

Qlikview Components Release Notes

Changes for V10.2

- Closed [Issue 25](#). Added a "Year-Quarter" field to Qvc.Calendar and Qvc.CalendarFromField.
*** **Warning:** If you have already created your own "Year-Quarter" field using the Calendar Extension, you will receive a script reload error due to the duplicate field. You should remove the "Year-Quarter" from your extension file. ***
- Closed [Issue 35](#), Calendar Code Extension not working in Sense.
- Added Romanian (RO) Language file. (Credit to Cotiso Hanganu).
- Started adding Qlik Sense examples. Qlik Sense examples are contained in the "Sense Examples" directory. The specific qvf example file must be dragged to the Qlik Sense desktop to be viewed. If you want to reload a Sense example, you must modify the script "QVC" Connection Folder to point to your Qlikview Components installation.

Changes for V10.1

- Closed [issue 23](#). Non-English characters not supported in log file **Qvc.Log**. (Thanks to Alkopfer)
- **Qvc.Log**, new variable **Qvc.Log.v.WriteToQvLog**, indicates if the logging should also be written to the QlikView/Sense application log file and script progress window.
- Closed [issue 24](#). **Qvc.IncrementalSetup** failing when QVD exists but is empty. (Thanks to Luca Jonathan Panetta)
- **Qvc.SegmentedStore** -- New segment type of 'day' added to create daily QVDs.

Changes for V10

- The QVACB script library by [Matt Fryer](#) has merged with QVC. [Read more here](#).
- QVC Documentation and Examples are Personal Edition enabled.
- New Sub from QVACB, **Qvc.Icons**. Bundle loads image files into a QVW.

- New sub from QVACB, **Qvc.QvdEmpty**. Empties data from a QVD.
- New sub from QVACB, **Qvc.SegmentedStore**. STOREs table into segmented QVDs by date range.
- New function from QVACB, **Qvc.DateDiff**. Calculates the number of passed intervals between two dates or timestamps. Supports years, quarters, months, weeks, days, hours, minutes and seconds.
- New function from QVACB, **Qvc.Days360**. Equivalent to MS Excel function days360(). Returns the number of days between two dates based on a 360-day year (twelve 30-day months), which is used in some accounting calculations.
- New function from QVACB, **Qvc.NVL**. Equivalent to the SQL NVL function. Checks to see if the first parameter is Null and if it is it returns the second parameter.
- New function from QVACB, **Qvc.InRange**. Equivalent to MS Excel function inrange(). Returns whether the past value is between the start and end values passed for the range.
- Fixed [Issue 12](#). **Qvc.ExpandInterval** script error when field name contains comma.
- Fixed [Issue 5](#). **Qvc.DbExtract** Initial extraction proceeds even when KeyField has typo in it.
- Fixed [Issue 4](#). **Qvc. DbExtract** Provide table specific BaseValue.

Changes for V9.1

1. Fixed [Issue 8](#). **Qvc.LinkTable** may delete rows from source table.
2. Fixed [Issue 10](#). **Qvc.Loader.v.KeyFieldIsUnique** fails if keyfield exist in more than one table.
3. Fixed [Issue 11](#). Include of qvc.qvs causes script error in Qlik Sense. QVC should now be Sense compatible although this has not been extensively tested.

Changes for V9.0

1. The Qlikview Components project has moved from google code to GitHub:
<https://github.com/RobWunderlich/Qlikview-Components>

2. [Issue 6](#). Typo in documentation for **Qvc.IncrementalStore**.

3. New Sub **Qvc.ColorTheme**. QvcColorTheme is used to populate Qvc.Color.v.* variables with RGB values from a color theme file. The color theme file may be either *.xlsx or *.xlsm format and requires two columns, ColorVariable and ColorValue. A sample color theme file is provided in etc\Colors.

```
CALL Qvc.ColorTheme('..\etc\Colors\ColorSample1.xlsm')
```

Any variable matching the name Qvc.Color.v.* will be loaded and available in the qvw. These variables, for example, "*Qvc.Color.v.Chart.Data.01*" can be used as object properties to implement a consistent color scheme throughout a document. A qvt theme file is provided at etc\QvcColors.qvt. Object color properties in this qvt are populated with the Qvc.Color.v.* variable references. This qvt file can be used by "Layout, Apply Theme" to populate your objects with the Qvc.Color.v.* variable references.

4. New Sub **Qvc.AsOfTable**. Closes [issue 2](#).

Changes for V8.2

1. [Issue 44](#). **Qvc.DbExtract** -- Script error in Qvc.DbExtract if using StaleAfter test and regional settings the use something other than "." for decimal separator. For example, when using:

```
SET DecimalSep='.';
```

The expression used to test the QVD age generated a script error due to a "," being used in the decimal position.

2. [Issue 45](#). **Qvc.Calendar** --Negative start month param in Calendar generates incorrect _QuarterSerial values. For example:

```
CALL Qvc.CalendarFromField('OrderDate', 'Fiscal Calendar', '-', '-8');
```

For twelve generated quarters, _QuarterSerial values should range from 1 to 12. Instead, they were being incorrectly generated as 1 to 3. This caused incorrect results to be output from the SetPreviousYearQTD variable.

Changes for V8.1

1. [Issue 39](#). **Qvc.Calendar**-- vSetXXX variables incorrect when script includes QV variable setting Verbatim=1.
2. [Issue 38](#). **Qvc.DbExtract** --Add support for database deletes. The DbExtract parameter table contains a new column Deletes. If Deletes=Y, The QVD will be INNER JOINed with the keys specified by parameter KeyField from database table TableName.
3. **Qvc.DbExtract** -- Any complete LOAD/SQL statement may now be specified. This is useful when the extract consists of a SQL JOIN. The statement is specified in parameter table column LoadStatement. If specified, it will be used in place of any values specified in Columns. If Delta load is specified, include the clause:

```
WHERE $(Qvc.Loader.v.IncrementalExpression)
```

For example:

```
SQL SELECT
    AccountNumber,
    h.SalesOrderID as SalesOrderID,
    SalesOrderNumber,
    SalesOrderDetailID,
    UnitPrice,
    UnitPriceDiscount
FROM SalesOrderHeader h
LEFT JOIN SalesOrderDetail d ON h.SalesOrderID = d.SalesOrderID
WHERE $(Qvc.Loader.v.IncrementalExpression)
;
```

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 range 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