



# Tips and Tricks with QVD Generators

Presented by: Result Data

Christina Adams

[christina.adams@resultdata.com](mailto:christina.adams@resultdata.com)

# Agenda

## QVD Generators Outline

- ▶ **Basics**
  - ▶ QVDs
  - ▶ Tiers Logic
- ▶ **Tips & Tricks**
  - ▶ Config Files/Path Variables
  - ▶ For Loop
  - ▶ Sub Routines

# QVD Basics

What is a QVD file?

- ▶ A QVD file is a native QlikView file format optimized and compacted for speed when reading data from within a QlikView load script.
- ▶ Reading data from a QVD file can be 10-100 times faster than reading records directly from other data sources.
- ▶ A QVD file contains a single table of data from a QlikView application. While that may seem somewhat restricting remember that table can be the result of a concatenation or a join so the structure you create in the application can greatly increase its use.
- ▶ You can also include all calculations and manipulations in the script that creates you QVD file further increases load performance at run time.

# QVD Basics

How could you use a QVD file?

There are several uses for a QVD file and in many cases more than one will be applicable at the same time.

- ▶ Decreasing Load Time
- ▶ Decreasing Database Server Loads
- ▶ Joining Data from Different QlikView Applications
- ▶ Incremental Data Loading

# QVD Tiers

## QVD Creation Methods

### Two-stage architecture

- ▶ Source Layer: composed of the source databases and original tables.
- ▶ Extract Layer: composed of QlikView documents, containing mainly script. These are used to pull the data from the source layer and store it into QVD files. The extraction scripts can either create a straight copy of the source tables to store them into the corresponding QVD files, or perform certain transformations before storing the result.
- ▶ QVD Layer: the set of QVDs resulting from the Extract Layer. These QVDs become the data sources used by the final QlikView document.
- ▶ Presentation Layer: the set of QlikView documents used to provide the data to the end user. These QlikView files will use the QVDs created in the previous layer as data sources, and sometimes perform additional transformations to create the final data model. No database calls are performed from the presentation layer.

# QVD Tiers

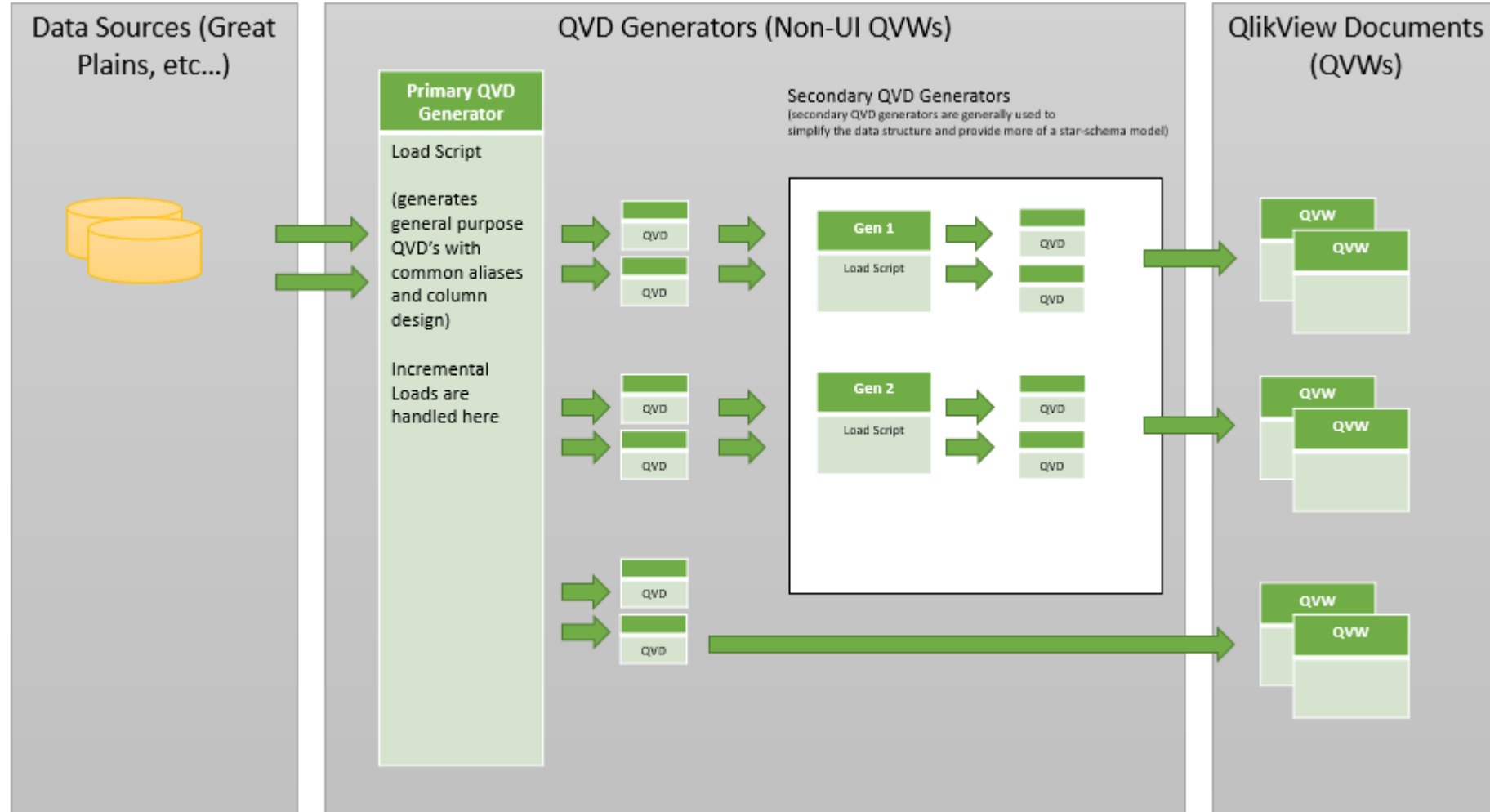
## QVD Creation Methods

### Three-stage architecture

- ▶ In this architecture, two additional layers are added: Transformation Layer and Transformed QVD Layer. The role of these two additional layers is to hold all transformations that need to be performed upon the source data before it can be integrated into the target data model.

# QVD Tiers

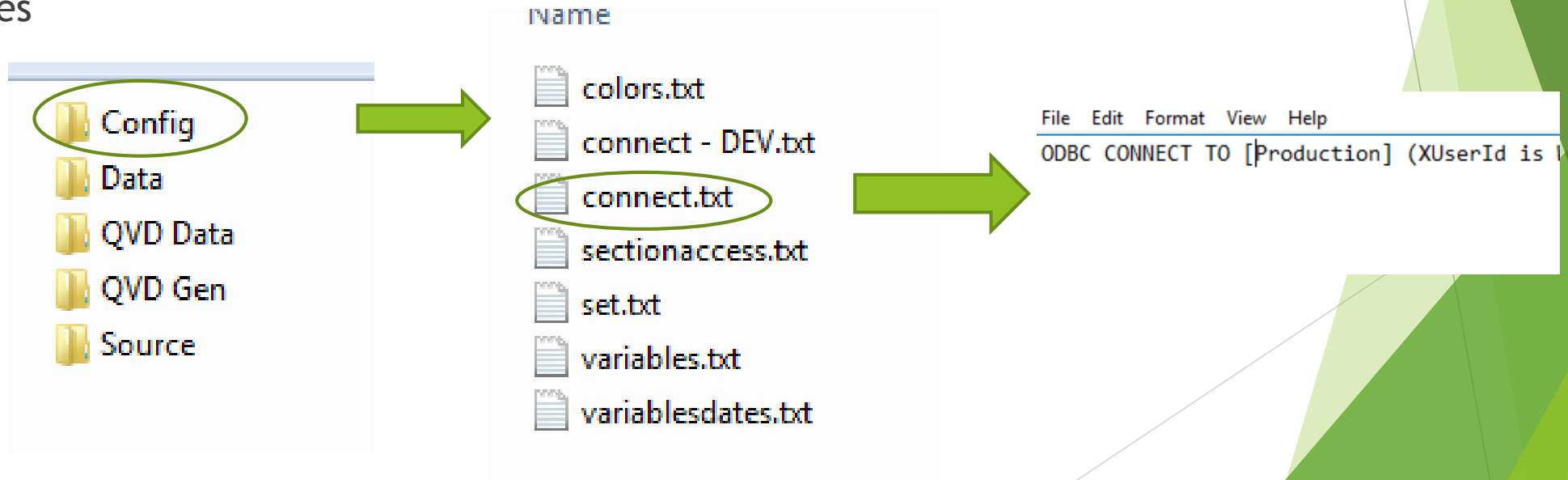
## QVD Creation Methods



# Tips for QVD Gen

## Config Files

- ▶ Text files contain snips of code that can be use multiple times across all QVW files
- ▶ Updates or changes only have to be applied once to the file
- ▶ Upon a reload, all QVWs that are pointing to these files will inherit any changes





# Tips for QVD Gen

## Path Variables

- ▶ Locations can be changed once at the beginning of the script

```
set vPath = '..\';  
set vQVDs = '$(vPath)QVDs';  
set vData = '$(vPath)Data';
```

//Sub Folders in QVDs

```
set vSalesQVDs = '$(vQVDs)Sales';  
set vBudgetQVDs = '$(vQVDs)Budget';
```

# Tricks for QVD Gen

For Loop

Use case: Tier 1 Generator, Always a full reload

Tables:

```
LOAD * INLINE [  
  TableNames  
  SUPPLIER_VW  
  PRODUCT_VW  
  DISTRIBUTION_CENTER_VW  
];
```

```
Let vTableCount = NoOfRows('Tables');
```

```
For i = 0 To $(vTableCount) -1  
  LET vMyTableName = Peek('TableNames', $(i), 'Tables');  
  
  $(vMyTableName):  
  SQL SELECT * FROM $(vMyTableName);  
  
  STORE $(vMyTableName)  
    INTO$(vQVD)$(vMyTableName).QVD (QVD);  
  
  DROP Table $(vMyTableName);  
  
Next i
```

# Tricks for QVD Gen

## Sub Routines

Use Case: Longer/more complex load script. Often need to only run parts manually. T1 or T2

### SUB ProductLoad

Product:

LOAD

ListPrice,

Name as ProductName;

SQL SELECT \*

FROM Product;

STORE Product INTO Product.QVD (QVD);

DROP Table Product;

END SUB;

```
//Call SalesLoad;  
Call ProductLoad;  
//Call CustomerLoad;
```



# Questions? Comments.

Presented by: Result Data

Christina Adams

[christina.adams@resultdata.com](mailto:christina.adams@resultdata.com)

# Resources

Please visit the following articles for further reading

- ▶ <https://www.resultdata.com/qlikview-incremental-loads-and-qvds/>
- ▶ <https://www.resultdata.com/three-types-of-qlikview-incremental-loads/>
- ▶ <https://www.resultdata.com/qlikview-folder-structure/>
- ▶ <https://www.resultdata.com/qlikview-include-files/>
- ▶ <https://www.resultdata.com/qvw-sub-routine/>
- ▶ <https://www.resultdata.com/qlikview-qvd-loop/>

Sources in the Presentation include the articles above and also

- ▶ Discovering QlikView
- ▶ QlikView 11 for Developers
- ▶ QlikView community & white papers

