How to use SBT in a scala project as external tool in Eclipse (JUNO 4.2) ?

HelloWorld example, with a generation of a standalone executable jar.

Author JL PASTUREL November 2012

Table des matières

1. Pre-requires

(November , 2012)

Tested with Eclipse JUNO 4.2

This install is described for a Windows box. There is no difficulty to adapt to a Linux Box

1.1. Install scala-Ide for JUNO :

http://download.scala-ide.org/nightly-update-juno-master-29x

when the stable release will be ready, switch to it

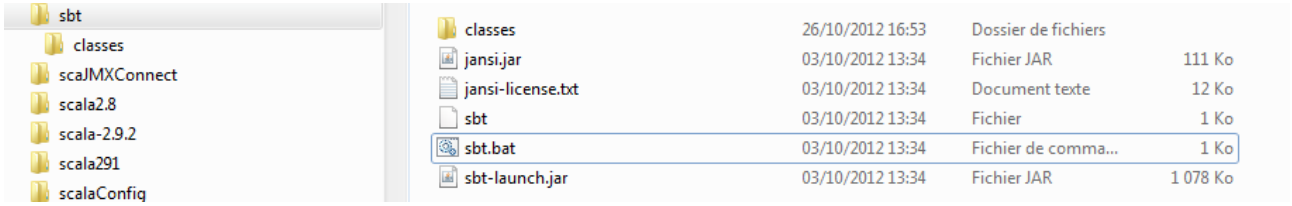
1.2. Install sbt for your box

Download **sbt.msi** from :

http://www.scala-sbt.org/release/docs/Getting-Started/Setup.html

For the Linux box, the page explain how to install it (download of sbt-launch.jar and create a script to launch it)

I install it in **d:\opt\sbt**



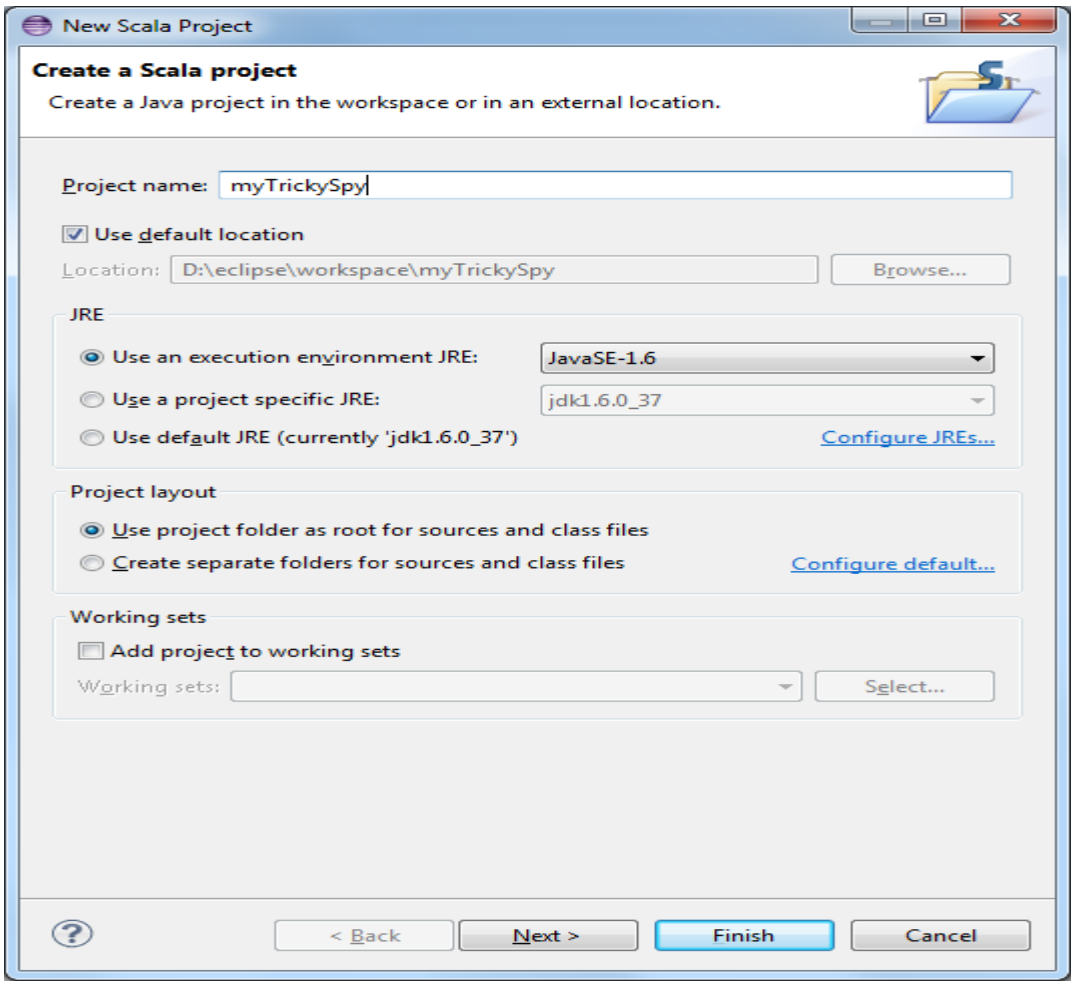
For the first launch, you need an internet connection.

Go to **d:\opt\sbt** and launch : **.\sbt**

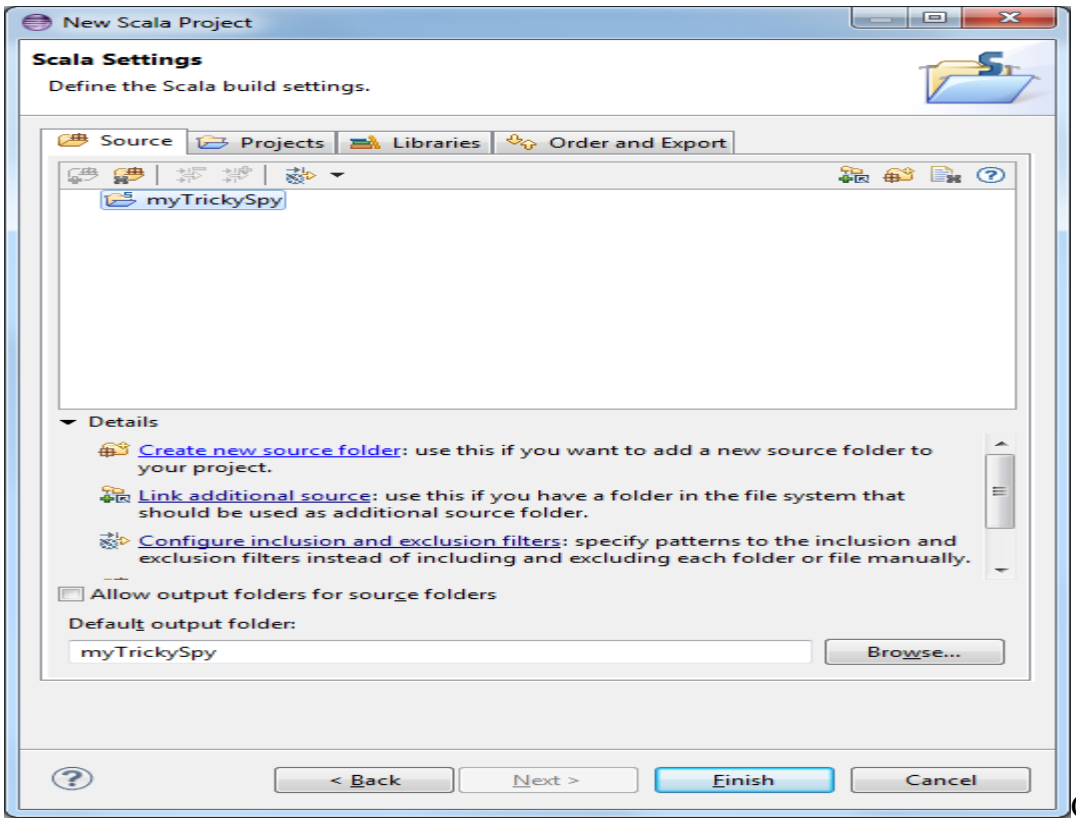
If you don't get errors, you can continue

2. Create a new Scala project, and customize it for sbt

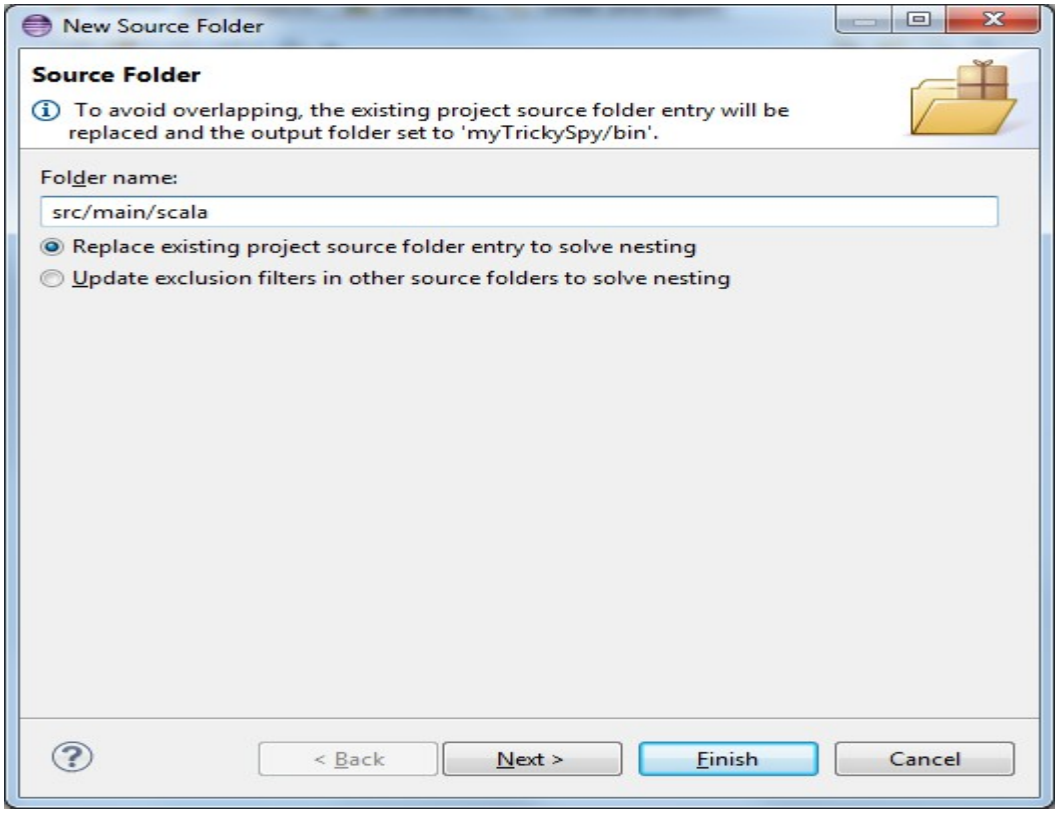
2.1. Create a project with src source in src/main/scala



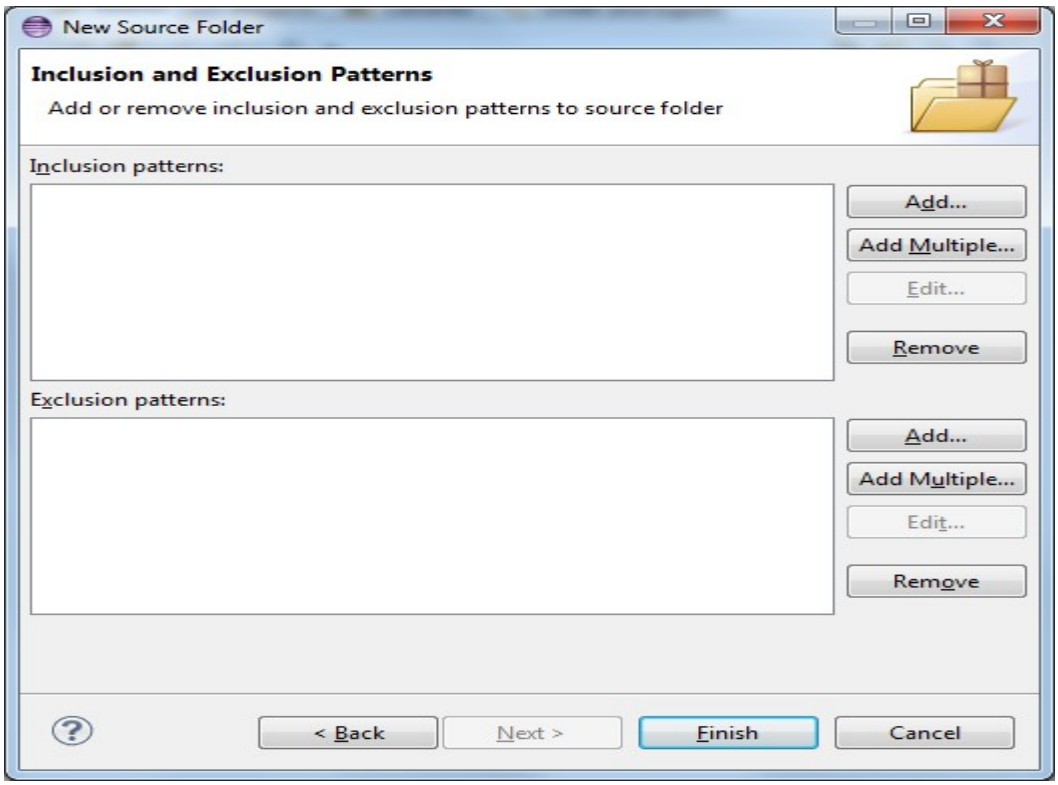
Next =>



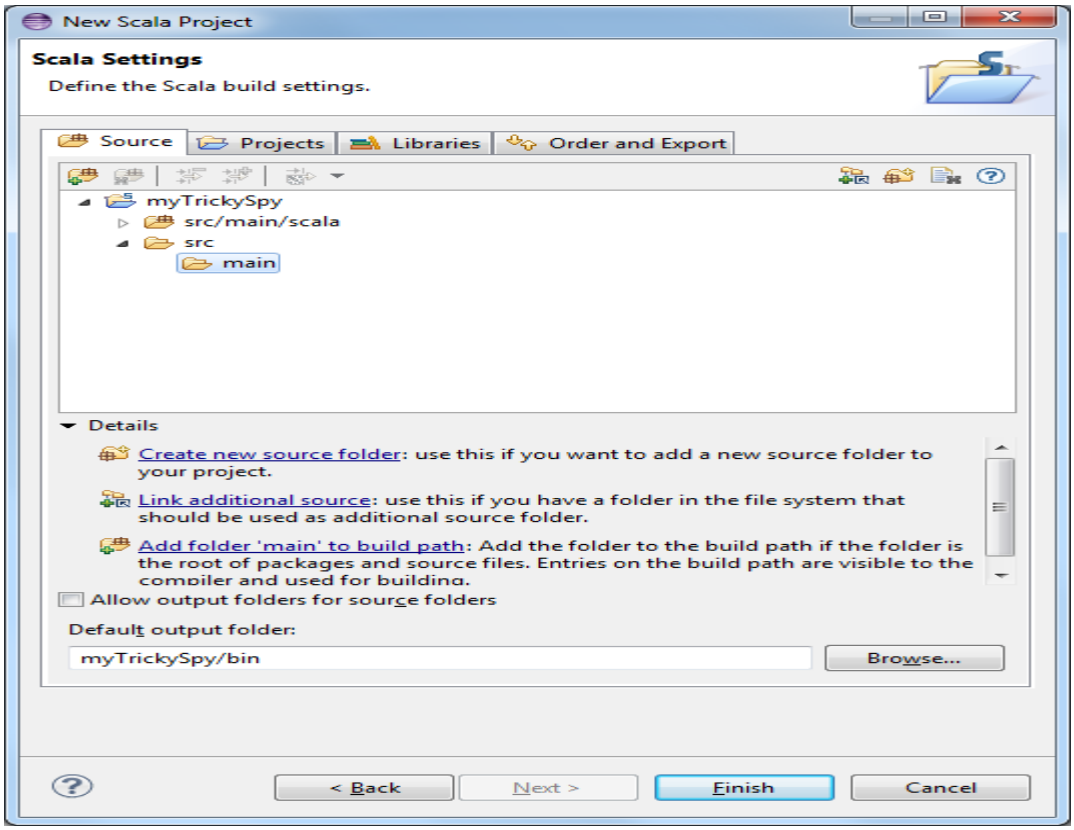
Create new source folder =>



Next =>

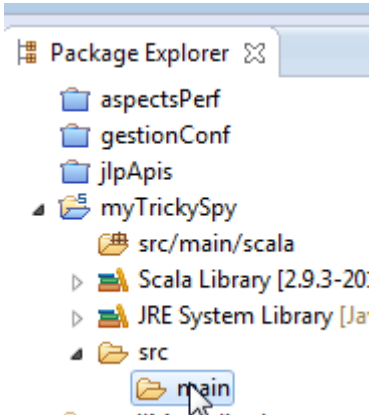


Nothing to fill; Finish =>

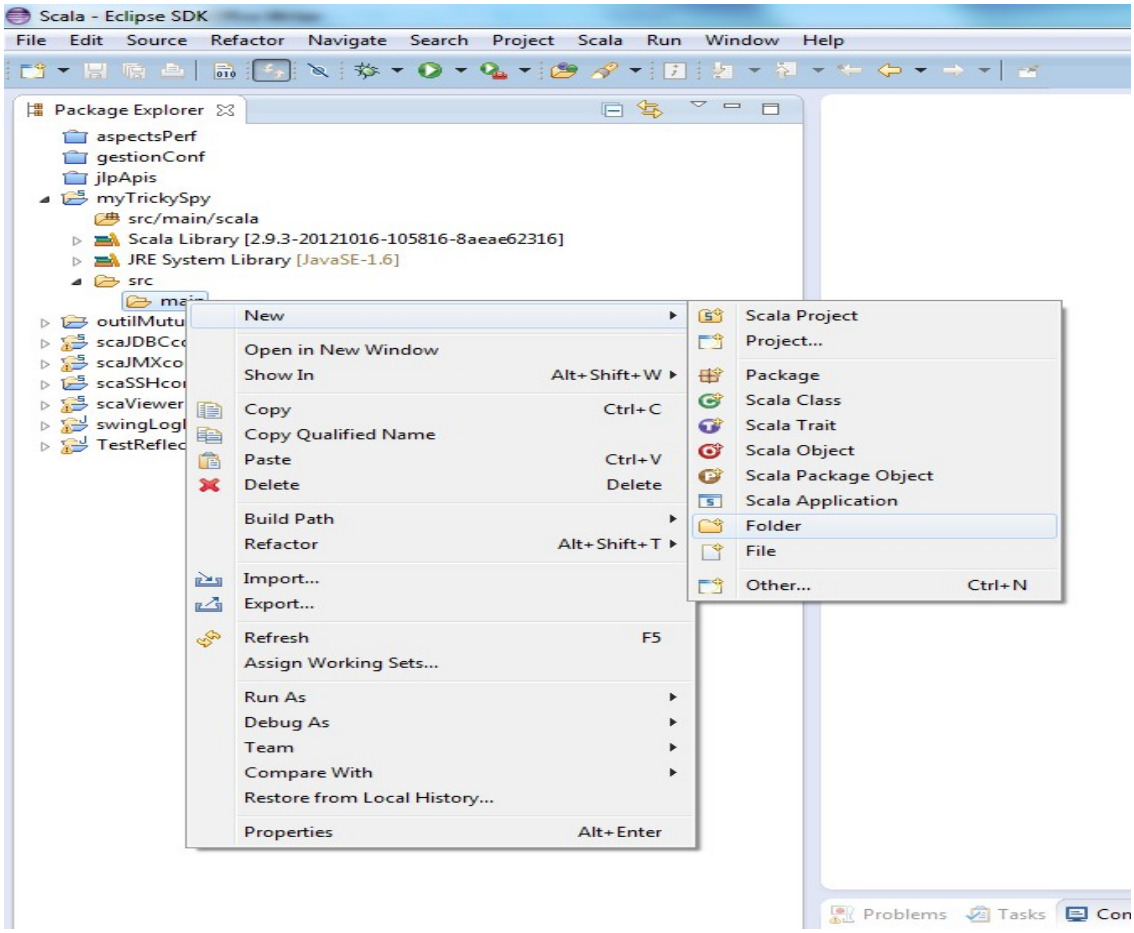


Finish

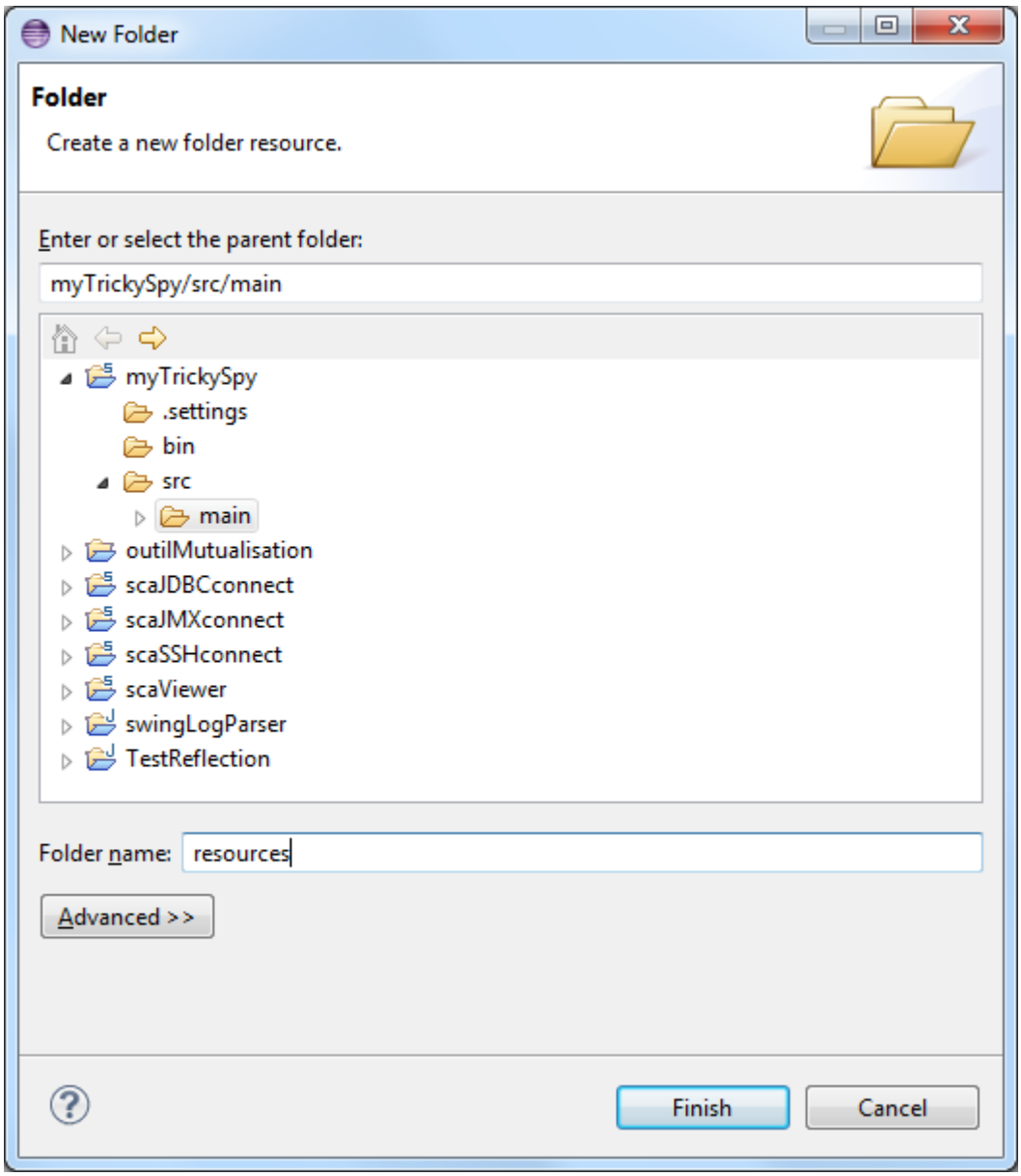
2.2. Create a “resources” folder in src/main/



Right click on src/**main** and add a folder **resources**



Choose New/Folder =>



and name it **resources** => Finish

If you wand to test, you can create the folders belows :

src/test/scala

src/test/resources

make src/test/scala as a source folder (in build path)

2.3. Create a folder "project" at the root of the project

Proceed as shown above.

Create a file **plugins.sbt** in the folder **project**.

Fill the file **plugins.sbt** with theses lines :

```
resolvers += Resolver.url("artifactory",
url("http://scalasbt.artifactoryonline.com/scalasbt/sbt-plugin-
releases"))(Resolver.ivyStylePatterns)
```

```
addSbtPlugin("com.eed3si9n" % "sbt-assembly" % "0.8.4")
```

This plug-in permits to generate a standalone executable jar.

https://github.com/sbt/sbt-assembly

2.4. Prepare a minimal build.sbt for your project

In this minimal configuration, there is no **Ivy/Maven** dependencies with external repositories.

To be add, with the requirement of your project.

build.sbt :

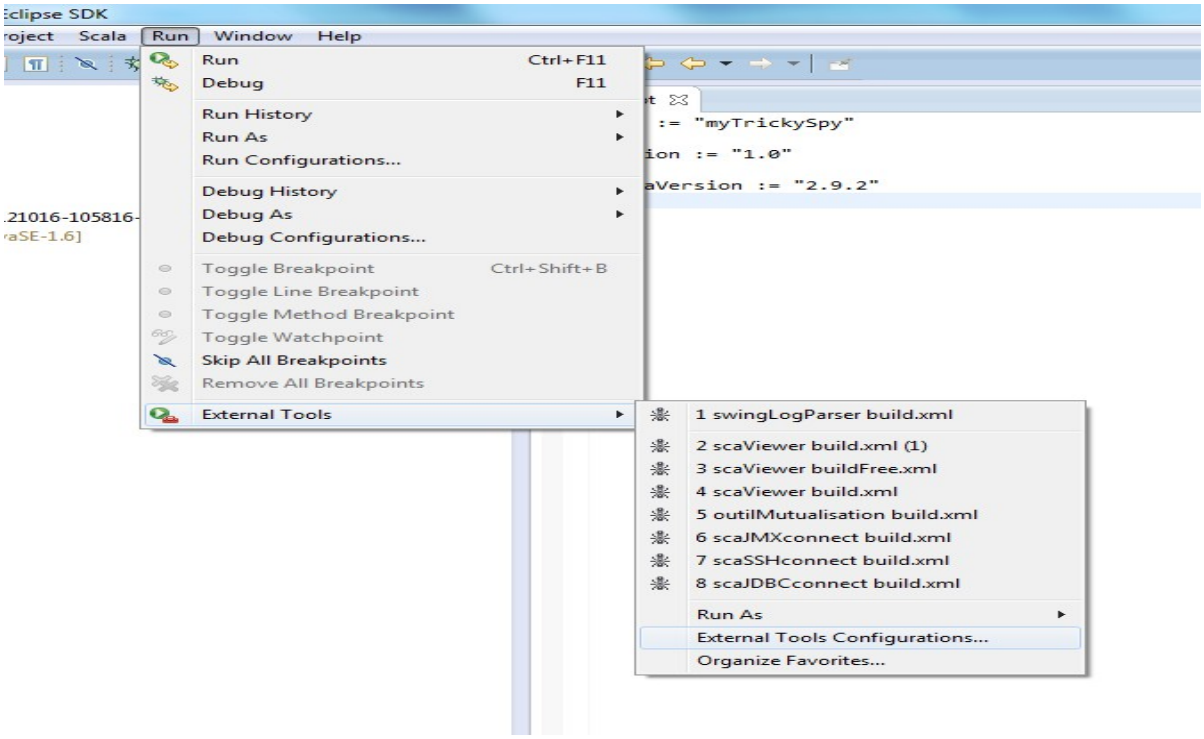
```
import AssemblyKeys._

name := "myTrickySpy"

version := "1.0"

scalaVersion := "2.9.2"

assemblySettings
```

Don't forget the new line after each statement !

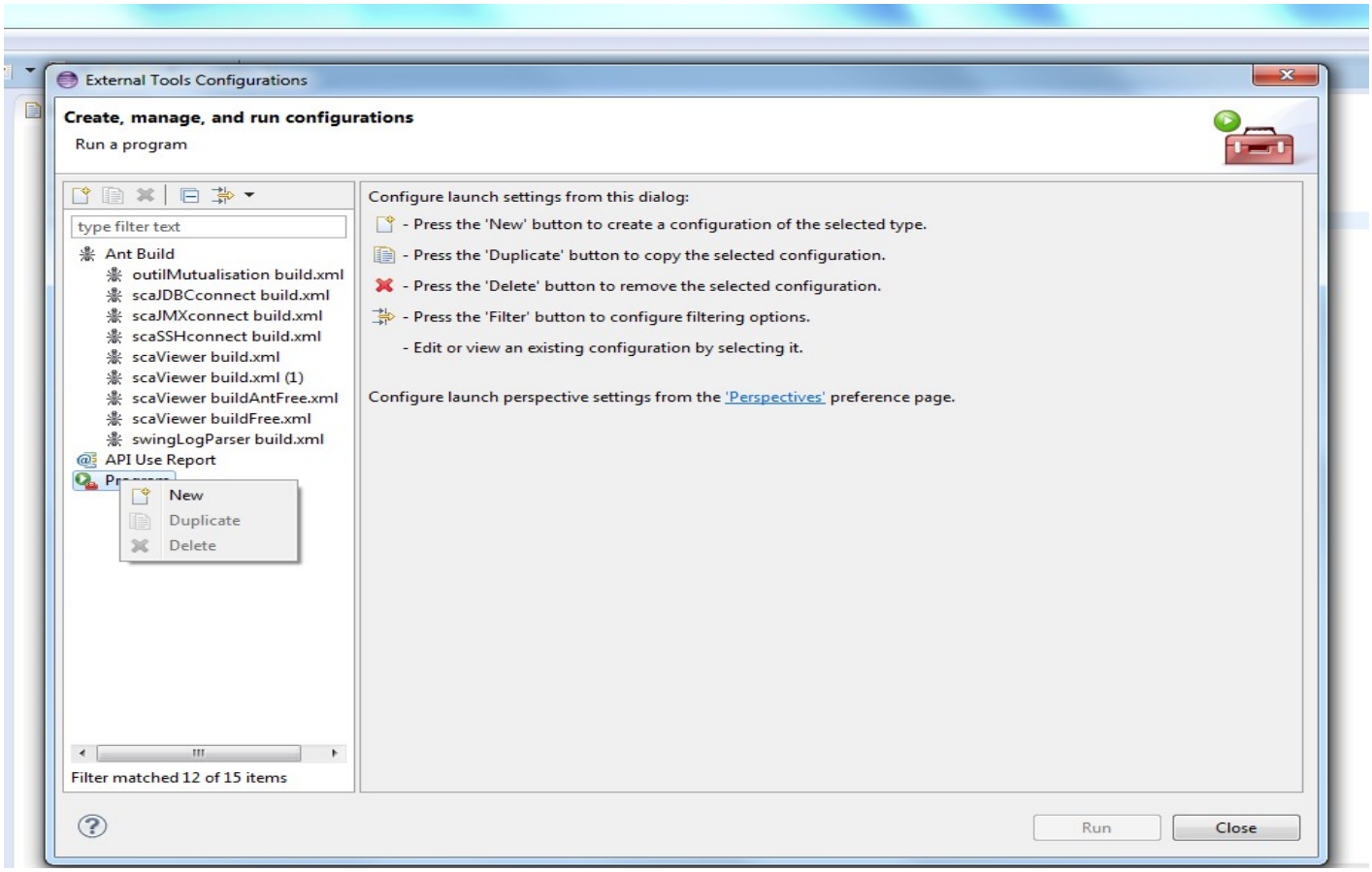
Put this file, in the root of the project directory, in Eclipse Workspace.

The statement **assemblySettings** permits to package a standalone executable jar.

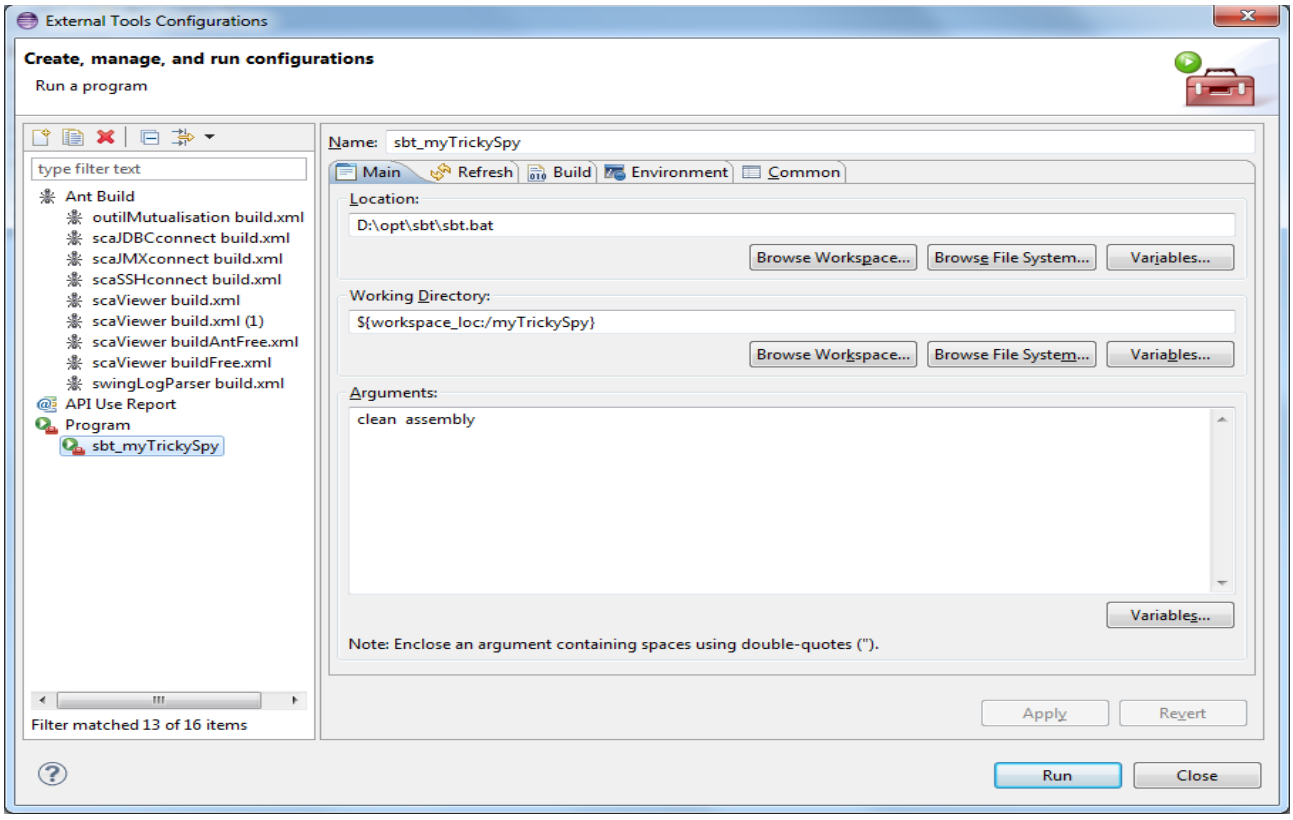
3. Configure sbt as external tool for Eclipse



Choose sub-menu External Tools Configuration =>



Program / Right Click / New =>



Fill the fields :

Name => **sbt_myTrickySpy**

Location => **full path to sbt.bat**

Working Directory => **the root directory of your Eclipse project**

Arguments => **clean assembly** (clean the target space and generate a jar with compiled classes and jar in src/main/resources). **assembly** is the task from the plugin sbt-assembly

https://github.com/sbt/sbt-assembly

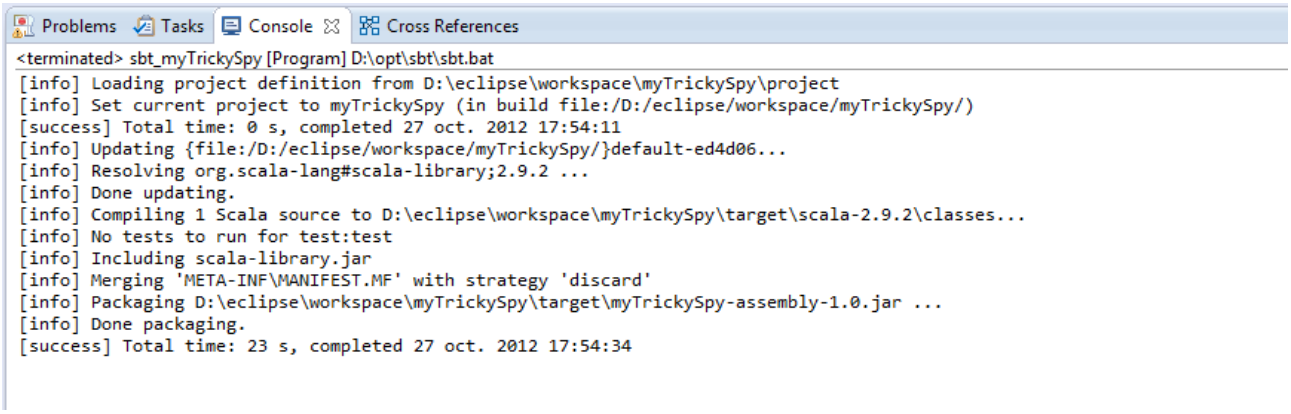
Other argument of sbt-core can be **run, compile, package** (you can create other program with these arguments)

In the tab **Common**, you can select Display in favorites menu / External Tools, so the Menu is permanent in External Tool sub-menu

4. The classic HelloWorld

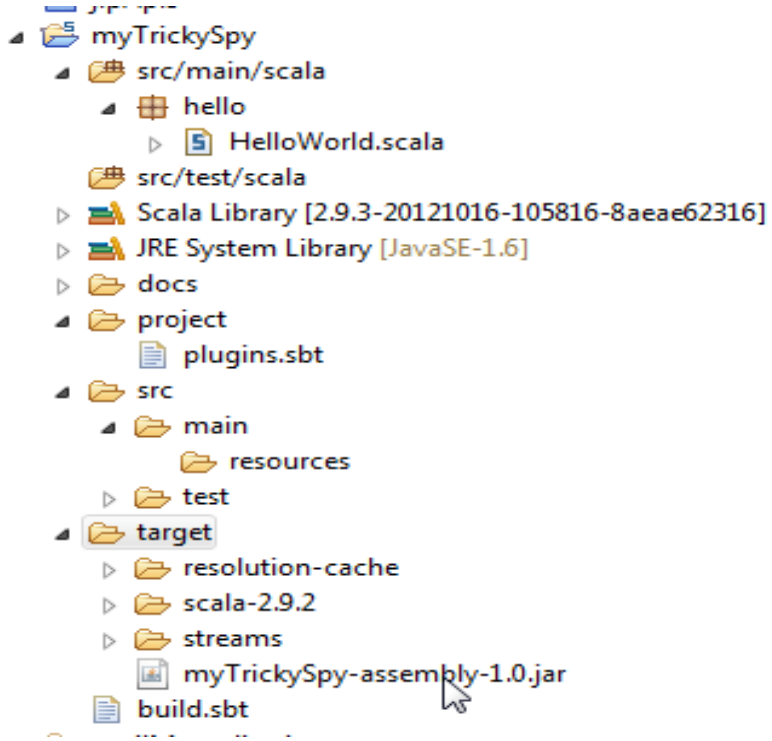
4.1. Create a Scala Object HelloWorld in package hello :

Choose menu **Run/External Tools/sbt_myTrickySpy**



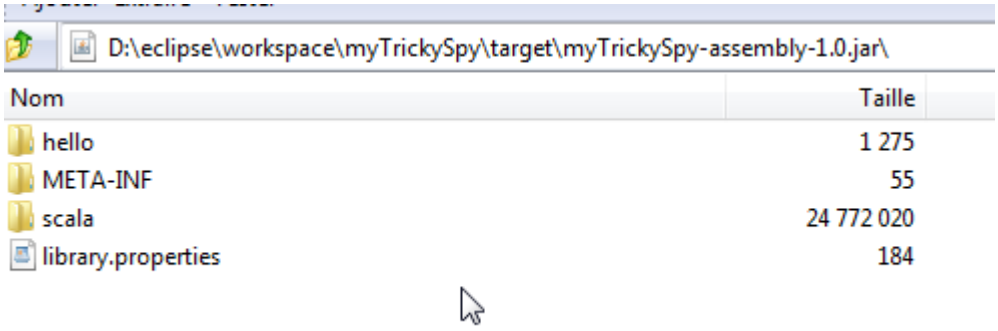
If the program compile correctly, you get a console as above.

The generated package jar is located in :



target/mytrickyspy_2.9.2-1.0.jar

Browsing this jar, you can see that the scala library is include.



Try it as standalone jar file :