

# CoffeeSaint



CoffeeSaint is a very much configurable Nagios status viewer. It is meant to be run on a screen on a wall, visible to all system operators. It is written in Java so you can run it on Linux, MacOS X, microsoft windows and all operating systems capable of running Java version 6 (or later) software. The name CoffeeSaint stems from the Oracle Java-logo being a cup of coffee and because the original name from Nagios is "Netsaint".



Written by Folkert van Heusden.

<http://www.vanheusden.com/java/CoffeeSaint/>

Released with GPLv2 license.

## Table of contents

Written by Folkert van Heusden.....	1
www.vanheusden.com.....	1
Released with GPLv2 license.....	1
Nagios configuration.....	5
Serve the file via inetd / xinetd.....	5
inetd.....	5
xinetd.....	6
Slow network links.....	6
Serve the file via a webserver.....	6
Serving as a compressed stream using Apache.....	6
Status via Livestatus.....	6
Starting CoffeeSaint without any tweaking (quick how-to).....	8
Retrieving the status from inetd/xinetd.....	8
Without compression.....	8
With compression.....	8
Retrieving the status from a webserver.....	8
Retrieving the status from a Liveupdate connection.....	8
Retrieving the status from a local file.....	8
Easy configuration via web-interface.....	8
Web-interface parameters.....	8
Example.....	9
What you see.....	11
Commandline switches to list parameters.....	13
--list-screens.....	13
--list-fonts.....	13
--list-bgcolors.....	13
Configuring CoffeeSaint.....	14
source.....	14
allow-all-ssl.....	14
disable-http-compression.....	14
proxy-host and proxy-port.....	14
nrows.....	15
interval.....	15
flash.....	15
webcam-timeout.....	15
use-host-alias.....	15
fullscreen.....	15
image.....	15
adapt-img.....	15
ignore-aspect-ratio.....	16
random-img.....	16
transparency.....	16
header-transparency.....	16
cam-cols / cam-rows.....	16
problem-columns.....	16
flexible-n-columns.....	17
logo.....	17
logo-position.....	17
anti-alias.....	17
max-quality-graphics.....	17

header.....	17
footer.....	18
no-header.....	18
sort-order.....	18
prefer.....	18
also-acknowledged.....	18
always-notify.....	18
also-scheduled-downtime.....	18
also-soft-state.....	18
also-disabled-active-checks.....	18
show-flapping.....	18
show-flapping-icon.....	19
filter-unknown.....	19
filter-down.....	19
show-services-for-host-with-problems.....	19
counter.....	19
counter-position.....	19
problem-row-gradient.....	19
bgcolor.....	20
bgcolor-fade-to.....	20
sound.....	20
exec.....	20
verbose.....	20
text-color.....	20
warning-textcolor.....	20
critical-textcolor.....	20
color-bg-state.....	21
warning-bg-color.....	21
critical-bg-color.....	21
nagios-unknown-bg-color.....	21
font.....	21
critical-font.....	21
warning-font.....	21
splitter.....	21
draw-problems-service-split-line.....	21
no-problems-text.....	22
state-problems-text.....	22
row-border.....	22
row-border-color.....	22
host-issue.....	22
service-issue.....	22
reduce-textwidth.....	22
scroll-if-not-fitting.....	23
scroll-pixels-per-sec.....	23
scrolling-header.....	23
hosts-filter-exclude.....	23
hosts-filter-include.....	23
services-filter-exclude.....	23
services-filter-include.....	23
sparkline-width.....	23
sparkline-mode.....	23
graph-color.....	23

performance-data-filename.....	24
Escapes.....	25
Font style escapes.....	25
Textual escapes.....	25
%CRITICAL.....	25
%WARNING.....	25
%OK.....	25
%UP.....	25
%DOWN.....	25
%UNREACHABLE.....	25
%PENDING.....	25
%TOTALISSUES.....	25
%STATE.....	26
%H:%M.....	26
%HOSTNAME / %SERVICENAME.....	26
%SERVERNAME.....	26
%HOSTSTATE / %SERVICESTATE.....	26
%HOSTSINCE / %SERVICESINCE.....	26
%HOSTSINCETS / %SERVICESINCETS.....	26
%HOSTFLAPPING / %SERVICEFLAPPING.....	26
%PREDICT / %HISTORICAL.....	26
%HOSTDURATION / %SERVICEDURATION.....	26
%OUTPUT.....	26
@FIELDDATEHOST^field@.....	27
@FIELDDATESERVICE^field@.....	27
@FIELDBOOLEANHOST^field@.....	27
@FIELDBOOLEANSERVICE^field@.....	27
@FIELDHOST^field@.....	27
@FIELDSERVICE^field@.....	27
@FIELD^x@.....	27
@EXEC^script@.....	27
%PERCENT.....	27
%AT.....	27
%CHECKLATENCY.....	28
%NACKED.....	28
%NFLAPPING.....	28

## Nagios configuration

Nagios keeps track of its status in a file (mostly) called *status.dat*. This file is located somewhere in the filesystem. CoffeeSaint uses the contents of this file to display what is happening. There's also another way which is discussed later on.

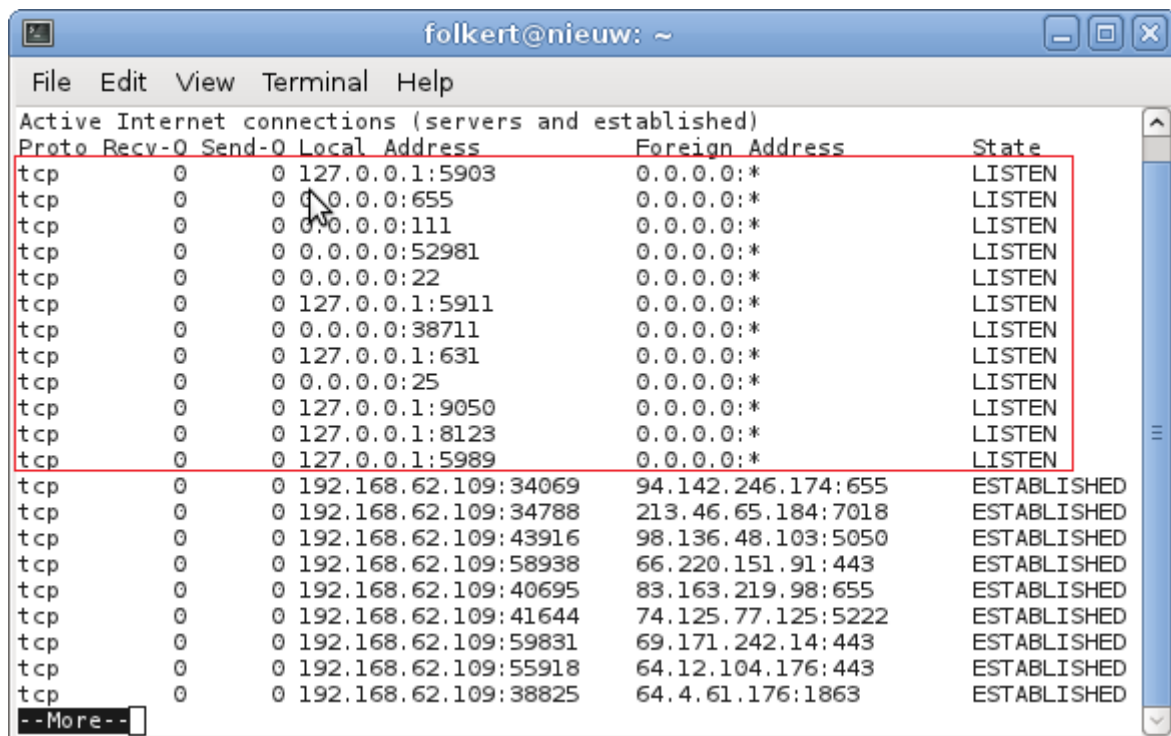
In case you choose to serve the *status.dat* file directly to CoffeeSaint then you need to find where it is. Open the file *nagios.cfg* and look for a line starting with **status\_file=**

- Default Nagios installation: */usr/local/nagios/var/status.dat*
- Debian: */var/cache/nagios3/status.dat*
- Groundwork: */usr/local/groundwork/nagios/var/status.log*
- EPEL (Fedora or RedHat enterprise Linux): */var/log/nagios/status.dat*

CoffeeSaint can access the *status.dat* file in a couple of ways.

### Serve the file via *inetd* / *xinetd*

Most UNIX systems have either *inetd* or *xinetd* running. This program listens on a couple of TCP and UDP ports and, if a connection is made, (x)inetd starts a program which handles the connection. The difference between *inetd* and *xinetd* is mostly the configurationfile. For both options you need to select a TCP port to listen on. This can be, for example port 33333. Verify using **netstat -na** if this port is free or not:



```
folkert@nieuw: ~
File Edit View Terminal Help
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:655             0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:111             0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:52981           0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:5911          0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:38711           0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:631          0.0.0.0:*                LISTEN
tcp        0      0 0.0.0.0:25              0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:9050          0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:8123          0.0.0.0:*                LISTEN
tcp        0      0 127.0.0.1:5989          0.0.0.0:*                LISTEN
tcp        0      0 192.168.62.109:34069    94.142.246.174:655      ESTABLISHED
tcp        0      0 192.168.62.109:34788    213.46.65.184:7018     ESTABLISHED
tcp        0      0 192.168.62.109:43916    98.136.48.103:5050     ESTABLISHED
tcp        0      0 192.168.62.109:58938    66.220.151.91:443     ESTABLISHED
tcp        0      0 192.168.62.109:40695    83.163.219.98:655     ESTABLISHED
tcp        0      0 192.168.62.109:41644    74.125.77.125:5222    ESTABLISHED
tcp        0      0 192.168.62.109:59831    69.171.242.14:443     ESTABLISHED
tcp        0      0 192.168.62.109:55918    64.12.104.176:443     ESTABLISHED
tcp        0      0 192.168.62.109:38825    64.4.61.176:1863      ESTABLISHED
--More--
```

In this example the entries with “LISTEN” at the right are in use. E.g. port 655, 111, 52981, etc.

### inetd

For *inetd*, you then need to edit */etc/inetd.conf* and add the following line:

```
33333 stream tcp nowait root /usr/sbin/tcpd /bin/cat /var/cache/nagios3/status.dat
```

Please note that you need to change the '33333' in this example to a free port number on that server. Also you need to replace */var/cache/nagios3/status.dat* to the location of the *status.dat* on your system. This example-location is for Debian.

Don't forget to restart inetd.

## xinetd

For xinetd you must create a file called `/etc/xinetd.d/nagios`.

```
service nagios
{
    disable      = no
    socket_type  = stream
    protocol     = tcp
    wait         = no
    user         = nagios
    server       = /bin/cat
    server_args  = /var/cache/nagios3/status.dat
}
```

After that add the following line to `/etc/services`:

```
nagios          33333/tcp
```

Just like the inetd example you need to adjust the portnumber (33333) and status.dat location.

Do not forget to restart xinetd.

## Slow network links

It can happen that the connection between the Nagios server and the CoffeeSaint program is slow, for example when there's an internet-connection between them. In that case it can help to first compress the status.dat first before serving it to clients. To do this, replace `/bin/cat` with `/bin/gzip` or `/bin/gzip -c9` for maximum compression.

## Serve the file via a webserver

This is just a matter of either:

- moving the **status.dat** to somewhere underneath the document root of your webserver. This might give problems with access rights.
- creating a symbolic link from the **status.dat** file to somewhere underneath the document root of the webserver. You might need to tweak the webserver configuration so that it allows symbolic links in its document tree.

## Serving as a compressed stream using Apache

Change your Apache configuration file to include the following lines. On Debian systems this can be found at `/etc/apache2/mods-enabled/deflate.conf`.

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/html text/plain ... application/x-ns-
proxy-autoconfig
</IfModule>
```

The part which matters is “application/x-ns-proxy-autoconfig”.

## Status via Livestatus

Nagios can deliver its status to client tooling via the Livestatus interface. This is a nagios add-on, e.g. it requires changes in the `nagios.cfg` file as well as adding a module. For details how to

configure Nagios, see the following web-site: [http://mathias-kettner.de/checkmk\\_livestatus.html](http://mathias-kettner.de/checkmk_livestatus.html)

For example this can be:

```
broker_module=/usr/local/mk-livestatus/lib/mk-livestatus/livestatus.o  
/var/lib/nagios3/rw/live event_broker_options=-1
```

This sets up a UNIX domain socket.

Livestatus sets up a UNIX domain socket. This socket can then be served via a TCP socket. Serving this TCP socket via inetd:

```
6557 stream tcp nowait root /usr/sbin/tcpd /usr/local/mk-livestatus/bin/unixcat  
/var/lib/nagios3/rw/live
```

Don't forget to restart inetd.

## Starting CoffeeSaint without any tweaking (quick how-to)

You need to adjust the parts written in bold. E.g. “mynagiosserver.local” should be replaced with the host-name or IP-address of the Nagiosserver, 33333 by a portnumber.

All examples below are for a Nagios version 2, 3 or Xi server. Replace e.g. 'tcp 3' by 'tcp 1' for old Nagios v1 compatibility.

### *Retrieving the status from inetd/xinetd*

Please note: behind the `--source` you add a 'pretty\_name': this parameter is then set to %SERVERNAME escape.

#### Without compression

```
java -jar CoffeeSaint.jar --source tcp 3 mynagiosserver.local 33333 pretty_name
```

Change the server as well as the port number ('mynagiosserver.local' and 33333).

#### With compression

```
java -jar CoffeeSaint.jar --source ztcp 3 mynagiosserver.local 33333 pretty_name
```

### *Retrieving the status from a webserver*

```
java -jar CoffeeSaint.jar --source http 3 http://myserver.local/status.dat pretty_name
```

### *Retrieving the status from a Liveupdate connection*

```
java -jar CoffeeSaint.jar --source ls 3 mynagioshost.company.com 6557 pretty_name
```

### *Retrieving the status from a local file*

```
java -jar CoffeeSaint.jar --source file 3 /var/cache/nagios3/status.dat pretty_name
```

## *Easy configuration via web-interface*

CoffeeSaint includes a build-in easy to use webserver. This webserver has a couple of functions, one which is configuring CoffeeSaint. If you start fresh without an existing configuration file, then first create an empty file which will be replaced with the settings you chose. On UNIX systems this can be accomplished with: **touch myconfigfile.conf**

Starting CoffeeSaint with the web-interface

```
java -jar CoffeeSaint.jar --listen-port 12340
```

You need to replace **12340** with a port number not in use.

## Web-interface parameters

These settings can also be put in the configuration file. “**--listen-port**” would then become “**listen-port**”.

- **--listen-port**: the TCP port on which the web-interface will listen. So if the CoffeeSaint pc has network name 'cs.local' and you select port 12340, you would then point your web-



browser to <http://cs.local:12340/>

- **--listen-adapter:** if the CoffeeSaint pc has multiple network-adapters then with this parameter you can select the network adapter to listen on. Default is to listen on all interfaces.
- **--web-username:** (use with **--web-password**) selects the username required for the webinterface.
- **--no-authentication:** disable authentication in the web-interface
- **--disable-http-fileselect:** Do not allow web-interface to select a file to write configuration to. Useful if CoffeeSaint is running on a public system.

## Example

### Configuring CoffeeSaint

The screenshot shows the CoffeeSaint v2.3-beta002 configuration web interface. The browser address bar shows <http://keetweej.vanheusden.com:12340/cgi-bin/config-menu.cgi>. The page title is "coffeesaint v2.3-beta002".

**Nagios handling parameters**

- Always notify:  Also display when notifications are disabled
- Also acknowledged:
- Also scheduled downtime:  Also display problems for which downtime has been scheduled
- Also soft state:
- Also disabled checks:  Also display problems for which active checks have been disabled
- Show services for host with problems:
- Show flapping:

**Network parameters**

Please note that after you did: on submit, the new network-settings are applied immediately.

Network interface to listen on:

Port to listen on:

**Nagios server(s)**

type	Nagios version	data source	remove?
tcp	3	localhost:33333	<input type="checkbox"/>

Use HTTP compression:

TCP requires an ip-address followed by a space and a port-number in the parameters field.

**Look and feel parameters**

Refresh interval:  > 1

Show counter:

### Performance graphs

Click on a host/service to zoom in into a graph.

**CoffeeSaint admin**

Reload configuration  
Select configuration file  
Write config to cs.conf

**Actions**  
Force reload disabled, not running GUI  
No sound selected

**Links**  
Links relevant to this program

vanheusden.com

Service	Type	Path	Metric 1	Metric 2	Metric 3	Metric 4	Metric 5
belle	Disk space	/home	7928MB	13280MB	10093.2800MB	2363.0982MB	1000
belle	Disk space	/oldroot	10317MB	10317MB	10317MB	0MB	1000
belle	Disk space	/usr/src	6589MB	6589MB	6589MB	0MB	1000
belle	Disk space	/idata	33530MB	34005MB	33758.5740MB	123.3735MB	1000
belle	Disk space	/idata3	34974MB	66530MB	53681.2510MB	13260.2320MB	1000
belle	Disk space	/idata4	98296MB	100441MB	98526.7180MB	195.1622MB	1000
belle	Disk space	/idata4/Backup	25235MB	34777MB	32204.7430MB	1734.9825MB	1000
belle	Disk space	/idata/movie	61708MB	61708MB	61708MB	0MB	1000
belle	FTP	time	0.0289s	0.2099s	0.0333s	0.0127s	1000
belle	HTTP	time	0.0008s	0.0710s	0.0026s	0.0038s	1000
belle	HTTP	size	977B	977B	977B	0B	1000
belle	NTP	offset	-0.0409s	0.0505s	-0.0001s	0.0058s	1000
belle	USERS	users	0	10	5.2910	2.0294	1000
fatelec_eus.nu	rta		22.5260ms	310.9500ms	42.4021ms	40.3365ms	1000
fatelec_eus.nu	pl		0%	66%	0.2660%	3.7796%	1000
fatelec_eus.nu	HTTP	time	0.0506s	7.0925s	0.1158s	0.2995s	1000
fatelec_eus.nu	HTTP	size	262B	262B	262B	0B	1000
fatelec_eus.nu	NTP	offset	-0.0663s	0.0368s	0.0009s	0.0154s	1000
fatelec_eus.nu	diskspace	/	25355MB	45072MB	31544.2730MB	7889.9082MB	1000
fatelec_eus.nu	diskspace	/lib/init/rw	0MB	0.0000MB	0MB	0MB	1000
fatelec_eus.nu	diskspace	/dev	0MB	0.0000MB	0MB	0MB	1000
fatelec_eus.nu	diskspace	/dev/shm	0MB	0.0000MB	0MB	0MB	1000
filmhuis	ping		85.7899	1131.3045	125.7074	81.3873	1000
filmhuis	#USERS	users	2	3	2.0100	0.0995	1000
filmhuis	FTP	time	0.1032s	4.5113s	0.1697s	0.2627s	1000
filmhuis	HTTP	time	0.1288s	4.1269s	0.1864s	0.2284s	1000
filmhuis	HTTP	size	2601B	2888B	2719.6600B	129.0026B	1000
filmhuis	btdevs records age	age	6	47038	8928.5410	9544.3405	1000
filmhuis	disk space	/	29488MB	39407MB	33580.8010MB	3138.0388MB	1000

### Applet mode

If you select “problems overview” in the menu, you can run CoffeeSaint as a web-browser applet. That way you can install it in 1 location and view it everywhere:

Cr 28/Wa 2/Un 0/Dn 8, 13:37

platform.ticketlafatelectron.soleu

mauer: See if ST (mauer: check pro

mauer: financial belle: age of the

mauer: financial fweerstationserve

mauer: See if ma

mauer: See if PO

mauer: room ten

mauer: # msgs i

Beelden: Rijk

## What you see

When the program is loading the Nagios status (via the network or a file) a blue block is drawn in the upper right top-row. If the loading fails in any way, this blue block is replaced by a red block and in the lower row an explanation is shown.

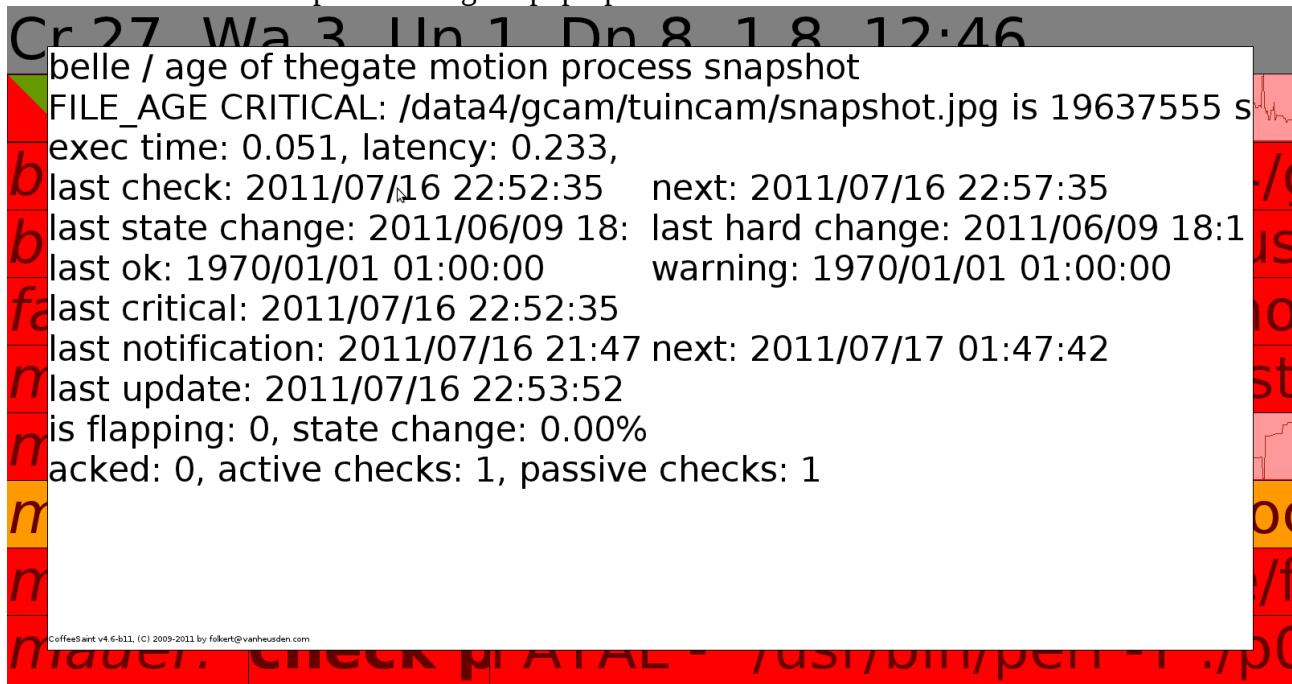


Also in the upper row a couple of values are displayed. These tell you the number of critical, warning and ok statuses as well as the number of up, down, unreachable and pending hosts. The contents of this row can be configured with the `--header` commandline switch as well as the `"header = ..."` configuration file setting.



Of course, you can disable the header-line. Use `"--no-header"` (or `"header = false"` in the configuration file) for that.

One can also click on a problem to get a pop-up with some more details:



If you click on the header-line instead, you'll get more general information:

Cr 27, Wa 3, Un 1, Dn 8, 1.8, 12:47  
Nagios source: status.dat  
Load time: 0.004s, hosts: 19, services: 120  
Total hosts: 19, services: 120  
OK: 80, warning: 3, critical: 27  
UP: 10, down: 8, unr.: 1, pending: 0  
Average check latency: 0.177, last check age: 49995s  
Problems acked: 0, flapping: 1

4

Coffes@aint v4.6-b11, (C) 2009-2011 by fokert@vanheesden.com

An example of a common configuration:

```
ok: 89, unreachable 0, down: 1, pending: 0,  
denhaag  
markjansen: Check Nagios summary  
thegate: buuf_internet
```

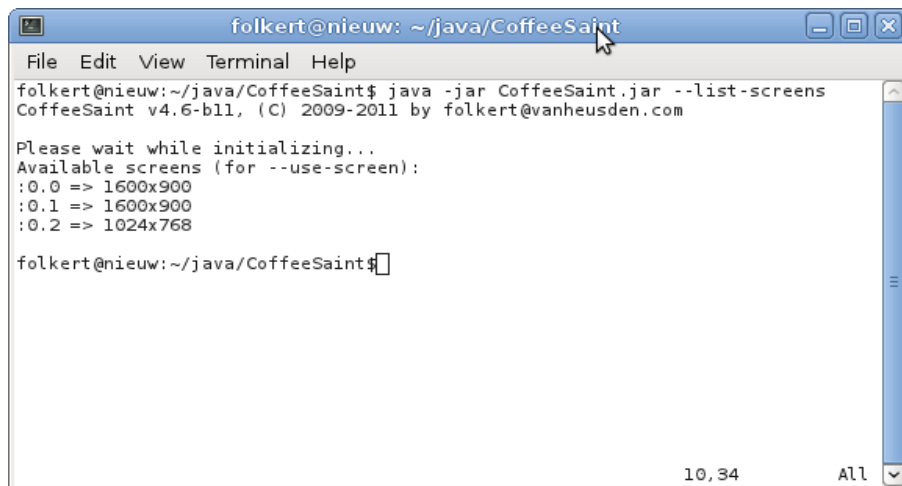
2009-12-07 19:15:34

## Commandline switches to list parameters

### **--list-screens**

Shows a list of screens in the system. For example multiple monitors connected to one system. You can then either use all screens (**--fullscreen allmonitors**) or select one (**--use-screen x** where x is one of the identifiers returned with **--list-screens**).

On UNIX/Linux this will look like this:

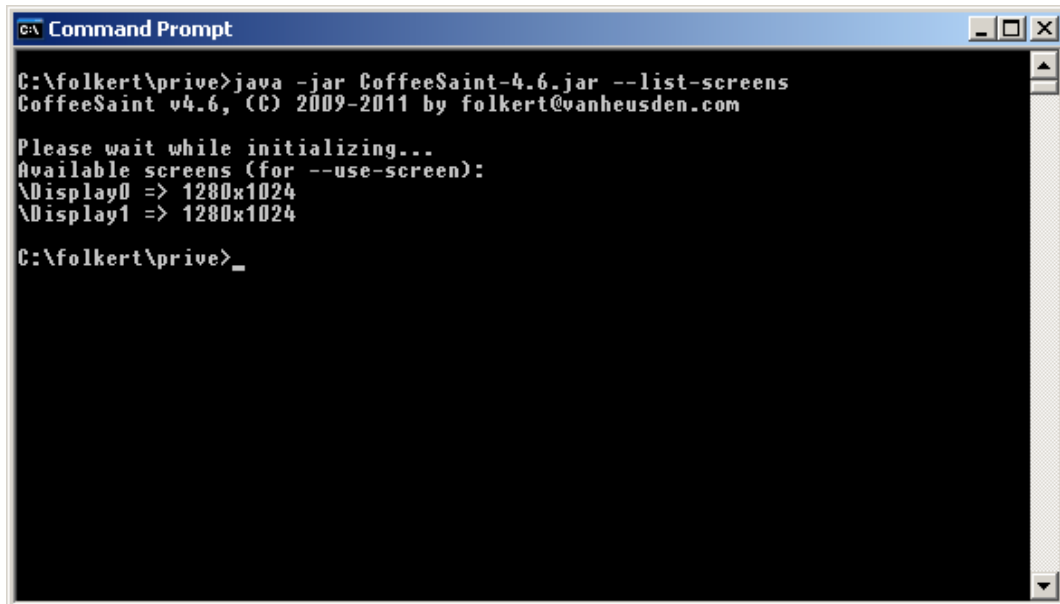


```
folkert@nieuw: ~/java/CoffeeSaint
File Edit View Terminal Help
folkert@nieuw:~/java/CoffeeSaint$ java -jar CoffeeSaint.jar --list-screens
CoffeeSaint v4.6-b11, (C) 2009-2011 by folkert@vanheusden.com

Please wait while initializing...
Available screens (for --use-screen):
:0.0 => 1600x900
:0.1 => 1600x900
:0.2 => 1024x768

folkert@nieuw:~/java/CoffeeSaint$
```

On microsoft windows xp this will look like this:



```
C:\folkert\prive>java -jar CoffeeSaint-4.6.jar --list-screens
CoffeeSaint v4.6, (C) 2009-2011 by folkert@vanheusden.com

Please wait while initializing...
Available screens (for --use-screen):
\Display0 => 1280x1024
\Display1 => 1280x1024

C:\folkert\prive>
```

### **--list-fonts**

Shows a list of fonts known by the system. Can then be used with e.g. “**--font fontname**”. Note: in Windows, Java ignores your vector, bitmap and PostScript fonts. It can only use the TrueType and OpenType fonts. Further it will only use TrueType and OpenType fonts with Unicode encodings. Older fonts or fonts with only a few specialty characters will often not work because they come only with 8-bit encodings. Modern fonts often come with several encodings. Java just uses the Unicode encoding. See <http://mindprod.com/jgloss/font.html> for more details.

## ***--list-bgcolors***

Shows a list of color names known by CoffeeSaint. Used with for example “**--warning-textcolor x**” and “**--textcolor x**”.

## Configuring CoffeeSaint

CoffeeSaint can be configured with parameters on the commandline and/or via a configuration file. To see a list of command line switches, run CoffeeSaint with the **--help** switch.

In short: a commandline switch can be placed “as is” in the configurationfile: **--interval 30** would become **interval = 30** in the configuration file. Start CoffeeSaint with **--config myconfigfile.conf** to use that file.

You select the configuration file with “**--config filename**”. Please note that “**--config**” must be the first parameter on the command-line as it overwrites all previous settings.

A couple of settings refer to nagios status.dat-file fields: those are listed in the chapter “Nagios status.dat fields”.

### **source**

Defines a Nagios status-source. You need to have at least one and can have many more of them.

- **source** = type version file/url/host+port pretty\_name
- **type** can be:
  - **http**: expects an URL to retrieve the status from a webserver
  - **http-auth**: retrieves the status from a http server with basic authentication. After the url you need to enter the username and password
  - **tcp**: retrieve the status.dat from a socket, e.g. via inetd/xinetd. Requires a hostname/IP-address and a port number.
  - **ztcp**: retrieve the status.dat compressed with gzip compression from a socket. Requires a hostname/IP-address and a port number.
  - **ls**: retrieve the status from a Livestatus source. Requires a hostname/IP-address and a port number.
  - **file**: retrieve the status from a local file (can be on an NFS or other network filesystem as well)
- **version** can be 1, 2 or 3. For Nagios Xi use 3. 2 and 3 are basically the same.
- **file/url/host+port/etc.** depends on the source-type.
- **pretty\_name** since version 4.6 of CoffeeSaint you can give each Nagios-source a pretty name. This then will be shown if there are problems (in the footer-line at the bottom) or when using the **%SERVERNAME** escape).

### **allow-all-ssl**

When using a http-source with an https-URL this makes CoffeeSaint ignore certificate issues.

Usefull for example if the webserver uses a self-signed certificate. If this is used on the command-line, it must be the first parameter (AFTER “**--config**”), *before* the **--source** switch. In the configuration file it expects either 'true' or 'false'.

### **disable-http-compression**

If you're retrieving the Nagios status via a webserver, in some cases the webser may automatically compress the data before transmitting it to the client. This may not be required on a LAN. With this parameter you can forcibly switch compression of. Must be 'true' or 'false'.

## ***proxy-host and proxy-port***

When retrieving the nagios status via a webserver you might need to configure a proxy-server. With proxy-host and proxy-port you can set this up.

## ***nrows***

This tell CoffeeSaint how many rows you want. Default is 10. This includes the header-line and the optional footer-line.

## ***interval***

How often to refresh the screen. This selects the interval in which the status is retrieved. In seconds. Default is 30.

## ***flash***

In some occasion you may have a couple of problems listed on the screen which have been there for a while. Then if a new problem shows up and no warning sound is selected you might miss that event. With '**flash = true**' you let CoffeeSaint flash this new entry for the duration of one refresh-interval.

## ***webcam-timeout***

Using the 'image' parameter you can show an image or webcam on the background. If you're displaying a remote image and the remote server is down, this might take a while. With this parameter you can select how long to try – in seconds.

## ***use-host-alias***

Show the “pretty name” of a nagios-server instead of the hostname.

## ***fullscreen***

Display CoffeeSaint in full-screen mode. This mode has a couple of modusses:

- **undecorated:** a window without borders
- **fullscreen:** no menu bars
- **allmonitors:** spread out over all monitors
- **none:** windowed mode

## ***image***

Selects one or more images to display on the background. Can be remote images, e.g. webcams in the serverroom. Requires 2 parameters: type and a path/url.

- **file:** selects a file in the local/mounted filesystem(s)
- **http/https:** select is an image on a remote webserver.
- **mjpeg:** selects a MJPEG stream on a remote webserver. Displays one new frame every '**interval**' seconds.



## ***adapt-img***

Shrinks (or enlarges) the image (selected with '**image**') to fit underneath the list of problems.

## ***ignore-aspect-ratio***

Grow/shrink all webcams with the same factor. In case you have images with different dimensions.

## ***random-img***

When selecting multiple images, CoffeeSaint randomly selects one to display each interval. Default behaviour is to display them in order of appearance on the commandline or configuration file.

## ***transparency***

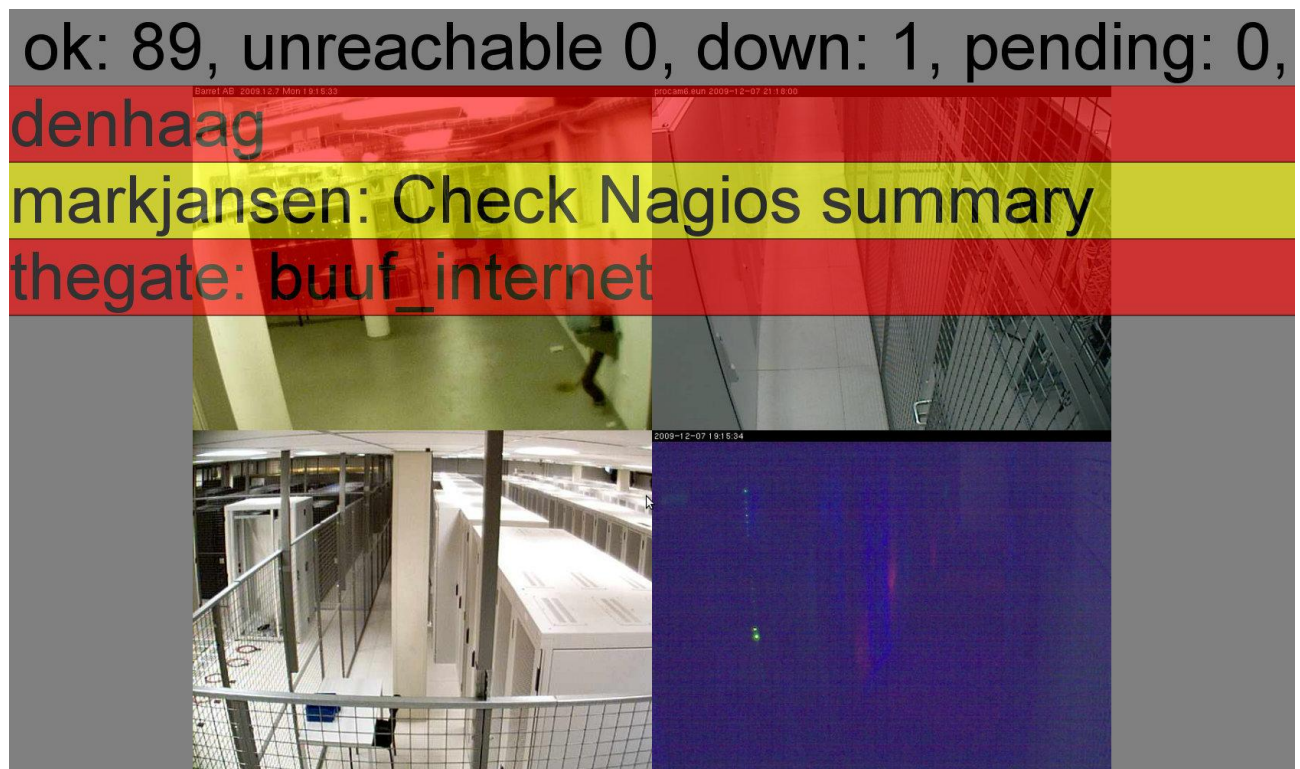
Gives a “see-through” effect. 0.5 means 50% see through. So you'll see the background image shimmer through the problem-text.

## ***header-transparency***

Like 'transparency' but only for the header-row.

## ***cam-cols / cam-rows***

Number of images to display on a row (**cam-cols**) and the number of image-rows to show (**cam-rows**).

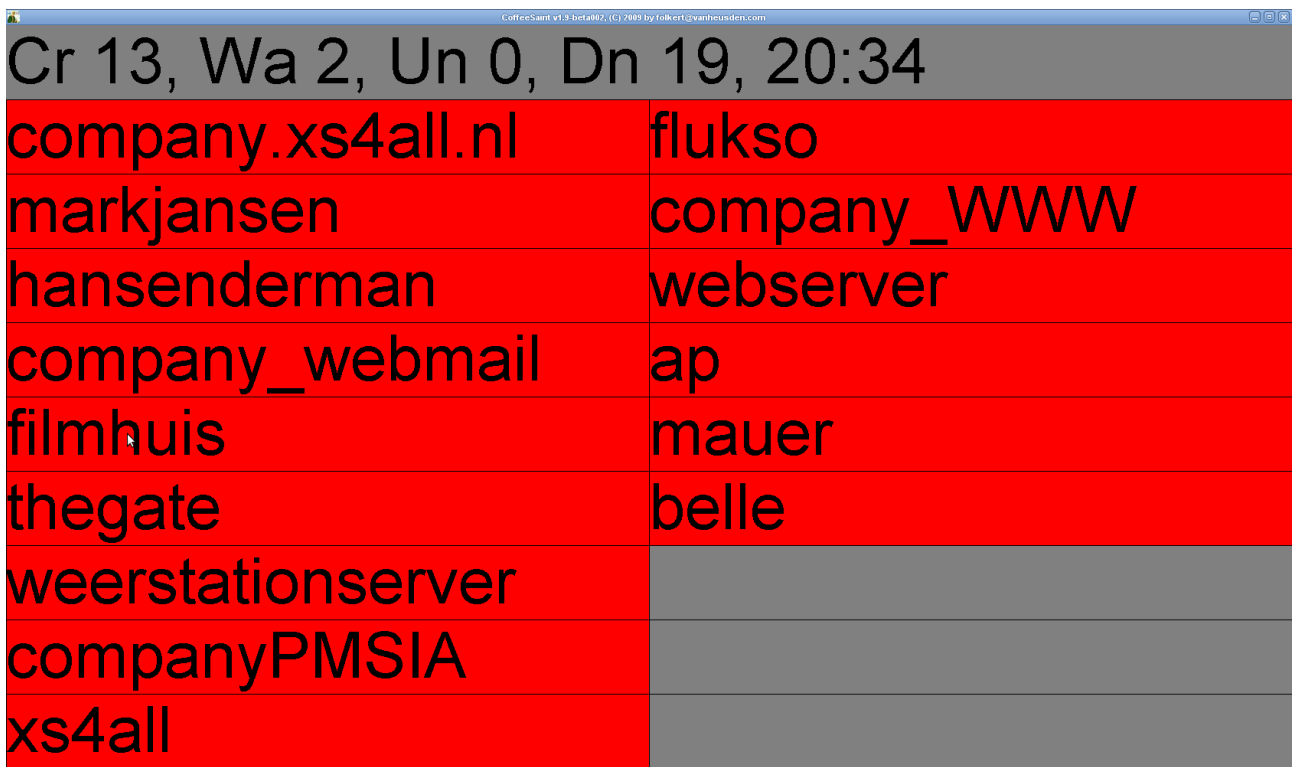


## ***problem-columns***

For really wide screens it might be usefull to split the screen in multiple columns. With this parameter you set the number of columns. Must be a value of 1 or bigger.

## ***flexible-n-columns***

Automatically increases the number of columns in case the number of problems is bigger than the 'nrows' parameter.



The screenshot shows a terminal window with a title bar that reads "Coffee-Saint v1.3-beta002 (C) 2003 by folkert@vanheesden.com". The terminal content is as follows:

```
Cr 13, Wa 2, Un 0, Dn 19, 20:34
company.xs4all.nl      flukso
markjansen            company_WWW
hansenderman         webserver
company_webmail      ap
filmhuis             mauer
thegate              belle
weerstationsserver
companyPMSIA
xs4all
```

company.xs4all.nl	flukso
markjansen	company_WWW
hansenderman	webserver
company_webmail	ap
filmhuis	mauer
thegate	belle
weerstationsserver	
companyPMSIA	
xs4all	

## ***logo***

Display a logo on the screen, in the header-line.

## ***logo-position***

Where to display the logo. Can be either "left" or "right".

Logo example (the Nagios-logo right):

## ***anti-alias***

When drawing the screen (the problems, images, etc.) use anti-aliasing. This improves how for example the fonts are rendered but it slows down the program.

## ***max-quality-graphics***

This setting improves the rendering settings. Slows down the program.

## ***header***

What to display in the header (top-most row). This string can contain escapes (%... and @...@, see chapter “scapes”).

## ***footer***

What to display in the footer (bottom row). If there's any problem, this row is replaced by the error message. Can also contain escapes. If this command is omitted, no footer line is displayed: makes more room for listing problems.

## ***no-header***

Don't display a header-line. Gives one more row for displaying problems.

## ***sort-order***

Selects the field to sort on. The names of the fields can be found in the nagios status.dat file which are explained at. If a field contains a numeric value, you can put “**numeric**” before the field-name. It'll then do a numeric-sort. If you put “**reverse**” in front of it, it'll do a reverse sort (biggest values at the top / strings starting with a 'z' at the top). See chapter “Nagios status.dat fields”.

## ***prefer***

Comma separated list of texts that are displayed at the top. This list can be a list of regular expressions. See <http://download.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html> for a description of what Java expects.

## ***also-acknowledged***

Display acknowledged (in Nagios itself) problems as well . Requires “**true**” or “**false**”.

## ***always-notify***

Also display problems for which notifications are disabled .

## ***also-scheduled-downtime***

Also display problems for which downtime has been scheduled .

## ***also-soft-state***

Also display problems that are not yet in hard state

### ***also-disabled-active-checks***

Also display problems for which active checks have been disabled .

### ***show-flapping***

Do not show hosts that are flapping .

### ***show-flapping-icon***

Show an icon in front of problems indicating wether they're flapping or not .



### ***filter-unknown***

Do not show hosts/services in unknown or pending state .

### ***filter-down***

Do not show hosts in down state .

### ***show-services-for-host-with-problems***

Default behaviour is not to display service-states for a host that is down as most of the time those services will be down as well.

### ***counter***

Display a counter which shows the number of seconds before the screen will refresh (see “interval”).

### ***counter-position***

Where to put the counter. Can be one of the following:

- upper-left
- upper-right
- lower-left
- lower-right
- center

- nowhere

### ***problem-row-gradient***

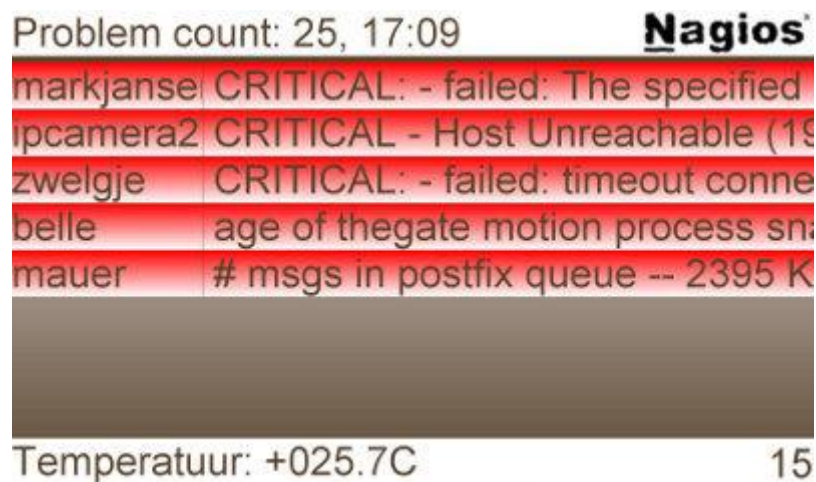
Don't let problems have a solid background color but display a gradient instead.

### ***bgcolor***

Select background color (get a list of available colors with “--list-bgcolors”).

### ***bgcolor-fade-to***

Select the color to fade the background to.



### ***sound***

Selects the audio-file to play when a new problem was detected. This must be in a format that is supported by Java. See <http://java.sun.com/products/java-media/sound/techReference/javasoundfaq.html#formats> for a list of formats.

### ***exec***

Selects a script/program to execute when a new problem is shown.

### ***verbose***

Let CoffeeSaint be a little more verbose on what it is doing. This for example shows what Nagios source it is retrieving. This information is shown in the header-row.

### ***text-color***

Color of header and footer texts.

### ***warning-textcolor***

Color of the warning problem texts.

### ***critical-textcolor***

Color of critical problem texts.

### ***color-bg-state***

When set to true, then the background color depends on the state. Critical errors will give for example a red background.

### ***warning-bg-color***

Background color to use when the biggest problem is a warning. Default is yellow.

### ***critical-bg-color***

Background color to use when the biggest problem is a critical. Default is red.

### ***nagios-unknown-bg-color***

Background color to use when the biggest issue has a unknown state.

### ***font***

Default font to use (e.g. header and footer texts).

### ***critical-font***

Font to use for critical problems.

### ***warning-font***

Font to use for warning-level problems.

### ***splitter***

This parameter can have one or more x-positions. They set the tab-stops. See the “\T” escape in the escapes section.

### ***draw-problems-service-split-line***

When set to “**true**” show a vertical line on each tab-stop.

Cr 27, Wa 3, Un 1, Dn 8, 0.2, 23:28

```
ap: PING OK - Packet loss = 0%, RTT = 0.000ms
belle: age of t FILE _AGE CRITICAL: /data4/
belle: procs r FATAL - "/usr/bin/python /us
fatelectr procs r FATAL - "/usr/sbin/munin-no
mauer: # msgs -- 163 Kbytes in 30 Request
mauer: See if P CRITICAL - file /usr/local
mauer: See if S FILE _AGE WARNING: /usr/lo
mauer: See if n FILE _AGE CRITICAL: /home/f
mauer: check p FATAL - "/usr/bin/perl -T .20
```

### ***no-problems-text***

Message to display when there are no problems. Used by the %STATE escape.

### ***state-problems-text***

Message to display when there are problems. Used by the %STATE escape.

### ***row-border***

Draws a border around each shown problem.

### ***row-border-color***

Color of the border around each problem. Default is black.

### ***host-issue***

When there's a problem with a host, what to show. Normally consists of a couple of escapes selecting what to show.

### ***service-issue***

When there's a problem with a service, what to show. Normally consists of a couple of escapes selecting what to show.

### ***reduce-textwidth***

When the text of a problem doesn't fit on the screen or in a column, shrink the font so that it fits.

### ***scroll-if-not-fitting***

If the problem text (with all escapes expanded) does not fit on the screen, this setting will make it scroll (from right to left).

### ***scroll-pixels-per-sec***

Number of pixels to scroll per second (default: 100) . This sets the scrolling speed.

### ***scrolling-header***

Sets if the header should scroll if it doesn't fit.

### ***hosts-filter-exclude***

Comma-separated list of hosts not to display .

### ***hosts-filter-include***

Comma-separated list of hosts to display. Use in combination with **hosts-filter-exclude**: will be invoked after the exclude.

### ***services-filter-exclude***

Comma-separated list of services not to display .

### ***services-filter-include***

Comma-separated list of services to display. Use in combination with **services-filter-exclude**: will be invoked after the exclude.

### ***sparkline-width***

Adds sparklines to the listed problems. With this setting you enable them AND you define the width in pixels. The sparklines are graphs of the performance data which is measured by Nagios.

### ***sparkline-mode***

Defines how the measured data will be scaled.

- **avg-sd**: scale to [average – standard deviation] to [average + standard deviation]
- **min-max**: scale to the whole y-axis bandwidth

### ***graph-color***

Selects the color with which the graphs are drawn.



### ***max-check-age***

With this CoffeeSaint will check if (all) Nagios server(s) are up. That is: if its(their) checks were not performed longer than what you select with this parameter. It expects the number of seconds. Two notes: the value must be equal or larger than the refresh-interval (the “**interval**” parameter), also: make this value somewhat larger than if “interval\_length” (in *nagios.cfg*, default is 60 seconds) multiplied by the smallest “normal\_check\_interval” in the services definitions.

### ***performance-data-filename***

In what file CoffeeSaint should store the performance data as measured by Nagios. Please note that if CoffeeSaint is not running, then there will be gaps.

## Escapes

CoffeeSaint has two kinds of escapes: escapes that show details on problems, retrieved from the Nagios status.dat (textual escapes), and escapes that set font style parameters.

### *Font style escapes*

Escapes can be combined (e.g. both italic and bold). One “stops” a setting by repeating it's escape: \B...\B.

- \B show the following text in **bold**
- \I show the following text in *italic*
- \U show the following text with underline
- \S show the following text with ~~strike-through~~
- \T let the following text start at the next tab-stop (see “**splitter**”) - this escape does *not* require a terminating \T (!)

### Textual escapes

#### %CRITICAL

Total number of critical problems.

#### %WARNING

Total number of warning problems.

#### %OK

Total number of services that have the OK state.

#### %UP

Total number of hosts that are up.

#### %DOWN

Total number of hosts that are down.

#### %UNREACHABLE

Total number of hosts that are unreachable.

#### %PENDING

Total of hosts for which the status is not yet known.

#### %TOTALISSUES

Sum of critical, warning, down and unreachable.

## %STATE

Contents of the “**no-problems-text**” or “**state-problems-text**” parameters, depending on the worst state.

## %H:%M

Current time.

## %HOSTNAME / %SERVICENAME

Use for “**host-issue**” and/or “**service-issue**”.

## %SERVERNAME

Name of the Nagios server on which the current problem is happening.

## %HOSTSTATE / %SERVICESTATE

Current state, e.g. “Ok”, “Critical”.

## %HOSTSINCE / %SERVICESINCE

Textual timestamp of when the problem started.

## %HOSTSINCETS / %SERVICESINCETS

Number of seconds since the start of a problem. These escapes are especially useful for conditional escapes.

## %HOSTFLAPPING / %SERVICEFLAPPING

Tells you if the state is flapping.

## %PREDICT / %HISTORICAL

To be described.

## %HOSTDURATION / %SERVICEDURATION

How long a host/service has been in the current state.

## %OUTPUT

Output of the Nagios plugin that indicated a problem.

## @FIELDDATEHOST^field@

Take 'field' from the host-field (from the Nagios status.dat) and convert it into a date-string. E.g. when you have a field in status.dat that contains a timestamp (e.g. last\_state\_change=1310935509), this will then convert that to something readable (year/month/day etc.).

## @FIELDDATESERVICE^field@

Take 'field' from the service-fields and convert it into a date-string .

## @FIELDBOOLEANHOST^field@

Take 'field' from the host-fields and interpret as yes/no. If there's a field containing a boolean (e.g. has\_been\_checked=1), this escape will convert that to 'yes' or 'no'.

## @FIELDBOOLEANSERVICE^field@

Take 'field' from the service-fields and interpret as yes/no .

## @FIELDHOST^field@

Take 'field' from the host-fields and display its contents. E.g. it'll display "0.030" when you pick the field check\_execution\_time.

## @FIELDSERVICE^field@

Take 'field' from the service-fields and display its contents.

## @FIELD^x@

x can be 'field|service|host' or 'field|host' or 'field'. Here you need to replace 'host' with the host from which you want to see the status.dat field "field". Same thing for service. So @FIELD^check\_type|ping^myhost@ will show the contents of the field "check\_type" from the service "ping" on host "myhost".

## @EXEC^script@

This escape invokes "script" and shows the first line of it's output.  
The script gets as output:

- hostname
- servicename (or an empty field in case of a host failure)
- current state
- plugin output

UNIX example:

```
@exec^usr/local/bin/my_script.sh@
```

Windows example:

```
@exec^c:\programs\myprogam.exe@
```

## **%PERCENT**

Is replaced by a '%'.

## **%AT**

Will be replaced by a '@'.

## **%CHECKLATENCY**

Shows the latency of the check that indicated the problem.

## **%NACKED**

The number of acknowledged problems.

## **%NFLAPPING**

The number of problems for which the state is flapping.

## **Nagios status.dat fields**

These fields can be found in the Nagios status.dat file. They can be used for sorting and filtering.

### ***Sorting***

#### **Time**

last\_check  
last\_hard\_state\_change  
last\_state\_change  
last\_time\_critical  
last\_time\_down  
last\_time\_ok  
last\_time\_unknown  
last\_time\_unreachable  
last\_time\_up  
last\_time\_warning  
last\_update  
next\_check  
next\_notification

#### **Measurements**

check\_execution\_time  
check\_latency  
percent\_state\_change

#### **IDs**

current\_event\_id  
current\_notification\_id  
current\_problem\_id  
last\_event\_id  
last\_problem\_id

#### **Miscellaneous fields**

acknowledgement\_type  
active\_checks\_enabled  
check\_command  
check\_interval  
check\_options  
check\_period  
check\_type  
current\_attempt  
current\_notification\_number  
current\_state  
event\_handler  
event\_handler\_enabled

failure\_prediction\_enabled  
flap\_detection\_enabled  
has\_been\_checked  
host\_name  
is\_flapping  
last\_hard\_state  
last\_notification  
long\_plugin\_output  
max\_attempts  
modified\_attributes  
no\_more\_notifications  
notification\_period  
notifications\_enabled  
obsess\_over\_host  
obsess\_over\_service  
passive\_checks\_enabled  
performance\_data  
plugin\_output  
problem\_has\_been\_acknowledged  
process\_performance\_data  
retry\_interval  
scheduled\_downtime\_depth  
service\_description  
should\_be\_scheduled  
state\_type

## Acknowledgements

- **Logo/ graphical design web-interface:** Bastiaan Schuilings ([www.properlydecent.com](http://www.properlydecent.com))