The Shift JIS (Japanese) text encoding does not have a character that maps to the bullet character (U+2022). Consequently, if your default text encoding is Shift JIS, when you save an experiment, Igor will be unable to save the history text as Shift JIS and will save it as UTF-8 instead.

Byte Order Marks

A byte order mark, called a BOM for short, is a special character sometimes used in Unicode plain text files to identify the file as Unicode and to identify the Unicode encoding form used by the file. The BOM, if present, must appear at the beginning of the file.

A FAQ about BOMs can be found at http://www.unicode.org/faq/utf_bom.html#BOM.

Most Unicode plain text files omit the BOM. However, including it has the advantage of unambiguously identifying the file's text encoding.

The BOM is the Unicode "zero width no-break space" character. It's code point is U+FEFF which is represented by the following bytes:

UTF-8	0xEF 0xBB 0xBF
UTF-16 Little Endian	0xFF 0xFE
UTF-16 Big Endian	0xFE 0xFF
UTF-32 Little Endian	0x00 0x00 0xFF 0xFE
UTF-32 Big Endian	0xFE 0xFF 0x00 0x00

If a Unicode file contains a BOM, Igor preserves it when saving the file.

By default Igor writes a BOM when writing a Unicode plain text file and removes the BOM, if present, when reading a plain text file into memory.

When Igor creates a new plain text file, it sets an internal writeBOM flag for that file to true. If the file is later saved to disk as Unicode and if the writeBOM flag is still set, Igor writes the BOM to the file.

When Igor opens a plain text file, it checks to see if the file contains a BOM. If not it clears the file's writeBOM flag. If the file does contain a BOM, Igor sets the writeBOM flag for the file and removes the BOM from the text as loaded into memory. When Igor writes the text back disk, if the writeBOM flag is set, Igor writes the BOM and then the text.

You can see the state of the writeBOM flag using the File Information dialog which you access via the Notebook or Procedure menu. If the file's text encoding is a form of Unicode and the writeBOM flag is set, the Text Encoding section of the File Information dialog will say "with byte order mark".

You can specify the value of the writeBOM flag for a plain text notebook file using **NewNotebook** with the /ENCG flag.

You can set or clear the writeBOM flag for a plain text notebook or procedure file using the Text Encoding dialog accessed via the Notebook or Procedure menu. The Write Byte Order Mark checkbox is visible only if a Unicode text encoding is selected. You can also set the writeBOM flag for a plain text notebook using the **Notebook** operation, writeBOM keyword.

The built-in procedure window is a special case. Its writeBOM flag defaults to false and it is set to false each time you do New Experiment. We make this exception to allow Igor6 to open an experiment whose procedure window text encoding is UTF-8 without generating an error. Igor6 does not know about UTF-8 so non-ASCII characters will be wrong, but at least you will be able to open the experiment.

If you modify a plain text file that is open as a notebook or procedure file using an external editor, Igor reloads the text from the modified file. This sets the internal writeBOM flag for the notebook or procedure file to true if the modified text includes a BOM or false otherwise. This then determines whether Igor writes