a TimeXtender project is about copying data from one place to another. Often, you simply copy data from a few data sources to a data warehouse via a staging database in a business unit. You can also consolidate data from multiple business units in one data warehouse or have multiple data warehouses and copy data between those.

Once the data has been copied, you can set up relations between the tables. Among other things, specifying the relations allows TimeXtender to do a referential integrity check during execution of the project.

Copying Tables From a Data Source to a Staging Database

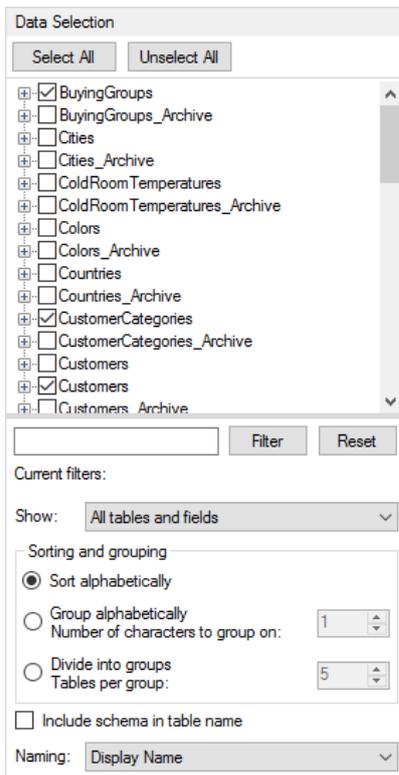
Note: If you use the [ODX Server](#), data will be copied directly from the ODX Server to a data warehouse. Consequently, you won't be using business units, including the staging database and the data sources that make up a business unit, and can safely ignore anything about staging databases in this chapter.

To get data into your staging database, you first have to connect at least one data source or adapter data source. See [Connecting to Data Sources](#) for more information.

Selecting Tables from a data source

To add tables to the staging database, you select the tables and fields that you want to extract from the data source.

1. Right click the data source and click **Read objects from data source**. This will get content for the list of tables in the **Data Selection** pane in the right-hand side of the window.



2. In the bottom of the pane, you will find filter and sorting options:

- To filter the list by specific terms, type a word in the box and click **Filter**. You can add more than one filter term. The terms you have entered are added to the **Current filters** list. Click **Reset** to remove the current filter.

Note: You can filter on both table and column names. With large data sources, loading the meta data to enable filtering on columns names can be time consuming and memory intensive. If you experience issues with this, you can disable column caching. For more information, see [Column Cache](#).

- In the **Show** list, you have the following filter options:
 - **All tables and fields:** No filter is applied.
 - **Selected tables and fields:** Only the tables and fields already selected from the data source are displayed. This is useful if you need to find an unselected a table or field.
 - **Unselected tables and fields:** Only the tables and fields that are not already selected from the data source are displayed.
 - **Only objects in project perspective:** Only the tables included in the currently active project perspective are displayed.
- Under **Sorting and grouping**, you have the following options:
 - **Sort alphabetically:** Sort the tables in alphabetical order.
 - **Group alphabetically:** Displays the tables in groups based on the number of characters entered in **Number of characters to group on**.

- **Divide into groups:** Divides the tables into groups with the number of tables by group based on the number entered in **Tables per group**.
- Select **Include schema in table name** to display schema name along with the table names.
- In the **Naming** list, you can choose what name you want to show for the tables. You have the following options:
 - **Display name**
 - **Database name**

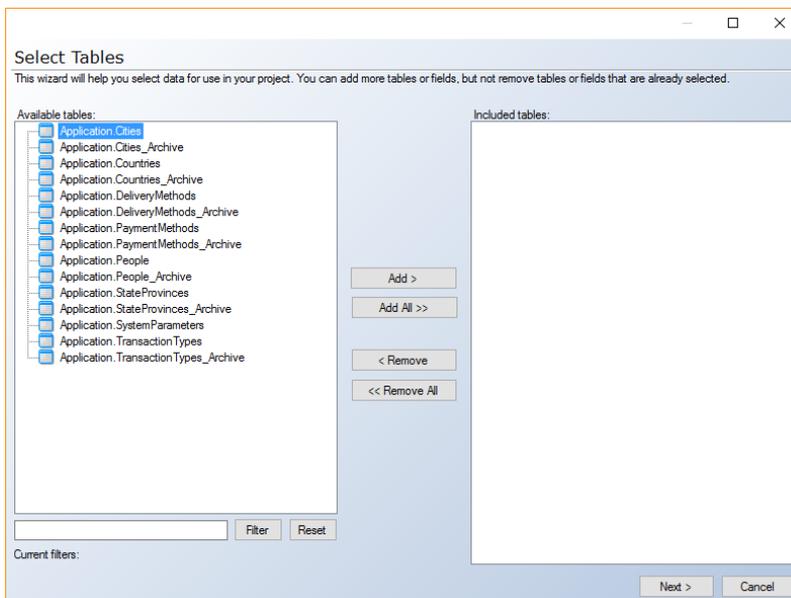
This only has effect on data sources where the two names differ.

3. Select the individual tables and fields that you want to copy from the data source into the staging database or click **Select all** to select all the tables in the source. For each table you select, a table is added to Tables under the staging database.
4. Right click on the staging database and click **Deploy and Execute** to create the table structure in the staging database and copy the data over.

Selecting More Tables from a data source

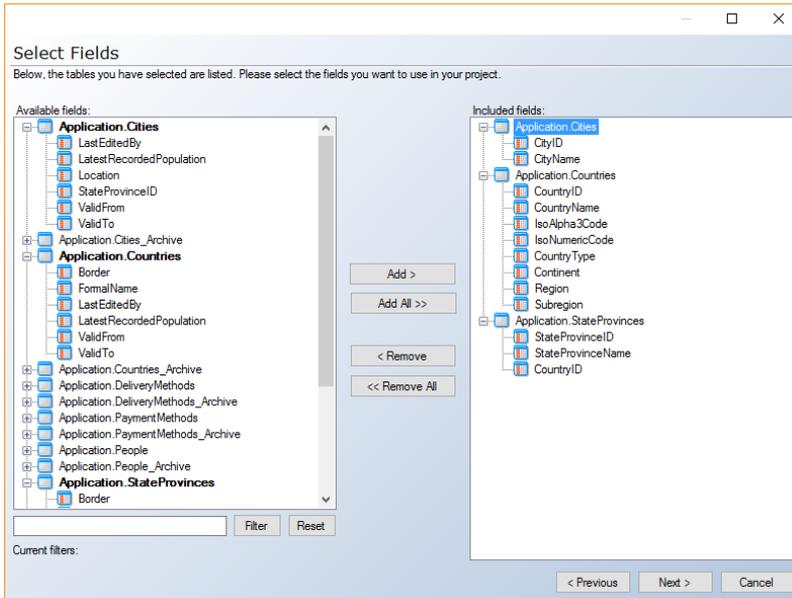
The Select Tables and Fields wizard gives you an alternative way to select more data from a data source. It is especially useful when you have a lot of tables and fields selected on a data source and need to add even more. Since the wizard only allows you to add tables and fields, you can't unselect a table or a field by accident. The select tables and fields from a data source with the wizard, follow the steps below.

1. Right click the data source and click **Read Objects from Data Source**.
2. Right click the data source, click **Automate** and click **Select Tables and Fields**. The wizard appears.



In the **Available tables** list, double-click the tables you want to select. You can also click a table and then click **Add** to add an individual table. Use the filter below the list to filter the list on table name. To add all visible tables, click **Add all**.

3. Click **Next**.

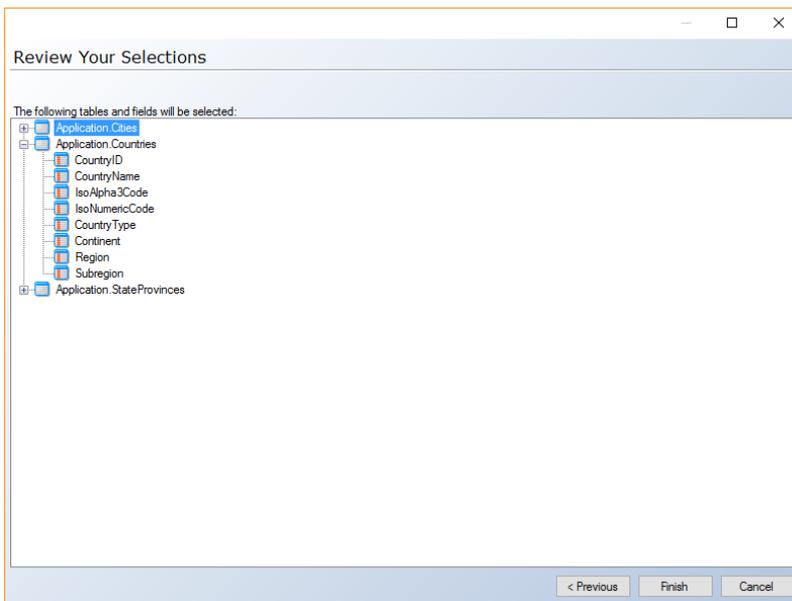


In the **Available fields** list, double-click the fields you want to select from the data source. You can also click a field and then click **Add** to add an individual field.

The tables in the Available fields list that have at least one field selected will be shown in bold.

In the **Included fields** list, any fields that are already selected on the tables you selected in the previous step are listed in gray. You cannot unselected fields on this page, only add new ones.

4. Click **Next**.



Your selections are listed for you to review. Click **Previous** to go back to an earlier

page and adjust the selections there or click **Finish** to apply the changes if they are correct.

5. Right click on the staging database and click **Deploy and Execute** to create the table structure in the staging database and copy the data over.

Copying Tables from an ODX to a Data Warehouse

To use data from an ODX in your TimeXtender project, you first need to [connect to an ODX server](#). You can then drag tables from the ODX to a data warehouse as explained below. Naturally, this requires a running and correctly configured ODX server connected to data sources and data storages. For more information on setting up an ODX server, see [ODX](#).

Synchronize Objects with an ODX Server

Data sources change and these changes will obviously propagate to the ODX data storage. For that reason, you should occasionally synchronize the ODX with your project when you have selected data. When TimeXtender synchronizes objects with an ODX server, the list of tables on the ODX server known by the project are compared to the current list of tables. If there are differences, TimeXtender will show you a list.

To synchronize objects

- Right click the ODX server and click **Synchronize Objects**.

Selecting Tables from the ODX Server

Selecting tables from the ODX server and moving them to a data warehouse is one operation when you use the ODX. Since the ODX is designed for large amounts of tables and views, you search for the tables you want to copy as opposed to selecting them from a list of all available tables.

To copy tables from an ODX server to a data warehouse, follow the steps below.

1. In the Solution Explorer, double-click the ODX server. The ODX opens in a tab.
2. In the tab you can search for the tables and views you want to copy. In the **Schema** and **Tables** boxes, type the string you want to compare with the names of tables and views in the ODX.
3. In the **Data source** list, select the sources you want to search.
4. Click **Search** to search the ODX data storage and add the matching tables to the **Results** list below.
5. Drag tables from the results list to the **Tables** node on a data warehouse to add them to the data warehouse. Hold Shift or CTRL to select multiple tables to drag.

When you drag and drop a single table, each operation has a default action and some secondary actions. The default action will be used if you drag and drop using the primary mouse button, while the secondary actions will be available in a shortcut

menu if you use the secondary mouse button to drag and drop. The available actions are as follows:

- **Add New Table (default):** Adds the table as a new table in the data warehouse. Default when no table in the data warehouse has the same name as the table being dragged.
- **Synchronize on Table Name (default):** Synchronizes the table with an existing table in the data warehouse with the same name. New fields from the new table are added and already existing fields are mapped to the fields from the new table. Default when a table in the data warehouse has the same name as the table being dragged.
- **Add New Table with Field Selection:** Adds the table with the fields you select in a window that opens.
- **Synchronize with Other Table:** Synchronizes the table with an existing table in the data warehouse chosen by you.

When you drag and drop multiple tables, TimeXtender will first try to synchronize each table on table name. If no table with the same name exists in the data warehouse, the table is added as a new table.

6. Repeat step 3-7 until you have copied all the tables and views you want to copy.

Copying Tables between Databases

Once the data has been selected from the source, moving tables between databases works in the same way no matter if it is between a staging database and a data warehouse or between two data warehouses. To TimeXtender, they are all just databases.

In this context, the difference between a staging database and a data warehouse is that one table in a data warehouse can get data from multiple source tables or views. This makes it possible to consolidate multiple tables or views from the staging database in the data warehouse. If you copy a table from one data warehouse to another, the copy will have the same mappings as the original.

You can find the mappings between source and destination tables on the fields on the destination table. If you expand a field, the mappings are displayed as "Copy From [schema].[table].[field]".

Copying a table From One Database to Another

To copy a single table from a staging database to a data warehouse or between data warehouses

- Drag the table from **Tables** under the source database to **Tables** under the destination data warehouse.

When you drag and drop a single table, there is a default action and some secondary

actions. The default action will be used if you drag and drop using the primary mouse button, while the secondary actions will be available in a shortcut menu if you use the secondary mouse button to drag and drop. The available actions are as follows:

- **Add New Table (default):** Adds the table as a new table in the data warehouse. Default when no table in the data warehouse has the same name as the table being dragged.
- **Synchronize on Table Name (default):** Synchronizes the table with an existing table in the data warehouse with the same name. New fields from the new table are added and already existing fields are mapped to the fields from the new table. Default when a table in the data warehouse has the same name as the table being dragged.
- **Add New Table with Field Selection:** Adds the table with the fields you select in a window that opens.
- **Synchronize with Other Table:** Synchronizes the table with an existing table in the data warehouse chosen by you.

Copying All Tables From One Database to Another

If you need to copy all tables from one database to another, there is an easier way than to copy all tables individually.

To copy all tables from a staging database to a data warehouse or between data warehouses

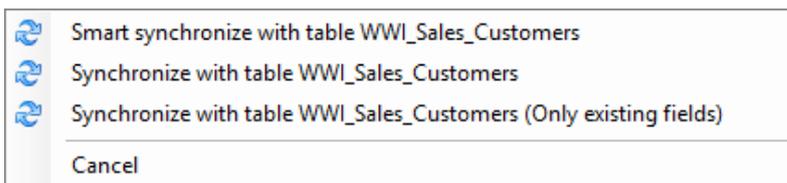
- Drag **Tables** from the staging database to **Tables** under the data warehouse.

TimeXtender will first try to synchronize each table on table name. If no table with the same name exists in the data warehouse, the table is added as a new table.

Mapping a Table to a Table on the Data Warehouse

To map a table from one database to an existing table in another database, follow the steps below:

- Drag a table from the source database to a table in the data warehouse. A menu appears.



You have the following options:

- **Smart synchronize with table [table name]:** TimeXtender compares the table you have dragged in with the other source tables on the table and add the fields

from the source table that matches fields from the other source tables.

- **Synchronize with table [table name]:** Add the fields of the source table to the destination table. When names are identical, a mapping is added, otherwise a new field is created on the table.
- **Synchronize with table [table name] (only existing fields):** Map the fields of the source table to existing fields on the destination table with the same name.

Copying a Field to a Table as a New Field

In addition to moving entire tables, you can also copy single fields.

To copy a field from a table in a staging database to a table on a data warehouse or between tables on two data warehouses

- Drag the field from the table in the source database to a table in the data warehouse.

Mapping a Field to a Field

To map a field from a table in a staging database to a table on a data warehouse or from a table in one data warehouse to a table in another data warehouse

- Drag the field from a table in the source database to a field on a table in the data warehouse.

When you map a field, the data type is automatically converted to match the data type of the data warehouse field if necessary.

Table Relations

For TimeXtender to know how tables are related, you have to specify relations. Among other things, the relations between tables you have defined are used for a referential integrity check on execution, for the default join when you create conditional lookup fields and for relating dimensions in an SSAS Multidimensional cube.

Under **Relations** under each table, you can see the relations that this particular table has to other tables.

Relations are grouped by the foreign table and the relation name defaults to “[foreign table name]_[foreign field name]”.

Each relation contains one or more field relations. On each field relation, the part on the left side of the equals sign is a field on the foreign table, while the part on the right side is a field on the table that has the relation.

Adding a New Relation

To add a new relation to a data warehouse or business unit table, follow the steps below.

1. Navigate to the table you want to relate to another table.
2. Click on the field you want to base the relation on and then drag and drop the field on a field on another table. Note that the fields must be of the same data type to create a relation.
3. TimeXtender will ask you if you want to create a relation. Click on **Yes**. The relation is created on the table on which you drop the field.

Setting a Default Relation

You can set one of the relations between two tables to be the default relation. A default relation is useful if you have more than one relation between two tables, for instance as join for lookup fields that do not have a specific join set and for auto-relation when you add a dimension to an SSAS Multidimensional cube.

- To set a relation as the default relation, right click the relation and click **Set as Default Relation**.

Setting the Severity of Violating the Referential Integrity Check

TimeXtender uses the relations you have defined to perform a referential integrity check, or foreign key constraint check, when the table is executed. TimeXtender checks the value of any field that are part of a relation to see if the value exists in the related field in the related table. If not, TimeXtender considers the record invalid.

For instance, a Sales Order table might contain a Customer ID field that is related to a Customer ID field in a Customer table. If a specific sales order record contains a Customer ID that is not in the Customer table, TimeXtender considers that record to be invalid.

You can define the severity of the violation on each relation.

To set the severity of a violation on a particular relation

- Right click the relation, click **Relation Type** and click your preferred type. You have the following options:
 - **Error**: TimeXtender moves the invalid record to the error table. This means data will be missing from the valid instance of the table.
 - **Error with physical relation**: The relation is stored in the database for other database tools to see. The behavior is otherwise the same as error. Note that the table needs to have a primary key and a unique index set. If index automation is disabled on the table, you will have to create the index yourself.
 - **Warning**: TimeXtender copies the invalid record to the warnings table and the valid instance of the table. You will not be missing data from the valid table. However, you might need to handle the violated rule in some way.
 - **Relation only**: TimeXtender ignores any violations of the check.

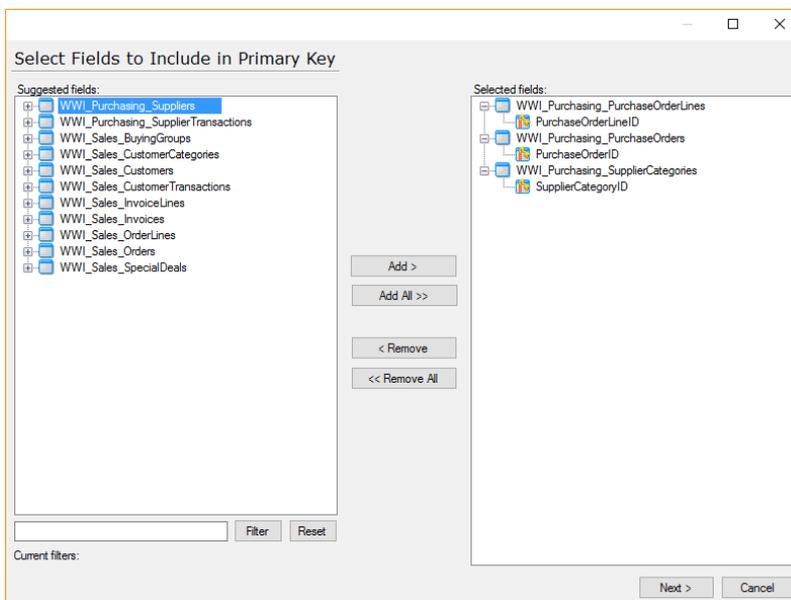
Constraint Suggestions

In addition to the foreign constraint check, TimeXtender can perform a primary key constraint check on execution. Both checks are based on the relations and primary keys you have added to the project. Sometimes information about the tables' primary keys and relations can be found in the data source and read by TimeXtender. The Constraint Suggestions wizard surfaces these suggestions and allows you to implement the ones you want to use.

Adding Relations and Primary Keys for Multiple Tables

To use the Constraint Suggestions wizard to add relations and primary keys to all tables in a staging database or data source, follow the steps below.

1. Right click the staging database or data source, click **Automate** and click **Add Suggested Constraints**. The wizard appears on the page where you can select what fields to include in the primary key on the respective tables.



In the **Suggested fields** list, double-click the fields you want to use as primary keys. You can also click a field and then click **Add** to add an individual field. To add all visible fields, click **Add all**. Use the filter below the list to filter the list on field name. The following wildcards are supported:

- **%**: Any string of zero or more characters.
 - **_**: Any single character.
 - **[]**: Any single character within the specified range ([a-f]) or set ([abcdef]).
 - **[^]**: Any single character not within the specified range ([^a-f]) or set ([^abcdef]).
2. Click **Next**. In the **Suggested fields** list, double-click the fields you want to create a relation from. You can also click a field and then click **Add** to select an individual field. In the parenthesis after the field name, you can see what table and field the rela-

tion will be to. The tables in the Available fields list that have at least one field select will be shown in bold.

3. Click **Next**. Your selections are listed for you to review. Click **Previous** to go back to an earlier page and adjust the selections there or click **Finish** to apply the changes if they are correct.

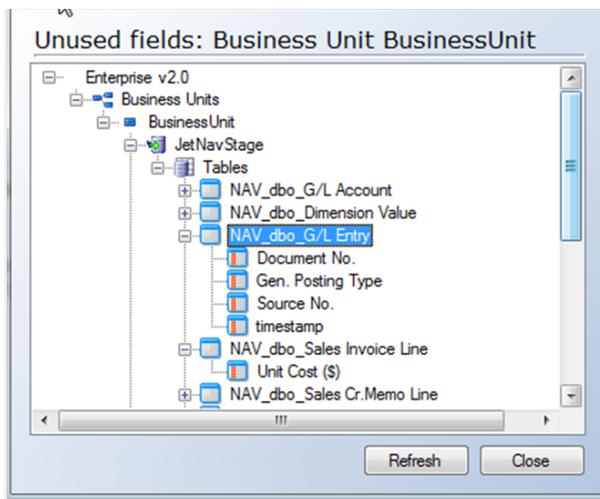
Unused Fields

TimeXtender can display all unused fields in the project. This feature is useful for removing unnecessary objects from the project to decrease clutter and improve performance.

Displaying Unused Fields

To display unused fields

- Right-click the business unit or the data warehouse and click **Find Unused Fields**. The list of unused fields appear.



In the staging database, the list would show fields that exist but are not:

- Copied to the data warehouse
- Used as a data selection rule
- Used as an incremental selection rule
- Used in a strongly typed custom table
- Used in a SQL snippet
- Used as a conditional lookup in another table

In the data warehouse database, the list would show fields that exist but are not:

- Copied to the SSAS Multidimensional Cubes as measures or dimensions
- Used as a data selection rule
- Used as an incremental selection rule
- Used in a strongly typed custom table

- Used in a SQL snippet
- Used as a conditional lookup in another table

Selecting, Validating and Transforming Data

Selecting the right data from the source, validating it and transforming the data if needed are central parts of the data warehouse process.

In TimeXtender you specify data selection rules to ensure that only the data needed for your analysis is extracted from the data source to the staging database.

On the staging database, you perform data cleansing by applying validation and transformation rules to the data. This ensures that only valid data is loaded into the data warehouse.

However, you can also apply selection, validation and transformation rules on a data warehouse. This is useful when you have moved data from different business units into the data warehouse and want to ensure the validity of the consolidated data.

Operators for Selecting and Validating Data

When defining a data selection or validation rule, you can use the operators listed below.

Values must be either integers or letters. You can also specify a list of values by entering comma-separated values.

Operator	Definition
Not Empty	Selects records where the value of a field is not empty or NULL
Equal	Selects records where the value of a field is equal to the specified value
Greater Than	Selects records where the value of a field is greater than the specified value
Less Than	Selects records where the value of a field is less than the specified value
Not Equal	Selects records where the value of a field is not equal to the specified value
Greater or Equal	Selects records where the value of a field is greater than or equal to the specified value
Less or Equal	Selects records where the value of a field is less than or equal to the specified value
Min. Length	Selects records that contain at least the specified number of characters
Max. Length	Selects records that contain no more than the specified number of characters
List	Selects records where the value of a field is equal to one of the specified comma separated values

Empty	Selects records where the value of a field is empty or NULL
Not in List	Selects records where the value of a field is not equal to one of the specified comma separated values
Like	Selects records where the value of a field is similar to the specified value. A percent sign (%) can be used as a wildcard. For instance, "ABC%" will return all records where the value in the specified field starts with "ABC".
Not Like	Selects records where the value of a field is not similar to the specified value. A percent sign (%) can be used as a wildcard. For instance, "ABC%" will return all records where the value in the specified field does not start with "ABC".

Data Selection Rules

Data selection rules are used to specify a set of conditions that data extracted from a source table must satisfy. By applying selection rules, only the subset of data that you actually need is loaded into the data warehouse or staging database.

On data warehouses you can add data selection rules on both the table level and the source table level. If more than one source table delivers data to a data warehouse table, you can set up different rules for each source table, but you can also set up general rules.

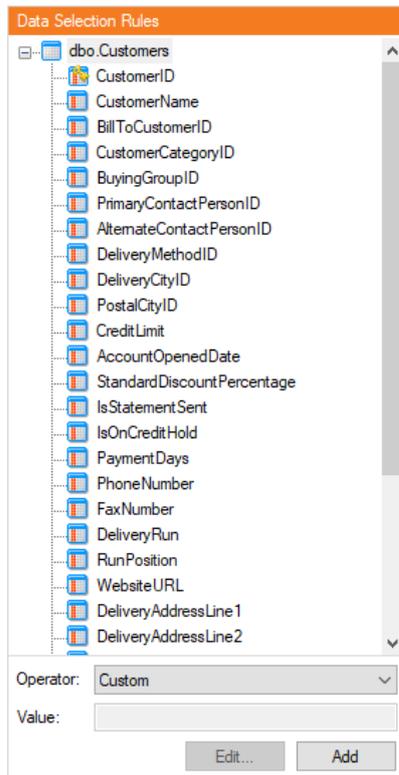
On Qlik models, data selection rules are set on the source table level.

You can add usage conditions to selection rules based on project variables. This enables you to e.g. load less data in a development environment than in the production environment.

Adding a Data Selection Rule

1. Expand the business unit that contains the data source you want to apply the selection rule to, expand **Data Sources** and then expand the relevant data source.
- OR -
Expand the data warehouse that contains the table you want to apply the selection rule to and expand **Tables**. If you want to apply the rule to a specific source table, expand the table and **Mappings** as well.
- OR -
Expand the Qlik model that contains the table you want to apply the selection rule to and expand **Tables**. If the table is concatenated, expand **Mappings** as well.
2. Right-click the table, or source table, you want to add the selection rule to and click **Add Data Selection Rule**.

The **Data Selection** pane appears in the right hand side of the window.



3. Click the field you want to use in the selection rule.
4. In the **Operator** list, click the operator you want to use. See [Operators for Selecting and Validating Data](#).
5. If applicable, type a value for the operator in the **Value** box.
6. Click **Add**.

All selection rules that you have applied to a table are displayed below the relevant table.

Adding a Usage Condition to a Selection Rule

To add a usage condition to a selection rule based on a project variable, follow the steps below.

1. Right click a selection rule and click **Add Usage Condition**. The **Usage Condition** panel is displayed in the right hand side of the application window.
2. In the **Usage Condition** panel, click the variable you want to use.
3. In the **Operator** list, click the operator you want to use. You have the following options:
 - Equal
 - NotEqual
 - GreaterThan
 - LessThan
 - GreaterEqual (Greater than or Equal to)
 - LessEqual (Less than or Equal to)

4. In the **Comparer** list, click the general data type of the variable, which TimeXtender will use in the comparison. You have the following options:
 - String
 - Date
 - Numeric
5. Type the value you want to compare the parameter with in the **Value** box.
6. Click **Add** to add the usage condition.

For more information about project variables, see [Project Variables](#).

Data Validation Rules

Validation rules ensure a high level of accuracy and reliability of the data in the data warehouse and are used to discover invalid data. You can apply validation rules at the field level in the staging database or at field level in the data warehouse.

While data is cleansed on the staging database, it often has to be cleansed again if you have consolidated data from different business units in the data warehouse.

You can make a validation rule conditional if you want the rule to apply in specific situations only.

For each validation rule you apply to a field, you must also classify the severity of a violation. The following classifications are available:

Severity Definition

Warning The violation is not critical to the data quality and does not require immediate attention. The data is considered valid and will still be made available to the end users.

Error The violation is critical to the data quality and requires immediate attention. The data is considered invalid and will not be made available to the end users.

Adding Data Validation Rules

You can add any number of validation rules to a field.

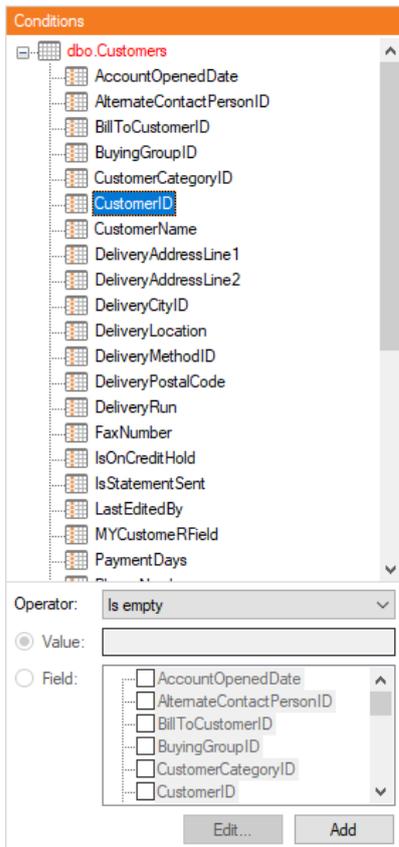
1. Expand the business unit that contains the data source you want to apply the validation rule to, expand **Data Sources**, expand the relevant data source and expand the relevant table.
- OR -
Expand the data warehouse that contains the table you want to apply the selection rule to, expand **Tables** and expand the relevant table.
2. Right-click the field, you want to apply the validation rule to, and click **Field Validations**. The **Field Validations** pane appears in the right-hand side of the window.
3. Click the field you want to use in the validation rule.
4. In the **Operator** list, click the operator you want to use. See [Operators for Selecting and Validating Data](#).

5. If applicable, type a value for the operator in the **Value** box.
6. Click **Error** to specify that as the severity level or leave it at **Warning**.
7. Click **Add** to add the rule.

Adding Conditions

You can add any number of conditions to your validation rules. Follow the steps below to add a validation rule.

1. Locate the selection rule you want to modify.
2. Right-click the rule and then click **Add Condition**. The **Condition** pane is displayed.



3. In the **Operator** list, click the operator you want to use. See [Operators for Selecting and Validating Data](#).
4. In the **Value** field, type the value you want to use in the comparison.
- OR -
Click **Field** and click the field, you want to use in the comparison.
5. Click **Add** to add the condition to the rule.

The condition is displayed below the validation rule or transformation rule it belongs to.

To View Validation Errors or Warnings

1. On the **Reports** menu, click **Errors** or **Warnings**.
2. In the **Database** list, click the database that contains the table you want to view errors or warnings for.

3. In the **Table** list, click the relevant table. The **No. of rows** box displays the number of errors or warnings in the table and the rows that violate the rules are displayed in the pane below.
4. Click any row to display the error or warning message in the **Error Message** or **Warning Message** box.

Data Transformation

Fields transformations lets you modify existing data in a number of ways. You can, for example, easily reverse the sign of numeric values, trim fields or return a specified number of characters from the original field value.

Adding Field Transformation Rules

1. Expand the business unit that contains the data source you want to apply the validation rule to, expand **Data Sources**, expand the relevant data source and expand the relevant table.
- OR -
Expand the data warehouse that contains the table you want to apply the selection rule to, expand **Tables** and expand the relevant table.
2. Right-click the field, you want to add a transformation rule to, and then click **Field Transformations**. The **Field Transformation** pane appears. In the pane, click the field you want to add a transformation to.
3. In the **Operator** list, click the operator, you want to use, and then click **Add**.

Operator	Description
Upper	Converts all text values to upper-case
Lower	Converts all text values to lower-case
First	Returns the number of beginning characters specified by the user
Last	Returns the number of ending characters specified by the user
TrimLeft	Trims padded spaces from the left of the data
TrimRight	Trims padded spaces from the right of the data
Trim	Trims padded spaces from the left and right of the data
Fixed	Inserts a fixed value that is specified by the user
Custom	Allows for custom SQL code to be executed
ReverseSign	Reverses the sign for numeric values
TimeOnly	Returns only the time portion of a datetime field
DateOnly	Returns only the date portion of a datetime field
Replace	Replaces one set of characters with another

4. If you have selected **First** or **Last** as the operator, enter how many characters you want to include in the **Length** field.

Adding Conditions

You can add conditions to transformation rules in the same way that you add conditions to validation rules. See " Adding Conditions" on page 143

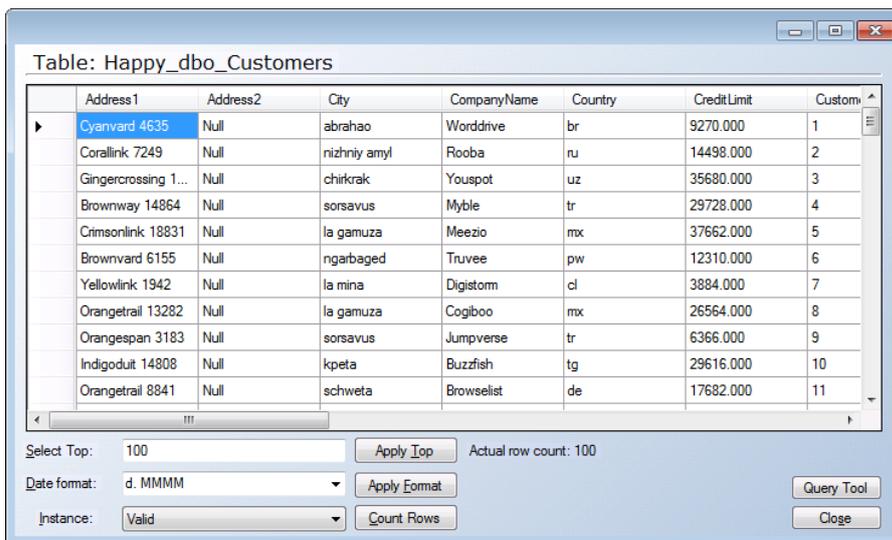
Previewing Data

During development, it is often useful to be able to see what data is present in different tables. For instance, you might want to check that a transformation works as intended. TimeXtender provides two different ways of viewing the content of a table.

Preview Table

The preview table feature gives you a basic overview of the content of a table. To preview the content of a table, follow the step below.

1. Right click a table on a data warehouse or staging database and click **Preview Table**. The preview table window opens.



The screenshot shows a window titled "Table: Happy_dbo_Customers" displaying a table with 11 rows and 7 columns. The columns are Address1, Address2, City, CompanyName, Country, CreditLimit, and Custom. The first row is selected. Below the table, there are controls for "Select Top" (set to 100), "Date format" (set to d. MMMM), and "Instance" (set to Valid). There are buttons for "Apply Top", "Apply Format", "Count Rows", "Query Tool", and "Close". The actual row count is shown as 100.

Address1	Address2	City	CompanyName	Country	CreditLimit	Custom
Cyanvard 4635	Null	abrahao	Worddrive	br	9270.000	1
Corallink 7249	Null	nizhniy amyl	Rooba	ru	14498.000	2
Gingercrossing 1...	Null	chirkrak	Youspot	uz	35680.000	3
Brownway 14864	Null	sorsavus	Myble	tr	29728.000	4
Crimsonlink 18831	Null	la gamuza	Meezio	mx	37662.000	5
Brownvard 6155	Null	ngarbaged	Truvee	pw	12310.000	6
Yellowlink 1942	Null	la mina	Digistom	cl	3884.000	7
Orangetrail 13282	Null	la gamuza	Cogiboo	mx	26564.000	8
Orangespan 3183	Null	sorsavus	Jumpverse	tr	6366.000	9
Indigoduit 14808	Null	kpeta	Buzzfish	tg	29616.000	10
Orangetrail 8841	Null	schweta	Browselist	de	17682.000	11

2. You have a number of options for previewing the data:
 - In **Select Top**, type the number of rows you want to fetch and display. Click **Apply Top** to apply the setting. Please notice that the select top is applied before any sorting of data.
 - In the **Date format** list, click the date format you want to use for dates in the data. Click **Apply Format** to apply the settings.
 - In the **Instance** list, click the instance of the table you want to preview.
 - Click **Count Rows** to
3. When you are done, click **Close** to close the window.

Query Tool

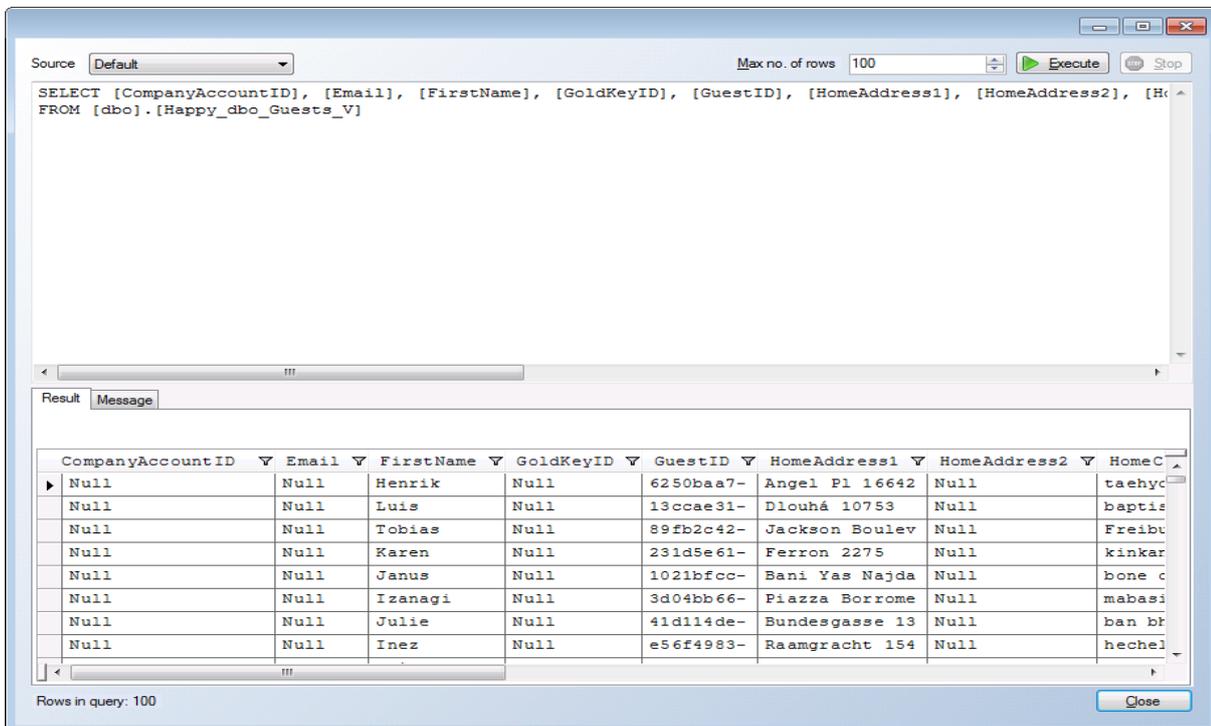
The Query Tool is a powerful supplement to the preview table feature that gives you more flexibility in exploring the content of a table. You can execute any SQL query to see the data you want to see the way you want to see it.

Opening the Query Tool

You can open the Query Tool in three different ways.

- >Right click a table, click **Preview Table** and click **Query Tool** in the **Preview Table** window.
- >Right click a table, click **Advanced** and click **Query Tool**.
- >Click table and press **F8** on your keyboard.

The Query Tool opens with a query that selects the content of the currently selected table, similar to the query that is executed to get the content for the preview table window.



Executing queries

To execute a query, follow the steps below.

1. >Open the **Query Tool** using one of the options described above.
2. >If available, choose the **Source** and **Account** you want to query. **Account** is only displayed when using an adapter with multiple possible accounts.
3. >Enter your query in the top text box of the **Query Tool** window. You can enter multiple queries that will be executed in sequence by TimeXtender.
Adjust **Max no. of rows** to the maximum number of rows you want to have returned.
4. >Click **Execute**
- OR -
Press **F5** on your keyboard.
5. >If you want to stop the query before it completes, click **Stop**.

When the query is complete, you can see the result in the **Result** tab. If you have entered multiple queries, you can select the query result you want to see in **Result set**. If your query resulted in a message, for example because of a syntax error, the Message tab will display this message.

Drag-and-drop and the Query Tool window

The Query Tool supports drag and drop of tables and fields.

- >You can drag a table or a field into the query to place the table name in the query.
- >If you drag a table to an empty query, the default query is generated. The default query fetches everything in the table.
- >If you drag a table from another source into the window, you will be asked if you want to change connection and generate a default query. If you answer **No**, the name is simply added to the query.

Sorting and filtering data

The Query Tool enables you to sort and filter the results.

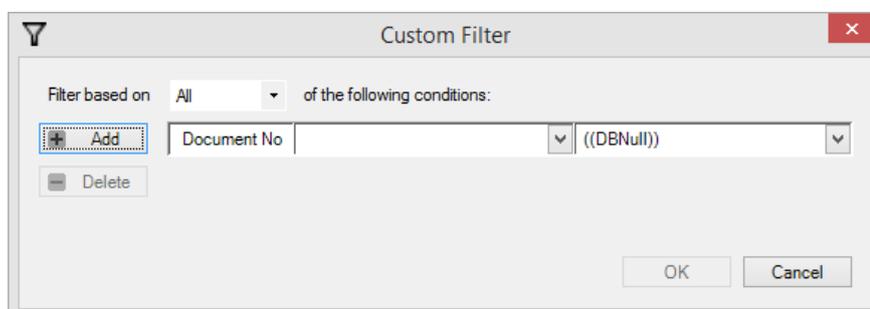
Note: Only the rows returned by the query are available for sorting and filtering in the **Results** tab. If you want to sort or filter all the rows in a table, the most efficient way is to include the conditions in the query, e.g. by using "order by" or "where" clauses. Fetching thousands of rows and sorting them using the tools provided in the **Results** tab can be very slow.

To sort the data, follow the steps below.

1. >Open the **Query Tool** and execute a query as described above.
2. >In the **Result** tab, click a column heading to sort the rows on the value in that column. Click again to switch between ascending or descending order.

To apply a filter, follow the steps below.

1. Open the **Query Tool** and execute a query as described above.
2. >In the **Result** tab, click the filter icon besides the name of the column you want to filter on. You have five filtering options:
 - >(All) is equal to no filtering.
 - >(Custom) opens the **Custom Filter** window, where you can add conditions for filtering.



Each condition evaluates the value of the row field compared to the possible

values in the column. The comparison can be made on **Equals**, **Does not equal**, **Less than**, **Less than or equals to**, **Greater than** and **Greater than or equal to**. Click **Add** to add an additional filter and click **Delete** the currently selected condition. You can choose to filter on **Any** or **All** conditions, i.e. stringing the conditions together with "or" or "and". Click **OK** to activate the filter.

- >(Blanks) shows rows where the column in question is blank, i.e. empty.
- >(NonBlanks) shows rows where the column in question is not blank.
- >A specific value. All unique values in the column is listed and can be chosen as a filter.

Tables

Most tables in TimeXtender are brought in from a data source and moved into the data warehouse directly from the ODX or via a staging database. See [Moving Tables from a Staging Database](#) for more information.

In addition to these tables TimeXtender has six other table types:

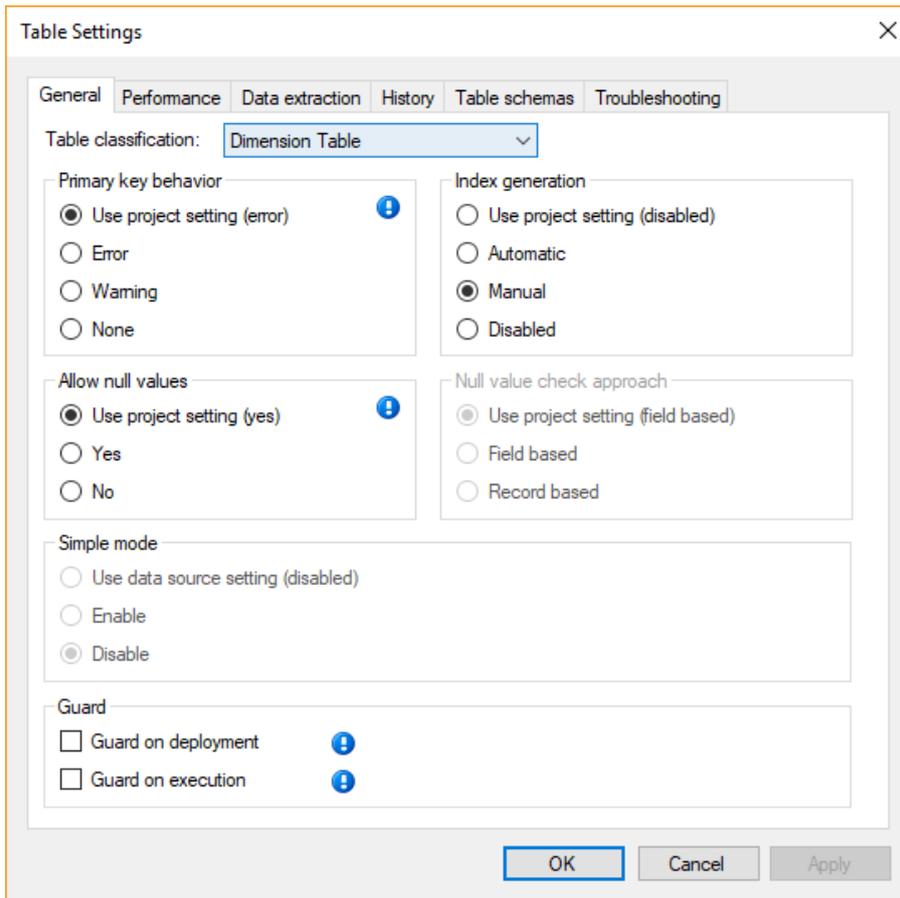
- **Custom tables:** An empty table than can be populated with custom fields.
- **Date tables:** Usually used for creating time dimensions on SSAS Multidimensional cubes.
- **Hierarchy tables:** Used to create special reporting structures based on the other tables in the data warehouse, especially for financial reporting.
- **Junk dimension tables:** A concept in dimensional modeling, junk dimension tables replace multiple fields in a table with a field referencing a row in another table containing the same combination of fields.
- **Aggregate tables:** Creates version of an ordinary table with aggregated data.
- **External tables:** Tables that are "sideloaded" into a data warehouse from another SQL Server database.

Changing Settings for a Table

Most settings for tables are consolidated in the Table Settings window.

To open the table settings window

- Right click a table and click **Table settings**.



Depending on the table type, not all settings are available. For instance, incremental load does not make sense for date and hierarchy tables, so incremental load settings are disabled for these table types.

Simple Mode

Simple mode is a setting on tables on business units aimed at maximizing performance when you need to copy large amounts of data into a staging database to create an exact copy. See [Simple Mode](#) for more information.

Per default, a table inherits the simple mode setting from the data source which in turn inherits the setting from the business unit.

To enable simple mode

- Right click the table, click **Table settings** and click **Enable** under **Simple Mode**.

Guarding a table

Guarding a table tells TimeXtender to skip the table on execution or deployment. This is useful if, for instance, the table contains old data from a legacy system that is no longer running.

To guard a table

- Right click the table, click **Table settings** and then select **Guard on deployment** and/or **Guard on execution** under **Guard**.

Enabling Batch Data Cleansing to Improve Data Cleansing Performance

You can choose to split the INSERT statement up in batches during data cleansing, i.e. when copying data from the transformation view for table to the valid table. This saves log space on the SQL Server which gives you better performance on large tables with 100,000s or millions of rows.

To enable batch data cleansing, follow the steps below.

1. Right click the table you want to use batch data cleansing on and click **Table settings**.
2. Click the **Performance** tab and select **Enable batch data cleansing**.
3. (Optional) Enter the number of records you would like each batch to contain in **Batch size**. The default is 100,000.
4. Click **OK**.

Set Distribution for Tables on Azure Synapse Analytics

For tables on a data warehouse deployed on Azure Synapse Analytics, additional settings are available on the **Performance** tab in **Table Settings**. The **Distribution** option controls how data in the table is distributed among the 60 distributions and x amount of compute nodes on the server.

The following settings are available:

- **Round-robin (default)**: Rows of data are distributed evenly across the server.
- **Replicate**: A copy of all data in the table is stored on all nodes.
- **Hash**: Rows of data are split among the distributions using a hash function based on the values in a 'distribution column' that you set. The default is the system field 'DW_id'.

To set distribution for the table

- Right click the table, click **Table setting**, click **Performance** and then click the setting you want to use under **Distribution**

To set a distribution column for the table

- Right click a field on the table and click **Distribution Column**

Note: Since tables with hash distribution enabled requires a distribution column, you cannot unselect a distribution column. However, you can set another column as your distribution column.

Please refer to Azure Synapse Analytics documentation for more information on distribution.

Custom Tables

Custom tables are basic tables that you can add to the data warehouse or staging database. While they do not initially contain any fields except the standard system fields, you can add new fields to custom tables.

With custom tables, you can, for instance, build your data warehouse first and then map the data in from the data sources.

Adding a custom table

To add a custom table to a data warehouse or staging database

- Right click **Tables**, click **Add Table/ AddCustom Table** and then, in the window that appears, type a name for the table in the **Name** box, and then click **OK**

Adding a custom table FROM A View

You can add a custom table based on a view. The table will be created with a table insert that feeds it data from the view.

To add a custom table with the same structure as a specific view

- Drag the view to the **Table** node on a data warehouse or staging database.

Note: You need to deploy the view and read fields to add a custom table from a view.

Date Tables

You will typically use date tables when you build SSAS Multidimensional cubes on top of the data warehouse. Most often, the cubes you create will contain a date dimension to make it possible to analyze data over time. For example, you may report data on a daily, weekly, or monthly basis. In TimeXtender you use date tables, stored in the data warehouse, as the basis for your time dimensions.

In addition to day of month, day of quarter, week of year and other ordinary information about each date, date tables also contain indexes. An index is a column in the table that tells you something about the date's relation to the current date. Date tables contain year, quarter, month and week indexes. All index values for today's date are 0, while for instance, any day last year would have a year index of -1. This makes it trivial to compare e.g. the same month, day or quarter across years.

Date tables can also contain custom periods, special periods of time, for instance holidays or yearly sales campaigns, that enables you to easily track data across these reoccurring time periods.

Adding a date table

To add a date table, follow the steps below.

1. On a data warehouse, right click **Tables** and click **Add Date Table**. The **Add Date Table** window opens.

The screenshot shows the 'Edit Date Table' dialog box with the following settings:

- Name:** Calendar
- Date range:** Start date: 01-01-2013, End date: 01-01-2015, Days ahead: 1095
- Date display:** Format: YYYY-MM-DD, Seperator: - (dash)
- Week numbering:** First day of the week: Monday, First week of the year: First 4-day week
- Fiscal year:** Calendar year: unselected, Staggered: selected, First month: July
- Custom periods:** Add, Edit, Delete buttons, list box containing US Holidays
- Buttons:** Custom names, OK, Cancel

2. In the **Name** box, type a name for the table.
3. Select a **Date range** by entering a **Start date** and an **End date**. Instead of entering an end date, you can enter a number of days to add to the current date in **Days ahead**. This way, your date table will effectively never end.
4. Under **Date display**, select the **Format** you want to use for dates. You have the following options:
 - YYYY-MM-DD
 - DD-MM-YYYY
 - MM-DD-YYYY
5. Select which **Separator** to use in the format you chose. You have the following options:

- - (dash)
 - / (slash)
 - . (dot)
6. Under **Week numbering**, click the **First day of the week**. You have the following options:
 - Sunday
 - Monday
 7. Select the how to define the First week of the year. You have the following options:
 - First 4-day week (following the ISO 8601 standard, common in Europe)
 - Starts on Jan 1 (common in North American)
 - First full week
 8. Under Fiscal year, click **Staggered** to use a staggered fiscal year and click the first month of the staggered fiscal year in the **First month** list.
 9. (Optional) Click **Add** under **Custom periods** if you want to add a custom period. In the custom period window that opens, you can type a **Name** for the custom period and **Name**, **Start date** and **End date** for the included periods. You can also import and export custom periods by clicking **Import** and **Export** respectively. Click **OK** when you are done.
 10. (Optional) Click **Custom names** if you want to change the names used for days, quarters and months. In the **Date Table Custom Names** window that opens, you can type the names you want to use. The default is derived from the regional settings on the deploying machine. Click **OK** when you are done.
 11. Click **OK** to add the date table.

Hierarchy Tables

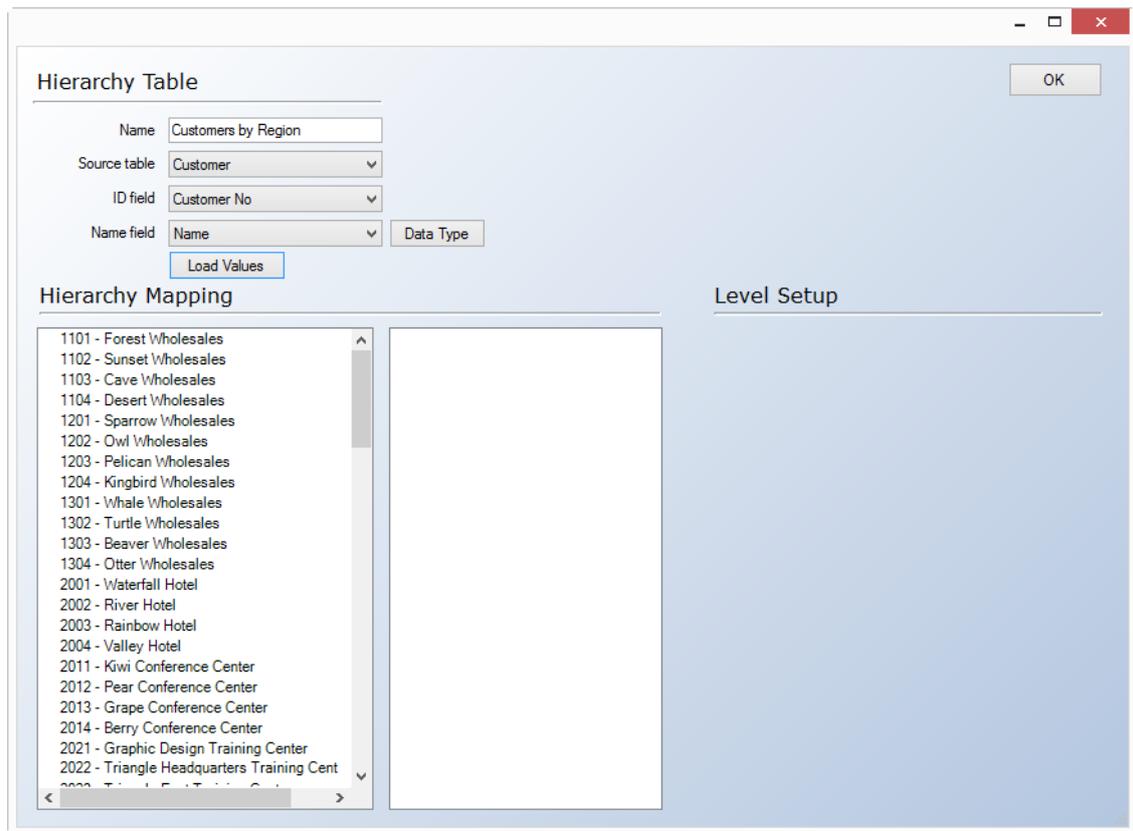
A hierarchy table is used to select data from a table and create a new reporting structure which is different from the structure in the data source. You will typically use a hierarchy table for financial reporting where you want to consolidate data from a number of different accounts, such as ledger accounts.

A hierarchy table is used in conjunction with a parent-child dimension. First, you create the hierarchy table and specify the contents of the table. Then you create a parent-child dimension and add it to a cube. When you build the structure, be sure to choose names that are meaningful to the end-user.

Adding a Hierarchy Table

1. From the **Solution Explorer**, open the relevant data warehouse.

2. Right-click **Tables** and then click **Add Hierarchy Table**. The **Hierarchy Table** window opens:



3. In the **Name** box, type a name for the table.
4. In the **Source table** list, click the table containing the desired data.
5. In the **ID field**, click the field that identifies the individual entries in the table; for example, the customer number. If you need more than one field to identify the entries, you have to create a concatenated field before you create the hierarchy table.
6. In the **Name field**, enter the name that identifies the individual entries; for example, account name, and then click **Load**. The Hierarchy Mapping pane is now populated with the entries of the source table.
7. You can now create the report structure. The structure you create corresponds to the structure of the report that is displayed to the end-user.
8. Right-click in the **Blank** pane, and then select **Add Root Heading**. The root headings become root nodes in the final report.

Hierarchy Table

Name:

Source table:

ID field:

Name field:

Hierarchy Mapping

1101 - Forest Wholesales
 1102 - Sunset Wholesales
 1103 - Cave Wholesales
 1104 - Desert Wholesales
 1201 - Sparrow Wholesales
 1202 - Owl Wholesales
 1203 - Pelican Wholesales

Hierarchy Mapping

1101 - Forest Wholesales
 1102 - Sunset Wholesales
 1103 - Cave Wholesales
 1104 - Desert Wholesales
 1201 - Sparrow Wholesales
 1202 - Owl Wholesales
 1203 - Pelican Wholesales
 1204 - Kingbird Wholesales
 1301 - Whale Wholesales
 1302 - Turtle Wholesales
 1303 - Beaver Wholesales
 1304 - Otter Wholesales
 2001 - Waterfall Hotel
 2002 - River Hotel
 2003 - Rainbow Hotel

Level Setup

From:

To:

Unary operator:

Rollup formula:

9. In **Level Setup**, type a name for the heading in the **Name** field.
10. Right-click the root heading, and select **Add Sub Heading** to add a child node to the structure.
11. In the **Name** box, type a name for the child node.
12. In the **Unary Operator** list, you specify how you want the value of the child node to be aggregated to the sum of all the values in the subheading. The unary operator ensures that the values are aggregated properly in the final report. You have the following options:

Operator	Definition
None	The value is ignored
Add	The value is added to the sum of the values
Subtract	The value is subtracted from the sum of the values
Multiply	The value is multiplied by the sum of values
Divide	The value is divided by the sum of the values

Repeat steps 8-12 for all root headings and subheadings you want to add.

13. Click and hold an entry in **Hierarchy Mapping** , and drag it to the preferred sub-heading in the **Blank** pane. Alternatively, you can specify a range of entries by typing the relevant numbers in **From** and **To**.
14. To exclude an entry from a given range, right-click the relevant subheading, click **Add Exclude**, and then specify a range by typing the relevant numbers in **From** and **To**. Alternatively, right-click the specific entry and select **Change to Exclude**.
15. If a root heading or subheading represents the sum of other subheadings, such as Contribution Margin, you can use a formula to determine the content of the heading. Type a formula in the **Roll-up formula**. Formulas are written in MDX.
16. Click **OK** when you have completed the structure. You can now create the parent-child dimension where the consolidation table will be used.

Note: When you create the parent-child dimension you will typically use Sort By Attribute. You therefore need to create a Sort order dimension level where the key column is Sort order. It is also necessary to enable Unary column and Roll-up column on the dimension. You can then set the parent-child dimension to Sort By Attribute.

Aggregate Tables

An aggregated table is an aggregated version of another table in your project. Often, you will not need the transactional level in financial or sales reports, but only data grouped by business unit or sales team. This makes the aggregated tables feature very useful if you are doing reporting directly from your data warehouse as opposed to using, for instance, SSAS Multidimensional cubes.

Adding an Aggregated Table

To add an aggregated table, follow the steps below.

1. Under **Tables** in a data warehouse, right click the table, you want to add an aggregated version of, click **Advanced** and click **Add Aggregate Table**. The **Add Aggregate Data Table** window opens.

Add Aggregate DataTable

Name:

GroupBy

	Table: dbo.Orders	Field name	GroupBy Type	Override data type
*	▼		▼	▼

Aggregate

	Table: dbo.Orders	Field name	Aggregation Type	Override data type
*	▼		▼	▼

OK Cancel

2. Under **GroupBy**, you can choose what columns on the table the aggregated table should use for grouping the aggregated data. Click the column you want to use in the empty list under **Table: [table name]**. Type a name for the field in **Field name**. If the field you have chosen contains date values, click the list under **GroupBy Type** to adjust the granularity of the grouping. You can choose second, minute, hour and all the way up to year. You can use the same date column multiple times with different **GroupBy** types. For other data types, the **GroupBy** type will always be **Value**.
3. Under **Aggregate**, you can choose what columns from the table you want to have aggregated. Click the column you want to use in the empty list under **Table: [table name]**. Type a name for the field in **Field name**. Click the list in the **Aggregation Type** column and click the method you want to use for calculating the aggregation. You have the following options:
 - **Min**: The lowest value of the field in question.
 - **Max**: The highest value of the field in question.
 - **Count**: The number of rows.
 - **Count_Big**: Same as count, but is able to count higher than 2^{31} , because it uses the bigint data type instead of the int data type.
 - **DistinctCount**: The number of unique values in the field.
 - **Sum**: The sum of all row values.
 - **Average**: The average of all row values.
4. (Optional) Click on **Yes** in the list in **Override data type** column if you want to be able to change the data type for the field. If you right click on the field on the aggregated table, when you have added it, and click **Edit aggregate field**, you will have options to change the data type.
5. Click **OK** to add the aggregated table.

Junk Dimension Tables

A junk dimension is a concept in dimensional modeling. It combines multiple low-cardinality attributes and indicators into a single dimension table as opposed to adding separate dimension tables. This reduces the size of the fact table and makes the dimensional model easier to work with.

The junk dimension table contains a row for all distinct combinations of the junk dimension attributes along with a key that identifies the specific combination. The junk dimension attribute fields can be removed from the fact table and replaced with the single field reference to the junk dimension table.

Multiple tables can utilize the same junk dimension table.

Adding a Junk Dimension Table

Junk dimensions can be added to tables in both data warehouses and staging databases. To add Junk Dimension table for a table, follow the steps below.

1. Right click on a table, click **Advanced** and click **Add Junk Dimension Table**.
2. In the window that appears, select the fields you want to include in you junk dimension table and click **OK**. A new window appears that allow you to customize you junk dimension table.
3. Enter a name for the table in the **Name** box or leave the default ("Dim[table name]Info").
4. (Optional) Click on a table in the **Available tables** list and click add to add it to the junk dimension table. A message will appear to ask you if you want to map fields automatically. The auto mapping algorithm maps a field on the table you are adding to a field on the junk dimension table if one of the following conditions is true: It has the same name as the junk dimension table field or it has the same name as another field that is mapped to the junk dimension table field.
5. Map the fields in the junk dimension table with the fields in the included tables. Each row in the table represents one field in the junk dimension table while each column represents a table.
 - To add a new field to the junk dimension table, click the empty field in the bottom row of the second column and type a name.
 - To remove a field from the junk dimension table, right click the first column of the corresponding row and click **Remove**.
 - To remove a table from the junk dimension table, right click the header row of the corresponding column - the table name - and click **Remove**.
 - To change the order the tables are loaded in, click on the header row of the corresponding column - the table name -and drag it to the desired location.
6. (Optional) In the **Hashing algorithm**list, click on the hashing algorithm you want to use for the dimension table key. See [Default hashing algorithm under Creating a Project](#) for information about the different algorithms. Junk dimensions have a special [link](#)

hashing algorithm available, "Legacy integer", for compatibility with older versions of Analysis Services. You should avoid this algorithm if possible since it is not very safe. It is only 8 bytes which means that the risk of two different data sets giving you the same hash is much higher than with any of the other algorithms.

When you have added a junk dimension table, it appears with a yellow table icon. You can add fields, lookup fields and transformations to the junk dimension table as well as custom data and data inserts.

When you have added a junk dimension table to a table in a staging base, the next step is to add the table and corresponding junk dimension table to the data warehouse. You do not need to add the fields on the table that are part of the junk dimension. This saves you storage in the database.

When the junk dimension table is executed, it will insert non-existing junk dimension combinations from the included table. The junk dimension table has no truncation of the raw instance of the table.

External Tables

External tables is a way to incorporate tables from an existing data warehouse into a TimeXtender project. This is useful if you, for instance, have a legacy data warehouse humming along that you would like to use data from without remodeling it in TimeXtender.

An external table will initially not be deployed or executed, but will be available for data movement to the data warehouse and can be used just as any other table on the data warehouse. By default TimeXtender will create views and read data from the external connection, but you can also chose to move the data into your data warehouse. You can also add a custom SSIS package to the table, which can then be executed.

Adding an External SQL Connection

To add an external table, you first need to add an external SQL connection. To add an external SQL connection, follow the steps below.

1. Right click your data warehouse or a business unit, navigate to **Advanced** and click **Add External SQL Connection**. The **Add External SQL Connection** window appears.

2. Type a **Name** for the connection.
3. In the connection type list, click on the type of connection you want to create. You have the following options:
 - **Cross-database (local)**: Connect to another database on the same server as the data warehouse.
 - **Cross-database (linked)**: Connect to another database on a linked server.
 - **Data transfer**: Connect to another database with an option to transfer data as well.
4. Type the name of the server in the **Server** box. This option is not available if you chose Cross-database (local) as your Connection type.
5. Type the name of the database in the **Database** box.
6. Select **Force codepage conversion** to convert all fields to the collation of the data warehouse.
7. Select **Force Unicode conversion** to declare all alphanumeric fields as **nvarchar**.
8. Select **Allow dirty reads** to allow reading from the source without locking the table.
9. Select **Transfer data** to copy the data from the external SQL Server to your local data warehouse, much like a regular SQL Server data source. This option is only available if you chose **Data transfer** as your **Connection type**.
10. (Optional) Enter additional connection properties in the **Additional connection properties** box.

11. (Optional) Click **Data Extraction Settings** if you want to limit the objects brought into TimeXtender before the data selection stage. For more information, see [Filtering What Objects to Extract](#).
12. Click **OK** to add the connection.

Adding an External Table

To add an external table, follow the steps below.

1. Navigate to **External SQL Connections** under your data warehouse or business unit, right click the connection you just created and click **Read Objects from Data Source**.
2. When TimeXtender has finished reading objects from the data source, the **Source Explorer** pane in the right hand side of the window is populated with the objects from the source. Select the tables, views and fields you want to use in you data warehouse.

Working with External Tables

The external tables in your project are displayed alongside the standard tables and you can use them in the same way. External tables can be used in dimensions and cubes, for reporting, in Qlik models etc. You can recognize an external table on the black table icon.

Some of the transformations and data cleansing you can do with standard tables can be done with external tables as well. You can add custom fields, but not lookup fields. For instance, you can add a custom field to the external table and apply a transformation to the field to concatenate two other fields on the table.

You can also add custom data to an external table.

Deploying an External Table

To deploy an external table, right click the table and click **Deploy**. A View will be created that selects from the external table.

Executing an External Table with an SSIS package

Since an external table is set up outside TimeXtender, TimeXtender expects it to be executed separately from your project. This means that you initially will not find any execute command on an external table. However, if you have a SSIS Package that is used to populate the table, you can add this package to the table and get the ability to execute the table.

1. Right click an external table, navigate to advanced and click **Customize code**. The **Customize Code** window appears.
2. Click the **Add** button to the right of **SSIS Package**. The **Custom Editor** window appears.
3. In the **Editor Name** list, click you editor of choice and click **OK**. The **Custom SSIS** window appears.

4. Make sure **Existing Package** is selected and click **OK**. The **Pick SSIS Package** window appears.
5. Type the server name in the **Server** box. Optionally, you can select **Use SQL Server Authentication** and type your credentials in the **User Name** and **Password** boxes as appropriate. In the **Location** list, click **File system** or **SQL Server** and then click ...next to the **Package Name** box to browse for the SSIS package. When you have found the package and clicked **Open** in the **Open** window, click **OK** and the editor of your choice opens.
6. Make any changes you want to make in the editor, save the package and close the editor.
7. While you edit the SSIS package, TimeXtender displays the **Custom Code Editor** dialog. When you return to TimeXtender, click **Import** to import the changes you made to the SSIS package.
8. In the **Customize Code** window you'll notice that the Add command next to **SSIS Package** has changed to **Edit** and that you can now click **Parameters** and **Delete** as well. Click **Close**.
9. Right click the table and choose **Execute** to run the SSIS package. You can also execute the table by including it in an execution package, executing the entire project etc.

Fields

Most fields in a TimeXtender projects comes from data sources, but you can also add new fields to tables on data warehouse or staging databases. You can add custom fields - simply called "fields" on data warehouse databases - conditional lookup fields, custom hash fields and supernatural keys.

Cloning a field

All fields a data warehouse or staging database can be cloned, no matter if they are brought in from a data source or added later.

To clone a field, follow the steps below.

1. Right click the field and click **Clone Field**. The **Clone Field** window appears
2. In the **Name** box, enter a name for the field.
3. Two options are available:
 - **Clone structure**: The cloned field will be an exact copy of the original field including transformations and validations
 - **Clone values**: The cloned field will be a custom field that gets its value from the original field.

Including fields in the Primary Key

All tables in your project can have a primary key that uniquely identifies every row in the table. The key can consist of one or more fields.

There are multiple advantages to and uses for primary keys:

- TimeXtender can enforce the primary key constraint, i.e. that all the primary keys are unique.
- You avoid duplicate values in your dimensions when you consolidating data from different business units.

To include a field in the primary key for at table

- Right click the field and click **Include in Primary Key**.

Hiding a Field from the Valid Instance of a Table (Raw-only Fields)

A raw-only field is a field that only exists in the raw instance of the table and not the valid instance. This way, the field won't show up in SSAS Multidimensional cubes, Qlik models and other front-ends that use the valid instance of the table. This is very useful if you have fields one the table with no other purpose than to be part of other fields, e.g. a surrogate key when you do dimensional modeling.

To make a field raw-only

- Right click the field and click **Raw-only field**.

Tagging A Field

Fields in a data warehouse or staging database can be tagged. The tags used as the basis for [project perspectives](#) and for the [data impact documentation type](#).

When you tag fields because you want to create data impact documentation, make sure to tag the fields as "early" as possible, i.e. in the staging database. That way, you will get the journey of the data from source to endpoint captured in the documentation.

Before you can add a tag to fields, you have to add the tag itself.

To add a new tag

- Right click **Tags** in the **Solution Explorer** and click **Add tag**. Name the tag and click **OK**.

Once you have added one or more tags, you can add the tags to fields.

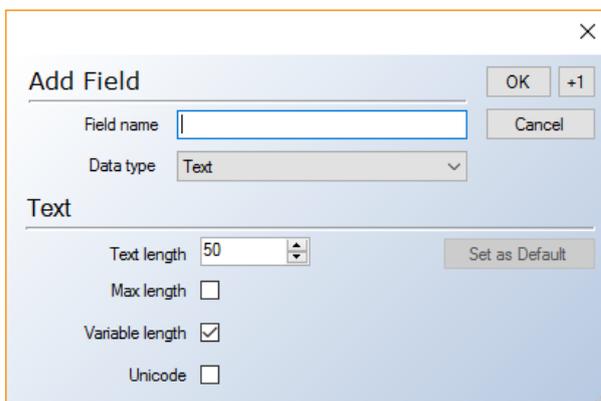
To add a tag to a field

- Right click the field, click **Tags** and click on the tag you want to add to the field.

Custom Fields

Custom fields - or just "field" in the data warehouse - is basically an empty shell that you need to add data to through transformations, scripting or data copy. It has many uses - for instance, you can build a model of your data warehouse before bringing data in. Follow the steps below to add a custom field.

1. Right click a table and click **Add Field/ Add Custom Field**. The **Add Field/ Add Custom Fields** window appear.



2. In the **Field name** box, type a name for the field.
3. In the **Data type** list, click on the data type you want to use for the field.
4. Define the attributes of the selected data type. You have the following options:

Data Type	Attribute	Description
Text	Text length	The maximum number of characters the field

		can contain.
	Max length	The field can contain any amount of characters up to a storage size of 2 GB.
	Variable length	The field can be of variable length.
	Unicode	Characters are encoded in Unicode
Integer	Type	The size of the integer: bigint, int, smallint, tinyint.
Numeric	Numeric precision	The number of digits in the field.
	Numeric scale	The number of digits to the right of the decimal point
	Mantissa bits	The precision of the floating point number if floating point is selected. 1-24 bits equals single precision, while 25-53 bits equals double precision.
	Floating point	The number is saved using floating point notation.
Binary	Length	The length of the binary field.
	Max length	The field can contain any amount of characters up to a storage size of 2 GB.

- (Optional) If you want to use the same attributes the next time you add a field of the same type, click **Set as Default**. The default the button is disabled, the current settings are the default settings.
- Click **OK** to add the field and close the window or **+1** to save the field and add another field.

Conditional Lookup Fields

A conditional lookup field gets - looks up - its value from a field in another table - the source table - in the same data warehouse or staging database. One conditional lookup field can look up values on more than one source field. Conditions can be applied to calculate which of the source fields will deliver the value of the conditional lookup field.

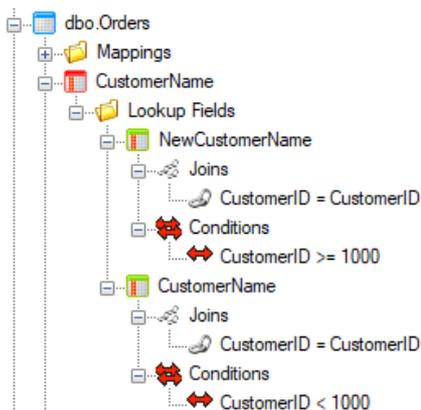
Example

Conditional lookup fields have many uses, and the following is just one example of how to use them.

You have two tables, Customers and Orders, and you want to add the CustomerName field from the Customers table (source) to the Orders table (destination). To do this, you add a

conditional lookup field on the Orders table that lookups up the value of CustomerName on the Customers table using the CustomerID field, that is available on both tables, as a join.

Later, you get a new CRM system while the old system is still online so users can access legacy data. This means that there are now two tables that could contain the CustomerName value: Customers and NewCustomers. To handle this, you add a new lookup field to the conditional look up field. The new field looks up the CustomerName field in NewCustomers table. Next, you have to add a condition to each lookup field to decide when to use the value from that field. Luckily, the CustomerIDs in the two systems do not overlap, so you can add conditions so that the value is sourced from the old table for any CustomerIDs under 1000, while the new table is the source for the rest.



Adding a Conditional Lookup Field

The easiest way to add a conditional lookup field is to use a drag-and-drop operation.

To add a conditional lookup field

- Drag the field to be looked up from the source table to the destination table. If you drag with the right mouse button to a field on another table, you can select whether to add the conditional lookup field above or below the field being dragged to.

Depending on the existing joins and relations, different options will be available after the drag-and-drop operation.

- If there are no relations between the source and destination tables, and no conditional lookup fields between the two tables that have joins, the **Add Join** window will appear to let you add a new join. See [Adding a Join to a Conditional Lookup Field](#) for more information on adding a join and joins in general.
- If relations or joins exist, a window will appear to let you choose how to proceed:
 - **Use default relation instead of joins:** The conditional lookup field will use the default relation between the two tables as a join. This option is only available if a relation exist between source and destination tables. When a default relation is used as joined, a "read-only" version of the join icon is displayed.

Warning: Use this option with care. The conditional lookup field will fail if you delete all relations between the source and destination tables and might begin to give different values if you change the default relation.

- **Add new joins:** Opens the **Add Join** window to let you add a new join.
- **Copy joins from:** Creates joins on the conditional lookup field based on an existing relation or conditional lookup field you select.

If you are dragging a field from a history-enabled table, you need conditions to specify for what date you want to look up the record. A window appears to help you set up the conditions. You have the following options:

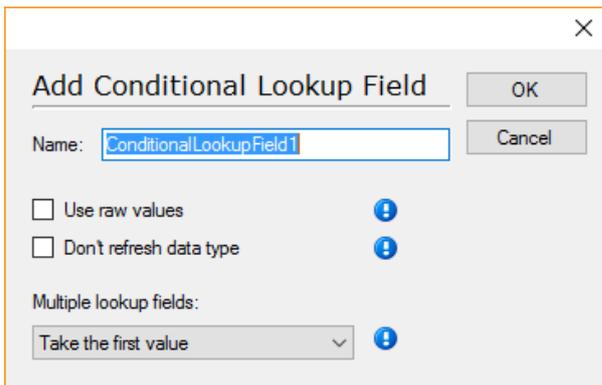
- **Use this field:** Select a date field on the destination table. The conditions will specify that the record should be the one valid on that date.
- **Use the current record:** The condition will be that the record should be the one that is marked as current.
- **I will set up any conditions myself:** TimeXtender will not set up any conditions.

Next, you can [add additional lookup fields](#) or [add conditions](#) to your lookup fields.

Adding a Conditional Lookup Field without Drag-and-Drop

While the easiest way to create a conditional lookup field is using a drag-and-drop operation, it can also be done in a more traditional way. To add a conditional lookup field, follow the steps below.

1. Right click a table in a staging database or data warehouse and click on **Add Conditional Lookup Field**.



2. In the **Name** field, type a name for the lookup field.
3. Select **Use raw values** to perform the lookup on the raw values of the source table instead of the valid values, i.e. before any transformations or other cleansing tasks are performed. Lookups are always inserted into the raw destination table, and this setting does not affect that.
4. Select **Don't refresh data type** to set the data type of the lookup field manually. Per default, TimeXtender will set the data type of the conditional lookup field to the data

type of the source field of the first lookup field. If you enable this option, you can right-click the conditional lookup field, when it has been added, and click **Edit Data Type**. If your conditional lookup field contains lookup fields with different data types, this option is useful to set a data type that can contain all possible values.

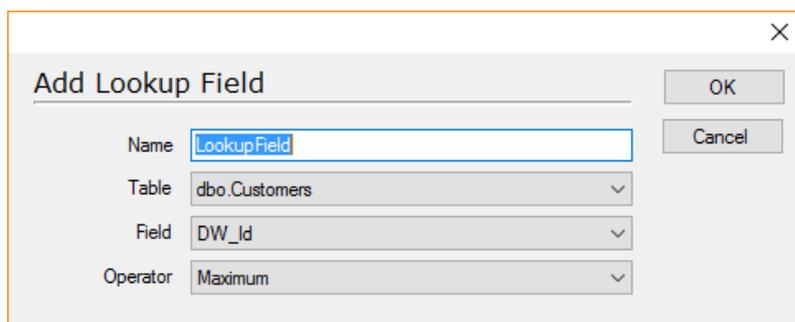
5. In the **Multiple lookup fields** list, select what value is used when a the field contains more than one lookup field. In any case, the lookup fields are evaluated in the same order as they appear in the tree. What happens when there is a match depends on this setting, which can be one of the following:
 - **Take the first value:** The value of the conditional lookup field will be the value of the first lookup field with a condition that evaluates to true.
 - **Take the first non-empty value:** The value of the conditional lookup field will be the value of the first lookup field with a condition that evaluates to true and is not empty.

Adding a Lookup Field to a Conditional Lookup Field

A conditional lookup field contains a lookup field for each source field that is the potential source of the conditional lookup field's value.

To add a lookup field to a cognition lookup field, follow the steps below.

1. Expand the conditional lookup field, right-click **Lookup Fields** and click **Add Lookup Field**.
- OR -
Drag a field from a table on a data warehouse to **Lookup Fields** under the conditional lookup field. This pre-fills some of the settings in the **Add Lookup Field** window that appears.



The screenshot shows a dialog box titled "Add Lookup Field" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Name:** A text input field containing "LookupField".
- Table:** A dropdown menu showing "dbo.Customers".
- Field:** A dropdown menu showing "DW_Id".
- Operator:** A dropdown menu showing "Maximum".
- Buttons:** "OK" and "Cancel" buttons are located on the right side of the dialog.

2. In the **Name** field, type a name for the field.
3. In the **Table** list, select the table containing the field you want to use.
4. In the **Field** list, select the field you want to use.
5. In the **Operator** list, specify how to return the values. You have the following options:
 - **Top:** Returns the value from the first record that matches the join criteria. When you select this operator, a **Sorting** node will be added under the lookup field. Right click this and click **Add Sorting** to define how the matching values are sorted before they are retrieved from the source table.

- **Sum:** Returns a sum of all the values that match the join criteria. This will only work on numeric values. Null values are ignored.
- **Count:** Returns a count of all the values that match the join criteria. Null values are ignored.
- **Maximum:** Returns the highest value of the values that match the join criteria. For strings, it will find the highest value in the collating sequence. Null values are ignored.
- **Minimum:** Returns the lowest value of the values that match the join criteria. For strings, it will find the lowest value in the collating sequence. Null values are ignored.
- **Average:** Returns the average value of the values that matches the join criteria. This will only work on numeric values. Null values are ignored.

6. Click **OK**.

In addition to adding a new lookup field, you can copy an existing one.

To copy a copy a lookup field from one conditional lookup field to another

- Drag the lookup field from one cognitional lookup to the **Lookup Fields** node under another conditional lookup field.

Adding a Join to a Conditional Lookup Field

All conditional lookup fields need at least one join. Joins are used to calculate what value the conditional lookup field should have. If, for example, you are looking up CustomerName from a Customers table, the join could be "CustomerID on the source table equals CustomerID on the destination table".

Less complex joins will make the lookup perform faster. Complexity is a combination of the number of fields in the join and the data type. To get the best performance, use one single numeric field for the join.

To add a join to a lookup field, follow the steps below:

1. Expand the lookup field, right-click **Joins**, and then select **Add Join**.

2. In the **Join Column** list, select the field that uses the lookup.
3. In the **Operator** field, specify when to look up a value.

4. Click **Field** or **Fixed Value** to specify if you want to compare the field selected in the join column list to a field on the destination table or a fixed value. The **Value** box changes to fit your choice.
5. Depending on the value type, click the relevant field in the **Value** list or enter a value in the **Value** box.

Note: If you do not add a join to a lookup field, it will use the default relation between the source and destination tables as a join if a relation exists.

In addition to adding a new join, you can copy existing ones when the source and destination tables are the same.

To copy a join from one lookup field to another

- Drag the join from one lookup to the **Joins** node under another lookup field.

To copy all joins from one lookup field to another

- Right click and drag the **Joins** node under one lookup field to the **Joins** node under another lookup field and click **Add to Existing Joins**.

To replace all joins on one lookup field with the lookups from another lookup field

- Drag the **Joins** node under one lookup field to the **Joins** node under another lookup field.

Adding Conditions to Lookup Fields

On each lookup field, you can add conditions for when to use that lookup field. The lookup will only be performed when the condition evaluates to true. For example, if you can determine that the lookup will only find related values when a certain field in the destination table has a certain value or apply a condition to avoid the lookup being performed on many records without finding a matching record.

Conditions must also be used when having multiple lookup fields within one conditional lookup field to determine which lookup field to use. Per default, the first lookup field where the condition evaluates to true will be used, even if it returns a NULL value or finds no matching records. If no conditions are specified, the first lookup field will always be used and any subsequent lookup fields will be ignored.

To add a condition to a lookup field, follow the steps below.

1. Expand the lookup field and click **Conditions**. The **Conditions** task pane appears.
 2. Click on a **Field** in the pane.
 3. In the **Operator** list, click the operator you want to use.
 4. Click **Value** and enter a value to use in the comparison in the box
- OR -

- Click **Fields** and select a field to use for the comparison in the list.
- Click **OK** to add the condition.

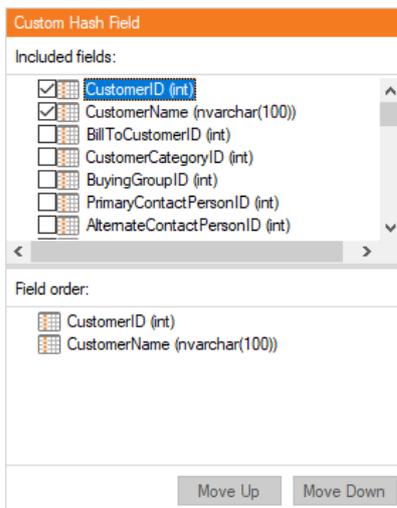
Note: If you select the Custom operator, you can script your own condition. When you drag a field from the list to the right in the script editor, remember to specify if you refer to the field in the raw or the transformation view instance of the table by prefixing the column with "R.[FieldName]" or "T.[FieldName]". If you don't do this, TimeXtender will not be able to tell what field to use and the deployment of the data cleansing script will fail.

Custom Hash Fields

In scenarios with multiple fields making up the primary key, hashing the values of those fields into a single field can improve lookup performance. You can also use the field to easily investigate whether changes has been made to a record or not. In TimeXtender, such a field is called a custom hash field.

Adding a custom hash field

- Right-click a table and click **Add Custom Hash Field**. A custom hash field is added to the table and selected. The **Custom Hash Key** pane appear in the right hand side of the window.



- In the **Custom Hash Key** pane, select the fields you want to include in the custom hash field.
- (Optional) Under **Field order**, you can reorder the fields using drag-and-drop or by selecting a field and pressing **ALT + Up** or **Down**. If you are comparing two fields, it is important that the sequence is the same on both custom hash fields. Otherwise, the hash value will be different even if the values of the individual fields are the same.

Changing the Hashing Algorithm for a Field

For compatibility reasons, we offer a number of different algorithms for hashing fields in TimeXtender. These hashing algorithms are available on all hashed fields in your project.

Apart from custom hash fields, there can be the following hashed fields:

- Junk dimension key
- Surrogate hash key, type I hash key and type II hash key (used for history)

There is usually no reason to change the hashing algorithm for an individual field. The default setting, "use project default", ensures that all hash fields use the same algorithm which in turn makes it possible to compare the values of the individual fields. The most common exceptions are debugging purposes and "upgrading" the hashing algorithm of a field that was created in an earlier version of TimeXtender.

To change the hashing algorithm of a hash field

- Right click the field, click **Hashing algorithm** and click the hashing algorithm you want to use.

Supernatural Key Fields

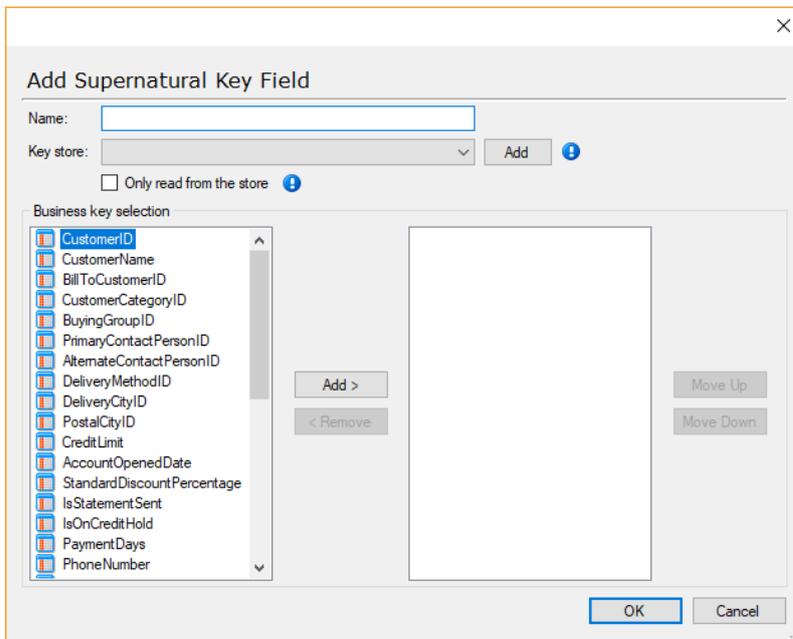
A supernatural key is a key that is independent of the natural keys, e.g. customer numbers, that are found in the data and is durable, i.e. it does not change. Since natural keys can change in the source system and are unlikely to be the same in different source systems, it can be very useful to have a unique and persistent key for each customer, employee etc.

In TimeXtender, supernatural keys work like this: When you create a key, you choose some of the other fields on the table to base the key on. These fields are then hashed together and the hashed value is compared to other values in a key store table. If the same hash already exists, the corresponding key value is returned. If the hash does not exist, it is inserted into the key store and a new key value is generated and returned.

Adding a Supernatural Key Field

To add a supernatural key to a table, follow the steps below.

1. Right click a table in a staging database or data warehouse and click on **Add Supernatural Key Field**.



2. In the **Name** box, type a name for the key.
3. In the **Key Store** list, click on the key store you want to use
- OR -

Click Add to add a new key store. See [Adding a Key Store](#) below for more information.

Note: All supernatural keys that you want to relate should use the same key store. The data type of the field depends on the key store you use.

4. Select **Only read from the store** if you do not want to create a new entry in the key store when the business keys do not match an existing entry. The value of the field will be null if no matching key is found. With managed execution enabled, tables with fields with this option enabled will be executed after tables where the option is not enabled. This ensures the greatest possibility of a matching key in the key store.
5. Under **Business key selection**, the available fields are listed in the left-hand list while the selected fields goes in the right-hand list. Add the fields you want to base the supernatural key on to the right-hand column.

Note: The order of fields matter since two lists of identical fields will only give the same result if they are ordered in the same way. Use the **Move Up** and **Move Down** buttons to reorder the fields.

6. Click **OK** to add the supernatural key field.

Adding a Key Store

Key stores are what ties supernatural key together. You will typically need a key store for each concept, e.g. customer or employee, you want to have a supernatural key for. Key stores exist on the data warehouse and staging database level, i.e. each data warehouse and staging database will have its own key stores.

To add a key store, follow the steps below.

1. If your data warehouse or staging database does not contain a supernatural key field, add one. See [Adding a Supernatural Key Field](#) above for more information. This
2. Expand the relevant data warehouse or staging database, right click **Key Stores** and click **Add Key Store**.

The screenshot shows the 'Add Key Store' dialog box. It has a title bar with a close button (X). The title is 'Add Key Store'. It contains several fields: 'Name' (text input), 'Database schema' (dropdown menu showing '<Default>'), 'Hashing algorithm' (dropdown menu showing 'Use Project Default [SHA-1, SQL Server 2005 +]'), and 'Key data type' (radio buttons). Under 'Key data type', there are three options: 'Unique identifier (GUID)', 'Database unique auto increment (bigint)' (which is selected), and 'Auto increment:'. Below 'Auto increment' are three sub-fields: 'First value' (spin box with '1'), 'Increment' (spin box with '1'), and 'Data type' (dropdown menu showing 'bigint'). There is a 'Set as Default' button next to the 'Database unique auto increment (bigint)' option. At the bottom right are 'OK' and 'Cancel' buttons.

3. In the **Name** box, type a name for the key store.
4. (Optional) In the **Database schema** list, click on the database schema you want to store the key store in.
5. (Optional) In the **Hashing algorithm** list, click on the hashing algorithm you want to use.
6. Click on the key data type you want to use. You have the following options:
 - **Unique identifier (GUID)**: A 16 byte string of characters that is, for all practical purposes, universally unique.
 - **Database unique auto increment (bigint)**: A 8 byte int that is only unique within the database, but has better performance than the unique identifier.
 - **Auto increment**: A customizable auto-incrementing value. You can customize the first value in the key store, the number to increment with when a new row is added and the data type.
7. Click **OK** to add the key store.

Views

A view is a virtual table in your data warehouse or in your staging database where you can group together information from two or more tables in your data source. Views can, for example, be used to provide a user with a simplified view of a table and to limit access to sensitive data.

In TimeXtender, you can add 'regular' views with code generated by the software or custom views where you write the SQL that creates the view.

Views

In TimeXtender, a regular view can be based on a group of regular fields from different tables or lookup fields.

Adding a View Based on Regular Fields

1. Expand the relevant database, right-click **Views** and then click **Add View**.

The screenshot shows the 'New View' dialog box in TimeXtender. The dialog is titled 'New View' and has 'OK' and 'Close' buttons. It contains several sections: 'Schema' (set to '<Default>'), 'Name' (set to 'New'), 'Outer join' (checkbox), 'Field Type' (set to 'Lookup Field'), 'Table' (set to 'NAV_dbo_Sales Invoice Line'), 'Field or function' (set to 'Amount'), 'Alias' (empty), 'Join' section with 'Lookup field' (set to 'Amount'), 'Operator' (empty), 'View table or field' (empty), and 'Fixed value' (checkbox). The 'Sorting' section has 'Lookup table field' (empty) and 'Sort order' (set to 'Ascending'). A dropdown menu is open next to 'Field or function', showing options: AVG, COUNT, MAX, MIN, SUM, TOP.

2. In the **Name** field, type a name for the view.
3. In the **Field** type field, select **Standard Table Field**.
4. In the **Table** list, select the table you want to retrieve data from.
5. In the **Field** list, select the field you want to use in the view.
6. In the **Alias** field, type a name for the view, and then click **OK**. The selected field is displayed in the **View** pane.
7. To add more fields from the same table or a field from another table, click **New**, and then repeat steps 3-6. Do this for all the tables you want in the view.

Note: If you want to add fields from more than one table, the tables must be related.

Adding a View Based on a Lookup Fields

Creating a view based on a lookup field consists of the following steps.

1. Expand the relevant database, right-click **Views** and then click **Add View**.

2. In the **Name** field, type a name for the view. You can also click **Schema** to select a schema and **Outer Join** to use outer join in the view.
3. In the **Field Type** list, select **Lookup Field**. The dialog changes so that you can create and specify the properties of the lookup field:
 - Click the **Table** list and select the table that holds the lookup field.
 - Click the **Field/Function** list, select the field or function you want to use and then specify which values to return. You have the following options:
 - **TOP** returns the value of the first record in the column.
 - **SUM** returns the sum of all field values in the column.
 - **COUNT** returns the number of records.
 - **MAX** returns the maximum value of the records in the column.
 - **MIN** returns the minimum value of the records in the column.
 - In the **Alias** field, type a name for the lookup field if you want the name to be different from the source field name.
4. Click **OK**. The field is displayed in the **View** pane of the window.
5. Click **New** if you want to create another field.

- You have to specify a join between the view table and the lookup tables.

- Click the **Lookup field** list and select the field to look up.
- In the Operator field, select the operator that determines how you want the columns compared.

Operator	Description
Equal	Returns values that are equal
Greater	Returns values that are greater than the value of the lookup field or the specified fixed value
Greater or Equal	Returns values that are greater than or equal to the value of the lookup field or the specified fixed value
Less or Equal	Returns values that are less than or equal to the value of the lookup field or the specified fixed value
Less Than	Returns values that are less than the value of the lookup field or the specified fixed value
Not Equal	Returns values that are different from that of the lookup field or the specified fixed value

Note: A default inner join is created which only returns results from the rows common to the two joined tables. For the complete set of records from the joined tables, check the Outer Join box at the top of the dialog.

- Click **OK**, and then click **New** if you want to specify a new join.
- You also have to define the sort order.

- In **Lookup Table Field** list, click the field you prefer.
- In the **Sort Order** list, select how you want the results sorted. Results can be sorted in either ascending or descending order.

- Click **OK**, and then click **New** if you want to specify a sort order for another field in the view.
- Once you have completed all steps, and created all the joins you need, click **OK** in the upper right corner of the dialog to create the view.

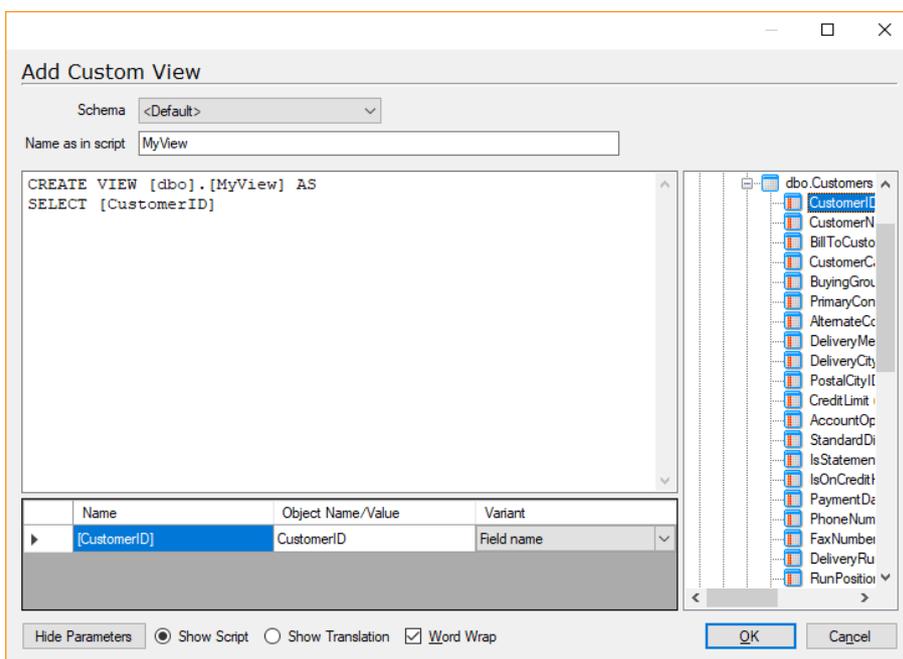
Custom Views

With custom views, you write the SQL that generates the view, giving you a large amount of flexibility.

Adding a Custom View

To add a custom view, follow the steps below.

- Expand the relevant database, right-click **Views** and then click **Add Custom View**.



- In **Name as in script**, type the name of the view. You must use the same name in the script, including schema.
- In the text box, enter the script that creates the view using the standard SQL syntax. Drag tables, fields and variables from the list to the right to use them as parameters. Fields used this way can be mapped to the view's fields to enable tracing - see Mapping Custom View Fields below.
- Click **OK** to save the view.
- Deploy the view and then right-click the view and click **Read View Fields** to show the view's fields in the tree.

To add a custom view based on an existing table

- Drag the table to a **Views** node on a data warehouse or staging database.

Synchronizing View Fields

If you change the view to add, edit or remove fields, you need to synchronize the fields with the project.

To synchronize view fields

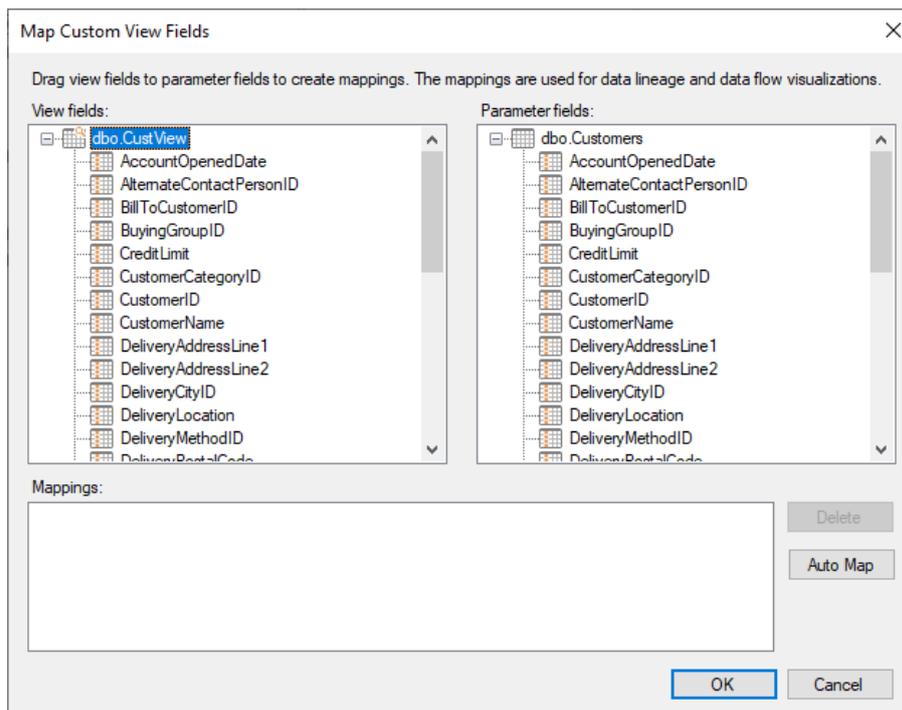
- Right click the view and click **Synchronize View Fields**.

Mapping Custom View Fields

Since a custom view is very flexible, it is not possible to calculate the relationship between the parameterized 'input' fields and the view's fields with certainty. This means that documentation and visualizations, such as data lineage, cannot trace a field from data source to destination if it passes through a custom view. However, this is possible with some manual mapping.

To map the custom view's fields to the parameterized fields, follow the steps below.

1. Make sure the fields have been read, then right click the view and click **Map Custom View Fields....**



2. Drag each field from the **View fields** list to one or more fields in the **Parameter fields** list to create mappings. To delete a mapping, click the mapping and then click **Delete**. Click **Auto Map** to map fields automatically based on name.
3. Once all view fields have been mapped, click **OK**.

Indexes

To achieve the optimal performance on your data warehouse, it is important to have the right indexes on your tables. TimeXtender can generate the necessary indexes automatically or assist you in creating indexes manually. You can also choose to use a legacy approach to indexes instead.

With the Index Automation feature, you can let TimeXtender handle all index creation and maintenance. Index Automation considers the following when designing indexes for the project:

- Relations between tables with relationship type set to Error or Warning
- Joins on conditional lookup fields
- Primary key fields (on the raw instance of the table)
- Selection rules on the data warehouse
- Incremental selection rules on the data warehouse
- Partitioning fields (DW_Partitionkey, DW_TimeStamp)

Index Automation will try to minimize the number of indexes. If two lookups can use the same index, TimeXtender will take advantage of that. In addition to that, TimeXtender takes any manually created indexes into consideration. It will not change your manually created indexes, but it will use them instead of creating similar indexes. The indexes created by Index Automation will be named AutoIndex and postfixed with a number for uniqueness within each table.

Setting up Index Automation

Index Automation is configured on the project level, but can be overwritten on the individual table. The following options are available:

- **Automatic** (default): Index automation updates the indexes whenever the user changes the project in a way that could trigger a new or altered index.
- **Manual**: The user can have TimeXtender create indexes on selected tables. However, these indexes are not managed by TimeXtender. Nothing happens automatically if the table is changed in a way that impacts the indexes.
- **Disabled**: TimeXtender will use the legacy index generation behavior. Indexes will be generated during execution when needed by a data cleansing procedure. However, the same index might be created multiple times, since the index generation behavior is not tuned for performance. In addition to that, these auto-generated indexes are not visible for the end user.

Configuring Index Automation for the Project

To configure the Index Automation setting on the entire project, follow the steps below.

1. Right click the project node, and click **Edit Project**. The **Edit Project** window appears. Click **Advanced Settings...**

2. In the **Index generation** list, click the option you want to use.
3. Click **OK**.

Configuring Index Automation for a Table

To configure the Index Automation setting on a specific table, follow the steps below.

1. Right click the table and click **Table Settings**. The table settings window appears.
2. On the **General** tab, in the **Index Automation** group, click the option you want to use.
3. Click **OK**.

Manual Index Generation

Setting the index automation setting to manual, makes it possible for you to use the index generation features of TimeXtender while maintaining complete control over the indexes in your project. When you run manual index generation on a table, data warehouse, staging database or the project, TimeXtender creates any indexes Index Automation finds necessary. However, you can delete and edit indexes as you see fit. TimeXtender will not create new indexes on the tables unless you run manual index generation again.

Generate Indexes Manually on the Project, a Data Warehouse or a Staging Database

To generate indexes on a data warehouse, staging database or the entire project, follow the steps below.

1. Set or make sure Index Automation is set to manual. See [Configuring Index Automation for the Project](#).
2. Right-click the project, data warehouse or staging database you want to use manual index generation on, click **Advanced** and click **Index Automation(manual)**.
3. A message will appear to tell you how many tables TimeXtender checked. Click **OK**.

Generate Indexes Manually on a Table

To generate indexes on a specific table, follow the steps below.

1. Set or make sure Index Automation is set to manual. See [Configuring Index Automation for the Project](#).
2. Right-click the table you want to use manual index generation on, click **Advanced** and click **Index Automation(manual)**.

Note: If the table you want to add an automatic index to already has one or more indexes, the **Index Automation (manual)** option is not available in the **Advanced** menu. Instead, expand the table, right click **Indexes** and click **Index Automation (manual)**.

History

The history feature is like "track changes" for tables. It allows you to record how data in a table changes over time so you can create reports that show how data looked at a particular moment in time.

The principle behind history on tables is quite simple. Instead of simply storing every unique record once, a version of the record is stored every time an important change is made to the record. To keep track of what record is the currently valid one, meta data fields are added to the table. Two of these, "valid from" and "valid to", can be combined to get the time span the table was valid in.

When a table has history enabled, TimeXtender compares the records loaded in from the data source with the records already in the data warehouse. If the record is new, it is added to the data warehouse. If there is a record with the same key already, hashed versions of the two records are compared to each other to find changes. If no changes are detected, nothing is updated in the data warehouse. If one or more changes are detected, what happens depend on the type of the field or fields that have changed. TimeXtender support the following types:

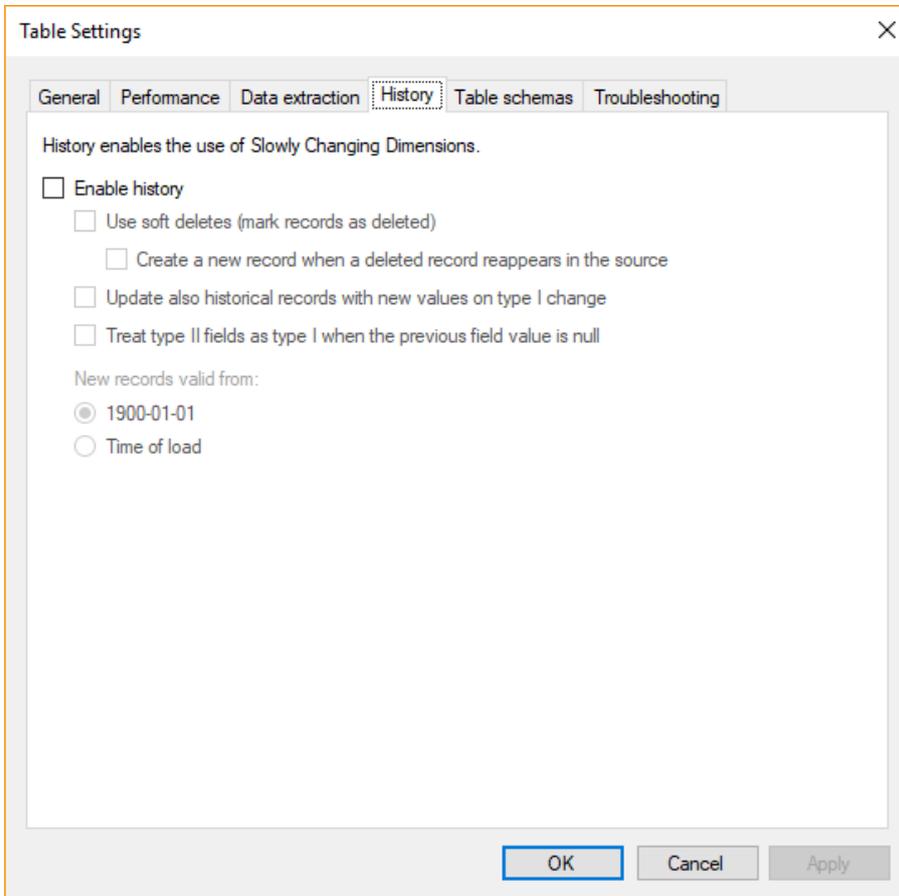
- **Type 0:** Fields that are basically ignored by the history logic. They are inserted together with the rest of the record when it is created – either on the initial load or on a type II change - but they do not trigger an update. Type 0 fields give you flexibility in your history-enabled tables. For instance, you might have a type 0 field with values that are calculated by custom code inside or outside TimeXtender.
- **Type I (default):** Fields that are overwritten with new data when data changes in the source. This means that there will be no history of the change. When a type I field changes, the current version of the record is updated with the new field value. You can also configure TimeXtender to overwrite all instances of the record with the new value. If, for instance, a customer changes name from ABC Consulting to Acme Consulting, the default behavior is to update the current record with the new name. Previous records will still contain the old name, ABC Consulting. If you enable the option, all instances will contain the new name, Acme Consulting. This is often useful in reporting where the purpose is to have a recognizable name for the customer, not the correct name at a specific time.
- **Type II:** Fields that will cause a new record to be created, thus creating history about the change. TimeXtender will create a new record if one of the type II fields on a record has changed, not a new record for each change. The new record will be a copy of the record from the data source. This means that any type 0 fields will be updated.

In addition to these three types, a field can be part of the natural key that uniquely identifies a record to the history logic. The natural key is usually identical with the primary key on the table.

Enabling History on a Table

Follow the steps below to enable history on a table.

1. Right click the table and click **Table Setting**. The Table Settings window appears. Click the **History** tab.



2. Click **Enable History**.
3. Check **Use soft deletes** to mark records that have been deleted in the source system as deleted in the table. This is done by setting the "Is Tombstone" system field on the record to 1. Check **Create a new record when a deleted record reappears in the source** to keep track of history when a record is deleted and later restored in the source system. When this option is disabled, the status of the record will simply change between deleted and not deleted with no information saved about the fact that the record was missing from the source system for a while.
4. Check **Update all records with new value on type I change** to update all versions of a record with the new value when a type I change is detected. The default behavior is to only change the value in the currently valid record.
5. Check **Treat type II as type I when field value is null** to not insert a new record when a type II field changes from null to a non-null value. Enable this when you are not interested in keeping track in this kind of change, e.g. when you have added a new field to the table, thus creating a field where all values are null.

6. Under **New columns valid from**, click **Time of load** to have TimeXtender insert the time of load in the "valid from" field when a new record is added. Depending on your reporting needs, this might make more sense than the default, **1900-01-01**.
7. Resolve any conflicting table settings (marked with error icons) and then click **OK**.
8. Locate the table. The table icon will be overlaid with an "H" to make it easy to identify as a history-enabled table. Expand the node and click on **History Settings**. The **History settings** pane appears.

9. Under **Natural Key**, select the fields you want to use to uniquely identify the records. The primary key fields are selected per default. Note that if you add another field to the primary key, you have to add it to the natural key as well to have it work with the history logic.
10. Select the fields you want to be either type 0 or type II under the respective headings - **Type 0 fields (ignore)** and **Type II fields (insert new record)**. Any fields that you do not select as any of these types will be type I.
11. Deploy the table to have the settings take effect.

Note: You might see a **Clean Up Tombstone Field** button under history settings. This refers to the "Is Tombstone" field used for keeping track of records that have been deleted in the source. Earlier versions of TimeXtender would create an "Is Tombstone" field on history-enabled tables even if deletes was not enabled. Click the button to remove the unnecessary field from the table.

Scripting

TimeXtender generates most of the code you need, but you can extend the functionality of TimeXtender by writing your own scripts. When you need to include custom SQL code in your project, you have different options depending on what you need to do.

- **User Defined Functions** and **Stored Procedures** are used to create reusable code on SQL Server. TimeXtender uses them when it generates the code for executing your project. You can create your own User Defined Functions and Stored Procedures and call them from [Execution Packages](#) or Script Actions.
- **Script Actions** enables you to add snippets of SQL code to be run before or after each step in the deployment or execution of a table.
- **Snippets** are small pieces of parameterized code you can use on the field level or as user defined functions, stored procedures or script actions. In addition to the SQL snippets you use in the data warehouse, snippets come in SSAS Multidimensional and Qlik flavors.
- **Custom Code** lets you replace the code generated by TimeXtender for deployment and execution with your own code written in your favorite development environment.

You can also create global **Project Variables** for use in your scripts.

User Defined Functions and Stored Procedures

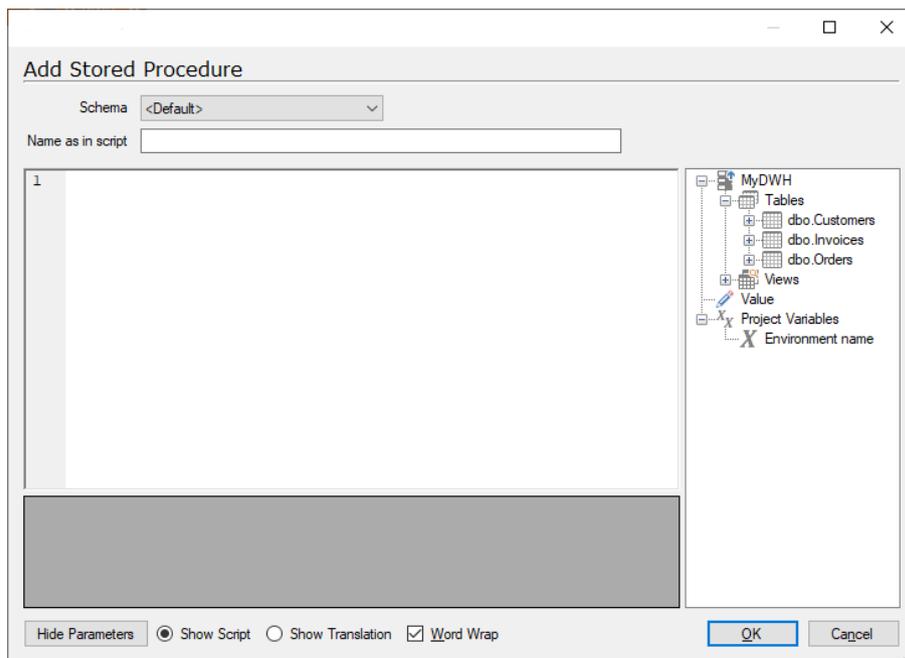
User Defined Functions allow you to define your own Transact SQL functions. A user defined function returns a table or a single data value, also known as a scalar value. You can, for example, create a function that can be used to perform complex calculations.

Stored Procedures allow you to define your own Transact SQL stored procedures.

Adding a User Defined Function or Stored Procedure

You can add user defined functions and stored procedures to both the staging database and to the data warehouse. The steps for adding either one are similar.

1. Under a data warehouse or a staging database, right-click **User Defined Functions** and click **Add User Defined Function**.
- OR -
Under a data warehouse or a staging database, right-click **Stored Procedures** and then click **Add Stored Procedure**.
The **Add User Defined Function** or **Add Stored Procedure** window will appear depending on your previous selection.



2. In the **Name as in script** box, type a name for the function/procedure.
3. In the main text box, enter the SQL code for the user function/procedure. You can drag in tables, fields, views, stored procedures, user defined functions and project variables from the list to the right to use in your function/procedure. Drag in "Value" to create your own custom variable.
4. Click **OK** to save the function/procedure. A **Script Action** can then be created, if necessary, to call the function/procedure.

See where a User-defined Function or Stored Procedure is Used

To see where a user-defined function or stored procedure is used

- Right click the user-defined function or stored procedure and click **Script Usage**.

A window will appear with a list of objects that are associated with the user-defined function or stored procedure.

To navigate to the object in the tree

- Click on an object in the **The script is associated with the following objects** list and then click **Go To**.

Note: Script Usage is also available on script actions.

Script Actions

Script Actions are SQL scripts that can be executed along with deployment or execution of a table to complete a number of different tasks. A Script Action can utilize the User Defined Functions and Stored Procedures that you have already created.

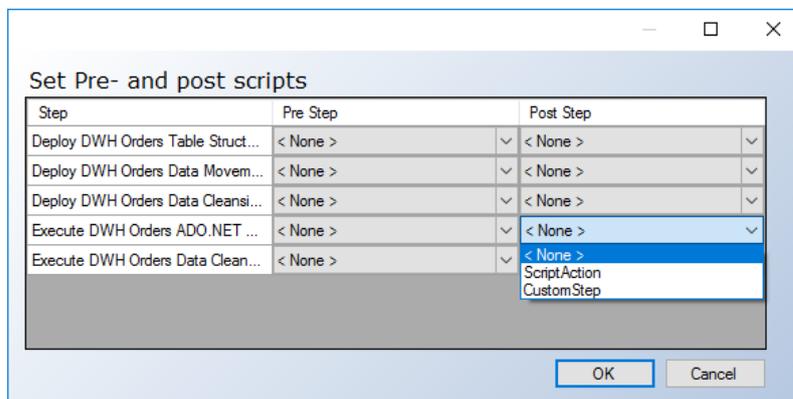
Adding a Script Action

1. Under a **Data Warehouse** or a **Staging Database**, right-click **Script Actions** and click **Add Custom Step**. The **Edit Custom SQL Script** window appears.
2. In the **Name** box, type a name for your script action.
3. Enter your SQL script in the text box. You can drag in tables, fields, views, stored procedures, user defined functions and project variables from the list to the right to use in your script action. Drag in "Value" to create your own custom variable.
4. Click **OK** to save the script action.

Adding a Script Action to a Table as Pre- or Postscript

Script actions are used in the project as pre- or postscripts for a table. As the names suggest pre- and postscripts are executed before - pre - or after - post - the execution steps for the table.

1. Right-click the table you want to associate with the **Script Action**, click **Advanced** and click **Set Pre- and Post Scripts**. The **Set Pre- and Post Scripts** window appears.



2. The deployment and execution steps on the table in question are listed in the window. The steps at which you can call the script are:
 - **Deploy Table Structure**
 - **Deploy Data Movement View**
 - **Deploy Data Cleansing Rules**
 - **Execute Transfer (pre-step)**: This will cause the script to be called prior to the beginning of the data transfer process which will copy data into the table.
 - **Execute Transfer (post-step)**: This will cause the script to be called after the transfer of data into the table has been complete, but prior to the beginning of the data cleansing process
 - **Execute Data Cleansing (pre-step)**: This will cause the script to be called after the transfer of data into the table has been complete, but prior to the beginning of the data cleansing process
 - **Execute Data Cleansing (post-step)**: This will cause the script to be called after the data cleansing process has completed.

Click **<None>** in the Pre Step or Post Step column of the step you want to add you

script to and click the name of the script.

3. Click **OK**.

See where a Script Action is used

Please see [See where a User-defined Function or Stored Procedure is Used](#)

Snippets

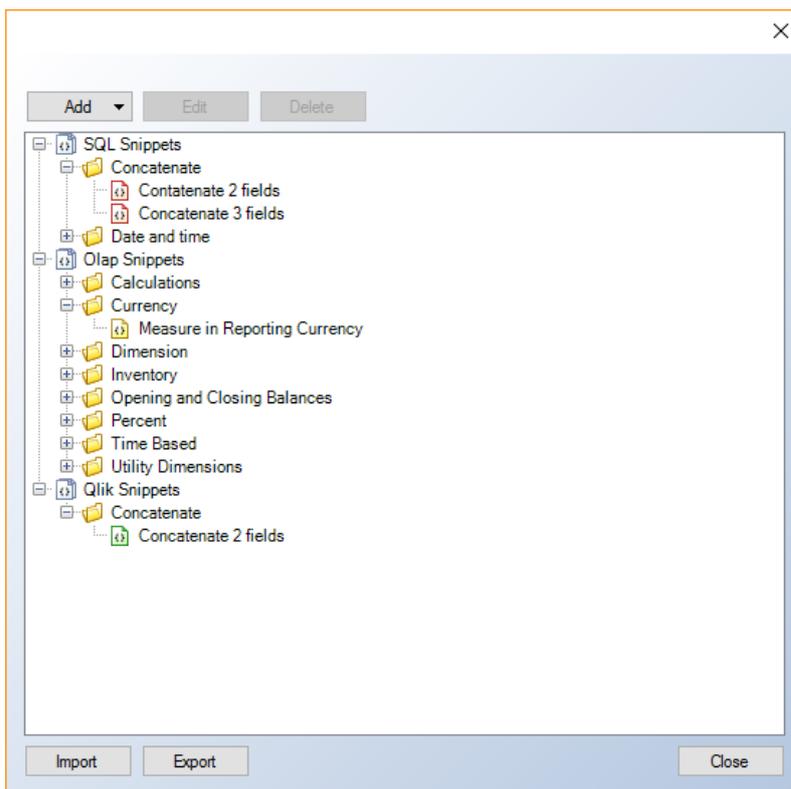
Snippets are reusable, parameterized pieces of code for use in field transformations and other parts of your project. This enables you to write the code once and use it in multiple places, saving you the trouble of maintaining the same functionality on a many fields.

Snippets come in three flavors: SQL, SSAS Multidimensional and Qlik. When you build a data warehouse, you will be using SQL snippets.

Adding a Snippet

Whether you want to create a SQL, SSAS Multidimensional or Qlik snippet, the steps are the same. Follow the steps below to create a snippet and substitute SSAS Multidimensional or Qlik for SQL if that is the type of snippet you want to create.

1. On the **Tools** menu, click **Snippets**. The **Snippets** window appears.



2. Click **Add** and click **SQL Snippet**. The **Create SQL Snippet** window appears.

The 'Create SQL Snippet' dialog box is shown. It features a title bar with minimize, maximize, and close buttons. The main area is divided into several sections:

- Name:** A text input field.
- Description:** A large text area for entering a description.
- Formula:** A large text area for entering the SQL script.
- Parameters:** A table with columns for Name, Type, and Description. The first row is highlighted with an asterisk in the Name column.
- Library Path:** A dropdown menu.

On the right side of the dialog, there are three buttons: 'OK', 'Cancel', and 'Add Parameter'.

3. In the **Name** box, type a name for your snippet.
4. (Optional) In the **Description** box, type a description of what the snippet does.
5. Enter the script in the **Formula** box. For any variables ('FieldName' in the example below), highlight the variable and click **Add Parameter**. This will add the highlighted

text as a parameter name under **Parameters**.

Create SQL Snippet

Name
Convert 1753 Date to Blank

Description

Formula

```
CASE
  WHEN FieldName = '1753-01-01'
  THEN ''
  ELSE
  CONVERT (VARCHAR, FieldName, 103)
END
```

Parameters

Name	Type	Description
▶ FieldName	Field	
*		

Library Path

6. Under **Parameters**, change the **Type** to match what the variable represents. You have the following options:
 - Table
 - Field
 - Database
 - User Defined Function
 - Stored Procedure
 - Value
7. Click **OK** to save the snippet.

Using A SQL Snippet

SQL snippets can be used in a number of situations.

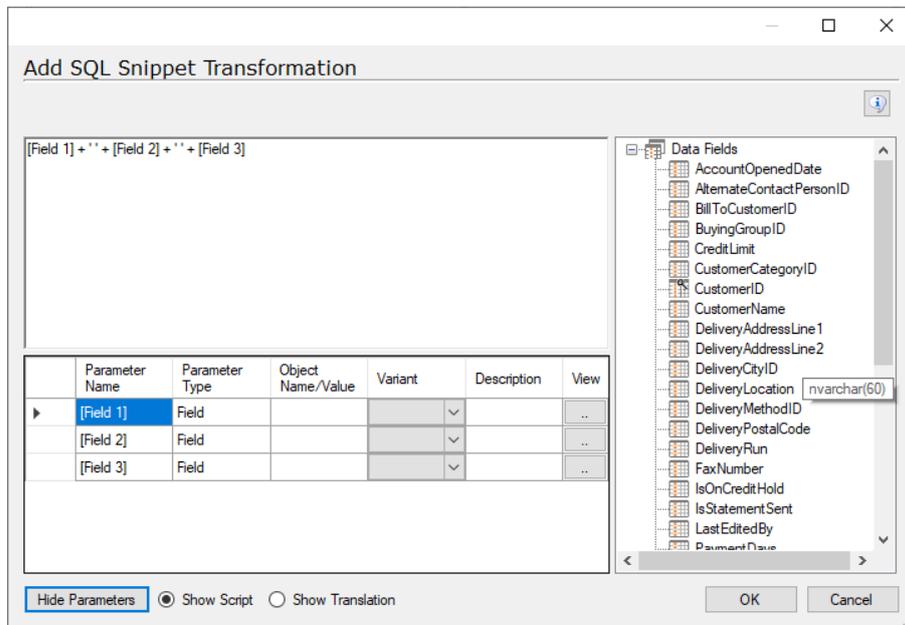
1. To use a SQL snippet in a field transformation, right-click the field, click **Add SQL Snippet Transformation** and click on the SQL snippet you want to use.
- OR -
To use a SQL snippet as a stored procedure , right-click **Stored Procedures**, click **Add Snippet Stored Procedure** and click on the SQL snippet you want to use.
- OR -
To use a SQL snippet as a used defined function, right-click **User Defined Functions**, click **Add Snippet User Defined Function** and click on the SQL snippet you want to

use.

- OR -

To use a SQL snippet as a script action custom step, right-click **Script Actions**, click **Add Snippet Custom Step** and click on the SQL snippet you want to use.

2. In the window that appears, map the available fields to the parameters in the snippet. Drag the field(s) from the list on the right and drop the field on the **Object Name/Value** column for the relevant variable. The **Object Name/Value** column and **Variant** column will populate automatically.



3. Click **OK**.

Changing a Snippet

When you have added an object based on a snippet, you cannot make any changes to the content of the object. If you want to change the content, you need to change the snippet.

To change a snippet for a transformation, stored procedure, user defined function or script action based on a snippet

- Right click the object in question, click **Edit <object type>**, click **Change Snippet...**, select the snippet you want to use and click **OK**.

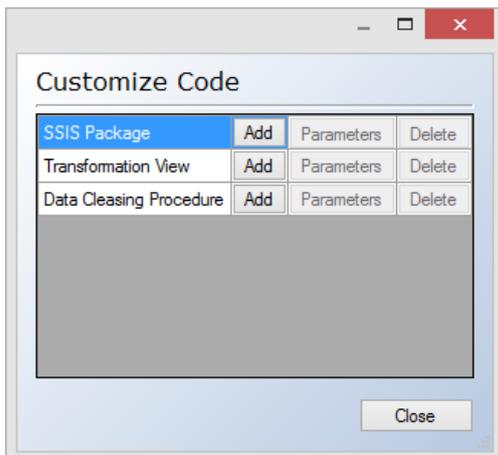
Customized Code

TimeXtender enables you to integrate "hand-written" code into a project by customizing the data cleansing procedure, transformation view and SSIS package on a given table.

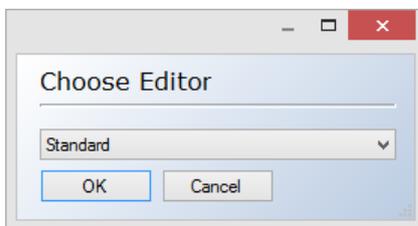
Adding customized code to a table

To customize the code on a given table, follow the steps below:

1. Right click the table in question, navigate to **Advanced** and click **Customize code**. The **Customize Code** window opens.



2. Click the **Add** button to the right of the step you want to customize. The **Choose Editor** window opens.



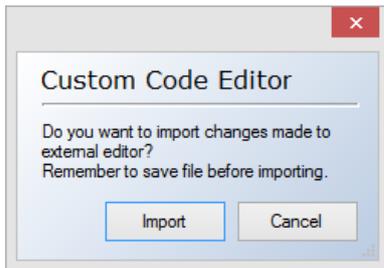
3. In the **Editor Name** list, you have the following options:
 - **Standard** is the basic built-in editor in TimeXtender.
 - **Default File Program** is the program that is set to open files of the type in question. For the data cleansing procedure and the transformation view, the file name extension is .sql. For SSIS packages, the file name extension is .dtsx.
 - Any custom editors you have added (see [Managing Custom Editors](#)).

If you are adding a SSIS package, the **Custom SSIS** window appears. Chose **Create Default Package** to edit the standard package, **Create Destination Only** to create a package that only contains the destination and **Existing Package** to import an existing package from the file system or an SQL Server.

Note: Some tables use multiple SSIS packages. When creating the Default package, TimeXtender will create the first SSIS package only. Examples of tables that will have multiple SSIS packages as default: Data warehouse tables that receive data from multiple staging tables, data source tables from NAV adapters with multiple companies, any data source table when template data sources are used.

4. If you chose the **Standard** editor, the **Edit** window opens. When you have finished editing the code, click **OK** to confirm you edits.
If you chose a custom editor, TimeXtender will open the code in editor you chose. When you have finished editing the code, save your changes and close the editor.

Back in TimeXtender, the **Custom Code Editor** window is open.



Click **Import** to import the changes you have made in the custom editor into your project.

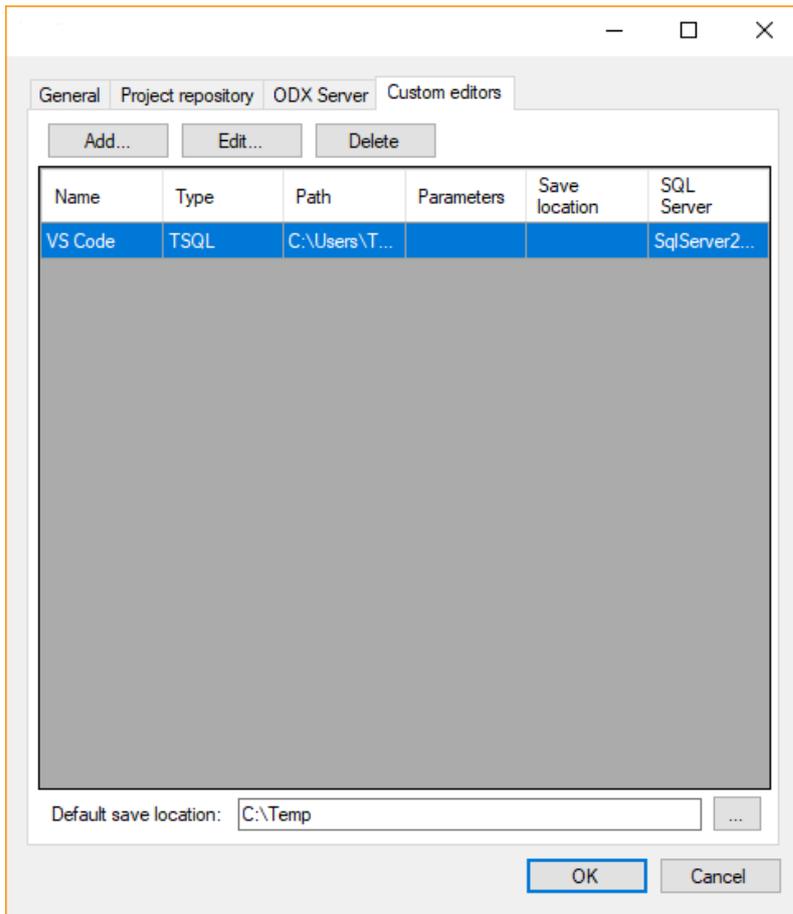
5. When you return to the **Customize Code** window you will notice that you can now click **Parameters** (if applicable) and **Delete**. Click **Delete** to remove the customization and return to having TimeXtender generate the code. Click **Parameters** to decide which parameters are sent to the code on execution.
6. Click Close to close the window.

Note: When editing the data cleansing procedure or the transformation view, make sure to have a "create procedure" or "create view" declaration in the code with the exact same name as TimeXtender would have used. This is what is called during execution. To be sure, simply keep the first line of the code generated by TimeXtender.

Managing Custom Editors

To add, edit or delete a custom editor

- On the **Tools** menu, click **Options**. Click the **Custom editors** tab in the **Options** window that appears.



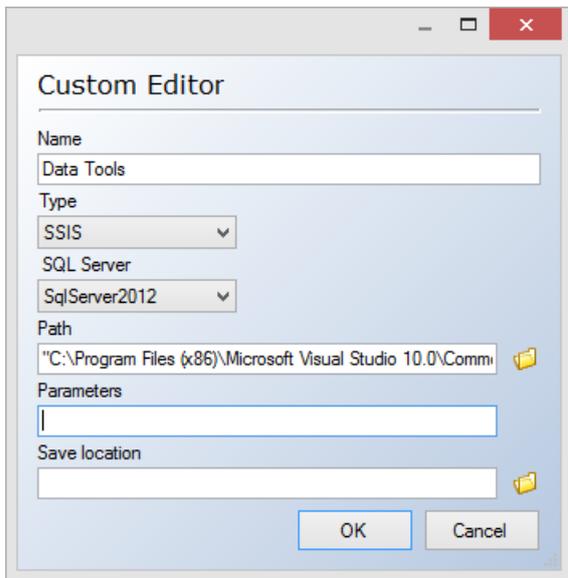
The list of custom editors is displayed. In the **Default save location** box, you can type the path to the folder where the custom code files are temporary stored (or click the folder icon to open a browse dialog).

To edit the settings for a custom editor, select the editor in the list and click **Edit**.

To remove a custom editor from the list, select the editor and click **Delete**.

To add a custom editor, follow the steps below:

1. Click **Add**. The **Add custom editor** window appears.



2. In the **Name** box, type a name for the editor.
3. In the **Type** list, click the type of editor you want to add. Choose TSQL if you want to use the editor with data cleansing procedures and transformation views and SSIS if you want to use it with SSIS packages.
4. In the **SQL Server** list, select the SQL Server version that you are using. Currently, this setting is only used for custom editors for SSIS packages. When you want to customize the code for a SSIS package, TimeXtender checks what version of SQL Server the table is stored on. You will only be able to select editors that are marked compatible with that version of SQL Server.
5. In the **Path** box, type the path that TimeXtender should call to start the program (or click the folder icon to open a browse dialog)
6. In the **Parameters** box, type any additional parameters for the program.
7. Optionally, in the **Save Location** box, type as save location for the editor (or click the folder icon to open a browse dialog).
8. Click **OK** to add the custom editor.

Project Variables

Project variables allows you to save information in project-wide variables. This is useful when you need to distinguish different environments in a script or

The value of a given variable is determined when you deploy the object that uses the variable. As such, when you have changed a variable it is important to deploy the objects that uses this variable. The exception is when you use project variables with customized code. Here, the value of the variables is determined on execution.

The variable does not have a specific data type. If, for instance, you want to use the variable as a string, make sure to enclose the variable in quotes in the script.

Adding a Project Variable

To add a new project variable, follow the steps below.

1. On the **Tools** menu, click **Project Variables**. The **Project Variables** window opens.
2. Click **Add**. The **Add New Variable** window appear.
3. In the **Name** column, type a name for the variable.
4. In the **Type** list, click the variable type you want to use. You have the following options:
 - **Fixed:** A fixed string.
 - **System:** One of the following system properties:
 - MachineName
 - EnvironmentName
 - UserName
 - UserDomainName
 - **Source Scope:** A property of the source of the current object. For instance, if you use a source scope variable in a custom transformation rule on a table in the data warehouse, the variable will have the value of property on the relevant staging database. Since different possible sources have different properties, the variable might not always have a value. Examples of properties include Database Name, API Version, Host, File Name.
 - **Destination Scope:** A property of the destination of the current object, similar to source scope.
 - **Contextual Scope:** A property of one specific element in the project, such as database name on a particular staging database.
 - **Dynamic:** The value of the variable is generated by a custom script you written.
5. If you are adding a dynamic variable, in the **Resolve Type** list, select when you want to resolve the value of the variable. You have the following options:
 - **Every Time:** Resolve the value every time the value is used.
 - **One Time:** Resolve the value when the variable is used for the first time and reuse the resolved value for the following uses until the project is closed.
 - **Each Batch:** Resolve the value once for each batch, e.g. an execution.
6. If you are creating a Contextual Scope variable, click the object you want to use in the **Context** list.
7. If you are adding a Source or Destination Scope variable, click the Value Filter list and click the type of object you want to see available properties for in the **Value** field.
8. If you creating a fixed variable, enter the value of the variable in the **Value** field.

- OR -

If you are creating a dynamic variable, click **Script Editor** to open the standard script editor in TimeXtender and write the script that generates the value.

- OR -

If you are creating a variable of a type other than fixed or dynamic, in the **Value** list, click the property you want to use as a value for the variable.
9. Click **OK**.

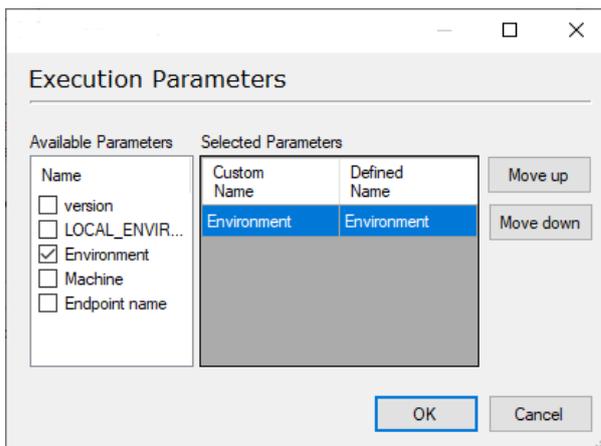
Using a Project Variable in a Script Action, Custom Transformations and Custom Views

Project variables are available for use when writing script actions as well as custom transformation rules and custom views. The available variables are listed in the tree view in the right hand side of the editor window. Simply drag the variable in from the tree to use it.

Using a Project Variable with Customized Code

Project variables are available as parameters when using the customized code feature. To make a project parameter available in a customized step, follow the steps below.

1. Right click a table with customized code. Click **Parameters** next to the step in which you would like to use a project parameter. The **Execution Parameters** window opens.



2. In the **Available Parameters** list, select the variables you want to have available.
3. (Optional) If you need to change the name of a parameter to match your custom code, double-click the parameter name in the **Custom Name** column to rename the parameter.
4. Click **OK**.

Database Schemas

Database schemas allow you to apply a certain schema to a table or a group of tables. You can use schemas to e.g. restrict access to tables that report designers do not need, thereby making reporting off of the data warehouse easier.

Schemas can be set on staging databases and data warehouses as well as data sources and individual tables. The schema settings are applied as follows: Table level settings take precedence over data source settings which in turn take precedence over business unit/ data warehouse settings.

Adding a Database Schema to a Data Warehouse or Business Unit

To create a database schema, follow the steps below.

1. On a data warehouse or business unit, right-click **Database Schemas** and click **Add Database Schema**.
2. In the **Name** box, enter a name for the new schema. In the **Owner** box, you can enter the owning role for the schema. The default is "dbo". Click **OK** to create the schema.
3. Assign a Schema Behavior by right-clicking on the newly created schema. You have the following options:
 - **None**: The schema will be applied to the tables you manually assign it to.
 - **Main default schema**: The schema will be applied to all tables and views in the region (data warehouse or staging).
 - **Main Raw default schema**: The schema will be applied to all Raw (_R postfix) tables in the region (data warehouse or staging).
 - **Main Transfer default schema**: The schema will be applied to all Transfer (_T postfix) views in the region (data warehouse or staging).
 - **Main Valid default schema**: The schema will be applied to all Valid (_V postfix) tables and views in the region (data warehouse or staging).
 - **Main Error/Warning default schema**: The schema will be applied to all Link and Message (_L and _M postfix) tables in the region (data warehouse or staging).
4. If you have selected **None** as the **Schema Behavior**, you need to assign the schema manually. Right-click the table, click **Table Settings** and click the **Table Schemas** tab. Here, you can then select a schema as **Default** (all instances of this table), **Raw**, **Transformation**, **Valid** or **Error/Warning**.
5. Assign user rights to the schema. This can be done through SQL Server Management Studio or T-SQL. See this article on the Microsoft website for details on how to grant user rights using T-SQL: <http://msdn.microsoft.com/en-us/library/ms187940.aspx>

Configuring Schemas for Tables and Data Sources

Schemas for tables and data sources are configured in the settings for the respective objects.

1. Right click a table and click **Table Settings**.
OR

Right click a data source and click **Data Source Settings**.

The settings window for the table or data source appears.

2. Click the **Schema** tab. Here, you can choose the schemas to use for the different instances of the table(s). The lists contain the schemas added to the data warehouse or business unit the table or data source belongs to. Click **Add new schema...** to add a new schema for use on the table or data source.
3. Click **OK**.

Data Security

In TimeXtender, you can control access directly on the data warehouse or staging database. You can restrict access to specific views, schemas, tables and columns on tables - object level permissions - or specific data in a table - data level permissions.

The access control features can be found under **Security** under any data warehouse or staging database.

Adding a Database Role

Object level security is based on SQL Server database roles. A user has access to an object if he is a member of a database role that has access to that object. To add a database role, follow the steps below.

1. Under **Security**, right click **Database Roles** and click **Add Database Role**. The **Database Role Setup** window opens.
2. In **Name**, type a name for the role.
3. If you are using the Multiple Environments feature: In the **Member** setup list, click **Environment Specific Role Members** if you want to have a different setup for different environments. The different environments will then each have a tab in the list below.
4. Next, you should add users to the role. Click **Add login** to role to add Active Directory or SQL Server logins that are known to the SQL Server that the data warehouse resides on. The **Select Login(s)** window opens. Select the logins you want to add in the list and click **OK** to add the user(s).
5. Click **Add manually** to add Active Directory or SQL Server users or groups that the SQL Server does not know, e.g. users on a production server. The **Enter User or Group ID** window opens. In **ID**, type the user or group id. Under **Type**, click **AD user/group** or **SQL user/group** depending on the type of ID you entered. Click **OK** to add the user.
6. Click **OK** to close the window and add the database role, which is listed under **Database Roles**.

Note: Roles are limited to the data warehouse or business unit it was created on, i.e. you cannot use a role created on one data warehouse on another data warehouse or business unit.

If you need to add a new login on the SQL Server, you can right click **Security** under a data warehouse or staging database and click **SQL Server Logins**. Here, you can add logins if you have the necessary permissions on the SQL Server. However, for safety reasons, you cannot delete users here.

On each deployment, TimeXtender drops existing roles on the database before recreating them. By default, TimeXtender only drops database roles related to the data warehouse or staging database being deployed. However, you can also set TimeXtender to drop more

database roles with a setting on the data warehouse or staging database. To access the setting, right click the data warehouse or staging database, click **Edit Data Warehouse** or **Edit Staging Database** as applicable and click **Advanced...** In the **Drop Role Option** list, click **Roles Created by Application** to drop all roles created by TimeXtender or **All Roles on Database** to drop all roles altogether.

Assigning Object Level Permissions to Database Roles

You can assign permissions to database roles on the object level. TimeXtender uses the same allow/deny concept as SQL Server with three possible states:

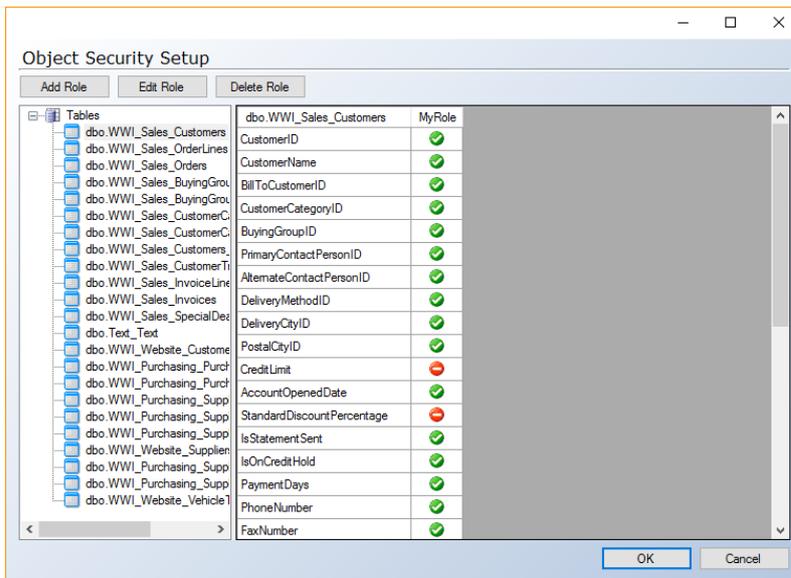
- **Not set** (gray dot): The database role is not allowed to access the object, but are not explicitly denied access.
- **Grant** (green with white checkmark): The database role is granted access to the object. However, if a user is a member of another database role that is denied access, he will not be able to access the object.
- **Deny** (red with white bar): The database role is denied access to the object. Even if a user is a member of another database role that is allowed access, he will still be denied access.

In addition to the three states described above, a table can have different mixed states depending on the column level permissions set on the table. The mixed states are:

- **Partially Granted** (green and gray icon). The database role is granted access to some columns on the table. Note that you will also see this icon if the database role is granted access to all columns on a table since this will not automatically set Allow on the table level.
- **Partially Denied** (red and gray icon): The database role is denied access to some columns on the table. Note that you will also see this icon if the database role is denied access to all columns on a table since this will not automatically set Deny on the table level.
- **Mixed Grant/Deny** (red and green icon): The database role is granted access to some columns and denied access to other columns on the table.

To assign object level permissions, or column level permissions on tables, to database roles, follow the steps below.

1. Under the data warehouse, right click **Security** and click **Object Security Setup**. The **Object Security Setup** window opens.



2. Click **Tables**, **Views** or **Schemas** in the left-hand column to choose the type of object you want to set up access for. Expand **Tables** and click an individual table to assign column level permissions for that table.
3. In the right-hand column, the table shows object names in the left-most column and database roles in the following columns. Click icon in the intersection between the object name and the database role to change the permission for the database role on that object. If you set column level permissions on a table, this will overwrite any current object level permissions set and the other way around.
4. (Optional) Click **Add Role**, **Edit Role** or **Delete Role** to add, edit or delete database roles as needed.
5. Click **OK** to save changes and close the window.

Assigning Data Level Permissions

In addition to configuring access on the object level, you can filter the data available to individual Active Directory users or SQL Server database roles. You might, for instance, want a sales person to be able to see all sales data, but only in his or her own region.

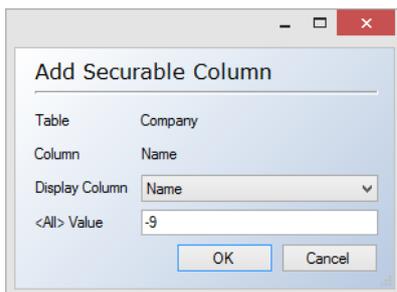
Data level security in TimeXtender is based on the concepts of securable columns, securable column setups, secured columns and secured views. This design allows you to create one security model and reuse it on any number of tables.

- A **securable column** contains the values that we want to use in a filter. Continuing the example above, it could be “sales region id” in a “sales regions” table.
- A **securable column setup** is a mapping between securable column values and users or database roles, e.g. what “sales region id” does the sales person have access to. Each securable column can have multiple securable column setups.
- A **secured column** is a column on the table containing the data we want to filter. This could be a “sales region id” column on a “sales transactions” table.

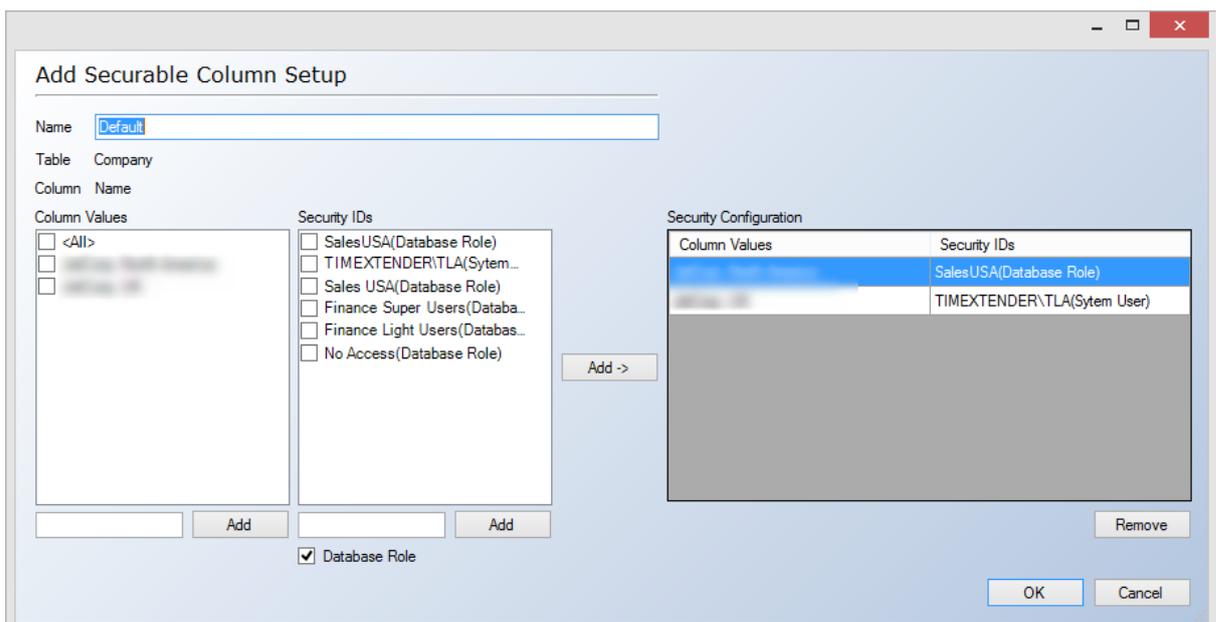
- A **secured view** is a view where all the data the user does not have access to is filtered out. For instance, all the “sales transactions” rows where the “sales region id” does not match the “sales region id” the sales person has access to. When using data level security, the secured view should be used for reporting instead of the table it secures.

To assign data level security to a table, follow the steps below.

1. Expand the table that contains the column you want base the permissions on, right click the field and click **Add Securable Column**. The **Add Securable Column** window opens.



2. (Optional) In the **Display Column** list, click the column value you want to display instead of the column you are adding as a securable column. If the securable column contains e.g. an ID, it might be helpful to choose something that is easier to understand, e.g. a name, as the display column.
3. (Optional) In the **<All> Value** box, type a value that will be used to indicate all values. This value should be a value that is guaranteed not to be among the values in the securable column.
4. Click **OK**. The **Add Securable Column Setup** window opens to let you add your first **Securable Column Setup**.

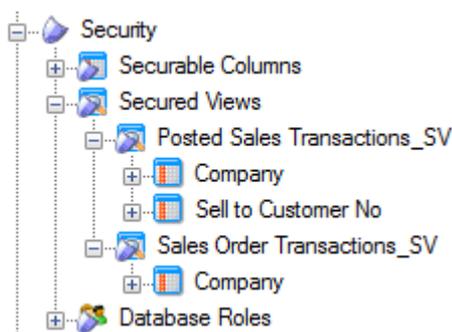


5. In **Name**, type a name for the securable column setup.
6. In the left-hand side of the window, you combine the values in the securable column with the users or database roles that should have permission to access the data. Select a number of values in the **Column Values** list and one or more users or database roles in the **Security ID** list and click **Add->**. The resulting pairs are displayed in the **Security Configuration** list.
7. (Optional) If you need to assign permissions to a value or a security ID that is not in either list, type the value or name in the box under the appropriate list and click **Add**. Select **Database Role** if you want to add a database role as opposed to an Active Directory user. For more information on database roles, see [Adding a Database Role](#).
8. Click **OK** when you have finished configuring the securable column setup. The securable column setup can be found under **Security, Securable Columns, [table name], [securable column name]**.

Applying Data Level Permissions

When you have created a securable column setup, you are ready to use it to apply data level permissions to a table. To do so, follow the steps below.

1. Drag and drop a securable column setup on a field in the table in the data warehouse that you want to secure. A secured view is created and can be found under **Security, Secured Views, [table name]_SV**.
2. If you want to add further permissions to the view, you can drag and drop a securable column setup on the view. The **Add Field** window opens.
3. In the **Field Name** list, select the field that contains the values you want to use in the filter with the securable column setup.
4. Click **OK**. The field is added to the secured view.



Documentation

Audits, regulatory compliance and similar purposes can require printable documentation of your project. Fortunately, TimeXtender lets you create full documentation of a data warehouse, a business unit or an entire project with a few clicks.

The documentation comes in two flavors:

- **Full documentation:** Documents everything in the project so you would, in principle, be able to recreate the project from the documentation. This includes names, settings and descriptions for every object - databases, tables, fields, security roles, etc. - in the project as well as code where applicable.
- **Data Impact documentation:** Documents the impact of fields with a specific tag, i.e. where data from the fields are used in the project.

You can customize the documentation in different ways. Content options - type of documentation, title, subtitle, logo - are saved in templates in the repository. Style options - colors, font, page orientation - are saved in the repository to be used for all templates.

Adding or Editing a Template

Before you generation your first documentation, you might want to add a new template with your custom settings. TimeXtender contains a default template that you are free to edit or delete as well. You cannot, however, delete your last template since TimeXtender needs at least one template.

To add or edit a template, follow the steps below.

1. Click **Generate Project Documentation** in the **Tools** menu to open the **Documentation** window.

2. Click **Manage...** next to the **Template** list.

Name	Type
Default template	Full

Name:

Title:

Subtitle:

Logo: ...

Type:

Include descriptions

- As plain text
- As rich text in attached RTF files

Include project tree

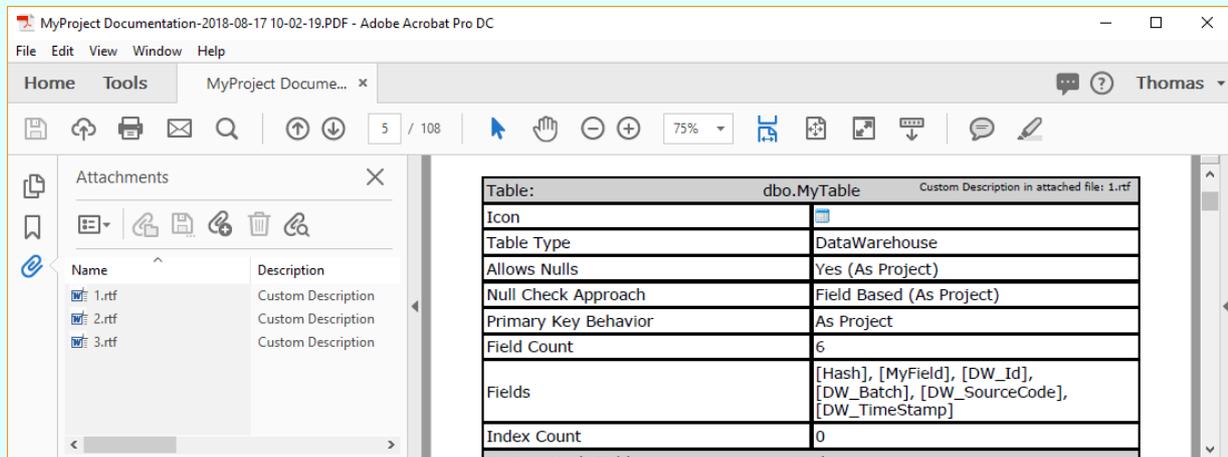
Include diagram of relations on tables

OK Cancel

3. In the window that opens, click **Add** or click on the template you want to edit in the list.
4. In the **Name** box, type a name for the template.
5. In the **Title** box, type a title for the documentation
6. (Optional) In the **Subtitle** box, type a short text to use as a subtitle for the documentation. The text will be displayed on the front page of the documentation.
7. (Optional) In the **Logo** box, enter the path to the logo you want to use. This will be displayed on the front page and in the header of the documentation.
8. In the **Type** list, click the type of documentation, **Full** or **Data Impact** you want the template to be.
9. If you have selected Full, the following options are available:
 - **Include descriptions:** Include the descriptions of objects.
 - **As plain text:** Descriptions will be included in the documentation as plain text without any formatting.
 - **As rich text in attached RTF files:** Descriptions will be included in RTF files attached to the PDF with all formatting retained.
 - **Include project tree:** include a rendering of the project tree in the document.
 - **Include diagram of relations on tables:** Include a diagram for each table that show the table's relations to other tables.
10. If you have selected Data Impact you should enter the tags you want the template to cover. Type a tag name in the box and click **Add** to add it to the list of included tags.

- Click **OK** to close the window. To save your template without generating documentation, click **Apply** and then close the **Documentation** window.

Note: If you have chosen to include descriptions in attached RTF files, they can be found under **Attachments** in Adobe Acrobat Reader. A link to the relevant file is included with the documentation of the object it belongs to.



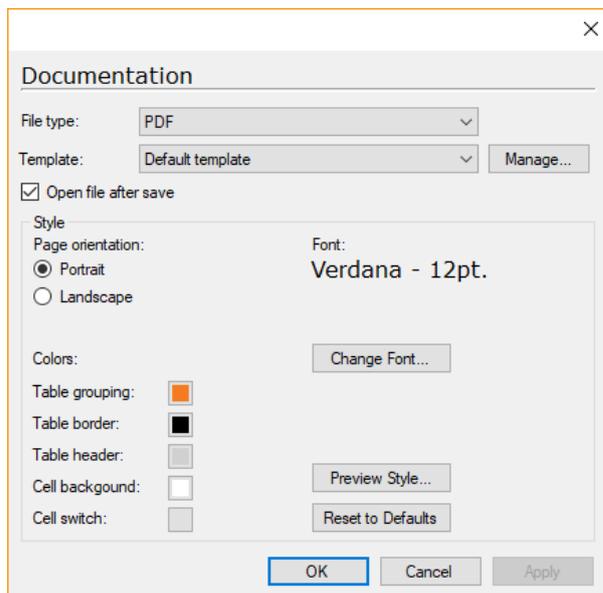
Generating Documentation

The documentation will always cover the project as it looks when the documentation is generated, even if it has not been deployed or even saved yet.

To generate documentation of a part of or your entire project, follow the steps below.

- Right click on the object you want to create documentation for and click **Documentation**. Documentation is available on the project level, data warehouses, business units, SSAS Multidimensional servers, data exports and Qlik models.

The **Documentation** window appears.



2. In the **File type** list, click the file format you want to use.
3. In the **Template** list, click on the template you want to use.
4. Select **Open file after save** to view the document when TimeXtender has generated it. TimeXtender shows the document using external viewers that needs to be present on the machine.
5. Under **Style**, you can configure the look of the documentation. The settings are saved in the repository and used for documentation as well as the files generated by [Export Deployment Steps](#).
6. Click **Change font...** to choose a font for the documentation. A preview is displayed under **Font**.
7. Click the color preview next to the different color details to choose a color.
8. Under **Page Orientation**, select the page orientation you want the documentation to use.
9. Click **Preview Style...** to generate and open a sample file with the colors you have chosen.
10. Click **Reset to Defaults** to reset the style settings to their defaults
11. Click **OK**. In the window that appears, choose a file name and location for the documentation and click **Save**. The default name is the project name followed by "documentation" and a timestamp.

Generating Documentation on a Remote Environment

In addition to the local project, you can also generate documentation for a project on a remote environment in a multiple environments setup. The documentation will contain information on the last deployed version on the environment.

- To generate documentation for a project on a remote environment, right click on the project, click on Multiple Environment Transfer, right click the remote environment and click **Project Documentation**.

Adding a Description to an Object

You can add an description to most objects in your project. This description can then be included in the documentation you generate. To add a description to an object

- Right click the object and click **Description**.

Per default, objects with a description have a bolded name in the user interface. However, this can be disabled if you find it distracting. To toggle the display of objects with a description in a bold font

- Click on **Highlight descriptions** in the **View** menu.

Visualization

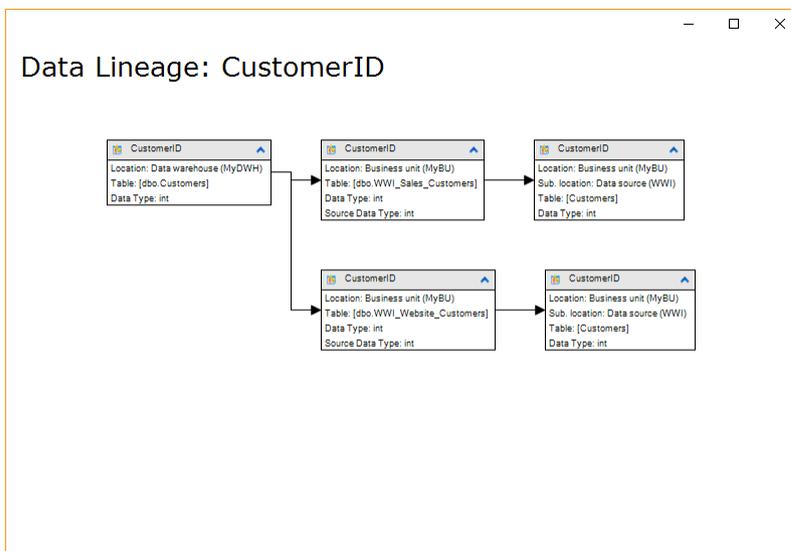
As your project grows, the increasing complexity can make it hard to keep track of all objects and their dependencies. To help you with this, TimeXtender contains three visualization features: Data lineage, impact analysis and relation diagram. The relation diagram is only available on tables, while the other types can be used on most objects in TimeXtender.

The purpose of data lineage is to show you where the object in question gets its data, while the impact analysis feature shows you where the data is used. The relation diagram shows you the relations to and from the table in question.

Showing data lineage, impact analysis or Relation Diagrams

To show one of the diagrams

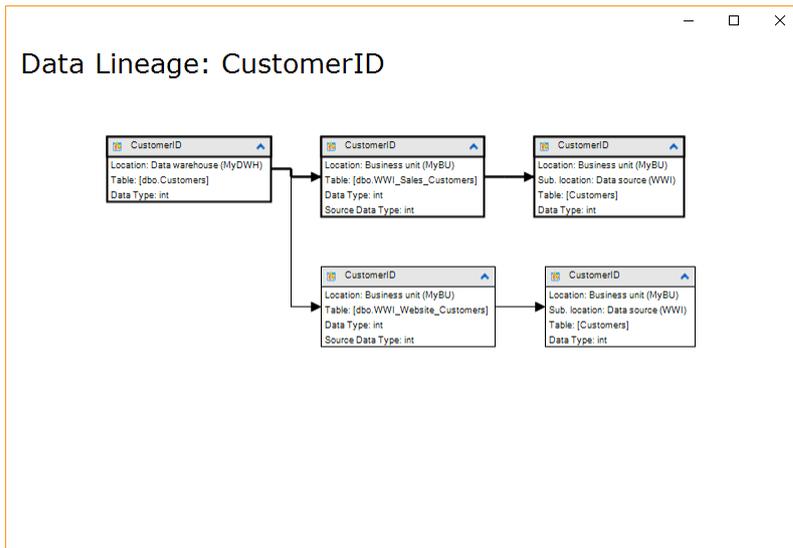
- Right click on a object, click **Visualizations** and click on the type of visualization you want to see.



Customizing the Display

In the visualization window, you have different options to customize the display:

- Click an object to have the connections to and from the object highlighted.



- Click and hold on a object to move it around and release the mouse button to place it.
- Click the arrow in the header of an object to hide or show object details.
- Right-click the background in the window to bring up a menu with different options for organizing and displaying the data:
 - Click **Auto Layout Vertical** or **Auto Layout Horizontal** to make TimeXtender reorder the objects.
 - Click **Collapse All** or **Expand All** to either collapse the objects to create a better overview, or expand the objects to see more details.
 - Click **Zoom In**, **Zoom Out** or **Zoom 100%** change the size of objects.
 - Click **Print** to bring up a **Print** window, where you can select print options and print the diagram.

Performance Recommendations

As you work on your project and add more data, more transformations and more lookups, it is natural that deployment and execution times increase.

To help you take advantage of the performance-improving features we release, the Performance Recommendations tool is available. As the name suggests, the Performance Recommendations tool analyzes your project and recommends changes that can improve the project's performance. You can then choose to apply - some of - the suggested changes. The tool can apply the following fixes:

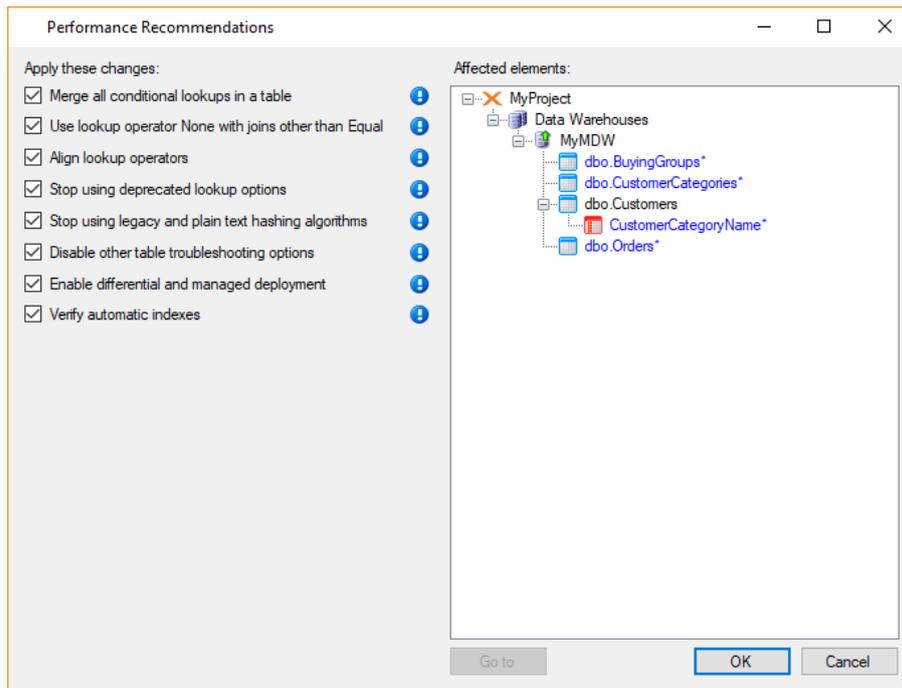
- **Merge all conditional lookups in a table:** Sets the **Merge conditional lookups** option on tables to **Merge all if possible (fastest)**. This can improve performance on tables with many conditional lookups with 20% or more.
- **Use lookup operator None with joins other than Equal:** Changes **Operator** option on lookup fields to **None** if the join on the lookup field does not use the **Equal** operator. This can improve performance since the other operators on lookup fields - average, maximum and minimum - are very slow when the join on the lookup field is not equals.
- **Align lookup operators:** Changes the **Operator** option from **Top** or **None** to **Maximum** on lookup fields on the same table with the same joins. If all operators are either average, count, maximum, minimum or sum the SQL statement can be merged, which improves performance.
- **Stop using deprecated lookup options:** Changes deprecated lookup options to the new defaults:
 - On conditional lookup fields, **Force sub select** is disabled and **Multiple lookup fields** is set to **Take the first value**.
 - On lookup fields, **SQL mode** is set to **Group by**.
This improves performance because it makes it possible to merge SQL statements.
- **Stop using deprecated and legacy hashing algorithms:** Changes the hashing algorithm option from **Legacy Binary**, **Legacy Plain Text** or **Plain Text** settings to **SHA-1 SQL Server 2005+** on the project level, which affects most hashed fields in the project. This improves performance since the new algorithm is faster in addition to being typesafe. The exceptions are supernatural keys and key stores, where changing the hashing algorithm would break existing data, and junk dimensions using the **Legacy Integer** setting, where the hashing algorithm can make sense.
- **Disable other table troubleshooting options:** Disables the **Disable SCHEMABINDING** and **Use legacy transformations** options on the **Troubleshooting** tab on tables. These options decrease performance if enabled and should only be enabled if absolutely necessary.
- **Enable differential and managed deployment:** Enables **Differential deployment** and **Managed deployment** on the project level. These options makes deployment faster by only deploying the objects that need to be deployed and in the right order.

- **Verify automatic indexes:** Verifies that automatically generated indexes are optimal and corrects them if necessary. This cannot be undone with the 'Undo changes' functionality.

Finding and Applying Performance Recommendations

To analyze your project to get performance recommendations and apply them, follow the steps below.

1. On the **Tools** menu, point to **Performance Recommendations** and click **Find...** The **Performance Recommendations** window appears.



2. Review the recommendations in the **Apply these changes** list and clear the check box next to recommendations you do not want to apply. Mouse over objects in the **Affected objects** to see what issue has been identified for the object and the proposed change. Then click **OK** to apply the changes.
3. Deploy the project for the changes to take effect.

Undoing and Accepting Changes

You can undo changes made by the Performance Recommendations tool since undo data was last cleared.

To undo changes

- On the **Tools** menu, point to **Performance Recommendations**, then click **Undo Changes**.

You can clear undo data to disable undoing of the changes made by the tool. This way, you can apply selected changes, test if they work as expected, accept them and move on to another set of changes.

To clear undo data

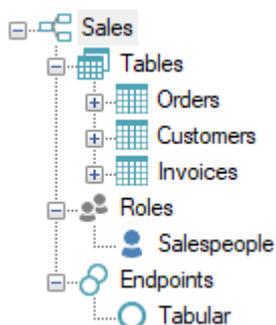
- On the **Tools** menu, point to **Performance Recommendations**, then click **Clear Undo Data**.

Building Semantic Models

Data is not very much use if it is not put to use. In TimeXtender, semantic models, along with SSAS Multidimensional Cubes, serve as the last stop before analysis and visualization tools. As the name suggest, semantic models are part of the semantic layer that "translates" data into forms usable by business users.

Each semantic model can be deployed to one or more endpoints. Endpoints are application specific adapters that connect to a client application such as PowerBI or Qlik Sense. How data is deployed to the endpoint depends on the application. Currently, the following endpoints are supported:

- Analysis Services Tabular ("SSAS Tabular" for short)
- QlikView
- Qlik Sense
- Tableau



Adding a Semantic Model

To add a semantic model, follow the steps below.

1. In the **Solution Explorer**, right click **Semantic Layer** and click **Add Semantic Model**. The **Add Semantic Model** window appears.
2. Enter a name for the model in the **Name** box.
3. Under **Show settings for these endpoints**, clear the checkbox next to endpoint types that you will not be using. Settings unique to these endpoints will then be hidden in the user interface.

Note: Settings that only applies to some endpoint types are marked with an info icon with a tooltip that lists the endpoint types.

4. (Optional) Click **Guard on deployment** or **Guard on execution** to prevent the model from being deployed or executed.

Note: If you use multiple environments, a related setting is available for each endpoint in **Environment Properties**. If you set the **Active** setting to **False**, the endpoint

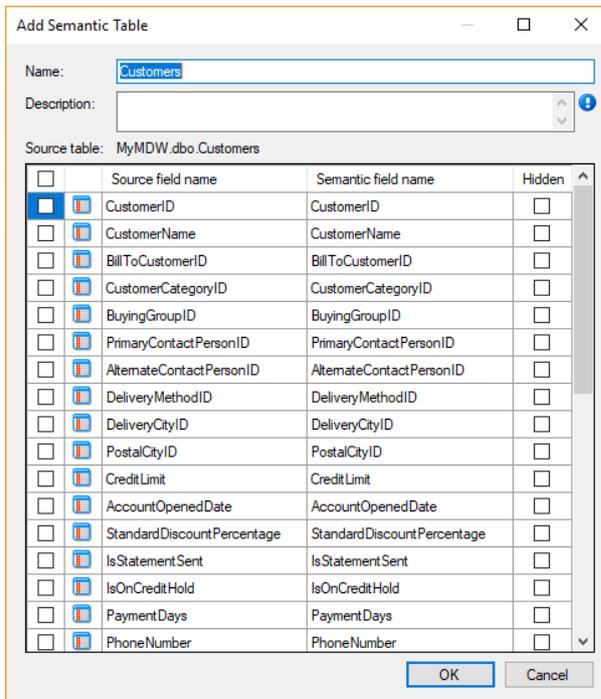
will not be deployed or executed in that specific environment. You can use this setting to have different active endpoints for e.g. your production, test and development environments.

Tables

Adding a Table to a Model

To add a table to a semantic model, follow the steps below.

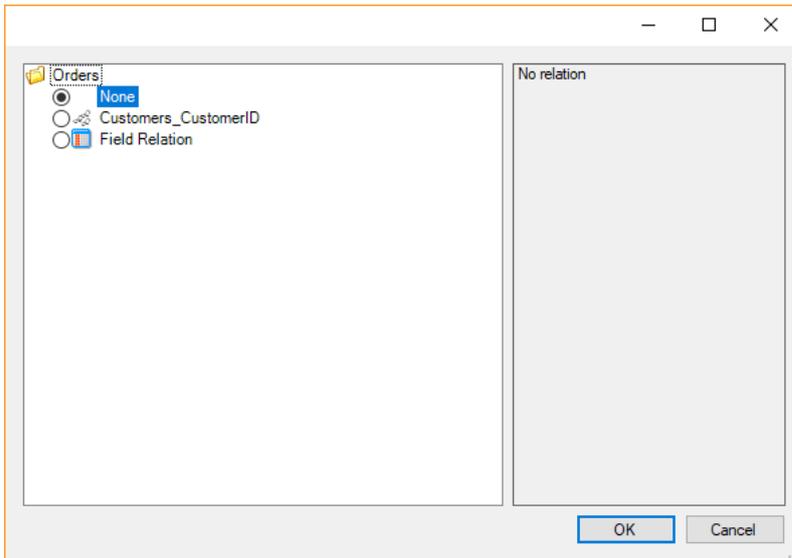
1. Drag a table from a data warehouse or business unit to **Tables** under the relevant model. The **Add Semantic Table** window appears.



2. If you would like the table to have a different name in the semantic model, type the new name in the **Name** box.
3. In the **Description** box, type a description of the model to use in front-end tools.

Note: This setting applies to SSAS Tabular endpoints only.

4. Select the fields you want in the semantic table.
5. (Optional) In the **Hidden** column, select the fields you want to add to the semantic table, but not show in the endpoint.
6. Click **OK** to add the table.
7. If there are other tables on the model that have an existing relation to the new table or a field with the same name as a field on the new table, the **Set Up Relations** window appears.



Here, you can set how the table you are adding is related to existing tables in the semantic model. For each existing table, you have the following options:

- **None:** No relations to that table model.
- An existing relation defined in the data warehouse (recommended).
- **Field relation:** Relate using identical field names on both tables.

8. Click **OK** to add the relations (if any).

Adding a Data Selection Rule to a Table

Data selection rules are used to specify a set of conditions that data extracted from a source table must satisfy. By applying selection rules, only the subset of data that you actually need is loaded into the semantic table.

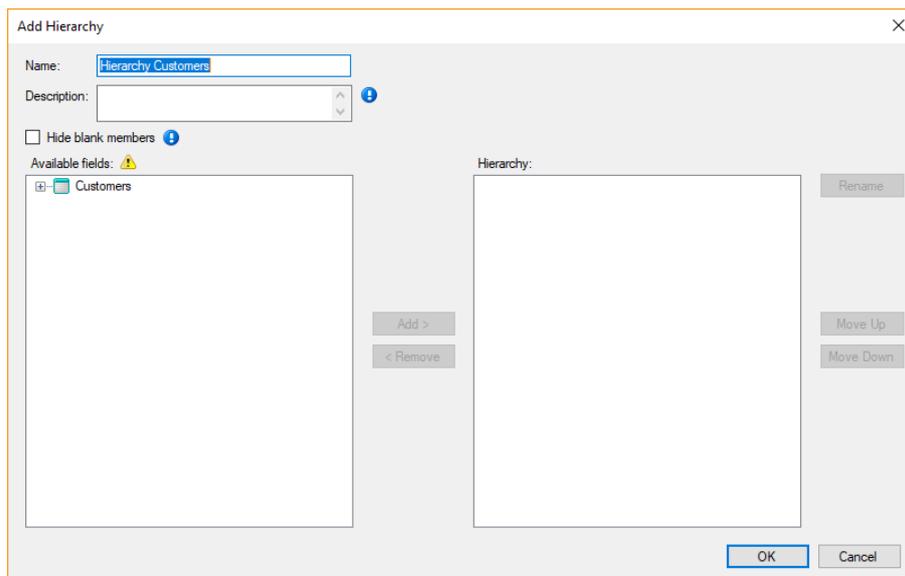
See [Data Selection Rules](#) for more information about adding data selection rules.

Adding a Hierarchy to a Table

Adding hierarchies makes it easier to browse data in front-end applications.

To add a hierarchy to a table, follow the steps below.

1. Right click the table and click **Add Hierarchy**. The **Add Hierarchy** window appears.



2. Type a name for the hierarchy in the **Name** box.
3. In the **Description** box, type a description of the hierarchy to use in front-end tools.

Note: This setting applies to SSAS Tabular endpoints only.

4. Select **Hide blank members** to hide blank members in the hierarchy caused by ragged hierarchies. For example, in a country-state-city hierarchy, some cities, e.g. Washington DC, doesn't actually belong to a state. These cities will then have a blank member above them in the hierarchy.

Note: This setting applies to SSAS Tabular endpoints only.

5. Click a field in the **Available fields** list and click **Add >** or double-click a field in the **Available fields** list to add it to the hierarchy.
6. Click a field in the **Hierarchy** list and click **Move Up** or **Move Down** to reorder the field.
7. (Optional) Click a field in the **Hierarchy** list and click **Rename** or press **F2** to rename the field.
8. Click **OK** to add the hierarchy. In the tree, you can find it in a Hierarchies folder under the table.

Setting a Default Date Table

Note: This setting applies to SSAS Tabular endpoints only.

Knowing what table is the default, or primary, date table enables additional features in Analysis Services Tabular client applications such as PowerBI.

Any table can be set as the default date table, including date tables added in the data warehouse and tables from a data source. To work, the table needs a field of the data type 'dat-

etime' to be used as key. On date tables created by TimeXtender, this would be the 'date-value' field.

To set a table as the default date table

- Right click the table and click **Set as Default Date Table**.

To remove a table as the default date table

- Right click the table and click **Remove as Default Date Table**.

6. In the **Category** list, click on the category you want the field to have.

Note: This setting applies to SSAS Tabular endpoints only.

7. In the **Summarize by** list, click on the type of aggregation you want to use for the field in e.g. Power BI.

Note: This setting applies to SSAS Tabular endpoints only.

8. In the **Sort by** list, click on the field that contains the values you want to sort by.

Note: This setting applies to SSAS Tabular endpoints only.

9. In the **Script** box, enter the script that generates the value of the custom field. Since syntax differs between endpoint types, you can add a script for each endpoint type. Click on an endpoint type in the **Endpoints** list to switch between endpoint types. Any endpoint type that does not have a specific script will use the Default script. You can drag in fields from the **Available parameters** list to use them as parameters in the script. Click **Show translation** to show the script TimeXtender will deploy to the endpoint.

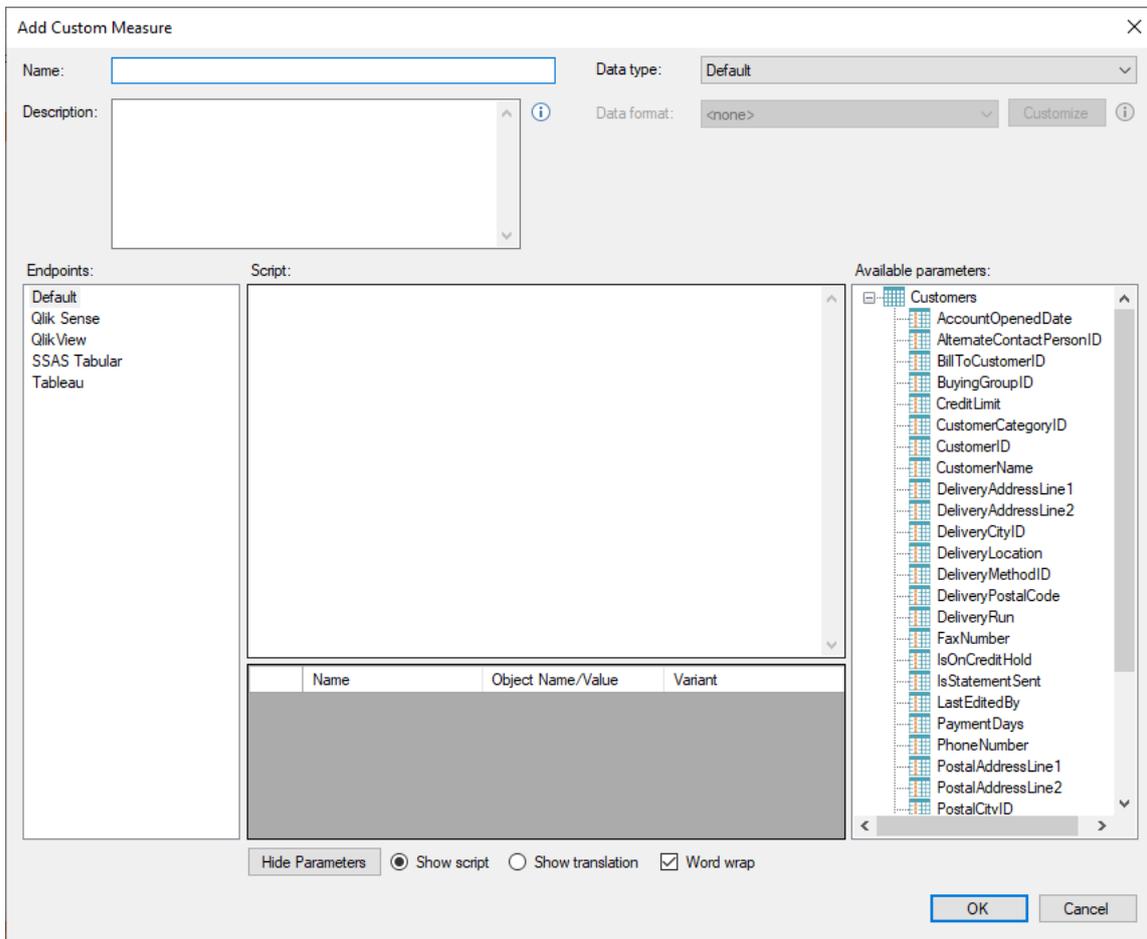
Adding a Custom Measure

Note: This setting applies to SSAS Tabular and Tableau endpoints only.

Custom measures use a script to calculate the value of the measure.

To add a custom measure, follow the steps below.

1. Right click a table and click **Add Custom Measure**. The **Add Custom Measure** window appears.



2. In the **Name** box, type a name for the field.
3. In the **Description** box, type a description of the custom measure to use in front-end tools.

Note: This setting applies to SSAS Tabular endpoints only.

4. In the **Data type** list, click on the data type you want to use for the custom field.
5. In the **Data format** list, click on the data format you want the field to have. For some data types, just one data format is available which is set and cannot be changed. Click **Customize** to customize the data format, e.g. number of decimal places for decimal numbers.

Note: This setting applies to SSAS Tabular endpoints only.

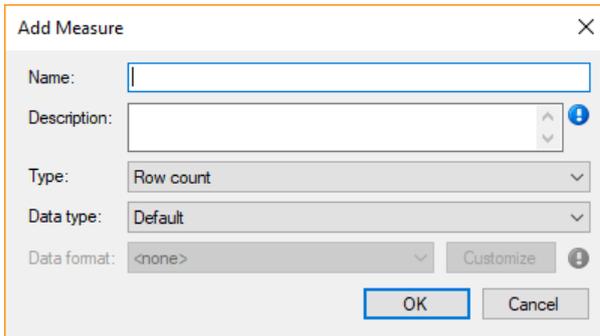
6. In the **Script** box, enter the script that generates the value of the custom measure. Since syntax differs between endpoint types, you can add a script for each endpoint type. Click on an endpoint type in the **Endpoints** list to switch between endpoint types. Any endpoint type that does not have a specific script will use the Default script. You can drag in fields from the **Available parameters** list to use them as parameters in the script.

Adding a Measure based on a Table

You can add measures to a table based on both the table itself and fields on the table.

To add a measure based on a table, follow the steps below.

1. Right click a table and click **Add Measure** The **Add Measure** window appears.



2. In the **Name** box, type a name for the measure.
3. In the **Description** box, type a description of the measure to use in front-end tools.

Note: This setting applies to SSAS Tabular endpoints only.

4. In the **Type** list, click on the type of measure you want to create. You have the following options:
 - **Row Count:** The value will be the number of rows in the table.
5. In the **Data type** list, click on the data type you want to use for the custom field.

Note: This setting applies to SSAS Tabular and Tableau endpoints only.

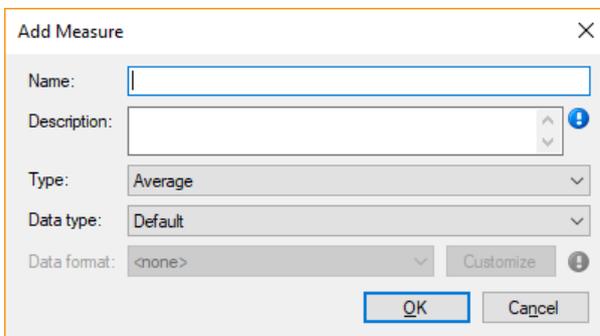
6. In the **Data format** list, click on the data format you want the field to have. For some data types, just one data format is available which is set and cannot be changed. Click **Customize** to customize the data format, e.g. number of decimal places for decimal numbers.

Note: This setting applies to SSAS Tabular endpoints only.

Adding a Measure based on a Field

To add a measure based on a field, follow the steps below.

1. Right click a field and click **Add Measure**. The **Add Measure** window appears.



2. In the **Name** box, type a name for the measure.

3. In the **Description** box, type a description of the measure to use in front-end tools.

Note: This setting applies to SSAS Tabular endpoints only.

4. In the **Type** list, click on the type of measure you want to create. You have the following options:

- **Average:** An average on the field values.
- **Count:** The number of field values.
- **Distinct count:** The number of unique field values.
- **Maximum:** The highest field value.
- **Minimum:** The lowest field value.
- **Sum:** The sum of the field values.

5. In the **Data type** list, click on the data type you want to use for the custom field.

Note: This setting applies to SSAS Tabular and Tableau endpoints only.

6. In the **Data format** list, click on the data format you want the field to have. For some data types, just one data format is available which is set and cannot be changed. Click **Customize** to customize the data format, e.g. number of decimal places for decimal numbers.

Note: This setting applies to SSAS Tabular endpoints only.

Setting Data Format and Category

Note: This setting applies to SSAS Tabular endpoints only.

By setting data format and category, you can control how a field's data will be displayed in client applications. For instance, text fields categorized as "Web URL" will be displayed as links in PowerBI.

To set data format and category for a field, follow the steps below.

1. Right click the field you want to sort and click **Edit Field**. The Edit Field window appears.

The screenshot shows the 'Edit Semantic Field' dialog box with the following settings:

- Name: CustomerID
- Description: (empty)
- Source field: CustomerID
- Data type: int
- Data format: <none>
- Category: Uncategorized
- Summarize by: Default
- Sort by: <none>

2. In the **Data format** list, click on the data format you want the field to have. The options depend on the data type which can be changed on the source field in the data warehouse or staging database. For some data types, just one data format is available which is set and cannot be changed.
3. Click **Customize** to customize the data format, e.g. number of decimal places for decimal numbers.
4. In the **Category** list, click on the category you want the field to have.

Choosing How a Field is Summarized

Note: This setting applies to SSAS Tabular endpoints only.

In analysis and reporting applications, a field can be aggregated in a number of different ways such as 'sum', 'average' or 'minimum'. The **Summarize by** option on fields in the semantic layer controls how Power BI aggregates the field.

To change the summarize by option

- Right click the field you want to set the option for, click **Edit Field** and, in the **Summarize by** list, click on setting you want to use.

Sorting a Field by another Field

Note: This setting applies to SSAS Tabular endpoints only.

Some fields have a certain conventional sort order. For instance, month names are usually ordered January - December, not alphabetically April - September. In this case, it would make sense to order the months according to a month number instead of the month name.

In TimeXtender you can set a field to be sorted by another field.

To sort one field by another field

- Right click the field you want to sort, click **Edit Field** and, in the **Sort by** list, click on the field that contains the values you want to sort by.

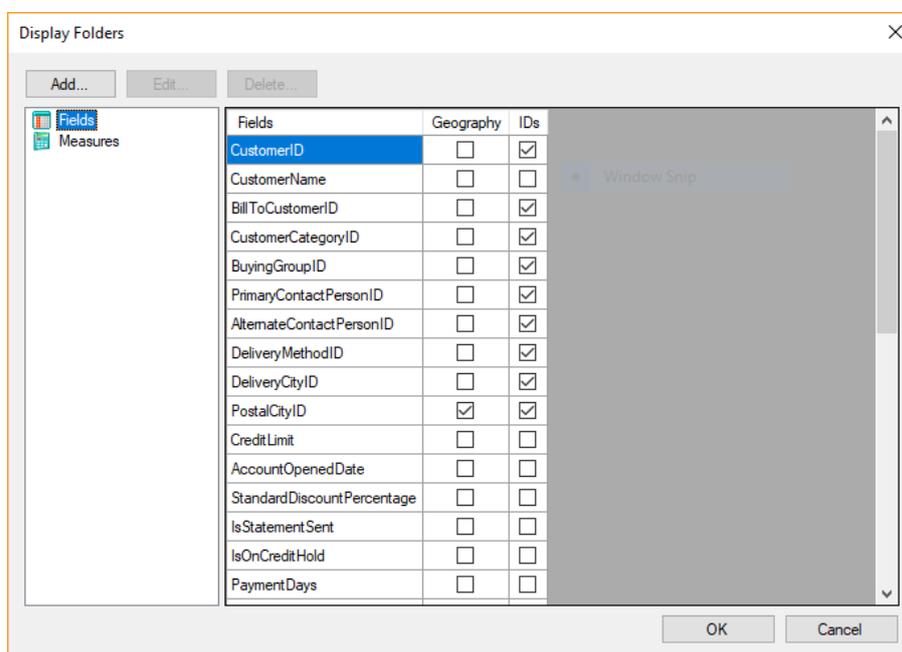
Organizing fields and Measures in Display Folders

Note: This setting applies to SSAS Tabular endpoints only.

With display folders, you can organize fields and measures in folders in client applications such as PowerBI.

To add a display folder and add fields to it, follow the steps below.

1. Right click the table that contains the fields and click **Display Folders**. The **Display Folders** window appears.



2. Click **Add** and type a name for the folder in the **Name** box in the window that appears. The syntax for display folders has two special characters:

- **Forward slash:** Use forward slash to create a hierarchy of display folders. For example, "A/B" will create a display folder "A" that contains a display folder "B".
- **Semicolon:** Behind the scenes, TimeXtender creates one display folder string for each field where each folder is separated by a semicolon. It is possible, but not recommended, to create display folders with semicolon in the name.

For example, mapping a display folder called "A;B" to a field is the same as mapping the field to a display folder "A" and a display folder "B".

Display folders are shared across the model.

3. Map the fields to display folders by clicking the check box where a field and a display folder intersect in the grid.

Hiding a Field or a Measure

Field and measures can be hidden to let the front-end application know that this object should no be displayed. This is very useful if you have intermediate fields or measures.

To hide a field or measure

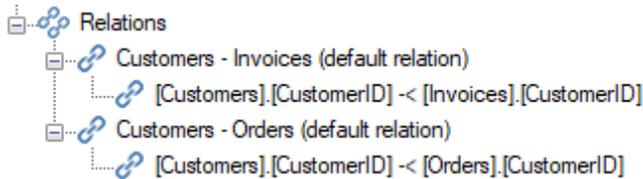
- Right click the field or measure and click **Hide**.

Note: Hiding measures only applies to SSAS Tabular and Tableau endpoints.

Relations

All relations for each table are listed under Relations under the table. This means that the same relation is listed under both the tables involved.

Each relation has a default name that consist of the two table names with a "-" between, but it can be renamed. A relation contains one or more relation items, i.e. relations between two fields.



Adding Relations

When adding relations, the available settings depend on the endpoint types you have selected for the model - see [Adding a Semantic Model](#).

To add a new relation on a model where Qlik is the only endpoint type enabled

- Drag a field from one table on the model to field on another table and the click **Yes** when asked if you want to add a relation.

To add a new relation on a model where the endpoint type SSAS Tabular or Tableau is enabled, follow the steps below.

1. Drag a field from one table on the model to field on another table. If the cardinality of the tables involved is not one-to-one, drag from the "one" table to the "many". A window appears with settings.
2. In the **Cardinality** list, click on the option the represents the cardinality of the table relationship.
3. In the **Filter direction** list, click on option you prefer:
 - **To [table]:** [Table] is filtered by the other table in the relationship.
 - **To both tables:** The tables filter each other.

Note: This setting applies to SSAS Tabular endpoints only.

To add a new relation item to an existing relation

- Drag a field from one table on the model to field on another table and click on the existing relation in the menu that appears.

Changing the Default Relation

You can have multiple relations between two tables. The first one will be designated as the default relation, which are necessary in some endpoints.

To set a relation as the default relation

- Right click the relation and click **Set as default relation**.

Changing a Relation's Cardinality

Note: This setting applies to SSAS Tabular and Tableau endpoints only.

Each relation has cardinality. In the tree, the cardinality can be identified on the relation items as follows:

- --: One to one
- -<: One to Many
- >: Many to one

To change the cardinality for a relation

- Right click the relation, click **Cardinality** and click on the cardinality you want for the relation.

Changing a Relation's Filter Direction

Note: This setting applies to SSAS Tabular endpoints only.

Each relation also has a filter direction. To change the filter direction

- Right click on the relation, click on **Filter Direction** and click on the filter direction you want to set for the table.

Security

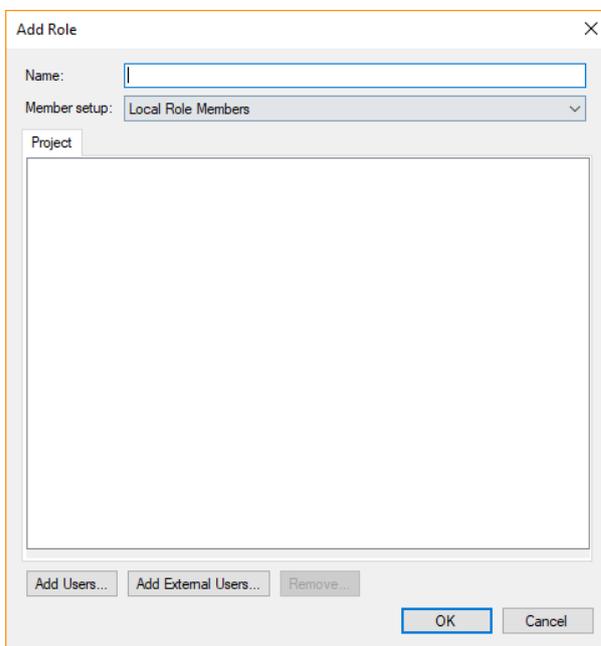
In the semantic layer, you can setup access to data on the row and model level. In other words, you can decide who has access to a model and what data in the model they have access to. For instance, a sales team could have access to a model while the individual sales people has access to data on the specific customers they work with.

To setup access on the model level, you add a role and map it to an endpoint. To refine the access to the row level, you add a row-level security setup and map this to one or more roles.

Adding a Role

To add a role, follow the steps below.

1. Navigate to the relevant model, right click Roles and click **Add Role**. The **Add Role** window appears.



2. Type a name for the role in the **Name** box.
3. Click **Add AD Users...** to add users from a local AD. The standard **Select Users and Groups** window appears.
4. Click **Add External Users...** to add an external user, e.g. an Azure AD user. The Add External Users window appears. Type the users email address and click **Add** to add him to the role.
5. Click **OK** to add the role.

Mapping a Role to an Endpoint

Mapping a role to an endpoint restricts access to data on that endpoint to members of the role.

To map a role to an endpoint

- Right-click the role, click **Endpoints** and click the endpoint you want to map the role to
- OR -
Drag the role to the endpoint

Adding a Row-Level Security Setup

Note: This setting applies to SSAS Tabular and Qlik endpoints only.

There are different ways of setting up row-level security depending on the endpoint(s) you are targeting and how security is handled in your organization. See [How to Setup Up Row-level Security](#) below for more information.

Two types of row-level security setups are available: Manual, where you map values and members in the GUI and dynamic, where the mappings are read from a table in the data warehouse.

To add a manual row-level security setup, follow the steps below.

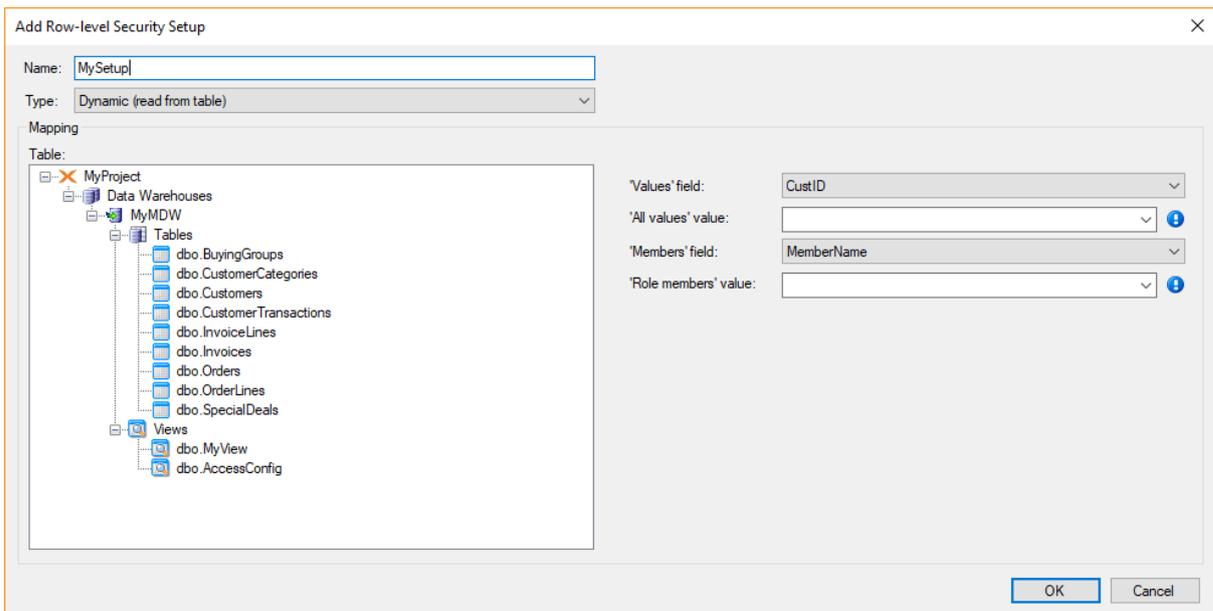
1. Right click a field and click **Add Row-level Security Setup**.

2. Type a name for the setup in the **Name** box.
3. Select one or more row values in the **Values** list and one or more members in the **Members** list and click **Add >** to map values and members. The "(Role members)" member maps the values to the members of the roles that are mapped to the setup.
4. Enter a username in the text box and click **Add Member** to add that user or group to the list of members. Usernames added this way will be deployed with the roles that are mapped to the setup.

Note: Members added this way are deployed to Qlik endpoints only.

To add a dynamic row-level security setup, follow the steps below.

1. Right click a field and click **Add Row-level Security Setup**.
2. In the **Type** list, click **Dynamic**.



3. In the **Table** list, click the table or view that contains the mapping you want to read.
4. In the **'Values'** field list, click the column that contains the values, e.g. customer IDs.
5. In the **'All values' value** box, click or type the value that TimeXtender should take to mean all values, i.e. members mapped to this value has access to every row.
6. In the **'Members'** field list, click the column that contains the members.
7. In the **'Role members' value** box, click or type the value TimeXtender should interpret as any member of a role mapped to the row-level security setup.
8. Click **OK** to save the setup.

Mapping a Row-level Security Setup to a Role

Mapping a row-level security setup to a role restricts access to data on that endpoint according to the mapping of row values and members in the setup.

To map a row-level setup to a role

- Right-click the row-level security setup, click **Roles** and click the role you want to map the setup to
- OR -
- Drag the row-level security setup to the role

Setting Up Row-level Security

You can add as many or as few roles and row-level security setups as you like and each role can be mapped to any number of row-level security setups and the other way around. This gives you a lot of flexibility to set up row-level security in a way that makes sense in your particular situation.

There are two basic approaches you can use:

- **One role and one setup:**
 - Add one role. If you target SSAS Tabular, all users and groups that you later add to the security setup must be members to have access. If you only target Qlik, the role can be empty since it only serves as a link between the security setup and the endpoint.
 - Add one setup. Add the users and groups as members in the setup and map the relevant values to the members you have added.
 - Map the setup to the role and the role to the endpoint(s).
 - This works well if you have the relevant groups in Active Directory and can manage membership from there.
- **Many roles and setups:**
 - Add a role for each user or group that should have access to a specific subset of data.
 - Add a setup for each of the roles. Map the relevant values to the "(Role Member)".
 - Map the setups to the roles and the roles to the endpoint(s).
 - This works well if you want to use roles as groups in the semantic layer.

As hinted above, Analysis Services and Qlik handles security differently:

- On Analysis Services, access is granted to a role and TimeXtender uses DAX scripting to give users and groups access on the row level.
- Qlik does not have roles, so all access is granted on the users/groups-level. For Qlik, roles in the semantic layer is simply an ad hoc collection of users and groups that have access to the same data.

Perspectives

Note: Perspectives apply to SSAS Tabular endpoints only.

Semantic models can contain a lot of tables, fields and measures. That can make them very complex for users of analysis and visualization tools, who might only need to interact with a small part of the model to get the data they require. Perspectives are a subset of the objects in the model that users of PowerBI and other front-end tools can apply to make it easier to work with the model.

Adding a Perspective

To add a perspective, follow the steps below.

1. Right click a semantic model and click **Manage Perspectives** .
2. Click **Add**. A new column is added to the grid with the name “Perspective”. To edit the name, click the name and click **Rename** or double-click the name. Click outside the field when you have finished typing.
3. Select the tables and/or fields you want to include in your perspectives using the check boxes. All tables and fields in the model are listed in the first column of the grid. Click the **+/-**-next to a table to show/hide fields.

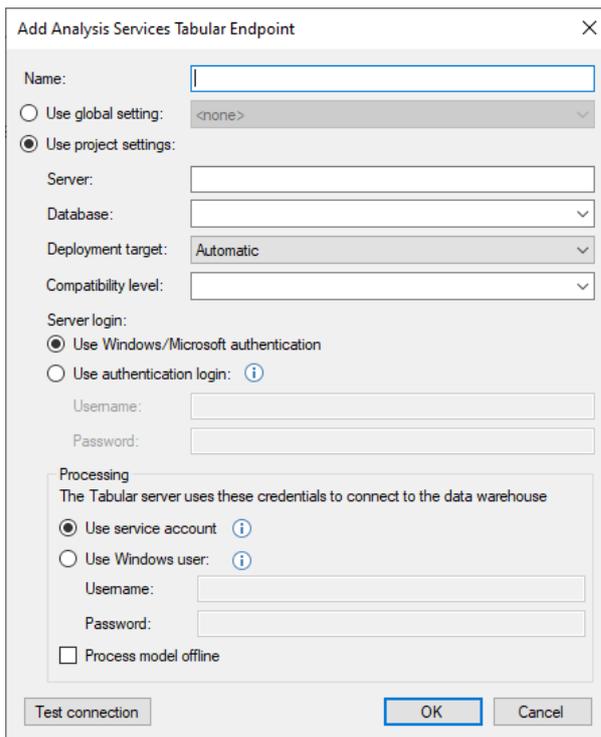
Endpoints

Your semantic models can have a number of endpoints. At the time of writing, TimeXtender supports three different endpoints: Qlik, Tableau and Analysis Services Tabular.

Adding a SSAS Tabular Endpoint

To add a Tabular Endpoint, follow the steps below.

1. Expand the model you want to add an endpoint to, right click Endpoints and click **Add Tabular Endpoint**. The **Add Tabular Endpoint** window appears.



2. In the **Name** box, enter a name for the endpoint.
3. In the **Server** box, type the name of the Tabular server. The server can be on-premise or in Azure.
4. In the **Database** box, type the name of the database.
5. In the **Deployment target** list, click the version of SSAS you are targeting. **Automatic** (default) or **Analysis Services Universal** are the recommended settings.

Note: Analysis Services 2016 and later versions are supported.

6. In the **Compatibility** box, click or type the compatibility level you want to use, or leave it blank to use the highest supported by the server.
7. Under **Server login**, the following settings are available:
 - **Use Windows/Microsoft authentication:** Uses the credentials of the user executing the endpoint.
 - **Use authentication login:** Uses the credentials of a specific user or an App Registration (if the SSAS server is on Azure). This can be useful when you SSAS

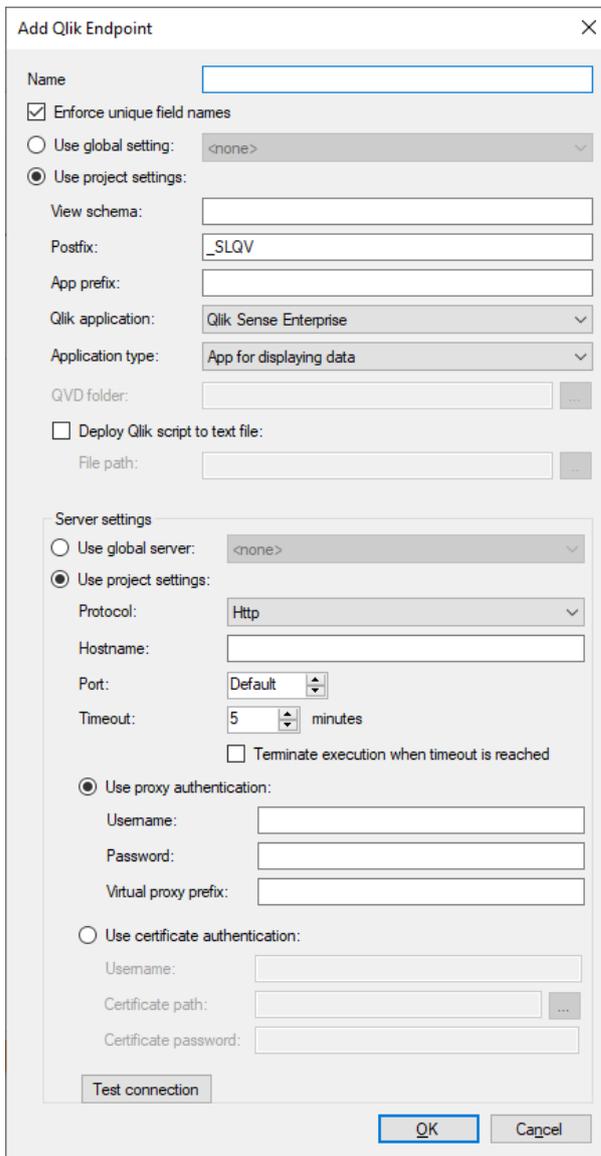
server is on Azure and the user executing the endpoint has two-factor authentication enabled, which will be triggered by an execution. Enter the user name for the user in the **Username** box and the corresponding password in the **Password** box. Prefix the username with 'app:' if it is an App Registration application key.

8. Under **Processing**, you can set how the Tabular service will connect to the data warehouse.
 - For authentication, the default is to use the SQL Server Analysis Services service account. Click **Use Windows user** to use another user and then enter the user name for the user in the **Username** box and the corresponding password in the **Password** box.
 - Select **Process model offline** to process the model "behind the scenes" and make the deployment seamless for the users. The offline database will have the endpoints name prefixed with "Offline_".
9. Click **OK** to add the endpoint.

Adding a Qlik Endpoint

To add a Qlik endpoint, follow the steps below.

1. Expand the model you want to add an endpoint to, right click Endpoints and click **Add Qlik Endpoint**. The **Add Qlik Endpoint** window appears.



2. In the **Name** box, enter a name for the endpoint.
3. (Optional) Clear the **Enforce unique field names** checkbox if you don't want TimeXtender to ensure that field names are always unique across tables by prefixing the name with the table name on deployment.
4. (Optional) In the **View schema** box, type the schema name you want to use for the views generated by TimeXtender.
5. (Optional) In **Postfix** box, type the postfix TimeXtender uses for views, folder names etc.
6. (Optional) In the **App prefix** box, type a string to be prefixed to the endpoint name to create the app name used in Qlik Sense.
7. In the **Qlik application** list, click on the Qlik application to target. You have the following options:
 - **Qlik Sense Enterprise**: Use a Qlik Sense Enterprise server. When you chose this application type, you need to enter server connection information under server settings.

- **Qlik Sense Desktop**
- **QlikView**

Note: For more information on deploying to the different applications, see [Qlik Endpoint Deployment](#) below.

- If you are deploying to Qlik Sense Enterprise or Qlik Sense Desktop, click on the app type you want to create in the **Qlik Sense app type** list. You have the following options:
 - **App for generating QVD file:** Creates an app that generates a QVD file with data from the model in the **QVD folder** you specify. The **QVD folder** should be accessible for both TimeXtender and Qlik Enterprise.
 - **App for displaying data:** Creates an app and loads data from the model into it.
- Select **Deploy Qlik script to text file** and enter a path in **File path** to have TimeXtender output the script it generates to a text file.
- If you are deploying to Qlik Sense Enterprise, enter settings under **Server:**
 - Type the your server's hostname in the **Hostname** box.
 - Type the port to connect to in the **Port** box if it is different from the default. The defaults are 4747 if you use certificate authentication, 80 if you use proxy authentication with HTTP and 443 if you use proxy authentication with HTTPS.
 - (Optional) In the **Timeout** box, enter the timeout you want to use in communication with the server.
 - Select **Terminate execution when timeout is reached** if you want TimeXtender to terminate - kill - an execution when the timeout is reached. This is useful in rare cases where executions will not terminate by themselves.
 - Click **Use proxy authentication** if you are using the proxy authentication method to authenticate with the Qlik Sense Enterprise server. Type your username in the **Username** box and your password in the **Password** box. Write the prefix from the virtual proxy in Qlik Sense in the **Virtual proxy prefix** box.
 - Click **Use certificate authentication** if you are using the certificate authentication method for authenticating with Qlik. Type your username in the **Username** box, enter the path to the certificate in the **Certificate path** box and the associated password in the **Certificate password** box.
- Click **OK** to add the endpoint.

Adding a Tableau Endpoint

To add a Tableau endpoint, follow the steps below.

- Expand the model you want to add an endpoint to, right click Endpoints and click **Add Tableau Endpoint**. The **Add Tableau Endpoint** window appears.

The screenshot shows a dialog box titled "Add Tableau Endpoint" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Name:** A text input field.
- Use global setting:** A radio button that is currently unselected, followed by a dropdown menu showing "<none>".
- Use project settings:** A radio button that is currently selected.
- File:** A text input field with a browse button (...).
- Schema:** A text input field.
- Extension:** A text input field.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

2. In the **Name** box, enter a name for the endpoint.
3. In the **File** box, enter the path and file name for the Tableau data source file generated by TimeXtender.
4. (Optional) In the **Schema** box, type the schema name you want to use for the views generated by TimeXtender.
5. (Optional) In the **Extension** box, type the postfix TimeXtender uses for views etc.
6. Click **OK** to add the endpoint.

Deployment and Execution

Deploying and executing a semantic model means deploying and executing the endpoints on the model. All endpoints can be deployed, but not all endpoints need to be executed. Your options will vary accordingly.

To deploy a model or endpoint

- Right click the model or endpoint, click **Deploy**, **Execute** or **Deploy and Execute** and click **Start** in the **Deploy and/or Execute** window that appears.

What happens during deployment and execution depends on the endpoint.

SSAS Tabular Endpoint Deployment

On deployment, the model is created on the SSAS Tabular server. To get data in the model, you need to execute the model as well.

Qlik Endpoint Deployment

For Qlik endpoints, the end product is a QVD file for each table in the model. QVD is a proprietary data format that stores data in the way that gives the best performance in Qlik apps. Since only Qlik applications can create QVD files, deployment and execution of Qlik endpoints create apps or scripts that a Qlik application can use to create QVD files.

Data for the QVD files is extracted from views. On deployment, a view for each table in the model is created in the data warehouse or staging database that house the table. The view name depends on the settings on the endpoint and has the format [view schema].[table name]_[postfix], e.g. "QView.Customers_QV".

Apart from creating the views, deployment is different depending on your choice of Qlik application:

- **Qlik Sense Enterprise:** An app called "[Endpoint name]_QVDApp" is created on the server. Unlike the other Qlik applications, Qlik Sense Enterprise has an execution step. On execution, the app on the server is executed and creates QVD files on the file path specified.
- **Qlik Sense Desktop:** You can right click the endpoint and click **Create Qlik Sense App** to create an app in the application. When you execute this app in Qlik Sense Desktop, it creates QVD files based on the tables in the semantic model on the file path specified.
- **QlikView:** You can right-click the endpoint and click **QlikView Scripts** to show and copy the script you need to use in QlikView to generate QVD files based on the tables in the semantic model.

Tableau Endpoint Deployment

On deployment, a view for each table in the model is created in the data warehouse or staging database that house the table. The view name depends on the settings on the endpoint and has the format [view schema].[model name]_[endpoint name]_[table name]_[postfix], e.g. "Tableau.MyModel_MyTableau_Customers_tab".

In addition to that, a TDS file is created on the file path specified in the endpoint. Use this file in Tableau to connect to the views.

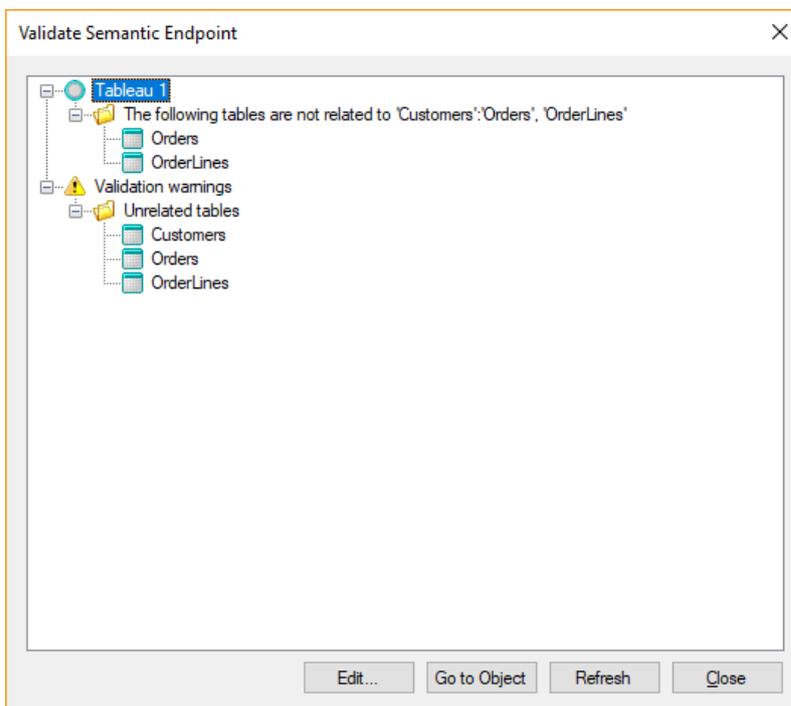
Validating a Semantic Model or Endpoint

You can run a validation on a semantic model or endpoint to catch issues that would cause problems in the front-end after deployment and execution.

To validate a semantic model or endpoint, follow the steps below

1. Right click a model and click **Validate Model**
- OR -
Right click an endpoint and click **Validate Endpoint**.

If the validation results in warnings, the **Validate Semantic Model** or **Validate Semantic Endpoint** window appears.



2. To help you fix the warnings, the window contains some shortcuts for each item in the list:
 1. Click **Edit...** to edit the selected object
 2. Click **Go to Object** to reveal the selected object in the tree.
 3. Right click an object and click **Delete** if you want to delete the object.

Building SSAS Multidimensional Cubes

TimeXtender includes support for designing SSAS Multidimensional cubes that end users can browse through Excel or specialized business intelligence front-ends. SSAS Multidimensional utilizes fact- and dimension tables you have created in your data warehouse.

Cubes allow you to present data in a multidimensional model. You can break down data in your data warehouse into smaller units, enabling you to drill-down, or roll-up through data, depending on the level of detail you want to view. You can, for example, create a sales cube, a production cube, a finance cube, and so on.

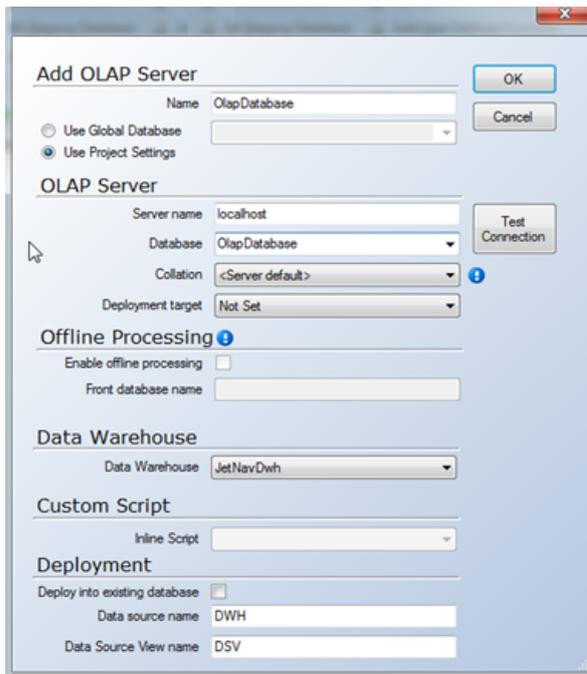
A cube consists of a number of dimensions and measures. The dimensions determine the structure of the cube, and the measures represent the numerical values. You can, furthermore, define hierarchies within a dimension by using dimension levels.

Dimensions define how a user looks at data in a cube. Dimensions are created independently of a particular cube and can be used within several cubes at the same time. The same dimension can also be utilized several times within the same cube, which is referred to as a role-playing dimension. A common example of this would be the Date dimension, which can represent both the Document Date and Posting Date in a cube, thus having a single dimension play two roles.

Adding an SSAS Multidimensional Server

To use SSAS Multidimensional, you will first need to add an SSAS Multidimensional Server.

1. In the **Solution Explorer**, right-click **Semantic Layer**, click **Add SSAS Multidimensional Server**, and then click **New SSAS Multidimensional server...**



2. In the **Name** field, type a name for the server. The name cannot exceed 15 characters in length.
3. In the **Server Name** box, type the name of the SSAS Multidimensional database server.
4. In the **Database** box, type a name for the database.
5. In the **Collation** list, click the database collation to use for the SSAS Multidimensional database. **<Server Default>** will inherit the default collation currently set in Analysis Services. **<Application Default>** will use Latin1_General_CI_AS. This collation should correspond with the collation that is set for SQL Analysis Services.
6. Select a data warehouse from the **Data Warehouse** list, and then click **OK**. Each SSAS Multidimensional database can pull from a single data warehouse database.

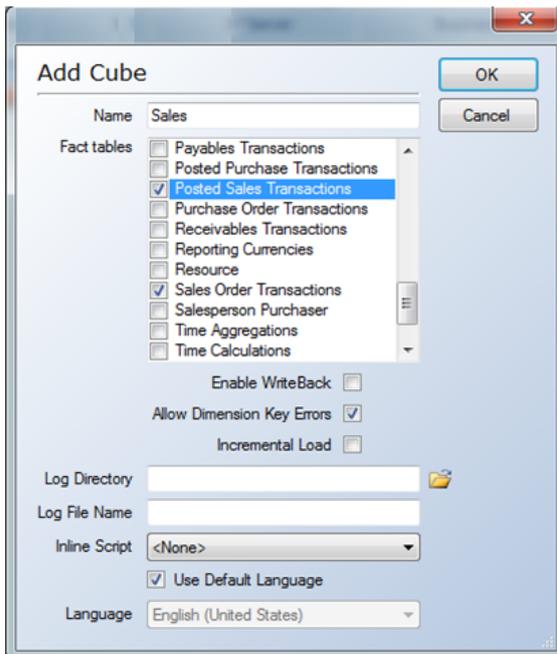
Cubes

Cubes are the cornerstone when presenting data through SSAS Multidimensional. Cubes are typically built around central functions in the company, such as sales, finance, inventory etc.

Adding SSAS Multidimensional Cubes

To add a cube, follow the steps below.

1. Open the server, right-click **Cubes** and click **Add Cube**. The **Add Cube** window appears.



2. In the **Name** field, type a name for the cube.
3. In the **Fact Table** list, select on the table(s) from the data warehouse you want to use for the cube.
4. If you want end users to be able to change cube data while they browse it, select **Enable Writeback**. Any changes the end users make are saved in the write-back table.

Note: You can only utilize write-back if the front-end application supports it.

5. If you want to continue processing the cube even if dimension key errors occur, select **Allow Dimension Key Errors**. When you allow dimension key errors, all errors are reported to a log and you need to specify where you want to store the log. Click the folder icon beside **Log Directory**. This opens a new window. Navigate to the folder you want to store the log in and click **OK**. In the **Log File Name** box, type a name for the log file.
6. Click **OK** to add the cube.

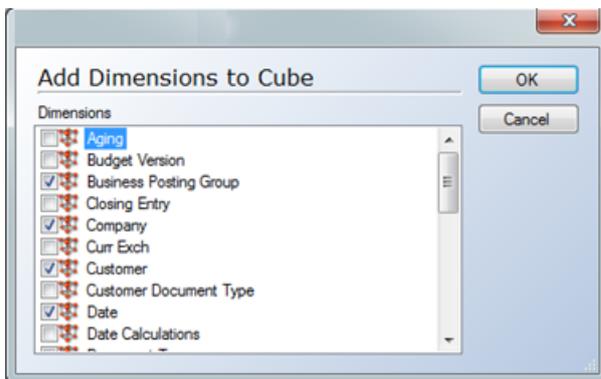
Adding a Single Dimension to a Cube

1. Expand **Dimensions** and drag-and-drop the dimension to the cube.
2. Set the relationship to the fact table in the cube. See [Adding Dimension Relationships](#).

Adding Multiple Dimensions to a Cube

You can also add multiple dimensions to a cube in one operation.

1. Open the server, expand to **Cubes**, and then expand the cube you want to add dimensions to.
2. Right-click **Dimensions**, and then select **Add Dimension to Cube**. The **Add Dimensions to Cube** window opens with all the dimensions in your project listed.



3. Select the dimension or dimensions you want to add, and click **OK**.

You must then set the relationship to the fact table in the cube. See [Adding Dimension Relationships](#).

Adding Role-Playing Dimensions

Role-playing dimensions are dimensions that are used more than once in the same cube. For example, you can use a Customer dimension more than once in the same cube.

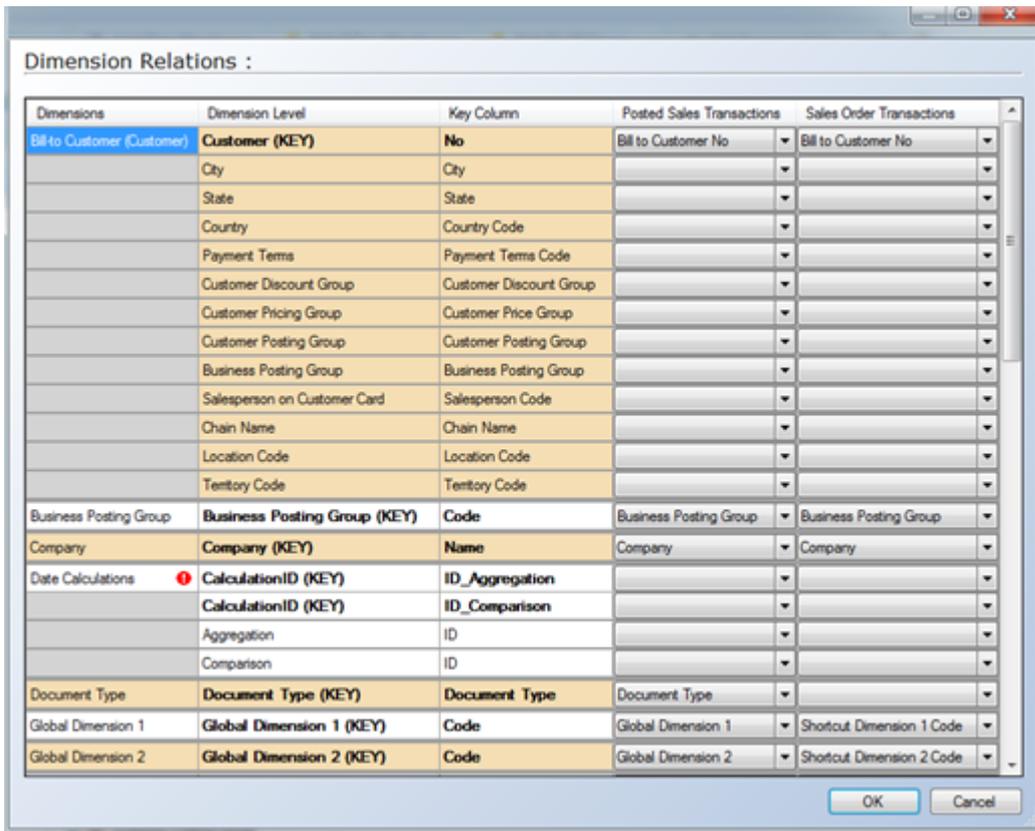
When you have added a role-playing dimension to a cube, you should follow the steps described above to add the dimension and then specify a new name for the dimension to distinguish it from the other dimensions of the same type. Then define the relationship to the proper field in the fact table.

For example, the Customer dimension may be used as the Bill-to Customer dimension, which relates to the Bill-to Customer No. field in the fact table, while the Sell-to Customer dimension may relate to the Sell-to Customer No. field in the same fact table.

Adding Dimension Relationships

Dimension relationships specify how a dimension is related to a fact table. You must define how each level in a dimension is related to a fact table.

1. Expand the server that contains the relevant cube. Expand the cube, right-click **Dimensions**, click **Dimension Relations** and click **All Fact Tables**.
2. The **Dimension Relations** window opens.



The table contains the following columns:

Column	Description
Dimension	Displays all dimensions in the cube
Dimension Level	Displays all dimension levels associated with each dimension
Key Column	Displays the key column for each dimension level
<Fact Table Name>	Displays the name of the fact table

3. In the Fact Table column(s), select the field where you will join the dimension key with the dimension. The dimension levels in **Bold** are the required values to be set for each dimension.
4. Click **OK** when you have created all of the necessary relationships.

Validating Cube and Dimensions

To validate a cube or a dimension

- Right-click the cube or dimension you want to validate, and then click **Validate Cube/Dimension**. If the cube or dimension is valid, an OK message is displayed. Click

OK to close the message dialog. If the cube or dimension is invalid, a message is displayed outlining the changes that you need to make.

The Cube Browser

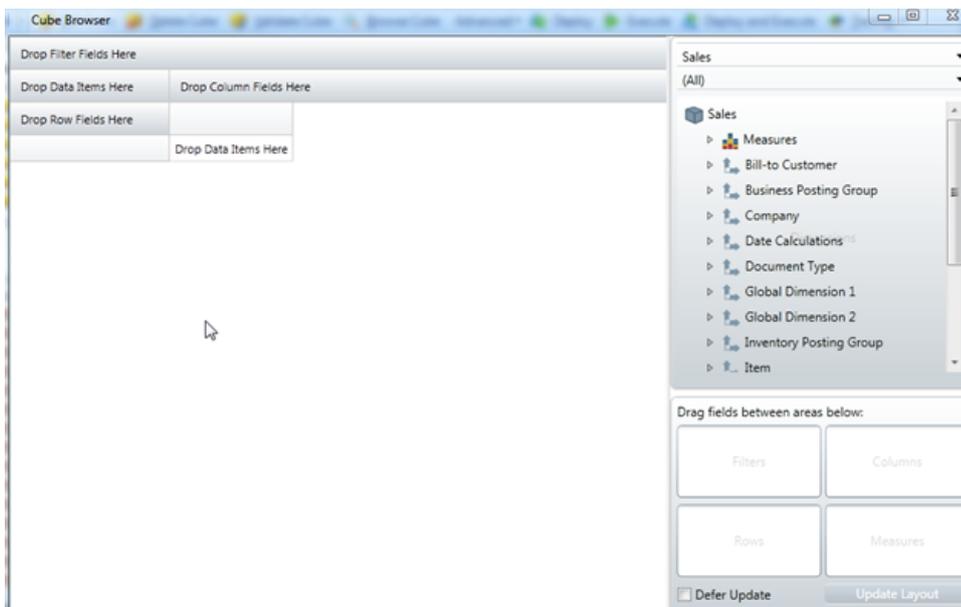
The Cube Browser allows a TimeXtender user to browse a cube from within TimeXtender without first leaving to go into another application such as Excel. The Cube Browser is not meant to replace a proper front-end tool, such as Excel, but is an easy way to browse the cube structure without having to navigate away from TimeXtender.

To open the Cube Browser

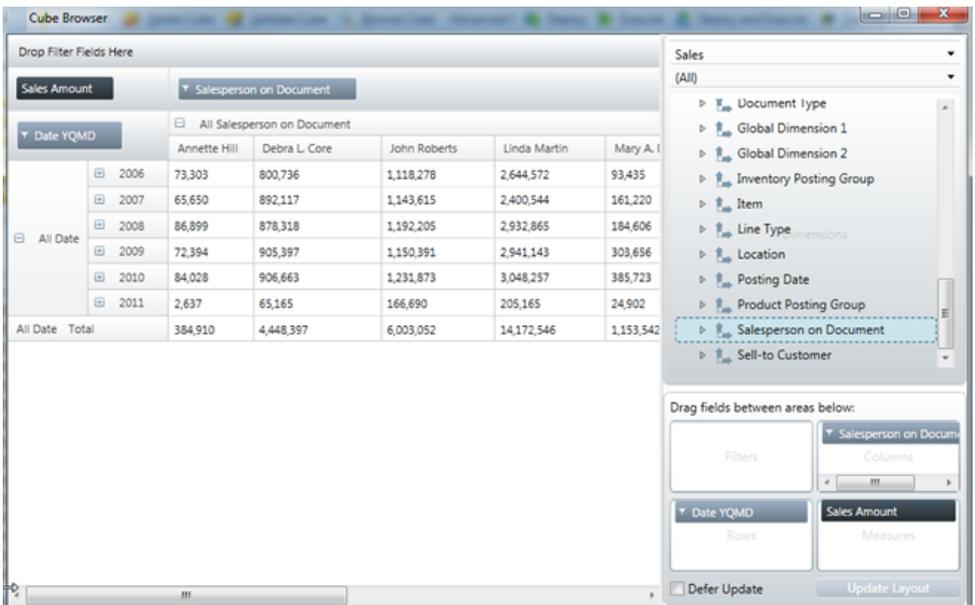
- Right-click the cube and click **Browse Cube**.

Using the Cube Browser

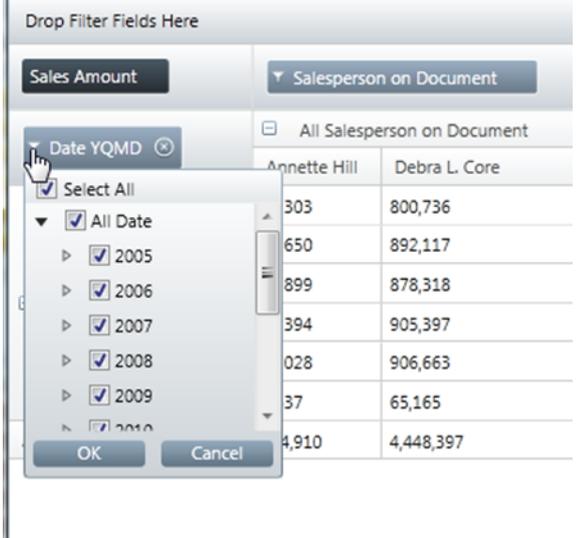
The user interface is similar to pivot tables in Excel. Measures and dimensions are dragged from the list on the right and dropped in either the boxes in the lower right-hand corner or directly onto the workspace pane on the left.



The following report was created from the Sales cube for NAV by dragging the **Date** dimension into the **Rows** box, the **Salesperson on Document** dimension into the **Columns** box, and the **Sales Amount** measure into the **Measures** box.



Filters for the rows and columns can be added by clicking the **Filter** icon to the left of the row and the column labels in the workspace pane on the left.



Dimensions can also be expanded and collapsed by clicking the plus and minus signs respectively.



Cube Writeback

Cube Writeback is a feature that allows users to update or add data to the cubes through the front-end. When Writeback is enabled for a cube, it will be enabled for all measure groups in that cube.

The structure of the fact tables will be maintained and updated when the structure of the cube changes.

It is important to understand how writeback works before implementing it. While this document does not target its full scope, it warrants a thorough understanding. Otherwise, users will find themselves performing actions that are incomplete.

To enable this feature

- Right-click the relevant cube, click **Edit Cube** and select **Enable WriteBack** in the window that appears.

When writeback has been enabled, a writeback table will be created in the data warehouse on the next deployment. Whenever anything is written to a measure, it will be added to the writeback table. Data will never be written directly to the original fact table.

For the best results, design a separate cube specifically targeted for cube writeback.

Offline Cube Processing

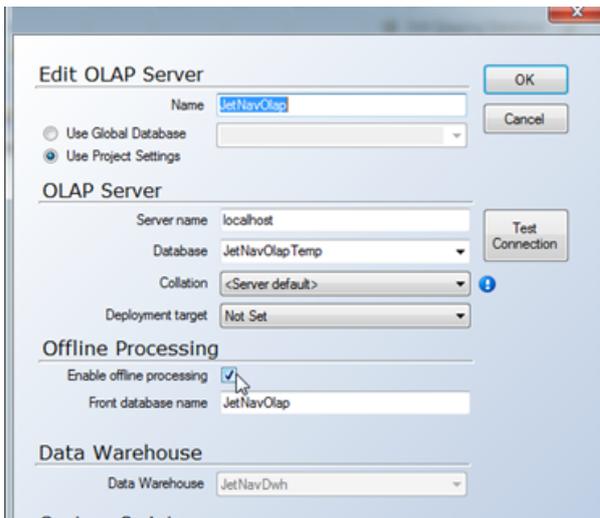
TimeXtender supports functionality that allows cubes to be processed without being taken offline. Normally, when a cube is processed and rebuilt in SQL Server Analysis Services, it is taken offline during the duration of the processing and is made unavailable to end-users.

SQL Server Enterprise Edition allows the cube to be kept online, but if you do not run this edition of SQL Server, the Offline Cube Processing feature of TimeXtender allows you achieve the same result.

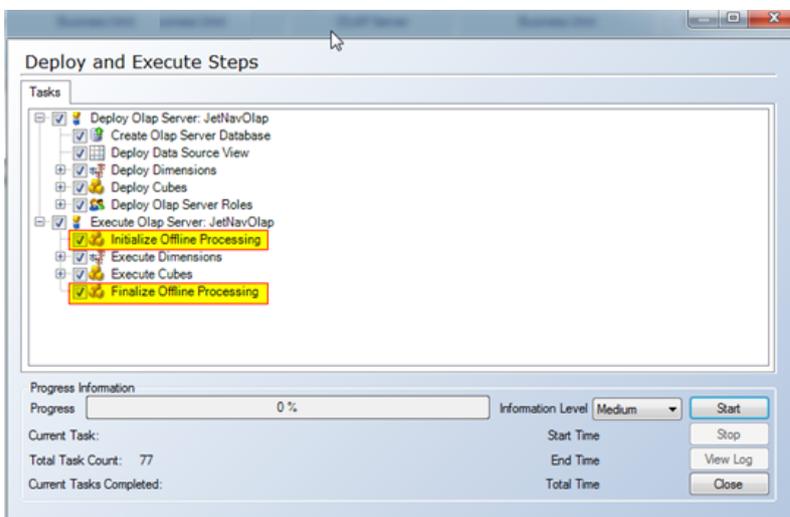
Enabling Offline Cube Processing means that the cubes can be updated throughout the business day without disturbing end-users. While the cube is being processed, the users will have access to the original version of the cube, which is replaced with the new version of the cube once processing has been completed.

Enabling High Availability Cube Processing

1. Right click the server and click **Edit SSAS Multidimensional Server**. The **Edit SSAS Multidimensional Server** window appears.



2. Rename the **Database** to represent a temporary SSAS Multidimensional database that will be used during processing. This is *not* the database that will be used by end-users.
3. Select **Enable Offline Processing**.
4. Type in the name of the database that will be used by end-users in **Front Database Name**.
5. Click **OK**.
6. Right-click the SSAS Multidimensional database and click **Deploy and Execute**.
7. You will notice two new steps under **Executed SSAS Multidimensional Server** that handles Offline Cube Processing. These are **Initialize Offline Processing** and **Finalize Offline Processing**.



8. Click the **Start** button to begin processing the cubes. Users will now be able to access the cubes as they are being processed.

Dimensions

Dimensions define how a user looks at data in a cube. Dimensions are created independently of a particular cube and can be used within several cubes at the same time. The same dimension can also be utilized several times within the same cube, which is referred to as a role-playing dimension. A common example of this would be the Date dimension, which can represent both the Document Date and Posting Date in a cube, thus having a single dimension play two roles.

Regular Dimensions

Regular dimensions are based on a snowflake or a star schema, and are used to create balanced or ragged hierarchies.

Creating Regular Dimensions

1. From the **Solution Explorer**, open the relevant server, right-click **Dimensions**, and then select **Add Dimension**. The Add Dimension window appears.



2. In the **Name** field, type a name for the dimension.
3. In the **Unknown Member** list, select **Visible** to apply an Unknown Member to dimension keys in the fact table with no matching dimension members. This will allow dimension values that exist in the fact table, but not the dimension table to be combined together and displayed to the user as an Unknown value. This is the default and recommended setting in TimeXtender. An example could be a Salesperson Code that exists in the sales transactions fact table but does not exist in the Salesperson dimension table.
4. Next to the **Type** box, click the ellipsis (...) and select the type of dimension you want to create. You can leave the **Type** blank for regular dimension types, which are the most common type.

Type	Description
Regular	Default for dimensions that are not set to a specified type
Time	Used for dimensions whose attributes represent time periods
Geography	Used for dimensions whose attributes represent geographical information
Organization	Used for dimensions whose attributes represent organizational information
BillOfMaterials	Used for dimensions whose attributes represent inventory and manufacturing information
Accounts	Used for dimensions whose attributes represent information used for financial reporting
Customers	Used for dimensions whose attributes represent information about customers
Products	Used for dimensions whose attributes represent information about products
Scenario	Used for dimensions whose attributes represent information about plans and strategies
Quantitative	Used for dimensions whose attributes represent quantitative information
Utility	Used for dimensions whose attributes represent utility information
Currency	Used for dimensions whose attributes represent currency information
Rates	Used for dimensions whose attributes represent currency rate information
Channel	Used for dimensions whose attributes represent channel information
Promotion	Used for dimensions whose attributes represent marketing promotion information

5. In the **All Member Name** box, type a name for the **All Member**. This is left blank by default, which means that Analysis Services creates the **All Member Name** automatically. The **All Member** is the dimension value which represents all members of the dimension. An example would be a dimension value of “All Customers”, which would represent every customer in the Customer dimension.
6. (Optional) In the **Description** box, type a description for the dimension.
7. Click **OK** to add the dimension.

When you have added a dimension, you also have to add at least one dimension level. The **Add Dimension Level** dialog is displayed when you click **OK** to add a dimension. For more information, see [Adding Dimension Levels](#).

Once you have created a dimension, you can use the dimension in several cubes at the same time. Dimensions that are used in more than one cubes at a time are known as role playing dimensions.

Parent-Child Dimensions

Parent-child dimensions are used to create unbalanced hierarchies where the branches descend to different levels, and where the parent and the child exist in the same table. Typically, parent-child hierarchies are used for creating organizational hierarchies.

Creating Parent-Child Dimensions

A parent-child dimension is a hierarchy that is defined by a parent column and a child column in the same table. A member of the hierarchy can appear more than once in the hierarchy.

To create a parent-child dimension, follow the steps below.

1. Expand **SSAS Multidimensional Servers**, expand the relevant server, right-click **Dimensions** and click **Add Parent-Child Dimension**. The **Add Parent-Child Dimension** window opens.



2. In the **Name** box, type a name for the dimension.

3. In the **Unknown Member** list, select **Visible** to apply an Unknown Member to dimension keys in the fact table with no matching dimension members.
4. In the **Table** list, select the main table of the dimension.
5. In the **Key Column** list, select the key column of the child table. This column identifies each member of the dimension.
6. In the **Parent Column** list, select the key column of the parent field. This column identifies the parent of each member.
7. In the **Lay-out** list, select how you want the dimension level displayed to the end user. The following options are available:

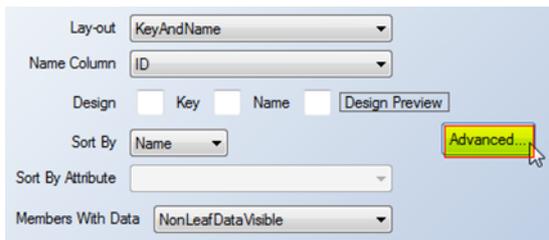
Setting	Description
Key	Displays only key column values
Name	Displays only name column values
KeyandName	Displays the key column first and then name column values
NameandKey	Displays the name column values first and then the key column values

8. In the **Name column** list, select the column that provides a meaningful value to the user. This field is only available if you have selected the **Name**, **KeyAndName**, or **NameAndKey** layout.
9. In the **Design** fields, specify which separator to use in the front-end application to separate Key and Name. This field is only enabled if you have selected KeyAndName or NameAndKey. The order in which the Key and Name text fields appear depends on your selection in the Layout list. To preview the design of the layout, move the pointer over **Design Preview**.
10. In the **Sort By** list, select whether you want the values sorted by Key or Name.
11. In the **Sort by Attribute** list, select the specific attribute key or name that you want the values sorted by. This list is only available when you are working with key levels, and Sort By is set to **AttributeKey** or **AttributeName**.

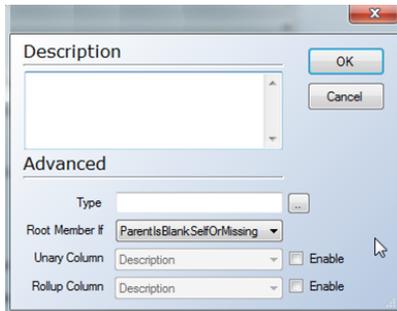
Note: When you create the parent-child dimension based on a consolidation table, you will typically use a **Sort By Attribute**. You therefore need to create a Sort order dimension level where the key column is **Sort Order**. Then, you must enable Unary column and Roll up column on the dimension. You can then set the parent-child dimension to **Sort By Attribute**.

Defining Advanced Parent-Child Dimension Settings

1. To access advanced settings for Parent-Child dimensions, click **Advanced...** in the **Add/Edit Parent-Child Dimension** window.



The **Advanced** window opens.



- Next to the **Type** box, click the ellipsis (...) and select the type of dimension you want to create. You can leave the **Type** blank for regular dimension types, which are the most common type. For a list of possible dimension types, see [Creating Regular Dimensions](#).
- In the **Root Member If** list, select one of the following options that controls when the dimension is the root member:

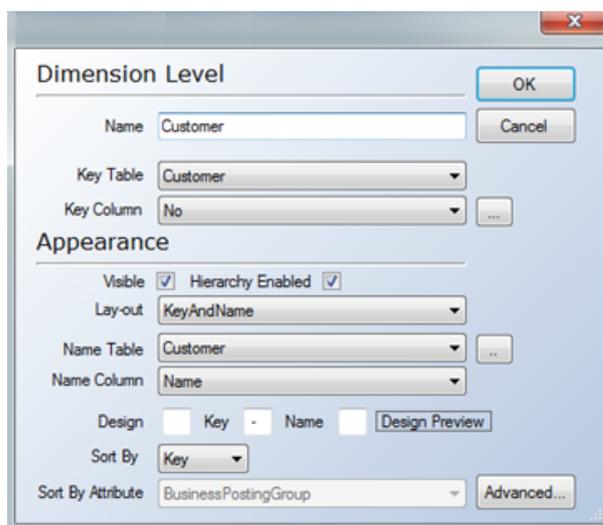
Option	Description
ParentsBlank	Hides the root if the member is a null, a zero or an empty string
ParentsBlankSelfOrMissing	Hides the root if the member is a null, a zero, an empty string, if the parent is missing, and if the member itself is a parent
ParentsMissing	Hides the root if the parent is missing
ParentsSelf	Hides the root if the member itself is a parent

- In the **Unary Column** list, select the column that contains the unary operators that are used in this dimension level. If you have to select **Enable** to select from this list.
- In the **Roll-up Column** list, select the column that contains the roll-up values used in this dimension level. You have to select **Enable** to select from this list.

Adding Dimension Levels

Dimension levels are used to create a dimension attribute within a cube, which enables a user to drill down or roll-up through data. A dimension must contain at least one dimension level.

1. Open and expand the server that contains the dimension you want to add a level to. Then expand **Dimensions**, right-click the dimension and click **Add Dimension Level**. The **Dimension Level** window opens.



2. In the **Name** box, type a name for the dimension level.
3. In the **Key Table** list, select the table in the data warehouse to add the dimension from.
4. In the **Key Column** list, select the column which uniquely identifies the records in the table. If the key is a composite key, click the ellipsis button (.), to select the attributes on which the Key column is based.
5. Select **Visible** if you want the level to be displayed in the front-end application.
6. In the **Layout** drop-down, select the way that the dimension level should be displayed to users. The options are:

Setting	Description
Key	Displays only key column values
Name	Displays only name column values
KeyandName	Displays the key column values first and then name column values
NameandKey	Displays the name column values first and then the key column values

7. In the **Name Table** list, select the table that provides a meaningful name to the user. To set the Name Table value to the same value as the Key Table value, click the ellipsis button.
8. In the **Name Column** list, select the column that provides a meaningful value to the user.
9. In the **Design** fields, type the separators to use in the front-end application to separate key and name. This field is only enabled if you have selected KeyAndName or NameAndKey. The order in which the Key and Name text fields appear depends on

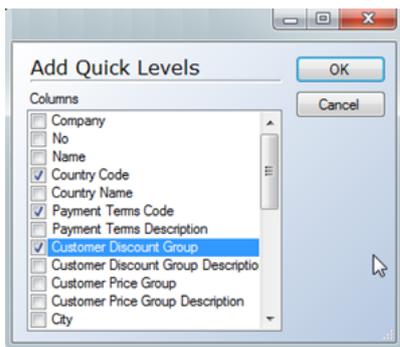
your selection in the Layout list. To review the design of the layout, move the pointer over **Design Preview**.

10. In the **Sort By** list, select if you want the values sorted by **Key** or **Name**. If you are adding a key level, you can also sort by **AttributeKey** and **AttributeName**.
11. In the **Sort By Attribute** list, select the specific attribute key or name that you want the value sorted by. This list is only available when you are working with key levels, and **Sort By** is set to AttributeKey or AttributeName.
12. Click **OK**. The dimension level is added to the **Dimensions** tree below the dimension it belongs to. The SSAS Multidimensional database must be deployed and executed before this change takes effect in the front-end.

Adding Quick Levels

Quick Levels provide an easy way to add new dimension levels. It will automatically set defaults which can later be changed by editing the dimension level.

1. Expand the relevant server, expand **Dimensions**, right-click the parent-child dimension or key level on the dimension to which you want to add a level and click **Add Quick Levels**. The **Add Quick Levels** window appears.



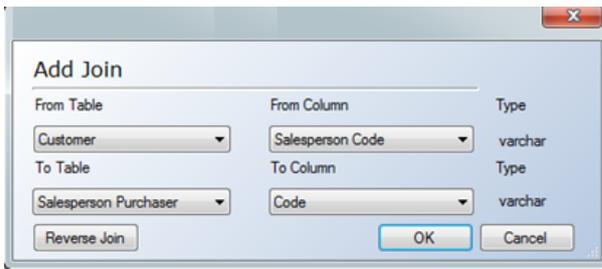
2. Select the columns you want to use as levels from the source table in the data warehouse, and then click **OK**.

The levels you have added are now available when you create a hierarchy. The SSAS Multidimensional database must be deployed and executed before this change takes effect in the front-end.

Adding Dimension Joins

Dimension joins are joins between two tables that are not directly related to the dimension's fact table. You only use dimension joins in snowflake schemas where you want more than one table in a dimension. The join is a one-to-many join.

1. Open and expand the server that contains the dimension you want to modify, expand **Dimensions**, expand the dimension to which you want to add a join, right-click **Dimension Joins**, and then click **Add SSAS Multidimensional join**. The **Add Join** window opens.

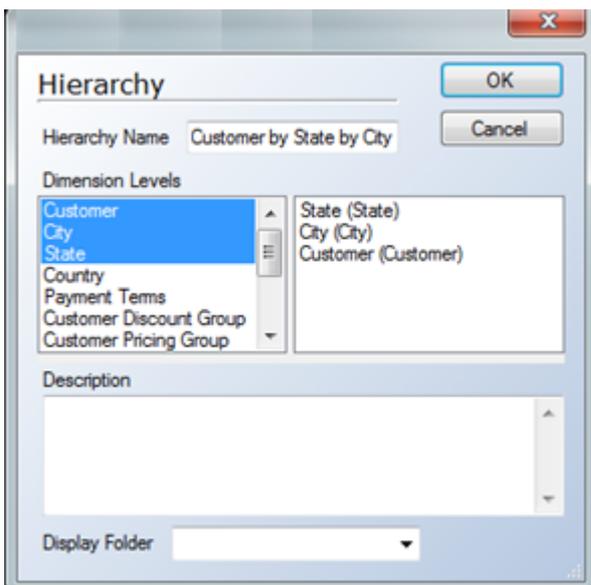


2. In the **From Table** list, select the table *from* which you want to create a join.
3. In the **To Table** list, select the table *to* which you want to create a join.
4. In the **From Column** list, select the column from which you want to create a join. The column's data type is displayed next to the list. You can only make joins between fields with compatible data types.
5. In the **To Column** list, select the column to which you want to create a join. If the column's data type is not compatible with the data type of the **From Column**, the data type is displayed in red.
6. If you want to reverse the direction of the join, click the **Reverse Join** button.
7. Click **OK**. The dimension join is displayed in the **Dimension Joins** folder in the **Dimensions** tree.

Adding Dimension Hierarchies

Once you have added dimension levels to a dimension, you can create a dimension hierarchy. Dimension hierarchies make it easier for users to look at commonly used dimension groupings by only having to drag one icon into a report. An example of this could be Customers by Country or Items by Item Category.

1. Expand the relevant server, expand **Dimensions**, right-click the dimension to which you want to add a hierarchy and then select **Add Hierarchy**. The **Hierarchy** window opens.



2. In the **Hierarchy Name** field, enter a name for the hierarchy. The name cannot be the same as the name of a dimension level.
3. In the **Dimension Levels** pane, click the levels you want to be part of the hierarchy. The hierarchy elements are then listed in the right pane. You can drag the dimension levels in the right pane up and down to specify the order they should exist in from top to bottom
4. In the **Description** field, type a description of the hierarchy. This field is optional.
5. In the **Display Folder** list, select the folder where the hierarchy is displayed by the front-end application. This is optional.
6. Click **OK**. The SSAS Multidimensional database must be deployed and executed for this to be finalized for the end users.

Note: Since the hierarchy is associated with the dimension itself, once the dimensions and cubes are deployed and executed, the hierarchy will automatically show up in all cubes in which the dimension exists.

Adding a Time Dimension

Date dimensions are based on time or date tables, so you have to create a time or date table in the data warehouse before you can create a time dimension. For more information, see [Adding Date Tables](#).

To add a date dimension on your server based on a date table, follow the steps below:

1. Right click **Dimensions** and click **Add Time Dimension**. The **Add Date Dimension** window opens.
2. Type a **Name** for your Date Dimension, click the date table you want to use as a basis for the date dimension in the Table list and click **OK**.
3. The date dimension will then be created based on the data table and appear in the SSAS Multidimensional tree under **Dimensions**. It includes date, week, month, quarter, half year and year dimension levels, in both fiscal year and non-fiscal year variations, as well as any custom periods and **Calendar** and **Fiscal Calendar** hierarchies.
4. (Optional) It might be useful to configure any custom period dimension levels to use the name of the custom period as a key. If, for instance, you have defined a number of national holidays across many years, these will then be grouped as opposed to a having “unique” yearly holidays for each year. To do so, right click any custom period level and click **Edit Dimension Level** and click **[custom period name]Name** in the **Key Column list**.

When you expand the date dimension, you can see that the levels which correspond to fields on the date table in the data warehouse have already been added. However, you can add more levels simply by adding [quick levels](#) or by [adding regular levels](#). You can add hierarchies as well.

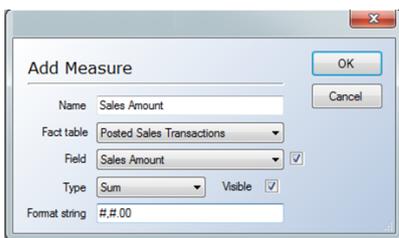
Measures

Measures determine the numerical values of a cube and a cube must contain at least one measure. You can define the following three types of measures:

- **Standard measures** obtain their values directly from a column in a source fact table.
- **Derived measures** are derived before aggregation or summing of columns. This means that they are calculated when the cube is processed and are stored in the fact table. You can use standard arithmetic operators and MDX statements to create derived measures.
- **Calculated measures** are calculated after aggregation and summing. They are calculated at query time and are never stored. You can create calculated measures using standard arithmetic operators and MDX statements and can also combine them with other measures.

Adding Standard Measures

1. Expand the cube to which you want to add a measure, right-click **Measures** and click **Add Standard Measure**. The **Add Measure** window appears.



2. In the **Name** box, type a name for the measure.
3. In the **Fact Table** list, select the fact table that you want to use for the measure.
4. In the **Field** list, select the field that you want as the measure. Disable this field by clicking the box next to it.
5. In the **Type** list, select the preferred aggregation method. You have the following options:

Aggregation Method	Description
SUM	Returns the sum of all values
COUNT	Counts all rows and returns the total number of rows
MIN	Returns the lowest value
MAX	Returns the highest value
DistinctCount	Returns the number of unique values

6. Select **Visible** if you want the measure to be displayed in the front-end application.
7. In the Format string field, specify how you want the numeric results displayed. You have the following options:

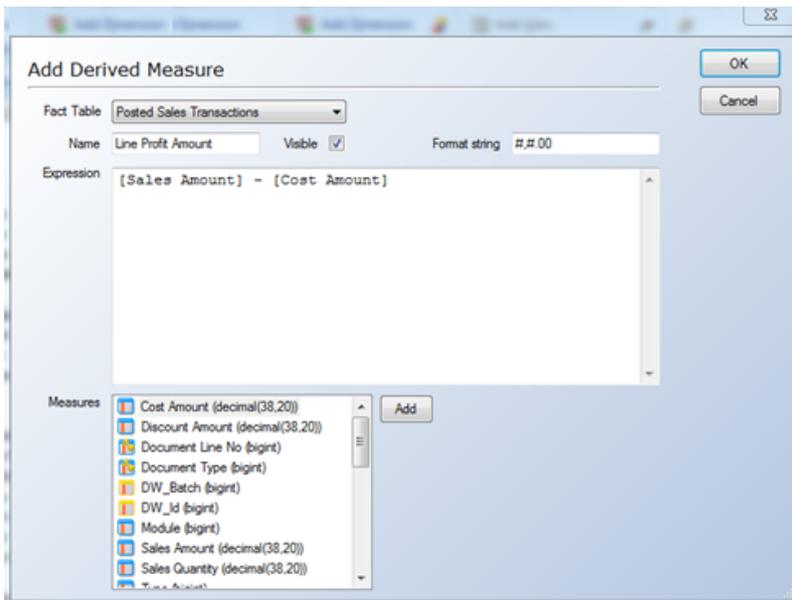
Format String	Description
None	Applies no formatting
0	Displays a digit if the value has a digit where the zero (0) appears in the string, otherwise a zero is displayed
#	Displays a digit if the value has a digit where the number sign (#) appears in the string, otherwise nothing is displayed
.	Determines the number of digits displayed to the left and right of the decimal separator.
%	Is a percentage placeholder
,	Separates thousands from hundreds
Percent	Typing Percent will default the measure to showing as a percentage with two decimal places

Below are examples of what the output will look like for various combinations of the format strings:

Format String	Output
None	1234567.89
#, #	1,234,567
#, #.00	1,234,567.89
#, #%	1%
#, #.0%	1.2%
Percent	1.23%

Adding Derived Measures

1. Expand the cube to which you want to add a measure, right-click **Measures** and click **Add Derived Measure**. The **Add Derived Measure** window appears.



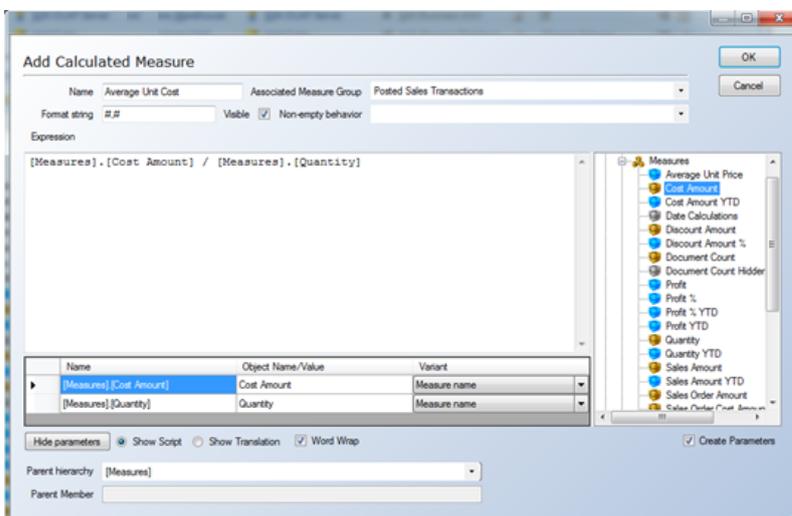
2. In the **Fact Table** list, select the fact table that you want to use for the measure.
3. In the **Name** box, type a name for the derived measure.
4. Select **Visible** if you want the measure to be displayed in the front-end application.
5. In the **Format String** box, type how you want the numeric results displayed. The format is the same as for standard measures. See [Adding Standard Measures](#).
6. 7. In the **Expression** box, enter an MDX statement
or

In the **Measures** list, select the measures from the fact table that you want to use for the derived measure and click **Add**. Your selections will be added to the expression, where they can be combined with mathematical operators to achieve the outcome you desire.

7. Click **OK** to add the derived measure.

Adding Calculated Measures

1. Expand the cube to which you want to add a measure, right-click **Measures** and click **Add Calculated Measure**. The **Add Calculated Measure** window appears.



2. In the **Name** box, type a name for the calculated measure.
3. Select **Visible** if you want the value to be displayed in the front-end application.
4. In the **Format string** field, specify how you want the numeric results displayed. The format is the same as for standard measures. See [Adding Standard Measures](#).
5. (Optional) In the **Non-empty Behavior** list, select the measure or measures used to resolve NON EMPTY queries in MDX.
6. In the **Expression** field, write an MDX statement or, in the **Measures** list, drag the measures to be used for the calculated measure into the workspace in the middle.

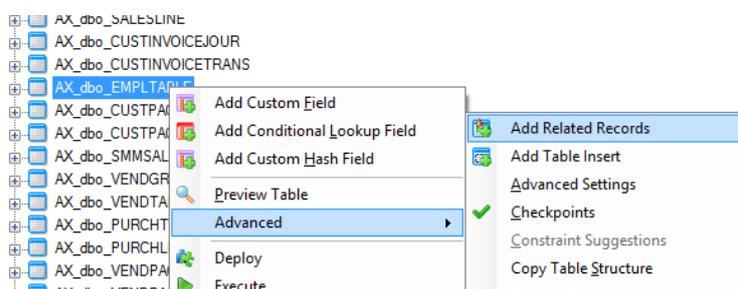
Handling Early Arriving Facts

In a live working environment, it is possible that transactional data may contain values that have not yet been added to the source database in the corresponding dimension table. An example of this could be a Sales Invoice that has a Salesperson Code where the Salesperson Code does not yet exist in the Salesperson table. When the data warehouse is updated and the cubes are processed, the values for this salesperson will fall under the “Unknown” member for the Salesperson dimension. This happens because the cube does not see the Salesperson Code on the transaction as being a known value when compared to the list of salespeople in the Salesperson dimension.

In TimeXtender, it is possible to handle these “early arriving facts” in such a manner that they will show at least partial information until the data source is properly updated with all of the normal dimension information. This prevents information from being placed into the “Unknown” member when the data is consumed by end-users. Once the dimension value is properly added to the ERP system or data source by a user, all fields for the previously missing record will then be populated according to the values in the data source.

Enabling Early Arriving Facts

1. Identify the dimension table to which relevant values from the transaction table should be added, right-click the table name, and go to **Advanced -> Add Related Records**.



The **Add Related Records** window opens.

Add Related Records (AX_dbo_EMPLTABLE) OK Cancel

Name:

Create Records from table:

Record Condition:

Data Destination Table:

Field Mapping

	Mapping	Fixed Value	Allow Default Value	Default Value
DATAAREAID	None		<input type="checkbox"/>	
EMPLID	None		<input type="checkbox"/>	

Conditions Add

- In **Name**, type a descriptive name for the **Add Related Records** rule that is currently being created.
- In the **Create Records from Table** list, select the transaction table that will identify the table from which to bring in potential new values. A window may appear stating that all mappings and conditions will be cleared. Click **Yes**.

Warning ×

Changing the table will clear all mappings and conditions!
 Do you want to continue?

- In the **Record Condition** list, select the option to determine when data will be inserted into the dimension table if new values are found in the transaction table. The most common option is **Not Exist**, which will add in values that do not currently exist in the dimension table.
- Select the **Data Destination Table** to insert the values into. The default option is the Raw table.
- In the **Field Mapping** table, specify the fields to be mapped from the transaction table and inserted into the dimension table. In the example below, the DW_Account field (Company) and Salesperson Code fields will be extracted from the transaction table and inserted into the dimension table.

Field Mapping

	Mapping	Fixed Value	Allow Default Value	Default Value
DATAAREAID	DATAAREAID		<input type="checkbox"/>	
EMPLID	SALESADMINISTRATOR		<input type="checkbox"/>	

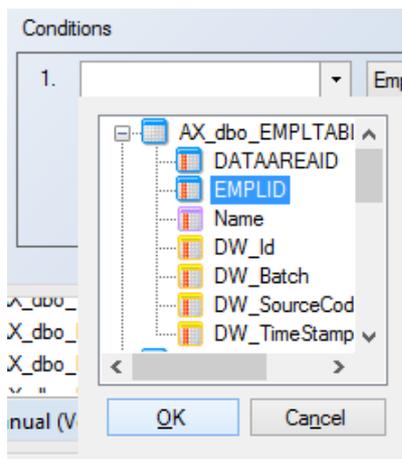
- It is possible to add in fixed values for fields in the dimension table that the transaction may not have data for. In the example below, the fixed value “Missing Salesperson” will be added in the **Name** field for all Salesperson Codes added from the transaction table. This is achieved by selecting the **Fixed Value** option in the **Mapping** column for the **Name** field and typing the desired fixed value in the **Fixed Value** column.

Field Mapping		
	Mapping	Fixed Value
DATAAREAID	DATAAREAID	
EMPLID	SALESADMINISTRA...	
Name	Fixed Value	Missing Salesperson

- If desired, a default value can be inserted instead of bringing in the values that exist in the transaction table. This could be used to assign fixed values to all data brought in for early arriving facts. This is achieved by clicking the checkbox in the **Allow Default Value** column and typing the corresponding fixed value in the **Default Value** column. This is not common.

Field Mapping				
	Mapping	Fixed Value	Allow Default Value	Default Value
DATAAREAID	DATAAREAID		<input type="checkbox"/>	
▶ EMPLID	SALESADMINISTRA...		<input checked="" type="checkbox"/>	Missing
Name	Fixed Value	Missing Salesperson	<input type="checkbox"/>	

- The last step is to define the relationship between the two tables. Click the **Add** button in the **Conditions** section.
- Select the first field to join in the dimension table (**Code**), and click **OK**.



- Select the operator to be used for the join. The most common operator is **Equal**.
- Select the matching field in the transaction table (**Salesperson Code**), and click **OK**.



- Repeat steps 9 through 12 for any additional joins that need to be made (such as Company). The final result will look similar to this:

Add Related Records (AX_dbo_EMPLTABLE)

Name: Add missing from CUSTINVOICEJOUR

Create Records from table: AX_dbo_CUSTINVOICEJOUR

Record Condition: Not Exists

Data Destination Table: Raw

Field Mapping

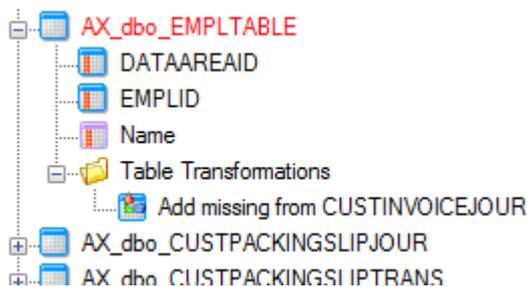
	Mapping	Fixed Value	Allow Default Value	Default Value
DATAAREAID	DATAAREAID		<input type="checkbox"/>	
▶ EMPLID	SALESADMINISTRATOR		<input checked="" type="checkbox"/>	Missing
Name	Fixed Value	Missing Salesperson	<input type="checkbox"/>	

Conditions

1.	EMPLID	Equal	Fixed Value <input type="checkbox"/>	SALESADMINISTRATOR	Remove
2.	DATAAREAID	Equal	Fixed Value <input type="checkbox"/>	DATAAREAID	Remove

Click **OK** when finished to save the settings, and close the **Add Related Records** window.

A folder for **Table Transformations** will be added to the bottom of the dimension table. The selection criteria that were previously set can be edited by right-clicking the transformation and selecting **Edit Related Record**.



- Deploy and execute the dimension table. Any records that exist in the transaction table, but not in the dimension table, will be added during the data cleansing process. A screen-shot of the result based on the example in this document is shown below.

Table: AX_dbo_EMPLTABLE

	DATAAREAID	EMPLID	Name	D
	ceu	7225	Null	83
	ceu	7230	Null	84
	ceu	7231	Null	85
	ceu	7232	Null	86
	ceu	7240	Null	87
	ceu	9001	Null	88
	ceu	9002	Null	89
▶	ceu		Missing Salesperson	90

The salesperson code “BP” existed on a sales document, but no corresponding Salesperson Code existed in the Salesperson table. Once the salesperson is properly added to the ERP system and the table is refreshed, all proper information will be pulled in from the ERP system, and the name will no longer say “Missing Salesperson.”

Slowly Changing Dimensions

Slowly Changing Dimensions (SCD) enable an organization to track how dimension attributes change over time. For example, it is possible that an item may be associated with a particular product group code but that it is later reclassified into a different product group. The organization wants to be able to analyze the historical sales data that occurred when the item was assigned to the original product group as well as more recent sales data that has occurred after the item was reclassified to the new product group.

Different Types of Slowly Changing Dimensions

Type I

Type I dimensions will automatically overwrite old data with updated data from the data source. An example of this would be a change in a customer name. In the data source, the name for a particular customer is changed from ABC Consulting to Acme Consulting. The next time that the data warehouse is updated the customer name will be changed from ABC Consulting to Acme Consulting and no historical record of the change is kept. All historical, current, and future transactions will be displayed under the new customer name of Acme Consulting. This is the default methodology of updating data in the data warehouse and no setup is required.

Type II

Type II dimensions will enable the tracking of dimension attributes historically by inserting additional records into the table as the values in specified fields are changed. TimeXtender will administer the tracking of the dimension values as well as the updating of the table. Each record for a particular value, such as an item number, can be viewed as a different version of this item. The transaction table can then be linked to this table to display which version of the item was associated with the transaction based on the transaction date.

The follow example illustrates what the Customer dimension table would resemble if the example above was tracked using Type II functionality.

Customer No	Name	City	State	Version	From Date	To Date
123	ABC Consulting	Portland	OR	1	1/1/1900	9/18/2012
123	Acme Consulting	Portland	OR	2	9/18/2012	12/31/9999

Implementing Type II Slowly Changing Dimensions

The steps below will explain how to utilize slowly changing dimensions - also known as History in TimeXtender - in a project. In the example, there will be an item named "Bicycle" that has historically had an Inventory Posting Group of "Finished". Recently, however, this item has been reclassified and is now associated with the Inventory Posting Group "Resale". The organization wishes to track sales for this item under both the historical Inventory Posting Group as well as the new one.

Enabling Slowly Changing Dimensions on the Dimension Table

See [Enabling History on a Table](#) to learn how to set up history on the dimension table.

Example

The screen-shot below illustrates what the Item table currently looks like for the item that is being used in this example. There is one record for the item and currently the Inventory Posting Group is set to "Finished".

DW_Account	No.	Description	Inventory Posting	Inventory Posting
CRONUS EXT U...	1000	Bicycle	FINISHED	FINISHED

The item is now reclassified into the Inventory Posting Group Code of "Resale.". An invoice is then posted that reflect a sale of 100 of the bicycles with the new Inventory Posting Group. To illustrate how the Item table now looks in the staging database, the table is executed to reflect the changes and the results are displayed below:

DW_Account	No.	Description	Inventory Posting	Inventory Posting	Gen. Prod.	Product Posting	Product Group	Product Group	Item Category	Item Category	Line Type	Line Type	DW_Id
CRONUS EXT U...	1000	Bicycle	FINISHED	FINISHED	RETAIL	Retail		Null		Null	2	Item	1
CRONUS EXT U...	1000	Bicycle	RESALE	RESALE	RETAIL	Retail		Null		Null	2	Item	145

The DW_Account and No. fields are the same, but the Inventory Posting Groups now reflect the new value. The DW_ID, which represents a unique record number in the table, is also different as illustrated on the right in the screenshot above.

There are a few more fields that pertain to the historical values that are useful as well. When the table is executed and notices a change in one of the Type II fields, it will automatically add in the dates for which the old value ended and the new value begins. These are the "SCD From Datetime", "SCD To Datetime", and "SCD IsCurrent" fields.

SCD From DateTime	SCD To DateTime	SCD Is Current
01-Jan-00	25-Sep-12	0
25-Sep-12	31-Dec-99	1
01-Jan-00	31-Dec-99	1

The To and From field represent the date ranges that this version of the dimension was used in and which record is the current record.

Bringing the Surrogate Key to the Transaction Table

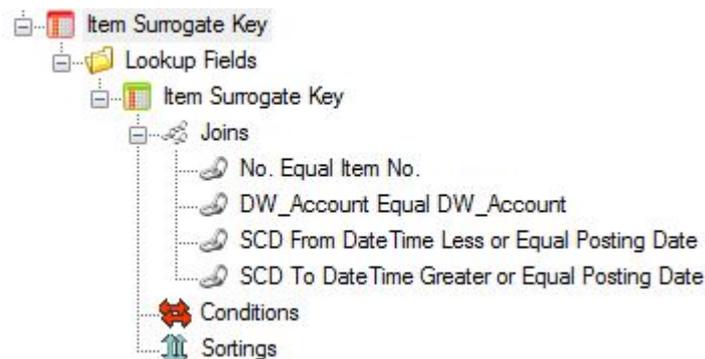
In a standard transaction table, the transaction itself will only be linked to the Item No. This is problematic, as the item number alone does not identify which version of the item record the transaction applies to. In order to see this detail, the surrogate key from the Item table will be brought into the transaction table. This surrogate key is based on the To and From dates in the Item table, as compared with the Posting Date of the transaction. In the screen-

shot above, all transactions between January 1, 1900 and September 25, 2012 will be associated with the first version of the Bicycle item where the Inventory Posting Group is "Finished". Any transaction after September 25, 2012 will be associated with the latest version of the Bicycle item where the Inventory Posting Group is "Resale".

A surrogate key is a substitution for the primary key in a table. The surrogate key most often represents the unique row number in the table. It can be used in one table to refer back to a specific record in another table without having to utilize the natural primary key. In TimeXtender, all tables in the staging database and data warehouse have a called named "DW_ID" which represents the surrogate key in each respective table.

In order to see the DW_ID field in TimeXtender, follow the steps below.

1. Right-click the table, click **Advanced** and click **Show System Control Fields**.
2. Move the **DW_ID** field from the Item table, and add it to the transaction table.
3. Rename the field to "Item Surrogate Key" to make it easily understandable to other users.
4. Add Standard joins between the two tables for DW_Account and the Item No.
5. Add Additional joins for "SCD From Date Time" Less Than or Equal to "Posting Date" and "SCD To Date Time" Greater Than or Equal to "Posting Date". This will capture the correction version of the item based on the Posting Date of the transaction.



6. Deploy and execute the transaction table to have the new field added and populated.

The process above should be repeated for any additional transaction tables where history needs to be tracked.

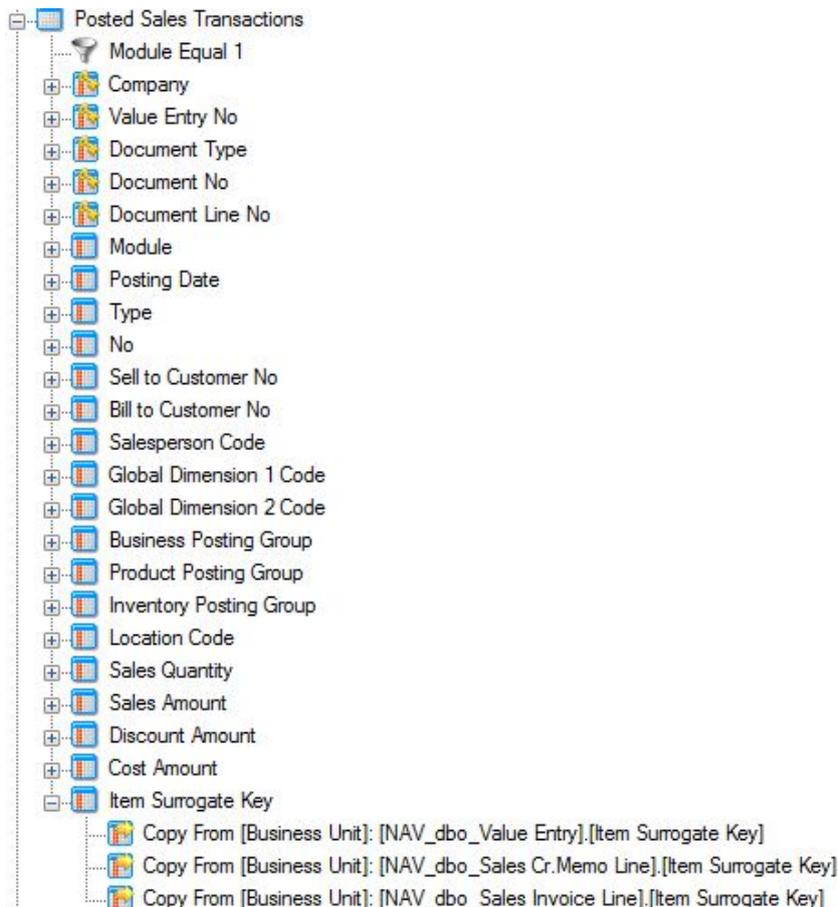
Moving the Surrogate Key from the Transaction Table in the Staging Database to the Data Warehouse

Now that the surrogate key has been added to the transaction table(s), this field needs to be added to the relevant transaction tables in the data warehouse.

To accomplish this, follow the steps below.

1. Drag the surrogate key field (in this case "Item Surrogate Key") from the table(s) in the staging database and drop them onto the relevant tables on the data warehouse.

2. Deploy and execute the transaction table in the data warehouse for the changes to take effect.

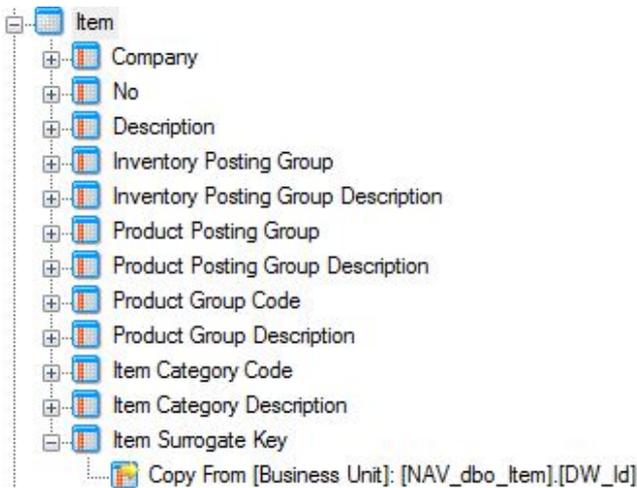


Adding the Surrogate Key to the Dimension Table in the Data Warehouse

The **DW_ID** field now needs to be added to the related dimension table in the data warehouse. This ensures that the proper mapping will be made between the dimension table and the transaction table in the cubes.

To accomplish this, follow the steps below.

1. Drag the DW_ID field from the dimension table in the staging data (in this example, the Item table) to the related dimension table in the data warehouse (in this example, the Item table).
2. Deploy and execute the dimension table for the changes to take effect.



Updating the Dimension Key

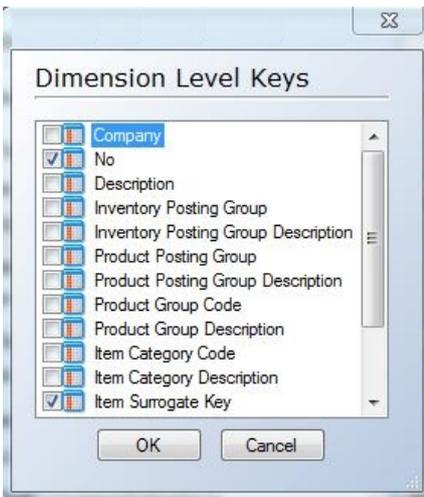
The dimension key should now be updated to include this surrogate key. This will ensure that Analysis Services sees the uniqueness of the dimension, not as the natural key (in this example Item No.), but as the combination of Item No. and the surrogate key.

To accomplish this, follow the steps below.

1. Expand **Dimensions**, expand the dimension, and edit the key level (in this example "Item").



2. To the right of the Key Column click the ellipsis (...), and add the surrogate key to the dimension key (in this example, it is the "Item")



Updating the Dimension Relationships in the Cube

The relationships between the dimension and the transaction table should be updated in the cube to reflect the change made to the dimension key in the previous step.

1. Right-click the relevant dimension in the cube(s), click **Dimension Relations** and click **All Fact Tables**.
2. Set the dimension relationship to use the surrogate key that was added to the transaction table in a previous step.

Dimension Relations : Item

Dimension Level	Key Column	Posted Sales Transactions
Item (KEY)	No	No
Item (KEY)	Item Surrogate Key	Item Surrogate Key
Inventory Posting Group	Inventory Posting Group	
Product Posting Group	Product Posting Group	
Product Group	Item Category Code	
Product Group	Product Group Code	
Item Category	Item Category Code	

3. Deploy and execute the SSAS Multidimensional database for the changes to take effect.

The final result is that users can see data based on the historical attributes that may no longer exist in their ERP system because the information has been overwritten. In the screen-shot below, the "Bicycle" item shows up twice with the sales amounts associated with the various Inventory Posting Groups that have been used for the item over time.

Row Labels	Inventory Posting Group	Sales Amount
1000 - Bicycle	FINISHED	150,000
1000 - Bicycle	RESALE	300,000
1896-S - ATHENS Desk	RESALE	1,327,390
1900-S - PARIS Guest Chair, black	RESALE	1,780,730
1906-S - ATHENS Mobile Pedestal	RESALE	136,797

Building Qlik Models

Warning: The Qlik Modeler has been deprecated. We recommend using the [Shared Semantic Layer](#) for building Qlik models.

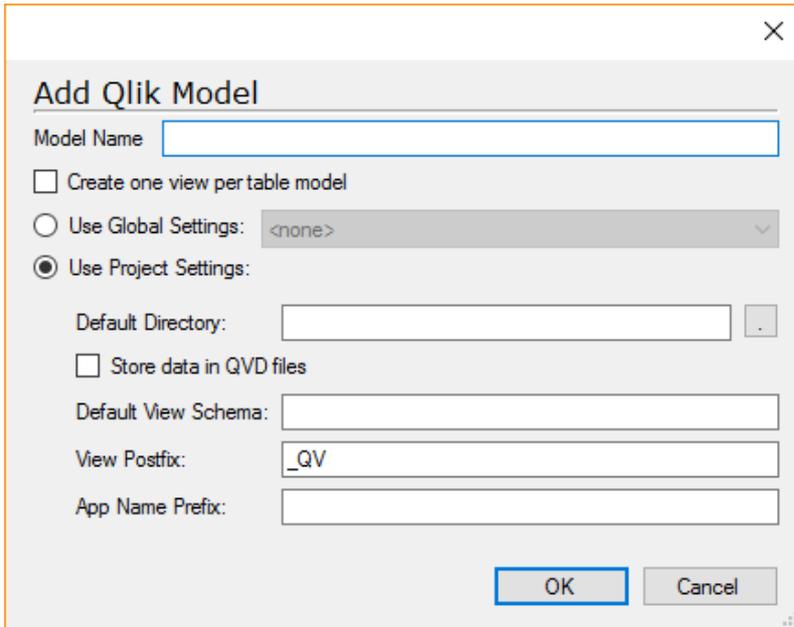
The Qlik Modeler in TimeXtender creates data structures that can be used by both QlikView and Qlik Sense. For Qlik Sense, TimeXtender can create an app directly in Qlik Sense Desktop and deploy an app to Qlik Sense Enterprise. For QlikView, the script generated by TimeXtender can be imported or included in the application.

Building a Qlik Model

Adding a Qlik Model

To add a new Qlik model, follow the steps below.

1. In the **Solution Explorer**, right click **Semantic Layer** and click **Add Qlik model**. The **Add Click Model** window appears.



2. In the **Model name** box, type a name for the model.
3. In the **Default directory** box, enter the directory where you would like to store the generated Qlik script files.
4. Select **Store data in QVD files** if you would like to store the retrieved data in QVD files for later reuse by the Qlik apps.

Note: If you want to deploy and execute to Qlik Sense Enterprise, using QVD files require additional setup. You also need to make sure that the **Default directory** path can be reached by the Qlik Sense server. See [Deploying to Qlik Sense Enterprise](#) for more information.

5. (Optional) In the **Default view schema** box, type a schema name if you would like the view generated by the Qlik Adapter to belong to a special schema for an easier overview.
6. (Optional) In the **View postfix** box, type a postfix for the generated views. Default is “_QV” (for “Qlik Valid”).
7. Select **Create one view per table model** if you would like TimeXtender to generate one view per table model in your solution, even if the same table with the same fields selected is used multiple times. The default behavior is to create one view per unique table model. This means that if two table models have the same fields, TimeXtender will only create one view.
8. Click **OK**.

Cloning a Qlik Model

Instead of creating a Qlik model from scratch, you can clone an existing Qlik model to create a new Qlik model that is identical to the one it was cloned from. Naturally, this is useful if you need to create a Qlik model that is very similar to an existing Qlik model.

- To clone a Qlik model, right click the model, click **Advanced** and click **Clone Qlik model**.

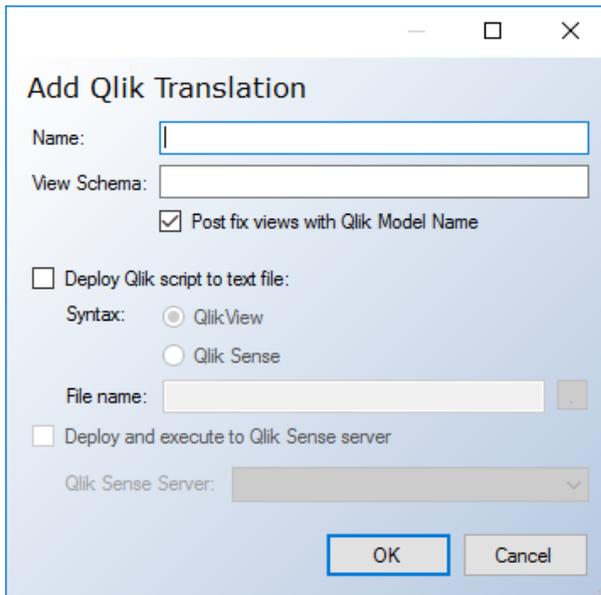
Adding a Qlik Model Translation

Each Qlik model can have a number of translations that you can use for internationalizing your solution. You need at least one translation, so when you create a Qlik model, TimeXtender will automatically create a translation for you. Translations are more than just your data labeled in a different language, however. Deployment of a Qlik Model is also done on the translation level.

Translations can be found under Model Translations under each Qlik model. To create a new translation, follow the steps below.

1. Right click **Model Translations** under the relevant Qlik model and click **Setup Qlik Model Translations**. The **Qlik Model Translations** window appears.

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

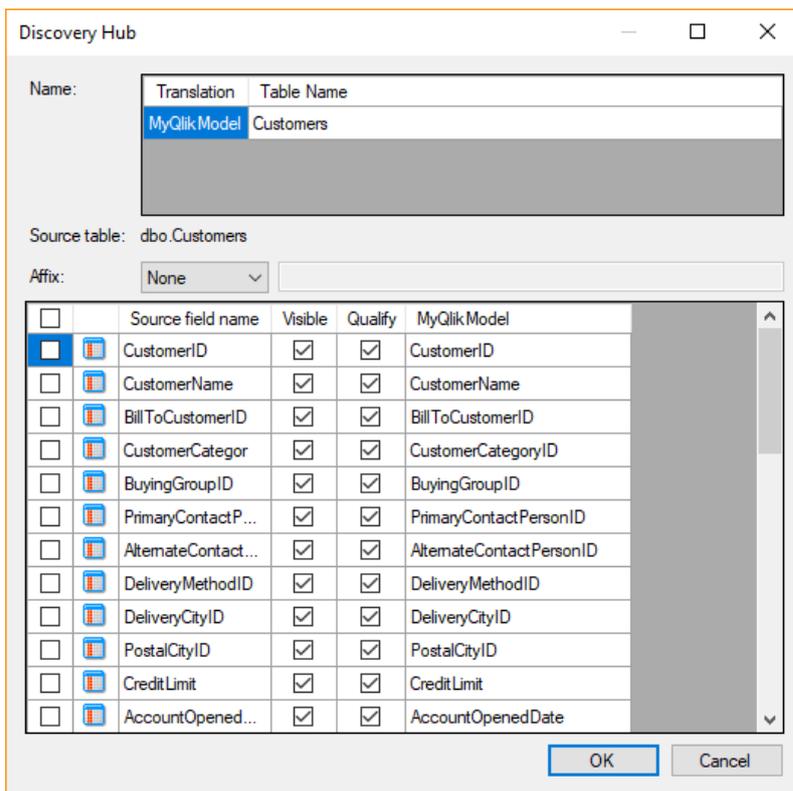


3. In the **Name** box, type a name for the translation, for instance the language of the translation.
4. (Optional) In **View Schema**, type a name for the schema the related views should use.
5. Clear **Post fix views with Qlik model name**, if you do not want TimeXtender to postfix the views with the model name. This is not recommended since it means that some views might end up with the same name. In that case, that last view to be deployed will be the only view there is of that name.
6. Select **Deploy Script to Text file** if you want to output the generated Qlik script to a text file. Under **Syntax**, make sure that the application, you will be using the script with, is selected. Type a path and name for the file in the **File name** box.
Note: If you want to deploy and execute to Qlik Sense Enterprise, you need to make sure that the **File name** path can be reached by the Qlik Sense server.
7. Select **Deploy and execute to Qlik Sense server** and select a server in the **Qlik Sense server** list to deploy the Qlik model to Qlik Sense Enterprise.
8. Click **OK**. The Add Translations window closes and a new column is added in the Qlik Model Translations window.
9. Type the translated names in the column under the name of your newly created translation and click **OK** when you are done.

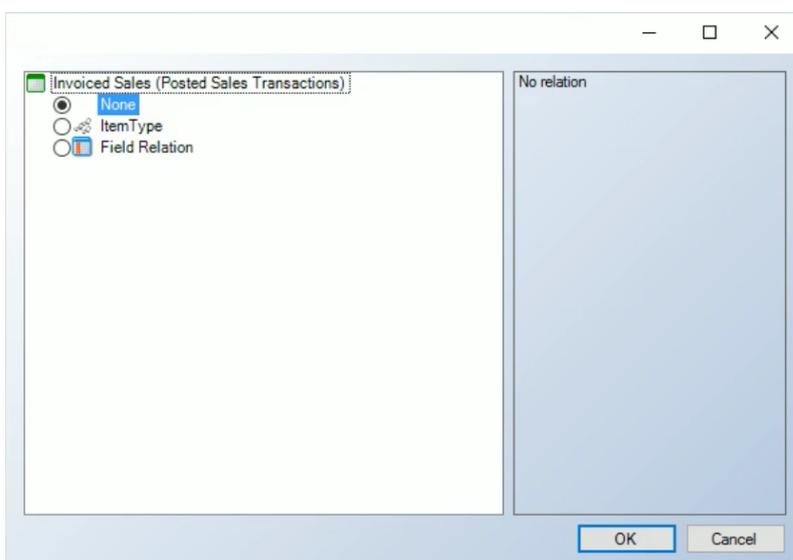
Adding A Qlik Table Model to a Model

A Qlik model should contain a number of table models. A table model is simply the term we use for a table that is part of a Qlik model. You can add the same table multiple times, for instance if your database contains a Sales table with both a Sell-to-customer and a Bill-to-customer. To add a table model, follow the steps below.

1. Drag a table from a data warehouse or staging database to a Qlik model. The **Add Qlik Table** window appears.



2. In the **Name** list, click a name in the **Table name** row to rename the table for the translation listed in **Translation** column on the same row.
3. Select the fields you want to include in your table.
 - Clear the checkbox in the visible column if you do not want to show the field in QlikView/ Qlik Sense, but need to include the field to e.g. create a relation.
 - Clear the checkbox in the Qualify column if you do not want to qualify the table name.
 - Each translation has a column. Click a name in that column to rename the field in that translation.
4. If the Qlik model already contains other table models, the **Setup Qlik Table Model Relations** window appears.



Here, you can set how the table you are adding is related to existing table models in the Qlik model. For each existing table model, you have the following options:

- None: No relations to that table model.
- An existing relation defined in the data warehouse (recommended).
- Field relation: Relate using identical field names on both tables.

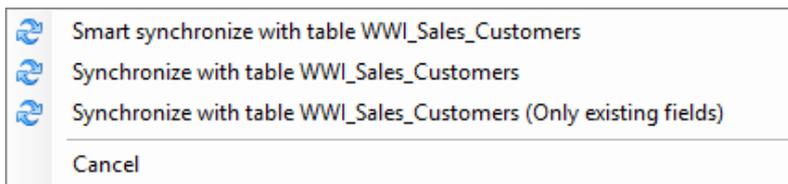
5. Click **OK**.

Adding a Concatenated Qlik Table Model

A concatenated table is a table that combines data from multiple source tables. In QlikView or Qlik Sense, concatenated tables are useful for e.g. avoiding circular references. Concatenated tables can be created in the data warehouse, but creating them in the Qlik Modeler allows you to have one architecture in the data warehouse and another in Qlik without duplicating a lot of data.

To create a concatenated table model

1. Drag a table from a data warehouse or staging database to an existing Qlik table model. A menu appears.



You have the following options:

- **Smart synchronize with table [table name]:** TimeXtender looks on the other source tables on the Qlik table model and adds the fields from the source table that matches fields from the other source tables.
- **Synchronize with table [table name]:** Adds the fields of the source table to the Qlik table model.
- **Synchronize with table [table name] (only existing fields):** Adds the fields of the source table that have the same name as a field already on the Qlik table model.

Click on the option you want to use.

Under **Mappings**, you can see the tables that delivers data to the table model. If you expand a field on the concatenated table model, you can see the fields from the tables that are mapped to that particular field.

You can also add new tables to a concatenated table model by dragging fields from a data warehouse or business unit opened in a new window to the Qlik table model.

1. Expand the concatenated table model and drag fields from a data warehouse or business unit table to the fields on the concatenated table model. You will see that new

fields are added under the table model fields and a new table is added under **Mappings** if the table was not already there.

When the Qlik model is deployed, the tables are set to be loaded in the order they appear under mappings.

- To switch the order in which the tables are loaded, click on a table under **Mappings** and press **Alt + Up/Down** to change the order.

Manually Relating Qlik Table Models

If you do not have any relations between table models defined in the data warehouse, you can add them manually. To add a new relation between two table models, follow the steps below.

1. Navigate to the table model you want to relate to another table model.
2. Click the field you want to base the relation on and then drag the field on a field on another table model.
3. TimeXtender will ask you if you want to create a relation. Click **Yes**.
4. (Optional) TimeXtender will manage the relation automatically and make sure that only the relations defined in the TimeXtender project will be used in Qlik. However, you can choose to an unmanaged approach to let the Qlik app create the relation. The Qlik behavior is to relate similarly named fields. To use this approach, right click on the newly created relation, point to **Relation Type** and click **By name**

Adding selection rules to a Qlik Table Model

Like tables on data sources and in data warehouses, Qlik table models support data selection rules. See [Data Selection Rules](#) for more information.

Renaming a Qlik Model, Model Translation or Table Model

It can be very useful to rename a Qlik table model, especially if you use the same table multiple times in your solution. In addition to table models, you can also rename Qlik models and model translations. To rename an object, follow the steps below.

1. Locate the model, model translation or table model and click it.
2. Press F2 to make the name editable.
3. Type the new name for the object and press Enter.

The original name of the table model will be displayed in parenthesis after the name you type.

Debugging Relations with Plain Text Keys

TimeXtender utilizes concatenated key fields for creating relations between Qlik table models. For performance reasons, the value of these fields are hashed to create a value that

uniquely identifies each record in the Qlik table model, but are shorter than the key would otherwise be.

For debugging purposes, it can be useful to see the values before the hashing is applied. In TimeXtender, that can be accomplished by changing the hashing algorithm. To change the hashing algorithm for a Qlik model to plain text

- Right click the Qlik model, click **Hashing Algorithm** and click **Plain Text (Debug)**.

You can then preview the data in a Qlik application.

Since the plain text algorithm is very slow, remember to change the hashing algorithm back when you have finished debugging.

Scripting in Qlik Models

The Qlik Modeler is powerful, but sometimes there are things that call for code. Like other parts of TimeXtender, the Qlik Modeler contains features that enable you to write your own scripts to supplement the scripts generated by TimeXtender .

- You can use Qlik snippets to create reusable pieces of code.
- You can create custom Qlik fields, where you write the code that controls what the field will contain.
- You can use pre- and postscripts to add code before and after the code generated by TimeXtender respectively.

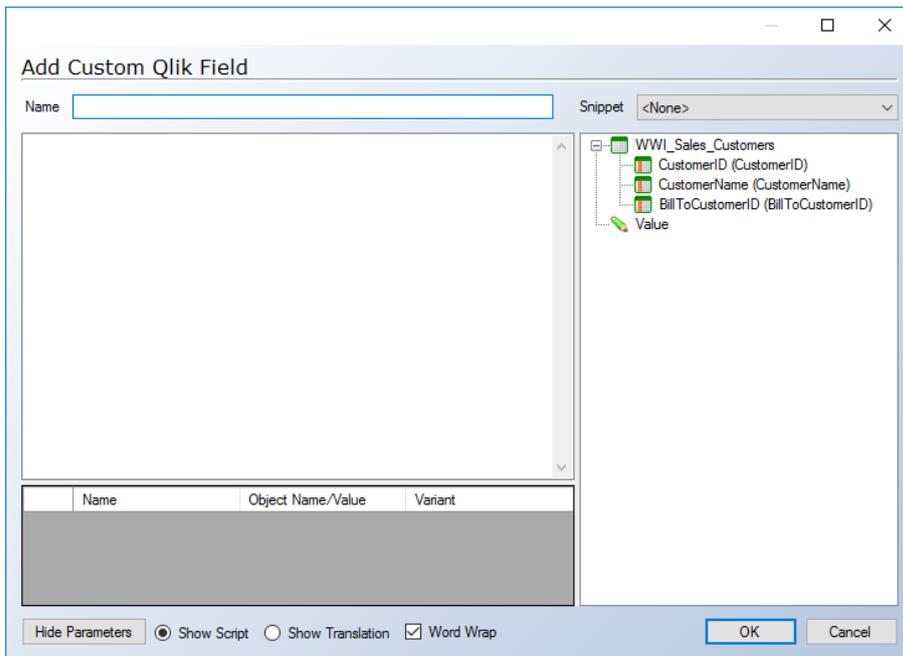
Adding a Qlik Snippet

Like SQL snippets on data warehouses and SSAS Multidimensional snippets on cubes, you can create Qlik snippets. These can be used in custom Qlik fields. See [Snippets](#) for more information.

Adding a Custom Qlik Field

Custom fields are fields where the value is calculated by a Qlik script that you write.

1. To add a custom field, right click a table model, click **Add Custom Qlik Field**. The **Add Custom Qlik Field** window appears.



2. Type a name for the custom field in the **Name** box.
3. If you want to use a Qlik snippet in your custom field, click on the Qlik snippet you want to use in the **Snippet** list.
4. Enter your script in the main text box. You can drag in fields from the list to the right to use in your script. Drag in "Value" to create your own custom variable.
5. Click **OK** to save the script.

Adding Pre- and Postscripts

TimeXtender generates the code that creates the Qlik model you build in the Qlik modeler. You can supplement that code with pre- and postscripts that are added before or after the generated code.

1. Right click **Prescripts**, click **Add Snippet Prescript** and click the snippet you want to use.
- OR -
Right click **Postscript**, click **Add Snippet Postscript** and click the snippet you want to use.
2. Type a name for the script in the **Name** box.
3. In the main text area, enter the code for the Qlik snippet. You can drag in tables, fields, and project variables from the list to the right to use in your function/procedure. Drag in "Value" to create your own custom variable.
4. Click **OK** to save the script.

Adding Pre- and Postscripts based on a Qlik Snippet

In addition to writing it from scratch, you can add a pre- or postscript based on a Qlik Snippet.

1. Right click **Prescripts** and click **Add Prescript**
- OR -
Right click **Postscript** and click **Add Postscript**
2. Type a name for the script in the **Name** box.
3. In the window that appears, map the available fields to the parameters in the snippet. Drag the field(s) from the list on the right and drop the field on the **Object Name/Value** column for the relevant variable. The **Object Name/Value** column and **Variant** column will populate automatically.
4. Click **OK** to save the script.

Adding a Custom Format Section

The custom format section - you can only have one - is added as a separate script before the script containing prescripts, postscripts and the script generated by TimeXtender. To keep things tidy in the Qlik application, any default formatting settings you want to override should be added to this script.

- To add a formatting script, right click **Script**, click **Add Custom Format Section**, type a name for the script in the **Name** box, type your script in the main text area and click **OK**.

Deploying and Executing Qlik Models

Just like data warehouses, Qlik models need to be deployed before you can use them in QlikView/Qlik Sense. Deployment generates the script that can be used in one of the Qlik applications. You can deploy a Qlik model, which deploys all underlying Qlik model translations, or the individual Qlik model translation.

- To deploy a Qlik model or Qlik model translation, right click it and click **Deploy**.

Depending on the Qlik application you are targeting and your setup, there are different ways to deploy and get data into the Qlik application:

- Deploy the Qlik model and copy and paste the Qlik script from TimeXtender to the Qlik application.
- Deploy the Qlik script to a text file and set up the Qlik application to read the file.
- Deploy to Qlik Sense Desktop. TimeXtender can create a Qlik Sense app and load the data.
- Deploy to a Qlik Sense Enterprise server. TimeXtender can create a Qlik Sense app and execute the Qlik Sense app to load data.

Reviewing Scripts and Deploying to Qlik with Copy and Paste

You can always view the latest script generated for a Qlik model or Qlik model translation. To get the model into your Qlik application, you can copy the script and paste it to your Qlik application.

- To review the scripts generated by TimeXtender, right click a Qlik model translation and click **Qlik Sense App script** or **QlikView Scripts**.

Deploying to a Text File

To set up a Qlik model translation to deploy to a text file, follow the steps below.

1. Right click on the Qlik model translation and click **Edit Translation**. The **Edit Translation** window appears.
2. Select **Deploy Qlik script to text file**.
3. Under **Syntax**, make sure that the application you will be using the script with, is selected.
4. Type a path and name for the file in the **File name** box.
5. Click OK.

Deploying to Qlik Sense Desktop

TimeXtender can create a Qlik Sense app and make it available in Qlik Sense Desktop. To create a Qlik Sense app based on a Qlik model, follow the steps below.

1. Deploy the Qlik model translation if you have not already done so.
2. Right click the relevant Qlik model translation and click **Create Qlik Sense App**. The **Create Qlik Sense App** window appears.
3. In **App** name, type a name for the app.
4. Click **Create**. TimeXtender will start Qlik Sense. If an app with the same name already exists, TimeXtender will ask you if you want to update the app.
5. When the app has been successfully created or updated, TimeXtender will ask you if you want to load data.
6. Lastly, TimeXtender will ask you if you would like to close Qlik Sense.

Deploying to Qlik Sense Enterprise

TimeXtender can deploy a Qlik Sense app to Qlik Sense Enterprise and make the app update data each time the project is executed. The setup consists of the following steps:

- Add a Qlik Sense Enterprise server to TimeXtender.
- Set up the Qlik model translation to deploy to the server.
- (Optional) Configure TimeXtender to save data as QVD files and Qlik Sense Enterprise to use these files.

Note: The Qlik Sense Server needs to be able to connect to the SQL Server that contains the data used in the Qlik model. Each staging database or data warehouse database that contain tables used in the Qlik model should have the server name set to a publicly accessible IP. If you use Windows authentication, the user running the Qlik services needs access to database.

To add a Qlik Sense Enterprise server, follow the steps below.

1. In the **Solution Explorer**, right click **Semantic Layer** and click **Add Qlik Sense Server**. The **Add Qlik Sense Server** window appears.

The screenshot shows the 'Add Qlik Sense Server' dialog box. It contains the following fields and options:

- Name:** A text input field.
- Use global Qlik Sense server:** A radio button (unselected) next to a dropdown menu showing '<none>'. This section is disabled.
- Use project settings:** A radio button (selected).
- Protocol:** A dropdown menu showing 'Http'.
- Hostname:** A text input field.
- Port:** A text input field.
- Use proxy authentication:** A radio button (selected).
 - Username:** A text input field.
 - Password:** A text input field.
 - Virtual proxy prefix:** A text input field.
- Use certificate authentication:** A radio button (unselected).
 - Username:** A text input field.
 - Certificate path:** A text input field with a file icon on the right.
 - Certificate password:** A text input field.

At the bottom of the dialog are three buttons: 'Test connection', 'OK', and 'Cancel'.

2. Type a name for the server in the **Name** box.
3. In the **Protocol** list, click the protocol you want to use.
4. Type the your server's hostname in the **Hostname** box.
5. Type the port to connect to in the **Port** box if it is different from the default. The defaults are 4747 if you use certificate authentication, 80 if you use proxy authentication with HTTP and 443 if you use proxy authentication with HTTPS.
6. Click **Use proxy authentication** if you are using the proxy authentication method to authenticate with the Qlik Sense Enterprise server. Type your username in the **Username** box and your password in the **Password** box. Write the prefix from the virtual proxy in Qlik Sense in the **Virtual Proxy Prefix** box.
7. Click **Use certificate authentication** if you are using the certificate authentication method for authenticating with Qlik. Type your username in the **Username** box, enter the path to the certificate in the **Certificate path** box and the associated password in the **Certificate password** box.
8. (Optional) Click **Test Connection** to verify that the connection settings you have specified are working.
9. Click **OK** to add the Qlik Sense Enterprise server.

For more information about adding a virtual proxy to Qlik Sense Enterprise or exporting a certificate for use in TimeXtender, please see the [Qlik Sense documentation](#).

To set up a Qlik Model Translation to deploy to Qlik Sense Enterprise, follow the steps below.

1. Right click on the Qlik model translation and click **Edit Translation**. The **Edit Translation** window appears.
2. Select **Deploy and execute to Qlik Sense server**.
3. Select a server to deploy to in the **Qlik Sense server** list.
4. Click OK.

To set up Qlik Sense Enterprise to be able to save data from TimeXtender as QVD files, follow the steps below:

1. Right click on the **Qlik model** and click on **Edit Qlik Model**. The **Edit Qlik Model** window appears.
2. Select **Store data in QVD files** and click **OK**.
3. If the Qlik model translation has not been deployed before, right click on it and click **Deploy**. The deployment will fail since the QVD files cannot be created, but it will create an app on the Qlik Sense Enterprise server.
4. Open Qlik Sense in your browser, open your Qlik Sense App, and click **Data Load Editor** to go to the data load editor.
5. Click **Create New Connection** and then click on **Folder**. Choose a folder on your machine. In the **Name** box, type "File Storage <Qlik model translation name>", where <Qlik model translation name> is the name of the Qlik Model translation you are deploying. Click **Create**.
6. Go to the Qlik Sense management console and click **Data Connections**. In the list, click on the data connection you just created to open it. In the **Name** box, delete the text in the brackets, so the name is simply "File Storage <Qlik model translation name>". Click **Apply**.
7. You should now be able to deploy your Qlik model translation without errors.

With the setup complete, you are ready to deploy and execute. The deployment generates the script and pushes the app to the server, while execute makes the app load new data.

- To deploy and execute all Qlik models in your project, right click the Qlik root node and click **Deploy and Execute**.
- To deploy and execute a Qlik model, right click the Qlik model and click **Deploy and Execute**.
- To deploy and execute a Qlik model translation, right click the Qlik model translation and click **Deploy and Execute**.

Warning: When a Qlik Sense app is deployed to a server, a data connection is created. This connection uses the names of the staging databases and data warehouses in the project. If you have another project that deploys to the same Qlik Sense Enterprise server, the names for these objects in this project have to be different to avoid overrides.

Data Export

The nature of a data warehouse is to be a means to an end, for instance getting up-to-date sales numbers every morning or the data needed for financial reporting.

TimeXtender supports these scenarios through SSAS Multidimensional cubes or data visualization tools such as QlikView or Qlik Sense. Through the Data Export feature, however, you also have the ability to push parts of - or the entire data warehouse - to another destination, such as an Oracle database or text files.

The feature uses the same external provider concept as the custom data source, which means that we have a framework that makes it possible to add new providers without releasing a new version of TimeXtender.

For each type of destination, you will need a data export provider that can be downloaded and installed through the Custom Component Setup application. Please see the support site for more information: <https://support.timextender.com/hc/en-us/articles/209604866>

At the time of writing, the following data export destinations are supported:

- Oracle
- SQL Server
- Text files

A data export, or data export destination, and the tables and fields it contains, supports a subset of the features found on regular data warehouses.

Table features:

- Selection rules
- Guard table
- Preview table and the query tool
- Tracing
- Pre- and postscripts
- Description

Field features:

- Include in primary key
- Edit name and data type
- Tracing
- Description

Adding a New Data Export

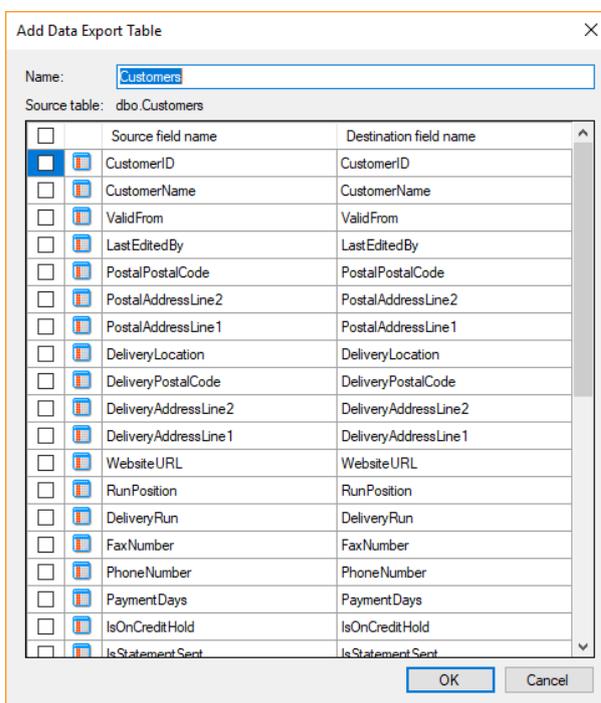
To add a new data export, follow the steps below.

1. In the **Solution Explorer**, right click **Semantic Layer** and click **Add Data Export**.
2. In **Name**, type a name for the data export destination.
3. In the **Provider** list, click the provider you want to use.
4. In the **Setup Property** list, click **Setup Properties**. In the grid below, enter the setup properties required by the selected data export destination.
5. (Optional) Click **Test connection** to test that the connection properties you entered allow you to connect.

Adding a Table to a Data Export

To add a table to a data export, follow the steps below.

1. Drag the table from a data warehouse to **Tables** under the relevant data export. The **Add Data Export Table** window opens.



2. (Optional) In **Data Export Table Name** type a name for the table when used in the data export.
3. Clear the selection for fields you do not want to include in the table on the data export. Click a name in the **Destination Name** to edit the name as it appears on the **Data Export**.
4. Click **OK**.

Adding All Tables and/or Views from a Data Warehouse to a Data Export

To add all tables and/or views from a data warehouse to a data export

- Drag **Tables** or **Views** under a data warehouse – or the entire data warehouse – to the relevant data export.

TimeXtender will add all views and all tables with all fields to the data export with the following exceptions:

- Tables or views that have already been added to the data export will be ignored.
- Tables or views that are not visible on the data warehouse in the currently selected project perspective will not be added.
- Tables or views that are not visible on the data export in the currently selected perspective, will be ignored if you add them again.

Previewing a Table

Once the data export has been executed, the regular Preview Table command can be used to view the content of the table.

Click on **Destination** in the **Instance** list to see the data stored in the data export and **Source** to view the data in the table in the data warehouse.

Deploying and Executing

Before a TimeXtender project is deployed and executed, it is simply a meta data model of your data warehouse. Deployment and execution generates and runs the code for extracting, transferring and loading your data as well as creating any SSAS Multidimensional cubes in the project. During development it can also be a good idea to deploy and execute the project to see if everything works as expected.

Deploying a Project

Deploying a project, or a part of a project, is the process of generating the structure of the staging database and the data warehouse, processing cubes and generating SQL code.

No data is loaded into the staging database or the data warehouse, and no cubes are processed at this time. When you successfully deploy a project, the project is automatically saved in the project repository.

Deployment in TimeXtender is optimized in two ways: It is managed, i.e. objects are deployed after any objects they depend on, and differential, i.e. only the steps that have changed since the last deployment are deployed again.

Executing a Project

Executing a project is the process of loading data into the staging database, the data warehouse, and then processing any SSAS Multidimensional cubes.

Executing a project involves the following steps:

1. **Transferring data:** The process of transferring data from the data source to the raw table of the staging database.
2. **Processing data:** The process of cleansing data; that is, validating the data against the business rules, and moving the validated data to the valid table. Status information is also generated at this point.
3. **Verifying data against checkpoints:** The process of checking the data that is being processed against the checkpoints you have specified. You can specify rules that will end the execution process if not met. This way, you avoid overwriting the data in your data warehouse with non-valuable data.
4. **Moving data:** The process of moving data from a business unit to a data warehouse, or from a data warehouse to a cube.
5. **Processing cubes:** The process of creating dimension hierarchies and retrieving values from the fact tables to populate the cubes with measures, including derived and calculated measures.

TimeXtender supports managed and threaded execution. This means that TimeXtender can execute a project in multiple threads while managing dependencies between objects and optimizing the execution to take the shortest amount of time.

Manual Deployment and Execution

While you are developing and maintaining your project, you may want to deploy or execute the entire project or individual objects to confirm that everything works as expected.

Manual execution is configured in the default execution package, that can be configured just like any other execution package. See [Execution Packages](#) for more information.

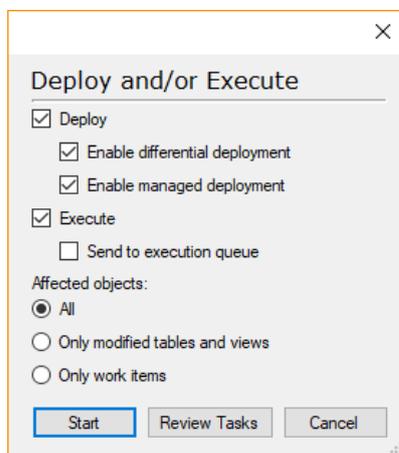
Deploying and/or Executing Individual Objects

Whether you want to deploy, execute or deploy and execute an object, the steps are similar.

1. Right-click the project or the project element you want to deploy and/or execute, and click **Deploy**, **Execute** or **Deploy and Execute**

- OR -

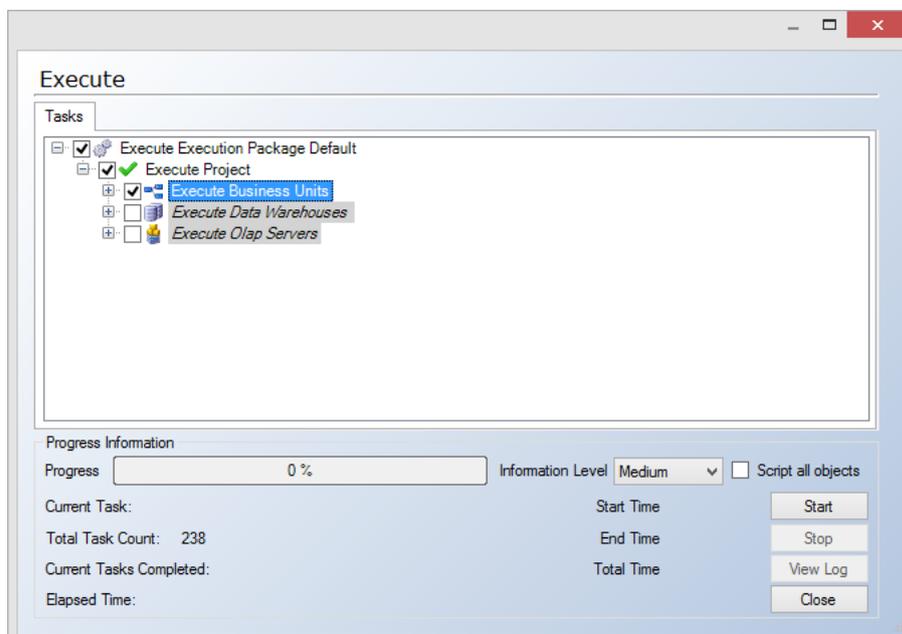
Click the object at press **CTRL + ALT + D** to deploy, **CTRL + ALT + E** to execute and **CTRL + ALT + K** to deploy and execute. The **Deploy and/or Execute** window appears.



The settings in the window depends on how you initialized the window – choosing deploy, execute or deploy and execute – and your default settings for deployment.

2. Select **Deploy** to deploy the selected objects. You have two options for deployment. Their initial setting is based on your project settings.
 - Select **Enable differential deployment** to take advantage of TimeXtender's differential deployment feature that calculates what steps have changed and need to be deployed and selects only those steps for deployment. When differential deployment is disabled, all steps are deployed.
 - Select **Enable managed deployment** to have TimeXtender calculate the correct order of deployment.
3. Select **Execute** to execute the selected objects. The execution settings are governed by the default execution package.
 - Select **Send to Execution Queue** to push the execution to the execution queue so you can continue working on the project while it is executed. For more information, see [Executing Objects with the Execution Queue](#) below.

4. Under **Affected objects**, select which of the objects you selected for deployment and/ or execution you want to deploy and/ or execute. Your options are **All**, **Only modified tables and views** and **Only work items**.
5. Click **Start** to begin the deployment and/ or execution process as soon as TimeXtender is ready or **Preview Tasks** to review the tasks and settings before you start the process.
6. If you use the differential deployment method, it will take TimeXtender a few moments to calculate what steps need to be deployed. If TimeXtender does not find any steps that needs to be deployed, you will be asked if you want to save the project as the deployed version. This has to do with the way the scheduler works. It will execute the last deployed version of a project, i.e. if you want the current version to be the one that is executed by the scheduler, it needs to be marked at such.
7. The **Deploy**, **Execute** or **Deploy and Execute** window opens. If you clicked on **Start** earlier, TimeXtender will begin the deployment and/ or execution process immediately. Otherwise, you can review the tasks before you start the process.



8. Clear the selection for any objects you do not want to deploy, execute or deploy and execute.
9. (Optional) In the **Information level** list, click your preferred level of information during the completion of the tasks. The following options are available:
 - **None:** Displays no progress information.
 - **Low:** Displays current task, the total task count, start time, end time, and total time.
 - **Medium (default):** Displays progress information, name of the current task that is being deployed, number of completed tasks, the total number of tasks that have to be completed, start time, end time, and total time.
 - **High:** Displays all deployment steps in the task window, progress information, name of the current task that is being deployed, number of completed tasks, the

total number of tasks that have to be completed, start time, end time, and total time.

10. (Optional) Select **Script all objects** to make all parts of the SQL script available in the log for debug purposes. This option disables the use of Shared Management Objects to create tables instead of executing "CREATE TABLE" statements against SQL Server since statements executed through Shared Management Objects are not in the log.
11. Click **Start** to deploy, execute or deploy and execute the objects you have chosen. Click **Stop** if you want to halt the processing prematurely.

If there are any errors during deployment or execution, the **Error** window is displayed. Click **Yes** to view the log. The object on which the deployment or execution failed, has an **Error Information** node. Double-click **Error Information** to view an error description.

Customized Code Warnings

TimeXtender will display a warning message if you try to deploy a table with customized code where the table has changed in TimeXtender since you last customized the code. Often, you need to update the customized code when you have made other changes to the table and the purpose of this feature is to help you catch some errors earlier.

TimeXtender displays the warning message when the deployment window opens. Click **OK** to close the message. On the **Customize Code Warning** tab, you can see what tables might have outdated customized code.

If you click **Start** after closing the message, TimeXtender will assume that you have things under control and not display the message again for the same issue.

Reviewing the Execution Log

On each execution, the execution diagram, message and setup is saved to the execution log.

- To view the execution log for an execution package, open the **Execution** tab, right click the execution package and then click **View Execution History Log**.

Resuming a Failed Execution

Executions that fail are a fact of life, but restarting an execution that has failed can be very time consuming if the error occurred two hours into the execution. Therefore, TimeXtender enables you to resume a failed execution from the point of failure.

This is useful since, with managed execution, you cannot always determine in what order TimeXtender will execute tables. This means that if an execution fails, a complete restart of the execution is usually the only way to ensure that everything is executed correctly.

You can also configure TimeXtender to allow some non-essential data sources to fail without failing the entire execution at the same time. See [Allowing a Data Source to Fail](#)

When an execution fails, resume the execution by following the steps below.

1. First, you need to identify the error. On the **Execution** tab, right click the failed execution package, and click **View Execution History Log**. The **Execution Log** window opens.

Start	End	Total Time	Succeede	Gantt Chart	Execution Message	Execution Package
2016-07-06 14:49:29	2016-07-06 14:49:33	00:00:04	False	View	View	View
2016-07-06 14:48:59	2016-07-06 14:49:03	00:00:04	True	View	View	View
2016-07-06 14:48:36	2016-07-06 14:48:40	00:00:04	True	View	View	View
2016-07-06 14:46:34	2016-07-06 14:46:39	00:00:04	True	View	View	View
2016-06-21 10:46:06	2016-06-21 10:46:10	00:00:04	True	View	View	View
2016-04-07 12:39:59	2016-04-07 12:40:10	00:00:10	True	View	View	View
2016-03-01 12:28:13	2016-03-01 12:28:19	00:00:06	True	View	View	View
2016-02-22 11:34:03	2016-02-22 11:34:09	00:00:06	True	View	View	View
2016-02-22 11:28:45	2016-02-22 11:28:51	00:00:06	True	View	View	View

2. Click **View** in the **Execution Message** column next to the failed execution to display the error message. Close the message and the Execution Log window.
3. Second, you need to solve the error you identified in the execution message. Remember to deploy any changed objects as necessary.
4. Now you are ready to resume the execution. Open the **Execution Log** again as described in step 1.
5. Right click the execution you want to resume and click **Resume Execution**. The **Execute** window opens.
6. Click **Start** to begin the execution.
7. Repeat steps 1-6 if the execution package fails again.

Deployment Status Report

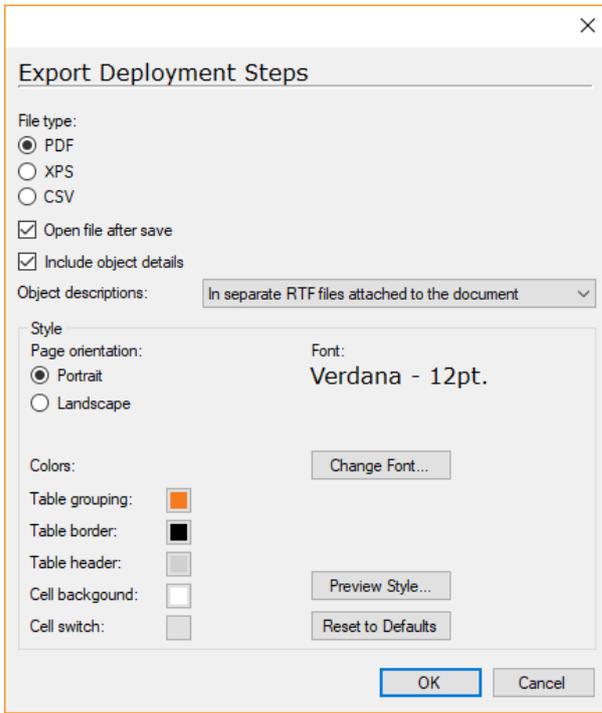
You can generate a deployment status report that contains a list of the objects to need to be deployed. This can be done on the project level, data warehouses, business units and SSAS Multidimensional servers as well as a remote environment.

The process is very similar to [generating documentation](#), however, a deployment status report can also be generated in the CSV file format in addition to PDF and XPS.

Generating a Deployment Status Report

To generate a deployment status report for a part of or your entire project, follow the steps below.

1. Right click on the object you want to create the report for, click **Advanced** and click **Export Deployment Steps**. The option is available on the project level, data warehouses, business units, SSAS Multidimensional servers, data exports and Qlik models. The **Export Deployment Steps** window appears.



2. Under **File type**, click the file format you want to use. If you choose CSV,
3. Select **Open file after save** to view the document when TimeXtender has generated it. TimeXtender shows the document using external viewers that needs to be present on the machine.
4. Select **Include object details** to include details of the objects that the steps that need to be deployed are part of.
5. In the **Object descriptions** list, you can choose how you want to use include the object descriptions. You have the following options:
 1. **In separate RTF files attached to the document:** The descriptions are attached to the document as RTF files. This option is useful if you have used rich text formatting or added pictures to your descriptions, but only available for the PDF file format.

Table:	AggrTabRegions	Custom Description in attached file: 2.rtf
Icon		
Table Type		Aggregation
Allows Nulls		Yes (As Project)
Null Check Approach		Field Based (As Project)
Primary Key Behavior		As Project
Field Count		6
Fields		[Name], [CountryRegionCode], [DW_Id], [DW_Batch], [DW_SourceCode], [DW_TimeStamp]
Index Count		0
Aggregations		Sum([CountryRegionCode])

2. **Only text placed in the document:** The descriptions are included in the documentation as plain text.

Table: AggrTabRegions	
Icon	
Table Type	Aggregation
Allows Nulls	Yes (As Project)
Null Check Approach	Field Based (As Project)
Primary Key Behavior	As Project
Field Count	6
Fields	[Name], [CountryRegionCode], [DW_Id], [DW_Batch], [DW_SourceCode], [DW_TimeStamp]
Index Count	0
Aggregations	Sum([CountryRegionCode])
Description	This is an example description

3. **No descriptions:** The descriptions will not be included in the generated documentation.
6. Under **Style**, you can configure the look of the status report. The settings are saved on a per-user basis and used for this as well as the files generated by [the documentation feature](#).
7. Under **Page Orientation**, select the page orientation you want the documentation to use.
8. Click the color preview next to the different color details to choose a color.
9. Click **Change font...** to choose a font for the documentation. a preview is displayed under **Font**.
10. Click **Preview Style...** to generate and open a sample file with the colors you have chosen.
11. Click **Reset to Defaults** to reset the style settings to their defaults
12. Click **OK**. Wait while the steps are calculated. In the window that then appears, choose a file name and location for the report and click **Save**. The default name is the project name followed by "deployment steps" and a timestamp.

Please note that you will need to have working connections to the relevant databases when you generate the status report. Otherwise, TimeXtender won't be able to calculate the differences that indicate that a deployment is necessary.

Generate Deployment Status Report on Remote Environment

In addition to the local project, you can also generate a deployment status report for a project on a remote environment in a multiple environments setup. The report will contain information on the last available version on the environment.

- To generate documentation for a project on a remote environment, right click on the project, click on Multiple Environment Transfer, right click the remote environment and click **Export Deployment Status**.

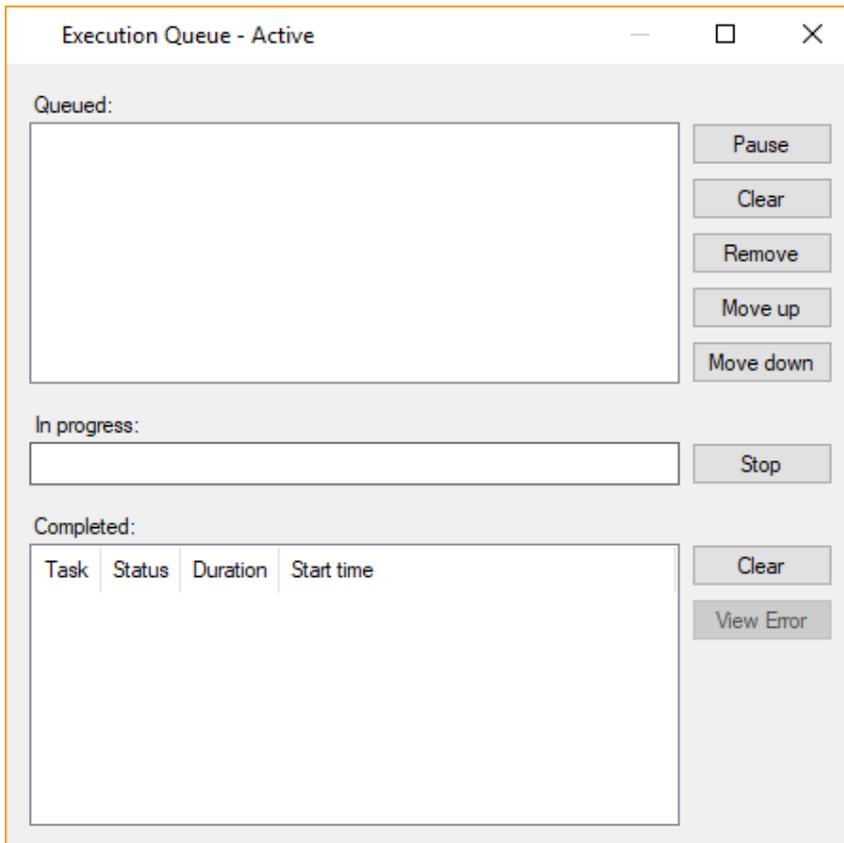
Executing objects with the Execution Queue

The Execution Queue enables you to continue working while tables or you entire project is executed in the background.

Opening the Execution Queue window

To open the Execution Queue window

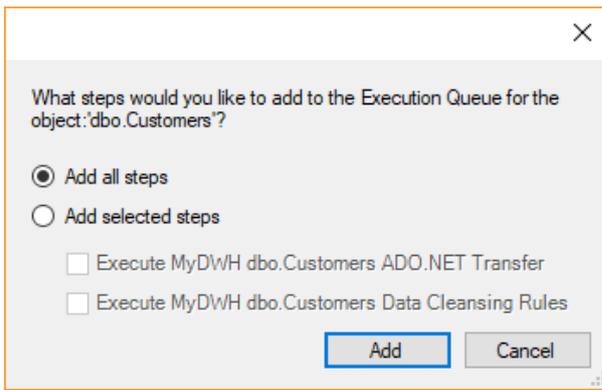
- On the **Tools** menu, click **Execution Queue**.



Adding an object to the Execution Queue

Adding an object to the Execution Queue is a simple drag-and-drop operation.

1. Drag-and-drop a table, a perspective, a data warehouse, a business unit, an execution package or another executable object to the Execution Queue window. A window appears to let you choose which execution steps from the object to add to the queue.



2. Select **Add all steps** or **Add selected steps** and choose which steps you would like to add to the queue. Click **Add** to add the object to the queue.
3. The object is now queued in the Execution Queue. If there is no other items in the queue, the object will be moved to **In Progress** and begin executing immediately.

Pausing and running the queue

The Execution Queued mode can be toggled between running and pause using the button in the top right corner of the window. When the queue is running, the button is called **Pause**. Clicking the button prevents further objects from being executed and changes the button text to **Resume**. Pressing **Pause** does not stop an object that is currently in progress. Pressing the **Resume** button resumes executing of the queue.

Moving and Removing Queued Items

The **Queued** list shows the items waiting to be executed.

The queued objects can be moved up and down in the list by selecting the item and using the **Move up** and **Move down** buttons. The top item in the list is the next to be executed.

An object can be removed from list by selecting it and clicking **Remove**. Clicking **Clear** removes all items from the list.

Stopping Current Execution

In Progress shows the object currently being executed. Pressing **Stop** halts the execution of the object and pauses execution of the queue.

Removing executed items and viewing errors

The **Completed** list shows the objects that have been executed. It lists the **Status** of the individual items, the **Duration** and the **Start Time**. The items can have one of four statuses:

- **Success**: The object was executed without errors.
- **Failed**: The execution was ended prematurely by an error.
- **Stopped**: The execution was stopped by the user before it completed.

You can view error messages for failed objects by selecting the object in the list and clicking **View Error**. This brings up a message box displaying the error message.

Closing the Execution Queue Window

You can close the Execution Queue window by clicking the X in the top right corner.

Closing the Execution Queue window or closing the entire project does not stop or pause the execution of the queued objects. It only hides the window, while the Execution Queue will continue working in the background. You can open the Execution Queue window again to review the status of the objects in the queue or to add more objects to the queue.

When you close TimeXtender, the Execution Queue will be stopped along with the rest of the application.

Scheduled Execution

When your project is ready, you would typically like the entire project, or part of it, to be executed according to a schedule. One common scenario is to execute the project every night to ensure that the business users have updated numbers in the morning.

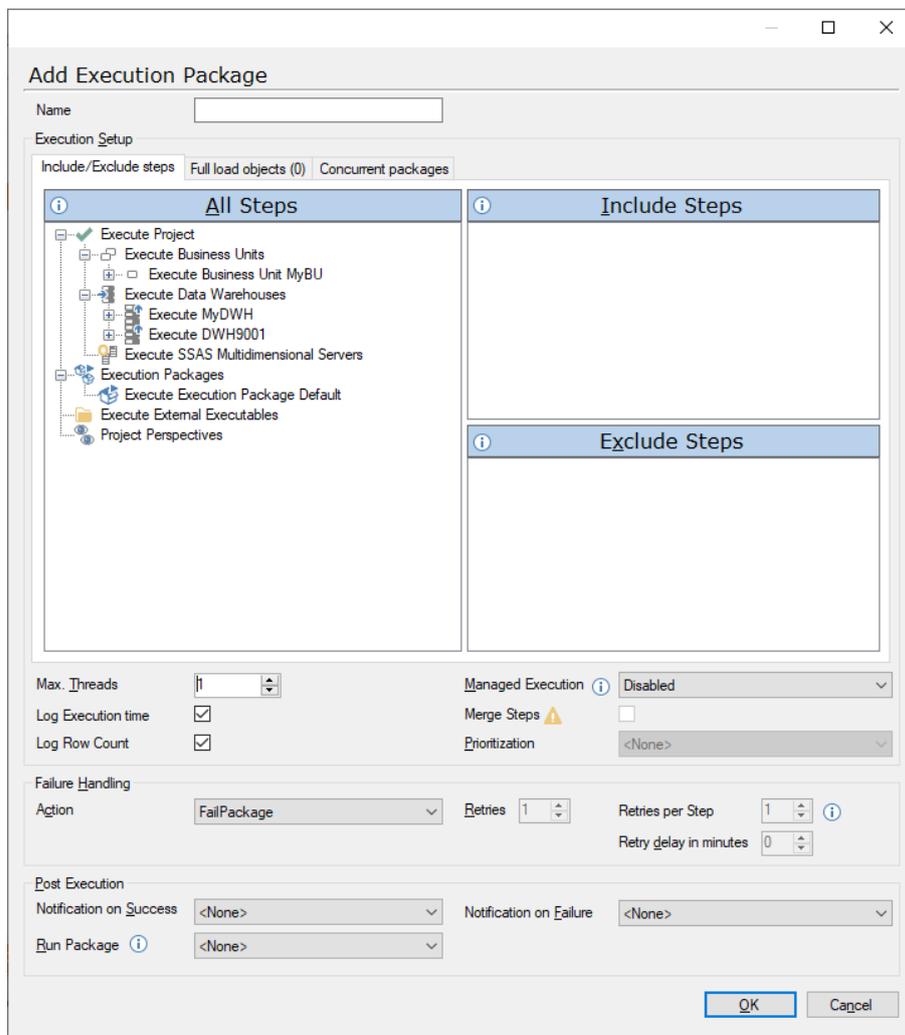
No matter what your needs are, the answer in TimeXtender is the execution package. An execution package contains tables, cubes and other objects that will be executed when the package is run. You can schedule the package to be run at specific times and set up notifications to e.g. alert you if a scheduled execution fails. You can also set up conditions that need to be fulfilled for an execution package to run.

Note: An execution package will only execute the objects in the package, not deploy them. If changes have been made to the project, but have not been successfully deployed, the scheduled execution package will most likely fail.

Adding an Execution Package

An execution package determines which objects in a project will be executed and how. To create an execution package, follow the steps below.

1. On the **Execution** tab, right-click **Execution Packages** and then click **Add Execution Package**. The **Add Execution Package** window appears.



2. Type a **Name** for the execution package.
3. In the **Include/Exclude Steps** tab, choose the steps you want to include in or exclude from the execution package by dragging objects from **All Steps** to **Include Steps** or **Exclude Steps**, respectively. For instance, simply drag the **Execute Project** step to **Included Steps** if you want to execute the entire project when the package is executed. If you want to exclude the entire project except one or more steps, simply drag those steps to **Exclude Steps**. Right click an object and click **Remove Step** to remove a step from the **Include Steps** or **Exclude Steps**.
4. (Optional) On the **Full load tables** tab, drag any incrementally loaded tables, you want to have full loaded in this execution package, from **All Tables** to **Full Load Tables**. You can also drag business units, data warehouses, SSAS Multidimensional servers or the entire project.
5. (Optional) On the **Concurrent packages** tab, select the other packages that can execute concurrently with the package you are adding. The scheduler service will not run two packages from the same project at the same time unless you explicitly allow it in this way.

Note: When you set package "A" to be able to execute concurrently with package "B", package "B" will also be set to be able to execute concurrently with package "A". In other words, if you want to be able to execute "A" when "B" is already running, you cannot use this functionality to prevent "B" from being run when "A" is already running.

6. Enter the maximum number of steps that can run in parallel during execution in **Max. Threads**. The optimal number of threads depend on server resources. Too few threads means slower than necessary execution times, while too many threads can cause the server to become slow or unresponsive.
7. Clear the **Log Execution time** and/ or the **Log Row Count** check boxes if you do not want to log this information.
8. Select a setting for **Managed Execution**. You have the following options:
 - **Disabled:** Managed execution is disabled. Objects will be executed in the order they are listed in the tree.

Warning: If you disable managed execution, tables are executed in the order in which they appear in the tree. To avoid errors during execution, you must ensure that tables are executed in a logical order. For example, an Order table must be executed before the related Order Detail table. Tables can be moved up and down the tree using drag-and-drop or the keyboard combination Alt+Up/ Down.

- **Execution Number:** When more than one object is ready to be executed, TimeXtender prioritizes the objects based on their position in the tree from top to bottom.
 - **Classification:** When more than one object is ready to be executed, TimeXtender prioritizes the objects based on their table classification. The order will be "Fact Table – Large", "Fact Table", "Dimension Table – Large", "Dimension Table". If two tables have the same classification TimeXtender will use the execution number as the secondary criteria.
 - **Execution Time:** When more than one object is ready to be executed, TimeXtender prioritizes the objects based on their average execution time so that the object with the longest execution time is executed first. If two tables have the same execution time (e.g. in case of new objects), TimeXtender will use the execution number as the secondary criteria. When execution time of the objects in the project are known, this option will result in the shortest execution time in most cases.
9. Check **Merge Steps** if you want to treat all individual sub-steps of the chosen steps as one big collection. This can speed up execution.
 10. (Optional) In the **Prioritizer** list, click the prioritization you want the execution package to use. For more information on prioritization, see [Adding a Prioritization](#).

11. Under **Failure Handling**, select what **Action** TimeXtender should perform if the execution fails. You have the following options:
 - **Fail Package**: When a step fails, the execution is stopped and the package is declared failed.
 - **Retry Step**: When a step fails, the step will be retried until the maximum number of retries for the entire package or the individual step is reached. Enter the maximum number of retries allowed for the package in the **Retries** box and the maximum number of retries allowed for an individual package in **Retries per Step**. Enter the amount of time to wait between retries in **Retry delay in minutes**.
12. Under **Post Execution**, select a Notification on Success and a Notification on Failure. You have to create a notification before it is available from the list. See [Adding Notifications](#) below.
13. If you want to run a package after the execution, select the package in **Run Package**.
14. Click **OK** to add the execution package.

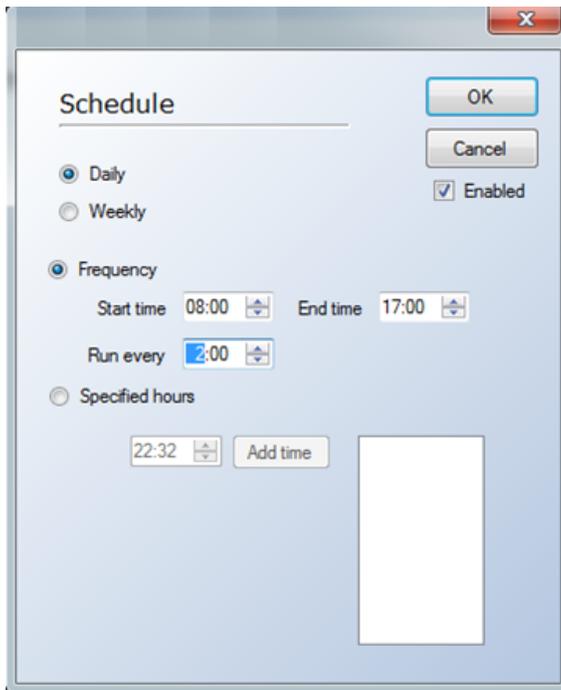
Adding an Execution Schedule

When you specify an execution schedule, the execution process is started automatically at the specified time.

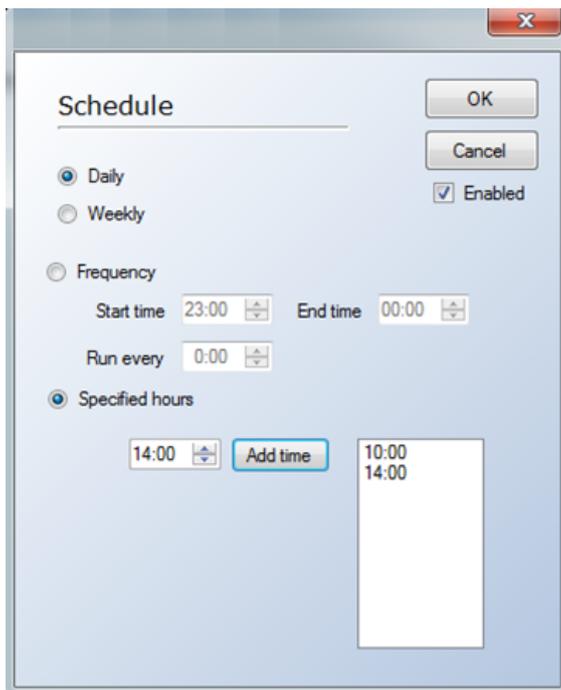
Note: For each project, make sure to schedule packages so that start times are at least two minutes apart. The scheduler service checks for packages to execute every two minutes. If more than one package is scheduled to start within two minutes, one of them will be executed, but it might not be the one you expect. In addition to this, it will skip a scheduled execution if a scheduled execution of a package in the same project is running and the two packages are not configured to be able to run concurrently.

To add an execution schedule, follow the steps below.

1. On the **Execution** tab, expand **Execution Packages**, right-click the relevant execution package, and click **Add Schedule**. The **Schedule** window appears.
2. You have three different options for specifying the schedule:
 - To set up multiple daily executions in a specific time frame, click **Daily** and then click **Frequency**. In the **Start time** field, enter the start time of the time interval. In the **End time** field, enter the end time of the interval. In the **Run every** field, specify the number of hours and minutes between each project execution.

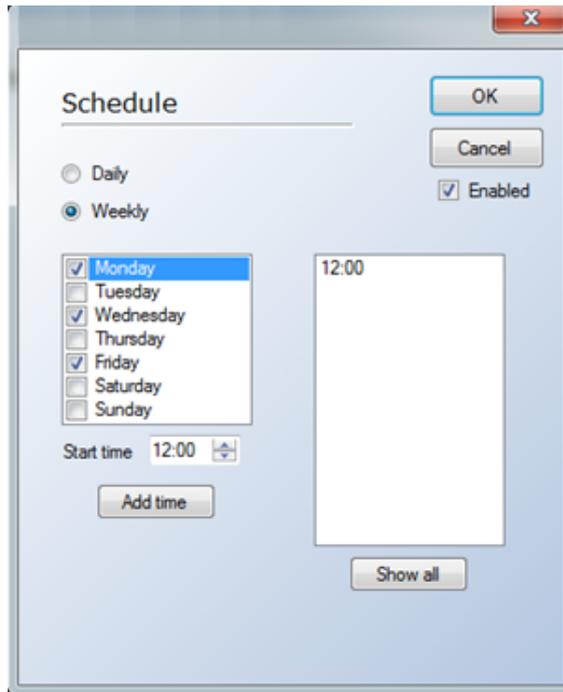


- To set up one or more daily executions on specified hours, click **Daily** and then click **Specified Hours**. Enter the exact time for the execution to run and then click **Add Time**.



- To set up weekly executions, click **Weekly**. Select the day(s) when the project should execute, enter the **Start time** for the execution, and then click **Add Time** to add the time to the schedule. Repeat this step for each day and time that you want to execute the project. A project can be executed several times a day and several times during the week. To view the entire weekly schedule, click **Show**

All.



3. Select **Enabled** to activate the schedule, and then click **OK**.

Note: You can get an overview of the previous executions of an object by right-clicking on it and clicking on **View Execution overview Log**. Additionally, all events that happen during execution are registered in the Windows Event Log.

Adding a Notification

Notifications can be used to alert specified individuals when the execution package was successfully run or in case something caused it to fail. Notifications are most commonly set up as email alerts, but can be saved to the Event Log as well.

1. On the **Execution** tab, right-click **Notifications**, and click **Add Notification**. The **Notification** window appears.

2. Enter a **Name** for the notification and select the **Type** the type of notification you want to create. You have the following options:

Option	Description
--------	-------------

Mail	Creates an email notification
EventLog	Writes a notification to the event log
Both	Creates both an email notification and writes to the event log

3. Enter the information for the SMTP server you want to use under **Mail Server**.
4. Enter the email addresses of the recipients and a **Subject** under **Mail Recipient**. In the subject, you can use the following variables:
 - **%Project%**: The name of the project .
 - **%Status%**: The status of the execution (Success / Fail).
 - **%ExecutionPackage%**: The name of the execution package.
5. Click **OK**.

The notification can now be selected when you create an execution package.

Adding a Prioritization

You can choose what objects to prioritize during a managed execution. This is useful if you, for instance, only have a small timeslot for extracting data from a source, or if you would like to have a certain cube ready for the users as early as possible during execution.

To set up a new prioritization for use in an execution package, follow the steps below.

1. On the **Execution** tab, right click **Prioritizations** and click **Add Prioritization**. The **Add Prioritization** window opens.
2. In **Name**, type a name for the prioritization.
3. Drag and drop an object from the **Available Objects** tree to **Selected Objects** to add this object to prioritized objects.
4. Click an object in the **Selected Objects** list to have the execution steps for that object displayed under **Object Settings**. Select or clear the individual steps under **Selected steps** to configure what steps will be prioritized in the execution. For instance, you might only want to prioritize the transfer of data from a specific table because your priority is to have the transfer completed as soon as possible.
5. Select **Blocking** if you want for the execution to halt all other execution tasks until the selected steps for the selected object has been completed.
6. Click **OK** to close the window and create the new prioritization.

Adding a Usage condition

If you want to execute an execution package only under certain conditions, you can add a usage condition to the execution package. For instance, if you use multiple environments, you could have an execution package execute only in your production environment. To add a usage condition to an execution package, follow the steps below.

1. Add the project variable you want to use in your usage condition. See [Adding a Project Variable](#).
2. On the **Execution** tab, expand **Execution Packages**, right-click the relevant execution package, and click **Add Usage Condition**. The **Usage Condition** pane appears:

The screenshot shows a window titled "Usage Conditions". Inside, there is a tree view with "Project Variables" and "Environment" listed. Below the tree view, there are three input fields: "Operator" with a dropdown menu showing "Equal", "Comparer" with a dropdown menu showing "string", and "Value" with an empty text box. An "Add" button is located at the bottom right of the pane.

3. The available project variables are listed in the window. Click the variable you want to use. In the example above, the "environment" variable has been selected.
4. In the **Operator** list, click the operator you want to use.
5. In the **Comparer** list, click the data type you want to use when comparing the variable to the value.
6. In the **Value** box, type the value you want to compare the variable against.
7. Click **Add** to add the usage condition.

If you try to manually execute a package with a usage condition that evaluates to "true", a warning message will pop up. The same message will be written to the log if you try to execute the package in a scheduled execution.

Incremental Loading

Incremental loading facilitates faster load times by only loading new data into the data warehouse and staging databases. This is especially useful when the volume of data in the data source causes scheduled execution to take too long.

Automatic Incremental Load from an ODX

To load data incrementally from the ODX data storage and into a data warehouse, you don't have to do anything. Per default, any tables that are incrementally loaded into the ODX will automatically be incrementally loaded into the data warehouse.

Disable Automatic Incremental LOAD

To disable automatic incremental load and force full load for a table

- Right click the table, click **Data extraction** and select **Full load**

Enable Handling of Records Deleted in the Source

If records can be deleted from the source system, it can be useful to have them deleted from the data warehouse as well.

To enable delete handling, follow the steps below.

1. Right click the table, click **Table Settings** and click the **Data extraction** tab.
2. Under **Incremental load**, click on the option that represents how you would like to handle records deleted in the source table. You have the following options:
 - **Don't handle deletes**
 - **Use hard deletes:** Records are deleted in the data warehouse when they are deleted in the source.
 - **Use soft deletes:** Records are marked as deleted in the data warehouse when they are deleted in the source. The value of the system field "IsTombstone" can be used to tell deleted rows from valid rows.

TimeXtender uses the primary key to tell what records have been deleted in the source. This requires the primary key defined in the data warehouse to match the key defined in the ODX when you have delete handling enabled.

To see if a field is part of the primary key in the ODX

- Expand the field in the tree and note if the mapping icon has a key like this:



Warning: If the ODX table does not have a primary key and/or delete handling enabled, delete handling will work, but it will be very slow. On the ODX source table, always set a primary key and enabled delete handling if possible.

How Automatic Incremental Loading Works

The basic logic behind the automation is as follows: The ODX stamps each load from the data source with a batch number. When data is transferred to the data warehouse, the system keeps track of which batches are transferred. Incremental load, then, is simply transferring all batches with a higher batch number than the latest batch transferred to the data warehouse.

Schema Drift Triggers a Full Load

When a new column is added to the table, or a data type is changed, in the source, it will trigger a full load of the table. Since the schema has changed, the new data structure no longer matches the old data.

Delete Handling Triggers Full Load for Tables without Delete Handling Enabled

If you map multiple incrementally loaded tables from the ODX into the same table in the data warehouse, they should all have the same setting for delete handling. Otherwise, the tables without delete handling enabled will be full-loaded.

This means that it can make sense to enable delete handling on the ODX even on tables where records are never deleted. If you do not, a comparatively small Customers table can, for instance, prevent the incremental load of a much larger Invoices table.

Note: This relates to the delete handling setting in the ODX for data extraction from data sources to ODX data storage. The setting for delete handling on tables in the data warehouse applies to delete handling when data is loaded into the data warehouse.

Delete Handling Requires Matching Primary Keys

As stated above, if you enable delete handling, the primary key defined in the data warehouse must match the key defined in the ODX. Otherwise, TimeXtender cannot match the records in the source with the records in the data warehouse to calculate what records to delete.

Source Tables from other Data Warehouses are Ignored

Automatic incremental load does not work with source tables from other data warehouses or staging databases. Incremental load from these tables use the logic described in the [Enabling Incremental Loading From Staging Database or Data Warehouse](#) section below, i.e. you have to add an incremental selection rule and explicitly enable incremental load.

Incremental load from Business Unit or Data Warehouse

Enabling Incremental Loading

To use incremental loading on a table, the table must contain a field that represents new data. This could be an identifier field, an entry number or a date. In addition to that, the

table must have a primary key defined.

To include a field in the primary key for the table

- Right-click a field and click **Include in Primary Key**.

To enable incremental loading, follow the steps below.

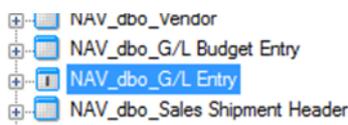
1. Right click the table for which you want to enable incremental loading, click **Add Incremental Selection Rule**, click **Yes** in the message that appears and go to step 7
- OR -

Right click the table for which you want to enable incremental loading, click **Table Settings** and continue to step 2.

2. Click the **Data extraction** tab and select **Incremental load**.
3. Under **Incremental load**, click on the option that represents how you would like to handle records deleted in the source table. You have the following options:
 - **Don't handle deletes**
 - **Use hard deletes**
 - **Use soft deletes**
4. Click **Keep field values up-to-date** to recalculate conditional lookup field and super-natural key field values on tables that are already on the data warehouse when the values being looked up have changed.

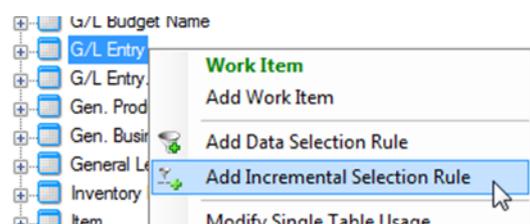
Note: This option cannot be used when the table is on a Azure Synapse Analytics data warehouse.

5. If an error icon appears next to the setting, it means that another setting needs to be changed to enable source based incremental loading. Move you mouse over the error icon to see the error message. Once any settings have been changed as required, click **OK**. The table icon will now be overlaid with an "I" to make it easy for you to identify it as an incrementally loaded table.

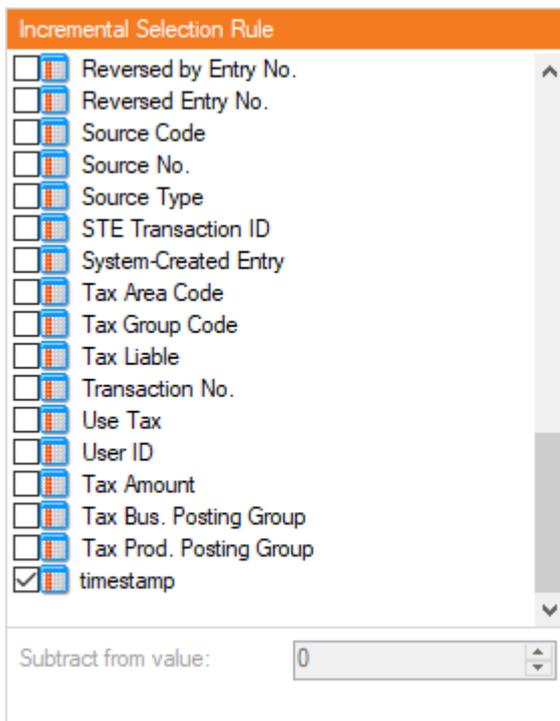


6. If the table belongs to the staging database, right click the corresponding source table under **Data Sources** and click **Add Incremental Selection Rule**.
- OR -

If the table belongs to the data warehouse, right click the table and click **Add Incremental Selection Rule**.



7. The **Incremental Selection Rules** pane appears. Select the fields identifying which records have been added or changed since the last incremental load.



Incremental Selection Rule

<input type="checkbox"/>	Reversed by Entry No.
<input type="checkbox"/>	Reversed Entry No.
<input type="checkbox"/>	Source Code
<input type="checkbox"/>	Source No.
<input type="checkbox"/>	Source Type
<input type="checkbox"/>	STE Transaction ID
<input type="checkbox"/>	System-Created Entry
<input type="checkbox"/>	Tax Area Code
<input type="checkbox"/>	Tax Group Code
<input type="checkbox"/>	Tax Liable
<input type="checkbox"/>	Transaction No.
<input type="checkbox"/>	Use Tax
<input type="checkbox"/>	User ID
<input type="checkbox"/>	Tax Amount
<input type="checkbox"/>	Tax Bus. Posting Group
<input type="checkbox"/>	Tax Prod. Posting Group
<input checked="" type="checkbox"/>	timestamp

Subtract from value:

In the **Subtract from value**, you can enter an integer (from numeric fields) or an amount of time (for date fields) to subtract from the value of the selected field before determining what records are new. This allows you to create an overlap in the records loaded to make sure any recent changes in existing records are reflected in the database.

The fields you choose should ideally be fields that are generated by the system and incremented sequentially when new records are added. Here are some recommended fields for the Microsoft Dynamics ERP systems:

- **Dynamics Business Central (NAV):** timestamp
- **Dynamics AX:** Modified_Datetime
- **Dynamics GP:** DEX_ROW_ID

8. Repeat steps 1-7 for all tables you want to enable incremental loading on.
9. Deploy and execute the table(s). TimeXtender will begin the first full load of the tables with source based incremental loading now enabled. The necessary incremental tables will be automatically added to the staging database and populated with the latest incremental values. The next time the table is executed, TimeXtender will query these tables to determine the last record that was previously loaded and will only extract the data from the data source that was added after the last execution.

Enabling Incremental Loading for Multiple Tables

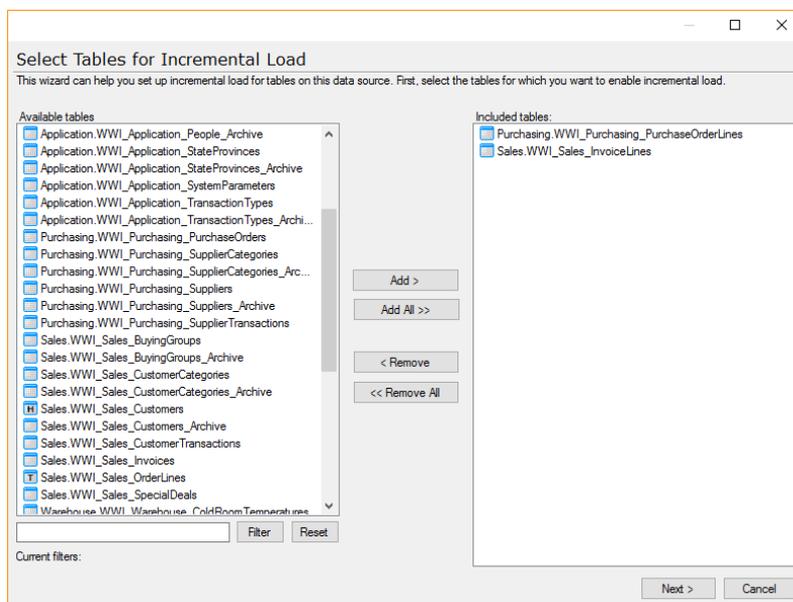
With the Set Up Incremental Load wizard, you can automate the set-up of incremental load to enable it for a number of tables in one go. You still have to choose primary keys and

what fields to use for the incremental load rule, but your selections are applied automatically. In addition to enabling incremental load on the tables you select, any settings that are incompatible with incremental load will also be changed.

Note: When you run the wizard on a data warehouse, additional help will be available in the form of button, **Auto Suggest**. It will suggest primary keys based on the primary keys set on the staging database. For selection rules, the suggestion depends on a rule based on a custom field created on the table and mapped to the 'DW_TimeStamp' system field of the source table. This rule is displayed in the wizard as 'IncrementalTimeStamp' and has a robot icon.

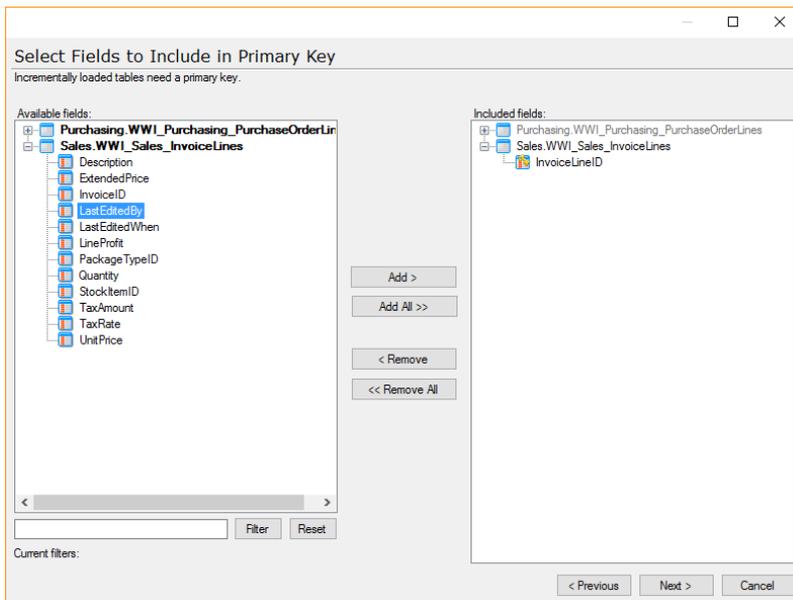
To use the Set Up Incremental Load wizard, follow the steps below.

1. Right click a data source, staging database or data warehouse, click **Automate** and click **Set Up Incremental Load**.
2. The wizard appears.



In the **Available tables** list, double-click the tables you want to enable incremental load for. You can also click a table and then click **Add** to add an individual table. To add all visible tables, click **Add all**. Use the filter below the list to filter the list on table name. The following wildcards are supported:

- **%**: Any string of zero or more characters.
 - **_**: Any single character.
 - **[]**: Any single character within the specified range ([a-f]) or set ([abcdef]).
 - **[^]**: Any single character not within the specified range ([^a-f]) or set ([^abcdef]).
3. Click **Next**. All incrementally loaded tables need to have a primary key, which you can add on this page. Any tables that do not have a primary key when you proceed to the next page, will be removed from the wizard.

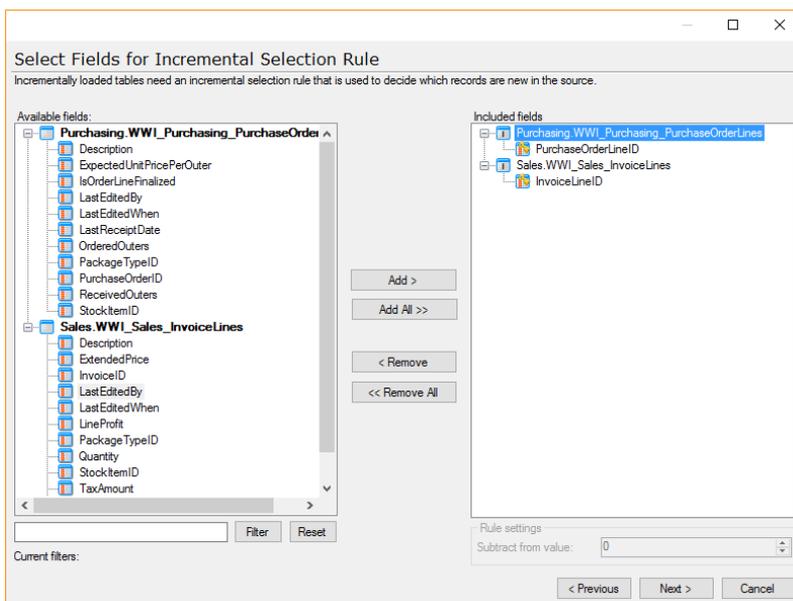


In the **Available fields** list, double-click the fields you want to use as primary keys on the tables. You can also click a field and then click **Add** to add an individual field.

The tables in the **Available fields** list that have at least one primary key field will be shown in bold.

In the **Included fields** list, any fields that are already primary keys on the tables you selected in the previous step are listed in gray. You cannot remove existing primary keys on this page, only add new ones.

4. Click **Next**. All incrementally loaded tables need an incremental selection rule. You select those on this page.

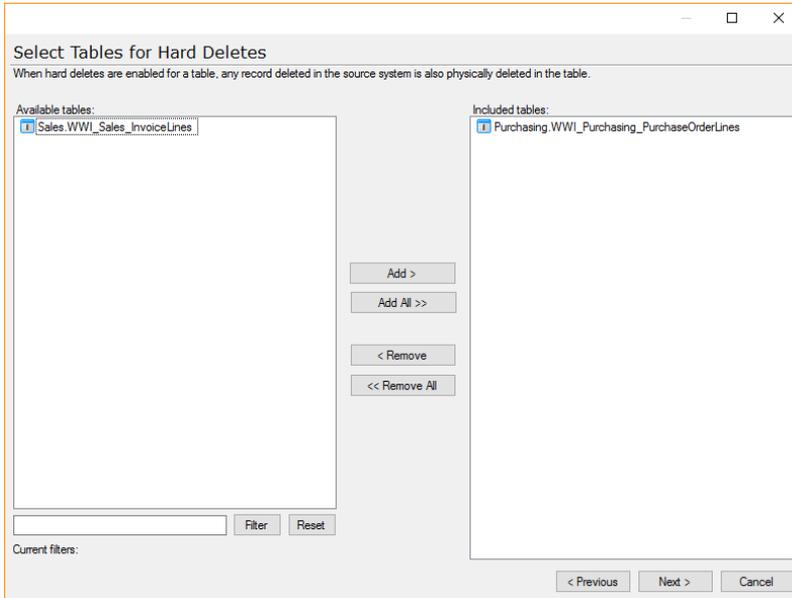


Select the fields you want to use as primary keys on the tables. Under **Rule Settings**, you can enter a subtraction value for a field if you want the incremental load to

overlap the previous load.

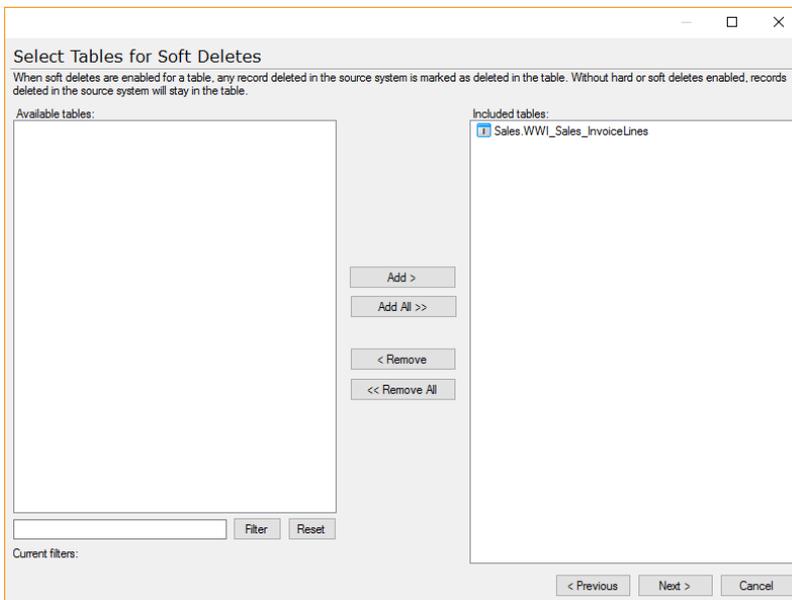
The tables in the **Available fields** list that have a field selected for incremental selection rule will be shown in bold.

5. Click **Next**. TimeXtender can handle records deleted in the data source for you. On tables with hard delete enabled, records deleted in the data source is also deleted from the data warehouse.



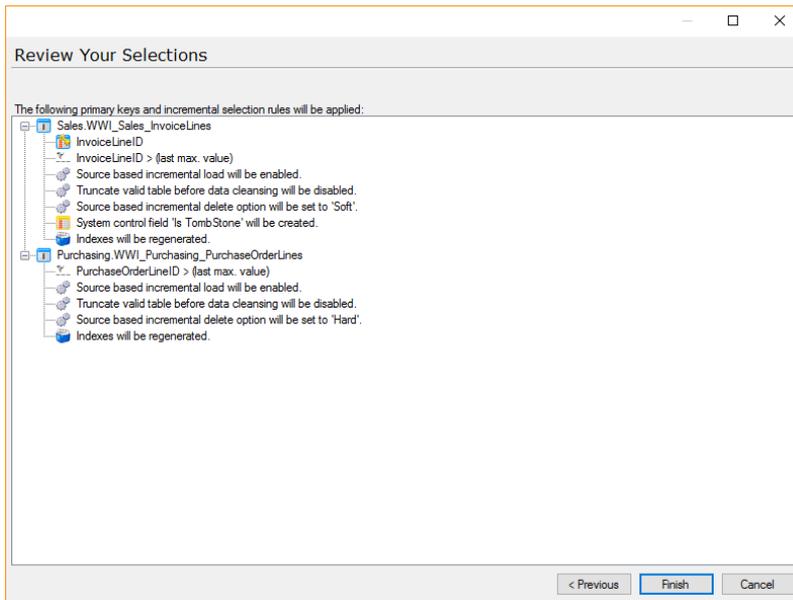
Select the tables you want to enable hard deletes for.

6. Click **Next**.



Select tables you want to enable soft deletes for. The tables you do not select for either hard and soft deletes, will not have any delete handling enabled.

7. Click **Next**.



Click **Previous** to go back to an earlier page and adjust the selections there or click **Finish** to apply the changes if they are correct.

Multiple Environments

With multiple environments, you can have a dedicated development environment with automatic transfer of the latest version of your project to the production environment. This ensures that the production environment is always online and available for end-users.

A dedicated development environment enables you to work within non-production environments. This is useful when an organization needs to ensure that the production environment is always available for end-users. For example, the organization could have an environment called "Development" where changes are made, dimensions are updated, and measures are created. Once these modifications are tested, they can be transferred to the live production environment directly from TimeXtender.

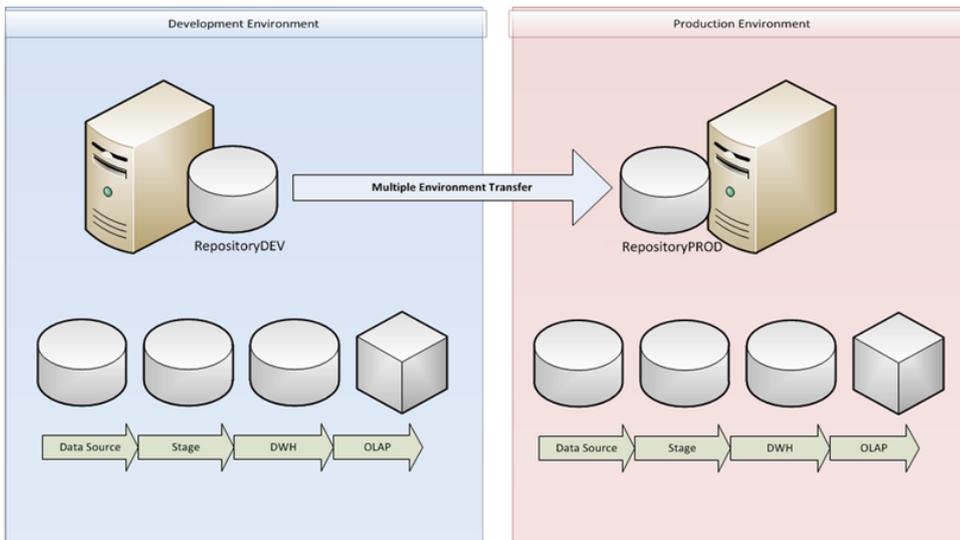
Prerequisites

Before setting up development environments, ensure that the following prerequisites are met:

- All servers used in the development and production environments must have the same version of TimeXtender installed.
- Ensure that TimeXtender service is installed and started on the server(s) you want to deploy on. Detailed instructions are provided below.
- Ensure that a project repository has been created on all the servers that are used in the production environment and development environment.
- The user account(s) that will be used to set up the multiple environments **may** need to have Read permissions to the Event Log. This can be set up in the Registry Editor under the "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\eventlog" node. You can right-click the "eventlog" node, select **Permissions**, and add the users that will utilize Multiple Environment Deployment and assign them "Read" permissions.

Setting Up a Development Environment

The following example shows multiple environments using a single **Development** server and a single **Production** server.

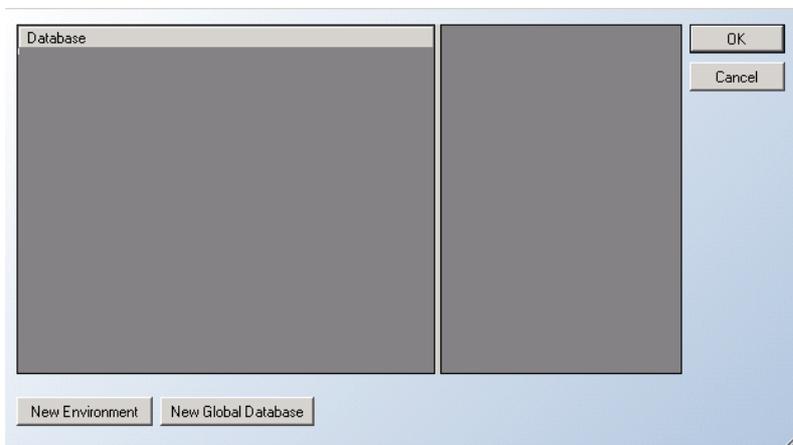


Note: You can set up as many environments as you need. Setup of additional environments follows the same steps as listed below.

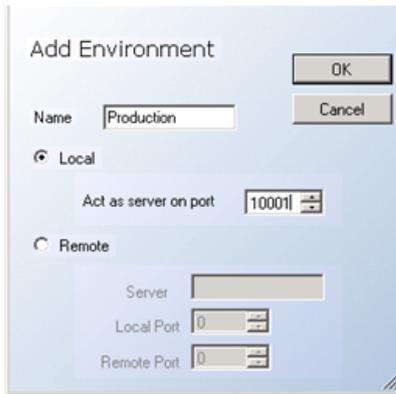
Setting Up the Production Server

The first step is to set up the **Production** environment on the production server.

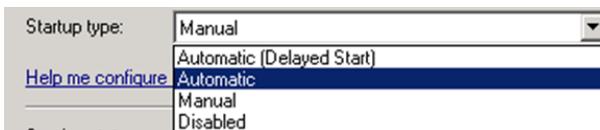
1. Log on to the production server and open TimeXtender.
2. On the **Tools** menu, click **Environment Properties**
3. The **Environment Properties** window will open. Click **New Environment**.



4. The **Add Environment** window will open:

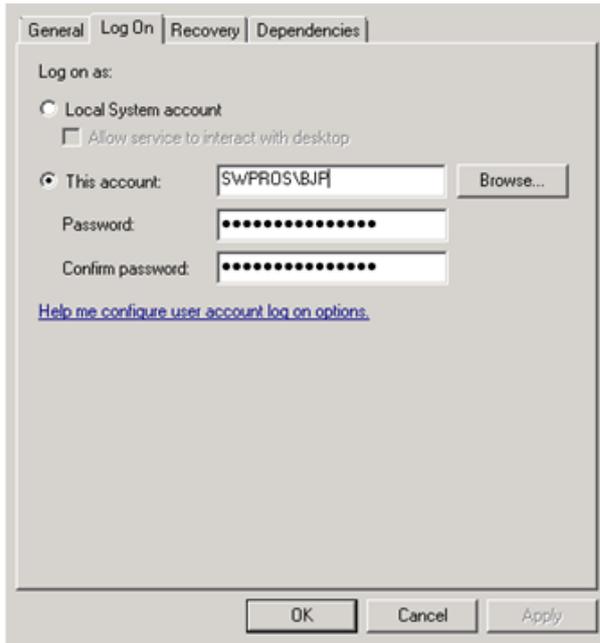


- Type a **Name**. In this example, the name is “Production”.
 - Select **Local**, as deployment will only be done into this environment and not from it.
 - Enter a port to use in the **Act As Server On Port** box. Make sure the port is free to avoid any conflicts on the network.
5. Close TimeXtender.
 6. Next you need to make sure that the TimeXtender server service is correctly set up. Click/right-click **Start**, click **Run**, type **Services.msc** and click **OK**. Locate the server service in the list. It will be named **TimeXtender Server [version]**.
 7. Right click the service and click **Properties**.
 8. On the **General** tab, in the **Startup type** list, click **Automatic**.



9. On the **Log On** tab, click **This account**. In **This Account**, enter the account that was used for setting up the production environment, i.e. the account you are logged in with. Type the password for the account in the **Password** and **Confirm password**

boxes. Click **OK**.

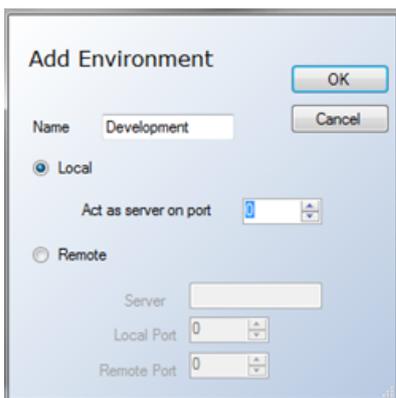


10. Right click the service in the list and click **Start**.
11. Log off from the production server.

Setting Up the Development Server

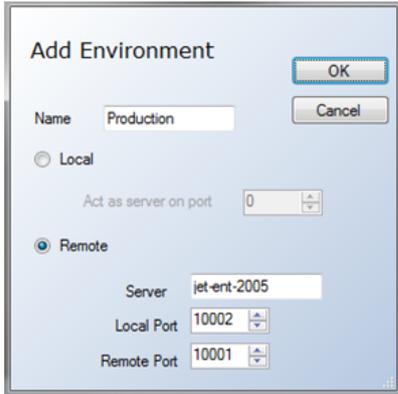
The next step is to set up the development environment on the development server.

1. Log on to the development server and open TimeXtender.
2. On the **Tools** menu, click **Environment Properties**
3. The Environment Properties window will open. Click **New Environment** to create the development environment.
4. The **Add Environment** window will open:



- Type a **Name**. In this example, the name is "Development".
 - Select **Local**, leave **Act as Server on Port** set to 0 and click **OK**.
5. Click **New Environment** again to specify the production environment on the development server. This is so that the development environment knows where the

production environment exists.



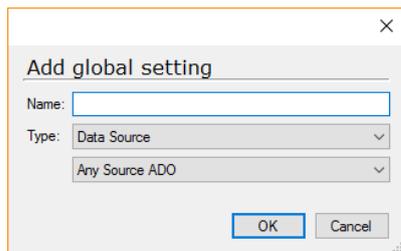
- Type a **Name**.
- Click **Remote** and fill in the remote server information:
 - **Server**: Type the server name or IP address of the production server.
 - **Local Port**: Enter any open port that is not being used by another application.
 - **Remote Port**: Enter the port you selected when setting up the production environment on the production server.

6. Click **OK**.

Creating Global Databases

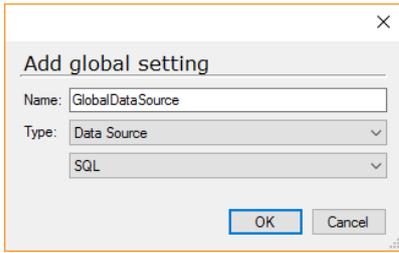
Global databases allow TimeXtender to know where the related databases reside for the Production and Development environments. For example, the location of the Staging Database for both the Production and Development environments will be specified.

1. Click **New Global Database** from the Environment Dialog window. The **Add Global Setting** window appears.



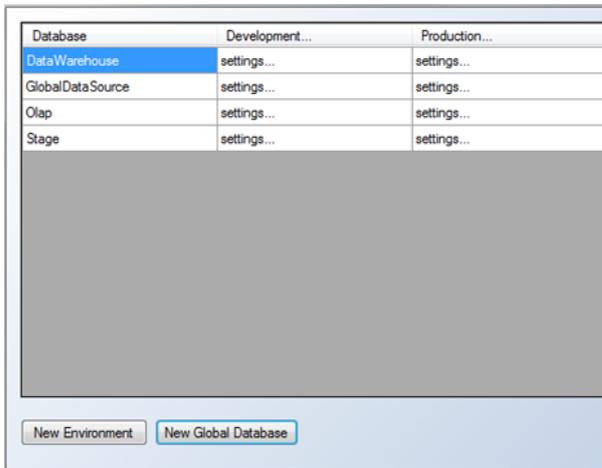
2. Using this window, you will be creating a series of databases that will be used in your project. You will create the following global databases:
 - Data source
 - Staging database
 - Data warehouse
 - SSAS Multidimensional
3. Assign a name to your data source, select **Data Source** in the **Type** section and select the relevant **Provider Type**. In this example, we will name our global database “Glob-

alDataSource” and use a provider type of Microsoft SQL.



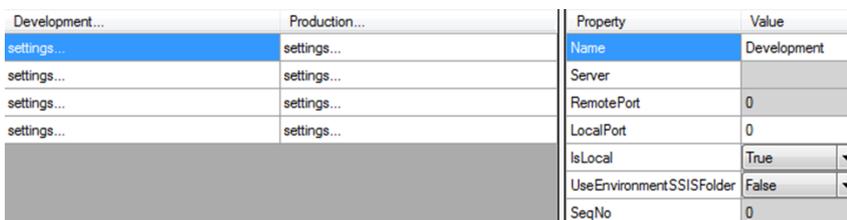
- Repeat the previous step for all databases in the project (staging, data warehouse, and SSAS Multidimensional).

Your results should look similar to those shown below:



Configuring Global Databases

Within the **EnvironmentProperty** window you should have a data source, data warehouse, SSAS Multidimensional, and staging database. Each environment, Development and Production, has a “**Settings...**” section for each database. You can also click the environment name to access additional settings.



Configuring the Data Source

- Select the “**Settings..**” field from the data source row in the Development column from the **Environment Properties** window. This will display the Settings pane to the right.

2. Enter the following information:

- **Server:** This will be the server address of the development server. Since this is currently on the development server, this can be localhost or the name of the server.
- **Name of the database from which data is extracted.** This will be the name of the NAV, AX, GP, or other database. In our example, this is JetCorpDemo.

Database	Development...	Production...	Property	Value
DataWarehouse	settings...	settings...	Server	localhost
GlobalDataSource	settings...	settings...	Catalog	JetCorpDemo
Olap	settings...	settings...	IntegratedSecurity	True
Stage	settings...	settings...	UserName	
			Password	
			ConnectionStringProperties	
			ConnectionTimeout	10
			CommandTimeout	300
			SSISApproach	As Parent
			ForceCodepageConversion	False
			ForceUnicodeConversion	False

3. Next click "**Settings..**" on the data source row in the **Production** column. Enter the following configuration:

- **Server:** This will be the name of the server on the Production Environment. In our example, the server name is jet-ent-2005.
- **Catalog:** This will be the name of the database from which data is extracted, for instance your ERP system. In our example, this is "JetCorpDemo".

Note: If you are using your live ERP database for extracting data in both the development and production environments, then the server name and catalog in both the Development and Production columns will be the same.

Database	Development...	Production...	Property	Value
DataWarehouse	settings...	settings...	Server	Jet-ent-2005
GlobalDataSource	settings...	settings...	Catalog	JetCorpDemo
Olap	settings...	settings...	IntegratedSecurity	True
Stage	settings...	settings...	UserName	
			Password	
			ConnectionStringProperties	
			ConnectionTimeout	10
			CommandTimeout	300
			SSISApproach	As Parent
			ForceCodepageConversion	False
			ForceUnicodeConversion	False

Configuring the Staging Database

1. Next you will need to configure another global database for the staging. Click "**Settings..**" on the **Stage** row in the **Development** column to display the **Settings** pane to the right.
2. Enter the following configuration:
 - **Server:** This will be the server address of the development server. In our example, this is localhost.

- **Catalog:** This will be the name associated with the staging database in the development environment. In our example, we use StageDev.

Database	Development...	Production...	Property	Value
DataWarehouse	settings...	settings...	Server	localhost
GlobalDataSource	settings...	settings...	Catalog	StageDev
Olap	settings...	settings...	Collation	
Stage	settings...	settings...	SSISServerName	
			IntegratedSecurity	True
			UserName	
			Password	
			ConnectionStringProperties	
			ConnectionTimeout	10
			CommandTimeout	300
			SSISApproach	As Parent
			MaxNumberOfRows	3000000
			Deployment Target	Not Set

3. Next click "**Settings...**" on the **Stage** row of the **Production** column.

4. Enter the following configuration:

- **Server:** This will be the name of the server for the Production Environment. In our example, the server name is "jet-ent-2005".
- **Catalog:** This will be the name associated with your staging database in the production environment. In our example, we use StageProd.

Database	Development...	Production...	Property	Value
DataWarehouse	settings...	settings...	Server	jet-ent-2005
GlobalDataSource	settings...	settings...	Catalog	StageProd
Olap	settings...	settings...	Collation	
Stage	settings...	settings...	SSISServerName	
			IntegratedSecurity	True
			UserName	
			Password	
			ConnectionStringProperties	
			ConnectionTimeout	10
			CommandTimeout	300
			SSISApproach	As Parent
			MaxNumberOfRows	3000000
			Deployment Target	Not Set

Configuring the Data Warehouse

Next you will need to configure another global database for the data warehouse.

1. Click "Settings...": on the data warehouse row in the Development column to display the Settings pane to the right.
2. Enter the following configuration:
 - **Server:** This will be the name of the server for the development environment. In our example, this is localhost.

- **Catalog:** This will be the name associated with your data warehouse in the development environment. In our example, we use DataWarehouseDev.

Database	Development...	Production...	Property	Value
DataWarehouse	settings...	settings...	Server	localhost
GlobalDataSource	settings...	settings...	Catalog	DataWarehouseDev
Olap	settings...	settings...	Collation	
Stage	settings...	settings...	SSISServerName	
			IntegratedSecurity	True
			UserName	
			Password	
			ConnectionStringProperties	
			ConnectionTimeout	10
			CommandTimeout	300
			SSISApproach	As Parent
			MaxNumberOfRows	0
			Deployment Target	Not Set

3. Next Click "Settings..." on the data warehouse row of the **Production** column.
4. Enter the following configuration:
 - **Server:** This will be the name of the server for the Production Environment. In our example the server name is "jet-ent-2005".
 - **Catalog:** This will be the name associated with your data warehouse in the production environment. In our example, we use "DataWarehouseDev".

Configuring the SSAS Multidimensional Database

Next you will need to configure another global database for the SSAS Multidimensional cubes.

1. Click "Settings." on the SSAS Multidimensional row in the Development column to display the Settings pane to the right..
2. Enter the following configuration:
 - **Server:** This will be the server address of the development server. In our example, this is "localhost".
 - **Catalog:** This will be the name associated with the SSAS Multidimensional database in the development environment. In our example, we use "OlapDev".

Database	Development...	Production...	Property	Value
DataWarehouse	settings...	settings...	Server	localhost
GlobalDataSource	settings...	settings...	Database	OlapDev
Olap	settings...	settings...	Collation	
Stage	settings...	settings...	Enable offline processing	False
			Front database	
			Deployment Target	Not Set

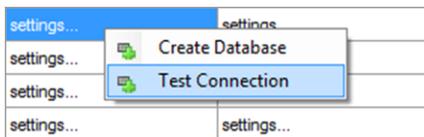
3. Next, Click "Settings..." on the SSAS Multidimensional row of the **Production** column.
4. Enter the following configuration:
 - **Server:** This will be the name of the server for the Production Environment. In our example, the server name is "jet-ent-2005".

- **Catalog:** This will be the name associated with your SSAS Multidimensional database in the production environment. In our example, we use "OlapProd".

Database	Development...	Production...	Property	Value
DataWarehouse	settings...	settings...	Server	jet-ent-2005
GlobalDataSource	settings...	settings...	Database	OlapProd
Olap	settings...	settings...	Collation	
Stage	settings...	settings...	Enable offline processing	False
			Front database	
			Deployment Target	Not Set

Creating the Global Databases

The final step in the configuration process is to test and create the global databases on SQL Server. This can be done from inside the Environmental Properties window. This needs to be done for both the development and production environments. Right-click "Settings..." and select Test Connection. If you get an error message, it generally means that the database has not been created yet. Right-click "Settings...", and select Create Database. Then retest the connection.



Perform this check on all Global Databases for both the Development and Production environments. Once this has been completed and all "Test Connection" responses return "Connection OK", click the OK button to close the Environment Properties window, and save all changes.

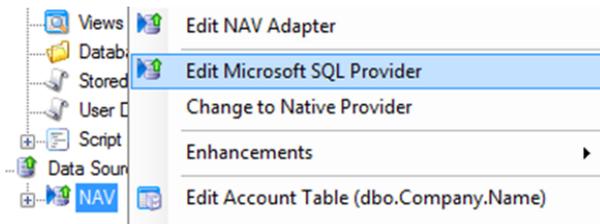
Note: The data source does not have the option to Create Database. This database represents the data source that TimeXtender is extracting from and will already exist in your infrastructure. An example of this will be your Dynamics Business Central (NAV), GP, or AX database.

Configure Project Connections

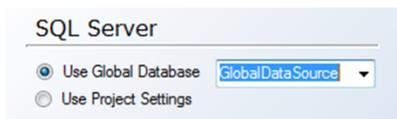
The environments have now been set up, and the global databases have been configured. The next step is to configure the connections in the project to utilize these Global Databases.

1. Right-click the adapter and select **Edit Microsoft SQL Provider**.

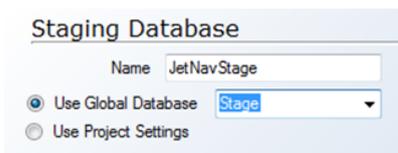
Note: This will vary depending on your data source type.



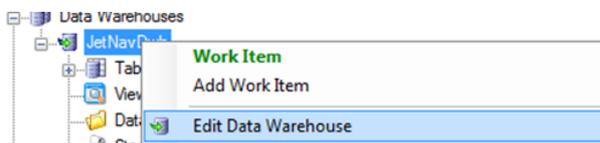
2. Select **Use Global Database** for the data source, choose the global database that represents your data source, and click OK. There will generally be only one global database displayed in the drop-down list.



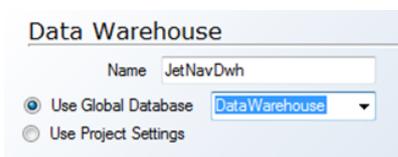
3. Navigate to your staging database, right-click the database, and select **Edit Staging Database**.
4. Select **Use Global Database** for the data source, choose the global database that represents your staging database, and click OK. There will generally be only one global database displayed in the drop-down list.



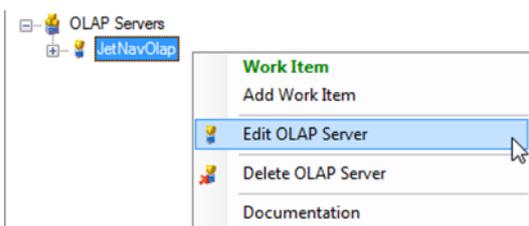
5. Navigate to your **Data warehouse**, right-click the database, and select Edit Data Warehouse.



6. Select **Use Global Database** for the data source, choose the Global Database that represents your data warehouse, and click There will generally be only one Global Database displayed in the drop-down list.



7. Right-click the SSAS Multidimensional Database, and select **Edit SSAS Multidimensional Server**.



8. Select **Use Global Database** for the data source, choose the global database that represents your SSAS Multidimensional database, and click OK. There will generally be only one Global Database displayed in the drop-down list.
9. The final step is to deploy and execute the project to ensure your project is properly configured and ready for transfer. On the **Tools** menu, click **Deploy and Execute Project** and then click **Start**.

Transfer the Project from Development to Production

You are now ready to transfer the project from the development to the production environment.

1. Log in to the **Development Environment** server, and open TimeXtender.
2. On the **Tools** menu, click **Multiple Environment Transfer**.
3. Click **Transfer** to migrate the project from the development server to the production server.

	Development		Production
Project	Jet Enterprise NAV v2.2		Jet Enterprise NAV v2.2
Version	2	Transfer ->	1
Date	7/18/2012 5:32 PM		7/19/2012 8:42 AM
Deployed	No		No
Next Scheduled Start			
Last Execution			
Execution Duration			
Import			

A dialog will appear asking you to confirm the transfer. Click **OK**.

4. Deploy the project on the production environment.
 - If you are transferring the project from development to production for the first time, or if you simply want to deploy all objects, right click the **Production Environment** folder and click **Deploy**. When the **Deployed** line in the **Production** column changes to "Yes", the process has finished.
 - If only some objects were changed in the development environment and are in need of deployment, right click the **Production Environment** folder and click **Partial Deployment**. The **Remote Deployment Window** opens. A list of deployable objects is displayed. Select the objects you want to deploy and click **Deploy**. While deployment is under way, the **Partial Deploy** window displays the deployment status. When deployment has finished, a window opens with a list of the deployment tasks completed. Click **Close** to close the window
5. Click **Close** to close the **Multiple Environment Transfer** window.

Execution Packages

Execution packages will automatically update the staging database, data warehouse, and SSAS Multidimensional cubes on a scheduled basis. Since projects deployed from the development environment will replace packages in the production environment, it is recommended that the desired execution packages be set up in the development environment.

This way, they are seamlessly transferred to the production environment with the package transfer. It may not be desirable to have automatic execution enabled in the development environment. This can be disabled by ensuring that the "TimeXtender Server Scheduler" service is disabled on the machine hosting the development environment. For more information regarding the configuration of execution packages, see [Execution Packages](#).

Thank you for using TimeXtender®

Have questions not answered in this user guide?

Need to get in touch with support?

Please visit our support site at

support.timextender.com