```
Print StringFromList(line+1, str,"\r")    // Print Path:
Print StringFromList(line+2, str,"\r")    // Print Symbolic Path:
Print StringFromList(line+3, str,"\r")    // Selection Start:
Print StringFromList(line+4, str,"\r")    // Selection End:
```

See Also

Saving a Window as a Recreation Macro on page II-47.

WinType

WinType(*winNameStr*)

The WinType function returns a value indicating the type of the named window.

Details

winNameStr is a string or string expression containing the name of a window or subwindow, or "" to signify the target window. When identifying a subwindow with *winNameStr*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.

WinType returns the following values:
0:   No window by that name.

1:	Graph
2:	Table
3:	Layout
5:	Notebook
7:	Panel
13:	XOP target window
15:	Camera window in Igor Pro 7.00 or later
17:	Gizmo window in Igor Pro 7.00 or later

Because command and procedure windows do not have *names* (they only have *titles*), WinType can not even be asked about those windows.

See Also

The **WinName**, **ChildWindowList**, and **WinList** functions.

WMAxisHookStruct

See **NewFreeAxis** for further explanation of WMAxisHookStruct.

```
Structure WMAxisHookStruct
    char win[200]        // Host window or subwindow name
    char axName[32]      // Name of axis
    char mastName[32]    // Name of controlling axis or ""
    char units[50]       // Axis units.
    Variable min         // Current axis range minimum value
    Variable max         // Current axis range maximum value
EndStructure
```

WMBackgroundStruct

See **CtrlNamedBackground**, **Background Tasks** on page IV-319, and **Preemptive Background Task** on page IV-335 for further explanation of WMBackgroundStruct.

```
Structure WMBackgroundStruct
    char name[32]        // Background task name
    UInt32 curRunTicks   // Tick count when task was called
    Int32 started        // TRUE when CtrlNamedBackground start is issued
    UInt32 nextRunTicks  // Precomputed value for next run
                         // but user functions may change this
EndStructure
```