# Building OWASP ZAP Using Eclipse IDE for Java… Pen-Testers

**Author:** Raul Siles (raul @ taddong.com)
Taddong – www.taddong.com
**Version:** 1.0
**Date:** August 10, 2011

This brief guide details the process required to build the OWASP Zed Attack Proxy (ZAP)[1] from source code using the Eclipse IDE for Java Developers. The guide has been developed for Windows 7 but the Eclipse build process should be very similar for other operating systems, such as Linux.

The official version of this document is available in PDF format for easy download, distribution, and usage at Taddong's Lab: http://www.taddong.com/en/lab.html.

The original guide detailing how to build ZAP is available from the OWASP ZAP project Wiki: http://code.google.com/p/zaproxy/wiki/Building.

## Software Requirements

The following software is required to gather the latest OWASP ZAP (ZAP from now on) source code revision from the official SVN repository, and build a working copy of ZAP using Eclipse:

- Base operating system: Windows 7 (64 bits)
- TortoiseSVN Windows client: http://tortoisesvn.net (v1.6.16.21511 – 64 bits)
- OWASP ZAP SVN repository: http://code.google.com/p/zaproxy/source/checkout

```
svn co http://zaproxy.googlecode.com/svn/trunk zaproxy-read-only
```

- Eclipse IDE for Java Developers Indigo (v3.7 – 32 bits) [2]:
  http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/indigor
- Java SE 6 Update 26 (JDK 6.0.26 - 32 bits):
  http://www.oracle.com/technetwork/java/javase/downloads/

_NOTE_: The Java Development Kit (JDK) and Eclipse must use the same version architecture, 32 or 64 bits, independently of the OS version used. The proposed environment makes use of the 32 bits version of Java and Eclipse over Windows 7 64 bits, but other combinations might work.

_If you find any improvement to this guide, or have any related comment to make it a more valuable resource for the web application development and pen-testing communities, do not hesitate to contact me and contribute to future versions of this "Building OWASP ZAP" guide._

---

[1] https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
[2] Older versions of Eclipse: http://wiki.eclipse.org/Older_Versions_Of_Eclipse

## Installing the Software

Install TortoiseSVN and the (Java) JDK following the default Windows installation. Install Eclipse by uncompressing the official ZIP file and placing the "eclipse" folder in the selected destination location, such as "C:\eclipse".

> *NOTE*: The usage of TortoiseSVN is out of the scope of this guide. Use TortoiseSVN, or your preferred SVN client (such as "svn"), to retrieve a copy of the latest ZAP source code revision, available at the Google Code URL detailed on the "Software Requirements" section.

## Starting Eclipse



Run "C:\eclipse\eclipse.exe". The first time Eclipse starts it asks the user to configure the workspace folder:
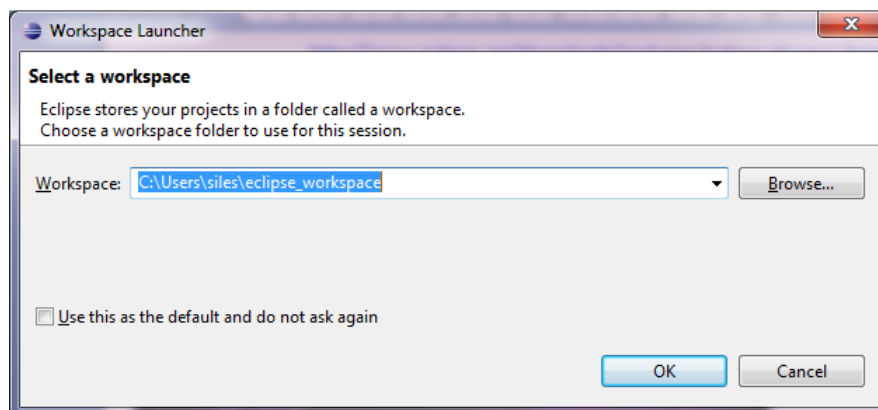


Enter your workspace folder (e.g. "C:\Users\siles\eclipse_workspace") and check the "Use this as the default and do not ask again" option so that Eclipse does not ask for the workspace folder every time it runs. In the "Welcome" screen, click on the right arrow to go to the workbench:

NOTE: You can always go back to the "Welcome" screen from the "Help – Welcome" menu.

## Checking the Java Environment (JRE or JDK) Detected by Eclipse

Go to the "Window – Preferences" menu to open the workbench preferences. Select the "Java – Installed JREs" page to confirm the Java environment has been detected by Eclipse and it is available:
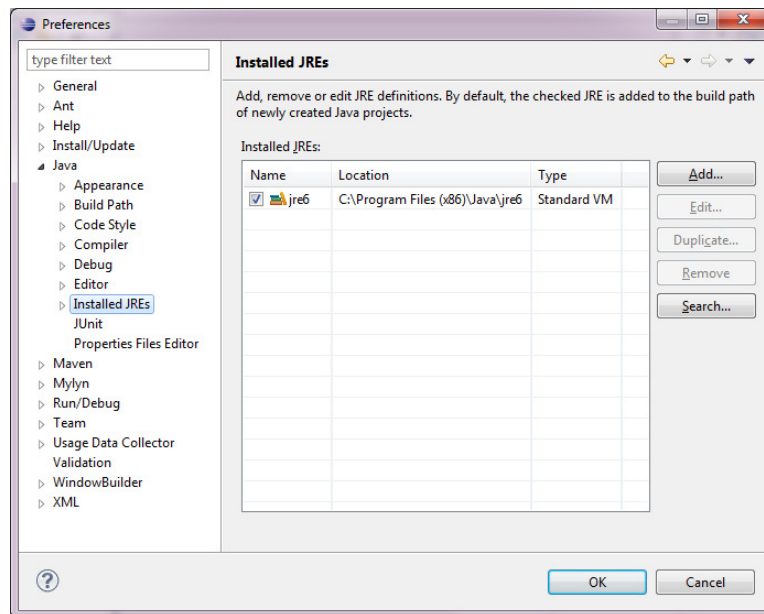


It is recommended to replace the JRE by the JDK if both are installed on the system. If only the JDK is installed, confirm it has been properly detected. Using the "Add…" or "Search…" buttons locate the JDK folder and add it. Uncheck (first screenshot below), or preferably "Remove" (second screenshot below), the original JRE entry (if both are available):
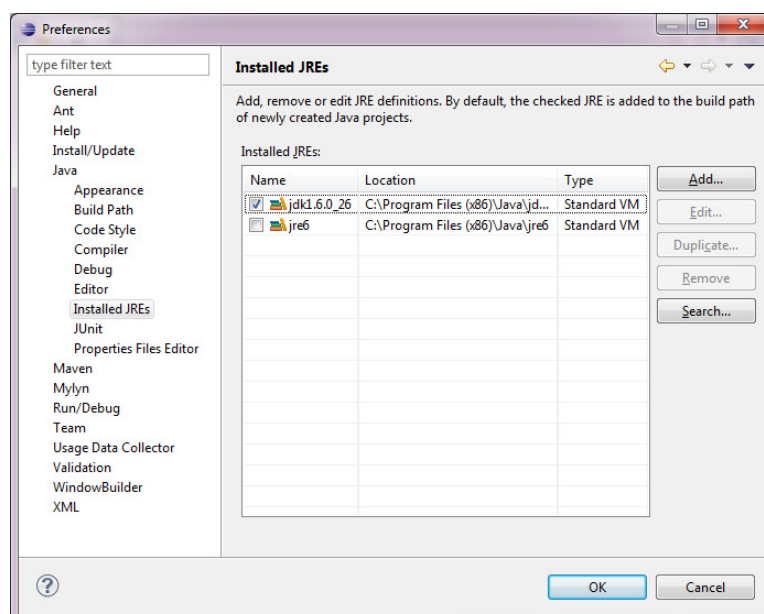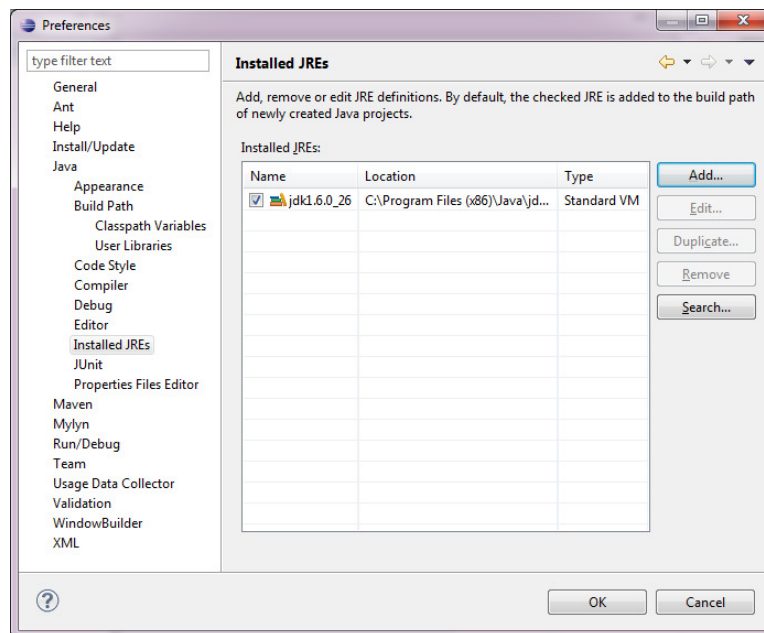
*NOTE*: The Eclipse Indigo help is available at: http://help.eclipse.org/indigo/index.jsp.

## Creating the ZAP Project in Eclipse

*NOTE*: Remember, as a prerequisite, use TortoiseSVN or any other SVN client, to download the latest ZAP source code revision from the official ZAP SVN repository to a local directory (e.g. "C:\Users\siles\SVN\zaproxy_20110721_r829").

Using the "File – New – Java Project" menu, select a project name (e.g. "zaproxy") and click "Finish".
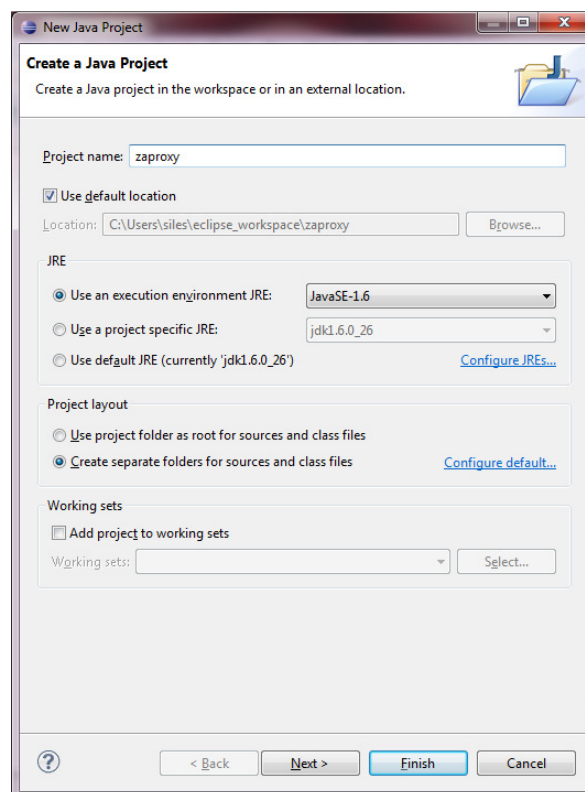
From the "Package Explorer", select the "zaproxy" project, right click and select "Import…" From the "Import" window, select "General – File System" and click "Next":
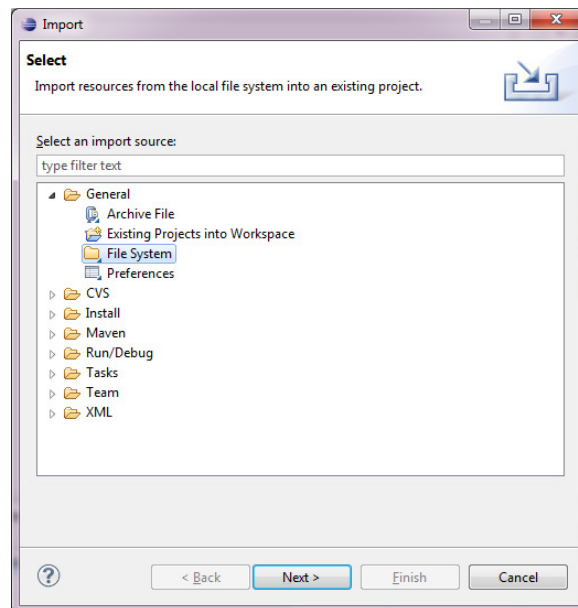


From the "File system" window, browse (using the "From directory:" option) to the directory where the ZAP source code revision has been saved. Expand the directory (e.g. "zaproxy_20110721_r829") and select all its contents, that is, the four "build", "lib", "src", and "wave" folders. Click "Finish".
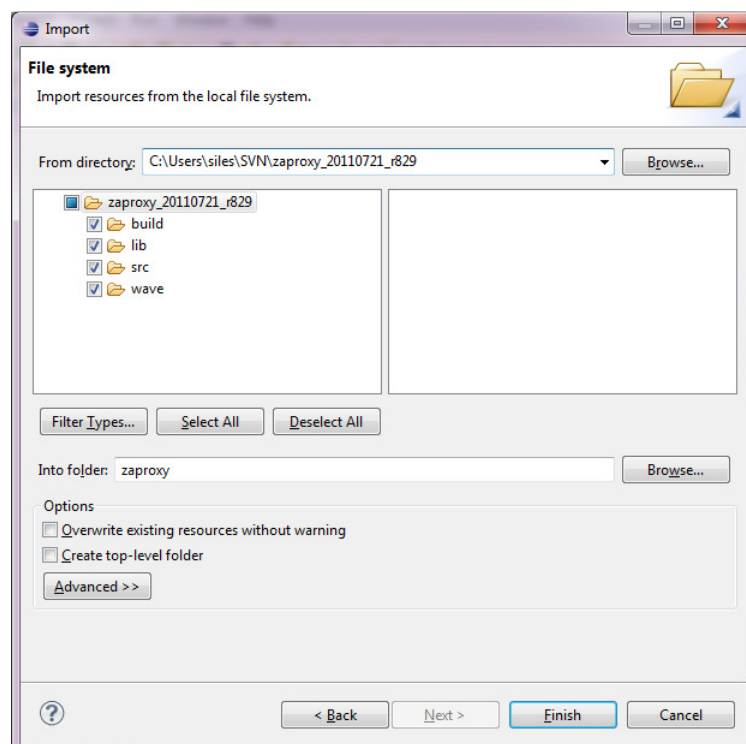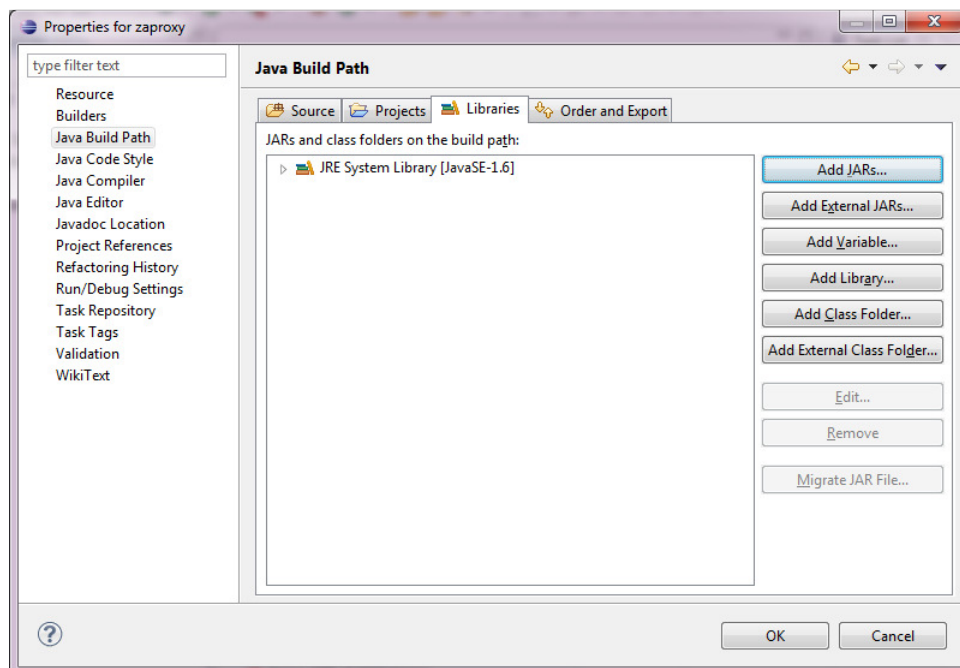


As a result, all the ZAP project contents (the four folders previously listed) are imported inside the Eclipse project:
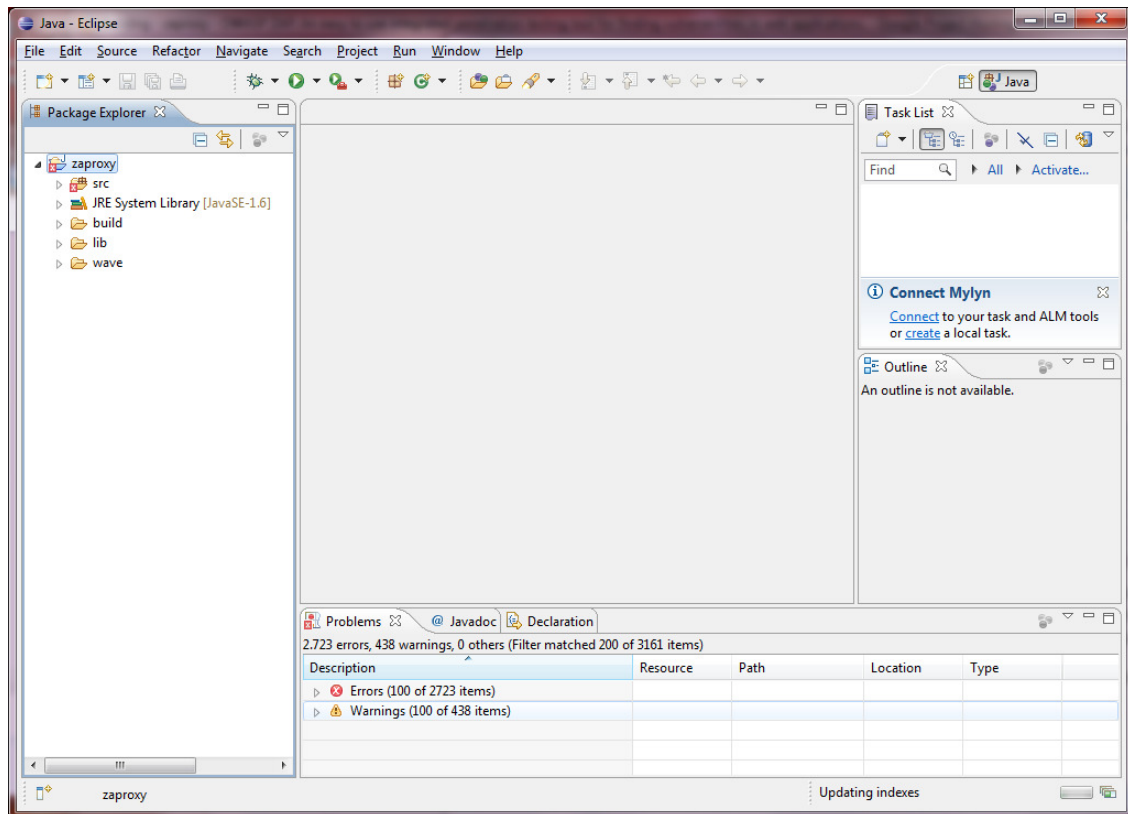
The "Problems" tab (at the bottom) will reflect several import errors due to the absence of the ZAP project libraries.

## Adding the ZAP Project Libraries to the Build Path

Go to the "Project – Properties" menu and select the "Java Build Path" page:

From the "Libraries" tab, use the "Add JARs…" button to include all the ZAP project libraries (.jar files). Form the "JAR Selection" window, expand the "zaproxy" Eclipse project, and manually select all the JAR files inside the "lib" folder. Click "OK".
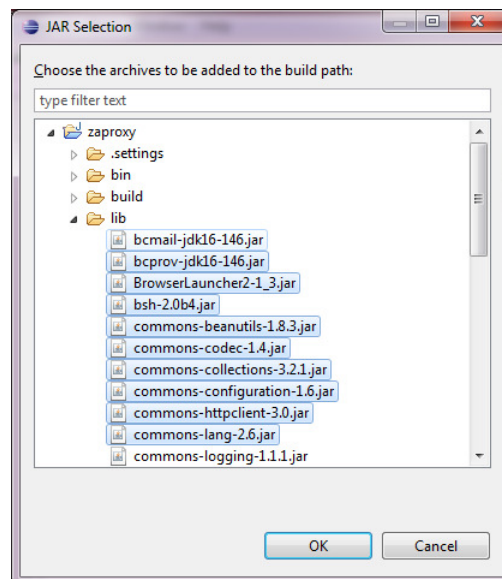


As a result, all the ZAP project JAR files are added to the build path. Click "OK".
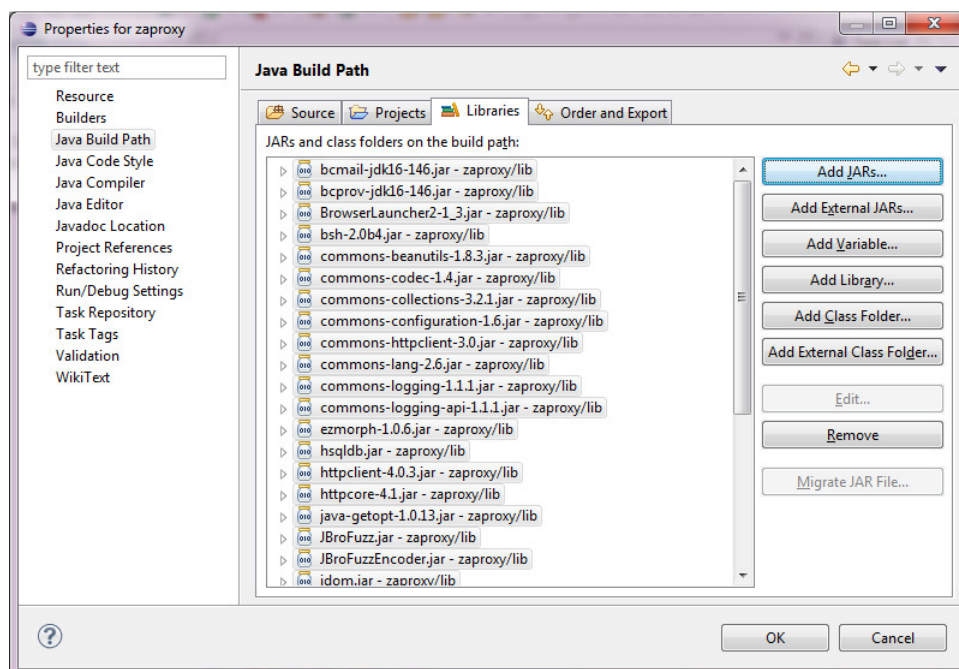


## Configuring the ZAP Runtime Environment in Eclipse

To be able to run ZAP from within Eclipse, both the Eclipse IDE working directory and the ZAP project's output directory must match. From the (previous) "Project – Properties" menu and the "Java Build Path" page, go to the "Source" tab (accept the unsaved modifications warning, if any, by clicking on "Apply"):
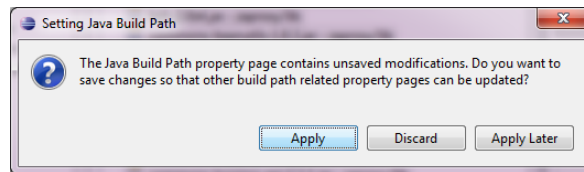
At the bottom of the "Source" configuration page you will find the "Default output folder" value. Record this value for a further step (e.g. "zaproxy/bin"). Click "OK".
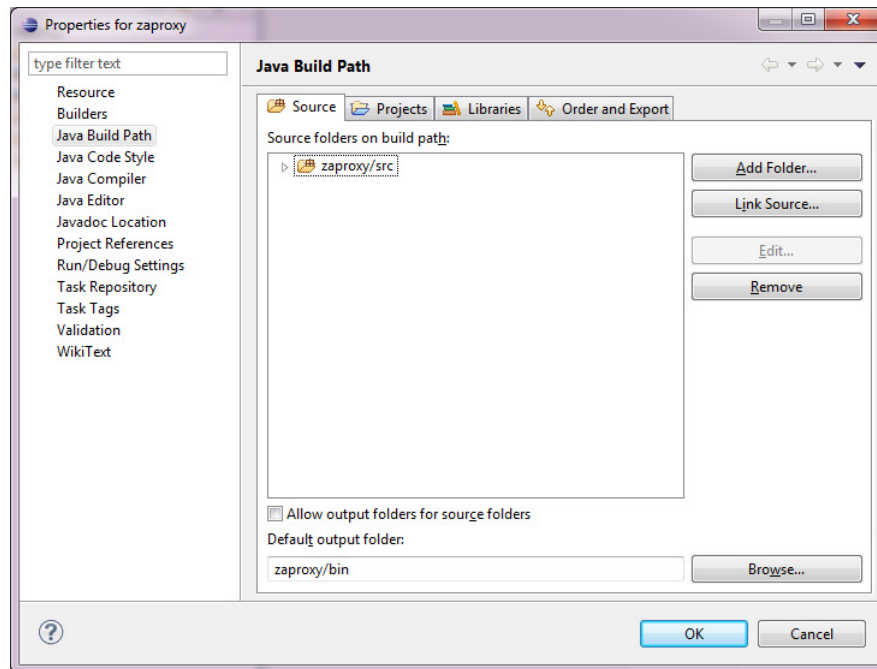


Go to the "Run – Run configurations…" menu. Select "Java Application" and press the "New" button:
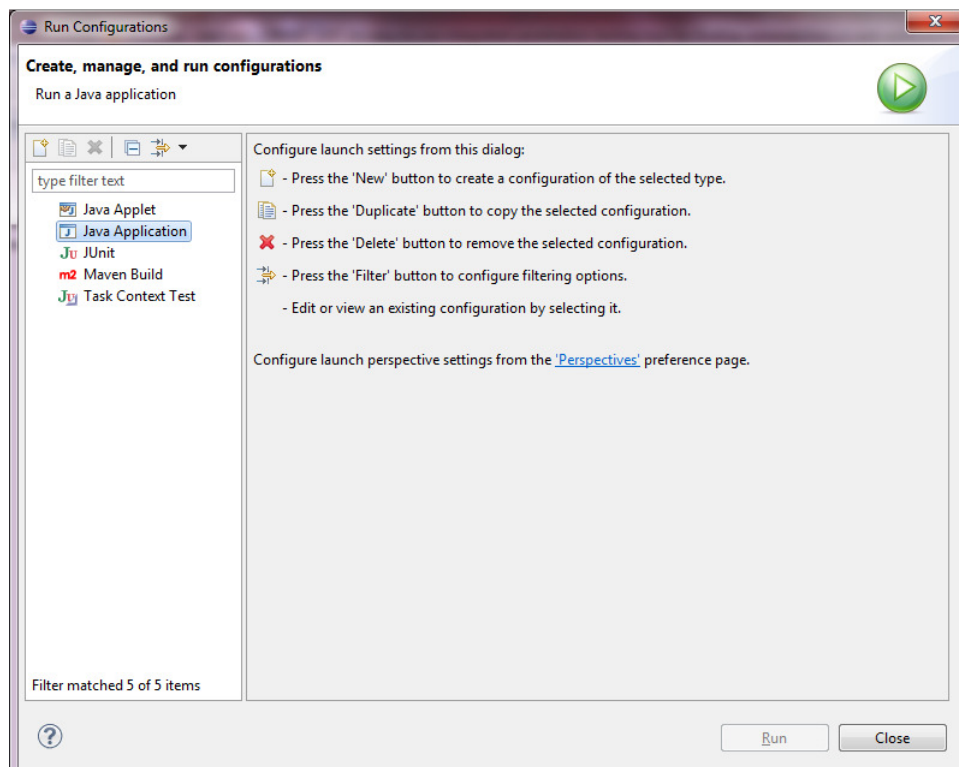
Change the name of the new run configuration to "ZAP" and, on the "Main" tab, define the ZAP project main class using the "Main class:" field (e.g. "org.zaproxy.zap.ZAP").
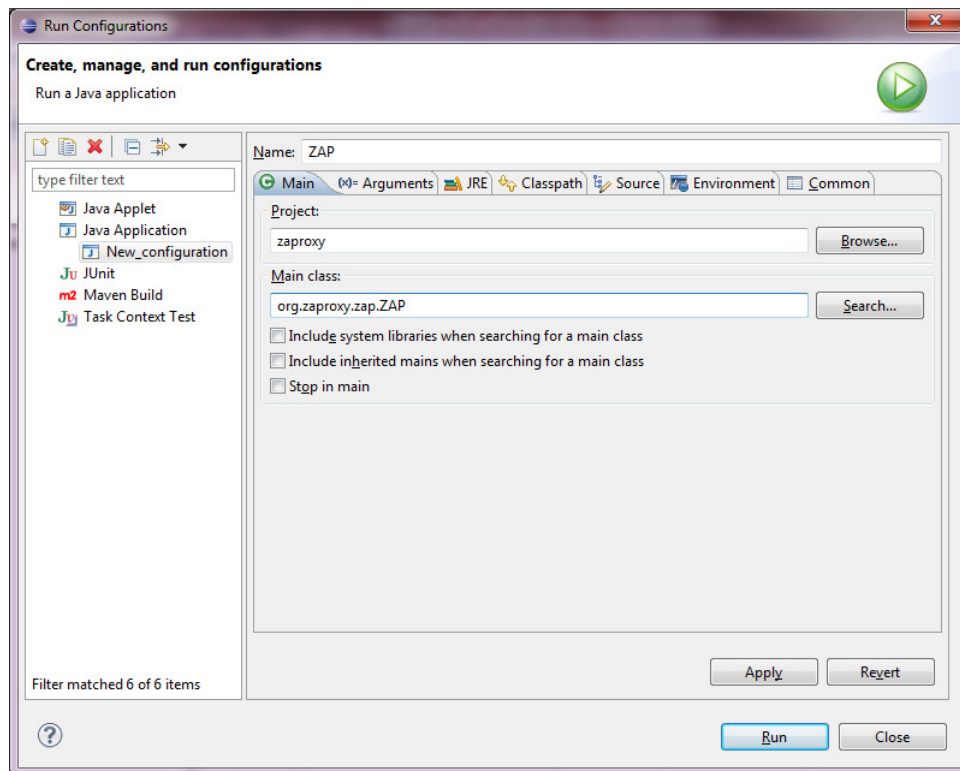


Go to the "Arguments" tab and change the working directory at the bottom to the ZAP project's output directory previously recorded, using this format: "${workspace_loc:zaproxy/bin}".
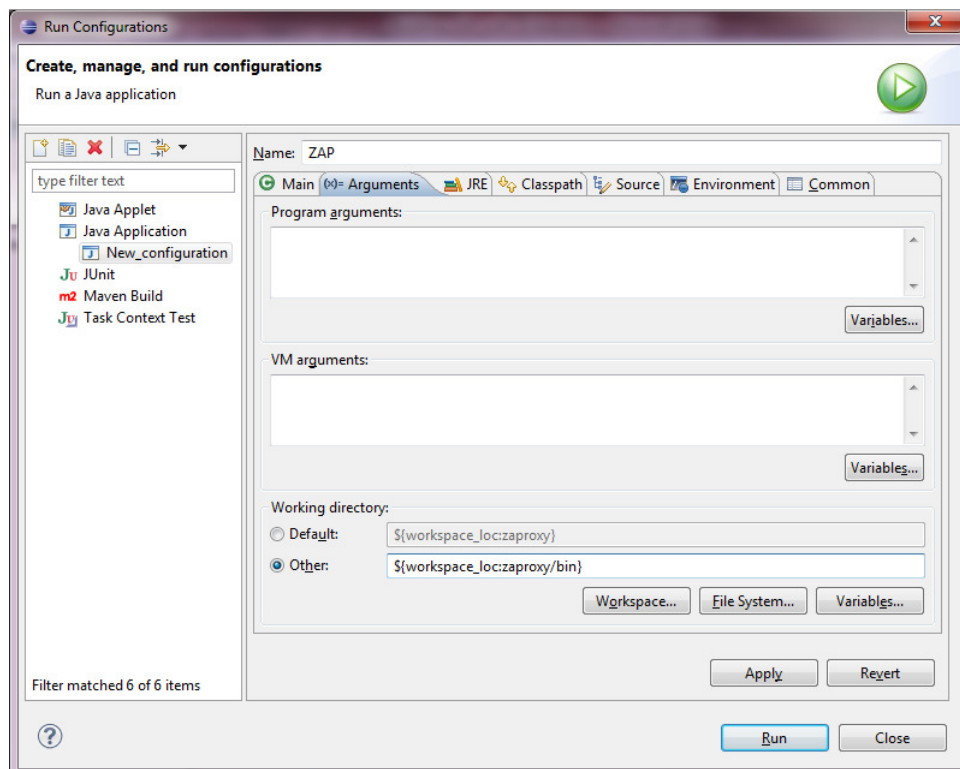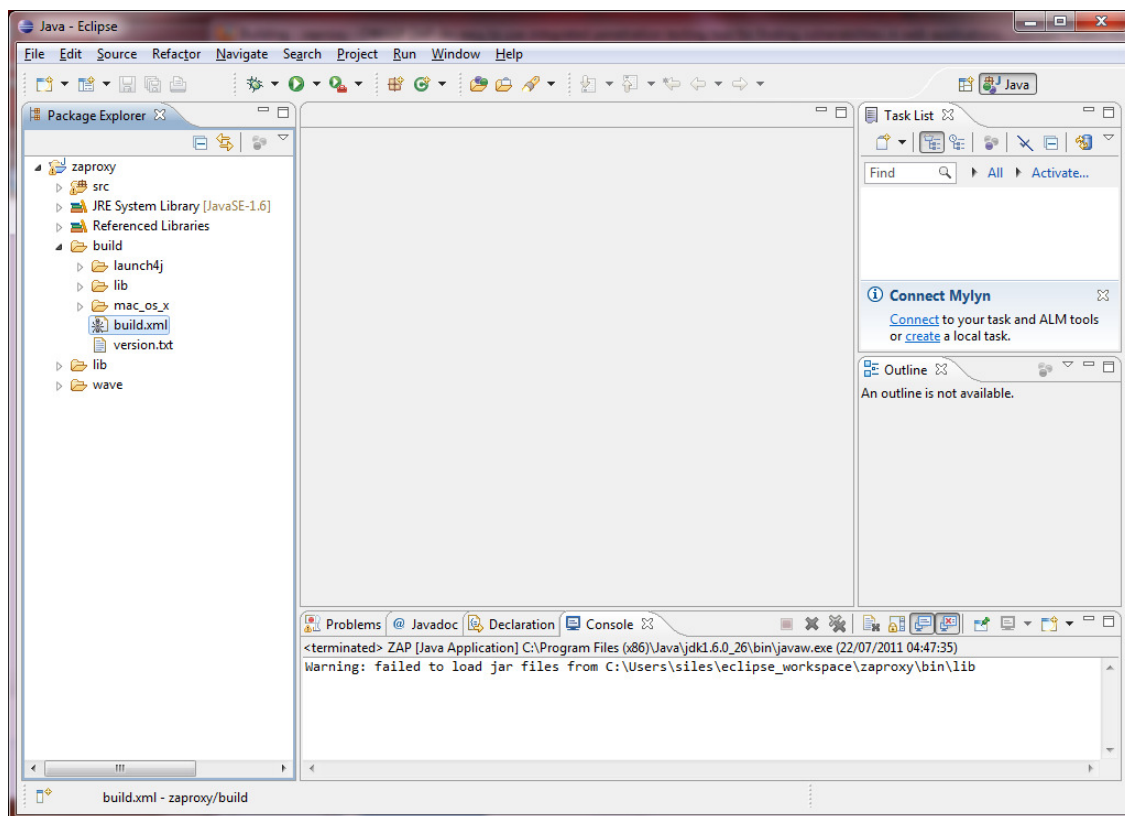
Click "Run" and the Eclipse "zaproxy" project should compile and run from within the Eclipse IDE (ZAP is using the Spanish language pack in the screenshot below):
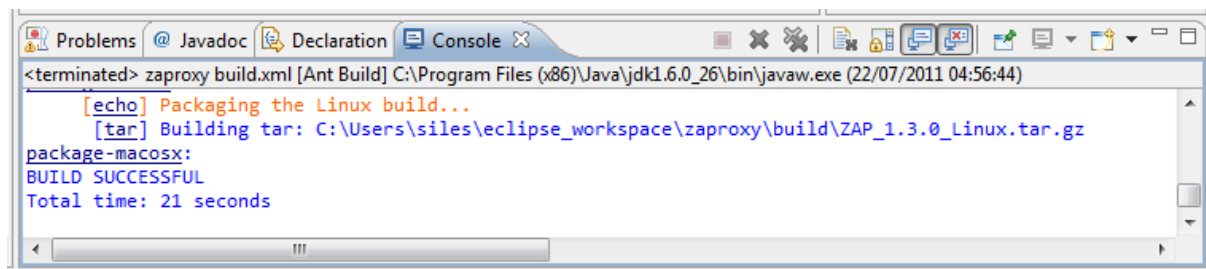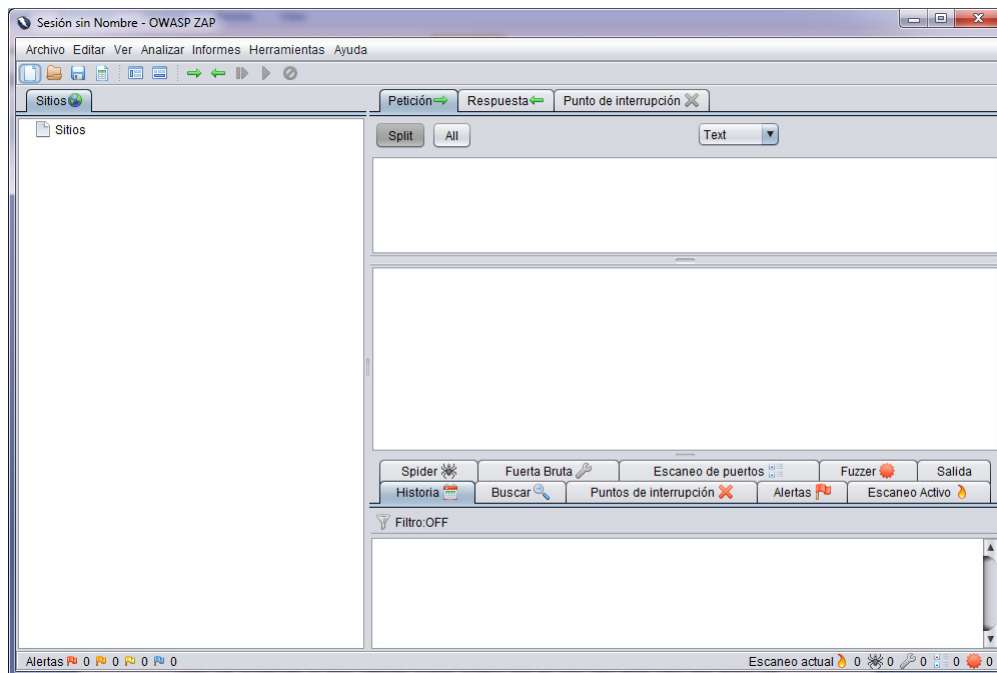


## Building ZAP

To build your own ZAP version, so that you are able to package, distribute and test your own ZAP version, go to the "Package Explorer" tab and select the "zaproxy – build – build.xml" file:

Right click on the "build.xml" file and select "Run As - Ant Build". Eclipse will compile the ZAP project and generate the ZAP.jar file under the local workspace "zaproxy\build\zap\" directory, as indicated by the Eclipse console at the bottom:



The full contents of the build directory (e.g. "C:\Users\siles\eclipse_workspace\zaproxy\build\zap\") can be packed in a ZIP file, distributed and tested (or you can modify the "build.xml" file to do it automatically).

This "zap" folder is a self contained ZAP project package ready to run, on both Linux ("zap.sh") and Windows ("zap.bat"). Simply unzip the contents of the ZIP file on any system with a Java Runtime Environment (JRE) and execute the corresponding shell or batch file.

# Appendix: Contributing to ZAP

If you plan to contribute to the ZAP project, by adding new features or capabilities to this tool or fixing known issues [3], the ZAP project lead (*psiinon*) recommends to use Subclipse (http://subclipse.tigris.org), a plug-in providing support for Subversion (SVN) within the Eclipse IDE, to easily integrate the current SVN ZAP revision in Eclipse and be able to manage source code changes from within Eclipse.

The current Subclipse download and install instructions from within the Eclipse IDE (for v3.0.2 on Windows XP) are available at http://subclipse.tigris.org/servlets/ProjectProcess?pageID=p4wYuA.

To briefly sum up the installation process for the current Eclipse IDE version Indigo (v3.7), go to the "Help – Install New Software…" menu. Enter the following URL for Subclipse 1.6.x (Eclipse 3.2+) into the "Work with" field, http://subclipse.tigris.org/update_1.6.x, and click the "Add..." (Repository) button. Select a name for the repository, such as "Subclipse", and click "OK".

Eclipse will contact the Subclipse repository and display all the available features on the intermediate pane. Select the Subclipse checkbox to select all the associated features, or select them individually, and click "Next".

Eclipse will guide you through the installation process, providing first the install details. Click "Next", accept the license agreements and click "Finish". The Subclipse software will be installed and, once finished, it is required to restart Eclipse for the installation changes to take effect.

After restarting Eclipse you have to define the ZAP SVN repository from the "Window – Open Perspective – Other" menu, by selecting the "SVN Repository Exploring" option and clicking "OK". From the "SVN Repositories" tab, right click and select "New – Repository Location…". Enter the ZAP SVN repository ("http://zaproxy.googlecode.com/svn/trunk" or an alternative branch) in the "Url" field and click "Finish". As a result, the ZAP SVN repository is available from Eclipse.

Other Subclipse options and settings can be managed from the "Window – Preferences" menu, by selecting the "Team – SVN" page. It is recommended to review the Subclipse documentation and help to learn how to use Subclipse within Eclipse, available from the "Help – Help Contents" menu, under the "Subclipse – Subversion Eclipse Plugin" section.

---

[3] OWASP ZAP Issues: https://code.google.com/p/zaproxy/issues/list