```
                                else
                                    msg += "closed"
                                endif
                            endif
                            DoAlert 0, msg
                            break
        endswitch
        return 0
    End

    Function ListBoxContextMenu()
        Make/N=(5,2)/T/O ListWaveContext
        ListWaveContext[][0] = "Column 1 should be:"
        ListWaveContext[][1] = "Row "+num2str(p)
        Make/N=(5,2)/O SelWaveContext
        SelWaveContext=0
        NewPanel /W=(150,53,450,253)
        ListBox list_w_context,pos={50,30},size={200,150},proc=ListBoxContextProc,widths={3,1}
        ListBox list_w_context,listWave=root:ListWaveContext,selWave=root:SelWaveContext
    EndMacro
```

Copy the code above to the Procedure window and compile. On the command line, invoke the function to build the panel:

```
ListBoxContextMenu()
```

In the listbox, right-clicking (Windows) or Ctrl-clicking (Macintosh) in the left column of the listbox will present a contextual menu with choices for what to do with the right column. If you choose Checkbox or Disclosure, and the cell is enabled, when you click on the cell in the right column, an alert is displayed telling you about what happened.

An example experiment that lets you easily experiment with ListBox settings is available in "Examples:Feature Demos 2:ListBox Demo.pxp".

**See Also**

Chapter III-14, **Controls and Control Panels**, for details about control panels and controls.

**Control Panel Units** on page III-444 for a discussion of the units used for controls.

The **GetUserData** function for retrieving named user data.

The **ControlInfo** operation for information about the control.

**Setting Bit Parameters** on page IV-12 for further details about bit settings.

# ListBoxControl

**ListBoxControl**

ListBoxControl is a procedure subtype keyword that identifies a macro or function as being an action procedure for a user-defined listbox control. See **Procedure Subtypes** on page IV-204 for details. See **ListBox** for details on creating a listbox control.

# ListMatch

**ListMatch(*listStr*, *matchStr* [, *listSepStr*])**

The ListMatch function returns each list item in *listStr* that matches *matchStr*.

*ListStr* should contain items separated by *listSepStr* which typically is ";".