

# Locate Statement

## Description

Finds the position of a substring in a string, where the substring can be delimited by any ANSI character.

## Syntax

**Locate** *substring* In *string* [Using *delim*] Setting *start* Then | Else *statements*

## Parameters

The Locate statement has the following parameters.

Parameter	Description
<i>substring</i>	Specifies the value whose position is to be located in string.
<i>String</i>	Designates the string that is to be searched.
<i>Delim</i>	Specifies the character that delimits the string. It may be any ANSI character. If dynamic arrays are being searched, <i>delim</i> should be a field mark (@FM), value mark (@VM), or a subvalue mark (@SVM). If a Using clause is not specified, a value mark is assumed. Do not include a <i>delim</i> character in the substring expression.
<i>Start</i>	When the substring has been found in string, the number that corresponds to its position in the string is assigned to <i>start</i> , specified in the Setting clause. The number assigned will relate to the last parameter that was specified in <i>delim</i> . That is, if a field mark is specified, the field number is assigned; if a value mark is specified, the value number is assigned; if a subvalue mark is specified, the subvalue number is assigned. If the substring following Locate cannot be found, the Setting variable identifier will be assigned a value of one greater than the number of positions in the string, and the Else statement is executed.
<i>Then</i>	If the first string is found, the statements following Then are executed.
<i>Else</i>	If the first string is not found, the statements following Else are executed.

## See also

[If, \[ \] \(Brackets operator\)](#), [Extract](#), [Field\(\)](#), [FieldStore\(\)](#), [Index\(\)](#), [InList\(\)](#), [Locate...By](#)

## Remarks

For information about string handling, refer to the "Dynamic Arrays" topic of your system documentation. For sorted data, refer to [Locate...By](#).

## Example

```
/* People is an array of names and phone numbers.
People<1> is a multi-value (@VM-delimited) list of names.
People<2> is a multi-value (@VM-delimited) list of phone numbers. */
Locate "Bob" In People<1> Using @VM Setting POS Then
  * Bob is in the list - delete him
  People = Delete(PEOPLE, 1, POS, 0) ; * delete the name.
  People = Delete(PEOPLE, 2, POS, 0) ; * delete the phone number.
End
```