Building OWASP ZAP Using Eclipse IDE for Java... Pen-Testers

Author: Raul Siles (raul @ taddong.com) Taddong – <u>www.taddong.com</u> Version: 2.0 Date: February 9, 2012



This 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 or Mac OS X.

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

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

<u>Changelog</u>: v2.0 (from version 1.0 of this guide)

- Significant document update, including:
 - New requirements: Subclipse & Software updates (Eclipse and Java JDK).
 - Removed requirement for TortoiseSVN (or any other external SVN client).
 - New sections: "Installing Subclipse", "Adding the ZAP SVN Repository to Eclipse", "ZAP Project Development", "Updating the Local ZAP Working Copy from SVN using Eclipse", "Committing Your Local ZAP Changes to SVN using Eclipse", "OWASP ZAP Extensions" and "Appendix: Java Version Updates and Eclipse".
 - Removed appendix from version 1.0.

¹ https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project



Contents

Software Requirements
Installing the Software4
Starting Eclipse4
Checking the Java Environment (JRE or JDK) Detected by Eclipse5
Installing Subclipse
Adding the ZAP SVN Repository to Eclipse7
Creating the ZAP Project in Eclipse7
Adding the ZAP Project Libraries to the Build Path10
Configuring the ZAP Runtime Environment in Eclipse11
Building ZAP14
ZAP Project Development15
Updating the Local ZAP Working Copy from SVN using Eclipse15
Committing Your Local ZAP Changes to SVN using Eclipse16
OWASP ZAP Extensions
Appendix: Java Version Updates and Eclipse19



Software Requirements

The following software is required to obtain 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)
- Java Development Kit (JDK) 6 Update 30 32 bits: <u>http://www.oracle.com/technetwork/java/javase/downloads/</u>
- OWASP Zed Attack Proxy (ZAP): <u>https://code.google.com/p/zaproxy/</u>
- OWASP ZAP SVN repository: <u>https://code.google.com/p/zaproxy/source/checkout</u>

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

- Eclipse IDE for Java Developers Indigo SR1 (v3.7 32 bits)²: <u>http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/indigosr1</u>
- Subclipse (v1.8.4): <u>http://subclipse.tigris.org</u>

<u>NOTE</u>: 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.

² Older versions of Eclipse: http://wiki.eclipse.org/Older_Versions_Of_Eclipse

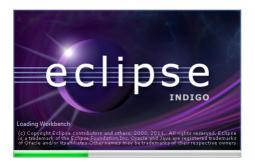


Installing the Software

Install the Java Development Kit (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".

<u>NOTE</u>: This version of the guide uses Subclipse, integrated within the Eclipse IDE, to retrieve a copy of the latest ZAP source code revision, available at the Google Code URL detailed on the "Software Requirements" section. The installation of Subclipse is detailed in the "Installing Subclipse" section.

Starting Eclipse



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

Workspace Launcher	X
Select a workspace	
Eclipse stores your projects in a folder called a workspace. Choose a workspace folder to use for this session.	
Workspace: C:\Users\siles\eclipse_SR1_workspace	▼ <u>B</u> rowse
Use this as the default and do not ask again	
	OK Cancel

Enter your workspace folder (e.g. "C:\Users\siles\eclipse_SR1_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 the arrow on the right to go to the workbench:



<u>NOTE</u>: 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:

sype filter text	Installed JRE	5		🗢 🖛 🚽 👻
 ▷ General ▷ Ant ▷ Help ▷ Install/Update 		r edit JRE definitions. By default, the d Java projects.	checked JRE is add	ed to the build pat
⊿ Java	Name	Location	Туре	<u>A</u> dd
 ▷ Appearance ▷ Build Path ▷ Code Style 	🔽 🛋 јгеб	C:\Program Files (x86)\Java\jre6	Standard VM	<u>E</u> dit
 Code style Compiler 				Dupli <u>c</u> ate
▷ Debug ▷ Editor				Remove
 ▶ Installed JREs JUnit Properties Files Editor ▶ Mayen ▶ Mylyn ▶ Run/Debug ▶ Team > Usage Data Collector Validation > WindowBuilder > XML 				<u>S</u> earch

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. If not, using the "Add..." or "Search..." buttons locate the JDK folder and add it. For example, through the "Search..." button, select the parent folder where the JDK has been previously installed: "C:\Program Files (x86)\Java" and click "OK". If both JRE and JDK are installed, remove the original JRE entry through the "Remove" button, leaving the following setup:

pe filter text	Installed JREs		<p th="" ⇒="" ▼="" ▼<=""></p>
General Ant Help Install/Update	Add, remove or edit JF of newly created Java Installed JREs:	RE definitions. By default, the checked JRE is ad projects.	ded to the build p
Java	Name	Location	Add
Appearance Build Path	🔽 🛋 jdk1.6.0_30	C:\Program Files (x86)\Java\jdk1.6.0_30	<u>E</u> dit
Code Style Compiler			Dupli <u>c</u> ate
Debug Editor			Remove
Installed JREs JUnit Properties Files Editor Mayen Mylyn Run/Debug Team Usage Data Collector Validation WindowBuilder XML			Search
	•	III	Þ

<u>NOTE</u>: The Eclipse Indigo help is available at: http://help.eclipse.org/indigo/index.jsp.



Installing Subclipse

If you plan not only to use ZAP from SVN but to contribute to the ZAP project, by adding new features or capabilities to this tool or fixing known issues³, the ZAP project lead (*psiinon*) recommends using Subclipse (http://subclipse.tigris.org), a plug-in providing support for Subversion (SVN) within the Eclipse IDE. Subclipse allows you to easily integrate the current SVN ZAP revision in Eclipse and be able to manage source code changes from within Eclipse.

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

To briefly sum up the installation process for the current Eclipse IDE version, Indigo SR1 (v3.7), slightly outdated on the reference above, go to the "Help – Install New Software..." menu. Enter the following URL for Subclipse 1.8.x (for Eclipse 3.2+) into the "Work with" field, "http://subclipse.tigris.org/update_1.8.x", and click the "Add..." (Repository) button. Select a name for the repository, such as "Subclipse 1.8.x", and click "OK":

add Re	pository	×
<u>N</u> ame:	Subclipse 1.8.x	L <u>o</u> cal
Location:	http://subclipse.tigris.org/update_1.8.x	Archive
?	ОК	Cancel

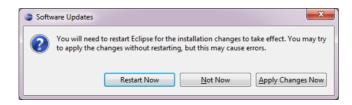
Eclipse will contact the Subclipse repository and display all the available features on the intermediate pane. Select the Subclipse checkbox to select all the associated features, or select them individually, and click "Next":

🗦 Install		 X
Available Software Check the items that you wish to install.		
Work with: Subclipse 1.8.x - http://subclipse.tigris.org/update_1.8.x		<u>A</u> dd
Find more software b	by working with the <u>"Available Software</u>	<u>Sites</u> preferences
type filter text		
Name	Version	
▲ 🔽 III Subclipse		
🖉 🌆 CollabNet Merge Client	3.0.5	
👿 🏟 Subclipse (Required)	1.8.4	
👿 🖗 Subclipse Integration for Mylyn 3.x (Optional)	3.0.0	
📝 🖗 Subversion Client Adapter (Required)	1.8.0	
📝 🖗 Subversion JavaHL Native Library Adapter (Required)	1.7.2	
📝 🏇 Subversion Revision Graph	1.1.1	
Select All Deselect All 6 items selected		
Details		
Subversion 1.7 plug-in for Eclipse 3.2 and higher.		More
Show only the latest versions of available software	items that are already installed	
	is already installed?	
	a <u>uncauj naturca</u> .	
Show only software applicable to target environment		
\overline{Z} <u>C</u> ontact all update sites during install to find required software		
(?)	Next > Einish	Cancel

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

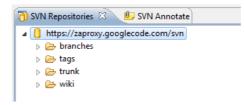


Eclipse will calculate requirements and dependencies (a process that can take some time, so be patient...), and will guide you through the installation process. The first screen shows the install details. Click "Next", accept the license agreements and click "Finish". The Subclipse software will be installed (be patient...) and, once the process finishes, Eclipse must be restarted for the installation changes to take effect; click on "Restart Now":



Adding the ZAP SVN Repository to Eclipse

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 location "https://zaproxy.googlecode.com/svn/" in the "Url" field and click "Finish". As a result, the ZAP SVN repository will be available within Eclipse:



<u>NOTE</u>: The SVN repository location must use "httpS" (HTTPS) if you plan to commit changes into the ZAP project (Google Code; see "Committing Your Local ZAP Changes to SVN using Eclipse"). The "http" version can only be used in read only mode to retrieve the SVN repository contents.

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.

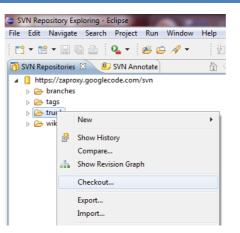
Creating the ZAP Project in Eclipse

Subclipse allows you to check out the ZAP project from the official SVN repository into your local Eclipse workspace, that is, create a local project working copy (in SVN terminology)⁴. You can check out the main "trunk" ("https://zaproxy.googlecode.com/svn/trunk"), or an alternative branch such as 1.3 ("https://zaproxy.googlecode.com/svn/branches/1.3").

Navigate to the "SVN Repository Exploring" perspective and from the "SVN Repositories" tab, right click on the repository directory you want to check out, such as "trunk", and select "Checkout...":

⁴ http://www.taddong.com/docs/SVN_CheatSheet.pdf

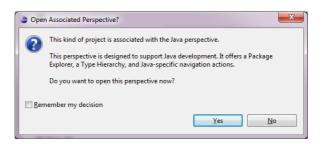




Select "Check out as a project configured using the New Project Wizard" as the method of check out to create the local project and click "Finish". On the "New project" screen, select "Java – Java Project" as the wizard and click "Next". On the "Create a Java Project" screen, select a project name (e.g. "zaproxy") and click "Finish":

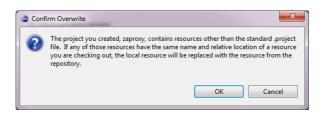
New Java Project	
Create a Java Project Create a Java project in the workspace or in ar	n external location.
Project name: zaproxy	
Use default location	pace\zaproxy Browse
JRE	
Ose an execution environment JRE:	JavaSE-1.6
Use a project specific JRE:	jdk1.6.0_30 💌
Use default JRE (currently 'jdk1.6.0_30')	Configure JREs
Project layout	
 Use project folder as root for sources an Create separate folders for sources and 	
Working sets	
Add project to working sets Working sets:	▼ Select
? <u>Sack</u>	ext > Finish Cancel

A warning message clarifies that the "Java" perspective is associated to the development of Java projects in Eclipse. Click "Yes":





A new warning message clarifies the check out (and merge) process between the local project and the SVN repository. Click "OK":



Eclipse will perform the SVN check out operation (be patient...) and build the project workspace:

SVN Checkout	
Checking out trunk	
2,8 MB C:\Users\siles\eclipse_SR1_workspace\zaproxy\lib	
Always run in background	
SVN Checkout	
Checking out trunk: 2,6 MB	
Run in <u>B</u> ackgro	und Cancel << Details
Kunni <u>B</u> ackgro	

As a result, the ZAP project contents from SVN have been populated into the local Eclipse project:

🧅 Java - Eclipse	Completion of the local division of the loca				
<u>File Edit Source Refactor Navigate S</u>	e <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp				
🗄 • 📬 • 🗐 🖻 🚔 🚸 • 🔾 •	∿ - ₩ <i>©</i> - <i>№ © ⊷ 2</i> -	Ş • ♥ ↔ •			😭 🔜 SVN Reposit 🖉 »
Image: Package Explorer Image: Package Explorer Image: Package Explorer Image: Package Explorer <th></th> <th></th> <th></th> <th>Find G</th> <th></th>				Find G	
				Connect <u>Connect</u> or <u>create</u> a E Outline S An outline is no	to your task and ALM tools a local task.
	Problems 🛛 @ Javadoc 😣 Declaration				<u>6</u> ° ∨ ⊓ □
	3.313 errors, 390 warnings, 0 others (Filter matched 20		-		
	Description	Resource	Path	Location	Туре
	 Errors (100 of 3313 items) Warnings (100 of 390 items) 				
Zaproxy		Updating	j indexes		© [?] (?) (?) (?)

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

Select the new project from the workspace "Package Explorer" tab. Go to the "Project – Properties" menu and select the "Java Build Path" page:

Properties for zaproxy		
ype filter text	Java Build Path	<p th="" ↔<=""></p>
Resource Builders Java Build Path Java Code Style Java Compiler Java Editor Javadoc Location Project References Refactoring History Run/Debug Settings Task Repository Task Tags Validation WikiText	Contract Source Projects Libraries Image: Contract of the source	Add JARs Add External JARs Add <u>V</u> ariable Add Libr <u>a</u> ry Add <u>C</u> lass Folder Add External Class Fol <u>d</u> er <u>E</u> dit <u>R</u> emove <u>M</u> igrate JAR File
?		OK Cancel

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":

⊿ 🔁 zaproxy	
Lupiony .settings	[
b 🗁 bin	
build	
🔺 🗁 lib	
📓 bcmail-jdk16-146.jar	
🔳 bcprov-jdk16-146.jar	
BrowserLauncher2-1_3.jar	
📓 bsh-2.0b4.jar	
commons-beanutils-1.8.3.j	ar
commons-codec-1.4.jar	
commons-collections-3.2.1	
commons-configuration-1.	
commons-httpclient-3.0.ja	r I
commons-lang-2.6.jar	
commons-logging-1.1.1.jar	r l'

As a result, all the ZAP project JAR files are added to the build path:



type filter text	Java Build Path	← ← → →
Resource Builders Java Build Path Java Code Style Java Compiler Java Editor Javadoc Location Project References Refactoring History Run/Debug Settings Task Repository Task Repository Task Tags Validation WikiText	Source Projects Libraries Order and Export JARs and class folders on the build path: Descrow-jdk16-146.jar - zaproxy/lib Descrow-jdk16-146.jar - zaproxy/lib Descrow-jdk16-146.jar - zaproxy/lib Descrow-order-14.jar - zaproxy/lib Descrow-order-10.jar - zaproxy/	-
	Jero greep r isozyja zaprosy/lib Jero Fuzzjar - zaprosy/lib Jero FuzzEncoder.jar - zaprosy/lib Jero dom.jar - zaprosy/lib	•

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"):

Setti	ng Java Build Path
?	The Java Build Path property page contains unsaved modifications. Do you want to save changes so that other build path related property pages can be updated?
	Apply Discard Apply Later

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".

type filter text	Java Build Path	⇔ - ⇔
Resource Builders Java Build Path Java Code Style Java Compiler Java Editor Javadoc Location Project References Run/Debug Settings Task Repository Task Tags Validation WikiText	Source Projects Libraries & Order and Export Source folders on build path:	Add Folder Link Source Edit <u>R</u> emove
	Allow output folders for sour <u>c</u> e folders Defaul <u>t</u> output folder:	



Go to the "Run – Run configurations..." menu. Select "Java Application" and press the "New" button:

Run Configurations	A REAL PROPERTY AND A REAL	×
Create, manage, and run conf Run a Java application	rigurations	
Ype filter text Ype filter text Ymax Applet Java Application JUnit Maven Build JY Task Context Test	Configure launch settings from this dialog: Configure launch settings from this dialog: Press the 'New' button to create a configuration of the selected type. Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. Press the 'Filter' button to configure filtering options. Edit or view an existing configuration by selecting it. Configure launch perspective settings from the <u>'Perspectives'</u> preference page.	
Filter matched 5 of 5 items		
?	ßun	Close

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").

Run Configurations		— X
Create, manage, and run conf Run a Java application	igurations	
Yee Image: Second	Name: ZAP Image: Source in the second sec	Common Browse
Filter matched 6 of 6 items	Apply	Revert
?	Run	Close

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:<u>zaproxy/bin</u>}".



reate, manage, and run cont Run a Java application	rigurations ()
Ŷ 📄 🗶 🖻 券 ▼ type filter text	Name: ZAP ^(C) Main ^(k) = Arguments AF JRE ^(k) Classpath ^(k) / ₂ Source ^(m) / ₂ Environment ^(k) / ₂ Common
🕎 Java Applet 🗊 Java Application	Program arguments:
 New_configuration JU JUnit Maven Build JU Task Context Test 	v Varjables
-	VM arguments:
	v Variables
	Working directory: Default: \${workspace_loc:zaproxy} Other: \${workspace_loc:zaproxy/bin} Workspace Eile System
ilter matched 6 of 6 items	Apply Revert

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):

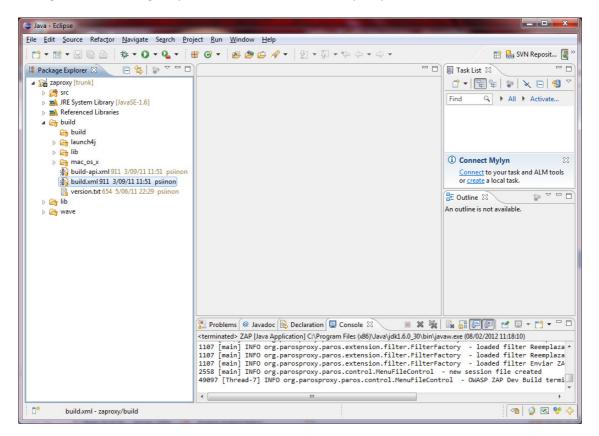
Sesión sin Nombre - OWASP ZAP	
Archivo Editar Ver Analizar Informes Herramient	tas Ayuda
	0
Sitios	Petición⇒ Respuesta⇔ Punto de interrupción 💥]
E Sitios	Split All Text T
	Fuerta Bruta 🖉 Escaneo de puertos 📰 Fuzzer 🌞 Parametros 🗐 Highlighter Salida
	📕 Historia 📅 🗍 Buscar 🔍 🗍 Puntos de interrupción 💥 🗍 Alertas 🏴 🗍 Escaneo Activo 👌 🗍 Spider 🛞
	√ Filtro:OFF
	4
	l l l l l l l l l l l l l l l l l l l
Alertas 🔑 0 🔑 0 🕫 0	Escaneo actual 🕷 0 👌 0 👑 0 👙 0 🍃

ZAP can be launched at any time through the "Run – Run" menu in Eclipse.



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 - (1) 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:

😰 Problems 🖉 Javadoc 🚱 Declaration 📮 Console 🗵 💿 🔳 🗶 🍇 🔐 🔐 🛃 💽 🛃 🛫 🗂 🕶 🗖
<terminated> zaproxy build.xml [Ant Build] C:\Program Files (x86)\Java\jdk1.6.0_30\bin\javaw.ex Show Console When Standard Out</terminated>
[copy] Copying 2 files to C:\Users\siles\eclipse_SR1_workspace\zaproxy\build\zap [jar] Building jar: C:\Users\siles\eclipse_SR1_workspace\zaproxy\build\zap.jar
<pre>[jar] Building jar: C:\Users\siles\eclipse_SR1_workspace\zaproxy\build\zap\zap-api.jar package-linux: [echo] Packaging the Linux build</pre>
<pre>[tar] Building tar: C:\Users\siles\eclipse_SR1_workspace\zaproxy\build\ZAP_1.3.0_Linux.ta package=macosx: BUILD SUCCESSFUL Total time: 18 seconds</pre>
Total time: 18 seconds

The full contents of the build directory can be packed in a ZIP file, distributed and tested (e.g. "C:\Users\siles\eclipse_SR1_workspace\zaproxy\build\zap\"), 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 Linux & Mac OS X ("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.



ZAP Project Development

The latest ZAP development version is always available in the SVN trunk ("/svn/trunk", all references belong to "https://zaproxy.googlecode.com/"), including fixes and new functionality, although important fixes are sometimes also applied to specific branches, such as the 1.3 branch ("/svn/branches/1.3/"), if the ZAP project is working on releasing a minor version, such as 1.3.4.

The SVN trunk is always updated when a new version is about to be released.

Updating the Local ZAP Working Copy from SVN using Eclipse

Typically, after a few days using or working with your local ZAP project you might be interested on integrating the latest changes from the official SVN repository into your local working copy within Eclipse.

The first step is to synchronize the local working copy with the SVN repository using Subclipse. Go to the "Java perspective" and the "Package Explorer" tab, select the local project, right click and select the "Team - Update to HEAD" (revision) option. Your local copy will be updated with the current SVN repository revision:

SVN Update	
Updating zaproxy-SVN	
1,6 kB	
Always r <u>u</u> n in background	
	Run in Background Cancel Details >>

The local project current SVN revision can be obtained by right-clicking on the local project and selecting the "Team – Show History" option. A new "History" tab will open at the bottom containing all the SVN details:

	ms 🕼 Javadoc 🔀		Console	His	
'trunk in ht	tp://zaproxy.google	ecode.com/svn			◈ \$ ≝ ♣ ▾ ≱ ᠒ ▣ ■ ▶ ▶
Revision	Date	A	uthor		Comment
*1152	6/02/12 10:35	al	la@gremwell.co	om	Fix for Spider extension so that spidering sites on port.
1149	2/02/12 13:51	p	psiinon@gmail.com		Added Persion translation, thanks to Mohsen
1146	2/02/12 13:46	p	psiinon@gmail.com		Persian translation, thanks to Mohsen Mostafa Jokar
11 <u>43</u> ∢	1/02/12 20:46	alla@gremwell.com			Fix for various extensions not showing multiple items in d
	ected paths	Description		Ŧ	Fix for Spider extension so that spidering sites on port 80 is possible again. The fix
•	111		۱.		is in ScanPanel, other changes are cosmetic

Once the local working copy is updated with the latest SVN repository revision, you can run ZAP or launch the ZAP building process (see "Building ZAP"). In both cases, Eclipse will automatically recompile the ZAP project if it is required (due to the existence of a new SVN revision).



Committing Your Local ZAP Changes to SVN using Eclipse

<u>NOTE</u>: In order to be able to commit any changes to the official ZAP SVN repository you must have permissions and a proper set of credentials (username and password). Contact the ZAP project leader to ask for these permissions and become a ZAP project contributor.

If you have been working on your ZAP project local working copy and want to share your changes with the rest of the world (*ok, just with the ZAP project... :-*), you need to commit your modifications. Before committing any changes, it is always recommended to synchronize the working copy with the most current updates (latest revision) from the official SVN repository (as described in "Updating the Local ZAP Working Copy from SVN using Eclipse").

Once the working copy is updated, and the local changes are ready in the working copy (as denoted by a "*" symbol – see images below), go to the "Java" perspective, select the project from the "Package Explorer" tab (or just the specific file(s) or folder(s) you want to commit), right click, and select "Team – Commit...":

⊿ ta xml	🔺 🗄 xml
alert.dtd 5 4/09/10 13:52 psiinon	🚯 alert.dtd 5 4/09/10 13:52 psiinon
👔 alert.xml 5 4/09/10 13:52 psiinon	🛐 alert.xml 5 4/09/10 13:52 psiinon
🛐 alertDef.dtd 5 4/09/10 13:52 psiinon	🚯 alertDef.dtd 5 4/09/10 13:52 psiinon
👔 alertDef.xml 5 4/09/10 13:52 psiinon	🛐 alertDef.xml 5 4/09/10 13:52 psiinon
🚯 config.dtd 5 4/09/10 13:52 psiinon	🚯 config.dtd 5 4/09/10 13:52 psiinon
🌇 config.xml 736 19/06/11 15:42 a.c.neumann	👔 config.xml 736 19/06/11 15:42 a.c.neumann
🛐 drivers.dtd 187 15/10/10 1:16 a.c.neumann	🚯 drivers.dtd 187 15/10/10 1:16 a.c.neumann
📝 drivers.xml 736 19/06/11 15:42 a.c.neumann	🔀 drivers.xml 736 19/06/11 15:42 a.c.neumann
🔒 log4j.properties 253 4/12/10 10:47 psiinon	log4j.properties 253 4/12/10 10:47 psiinon

The "Commit" window will open and allows you to add a commit description and select the files you want to commit. Click "OK":

Commit	
Commit to: https://zaproxy.googlecode.com/svn/	runk/src/xml
Enter a comment for the commit operation.	
drivers.xml updated: new drivers naming sch eID added (issue 182).	me (OS & bits - x86 or x64), list reordered and Belgium 🔺
diver you undated new divers naming scheme (OS &	 x86 or x64), list reordered and Belgium eID added (issue 182).
Configure Comment Templates	s - xoo or xo4), list reordered and beigium eto added (issue 102).
Changes	🗏 Keep locks 🛛 📰 ն
v ≥ zaproxy v ≥ src v ≥ xml v w drivers.xml	



Eclipse will request the credentials (username and password) required to complete the commit operation:

Enter Username and Password					
Repository:	<https: zaproxy.googlecode.com:443=""> Google Code Sub</https:>				
Username:	USERNAME				
Password:	**********				
Save Pass	sword:				
?	OK Cancel				

If you are using a Google Account to contribute to the ZAP project, the username is your Google Account username, and the password is the randomly generated Google Code password available at "<u>https://code.google.com/hosting/settings</u>" after logging in with your Google Account.

The Google Code password used for SVN access is not your Google Account password.

<u>NOTE</u>: The SVN repository location must use "httpS" (HTTPS) if you plan to commit changes into the ZAP project (Google Code). If you selected by mistake the "http" version when adding the SVN repository to Eclipse, the commit operation will fail with a "405 Method Not Allowed" error.

However, you can relocate the SVN repository from the "SVN Repository Exploring" perspective. Right-click in the SVN repository and select the "Relocate..." option. Eclipse will show the affected projects (click "Next") and will allow you to change the SVN repository location through the "New URL" field.

NOTE: If you select the "Save Password" checkbox in the window asking for the SVN credentials, Eclipse and Subclipse will store your SVN username and password (using wincrypt) in a file under the following folder "C:\Users\<username>\AppData\Roaming\Subversion\auth\svn.simple" in Windows 7. The location changes in Eclipse for Linux and Mac OS X: "~/.subversion/auth/svn.simple".

You must decide if you allow Eclipse to store your username and password in your local file system, and know that this is the location where the credentials are stored. In case you decide to regenerate (change) your Google Code password, delete the file.

If the commit is successful, Eclipse will not generate any message. You can verify that the commit has been added to the project by updating your working copy to the latest SVN revision and verifying the SVN history (as explained in "Updating the Local ZAP Working Copy from SVN using Eclipse"):

Design Source							
Problems	@ Javadoc 😣 Declaration	🖳 Console 📄 Histor	y 🛙	🔗 🕏 🛃 🖾 ▾ 券 /Z 🛡 📰 ▷ 💌 ▽ 🗆 🗖			
/trunk in https	://zaproxy.googlecode.com/sv	n					
Revision	Date		or	Comment			
*1153	8/02/12 14:06		iles@gmail.com	drivers.xml updated: new drivers naming scheme (OS & bits - x86			
1152	6/02/12 10:35		gremwell.com	Fix for Spider extension so that spidering sites on port 80 is possible ag			
1149	2/02/12 13:51		on@gmail.com	Added Persion translation, thanks to Mohsen			
•				<u> </u>			
Action Affe	Action Affected paths Description			drivers.xml updated: new drivers naming scheme (OS & bits - x86 or x64), list reordered and Belgium eID added (issue			
٠ III			•	182).			



OWASP ZAP Extensions

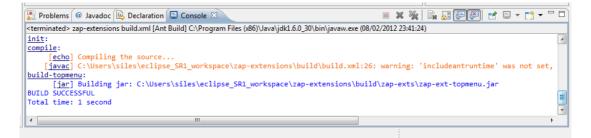
Recently, the OWASP ZAP project has been enhanced through the OWASP ZAP Extensions project. The ZAP Extensions project contains templates for extending OWASP ZAP without rebuilding it, including, for example, active or passive scanners, or extensions with right or top menu items:

- OWASP ZAP Extensions: <u>https://code.google.com/p/zap-extensions/</u>
- OWASP ZAP Extensions SVN repository: <u>https://code.google.com/p/zap-extensions/source/checkout</u>

```
svn co https://zap-extensions.googlecode.com/svn/trunk/ zap-extensions-ro
```

The ZAP Extensions project can be built following the same process described throughout this guide for the OWASP ZAP project, starting with section "Adding the ZAP SVN Repository to Eclipse", by simply changing the associated SVN repository URL: "https://zap-extensions.googlecode.com/svn/".

The only difference is that you do not have to define a Java project main class. Close the "Run Configuration" window after setting everything up (except the main class), and build the ZAP Extensions project from the associated "build.xml" file. Eclipse will create the corresponding ZAP extensions JAR files:

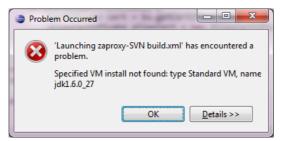




Appendix: Java Version Updates and Eclipse

Every time you update your (Java) JDK version, for example from Java 6 Update 27 to Update 29, it is mandatory to change the general Eclipse Java VM settings, as well as the specific references in all your Java projects to the new folder where the new Java version has been installed (e.g." C:\Program Files (x86)\Java\jdk1.6.0_29").

If the specific Java projects references are not changed, when the project Ant building process is launched, the following error is generated: "Specified VM install not found: ...":



After completing all the steps below, you need to restart Eclipse so the new project settings are reloaded.

First of all, ensure you have the new Java VM version set within Eclipse, as specified in the "Checking the Java Environment (JRE or JDK) Detected by Eclipse" section:

Preferences				
type filter text	Installed JREs			\ ← ♥ ♥ ♥
Editor Installed JREs	Name	Location	Туре	<u>A</u> dd
JUnit Properties Files Edito	📝 🛋 jdk1.6.0_29	C:\Program Files (x86)\Java\jdk1.6.0_29	Standard VM	<u>E</u> dit
→ Maven ▼				Dupli <u>c</u> ate
0			ОК	Cancel

In order to change the specific project references to the new Java VM, edit the files contained in the "C:\Users\siles\eclipse_SR1_workspace\.metadata\.plugins\org.eclipse.debug.core\.launches\" folder in Windows 7, and change the Java version number for every project file on that folder to the new one, that is, point them to the new Java VM name and version:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<launchConfiguration type="org.eclipse.ant.AntLaunchConfigurationType">
...
<stringAttribute key="org.eclipse.jdt.launching.VM_INSTALL_NAME"
value="jdk1.6.0_29"/>
...
</launchConfiguration>
```

<u>NOTE</u>: Do not forget to uninstall the previous vulnerable JDK version from the operating system.