MatParse Statement

Description

Assigns the value of each successive field of a dynamic array to successive elements in a matrix.

Syntax

MatParse *variable* Into *matrix* [Using *delimiter*]

Parameters

The MatParse statement has the following parameters.

Parameter	Description
variable	Designates the dynamic array that contains the fields to be parsed into the matrix.
matrix	The designated matrix must have been previously named and dimensioned by a Dimension statement. The data contents of each field in the dynamic array will become the data contents of an element in matrix. The first field will be assigned to the first element, the second field will be assigned to the second element, and so on. If the dynamic array has more fields than the matrix has elements, the remaining fields will be assigned to the last element of the array as a dynamic array.
delimiter	Specifies the character that is to be used in the assignment of string elements to matrix elements. If dynamic arrays are being searched, the delimiter should be a field mark (@FM), a value mark (@VM), or a subvalue mark (@SVM). If a Using clause is not specified, a field mark is assumed.

Using a dimensioned array can be more efficient than using a dynamic array, when the number of elements is fixed. Access to any element of a dimensioned array is always faster than access to an element of a dynamic array.

Each element of a dimensioned array can contain a dynamic array within it. For example, if you had a data structure that had a fixed number of elements, but where each element had a varying data structure, you could combine the dynamic array structure within a dimensioned array.

Remarks

```
* Parsing an array into a matrix...
/* The invoice keys are kept in field 5 in the customer row. The value marks are converted to field marks. The
string is then parsed into the INV.KEYS matrix. */
Dim INV.KEYS(20)
Open "CST" To FILE.CUST Else null
Open "INV" To FILE.INV Else null
KEY = 100
read CUST.REC From FILE.CUST, KEY else
     null
end
INV = CUST.REC<5>
Convert @VM To @FM In INV
MatParse INV Into INV.KEYS
Total = 0
I = 1
Loop Until INV.KEYS(I) = "" Or I = 20
     read LINE from FILE.INV, INV.KEYS(I) then
          Total + LINE<6>
     end
I += 1
repeat
```