

There is currently no fix for plain text notebooks other than setting "Override Experiment" and setting the default text encoding to the correct encoding and reloading or converting the file to UTF-8 with a byte order mark.

### The default text encoding

If set to MacRoman or Windows, the default text encoding will always succeed but it might be wrong. For example, Japanese can be successfully interpreted as MacRoman or Windows but you will get the wrong characters.

For a procedure file you can fix this by adding a TextEncoding pragma.

There is currently no fix for plain text notebooks other than setting "Override Experiment" and setting the default text encoding to the correct encoding and reloading or converting the file to UTF-8.

### Plain Text Text Encoding Conversion Errors

When Igor opens a plain-text file or an Igor6-compatible experiment file, it must convert plain text to UTF-8. In order to do this, it needs to determine the text encoding of the plain text file or the plain text item within an experiment file.

As the previous section briefly outlined, it is possible that Igor might get this wrong. To illustrate the types of problems that can occur in this situation, we will consider the following scenario:

In Igor6.x you have created a standalone (not part of an experiment) plain text file on Windows with English as the system locale so the text in the file is encoded as Windows-1252. The file contains some non-ASCII characters. You transfer the file to Macintosh. Your default text encoding, as selected in the Default Text Encoding submenu of the Misc menu, is Western. You open the plain text file as a notebook in Igor.

Since the file is a plain text file, it contains nothing but plain text; it does not contain information indicating the platform of origin or the text encoding. Consequently Igor uses your selected default text encoding - Western. In the case of a plain text file, Western means "MacRoman when running on Macintosh, Windows-1252 when running on Windows", so Igor uses MacRoman to convert the text to UTF-8. This does not cause an error but gives an incorrect translation of the non-ASCII characters in the file.

Unfortunately, Igor has no way to detect this kind of misidentification error. The conversion completes without error but produces some incorrect characters and there is no way for Igor to know that this has happened. This is an example of a conversion that succeeds but is incorrect.

To get the correct characters you must close the file, choose Windows-1252 from the Default Text Encoding menu, and reopen the file. Now Igor will correctly translate the text to UTF-8. You should restore the default text encoding to its normal value for future use.

An alternative solution is to execute OpenNotebook/ENCG=3. This explicitly tells Igor that the file's text encoding is Windows-1252. Igor's text encoding codes are listed under **Text Encoding Names and Codes** on page III-490.

Now let's change the scenario slightly:

Instead of your default text encoding being set to Windows-1252, it is set to Japanese. You open the standalone plain text file containing Windows western non-ASCII characters as a notebook on either Macintosh or Windows.

Since the file does not contain any text encoding information, Igor tries to convert the text to UTF-8 using default text encoding, Japanese, as the original text encoding. Two things may happen:

- The conversion may succeed. In this case you will see some seemingly random Japanese characters in your notebook because, although the conversion succeeded, it incorrectly interpreted the text.
- The conversion may fail because the file contains byte patterns that are illegal in the Shift JIS text encoding. In this case Igor will display the Choose Text Encoding dialog and you can choose the