

Qlikview Components Release Notes

Changes for V8

1. Qvc.IncrementalSetup-- new variable, `Qvc.Loader.v.StoreMaxModFieldValue`, indicates max ModField value should be stored in the QVD. This greatly speeds up the incremental reload process for large QVD files.

```
SET Qvc.Loader.v.StoreMaxModFieldValue = -1;
```

If specified as -1 (True), a new field named "`Qvc.MaxModFieldValue`" will be added to the QVD. This field will hold the max value of the ModField. On subsequent incremental reloads, only the first row of the QVD needs to be read to extract this value. The default is 0 (False). This feature must be turned on by setting the variable.

The `Qvc.MaxModFieldValue` field should be dropped later in a Transform or dashboard load of the QVD. The field is only required in the extract QVD.

2. Qvc.Calendar-- added code extension point for SET variables.

--Code Extension-- ExtSetVariables

Default file: `CalendarExtSetVariables.qvs`

Specified by variable: `Qvc.Calendar.v.ExtSetVariables`

The contents of the extension file are Included in the section that creates SetXXX period analysis variables. You may add any complete script statements. Usually this would be SET statements to create additional variables.

3. Added new Sub, Qvc.QvdFieldNumber. This sub provides the same functionality as the native QV "FieldNumber" function for QVD files.

```
CALL Qvc.QvdFieldNumber('Last Payment', '..\QVD\myfile.qvd', ['vReturn']);
```

Returns the number of the specified field within a QVD. If the field is not found, 0 is returned. The optional third parameter is a variable name used to return the result. If omitted, the default of "`Qvc.v.Return`" will be used.

4. Removed duplicate Qvc.Calendar & Qvc.CalendarFromField code from qvc.qvs. The duplicates were present in earlier versions. The duplicates caused no functional problem, but did make qvc.qvs larger than necessary.

Changes for V7

1. New SUB **Qvc.ExplodeTable** is an implementation of expanding a sparsely populated table into a symmetrical fully populated table. The problem and desired results are explained very well by Henric Cronström here: <http://community.qlikview.com/blogs/qlikviewdesignblog/2013/02/05/populating-a-sparsely-populated-field>. Additional Documentation on **Qvc.ExplodeTable** is available in Doc\ExpandInterval_Doc.pdf.
2. [Issue 34](#). Enhancement to Calendar Set variables such that only Calendar selections will be used to determine period ranges. For example:

Today's date is 5/13/2013. The data contains these sales:

Product,	Date
A,	5/10/2013
A,	5/12/2012

Prior to this enhancement, selection of Product "A" would constrain the YTD to 5/10. Therefore the sales on 5/12/2012 would be excluded from the Previous YTD (2012) sales. Now the Product selection will effectively be ignored and the Previous YTD reange will extend to 5/13/2012.

3. [Issue 31](#). Qvc.InspectTable has a new input variable "Qvc.v.QvdViewer.Path" that may be used to specify a full path to the QVD Viewer program. The background command window is also eliminated.
4. Documentation corrections.
5. [Issue 35](#). Qvc.Log.v.LogFileName incorrectly spelled as Qvc.Log.v.LogFile when searching for existing external log. This caused the Qvc.Log.v.KeepDays variable to be ineffective.

Changes for V6.1

1. New SUB **Qvc.InspectTable** provides for suspending script execution and viewing the contents of a table. InspectTable requires an installed QVD viewer such as QViewer. QViewer is available as

a free download from <http://www.easyqlik.com/>.

InspectTable will cause an extra command window to appear in addition to the Qviewer window. The command window will disappear when the Qviewer window is closed.

2. Fixed [Issue 30](#). Calendar QTD values incorrect when StartMonth causes quarters to cross calendar years.

Changes for V6

1. New SUB **Qvc.DbExtract** provides for table driven extract from SQL to QVD of multiple tables. A parameter table contains one row for each table to be extracted. Both Full and Delta load are supported.
2. **Qvc.Calendar**. [Issue 26](#). Additional Set variables have been created that represents the modifier portion of the Set expression. This allows for additions to the expression. The new variables are named vSetXxxModifier.
 - a. Using the classic VSetXxx variable:

```
Sum($ (vSetYTD) Sales) // YTD Sales
```

- b. The vSetXxxModifier represents the modifier arguments. Note that you must add the “{< >}” syntax of the set.

```
Sum({< $ (vSetYTDModifier), Region={US}>} Sales) // YTD Sales, US only
```

3. [Issue 29](#). **Qvc.PopulateVariables**. Add [] around variable names to allow for special characters and spaces.
4. [Issue 28](#). **Qvc.ListFiles**. Default filemask missing.
5. [Issue 27](#). **Qvc.LogWarning**. Documentation correction.
6. [Issue 25](#). **Qvc.AvailableUpdate**. Documentation correction