Specify which window to close using either the /NAME or /FILE flag. You must use one or the other. Usually you would use /NAME, as it is usually more convenient. If by some chance two procedures have the same name, /FILE can be used to distinguish between them.

You cannot call CloseProc on a nonmain procedure window that someone has had the bad taste to call "Procedure".

See Also

Chapter III-13, **Procedure Windows**.

The **Execute/P** operation.

cmplx

cmplx(*realPart*, *imagPart*)

The cmplx function returns a complex number whose real component is *realPart* and whose imaginary component is *imagPart*.

Use this to assign a value to a complex variable or complex wave.

Examples

Assume wave1 is complex. Then:

```
wave1(0) = cmplx(1,2)
```

sets the Y value of wave1 at x=0 such that its real component is 1 and its imaginary component is 2.

Assuming wave2 and wave3 are real, then:

```
wave1 = cmplx(wave2,wave3)
```

sets the real component of wave1 equal to the contents of wave2 and the imaginary component of wave1 equal to the contents of wave3.

You may get unexpected results if the number of points in wave2 or wave3 differs from the number of points in wave1. If wave2 or wave3 are shorter than wave1, the last element of the short wave is copied repeatedly to fill wave1.

See Also

conj, **imag**, **magsqr**, **p2rect**, **r2polar**, and **real** functions.

CmpStr

CmpStr(*str1*, *str2* [, *flags*])

The CmpStr function returns zero if *str1* is equal to *str2* or non-zero otherwise.

Parameters

flags controls the type of comparison that is done. It defaults to 0 and supports the following values:

0:	Case-insensitive text comparison
1:	Case-sensitive text comparison
2:	Binary comparison

Binary comparison (*flags*=2) was added Igor Pro 7.05 and is appropriate if *str1* and *str2* contain binary data, which may contain null bytes (bytes with the value 0), rather than human-readable strings.

Details

If *flags* is omitted, 0, or 1, CmpStr does a text comparison and returns the following values:

-1:	*str1* is alphabetically before *str2*
0:	*str1* and *str2* are equal
1:	*str1* is alphabetically after *str2*

The alphabetic order represented by -1 and 1 is valid only for ASCII text. It is not valid for non-ASCII text, such as text containing accented characters.