



# Remote Debugging, Monitoring, Profiling, Managing



# Topics

- Remote debugging
- Remote profiling
- Remote monitoring & management

# Remote Debugging

# Needs for Remote Debugging

- Remote debugging is useful when you are developing an application that runs on a web server or on a different environment than the computer on which you are developing the application.
- Debugging remote objects without remote debugging capability requires inserting a myriad of `System.out.println( )` statements or other logging code.
  - > Using logging code for debugging is slow and inefficient, whereas using NetBeans for remote debugging is much more powerful.

# Remote Debugging via NetBeans

- The NetBeans debugger can attach to a Java process already running in a separate JVM, either on the same computer or on a remote machine.
- This gives a developer the same power for debugging remote applications and components running in Java-based server containers.

# Remote Debugging via NetBeans

- Attaching to a remote JVM makes it possible to use breakpoints, conditionals, watches, and other debugging features with Java applications, applets, servlets, Enterprise JavaBeans, and RMI or CORBA server objects.

# How to build and run the target application

- Sun's VM implementations require command line options to load the JDWP agent for debugging for the target application
- On the computer where the application (debuggee) is located, start the application in debugging mode.
  - > JDK5 & JDK6: (Using new JVM TI interface)
    - -agentlib:jdwp=<sub-options>
  - > Prior version of JDK5, JDK5 & JDK6
    - -Xdebug
    - -Xrunjdwp:<sub-options>

## <sub-options>

- **transport**
  - > is a method of communication between a debugger and the virtual machine that is being debugged.
- **address**
  - > when establishing a connection, transport addresses is used to identify the end-point of the connection.
- **server**
  - > if the server property is 'y', the application will listen for a debugger to attach; otherwise, it will attach to the debugger at the specified address.
- **suspend**
  - > if suspend is 'n', the application will start immediately and will not wait for a debugger to attach to it. If 'y', the application will be suspended until you attach to it.

# How to build and run the target application: Example1

- `-agentlib`  
`:jdwp=transport=dt_socket,server=y,address=8000`
  - > Listen for a socket connection on port 8000.
  - > Suspend this VM until attached (connected) by debugger application (`suspend=y` by default).
  - > Once the debugger application connects, it sends a JDWP command to resume the VM.

# How to build and run the target application: Example2

- -agentlib  
:jdwp=transport=dt\_socket,server=y,address=localhost  
:8000,timeout=5000
  - > Listen for a socket connection on port 8000 on the loopback address only.
  - > Terminate if the debugger does not attach within 5 seconds.
  - > Suspend this VM until attached (connected) by debugger application (suspend=y by default).
  - > Once the debugger application connects, it sends a JDWP command to resume the VM.

# How to build and run the target application: Example3

- -agentlib  
:jdwp=transport=dt\_shmem,server=y,suspend=n
  - > Choose an available shared memory transport address and print it to stdout.
  - > Listen for a shared memory connection at that address.
  - > Allow the VM to begin executing before the debugger application attaches.

# How to build and run the target application: Example4

- -agentlib  
:jdwp=transport=dt\_socket,address=myhost:8000
  - > Attach to a running debugger application via socket on host myhost at port 8000.
  - > Suspend this VM until attached (connected) by debugger application (suspend=y by default).

# How to build and run the target application: Example5

- -agentlib
  - :jdwp=transport=dt\_socket,server=y,address=8000,onthrow=java.io.IOException,launch=/usr/local/bin/debugstub
  - > Wait for an instance of java.io.IOException to be thrown in this VM.
  - > Suspend the VM (suspend=y by default).
  - > Listen for a socket connection on port 8000.
  - > Execute the following: "/usr/local/bin/debugstub dt\_socket myhost:8000".
  - > This program can launch a debugger process in a separate window which will attach to this VM and begin debugging it.

# How to build and run the target application: Example6

- -agentlib  
:jdwp=transport=dt\_shmem,server=y,onuncaught=y,launch=d:\bin\debugstub.exe
  - > Wait for an uncaught exception to be thrown in this VM. Suspend the VM.
  - > Select a shared memory transport address and listen for a connection at that address.
  - > Execute the following: "d:\bin\debugstub.exe dt\_shmem <address>", where <address> is the selected shared memory address.
  - > This program can launch a debugger process in a separate window which will attach to this VM and begin debugging it.

# Remote Profiling

# Remote Profiling with NetBeans

- You can profile an application that is running on a remote system such as a web server by attaching the profiler to the application.
- When you use this mode, the remote application starts after the profiler is attached.
  - > This mode enables you to obtain profiling data on the startup of the target JVM.

# NetBeans Profiler Remote Pack

- To attach profiler to an application on a remote system, you need to download and install the Profiler Remote Pack on the remote system.
- The remote system needs to be started on the Profiler Remote Pack and configured to support remote profiling.
- You can download the Profiler Remote Pack from the Profiler web site:
  - > <http://profiler.netbeans.org>

# How to use Attach Wizard

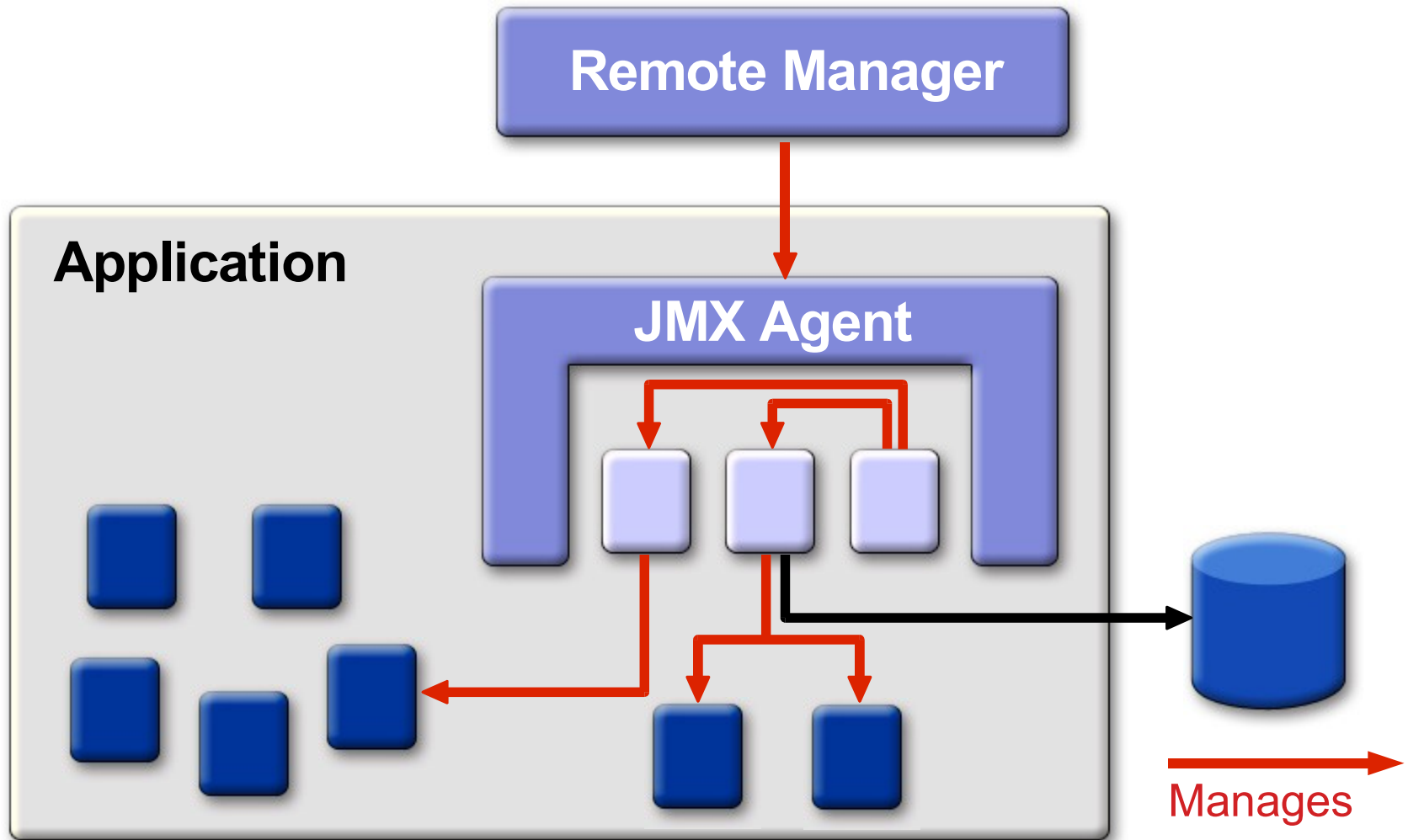
- You use the Attach Wizard to specify the attachment settings for your project.
- In the Attach Wizard you specify the type of application and the remote location.
- Based on the details that you provide, the Attach Wizard provides you with a set of instructions on how to configure the remote system to support profiling.

# How to use Attach Wizard

- After configuring the remote system according to the instructions, you can attach the profiler to the remote location.
  - > You only need to configure the attach mode once.
  - > The attachment settings are associated with that project.
  - > You can go through the Attach Wizard at any time to change any of the attachment settings.

# Remote Monitoring & Management

# JMX Architecture



# Tools Remote Monitoring and Management

- JConsole
- VisualVM (with JConsole plug-in)
- jstat