How to Build Your First QlikView Document

An introduction to the basic principles of building a QlikView document

Last updated: March 2011

Acknowledgements

This 'getting started' guide is the brainchild of Mark Rodell, a Pre-Sales technical team leader in QlikTech's UK branch. Mark wrote this guide after working directly with many 'newbies' to the QlikView product to help them answer their initial questions about how QlikView works and how to build their first application.

QlikTech is delighted to provide this guide as a free resource to help people get up-and-running with QlikView and to give them a taste of the amazing power of the QlikView Business Discovery platform.

Once you are done with this guide and want to learn more about how to master the art of Business Discovery, we suggest you take a look at the QlikView Educational Services catalog of training:

QlikView Education Services has expertly designed coursework and materials to give your organization the knowledge and skills needed to develop, deploy and adopt powerful QlikView applications. With training developed by QlikView Education Services, organizations are able to rapidly develop meaningful applications and put the power of business discovery in the hands of end-users - allowing them to make smarter, more informed decisions.

With training from QlikView Education Services you will:

- Build your knowledge of key capabilities and functionality within the QlikView platform
- Learn to build powerful, meaningful applications that will enable end users unlike ever before
- Accelerate time-to-value and increase return on investment with QlikView

To learn more about QlikView Education Services, register for one of our upcoming classes or view one of our free e-learning modules visit:

http://www.qlikview.com/us/services/training

How To Build Your First QlikView Application

Table of Contents

Introduction

This document has been designed to provide step by step guidance to building your first QlikView document. The data used in the build process documented has been included in the material provided. In-order to complete the documented build process, you will need the following files:

- Northwind.mdb – A Microsoft Access database that is freely and publically available. It contains data based on a fictitious product sales organisation.

- CustomerMaster.csv – A comma-separated values file containing a list of customers and their details.

- ProductMaster.xls – A Microsoft Excel spreadsheet containing a list of products sold by the fictitious sales organisation.

- ProductCategory.txt – A text file containing the product to product group relationships.

By combining the data from these sources you will be able to build a QlikView document detailing Sales and Margin Analysis across time for the organisation concerned. You will have the ability to review by product and/or customer and drill down to the transaction level detail. The document will also contain a speedometer style gauge to graphically represent the company KPI of average discount percentage rate.

Aim of the document

The aim of the document is to provide a user with the basic skills required to build a simple QlikView document. It is not intended to act as a replacement for product training. Any QlikView developer's knowledge and skills can be further enhanced by training and through use of the QlikCommunity.

QlikView Product Training can be obtained directly from your local QlikView office, or the QlikView website:

http://www.qlikview.com/uk/services/training

QlikCommunity can be located at:

http://community.qlikview.com/

There is also a very useful document readily available with QlikView developer software. This document is the QlikView Reference Manual, and is typically found in the following folder:

<Drive>:\ Program Files \QlikView\Documentation

This document is named "QlikView Reference Manual.pdf".

## Components of a QlikView document

There are a number of components within a QlikView document which can be developed in isolation, but are more commonly developed together. These QlikView components are:

- Load Script – The method through which data is loaded into the "in-memory" associative database. It is created either by entering a script manually into the script editor or by using a series of wizards designed to generate script for less experienced users. It allows data from multiple data sources to be compiled into a single QlikView document.

- In-Memory Associative Database – A visual, table based representation of the data that has been loaded into the QlikView document. Whilst the individual tables loaded by the script are shown in the "Table Viewer", in fact, QlikView writes all of the data into a single table in memory allowing the associative search capabilities to function.

- End User Objects – The objects such as charts, tables and gauges that allow end users to interact with the document. It is important to remember that all objects are interactive and by clicking on them you are effectively making selections in fields.

- Green, White & Grey – These 3 colors are very important within QlikView. They represent the data associations and form an integral part to deriving value from all QlikView documents.

    o Green – Any field that is highlighted green in any selection list indicates that it has been physically selected either by clicking on it directly, or by making a selection via a chart or table.

    o White – All items in selection lists that are displayed white are directly related to selected (green) items, meaning that there is a row of data that contains that combination of data values. In Charts and Tables only related data is shown.

    o Grey – All items in selection lists that are displayed grey are not directly related to selected (green) items, meaning that there are no rows of data that contain that combination of values. Charts and Tables typically do not contain greyed out data as they are designed to only show related data.

Loading Data

Launch QlikView from the Start menu (Start -> All Programs -> QlikView)

1. When creating a new QlikView document you are prompted to use a Getting Started Wizard. The screen shot below demonstrates the wizard. This wizard is used to create all 3 components, the script, the in-memory associative database and the user interface without any other user action. (If you don't see the wizard, in QlikView select Help -> Show Start Page and on the screen that appears, select 'Create a new QlikView document by loading data from an Excel file")
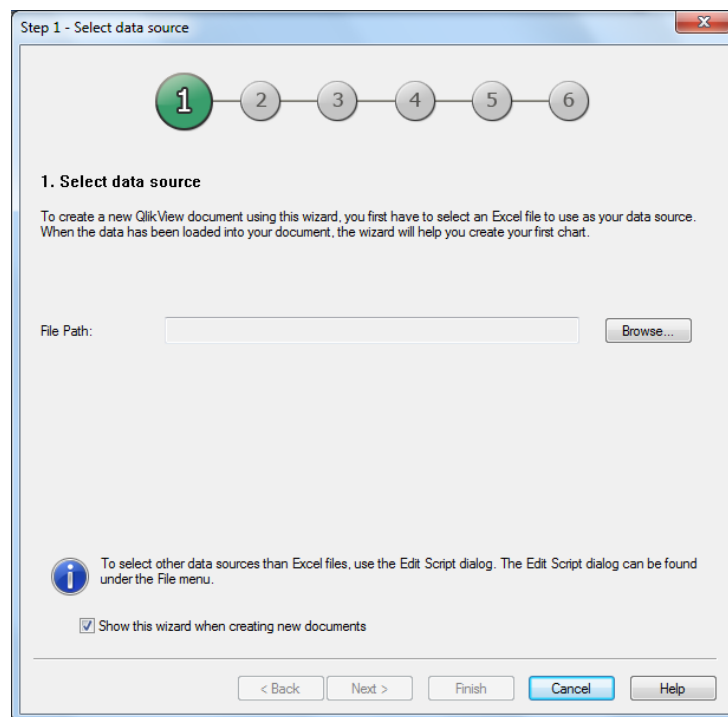
2. For the purpose of this guide we will <u>not</u> be using the getting started wizard, as it only allows us to load data from MS Excel spreadsheets. Through the course of this guide we will be drawing data from multiple data sources, so it would not allow us to complete our task fully. Therefore, click the "Cancel" button.

Now, our new starting point – a blank screen as shown, can feel a little intimidating at first.
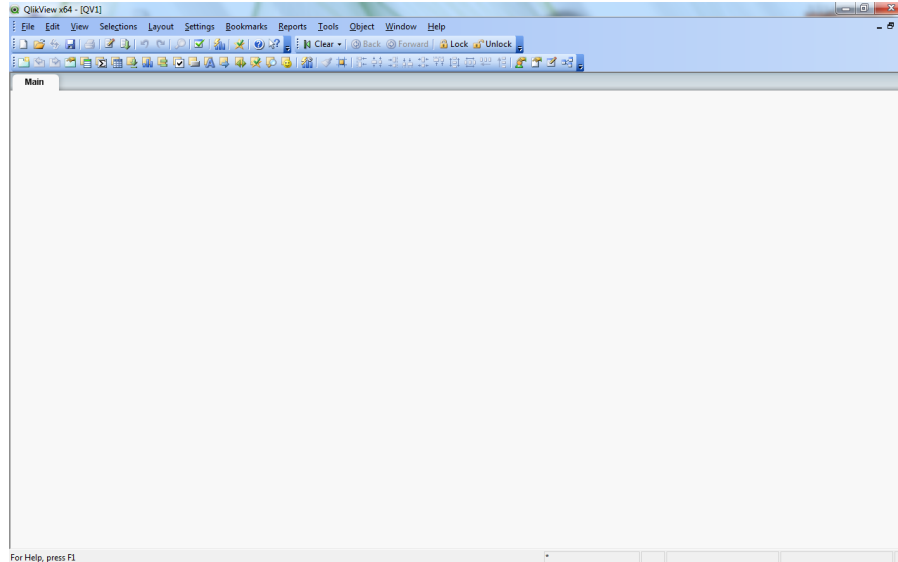
However this part of the document that you are looking at, the end user interface, will remain blank until objects are built in a later section of this document. The first step is to load the data from its sources into the in-memory associative database. Once there, the data can be used by the objects that act as the end user interface. To load the data we have to create a load script, this is done in a separate area of the product, called the Script Editor.

3. To open the script editor, click the Edit Script Button (located on the Standard toolbar), pictured below; alternatively, you can use the shortcut by pressing CTRL & E.

4. Once you are in the script editor the screen will look similar to the screen image below.
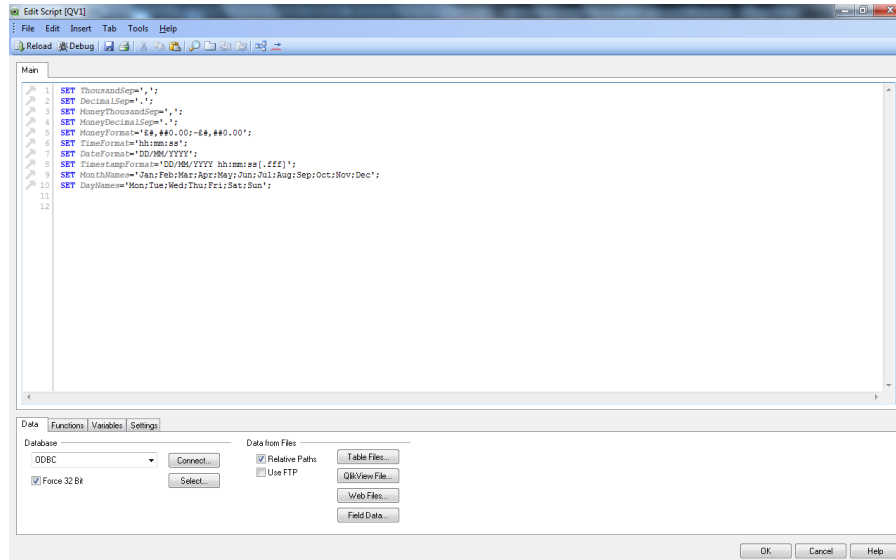
You can see that some system variables are already established for you. You typically do not need to change these as they are generated via interaction with your computer's regional settings during the software installation process. The whole process of writing a script can be done manually if you are competent and confident in doing so, but normally it is faster and easier to use one of the wizards that are available. We will use the wizards.

Our first data source is an MS Access database, from which we will need 3 tables.

When connecting to a database, you will need to establish the connection before reading the tables.

5. To do this you must first ensure that you have selected the correct type of connection from the database drop down list, located on the data tab in bottom left of the screen;

The possible connection types are:

- ODBC – This will allow you to use any defined ODBC connectors within your local environment. You will need to discuss this type of connection with your IT services provider if you wish to connect to a database that is not on your own local machine.

- OLE DB – This is another type of connector used for external databases, again you should discuss the use of this with your IT services provider.

- QVSAdminDataProvider.DLL – This custom connector gives you the possibility to connect to your QlikView Server and load information from the DMS and Server objects. We will not use this connector in this instance.

We will now connect to the MS Access Database, Northwind, (which has been supplied for you) and load the first three tables from there.

6. Firstly, you should ensure that the database drop down list has ODBC selected and then click the "Connect" Button.

Connect...

7. Next, select the Data Source" MS Access Database" and click on "OK" to let QlikView know that you will be using Microsoft Access as a data source. If you do not have "MS Access Database" defined in your list, then you should contact your local IT services provider, who can give you the necessary access and functionality.
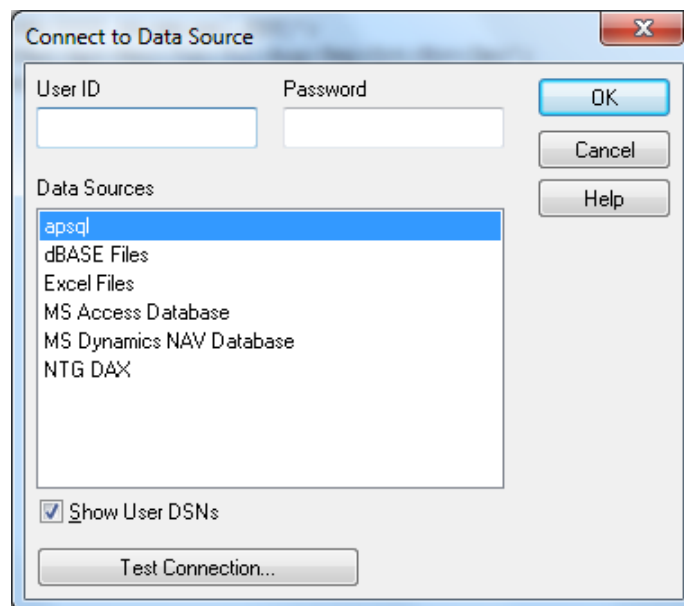
8. Then you should locate the file Northwind.mdb, from the data that has been issued along with this document from within your files system.

9. Select the relevant Drive and Folder Path to find the file, similar to that shown in the diagram below;
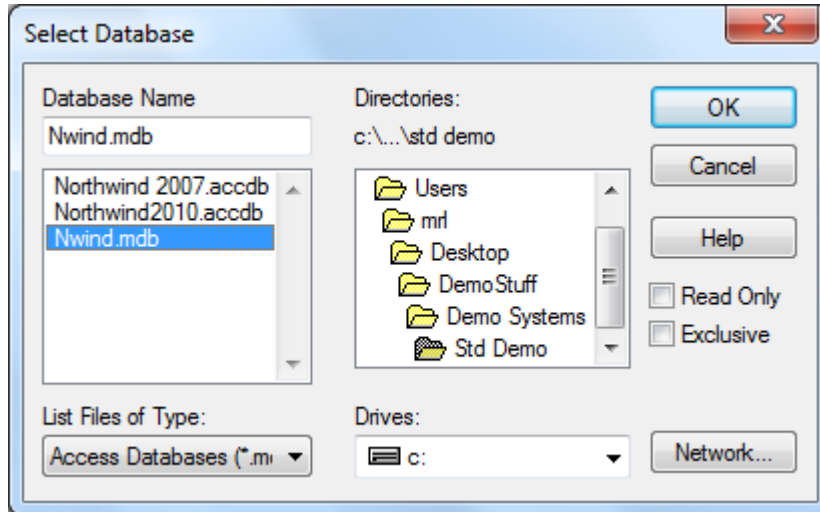
10. Once you have located and highlighted the file, click the "OK" button.

The connection wizard will create the necessary script for you and place it into the script block in the script editor window, as shown on the next page ; (**Please Note:** Yours will be slightly different as it will contain your folder path and not the one shown!)
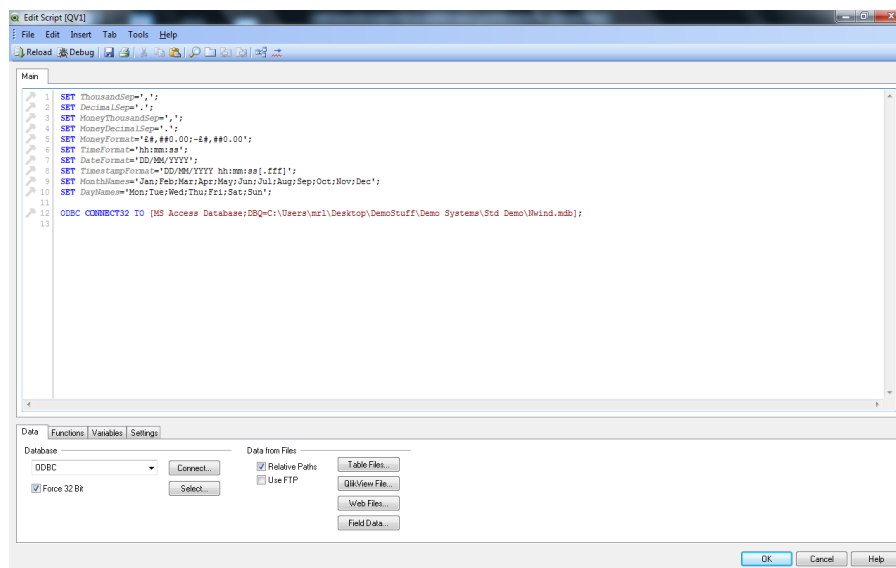
11. The next action you should take is to use the "Select" button . This will open a window as shown below to allow you to select your tables from the nominated database.
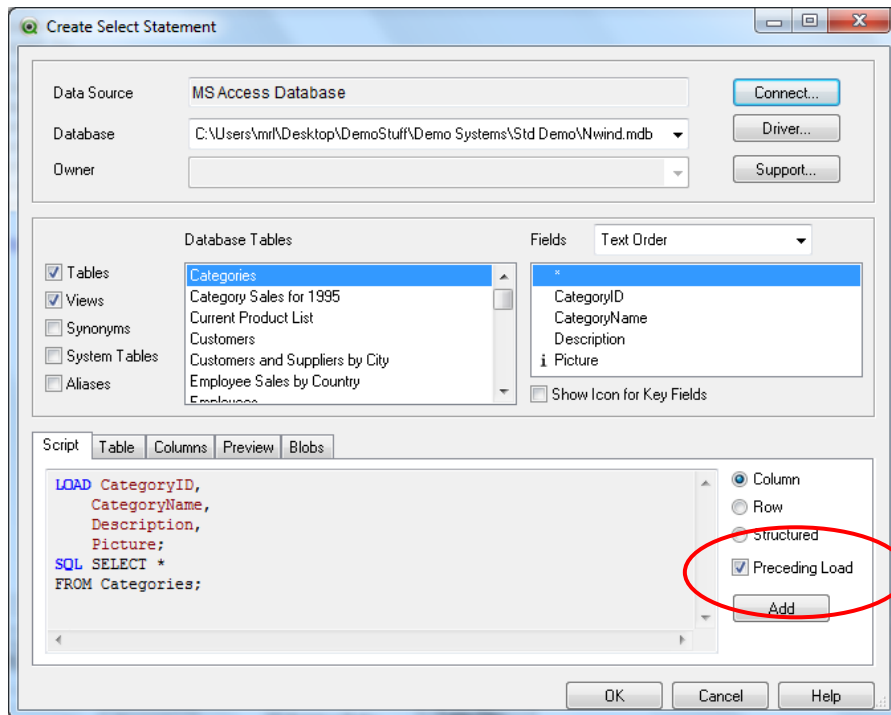
IMPORTANT: Ensure that you have the option for Preceding load checked!

If the "Preceding Load" check box is not checked then the script engine will only write an abbreviated script, all of the fields within the table will not be listed individually and therefore some of the changes that you are required to make later will not be possible!

From the database you will need to load the following tables:

Table Name	Contents
Orders	The summarised information regarding the orders that have been taken and fulfilled by the company.
Orders Details	The order lines and details for all of the orders that have been taken and fulfilled by the company.
Employees	The list of all employees within the company.

Initially, it is best if you load in one table at a time so that you can edit the script. It will also allow you to name the table in memory so that the users will be able to understand the data more easily. Once you are more familiar with this process you can add more than a single table at a time by using the "ADD" button.

12. Click on the table that you wish to select, we will start with the Orders Table.
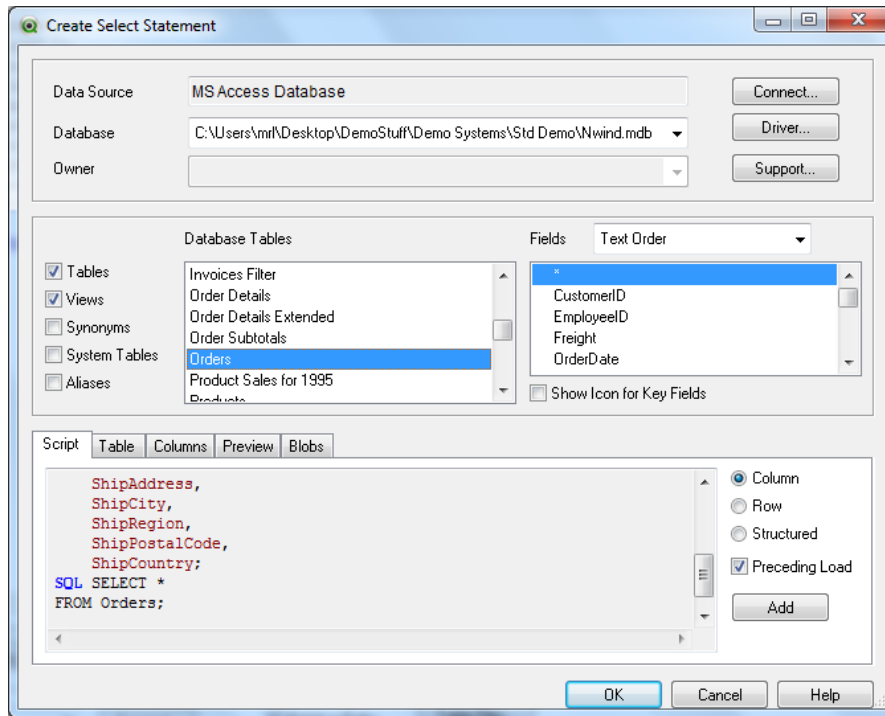
13. Click "OK".

The script will be written into the script window for you. As pictured in the screen image on the next page:
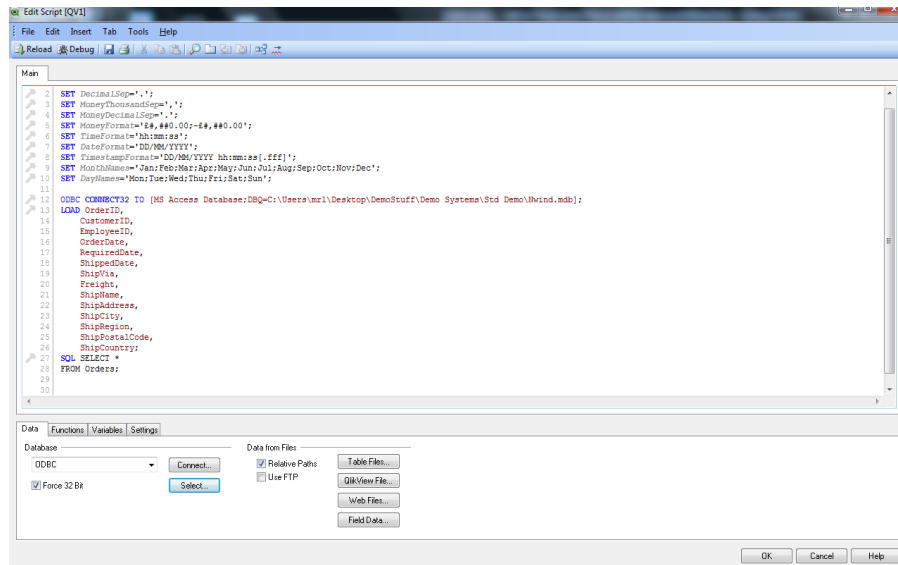
14. Once the script is in place you should name the table (for use by QlikView's in-memory associative database) by typing in the table name followed by a colon (:) on the line above the script. As pictured in the screen image below.
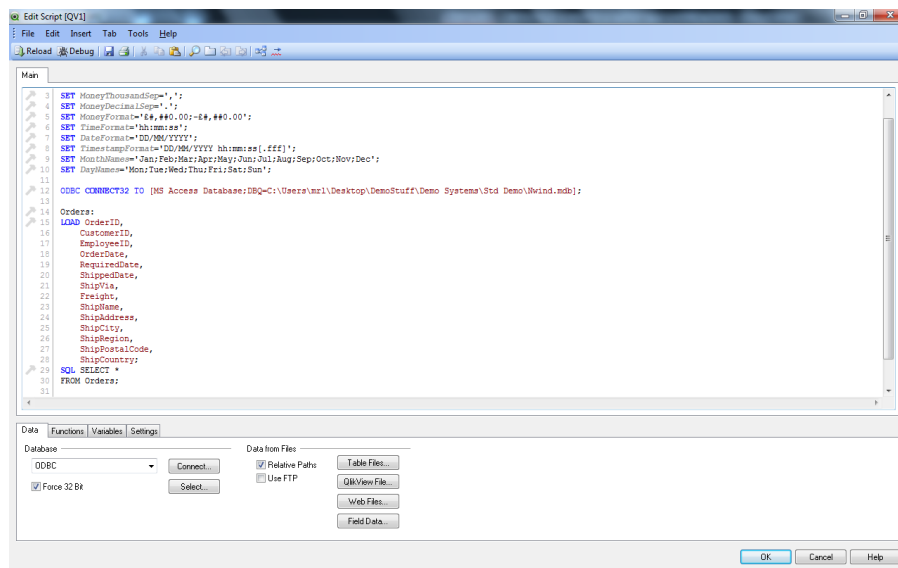
15. You should repeat the same steps for the other two tables that we wish to utilize, Order Details and Employees.

Your finished script should read as follows:

```
Orders:
LOAD OrderID,
CustomerID,
EmployeeID,
OrderDate,
RequiredDate,
ShippedDate,
ShipVia,
Freight,
ShipName,
ShipAddress,
ShipCity,
ShipRegion,
ShipPostalCode,
ShipCountry;
SQL SELECT *
FROM Orders;

OrderDetails:
LOAD OrderID,
ProductID,
UnitPrice,
Quantity,
Discount;
SQL SELECT *
FROM `Order Details`;

Employees:
LOAD EmployeeID,
LastName,
FirstName,
Title,
TitleOfCourtesy,
BirthDate,
HireDate,
Address,
City,
Region,
PostalCode,
Country,
HomePhone,
Extension,
Photo,
Notes,
ReportsTo;
SQL SELECT *
FROM Employees;
```

Once you have this you are ready to save and load the data for the first set of tables, from the MS Access database.

16. On the toolbar you will see a "Save" button, as pictured below. Click the Save button. As this is the first save you will need to specify the name of the file and the path that you would like the document to be saved to.

(Save Button)

17. Once you have saved the document then you should load the data. This is done by using the "Reload" button from the toolbar at the top right of the screen, as pictured below.

(Reload Button)

The script will then run and load the data into the in-memory associative database. This should only take a few seconds and should not return any errors at this point. (**Please Note:** If you do receive any errors then please compare your script against the one shown above and ensure that they are the same.)

Once the load has completed you will be given the opportunity to select fields to be placed on to the documents interactive usage area. You should **NOT** do so at the moment.

18. You should now return to the script and create a blank line after the existing script by pressing the "Enter" button in the script window. Now you should add the three remaining required tables from other data sources;

Table Name	Contents
CustomerMaster	The master list of all customers, including addresses etc.
ProductMaster	The master list of all products , including descriptions etc.
ProductCategories	The list of product codes that aggregate into categories and the category descriptions.

These tables are not coming from a database; they are coming from files on the file system. To be able to load these tables, click the "Table Files" button situated at the bottom towards the left of the
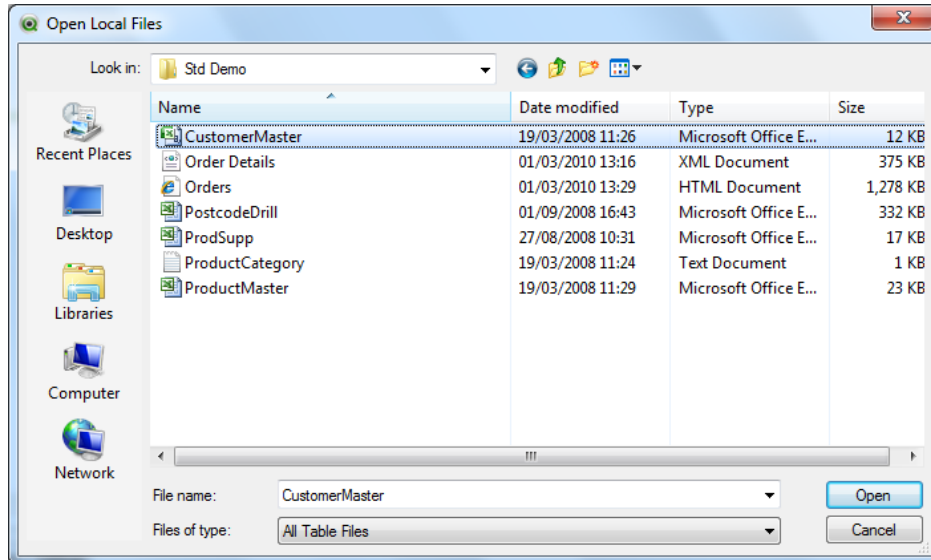
screen, as pictured below.
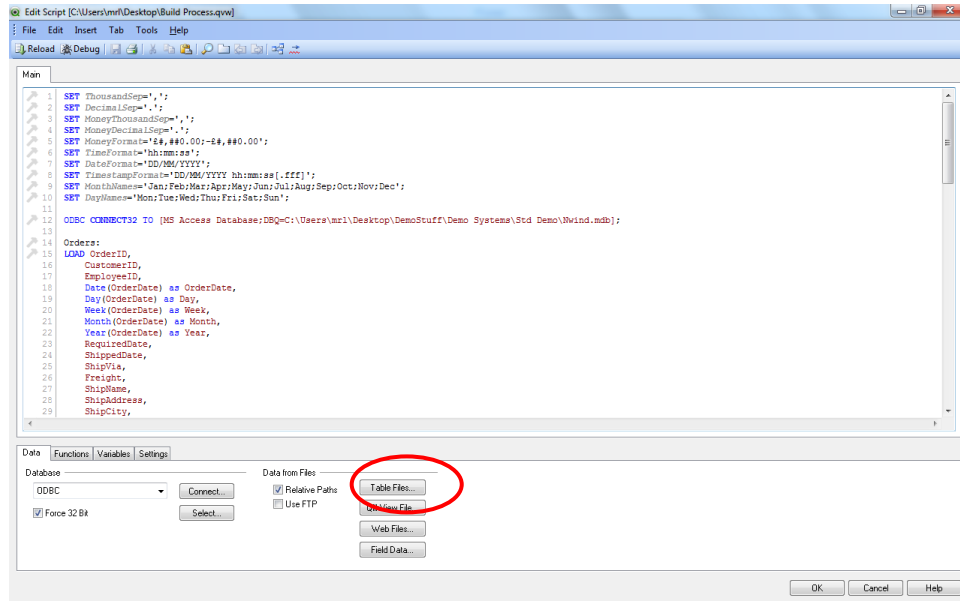
The "Table Files" button is used to initiate a wizard that writes the script for loading MS Excel files, CSV files and QlikView data (QVD) files from the files system on either your local machine or a network share.

You will be prompted with a window that allows you to select files from the file system, an example of which is shown below:

20. Go to the correct location on your file system, where you have placed the files that were distributed with this document and select the first of the files. We shall start with the file – CustomerMaster.CSV.

21. Once you have selected the file you will be prompted with a wizard window which guides you through interpreting the source file. As pictured below;
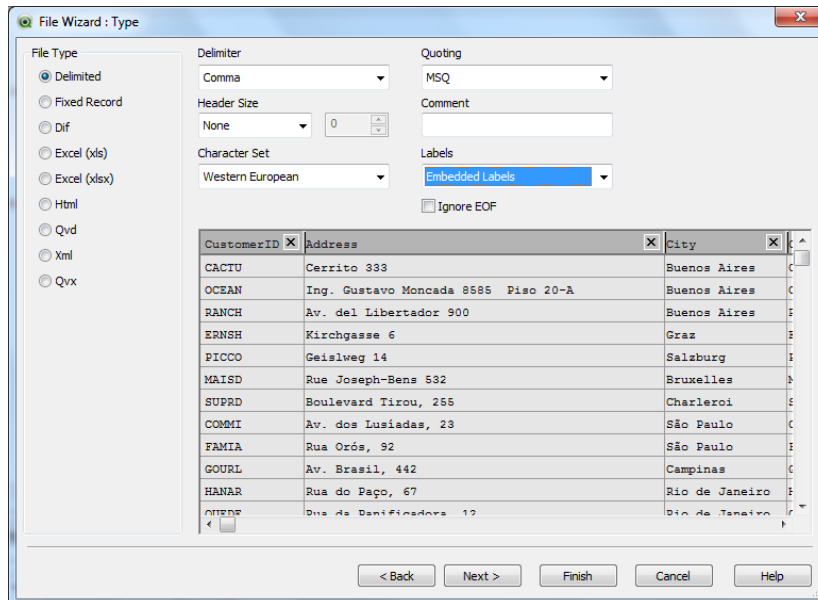
Table Name	File Type	Delimiter	Header Size	Labels
CustomerMaster	Delimited	Comma	Lines 0	Embedded
ProductMaster	Excel (xls)	N/A	None	Embedded
ProductCategory	Delimited	Tab	Lines 0	Embedded

22. Click the "Finish" button.

23. Select each of the three files in turn and don't forget to name the tables as we did with the others, by using the table name (**without any spaces**) followed by a colon (:).

When you have completed all tables your script should look like the one detailed below:

```
ODBC CONNECT TO [MS Access Database;DBQ=C:\USERS\MRL\DESKTOP\DEMOSTUFF\DEMO
SYSTEMS\STD DEMO\Nwind.mdb];

Orders:
LOAD OrderID,
CustomerID,
EmployeeID,
OrderDate,
RequiredDate,
ShippedDate,
ShipVia,
Freight,
ShipName,
ShipAddress,
ShipCity,
ShipRegion,
ShipPostalCode,
ShipCountry;
SQL SELECT *
FROM Orders;

OrderDetails:
LOAD OrderID,
ProductID,
UnitPrice,
Quantity,
Discount;
SQL SELECT *
FROM `Order Details`;

Employees:
LOAD EmployeeID,
```

```
LastName,
FirstName,
Title,
TitleOfCourtesy,
BirthDate,
HireDate,
Address,
City,
Region,
PostalCode,
Country,
HomePhone,
Extension,
Photo,
Notes,
ReportsTo;
SQL SELECT *
FROM Employees;

Customers:
LOAD CustomerID,
Address,
City,
CompanyName,
ContactName,
ContactTitle,
Country,
Fax,
Phone,
PostalCode,
Region
FROM
[Demo Systems\Std Demo\CustomerMaster.csv]
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

Product:
LOAD CategoryID,
ProductID,
ProductName,
QuantityPerUnit,
SupplierID,
UnitsInStock,
UnitsOnOrder
FROM
[Demo Systems\Std Demo\ProductMaster.xls]
(biff, embedded labels, table is ProductMaster$);

Category:
LOAD CategoryID,
[Product Category]
FROM
[Demo Systems\Std Demo\ProductCategory.txt]
(txt, codepage is 1252, embedded labels, delimiter is '\t', msq);
```
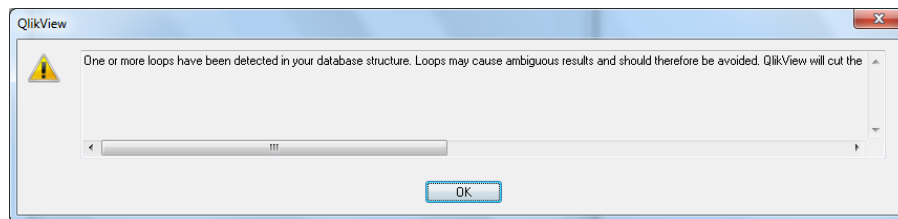
24. Once the script looks correct you should "Save" the document again. To do so, click the "Save" button on the toolbar.

25. Then, once again "Reload" the data, by using the "Reload" button on the tool bar.

 The following message will appear as the reload process finishes!

This message tells us that there has been a loop created via the joins in the data. A loop in the in-memory associative database is created where there is a more than a single route through the table structure from one end to another.

All loops must be resolved in any QlikView document. If a loop is not resolved or removed then incorrect data associations may appear in your data and the results shown in the application may not be correct.

26. Click the "OK" button to continue.

To understand where the loop is you will need to use the "Table Viewer", which visualises the table structure that has been created in memory by the data load.

27. To do this, press the keys "Control" & "T" keys down together. The following screen should appear.
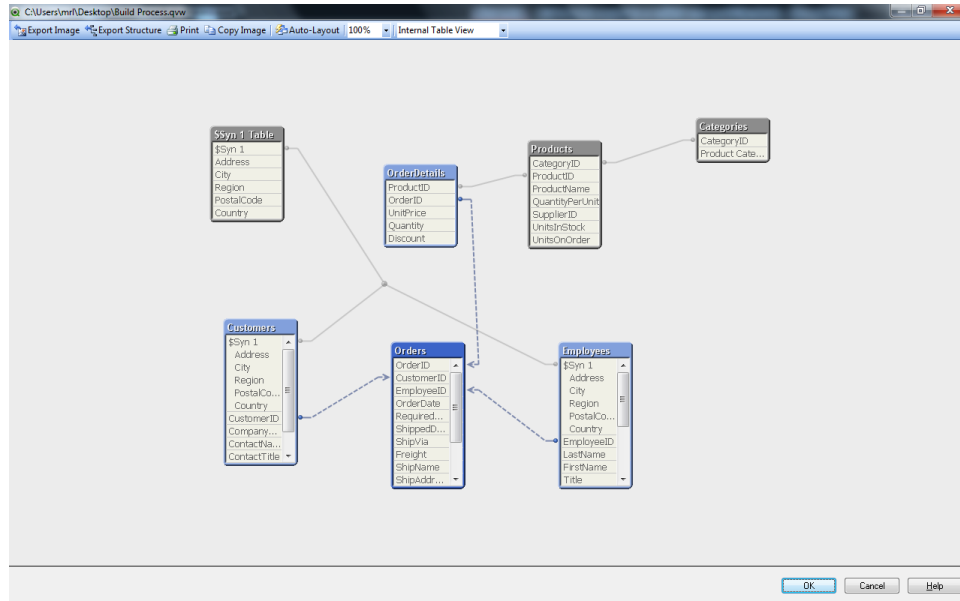
From this screen we can see that the loop is highlighted by a dotted line and is caused by having address fields in both the customer table and the employee table. To resolve this we need to remove the address fields from the Employees tables, as we do not need them for this document.

NOTE: The load process in this case, as is fairly typical when loading data from a database, creates a synthetic key. (I.e. a key that combines more than one field into a single key.) It is best practice to try to eliminate any synthetic keys from the QlikView in-memory database. Typically, this will mean creating a true composite key for the in-memory database from the source fields and then not loading the source fields individually.

28. To remove a field from a load process it can be commented out by using a "//" ahead of the fields within the script. (If you have done this correctly the text will change color.)

Use the shortcut keys "Control" & "E" to go to the script editor, then edit the script for the Employees tables so that it looks like the example below:

```
Employees:
LOAD EmployeeID,
LastName,
FirstName,
Title,
TitleOfCourtesy,
BirthDate,
HireDate,
//     Address,
//     City,
//     Region,
//     PostalCode,
//     Country,
```

```
//     HomePhone,
    Extension,
Photo,
Notes,
ReportsTo;
SQL SELECT *
FROM Employees;
```

29. Once this is done, once again "Save" the file and then "Reload".

To complete the in-memory associative database there are a couple of extra things that we need to do to the script. This will ensure that we have all of the fields that we require for our model. We need to ensure that the "OrderDate" field is actually a date and recognised as one by the QlikView document. We also need to add four new fields to make dates selection easier for the end user. The functions that we are using are well documented in the HELP screens and also the Reference Manual that we detailed earlier in this document.

30. Now, return to the Script Editor ("Control + E").

31. Next, we are going to leverage the existing OrderDate field to create the fields Day, Week, Month and Year within our Orders table using Qlikview date functions.

Modify your script as highlighted below

```
Orders:
LOAD OrderID,
CustomerID,
EmployeeID,
Date(OrderDate) as OrderDate,
Day(OrderDate) as Day,
Week(OrderDate) as Week,
Month(OrderDate) as Month,
Year(OrderDate) as Year,
RequiredDate,
ShippedDate,
ShipVia,
Freight,
ShipName,
ShipAddress,
ShipCity,
ShipRegion,
ShipPostalCode,
ShipCountry;
SQL SELECT *
FROM Orders;
```

Next, we are going to create two calculated measures Gross Revenue and Net Revenue for each invoice line within our OrderDetails table.

Modify your script as highlighted below

```
OrderDetails:
LOAD OrderID,
ProductID,
UnitPrice,
Quantity,
Discount,
(UnitPrice*Quantity) as GrossRevenue,
((UnitPrice*Quantity) - ((UnitPrice*Quantity)*Discount)) as NetRevenue;
SQL SELECT *
FROM `Order Details`;
```

32. Once you have made the changes, "Save" and "Reload" the data.

You have now completed the load script.

Building the End-User Interface

Now that the script is complete we can start to build the End User interface. QlikView has a wealth of options and possibilities for the End User interface with many different chart and table types as well as extended capability for the inclusion of .net or java based browser based objects, all fully interactive facilitating an intuitive informative interface.

For this document we are going to use just a few simple objects;

- A Set of time selection filters – allowing the end users to make selection from the data based on the order date within the data.

- A Drill Down Group – which allows the end users to drill through the time filters.

- A fully interactive "Time" based line chart, with drill down, showing Gross and Net Revenue.

- A Detailed list of Invoices that add value to the chart and give as much detail as is required by the end user.

- A fully reactive Speedometer Gauge showing the average levels of discount based on the selections made by the user.

- An Associative Search Object offering unrivalled search capabilities giving the end user the ability to search across all fields in the database, even when not on the display.

The first of these objects we need to create is the Drill Down Group. This will enable the Time based drill down functionality. To do this, follow the instructions below:

Drill Down Group.

A drill down group is used to allow users to interactively drill through a hierarchical layer either within a dimension or across dimensions. In our case we will drill down through a hierarchical layering of Time, being years into months, into weeks and finally into days.

1. From the menu, select "Settings" followed by "Document Properties". A window will appear that looks like the one shown in the image below.
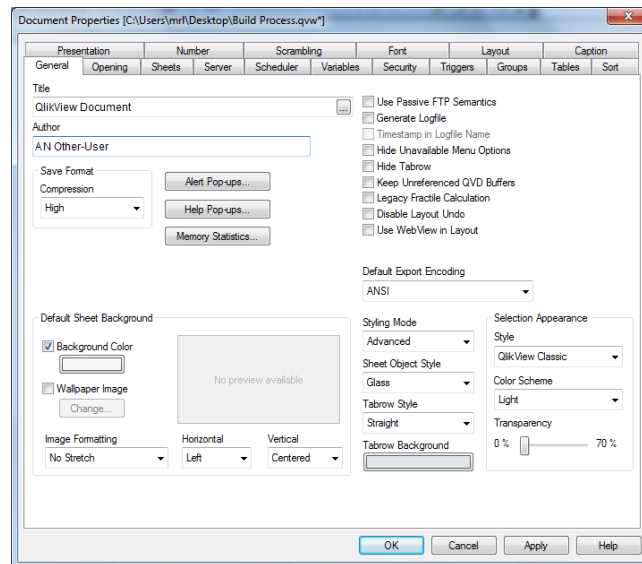
2. Select the "Groups" tab to create a new group. The screen will now look like the image shown below.
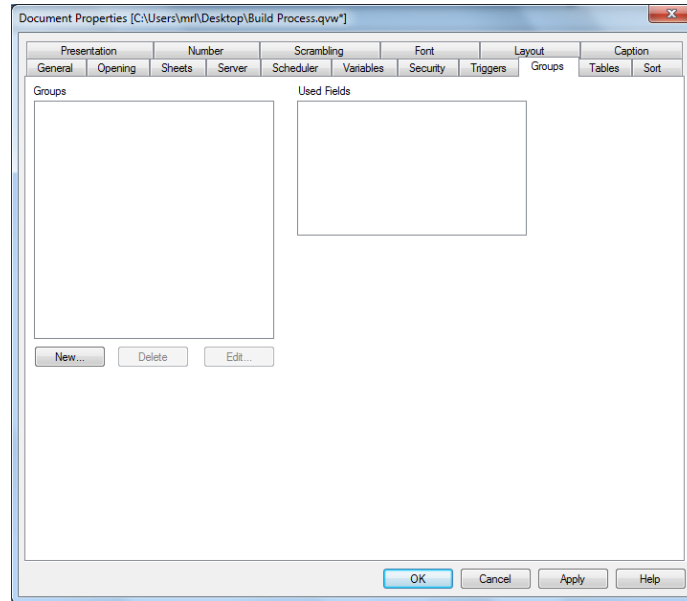
3. Click the "New" button to create a new group.

4. Next set the group up as demonstrated in the screen image below;
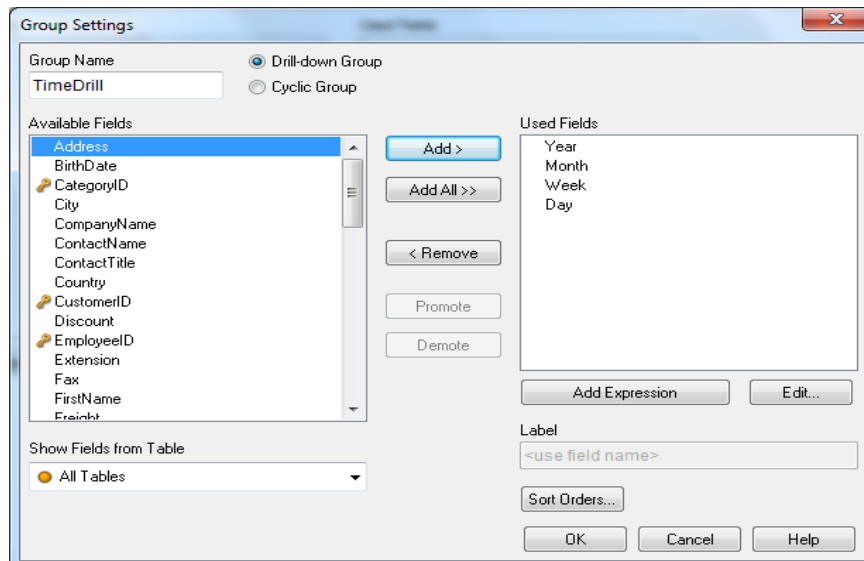
5. Click on "OK" once you are done.

6. Click on "OK" again to return to the document. You have now created the drill down group that will be used later on in the document.

Time Selection Fields

The first objects that you are going to build are the time selection fields that will act as filters for the end users.

1. To do this, right-click on a blank part of the screen and select the "Select Fields" option.
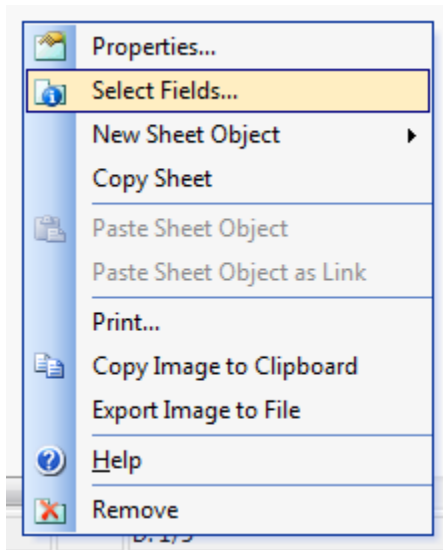
2. Next, select the following fields in the left half of the screen and use the "Add>" button to move them across to the right;

   Year, Month, Week, Day

3. Click "OK" to place them on the screen. They will appear in the top left. Move them to where you would like them to be.

Gross & Net Revenue Chart

Follow these steps to build the Gross and Net Revenue Chart:

1. Right-click on the screen and click New Sheet Object.

2. Next, click Chart to launch the chart wizard, as demonstrated in the screen image below.
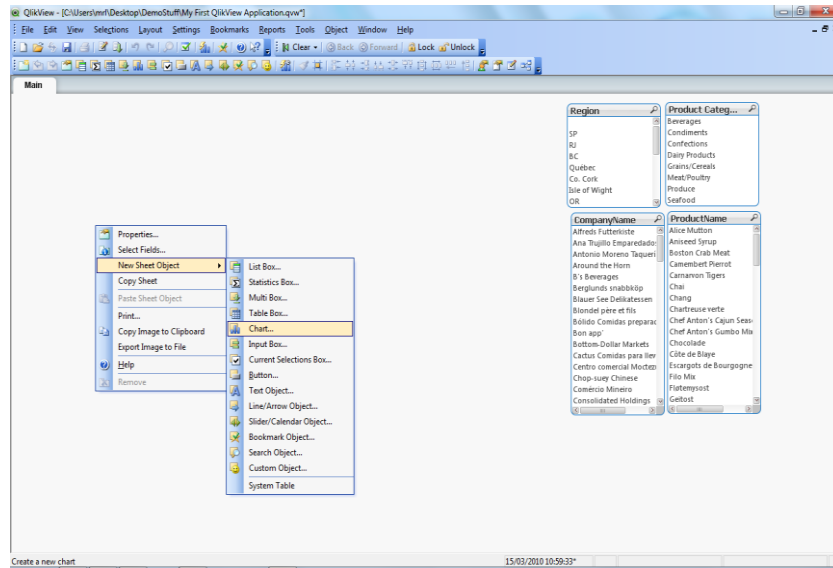
The chart wizard will appear. Follow the steps through the wizard ensuring that each page replicates the screen images following.

Step 1: Enter "Revenue Over Time" into the Window title expression field.

Step 2: In the Chart Type area, click the Bar chart image in the top left corner.
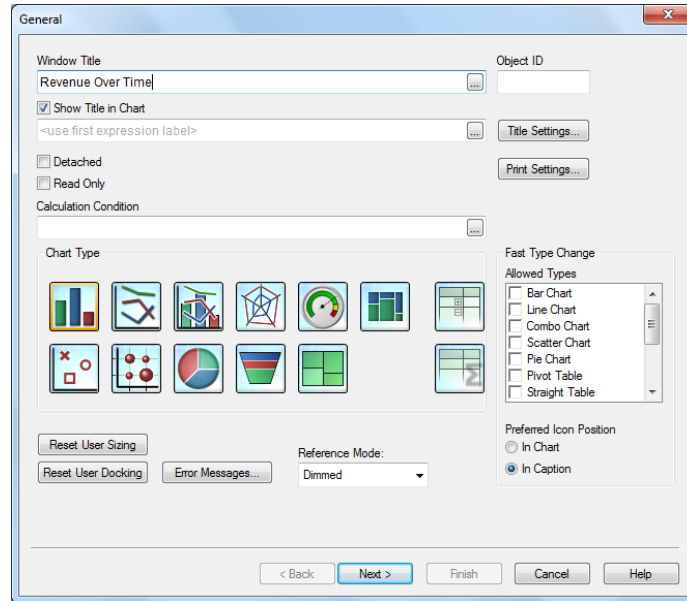
3. Click "Next" to continue through the wizard.

    Step 1: Select the group TimeDrill in the Available Fields/Groups area and click Add to move the group into the Used Dimensions area.
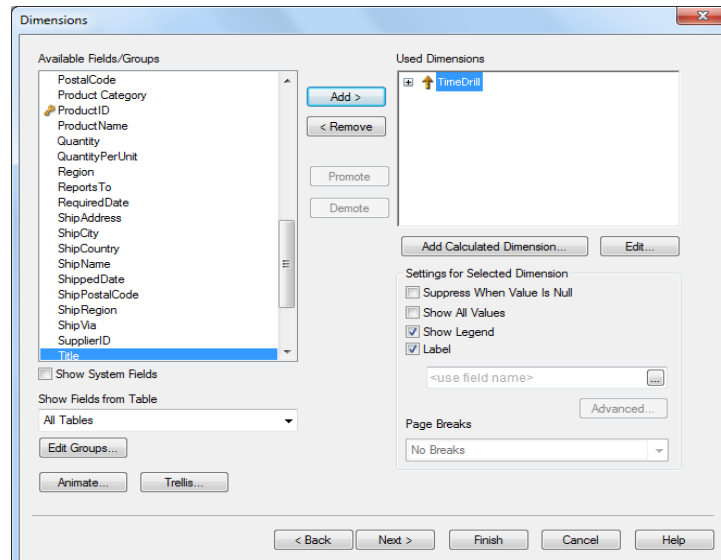
4. Click "Next" to continue through the wizard.

5. Because we intend to plot 2 different lines on the chart, we need to create 2 different expressions. ("Expression" is the term used for any calculation(s) used in QlikView to generate a value to be placed in a chart or table.)

The Expression editor will automatically open, allowing you to enter the first of the expression.

    i.   The first expression is used for the Gross Revenue of the invoice lines.

        i.   Enter the definition **"Sum (GrossRevenue)"** into the expression editor screen.
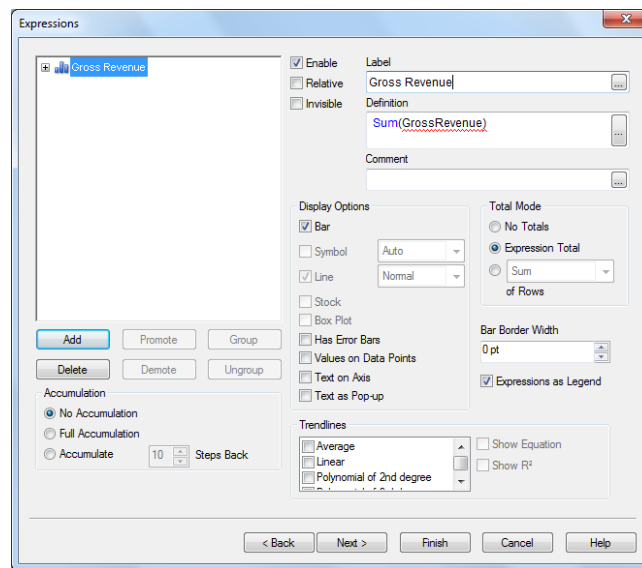
        ii.   Click OK to confirm the expression.

        iii.   Enter Gross Revenue into the Label above the expression definition.

        •

    ii.   Next, to create the second expression, click Add. The second expression is used for the Net Revenue of the invoice lines.

        i.   Enter the definition "**Sum (NetRevenue)"** into the expression editor screen.
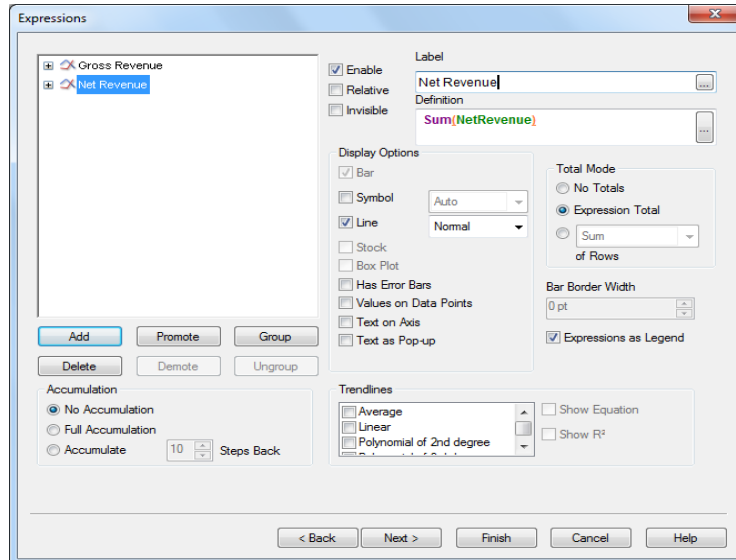
ii. Click OK to confirm the expression.
iii. Enter Net Revenue into the Label above the expression definition.

The two following expressions should now have been created.

Label	Expression
Gross Revenue	Sum(GrossRevenue)
Net Revenue	Sum(NetRevenue)

6. Once you have built both expressions, click "Next" to continue through the wizard.
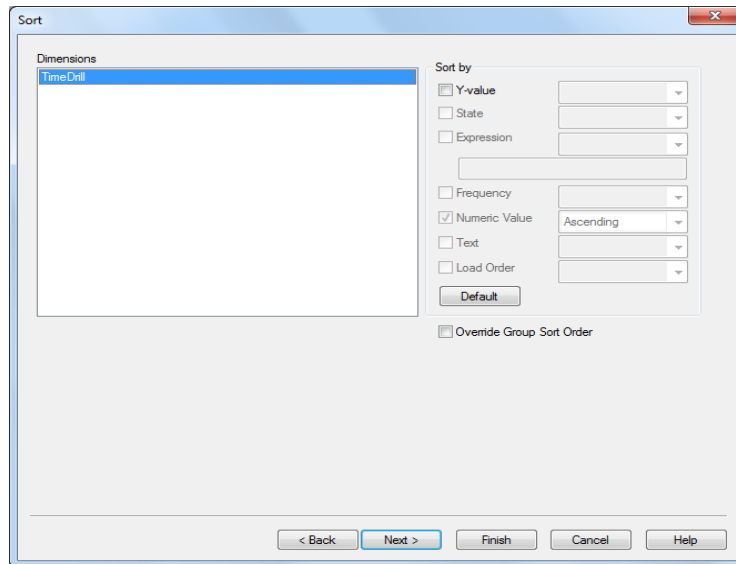
7. No sorting of the data is required, so just click "Next" to continue through the wizard.
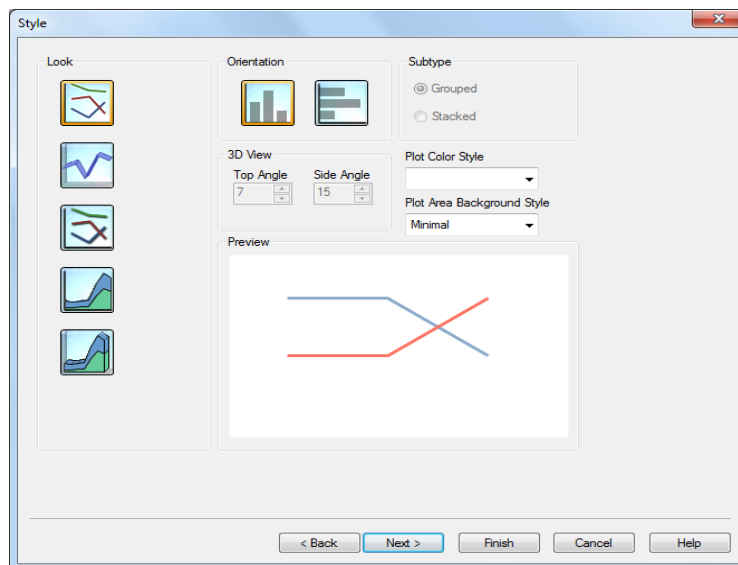
8. Here you can choose the appropriate chart look and feel. . Once you have selected your chart type, click "Next" to continue.
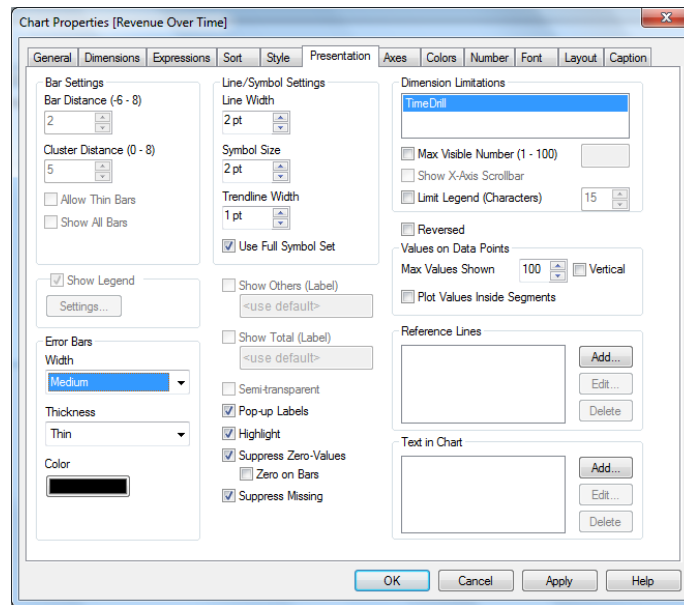
9. There are no changes required to the presentation tab, so just click "Next" to continue.
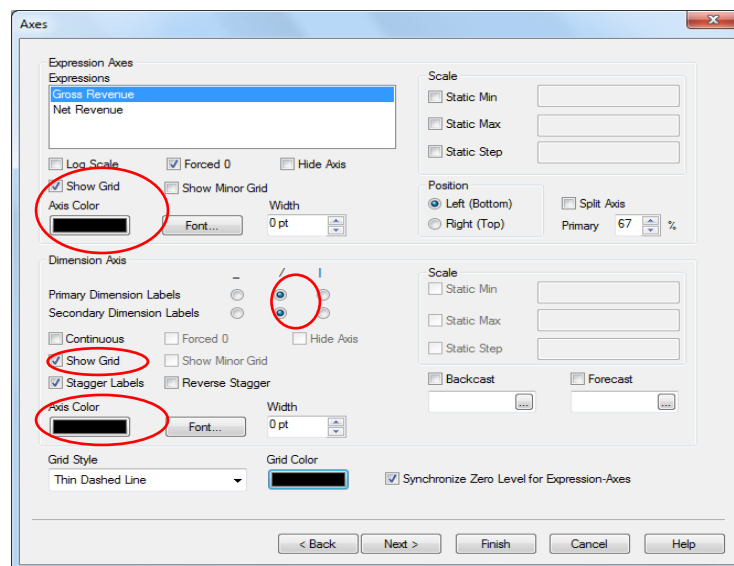
10. Change the Axis and Grid color; and the Dimension Label Orientation on the axis to ensure a clean presentation; then click "Next" to continue.
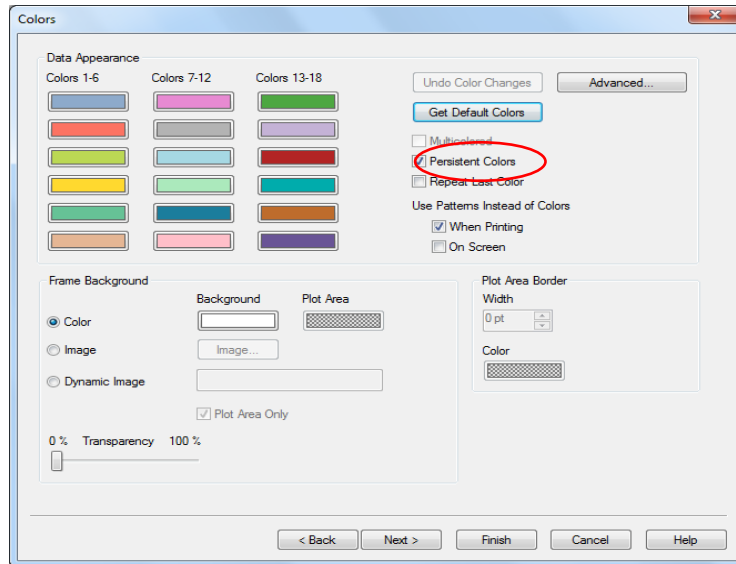
11. Here, ensure that "Persistent Colors" is checked and click "Next" to continue. If "Persistent Colors" is not checked, then when a drill down takes places the color of the line / bar or chart area may change and may make the chart more difficult to follow.
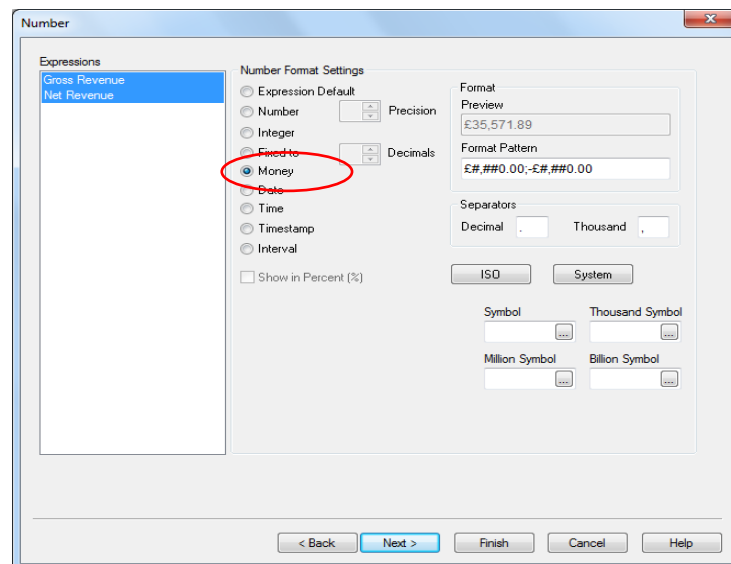
12. The final change to make to the chart wizard is to ensure that both expressions have their number format set to "Money".

13. Click "Finish" and your chart should appear very similar to that shown in the screen image below:
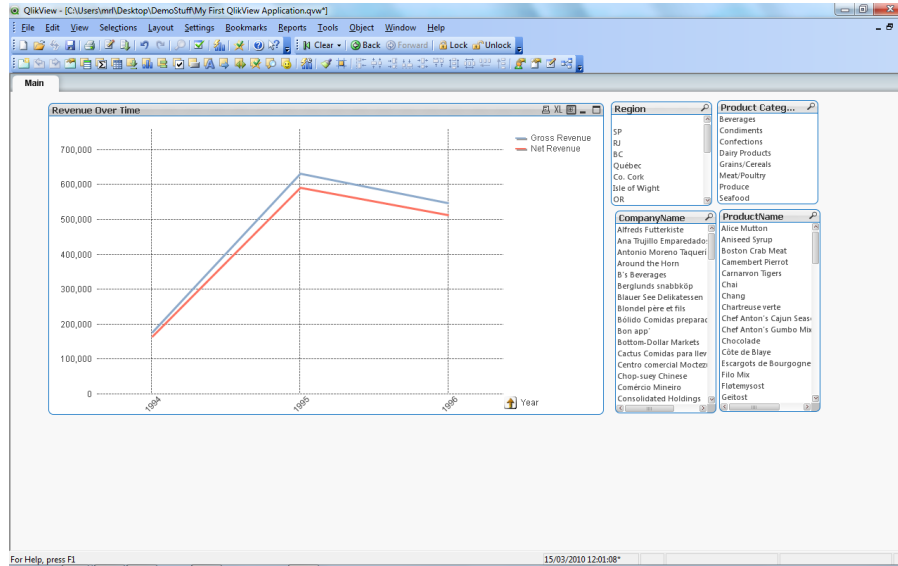
Now that the chart is complete we can move on to the Invoice Details Table.

Invoice Detail Table

To create the Invoice Details Table, you will need to create a "Table Box" object containing all of the necessary fields. This type of table is very important to QlikView as it is used to detail all of the lowest level data in the tables; which in this case are invoice lines or transactions. Using a traditional B.I. tool it is likely that you would not be able to have both the most granular level of data and KPIs in the same application, due to data volume and functionality restrictions.

1. To create this object, right-click on a blank part of the screen and select "New Sheet Object" followed by "Table Box". The following wizard will appear. Once again complete the steps as illustrated in the following screen shots:
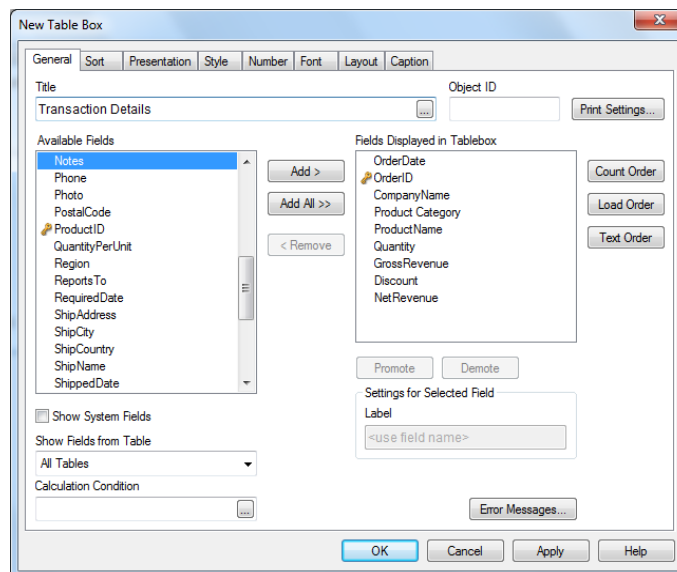
2. Select the fields in the list on the left hand side of the screen, then use the "Add>" button to move them across to be included within the table box.

3. Once the fields have been moved across, you should use the "Promote" and "Demote" buttons to get them into the correct order, as listed below.

- OrderDate
- OrderID
- CompanyName
- Product Category
- ProductName
- Quantity
- Gross Revenue
- Discount

- Net Revenue

4. Next, click "OK" button to finish.

Your report object should look similar to the one shown in the screen image below:
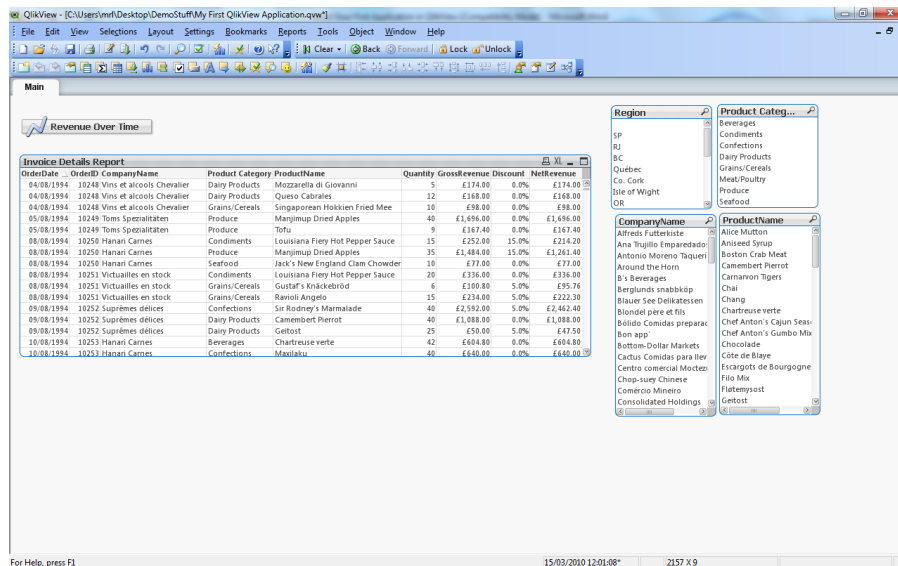
The differences between your screen and the image above are due to number formatting. To achieve the same formatting enter the properties of the Table box (by right-clicking on the Table and selecting 'Properties' from the menu) and modify the number formatting on the "Number" tab. This table box now contains selected fields from each "transaction". The number of records available in the table box is reduced when a selection is made in one of the field selection boxes created earlier.

We can now move onto to the "Average Discount" Speedometer gauge

Average Discount % Speedometer Gauge

The next reporting object to be created will be a speedometer gauge that shows the KPI for the Average Discount % rate. It is important to note that the value shown will change as selections are made in the selection fields that have already been created.

1. To create the gauge, right-click on the screen and select the "New Sheet Object" followed by the "Chart" option. The following wizard will appear. Follow the wizard steps and amend each screen in line with the following screen images:
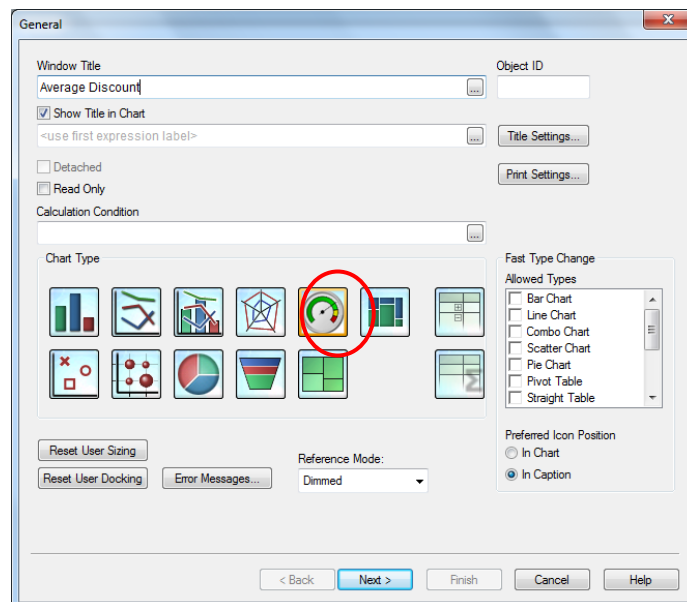
2. Complete the fields as shown and click "Next" to continue through the wizard.
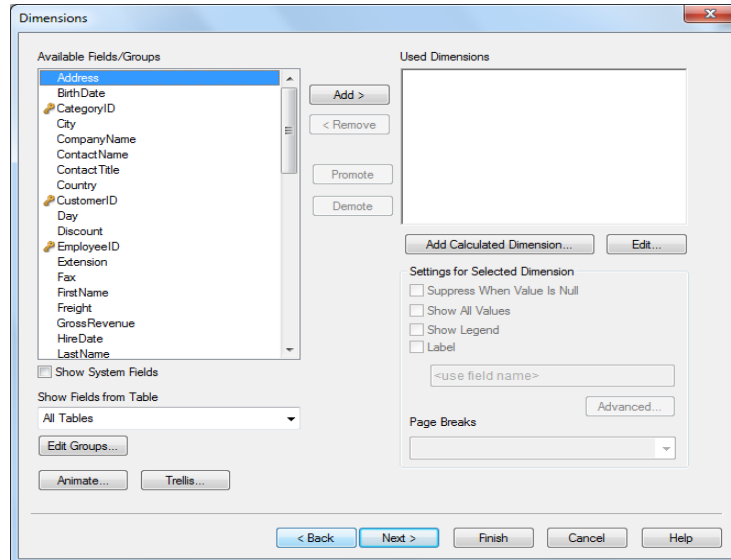
3. In QlikView, Gauges do not have dimensions, so click "Next" to continue.

4. The next step is to write the expression that will be used to put the value into the gauge, this one is quite simple, just an average statement. You could either manually type the expression **"Avg(Discount)"** or build the expression using the aggregation and field drop down lists and click "Paste".

   The "Aggregation" drop down will allow you to select the function that you wish to use in the expression and the "Field" drop down will allow you to select the field that you wish to apply the function to. Normally any syntactical brackets are written for you when using the drop downs and the "Paste" button.
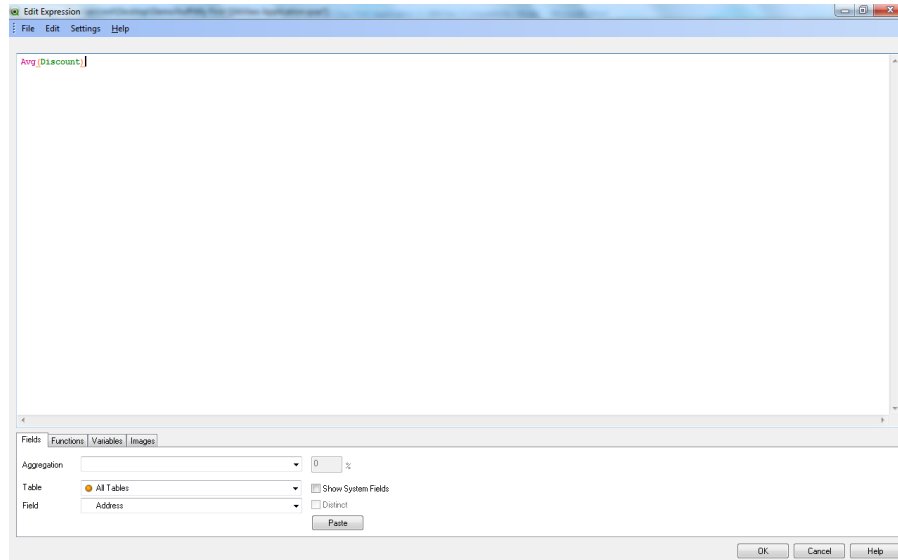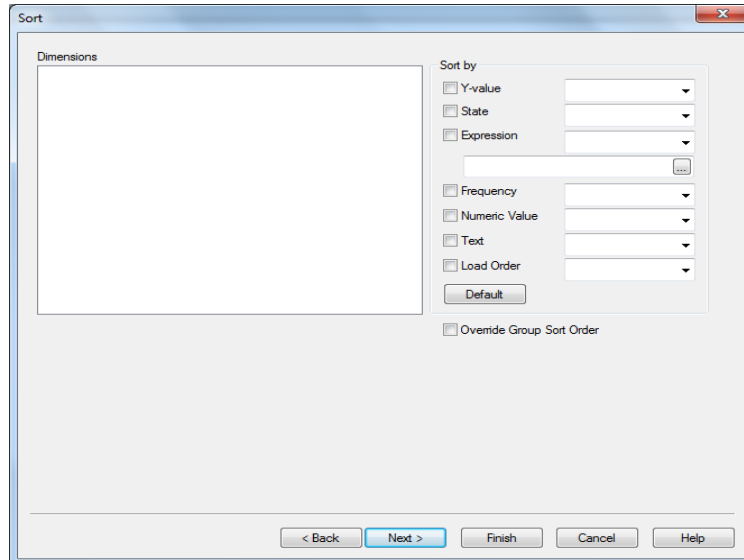
5. Next, click on "OK" to finish the expression.

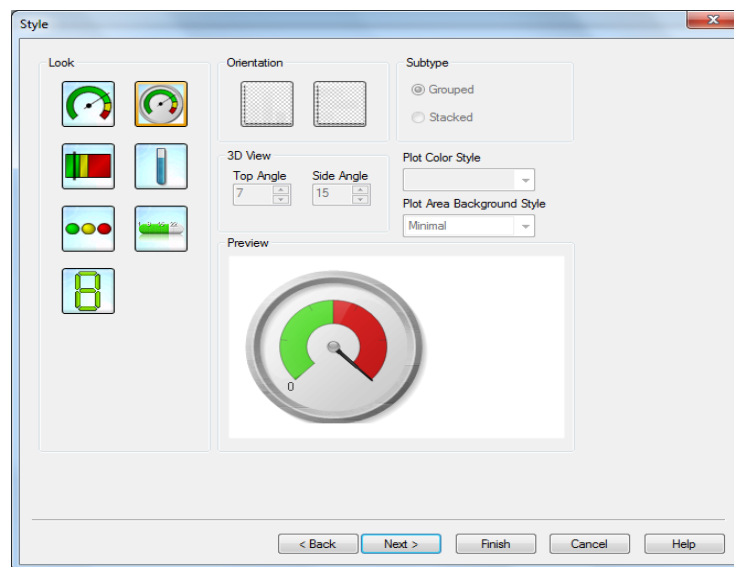6. Now, label the expression "Average Discount"
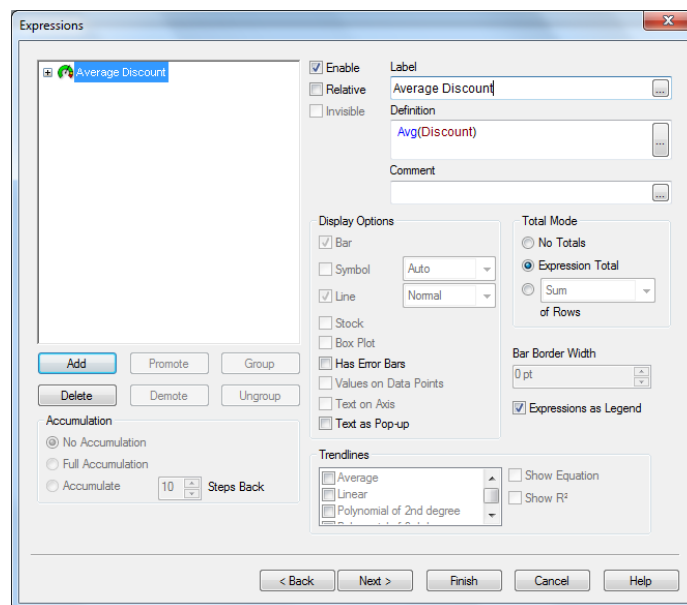
Click "Next" to continue.

7. As a gauge does not have a dimension no sorting is required, so click "Next" to continue.
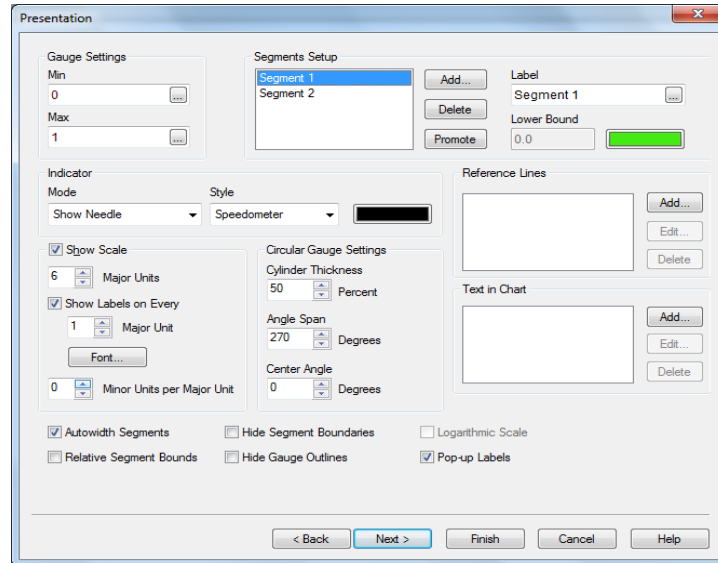
8. At this point decide what you would like your gauge to look like
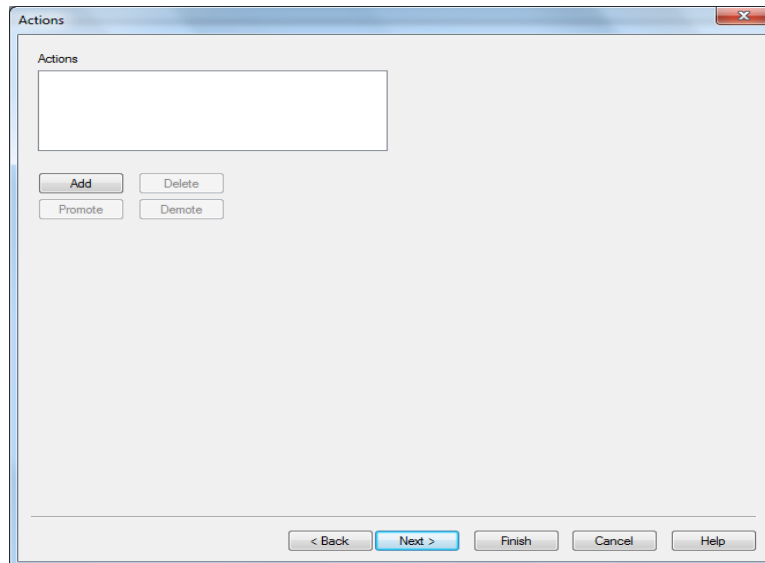

Click "Next" to continue.

9. On this screen, define the scope of the color ranges to be shown in the gauge and the number of labels and markers on the dial.
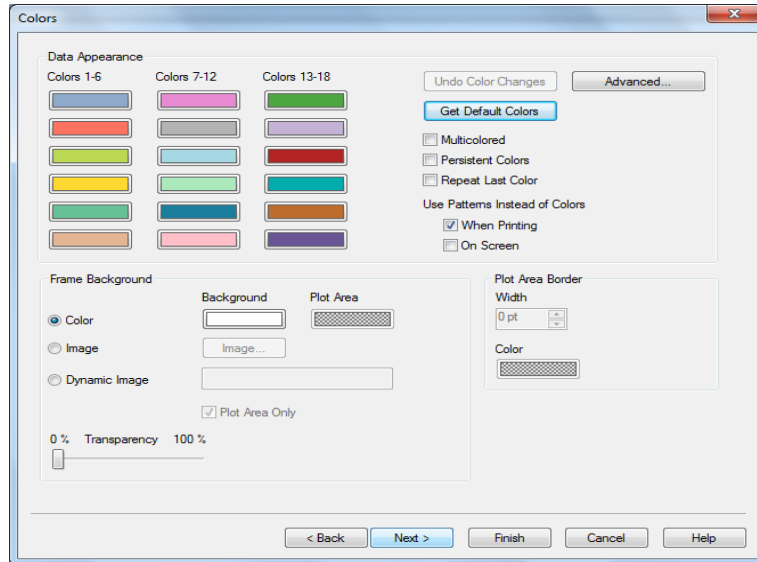

Once you have set the values as shown above, click "Next" to continue.
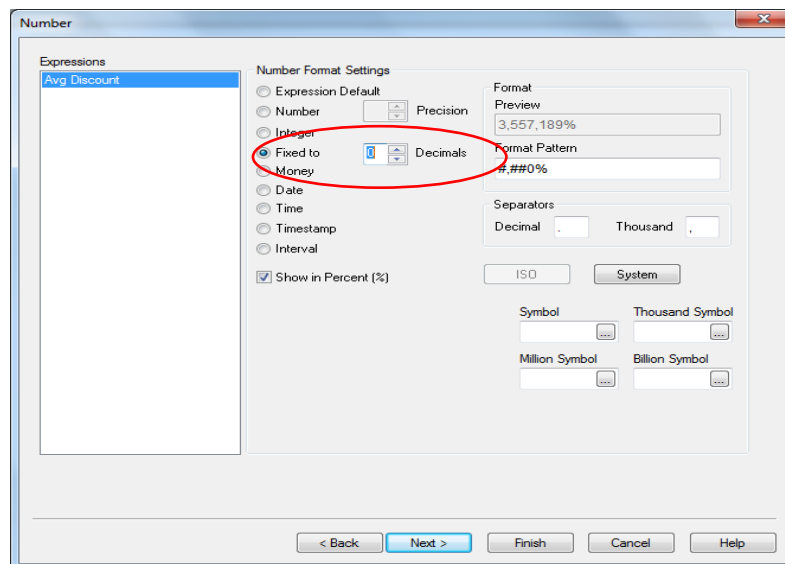
10. There are no actions required for this gauge,

Click "Next" to continue.

11. There are no additional color requirements.

Click "Next" to continue through the wizard.

12. The last change to make in this wizard is to define how the value will be presented as a number. Please make the change to the wizard as detailed below:

Once done, click Finish to end the wizard and present the gauge to the document screen.

Your Gauge should look similar to the one shown in the screen image below:
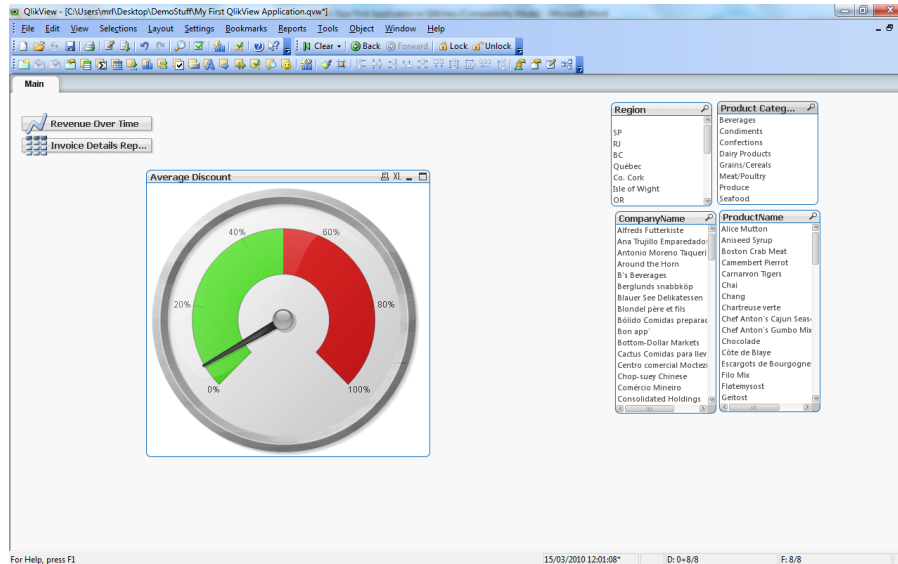
The gauge is now complete. Next we will build the associative search object.

![QlikView]

An Associative Search Object

An associative search object is an object that allows the end user to search through the entire data set, whether currently displayed or not, and find any or all matches to their chosen text string. It could be the name of a field or a data item value in a field. Any matches found will be available as a selection for the user to choose.

1. To build the associative search object, right-click on a blank part of the screen and select 'New Sheet Object" followed by the "Search Object" option. This will produce the following screen?
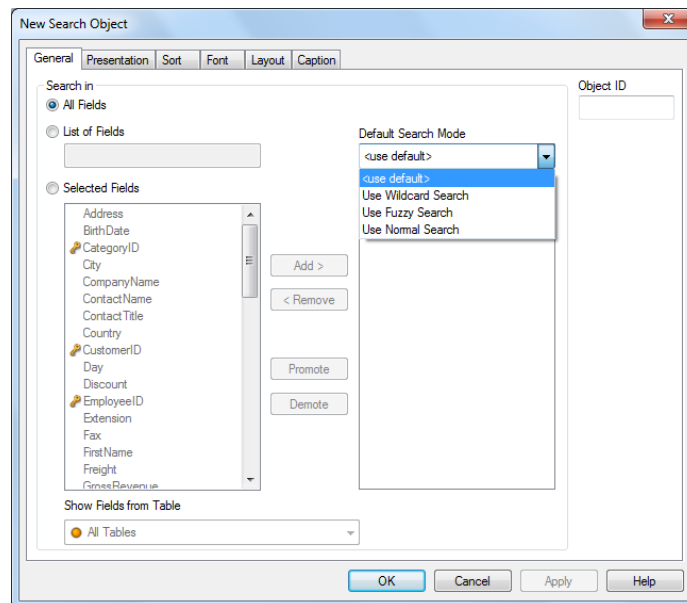
2.   Leave the settings as they are as these default settings will allow the search object to search across all fields in the in-memory associative database. Click "OK" to create the object.

3.   Place the object on the screen where you think it would be most useful to the end user. You may want to bear in mind that the results from the search engine are presented downwards from the location of the search object. The ideal location for the search object is near the top of the screen.

Summary

Congratulations! You have built your first QlikView document. You can improve the

aesthetics of the document by further manipulation of the settings for the objects. Please feel free to do so in your own time. With practice, you can make your QlikView documents look like this...