SDSS development environment setup

Politecnico di Torino

version 0.2.0 - 13 December 2013

Contents

1   Introduction

This manual provides the developer a handbook which describes how to setup an environment suitable for extending and modifying the SDSS source code. Hence, it is assumed that the reader is familiar with the technologies used in the Security Decision Support System such as the Java programming language and the Eclipse framework.

Additional information about the internal structure of each SDSS component can be found in the following documents:

- D2.7 - Software for the IT policy language;

- LA Generation Service - programmer manual;

- Infrastructure Configuration Service - programmer manual;

- Analysis Service - programmer manual;

- Configuration Editor - programmer manual.

This document is structured as follows. The Section 2 describes how a suitable Eclipse IDE installation can be achieved. The Section 3 describes how the developer can extract the APIs documentation directly from the source code comments. The Section 4 lists a series of useful tools that can be used to modify the non-source components of the SDSS such as the ontologies. Finally, the Appendix A contains the answers to a number of commonly encountered issues.

2 IDE setup

In order to modify the SDSS source code, the developer must setup a working copy of the Eclipse IDE[1] on its machine. The following paragraphs describe how to download it and configure it for editing the SDSS code.

Installing the Eclipse IDE

The Eclipse IDE comes in many flavors for many different tasks such as implementing Java programs, C/C++ binaries or mobile applications. In particular, the *Eclipse for RCP and RAP Developers* version is tailored for developing an Eclipse-based plug-in system such as the SDSS. Hence, the first step is to download and install the aforementioned variant from `http://www.eclipse.org/downloads/`, as shown in Figure 1. Please, be sure to select an Eclipse Kepler 4.x version since the older IDEs are not compatible with the Security Decision Support System.
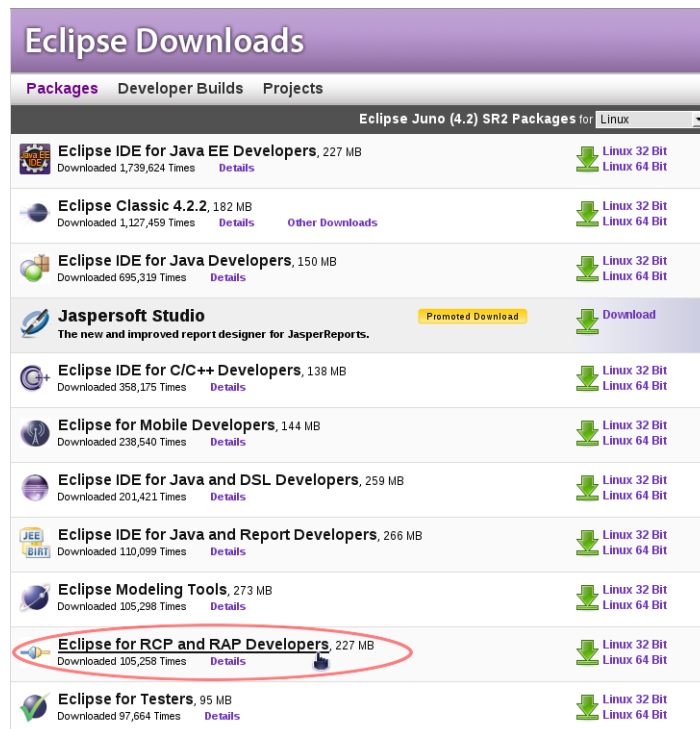

Figure 1: The Eclipse download page.

Configuring the target platform

The Eclipse for RCP and RAP Developers version contains nearly everything needed to develop a proper Eclipse plug-in but Remote Application Platform (RAP)[2] which must be installed manually. Note that the SDSS makes use of RAP version 2.2, so that the previous releases are not supported.

The developer needs to create and configure a new target platform in its Eclipse IDE copy in order to locally launch and debug the Security Decision Support System. A *target platform* is a set of bundles which are used to execute a RAP server on a machine so that a RAP application can be locally launched without deploying it into a servlet container such as Tomcat.

Apart the basic RAP plug-ins, the SDSS makes use of several other bundles[3]:

- the RAP File Upload Support;

---

[1]More information is available at `http://eclipse.org/`.
[2]More information is available at `http://eclipse.org/rap/`.
[3]All of these plug-in are available at `http://eclipse.org/rap/incubator/`.

- the Common Navigator Framework;

- PDE Runtime;

- the Graphical Editing Framework.

In order to create and configure a target platform, the developer needs to:

1. download the following files:

    - http://mirrors.ibiblio.org/eclipse/modeling/emf/emf/downloads/drops/
      2.7.2/R201201300943/emf-xsd-Update-2.7.2.zip

    - http://mirrors.ibiblio.org/eclipse/modeling/mdt/uml2/downloads/drops/
      3.2.1/R201109132310/mdt-uml2-Update-3.2.1.zip

    - http://repository.grepcode.com/java/eclipse.org/3.7.2/plugins/com.
      ibm.icu_4.4.2.v20110823.jar

2. start eclipse;

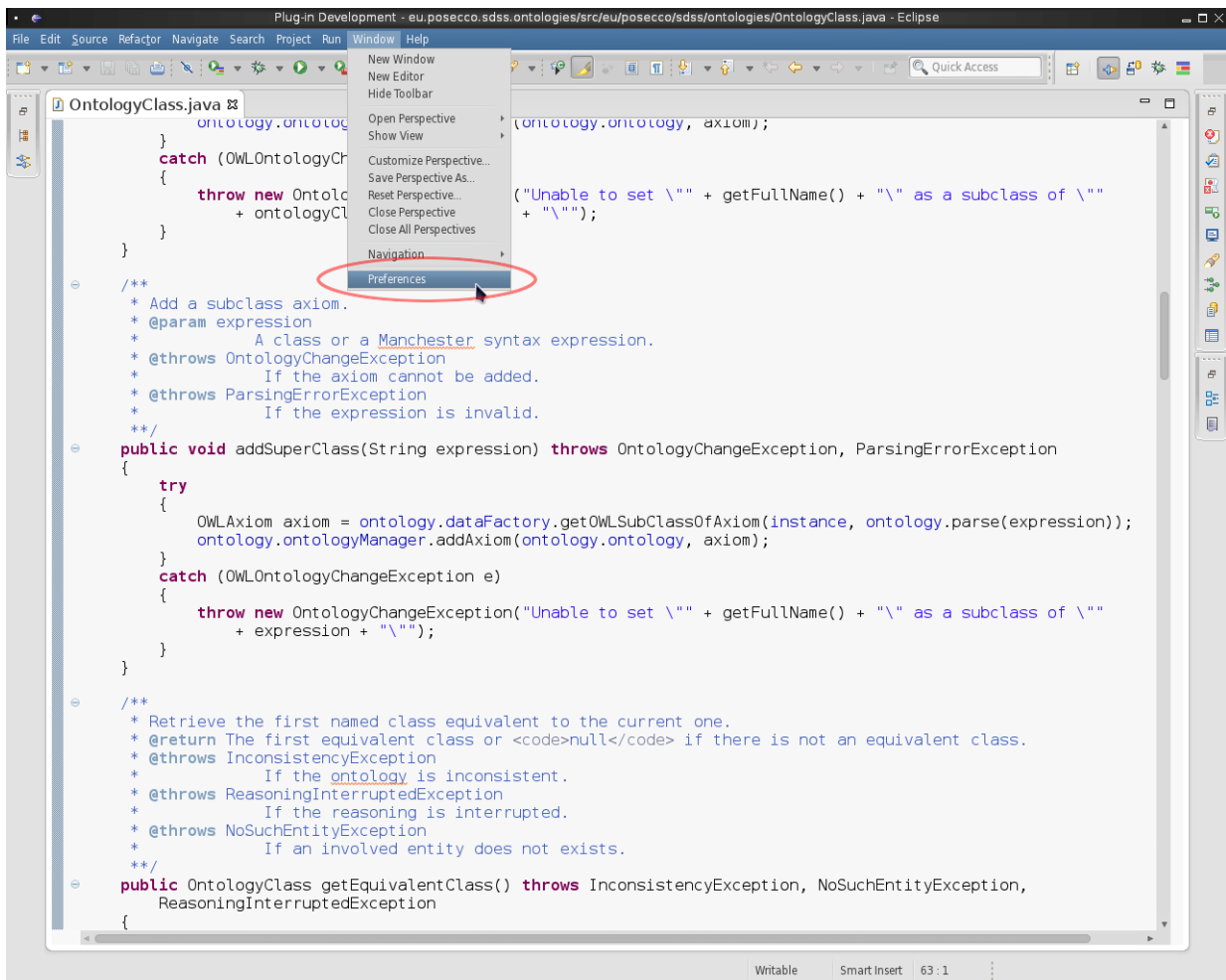3. choose Window → Preferences, as show in Figure 2;

Figure 2: The Preferences menu item.

4. in the newly opened Preferences window, choose the Plug-in Development → Target Platform tab
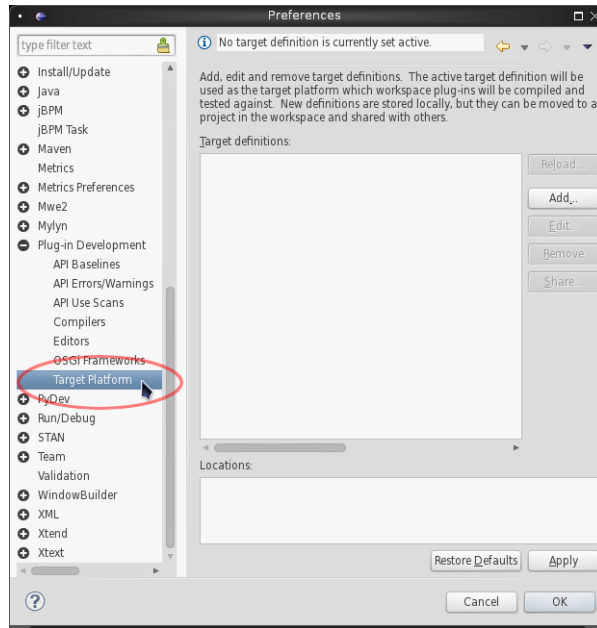   on the right, as show in Figure 3;

Figure 3: The Preferences window.

5. click on the Add... button in the Preferences window to open the wizard New Target Definition, as show in Figure 4;
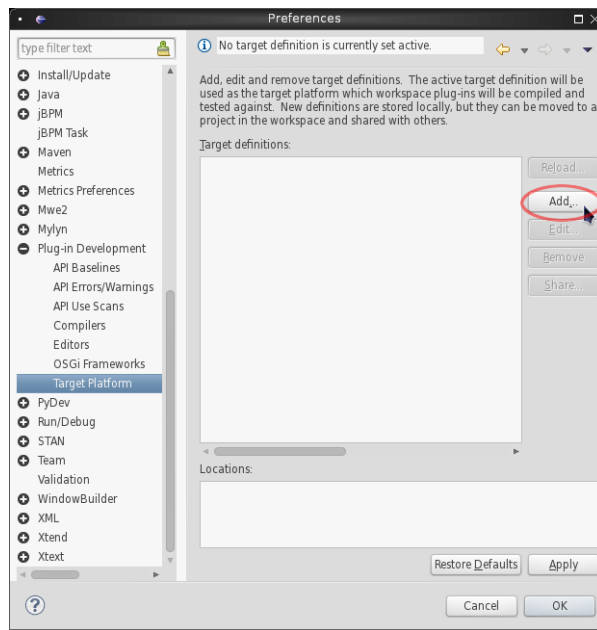

Figure 4: Adding a new target platform.

6. select Nothing: Start with an empty target definition and click on Next >, as show in Figure 5;
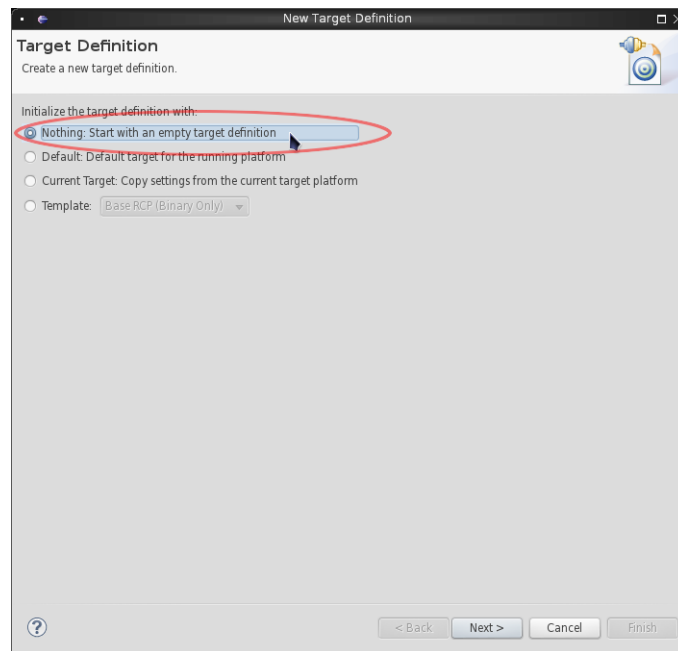
Figure 5: The New Target Definition wizard.

7. in the Target Content page that should appear, insert a suitable name in the Name text field, then click on the Add button several times, as show in Figure 6, in order to add the following software sites:

- http://download.eclipse.org/releases/kepler
  - Equinox Target Components
  - RAP Basic Equinox Target
- http://download.eclipse.org/rt/rap/2.2
- http://download.eclipse.org/birt/update-site/4.3
  - BIRT 4.3 Charting SDK (a,c)
  - BIRT 4.3 Engine OSGi Runtime SDK (a,b)
  - BIRT 4.3 WTP Integration SDK (a,b,c,d)
- http://download.eclipse.org/rt/rap/incubator/nightly/;
  - RAP File Upload Support
  - RAP GEF Port
  - RAP PDE
  - RAP CNF
  - FILEDIALOG
  - FILEUPLOAD
- emf-xsd-Update-2.7.2.zip
  - EMF Ecore Mapping
  - EMF RAP Target Components
- mdt-uml2-Update-3.2.1.zip
  - UML2 Extender SDK

8. Extract com.ibm.icu_4.4.2.v20110823.jar and the directory to the target platform.
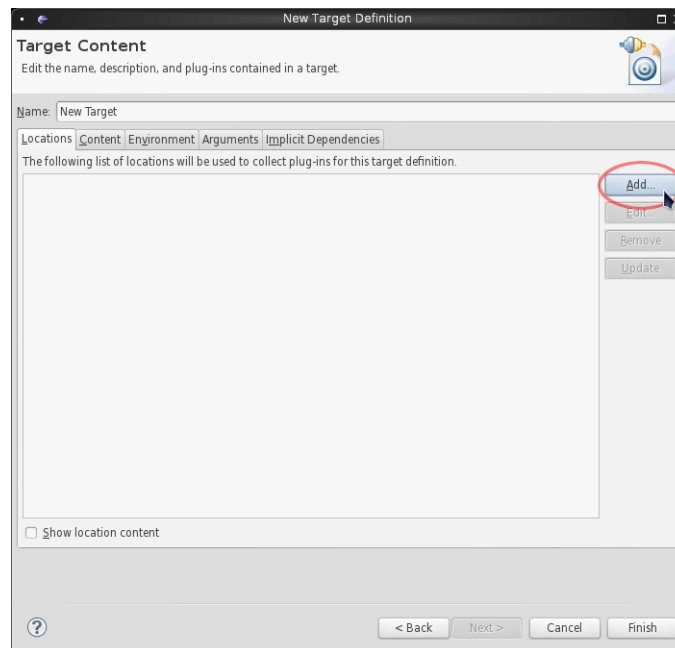
Figure 6: The Target Content wizard.

9. Click on Finish, then make sure that the newly created target platform is selected in the Preferences window, as show in Figure 7, and click on the OK button.
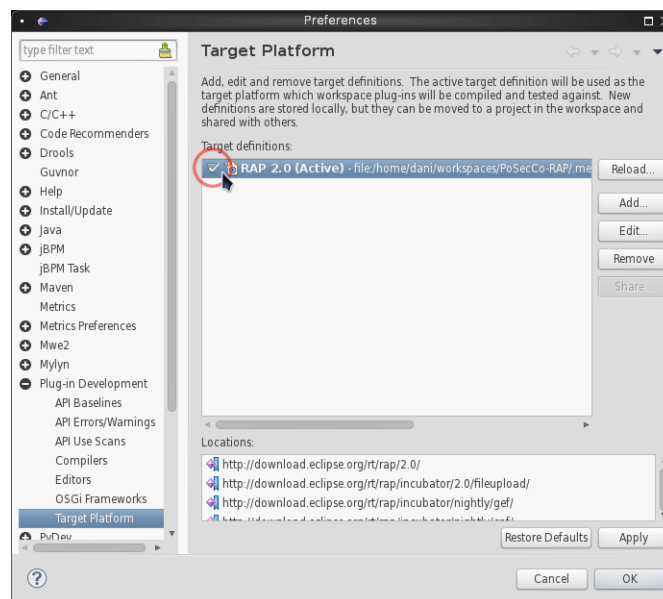

Figure 7: Selecting the active target platform.

Importing the SDSS code

Once the source code of the Security Decision Support System has been acquired, it must be inserted into an Eclipse workspace with a proper target platform configured, as explained in Section 2.

To add the SDSS source code in the current workspace:

1. select File → Import... from the menu of the Eclipse IDE, as show in Figure 8;
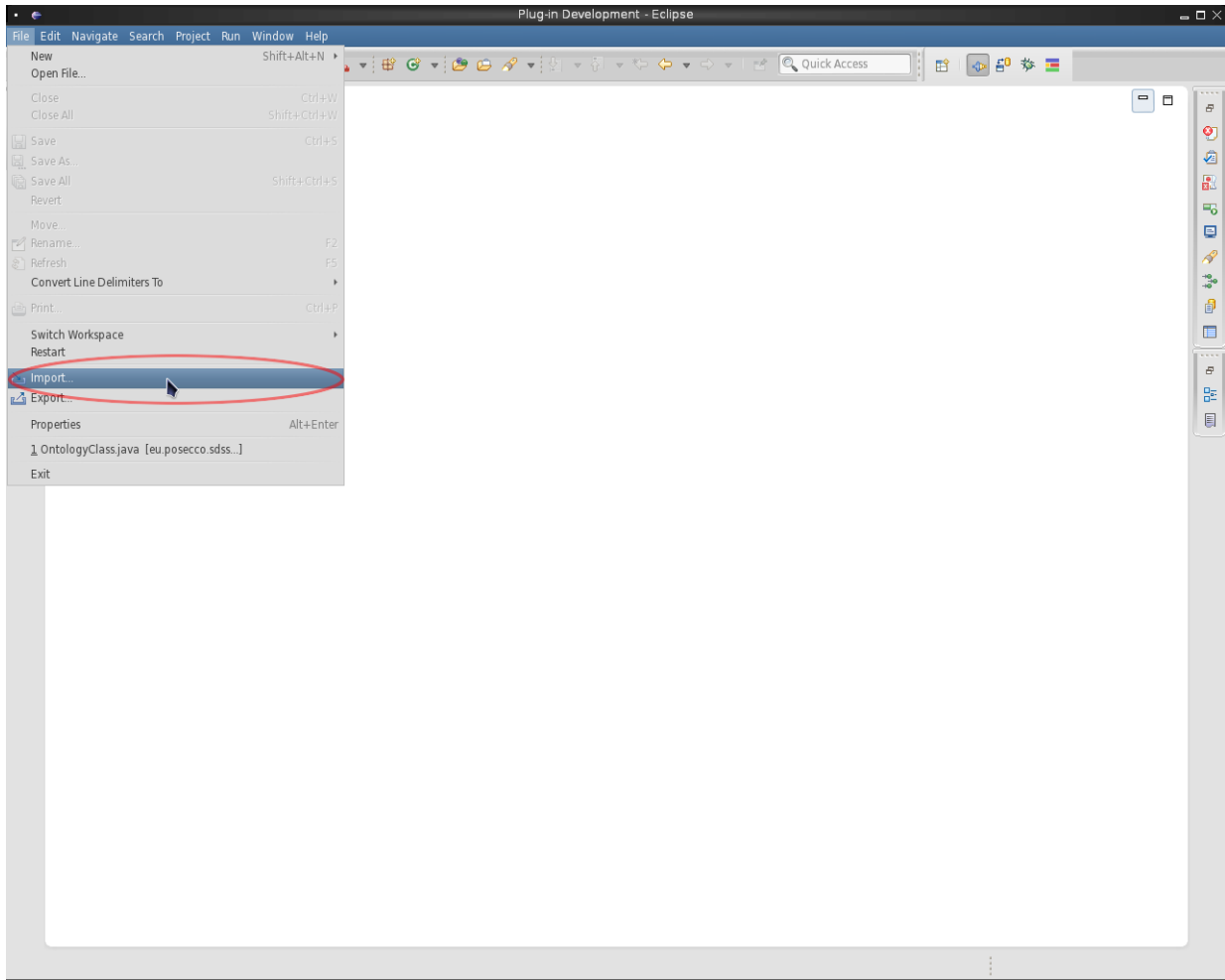
Figure 8: The Import menu item.

2. select Existing Projects into Workspace from the list in the newly opened wizard, as show in Figure 9;
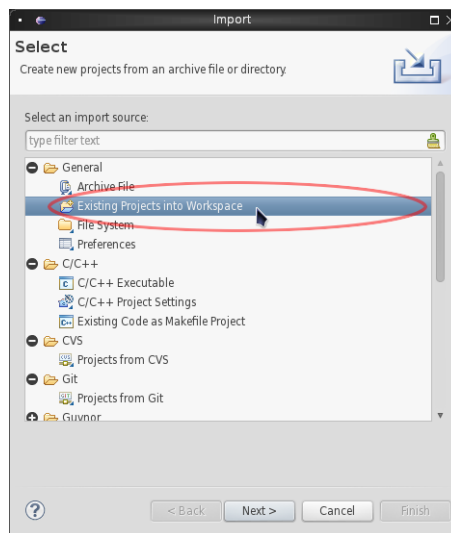

Figure 9: The Import wizard.

3. Insert the directory where the SDSS code resides in the Select root directory text field, then select all the listed plug-ins by clicking on the Select All button, as show in Figure 10, and click on Finish to import the code.
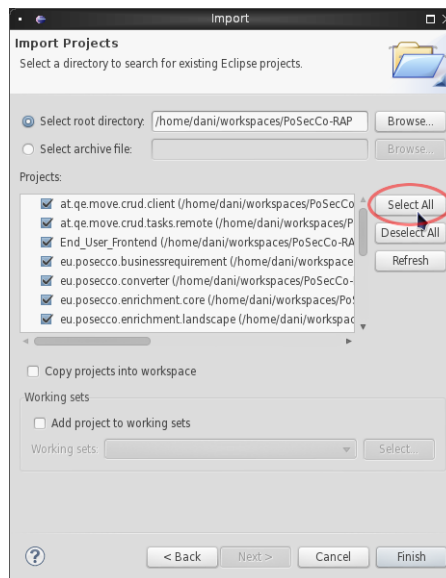
Figure 10: Importing all the plug-ins.

It is also needed to import third party libraries. The following third party libraries are used:

- Commons JXPath 1.3, available at `http://mirror.nohup.it/apache//commons/jxpath/binaries/commons-jxpath-1.3-bin.zip`

- Jgraph 5, available at `https://github.com/jgraph/legacy-jgraph5/blob/62379bbef721b3271158dd9f3adea2b693676e9b/lib/jgraph.jar?raw=true`

- Jgrapht 0.8.3, available at `http://sourceforge.net/projects/jgrapht/files/JGraphT/Version%200.8.3/jgrapht-0.8.3.zip/download`

- OWL API 3.2.4, available at `http://sourceforge.net/projects/owlapi/files/OWL%20API%20%28for%20OWL%202.0%29/3.2.4/owlapi-3.2.4.zip/download`

- SPARQL-DL API 1.0.0, available at `http://www.derivo.de/fileadmin/externe_websites/ext.derivo/content/SPARQL-DL/sparqldl-api1.0.0.zip`

The following reasoners are supported (at least one is required):

- Pellet 2.3.0, available at `http://clarkparsia.com/pellet/download/pellet-2.3.0`

- Hermit 1.3.6, available at `http://www.hermit-reasoner.com/download/1.3.6/HermiT.zip`

The following mathematical solvers are supported (at least one is required):

- lp_solve 5.5.2.0, available at:

  - `http://sourceforge.net/projects/lpsolve/files/lpsolve/5.5.2.0/lp_solve_5.5.2.0_dev_win32.zip/download` (Win32 development binaries)

  - `http://sourceforge.net/projects/lpsolve/files/lpsolve/5.5.2.0/lp_solve_5.5.2.0_dev_win64.zip/download` (Win64 development binaries)

  - `http://sourceforge.net/projects/lpsolve/files/lpsolve/5.5.2.0/lp_solve_5.5.2.0_exe_ux32.tar.gz/download` (Unix 32 binaries)

  - `http://sourceforge.net/projects/lpsolve/files/lpsolve/5.5.2.0/lp_solve_5.5.2.0_dev_ux32.tar.gz/download` (Unix 32 development binaries)

- http://sourceforge.net/projects/lpsolve/files/lpsolve/5.5.2.0/lp_solve_5.5.2.0_exe_ux64.tar.gz/download (Unix 64 binaries)

- http://sourceforge.net/projects/lpsolve/files/lpsolve/5.5.2.0/lp_solve_5.5.2.0_dev_ux64.tar.gz/download (Unix 64 development binaries)

- http://sourceforge.net/projects/lpsolve/files/lpsolve/5.5.2.0/lp_solve_5.5.2.0_java.zip/download (Java interface)

- CPLEX, available at http://www.ibm.com/software/commerce/optimization/cplex-optimize

Configuring a RAP application

The last step to setup the development environment is to configure the Eclipse IDE to launch the SDSS on the local machine using a run configuration. To create a suitable run configuration:

1. click on the Run → Run Configurations... menu in the Eclipse IDE, as shown in Figure 11;
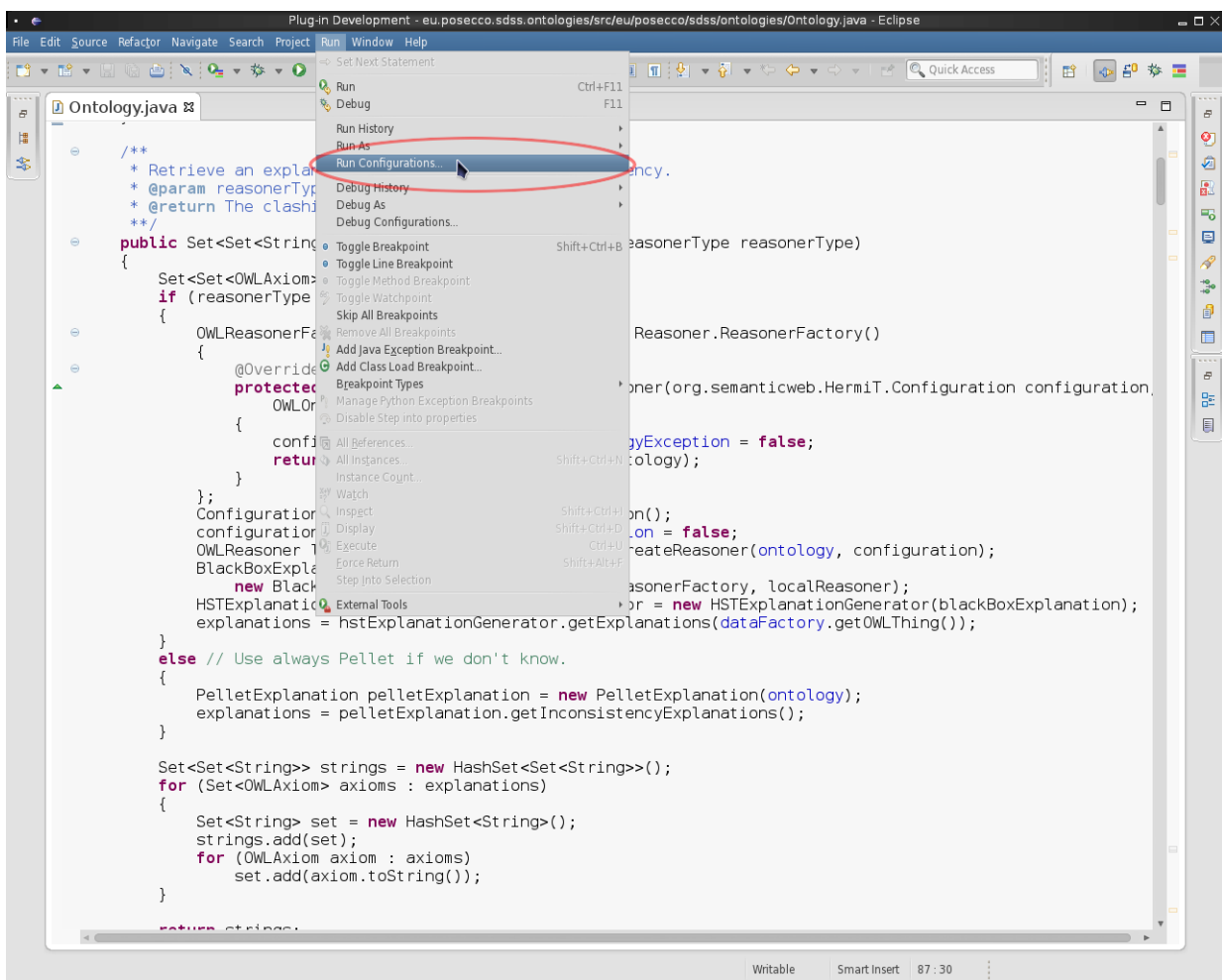

Figure 11: The Run Configurations... menu item.

2. create a new RAP Application configuration and be sure to set /projectmanager in the Servlet path text field, as shown in Figure 12;
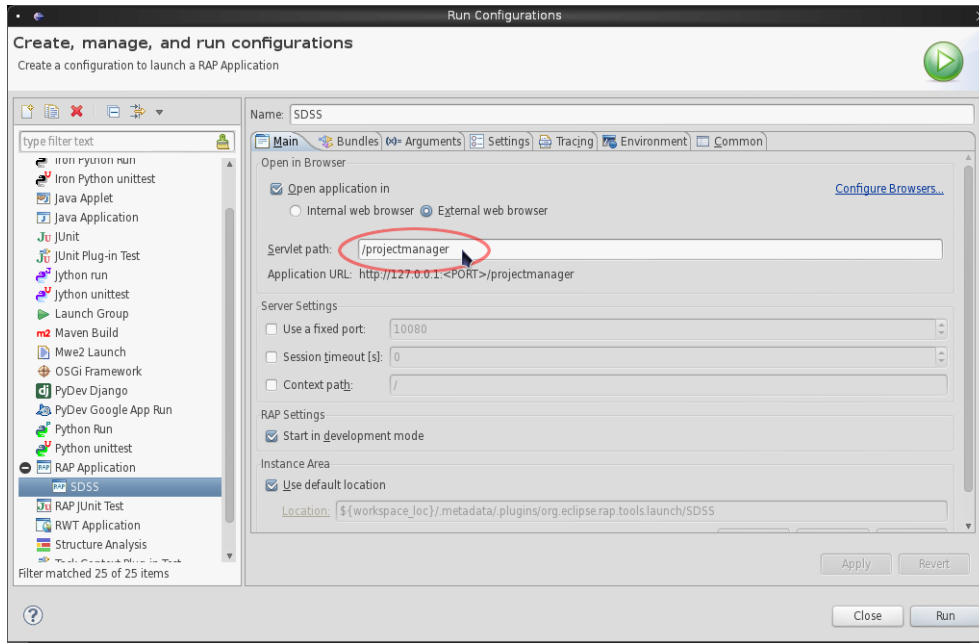
Figure 12: The Eclipse Preferences window.

3. this configuration can now be used to execute and debug the SDSS.

3 Javadoc extraction

Most of the classes, interfaces and enumerations in the SDSS are documented using a number of Javadoc comments directly embedded in the source code.

The Eclipse IDE can automatically shows the Javadoc comments for a particular type when editing a source file, however this documentation can also be extracted into a set of HTML pages.

Using the Eclipse IDE to extract the API documentation

The Eclipse IDE has a built-in function that is able to easily extract the Javadoc comments from a set of plug-ins. In order to generate the HTML pages:

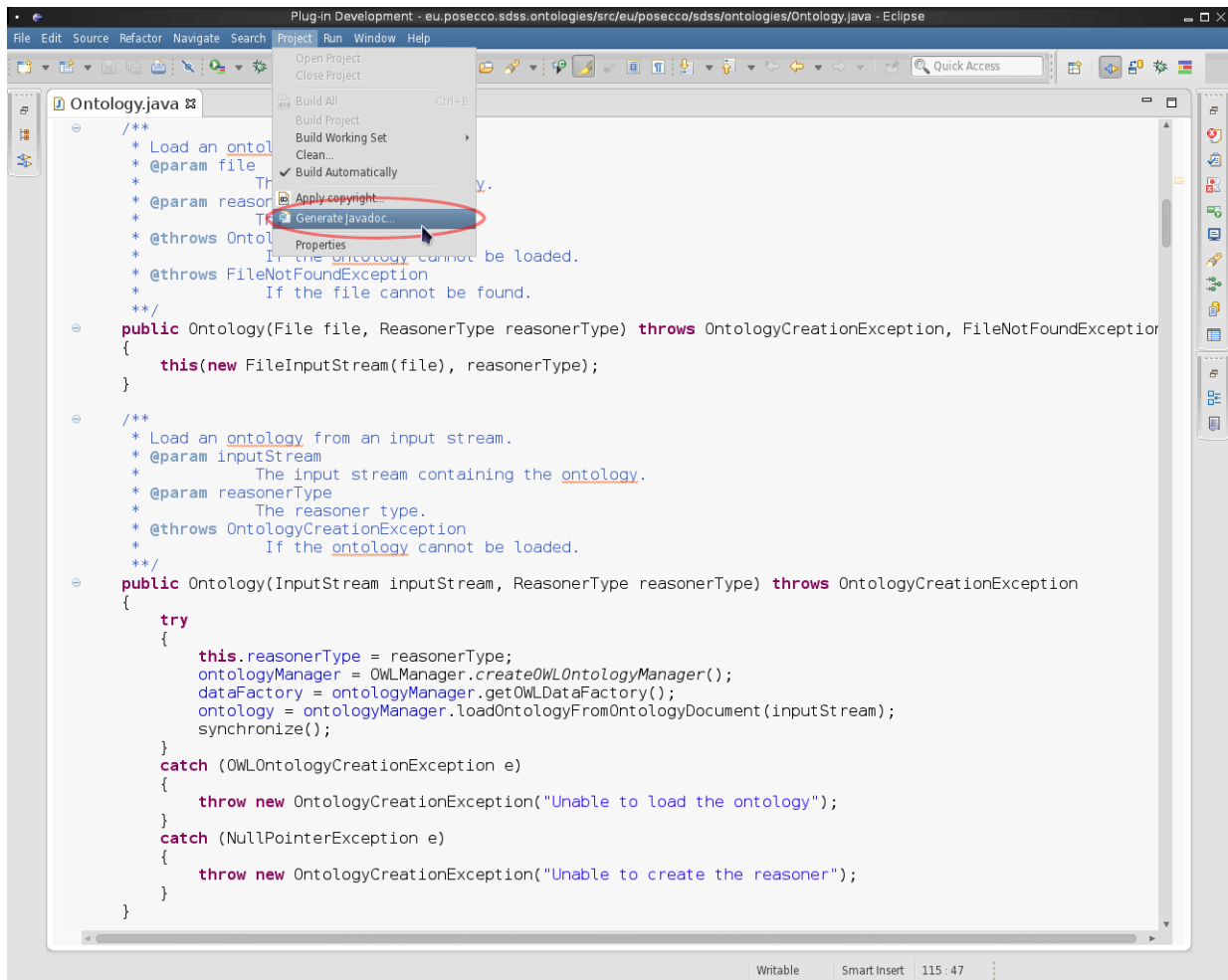1. click on the Project → Generate Javadoc... menu in the Eclipse IDE, as shown in Figure 13;

Figure 13: The Eclipse Generate Javadoc... menu item.

2. in the newly opened Generate Javadoc wizard, select the desired plug-ins from the list, put a destination directory in the Destination text field and click on Finish to generate the HTML pages, as shown in Figure 14.
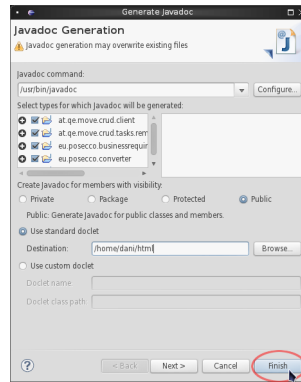
Figure 14: The Eclipse Generate Javadoc wizard.

Using the javadoc command to extract the API documentation

The basic mean of extracting the Javadoc comments is through the use of the `javadoc` command. This is a portable and universal way to generate a set of HTML pages documenting an application API.

Its basic syntax is `javadoc -d <dir> <target>` where:

- `<target>` is a list of space separated packages or classes that will be analyzed;

- `<dir>` specifies the directory that will contain the generated HTML pages.

For example, the command `javadoc -d html eu.posecco.sdss.ontologies` can be used to extract all the API documentation of the package `eu.posecco.sdss.ontologies` into a directory named `html`. Note that this command must be executed in the `src` directory of the plug-in containing the aforementioned package.

To extract the complete SDSS API documentation, a script can be used to automatize the extraction process. Under GNU/Linux, the command `javadoc -d html`find -name '*.java'` can also be executed in the SDSS workspace directory to achieve the same results.

4 Additional tools

The Eclipse IDE is the primary tool used to develop and extend the SDSS, but the Security Decision Support System does not consist only of Java code, so a number of other tools are useful to maintain it in its entirety, such as:

Protégé (http://protege.stanford.edu/)
a powerful plug-in based editor for creating and manipulating ontologies implemented with the OWL technology

A   Troubleshooting

This section includes several frequently asked questions related to the SDSS development cycle.

Q: After that the SDSS is launched, the web browser reports a 404 Not found error.

   A: Eclipse takes a while before initializing its internal RAP server. If the web browser is executed before this setup time, this error message can arise. Wait a few seconds and try to reload the page to solve the problem.