

Overview vxl-1

- Program Requirements vxl-1
- Installation of ViewXL Add-in vxl-1
- Authorization Code vxl-1

Menu vxl-2

Toolbar vxl-3

Using the ViewXL Program API vxl-8

An Example Macro vxl-9

ViewXL API Definition vxl-11

Overview

ViewXL is a Microsoft Excel Add-In that provides setup and data acquisition for personal computers running 32-bit versions of Microsoft Windows. The features of Excel and the selected data acquisition server, for example, ChartView, combine seamlessly to form a powerful data acquisition tool.

ViewXL:

- Augments Microsoft Excel with data acquisition capability.
- Automatically converts data to engineering units.
- Allows multiple data acquisitions [obtained using auto-rearm] to be placed into multiple worksheets.
- Can import data from previously acquired data files.

Once the ViewXL add-in is installed, a ViewXL toolbar and related menu appear in Excel. These provide all configuration and data acquisition controls.

With ViewXL, you can obtain automated reports by taking advantage of the Software Components concept. For example, you can embed an Excel spreadsheet object into a Word document. When the spreadsheet object is activated, it gives you access to ViewXL. In this case, the Word document holds all text, as well as an embedded Excel worksheet object. The Excel worksheet object holds raw and calculated data and graphics.

Double-clicking on the Excel worksheet object in Word can access ViewXL, allowing the user to efficiently prepare a statistical report. This compound document allows you to collect, configure, analyze, graph, and annotate data, with a just a few clicks of a mouse button.

Program Requirements

ViewXL requires the following software:

- A data acquisition server application, such as 32-bit ChartView, that can implement the View-App Interface standard.
- Microsoft Excel 2000 or later
- 32 bit Microsoft Windows Operating System (Windows 9x, ME, NT, 2000, XP, etc.)

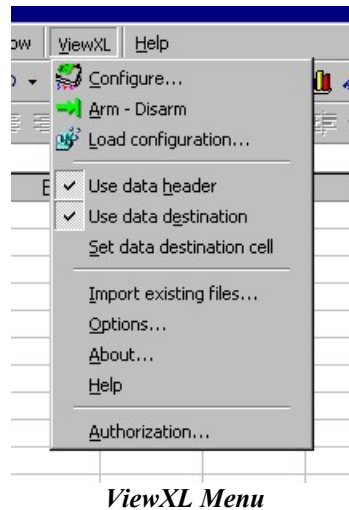
Installation of ViewXL Add-in

ViewXL is a standard Microsoft Excel Add-in. It is installed by selecting the **Excel Tools/Add-Ins** menu item. If ViewXL does not appear in the list of available add-ins, press the Browse... button, and then select the **ViewXL.xla** file. When the Add-In appears in the list, click the check-box next to ViewXL to enable the program.

Authorization Code

ViewXL is a purchased product that is enabled by entering an appropriate [authorization code](#) via the Authorization dialog (accessed via the pull-down menu). ViewXL can also be enabled for a 30-day trial period by pressing the “Start 30 Day Trial” button in the same dialog. See page vxl-3 for details.

Menu



The ViewXL menu gives you access to all of the Add-Ins features. The first six menu items are also available from the ViewXL Toolbar.



Reference Note:

Refer to the [Toolbar](#) section, beginning on page vxl-3, for a description of the functions that can be accessed by both the Toolbar and the pull-down Menu.

Import Existing Files

Brings up the Import Files dialog box. Normally ViewXL gets data from an acquisition performed by the connected data acquisition server application (e.g., ChartView). This is an alternate way of selecting data files for viewing in the Excel workbook.



Reference Note:

For detailed information, refer to page vxl-6, [Import Files](#).

Options

Brings up the Options dialog box. The ViewXL options settings are made from the Options dialog box. From there you can select, among other options, the desired data acquisition application.



Reference Note:

For detailed information, refer to page vxl-4, [Options](#).

About

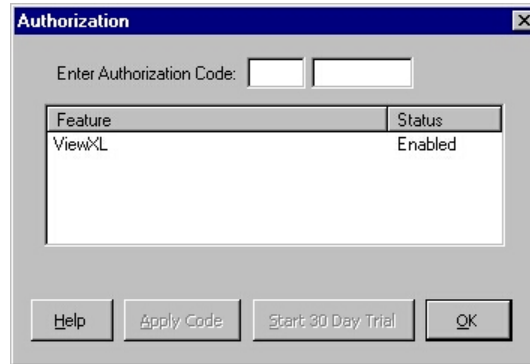
Brings up the About dialog box. This displays copyright and version information about the Add-In.

Help

Displays the Help File.

Authorization

Brings up the Authorization dialog box. From there you can enter product authorization codes that enable specific features of the Add-In. You can also choose to evaluate the Add-In at no cost for up to 30 days.



Authorization Dialog Box

ViewXL is a purchased Excel Add-In program. You may examine the program at no cost for up to 30 days. After that, you must purchase a license to use the software. The Authorization dialog box allows you to either start a 30-day trial period, or enter the authorization code that you received if you purchased a license.

Toolbar

The ViewXL toolbar provides access to the Add-Ins most frequently used features. These features, and a few others, are also available from the ViewXL Menu. The toolbar contains six buttons separated into two groups. From left to right these are:



The first three buttons are used to control the data acquisition application [also referred to as “server.”] The remaining three are used to control the way data is displayed in the worksheets. The toolbar can be left floating over the worksheet, as shown in the above figure, or docked along with the other Excel toolbars. To dock the toolbar, simply drag it to the toolbar area of the Excel main window.

A description of each button function follows.



Configure

This button is used to launch the data acquisition application, such as ChartView. If a data acquisition application [server] has not been selected, a message will be displayed to that effect. To choose a data acquisition server, select the ViewXL/Options menu item, then select an application from the list of available servers.



Arm / Disarm

This push-push button is used to arm or disarm the acquisition. If the acquisition application has been set for manual trigger, pushing this button will initiate the trigger. When pushed, the Arm button remains "depressed" until the acquisition finishes. Pushing the button again, during the acquisition, halts it.



Load Configuration

When pushed, the Load Configuration dialog box appears with a list of configuration files that are compatible with the running data acquisition application [server]. To load a new configuration, select one of the files in the list, then press the "Load File" button.



Display Header

This push-push button instructs ViewXL to display [or hide] Header Information in the worksheet along with the acquisition data. Header Information includes Channel Labels, Channel Units, and in some cases, the Source data file.



Use Data Destination

This push-push button instructs ViewXL to start loading [or stop loading] acquisition data and header information into the worksheet, starting from the data destination cell that is the upper left corner where the data will be placed.

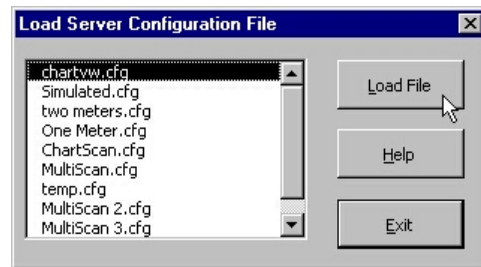


Set Data Destination Cell

Pushing this button saves the position, of the *currently selected cell*, as the data destination location where acquisition header and data information will be saved.

Load Configuration

The Load Configuration dialog box presents a list of configuration files that are compatible with the currently running data acquisition application (such as ChartView).



Load Server Configuration File Dialog Box

Options

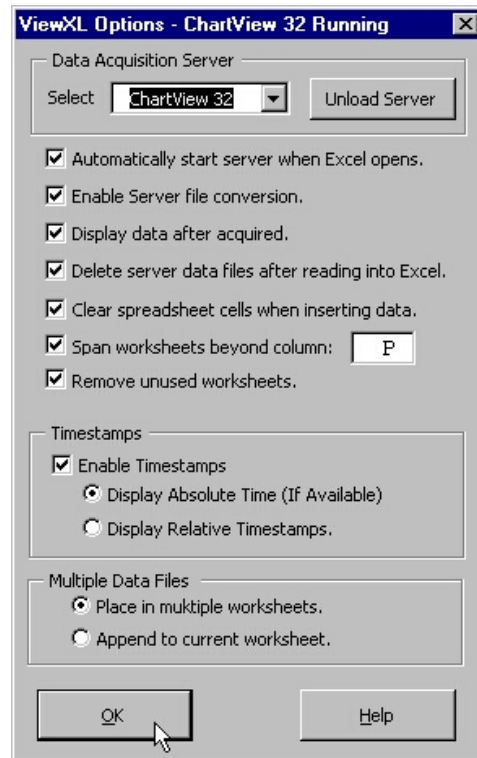
The Options dialog allows you select a data acquisition server program, and choose how the Add-In performs specific operations.

Data Acquisition Server

If a server program, such as ChartView, is running you can stop it by selecting the application from the Select pull-down list, and pressing the <Unload Server> button. After selecting a new server program from the list, press the **OK** button to enable the selection.

Before the ViewXL Add-In can control a data acquisition application [server], you must select the desired application from the list. The selected server name will be saved when Excel is closed, and restored the next time Excel is started. This is the default server that will be started when the Configure Toolbar button is pressed, or Menu item is selected.

- Automatically start server when Excel opens.**
When checked, this informs the Add-In that the default data acquisition application [server] should be automatically started when Excel is started. The Configure button does not have to be pressed to start the server.



ViewXL Options Dialog Box

- Enable Server file conversion.**
This option enables the post acquisition file conversion for the specified format type, such as ASCII, and DADISP. If the option is not checked, then the post acquisition file conversion process is disabled.
- Display data after acquired.**
If checked, the data acquired by the data acquisition application, such as ChartView, is automatically read and displayed in Excel. Checked is the default setting for this option. Un-check this option if you want to acquire data without automatically loading it into Excel.
- Delete server data files after reading into Excel.**
Check this option if you want the Add-In to automatically delete the server data files after they have been read and displayed in Excel.
- Clear spreadsheet cells when inserting data.**
Check this option if you want ViewXL to automatically clear the worksheet before displaying new data.
- Span worksheets beyond column:**
Check this option if you want to control the number of columns of data that can be viewed in a single worksheet.

Example:
You have an acquisition that contains 24 channels of data. Check the “**Span worksheets beyond column**” option and then enter either '16', or the letter 'P' into the column entry field. ViewXL will clip the display at column P, then create an additional worksheet for the remaining channels.

When Spanning is enabled, the number of channels displayed on a worksheet, and the number of worksheets created, will be determined by factors such as: (1) The number of channels in the acquisition, (2) The column specified, (3) The Data Destination settings, and (4) Whether or not Timestamps are displayed. Scans of data that span multiple worksheets, are placed in the same row on each sheet, along with timestamp information, if enabled.
- Remove unused worksheets.**
Check this option if you want ViewXL to automatically delete any worksheets that are not needed to display the currently Imported data files, or Server acquisition data

Timestamps

- Enable Timestamps**
If checked, the Add-In attempts to read timestamp information from the data file and add it to the worksheet. Timestamps are placed in the first column of the worksheet. All channels are moved to the right one column, to make room for the timestamps. If timestamps are not supported, they are not displayed in the worksheet. You can check if timestamps are supported in a particular file by selecting the file in the "Import Files" dialog, and checking the Preview Info button. With timestamps enabled, you can select the type of timestamp to be displayed.
 - Display Absolute Time (if available).**
If supported, Absolute timestamps are displayed in a Date / Time format. For example "01/24/02 10:33:20 AM". Since these timestamps only have a One Second resolution, they are not practical for data that was acquired at moderate to fast scan rates. **Unless your scan rate is very slow, it is recommended that you use Relative timestamps.**
 - Display Relative Timestamps.**
Relative timestamps show the time, in seconds, of a scan relative to the trigger position. For example, if you have an acquisition that has three pretrigger scans, and the pre and post trigger scan rates were 1000 scans / second, the timestamps for the first four scans would be -0.003, -0.002, -0.001 and 0.000 respectively. This indicates that the first scan was taken, 3 ms prior to the trigger point. The next scan 2 ms, and the next 1 ms. Scan number four was the trigger point so it's relative time is zero.

Multiple Data Files

If an acquisition contains multiple data files acquired using auto-rearm, or multiple files have been selected for Import, this option selects how the files are displayed.

● **Place in multiple worksheets.**

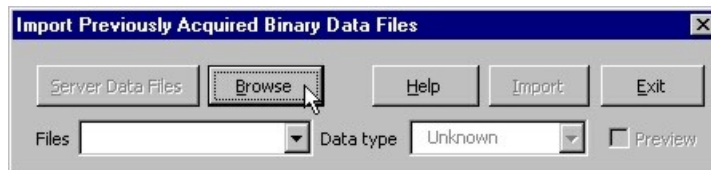
This tells the Add-In that you want multiple files to be placed in multiple worksheets. Excel will create a new worksheet for each file selected, or each file in the acquisition. The sheets will be named with the file name, excluding the extension. Existing worksheets will be reused if the file names are the same. Only the data on the sheet will be updated.

● **Append to current worksheet.**

Places data from all files end to end on the current worksheet. Files are separated by a single blank row. Make sure your version of Windows and Excel is capable of supporting the number of rows of data that will be produced.

Import Files

The Import Previously Acquired Binary Data Files dialog box allows you to open and view data files that were previously acquired by one of the supported data acquisition software programs. These include: DaqView, WaveView, LogView, ChartView Binary, Personal DaqView, DaqExplorer, and data files that were saved by the DaqCOM data acquisition component.



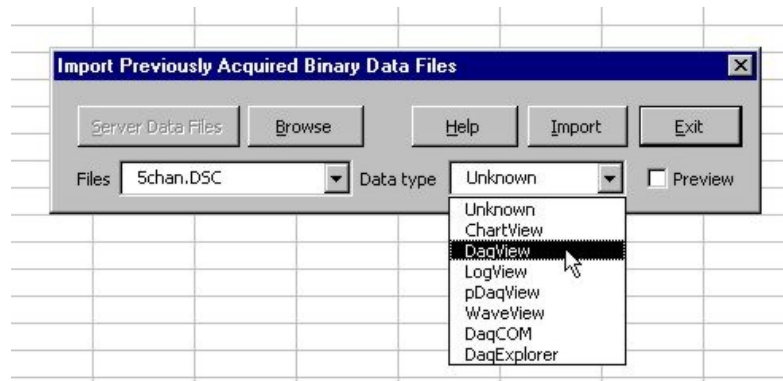
The “Import Previously Acquired Binary Data Files” Dialog Box

As it appears when ViewXL is running independent of a data acquisition application.

When the ViewXL Add-In is running independent of a data acquisition application, then the “Import Previously Acquired Binary Data Files” dialog box opens with an appearance similar to that shown in the above figure. You can use the Browse feature to locate data files that are accessible to your PC. Simply press the “Browse” button, select the desired file, and then press the “Open” button.

Note: The files listed are the *data descriptor files*. Descriptor files contain information about their associated binary data files. At this point, it is the descriptor that you are selecting, not the actual binary data file.

After selecting a compatible descriptor file, select the *data type* of the file from the drop down list (see the following figure).

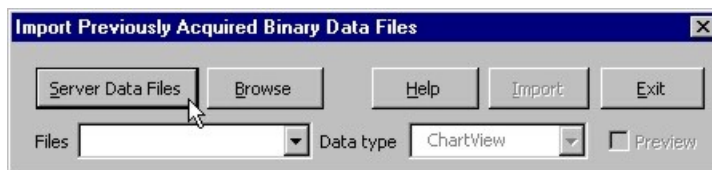


Selecting a Descriptor File and a Compatible Data Type



If you do not know the data file type, you can select *Unknown* from the list. However, there exists a possibility that an *Unknown* file may be identified as the wrong data type, and produce undefined results.

If the ViewXL Add-In is running and is connected to a data acquisition application (for example, ChartView), then the "Server Data Files" button is enabled, as indicated in the following figure. This button brings up the File Open dialog that has been initialized to display the sub-directory when the currently active acquisition application [server] stores its data files. The Data Type selection is disabled and is preset to the data type of the running server.



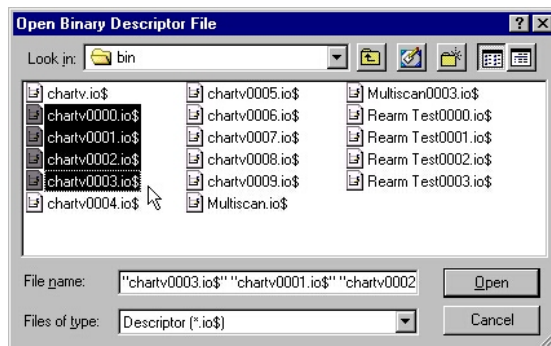
Importing-Server Data Files Dialog Box

Select the desired files from the Open Binary Descriptor File dialog box; then press the "Open" button. See the figure to the right.

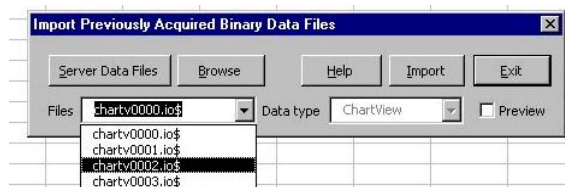
The selected file names will be placed in the drop-down list, in the *Import Previously Acquired Binary Data Files* dialog box (shown below).

When one or more files have been added to the drop down file list (following figure), you can:

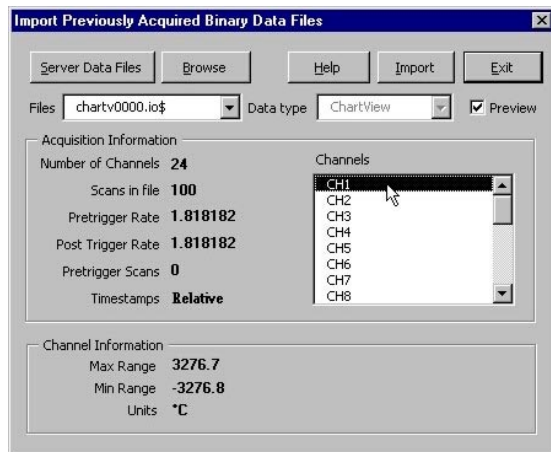
- (a) press the Import button to view the data in an Excel worksheet, or,
- (b) check the "Preview Info" box to view a summary of the highlighted file, as indicated in the *Using the Preview Option ...* figure.



Selecting Binary Descriptor Files



Drop-Down List of Previously Acquired Binary Files



Using the Preview Option to View a File Summary

The "Acquisition Information" panel (see previous figure, right-side, central section) displays a summary of the selected file. Included are: the number of channels in the file, number of data scans in the file, pre- and post-trigger rates in samples per second, the number of pre-trigger scans, and the type of time stamp information supported by the file. Time stamp options are: "None", "Relative", or "Rel + Abs" [for both relative and absolute]. Relative time stamps show the time interval between scans. Absolute time stamps show a Date and Time associated with a scan.

Note: For high-speed acquisitions, the one-second resolution of Absolute time stamps may not provide any meaningful information.

The "Channel Information" panel (see previous figure, right-side, lower section) displays a summary of the channel selected in the channels list. The summary shows the maximum and minimum range limits of the channel data, and the channel engineering Units.

When the "Import" button is pressed, the selected files are opened and read into the Excel workbook. If the option to *place multiple files into multiple worksheets* is **enabled**, then a worksheet is created for each file, and each worksheet is renamed with name of the file [excluding the file extension]. If the option to *place multiple files into multiple worksheets* is **not enabled**, then all of the files are placed on the same worksheet, separated by a blank row.



The number of rows, that an Excel spreadsheet can have, is determined by the version of Microsoft Windows being used. Make sure that you do not exceed the limits, as described in the Excel documentation, for your version of Windows.

41	25.1	25.1	27.3	-3276.7
42	25.1	25	27.4	-3276.7
43	25.1	25.1	27.3	-3276.7

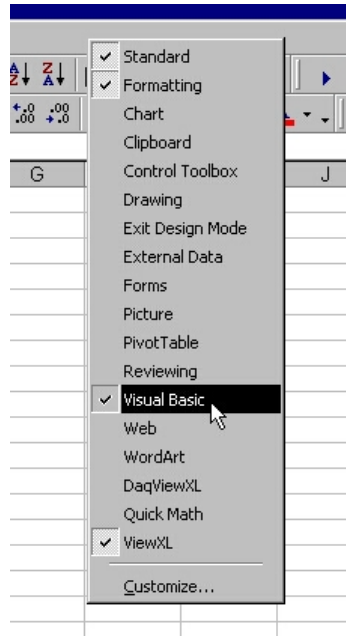
Ready

File Information in Excel Worksheet

Using the ViewXL Program API

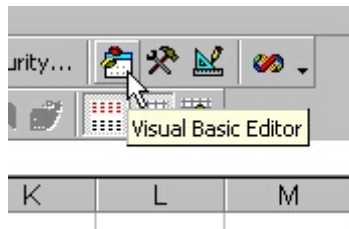
The ViewXL Add-In exposes several methods that you can call from your own Excel Macros. Before you can use the Add-In you must perform the following steps.

1. Make the Excel Visual Basic toolbar visible. This is accomplished by right-clicking in a blank area on the Excel Menu or Toolbar space. When the pop-up menu appears, check the "Visual Basic" selection, as indicated in the following figure.



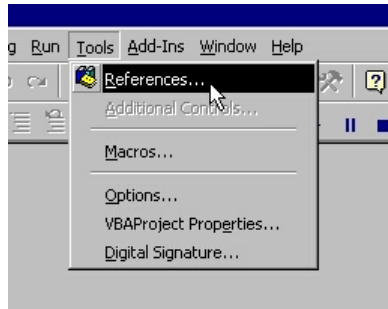
Selecting Visual Basic

2. With the Visual Basic toolbar visible, click the "Visual Basic Editor" button.



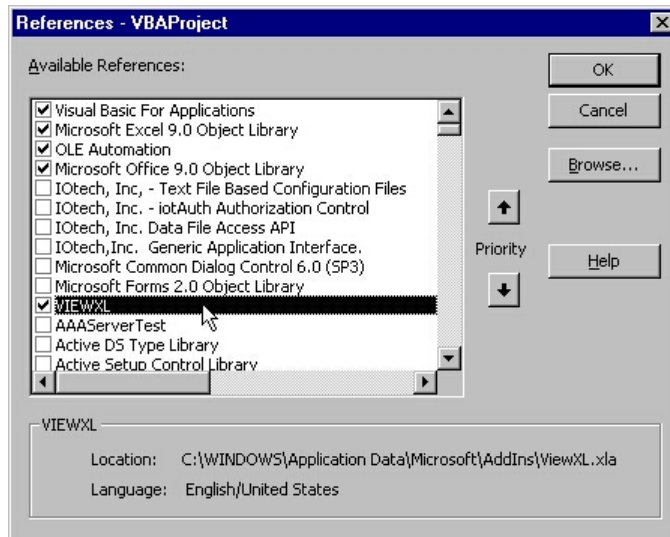
Selecting the Visual Basic Editor

3. Click the "Tools" menu, then select the "References" item.



Selecting References in the Tools Pull-down Menu

4. Select VIEWXL from the list of Available References (following figure). Be sure a check mark is next to the VIEWXL selection, then press the OK button.



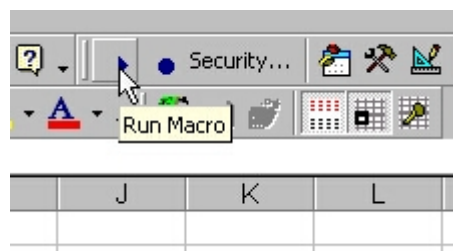
Selecting VIEWXL from Available References

You have now performed the tasks needed to use the ViewXL Add-In from your own Excel Macros.

An Example Macro

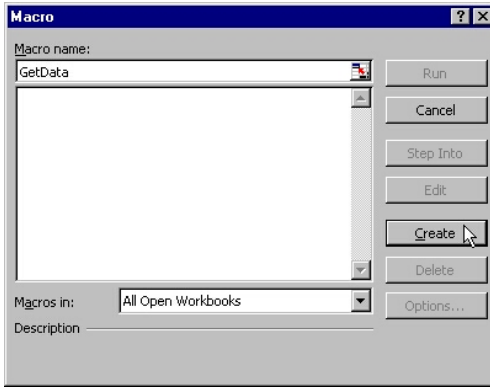
We will now be looking at an example of a macro that uses ViewXL to tell a data acquisition application [server], to Load a Configuration File, and to Arm the acquisition. This macro assumes that the specified file is known to be compatible with the currently running acquisition application, for example ChartView.

1. Start by clicking the "Run Macro" toolbar button (following figure). This brings up the Macro dialog box, where we will create a new macro.



Clicking the <Run Macro> Button

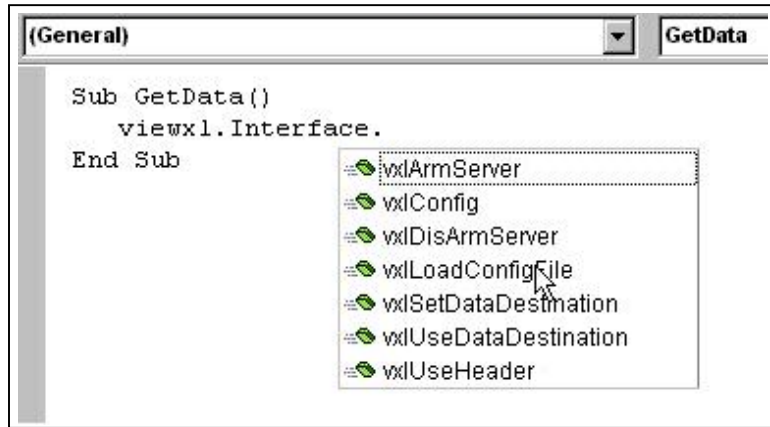
2. In the "Macro name" field, enter the name "GetData", then press the "Create" button, as indicated in the following figure.



Creating a Macro Named "GetData"

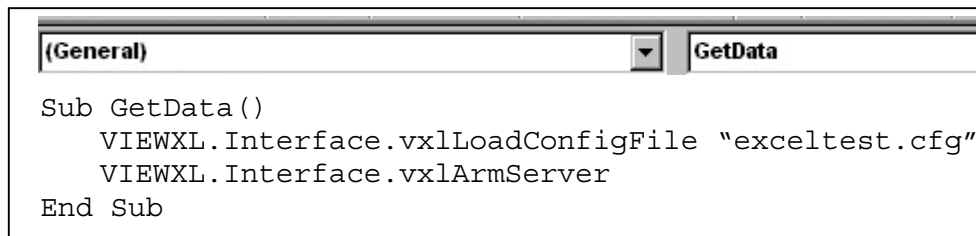
In the Visual Basic editor, you'll see the new macro that was created.

3. Inside the Macro (between the Sub and End Sub lines) type in the string "**viewxl.interface**" (see following figure). As you type the Period character after the word "**interface**" you will see a list pop-up that shows the ViewXL Add-In functions.
4. Select "**vx1LoadConfigFile**" from the list.



GetData Macro

5. Enter a Blank Space after the macro and enter the name of the configuration file. In this case we will use "**exceltest.cfg**" as indicated in the following figure.



Entering "exceltest.cfg" as the Configuration File Name

6. Finally, add a call to the "**vx1ArmServer**" function.

When done, your macro should look like the one shown above.

At this point you can return to the Macro dialog box and Run your "GetData" macro.

ViewXL API Definition

ViewXL provides a limited set of methods that you can use in your own Excel macros to leverage the power of the Add-In. The methods below make up the ViewXL Programming API. If you use these methods in your own Excel macros, there are certain precautions that you must take when designing your programs.

- If you start an acquisition, your program must periodically relinquish control so the ViewXL Add-In can perform its internal tasks. If your macro is simple, like the example given in the "Using the ViewXL Program API" section, then this is unnecessary. The sample macro is called, directs ViewXL to perform two tasks, then exits. During an acquisition, ViewXL polls the server every two seconds. Your macro must not run longer than this without releasing control.
- You must make certain that the currently running server can implement, or is compatible with, the API call being made. For example, In the *example macro*, page vxl-9, we trusted that the "exceltest.cfg" file was a valid configuration file for the currently running server.
- You must ensure that the server is running, before making a call that would be directed to it. You can't call the server if it is not running, or has not been selected in the *Options dialog box*. The Options dialog box is discussed on page vxl-4.

Function vxlArmServer() As Boolean

This function Arms the acquisition. If the server is not currently loaded, it will be loaded and Armed. The function returns True if successful, otherwise False. A False value would be returned if the server is already Armed, when the call is made.

Function vxlDisArmServer() As Boolean

This function dis-arms the server and aborts the acquisition. The function returns True if successful, otherwise False. A False value would be returned if the server is already dis-armed, when the call is made.

Sub vxlConfig()

This method starts the server. A server must have been previously selected in the *Options dialog box* before this call is made. The Options dialog box is discussed on page vxl-4.

Sub vxlLoadConfigFile(Optional ByRef sCfgFullPathName As String)

This method instructs the server to load the specified configuration file. The file name can not contain a drive or path specification. If the file name is omitted from the call, the *Load Configuration dialog box* is displayed, allowing you choose a configuration file. The Load Configuration dialog box is discussed on page vxl-4.

Sub vxlUseDataDestination(Optional ByVal bState As Boolean)

This is used to enable or disable the use of the *data destination* worksheet cell. If a True or False value is passed in the call, the data destination is enabled or disabled. If a value is omitted from the call, the data destination state is toggled on and off. Data destination is discussed on pages vxl-3 and 4.

Sub vxlUseHeader(Optional ByVal bState As Boolean)

This is used to enable or disable the use of the *data header*. If a True or False value is passed in the call, the data header is enabled or disabled. If a value is omitted from the call, the data header state is toggled on and off. The data header is discussed on pages vxl-3 and 4.

Sub vxlSetDataDestination(ByRef wksCell As Range)

This sets the data destination to the worksheet cell identified by the Excel Range object reference passed to the method. The Range object must not reference a range of cells that span multiple rows or columns. For example, the following call would set the data destination to Row 3 and Column C, then enable usage of the data destination.

```
VIEWXL.Interface.vxlSetDataDestination Range("C3")  
VIEWXL.Interface.vxlUseDataDestination True
```

Public Function vxlIsServerReady() As Boolean

Returns True if the Server is in a state where it can receive commands from an Excel VBA program. The Server would not be ready if it was busy performing a lengthy task, or if a message or dialog box that requires user action is displayed.

Public Function vxlIsAcquiring() As Boolean

This function returns True if the Server is Armed and acquiring data.

Public Function vxlGetAcqFiles(ByRef sArray() As String) As Boolean

This function gets the names of the files that have been acquired by the Server. If the Function returns True, there are files that have been acquired. The sArray argument is an undimensioned string array. On return, the array will be dimensioned and filled with the names of the acquired data files. Note. If the option to delete data files after reading them into Excel is checked, the file names returned by this function call may have already been deleted. The function can be used as follows:

```
Dim i As Integer
Dim sArray() As String
If (VIEWXL.Interface.vxlGetAcqFiles(sArray) Then
    For i = LBound(sArray) To UBound(sArray)
        ' Process Files
        sArray(i)
    Next
EndIf
```

Public Function vxlIsServerRunning() As Boolean

This function returns True if the Server is currently running.

Public Sub vxlUpdateViewXL()

This method synchronizes the ViewXL Menus and Toolbar with the current state of the Server. This is not normally needed if an acquisition has been started from the ViewXL interface. However, if an acquisition has been started from the Server application, this call can be used to update the ViewXL interface. The following example demonstrates the use of this, and other API functions.

```
' Check if the Server is running
If (Not VIEWXL.Interface.IsServerRunning()) Then
    Beep
    ' ask to start server if not running
    If (MsgBox("The server is not running. Do you want to start it?", _
        vbQuestion + vbYesNo) <> vbYes) Then
        Exit Sub
    End If
    VIEWXL.Interface.vxlConfig
End If

' update current server status
VIEWXL.Interface.vxlUpdateViewXL

' if the server is acquiring, stop it
If (VIEWXL.Interface.vxlIsAcquiring) Then
    ' halt the acquisition in progress
    VIEWXL.Interface.vxlDisArmServer
End If

' Continue with Excel VBA program
```