



Project Acronym: **OPTIMIS**  
Project Title: **Optimized Infrastructure Services**  
Project Number: **257115**  
Instrument: **Integrated Project**  
Thematic Priority: **ICT-2009.1.2 – Internet of Services, Software and Virtualisation**

## Integrated Development Environment Installation Guide

*Activity 2: Service Construction*

*WP 2.1: Programming Model and IDE*

<b>Due Date:</b>	M34
<b>Submission Date:</b>	31/04/2013
<b>Start Date of Project:</b>	01/06/2010
<b>Duration of Project:</b>	36 months
<b>Organisation Responsible for the Deliverable:</b>	BSC
<b>Version:</b>	1.0
<b>Status</b>	Final
<b>Author(s):</b>	Jorge Ejarque BSC



Project co-funded by the European Commission within the Seventh Framework Programme

**Dissemination Level**

<b>PU</b>	Public	<b>X</b>
<b>PP</b>	Restricted to other programme participants (including the Commission)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission)	



## Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	10/04/2012	First version	Jorge Ejarque
0.2	20/09/2012	Updated Version	Jorge Ejarque
0.3	26/03/2013	Updated Version	Jorge Ejarque



## Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>5</b>
1.1	GLOSSARY OF ACRONYMS.....	5
<b>2</b>	<b>INSTALLATION GUIDE .....</b>	<b>6</b>
2.1	RELEASE INFORMATION .....	6
2.2	MINIMAL SYSTEM REQUIREMENTS .....	6
2.3	PLATFORMS SUPPORTED .....	6
2.4	SOFTWARE PRE-REQUISITES AND DEPENDENCIES.....	6
2.5	INSTALLATION INSTRUCTIONS .....	7
2.5.1	<i>Installation from source code .....</i>	<i>7</i>
2.5.2	<i>Installation from Service Construction VM .....</i>	<i>7</i>
2.6	GETTING STARTED.....	8
2.6.1	<i>Using the Software .....</i>	<i>8</i>
2.6.2	<i>Testing the Software.....</i>	<i>13</i>
2.6.3	<i>Configuration.....</i>	<i>13</i>
<b>3</b>	<b>REFERENCES .....</b>	<b>14</b>

## Index of Tables

Table 1	Software dependencies.....	6
---------	----------------------------	---

## Index of Figures

Figure 1.	OPTIMIS IDE Service Editor .....	9
Figure 2.	Service Class wizard.....	9
Figure 3.	Orchestration Element Wizard.....	10
Figure 4.	Core Element Wizard.....	11
Figure 5.	Orchestration Element implementation .....	12
Figure 6.	Building and Deployment Tab.....	13



## 1 Introduction

This document includes the installation guide for the software component Integration Development Environment (IDE). The following sections include the pre-requisites and steps to install the IDE plugin in the Eclipse Platform.

### 1.1 Glossary of Acronyms

Table 1 -Acronyms table

Acronym	Definition
D	Deliverable
WP	Work Package
IDE	Integrated Development Environment
OS	Operating System



## 2 Installation Guide

### 2.1 Release information

Table 2 - Release component information

Component Name	Release Number	Release Date
Integrated Development Environment	1.1.0	31/3/2012

### 2.2 Minimal System Requirements

The minimal System Requirements for IDE components are the same as for Eclipse platform [1]. In addition to these minimal requirements, the installation of the IDE plug-in requires 40 Mbytes of space in the hard disk of the machine. It also requires network connectivity with the OPTIMIS Image Creation and Deployment services.

### 2.3 Platforms Supported

The IDE plug-in for eclipse is currently tested and supported for Linux OS. We are working on providing it for other platforms.

### 2.4 Software Pre-requisites and Dependencies

The following list provides the pre-requisites and dependencies of the IDE plug-in.

Table 3 Software dependencies

Product	Version	Licence
Eclipse	Indigo ( or higher)	Eclipse license
Eclipse Web Tool Plugin <sup>i</sup>	(bind to the Eclipse version)	Eclipse license
Eclipse Zest/dot Plugin	(bind to the Eclipse version)	Eclipse license
Service Manifest API	1.0.8	
License Manager API	0.2.1	

The Eclipse Platform and their required plugins have to be installed before the OPTIMIS IDE installation. If you do not have the Eclipse platform installed follow the instruction described in [1] in order to install it. Afterwards, use the Eclipse Marketplace (located in Eclipse Help menu) to install the other required plugins.

The Service Manifest API will be automatically detected by maven, so the user does not have to do anything to install this dependency. Regarding the License Manager API, you need an account for the elasticLM maven repository. In case you do not have it, contact [Wolfgang.Ziegler@scai.fraunhofer.de](mailto:Wolfgang.Ziegler@scai.fraunhofer.de)



## 2.5 Installation Instructions

This section describes the steps to install the IDE plug-in in the Eclipse platform. There are two ways to get and install the OPTIMIS IDE: from the source code available in the OPTIMIS Subversion [2]; or getting the Service Construction VM available in the OPTIMIS web site.

### 2.5.1 Installation from source code

Once Eclipse and the required plugins are already installed in your machine follow the steps described below to install the OPTIMIS IDE plug-in.

1.- Download the source code from the OPTIMIS Subversion [2]

```
Svn checkout http://<optimis/svn/path>/IntegratedDevelopmentEnvironment
```

2.- Compile the code and build the plug-in using Maven [3]

```
mvn clean install
```

3.- Copy the plug-in jar in the dropins folder of your Eclipse installation

```
cp target/IDE-1.XXX.jar ECLIPSE_HOME/dropins/
```

4.- Initiate the eclipse platform

### 2.5.2 Installation from Service Construction VM

The instructions for installing the Service Construction VM has been detailed for the VirtualBox manager version 4 or higher[5].

To install the Service Construction Virtual Appliance you need to perform the following steps:

1. Download the Virtual Appliance file from the web site ([www.optimis-project.eu](http://www.optimis-project.eu))
2. Open the VM VirtualBox [29] Administrator
3. Select the *Import Virtualised service* option in the *File* menu and an import wizard will be open.
4. Click on *Select...* and choose the downloaded *.ova* file and click *Next*.
5. The following page will provide a default VM configuration for this appliance. In case some parameters are not suitable for your system change them. When finish, click *Import* to start the VM import process.
6. After a successful import process, a new VM will appear in the main window of the VM VirtualBox Administrator.
7. Select the newly created VM and click *Start*.

The IDE is installed as an Eclipse plug-in. The plugin has also prepared to work with a PM runtime libraries installed in `/home/user/OPTIMIS/ProgrammingModel/runtime/integratedtoolkit`. This is PM runtime location you have to set when a new OPTIMIS project is created with the IDE.



Once the VM is running, you will see an Eclipse icon in the desktop. Click on it to launch the Eclipse platform and start using the OPTIMIS IDE plug-in. Details on how to use the IDE can be found in next section and IDE user guide.

For a better integration of the VM with your desktop, you can also install the Virtual Box Guest Additions. To install it follow these steps once the VM is started.

1. click on VBOXADDITIONS\_4.1.20\_80170
2. `sudo /media/VBOXADDITIONS_4.1.20_80170/VBoxLinuxAdditions.run`
3. root password is "user"

## 2.6 Getting started

A Service developed by the OPTIMIS Programming Model is composed by Orchestration and Core Elements. A Core Element is a piece of code which either is repeated several times in the service code and potentially in parallel or is performing a computation which requires a lot of resources or both. An Orchestration Element is the code which implements the service functionality invoking several defined Core Elements.

### 2.6.1 Using the Software

If the IDE plug-in has been successfully installed an OPTIMIS menu with different actions should appear in the Menu bar and a set of OPTIMIS wizards should appear in the "File->New..." sub menu when the Eclipse platform is initiated. There is also a Service Editor () which will be available when a new project is created. You can use these actions, wizards and editor to easily create a new service with different Orchestration and Core Elements in ten steps.

#### Step 1.

Create a new OPTIMIS project with the New OPTIMIS project wizard located in "File ->New->Project". Select the name of the project, the name of the main packages and the location where the OPTIMIS Programming Model is installed in your machine and click "Finish". As result of this wizard a new project and Service Editor will be opened as shown in Figure 1.



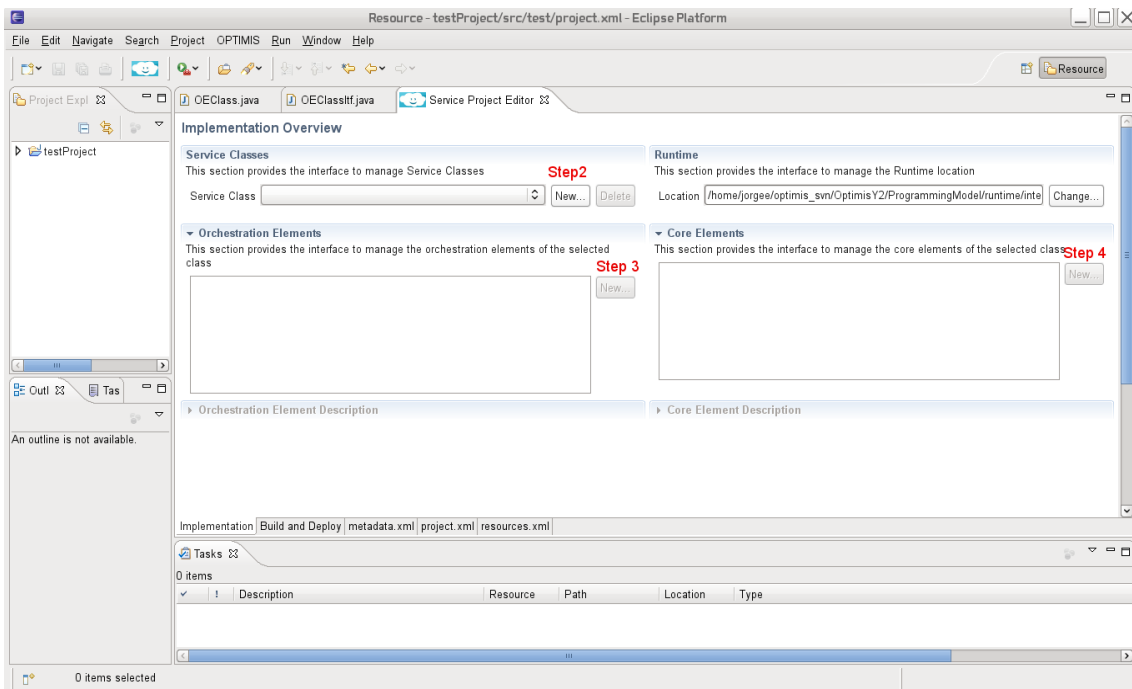


Figure 1. OPTIMIS IDE Service Editor

## Step 2.

Create a new Service Class using the Implementation Tab of the Service Editor. A Service Class wizard (such as Figure 2) will be opened by clicking the “New...” button in the Service Classes section (Figure 1. Step 2). After fulfilling the Service class name, a new class and a Core Element interface will be created in the main package of the project.

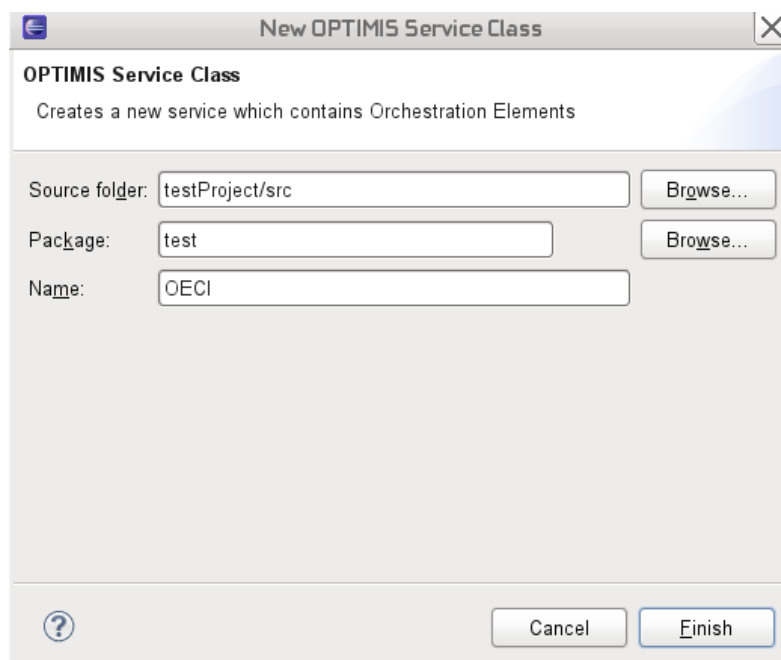


Figure 2. Service Class wizard



**Step 3.**

After the creation of the Service Class, you can include an Orchestration Element in this class where a workflow with different Core Element invocations will be programmed. You can create an Orchestration Element by using the Orchestration Element Wizard (Figure 3) opened when clicking the “New...” button of the Orchestration Element section located in the Implementation Tab of the Service Editor (Figure 1 Step 3)

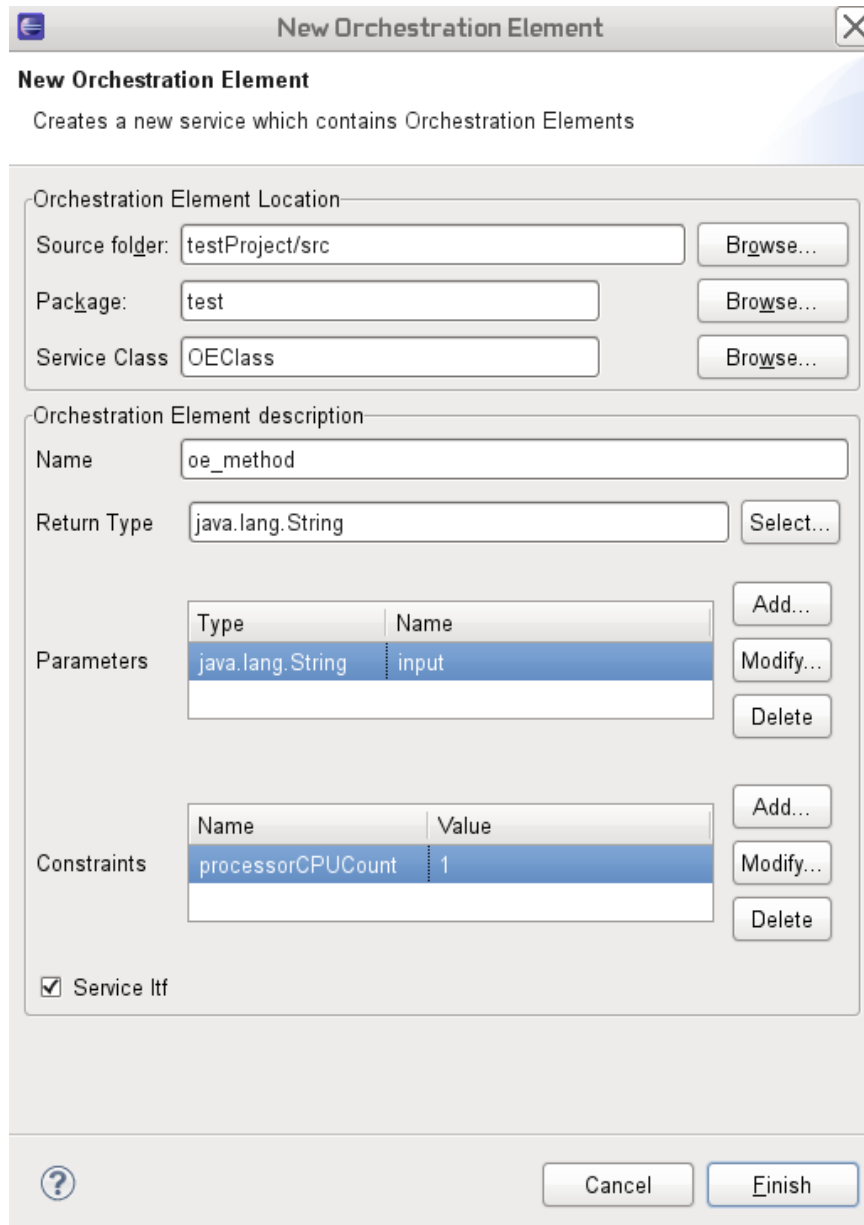
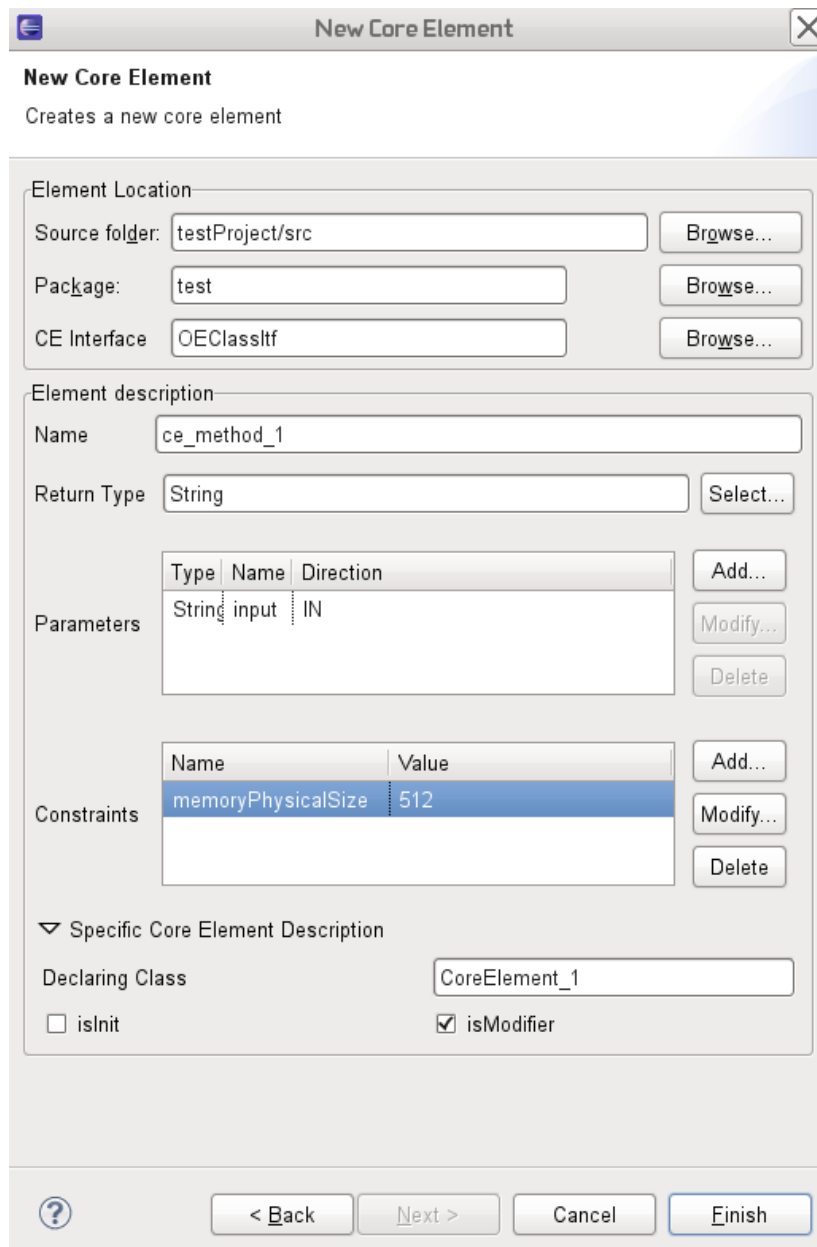


Figure 3. Orchestration Element Wizard



**Step 4.**

The Core Elements invoked from the Orchestration Elements of a Service class must be defined in the Core Element Interface created when adding a Service Class. To do it the OPTIMIS ide offers a Core Element wizard which is open by clicking the “New...” button of the Core Element section located in the Implementation Tab of the Service Editor (Figure 1 Step 4). You currently have 3 options to create a new Core Element: From scratch, creating a new method in a Core Element class, from an existing JAR package or from an existing Web Service described by its WSDL file.



**Figure 4. Core Element Wizard**



### Step 5.

Once a Core Element has been defined to the Core Element interface, the developer can implement its invocation inside the Orchestration Element code (as depicted in . Repeat step 4 and 5 to implement the service functionality as a composition Core Elements invocations in a sequential fashion.

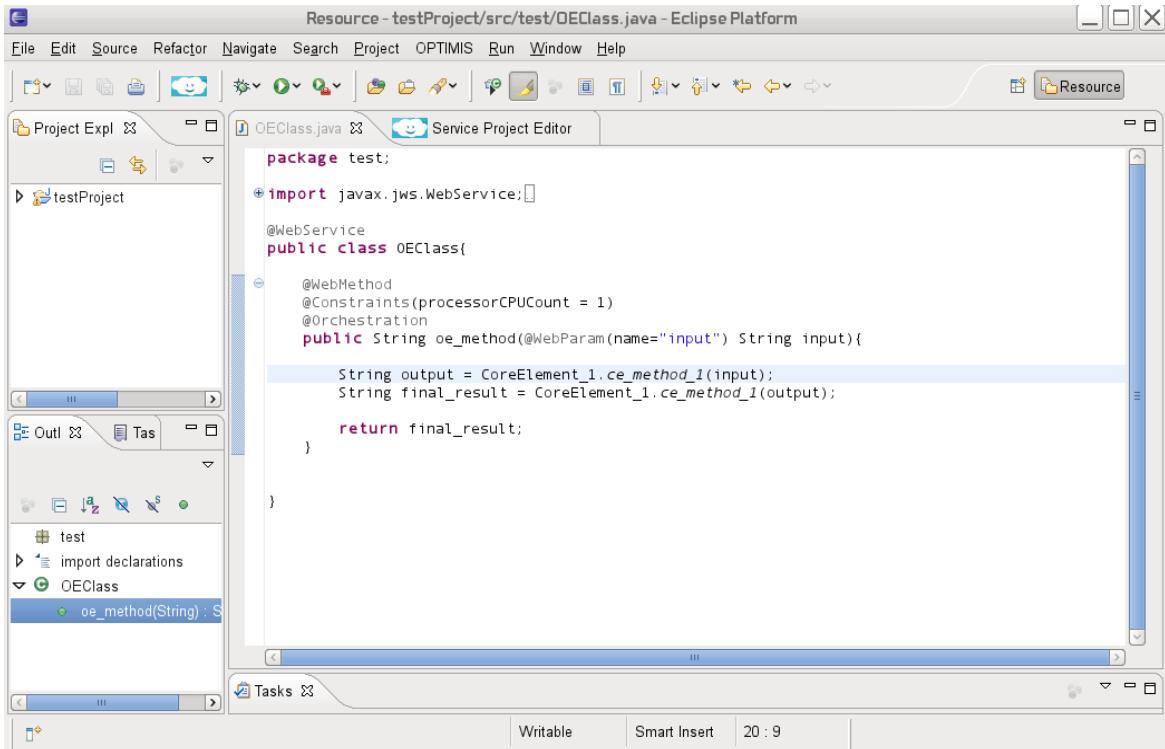


Figure 5. Orchestration Element implementation

### Step 6.

Once you have implemented the service, move to the Building and Deployment Tab (Figure 6) of the Service Editor. Define the service packages by grouping the Core Elements with similar requirements or functionality by clicking “New...” button of the Service Packages section (Figure 6 Step 6). Once the Service Developer has defined the packages for all the Core Elements, click the “Generate” button to generate the defined packages (Figure 6 Step 6b). As result of the package generation process a set of war, jar and zip files will be generated in the “output” folder of the OPTIMIS Service project.

### Step 7.

Check the correct behavior with a local deployment selecting the “Localhost” deployment option (Figure 6 Step 7) will deploy the Orchestration in a Web Application Server (such as Tomcat) and Core Elements in a selected folder of your machine. Once the developer has checked the correct behavior select the “OPTIMIS cloud” deployment option.

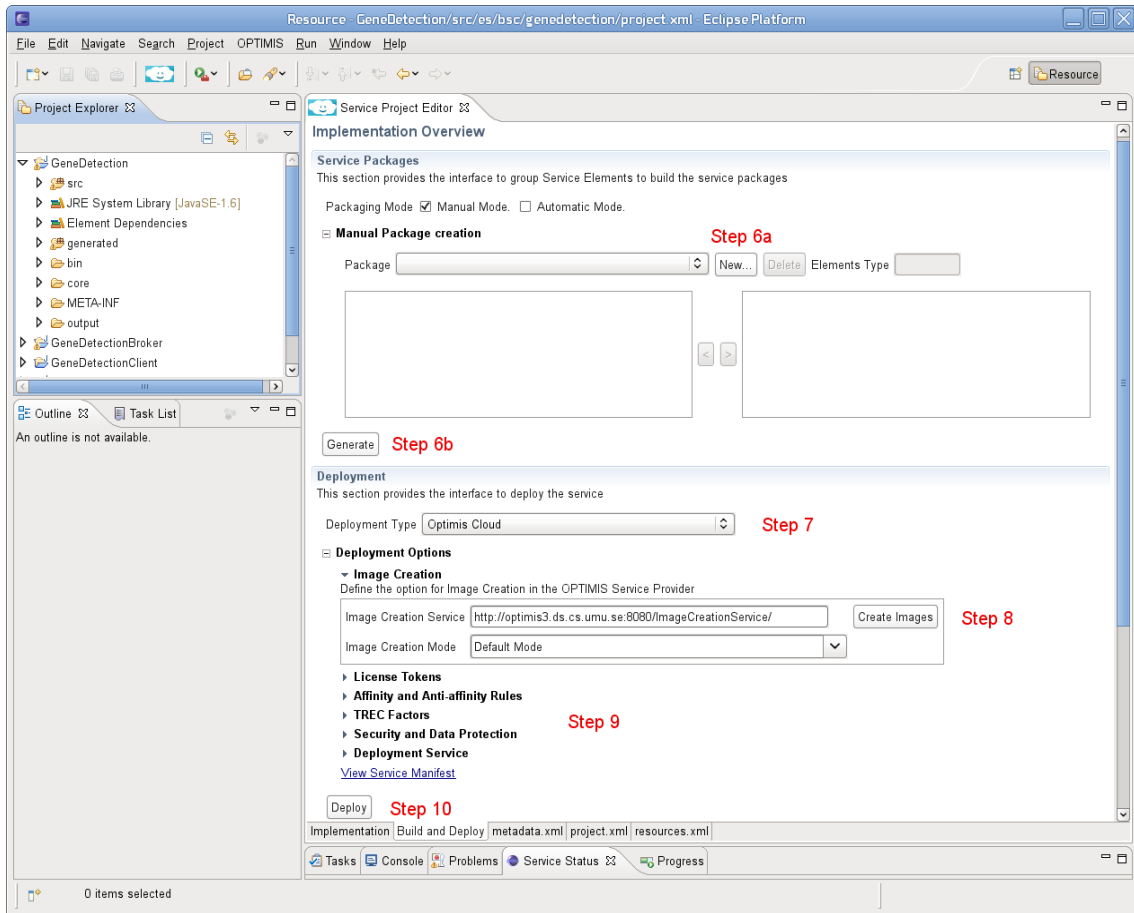


Figure 6. Building and Deployment Tab

### Step 8.

Create Service VM images by opening the Image Creation options, setting the location of Image Creation Service and clicking the “Create Images” button. (Figure 6 Step 8)

### Step 9.

Define TREC, Security & Data protection, Affinity & Anti-affinity rules between for the service packages. Generate license tokens in case of your service require them (Figure 6 Step 9)

### Step 10.

Deploy in the OPTIMIS Cloud by clicking “Deploy” (Figure 6 Step 10)

## 2.6.2 Testing the Software

Tests are not provided

## 2.6.3 Configuration

Not required



### 3 References

- [1] Eclipse Indigo <http://www.eclipse.org/indigo/>
- [2] Apache Subversion <http://subversion.apache.org/>
- [3] Apache Maven <http://maven.apache.org/>
- [4] Integrated Development Environment User's Guide, OPTIMIS project.
- [5] Oracle VirtualBox <https://www.virtualbox.org/>

---

<sup>i</sup> The Web Tool Plugin is already installed in the Eclipse version for J2EE developers. If you have installed this eclipse version, you do not need to install this requirement