Examples

```
Function PrintAllWaveNames()
   String objName
   Variable index = 0
   do
      objName = GetIndexedObjName(":", 1, index)
      if (strlen(objName) == 0)
         break
      endif
      Print objName
      index += 1
   while(1)
End
```

See Also

The **CountObjects** function, and Chapter II-8, **Data Folders**.

GetIndexedObjNameDFR

GetIndexedObjNameDFR(*dfr*, *objectType*, *index*)

The GetIndexedObjNameDFR function returns a string containing the name of the indexth object of the specified type in the data folder referenced by *dfr*.

GetIndexedObjNameDFR is the same as GetIndexedObjName except the first parameter, *dfr*, is a data folder reference instead of a string containing a path.

Parameters

index starts from zero. If no such object exists a zero length string ("") is returned.

objectType is one of the following values:

objectType	**What You Get**
1	Waves
2	Numeric variables
3	String variables
4	Data folders

Examples

```
Function PrintAllWaveNames()
   String objName
   Variable index = 0
   DFREF dfr = GetDataFolderDFR()    // Reference to current data folder
   do
      objName = GetIndexedObjNameDFR(dfr, 1, index)
      if (strlen(objName) == 0)
         break
      endif
      Print objName
      index += 1
   while(1)
End
```

See Also

Data Folders on page II-107, **Data Folder References** on page IV-78, **Built-in DFREF Functions** on page IV-81, **CountObjectsDFR**

GetKeyState

GetKeyState(*flags*)

The GetKeyState function returns a bitwise numeric value that indicates the state of certain keyboard keys.

To detect keyboard events directed toward a window that you have created, such as a panel window, use the window hook function instead of GetKeyState. See **Window Hook Functions** on page IV-293 for details.

GetKeyState is normally called from a procedure that is invoked directly through a user-defined button or user-defined menu item. The procedure tests the state of one or more modifier keys and adjusts its behavior accordingly.

Another use for GetKeyState is to determine if Escape is pressed. This can be used to detect that the user wants to stop a procedure.

GetKeyState tests the keyboard at the time it is called. It does not tell you if keys were pressed between calls to the function. Consequently, when a procedure uses the escape key to break a loop, the user must press Escape until the running procedure gets around to calling the function.

Parameters

flags is a bitwise parameter interpreted as follows:

Bit 0:	If set, GetKeyState reports keys keys even if Igor is not the active application. If cleared, GetKeyState reports keys only if Igor is the active application.

All other bits are reserved and must be zero.

Details

When set, the return value is interpreted bitwise as follows:

Bit 0:	Command (*Macintosh*) or Ctrl (*Windows*) pressed.
Bit 1:	Option (*Macintosh*) or Alt (*Windows*) pressed.
Bit 2:	Shift pressed.
Bit 3:	Caps Lock pressed.
Bit 4:	Control pressed (*Macintosh only*).
Bit 5:	Escape pressed.
Bit 6:	Left arrow key pressed. Supported in Igor Pro 7.00 or later.
Bit 7:	Right arrow key pressed. Supported in Igor Pro 7.00 or later.
Bit 8:	Up arrow key pressed. Supported in Igor Pro 7.00 or later.
Bit 9:	Down arrow key pressed. Supported in Igor Pro 7.00 or later.

To test if a particular key is pressed, do a bitwise AND of the return value with the mask value 1, 2, 4, 8, 16, 32, 64, 128, 256 and 512 for bits 0 through 9 respectively. To test if a particular key and only that key is pressed compare the return value with the mask value.

On Macintosh, it is currently possible to define a keyboard shortcut for a user-defined menu item and then, in the procedure invoked by the keyboard shortcut, to use GetKeyState to test for modifier keys. This is not possible on Windows because the keyboard shortcut will not be activated if a modifier key not specified in the keyboard shortcut is pressed. It is also possible that this ability on Macintosh will be compromised by future operating system changes. On both operating systems, however, you can test for a modifier key when the user chooses a user-defined menu item or clicks a user-defined button using the mouse.

Examples

```
Function ShiftKeyExample()
    Variable keys = GetKeyState(0)

    if (keys == 0)
        Print "No modifier keys are pressed."
    endif

    if (keys & 4)
        if (keys == 4)
            Print "The Shift key and only the Shift key is pressed."
        else
            Print "The Shift key is pressed."
        endif
    endif
End
```