See Also
Chapter II-8, **Data Folders**.

RenamePath

RenamePath ***oldName, newName***
The RenamePath operation renames an existing symbolic path from *oldName* to *newName*.

See Also
Symbolic Paths on page II-22

RenamePICT

RenamePICT ***oldName, newName***
The RenamePICT operation renames an existing picture to from *oldName* to *newName*.

See Also
Pictures on page III-509.

RenameWindow

RenameWindow ***oldName, newName***
The RenameWindow operation renames an existing window or subwindow from *oldName* to *newName*.

Parameters
oldName is the name of an existing window or subwindow.

When identifying a subwindow with *oldName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.

See Also
The **DoWindow** operation.

ReorderImages

ReorderImages [**/W=*winName***] ***anchorImage*, {*imageA*, *imageB*, …}**
The ReorderImages operation changes the ordering of graph images to that specified in the braces.

Flags

/W=*winName* — Reorders images in the named graph window or subwindow. When omitted, action will affect the active window or subwindow. This must be the first flag specified when used in a Proc or Macro or on the command line.

When identifying a subwindow with *winName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.

Details
Igor keeps a list of images in a graph and draws the images in the listed order. The first image drawn is consequently at the bottom. All other images are drawn on top of it. The last image is the top one; no other image obscures it.

ReorderImages works by removing the images in the braces from the list and then reinserting them at the location specified by *anchorImage*. If *anchorImage* is not in the braces, the images in braces are placed before *anchorImage*.

If the list of images is A, B, C, D, E, F, G and you execute the command

```
ReorderImages F, {B,C}
```

images B and C are placed just before F: A, D, E, **B**, **C**, **F**, G.

The result of

```
ReorderImages E, {D,E,C}
```

is to reorder C, D and E and put them where E was. Starting from the initial ordering this gives A, B, **D**, **E**, **C**, F, G.