

JaCaMo Eclipse Plugin

Maicon Rafael Zatelli
September 2014

Contents

[Installation](#)
[Agent creation](#)
[Internal Action](#)
[Artifacts](#)
[Organization](#)
[Protocols](#)
[Importing projects](#)
[Exporting projects](#)
[Update](#)
[Uninstall](#)

Installation

In order to install the [JaCaMo](#) plug-in for Eclipse you should follow the steps below and have Eclipse version 4.2.0 (Juno) or greater (**Only Java-EE or Committers**).

Step 1

Download the latest version of JaCaMo at the link: <http://sourceforge.net/projects/jacamo/files/>

Step 2

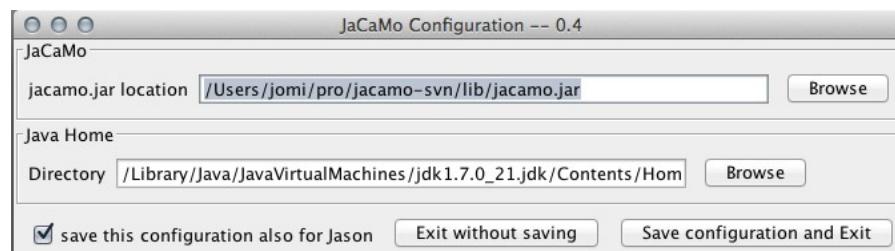
After the download, unpack it in any directory of your machine.

Step 3

Even if you had run JaCaMo on your computer before, execute the file "lib/jacamo.jar" by double clicking on it. You also could execute this file by with the following command:

```
java -jar lib/jacamo.jar
```

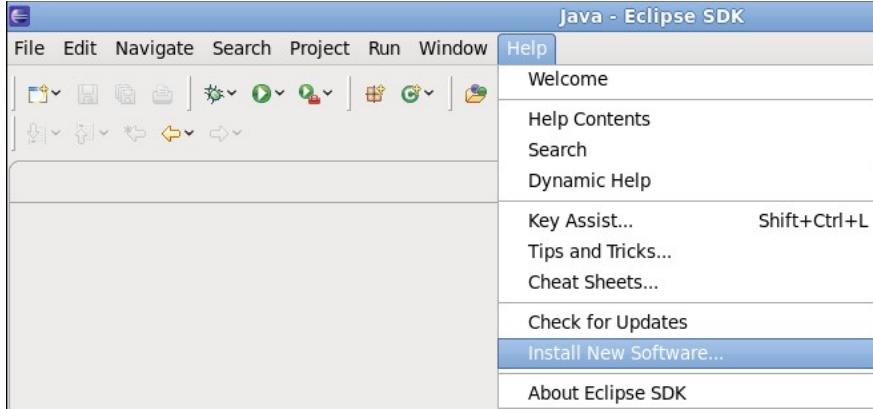
The following window appears. We suggest that you only change the "Java Home" directory (notice that it has to point to a JDK installation).



*** If your internet access pass by a proxy, please click [here](#) to see how to configure the proxy in your Eclipse before to continue the installation of the plugin.

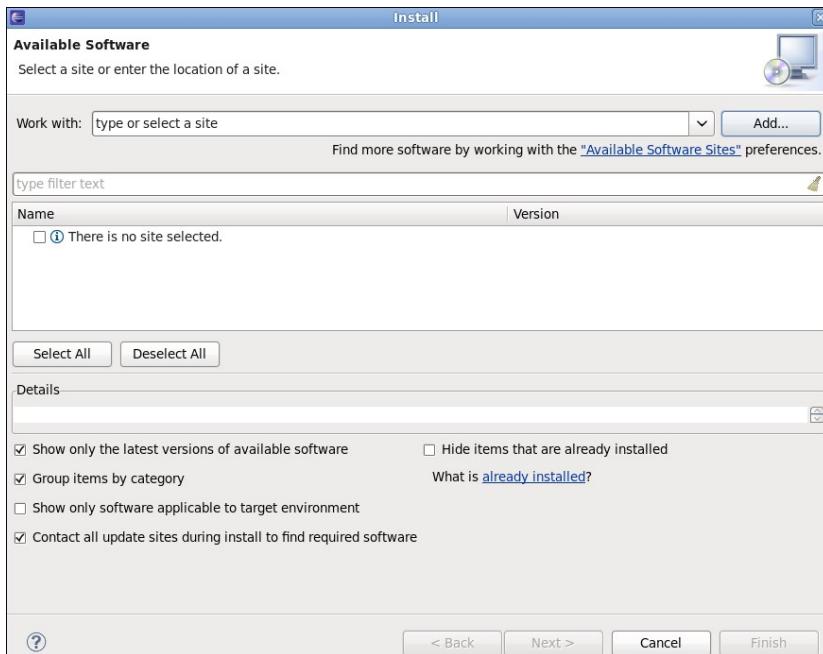
Step 4

Finally you could install the JaCaMo plugin for Eclipse opening the Eclipse platform and going to the option "Install New Software..." at the "Help" menu:



Step 5

So, the following window will appear.



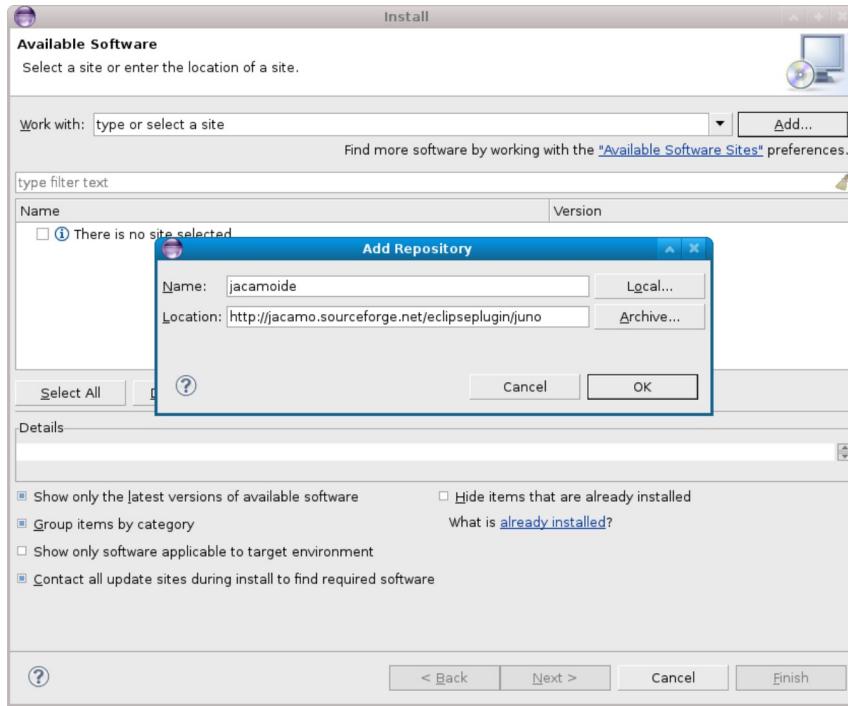
Step 6

Click over the "Add" button, and fill out the form as shown in the next figure. The parameters are

Name: jacamoide
 Location (for JaCaMo >= 0.4 and Eclipse* Juno/Kepler/Luna/Mars/Neon/Oxygen/Photon/2018/2019):
 http://jacamo.sourceforge.net/eclipseplugin/juno
 Location (for JaCaMo <= 0.3a and Eclipse* Juno/Kepler/Luna/Mars/Neon/Oxygen/Photon/2018/2019):
 http://jacamo.sourceforge.net/eclipseplugin/juno_old

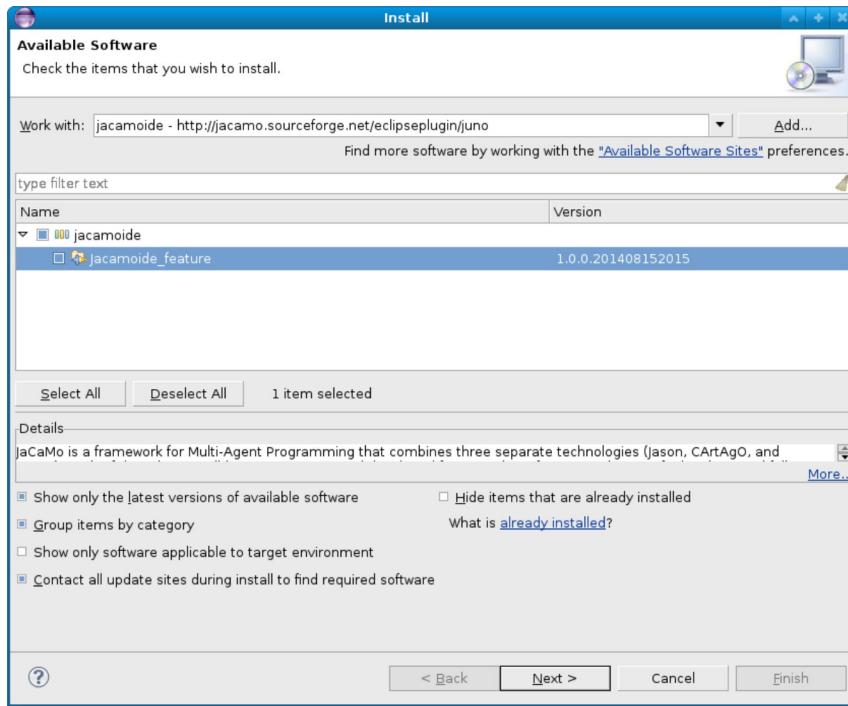
* Only Java-EE or Committers.

To finish, click on the "OK" button.



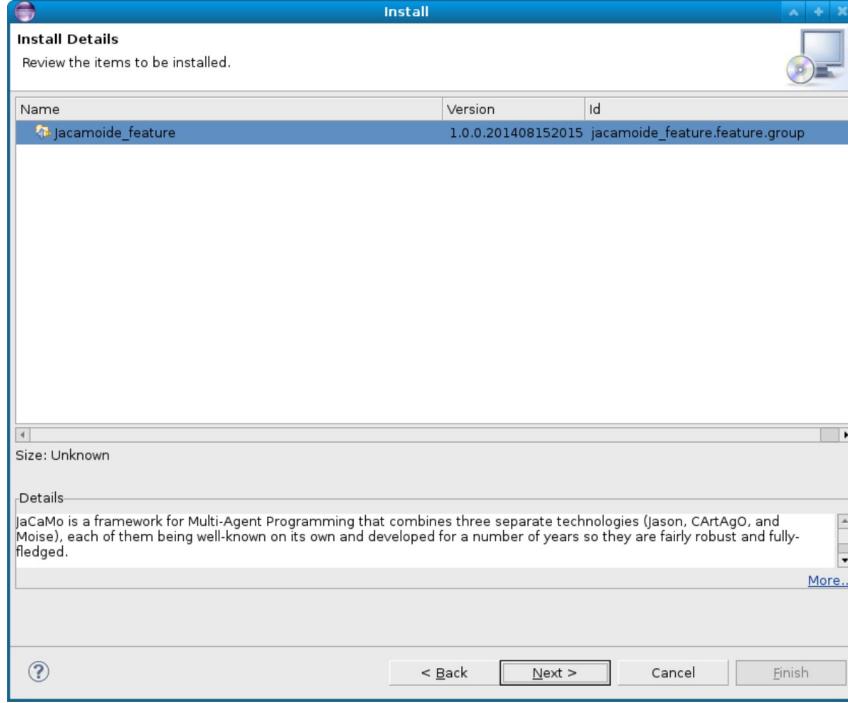
Step 7

Tick the option "jacamoide" and then press the "next" button. So, you have to wait a moment while Eclipse searches the dependences.



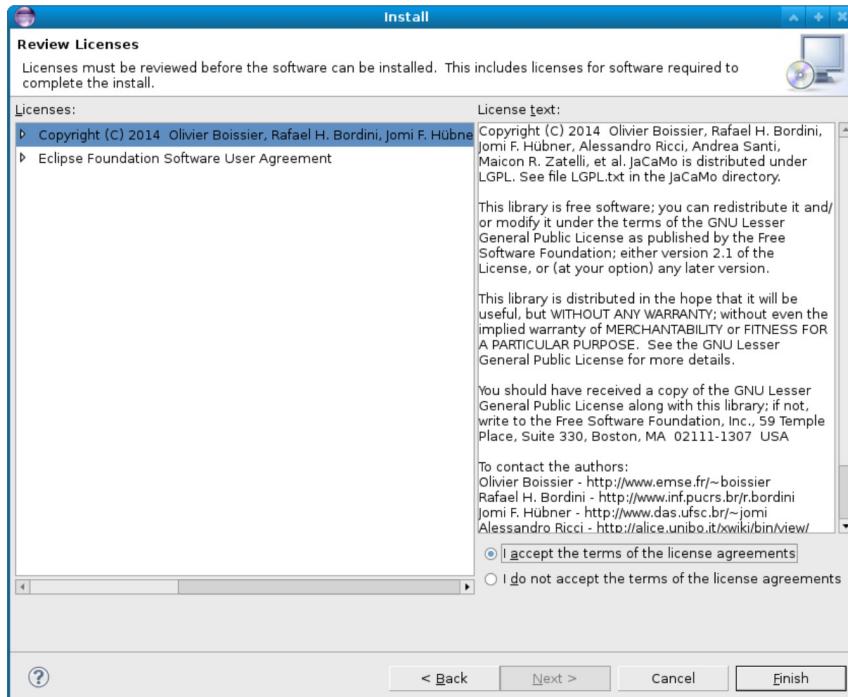
Step 8

In the next windows just press the "next" button again.



Step 9

The last window that will be shown for you is about the license. Tick the option "I accept the terms of the license agreements" and then press the "finish" button. Then the installation is proceeded, it could take several minutes, so please wait.



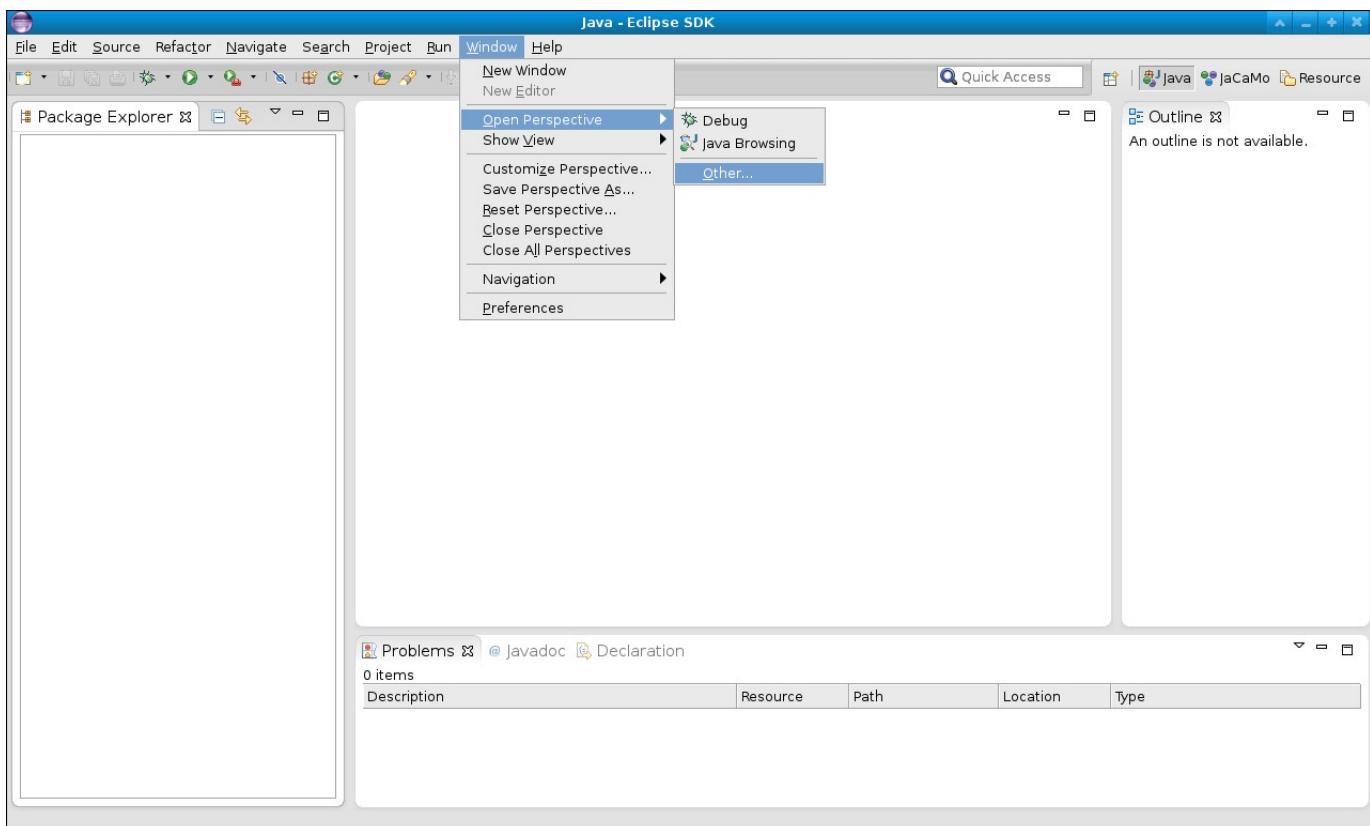
Step 10

In the end of these process will be shown a window in order to complete the installation. Choose the option "Restart Now".



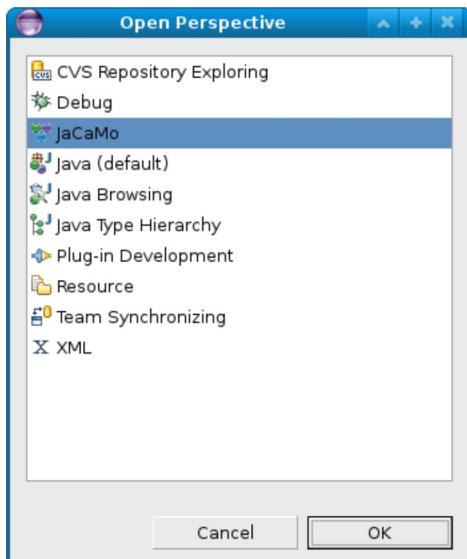
Step 11

Now, you can open the JaCaMo perspective. You could do it in the menu (Window > Open Perspective > Other)



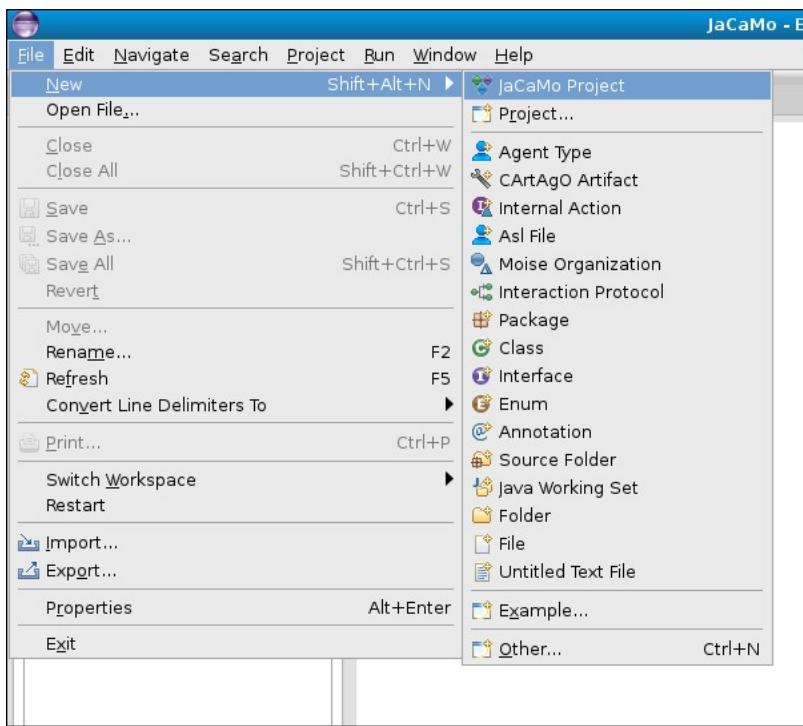
Step 12

Then, you can select JaCaMo and press "OK".



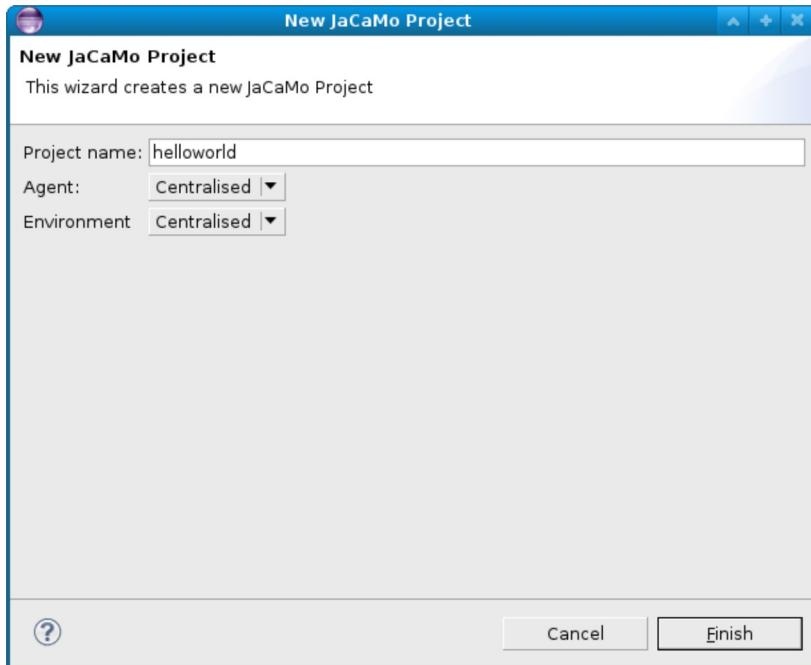
Step 13

Finally, you have all set. In order to test the installation of the plug-in, we suggest the creation of a simple hello world project. You could do it in the menu (File > New > JaCaMo Project) or (File > New > Other > JaCaMo > JaCaMo Project).



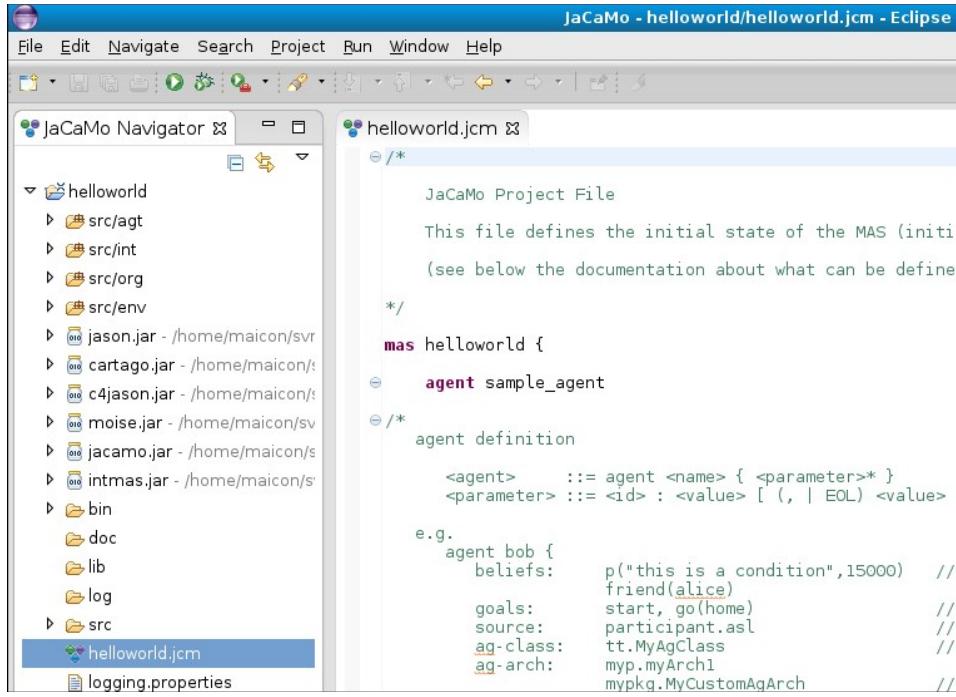
Step 14

Fill out the field "Project name" and press the "Finish" button.



Step 15

If everything is fine, you will have your first project created!



The screenshot shows the Eclipse IDE interface with the title "JaCaMo - helloworld/helloworld.jcm - Eclipse". The left pane is the "JaCaMo Navigator" showing the project structure of "helloworld" with files like "src/agt", "src/int", "src/org", "src/env", "jason.jar", "cartago.jar", "c4jason.jar", "moise.jar", "jacamo.jar", "intmas.jar", "bin", "doc", "lib", "log", and "helloworld.jcm". The right pane is the "helloworld.jcm" editor showing the JaCaMo Project File content. The code defines a sample agent with beliefs, goals, and source, and specifies its class and architecture.

```

/*
JaCaMo Project File

This file defines the initial state of the MAS (initial
(see below the documentation about what can be defined

*/
mas helloworld {
    agent sample_agent
    /*
        agent definition

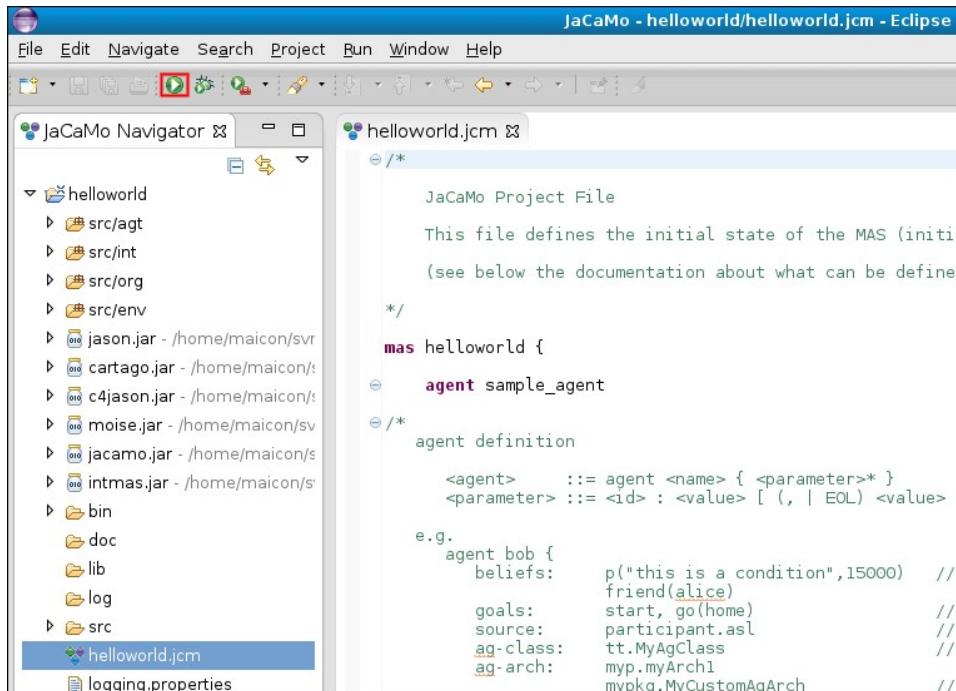
        <agent> ::= agent <name> { <parameter>* }
        <parameter> ::= <id> : <value> [ (, | EOL) <value> ]

        e.g.
        agent bob {
            beliefs: p("this is a condition",15000) // friend(alice)
            goals: start, go(home) // ...
            source: participant.asl // ...
            ag-class: tt.MyAgClass // ...
            ag-arch: myp.myArch1 mypkg.MyCustomAgArch // ...
    */
}

```

Step 16

Now you can run the application by pressing the Run button.



The screenshot shows the Eclipse IDE interface with the title "JaCaMo - helloworld/helloworld.jcm - Eclipse". The left pane is the "JaCaMo Navigator" showing the project structure of "helloworld" with files like "src/agt", "src/int", "src/org", "src/env", "jason.jar", "cartago.jar", "c4jason.jar", "moise.jar", "jacamo.jar", "intmas.jar", "bin", "doc", "lib", "log", and "helloworld.jcm". The right pane is the "helloworld.jcm" editor showing the JaCaMo Project File content. The code defines a sample agent with beliefs, goals, and source, and specifies its class and architecture.

```

/*
JaCaMo Project File

This file defines the initial state of the MAS (initial
(see below the documentation about what can be defined

*/
mas helloworld {
    agent sample_agent
    /*
        agent definition

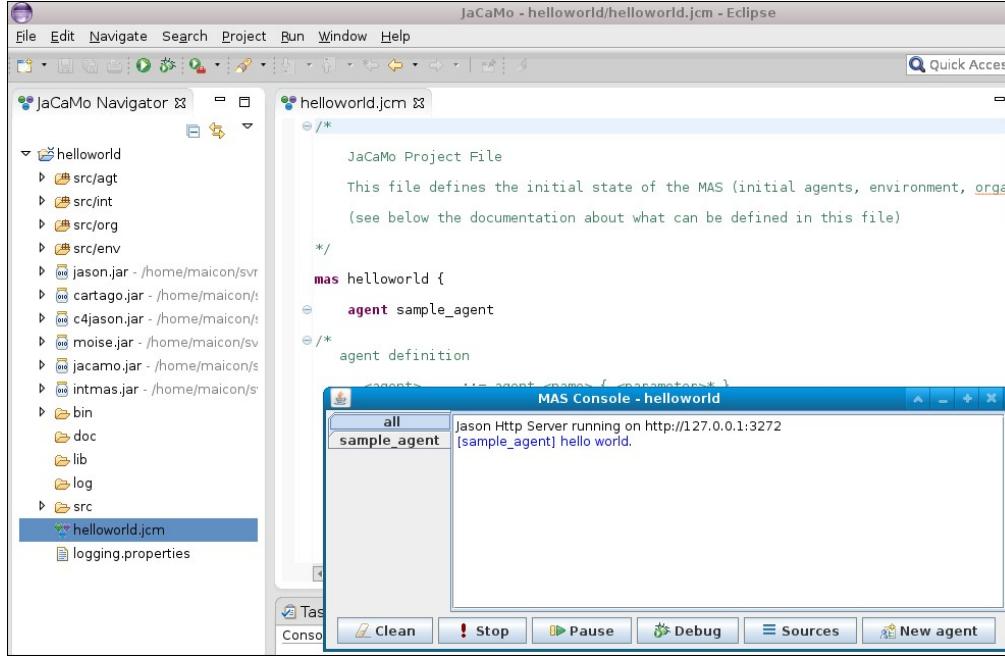
        <agent> ::= agent <name> { <parameter>* }
        <parameter> ::= <id> : <value> [ (, | EOL) <value> ]

        e.g.
        agent bob {
            beliefs: p("this is a condition",15000) // friend(alice)
            goals: start, go(home) // ...
            source: participant.asl // ...
            ag-class: tt.MyAgClass // ...
            ag-arch: myp.myArch1 mypkg.MyCustomAgArch // ...
    */
}

```

Step 17

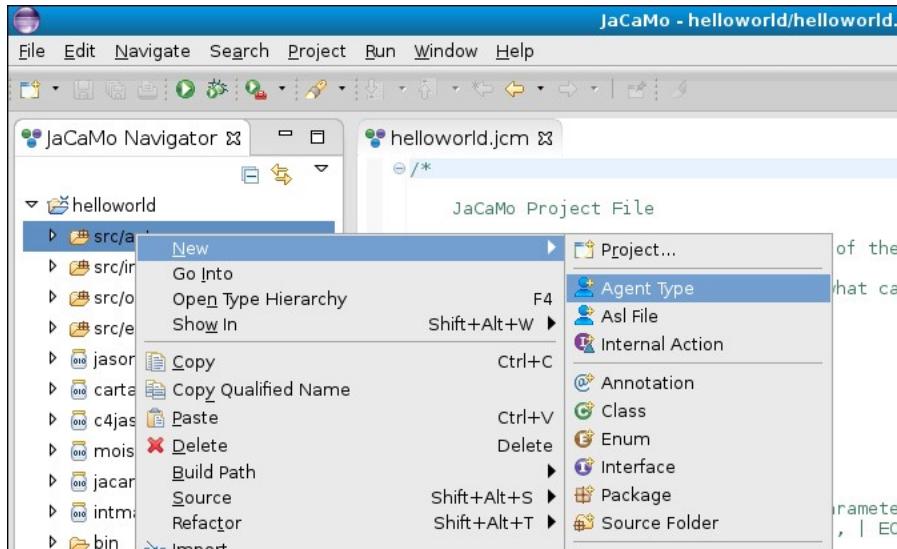
The result will be a "hello world" message in your screen.



How to create an agent type

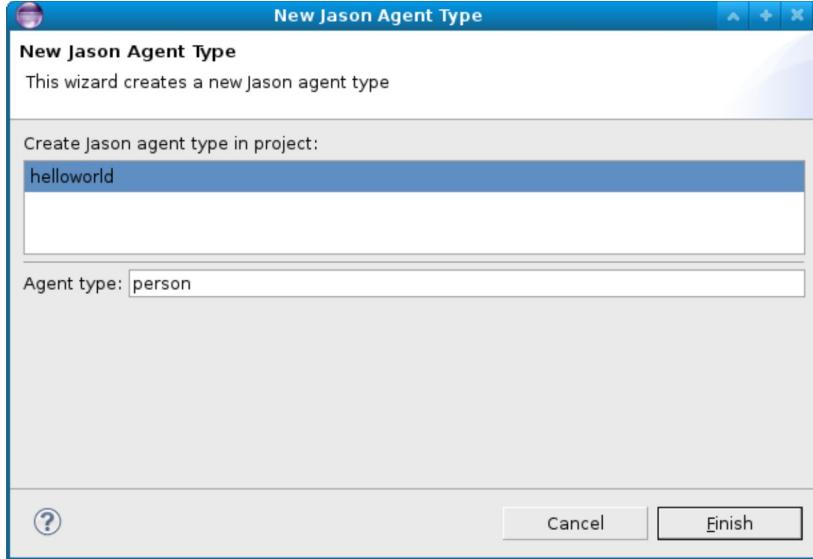
Step 1

Click on the source folder named "src/agt" using the right button and go to the option New > Agent Type.



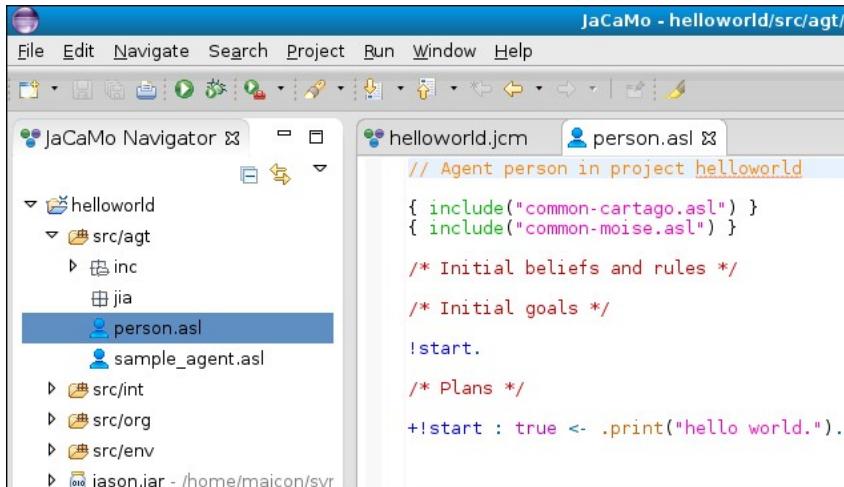
Step 2

Fill out the form. The only required field is the name of the agent type.
After it, press the "Finish" button.



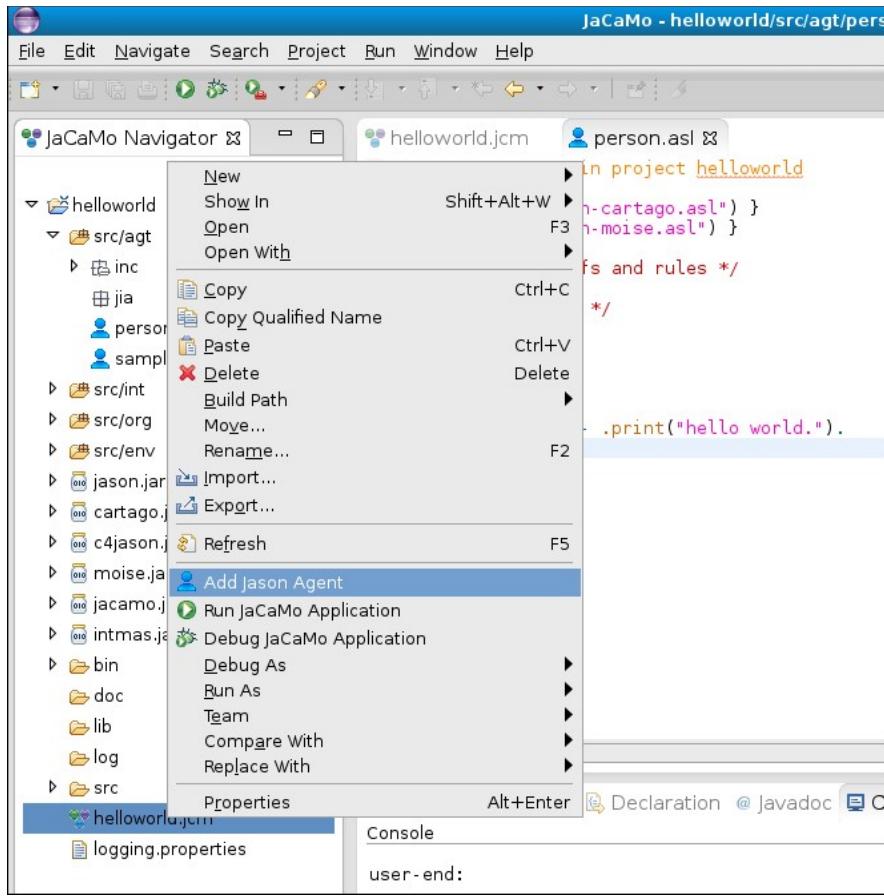
Step 3

The agent type will be created and the following window will appear.



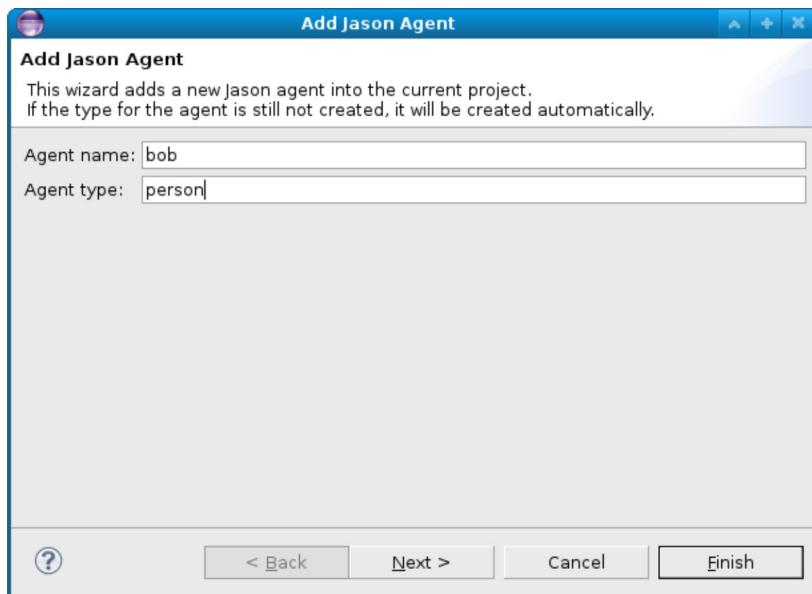
Step 4

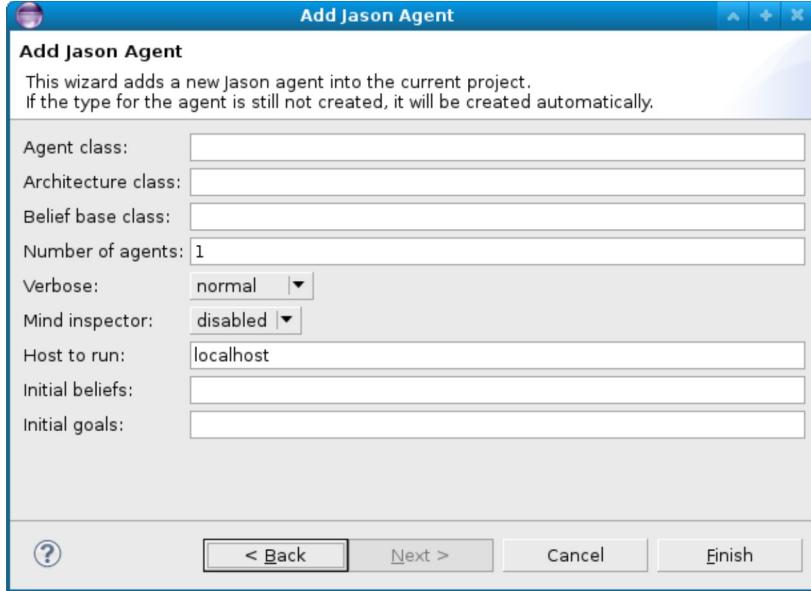
New agents can be created using the just created agent type. Click on the project file using the right button and go to the option Add Jason File.



Step 5

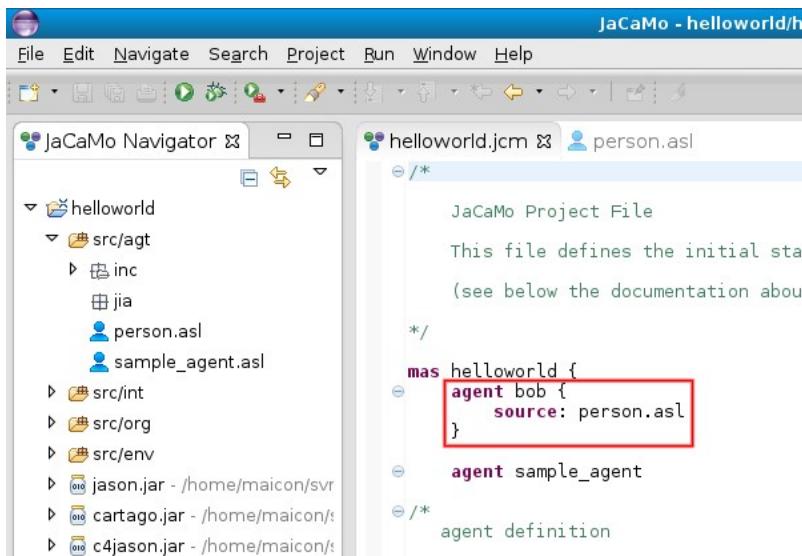
Fill out the form. The only required fields are the agent name and agent type. After it, press the "Finish" button, if no more configuration will be performed, otherwise press the "Next" button for an advanced configuration.





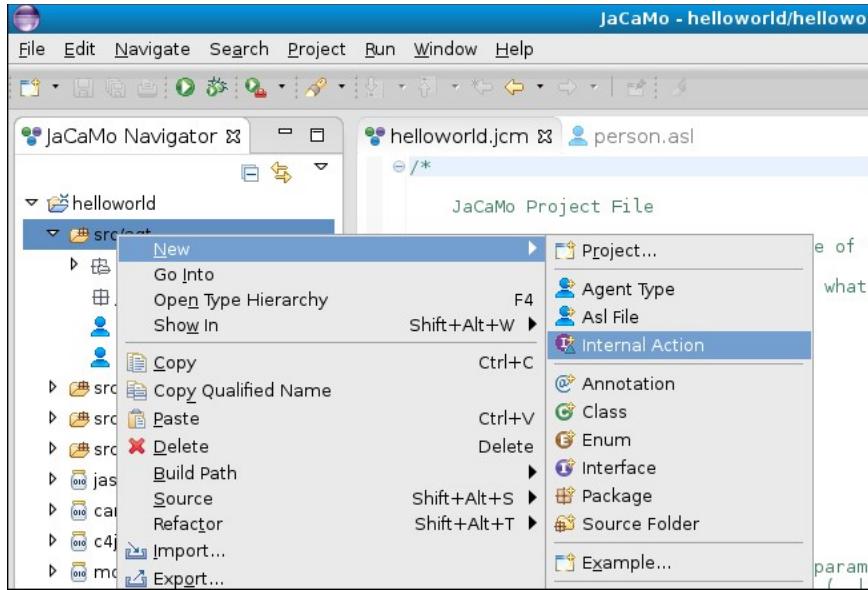
Step 6

The new agent will be automatically added in the project file.



Step 1

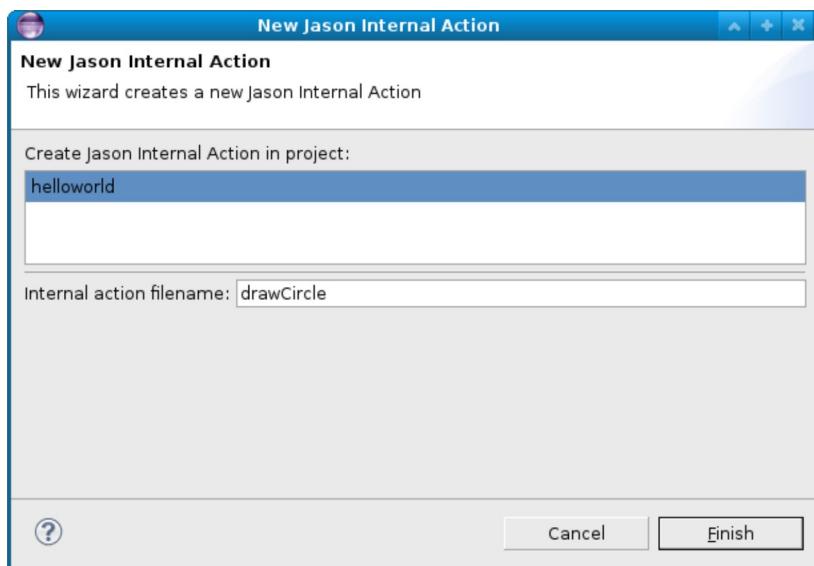
Select the source folder named "src/agt" and go to the option New > Internal Action.



Step 2

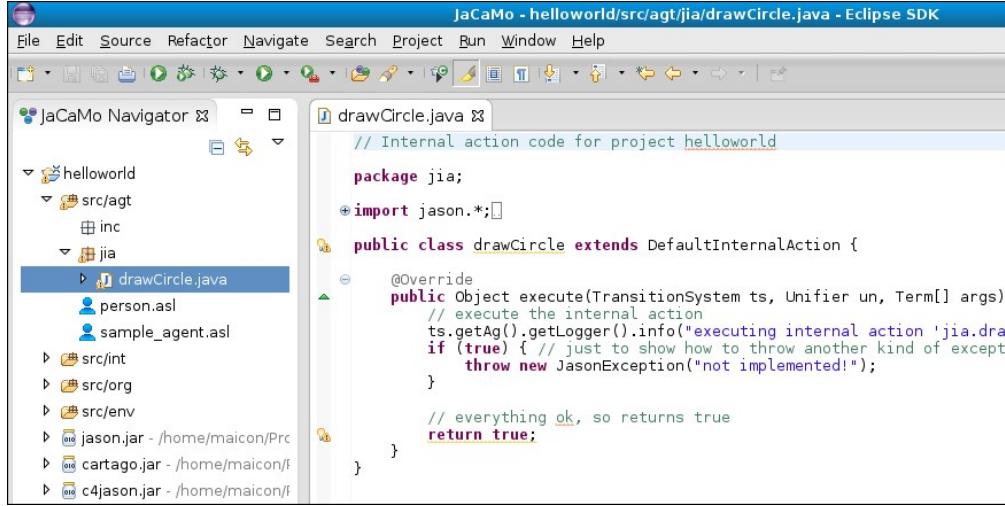
Fill out the form. An internal action is a java class, so the only required field is the name of the class.

Note: we suggest you to give a name using the first letter in lower case.



Step 3

The internal action will be created.



```

JaCaMo - helloworld/src/agt/jia/drawCircle.java - Eclipse SDK

File Edit Source Refactor Navigate Search Project Run Window Help

JaCaMo Navigator
helloworld
  src/agt
    inc
    jia
      drawCircle.java
      person.asl
      sample_agent.asl
  src/int
  src/org
  src/env
  jason.jar - /home/maicon/P...
  cartago.jar - /home/maicon/...
  c4jason.jar - /home/maicon/...

drawCircle.java
// Internal action code for project helloworld

package jia;

import jason.*;□

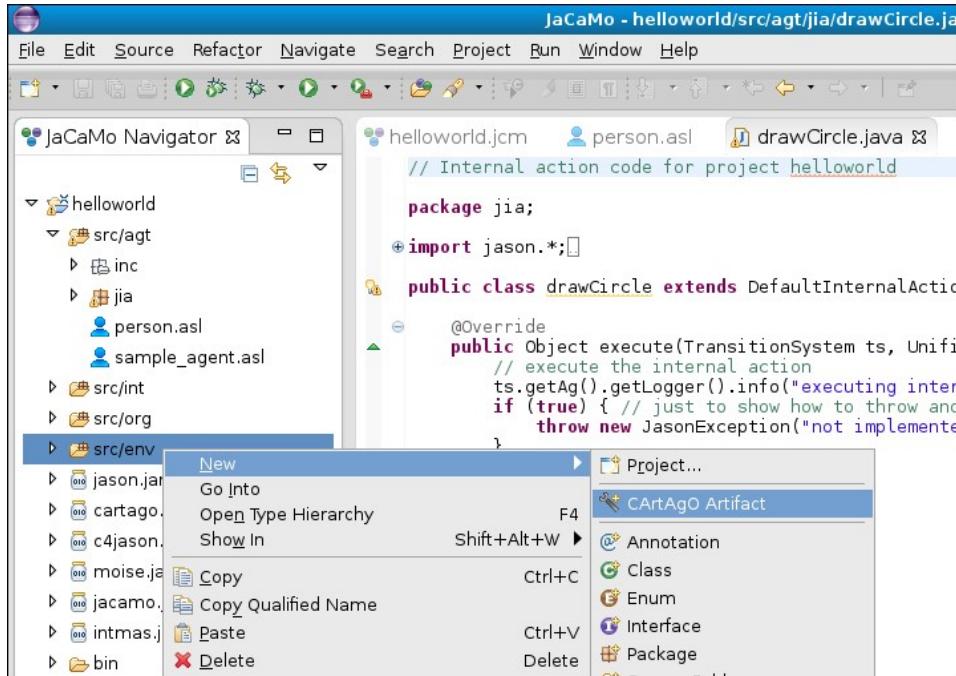
public class drawCircle extends DefaultInternalAction {
    @Override
    public Object execute(TransitionSystem ts, Unifier un, Term[] args)
        // execute the internal action
        ts.getAg().getLogger().info("executing internal action 'jia.draw");
        if (true) { // just to show how to throw another kind of exception
            throw new JasonException("not implemented!");
        }
        // everything ok, so returns true
    }
}

```

How to create an Artifact

Step 1

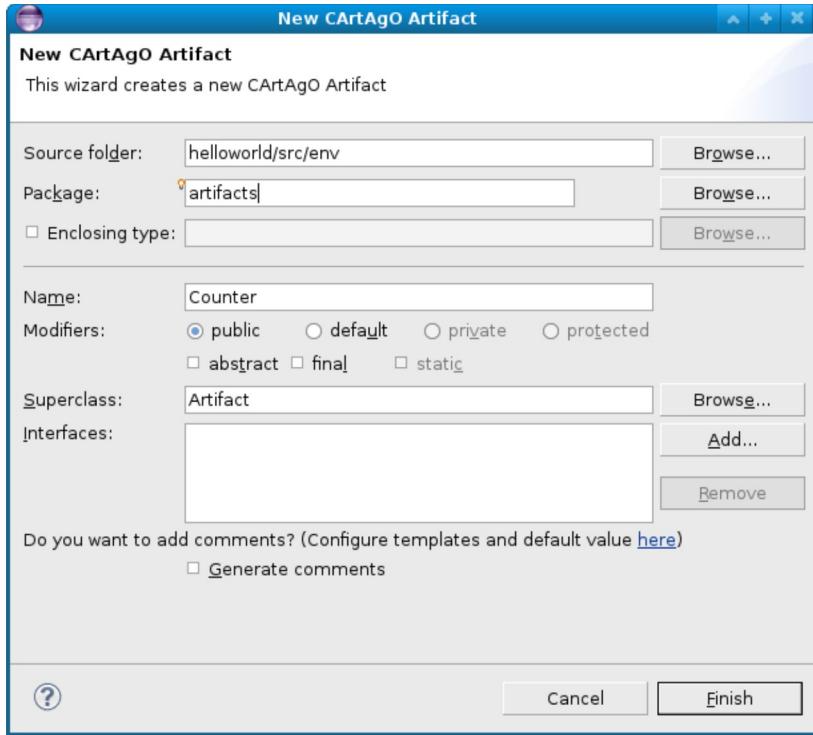
Click on the source folder named "src/env" using the right button and go to the option New > CArtAgO Artifact.



Step 2

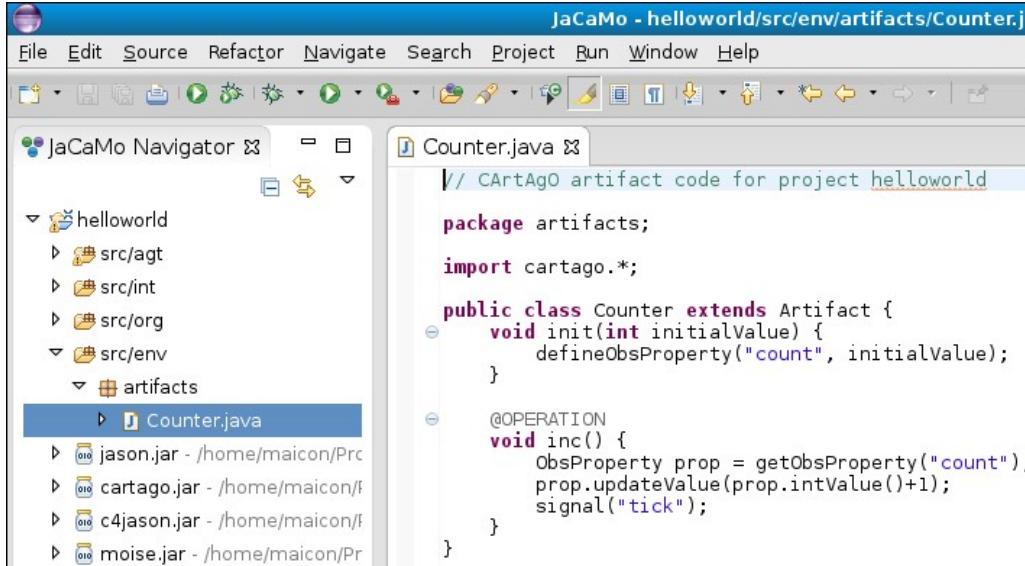
Fill out the form. A CArtAgO artifact is a java class, so the only required field is the name of the class.

Note: in contrast to an internal action, in this case you could use a name with a first letter in upper case, and also we suggest you to name the package.



Step 3

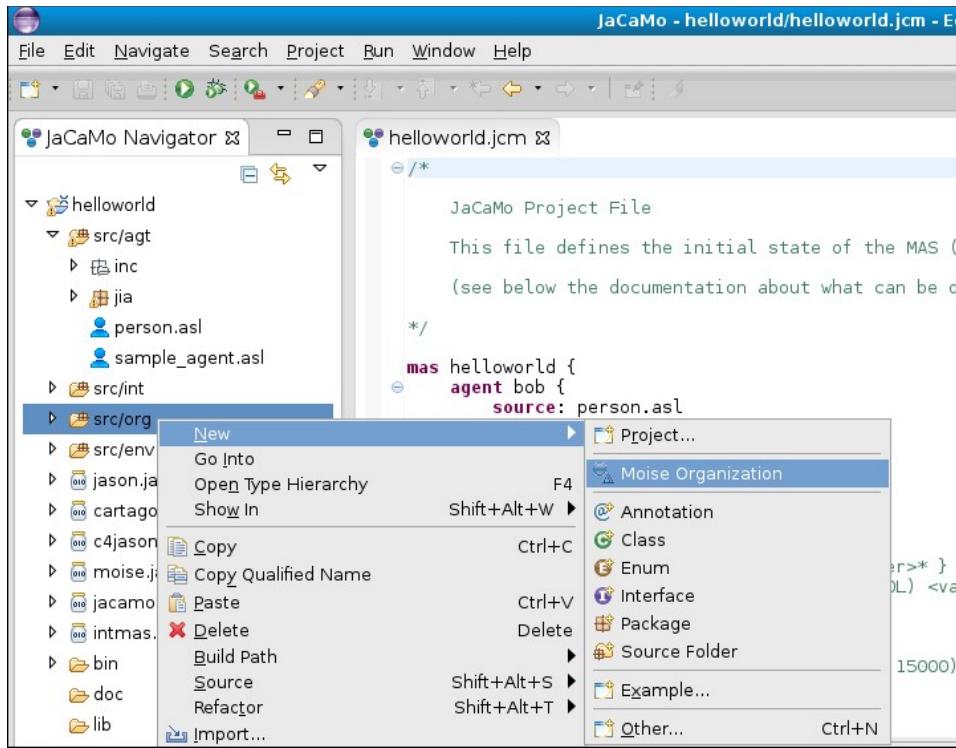
The CArtAgO artifact will be created.



How to create a Moise Organization

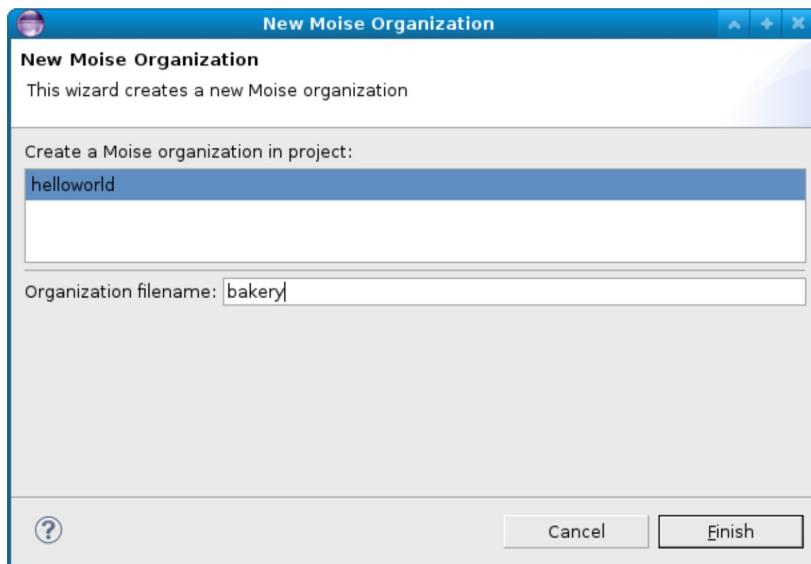
Step 1

Click on the source folder named "src/org" using the right button and go to the option New > Moise Organization.



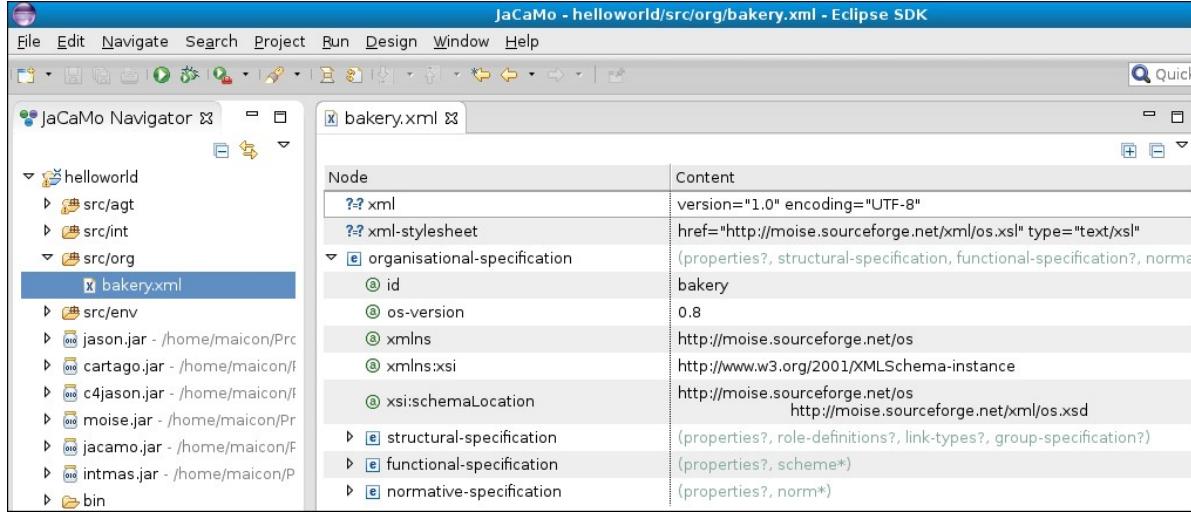
Step 2

Fill out the form. The only required field is the organization filename.
After it, press the "Finish" button.



Step 3

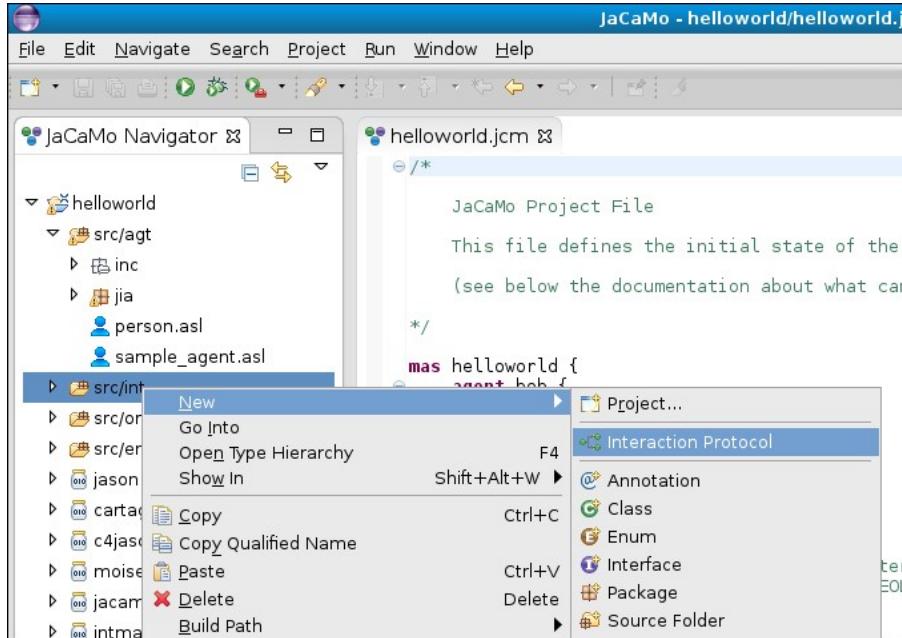
The Moise organization will be created.



How to create an Interaction Protocol

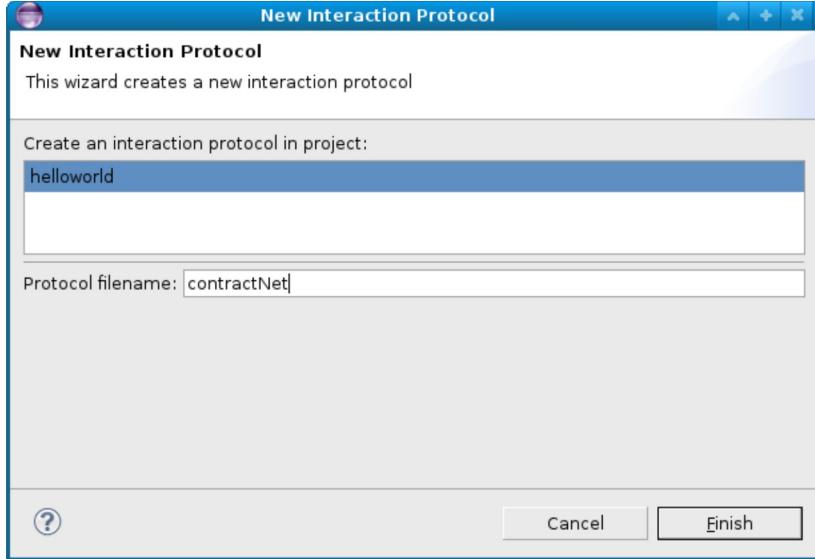
Step 1

Click on the source folder named "src/int" using the right button and go to the option New > Interaction Protocol.



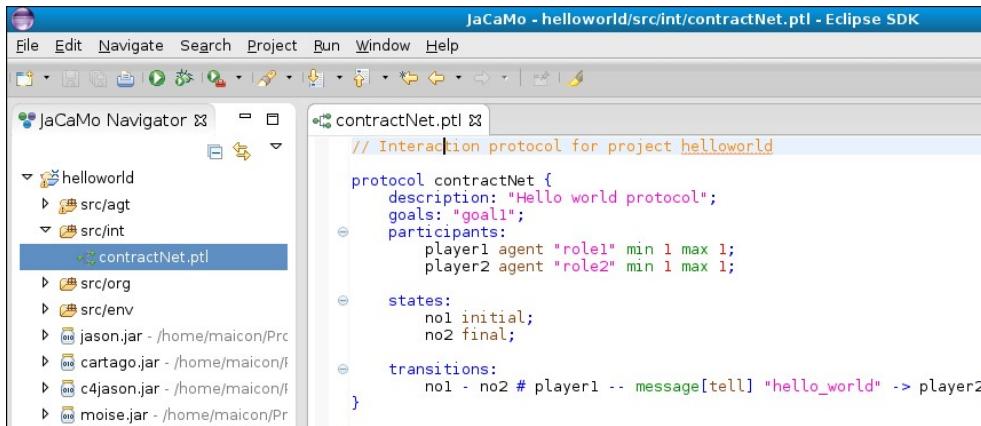
Step 2

Fill out the form. The only required field is the protocol filename.
After it, press the "Finish" button.



Step 3

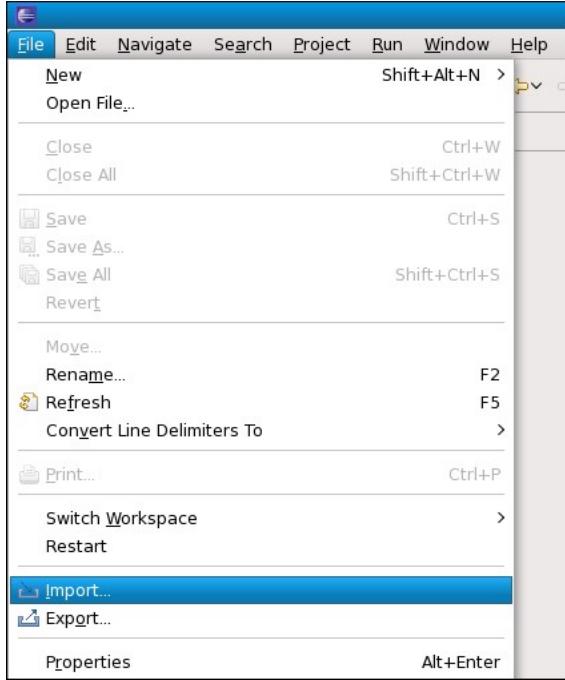
The interaction protocol will be created.



How to import a JaCaMo project

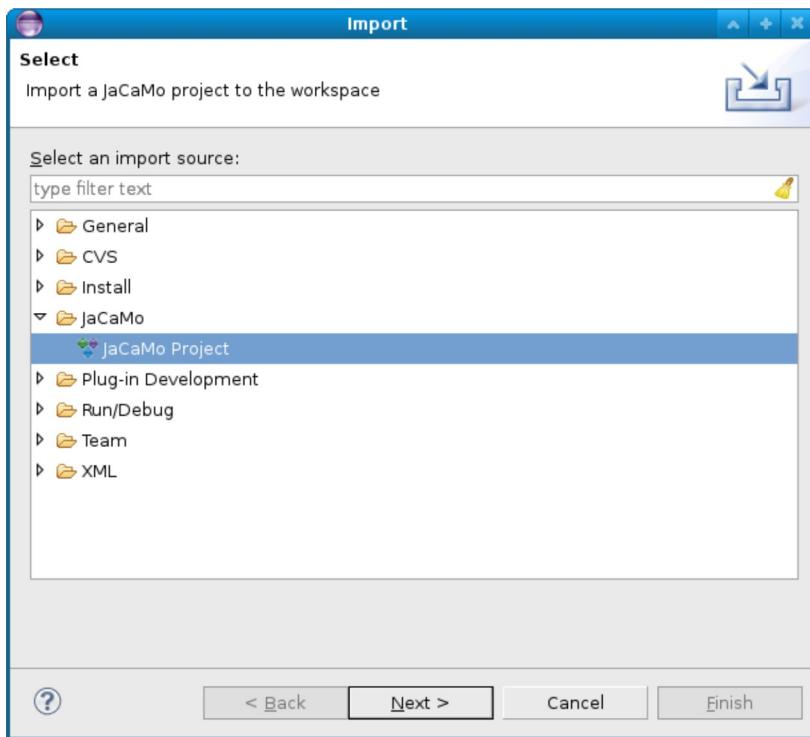
Step 1

Click on the "File" menu and go to the option "Import...".



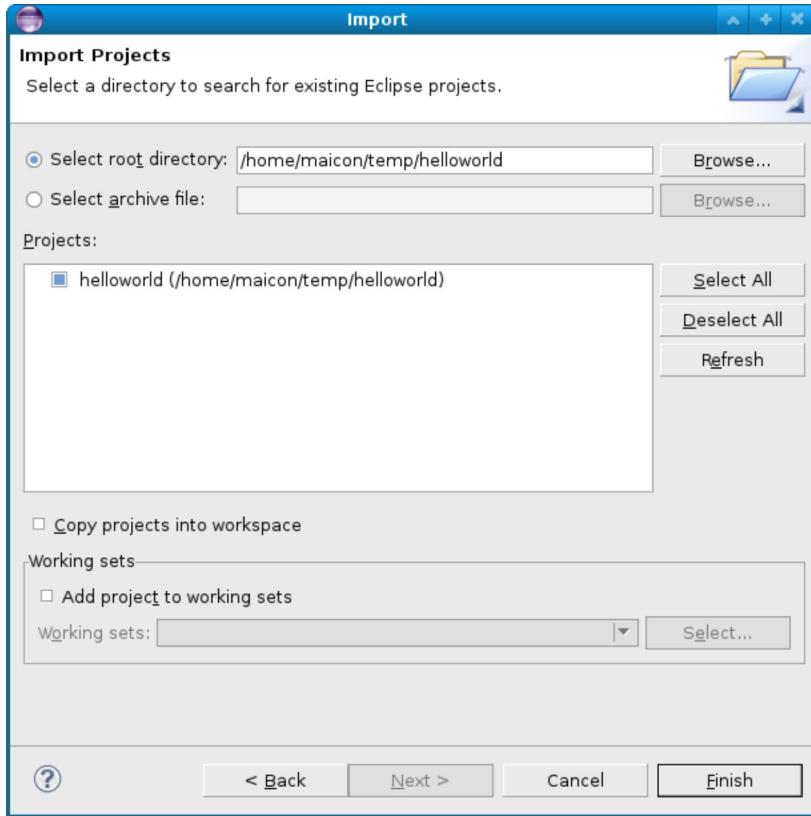
Step 2

Select the option JaCaMo > JaCaMo Project.



Step 3

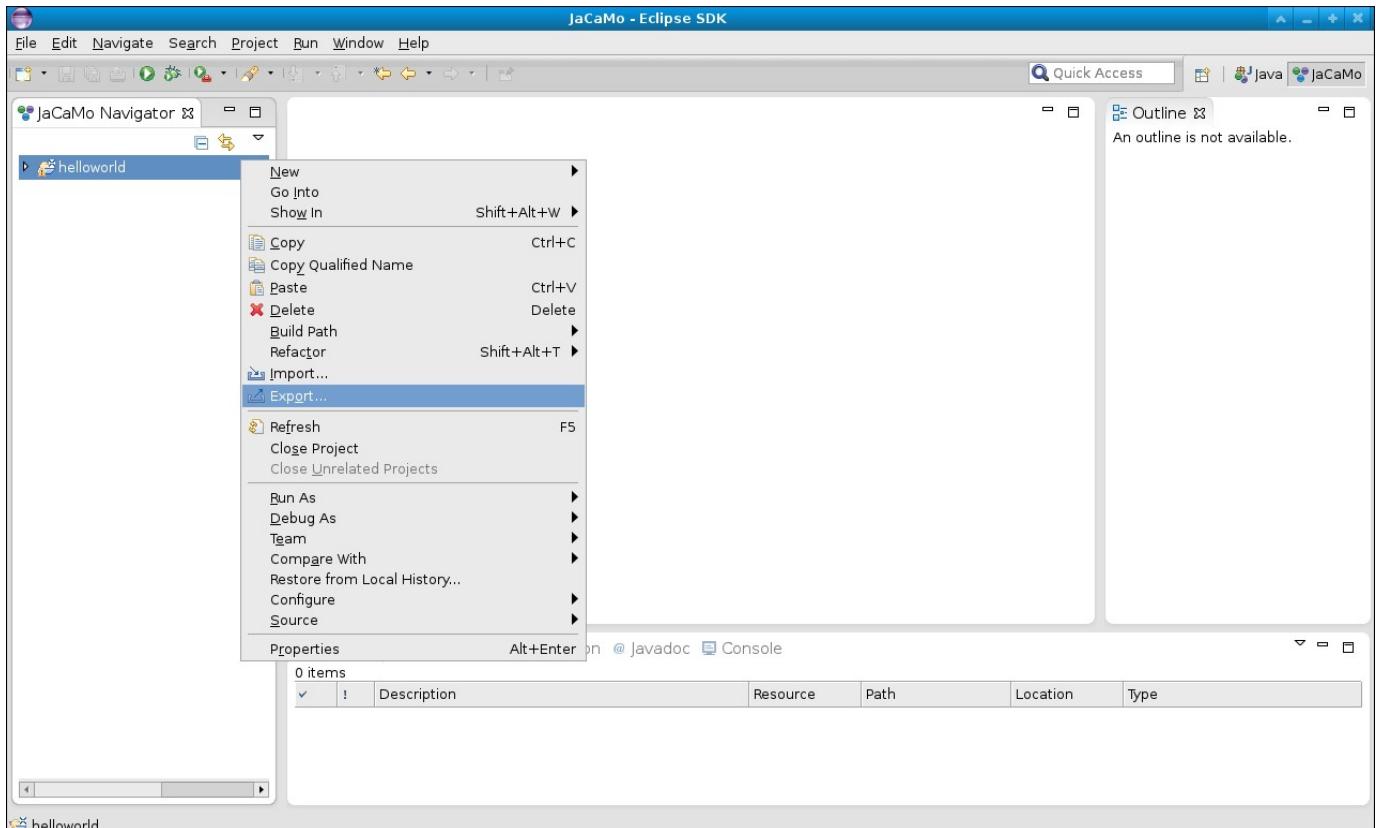
Click on the "Browse" button and choose the directory of the project, tick the project that you wish to import and finally press the "Finish" button.



How to export a JaCaMo project

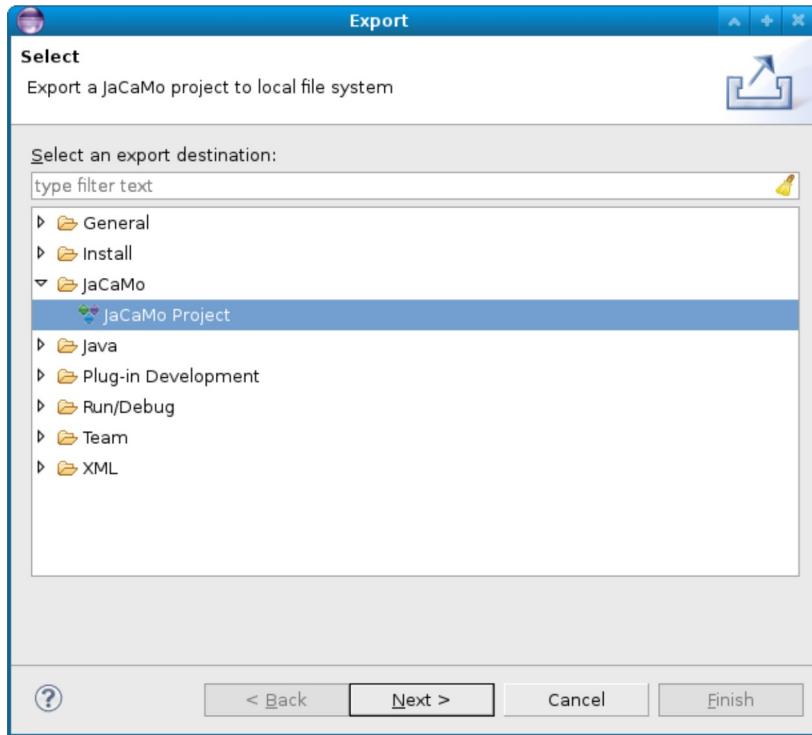
Step 1

Click on the project using the right button and go to the option "Export...".



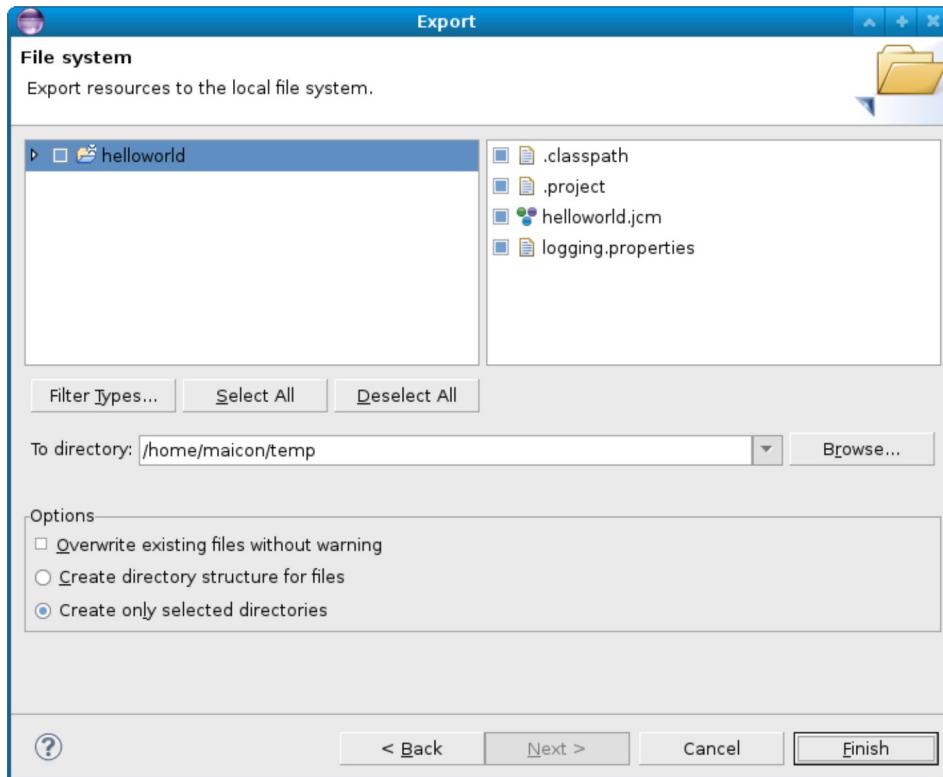
Step 2

Select the option JaCaMo > JaCaMo Project.



Step 3

Click on the "Browse" button and select the directory that you wish to export the project and press the "Finish" button.

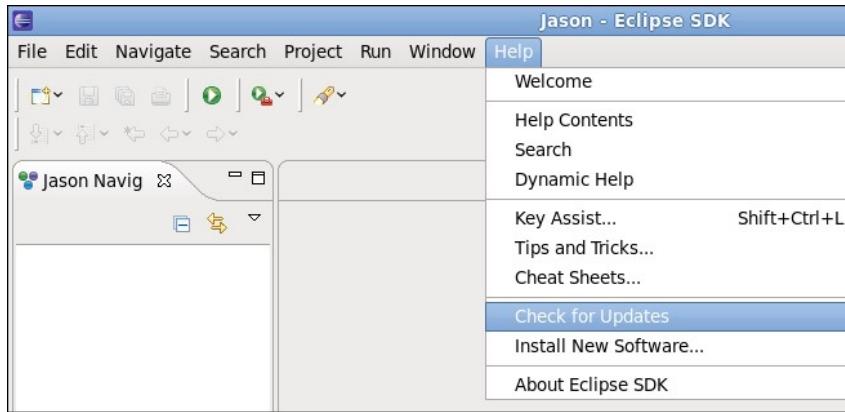


How to update the JaCaMo plugin

You have two ways to update your JaCaMo eclipse plugin.

First way

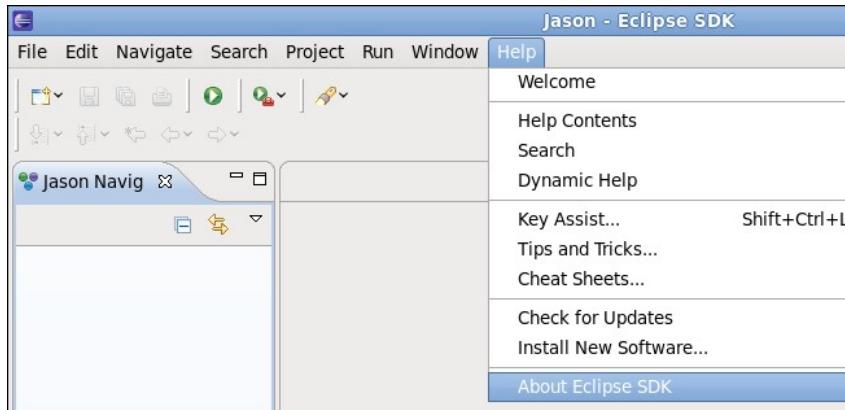
Simply click on the "Help" menu and go to the option "Check for Updates".



Second way

Step 1

Click on the "Help" menu and go to the option "About Eclipse SDK".



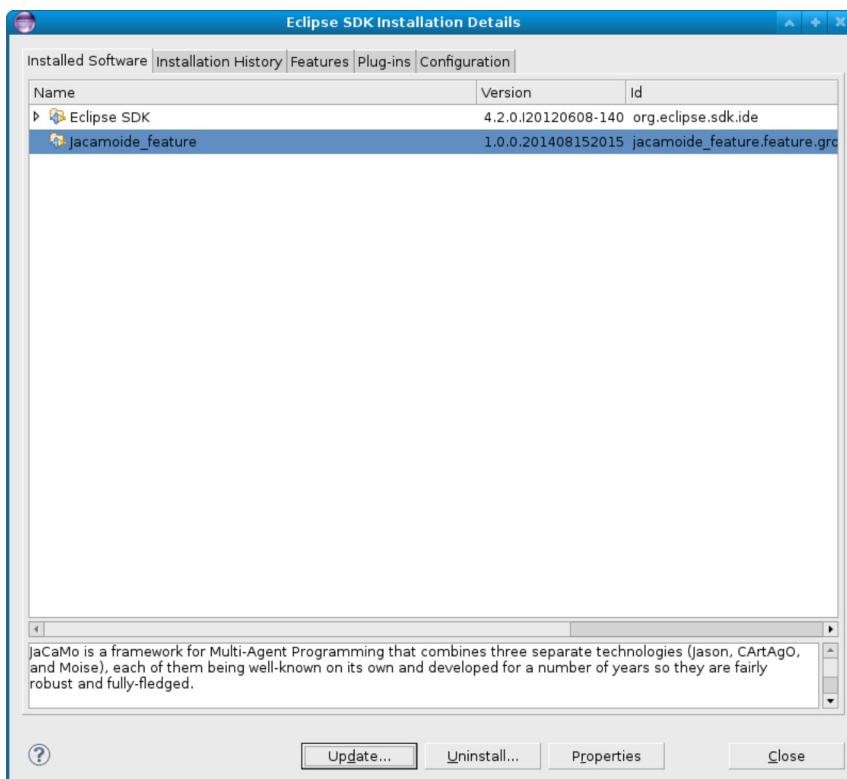
Step 2

Press the "Installation Details" button.



Step 3

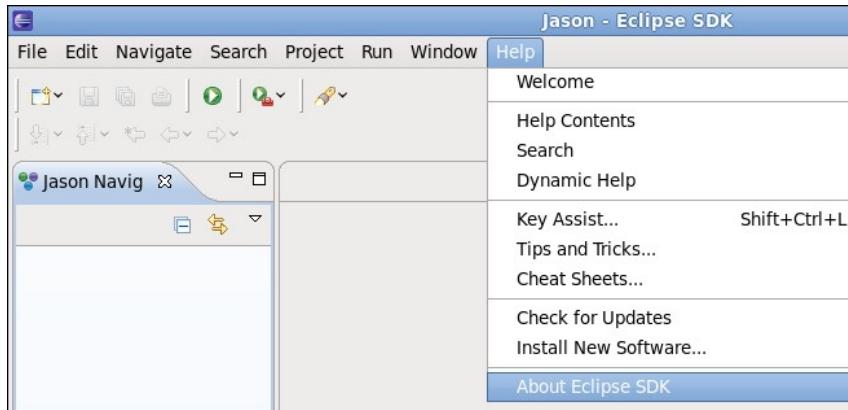
Select the "jacamoide_feature" and click on the "Update..." button.



How to uninstall the JaCaMo plugin

Step 1

Click on the "Help" menu and go to the option "About Eclipse SDK".



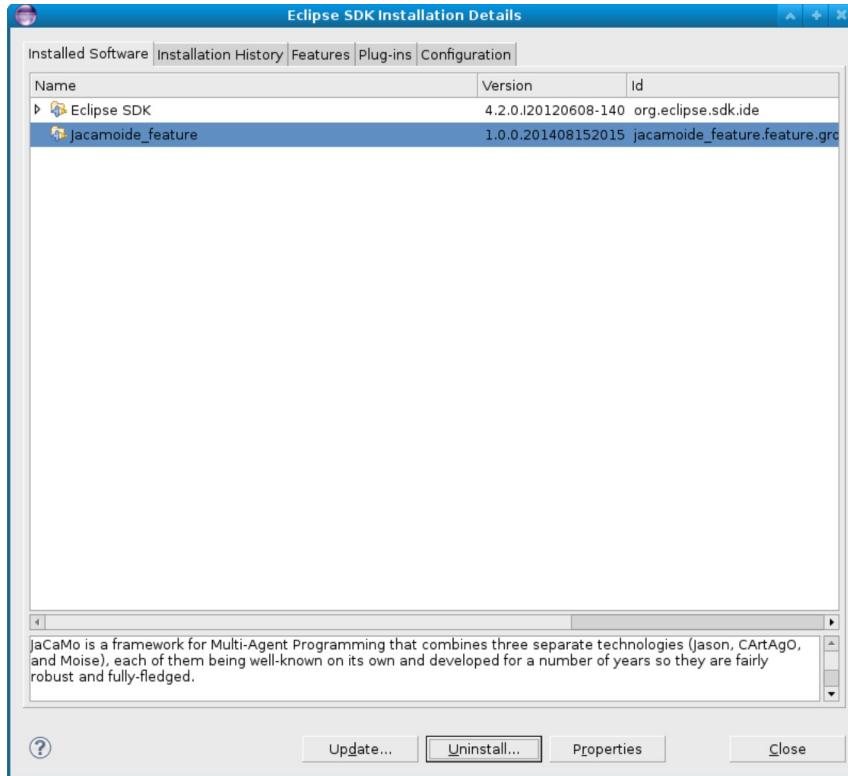
Step 2

Press the "Installation Details" button.



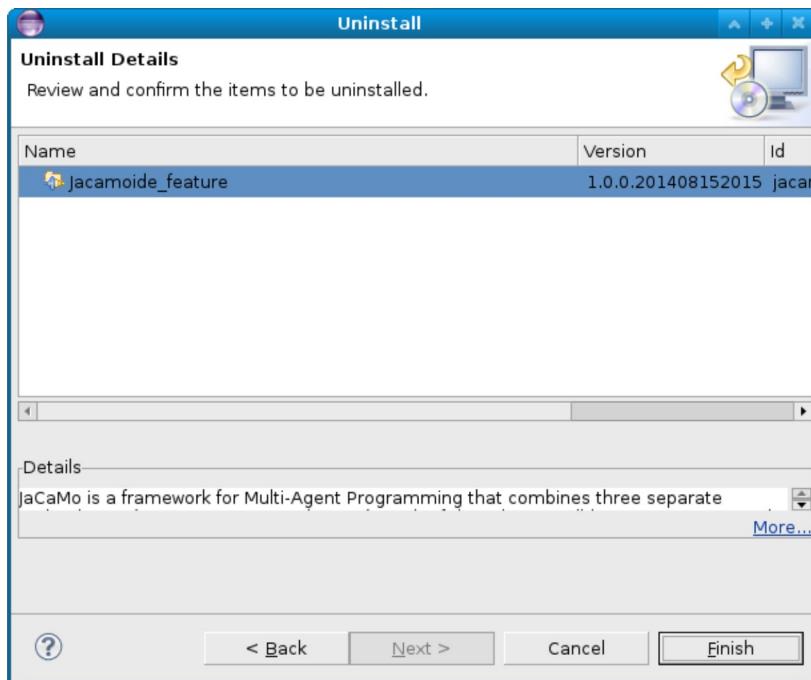
Step 3

Select the "jacamoide_feature" and click on the "Uninstall..." button.



Step 4

Confirm the process of uninstallation pressing the "Finish" button.



Step 5

In order to complete the uninstallation press the "Restart Now" button.



Thanks for your interest on JaCaMo. You will find more information at <http://jacamo.sf.net>.