The characters represented by the ASCII codes 0x00 to 0x7F are called "ASCII characters" with all other characters called "non-ASCII characters". All byte-oriented text encodings that you are likely to encounter are supersets of ASCII.

Multiple-byte text encodings, such as Shift-JIS and UTF-8, typically use one byte for ASCII characters, and for some frequently-used non-ASCII characters, and two or more bytes for other non-ASCII characters.

In order to properly display and process text, a program must convert any text it reads from outside sources (e.g., files) into whatever text encoding the program uses for internal storage if the source text encoding and the internal text encoding are different.

Since all byte-oriented text encodings that you are likely to encounter are supersets of ASCII, ASCII characters do not require conversion when treated as any byte-oriented text encoding. Non-ASCII characters require conversion and problems arise if the conversion is not done or not done correctly.

Text Encodings Commonly Used in Igor

In Igor the most commonly-used text encodings are:

- MacRoman (Macintosh Western European)
- Windows-1252 (Windows Western European)
- Shift JIS (Japanese)
- UTF-8 (Unicode using a byte-oriented encoding form)

Igor6 and earlier versions of Igor stored text in "system text encoding". For western users this means MacRoman on Macintosh and Windows-1252 on Windows. For Japanese users it means Shift JIS on both platforms.

Igor now uses UTF-8 exclusively.

The system text encoding is determined by the system locale. The system locale determines the text encoding used by non-Unicode programs such as Igor6. It has no effect on Igor7 or later.

On Macintosh you set the system locale through the preferred languages list in the Language & Region preference panel. On Windows you set it through the Region control panel. Most users never have to set the system locale because their systems are initially configured appropriately for them.

Because Igor now uses UTF-8 internally, when it loads a pre-Igor7 file, it must convert the text from system text encoding to UTF-8. If the file contains ASCII text only this conversion is trivial. But if the file contains non-ASCII characters, such as accented letters, Greek letters, math symbols or Japanese characters, issues may arise, as explained in the following sections.

In order to convert text to UTF-8, Igor must know the text encoding of the file from which it is reading the text. This is tricky because plain text files, including computer program source files, data files, and plain text documentation files, usually provide no information that identifies the text encoding they use. As explained below, Igor employs several strategies for determining a file's text encoding so that it can correctly convert the file's text to UTF-8 for internal storage and processing.

Western Text Encodings

MacRoman and Windows-1252 (also called "Windows Latin 1") are used for Western European text and are collectively called "western".

In MacRoman and Windows-1252, the numeric code for each character is stored in a single byte. A byte can represent 256 values, from 0x00 to 0xFF. The 128 values 0x00 through 0x7F represent the same characters as ASCII in both text encodings. The remaining 128 values, from 0x80 to 0xFF, represent different characters in each text encoding. For example the value 0xA5 in MacRoman represents a bullet character while in Windows-1252 it represents a yen symbol.