

## FindAPeak

Searches for *listSepStr* are always case-sensitive. The comparison of *itemStr* to the contents of *listStr* is usually case-sensitive. Setting the optional *matchCase* parameter to 0 makes the comparison case insensitive.

In Igor6, only the first byte of *listSepStr* was used. In Igor7 and later, all bytes are used.

If *startIndex* is specified, then *listSepStr* must also be specified. If *matchCase* is specified, *startIndex* and *listSepStr* must be specified.

### Examples

```
Print FindListItem("w1", "w0;w1;w2," )           // prints 3
Print FindListItem("v2", "v1,v2,v3,", ",")       // prints 3
Print FindListItem("v2", "v0,v2,v2,", ",", 4)    // prints 6
Print FindListItem("C", "a;c;C;")                // prints 4
Print FindListItem("C", "a;c;C;", ";", 0, 0)      // prints 2
```

### See Also

The **AddListItem**, **strsearch**, **StringFromList**, **RemoveListItem**, **RemoveFromList**, **ItemsInList**, **WhichListItem**, **WaveList**, **TraceNameList**, **StringList**, **VariableList**, and **FunctionList** functions.

## FindAPeak

**FindAPeak** [/B=*baseWaveName*] *minamp*, *pol*, *box*, *peakWave* [ (*startX*,*endX*) ]

FindAPeak locates the maximum or minimum of a peak by analyzing smoothed first and second derivatives.

The FindAPeak operation is used primarily by the Igor Technical Note #20 and its variants. For most purposes, use the more flexible **FindPeak** operation instead of FindAPeak.

### Parameters

*minamp* is minimum amplitude ("threshold") of a peak. Use it to reject small or spurious peaks.

*pol* is the expected peak polarity. Specify 1 to search for a positive-going peak or 2 to search for a negative-going peak.

*box* is the number of peak values to include in the sliding average when smoothing the derivatives. If you specify an even number, the next-higher odd number is used.

*peakWave* specifies the wave containing the peak.

[*startX*,*endX*] is an optional subrange to search in point numbers.

(*startX*,*endX*) is an optional subrange to search in X values.

If you omit the subrange, *startX* defaults to the first point in *peakWave* and *endX* defaults to the last point in *peakWave*.

The search always with *startX* and ends at *endX*, regardless of whether *startX* is less than or greater than *endX*. You can use this to control the direction of the search.

### Flags

/B=*baseWave*                      Specifies a base wave containing values to subtract from *peakWave* to compute the derived data which FindAPeak searches for peaks.

### Details

FindAPeak creates a temporary smoothed version of *peakWave* and a temporary first derivative of the smoothed data. It scans through the first derivative for the first zero-crossing where the smoothed data exceeds the minimum amplitude as specified by *minamp*. The location of the zero-crossing is then more accurately determined by reverse linear interpolation. The smoothed second derivative is computed at that point to see if the peak is a positive-going or negative-going peak.

### Output Variables

FindAPeak reports results through these output variables:

V_Flag	0 if a peak is found and to 1 if no peak is found.
V_peakX	The interpolated X value of the peak center.
V_peakP	The interpolated fractional point number of the peak center.