Creating Custom Menus to Organize Programs in Excel

Anjian Lu, CPD

After you have accumulated several reusable Excel templates, Add-Ins, UserForms, and other programs, you may want to organize them for convenient use. Fortunately, Excel has a feature to help you do this.

You can create custom menus on the main menu toolbar to hold your plumbing calculation tools. Figure 1 shows two custom menus: Time Sheet and Plumbing. Under the Plumbing menu are several submenus. You can create as many submenus and sub-submenus as you need. For example, Figure 1 shows the Water Demand submenu's three sub-submenus: Max Daily/Hourly, Fixture Unit, and FU_Flow Conversion. The latter is the calculator featured in this column in *PSD*'s May/June 2005 issue.

Now let's discuss how to add custom menus to the existing main menu bar, using the Plumbing menu as an example.

Writing the Code

To add a custom menu, you first need to write a subprocedure, or subroutine.

Open the file you created for converting water supply fixture unit (WSFU) to flow from the last issue's column. Save this file as an Excel Add-In, naming it PlbgMenu.xla, in the directory C:\PlumAl\Addins. Click the Visual Basic Editor (VBE) icon on the Visual Basic toolbar to bring up the Visual Basic window. In the Project — VBAProject window you will see the file you just saved, named VBAProject (PlbgMenu), in the directory tree. Highlight the file. Select Module from Insert on the main menu bar or right click on the highlighted file and select Insert — Module. The VBE window will appear automatically with this caption: Microsoft Visual Basic — PlbgMenu — [Module1 (Code)]. Change the Module1 default name to modAddPlbgMenu in its Property window. Now the Project Explore window will be similar to the one shown in Figure 2. (Based on the contents of your Excel application, the directory tree may look different.)

Write the subroutine found in the sidebar and name it AddPlbgMenu(). The first line creates an object variable cbMSMenuBar as a CommandBar type. The second line creates a control as type CommandBarControl (the Plumbing menu). The third line creates a variable to hold the index (or position) of an existing menu. Line four defines the command bar. The fifth line gets the index of the existing Window menu. This variable, iWindowIndex, will be used to define the position of the Plumbing menu. Line six places the Plumbing menu before the Window menu on the main menu bar.

The With muPlumbing — End With section holds all the tools you have created. The first line sets the menu name as Plumbing on the main menu bar. The embedded With — End With section adds an item to the drop-down menu, in this case Begin New Project. The sentence .OnAction = "ToBeDone" tells Excel to call the subroutine ToBeDone. You put a prime (') before this sentence to make it a comment, so the program won't execute it.

Figure 1. The Custom Menus

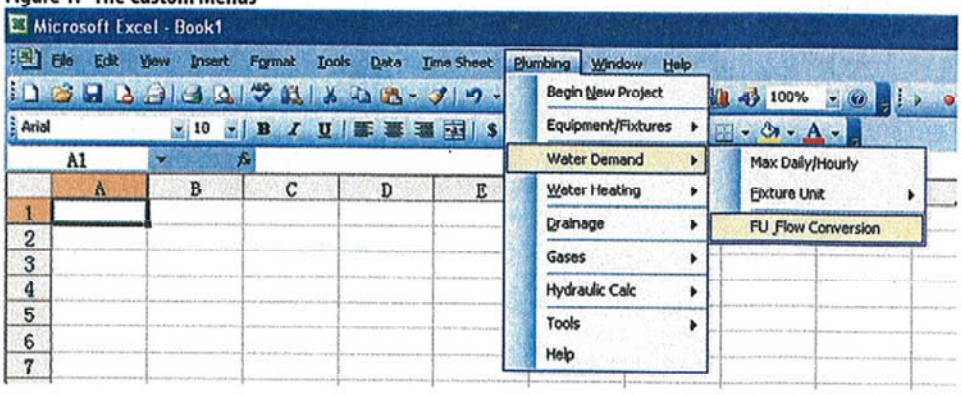
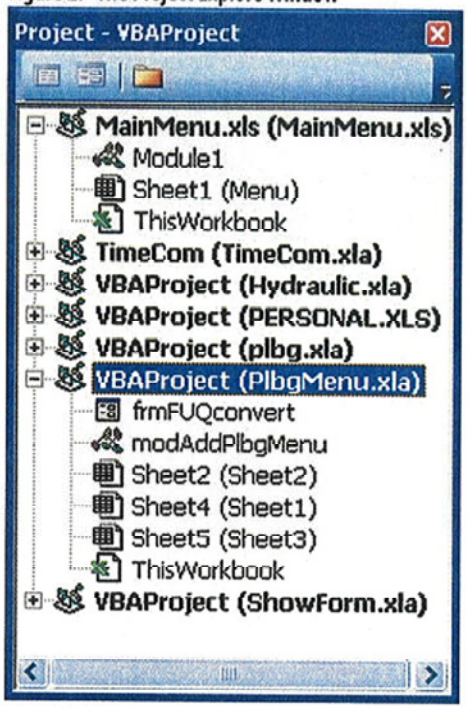


Figure 2. The Project Explore Window



Now let's look at the third embedded With — End With section. Within this section are three embedded With — End With sections. The first and second lines add the Water Demand submenu to the drop-down menu list. The .BeginGroup = True line in the first embedded section places a line on the main menu bar between the different menus. These three With — End With sections add the Water Demand submenu's sub-submenus: Max Daily/Hourly, Fixture Unit, and FU_Flow Conversion.

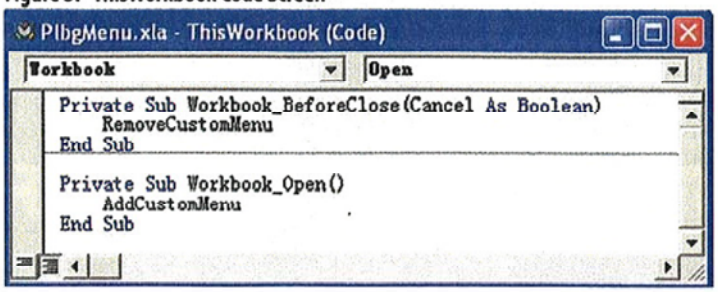
Writing Responding Subroutines

The menu's added control has a property called OnAction. You need to write subroutines to respond to it. To bring up the fixture unit versus flow calculator form, write the following subroutine:

```
Sub ShowfmFUQconvert()
   frmFUQconvert.Show
vbModeless
End Sub
```

The keyword vbModeless enables you to do other work without closing the form.

Figure 3. ThisWorkbook Code Screen



Similarly, you can include the hot water circulation system sizing template introduced in the November/December 2004 issue by writing the following subroutine:

```
Sub HWRecirculation()
   Workbooks.Add Template:=
"C:\PlumAl\WorkSheets\
HWRcalc.xlt"
End Sub
```

This subroutine uses the file HWRcalc.xlt as the template and creates a worksheet named HWRcalc1.xls.

To remind yourself and other users about menus that need to be developed, write the following subroutine:

```
Public Sub ToBeDone()
   MsgBox "To Be Developed",
vbInformation, "PlumAl"
End Sub
```

Code for Removing the Menu

To delete the Plumbing menu, write the following subroutine:

```
Public Sub RemovePlbgMenu()
  Dim cbWSMenuBar As
CommandBar
  On Error Resume Next
  Set cbWSMenuBar =
CommandBars("Worksheet Menu
Bar")
  cbWSMenuBar.
Controls("Plumbing").Delete
End Sub
```

This is the counterpart to the subroutine previously created. The line On Error Resume Next handles errors. Without this sentence, an error message will be thrown out and the program will crash if the Plumbing menu doesn't exist. With it, the program will bypass the error and execute to the end.

Adding and Removing a Menu

The last step is to add the Plumbing menu to the main menu bar and remove it when the PlbgMenu.xla Add-In file is excluded from the Add-In menu.

In the Project Explore window there is a file called ThisWorkbook (see Figure 2). Double click the file to bring up the code screen (see Figure 3).

Choose Workbook from the left box on the top and choose Open from the right box. A two-line subroutine will be created automatically. Add a line between the two lines, so the whole subroutine reads as follows:

```
Private Sub Workbook_Open()
   AddPlbgMenu
End Sub
```

Then choose BeforeClose from the right box and write a line between the two lines automatically created, so the whole subroutine reads as follows:

```
Private Sub Workbook_
BeforeClose(Cancel As
Boolean)
   RemovePlbgMenu
End Sub
```

Save this file and go to the Excel Worksheet screen. Add the file to the list in the Add-Ins form by selecting Add-Ins ... from Tool on the main menu. The Plumbing menu will be created and inserted automatically before the Window menu on the main menu bar. ■

Anjian Lu is an engineer with Vanderweil – NCR in Alexandria, Va. To comment on this article, or for more information, contact articles@psdmagazine.org.

The AddPlbgMenu Subroutine Code

```
Option Explicit
Public Sub AddPlbgMenu()
 Dim cbWSMenuBar As CommandBar
 Dim muPlumbing As CommandBarControl
 Dim iWindowIndex As Integer
 Set cbWSMenuBar = CommandBars("Worksheet Menu Bar")
 iWindowIndex = cbWSMenuBar.Controls("Window").Index
 Set muPlumbing = cbWSMenuBar.Controls.Add(Type:= _
     msoControlPopup, Before:=iWindowIndex)
 With muPlumbing
   .Caption = "&Plumbing"
   With .Controls.Add(Type:=msoControlButton)
    .Caption = "Begin &New Project"
'.OnAction = "ToBeDone"
   End With
   With .Controls.Add(Type:=msoControlPopup)
    .Caption = "Equipment/Fixtures"
    .BeginGroup = True
'.OnAction = "ToBeDone"
   End With
   With .Controls.Add(Type:=msoControlPopup)
    .Caption = "Water Demand"
    .BeginGroup = True
    With .Controls.Add(Type:=msoControlButton)
     .BeginGroup = True
     .Caption = "Max Daily/Hourly"
'.OnAction = "ToBeDone"
    End With
    With .Controls.Add(Type:=msoControlPopup)
     .Caption = "&Fixture Unit"
'.OnAction = "ToBeDone"
   End With
   With .Controls.Add(Type:=msoControlButton)
     .BeginGroup = True
     .Caption = "FU & Flow Conversion"
     '.OnAction = "ShowfmFUQconvert"
   End With
  End With
  With .Controls.Add(Type:=msoControlPopup)
   .Caption = "&Water Heating"
   With .Controls.Add(Type:=msoControlPopup)
     .Caption = "&Hot Water Demand"
'.OnAction = "ToBeDone"
   End With
   With .Controls.Add(Type:=msoControlButton)
     .Caption = "Water Heater Selection"
'.OnAction = "ToBeDone"
   End With
   With .Controls.Add(Type:=msoControlButton)
     .Caption = "Hot Water Circulation"
     .OnAction = "HWRecirculation"
   End With
   With .Controls.Add(Type:=msoControlButton)
     .Caption = "Expansion Tank"
'.OnAction = "ToBeDone"
   End With
  End With
  With .Controls.Add(Type:=msoControlPopup)
   .BeginGroup = True
   .Caption = "&Drainage"
'.OnAction = "ToBeDone"
  End With
  With .Controls.Add(Type:=msoControlPopup)
   .BeginGroup = True
   .Caption = "Gases"
'.OnAction = "ToBeDone"
  End With
  With .Controls.Add(Type:=msoControlPopup)
   .BeginGroup = True
   .Caption = "Hydraulic Calc"
'.OnAction = "ToBeDone"
  End With
  With .Controls.Add(Type:=msoControlPopup)
   .BeginGroup = True
   .Caption = "Tools"
   With .Controls.Add(Type:=msoControlButton)
     .Caption = "Calculator"
'.OnAction = "ToBeDone"
   End With
   With .Controls.Add(Type:=msoControlButton)
     .Caption = "... ..."
'.OnAction = "ToBeDone"
   End With
  End With
  With .Controls.Add(Type:=msoControlButton)
   .Caption = "Help"
'.OnAction = "ToBeDone"
  End With
 End With
End Sub
```