Remove Statement

Description

Extracts substrings, including fields, values, and subvalues, from dynamic arrays, using ANSI characters 249 through 255 as delimiters.

Syntax

Remove *variable* From string *At column Setting flag*

Parameters

The Remove statement has the following parameters.

Parameter	Description
variable, string	Variable will contain the substring that is extracted from string.
Column	Indicates the starting position of the string to be extracted. It is important to note that the Remove statement changes the column to point to the start of the next substring. The end of the substring occurs when an ANSI character from 249 to 255 is encountered.
Flag	Set with the following values, according to the delimiter found:

Value	Meaning
0	End of string.
1	Record mark ASCII character 255.
2	Field mark ASCII character 254.
3	Value mark ASCII character 253.
4	Subvalue mark ASCII character 252.
5	Text mark ASCII character 251.
6	ASCII character 250.
7	ASCII character 249.

Remove extracts data from a long string faster and more efficiently than does Extract, if doing sequential access through the array.

See also

Extract() function, BRemove statement

Example

```
* This code segment demonstrates the fastest way to sequentially access each element of a dynamic array.
dyn_array = "123": @FM: "678": @FM: "ABC"
position = 1
flag  = ""
Loop
    Remove current_element From dyn_array At position Setting flag
While flag
Repeat
```

Example 2

```
* Change row data into column data
equ ValueMark$ to 3
equ FieldMark$ to 2

authors = "" ; works = ""

dataList = "William Carlos Williams" : @vm : "Paterson" : @fm
dataList := "Eugene O'Neill" : @vm : "Desire Under the Elms" : @fm
dataList := "Barnard Malamud" : @vm : "The Natural" : @fm
dataList := "Mark Twain" : @vm : "The Mysterious Stranger" : @fm
dataList := "Richard Brautigan" : @vm : "Trout Fishing in America" : @fm

vPos = 0 ; vFlag = "" ; itemCtr = 0

loop
   remove thisItem from theList at vPos setting vFlag
   * Create two @vm-delimited arrays - one for Authors, one for Works
   begin case
      case ValueMark$
          authors := thisItem : @vm
      case FieldMark$
         works := thisItem : @vm
   end case
while vFlag
repeat

* Strip trailing @vm
authors[-1,1] = ""
works[-1,1] = ""
```