The /FILT=*fileFilterStr* flag provides control over the file filter menu in the Open File dialog. This flag was added in Igor Pro 7.00. The construction of the *fileFilterStr* parameter is the same as for the /F=*fileFilterStr* flag of the Open operation. See **Open File Dialog File Filters** on page IV-149 for details.

In Igor7 and later, the *macFilterStr* and *winFilterStr* parameters of the /I flag are ignored. Use the /FILT flag instead.

**Output Variables**

GBLoadWave sets the following output variables:

| | |
|---|---|
| `V_flag` | Number of waves loaded or -1 if an error occurs during the file load. |
| `S_fileName` | Name of the file being loaded. |
| `S_path` | File system path to the folder containing the file. |
| `S_waveNames` | Semicolon-separated list of the names of loaded waves. |

S_path uses Macintosh path syntax (e.g., "`hd:FolderA:FolderB:`"), even on Windows. It includes a trailing colon.

When GBLoadWave presents an Open File dialog and the user cancels, V_flag is set to 0 and S_fileName is set to "".

**Example**

```
// Load 128 point single precision version 2 Igor binary wave file
GBLoadWave/S=126/U=128 "fileName"

// Load 8 256 point arrays of 16 bit signed integers into single-precision waves
// after skipping 128 byte header
GBLoadWave/S=128/T={16,2}/W=8/U=256 "fileName"

// Load n 100 point arrays of double-precision floating point numbers
// into double-precision Igor waves with names like temp0, temp1, etc,
// overwriting existing waves. n is determined by the number of bytes
// in the file.
GBLoadWave/O/N=temp/T={4,4}/U=100 "fileName"

// Load a file containing a 1024 byte header followed by a 512 row
// by 384 column array of unsigned bytes into an unsigned byte matrix
// wave and display it as an image
GBLoadWave/S=1024/T={8+64,8+64}/N=temp "fileName"
Rename temp0, image
Redimension/N=(512,384) image
if (<file uses row-major order>)
       MatrixTranspose image
   endif
Display; AppendImage image
```

"Row-major order" relates to how a 2D array is stored in memory. In row-major order, all data for a given row is stored contiguously in memory. In column-major order, all data for a given column is stored contiguously in memory. Igor uses column-major order but row-major is more common.

**See Also**

**Loading General Binary Files** on page II-166.

**FBinRead** operation for more complex applications such as loading structured data into Igor structures.

# gcd

**gcd(*A*, *B*)**

The gcd function calculates the greatest common divisor of *A* and *B*, which are both assumed to be integers.

**Examples**

Compute least common multiple (LCM) of two integers:

```
Function LCM(a,b)
   Variable a, b

   return((a*b)/gcd(a,b))
End
```