Chapter
II-4

Windows

Windows

This section describes Igor's windows in general terms, the Windows menu, and window recreation macros.

Detailed information about each type of window can be found in these chapters:

Window Type	Chapter
Command window	Chapter II-2, **The Command Window** Chapter IV-1, **Working with Commands**
Procedure windows	Chapter III-13, **Procedure Windows**
Help windows	Chapter II-1, **Getting Help**
Graphs	Chapter II-12, **Graphs**
Tables	Chapter II-11, **Tables**
Layouts	Chapter II-17, **Page Layouts**
Notebooks	Chapter III-1, **Notebooks**
Control panels	Chapter III-14, **Controls and Control Panels**
Gizmo windows	Chapter II-16, **3D Graphics**
Camera windows	**NewCamera** on page V-566

The Command Window

When Igor first starts, the **command window** appears at the bottom of the screen.

Commands are automatically entered and executed in the command window's **command line** when you use the mouse to "point-and-click" your way through dialogs. You may optionally type the commands directly and press Return or Enter. Igor preserves a history of executed commands in the **history area**.

For more about the command window, see Chapter II-2, **The Command Window**, and Chapter IV-1, **Working with Commands**.

The Rest of the Windows

At startup, by default, Igor displays a table window. There are also a number of additional windows which are initially hidden:

- The main procedure window
- The Igor Help Browser
- Help windows for files in "Igor Pro 7 Folder/Igor Help Files" and "Igor Pro User Files/Igor Help Files"

You can create additional windows for graphs, tables, page layouts, notebooks, control panels, Gizmos (3D plots), and auxiliary procedure windows, as well as more help windows.

The Target Window

Igor commands and menus operate on the **target window**. The target window is the top graph, table, page layout, notebook, control panel or XOP target window. The term "target" comes from the fact that these windows can be the target of command line operations such as ModifyGraph, ModifyTable and so on. The command window, procedure windows, help windows and dialogs can not be targets of command line operations and thus are not target windows.

Prior to version 4, Igor attempted to draw a special icon to indicate which window was the target. However, this special target icon is no longer drawn because of operating system conflicts.

The menu bar changes depending on the top window and the target window. For instance, if a graph is the target window the menu bar contains the Graph menu. However, you may type any command into the command line, including commands that do not apply to the target window. Igor will apply the command to the top window of the correct type.

Sometimes the top window isn't a target window, but it causes the menu bar to change. For example, if you activate a procedure window, the Procedure menu appears in the menu bar.

Window Names and Titles

Each graph, table, page layout, control panel, notebook, and Gizmo has a **title** and a **name**.

The title is what you see at the top of the window frame and in the Windows menu. Its purpose is to help you visually identify the window, and is usually descriptive of its contents or purpose.

The window *name* is not the same as the *title*. The purpose of the name is to allow you to refer to the window from a command, such as the DoWindow or AppendToGraph operations.

When you first create one of these windows, Igor gives it a name like Graph0, Table0, Layout0 or Panel0, and a title based on the name and window contents. You can change the window's title and name to something more descriptive using the Window Control dialog (Windows→Control submenu). Among other things, it renames and retitles the target window.

The Window Control dialog is also a good way to discover the name of the top window, since the window shows only the window title.

The command window, procedure windows, and help windows have *only* a title. The title is the name of the file in which they are stored. These windows do not have names because they can not be affected by command line operations.

Allowable Window Names

A window name is used for commands and therefore must follow the standard rules for naming Igor objects:

- The name must start with a letter.
- Additional characters can be alphanumeric or the underscore character.
- No other characters, including spaces, are allowed in standard Igor object names.
- No more than 31 bytes are allowed.
- The name must not conflict with other object names (you see a message if it does).

For more information, see **Object Names** on page III-443.

The Open File Submenu

The File menu contains the Open File submenu for opening an existing file as a notebook, Igor help window, or procedure window.

When you choose an item from the submenu, the Open File dialog appears for you to select a file.

The Windows Menu

You can use the Windows menu for making new windows, and for showing, arranging and closing (either hiding or "killing") windows. You can also execute "window recreation macros" that recreate windows that have been killed and "style macros" that modify an existing window's appearance.

Making a New Window

You can use the various items in the Windows menu and Windows→New submenu to create new windows. Most of these items invoke dialogs which produce commands that Igor executes to create the windows.

You can also create windows by typing these commands yourself directly in the command line. For example,

```
Display yData vs xData
```

creates a graph of the wave named yData on the Y axis, versus xData on the X axis.

You can create a new window by selecting the name of a window recreation macro from the Windows menu. See **Window Macros Submenus** on page II-43.

You can also create a window using the File→Open File submenu.

Activating Windows

To activate a window, click it, or choose an item from Windows menu or its submenus.

The Recent Windows submenu shows windows recently activated. This information is saved when you save an experiment to disk and restored when you later reopen the experiment.

By default, just the window's title is displayed in the Windows menu. You can choose to display the title or the name for target windows using the Windows Menu Shows pop-up menu in the Miscellaneous section of the Miscellaneous Settings dialog.

Showing and Hiding Windows

All types of Igor windows can be hidden.

To hide a window, press Shift and choose Windows→Hide or use the keyboard shortcut Command-Shift-W (*Macintosh*) or Ctrl+Shift+W (*Windows*). You can also hide a window by pressing Shift and clicking the close button.

You can hide multiple windows at once using the Windows→Hide submenu. For example, to hide all graphs, choose Windows→Hide→All Graphs. If you press Shift while clicking the Windows menu, the sense of the menu items changes. For example, Hide→All Graphs changes to Hide→All Except Graphs.

The command window is not included in mass hides of any kind. If you want to hide it you must do so manually.

Similarly, you can show multiple windows at once using the Windows→Show submenu. For example, to show all graphs, choose Windows→Show→All Graphs. If you press Shift while clicking the Windows menu, the sense of the menu items changes. For example, Show→All Graphs changes to Show→All Except Graphs.

The Show All Except menu items do not show procedure windows and help files because there are so many of them that it would be counterproductive.

The Windows→Show→Recently Hidden Windows item shows windows recently hidden by a mass hide operation, such as Hide→All Graphs, or windows recently hidden manually (one-at-a-time using the close button or Command-Shift-W or Ctrl+Shift+W). In the case of manually hidden windows, "recently hidden" means within the last 30 seconds.

XOP windows do participate in Hide All XOP Windows and Show All XOP Windows only if XOP programmers specifically support these features.

Closing a Window

You can close a window by either choosing the Windows→Close menu item or by clicking in the window's close button. Depending on the top window's type, this will either kill or hide the window, possibly after a dialog asking for confirmation.

Killing Versus Hiding

"Killing" a window means the window is removed from the experiment. The memory used by the window is released and available for other purposes. The window's title is removed from the Windows menu.

Killing a window that represents a file on disk does not delete the file. You can also kill a window with a **KillWindow** command.

"Hiding" a window simply means the window is made invisible, but is still part of the experiment and uses the same amount of memory. It can be made visible again by choosing its title from the Windows menu.

The command window and the built-in procedure window can be hidden but not killed. All other built-in windows can be hidden or killed.

When you create a window from a procedure, you can control what happens when the user clicks the close button using the /K=<*num*> flag in the command that creates the window.

You can hide a window programmatically using the DoWindow/HIDE=1 operation.

Saving the Window Contents

Notebooks and procedure windows can be saved either in their own file, or in a packed experiment file with everything else. You can tell which is the case by choosing Notebook→Info or Procedure→Info. When you kill a notebook or a procedure window that contains unsaved information, a dialog will allow you to save it before killing the window.

Graph, table, control panel, page layout, and Gizmo windows are not saved as separate files, and are lost when you kill them unless you save a **window recreation macro** which you can execute to later recreate the window. Killing these windows and saving them as window recreation macros (stored in the built-in procedure window) frees up memory and reduces window clutter without losing any information. You can think of window recreation macros as "freeze-dried windows".

Close Window Dialogs

When you close a graph, table, layout or control panel, or Gizmo window, Igor presents a Close dialog.

If you click the Save button Igor creates a window recreation macro in the main procedure window. It sets the macro's subtype to Graph, Table, Layout, Panel, or Gizmo, so the name of the macro appears in the appropriate Macros submenu of the Windows menu. You can recreate the window using this menu.

If you don't plan to use the window again, you should click the No Save button and no window recreation macro will be created.

If you have previously created a recreation macro for the window then the dialog will have a Replace button instead of a Save button. Clicking Replace replaces the old window recreation macro with a new one. If you know that you won't need to recreate the window, you can delete the macro (see **Saving a Window as a Recreation Macro** on page II-42).

When you close a notebook or procedure window (other than the built-in procedure window), Igor presents a "hide or kill dialog".

To hide a window, press Shift while clicking the close button.

To kill a graph, table, layout, control panel, or Gizmo window without the Close dialog, press Option (*Macintosh*) or Alt (*Windows*) while clicking the close button.

If you create a window programmatically using the Display, Edit, NewLayout, NewPanel, NewNotebook, or NewGizmo operation, you can modify the behavior of the close button using the /K flag.

Saving a Window as a Recreation Macro

When you close a window that can be saved as a recreation macro, Igor offers to create one by displaying the Close Window dialog. Igor stores the window recreation macro in the main procedure window of the current experiment. The macro uses much less memory than the window, and reduces window clutter. You can invoke the window recreation macro later to recreate the window. You can also create or update a window recreation macro using the Window Control dialog.

The window recreation macro contains all the necessary commands to reconstruct the window provided the underlying data is still present. For instance, a graph recreation macro contains commands to append waves to the graph, but does not contain any wave data. Similarly, a page layout recreation macro does not contain graphs or tables or the commands to create them. The macros refer to waves, graphs and tables in the current experiment by name.

Here is how you would use recreation macros to keep a graph handy, but out of your way:

The window recreation macro is evaluated in the context of the root data folder. This detail is of consequence only to programmers. See **Data Folders and Commands** on page II-103 for more information.

You can create or replace a window recreation macro without killing the window using **The Window Control Dialog** described on page II-44. The most common reason to replace a window recreation macro is to keep the macro consistent with the window that it creates.

When Igor displays the Close Window dialog, the proposed name of the window recreation macro is the same as the name of the window. You can save the window recreation macro under a different name, if you want, by entering the new name in the dialog. If you do this, Igor creates a new macro and leaves the original macro intact. You can run the new macro to create a new version of the window or you can run the old macro to recreate the old version. This way you can save several versions of a window, while displaying only the most recent one.

Window recreation macros stay in an experiment's procedure window indefinitely. If you know that you won't need to recreate a window for which a window recreation macro exists, you can delete the macro.

To locate a window recreation macro quickly:

- Activate any procedure window, press Option (*Macintosh*) or Alt (*Windows*), and choose the window recreation macro name from the appropriate macro submenu in the Windows menu.

To delete the macro, if you're sure you won't want it again, simply select all the text from the Macro declaration line to the End line. Press Delete to remove the selected text.

See **Saving and Recreating Graphs** on page II-261 for details specific to graphs.

Window Macros Submenus

The Windows menu has submenus containing graph, table, page layout, control panel, and Gizmo recreation macros. These menus also include graph, table, and page layout style macros.

Window recreation macros are created by the Close Window and Window Control dialogs, and by the DoWindow/R command. Style macros are created by the Window Control dialog and the DoWindow/R/S command.

Igor places macros into the appropriate macro submenu by examining the macro's subtype. The subtypes are Graph, Table, Layout, Panel, Gizmo, GraphStyle, TableStyle and LayoutStyle. See **Procedure Subtypes** on page IV-193 for details.

When you choose the name of a recreation macro from a macro submenu, the macro runs and recreates the window. Choosing a style macro runs the macro which changes the target window's appearance (its "style").

However, if a procedure window is the top window and you press Option (*Macintosh*) or Alt (*Windows*) and then choose the name of any macro, Igor displays that macro but does not execute it.

The Name of a Recreated Window

When you run a window recreation macro, Igor recreates the window with the same name as the macro that created it unless there is already a window by that name. In this case, Igor adds an underscore followed by a digit (e.g. _1) to the name of the newly created window to distinguish it from the preexisting window.

Changing a Window's Style From a Macro

When you run a style macro by invoking it from the Windows menu, from the command line or from another macro, Igor applies the commands in the macro to the top window. Usually these commands

change the appearance of the window. For example, a graph style macro may change the color of graph traces or the axis tick marks.

Style macros are used most effectively with graph windows. For more information, see **Saving and Recreating Graphs** on page II-261 and **Graph Style Macros** on page II-262.

The Window Control Dialog

Choosing Control→Window Control displays the Window Control dialog which you can use to change the top window's title and name, and create or update its recreation and style macros. You can access this dialog quickly by pressing Command-Y (*Macintosh*) or Ctrl+Y (*Windows*).

You can also change the window's name. The window name is used to address the window from command line operations such as MoveWindow and also appears in the macro submenus of the Windows menu.

For more about names and titles, see **Window Names and Titles** on page II-40. Also see **Saving a Window as a Recreation Macro** on page II-42 for a discussion of window recreation macros, and see **Graph Style Macros** on page II-262 for details on style macros.

Arranging Windows

You can tile or stack windows by choosing the appropriate items from the Control submenu in the Windows menu.

You can customize the behavior of the Tile and Stack items using the Tile or Stack Windows dialog.

You can also move windows around using the MoveWindow, StackWindows, and TileWindows commands.

The Tile or Stack Windows Dialog

The Tile or Stack Windows dialog is useful for tiling a few windows or even for setting the size and position of a single window.

Select individual windows from the Windows to Arrange list, and entire classes of windows with the checkboxes.

If you want subsequent selections of the Tile (or Stack) menu item to stack the same types of windows with the same rows, columns, grout, tiling area, etc., check the "Capture as pref" checkbox. Windows selected in the Windows to Arrange menu aren't remembered by the preferences; only the window type checkboxes are remembered. There are separate settings and preferences for Stack and for Tile.

Although the TileWindows and StackWindows operations can tile and stack panels, panels don't show up here because they don't resize very well.

The Window Tiling Area subdialog specifies the area where tiling *and stacking* take place.

You can specify the tiling area in one of four ways:

- By entering screen positions in units of points
- By dragging the pictorial representation of the tiling area
- By clicking the "Use default area" button to use the default tiling area
- By positioning any non-dialog window before you enter the dialog, and clicking the "Use top window" button

Window Position and Size Management

There are four items in the Control submenu of the Windows menu that help you manage the position and size of windows.

Move to Preferred Position

Moves the active window to the position and size determined by preferences. For each type of window, you can set the preferred position and size using the Capture Prefs dialog (e.g., Capture Graph Prefs for graphs).

Move to Full Size Position

Moves and sizes the active window to display as much of the content as practical. On Macintosh, this is the same as clicking the zoom button. On Windows, the size is limited to the size of the frame window.

Retrieve Window

Moves the active window and sizes it if necessary so that all of the window is visible.

Retrieve All Windows

Moves all windows and sizes them if necessary so that all of each window is within the screen on Macintosh or within the frame on Windows. This is often useful when you open an experiment that was created on a system with a larger screen or Windows frame than yours.

Send to Back — Bring to Front

The Send to Back item in the Windows menu sends the top window to the bottom of the desktop, behind all other windows. This function can also be accessed by pressing Command-' (*Macintosh*) or Ctrl+E (*Windows*). After sending a window behind, you can bring it to the front by choosing Bring to Front or by pressing Command-Shift-' (*Macintosh*) or Ctrl+Shift+E (*Windows*). You can also press these keys repeatedly to cycle through all windows.

You can also send a window to the back with the `DoWindow/B` command and bring it to the front with the `DoWindow/F` command.

Text Windows

Igor Pro displays text in procedure, notebook, and Igor help windows as well as in the command and history areas of the command window. This section discusses behavior common to all of these windows.

Executing Commands

You can execute commands selected in a notebook, procedure or help window by pressing Control-Enter or Control-Return. You can also execute selected commands by Control-clicking (*Macintosh*) or right-clicking (*Windows*) and choosing Execute Selection.

For more on this, see **Notebooks as Worksheets** on page III-4.

Text Window Navigation

The term "keyboard navigation" refers to selection and scrolling actions that Igor performs in response to the arrow keys and to the Home, End, Page Up, and Page Down keys. Macintosh and Windows have different conventions for these actions in windows containing text. You can use either Macintosh or Windows conventions on either platform.

By default, Igor uses Macintosh conventions on Macintosh and Windows conventions on Windows. You can change this using the Keyboard Navigation menu in the Misc Settings section of the Miscellaneous Settings dialog. If you use Macintosh conventions on Windows, use Ctrl in place of the Macintosh Command key. If you use Windows conventions on Macintosh, use Command in place of the Windows Ctrl key.

Macintosh Text Window Navigation

Key	No Modifier	Option	Command
Left Arrow	Move selection left one character	Move selection left one word	Move selection to start of line
Right Arrow	Move selection right one character	Move selection right one word	Move selection to end of line
Up Arrow	Move selection up one line	Move selection up one paragraph	Move selection up one screen

Macintosh Text Window Navigation

Key	No Modifier	Option	Command
Down Arrow	Move selection down one line	Move selection down one paragraph	Move selection down one screen
Home	Scroll to start of document	Scroll to start of document	Not used
End	Scroll to end of document	Scroll to end of document	Not used
Page Up	Scroll up one screen	Scroll up one screen	Not used
Page Down	Scroll down one screen	Scroll down one screen	Not used

Windows Text Window Navigation

Key	No Modifier	Ctrl
Left Arrow	Move selection left one character	Move selection left one word
Right Arrow	Move selection right one character	Move selection right one word
Up Arrow	Move selection up one line	Move selection up one paragraph
Down Arrow	Move selection down one line	Move selection down one paragraph
Home	Move selection to start of line	Move selection to start of document
End	Move selection to end of line	Move selection to end of document
Page Up	Scroll up 1 screen	Scroll up 1 screen
Page Down	Scroll down 1 screen	Scroll down 1 screen

Finding Text in the Active Window

You can find text active windowby choosing Edit→Find or by pressing Command-F (*Macintosh*) or Ctrl+F (*Windows*). This displays the Find bar.

On Macintosh, you can search for the next occurrence of a string by selecting the string and choosing Edit→Use Selection For Find or pressing Command-E to enter the selected text as the find string and Command-G to find it.

On Windows, you can search for the next occurrence of a string by selecting the string and choosing Edit→Find Selection or pressing Ctrl+H to enter the selected text as the find string and find it.

Find and Replace

You can find and replace text by choosing Edit→Replace Text or by pressing Command-R (*Macintosh*) or Ctrl+R (*Windows*). This displays the Find and Replace bar.

Here is another method for finding and replacing text:

1. Move the selection to the top of the active window.
2. Use Edit→Find to find the first instance of the target string.
3. Manually change the first instance, then copy the new text to the Clipboard.
4. Press Command-G (*Macintosh*) or Ctrl+G (*Windows*) to find the next occurrence.
5. Press Command-V (*Macintosh*) or Ctrl+V (*Windows*) to paste.
6. Repeat steps 4 and 5 until done.

Finding Text in Multiple Windows

You can find text in multiple windows by choosing Edit→Find Text in Multiple Windows. This displays the Find Text in Multiple Windows window which allows you to search all help windows, all procedure windows, and all notebooks.

You can also use the Igor Help Browser to search in multiple files, including files that are not open in your current experiment. See **The Igor Help Browser** on page II-2 for details.

Text Magnification

You can magnify the text in any window to make it bigger or smaller to suit your taste.

In help windows, procedure windows, plain text notebooks, and formatted text notebooks, you can use the magnifying glass icon in the bottom-left corner of the window. You can also use the Magnification submenu in the contextual menu for the window. To display the contextual menu, Control-click (*Macintosh*) or right-click (*Windows*) in the body of the window.

You can also set the magnification for the command line, history area, and the debugger. These areas do not display the magnifying glass icon so you must use the contextual menu.

You may notice some anomalies when you use text magnification. For example, in a formatted text notebook, text may wrap at a different point in the paragraph and may change in relation to tab stops. This happens because fonts are not available in fractional sizes and because the actual width of text does not scale linearly with font size.

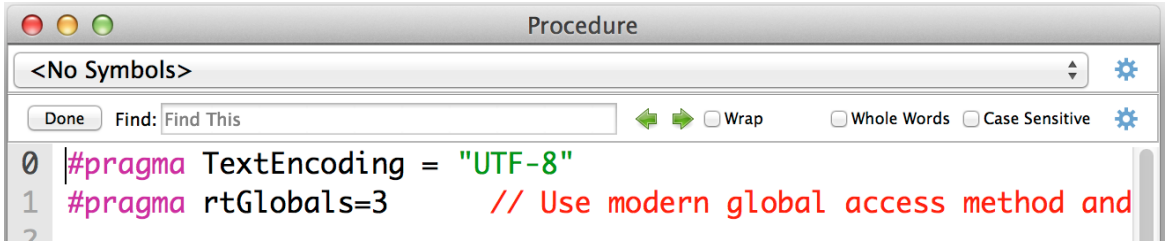
You can set the default magnification for each type of text area by choosing a magnification from the Magnification popup menu and then choosing Set As Default from the same popup menu. Any text areas whose magnification is set to Default will use the newly specified default magnification. For example, if you want text in all help files to appear larger, open any help file, choose a larger magnification, 125% for example, and then choose Set As Default For Help Files. All help files whose current magnification is set to Default will be updated to use the new default.

The default magnification for the command line and history area controls the magnification that will be used the next time you launch Igor Pro.

The magnification setting is saved in formatted notebooks and help files only. If you change the magnification setting for one of these files and then save and close the file, the magnification setting will be restored when you reopen the file. For all other types of text areas, including procedure windows and plain text notebooks, the magnification setting is not stored in the file. If you close and reopen such a file, it will reopen using the default magnification for that type of text area.

The Find Bar

The find bar is available in help, procedure and notebook windows, and other places where you might need to search for text. To display it, choose Edit→Find or press Command-F (Macintosh) or Ctrl+F (Windows). The Find Bar appears at the top of the host window:



Enter text in the Find edit box and press the right arrow icon to find the next occurrence of the text. Press the left arrow icon to find the preceding occurrence.

You can also find the next occurrence by pressing Command-G (Macintosh) or Ctrl+G (Windows). You can also find the preceding occurrence by pressing Command-Shift-G (Macintosh) or Ctrl+Shift+G (Windows).

You can set the text to be found by selecting it in the active window and pressing Command-E (Macintosh) or Ctrl+H (Windows).

By default, when you use these keyboard equivalents, if the find bar is not showing, Igor displays it. If you prefer that Igor not display the find bar when you use these equivalents, click the gear icon and check the Show Find Bar on Edit→Find Only checkbox.

You can set the size of the find bar using the gear icon.

To hide the find bar, click the Done button or press the Esc key.

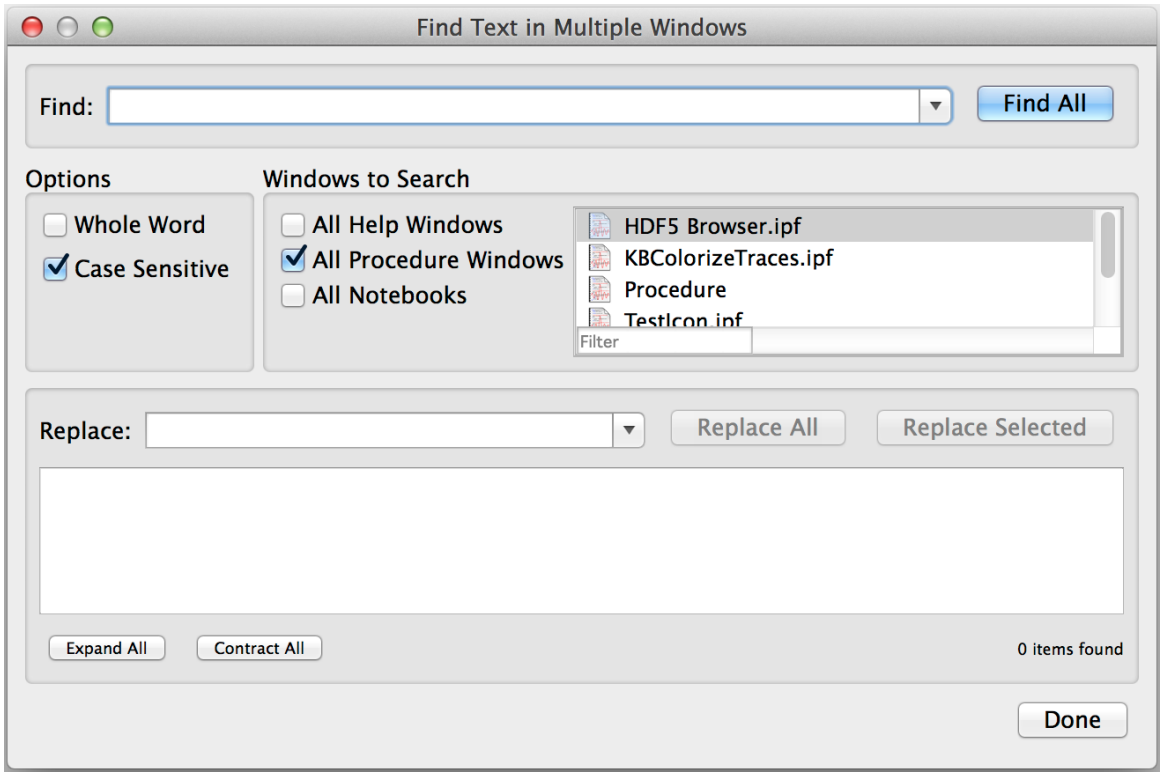
Search and Replace

If the window you are working on contains editable text, you can choose Edit→Replace or press Command-R (Macintosh) or Ctrl+R (Windows) to display the replace text portion of the find bar:

While replacing text is undoable, the potential for unintended and wide-ranging consequences is such that we recommend saving the file before doing a mass replace using Replace All so you can revert-to-saved if necessary.

The Find Text in Multiple Windows Dialog

You can perform a Find on multiple help, procedure and notebook windows at one time by choosing Edit→Find in Multiple Windows or by pressing Command-Shift-F (Macintosh) or Ctrl+Shift+F (Windows). This invokes the Find Text in Multiple Windows dialog:



Enter text to be found in the Find box. Use the checkboxes to select the types of windows to search. To narrow the list of files to search, you can enter a filter string in the Filter box at the bottom of the window list. The list then shows only list items whose names contain the filter string.

When you click the Find All button, Igor commences searching all the text in the listed windows. During the search, Igor displays a small progress dialog showing how far through the window list the search has gotten. You can stop the search by clicking the Cancel button.

For each window in which the text is found, an entry appears in the panel at the bottom of the dialog. A number in parentheses indicates how many instances of the search text were found in that window.

Each of these window entries can be opened using the disclosure control at the left end of the item. Then a snippet of text around the search string is shown, with the search string highlighted. The number at the left end of the found text snippet is the line number within the searched window.

Double-clicking a text snippet item takes you to the found text in the window containing the text.

Search and Replace in Multiple Windows

After a search is finished, you can use the found text items to replace all instances of the text that were found in editable windows. Enter the replacement text in the Replace edit box.

Clicking the Replace All button replaces all the found instances with the replacement text.

NOTE: Replace All in multiple windows cannot be undone. Use it with great care.

If you make a mistake, choose File→Revert Experiment, File→Revert Notebook, or File→Procedure and make sure you have reverted all affected windows. This is the only way to recover.

You can limit the replacement by selecting a subset of the found text items and clicking the Replace Selected button. Click a found text item to select it. Shift-click another item to select all items between that item and the one previously clicked. To select non-contiguous items, Command-click (Macintosh) or Ctrl-click (Windows). You can also click and drag over multiple items. It is still a good idea to save before replacing.

Using an External Text Editor

Igor Pro 7 supports the use of external editors for editing Igor procedure files and plain text notebooks. This allows you to use your favorite text editor rather than Igor for editing plain text files if you prefer. This is mostly intended for use by advanced programmers who are accustomed to using external editors in other programming environments.

In the past, Igor kept plain text files open as long as the corresponding procedure or notebook window was open in Igor. This interfered with the use of external editors.

In Igor Pro 7, a procedure or plain text notebook window opens its file just long enough to read the text into memory and then close it. If you modify the text in Igor and do a save, Igor reopens the file, writes the modified text to it, and closes the file.

If you modify the file using an external text editor instead of Igor, Igor notices the change, reopens the file, reloads the modified text into memory, and closes the file.

Supporting external editors creates issues that Igor must deal with:

- If you modify the file in an external editor, the text is now out of sync with the text in Igor's window. In this case, Igor notices that the file has been changed and either reloads the text into memory or notifies you, depending on your external editor miscellaneous settings.
- If you save modifications to the file in an external editor and also edit the document in Igor, the text in Igor is in conflict with the external file. In this case Igor informs you of the conflict and lets you choose which version to keep.
- If you move the disk file to new location, delete the file or rename it, Igor's information about the

file is no longer valid. In this case Igor notices that the file has disappeared and gives you options for dealing with it.

The following sections explain how Igor deals with these issues in more detail.

A Plain Text File Has Been Modified Externally

Once per second Igor checks every procedure and plain text notebook window to see if its file has been modified externally. If so then it either automatically reloads the file into memory or notifies you about the external modification and allows you to reload the file at your leisure. The factory default behavior is to automatically reload externally modified files and to check for modified files only when Igor is the active application.

You can control Igor's behavior using the Miscellaneous Settings Dialog. Choose Misc→Miscellaneous Settings, click Text Editing in the lefthand pane, and click the External Editor tab.

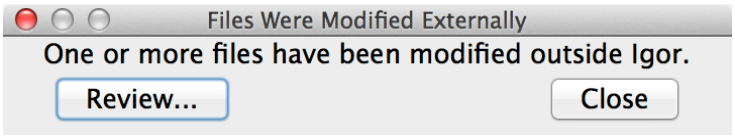
If you select Ask Before Reloading, two things happen when the file is modified externally:

- Igor displays a **Reload** button in the status area at the bottom of the document window:

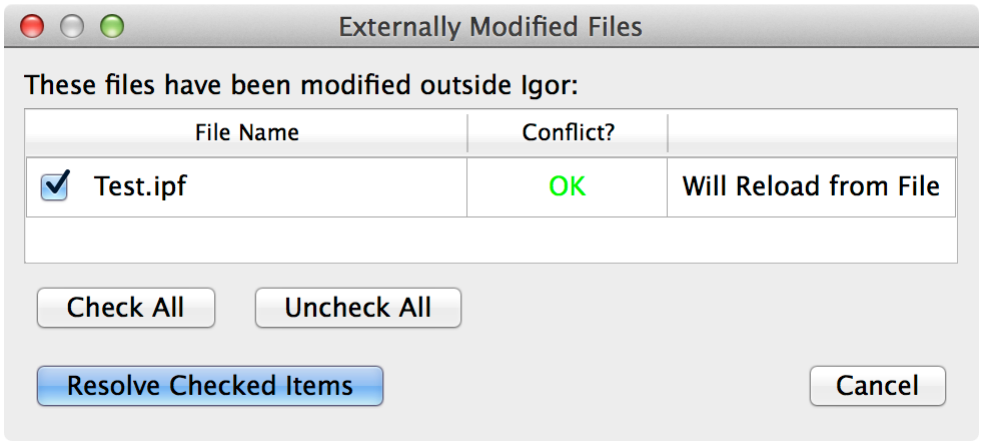
  The Reload button will be visible only if the window is visible.

- Igor displays a small floating notification window:



Clicking the **Reload** button causes Igor to reload the file's text.

Clicking the **Review** button in the notification window displays a dialog in which you can review all the files that are currently modified externally:



Click **Resolve Checked Items** to reload the file into memory.

A Plain Text File Has Been Modified Externally and In Igor

When a file has been modified both in an external editor and by editing in an Igor window, we say it is "in conflict". Igor never automatically reloads a file that is in conflict.

When a file is in conflict, Igor displays a **Resolve Conflict** button in the status area in the Igor document window. Clicking that button brings up a dialog giving four choices to deal with the conflict:

- Reload External Changes

  Loads the contents of the file into memory, discarding changes made in Igor.

- Adopt Document

  Severs the tie between the document and the file, keeping the Igor changes and discarding the changes to the file. To learn more about adopting a file, see **Adopting Notebook and Procedure Files** on page II-23.

- Close the Window

  Closes the procedure or notebook window in Igor, discarding changes made in Igor.

- Save Igor Changes

  Writes changes made in Igor to the file. This resolves the conflict in Igor and creates a conflict in the external editor. Different external editors have different approaches to such conflicts.

Files that are in conflict also cause the notification window to appear. Clicking the Review button in the notification window displays a dialog which gives you the same options for resolving the conflict.

Editing a File with External Modifications

If a file has been modifed externally and not re-loaded in Igor, typing in the Igor window creates a conflict even if there wasn't one previously. This may be undesirable, so when Igor detects this situation, it displays a dialog, similar to the Resolve Conflict dialog, asking you what should be done. The choices are:

- Allow Typing

This choice tells Igor that you want to be allowed to modify the document in Igor. You are putting off resolving the conflict for later. You will have to decide at some point to resolve the conflict in one of the ways described above.

- Reload External Changes

The external modifications will be loaded into the Igor window. Modifications in Igor's copy are discarded.

- Adopt Document

Severs the tie between the document and the file. To learn more about adopting a file, see Adopting Notebook and Procedure Files.

You also have the option of clicking the Cancel button. In that case, the situation is not resolved, and any further attempt to type in the window will cause the dialog to be displayed again.

A Plain Text File Is Missing

If the file associated with an Igor plain text document window is moved, renamed or deleted, Igor will note that the file is missing. In this case, Igor displays a **File Missing** button in the status area in the Igor document window. Clicking that button displays a File Missing dialog giving four choices to deal with the conflict:

- Adopt Document

  Severs the tie between the document and the file. To learn more about adopting a file, see Adopting Notebook and Procedure Files.

- Close the Window

  Closes the procedure or notebook window in Igor.

- Find Missing File

  Displays an Open File dialog allowing you to locate the moved or renamed file.

- Save As

  Displays a Save As dialog allowing you to save the document to a new file.

Missing or Modified Text Files Dialog

A plain text file (procedure file or plain text notebook) may be missing because you deleted it, renamed it, or moved it, or it was deleted, renamed or moved by a program.

A plain text file may have been modified by another program if you edited it in an external editor, for example.

If you try to do a save while a plain text file is missing or modified, Igor displays the Missing or Modified file dialog which asks if you want to cancel the save or continue it. Usually you should cancel the save and use the Files Were Modified Externally window, which appears in the top/right corner of the screen, to review and address the situation.

If you elect to continue the save, it is likely that an error will occur.

To prevent an unattended procedure from hanging, the check for missing or modified files is not done if the save is invoked from a procedure.

Window Shortcuts

Action	**Shortcut** (*Macintosh*)	**Shortcut** (*Windows*)
To close a window	Click the close button or press Command-W.	Click the close button or press Ctrl+W.
To kill a window with no dialog	Press Option and click the close button or press Command-Option-W.	Press Alt and click the close button or press Ctrl+Alt+W.
To hide a window	Press Shift and click the close button or press Command-Shift-W.	Press Shift and click the close button or press Ctrl+Shift+W.
To invoke the Window Control dialog	Press Command-Y.	Press Ctrl+Y.
To send the top window behind all others	Press Control-Command-E.	Press Ctrl+E.
To bring the bottom window on top of all others	Press Shift-Control-Command-E.	Press Ctrl+Shift+E.
To send all windows that are completely visible behind all others	Press Command-Option-E.	Press Ctrl+Alt+E.
To activate a recently activated window	Press Command, click the main menu bar, and select from the Recent menu.	Press Ctrl, click the main menu bar, and select from the Recent menu.
To move a window to its preferred size and position	Click the zoom button.	Press Alt and click the maximize button.
To move a window to its full-size position	Press Shift-Option and click the zoom button.	Press Shift +Alt and click the maximize button.
To activate the command window	Press Command-J.	Press Ctrl+J.
To clear the command buffer	Press Command-K.	Press Ctrl+K.
To open the built-in procedure window	Press Command-M.	Press Ctrl+M.
To cycle through all procedure windows	Press Command-Option-M.	Press Ctrl+Alt+M.
To open the Igor help browser	Choose Help→Igor Help Browser.	Press F1.
To find a phrase in a text window	Press Command-F.	Press Ctrl+F.
To find the same phrase again	Press Command-G.	Press Ctrl+G.
	Press Command-Shift-G to search backward.	Press Ctrl+Shift+G to search backward.
To find the selected phrase	Press Command-E and Command-G.	Press Ctrl+H.
	Press Command-E and Command-Shift-G to search backward.	Press Ctrl+Shift+H to search backward.
	This shortcut can be changed through the Miscellaneous Settings dialog.	