

If you specify a range of FIFO data points, using `/R=[startPoint,endPoint]` then FIFO2Wave dumps the specified FIFO points into the wave after clipping *startPoint* and *endPoint* to valid point numbers.

The valid point numbers depend on whether the FIFO is running and on whether or not it is attached to a file. If the FIFO is running then *startPoint* and *endPoint* are truncated to number of points in the FIFO. If the FIFO is buffering a file then the range can include the full extent of the file.

If you specify no range then FIFO2Wave transfers the most recently acquired FIFO data to the wave. The number of points transferred is the smaller of the number of points in the FIFO and number of points in the wave.

FIFO2Wave may or may not change the wave's X scaling and number type, depending on the current X scaling and on the `/S` flag.

Think of the wave's X scaling as being controlled by two values, x_0 and dx , where the X value of point p is $x_0 + p \cdot dx$. FIFO2Wave always sets the wave's dx value equal to the FIFO's ΔT value (as set by the `CtrlFIFO` operation). If you use no `/S` flag, FIFO2Wave does not set the wave's x_0 value nor does it set the wave's number type.

If you are using FIFO2Wave to update a wave in a graph as quickly as possible, the `/S=0` flag gives the highest update rate. The other `/S` values trigger more recalculation and slow down the updating.

If the wave's number type (possibly changed to match the FIFO channel) is a floating point type, FIFO2Wave scales the FIFO data before transferring it to the wave as follows:

```
scaled_value = (FIFO_value - offset) * gain
```

If the FIFO channel's gain is one and its offset is zero, the scaling would have no effect so FIFO2Wave skips it.

If the specified FIFO channel is an image strip channel (one defined using the optional `vectPnts` parameter to `NewFIFOChan`), then the resultant wave will be a matrix with the number of rows set by `vectPnts` and the number of columns set by the number of points described above for one-dimensional waves. To create an image plot that looks the same as the corresponding channel in a Chart, you will need to transpose the wave using **MatrixTranspose**.

See Also

The **NewFIFO** and **CtrlFIFO** operations, and **FIFOs and Charts** on page IV-313 for more information on FIFOs and data acquisition. For an explanation of waves and wave scaling, see **Changing Dimension and Data Scaling** on page II-68.

FIFOStatus

FIFOStatus [`/Q`] *FIFOName*

The **FIFOStatus** operation returns miscellaneous information about a FIFO and its channels. FIFOs are used for data acquisition.

Flags

`/Q` Doesn't print in the history area.

Details

FIFOStatus sets the variable `V_flag` to nonzero if a FIFO of the given name exists. If the named FIFO does exist then **FIFOStatus** stores information about the FIFO in the following variables:

<code>V_FIFORunning</code>	Nonzero if FIFO is running.
<code>V_FIFOChunks</code>	Number of chunks of data placed in FIFO so far.
<code>V_FIFOn chans</code>	Number of channels in the FIFO.
<code>S_Info</code>	Keyword-packed information string.

The keyword-packed information string consists of a sequence of sections with the following form: *keyword:value*;

You can pick a value out of a keyword-packed string using the **NumberByKey** and **StringByKey** functions. Here are the keywords for `S_Info`:

In addition, **FIFOStatus** writes fields to `S_Info` for each channel in the FIFO. The keyword for the field is a combination of a name and a number that identify the field and the channel to which it refers. For example, if channel 4 is named "Pressure" then the following would appear in the `S_Info` string: `NAME4:Pressure`.