

Each wave stores five settings corresponding to these five plain text elements. The settings record the text encoding used for the corresponding element.

You can inspect the text encodings of the wave elements of a particular wave using the Data Browser. To do so, you must first enable the Text Encoding checkbox in the Info pane of the Data Browser section of the Miscellaneous Settings dialog.

For waves created prior to Igor6.3x, all of the settings are set to 0 which means "unknown". This is because the notion of wave text encodings was added only in Igor Pro 6.30, in anticipation of Igor7 and its use of Unicode as the primary text storage format.

When you make a wave in Igor6.3x the text encoding for each element is set to the system text encoding (typically MacRoman, Windows-1252 or Japanese) except for the text wave content element which is set to "unknown". The reason for treating the text wave content element differently is because the data may be binary data as explained below under **Text Waves Containing Binary Data** on page III-475.

When you make a wave in Igor7 or later, including when you overwrite an existing wave, the text encoding for each element, including the text wave content element, is set to UTF-8.

These defaults are set when you make or overwrite a wave using Make, Duplicate, LoadWave and any other operation that creates a wave. After creating the wave, you can change them explicitly using **Set-WaveTextEncoding**, but only advanced users should do this.

If you overwrite an existing text wave, the text encoding for each element, including the text wave content element, is set to UTF-8, the same as when you first create the wave. However, the wave's text content is not cleared or converted to UTF-8 because it is presumed that you will be completely rewriting the content and, if Igor cleared or converted it on overwriting, it would be a waste of time. If the wave contains non-ASCII text and its previous text content text encoding was not UTF-8, this leaves the wave with invalid text content. This is normally not a problem since you will completely rewrite the text anyway. But if you want to make sure that there is no invalid text, you can clear the wave's text content in the Make/T/O statement. For example:

```
Make/T/O textWave = ""
```

In addition, the text encoding setting for the wave name is set if you rename the wave and the text encoding for the wave note is set if you change the wave note text. However the text encoding for the wave units is not set if you set the units using SetScale, the text encoding for the dimension labels is not set if you set a dimension label using SetDimLabel, and the text encoding for the text content of a text wave is not set if you store text in an element of the wave.

Opening an experiment does not change the text encoding settings for the waves in the experiment. Neither does saving an experiment.

When Igor uses one of these wave elements, it must convert the text from its original text encoding into UTF-8. It uses the rules listed in the next section to determine the original text encoding.

### Determining the Text Encoding for Wave Elements

This section describes the rules that Igor uses to determine the source text encoding for a wave element when loading a wave from an Igor binary wave file or from a packed experiment file. You don't need to know these rules. We present them for technically-oriented users who are interested or who need to understand the details to facilitate troubleshooting.

1. If the element's text encoding is set to UTF-8, Igor uses UTF-8 as the source text encoding.

Igor saves text as UTF-8 if it can not be represented in another text encoding. This rule ensures that Igor correctly loads such text.

2. Otherwise if Override Experiment is turned on, Igor uses the selected default text encoding as the source text encoding.