Find

The *Find* method finds a specified text in a standard or class **module**.

tags: Methods

Applies to ...

Object	Description
Module	The representation of a Basic script.

Syntax

module.`Find(`*target, startline, startcolumn, endline, endcolumn, wholeword, matchcase, patternsearch*`)`

Argument	Optional	Type	Description	Returned value
module		**Module object**	The Basic script	If the string is found, the *Find* method returns `True`.
target	N	String	The text that you want to find.	
startline	N	Long	The line on which to begin searching. If a match is found, the value of the *startline* argument is set to the line on which the beginning character of the matching text is found.	
startcolumn	N	Long	The column on which to begin searching. If a match is found, the value of the *startcolumn* argument is set to the column on which the beginning character of the matching text is found.	
endline	N	Long	The line on which to stop searching. If a match is found, the value of the *endline* argument is set to the line on which the ending character of the matching text is found.	
endcolumn	N	Long	The column on which to stop searching. If a match is found, the value of the *endcolumn* argument is set to the column on which the ending character of the matching text is found.	
wholeword	Y	Boolean	`True` results in a search for whole words only. The default is `False`.	
matchcase	Y	Boolean	`True` results in a search for words with case matching the *target* argument. The default is `False`.	
patternsearch	Y	Boolean	`True` results in a search in which the *target* argument may contain wildcard characters such as an asterisk (*) or a question mark (?). The default is `False`.	

Remarks

- To determine the position in the module at which the search text was found, pass **empty variables** to the *Find* method for the *startline*, *startcolumn*, *endline*, and *endcolumn* arguments.
  If a match is found, these arguments will contain the line number and column position at which the search text begins (*startline*, *startcolumn*) and ends (*endline*, *endcolumn*).
  For example, if the search text is found on line 15, begins at column 20, and ends at column 30, the values of these arguments will be:
  `startline = 15, startcolumn = 20, endline = 15, endcolumn = 30`.
- If the input arguments do not allow a fair search (f.i. *endline* < *startline*), then *Find* will return `False`.
- To pass an empty value, do not initialize the argument or use the `Empty` LO/AOO Basic reserved word.

```
Dim vStartLine As Variant
        vStartLine = Empty
```

Error messages

Argument nr.1 (Value='...') is invalid

See also

Module

Example

Query the properties of a Basic module

```
Const cstModule = "myModule"
Const cstProc = "mySub"
Const vbext_pk_Proc = 0                 '       A Sub or Function procedure
Const cstStringToFind = "some string"

Dim oModule As Object, sProc As String, iProcType As Integer
Dim vStartLine As Variant, vStartColumn As Variant, vEndLine As Variant, vEndColumn As Vari

        Set oModule = Application.AllModules(cstModule)
        With oModule
                DebugPrint "Name = " & .Name
                DebugPrint "# of lines = " & .CountOfLines
                DebugPrint "# of declaration lines = " & .CountOfdeclarationLines
                DebugPrint "Lines 26 to 31 = " & .Lines(26, 6)
                DebugPrint "# of lines in proc " & cstProc & " = " & .ProcCountLines(cstPro
                DebugPrint "Start line in proc " & cstProc & " = " & .ProcStartLine(cstProc
                DebugPrint "Start body line in proc " & cstProc & " = " & .ProcBodyLine(cst
                '       Line 35  is located within procedure sProc (of type iProcType)
                sProc = .ProcOfLine(35, iProcType)
                '       Arguments are left uninitialized to consider the whole module
                If .Find(cstStringToFind, vStartLine, vStartColumn, vEndLine, vEndColumn) T
        End With
        TraceConsole()
```