SRP_Array Join

Logically combines two arrays.

Syntax

```
NewArray = SRP_Array("Join", LeftArray, RightArray, Operation, Delim)
```

Returns

The logical combination of both arrays.

Parameters

Parameter	Description
LeftArray	The first dynamic array to be merged. **(REQUIRED)**
RightArray	The second dynamic array to be merged. **(REQUIRED)**
Operation	Determines how the arrays are merged. Options are AND, OR, NOT, and XOR. For case insensitive operations, use ANDC, ORC, NOTC, and XORC. *(OPTIONAL)*
Delim	The delimiter that separates elements in both arrays. *(OPTIONAL)*

Remarks

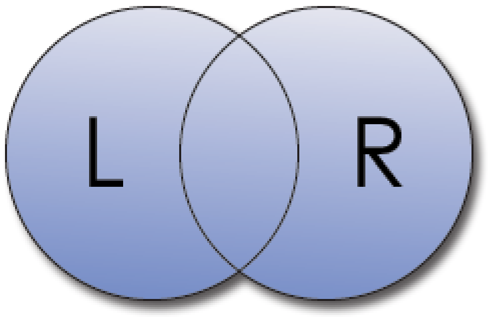
It's not unusual to have at your disposal two arrays of unique values, especially arrays of keys. The Join service takes two arrays and merges them into a new array. For example, imagine an array of employee keys that references all managers and another array that references all female employees. The Join service can quickly produce the intersection of these two arrays, giving you an array of keys for all female managers.

The Join service can join two arrays using four logical operations: AND, OR, NOT, and XOR. Each operation is described in detail below. If omitted, the "OR" operation is used. **The merge is case sensitive by default. For case insensitive merges, append a C to the operation, i.e., ANDC, ORC, NOTC, or XORC.**

IMPORTANT: Keep in mind that no matter which operation you choose, the Join service only returns unique values.

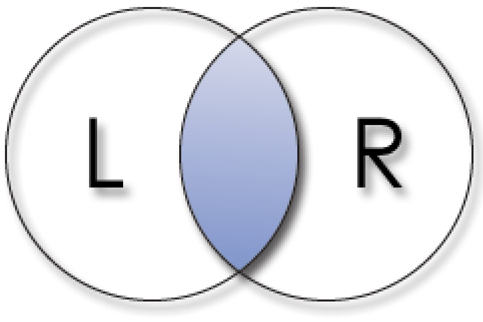
OR

The logical OR operation returns all items that appear in either LeftArray or RightArray. This creates the union of two lists. It's effectively the same as append the two arrays, with the exception that it still only returns unique values. Here is the Venn diagram for the OR operation:



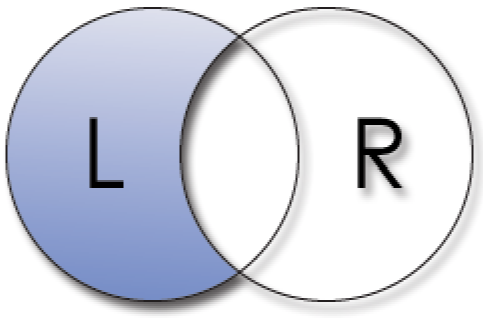
AND

The logical AND operation returns only those items that appear in both LeftArray and RightArray. This is what we call the intersection of the two lists as demonstrated by the following Venn diagram:



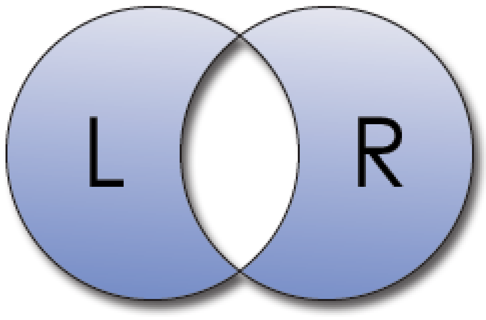
NOT

The logical NOT operation returns only those items that appear in LeftArray but not in RightArray. In other words, think of RightArray as an exclusion list. Here's the Venn diagram for the NOT operation:



XOR

The logical XOR operation is shorthand for Exclusive OR. It returns all values that appear in LeftArray or RightArray, but not both. The following Venn diagram illustrates how this is the inverse of the AND operation:



Note: The Join service is case sensitive. That means that "John" and "JOHN" are considered different values. Since the Join service was designed for arrays of keys, we felt this approach was appropriate.

The Delim parameter is the delimiter for both arrays. It is also the delimiter used to build the resulting array. If omitted, this parameter is set to @FM.

Examples

```
// Make two big arrays, the first is every two numbers, the second is every three numbers
LeftArray = ""
RightArray = ""
For iRow = 1 to 10000
   LeftArray := iRow * 2:@FM
   RightArray := iRow * 3:@FM
Next iRow
LeftArray[-1, 1] = ""
RightArray[-1, 1] = ""

// Get the intersection, only those numbers that appear in both arrays, without duplicates
ArrayIntersect = SRP_Array("Join", LeftArray, RightArray, "AND", @FM)

// Remove all items in the left array that also appear in the right array (omitting Delim defaults to @FM)
ArrayNot = SRP_Array("Join", LeftArray, RightArray, "NOT")

// Get an array of those numbers do not appear in both lists
ArrayXor = SRP_Array("Join", LeftArray, RightArray, "XOR")

// Get the union, all numbers from both lists, without duplicates
ArrayUnion = SRP_Array("Join", LeftArray, RightArray, "OR")

// We can also get the union of @FM delimited arrays by omitting both optional parameters
ArrayUnion = SRP_Array("Join", LeftArray, RightArray)
```