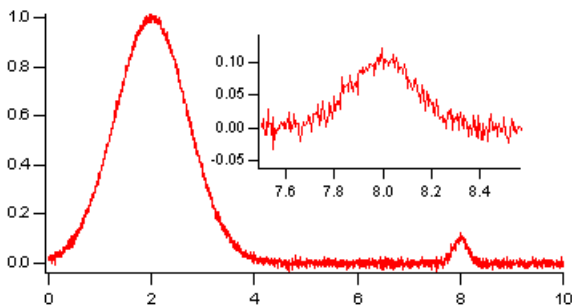


Embedding and Subwindows

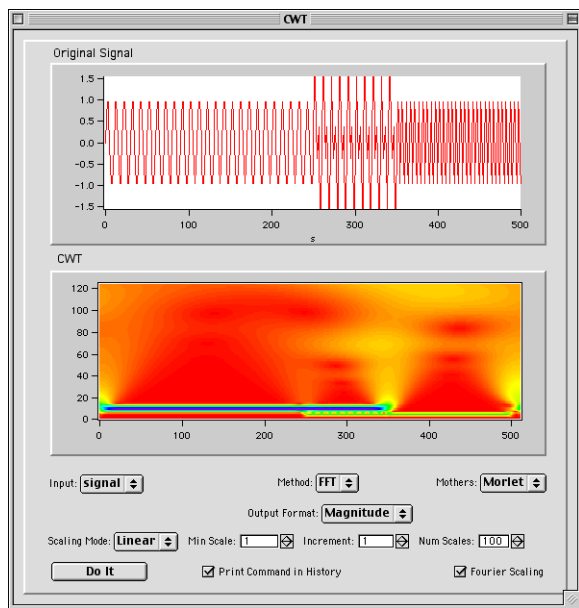
Overview	86
Subwindow Terminology	86
Restrictions.....	87
Creating Subwindows.....	88
Positioning and Guides.....	88
Frames	89
Subwindow User-Interface Concepts	89
Subwindow Layout Mode and Guides	90
Layout Mode and Guide Tutorial	91
Graph Control Bars and Subpanels.....	93
Page Layouts and Subwindows	94
Notebooks as Subwindows in Control Panels.....	94
Subwindow Command Concepts.....	95
Subwindow Syntax.....	95
Subwindow Sizing.....	95
Subwindow Operations and Functions.....	96

Overview

You can embed graphs, tables, and control panels into other graph, control panel, and page layout windows. In addition, you can embed notebooks in control panels only. The embedded window is called a *subwindow* and the enclosing window is called the *host*. Subwindows may be nested in a hierarchy of arbitrary depth. The top host window in the hierarchy is known as the *base*. In the following example, the smaller, inset graph is a subwindow:



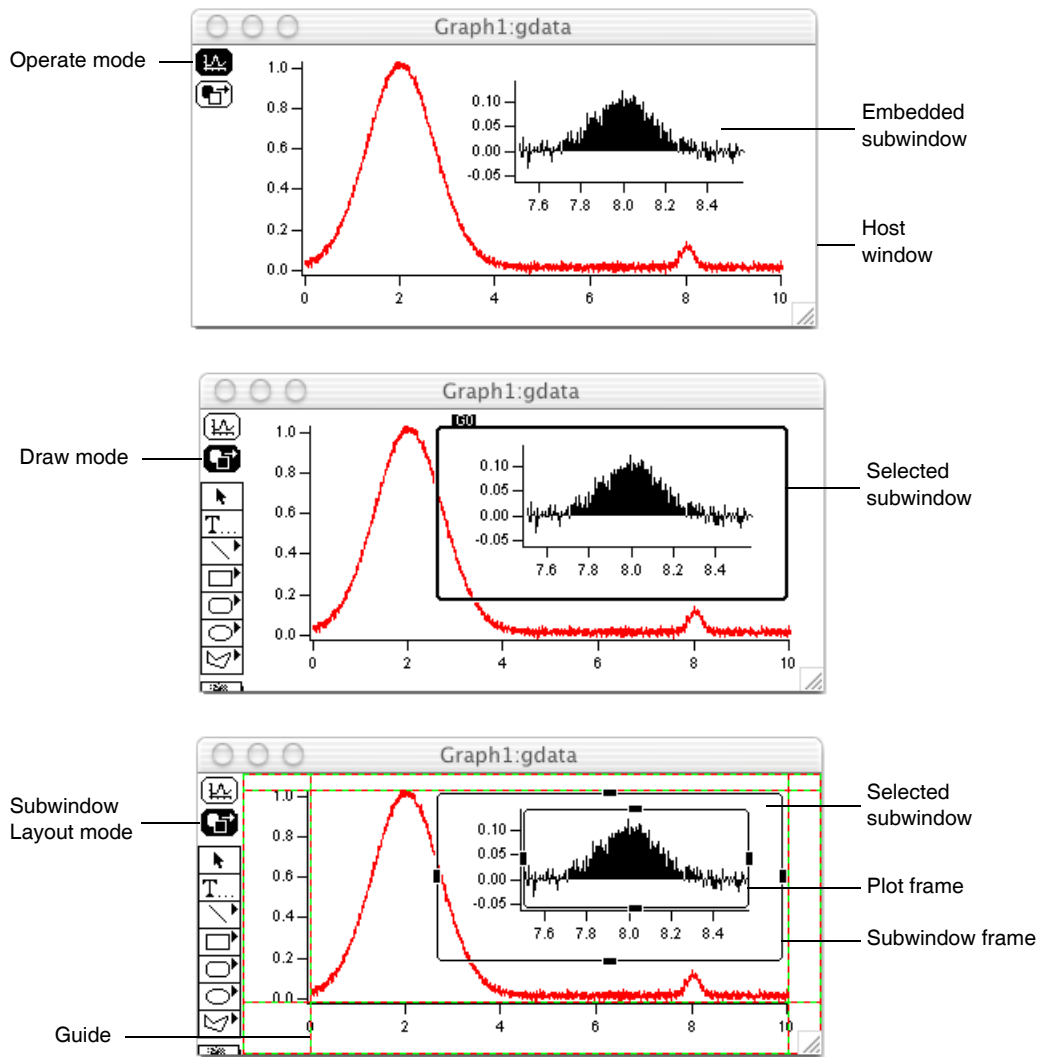
Although you can create graphs like this by careful positioning of free axes, it is much easier to accomplish using embedding. In the next example, the two graphs are subwindows embedded in a host panel:



This example is derived from the CWT demo experiment which you can find in the Analysis section of your Examples folder.

Subwindow Terminology

When a window is inserted into another window it is said to be *embedded*. In some configurations (see **Restrictions** on page III-87), an embedded window does not support the same functionality that it has as a standalone window. It is then called a *presentation-only* object. For example, when a table is embedded in a panel, it has scroll bars and data entry features just like a standalone table. But when a table is embedded in a graph or in a page layout, it is a presentation-only object with no scroll bars or other user interface elements.



Restrictions

The following table summarizes the rules for allowed host and embedded subwindow configurations.

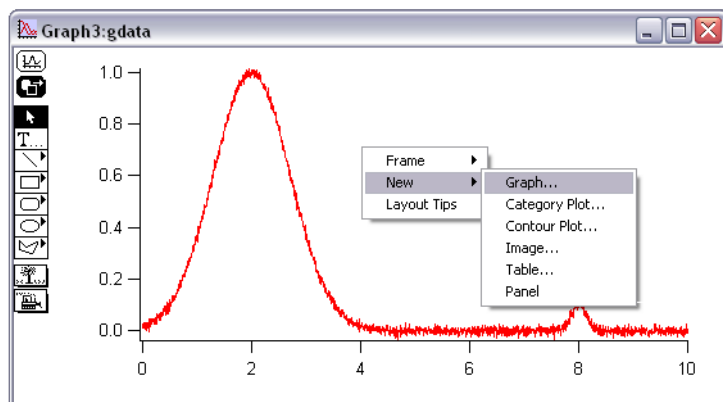
		Host			
		Graph	Table	Panel	Layout
Subwindow	Graph	Yes	No	Yes	Yes
	Table	Yes*	No	Yes	Yes*
	Panel	Yes [†]	No	Yes	No
	Layout	No	No	No	No
	Notebook	No	No	Yes	No

* Tables embedded in a graphs or layouts are presentation-only objects. They do not support editing of data.

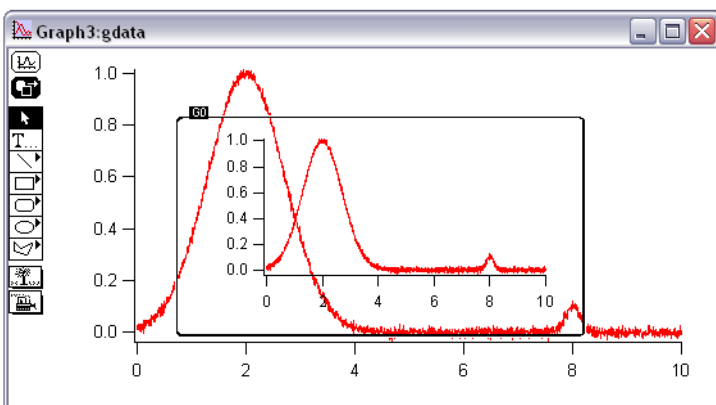
† Panels can be embedded in base graphs only.

Creating Subwindows

You can create subwindows either from the command line (see **Subwindow Command Concepts** on page III-95) or interactively using contextual menus. To add a subwindow interactively, add tools to the target window (Show Tools menu item), click the lower icon to enter Drawing mode, and then right-click (*Windows*) or Control-click (*Macintosh*) in the interior of the window and choose the desired type of subwindow from the New menu:



You will be presented with the standard dialog for creating a new window but the result will be a subwindow:



You can position the subwindow by clicking on its heavy frame to enter Subwindow-Layout mode (see **Subwindow Layout Mode and Guides** on page III-90). Finally, click the top icon of the tools to adjust the graphs using Operate (Normal) mode.

Positioning and Guides

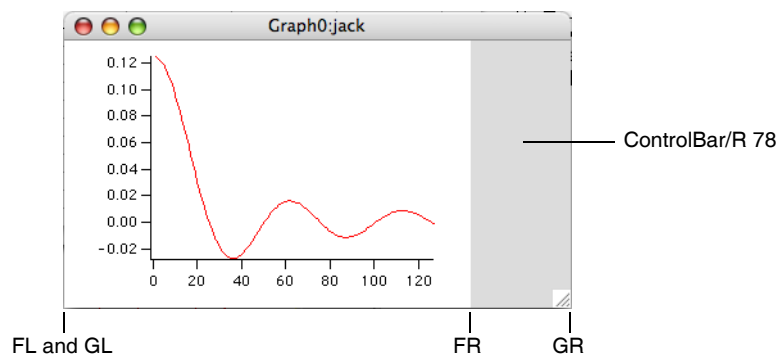
Subwindows may be positioned in their hosts in a wide variety of ways. You can specify the position of a subwindow numerically using either absolute (fixed distance) or relative modes. You can also attach key locations in a subwindow to named guides.

Guides are horizontal or vertical reference locations defined by the immediate host of a subwindow and may be either fixed (built-in) or moveable (user-defined). Built-in guides represent fixed locations of the host such as its frame or the interior plot area of graphs. Built-in guides can not be moved except by moving the object to which the guide refers.

All host windows have built-in guides named *FL*, *FT*, *FR*, and *FB* for Frame Left, Frame Top, Frame Right, and Frame Bottom. Graphs also have the corresponding *PL*, *PR*, *PT*, and *PB* for the interior plot area. In addition, base graphs (top level host graph windows) have built-in guides *GL*, *GR*, *GT*, and *GB* for the Graph area.

	Left	Right	Top	Bottom
Host Window Frame	FL	FR	FT	FB
Host Graph Rectangle	GL	GR	GT	GB
Inner Graph Plot Rectangle	PL	PR	PT	PB

The graph area is the total area of the graph window excluding the areas occupied by the tool palette and the cursor information panel. The frame area is the total area of the graph window excluding the areas occupied by the tool palette, the cursor information panel and the control bar.



User-defined guides may be based on built-in or other user-defined guides. They may be defined as being either a fixed distance from a guide or a relative distance between two guides.

Reference points of a subwindow that may be attached to guides include the outer left, right, top and bottom for all subwindow types and, for graphs only, the interior plot area.

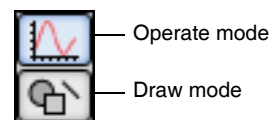
Guides are especially useful when creating stacked graphs. By attaching the plot left (PL) location on each graph to a user-defined guide, all left axes will be lined up and will move in unison when you drag the guide around. This is illustrated in **Layout Mode and Guide Tutorial** on page III-91.

Frames

You may specify a frame style for each subwindow. Frames, if any, are drawn inside the rectangle that defines the location of the subwindow and the normal content is then inset by the frame thickness. Frames may also be specified for base graph and panel windows. This is handy when you want to include a frame when you export or print a graph. You can adjust the frame for a window or subwindow using a contextual menu (right-click (*Windows*) or Control-click (*Macintosh*)).

Subwindow User-Interface Concepts

Each host window has two main modes corresponding to the top two icons in the window's toolbar. Choose Show Tools from the Graph or Panel menu to show the toolbar in which clicking the top icon selects Operate (Normal) mode and clicking the second icon selects Drawing mode.



When using subwindows, there is a third mode: Subwindow Layout (see **Subwindow Layout Mode and Guides** on page III-90).

When not using subwindows, a particular window is the *target window* — the default window for command-line commands that do not explicitly specify a window. The addition of subwindows leads to the analogous concept of the *active subwindow*.

You make a subwindow the active subwindow by clicking it. In Operate mode the active subwindow is indicated by a yellow and black border. In Drawing mode it is indicated by a heavy black border with the name of the subwindow shown in the upper left corner.

Chapter III-4 — Embedding and Subwindows

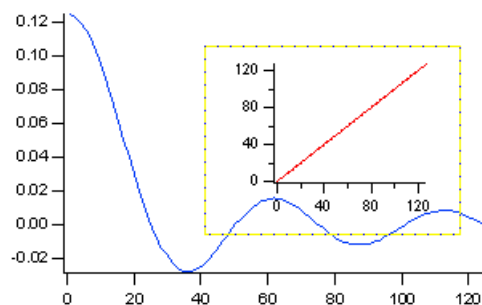
Panel subwindows are exceptions in that clicking them in Operate mode does not make them the active subwindow. You must click them while in Drawing mode.

As an example, execute the following:

```
Make/O jack=sin(x/8)/x, sam=x
Display jack
Display/W=(0.5,0.14,0.9,0.7)/HOST=# sam
```

Notice the yellow and black border around the newly created subwindow:

This indicates that it is the active subwindow. Now double click on the curve in the host window (but not within the subwindow border). Change the color of the trace *jack* to blue and notice that the subwindow is no longer active. Now move the mouse over the subwindow and notice that the cursor changes to the usual shapes corresponding to the parts of the graph that it is hovering over. Drag out a selection rectangle in the plot area of the subwindow and notice that the Marquee pop-up menu is available for use on the subwindow and that the subwindow has been activated. Depending on your actions in a window, Igor activates subwindows as appropriate and generally you do not have to be aware of which subwindow is active.

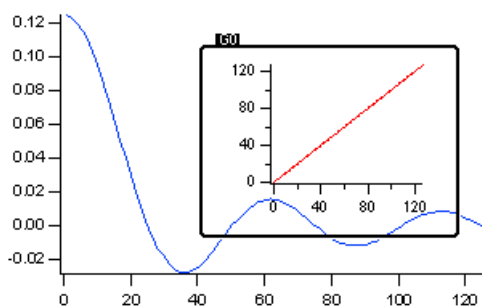


Choose Show Tools from the Graph menu and notice that the tools are provided by the main window. Tools and the cursor information panel are hosted by the base window but may apply to subwindows.

Click the drawing icon in the tool bar.

Notice the subwindow, which had been indicated as active using the yellow/black border, now has a heavy frame.

You are now in a mode where you can use drawing tools on the subwindow. To draw in the main window, click outside the subwindow to make the host window active.



When in Operate mode, the main menu bar always references the base window. In order to make changes to a subwindow of a different type, you can use a context click to access a menu specific to the subwindow. In Drawing mode, the main menu references the active subwindow. For example, if the base window is a graph with an embedded table, the menu bar contains a Graph menu when in Operate mode. However, when the embedded table is selected in Drawing mode, the main menu bar contains a Table menu.

When in Drawing mode, you can right-click (*Windows*) or Control-click (*Macintosh*) to get a pop-up menu from which you can choose frame styles and insert new subwindows or delete the active subwindow. Deleting a subwindow is not undoable.

The info box (see **Info Box and Cursors** on page II-277) in a graph targets the active subgraph. You can not simultaneously view or move cursors in two different subgraphs.

Subwindow Layout Mode and Guides

To layout one or more subwindows in a host window, enter Drawing mode, click the selector (arrow) tool and click in a subwindow. A heavy frame will be drawn with the name of the subwindow in the upper left. Now click on the frame to enter subwindow layout mode. In this mode, the subwindow is drawn with a light frame with handles in the middle of each side. In addition, built-in and user guides are drawn as dashed lines.

A subwindow can be moved by dragging its frame and resized using the handles. If a handle is positioned near a guide, it will snap in place and attach itself to the guide. However, if one or more handles are attached to guides and then the subwindow is moved using the frame, all attachments will be deleted.

A graph subwindow is drawn using two frames. The inner frame represents the plot area of the graph. Its handles can be attached to guides to allow easy alignment of multiple graph subwindows.

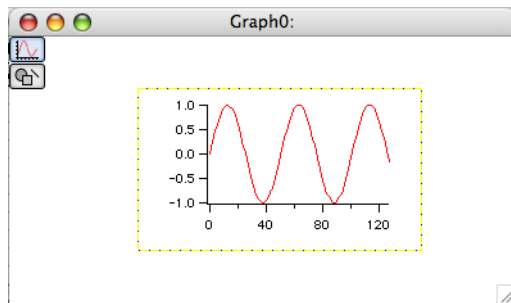
You can create user-defined guides by pressing Alt (*Windows*) or Option (*Macintosh*) and then click-dragging an existing guide. By default, the new guide will be a fixed distance from its parent. You can convert the new guide to relative mode (where the guide is specified as a fraction of the distance between two guides) by right-clicking (*Windows*) or Control-click (*Macintosh*) on the new guide and then choosing a partner guide from the “make relative to” list of other guides. You can also use the right-click menu to convert a relative guide to fixed or to delete a guide, if it is not in use.

Layout Mode and Guide Tutorial

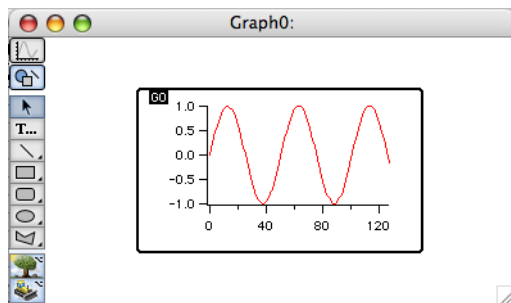
In a new experiment, execute these commands:

```
Make/O jack=sin(x/8), sam=cos(x/8)
Display
Display/HOST=# jack
ShowTools
```

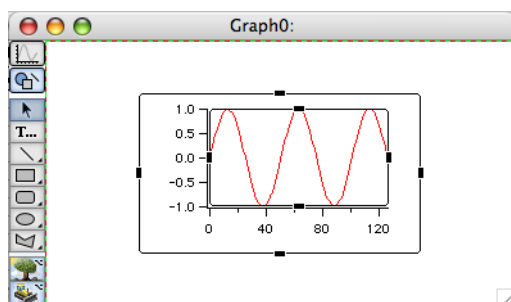
The first `Display` command created an empty graph and the second inserted a subgraph. We used the command line just to get going quickly. The graph is in Operate mode and it looks like this:



Click the lower icon in the tool bar to enter Drawing mode and notice the subwindow is drawn with a black frame with the name of the subwindow (G0):



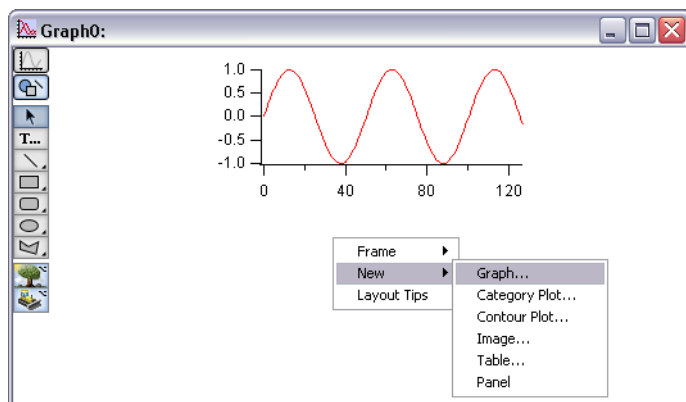
Use the arrow tool to click on the black frame around the subgraph. You are now in Subwindow Layout mode as indicated by the two rectangles with handles on each edge of the graph.



Chapter III-4 — Embedding and Subwindows

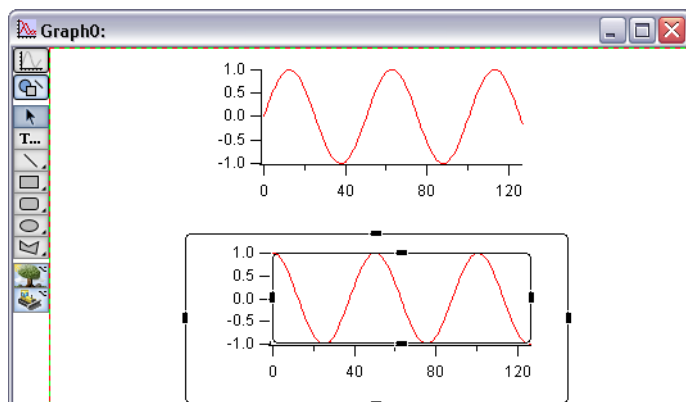
Position the mouse over the outer rectangle until the cursor changes to a four-headed arrow. Drag the subwindow up as high as it will go and then drag the bottom handle up to just above the halfway point so that the subgraph is in the upper half of the window.

Click outside the subwindow to leave Subwindow Layout mode and then click again to select the main (empty) graph as the active subwindow. Right-click (*Windows*) or Control-click (*Macintosh*) below the subgraph and choose the New→Graph menu item:



Pick `Sam` as the Y wave in the resulting dialog and click Do It. This creates a new subwindow and makes it active. Click on the heavy frame to enter Subwindow Layout mode for the new subgraph and position it in the lower half of the window.

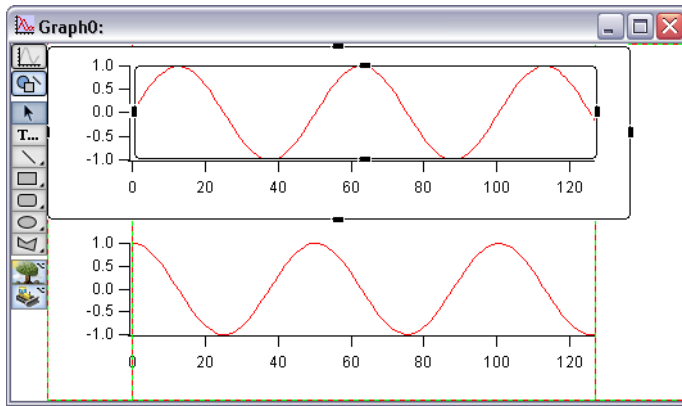
While still in Subwindow Layout mode for the second graph, notice the red/green dashed lines around the periphery. These are fixed guides and are properties of the base window. Hold down `Alt` (*Windows*) or `Option` (*Macintosh*) and move the mouse over the left hand dashed line. When you notice the cursor changing to a two headed arrow, click and drag to the right about 3 cm to create a user-defined guide. Use the same technique to create another user-defined guide based on the right edge also inset by about 3 cm:



Move the mouse over the new guides and notice the cursor changes to a two headed arrow indicating they can be moved.

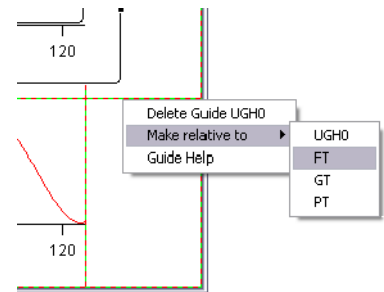
While still in Subwindow Layout mode for the second graph, click in the black handle centered on the left axis and drag the handle over the position of the left user guide. Notice that it snaps into place when it is near. Release the mouse button and use the same technique to connect the right edge of the interior plot area to the right user guide.

Now place the top subgraph in Subwindow Layout mode and connect its left and right plot area handles to the user guides:

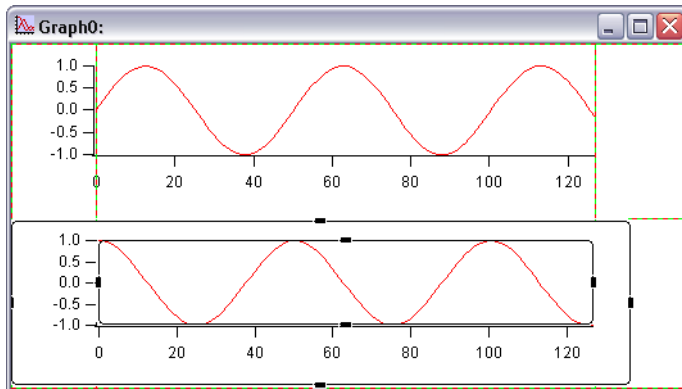


While still in Subwindow Layout mode, drag the user guides around and notice that both graphs follow.

The two guides we created are of a type that is a fixed distance from another (the frame left (FL) and frame right (FR) in this case). We will now create a relative guide. Hold down *Alt* (*Windows*) or *Option* (*Macintosh*) and move the mouse over the bottom dashed line near the window frame. When you notice the cursor changing to a two headed arrow, click and drag up to about the middle of the graph to create another user-defined guide. Position the mouse over the new guide, right-click (*Windows*) or Control-click (*Macintosh*), and choose *Make Relative to*→*FT* from the menu.



Now, as you resize the window, the guide will remain at the same relative distance between the bottom (FB) and the top (FT). Use the handles to attach the bottom of the top graph to the new guide and then put the bottom graph into Subwindow Layout mode and attach its top to the guide:



Graph Control Bars and Subpanels

Prior to Igor Pro 5, you could use the arrow tool in a graph to drag down a control bar from the top of the window. You would then use this area for buttons. Now you can drag out control areas from all four sides of a graph window. You do this by entering Drawing mode and clicking just inside an outer edge of the window. Subpanels are automatically created and anchored to appropriate guides. If you want to create an Igor Pro 4-compatible control area with no subpanel, you can either delete the automatically generated subpanel or you can use the *ControlBar* command.

To adjust the size of a control bar, be sure the subpanel is not currently active to avoid putting it into layout mode by a click near the frame. Unlike most other draggable objects in Igor, the cursor does not change shape to indicate it is over the control area margins.

Page Layouts and Subwindows

Subwindows in Page Layouts can be a bit confusing because you use two different modes to position conventional graph and table objects versus subwindows. In the normal mode for a page layout (top icon in the toolbar), you can adjust the positioning of conventional layout objects but not subwindows because they are in operate mode.

In operate mode, subwindow graphs act just like conventional graph windows and allow marquee expansion, double clicking on graph elements to bring up dialogs, dragging textboxes and axes and other normal graph behavior. But to adjust the position of subwindows, you have to enter Subwindow Layout (see **Subwindow Layout Mode and Guides** on page III-90).

Although a bit confusing, the use of subwindows in Page Layouts is very useful because the Page Layout is then self-contained and need not refer to other windows. An additional feature of subwindows is the ability to see more detail when you expand the layout above 100%. Conventional layout objects are simply expanded images originally taken at 100% while subwindows are actually drawn at the increased resolution.

Here are a few reasons to use the conventional window or object layout method rather than subwindows:

- You can use cursors and buttons in a full graph but not in a subgraph.
- You can place the same graph in multiple layouts.
- You can have a graph window be a different (and more convenient) size than the layout object.

In a page layout, you can insert a graph subwindow or table by first using the marquee tool to specify the desired location and then using the pop-up menu available in the interior of the marquee to choose one of several subwindow types. If you need to reposition the new subwindow, you will need to enter Drawing mode and use the selector (arrow) tool of the drawing palette, not the layout arrow tool.

You can convert conventional layout graph and table objects to subwindows via the contextual menu for the object. In operate mode (select the top icon in the layout tool panel), Control-click or right-click anywhere on the layout object and choose Convert Graph/Table To Embedded. Portions of a graph that are not allowed in layouts, such as buttons and subpanels, will be lost in the conversion.

Similarly, you can convert a subgraph or subtable to a conventional window and layout object via the contextual menu. In operate mode, Control-click or right-click and choose Convert To Graph/Table and Object. In a graph you must click in an area free of traces or axes, such as in the graph margin, to get the correct pop-up menu.

Notebooks as Subwindows in Control Panels

You can create a notebook subwindow in a control panel using the NewNotebook operation. A notebook subwindow might be used to present status information to the user or to permit the user to enter multi-line text. Here is an example:

```
NewPanel /W=(150, 50, 654, 684)
NewNotebook /F=1 /N=nb0 /HOST=# /W=(36, 36, 393, 306)
Notebook # text="Hello World!\r"
```

The notebook subwindow can be plain text (/F=0) or formatted text (/F=1).

By default, the notebook ruler is hidden and the notebook status area width is set to zero when a notebook subwindow is created. You can change this using the Notebook operation.

To make it easier to use for text input or display, when a formatted text notebook subwindow is first created and when you resize the width of the subwindow, Igor automatically adjusts the Normal ruler's right indent so that all of the text governed by the Normal ruler fits in the subwindow. This adjustment is done for the Normal ruler only. Other rulers, including Normal+ (variations of Normal) rulers, are not adjusted.

You can programmatically insert text in the notebook using the **Notebook** operation.

If you create a window recreation macro for the control panel, by default the contents of the notebook subwindow are saved in the recreation macro. If you later run the macro to recreate the control panel, the notebook sub-

window's contents will be restored. This also applies to experiment recreation which automatically uses window recreation macros.

If you do not want the contents of the notebook subwindow to be preserved in the recreation macro, you must disable the autosave property, like this:

```
Notebook Panel0#nb0, autosave=0
```

When you create a window recreation macro while autosave is on, it will contain commands that look something like this:

```
Notebook kwTopWin, zdata="GaqDU%ejN7!Z)ts!+J\\\.F^>EB"
Notebook kwTopWin, zdata= "jmRiCVsF?/]21,HG<k,\"@i1,&\\\.F^>EB"
Notebook kwTopWin, zdataEnd=1
```

The Notebook `zdata` command sends to the notebook encoded binary data in an Igor-private format that represents the contents of the notebook when the recreation macro was created. In real life, there would be a number of `zdata` commands, one after the other, which cumulatively define the contents of the notebook. The notebook accumulates all of the `zdata` text. The `zdataEnd` command causes the notebook to decode the binary data and use it to restore the notebook's contents.

When you save an experiment containing a control panel, a window recreation macro is created for you by Igor and when you open the experiment, Igor runs the recreation macro to recreate the control panel. If autosave is off, after saving and reopening the experiment, the notebook will be empty. If autosave is on, the window recreation macro will include `zdata` and `zdataEnd` commands that restore the contents of the notebook subwindow.

The encoded binary data includes a checksum. If the Notebook `zdata` commands have been altered, the checksum will fail and you will receive an error when the Notebook `zdataEnd` command executes.

Subwindow Command Concepts

All operations that create window types that can be subwindows can take a `/HOST=hcSpec` flag in order to create a subwindow in a specific host. In addition, operations and functions that can modify or operate on a subwindow can affect a specific subwindow using the `/W=hcSpec` flag (for operations) or an `hcSpec` as a string parameter (for functions).

Subwindow Syntax

The Command Line syntax for identifying a subwindow for a command is summarized in this table.

Subwindow Specification	Location
<code>#</code>	Active subwindow.
<code>##</code>	Host of active subwindow.
<code>#sub1 [#sub2 [#...]]</code>	Subwindow within active subwindow.
<code>baseName#sub1 [#sub2 [#...]]</code>	Absolute path from base host window.
<code>baseName</code>	Base host window.

The window “path” uses the # symbol at the beginning of each subwindow name identified in the window-subwindow hierarchy. For each subwindow following the first one, the subwindow names lower in the hierarchy must be contained in brackets.

Subwindow Sizing

When `/HOST` is used in conjunction with **Display**, **NewPanel**, **NewWaterfall**, **NewImage** and **Edit** commands to create a subwindow, the values used with the window size `/W= (a, b, c, d)` flag can have one of two different meanings. If all the values are less than 1.0, then the values are taken to be fractional relative

Chapter III-4 — Embedding and Subwindows

to the host's frame. If any of the values are greater than 1.0, then they are taken to be fixed locations measured in points or, for panels, pixels relative to the top left corner of the host.

Guides may override the numeric positioning set by /W. All operations supporting /HOST may take the /FG=(*gleft, gtop, gright, gbottom*) flag where *gleft-gbottom* are the names of built-in or user-defined guides. FG stands for Frame Guide and this flag specifies that the outer frame of the subwindow is attached to the guides. A * character may be substituted for a name to indicate that the default value should be used.

The inner plot area of a graph subwindow may be attached to guides using the analogous PG flag. Thus a subgraph may need up to three specifications. For example:

```
Display/host=#/W=(0,10,400,200)/FG=(FL,*,FR,*)/PG=(PL,*,PR,*) sam
```

When the subwindow position is fully specified using guides, the /W flag is not needed but it is OK to include it anyway.

Subwindow Operations and Functions

Here are the main operations and functions that will be useful in dealing with subwindows. For full documentation, see Chapter V-1, **Igor Reference**.

ChildWindowList (*hostName*)

DefineGuide [/W=*winName*] *newGuideName* = {[*guideName1*, val
[, *guideName2*]]} [,...]

KillWindow *winNameStr*

MoveSubwindow [/W=*winName*] *key* = (*values*)[, *key* = (*values*)]...

RenameWindow *oldName*, *newName*

SetActiveSubwindow *subWinName*