

NAME

zupsd.conf, zupsd-hosts – ZEDAT UPS Daemon configuration

SYNOPSIS

/usr/local/etc/zupsd.conf

/usr/local/etc/zupsd-hosts

DESCRIPTION

zupsd(8) reads configuration data from */usr/local/etc/zupsd.conf* or the file specified with **-f** on the command line. See the CONFIGURATION FILE section for a description of the configuration options and the format of this file.

The list of remote hosts (or other strings optionally used within shutdown commands) and their corresponding shutdown command delays is read from */usr/local/etc/zupsd-hosts* or the file specified via the *hostsfile* keyword in the configuration file (see below). See the HOSTS FILE section for a description of the format of this file.

In both files, empty lines and the rest of a line following a hash mark (“#”) will be ignored, unless the “#” character is specified as a literal as described in the respective sections below. If the last character of a line is a backslash (“\”), the following line will be appended prior to parsing.

CONFIGURATION FILE

The configuration file is split into sections, each section beginning with a [*module*] statement (on a line of it’s own), where *module* is one of the module type names *general*, *ups* or *plugin*, described by the respective subsections below. A section ends if a new section begins or the file ends. The configuration file must begin with a [*general*] section, which may only be specified once. All other module types may be used repeatedly in order to configure multiple checks (e.g., for multiple UPS devices). At least one module must be specified next to the [*general*] section.

Each [*module*] statement is followed by one or more module-specific *keyword=value* pairs, one per line, described within the respective module subsections below. Keywords can be of type “string”, “integer” or “bool”. The type of each keyword is denoted below and the default value is specified in brackets. Regardless of the type, the keyword value may either be quoted with single (“”) or double (“”) quote characters (which makes no difference), or not quoted at all. If a string is quoted, the quote character can be escaped with a backslash in order to be read as a literal instead of an interpreted character, which would end the quoted string. If a string is not quoted, spaces, tabs and “#” characters must also be escaped by using a backslash. A literal backslash must be escaped with another backslash in any case. Integers must be specified as positive, decimal values, boolean values as 0 (for “false”) or 1 (for “true”).

Spaces and tabs preceding or following module type names, keywords or values are ignored. The module type names and keywords are case insensitive.

[general]

Within the [*general*] section, the overall **zupsd** operation is configured.

The following keywords are recognized:

check_interval = <*integer*> [30]

The interval between the executions of a given module check in seconds. Note that if a module check takes longer than the specified number of seconds, the following checks will be deferred. If *check_interval* is set to 0, the default value of 30 will be used.

commit_suicide = <*bool*> [0]

After a shutdown process was executed, **zupsd(8)** will terminate if this keyword is set to 1 and continue to run if it is set to 0 (which is the default). In the latter case, a new shutdown process will not be initiated before the number of modules reporting a

non-CRITICAL and non-ALERT status has fallen beneath the threshold configured via the *num_critical* keyword (see below).

debug_level = <integer> [0]

If set to a value greater than 0, additional messages that contain information normally of use only when debugging **zupsd**(8) will be generated. Additionally, if set to a value greater than 1, the source code file and line number will be prepended to each message (this is not supported on all platforms though). The specified value will be overridden if **zupsd**(8) is called with the **-d** option.

hostsfile = <string> ["/usr/local/etc/zupsd-hosts"]

The hosts file which is read by **zupsd**(8) if a shutdown process is invoked (see the HOSTS FILE section).

max_shutdowns = <integer> [0]

If the specified number of parallel shutdown command executions is reached, **zupsd**(8) will wait for a shutdown command to complete before issuing the next shutdown command. Note that *max_shutdowns* will be ignored in case of an "alert" state, in which all shutdown commands will be executed immediately (see the *num_alert* keyword). If *max_shutdowns* is set to 0 (which is the default), the number of parallel shutdown command executions is not limited by **zupsd**(8).

num_alert = <integer> [0]

If (at least) the specified number of modules report an ALERT condition, **zupsd**(8) will ignore both the specified *max_shutdowns* value and the *delay* field specified within the hosts file (see the HOSTS FILE section), which means that all shutdown commands will be run immediately. If *num_alert* is set to 0 (which is the default), this will happen only if *all* configured modules report an ALERT condition. If *num_alert* is set to a number greater than the number of configured modules, this will never happen (unless **zupsd**(8) receives an USR2 signal).

num_critical = <integer> [0]

If (at least) the specified number of modules report a CRITICAL or ALERT condition, **zupsd**(8) will initiate a shutdown process which executes the shutdown commands according to the specified *shutdown_cmd* string and the hosts file (see below). If *num_critical* is set to 0 (which is the default), this will happen only if *all* configured modules report a CRITICAL condition. If *num_critical* is set to a number greater than the number of configured modules, this will never happen (unless **zupsd**(8) receives an USR1 signal). The shutdown process will be terminated as soon as all configured shutdown commands are executed or the number of modules which report a CRITICAL or ALERT condition falls below the specified number (or **zupsd**(8) receives an USR1 signal).

pidfile = <string> ["/var/run/zupsd.pid"]

Upon startup, **zupsd**(8) writes its process ID to the specified file.

shutdown_cmd = <string> [-]

If a shutdown process is initiated (see the *num_critical* keyword), the specified shutdown command will be executed once for each entry in the hosts file (see the HOSTS FILE section). This keyword is mandatory (though it may be set to, for example, "/usr/bin/true", if for some reason no actual shutdown command should be executed). If the character sequence "%h" is used within the specified string, it will be replaced by the string specified in the *hosts* field of the according hosts file entry. If the character sequence "%t" is used within the specified string, it will be replaced by the string specified in the *type* field of the according hosts file entry. Note that these character sequences cannot be escaped. The resulting string will be handed over to the system shell for execution after the delay specified via the third field of the according hosts file entry has elapsed, unless some "alert" condition is encountered (see the *num_alert* keyword).

suicide_cmd = <string> [-]

If this keyword is set, the specified string will be handed over to the system shell for execution after all other shutdown commands of a running shutdown process have been executed.

syslog_facility = <string> ["DAEMON"]

The **syslog(3)** facility **zupsd(8)** will use for logging. One out of "DAEMON", "USER" or "LOCAL0" through "LOCAL7" may be specified (case insensitive). Note that **zupsd(8)** will use the "DAEMON" facility in any case for logging errors which occur prior to or while parsing the configuration file upon startup.

[ups]

Any RFC 1628 (UPS MIB) compliant UPS device can be monitored via SNMP using an *[ups]* module. This module returns a CRITICAL or ALERT status (only) if the UPS device reports that it is running on battery (see the *alert_threshold* keyword for details). A WARNING status is returned if some other unusual UPS state is encountered and an UNKNOWN status if the SNMP communication fails or an internal error occurs. Otherwise, an OK status is returned.

All of the *[ups]*-specific keywords may also be specified within the *[general]* section. A keyword value specified within the *[general]* section will be used if the same keyword is not specified within the *[ups]* module. The latter would override the former.

The following keywords are recognized:

alert_threshold = <integer> [0]

If the monitored UPS device reports to be running on battery *and* to have a remaining battery time less than or equal to the number of minutes specified via *alert_threshold*, the module will report an ALERT status. If set to 0 (which is the default), the module will never report an ALERT status.

community = <string> ["public"]

Specifies the community string required for (read-)access to the monitored SNMP agent.

email_cmd = <string> [-]

The specified string will be handed over to the system shell for execution if the module state changes and **zupsd(8)** was configured to call the *email_cmd* in this case (see the *email_infolevel* keyword for details on when it will be executed). If the character sequence "%m" is used within the specified string, it will be replaced by a (short) message naming and describing the new module state. If the character sequence "%r" is used within the specified string, it will be replaced by the string specified via *email_contact*. Note that these character sequences cannot be escaped.

email_contact = <string> [-]

The specified string can be used for substitution within the *email_cmd* (see the *email_cmd* keyword for details). This keyword will be ignored if no *email_cmd* is specified.

email_infolevel = <integer> [4]

Determines the circumstances under which the command specified via the *email_cmd* keyword will be executed. If set to 0, it will never be executed. If set to 1, it will be executed when the module initially goes into an ALERT state. If set to 2, it will additionally be executed when the module initially goes into a CRITICAL state. If set to 3, it will additionally be executed when the module initially goes into a WARNING state. If set to 4 (which is the default), it will additionally be executed when the module initially goes into an UNKNOWN state. If the module recovers to an OK state after an *email_cmd* was executed, another *email_cmd* will be executed in order to report the recovery. This keyword will be ignored if no *email_cmd* is specified.

hostname = <string> [-]

Specifies the hostname or IP address of the monitored SNMP agent. Multiple hosts may be specified as a comma-separated string, the additional agents will be used as fallback in case of connection problems or SNMP errors. A port number (or service name) may optionally be appended, separated from the host(s) by a colon. This keyword is mandatory for each *[ups]* module.

pager_cmd = <string> [-]

The specified string will be handed over to the system shell for execution if the module state changes and **zupsd**(8) was configured to call the *pager_cmd* in this case (see the *pager_infolevel* keyword for details on when it will be executed). If the character sequence “%m” is used within the specified string, it will be replaced by a message naming and describing the new module state. If the character sequence “%r” is used within the specified string, it will be replaced by the string specified via *pager_contact*. Note that these character sequences cannot be escaped.

pager_contact = <string> [-]

The specified string can be used for substitution within the *pager_cmd* (see the *pager_cmd* keyword for details). This keyword will be ignored if no *pager_cmd* is specified.

pager_infolevel = <integer> [2]

Determines the circumstances under which the command specified via the *pager_cmd* keyword will be executed. If set to 0, it will never be executed. If set to 1, it will be executed when the module initially goes into an ALERT state. If set to 2 (which is the default), it will additionally be executed when the module initially goes into a CRITICAL state. If set to 3, it will additionally be executed when the module initially goes into a WARNING state. If set to 4, it will additionally be executed when the module initially goes into an UNKNOWN state. If the module recovers to an OK state after an *pager_cmd* was executed, another *pager_cmd* will be executed in order to report the recovery. This keyword will be ignored if no *pager_cmd* is specified.

retries = <integer> [5]

The number of retries if the SNMP connection fails.

timeout = <integer> [1000]

The number of milliseconds until the first SNMP connection timeout. For retries, **Net-SNMP** will increase this value. 1000 (which is the default) gives one second.

[plugin]

The *[plugin]* module allows for running arbitrary external check commands.

All of the *[plugin]*-specific keywords may also be specified within the *[general]* section. A keyword value specified within the *[general]* section will be used if the same keyword is not specified within the *[plugin]* module. The latter would override the former.

The following keywords are recognized:

check_cmd = <string> [-]

The external check command which should be run by **zupsd**(8). This keyword is mandatory for each *[plugin]* module. The specified string will be handed over to the system shell for execution. The executed command must return one of the following exit codes in order to signal the module status to **zupsd**(8):

- 0 – OK
- 1 – UNKNOWN
- 2 – WARNING
- 3 – CRITICAL

- 4 – ALERT

Any other exit code will result in an UNKNOWN status and an error being logged.

test_cmd = <string> [-]

The external command which is executed instead of the command specified via *check_cmd* if **zupsd**(8) was called with the **-t** option. If this keyword is not set, the command specified via *check_cmd* will be used. Apart from that, the same notes apply as for the *check_cmd* keyword.

hostname = <string> [-]

Any occurrence of the character sequence “%h” within the *check_cmd* and *test_cmd* values will be replaced by the specified string. Note that within the *check_cmd* and *test_cmd* values, this character sequence cannot be escaped.

HOSTS FILE

The hosts file contains a table with three columns separated by one or more spaces and/or tabs. Spaces and tabs preceding the first and following the last field are ignored. One row must be specified for each shutdown command which should be executed during a shutdown process. From left to right, the following fields must be specified in each row:

host A string. If the character sequence “%h” is used within the *shutdown_cmd* value (see the CONFIGURATION FILE section), it will be replaced by *host*. Otherwise, *host* may be set to an arbitrary “dummy” string. For example, this field could be used for specifying the name or IP address of a remote host.

type A string. If the character sequence “%t” is used within the *shutdown_cmd* value (see the CONFIGURATION FILE section), it will be replaced by *type*. Otherwise, *type* may be set to an arbitrary “dummy” string. For example, this field could be used for specifying the operating system of a remote host.

delay A positive integer which specifies the shutdown delay in minutes. The shutdown command for the current row will be executed (not earlier than) *delay* minutes after a shutdown process was initiated. Note that depending on the configuration, this delay will be ignored if some “alert” condition is encountered (see the *num_alert* keyword within the CONFIGURATION FILE section).

Note that the specified strings cannot be quoted. A literal “#” can be specified within or as the last, but not as the first character of a string. Apart from that, all characters (including quote signs and backslashes) may be specified as literals as far as **zupsd**(8) is concerned, but they might be interpreted by the system shell when the shutdown command is executed (see the CONFIGURATION FILE section).

CAVEATS

The hosts file is read each time a shutdown process is initiated, but not upon **zupsd**(8) startup, unless the **-t** option was used on the command line. This behaviour might change in a future release of **zupsd**.

SEE ALSO

snmp.conf(5), **zupsd**(8)

AUTHOR

Written by Holger Weiß <holger@ZEDAT.FU-Berlin.DE>.

COPYRIGHT

Copyright © 2005, 2006, 2007, 2009, 2011 Freie Universität Berlin. All rights reserved.