

<code>/V=rValueWave</code>	Specifies the match sequence wave in the case of single/double precision numbers.
<code>/Z</code>	No error reporting.

Details

If the match sequence is specified via the `/V` flag, it is considered to be a floating point wave (i.e., single or double precision) in which case it is compared to data in the wave using a tolerance value. If the tolerance is not specified by the `/T` flag, the default value 1.0^{-7} .

If the match sequence is specified via the `/I` flag, the sequence is assumed to be an integer wave (this includes both signed and unsigned char, signed and unsigned short as well as long). In this case *srcWave* must also be of integer type and the operation searches for the sequence based on exact equality between the match sequence and entries in the wave as signed long integers.

If the match sequence is unsigned long wave use the `/U` flag to specify the value for an integer comparison.

You can also use this operation on waves of two or more dimensions. In this case you can calculate the rows, columns, etc. For example, in the case of a 2D wave:

```
col=floor(V_value/rowsInWave)
row=V_value-col*rowsInWave
```

See Also

FindValue, **StringToUnsignedByteWave**

FindValue

FindValue [*flags*] *srcWave*

FindValue [*flags*] *txtWave*

The FindValue operation finds the location of the specified value starting the search from the specified start point. It stores the results of the search in the variables `V_value`, `V_row`, `V_col`, `V_layer`, and `V_chunk`.

Flags

<code>/FNAN</code>	Specifies searching for a NaN value when <i>srcWave</i> is floating point. The <code>/FNAN</code> flag was added in Igor Pro 7.00.
<code>/I=iValue</code>	Specifies an integer value for integer search.
<code>/R</code>	Reverses the order of the search. In the absence of <code>/S</code> and <code>/RMD</code> , the search starts from the end of the wave. <code>/R</code> is not compatible with <code>/UOFV</code> . The <code>/R</code> flag was added in Igor Pro 9.00.

`/RMD=[firstRow,lastRow][firstColumn,lastColumn][firstLayer,lastlayer][firstChunk,lastChunk]`

Designates a contiguous range of data in the source wave to which the operation is to be applied. This flag was added in Igor Pro 8.00.

You can include all higher dimensions by leaving off the corresponding brackets. For example:

```
/RMD=[firstRow,lastRow]
```

includes all available columns, layers and chunks.

You can use empty brackets to include all of a given dimension. For example:

```
/RMD=[][firstColumn,lastColumn]
```

means "all rows from column A to column B".

You can use a `*` to specify the end of any dimension. For example:

```
/RMD=[firstRow,*]
```

means "from firstRow through the last row".

FindValue

<code>/S=start</code>	Sets start of search in the wave. If <code>/S</code> is not specified, start is set to 0.
<code>/T=tolerance</code>	Use this flag when comparing floating point numbers to define a non-negative tolerance such that the specified value \pm <i>tolerance</i> will be accepted.
<code>/TEXT=templateString</code>	Specifies a template string that will be searched for in <i>txtWave</i> .
<code>/TXOP=txOptions</code>	Specifies the search options using a combination of binary values. 1: Case sensitive 2: Whole word 4: Whole wave element
<code>/U=uValue</code>	Specifies the match value in case of unsigned long range.
<code>/UOFV</code>	The <code>/UOFV</code> (unordered find value) flag, which was added in Igor Pro 8.00, runs the search using multiple threads with each thread searching a different section of the wave. The search terminates when any thread finds a matching value. Use <code>/UOFV</code> when you need the fastest result and you do not care if it finds the first matching value in the wave or a subsequent matching value. By default <code>/UOFV</code> is ignored if <i>srcWave</i> contains fewer than 2,000 points. You can modify this value using MultiThreadingControl . <code>/UOFV</code> is also ignored if you use the <code>/RMD</code> flag.
<code>/V=rValue</code>	Specifies the match value in the case of single/double precision numbers. For most purposes you should also use <code>/T</code> to specify the tolerance.
<code>/Z</code>	No error reporting.

Details

If the match value is specified via the `/V` flag, it is considered to be a floating point value in which case it is compared to data in the wave using a tolerance value. If the tolerance is not specified by the `/T` flag, the value 10^{-7} is used.

If the match value is specified via the `/I` flag, the value is assumed to be an integer. In this case *srcWave* must be of integer type and the operation searches for the value based on exact equality between the match value and entries in the wave as signed long integers.

If the match value is unsigned long use the `/U` flag to specify the value for an integer comparison.

The result of the search is stored in the output variables `V_value`, `V_row`, `V_col`, `V_layer`, and `V_chunk`. `V_value` is set to the index in *srcWave*, treating it as 1D regardless of its dimensionality, where the searched value was found or to -1 if it was not found. `V_row`, `V_col`, `V_layer`, and `V_chunk` are set to the row, column, layer, and chunk number where the searched value was found or to -1 if it was not found but the latter three are set to NaN if the corresponding dimension does not exist in *srcWave*.

When searching for text in a text wave the operation creates the variable `V_value` as above but it also creates the variable `V_startPos` to specify the position of *templateString* from the start of the particular wave element.

Example

```
Make jack = sin(x/8)          // Single-precision floating point
Display jack

// This prints -1 because 0.5 +/- 1.0E-7 does not occur in wave jack
FindValue /V=.5 jack; Print V_value

// This prints 21 because 0.5 +/- 0.01 does occur in wave jack
FindValue /V=.5 /T=.01 jack; Print V_value

// The value of jack(21), to 6 decimal digits of precision, is 0.493920
Print jack(21)
```

See Also

FindSequence, FindLevel, FindLevels, FindDuplicates