EL4J

SetupEL4J

Generated from Wiki page:
http://wiki.elca.ch/twiki/el4j/bin/view/EL4J/SetupEL4J

Version	Date	Author(s)	Visa
1.9	20 Mar 2009 - 11:14	Main.DavidBernhard, Main.DominikZindel, Main.FabianReichlin, Main.StefanWismer, StefanWismer	

©ELCA Informatique SA, Switzerland 2009

Table of Contents

1 Getting Started 1

1.1 Setting up your environment 1

1.1.1 JDK 1.5 SE 1

1.1.2 Cygwin 1

1.1.3 EL4J directory structure 2

1.1.4 Maven 2 2

1.1.5 Eclipse 2

1.1.6 Optional tools (Optional for EL4J users / mandatory for EL4J developers) 3

1.1.6.1 JAD 3

1.1.6.2 Subversion 3

1.1.7 Congratulations 3

1.2 For Developers working on the EL4J framework 3

1.2.1 Download sourcecode of EL4J framework (the external part from sourceforge) 3

1.2.2 Change settings 4

1.2.3 JDBC Drivers 4

1.2.4 Build project 4

1.2.5 Configure Eclipse 4

1.2.6 Import EL4J modules into Eclipse 5

1.2.7 Synchronize EL4J modules in Eclipse 5

1.2.8 For internal developers 6

1 Getting Started

This guide is intended to help new EL4J developers (=developers working on EL4J itself) and EL4J users (=developers building applications on top of EL4J) to set up the environment, downloading the sources and learning the ropes of EL4J and its tools.

Please go step by step through the following sections. It's recommended to use the same directory structures and names as documented here. If you change directory names - you're on your own! Not all maven tools are tolerant when you change the directories only in some config files.

Hint: the swung dash ~ (=tilde) used in path descriptions is a shortcut for the home directory used by unix-systems (and by cygwin as well). On Windows, this is C:\Documents and Settings\{username}.

1.1 Setting up your environment

This section will guide you through the installation of all necessary tools which are required to develop and/or use EL4J and its buildsystem *Maven 2*.

1.1.1 JDK 1.5 SE

- Download the most recent update of JDK 5.0 (*Standard edition*, revision must be grater than 14) from http://java.sun.com/javase/downloads/index_jdk5.jsp
- Follow the instruction of the installation guide and install the Developer Kit to `C:\jdk<version>`, where `<version>` is your update version. The reason for this is that whitespaces in the path could lead to problems.
- At some time, the installation guide will ask you to install the JRE. Change the standard installation directory to `C:\jre<version>`
- Check your environment variables. You should have:
  - `JAVA_HOME` pointing to `C:\jdk<version>`
  - an entry in your `PATH` variable pointing to `%JAVA_HOME%\bin` (add all entries in the path at the *beginning*)
- Go to `C:\jdk<version>\bin` and make a copy of `javaw.exe`. Name it `eclipse_javaw.exe`. You will find this very handy, because it will prevent you from killing Eclipse when killing Java jobs.

1.1.2 Cygwin

Cygwin is a Linux-like environment for Windows. We use it to run shell-scripts and maven build commands.

- Go to http://www.cygwin.com/ and download the latest version of Cygwin.
- Install it to `C:\cygwin`
- Check your environment variables. You should have:
  - an entry in your `PATH` variable pointing to `C:\cygwin\bin`
- *Please skip this step if you plan to install Maven 2 as described below. The `finishInstallation` script will automatically do this for you.*
  create a `.bash_profile` file in your home directory and add following lines:

```
alias debugmaven='export MAVEN_OPTS="-Xmx1024M -Xss128k
-XX:MaxPermSize=256M -Xdebug
-Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8000"'
alias runmaven='export MAVEN_OPTS="-Xmx1024M -Xss128k
-XX:MaxPermSize=256M -Duser.language=en -Duser.region=US"'
```

More information about Cygwin can be found at CygwinFaqAndTips. We recommend to install Console2 described there.

1.1.3 EL4J directory structure

We recommend the following directory structure:

- A directory `Projects` for your projects, e.g. `D:\Projects`
- A directory `EL4J` in your `Projects` directory for the EL4J resources, e.g. `D:\Projects\EL4J`
- A directory `tools` in your `EL4J` directory for the tools (i.e. Maven) needed for development, e.g. `D:\Projects\EL4J\tools`
- A directory `m2repository` where Maven stores the downloaded and generated libraries, e.g. `D:\m2repository`. The default in Maven is `~/.m2/repository` but this is not recommended due the home path under Windows normally contains blanks that can not be handled by some Maven plugins.

1.1.4 Maven 2

Maven 2 is the build-system used to by the EL4J framework. We prepared a zip-file for convenient installation of Maven 2 containing a patched version of Maven (due to few unresolved bugs in the standard edition) and shell-scripts to set environment variables. Just go through the following steps:

- Download the latest version of the EL4J convenience zip from sourceforge (http://sourceforge.net/project/showfiles.php?group_id=147215) and unzip it to `D:\Projects\` (Windows) or `/data/Projects` (Linux).
- rename the so created folder (should look similar to el4j-1.3) to EL4J.
- open a cygwin or bash shell and change to the EL4J-directory (`cd Projects/EL4J`).
- execute `chmod 755 *sh`
- execute `chmod 755 tools/maven/bin/mvn`
- execute `./finishInstallation.sh` to finalize the installation. You need to do this only once.
- if the script printed a line starting with `Aliases installed.` the following is not necessary: execute `source ./setupPathsAndEnvironment.sh` to set up your environment. This is required each time you open a new shell. Alternatively, add the line `source D:/Projects/EL4J/setupPathsAndEnvironment.sh` (Windows) or `source /data/Projects/EL4J/setupPathsAndEnvironment.sh` (Linux) to your `.bash_profile` file in your home directory. Now, every time you open a shell, the setup script will be executed.
- execute `./checkInstallation.sh` to check your installation. Compare the output with the following expectations:
  - ♦ java -version
    - ◊ Must print out the version number of a Java 5 JDK or newer (e.g. `1.5.0_13`).
  - ♦ javac -version
    - ◊ Must print out the same version number as above.
  - ♦ mvn -version
    - ◊ Must print out the version `2.0.7` or newer.
  - ♦ echo $MAVEN_OPTS
    - ◊ Must print something like `-Xmx1024M -Xss128k -XX:MaxPermSize=256M -Duser.language=en -Duser.region=US`.
      If not, execute `export MAVEN_OPTS="-Xmx1024M -Xss128k -XX:MaxPermSize=256M -Duser.language=en -Duser.region=US"`

Note for internal developers/users: please follow the additional steps described in the corresponding section in the InternalGettingStarted#Maven_2.

1.1.5 Eclipse

Note for internal developers/users: please follow the corresponding section in the InternalGettingStarted#Eclipse guide and skip this one.

- Go to http://www.eclipse.org/ or http://el4.elca-services.ch/el4j/tools/eclipse/ and download the latest version of Eclipse.
- Unzip it to your program directory, e.g. to `C:\Program Files\eclipse`.
- If you like, you can set a shortcut. After creating the shortcut, right click on it and set the target to `C:\Program Files\eclipse\eclipse.exe -vm C:\jdk<version>\bin\eclipse_javaw.exe -Duser.language=de -Duser.region=CH -vmargs -Xmx384M`

1.1.6 Optional tools (Optional for EL4J users / mandatory for EL4J developers)

The following tools are not required to run Maven or to use EL4J, but they are very useful and thus recommended.

1.1.6.1 JAD

JAD is a free Java Decompiler.

- Download the most recent version of JAD from http://www.kpdus.com/jad.html#download
- Copy the `jad.exe` file into your `C:\jdk<version>\bin` directory.

1.1.6.2 Subversion

Subversion is the recommended version control system. If you are an EL4J developer and want to develop the EL4J-framework itself, you need subversion to checkout the source code later on.

- Please check out the info under http://intranet.elca.ch/Business_Process/Utilities/Subversion/Subversion.php as there are some disturbing bugs in Svn clients (it's slowly getting better - YMA knows more about it)
- Download the correct version of Svn to your favorite directory, such as `C:\Subversion`.
- Check your environment variables. You should have:
  - `APR_ICONV_PATH` pointing to `C:\Subversion\iconv`
  - an entry in your `PATH` variable pointing to `C:\Subversion\bin`
- Copy the `config` file http://el4j.svn.sourceforge.net/viewvc/*checkout*/el4j/trunk/el4j/etc/subversion/config as `config` (remove the htm extension) into path `~\Application Data\Subversion`
- Open a cygwin console and type in `svn --help` to check the correct installation of subversion.

1.1.7 Congratulations

Congratulations, you've completed the setup of your environment to be ready to work with the EL4J framework.

- If you are a developer who wants to work **on the EL4J framework itself**, continue with the next section to get the source code and build the framework.
- As a developer **using the EL4J Framework to build its own application**, you don't necessarily need the source code and you can skip the section after the next section that indicates how to get the EL4J application templates.

1.2 For Developers working on the EL4J framework

This section is intended both for internal and external developers of EL4J.

1.2.1 Download sourcecode of EL4J framework (the external part from sourceforge)

- Open a cygwin console and cd to `D:\Projects\EL4J`.
- Check out the external repository with the following command:

- `svn co https://el4j.svn.sourceforge.net/svnroot/el4j/trunk/el4j external`

1.2.2 Change settings

- Go to `~/.m2` directory. There should be a `settings.xml` file, which has been copied from the EL4J maven convenience-zip by the shell-script `finishInstallation.sh` executed in the GettingStarted. If there is no `settings.xml`, copy it from `D:\Projects\EL4J\external\etc\m2\` to your `~/.m2` directory.
- Change the `settings.xml` file in your `~/.m2` directory as follows:
  - Change the value of the `localRepository` tag to `D:/m2repository`
  - Remove the comments around the `proxy` tag
  - Remove the comments around the `el4j.root`, `el4j.external` and `el4j.internal` tags (under profile `el4j.general`)
  - Remove the comments around the `mirror` tags, as described in the xml comments
  - Change the value of `el4j.root` to `D:/Projects/EL4J`
  - The value of `el4j.external` should be `${el4j.root}/external`
  - Change the value of `el4j.project.home` to `${el4j.root}`

1.2.3 JDBC Drivers

Internal developers should skip this section. If you are not inside the ELCA network, you will have to download the JDBC Driver for Oracle yourself, because we are not allowed to distribute them.

To download and deploy this driver into your local repository, do the following:

- Download the Oracle driver (ojdbc14.jar) from "http://www.oracle.com/technology/software/tech/java/sqlj_jdbc/". Note: You can also use the 10g driver for version 9 database.
- Open a console and change to the directory where you have saved the downloaded jar files.
- Execute the following commands:
  - `mvn install:install-file -DgroupId=com.oracle -DartifactId=ojdbc14_g -Dversion=10.2.0.1.0 -Dpackaging=jar -Dfile=ojdbc14_g.jar`

Note that the used file names of the jar files in the commands above may depend on the downloaded version. The used version number (`10.2.0.1.0`) is the one used in `module-database`. Leave this version number as it is to avoid dependency conflicts.

1.2.4 Build project

- Go to `D:\Projects\EL4J\external` in your cygwin console and type `mvn clean install`. This will probably take a while (~30min) and will build the external part of the EL4J framework.
- Type `mvn eclipse:clean eclipse:eclipse -DdownloadSources=true` to let Maven 2 generate Eclipse project files.

1.2.5 Configure Eclipse

- Open Eclipse and set up a new `D:\Projects\EL4J\workspace` workspace.
- Go to `File -> Import`
- Select `General -> Preferences`
- Choose `D:\Projects\EL4J\external\etc\eclipse\eclipse<version>.epf` (according to your eclipse version)
- Import preferences by clicking on finish.

1.2.6 Import EL4J modules into Eclipse

You could add all EL4J modules into your workspace, but this can be confusing. Therefore, we recommend to set up working sets as follows:

- Open Eclipse with your `D:\Projects\EL4J\workspace` workspace.
- Go to `Window -> Preferences`
- Go to `Java -> Build Path -> Classpath Variable`
- Add a new variable with the name `M2_REPO` and the path `D:\m2repository`
- Add another variable, set the name to `EL4J_HOME` and the path to `D:\Projects\EL4J`

- Close the Preferences window and click on the little triangle in the uppermost right corner of your `Package Explorer`.
- Go to `Select Working Sets...`.
- Create the working sets `modules`, `applications`, `demos`, `tests` and `plugins` (`New -> Java`).
- Click on the triangle again and choose `Top Level Elements -> Working Sets`.

- Go to `File -> Import` and choose Existing Projects into Workspace
- Click on Next and on the next page on Browse next to "Select root directory". Go to `D:\Projects\EL4J\external\framework\modules`, click on Ok and then Finish.
- Move the projects in your `modules` set.
- Repeat the same with
  - `D:\Projects\EL4J\external\framework\tests` (for `tests`),
  - `D:\Projects\EL4J\external\applications\demos` (for `demos`)
  - `D:\Projects\EL4J\external\applications\templates` (for `applications`).
  - `D:\Projects\EL4J\external\maven` (for `plugins`).
- Additionally you can exclude external libraries. Click on the triangle and on `Filters...`. Choose `Libraries from external` there.

1.2.7 Synchronize EL4J modules in Eclipse

Within Eclipse, you may also want to connect to the external repository to keep EL4J modules up-to-date. If you downloaded Eclipse from http://el4.elca-services.ch/el4j/tools/eclipse/ a Subversion plug-in comes pre-installed, otherwise you can try Subversive.

- Open the SVN Repositories view (`Window -> Show View -> Other... -> SVN -> SVN Repositories`).
- In the top-right of this view, click on the `New Repository Location` icon.
- For the URL, enter `https://el4j.svn.sourceforge.net/svnroot/el4j`. You can leave the Authentication-section open, unless you have an user account and a password.
- In the `Advanced` register, make sure that `Enable Structure Detection` is activated.
- To synchronize the already imported EL4J modules, go to the Navigator view (`Window -> Show View -> Navigator`).
- Having this view open, you'll see a list of all imported modules. To synchronize them with the external repository, mark them all, right-click on the resulting selection, and choose `Team -> Share Projects...`.
- A box will open where the path to the repository is asked (`Reconnect all from:`). There will be multiple options, whereas `https://el4j.svn.sourceforge.net/svnroot/el4j/trunk/el4j` is the path in question. Just try the different options until this path appears.
- Confirm and a new box pops up. Choose `Use project settings` and click on `Finish`.
- Now you're done, as the EL4J modules are now connected to the external repository. You can go back to the Package Explorer view and start exploring the modules.

1.2.8 For internal developers

Internal developers: in addition to this section, please follow the InternalGettingStarted too.