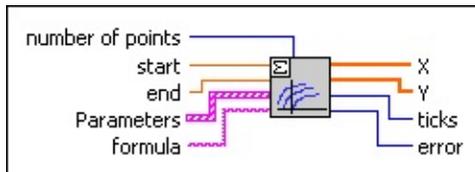


# Eval $y=f(a,x)$ VI

Owning Palette: [1D & 2D Evaluation VIs](#)

Installed With: Full Development System

A generalized version of the [Eval  \$y=f\(x\)\$](#)  VI with the possibility of adding some parameters into the formula.



■ Place on the block diagram ■ Find on the **Functions** palette

**U32** **number of points** is the number of all calculated points. The independent variable is split into equidistant subpoints. The default is 10.

**DBL** **start** is the start point of the interval. The default is 0.0.

**DBL** **end** is the end point of the interval. The default is 1.0.

**Parameters** is an array of clusters describing the parameters.

**abc** **name** of the parameter that uses the [conventions](#) of the [Formula Parsing](#) VIs.

**DBL** **value** is the user-defined value of the parameter.

**abc** **formula** is a string describing the function under investigation. The formula can contain any number of valid [variables](#).

**DBL** **X** is the array of equidistant points between **start** and **end**.

**DBL** **Y** is the function values at the points **X**.

**U32** **ticks** is the time in milliseconds to analyze the formula and to produce the **X** and the **Y** array.

**I32** **error** returns any [error](#) or warning from the VI. You can wire **error** to the [Error Cluster From Error Code](#) VI to convert the error code or warning into an error cluster.

## Formula Parsing VI Variables

The [Formula Parsing](#) VIs accept only the following variables:

*a*, *a0*, ..., *a9*

*b*, *b0*, ..., *b9*

.

.

.

*z*, *z0*, ..., *z9*



**Note** For variable and function names, only lowercase letters are allowed. The VIs interpret capital letters as errors.

All numbers in exponential notation use the 1E–1 convention with the capital letter E. Using 1e–1 with the lowercase letter e results in an error message.

# Formula Parsing VIs

Owning Palette: [Scripts & Formulas VIs](#)

**Installed With:** Full Development System. This topic might not match its corresponding palette in LabVIEW depending on your operating system, licensed product(s), and target.

Use the Formula Parsing VIs to interpret input strings as formulas, transform the formulas into numeric calculations, and return the results.

The VIs on this palette can return [formula parsing and mathematics error codes](#).

Palette Object	Description
<a href="#">Parse Formula Node</a>	Analyzes <a href="#">Eval Formula Node</a> VI inputs and yields an intermediate state as an input for the <a href="#">Eval Parsed Formula Node</a> VI.
<a href="#">Parse Formula String</a>	Analyzes a string as a formula and produces two numeric arrays. These arrays can be used by the <a href="#">Eval Parsed Formula String</a> VI.
<a href="#">Substitute Variables</a>	Substitutes a formula string by given rules. The rules have a <b>parameter name - parameter content</b> structure.