QlikView

WorkBench

Copyright © 1994-2010 Qlik®Tech International AB, Sweden.

Under international copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written permission of QlikTech International AB, except in the manner described in the software agreement.

Qlik®, QlikTech® and QlikView® are trademarks and registered trademarks of QlikTech International AB.

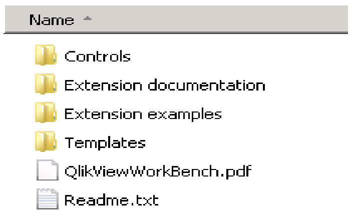
Other trademarks are the property of their respective owners and are hereby acknowledged.

Content

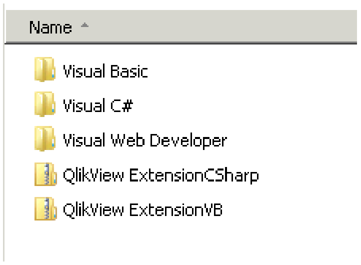
1 Installing QlikView WorkBench

1. Start the installation program, `QlikViewWorkBench_x64Setup.exe or QlikViewWorkBench_x86Setup.exe`.
2. The installation unpacks the files and computes the space needed for the installation. A welcome screen is then displayed. Click **Next** to continue.
3. You now come to the software license agreement. Read it, and click **I accept the terms of the license agreement**, then click **Next**.
4. In the **Customer Information** dialog you specify for whom the installation will be personalized. Click **Next** to continue.
5. The next dialog shows the default installation path of the program. Click **Change** to enter an alternate path for the installation. Then click **Next**.
6. In the **Default Values** dialog, you enter the path to the **`QvAjaxZfc`** virtual directory on you web server, either QlikView Web Server or Microsoft IIS. Click **Test url** to confirm the path to the directory.
7. You are now ready to install the program. Click **Next** to install the files.
8. After the installation is done, click **Finish** to complete the process.

The following folders are now installed in either the default installation path, `C:\Program Files\QlikView\WorkBench`, or the path you choose during the installation.


Name
Controls
Extension documentation
Extension examples
Templates
QlikViewWorkBench.pdf
Readme.txt

The templates that come with QlikView WorkBench are installed to C:\Program Files\QlikView\WorkBench. They are also installed in Visual Studio and available when creating a new web site.


Name
Visual Basic
Visual C#
Visual Web Developer
QlikView ExtensionCSharp
QlikView ExtensionVB

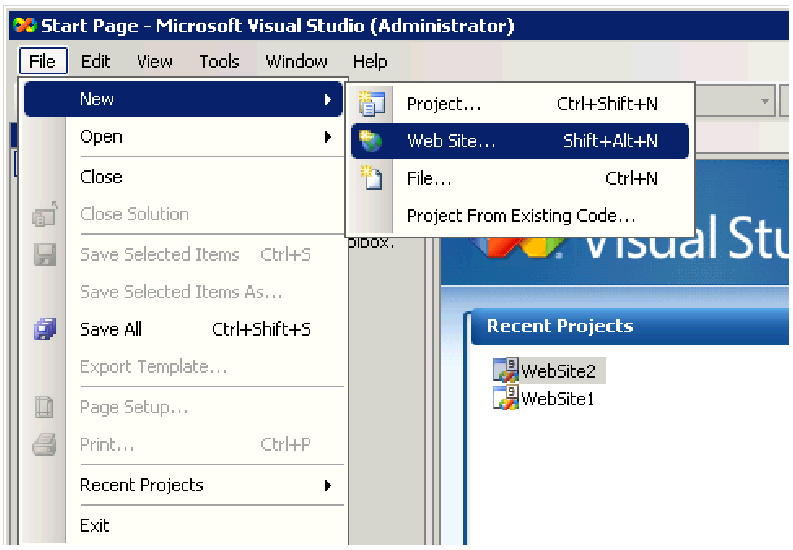
Using Proxy

If you do not have your AccessPoint (the QvAjaxZfx virtual directory) on the same computer as your QlikView WorkBench site, you must use a proxy to avoid cross-site scripting issues. You can use Proxy.aspx for asp.net sites. If you use the **QlikView WorkBench** template when you create your web site, the proxy is automatically used. You can also create your own proxy . When you create your own proxy the following conditions must be met:

- Server request are coming ("escaped") in client request query string "u"
- Cookies are copied from client request to server request
- Headers are copied from server response to client response
- Server response is copied in binary form to client response.

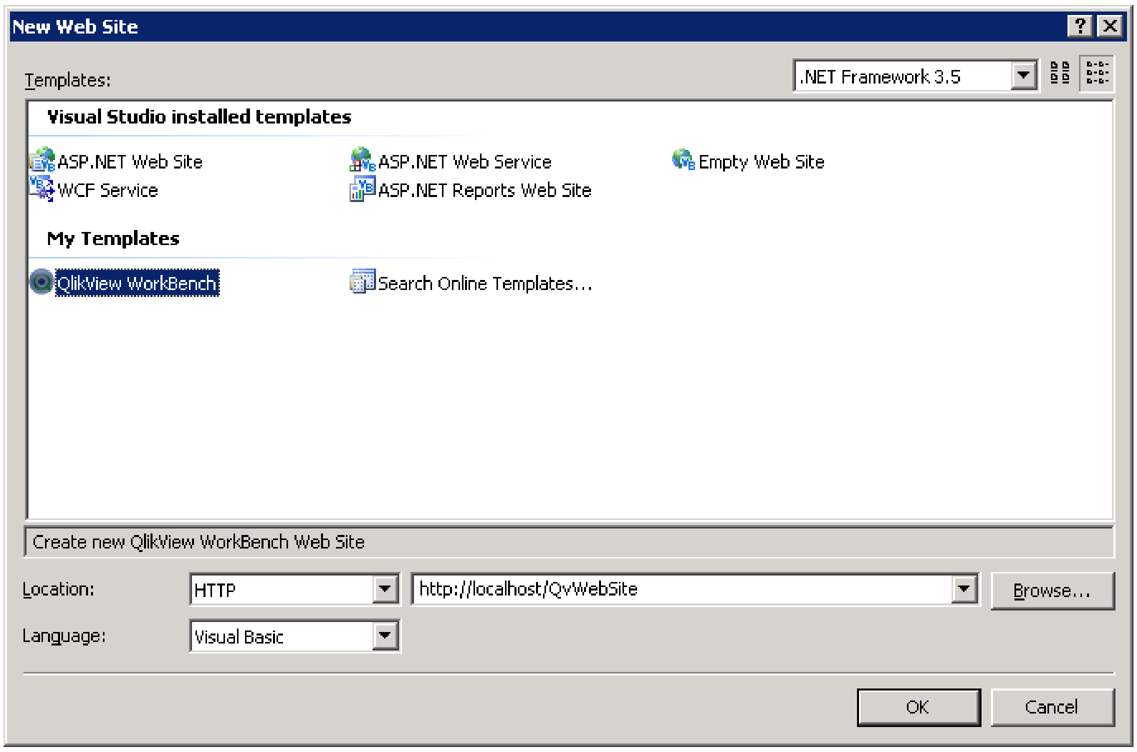
2 Creating a Web Site

Open **Microsoft Visual Studio** and choose **File**, then **New** and **Web Site** to create a new empty web site.



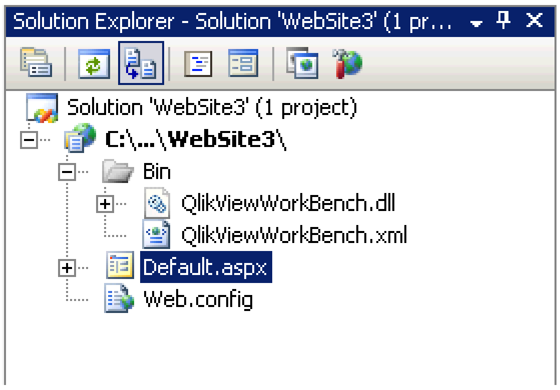
You will be presented with the **New Web Site** dialog, where you choose the **Language** (C#, Visual Basic), and the **Location** for your web site. There is a QlikView WorkBench template included in the installation program. This template contains the Proxy.aspx page and settings in web.config, which makes it possible to run the web site on any computer, no matter where the QvAjaxZfc is located. Click **OK** to create it.

Note! The **Location** must be **http** when using Microsoft IIS 7 or newer! If you use the WorkBench template you may use File System as **Location**.



The web site has now been created and you can see it in the **Solution Explorer** (if you do not see it, you can bring it up from the **View** menu).

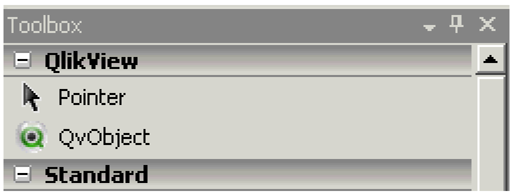
The **Solution Explorer** shows the structure of the web site. The items `QlikViewWorkBench.dll` and `QlikViewWorkBench.xml` are not displayed until the QvObject has been added on a page in the solution.



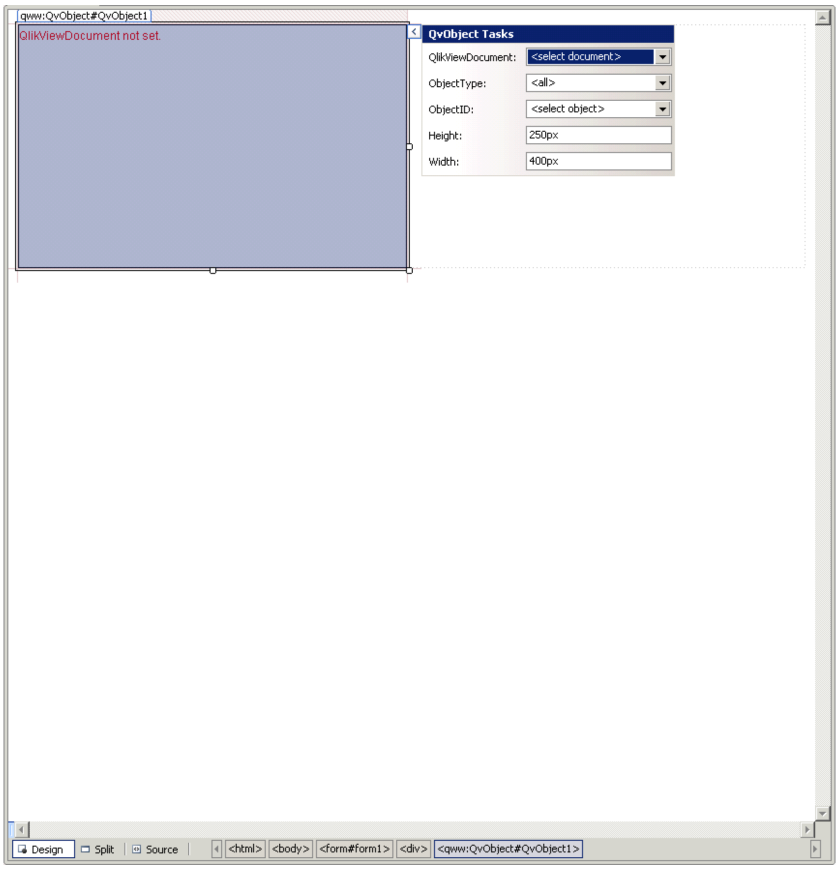
3 Building a Web Page

Bring up the `default.aspx` web page in **Design Mode**. Open the **Toolbox** (if you cannot see it, you can bring it up from the **View** menu).

In the **Toolbox** you now see the **QlikView** area.



Select the **QvObject**. Drag the control to the web page. When you do this, you will see the object and a **Smart Tag**. A **Smart Tag** is a feature of Visual Studio which presents the most common or essential properties that you need to set on the control. To bring up the **Smart Tag** at any time, click on the arrow tab on the top right corner of the control.


qww:QvObject#QvObject1

QlikViewDocument not set.

QvObject Tasks

QlikViewDocument:	<select document>
ObjectType:	<all>
ObjectID:	<select object>
Height:	250px
Width:	400px

The **Smart Tag** contains the following settings:

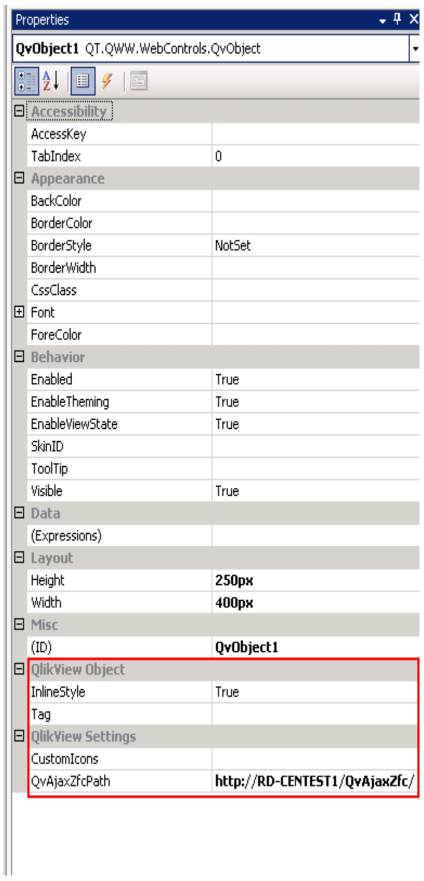
QlikViewDocument	Select a document present on the QlikView Server to connect to.

Object Type	Select an object type from the QlikView document. This will filter the list for **Object ID**.
Object ID	Select an object from the QlikView document.
Height	Set the height of the object on the web page.
Width	Set the width of the object on the web page.

Note! If you cannot see any documents in the **QlikViewDocument** setting, you may have the wrong path in the setting **QvAjaxZfcPath**. See below.

Property Viewer

The object also has a properties view. Right-click on the QvObject to bring up the **Properties** pane. Only the properties under **QlikView Object** and **QlikView Settings** are relevant to the QlikView Object.



Properties

QvObject1 QT.QWW.WebControls.QvObject

Accessibility	
AccessKey	
TabIndex	0
Appearance	
BackColor	
BorderColor	
BorderStyle	NotSet
BorderWidth	
CssClass	
Font	
ForeColor	
Behavior	
Enabled	True
EnableTheming	True
EnableViewState	True
SkinID	
ToolTip	
Visible	True
Data	
(Expressions)	
Layout	
Height	250px
Width	400px
Misc	
(ID)	QvObject1
QlikView Object	
InlineStyle	True
Tag	
QlikView Settings	
CustomIcons	
QvAjaxZfcPath	http://RD-CENTEST1/QvAjaxZfc/

QlikView Object

InlineStyle

Some styles are set using a stylesheet created by the QlikView Server’s Ajax engine. These styles can be overridden in your own custom stylesheet. However, other styles are added to the inline HTML generated by the QlikView Server Ajax engine. These styles cannot normally be overridden in your own custom stlyesheet. However, changing this setting from the default **True** to **False** will allow these inline styles to be overridden in your custom stylesheet. The styles that are provided inline by the QlikView Server Ajax engine are styles regarding fonts, borders and colors. Read more about the inline styles in See "Inline Styles" on page 17

Tag

Define you own tag for the object that can be used in, for instance JavaScript. This can be used for customizing a QvObject, marking a QvObject for special action, distinguishing between one QvObject and another at run time, etc. This information can be used to better integrate your QvObject. Note: The tag is added as an attribute of the QvObject's `div` tag. Here is an example of a function that retries an array of tag attributes from all the QvObject on a webpage:

```
<script type="text/javascript">
        GetAllQvObjectsByTag = function(tag) {
                var tagObjs = [];
                var divs = document.getElementsByTagName("div");
                for (var i = 0; i < divs.length; i++) {
                        if (divs[i].getAttribute("Tag") == tag) {
                                tagObjs.push(divs[i]);
                        }
                }
                return tagObjs;
        }
</script>
```

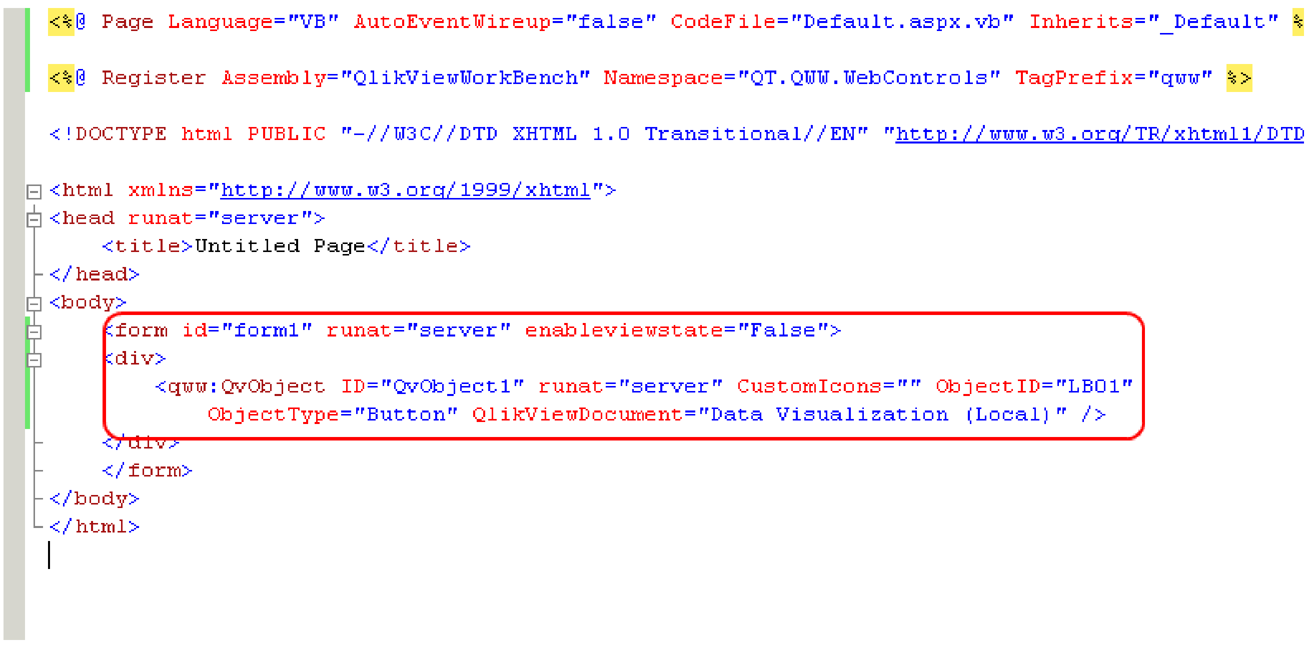
QlikView Settings

CustomIcons	Enter substitute images for caption icons. The syntax is `icon code:icon url`. The following icons can be replace by custom icons. Custom icons must use relative paths. Separate icons with comma if more than one. The icon code precede the name of the icon. LS - Lock Excluded US - Unlock Selected CA - Clear All CA.Disabled - Clear All Disabled SE - Select Excluded SP - Select Possible SA - Select All SEARCH - Search XL - Send to Excel CD - Clear PR - Print HELP - Help
QvAjaxZfcPath	Enter the path to your QlikView Web Server and the Ajax folder, e.g. `/QvAjaxZfc/` for using a local web server. To use a remote server enter the http address, e.g. `http://Qv-WebServer/QvAjaxZfx`/.

The properties defined in the **QlikView Settings** group affect all QlikView objects on the page.

The Source View

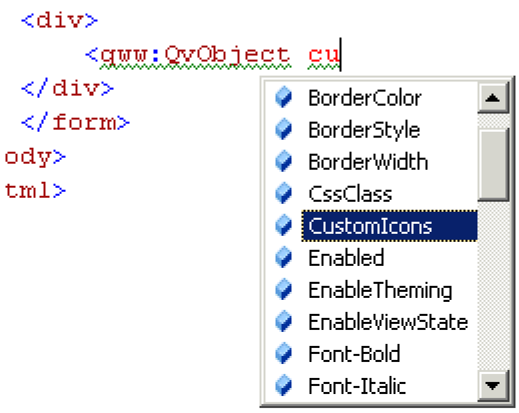
A web page can be viewed in **Design** mode and in **Source** mode. The latter shows the actual code of the web page.



```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb" Inherits="_Default" %

<%@ Register Assembly="QlikViewWorkBench" Namespace="QT.QWW.WebControls" TagPrefix="qww" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server" enableviewstate="False">
    <div>
        <qww:QvObject ID="QvObject1" runat="server" CustomIcons="" ObjectID="LB01"
            ObjectType="Button" QlikViewDocument="Data Visualization (Local)" />
    </div>
    </form>
</body>
</html>
```

The code in the highlighted area is the code created by inserting the QvObject on the page.

It is possible to type in the properties you want here. Thanks to full integration with Visual Studio, you have the advantage of IntelliSense, that is, Visual Studio shows you all of the available properties and methods that the control has.



```
 <div>
     <qww:QvObject cu
 </div>
 </form>
ody>
tml>
```

Customizations

If you have your own Java Script functions you want to run after the QlikView object has been rendered, you can add functions in the array "`Qva.BodyOnLoadFunctionNames`".

To run the function MyInit, for example, you would add the following line to the Java Script of the page:
`Qva.BodyOnLoadFunctionNames.push('MyInit');`

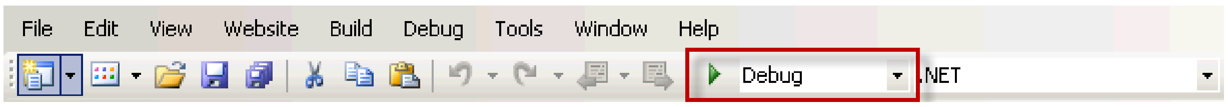
If you want to add your own function that is run when a specific QlikView object is changed, you can use the function "`GetQvObject`". In the example below the number of values that are Enabled when a change is made. "qva" below refers to the Document manager that is created on the generated page when you have added at least one QvObject from the Toolbox. If you have objects from more than one document, the variables for the document managers will be "qva2", "qva3" etcetera.

```
<script type="text/javascript">

      var MyQvObject;

      MyListIsUpdated = function() {

            alert("Number of enabled values: " +
             MyQvObject.QvaPublic.Data.GetEnabled().length);

      }

      MyInit = function() {

            MyQvObject = qva.GetQvObject("LB1457", MyListIsUpdated);

      }

      Qva.BodyOnLoadFunctionNames.push('MyInit');

</script>
```

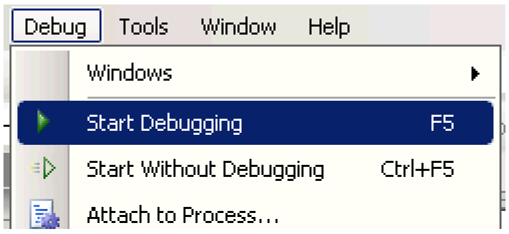
Instead of using generated qva-variables, you can use the function `Qva.GetBinder("Document name")`. The entry `MyQvObject = qva.GetQvObject("LB1457", MyListIsUpdated);` should then look like this:
`MyQvObject = Qva.GetBinder("Films").GetQvObject("LB1457", MyListIsUpdated);`

4 Running the Page in Visual Studio

Now that an object is on the page we can run the web site from Visual Studio. You can do this in a number of ways. You can press F5, click the **Run** button on the toolbar



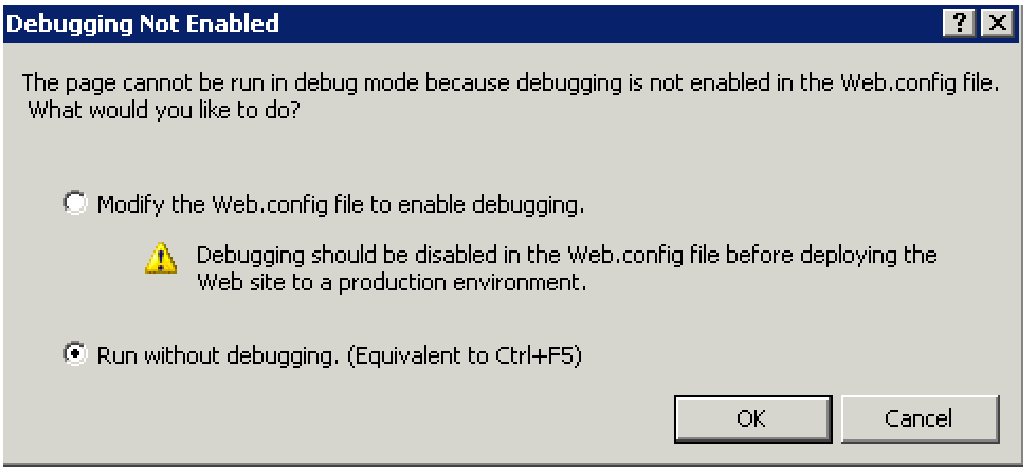
or from the **Debug** menu



Visual Studio will prompt you with a message concerning debug information. Choose the first alternative to have Visual Studio edit your `web.config` file and you will not receive this message again. To re-enable the message, set `compilation debug` to `false` in `web.config`:

```
<compilation debug="false" strict="false" explicit="true">
```

If you do not want to edit the `web.config` file, you should choose **Run without debugging**.



Debugging Not Enabled

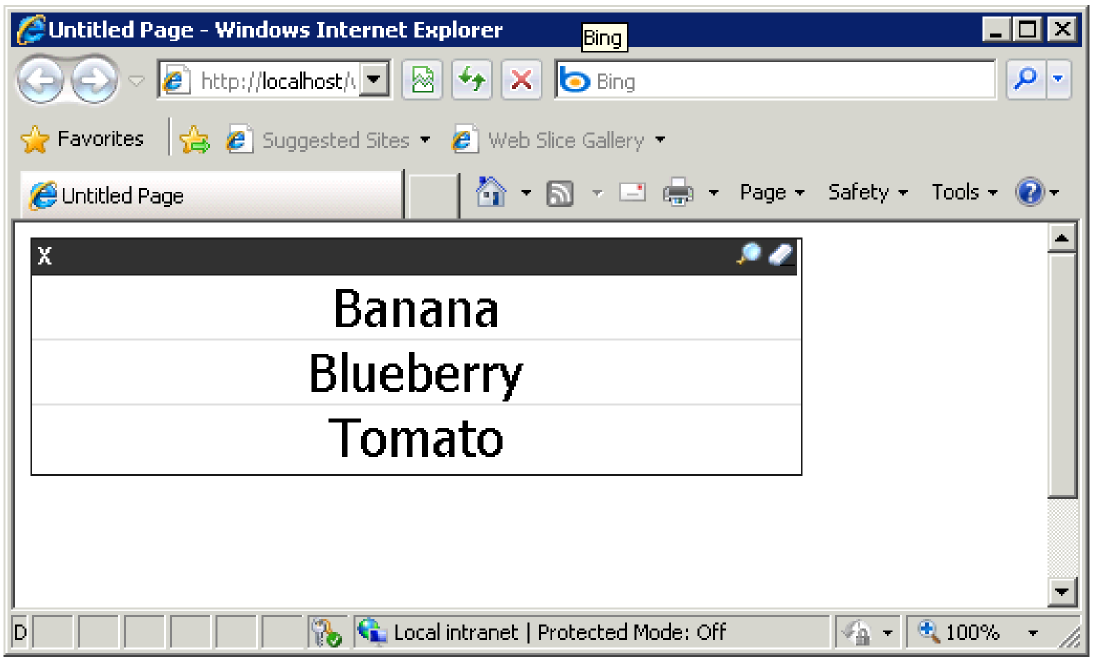
The page cannot be run in debug mode because debugging is not enabled in the Web.config file. What would you like to do?

[ ] Modify the Web.config file to enable debugging.

Debugging should be disabled in the Web.config file before deploying the Web site to a production environment.

[✓] Run without debugging. (Equivalent to Ctrl+F5)

Your web site is now displayed in you Internet browser.



Banana
Blueberry
Tomato

5 Appendix A

Logging

All errors are per default logged in the event log. If you wish to expand the logging for debugging purposes, you can add the section "`LogFile`" in web.config for the site. Create the "`QlikViewWorkBech`" group, if it does not already exist (the sections exist if you chose the **QlikView WorkBench** template when you created you web site):

```
<configuration>

      <configSections>

            <sectionGroup name="QlikViewWorkBench">

                  <section name="General" type=
                  "System.Configuration.NameValueSectionHandler"
                  requirePermission="false"/>

            </sectionGroup>

      </configSections>
```

Then you add the section itself. Example:

```
<configuration>

      <configSections>

            <sectionGroup name="QlikViewWorkBench">

                  <section name="General" type=
                  "System.Configuration.NameValueSectionHandler"
                  requirePermission="false"/>

            </sectionGroup>

      </configSections>

      <QlikViewWorkBench>

            <General>

                  <add key="Proxy" value="/Proxy.aspx"/>

                  <add key="LogFile" value="/wblog.txt"/>

            </General>

      </QlikViewWorkBench>
```

You can also log what you do in Visual Studio. To initiate the logging you must set the "`Log-File`" entry in the Registry, `HKEY_LOCAL_MACHINE\SOF-TWARE\QlikTech\QlikViewWorkBench\`. On a 64-bit computer the path is `HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\QlikTech\QlikViewWorkBench`.

Inline Styles

Some styles are set using a stylesheet created by the QlikView Server's Ajax engine. These styles can be overridden in your own custom stylesheet. However other styles are added to the inline HTML generated by the QlikView Server Ajax engine. These styles cannot normally be overridden in your own custom stlyesheet. However, changing this setting from the default True to False will allow these inline styles to be overridden in your custom stylesheet.

The styles that are provided inline by the QlikView Server Ajax engine are:

fontfamily, fontsize, fontstyle, fontweight, textalign, verticalalign, textdecoration, paddingTop, paddingLeft, paddingRight, paddingBottom, color

background-color, color, text-align, font-style, font-weight, text-decoration, font-size, border-bottom, border-top, border-left, border-right

MozBorderRadiusTopleft, MozBorderRadiusTopright, MozBorderRadiusBottomleft, MozBorderRadiusBottomright

WebkitBorderTopLeftRadius, WebkitBorderTopRightRadius, WebkitBorderBottomLeftRadius, WebkitBorderBottomRightRadius