
The config file is /etc/smsd.conf. You may specify another file using the option -c. During installation an easy example file will be copied to **smstools3/examples** directory.

The config file has the following structure:

```
some_global_settings

[first_modem_name]
...
[second_modem_name]
...
[third_modem_name]
...
and so on
```

Configuring the provider-sorting is shown in different document.

General

In case of yes/no settings, you can use the following keywords:

yes , no
true , false
on , off
1 , 0

Since version 3.1 also "no" values should be typed correctly, "no" is no more a default.

In case of lists, you need to use the comma character to separate items. Example: modem1, modem2, modem3

Since version 3.1.5 only whole line comments are allowed. For example:
 # This comment line is valid.
devices = GSM1 # This kind of comment is invalid.

Multiple different values

Since version 3.1 multiple different values can be defined for each setting. If smsd is started with command line argument -a (ask), settings with multiple choices are prompted and suitable value can be selected for a run. For example:

devices = ? GSM1 | GSM1, GSM2, GSM3 | GSM88, GMS99

A question mark tells that more than one value is defined. Possible values are separated using pipe. In this example the smsd will prompt:

Value for "global devices" (Enter for the first value, 0 to exit):
1) GSM1
2) GSM1, GSM2, GSM3

3) GSM88, GMS99
4) Other
Select 1 to 4:

When smsd is started without -a, first value is always choosen.

default and group sections

Since version 3.1.5 default settings for all modems can be defined in the **[default]** section. If setup has large count of similar modems, almost all settings can be defined once in [default] section and only device depended settings like *device* are required to define in the modem sections.

As "default" is now reserved name, it cannot be used as a modem name. Also "modemname" and "ALL" are reserved. Since version 3.1.20 also "communicate" is reserved.

Since version 3.1.19 "group" sections can be defined. If the system has *devices = ELISA* 11-13, GSM1, DNA33* 1-5*, the following groups can be used:

[ELISA*]
Settings for ELISA11, ELISA12 and ELISA13 SIM's

[DNA*]
Settings for DNA331, DNA332 etc. SIM's

Grouping works with device names which do not start with digit. Asterisk is placed to the position where the first digit exists.

All modems read [default] section first, if it exists. Next the group section is read, if it exists. Finally the modems own section is read, if it exists.

communicate section

Since version 3.1.20 special section **[communicate]** can define shortcuts which can be used when communicating with a modem (e.g. *smsd -C GSM1*). Possible keys are Alt-0 ... Alt-9, and the syntax is like the following:

[communicate]
a0 = ATZ
a5 = AT+CPMS?

When communicating starts, available shortcuts are printed to the console. Alt-? also prints available shortcuts. Shortcuts can be used after a modem is opened. Shortcut does not include CR character, it must be pressed by the user.

Since version 3.1.20 also the Ctrl-Z character can be sent using the Alt-Z key, required when terminating the PDU.

Global part

The global part begins at the top of the config file.

A top

adminmessage_device = name
Default value: *first available modem*.
Available from version 3.1.5. Defines which modem is used to send administrative messages from the mainspooler. This feature uses shared memory and works only if libmm is installed and statistics functionality is enabled.

admin_to = phone number
Default value: *not in use*.
Available from version 3.1. Specifies a destination number for administrative messages created by smsd. Messages are sent without using the filesystem.

alarmhandler = filename
Default value: *not in use*.
You can specify here an external program that is started whenever an alarm occurs.

alarmlevel = number
Default value: *LOG_WARNING*.
Specifies what levels start an alarmhandler. You can use value between 2 and 7.
Since version 3.1 a value can be defined as a word, like LOG_INFO, INFO or info.

alphabet = number
Default value: *ISO*.
Available from version 3.1.16. This setting defines how message body of outgoing file is handled, when there is no *Alphabet* header included. Choices are *ISO*, *Latin* or *Ansi* for ISO-8859-15, and *UTF* for UTF-8.

autosplit = number
Default value: *3*.
Controls if and how the program splits large text messages. The program does not split text messages with UDH.
If splitting is disabled, binary messages requiring more than one part are not sent.

0	disabled
1	enabled, no part-number
2	enabled, text numbers
3	enabled, concatenated format (not supported by some phones)

B

top

blacklist = filename
Default value: *not in use*.
Name of the blacklist file.

blockafter = number
Default value: *3*.
Available from version 3.0.9. A modem is blocked after n number of errors while **sending** messages. A successfull sending will reset this counter.

blocktime = number
Default value: *3600*.
A modem is not used so many seconds when it seems to be out of order.

C

top

check_pid_interval = number
Default value: *10*.
Available from version 3.1.17. Defines how often mainprocess will check the pidfile, to see if there was another smsd started. If another smsd is started, current smsd will stop immediately. Value 0 disables this checking. Value is seconds.

checked = directory
Default value: */var/spool/sms/checked*.
Path of the default Queue directory in case you do not use provider-sorting.

checkhandler = filename
Default value: *not in use*.
External program that checks if a message file is valid. (This script can also be used to convert message file from UTF-8 to ISO format, which is internal format used in smsd. See scripts/checkhandler-utf-8 for sample code.)
Since version 3.0.9 the smsd converts messages automatically from UTF-8 to ISO, if it is necessary.
If the checkhandler return a non-zero (other than 2) exitcode the message will not be sent.
Since version 3.1 the checkhandler can also modify a message file.
Exitcode 2 means that the checkhandler has moved a message to the spooler by itself.

child = filename
Default value: *not in use*.
Available from version 3.1.17. This setting creates a child process to the mainspooler. It starts when smsd starts, and stops when smsd stops. If child has created one or more childs, they are stopped too when smsd stops.

Child can be used for various purposes, for example feeding spooler from SQL database, taking care that the number of files in the spooler does not grow too much, and feeding is done only when smsd is running.

Filename must be /path/to/executable only, because executable is checked when smsd starts. Use the setting *child_args* if any arguments are required.

child_args = string
Default value: *empty*.
Available from version 3.1.17. Defines arguments to the child.

conversation_delimiter = string
Default value: *"-- " (without quotation marks)*.
Available from version 3.1.23. Each message in the conversation file is delimited with this string.

conversation_header = string
Default value: *Message - .*
Available from version 3.1.23. Defines a header for the header line.

conversations = directory
Default value: *not in use*.
Available from version 3.1.23. It set, content of sent and received messages are stored into this directory if alphabet of message is UTF-8. Each modem and each sender/receiver has a single file which is named as <modem>-<number>, for example GSM1-358401234567.

country_code = number
Default value: *empty*.
Available from version 3.1.23. This is used to target the conversation to the right file when both international and national number formats are used. The number in the file name is always in international format if it is not a short number.

D

top

date_filename = number
Default value: *0*.
Available from version 3.1. Defines if date is included to the filename of incoming message. With value 1 like 2007-09-02.GSM1.xxxxxx and with value 2 like GSM1.2007-09-02.xxxxxx.

date_filename_format = format string
Default value: *compatible with previous versions*.
Available from version 3.1.7. Specifies a format for date string in filenames. See *man strftime* for usage details.

datetime_format = format string
Default value: *compatible with previous versions*.
Available from version 3.1. Specifies a format for timestamps. See *man strftime* for usage details.

Before the version 3.1.7 a name for this setting was *datetime*. The old name can still be used because of backward compatibility.

decode_unicode_text = yes/no
Default value: *no*.
Available from version 3.0. Defines if the incoming Unicode text is decoded internally.

delaytime = number
Default value: *10*.
Smsd sleep so many seconds when it has nothing to do.

delaytime_mainprocess = number
Default value: *not in use*.
Available from version 3.1.4. If this value is not set, *delaytime* setting is used in the main process. With this setting outgoing messages can be checked more frequently than incoming messages.

devices = names
Default value: *empty*.
List of names of your modems, maximum 64 devices. This limit is changeable.

Since version 3.1.12 this setting can be given in shortened form:
devices = GSM* 101-164

"GSM" defines a prefix, "101" is the number of first modem and "164" is the number of last modem. The example is extracted to: devices = GSM101,GSM102,GSM103 etc...

Since version 3.1.19 multiple wildcard definitions can be used, together with static names. For example:
devices = ELISA* 11-15, SONERA* 11-15, GSM1

E top

errorsleeptime = number
Default value: *10*.
A modem sleeps so many seconds when it answers a command with ERROR.

eventhandler = filename
Default value: *not in use*.
Specifies an external program or script that will execute whenever a message was sent, received or failed.
(If your locale is UTF-8, you can use this script to convert received messages from smsd's internal format (ISO) to UTF-8. See scripts/eventhandler-utf-8 for code sample.)
Since version 3.0.9 there is *incoming_utf8 = yes* setting available. Using this setting the external conversion is not required.

eventhandler_use_copy = yes/no
Default value: *no*.
Available from version 3.1.17. If a copy of failed / incoming / sent / report message is created, this setting defines if a copied file is given to the eventhandler instead of original file.

executable_check = yes/no
Default value: *yes*.
Available from version 3.1.1. This setting defines if all executables are checked during the startup check. Usually eventhanler, alarmhandler etc. are shell scripts or some other single files which can be executed and therefore checked simply. If using a settings like *eventhandler = /usr/local/bin/php -f /usr/local/bin/smsd_eventhandler.php*, the check will fail and smsd will not start unless *executable_check = no* is defined.

F top

failed = directory
Default value: *not in use*.
Path of the Failed Folder. Delete this line if you do not want to keep failed files.

failed_copy = directory
Default value: *not in use*.
Available from version 3.1.17. It not empty, copy of all failed messages are stored into this directory. Depending on the setting eventhandler_use_copy smsd does not do anything with these files, so external application can use them in whatever way it wants.

filename_preview = number
Default value: *not in use*.
Available from version 3.1. Defines how many characters of message text is concatenated to the name of a message file. Characters used are a...z, A...Z, 0...9, - and period. All other characters are replaced with _'s.

G top

group - see here

H top

hangup_incoming_call = yes/no
Default value: *no*.
Available from version 3.1.5. If set to *yes* and detected unexpected input contains "RING", incoming call is ended. Use modem setting *voicecall_hangup_ath* to define if "ATH" is used to make hangup instead of "AT+CHUP".

I top

ic_purge_hours = number
Default value: *24*.
ic_purge_minutes = number
Default value: *0*.
ic_purge_read = yes/no
Default value: *yes*.

ic_purge_interval = number
Default value: *30*.
Available from version 3.1.5. These settings defines how concatenation storage is purged when *internal_combine* is used and messages with missing parts are received. Storage is checked every *ic_purge_interval* minutes. If there are message parts older than defined with *ic_purge_hours* and/or *ic_purge_minutes* settings, message parts are deleted. If *ic_purge_read* is *yes*, message is stored to the incoming folder. In this case there will be only one part in the file and a header *Incomplete: yes* indicates that message is broken. Value 0 for both *ic_purge_hours* and *ic_purge_minutes* disables this feature.

ignore_exec_output = yes/no
Default value: *no*.
Available from version 3.1.7. Defines if an unexpected output from eventhanders is ignored.
Usually eventhandlers should not output anything. If there is some output, it usually means that some error has occurred. These errors are often problematic, because that output cannot be seen and error is therefore not notified. Smsd will print any output from the evenhandlers to the log file. This output can also be used for debugging purposes, but usually debugging should be done by running eventhandler manually.

ignore_outgoing_priority = yes/no
Default value: *no*.
Available from version 3.1.5. With this setting *Priority: high* header is not checked. This speeds up spooling on systems where priorizing is not required and there are lot of files in the spooler directory.

incoming = directory
Default value: */var/spool/sms/incoming*.
Path of the Incoming Folder.

incoming_copy = directory
Default value: *not in use*.
Available from version 3.1.16. It not empty, copy of all incoming messages are stored into this directory. Smsd does not do anything with these files, so external application can use them in whatever way it wants.

Note since version 3.1.17: the setting eventhandler_use_copy defines which one file is given to the eventhandler.

incoming_utf8 = yes/no
Default value: *no*.
Available from version 3.0.9. With this setting messages using ISO or GSM alphabet are stored using UTF-8 character set.

infofile - see here

internal_combine = yes/no
Default value: *yes*.
Available from version 3.0. Defines if the incoming multipart message is combined internally.

internal_combine_binary = no
Default value: *internal_combine which defaults to* ***yes***.
Available from version 3.1.5. Defines if the incoming multipart binary message is combined internally.

international_prefixes = list of numbers
Default value: *not in use*.
Available from version 3.1.5. See SMS file (Using Type Of Address selection) for details.

K

top

keep_filename = yes/no
Default value: *yes*.
Available from version 3.1. If this is set to no, an unique filename is created each time when a message file is moved from directory to another directory. This ensures that any previously sent message file is not overwritten, even if the original filename was the same. Also if an user has created filename with space character, this creates a new name without spaces and therefore spaces can't effect to operation of an eventhandler.

keep_messages = yes/no
Default value: *no*.
Available from version 3.1.5. This is for testing purposes. Smsd runs as usual but messages are not removed from the modem. After all messages are read, smsd will stop. Use this with one modem only.

L

top

language_file = filename
Default value: *not in use*.
Available from version 3.1. Message files can be written using localized headers. See the localizing for details.

log_charconv = yes/no
Default value: *no*.
Available from version 3.0.9. With this setting a details of character set conversions (outgoing UTF-8 to ISO conversion and incoming GSM/ISO to UTF-8 conversion) is printed to the log. If smsd is compiled using DEBUGMSG definition, details are also printed to the console. Logging feature can be useful if you have some troubles with characters and like to know what exactly happens inside the smsd.

logfile = filename
Default value: *empty*.
Name of the log file. Delete this line, if you want to use the syslog for logging. You can use "1" to write to the console (stdout).

This setting can be overridden by the -l (ell) command line argument.

loglevel = number/word
Default value: *4* for logfile, *7* for syslog.
Sets the verbosity of a log file.

This affects also syslog. If you want all messages in syslog, you need to set it to "7" (or higher) here and "*" in the config file of syslog. If you want less messages, you can reduce it here or in the config file of syslog, both will work.

debug	7	All AT-Commands and modem answers and other detailed informations useful for debugging
info	6	Information what is going on at the moment. Not detailled enough for debugging but maybe interesting.
notice	5	Information when a message was received or sent and when something not normal happens but program still works fine (for example wrong destination number in SMS file).
warning	4	Warning when the program has a problem sending a single short message.
error	3	Error message when the program has temporary problem (for example modem answered with ERROR during initialisation or a file can not be accessed).
critical	2	Error message when the program has a permament problem (for example sending failed many times or wrong permissions to a queue).

The numbers in this table are taken from GNU/Linux. Probably all operating systems use the same numbers.

Since version 3.1 a value can be defined as a word, like LOG_INFO, INFO or info.

log_read_from_modem = yes/no
Default value: *no*.
Available from version 3.1.9. In some cases, when resolving troubles with a modem, it's useful to see exact data which is received from the modem. This setting is similar than *log_unmodified*, but displays the data as a hexadecimal dump.

log_read_timing = yes/no
Default value: *no*.
Available from version 3.1.16. This setting is for tuning purposes. When testing and searching suitable value for the *poll_faster* factor, it's important to see how the modem communicates and how much modem process generates server load.

log_response_time = yes/no
Default value: *no*.
Available from version 3.1.16. With this setting response time in milliseconds is included in the log.

log_single_lines = yes/no
Default value: *yes*.
Available from version 3.1. Defines if linefeeds are removed from the modem response.

logtime_format = format string
Default value: *compatible with previous versions*.
Available from version 3.1.7. Specifies a format for date string logging. See *man strftime* for usage details.

Since version 3.1.14 this setting can have special keywords included: *timeus* or *timems*. Keywords are replaced with a current value of microseconds or milliseconds.

logtime_ms = yes/no
Default value: *no*.
logtime_us = yes/no
Default value: *no*.
Available from version 3.1.14. With these settings the timestamp in the log file can have microseconds or milliseconds shown. These settings can be used when a default format for timestamp is in use. Value is shown after seconds and is delimited with a dot.

If *logtime_format* is defined, these settings have no effect. Milliseconds or microseconds can be included in customized *logtime_format* using keywords *timeus* or *timems*.

log_unmodified = yes/no
Default value: *no*.
Available from version 3.1.7. In some cases, when resolving troubles with a modem, it's useful to see what kind of line ends were received from the modem. With this setting spaces and line ends are not removed from the string which is logged. This setting overrides the setting *log_single_lines*.

M top

max_continuous_sending = number
Default value: *300 (5 min)*.
Available from version 3.1.5. This setting is in seconds and defines how long modem can send messages without doing anything else. After *max_continuous_sending* time is reached, received messages are checked and other tasks are run.

Since version 3.1.18 continuous sending breaks if a signal USR2 is received. Sending continues as soon as receiving and other tasks are done.

N top

national_prefixes = list of numbers
Default value: *not in use*.
Available from version 3.1.5. See SMS file (Using Type Of Address selection) for details.

notifier = yes/no
Default value: *no*.
Available from version 3.1.17. If set to *yes*, mainspooler creates a child process which will use *inotifywait* to monitor *outgoing* directory. As soon as a new file appears in the directory, mainprocess gets a signal SIGCONT and continues immediately. New files are spooled as fast as possible, even when mainprocess has long sleeping times to reduce CPU load.

This feature is available only on GNU/Linux systems. When compiling smsd on other systems, edit the Makefile and uncomment the line which defines DISABLE_INOTIFY. By default inotify is enabled in

Makefile.

O

top

os_cygwin = yes/no
Default value: *no*.
Available from version 3.0.10. Defines if smsd is running on Cygwin environment. This is needed if outgoing file permissions should be checked and changed by the smsd.

outgoing = directory
Default value: */var/spool/sms/outgoing*.
Path of the Outgoing Queue folder.

P

top

phonecalls = directory
Default value: *empty, incoming directory is used*.
Available from version 3.1. Defines where *phonecalls* data is stored.

Note since version 3.1.16: If incoming_copy is used, and phonecalls are stored into the incoming directory, phonecalls are also copied. If phonecalls have their own directory, copying is not done.

pidfile - see here

priviledged_numbers = list of numbers
Default value: *not in use*.
Available from version 3.1.5. This list can be used with *check_memory_method* values 31, 41, and 5. Global list is default for each modem. List can be comma separated list of numbers or their parts starting from the left. Maximum 25 priviledged numbers can be defined. When messages are received, messages from priviledged numbers are handled first. First number in the list has highest priority.

R

top

receive_before_send = yes/no
Default value: *no*.
Forces smsd to empty the first SIM card memory before sending SM. This is a workaround for modems that cannot send SM with a full SIM card.

regular_run = filename
Default value: *not in use*.
regular_run_interval = number
Default value: *300*.
Available from version 3.1. A regular_run is an external script or program which is run regularly while the smsd is running. A value regular_run_interval describes number of seconds between each run.
See Running for more information and sample usage.

report = directory
Default value: *not in use*.
Available from version 3.1. Path of the Report Folder. Without this setting status report messages are stored to the Incoming Folder.

report_copy = directory
Default value: *not in use*.
Available from version 3.1.17. It not empty, copy of all report messages are stored into this directory. Depending on the setting eventhandler_use_copy smsd does not do anything with these files, so external application can use them in whatever way it wants.

S

top

saved = directory
Default value: *not in use*.
Available from version 3.1. Path of the Saved Folder. If defined, smsd will store concatenation storage's to this directory (otherwise incoming directory is used). At startup check existing concatenation storages are moved from incoming directory to saved directory. Zero sized files are ignored. If both directories has a storage file with data, fatal error is produced and the smsd does not start.

sent = directory
Default value: *not in use*.
Path of the Sent Folder. Delete this line, if you do not want to keep copies of each sent message file.

sent_copy = directory
Default value: *not in use*.
Available from version 3.1.17. It not empty, copy of all sent messages are stored into this directory. Depending on the setting eventhandler_use_copy smsd does not do anything with these files, so external application can use them in whatever way it wants.

shell = filename
Default value: */bin/sh*
Available from version 3.1.5. Defines which shell is used to run eventhandler and other external scripts.

shell_test = yes/no
Default value: *yes*
Available from version 3.1.14. When *executable_check* is enabled, testing of the *shell* can be omitted with this setting.

sleeptime_mainprocess = value
Default value: *1*
Available from version 3.1.17. Value is seconds. This setting defines how long mainprocess sleeps when no any messages are spooled. While looping, another smsd is detected, modem processes are listened and statistic files are written. This setting does not affect for picking up new outgoing files, there is another setting *delaytime_mainprocess* for it. Also notice that if an external child with inotifywait is used, mainprocess will get the signal CONT and continues spooling immediately after a new file exists. Because of this, delaytime can be high to save resources on small systems.

If it is needed that smsd really sleeps using *sleep()* while idle, this value can be set to higher than 1. However, in usual systems there is no need for that. Even if value is higher, regular_run script is executed on time, if in use. PID and processes are checked less often, and statistics are written less often if this value is higher than values set to those tasks.

smart_logging = yes/no.
Default value: *no*.
Available from version 3.1.5. This feature is available when file based logging is used. If *loglevel* is less than 7 (for example "notice" is a good choise with smart_logging), trouble log (with loglevel 7) about whole communication is written to different file if there has been any errors.

"Whole communication" means sending single SMS, receiving single SMS, and other things what smsd will do after idle time is spent. When communication starts, all possible log lines are collected to internal buffer and only *loglevel* lines are written to the *logfile*. If during communication there are any errors, all collected lines are printed to trouble log when communication reaches it's end.

This feature was made because with loglevel 7 logfile grows very much and fast, and debug level is not usually needed when there was no any errors. In case of errors it's important to see whole communication, not just a single line which says that "error happened".

File name is created with the rule: if lenght of *logfile* setting is more than 4 characters and setting ends with ".log", trouble log filename will end with "_trouble.log". If length is less than 4 or setting does not end with ".log", trouble log filename is *logfile* appended with ".trouble". In usual cases *logfile* is **/var/log/smsd.log** and trouble log filename will be **/var/log/smsd_trouble.log**, or in some (Debian, Ubuntu, ...) distributions: **/var/log/smstools/smsd.log** and **/var/log/smstools/smsd_trouble.log**.

spool_directory_order = yes/no
Default value: *no*.
Available from version 3.1.9. With this setting files are handled in the order they are on disk. This disables priorizing and also the age of files are not checked.

start = filename
Default value: *not in use*.
Available from version 3.1.18. Defines a script/program which is executed when smsd starts. This should be /path/to/executable only. If any arguments are required, use *start_args*. If return value of this script is anything else than 0, smsd stops.

start_args = string
Default value: *empty*.
Available from version 3.1.18. Defines arguments for *start* script.

stats = directory
Default value: *not in use*.
Specifies the directory where smsd stores statistic files. The directory must exist before you start smsd. If not given, then the program does not write statistic files. Since version 3.1.1 message counter files are stored to this directory even if smsd is compiled without statistics enabled.

stats_interval = number
Default value: *3600*.
Smsd writes statistics files every n seconds. Value 0 disables statistics but counters are still updated if stats directory is defined.

stats_no_zeroes = yes/no
Default value: *no*.
Smsd does not write statistic files when no message was sent or received (Zero-Counters) if this is set to yes.

status_include_counters = yes/no
Default value: *yes*.
status_signal_quality = yes/no
Default value: *yes*.
Available from version 3.1.5. These settings define if message counters and explained signal quality is included in the line of *status* file.

status_include_uptime = yes/no
Default value: *no*.
Available from version 3.1.16. Defines if start timestamp and uptime are printed to the end of status file, with empty line as delimiter.

For example:

```
Status:     16-06-07 00:16:49, irrri-------
SONERA:     16-06-07 00:16:45, Idle,      131267, 0, 4666, ssi: -71 dBm (Excellent), ber: < 0.2 %
ELISA:      16-06-07 00:16:45, Receiving, 133877, 3, 4741, ssi: -79 dBm (Good), ber: < 0.2 %
DNA:        16-06-07 00:16:47, Receiving, 127181, 0, 4708, ssi: -65 dBm (Excellent), ber: < 0.2 %
SAUNALAHTI: 16-06-07 00:16:48, Receiving, 128495, 1, 4678, ssi: -79 dBm (Good), ber: < 0.2 %
SAUNA2:     16-06-07 00:16:45, Idle,      5640,   9, 488,  ssi: -77 dBm (Good), ber: < 0.2 %

Start:      16-04-03 19:30:38, up 64 days, 4:46
```

status_interval = number.
Default value: *1*.
Available from version 3.1.5. If statistics function is enabled and *stats* directory is defined, smsd writes file named ***status*** into this directory. The file contains status of all modems in the first line using *Status:* header (this is similar than smsd -s outputs to console) and explained status in the next lines using modem's name as a header. Smsd writes status file every *status_interval* seconds if a status has changed. Value 0 disables this feature.

For example, the output is like:

```
Status:      09-05-27 20:46:17,  irir------------
SONERA:      09-05-27 20:46:09,  Idle,       123,  0,  321,  ssi: -63 dBm, ber: < 0.2 %
ELISA:       09-05-27 20:46:12,  Receiving,  234,  0,  432,  ssi: -73 dBm, ber: < 0.2 %
```

```
DNA:         09-05-27 20:46:06,  Idle,       456,  0,  543,  ssi: -77 dBm, ber: < 0.2 %
SAUNALAHTI:  09-05-27 20:46:14,  Receiving,  678,  0,  654,  ssi: -69 dBm, ber: < 0.2 %
```

Timestamp value tells when status is created or modem initialisation was last started.

Status can be: (s) Sending, (r) Receiving, (i) Idle, (b) Blocked, (t) Trouble, (-) Unknown. Trouble -status means that something abnormal has happened and if smart_logging is in use, trouble log will be written.

Counters are: sent, failed, received. Sent counter is the value from *stats/<modemname>.counter* file. Smsd does not clear this value. Failed and received counters are from original statistics data and they are cleared each time when stats are stored (*stats_interval*), or smsd is restarted.

store_original_filename = yes/no
Default value: *yes*.
Available from version 3.1. Together with keep_filename this controls when the original filename is stored to message file when it's moved from outgoing directory to the spool.

store_received_pdu = number
Default value: *1*.
Available from version 3.0. Controls when the incoming PDU string(s) is stored to message file.

0	no PDU's are stored
1	unsupported PDU's are stored
2	unsupported and PDU's with 8bit binary data or Unicode text are stored
3	all PDU's are stored

Header is "PDU: " and PDU's of a multipart message are stored from 1 to n order.

store_sent_pdu = number
Default value: *1*.
Available from version 3.0.9. Controls when the outgoing PDU string(s) is stored to message file.

0	no PDU's are stored
1	failed (to send) PDU's are stored
2	failed and PDU's with 8bit binary data or Unicode text are stored
3	all PDU's are stored

Header is "PDU: " and PDU's of a multipart message are stored from 1 to n order.

suspend = filename
Default value: *not in use*.
Available from version 3.1.7. With this file, any modem process can be suspended. When a process is suspended, the modem port is closed and modem process does not do anything.

The file is checked before smsd starts sending or receiving. If a name of the device is found from the file, suspend will start. The format of a line in file is: **<devicename>: <reason>**. Colon is needed, but the <reason> is optional. It is printed to the log. A special line **ALL: <reason>** can also be used, and this affects for all modem processes.

When a process is suspended, the file is checked every ten seconds, or more frequently if the *delaytime* value is smaller.

When a process is suspended, it listens a signal SIGUSR2. If this signal is received, modem process will send one message if there is something to send, and receive messages once.

Since version 3.1.18 configuration changes can be done on the fly, if a modem setting read_configuration_after_suspend is set to *yes*. Note that if a modem process is idle and delaytime is very long, it takes a while until process goes to suspended. This delay can be bypassed by sending a signal CONT to a process.

Since version 3.1.19 wildcard definitions can be used in the suspend file, for example: **ELISA*: <reason>**. The letters before ***:** are tested, and if the name of a modem matches with them, suspend is applied.

T

top

terminal = yes/no
Default value: *no*.
Available from version 3.1. Enables terminal mode like command line argument *-t*.

trim_text = yes/no
Default value: *yes*.
Available from version 3.1.7. With this setting trailing whitespaces are removed from the outgoing message. Does not effect with messages written using Unicode or GSM alphabet.

Since version 3.1.9, if the message is going blank, the removal is not done. This is because some people, really, need to send a message which contains only single space character.

trust_outgoing = yes/no
Default value: *no*.
Available from version 3.1.5. This setting can be used to speed up spooling, but only if it is completely sure that files are created by rename and permissions are correct.

U

top

umask = value
Default value: *empty*.
Available from version 3.1.7. Effective umask for smsd can be set in the configuration file. Value can be hexadecimal, decimal or octal format.

use_linux_ps_trick = yes/no
Default value: *no*.

Available from version 3.1.7. This can be used on GNU/Linux systems to create short process names like:

```
smsd: MAINPROCESS
smsd: GSM1
```

Next four settings are available from version 3.0.2:
user = username
Default value: *not in use*.
group = groupname
Default value: *not in use*.

If the smsd is started by the root, these two settings can be used to switch smsd to run as an unpriviledged user.
Since version 3.0.9, if *user* is set but *group* is unset, that user's normal groups (e.g. from /etc/groups) are used. This means you can allow other users on the system access to write messages to the outgoing spool without giving them direct access to the serial port.

infofile = filename
Default value: */var/run/smsd.working*.
pidfile = filename
Default value: */var/run/smsd.pid*.

Location of *infofile* and *pidfile* can be changed with these settings. This is usually necessary if the smsd is running as an unpriviledged user. If a sms3 script is used to start and stop the smsd, these settings should be defined in the script.
These four settings can be overridden by the command line argument(s):

- -ix set infofile to x
- -px set pidfile to x
- -ux set username to x
- -gx set groupname to x

See Running for more information.

V

top

validity = number
Default value: *255*.
Available from version 3.0. See SMS file for details of possible values.

voicecall_hangup_ath = yes/no
Default value: *no*.
Available from version 3.1.5. Defines if **ATH** is used to hangup call instead of **AT+CHUP**.

W

top

whitelist = filename
Default value: *not in use*.
Name of the whitelist file. The black list takes precedence before the white list. See Blacklist and Whitelist for more details and sample usage.

Modem settings

[modem name]
Begin of a modem settings block. The modem name must be the same as in the devices= line in the global part.

Name of a modem can contain alphanumeric characters, underscore, minus-sign and dot. Other characters are not allowed, because the name is used to create filenames too.

Name of a modem cannot be *default*, because it's a name of a [default] block. Also a name cannot be *modemname*, because it's a macro for several other settings. And the name cannot be *ALL* or *communicate*. If the name is not acceptable, smsd gives an error and does not start.

NOTE for Cygwin users: Do not use a device name, like COM1, as a modem name. While a message is received, a file starting with this name is created and Windows handles it as a device. This will cause a modem process to be freezed.

Since version 3.1.12 it is no more mandatory that the modem settings block exists for all modems. When using large number of similar modems, it is often enough that [default] block exists. All modem processes will read and apply [default] settings first.

Since version 3.1.17, if using *child*, the name of it's process is CHILD. And if using *notifier* on GNU/Linux systems, the name of it's process is NOTIFIER. Avoid using those names as modem names.

a

top

adminmessage_limit = number
Default value: *not in use*.
adminmessage_count_clear = number
Default value: *not in use*.
Available from version 3.1.5. Adminmessage_limit specifies the maximum number of administrative messages to send. After this limit is reached, no more administrative messages will be sent until the smsd is restarted or message counter is cleared by the adminmessage_count_clear setting. The value of this setting is minutes.

admin_to = phone number
Default value: *not in use*.

Available from version 3.1. Specifies a destination number for administrative messages created by smsd. This setting overrides the setting in the global part.

alarm_command_is_sent_time = number
Default value: *2000*.
Available from version 3.1.23. Number is milliseconds. Usually it takes less than 0 milliseconds to send a command to modem. In rare cases the system may have collisions in serial communication, and remarkable delay may exist. If *alarm_command_is_sent_time* is set to more than 0, alarmhandler is called if time to send a command was equal or bigger than the setting. The alarm is "*ALARM_COMMAND_IS_SENT_TIME EXCEEDED: n (x/y)*", where n says how long it took to send a command. Value x says how many times the limit was exeeded, and y says how many times a command was sent fast. With this alarm and counters the alarmhandler could decide if some actions are required.

b

top

baudrate = number
Default value: *115200*.
Specifies the speed of the serial communication in bits per second. Most modems including old devices work well with 115200. If this speed is not supported by the system, 19200 is used. Some very old devices may need lower speed like 9600 baud.

c

top

check_memory_method = number
Default value: *1*.
Available from version 3.1.5. Defines how incoming messages are checked:

0	CPMS is not supported. Default values are used for *used_memory* and *max_memory*.
1	CPMS is supported and must work. In case of failure incoming messages are not read.
2	CMGD is used to check messages. Some devices does not support this. To see if this is supported, check the answer for **AT+CMGD=?** command. Answers like: *+CMGD: (1,2,3),(0-4)* *OK* = is supported, there are messages number 1, 2 and 3 in the memory. *+CMGD: (),(0-4)* *OK* = is supported, no messages in the memory. *+CMGD: (1-30)* *OK* = is not supported, a modem does not give message numbers even if there is messages available.
3	CMGL is used to check messages. Message is deleted after it is read.
4	CMGL is used to check messages. Messages are deleted after all messages are read.
31	CMGL is used to check messages. Message is deleted after it is read. CMGL data is checked and PDU is taken directly from the list.
41	CMGL is used to check messages. Messages are deleted after all messages are read. CMGL data is checked and PDU is taken directly from the list.
5	CMGL is used to check messages. Messages are deleted after all messages are read. CMGL data is checked and PDU is taken directly from the list. Multipart message is handled after all of it's parts are available. After multipart message is handled, only the first part is deleted by the smsd. The modem will delete rest of parts by itself. This is SIMCOM SIM600 compatible.

NOTE: Some devices are incompatible with CMGL method.
NOTE: With values 31, 41 and 5 *priviledged_numbers* sorting can be used.

check_network = value
Default value: *1*.
Available from version 3.1, enhanced in version 3.1.5. Defines how network registration is checked:

0 no	Network registration is not checked.
1 yes	Network registration is always checked
2	Network registration is checked only when preparing to send messages.

If a modem does not support network checking, checking is automatically ignored.
With value 2 incoming messages are processed faster.

check_sim = yes/no / once
Default value: *no*.
check_sim_* settings are available from version 3.1.21. If the device and smsd are started before the SIM is inserted to the device, communication will fail and modem process stops. With some devices modem process can wait until the SIM becomes ready.

A setting *check_sim* defines if the SIM is checked every time when a modem is initialized. Value *once* means that checking is only done when a modem is initialized for the first time.

All devices are not compatible with this feature. Some tested devices, like Huawei E353 and Neoway M590E do not work if the SIM is not in place when a device is switched on. Telit EZ-10 worked well and any reset was not required. SIM800L also worked well, but RADIO_OFF_ON was required. Some devices like M590E stop the communication if the radio is switched off, so that kind of reset cannot be used. If this feature is going to be used, carefull tests should be done.

When an error with SIM is detected, modem port is closed while modem process is waiting. USB modem can be disconnected and reconnected, if the port remains the same.

check_sim_cmd = string
Default value: *AT+CPIN?*.
Defines a command which is used to check the SIM. Default value *AT+CPIN?* is suitable for most devices. The command should return ERROR when the SIM is not ready.

check_sim_keep_open = yes/no
Default value: *no*.
Defines if a modem is closed when SIM was not ready and modem process is waiting for retrying.

check_sim_reset = string / RADIO_OFF_ON / RADIO_OFF_ON_SLOW
Default value: *not used*.
Defines if reset with AT commands is done before retrying after the failure. Special value *RADIO_OFF_ON* is translated to **AT+CFUN=0;+CFUN=1** which sets the radio off and immediately back to on without delays. Value *RADIO_OFF_ON_SLOW* is translated to **AT+CFUN=0[3]AT+CFUN=1[5]** where numbers in square brackets defines a delay in seconds. This setting sets the radio off, waits three seconds, sets the radio back on and waits five seconds.

check_sim_retries = number / forever
Default value: *10*.
Defines how many times modem process will retry. With a value *forever* modem process will retry, let's say, forever.

check_sim_wait = number
Default value: *30*.
Defines how many seconds modem process will wait until retry.

cmgl_value = string
Default value: *4*.
Available from version 3.1.5. If *check_memory_method = 3, 4, 31, 41 or 5* is used, correct value for AT+CMGL= command must be defined. This value depends on the modem.

communication_delay = number
Default value: *0*.
Available from version 3.1.5. Only some very problematic modems may need this setting. Defines minimum time in milliseconds between latest answer from modem and next command which will be sent to modem.

conversation_numbers = list of numbers
Default value: *empty*.
Available from version 3.1.23. If *conversations* is set in the global settings, a modem can select which conversations are created, if a conversation file does not already exist. The value is a list of numbers in international format separated by a comma. Short numbers must be listed without the 's' prefix. Each number may have wildcard definitions which is *fnmatch* styled.

If you want that the modem process does not start any conversation, use the setting **conversation_numbers = NONE**. In this case smsd still saves the conversation if the conversation file exists. If this setting is missing, any conversations are started and stored, if the feature is enabled.

crlf = yes/no
Default value: *no*.
Available from version 3.1.23. By default modem commands are terminated with CR only. With this setting CRLF can be used, if some modem requires it.

cs_convert = yes/no
Default value: *yes*.
The program converts normal text messages into GSM character set. You need this to display german umlauts and control characters correctly.

cs_convert_optical = yes/no
Default value: *yes*.
Available from version 3.1.16. When a character cannot be represented in the GSM character set, it can be approximated through one or several similarly looking characters.

d top

decode_unicode_text = yes/no
Default value: *use the global part setting which defaults to* ***no***.
Available from version 3.0. Specifies an internal Unicode decoding like in the global part.

delaytime = number
Default value: *use the global part setting which defaults to* ***10***.
Available from version 3.1.18. Defines how many seconds a modem process sleeps when it has nothing to do. This setting overrides the global *delaytime* value.

delaytime_random_start = yes/no
Default value: *no*.
Available from version 3.1.18. Defines if the first sleep of a modem process is randomized, using a value between 0 and *delaytime*. With very large number of modems it may be good that all processes are not working at the same time.

description = string
Default value: *empty*.
Available from version 3.1.16. If set, this description is written to each SMS file as an additional header.

detect_message_routing = yes/no
Default value: *yes*.
Available from version 3.1.5. By default smsd tries to detect if a modem is in message routing mode. Before sending a command smsd listens if some data with routed message is available. Also, after a command is sent, smsd checks the answer. In both cases, if there is one or more routed message coming, a notification is written to the log and alarmhandler is called. Routed messages are saved and handled later when smsd can do it.

NOTE: This checking is done to avoid errors and loss of messages. Routing mode SHOULD NOT BE USED in normal operation. With routing mode it is NOT quaranteed that all messages are delivered. Some devices are in routing mode by default and this feature helps to detect it. Use *init = AT+CNMI=...* with suitable values to disable routing mode. Values depend on modem, see the manual of a modem for details. All received messages must be stored to the message store which is usually "SM" (SIM card memory).

detect_unexpected_input = yes/no
Default value: *yes*.
Available from version 3.1.5. Before any command is sent to the modem, smsd checks if there is some unexpected input. For example some modem may send new message identification (CMTI) if settings are incorrect. Any unexpected input will be logged.

device = name of serial port / definition of internet host
Default value: *empty*.
Specifies the device name of the serial port or internet host to the modem. GNU/Linux example: /dev/ttyS0. Windows example: /dev/com1 or /dev/ttyS0. Solaris example: /dev/cuaa. Network modem example: @10.1.1.1:5000.

Since version 3.1.7 this setting can define a socket if starting with character @. Format for the internet host is: *@<host_or_ip>:<port>*. Host definition can be name or IP address.

Since version 3.1.12 this setting can contain a special word "modemname" (without quotation marks). For example:

> **[default]**
> **device = /dev/tty*modemname***

When a device is *USB0*, the setting is extracted as */dev/ttyUSB0*.

Since version 3.1.23 this setting can contain device candidates as comma separated list Candidates can be used with serial devices and in single modem configuration only. The first candidate which opens successfully is selected. For example: **device = /dev/ttyUSB3, /dev/ttyUSB2** causes that ttyUSB3 is tried first, and if it does not open, ttyUSB2 is used. The list can have many candidates.

device_open_alarm_after = number
Default value: *0*.
Available from version 3.1.7. After defined number of retries, an alarmhandler is called. Smsd still continues trying, if *device_open_retries* value is bigger.

device_open_errorsleeptime = number
Default value: *30*.
Available from version 3.1.7. Defines how many seconds the smsd will sleep after an error when trying to open the device.

device_open_preparator = filename
Default value: *not in use*.
Available from version 3.1.23. This setting defines a script which is executed every time before a modem port is opened. A special keyword **modemname** can be used, it's replaced with a name of modem. Arguments for the script are: $1 = name, $2 = device, $3 = description. The script can be used to adjust symlink which points to the actual device.

device_open_retries = number
Default value: *1*.
Available from version 3.1.7. Defines how many times smsd will retry when cannot open a device. When maximum number of retries is reached, modem process will call alarmhandler and stop. With value -1 smsd will retry forever.

e

top

eventhandler = filename
Default value: *empty*.
Specifies an eventhandler script like in the global part. Since version 3.1.16 special keyword **modemname** can be used, it's replaced with a name of modem. If you use this setting, then this modem will use its own individual eventhandler instead of the global one.

eventhandler_ussd = filename
Default value: *not in use*.
Available from version 3.1.7. This setting defines an eventhandler to use with USSD messages. Since version 3.1.16 special keyword **modemname** can be used, it's replaced with a name of modem. It is possible to use the same script or program which is used as *eventhandler*, but it's not a default because basically those scripts are not compatible without modifications.

After an USSD message is received, and probably *ussd_convert* is done, eventhandler_ussd is called. Smsd checks what is the current character set of a modem and creates a temporary file which contains the USSD answer. Arguments for the eventhandler_ussd are:

$1	"USSD" keyword.
$2	Filename (which contains the answer).
$3	Devicename.
$4	Character set.
$5	Command what was used to get the USSD answer.

Eventhandler_ussd can do whatever is needed with the USSD answer. It can also modify the answer, or delete the file. After eventhandler_ussd returns, smsd will check if the file still exists. If it exists, it's first line is taken as a new answer. Modified answer is then logged and probably printed to the regular_run_statfile.

g

top

give_up_on_io_error = yes/no
Default value: *yes*.
Available from version 3.1.23. If an error occurs when writing to the modem, process stops the current job, closes the modem and starts normal sleeping.

h

top

hangup_incoming_call = yes/no
Default value: *no*.
Available from version 3.1.5. If set to *yes* and detected unexpected input contains "RING", incoming call is ended. Use modem setting *voicecall_hangup_ath* to define if "ATH" is used to make hangup instead of "AT+CHUP". This setting overrides global setting.

i

top

ignore_unexpected_input = string
Default value: *none*.
Available from version 3.1.16. If the device continually sends some unexpected input, and there is not other way to get rid of it, suitable phrase can be defined to keep the log clean. Multiple phrases can be defined with multiple settings.

incoming = no/yes/high or 0/1/2
Default value: *no*.
Specifies if the program should read incoming SM from this modem. "Yes" or "1" means that smsd receives with less priority. The value "high" or "2" means that smsd receives with high priority. "No" or "0" means that smsd does not receive messages.

init = modem command
Default value: *not in use*.
Specifies a modem initialisation command. *init1* and *init* define both the same. See the manual of your modem for more details of modem commands.

init2 = modem command
Default value: *not in use*.
Specifies a second modem initialisation command. Most users do not need this.

internal_combine = yes/no
Default value: *use the global part setting which defaults to* ***yes***.
Available from version 3.0. Like in the global part, defines if multipart message is combined internally.

internal_combine_binary = no
Default value: *use the global part setting which defaults to global internal_combine which defaults to* ***yes***.
Available from version 3.1.5. Like in the global part, defines if multipart binary message is combined internally.

k

top

keep_messages = yes/no
Default value: *no*.
Available from version 3.1.7. Defines if messages are not deleted from the device. Unlike a global setting *keep_messages*, smsd continues running.

keep_open = yes/no
Default value: *yes*.
Available from version 3.1. If this is changed to *no*, a modem is closed while it's not used.

l

top

language = string or number
Default value: *none*.
language_ext = string or number
Default value: *none*.
Available from version 3.1.16. These settings set default values for National Language Shift Tables used in the outgoing message.
Value can be number, or variable length string which first macthes (case insensitive).
Choices are:

- 0 = basic
- 1 = Turkish
- 2 = Spanish
- 3 = Portuguese
- 4 = Bengali and Assemese, or Bengali, or Assemese
- 5 = Gujarati
- 6 = Hindi
- 7 = Kannada
- 8 = Malayalam
- 9 = Oriya
- 10 = Punjabi
- 11 = Tamil
- 12 = Telugu
- 13 = Urdu

logfile = filename
Default value: *empty, using a global log file*.
Available from version 3.1. Defines a log file if a global log is not used.

Since version 3.1.12 this setting can contain a special word "modemname" (without quotation marks). For example:
logfile = /var/log/smstools/*modemname*.log

When a device is *GSM1*, the setting is extracted as */var/log/smstools/GSM1.log*.

loglevel = number/word
Default value: *same as in the global setting*.
Available from version 3.1. Sets the verbosity of a log file. See more details in the global part.

loglevel_lac_ci = number/word
Default value: *LOG_INFO*.
Available from version 3.1.14. Sets the verbosity of logging the *Location area code* and *Cell ID* and their changes. This requires that AT+CREG? returns location information. It is automatically enabled by *pre_init* using a command *CREG=2*. After the Location are code or Cell ID changes, quality of signal is also logged. This feature can be disabled with the setting which is more than current loglevel, for example *8*.

log_not_registered_after = number
Default value: *0*.
Available from version 3.1.14. If it's known that the modem gives "not registered" after a message is sent or received, with this setting number of log messages can be avoided.

m top

max_continuous_sending = number
Default value: *300 (5 min)*.
Available from version 3.1.5. This setting is in seconds and defines how long modem can send messages without doing anything else. After *max_continuous_sending* time is reached, received messages are checked and other tasks are run. This setting overrides global setting.

Since version 3.1.18 continuous sending breaks if a signal USR2 is received. Sending continues as soon as receiving and other tasks are done.

memory_start = number
Default value: *1*.
Tells the first memory space number for received messages. This is normally 1, Vodafone Mobile Connect Card starts with 0.

messageids = number
Default value: *2*.
Available from version 3.1.1. Defines how message id's are stored: 1 = first, 2 = last, 3 = all. When all id's are stored, numbers are delimited whith space and there is one space and dot in the end of string.

message_limit = number
Default value: *not in use*.
message_count_clear = number
Default value: *not in use*.
Available from version 3.1. Message_limit specifies the maximum number of messages to send. After this limit is reached, no more messages will be sent until the smsd is restarted or message counter is cleared by the message_count_clear setting. The value of this setting is minutes.
If *admin_to* is specified, an administrative message is sent when the limit is reached.

mode = old/new
Default value: *new*.
Specifies version of modem command set. Almost everybody needs to use this as a "new".

Since version 3.0.9 this effects mainly to the sending side. In the receiving side the incoming PDU is checked, and if it does not match to the selected mode, another mode is tried automatically.

old	For Falcom A1 and maybe some other old modems of GSM phase 1 (1990-1995). In the receiving side this mode **does not have SCA** information in the begin of PDU.
new	For nearly all mobile phones and modems. In the receiving side this mode **has SCA** information in the begin of PDU.

(SCA stands for Service Centre Address).

modem_disabled = yes/no
Default value: *no*.
Available from version 3.0.9. This is for testing purposes too.
Whole messaging system including eventhandlers etc. can be tested without any working modem existing. Sending of messages is simulated in the similar way than with *sending_disabled* setting. Incoming messages are taken only from the file, if *pdu_from_file* is defined. No any communication is made between smsd and modem, but a device setting should still exist because smsd wants to open and close a device. If in you testing environment you do not have a priviledges to the usual modem device, like /dev/ttyS0, you can use a definition like *device = /tmp/modemfile*. If this file exists and is writable for the process owner, it's enough for smsd.

ms_purge_hours = number
Default value: *6*.
ms_purge_minutes = number
Default value: *0*.
ms_purge_read = yes/no
Default value: *yes*.
Available from version 3.1.5. These settings are used only with SIM600 (or compatible) modems (*check_memory_method* 5). If multipart message is received with one or more parts missing, incomplete message is removed from the message storage after time defined. Time is calculated from the timestamp of the first available part. Value 0 for both settings disables this feature. Setting *ms_purge_read* defines if parts are read before they are deleted. If *internal_combine* setting is used, parts are stored to the concatenation storage. If missing part(s) are later received, there will be similar timeout before parts are purged. After this the message is complete and is stored to the incoming folder. See also global settings *ic_purge_**.

n top

national_toa_unknown = yes/no
Default value: *no*.
Available from version 3.1.16. When destination number is national, some operators require that "unknown" format is defined.

needs_wakeup_at = yes/no
Default value: *no*.
Available from version 3.1.7. After being idle, some modems do not answer to the first AT command. For example with BenQ M32, there can be OK answer, but in many times there is not. To avoid error messages, smsd first send AT and read the answer if it's available.

notice_ucs2 = number
Default value: *2*.
Available from version 3.1.16. Message written using UTF-8 alphabet is tried to convert to GSM alphabet, to save sending costs. Any character outside the GSM alphabet is reported, if value of *notice_ucs2* is 2. If the value is at least 1, total number of missing characters is reported. Each character is presented in the sent message file using *NOTICE:* header. With cyrillic languages the conversion is always done, and the information is not important. Notification can be disabled with value 0.

o

top

outgoing = yes/no
Default value: *yes*.
Available from version 3.0.9. Specifies if a modem is used to handle and send outgoing messages.

p

top

pdu_from_file = filename / directoryname/
Default value: *not in use*.
Available from version 3.0. This is for testing purposes.
Since version 3.1.16 special keyword **modemname** can be used, it's replaced with a name of modem.
You can test you eventhandler and some other things without actually receiving a message from the modem/phone. This is especially important when it's not possible to receive the same message again because the sender cannot be reached.
You may have the original PDU string stored to the incoming message file, or you may see it in log file (depending of the loglevel). This PDU string can be stored to the *pdu_from_file* named file, and when this file exists the smsd will read the PDU from there. Rest processing will be done similarry than with normally received messages and you can then debug possible problems and see when they are fixed.
This file can contain empty lines and comment lines starting with # character.
Actual data can be stored as one line containing the PDU string, or two lines containing (first) the modem answer and (second) the PDU string.
For example:
```
#2006-09-13 13:12:10,7, GSM1: <-
+CMGR: 0,,40
07915318111111111240C915318325476980F160903131417421185474 7A0E4ACF416110BD3CA783DAE5F93C7C2EBB14
```
or simply one line only:
```
07915318111111106BC0C9153183254769860903131417421609031314184 2100
```

NOTE: After this file is processed, it is **removed**.
Since version 3.0.9 the setting can be a directory.
If this setting ends with a slash and a directory with that name exists, file(s) are read from this directory (and deleted after processing). All files found from the given directory are processed one by one, expect hidden files (name begins with a dot). When this setting points to the directory, no dot's are allowed in any position of a path. Be very careful with this setting while it will delete the content of a whole directory.
Since version 3.0.9: while reading a PDU from file, a first line starting with PDU: and space is taken if any exists.
Since version 3.1.5: this can be used only with check_memory_method values 0 or 1.

phonecalls = yes/no/clip
Default value: *no*.
Available from version 3.1. Specifies if missed phonecalls are reported. Since version 3.1.7 a value **clip**, or **2** can be used. This value could be used with modems which do not support reading of phonebook, or phobebook cannot be used because entries cannot be deleted. Phocecall is detected using the *+CLIP Calling line identification report* from a modem.

When *phonecalls = clip* is used, a setting *hangup_incoming_call* is automaticatty set to yes. This is because smsd must hangup a call before it's handled.

phonecalls_purge = yes/no/command
Default value: *no*.
Available from version 3.1.7. Specifies if missed phonecalls are removed using a purge command. Value *yes* specifies that **AT^SPBD="MC"** command is used, this works with some Siemens based devices. Other command can be given as a value.

pin = 4 digit number / ignore
Default value: *not in use*.
Specifies the PIN number of the SIM card inside the modem. See also *pinsleeptime*.
Note since version 3.1.1: Even if a PIN is not defined, it's still checked if a PIN is needed. Some phones may give an incorrect answer for the check when a pin is not needed, like *SIM PIN2* or *SIM PUK2* instead of *READY*. In this case define *pin = ignore* to skip the whole PIN handling procedure.

pinsleeptime = number
Default value: *0*.
Available from version 3.0.9. Specifies how many seconds the program will sleep after a PIN is entered. Some modems do not work without some delay.

phonecalls_error_max = number
Default value: *3*.
Available from version 3.1.7. Specifies the maximum number of errors before phonecalls are ignored.

poll_faster = number
Default value: *5*.
Available from version 3.1.16. This setting speeds up the polling, when it is known what to expect as an answer from the modem. Use the global setting *log_read_timing* to find out the best performance with your device.

pre_init = yes/no
Default value: *yes*.
Available from version 3.0.8. Specifies is an "echo off" and "CMEE=1" commands are sent to the modem before anything else is done.

primary_memory = memory name
Default value: *not in use*.
secondary_memory = memory name
Default value: *not in use*.
secondary_memory_max = number
Default value: *accept what device returns*.
These three settings are used to control **dual-memory handler**, available from version 3.0.
If your modem/phone receives messages to the Mobile Equipment (ME) memory after the SIM memory (SM) has been filled up, with dual-memory handler messages can be read from the Mobile Equipment memory too.
Defining secondary_memory_max is needed, if your device does not tell how much there is space in the Mobile Equipment memory. For example the Nokia 6210 does not tell (it returns 0 as max value) and with this device 150 is reasonable value.
Since version 3.1, multiple parameters can be defined for memories, like SM,SM,SM. Double-quotation marks are not necessary to use in the string.

priviledged_numbers = list of numbers
Default value: *not in use*.
Available from version 3.1.5. This setting overrides the setting in the global part. See the global part for details.

q

top

queues = list of queue names
Default value: *not in use*.
Specifies the Provider Queues that this modem shall serve. Use the same provider names as in [queues] and [providers]. If you do not use the provider-sorting feature, then leave this line out. Since version 3.1.5 special keyword **modemname** can be used, it's replaced with a name of modem.
Since version 3.1.16, when queues are used, this setting is no more mandatory if *outgoing* is disabled.

r

top

read_configuration_after_suspend = yes/no
Default value: *no*.
Available from version 3.1.18. If set to *yes*, modem process will read the configuration when suspend is going to break or end. Configuration is checked and modem port (*device*) is tested. If there are any problems, suspend continues. Problems are shown in the log, and after problems are fixed, a signal USR2 must be sent to the modem process. When configuration has no problems, modem process will continue as usual.

Changes for device settings *logfile* and *loglevel* do not apply without restarting smsd.

read_delay = number
Default value: *0*.
Available from version 3.1.16. Some modems may require small delay before reading is started. This time is in milliseconds.

read_identity_after_suspend = yes/no
Default value: *yes*.
Available from version 3.1.18. Defines if IMEI and IMSI are refreshed after the suspend has ended. This is useful if modem was suspended, and the SIM was changed on the fly, without restarting the whole daemon and without reading the whole modem setup.

read_timeout = number
Default value: *5*.
Available from version 3.1.5. When smsd reads data from a modem, timeout will occur after *read_timeout* seconds if an acceptable answer is not received. Some very slow devices might need greater value than 5 seconds.

read_timeout_XXX = number
Default value: *varies*.
Available from version 3.1.16. Defines a timeout for XXX operation. Use *report_read_timeouts = yes* to see possible operations (XXX) and current values.

This is for problematic environments, usually there is no need to change any value. Minimum value is 1. The value defines how many *read_timeout*'s are spent. The setting *read_timeout* is in seconds and defaults to 5.

receive_before_send = yes/no
Default value: *no*.
Available from version 3.1.17 as a modem setting. Without this setting a value of global setting is used. Forces smsd to empty the first SIM card memory before sending SM. This is a workaround for modems that cannot send SM with a full SIM card.

regular_run = filename
Default value: *not in use*.
Regular_run for a modem is available from version 3.1:
This setting defines an external script or program to execute. A modem can be used as it's closed by the smsd, if *regular_run_keep_open* is not set to *yes*.
Since version 3.1.16 special keyword **modemname** can be used, it's replaced with a name of modem.

regular_run_cmd = string
Default value: *empty*.
Like regular_run_cmdfile, this string can be used to define modem commands. This setting can be used more than once to define multiple commands.

regular_run_cmdfile = filename
Default value: *not in use*.
Since version 3.1.16 special keyword **modemname** can be used, it's replaced with a name of modem.

This file can contain command lines which smsd will write to the modem. Modem result of each line is logged and written to the regular_run_statfile (if defined). After a file is processed, it is removed. If you need to use permanent commands on each run, use regular_run script to create this file or define commands using regular_run_cmd settings.

Since version 3.1.12 the expected answer can be defined. By default smsd will expect "OK" or "ERROR" from the modem. Any other answer will cause timeout. Line can start with opening square bracket, and the expected answer can be given between square brackets. The expected answer must be a valid regular expression (see *man regcomp* for details).

Example:

> **[(>)|(ERROR)]AT+STGR=3,1;**
> Smsd is now expecting ">" or "ERROR" instead of "OK" or "ERROR".

regular_run_interval = number
Default value: *300*.
Describes number of seconds between each run. Value *0* disables this feature.

regular_run_keep_open = yes/no
Default value: *no*.
Available from version 3.1.16. This setting defines if a modem is kept open when modem process is executing external regular_run scripts.

regular_run_logfile = filename
Default value: *not in use*.
Defines a log file for regular_run. Syslog cannot be used for this. If a log file is not defined, smsd's main log is used. Since version 3.1.16 special keyword **modemname** can be used, it's replaced with a name of modem.

regular_run_loglevel = number
Default value: *LOG_NOTICE*.
Defines a level of logging.

regular_run_post_run = filename
Default value: *not in use*.
Available from version 3.1.7. This setting can define the second script or program which is executed regularly. Since version 3.1.16 special keyword **modemname** can be used, it's replaced with a name of modem. The same script with *regular_run* can be used. The script gets an argument *$1* which is *PRE_RUN* for *regular_run* and *POST_RUN* for *regular_run_post_run*. There is also the second argument *$2*, which includes a definition of *regular_run_statfile*.

This is how the regular_run for a modem currently works:

- If *regular_run* is defined, it's executed with arguments *PRE_RUN* and *regular_run_statfile*. A modem is closed while the script is running, depending on the setting *regular_run_keep_open*. *regular_run_statfile* is available from the previous run.
- *regular_run_statfile* is deleted.
- If *regular_run_cmdfile* is defined and the file exists, commands from this file are sent to the modem. If *regular_run_logfile* is defined, results are written to the file. If *regular_run_statfile* is defined, results are written to this file too. NOTE: *regular_run_cmdfile* is deleted after commands are handled. Use PRE_RUN phase or an external process to create this file.
- If *regular_run_post_run* is defined, it's executed with arguments *POST_RUN* and *regular_run_statfile*.
- If *regular_run_cmd*, one or more, is defined, commands are sent to the modem. Results are logged and written to the statfile, if defined.

Order of *regular_run_cmd* and *regular_run_post_run* has changed in version 3.1.16.

regular_run_statfile = filename
Default value: *not in use*.
If defined, results of commands are written to this file. Old file is cleared before each run. Since version 3.1.16 special keyword **modemname** can be used, it's replaced with a name of modem.

reply_path = yes/no
Default value: *no*.
Available from version 3.1.16. Defines a default value for Reply Path (TP-RP) field. Header in the message file overrides this value.

report = yes/no/disabled
Default value: *no*.
If you enable this, the program requests a status report SM from the SMSC for each sent message. Since version 3.1.16 this can have a value *disabled*, which means that the report is not requested even when the message file has a *Report: yes* header.

report_device_details = yes/no
Default value: *no/yes*.
Available from version 3.1.7. Defines if a details from device are printed to the log when modem process is starting. With beta versions of smsd this setting defaults to *yes*, otherwise it defaults to *no*.

report_read_timeouts = yes/no
Default value: *no*.
Available from version 3.1.16. Defines if all timeout values are written to the log when modem process starts.

routed_status_report_cnma = yes/no
Default value: *yes*.
Available from version 3.1.7. Defines if +CNMA acknowledgement is needed to send after routed status report was received.

rtscts = yes/no
Default value: *yes*.
You can disable usage of hardware handshake wires by setting this option to "no". Please don't use this feature in commercial applications because the hardware handshake wires ensure proper communications timing between the computer and the modem.

secondary_memory - see here

secondary_memory_max - see here

select_pdu_mode = yes/no
Default value: *yes*.
Available from version 3.1.16. Defines if PDU mode is selected before trying to send or read messages. NOTE that Smstools3 still works with PDU mode only. Ascii mode is too limited and problematic in many cases. Some modems work with PDU mode only, and do not support the command which selects the PDU mode. With this setting the command can be skipped. Also, little bit time can be saved when there is no need to select PDU mode every time.

send_delay = number
Default value: *0*.
If your modem does not support hardware handshake you should use the lowest possible baudrate to ensure that the program does not run faster than the modem can do. However, if the lowest possible baudrate is still too fast, then you can use this parameter to make it even slower. A value of 300 means that the program waits 300 milliseconds between sending each single character to the modem which makes the program very slow.
Since version 3.1.5, value 0 means that whole string is sent at once without any delays. This resolves slow communication with LAN based multiport servers, like Digi Portserver II. If, for some reason, "tcdrain" is still needed after sending, use value -1.

send_handshake_select = yes/no
Default value: *yes*.
Available from version 3.1.9. Instead of checking the flag TIOCM_CTS, select() function is used when the serial transmitter is busy. If because of some reason the old style should be used, this setting allows it.

send_retries = number
Default value: *2*.
Available from version 3.1.16. Defines how many times smsd will retry if sending fails.

sending_disabled = yes/no
Default value: *no*.
Available from version 3.0. This is for testing purposes.
You can test your eventhandler and whole system around the smsd without sending any messages to the GSM network. All other functionality is working as usual, so this is some kind of "mute" to the modem. However the modem should be connected and working. This does not have an effect to the incoming messages.

sentsleeptime = number
Default value: *0*.
Available from version 3.1.16. Some modems may need little delay after each part of SMS is sent. This time is in seconds.

signal_quality_ber_ignore = yes/no
Default value: *no*.
Available from version 3.1.14. Some devices do not support *Bit Error Rate* when signal quality is asked, and this always provides "Bit Error Rate: not known or not detectable" to the log line and to the status file. With this setting *ber* can be ignored.

smsc = number
Default value: *not in use*.
Specifies the SMSC number that this modem should use to send SM. You need this setting only if the default of the SIM card is bad. Write the phone number of the SMSC in **international format** without the starting "+".

smsc_pdu = yes/no
Default value: *no*.
Available from version 3.1.12. If the number of SMSC is set in the configuration, or in the message file, the number is included in the PDU instead of changing SMSC with a command AT+CSCA. The number can be presented in international format, or in national format starting with 0.

socket_connection_alarm_after = number
Default value: *0*.
Available from version 3.1.7. After defined number of retries, an alarmhandler is called. Smsd still continues trying, if *socket_connection_retries* value is bigger.

socket_connection_errorsleeptime = number
Default value: *5*.
Available from version 3.1.7. Defines how many seconds the smsd will sleep after an error with socket connection.

socket_connection_retries = number
Default value: *11*.
Available from version 3.1.7. Defines how many times smsd will retry when a socket connection fails. When maximum number of retries is reached, modem process will call alarmhandler and stop. With value -1 smsd will retry forever.

start = modem command
Default value: *not in use*.
startsleeptime = number
Default value: *3*
Available from version 3.1.7. If defined, *start* command is sent to the modem when a modem process starts. After a command is sent, *startsleeptime* is spent.

status_include_counters = yes/no
Default value: *yes*.
status_signal_quality = yes/no
Default value: *yes*.
Available from version 3.1.5. These settings define if message counters and explained signal quality is included in the line of *status* file. Modem setting overrides global setting.

stop = modem command
Default value: *not in use*
Available from version 3.1.7. When a modem process is stopping, *stop* command is sent to the modem.

t

top

telnet_cmd = string
Default value: *empty*.
telnet_cmd_prompt = string
Default value: *empty*.
Available from version 3.1.16. Some devices have more than one modem modules, and the Telnet interface wants to know what module to use. The interface may prompt something like "***module1, module2, state1, state2, info.***", as PORTech MV-372 does. For example the command ***module1*** should be sent to the interface. Use communication mode or some Telnet client to see what exactly is the prompt. Define it as *telnet_cmd_prompt*, exactly as it was received. Define suitable command as *telnet_cmd*. Whenever the prompt occurs in traffic, the cmd is sent and wanted module is selected.

telnet_crlf = yes/no
Default value: *yes*.
Available from version 3.1.16. Defines if CRLF is used instead of LF when communicating with Telnet. This defaults to *yes*, because many if not all Telnet servers require CRLF.

telnet_login = string
Default value: *empty*.
Available from version 3.1.12. If a network modem requires telnet login, it can be defined with this setting.

telnet_login_prompt = string
Default value: *login:*.
Available from version 3.1.12. If *telnet_login* is used, this setting can be used to change the login prompt.

telnet_login_prompt_ignore = string
Default value: *Last login:*.
Available from version 3.1.12. If *telnet_login* is used and after successful login motd contains a string which is the same as *telnet_login_prompt*, this setting can be used to define which kind of a string is ignored. For example, *telnet_login_prompt* can be *login:* and *telnet_login_prompt_ignore* could be *Last login:*.

telnet_password = string
Default value: *empty*.
Available from version 3.1.12. If *telnet_login* is used and device requires password, it can be defined with this setting.

telnet_password_prompt = string
Default value: *Password:*.
Available from version 3.1.12. If *telnet_password* is used, this setting can be defined to change the prompt for password.

text_is_pdu_key = string
Default value: *not in use*.
Available from version 3.1.16. In some systems the PDU is generated outside the smsd. As the modem is kept open by smsd, external program cannot use it. In this case external program can create SMS file for smsd, and in this file the message body is PDU in the hex format, and *To:* number should match with this key.

trust_spool = yes/no
Default value: *yes*.
Available from version 3.1.9. When a modem process is searching for a file from the spooler, it assumes that the file is complete when it exists and it's not locked. This will speed up sending, because one second delay is avoided. If some other process or checkhandler is creating files directly to the spooler, it may be necessary to set this to *no*.

u

top

unexpected_input_is_trouble = yes/no
Default value: *yes*.
Available from version 3.1.5. With smart_logging, this setting defines if unexpected input activates trouble log.

using_routed_status_report = yes/no
Default value: *no*.
Available from version 3.1.7. Smsd can detect routed status reports, but usually it's not recommended to use them. Modems should store status reports as an ordinary messages, which can be read when smsd will need them. However, some modem cannot store status reports, and therefore routing is the only way to get them. With this setting smsd will change some severity and text of a logging.

ussd_convert = number
Default value: *0*.
Available from version 3.1.7. Defines if a text part from incoming USSD message is decoded. Possible values are:

1	Unicode format is converted to UTF-8.
2	GSM 7bit packed format is converted to ISO or UTF-8.
4	Hexadecimal dump is converted to ASCII. (Available from version 3.1.11.)

Decoded text is appended to the original answer with two slashes and one space character.

v

top

verify_pdu = yes/no
Default value: *no*.

Available from version 3.1.14. This setting is for testing purposes. When trying to send SMS and modem does not accept the PDU, there may be a communication problem which breaks the sent PDU. With this setting smsd changes "echo on", and verifies the string which is echoed back after the PDU was sent. In case of mismatch "Verify PDU: ERROR" is printed to the log and also both sent and received strings are logged. In case of success "Verify PDU: OK" is printed to the log. After the verification is done, "echo" is changed back to "off".

voicecall_clcc = yes/no
Default value: *no*.
Available from version 3.1.12. Defines if AT+CLCC is used to detect when a call is answered. This is required if modem returns OK immediately after the call and *voicecall_cpas* cannot be used.

voicecall_cpas = yes/no
Default value: *no*.
Available from version 3.1.7. Defines if AT+CPAS is used to detect when a call is answered. This is required if modem returns OK immediately after the call.

voicecall_hangup_ath = yes/no
Default value: *no*.
Available from version 3.1.5. Defines if **ATH** is used to hangup call instead of **AT+CHUP**. This setting overrides the setting in the global part.

voicecall_ignore_modem_response = yes/no
Default value: *no*.
Available from version 3.1.5. When a voicecall is ringing, some devices give OK answer after couple of seconds even if a call is not yet answered. With this kind of device DTMF tones cannot be sent. If a ringing time is defined in the message file (using **TIME: n**), the waiting loop is breaked too soon. To avoid this, use *voicecall_ignore_modem_response = yes* in the modem settings. With this setting call rings n seconds (if not answered) and after this voicecall is over.

voicecall_init = modem command
Default value: *not in use*.
Available from version 3.1.23. Specifies the command to prepare a modem for a voice call.

voicecall_vts_list = yes/no
Default value: *no*.
Available from version 3.1.3. Defines how VTS command is used to create DTMF tones: *yes* = AT+VTS=1,2,3,4,5 (list is used), *no* = each tone is created with single command like AT+VTS="1";+VTS="2" etc.

voicecall_vts_quotation_marks = yes/no
Default value: *no*.
Available from version 3.1.7. Defines if quotation marks are used when sending VTS command to the modem. NOTE: previously quotation marks were used, now this setting default to *no*.

W

top

wakeup_init = string
Default value: *empty*.
Available from version 3.1.16. Defines a string or command which is sent to the device when it is initialised, before anything else is done. Any or missing answer is accepted, and smsd continues after a small delay.
