Lesson 4: Modifying the Menu

In this lesson, you will use the Menu Editor to customize the EmptyShellMenu entity. To this existing menu structure, you will add a new menu command, which will be used in the main window (StandardShellWindow) of the resulting application to open the Customer Orders data window you just created.

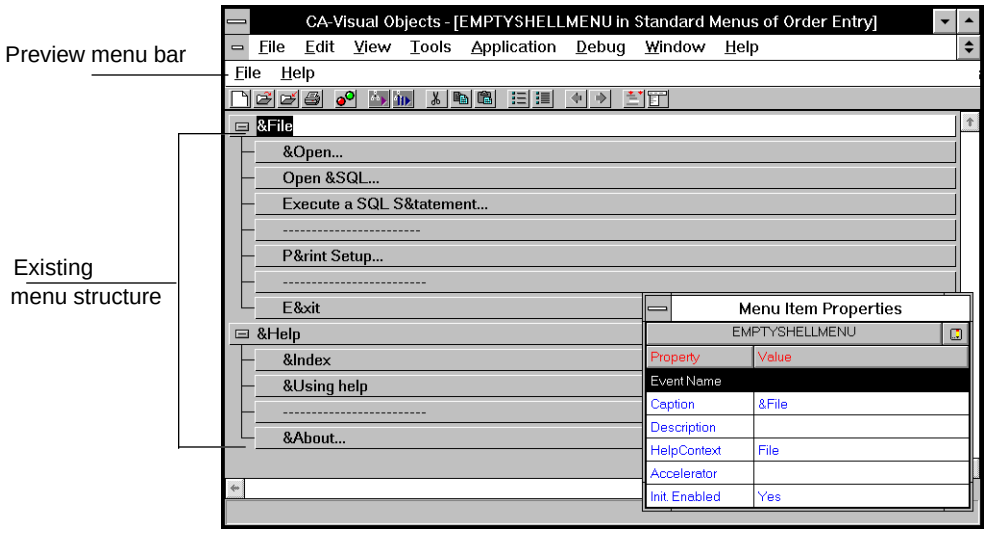
Note: As you probably recall, EmptyShellMenu and StandardShellWindow were generated as part of the Standard Program back in the first lesson.

Starting the Menu Editor

To modify this menu:

1. Close the App Windows Entity Browser if it is still open.
2. Double-click on the Standard Menus module to open its Entity Browser.
3. Double-click on the EmptyShellMenu menu entity.

The Menu Editor is launched—your desktop should now look as follows:

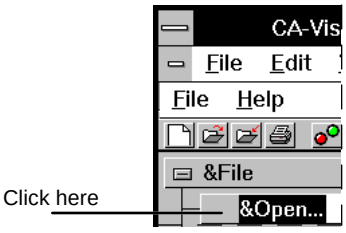


The menu structure for the empty shell menu appears as a tree structure in the Menu Editor workspace, and the preview menu bar shows the two existing menus, File and Help.

Adding the Customer Orders Menu Command

We're going to add a menu command titled Customer Orders to the File menu, just below the existing Open command:

1. Click on the line containing the text &Open.



2. Press Enter.

   This opens up a blank line, in which you can define the new menu command.

3. In the new, blank line, type **C&ustomer Orders...**.

   This is the name of the menu command as it will appear in the File menu. Note, however, that the ampersand (&) character will not appear in the menu; instead, it causes the letter "u" to be underlined in the menu, indicating it as the key for selecting the command.

   Note: The "..." is a Windows convention used to indicate that a menu command displays a dialog box or window.

4. Next, click on Event Name in the Properties window, type **CustOrd**, and press Enter.

   A menu command's Event Name property is used to define the action that should occur when the user chooses the menu command in the resulting application. We want the program to display the Customer Orders data window—therefore, all we need to do is supply the name of the data window (note that CustOrd is the data window's *name*, while Customer Orders is its title, or *caption*).

That's all the program needs to know to open the window when this menu command is chosen.

5. Finally, click on the Description property, type **Open the customer orders window**, and press Enter.

   This description will appear in the status bar when the menu command is highlighted.

More on Event Handling

Our work in the Menu Editor is just about complete. Before we move on, however, let's explore in more detail exactly how the menu command will display the Customer Orders data window. If you'll remember, we discussed event handling in the first lesson of this tutorial, including the hierarchy within the automatic event-handling system. This is an important feature of CA-Visual Objects, however, so it bears repeating.

By default, when the system encounters an event name, it first looks for a method name that matches the event name. If none is found, it then looks for a window name that matches. Finally, if no matching method or window name is found, it looks for a report. Thus, by assigning CustOrd to the event name, choosing the File Customer Orders menu command will automatically display the Customer Orders data window.

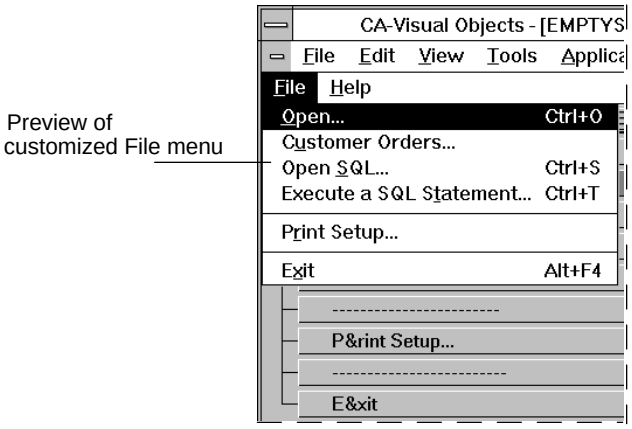
This feature makes it easy to connect events to menu commands and window controls (using the Menu and Window Editors, respectively). You simply enter the name of the method, window, or report that you want to activate as the Event property (as you did in the steps above), and the rest is taken care of automatically.

Previewing Your Work

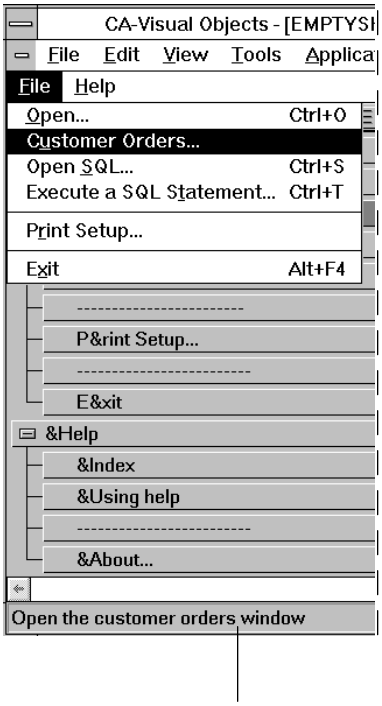
At any time, you can select the entries in the Menu Editor's preview menu bar (just as you would a real menu) to preview what your menus look like.

Note, however, that the preview menu bar is only partially operational. That is, it shows status bar descriptions and allows submenus to be pulled down, but nothing actually happens if you select a menu command. Its principal intent is to provide visual feedback while you are designing a menu structure.

To take a look at your new menu command, click on the File menu in the preview menu bar—you'll see your new menu command.



Then, if desired, use the Down arrow key to highlight the Customer Orders command—you'll see its description in the status bar (just as you will when you run the application in the conclusion to this tutorial).



Description defined for new menu command

Summary

Your menu is done. Double-click on the system menu to close the Menu Editor and choose Yes when prompted to save your work. The majority of the source code for this menu was reviewed in the first lesson, but if you look at it now, you will see several new define entities for the new menu command you just added.

As you have seen, the Menu Editor is very straightforward and easy to use. In this lesson, you've learned a key principle behind the automatic event handling inherent in all CA-Visual Objects applications—event handling by name. In the next lesson, you will use this feature once again to link two new methods to commands that you will add to the StandardShellMenu.