"Elapsed" means that the hour is a number from -9999 to 9999. The result for hours outside that range is undefined.

The *fracDigits* parameter is optional and specifies the number of digits of fractional seconds. The default value is 0. The *fracDigits* parameter is ignored for format=0, 1, 2,and 4.

Examples

```
Print Secs2Time(DateTime,0)                 // prints 1:07 PM
Print Secs2Time(DateTime,1)                 // prints 1:07:28 PM
Print Secs2Time(DateTime,2)                 // prints 13:07
Print Secs2Time(DateTime,3)                 // prints 13:07:29
Print Secs2Time(30*60*60+45*60+55,4)        // Prints 30:45
Print Secs2Time(30*60*60+45*60+55,5)        // Prints 30:45:55
```

See Also

For a discussion of how Igor represents dates, see **Date/Time Waves** on page II-85.

The **Secs2Date**, **date**, **date2secs** and **DateTime** functions. Also, **Operators** on page IV-6 for ?: details.

SelectNumber

SelectNumber(*whichOne*, *val1*, *val2* [, *val3*])

The SelectNumber function returns one of *val1*, *val2*, or (optionally) *val3* based on the value of *whichOne*.

SelectNumber(*whichOne*, *val1*, *val2*) returns *val1* if *whichOne* is zero, else it returns *val2*.

SelectNumber(*whichOne*, *val1*, *val2*, *val3*) returns *val1* if *whichOne* is negative, *val2* if *whichOne* is zero, or *val3* if *whichOne* is positive.

Details

SelectNumber works with complex (or real)*val1*, *val2*, and *val3* when the result is assigned to a complex wave or variable. (Print expects a real result, see the "causes error" example, below).

If *whichOne* is NaN, then NaN is returned.

whichOne must always be a real value.

Unlike the ? : conditional operator, SelectNumber always evaluates all of the numeric expression parameters *val1*, *val2*, …

SelectNumber works in a macro, whereas the conditional operator does not.

Examples

```
Print SelectNumber(0,1,2)                       // prints 1
Print SelectNumber(0,1,2,3)                     // prints 2
wv=SelectNumber(numtype(wv[p])==2,wv[p],0)      // replace NaNs with zeros

// chooses among complex values
Variable/C cx= SelectNumber(negZeroPos,cmplx(-1,-1),0,cmplx(1,1))

// causes error because Print expects a real value (not complex)
Print SelectNumber(negZeroPos,cmplx(-1,-1),0,cmplx(1,1))

// The real function expects a complex result
Print real(SelectNumber(negZeroPos,cmplx(-1,-1),0,cmplx(1,1)))
```

See Also

The **SelectString** and **limit** functions, and **Waveform Arithmetic and Assignments** on page II-74. Also, **Operators** on page IV-6 for details about the ?: operator.

SelectString

SelectString(*whichOne*, *str1*, *str2* [, *str3*])

The SelectString function returns one of *str1*, *str2*, or (optionally) *str3* based on the value of *whichOne*.

SelectString(*whichOne*, *str1*, *str2*) returns *str1* if *whichOne* is zero, else it returns *str2*.

SelectString(*whichOne*, *str1*, *str2*, *str3*) returns *str1* if *whichOne* is negative, *str2* if *whichOne* is zero, or *str3* if *whichOne* is positive.