ModifyProcedure

```
ModifyProcedure [/A[=all] /W=procWinTitleListStr] /Z[=z]
    [procedure=functionOrMacroNameStr, hide=h, lock=ro, writeProtect=wp,
    userCanOverride=ovr]
```

The ModifyProcedure operation modifies one or more procedure windows, depending on the /A and /W flags and on the procedure keyword.

The ModifyProcedure operation was added in Igor 8.03.

Use /W="Procedure" to modify the built-in procedure window.

Use /W=procWinTitleListStr to modify a procedure window specified by its title, such as /W="MyProc" for packed procedure files or /W="MyProc.ipf" for standalone procedure files.

Use procedure=*functionOrMacroNameStr* to modify a procedure window containing a particular function.

See **Specifying Which Procedure Windows to Modify** below for other usages.

If the procedure window is associated with a standalone file, as opposed to being packed, the file's read-only setting may be set or cleared, as described below for the lock and writeProtect keywords.

Parameters

procedure=*functionOrMacroNameStr*

functionOrMacroNameStr is a string containing the name of a function or macro in the procedure window to modify.

functionOrMacroNameStr may be a simple name or may include independent module and/or module name prefixes to designate static functions. For use with independent modules, see **Using ModifyProcedure With Independent Modules** below.

lock=*ro*

Sets or clears the read-only state of the targeted procedure windows.

ro=-2: Clears the read-only state, like *ro*=0.

In addition, if the procedure window is associated with a standalone file, as opposed to being packed, the file's read-only setting is set to unlocked, as with **SetFileFolderInfo**/RO=0.

ro=-1: The read-only state is not changed.

ro=0: Clears the read-only state. This allows the user to change the text if the procedure window is also not write-protected.

ro=1: Sets the read-only state. The procedure window shows a lock icon in the lower left corner.

ro=-2: Sets the read-only state, like *ro*=1.

In addition, if the procedure window is associated with a standalone file, as opposed to being packed, the file's read-only setting is set to locked, as with **SetFileFolderInfo**/RO=1.

hide=*h*

Hides or shows the targeted procedure windows. Added in Igor 9.00.

h=-1: The procedure window visibility is not changed.

h=0: Shows the procedure window.

h=1: Hides the procedure window.

writeProtect=*wp* Sets or clears the write-protect state of the targeted procedure windows.

wp=-2: Clears the write-protect state, like *wp*=0.

In addition, if the procedure window is associated with a standalone file, as opposed to being packed, the file's read-only setting is set to unlocked, as with **SetFileFolderInfo**/RO=0.

wp=-1: The write-protect state is not changed.

wp=0: Clears the write-protect state. The user can change the text if the procedure window is also not locked.

wp=1: Sets the write-protect state. The procedure window shows a write-protect icon (a pencil with a red circle-and-line indicator).

wp=2: Sets the write-protect state, like *wp*=1.

In addition, if the procedure window is associated with a standalone file, as opposed to being packed, the file's read-only setting is set to locked, as with **SetFileFolderInfo**/RO=1.

userCanOverride=*ovr*

Controls the ability of the user to change the write-protect state of the targeted procedure windows.

ovr=0: Prevents the user from changing the write-protect state by clicking the write-protect icon.

ovr=1: Allows the user to change the write-protect state by clicking the write-protect icon.

Flags

/A[=*all*] Specifies all procedure windows as the targets of the ModifyProcedure operation.

Use /A=1 to target all procedure windows in the ProcGlobal module. /A alone is the same as /A=1.

Use /A=2 to target all procedure windows in independent modules. See **Using ModifyProcedure With Independent Modules** for details.

/W=*procWinTitleListStr* Specifies a particular procedure window or a list of procedure windows as the targets of the ModifyProcedure operation.

If *procWinTitleListStr* is the title of a single procedure window, this specifies that procedure window as the single target.

If *procWinTitleListStr* contains a semicolon-separated list of procedure window titles, this specifies those procedure windows as targets.

See **Specifying Which Procedure Windows to Modify** below for details.

/Z[=*z*] Prevents procedure execution from aborting if an error occurs. Use /Z or /Z=1 if you want to handle errors from ModifyProcedure in your procedures rather than having execution abort.

Details

When changing the lock or write-protect states for a standalone file (lock=+/-2 or writeProtect=+/-2), there are scenarios where the file's locked state can't be changed. This would occur, for example, if the file is open in another program or if you do not have sufficient privileges to modify the file.

Specifying Which Procedure Windows to Modify

The simplest way to modify one procedure window whose title you know is to use /W:

```
ModifyProcedure/W="My Procedure.ipf" writeProtect=2   // 2 for standalone file
ModifyProcedure/W="Procedure" writeProtect=1      // Modifies built-in procedure window
```

Use /A=1 to target all procedure windows in the ProcGlobal module:

```
ModifyProcedure/A=1 writeProtect=1
```

Don't specify both /A and /W as this is ambiguous and generates an error.

Use procedure=*functionOrMacroNameStr* to modify all procedure windows in the ProcGlobal module containing a procedure with the specified name:

```
ModifyProcedure procedure=MyFunction, writeProtect=1
```

If you omit procedure=*functionOrMacroNameStr*, /A, and /W then the built-in procedure window is modified.

Use *functionOrMacroNameStr* only when you are not certain which procedure window contains the function or macro. If a procedure window has syntax errors that prevent Igor from determining where functions and macros start and end, then ModifyProcedure may not be able to locate the correct procedure window in which case it returns an error.

You can use /W and procedure=*functionOrMacroNameStr* to search for a function or macro in one or more procedure windows. This is useful when *functionOrMacroNameStr* is the name of a static function in a procedure window that uses #pragma moduleName.

Using ModifyProcedure With Independent Modules

To work with independent modules, you must enable independent module development. This is for advanced programmers only. See **Independent Modules** on page IV-238 for details. The material in this section assumes that independent module development is enabled.

/W=*procWinTitleListStr* can be a list of procedure window titles each followed by a space and an independent module name in brackets. Then procedure=*functionOrMacroNameStr* applies to the specified procedure windows and independent modules.

For example, if any procedure file contains these statements:

```
#pragma IndependentModule=myIM
#include <Axis Utilities>
```

then the commands

```
ModifyProcedure/W="Axis Utilities.ipf [myIM]" procedure="HVAxisList"
Print S_windowList, V_isReadOnly, V_writeProtect
```

report the title, read-only state, and write-protect state of the procedure window that contains the HVAxisList function, which is in the "Axis Utilities.ipf" file and the independent module myIM.

Similarly, procedure=*functionOrMacroNameStr* may also specify an independent module prefix followed by the # character. The preceding ModifyProcedure command can be rewritten as:

```
ModifyProcedure/W="Axis Utilities.ipf" procedure="myIM#HVAxisList"
```

or simply:

```
ModifyProcedure procedure="myIM#HVAxisList"
```

You can use the same syntax to modify the procedure window containing a static function in a non-independent module procedure file using a module name instead of, or in addition to, the independent module name.

procWinTitleListStr can also be just an independent module name in brackets to target all procedure windows that belong to the named independent module containing the specified function:

```
ModifyProcedure/W="[myIM]" procedure="HVAxisList", writeProtect=0
```

Output Variables

The ModifyProcedure operation returns information in the following variables for the last procedure window targeted by the parameters. The information returned reflects the state of affairs after the ModifyProcedure command performs any requested modifications except for V_wasHidden.

`V_wasHidden`	If the procedure window was previously hidden, V_wasHidden is set to 1, otherwise to 0.
`V_isReadOnly`	If the procedure's lock icon is showing, V_isReadOnly is set to 1, otherwise to 0.
`V_writeProtect`	If the pencil with the red line icon is showing, V_writeProtect is set to 1, otherwise to 0.
`V_userCanOverride`	Set to 1 (the default) if the user can change the write-protect state to writable.