ProcOfLine

The *ProcOfLine* property returns the name of the procedure that contains a specified line in a standard or a class **module**.

tags: Properties

Applies to ...

Object	Description
Module	The representation of a Basic script.

Syntax

module.`ProcOfLine(`*Line, ProcKind*`)`

Returned values / Arguments

Argument	Type	Optional	Description	Returned type
module	Object		The concerned module.	String
Line	Long	N	The number of a line in the module.	
ProcKind	Integer	N	The type of procedure. After execution it will contain one of the following constants: vbext_pk_Get vbext_pk_Let vbext_pk_Proc vbext_pk_Set	

Remarks

- Lines in a module are numbered beginning with 1.
- For any given line number, the *ProcOfLine* property returns the name of the procedure that contains that line. Since comments immediately preceding a procedure definition are considered part of that procedure, the *ProcOfLine* property may return the name of a procedure for a line that isn't within the body of the procedure. The **ProcStartLine** property indicates the line on which a procedure begins; the **ProcBodyLine** property indicates the line on which the procedure definition begins (the body of the procedure).
- Note that the *ProcKind* argument indicates whether the line belongs to a `Sub` or `Function` procedure, a `Property Get` procedure, a `Property Let` procedure, or a `Property Set` procedure. To determine what type of procedure a line is in, pass a variable of type `Integer` to the **ProcOfLine** property, then check the value of that variable.
- 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 'ProcOfLine' not applicable in this context
Argument nr. ... [Value = '...'] is invalid

See also

ProcBodyLine
ProcCountLines

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()
```

in a standard or a class **module**.

Bookmark this page » » ProcOfLine