

How to Set Up GECAMed on Eclipse

At first you need the GECAMed sources. If you like to download the sources, see the download link of the current version in the download section ([GECAMed Download](#)). Download the sources and unzip them into any folder.

Furthermore you will need an IDE. We suggest Eclipse, but you can use any other Java IDE. Setting up GECAMed for this IDE will probably be similar, but this guide is specially for Eclipse.

You can download the latest version at the Eclipse homepage: www.eclipse.org/downloads. We suggest to download Eclipse IDE for Java EE Developers.

Create a New Project

After you have downloaded the sources and started Eclipse, choose “File” - “New” - “Java Project” in Eclipse.

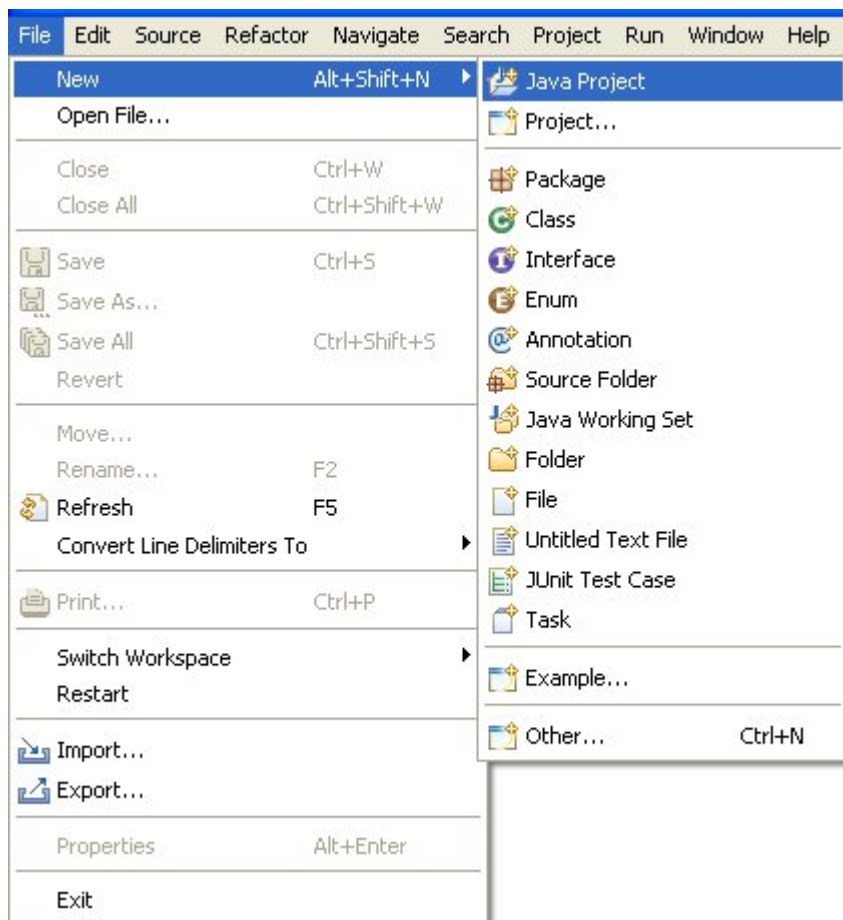


Fig. 1: Create New Java Project

The “New Java Project”-dialog will pop up. Give the project a name (e.g. “GECAMed”) and choose your JRE, respectively your JDK.

You need Java version 1.6 or higher, to support the GECAMed form editor, but the currently used JBoss server needs Java 1.5 to work correctly. Therefore you need to choose a **Java JDK version 1.6 or higher** as library and set the **compiler compatibility to version 1.5**.

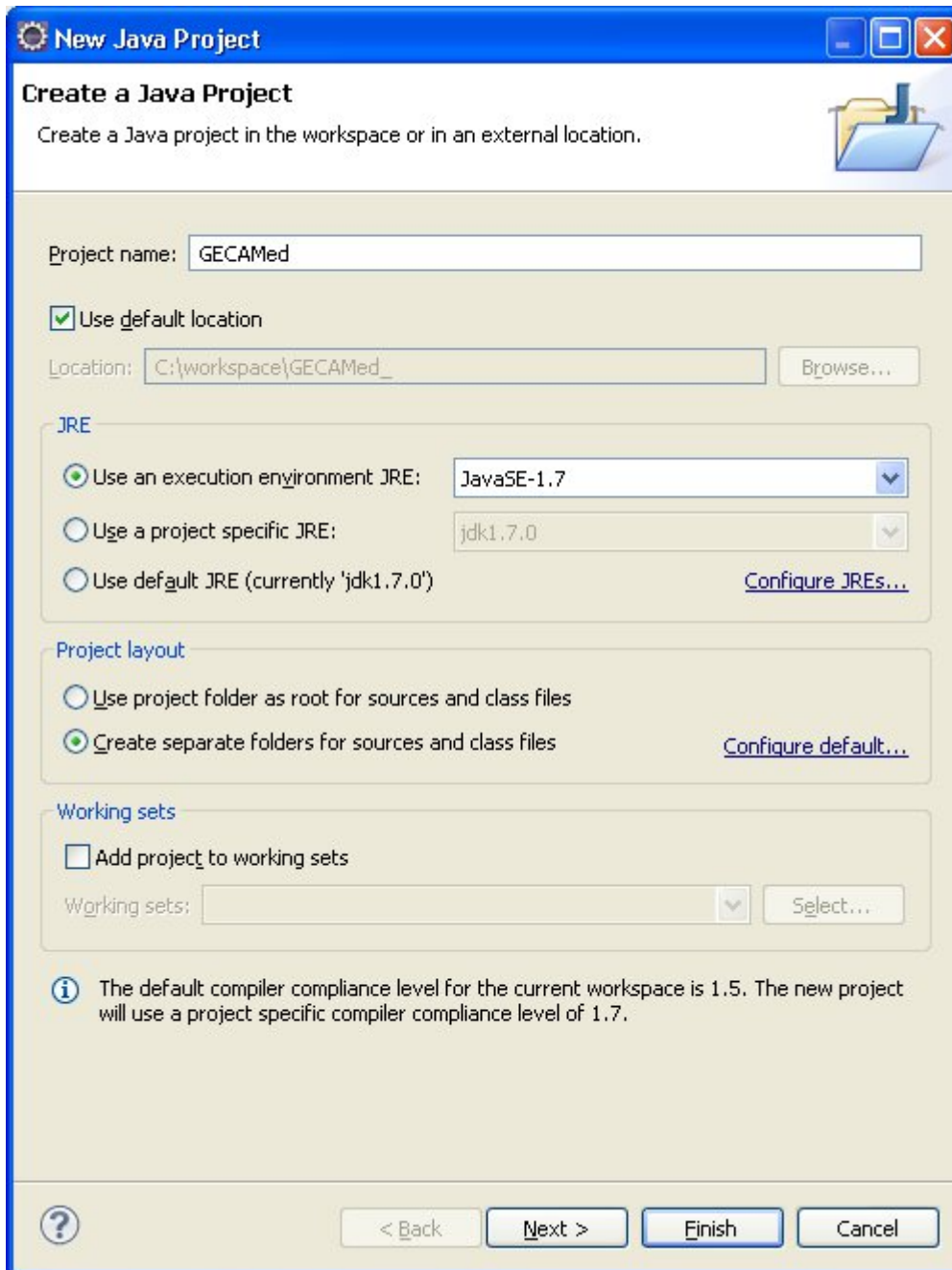


Fig. 2: New Java Project

You set the library under the point “JRE”. As mentioned, you need to choose Java version 1.6 or higher. Make sure the chosen library is a JDK and not only a JRE. To change the compiler compatibility you need to click on “Configure JREs...” under the point “JRE”. The “Preferences”-dialog will open. Choose “Compiler” on the left tree and set the “Compiler compliance level” to 1.5 and confirm the changes with “OK”.

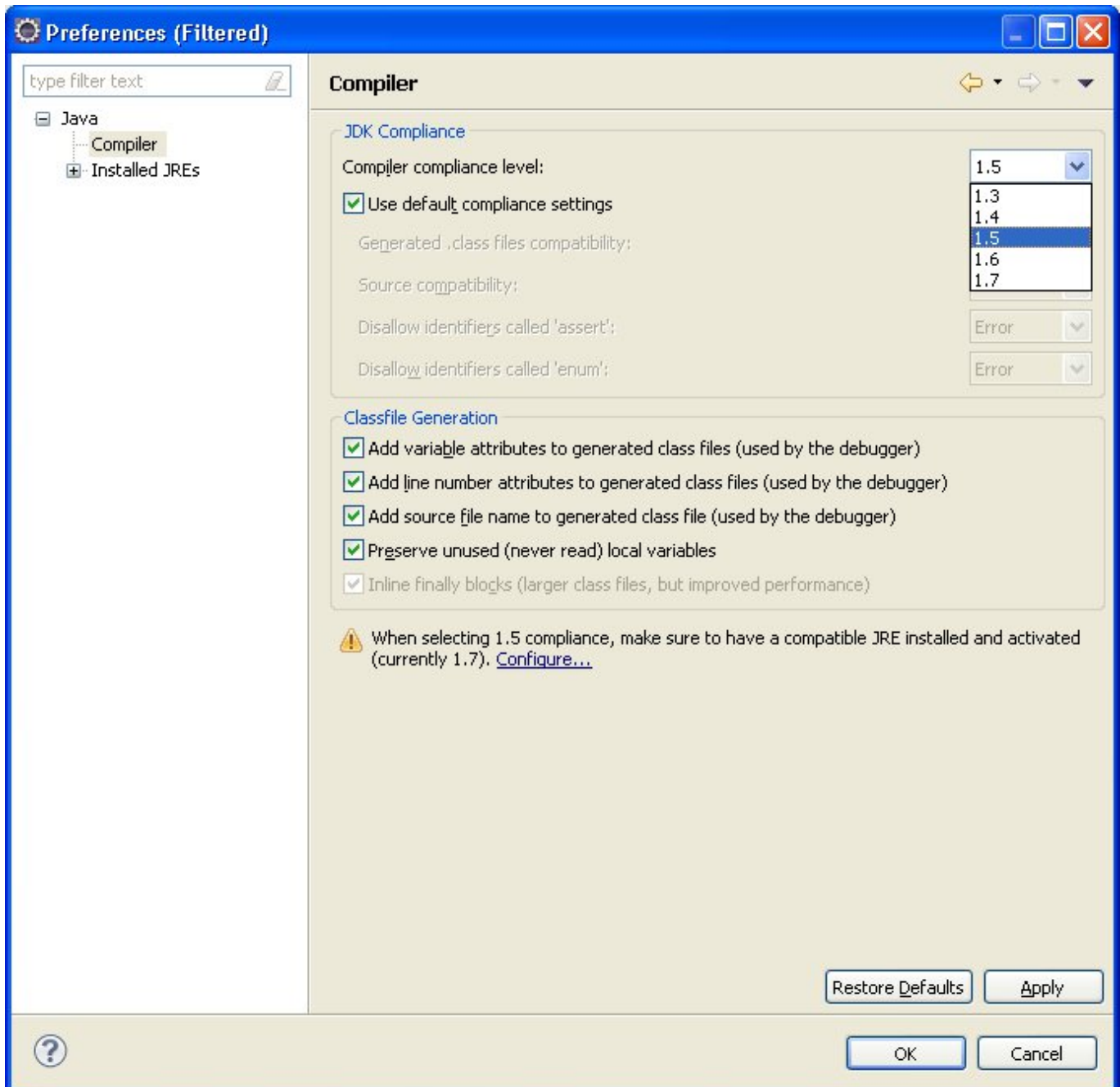


Fig. 1

Click the “*Finish*” button in the “*New Java Project*”-dialog and Eclipse creates a new Project with the given name and specified settings.

Integrate the GECAMed Sources

Open a file manager and go to the folder you unpacked the downloaded GECAMed sources to. Copy all unpacked folders into the project folder (do NOT use Eclipse to copy, but your file manager - the project folder location is “*«Eclipse workspace»/«project name»*”). You might need to replace the folder “*src*”.

Open the Package Explorer view (“*Window*” - “*Show View*” - “*Package Explorer*” - default in Java perspective: “*Window*” - “*Open Perspective*” - “*Java*”). You will find your new project there. You might need to refresh the project to see the copied data (right click on the project and click “*Refresh*” or

select the project and press F5).

If the folder “src” isn't yet declared as source folder, you need to do this. Therefore right click on the folder and select “Build Path” - “Use as Source Folder”. If there is any other folder declared as source folder, right click it and select “Build Path” - “Remove from Build Path”.

You may have noticed that there are a lot of errors in the sources. That is because the necessary libraries aren't added to the build path. Right click the project and select “Build Path” - “Configure Build Path”. The properties dialog will open, showing the Java build path properties. Choose the “Libraries” tab and click the “Add JARs...” button on the right. You will need to add the following JARs:

- /lib/* (all JARs in this folder and its subfolders)
- /lib_development/jboss/scheduler-plugin.jar
- /lib_development/drools-compiler-4.0.7.jar
- /lib_development/drools-core-4.0.7.jar
- /lib_development/junit-4.3.1.jar
- /lib_server/mailapi.jar

Notice that you can select several JARs in different folders and that the perspective will show only those JAR files, that are not yet added to the build path.

After you change the build path, eclipse will start rebuilding the project. As soon as Eclipse finished building, there shouldn't be any errors left. If there still are errors, check whether all necessary libraries are added to the build path and your used Java library is a JDK (not a JRE) version 1.6 or higher.

Set Up the JBoss and the Database

To start GECAMed out of Eclipse you need the GECAMed JBoss server and database running. The simplest way to create these things is to install GECAMed with the [GECAMed installer](#).

After you installed GECAMed, you need to stop the JBoss service and prevent it from starting automatically at system start up. The JBoss service is called “JBoss Application Server 4.2”, the process is simply called “java”. You need to do this because you want to start your own JBoss server. That is necessary to deploy your code and with that change the JBoss, when changes in the code affect the server part (like the entity beans). How to stop a service and prevent it from running at startup depends on your OS.

After the JBoss service is stopped, you need to copy the folder “jboss-4.0.5.GA” from the GECAMed install folder to your project folder in your Eclipse workspace. After everything is copied, refresh your project again.



You must not delete or move the GECAMed installation folder, as the JBoss refers to the Java and the scripts in that folder. Furthermore the PostgreSQL database is installed there.

You can now start the JBoss server. To do so, just go into the JBoss folder, you copied into the project folder. Go into the folder “bin” and run the “run.bat” for Windows or the “run.sh” for Mac and Linux.

Start GECAMed

When the JBoss is running, you can start GECAMed. To start GECAMed you need to open the class `lu.tudor.santec.gecamed.core.gui.LoginScreen`. That's the class, that contains the main method. You can launch it by clicking "Run" - "Run As" - "Java Application".

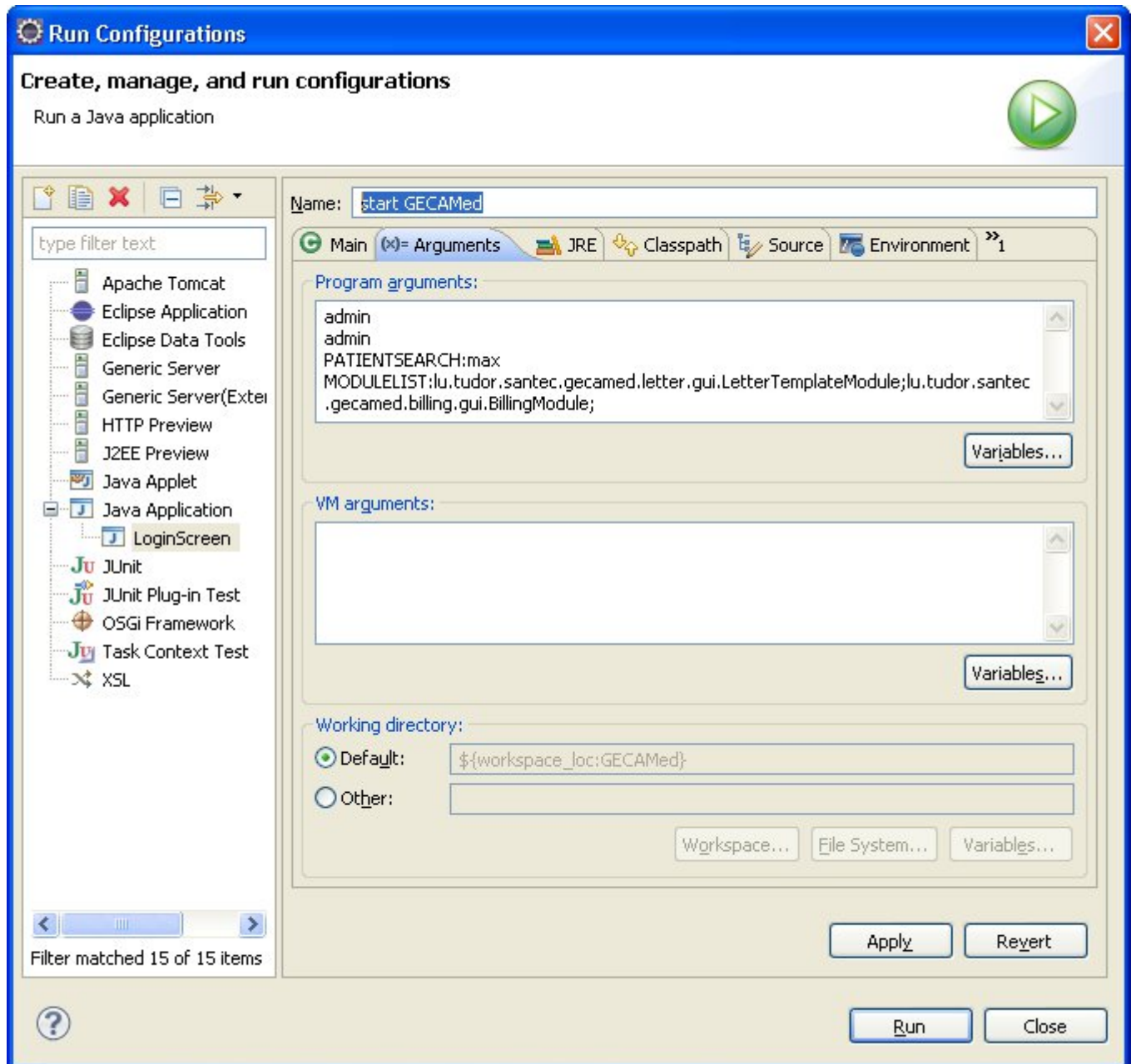


Fig. 4: Run Configurations

As you need to start GECAMed several times during the development there is the option to accelerate starting by defining arguments.

You can configure the arguments in Eclipse in the *Run Configurations* ("Run" - "Run Configurations ..."). Select the Java application you want to configure on the left side. The GECAMed you started will be called "LoginScreen". On the right side you can give it another name, like "start GECAMed". In the tab "Arguments" you can specify the arguments given. The following things can be configured with the arguments:

- **Login name:** The login name written into the login screen of GECAMed. *This must be the first*

argument.

- **Password:** The password that belongs to the login name. *This must be the second argument.* If this argument is set, GECAMed will login automatically, assumed that the user name and the password are correct.
- **Patient search:** Write "PATIENTSEARCH:" and your search string behind it (without spaces). That is what will be typed into the patient search at GECAMed startup.
- **Loaded modules:** Write "MODULELIST:" and the class name of the modules (including the package structure) you want to load. You can define several modules by separating them with a semicolon. Only the default modules (like patient and admin module) and the specified modules will be loaded.


Deploy Changes on the Server

Via Apache Ant you can create the EAR-file for the JBoss, that contains the GECAMed server application. The server doesn't contain the whole code, as the GUI-part is obviously not needed by the server.

Every module can have the "ejb", "util", "gui", "sql" packages and more. But only the code contained in the "ejb" and "util" package (and its subpackages) is used in the server EAR-file. Therefore you need to rebuild the EAR-file whenever changes in one of these packages are made. This means, build the binaries, put them into an archive and copy the archive into the JBoss deploy folder. This is called deploying and can be done automatically by Apache Ant.

Configure the Ant Build File

The Ant plug in is delivered with the *Eclipse version for Java EE developers*. If you haven't got the Ant plug in, you need to install it.

Open the Ant view in Eclipse ("Window" - "Show View" - "Ant"). Click on the "Add Buildfiles"  icon in the Ant view. Open your project folder in the opening dialog and double click the "build_sources.xml" file in the "installer" folder. A new build file was added to your Ant view, called "GECAMed Installer". Expand it and you will see the different tasks that can be done with this build file. The task needed to deploy the server EAR is called "build+deploy EAR File".

If you start the task, you will notice that it will fail, because the file "package-build.xml" is missing. This file specifies, which packages are added. You can download and use this file: [packaging-build](#). Copy it directly into your project folder.

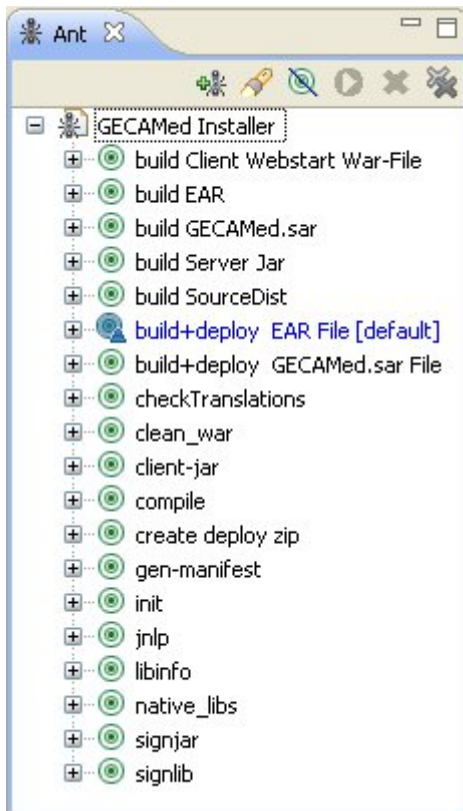


Fig. 5: Ant View

The Ant script will copy the EAR file by default into the folder "*«Eclipse workspace»/«project folder»/jboss/server/default/deploy/*". As the JBoss folder we are using is called "*jboss-4.0.5.GA*" you can either change the folder name in the script (at 3 places, as other scripts also refer to that folder) or change the name of the folder itself. Now you should be able to deploy correctly and start developing GECAMed.

Sign Your JARs for the GECAMed-Webstart

For testing, you can start the GECAMed Client via Eclipse. To be able to launch GECAMed via Java webstart, you need to create and deploy a war-file. This is also done by the *build_sources.xml* Ant script.

To create and deploy a new war-file start the task "*build Client Webstart War-File*". This will unsign an sign your client libraries and GECAMed-Jar, as this is necessary for the webstart.

To sign your Jars, you need to enter your keystore data in the Ant script.

To do so edit the script and replace the "XXX" with the appropriate values.

```
<!-- sign the libs -->
<target name="signlib">
  <signjar jar="${file}"
    keystore="XXX"
    alias="XXX"
    storepass="XXX"
  />
</target>

<!-- clean the war dir-->
<target name="clean_war">
```

```
<delete includeemptydirs="true">  
  <fileset dir="${war.dir}" includes="**/*" />  
</delete>  
</target>
```

Replace the "XXX" value of "keystore" and enter where the keystore file is located. "alias" is the alias of the signer to show and "storepass" is the password, to encode the keystore.

How to create my own keystore



Oracle is increasing the security settings for the Java Web Start. Therefore there is a warning that always has to be confirmed, when starting GECAMed, telling that the publisher of the application is UNKNOWN and that **applications of unknown publishers will be blocked in future Java releases**.

Therefore it will become necessary to get a key from an official signer.

You can use the Java tool "keytool" to create your own keystore. Using Windows, you'll find it in the *bin* folder of your JRE. On Linux or Mac OS the command is system-wide available.

Use it like this:

```
keytool -genkey -keyalg RSA -alias <<ALIAS>> -keystore <<KEYSTORE FILE  
NAME>> -storepass <<PASSWORD>>
```

You will be asked to enter an additional password and some data, that will be shown, when the GECAMed user needs to accept your certificate (like name, company, country, etc.).



Both passwords must be identical, otherwise the signing may fail.

Copy the keystore file into your workspace and refer to it in the Ant script ("keystore" attribute of the signing target). Also set the entered password and alias in the Ant script.

From:

<https://gecamed.lu/> - **GECAMed - Gestion de Cabinets Médicaux**

Permanent link:

https://gecamed.lu/faq/developement/set_up_gecamed_on_eclipse

Last update: **2019/12/09 11:19**

