Dynamic Escape Codes for Tags

The Dynamic pop-up menu inserts escape codes that apply only to tags. These codes insert information about the wave or point in the wave to which the tag is attached. This information automatically updates whenever the wave or the attachment point changes.

Dynamic Item	Effect
Wave name	Displays the name of the wave to which the tag is attached.
Trace name and instance	Same as wave name but appends an instance number (e.g., #1) if there is more than one trace in the graph associated with a given wave name.
Attach point number	Displays the number of the tag attachment point.
Attach point X value	Displays the X value of the tag attachment point.
Attach point Y value	Displays the Y value of the tag attachment point.
Attach point Z value	Displays the Z value of the tag attachment point. Available only for contour traces, waterfall plots, or image plots.
Attach X offset value	Displays the trace's X offset.
Attach Y offset value	Displays the trace's Y offset.

See also **TagVal and TagWaveRef Functions** on page III-38. These functions provide the same information as the Dynamic pop-up menu items but with greater flexibility.

Other Dynamic Escape Codes

You can enter the dynamic text escape sequence which inserts dynamically evaluated text into any kind of annotation using the escape code sequence:

\{*dynamicText*}

where *dynamicText* may contain numeric and string expressions. This technique is explained under **Dynamic Text Escape Codes** on page III-56.

TagVal and TagWaveRef Functions

If the annotation is a tag, you can use the functions **TagVal** (page V-1022) and **TagWaveRef** (page V-1023) to display information about the data point to which the tag is attached. For example, the following displays the Y value of the tag's data point:

```
\{"%g", TagVal(2)}
```

This is identical in effect to the "\0Y" escape code which you can insert by choosing the "Attach point Y value" item from the Dynamic pop-up menu. The benefit of using the TagVal function is that you can use a formatting technique other than %g. For example:

```
\{"%5.2f",TagVal(2)}
```

TagVal is capable of returning all of the information that you can access via the Dynamic menu escape codes. Use it when you want to control the numeric format of the text.

The TagWaveRef function returns a reference to the wave to which the tag is attached. You can use this reference just as you would use the name of the wave itself. For example, given a graph displaying a wave named wave0, the following tag text displays the average value of the wave:

```
\{"%g",mean(wave0)}
```

This is fine, but if you move the tag to another wave it will still show the average value of wave0. Using TagWaveRef, you can make this show the average value of whichever wave is tagged: