ColdFusion CodeCount™

Counting Standard

University of Southern California

Center for Systems and Software Engineering

May , 2010

Revision Sheet

Date	Version	Revision Description	Author
5/11/2010	1.0	Original Release	CSSE
1/2/2013	1.1	Updated document template	CSSE

Table of Contents

No.	Contents	Page No.
1.0	Definitions	4
	1.1 SLOC	4
	1.2 Physical SLOC	4
	1.3 Logical SLOC	4
	1.4 Data declaration line	4
	1.5 Compiler directive	4
	1.6 Blank line	4
	1.7 Comment line	4
	1.8 Executable line of code	5
2.0	Checklist for source statement counts	6
3.0	Examples of logical SLOC counting	7
	3.1 Executable Lines	7
	3.1.1 Selection Statements	7
	3.1.2 Iteration Statements	8
	3.1.3 Jump Statements	8
	3.1.4 Expression Statements	9

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 a ColdFusion server to determine all ColdFusion variables declared in the program.

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

ColdFusion comment delimiters are "<!---" and "--->". A whole comment line may span one line and does not contain any compliable source code. An embedded comment can co-exist with compliable 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, switch)
  - Iteration statements (foreach, loop)
  - Empty statements (pass)
  - Jump statements (return, goto, exit function)
  - Expression statements (function calls, assignment statements, operations, etc.)
  - Block statements
  - Database statements
- An executable line of code may not contain the following statements:
  - 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	NA	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	All ColdFusion tags beginning with “cf” with no nesting like <cfform>, <cfinput>, etc.	1	Count once	All occurrences of such tags until corresponding end tags </> is assumed to be one logical statement
R02	<cfcase>, <cfloop>, <cfswitch>, either all tags having multiple steps of execution statement	2	Count once	Logically different tags on the same line are to be counted independently
R03	Comment delimiter	3	Count once per combination of start tag and end tag statement, including empty statement.	Comments in ColdFusion are similar to HTML comments<!--- this is a comment --->
R04	Compiler directive	4	NA	NA

3. Examples

EXECUTABLE LINES

SELECTION Statements

ESS1 - cfif, cfelseif, cfelse and nested cfif statements

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cfif *expression*>	<cfif name == "USC">	1
statements	some logic	0
</cfif>	</cfif>	0
<cfif *expression*>	<cfif password == "name">	1
statements	some code	0
<cfelse>	<cfelse>	1
statements	statements	0
</cfif>	</cfif>	0
<cfif *expression*>	<cfif num > 0>	1
statements	some code	0
<cfelseif *expression*>	<cfelseif num < 0>	1
statements	statements	0
<cfelse>	<cfelse>	1
statements	code	0
</cfif>	</cfif>	0

ESS2 - cfswitch, cfcase, cfdefaultcase

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cfswitch *expression = "expression"*>	<cfswitch expression = "#State#">	1
<cfcase *value ="value"*>	<cfcase value="CA">	1
HTML or CFML code	California	0
</cfcase>	</cfcase>	0
<cfdefaultcase>	<cfdefaultcase>	1
HTML or CFML code	one of the other 47 states	0
</cfdefaultcase>	</cfdefaultcase>	0
</cfswitch>	</cfswitch>	0

ESS3 - cftry-cfcatch

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cftry >	<cftry>	1
do something	try: 1/0	0
</cftry>	some code	0
<cfcatch>	</cftry>	0
cleanup	<cfcatch ZeroError>	1
</cfcatch>	some code	0
	</cfcatch>	0

ITERATION Statements

EIS1 - cfloop

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cfloop *expression*>	<cfloop index = "LoopCount" from = 1 to =	1
statements	5>	0
</cfloop>	The loop index is	0
	<cfoutput>#LoopCount#</cfoutput>.	1
	<br>	0
	</cfloop>	0
	<cfloop condition =”Expression”>	1
	<cfloop>	0
	<cfloop query =”Query name”	1
	startRow =”Start Row value”	0
	endRow = “ End Row value”	0
	</cfloop>	0

JUMP Statements
(are counted as they invoke action-pass to the next statement)

EJS1 – cfreturn

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cfreturn *expression*>	<cfreturn true>	1

EJS2 – cfbreak

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cfbreak>	<cfloop expression>	1
	<cfbreak>	1
	</cfloop>	0

EJS3 - cfexit

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cfexit>	<cfloop expression> <cfexit> </cfloop>	1 1 0

EJS4 - cfcontinue

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cfcontinue>	<cfloop expression> <cfcontinue> </cfloop>	1 1 0

EXPRESSION Statements

EES1 – function call

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cffunction > name = “function name” description “ function description” return Type=”Data to be returned </cffuntion>	Any example	1

EES2 – assignment statement

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
<cfset *variable_name=”value”*>	<cfset age=”22”>	1

EES3 - CFScript

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT

<cfscript>
CFScript code
</cfscript>
	<cfscript>
	for (i=1 ; i LE 4; i = i+1) {
	if(find("key",strings[i],1))
	break;
	}
	</cfscript>
		1
		0 (3 scr)
		0
		0
		0
		0

EES4 – database query

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT

<cfquery datasource=”DB name” name=
“Some name”>
query or stored procedure call
</cfquery>
	<cfquery datasource=“ABCD”
	name=”Student Records”>
	SELECT * FROM Students
	</cfquery>
		1
		0
		0 (1 sql)
		0