JASMINe User's Guide

JASMINe User's Guide

Table of Contents

Chapter 1. Introduction

1.1. Goal

The Objectweb JASMINe project aims to develop an administration tool dedicated to Java EE (Apache, JOnAS, EasyBeans, ...), MOM (JORAM, ...) or SOA distributed applications (Orchestra, Bonita, Petals, ...) in order to facilitate the job of the system administrator. It relies on advanced management features and on autonomous behaviour capabilities to reduce the management costs of such architectures.

1.2. Benefits

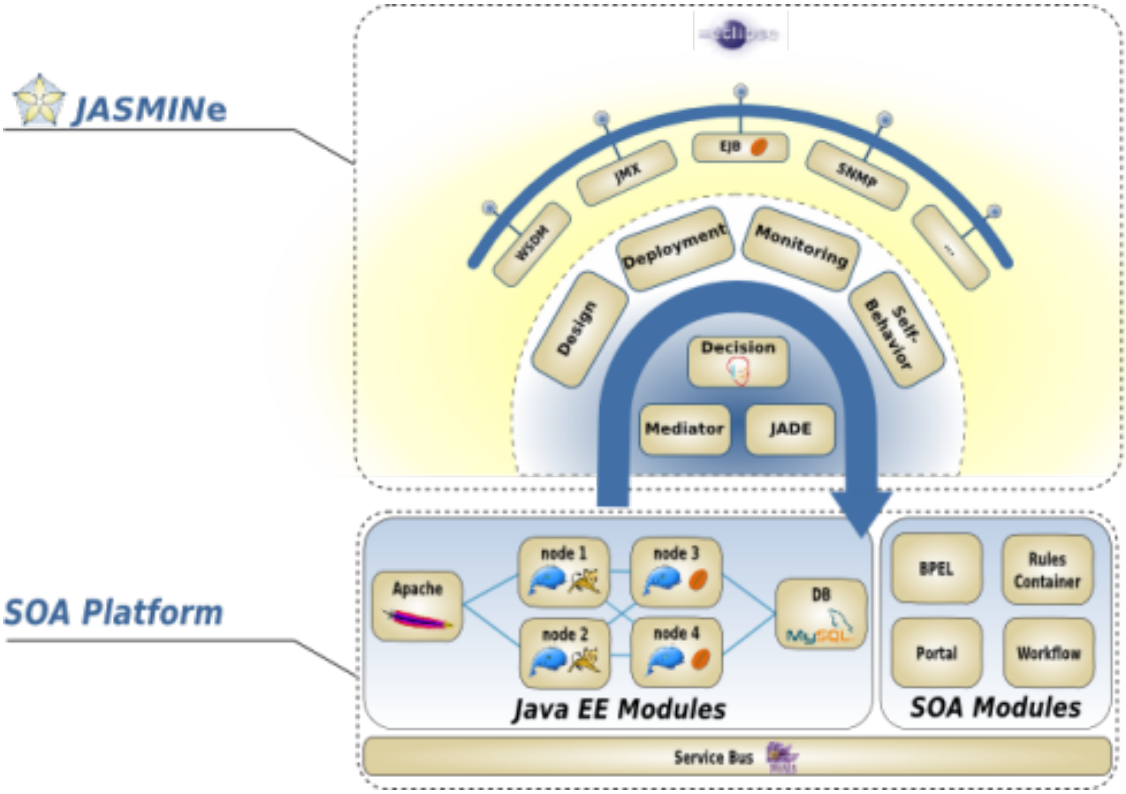
JASMINe main benefits are:

- To help the administrator during the cluster design phase.
- To easily deploy the designed architecture onto the physical machines.
- To reduce the risk of human error in configuration and management operations.
- To improve the response time to eliminate malfunctions occuring in the system.
- To improve the global availability of applications (by minimizing service interruption periods).
- To optimize global performance.

1.3. JADE

JASMINe is based on the JADE project which provide a framework to build autonomic systems.

1.4. Schema



Chapter 2. JASMINe JADE Package

2.1. Pre-requisites

- JDK 5.0
- Ant 1.6.5
- The jade bundle repository should be accessible from all the nodes (via NSF, http, ...). The repository location have to be specified in the jadeboot and jadenode folders in the configuration file `<jade-folder>/conf/config.properties`

```
obr.repository.url=file:<repository-location>/repository.xml
```

2.2. JADE control through JMX

This package acts as a communication bridge between jadeboot and the JOnAS. Executing ant inside the folder where `jade-jmx.zip` was unzipped will start the package:

```
$ ant Buildfile: build.xml

jade-jmx:
[java] service:jmx:rmi:///jndi/rmi://127.0.1.1:9098/server
[java] RMI registry ready.
```

The RMI registry is started on the port 9098 by default.

This package can be executed in any node of the cluster. Its location is a decision of the cluster administrator. It's not mandatory to run it on a specific node.

Note

The package's location have to be specified in the UI config files, otherwise the UI will ask for it.

2.3. JADE Boot

The next step is to run the jade boot.

The JadeBoot can be configured through 2 properties files :

- `<jade-folder>/conf/config.properties`

```
jadeboot.registry.host=localhost
jadeboot.registry.port=1238

jadeboot.discovery.host=localhost
jadeboot.discovery.port=9998

jadeboot.urls.deployable.file=file:./examples/j2ee/;file:./examples/j2ee/
org.ow2.jasmine.jade.resource.j2ee-2.0.0.jar

jadeboot.jndi.port=1239
```

- `<jade-boot-folder>/conf/system.properties`

```
jonathan.connectionfactory.host=localhost

#http.proxyHost =
#http.proxyPort =
#http.proxyAuth = login:password
```

Note

It's recommended to write a completely qualified hostname (host.domain) or IP for *.host properties, instead of localhost.

To start the JADE boot from `<jade-boot-home>/` :

```
$ sh jadeboot.sh

Welcome to Felix.
=================

JadeBoot starting ...
Fractal registry is started on port <jasmine-control-node-port>
[NodeDiscovery service] listen on port 9998
[NodeLauncher] Node "<jasmine-control-node-address>_0" registered
[Joram server] started
[NodeDiscovery service] started
[JNDI] connected to fr.dyade.aaa.jndi2.client.NamingContextFactory on
 <jasmine-control-node-address>:1239
[Allocator] started
[Deployer] started
[Registry cleaner] started
JadeBoot started
```

<jasmine-control-node-address> is the name of the machine and <jasmine-controle-node-port> is the port number where the Fractal registry will listen.

The next step is to run the jade nodes on every host part of the managed cluster.

2.4. JADE Node

All the nodes that will be monitored and managed need a JADE node package.

The JadeNode can be configured through 2 properties files :

- `<jadenode-folder>/conf/config.properties`

```
jadeboot.registry.host=localhost
jadeboot.registry.port=1238

jadeboot.discovery.host=localhost
jadeboot.discovery.port=9998
```

- `<jadenode-folder>/conf/system.properties` containing the same properties than the JadeBoot system.properties.

To start the JADE node from <jade-node-home>/:

```
$ sh jadenode.sh

Welcome to Felix.
=================

JadeNode starting ...
Fractal Registry: <jasmine-control-node-address>:1238
[NodeLauncher] Node "<jasmine-monitored-node-address>_<node_id>" registered
[Heartbeat] started
JadeNode started
```

If the connection is successful, this message will be displayed in the jadeboot node console:

```
[Allocator] receive newNode jmsMessage : <jasmine-monitored-node-address>_<node-id>
```

2.4.1. User permissions & Apache

The administrator has to take care of the permissions of the user that is running the jadenode, because JADE may need to deploy Apache HTTPd and start it, the user will need some grants to launch it on a restricted port (<1024).

A simple solution is to give to the JADE user some admin rights. But a better solution is to use `sudo` for executing apachectl (the script that launches Apache HTTPd).

To do that it's necessary to edit `/etc/sudoers` file using the `visudo` command (we'll need root access) and add this line:

```
# User privilege specification <user-for-jadenode>
<jasmine-host-name> = (root)
NOPASSWD:<path-to-the-deployed-apachectl>
```

Note

This solution is just necessary if we need an Apache HTTPd server to listen at port 80 (or <1024). Otherwise the user will not need administrator rights and this part can be skipped.

Chapter 3. JASMINe Control Package

3.1. Pre-requisites

- JDK 5.0
- JOnAS 4.8.4
- Configure the environment variable $JASMINE_RULES. This variable indicates the directory where the files .xml and .drl are saved for the server. For example:

```
export JASMINE_RULES=/home/jasmine/Jasmine_rules/
```

3.2. Install the module

It's necessary to have a JOnAS installed. `JONAS_BASE` must be set. Unzip jasmine-control.zip and execute the Ant installer:

```
$ ant install
Buildfile: build.xml

init:

install:
     [copy] Copying 4 files to /home/user/JONAS_4_8_4

BUILD SUCCESSFUL
Total time: 1 second
```

It will automatically do:

- Copy the `drools.rar` file in the `$JONAS_BASE/rars/autoload/` directory. This is a resource adapter for the rule engine Drools.
- Copy the `jasmine-shared.jar` file in the `$JONAS_BASE/lib/ext/` directory. A common library.
- Copy the `jasmine-rules.jar` file in the `$JONAS_BASE/ejb3s/` directory. One EJB which parses the XML and uses Drools. It contains a second EJB for creating rule logs.
- Copy the `jade-ejb.jar` file in the `$JONAS_BASE/ejb3s/` directory. The EJB that comunicates with the UI.
- Copy the `defaultRules.drl.xml` and `test.drl.xml` files in the $JASMINE_RULES directory. This file contains the default rules for the autonomous module.

3.3. Run the module

Once everything is installed, JOnAS must be started:

```
$ jonas start
JONAS_BASE set to /home/user/JONAS_4_8_4
2007-04-11 14:50:35,452 : Server.<init> : JVM used is ...

...

2007-04-11 14:50:42,019 : Rar.processRar : Starting deployment of
 /home/user/JONAS_4_8_4/rars/autoload/drools.rar
2007-04-11 14:50:42,041 : Logger.info : Configuration of the packageBuiler for the new
 ruleBase...
2007-04-11 14:50:42,148 : Logger.info : Load the file JASMINE_RULES/defaultRules.drl.xml
2007-04-11 14:50:43,757 : Logger.info : Init the Working Memory...
2007-04-11 14:50:43,820 : Rar.processRar : /home/user/JONAS_4_8_4/rars/autoload/drools.rar
 available
```

```
2007-04-11 14:50:44,642 : Rar.processRar : Starting deployment of
 /home/user/JONAS_4_8_4/rars/autoload/ow_easybeans_for_jonas.rar

...

2007-04-11 14:50:59,226 : ComponentManager.startComponents : [ Component(s) started : Carol
 SmartClientEndPoint ]
2007-04-11 14:50:59,282 : ContainersMonitor.scanNewContainers : Creating container for archive
/home/user/JONAS_4_8_4/ejb3s/jasmine-rules.jar.
2007-04-11 14:50:59,569 : ENCManager.getInterceptorClass : Detecting JOnAS : using JOnAS ENC for
 the naming.
2007-04-11 14:50:59,585 : JContainer3.start : Analyze elapsed during : 223 ms
2007-04-11 14:50:59,749 : JContainer3.start : Enhancement elapsed during : 135 ms

...

2007-04-11 14:51:00,324 : Ejb3Configuration.scanForClasses : found EJB3 Entity bean:
 org.objectweb.jasmine.rules.logs.LogEntity
2007-04-11 14:51:00,700 : Configuration.addResource : Reading mappings from resource:
 META-INF/orm.xml
2007-04-11 14:51:00,702 : Ejb3Configuration.addClassesToSessionFactory : [PersistenceUnit:
 entity] no META-INF/orm.xml found
2007-04-11 14:51:00,801 : AnnotationBinder.bindClass : Binding entity from annotated class:
 org.objectweb.jasmine.rules.logs.LogEntity
2007-04-11 14:51:00,928 : EntityBinder.bindTable : Bind entity
 org.objectweb.jasmine.rules.logs.LogEntity on table LogEntity

...


2007-04-11 14:51:02,269 : ContainersMonitor.scanNewContainers : Creating container for archive
/home/user/JONAS_4_8_4/ejb3s/jade-ejb.jar.
2007-04-11 14:51:02,289 : JContainer3.start : Analyze elapsed during : 19 ms
2007-04-11 14:51:02,331 : JContainer3.start : Enhancement elapsed during : 40 ms2007-04-11
 14:51:02,385 :
JContainer3.start : Container started in : 114 ms

...


2007-04-11 14:51:12,099 : Logger.info : Execute the rules...
2007-04-11 14:51:12,099 : Logger.info : Execute the rules...
```

Chapter 4. User Interface

4.1. Install the JASMINe console

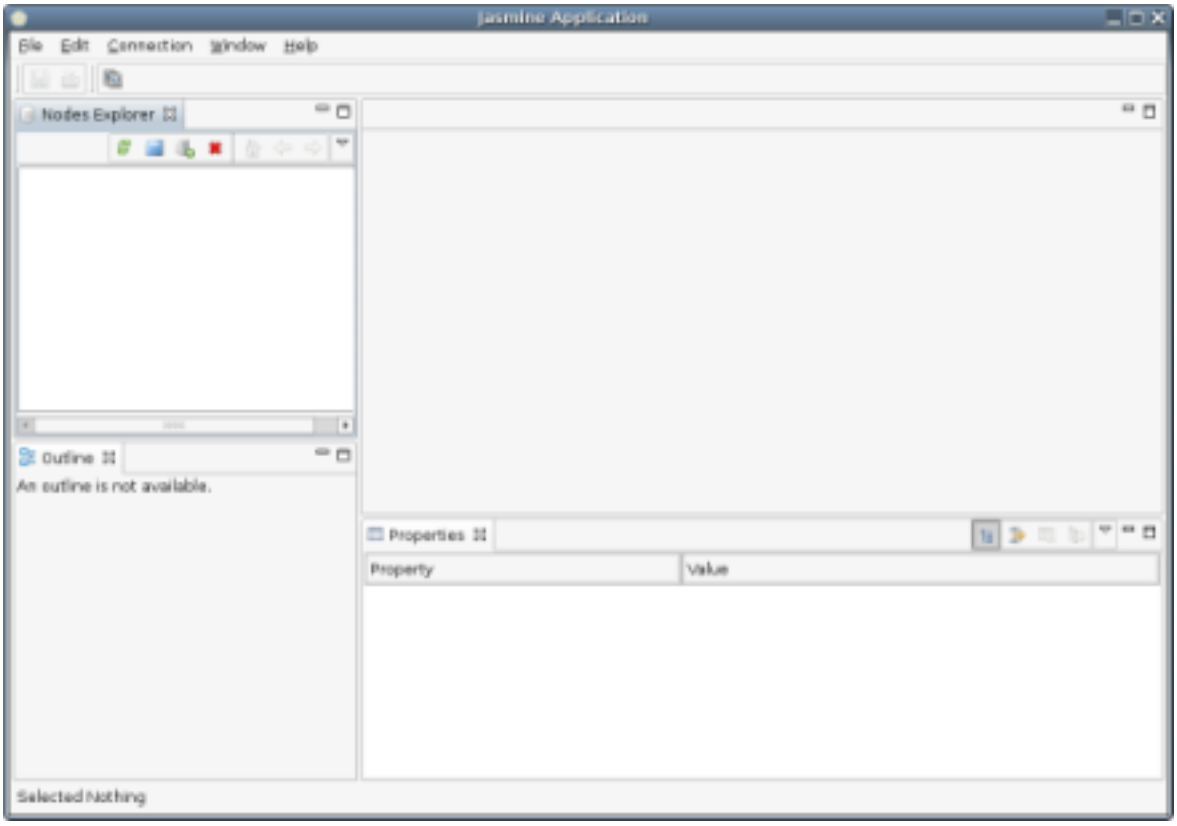
The UI is provided in a separate package named `jasmine-ui-<version>.zip`. For install JASMINe, just unzip this file in a separate directory.

4.2. Configure the JASMINe console

You can modify the `configuration/config.ini` to configure the UI.

4.3. Run the JASMINe console

To launch the UI, you just have to launch `./jasmine` or `jasmine.exe`.



Note

Known bug : The ActionBar of the "Nodes Explorer" view doesn't appear correctly. You can fix this by expandind and reducing it.

4.4. Create a JASMINe diagram

In the menu, File > New > Jasmine Diagram

The Connection Manager window opens.

The different parts are:

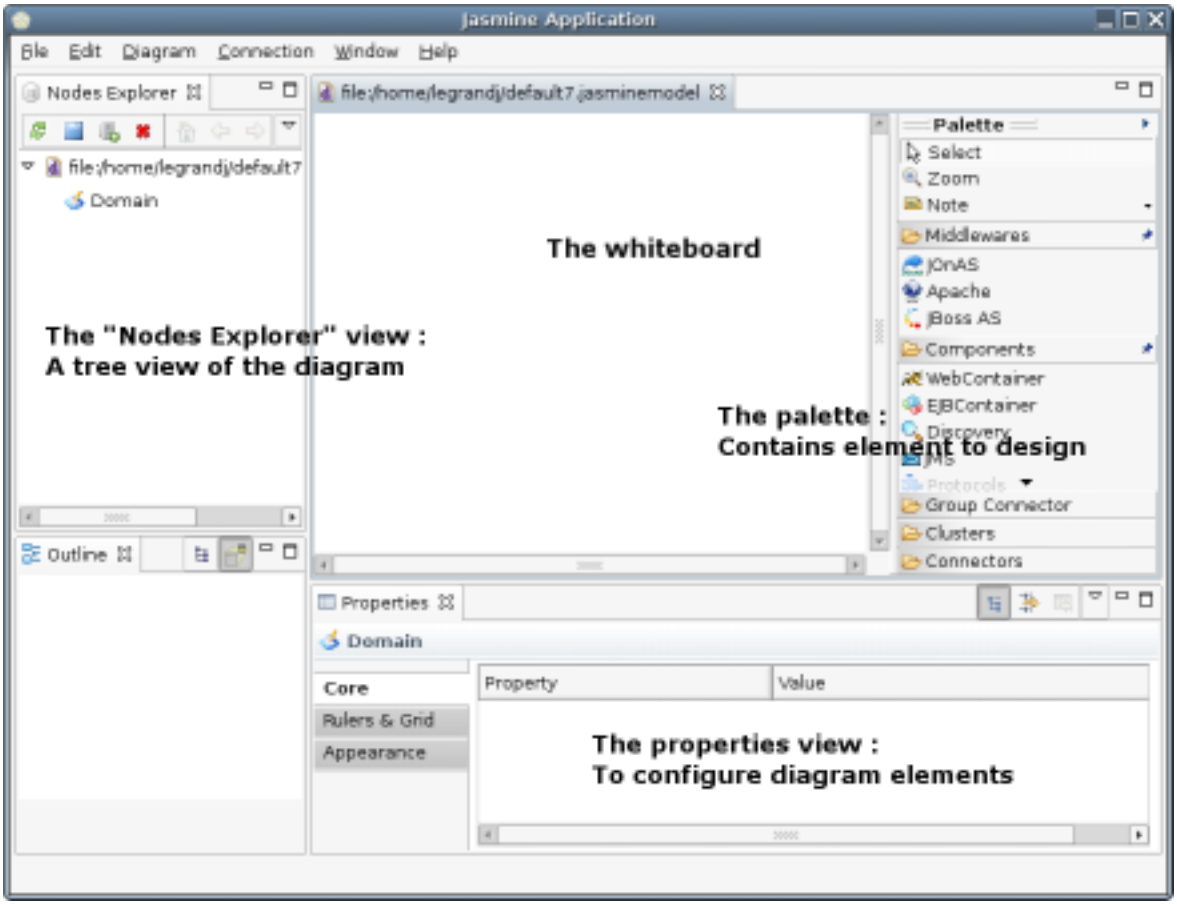
- Jade Front: Corresponds to the connection with the `jade-ejb.jar` EJB. It also needs the location where the JMX bridge creates the RMI registry and the location where the jadeboot creates the Fractal RMI registry.
- Probe Front: It's the connection used for the real-time monitoring. It creates a subscriber to the JMS topic where the monitoring system sends the data.
- Rulelogs Front: It connects with the logs EJB and it's used for retrieving from the UI the logs data.

In each part, there are a connection status, the IP and the port for each part of the server side. There are three possible status :

- Connected: the connection is established.
- Disconnected: the connection has been closed.
- Connection Failed: there is a problem. The connection can't be established.

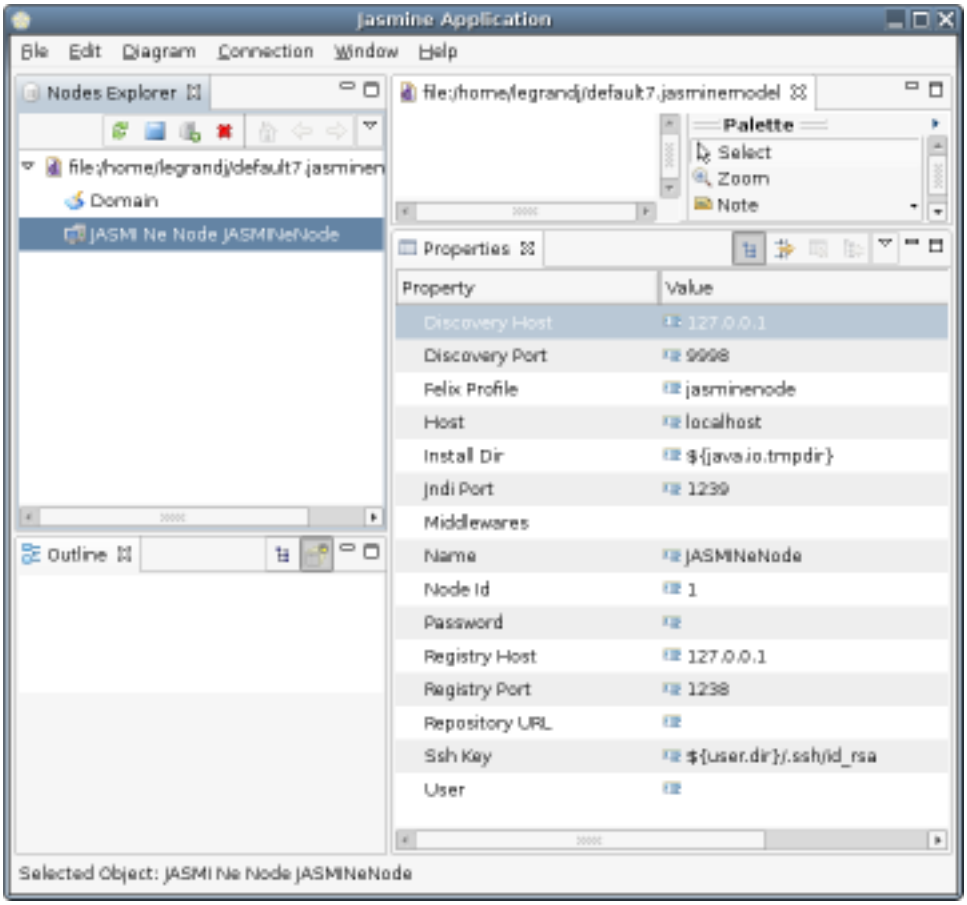
It's possible change the IP and the port for each part and test the new values with the connection button. To continue starting of JASMINe, press the continue button. It's possible to start JASMINe even if a part isn't connected.

Now, you should have a blank JASMINe diagram and the different views of the console.



4.5. Create the JASMINe nodes

Right-click on the "Nodes Explorer" view : a pop-up menu, with 2 entries, should appear. Choose the "Add Node" menu. A new JASMINe node has been created. You can configure it in the properties view



A JASMINe node represents a host inside the cluster.

Currently, only 2 properties are important :

- Node ID : This is the ID given by the jadeboot to the node. By default is 0, but if we start more than one jadenode on a host, it will be incremented.

```
[java] [Allocator] receive newNode jmsMessage : <jasmine-monitored-node-address>_<node-id>
```

- Host :

The other properties are not currently used.

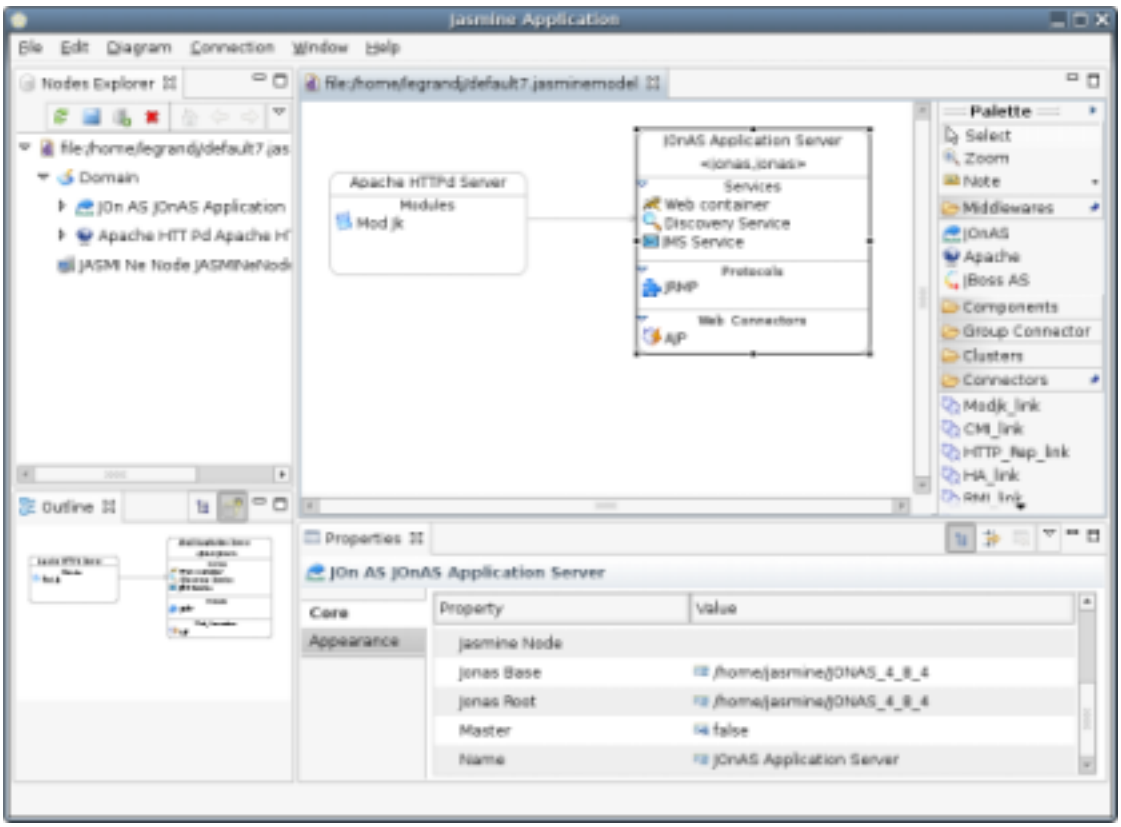
4.6. Design an architecture

Note

WARNING : In this version, you should manually synchronize the properties view, the JASMINe diagram editor and the "Nodes Explorer" view. This can be done with the "refresh" button

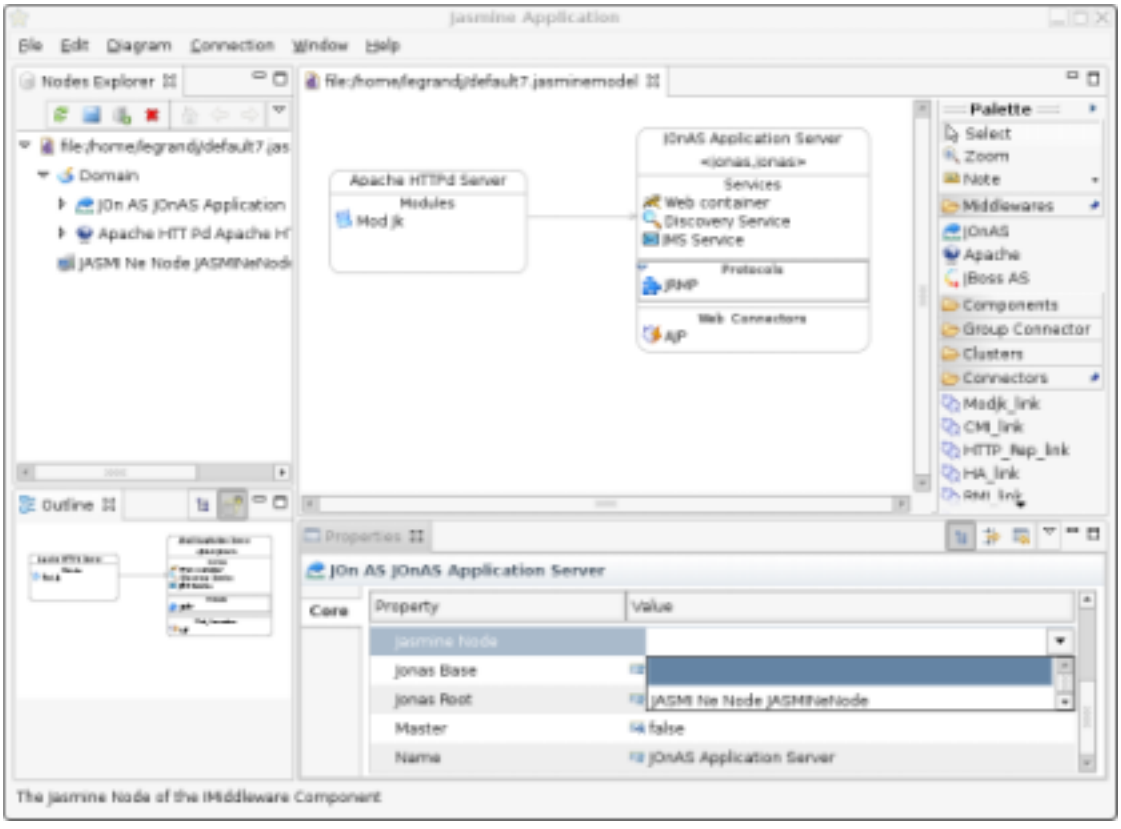
of the "Nodes Explorer" view. This synchronization issue will be fixed as soon as possible.

Choose an element in the "palette tool" and add it on the diagram editor.



To create connections between middleware, use the connectors in the "palette tool".

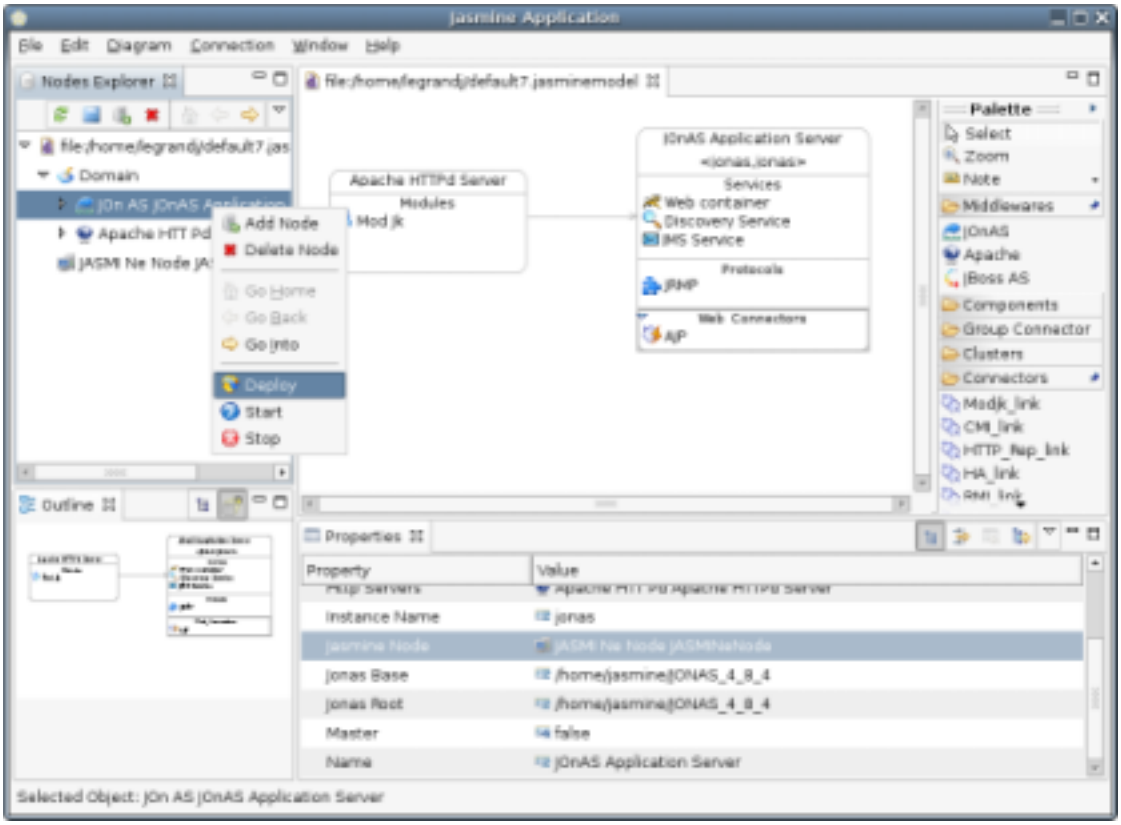
The next step is to choose the JASMINe node where the middleware will be deployed.



Your architecture is now ready to be deployed and started.

4.7. Deployment and life cycle of a middleware

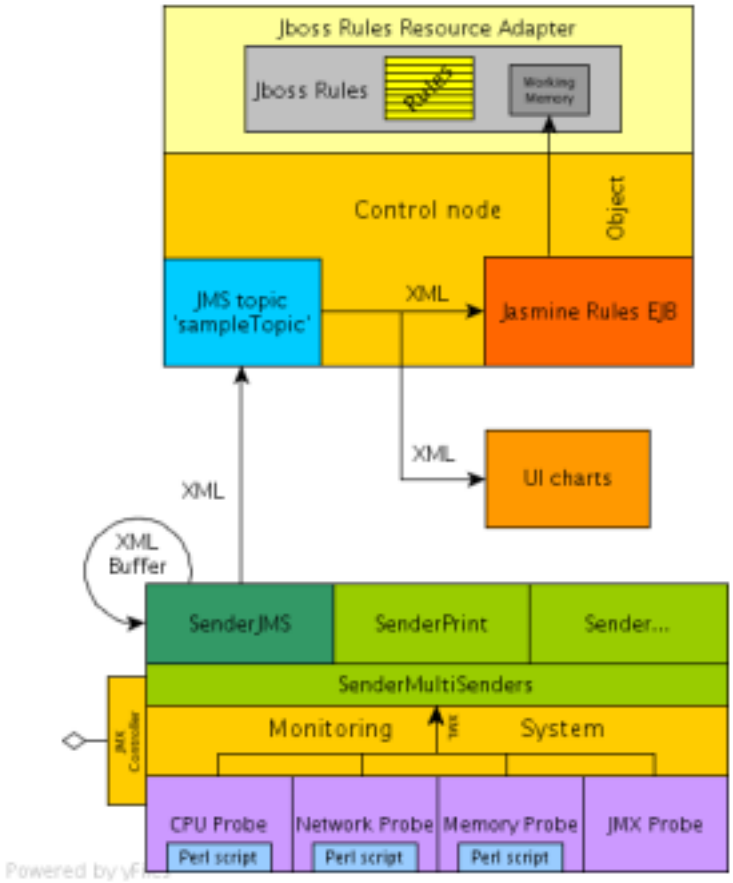
Deployment and lifecycle operations can be launch from the "Nodes Explorer" view, with a right-click on the middleware to manage.



Chapter 5. The monitoring system

5.1. Structure

The monitoring system consists in a package of probes which can be controlled via JMX. This schema will help us to understand its structure:



The probes send the data in XML format, and the monitoring system will nest this data inside another XML with the host descriptors. The SenderMultiSenders will give this data to all the senders we have specified in our monitoring configuration. This can be done editing the fractal file `org.objectweb.jasmine.monitoring.composite.Sender.fractal`

The JMS sender will deliver the XML data to the 'sampleTopic' topic in the JMS server. In JASMINe, the subscribers to this topics are the user interface, which will use this data to draw the graphs; and the rules system, where we have a MessageDrivenBean listening to the topic. It will parse the XML data and will deliver the object containing it to the working memory of Jboss Rules™. It will use the rules we provide (contained in the JASMINE_RULES directory) to analyse the data and act if necessary.

5.2. Installation

Right now, the monitoring system, can be deployed using the user interface by doing:

1. Right-click on Domain.
2. Choose New Child/Monitoring System