Virtual Developer Day

Oracle WebLogic Server 12*c*

Modern, Lightweight Development
with Java EE 6 and Oracle Coherence

ORACLE
TECHNOLOGY NETWORK

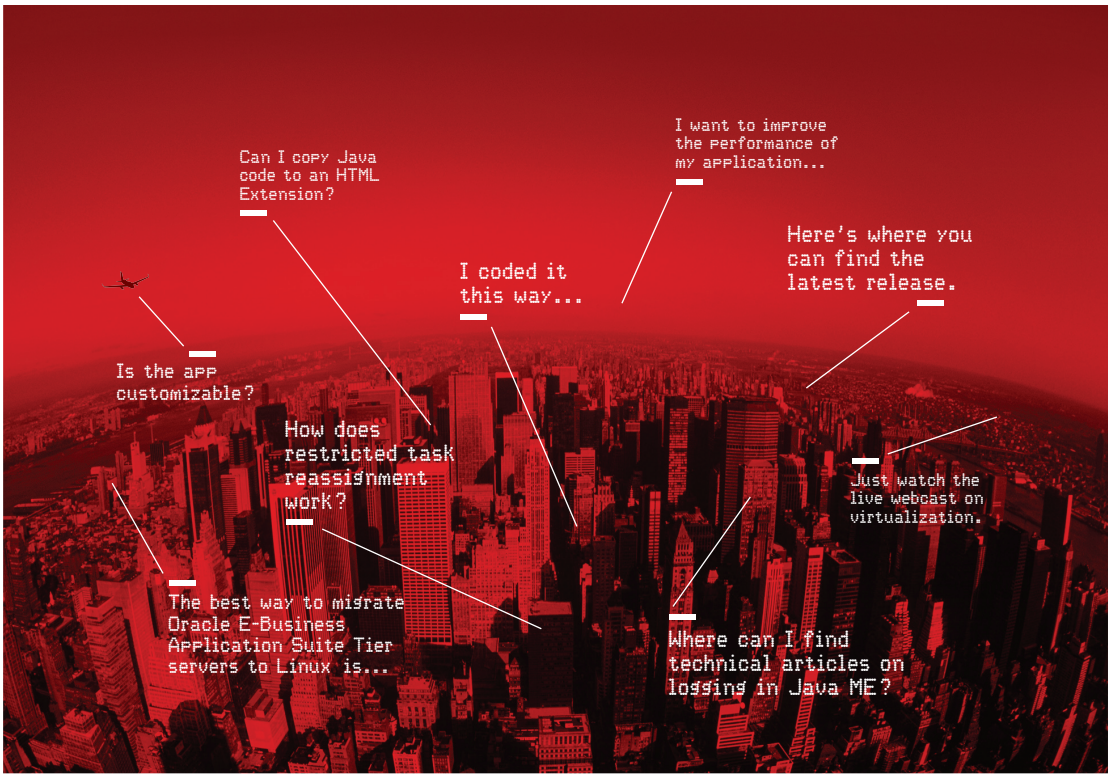
Hands on Lab Manual

Setting Up the Coherence Example Projects in Eclipse

ORACLE
TECHNOLOGY NETWORK

http://www.oracle.com/technetwork

ORACLE



Lab and User Details

Operating System User:	oracle/welcome1
Root:	root/welcome1
OS Environment settings:	/home/oracle/.bashrc
User Home Directory:	/home/oracle
Java Home	/labs/wls1211/jdk160_29
Coherence Home:	/labs/wls1211/coherence_3.7
Lab-specific resources:	/home/oracle/labs/Coh_labs/lab_resources
Eclipse workspace:	/home/oracle/labs/Coh_labs/workspace
Coherence examples:	/labs/wls1211/ coherence_3.7/examples

Coherence Example directory structure

/labs/wls1211/coherence_3.7/examples/java/src
All example source that you can either run from the command line or import into an IDE, which is the goal of this lab. The examples are in the com.tangosol.examples.<example name> package. The classes for objects stored in the cache used by the examples are in the com.tangosol.examples.pof package.

/labs/wls1211/coherence_3.7/examples/java/classes
The class files output from a build. This directory will not exist until the build script is executed.

/labs/wls1211/coherence_3.7/examples/java/resource/config
The common coherence configuration files required by the examples.

/labs/wls1211/coherence_3.7/examples/java/resource/<example name>
If an example has configuration that is required instead of the common configuration, it will have its own directory. The security example uses configuration files from java/resource/security.

/labs/wls1035/coherence_3.7/examples/resource
The data file used for the contacts LoaderExample: contacts.csv.

$COHERENCE_HOME/lib
Coherence libraries used for compiling and running the examples.

Introduction

In this lab, you will be simulating working in a development environment that consists of the following software components:

- Oracle Coherence 3.7

- Eclipse 3.7 (Helios) with Oracle Enterprise Pack for Eclipse (OEPE) 12.1.1

In this lab today, you will be performing the following:

- **Learning capabilities** of the OEPE's Coherence tools
- Working with **organized as collections of code** that show how to use one or more features
- Show how to use Eclipse to **quickly configure and run** a default cacheserver
- **Configuring** a subset of the feature examples and security examples

Coherence Features Examples:

- Basic Data Access - "Getting", "putting" and "removing" data from the Coherence Data Grid.
- Data Loading - Loading example data into the Coherence Data Grid.
- Parallel Querying - Querying the Coherence Data Grid including the use of indexes.
- Observable - Listening for changes to data in the Coherence Data Grid.
- Processing - Co-locating data processing with the data itself in the Coherence Data Grid.
- Query Language - How to use the new 3.6 Coherence Query Language.

Coherence Security Examples:

- Password Example - Requiring a password to access Coherence.
- Access Control Example - Simplified role based access control.
- Password Identity Transformer - creates a custom security token that contains the required password and then adds a list of Principal names.
- Password Identity Asserter - asserts that the security token contains the required password and then constructs a Subject based on a list of Principal names.
- Entitled Cache Service - wraps a cache service for access control.
- Entitled Invocation Service - wraps an invocation service for access control.
- Entitled Named Cache - wraps a named cache for access control.

Section 1: Starting a default cache server

In this section you create basic project and start a cache server with a default configuration, essentially making the connection between the design time IDE and the coherence run time JVM, called a cache server.

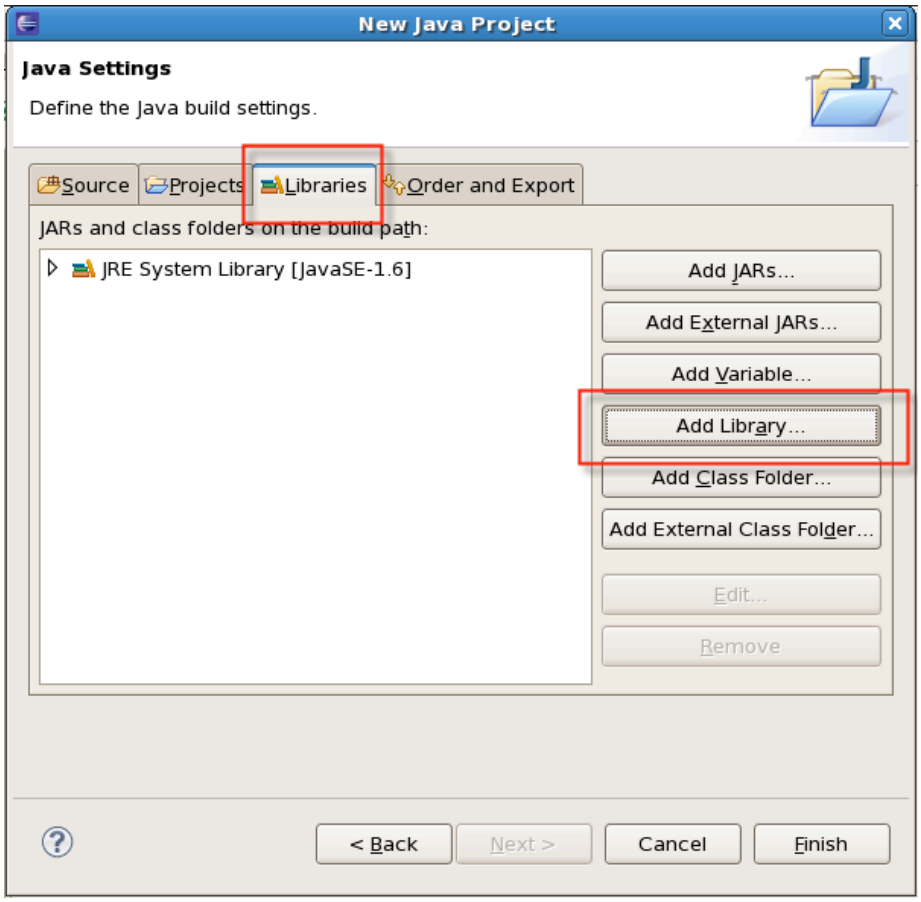
Launch OEPE 12.1.1 and choose workspace

Double-click on the desktop icon for OEPE to launch, and choose the workspace /home/oracle/labs/WS_labs/labHome/Coh_labs/workspace

Create new Java Project

Click on File->New Project. In the Java grouping, expand and choose Java Project.

Name it **examples_project** and click **Next**. Then in the libraries tab, choose add library.



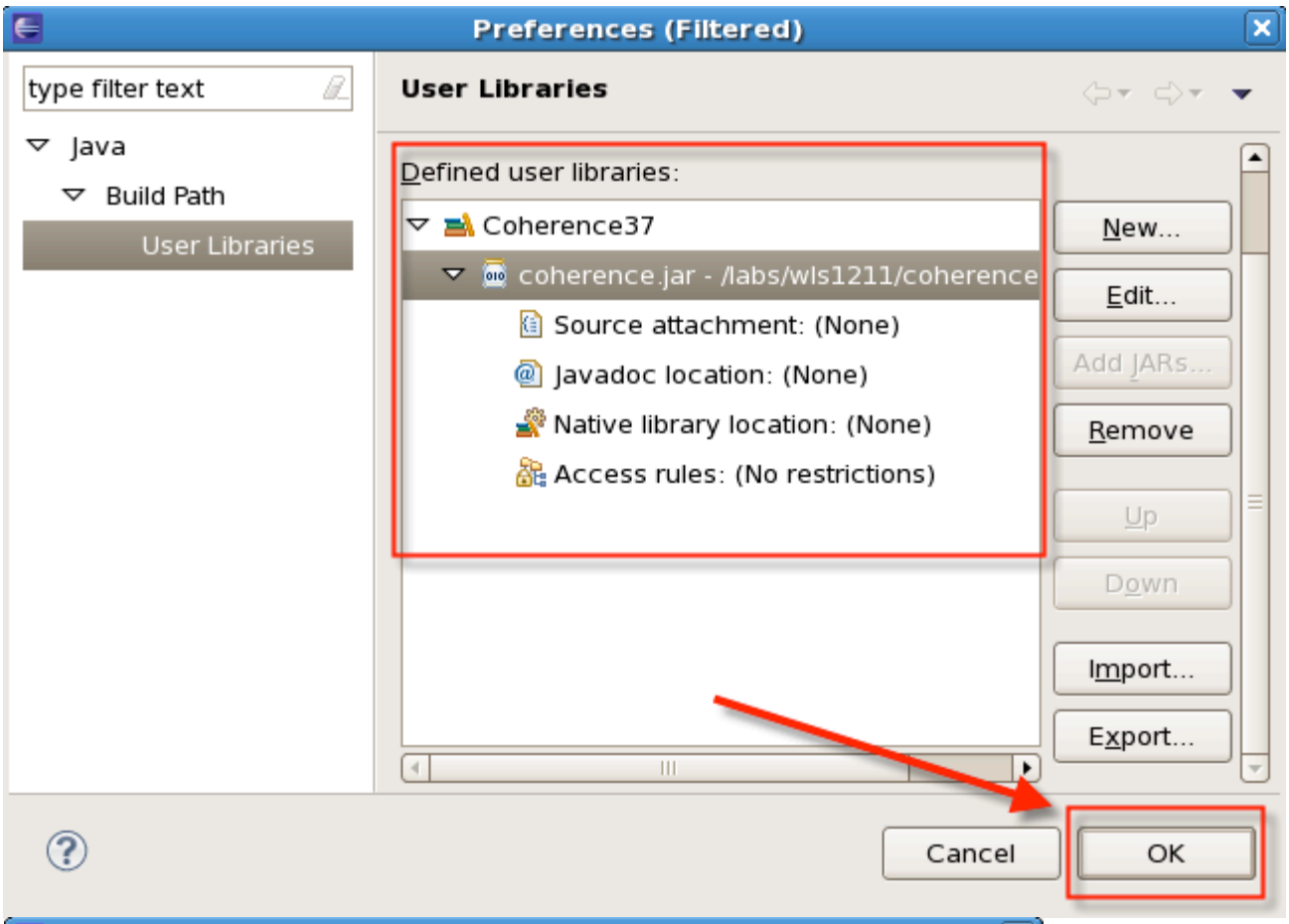
In this case we'll use a User Library, so choose that from the list, and click **Next**. Then click the "**User Libraries** button to define it for the first time. Click **New** and call it **Coherence371**. Then click **add JARs** to tell it which library to use.

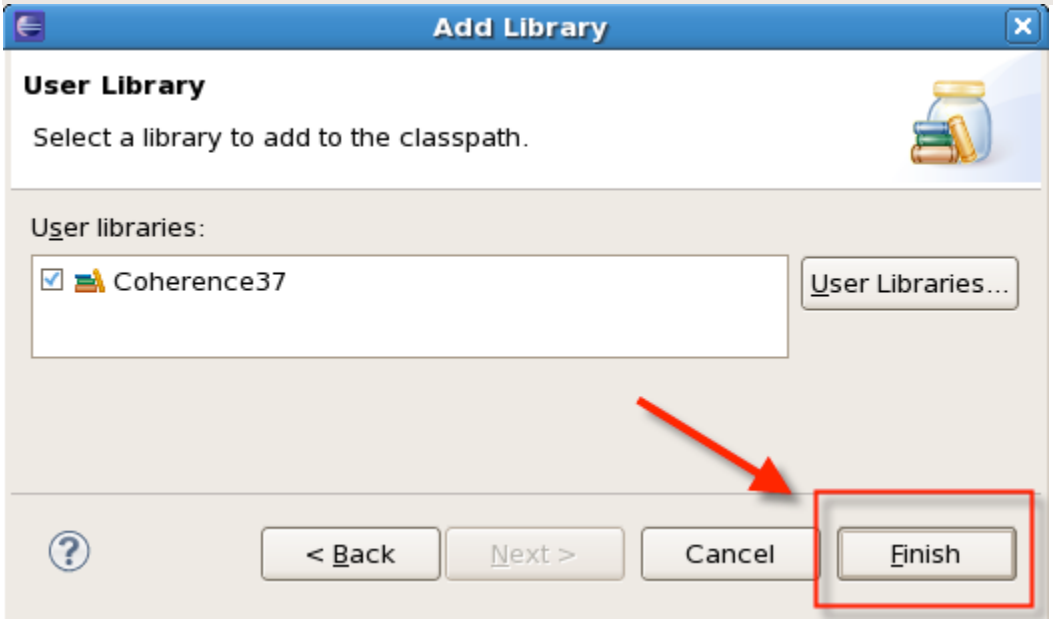
The library in this case ships with Coherence 3.7.1 and WebLogic 12.1.1, and is located in

/labs/wls1211/coherence_3.7/lib/coherence.jar

So simply navigate there in the JAR selection dialog and choose **OK**.

Then click **Finish** to complete the User Library Definition.



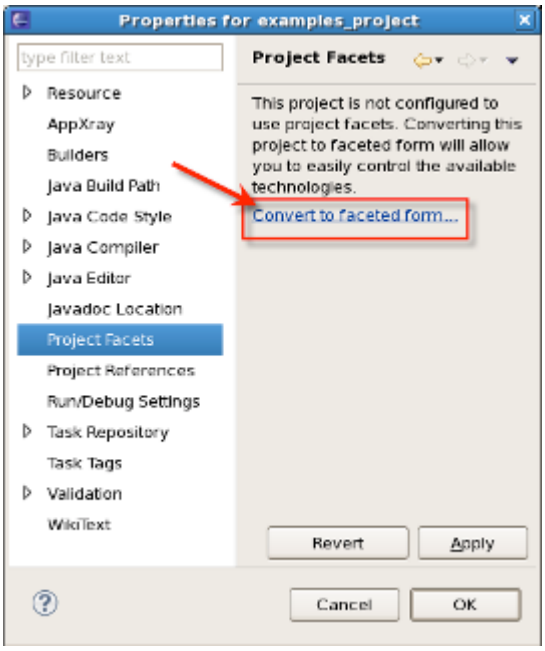


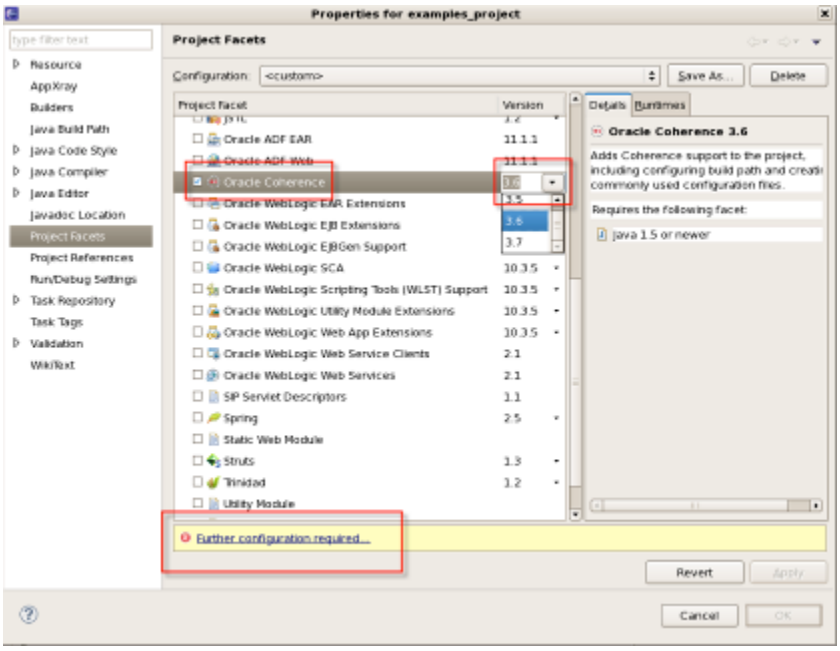
Then click **Finish** again on the “new Java Project” Dialog to complete the project definition.

(If you are prompted to switch perspectives to “Java” click **Yes**.)

Configure new Java Project

Right click on the newly created Java project and choose **Properties**. In the Project Facets properties, click on the hyperlink for **Convert to Faceted Form…** this will allow you to add facets to your Java project. Facets are an Eclipse feature, but vendors provide the facets for a given technology to help create default configuration files for project, put libraries in the right place, so you don’t have to do it all manually. Once converted, check the box for the **Oracle Coherence facet** and change the version to 3.7.1 in the dropdown. (You will have to re-size the window to see it). Then click on the hyperlink **Further Configuration Required…** to configure the facet.





In the resulting “Modify Faceted Project” dialog -- make sure the Coherence371 user library you previously defined is **checked**. Leave the other defaults and click **OK**, which will generate a default:

- operational configuration override file
- cache server configuration
- data model configuration file for the portable object format (POF).

We’ll delete these later and use the files from the coherence examples project, but this will suffice for starting a default cache server from within OEPE.

Lastly, click **OK** in the Project Properties dialog to apply the changes.

Create Run Configuration for Default Cache Server

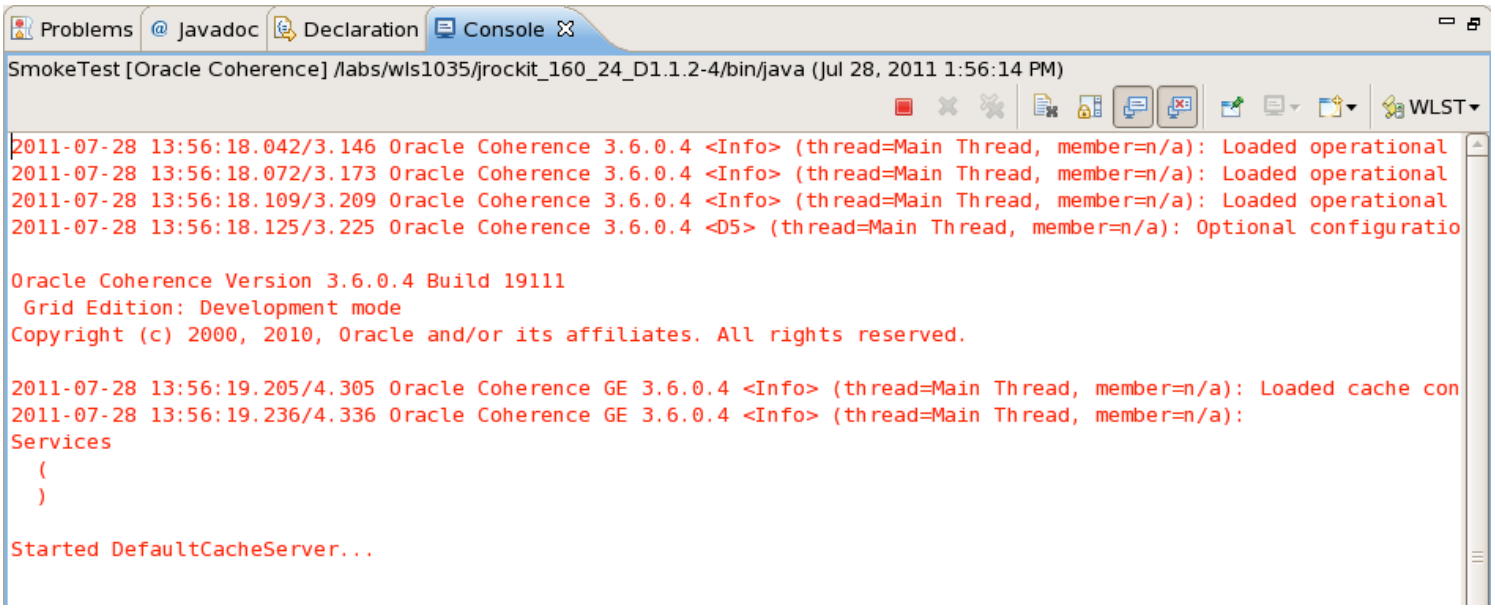
To give you the basic idea on setting up runtime cache servers, you'll start with a simple, default example.

Go to the main menu in OEPE and choose **Run->Run Configurations…**

From the list of run configurations on the left hand side of the dialog, choose **Oracle Coherence** by double clicking it in the list, and call it **SmokeTest**

In the main tab (shown by default) name the main class: **com.tangosol.net.DefaultCacheServer.** Then click **Apply** to preserve the changes.

Click **Run** to start the cache server, and monitor the output in Eclipse's **Console** tab to see the running process. If needed, the console can be enabled from **Window->Show View->Console**.



Section 2:
Setup the Coherence Feature Examples

http://coherence.oracle.com/display/EXAMPLES/Coherence+Features+Examples

- Basic Data Access - "Getting", "putting" and "removing" data from the Coherence Data Grid.
- Data Loading - Loading example data into the Coherence Data Grid.
- Parallel Querying - Querying the Coherence Data Grid including the use of indexes.
- Observable - Listening for changes to data in the Coherence Data Grid.

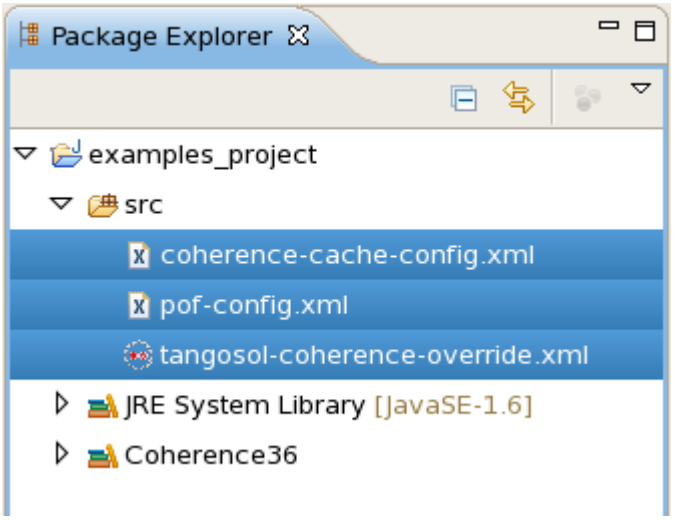
- Processing - Co-locating data processing with the data itself in the Coherence Data Grid.
- Query Language - How to use the Coherence Query Language.

Importing Coherence Example Project Code into OEPE

This corresponds to the “Contacts” source folder and example code shipped with the coherence product.

We’ll start by deleting the automatically generated code from the project:

coherence-cache.config.xml, pof-config.xml, tangosol-cohernce-override.xml



DO THIS NEXT PART CAREFULLY. ERRORS HERE MAKE FOR DIFFICULT TROUBLESHOOTING LATER, AND CONFUSING ERROR MESSAGES.

In the **Package Explorer** View, Create the following packages in examples_project/src, by right clicking on examples_project/src and choosing **New → Package**.

- com.tangosol.examples.contacts
- com.tangosol.examples.pof
- com.tangosol.examples.security

Create the following folder in examples_project/src, by right clicking on examples_project/src and choosing **New → Folder**.

- resource

Create the following packages in examples_project/src, by right clicking on examples_project/src and choosing **New → Package**.

- resource.config
- resource.security

Use the OS file system explorer to immediately drag and drop the coherence examples source from file system to the packages/folder you created in OEPE. When prompted during drag and drop, it's recommended to copy the files into your source eclipse project rather than link to them in place so you retain the originals.

1. Copy the contacts.csv file in this folder:
   /labs/wls1211/coherence_3.7/examples/resource/

→ To this folder: src/resource

* NOTE: you many need to use the **Window → Show View → Navigator** view to drag and drop this source file into the project since Eclipse's Package Explorer tends to hide empty folders in the project view by default.

2. Copy all source in this folder:
   /labs/wls1211/coherence_3.7/examples/java/src/com/tangosol/examples/contacts

→ To this package: src/com.tangosol.examples.contacts

3. Copy all source in this folder:
   /labs/wls1211/coherence_3.7/examples/java/src/com/tangosol/examples/pof

→ To this package: src/com.tangosol.examples.pof

4. Copy all source in this folder:
   /labs/wls1211/coherence_3.7/examples/java/src/com/tangosol/examples/security

→To this package: src/com.tangosol.examples.security

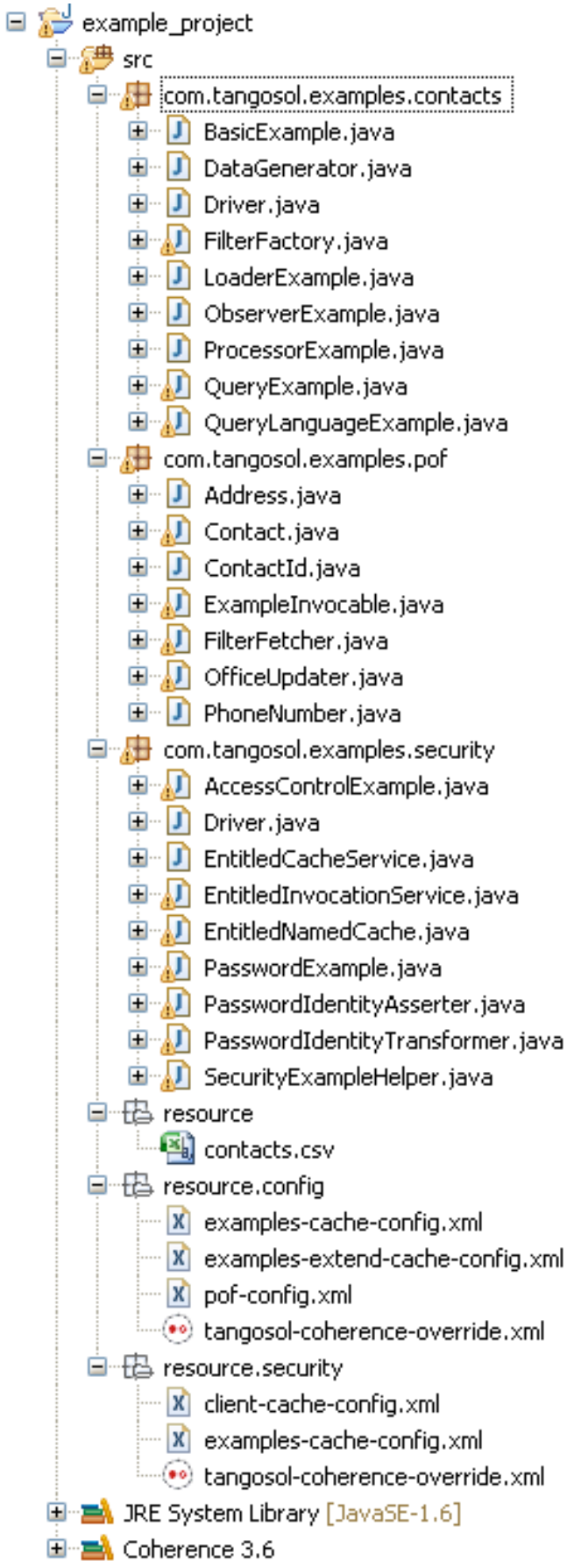
5. Copy all source in this folder:
   /labs/wls1211/coherence_3.7/examples/java/resource/config

→ To this package: src/resource.config

6. Copy all source in this folder:
   /labs/wls1211/coherence_3.7/examples/java/resource/security

→ To this package: src/resource.security

The resulting should look like this, or it needs to be fixed:


example_project
src
com.tangosol.examples.contacts
BasicExample.java
DataGenerator.java
Driver.java
FilterFactory.java
LoaderExample.java
ObserverExample.java
ProcessorExample.java
QueryExample.java
QueryLanguageExample.java
com.tangosol.examples.pof
Address.java
Contact.java
ContactId.java
ExampleInvocable.java
FilterFetcher.java
OfficeUpdater.java
PhoneNumber.java
com.tangosol.examples.security
AccessControlExample.java
Driver.java
EntitledCacheService.java
EntitledInvocationService.java
EntitledNamedCache.java
PasswordExample.java
PasswordIdentityAsserter.java
PasswordIdentityTransformer.java
SecurityExampleHelper.java
resource
contacts.csv
resource.config
examples-cache-config.xml
examples-extend-cache-config.xml
pof-config.xml
tangosol-coherence-override.xml
resource.security
client-cache-config.xml
examples-cache-config.xml
tangosol-coherence-override.xml
JRE System Library [JavaSE-1.6]
Coherence 3.6

(note: resource.config.tangosol-coherence-override.xml is created later and is not in the original source tree, so it's pictured incorrectly above.)

Editing com.tangosol.examples.contacts/Driver.java

Edit line 60 of com.tangosol.examples.contacts/Driver.java

String sFile = asArg.length > 1 ? asArg[1] + "/../resource/" +

Delete the "/../resource/" +

So that line 60 reads:

String sFile = asArg.length > 1 ? asArg[1] +

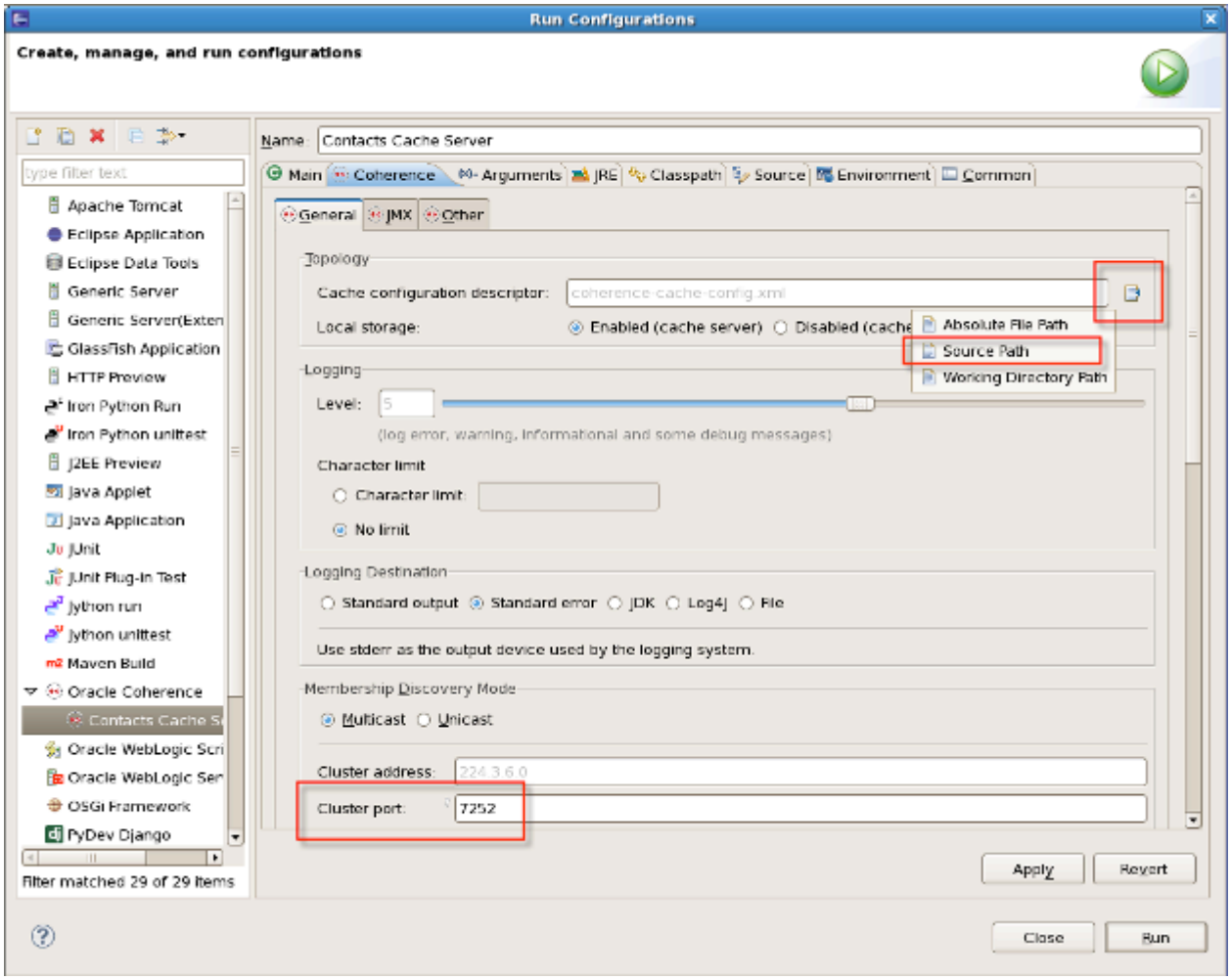
This corrects for a different path that what the out of the box examples project expects when building this at the command line.

Run Configuration for "Contacts" Cache Server

Edit the SmokeTest Cache Server configuration to be called Contact Cache Server. Go to the main menu in OEPE and choose **Run->Run Configurations…**

From the list of run configurations on the left hand side of the dialog, click on the **SmokeTest** run configuration in the left hand side of the dialog, and on the right in the **Main** tab, edit the name to be **Contacts Cache Server**.

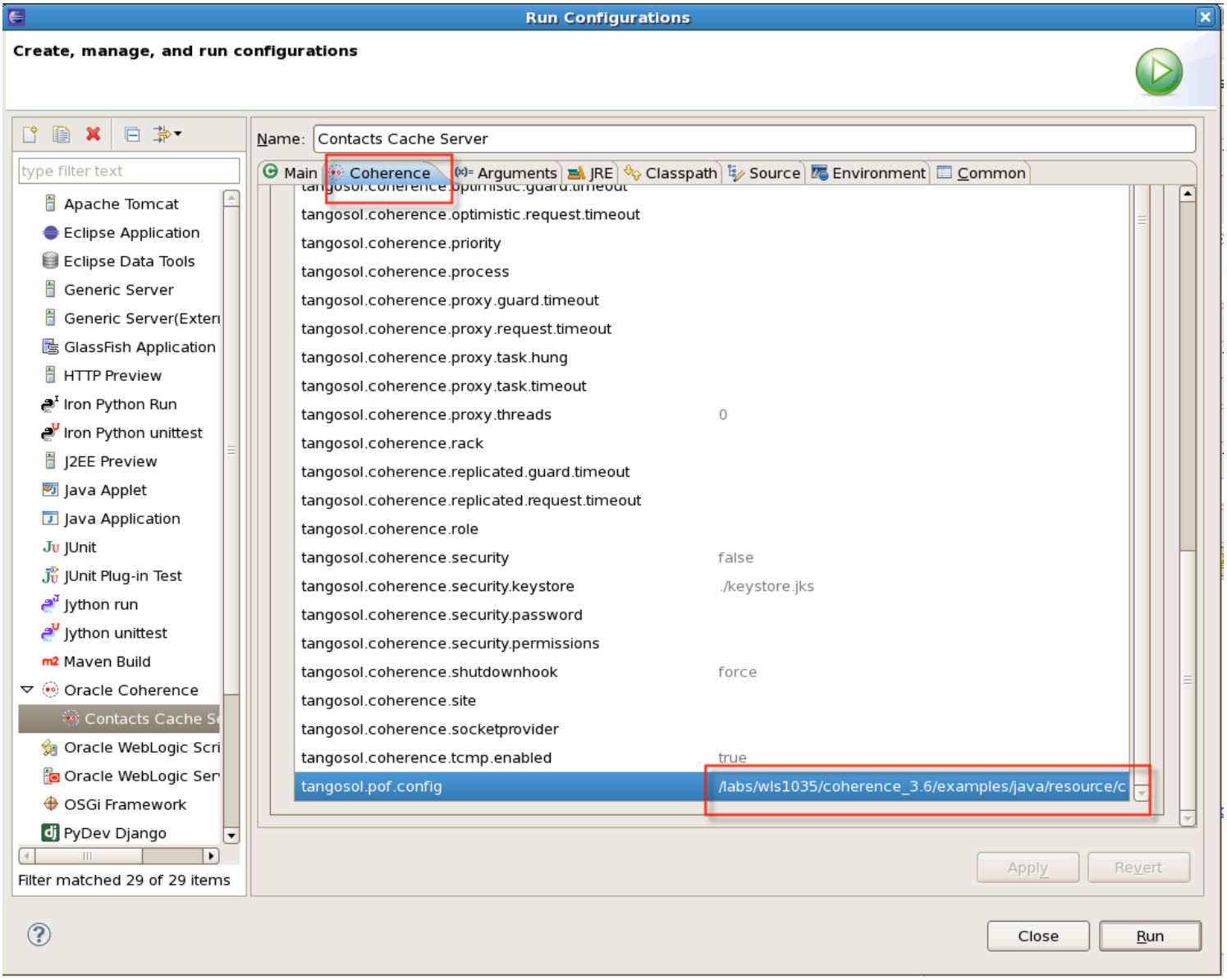
Then on the **Coherence** -> **General** tab, in the topology section, configure the cache server's configuration file to point at the examples project cache server configuration file you imported into the project earlier (shown below).



Click the **file browser** button as shown above, and choose **source paths**, and locate the /resource/config/examples-cache-config.xml file. Then configure the cluster port with some unique port number on your network segment so there are no multicast traffic conflicts.

Then click on the **coherence** -> **other** tab, scroll all the way to the bottom, and specify the location of the pof.xml for the contacts example (shown below). This will define the serialized object data model.

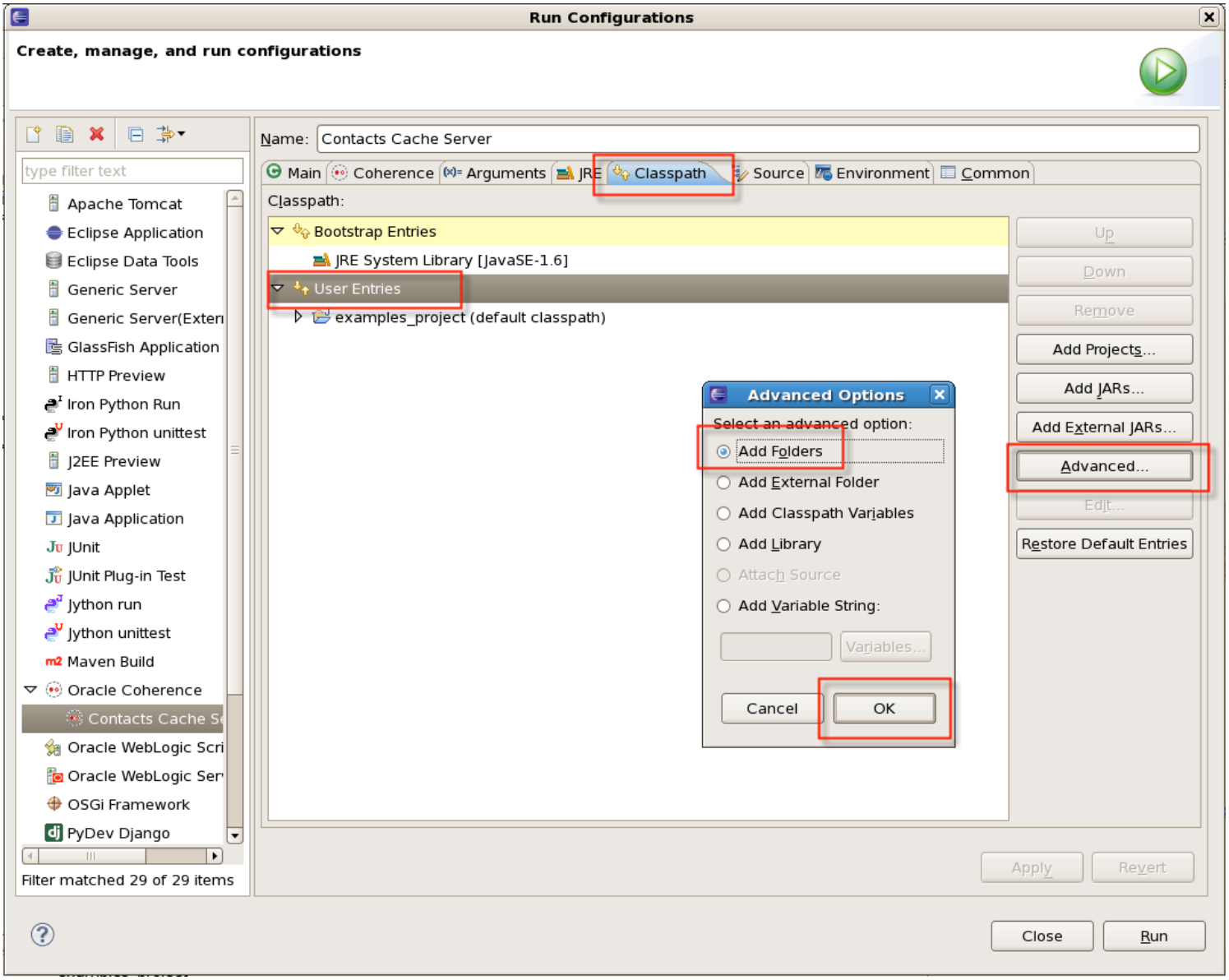
/labs/wls1211/coherence_3.7/examples/java/resource/config/pof-config.xml



Then add a classpath entry so the cache server can find the override configuration XML (tangosol-coherence-override.xml) files in the classpath. While still in the Run Configurations window, in the **Classpath** tab, click **User Entries,** then **Advanced**, and choose the radio button for **Add Folders**. Then add the directory containing the override file for the "Contacts" example project:

/examples_project/src/resource/config

..and click **Apply** when done. (shown below)



Troubleshooting tip: make doubly sure that the classpath entry gets saved! Try closing the Run Configuration Dialog and opening it again, ensuring your change is still there.

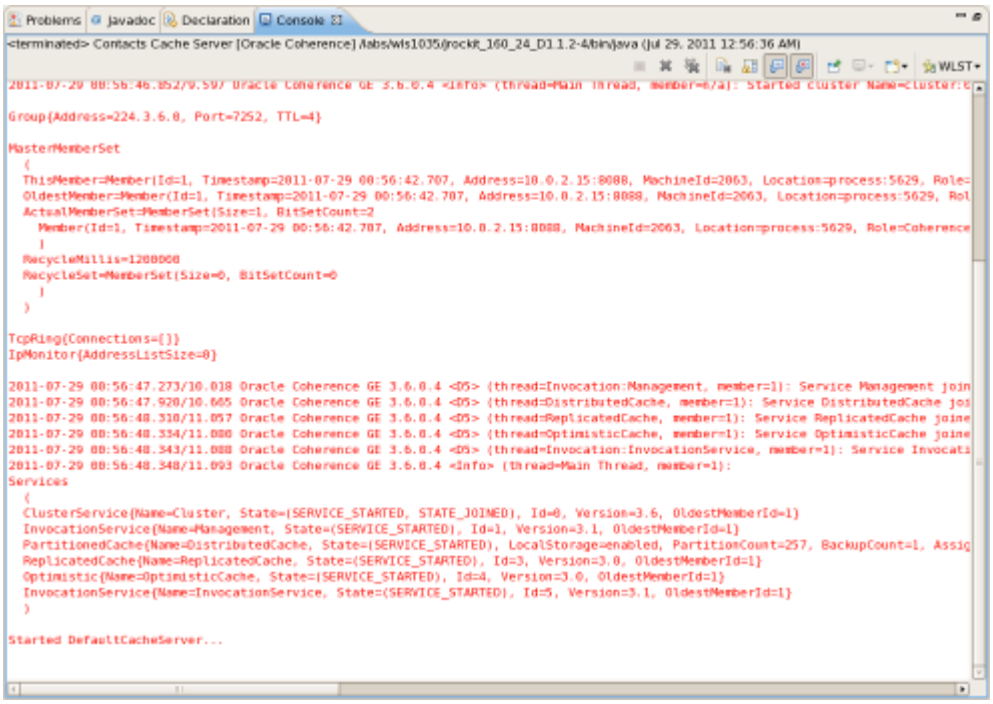
Before you run the server, let's take a moment to understand what's just happened with the classpath entry. This is set so that the project can locate the configuration XML in the classpath. There are several levels to set coherence configuration options at.

1) Java Options (-D) flags – options set here are the highest precedence
2) tangosol-coherence-override.xml – overrides tangosol-coherence.xml
3) tangosol-coherence.xml - options set here are the lowest precedence

The tangosol-coherence.xml descriptor is where you specify the operational and runtime elements that control clustering, communication, and data management services. The optional tangosol-coherence-override.xml override file is where you specify only the subset of the operational descriptor which you want to adjust. See "Operational Override File (tangosol-coherence-override.xml)" for more information.

All set to run!

Now click **Run** to start the cache server, and monitor the output in Eclipse's **Console** tab to see the running process. If needed, the console can be enabled from **Window->Show View->Console**.



Examine Cache Configuration for "Contacts" Cache Server

Now let's take a look at the cache configuration file for this server.

The simplest and most flexible way to create or edit caches in Coherence is to use OEPE's cache configuration descriptor editor to define attributes and names for your application's or cluster's caches, and to instantiate the caches in your application code referring to them by name that matches the names or patterns as defined in the descriptor.

This approach to configuring and using Coherence caches has a number of very important benefits. It separates the cache initialization and access logic for the cache in your application from its attributes and characteristics. This way your code is written in a way that is

independent of the cache type that will be utilized in your application deployment and changing the characteristics of each cache (such as cache type, cache eviction policy, and cache type-specific attributes, etc.) can be done without making any changes to the code whatsoever. It allows you to create multiple configurations for the same set of named caches and to instruct your application to use the appropriate configuration at deployment time by specifying the descriptor to use in the java command line when the node JVM is started.

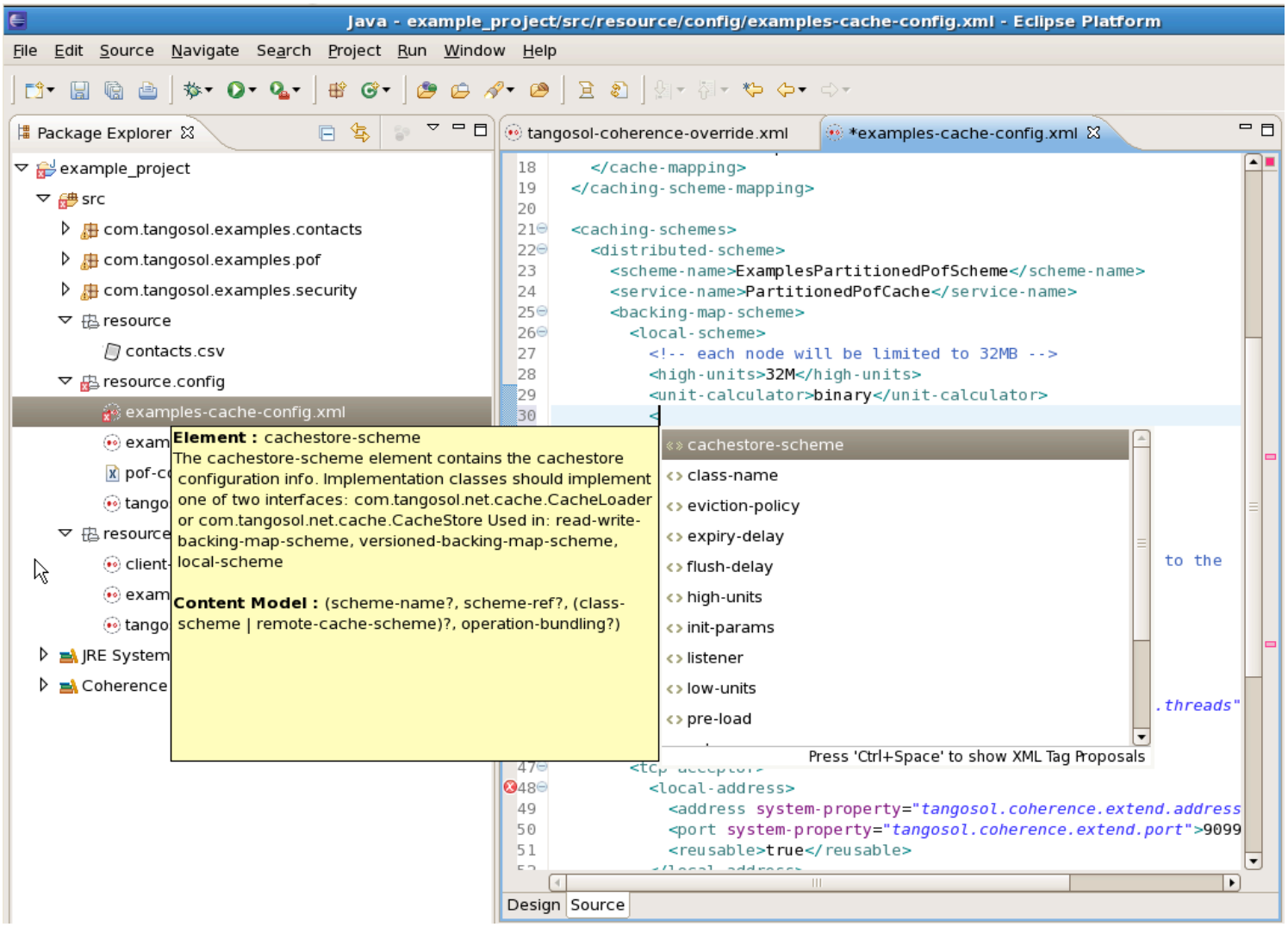
The cache attributes and settings are defined in the cache configuration descriptor. Cache attributes determine the cache type (what means and resources the cache will use for storing, distributing and synchronizing the cached data) and cache policies (what happens to the objects in the cache based on cache size, object longevity and other parameters).

The structure of the cache configuration descriptor (described in detail by the cache-config.dtd included in the coherence.jar) consists of three primary sections: *general*, *caching-schemes,* and the *caching-scheme-mapping* section.

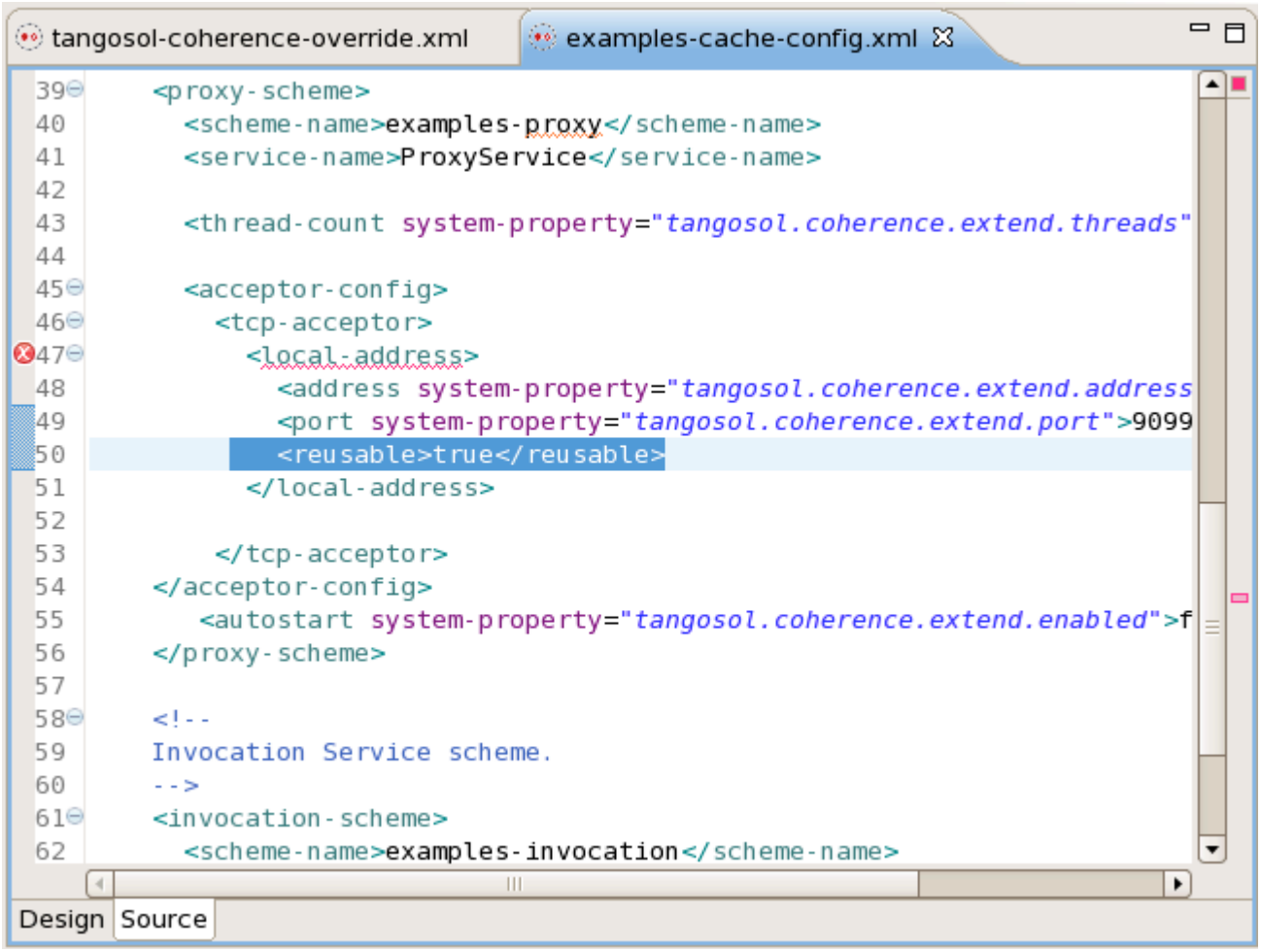
The caching-schemes section is where the attributes of a cache or a set of caches get defined. The caching schemes can be of a number of types, each with its own set of attributes. The caching schemes can be defined completely from scratch, or can incorporate attributes of other existing caching schemes, referring to them by their scheme-names (using a scheme-ref element) and optionally overriding some of their attributes to create new caching schemes. This flexibility enables you to create caching scheme structures that are easy to maintain, foster reuse and are very flexible.

In OEPE, double click on src/resource.config/examples-cache-config.xml to open the editor and then change to the **source view**. The editor provides code completion and integrated documentation for the schema with the usual eclipse **control + space command**.

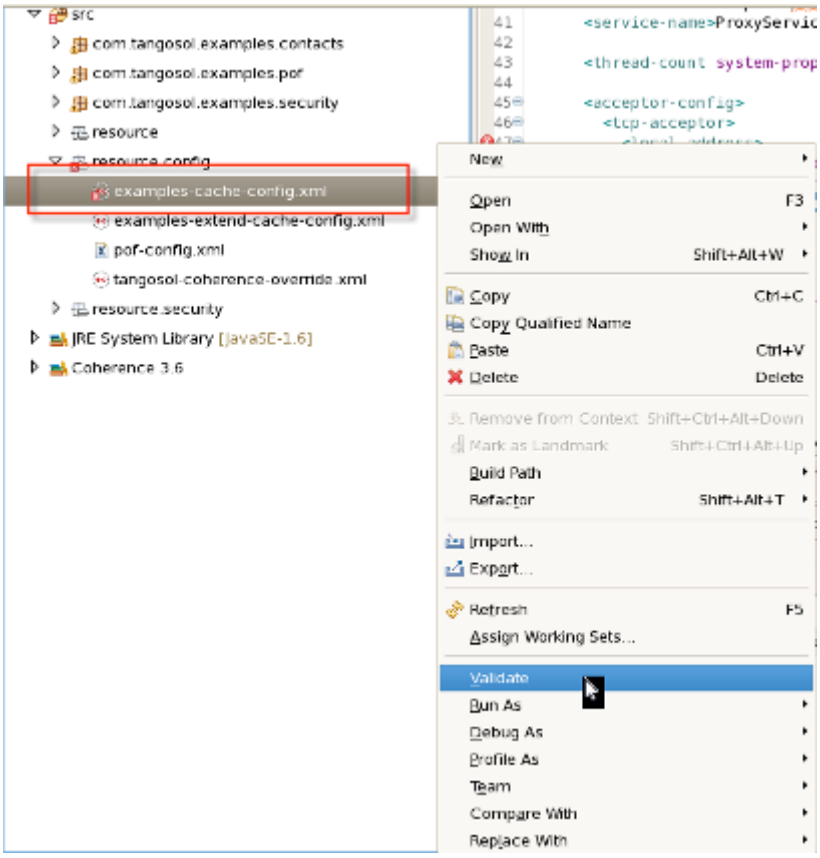
(shown below)



If you scroll down farther, you'll see that the validator has detected a problem with the file for you that needs your attention: the format of the TCP Acceptor IP address is incorrect. What is a TCP Acceptor? Simply **left click** and **hover** over the <tcp-acceptor> element in source view for a brief explanation. So, let's fix the problem the validator discovered. As of Coherence 3.5 the "reusable" element is deprecated; you should use the "reuse-address" element for the tcp-acceptor and tcp-initiator instead. So in the source view, find line 50, or simply click on the red mark in the margin. Delete the <resusable>true</reusable> element and save the file.



You may need to **right click** on src/resource.config/examples-cache-config.xml and choose **validate** to clear the validation error as shown below.



Run Configuration for “Contacts” Cache Server Client

Now we’ll connect a cache client in a second JVM process, to drive some load through the cache. This exercise just shows the most basic data access features of Coherence including getting, putting and removing data.

In the Run Configurations Window, duplicate the Contacts Cache Server, rename it “Contacts Driver Client”. Now let’s edit the new, duplicate configuration. In the **Main** tab, change the main class to: **com.tangosol.examples.contacts.Driver** so that this run configuration triggers the load driver class.

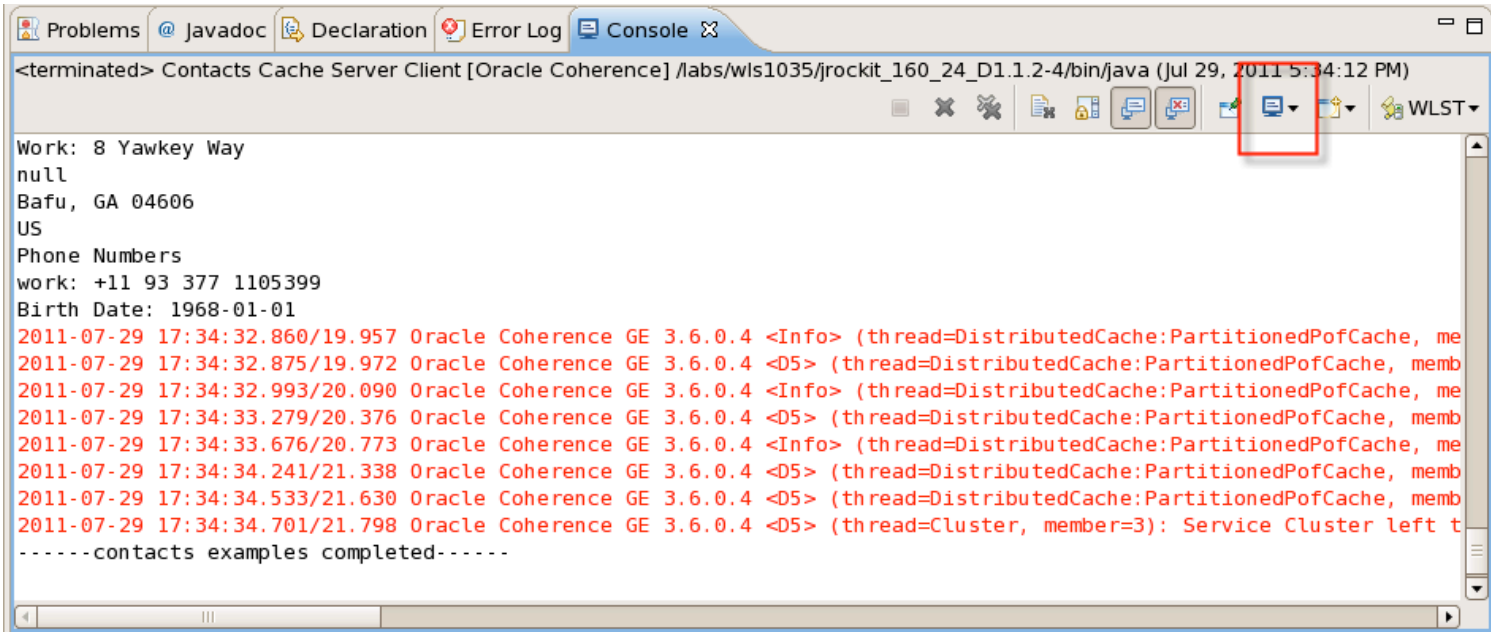
Then, in the **Arguments** tab in **Run Configurations** window, then in **Program Arguments** section, enter

contacts /labs/wls1211/coherence_3.7/examples/resource/

This is an argument we supply to the Driver.java class and is simply an artifact of the way the driver class is written. Reading the source for Driver.java will make it clear why these are necessary.

All set to run!

Now click **Run** to start the cache client (cache server should still be running), and monitor the output in Eclipse's **Console** tab to see the running process. If needed, the console can be enabled from **Window->Show View->Console**. You can switch console views by clicking the "Display Selected Console" button as shown below, as you'll have one for the cache server, and one for the cache client.



Section 3: Setup the Coherence Security Examples

http://coherence.oracle.com/display/EXAMPLES/Coherence+Security+Examples

- Password Example - Shows how a Coherence Proxy can require a password to access a cache.
- Access Control Example - Shows simplified role based access control.
- Password Identity Transformer - creates a custom security token that contains the required password and then adds a list of Principal names.
- Password Identity Asserter - asserts that the security token contains the required password and then constructs a Subject based on a list of Principal names.
- Entitled Cache Service - wraps a cache service for access control.
- Entitled Invocation Service - wraps an invocation service for access control.
- Entitled Named Cache - wraps a named cache for access control.

Goal:

To build 2 Run Configurations to demo security examples:

- A Proxy/Client for the cache server
- A Cache Server for the security examples

Proxies enforce the network segment point-of-origin that the client code is coming from, as you may not want your cache client join a cluster in a remote network segment (say, one that is geographically disparate). Instead, the proxy joins the cluster (and your "true" client communicates with that). So normally this topology would involve three elements:

Client → Proxy → Cache Server

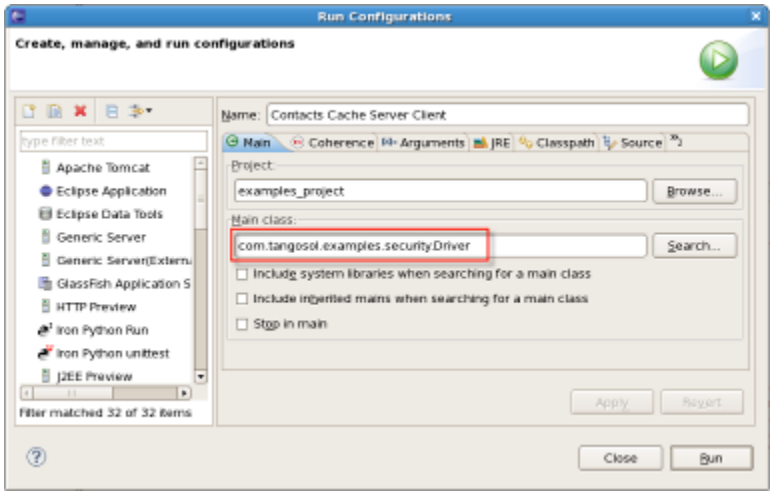
In this example, for purposes of the lab, it's just two elements:

Client → Cache Server/Proxy

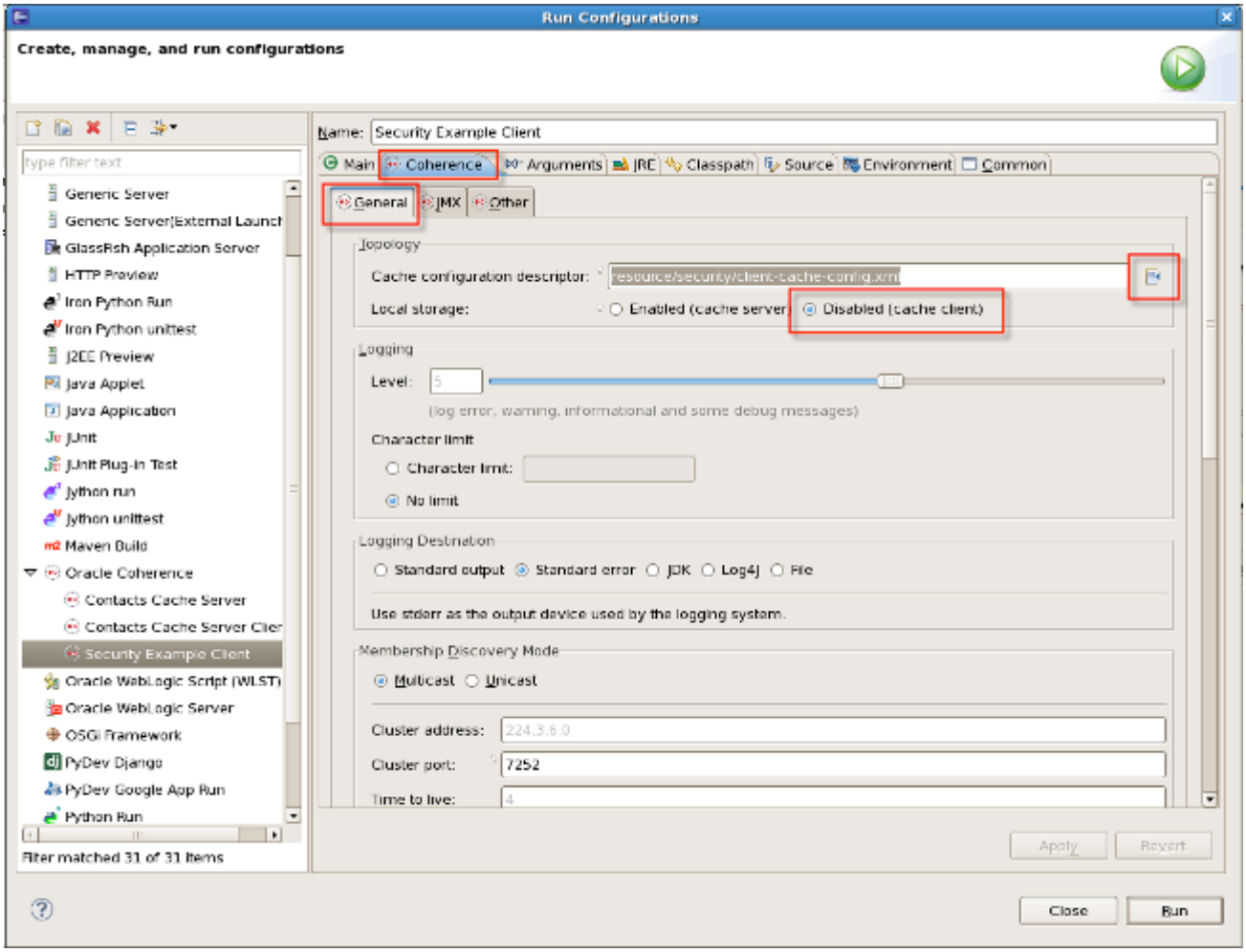
Rather than having all three.

Run Configuration for "Security" Cache Client

In the Run Configurations Window, right click on the Contacts Cache Server and choose **Duplicate**. Then rename it "Security Example Client". Now let's edit the new, duplicate configuration. In the **Main** tab, change the main class to: **com.tangosol.examples.security.Driver** so that this run configuration triggers the correct load driver class.



Then on the **Coherence → General** tab, in the topology section, configure the cache server's configuration file to point at the security project cache client configuration file you imported into the project earlier (shown below).



Click the **file browser** button as shown above, and choose **Source Path**, and locate the /resource/security/client-cache-config.xml file.

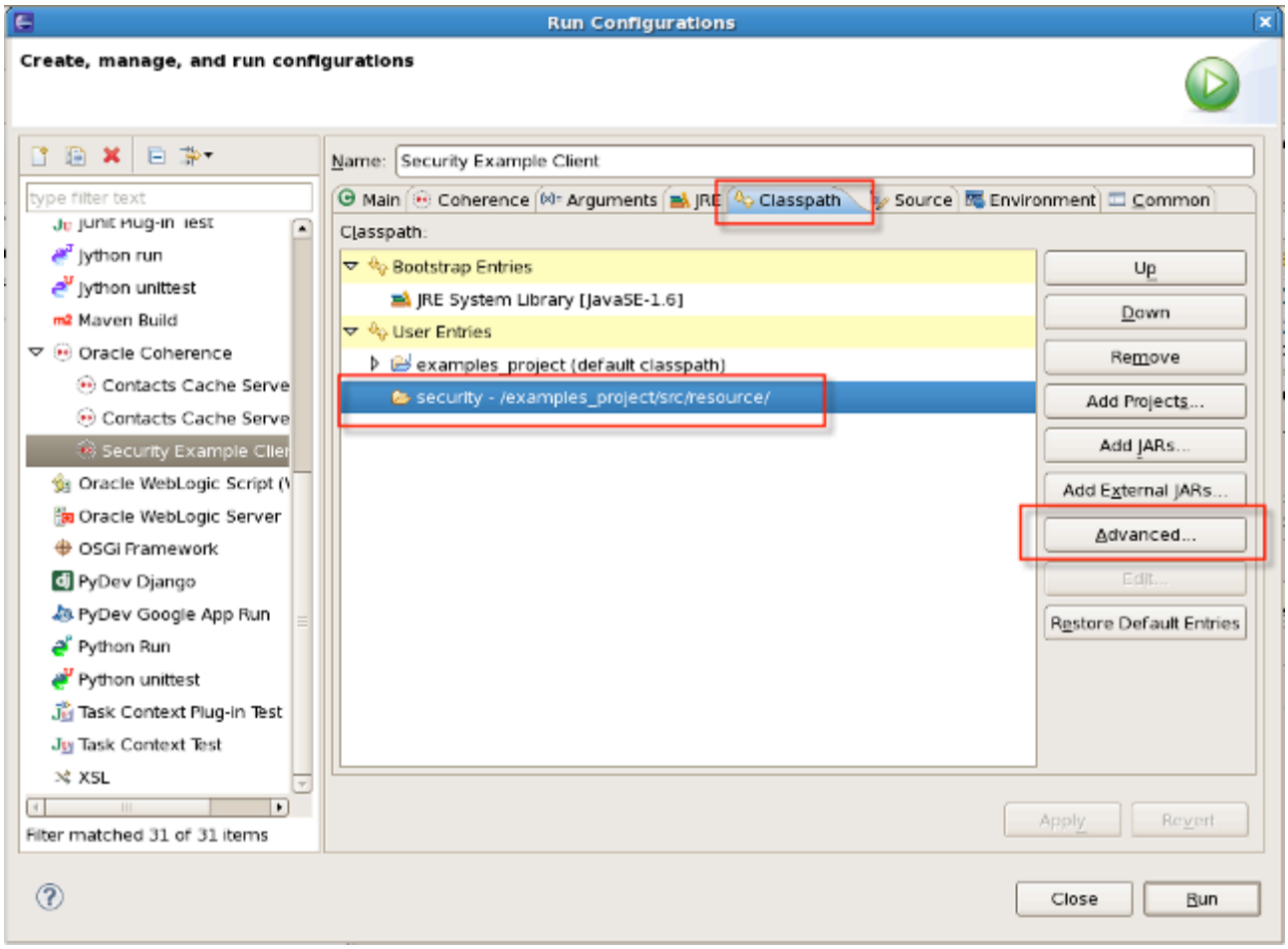
Also, while you don't have to do this as it's specified in the configuration files, change Local Storage to **Disabled (Cache Client)**, since this run configuration will function as a client, not a server.

Then modify a classpath entry so the cache server can find the configuration override XML (tangosol-coherence-override.xml) files in the classpath. While still in the Run Configurations window, in the **Classpath** tab, click **User Entries,** then **Advanced**, and choose the radio button for **Add Folders**. Then add the directory containing the override file for the "Security" example project:

/examples_project/src/resource/security

Then remove the classpath entry for

/examples_project/src/resource/config



Click **Apply** to preserve the changes.

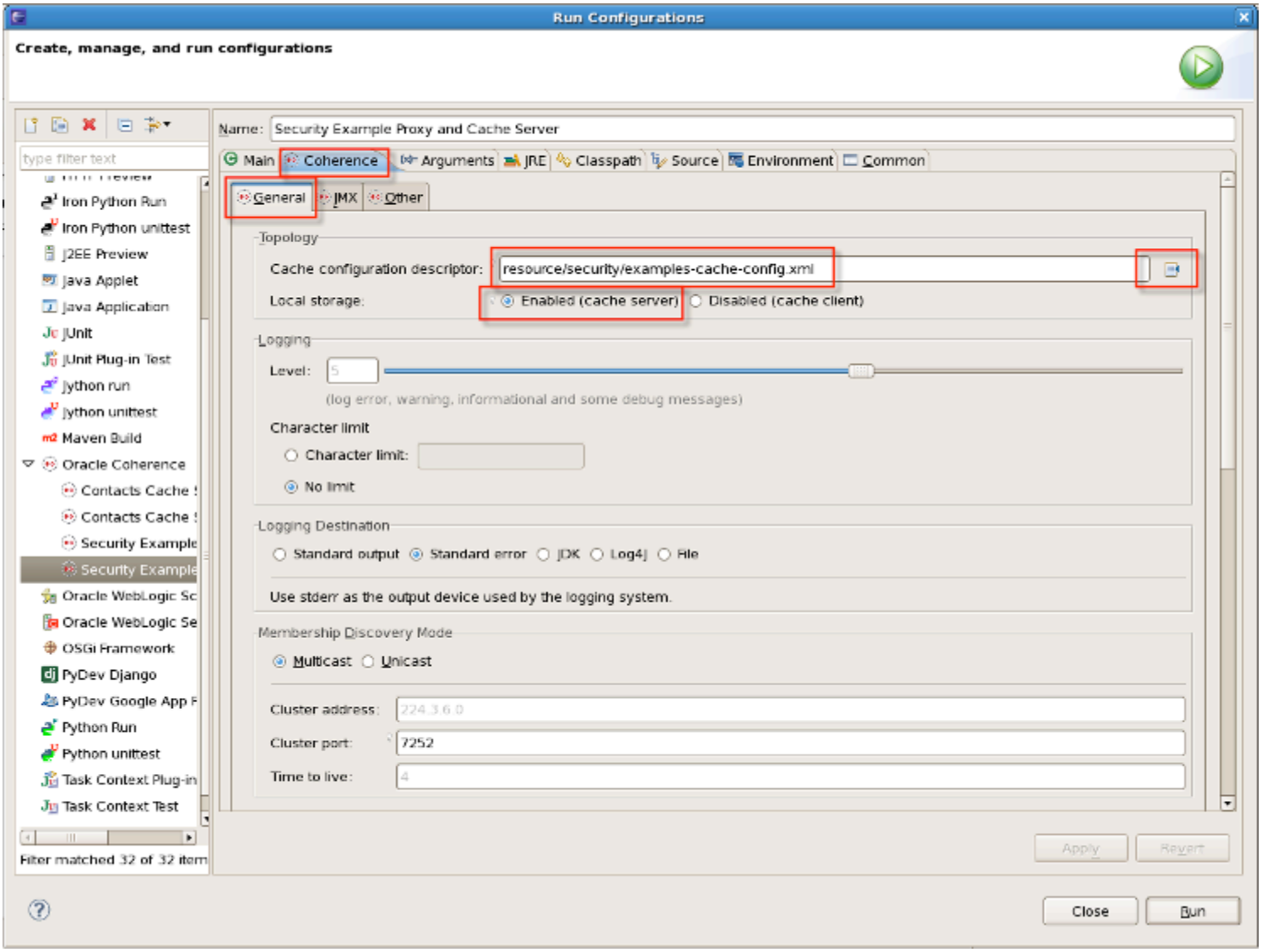
Run Configuration for “Security” Cache Server/Proxy

Again, we are configuring this proxy as a cache server for the sake of the lab and demonstration – as normally a proxy would be a client.

In the Run Configurations Window, right click on the Security Example Client and choose **Duplicate**. Then rename it “Security Example Proxy and Cache Server”. Now let’s edit the new, duplicate configuration. In the **Main** tab, change the main class to:

com.tangosol.net.DefaultCacheServer so that this run configuration triggers the default cache server class.

Then on the **Coherence → General** tab, in the topology section, configure the cache server's configuration file to point at the security project cache client configuration file you imported into the project earlier (shown below).



Click the **file browser** button as shown above, and choose **Source Path**, and locate the /resource/security/examples-cache-config.xml file.

Also, while you don't have to do this as it's specified in the configuration files, change Local Storage to **Enabled (Cache Server)**, since this run configuration will function as a combined proxy and server.

All the other values you need have been duplicated from previous configurations:

- POF config file
- Classpath setting
- Cluster port
- Arguments (or lack thereof, in the case of the security example)

Click **Apply** to preserve the changes.

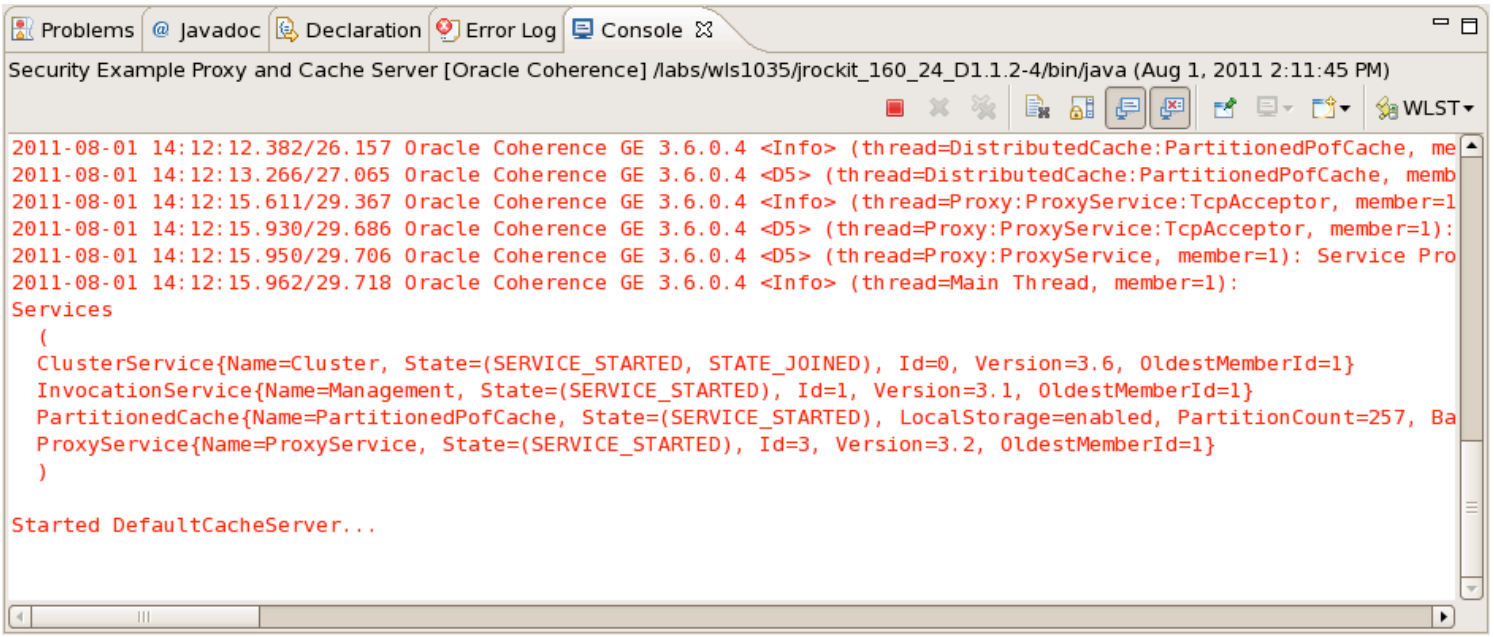
All set to run!

Now click **Run** to start the proxy/cache server, and monitor the output in Eclipse's **Console** tab to see the running process.

Then open the **Run → Run Configurations** window again, and select the security example client, and click **Run**. Monitor the output in Eclipse's **Console** tab to see the running process.

If needed, the console can be enabled from **Window->Show View->Console**. You can switch console views by clicking the "Display Selected Console" button as shown below, as you'll have one for the cache server/proxy, and one for the cache client.

"Access Denied" exceptions are expected since this is a security example.



Configuration Overrides

Starting with a quick review, as there are several levels to set coherence configuration options at:

1. Java Options (-D) flags – options set here are the highest precedence
2. tangosol-coherence-override.xml – overrides tangosol-coherence.xml
3. tangosol-coherence.xml - options set here are the lowest precedence

The tangosol-coherence.xml descriptor is where you specify the operational and runtime elements that control clustering, communication, and data management services. The optional tangosol-coherence-override.xml override file is where you specify only the subset of the operational descriptor which you want to adjust. See "Operational Override File (tangosol-coherence-override.xml)" for more information.

To see this in action with a simple example, **copy** and **paste** the override config file for the security examples to the contacts example:

src/resource.security/tangosol-coherence-override.xml

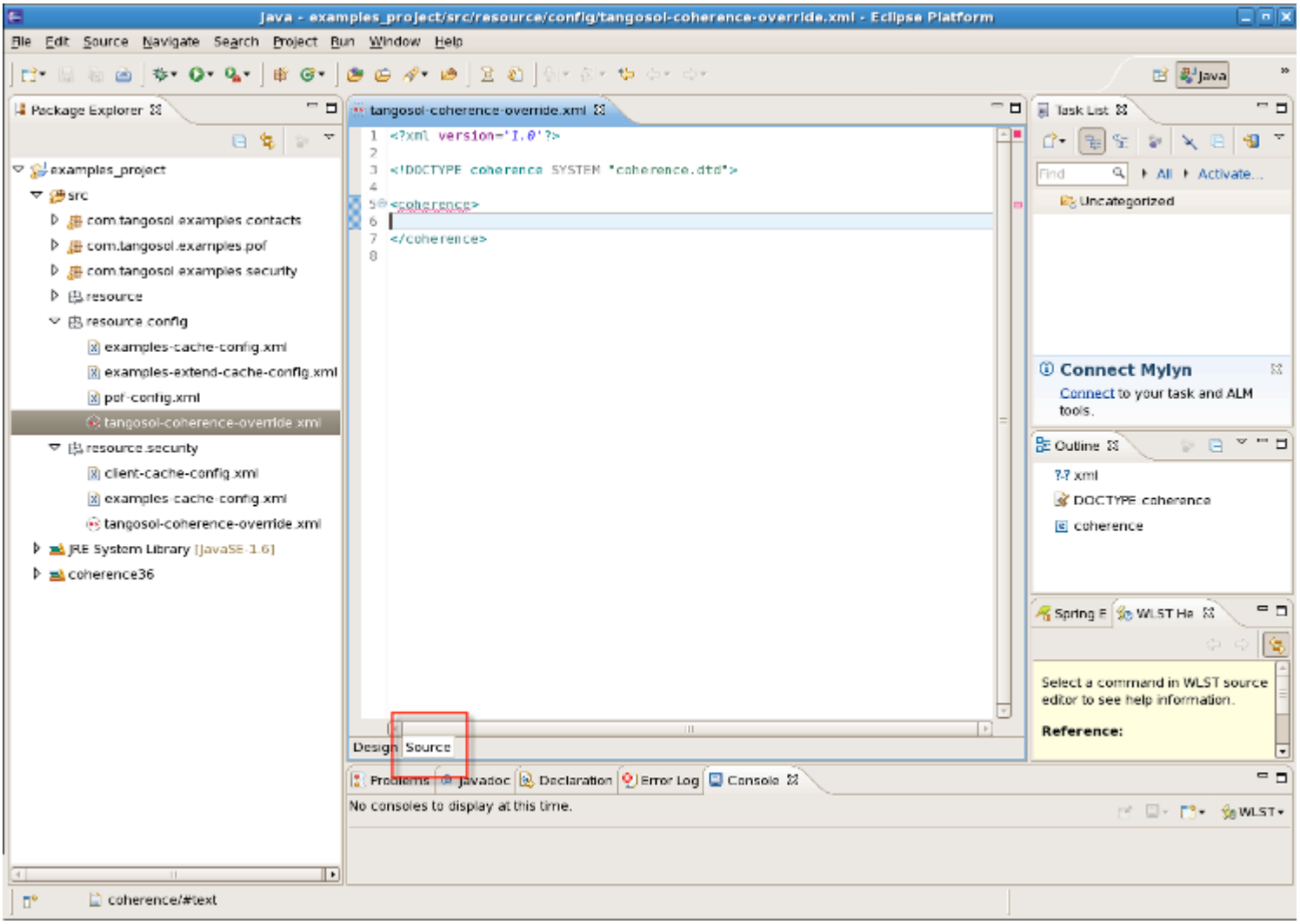
to

src/resource.config/tangosol-coherence-override.xml

In the new file that you've copied, go to the **source** view, and delete the overrides for specific to the security example by deleting the existing XML stanzas in src/resource.config/tangosol-coherence-override.xml, as shown below, so it looks like this:

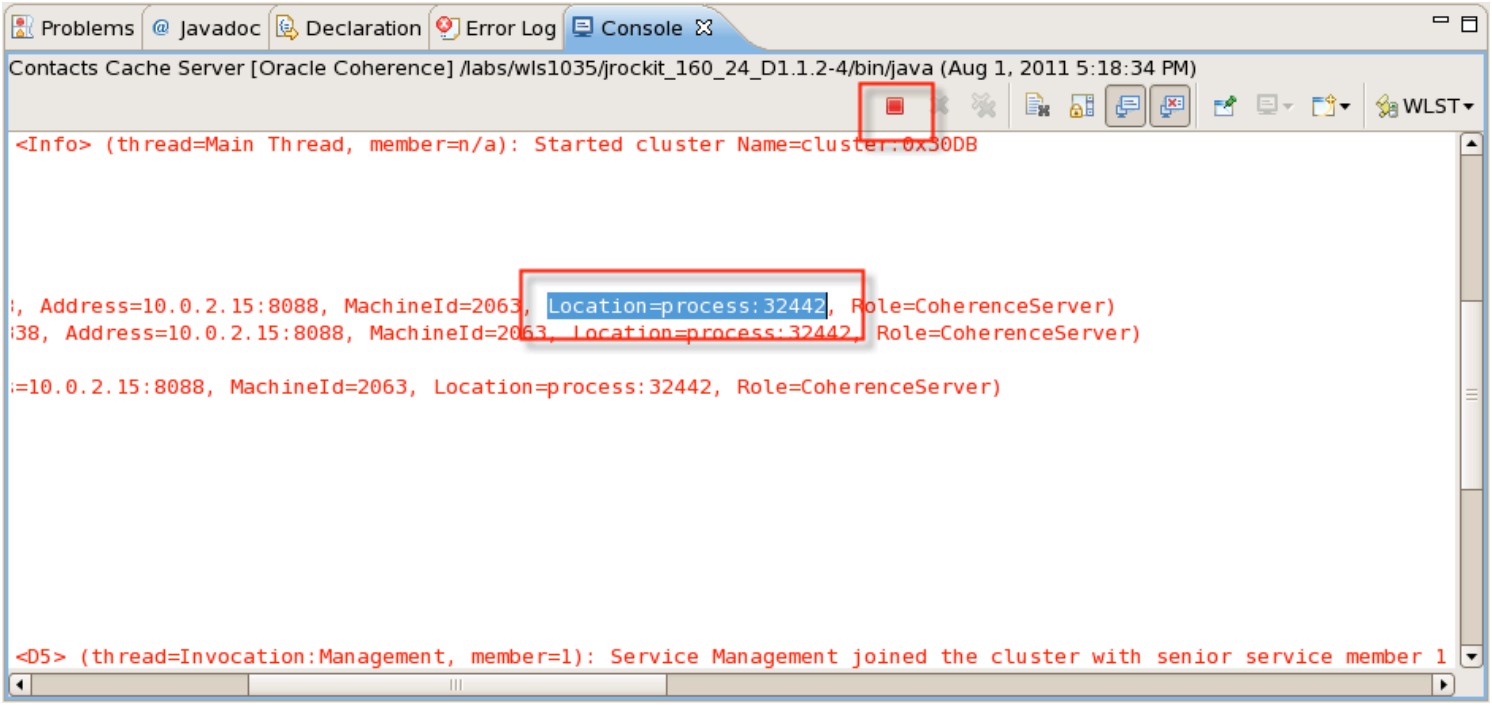
```
<?xml version='1.0'?>

<!DOCTYPE coherence SYSTEM "coherence.dtd">

<coherence>

</coherence>
```

Then save the file. (screenshot follows)



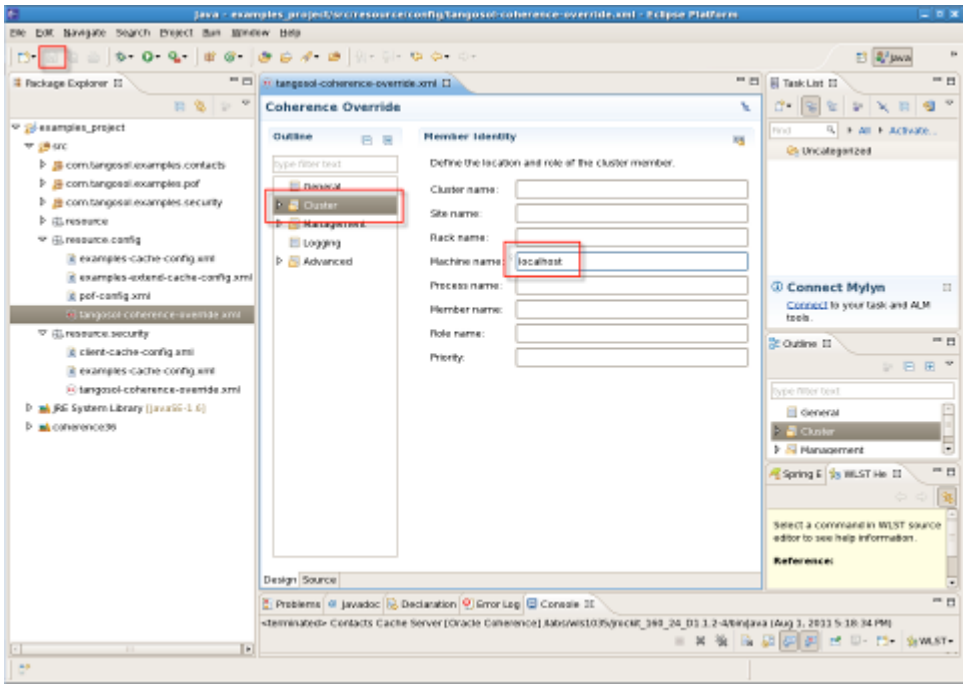
Before we edit in edit in some overrides to test the functionality, let's see how it behaves by default.

Go to the main menu in OEPE and choose **Run->Run Configurations…** and select the **Contacts Cache Server** and click **Run**. Scroll through the console output until you see the "Location=" value, and notice how it doesn't have a machine name.

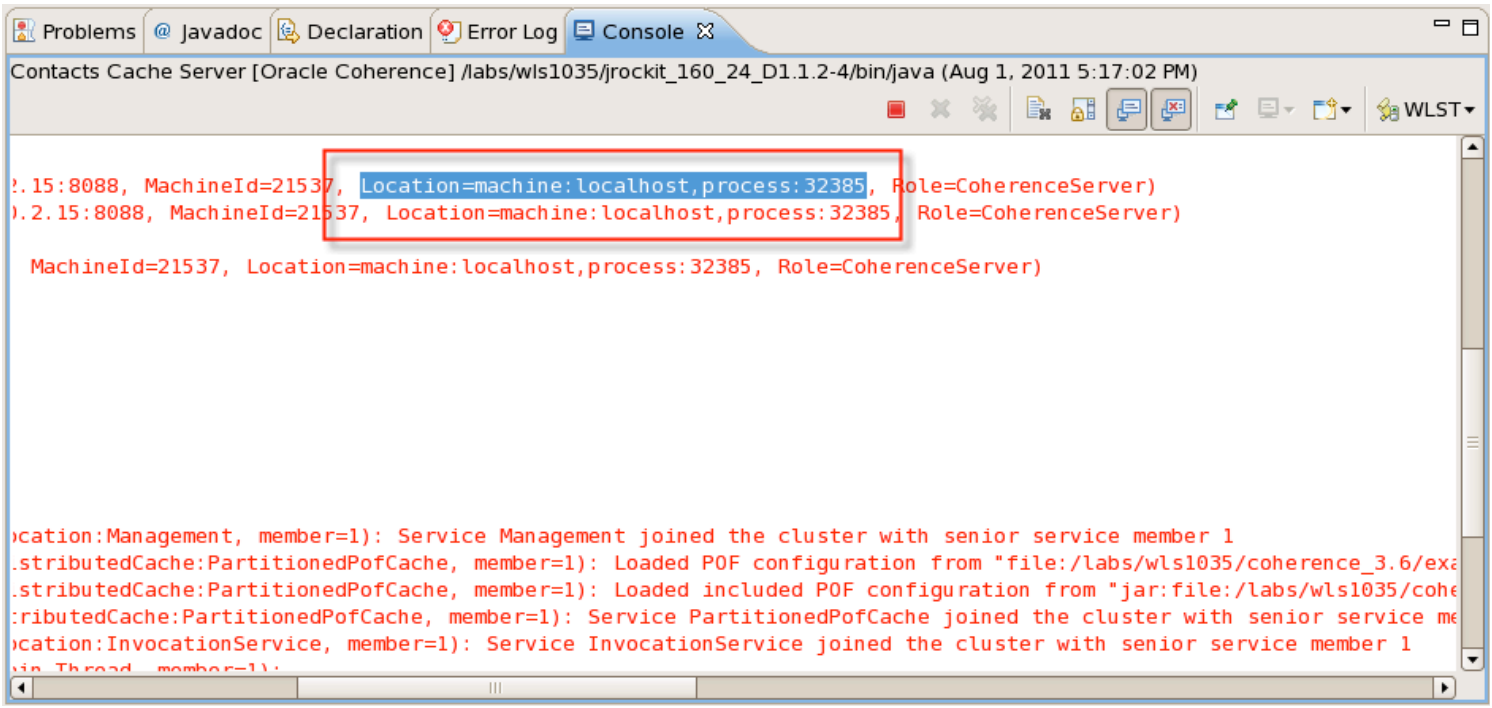


Then stop the process by clicking the red **stop** button.

Now in the src/resource.config/tangosol-coherence-override.xml file, switch to **Design** view, click on the **Cluster** accordion, and in the **Machine name** field enter **localhost**.



Go to the main menu in OEPE and choose **Run->Run Configurations…** and select the **Contacts Cache Server** again and click **Run**. Scroll through the console output until you see the “Location=” value, and notice how it reflects the localhost value.



Congratulations, all done!

You can learn more online at

http://coherence.oracle.com/display/EXAMPLES/Home