Use `Edit M_WaveStats.ld` to display the results in a table with dimension labels identifying each of the row statistics.

WaveStats is not entirely multidimensional aware. Even so, much of the information computed by WaveStats is useful. See **Analysis on Multidimensional Waves** on page II-95 for details.

See Also

Chapter III-12, **Statistics** for details on other statistics.

See the **ImageStats** operation for calculating wave statistics for specified regions of interest in 2D matrix waves.

See the **APMath** operation if you require higher precision than provided by double-precision floating point.

WaveMax, **WaveMin**, **WaveMinAndMax**, **mean**, **median**, **Variance**

WaveTextEncoding

`WaveTextEncoding(`*`wave`*`, `*`element`*`, `*`getEffectiveTextEncoding`*`)`

The WaveTextEncoding function returns the text encoding code for the specified element of a wave. See **Wave Text Encodings** on page III-472 for background information.

This function is used to deal with text encoding issues that sometimes arise in when you load pre-Igor Pro 7 experiments. Most users will have no need to use it.

The WaveTextEncoding function was added in Igor Pro 6.30. The *getEffectiveTextEncoding* parameter was added in Igor Pro 7.00.

Parameters

wave specifies the wave of interest.

element specifies a part of the wave, as follows:

Value	Meaning
1	Wave name
2	Wave units
4	Wave note
8	Wave dimension labels
16	Text wave content

getEffectiveTextEncoding determines if WaveTextEncoding returns a raw text encoding code or an effective text encoding code as explained below.

Details

WaveTextEncoding returns a integer text encoding code. See **Text Encoding Names and Codes** on page III-490 for details.

As explained under **Wave Text Encodings** on page III-472, each of the wave elements has a corresponding text encoding setting. Because the notion of text encoding settings was added in Igor Pro 6.30, waves created by earlier versions have their text encoding settings set to unknown (0).

The text encoding setting stored for a given element is the "raw" text encoding. If it is unknown, then Igor applies some rules when the wave is accessed to determine an "effective" text encoding for the element being accessed. The rules are explained under **Determining the Text Encoding for a Plain Text File** on page III-467.

If *getEffectiveTextEncoding* is non-zero then WaveTextEncoding returns the effective text encoding. If *getEffectiveTextEncoding* is zero it returns the raw text encoding.

See Also

Wave Text Encodings on page III-472, **Text Encoding Names and Codes** on page III-490, **Determining the Text Encoding for a Plain Text File** on page III-467