Technical Paper:

Using Dynamic Update in QlikView

v1.0

CBO

QlikTech

June 2010

www.qlikview.com

QlikView

Technical Paper:

Using Dynamic Update in QlikView

Table of Contents:

Introduction

This application uses the Dynamic Update function to automatically update a daily stock dashboard application each minute during normal trading hours. The application allows the user to view predetermined stocks as well as daily market information for the Dow Jones, Nasdaq and the S&P 500 in a dashboard form. Subsequent tabs allow the user to see detailed company information for a single company. The purpose of this application is to show how QlikView's dynamic update function can be used to show real time data updates within a qvw. For this example we are using only 1 QlikView Server (QVS).

Dynamic Update vs. Reload

QlikView's reload function is a "pull" function. The reload function pulls the data from the data source and stores it in the qvw. Conversely, the dynamic update function is a "push" function. The dynamic update function pushes the data from the data source into the QlikView server memory. The dynamic update data does not get saved into the qvw and thus if there is any interruption with the QlikView server the dynamic update data is lost.

One way to help minimize the data loss in a dynamically updated application is to store the data that is pushed into the dynamically updated application into a QVD and then reload the QVD into the dynamically updated application every couple of minutes.

Software and Versions Used

QlikView – version 9.00.7320.0409

Microsoft Excel 2007

Files Used and Workflow

Files:

Stock App Stocks.xlsx – This file contains the companies that are to be tracked. It is referenced in the Dynamic_Update.qvw.

Stock_Images.xlsx – This file contains the paths for the bundles images within the dashboard. It is referenced in the Real Time Stocks.qvw.

My Share Breakdown (inline table) – This table is a list of stocks within the users portfolio. It contains the buy price for each stock. It is referenced in the Real Time Stocks.qvw

Markets.qvd – This QVD contains a time stamped history for each of the 3 stock markets tracked. It is referenced in both the Dynamic_Update.qvw where each new instance of market data is written out to the QVD and it is referenced the Real Time Stocks.qvw when a full reload of the QVW is performed.

StocksDU.qvd – This QVD contains a time stamped history for each of the stocks referenced within the Stock App Stocks.xlsx. It is referenced in both the Dynamic_Update.qvw where each new instance of stock data is written out to the QVD and it is referenced the Real Time Stocks.qvw when a full reload of the QVW is performed.

RealtimeDU.bat – This batch file triggers the Dynamic_Update.qvw to reload on a 1 minute interval each weekday from 9:45am EST to 4:15pm EST.

QVW:

Dynamic_Update.qvw – This QVW goes out to MSN Money to scrape stock information for each of the companies defined within the ***Stock App Stocks.xlsx*** and it also scrapes the most recent stock market data for the Dow Jones, Nasdaq and the S&P 500. After the data scrape, the new data instances are then formatted into a Dynamic Update string using variables. The data is also written out to both ***Stocks.qvd*** and the ***Markets.qvd*** for historical purposes. There is a macro within the application (***Insert***) that triggers upon post reload. The ***Insert*** macro performs the dynamic update of the new data into the ***Real Time Stocks.qvw.*** The ***Dynamic_Update.qvw*** is reloaded every minute.

Real Time Stocks.qvw – This QVW is a daily stock and stock market dashboard. It also contains detailed information for each of the companies defined in the ***Stock App Stocks.xlsx. The Real Time Stocks.qvw*** should be reloaded Monday-Friday 9:45am-4:15pm EST. To view the dynamic update of the ***Real Time Stocks.qvw***, the user should open the qvw in server mode.

Workflow:

1. .bat file is a scheduled task that is triggered every minute M-F from 9:45am-4:15pm. The .bat file triggers the reload of the Dynamic_Update.qvw.

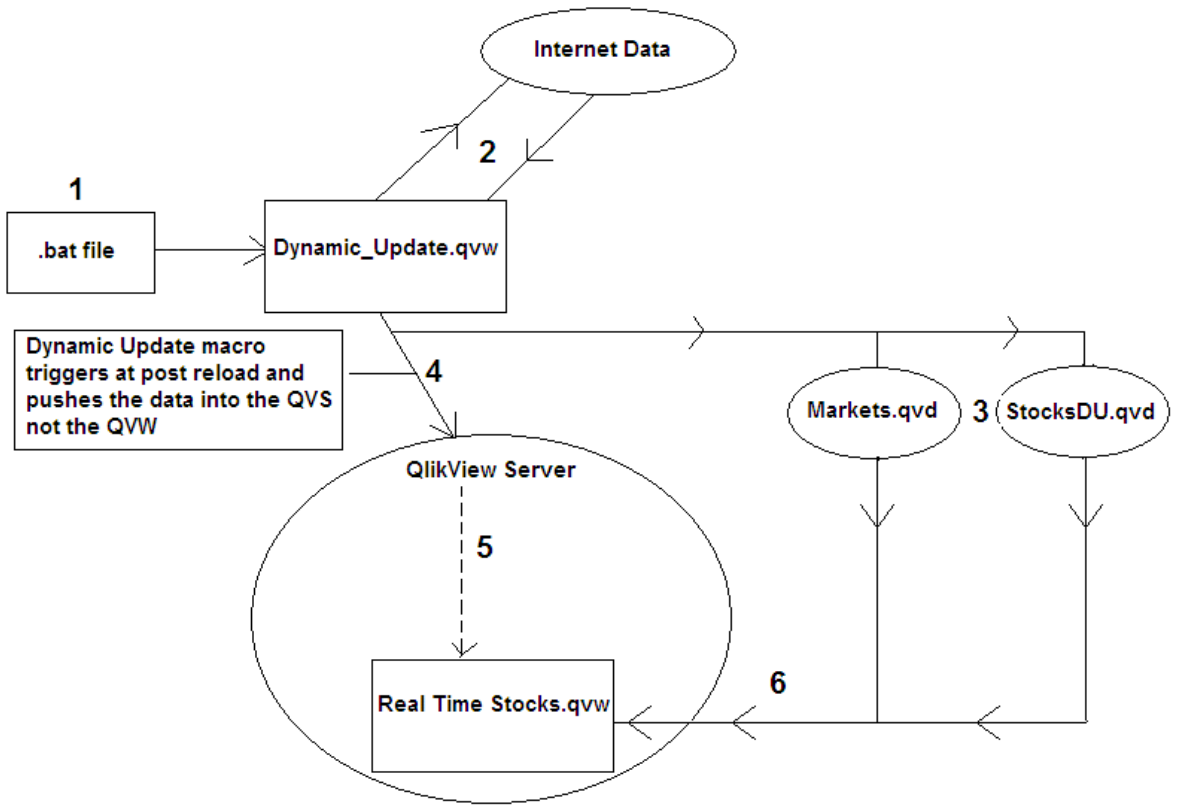
2. The Dynamic_Update.qvw scrapes MSN money to bring back both individual stock information and market information.

3. The Dynamic_Update.qvw writes the data out to both Markets.qvd and StocksDU.qvd for historical purposes.

4. After the reload of the Dynamic_Update.qvw is complete, a macro is triggered that pushes the stock data to the QVS and it is reflected in Real Time Stocks.qvw. ****Note** – The dynamically updated data sits in the server memory and it **IS NOT** stored in the qvw. If the QVS is interrupted in any way, the dynamically updated data is lost and the qvw will revert back to the data from the last application reload.

5. The dynamically updated data is available for display within Real Time Stocks.qvw.

6. The Real Time Stocks.qvw should be reloaded at least every 5 minutes to help minimize the dynamic data loss should the server be interrupted.



Dynamic Update Macro

Creation of the Insert statement

We will use variables to create the Insert statement for the dynamic update. The script below uses the peek function to loop through the Stocks table and pulls out the data for each of the predefined stocks. The first time through the variable myValues is created. Each additional time through the loop the data is appended to the preceding data until all the data has been appended to the myValues variable. After the myValues variable is created, the mySQL variable is created. The mySQL variable creates the insert statement that pushes the dynamic data into the Real Time Stocks.qvw. This process is repeated in the mySQL2 variable to create the insert statement for the Market data.

```
SET mySQL = '';

For r = 0 to NoOfRows('Stocks')-1

LET vField1 = chr(39) & peek('Ticker Symbol',$(r), 'Stocks') & chr(39);
LET vField2 = chr(39) & peek('Date', $(r), 'Stocks') & chr(39);
LET vField3 = chr(39) & peek('Daily_Date', $(r), 'Stocks') & chr(39);
LET vField38 = chr(39) & peek('Day Time', $(r), 'Stocks') & chr(39);
LET vField4 = chr(39) & peek('Hour', $(r), 'Stocks') & chr(39);
LET vField5 = chr(39) & peek('Minute', $(r), 'Stocks') & chr(39);
LET vField6 = chr(39) & peek('Quote', $(r), 'Stocks') & chr(39);
LET vField7 = chr(39) & peek('Current Stock Price', $(r), 'Stocks') & chr(39);
LET vField8 = chr(39) & peek('Current Stock Price Change', $(r), 'Stocks') & chr(39);
LET vField9 = chr(39) & peek('% Price Change', $(r), 'Stocks') & chr(39);
LET vField10 = chr(39) & peek('Timestamp', $(r), 'Stocks') & chr(39);
LET vField25 = chr(39) & peek('Time', $(r), 'Stocks') & chr(39);
LET vField26 = chr(39) & peek('Prev Close', $(r), 'Stocks')& chr(39);
LET vField27 = chr(39) & peek('Open', $(r), 'Stocks')& chr(39);
LET vField28 = chr(39) & peek('Day High', $(r), 'Stocks')& chr(39);
LET vField29 = chr(39) & peek('Day Low', $(r), 'Stocks')& chr(39);
LET vField30 = chr(39) & peek('Volume', $(r), 'Stocks')& chr(39);
LET vField31 = chr(39) & peek('Avg Daily Vol (13 wk)', $(r), 'Stocks')& chr(39);
LET vField32 = chr(39) & peek('Bid', $(r), 'Stocks')& chr(39);
LET vField33 = chr(39) & peek('Bid Size', $(r), 'Stocks')& chr(39);
LET vField34 = chr(39) & peek('Ask', $(r), 'Stocks')& chr(39);
LET vField35 = chr(39) & peek('Ask Size', $(r), 'Stocks')& chr(39);
LET vField36 = chr(39) & peek('52 Wk High', $(r), 'Stocks')& chr(39);
LET vField37 = chr(39) & peek('52 Wk Low', $(r), 'Stocks')& chr(39);

IF r = 0 then

LET myValues = '(' & vField1 & ', ' & vField2 & ', ' & vField3 & ', ' & vField38 & ', ' & vField4
& ', ' & vField5 & ', ' & vField6 & ', ' & vField7 & ', ' & vField8 & ', ' & vField9 & ', ' &
vField10 & ', ' & vField25 & ', ' & vField26 & ', ' & vField27 & ', ' & vField28 & ', ' &
vField29 & ', ' & vField30 & ', ' & vField31 &  ', ' & vField32 & ', ' & vField33 & ', ' &
vField34  & ', ' & vField35 & ', ' & vField36 & ', ' & vField37  & ')';

ELSE

LET myValues = myValues & ', (' & vField1 & ', ' & vField2 & ', ' & vField3 & ', ' & vField38 &
', ' & vField4 & ', ' & vField5 & ', ' & vField6 & ', ' & vField7 & ', ' & vField8 & ', ' &
vField9 & ', ' & vField10 & ', ' & vField25 & ', ' & vField26 & ', ' & vField27 & ', ' & vField28
& ', ' & vField29 & ', ' & vField30 & ', ' & vField31 &  ', ' & vField32 & ', ' & vField33 & ', '
& vField34  & ', ' & vField35 & ', ' & vField36 & ', ' & vField37  & ')';

END IF
NEXT

LET mySQL = mySQL & 'INSERT INTO * ([Ticker Symbol],Date,Daily_Date,[Day
Time],Hour,Minute,Quote,[Current Stock Price],[Current Stock Price Change],[% Price
Change],Timestamp,Time,[Prev Close],Open,[Day High],[Day Low],Volume,[Avg Daily Vol (13
wk)],Bid,[Bid Size],Ask,[Ask Size],[52 Wk High],[52 Wk Low]) VALUES ' & myValues ;
SET mySQL2 = '';
```

```
For s = 0 to NoOfRows('Markets')-1

LET vField11 = chr(39) & peek('Date',$(s), 'Markets') & chr(39);
LET vField12 = chr(39) & peek('Market Name',$(s), 'Markets') & chr(39);
LET vField14 = chr(39) & peek('Market Volume', $(s), 'Markets') & chr(39);
LET vField15 = chr(39) & peek('Change', $(s), 'Markets') & chr(39);
LET vField16 = chr(39) & peek('Change %', $(s), 'Markets') & chr(39);

IF s = 0 then

LET myValues2 = '(' & vField11 & ', ' & vField12 & ', ' & vField14 & ', ' & vField15 & ', ' &
vField16 & ')';

ELSE

LET myValues2 = myValues2 & ', (' & vField11 & ', ' & vField12 & ', ' & vField14 & ', ' &
vField15 & ', ' & vField16 & ')';

END IF

NEXT

LET mySQL2 = mySQL2 & 'INSERT INTO * (Date,[Market Name],[Market Volume],[Change],[Change %])
VALUES ' & myValues2 ;
```

Creation of the Insert Macro

The Insert macro takes the mySQL and mySQL2 variables created in the script and performs the dynamic update function to push the data into Real Time Stocks.qvw.

```
sub Insert
    set vmySQL = ActiveDocument.Variables("mySQL")
    set vmySQL2 = ActiveDocument.Variables("mySQL2")

   'msgbox(vmySQL.GetContent.String)
   'msgbox(vmySQL2.GetContent.String)

   set App = ActiveDocument.GetApplication
   set newdoc = App.OpenDoc("qvp://localhost/Stocks/Real Time Stocks.qvw","","")

   Rem ** back to ActiveDocument (= document running macro) **

   SET Result = newdoc.DynamicUpdateCommand (vmySQL.GetContent.String)

   rem ** let QV sleep for .7 seconds **
   ActiveDocument.GetApplication.Sleep 700

   SET Result = newdoc.DynamicUpdateCommand (vmySQL2.GetContent.String)

   rem ** let QV sleep for .7 seconds **
    ActiveDocument.GetApplication.Sleep 700

   if Result = false then
   MsgBox Result.ErrorMessage
   end if

   Rem ** close new document again **
end sub
```

Tips and Tricks

1. When creating the macro for the dynamic update, be sure to add in the sleep command to allow the process to update correctly. Without the sleep command, the dynamic update command tends to misfire and it doesn't update the data correctly. This leaves null where the dynamically updated data should be.

```
sub Insert

    set vmySQL = ActiveDocument.Variables("mySQL")
    set vmySQL2 = ActiveDocument.Variables("mySQL2")

    'msgbox(vmySQL.GetContent.String)
    'msgbox(vmySQL2.GetContent.String)


set App = ActiveDocument.GetApplication
set newdoc = App.OpenDoc("qvp://localhost/Stocks/Real Time Stocks.qvw","","")
Rem ** back to ActiveDocument (= document running macro) **
ActiveDocument.Activate
  SET Result = newdoc.DynamicUpdateCommand (vmySQL.GetContent.String)
   rem ** let QV sleep for .7 seconds **
     ActiveDocument.GetApplication.Sleep 700

  SET Result = newdoc.DynamicUpdateCommand (vmySQL2.GetContent.String)
   rem ** let QV sleep for .7 seconds **
     ActiveDocument.GetApplication.Sleep 700
if Result = false then
   MsgBox Result.ErrorMessage
 end if
Rem ** close new document again **
    newdoc.CloseDoc

end sub
```

2. One thing to note When using dynamic update is that the data that is pushed to the application is stored in memory and therefore if the QlikView Server (QVS) needs to be restarted or there is any other interruption to the QVS, the data that is in memory will be lost and the application will revert back to show only the data that was in the qvw from the last reload.

Summary

This document outlined the set-up and the process flow for a dynamic update application. Topics discussed included real time vs. reload, software used, files used and workflow as well as other set-up tips and tricks. As each situation is different it is recommended that you use these techniques as a reference for your situation. Depending on the desired outcome, you will need to tweak the processes outlined in the document.