

SRP_Array SortSimpleList

Sorts a one dimensional dynamic array.

Syntax

```
NewArray = SRP_Array("SortSimpleList", Array, Option, Delim)
```

Returns

The sorted array.

Parameters

Parameter	Description
Array	The one dimensional dynamic array to be sorted. (REQUIRED)
Option	A keyword indicating how the array is to be sorted. <i>(OPTIONAL)</i>
Delim	The delimiter that separates elements of the Array. <i>(OPTIONAL)</i>

Remarks

While the [SortRows](#) provides everything needed to sort both one and two dimensional arrays, this service simplifies the process greatly of sorting single dimensional arrays. Have a small simple list to sort? This is the service for you. Simply pass in the array and choose one of sorting options:

- "AscendingText" performs a left sort in alphabetical order.
- "DescendingText" performs a left sort in reverse alphabetical order.
- "AscendingNumbers" performs a sort in numerical order.
- "DescendingNumbers" performs a sort in reverse numerical order.
- "AscendingRight" performs a right-aligned sort in alphabetical order.
- "DescendingRight" performs a right-aligned sort in reverse alphabetical order.

Note: You can append a "C" to the end of the above options to make them case insensitive.

The Delim parameter tells the service what delimiter is used to delimit elements in the given array. When omitted, @FM is assumed.

Examples

```
// Sort a simple list using the default delimiter
List = "Don":@FM:"Paul":@FM:"Frank":@FM:"Bob":@FM:"Kevin"
List = SRP_Array("SortSimpleList", List)
// List will be: "Bob":@FM:"Don":@FM:"Frank":@FM:"Kevin":@FM:"Paul"

// Sort a simple list in reverse order
List = "Don":@FM:"Paul":@FM:"Frank":@FM:"Bob":@FM:"Kevin"
List = SRP_Array("SortSimpleList", List, "DescendingText")
// List will be: "Paul":@FM:"Kevin":@FM:"Frank":@FM:"Don":@FM:"Bob"

// Sort a simple list using a custom delimiter
List = "81,90,44,66,52"
List = SRP_Array("SortSimpleList", List, "AscendingNumbers", ",")
// List will be: "44,52,66,81,90"
```