# XML/HTML Code Count™ Counting Standards

*University of Southern California*

**Center for Systems and Software Engineering**

OCT, 2007

## Revision Sheet

| Date | Version | Revision Description | Author |
|---|---|---|---|
| 10/22/07 | 1.0 | Original Release | CSSE |
| 10/20/08 | 1.1 | Added example E04. | 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 | 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 | N/A | N/A | N/A | N/A |

**Table 1 Physical and Logical SLOC Counting Counts**

### LOGICAL SLOC COUNTING RULES

| No. | Structure | Order of Precedence | Logical SLOC Rules | Comments |
|---|---|---|---|---|
| R01 | • Declarations (document type, attribute, entity, text, notation, etc)<br>• Processing instruction<br>• CDATA section, and<br>• 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 | |

**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** – IT is language specific which measures the number of "statements“dependent on language syntax

A count of “blank lines, comment lines" in the text of the program's source code

**2.4 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).

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

## 3.0 EXAMPLES OF SLOC COUNTING

| DECLARATION | | | | |
|---|---|---|---|---|
| **ID** | **STATEMENT DESCRIPTION** | **GENERAL FORM** | **SPECIFIC EXAMPLE** | **SLOC COUNT** |
| D01 | Processing instruction | '<?' PITarget (S (Char* - (Char* '?>' Char*)))? '?>' | <?xml version="1.0"?> | 1 |
| D02 | Document type | '<!DOCTYPE' S Name (S ExternalID)? S? ('[' (markupdecl \| DeclSep)* ']' S?)? '>' | <?xml version="1.0"?><br><!DOCTYPE greeting SYSTEM "hello.dtd"> | 1<br>1 |
| D03 | Element type | '<!ELEMENT' S Name S contentspec S? '>' | <!ELEMENT br EMPTY><br><!ELEMENT p (#PCDATA\|emph)* ><br><!ELEMENT %name.para; %content.para; ><br><!ELEMENT container ANY> | 1<br>1<br>1<br>1 |
| D04 | Attribute type | '<!ATTLIST' S Name AttDef* S? '>' | <!ATTLIST termdef<br>id ID #REQUIRED<br>name CDATA #IMPLIED><br><br><!ATTLIST form<br>method CDATA #FIXED "POST"> | 1<br><br><br><br>1 |
| D05 | Entity | '<!ENTITY' S Name S EntityDef S? '>' | <!ENTITY % YN '"Yes"' ><br><!ENTITY WhatHeSaid "He said %YN;" > | 1<br>1 |

| ELEMENT | | | | |
|---|---|---|---|---|
| **ID** | **STATEMENT DESCRIPTION** | **GENERAL FORM** | **SPECIFIC EXAMPLE** | **SLOC COUNT** |
| E01 | Start-tag & end-tags<br><br>Empty element | `'<' Name (S Attribute)* S? '>'`<br><br>`'<' Name (S Attribute)* S? '/>'` | <?xml version="1.0"   encoding="ISO8859-1" ?>  -<br><note><br>  <to>Tove</to><br>  <from>Jani</from><br>  <heading>Reminder</heading><br>  <body>Don't forget me this weekend!</body><br></note><br><br><HTML><br><BODY><br><br><P><br>to break<br>lines<br>in<br>a<br>paragraph,<br>use the br tag.<br></P><br><br/><br></BODY><br></HTML> | 1<br>1<br>1<br>1<br>1<br>1<br><br><br>1<br>1<br><br>1<br>2<br>2<br><br>1 |
| E02 | Comment Delimiter | Comments in XML are done in the same manner as HTML comments<br>**<!-- this is a comment -->** | <html><br><!--To check the lines--><br><body>The content of the body element is displayed in your browser.</body><br></html><br><br><?xml version="1.0" encoding="ISO8859-1" ?><br><message to="you@yourAddress.com" from="me@myAddress.com"<br>  subject="XML Is Really Cool"><br>  <!-- This is a comment --><br>  <text><br>    How many ways is XML cool? Let me count the ways...<br>  </text><br></message> | 1<br><br>1<br><br><br>1<br>1<br><br><br>1 |
| E03 | String delimiter(s); literals; escape characters; nesting, etc. | Escape Character –Both Entity and character references may be used to escape the delimiters.<br>XML- &lt; &gt; &amp; &apos; &quot;<br>HTML 4 DTD explicitly declares 252 character | <?xml version="1.0" encoding="utf-8"?><br><string xmlns="http................"><br>&lt; DataSet&gt;<br>&lt;Order&gt;<br>&lt;Customer<br>&gt;439<br>&lt;/Customer&gt;<br>&lt;/Order&gt;<br>&lt;/DataSet&gt;</string> | 1<br>1 |

| | | | | |
|---|---|---|---|---|
| | | entities<br>For E.g. &cent; &nbsp; &curren<br>Numeric character - references can also be used in XML; they are expanded immediately when recognized and must be treated as character data. | <html><br><body>The content of the body element is displayed in your browser.</body><br><b><tt><br>C &nbsp; H &nbsp; E &nbsp; E &nbsp; S &nbsp; E<br></tt></b><br></html> | 1<br>1<br><br><br>2 |
| E04 | Tags with Attributes | XML and HTML can have attributes in the start tag. Attributes provide additional information about the element | <note date="12/11/2002"><br><to>Tove</to><br><from>Jani</from><br><heading>Reminder</heading><br><body>Don't forget me this weekend!</body><br></note> | 1<br>1<br>1<br>1<br>1 |