Advanced Configuration

Performance data parameters

Performance Data Processing Option

Format:

process_performance_data=<0/1>

Example:

process_performance_data=1

This value determines if Shinken Enterprise will process host/check performance data.

- 0 = Don't process performance data
- 1 = Process performance data (default)

Advanced scheduling parameters

Orphaned Host/Service Check Option

Format:

check_for_orphaned_services=<0/1>
check_for_orphaned_hosts=<0/1>

Example:

check_for_orphaned_services=1
check_for_orphaned_hosts=1

This option allows you to enable or disable checks for orphaned host/check. Orphaned checks are checks which have been launched to pollers but did not get any results reported for a long time.

Since there is no results back, it is not rescheduled in the event queue. This can cause checks to stop being executed. Normally it should happen very rarely - it might happen if an external user or a process killed off the process that was being used to execute a check.

If this option is enabled and Shinken Enterprise does not find results back for a particular check, it will log an error message and reschedule the check. If you see checks that never seem to get rescheduled, enable this option and see if you notice any log messages about orphaned services.

- 0 = Don't check for orphaned service checks
- 1 = Check for orphaned service checks (default)

Performance tuning

Cached Host/Service Check Horizon

Format:

cached_host_check_horizon=<seconds>
cached_service_check_horizon=<seconds>

Example:

cached_host_check_horizon=0
cached_service_check_horizon=0

This option determines the maximum time (in seconds) the state of a previous host check is considered current. Cached host states (from host/check that were performed more recently than the time specified by this value) can hugely improve host check performance. If the value is too high for this option it may result in (temporarily) inaccurate host/check states. A low value may result in a performance hit for host/check. Use a value of 0 if you want to disable host/check caching.

Flapping parameters

Flap Detection Option

Format:

enable_flap_detection=<0/1>

Example:

enable_flap_detection=1

This option determines whether or not Shinken Enterprise will try and detect hosts and checks that are “flapping". Flapping occurs when a host or check changes between states too frequently, resulting in a barrage of notifications being sent out. When Shinken Enterprise detects a flapping host or check, it will temporarily suppress notifications for that host/check until it stops flapping.

More information on how flap detection and handling works can be found here <advanced/flapping>`.

- 0 = Don't enable flap detection (default)
- 1 = Enable flap detection

Low Service/Host Flap Threshold

Format:

low_service_flap_threshold=<percent>
low_host_flap_threshold=<percent>

Example:

low_service_flap_threshold=25.0
low_host_flap_threshold=25.0

This option is used to set the low threshold for detection of host/check flapping. For more information on how flap detection and handling works (and how this option affects things) read this.

High Service/Host Flap Threshold

Format:

high_service_flap_threshold=<percent>
high_host_flap_threshold=<percent>

Example:

high_service_flap_threshold=50.0
high_host_flap_threshold=50.0

This option is used to set the high threshold for detection of host/check flapping. For more information on how flap detection and handling works (and how this option affects things) read this.

Various commands Timeouts

Format:

event_handler_timeout=<seconds> # default: 30s
notification_timeout=<seconds> # default: 30s

Example:

event_handler_timeout=60
notification_timeout=60

This is the maximum number of seconds to run event handlers & notification. If a command exceeds this time limit it will be killed and a warning will be logged.

There is often widespread confusion as to what this option really does. It is meant to be used as a last ditch mechanism to kill off commands which are misbehaving and not exiting in a timely manner. It should be set to something high (like 60 seconds or more for notification), so that each event handler command normally finishes executing within this time limit. If an event handler runs longer than this limit, Shinken Enterprise will kill it off thinking it is a runaway process.

Freshness check

Host/Service Freshness Checking Option

Format:

check_service_freshness=<0/1>
check_host_freshness=<0/1>

Example:

check_service_freshness=0
check_host_freshness=0

This option determines if Shinken Enterprise will periodically check the “freshness" of host/check. Enabling this option is useful for helping to ensure that passive checks are received in a timely manner. More information on freshness checking can be found here.

- 0 = Don't check host/check freshness
- 1 = Check host/check freshness (default)

Host/Service Freshness Check Interval

Format:

service_freshness_check_interval=<seconds>
host_freshness_check_interval=<seconds>

Example:

service_freshness_check_interval=60
host_freshness_check_interval=60

This setting determines how often (in seconds) Shinken will periodically check the “freshness" of host/check results. If you have disabled host/check freshness checking (with the check_service_freshness option), this option has no effect. More information on freshness checking can be found here.

Notifications Option

Format:

enable_notifications=<0/1>

Example:

enable_notifications=1

This option determines if Shinken Enterprise will send out notifications. If this option is disabled, Shinken will not send out notifications for any host or check.

Values are as follows:

- 0 = Disable notifications
- 1 = Enable notifications (default)

External Command Check Option

Format:

check_external_commands=<0/1>

Example:

check_external_commands=1

This option determines if Shinken Enterprise will check the External Command File for commands that should be executed with the arbiter daemon .

- 0 = Don't check external commands
- 1 = Check external commands (default)

Scheduling parameters

Service/Host Check Execution Option

Format:

execute_service_checks=<0/1>
execute_host_checks=<0/1>

Example:

execute_service_checks=1
execute_host_checks=1

This option determines if Shinken Enterprise will execute host/check. Do not change this option unless you use a old school distributed architecture. And even if you do this, please change your architecture with a cool new one far more efficient.

- 0 = Don't execute checks
- 1 = Execute checks (default)

Passive Service/Host Check Acceptance Option

Format:

accept_passive_service_checks=<0/1>
accept_passive_host_checks=<0/1>

Example:

accept_passive_service_checks=1
accept_passive_host_checks=1

This option determines if Shinken will accept passive host/checks. If this option is disabled, Shinken Enterprise will not accept any passive host /checks.

- 0 = Don't accept passive host/checks
- 1 = Accept passive host/checks (default)

Event Handler Option

Format:

enable_event_handlers=<0/1>

Example:

enable_event_handlers=1

This option determines if Shinken Enterprise will run event handlers.

- 0 = Disable event handlers
- 1 = Enable event handlers (default)

Notification Logging Option

Format:

log_notifications=<0/1>

Example:

log_notifications=1

This variable determines if notification messages are logged. If you have a lot of contacts or regular check failures your log file will grow (let say some Mo by day for a huge configuration, so it's quite OK for nearly every one to log them). Use this option to keep contact notifications from being logged.

- 0 = Don't log notifications
- 1 = Log notifications

Event Handler Logging Option

Format:

log_event_handlers=<0/1>

Example:

log_event_handlers=1

This variable determines if check and host event handlers are logged. Event handlers are optional commands that can be run whenever a check or hosts changes state. Logging event handlers is most useful when debugging Shinken Enterprise or first trying out your event handler scripts.

- 0 = Don't log event handlers
- 1 = Log event handlers

External Command Logging Option

Format:

log_external_commands=<0/1>

Example:

log_external_commands=1

This variable determines if Shinken Enterprise will log external commands that it receives.

- 0 = Don't log external commands
- 1 = Log external commands (default)

Passive Check Logging Option (Not implemented)

Format:

log_passive_checks=<0/1>

Example:

log_passive_checks=1

This variable determines if Shinken Enterprise will log passive host/checks that it receives from the external commands.

- 0 = Don't log passive checks
- 1 = Log passive checks (default)

Timing Interval Length

Format:

interval_length=<seconds>

Example:

interval_length=60

This is the number of seconds per “unit interval" used for timing in the scheduling queue, re-notifications, etc. "Units intervals" are used in the object configuration file to determine how often to run a service check, how often to re-notify a contact, etc.

The default value for this is set to 60, which means that a "unit value" of 1 in the object configuration file will mean 60 seconds (1 minute).

Note that set this option top 1 is not a good thing with Shinken Enterprise. It's not design to be a hard real time (<5seconds) monitoring system.

Others

Illegal Object Name Characters

Format:

illegal_object_name_chars=<chars...>

Example:

illegal_object_name_chars=`-!$%^&*"\|'<>?,()=

This option allows you to specify illegal characters that cannot be used in host names, service descriptions, or names of other object types. Shinken Enterprise will allow you to use most characters in object definitions, but we recommend not using the characters shown in the example above. Doing may give you problems in the web interface, notification commands, etc.