VB Code Count™ Counting Standard

University of Southern California

Center for Systems and Software Engineering

OCT, 2007

Revision Sheet

Date	Version	Revision Description	Author
10/17/07	1.0	Original Release	CSSE

1.0 CHECKLIST FOR SOURCE STATEMENT COUNTS

PHYSICAL AND LOGICAL SLOC COUNTING RULES

Measurement Unit	Order of Precedence	Physical SLOC	Logical SLOC	Comments
Executable lines	1	One per line	See table below	Defined in 2.7
Non-executable lines				
Declaration (Data) lines	2	One per line	See table below	Defined in 2.4
Comments				Defined in 2.6
On their own lines	4	Not included (NI)	NI	
Embedded	5	NI	NI	
Banners	6	NI	NI	
Empty comments	7	NI	NI	
Blank lines	8	NI	NI	Defined in 2.5
Compiler Directives	3	One per line	See table below	Defined in 2.8

Table 1 Physical and Logical SLOC Counting Counts

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.	

R03	*do while* (…) statement loop
	for (….)
	statement
	next
	do
	statements
	until(…..)
	while(…..)
	statements
	wend
		2	Count once.	

R04	Block delimiters Private Sub	3	Count once per pair of Private Sub	

No.	Structure	Order of Precedence	Logical SLOC Rules	Comments
	End Sub		and End Sub	
R05	Compiler directive	4	Count once per directive.	

Table 2 Logical SLOC Counting Rules

2.0 DEFINITIONS

2.1 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.

2.2 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.

2.3 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), or a new line in a stored procedure or a function in SQL etc. Logical SLOC are not sensitive to format and style conventions, but they are language-dependent.

2.4 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 VB keywords that denote data declaration lines:

Class	Const	Declare	Delegate
Dim	Enum	Event	Function
Interface	Module	Operator	Property`
Structure	Sub		

Table 1 Data Declaration Types

2.5 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).

2.6 Comment line – – A comment is defined as a string of zero or more characters that follow language-specific comment delimiter.
VB comment delimiter is "' ". A whole comment line may span only one line if it exceeds then we should start the next line with the comment delimiter

2.7 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:
  - Control statements (if)
  - Iteration statements (loop)
  - Jump statements
  - Procedure/Function Calls

- An executable line of code may not contain the following statements:
  - Compiler directives
  - Data declaration (data) lines
  - Whole line comments
  - Blank lines

2.8 Compiler directive –

A list of VB directives is presented as follows:

- Define a compiler constant: #Const
- Compile selected blocks of code: #If...Then...#Else
- Collapse and hide sections of code: #Region
- Indicate a mapping between source lines and text external to the source: #ExternalSource

3.0 EXAMPLES OF LOGICAL SLOC COUNTING

EXECUTABLE LINES

SELECTION STATEMENTS				
ID	**STATEMENT DESCRIPTION**	**GENERAL FORM**	**SPECIFIC EXAMPLE**	**SLOC COUNT**

ESS1	if, else if, else and
	nested if statements
		if *condition*
		code to be executed if condition is true
		end if
		
		if *condition*
		code to be executed if condition is true
		else
		code to be executed if condition is not true
		end if
		
		if *condition1*
		code to be executed if condition1 is true
		else if *condition2*
		code to be executed if condition2 is true
		else
		code to be executed if condition is not true
		end if
		
		NOTE: complexity is not considered, i.e. multiple "AND" or "OR" as part of the expression.
			If total = firstnum + secondnum And Val(sum.Text) <> 0 Then
			correct.Visible = True
			wrong.Visible = False
			End If
			
			If total = firstnum + secondnum And Val(sum.Text) <> 0 Then
			correct.Visible = True
			wrong.Visible = False
			Else
			correct.Visible = False
			wrong.Visible = True
			End If
			
			If total = firstnum + secondnum And Val(sum.Text) <> 0 Then
			correct.Visible = True
			wrong.Visible = False
			Else If total = firstnum – secondnum then
			correct.Visible = False
			wrong.Visible = True
			Else
			correct.Visible = False
			wrong.Visible = True
			End If
				1
				
				1
				1
				0
				
				1
				
				1
				1
				0
				1
				1
				0
				
				1
				
				1
				1
				1
				
				1
				1
				0
				1
				1
				0

ESS2	Select Case	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 1 1 1 1 1 1

				1 1 1
ESS3	Error Handler	OnError Goto	Private Sub CodeWithErrorHandler() On Error GoTo ErrHandler '...Procedure code ... '...	1 1 1 0

ITERATIONS STATEMENTS

ID	STATEMENT DESCRIPTION	GENERAL FORM	SPECIFIC EXAMPLE	SLOC COUNT
EIS1	for	For num = 1 To 10 STATEMENTS Next	For num = 1 To 10 studentName(num)= 999 Next	1 1 0
EIS2	while	While condition Statements Wend	While Not IsEmpty(ActiveCell) MsgBox ActiveCell.Value ActiveCell.Offset(1, 0).Select Wend	1 1 1 1 0
EIS3	Do - until	Do Statements Until condition	Do MsgBox ActiveCell.Value ActiveCell.Offset(1, 0).Select Until IsEmpty(ActiveCell)	0 1 1

EIS4	do-while	Do while condition
		Statements
		Loop
			Do while counter <=1000
			num.Text=counter
			counter =counter+1
			Loop
				1
				1
				1
				0

JUMP STATEMENTS
(are counted as they invoke action – pass to the next statement)

ID	STATEMENT DESCRIPTION	GENERAL FORM	SPECIFIC EXAMPLE	SLOC COUNT

EJS1	exit	exit *sub*;	Exit Sub;	1

DECLARATION (DATA) LINES

ID	STATEMENT DESCRIPTION	GENERAL FORM	SPECIFIC EXAMPLE	SLOC COUNT
DDL1	Sub routine Declaration	Private Sub Name(var_list) Statements End Sub	Private Sub Start_Click() Form1.Cls addName End Sub	1 1 1 0

COMPILER DIRECTIVES

ID	STATEMENT DESCRIPTION	GENERAL FORM	SPECIFIC EXAMPLE	SLOC COUNT
CDL1	directive types	#Const Directive	#Const Directive	1