EL4J

GettingStarted

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

Version	Date	Author(s)	Visa
1.42	16 Jan 2008 - 11:54	ClaudeHumard, Main.ClaudeHumard, Main.DavidStefan, Main.MarcusDeluigi, Main.MartinZeltner, Main.PhilippHOser, Main.StefanWismer	

Table of Contents

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*) 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`
- create a `.bash_profile` file in your home directory and add following lines:

```
alias debugmaven='export MAVEN_OPTS="-Xmx1024M -Xss128k
-XX:MaxPermSize=512M -Xdebug
-Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8000"'
alias runmaven='export MAVEN_OPTS="-Xmx1024M -Xss128k
-XX:MaxPermSize=512M -Duser.language=en -Duser.region=US"'
```

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).
- 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.
- 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=512M -Duser.language=en -Duser.region=US`.
      If not, execute `export MAVEN_OPTS="-Xmx1024M -Xss128k -XX:MaxPermSize=512M -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/ and download the latest version of Eclipse.
- 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`

1.1.6 Optional tools

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

(Optional) 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

(Optional for EL4J users / mandatory for EL4J developers) 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 Congratulation

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

- If you are an **EL4J developer** who wants to work on the EL4J framework itself, see the GettingStartedDeveloper guide on how to get the source code and build the framework.
- As an **EL4J user** who wants to use the EL4J Framework to build its own application, you don't necessarily need the source code and you can go on with the next section to get the EL4J Template Application.

1.2 EL4J Demo Applications

1.2.1 Simple Project

To start with a trivial program (1 class and 1 test) that contains a dependency to EL4J module core, do the following:

- open a shell, goto `D:\Projects\EL4J`
- execute `source ./setupPathsAndEnvironment.sh` (if not set in `~/.bash_profile`)
- execute `mvn archetype:create -DarchetypeGroupId=ch.elca.el4j -DarchetypeArtifactId=EL4JArchetypeCore -DarchetypeVersion=1.5 -DgroupId=ch.elca.el4j -DartifactId=myFirstProject -DremoteRepositories=http://el4.elca-services.ch/el4j/maven2repository`
- cd myFirstProject
- mvn install
- mvn exec:java

(Please consult the `README.txt` located in your Projects/EL4J-folder to get the current version to be set for `-DarchetypeVersion`)

1.2.2 GUI-Template

The GUI demo application contains a standalone-gui-application, a thin-client/server-application and a java Webstart demo-application to demonstrate the functionality of the EL4J framework. The GUI template can also be used as a start for a new EL4J application.

- Download the latest GUI Application Template zip-file from http://leaffy.elca.ch/java/el4j/templates/ (e.g. `gui-template-1.2.zip`).
- Extract it to `D:\Projects\EL4J` and follow the `gui\README.txt` and `gui\webstart\README.txt` file for instructions on how to run it.
- Generate Eclipse project files (`mvn eclipse:clean eclipse:eclipse -DdownloadSources=true`) to include it into your Eclipse workspace.

See ModuleSwing#How_to_get_started_with_our_demo for further details about the demo application and its components.

Note for internal developers/users: There is also a web application template available. See the corresponding section in the InternalGettingStarted#Web_Application_Template.

1.3 EL4J introduction

This section gives you a short overview over the build system Maven 2 and the project structure of a typical EL4J application. At the end, you find links to additional documents.

1.3.1 Maven2 introduction

This section will give you a brief introduction to the Maven2 build system. It will explain you the basic terms of Maven and the use of `archetypes`. Maven2 is a tool to manage software projects. Maven2 is able to manage a project's build, reporting and documentation based on a project object model called `POM`.

1.3.1.1 Structure of a Maven Project

EL4J is built with Maven and consists of several subprojects. Each of these subprojects (called `artifacts` in Maven) has the following structure:

- `src` directory containing the source files
- `pom.xml` file with the description of the `artifact` for Maven
- `.settings` directory as well as a `.classpath` and `.project` file if you invoke `mvn eclipse:eclipse`
- `target` directory if you invoke `mvn install`

`Artifacts` are hierarchically structured having a `root pom.xml` file, in our case `D:\Projects\EL4J\external\pom.xml`. The are linked with help of a `parent` tag that a `pom.xml` file can have.

1.3.1.2 Maven commands

There are only two Maven commands you will need at the beginning.

The first one is `mvn clean install`, which will do the following to the artifact and any child artifact

- `clean` deletes existing `target` directories in the artifact directory

- `install` compiles all sources in the `src` directory into a `artifactName.jar` file, runs JUnit tests, if there are any, creates the `target` directory, copies the jar file in the `target` directory. Moreover it copies the jar file into the local repository, in our case `D:\m2repository`

Note: To make changes on your artifact effective, you always have to invoke `mvn clean install`. *This will cause Maven to deploy the jar file into the local repository.*

The second command is of the form `mvn <plugin>:<goal>`. You will need the Maven Eclipse plugin to generate Eclipse project files for your projects. You do this using the command `mvn eclipse:clean eclipse:eclipse -DdownloadSources=true`. For further instructions on how to import a project into Eclipse, please read the `Eclipse` section under `Setting up EL4J`.

1.3.1.3 Dependencies & repositories

Now, go to your `D:\Projects\EL4J` directory in a cygwin console and

- invoke `mvn archetype:create -DarchetypeGroupId=ch.elca.el4j -DarchetypeArtifactId=EL4JArchetypeCore -DarchetypeVersion=1.3 -DgroupId=ch.elca.el4j -DartifactId=myFirstProject -DremoteRepositories=http://el4.elca-services.ch/el4j/maven2repository`. This will generate you a new Maven project
- Change to `cd myFirstProject`. As you see, you can find the `src` directory and the `pom.xml` file typical for a Maven project.
- Take a look at the `pom.xml` file:
  - You will see that our `pom.xml` file doesn't have a parent, because it's the top level pom of an independent project.
  - There are `dependencies` to `junit` and `module-core`. The first one is needed to run the tests of our projects (you'll see them later) and the second is the Core Module of EL4J. It's there because we want to build our project upon the EL4J framework.

Maven tries to resolve dependencies from the local repository, i.e. it checks if you have a jar file with the same `groupId`, `artifactId` and `version` in your local repository. If this is not the case, Maven will try to download these artifacts from the remote repositories to your local repository.
As you can easily see, Maven will have to download the artifacts from the remote repository only for the first time and will look it up in the local repository afterwards.

1.3.2 EL4J project structure

An EL4J project will have a typical structure:

- `src`
  - `main`
    - `java` This is where all the source (i.e. java) files go to.
    - `resources` This is where all additional files go to like configuration files.
    - `env`
      - `env` This is where the `env.properties` file goes to. If you invoke `mvn clean install` it will be copied to the `target` directory and will be accessible in the progam with help of `module-env`
  - `test` This is where all test files go to. It has the same structure as `main`, but is there for testing.
    - `java`
    - `resources`
    - `env`
      - `env`

We recommend you to go on with reading some of the additional material now.
Alongside, try to play around with the `myFirstProject` a little bit. Try, to import the project into eclipse. Add

then a `env.properties` file to your project, add a new dependencies to `module-env` from EL4J and use the class `EnvPropertiesUtils` from `module-env` to read out some properties you create.

1.4 Reading

For reading material, take a look at http://el4j.sourceforge.net/documentation.html

Note for internal developers: please see the corresponding section in the InternalGettingStarted#Reading guide for additional readings.

-- ClaudeHumard - 18 Dec 2007