

Converting to UTF-8

In Igor Pro 7 and later, Igor uses UTF-8 text encoding, a form of Unicode, to store text.

Non-Unicode text encodings, including MacRoman, Windows-1252, and Shift JIS, which were used in Igor Pro 6 and before, are antiquated. During the transition from Igor6 to Igor7, backward compatibility with Igor6 was needed so that Igor7 users could share files with Igor6 users. Starting with Igor9, we operate on the premise that compatibility with Igor6 is no longer needed.

Although non-Unicode text encodings are antiquated, Igor can continue to open old files that use them and it is often best to leave those files as they are. Nonetheless, there may be situations where it is beneficial to convert old files to Unicode. For example, if you are exporting data to another program, you may want to convert to UTF-8 for compatibility. This is especially true when exporting as HDF5 because HDF5 has no support for text encodings other than ASCII and UTF-8.

In Igor, text encoding is an issue primarily for three types of objects:

- Waves (for both wave properties and text wave data)
- Text files (plain text notebooks and procedure files)
- String variables

Igor stores text encoding settings for waves and text files. In Igor Pro 9 and later, Igor defaults to UTF-8 for new waves and text files. Igor7 and Igor8 defaulted to UTF-8 for new waves but for new text files it defaulted to the user-specified default text encoding set using the Misc→Text Encoding→Default Text Encoding submenu.

Unlike waves and text files, Igor does not store text encoding settings for string variables. Consequently Igor assumes that a string variable is encoded as UTF-8 when printing it to the history area or when displaying it in an annotation or control panel or otherwise treating it as text. This assumption will be correct for string variables containing only ASCII text and for non-ASCII string variables created by Igor7 or later. It will be incorrect for non-ASCII string variables created by Igor6 or before; these need to be converted to UTF-8 for proper display.

As explained under **Text Encoding Overview** on page III-460, the UTF-8 text encoding includes the ASCII text encoding as a subset. Consequently, converting ASCII to UTF-8 is a trivial operation that does not change the stored bytes. Converting non-ASCII to UTF-8 changes both the number of bytes stored and their values.

To facilitate conversion to UTF-8, Igor9 and later provide the Convert to UTF-8 Text Encoding dialog which you can invoke via the Misc→Text Encoding submenu.

The Convert to UTF-8 Text Encoding Dialog

The Convert to UTF-8 Text Encoding dialog facilitates conversion of waves, global string variables, and text files to UTF-8. The dialog presents four tabs - a Summary tab plus a tab for each type of object that can be converted.

When you click the Convert Waves, Convert Strings, or Convert Text Files buttons, conversions are done on objects in memory only. If you then save the experiments, these objects are written to disk.

Back Up Before Converting to UTF-8

You should back up all data before using the Convert to UTF-8 Text Encoding dialog. Possible problems include:

- Converting objects to UTF-8 that you need to access using Igor Pro 6
- Choosing the wrong source text encoding when converting strings and waves whose text encoding is unknown