

QMonitor

Authors: Fengwei Dai, Hamid Nadimi

Contact: hamid.nadimi@gmail.com

July. 2010

Version 0.4

Introduction.....	2
Features.....	2
Quickstart.....	2
Monitoring Panel.....	4
Tabs.....	5
Actions.....	6
mail recipients.....	7
retry options.....	8
system commands.....	8
exclude times.....	9
Log-Panel.....	10
Cronjob-Mode.....	11
Disclaimer.....	12

Introduction

QMonitor is a graphical user interface for Monitoring q-Servers on multiple Hosts. For each Host you can specify multiple servers. The servers will be monitored for being alive (server running) and being available (server not blocked by a process).

Features

- Can send Emails when errors occur.
- Can execute System commands when errors occur.
- Can exclude Servers from monitoring for defined weekdays and times.
- Can run in a Cronjob (while it's running you can send commands to stop / resume monitoring for one or all servers)
- Requires Java 1.6

Quickstart

Unpack zipfile.

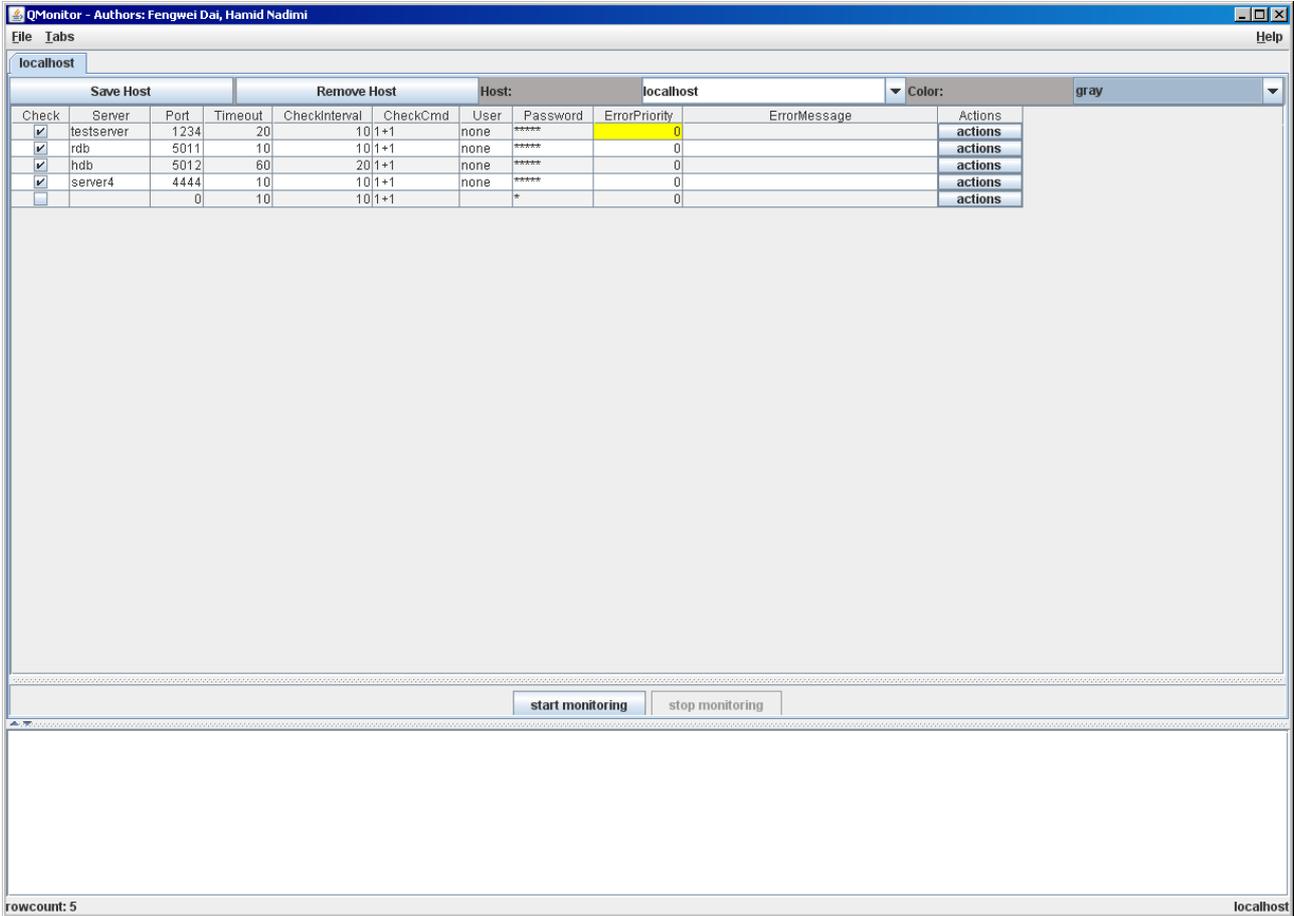
Doubleclick QMonitor.jar.

Enter host name, enter your Server parameters/ set up actions (if any) and press "Save Host" if you want to keep this configuration.

Click Start monitoring.

To stop monitoring click: Stop Monitoring

Monitoring Panel



On each row you can specify a server for the selected host.

Description of columns:

Check: if you choose not to monitor a server, uncheck here.

Server: unique name for the Server you are going to monitor.

Port: Port at which q-server is running.

Timeout: How-long to wait (in seconds) before producing an error message if server does not respond.

CheckInterval: How often to check (interval in seconds).

CheckCmd: What query to send to check if server responds.

Username: Username for login (enter "none", if no user needed).

Password: Password required for login (enter "none", if no password needed)

ErrorPriority: Displays the Priority of the ErrorMessage shown (0 if no error).

ErrorMessage: Displays the ErrorMessage (if any)

In case you have many servers, you can also edit `cfg/KXConnections.xml` directly. The xml format is simple:

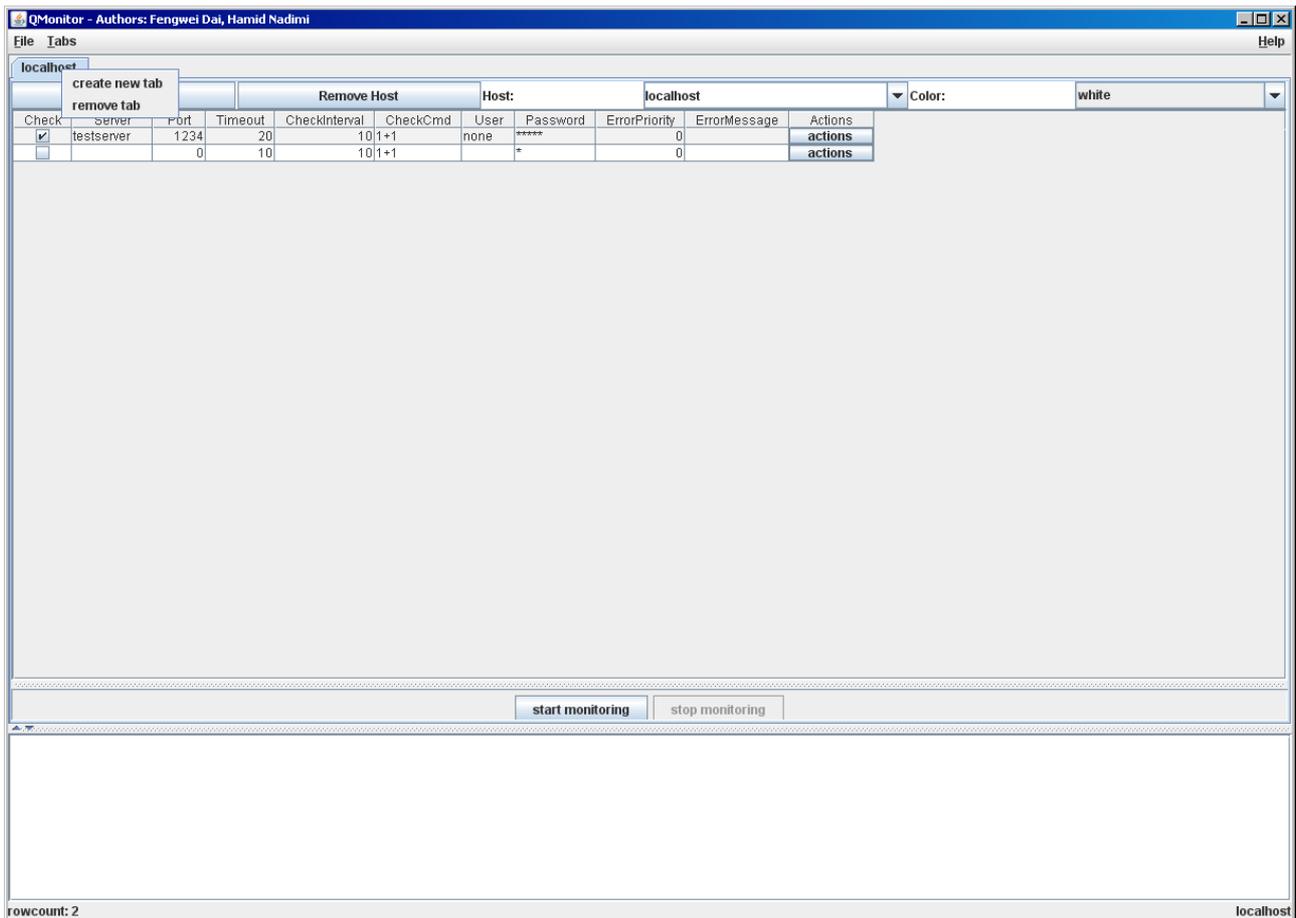
```

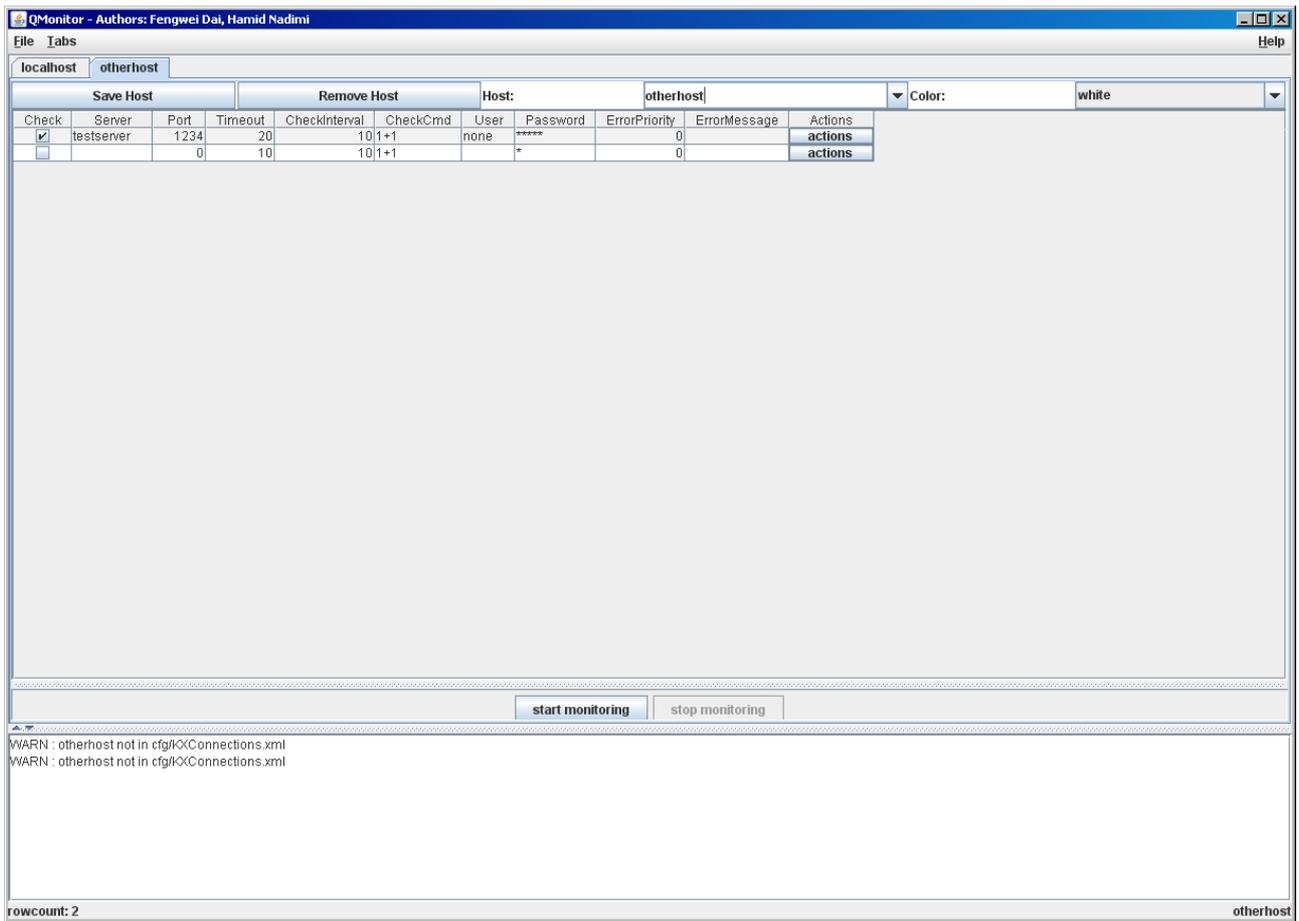
<KXConnections>
  <Host name="localhost" description="localhost" color="white">
<Server name="testserver" port="1234" timeout="20" checkInterval="10"
checkCmd="1+1">
  <User name="none" password="none" />
  <Actions>
    <Retry retryTimes="5" interval="10" />
    <SendEmail email="youremail@yourdomain" when="always" />
    <ExcludeTime startDay="ALL" endDay="ALL" startTime="2300" endTime="0155"/>
  </Actions>
</Server>
</Host>
</KXConnections>

```

Tabs

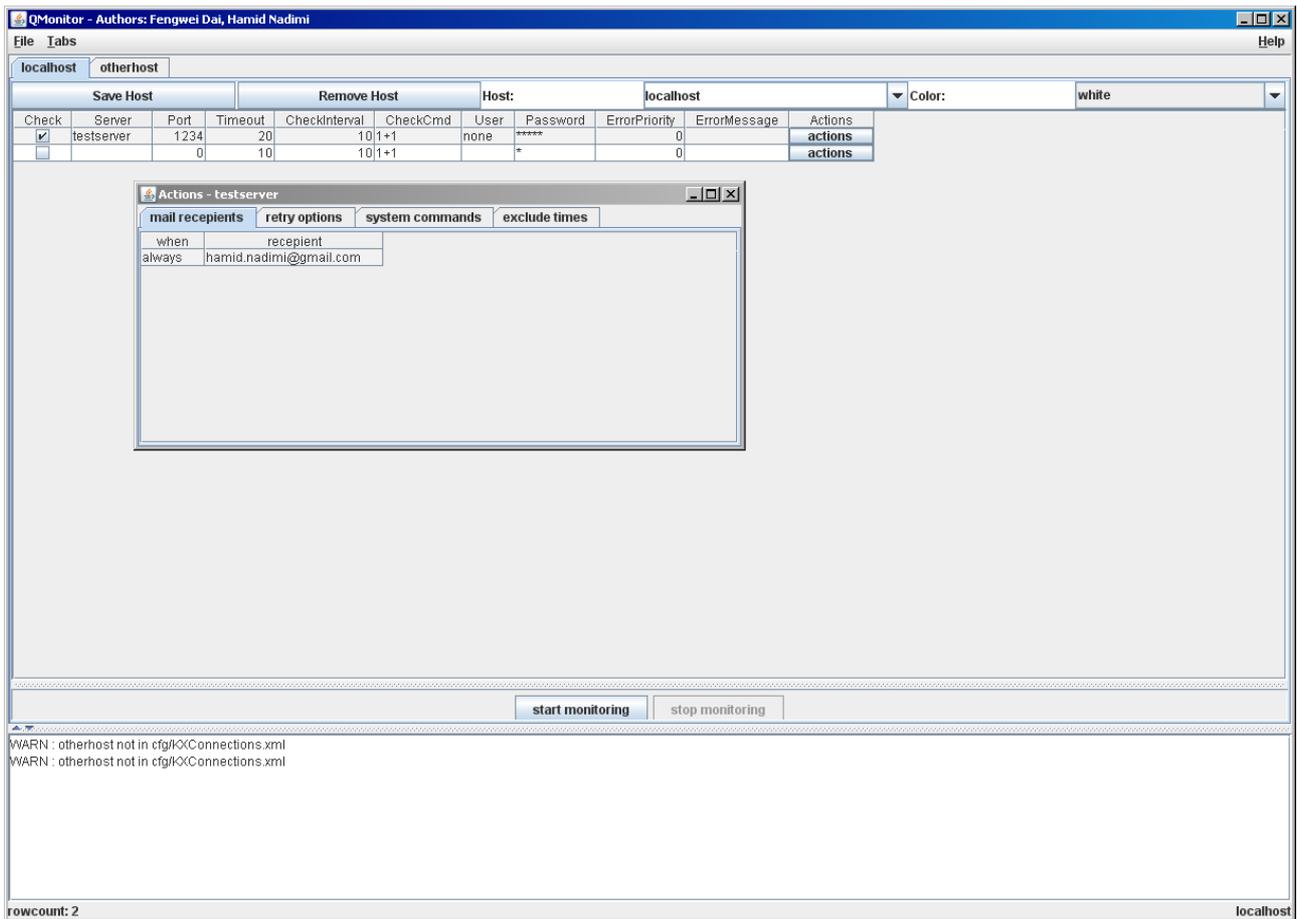
You can open multiple tabs to connect to monitor different hosts via shortcut or menu or mouse-right-click. In the same way you can close those tabs. You can reorganize tabs by simple drag and drop.





Actions

For each server you can click on actions to configure different actions and parameters. In order to save them permanently, you need to click on "Save Host"



Each Action frame contains several tabs.

mail recipients

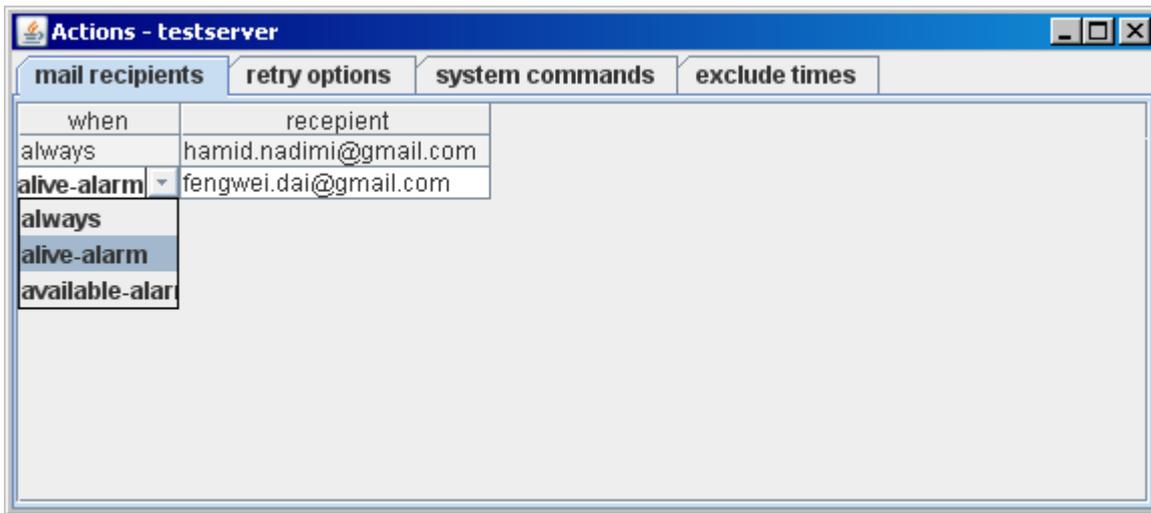
In order for this Feature to work, you have to specify a smtp server in `cfg/QMonitor.cfg`.

`mail.host=<your smtp server>`

On Windows, You can find this parameter in you Outlook settings.

To test if it works, you must restart the program. Select `File->Send Test Email` and enter your email address.

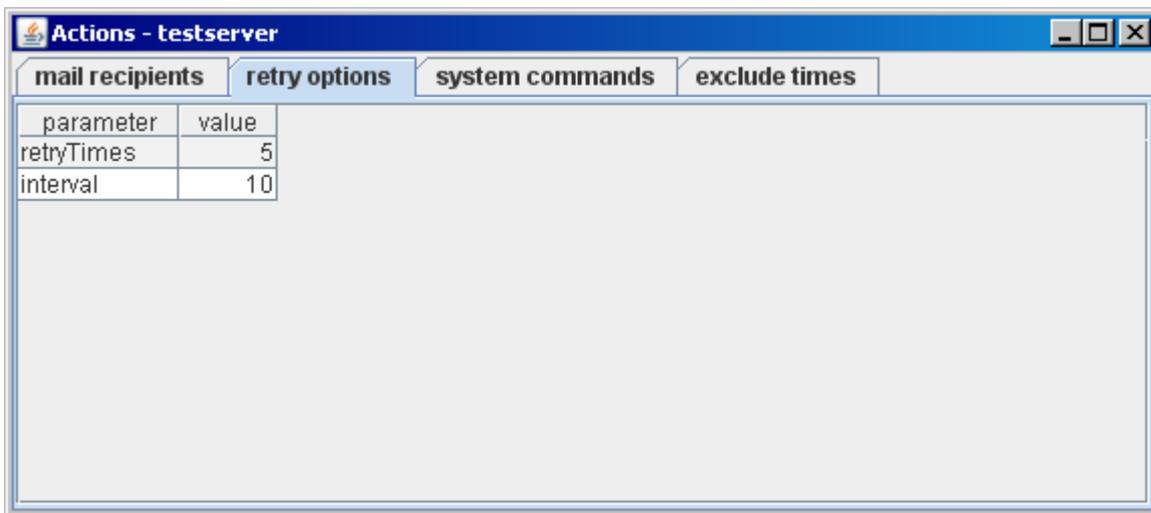
On mail recipients tab, you can enter email-addresses and specify when to send an Email.



Always = Send Email for all kind of alarms.
 Alive-Alarm = Send Email only if Server is gone.
 Available-Alarm = Send Email only if Server is blocked.

To add a row, right-click on a row and choose add-row. To remove a row, same way. Last row can not be removed. If you don't want send any email, delete email-address.

retry options



retryTime = This parameter specifies how often to retry monitoring.
 Interval = How many seconds to wait before each retry.

system commands



Here you can specify what to execute, when to do it and which parameters to use.

Always = Execute for all kind of alarms.

Alive-Alarm = Execute only if Server is gone.

Available-Alarm = Execute only if Server is blocked.

You can use the following Keywords as parameters. They will be replaced accordingly when the system command is called.

--HostName--

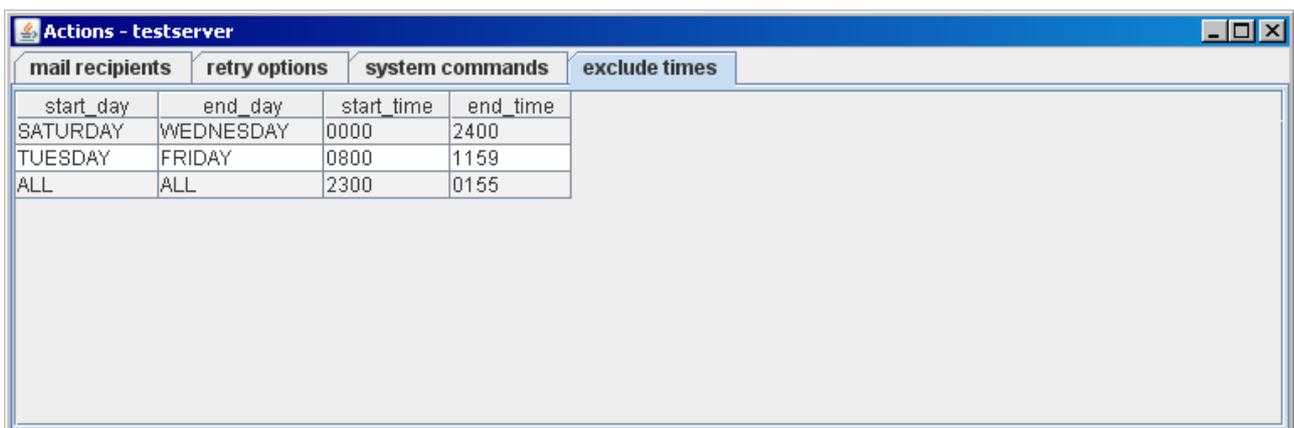
--ServerName--

--ServerPort--

--ErrorMessage--

To add a row, right-click on a row and choose add-row. To remove a row, same way. Last row can not be removed. If you don't want to execute any command, remove command and parameter.

exclude times



This page says when **not** to monitor.

Start_day: First day of exclusion. choose ALL (every day) or one of SUNDAY-SATURDAY.

Week starts with SUNDAY (1) and ends with SATURDAY (7).

End_day: Last day of Exclusion.

Start_time: when to stop monitoring.

End_time: when to start monitoring again.

Start and end are used for each day.

Examples:

TUESDAY-FRIDAY 0800-1159 means do not monitor on Tuesday, Wednesday, Thursday and Friday between 08:00 and 11:59.

ALL-ALL 2300-0155 means do not monitor between 23:00h and 0155 of the following day.

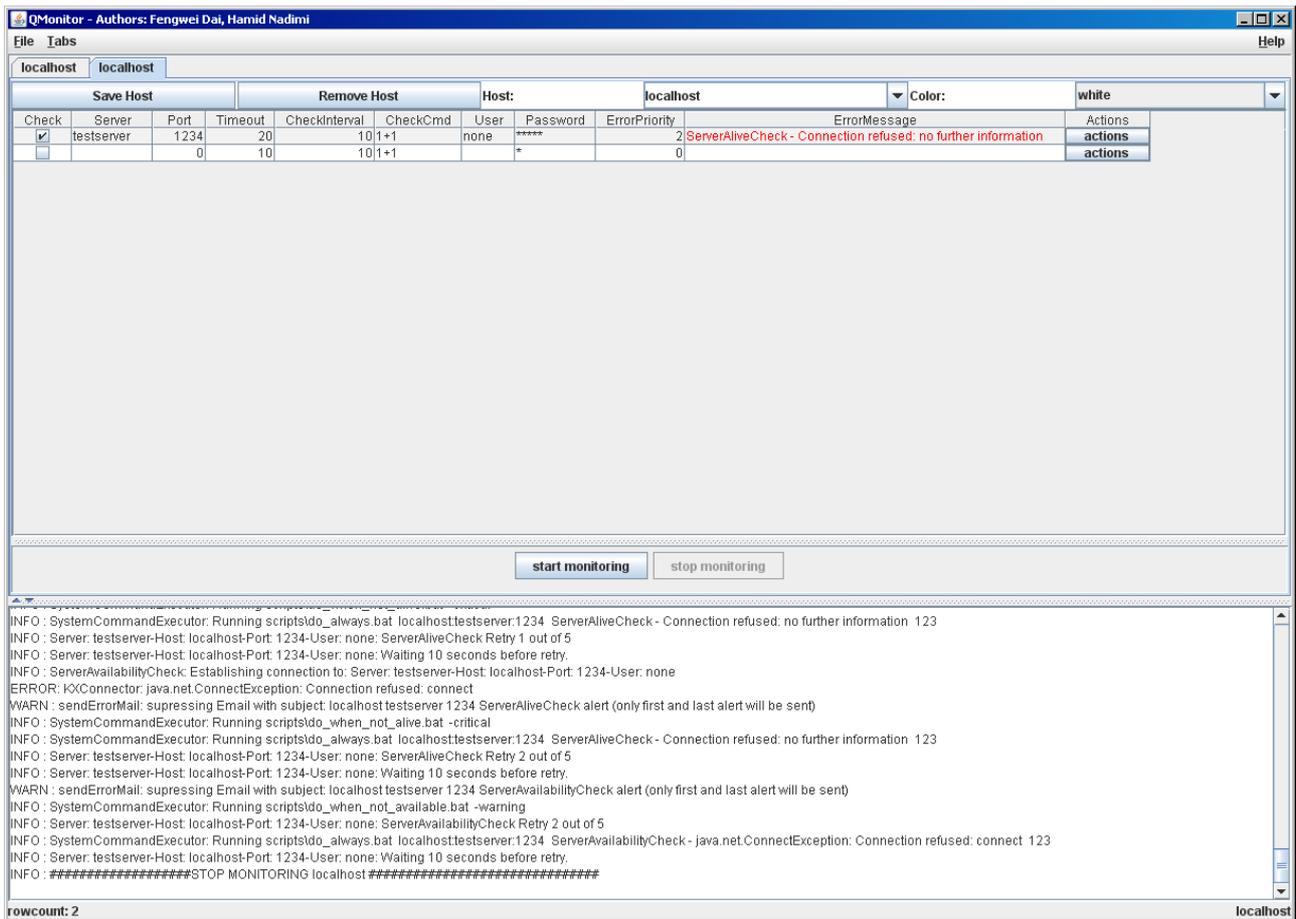
SATURDAY-WEDNESDAY 0000-2400 means do not monitor on Saturday, Sunday, Monday, Tuesday and Wednesday.

When you first start monitoring these parameters are ignored. They will only be considered if an error occurs. If the error occurred during exclusion-time, it will be suppressed. A retry will happen only after exclusion-time is over.

Log-Panel

Logging is done via log4j, every thing is logged to stdout, logfile and the logpanel. Logfiles are dailyrolled and stored in log/.

log4j.rootLogger=INFO, stdout, F, TEXTAREA



Cronjob-Mode

You can use QMonitor.bash to run this program in via without GUI on Unix or Linux. If you want to use your own script, remember that you need to disable graphics by using:

`-Djava.awt.headless=true`

In Cron-Mode a socket has to be opened for communication. Default port in the script is 23456. If this port is already used, change it.

The following commands can be sent to a running monitor process:

STOP: Stop monitoring all servers. (but keep process running)
 RESUME: Resume monitoring all servers.
 STOP_<port> Stop monitoring server with the specified port (for example STOP_1234 stops monitoring the server with port 1234)
 RESUME_<port> Resume monitoring server with the specified port
 TERMINATE Terminate monitor (stop process).

Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Please send your suggestions and bug reports to hamid.nadimi@gmail.com.