



QLIKVIEW RECIPES

Byte size QlikView tips and tricks to solve real world business discovery and dashboarding problems.



By Rajesh Pillai & Radhika Pillai

Contents

Prologue 2

Section 1 – Charts 3

 1. Fast Change Chart Using Action button (Macro) 3

 2. Ad Hoc Analysis 10

Prologue

The story goes to couple of years back, when I was getting into this beautiful world of QlikView, when I was first introduced and since then have been making every effort to master this tool in every way possible. Since last two or more years, I have trained more than 200 people on this tool and every time I go for a training, a new problem or data modelling challenge comes up.

Most of these challenges are already solved in some or other way, but there is no single documentation of the collective learnings of QlikView experts.

This book serve to fulfill that void and build up a treasure house of tips and tricks for every possible QlikView problems (well at least theoretically).

Also, this is an effort to learn from my past writing mistakes. My first QlikView book still in development is “QlikView for Ninjas”, available on leanpub.com, but it was delayed a lot, not due to time constraint but due to the lack of effective understanding of the publishing process, the editing part etc. “QlikView for Ninjas” has a specific purpose and will be continuously updated over a period of time.

The beauty of eBook is that the content can be updated, rewritten any number of times, without worrying about the process of printing, distribution. In that sense, this book aim to server as a ready to apply recipes for some of the recurring problems that a user faces while working with QlikView (or in some sense QlikSense as well).

- Rajesh Pillai

Section 1 – Charts

This section contains tips and tricks for manipulating charts using advanced expressions, macros etc.

1. Fast Change Chart Using Action button (Macro)

One of the frequent requirement is to have “Fast change chart type” using button, rather than using the built in “Fast Change Icon”. There are various reasons for this, one being, the dashboard being usable on mobile devices.

Steps

1. Create a new qlikview application.
2. Add the below script. The script creates a “Sales” table with some sample data.

Sales:

```
LOAD * INLINE [  
    Day, Month, Year, Accounts, Sales  
    1, Jan, 2014, 1, -900  
    2, Feb, 2015, 2, 200  
    7, Mar, 2015, 3, 250  
    9, Jan, 2014, 4, -100  
    5, Apr, 2014, 7, 700  
    6, Sep, 2014, 5, 200  
    3, Oct, 2014, 3, 200  
    8, Dec, 2014, 8, -900  
    10, Aug, 2014, 5, 500
```

```
];
```

3. Save and Reload the script.
4. Create a Line Chart with the following configurations.
 - a. Chart Properties: General Tab: Fast Type Change
 - i. Select Line Chart and Bar Chart
 - b. Dimensions: Accounts
 - c. Expressions: Sum (Sales), Label: Sales \$

Chart Properties [Sales]

General Dimensions Dimension Limits Expressions Sort Style Presentation Axes Colors Number Font

Window Title

Sales

Object ID

CH04

Show Title in Chart

<use first expression label>

Title Settings...

Detached

Print Settings...

Read Only

Calculation Condition

Chart Type



Fast Type Change

Allowed Types

- Bar Chart
- Line Chart
- Combo Chart
- Scatter Chart
- Pie Chart
- Pivot Table
- Straight Table

Preferred Icon Position

- In Chart
- In Caption

Reset User Sizing

Reference Mode:

Reset User Docking

Error Messages...

Dimmed

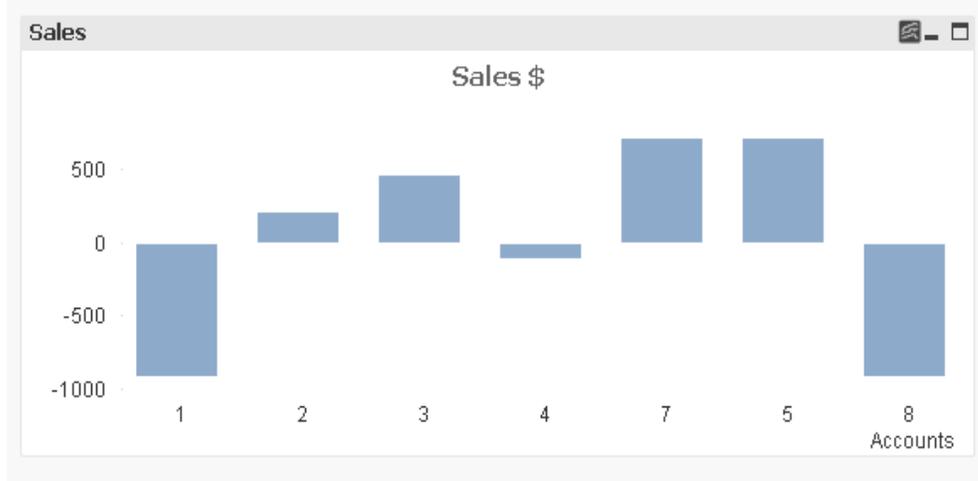
OK

Cancel

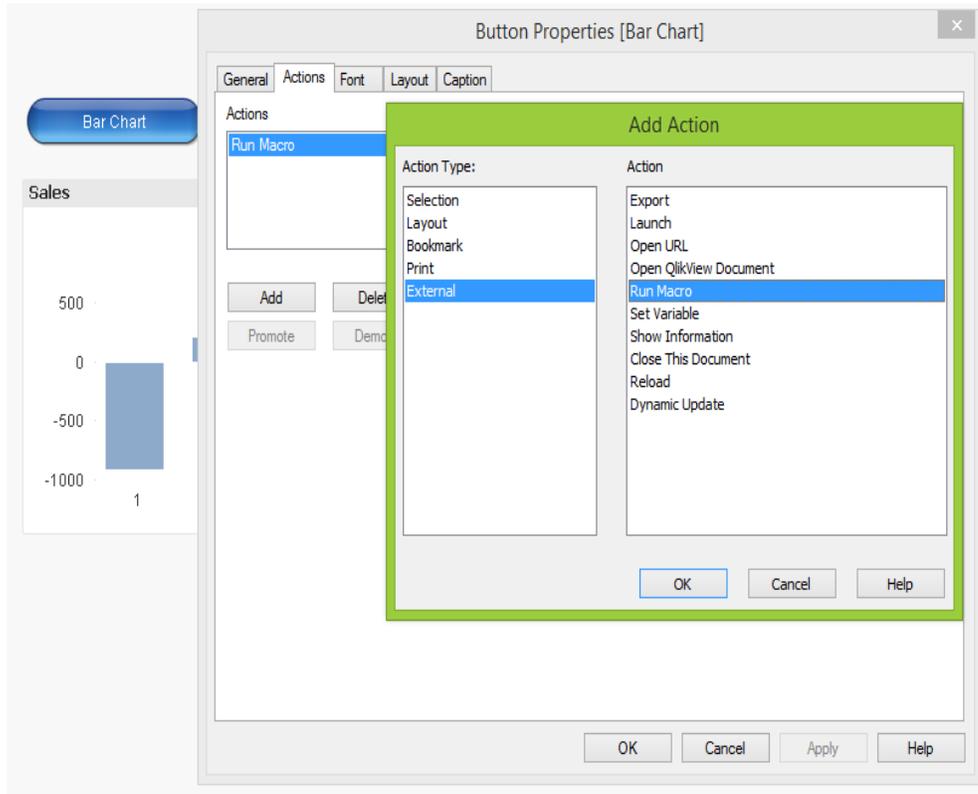
Apply

Help

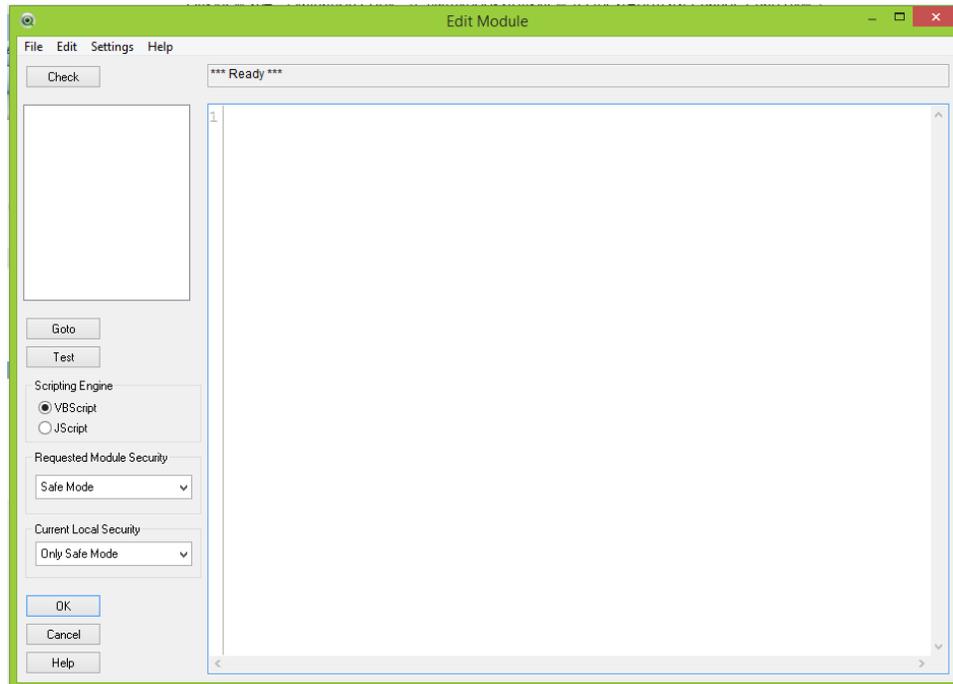
5. The chart should look like the figure below.



6. Add a button to the dashboard with the following properties
- General Tab: Text: Bar Chart
 - Actions Tab: Click on Add button. From the “Action Type” select “External” and from “Action” select “Run Macro” and Click “Ok”. Refer the figure below.



- c. In the “Macro Name” enter the name “SetBarChart” and click on “Edit Module” to bring up the macro editor as shown below.



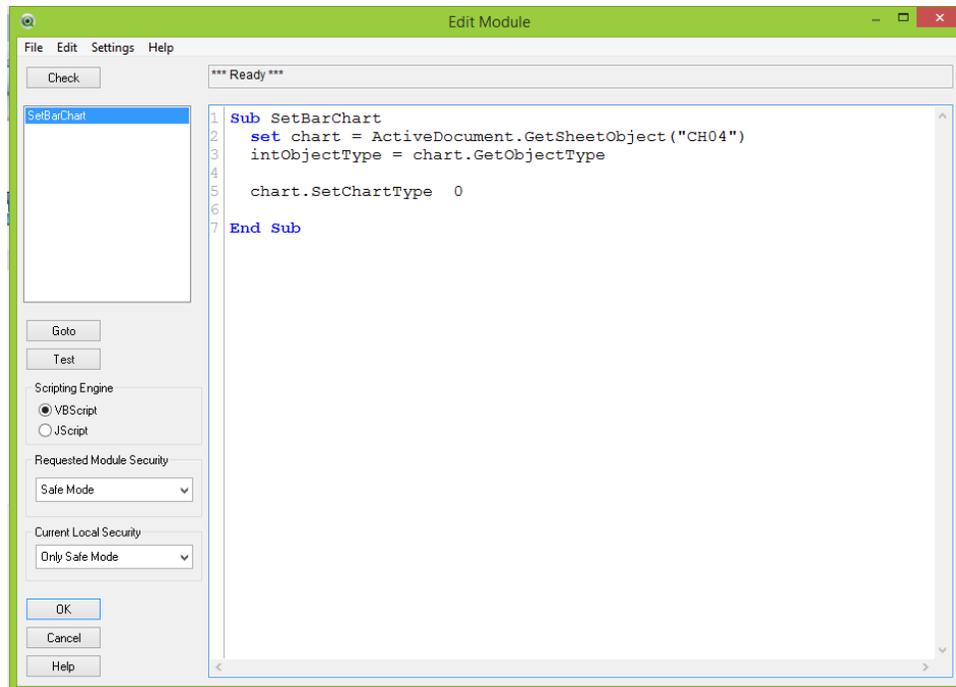
- d. Paste the below code in the editor and click on “Check” button to verify for any error. Please check the ID of the chart and pass that to the GetSheetObject parameter. In my case the ID of the chart is “CH04”

```
Sub SetBarChart
    set chart = ActiveDocument.GetSheetObject("CH04")
    intObjectType = chart.GetObjectType

    chart.SetChartType 0

End Sub
```

- e. The screen should look like below figure.



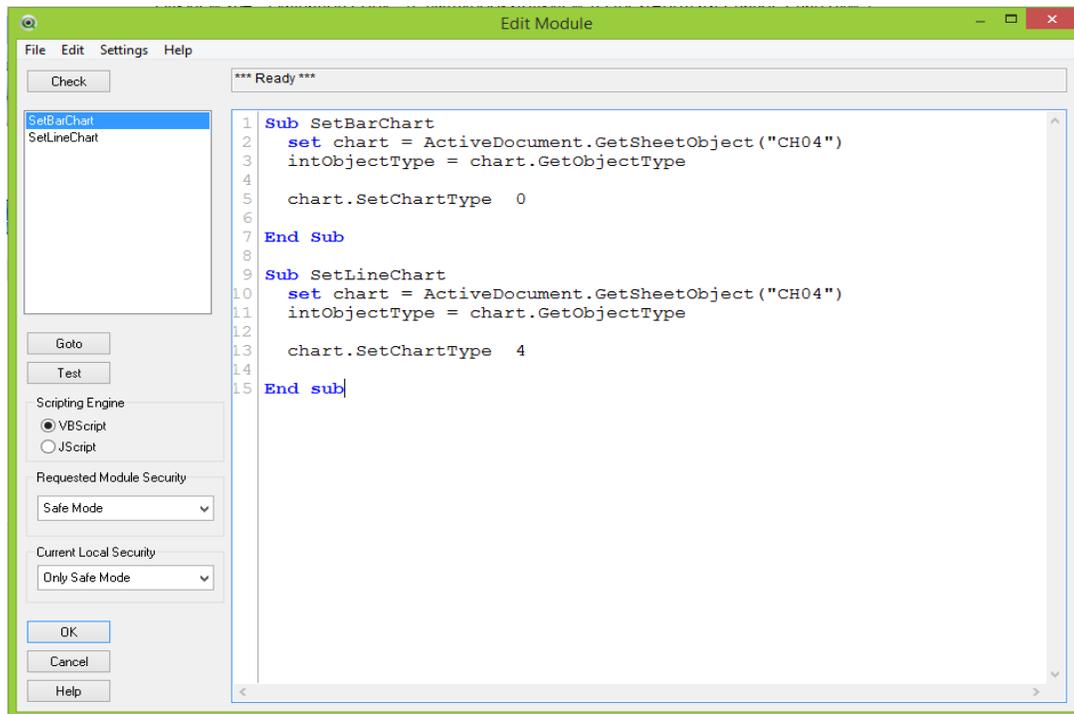
7. Add one more “Button” and change the properties as below.
 - a. Text: Line Chart
 - b. Actions Tab: Click on Add button. From the “Action Type” select “External” and from “Action” select “Run Macro” and Click “Ok”.
 - c. In the “Macro Name” enter the name “SetLineChart” and click on “Edit Module” to bring up the macro editor as shown below.

```
Sub SetLineChart
    set chart = ActiveDocument.GetSheetObject("CH04")
    intObjectType = chart.GetObjectType

    chart.SetChartType 4

End sub
```

- d. The full macro script is shown below. Click on the “Buttons” and see the effect on the chart.



Explanation:

```
1 Sub SetBarChart
2   set chart = ActiveDocument.GetSheetObject("CH04")
3   intObjectType = chart.GetObjectType
4
5   chart.SetChartType 0
6
7 End Sub
```

Line 1 Defines a new function/procedure with the name SetBarChart

Line 2 Gets the chart object with the ID “CH04” and stores it in a variable named chart.

Line 3. Reads the Object Type of the chart and stores it in a variable. Note this value is not used here. If you want you can skip this step.

Line 5. Sets the chart type to 0. (0 means Bar Chart and 4 means Line Chart)

The following figure shows the object types and chart types.

Fast Change Chart Types	Object Types
0=Bar	0=Unknown
1=Pie	1=List Box
2=Pivot	2=Multi Box
3=Scatter	3=Statistics Box
4=Line	4=Table Box
5=Straight Table	5=Button
6=Combo	6=Text Object
7=Radar	7=Current Selections Box
8=Gauge	8=Input Box
9=Grid	9=Line/Arrow Object
10=Block	10=Pivot Table
11=Funnel	11=Straight Table
	12=Bar Chart
	13=Pie Chart
	14=Scatter Chart
	15=Line Chart
	16=Combo Chart
	17=Custom Object
	18=Bookmark Object
	19=Slider/Calendar Object
	20=Grid Chart
	21=Radar Chart
	22=Gauge Chart

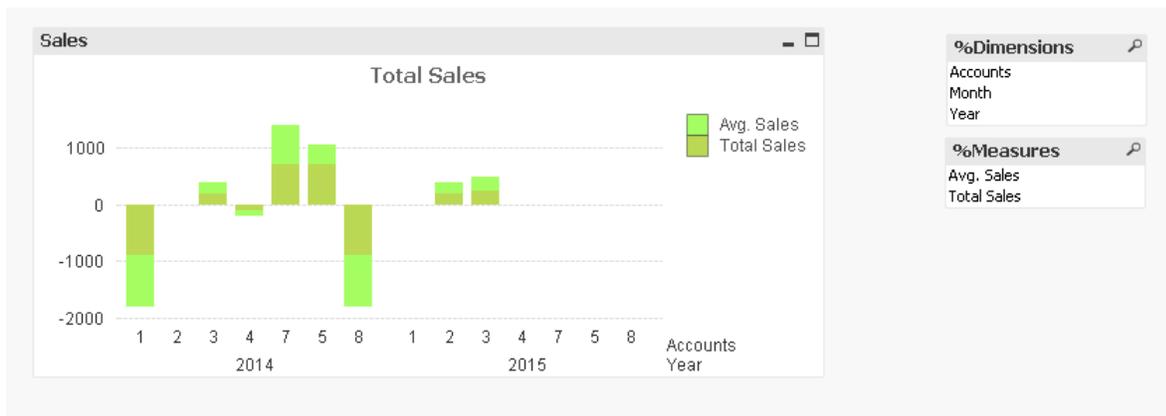
NOTE: This macro works also on the QlikView server.

2. Ad Hoc Analysis

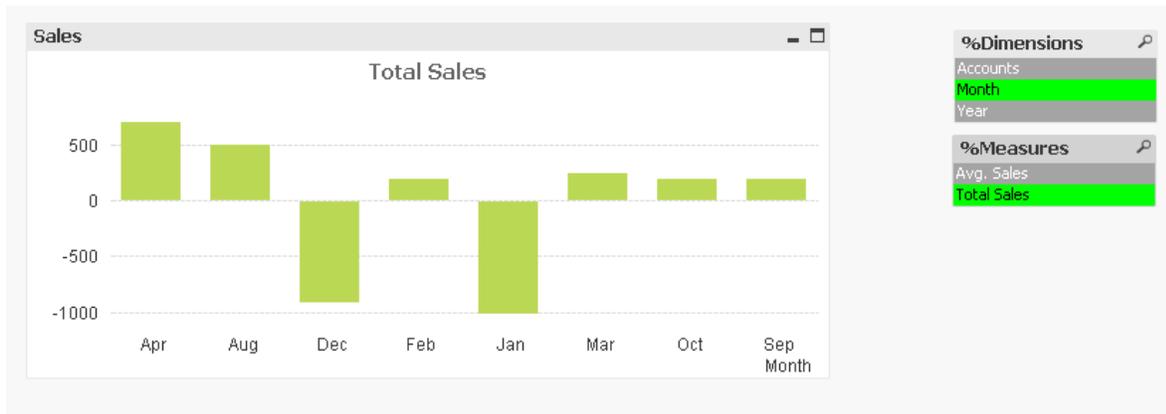
One of the frequent requirements is to have a dynamic capability where the user can select the dimensions and the expressions for the chart. For this recipe we will show all dimensions and expressions initially and then allow users to further filter the chart using a dimension and measures listbox.

The completed recipe should look like the figure below.

User selections: None



User selections: Month, Total Sales



User selections: Month, Avg. Sales, Total Sales



Steps

1. Create a new qlikview application.
2. Open up the edit scripts and create two tabs, "Inline Tables" and "Facts".
3. In the "Inline Tables" tab copy the below script. This script loads up the dimensions and measures for dynamic filtering. Please note, for demo I am writing most of the data loading as inline scripts, but feel free to source this from any of your data source as required.

```
// Load the list of dimensions
DimensionList:
LOAD * Inline [
%Dimensions
Accounts
Year
Month
];

// Load the list of expressions
ExpressionList:
LOAD * Inline [
%Measures
Total Sales
Avg. Sales
];
```

4. In the "Facts" tab in the edit script window, add the below script.

```
Sales:
LOAD * INLINE [
Day, Month, Year, Accounts, Sales
1, Jan, 2014, 1, -900
2, Feb, 2015, 2, 200
7, Mar, 2015, 3, 250
```

```

9, Jan, 2014, 4, -100
5, Apr, 2014, 7, 700
6, Sep, 2014, 5, 200
3, Oct, 2014, 3, 200
8, Dec, 2014, 8, -900
10, Aug, 2014, 5, 500
];

```

5. Save and Reload the script.
6. Create a simple “Bar Chart” with the following properties
 - a. Title: Sales
 - b. Dimensions:
 - i. Year
Check “Enable Conditional” and put the below condition.
`=SubStringCount('|' & Concat(distinct %Dimensions, '|') & '|', '|Year|')`
 - ii. Accounts
Check “Enable Conditional” and put the below condition.
`=SubStringCount('|' & Concat(distinct %Dimensions, '|') & '|', '|Accounts|')`
 - iii. Month
Check “Enable Conditional” and put the below condition.
`=SubStringCount('|' & Concat(distinct %Dimensions, '|') & '|', '|Month|')`
 - c. Expressions:
 - i. Sum (Sales), Label: Total Sales
Check “Conditional” and put the below condition.

`=SubStringCount('|' & Concat(distinct %Measures, '|') & '|', '|Total Sales|')`
 - ii. Avg (Sales), Label: Avg. Sales
Check “Conditional” and put the below condition.

`=SubStringCount('|' & Concat(distinct %Measures, '|') & '|', '|Avg. Sales|')`

Explanation

The script to achieve this functionality is very simple. First we add all the dimensions and expressions that is required and then we put conditions on each of the dimensions and the expressions.

The explanation of the two functions is extracted for QlikView help for your reference.

SubStringCount - script and chart function

SubStringCount() returns the number of occurrences of the specified substring in the input string text. If there is no match, 0 is returned.

Syntax:

```
SubStringCount(text, sub_string)
```

Return data type: integer

Arguments:

Argument	Description
text	The original string.
sub_string	A string which may occur one or more times within the input string text .

[Examples and results:](#)

Example	Result
SubStringCount ('abcdefgdcxyz', 'cd')	Returns '2'
SubStringCount ('abcdefgdcxyz', 'dc')	Returns '0'

In our example the below expression

```
=SubStringCount(' | ' & Concat(distinct %Dimensions, '|') & '|', '| Accounts|')
```

Will return true or -1 when “Accounts” is selected or no selection is made at all. If some other value is selected then this expression will return false or 0.

Concat - chart function

Concat() is used to combine string values. The function returns the aggregated string concatenation of all the values of the expression evaluated over each dimension.

Syntax:

```
Concat( {[SetExpression] [DISTINCT] [TOTAL [<fld {, fld}>]] string[, delimiter[, sort_weight]] )
```

The expression =Concat(distinct %Measures, '|') will give the output as Avg. Sales | Total Sales if no measure is selected.

If “Total Sales is selected then the output will be **Total Sales**.

NOTE: If the conditional expression evaluates to true then the expression is shown otherwise not shown.