ProcCountLines

The *ProcCountLines* property eturns the number of lines in a specified procedure in a standard or a class **module**.

tags: Properties

Applies to ...

Object	Description
Module	The representation of a Basic script.

Syntax

module.`ProcCountLines(`*ProcName, ProcKind*`)`

Returned values / Arguments

Argument	Type	Optional	Description	Returned type
module	Object		The concerned module.	Long
ProcName	String	N	The name of a procedure in the module.	
ProcKind	Integer	N	The type of procedure. It can be one of the following constants: vbext_pk_Get vbext_pk_Let vbext_pk_Proc vbext_pk_Set	

Remarks

- A procedure begins with any comments that immediately precede the procedure definition, denoted by one of the following:
  - A `Sub` statement.
  - A `Function` statement.
  - A `Property Get` statement.
  - A `Property Let` statement.
  - A `Property Set` statement.
- The *ProcCountLines* property returns the number of lines in a procedure, beginning with the line returned by the **ProcStartLine** property and ending with the line that ends the procedure. The procedure may be ended with `End Sub`, `End Function`, or `End Property`.
- If useful, insert the following constants in your own code:

```
Const vbext_pk_Get = 1                  '       A Property Get procedure
Const vbext_pk_Let = 2                  '       A Property Let procedure
Const vbext_pk_Proc = 0                 '       A Sub or Function procedure
Const vbext_pk_Set = 3                  '       A Property Set procedure
```

Error messages

Property 'ProcCountLines' not applicable in this context
Argument nr. ... [Value = '...'] is invalid

See also

ProcBodyLine
ProcOfLine
ProcStartLine

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()
```

Bookmark this page » » ProcCountLines