

TX DWA 16.7

RELEASE DOCUMENTATION

Contents

Introduction.....	2
New Features.....	2
Differential Deployment.....	3
Deployment Status Report.....	3
Managed Deployment	4
Data Export.....	4
Dynamic Project Variables	6
Custom Table Inserts.....	7
Guard Data Source	7
SQL Server 2016 Support	7
Show Data Types on Fields.....	8
Enable relations to/from system fields	8
Improved Features	9
New Deploy/execute Workflow	9
Query Tables on SQL, Oracle, Dynamics NAV and Dynamics AX Data Sources.....	10
Environment Security	12
Object Level Security – Column Permissions.....	13
Incremental Cube Load Improvements	14
Faster Setup of History Tables	15
Easier to Choose Tables for Full Load	15
Improved Performance with Batch Data Cleansing.....	15
Better Performance when Full Loading Incremental Tables.....	15
Related Tables Included in Dynamic Perspectives	15
Execution Package Setup is now Logged.....	16
Bug Fixes.....	17



Introduction

First of all, I would like to thank you for using our software. We are proud of our product and happy to know that our software helps organizations around the globe by democratizing the access to corporate data and liberate IT departments.

In this release of TX DWA we have included 10 new features and 11 improved features.

I would like to emphasize a few of the new features:

Managed and Differential Deployment adds more automation to the deployment process. TX DWA calculates the correct deployment order and deploys only the changes.

Data Export allows export of Data Warehouse data to other platforms like SQL Azure, Oracle, and text files. More export destinations will be added and if you have development skills you can even add your own custom destinations!

We have also added Environment Security that enables enterprise organizations to control who can transfer, deploy, execute project or make changes to data connections in each environment.

Last but not least I would like to mention the Deployment Status Report that will give you a complete overview of what TX DWA needs to deploy to the database(s). The status report can be called across environments to give you a good input to your changelog and deployment documentation.

As always we are eager to get your feedback on our software and we encourage you to visit our support forum on <http://support.timextender.com> where you can interact with other TX DWA users and the product team.

On behalf of the entire product team,

Thomas Lørup Duun CTO, TimeXtender



New Features

Differential Deployment

TX DWA 16.6 introduces differential deployment, a more efficient way of deploying your project. Instead of deploying everything, it is based on the idea that there is no need to deploy something that has already been deployed.

Each object in TX DWA can have many underlying objects on SQL Server. For instance, a table in a TX DWA project can have a raw and a valid table instance. On top of that there is an error table, a log table, an incremental table, a transformation view, a SSIS package and a data cleansing procedure. When a transformation is changed or added on a field, all that needs to be deployed is the transformation view. If a new lookup field is added to a table, all it needs to be deployed is the raw table, the valid table and the data cleansing procedure. The rest of the objects does not require any deployment.

Differential deployment is available for objects on the Data tab in TX DWA.

Differential deployment is enabled by default on new projects. To change the default setting, right click the project node, click on **Project Settings** and change the setting for **Enable Differential Deployment**. You can turn differential deployment on or off for a single deployment in the new **Deploy/execute Setup** window - see [New Deploy/execute Workflow](#).

This feature is also available when you deploy on a remote environment.

Deployment Status Report

Using the new differential deployment feature, TX DWA now lets you export a deployment status report that contain a list of what needs to be deployed. You can generate the report for projects, data warehouses, business units and OLAP servers. You can also generate the report for a remote environment. The report is formatted as a CSV file.

To generate a deployment status report for a project, data warehouse, business unit or OLAP server, right click on the object, click on **Advanced** and click on **Export Deployment Status**.

To generate a deployment status report for a remote environment, open **Multiple Environment Transfer**, right click on the remote environment and click on **Export Deployment Status**.

TX DWA will ask you to choose a location for the file. When the report has been successfully generated, TX DWA will ask you if you want to open it.



Managed Deployment

For almost two years, TX DWA has supported managed execution that, among other things, manages dependencies for you. In this release, we have used some of the same ideas to improve deployment.

TX DWA will take all dependencies it knows into account to make sure that objects are deployed in the correct order. It can calculate the dependencies of views, stored procedures and user-defined functions by looking at the objects that are referenced in the scripts. It will also deploy a table that has a lookup field after the table that is the source of the lookup field.

As in previous releases, you can add dependencies you want TX DWA to take into account. In addition to table dependencies, you can now define dependencies between tables, views, stored procedures, user-defined functions and any combination of those objects.

To add a dependency to a table, view, stored procedure or user-defined function, right click on the object, click on **Advanced**, click on **[Object] Dependencies** and then select the objects that the object depends. Click on **OK** when you are done.

Managed deployment is enabled by default on new projects. See [Differential Deployment](#) to learn how to set the default. You can turn managed deployment off for a single deployment in the new **Deploy/execute Setup** window - see [New Deploy/execute Workflow](#). This feature is also available when you deploy on a remote environment.

Data Export

Data Export is a new feature in TX DWA that allows you to push the content of a data warehouse to another database. At the moment, SQL Server, Oracle and text files are supported as destinations. 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 TX DWA.

You can download data export destination providers through the Custom Components setup application available on our support site: <https://support.timextender.com/hc/en-us/articles/209604866>

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. Click on the **Data Export** tab, right click on **Data Exports** and click on **Add Data Export**.
2. In **Name**, type a name for the data export destination.
3. In the **Provider** list, click on the provider you wish to use.
4. In the **Setup Property** list, click on **Setup Properties**. In the grid below, enter the setup properties required by the selected data export destination.
5. (Optional) Click on **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. On the **Data Export** tab, expand **Data Warehouses** until you have located the table you want to add. Drag the table from **Tables** under Data Warehouses and drop it on **Tables** under the relevant data export under **Data Exports**.

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 on a name in the **Destination Name** column to edit the name as it appears on the **Data Export**.
4. Click on **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, do as follows. On the **Data Export** tab, expand **Data Warehouses** until you have located the data warehouse that contains the tables or views you want to add. Drag **Tables** or **Views** under the data warehouse – or the entire data warehouse – and drop it on the relevant data export under **Data Exports**.

TX DWA 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.

Dynamic Project Variables

This release introduces dynamic variables, a new kind of project variable where the value of the variable is generated by a custom script written by the user.

Adding a Dynamic Project Variable

To add a dynamic project variable to your project, follow the steps below.

1. Right click on your project in the project tree and click on **Project Variables**. The **Project Variables** window opens. Click on **Add**.
2. Type a name for the variable in the **Name** box.
3. In the **Type** list, select **Dynamic**.
4. In the **Resolve Type** list, select when you wish to resolve the value of your dynamic 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.



5. In the **Context** list, select the context you want the variable to be resolved in.
6. Click on **Script Editor** to open the standard script editor in TX DWA and use it to write the script that calculates the value of the variable.
7. Click on **OK**.

Custom Table Inserts

With Custom Table Inserts, a new feature in TX DWA 16.7, you can now write your own table inserts. This gives you more flexibility compared to the Table Inserts feature already available in the product.

Adding a Custom Table Insert

To add a Custom Table Insert to a table, follow the steps below.

1. On the Data tab, in a data warehouse, right click on the table you wish to add the Custom Table Insert to, click on **Advanced** and click on **Add Custom Table Insert**. The **Add Custom Table Insert (<table name>)** window opens.
2. The editor is similar to other custom code editors in TX DWA. In the text field, write the SQL SELECT statement that gives you the data you wish to insert into the table. Note: The fields you include in your SELECT statement must exist in the table you are inserting data into. Use AS when the names do not match.
3. In the **Destination table** list, click on the instance of the table you wish to insert the data into. The default is Raw.
4. Click on **OK**.

Guard Data Source

You can now guard a data source in TX DWA, which prevents tables that get their data from the data source from being deployed, executed or both. It has long been possible to guard a table in the data warehouse in TX DWA and this feature simply makes it easier to guard all tables that originate from a particular data source. In general, the guard feature is useful for tables that contain data that never changes, e.g. from a legacy system.

Guarding a Data Source

To guard a data source, follow the steps below.

1. On the **Data** tab, right click on the data source, click on **Advanced** and click on **Guard**. The **Guard** window opens.
2. Select **Guard on deployment** to prevent TX DWA from deploying the table and/or **Guard on execution** to prevent TX DWA from executing the table.
3. Click on **OK**.

SQL Server 2016 Support

TimeXtender is firmly dedicated to SQL Server. When Microsoft releases a new version of SQL Server, we aim to release a version of TX DWA that supports the new SQL Server within



three months. The final version of SQL Server 2016 was released on June 1, and with this release of TX DWA, we officially support the new version.

Show Data Types on Fields

You can now show the data type of fields in the project tree, which can be useful in many cases. To enable this, click on **Tools** in the ribbon, click on **Window and Menu Settings** and then select **Show Data Types**. The data type will be shown in a parenthesis after the field name.



Enable relations to/from system fields

In previous versions of TX DWA, you could not create relations between the system fields added by TX DWA – DW_Id, DW_Batch, DW_SourceCode, DW_TimeStamp – and other system fields or regular fields. In this new release of TX DWA, this is now possible.

Improved Features

New Deploy/execute Workflow

To accommodate the new deployment features in this release of TX DWA, we have redesigned the deploy and/or execution workflow. In previous versions, you would choose to deploy, deploy modified objects, deploy and execute etc. in the context menu on the individual objects. Since the number of possible combinations would get quite large if we added the new features in a similar way, we have added a Deploy/Execute Setup window.

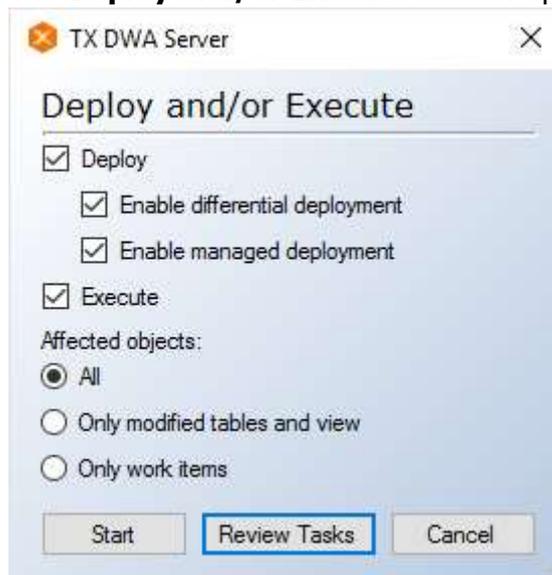
Deploying and/or executing an Object

To deploy and/or execute an object in a data warehouse, follow the steps below.

1. Right-click on the object and click on **Deploy**, **Execute** or **Deploy and Execute**.
- OR -

Click on 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 deployment options. Their initial setting is based on your project settings.
 - Select **Enable differential deployment** to take advantage of TX DWA'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 as in previous versions of TX DWA.
 - Select **Enable managed deployment** to have TX DWA 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, just like in previous versions of TX DWA.

3. Select **Execute** to execute the selected objects. The execution settings are governed by the default execution package.
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 on **Start** to begin the deployment and/or execution process as soon as TX DWA 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 TX DWA a few moments to calculate what steps need to be deployed. If TX DWA does not find any steps that needs to be deployed, you will be asked if you wish 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 wish 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, TX DWA will begin the deployment and/or execution process immediately. Otherwise, review the tasks and settings and click on **Start**.

Query Tables on SQL, Oracle, Dynamics NAV and Dynamics AX Data Sources

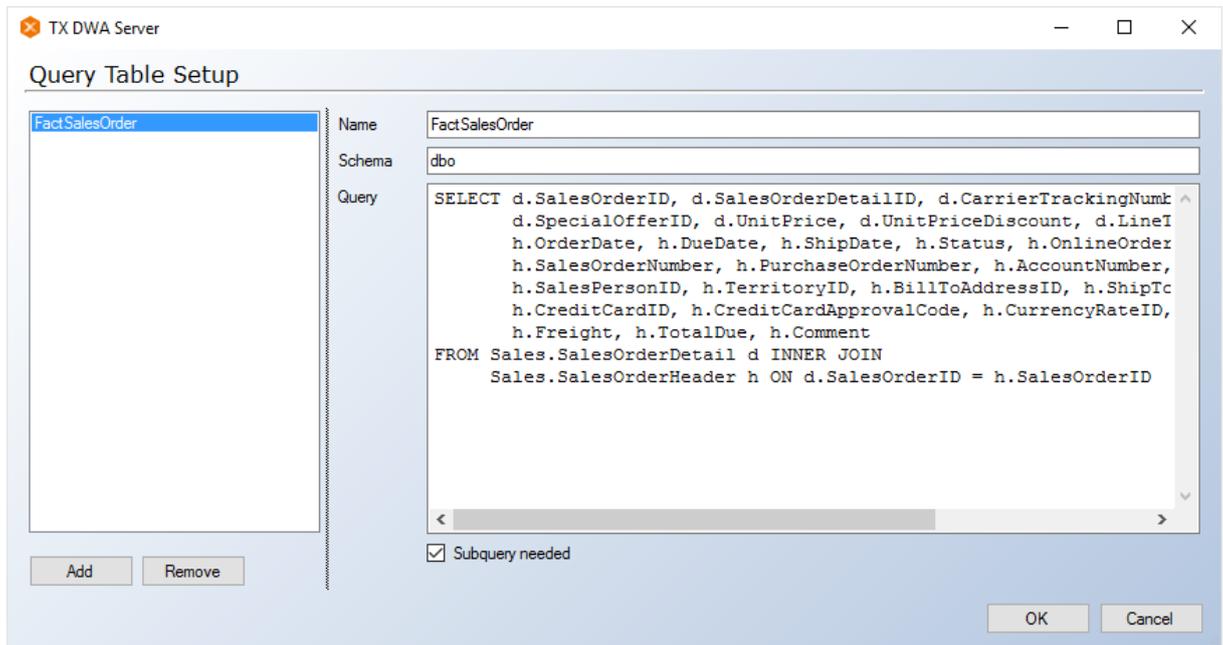
Previously, the query tables feature was only available on the Custom and AnySource data sources. This releases adds support for the feature on the SQL and Oracle data sources and the Dynamics NAV and Dynamics AX adapters.

Adding a Query Table

To add a query table, follow the steps below.

1. Connect to one of the supported data sources.
2. Right click on the data source, click on **Advanced** and click on **Query Table Setup**. The **Query Table Setup** window opens.





3. Click on **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 wish 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 or if the joined tables have ambiguous field names. Otherwise, selection rules will fail.
8. Repeat step 3-7 to add the tables you need and click on **OK**.
9. Right click on the data source and click on **Read Objects from Data Source**. The tables are listed in the panel in the right-hand side of TX DWA and can be included in the project like any other table.

Handling Accounts in Dynamics NAV

When you create query tables for Dynamics 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 NAV Adapter does it, you can use placeholders:

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

TX DWA 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]

```

Environment Security

In the new release, we have added security functionality to the environment setup. You can now limit access to certain functionality to members of specific Active Directory (AD) groups.

In relation to this feature, TX DWA will now also log events in the environment to the ServerLog table in the repository. This can be disabled by clearing **Enable detailed logging** under **Additional Settings** for the environment.

Adding Security to an Environment

Follow the steps below to add security to an environment.

1. In the ribbon, on the **Tools** tab, click on the **Environment Properties**. The **Environment Properties** window opens.
2. Right click on an environment and click on **Environment Security**. The **Environment Security** window opens.



Each line in the window represents a permission that can be granted to an AD group. You can click on .. to browse AD groups in the standard interface. The AD groups you add need to exist on the current domain and you need to be logged in with a user account that is a member of the AD groups you add.

3. In the **Admin** box, enter the name of the AD group that should have full access to the environment.
4. In the **Transfer** box, enter the name of the AD group that should be able to transfer a project to the environment.
5. In the **Deploy** box, enter the name of the AD group that should be able to deploy projects on the environment.



6. In the **Execute** box, enter the name of the AD group that should be able to execute projects on the environment.
7. In the **Database** box, enter the name of the AD group that should be able to add, edit and delete the global databases defined in the environment.
8. Click on **OK**.

Object Level Security – Column Permissions

We have improved the object level security feature of TX DWA with a new level of granularity. You can now set permissions on specific columns/fields on a table, not just the entire table.

TX DWA uses the same allow/deny concept as SQL Server with three possible states:

- **Not set** (grey 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:

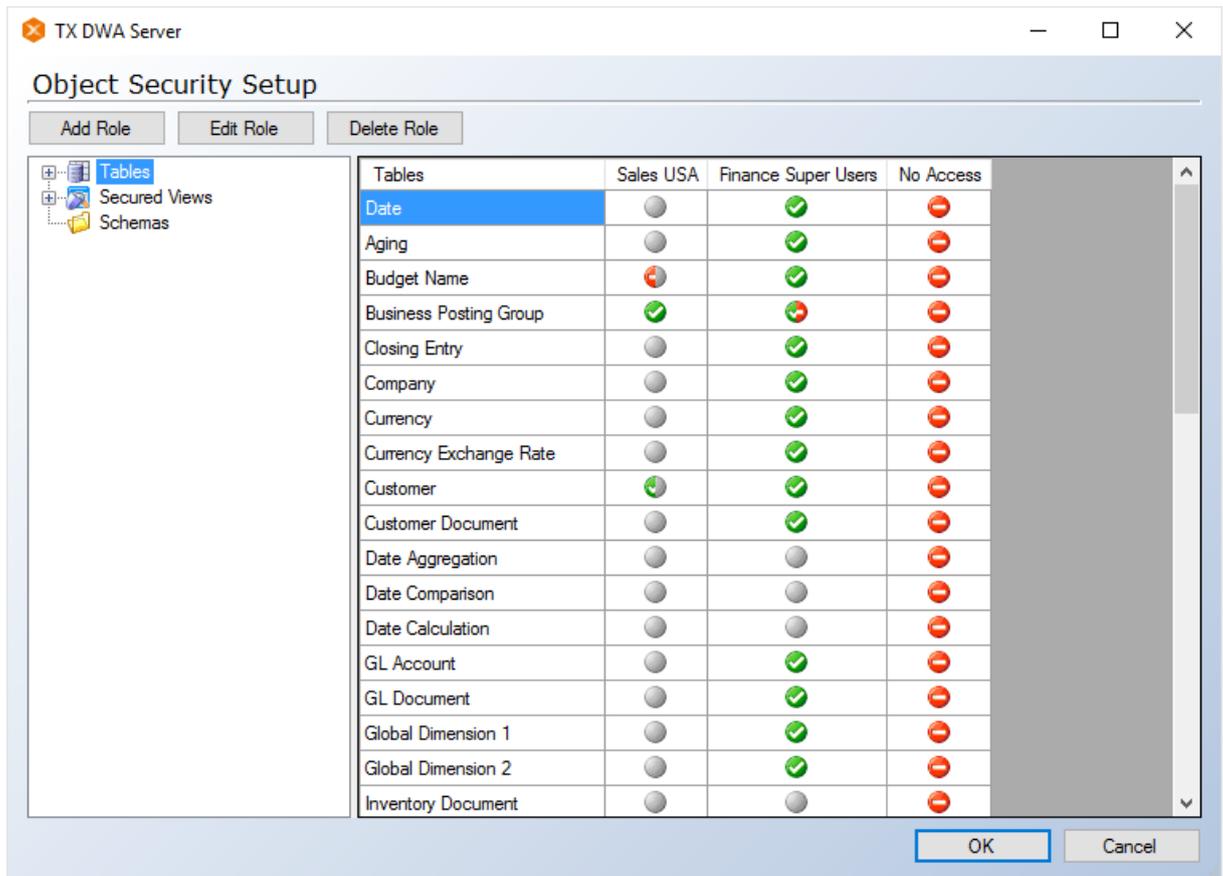
- **Partially Granted** (green and grey icon). The database role is granted access to some columns on the table. Note that you will also see this icon if the database role is granted access to all columns on a table since this will not automatically set Allow on the table level.
- **Partially Denied** (red and grey 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 allowed access to some columns and denied access to other columns on the table.

Assigning Object or Column Level Permissions to Database Roles

To assign object level permissions, or column level permissions on tables, to database roles, follow the steps below.

1. On the **Data** tab, in the project tree, under **Data Warehouses** and the relevant data warehouse, right click on **Security** and click on **Object Security Setup**. The **Object Security Setup** window opens.





2. Click on **Tables**, **Views** or **Schemas** in the left-hand column to choose the type of object you wish to set up access for. Expand **Tables** and click on 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 on icon in the intersection between the object name and the database role to change the permission for the database role on that object. Note that if you set column level permissions on a table, this will overwrite any current object level permissions set and vice versa.
4. (Optional) Click on **Add Role**, **Edit Role** or **Delete Role** to add, edit or delete database roles as needed.
5. Click on **OK** to save changes and close the window.

Incremental Cube Load Improvements

With Enterprise edition of SQL Server table partitioning is an option. Incremental cube update checks which fact table partitions that has been modified and updates these partitions only when performing a processing of the cube. Prior to this release the deployment target on the OLAP database had to be set to Enterprise Edition, but now it just checks if the underlying table is partitioned or not. While working on this we also found a better way to process the individual partitions to further improve the performance.



Faster Setup of History Tables

In this release, we have made some small improvements that makes it faster to set up history/slowly changing dimensions on tables. When you enable history on a table, any fields currently included in the primary key on a table will automatically be included in the natural key in history settings. In addition to this, all fields on the table are now automatically type 1 and you only have to select the fields you want as type 2.

Easier to Choose Tables for Full Load

The first release of TX DWA added the option on execution packages to choose some incrementally loaded tables for full load. The new release improves on this by making it possible to add a business unit, data warehouse, OLAP server or the entire project to the list of objects that should be fully loaded when the execution package is run. This way, you can simply add e.g. a business unit to the list and all current and future incrementally loaded tables in that business unit will be fully loaded when the execution package runs.

Improved Performance with Batch Data Cleansing

You can now 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.

Enabling Batch Data Cleansing

To enable batch data cleansing, follow the steps below.

1. On the Data tab, in a data warehouse, right click on the table you wish to use batch data cleansing on and click on **Settings**.
2. Click on 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 on **OK**.

Better Performance when Full Loading Incremental Tables

We have made a small improvement that disables generation of indexes when you perform a full load on a source-based incrementally loaded table. While indexes are important for incremental loads, updating indexes while doing a full load carries an unnecessary performance penalty.

Related Tables Included in Dynamic Perspectives

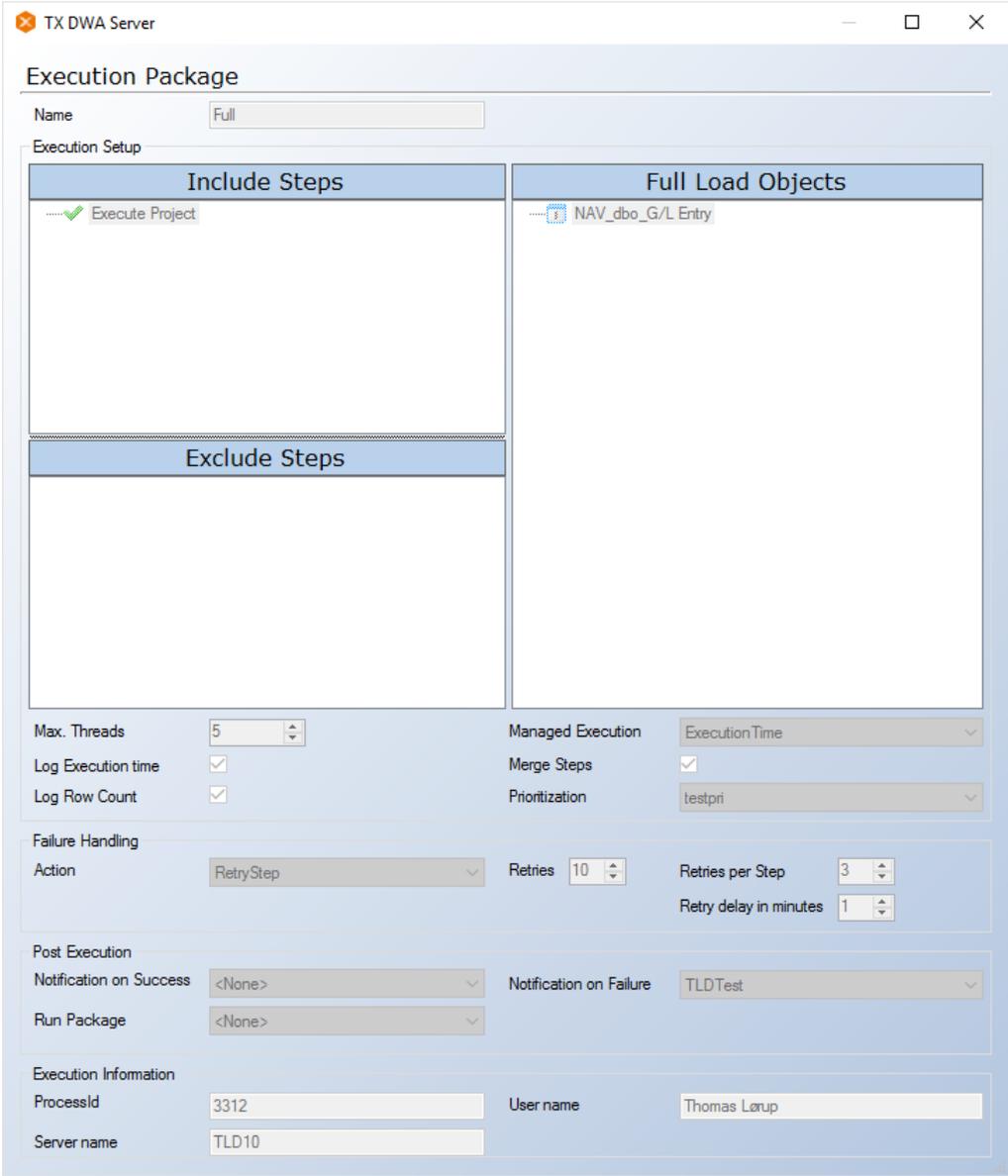
TX DWA will now take table relations into consideration when calculating what tables will be included in a dynamic perspective. Any tables that are related to a table in the perspective with a relationship type that is either "error" or "error with physical relation" will be included. Relations of the types "warning" or "relation only" are not included, since these relation types imply a weak connection between the tables.



Execution Package Setup is now Logged

The new release contains an addition to the execution log. In addition to the execution diagram and message, you can now view the settings used for the execution. The settings are stored in an XML format and displayed as a read-only version of the execution package setup window.

To view the execution log, go to the **Execution** tab, right click on an execution package and click on **View Execution History Log**. You will find the new option in the rightmost column of the list.



Bug Fixes

16.7

ID	Title
B00002	Execution errors not logged to trace log or event log
B00003	Error in Allow Failing Data Source with BK Hash Key enabled
B00006	Issue with negative value and zeros in Incremental load
B00007	Export project with invalid file character as project name error
B00009	F2 rename on Data Warehouse does not mark tables as dirty
B00010	Junk Dimension: Does not update datatype if changed due to synchronization
B00017	Activating project perspective does not check for dynamic recalculations
B00018	Importing a project where the name already exists but the ids are different causes an error
B00023	CTRL+E Shortcut does not function in Conditional Lookups
B00032	Edit system type project variables issue
B00033	Descriptions doesn't work properly when using perspectives
B00034	Primary key removed after table partitioning
B00041	Sun Adapter: Object reference not set to an instance of an object
B00043	Bug in datetime conversion for SQL Execution Logger
B00044	Securable Views cannot deploy when a Default valid target schema is defined
B00045	QlikView vertical resize of splitter
B00048	Date Table: Adding Conditions to Field Validations problem
B00052	Project variable usage on conditional lookup
B00053	Case sensitive sql problem on conditional lookup
B00054	Execution package: Excluded external business doesn't get excluded
B00061	Missing index auto indexes on _V with source based incremental load
B00062	Data Selection rules with environment selectors Full Load Issue
B00065	Date Table: Fiscal Year is wrong
B00073	Execution Packages issue
B00074	Missing list of data fields when adding a custom selection rule on an agg. table based upon another agg. table



B00078	Change physical valid table settings with no lookup fields on another table
B00081	Connection manager adds additional data sources twice
B00083	Disable bk_hashkey used as primary key with history enabled
B00085	Data movement view problem for none mapped columns
B00088	Screen sizing problem - Custom Code
B00096	Execute table from data source with reserved column names
B00097	Error when importing project with External Step when it references a project you do not have in your repository.
B00098	Synchronize data source with new fields does not take reserved names into account
B00102	Oracle convert out of range dates missing name qualifier
B00103	Oracle is not supporting data type "timestamp(x) with time zone"
B00109	External business unit icons does not have transparent background
B00110	Qlik Naming templates is not supporting Views
B00112	Dynamic Perspective bug
B00113	Error when deploying after setting relation to "Error with physical connection"
B00117	Error previewing none selected tables on an External SQL Connection
B00121	Aggregated tables loses its connection to the original table when importing an exported project
B00127	Option to Show Data Types on Fields
B00131	Sun adapter: incremental load issue
B00137	Description editor empty value error
B00138	SQL Case sensitive problem with BK Hash Key and validation rules
B00141	Regression: Index Compression
B00142	Synchronize external business unit field names error
B00143	Edit mapping field without source field error
B00146	"Save project as" corrupts aggregated tables
B00147	Performance issue with perspectives
B00154	Execution Package: Possible to resume an already running package
B00155	Adding a condition to a transformation missing a check when adding a second condition
B00164	Delete index collection with more than one index and then delete its field causes an exception



B00165	Add Custom data command sets custom data with null data row when no valid fields exists
B00169	Schema error on like when Create one view per table model is disabled
B00177	Multiple environments transfer dialog sometimes freezes TX DWA
B00191	Prioritization clean-up after last tables in collection missing
B00199	When execution fails the execution package will link its message to the failed steps message
B00214	Aggregate Tables: Changes to the fields doesn't get saved to the repository
B00220	Qlik - Wrong scripts when table model has no visible fields
B00223	Issue with Physical Cube Perspectives
B00249	Enabling/disabling SSIS logging should be included in the checksum of differential deployment

16.7.2

ID	Title
B00254	Differential deployment on NAV adapter with custom selection rule containing a dynamic variable
B00256	Refresh project nodes without data export feature results in errors
B00257	Customized code: Trying to update tree before objects have been loaded result in error

16.7.3

ID	Title
B00261	Ax Enum table synchronization problem
B00262	SalesForce adapter writes to system fields when transferring data

16.8.1

ID	Title
B00264	Custom condition on a transformation on a field gets object null reference.
B00265	Selection rule using a project variable as condition gets object null reference
B00267	Global Database translates EscapeCharacter wrongly
B00272	Dynamic project perspective Table Insert error



B00273	Domain group membership check should not be performed if environment security has not been defined.
B00274	DateTable: DateValue not inside FiscalFirstDayOfYear and FiscalLastDayOfYear range
B00275	Errors when trying to set up object dependencies on a table with a table insert
B00278	Spelling error in new deploy and execute dialog

16.8.2

ID	Title
B00279	Unable to add custom SSIS packages when SSIS folders is enabled
B00280	FTP task is difficult to get deployed
B00288	CRM synchronization error

16.8.4

ID	Title
B00296	Differential deploy does not check the hash keys setting on Target Base Incremental Load and History tables
B00297	M3 db2 adapter using wrong schema in read objects
B00299	The advanced synchronize dialog closes on errors without a message
B00300	DB2 data type white space errors
B00301	Error when remote environment actions fails with non serializable exceptions
B00307	Salesforce source fields in picklist/multipicklist issue
B00317	Issue with CRM Adapter Organization List
B00318	CRM Adapter truncates strings when they are longer than the field definition
B00319	NULLs not being retained on Text files

16.8.5

ID	Title
B00270	Wrong Editable Name for Adapter fields in Business Unit



16.8.6

ID	Title
B00353	AX adapter incremental load date formats issue
B00348	Resume Execution Process Id/SSID Check issue