

The next generation CONTROL®

# version 10.5

Release Announcement

---

# CONTROL® 10.5 Release Announcement

## Table of Contents

CONTROL® 10.5 – Overview .....	4
Enhancements for the Information Navigator .....	4
Flex views .....	4
Query functions .....	4
Optimized flex tables .....	8
Save a new flex view for all users .....	9
Pivot table and pivot chart view styles .....	10
Easy Power BI .....	11
Tabular Editor .....	14
Login performance .....	15
Enhancements for the Administrator .....	15
AS Query datasources .....	15
New option for SQL Query datasources .....	16
Power BI integration enhancements .....	17
Power pivot datasource properties .....	17
New tab in power pivot model edit book .....	18
Materialization enhancements .....	18
Export options .....	18
Exported dimension tables .....	18
Rule-based forecasting .....	18
Transform enhancements .....	19
Overrides for transform and mapping steps in action scripts .....	20
Column store model and datasource data tables .....	27
Passwords for views and sheets .....	29
Scenario-Hierarchy administrative model and view .....	31
Improvements to hierarchy maintenance .....	32
CONTROL® Web .....	35
Overview .....	35
Features .....	35
New web options for 10.5 .....	36



# CONTROL<sup>®</sup> 10.5 Release Announcement



# CONTROL® 10.5 Release Announcement

## CONTROL® 10.5 – Overview

CONTROL® 10.5 is a short-cycle release that is focused on enhancing three product capabilities, which are proving very popular with our clients and prospective clients:

- Flex Views – for budget and forecast collection, management reviews, and analyses
- Power BI – for great visualizations
- Rule-based forecasting – for improving the speed, accuracy and workload of this critical function

In addition, there is a technology component in this release that we believe will be very important to our clients in the future. With the release of Microsoft's SQL Server 2019, there are significant opportunities to leverage new capabilities providing both functional and performance improvements. KCI strongly encourages its SQL Server clients running on-premise to consider adopting this release for use with CONTROL®.

Also available with this release, the CONTROL® Web product supports a streamlined version of view design, allowing browser users to customize and save reports.

## Enhancements for the Information Navigator

### Flex views

This release contains numerous functional and performance improvements to flex views.

### Query functions

There are two new flex functions that allow you to integrate data from external sources into your flex views:

- **SQLQuery** accesses data from relational tables and views, which are commonly used in operational systems, like ERP and CRM.
- **ASQuery** accesses data from Microsoft Analysis Services, the central repository for business intelligence and analytic applications.

The query results can be accessed by row and column or via a flex table:



# CONTROL® 10.5 Release Announcement

The screenshot displays the CONTROL 10.5 interface. On the left, a data table is visible with columns: DEPARTMENT, ACCOUNT, PERIOD, VOUCHER\_NUMBER, CHECK\_NUMBER, SOURCE, JOURNAL\_NUMBER, PO\_NUMBER, PO\_LINE\_NUMBER, and VENDC. The table contains 27 rows of data. On the right, a 'View - AP Detail - Flex SQLQuery' window is open, showing the configuration for a SQL query. The 'SQLSource' is set to 'DataSource(EXMP\_TBL\_AP\_DETAIL)'. The 'Query Expression' is 'Select \* from cntadm.tbl\_ap\_detail'. The 'Row Number' is '<2>' and the 'Column' is '<1>'. The 'Reference type' is 'Absolute'. The 'Source type' is 'Specific Data Source'. The 'Name' field is empty, and the 'Argument Value' field is empty. The 'Apply to cell A2' button is visible, along with 'Is Flex' and 'Is CRF' checkboxes, and an 'Update' button.

## SQLQuery

Some important details about the **SQLQuery** flex function:

- The data is always read-only. (If you want to update, you must use the **Data** flex function on a source data view).
- The **Query Expression** argument supports worksheet references, which is a convenient way to assemble, customize, and experiment with the query.
- For security reasons, there are two **Source types**:
  - **Specific Data Source** – a *datasource* where the user has read access. The query is restricted to accessing the underlying table or view. (For non-relational sources, the query will be applied to the relational image of the file or Excel range)
  - **General Query** – a *datasource* in the **Dynamic Datasources** category, to which your administrator must give you access. You can write any query you want, but you must have **SELECT** privilege on the tables and views in your query. (For more details about the security, see [Enhancements for the Administrator.](#))
- **Query Expression** – SQL expression or fragment:
  - Where the **Source type** is **Specific Data Source**, the query expression can be a query, a **WHERE** clause, an **ORDER BY** clause or both, but can only reference the underlying table or view of the *datasource*.
    - If the **SELECT** portion of the query isn't specified, **SELECT \* FROM table** is presumed
    - If the *datasource* has a non-empty SQL Query property, it is added to the WHERE clause with an **AND**.
    - If the supplied query has no **ORDER BY** clause, the ordering in the *datasource* is used



# CONTROL® 10.5 Release Announcement

- Where the **Source type** is **General Query**, the query expression executes without modification.
- **Row Number** – the row of the result set. **Row 1** contains the column names; **Row 2** is the first data row. You can select multiple rows. (Multi-select supported)
- **Column** – the column of the result set (Multi-select supported)

A few helpful hints:

- **Page Relative** is the default **Reference type** for the **Row Number** and **Column** arguments. This makes it simple to select a few rows and columns, apply the function, and then turn the result into a flex table, which will contain all the rows and columns of the result.
- **General Query** – Carefully consider the security ramifications before adding sources, and grant access to appropriate users only.

## ASQuery

The **ASQuery** flex function is very similar to **SQLQuery**, but it queries either multi-dimensional or tabular models in Analysis Services, and the query is written in either DAX or MDX.

The screenshot displays an Excel spreadsheet with the following data:

REGION_NAME	PRODUCT_GROUP_NAME	YEAR	UNITS	GROSS_SALES	NET_REVENUE
Asia	Product Group 0001	2018	5,224,151	33,889,862	31,269,796
Asia	Product Group 0001	2019	3,881,876	24,083,910	22,221,954
Asia	Product Group 0002	2018	4,313,789	19,324,507	18,106,846
Asia	Product Group 0002	2019	3,065,605	13,733,006	12,867,673

The Flex Pane on the right shows the configuration for the ASQuery function:

- ASSource: \$B\$1
- Query Expression: \$B\$2
- Row Number: <1>
- Column: <1>
- Reference type: Worksheet
- Excel cell address: \$B\$1
- Value: Model[REVENUE\_VARANCEPOWERPOT]

Some important details about **ASQuery**:

- The data is always read-only.
- The **Query Expression** argument supports worksheet references, which is a convenient way to assemble, customize, and experiment with the query.
- The following **Source types** are available in the **Flex Pane**:

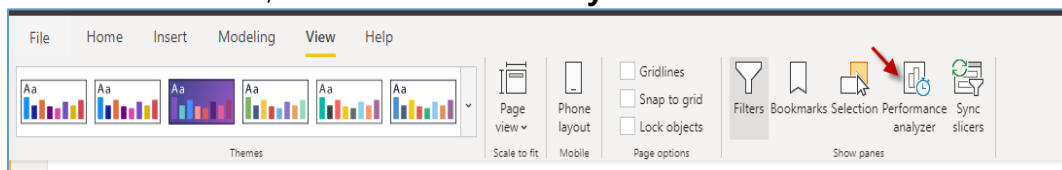


# CONTROL® 10.5 Release Announcement

- **PowerPivot Models** – any Power Pivot *model* with an **Analysis Services Usage** property of **Exported to AS** or **Managed externally** - uses the **AS Server** and **AS Database** associated with the PowerPivot *model*.
- **AS Query Datasources** - uses the **AS Server** and **AS Database** associated with the *datasource*.
- **All** – allows selection of any database on the default AS Server. To reference a database on a different AS Server, use `<ServerName.Database>`.
- **Query Expression:**
  - DAX queries begin with **EVALUATE**.
  - MDX queries begin with **SELECT**.
  - For an **AS Query Datasource**, if the query expression is blank, the unmodified *datasource* expression is used.
- **Row Number** – the row of the result set. **Row 1** contains the column names; **Row 2** is the first data row (Multi-select supported).
- **Column** – the column of the result set (Multi-select supported).

A few helpful hints:

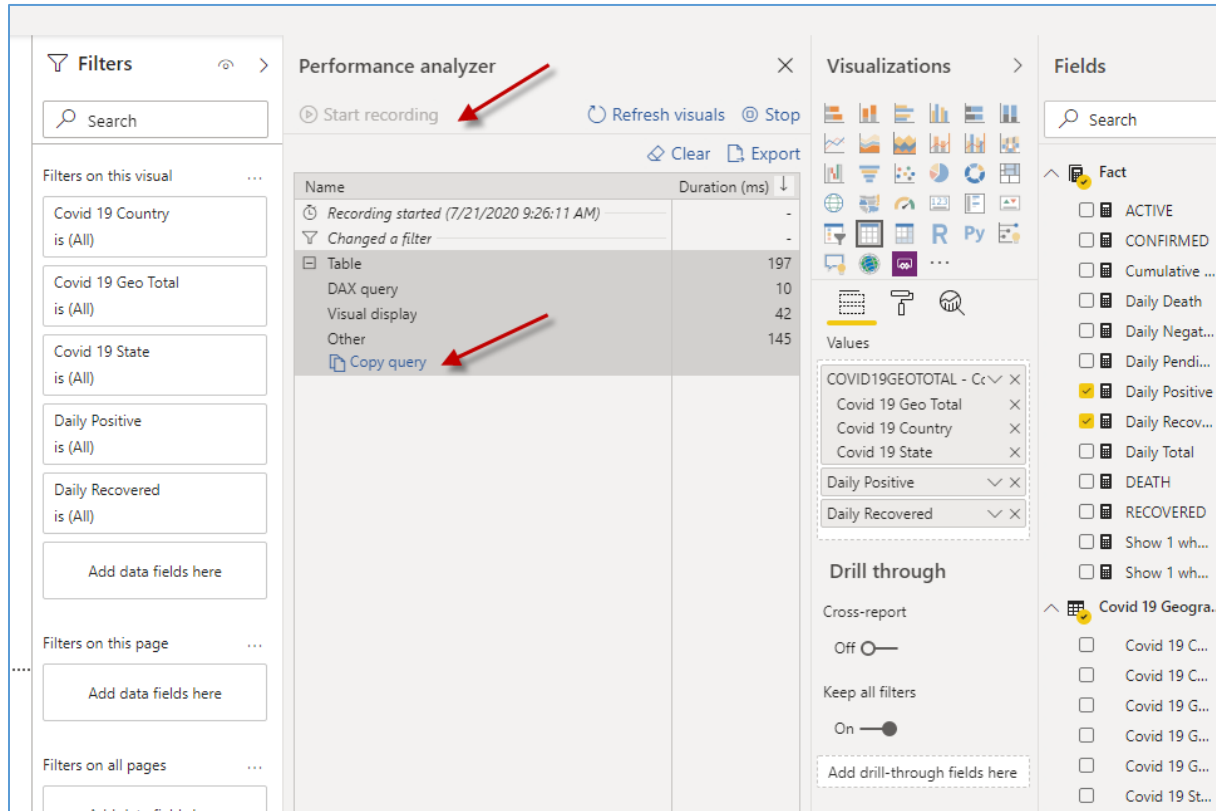
- **Page Relative** is the default **Reference type** for the **Row Number** and **Column** arguments. This makes it simple to select a few rows and columns, apply the function, and then turn the result into a flex table, which will contain all the rows and columns of the result.
- A useful resource for DAX is *The Definitive Guide to DAX* by Russo and Ferrari. The authors have published other books on Analysis Services, which are also very good.
- In Microsoft Power BI Desktop, to capture the DAX query used by Power BI Desktop:
  1. Open Power BI Desktop and select an AS Tabular Server and Model
  2. On the **View** tab, click **Performance analyzer**.



3. Click **Start Recording**, create a visualization (it is helpful to do a table), and **Copy Query**.



# CONTROL® 10.5 Release Announcement



You can paste the query into Notepad to examine it or paste it directly into the Flex Pane or worksheet.

## Optimized flex tables

Flex tables have proven to be a very popular component in flex views, particularly where insights into operational or transactional data is important.

When a flex table gets very large (1000's of rows), the time spent populating or updating the flex formulas can be significant.

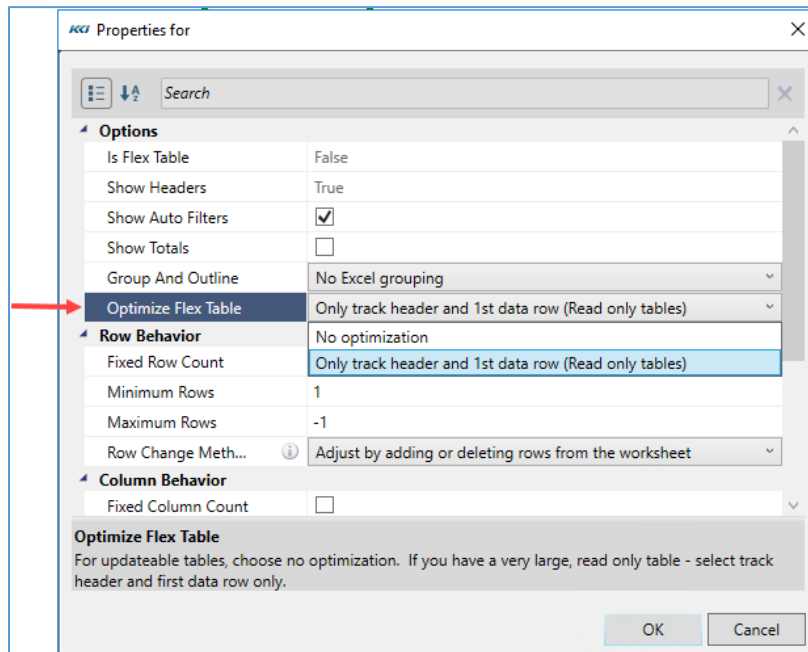
If the flex table contains read-only data, CONTROL® can populate the table by tracking the flex formulas in just the table header and first data row.

When you launch the flex table **Property** dialog, you will see a new property **Optimize Flex Table**.





# CONTROL® 10.5 Release Announcement



When you make an area of a flex view a flex table, the property is automatically set based on whether the content of the table is updateable.

You may change this default behavior to suit your specific application.

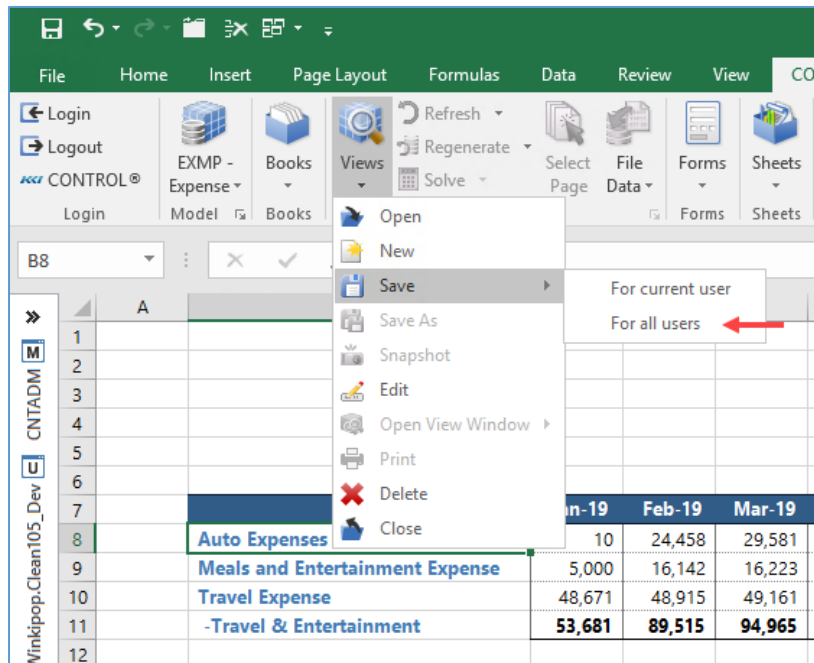
## Save a new flex view for all users

A popular approach to creating a brand-new flex view is to copy cells from an existing flex view using **Copy** (or **Copy Flex**), pasting onto a blank worksheet, and then clicking **Views > Save**. This creates a new flex view for the current user.

You may now follow the same steps but create a view for all users.



# CONTROL® 10.5 Release Announcement



## Pivot table and pivot chart view styles

While these *view* styles have been available in CONTROL® in previous releases, their behavior has been significantly changed and improved. The motivation for this enhancement is to support casual users who are familiar and comfortable with Excel pivot tables, but are not trained in all the features of CONTROL® *views*.

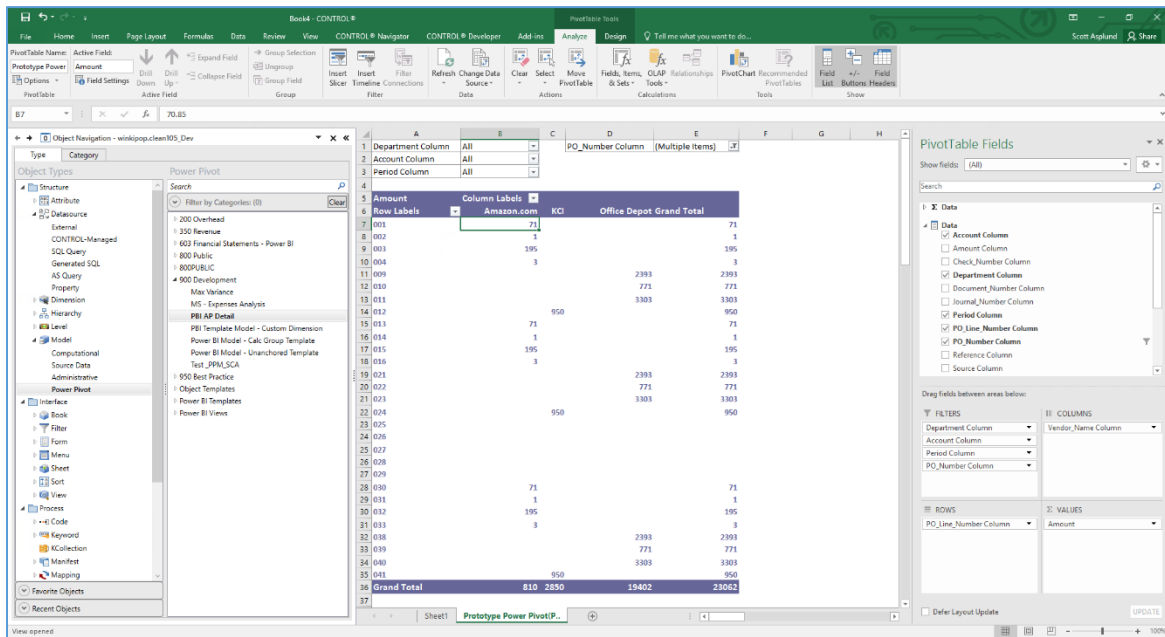
When you save a pivot table or pivot chart style *view*, the worksheet containing the pivot table or chart is saved, including all customizations such as field assignments, formatting, added calculations, slicers, etc.

When you re-open the *view*, the data is refreshed from the underlying CONTROL® *model* and *view*, preserving all of your pivot table/chart customizations.

When you create a new *view* on a power pivot *model* that has a CONTROL®-managed or externally managed Analysis Services model, the pivot table or chart is connected to the Analysis Services model directly, and you can make use of all the capabilities on Excel's **PivotTable Analyze** ribbon, including the calculation tools for OLAP:



# CONTROL® 10.5 Release Announcement



Pivot table and pivot chart view styles are now supported for source data *views* as well.

In this release, every view style is available for every subclass of *view*, except for the following:


- Object views - **Multi-Page**
- Power Pivot views – **Standard, Multi-Page, List**

The **Power Pivot Table** and **Power Pivot Chart** views introduced in release 10.2 are no longer supported.

## Easy Power BI

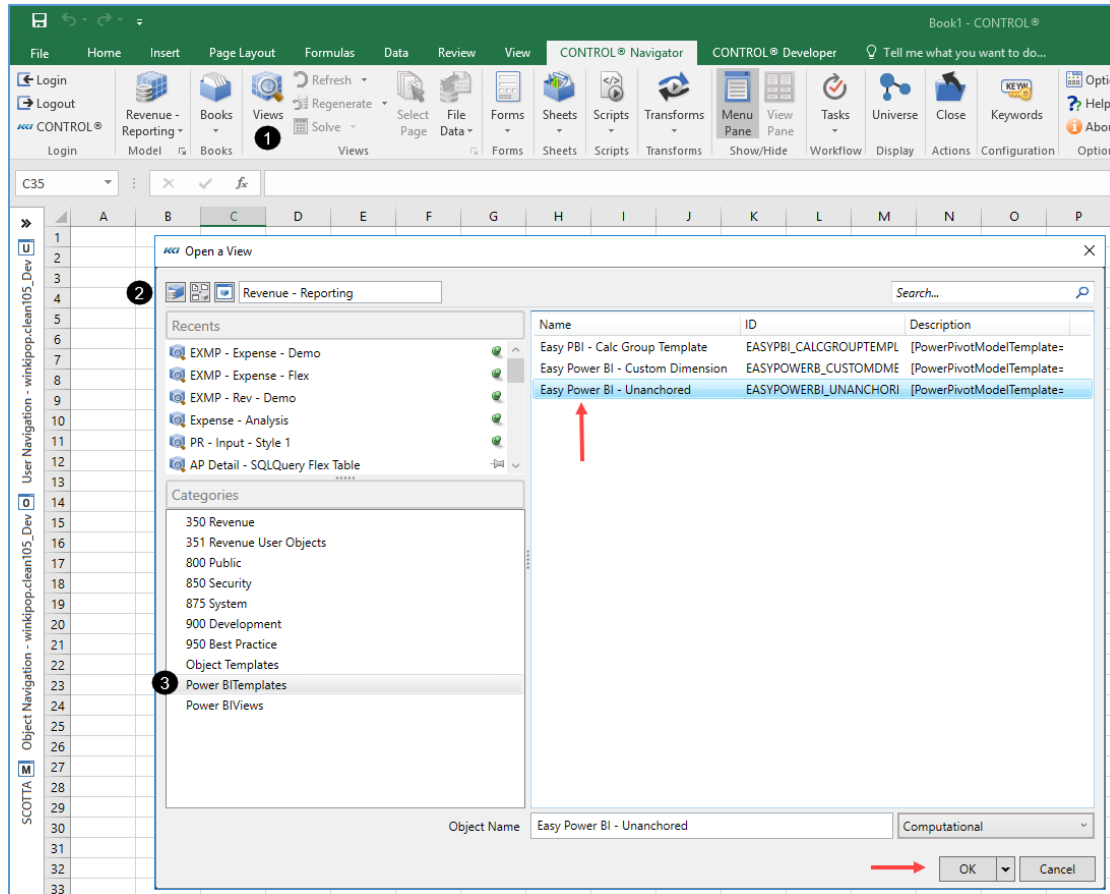
This release introduces a simple mechanism to get CONTROL® data into an Analysis Services model where it can be used with MS Power BI, a pivot table or other visualization tools.

The steps are simple:

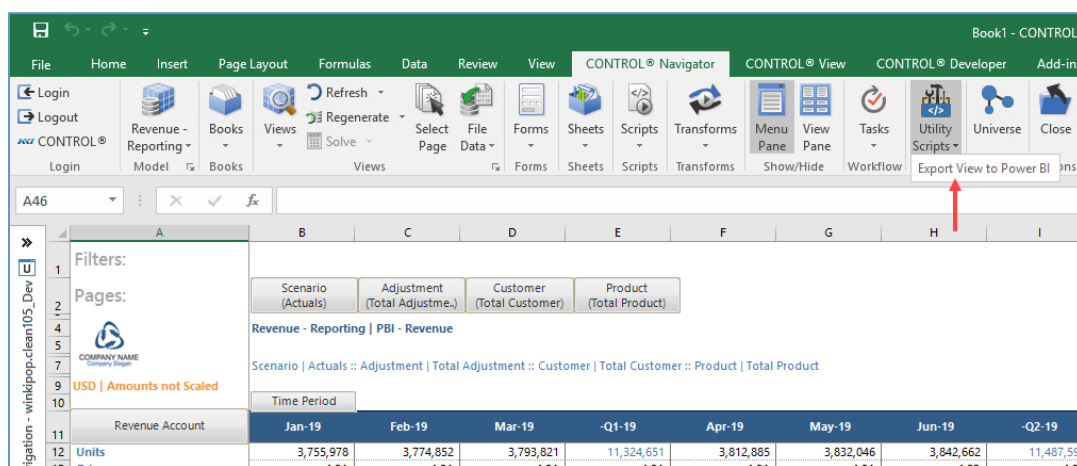
1. From the **CONTROL® Navigator** tab, in the **Views** group, click **Views**.
2. Click the **Select Model** icon  and select a *model*.
3. Click the **Power BI Templates** category, select one of the *views*, and then click **OK**.



# CONTROL® 10.5 Release Announcement



4. Modify the *view's filters* and branches as needed.
5. Click **Views** > **Save As** and give it a descriptive name.
6. On the **CONTROL® Navigator** ribbon, in the **Utilities** group, click the down arrow under **Utility Scripts**, and click **Export View to Power BI**.

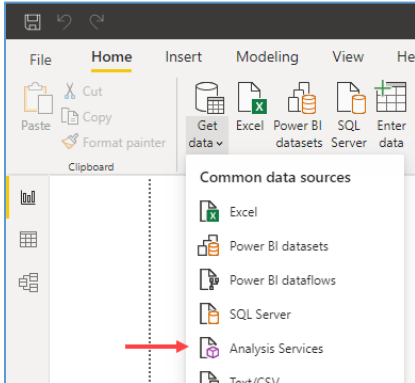


7. Launch MS Power BI Desktop.

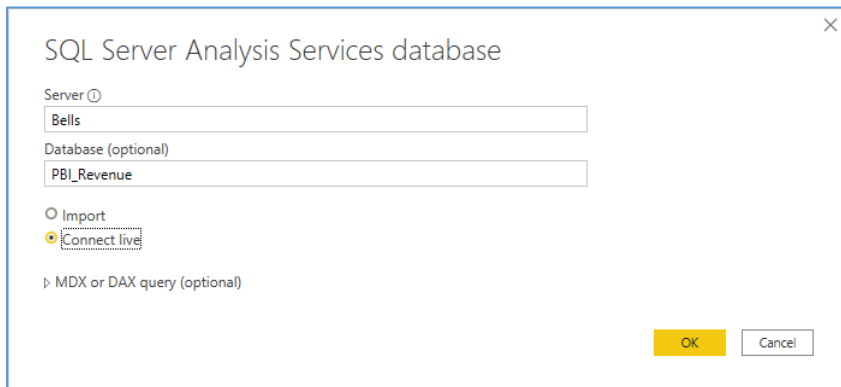


# CONTROL<sup>®</sup> 10.5 Release Announcement

8. On the **Home** tab, in the **Data** group, click **Get data** > **Analysis Services**.



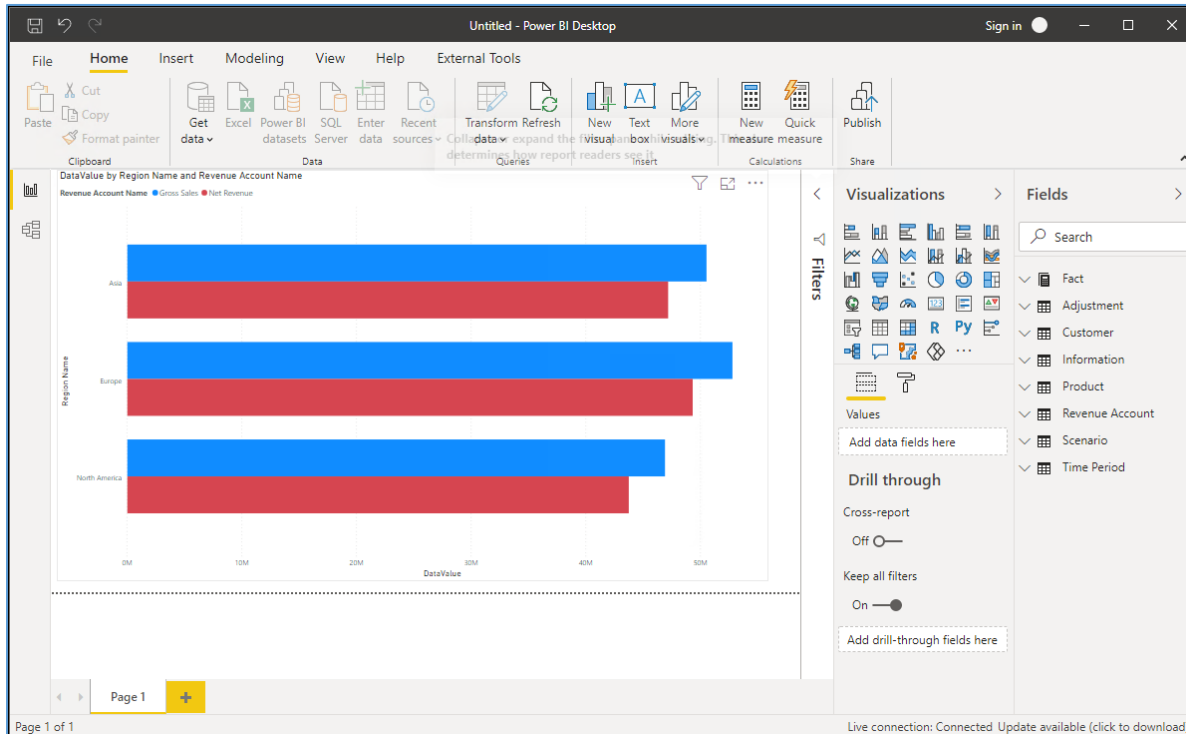
9. Enter your Analysis Services server name and database name (it's the same as the ID of the new view).



10. Start creating visualizations!



# CONTROL® 10.5 Release Announcement



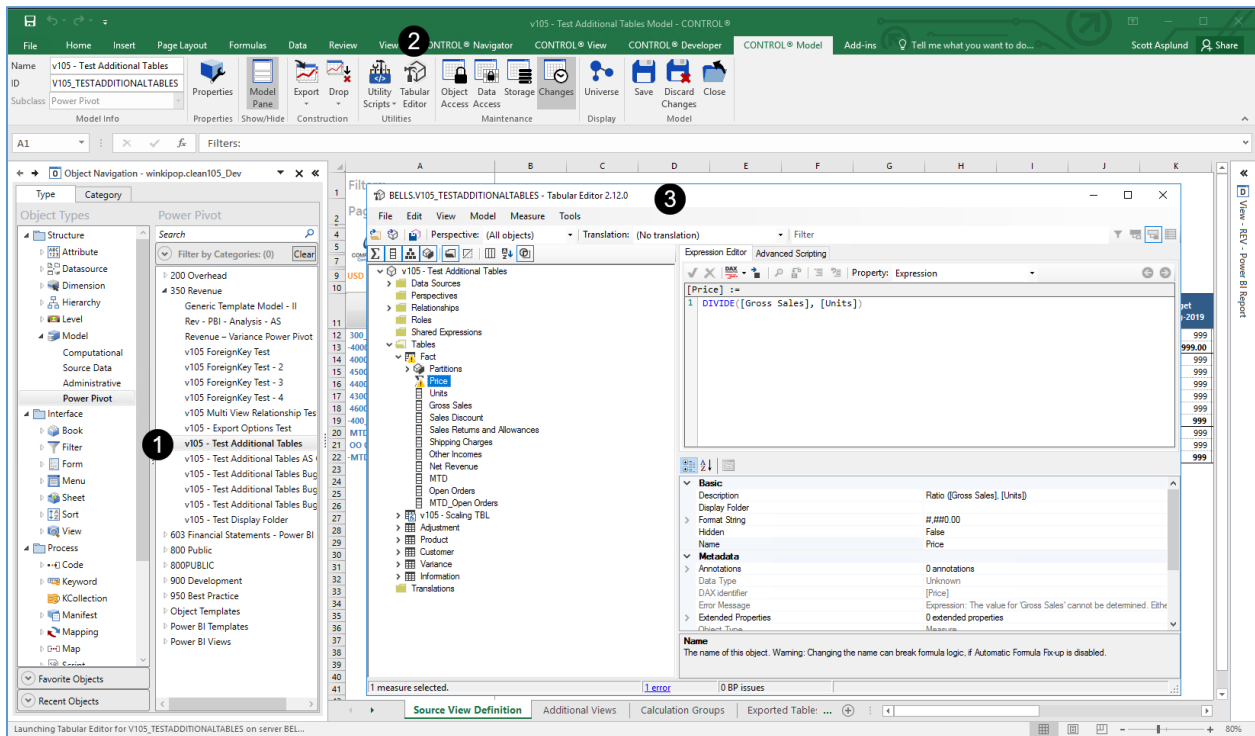
When data or meta-data changes, simply reopen the view and re-run the *script*. In most cases the visualizations will just keep working.

## Tabular Editor

CONTROL® continues to be a single, unified software solution with the ability to manage power pivot *models* using **Tabular Editor**. On the **CONTROL® Model** tab, in the **Utilities** group, select **Tabular Editor** to view and manage any of your AS models directly from this easy to use and popular tool. This enhanced integration with one of the industry's leading Power BI tools demonstrates how CONTROL® improves productivity by delivering end-to-end centralized management of all your data and reporting requirements.



# CONTROL® 10.5 Release Announcement



## Login performance

Several technical improvements in this release result in reduced time to load the CONTROL® add-ins and to perform the initializations associated with logging in.

## Enhancements for the Administrator

### AS Query datasources

With the growing popularity of CONTROL®'s integration with MS Power BI, KCI has recognized an opportunity to leverage analytic data managed externally in Microsoft Analysis Services (AS).

AS offers two fundamental structures – multi-dimensional and tabular, and two query languages – MDX and DAX. (The tabular representation and DAX are newer technologies, so they are growing in prominence.) The new **AS Query** *datasource* allows access to both structures and languages.

There are two important *datasource* properties:

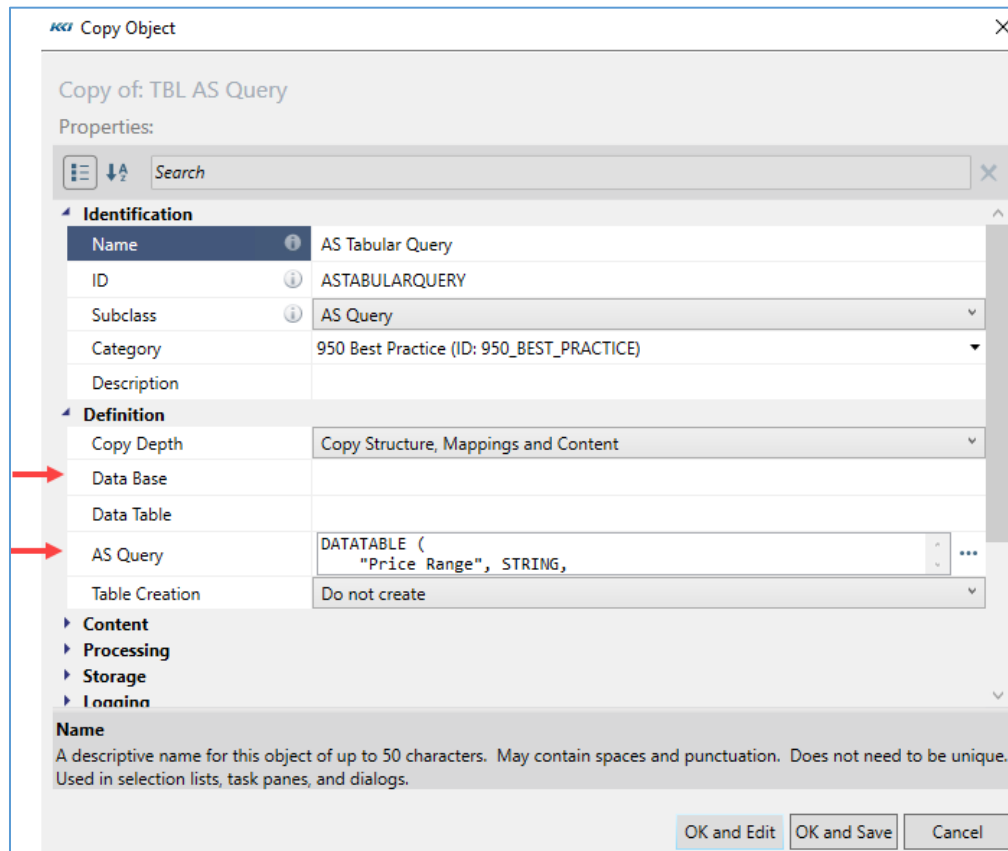
- **Data Base** – the Catalog or Server.Catalog where the Catalog equates to the database for AS Tabular. If the server is not supplied, the replacement value of the **&KCI\_ASServer** *keyword* is used to identify the server.



# CONTROL® 10.5 Release Announcement

**Note:** You can also use a *keyword* in this property.

- **AS Query** – the MDX or DAX query.



An AS Query *datasource* can be used in a *view* or *flex view*, but currently cannot be used in a source data *model*. *Mappings* can use the AS Query *datasource* as a source, but not as a target.

## New option for SQL Query datasources

The intent of this enhancement is to support queries on either the home (CONTROL®) database or any other ODBC accessible database without exposing CONTROL® or any other data to unauthorized access.

While specifically intended to support the **SQLQuery** function in *flex views*, these *datasources* can be viewed in non-*flex views*. However, as of this release their use in source data *models* and *mappings* is limited.





# CONTROL® 10.5 Release Announcement

The approach leverages CONTROL®'s object access *roles* to grant the privilege to query data from particular servers and databases to individual users or groups. The administrator of the target database can define the security to access specific tables or relational views.

**Note:** SQL Query *datasources* are not updatable but read only.

The **Data Base** property of a SQL *datasource* may now contain an ODBC connection string, implying that any queries are subject to the database privileges granted to the user ID associated with that connection string. The string can have a specific user ID and password or use Windows authentication.

Here are examples connecting with a specific user ID and password:

A SQL Server database:

```
CNTADM,CNTADM,DRIVER={SQL Server Native Client 11.0};SERVER=SUNSET;DATABASE=BAT_104
```

An Oracle database:

```
CNTADM,CNTADM,Driver={Oracle in OraClient11g_home1};Dbq=MAV_INST5
```

Here are examples connecting using the user's Windows account:

A SQL Server database:

```
„DRIVER={SQL Server Native Client 11.0};SERVER=SUNSET;DATABASE=BAT_104;Trusted_Connection=yes
```

An Oracle database:

```
„Driver={Oracle in OraClient11g_home1};Dbq=MAV_INST5
```

## Power BI integration enhancements

This release has several enhancements designed to improve the flexibility and functionality of the integration with Microsoft Power BI and SQL Server Analysis Services (AS).

These changes are intended to allow the creation of richer AS models and to preserve the customizations made either in CONTROL® or via an external tool (such as the popular SQLBI Tabular Editor).

These changes are extensive. For a comprehensive description, see the *Power BI Integration 10.5* document. Here are the highlights:

## Power pivot datasource properties

*Datasources* created to support power pivot *models*, for example, Fact, Dimension, and Information tables, include these new column properties that translate to AS properties:



# CONTROL® 10.5 Release Announcement

- **AS Usage** – to customize the list of column properties to those relevant to export processing.
  - A new **KPI** value to create a Key Performance Indicator in the AS model that includes options for Status, Target, and Trend.
- **AS Sort By Column** – to reference another column to override the sorting of members
- **AS Display Folder** – to group measures in a folder structure

## New tab in power pivot model edit book

There is a new **Exported Tables** tab in the power pivot *model* edit *book* that:

- Allows overrides to the table relationships defined in AS
- Simplifies the specification of the AS table name
- Supports addition of other relational sources to the AS model (including their relationships)
- Creates a calculated table in AS when the data source is an AS Query

## Materialization enhancements

- Materialized data tables on SQL Server 2019 and later will use a highly efficient Column Store Index
- Materialization without the foreign keys option simplifies debugging

## Export options

New export options support granular control over the lifecycle of the exported data:

- **Create or Replace**
  - **Relational Objects**
  - **All**
- **Update**
  - **Relational Objects**
  - **All**
- **Reprocess**
  - **Recreate** (AS objects)
  - **Process only** (for external customizations)

## Exported dimension tables

Exported *dimension* tables include new columns for non-aggregated *dimensions* (scenario and variable) to facilitate certain types of Power BI visualizations.

## Rule-based forecasting

Due to the growing popularity among our clients of generating forecasts based on trends and patterns, we have implemented several enhancements to *transforms*, *mappings*, and *scripts* to make it easier to implement these processes.

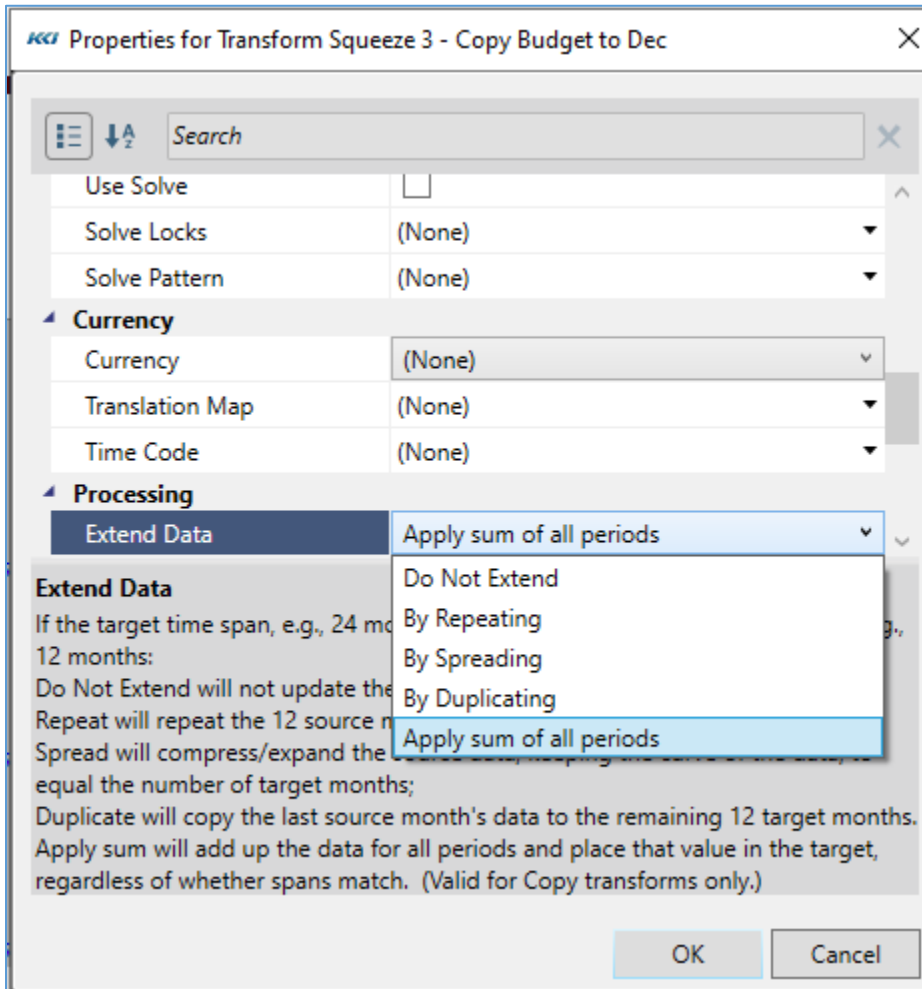


# CONTROL® 10.5 Release Announcement

## Transform enhancements

The following enhancements facilitate certain data manipulations:

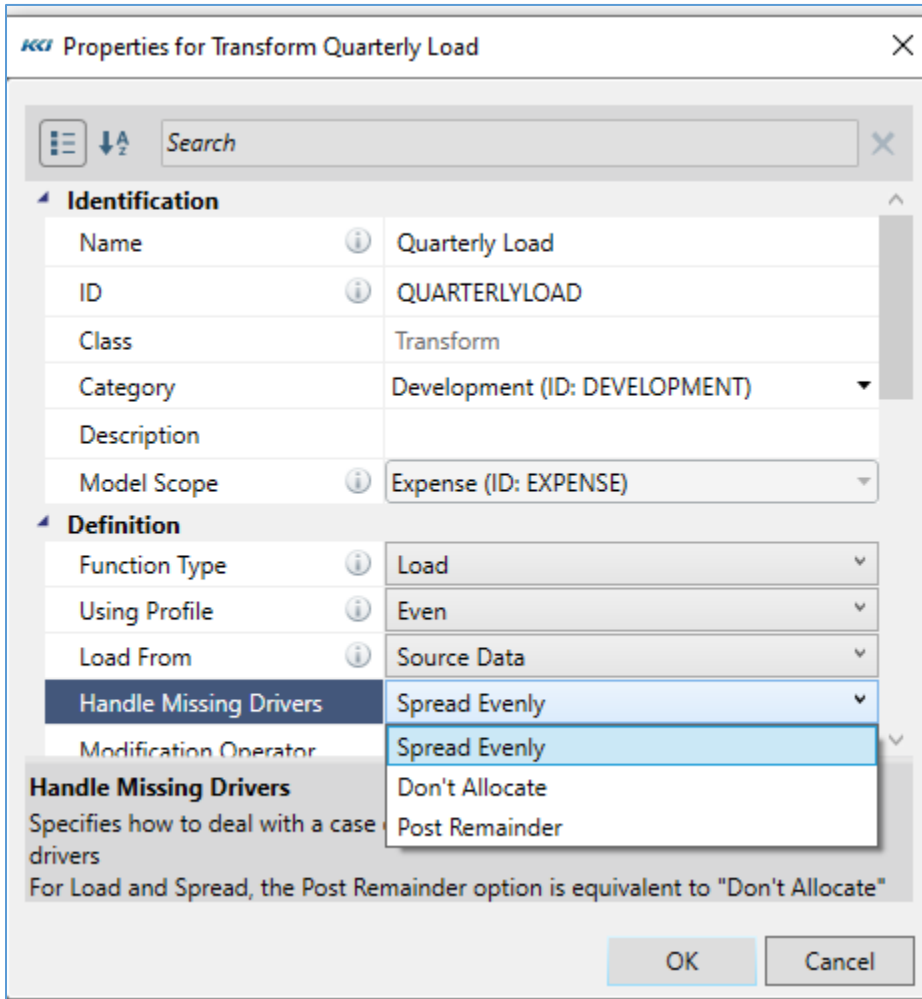
- There is a new option of the **Extend Data** property of a *Copy transform* that sums the source data for all periods before assigning the target values.



- The Load and Spread *transforms* now expose the **Handle Missing Drivers** property, which supports the even spreading of source data when the pattern data is identically zero.



# CONTROL® 10.5 Release Announcement



- There is a change of behavior of the Run *transform* in cases where one or more of the constituent *transforms* target the active *view*.
- For *views* with custom *dimensions*, there is improved logic to determine the appropriate non-custom sub-cube for source or target data. Earlier releases did not support most cases involving custom *views*.

## Overrides for transform and mapping steps in action scripts

This enhancement opens a world of possibilities for creating forecasts and budgets in new and imaginative ways by:

- Allowing any CONTROL-CONTROL or CONTROL-External data *mapping* to use the active *view* (or other open *view*) as its source or target
- Allowing any *transform*, except allocations, to use the active *view* (or any other open *view*) as its source, target, or pattern



# CONTROL<sup>®</sup> 10.5 Release Announcement

- Allowing both *mappings* and *transforms* to have their target data limited by a data *filter* (equivalent to an exception condition)

This is accomplished via six new properties. The **Script Item Properties** pane for *mapping* and *transform* actions enable this new functionality.

The screenshot displays the 'Script - REV - Data Filter Only' window. The 'Selected Content' pane on the left lists various actions, with 'Run Transform' selected. The main area shows a table of script items, with item 14, 'Run Transform', selected. The 'Script Item Properties' pane at the bottom is expanded, showing the following properties:

Property	Value
Keyword Collection In...	
Transform	REV - VW - Forecast Avg (ID: REV_VW_FOREC...
Target Override	Active View (ID: ACTIVEVIEW)
Source Override	Active View (ID: ACTIVEVIEW)
Pattern Override	Active View (ID: ACTIVEVIEW)
Data Filter	[EXMP_REV_ACCT 300_Units] == 2000
File Target View	DoNotFileData
Acquire Lock	

At the bottom of the pane, there is an 'Immediate update' checkbox and 'Update' and 'Maximize' buttons.



# CONTROL® 10.5 Release Announcement

## Target Override

The target override is typically the active *view*, but it may be any other updateable *view* on the target *model*. The results of the *mapping* or *transform* will update the appropriate data in the *view* but will not file the data to the database. The *view* can be a standard *view* with or without custom *dimensions*, or a flex *view*. If your template *sheet* contains the new **Transformed** style, the changed data will be highlighted.

If the target override is blank, the defined target of the *transform* or *mapping* is used. For *transforms*, a target override of **Model** will post the results to the database.

## Source Override

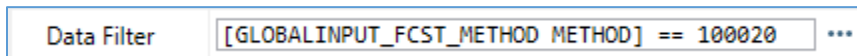
The source override can also be the active *view*, another *view*, or blank. If it is a *view*, the data in the *view* does not need to have been filed to the database. If blank, the source of the *transform* or *mapping* is used.

## Pattern Override

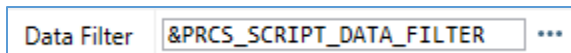
The pattern override can be specified for *transforms* (such as **Load** or **Spread**) that use a pattern. The rules are the same as those for the source override.

## Data Filter

The data *filter* optionally restricts the affected target data and has the same form and syntax as an exception condition in a *view*.



It can be either a pre-defined, or an ad hoc, condition. This property can also include *keywords*.



An important distinction between a data *filter* and a dimensional *filter* is that the data *filter* can select combinations of members of different *dimensions*. For example, any department with expenses for accounts greater than \$10,000 could be the Travel account in the Sales department and Salaries in the Finance department.

There are several options that can override the default behavior of the data *filter*, whose general syntax includes a prefix, to the left of a colon:

[ViewSelected And/Or ] [Dimensions(dimID1, dimID2,..)] [View(viewID)]:filterExpression

- The data *filter* is typically evaluated on the active *view*. However, if the prefix contains **View(viewID):**, then the *filter* is evaluated on the specified *view*.
- The data *filter* produces a list of member combinations that restrict the target of the *transform* or *mapping*. By default, the member combinations only include the members



# CONTROL<sup>®</sup> 10.5 Release Announcement

of the organization and variable *dimensions* in the *view*. If the prefix contains **Dimensions(dimID1, dimID2,...)**, then the member combinations of those *dimensions* apply.

- If the prefix begins with **ViewSelected**, the current selection in the active *view* (highlighted cells on the worksheet) add an additional constraint on the target data:
  - **ViewSelected And filterExpression** restricts the target to data that is both selected in the *view* and included in the data *filter*
  - **ViewSelected Or filterExpression** restricts the target to data that is either selected in the *view* or included in the data *filter*
  - **ViewSelected** with no **filterExpression** restricts the target to the selected member combinations in the *view*

**Note: ViewSelected** may only be used with data *filters* that target the active *view*.

The data *filter* always limits the result of the *transform* or *mapping*. If the target *role* or target *filters* do not include certain members, their data will be unaffected by the *script*.

## Data filter examples

Here are some examples of different data *filter* definitions using a *transform* to copy the year-to-date (Jan-19 thru Aug-19) averages to the remaining months of the year (Sep-19 thru Dec-19).

Here are the year-to-date averages:

Revenue Account	Product	Customer	YTD Average
Units	Product SKU 0001	Customer 0001	500
	Product SKU 0002	Customer 0002	400
	Product SKU 0003	Customer 0003	300
	Product SKU 0004	Customer 0004	100

Here is the target data defined with a data *filter* that highlights **Units** that equal **2000**:

Revenue Account	Product	Customer	Sep-19	Oct-19	Nov-19	Dec-19
Units	Product SKU 0001	Customer 0001	1,000	2,000	1,000	2,000
	Product SKU 0002	Customer 0002	1,000	2,000	1,000	2,000
	Product SKU 0003	Customer 0003	1,000	2,000	1,000	1,000
	Product SKU 0004	Customer 0004	1,000	1,000	1,000	2,000



# CONTROL® 10.5 Release Announcement

## Filter expression only

In this example, the data *filter* only includes a *filter* expression where **Units** equal **2000**:

```
Data Filter [EXMP_REV_ACCT 300_Units] == 2000 ...
```

By default, the member combinations only include the members of the organization and variable *dimensions* in the *view*. Therefore, in this case, all of the data combinations will update because every row meets the criteria, that is, every unit, product, and customer row include a value of **2000**.

Here is the result after running the *script* with this data *filter*:

Revenue Account	Product	Customer	Sep-19	Oct-19	Nov-19	Dec-19
Units	Product SKU 0001	Customer 0001	500	500	500	500
	Product SKU 0002	Customer 0002	400	400	400	400
	Product SKU 0003	Customer 0003	300	300	300	300
	Product SKU 0004	Customer 0004	100	100	100	100

## Time dimension and filter expression

In this example, the **Time Period** *dimension* further defines *filter* expression:

```
Data Filter Dimensions(TimePeriod):[EXMP_REV_ACCT 300_Units] == 2000
```

Therefore, in addition to selecting the rows where units, products, and customers equal **2000**, it also includes the months that meet that condition as well.

Here is the result after running the *script* with this data *filter*:

Revenue Account	Product	Customer	Sep-19	Oct-19	Nov-19	Dec-19
Units	Product SKU 0001	Customer 0001	1,000	500	1,000	500
	Product SKU 0002	Customer 0002	1,000	400	1,000	400
	Product SKU 0003	Customer 0003	1,000	300	1,000	300
	Product SKU 0004	Customer 0004	1,000	100	1,000	100

## Time and product dimensions with a filter expression

In this example, the **Time Period** and **Product** *dimensions* further define the *filter* expression:

```
Data Filter Dimensions(TimePeriod,Exmp_Prod):[EXMP_REV_ACCT 300_Units] == 2000
```

Therefore, it selects the combinations where units, products, and months equal **2000**.

Here is the result after running the *script* with this data *filter*:





# CONTROL<sup>®</sup> 10.5 Release Announcement

Revenue Account	Product	Customer	Sep-19	Oct-19	Nov-19	Dec-19
Units	Product SKU 0001	Customer 0001	1,000	500	1,000	500
	Product SKU 0002	Customer 0002	1,000	400	1,000	400
	Product SKU 0003	Customer 0003	1,000	300	1,000	1,000
	Product SKU 0004	Customer 0004	1,000	1,000	1,000	100

View selected with a filter expression

In this example, both **ViewSelected** and the *filter* expression are in the data *filter*:

Data Filter	ViewSelected AND [EXMP_REV_ACCT 300_Units] == 2000
-------------	--

If you select the second and third rows, then the *filter* expression only applies to the selected cells.

Revenue Account	Product	Customer	Sep-19	Oct-19	Nov-19	Dec-19
Units	Product SKU 0001	Customer 0001	1,000	2,000	1,000	2,000
	Product SKU 0002	Customer 0002	1,000	2,000	1,000	2,000
	Product SKU 0003	Customer 0003	1,000	2,000	1,000	1,000
	Product SKU 0004	Customer 0004	1,000	1,000	1,000	2,000

Here is the result after running the *script* with this data *filter*:

Revenue Account	Product	Customer	Sep-19	Oct-19	Nov-19	Dec-19
Units	Product SKU 0001	Customer 0001	1,000	2,000	1,000	2,000
	Product SKU 0002	Customer 0002	400	400	400	400
	Product SKU 0003	Customer 0003	300	300	300	300
	Product SKU 0004	Customer 0004	1,000	1,000	1,000	2,000

Time and product dimensions combined with view selected AND a filter expression

In this example, the data *filter* includes both **Time Period** and **Product dimensions**, the **ViewSelected**, and the *filter* expression:

Data Filter	Dimensions(TimePeriod, Exmp_Prod):ViewSelected AND [EXMP_REV_ACCT 300_Units] == 2000
-------------	--

If you select the second and third rows again, you will get this result after running the *script* with this data *filter*:



# CONTROL® 10.5 Release Announcement

Revenue Account	Product	Customer	Sep-19	Oct-19	Nov-19	Dec-19
Units	Product SKU 0001	Customer 0001	1,000	2,000	1,000	2,000
	Product SKU 0002	Customer 0002	1,000	400	1,000	400
	Product SKU 0003	Customer 0003	1,000	300	1,000	1,000
	Product SKU 0004	Customer 0004	1,000	1,000	1,000	2,000

Time and product dimensions combined with view selected OR a filter expression  
 In this example, the data *filter* includes both **Time Period** and **Product dimensions**, the **ViewSelected**, or the *filter* expression:

```
Data Filter Dimensions(TimePeriod, Exmp_Prod):ViewSelected OR [EXMP_REV_ACCT 300_Units] == 2000
```

If you select the second and third rows again, you will get this result after running the *script* with this data *filter*:

Revenue Account	Product	Customer	Sep-19	Oct-19	Nov-19	Dec-19
Units	Product SKU 0001	Customer 0001	1,000	500	1,000	500
	Product SKU 0002	Customer 0002	400	400	400	400
	Product SKU 0003	Customer 0003	300	300	300	300
	Product SKU 0004	Customer 0004	1,000	1,000	1,000	100

## File Target View

When there is a target override of **ActiveView** or a specified *view*, then **FileData** will file and commit data changes. The default option is **DoNotFileData**, which does not file or commit data changes.

File Target View	FileData
Acquire Lock	DoNotFileData
	FileData

## Acquire Lock

This property will set a resource lock to prevent two processes from executing simultaneously. The step in the *script* will fail if another process is running that conflicts with the *transform* or *mapping*.

To prevent this, enter the resource ID and the number of attempts followed by a wait time (in seconds) between each attempt. For example, if the *transform's* ID were **ActualUpdate**, then the syntax to attempt this step for three times, waiting five seconds between each attempt would be **ActualUpdate, 3, 5**. (Use a comma to separate each argument.)

```
Acquire Lock ActualUpdate, 3, 5
```



# CONTROL® 10.5 Release Announcement

## Column store model and datasource data tables

Microsoft has introduced a new feature in its SQL Server database called **column store indexes**. This option can provide a dramatic improvement in query performance and reduction in storage. KCI's testing with one customer application showed seven times improvement in average query speed and a 90% saving in storage space!

The performance and size advantages are most significant for large applications, in which a *model's* data table has tens of millions of records or more.

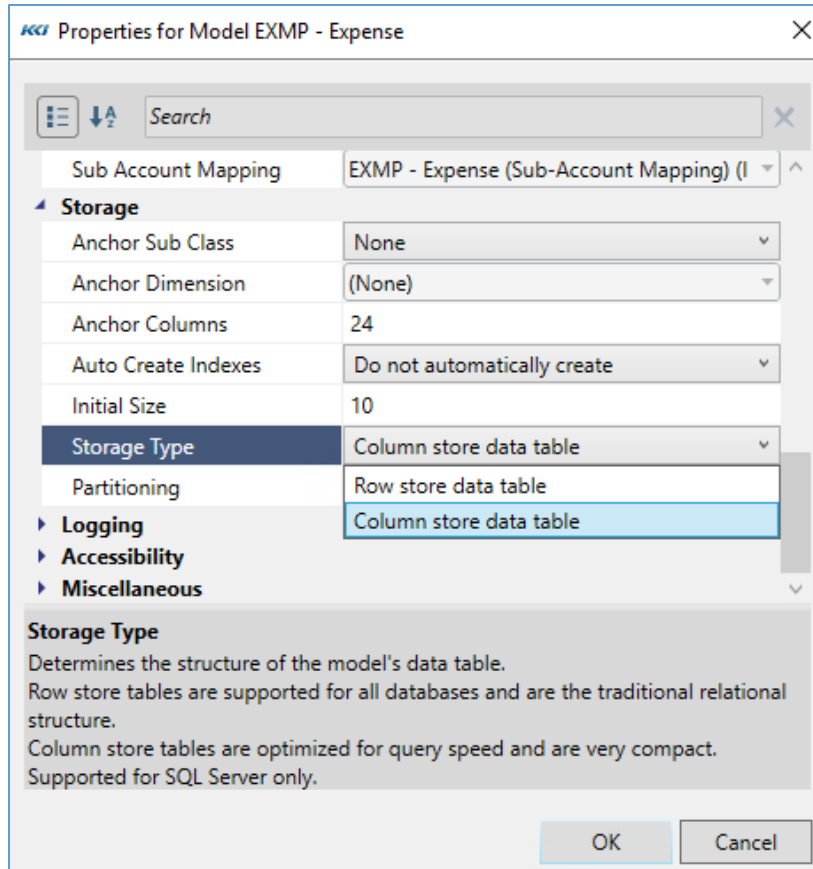
While the column store option is available in SQL Server 2017, our testing has shown benefits that are far more consistent in SQL Server 2019, so this option is available only to clients running SQL Server 2019, which applies to all Azure cloud customers.

To convert a *model* or *datasource* to use this option, you must login as an administrator with unlimited write privilege to the *model/datasource*. The change should take place when users are not accessing it.

From the **Object Navigation** task pane, right click on the *model* and click **Properties**.



# CONTROL® 10.5 Release Announcement



Change the **Storage Type** property to **Column store data table**. When you click **OK**, the table will convert automatically. For very large applications this change may take a few minutes or hours, so please be patient.

A program *script* will also convert a *model* by using the **ChangeStorageType** engine method. This new method accepts the following options:

- 1) Convert a specific table
- 2) Convert a specific *model* or *datasource*
- 3) Convert all computational *models* or all *datasources* of following subclass: **External**, **CONTROL-Managed**, **SQL Query**, and **Generated SQL**

The new **&KCI\_KeepOldStorageTable** *keyword* allows you to keep the "old" table in case you wish to revert to the original storage type, which is useful for testing. If this *keyword* is set to **Yes**, CONTROL® will rename the pre-converted table with a **\_RS** suffix for a regular table, or a **\_CS** suffix for a column store table. Please note that with this *keyword* enabled, any changes made to the data table after the conversion will NOT be present when you revert to the pre-converted table.



# CONTROL<sup>®</sup> 10.5 Release Announcement

KCI's testing with several customer applications resulted in no significant performance impact on saving data to the database. However, as the column store can have differing impact on update performance depending on the frequency and the size of the update, as well as the size and the makeup of the table, we recommend clients test using their own applications/environment to determine if the column store is beneficial in their specific use cases.

## Passwords for views and sheets

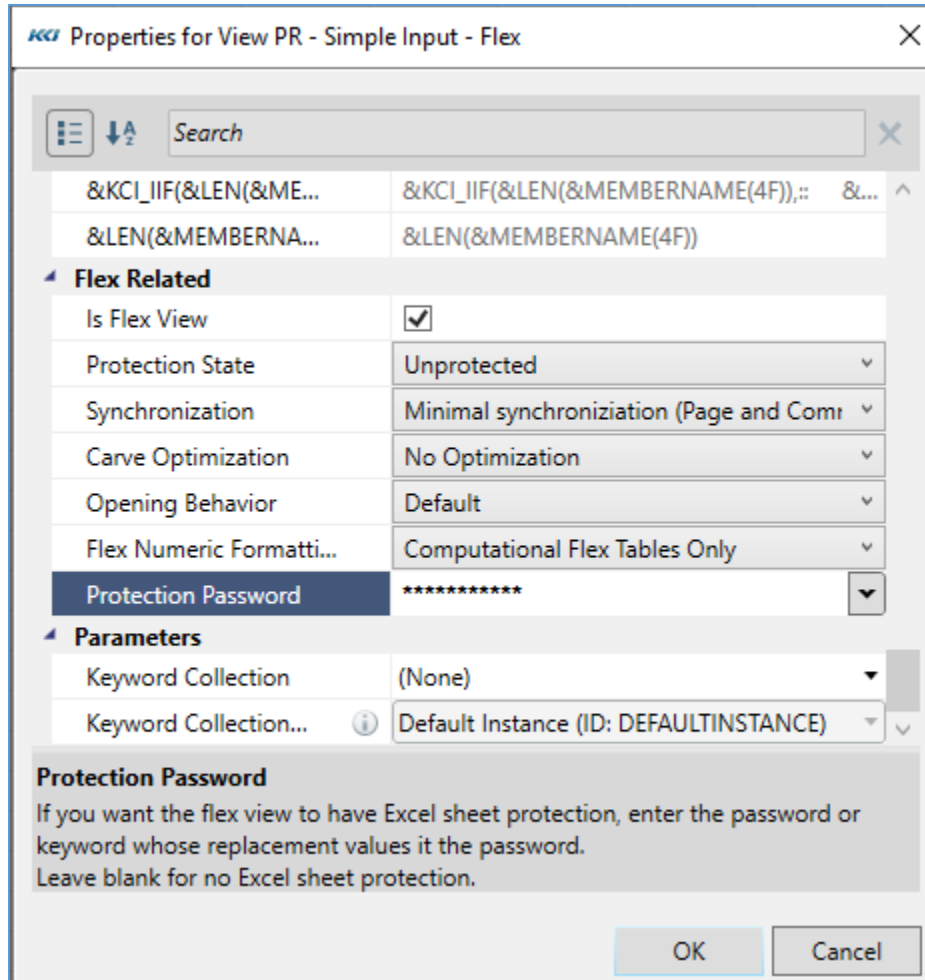
The mechanism for managing Excel worksheet protection for *scratchpads* and *flex views* has been dramatically improved in this release.

Previously, all *scratchpads* and *flex views* that used Excel's worksheet protection were protected using the same password specified by the **&KCI\_ProtectSheet** keyword. This was not terribly secure and did not accommodate use cases where multiple users were creating *scratchpads* and *flex views* and wanted to use their own password.

Starting with this release, these objects now have a **Protection Password** property.



# CONTROL® 10.5 Release Announcement



The property can be set to the value you want used to protect the worksheet from changes. The property can be set to a *keyword* if you want multiple flex views and *scratchpads* to be protected using the same password.

All objects that are stored in the database are saved without their Excel Protection Password. Therefore, if you want to change the password, you can do so in the property grid when the object is not open. Objects saved in Excel workbooks are saved with password protection in place. Changing the passwords for these objects must be coordinated with the external workbook and the object's **Protection Password** property.



# CONTROL® 10.5 Release Announcement

## Scenario-Hierarchy administrative model and view

To provide better transparency for the maintenance of *models* with extensive alternate *hierarchies*, there is a new administrative *model*, with a sample *view* shown below:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Filters:	Time (None)														
2	Pages:	Models (Revenue Copy-)														
4																
5																
7																
9																
10		Dimensions	Control													
11		Scenarios	Customer	Product												
		Hierarchy	Definition Type	Hierarchy	Definition Type	Hierarchy	Definition Type	Scenario Filter	Storage	Auto-Recalc	Recalc Transform	Hierarchy	Definition Type	Hierarchy	Definition Type	
13	2010 Budget	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
14	Annual Budget	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
15	Actual	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
16	Current Forecast	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
17	February 2008 Forecast	ACCOUNTMANAGER	Explicit	PRODUCT	None	REVENUEANALYSISBUDGET	Explicit					SCENARIOS	None	TIME_PERIOD	None	
18	March 2008 Forecast	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
19	April 2008 Forecast	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
20	May 2008 Forecast	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
21	2007 Budget	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
22	2008 Budget	CUSTOMER	None	PRODUCT	None	REVENUEANALYSISBUDGET	Explicit		Store Some	No	MAXRECALC2	SCENARIOS	None	TIME_PERIOD	None	
23	2009 Budget	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
24	What If 1	CUSTOMER	None	PRODUCT	None	REVENUE	Scenario Filter	AdHoc(SCENARIOS WHATIF1,WHATIF2)				SCENARIOS	None	TIME_PERIOD	None	
25	What If 2	CUSTOMER	None	PRODUCT	None	REVENUE	Scenario Filter	AdHoc(SCENARIOS WHATIF1,WHATIF2)				SCENARIOS	None	TIME_PERIOD	None	
26	Comments	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
27	Fcst B/(W) Budget	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
28	Actual B/(W) Budget	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
29	Long Range Plan (STEP)	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
30	Budget Var to STEP	CUSTOMER	None	PRODUCT	None	REVENUE	None					SCENARIOS	None	TIME_PERIOD	None	
31																
32																
33																
34																

Here are some important features of this *model* and *view*:

- The two *model dimensions* – **Model** and **Dimension/datasource** permit viewing *hierarchy* assignments across multiple *models*.
- The *view* is always read-only. Use the *model's* edit book to make any updates.
- The scenario *dimension* of the administrative *model* must be the scenario *dimension* of the *model(s)* selected in the *view* and should use the same scenario *hierarchy*. Most customers have a single scenario *dimension* and *hierarchy* that all *models* use. If you have more than one *dimension* or *hierarchy*, you should create a separate administrative *model* for each one.
- The **Definition Type** variable member indicates how the *hierarchy* assignment has been made:
  - **None** - there is no assignment, so the scenario uses the *dimension's* default *hierarchy*.
  - **Explicit** - the *hierarchy* is associated specifically to that scenario.
  - **Scenario Filter** – the scenario member is included in a *filter* and the *hierarchy* is associated with that *filter*.

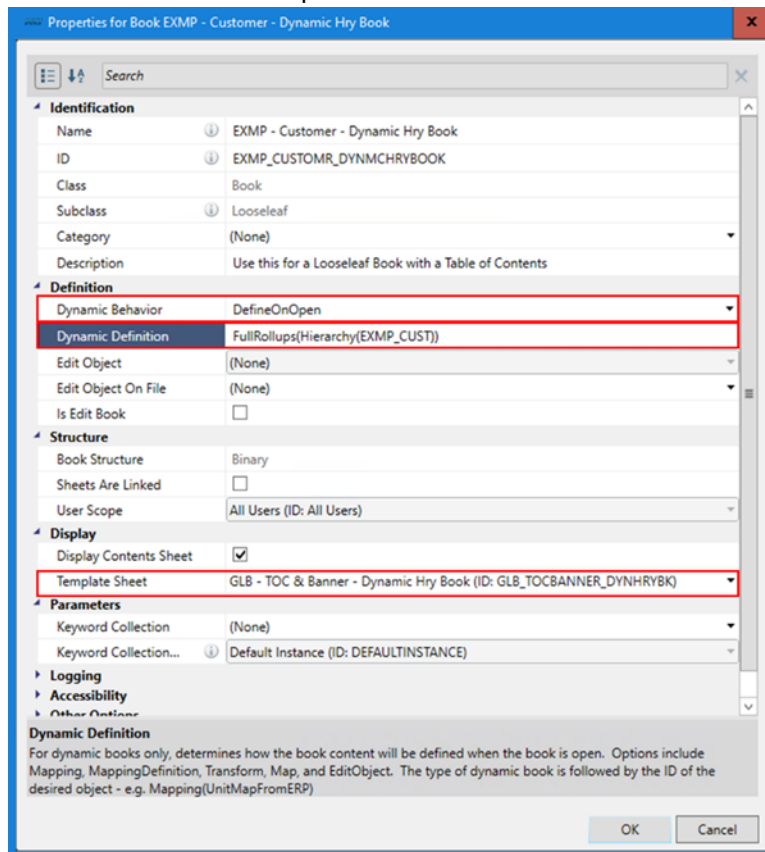


# CONTROL® 10.5 Release Announcement

## Improvements to hierarchy maintenance

There have been a number of improvements to the dynamic hierarchy *book* used to maintain complex organization dimensions:

- The method for creating a dynamic hierarchy *book* remains the same. The **Dynamic Definition** *book* property of **FullRollups (Hierarchy(hierarchyID))** will create a loose-leaf *book* for the organization hierarchy of a complex organization dimension that facilitates the review and maintenance of the reporting relationships across multiple branches. The ability of CONTROL® to create a loose-leaf *book* for more complex organization hierarchies with multiple branch structures has been improved in CONTROL® 10.5



- The manual insertion of a *sheet* object or a *view* object can now be performed into a dynamic hierarchy *book*. You can add your own *sheet* or *view* to a dynamic hierarchy *book* by editing the dynamic hierarchy *book* object and adding the sheet or view to the Content of the book. You can determine the position that the sheet or view appears in the book by filling out the "**Property Overrides**" property using the following "**Transient Property**" syntax.

[ **Position = 1** ] - Puts the Sheet in the first position (assumes origin 1)





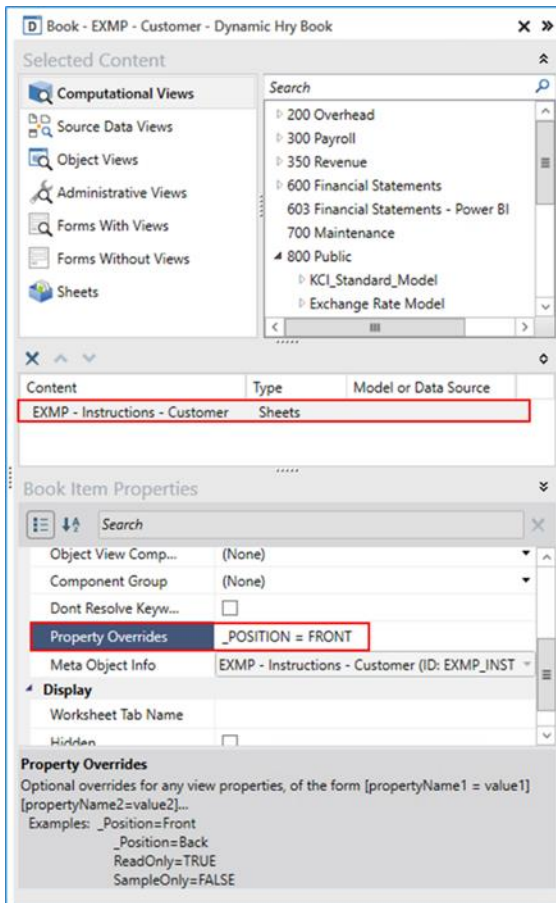
# CONTROL® 10.5 Release Announcement

[ **Position = Front** ] - Puts the Sheet in the first position (after Table of Contents)

[ **Position = Front + 1** ] - Puts the Sheet in the second position

[ **Position = Back** ] - Puts the Sheet last

[ **Position = Back - 1** ] - Puts the Sheet second from the last



# CONTROL® 10.5 Release Announcement

COMPANY NAME  
Company Logo

Print Book

BOOK: EXMP - Customer - Dynamic Hry Book

### TABLE OF CONTENTS

No.	Worksheet	Contents	Content Description	Model/Object Description
1)	<a href="#">Instructions-Dimension Diagrams</a>	EXMP - Instructions - Customer		
2)	<a href="#">Customer Rollups</a>	RootRollups1	All direct reporting relationships for Level EXMP_CUST: Customer FROM	EXMP_CUSTOMER (Standard Hierarchy)
3)	<a href="#">EXMP_LE - EXMP_CNTRY</a>	BranchRollups1	EXMP_LE: Legal Entity TO EXMP_CNTRY: Country	EXMP_CUSTOMER (Standard Hierarchy)
4)	<a href="#">Country Rollups</a>	RootRollups2	All direct reporting relationships for Level EXMP_CNTRY: Country FROM	EXMP_CUSTOMER (Standard Hierarchy)
5)	<a href="#">EXMP_REG - EXMP_TOT_CUST</a>	BranchRollups2	EXMP_REG: Region TO EXMP_TOT_CUST: Total Customer	EXMP_CUSTOMER (Standard Hierarchy)
6)	<a href="#">Full Hierarchy</a>	Full Rollups	Complete Bottom to Top Hierarchy with all Levels (Read Only)	EXMP_CUSTOMER (Standard Hierarchy)

TOC EXMP\_CUSTOMR\_DYNMCHRY Instructions-Dimension Diagrams Customer Rollups EXMP\_LE - EXMP\_CNTRY Country Rollups EXMP\_ ...

- You can now combine the member IDs and member names within the object views that are automatically created in a dynamic hierarchy *book*. Combining the member IDs and member names can make updating the hierarchies in the object views more user friendly. You can do this by changing the replacement value of the **ID\_COMBINE\_OR\_SEPARATE** keyword to either SEPARATE or COMBINE. By default, the **ID\_COMBINE\_OR\_SEPARATE** keyword is set to SEPARATE.

**ID\_COMBINE\_OR\_SEPARATE** keyword = **SEPARATE**

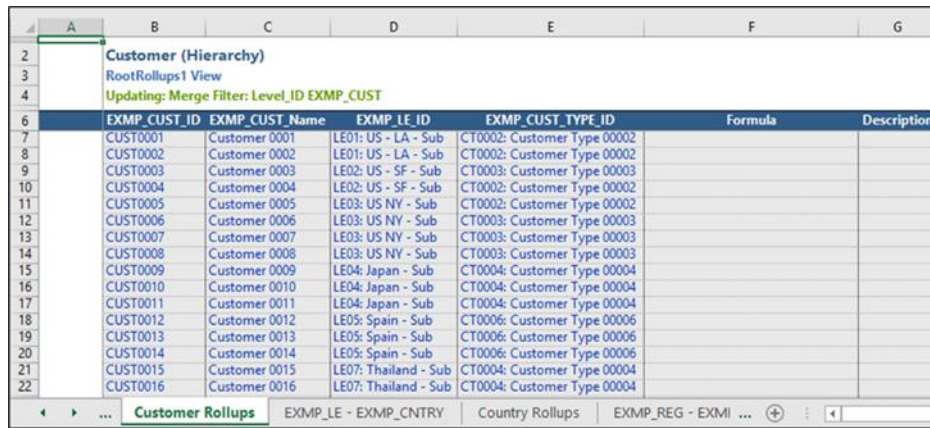
	A	B	C	D	E	F	G	H	I
2		Customer (Hierarchy)							
3		RootRollups1 View							
4		Updating: Merge Filter: Level_ID EXMP_CUST							
6		EXMP_CUST_ID	EXMP_CUST_Name	EXMP_LE_ID	EXMP_LE_Name	EXMP_CUST_TYPE_ID	EXMP_CUST_TYPE_Name	Formula	Description
7		CUST0001	Customer 0001	LE01	US - LA - Sub	CT0002	Customer Type 00002		
8		CUST0002	Customer 0002	LE01	US - LA - Sub	CT0002	Customer Type 00002		
9		CUST0003	Customer 0003	LE02	US - SF - Sub	CT0003	Customer Type 00003		
10		CUST0004	Customer 0004	LE02	US - SF - Sub	CT0002	Customer Type 00002		
11		CUST0005	Customer 0005	LE03	US NY - Sub	CT0002	Customer Type 00002		
12		CUST0006	Customer 0006	LE03	US NY - Sub	CT0003	Customer Type 00003		
13		CUST0007	Customer 0007	LE03	US NY - Sub	CT0003	Customer Type 00003		
14		CUST0008	Customer 0008	LE03	US NY - Sub	CT0003	Customer Type 00003		
15		CUST0009	Customer 0009	LE04	Japan - Sub	CT0004	Customer Type 00004		
16		CUST0010	Customer 0010	LE04	Japan - Sub	CT0004	Customer Type 00004		
17		CUST0011	Customer 0011	LE04	Japan - Sub	CT0004	Customer Type 00004		
18		CUST0012	Customer 0012	LE05	Spain - Sub	CT0006	Customer Type 00006		
19		CUST0013	Customer 0013	LE05	Spain - Sub	CT0006	Customer Type 00006		
20		CUST0014	Customer 0014	LE05	Spain - Sub	CT0006	Customer Type 00006		
21		CUST0015	Customer 0015	LE07	Thailand - Sub	CT0004	Customer Type 00004		
22		CUST0016	Customer 0016	LE07	Thailand - Sub	CT0004	Customer Type 00004		

TOC EXMP\_CUSTOMR\_DYNMCHRY Instructions-Dimension Diagrams Customer Rollups EXMP\_LE ...

**ID\_COMBINE\_OR\_SEPARATE** keyword = **COMBINE**



# CONTROL® 10.5 Release Announcement



The screenshot shows an Excel spreadsheet with the following data:

EXMP_CUST_ID	EXMP_CUST_Name	EXMP_LE_ID	EXMP_CUST_TYPE_ID	Formula	Description
CUST0001	Customer 0001	LE01: US - LA - Sub	CT0002: Customer Type 00002		
CUST0002	Customer 0002	LE01: US - LA - Sub	CT0002: Customer Type 00002		
CUST0003	Customer 0003	LE02: US - SF - Sub	CT0003: Customer Type 00003		
CUST0004	Customer 0004	LE02: US - SF - Sub	CT0002: Customer Type 00002		
CUST0005	Customer 0005	LE03: US NY - Sub	CT0002: Customer Type 00002		
CUST0006	Customer 0006	LE03: US NY - Sub	CT0003: Customer Type 00003		
CUST0007	Customer 0007	LE03: US NY - Sub	CT0003: Customer Type 00003		
CUST0008	Customer 0008	LE03: US NY - Sub	CT0003: Customer Type 00003		
CUST0009	Customer 0009	LE04: Japan - Sub	CT0004: Customer Type 00004		
CUST0010	Customer 0010	LE04: Japan - Sub	CT0004: Customer Type 00004		
CUST0011	Customer 0011	LE04: Japan - Sub	CT0004: Customer Type 00004		
CUST0012	Customer 0012	LE05: Spain - Sub	CT0006: Customer Type 00006		
CUST0013	Customer 0013	LE05: Spain - Sub	CT0006: Customer Type 00006		
CUST0014	Customer 0014	LE05: Spain - Sub	CT0006: Customer Type 00006		
CUST0015	Customer 0015	LE07: Thailand - Sub	CT0004: Customer Type 00004		
CUST0016	Customer 0016	LE07: Thailand - Sub	CT0004: Customer Type 00004		

## CONTROL® Web

After the release of CONTROL® 10.4, KCI introduced a new interface option targeted toward a broader audience of users. This introduction will bring you up to speed. For a more comprehensive description of this interface, see the “CONTROL Web 10.5” release announcement.

### Overview

As its name suggests, CONTROL® Web provides access to your CONTROL® data from a web browser. This provides a new alternative to CONTROL®’s Excel interface. CONTROL® Web supports modern web browsers, such as Edge, Chrome, or Safari, from a PC, phone, or tablet.

The primary focus of CONTROL® Web is to make current financial information instantly available to executives, managers, and analysts from anywhere, at any time, without having to extract, publish, or e-mail the information. Moreover, it is a self-service interface, so they can select, pivot, drill, and see transactional data to answer their own questions.

In addition, users who are not comfortable or familiar with Excel can use the web interface to input their budget or forecast data.

All this functionality is available with virtually no effort or cost because your current securitized CONTROL® views are available over the web.

### Features

CONTROL® Web delivers a significant subset of the Information Navigator capabilities with a simple, intuitive interface.

You can:



# CONTROL® 10.5 Release Announcement

- Open and navigate any *view*, except flex *views*, using your existing application menus
- Perform drill operations, including drill-to-source
- Update data and see the impact of your changes, and file to the database
- Enter row comments
- Save a personal *view* presentation (user scope)
- Save a *view* as a new *view*
- Run predefined *scripts*
- Print or export to Excel

A new user should be able to access data and perform basic analysis with virtually no training.

## New web options for 10.5

Based on early customer feedback, we added some simple *view* design functionality, integrated with the **View Operations** drawer:

The screenshot displays the CONTROL 10.5 interface. The main window shows a financial data table with columns for 'Time Period' (Jan-19, Feb-19, Mar-19, Apr-19) and rows for various expense categories. The table is titled 'Exp - Basic - FCST' and includes a note 'NOTRANSLATION | Amounts not Scaled'. The 'View Operations' drawer is open on the right, showing options for 'Dimensions, Branches and Filters', 'View Edge Assignments', 'Pages', 'Columns', and 'Rows'. The 'Dimensions' section is currently selected, showing 'Time Period' as the active dimension. The 'Branches' section shows 'M/Y' as the selected branch. The 'Columns' section shows 'M/Y' and 'CURRENT YEAR' as selected columns. The 'Rows' section shows 'All Levels' and 'A05 Operating Expenses' as selected rows.

	A	B	C	D	E
		Time Period			
		Jan-19	Feb-19	Mar-19	Apr-19
1					
2	<b>FS - Account</b>				
3	Full Time Equivalent	0	0	0	0
4	Base Compensation	0	0	0	0
5	Overtime	0	0	0	0
6	Employee Other Compensation	0	0	0	0
7	Payroll Tax Expense	0	0	0	0
8	Employee Benefit, Health Insurance	0	0	0	0
9	-Total Employee Compensation	0	0	0	0
10	Auto Expenses	10	24,458	29,581	39,704
11	Meals and Entertainment Expense	5,000	16,142	16,223	14,779
12	Travel Expense	48,671	48,915	49,161	49,408
13	-Travel & Entertainment	53,681	89,515	94,965	103,891
14	Bad Debt Expense	0	0	0	0
15	Bank Fees	24,335	24,458	24,581	24,704
16	Commissions and Fees Expense	0	0	0	0
17	Legal and Professional Expense	0	0	0	0
18	Office Expense	0	0	0	0
19	Supplies Expense, Office	0	0	0	0
20	Training Expense	0	0	0	0
21	-Supplies & Services	24,335	24,458	24,581	24,704
22	Telephone Expense	24,335	24,458	24,581	24,704
23	Cellular Communications	7,301	7,337	7,374	7,411
24	-Communications & IT	31,636	31,795	31,955	32,115
25	Freight Expense	73,006	73,373	73,742	74,112
26	Postage Expense	4,867	4,892	4,916	4,941
27	Printing	9,734	9,783	9,832	9,882
28	-Freight & Shipping	87,607	88,048	88,490	88,935
29	Rent or Lease Expense	36,503	36,687	36,871	37,056
30	-Occupancy	36,503	36,687	36,871	37,056
31	Maintenance Expense	0	0	0	0

The new *view* design options allow you to:

- Change the *filter* of any *dimension*



# CONTROL<sup>®</sup> 10.5 Release Announcement

- Change the branch of any *dimension*
- Add or remove *dimensions* from the *view*
- Add custom *dimensions*
- Save the resulting *view* as a new *view* for the current user

To keep with the level of CONTROL<sup>®</sup> sophistication low for the target users, the web interface only supports pre-defined *filters*, branches, and custom *dimensions*.

