ASP Code Count™

Counting Standard

University of Southern California

Center for Systems and Software Engineering

December , 2016

Revision Sheet

Date	Version	Revision Description	Author
8/25/2016	1.0	Original Release	Matthew Swartz

Table of Contents

1. Definitions

1.1. **SLOC –** Source Lines of Code is a unit used to measure the size of software program. SLOC counts the program source code based on a certain set of rules. SLOC is a key input for estimating project effort and is also used to calculate productivity and other measurements.

1.2. **Physical SLOC –** One physical SLOC is corresponding to one line starting with the first character and ending by a carriage return or an end-of-file marker of the same line, and which excludes the blank and comment line.

1.3. **Logical SLOC –** Lines of code intended to measure "statements", which normally terminate by a semicolon (C/C++, Java, C#) or a carriage return (VB, Assembly), etc. Logical SLOC are not sensitive to format and style conventions, but they are language-dependent.

1.4. **Data declaration line or data line –** A line that contains declaration of data and used by an assembler or compiler to interpret other elements of the program.

The following table lists ASP data keywords:

boolean	byte	collection	const	currency
date	dim	double	integer	item
long	new	object	option	private
public	redim	single	static	string
time	variant			

ASP DATA KEYWORDS

Data declarations in Visual Basic Script are of the form :

```
Dim <variable name>
Const <constant name>
Static <variable name>
```

1.5. **Compiler Directives –** A statement that tells the compiler how to compile a program, but not what to compile.

The following table lists the ASP keywords that denote data declaration lines:

#externalsource	#elseif	#region	#const
#else	#end	#if	

Compiler Directives

1.6. **Blank Line –** A physical line of code, which contains any number of white space characters (spaces, tabs, form feed, carriage return, line feed, or their derivatives).

1.7. **Comment Line –** A comment is defined as a string of zero or more characters that follow language-specific comment delimiter.

ASP comment delimiters are is '. A whole comment line may span one line and does not contain any compilable source code. An embedded comment can co-exist with compilable source code on the same physical line. Banners and empty comments are treated as types of comments.

1.8. **Executable Line of code –** A line that contains software instruction executed during runtime and on which a breakpoint can be set in a debugging tool. An instruction can be stated in a simple or compound form.

- An executable line of code may contain the following program control statements:
  - Selection statements (If, Select)
  - Iteration statements (for, while, until)
  - Jump statements (break, exit)
  - Expression statements (procedure/function calls, assignment statements, operations, etc.)
- An executable line of code may not contain the following statements:
  - Compiler directives
  - Data declaration (data) lines
  - Whole line comments, including empty comments and banners
  - Blank lines

2. Checklist for source statement counts

PHYSICAL SLOC COUNTING RULES			
MEASUREMENT UNIT	ORDER OF PRECEDENCE	PHYSICAL SLOC	COMMENTS
Executable lines	1	One per line	Defined in 1.8
Non-executable lines			
Declaration (Data) lines	2	One per line	Defined in 1.4
Compiler directives	3	One per line	Defined in 1.5
Comments			Defined in 1.7
On their own lines	4	Not included	
Embedded	5	Not included	
Banners	6	Not Included	
Empty comments	7	Not Included	
Blank lines	8	Not Included	Defined in 1.6

LOGICAL SLOC COUNTING RULES				
NO.	STRUCTURE	ORDER OF PRECEDENCE	LOGICAL SLOC RULES	COMMENTS
R01	If/Elseif condition Then statement Else statement Endif Select var Case cond:statement . . Case Else :statement End Select	1	Count once.	
R02	do while (...) statement loop for (...) statement next do statements until(...) while(...) statements wend	2	Count once.	.
R03	Block delimiters Private Sub End Sub	3	Count once per pair of Private Sub and End Sub	
R04	Compiler Directive	5	Count once per directive	

3. Examples

EXECUTABLE LINES

SELECTION Statement

ESS1 - If, ElseIf, Else and nested if statements

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
If conditionThen	If (total = firstnum + secondnum	1
<statement>	And Val(sum.Text) <> 0) Then	0
End If	correct.Visible = True	1
	wrong.Visible = False	1
If condition Then	End If	0
<statement>		
Else	If (total = firstnum + secondnum	1
<statement>	And Val(sum.Text) <> 0) Then	0
End If	correct.Visible = True	1
	wrong.Visible = False	1
If condition1 Then	Else	0
<statement>	correct.Visible = False	1
Else If condition2 Then	wrong.Visible = True	1
<statement>	End If	0
Else		
<statement>	If (total = firstnum + secondnum	1
End If	And Val(sum.Text) <> 0) Then	0
	correct.Visible = True	1
NOTE: complexity is not considered, i.e. multiple “And” or “Or” as part of the expression.	wrong.Visible = False	1
	Else If (total = firstnum –	1
	secondnum) Then	0
	correct.Visible = False	1
	wrong.Visible = True	1
	Else	0
	correct.Visible = False	1
	wrong.Visible = True	1
	End If	0

ESS2 – SELECT CASE STATEMENTS

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT

Select Case Case <constant 1> : <statements> Case Else <statements> End Select	Select Case Err.Num Case 53 'File not found answer=MsgBox("File not found. Try again?", _ vbYesNo) Case 76 'Path not found answer=MsgBox("Path not found. Try again?", _vbYesNo) Case Else 'unknown error MsgBox "Unknown error. Quitting now." 'SHOULD LOG ERROR! Unload Me End Select	1 0 0 0 1 0 0 0 1 0 0 0 1 1 0

ITERATION Statement

EIS1 - FOR

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
For num = 1 To 10 STATEMENTS Next	For num = 1 To 10 studentName(num)= 999 Next	1 1 0

EIS2 – For Each …In statements

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
For Each <var> In <vars> STATEMENTS Next	For Each x In cars response.write(x & " ") Next	1 1 0

EIS3 – Do While

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT

Do While condition
Statements
Loop
	Do While counter <=1000
	num.Text = counter
	counter = counter+1
	Loop
		1
		1
		1
		0

EIS4 – do-while

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT

do
{
<statements>;
} while (<boolean expression>);
	Do
	{
	Console.Writeline(“i”);
	} while (x < 5);
		0
		0
		1
		1

EXPRESSION Statement

EES1 - FUNCTION CALL

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<function_name> (<parameters>)	call vbproc(3,4)	1

EES2 - assignment statement

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<name> = <value>;	cars(0) = "Volvo"	1

DECLARATION OR DATA LINES

DDL1 – subroutine/function declaration, variable declaration

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
Private Sub Name(var_list) Statements End Sub Dim <var> As <type>	Sub Start_Click() Form1.Cls addName End Sub Dim var As String	1 1 1 0 1

COMPILER DIRECTIVES

CDL1 - directive types

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
#<directive>	#const MAX	1

4. Complexity

Complexity measures the occurrences of different keywords in code baseline. Below table identifies the categories and their respective keywords that are counted as part of the complexity metrics.

Math Functions	Trig	Log	Calculations	Conditionals	Logic	Pre-processor	Assignment	Pointer
randomize	atan	log	+	for each	andalso	#externalsource	=	
round	cos		-	do while	isfalse	#elseif		
sign	sin		*	do until	orelse	#region		
sqrt	tan		/	elseif	istrue	#const		
abs	atan		\	select	and	#else		
exp			^	while	not	#end		
rnd				case	xor	#if		
				for	<>			
				if	>=			
					=<			
					or			
					>			
					<			