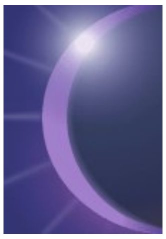
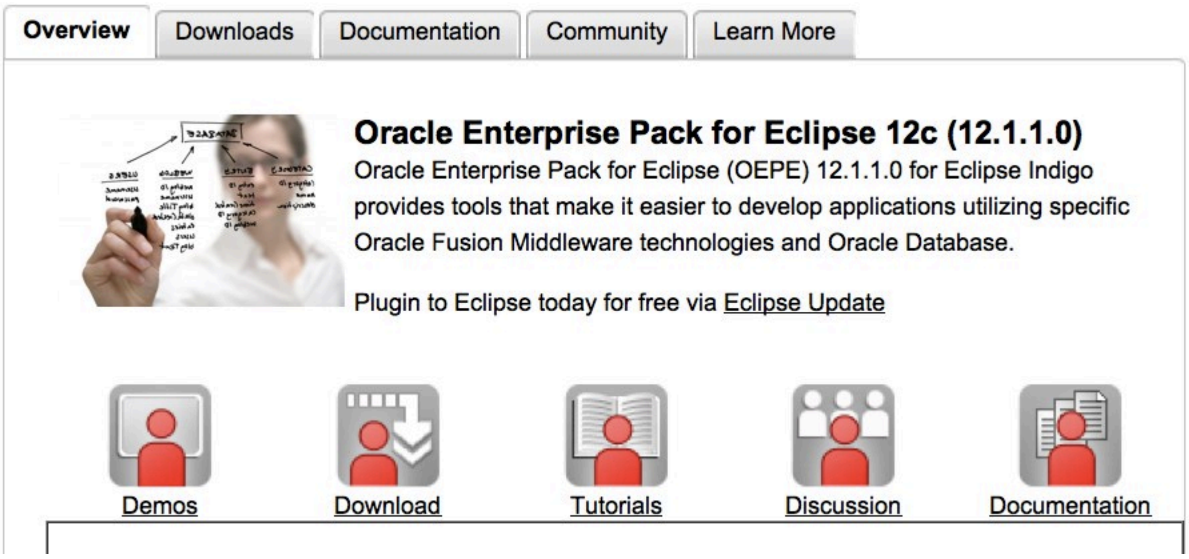
WLS12c and the Oracle Enterprise Pack for Eclipse



The Oracle Enterprise Pack for Eclipse extends the IDE to tailor it to the use with WLS. It contains a number of editors and features which makes the interworking smoothly. We will have a look at the OEPE and how it plays together with WLS12c on an OSX installation.
WLS ships with ready to use and configured examples. We will look at one example that demonstrates JavaEE6 features. We want to use Eclipse not only to analyse the examples, but also demonstrate a setup as an Eclipse project to go through the full edit – compile –install- test cycle. This exemplifies the usage of the WLS examples as mini projects, which can be used in situation where we want to run simple tests or in prototyping situations.

1. Installation of OEPE

A good starting point to look at the OEPE is Oracles web page [1], which gives an overview and provides links to tutorials, downloads, etc.



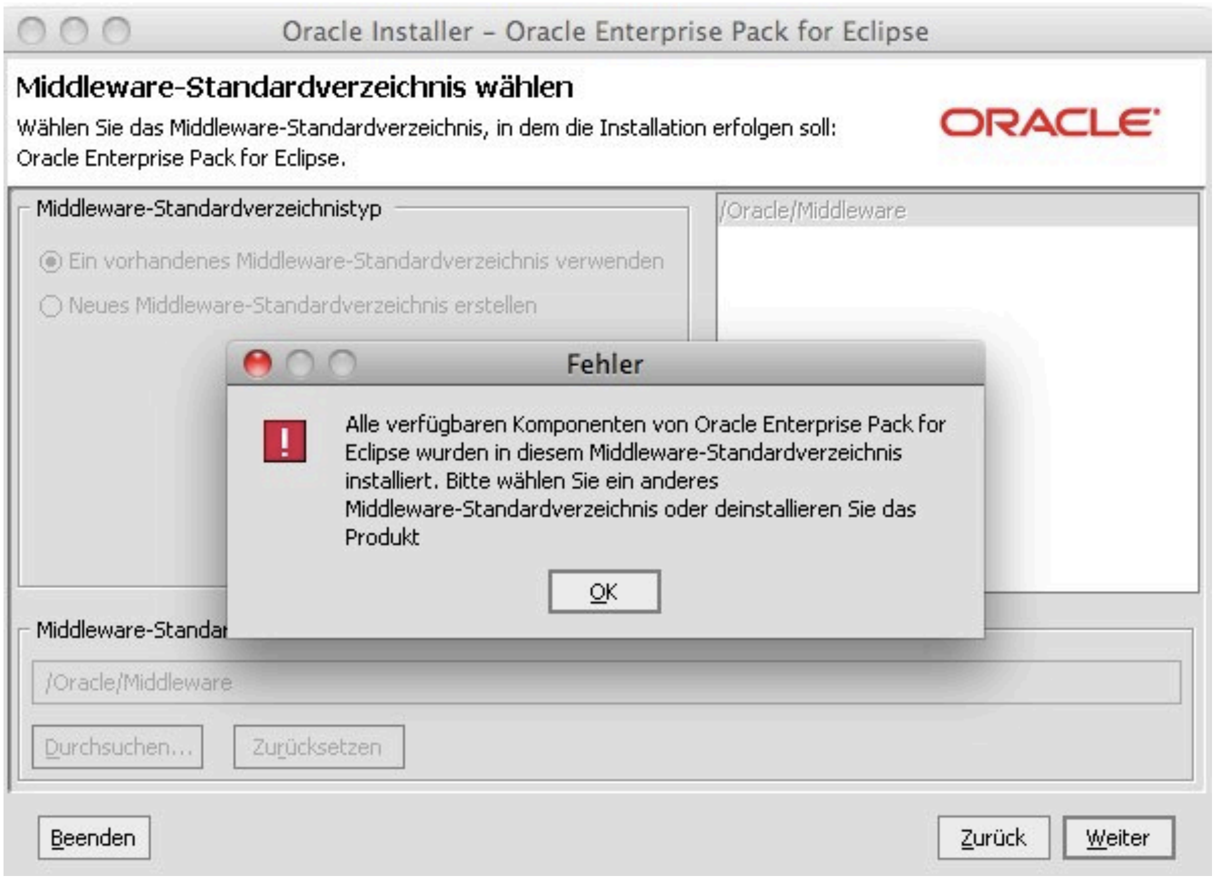
Oracle Enterprise Pack for Eclipse 12c (12.1.1.0)
Oracle Enterprise Pack for Eclipse (OEPE) 12.1.1.0 for Eclipse Indigo provides tools that make it easier to develop applications utilizing specific Oracle Fusion Middleware technologies and Oracle Database.

Plugin to Eclipse today for free via Eclipse Update

Figure 1. Home of the OEPE on the Oracle Website.

There are several installation options for the OEPE which are described in the installation guide .[2] We use the Oracle Installer which is available as a 1,4 GByte jar file: oepe-indigo-installer-12.1.1.0.0.201112072225-12.1.1-macosx-cocoa-x86_64.jar. After choosing the Middleware Home of my WLS12c installation, the installer informs me that it is already installed.

[1] OEPE Home Page: http://www.oracle.com/technetwork/developer-tools/eclipse/overview/index.html
[2] Installation Guide
http://docs.oracle.com/cd/E27086_01/help/oracle.eclipse.tools.common.doc/html/install.html



Oracle Installer - Oracle Enterprise Pack for Eclipse

Middleware-Standardverzeichnis wählen

Wählen Sie das Middleware-Standardverzeichnis, in dem die Installation erfolgen soll: Oracle Enterprise Pack for Eclipse.

Middleware-Standardverzeichnistyp

(● Ein vorhandenes Middleware-Standardverzeichnis verwenden)

Neues Middleware-Standardverzeichnis erstellen

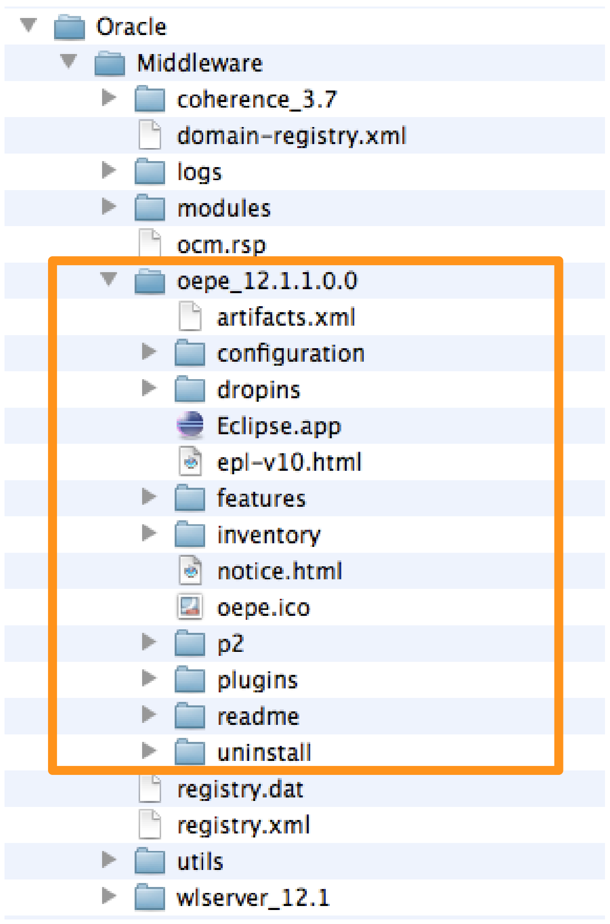
/Oracle/Middleware

Fehler

Alle verfügbaren Komponenten von Oracle Enterprise Pack for Eclipse wurden in diesem Middleware-Standardverzeichnis installiert. Bitte wählen Sie ein anderes Middleware-Standardverzeichnis oder deinstallieren Sie das Produkt

Figure 2. The Oracle installer says that the OEPE is already installed.

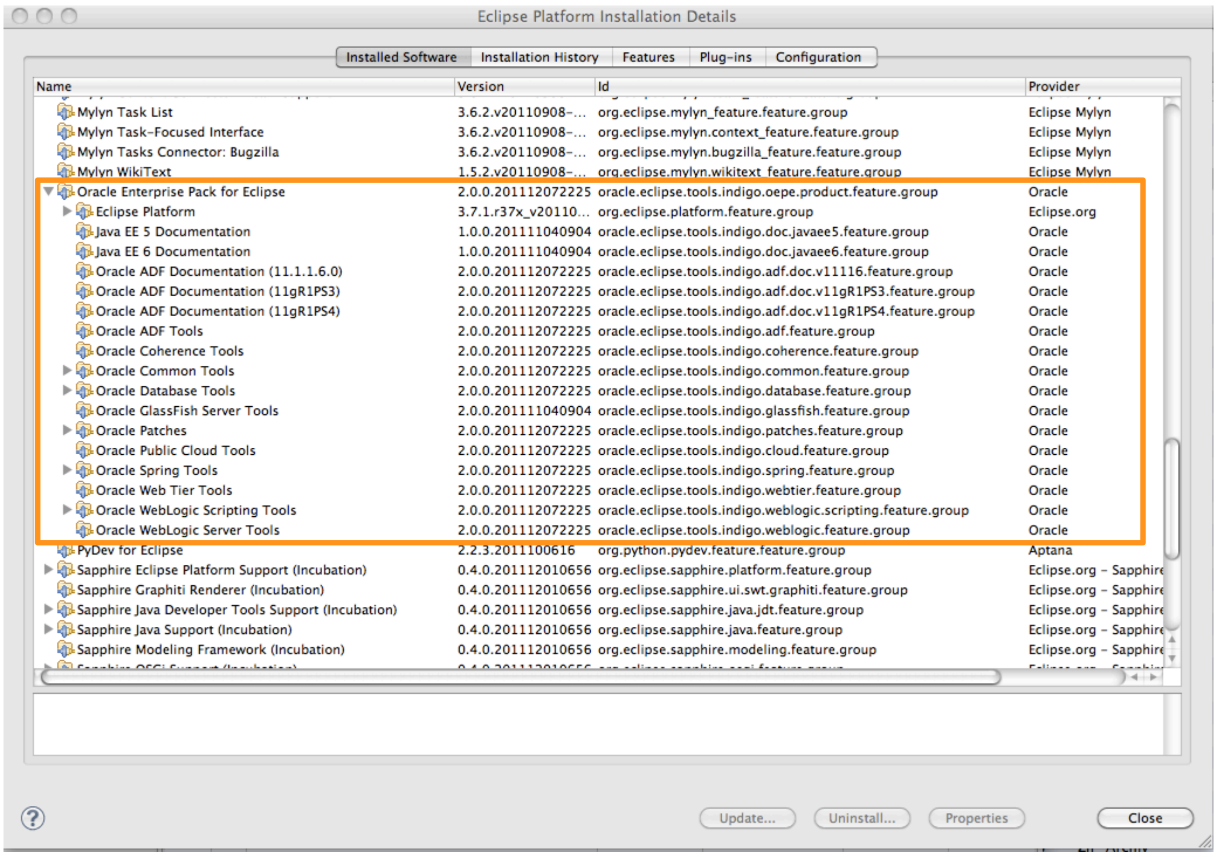
A quick look at the finder reveals that it is already there. I originally installed WLS12c from the package oepe-indigo-installer-12.1.1.0.0.201112072225-12.1.1-macosx-cocoa-x86_64.jar which in fact is the same file. The eclipse version installed is the Indigo release, version 3.7.1.



```
Oracle
  Middleware
    coherence_3.7
    domain-registry.xml
    logs
    modules
    ocm.rsp
    oepe_12.1.1.0.0
      artifacts.xml
      configuration
      dropins
      Eclipse.app
      epl-v10.html
      features
      inventory
      notice.html
      oepe.ico
      p2
      plugins
      readme
      uninstall
    registry.dat
    registry.xml
    utils
    wlserver_12.1
```

Figure 3. The Mac Finder showing the Eclipse installation.

We start eclipse by double clicking on the Eclipse.app icon. We have to chose a workspace directory, e.g. /Eclipse/Indigo/workspace. In the installation details window we can see the modules of the OEPE.

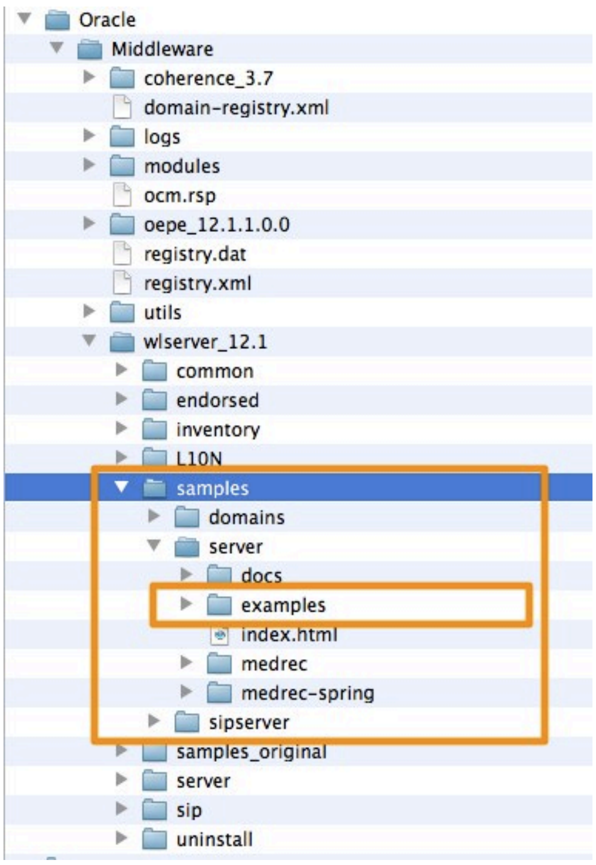

Eclipse Platform Installation Details

Name	Version	Id	Provider
Mylyn Task List	3.6.2.v20110908-...	org.eclipse.mylyn_feature.feature.group	Eclipse Mylyn
Mylyn Task-Focused Interface	3.6.2.v20110908-...	org.eclipse.mylyn.context_feature.feature.group	Eclipse Mylyn
Mylyn Tasks Connector: Bugzilla	3.6.2.v20110908-...	org.eclipse.mylyn.bugzilla_feature.feature.group	Eclipse Mylyn
Mylyn WikiText	1.5.2.v20110908-...	org.eclipse.mylyn.wikitext_feature.feature.group	Eclipse Mylyn
Oracle Enterprise Pack for Eclipse	2.0.0.201112072225	oracle.eclipse.tools.indigo.oepe.product.feature.group	Oracle
Eclipse Platform	3.7.1.r37x_v20110...	org.eclipse.platform.feature.group	Eclipse.org
Java EE 5 Documentation	1.0.0.201111040904	oracle.eclipse.tools.indigo.doc.javaee5.feature.group	Oracle
Java EE 6 Documentation	1.0.0.201111040904	oracle.eclipse.tools.indigo.doc.javaee6.feature.group	Oracle
Oracle ADF Documentation (11.1.1.6.0)	2.0.0.201112072225	oracle.eclipse.tools.indigo.adf.doc.v11116.feature.group	Oracle
Oracle ADF Documentation (11gR1PS3)	2.0.0.201112072225	oracle.eclipse.tools.indigo.adf.doc.v11gR1PS3.feature.group	Oracle
Oracle ADF Documentation (11gR1PS4)	2.0.0.201112072225	oracle.eclipse.tools.indigo.adf.doc.v11gR1PS4.feature.group	Oracle
Oracle ADF Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.adf.feature.group	Oracle
Oracle Coherence Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.coherence.feature.group	Oracle
Oracle Common Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.common.feature.group	Oracle
Oracle Database Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.database.feature.group	Oracle
Oracle GlassFish Server Tools	2.0.0.201111040904	oracle.eclipse.tools.indigo.glassfish.feature.group	Oracle
Oracle Patches	2.0.0.201112072225	oracle.eclipse.tools.indigo.patches.feature.group	Oracle
Oracle Public Cloud Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.cloud.feature.group	Oracle
Oracle Spring Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.spring.feature.group	Oracle
Oracle Web Tier Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.webtier.feature.group	Oracle
Oracle WebLogic Scripting Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.weblogic.scripting.feature.group	Oracle
Oracle WebLogic Server Tools	2.0.0.201112072225	oracle.eclipse.tools.indigo.weblogic.feature.group	Oracle
PyDev for Eclipse	2.2.3.2011100616	org.python.pydev.feature.feature.group	Aptana
Sapphire Eclipse Platform Support (Incubation)	0.4.0.201112010656	org.eclipse.sapphire.platform.feature.group	Eclipse.org - Sapphire
Sapphire Graphiti Renderer (Incubation)	0.4.0.201112010656	org.eclipse.sapphire.ui.swt.graphiti.feature.group	Eclipse.org - Sapphire
Sapphire Java Developer Tools Support (Incubation)	0.4.0.201112010656	org.eclipse.sapphire.java.jdt.feature.group	Eclipse.org - Sapphire
Sapphire Java Support (Incubation)	0.4.0.201112010656	org.eclipse.sapphire.java.feature.group	Eclipse.org - Sapphire
Sapphire Modeling Framework (Incubation)	0.4.0.201112010656	org.eclipse.sapphire.modeling.feature.group	Eclipse.org - Sapphire

Figure 4. Installed OEPE modules in the Eclipse Indigo release.

1.1. Integrating the WLS API Examples

WLS12c ships with a number of API examples that are already built and installed in the example server domain "wl_server". They are delivered with a complete ant based built environment which supports the full edit-built-deploy-test lifecycle on the command line.



```
Oracle
  Middleware
    coherence_3.7
    domain-registry.xml
    logs
    modules
    ocm.rsp
    oepe_12.1.1.0.0
    registry.dat
    registry.xml
    utils
    wlserver_12.1
      common
      endorsed
      inventory
      L10N
      samples
        domains
        server
          docs
          examples
          index.html
          medrec
          medrec-spring
        sipserver
      samples_original
      server
      sip
      uninstall
```

Figure 5. OSX Finder showing the WLS API examples in the samples directory.

We want to integrate these examples into eclipse to better analyse, test and change them. Furthermore we want to use the eclipse ant tool to built, deploy and run the examples directly from within the IDE.

We set up a new project, import these examples, change some properties and include additional ant tasks:

We start the new project dialog. Since we want to make use of the integrated java documentation and code expansion in java source files we need to choose a Java project nature. Otherwise the project will not contain a Java Builder and thus no code expansion will be available. Note that we do not intend to use the Java Builder to build the project. This will be done by the ant files that ship with the examples. We name the project wl_server after the domain.

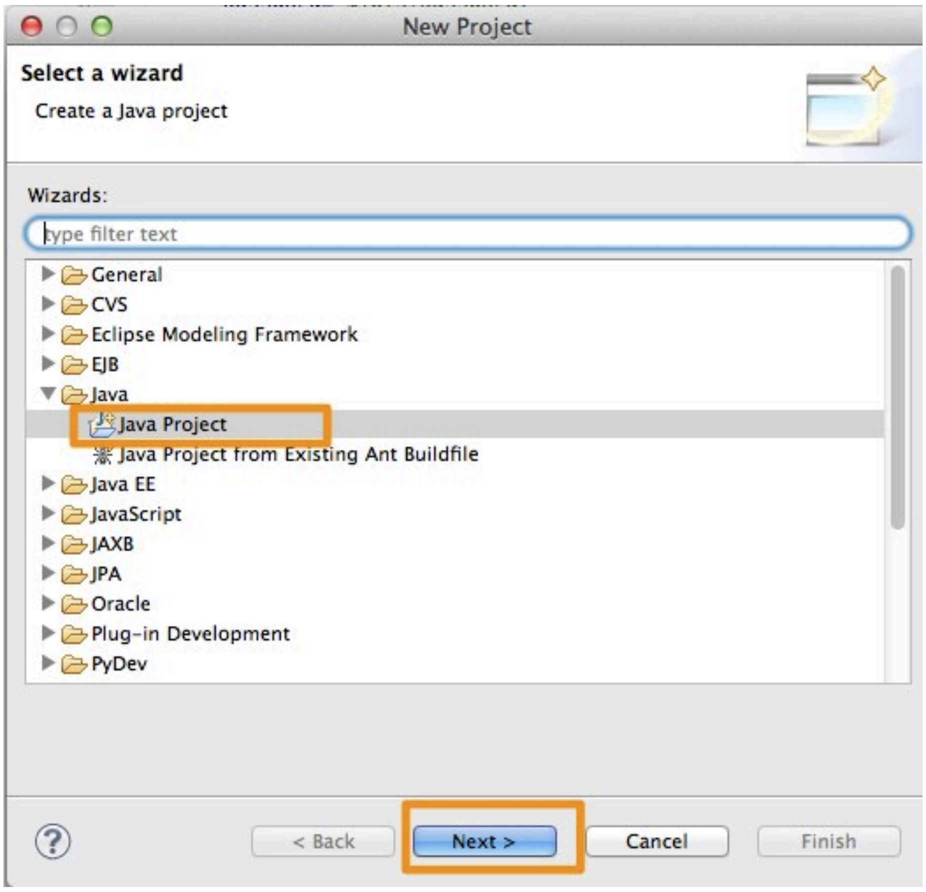
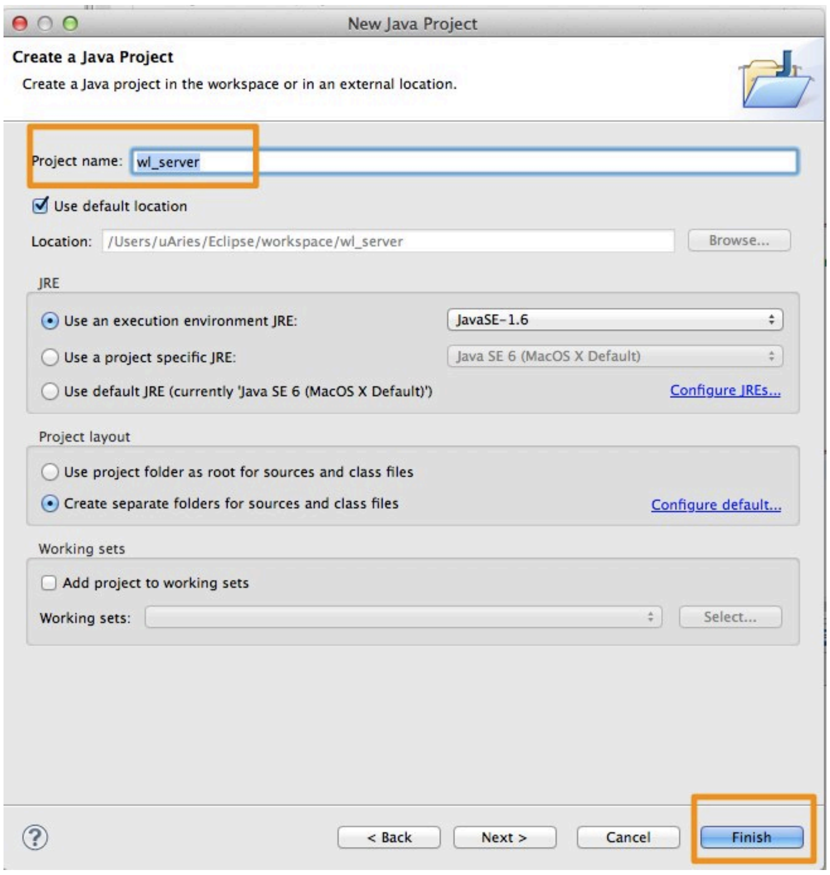


Figure 6. Choosing the Java Project Nature.

We proceed to the next step an give the project a name.


New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
< Back
Next >
Cancel
Finish

Project name: wl_server

[✓] Use default location

Location: /Users/uAries/Eclipse/workspace/wl_server Browse...

JRE

(●) Use an execution environment JRE: JavaSE-1.6

( ) Use a project specific JRE: Java SE 6 (MacOS X Default)

( ) Use default JRE (currently 'Java SE 6 (MacOS X Default)') Configure JREs...

Project layout

( ) Use project folder as root for sources and class files

(●) Create separate folders for sources and class files Configure default...

Working sets

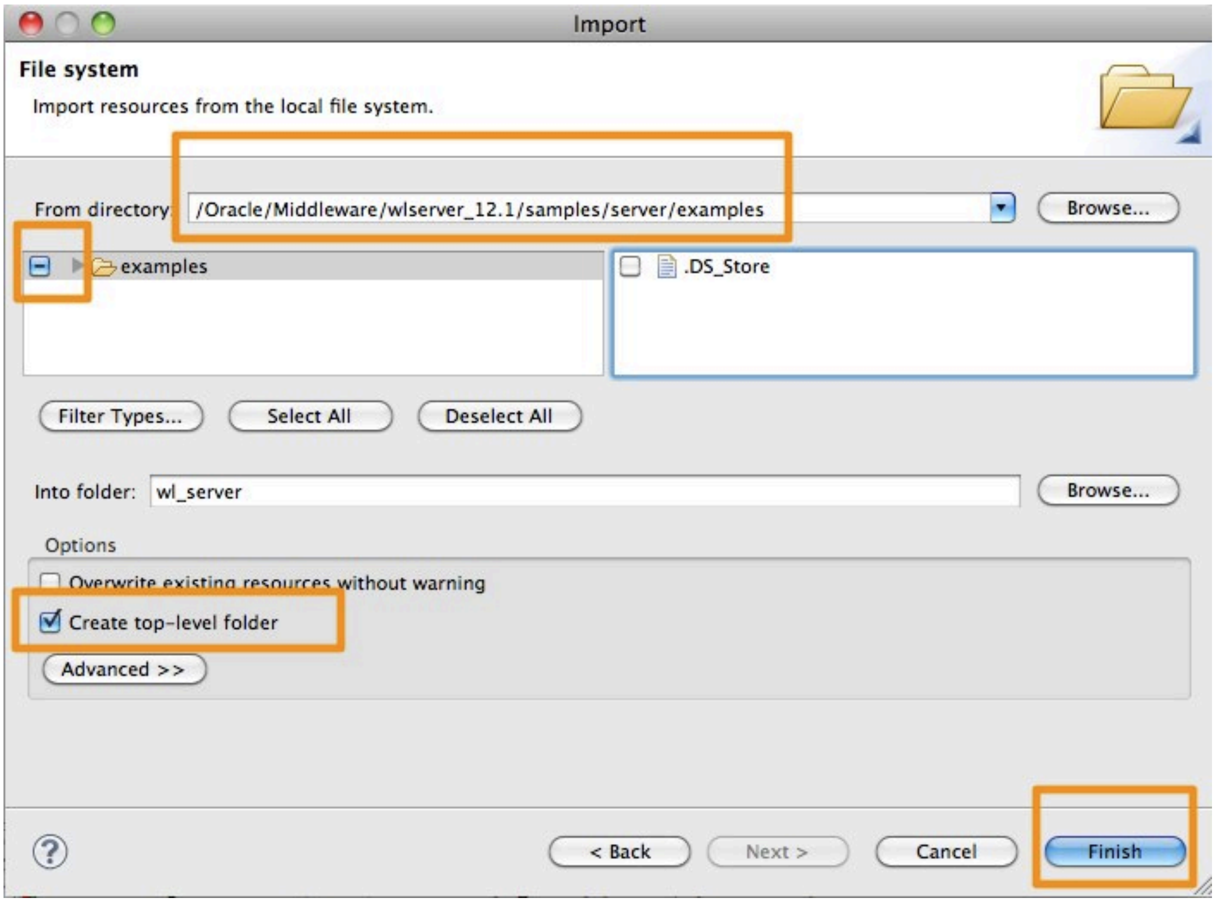
[ ] Add project to working sets

Working sets: Select...

Figure 7. Setting up the wl_server project with the new project dialog.

We finish the dialog in this step. All we can do all further settings later.

We right-click on "wl_server" in the project explorer and start the Import dialog. We choose "General->File System" as import source and browse to the examples directory.


Import

File system

Import resources from the local file system.

From directory: /Oracle/Middleware/wlserver_12.1/samples/server/examples

examples

.DS_Store

Into folder: wl_server

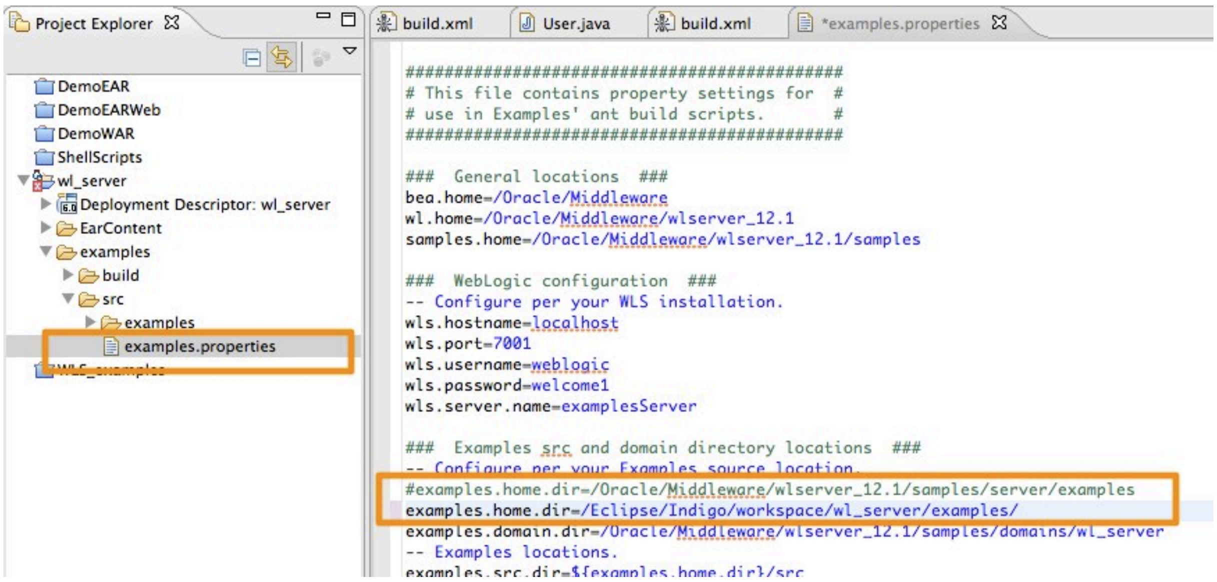
Options

Overwrite existing resources without warning

Create top-level folder [✓]

Figure 8. Choosing the examples folder in the Import dialog.

This copies all the example files into the eclipse workspace. Now we change the property example.home.dir to tell the ant files to work on your copy of the examples.



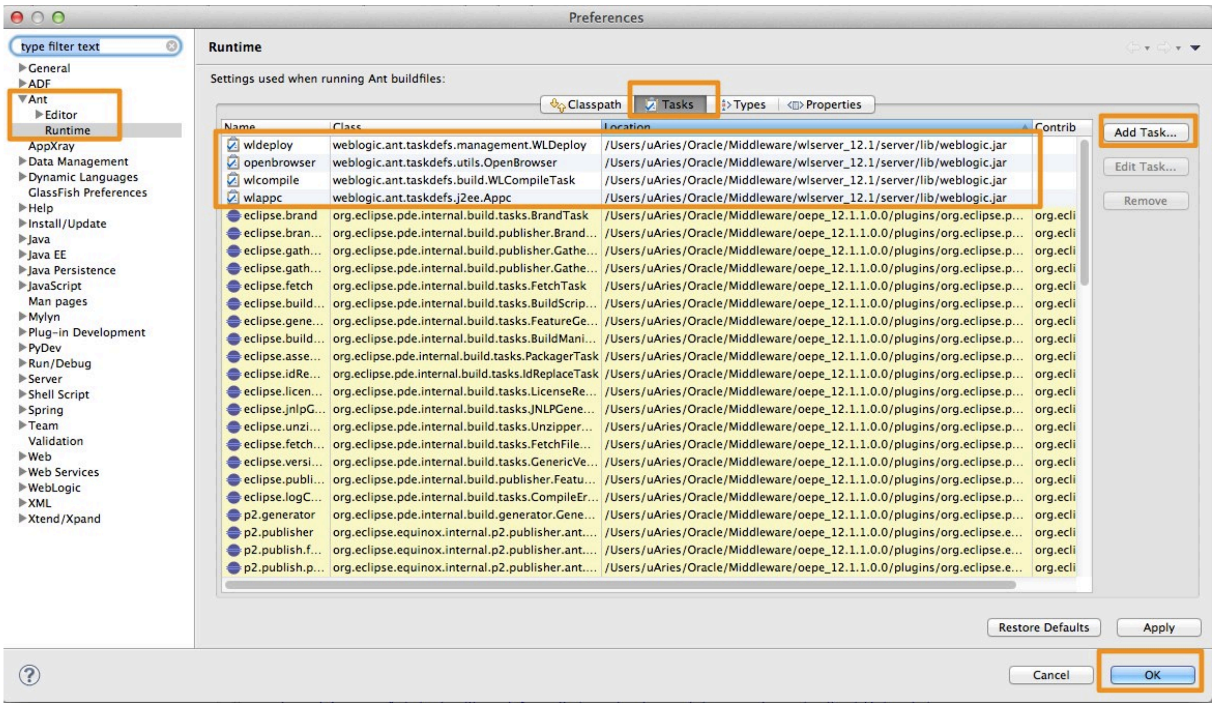
```
###############################################
# This file contains property settings for  #
# use in Examples' ant build scripts.       #
###############################################

###  General locations  ###
bea.home=/Oracle/Middleware
wl.home=/Oracle/Middleware/wlserver_12.1
samples.home=/Oracle/Middleware/wlserver_12.1/samples

###  WebLogic configuration  ###
-- Configure per your WLS installation.
wls.hostname=localhost
wls.port=7001
wls.username=weblogic
wls.password=welcome1
wls.server.name=examplesServer

###  Examples src and domain directory locations  ###
-- Configure per your Examples source location.
#examples.home.dir=/Oracle/Middleware/wlserver_12.1/samples/server/examples
examples.home.dir=/Eclipse/Indigo/workspace/wl_server/examples/
examples.domain.dir=/Oracle/Middleware/wlserver_12.1/samples/domains/wl_server
-- Examples locations.
examples.src.dir=${examples.home.dir}/src
```

Figure 9. Changing examples.home.dir in the example.properties file.

The build.xml files make use of the ant tasks wldeploy and openbrowser which are included in the weblogic.jar. We need to add them to the eclipse ant configuration.

In the eclipse Preferences pane we go to Ant->Runtime->Classpath. In "Global Entries" we add the external jar file: /Oracle/Middleware/wlserver_12.1/server/lib/weblogic.jar

In the "Task" pane we add the tasks wldeploy and openbrowser, as indicated in the following figure. We choose the weblogic.jar file as locationin the "Add Task …" dialog.

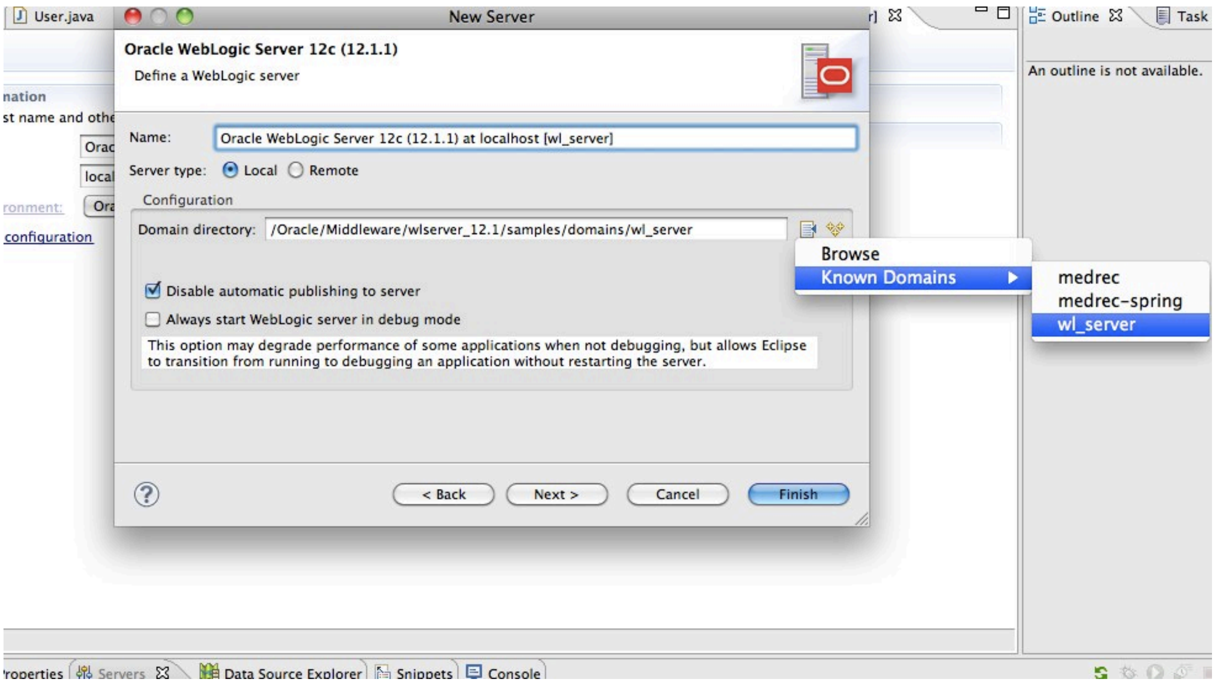


Settings used when running Ant buildfiles:

Name	Class	Location	Contrib
wldeploy	weblogic.ant.taskdefs.management.WLDeploy	/Users/uAries/Oracle/Middleware/wlserver_12.1/server/lib/weblogic.jar	
openbrowser	weblogic.ant.taskdefs.utils.OpenBrowser	/Users/uAries/Oracle/Middleware/wlserver_12.1/server/lib/weblogic.jar	
wlcompile	weblogic.ant.taskdefs.build.WLCompileTask	/Users/uAries/Oracle/Middleware/wlserver_12.1/server/lib/weblogic.jar	
wlappc	weblogic.ant.taskdefs.j2ee.Appc	/Users/uAries/Oracle/Middleware/wlserver_12.1/server/lib/weblogic.jar	
eclipse.brand	org.eclipse.pde.internal.build.tasks.BrandTask	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.bran...	org.eclipse.pde.internal.build.publisher.Brand...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.gath...	org.eclipse.pde.internal.build.publisher.Gathe...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.gath...	org.eclipse.pde.internal.build.publisher.Gathe...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.fetch	org.eclipse.pde.internal.build.tasks.FetchTask	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.build...	org.eclipse.pde.internal.build.tasks.BuildScrip...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.gene...	org.eclipse.pde.internal.build.tasks.FeatureGe...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.build...	org.eclipse.pde.internal.build.tasks.BuildMani...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.asse...	org.eclipse.pde.internal.build.tasks.PackagerTask	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.idRe...	org.eclipse.pde.internal.build.tasks.IdReplaceTask	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.licen...	org.eclipse.pde.internal.build.tasks.LicenseRe...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.jnlpG...	org.eclipse.pde.internal.build.tasks.JNLPGene...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.unzi...	org.eclipse.pde.internal.build.tasks.Unzipper...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.fetch...	org.eclipse.pde.internal.build.tasks.FetchFile...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.versi...	org.eclipse.pde.internal.build.tasks.GenericVe...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.publi...	org.eclipse.pde.internal.build.publisher.Featu...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
eclipse.logC...	org.eclipse.pde.internal.build.tasks.CompileEr...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
p2.generator	org.eclipse.pde.internal.build.generator.Gene...	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.p...	org.ecli
p2.publisher	org.eclipse.equinox.internal.p2.publisher.ant....	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.e...	org.ecli
p2.publish.f...	org.eclipse.equinox.internal.p2.publisher.ant....	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.e...	org.ecli
p2.publish.p...	org.eclipse.equinox.internal.p2.publisher.ant....	/Users/uAries/Oracle/Middleware/oepe_12.1.1.0.0/plugins/org.eclipse.e...	org.ecli

Figure 10. Adding additional ant tasks in eclipse.

We want to add a server configuration to start the wl_server from within eclipse. From the "Java EE" perspective we right-click into the server tab at the bottom of the workplace and select the "New->Server" dialog. We chose Oracle WebLogic Server 12c (12.1.1) as runtime environment and in the next step we select wl_server from the "Known Domains".


New Server

Oracle WebLogic Server 12c (12.1.1)

Define a WebLogic server

Name: Oracle WebLogic Server 12c (12.1.1) at localhost [wl_server]

Server type: [✓ Local] Remote

Configuration

Domain directory: /Oracle/Middleware/wlserver_12.1/samples/domains/wl_server

[✓] Disable automatic publishing to server

[ ] Always start WebLogic server in debug mode

This option may degrade performance of some applications when not debugging, but allows Eclipse to transition from running to debugging an application without restarting the server.

Figure 11. **Setting up a server configuration in eclipse.**

Note that we select the server type local. For a remote server we would have to apply remote host and port as well as admin user and password. In the next step, we do not configure any resources, since these are available eclipse projects and we don't want to have them installed. Instead we want to use the ant deployment targets.

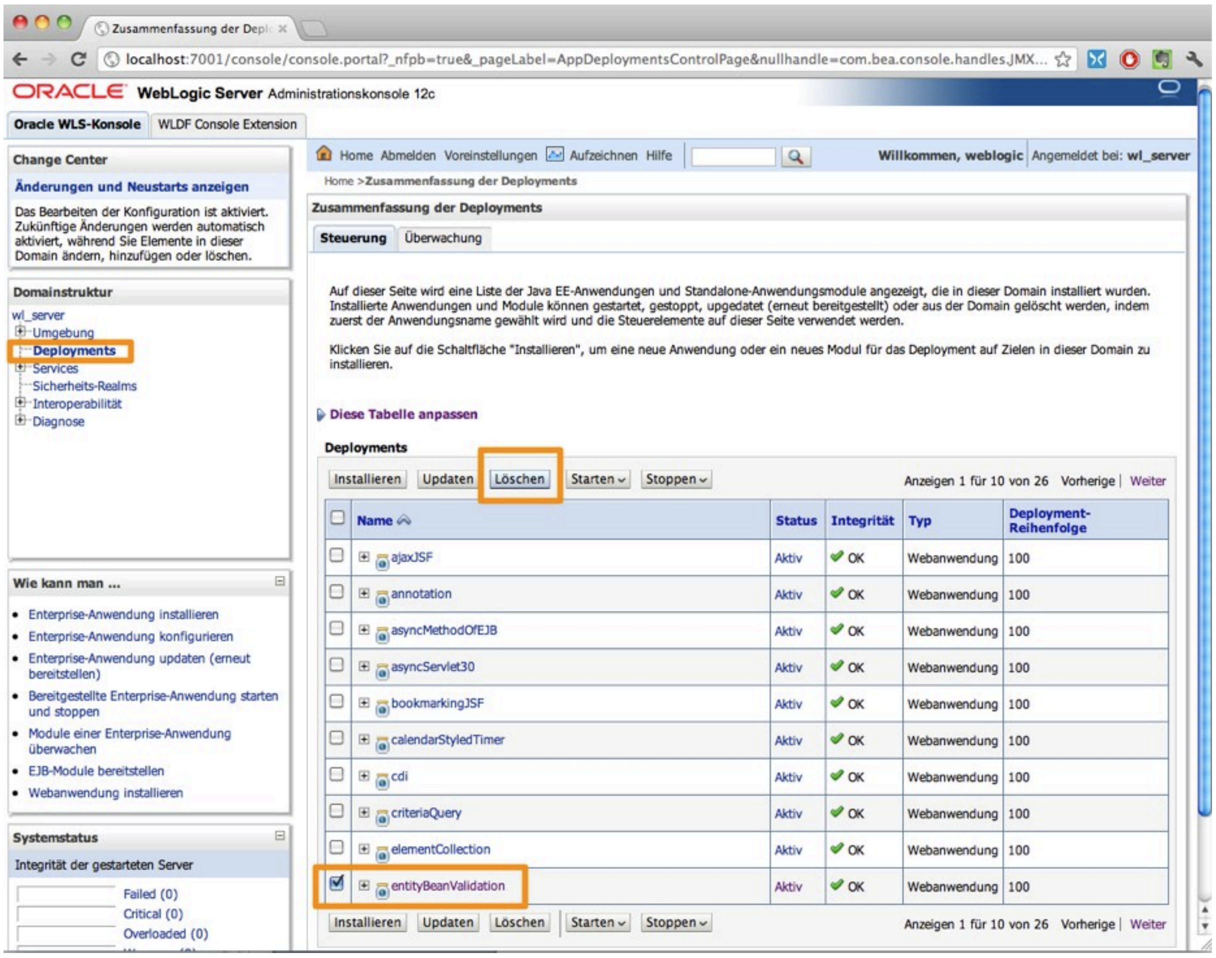
We start the wl_server from within the IDE and check the deployment in the administration console. The URL is http://localhost:7001/console and user and password are weblogic/welcome1.

Under deployments we can see all deployment in the wl_server, which are the deployments of the sample server installation. Currently there are 26 depoyments, mostly web applications.

1.2. Testing the API examples

Let's focus our attention to the entityBeanValidation example. This is a small API example, demonstrating the bean validation feature of the Java EE6 release. We want to undeploy this application via the administration console and newly deploy it from within eclipse. Then we will run the example to test it.

In the administration console we locate the example and press delete to uninstall it.



Zusammenfassung der Depl

localhost:7001/console/console.portal?_nfpb=true&_pageLabel=AppDeploymentsControlPage&nullhandle=com.bea.console.handles.JMX...

ORACLE WebLogic Server Administrationskonsole 12c

Oracle WLS-Konsole | WLDF Console Extension

Change Center

Änderungen und Neustarts anzeigen

Das Bearbeiten der Konfiguration ist aktiviert. Zukünftige Änderungen werden automatisch aktiviert, während Sie Elemente in dieser Domain ändern, hinzufügen oder löschen.

Domainstruktur

wl_server
- Umgebung
- Deployments
- Services
- Sicherheits-Realms
- Interoperabilität
- Diagnose

Wie kann man ...

- Enterprise-Anwendung installieren
- Enterprise-Anwendung konfigurieren
- Enterprise-Anwendung updaten (erneut bereitstellen)
- Bereitgestellte Enterprise-Anwendung starten und stoppen
- Module einer Enterprise-Anwendung überwachen
- EJB-Module bereitstellen
- Webanwendung installieren

Systemstatus

Integrität der gestarteten Server

Failed (0)
Critical (0)
Overloaded (0)

Home Abmelden Voreinstellungen Aufzeichnen Hilfe | Willkommen, weblogic | Angemeldet bei: wl_server

Home >Zusammenfassung der Deployments

Zusammenfassung der Deployments

Steuerung | Überwachung

Auf dieser Seite wird eine Liste der Java EE-Anwendungen und Standalone-Anwendungsmodule angezeigt, die in dieser Domain installiert wurden. Installierte Anwendungen und Module können gestartet, gestoppt, updatet (erneut bereitgestellt) oder aus der Domain gelöscht werden, indem zuerst der Anwendungsname gewählt wird und die Steuerelemente auf dieser Seite verwendet werden.

Klicken Sie auf die Schaltfläche "Installieren", um eine neue Anwendung oder ein neues Modul für das Deployment auf Zielen in dieser Domain zu installieren.

Diese Tabelle anpassen

Deployments

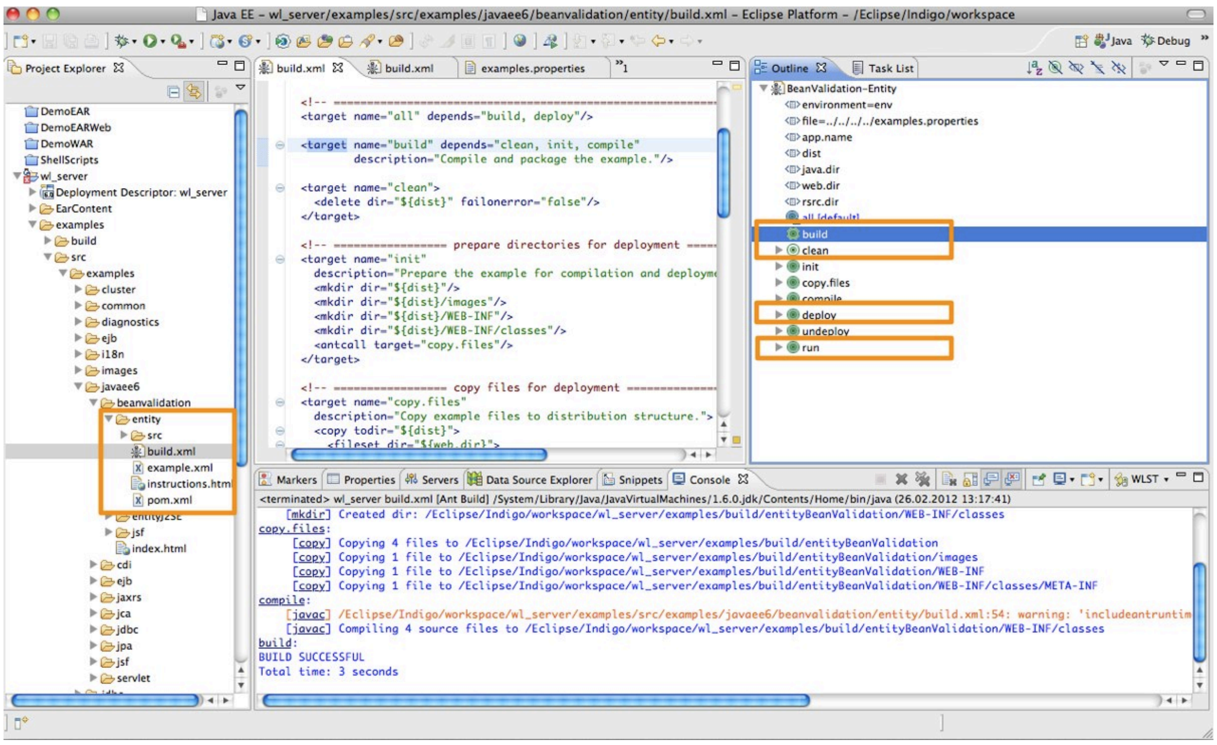
Installieren | Updaten | Löschen | Starten | Stoppen — Anzeigen 1 für 10 von 26 Vorherige | Weiter

	Name	Status	Integrität	Typ	Deployment-Reihenfolge
☐	ajaxJSF	Aktiv	✓ OK	Webanwendung	100
☐	annotation	Aktiv	✓ OK	Webanwendung	100
☐	asyncMethodOfEJB	Aktiv	✓ OK	Webanwendung	100
☐	asyncServlet30	Aktiv	✓ OK	Webanwendung	100
☐	bookmarkingJSF	Aktiv	✓ OK	Webanwendung	100
☐	calendarStyledTimer	Aktiv	✓ OK	Webanwendung	100
☐	cdi	Aktiv	✓ OK	Webanwendung	100
☐	criteriaQuery	Aktiv	✓ OK	Webanwendung	100
☐	elementCollection	Aktiv	✓ OK	Webanwendung	100
☑	entityBeanValidation	Aktiv	✓ OK	Webanwendung	100

Installieren | Updaten | Löschen | Starten | Stoppen — Anzeigen 1 für 10 von 26 Vorherige | Weiter

Figure 12. **Uninstalling the entityBeanValidation application from the WLS console.**

This works fine, no restart is required. Now we go to the beanvalidation folder in eclipse open the corresponding ant build.xml file and clean, build and deploy the example. Eclipse builds the example under the eclipse workspace directory, as it should. During deployment, it advertises this directory to WLS and the wl_server deploys it directly from there, in contrast to the original deployments, which were taken from the WLS directory.



```
<!-- ================================================================
<target name="all" depends="build, deploy"/>

<target name="build" depends="clean, init, compile"
        description="Compile and package the example."/>

<target name="clean">
  <delete dir="${dist}" failonerror="false"/>
</target>

<!-- ================= prepare directories for deployment =====
<target name="init"
  description="Prepare the example for compilation and deployme
  <mkdir dir="${dist}"/>
  <mkdir dir="${dist}/images"/>
  <mkdir dir="${dist}/WEB-INF"/>
  <mkdir dir="${dist}/WEB-INF/classes"/>
  <antcall target="copy.files"/>
</target>

<!-- ================= copy files for deployment ==============
<target name="copy.files"
  description="Copy example files to distribution structure.">
  <copy todir="${dist}">
    <fileset dir="${web.dir}">
```

```
<terminated> wl_server build.xml [Ant Build] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (26.02.2012 13:17:41)
    [mkdir] Created dir: /Eclipse/Indigo/workspace/wl_server/examples/build/entityBeanValidation/WEB-INF/classes
copy.files:
     [copy] Copying 4 files to /Eclipse/Indigo/workspace/wl_server/examples/build/entityBeanValidation
     [copy] Copying 1 file to /Eclipse/Indigo/workspace/wl_server/examples/build/entityBeanValidation/images
     [copy] Copying 1 file to /Eclipse/Indigo/workspace/wl_server/examples/build/entityBeanValidation/WEB-INF
     [copy] Copying 1 file to /Eclipse/Indigo/workspace/wl_server/examples/build/entityBeanValidation/WEB-INF/classes/META-INF
compile:
    [javac] /Eclipse/Indigo/workspace/wl_server/examples/src/examples/javaee6/beanvalidation/entity/build.xml:54: warning: 'includeantruntim
    [javac] Compiling 4 source files to /Eclipse/Indigo/workspace/wl_server/examples/build/entityBeanValidation/WEB-INF/classes
build:
BUILD SUCCESSFUL
Total time: 3 seconds
```

Figure 13. Building, and deploying the entitybeanvalidation example from within eclipse.

A quick check in the WLS administration console shows that the application is properly installed.

The ant run target should just open a browser and point to the location of the example at http://localhost:7001/entityBeanValidation but the openbrowser ant task which is used here, produces an error:

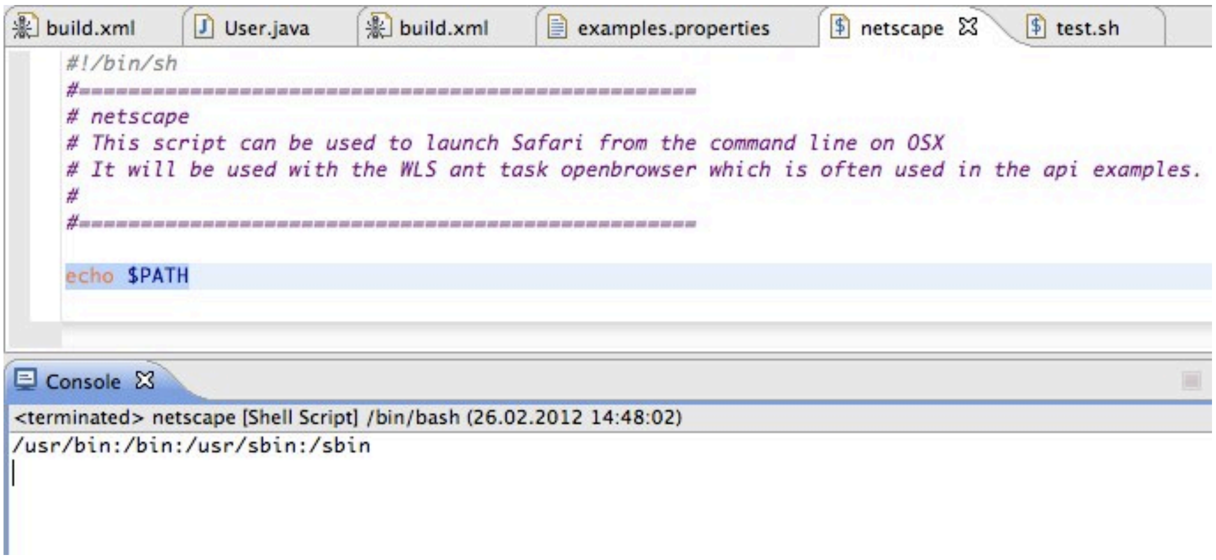
```
BUILD FAILED
/Eclipse/Indigo/workspace/wl_server/examples/src/examples/javaee6/beanvalidation/entity/build.
xml:86: Could not invoke browser, command=netscape -remote
openURL(http://localhost:7001/entityBeanValidation)'.  Windows: Please make sure that default
browser can open.  Unix: Please make sure that 'netscape' can open (use unixBrowser attribute
to change browser on Unix, ie unixBrowser="mozilla").
java.io.IOException: Cannot run program "netscape": error=2, No such file or directory
```

The command does not work on OSX, even after installing netscape and playing around with the unixBrowser variable I could not get it to work. As a workaround we program a small shell script named “netscape”, which accepts an URL as an argument and starts Safari. The ShellEd plug-in is very useful, see also the installation instructions at Chaper 1.4.

The program is very simple and looks like this.

```
#!/bin/sh
#===================================================
# netscape
# This script can be used to launch Safari from the command line on OSX
# It will be used with the WLS ant task openbrowser which is often used in the api examples.
#===================================================
#echo $PATH
open -a Safari $2
```

We echo the $PATH variable to check which location eclipse can use for looking up external programs.



```
#!/bin/sh
#==================================================
# netscape
# This script can be used to launch Safari from the command line on OSX
# It will be used with the WLS ant task openbrowser which is often used in the api examples.
#
#==================================================

echo $PATH
```

```
<terminated> netscape [Shell Script] /bin/bash (26.02.2012 14:48:02)
/usr/bin:/bin:/usr/sbin:/sbin
```

Figure 14. **Script echoing the PATH variable available to eclipse.**

While it is certainly no good practice to install our little script there we do it now for the purpose of this example.

```
macbook-pro:/>sudo su
Password:
macbook-pro:/>cp /Eclipse/Indigo/workspace/ShellScripts/netscape /usr/bin/
```

Another alternative would be to set the PATH variable in a Terminal and start eclipse from that terminal.

```
macbook-pro:/Oracle/Middleware/oepe_12.1.1.0.0/Eclipse.app/Contents/MacOS>export
PATH=$PATH:/Eclipse/Indigo/workspace/ShellScripts
macbook-pro:/Oracle/Middleware/oepe_12.1.1.0.0/Eclipse.app/Contents/MacOS>./eclipse
```

Now the ant target run opens Safari with the correct URL.

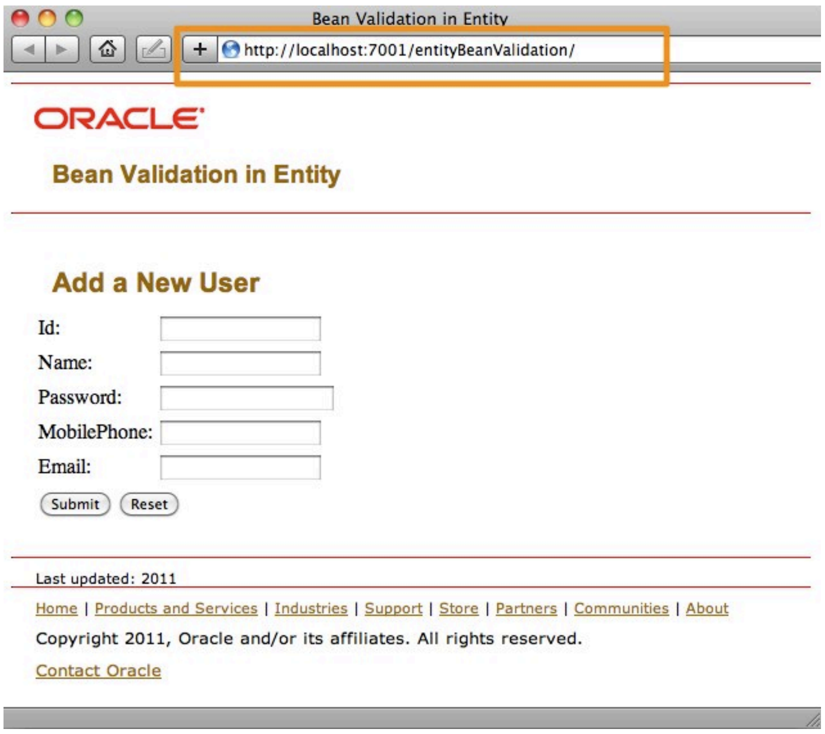

Bean Validation in Entity
http://localhost:7001/entityBeanValidation/
ORACLE
Bean Validation in Entity
Add a New User
Id:
Name:
Password:
MobilePhone:
Email:
Submit
Reset
Last updated: 2011
Home | Products and Services | Industries | Support | Store | Partners | Communities | About
Copyright 2011, Oracle and/or its affiliates. All rights reserved.
Contact Oracle

Figure 15. The bean validation example in the Safari browser.

1.3. Notes

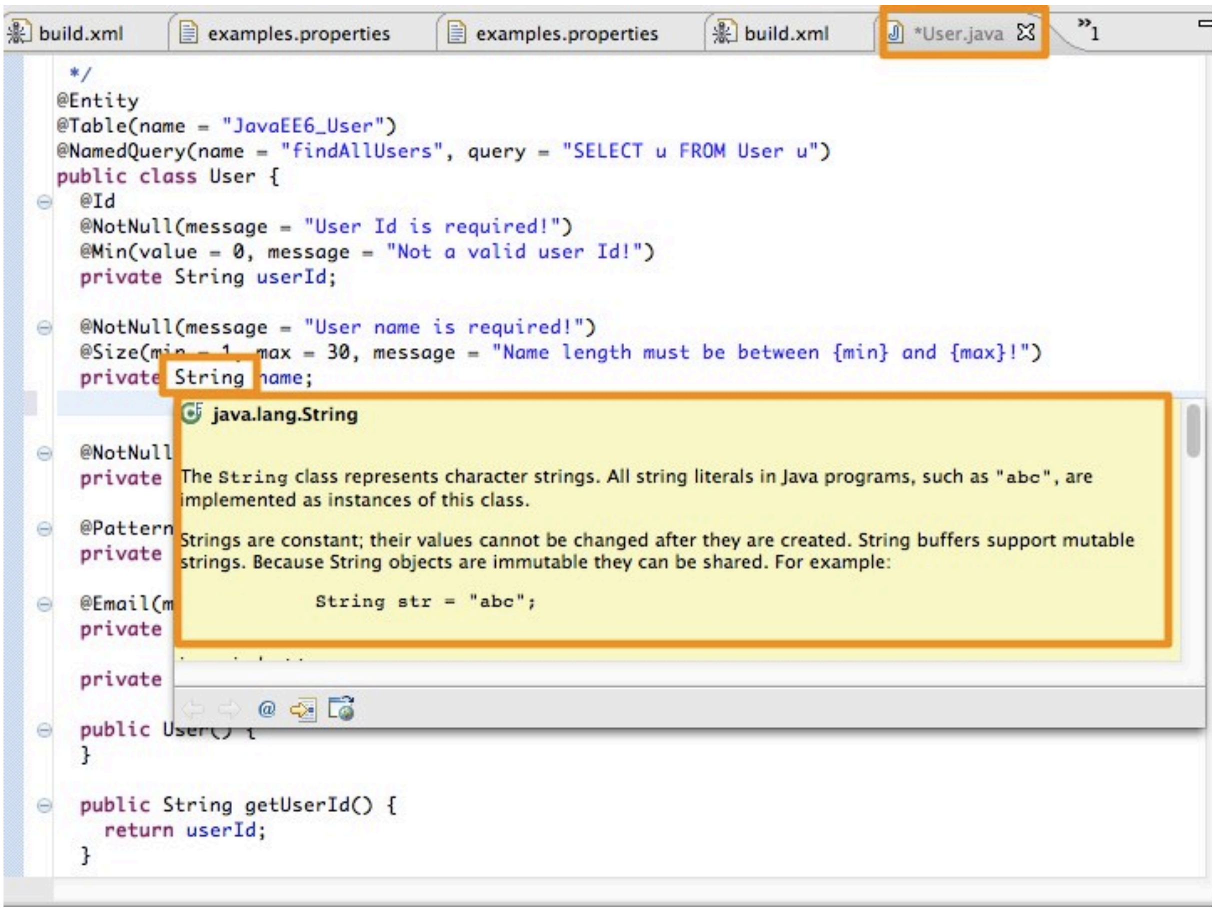
During the course of these experiments I deleted all deployments of the example server which worked fine. However there is no single step activity to deploy all examples again. So I uninstalled the example server via the WLS uninstall tool and reinstalled it via the original deployment package of WLS12c. As a result eclipse could not reconnect to the wl_server because during installation there where new credentials generated and eclipse advised me to recreate the server in the IDE. This however didn’t work. It seemed that eclipse did not really delete all its information about this server although it disappeared from the server pane. Eclipse then refused to recreate the wl_server for the same reason, i.e. that the credentials do not fit. I decided to restart eclipse but it became unstable during shutdown, so I had to kill the process. After restarting eclipse the wl_server entry could be properly set up.

This problem demonstrates that the integration of WLS and eclipse is not fully stable and restarting eclipse might be a good solution before diving into problem analysis.

Another surprise was that my ShellEd plug-in had disappeared. It seemed that the crash during eclipse shutdown has violated the IDE’s integrity. I reverted eclipse to a previous timestamp in the Installation History and restarted it. (About Eclipse Platform->Installation Details->Installation History). After that ShellEd worked again.

1.4. Adding code completion and javadoc integration

So far we can use javadoc integration of the JavaSE types and classes only, as depicted below.



```
  */
@Entity
@Table(name = "JavaEE6_User")
@NamedQuery(name = "findAllUsers", query = "SELECT u FROM User u")
public class User {
    @Id
    @NotNull(message = "User Id is required!")
    @Min(value = 0, message = "Not a valid user Id!")
    private String userId;

    @NotNull(message = "User name is required!")
    @Size(min = 1, max = 30, message = "Name length must be between {min} and {max}!")
    private String name;

    @NotNull
    private

    @Pattern
    private

    @Email(m
    private

    private

    public User() {
    }

    public String getUserId() {
        return userId;
    }
```

java.lang.String

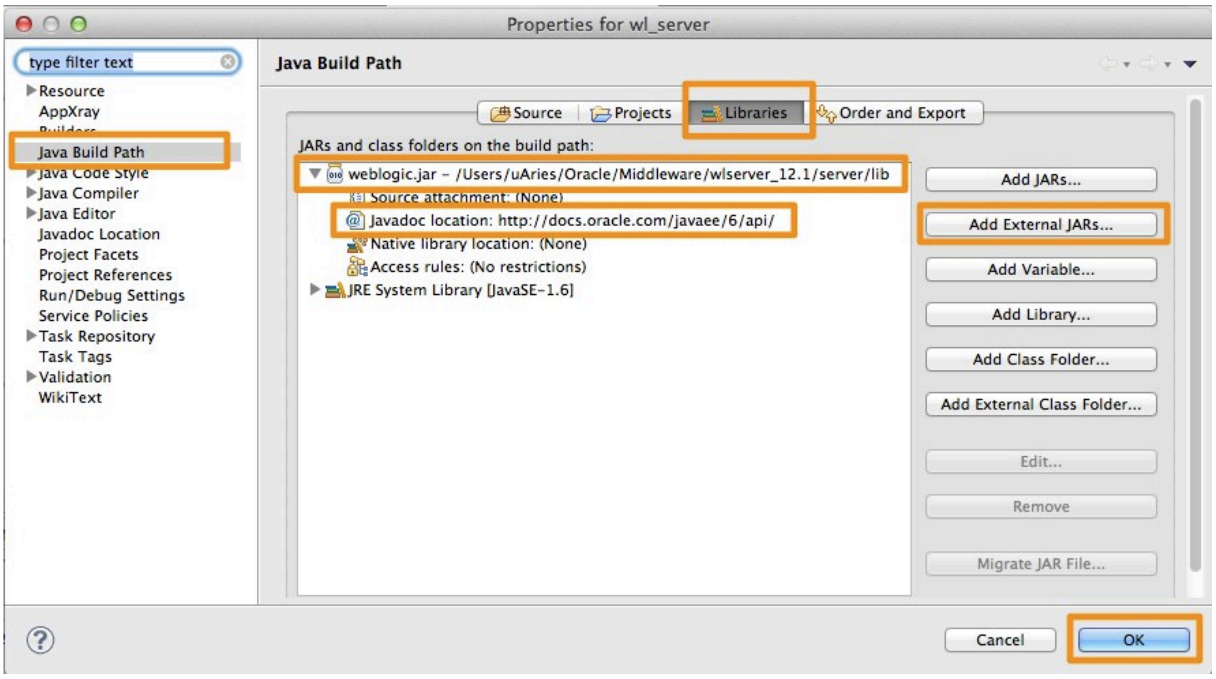
The String class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because String objects are immutable they can be shared. For example:

```
        String str = "abc";
```

Figure 16. Context help for JavaSE classes in the Java Editor.

The JavaEE classes are not recognized by eclipse so far. Since most of the examples use JavaEE classes, we add weblogic.jar to the Project Builder's library path and also include the Javadoc location.



JARs and class folders on the build path:

- weblogic.jar – /Users/uAries/Oracle/Middleware/wlserver_12.1/server/lib
  - Source attachment: (None)
  - Javadoc location: http://docs.oracle.com/javaee/6/api/
  - Native library location: (None)
  - Access rules: (No restrictions)
- JRE System Library [JavaSE-1.6]

Figure 17. Library path configuration in the project property pane.

Now we also have context help for JavaEE classes, however we can't use code completion and on the fly compilation so far.

1.5. Adding on-the-fly compilation support

For compiling, building and deploying the project we use ant build files independently of the Eclipse Java Builder. If we want to use code completion, we need to add our sources to the source path of the project, which will also compile these files and indicate syntax and other compilation errors.

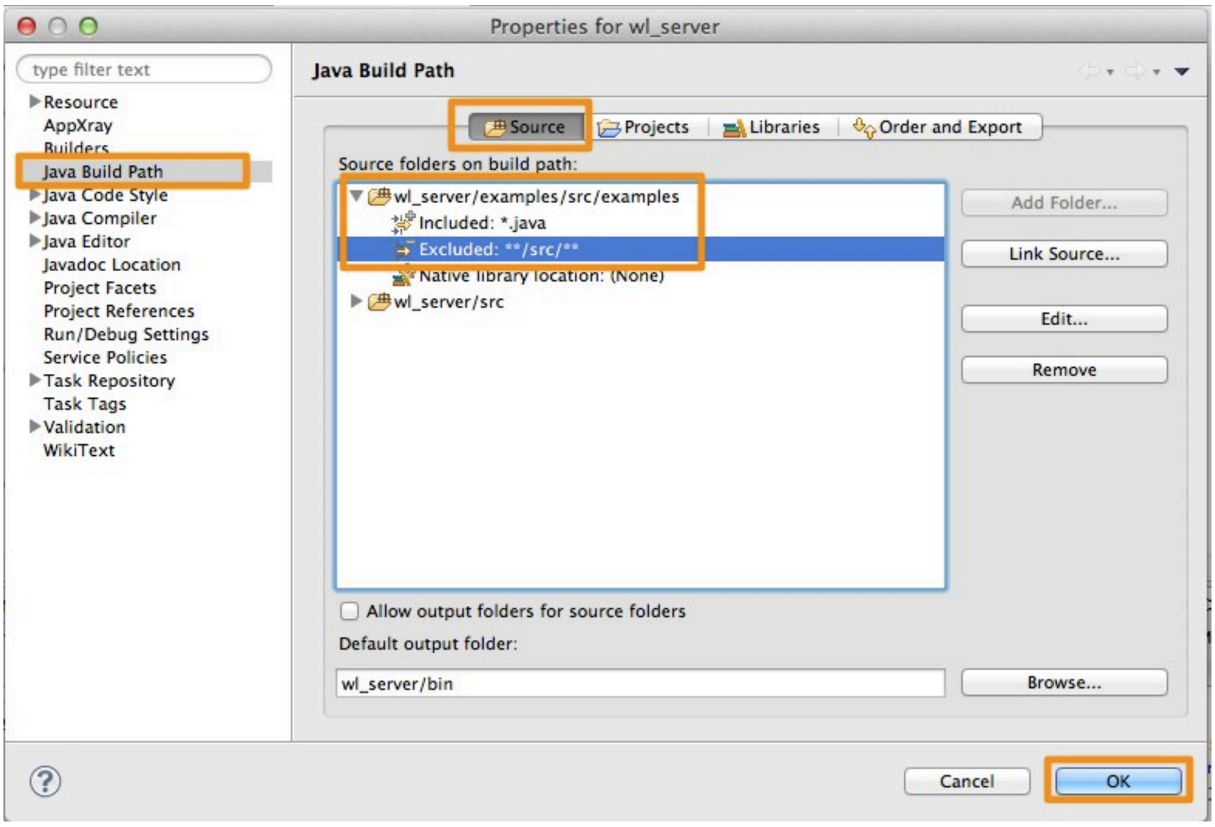
The Eclipse source path should point to the beginning of the package structure. The examples that we've imported from the wls samples directory are grouped into individual folders where most of them have their own source folder. However some examples don't use a separate source folder, their packet structure starts from the examples src folder. If we look at the example to the ejb20/basic/beanManaged classes, we see that they start at the examples/src folder.

```
aries:examples uAries$ pwd
/Users/uAries/Eclipse/workspace/wl_server/examples
aries:examples uAries$ find . -name *.java | grep ejb20/basic/beanM
./src/examples/ejb/ejb20/basic/beanManaged/AccountBean.java
./src/examples/ejb/ejb20/basic/beanManaged/Client.java
./src/examples/ejb/ejb20/basic/beanManaged/ProcessingErrorException.java
```

If we however look at the ejb30/session example we can see that it has it's own source folder, namely ./src/examples/ejb/ejb30/src.

```
aries:examples uAries$ find . -name *.java | grep ejb30/session
./src/examples/ejb/ejb30/src/java/examples/ejb/ejb30/session/ReviewManager.java
./src/examples/ejb/ejb30/src/java/examples/ejb/ejb30/session/ReviewManagerBean.java
./src/examples/ejb/ejb30/src/java/examples/ejb/ejb30/session/ReviewStatManager.java
./src/examples/ejb/ejb30/src/java/examples/ejb/ejb30/session/ReviewStatManagerBean.java
```

The build path needs to reflect this difference. In a first step we add examples/src to the eclipse source path, however we apply an exclude pattern, excluding all examples that have their own src folder.


Properties for wl_server

Java Build Path

Source folders on build path:

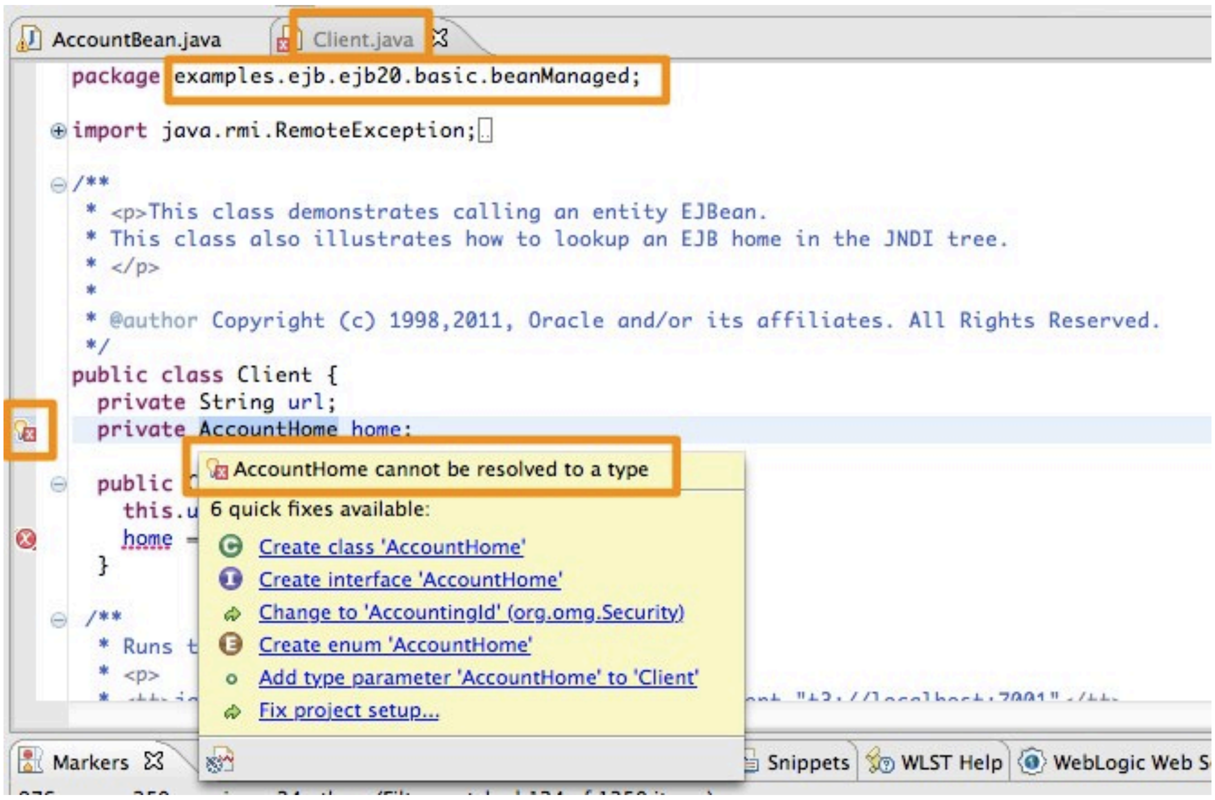
```
▼ wl_server/examples/src/examples
    Included: *.java
    Excluded: **/src/**
    Native library location: (None)
▶ wl_server/src
```

☐ Allow output folders for source folders

Default output folder: wl_server/bin

Figure 18. Configuring the source path using exclude patterns.

Now if we look at the file ./src/examples/ejb/ejb20/basic/beanManaged/Client.java we can see that it get's compiled and eclipse shows a compilation error, stating that it cannot find the type AccountHome.



```
package examples.ejb.ejb20.basic.beanManaged;

import java.rmi.RemoteException;

/**
 * <p>This class demonstrates calling an entity EJBean.
 * This class also illustrates how to lookup an EJB home in the JNDI tree.
 * </p>
 *
 * @author Copyright (c) 1998,2011, Oracle and/or its affiliates. All Rights Reserved.
 */
public class Client {
  private String url;
  private AccountHome home;

  public
    this.u
    home =
  }

  /**
   * Runs t
   * <p>
```

AccountHome cannot be resolved to a type

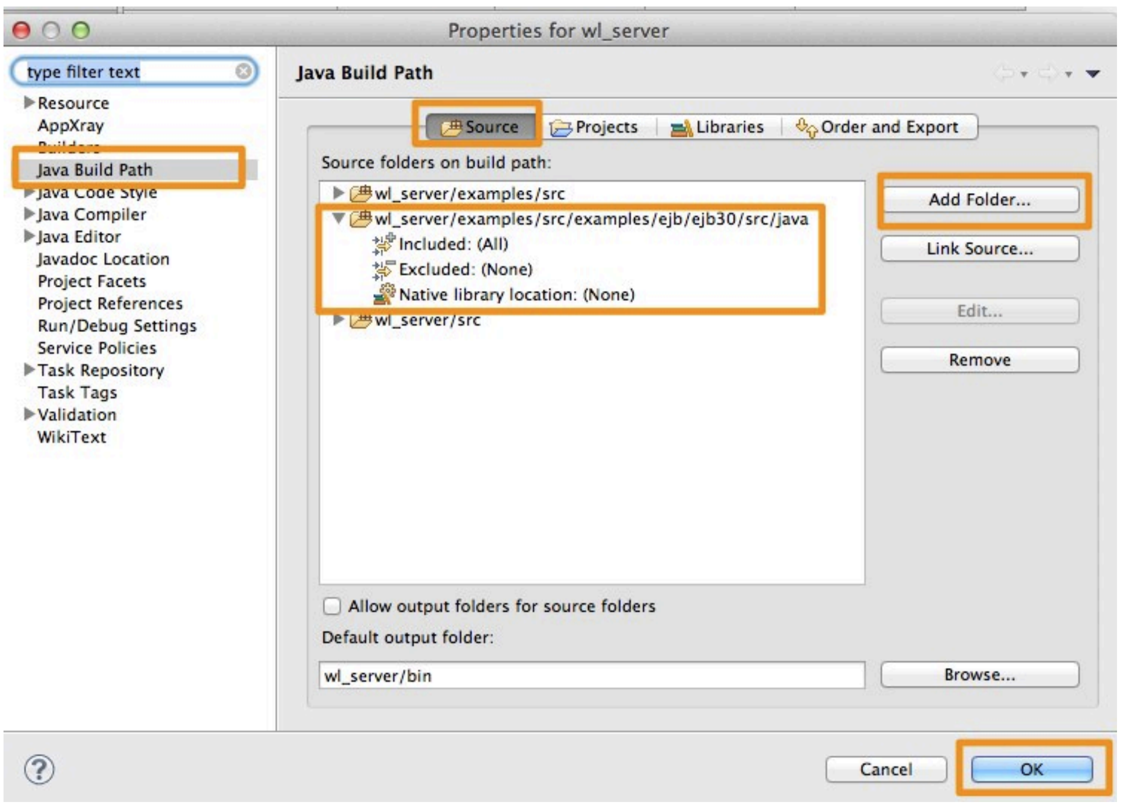
6 quick fixes available:

- Create class 'AccountHome'
- Create interface 'AccountHome'
- Change to 'AccountingId' (org.omg.Security)
- Create enum 'AccountHome'
- Add type parameter 'AccountHome' to 'Client'
- Fix project setup...

Figure 19. The Eclipse Builder displays compilation errors.

In this case AccountHome will be created during the build process and we could add the path of AccountHome.class to eclipse's build path, however we don't worry about it now, because we use the external ant compile anyway, which works nevertheless.

If we also want to use Eclipse compiler for the ebj30 example we have to add the example specific src-path. This will give us the ability to edit all java files of the ejb30 example with on-the-fly compilation support.



Source folders on build path:

```
▶ wl_server/examples/src
▼ wl_server/examples/src/examples/ejb/ejb30/src/java
    Included: (All)
    Excluded: (None)
    Native library location: (None)
▶ wl_server/src
```

☐ Allow output folders for source folders

Default output folder: wl_server/bin

Figure 20. Adding the source branch of the ejb30 examples to the source path of the project wl_server.

The source path of the project gets quite confusing. If we want to work on single branch of the wls examples, i.e. only on the ejb30 examples we rather set up an eclipse project with only these files. This will be shown in a separate paper.

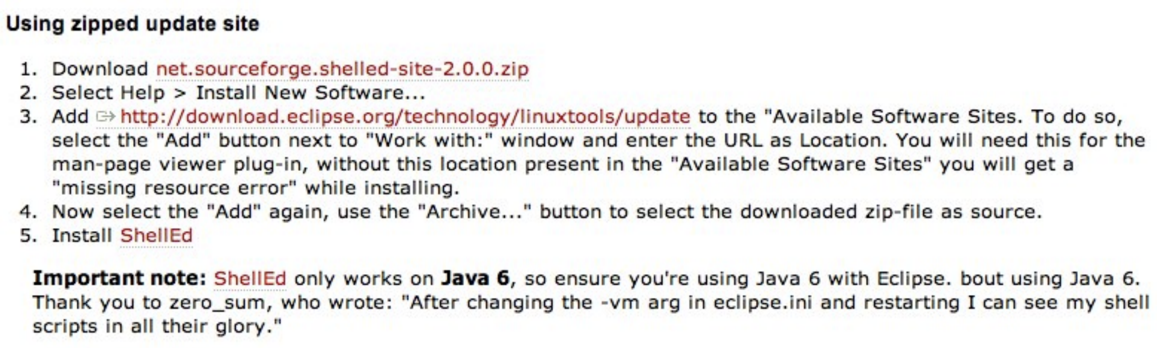
2. Installing ShellEd

We install the ShellEd plugin for eclipse for convenience in shell script editing. The installation via the update site did not work on my computer. It complained about missing Linux Tools and wouldn't install it. However the ShellEd installation guide describes a zip installation, which works fine.

We download the file net.sourceforge.shelled-site-2.0.1.zip

See http://sourceforge.net/apps/trac/shelled/wiki/Documentation/InstallGuide

Here are the instructions from shelled wiki:



Using zipped update site

1. Download net.sourceforge.shelled-site-2.0.0.zip
2. Select Help > Install New Software...
3. Add http://download.eclipse.org/technology/linuxtools/update to the "Available Software Sites. To do so, select the "Add" button next to "Work with:" window and enter the URL as Location. You will need this for the man-page viewer plug-in, without this location present in the "Available Software Sites" you will get a "missing resource error" while installing.
4. Now select the "Add" again, use the "Archive..." button to select the downloaded zip-file as source.
5. Install ShellEd

Important note: ShellEd only works on **Java 6**, so ensure you're using Java 6 with Eclipse. bout using Java 6. Thank you to zero_sum, who wrote: "After changing the -vm arg in eclipse.ini and restarting I can see my shell scripts in all their glory."