

Debugging WebSphere applications with WSAD

By:

David Ramirez

iSOA Group, a *Commerce Solutions Inc.* subsidiary

Solutions Consultant

Contact: cmccall@isoagroup.com

Copyright © 2002, 2007 iSOA Group



The Importance of Debugging

- Enables developers to produce robust software
- A necessity to software development
- If done in an organized fashion, can lead to better productivity
- The core of proper Unit testing

Proper Unit Testing

- You must be able to identify *what* you are going to code
- You must be able to identify *why* you are going to code it
- You must be able to identify *how* you are going to code it
- You must be able to identify what coded *works the way you designed it.*

JUnit Framework

- Open source unit testing framework
- Written by Kent Beck and Erich Gamma (of Gang of Four)
- Based on 3 design patterns
 - Command: individual tests are executed with a single method with no parameters and a void return type
 - Composite: TestCases and TestSuites
 - TestCase – a single Java executable containing a set of tests
 - TestSuite – a set of TestCases and/or TestSuites
 - Decorator: For test extensions (ie. Timed Tests)
- <http://www.junit.org>

Why WSAD?

- IBM created an open source platform called Eclipse which they built a set of tools on for building enterprise Java applications(WSAD)
- Since the platform is open source, third party vendors can create tools you can utilize
- WSAD is very “pluggable”(JVM, App server, ...)
- WSAD gives the developer a one-stop shop for development
- With WSAD we can use one debugger for both local and remote debugging(unlike VJava did)

WSAD Debugger

- The debugger allows you to control the execution of your program
 - Setting breakpoints
 - Suspending launches
 - Stepping through your code, and examining the contents of variables

Breakpoints

- Breakpoints are temporary markers you place in your program to tell the debugger to stop your program at a given point
- You can set break points in Java classes, Java Servlets, JSP's, and even EJBs.
- You can set a HitCount on a certain break point which will tell the debugger to halt at the break point when the HitCount value is reached

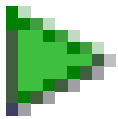
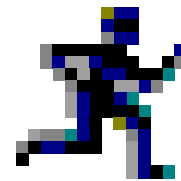
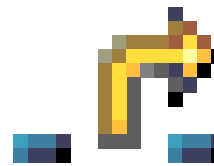
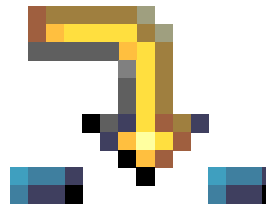
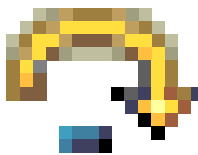
Suspending launches

- When the WSAD is running a program and encounters a breakpoint, it suspends execution. The corresponding thread is suspended (i.e., temporarily stops running) so that you can see the stack for the thread.

Stepping through your code

- Once the thread has been stopped on a breakpoint, you can then **step over** statements, **step into** other methods or classes, **step to return** of a method, continue running until the next breakpoint is reached, or continue running until you reach the end of the program.

Icons



Evaluating variables

- Once the thread has been stopped on a breakpoint, you can perform the following actions on variables of your program:
 - **Inspect:** This will show you the value of a specified variable
 - **Change Value:** This allows you to change the value of a specified variable “on the fly”
 - **Show Type Names:** This displays the type of the variable
 - **Show Qualified Names:** This allows you to see the fully qualified name of the variable type

Caught/Uncaught Exception Breakpoints

- The debugger also allows you to set breakpoints on specific Exceptions
- You can specify to halt on the Exception for both Caught or Uncaught

Some notes about JSP debugging

- **Note:** If you are running the JSP file on a remote WebSphere Application Server, the JSP cache directory must be cleared in order to force the class file to be in the correct JSP source debug mode. If the cache is not cleared, some compiled unchanged JSP files may still be compiled in the previous JSP source debug mode.
- **Note:** JSP source debugging is not supported on Tomcat.



It's Demo time....

Copyright © 2002, 2007 iSOA Group



Steps for Remote Debugging

- In Console set JVM setting under Application Server
 - Click JVM setting
 - Enable Debug Mode
 - Add to debug string
 - Xrunjwp:transport=dt_socket,server=y,suspend=n,address=0317
 - Uncheck Enable IBM Distributed Debugger
 - Disable JIT
 - Click OK, Apply changes, Start Application server
- In WebSphere Studio Application Developer
 - Insert breakpoints
 - Click Bug dropdown, select Remote Java Application

Remote debugging (cont.)

- Enter hostname and Port address specified earlier
- Click Finish
- Breakpoints are in green not blue (not available)

Shameless plug

- Watch for the WebSphere Application Server 4.0 Bible in your bookstores in July 2002
- Authors: Bryon Kataoka, Dave Ramirez and Alan Sit
- Publisher: Hungry Minds, Inc; ISBN: 0764548964; 1st edition