LSim in Eclipse

1. First of all, you need to have the Subversion plugin for Eclipse. If you don't have it: Go to "Help" \rightarrow "Install new software". Then, search "Subversive" and check "Subversive SVN Team Provider". Click on "Next", accept the terms and click on "Finish". You will need to restart Eclipse.

8 Install	
Available Software	
Check the items that you wish to install.	
Work with: -All Available Sites-	▼ Add
Find more software by working wit	h the <u>"Available Software Sites"</u> preferences.
subversive	E
Name	Version
Collaboration	
We subversive Revision Graph (Optional) We subversive SVN Integration for the Mylyn Project (Optional)	1.0.0.120120601-1700
Subversive SVN Integration for the Mytyn Project (Optional)	1.0.0.120120601-1700
Subversive SVN Team Provider	1.0.0.120130122-1700
Select All Deselect All 1 item selected	
Details	
The Eclipse Team Provider for the Subversion version control system. For instr	uctions about installing SVN connectors 🛛 🕃
Show only the latest versions of available software Hide items t	More hat are already installed
	vinstalled?
Group Items by category	y installed:
Show only software applicable to target environment	
Sontact all update sites during install to find required software	
C Back	Next > Cancel Finish

2. Create a new repository location with the URL svn://eimtserv.uoc.edu/LSim/trunk and your user and password.

😣 New Re	pository	Location			
Enter Repos	itory Lo	cation Infor	mation		SVN
Define the SN additional se	/N reposi ettings for	tory location proxy and sy	information. Yo n+ssh, https cor	u can specify inections.	0
General Ac	lvanced	SSH Settings	SSL Settings		
URL: svn://	eimtserv.	uoc.edu/LSin	n/trunk		▼ Browse
Label					
🖲 Use the	e reposito	ry URL as the	label		
🔿 Use a c	ustom lab	el:			
Authentica	tion				
User:	dllamaz	ares@uoc.ed	u		~
Password:	•••••				
Save au	Ithenticat	ion (could tri	gger secure stor	age login)	
To manage	your secu	rity data, ple	ase see <u>"Secure</u>	Storage"	
Show Creden	tials For:	<repository< td=""><td>/Location></td><td></td><td>: *</td></repository<>	/Location>		: *
Validate	Repositor	y Location or	n finish		Reset Changes
?				Cancel	Finish

3. Right click on the repository created and do "Find/Check Out As...". Select "Check out as a project with the name specified:" and click on "Finish".

😣 Check Out As
Check Out As You can checkout the selected repository resource by different ways. Select the method of checkout you prefer to use.
Choose how to check out folder 'LSim' Check out as a project configured using the New Project Wizard Find projects in the children of the selected resource
 Check out as a folder into existing project Check out as a project with the name specified:
LSim
Depth: Recursively Revision Head Revision Date: 3/5/2013 9:59:31 AM Revision: Browse
Cancel Finish

4. After doing the checkout, it's time to import the maven projects. Go to "File" → "Import" → "Existing Maven Projects". Then, browse the directory of "Isim-parent" in "LSim" folder.
Now, select all projects and click "Finish".

Root Directory:	/home/o	dllamazares	/workspace/l	Sim/lsim-parent	▼	Browse
projects:	edu ues i	do co leioulo	im paceptil 0	CNIADCUIOTicaco		
/pom.xmi	eau.uoc.o	oom.xml.ec	du.uoc.dpcs.ls	im:lsim-commons:	S{lsim.v	Select <u>A</u> ll
✓/lsim-co	des/pom	.xml edu.u	oc.dpcs.lsim:l	sim-codes:\${lsim.v	ersion}:j	<u>D</u> eselect A
👿/lsim-lib	orary/pon	n.xml <mark>edu.u</mark>	oc.dpcs.lsim:	lsim-library:\${lsim	version}	<u>R</u> efresh
🛛/lsim-st	orage/po	m.xml <mark>edu</mark>	.uoc.dpcs.lsim	:lsim-storage:\${lsi	m.versio	
🗹/lsim-lo	aging/po	m.xml edu.	uoc.dpcs.lsim	lsim-logging:\${lsir:	n.versio	n]
	999/ 00				A (1 1	
✓/lsim-di	spatcher/	/pom.xml edu	du.uoc.dpcs.l	sim:lsim-dispatche	er:\${lsim.	N
☞/lsim-di ☞/lsim-la	spatcher/ uncher/po	/pom.xml e om.xml edu	du.uoc.dpcs.l u.uoc.dpcs.lsir	sim:lsim-dispatche n:lsim-launcher:\${l	er:\${lsim. sim.vers	N i¢
☞/lsim-di ☞/lsim-la	spatcher/ uncher/po	/pom.xml e om.xml edu	du.uoc.dpcs.l J.uoc.dpcs.lsir	sim:lsim-dispatche n:lsim-launcher:\${l	er:\${lsim. sim.vers	N İ¢
☑/lsim-di ☑/lsim-la	spatcher/ uncher/po	/pom.xml edu	du.uoc.dpcs.l	sim:lsim-dispatche n:lsim-launcher:\${l	er:\${lsim.	ic D
 ✓/lsim-di ✓/lsim-lai ✓/lsim-lai ✓/lsim-lai 	spatcher/ uncher/po (s) to wor	/pom.xml edu om.xml edu king set	edu.uoc.dpcs.lsir	sim:lsim-dispatchen:lsim-launcher:\${l	er:\${lsim. sim.vers	i e
 ✓/lsim-di ✓/lsim-lai ✓/lsi	(s) to wor	/pom.xml edu om.xml edu king set	edu.uoc.dpcs.lsir	sim:lsim-dispatchen:lsim-launcher:\${l	er:\${lsim. sim.vers	Mor <u>e</u>
 ✓/lsim-di ✓/lsim-la ✓/lsim-la ✓/lsim-la ✓/lsim-la ✓/lsim-la 	(s) to wor	/pom.xml edu om.xml edu king set	edu.uoc.dpcs.lsir	sim:lsim-dispatchen:lsim-launcher:\${l	er:\${lsim. sim.vers	Nic Mor <u>e</u>

5. To compile all projects with maven: Right click on "lsim-parent" \rightarrow "Run As" \rightarrow "Maven install". It is also possible to compile with the Console, doing "mvn install" in the "lsim-parent" directory. Anyway, you should get something like this:

🕄 Problems 🛛 🖉 Javadoc 😟 Declaration 🖳 Console 🛱 🍇 i 🔒 🚮 🖨 🖉 **e X** 📃 🔻 📑 🔻 [INFO] -----[INFO] BUILD SUCCESS [INF0] -----[INFO] Total time: 4.487s [INFO] Finished at: Tue Mar 05 10:23:43 CET 2013 [INF0] Final Memory: 11M/103M [INFO] -----(1)

Maven

LSim projects have maven dependencies and sometimes it is needed to update one of the libraries used. For example, imagine that we have made some changes in the source code corresponding to the library runenvironment-1.0.jar and we have generated a new version of this library (a new JAR), called runenvironment-2.0.jar.

How to incorporate this new library to the project LSim? It's easy: First, the new library has to be uploaded to the maven repository, following the maven structure. Then, the pom.xml file of every project with dependencies to the library has to be edited indicating the desired version of it.

The maven repository is located at dpcscodes.uoc.edu, in the directory "/var/www/maven". So, it is only needed to upload the new files to the server. For example:

scp -P 55055 ./runenvironment-2.0.jar dpcscodes.uoc.edu:/home/dllamazares

Then, move the files to the right directory. Continuing with our example:

```
ssh -p 55055 dpcscodes.uoc.edu
sudo cp runenvironment-2.0.jar /var/www/maven/edu/uoc/runenvironment/2.0/
```

Once the repository have the new library, it's time to edit the pom.xml files. A good way to find these projects is going to the Eclipse top menu and clicking "Search" \rightarrow "Search". Then select the left tab ("File Search") and search "<artifactId>runenvironment". The pom.xml files to be edited will be shown. In our example:



For every pom.xml, open it and change the "Overview" tab to the "pom.xml" tab. Then, look for the dependency that has to be edited and change the version of it. In our example, we have changed the version from 1.0 to 2.0:



That's all. For more information about maven, you can check this tutorial: <u>http://maven.apache.org/guides/getting-started/</u>