Importing Data

Most Igor users create waves by loading data from a file created by another program. The process of loading a file creates new waves and then stores data from the file in them. Optionally, you can overwrite existing waves instead of creating new ones. The waves can be numeric or text and of dimension 1 through 4.

Igor provides a number of different routines for loading data files. There is no single file format for numeric or text data that all programs can read and write.

There are two broad classes of files used for data interchange: text files and binary files. Text files are usually used to exchange data between programs. Although they are called text files, they may contain numeric data, text data or both. In any case, the data is encoded as plain text that you can read in a text editor. Binary files usually contain data that is efficiently encoded in a way that is unique to a single program and can not be viewed in a text editor.

The closest thing to a universally accepted format for data interchange is the “delimited text” format. This consists of rows and columns of numeric or text data with the rows separated by carriage return characters (CR - *Macintosh*), linefeed return characters (LF - *Unix*), or carriage return/linefeed (CRLF - *Windows*) and the columns separated by tabs or commas. The tab or comma is called the “delimiter character”. The CR, LF, or CRLF characters are called the “terminator”. Igor can read delimited text files written by most programs.

FORTRAN programs usually create fixed field text files in which a fixed number of bytes is used for each column of data with spaces as padding between columns. The Load Fixed Field Text routine is designed to read these files.

Text files are convenient because you can create, inspect or edit them with any text editor. In Igor, you can use a notebook window for this purpose. If you have data in a text file that has an unusual format, you may need to manually edit it before Igor can load it.

Text files generated by scientific instruments or custom programs often have “header” information, usually at the start of the file. The header is not part of the block of data but contains information associated with it. Igor’s text loading routines are designed to load the block of data, not the header. The Load General Text routine can usually automatically skip the header. The Load Delimited Text and Load Fixed Field Text routines needs to be told where the block of data starts if it is not at the start of the file.

An advanced user could write an Igor procedure to read and parse information in the header using the Open, FReadLine, StrSearch, sscanf and Close operations as well as Igor’s string manipulation capabilities. Igor includes an example experiment named Load File Demo which illustrates this.

If you will be working on a Macintosh, and loading data from files on a PC, or vice-versa, you should look at **File System Issues** on page III-450.

The following table lists the data loading routines available in Igor and their salient features.

File Type	Description
Delimited text	Created by spreadsheets, database programs, data acquisition programs, text editors, custom programs. This is the most commonly used format for exchanging data between programs. Row Format: `<data><delimiter><data><terminator>` Contains one block of data with any number of rows and columns. A row of column labels is optional. Can load numeric, text, date, time, and date/time columns. Can load columns into 1D waves or blocks into 2D waves. Columns may be equal or unequal in length. See **Loading Delimited Text Files** on page II-129.

File Type	Description
Fixed field text	Created by FORTRAN programs.
	Row Format: <data><padding><data><padding><`terminator`>
	Contains one block of data with any number of rows and columns.
	Each column consists of a fixed number of bytes including any space characters which are used for padding.
	Can load numeric, text, date, time and date/time columns.
	Can load columns into 1D waves or blocks into 2D waves.
	Columns are usually equal in length but do not have to be.
	See **Loading Fixed Field Text Files** on page II-137.
General text	Created by spreadsheets, database programs, data acquisition programs, text editors, custom programs.
	Row Format: `<number><white space><number><terminator>`
	Contains one or more blocks of numbers with any number of rows and columns. A row of column labels is optional.
	Can not handle columns containing non-numeric text, dates and times.
	Can load columns into 1D waves or blocks into 2D waves.
	Columns must be equal in length.
	Igor's Load General Text routine has the ability to automatically skip nonnumeric header text.
	See **Loading General Text Files** on page II-138.
Igor Text	Created by Igor, custom programs. Used mostly as a means to feed data and commands from custom programs into Igor.
	Format: See **Igor Text File Format** on page II-151.
	Can load numeric and text data.
	Can load data into waves of dimension 1 through 4.
	Contains one or more wave blocks with any number of waves and rows.
	Consists of special Igor keywords, numbers and Igor commands.
	See **Loading Igor Text Files** on page II-150.
Igor Binary	Created by Igor, custom programs. Used by Igor to store wave data.
	Each file contains data for one Igor wave of dimension 1 through 4.
	Format: See Igor Technical Note #003, "Igor Binary Format".
	See **Loading Igor Binary Data** on page II-154.
Image	Created by a wide variety of programs.
	Format: Always binary. Varies according to file type.
	Can load JPEG, PNG, TIFF, BMP, Sun Raster graphics files.
	Can load data into matrix waves, including TIFF image stacks.
	See **Loading Image Files** on page II-157.
General binary	General binary files are binary files created by other programs. If you understand the binary file format, it is possible to load the data into Igor.
	See **Loading General Binary Files** on page II-166.