

GBLoadWave can load one or more 1D arrays from a file. When multiple arrays are loaded, they can be stored sequentially in the file or they can be interleaved. Sequential means that all of the points of one array appear in the file followed by all of the points of the next array. Interleaved means that point zero of each array appears in the file followed by point one of each array.

GBLoadWave And Very Big Files

Most data files are not so large as to present major issues for GBLoadWave or Igor. However, if your data file approaches hundreds of millions or billions of bytes, size and memory issues may arise.

If you want GBLoadWave to convert the type of the data, for example from 16-bit signed to 32-bit floating point, this requires an extra buffer during the load process which takes more memory.

When dealing with extremely large files, you may need to load part of your data file into Igor at a time using the GBLoadWave /S and /U flags.

The Load General Binary Dialog

When you choose Data→Load Waves→Load General Binary File, Igor displays the Load General Binary dialog. This dialog allows you to choose the file to load and to specify the data type of the file and the data type of the wave or waves to be created.

A few of the items in the dialog require some explanation.

The Number of Arrays in File textbox and the Number of Points in Array textbox are both initially set to 'auto'. Auto means that GBLoadWave automatically determines these based on the number of bytes in the file.

If you leave both on auto, GBLoadWave assumes that there is one array in the file with the number of points determined by the number of bytes in the file and the data length of each point.

If you set Number of Arrays in File to a number greater than zero and leave Number of Points in Array on auto, GBLoadWave determines the number of points in each array based on the total number of bytes in the file and the specified number of arrays in the file.

If you set Number of Points in Array to a number greater than one and leave Number of Arrays in File on auto, GBLoadWave determines the number of arrays in the file based on the total number of bytes in the file and the specified number of points in each array.

You can also specify the number of arrays in the file and the number of points in each array explicitly by entering a number in place of 'auto' for each of these settings.

GBLoadWave creates one or more 1D waves and gives the waves names which it generates by appending a number to the specified base name. For example, if the base name is "wave", it creates waves with names like wave0, wave1, etc.

If the Overwrite Existing Waves checkbox is checked, GBLoadWave uses names of existing waves, overwriting them. If it is unchecked, GBLoadWave skips names already in use.

Checking the Apply Scaling checkbox allows you to specify an offset and multiplier so that GBLoadWave can scale the data into meaningful units. If this checkbox is unchecked, GBLoadWave does no scaling.

VAX Floating Point

GBLoadWave can load VAX "F" format (32 bit, single precision) and "G" format (64 bit, double precision) numbers.

Do not use the GBLoadWave byte-swapping feature (/B flag) for VAX data. This does Intel-to-Motorola byte swapping, also called little-endian to big-endian. VAX data is byte-swapped relative to the way Igor stores data, but not in the same sense. Specifically, each 16-bit word is big-endian but each 8-bit byte is little-endian. When you specify that the input data is VAX data, using /J=2, GBLoadWave does the swapping required for VAX data.