Double

double *localName*

Declares a local 64-bit double-precision variable in a user-defined function or structure.

Double is another name for Variable. It is available in Igor Pro 7 and later.

DoUpdate

DoUpdate [*/E=e /W=targWin /SPIN=ticks*]

The DoUpdate operation updates windows and dependent objects.

Flags

/E=*e* Used with /W, /E=1 marks window as a progress window that can accept mouse events while user code is executing. Currently, only control panel windows can be used as a progress window.

/W=*targWin* Updates only the specified window. Does not update dependencies or do any other updating.

Currently, only graph and panel windows honor the /W flag.

V_Flag is set to the truth the window exists. See **Progress Windows** on page IV-156 for other values for V_Flag.

/SPIN=*ticks* Sets the delay between the start of a control procedure and the spinning beachball. *ticks* is the delay in ticks (60th of a second.) Unless used with the /W flag, /SPIN just sets the delay and an update is not done.

Details

Call DoUpdate from an Igor procedure to force Igor to update any objects that need updating. Igor updates any windows that need to be updated and also any objects (string variables, numeric variables, waves, controls) that depend on other objects that have changed since the last update. Page layout windows may not be immediately updated. For more information on page layout updates, see **Automatic Updating of Layout Objects** on page II-487.

Igor performs updates automatically if:

- No user-procedure is running.
- An interpreted procedure (Macro, Proc, Window type procedures) is running and PauseUpdate or DelayUpdate is not in effect.

Igor does not perform an automatic DoUpdate while a user-defined function is running. You can call DoUpdate from a user-defined function to force an update.

See Also

The **DelayUpdate**, **PauseUpdate**, and **ResumeUpdate** operations, **Progress Windows** on page IV-156.

DoWindow

DoWindow [*flags*] [*windowName*]

The DoWindow operation controls various window parameters and aspects. There are additional forms for DoWindow when the /S or /T flags are used; see the following DoWindow entries.

DoWindow does not support **Subwindow Syntax**.

Parameters

windowName is the name of a top-level graph, table, page layout, notebook, panel, Gizmo, camera, or XOP target window. *windowName* can not be a subwindow path.

A window’s name is *not* the same as its title. The title is shown in the window’s title bar. The name is used to manipulate the window from Igor commands. You can check both the name and the title using the Window Control dialog (in the Arrange submenu of the Window menu).

Flags

/B[=*bname*] Moves the specified window to the back (to the bottom of desktop) or behind window *bname*.

/C Changes the name of the target window to the specified name. The specified name must not be used for any other object except that it can be the name of an existing window macro.

/C/N Changes the target window name and creates a new window macro for it. However, /N does nothing if a macro or function is running. /N is not applicable to notebooks.

/D Deletes the file associated with window, if any (for notebooks only).

/F Brings the window with the given name to the front (top of desktop).

/H Specifies the command window as the target of the operation. When using /H, *windowName* must not be specified and only the /B and /HIDE flags are honored.

Use /H to bring the command window to the front (top of desktop).

Use /H/B to send the command window to the bottom of the desktop.

Use /H/HIDE to hide or show the command window.

/HIDE=*h* Sets hidden state of a window.

h=0: Visible.
h=1: Hidden.
h=?: Sets the variable V_flag as follows:
0: The window does not exist.
1: The window is visible.
2: The window is hidden.

You can also read the hidden state using **GetWindow** and set it using **SetWindow**.

/K Kills the window with the given name.

We recommend using **KillWindow** instead of DoWindow/K.

/N Creates a new window macro for the window with the given name. However, /N does nothing if a macro or function is running. /N is not applicable to notebooks.

/R Replaces (updates) the window macro for the named window or creates it if it does not yet exist. However, /R does nothing if a macro or function is running. /R is not applicable to notebooks.

/R/K Replaces (updates) the window macro for the named window or creates it if it does not yet exist and then kills the window. However, /R does nothing if a macro or function is running. /R is not applicable to notebooks.

/W=*targWin* Designates *targWin* as the target window; it also requires that you specify *windowName*. Use this mainly with floating panels, which are always on top. You can use a subwindow specification of an external subwindow only with the /T flag or without any flags.

Details

DoWindow sets the variable V_flag to 1 if there was a window with the specified name after DoWindow executed, to 0 if there was no such window, or to 2 if the window is hidden.

You can call DoWindow with a *windowName* and no flags to check if a window exists without altering the window. A better method is to use **WinType** which supports subwindows.

When used with the /N flag, *windowName* must not conflict with the name of any other object. When used with the /C flag, *windowName* must not conflict with the name of any other object except that it can be the name of an existing window macro.