See Also

For use in user-defined functions, see **The Simple Input Dialog** on page IV-144.

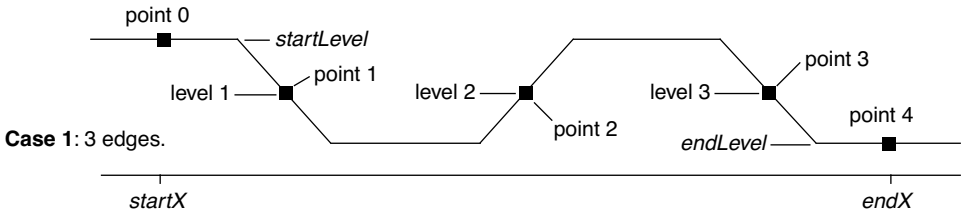
For use in macros, see **The Missing Parameter Dialog** on page IV-121.

For use in functions and macros, see the **DoPrompt** and **popup** keywords.

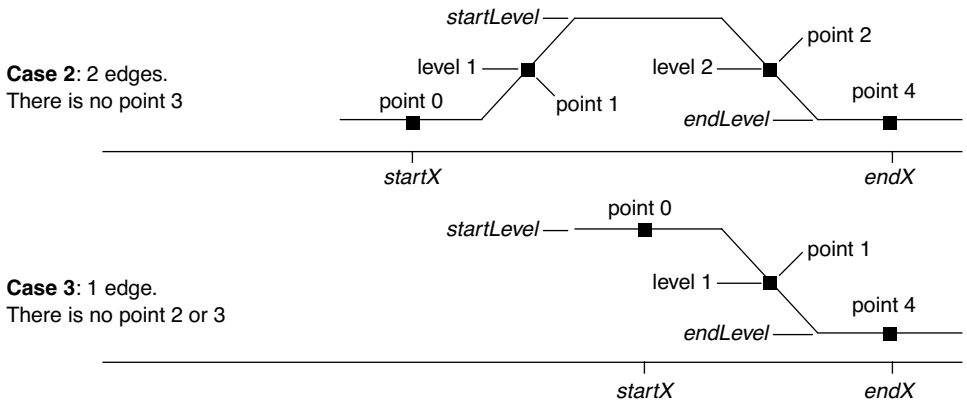
PulseStats

PulseStats [***flags***] ***waveName***

The PulseStats operation produces simple statistics on a region of the named wave that is expected to contain three edges as shown below. If more than three edges exist, PulseStats works on the first three edges it finds.



PulseStats handles other cases in which there are only one or two edges.



Flags

/A=*n* Determines *startLevel* and *endLevel* automatically by averaging *n* points centered at *startX* and *endX*. This does not work in case 2, which requires that you use the /L flag. Default is /A=1.

/B=*box* Sets box size for sliding average. This should be an odd number. If /B=*box* is omitted or *box* equals 1, no averaging is done.

/F=*f* Specifies levels 1, 2, and 3 as a fraction of (*endLevel*-*startLevel*):

```
level1 = level2 = level3 = f*(endLevel-startLevel) + startLevel
```

f must be between 0 and 1. The default value is 0.5 which sets the levels to midway between the base levels.

/L=(*startLevel*, *endLevel*)

Sets *startLevel* and *endLevel* explicitly.

/M=*dx* Sets minimum edge width. Once an edge is found, the search for the next edge starts *dx* units beyond the found edge. Default *dx* is 0.

/P Output edge locations (see **Details**) are set in terms of point number. If /P is omitted, edge locations are set in terms of X values.

/Q Prevents results from being printed in history and prevents error if edge is not found.

/R=(*startX*,*endX*)	Specifies an X range of the wave to search. You may exchange *startX* and *endX* to reverse the search direction.
/R=[*startP*,*endP*]	Specifies a point range of the wave to search. You may exchange *startP* and *endP* to reverse the search direction. If you specify the range as /R=[*startP*] then the end of the range is taken as the end of the wave. If /R is omitted, the entire wave is searched.
/T=*dx*	Forces search in two directions for a possibly more accurate result. *dx* controls where the second search starts.

Details

The /B=box, /T=dx, /P and /Q flags behave the same as for the **FindLevel** operation.

PulseStats considers a region of the input wave between two X locations, called *startX* and *endX*. *startX* and *endX* are set by the /R=(*startX*,*endX*) flag. If this flag is missing, *startX* and *endX* default to the start and end of the entire wave.

The *startLevel* and *endLevel* values define the base levels of the pulse. You can explicitly set these levels with the /L=(*startLevel*, *endLevel*) flag or you can let PulseStats find the base levels for you by using the /A=*n* flag. With this flag, PulseStats determines *startLevel* and *endLevel* by averaging *n* points centered at *startX* and at *endX*. In case 2, you must use /L=(*startLevel*, *endLevel*) since *startLevel* is not at point 0.

Given *startLevel* and *endLevel* and an *f* value (which you can set with the /F=*f* flag), PulseStats computes level1, level2 and level3 which are always equal. With the default *f* value of 0.5, level1 is midway between *startLevel* and *endLevel*.

With these levels defined, PulseStats searches the wave from *startX* to *endX* looking for one, two or three level crossings. PulseStats sets the following variables:

`V_flag`	0: All three level crossings were found. 1: One or two level crossings were found. 2: No level crossings were found.
`V_PulseLoc1`	X location where level1 was found.
`V_PulseLoc2`	X location where level2 was found.
`V_PulseLoc3`	X location where level3 was found.
`V_PulseLvl0`	*startLevel* value.
`V_PulseLvl123`	Level1 value that is the same as level2 and level3.
`V_PulseLvl4`	*endLevel* value.
`V_PulseAmp4_0`	Pulse amplitude (*endLevel* - *startLevel*).
`V_PulseWidth2_1`	Left pulse width (x distance between point 2 and point 1).
`V_PulseWidth3_2`	Right pulse width (x distance between point 3 and point 2).
`V_PulseWidth3_1`	Pulse period (x distance between point 3 and point 1).
`V_PulsePolarity`	Trend of the edge at point 1 (-1 if decreasing, +1 if increasing).

X locations and distances are in terms of the X scaling of the source wave, unless you use the /P flag in which case they are in terms of point number.

If any level crossings are missing then PulseStats sets the associated variables to NaN (Not a Number). If one crossing is missing, variables depending on point 3 are set to NaN. If two crossings are missing, variables depending on points 2 and 3 are set to NaN. If all crossings are missing, variables depending on points 1, 2, and 3 are set to NaN. You can use the numtype function to test a variable to see if it is NaN.

The PulseStats operation is not multidimensional aware. See **Analysis on Multidimensional Waves** on page II-95 for details.