




 **Centreon**

Poller States	Hosts	Up	Down	Unreachable	Pending	Services	Ok	Warning	Critical	Unknown	Pending
  	208	207	1	0	0	3652	3346	11/24	69/203	69/79	0














 Documentation - You are service.desk  Logout

Monitoring Views Reporting

Services | Hosts | Event Logs



Monitoring Services Details

2013/05/11 23:16

- >> By Status
-  Unhandled Problems
-  Service Problems
-  All Services
- >> By Host
-  Details
-  Summary
- >> By Host Group
-  Details
-  Summary
- >> By Service Group
1.  Details
2. - Problems
3. - Acknowledged
- Not Acknowledged
-  Summary
- >> Meta Services
-  Meta Services
- >> Nagios
-  Scheduling Queue
-  Downtime
-  Comments

Nagios®

CGI Configuration File Options

 Up To: Contents See Also: Configuration Overview, Information on the CGIs, Authentication And Authorization In The CGIs, CGI Footers and Headers

Notes

When creating and/or editing configuration files, keep the following in mind:

1. Lines that start with a '#' character are taken to be comments and are not processed
2. Variables names must begin at the start of the line - no white space is allowed before the name
3. Variable names are case-sensitive

Sample Configuration



Tip: A sample CGI configuration file (`/usr/local/nagios/etc/cgi.cfg`) is installed for you when you follow the quickstart installation guide.

Config File Location

By default, Nagios expects the CGI configuration file to be named **cgi.cfg** and located in the config file directory along with the main config file. If you need to change the name of the file or its location, you can configure Apache to pass an environment variable named `NAGIOS_CGI_CONFIG` (which points to the correct location) to the CGIs. See the Apache documentation for information on how to do this.

Configuration File Variables

Below you will find descriptions of each main Nagios configuration file option...

Main Configuration File Location

Format:	main_config_file=<file_name>
Example:	main_config_file=/usr/local/nagios/etc/nagios.cfg

This specifies the location of your main configuration file. The CGIs need to know where to find this file in order to get information about configuration information, current host and service status, etc.

Physical HTML Path

Format:	physical_html_path=<path>
Example:	physical_html_path=/usr/local/nagios/share

This is the *physical* path where the HTML files for Nagios are kept on your workstation or server. Nagios assumes that the documentation and images files (used by the CGIs) are stored in subdirectories called *docs/* and *images/*, respectively.

URL HTML Path

Format:	url_html_path=<path>
Example:	url_html_path=/nagios

If, when accessing Nagios via a web browser, you point to an URL like **http://www.myhost.com/nagios**, this value should be */nagios*. Basically, its the path portion of the URL that is used to access the Nagios HTML pages.

Authentication Usage

Format:	use_authentication=<0/1>
Example:	use_authentication=1

This option controls whether or not the CGIs will use the authentication and authorization functionality when determining what information and commands users have access to. I would strongly suggest that you use the authentication functionality for the CGIs. If you decide not to use authentication, make sure to remove the command CGI to prevent unauthorized users from issuing commands to Nagios. The CGI will not issue commands to Nagios if authentication is disabled, but I would suggest removing it altogether just to be on the safe side. More information on how to setup authentication and configure authorization for the CGIs can be found here.

- 0 = Don't use authentication functionality

- 1 = Use authentication and authorization functionality (default)

Default User Name

Format:	default_user_name=<username>
Example:	default_user_name=guest

Setting this variable will define a default username that can access the CGIs. This allows people within a secure domain (i.e., behind a firewall) to access the CGIs without necessarily having to authenticate to the web server. You may want to use this to avoid having to use basic authentication if you are not using a secure server, as basic authentication transmits passwords in clear text over the Internet.

Important: Do *not* define a default username unless you are running a secure web server and are sure that everyone who has access to the CGIs has been authenticated in some manner! If you define this variable, anyone who has not authenticated to the web server will inherit all rights you assign to this user!

System/Process Information Access

Format:	authorized_for_system_information=<user1>,<user2>,<user3>,...<user n>
Example:	authorized_for_system_information=nagiosadmin,theboss

This is a comma-delimited list of names of *authenticated users* who can view system/process information in the extended information CGI. Users in this list are *not* automatically authorized to issue system/process commands. If you want users to be able to issue system/process commands as well, you must add them to the `authorized_for_system_commands` variable. More information on how to setup authentication and configure authorization for the CGIs can be found here.

System/Process Command Access

Format:	authorized_for_system_commands=<user1>,<user2>,<user3>,...<user n>
Example:	authorized_for_system_commands=nagiosadmin

This is a comma-delimited list of names of *authenticated users* who can issue system/process commands via the command CGI. Users in this list are *not* automatically authorized to view system/process information. If you want users to be able to view system/process information as well, you must add them to the `authorized_for_system_information` variable. More information on how to setup authentication and configure authorization for the CGIs can be found here.

Configuration Information Access

Format:	authorized_for_configuration_information=<user1>,<user2>,<user3>,...<user n>
Example:	authorized_for_configuration_information=nagiosadmin

This is a comma-delimited list of names of *authenticated users* who can view configuration information in the configuration CGI. Users in this list can view information on all configured hosts, host groups, services, contacts, contact groups, time periods, and commands. More information on how to setup authentication and configure authorization for the CGIs can be found here.

Global Host Information Access

Format:	authorized_for_all_hosts=<user1>,<user2>,<user3>,...<user n>
Example:	authorized_for_all_hosts=nagiosadmin,theboss

This is a comma-delimited list of names of *authenticated users* who can view status and configuration information for all hosts. Users in this list are also automatically authorized to view information for all services. Users in this list are *not* automatically authorized to issue commands for all hosts or services. If you want users able to issue commands for all hosts and services as well, you must add them to the `authorized_for_all_host_commands` variable. More information on how to setup authentication and configure authorization for the CGIs can be found here.

Global Host Command Access

Format:	authorized_for_all_host_commands=<user1>,<user2>,<user3>,...<user n>
Example:	authorized_for_all_host_commands=nagiosadmin

This is a comma-delimited list of names of *authenticated users* who can issue commands for all hosts via the command CGI. Users in this list are also automatically authorized to issue commands for all services. Users in this list are *not* automatically authorized to view status or configuration information for all hosts or services. If you want users able to view status and configuration information for all hosts and services as well, you must add them to the `authorized_for_all_hosts` variable. More information on how to setup authentication and configure authorization for the CGIs can be found here.

Global Service Information Access

Format:	authorized_for_all_services=<user1>,<user2>,<user3>,...<user n>
Example:	authorized_for_all_services=nagiosadmin,theboss

This is a comma-delimited list of names of *authenticated users* who can view status and configuration information for all services. Users in this list are *not* automatically authorized to view information for all hosts. Users in this list are *not* automatically authorized to issue commands for all services. If you want users able to issue commands for all services as well, you must add them to the `authorized_for_all_service_commands` variable. More information on how to setup authentication and configure authorization for the CGIs can be found here.

Global Service Command Access

Format:	authorized_for_all_service_commands=<user1>,<user2>,<user3>,...<user n>
Example:	authorized_for_all_service_commands=nagiosadmin

This is a comma-delimited list of names of *authenticated users* who can issue commands for all services via the

command CGI. Users in this list are *not* automatically authorized to issue commands for all hosts. Users in this list are *not* automatically authorized to view status or configuration information for all hosts. If you want users able to view status and configuration information for all services as well, you must add them to the `authorized_for_all_services` variable. More information on how to setup authentication and configure authorization for the CGIs can be found [here](#).

Lock Author Names

Format: `lock_author_names=[0/1]`
Example: `lock_author_names=1`

This option allows you to restrict users from changing the author name when submitting comments, acknowledgements, and scheduled downtime from the web interface. If this option is enabled, users will be unable to change the author name associated with the command request.

- 0 = Allow users to change author names when submitting commands
- 1 = Prevent users from changing author names (default)

Statusmap CGI Background Image

Format: `statusmap_background_image=<image_file>`
Example: `statusmap_background_image=smbackground.gd2`

This option allows you to specify an image to be used as a background in the statusmap CGI if you use the user-supplied coordinates layout method. The background image is not be available in any other layout methods. It is assumed that the image resides in the HTML images path (i.e. `/usr/local/nagios/share/images`). This path is automatically determined by appending `/images` to the path specified by the `physical_html_path` directive. Note: The image file can be in GIF, JPEG, PNG, or GD2 format. However, GD2 format (preferably in uncompressed format) is recommended, as it will reduce the CPU load when the CGI generates the map image.

Default Statusmap Layout Method

Format: `default_statusmap_layout=<layout_number>`
Example: `default_statusmap_layout=4`

This option allows you to specify the default layout method used by the statusmap CGI. Valid options are:

<layout_number> Value	Layout Method
0	User-defined coordinates
1	Depth layers
2	Collapsed tree
3	Balanced tree
4	Circular
5	Circular (Marked Up)
6	Circular (Balloon)

Statuswrl CGI Include World

Format: `statuswrl_include=<vrml_file>`
Example: `statuswrl_include=myworld.wrl`

This option allows you to include your own objects in the generated VRML world. It is assumed that the file resides in the path specified by the `physical_html_path` directive. Note: This file must be a fully qualified VRML world (i.e. you can view it by itself in a VRML browser).

Default Statuswrl Layout Method

Format: `default_statuswrl_layout=<layout_number>`
Example: `default_statuswrl_layout=4`

This option allows you to specify the default layout method used by the statuswrl CGI. Valid options are:

<layout_number> Value	Layout Method
0	User-defined coordinates
2	Collapsed tree
3	Balanced tree
4	Circular

CGI Refresh Rate

Format: `refresh_rate=<rate_in_seconds>`
Example: `refresh_rate=90`

This option allows you to specify the number of seconds between page refreshes for the status, statusmap, and extinfo CGIs.

Audio Alerts

Formats: `host_unreachable_sound=<sound_file>`
`host_down_sound=<sound_file>`
`service_critical_sound=<sound_file>`
`service_warning_sound=<sound_file>`
`service_unknown_sound=<sound_file>`
Examples: `host_unreachable_sound=hostu.wav`
`host_down_sound=hostd.wav`
`service_critical_sound=critical.wav`
`service_warning_sound=warning.wav`
`service_unknown_sound=unknown.wav`

These options allow you to specify an audio file that should be played in your browser if there are problems when you

are viewing the status CGI. If there are problems, the audio file for the most critical type of problem will be played. The most critical type of problem is on or more unreachable hosts, while the least critical is one or more services in an unknown state (see the order in the example above). Audio files are assumed to be in the **media/** subdirectory in your HTML directory (i.e. `/usr/local/nagios/share/media`).

Ping Syntax

Format:	ping_syntax=<command>
Example:	ping_syntax=/bin/ping -n -U -c 5 \$HOSTADDRESS\$

This option determines what syntax should be used when attempting to ping a host from the WAP interface (using the statuswml CGI. You must include the full path to the ping binary, along with all required options. The \$HOSTADDRESS\$ macro is substituted with the address of the host before the command is executed.

Escape HTML Tags Option

Format:	escape_html_tags=[0/1]
Example:	escape_html_tags=1

This option determines whether or not HTML tags in host and service (plugin) output is escaped in the CGIs. If you enable this option, your plugin output will not be able to contain clickable hyperlinks.

Notes URL Target

Format:	notes_url_target=[target]
Example:	notes_url_target= _blank

This option determines the name of the frame target that notes URLs should be displayed in. Valid options include `_blank`, `_self`, `_top`, `_parent`, or any other valid target name.

Action URL Target

Format:	action_url_target=[target]
Example:	action_url_target= _blank

This option determines the name of the frame target that action URLs should be displayed in. Valid options include `_blank`, `_self`, `_top`, `_parent`, or any other valid target name.

Splunk Integration Option

Format:	enable_splunk_integration=[0/1]
Example:	enable_splunk_integration=1

This option determines whether integration functionality with Splunk is enabled in the web interface. If enabled, you'll be presented with "Splunk It" links in various places in the CGIs (log file, alert history, host/service detail, etc). Useful if you're trying to research why a particular problem occurred. For more information on Splunk, visit <http://www.splunk.com/>.

Splunk URL

Format:	splunk_url=<path>
Example:	splunk_url=http://127.0.0.1:8000/

This option is used to define the base URL to your Splunk interface. This URL is used by the CGIs when creating links if the `enable_splunk_integration` option is enabled.