NAME

smd is a snapshot manager and MPEG−TS stream analyzer.

SYNOPSIS

smd PARAMETERS [OPTIONS]

DESCRIPTION

smd(1) *(sa−me−di*) is a tool serving two main purposes: 1) capturing extracts of a stream; 2) analyzing stream in real time and issuing alerts based on the analysis.

smd(1) continuously reads a live video stream, caching N seconds/Mb worth of it in memory, maintaining a *rolling window* that can be saved to a persistent storage when needed. This allows one to capture a stream segment (of interest) for further (offline) analysis. Such an extract will be hereinafter referred to as 'snapshot'. **smd**(1) takes a snapshot when it receives a **SIGUSR2** signal, either saving the 'window' as a *snapshot file* at once or pausing for a while to accumulate data around the area of interest.

Another function of **smd** is to analyze the stream (in real time) and issue alerts based on various conditions, such as: **audio/video synchronization** deviations, **PCR record frequency**, **PCR sync with PTS/DTS**. Alerts get captured in the application log as well as *alert log*, with a well-defined structure, easy to process by text parsing tools.

PARAMETERS

The following parameters are accepted:

−i|−−source {URL}

URL for the stream to read from (in ffmpeg−compliant format). **smd**(1) accepts tw o protocol types in the URL: **UDP** and **http**.

Example 1: *http://acme.tv:8080/travel-channel*

Example 2: *udp://226.0.3.44:5656*

Accepted stream formats are: *MPEG−TS* and *RTP* over MPEG-TS.

OPTIONS

The following options are accepted:

−Z, −−demo

allows to run a single instance of **smd** without a license. If another instance is started, the application will exit. Running without this parameter would have **smd** check for a valid GigA+ or GigaTools license.

−N, −−nid *num* **[0]**

specifies a positive *Numeric ID* (NID) for the source. The NID could be used as a macro specifying log path(s). See −l option for details.

−C, −−config *path*

path to the configuration file. All options could be specified in a file−based configuration, but command−line options always override whatever is specified in a config file. **smd** would always try to load *smd.conf* from the current directory, then from */etc*. If no such file exists, *smd* will proceed without one, using default values.

−l, −−logfile *path*
path to the *application log* file. Please do mind access permissons for the directory. The**NID** (numeric id from −N option) could be put into the file path as {N}. For instance, with **−N 2 -l mylog{N}.log** would translate the log path to: mylog2.log. Only one substitution is performed.

−L, −−level *crit|err|norm|warn|info|debug*
log level. The default level is *info*.

−E, −−cachesz *mb* **[8 Mb]**
cache size in megabytes. This is the size of the 'rolling window' kept in memory. Please bear in mind that this will hold different durations, depending on the stream's bit rate.

−t, −−tail {mb}:{ms} [2:500]
size/duration of the 'tail' to save after a snapshot signal. Let's say that we have an event at time T that warrants taking a snapshot. If we sent the signal at time T, however, very little post−event data would be saved. It might be reasonable to wait a little and have the data pertaining to the event be in the middle of the snapshot. This option defines how much we want to save *past* the event. **smd** notices time T and starts counting, both in (real) time and in bytes. As soon as either threshold is hit (for instance, 2.1 Mb received or 501 ms passed), a snapshot gets saved.

NB: As noted before, the time parameter is *real/clock* time, which is **not** to be confused with the stream's own PCR/PTS/DTS time.

−m, −−mcastif {name} [all]
name of the multicast interface for the source stream.

−R, −−minread {nbytes} [4096]
number of bytes to free up in the cache for reading. **smd** will need to free up some data at the tail of the cache before reading more from the stream. This parameter defines how much is to be freed.

−D, −−snapdir {directory} [.]
directory for the snapshots. Snapshot files will be saved there. No subdirectories get created.

−x, −−snapfx {prefix} [snap]
prefix for snapshot files.

−S, −−srctag {tag} [{URL}]
alternate tag for the source stream (used by the analyzer). By default, analyzer would use the source URL to reference the stream. The URL might be too long or not self−explanatory, so this option is given to address this issue.

−k, −−track o1:o2:..oN []
directs analyzer to track additional data. This option takes a number of colon−separated parameters. Each parameter signifies a category of data the we want the analyzer to display. The parameters are as below: **sdt_i** − track *SDT* (Service Description Table) packets, ignored by analyzer; **pat_pmt** − track *PAT*/*PMT* packets, even after the analyzer selected either or both for the current stream; **pcr_u** − track PCR updates; **misc_i** − track non−specific ignored packets; **pdts_u** − track updates of *PTS*/*DTS*. Example: −−track pdts_u:sdt_i

−A, −−analyze param=value
supplies an analyzer−specific configuration parameter in the form of *parameter=value*, where parameter can be one of the following: **skipped_pkts**, **pdts_delta**, **pcr_delta**, **pcr_freq**. See **CONFIGURATION** below for details (in the **stream_alalyzer** section). The option should be specified multiple times for multiple parameters. Example: −A skipped_pkts=1800 −A pcr_freq=800.

−H, −−http10
use HTTP/1.0 in **HTTP GET** requests in order to avoid chunked output. Analyzer does **not** parse chunked input and, if encountered, the **smd** would exit with **BAD stream − stype=TS expected** error.

−T, −−term
run as a terminal (non-daemon) application. This is the default behavior when run by a non−privileged user. −T could be specified when run as root in order **NOT** to become a daemon, for instance, for debugging purposes.

−K, −−syskey
generate system key (to use in licensing) and exit.

−V, −−version
output application's version and quit.

−h, −−help, −?, −−options
output brief option guide. This is output when run without parameters.

CONFIGURATION

smd(1) uses a configuration file, the path to it can be specified at the command line. The module starts by initializing config settings to default values, then it loads settings from the config file (overriding the defaults). Config values will be then updated by whatever is specified from the command line.

smd.conf is supplied as an example configuration.

All config settings have **smd.** prefix. For instance, *A* (setting) fully reads in the config as *smd.A*. The settings are as below:

log.*
log file settings:

file path to the application log file. {N} substitution applies (see −l option).

level log level: *crit*|*err*|*norm*|*warn*|*info*|*debug*

max_size_mb = *N*
maximum log size in MB (1MB = 1048576). Every time a log file reaches above this size it is renamed as an archive and a fresh log is started.

max_files = *n*
maximum number of log files before rotation. Every time the number of archive logs exceeds this number, the oldest archive is removed.

eod_rotate = true|false [false]
Rotate log at the end of the day (each day). This means that a new log would be started every new day.

alert_log.*
analyzer's alert log file settings:

file path to the log file. {N} substitution applies (see −l option).

max_size_mb = *N*
maximum log size in MB (1MB = 1048576). Every time a log file reaches above this size it is renamed as an archive and a fresh log is started.

max_files = *n*
maximum number of log files before rotation. Every time the number of archive logs exceeds this number, the oldest archive is removed.

runtime.*
settings regulating how the application starts/runs.

pidfile []
path to the pidfile (no pidfile unless specified).

run_as_user *username* **[]**
run as *username* if started as a daemon. Running as root by default.

non_daemon = true|false [false]
run as a terminal (non-daemon) application. By default, an instance started under root runs as a daemon.

source.*
settings for the source stream.

mcast_ifc
multicast interface for the source stream.

use_http10_get
use HTTP/1.0 in **HTTP GET** requests in order to avoid chunked output. Analyzer does **not** parse chunked input and, if encountered, the **smd** would exit with **BAD stream − stype=TS expected** error.

cache.*
settings for the 'rolling window'.

size_mb cache size in megabytes. (See −−**cachesz** option for details.)

tail_size_mb
size of post−event data segment. (See −−**tail** option for details.)

tail_duration_ms
clock−time duration of post−event data segment. (See −−**tail** option for details.)

min_bytes_left
number of bytes to free up to before each read from the stream. (See −−**minread** option for details.)

snapshot.*
settings for the snapshot files.

dir_path
path to the snapshot directory. Snapshot files are to be saved there. Please bear in mind that **smd** saves snapshots in the same (and only) thread it does other things in; I/O speed for the file system of the snapshot directory truly matters.

file_prefix
prefix for snapshot files, defaulting to 'snap'.

stream_analyzer.*
parameters of stream analysis.

enabled specifies if this **smd** instance would be doing stream analysis. If set to *false*, further parameters of this section are disregarded.

source_tag
is the alias for the processed stream. The analyzer would use this alias to refer to the stream in the logs (both *application* and *alert*). If omitted, the source URL (see the −i|−−source option) is used.

max_skipped_pkts
is the number of packets that the analyzer considers normal to skip before reaching the point when it can start the analysis. In order to start, the analyzer needs to get to *PAT* first, then to a valid *PMT*. It depends on the bitrate, as well as other parameters of the stream, how soon the required PAT/PMT combination is found. If it's not found within N=*max_skipped_pkts* MPEG-TS packets, the analyzer will return an error.

max_pdts_delta_ms
is the maximum 'gap' (in milliseconds) between the corresponding PTS or DTS values of a reference (**video**) track and **audio** tracks within the stream. If this parameter is specified, the analyzer checks that audio tracks are in sync with the video track. It does so by comparing the last−encountered PTS (and DTS) value of each audio track with the corresponding (last seen) value of the video track. If the difference exceeds *max_pdts_delta_ms*, an **alert** is issued (*AV−DELTA0* or *AV−DELTA+*). Please see **ALERTS** section below for details. The 0 (zero) value turns off the check.

max_pcr_delta_ms
is the maximum 'gap' (in milliseconds) between the corresponding PTS or DTS values of a reference (**PCR**) track and other tracks within the stream. If this parameter is specified, the analyzer checks that the stream's tracks are in sync with the PCR values. It does so by comparing the last−encountered PTS (and DTS) value of each (non−PCR) track with the corresponding (last seen) PCR value of the PCR track. If the difference exceeds *max_pcr_delta_ms*, an **alert** is issued (*PCR-DELTA*). Please see **ALERTS** section below for details. Also, bear in mind that streams may **NOT** be PCR−sync'ed at all w/o ill effects on the playback in popular players. The 0 (zero) value turns off the check.

pcr_freq_ms
is the maximum (PCR) time in milliseconds allowed between two consecutive PCR values. If the gap between the two exceeds *prc_freq_ms*, the analyzer issues **PCR−FREQ** alert. Please see **ALERTS** section below for details. The 0 (zero) value turns off the check.

access_log_alerts
directs the analyzer to write a fixed−format entry into an *alert log* for every alert issued, if set to true. If *false*, alerts are only written in the *application log* in a human−readable form. For the formats of alert entries in the alert log, please refer to **ALERTS** section below.

dump_stats_period
set to a non-zero value directs the analyzer to write its statistics (in a human−readable form) into the *application log* every N=*dump_stats_period* seconds. The 0 (zero) value disables this

function.

stream_analyzer.track.*
This section specifies additional (often **very** fine−grained) information for the analyzer to display. Beware that enabling some (or all) of the options would result in a significant increase of logging, so please plan accordingly.

sdt_ignore
tracks all *SDT (Service Description Table)* packets, normally ignored by the analyzer.

misc_ignore
tracks non−specific packets ignored by the analyzer, which ignores all packets with **pid**s not being within the selected *PDT* or not belonging to an audio, video or a PCR track. This option instructs the analyzer to make a short *application−log* entry on behalf of every packet ignored.

pdts_update
tracks all PTS/DTS updates, adding them to the application log.

pcr_update
tracks all PCR updates, adding them to the application log.

pat_pmt_pkts
tracks PAT/PMT occurrences after the required PAT/PMT combination has been processed by the analyzer.

Miscellaneous
non−specific settings.

pkt_reset_idx
directs to reset current−packet index to 1 after the index exceeds the parameter value. Not having it set would have the index roll over eventually by overflowing the 64−bit value of the index counter.

config_reload_period
will define (if > 0) how often (in seconds) **smd** should check for config−file changes. If a change is detected, the config is reloaded and re−applied.

lost_sync_alert = true|false [true]
issue **LOST−SYNC** alert when the stream goes out of sync, i.e. **smd** cannot find sync byte(s) appropriate for the stream's format. If this is set to *false*, **smd** would quit. If there are more than 10 alerts within a 3-second period, the app would still quit.

ALERTS

Alerts are issued by the analyzer on a number of pre−defined conditions. Each alert is entered as a *warning* into the application log, defined by the **log.*** section above.

Additionally, alerts can be written to a special *alert log* that, unlike the application log, has a strict format defined for each entry type. Alert log is described by the **alert_log.*** section of the configuration.

The following alerts are defined:

AV−DELTA0 (delta zero)
is issued when audio/video difference in PTS/DTS values **appears** to be zero, but the two compared values nevertheless are **not equal**. Added to detect PTS/DTS abnormalities and roll−over conditions.

Format: **ALERT AV−DELTA0 {source−tag} pts|dts {ref−pid} {ref−time} {src−pid} {src−time} {delta} {ptag}**

source−tag is the tag specified either by −−srctag command−line parameter or *stream_analyzer.source_tag* config setting.

pts|dts (one of the two used) indicates what metric the alert's timestamps relate to.

ref−pid is the (reference) video track PID.

ref−time is the (reference) video track timestamp, PTS or DTS.

src−pid is the (source) audio track PID.

src−time is the (source) audio track timestamp, PTS or DTS.

delta is the difference (absolute value) between reference and source timestamps.

ptag is a reference to the TS packet where the alarm was triggered. It is given in the form: [pkt[*N*]:(pcr=*X*)+*M*], where *N* is the index of the TS packet, *X* is the value of the last encountered PCR, and *M* is the offset from that PCR to the packet.

AV−DELTA+ (delta plus)
is issued when the difference (delta) between a video (reference) and an audio (source) track exceeds *max_pdts_delta_ms* milliseconds.

Format: **ALERT AV−DELTA+ {source−tag} pts|dts {ref−pid} {ref−time} {src−pid} {src−time} {delta} {ptag}**

The fields are the same as with the previous alert (with the exception of the alert's name).

PCR−DELTA
is issued when an absolute difference (delta) between a PCR and a PTS/DTS timestamp exceeds *max_pcr_delta_ms* milliseconds.

Format: **ALERT PCR−DELTA {source−tag} pts|dts {pcr−pid} {pcr−time} {src−pid} {src−time} {delta} {ptag}**

pcr−pid is the PCR (reference) track PID.

src−pid is the non−PCR (source) track PID.

pcr−time is the last encountered PCR timestamp.

All other fields match in the meaning the explanations above.

PCR−FREQ
is issued when the time difference (in ms) between two consecutive PCR timestamps exceeds *pcr_freq_ms* value.

Format: **ALERT PCR−FREQ {source−tag} {pcr−pid} {pcr1} {pcr2} {delta} {ptag}**

pcr−pid is the program ID for the PCR (track).

pcr1 is the former PCR timestamp.

pcr2 is the latter PCR timestamp.

All other fields match the descriptions given above.

LOST−SYNC
is issued when the application cannot identify the next packet as being of the established format. In the beginning, the logic scans the data and determines the format of the stream, as well as the size of the datagrams for that format. It then assumes that all further data would be of that same format.

Format: **ALERT LOST-SYNC {source−tag} {alert−count} {ptag}**

alert−count is the total number of LOST−SYNC alerts in a sequence. If enough time (3+ seconds) passed since the last alert the sequence counter resets to zero. After 10 consecutive alerts, **smd** would exit.

STATISTICS

smd logs various types of statistics into the *alert log*. All statistic records are prefixed with **STAT** {kind} (same way that alerts are prefixed with **ALERT**), where *kind* is one of the statistic types. This section will describe the format of different **STAT** entries.

TPS (throughput statistics)

lists counters related to the stream w/o regard for the underlying data.

Format: **STAT TPS {source−tag} {total_KB} {total_segs} {local_KBps} {global_KBps} {cached_KB}**

This entry uses floating−point numbers for most of its metrics, in KB (1024 bytes).

source−tag is the tag specified either by −−srctag command−line parameter or *stream_analyzer.source_tag* config setting. If analyzer disabled, '−' is output.

total_KB − total KB read in.

total_segs − total number of segments read in. For UDP stream it would be equal to the number of UDP datagrams, for HTTP it would be the number of (possibly unaligned) segments read. *local_KBps* − a verage speed as observed from the last report till now, in kilobytes per second.

global_KBps − average speed for the whole runtime, in kilobytes per second.

cached_KB − number of kilobytes in cache.

PMTs (PMT summary)

provides basic information on the PMT used for the analysis.

Format: **STAT PMTs {source−tag} {pmt_pid} {num_tracks}**

source−tag is the symbolic tag for the source stream (see −−srctag CLI option for details).

pmt_pid − the MPEG−TS **PID** for the PMT being used.

num_tracks − number of tracks (payload PIDs) used by the analyzer (i.e. audio/video).

PMTi (PMT item/track)

provides information on particular tracks used by the analyzer.

Format: **STAT PMTi {source−tag} {pid} {type} {p_count} {a_count} {description}**

source−tag is the symbolic tag for the source stream (see −−srctag CLI option for details).

pid − MPEG−TS PID for the given track.

type of the track: **A** for audio, **V** for video.

p_count − total number of *TS datagrams* processed (read in). Datagrams here are NOT UDP packets, but MPEG−TS datagrams (usually 188 bytes long).

a_count − total number of datagrams analyzed (those relevant to the analyzer for its tasks).

description − human−readable type of the stream, in square brackets. Example: [13818-3 audio (MPEG-2)].

PRG (program info)

provides attribites and counters for the program being analyzed.

Format: **STAT PRG {source−tag} {pcr_pid} {p_count} {a_count} {v_pcount} {v_acount} {a_pcount} {a_acount}**

source−tag is the symbolic tag for the source stream (see −−srctag CLI option for details).

pcr_pid − is the PID used for PCR data. It could match the video track or be a track on its own.

p_count − total number of MPEG−TS datagrams processed, for all tracks.

a_count − total number of MPEG−TS datagrams analyzed, for all tracks.

v_pcount − total number of MPEG−TS datagrams processed, for **video** tracks.

v_acount − total number of MPEG−TS datagrams analyzed, for **video** tracks.

a_pcount − total number of MPEG−TS datagrams processed, for **audio** tracks.

a_acount − total number of MPEG−TS datagrams analyzed, for **audio** tracks.

ALERT−count

provides accumulated values for issued alerts.

Format: **STAT ALERT−count {source−tag} {delta+} {delta0} {pcr−delta} {pcr−freq} {lost−sync}**

source−tag is the symbolic tag for the source stream (see −−srctag CLI option for details).

delta+ is the number of *AV−DELTA+* alerts issued so far.

delta0 is the number of *AV−DELTA0* alerts issued so far.

pcr−delta is the number of *PCR−DELTA* alerts issued so far.

pcr−freq is the number of *PCR−FREQ* alerts issued so far.

lost−sync is the number of *LOST−SYNC* alerts issued so far.

PCR-delta

provides various metrics on the difference between PCR and PTS/DTS values.

Format: **STAT PCR-delta {source−tag} {pcr−pid} {src−pid} {l_avg} {l_sum} {l_num} {l_min} {l_max} {g_avg} {g_sum} {g_num} {g_min} {g_max}**

source−tag is the symbolic tag for the source stream (see −−srctag CLI option for details).

pcr−pid is the PID for the PCR track.

src−pid is PID for the distinct track with PTS/DTS values (−1 = **all tracks**).

l_avg − 'local' average value, i.e. the average for the duration from the last report. All 'l_'−prefixed values are 'local'.

l_sum − sum of the metrics for the period.

l_num − number of measurements taken.

l_min − minimum value.

l_max − maximum value.

g_avg − 'global' average value, i.e. the average for the whole runtime so far. All 'g_'−prefixed values are 'global'.

g_sum − sum of the metrics for the whole run.

g_num − number of measurements taken.

g_min − minimum value.

g_max − maximum value.

PCR−freq

provides statistics for the PCR frequency in the same was as described for *PCR−delta*.

Format: **STAT PCR-freq {source−tag} {pcr−pid} −1 {l_avg} {l_sum} {l_num} {l_min} {l_max} {g_avg} {g_sum} {g_num} {g_min} {g_max}**

The parameters have the same meaning as for *PCR−delta* (above).

AV−delta

provides statistics for the difference between PTS/DTS of audio and video streams.

Format: **STAT AV-delta {source−tag} {ref−pid} {src−pid} {l_avg} {l_sum} {l_num} {l_min} {l_max} {g_avg} {g_sum} {g_num} {g_min} {g_max}**

source−tag is the symbolic tag for the source stream (see −−srctag CLI option for details).

ref−pid is the PID for the video track.

src−pid is PID for the distinct track with PTS/DTS values (−1 = **all tracks**).

The rest of the parameters are the same as for *PCR−freq* (above).

AUTHORS

Pavel V. Cherenkov

SEE ALSO

dbsink(1)

NAME

dbsink converts **smd** alert logs to formats suitable for different types of databases.

SYNOPSIS

dbsink PARAMETERS [OPTIONS]

DESCRIPTION

dbsink(1) generates files from **smd** alert logs, suitable for batch−loading into databases of different types. For **InfluxDB**, direct loading is supported via its *UDP−based* line protocol.

dbsink takes input either from a file (an alert log) or from *STDIN*. The input file could be either **static** (closed, not being added to any more) or one that **smd** is still adding to. **dbsink** allows to *follow* a log that is being actively used, the additional data would be read by **dbsink** as it comes (please see −f in **OPTIONS**).

The output could be segmented, a new file would be created every N lines or M **STAT**s of a designated kind. As the application switches over to a new segment, it renames the original output file to its 'final' name, by appending it with a (UNIX) timestamp, a sequence number and **.rota** extension. The goal here is to allow a (user−defined) batch loader app to process segments based on the final name/extension.

PARAMETERS

The following parameters are accepted:

−i|−−source {path}

path to the input file, or − (dash) for STDIN. Reading from STDIN allows to specify **dbsink** as an alert−log handler (i.e. opened via **popen(2)**) in **smd**(1) configuration.

If input is a regular file, unless −f (−−follow) is specified, **dbsink** will process all file data until **EOF** and exit.

−o|−−dest {path}

path to the destination file or a **UDP** URL for destination endpoint. If this parameter is omitted, **STDOUT** is used as the output stream. If a file path is specified, output is written to that file, which may be periodically *rotated*, depending on the −R|−−rotate option. For output into a UDP socket one should use a URL in the format: **upd://address:port**. If so, each line would be sent as a distince packet to the designated UDP socket.

NB: UDP output could be used to upload data to a database that supports UDP input. It was added to **dbsink** to take advantage of UDP line protocol in *InfluxDB*.

−D|−−db {type}

specifies the target type of DBMS. By default, *influx* is assumed. Right now it is the only one format supported. **CSV** format support is in the short−term plans. Based on this parameter, the application would pick a format to produce the output in, so that data segments could be batch−loaded into the database.

OPTIONS

The following options are accepted:

−l, −−logfile *path*

path to the *application log* file. Please do mind access permissons for the directory.

−L, −−level *crit|err|norm|warn|info|debug*
log level. The default level is *info*.

−S, −−skipold {sec}
skip log entries older that N seconds. Allows to resume processing from an old log and (auto−magically) skip irrelevant data.

−f, −−follow
do NOT stop at **EOF** of the input file, expect more data, read as it comes. Unless this option is specified, the application would exit after reaching the EOF point of the input file. **dbsink** can be given path to a log that is being actively appended, with this option

−k, −−clktime
use clock time, NOT log-parsed timestamps. **dbsink** by default parses the log−supplied timestamp and uses it for the entry. This option allows to disregard it and use real time (of entry parsing) instead.

−R, −−rotate {stat}:N [TPS:10]
rotate every N records/STATs. This setting dictates how often the output file would be rotated. The value of the parameter contains two parts, separated by a colon: 1) name of the **STAT** entry; 2) number of records (N). Rotation is performed for every N records of the given STAT. If the first part has been omitted (no colon), then any record type is counted in (meaning 'rotate every N records'). For non−file output the parameter is ignored.

−T, −−term
run as a terminal (non-daemon) application. This is the default behavior when run by a non−privileged user. −T could be specified when run as root in order **NOT** to become a daemon, for instance, for debugging purposes.

−K, −−syskey
generate system key (to use in licensing) and exit.

−V, −−version
output application's version and quit.

−h, −−help, −?, −−options
output brief option guide. This is output when run without parameters.

EXAMPLES:

dbsink -i /var/log/c1-alerts.log -o /var/db/c1.influx -f -R TPS:20 -L debug

The above example 'follows' /var/log/c1-alerts.log file and rotates the output every 20 *STAT TPS* entries. At the first rotation point (UNIX time=1515676976), *c1.influx* would be renamed to *c1.influx.1515676976-1.rota*. Logging level is set to *debug*. Since log file is unspecified, **dbsink** logs to STDERR.

dbsink -i - -o udp://192.168.0.30:8089 -k

This example reads data from STDIN (timestamping each entry at receipt point) and outputs to UDP port 8089 at host 192.168.0.30. No rotation is done (or would be possible). Log level is **info**, as by default.

AUTHORS

Pavel V. Cherenkov

SEE ALSO

smd(1)