

correct text encoding.

The best solution for text encoding issues is to convert the text to valid UTF-8. Once you have successfully opened the file in Igor, you can convert it to UTF-8 by clicking the text encoding button in the status bar. You can convert all open non-UTF-8 text files to UTF-8 using the Convert to UTF-8 dialog; see [Converting to UTF-8](#) on page III-485 for details.

UTF-8 Procedure Files Containing Invalid UTF-8

Though rare, it is possible to create a UTF-8 procedure file containing invalid UTF-8. This occurs if you use invalid UTF-8 in a window and save the window as a recreation macro. For example:

```
Display/TestGraph
Textbox "\xC5"           // Invalid UTF-8 text in annotation
DoWindow/R TestGraph    // Save graph as recreation macro
```

This creates a TestGraph window macro that recreates the annotation which contains invalid UTF-8. This is a contrived example, but the situation occurs in real life if, for example, you obtain the text for the annotation from a source (e.g., database) that provides non-ASCII text in a text encoding other than UTF-8 and you fail to convert it to UTF-8.

If you now save the experiment to disk, you will have invalid UTF-8 procedure text in the experiment's built-in procedure window and also in the experiment's recreation macros that Igor executes to recreate the graph window when you reopen the experiment.

Prior to Igor Pro 9.00, you would get confusing errors on reopening the experiment. In Igor Pro 9.00 and later, if a procedure file contains a UTF-8 byte order mark or UTF-8 TextEncoding pragma, the file is loaded as UTF-8 even if it contains byte sequences that are not valid UTF-8. The invalid bytes are preserved in the text and are displayed as Unicode replacement characters (U+FFFD). This allows Igor to recreate the experiment as it was, invalid UTF-8 and all.

Some Plain Text Files Must Be Saved as UTF-8

When Igor saves a plain text file, it tries to save using the file's original text encoding. Sometimes this is not possible. For example:

You open a file that contains MacRoman text including non-ASCII characters as an Igor procedure file and Igor converts it to UTF-8 for internal storage. Now you add a Japanese character to the procedure file and save it.

The file can not be saved as MacRoman because MacRoman can not represent Japanese characters. In this event, Igor saves the file in UTF-8 and displays an alert informing you of that fact. If you open this file in Igor6, any non-ASCII characters will be incorrect.

If you tell Igor to save a plain text file in a specific text encoding, for example via `SaveNotebook/ENCG=<textEncoding>`, and if it is not possible to save in that text encoding, Igor returns a text encoding conversion error. You can then explicitly tell Igor to save as UTF-8 using `SaveNotebook/ENCG=1`.

History Text Encoding

When an experiment is saved, Igor saves the history text as plain text.

When you create a new experiment, Igor sets the history area text encoding to your chosen default text encoding. When you save the experiment, Igor tries to save the history text using that text encoding. If the history text contains characters that can not be represented in your default text encoding, Igor then saves the history text as UTF-8 and sets the history area text encoding to UTF-8. If you open this experiment in Igor6, any non-ASCII characters in the history, including the bullet characters used to mark commands, will appear as garbage.