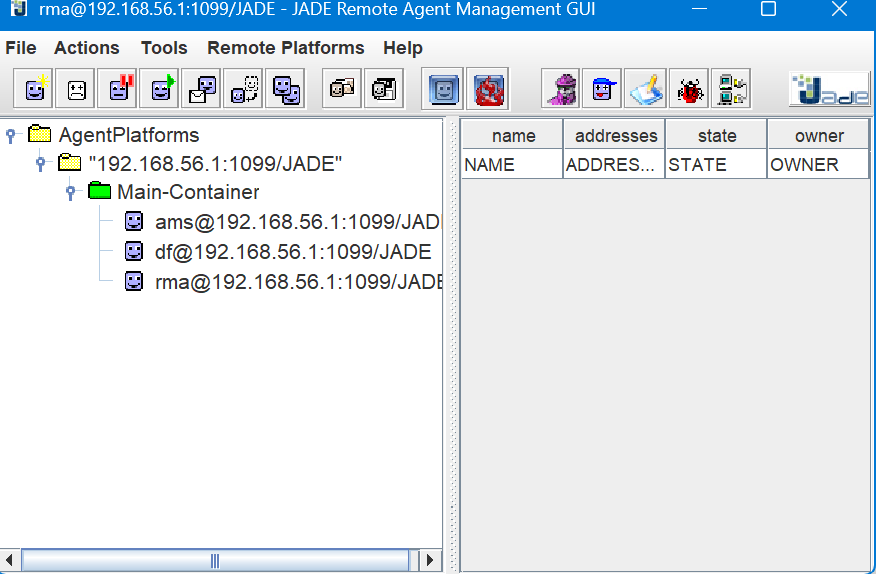
**JADE day01**

First install the software and make the folder structure   
then according to the tutorial guide we have to go to the jade directory and start the JADE GUI

Code we have to run to start JADE GUI  
java -cp lib\jade.jar jade.Boot -gui

Some time we have to mention where is our class files are locatingjava -cp lib\jade.jar;classes jade.Boot -gui

Then when its run we get the GUI like this

****



In this GUI you can see we get default 3 agents

ams , df , rma these are default agent coming with our system

------------------------------------------------------------------------------------------------------------------**what are these default agents**

1. AMS (Agent Management System)

This is supervise agent for whole system ,that manage all agent identities and address like a phone book

1. DF (Directory Facilitator)

This is called as yellow agents in the system

They are providing their services , hi I am agent 1 , I am selling vanilla ice-cream

1. RMA (Remote Monitoring Agent)

This is Your GUI, that provide interface to manage the agents

------------------------------------------------------------------------------------------------------------------**create a new agent**

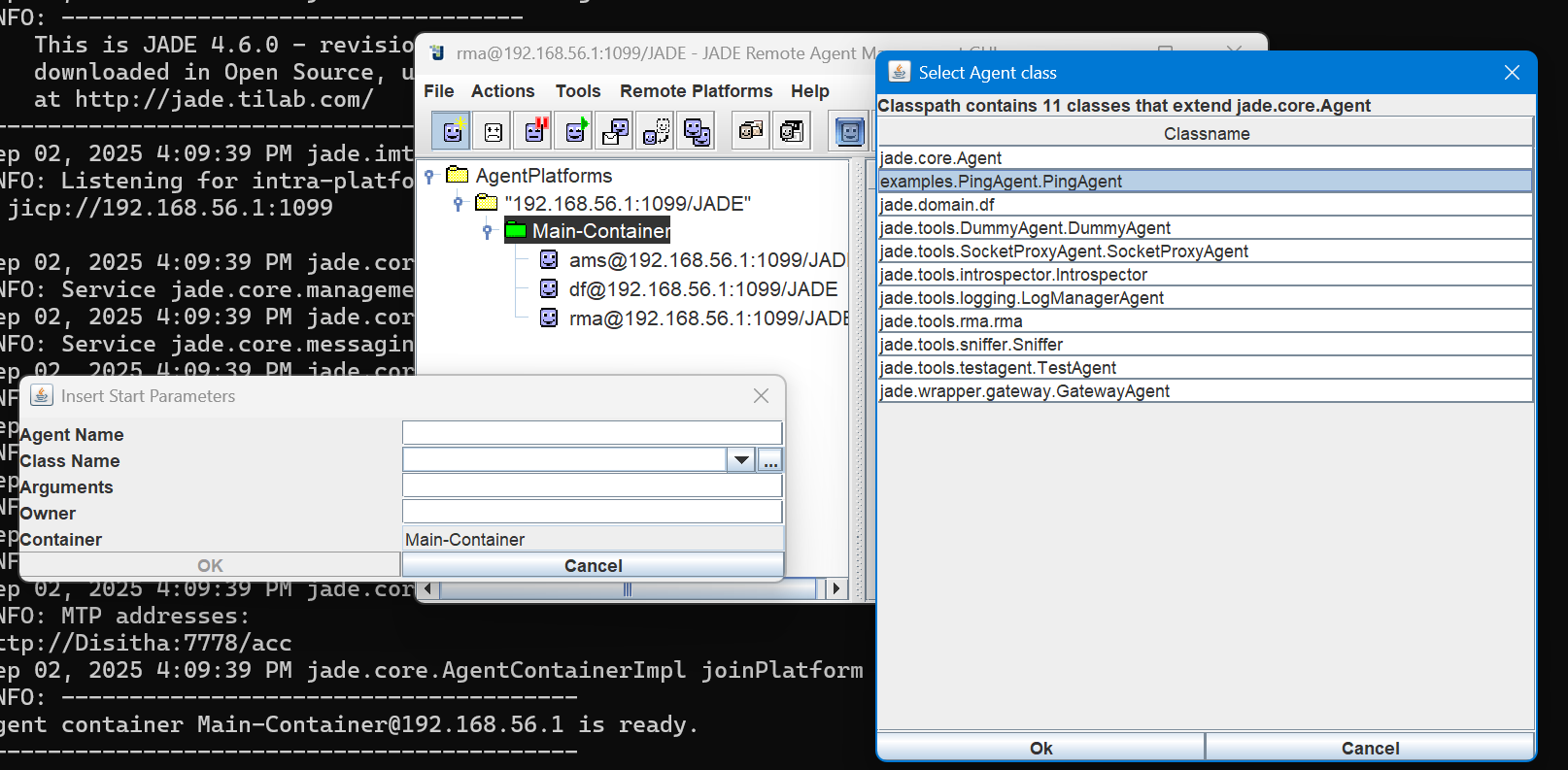
If you want to create a new agent you have to compile the agent first , after that you can create the agent   
so in first how to compile  
  
You have to stay in your same Jade directry and type  
javac -classpath lib\jade.jar -d classes src\examples\PingAgent\\*.java

This PingAgent is example agent class file that come with jade platform to test some message passing, this Ping-Agent class file is defined how should our agent must behave

Using this class we are creating agents , so these agents have behave of Pingagent class file , if you want any changes of the your agent behave , you have to change your class file code

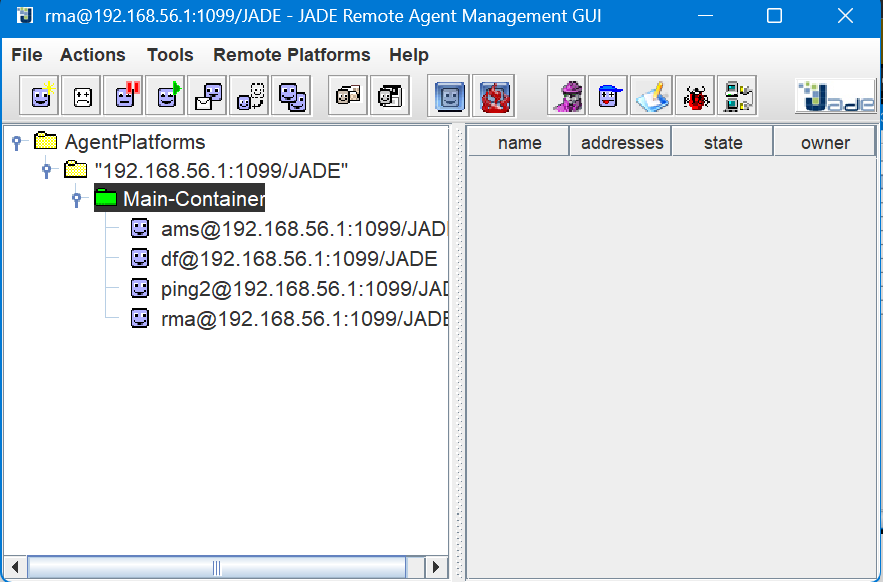
So before that we are going with PingAgent example to familiar with this framework

After compiling we can load the agent in two different way , by using CMD or JADE GUI, in cmd we have to type<java -cp lib/jade.jar;classes jade.Boot -gui -agents ping1:examples.PingAgent.PingAgent> But is good way to start with GUI,



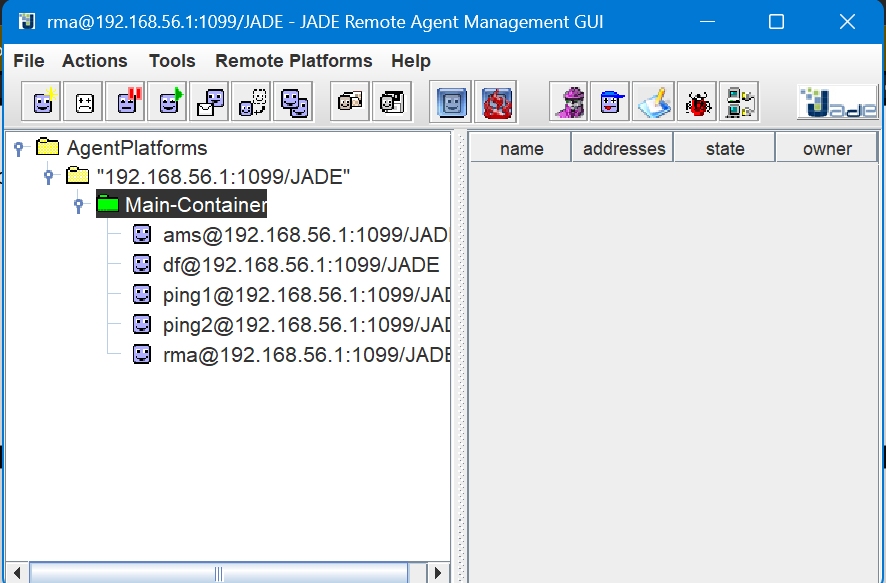


Give the agent name as ping2 and , Now you ping2, agent is created





Now create two agents



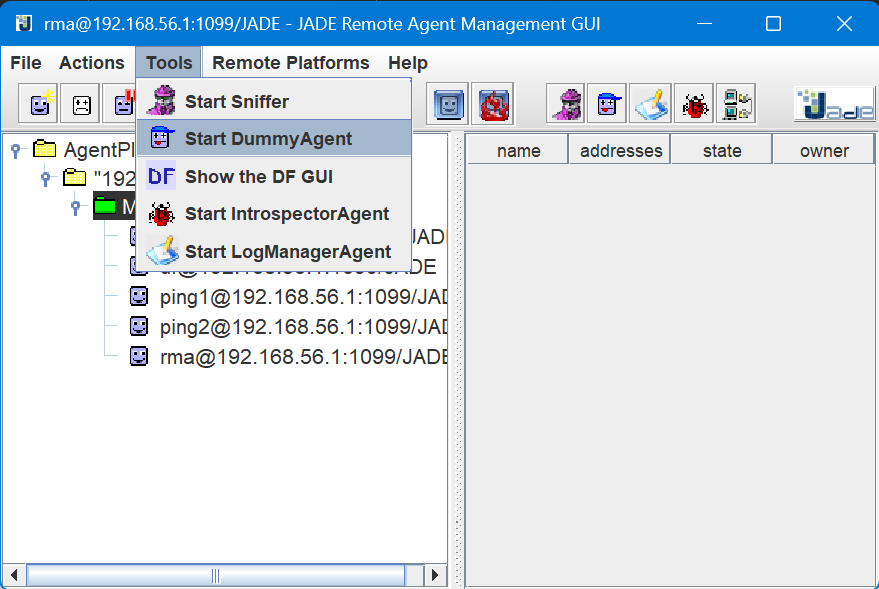


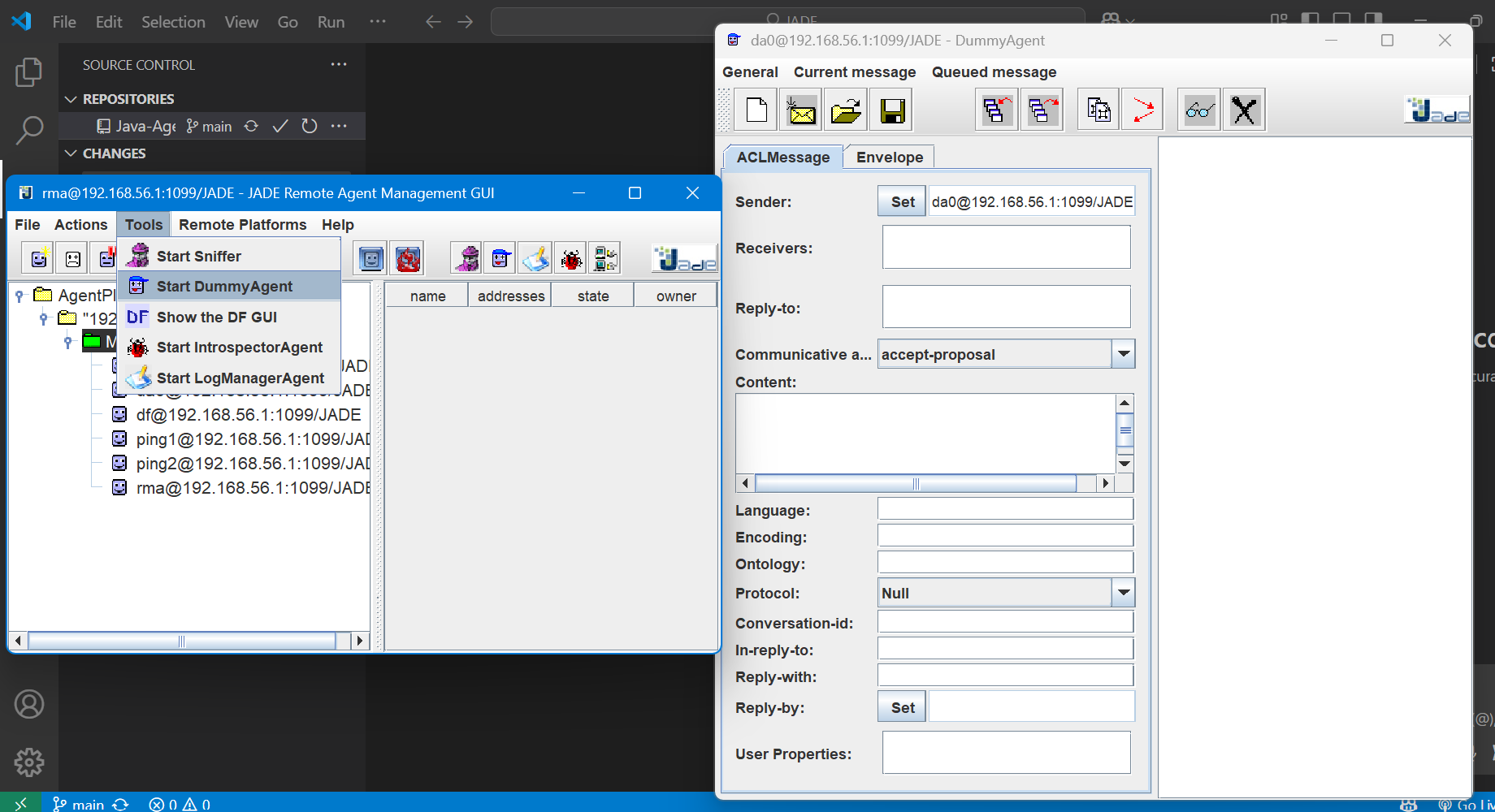
**DummyAgent**

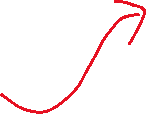
Actually why we create dummyagent for pass message,

Actually dummyagent is tool given by JADE , to test other agents, by using dummyagent we can test other agents communication is doing well , actually we don’t need dummy agent , previously we created ping1,and ping2 can communicate each other directly , but it require coding, so in first time we use dummyagent , to test our agent

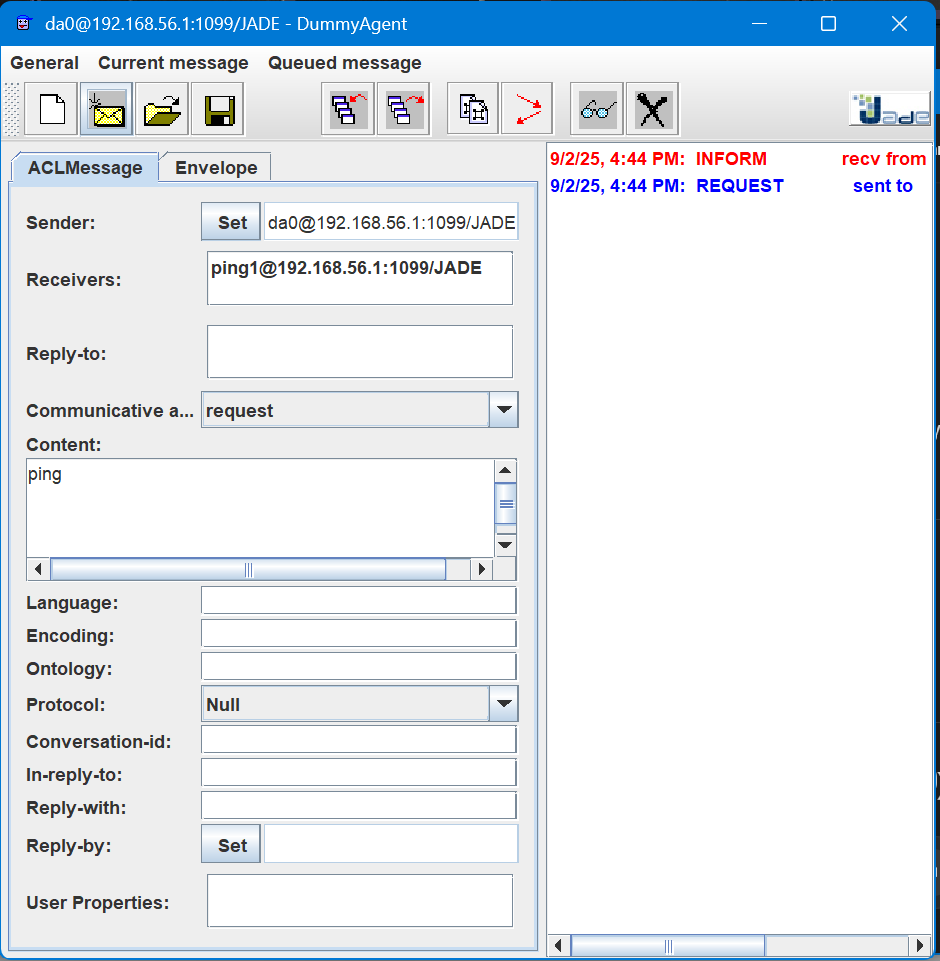
Start dummyagent







In here if you want to send a message to ping1(our agent 1) we simply fill this form



Make sure we select this one as a “request”

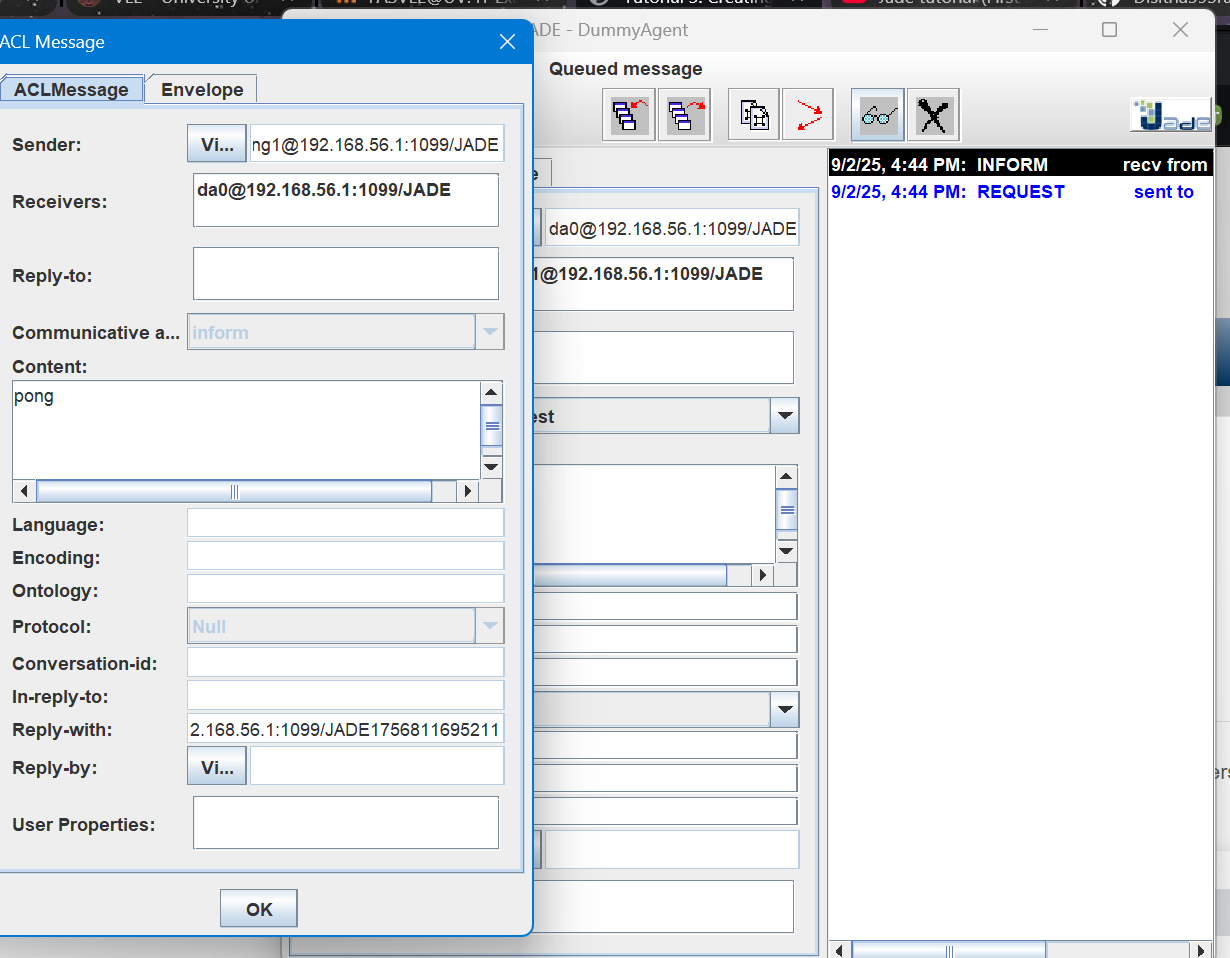


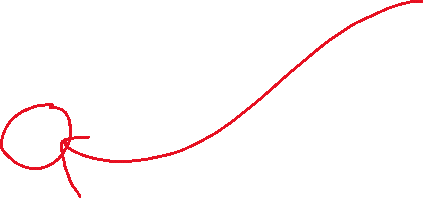
When we are doing we add the ping1 as a receiver in that movement we put a click on the name field that mean is

“I’m only giving the **local name** of the agent.(like ping1)  
JADE can automatically append the platform name.”

Bcz JADE using unique identifiers for agents, they are adding platform name, local name, host name make it unique so , choosing the tick mark , we give opportunity for JADE to make it unique using our agent name

------------------------------------------------------------------------------------------------------------------**Now Request & Response**





Now we can click the message and spec I can see the reply   
This is first message passing

------------------------------------------------------------------------------------------------------------------**Creating Multiple Containers**

In Jade Container is java processor that host agents, main container is host of the platform , we can create so many subcontainers, but the thing is these subcontainers can’t live alone , They must have connect with main container ,

As a example of real world example , think Google headoffice(main container), it has sub containers like London branch, New York branch like wise

java -cp lib\jade.jar;classes jade.Boot -container

this -container start peripheral container instead of main container, but this sub container cant live in alone so JADE assume that container

 Host = localhost (your current computer)

 Port = 1099 (default RMI port)

Now after creating the peripheral container we have to connect it with main container , we can do it using this command

java -cp lib\jade.jar;classes jade.Boot -container -host jade.tilab.com -port 1099

How this code actually works

 java → starts the Java Virtual Machine.

 -cp lib\jade.jar;classes → **classpath**, tells Java where to find JADE classes (jade.jar) and your compiled agent classes (classes).

 jade.Boot → the JADE bootstrap class that **starts a container**.

 -container → tells JADE: **this is a peripheral container**, not the Main Container.

 -host frodo → specifies the host where the **Main Container** is running.

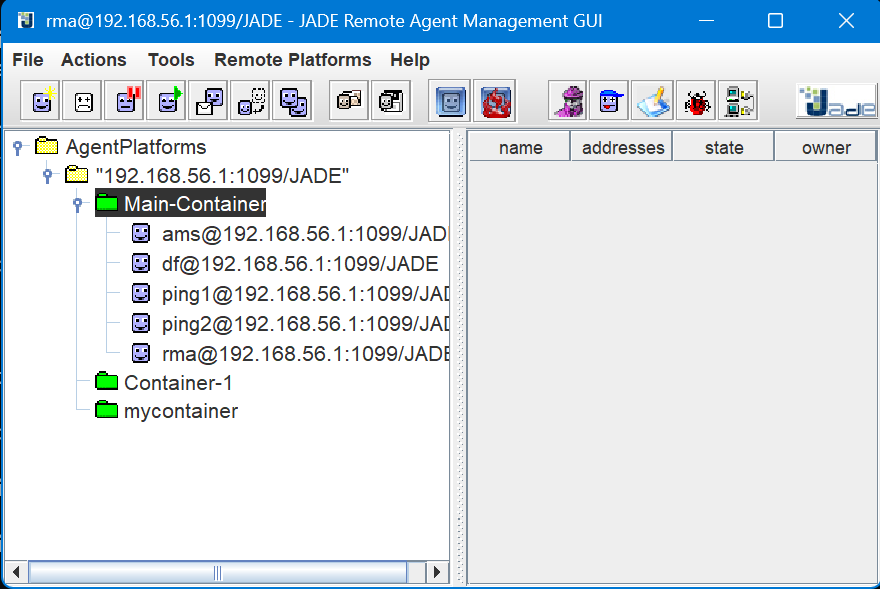
 -port 1099 → specifies the port of the **Main Container** (default RMI port).

In the thing is if we don’t specify our container name JADE automatically generate container name example(Container-1, Container-2, Container-3, …)

But also we can give any name what we want

java -cp lib\jade.jar;classes jade.Boot -container -container-name mycontainer

Host and port can be skipped if the Main Container is on **localhost** and **default port 1099**.

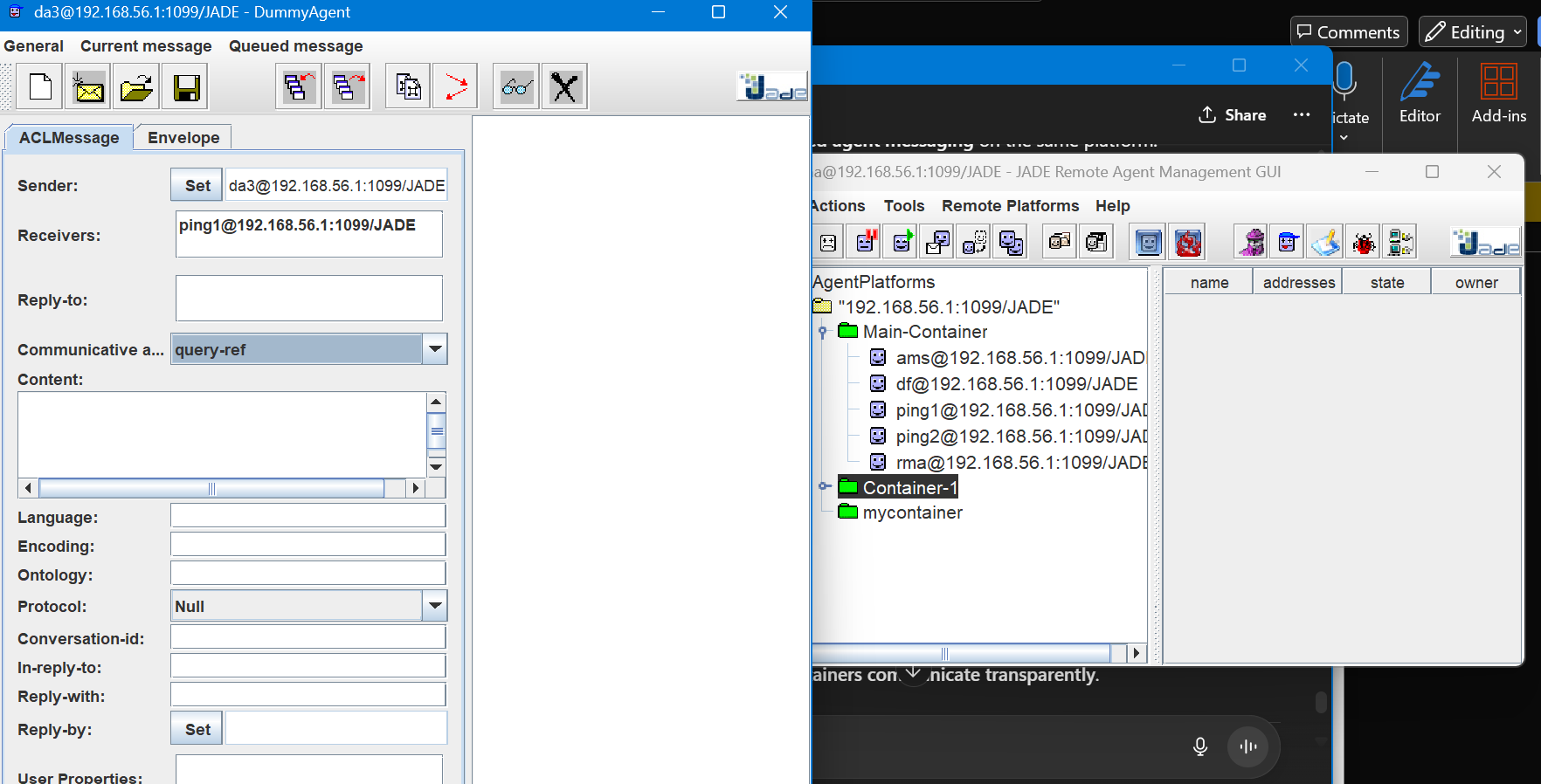




------------------------------------------------------------------------------------------------------------------ **cross-container messaging**

New we are doing check the agents can communicate in different container

So in first click on container-1 and start dummy agent ,





This is give replay as same as “pong”

------------------------------------------------------------------------------------------------------------------**Remote Container**

Until now we create all the containers(main or peripheral) in same computer that mean under the same host

So remote containers are running on under the different host(computers)