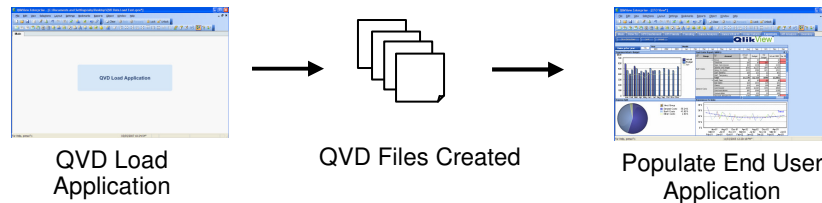


QlikView QVD's

A QVD (QlikView Data) file is a file containing a table of data that QlikView has extracted from one or more data sources. QVD is a native QlikView format and can only be written to and read by QlikView.

Creating and loading from QVD Files

QVD files should be created via one QlikView application then the final end user applications should be sourced from these files. The simple diagram below shows this 2 step process.



To create a QVD, the Store function should be used. A simple script example is below. It is also good practice to delete all of the tables created within your load application (using the Drop function) to free memory.

```
Data:
Load ID,
      Description
FROM C:\Data.xls (biff, embedded labels, table is [Core Data$]);

Store Data into C:\Data.qvd;
Drop Table Data;
```

In this example, you now have a QVD file on your C drive that can be used as a source for your end user analysis.

Why Use QVD's?

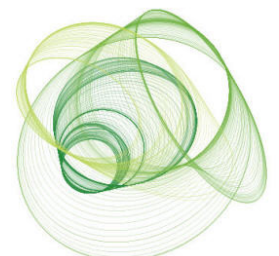
- Faster data loads
- Incremental data loads
- Large data loads
- Joining QlikView Applications/Data

Faster Data Loads

Loading data in to a QlikView application from a QVD file will be 10-100 times faster than pulling your data direct from the source database. QVD files will be read in two different ways, Fast and Super-Fast.

Fast QVD's – Loading data from a QVD file, when you have then added some calculations/IF statements.

Super-Fast QVD's – If you can build the QVD to hold all of the data you require (therefore when the data is added to your application all you have to do is rename columns and add no calculations, extra columns or joins), the QVD will be read at the optimum speed.



Incremental Data Loads

In organisations where data volumes are large or batch windows are small, there will be a requirement to implement incremental loads therefore only load data that has changed or added since your last load. These types of data loads can only be implemented if QVD files are used.

Various papers are available that detail the Incremental load process further.

Large Data Loads

QVD files can also be used to massively improve the loading of historical data in to your QlikView application. QVD files can load millions of rows in seconds.

Large Data Example

In an organisation that has masses of transactional data going back 5 years, you don't want to have to load all 5 years including the current years (growing) data every night. One solution to this is below:

Create QVD Files for each year you wish to analyse. These can be created dynamically based on how many years you wish to look at. The script below show how to create a simple QVD file based on a moving date clause.

```
'Year-5':  
Load      Id,  
          StoreName,  
          StoreLocation,  
          Date,  
          Product,  
          Sales  
FROM C:\Data.xls (biff, embedded labels, table is [Core Data$])  
Where Year(Date) = YEAR(TODAY())-5;  
  
Store 'Year-5' into C:\Year-5.qvd;  
Drop Table 'Year-5';
```

Creating multiple instances of this script above and simply changing the Where clause will enable you to create a set of yearly QVD files based on the number of years of analysis you require, along with a current year QVD file (`Where Year(Date) = YEAR(TODAY())`).

Once all of the necessary QVD files are created, you can then source your end user application on these files.

Joining QlikView Applications/Data

Over time, as you construct more and more QlikView applications you will begin to create multiple QVD files across your business on various data sources. The nature of loading QVD 's means that as you create these files they can be used by many of your QlikView applications over and over again. This will greatly reduce your development time in all of your future applications.

As business questions become more and more complex, you can begin to merge many of your QlikView applications in to increasingly powerful analysis suites by simply porting QVD data sources across applications with ease.

