See **Saving Experiments** on page II-16 for a discussion of the various experiment file formats.

Selector = 16

IgorInfo(16) returns the total number of waves of all kinds (global, free, local) that currently exist. This selector value was added in Igor Pro 9.00.

IgorInfo(16) can help advanced Igor programmers detect wave leaks in their procedures. For details, see **Detecting Wave Leaks** on page IV-206.

Examples

```
Print NumberByKey("NSCREENS", IgorInfo(0))     // Number of active displays

Function RunningWindows()                      // Returns 0 if Macintosh, 1 if Windows
    String platform = UpperStr(IgorInfo(2))
    Variable pos = strsearch(platform,"WINDOWS",0)
    return pos >= 0
End
```

IgorVersion

#pragma IgorVersion = *versNum*

When a procedure file contains the directive, `#pragma IgorVersion=`*versNum*, an error will be generated if *versNum* is greater than the current Igor Pro version number. It prevents procedures that use new features added in later versions from running under older versions of Igor in which these features are missing. However, this version check is limited because it does not work with versions of Igor older than 4.0.

See Also

The **The IgorVersion Pragma** on page IV-54 and **#pragma**.

IgorVersion

The IgorVersion function returns version number of the Igor application as a floating point number. Igor Pro 8.00 returns 8.00, as does Igor Pro 8.00A.

Details

You can use IgorVersion in conditionally compile code expressions, which can be used to omit calls to new Igor features or to provide backwards compatibility code.

```
#if (IgorVersion() >= 8.00)
    [Code that compiles only on Igor Pro 8.00 or later]
#else
    [Code that compiles only on earlier versions of Igor]
#endif
```

If at all possible, it is better to require your users to use a later version of Igor rather than writing conditional code. Attempting this kind of backward-compatibility multiplies your testing requirements and the chances for bugs.

See Also

IgorInfo, **Conditional Compilation** on page IV-108, **The IgorVersion Pragma** on page IV-54

ilim

ilim

The ilim function returns the ending loop count for the inner most iterate loop Not to be used in a function. iterate loops are archaic and should not be used.

imag

imag(*z*)

The imag function returns the imaginary component of the complex number *z* as a real (not complex) number.

See Also

The **cmplx**, **conj**, **p2rect**, **r2polar**, and **real** functions.