This is a brief and initial guide to using Boris to link Java and Lisp agents.

downloads

A number of files are required, these can all be downloaded from www.agent-domain.org:

1. matcher(1.6).lsp
   a general purpose Lisp pattern matcher

2. utils.lsp
   a collection of general purpose Lisp utilities used by (but not bundled with) Boris

3. boris-lisp-(v0.3b).cl
   the Lisp bundle for Boris

4. calculator.cl
   some sample code to build a simple MAS

5. boris.jar
   the Java bundle for Boris including router, IDE and Java API

loading Lisp

Loading on Lisp side involves a few steps, this process can be automated using startup files. NB: the <load> statements imply that you should do the loads manually via the truck icon in the allegro IDE.

```
;set up package structure
(in-package :user)
(use-package :mp)
(use-package :socket)

<load> matcher(1.6).lsp
(use-package :matcher)

<load> utils.lsp
<load> boris-lisp-(v0.3b).cl
```

loading Java

To load the Java side just double-click Boris.jar, this opens an IDE and starts a router. You will be asked to name the router (choose anything eg: "R1").

building a small MAS and connecting Lisp & Java

in the Boris Java IDE...

1. click advertise
2. create a new portal (menu>Portals>Create New)
3. create a couple of debug agents called (eg) "sam" & "sue" (menu>agents>run debug agent)
4. check that sam & sue can exchange messages

in Lisp

5. open calculator.cl in the Lisp editor
6. mark & compile the various parts (do this section by section & take time to understand the definitions)

running

7. use one of the debug agents to send a message to calculator saying (eg) 5 plus 7

a sample system

8. write a Lisp agent to broadcast any messages it receives to all other agents it has ever received a message from
9. test this with Java debug agents
10. write a new Java agent (in its own frame) to act as a chat room client, using the Lisp broadcast agent as a chat room server, use the IDE to launch this agent (NB: to avoid everything closing down when you close a chat room client ensure that you do not allow system exit to run if your frame is killed