JFreeMon

User Guide

13.4.2007
Tuomas Tikka

Contents

1. Overview ... 3

 Purpose of the Document ... 3

 Intended Audience ... 3

 Used Terms ... 3

 Version History ... 4

2. Client Systems ... 5

 Adding Client Systems ... 5

 Removing Client Systems ... 7

 Adding Meters ... 7

 Removing Meters ... 8

 Configuring Alerts ... 8

 Viewing Client Data Charts ... 9

3. Preferences ... 10

 Deskop Appearance ... 10

 Logging ... 12

 Connections ... 13

 Database ... 14

4. Server ... 16

 Starting/Stopping ... 16

 Pausing/Resuming ... 16

 Viewing Traffic Summary ... 17

 Server Log ... 18

5. Help ... 19

 Contents ... 19

 System Info ... 19

 Software Update ... 20

 Viewing the Software License ... 20

 About JFreeMon ... 21

Appendix A – Automated Scripts ... 22

1. Overview

This document is a user guide to JFreeMon, a general purpose resource monitoring system.

Purpose of the Document

The purpose of this document is to provide the information necessary to take advantage of all the functionality in JFreeMon. This includes:

- Managing client systems
- Managing individual client meters
- Configuring alert levels and automated tasks for client meters
- Providing reports of total traffic
- Providing reports of client data
- Managing the packet listener (server)

Intended Audience

The intended audience of this document is personnel involved in day-to-day use of JFreeMon.

Used Terms

The table below explains the terms used in this document.

Term	Explanation
Alert	An event that is triggered when a incoming data reaches a predefined level. Alerts can be warnings or errors.
Automated Script	A script (currently JavaScript) file that is evaluated automatically if an alert is triggered. Separate scripts can be defined for warnings and errors.
Client System	A 3rd party system that integrates to JFreeMon by sending resource data in packets to the JFreeMon Server.
JFreeMon	Open source general purpose resource monitoring software.
Meter	A way of categorizing data, for example a meter can identify data as CPU usage, Disk space usage, etc.

Version History

The table below shows the version history of this document.

Version	Created	Description
0.1	13.4.2007	First version of document.

2. Client Systems

This chapter describes functions related to managing client systems in JFreeMon. Client systems are the source of monitored data – these clients connect to the JFreeMon Server to send resource data. This chapter describes the following functions:

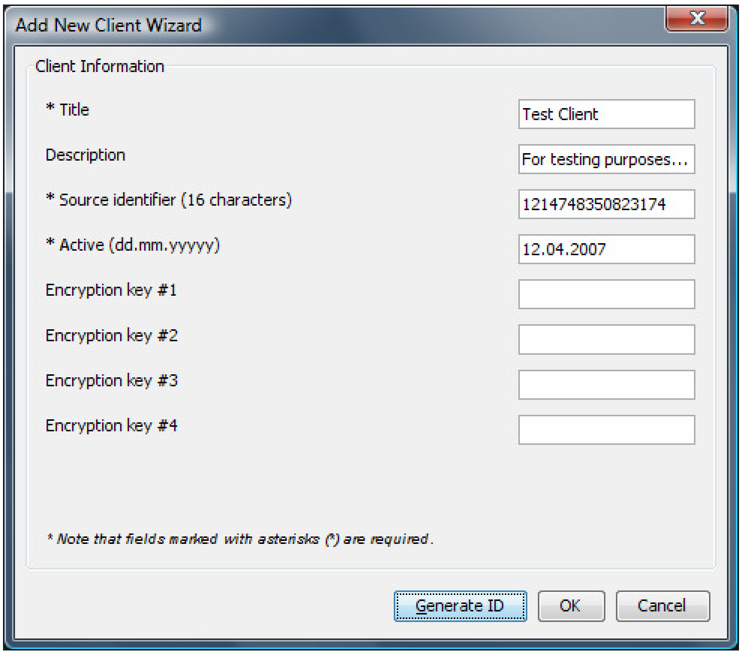
- Adding a new client system
- Removing client systems
- Adding meters to clients
- Removing meters from clients
- Configuring alert levels for clients' meters
- Viewing client data charts

Adding Client Systems

Before a client system can be monitored, its basic information must be set up in JFreeMon. This can be done by selecting **New Client Wizard** from the **File** menu. The following information can be provided:

Parameter	**Description**
Title	A free-form display name for the client system.
Description	A brief free-form description of the client system.
Source identifier	The unique identifier by which client system is identified in JFreeMon. This information must also be present in client systems when sending data.
Active date	A date after which the client system becomes active. Prior to this date, all data for the client system is ignored.
Encryption keys	Not implemented in JFreeMon 0.1.

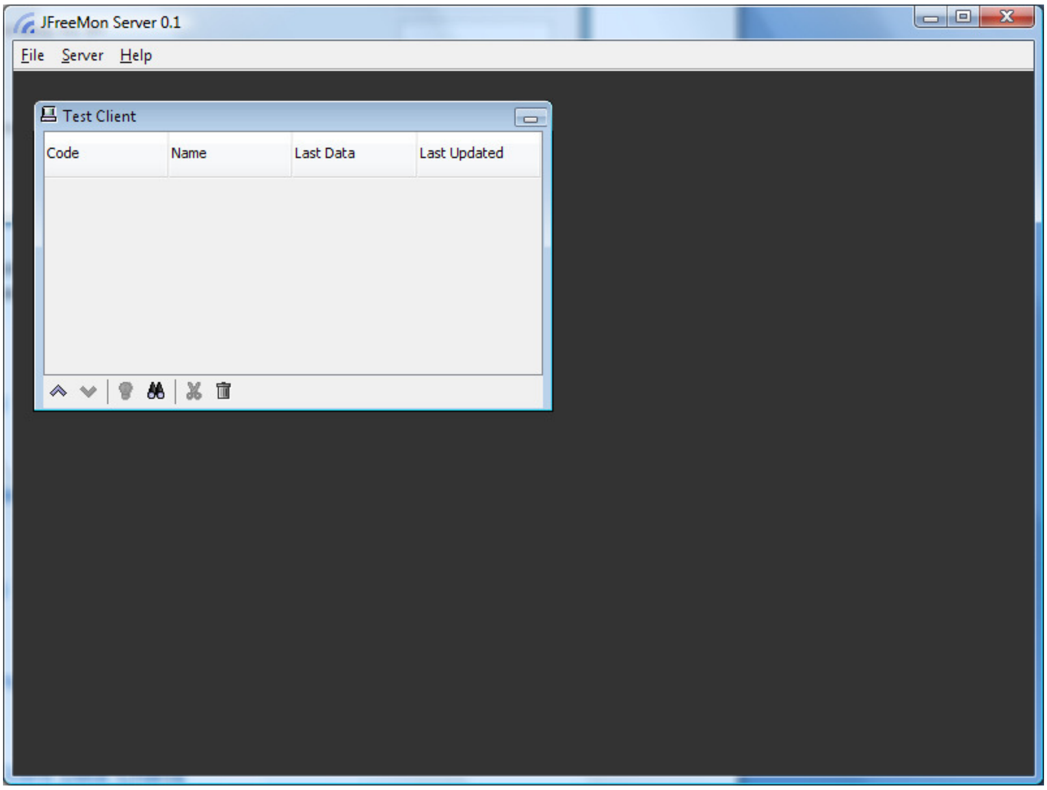
The screenshot below shows the New Client Wizard in JFreeMon:



* Title	Test Client
Description	For testing purposes...
* Source identifier (16 characters)	1214748350823174
* Active (dd.mm.yyyyy)	12.04.2007
Encryption key #1	
Encryption key #2	
Encryption key #3	
Encryption key #4	

* Note that fields marked with asterisks (*) are required.

Once the client is saved by clicking **OK**, a window will appear onto the JFreeMon desktop which represents the client. A screenshot of the newly created client is shown below:



Code	Name	Last Data	Last Updated

Removing Client Systems

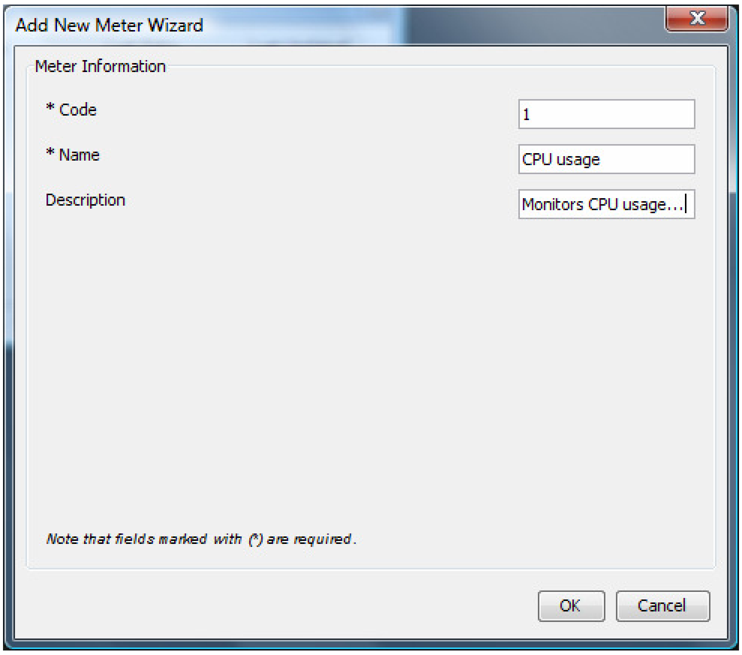
To remove a client system from JFreeMon, click on the “trashcan” button on the client system window’s toolbar.

Adding Meters

Before the client system’s data can be monitored, meters must be created to represent the type of data that the client is sending. Meters can represent any kind of resource, for example CPU usage, warehouse stock, etc. To add a new meter to a client system, click on the “up arrow” button on the client window’s toolbar. The following information can be provided:

Parameter	**Description**
Code	A number code that identifies the meter inside the client. This information must also be present in client systems when sending data.
Name	A free-form name for the meter, for example “CPU usage”.
Description	A free-form brief description of the meter.

The screenshot below shows the New Meter Wizard in JFreeMon:



Add New Meter Wizard

Meter Information

* Code: 1

* Name: CPU usage

Description: Monitors CPU usage...

Note that fields marked with (*) are required.

Once the meter is created, a row representing the meter will be added into the client system's window. This is where the latest data is also shown.

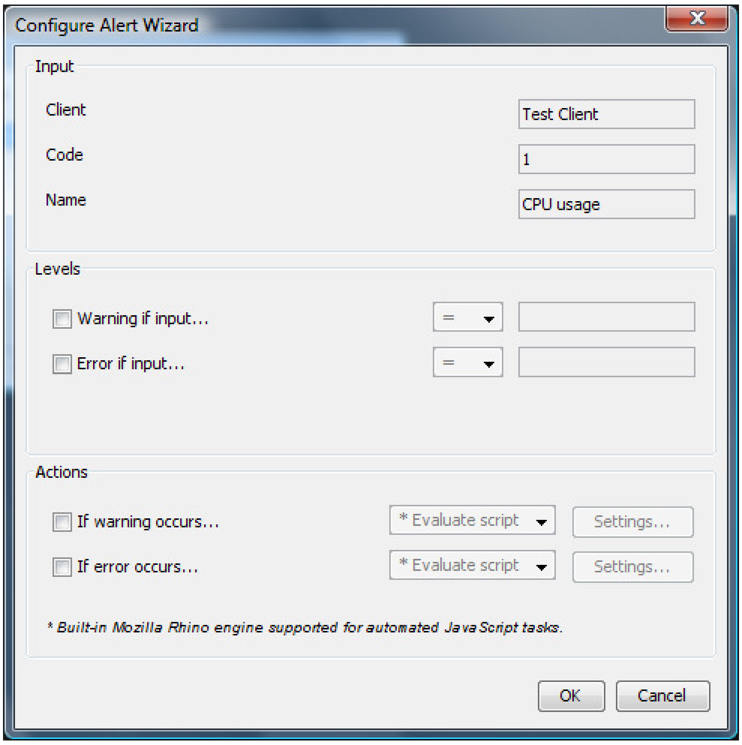
Removing Meters

To remove a meter from the client system, click on the "down arrow" button in the client system window's toolbar after selecting the meter from the meter list in the client system's window.

Configuring Alerts

When monitoring data sent from client systems, it is sometimes necessary to identify certain levels for the data. In JFreeMon, these levels are warnings and errors. If these levels are configured for a certain meter, data reaching these levels will trigger warning and/or error flags for the meter. These flags are represented by coloring the meter in question in the client system window's meter list. For example, by default warnings are marked yellow and errors red. To configure alert levels for a meter, click on the "light bulb" button on the client system window's toolbar.

The screenshot below shows the Configure Alert Wizard in JFreeMon:


Configure Alert Wizard

Input

Client: Test Client

Code: 1

Name: CPU usage

Levels

□ Warning if input... =

□ Error if input... =

Actions

□ If warning occurs... * Evaluate script

□ If error occurs... * Evaluate script

* Built-in Mozilla Rhino engine supported for automated JavaScript tasks.

The **Input** section shows basic read-only information, like the client system in question as well as the meter's code and name.

The **Levels** section is used to define the warning and error levels for the meter. These can be enabled by selecting the checkbox "If warning occurs..." or "If error occurs..." respectively. Once this is done, the operator and level can be entered. The operator is used when comparing incoming data with the level entered. For example, if the operator is “>” and the warning level “10”, then any data over “10” will trigger the warning.

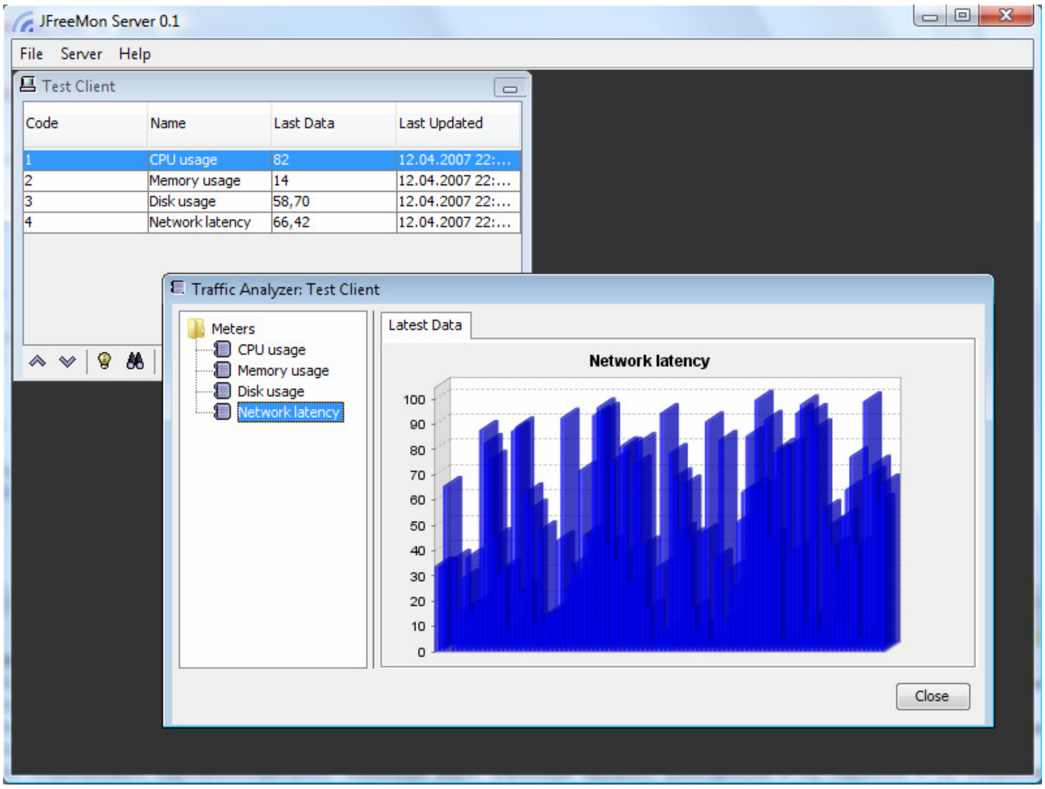
The **Actions** section is used to set up automated actions if warning or error levels are reached. Currently JFreeMon supports JavaScript actions which are evaluated by the JVM’s built-in Mozilla Rhino script engine. If you want to run a certain script each time a warning or error level is reached, enable the “If warning occurs…” or “If error occurs…” respectively, make sure the “Evaluate script” operation is selected and choose the script file by pressing the **Settings** button. Using scripts provides a powerful mechanism to dynamically extend the capabilities of JFreeMon, for example showing dialogs, sending email, 3rd party logging, etc. – basically anything that can be written as a script. See *Appendix A – Automated Scripts* for details.

Note! The Java Runtime Environment (JRE) version 1.6 is required for scripting support.

Viewing Client Data Charts

To view a real-time graphical representation of a client system’s latest data, click on the “binoculars” button on the client system window’s toolbar. This starts the client data traffic analyzer, which shows the latest 100 packets per client meter.

The screenshot below shows the client data history for an imaginary “Network latency” meter:



Code	Name	Last Data	Last Updated
1	CPU usage	82	12.04.2007 22:...
2	Memory usage	14	12.04.2007 22:...
3	Disk usage	58,70	12.04.2007 22:...
4	Network latency	66,42	12.04.2007 22:...

3. Preferences

This chapter describes the configuration parameters in JFreeMon. These can be divided into the following areas:

- Desktop appearance related settings
- Server related settings
- Logging related settings
- Connection related settings
- Database related settings

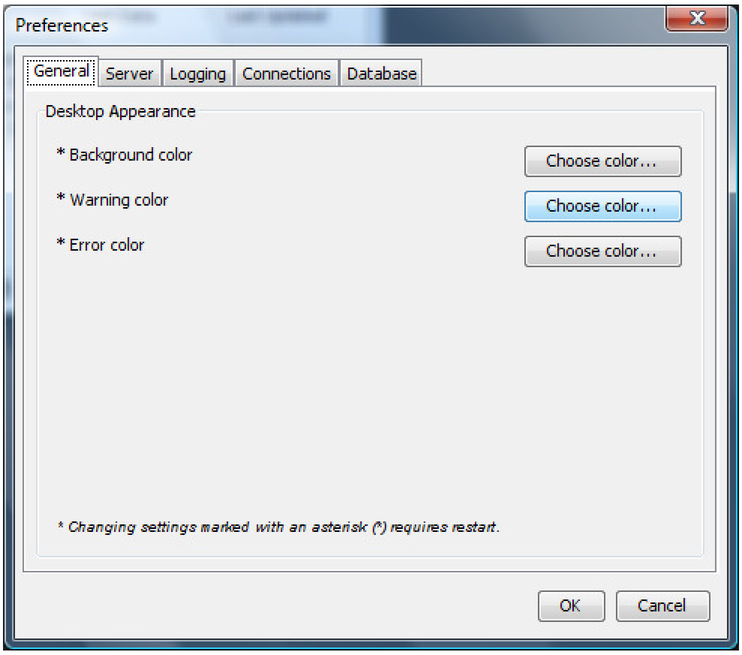
Deskop Appearance

The following desktop appearance related settings can be adjusted in JFreeMon.

Setting	Description
Background color	The desktop background solid color.
Warning color	The background color of a meter in the client system view that has received a warning.
Error color	The background color of a meter in the client system view that has received an error.

Note! The application must be restarted for the changes to take effect.

The screenshot below shows the desktop appearance related settings in JFreeMon:



Desktop Appearance

* Background color [Choose color...]

* Warning color [Choose color...]

* Error color [Choose color...]

** Changing settings marked with an asterisk (*) requires restart.*

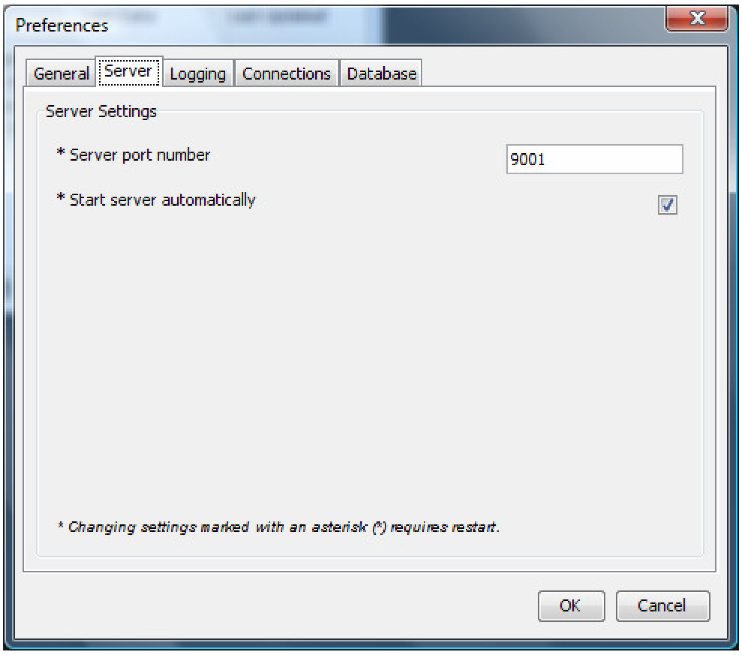
Server

The following server related settings can be adjusted in JFreeMon:

Setting	Description
Port number	The port number on which the server listens for incoming packets (must be in the range 1-65535).
Start automatically	Option to start listening for packets automatically once the application is started (otherwise must be manually started by selecting **Start** from the **Server** menu).

Note! The application must be restarted for the changes to take effect.

The screenshot below shows the server related settings in JFreeMon:



Server Settings

* Server port number 9001

* Start server automatically ☑

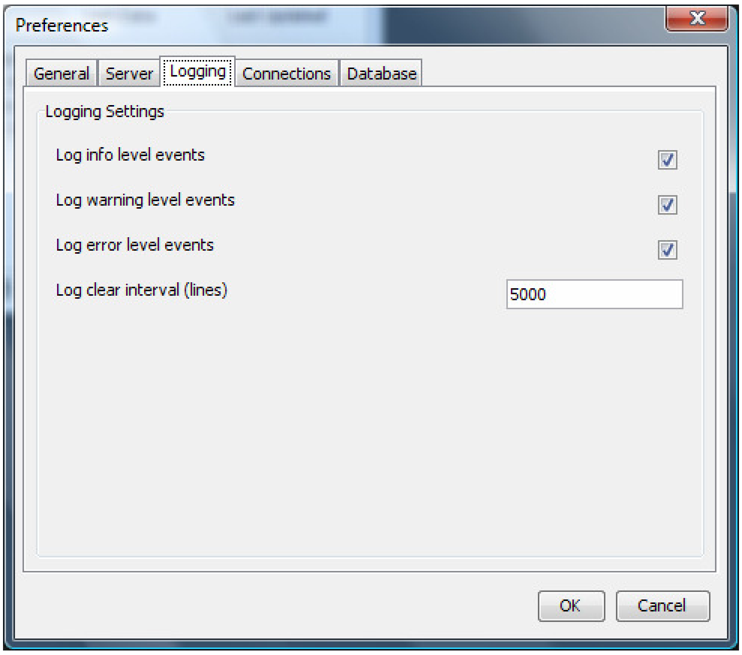
* Changing settings marked with an asterisk (*) requires restart.

Logging

The following logging related settings can be adjusted in JFreeMon:

Setting	Description
Log INFOs	Option to enable/disable logging of INFO level system events.
Log WARNINGs	Option to enable/disable logging of WARNING level system events.
Log ERRORs	Option to enable/disable logging of ERROR level system events.
Log clear interval	Number of log lines after which the server log is cleared.

The screenshot below shows the logging related settings in JFreeMon:



Logging Settings

Log info level events ☑

Log warning level events ☑

Log error level events ☑

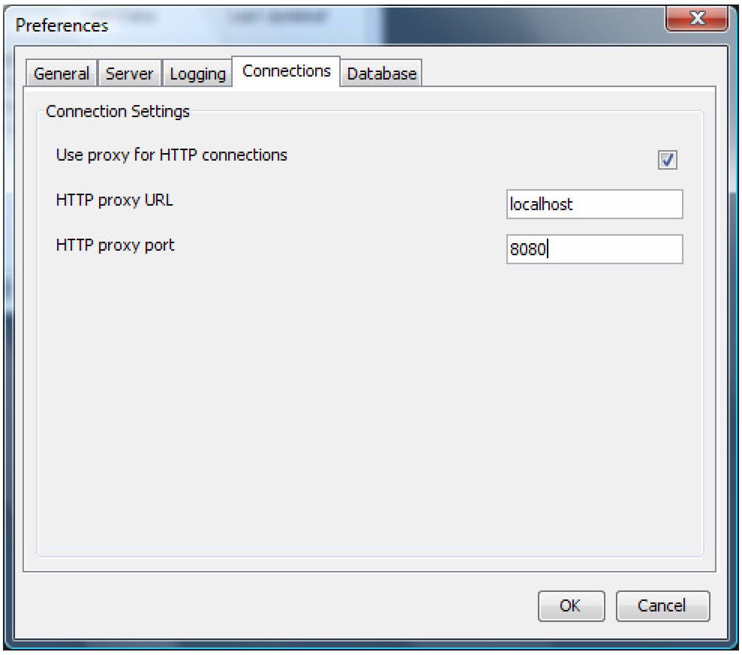
Log clear interval (lines) 5000

Connections

The following connection related settings can be adjusted in JFreeMon:

Setting	Description
Use HTTP proxy	Option to enable/disable the use of a proxy for HTTP connections.
HTTP proxy URL	The proxy URL (for example “host.domain”).
HTTP proxy port	The proxy port (must be in the range 1-65535).

The screenshot below shows the connection related settings in JFreeMon:



Use proxy for HTTP connections ☑

HTTP proxy URL: localhost

HTTP proxy port: 8080

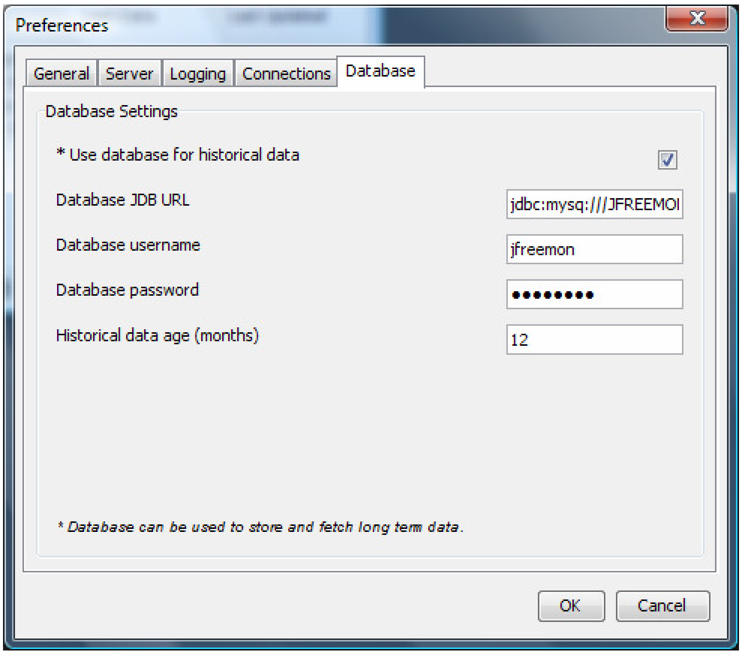
Database

The following database related settings can be adjusted in JFreeMon:

Setting	Description
Use database	Option to enable/disable database use for storing historical data from client systems.
JDBC URL	The database JDBC connection URL.
Database username	The database user account username.
Database password	The database user account password.
Historical data age	The age in months to store historical data (for example 12).

Note! Using the database is optional. If used, be sure to include the appropriate JDBC driver in the application startup scripts.

The screenshot below shows the database related settings in JFreeMon:



Database Settings

* Use database for historical data	☑
Database JDB URL	jdbc:mysq:///JFREEMOI
Database username	jfreemon
Database password	••••••••
Historical data age (months)	12

** Database can be used to store and fetch long tem data.*

4. Server

This chapter describes the server related functions in JFreeMon. It includes the following topics:

- Starting and stopping the server
- Pausing and resuming the server
- Viewing traffic summary
- Viewing the server log

Starting/Stopping

The packet listener (server) can be started and stopped at any time. To start the server, select **Start** from the **Server** menu. Once the server has started, the following system event will be displayed:

```
<INFO> <01.04.2007 21:01:07> <UDPListerner> <listen> <Start
requested with port 9001>
<INFO> <01.04.2007 21:01:07> <Worker> <run> <Listening for
datagram packets on port 9001>
```

To stop the server, select **Stop** from the **Server** menu. Once the server has stopped, the following system event will be displayed:

```
<INFO> <01.04.2007 21:06:01> <TCPListerner> <end> <End
requested>
<INFO> <01.04.2007 21:06:01> <Worker> <run> <Ended listening
for datagram packets.>
```

Pausing/Resuming

The packet listener (server) can be paused and resumed at any time. To pause the server, select **Pause** from the **Server** menu. Once the server has paused, the following system event will be displayed:

```
<INFO> <01.04.2007 21:06:35> <TCPListerner> <pause> <Pause
requested>
```

To resume the server, select **Resume** from the **Server** menu. Once the server has resumed, the following system event will be displayed:

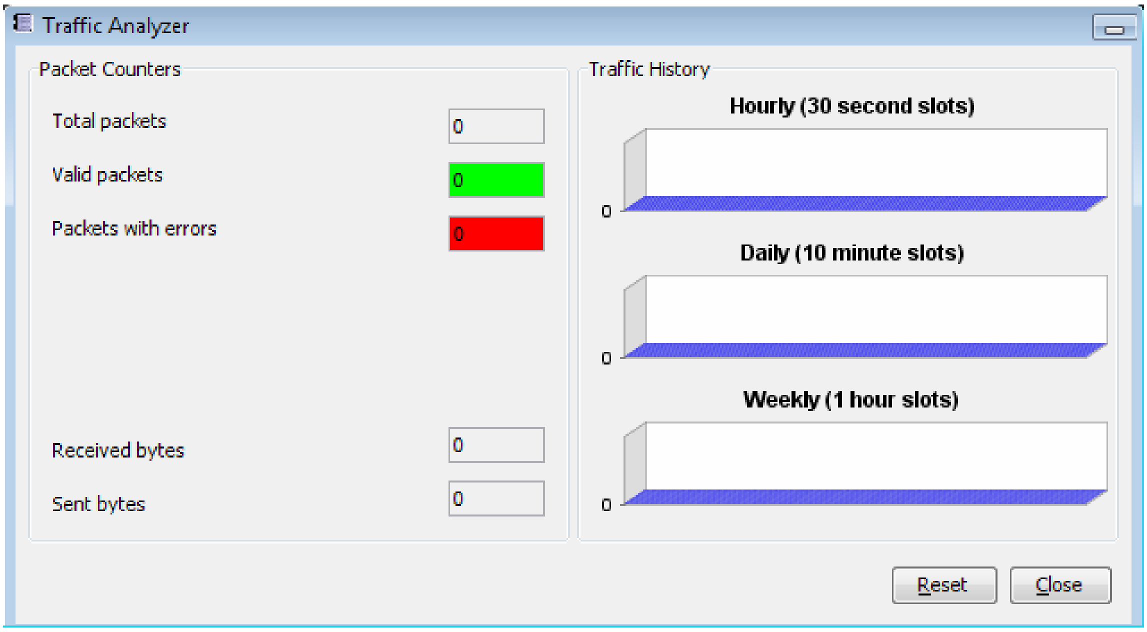
```
<INFO> <01.04.2007 21:07:08> <TCPListerner> <unpause> <Resume
requested>
```

Viewing Traffic Summary

The traffic summary window displays information on the number of packets going through the system. The following information is displayed in textual form:

Field	Description
Total Packets	The total number of packets that went through the system.
Valid Packets	Number of packets that were parsed successfully.
Packets with errors	Number of packets that were erroneous in format.
Received bytes	The total number of bytes received.
Sent bytes	The total number of bytes sent.

Additionally, traffic history is shown graphically as charts. These three charts monitor the total number of latest packets. The hourly chart shows packets within the latest hour, the daily chart shows packets within one day and the weekly chart shows packets within one week. The screenshot below shows the traffic analyzer window in JFreeMon:



Total packets 0

Valid packets 0

Packets with errors 0

Received bytes 0

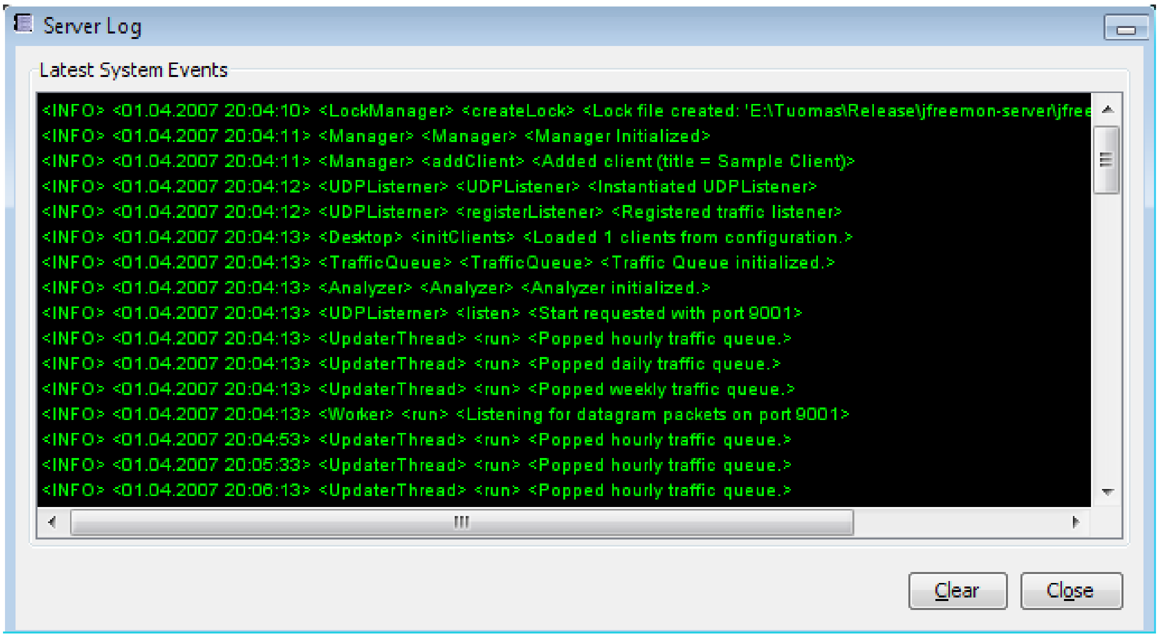
Sent bytes 0

Server Log

The server log window displays system events in a chronological order (latest event at the bottom). System events are categorized as follows:

Level	Description
INFO	Nice-to-know information (for example server startup notices)
WARNING	Information that may lead the system to perform unpredictably (for example invalid packets sent by client systems)
ERROR	Application errors that may lead to system crashes (for example memory full, database not available, etc.)

To access the server log window, select **Server Log** from the **Server** menu. The screenshot below shows the server log window in JFreeMon:



```
<INFO> <01.04.2007 20:04:10> <LockManager> <createLock> <Lock file created: 'E:\Tuomas\Release\jfreemon-server\jfree
<INFO> <01.04.2007 20:04:11> <Manager> <Manager> <Manager Initialized>
<INFO> <01.04.2007 20:04:11> <Manager> <addClient> <Added client (title = Sample Client)>
<INFO> <01.04.2007 20:04:12> <UDPListerner> <UDPListener> <Instantiated UDPListener>
<INFO> <01.04.2007 20:04:12> <UDPListerner> <registerListener> <Registered traffic listener>
<INFO> <01.04.2007 20:04:13> <Desktop> <initClients> <Loaded 1 clients from configuration.>
<INFO> <01.04.2007 20:04:13> <TrafficQueue> <TrafficQueue> <Traffic Queue initialized.>
<INFO> <01.04.2007 20:04:13> <Analyzer> <Analyzer> <Analyzer initialized.>
<INFO> <01.04.2007 20:04:13> <UDPListerner> <listen> <Start requested with port 9001>
<INFO> <01.04.2007 20:04:13> <UpdaterThread> <run> <Popped hourly traffic queue.>
<INFO> <01.04.2007 20:04:13> <UpdaterThread> <run> <Popped daily traffic queue.>
<INFO> <01.04.2007 20:04:13> <UpdaterThread> <run> <Popped weekly traffic queue.>
<INFO> <01.04.2007 20:04:13> <Worker> <run> <Listening for datagram packets on port 9001>
<INFO> <01.04.2007 20:04:53> <UpdaterThread> <run> <Popped hourly traffic queue.>
<INFO> <01.04.2007 20:05:33> <UpdaterThread> <run> <Popped hourly traffic queue.>
<INFO> <01.04.2007 20:06:13> <UpdaterThread> <run> <Popped hourly traffic queue.>
```

5. Help

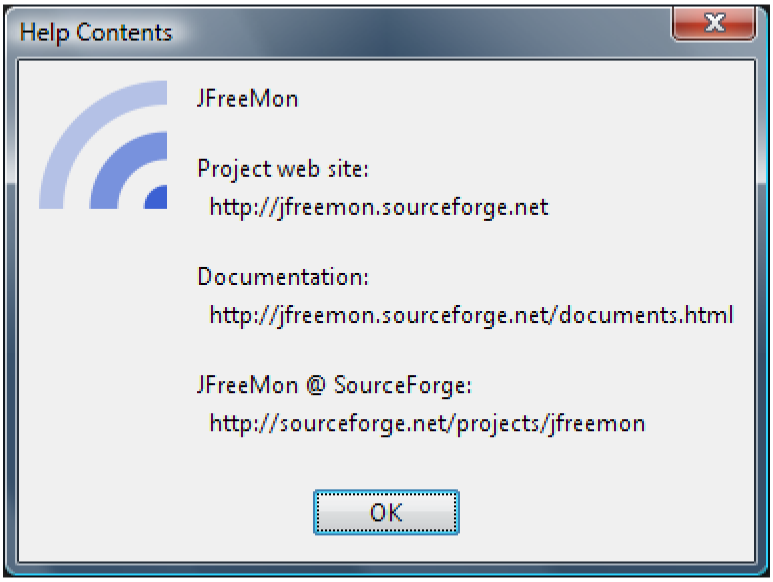
This chapter describes the help related functions in JFreeMon. It includes the following topics:

- Help Contents
- Viewing System Info
- Software Update
- Viewing the Software License
- About JFreeMon

Contents

The help contents in JFreeMon 0.1 is only a dialog which provides links to further documentation from the JFreeMon website. Future releases of JFreeMon will include a fully functional HTML based help window.

To access the help contents dialog, select **Help Contents** from the **Help** menu. The screenshot below shows the help contents in JFreeMon 0.1.



JFreeMon

Project web site:
http://jfreemon.sourceforge.net

Documentation:
http://jfreemon.sourceforge.net/documents.html

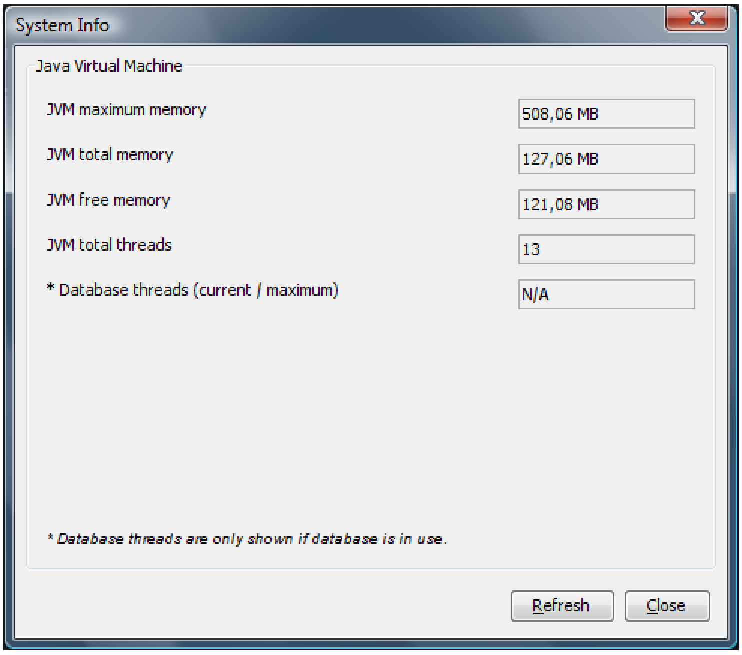
JFreeMon @ SourceForge:
http://sourceforge.net/projects/jfreemon

System Info

The system info dialog in JFreeMon is a tool displaying nice-to-know information about system resources. The following information is included:

- The maximum amount of memory available to the JVM (Java Virtual Machine)
- The total amount of memory used by the JVM
- The amount of free memory available to the JVM
- The number of threads used in the JVM
- The number of threads currently used for database operations

To access the system info dialog, select **System Info** from the **Help** menu. The screenshot below shows the system info dialog in JFreeMon 0.1.



System Info

Java Virtual Machine

JVM maximum memory: 508,06 MB

JVM total memory: 127,06 MB

JVM free memory: 121,08 MB

JVM total threads: 13

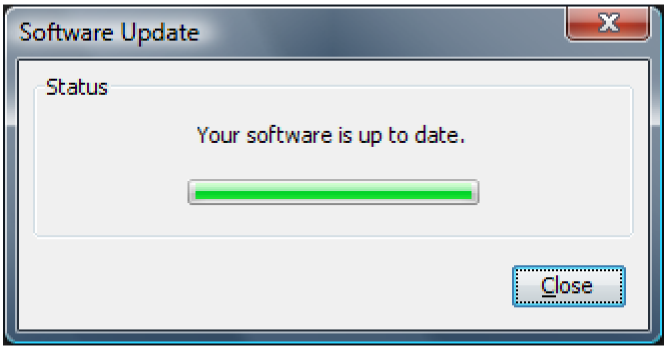
* Database threads (current / maximum): N/A

** Database threads are only shown if database is in use.*

Software Update

The software update function in JFreeMon can be used to check if a newer version of JFreeMon exists. This information is requested via HTTP from the JFreeMon project website. Note that in JFreeMon 0.1 this function is only limited to detecting newer versions, no automatic upgrade is available. If a newer version exists, it must be manually downloaded and installed.

To access the software update, select **Software Update** from the **Help** menu. The screenshot below shows the software update when no updates were found:



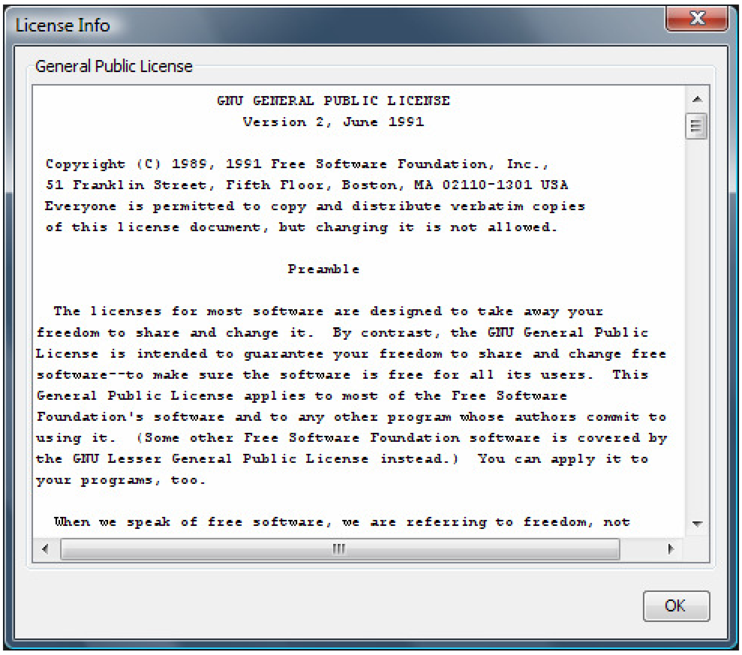
Software Update

Status

Your software is up to date.

Viewing the Software License

JFreeMon is distributed under the GPL (General Public License) license. To view the terms and conditions of this license, select **View License** from the **Help** menu. The screenshot below shows the license dialog in JFreeMon.

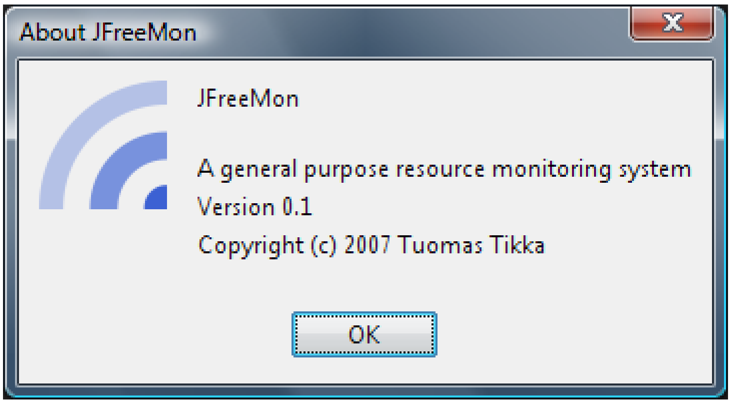


```
                    GNU GENERAL PUBLIC LICENSE
                       Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Lesser General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
```

Note! The GPL license should always be found in the LICENSE text file in the project installation package.

About JFreeMon

The about dialog in JFreeMon contains the version number of the software in use as well as the copyright notice. To access the about dialog, select About JFreeMon from the Help menu. The screenshot below shows the about dialog in JFreeMon:



JFreeMon

A general purpose resource monitoring system
Version 0.1
Copyright (c) 2007 Tuomas Tikka

Appendix A – Automated Scripts

This chapter provides basic information for setting up scripts, which are run if warning or error levels are reached. Currently JFreeMon supports JavaScript script files, which are evaluated using the JVM’s built-in Mozilla Rhino engine.

Certain parameters are passed for the evaluation, which can be used to query information on the current data packet, meter and client system. This information can be used by the scripts.

Parameter name	**Description**
`client`	The current client data type object.
`meter`	The current meter data type object.

The example script below provides a basic script, which only displays a dialog if the level is reached.

```
importPackage(javax.swing);

var optionPane = JOptionPane.showMessageDialog(null, 'Warning
occurred');
```