



# Boris Bokowski

Travelling across Canada

Monday, January 15, 2007

## Circular Debugging

Did you know you can attach the Eclipse debugger to itself, i.e. the currently running Eclipse instance?

What the ...? Attach to itself? Yes, to itself. Sort of like the MetaClass circularity in Smalltalk, but different. But I digress...

Imagine a single Java VM which runs Eclipse. To be able to connect a debugger to it, it has to be started in debug mode. With a Sun VM, the following magic command line does the trick:

```
-vmargs -Xdebug -Xnoagent -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n
```

Note how we use suspend=n to let Eclipse start without waiting for a debugger to connect. If you start Eclipse by running the executable directly, you can put the options in your eclipse.ini file instead like this:

```
-consoleLog
-vmargs
-Xms40m
-Xmx256m
-Xdebug
-Xnoagent
-Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n
```

I also added the -consoleLog option in order to see any strings printed to the console by Eclipse.

Alright, so now we have Eclipse running as usual, except it is running in debug mode. This is not very exciting yet, but wait for this - here is how you attach it to itself, in three easy steps:

1. *Pick a project containing the Eclipse code you want to debug.*

We will be debugging what is called a "Remote Java Application", except that the application is not very remote ;-). The debugger needs a project when creating the launch configuration because otherwise it would not be able to look up classes and their source code. You can pick your favourite Eclipse plug-in from your workspace, but make sure that the code in the project matches the exact version of your running Eclipse. If you want to play it safe, I recommend creating a project as follows: Open the Plug-ins view, select all, and pick "Add to Java Search" from the context menu. This will create a project called "External Plug-in Libraries" that matches your currently running Eclipse instance.

2. *Create conditional breakpoints (optional):*

Since there is only one VM and thus only one UI thread, you will not be able to single-step through code that runs on the UI thread. Don't add *unconditional* breakpoints to any UI code because this will lock up Eclipse. You can, however, add conditional breakpoints that don't actually suspend the current thread. Create a breakpoint, then right-click on it and pick "Breakpoint Properties", click on "Enable Condition" and put something like this in the space below (by returning false after printing to the console, the thread will not suspend).

```
System.out.println("Breakpoint hit");
Thread.dumpStack();
return false;
```

3. *Create a launch configuration for a "Remote Java Application":*

Select the project from step one. Open the debug launch configuration dialog using Run->Debug..., select "Remote Java Application", and click the button for creating a new launch configuration. The default values should be good, with your project from step one, and the port number matching the one from the command line or eclipse.ini.

P.S. Why am I putting this on my blog? I couldn't easily find the magic command line options for starting a VM in debug mode. Luckily, [zx](#) on IRC knows these kinds of things. Rather than writing it down on a piece of paper that I would lose instantly, I wanted to record it as a blog entry and then got carried away... but if I'm lucky, I will soon be able to google for "circular debugging" to find this page.

Posted by Boris Bokowski at [Monday, January 15, 2007](#)

### 2 comments:

[Eugene Kuleshov](#) said...

### About Me



**BORIS BOKOWSKI**

OTTAWA, CANADA

Eclipse Platform UI  
lead

[View my complete](#)

[profile](#)

There was an error in this gadget

### Blog Archive

- 2011 (4)
- 2010 (8)
- 2009 (11)
- 2008 (9)
- ▼ 2007 (9)
  - November 2007 (2)
  - September 2007 (2)
  - July 2007 (1)
  - June 2007 (1)
  - March 2007 (1)
  - ▼ January 2007 (2)
    - [Circular Debugging](#)
    - [Levels of Abstraction](#)
- 2006 (4)
- 2005 (2)

### Subscribe To

Posts

Comments

7/21/2017

Boris Bokowski: Circular Debugging



I don't remember parameters either, but searching "jpda debug" on Google is probably faster than hunting Chris on irc.

Anyways, cool stuff! I was planning to try this, but never got into actually setting this up. :-)

[January 15, 2007 7:28 PM](#)



[Boris Bokowski](#) said...

You can use hot code replace, too!

[January 15, 2007 7:46 PM](#)

[Post a Comment](#)

#### Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)