

## Menu Design

So far, we have been using command buttons to allow the user to issue a particular command. The use of *menus* in Windows applications is another common way to allow a user to execute different commands. A *menu* is a list of commands that appears under an application's title bar. A further list of commands will usually appear under each menu command. Let's create the following menu (with the shown sub-menus).

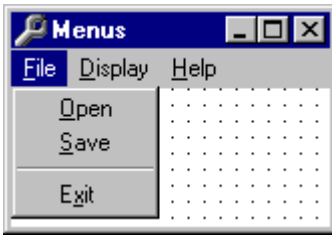


Figure 1-1: File menu

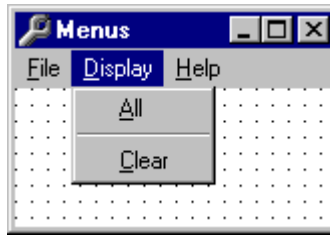


Figure 1-2: Display menu

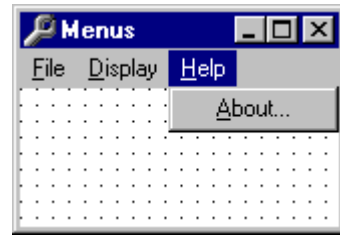


Figure 1-3: Help menu

- ☐ Start a new project. Change the form's *Caption* to **Menus** using the properties window.
- ☐ Click on the *Menu Editor* icon of the toolbar or select the *Menu Editor* command under the *Tools* menu. A window similar to the one shown below should appear.

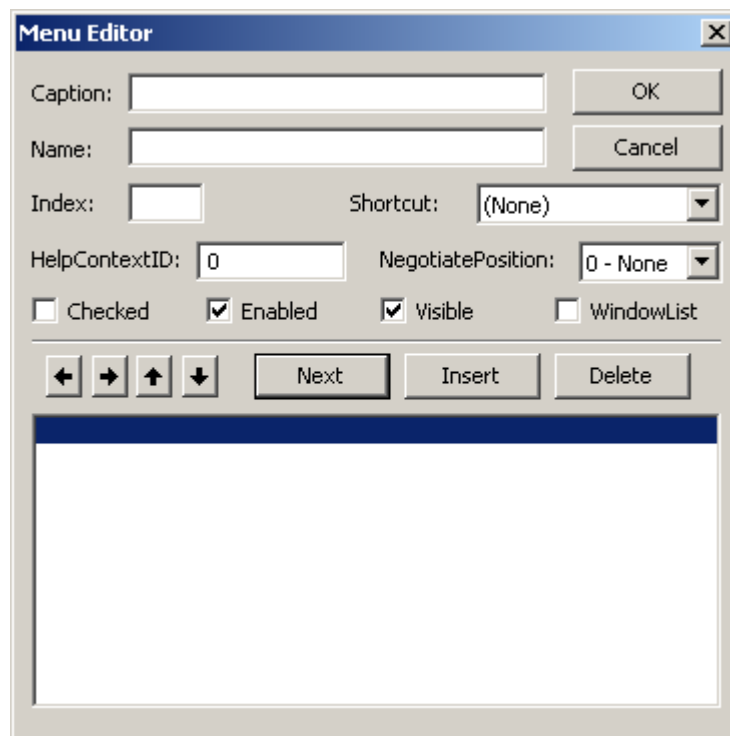


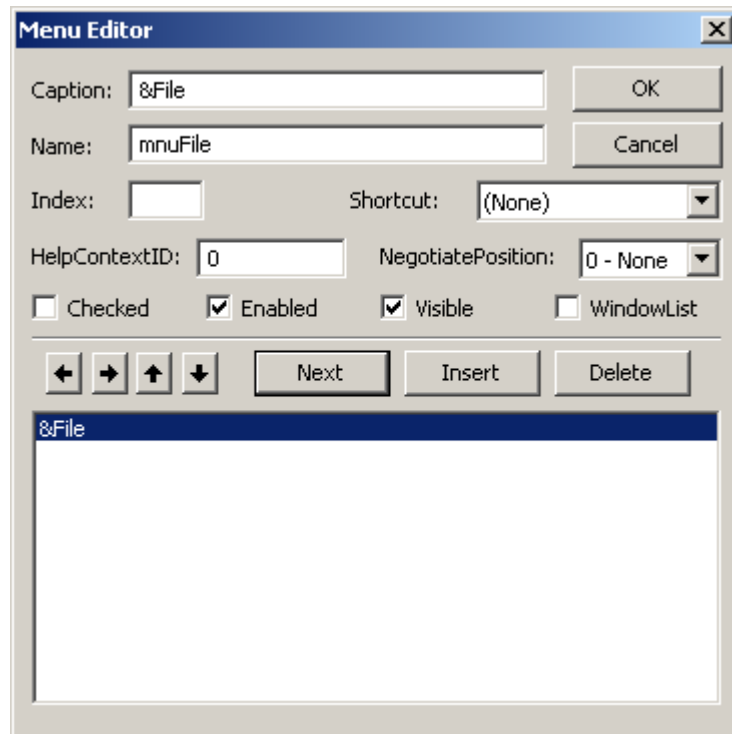
Figure 1-4: Menu Editor window

- ☐ In the *Caption* text box, type what you want to appear in the main menu. In our case, we want to have the command *File* appear. The value entered in the *Name* text box serves the same purpose as that of the *Name* property (to access through code). Type the following values in the *Caption* and *Name* text boxes:

Caption:	<b>&amp;File</b>
Name:	<b>mnuFile</b>

The ampersand character (&) that appears before a menu name, causes the letter following the ampersand to be underlined. This indicates to the user that the menu can be accessed by pressing the underlined letter after pressing the ALT key to activate the menu (in our case, the ALT and F keys would select the *File* menu when pressed simultaneously).

Your menu design window will now look like the one shown in *Figure 1-5* below.

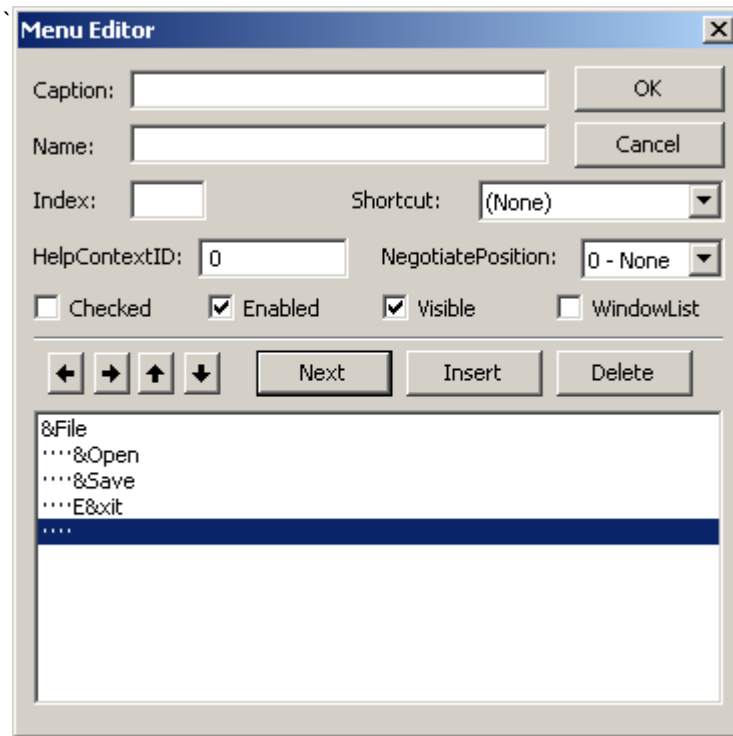


**Figure 1-5:** Main menu's *File* command

- ☐ Click on the *Next* button. This will clear the contents of the *Caption* and *Name* text boxes and will enable you to enter another menu command.
- ☐ Type the values **&Open** and **mnuOpen** in the *Caption* and *Name* text boxes respectively.

Recall that in a Windows application, we have a main menu with each main menu command having other commands beneath it. When building a menu, we need to specify which commands should appear at the main menu level and which commands do not. To specify a sub-menu, we indent the sub-menu command by clicking on the button with the arrow pointing to the right. If you indent the wrong command or indent a command too many times, you can remove the last indentation by clicking on the button with the arrow pointing to the left. In our example, the *Open* command needs to be indented once since we want it to appear under the *File* menu.

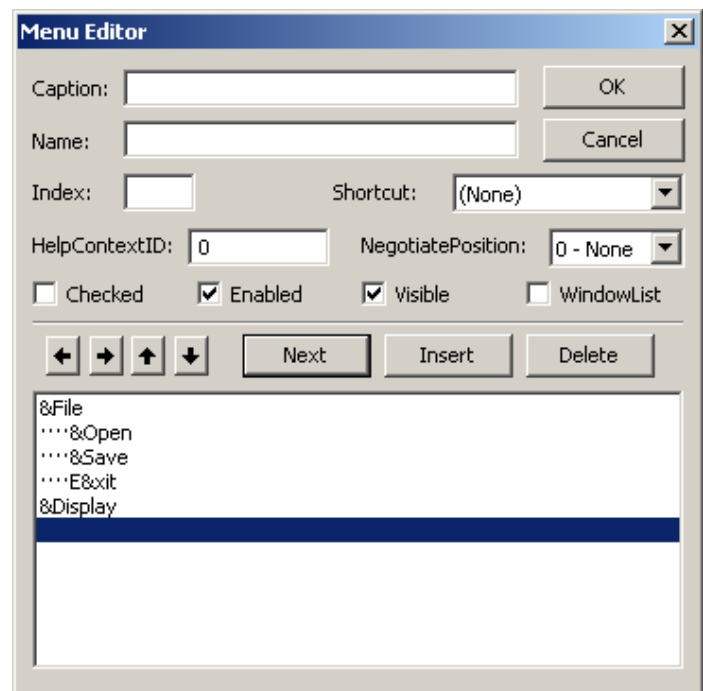
- ☐ Click on the command button with the arrow pointing to the right. This will indent the *Open* command. Click on the *Next* button to allow you to enter another menu command.
- ☐ Type the values **&Save** and **mnuSave** in the *Caption* and *Name* text boxes respectively. Click on the *Next* button to allow you to enter another menu command. Notice that you did not have to indent this command. Visual Basic automatically indents the menu commands that follow an indentation made by the user.
- ☐ Type the values **E&xit** and **mnuExit** in the *Caption* and *Name* text boxes respectively. Click on the *Next* button to allow you to enter another menu command. Notice that the ampersand does not have to appear before the first letter. In this case the letter *x* will be underlined in the menu. Your menu design window will now look like the one shown in *Figure 1-6* below.



**Figure 1-6:** File menu being built

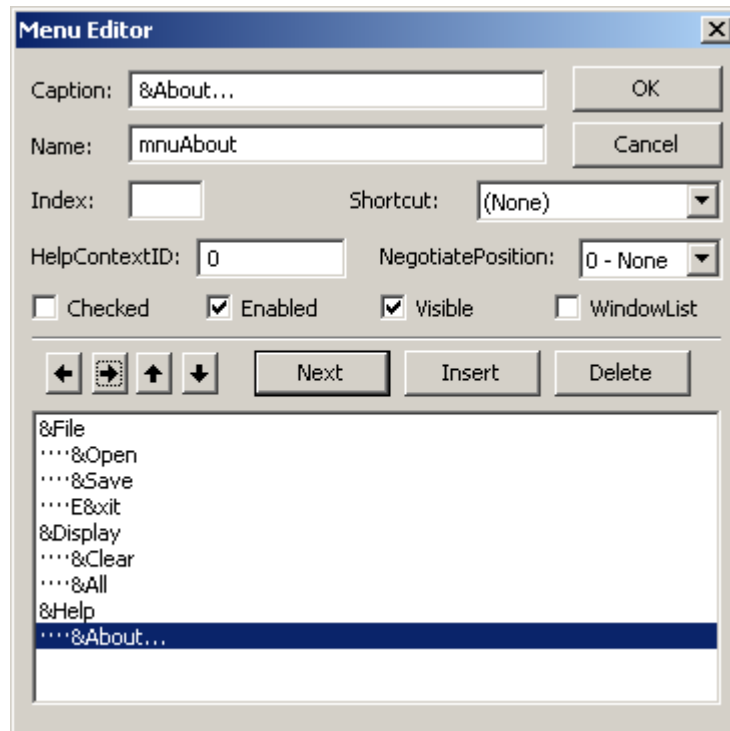
- ❑ The next menu command is the *Display* command. Type the values **&Display** and **mnuDisplay** in the *Caption* and *Name* text boxes respectively. Notice that the *Display* command is indented just like the previous three commands. If you leave it this way, this last command will appear under the *File* menu which is not what we want. In order to bring it to the main menu, we must remove the indentation.
- ❑ Remove the indentation by clicking on the command button with the arrow pointing to the left. Click on the *Next* button to allow you to enter another menu command. Your menu design window will now look like the one shown in *Figure 1-7* below.

- ❑ Type the values **&Clear** and **mnuClear** in the *Caption* and *Name* text boxes respectively. Indent this command by clicking on the button with the arrow pointing to the right (since this command will appear under the *Display* main menu command). Click on the *Next* button to allow you to enter another menu command.
- ❑ Type the values **&All** and **mnuAll** in the *Caption* and *Name* text boxes respectively. Click on the *Next* button to allow you to enter another menu command.
- ❑ Type the values **&Help** and **mnuHelp** in the *Caption* and *Name* text boxes respectively. Remove the indentation by clicking on the button with the arrow pointing to the left. Click on the *Next* button to allow you to enter another menu command.



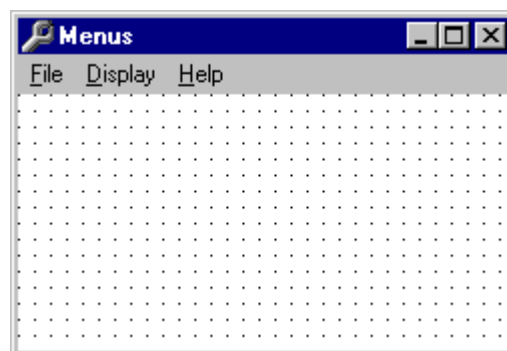
**Figure 1-7:** *Display* menu being built

- Type the values **&About...** and **mnuAbout** in the *Caption* and *Name* text boxes respectively. Indent this command by clicking on the button with the arrow pointing to the right. Your menu design window will now look like the one shown in *Figure 1-8* below.



**Figure 1-8:** Menu design

- Click on the OK button. The menu design window will close and you will be shown the main form which should look similar the one shown in *Figure 1-9* below.



**Figure 1-9:** Main form with menu

- Run your application and click on the *File*, *Display* and *Help* menus to see their sub-menus which should look similar to the ones shown in *Figures 1-1*, *1-2* and *1-3*. If you compare them carefully however, you will see the line before the *Exit* command under the *File* menu is missing in your menu; also, the line before the *All* command under the *Display* menu is also missing. A line which separates a set of commands within a menu is known as a *separator* and we will go back to our menu design window to add the two missing separators.
- Click on the *Menu Editor* icon of the toolbar or select the *Menu Editor* command under the *Tools* menu of Visual Basic. The menu editor window should appear showing the current form's menu structure. It should look similar to the one shown in *Figure 1-8*.
- Select the *E&xit* command of the menu by clicking on it; it should become highlighted. We selected this command, because we want to insert another command before it (namely the separator line).

- ❑ Click on the *Insert* command button. A blank line should appear between the *&Save* and the *E&xit* commands. The *Insert* button is used to insert a new command preceding our currently highlighted command. Similarly, the *Delete* command button can be used to delete the currently highlighted command.
- ❑ Type the values - and **mnuSeparator** in the *Caption* and *Name* text boxes respectively. The important part here, is the hyphen typed in the *Caption* text box. It represents a separator line.
- ❑ Click on the *&All* command and then click on the *Insert* button to enter a blank line between the *&Clear* and *&All* commands.
- ❑ Once again type the values - and **mnuSep** in the *Caption* and *Name* text boxes respectively (make sure you type a hyphen and not an underscore).
- ❑ Click on the OK button to exit from the menu editor window and click through the form's main menu to see that the separator lines have now been inserted.
- ❑ Save your project as MENUS (save both the form and the project with the same name).
- ❑ Click on the *File* menu and then the *Exit* command at design time. This is the most common method used to allow you to write code for a menu command. The code window will appear showing you the *mnuExit\_Click* event procedure. This procedure will be executed whenever the user clicks on the *Exit* command. Type the necessary code to end the execution of your program.

```
Private Sub mnuExit_Click ()
    End
End Sub
```

- ❑ Similarly, enter the following code for the *About...* command under the *Help* menu.

```
Private Sub mnuAbout_Click ()
    MsgBox "Menu Design and File Access"
End Sub
```

This menu command will display a simple message box. The code for the rest of the commands will be inserted with additional exercises (to follow).