

Java Platform Debugger Architecture (JPDA)

Sang Shin
www.javapassion.com

Topics

- What is JPDA?
- JPDA architecture
- Using JPDA

What is JPDA?

What is JPDA?

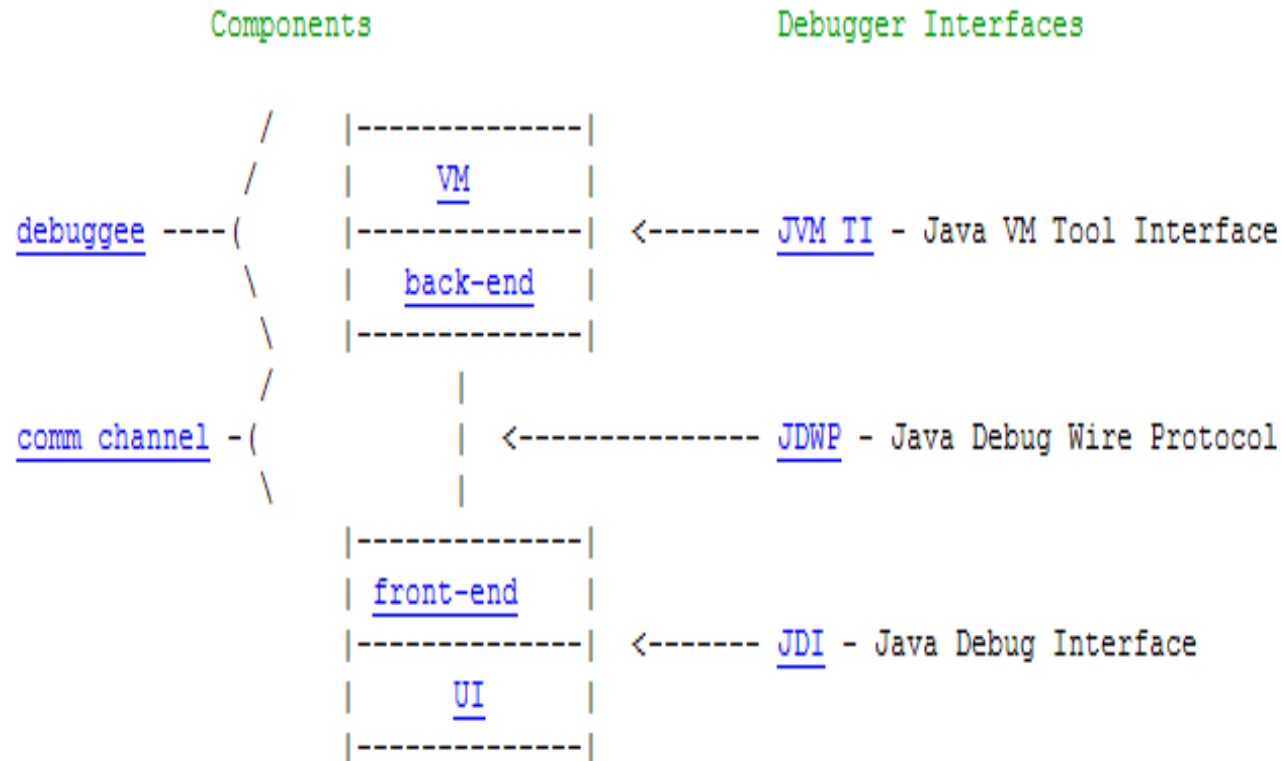
- Java Platform Debugger Architecture
- JPDA is a multi-tiered debugging architecture that allows tools developers to easily create debugger applications which run portably across platforms, virtual machine (VM) implementations and JDK versions.

Goals of JPDA

- To provide standard interfaces which allow Java programming language debugging tools to be easily written without regard to platform specifics such hardware, operating system and virtual machine implementation.
- To describe a complete architecture for implementing these interfaces, including remote and cross-platform debugging.
- To provide a reference implementation of this architecture.
- To provide a highly modular architecture where the implementation and/or client of an interface can be different than the reference implementation or different from the JPDA component.

JPDA Architecture

JPDA Architecture



JPDA Architecture

- JVM TI (Java VM Tool Interface)
 - > JVM TI is a new interface introduced in J2SE 5.0 which replaces JVMDI.
 - > It defines the debugging services a VM provides.
- JDWP (Java Debug Wire Protocol)
 - > Defines the communication between debuggee and debugger processes.
- JDI (Java Debug Interface)
 - > Defines a high-level Java language interface which tool developers can easily use to write remote debugger applications.

JPDA is Layered Architecture

- Reference implementation includes
 - > JVM TI implementations on multiple Sun VMs (see VM documentation).
 - > A back-end which uses JVM TI to implement the debuggee side of JDWP.
 - > A front-end which uses the debugger side of JDWP to implement JDI.
 - > Two simple example debugger applications which are built on JDI.
- Implementations may be substituted.

Using JPDA

Using JPDA

- A debugger developer may hook into JPDA at any layer.
- Since the JDI is the highest level and easiest to use we encourage developers to use this interface.
 - > Example scenario: Suppose a company develops a debugger using JDI. They can use it with the reference implementation and it will automatically work with the VMs and platforms Sun supports. It can also work, for example, with the reference implementation front-end and a debuggee running another company's VM that implements JDWP (which might use or by-pass JVM TI).

Using JPDA (Continued..)

- Some debuggers are built on top of lower layers, JDWP (for example if the front-end is not written in the Java language) or JVM TI (for specialized debuggers which need low-level functionality).

Java Platform Debugger Architecture (JPDA)

Sang Shin
www.javapassion.com