

QlikView

Best Practices

Design & Functionality

Version: 3
Date: 24/04/2007
Author: ABY/SNA

"A best practice is a technique or methodology that, through experience and research, has proven to reliably lead to a desired result."



Contents

CONTENTS	2
DESIGNING YOUR QLIKVIEW	3
SCRIPTING BEST PRACTICES	4
Script Structure	4
Naming Tables and Columns.....	5
Qualify or Unqualify?.....	5
QVD FILES & DATA SOURCES	6
What is a QVD file?	6
When to use a QVD file.....	6
ODBC/OLE DB Data Sources.....	6
Inline Tables	6
AESTHETICS	8
Using the right charts	8
USABILITY & DESIGN POINTERS.....	10
Sheets.....	10
Colours.....	10
Layout.....	10
A GOOD QLIKVIEW APPLICATION.....	10



Designing Your QlikView

The design and functionality of your QlikView application will determine the amount of benefit your customers/users receive. There are a series of standards and best practices you should look to adhere to when constructing your application and translating end user requirement into a real working, successful QlikView application.

QlikView functionality and design can incorporate a wide variety of development aspects.

- Scripting best practices
- Aesthetics
- Data Loading
- Preferences and settings
- Usability

This document will aim to provide you with a series of best practices methods and development tips to help you get the best out of your QlikView application.

Scripting Best Practices

“The QlikView script is a small program that is run by QlikView. It is a description of the location of source data, what fields to load and, if needed, how data is to be processed. When executed, it connects your QlikView document to one or several data sources and/or opens table files, reading the specified information into QlikView. Also what data fields to be included is defined here.” *QlikView Help*

Script Structure

Multiple Scripting Tabs

Using multiple tabs within a script will split out the various parts, enabling a simple view of the information for future development and support. Depending on the complexity of your application, you may have a variety of different script sections. The common parts are below.

- Security (usually hidden script)
- Dates and Calendar information
- Tab per data source
- Tab per key measure/core table
- Tab per lookup table

Security Tab (Hidden Script)

In QlikView it is possible to restrict the privileges of a document user from the **Document Properties: Security** and the **Sheet Properties: Security pages**. Any settings can be altered if the document user is logged in as ADMIN.

The user identity and password needed for opening a user restricted document are specified in the load script and will show up in the log file if you allow QlikView to generate one. However, by having the user access in the hidden script instead, the log file will not give away any login information. The Hidden Script button opening the hidden script is found in the Edit Script dialog.

Sample Script:

```
section access;  
load * inline [  
UserID,DEPT,Access  
John,FINANCE,Admin  
ANN,IS,Admin  
Peter,SALES,User  
Ben,TECH,User ] ;
```

UserID	DEPT	Access
John	FINANCE	Admin
Ann	IS	Admin
Peter	SALES	User
Ben	TECH	User

```
section application;  
star is *;  
Load * inline [  
    DEPT,CostCenter  
    FINANCE,*  
    IS,5020  
    SALES,5030  
    TECH,5040 ] ;
```

The field DEPT now exists with the same UPPERCASE name in both section access and section application. All field values are also UPPERCASE. The two fields would normally be totally different and separated, but if Initial data reduction based on section access has been checked, the following will result:

John may see data for all cost centres
Ann may only see data for cost centre 5020
Peter may only see data for cost centre 5030
Ben may only see data for cost centre 5040

Naming Tables and Columns

For professional users the QlikView developer should apply business names and meaningful fields for the ease of building new applications.

Naming conventions should be driven by the core business user.

Naming your Key Fields is also important. One tip is to give all of your keys the same prefix. For example %KEY_#####. Then when developing etc all of your key columns appear at the top of the list together.

Qualify (or Unqualify?)

This is a decision to be made at the point where the development commences.

Pros – Automatically removes synthetic keys.
 Development speed increased.

Cons - Applies non business/intuitive names to fields for professional users.

QVD Files & Data Sources

What is a QVD file?

A QVD (QlikView Data) file is a file containing a table of data exported from QlikView. QVD is a native QlikView format and can only be written to and read by QlikView. The file format is optimized for speed when reading data from a QlikView script but it is still very compact. Reading data from a QVD file is typically 10-100 times faster than reading from other data sources.

A QVD file is essentially a large table of data. If you want to store multiple source tables in one QVD you should join your source data to create one larger table to be stored.

When to use a QVD file

When to use QVD files should be driven by the overall size and performance of your environment and actual QlikView application.

Things to drive the decision of QVD usage:

- Large number of database rows and load times are significant.
- Require a frequent load.
- Desire to reduce impact on your operational system (e.g. DB)
- Re-usability of data sources. I.e. multiple QlikView applications using the same QVD as a data source.

ODBC/OLE DB Data Sources

If an OLE DB driver is available, this should be used ahead of an ODBC connection. The use of OLE DB negates the need for setting up an ODBC connection. OLE DB simply requires one setup step within QlikView.

ODBC is Open Data Base Connectivity, which is a connection method to data sources and other things. It requires that you set up a data source, or what's called a DSN using an SQL driver or other driver if connecting to other database types. Most database systems support ODBC.

OLEDB is the successor to ODBC, a set of software components that allow a "front end" such as GUI based on VB, C++, Access or whatever to connect with a back end such as SQL Server, Oracle, DB2, mySQL et al. In many cases the OLEDB components offer much better performance than the older ODBC.

Inline Tables

Inline tables can be used for a number of reasons. A key area is to create user defined dimensions to override sort orders in an application. For example the creation of Quarters, sorting dates.



Inline tables are primarily used to create tables that do not reside in any of your data sources. For example, creation of manual dimension tables applying your own descriptions to codes.

You should be aware that if used within an application, they can only be manipulated/changed by the Enterprise user. If you wish to create a user defined dimension/lookup there is also the option to use excel/csv/txt files as a data source. Stored on shared areas, these can be maintained by the employee responsible for the data.

Aesthetics

What are bad aesthetics?

- Cluttered
- Too many colours in one
- Dark colours
- Inconsistent object placement
- Wrong choice in object type
- Objects placed on top of each other
- Objects placed off the screen

What are good aesthetics?

- Uncluttered
- Crisp
- Aligned
- Light colour choices
- Consistency with object placement in a sheet

Using the right charts

When to use a bar chart

Comparisons for similar measurements

Actual vs. Budget → money to money, qty to qty

Amount over time → time to time

Top N → category values (e.g. sales reps)

Line Charts

Measurement to follow its movements

Comparisons among items

Performance Trends → Amount movements over time

Compare 2 dimensions → Annual Comparisons over months

Combo Chart

Comparisons for multi-measurements

Two measures → Currency amount and quantity

→ Amount and percentage

Two axis → Left-axis and right axis for measurements

Two or more axis → Split axis horizontally



Radar Chart

Comparisons of high level groups
One or Two measurements → Sales and Profit
(Avoid a dimension with many values)

Gauge Charts

Ratio of measurements
Percentages → Margin, Quota Achieved, YTD vs. Last YTD
(Avoid actual amounts)

Scatter Chart

Measurements act as dimensions (3D chart)
3D measurement chart → Amounts, quantity and # of customers
Map chart → Sales on geographical map

Grid Chart

2 or 3 dimensions with a measurement
Multi-cross dim measures → Amount over region and product
over time

Pie Chart

Visual distribution measurements
What contributes how to the whole → Regional Sales
Yes / No analysis

Block Chart

One dimension with multiple measurements
KPI Overview → Cycle group dimension with Sales, Qty, Margin
Sort in any columns (No grouping)

Pivot Table

Multi-dimensions with one or more measurements
Grouping of dimensions → Product sales by item
Dimension across table → Annual sales by months
→ Variance calculation using before()
No repetition of the same values (grouped)
No end-user sort by double-click

Usability & Design Pointers

Sheets

- Max Sheets – Keep the number of sheets/tabs to a minimum (e.g.10)
- Always include a Help / Get Started tab and/or a link to a help site on our website.
- Name each sheet and object with descriptive headers

Colours

- Black & White charts viewing & Printing (Colour Blindness)
- Red & Green - Many people are red/green colour-blind - consider this e.g. when using visual cues.
- Red and green are also associated with good and bad indicators / performance. Only use red and green when you mean to indicate good and bad.
- Colouring is very individual so we recommend neutral colours (e.g. white background) but if you can match the design of an application to the look and feel of a customer that is sometimes good (always include the logo somewhere, perhaps on an intro sheet instead of having it on all sheets – steals too much real estate).

Layout

- Design for a fixed resolution that applies to your organisations desktops (e.g. 1024 x 768).
- Always consider sort order and whether to present frequency (# or %) in list boxes (sometimes very useful but definitely not always)
- Repeated objects (clear buttons) at the same position in every sheet
- Multi boxes can be good for people that are used to working with QV but they are not very intuitive. List boxes take more space but are better (you can e.g. see the gray areas better).
- Clean layout in charts
- Hierarchy dimensions placed in order
- Time and Dates are crucial elements of most apps and they must be highly intuitive to search and use (do we always need to display quarter / tertial when users can select the interesting months in the list box? When and in what graphs does it make sense to display Period / YearMonth instead of Month?)
- Table columns should always be searchable (display totals in tables whenever it makes sense)

A Good QlikView Application

Below are some screenshots of a good QlikView application....





