

JIVA Administrator's Guide

Brain Murmurs, Incorporated

Document History

Version	Date	Comments
1.0	06/08/06	Initial Release

Document History	1
Definitions.....	2
Introduction	3
JIVA Server	3
JIVA Client.....	3
The JIVA Server.....	4
Client-Server Transactions	4
Client Kernel Updates.....	4
Installing an updated Kernel Image.....	4
Application Task Updates	4
Installing a new Application Task	4
Application Task Data	5
Logging	5
Log File Location	5
Log Configuration File.....	5
The JIVA Client.....	6
JIVA Client Internals.....	6
JIVA Client Boot loader.....	6
JIVA Client Boot Process	6
“startClient” script	7
JIVA Client Kernel.....	7
JIVA Kernel Updates	7
JIVA Client Task	8
Configuration Editor.....	8
Configuration Editor Parameters	8
Configuration Editor Examples	8

Definitions

Term	Description
JIVA	Java Integrated Virtual Machine Array – a Grid Network System that uses Java for platform independence and security.
JIVA Client	A worker in the JIVA Grid Network, typically a screen saver or an idle process that runs any time the computer is not in use.
JIVA Server	An application used to assign work to JIVA Clients. Also used to provide code updates to the Clients on an as-needed basis
JIVA Task	An application plugin executed by the JIVA Clients in a massively parallel fashion. The JIVA Task actually determines what the JIVA Grid Network does (protein folding, ray tracing, etc).
JIVA Task Data	Input data specific to the current JIVA Task (model and camera position for ray tracing, input parameters for protein folding, etc).
JIVA SDK	A Software Development Kit used to create new tasks to run on the JIVA Grid Network
Code Signing	A process by which code is digitally signed. This process allows the JIVA Client to verify the origins of the code it runs.
Java	A programming language
Java bytecode	A binary format into which programs written in the Java programming language can be compiled. Can be executed on most computers for which a Java Virtual Machine exists.
JVM	Java Virtual Machine. An engine capable of loading and executing applications compiled into Java bytecode.
Java Security Manager	Phenomenally restrictive and draconian subsystem of the JVM used to provide run-time checking on applications. Usually deactivated for most standard Java applications as it prohibits access to the file system, creation of class loaders, access to almost all network functions. Applications attempting to perform invalid actions are terminated by the Java Security Manager before the offending instruction can be executed.
JIVA Security Policy	Document passed to the Java Security Manager at Client boot. It grants a severely limited set of privileges to the JIVA Client, i.e., the ability to make a network connection to the JIVA Server.

Introduction

A basic JIVA Grid Network consists of a JIVA Server, one or more JIVA Clients, and an optional FlexNET Server for managing JMSL floating licenses.

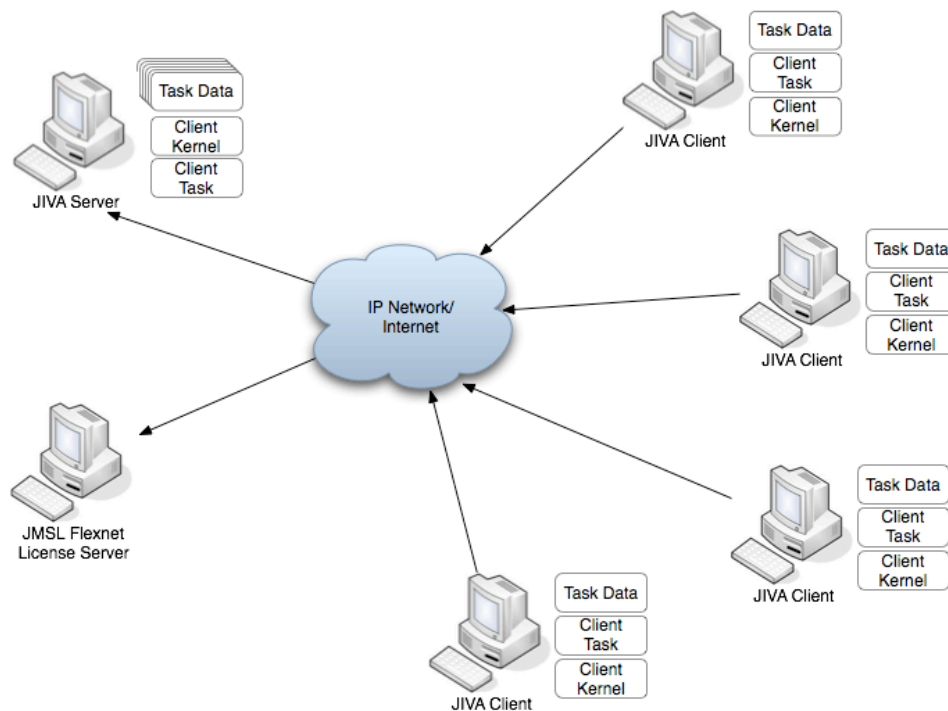


Figure 1 - Simple JIVA Grid Network

JIVA Server

The JIVA Server dispatches work to clients and collects solutions back from them. It also maintains the current versions of the JIVA Client Kernel and the JIVA application task library.

JIVA Client

The JIVA Clients are the workhorses of the network, executing the JIVA application task library whenever their individual host computers are idle.

Each client is configured with the IP address of the JIVA Server and, optionally, the FlexNET Server. Clients receive their application task code and data from the server, as well as periodic updates of the JIVA Client Kernel. If the current application task makes use of the JMSL libraries it will attempt to check out a JMSL license from the FlexNET license server. Otherwise the FlexNET Server IP address will be ignored.

The JIVA Server

The JIVA Server works as the dispatcher in the JIVA Grid Network: it tracks what application the JIVA Clients are running and dispatches new inputs for the Clients to process as they become available to do more work. The JIVA Server runs as a Java Servlet and can technically run with any compliant Servlet container. Brain Murmurs currently only supports the Apache Tomcat at the present time.

Client-Server Transactions

All transactions between the client and the server are initiated by the client. The client does not listen for incoming messages from the server, nor does the server ever attempt to open a connect to the client. Any time the client needs to send a message to the server it issues a single HTTP POST message to the server. The server can then use this connection opened by the client to issue a single response back to it.

Client Kernel Updates

The JIVA Server controls what version of the Kernel the JIVA Clients use. If the core JIVA Client code base is updated, a new kernel image can be placed on the server. If a Client connects to the server with an out dated kernel version the Server will order it to upgrade to the new version.

Installing an updated Kernel Image

To install an updated kernel image, simply copy the new jiva-client.jar file to tomcat/webapps/jiva/client. The JIVA Server will automatically update clients and they call in for more work.

Application Task Updates

The JIVA Server also controls application task version the JIVA Clients use. If the application task is updated, or if a different application is to be run, the updated task bundle can be placed on the server. If a Client connects to the server with an invalid task id or version the Server will order it to install the new application task immediately.

Installing a new Application Task

New application task bundles can simply be copied to tomcat/webapps/jiva and unzipped. The entire directory structure for the new task will be overlaid on top of the server file system.

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

Note: version 1.3 of the JIVA Server will feature a web interface to simply this process.

Application Task Data

The primary job of the JIVA Server is to assign task data to JIVA Clients. This is done whenever a JIVA Client send an IDLE message the server, provided that the client's reported kernel and task code are up-to-date.

Logging

Log File Location

Logs appear in the tomcat directory under the "logs" directory. The JIVA Server log file is called jiva-server.log

Log Configuration File

The JIVA Server uses Log4J for its logging. The Log4J properties file is located at tomcat/webapps/jiva/WEB-INF/classes/log4j-jiva.properties. It is suggested that most of the entries be left as-is unless you are knowledgeable in the ways of Log4J.

Parameters of interest:

- log4j.rootLogger=DEBUG, stdout, R
The "DEBUG" indicates the current logging level. The log level determines the verbosity (and therefore the size) of the log files. Acceptable values in order of increasing verbosity are FATAL, ERROR, WARN, INFO, DEBUG.
- log4j.appender.R.MaxFileSize=1000KB
This is the maximum allowable size of the log file. Once this value has been exceeded the log system will either roll the log and start a new one or wrap around and start overwriting the existing file.
- log4j.appender.R.MaxBackupIndex=20
Maximum number of archived logs to maintain. Once this number has been exceeded the old files will start to get overwritten.

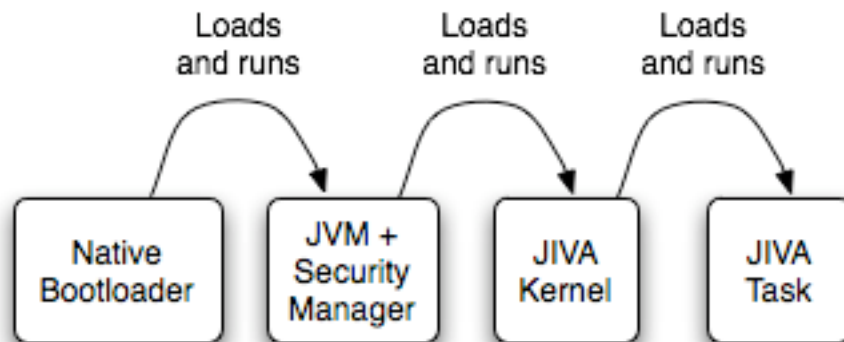
The JIVA Client

The JIVA Client is a generic processing engine available as either a screen saver or a daemon (UNIX process) for a wide range of platforms.

JIVA Client Internals

The client consists of four major components:

- Boot loader
- Java Virtual Machine (JVM)
- JIVA Kernel
- JIVA Task



JIVA Client Boot loader

The JIVA Client Boot loader is typically a screen saver or daemon written in native code (i.e., C, C++, or Objective-C) for a specific computer type and operating system. The primary purpose of the boot loader is to detect when a computer is no longer in use and start up the JIVA Client.

The secondary purpose of the screen saver version is to render graphics created by the current JIVA Task.

JIVA Client Boot Process

The Client boot process consists of the following steps

1. Launch an embedded JVM with an activate Java Security Policy Manager and the JIVA Security Policy
2. Initialize the JIVA Client Kernel
3. Initialize the JIVA Client Kernel Graphics Manager (Screen Saver only)
4. Load and execute the JIVA Client Kernel in the newly loaded JVM

It is worth noting that the JIVA Kernel can only read the JIVA Security Policy. This means that it is not technically feasible for code downloaded and run by the JIVA Client to alter the Client's security settings in any way.

"startClient" script

There is also a command line version of the boot loader for testing and diagnostic purposes. The file is called startClient.bat under Windows and startClient.sh under UNIX operating systems. Both files are simply shell scripts that launch the kernel along with the security manager from the command line.

These scripts are very useful for debugging your network setup because they allow you to see everything written to standard output, including any exceptions that may have occurred.

JIVA Client Kernel

The JIVA Client Kernel is called jiva-client.jar. The kernel is really the heart of the JIVA Client. This contains all of the code used by the client (network code to reach the server, code to load and execute tasks, draw graphics, etc) once it has been booted and launched by the boot loader.

JIVA Kernel Updates

The JIVA Server maintains a copy of the latest version of the JIVA Client Kernel. If the Server gets a new update it will let the Client know the next time it calls in.

In order to use the new kernel image the client must restart itself. In the case of the screen savers this can be quite surprising to watch happen; a message saying "JIVA is downloading a new kernel" appears, followed by a message saying "JIVA will restart in 5 seconds" followed by an abrupt halt of the screen saver.

The screen saver will eventually restart again. On Windows and OS X, the regular inactivity timer restarts and the screen saver will resume activity after the timer expires again. Xscreensaver (the Linux screen saver framework) actually uses two timers. One determines how long a period of activity should pass before the screen saver turns on. The other timer is used to restart the entire Xscreensaver framework.

On the Linux platform the later, "cycle" time value must pass before JIVA will restart itself.

JIVA Client Task

Updated JIVA Client Tasks are automatically downloaded from the JIVA Server shortly after startup. The task is updated anytime a new update is installed and configured under the JIVA Server's task directory.

Configuration Editor

The configuration editor is an application stored inside the Jiva Kernel archive (jiva-client.jar) and is designed to allow administrators to easily and rapidly customize and entire network easily after initial installation. To reconfigure a client you must log onto the workstation the client is running on and execute the following command in the Client's working directory.

```
java -classpath jiva-client.jar com.brainmurmurs.jiva.client.config.ConfigEditor
```

(This command will be wrapped into a slightly less verbose command in the future).

The configuration editor is used change the JIVA Client's configuration. Currently the configuration allows the following settings to be edited:

- JIVA Server IP Address
- JIVA Client IP Address

Configuration Editor Parameters

-h	run headless (suppress dialog box)
-w	write files (totally rewrite all configuration files after accepting parameters)
-j <i>hostname</i>	new jiva server hostname or ip address
-l <i>hostname</i>	new jmsl FlexNET server hostname or ip address

Configuration Editor Examples

```
java -classpath jiva-client.jar com.brainmurmurs.jiva.client.config.ConfigEditor
```

Allows the administrator to view the current settings in a GUI without making any changes.

```
java -classpath jiva-client.jar com.brainmurmurs.jiva.client.config.ConfigEditor -w
```

Allows the administrator to edit the configuration settings from the GUI and completely rewrite the existing configuration files

```
java -classpath jiva-client.jar com.brainmurmurs.jiva.client.config.ConfigEditor -j
douglas -l adams -w -h
```

Instructs the config editor to completely rewrite the client's configuration files in command line mode (no GUI) using "douglas" as

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

the hostname for the JIVA Server and "adams" as the hostname for the JMSL libraries.