Table of Contents

Table of Contents

1 CourseAboutEl4j

1.1 Course about EL4J

Here's the content of the short EL4J course. **Please don't forget to look at the notes of each page (it contains info about further readings).**

- Spring: This presentation explains the core concepts and ideas of spring
  - ♦ SpringIntroductionPresentation.ppt
  - ♦ illustrative sample code (as simple as possible, not production quality, use the application template to get started!)
- Maven: http://leaffy.elca.ch/java/el4j/Documentation_Mirror/maven2/Maven2Course.ppt
- EL4J: http://leaffy.elca.ch/java/el4j/Documentation_Mirror/El4jFasttrackCourse.ppt

Other important topics:

- JSF-Slides (as intro to JSF):
  http://leaffy.elca.ch/java/el4j/Documentation_Mirror/jsf/jsfTemplatePresentation.ppt
- Lighter Config for DAOs (from slide 4 on):
  http://leaffy.elca.ch/java/el4j/Marketing_Mirror/MonthlyNews/NewEL4JFeaturesJan08_3.ppt
- J2EE? course: http://el4j.elca.ch/java/el4j/Documentation_Mirror/miscellaneous/j2eeCourse/
- OOAD course: AdvancedOoadCourse

2 GettingStartedWithTheJ2EE

2.1 Some links about getting started with the J2EE:

Remarks about relevant technologies: You need a J2EE implementation. The sun reference implementation is not very popular, the most frequently used implementations are jboss (free and light), weblogic (commercial and good), websphere (commercial and heavy). I recommend to use jboss. In general, EJB Entity Beans are considered a legacy technology now. EJBs are often considered too heavy (they will soon become ligher, search for EJB 3.0 for more details. Nobody serious uses today servlets and JSP alone, one uses a web framework with them (e.g. Struts or even better, Spring MVC). Most people that know the J2EE, don't use the J2EE alone, but typically uses Spring to make the J2EE more convenient.

2.1.1 Short introduction (do this first)

For a short introduction in 5 lessons, follow the tutorial here:
http://java.sun.com/javaee/5/docs/firstcup/doc/toc.html

Try to run the examples in the jboss J2EE implementation available here:
http://www.jboss.com/products/jbossas Try to understand the sample applications in the tutorial and slightly change them and rerun them again to verify whether you understand each application.

2.1.2 Longer introduction:

Standard sun tutorial: http://java.sun.com/javaee/5/docs/tutorial/doc/

Focus first on chapters 1, 3, 11, 12, 13, 22, 23, 24 , 25, 28, 30, 31, 32, 33, 36, Do not look much into Entity Beans!

2.1.3 More advanced topics

Spring framework (is almost standard today when using the J2EE): Website:
http://www.springframework.org/

Tutorial: http://www.springframework.org/docs/MVC-step-by-step/Spring-MVC-step-by-step.html

J2EE Design patterns I: http://java.sun.com/blueprints/corej2eepatterns/index.html
http://java.sun.com/blueprints/patterns/catalog.html

2.1.4 Other tools considered essential:

Junit, ant, eclipse and its J2EE plugins

2.1.5 Some reference documentation:

http://java.sun.com/javaee/index.jsp

Book on EJB (available as PDF): http://www.theserverside.com/news/thread.tss?thread_id=31942

3 InternalGettingStarted

3.1 Internal Getting Started

This guide is intended as an add-on for internal EL4J developers and users.

As a new (internal) EL4J developer or user you should go through the GettingStarted guide and jump to this page whenever you see a *Note for internal developers*. After you have set up EL4J external as in GettingStarted#ForDevelopers, continue with the setting up EL4J - internal section.

3.1.1 Setting up your environment

3.1.1.1 Proxy

In order to speed up your internet connection you have to set the proxy at different places.

3.1.1.1.1 Internet Options

Open the `Internet Options` from the `Control Panel` and click on the button `LAN Settings...` in the tab `Connections`. This opens the window `Local Area Network (LAN) Settings`. Deselect all options but the second one (`Use automatic configuration file`). In the field `Address` there should be the URI `http://wpad.elca.ch/wpad.dat`. If not, put it and confirm with `Ok`

3.1.1.1.2 Firefox

Open `Tools -> Options` and go to the tab `Network` in the section `Advanced`. Click on the button `Settings...`. In the opening window `Connection Settings` select the very last option (`Automatic proxy configuration URL:`) and write the URI `http://wpad.elca.ch/wpad.dat` into the field.

3.1.1.1.3 Java

Open the `Java Control Panel` from the `Control Panel` and then click on the button `Network Settings` in the tab `General`. Select the option `Use browser settings`.

3.1.1.2 Eclipse

- Go to http://leaffy.elca.ch/java/tool-downloads/eclipse/ or http://el4.elca-services.ch/el4j/tools/eclipse/ and download the latest zip of Eclipse. This is optional and you could use the version from the Eclipse homepage too, but this version has already got some recommended plugins included.
- Unzip it to `D:\Projects\EL4J\tools\eclipse`.
- If you like, you can set a shortcut. After creating the shortcut, right click on it and set the target to `D:\Projects\EL4J\tools\eclipse\eclipse.exe -vm C:\jdk<version>\bin\eclipse_javaw.exe -Duser.language=de -Duser.region=CH -vmargs -Xmx384M`
- In order for svn to work correctly with Eclipse, please do what is described under the section "Subversion with Java application" on the following page: http://intranet.elca.ch/Business_Process/Utilities/Subversion/Subversion.php
- In case you want to configure your existing eclipse, here are the details hot to set it up: ExternalToolEclipse

3.1.1.3 Maven 2

In addition to the steps described in the SetupEL4J#SetupMaven2 section, do the following:

- Go to your `~/.m2` directory and open the `settings.xml`.
- Uncomment the proxy-tag.
- Within the `<mirrors>` tag, uncomment all mirror-elements according to the comments.
- Provide username and password for server `el4jRepositoryInternal: el4/qMZZGo4G`

3.1.2 Setting up EL4J - Internal

In addition to the instruction in the external guide, this instructions will show you how to download the internal part of the framework and set it up for Eclipse.

3.1.2.1 Download sources

- Open a cygwin console and change to your `Projects/EL4J` directory.
- Check out the internal repository with `svn co https://cvs.elca.ch/subversion/el4j-internal/trunk internal`

3.1.2.2 Build project

- Go to `D:\Projects\EL4J\internal` and repeat the `mvn clean install` to build the internal part.
- Do the same with `mvn eclipse:clean eclipse:eclipse -DdownloadSources=true` to create the Eclipse project files.

3.1.2.3 Include EL4J into Eclipse

You could add another working set for the internal part of the framework:

- Open Eclipse with your `D:\Projects\EL4J\workspace` workspace.
- Find the little triangle in the uppermost right corner of your `Package Explorer`.
- Go to `Configure Working Sets....`
- Create the working set `internal`.

- 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\internal`, click on Ok
- Remove projects that you don't want to import (sandbox etc.)
- Click on Finish.
- Move the projects in your `internal` set.

3.1.3 Reading

For reading material, take a look at IntroductoryReadingListForEl4J. You could

- Read the datasheet,
- Read some of the TSS article, especially about Dependency Injection, AOP and IoC? if you haven't heard about it yet,
- Read some chapters of the Spring Manual,
- Take a look at a book like `Spring Pro` (please ask POS for available books on Spring)
- Study the Maven2 Presentation if you are confused about Maven.

If you got the basics of Spring, Maven & Co and would like to start developing, take a look at the two following links. You could do this while going ahead with the next section so that you learn some of the EL4J conventions and practice at the same time.

- Take a look at the ModuleCore documentation. Read the sections up to `Convenience Attributes for Transactions`, especially the `Conventions`, how to use the `ModuleApplicationContext` and the use of `Java 5 annotations for transaction` (if you'll forget to include the configuration for the java5annotations you'll sooner or later wonder, why your hibernate updates won't work!)

- Keep an eye on the ReferenceDocumentation. You'll find a lot of useful knowledge there, like the DeveloperGuidelines.

- If you want to know more about useful Maven commands, take a look at the MavenCheatSheet

3.1.4 Initial development

In addition to the GettingStarted#InitialDevelopment, you will learn about the internal part of the framework here and set up the Web Application Template.

3.1.4.1 The EL4J framework (internal & external)

The EL4J internal is intended as an extension to the EL4J external. It consists of modules that ELCA cannot or doesn't want to make available to the public. It's structure is similar to the one of EL4J external and the parent of it's root pom `D:\Projects\EL4J\internal\pom.xml` is the `external\pom.xml`.

3.1.5 Web Application Template

- Download the latest Web Application Template zip file from http://leaffy.elca.ch/java/el4j/templates/ (e.g. `web-template-1-2.zip`).
- Extract it to `D:\Projects\EL4J` and follow the `README.txt` file for instructions on how to run it.
- Then generate Eclipse file (`mvn eclipse:clean eclipse:eclipse -DdownloadSources=true`) and include it into your Eclipse workspace.

Play around with it a little bit. Try to understand the three tier architecture we used in this template. Take a look at the `dao` layer. How is the persistence of the domain objects implemented? How do bean configurations and POJOs work together? See WebApplicationTemplate for additional information.

4 IntroductoryReadingListForEl4J

4.1 Introductory Reading List for EL4J

The goal of this topic is to provide a list of documents to read to get to know EL4J.

- Browse our EL4J wiki to get an idea about the essential content. We recommend 4 ways to find your information in the twiki (the 4 ways are described here).
- A datasheet to get the motivation and a basic overview of EL4J
  http://leaffy.elca.ch/java/el4j/Marketing_Mirror/El4JDatasheet.pdf
- A basic introduction to Spring, which is at the heart of EL4J, can be gained through the following articles:
  - ♦ A basic introduction to dependency injection:
    http://leaffy.elca.ch/java/el4j/Documentation_Mirror/spring/Spring_Recipes_A_Problem_Solution_Ap
  - ♦ For persons that are very new to enterprise Java, there are gentle introduction books, such as *Pro Spring* (please contact POS, he has a personal paper version of this book).
  - ♦ TSS article: http://www.theserverside.com/tt/articles/article.tss?l=IntrotoSpring25
  - ♦ The Spring presentation from the EL4J course: CourseAboutEl4j
  - ♦ Spring reference manual, read chapters 1-5:
    http://static.springframework.org/spring/docs/2.5.x/spring-reference.pdf
  - ♦ Look at some Spring sample applications in the sample directory of the spring download
- Introduction to Maven, our build system
  - ♦ MavenBuildSystem: the best thing is to start with our Maven ppt presentation (available on that wiki page)
  - ♦ Online Maven 2 book
- More detailed introduction to EL4J:
  - ♦ Look at the other presentations of the EL4J course: CourseAboutEl4j. Don't forget to check out the slide notes!
  - ♦ The reference documentation for EL4J:
    http://wiki.elca.ch/twiki/el4j/bin/view/EL4J/ReferenceDoc?skin=pdf
  - ♦ Go through the FrequentlyAskedQuestions
  - ♦ An article for architects, explaining the features of EL4J:
    http://el4j.sourceforge.net/docs/pdf/EL4J_IntroductionArticle1.pdf
- For more information, please consult the twiki or consult PhilippHOser

5 ProjectLifecycle

5.1 The Project Lifecylce of a EL4J based project

5.1.1 Before starting a project

- For new projects starting with Spring or EL4J, please contact PhilippHOser to decide on the most appropriate use of the technologies. We also collect information on how other projects use EL4J and Spring (to give you interesting links to other parties). Ideally there is an experienced Spring/ EL4J programmer in the team of a new project.
- I want to migrate from LEAF2. What should I know?
  - ♦ A comment on the migration plan from LEAF 2 to EL4J
  - ♦ What is new for LEAF 2 users

5.1.2 Starting a new project

5.1.2.1 Choosing a base where to start

EL4J provides a small number of templates that simplify setting up a project very much. It is strongly recommended to contact the EL4J team. We can help you setting the environment up effectively.

5.1.2.1.1 Simple project

To start with a trivial program (1 class and 1 test) that contains a dependency to EL4J module core, follow the steps of the README.txt file in the current EL4J convenience zip.

5.1.2.1.2 GUI application template

Refer to SwingApplicationTemplate.

5.1.2.1.3 JSF application template

Refer to TenderTracker and ModuleJsf.

5.1.2.1.4 Old web application template

Refer to the old web application template.

5.1.2.1.5 General purpose template

This template is implemented as maven archetype: `mvn archetype:generate -DarchetypeGroupId=ch.elca.el4j.archetypes -DarchetypeArtifactId=EL4JProject -DarchetypeVersion=1.7 -DgroupId=ch.elca.el4j -DartifactId=myFirstProject -DremoteRepositories=http://el4.elca-services.ch/el4j/maven2repository`

The archetype will ask for values for the following parameters:

- `projectNaturalName`: The official name of the Project (spaces allowed)
- `projectNameOneWord`: The short name of the Project as one word (to be used as identified, e.g. use camel case; no spaces allowed)

5.1.2.2 Choosing where to deploy maven artifacts that are not deployed to any public repository

Most of the projects (sooner or later) need libraries that are not accessible on any public repository. In this case, it is recommended to set up an own repository for the project. The same holds for SNAPSHOT dependencies. Whenever they break something in their artifact, it is likely that your build will not succeed either. See MavenRepositoryForOwnProject for more details.

5.1.2.3 Project's stucture on disk

We recommend that every developer places the project files in the same structure. Here an example file structure:

```
D:\Projects\MY_PROJECT_NAME
    \trunk (Subversion sources)
        \my-project-name
            \etc
                \m2 (place a Maven2 settings.xml here for each developer and server, also custome
                \cygwin (place here the file mentioned in chapter "Cygwin and Console2 configurat
    \tools (Maven, Tomcat, Weblogic ...)
    \workspace (Eclipse workspace)
```

5.1.2.4 Customize Maven2

Every project should have its own Maven2 instance. Do **not** use one Maven2 installation for multiple projects. Use the newest Maven2 provided by EL4J (in convenience zip downloadable from http://el4j.sourceforge.net). As explained in the previous chapter create a settings.xml for each developer and server. Adapt the two Maven2 run scripts `mvn` and `mvn.bat` for every machine. Adapt the project name and settings file location.

mvn (right after the comment of the shell script):

```
echo "**************************************"
echo "* Maven for MY PROJECT - Maven 2.2.1 *"
echo "**************************************"

SETTINGS_FILE="D:\Projects\MY_PROJECT_NAME\trunk\my-project-name\etc\m2\settings-developer-abc.xm
```

mvn.bat (right after the "@echo off" of the bat script):

```
echo "**************************************"
echo "* Maven for MY PROJECT - Maven 2.2.1 *"
echo "**************************************"

set SETTINGS_FILE="D:\Projects\MY_PROJECT_NAME\trunk\my-project-name\etc\m2\settings-developer-ab
```

When you now run "mvn" in your console you will see always the first three lines and be sure that you use the correct Maven2 installation. In Cygwin/Unix you will even see different colors for the different log levels.

5.1.2.5 Cygwin and Console2 configuration

Cygwin without a good console is no fun. It is strongly recommended to use Console2 described here: CygwinFaqAndTips#ConsoleTwo

5.1.2.5.1 Files to provide

For an optimal cygwin environment, the following resources should be provided (located at `etc/cygwin`):

- a default `.bash_profile` (in case the user has none). The most important part are the last two lines, which allow to configure the environment from outside (the user's .bash_profile should not have to contain project specific settings or at least must not conflict with other projects) :

```
if [ ! -z $PROJECT_DIRECTORY ]; then cd "$PROJECT_DIRECTORY"; fi
if [ ! -z $PROJECT_INIT_SCRIPT ]; then source "$PROJECT_INIT_SCRIPT"; fi
```

- the `init-script.sh`. This script extends the PATH variable (e.g. to include the right maven path) and sets some aliases.
- the `cygwin.bat`. This file has to be referenced in Console2 (Edit - Settings - Tabs - Shell). The batch file sets `PROJECT_DIRECTORY` and `PROJECT_INIT_SCRIPT` before starting cygwin.
- for even more convenience, bash completion files could be provided

5.1.2.5.2 How it works (example)

- The user opens a new tab in Console2. Console2 supports different types of consoles and each console gets configured to start an exe or bat file. Therefore to be precise, the user starts a new console for project XY.
- The start bat file is configured to be located at `D:\Projects\XY\etc\cygwin\cygwin.bat` and sets some environment variables (`PROJECT_DIRECTORY` and `PROJECT_INIT_SCRIPT`)
- Cygwin starts up and executes `.bash_profile` (the project has provided an example file that the user copied to his home directory)
- This script jumps into the base directory and executes the project specific init script.

Tip: To use a different maven `settings.xml` file, add something like this to the init script file:

```
alias mvn='mvn -s "$PROJECT_DIRECTORY/etc/m2/alternative_settings.xml"'
```

Have a look at the EL4J trunk how this could look like.

See also CygwinFaqAndTips

5.1.2.6 Eclipse workspace

The eclipse workspace folder used by maven-eclipse-plugin is set to `${el4j.project.home}/workspace` by default (EL4J root pom.xml). This is important, otherwise the plugin cannot make project dependencies and will fall back to jar dependencies. If each developer should be free where to put his eclipse workspace directory, it is recommended to define another variable in (the user specific) settings.xml (`yourProject.eclipse.workspace`) and override the `${yourProject.eclipse.workspace}` in your project's root pom.xml.

5.1.3 Make a release

The two scripts presented in this section are included in the general purpose template (see above) or can be download from
http://el4j.svn.sourceforge.net/viewvc/el4j/trunk/el4j/maven/archetypes/project-template/src/main/resources/archetype-

5.1.3.1 Increase version number

- Go to the folder containing the root pom.xml file
- Execute `./etc/release-scripts/updateVersion.sh`. The script will ask you to insert the new version number.
- Check all modified files.
- Modify the script according to your needs (include more files, add hints which files have to be modified manually etc.)

5.1.3.2 Source distribution for offline environments

In several projects it was decided to deliver the source code to the customer and build it on his server. As maven downloads the necessary libraries on demand, one has to do additional work for servers that are not connected to the Internet: Not only the source but also a complete local repository has to be delivered. The script `etc/release-scripts/create-m2repoDiff.sh` creates a zip containing all new or modified contents. Of course, the first time it zips everything. This prevents you from having to send nearly the same hundreds of megabytes for each release.

- Open `etc/release-scripts/create-m2repoDiff.sh` and adapt the values in the configuration section
- Go to the folder containing the root pom.xml file
- Execute `./etc/release-scripts/create-m2repoDiff.sh`. It zips all files that are newer than the logfile `$REPO_DIFF_FOLDER/incrM2repoLog.log`.
- If you want to set the timestamp of the logfile, use `touch -t 200905130000 $REPO_DIFF_FOLDER/incrM2repoLog.log`

6 SetupEL4J

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}.

6.1.1 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*.

6.1.1.1 JDK 1.6 SE

- Download the most recent update of JDK 6 (*Standard edition (SE)*, revision must be grater than 15) from http://java.sun.com/javase/downloads/index.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.

6.1.1.2 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.

6.1.1.3 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.

6.1.1.4 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 6 JDK or newer (e.g. `1.6.0_16`).
    - ♦ 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#SetupMaven2.

6.1.1.5 Eclipse

Note for internal developers/users: please follow the corresponding section in the InternalGettingStarted#SetupEclipse 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`

6.1.1.6 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.

6.1.1.6.1 JAD

JAD is a free Java Decompiler.

- Download the most recent version of JAD from http://www.varaneckas.com/jad (or if original site is online again: http://www.kpdus.com/jad.html#download)
- Copy the `jad.exe` file into your `C:\jdk<version>\bin` directory.

6.1.1.6.2 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.

6.1.1.7 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.

6.1.2 For Developers working on the EL4J framework

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

6.1.2.1 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`

6.1.2.2 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}`

6.1.2.3 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.

6.1.2.4 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

6.1.1.7 Congratulations   14

framework.

- Type `mvn eclipse:clean eclipse:eclipse -DdownloadSources=true` to let Maven 2 generate Eclipse project files.

6.1.2.5 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.

6.1.2.6 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.

6.1.2.7 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.

6.1.2.8 For internal developers

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