Genius Project Setup

W.Pasman, 5 september 2017

## Introduction

This document describes how to to (re)build the Genius components. The main components are described.

## Components

Table 1 shows the four components of the Genius system. All components are available through svn: https://ii.tudelft.nl/svn/nego. You can also browse the sources, using http://ii.tudelft.nl/trac/negotiation/<project>.

Table . The four compoennts of the Genius system.

|  |  |
| --- | --- |
| component | description |
| core | actual genius core: classes for bids, profiles, negotiation sessions, etc |
| agents | All agents collected over the years: agents from competitions, assignments, etc. Some of these are included with Genius. |
| domains | Sample domains and utility profiles collected over the years. Some of these are included with Genius. |
| NegotiationGUI | wraps together different components for a genius zip file distribution |

All components are configured to be a separate "eclipse project" in Eclipse.

## The compilation circus

The various Genius components have dependencies on another, as shown in . This means that if you change a component down the line, all higher layered components need to be updated as well, hence the "compilation circus". Therefore always start building at the bottom of the dependency graph and work upwards.

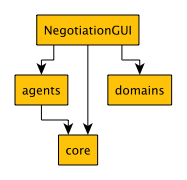


Figure . Dependency graph of components.

To start the build of a component, right click on the build.xml inside that project and select "Run As/Ant build". This runs a script that updates the files in the target/ directory of the same project.

After the files have been built, you need to copy the updated files in the component's target directory to the dep/ directory of the dependent projects, as show in Table 2.

Table . Delivered files and required copy actions after rebuilding them.

|  |  |  |
| --- | --- | --- |
| **component** | **build delivers in component/target/** | **copy to, and then build:** |
| core | GeniusCore-javadoc.jar  GeniusCore.jar  ProtocolsRepo.jar | agents/dep  NegotiationGUI/dep |
| agents | AgentRepo.jar  Agents.jar  BoaRepo.jar  PartyRepo.jar | NegotiationGUI/dep |
| domains | Domains.jar  DomainsRepo.jar | NegotiationGUI/dep |
| NegotiationGUI | Genius-X.Y.Z.zip  also writes to debug/ | - |

## Building in Eclipse

Eclipse can automatically handle these component dependencies. That way, you can test run and debug your code changes without needing to do the whole recompilation circus. Then, you only need to run the circus when you actually need to generate the jar files (eg, for distribution).

To set up Eclipse, do the following

* download all components from svn as mentioned above.
* Modify the project settings as follows
  + In NegotiatorGUI component,
    - remove Agents.jar and GeniusCore.jar from the Libraries (Project/Properties/Java Build Path/Libraries)
    - Add project dependencies for agents/ and core/ (Project/Properties/Java Build Path/Projects).
  + In agents component similarly:
    - Remove GeniusCore.jar from the agents project Libraries (Project/Properties/Java Build Path/Libraries)
    - Add project dependencies for core/ (Project/Properties/Java Build Path/Projects).

Eclipse can not handle changes in domains, because these are XML files (not java code). So if you change anything here, you need to do the copy/ build actions anyway, plus the Run/Debug preparation below.

For quick testing of changes in domains, you can also edit your runtime profiles (typically in NegotiationGUI/debug) manually.

You still have to follow the "Debugging with Eclipse" steps below.

## Run/Debug preparation

To run or debug the NegotiationGUI or stand-alone runners, you need to build a run/debug environment. This extracts a number of prepared files in specific locations, which is needed for the core.

1. Run build.xml with the target "debug":
   1. right click on build
   2. run as/"Ant Build..."
   3. check only 'debug'
   4. run the script
2. Create run script in Eclipse
   1. select the NegotiationGUI project in package explorer/navigator
   2. click on debug icon down arrow to open the options
   3. click "Debug Configurations..."
   4. click "new launch configuration" (the icon in top left)
   5. Switch to Main tab
   6. check the project is NegotiationGUI
   7. enter "negotiator.gui.NegoGUIApp" as Main class
   8. switch to the Arguments tab
   9. in "Working Directory" enter "${workspace\_loc:NegotiatorGUI/debug}"
   10. switch to JRE tab and check Java SE 7 is selected
3. 2b. Create run script in IntelliJ
   1. Run/Edit Configurations...
   2. + and select "Application"
   3. enter "negotiator.gui.NegoGUIApp" as Main class
   4. in "Working Directory" select the NegotiatorGUI/debug directory
   5. click "Use classpath of module: and select "NegotiatorGUI voor AAMAS competitie"
   6. JRE: select "1.7". higher might work ok.

**NOTICE**!

The files in the debug folder that are generated by this preparation are ONLY FOR DEBUGGING. Modifying these affects your testing setup but has no effect on the release version of Genius-X.Y.Z.zip that is build by the build script.

## Running the junit tests

To run the junit tests fo NegotiatorGUI, you need to prepare the debug/ directory as described above. The other components do not need preparation.

Then, you can run the junit tests by right clicking on the eclipse project and selecting Run As/JUnit Test.