HTML/XML CodeCount™

Counting Standard

University of Southern California

Center for Systems and Software Engineering

October , 2007

Revision Sheet

Date	Version	Revision Description	Author
10/22/2007	1.0	Original Release	CSSE
10/20/2008	1.1	Added example E04	CSSE
1/2/2013	1.2	Updated document template	CSSE

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. For HTML/XML, logical statements are based on tags.

1.4. **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).

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

HTML/XML 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.

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 2.7
Non-executable lines			
Declaration	2	One per line	Defined in 2.4
Comments			Defined in 2.6
On their own lines	4	Not included (NI)	
Embedded	5	NI	
Banners	6	NI	
Empty comments	7	NI	
Blank lines	8	NI	Defined in 2.5
Compiler Directives	N/A	N/A	N/A

LOGICAL SLOC COUNTING RULES				
NO.	STRUCTURE	ORDER OF PRECEDENCE	LOGICAL SLOC RULES	COMMENTS
R01	• Declarations (document type, attribute, entity, text, notation, etc.) • Processing instruction • CDATA section, and • Conditional section	1	Count once per occurrence	
R02	A pair of start-tag and end-tag	2	Count once	
R03	Empty element tag	3	Count once per occurrence	

3. Examples

DECLARATION

D01 – processing instruction

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
'<?' PITarget (S (Char* - (Char* '?>' Char*)))? '?>'	<?xml version="1.0"?>	1

D02 – document type

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
'<!DOCTYPE' S Name (S ExternalID)? S? ('[' (markupdecl | DeclSep)* ']' S?)? '>'	<?xml version="1.0"?> <!DOCTYPE greeting SYSTEM "hello.dtd">	1 1

D03 – element type

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
'<!ELEMENT' S Name S contentspec S? '>'	<!ELEMENT br EMPTY> <!ELEMENT p (#PCDATA|emph)* > <!ELEMENT %name.para; %content.para; > <!ELEMENT container ANY>	1 1 1 1

D04 – attribute type

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
'<!ATTLIST' S Name AttDef* S? '>'	<!ATTLIST termdef id ID #REQUIRED name CDATA #IMPLIED> <!ATTLIST form method CDATA #FIXED "POST">	1 1

D05 - entity

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
'<!ENTITY' S Name S EntityDef S? '>'	<!ENTITY % YN '"Yes"' > <!ENTITY WhatHeSaid "He said %YN;" > 	1 1

ELEMENT

E01 – start-tag and end-tags, empty element

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
'<' Name (S Attribute)* S? '>'	<?xml version="1.0" encoding="ISO8859-1" ?> - <note> <to>Tove</to> <from>Jani</from> <heading>Reminder</heading> <body>Don't forget me this weekend!</body> </note>	1 1 1 1 1 0
'<' Name (S Attribute)* S? '/>'	<HTML> <BODY> <P> to break lines in a paragraph, use the br tag. </P> </BODY> </HTML>	1 1 1 2 2 0 1 0 0

E02 - comment delimiter

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
Comments in XML are done in the same manner as HTML comments <!-- this is a comment -->	<html> <!--To check the lines--> <body>The content of the body element is displayed in your browser.</body> </html> <?xml version="1.0" encoding="ISO8859-1" ?> <message to="you@yourAddress.com" from="me@myAddress.com" subject="XML Is Really Cool"> <!-- This is a comment --> <text> How many ways is XML cool? Let me count the ways... </text> </message>	1 1 0 1 1 0 1 0 0

E03 - string delimiter(s), literals, escape characters, nesting, etc.

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
Escape Character –Both Entity and character references may be used to escape the delimiters. XML- < > & ' " HTML 4 DTD explicitly declares 252 character entities For E.g. ¢ &curren Numeric character - references can also be used in XML; they are expanded immediately when recognized and must be treated as character data.	<?xml version="1.0" encoding="utf-8"?> <string xmlns="http......"> < DataSet> <Order> <Customer >439 </Customer> </Order> </DataSet></string> <html> <body>The content of the body element is displayed in your browser.</body> <tt> C H E E S E </tt> </html>	1 1 1 1 2 0 0

E04 – tags with attributes

GENERAL EXAMPLE	SPECIFIC EXAMPLE	SLOC COUNT
XML and HTML can have attributes in the start tag. Attributes provide additional information about the element	<note date="12/11/2002"> <to>Tove</to> <from>Jani</from> <heading>Reminder</heading> <body>Don't forget me this weekend!</body> </note>	1 1 1 1 1 0 0