

JIVA Grid Server For Windows User Manual

Document History

Version	Date	Comments
1.0	06/08/06	Initial Release

Introduction	2
Installation	3
1. Install the Server	3
2. Install Clients on your Network.....	3
3. Update your license	3
4. Start the Server	3
Analyzing your Network	4
Setting up a new task in the JIVA Servlet File System	5
1. Put your task jar in the tasks folder	5
2. Place your data files in the data folder	5
3. Modify JIVA's web.xml.....	6
4. Restart Tomcat.....	6
Distributing a new JIVA Client Jar to your Network.....	7
Customizing your JIVA Server Installation.....	7
Explanation of the Client File System.....	8
Floating Licenses, JMSL and FlexNET	9

```
// innovate  
public class BrainMurmurs extends Software  
implements Design, Development, Innovation {
```

Introduction

This documents describes the JIVA Server software that runs under Windows. It contains information about setting up, configuring and troubleshooting the JIVA Server. This document supplements the general JIVA documentation and it is a good idea to consult this first to get a general idea of how the JIVA architecture works. This document contains information about configuring JIVA tasks but information about the creation of new JIVA tasks is found in the JIVA SDK Documentation (forthcoming).

The JIVA Server is software that allows a network to perform computations in parallel, much the same way that a super computer does. It works by assigning a specialized kind of application called a **task** to the network. The task will be some sort of computational operation that you need to perform many times on different pieces of data. The JIVA Server distributes this task to a network of computers and assigns data to individual computers to be processed. A demo version of the JIVA server is available to give you an idea of how easy it is to set up a JIVA network and to demonstrate what kind of performance you can expect from your network. Once you've run the JIVA Windows Server installer most of the work has been done for you. To create a fully working grid network all you need to do is to install clients on all of the computers in your network and you're ready to go.

Installation

Setting up a functioning JIVA Network can be done in three steps.

1. Install the Server

Run the Server Installer available from the [Brain Murmurs website](#). This will install the Python programming language, Tomcat and the JIVA Server onto your computer.

2. Install Clients on your Network

The client installers for all supported operating systems can be downloaded at the Brain Murmurs website (<http://www.brainmurmurs.com/products/jiva/downloads/>).

Even though you are running the Windows server, your network can be made up of any combination of operating systems and client types. Each installer comes with instructions for setup. You will need to enter the IP address that your JIVA Server uses. You can find the IP address of your JIVA Server by typing ipconfig in a command prompt on the machine your JIVA Server is installed on.

3. Update your license

If you have purchased a license for your network, copy your ServerLicense.dat file into the JIVA Server install directory.

4. Start the Server

Just select start JIVA from your Windows "Start" menu. It will open a window that will report on JIVA's progress but it shouldn't require any further input from you. The JIVA Server will begin to distribute work to its clients. The Server comes preconfigured to distribute a task called Scimark2 which is a benchmark computation used in supercomputing. How to configure the JIVA Server to do other tasks (including those designed by you) will be explained below.

Analyzing your Network

The Scimark2 task that is preinstalled with your JIVA Server provides a means of analyzing the performance of your network. If you're evaluating JIVA for purchase, this is an excellent way to predict what kind of performance you'd expect from your network.

As the Scimark2 task runs, JIVA periodically gathers information about its performance and creates a report. Once you have the Server running the Scimark2 task, point a web browser on your server to <http://localhost/jiva/scripts/stats.html>. This page will show you the contributions that the different computers on your network are making. It will also compare your JIVA network to other computing options.

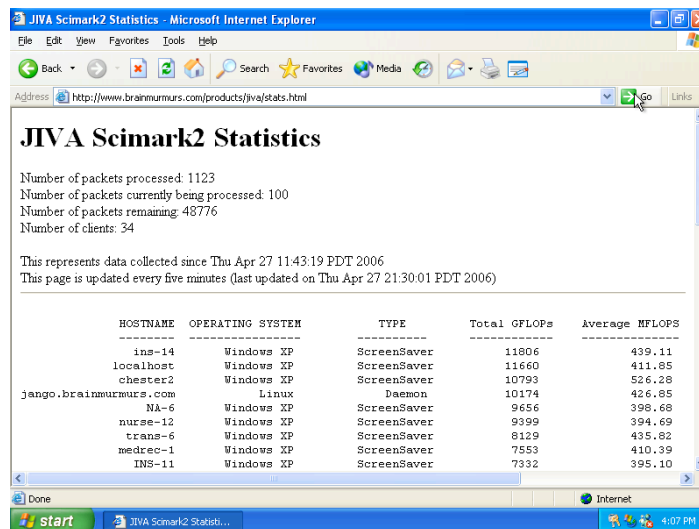


Figure 1 - JIVA Scimark2 Statistics

Setting up a new task in the JIVA Servlet File System

The purpose of the JIVA Server is to run task processes written in Java on a set of data files. To set up the JIVA Server to run a task you will need the following:

- * A JIVA task jar
- * A set of ".dat" input files for your task jar

1. Put your task jar in the tasks folder

After it is installed, the JIVA filesystem will be located in the "webapps" folder in your Tomcat installation. Within the JIVA install folder is a directory called "tasks." This is where you'll put your JIVA task jar. Within "tasks" create a subdirectory with the same name as the task jar without the ".jar" (i.e. foo.jar would go in the directory tasks/foo/).

2. Place your data files in the data folder

Within the JIVA folder there is a folder called "data." This is the folder that will contain all of your input and output data. Inside the data folder are three folders labeled "in", "out" and "proc." These work like the in/out boxes that people sometimes have on their desks. When you want a data file processed you put it in the "in" directory. While JIVA is working on the file, it will be moved to the "proc" directories. Once the data has been processed the original data along with its solution will be put into the "out" directory for your use.

JIVA can keep track of the data from multiple tasks at once. For this reason, you must put data files in a folder specific to their task. This folder name should be the same as the task jar without the ".jar". For example, if you wanted to run a task called foo.jar. You would create the folder jiva\data\in\foo and place all of your dat files inside it. JIVA will automatically create the directories for the out and proc directories.

```
// innovate
public class BrainMurmurs extends Software
implements Design, Development, Innovation {
```

3. Modify JIVA's web.xml

In the WEB-INF folder of your JIVA installation there is web.xml file. Open it with a text editor such as notepad and find the section that looks like this.

```
<init-param>
    <param-name>task_wwuid</param-name>
    <param-value></param-value>
</init-param>
```

Insert the name of your task jar (minus the ".jar") between the param-value tags. i.e.:

```
<init-param>
    <param-name>task_wwuid</param-name>
    <param-value>foo</param-value>
</init-param>
```

The name in the param-value should be the same as the name of the sub directories in data\in and tasks.

4. Restart Tomcat

This varies for different installations of tomcat. All clients configured to use this server will begin working on the task you've set up once they become available. You can check the jiva-server.log file in your Tomcat installation's logs folder to make sure that your server is running correctly. To configure logging see the section of this document called "Customizing your JIVA Server Installation" (below).

Distributing a new JIVA Client Jar to your Network

In the server's JIVA directory there is a directory called "client". If you replace the jar contained in this folder with a more recent version and restart Tomcat, the JIVA server will automatically distribute the new code to all of your clients. Be absolutely certain that the jar you place in this folder is called `jiva-client.jar` and that it is a working version of the JIVA client.

Customizing your JIVA Server Installation

JIVA has a number of parameters that you can customize to alter JIVA's performance. These properties are located in the `jiva\WEB-INF\web.xml` file and are stored in the following manner:

```
<init-param>  
  <param-name>The name of the parameter</param-name>  
  <param-value>The value of the parameter</param-value>  
</init-param>
```

The customizable parameters are as follows:

task_wwuid

The name of your JIVA task's jar (minus the `.jar` extension).

task_clz_name

The full (i.e. `com.brainmurmurs.example.myClass`) name of the class inside the jar specified by `task_wwuid`.

max_allowed_proc_time

The amount of time (in milliseconds) that a client has to finish processing a data file before the JIVA Server considers the data file "orphaned" and reassigns it to another client.

orphan_manager_sleep_time

The interval (in milliseconds) that JIVA waits between checking for expired data files.

log4j-config

The path of the configuration file used in logging.

```
// innovate  
public class BrainMurmurs extends Software  
    implements Design, Development, Innovation {
```

logging-active

Whether or not to create a log of JIVA server activity. Should be either true or false.

client_upload_dir

This is the directory of the folder that contains the jiva-client.jar file that is distributed to the clients.

in_dir, out_dir, proc_dir

The directory of JIVA's in, out and processing directories respectively.

license_location

The path of the license file for your JIVA Server. By default, this is installed in the JIVA install directory.

pystats_location

The path of the pystats.py file. The pystats.py is a python script that gathers JIVA performance data on the Scimark2 benchmarking task. By default pystats.py is installed in the "scripts" directory within the JIVA install directory. Pystats is run automatically by the server at intervals set in the web.xml file (see below).

pystats_update_frequency

The amount of time pystats waits between each run (in milliseconds).

Explanation of the Client File System

The JIVA client file system strongly parallels that of the JIVA Server. It contains a data folder that contains the data that the client is currently working on with a similar in/out/proc directory structure. The task jar is stored in the jiva-client/data/tasks folder.

Floating Licenses, JMSL and FlexNET

Many libraries and applications have a license scheme that allows an organization to run only a certain number of copies of a program at any given time. Thus a program could be installed on any number of computers so long as only some of them were running at any given time. Since in this scheme licenses “float” from computer to computer, these are called floating licenses.

One product that utilizes a floating license is Visual Numerics’ powerful JMSL Libraries. The licenses for JMSL are managed by a program called FlexNet. JIVA features built in support for JMSL and is capable of distributing and running JMSL programs as long as the clients are configured with the location of the FlexNet Server. Consult the client specific documentation for more information.

Removing the JIVA Grid Server

The JIVA Grid Server is removed by selecting it in the Add or Remove Programs control panel and clicking the remove button.

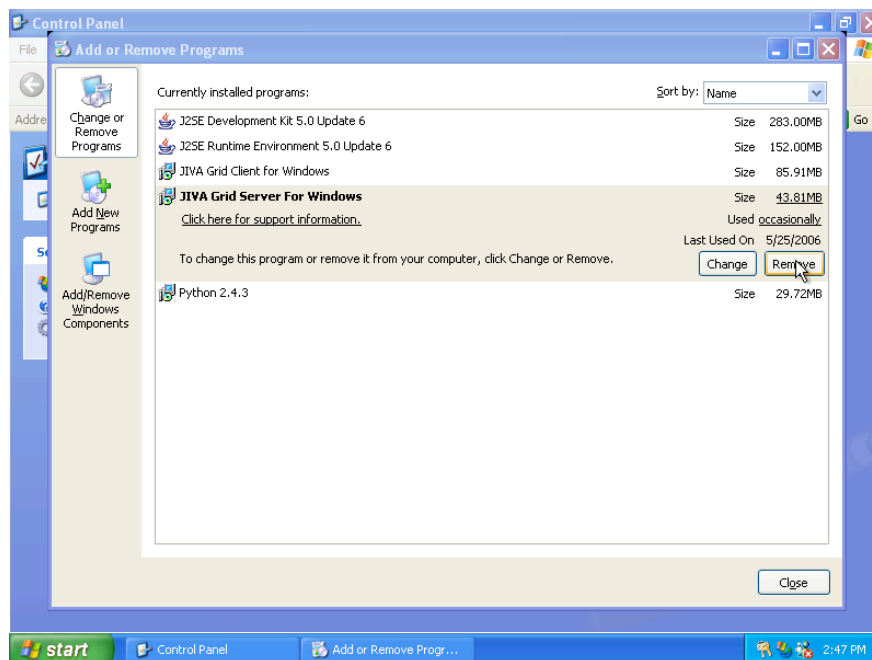


Figure 2 - Uninstalling the JIVA Server

```
// innovate  
public class BrainMurmurs extends Software  
    implements Design, Development, Innovation {
```

Trouble Shooting

If your JIVA Server isn't working

1. Make sure port 80 isn't blocked on your server. JIVA communicates with its clients through port 80. Certain firewalls may prevent this. You should be sure that your firewall allows connections on port 80.
2. Make sure your clients are set to use JIVA as their screensaver. If you are using screen saver clients make sure the computers they are installed on are set to use JIVA as their screensaver. This is usually set in a system screen saver control panel.