The next generation CONTROL®

version 10.6

Power Query Integration

Table of Contents

Overview 2
Integration Points 2
Data Sources 3
Scripts 3
An Example Data Source 3
Loading and Editing Saved Queries 8
Combining Multiple Power Queries 10
Credentials 11
Some Technical Details 12
Reloading the CONTROL data source 12
Using keywords in the Power Query script 15
Customizing the Power Query data source 15
Memory and Performance considerations 18
Setup considerations 18

Overview

Microsoft Power Query is the key technology used in importing data into Microsoft's business intelligence architecture for presentation in Power BI and Excel.

It is an incredibly useful tool for connecting to external sources and:

- Cleansing the data
- Merging and appending multiple data sets
- Transforming the data to meet your needs
- Enriching the data with additional computations

Power Query really shines when the same data needs to be processed on a repetitive basis. Power Query is both enormously functional and very easy to use, so KCI has chosen to embrace it in CONTROL.

Here are the benefits we see in the financial planning and reporting domain:

- There is an awesome array of ways to manipulate, combine, and cleanse incoming data, many of which are frequently needed when preparing financial information for use in CONTROL
- Power Query supports access to a broad spectrum of sources including files, relational data, web pages, web data services, etc.
- In addition to these sources, many vendors of line of business systems (e.g., SalesForce and SAP) have created Power Query connectors
- Excel and Power BI users can leverage their familiarity with the capabilities of Power Query
- CONTROL customers may be able to leverage Power Queries they have already created for use with other applications
- Power Query greatly extends the scope of data (and meta-data) that can be mapped, drilled to, reported, and analyzed in CONTROL

Power Query fits right in with our goal of end-user empowerment that is at the heart of CONTROL.

Integration Points

Power Query uses a scripting language (called "M") which produces tabular data as a result or output.

Therefore, it is integrated as a sub-class into two CONTROL objects – scripts and data sources.

Data Sources

A Power Query data source refers to a Power Query script and a creates relational table that contains the result of processing that script.

The data source can be used in on-demand or dynamic mappings, in source data models, or as a drill-source. Because the result is a relational table, it can be queried with SQL expressions and joined to other tables. Since the Power Query data source is an object, it can be restricted for use or modification by CONTROL's object access roles. If used in a source data model, more granular access can be imposed using data access roles.

Scripts

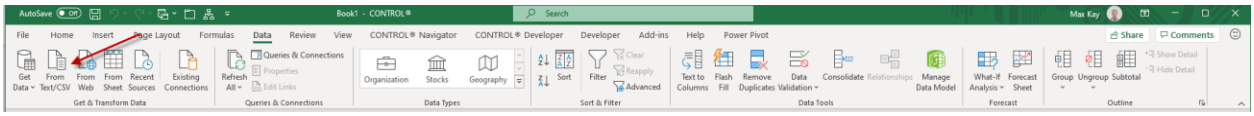
A Power Query script can be used to manipulate data in an open view or book, triggered by an event. Power Query scripts can also be used as component parts of a complex query or as a parametrized function, which will be described later in this document.

An Example Data Source

If you are supplied data from another application in the form of a CSV file which you want to use in CONTROL, the file may have a few issues with its content and format that you need to correct.

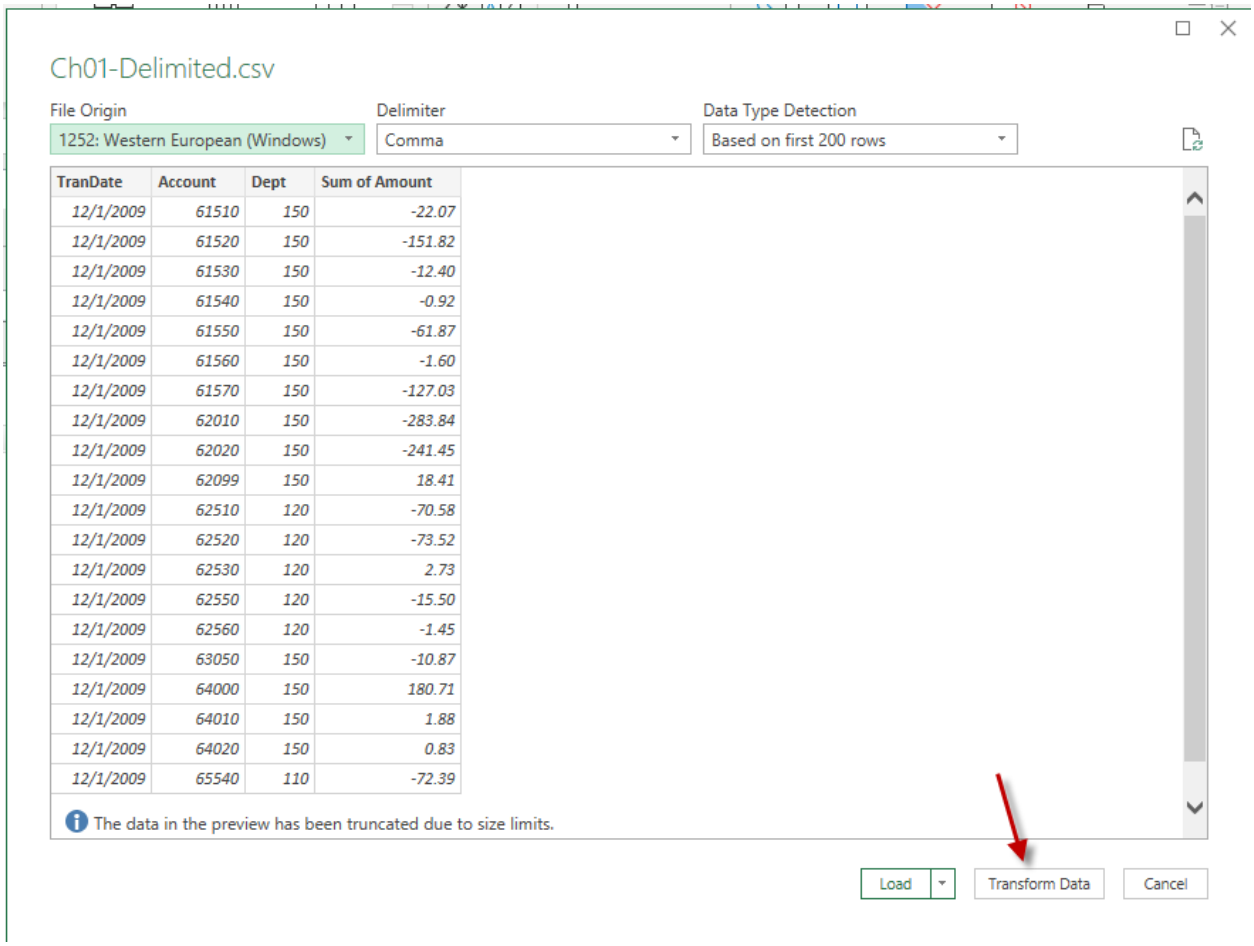
Instead of going through and manually editing the file, we will use Power Query to fix the file.

From the Excel Data ribbon, in the Get & Transform Data group, we'll click on From Text/CSV[1]:



Navigate to the desired file and then click Transform Data:

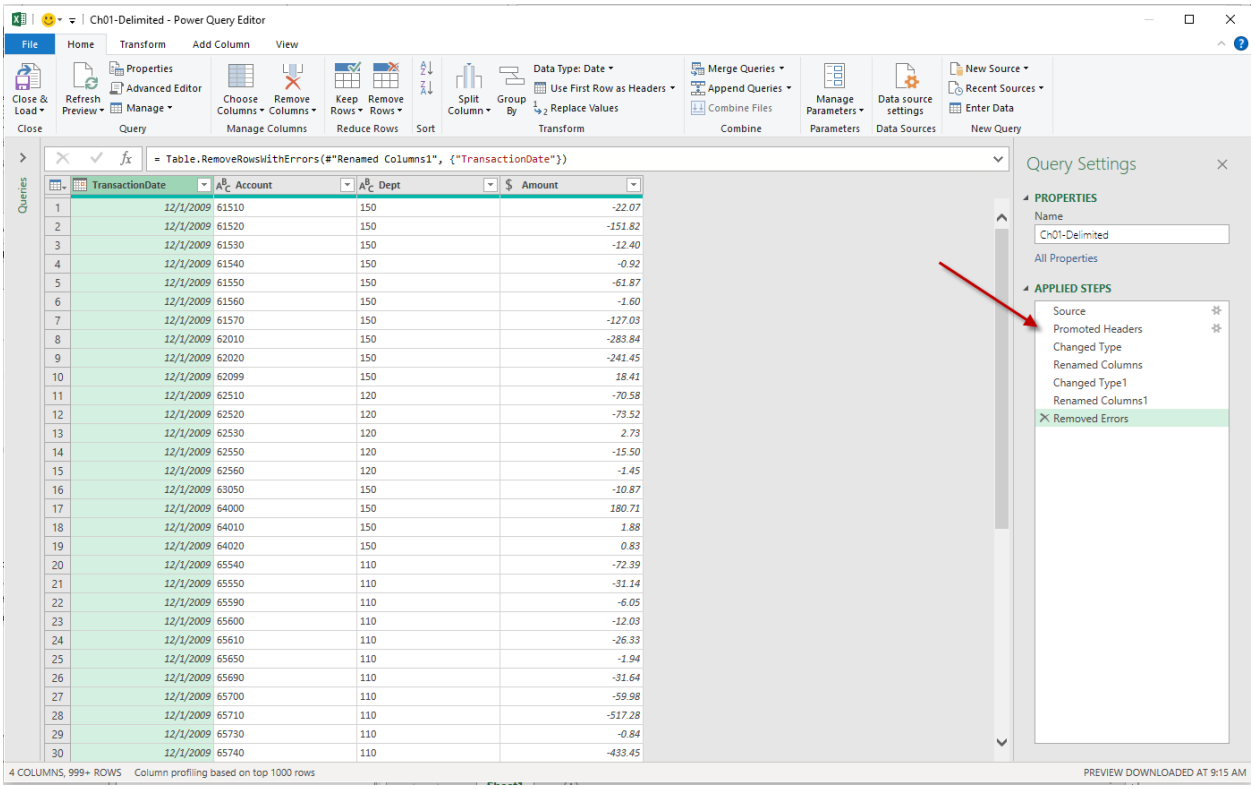
[1] The examples in this document were prepared using Office 365. Other versions of Excel offer the same functionality but the ribbon and supporting dialogs may look slightly different.



This puts us in the Power Query editor where we can address the issues:

- Change the name of the first column to "TransactionDate"
- Change the column type of Account and Department to Text
- Change the name of the last column to "Amount"
- Remove errors from the Data column (there was a total row in the file with not date which we do not need)

You can see the various steps in the task pane on the right:



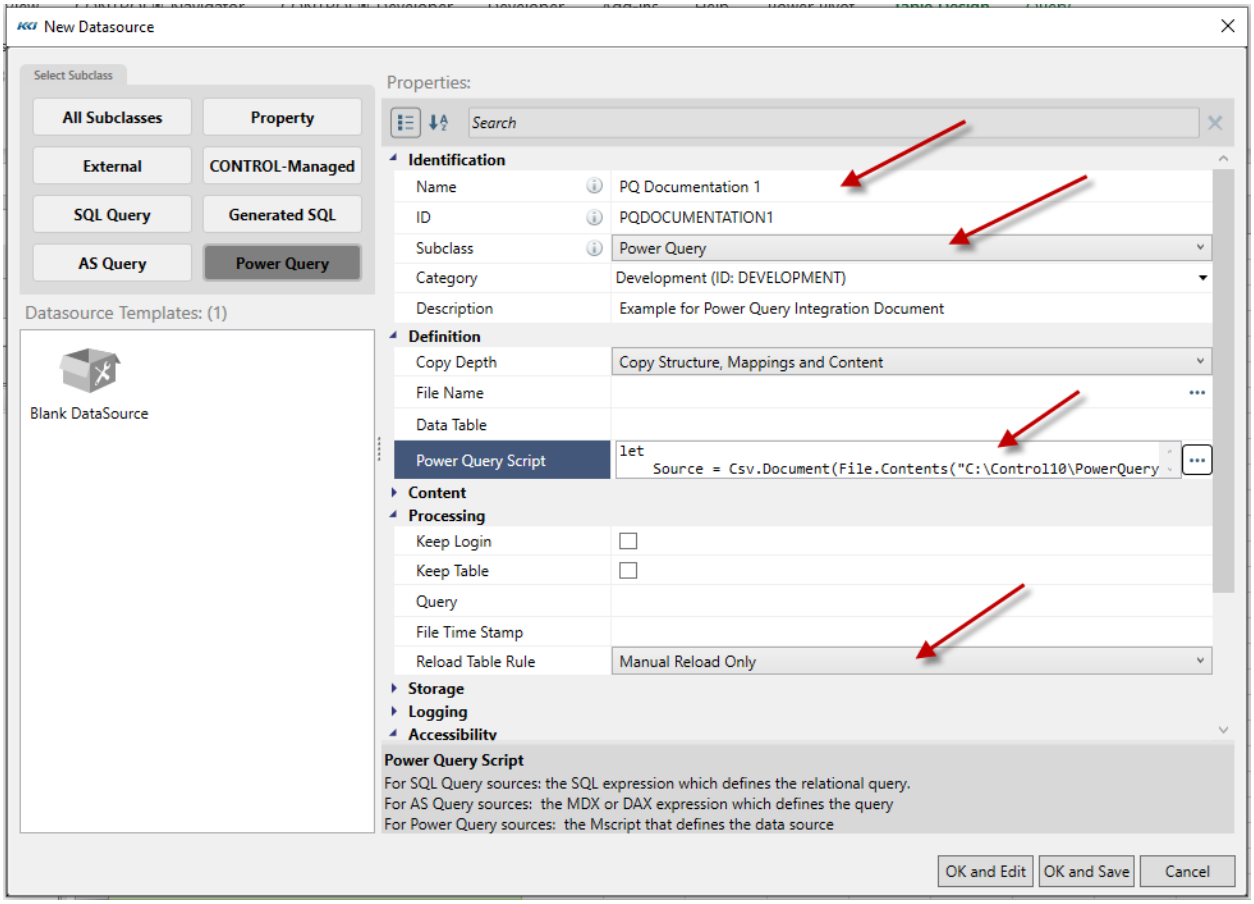
	TransactionDate	Account	Dept	Amount
1	12/1/2009	61510	150	-22.07
2	12/1/2009	61520	150	-151.82
3	12/1/2009	61530	150	-12.40
4	12/1/2009	61540	150	-0.92
5	12/1/2009	61550	150	-61.87
6	12/1/2009	61560	150	-1.60
7	12/1/2009	61570	150	-127.03
8	12/1/2009	62010	150	-283.84
9	12/1/2009	62020	150	-241.45
10	12/1/2009	62099	150	18.41
11	12/1/2009	62510	120	-70.58
12	12/1/2009	62520	120	-73.52
13	12/1/2009	62530	120	2.73
14	12/1/2009	62550	120	-15.50
15	12/1/2009	62560	120	-1.45
16	12/1/2009	63050	150	-10.87
17	12/1/2009	64000	150	180.71
18	12/1/2009	64010	150	1.88
19	12/1/2009	64020	150	0.83
20	12/1/2009	65540	110	-72.39
21	12/1/2009	65550	110	-31.14
22	12/1/2009	65590	110	-6.05
23	12/1/2009	65600	110	-12.03
24	12/1/2009	65610	110	-26.33
25	12/1/2009	65650	110	-1.94
26	12/1/2009	65690	110	-31.64
27	12/1/2009	65700	110	-59.98
28	12/1/2009	65710	110	-517.28
29	12/1/2009	65730	110	-0.84
30	12/1/2009	65740	110	-433.45

Clicking on the Advanced Editor button in the Query group on the ribbon will display the complete text of the Power Query:

This is the script that will define the Power Query data source. (Don't worry – you don't need to understand the cryptic language of the script.) While you're on this dialog, you might as well select the script and copy it to the Windows clipboard (Ctrl+c).

If you pick "Close and Load" on the left of the Power Query Editor ribbon, your transformed data will be loaded onto a worksheet. You do not need to load or save the data to use it as a CONTROL data source. The original file will be accessed and processed according to the content of the script.

To create the data source, go to the CONTROL Object Navigation pane, select data sources, right click, and New. Here is the completed dialog:



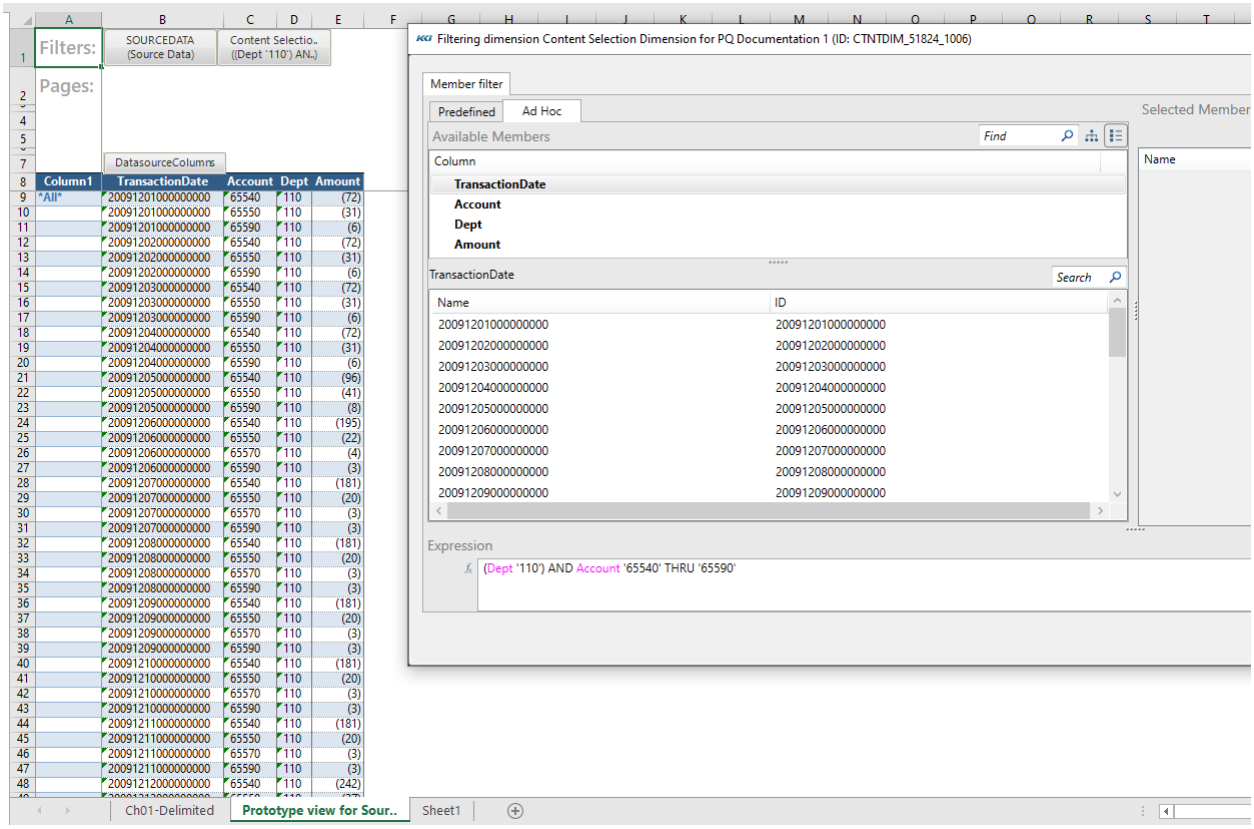
These are the important properties:

- The data source name, which will be translated to create the object ID.
- The subclass, which must be Power Query.
- The Power Query script, which defines the source(s) and transformations. Typically, you will create the script in the Power Query Editor and then copy and paste (Ctrl+v) it into this property.
- The Reload Table Rule, which determines the frequency and condition under which the relational table will be reloaded from the source(s). More on this later.

The name of the relational table is automatically derived from the data source's ID, but you can supply a different name if you wish.

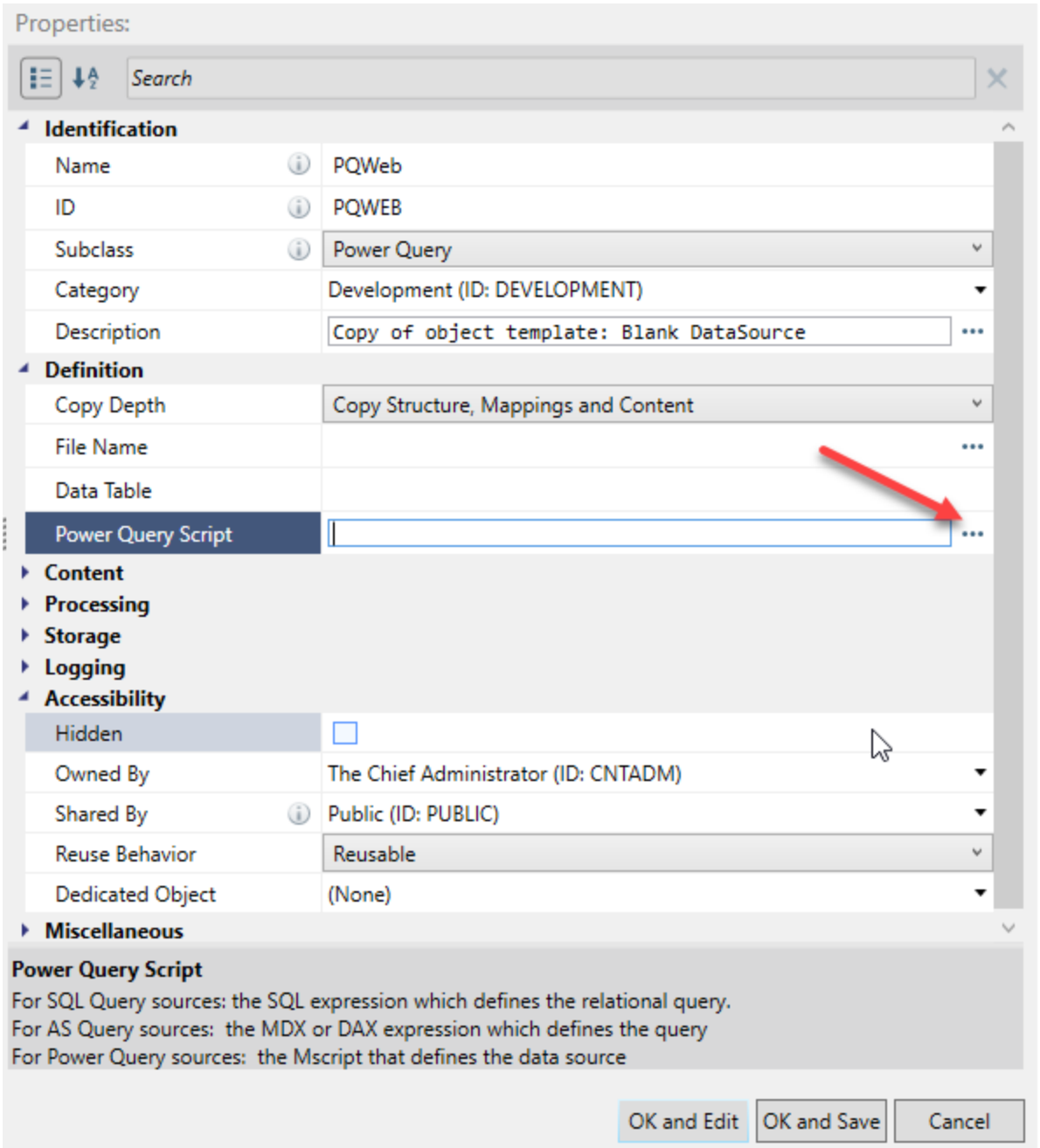
When you press OK and Save, the query will be processed and the relational table will be created and can be viewed, mapped, queried, etc. If there is a problem with processing the Power Query script, the problem will be reported (but the data source will still be created). You can fix the problem, open the data source edit book, and load the table using the Create button on the ribbon.

Here is a view of the data source filtered for a specific department and account range:

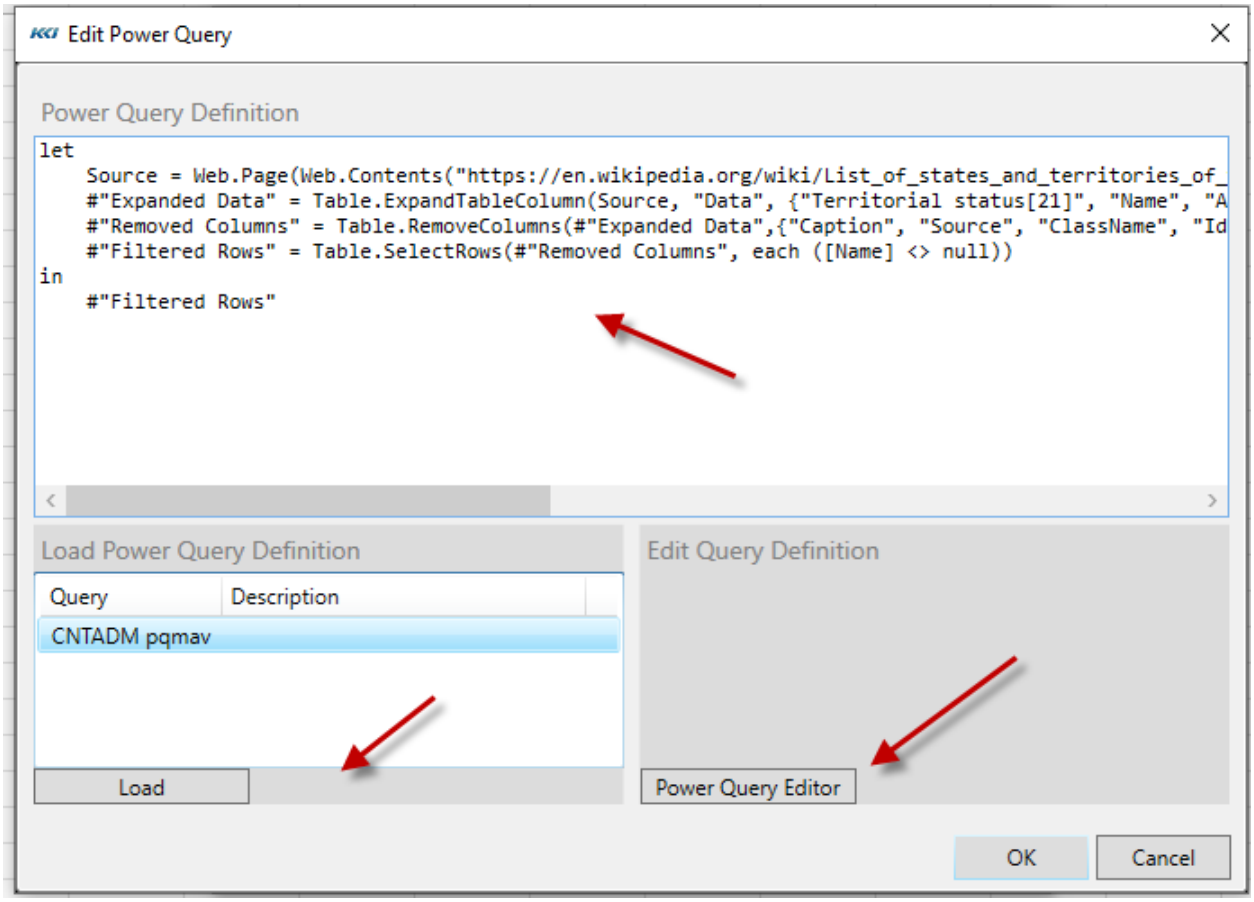


Loading and Editing Saved Queries

If you have a previously created query that you want to turn into a CONTROL data source, you need to first open the workbook containing the query you want to work with, then from the new object dialog or the properties dialog, you can click on the "…" button next to the Power Query Script property:



This will launch the Edit Power Query dialog for the first query in the active workbook:



You can:

- Directly edit the script currently loaded into the dialog in the upper text box
- Select a query from the list of queries in the active workbook and load it into the dialog
- Launch the Power Query Editor to make changes to the loaded script

Pressing OK will save the contents to the datasource.

Combining Multiple Power Queries

One very useful feature of Power Query is the ability to use the results of one query in another, much like a subroutine in a programming language. You can use this to combine results of queries from different sources or to break up a complicated process into manageable components.

In CONTROL, you can leverage this capability, and extend it further by re-using a Power Query script or data source in any number of Power Query data sources.

To use this capability, you must:

- Add each referenced query as either a Power Query data source or a Power Query script object.
- The object's name must be exactly the same (including case) as the name of the query.
- The name must be unique within the set of Power Query data sources and Power Query scripts.
- Any user reloading a data source's table or running a script must have object access to all the referenced scripts (at all levels of recursion).
- Add a Power Query script rather than a data source for a parameterized function
- Add a Power Query data source rather than a script if you would like to be able to use the intermediate result in mappings, views, etc.

All required objects will be detected and loaded automatically when you reload the table for a data source or execute a script.

Credentials

To process a query, access to the source or sources must be validated. For some types of sources attempting access without an explicitly designated authorization will produce an error.

Authorization is accomplished via a "Credential" which is specific to each type of source that is supported by Power Query.

Credentials are contained in the replacement values of CONTROL keywords. These keywords have an ID of "PowerQueryCredentialnn", where nn is a 2-digit number between 01 and 99.

The table below details the types of sources, available properties, and gives an example of the keyword replacement value:

Type	Properties	Example
Folder	Path	[Type=Folder] [Path=C:\]
File	Path	[Type=File] [Path=C:\PQ\MyFiles\taPMTransactionInsert.xml]
OData	Username Password Url	[Type=OData] [Username=Fred] [Password=Fred123] [Url=.....]
SQL	Username Password SQL	[Type=SQL] [Username=Joe] [Password=Joe123] [SQL=MyServerName;MyDatabaseName]
Web	Username Password Url	[Type=Web] [Username=Fred] [Password=Fred123] [Url=.....]

Note that Folder credentials authorize access to any files in the specified folder or any sub-folder, so if you require that any source files be placed on one drive and within a particular directory, you only need a single Folder credential.

When specifying the credentials for files and folders, if the folder path and or file name has spaces in it, the full path should be enclosed in double quotes. Additionally, if the folder or file path is pointing to a mapped network drive, the path should specified using UNC naming convention and not the mapped drive. Using UNC naming convention prevents issues when different CONTROL users have different network file mappings setup.

To simplify administration, CONTROL will recognize a keyword collection with the ID "PowerQueryCredentials", whose contents could contain a base set of "PowerQueryCredentialnn" keywords. While not required, this keyword collection would support centralized management of valid sources and free individual users from worrying about defining credential keywords.

Some Technical Details

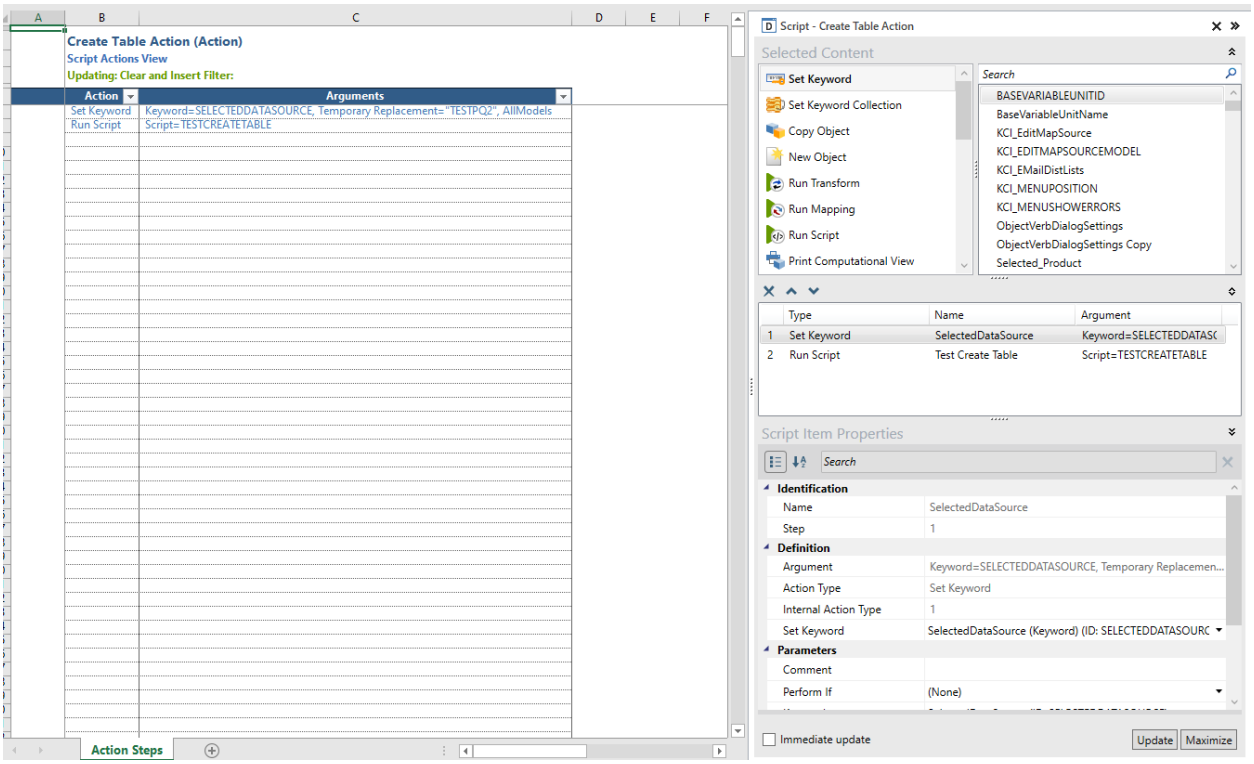
A few details must be considered when incorporating Power Query into your CONTROL processes.

Reloading the CONTROL data source

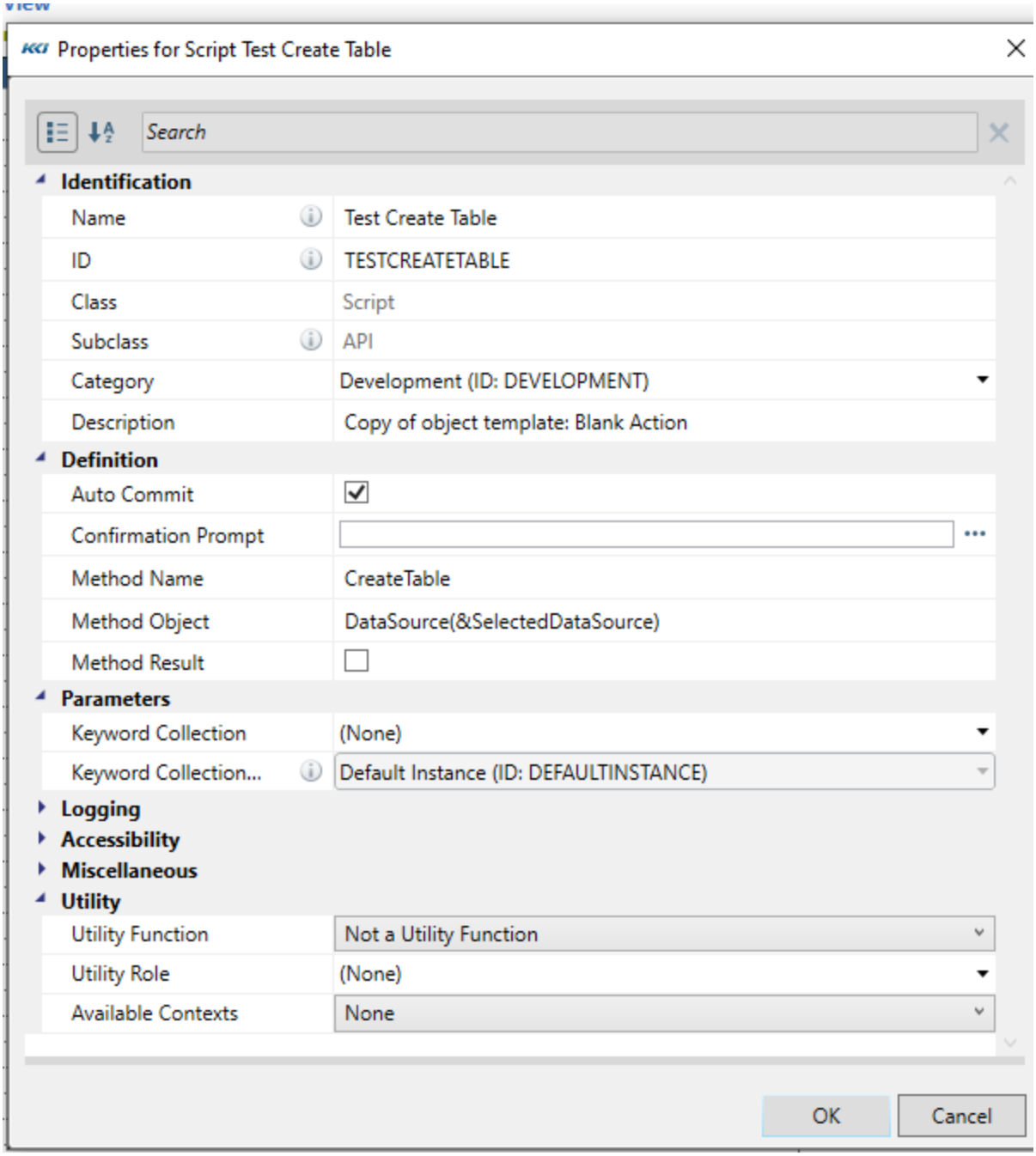
When you add a Power Query data source, define its properties, and click OK and Save, CONTROL will attempt to execute the Power Query script and create a relational table in the primary database containing all CONTROL objects. If the sources change, you can open the edit book for the data source and click Create Table in the ribbon to drop and re-create the table.

If this manual step is not convenient, the table creation can be scripted, using an action script which:

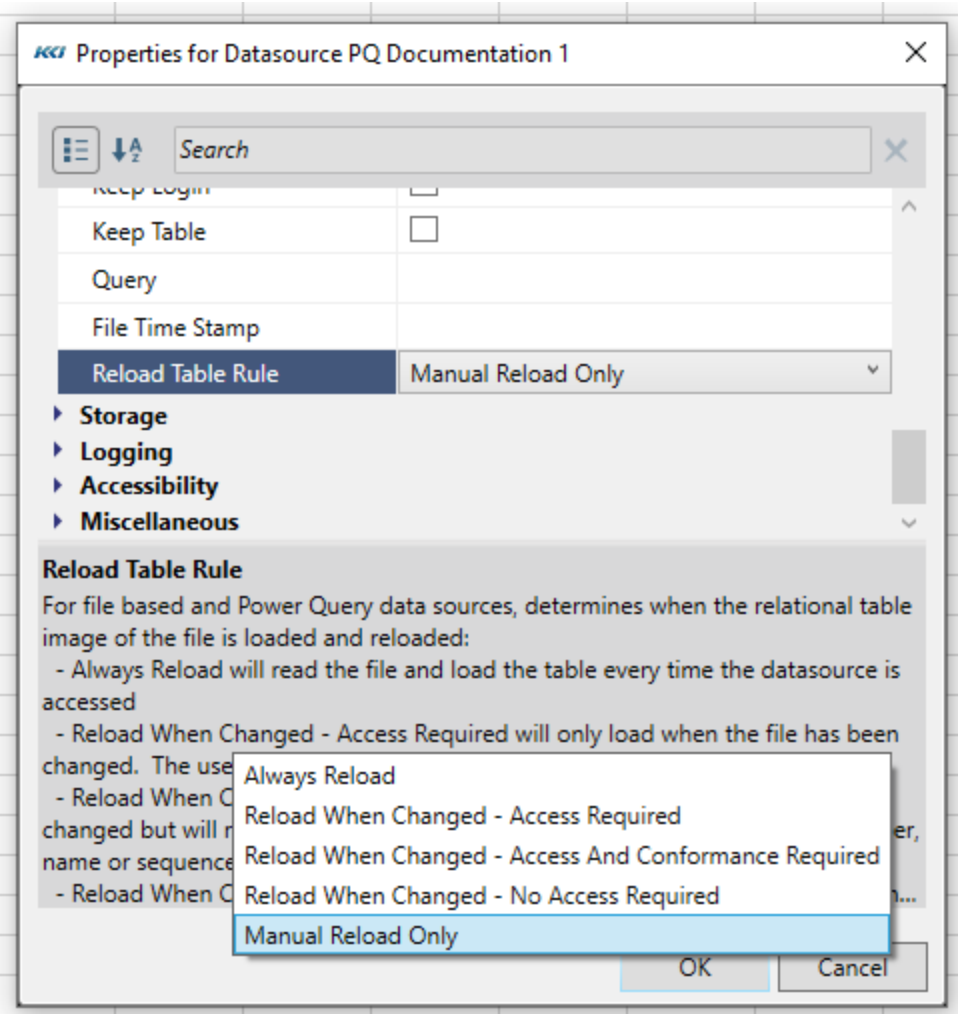
- Sets the keyword &SelectedDataSource to the target data source ID
- Calls an API script which performs a "CreateTable" command



The API script is included with the 10.6 update and its definition is shown below:



The table may also be recreated when it is referenced or when the timestamp on a file is changed by choosing a different option for the data source's Reload Table Rule:



If you specify any of the "Reload When Changed …" options, you must specify a File (File Name property) whose timestamp will be used to trigger the reload.

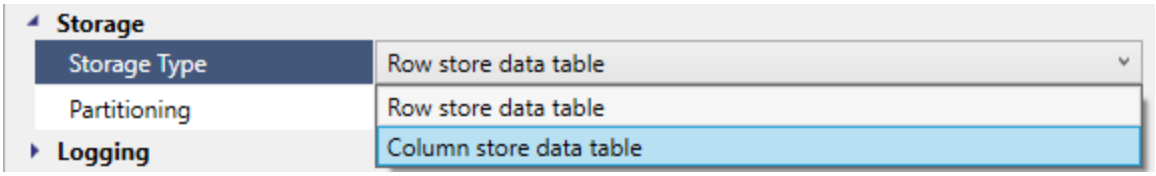
Using keywords in the Power Query script

Any CONTROL keywords in a Power Query script are resolved in the specific user and All Model scope prior to attempted execution of the script.

CONTROL uses the ampersand character (&) to indicate the presence of a keyword and Power Query's M language uses it for concatenation. Therefore, you should be careful of conflicts between keyword names and M variable names. Leaving a blank after the & in your M code will avoid unintended keyword replacement.

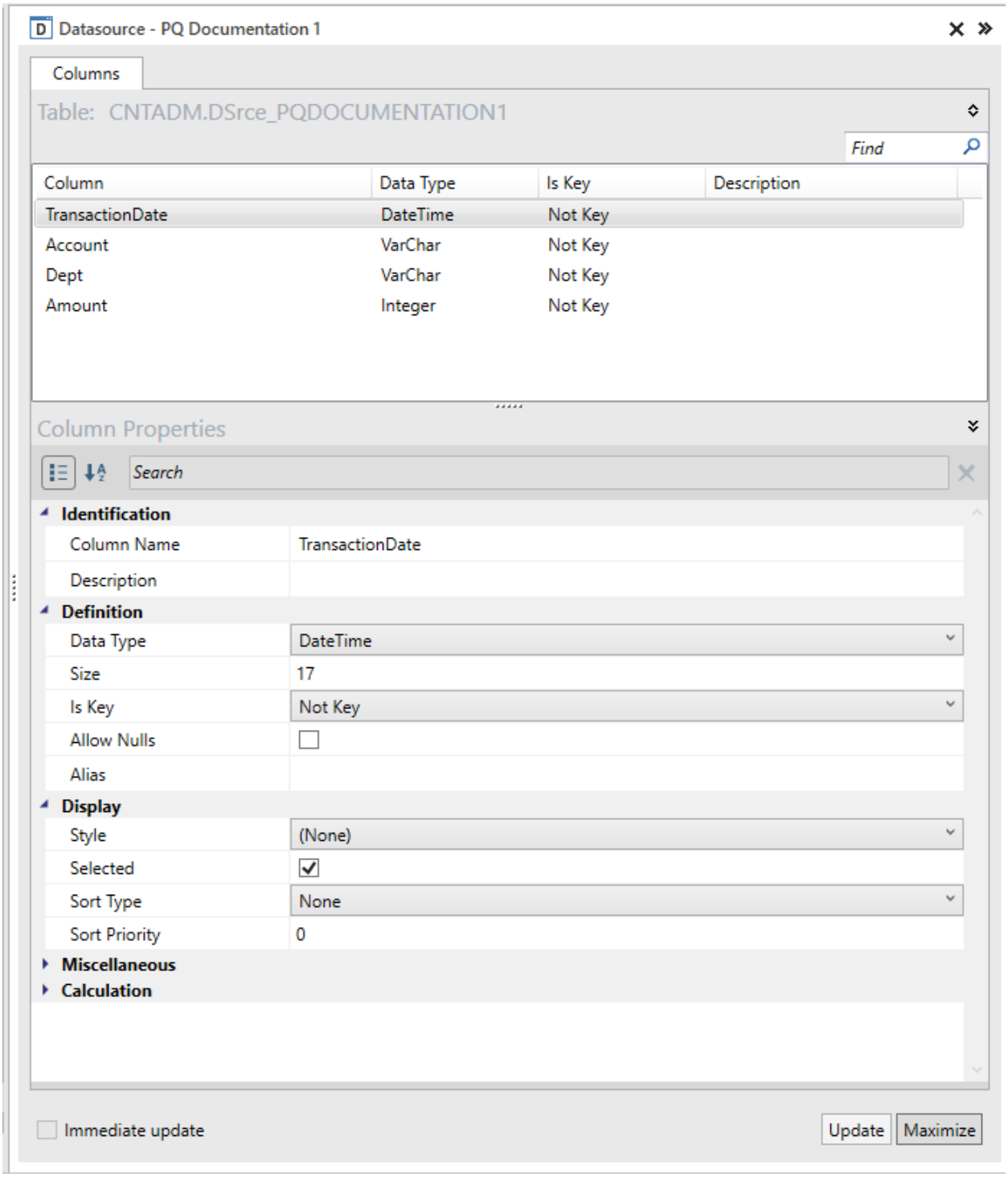
Customizing the Power Query data source

If you are on Microsoft SQL Server 2019 or later, you can choose to have the relational table created as a Column store data table.



These tables are very compact and efficient in read-only applications.

You may also change the following properties of the columns of the table, to make it more usable in the CONTROL application:



- Description – for documentation
- Is Key – for selection in flex views
- Alias – for titling or export to Analysis Services
- Style – for formatting in source data views

- Selected – to exclude from source data views
- Sort Type and Sort Priority – for ordering

Memory and Performance considerations

Power Query scripts are executed on all the records of the included data sources, so that the result is materialized in memory prior to being written to the target relational table. For very large files, this may consume more memory than using a delimited file, which CONTROL processes in smaller blocks.

Note that all subsequent functions which involve the data source, such as views and mappings, will only retrieve the data required for that function. Aggregation will be performed in the relational database and the memory impact should not be of concern.

Setup considerations

CONTROL Power Query integration requires that the Microsoft Power Query SDK files be installed on the CONTROL client system. The Microsoft Power Query SDK is available for download from Microsoft.

Download the Microsoft Power Query SDK using this link:

https://marketplace.visualstudio.com/items?itemName=Dakahn.PowerQuerySDK

The downloaded PowerQuerySdk.vsix file should be placed in the same directory as the CONTROL setup package. When the CONTROL setup package runs, it will detect the presence of the PowerQuerySdk.vsix file, and install the necessary files to allow CONTROL to integrate with Power Query.