

# Count Function

## Description

Counts the number of occurrences of a substring in a string.

## Syntax

*instances* = **Count** (*source\_string*, *search\_string*)

## Parameters

The Count function has the following parameters.

Parameter	Description
<i>source_string</i>	Identifies the string to search for in <i>search_string</i> . Can be a literal string, expression, constant, or variable.
<i>search_string</i>	Identifies the string to search for in <i>source_string</i> . <i>search_string</i> may be a literal string, expression, constant, or variable. If <i>search_string</i> does not appear in <i>source_string</i> , or if <i>search_string</i> is null, a 0 (zero) will be returned.

The **Count()** and **DCount()** functions are usually used to determine the size of a dynamic array. They actually have a more general purpose use, because the search string can be any string used as a delimiter, even multiple character strings.

**Note:** To count the number of values in a dynamic array, you need to account for the possibility that the array has one value, with no delimiter. The *count()* function in this case will return 0 because there is no delimiter in the string. To return the correct answer (1), you need to add the boolean expression ( *var #* ), which returns 1 if the variable contains a value but 0 if it is null. See the example below.

**Note:** If you use **DCount()** to count the number of values in a dynamic array, you do not need to add the boolean expression because **DCount()** returns the proper result (1) if the array has one value with no delimiter.

## See Also

[DCount\(\)](#)

## Example

```
* count the number of items in an @fm-delimited list
cnt = count(List, @fm) + (List# '')
* this is equivalent to:
declare function dcount
cnt = dcount(List, @fm)
```