

# configuration file (shinken.cfg)

## Configuration Options

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

- Lines that start with a "#" character are considered to be comments and are not processed
- Variable names are case-sensitive

The main configuration file is "shinken.cfg". It is located in the "/etc/shinken/" directory.

Below are listed parameters currently used in the file. For other parameters (not set by default) see [Advanced Configuration](#).

## Default used options

### Cfg dir and Cfg files

Format :

```
cfg_dir=<directory_name>  
cfg_file=<file_name>
```

Those are statements and not parameters . The arbiter considers them as order to open other(s) configuration(s) file(s)  
For the `cfg_dir` one, the arbiter only reads files ending with ".cfg".  
The arbiter does read recursively directory for files but does not consider lines into those files as statements anymore.

This means that a `cfg_dir` or `cfg_file` is considered as a parameter outside of `shinken.cfg` (or any configuration file directly given to the arbiter as parameter in a command line)  
The arbiter handles main configuration files differently than any other files.

### Automatic State Retention Update Interval

Format:

```
retention_update_interval=<minutes>
```

Default:

```
retention_update_interval=60
```

This setting determines the interval (in minutes) between automatic retention data backups by the scheduler during normal operation.  
If you set this value to 0, it will not save retention data at regular intervals, but it will still save retention data before shutting down or restarting.

### Maximum Host/Service Check Spread

Format:

```
max_service_check_spread=<minutes>  
max_host_check_spread=<minutes>
```

Default:

```
max_service_check_spread=30  
max_host_check_spread=30
```

This option determines the maximum time spread (in minutes) to check all hosts/checks (which are scheduled to be regularly checked) from Shinken Enterprise's launch. This option will ensure that the initial checks of all hosts/checks occur within the timeframe you specify. Default value is 30 (minutes).

## Service/Host Check Timeout

Format:

```
service_check_timeout=<seconds>  
host_check_timeout=<seconds>
```

Default:

```
service_check_timeout=60  
host_check_timeout=30
```

This is the maximum number of seconds allowed by Shinken Enterprise to run host/checks. If checks exceed this limit, they are killed and a CRITICAL state is returned. A timeout error will also 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 plugins which are misbehaving and not exiting in a timely manner. It should be set to something high (like 60 seconds or more), so that each check normally finishes executing within this time limit. If a check runs longer than this limit, Shinken Enterprise will kill it off thinking it is a runaway process.

## Timeout Exit Status

Format:

```
timeout_exit_status=[0,1,2,3]
```

Default:

```
timeout_exit_status=2
```

State set by Shinken Enterprise in case of timeout.

## Flap History

Format:

```
flap_history=<int>
```

Default:

```
flap_history=20
```

This option is used to set the history size of states kept by the scheduler to make the flapping calculation. By default, the value is 20 states kept.

The size in memory is for the scheduler daemon : 4Bytes flap\_history (nb hosts + nb checks). For a big environment, it costs 4 20 (1000+10000) - 900Ko. So you can raise it to higher value if you want.

## Max Plugins Output Length

Format:

```
max_plugins_output_length=<int>
```

Default:

```
max_plugins_output_length=8192
```

This option is used to set the max size in bytes for the checks plugins output. So if you saw truncated output like for huge disk check when you have a lot of partitions, raise this value.

## Enable problem/impacts states change

Format:

***enable\_problem\_impacts\_states\_change=<0/1>***

Default:

***enable\_problem\_impacts\_states\_change=0***

This option is used to know if we apply or not the state change when a host or check is impacted by a root problem (like the check's host going down or a host's parent being down too). The state will be changed by UNKNOWN for a check and UNREACHABLE for a host until their next schedule check. This state change does not count as an attempt, it's just for console so the user knows that these objects got problems and the previous states are not sure.

## Disable Old Nagios Parameters Whining

Format:

***disable\_old\_nagios\_parameters\_whining=<0/1>***

Default:

***disable\_old\_nagios\_parameters\_whining=0***

If 1, disable all notice and warning messages at configuration checking

## Timezone Option

Format:

***use\_timezone=<tz from tz database>***

Default:

***use\_timezone=""***

This option allows you to override the default timezone in which an instance of Shinken Enterprise runs. Useful if you have multiple instances of Shinken Enterprise which need to run from the same server, but have different local times associated with them. If not specified, Shinken Enterprise will use the system configured timezone.

## Environment Macros Option

Format:

***enable\_environment\_macros=<0/1>***

Default:

***enable\_environment\_macros=0***

This option determines if the Shinken Enterprise daemon will make all standard `:ref:macros <thebasics/macrolist>` available as environment variables to your check, notification, event handler, etc. commands. In large installations this can be problematic because it takes additional CPU to compute the values of all macros and make them available to the environment. It also cost a increase network communication between schedulers and pollers.

- 0 = Don't make macros available as environment variables
- 1 = Make macros available as environment variables

## Initial States Logging Option

Format:

***log\_initial\_states=<0/1>***

Default:

***log\_initial\_states=1***

This variable determines if Shinken Enterprise will force all initial host and check states to be logged, even if they result in an OK state. Initial check and host states are normally only logged when there is a problem on the first check. Enabling this option is useful if you are using an application that scans the log file to determine long-term state statistics for checks and hosts.

- 0 = Don't log initial states
- 1 = Log initial states

## Event Handler during downtimes

Format:

***no\_event\_handlers\_during\_downtimes=<0/1>***

Default:

***no\_event\_handlers\_during\_downtimes=1***

This option determines if Shinken Enterprise will run `:ref: event handlers <advanced/eventhandlers>` when the host or check is in a scheduled downtime.

- 0 = Launch event handlers (Nagios behavior)
- 1 = Don't launch event handlers

## Arbiter daemon part

The following parameters are common to all daemons.

### Workdir

Format:

***workdir=<directory>***

Default :

***workdir=/var/run/shinken/***

This variable specify the working directory of the daemon.  
In the arbiter case, if the value is empty, the directory name of `lock_file` parameter. See below

### Arbiter Lock File

Format:

***lock\_file=<file\_name>***

Example:

***lock\_file=/var/lib/shinken/arbiterd.pid***

This option specifies the location of the lock file that arbiter daemon should create when it runs as a daemon (when started with the "-d" command line argument). This file contains the process id (PID) number of the running arbiter process.

## Local Log

Format:

***local\_log=<filename>***

Default:

***local\_log=/var/log/shinken/arbiterd.log'***

This variable specifies the log file for the daemon.

## Log Level

Format:

***log\_level=[DEBUG,INFO,WARNING,ERROR,CRITICAL]***

Default:

***log\_level=WARNING***

This variable specifies which logs will be raised by the arbiter daemon. For others daemons, it can be defined in their local \ d.ini files.

## Arbiter Daemon User

Defined in brokerd.ini, brokerd-windows.ini, pollerd.ini, pollerd-windows.ini, reactionnerd.ini, schedulerd.ini and schedulerd-windows.ini.

Format:

***shinken\_user=username***

Default:

***shinken\_user=<current user>***

This is used to set the effective user. The Arbiter process (main process) will run as this effective user. After initial program startup, Shinken Enterprise will drop its effective privileges and run as this user.

## Arbiter Daemon user Group

Format:

***shinken\_group=groupname***

Default:

***shinken\_group=<current group>***

This is used to set the effective group of the user used to launch the arbiter daemon.

## Modules directory

Format:

***modules\_dir=<dirname>***

Default:

***modules\_dir=/var/lib/shinken/modules***

Path to the modules directory

## Daemon Enabled

Format:

***daemon\_enabled=[0/1]***

Default:

***daemon\_enabled=1***

Set to 0 if you want to make this daemon (arbiter) NOT to run

## Use SSL

Format:

***use\_ssl=[0/1]***

Default:

***use\_ssl=0***

Use SSI or not. You have to enable it on other daemons too.

## Ca Cert

Format:

***ca\_cert=<filename>***

Default:

***ca\_cert=etc/certs/ca.pem***

Certification Authority (CA) certificate. Put full paths for certs

## Server Cert

Format:

***server\_cert=<filename>***

Default:

***server\_cert=etc/certs/server.cert***

Server certificate for SSL. Put full paths for certs

## Server Key

Format:

***server\_key=<filename>***

Default:

***server\_key=/etc/certs/server.key***

Server key for SSL. Put full paths for certs

## Hard SSL Name Check

Format:

***hard\_ssl\_name\_check=[0/1]***

Default:

***hard\_ssl\_name\_check=0***

Enable SSL name check.

## HTTP Backend

Format:

***http\_backend=[auto, cherrypy, swsgiref]***

Default:

***http\_backend=cherrypy***

Specify which http\_backend to use.