**Arbi Sookazian**

**April 16, 2010**

**Version 1.0.0**

**Using JRebel 3 with Seam 2.x Tutorial**

**What is JRebel?**

JRebel is a JVM plugin (-javaagent) in the form of a commercial software developed by ZeroTurnaround (<www.zeroturnaround.com>) that enables you to see changes to your code the next time your modified class is accessed, without the need to redeploy an application or perform a container restart. If you're tired of watching the logs roll by, and want to see your changes so that you can keep going - JRebel is your new best friend. You will save much time during your daily dev/build/deploy/test cycles when using JRebel with your Seam/EE5 apps. One significant and exciting new feature in JRebel 3 is the ability to hot redeploy interface changes to EJB3 session beans as well as hot redeploy additions/subtractions of Seam and EJB annotations in your JavaBeans and EJB components.

<http://www.zeroturnaround.com/reference-manual/intro.html#intro>

**How does JRebel work?**

<http://www.zeroturnaround.com/jrebel/faq/#How_does_jrebel_work>

<http://www.zeroturnaround.com/reference-manual/intro.html#intro>

**What does JRebel provide that HotSwap and Hot redeploy don’t?**

Read this article on the feature comparison matrix: <http://www.zeroturnaround.com/jrebel/comparison>

**What about the special Seam classloader in Seam 2.x?**

The Seam incremental hot deploy feature does not support incremental hot deploy of EJB 3 session beans or entity classes. JRebel 3 does (there is experimental support for entities for OpenJPA currently). JRebel also supports hot redeploy of @In and @EJB annotation modifications (add/delete).

**What is the latest version of JRebel?**

Download JRebel here:[**http://www.zeroturnaround.com/jrebel/current/**](http://www.zeroturnaround.com/jrebel/current/)

As of this writing, the latest stable version is JRebel 3.0 stable release (**~ 10mb** | **16th April 2010**)

This tutorial is written using 3.0.1 due to a bug in the 3.0 stable release which will prevent the successful hot redeployment of interface changes in a session bean.

**What are the new features in JRebel 3?**

<http://www.zeroturnaround.com/news/jrebel-3-0/>

**What does this tutorial cover?**

This tutorial uses the Seam distro’s hotel booking application to demonstrate

1. how to incrementally hot deploy (aka hot redeploy) interface changes to a stateful session bean by adding a public method and accessing that method without a full rebuild/deploy cycle to JBoss AS
2. how to hot redeploy adding a @In Seam injection for a newly coded Seam component (JavaBean) in the same stateful session bean

**Tutorial Pre-requisite installations**

Please install the following prior to installing JRebel 3:

1. Sun JDK 5 <http://java.sun.com/javase/downloads/index_jdk5.jsp> - as of this writing it’s JDK 5.0 Update 22)
2. JBoss AS 4.2.x <http://www.jboss.org/jbossas/downloads/>
3. JBoss Tools 3.x <http://relation.to/Bloggers/JBossTools31Final> (we will be using the JBoss server view)
4. Eclipse Galileo 3.5 <http://www.eclipse.org/downloads/> - ([Eclipse IDE for Java EE Developers](http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/galileo/SR2/eclipse-jee-galileo-SR2-win32.zip) is a good option)
5. Seam 2.1.x.GA <http://www.seamframework.org/Download> (Seam 2.0 and 2.1 examples target JBoss Application Server 4.2 or 5.0.)
6. Apache Ant 1.7.x (simply extract to any directory and setup ANT\_HOME environment variable like this in Windows: C:\java\apache-ant-1.7.1)

if you’re going to build from command line rather than Eclipse

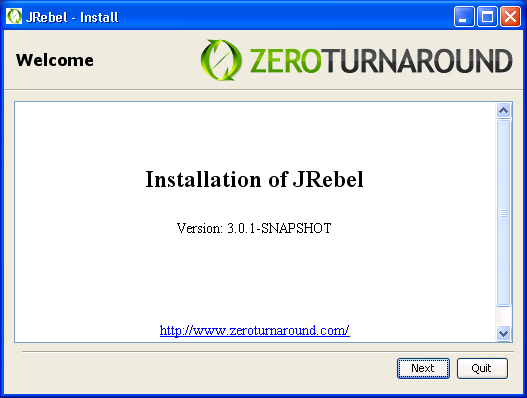
You are welcome to try JBoss AS 5.1 with Seam 2.2.x but I have not tried that combo yet with JRebel 3.x.

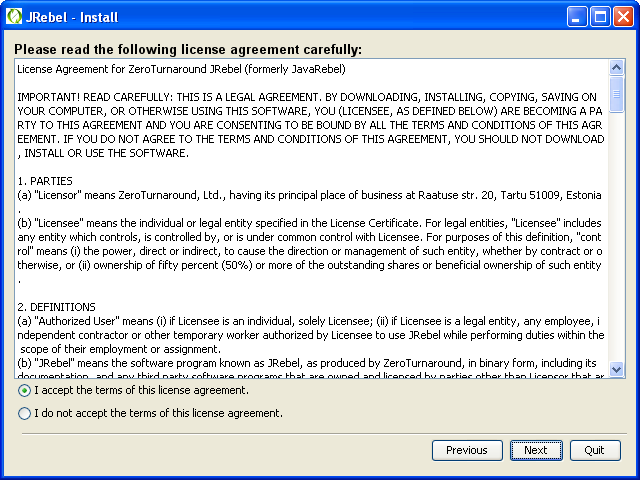
**JRebel Installation (Windows installer):**

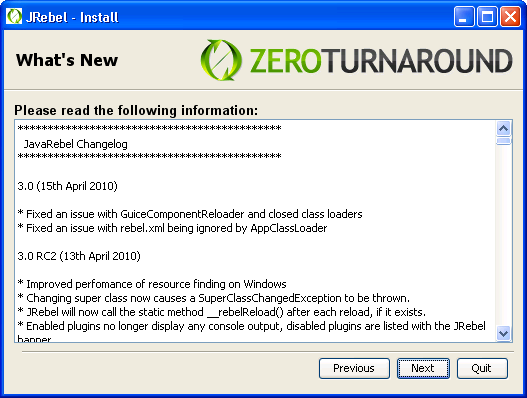
Download JRebel 3.0.1 (or later) from this link: <http://www.zeroturnaround.com/jrebel/next-releases/> (select the Windows installer)

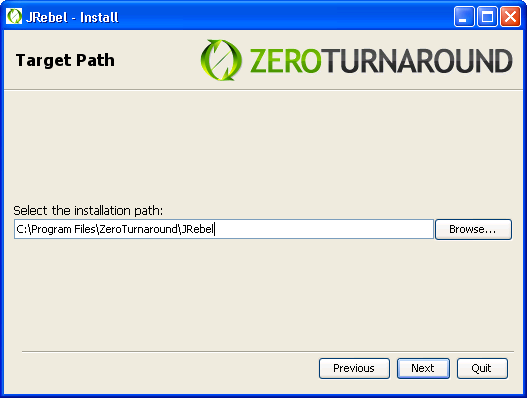
****

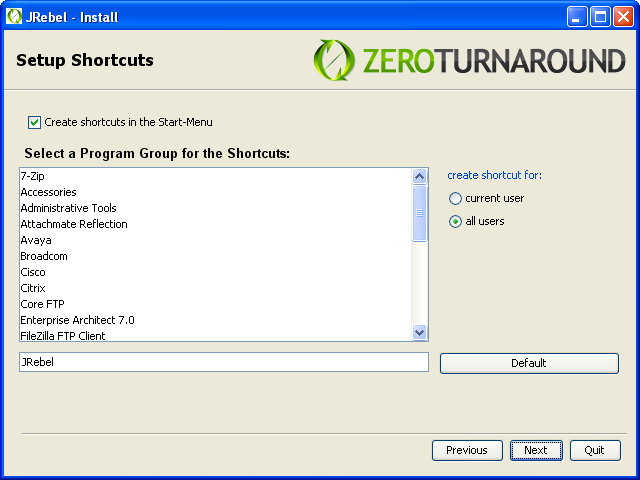
Double-click the jrebel-nightly-setup.exe above; keep clicking next (accepting defaults) and then Finish. The JRebel configuration wizard is optional for this tutorial but I have included the screenshots just in case.

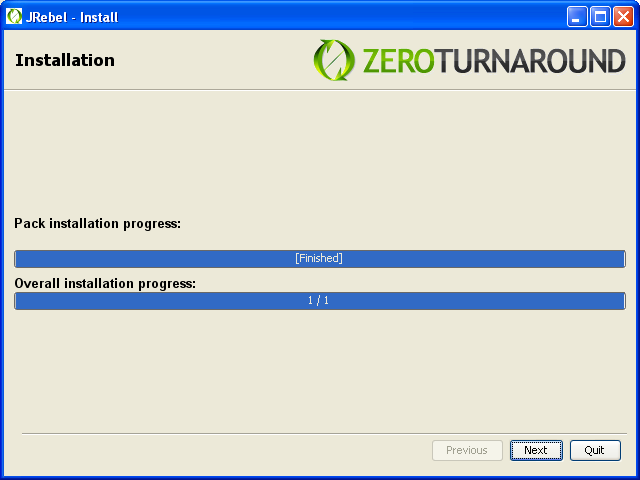
****

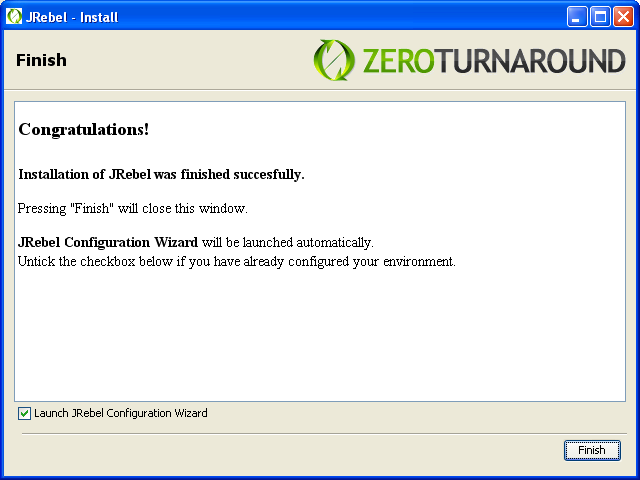
****

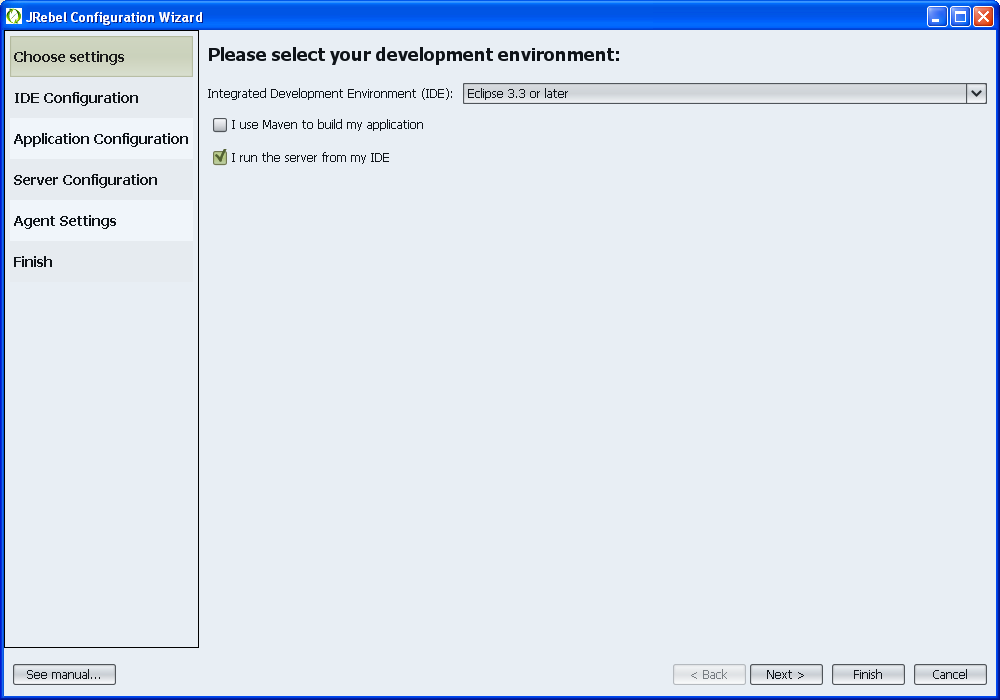
****

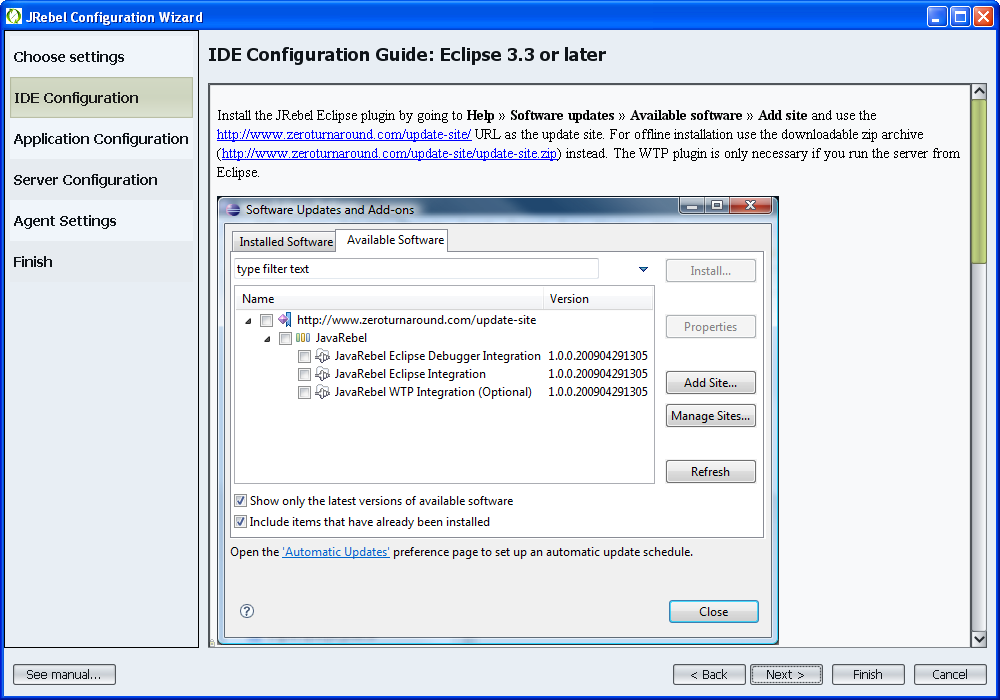
****

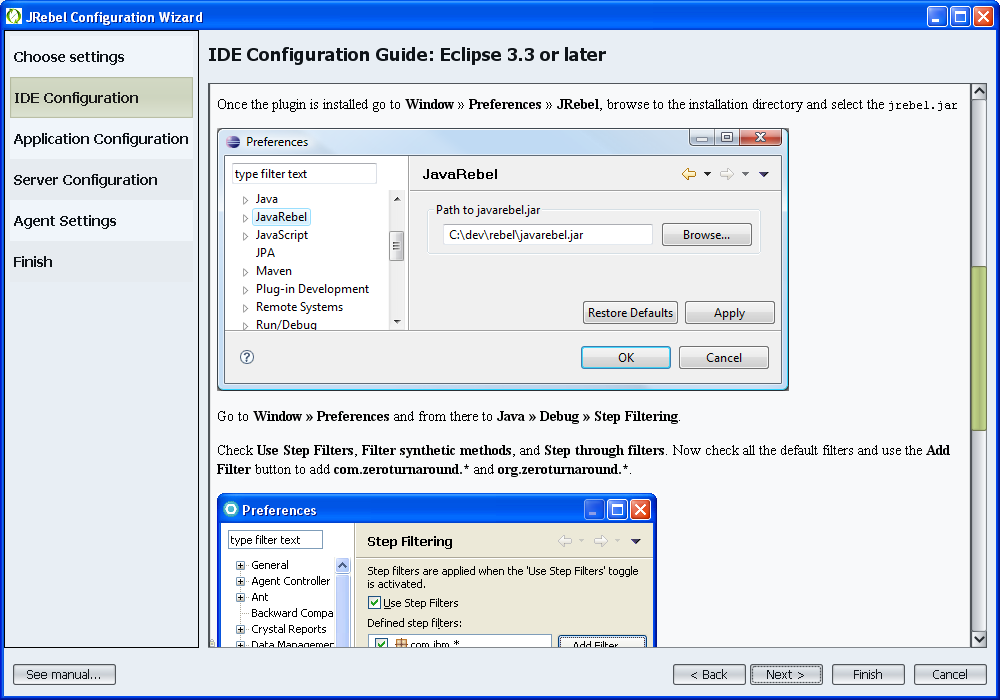
****

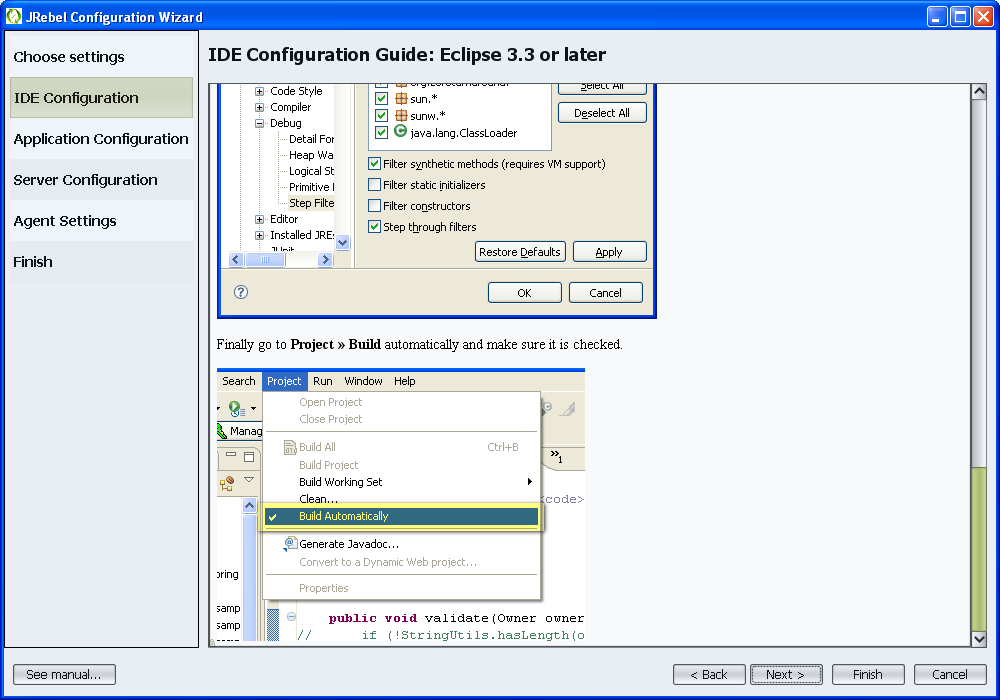
****

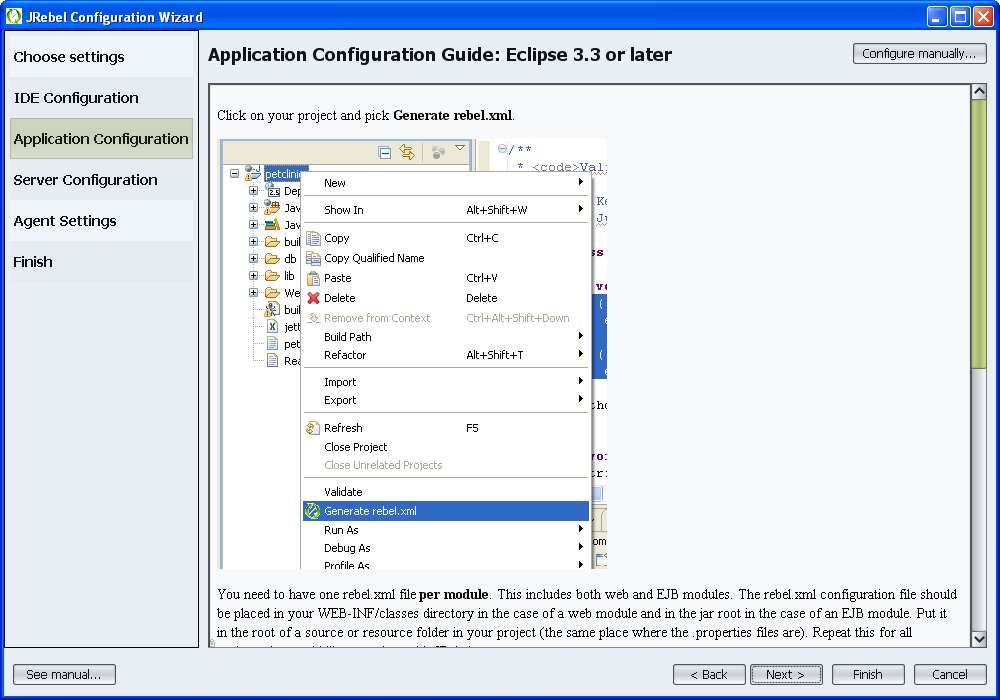
****

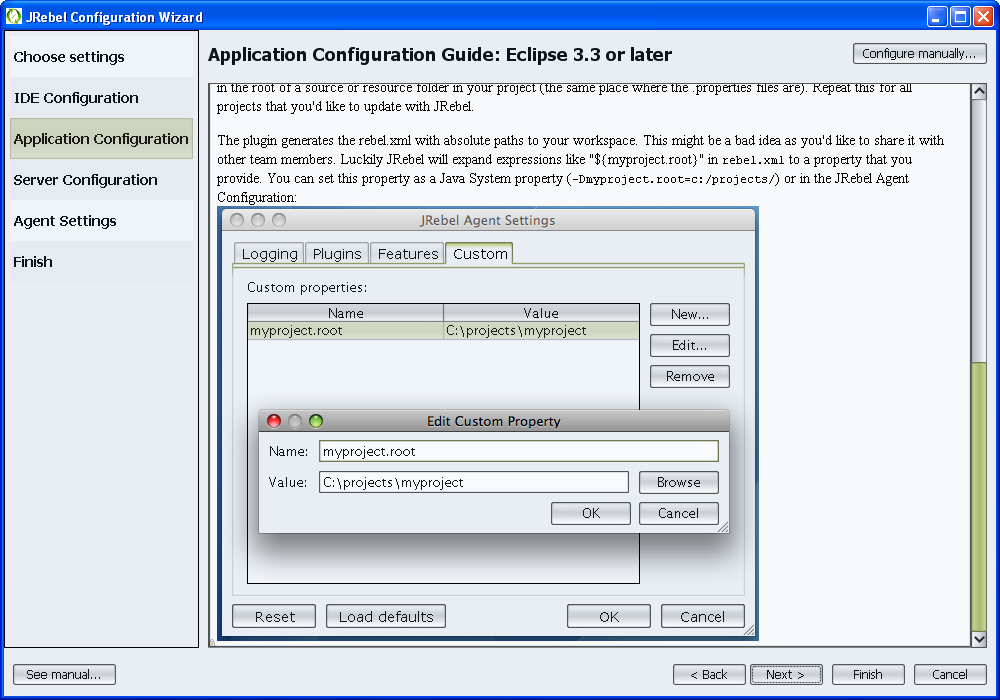
****

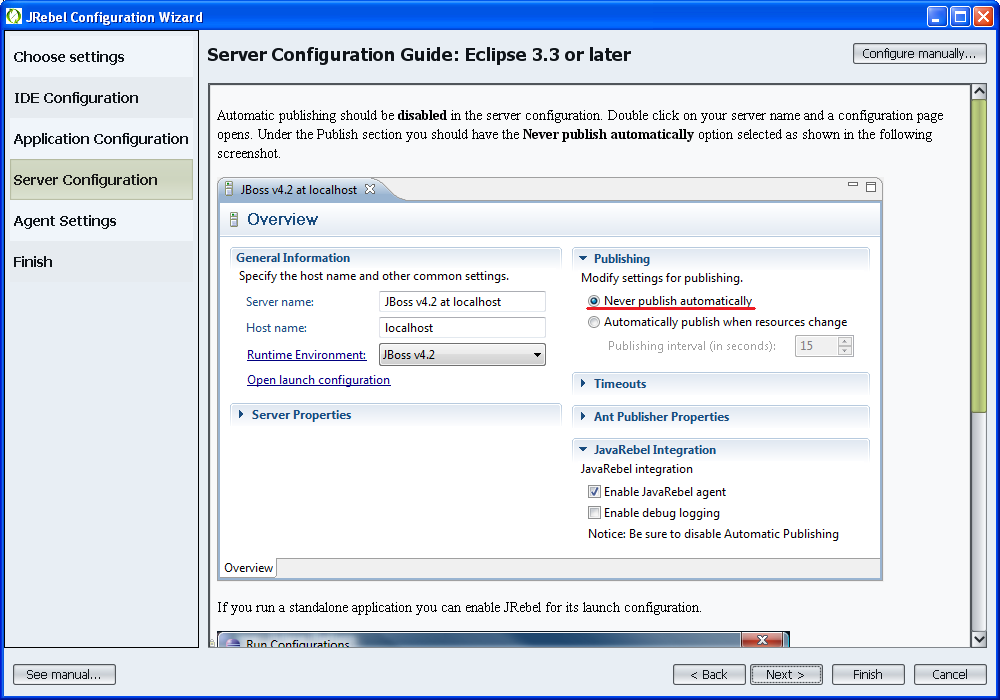
****

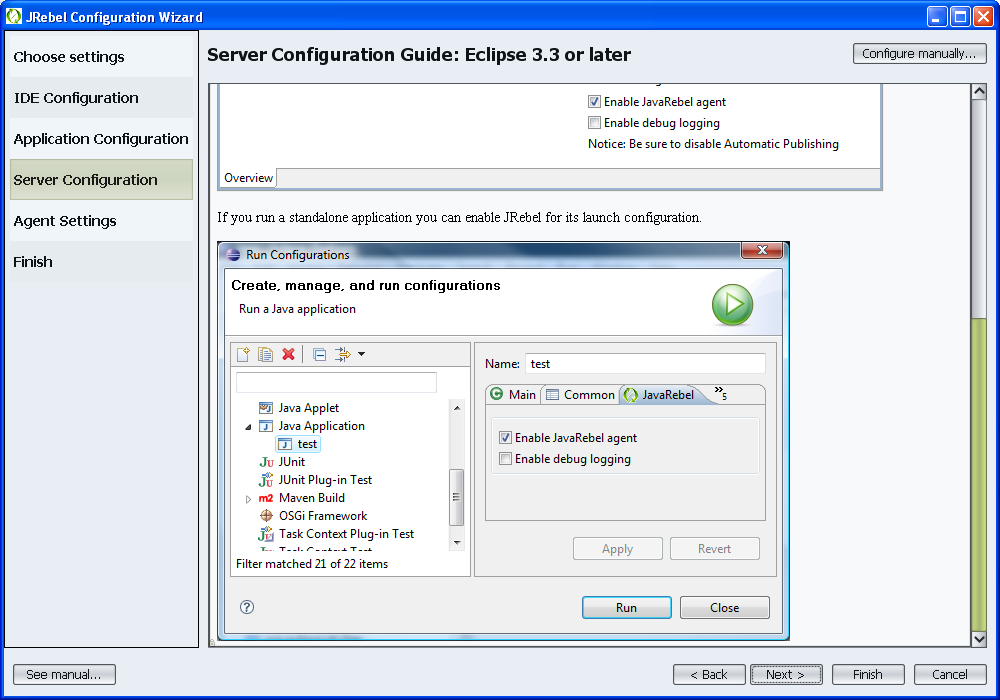
****

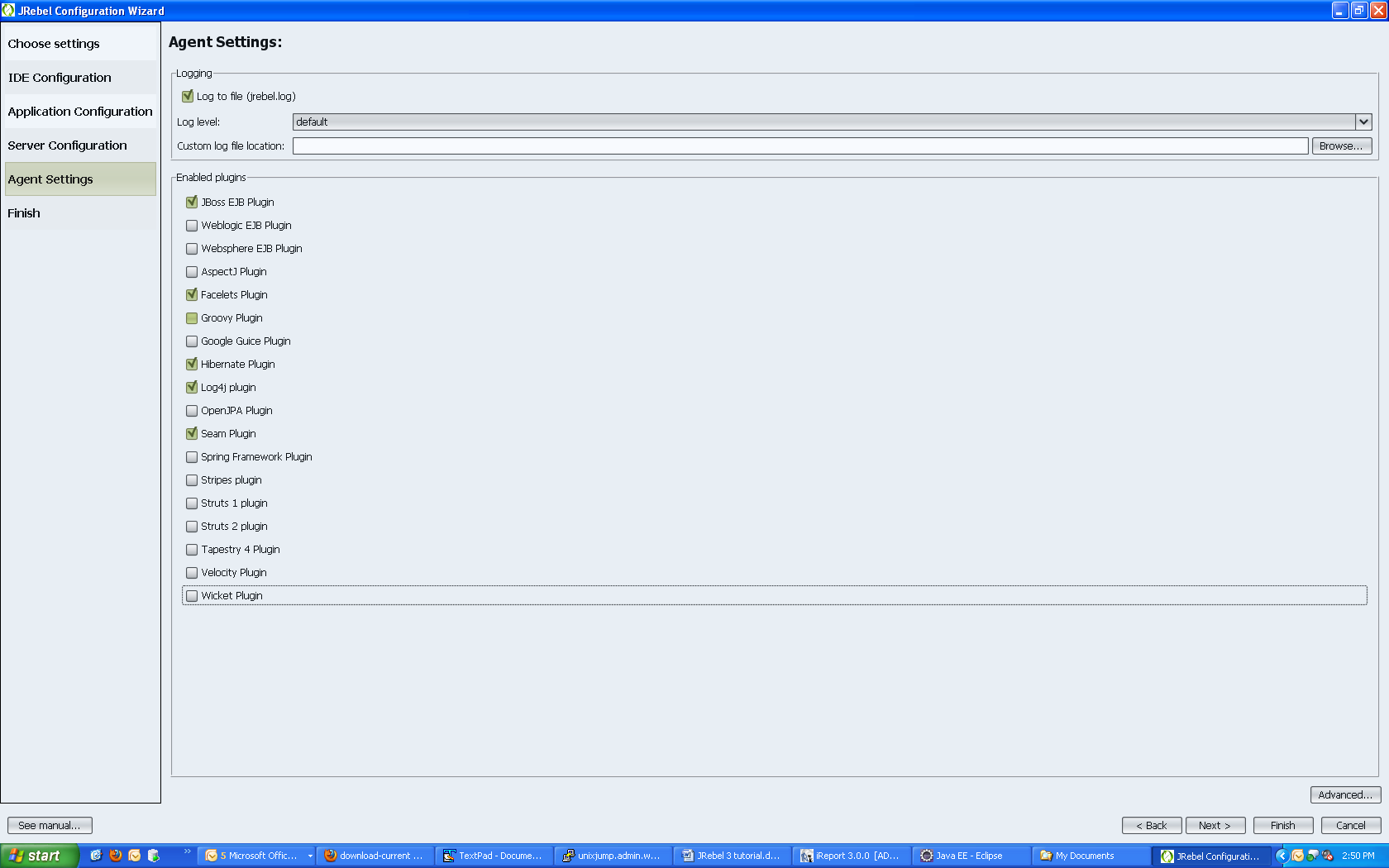
****

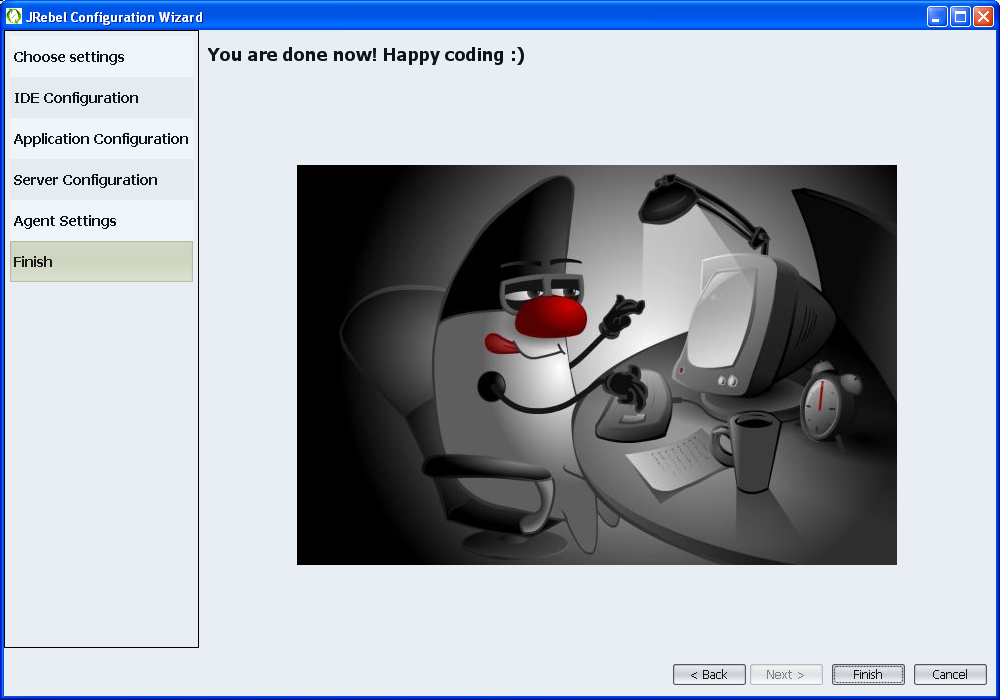
****

****

****

****

****

****

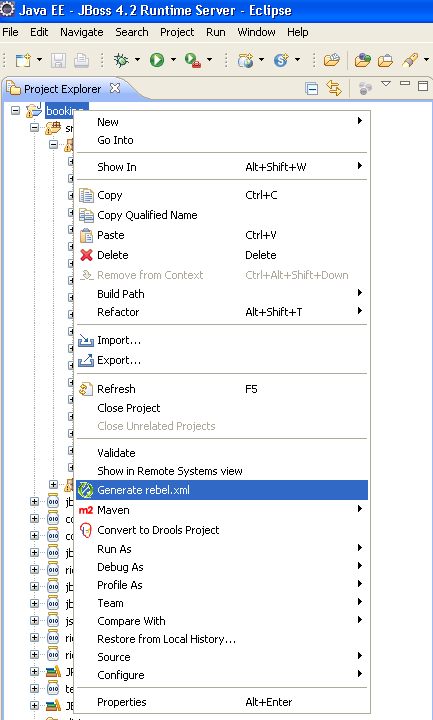
Click Finish

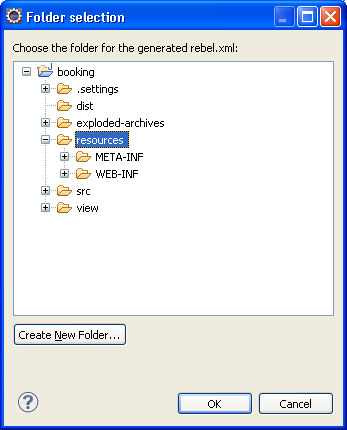
Launch Eclipse

Create a new workspace

Import the Seam hotel booking application from the Seam 2.2.x distro located here: %SEAM\_HOME%\examples\booking

Generate a rebel.xml via the plugin

****

****

Click ok to save rebel.xml in booking/resources directory**.** NOTE: this xml file will need to be located in %SEAM\_HOME%\examples\booking\exploded-archives\jboss-seam-booking.jar directory. We will modify the build.xml file later to ensure it is added to the JAR artifact. The rebel.xml tells JRebel what directory or directories to monitor for changes and subsequent reloads. To reload the EJB components put a rebel.xml into the root of the EJB jar(s). To reload the WAR classes put rebel.xml into WEB-INF/classes. So in the end you'll need 2 rebel.xml files for most projects but for this tutorial we only need to generate one rebel.xml for the JAR because we are only working with one SFSB.

JRebel plugin creates this rebel.xml file for you:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<application xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://www.zeroturnaround.com"* xsi:schemaLocation=*"http://www.zeroturnaround.com/alderaan/rebel-2\_0.xsd"*>

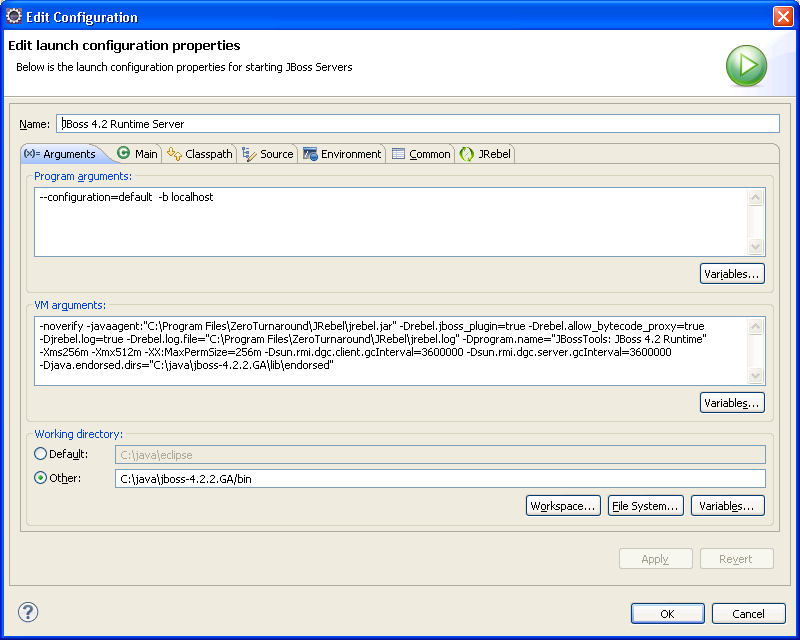
<classpath>

<dir name=*"C:/java/jboss-seam-2.1.1.GA/examples/booking/exploded-archives/jboss-seam-booking.jar"*>

</dir>

</classpath>

</application>

****

**Hot redeploy scenarios**

Add a HtmlCommandButton to main.xthml in the form with id=”searchCriteria”:

<h:commandButton action="#{hotelSearch.jrebel}" value="JRebel!"/>

Add a method to HotelSearching local interface:

//public void jrebel();

Add a method to HotelSearchingAction SFSB class:

//public void jrebel(){ System.out.println("JRebel!"); }

Modify C:\java\jboss-seam-2.1.1.GA\examples\ build.xml to add the rebel.xml reference below so the rebel.xml is included in jboss-seam-booking.jar:

<fileset id="jar.resources" dir="${resources.dir}">

<include name="import.sql" />

<include name="seam.properties" />

<include name="rebel.xml"/>

<include name="seam.quartz.properties" />

<include name="\*.drl" />

<include name="META-INF/persistence.xml" />

<include name="META-INF/ejb-jar.xml" />

<include name="META-INF/orm.xml" />

<include name="META-INF/components.xml" />

<include name="META-INF/jboss.xml" />

<include name="META-INF/security-config.xml" />

<include name="META-INF/security-rules.drl" />

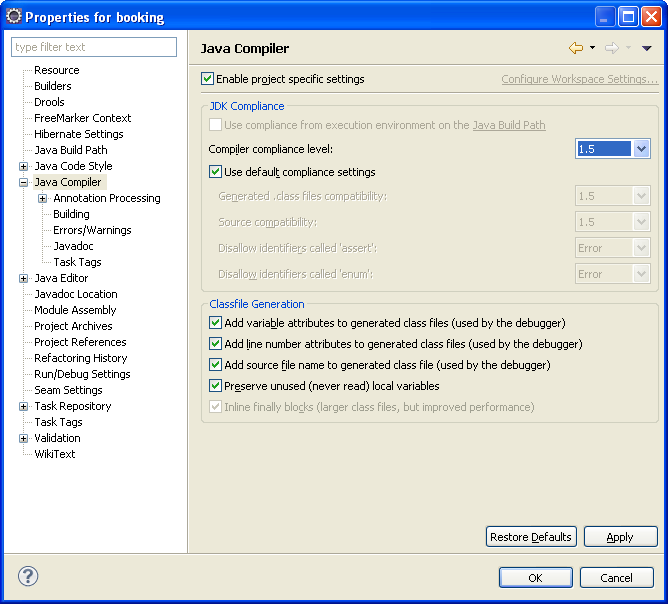
<include name="META-INF/standard-jaxws-endpoint-config.xml" />

<include name="\*.jpdl.xml" />

<exclude name=".gpd\*" />

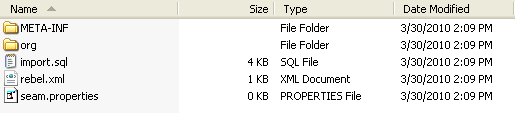
</fileset>

In case you have JDK 1.6 and 1.5 installed on your workstation, make sure your Eclipse compiler settings are set to 1.5 by going to project -> properties -> java compiler and checking the “enable project specific settings” and then select 1.5 for “compiler compliance level”.



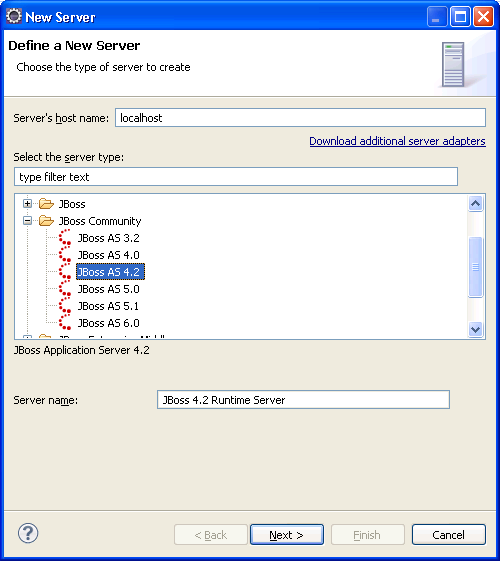
Build and deploy app using ‘ant clean explode’ command (run ‘ant unexplode’ or ‘ant undeploy’ first as required)

In this directory of the exploded EAR, C:\java\jboss-4.2.2.GA\server\default\deploy\jboss-seam-booking.ear\jboss-seam-booking.jar, you should see the rebel.xml file. If it’s not there then the rebel.xml was placed in the incorrect directory (it should be in booking/resources) and double-check the build.xml re-config above.

****

If not already created, add a JBoss server in the JBoss server view

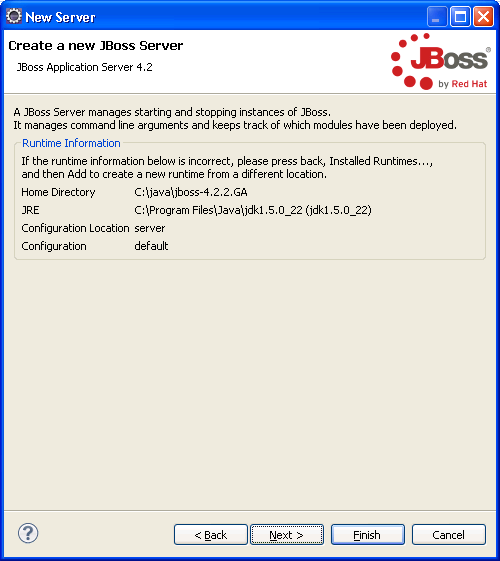
Right-click in white space; select new 🡪 server



Select JBoss AS 4.2, click Next

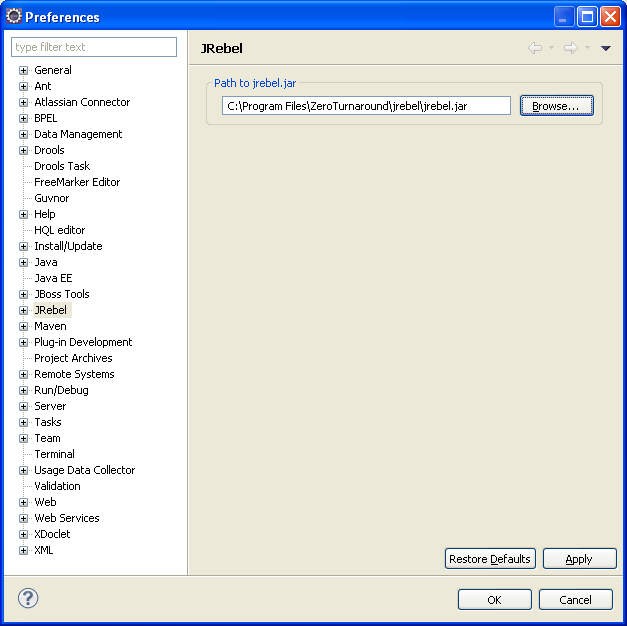


Enter your JBoss home directory, select JDK 1.5, select default instance, click Next

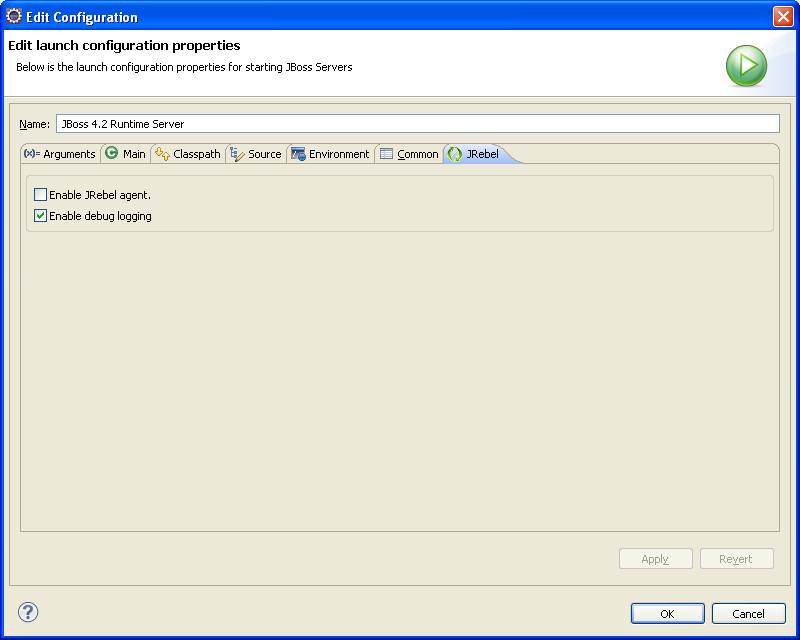


Click Finish

Enter the path to jrebel.jar via Window 🡪 preferences 🡪 JRebel



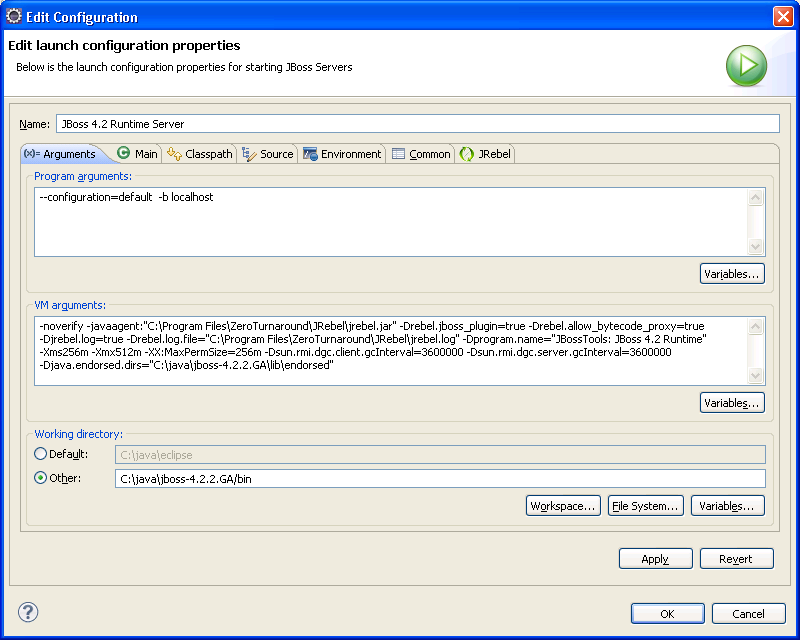
Click the enable debug logging checkbox in JRebel tab of Edit launch configuration properties.



Make sure the VM args in the edit configuration screen (from JBoss server view, double-click your server, click open launch configuration link) look similar to this (the JRebel-specific args are highlighted and required):

-noverify -javaagent: "C:\Program Files\ZeroTurnaround\JRebel\jrebel.jar" -Drebel.jboss\_plugin=true -Drebel.allow\_bytecode\_proxy=true -Drebel.log=true -Dprogram.name="JBossTools: JBoss 4.2 Runtime" -Xms256m -Xmx512m -XX:MaxPermSize=256m –

Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.endorsed.dirs="C:\java\jboss-4.2.2.GA\lib\endorsed"



Start JBoss AS 4.2 via JBoss Server view (right-click on JBoss 4.2 Runtime Server and select debug)

Check the Eclipse console and/or JBoss server.log to make sure that there are no exceptions during server startup and booking app deployment

At the very top of your Eclipse console, you should see this output:

#############################################################

JRebel 3.0.1-SNAPSHOT (201004161402)

(c) Copyright ZeroTurnaround OU, Estonia, Tartu.

Over the last 1 days JRebel prevented

at least 0 redeploys/restarts saving you about 0 hours.

You are running JRebel evaluation license.

You have 2 days until the license expires.

You will see this notification until you obtain a

full license for your installation.

Visit www.jrebel.com for instructions on obtaining

a full license. If you wish to continue your evaluation

please e-mail to support@zeroturnaround.com.

If you think you should not see this message contact

support@zeroturnaround.com or check that you have your

license file in the same directory as the JAR file.

The following plugins are disabled at the moment:

\* Hibernate Plugin (set -Drebel.hibernate\_plugin=true to enable)

Reloads SessionFactory when configuration changes

\* Log4j plugin (set -Drebel.log4j-plugin=true to enable)

Reloads full configuration of log4j

\* OpenJPA Plugin (set -Drebel.openjpa\_plugin=true to enable)

Reloads EntityManagerFactory when configuration changes

\* Stripes plugin 1.0.11 (set -Drebel.stripes\_plugin=true to enable)

Adds reloading of Stripes ActionBeans.

#############################################################

**You should see output similar to this:**

JRebel: Directory 'C:\java\jboss-seam-2.1.1.GA\examples\booking\exploded-archives\jboss-seam-booking.jar' will be monitored for changes.

**which is based on the metadata provided in the rebel.xml file for the directory for JRebel to monitor for changes:**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<application xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://www.zeroturnaround.com"* xsi:schemaLocation=*"http://www.zeroturnaround.com/alderaan/rebel-2\_0.xsd"*>

<classpath>

<dir name=*"C:/java/jboss-seam-2.1.1.GA/examples/booking/exploded-archives/jboss-seam-booking.jar"*>

</dir>

</classpath>

</application>

**You should also see JRebel specific logging for the various enabled JRebel plugins:**

=============================== [JRebel JBoss Plugin] ===============================

You can disable this plugin using Agent Settings or by setting -Drebel.jboss\_plugin=false.

------------------------------------------------------------------------------------------

Description: Supports adding new methods to ejb interfaces and injecting new

@EJB fields.

=============================== [/JRebel JBoss Plugin] ==============================

=============================== [JRebel Hibernate Plugin] ===============================

You can disable this plugin using Agent Settings or by setting -Drebel.hibernate\_plugin=false.

------------------------------------------------------------------------------------------

Description: Reloads SessionFactory when configuration changes

=============================== [/JRebel Hibernate Plugin] ==============================

=============================== [JRebel Mojarra (JSF-RI) Plugin] ===============================

You can disable this plugin using Agent Settings or by setting -Drebel.mojarra\_plugin=false.

------------------------------------------------------------------------------------------

Description: Turns on Mojarra development mode. Reloads JSF configuration and

reconfigures managed beans.

=============================== [/JRebel Mojarra (JSF-RI) Plugin] ==============================

JRebel: Directory 'C:\java\jboss-seam-2.1.1.GA\examples\booking\exploded-archives\jboss-seam-booking.jar\org\jboss\seam\example\booking' will be monitored for changes.

=============================== [JRebel Seam Plugin] ===============================

You can disable this plugin using Agent Settings or by setting -Drebel.seam\_plugin=false.

------------------------------------------------------------------------------------------

Description: Turns on seam debug mode and reconfigures modified seam components.

=============================== [/JRebel Seam Plugin] ==============================

=============================== [JRebel Facelets Plugin] ===============================

You can disable this plugin using Agent Settings or by setting -Drebel.facelets\_plugin=false.

------------------------------------------------------------------------------------------

Description: Turns on facelet development mode, facelet reloading and reloads

facelet tag libraries.

=============================== [/JRebel Facelets Plugin] ==============================

**First Demo: adding interface method to SFSB**

For the first demo, we are going to add a local interface method and SFSB impl class method for hot redeploy in the hotelSearching Seam component.

1. Stop the JBoss server
2. Execute ‘ant clean unexplode explode’ from the Ant view in Eclipse (one target at a time) or all at once from command line here: C:\java\jboss-seam-2.1.1.GA\examples\booking
3. Start the JBoss server in debug mode (or regular start/run mode)
4. Navigate to <http://localhost:8080/seam-booking>
5. Login using demo/demo credentials
6. On the main.xhtml page, click the new <JRebel!> Button
7. You should see this in your console or server.log:

16:01:14,474 ERROR [[Faces Servlet]] Servlet.service() for servlet Faces Servlet threw exception

javax.el.MethodNotFoundException: /main.xhtml @42,71 action="#{hotelSearch.jrebel}": Method not found: HotelSearchingAction:4si4k5i-6szw7-g7zb48zq-1-g7zbee2g-i.jrebel()

1. Edit the HotelSearching local interface by uncommenting the newly added line:

**public** **void** jrebel();

1. Edit the HotelSearchingAction SFSB class by uncommenting the newly added line:

**public** **void** jrebel(){ System.*out*.println("JRebel!"); }

1. **navigate back to main.xhtml (click browser back button)**
2. click <JRebel!> button again
3. You will see the following output in the console: “JRebel!”
4. You will also see the following output in the console:

JRebel-JBoss: Reconfiguring EJB 'jboss.j2ee:service=EJB3,name=HotelSearchingAction' [org.jboss.seam.example.booking.HotelSearchingAction]

**Second Demo: adding @In to SFSB**

Now for the second demo we are going to dynamically add a Seam @In injection to the HotelSearchingAction SFSB and see what happens.

1. Write this class:

**package** org.jboss.seam.example.booking;

**import** org.jboss.seam.annotations.AutoCreate;

**import** org.jboss.seam.annotations.Name;

@Name("testInject")

@AutoCreate

**public** **class** TestInject {

**private** String foo = "bar";

**public** String getFoo() {

**return** foo;

}

**public** **void** setFoo(String foo) {

**this**.foo = foo;

}

}

Add the following commented code to the HotelSearchingAction SFSB:

//@In(required=**false**)

//**private** TestInject testInject;

Then change the jrebel() method to look like this (we’re adding the commented line to output the value of foo):

**public** **void** jrebel(){

System.*out*.println("JRebel!");

//System.*out*.println("testInject.getFoo() = "+testInject.getFoo());

}

1. Stop the JBoss server
2. Execute ‘ant clean unexplode explode’ from the Ant view in Eclipse (one target at a time) or all at once from command line here: C:\java\jboss-seam-2.1.1.GA\examples\booking
3. Start the JBoss server in debug mode (or regular start/run mode)
4. Navigate to <http://localhost:8080/seam-booking>
5. Login using demo/demo credentials
6. On the main.xhtml page, click the new <JRebel!> button
7. You will see the output in the console: “JRebel!”
8. Uncomment the commented lines in HotelSearchingAction SFSB above.
9. Save (if using Eclipse, make sure Project 🡪 Build automatically is checked)
10. Click the <JRebel!> button again
11. You will see the following output in the console:

10:22:04,519 INFO [STDOUT] JRebel!

10:22:04,519 INFO [STDOUT] testInject.getFoo() = bar

So JRebel was able to hot redeploy the Seam @In addition to our EJB!

This concludes the tutorial for using JRebel 3 with Seam 2.x. There will be followup Weld/JRebel3 and Seam3/JRebel3 tutorials as well in the near future so stay tuned.

Thanks!

**References**

<http://www.zeroturnaround.com/reference-manual/>

<http://www.zeroturnaround.com/jrebel/faq/>

<http://repos.zeroturnaround.com/svn/>

## [Reloading Java Classes 101: Objects, Classes and ClassLoaders](http://www.zeroturnaround.com/blog/reloading-objects-classes-classloaders/)

## [Reloading Java Classes 201: How do ClassLoader leaks happen?](http://www.zeroturnaround.com/blog/rjc201/)

## [RJC301: Classloaders in Web Development — Tomcat, GlassFish, OSGi, Tapestry 5 and so on](http://www.zeroturnaround.com/blog/rjc301/)

## [RJC401: HotSwap and JRebel — Behind the Scenes](http://www.zeroturnaround.com/blog/reloading_java_classes_401_hotswap_jrebel/)

## [RJC501: How Much Does Turnaround Cost?](http://www.zeroturnaround.com/blog/reloading_java_classes_social_cost_of_turnaround/)

## <http://relation.to/Bloggers/IncrementalHotDeploymentWithSeamAndJBossTools>

## <http://seamframework.org/Community/SpecialClassloaderForWeldHotIncrementalDeployment>

## <http://seamframework.org/Community/ForThoseInterestedInHotDeployment>

# <http://java.dzone.com/articles/5-jrebel-features-you-couldn%E2%80%99t>