Oracle: The Java® Virtual Machine Specification JVM

return to Oracle de facto Standards

Table 1: Data sheet for The Java® Virtual Machine Specification

Title	The Java® Virtual Machine Specification
Acronym	JVM
Version	Java SE 8 Edition
Document Number	
Release Date	13 February 2013
Reference	https://docs.oracle.com/javase/specs/jvms/se8/html/

Note: The following is an excerpt from the official site. It is provided here as a convenience and is not authoritative. Refer to the original document as the authoritative reference.

The Java Virtual Machine

The Java Virtual Machine is the cornerstone of the Java platform. It is the component of the technology responsible for its hardware- and operating system-independence, the small size of its compiled code, and its ability to protect users from malicious programs.

The Java Virtual Machine is an abstract computing machine. Like a real computing machine, it has an instruction set and manipulates various memory areas at run time. It is reasonably common to implement a programming language using a virtual machine; the best-known virtual machine may be the P-Code machine of UCSD Pascal.

The first prototype implementation of the Java Virtual Machine, done at Sun Microsystems, Inc., emulated the Java Virtual Machine instruction set in software hosted by a handheld device that resembled a contemporary Personal Digital Assistant (PDA). Oracle's current implementations emulate the Java Virtual Machine on mobile, desktop and server devices, but the Java Virtual Machine does not assume any particular implementation technology, host hardware, or host operating system. It is not inherently interpreted, but can just as well be implemented by compiling its instruction set to that of a silicon Central Processing Unit (CPU). It may also be implemented in microcode or directly in silicon.

The Java Virtual Machine knows nothing of the Java programming language, only of a particular binary format, the class file format. A class file contains Java Virtual Machine instructions (or bytecodes) and a symbol table, as well as other ancillary information.

For the sake of security, the Java Virtual Machine imposes strong syntactic and structural constraints on the code in a class file. However, any language with functionality that can be expressed in terms of a valid class file can be hosted by the Java Virtual Machine. Attracted by a

generally available, machine-independent platform, implementors of other languages can turn to the Java Virtual Machine (VM) as a delivery vehicle for their languages.

The Java Virtual Machine specified here is compatible with the Java SE 8 platform, and supports the Java programming language specified in The Java Language Specification, Java SE 8 Edition.