# ISO/IEC JTC 1/SC 22/OWGV N 0114

*Automatically generated code*

| | |
|---|---|
| **Date** | 15 December 2007 |
| **Contributed by** | Robert Seacord |
| **Original file name** | Email note to mailer dated 13 December 2007 |
| **Notes** | Addresses action item #07-01 |

## Moore, Jim

**From:** Robert C. Seacord [rcs@CERT.ORG]
**Sent:** Thursday, December 13, 2007 12:05 PM
**To:** sc22-owgv-list ISO/IEC JTC 1/SC 22/OWGV Standards Development
**Subject:** [SC22-OWGV] Automatically Generated Code

This document is response to an action item I took today at the OWGV meeting to clarify the applicability of coding guidelines to various classes of software.

---

Secure coding guidance may vary depending on whether code is hand-coded versus automatically generated. Categories of code include:

* Tool-generated, tool-maintained - code which is specified and maintained in a higher-level format, from which language specific source code is generated. The source code is generated from this higher level description and then provided as input to the language compiler. The generated source code is never viewed or modified by the programmer.
* Tool-generated, hand-maintained - code which is specified and maintained in a higher-level format, from which language specific source code is generated. It is expected or anticipated that at some point in the development cycle, however, that the tool will cease to be used and that the generated source code will be visually inspected and/or manually modified and maintained.
* Hand-coded - code that has been manually written by a programmer using a text editor or interactive development environment where the programmer maintains source code directly in the source code format which is provided to the compiler.
* JIT compilation - also known as dynamic translation, is a technique for improving the runtime performance of a computer program. JIT converts code at runtime prior to executing it natively, for example bytecode into native machine code.
* Interpreted execution - Interpretation is one of the two major ways in which a programming language can be implemented, the other being compilation. The term interpreter may refer to the program that executes source code that has already been translated to some intermediate form, or it may refer to the program that performs both the translation and execution.

Source code that is written and maintained by hand needs to have the following properties:

* readability
* program comprehension

These requirements do not exist for source code that is never handled directly by a programmer, although requirements for correct behavior are still applicable. Readability and program comprehension requirements exist for the source code which is hand-coded, regardless of if this source code is interpreted, compiled, or compiled just-in-time (JIT). Reading and comprehension requirements apply to code that is tool-generated but hand-maintained, but does not apply to code that is tool-generated and tool-maintained. Readability and program comprehension requirements do not apply to intermediate forms such as pcode, assembly language, byte-codes that are never meant to be maintained by the programmer. Adequate verification must be performed to ensure that the safety and security properties of the code are maintained at each level of translation and interpretation although this is outside the scope of these language standards.