

OIO Service Oriented Infrastructure

**RASP Library for Java**

**Version 2.0.0**

**Installation Guide**

# Contents

1	Introduction .....	3
2	System requirements .....	4
3	Installation.....	5
3.1	Working on a local copy of the RASP library .....	5
3.2	Creating a branch in the online code repository .....	6
4	Verifying your setup .....	11

# 1 Introduction

The OIOSI RASP Library for Java is a java based toolkit for implementation of RASP business applications.

This distribution is Version 2.0.0.

The distribution is part of the OIOSI work for exchanging business documents in a secure and reliable way using the internet. See <http://www.digst.dk/Loesninger-og-infrastruktur/NemHandel/For-it-udviklere> for more information.

The framework can be downloaded from <http://digitaliser.dk/group/405442/resources/type/150019>.

## 2 System requirements

- JDK 6 or later
  - The java runtime environment must have Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files installed to support strong encryption.

JDKs and JCE package can be found here – installation instructions are included in the downloaded packages.

<http://java.sun.com/javase/downloads/index.jsp>

- Apache Axis2  
[http://ws.apache.org/axis2/download/1\\_5\\_1/download.cgi](http://ws.apache.org/axis2/download/1_5_1/download.cgi)
- Apache Rampart - WS-Security implementation for Axis2  
<http://ws.apache.org/rampart/download/1.4/download.cgi>
- WSO2 Commons – Mercury – WS-ReliableMessaging implementation for Axis2  
<http://dist.wso2.org/products/commons/mercury/0.91/mercury-0.91-bin.zip>
- Apache Commons – Configuration, Lang, Collections, JXPath  
<http://commons.apache.org/>
- BouncyCastle  
<http://www.bouncycastle.org/>
- JUnit  
<http://www.junit.org/>
- OpenUddi  
<http://sourceforge.net/projects/openuddi/>

The requirements can also be found in the code release of RASP.

Recommendations for development:

- Eclipse - <http://www.eclipse.org/>
- Ant 1.6.x, with JUnit – <http://ant.apache.org> (1.6.x releases can currently be found at <http://archive.apache.org/dist/ant/binaries/> )  
(Ant also comes with Eclipse)
- Apache Tomcat 6.0.x – <http://tomcat.apache.org> (Core package)

Recommended tools for contributors to the online community

- Subclipse – Subversion plugin for Eclipse - <http://subclipse.tigris.org>

The library has been tested on the following operating systems:

- Windows XP
- Windows 7

## 3 Installation

### Binary release

The binary release is located at <http://digitaliser.dk/group/405442/resources/type/150019>.

To use this binary jar file one must have all dependencies present on the classpath. The RASP library depends on a wide range of external libraries. These jar files can either be found in the code release zip file or downloaded manually.

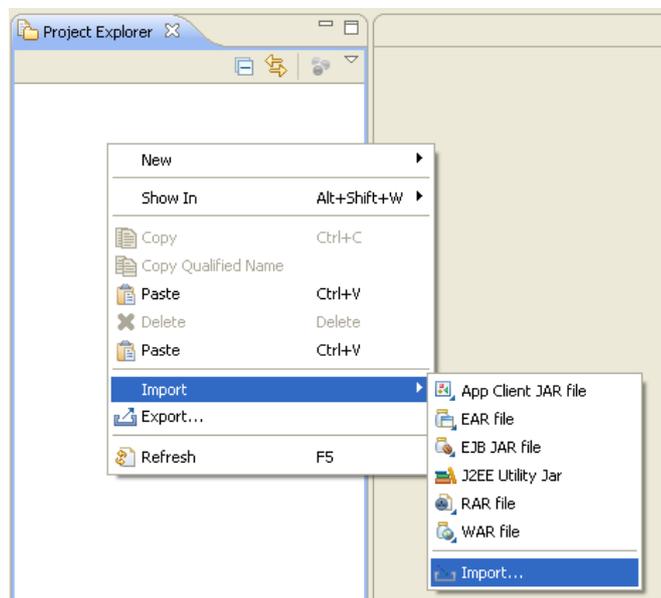
### Code release

The code release is also located at <http://digitaliser.dk/group/405442/resources/type/150019>.

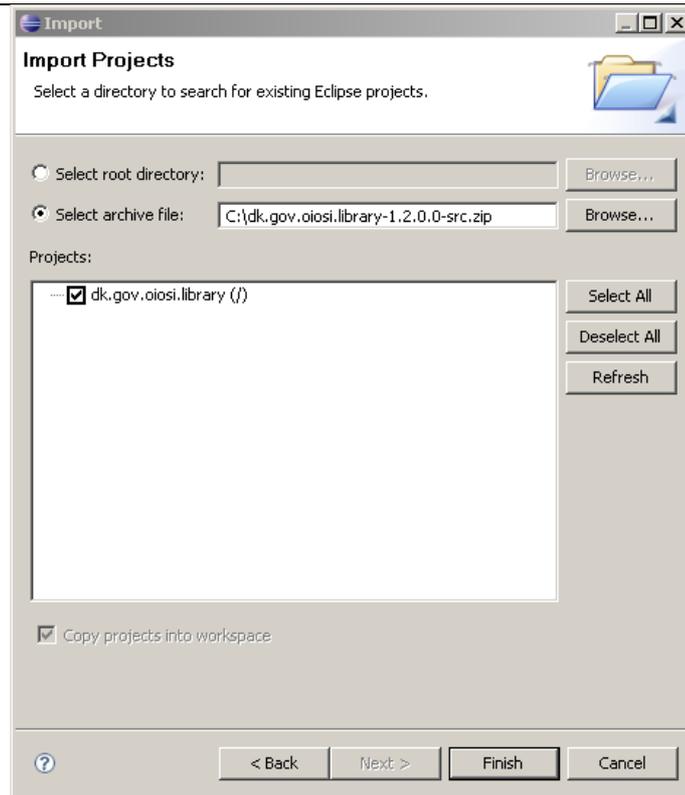
## 3.1 Working on a local copy of the RASP library

This section describes how to import a local copy of the RASP source code into Eclipse. In case you'd like to contribute to the open source community please read the next section instead.

- Start Eclipse in a workspace of your choice.
- Right click in the project explorer and select "Import.."



- Select "General>Existing Projects into Workspace"
- In the new dialog window, select the zip file.



- Click finish, and you should now have the RASP library code in your project explorer.

The examples code can be imported in the same way.

## 3.2 Creating a branch in the online code repository

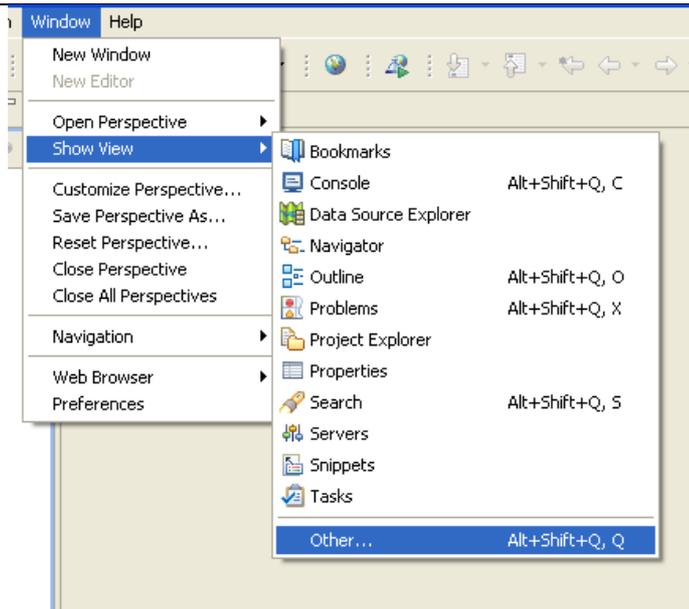
The RASP library code is kept in a Subversion (<http://subversion.tigris.org/>) repository on the address

<https://svn.softwareborsen.dk/openebusiness/dk.gov.oiosi/library/java/>

For accessing this code the Eclipse plug-in Subclipse is recommended. Please start by installing it, following the instructions on the following page:

<http://subclipse.tigris.org/install.html>

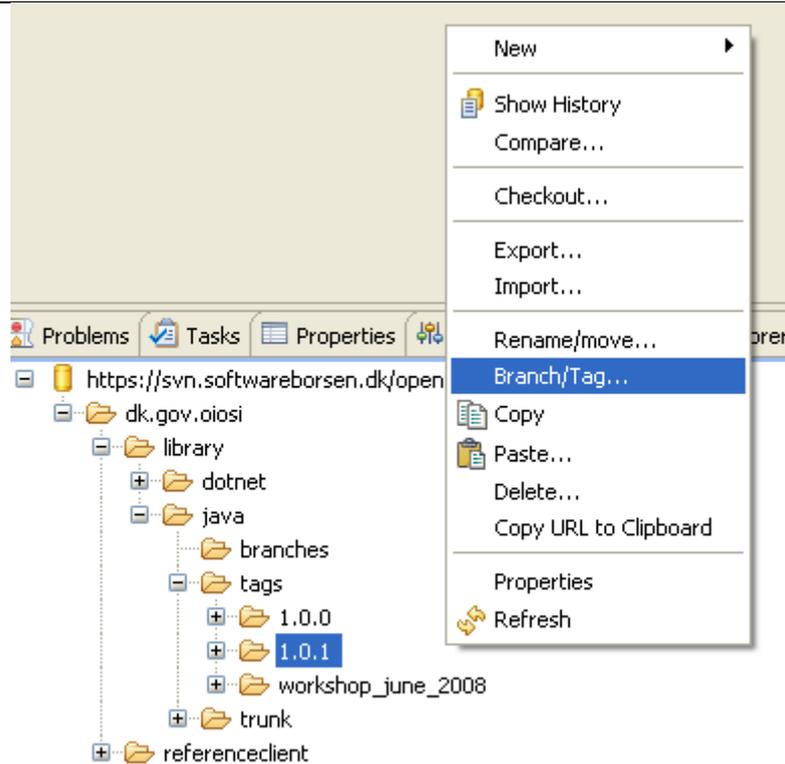
- After Subclipse has been installed, we can browse the repository by selecting “Show View > Other...” under the Window menu.



- In the list of options, select SVN > “Svn Repositories”
- In the newly opened view, right click and select “New > Repository Location...”



- Choose the RASP repository by entering the address <https://svn.softwareborsen.dk/openebusiness/dk.gov.oiosi/library/java/>
- Create your own branch, either from the main development line (the trunk) or from a stable release. It is recommended that you branch your code from the latest stable release. Browse to the tags\1.2.2 directory and right click on it.

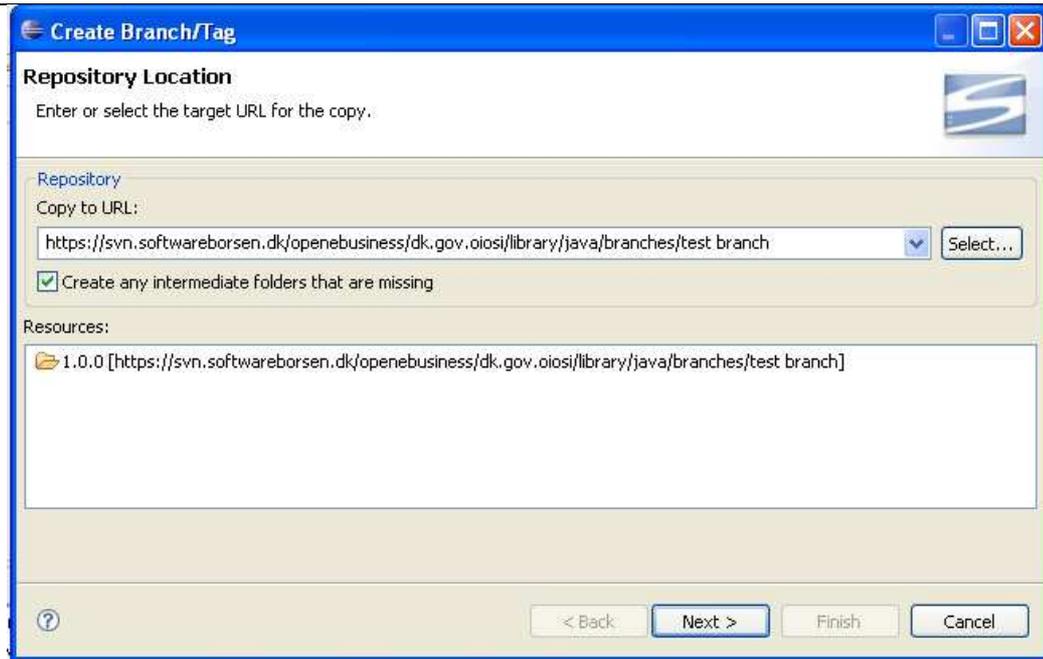


- Select “Branch/Tag..”, and enter a directory to which you’d like to branch the code. Make sure the branch goes under the **branches** directory and not the tags (which is for official releases).

The path should then be:

<https://svn.softwareborsen.dk/openebusiness/dk.gov.oiosi/library/java/branches/my-branch-name>

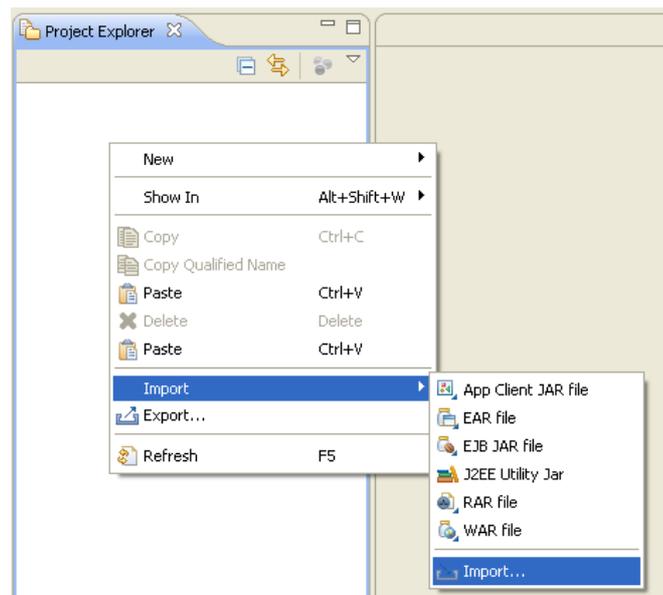
When done, click next.



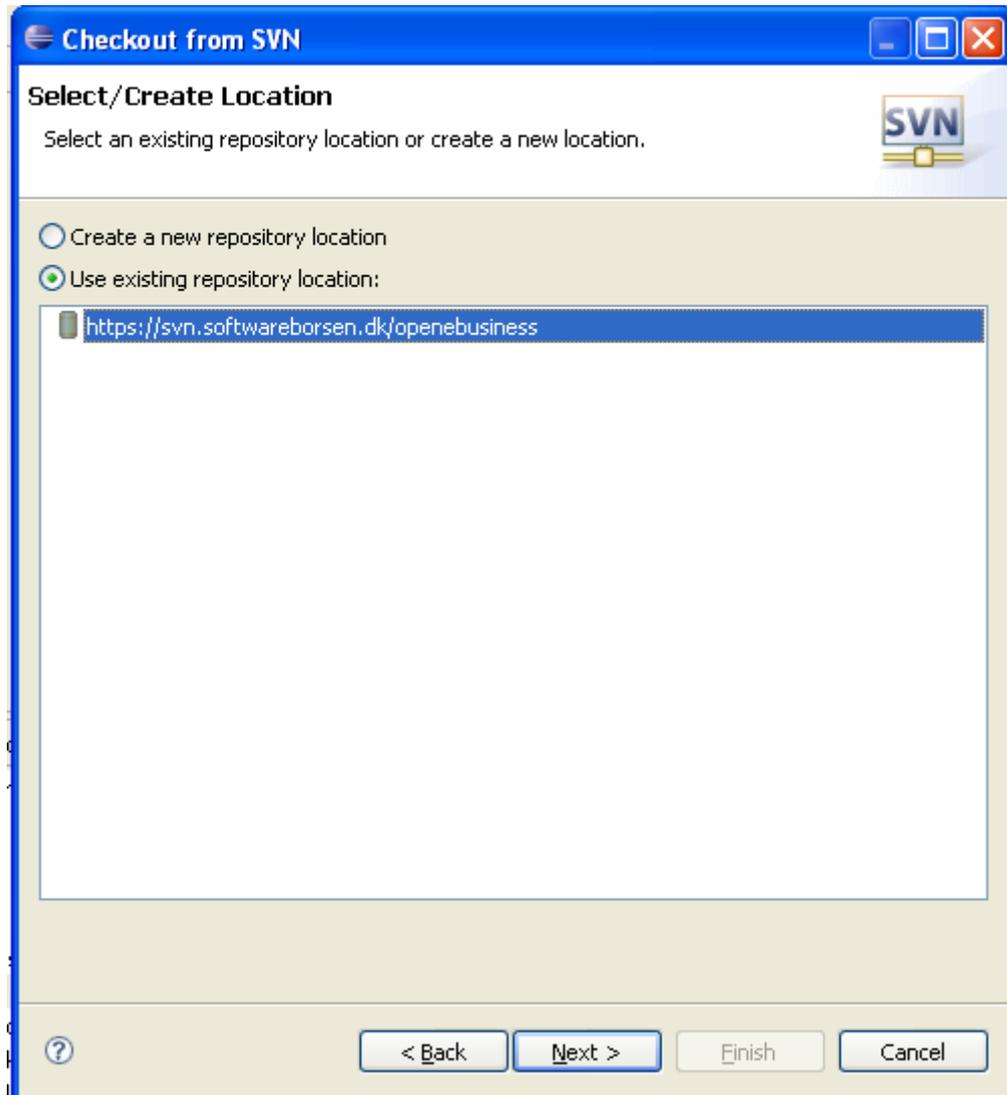
- Select HEAD revision to get the latest code. Click next.
- Add a comment, one that describes who is developing this branch and what the purpose is. When you're done, click finish and you have made a branch.

Now it's time to download a working copy of the branch to your workspace.

- Right click in the project explorer and select "Import > Import...". In the main list select "SVN>Checkout projects from SVN"



- The RASP repository address should be in the list (if not select “create a new repository location”). Select it and press next



- Browse to your newly created branch and press “Finish”. The code should download, and you have a local working copy. For instructions on how to check in code changes please see the Subclipse documentation.

## 4 Verifying your setup

### Library code release

To verify that your eclipse project has been set up properly you can run the main method located in the `RaspEnvironmentCheck.java` file.

The output should look similar to,

```
Checking OS - Check OK...

Checking Axis2 - Check OK...
Checking Mercury - Check OK...
Checking Rampart - Check OK...

Checking encryption - Check OK...
Checking for Ant - Check OK...

Checking for Bouncy-castle - Check OK...
```

The first line simply tests that you are running on a tested operating system.

The next three lines indicate whether Axis2 and Mercury can be found. **Note that for these tests to pass, all .jar files that come with Axis2 1.5.1, Rampart 1.6 and Mercury 0.91 (in those exact versions) need to be present in the /lib directory.**

After that three lines check the Java environment, to make sure you have the correct version of Java installed, and with Java cryptography extensions, and that Ant is accessible from the command line prompt.