GENESIS64
Array Expressions

APPLICATION NOTE

October 2015

Description: Guide to working with an array of elements.

Introduction

GENESIS64 offers you a set of functions which will ease your work with arrays of elements. These functions can be found in the "Array Functions" submenu of the Expression editor, which makes them available in almost all products of the GENESIS64 suite. In this application note we will use these array functions in GraphWorX64, but you can use them in the same way in AssetWorX, Unified Data Manager, etc.

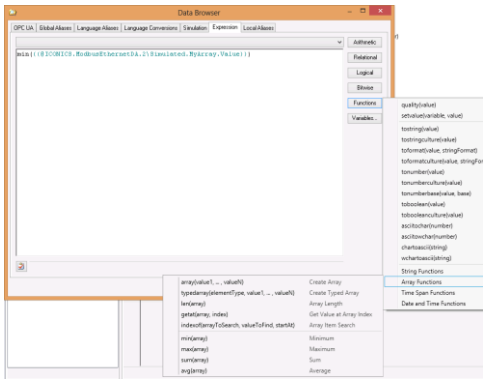

Figure 1 - Array Functions Accessible from Expression Editor

NOTE: You can also use a limited set of array functions by means of Extended Point Syntax as described in the application note *GENESIS64 – Using Extended Point Syntax*.

Create an Array

We can create an array of mixed types or a typed array.

array Function

This function creates a new array containing the specified values. The element type for the array is unspecified, so the array can contain values of different datatypes simultaneously (such as numbers and strings).

Parameters:
- Parameter1: The first value to include in the array
- ...
- ParameterN: The Nth value to include in the array

Returns:
- An array of objects

typedarray Function

This function creates a new array containing the specified values, where the values are all of the specified datatype.

Parameters:
- Parameter1: The datatype of the elements in the array
- Parameter2: The first value to include in the array
- ...
- ParameterN: The Nth value to include in the array

Returns:
- An array of the specified elementType

The elementType is specified as the string name of the datatype. Valid values are:

- "object"
- "string"
- "double"
- "single" or "float"
- "int16" or "short"
- "int32" or "int"
- "int64" or "long"
- "uint16" or "ushort"
- "uint32" or "uint"
- "uint64" or "ulong"
- "boolean" or "bool"
- "byte"
- "sbyte"
- "datetime"

Write an Array to an Array tag

It is not uncommon for a user to want to write multiple values to an array tag at once. This is possible via a script, but the script cannot be used inside Silverlight or HTML5 deployments. In these cases you should use the array functions to specify an arbitrary number of parameters, and it will then simply turn this into an array that can be written to the array tag. For instance, you can create a button with a Pick action with the following properties:

Command: *Write Value*
DataSource:
@ICONICS.ModbusEthernetDA.2\Simulated.MyArray.Value
OnUpValue:
x=typedarray("int", {{localsim::Tag1:Int32}}, {{localsim::Tag2:Int32}})

Upon the release of this button the values of *Tag1* and *Tag2* will be written into the two-element *MyArray* array tag in the ICONICS Modbus OPC DA server.

Get a Value in the Array

In order to get a value in the specified array at the specified index or to get the character in the specified string at the specified index we can use the getat function.

getat Function

Parameters:
Parameter1: a string or array
Parameter2: a zero-based integer index

Returns:
The value in the array or string at the specified index

Get the Length of an Array

If you need to get the length of a string or get the number of elements in an array, you can use the len function.

len Function

Parameters:
Parameter1: a string or array
Returns:
The length of the string or array passed to Parameter1

Search for an item in an Array

You can use the indexof function to retrieve the zero-based index of the first occurrence of an item in an array. You can also use it to get the zero-based index of the first occurrence of a substring within a string.

indexof Function

Parameters:
Parameter1: The string or array in which to search
Parameter2: The value to search for
Parameter3: The zero-based index at which to start searching in Parameter1

Returns:
The zero-based position of Parameter2 within Parameter1, or -1 if not found

Get Min, Max, Avg or Sum of numbers

If you need to find out the smallest, the largest, or the average value of all numbers in an array you can use the min, max, or avg functions. You can also use the sum function to get the total value of all numbers in an array. You can either pass individual parameters into the functions, or you can pass an array into these functions and they will determine the result of all of the values in the array.

min Function

Parameters:
Parameter1: The first number to compare
Parameter2: The second number to compare
...
ParameterN: The Nth number to compare

Returns:
The smallest value of all of the specified parameters

max Function

Parameters:
Parameter1: The first number to compare
Parameter2: The second number to compare
...
ParameterN: The Nth number to compare

Returns:
The largest value of all of the specified parameters

avg Function

Parameters:
Parameter1: The first number to include in the average
Parameter2: The second number to include in the average
...
ParameterN: The Nth number to include in the average

Returns:
The average of all of the specified parameters

sum Function

Parameters:
Parameter1: The first number to add
Parameter2: The second number to add
...
ParameterN: The Nth number to add

Returns:
The sum of all of the specified parameters

Copyright 2015 ICONICS, Inc.