

SRP_Reorder_Array

Moves elements in a dynamic array into any explicit order.

Syntax

```

NewArray = SRP_Reorder_Array(Array, Order, Delim)

```

Returns

The reordered array.

Parameters

Parameter	Description
Array	The dynamic array to be reordered. Required.
Order	An @FM delimited list of the element indexes in their new order. For instance, if you have an array with five fields, you could pass 5:@FM:4:@FM:3:@FM:2:@FM:1 to reverse its order. Or you could pass 3:@FM:2:@FM:1 to change the order of the first three fields. Note that the routine is smart enough to handle too few, too many, or duplicate indexes. Required.
Delim	The delimiter that separates elements in Array. Optional. If you omit this parameter, @FM will be assumed.

Remarks

There are times when you need to move array elements into different positions, whether it be for reporting purposes or for preparing an array for sorting with V119. Often times, doing this in BASIC+ is too slow. SRP_Reorder_Array accomplishes this task with high performance.

To understand what is meant by reordering, let's look at an array. This array has five fields, and the contents of those fields can be single valued or multi-valued. SRP_Reorder_Array can handle both cases.



As an example, let's reverse the order of the array. To do this, we pass an @FM delimited array of indexes. The indexes refer to the field we want move, and the position of that index determines the order. So, to reverse the order, we pass 5:@FM:4:@FM:3:@FM:2:@FM:1 to the Order parameter. The result is this:



SRP_Reorder_Array elegantly handles too few, too many, or duplicate indexes in the Order parameter. For instance, if you only want to reorder the first three fields of our original array, you can just pass 3:@FM:2:@FM:1 to the Order parameter. The result is then:



When there are too many or duplicate indexes, SRP_Reorder_Array just ignores them.

Examples

```
* Reverse the order
Array = "Don":@FM:"Paul":@FM:"Frank":@FM:"Bob":@FM:"Kevin"
NewArray = SRP_Reorder_Array(Array, 5:@FM:4:@FM:3:@FM:2:@FM:1)
// NewArray will be: "Kevin":@FM:"Bob":@FM:"Frank":@FM:"Paul":@FM:"Don"

* Reverse the order of only the first three elements
Array = "Don":@FM:"Paul":@FM:"Frank":@FM:"Bob":@FM:"Kevin"
NewArray = SRP_Reorder_Array(Array, 3:@FM:2:@FM:1)
// NewArray will be: "Frank":@FM:"Paul":@FM:"Don":@FM:"Bob":@FM:"Kevin"

* Swap elements 3 and 5
Array = "Don":@FM:"Paul":@FM:"Frank":@FM:"Bob":@FM:"Kevin"
NewArray = SRP_Reorder_Array(Array, 1:@FM:2:@FM:5:@FM:4:@FM:3)
// NewArray will be: "Don":@FM:"Paul":@FM:"Kevin":@FM:"Bob":@FM:"Frank"

* Reverse the order of a comma delimited array
Array = "Don,Paul,Frank,Bob,Kevin"
NewArray = SRP_Reorder_Array(Array, 5:@FM:4:@FM:3:@FM:2:@FM:1, ",")
// NewArray will be: "Kevin,Bob,Frank,Paul,Don"
```