Details

The main uses for saving a table as a packed experiment are to save an archival copy of data or to prepare to merge data from multiple experiments (see **Merging Experiments** on page II-19). The resulting experiment file preserves the data folder hierarchy of the waves displayed in the table starting from the "top" data folder, which is the data folder that encloses all waves displayed in the table. The top data folder becomes the root data folder of the resulting experiment file. Only the table and its waves are saved in the packed experiment file, not variables or strings or any other objects in the experiment.

SaveTableCopy does not know about dependencies. If a table contains a wave, wave0, that is dependent on another wave, wave1 which is not in the table, SaveTableCopy will save wave0 but not wave1. When the saved experiment is open, there will be a broken dependency.

The main use for saving as a tab or comma-delimited text file is for exporting data to another program.

When calling SaveTableCopy from a procedure, you should call DoUpdate before calling SaveTable copy. This insures that the table is up-to-date if your procedure has redimensioned or otherwise changed the number of points in the waves in the table.

SaveTableCopy sets the variable V_flag to 0 if the operation completes normally, to -1 if the user cancels, or to another nonzero value that indicates that an error occurred. If you want to detect the user canceling an interactive save, use the /Z flag and check V_flag after calling SaveTableCopy.

The **SaveData** operation also has the ability to save a table to a packed experiment file. SaveData is more complex but a bit more flexible than SaveTableCopy.

Examples

This function saves all tables to a single tab-delimited text file.

```
Function SaveAllTablesToTextFile(pathName, fileName)
    String pathName         // Name of an Igor symbolic path.
    String fileName

    String tableName
    Variable index

    index = 0
    do
        tableName = WinName(index, 2)
        if (strlen(tableName) == 0)
            break
        endif

        SaveTableCopy/P=$pathName/W=$tableName/T=1/A=1 as fileName

        index += 1
    while(1)
End
```

See Also

SaveGraphCopy, **SaveGizmoCopy**, **SaveData**, **Merging Experiments** on page II-19

sawtooth

sawtooth(*num*)

The sawtooth function returns ((*num* +n2π) mod 2π)/2π where n is used to correct if *num* is negative. Sawtooth is used to create arbitrary periodic waveforms like sine and cosine.

Examples

```
wave1 = sawtooth(x)
```

creates a sawtooth in wave1 whose Y values range from 0 to 1 as its X values go through 2π units.

```
wave1 = exp(sawtooth(x))
```

creates a series of exponentials in wave1 of amplitude exp(1) and period 2π.

You can also use sawtooth to create periodic repetitions of a given part of a wave:

```
wave1 = wave2(sawtooth(x))
```

creates a periodic repetition of wave2 in wave1 given the correct X scaling for the waves.