



REFCARDSTREND REPORTSEVENTS

Log In / Join



Culture and Methodologies

Data Engineering

Software Design and Architecture

Coding

Testing, Deployment, and Maintenance

Data Engineering: The industry has come a long way from organizing unstructured data to adopting today's modern data pipelines. See how.

Download the NEW Trend Report

What metrics does your organization use to measure success? MTTR? Frequency of deploys? Other? Tell us!

Join Our Research

Is your data pipeline ready for the enterprise? Join us to discuss streamlining application workflows, data orchestration, and more!

Save Your Seat


Managing API integrations: Assess your use case and needs — plus learn *patterns* for the design, build, and maintenance of your integrations.


Read the Refcard


Partner Resources

Welcome

This site asks for consent to use your data

Personalised advertising and content, advertising and content measurement, audience research and services development

Store and/or access information on a device

Learn more

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.

Manage options


Consent

TRENDING


- RAG Systems: A Brand New Architecture Tool
- Lifecycle Microservices With GenAI Tools
- Leveraging Event-Driven Data Mesh Architecture With AWS for Modern Data Changes
- Author's Article Submission Guidelines

Welcome

This site asks for consent to use your data



Personalised advertising and content, advertising and content measurement, audience research and services development



Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.

Applications With JDWP

Take a look at this simple breakdown of how to use the JDWP to debug Java applications.

By  Mahmoud Anouti  MVB · Jul. 16, 19 · Tutorial

 Like (1)

 Comment (0)

 Save

 Tweet

 Share

 29.3K Views

Join the DZone community and get the full member experience.

JOIN FOR FREE

Most Java developers have had to debug their applications, usually to find and fix an issue there. In many cases, the application (the “*debuggee*”) is launched from the IDE, and the debugger is also integrated into the IDE. In such scenarios, the JVM is launched from a separate host. In such scenarios, debugging options suitable for debugging are available to it. This is where [JDWP \(Java Debug Wire Protocol\)](#) comes in.

What is JDWP?


In order to debug remotely executed applications, the application is launched locally or on another host, and the debugger is launched for communication between the two. The debugger sends the format of the commands sent to the JVM, and the JVM sends back the and replies by the JVM. The execution is controlled by the debugger specified and is up to the implementation of the JVM. What JDWP specifies is the format of the commands sent and those containing replies. Therefore it is conceptually very simple.

JDWP is only one part of the debugging infrastructure in the Java platform. The endpoints (debugger and debuggee) communicating over JDWP implement other specifications to provide the actual debugging functionality. The JVM implements the JVM Tool Interface (JVMTI) to provide debugging functionality for it, for example, to control executions using breakpoints or inspecting the current object. JVMTI is the low-level layer implemented natively in the JVM. The debugger implements another interface called the Java Debug Interface (JDI) that provides a high-level way to carry debugging requests from the debugger process. JDI is a pure Java interface. Together, JVMTI, JDWP, and JDI form the main layers of the Java Platform Debugger Architecture. Links to official references about all these specifications are provided at the end.


In the Oracle Java implementation, there are two transport mechanisms provided: the socket transport, and the shared memory transport for Windows only. The socket transport (`dt_socket`) relies on TCP sockets bound to listen on a port for connections, and using that connection to transfer the debug session packets. Shared memory transport (`dt_shmem`) uses shared memory to send

Welcome

This site asks for consent to use your data




Personalised advertising and content, advertising and content measurement, audience research and services development




Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.



REFCARDSTREND REPORTSEVENTS

Log In / Join

Culture and MethodologiesData EngineeringSoftware Design and ArchitectureCodingTesting, Deployment, and Maintenance

Community researchSitemap

CONTRIBUTE ON DZONEArticle Submission GuidelinesBecome a ContributorCore ProgramVisit the Writers' Zone

LEGALTerms of ServicePrivacy Policy

CONTACT US3343 Perimeter Hill DriveSuite 100Nashville, TN 37211support@dzone.com

certain address, namely an assigned IP address and port number.

2. Attach the other part to the listening server on that address.

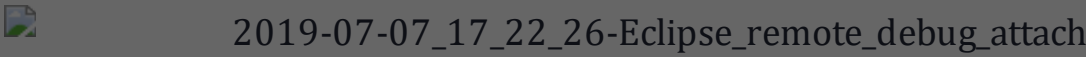
For example, to launch the JVM with debug options to listen on an address, we use the following option with the Java executable:

```
1 java -agentlib:jdwp=transport=dt_socket,server=y,suspend=y,address=8000 -jar MainClass.jar
2
```

The `-agentlib:jdwp` with the `transport=dt_socket` tells the JVM to load the JDWP agent. Here's what each suboption does:

- `transport=dt_socket` tells the JVM to use the socket transport.
- `server=y` means that the JVM will act as a server and listen for incoming connections.
- `suspend=y` means the JVM will immediately execute the main class after establishing a connection.
- `address=8000` specifies the address to listen on. In this case, the JVM will listen on port 8000 from the local host (localhost).

The second step is to attach the debugger at that address. All popular IDEs provide a way to easily do this. In Eclipse, for example, it can be configured by going to Run -> Debug Configuration and creating a Remote Java Application configuration:




Notice that the host and port must match the address of the JDWP agent on JVM side.

[JDK 9+] Binding the Listening Socket to All Addresses


In the previous example, the `address` was set to 8000 (port number) without any hostname or IP address. Before JDK 9, this would mean the JVM would listen on all available IP addresses, making the socket accessible by debuggers on remote machines. Starting with JDK 9, this was changed to only allow local connections for better security. In other words, `-agentlib:jdwp=transport=dt_socket,server=y,address=8000` is now equivalent to `-agentlib:jdwp=transport=dt_socket,server=y,address=8000,localhost=true`.

Welcome

This site asks for consent to use your data



Personalised advertising and content, advertising and content measurement, audience research and services development



Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.

Page 4 of 6

port with the host name, IP address, or an asterisk (`*`) to bind to all available IP addresses:

```
-agentlib:jdwp=transport=dt_socket,server=y,address=host1:8000
```

or

```
-agentlib:jdwp=transport=dt_socket,server=y,address=*:8000
```

More Examples

Adding a Timeout

We can add a timeout for the JDWP agent listening for the debugger. To make the JVM exit after 10 seconds without any debugger attaching:

```
-agentlib:jdwp=transport=dt_socket,server=y,address=*:8000,timeout=10000
```

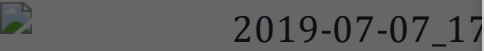
Listening at A Dynamic Address

If `server=y` (i.e. JVM is listening) and `server=n` (i.e. JVM is not listening) option, which will make it use the address specified, this allows only the address displayed at stdout of the JVM.

```
Listening for dt_socket at address 127.0.0.1:8000
```

The Other Way Around

We can set `server=n` on the JVM. The `server` option as it defaults to `y` (yes) address. We would first run the



Let’s say the debugger was started on `host2` . We would then run the JVM with the option:

```
-agentlib:jdwp=transport=dt_socket,address=host2:8000
```

Delaying JDWP Connection Establishment until A Specific Exception Is Thrown


A useful option to the JDWP agent is to start the JVM as normal and wait until a specific exception is thrown. For example, say you want to debug a failing application with a `MyCustomException` but don’t want to initiate the debugger connection until it is thrown. This can be done with the `onthrow` option:

```
-agentlib:jdwp=transport=dt_socket,server=y,address=*:8000,onthrow=com.example.MyCustomException,launch=notify_script
```


This would start the application normally without listening on the address. When the exception is thrown, the agent will listen on port 8000 and a debugger can be attached to it. The `launch` option is a mandatory option along

Welcome

This site asks for consent to use your data



Personalised advertising and content, advertising and content measurement, audience research and services development



Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.

listening VM upon the exception being thrown.

References

- Java Platform Debugger Architecture
- JDWP spec
- JPDA Connection and Invocation Details
- [JDK 9 Release Notes] JDWP socket connector accept only local connections by default


Published at DZone with permission of [DZone.com](#). See the [full article here](#).

- Java (Programming Language)
- Shared Memory
- Connection (Data Transfer)


Published at DZone with permission of [DZone.com](#). See the [full article here](#).
Opinions expressed by DZone contributors are their own.

Welcome

This site asks for consent to use your data



Personalised advertising and content, advertising and content measurement, audience research and services development



Store and/or access information on a device

Your personal data will be processed and information from your device (cookies, unique identifiers, and other device data) may be stored by, accessed by and shared with [134 TCF vendor\(s\)](#) and [63 ad partner\(s\)](#), or used specifically by this site or app.

Some vendors may process your personal data on the basis of legitimate interest, which you can object to by managing your options below. Look for a link at the bottom of this page to manage or withdraw consent in privacy and cookie settings.