

JASMINe User's Guide

JASMINe User's Guide

Table of Contents

1. Introduction	1
1.1. Goal	1
1.2. Benefits	1
1.3. JADE	1
1.4. Schema	2
2. JASMINe JADE Package	3
2.1. Pre-requisites	3
2.2. JADE control through JMX	3
2.3. JADE Boot	3
2.4. JADE Node	4
2.4.1. User permissions & Apache	4
3. JASMINe Control Package	6
3.1. Pre-requisites	6
3.2. Install the module	6
3.3. Run the module	6
4. User Interface	8
4.1. Install the UI	8
4.2. Configure the UI	8
4.3. Running the UI	8
4.4. Creating a new modelcluster	10
4.5. Creating the Servers and Jasmine Hosts	13
4.6. Creating the Applications	14
4.7. Deployment and life cycle of an application	15
4.8. Using JOnAS Admin with JASMINe	16
4.9. Monitoring from the UI	16
4.10. Rules logs	19

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

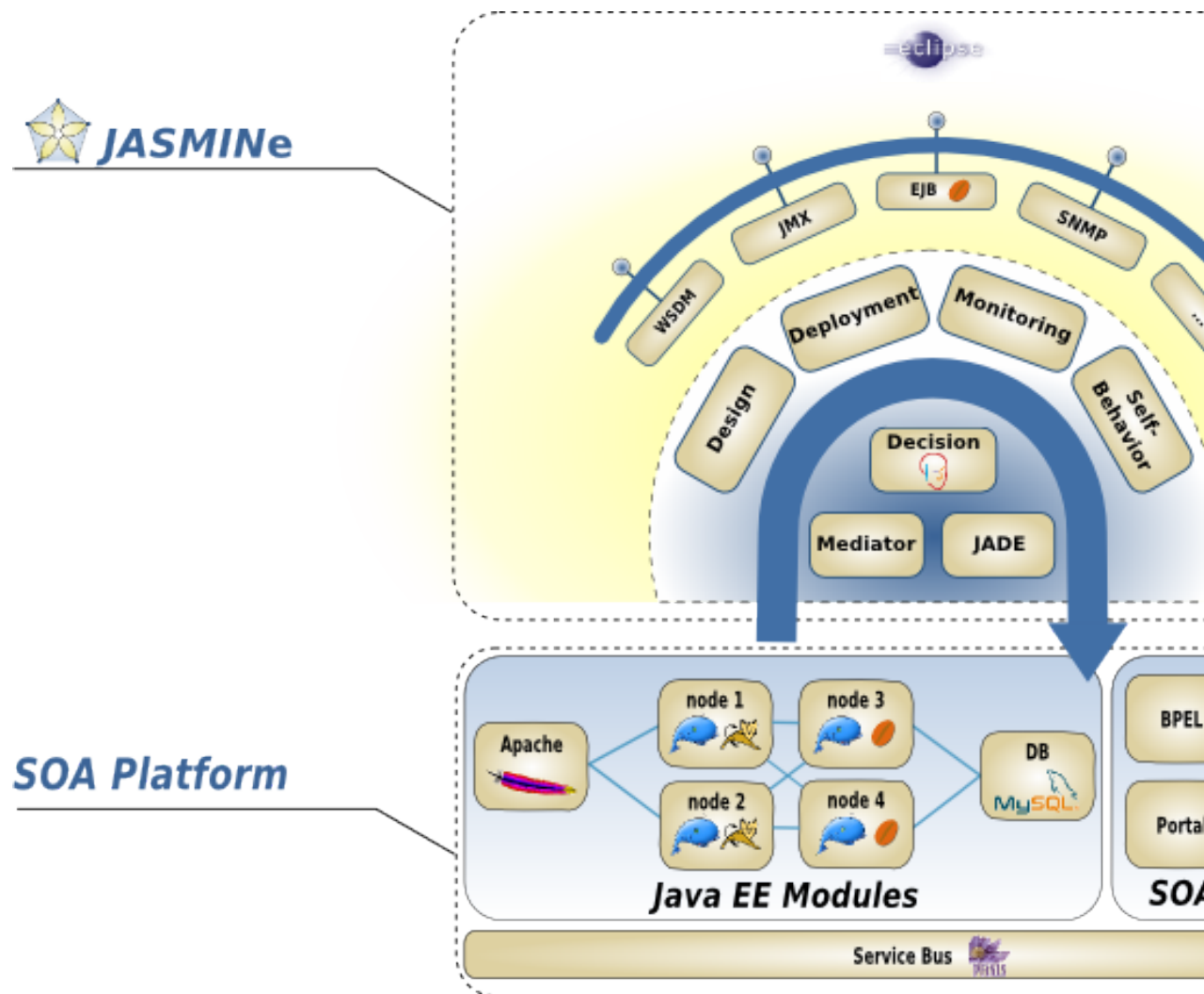
JASMINe main benefits are:

- 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 occurring 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 these files:

- `<jade-folder>/etc/osgi/system.properties` should be modified in:

```
oscar.auto.start.1= \
file:///<repository-location>/shell.jar \
file:///<repository-location>/bundlerepository.jar \
file:///<repository-location>/logger.jar \
file:///<repository-location>/log.jar \
file:///<repository-location>/properties.jar \
file:///<repository-location>/fractalbootstrap-2.1.0.1.jar
```

- `<jade-folder>/etc/osgi/bundle.properties` should be modified in:

```
oscar.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. To change this, use:

```
$ ant -DrmiPort=<port>
```

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.

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

```
$ ant start -DregistryHost=<jasmine-control-node-address>
-DregistryPort=<jasmine-control-node-port>
Buildfile: build.xml

compile:

start:
 [java] JadeBoot starting ...
 [java] Fractal registry is ready.
```

```
[java] [NodeLauncher] Node "<jasmine-controlle-node-address>_<node-id>" registered
[java] [OSGI] starting oscar
[java] INFO : 3 : Logger.start : 1.0.0
[java] INFO : 5 : ServiceEvent.REGISTERED
[java] INFO : 5 : ServiceEvent.REGISTERED
[java] INFO : 6 : ServiceEvent.REGISTERED
[java] INFO : 4 : BundleEvent.STARTED
[java] INFO : 5 : BundleEvent.STARTED
[java] INFO : 6 : BundleEvent.STARTED
[java] INFO : 0 : BundleEvent.STARTED
[java] INFO : 0 : FrameworkEvent.STARTED
[java] [OSGI] oscar started
[java] [JoramServerWrapper] server started
[java] [UDPNODEDiscovery] listen on port 9998
[java] [JNDI] connected to fr.dyade.aaa.jndi2.client.NamingContextFactory on
<jasmine-controlle-node-address>:1239
[java] [Allocator] started
[java] JadeBoot started
[java] [Allocator] receive newNode jmsMessage : jadeboot
```

<jasmine-controlle-node-address> is the name of the machine and <jasmine-controlle-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.

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

```
$ ant start -DregistryHost=<jasmine-control-node-address>
-DregistryPort=<jasmine-controlle-node-port>
Buildfile: build.xml

compile:

start:
    [java] JadeNode starting ...
    [java] [NodeLauncher] Node "<jasmine-monitored-node-address>_<node-id>" registered
    [java] [OSGI] starting oscar
    [java] INFO : 3 : Logger.start : 1.0.0
    [java] INFO : 4 : BundleEvent.STARTED
    [java] INFO : 5 : ServiceEvent.REGISTERED
    [java] INFO : 5 : ServiceEvent.REGISTERED
    [java] INFO : 5 : BundleEvent.STARTED
    [java] INFO : 6 : ServiceEvent.REGISTERED
    [java] INFO : 6 : BundleEvent.STARTED
    [java] INFO : 0 : BundleEvent.STARTED
    [java] INFO : 0 : FrameworkEvent.STARTED
    [java] [OSGI] oscar started
    [java] JadeNode started
```

It's important to write a completely qualified hostname (host.domain) or IP, as sometimes it fails to find the nodes. It's possible to have some problems trying to connect nodes at different networks, so it's recommended, if it's possible, to have them at the same network. If the connection is successful, this message will be displayed in the jadeboot node console:

```
[java] [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 communicates 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 packageBuilder 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 UI

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 UI

Inside the `jasmine-gui` folder there is one folder called `conf`, which contains configuration files. The `jasmine.properties` file contains the listening address and ports of the node which acts as a control node. The other files (`apache.properties`, `jonas.properties`, ...) contain the default values used for the wrappers in the deployment. These values can be changed later in the GUI. These properties allow us to customize the wrappers to each node dynamically.

4.3. Running the UI

If `jasmine.properties` aren't correctly configured or there is a problem on the sever side, the following screen will appear.

The screenshot shows a Java Swing window with a blue title bar. The window contains three vertically stacked sections, each with a title and several input fields. The first section, 'jade front', has a 'Connection status' field with the text 'CONNECTION_FAILED'. Below it are six input fields: 'EJB Ip', 'EJB Port', 'JMX Ip', 'JMX Port', 'RMI Ip', and 'RMI Port'. The second section, 'Probe front', has a 'Connection status' field with 'CONNECTION_FAILED' and two input fields: 'Ip' and 'Port'. The third section, 'Rulelogs front', also has a 'Connection status' field with 'CONNECTION_FAILED' and two input fields: 'Ip' and 'Port'. At the bottom of the window are four buttons: 'Connection', 'Decor', 'Continue', and 'Ca'.

Section	Connection status	EJB Ip	EJB Port	JMX Ip	JMX Port	RMI Ip	RMI Port	Ip	Port
jade front	CONNECTION_FAILED								
Probe front	CONNECTION_FAILED								
Rulelogs front	CONNECTION_FAILED								

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.

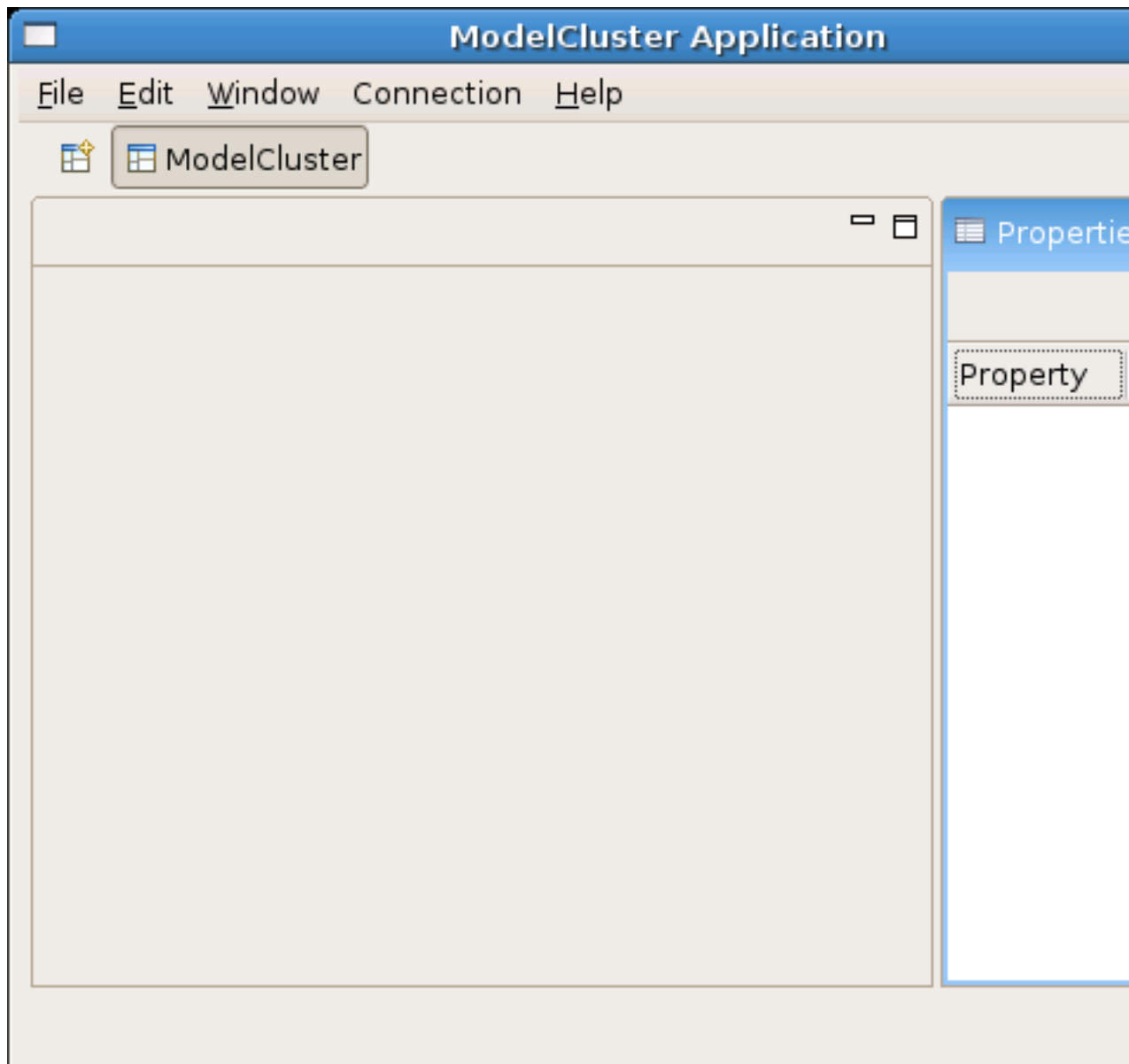
In each part, there are a connection status, the IP and the port for each part of the server side. There are three possible status :

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

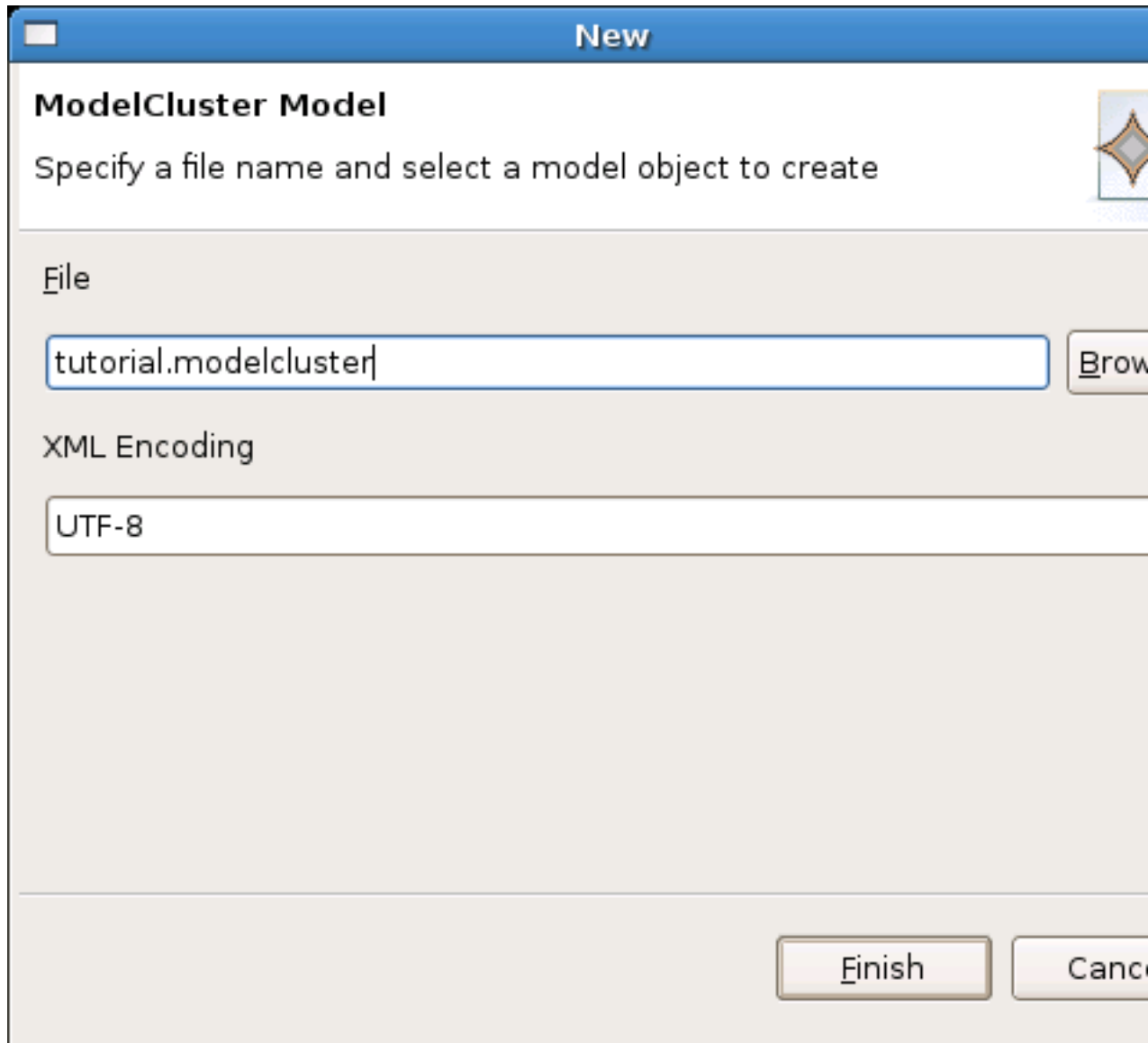
It's possible to change the IP and the port for each part and test the new values with the connection button. To continue starting of JASMINe, press the continue button. It's possible to start JASMINe even if a part isn't connected.

4.4. Creating a new modelcluster

This is the main screen of JASMINe, the ModelCluster perspective. This perspective is used to design and configure the cluster.



For creating a new project, click on File -> New -> ModelCluster Model and write the name of the project. The project will be saved in a modelcluster file with this name. It's possible to use the Browse... button to choose the desired folder for saving the project.



The image shows a 'New' dialog box with a blue title bar. The main title is 'ModelCluster Model'. Below it, the instruction 'Specify a file name and select a model object to create' is displayed. There is a 'File' section with a text input field containing 'tutorial.modelcluster' and a 'Browse...' button. Below that is an 'XML Encoding' section with a dropdown menu set to 'UTF-8'. At the bottom right, there are 'Finish' and 'Cancel' buttons.

New

ModelCluster Model

Specify a file name and select a model object to create

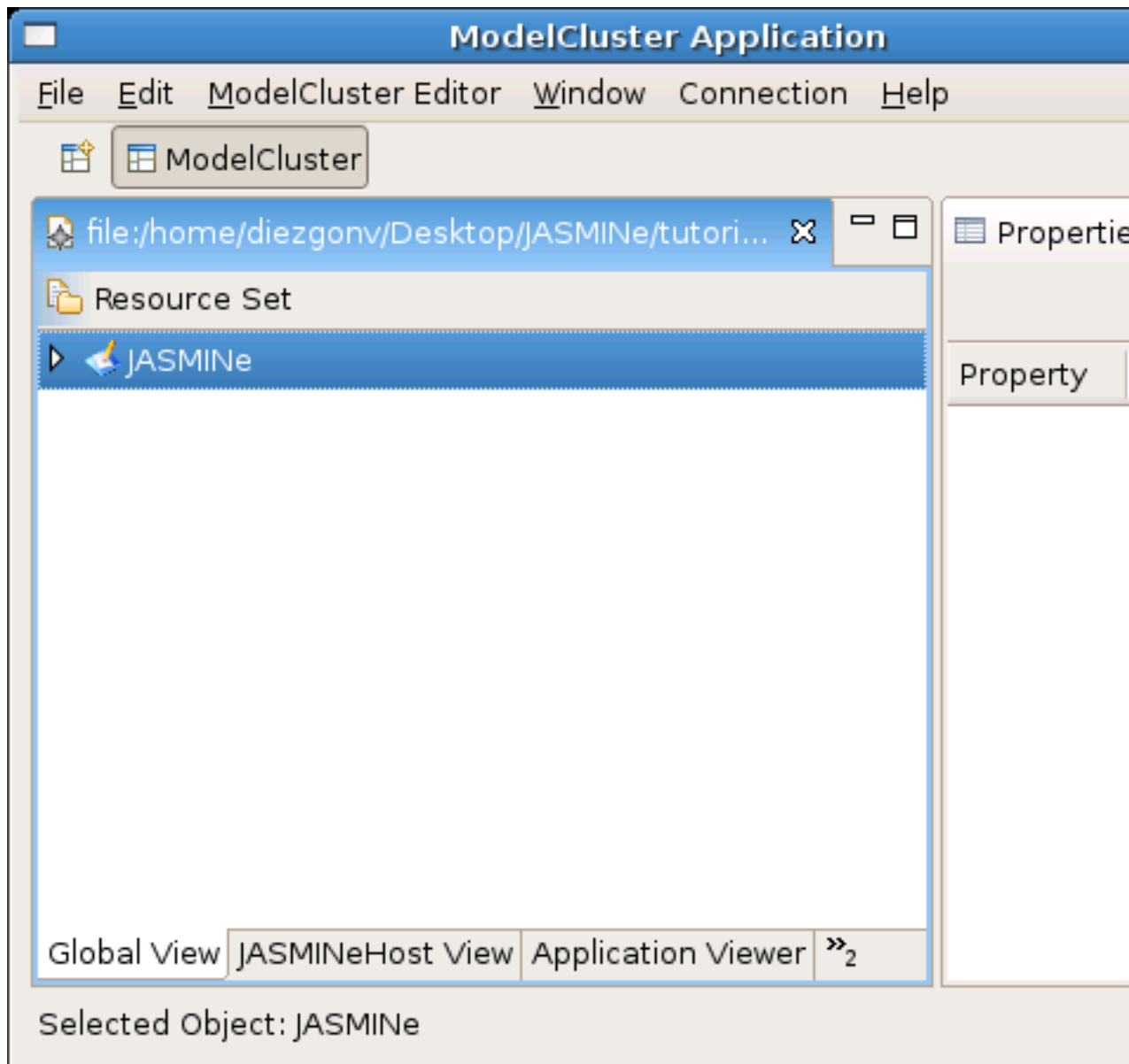
File

tutorial.modelcluster

XML Encoding

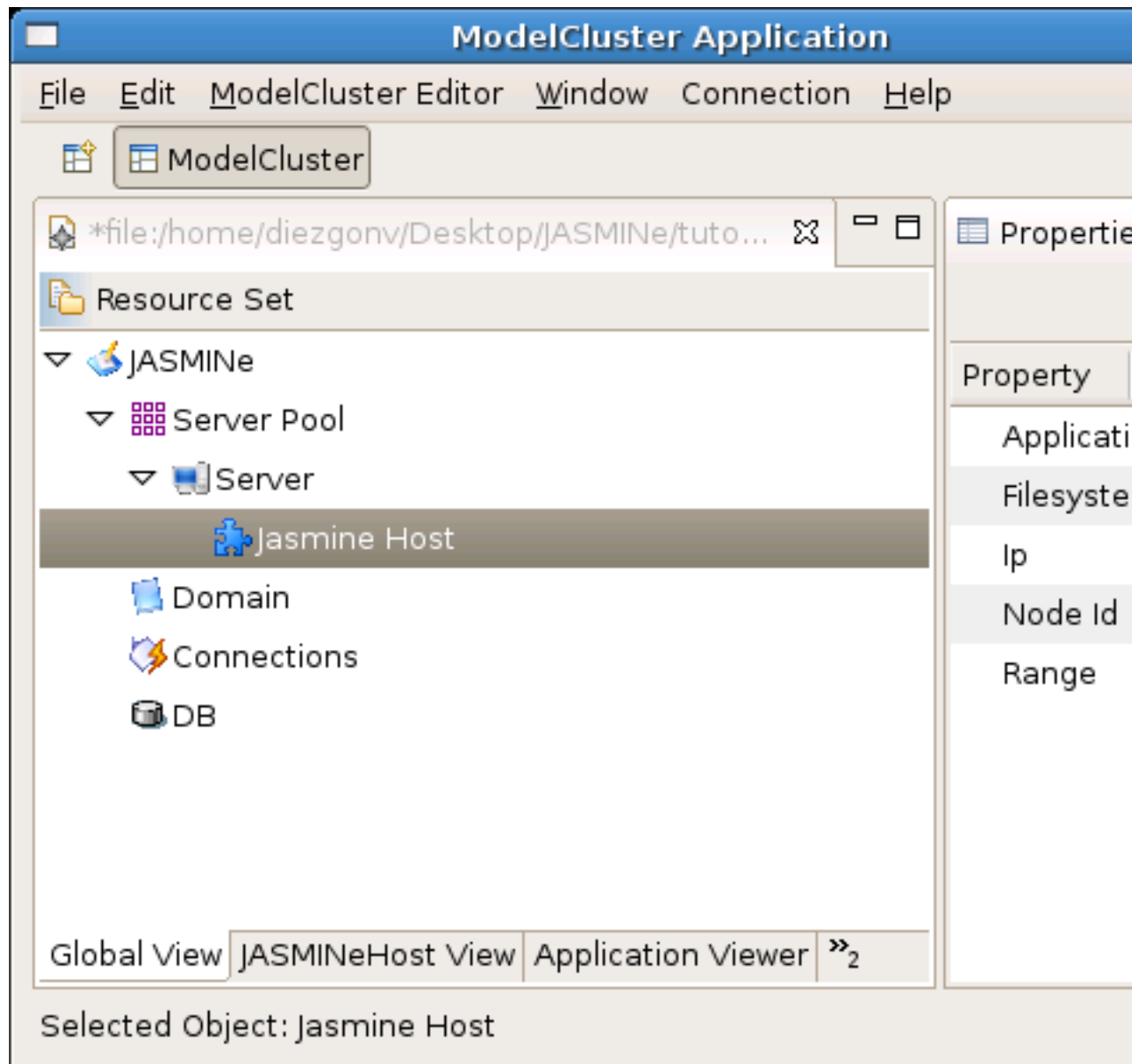
UTF-8

After clicking Finish the project will be created and it will return to the main screen:



4.5. Creating the Servers and Jasmine Hosts

1. Extend Jasmine (click on the arrow).
2. Right-click on Server Pool.
3. Choose New Child/Server.
4. Right-click on the new Server.
5. Choose New Child/Jasmine Host.



A server represents a machine inside the cluster. A Jasmine Host corresponds to a jadenode, to make an equivalence between Jade and the UI. The node id property corresponds to the number given by the jadeboot to the nodes. By default is 0, but if we start more than one jadenode by machine, it will increase. The jadeboot will tell us this number when the nodes connect:

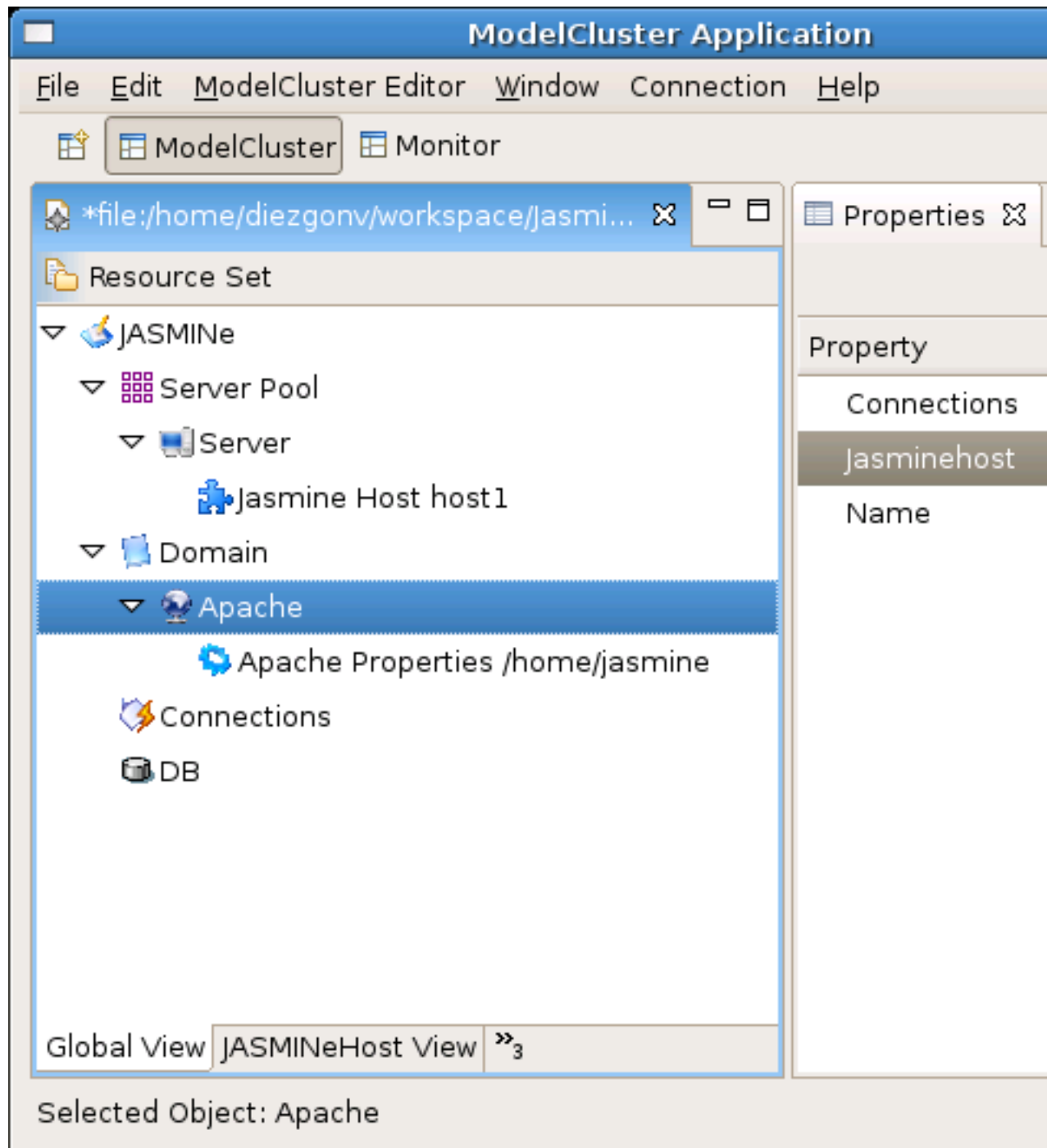
```
[java] [Allocator] receive newNode jmsMessage : <jasmine-monitored-node-address>_<node-id>
```

The Applications property contains all the applications in the domain that are associated with the node. For the Ip property we can specify the hostname or the IP. The rest of the properties are not currently used.

4.6. Creating the Applications

For each application to deploy the steps to follow are:

1. Right-click on Domain.
2. Choose New Child/<Application>



For deploying an application click on Jasmine Host property in the Properties view and choose the Jasmine Host on which you want to use the Application.

4.7. Deployment and life cycle of an application

To deploy an application, right-click on the application you want to deploy and choose Deploy on the contextual menu.

To start an application, right-click on the application you want to start and choose Start on the contextual menu.

4.8. Using JOnAS Admin with JASMINe

1. Choose a JOnAS which is running.
2. Right-click on the JOnAS.
3. Choose Browser on contextual menu.
4. JASMINe will automatically switch to the browser perspective.
5. In this perspective, you can open several browsers for each JOnAS.

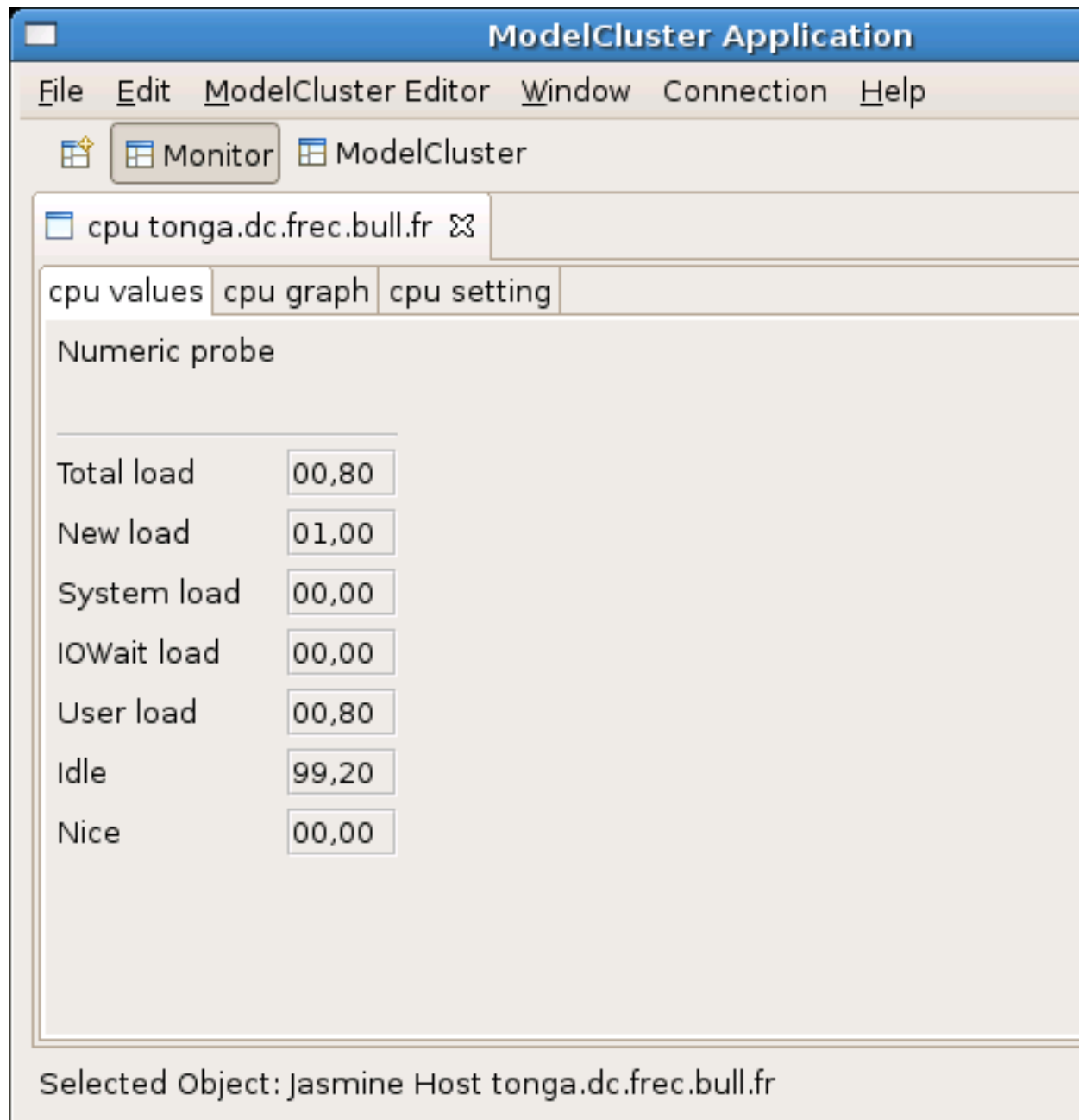
4.9. Monitoring from the UI

To monitor a Jasmine Host, the monitoring system must be deployed and started on that node:

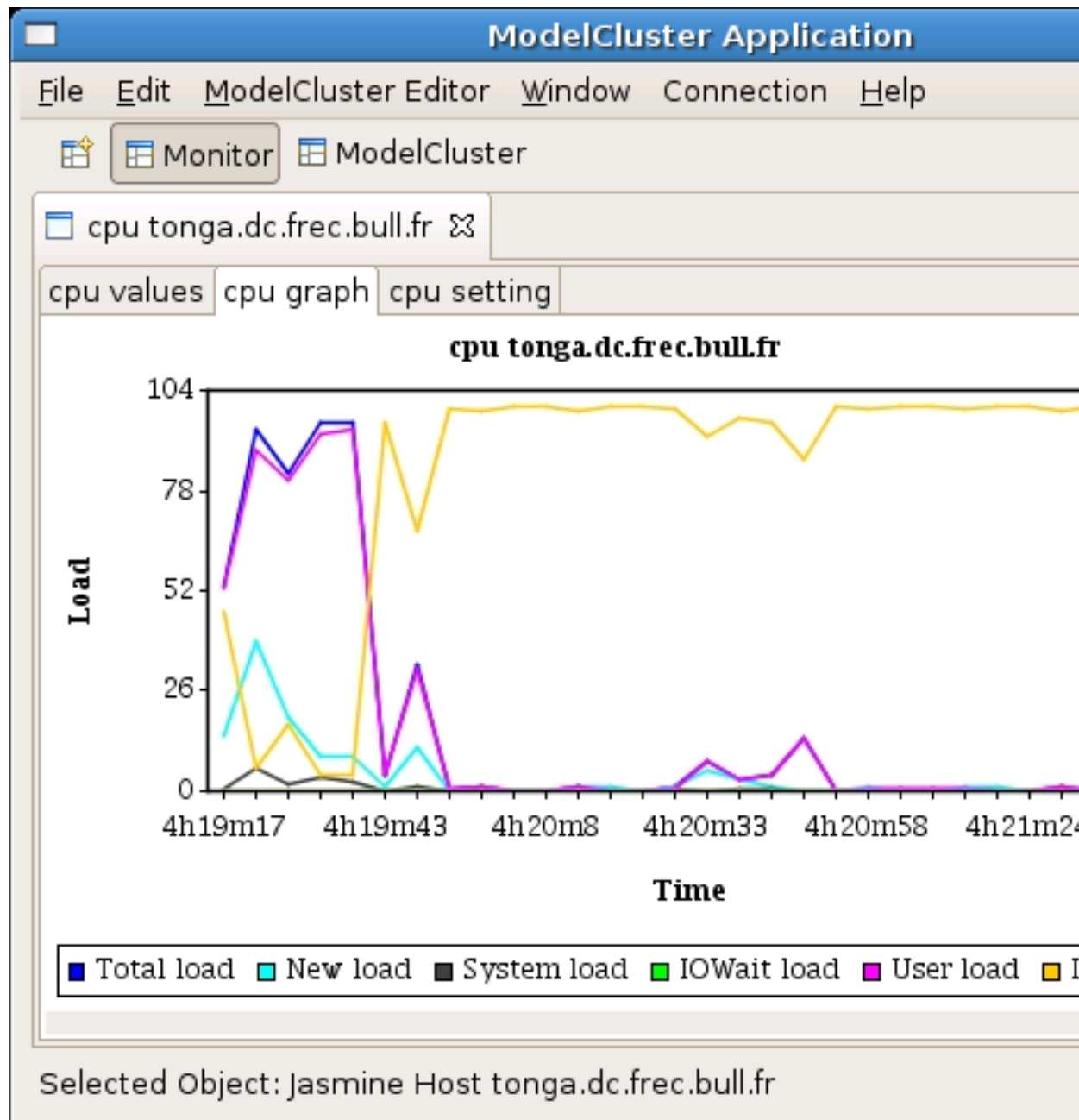
1. Right-click on a Jasmine Host.
2. Choose one of the three possible monitors inside the Monitoring option.
3. JASMINe will automatically switch to the monitoring perspective.

For each view of monitoring, you have three tabs:

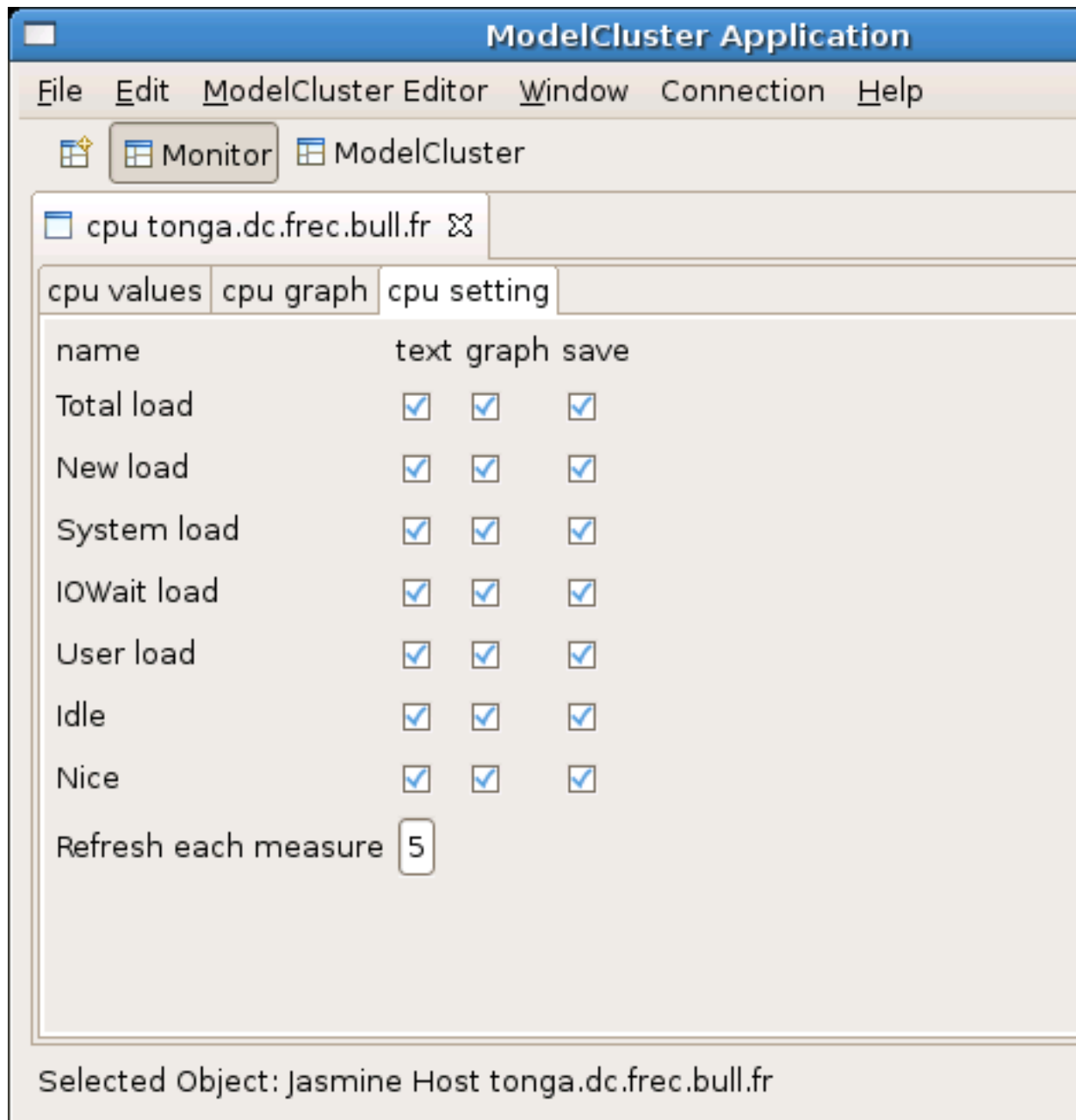
- The values tab where you can read numeric values.



- The graph tab where you can see curves.



- The setting tab where you can choose which data you want to display, and the refresh rate.



4.10. Rules logs

To view the rules logs, it's necessary to open the RuleLogs perspective. On this perspective there are several actions :

- To sort logs by type or date (push buttons on the top of the table).
- To filter logs by type or date.
- To save logs in an Excel file logs by type or date.
- To delete logs on the server side by type.
- To change the logs refresh rate (choose a value on the list).

The screenshot shows the 'ModelCluster Application' window. The menu bar includes 'File', 'Edit', 'ModelCluster Editor', 'Window', 'Connection', and 'Help'. Below the menu is a toolbar with icons for 'RuleLogs', 'Monitor', and 'ModelCluster'. The 'RuleLogs' tab is active, displaying 'Automanagement logs'. Above the log table are buttons for 'Filtre', 'Export', and 'Delete', along with a 'Refresh rate' set to '1 second'.

Level	Date	Message
ALERT	2007-04-13 15:55:53.74	CPU ALERT : cpuload = 96.41%
INFO	2007-04-13 15:55:54.166	CPU cpuload = 72.65% on ton
ALERT	2007-04-13 15:55:59.194	CPU ALERT : cpuload = 96.21%
INFO	2007-04-13 15:56:08.717	CPU cpuload = 16.37% on ton
INFO	2007-04-13 15:56:13.737	CPU cpuload = 3.99% on tong
INFO	2007-04-13 15:56:18.757	CPU cpuload = 56.29% on ton
ALERT	2007-04-13 15:56:23.757	CPU ALERT : cpuload = 96.21%
ALERT	2007-04-13 15:56:28.777	CPU ALERT : cpuload = 93.01%
WARN	2007-04-13 15:56:34.467	CPU WARN : cpuload = 80.84%
INFO	2007-04-13 15:56:39.476	CPU cpuload = 42.32% on ton
INFO	2007-04-13 15:56:44.496	CPU cpuload = 0.2% on tonga