

ПО для осуществления массовых SMS-рассылок "SMS-Sender"

Содержание

1. Общее описание.....	4
2. Технические характеристики и системные требования.....	4
3. Установка, запуск и управление.....	6
3.1. Смена режима управления. Systemd-mode.....	9
4. Конфигурирование.....	10
4.1. [LICENSING].....	11
4.2. [MAIN].....	11
4.3. [SOCKET_TIMEOUTS].....	13
4.4. [DGRAM_PARAMS].....	13
4.5. [UDP_PARAMS].....	14
4.6. [UDP_PORTS].....	14
4.7. [UDP_ALLOW_IP].....	14
4.8. [CONNECTS].....	15
4.9. [CONNECTS_SMPP_DEFIN_TO_DSM].....	16
4.10. [CONNECT_TIMEOUTS].....	16
4.11. [CONNECT_ALIVE_PARAMS].....	17
4.12. [LOG_PARAMS_FOR_CONNECT].....	18
4.13. [SMPP_READ_PDU_SPEED].....	19
4.14. [RESP_CODES_FOR_RECONNECT].....	20
4.15. [SENDER_N_COMMON_TIMEOUTS].....	20
4.16. [SUBMIT_SM_TIMEOUTS].....	21
4.17. [FILE_QUEUE_CONTROL].....	22
4.18. [FILE_QUEUE_CONTROL_HELP_SEND].....	23
4.19. [CONNECT_SPEED].....	24
4.19.1. Максимальная суммарная скорость отправки.....	24
4.19.2. Максимальная скорость отправки SMS для определённого smpp-коннекта.....	25
4.20. [CONNECTS_SYNONYM_TYPE].....	26
4.21. [CONNECTS_SYNONYM_SRC].....	27
4.22. [CONNECTS_SYNONYM_DST].....	29
4.23. [RX_COMMON_TIMEOUTS].....	29
4.24. [DSM_COMMON_TIMEOUTS].....	30
4.25. [PROCESSING_DELIVER_SM_PARAMS].....	31
4.26. [DSM_NOTIF_FUNC_TEXT].....	31
4.27. [DSM_NOTIF_REPORT_LISTS].....	34
4.28. [SMPP_RESP_CODES].....	35
4.29. [LATIN_CHANGE].....	35
5. Отправка SMS. Общие сведения.....	35
5.1. Отправка SMS. Формат строки-команды.....	36
5.1.1. Описание возможных опций параметра r_deliv0.....	37
5.2. Отправка SMS. Формат имени файла-задания.....	38
6. Результат обработки отчётов о доставке.....	39
6.1. Результат обработки отчётов о доставке. Общий результат-файл.....	40
6.2. Результат обработки отчётов о доставке. Специальный результат-файл.....	43

7. Логирование. Общие сведения.....	46
7.1. Логирование. SMPP-трафик.....	47
7.2. Логирование. UDP-трафик.....	48
7.3. Логирование. DGRAM-трафик.....	49
8. UDP-запросы к сервисам.....	49
9. DGRAM-запросы к сервисам.....	51
10. Дополнительный функционал на основе ПО.....	52
10.1. Простой SMS-чат (simple_chat).....	52
10.1.1. Файл конфигурации simple_chat.cfg (описание).....	52
11. Используемые в ПО модули Perl5.....	56

1. Общее описание

ПО SMS-Sender предназначено для отправки коротких сообщений посредством протокола SMPP 3.4 и обработки отчётов о доставке (если при отправке короткого сообщения был выставлен соответствующий флаг) с сохранением результатов в машино-понятном виде для последующей обработки.

Взаимодействие с ПО для отправки коротких сообщений (SMS) осуществляется посредством файловой очереди, где строка в файле (в определённом формате) является заданием на отправку SMS. Данная схема обеспечивает простую и быструю интеграцию с любым сторонним сервисом.

Исходя из функционала ПО делится на три взаимосвязанных типа:

- 1) *рассыльщик* (sms_sender_N, где N – номер потока, начиная с 1);
- 2) *ресивер* (sms_sender_Nrx, где N – номер потока, начиная с 0);
- 3) *обработчик отчётов о доставке* (sms_sender_Ndsm, где N – номер потока, начиная с 0).

ПО является условно-бесплатным (shareware) и может использоваться на безвозмездной основе, но со следующими ограничениями:

- 1) разрешён запуск только sms_sender_1.exe (рассыльщик), sms_sender_0rx.exe (ресивер) и sms_sender_0dsm.exe (обработчики отчётов о доставке);
- 2) суммарная скорость отправки (для sms_sender_1.exe) равна 5 смс/сек.

ПО не имеет ограничений на запуск (в пределах shareware-ограничений) в виртуальных машинах.

Для написания ПО использовался ЯП Perl5.

2. Технические характеристики и системные требования

Системные требования:

- 1) операционная система CentOS7 (RHEL7-based) или AlmaLinux 8 (RHEL8-based);
- 2) kernel 3.10 (дефолтовое ядро для RHEL7-based) или kernel 4.18 (дефолтовое ядро для RHEL8-based);

- 3) наличие на целевом хосте компилятора gcc;
- 4) локаль en_US.UTF-8;
- 5) от 1Gb RAM;
- 6) от 1 vCPU (виртуальный ЦПУ), если запуск планируется на виртуальной машине.

Рассыльщик (sms_sender_N):

- 1) максимальная скорость рассылки ~ 2500 смс в секунду (подразумевается submit_sm в секунду) на один поток (например, sms_sender_1) с одним smpp-соединением (т.е. большее кол-во потоков = большая совокупная скорость рассылки);
- 2) настраиваемая скорость отправки для каждого конкретного smpp-соединения;
- 3) динамическое понижение скорости отправки при получении от целевого smpp-коннекта (SMS-центра) заранее определённых (в файле конфигурации) кодов ошибок;
- 4) простая отправка SMS-сообщений (SMS-центр статично определён);
- 5) отправка SMS с использованием функций выбора целевого smpp-коннекта (любой доступный, по номеру отправителя, по номеру получателя);
- 6) отложенная отправка (с указанием конкретной даты начала рассылки);
- 7) настраиваемая совместная обработка очередей сообщений различными приложениями-рассыльщиками (например, sms_sender_1 + sms_sender_3);
- 8) настраиваемый контроль smpp-соединений (отправка/приём enquire_link/enquire_link_resp);
- 9) возможно исполнение функции ресивера (т.е. осуществление приёма отчётов о доставке);
- 10) настраиваемая скорость приёма входящего трафика (отчёты о доставке и проч.) для каждого smpp-соединения;
- 11) возможность скрывать содержание входящих и исходящих сообщений в лог-файлах для определённого smpp-коннекта;
- 12) настраиваемые условия (по коду ошибки в resp-ax) автоматического переподключения к smpp-соединению.
- 13) возможность автоматической транслитерации содержания отправляемых сообщений.

Ресивер (sms_sender_Nrx):

- 1) максимальная скорость приёма служебных сообщений (submit_sm_resp, deliver_sm, etc) ~ 3000 ед. в секунду (в большинстве случаев скорость приёма всегда выше скорости отправки) на один поток ресивера с одним smpp-коннектом;
- 2) настраиваемый контроль smpp-соединений (отправка/приём enquire_link/enquire_link_resp);
- 3) настраиваемая скорость приёма входящего трафика (отчёты о доставке и проч.) для каждого smpp-соединения;
- 4) возможность скрыть содержание входящих сообщений в лог-файлах для определённого smpp-коннекта;
- 5) настраиваемые условия (по коду ошибки в resp-ax) автоматического переподключения к smpp-соединению.

Обработчик отчётов о доставке (sms_sender_Ndsm):

- 1) скорость обработки входящего трафика с информацией о статусе доставки ~ 30-50 ед. в секунду;
- 2) возможность отправки служебных sms-сообщений (через sms_sender_N) абоненту-отправителю по факту доставки/недоставки сообщения абоненту-получателю;
- 3) возможность изменять номер получателя и содержание служебных сообщений в зависимости от номера DSM-потока (например, sms_sender_1dsm) и smpp-соединения, на которое было отправлено сообщение с флагом о подтверждении доставки;
- 4) возможность отправки запроса о статусе доставки (query_sm), если истёк срок ожидания отчета о доставке (в виде deliver_sm).

3. Установка, запуск и управление

Первый запуск:

- 1) создаем (под root-ом) директорию /opt/ext_services (mkdir /opt/ext_services). Директорию можно выбрать на своё усмотрение;
- 2) меняем владельца директории /opt/ext_services (chown some_user:some_user /opt/ext_services). Опционально;
- 3) меняем пользователя (su - some_user) на того, которого назначили владельцем директории /opt/ext_services. Опционально;

4) переходим в директорию /opt/ext_services (cd /opt/ext_services);

5) скачиваем архив с ПО -

wget "https://github.com/vladimir-chursin000/sms_sender/raw/master/sms_sender.zip" (для RHEL7-based)

или

wget "https://github.com/vladimir-chursin000/sms_sender/raw/master/sms_sender_alma8.zip" (для RHEL8-based)

(в дальнейшем вводим команды с учётом выбранного архива);

6) даём команду на распаковку архива (unzip sms_sender.zip). В результате распаковки появится директория /opt/ext_services/sms_sender;

7) переходим в директорию /opt/ext_services/sms_sender (cd /opt/ext_services/sms_sender);

8) производим первый запуск ПО посредством команды "./sms_sender_1.exe". В результате на экран терминала будет выведена строка

"fail [self_sender_cfg_reader] --> fail. ipv4='some_ipv4' is not correct",

а в директории с исполняемым файлом появится директория cfg_files (/opt/ext_services/sms_sender/cfg_files);

9) переходим в директорию **cfg_files**, открываем на запись файл **sms_sender.cfg** (общий файл конфигурации сервисов), находим строку "self_ip=some_ipv4" и вместо "some_ipv4" вписываем IP-адрес хоста;

10) переходим в директорию с исполняемым файлом ("cd .." или "cd /opt/ext_services/sms_sender") и снова запускаем сервис (./sms_sender_1.exe). В результате сервис sms_sender_1.exe успешно запустится, а в директории /opt/ext_services/sms_sender (корневая директория сервиса) появятся ещё несколько поддиректорий: **cfg_history** (история изменений файла конфигурации sms_sender.cfg), **proc_control** (сервисная папка), **queues** (очереди команд на отправку/обработку), **sms_sender_dgram** (сервисная папка с dgram-сокетами), **sms_sender_log** (папка с файлами логов работы сервиса). Также в корневой директории сервиса можно будет увидеть 2 файла: **sms_sender_1.cfg_notice** и **sms_sender_1.lic_inf**. Первый содержит рекомендации по изменению файла конфигурации, а второй – статусы обращений к системе лицензирования.

Установка ресивера и обработчика отчётов о доставке:

1) переходим в корневую директорию сервиса (`cd /opt/ext_services/sms_sender`);

2) для создания ресивера (`sms_sender_0rx.exe`) исполняем команду

`"cp sms_sender_1.exe sms_sender_0rx.exe";`

3) для создания обработчика отчётов о доставке (`sms_sender_0dsm.exe`) исполняем команду

`"cp sms_sender_1.exe sms_sender_0dsm.exe";`

4) аналогичным образом возможно создать дополнительные потоки
рассылщика/ресивера/обработчика отчётов:

`sms_sender_N.exe`, `sms_sender_Nrx.exe`, `sms_sender_Ndsm`, где N – порядковый номер сервиса.

Управление (на примере `sms_sender_1.exe`):

1) переходим в корневую директорию сервиса (`cd /opt/ext_services/sms_sender`);

2) `"/sms_sender_1.exe"` – запуск сервиса;

3) `"/sms_sender_1.exe stop"` – останов;

4) `"/sms_sender_1.exe restart"` – перезапуск;

5) `"/sms_sender_1.exe reload_cfg"` – перечитать файл конфигурации и применить изменения (без перезапуска сервиса);

6) для останова и перезапуска без возможной потери трафика предусмотрены команды

`"/sms_sender_1.exe save_stop"` и `"/sms_sender_1.exe save_restart"`, обеспечивающие сохранение информации из рабочих программных очередей (ассоциативных массивов) на время неактивности/перезапуска ПО;

7) возможно управление сервисом без перехода в корневую директорию, например,

`"/opt/ext_services/sms_sender/sms_sender_1.exe stop"`.

3.1. Смена режима управления. Systemd-mode

Переключение режима управления со стандартного на **systemd-mode** (на примере

sms_sender_1.exe):

- 1) переходим в директорию сервиса (`cd /opt/ext_services/sms_sender`);
- 2) выполняем команду `./sms_sender_1.exe use_systemd`. В директории сервиса должна появиться поддиректория **for_systemd**, внутри которой будет находиться папка **sms_sender_1**;
- 3) переходим в директорию **"for_systemd/sms_sender_1"** (`"cd for_systemd/sms_sender_1"` или `"/opt/ext_services/lic_serv/for_systemd/sms_sender_1"`). Тут можно обнаружить 3 файла: **sms_sender_1.service** (unit-файл для systemd), **enable_using_systemd.sh** (скрипт перехода к управлению сервисом через systemd), **disable_using_systemd.sh** (скрипт возвращения к стандартному управлению сервисом);
- 4) правим файл **sms_sender_1.service**, если это необходимо. Не рекомендуется изменять следующие параметры: Type, ExecStartPre, ExecStart, ExecReload, WantedBy;
- 5) от имени root (или от имени пользователя с правами, позволяющими конфигурировать systemd) запускаем скрипт **enable_using_systemd.sh**. Перед запуском необходимо остановить сервис **sms_sender_1.exe** (`./sms_sender_1.exe stop`), если он был запущен ранее.

После запуска **enable_using_systemd.sh** стандартное управление становится недоступно, а управление сервисом будет возможно исключительно через systemd:

- 1) `"systemctl start sms_sender_1"` – запустить сервис;
- 2) `"systemctl restart sms_sender_1"` – перезапустить сервис;
- 3) `"systemctl reload sms_sender_1"` – перечитать файл конфигурации;
- 4) `"systemctl status sms_sender_1"` – запросить статус;
- 5) `"systemctl stop sms_sender_1"` – остановить сервис.

Примечание 1. Для перехода обратно на стандартную схему управления необходимо: а) остановить сервис; б) запустить от имени рута скрипт ***disable_using_systemd.sh***.

Примечание 2. Если меняется директория сервиса (например, с `"/opt/ext_services/sms_sender"` на `"/opt/services/sms_sender"`), то шаги с 1-го по 5-й потребуется повторить.

4. Конфигурирование

Конфигурирование сервиса смс-рассылок осуществляется посредством правки файла конфигурации **`cfg_files/sms_sender.cfg`** (находится в корневой директории сервиса).

Структура файла конфигурации основана на логических блоках, выделенных с помощью строк вида **`"[ИМЯ_ЛОГИЧЕСКОГО_БЛОКА]"`** (1-ая строка блок "открывает", 2-ая – "закрывает"). Также возможно в тексте конфигурационного файла оставлять комментарии и пояснения посредством символа **`"#"`** в начале строки.

По умолчанию файл конфигурации содержит поясняющие комментарии касательно возможностей ПО.

4.1. [LICENSING]

Данный блок предназначен для связывания экземпляров ПО (sms_sender_N, sms_sender_Nrx, sms_sender_Ndsm) с агентом сервера лицензий (lic-serv-agent или LSA) и получения лицензионной ёмкости (submit_sm/sec) для рассыльщика (sms_sender_N) или возможности запуска (sms_sender_Nrx / sms_sender_Ndsm). Ресиверы и обработчики отчетов с нулевых идентификатором (sms_sender_0rx и sms_sender_0dsm) не требуют конфигурирования в блоке LICENSING.

Чтобы экземпляр ПО обращался при запуске к агенту сервера лицензий, необходимо (в рамках конф. блока) прописать строку вида "**IP_LSA:UDP_PORT_LSA proc_id NUM**", где **IP_LSA** – ip-адрес (v4) агента сервера лицензий во внутренней сети, **UDP_PORT_LSA** – udp-порт агента сервера лицензий, **proc_id** – идентификатор экземпляра ПО (например, рассыльщик sms_sender_1.exe имеет **proc_id=1**, ресивер sms_sender_1rx.exe – **proc_id=1rx**, обработчик отчётов sms_sender_1dsm.exe – **proc_id=1dsm**), **NUM** – желаемый объём лицензионной ёмкости.

Наглядный пример конфигурации.

```
[LICENSING]
#lic_serv_agent - adres agenta servera licenzij v lokalnoj seti
###
#proc_id - libo nomer sender-a (1,2,3, etc), libo id obrabotchika otchetov o dostavke (1dsm, 2dsm, etc),
#libo id resivera otchetov o dostavke (1rx, 2rx, etc).
#Obrabotchik otchetov s id "0dsm" vnosit v etot blok konfiguracii ne trebuetsya.
#Resiver otchetov s id "0rx" vnosit v etot blok konfiguracii ne trebuetsya.
###
#desired_lic_count - zhelaemoe kolichestvo licenzionnoj emkosti (sms v sekundu) na odin sender.
#Dannyy parametr opredelyaet maksimalnoe znachenie #max_common_speed dlya #proc_id v bloke [CONNECT_SPEED].
#Neobhodimoe kol-vo licenzij na odin potok Ndsm (obrabotchik otchetov) ili Nrx (resiver otchetov) = 1.
###
#lic_serv_agent(ip:port)           #proc_id       #desired_lic_count
10.9.8.7:7766   1           100
10.9.8.7:7766   1rx        1
10.9.8.7:7766   1dsm       1
[LICENSING]
```

4.2. [MAIN]

Блок содержит ряд настроек, определяющих директории и методы взаимодействия с ПО. Сами настройки представляют собой пары вида "ключ=значение". Каждый параметр имеет описательную часть в виде комментария.

log_file_write_buf = 1 (или 0). Параметр управления буферизацией вывода для лог-файлов.

node_id=1 (или любое другое число). Идентификатор ПО, присутствующий в имени лог-файла.

self_ip=ipv4. IP-адрес хоста, где работает ПО.

log_dir=_STD_DIR_/sms_sender_log/. Параметр, определяющий директорию для записи лог-файлов. Тег "_STD_DIR_" – директория, где расположен исполняемый файл сервиса. Если директория log_dir расположена в _STD_DIR_, то она создаётся автоматически при запуске (если не была создана до этого).

check_logs_fh_timeout=3. Период проверки (в секундах) лог-файлов сервиса. Если лог-файл не существует, то происходит его автоматическое пересоздание.

dgram_dir=_STD_DIR_/sms_sender_dgram/. Служебная директория для размещения dgram-сокетов сервиса. Если располагается в _STD_DIR_, то создаётся автоматически.

submit_sms_dir=_STD_DIR_/queues/submit_sms_queue/. Директория для размещения заданий на отправку SMS (submit_sm). Описание формата файлов и их содержания - отдельным пунктом.

query_sm_dir=_STD_DIR_/queues/query_sm_queue/. Служебная директория для отправки query_sm (запрос статуса доставки короткого сообщения), поступающих от сервиса sms_sender_0dsm (обработчик отчётов о доставке).

dsm_queue_dir=_STD_DIR_/queues/dsm_queue_dir/. Служебная директория для передачи информации (от рассыльчиков или ресиверов) обработчику отчётов о доставке.

dsm_result_dir=_STD_DIR_/queues/dsm_result_dir/. Директория для записи результатов обработки отчётов о доставке.

done_submit_sms_dir=def. Директория для обработанных файлов очереди на отправку SMS. Если данный параметр равен "**def**", то обработанные файлы очереди сохраняются в директории "**submit_sms_dir/done**".

done_query_sm_dir=def. Директория для хранения обработанных заданий на отправку query_sm. Если данный параметр равен "**def**", то обработанные файлы очереди сохраняются в директории "**query_sm_dir/done**".

done_dsm_queue_dir=def. Директория хранения исполненных запросов к обработчику отчётов о доставке. Если данный параметр равен "**def**", то обработанные файлы очереди сохраняются в директории "**dsm_queue_dir/done**".

dsm_result_write_by_hour=0 (или 1). По умолчанию запись результатов обработки отчётов (в sms_sender_Ndsm) происходит раз в N секунд (и каждый раз создаётся новый файл с данными), где N определяется в блоке [DSM_COMMON_TIMEOUTS] (параметр #**fin_save_res_t**). Если параметр dsm_result_write_by_hour выставить равным 1, то поведение DSM-сервиса (обработчик отчётов) можно изменить таким образом, чтобы результаты группировались по файлам с почасовой разбивкой.

wr_inf2fqueue_timeout=1. Таймаут (в секундах) записи служебной информации в директории с файлами-очередями для обмена информацией между рассыльщиками, ресиверами и обработчиками отчётов.

read_inf_dsm_queue_timeout=1. Таймаут (в секундах) чтения служебной информации (submit_sm_resp-ы и deliver_sm-ы) из директории **dsm_queue_dir** обработчиком отчётов о доставке.

ch_n_recr_fq_timeout=10. Таймаут (в секундах) проверки на существование служебных директорий (submit_sms_dir и т.д.), включая done-директории (например, done_submit_sms_dir). Если какая-либо директория отсутствует, то она будет создана автоматически.

try2create_serv_dirs_at_start_anyway=0 (или 1). По умолчанию, если служебные директории расположены (через файл конфигурации) в корневой директории сервиса (например, submit_sms_dir=_STD_DIR_/queues/submit_sms_queue/), то их создание при старте сервиса происходит автоматически. Если же (через файл конфигурации) указать директорию, отличную от стандартных путей, то создание служебных подпапок – задача администратора сервиса. Поведение сервиса в этом случае можно изменить, если задать **try2create_serv_dirs_at_start_anyway=1**, не забыв при этом выдать необходимые права (chmod / chown / selinux) на целевую директорию пользователю, из-под которого происходит запуск ПО.

done_q_files_exp_time=0. Время (в секундах) хранения обработанных файлов очереди в done-директориях. "0" – хранение без ограничений (необходимо удалять вручную). ">0" (например, 30) – удаление файлов очереди, обработанных более N секунд назад. "del" – не хранить обработанные задания. Функцию, связанную с данным параметром, исполняет рассыльщик с id=1 (т.е. sms_sender_1.exe).

4.3. [SOCKET_TIMEOUTS]

Блок конфигурации отвечает за управление поведением сокетов (udp, dgram, smpp) в плане скорости обработки входящего трафика. Конфигурация (в рамках блока) представляет собой строку "**proc_id udp_dgram_value smpp_value**", где **proc_id** – идентификатор сервиса (например, 1, 0rx, 0dsm), **udp_dgram_value** – значение таймаута для udp и dgram, **smpp_value** – значение таймаута для smpp-сокета.

Пример конфигурации.

```
[SOCKET_TIMEOUTS]
#Umenshenie taimauta uvelichivaet skorost obrabotki vhodyashego trafika, pri etom uvelichivaetsya zagruzka CPU
#proc_id      #udp_dgram      #smpp
1              0.1           0.01
0rx            0.1           0.01
0dsm           0.1
[SOCKET_TIMEOUTS]
```

Для обработчика отчётов о доставке нет необходимости указывать smpp_value.

4.4. [DGRAM_PARAMS]

Блок связан с редактированием параметров dgram-соединения: размер буфера вх. трафика и возможность скрыть содержание входящих / исходящих dgram-сообщений. Конфигурация (в рамках блока) выглядит так – "**proc_id recv_buffer is_hide_data**", где **proc_id** – идентификатор сервиса (sms_sender_1.exe, sms_sender_0rx.exe, sms_sender_0dsm.exe), **recv_buffer** – буфер вх. трафика, **is_hide_data** – если "1", то сообщения (входящие, исходящие) в логах сервиса будут скрыты посредством стандартного текста.

Значения по умолчанию (если для конкретного `proc_id` нет конфигурации в `DGRAM_PARAMS`): **recv_buffer** = 16384 (байт), **is_hide_data** = 0.

Пример конфигурации.

```
[DGRAM_PARAMS]
#если для какого-либо proc_id параметры не определены, то выставяутся def-параметры: recv_buffer=16384, hide_data=0
#recv_buffer: buffer (вроде в baitah) для операции socket->recv
#hide_data: если 1, то данные (in/out) в логah скрываются
#proc_id      #recv_buffer  #hide_data
1             16384        0
0rx           16384        0
0dsm          16384        0
[DGRAM_PARAMS]
```

4.5. [UDP_PARAMS]

Конфигурация, аналогичная `[DGRAM_PARAMS]`, но для UDP.

4.6. [UDP_PORTS]

Указание сервису с определённым `proc_id` открыть при запуске определённый UDP-порт для приёма служебного трафика.

```
[UDP_PORTS]
#proc_id: если sms_sender_NUMBER.pl - sender, *_0rx.pl (или Nrх.pl, где N=номер tx-потокa) - приёмщик deliver_sm,
#* 0dsm.pl (или Ndsm.pl, где N=номер tx-потокa) - сервис обработки deliver_sm
#UDP-порты для мониторинга по udp и посылки каких-либо сигналов
#proc_id      #self_udp_port
#1            some_udp_port
#0rx          some_udp_port
#0dsm         some_udp_port
[UDP_PORTS]
```

4.7. [UDP_ALLOW_IP]

Конфигурация предназначена для ограничения возможности обращения к сервису по UDP. Настройка представляет собой строку вида "**proc_id ip_or_list_of_ip**", где **proc_id** – идентификатор сервиса, а **ip_or_list_of_ip** – отдельный ip-адрес (ipv4) или список адресов, разделённых запятой. Если не вносить в блок настроек по конкретному `proc_id`, то по UDP смогут обращаться любые хосты без ограничений.

Пример конфигурации.

```
[UDP_ALLOW_IP]
#allow_ip: можно через запятую
#proc_id      #allow_ip
1            10.10.9.9
0rx          12.13.5.6
0dsm         172.6.7.9
[UDP_ALLOW_IP]
```

4.8. [CONNECTS]

Блок отвечает за конфигурирование smpp-коннектов, подключение к которым происходит при запуске рассыльщиков (sms_sender_N.exe) и ресиверов (sms_sender_Nrx.exe). Строка конфигурации коннекта - “**proc_id** **syn_params** **host** **port** **login** **pass** **con_type**”.

proc_id – идентификатор рассыльщика (sms_sender_N.exe). Если вместо конкретного идентификатора (например, 1, 2 или 3) использовать слово "all", то подключаться к данному smpp-хосту будет каждый запущенный рассыльщик.

syn_params – строка вида “**dst:defs+src:defs**”, где defs (definitions) может принимать значения “**int** / **unk** / **alpha**”. Например, запись "**dst:int,unk,alpha+src:int,unk,alpha**" означает, что возможно через данный коннект отправлять SMS на федеральные номера (**int**), короткие номера (**unk**) и номера символьные (**alpha**), подписываясь федеральными (**int**), короткими (**unk**) или символьными номерами (**alpha**). Возможность отправки на те или иные номера (как и возможность подписываться теми или иными номерами) зависит от настроек на стороне SMS-центра.

host и **port** – ip-адрес (v4) SMS-центра и соответствующий TCP-порт.

login и **pass** – учётные данные для подключения к SMS-центру.

con_type – тип коннекта. Возможные значения: **tx** (без приёма отчётов о доставке, т.е. только отправка), **tx0** (приём отчётов о доставке осуществляется ресивером с **proc_id=0rx**), **txN** (приём отчётов о доставке осуществляется ресивером с **proc_id=Nrx**, где N = 1, 2, 3, etc), **trx** (приём отчётов о доставке осуществляется через тот же коннект, через который происходит отправка SMS). При выставлении данного параметра необходимо учитывать наличие разрешения (или возможности) со стороны SMS-центра принимать соединения по схеме TX + RX, т.е. когда отправка и приём возможны с разных коннектов при одних и тех же учётных данных.

Пример конфигурации.

```
[CONNECTS]
#syn_params=stroka vida "dst:defs+src:defs". gde defs mozhet primimat znacheniya "int/unk/alpha"
#naprimer zapis "dst:int,unk,alpha+src:int,unk,alpha" nominalno oznachaet, chto mozhno otpravlyat
#sms na federalnye nomera(int), korotkie nomera (unk) i simvolnye nomera (alpha)
#i ispolzovat podpis int,unk,alpha
###
#v proc_id=0rx iniciiruyutsya konnekty na priem sluzhebnogo trafika (deliver_sm s otchetom o dostavke, alarm_notification, etc),
#esli tip konnekta (s id=N) raven tx0
###
#con_type: tx - tolko otpravka (bez priema v proc_id=rx), tx0-otpravka zherez id=N + priem cherez id=0rx,
#trx - priem i otpravka v odin konnekt. Esli zhe con_type=txN, to priem oshestvlyaetsya v rx-konnekt s id=Nrx (pomimo nulevogo)
###
#esli proc_id=all, to danniy konnekt ispolzuyut vse potoki sendirov
###
#proc_id      #syn_params      #host      #port      #login      #pass      #con_type
#1      dst:int+src:unk 192.168.0.3      9900      GSMGW      secret      trx
1      dst:int,unk,alpha+src:int,unk,alpha      192.168.1.5      7755      login1      pass1      trx
all     dst:int,unk,alpha+src:int,alpha      192.168.1.6      7755      login2      pass2      trx
2      dst:int,unk,alpha+src:int,unk      192.168.1.5      7755      login3      pass3      tx
3      dst:int,alpha+src:int,unk      192.168.1.8      7756      login4      pass4      tx0
3      dst:int,alpha+src:int,unk      192.168.1.8      7757      login5      pass5      tx2
[CONNECTS]
```


Непосредственно с данным блоком связан важный составной параметр **smpp_defin** (smpp definition), который используется во многих других настройках и состоит из ip хоста (**host**), порта (**port**) и логина на SMS-центре (**login**). **smpp_defin=host:port:login**.

4.9. [CONNECTS_SMPP_DEFIN_TO_DSM]

Данный блок настроек служит для перенаправления отчетов о доставке (deliver_sm) и информации о статусе отправки (submit_sm_resp) на конкретный обработчик отчётов (sms_sender_Ndsm) в зависимости от **smpp_defin** (host:port:login). Для привязки **smpp_defin** к определённому обработчику отчётов необходимо прописать два параметра, разделённые пробелом или символом табуляции, – "**smpp_defin dsm-stream_num**", где **dsm-stream_num** – это числовой идентификатор обработчика отчётов о доставке (например, для sms_sender_2dsm.exe таким идентификатором будет 2).

Пример конфигурации.

```
[CONNECTS_SMPP_DEFIN_TO_DSM]
#Dannaya nastroyka ukaziyvaet na to, kakoj potok dsm-servisa obrabatyvaet deliver_sm konkretnogo smpp-konnekta (#s
#Esli dlya nekotorigo smpp-konnekta dsm-servis ne ukazan, to po umolchaniyu ispolzhuetsya DSM-servis s id=0(0dsm).
#smpp_defin      #dsm-stream_num
192.168.168.120:7444    2
192.168.168.121:7444    3
[CONNECTS_SMPP_DEFIN_TO_DSM]
```

Если для какого-либо **smpp_defin** не назначен определённый dsm-сервис (сервис обработки отчётов), то по умолчанию используется dsm-сервис с нулевым идентификатором (т.е. sms_sender_0dsm.exe).

4.10. [CONNECT_TIMEOUTS]

Блок конфигурации позволяет изменять таймауты взаимодействия (для каждого отдельного **smpp_defin**) рассылщиков и ресиверов с SMS-центром (или SMS-центрами).

Строка конфигурации выглядит так (в общих чертах) -

"smpp_defin wait_bind_resp_t enq_link_t wait_enq_link_resp_t reconnect_if_err_t".

smpp_defin = smsc_ip:smc_port:smc_login (**host:port:login** из блока [CONNECTS]).

wait_bind_resp_t – время ожидания (в секундах) ответа на bind-запрос (запрос авторизации) от SMS-центра. Если за это время ответ от другой стороны не получен, то происходит повторная попытка подключиться к SMS-центру.

enq_link_t – таймаут отправки (в секундах) запроса вида "**enquire_link**" (подтверждение активности коннекта) на SMS-центр через соединение **smpp_defin**.

wait_enq_link_resp_t – время ожидания (в секундах) ответа на **enquire_link** (т.е. ожидаем enquire_link_resp). Если в указанный промежуток времени (с момента отправки) ответ получен не был, то увеличивается на единицу внутренний счётчик **no_enq_link_resp_count**, связанный с

параметром **max_no_enq_link_r_cnt** из блока [CONNECT_ALIVE_PARAMS]. Когда **no_enq_link_resp_count** > **max_no_enq_link_r_cnt**, происходит реконнект к SMS-центру (с учётом соответствующего **smpp_defin**).

reconnect_if_err_t – время ожидания (в секундах) между попытками реконнекта, если не удалось подключиться или возникли ошибки, требующие переподключения к SMS-центру.

Если оставить данный блок конфигурации незаполненным, то будут применены значения по умолчанию: **wait_bind_resp_t** = 5 (сек.), **enq_link_t** = 60 (сек.), **wait_enq_link_resp_t** = 10 (сек.), **reconnect_if_err_t** = 30 (сек.).

Пример конфигурации.

```
[CONNECT_TIMEOUTS]
#wait_bind_resp_t - v sekundah. Vremya ozhidaniya resp-a na bind. Esli vremya isteklo, to rekonnekt
#enq_link_t - (v sekundah). Opredelyaet period otpravki (kazhdye Nsec) enq_link
#wait_enq_link_resp_t - v sekundah. Esli enq_link_resp ne prishel za eto vremya, to evelichivaetsya schetchik no_enq_link_resp_count
#reconnect_if_err_t - v sekundah. Ozhidanie mezhdu popytkami rekonnekta, esli obnaruzheny oshibki
#smpp_defin      #wait_bind_resp_t      #enq_link_t      #wait_enq_link_resp_t      #reconnect_if_err_t
192.168.156.55:12555:login_smsc 10      120      10      60
[CONNECT_TIMEOUTS]
```

4.11. [CONNECT_ALIVE_PARAMS]

Здесь можно задать параметры поддержания smpp-соединения.

Шаблон строки конфигурации (параметры разделены пробелами или символами табуляции) –

"smpp_defin max_no_enq_link_r_cnt max_resp_err_cnt allow_external_enq_link any_traf_alive".

smpp_defin = smsc_ip:smsc_port:smsc_login (**host:port:login** из блока [CONNECTS]).

max_no_enq_link_r_cnt – лимит "потерянных" enquire_link_resp-ов подряд, после достижения которого требуется произвести переподключение к SMS-центру (исходя из **smpp_defin**).

max_resp_err_cnt – лимит resp-ов (для любых операций) со статусом, не равным 0 (если "пришла" ошибка со стороны SMS-центра), полученных подряд, после достижения которого требуется произвести переподключение к SMS-центру (исходя из **smpp_defin**).

allow_external_enq_link – если выставить "1", то приходящие от SMS-центра enquire_link-и будут сдвигать время отправки enquire_link (от рассыльщика / ресивера) на SMS-центр.

any_traf_alive – если выставить "1", то любой входящий трафик со стороны SMS-центра будет сдвигать время отправки enquire_link-ов на SMS-центр.

Значения по умолчанию (если конкретный smpp_defin не был сконфигурирован в данном блоке):

max_no_enq_link_r_cnt = 5, **max_resp_err_cnt** = 15, **allow_external_enq_link** = 0, **any_traf_alive** = 0.

Пример конфигурации.

```
[CONNECT_ALIVE_PARAMS]
#smpp_defin=host:port:login
#max_no_enq_link_r_cnt: kol-vo poteryannyh enq_link_resp podryad, posle kotoryh nuzhen rekonnekt
#max_resp_err_cnt: kol-vo resp!=0 (po lyubym operaciyam) podryad, posle kotoryh nuzhen rekonnekt
###
#allow_external_enq_link: esli "1", to vhodyshie enq_link ot smpp-servera sdvigayut vremya posylki enq_link na etot smpp_host
#udalyaet tekushie dannye po ozhidaniyu respov na enq_linki dlya dannogo smpp_defin
###
###
#any_traf_alive: esli "1", to lyuboj vhodyashij trafik ot smpp-server sdvigaet vremya posylki enq_link na smpp-server
#udalyaet tekushie dannye po ozhidaniyu respov na enq_linki dlya dannogo smpp_defin
###
#smpp_defin      #max_no_enq_link_r_cnt  #max_resp_err_cnt  #allow_external_enq_link  #any_traf_alive
192.192.4.5:8876:some_login1      10      30      1      1
192.192.4.5:8776:some_login2      11      33      1      0
[CONNECT_ALIVE_PARAMS]
```

4.12. [LOG_PARAMS_FOR_CONNECT]

Конфигурирование настроек логирования для каждого smpp-коннекта.

Шаблон строки конфигурации -

"smpp_defin hide_msg extra_log save_pdu_dump".

smpp_defin = smsc_ip:smc_port:smc_login (**host:port:login** из блока [CONNECTS]).

hide_msg – если выставить равным "1", то содержание исходящих (submit_sm) и входящих сообщений (deliver_sm) в логах скрывается (в рамках конкретного коннекта).

extra_log – если "1", то активируется дополнительное логирование.

save_pdu_dump – если "1", то все входящие пакеты (со структурой) сохраняются в соответствующую поддиректорию **log_dir** (имя поддиректории содержит **smpp_defin**).

Дефолтовые значения (если не конфигурировать **smpp_defin** в данном блоке): **hide_msg** = 0, **extra_log** = 0, **save_pdu_dump** = 0.

Пример конфигурации.

```
[LOG_PARAMS_FOR_CONNECT]
#hide_msg: esli 1, to vse sodержimoe abon-trafika v logah skryvaetsya. Naprimer, tam cho-to sekretnoe :)
#extra_log: pisat extra-logirovanie dlya smpp-soedinenij. Esli 1, to pishetsya v log dop. informaciya iz PDU-read
#save_pdu_dump: esli 1, to vse vhodyashie pdu sohranyayutsya d sootvetstvuyushyuy poddirektoriyu "log_dir"
#smpp_defin      #hide_msg      #extra_log      #save_pdu_dump
10.6.7.8:9999:login1      1      0      0
10.6.7.8:9998:login2      0      1      1
[LOG_PARAMS_FOR_CONNECT]
```

4.13. [SMPP_READ_PDU_SPEED]

Отвечает за усреднённую скорость приёма вх. трафика (responses, deliver_sm, etc).
Конфигурирование возможно как для отдельных smpp-коннектов, так и для рассыльщиков (sms_sender_N.exe) / ресиверов (sms_sender_Nrx.exe).

Чтобы задать максимальную общую скорость приёма для рассыльщика / ресивера, необходимо вписать строку – "**proc_id max_common_speed**", где **proc_id** – идентификатор рассыльщика / ресивера, а **max_common_speed** – максимальная общая скорость приёма трафика (ед. / сек) для конкретного рассыльщика / ресивера.

Конфигурация скорости приёма трафика для отдельного smpp-коннекта -

"smpp_defin max_speed".

smpp_defin = smsc_ip:smc_port:smc_login (**host:port:login** из блока [CONNECTS]).

max_speed – скорость (ед. / сек) приёма трафика отдельным коннектом smpp.

Особенности:

- 1) суммарная скорость приёма на всех smpp-коннектах (**smpp_defin**) в рамках конкретного smpp-сервиса (рассыльщика и ресивера) должна быть \geq скорости приёма, заданной для smpp-сервиса (рассыльщика и ресивера);
- 2) если максимальная общая скорость не задана, то она выставляется равной сумме скоростей всех smpp-коннектов, активных на конкретном сервисе (sms_sender_N.exe / sms_sender_Nrx.exe);
- 3) если для конкретного smpp-коннекта скорость приёма не задана, то по умолчанию она равна скорости отправки (блок [CONNECT_SPEED]);
- 4) если скорость приёма < 100 ед. / сек. (для конкретного **smpp_defin**), то значение поднимается до 100;
- 5) режим максимальной производительности на приём трафика конфигурируется посредством значения "**no_lim**". При этом существует две схемы использования "**no_lim**": а) **no_lim** выставлен как для сервиса (рассыльщик и / или ресивер), так и для определённого smpp-коннекта (или коннектов); б) **no_lim** выставлен для определённого smpp-коннекта, но не назначен соответствующему сервису (рассыльщик и / или ресивер). В первом случае (а) коннект(ы) с настройкой **no_lim** работают на приём трафика с макс. производительностью без условной верхней планки, а во втором – верхней планкой является макс. общая скорость (**max_common_speed**) приёма сервиса (sms_sender_N.exe / sms_sender_Nrx.exe).

```
[SMPP_READ_PDU_SPEED]
#Usrednennaya skorost priema vhodyashego trafika. Konfiguriruetsya kak dlya potokov N/Nrx, tak i dlya smpp-konnektov,
#ispolzuemyh v kazhdom iz potokov N/Nrx. Izmeryaetsya v rx-responses/sec.
#####
#Dlya vystavleniya obshej skorosti priema izpolzuetsya stroka vida "proc_id      max_speed", gde proc_id raven nomeru potoka otpravki N
#ili nomeru potoka priema Nrx. Maksimalno vozmozhnaya skorost priema na odin potok mozhet dostigat 4600 rx-responses/sec
#pri ispolzovanii processora Intel(R) Core(TM) i3-2100 CPU @ 3.10GHz (4 yadra) i RAM 8G DDR3. Veroyatno, na bolee moshnom oborudovanii
#mozhet imet bolshie znacheniya.
#Summarnaya skorost priema vseh smpp-konnektov na konkretnom potoke (N/Nrx) dolzha >= obshej skorosti priema na potok (N/Nrx).
#Esli obshaya skorost priema ne zadana, to po umolchaniyu ona ravna summe skorostej priema dlya kazhdogo smpp_defin.
#nizhe - primery
#0rx      500
#1        200
#####
#Dlya skorosti priema konkretnogo smpp-konnekta izpolzuetsya stroka vida "smpp_defin  max_speed", gde smpp_defin=ip:port:login
#Esli skorost priema konkretnogo smpp-konnekta ne zadana, to po umolchaniyu ona ravna skorosti otpravki ([CONNECT_SPEED]).
#Esli skorost priema dlya konkretnogo smpp_defin menshe 100, to skorost priema = 100.
#nizhe - primer
#ip:port:login 100
#####
#Vazhno! Esli neobhodim rezhym maksimalnoj proizvoditelnosti na priem trafika, to vystavlyaem paramtr "no_lim" na nuzhnye potoki i/ili
#konkretnye smpp-konnekty. Esli paramtr "no_lim" dlya konkretnogo smpp-konnekta ne ustanovlen, to dejstvuet zadannoe ogranichenie.
#Esli dlya konkretnogo smpp-konnekta vystavljen "no_lim", no sam potok imeet ogranichenie, to smpp-konnekt budet imet sushestvennoe prei
#na priem trafika po skorosti obrabotki v ramkah dannogo potoka
#####
#Vazhno! Chem vyshe paramtr skorosti obrabotki vhodyashego trafika, tem bolshe zagruzka CPU pri nepreryvnom trafike
[SMPP_READ_PDU_SPEED]
```

4.14. [RESP_CODES_FOR_RECONNECT]

Позволяет задать дополнительные условия для переподключения к определённому smpp-соединению (помимо обозначенных в блоке [CONNECT_ALIVE_PARAMS]). Функционал основан на анализе поля "status" в ответе (**resp**) на какой-либо smpp-запрос (например, submit_sm / submit_sm_resp).

Шаблон строки конфигурации – “**smpp_defin resp_name list_of_resp_codes**”.

smpp_defin = smsc_ip:smc_port:smc_login (host:port:login из блока [CONNECTS]).

resp_name – имя ответного “пакета” от SMS-центра, например, *submit_sm_resp*.

list_of_resp_codes – список цифровых кодов, разделённых запятой (без пробелов), из ответа (например, *submit_sm_resp*) на определённый запрос (например, *submit_sm*).

Пример конфигурации.

```
[RESP_CODES_FOR_RECONNECT]
#Opciya/nastroyka (dlya smpp-soedinenij) neobhodima v teh sluchayah, kogda zaranee izvestno, chto po faktu prihoda resp-
#dlya konkretnogo resp_name (naprimer, submit_sm_resp) iz spiska neobhodim rekonnekt
###
#list_of_resp_codes: cherez zapyatuyu i bez probelov
#smpp_defin  #resp_name      #list_of_resp_codes
192.55.66.77:5432:some_login  submit_sm_resp  6,55
192.55.66.78:5431:some_login  submit_sm_resp  6,56
[RESP_CODES_FOR_RECONNECT]
```

4.15. [SENDER_N_COMMON_TIMEOUTS]

Конфигурирование различных таймаутов, связанных с общей логикой работы рассыльщика (sms_sender_N.exe).

Шаблон конфигурации – “**proc_id p0 p1 p2 p3 p4**”.

proc_id – идентификатор экземпляра ПО. Например, рассыльщик `sms_sender_1.exe` имеет **proc_id=1**, ресивер `sms_sender_1rx.exe` – **proc_id=1rx**, обработчик отчетов `sms_sender_1dsm.exe` – **proc_id=1dsm**).

p0 (enq_link_c_t) - таймаут (в секундах) проверки списка активных коннектов на предмет необходимости отправки **enquire_link**-а.

p1 (recr_smppiosel_err) - таймаут (в секундах) пересоздания селектора (структура данных, содержащая ссылки на инициализированные smpp-коннекты), если селектор вышел из рабочего состояния.

p2 (com_smpp_sta_err) – таймаут (в секундах) проверки smpp-соединений на предмет необходимости реконнекта после ошибки.

p3 (r_subm_sm_t) - таймаут (в секундах) чтения файловых очередей **submit_sm** и **query_sm** (директории **submit_sms_dir/query_sm_dir** в блоке [MAIN]), где очередь **submit_sm** - пользовательский трафик для отправки SMS, а очередь **query_sm** - служебная очередь (для рассыльщика - **sms_sender_N.exe**) на отправку запросов статуса доставки **submit_sm**, формируемая сервисом обработки отчетов о доставке (**sms_sender_Ndsm.exe**).

p4 (resend_if_smpp_defin_status0) – таймаут (в секундах) повторной отправки SMS в файловую очередь, если целевой коннект (smpp_defin) недоступен (и это стало известно до попытки отправки).

Дефолтовые значения (если не конфигурировать рассыльщик): **enq_link_c_t** = 1, **recr_smppiosel_err** = 3, **com_smpp_sta_err** = 2, **r_subm_sm_t** = 1, **resend_if_smpp_defin_status0** = 120.

Пример конфигурации.

```
[SENDER_N_COMMON_TIMEOUTS]
#raznye taimauty, svyazannye s obshej logikoj servisa sms_sender_N
#p0=enq_link_c_t - v sekundah. Taimaut prohozhdeniya po spisku konnektov na predmet neobходимosti otpravki enq_link
#p1=recr_smppiosel_err - recreate_smpp_io_select_if_error. v sekundah. Taimaut peresozdaniya select-var, v kotoruyu pomeshayutsya
#smpp_h
#p2=com_smpp_sta_err - v sekundah. Taimaut proverki connects na neobходимost smpp-rekonnekta posle oshibki
#p3=r_subm_sm_t - v sekundah. Taimaut chteniya failovyh ocheredej submit_sm/query_sm (MAIN:submit_sms_dir/query_sm_dir).
#Format imeni faila dlya submit_sm: "name.submN" (N=nomer potoka). Query_sm = sluzhebnyj trafik ot Ndsm
#p4=resend_if_smpp_defin_status0 - v sekundah. Taimaut na povtornuyu popytku postavit v ochered otpravku sms, esli konkretnyj
#smpp_defin nedostupen
#proc_id      #p0      #p1      #p2      #p3      #p4
1             1       4       1       2       122
1             2       3       1       2       125
[SENDER_N_COMMON_TIMEOUTS]
```

4.16. [SUBMIT_SM_TIMEOUTS]

Конфигурирование таймаутов, связанных с отправкой **submit_sm**.

Общий шаблон – "**smpp_defin** **submit_err_resend_t** **bad_resp_resend_t**".

smpp_defin = `smc_ip:smc_port:smc_login` (**host:port:login** из блока [CONNECTS]).

submit_err_resend_t – таймаут (в секундах) между попытками отправки **submit_sm**, если произошла ошибка при отправке **submit_sm**.

bad_resp_resend_t – таймаут (в секундах) между попытками отправки **submit_sm**, если **resp_code** не равен "0" (поле "status" в ответном **submit_sm_resp**).

Значения по умолчанию (если не конфигурировать smpp-коннект в этом блоке конфигурации):
submit_err_resend_t = 5, **bad_resp_resend_t** = 5.

Пример конфигурации.

```
[SUBMIT_SM_TIMEOUTS]
#submit_err_resend_t - v sekundah. Taimaut mezhdy popytkami submit_sm, esli sluchilas oshibka pri otravke submit_sm
#bad_resp_resend_t - v sekundah. Taimaut mezhdy popytkami submit_sm, esli resp!=0
#smpp_defin      #submit_err_resend_t      #bad_resp_resend_t
192.168.10.10:9812:login123      10      15
[SUBMIT_SM_TIMEOUTS]
```

4.17. [FILE_QUEUE_CONTROL]

Блок конфигурации, отвечающий за использование оперативной памяти на этапе чтения файлов-заданий.

Период проверки директории-очереди (**submit_sms_dir**) на наличие файлов-заданий на отправку SMS задаётся в блоке [SENDER_N_COMMON_TIMEOUTS] (p3=r_subm_sm_t).

Общий шаблон конфигурации – "**proc_id** **p0** **p1**".

proc_id – идентификатор экземпляра рассылщика (например, для **sms_sender_1.exe** **proc_id**=1).

p0 (read_queue_file_by_file) – параметр, отвечающий за количество файлов-заданий, которое за раз считывает рассылщик (**sms_sender_N.exe**), для последующей отправки SMS (**submit_sm**). Если равен "0", то рассылщик будет читать все файлы-задания, находящиеся в директории-очереди **submit_sm_dir**.

p1 (max_file_size2read) – параметр, отвечающий за максимальное количество строк (одна строка = одно задание на отправку SMS), которое рассылщик сможет за один раз считать из файлов-заданий. Если параметр равен "0", то ограничение максимальное количество считываемых строк отсутствует.

Значения по умолчанию: **read_queue_file_by_file** = 0, **max_file_size2read** = 0.

Пример конфига.

```
[FILE_QUEUE_CONTROL]
#Period chteniya failov s komandami na otpravku sms iz direktorij-ocheredej opredelyaetsya v bloke
#[SENDER_N_COMMON_TIMEOUTS] (p3=r_subm_sm_t).
#Chem menshe period, tem bystree programma obnaruzhit faily-ocheredi.
###
#p0=read queue file by file. Esli >0, to faily-zadaniya na otpravku sms iz direktorij file-queue chitayutsya i obrabatyvayutsya
#posledovatelno po N shtuk za raz, gde N=p0. Znachenie po umolchaniyu p0=0 (t.e. obrabatyvayutsya vse faily,
#nahodyashiesya v direktorii-ocheredi)
###
#p1=max file size2read. Maksimalnoe kol-vo strok v chitaemyh failah-ocheredyah. Esli maksimum prevyshen, to takie faily
#"razrezayutsya" postrochno v sootvetstvii s maksimalnym kol-vom strok p1.
#Chem vyshe znachenie, tem vyshe ispolzovanie operativnoj pamyati. Znachenie po umolchaniyu=0 (bez ogranichenij).
#Esli vystavit 0, to razmer failov ocheredi proveryatsya ne budet (est risk prevysit dostupnyj razmer RAM).
#Vazhno! Poka tekushaya porciya dannyh (v sootv. s p0 i p1) obrabatyvaetsya, ostalnye faily-ocheredi ozhidayut obrabotki.
#Nastroiki p0 i p1 ispolzuyutsya v tom chisle i dlya bloka [FILE_QUEUE_CONTROL_HELP_SEND].
###
#proc_id      #p0      #p1
1              0        0
[FILE_QUEUE_CONTROL]
```

4.18. [FILE_QUEUE_CONTROL_HELP_SEND]

Блок конфигурации, позволяющий включить режим поддержки при обработке файлов-заданий, т.е. исполнение файлов-заданий возможно не только тем рассыльщиком, которому непосредственно назначено задание (посредством указания идентификатора рассыльщика в расширении файла, например, **".subm1"** для sms_sender_1.exe), но и рассыльщиком, указанным в данном блоке.

Общий шаблон конфигурации – **"proc_id p0 p1 p2"**.

proc_id – идентификатор экземпляра рассыльщика (например, для sms_sender_1.exe proc_id=1).

p0 (help_send) – параметр, определяющий список рассыльщиков, которые поддерживаются рассыльщиком с **proc_id**. Возможные значения: а) **"all"** – рассыльщик с идентификатором **proc_id** осуществляет поддержку всех работающих на данный момент рассыльщиков; б) **"list:2,3"** (без пробелов) – рассыльщик с идентификатором **proc_id** (например, 1) осуществляет поддержку рассыльщиков с идентификаторами 2 и 3.

p1 (check_help_queue_timeout) – таймаут (в секундах) проверки файловой очереди (директории-очереди) на предмет наличия файлов-заданий, обработку которых необходимо произвести.

p2 (expire_time) – время (в секундах), по истечении которого считается, что требуется помощь в обработке файла-задания. Например, был создан файл "some_file.subm2" (для обработки рассыльщиком sms_sender_2.exe) с командами на отправку SMS в "2018-10-15 22:05:30", но с момента создания прошло больше **"p2 секунд"**.

Особенности:

1) рассыльщик (допустим, с **proc_id=1**, т.е. sms_sender_1.exe), настроенный в качестве резервного, должен иметь коннекты (блок [CONNECTS]), аналогичные тем, к которым коннектиться резервируемый рассыльщик (допустим, с **proc_id=2**, т.е. sms_sender_2.exe). Например, если sms_sender_2.exe коннектиться к SMS-центру А1 с разрешением отправлять sms

на федеральные номера с возможностью подписываться только федеральными номерами (т.е. `syn_params="dst:int+src:int"`), то и `sms_sender_1.exe` должен быть подключён к SMS-центру (возможно, что не к A1, а, допустим, к A2) с использованием настройки "`dst:int+src:int`" (блок [CONNECTS]);

2) параметр `p1 (check_help_queue_timeout)` не может быть меньше "2";

3) параметр `p2 (expire_time)` не может быть меньше "3";

4) если файл-задание относится к типу "отложенная отправка" (т.е. формат имени файла-задания имеет вид "`YYYYMMDDHHMISSafr-some_str.subm1`"), то время, когда резервирующий рассыльщик может взять на обработку файл-задание, отсчитывается от "`YYYYMMDDHHMISS`" (`YYYY` – год, `MM` – месяц, `DD` – день, `HH` – час, `MI` – минута, `SS` – секунда).

По умолчанию, если не вносить изменений в данный блок конфигурации, каждый рассыльщик обрабатывает только свой файлы-задания (в соответствии с расширением "`.submN`", где `N` – `proc_id` рассыльщика).

Пример конфигурации.

```
[FILE_QUEUE_CONTROL_HELP_SEND]
#Esli net konfiguracii dlya proc_id, to kazhdyy sender (potok) obrabatyvaet tolko svoyu ochered (.submN).
#Esli konfiguraciya dlya opredelennogo potoka prisutstvuet, to imeet mesto byt rezervirovanie :)
###
#p0=help_send.
#Vozmozhnye varianty dlya p0 (help send):
#1) "all" (bez kavyчек). Osushestvlenie podderzhki vseh senderov, t.e. esli lyboj iz skonfigurirovannyh potokov po kakoj-to prichine
#ne uspevaet obrabatyvat ochered, to eto sdelayet potok c id=proc_id. V etom sluchae potok N (proc_id) dolzhen byt podklychen k tem zhe
#(ili k konnektam s analogichnymi razresheniyami na otppravku). Razresheniya vystavlyuyutsya so storony SMSC
#(naprimer, sotovogo operatora);
#2) "list:2,3,..." (bez kavyчек), gde 2,3 - nomera senderov, t.e. potok s id=#proc_id prinimaet uchastie v obrabotke ocheredej
#dlya sms_sender_2 i sms_sender_3.
###
#p1=check_help_queue_timeout. V sekundah. Periodichnost proverki failovoj ocheredi na predmet nalichiya failov-zadaniy
#(dlya drugih potokov), obrabotku kotoryh neobhodimo osushestvit.
###
#p2=expire_time (v sekundah). Vremya, posle istecheniya kotorogo schitaetsya, chto trebuetsya pomosh v obrabotke faila-zadaniya.
#Naprim, sozdan fail s komandami na otppravku sms queue.subm2 (dlya sendera N2) v 2018-10-15 22:05:30, no proshlo uzhe #p1 sekund,
#t.e. tekushee vremya="2018-10-15 22:05:30"+"#p2 sec". Dlya otlozhennoj otpravki za vremya sozdaniya faila beretsya data iz imeni faila
#YYYYMMDDHHMISSafr-some_str.subm1
#Esli p0 opredelen (t.e. servis rassylki s #proc_id skonfigurirovan v etom bloke), to #p2 ne mozhet byt menshe 3, a #p1 - menshe 2.
#proc_id      #p0      #p1      #p2
1             all      5        30
2             list:3    2        3
4             list:5,6  3        5
[FILE_QUEUE_CONTROL_HELP_SEND]
```

4.19. [CONNECT_SPEED]

Блок конфигурации, позволяющий выставить: а) максимальную суммарную скорость (для последующего распределения по smrr-коннектам) на один экземпляр рассыльщика (`sms_sender_N.exe`); б) максимальную скорость отправки SMS (`submit_sm / сек`) для конкретного smrr-соединения.

4.19.1. Максимальная суммарная скорость отправки

Шаблон конфигурации максимальной суммарной скорости на экземпляр рассыльщика – "`proc_id max_common_speed`".

proc_id – идентификатор экземпляра рассыльщика (например, для sms_sender_1.exe proc_id=1).

max_common_speed – макс. суммарная скорость на экземпляр рассыльщика.

Пример конфигурации максимальной суммарной скорости на экземпляр рассыльщика.

```
[CONNECT_SPEED]
#####Konfigurirovanie obshej maksimalnoj skorosti rassylki (submit_sm/query_sm) na odin potok rassylshika
###Esli max_common_speed dlya potoka ne vystavlena (pri otsutstvii konfiguracii v [LICENSING]), to po umolchaniyu ona ravna 5 sms/sec.
#Maksimalno vozmozhnaya skorost otpravki (pri idealnyh usloviyah) na odin ekzempliar rassilshika (odin potok) mozhet
#dostigat 2800-3000 sms/sec pri tehnikeskikh harakteristikah: Intel(R) Core(TM) i3-2100 CPU @ 3.10GHz (4 yadra) + 8G RAM DDR3.
#Veroyatno, na bolee moshnom oborudovanii etot pokazatel mozhet imet bolshie znacheniya.
#Vazhno! Chem vyshe skorost otpravki, tem vyshe zagruzka CPU pri nepreryvnoj otpravke sms.
#Sootvetstvenno, net trafika - net vysokoj zagruzki CPU.
#Vazhno! #max_common_speed pervonachalno zadaetsya v bloke [LICENSING].
#proc_id      #max_common_speed
1             50
2             55
3             60
#####
```

Особенности:

- 1) максимальная суммарная скорость первоначально задаётся в блоке [LICENSING] (при наличии действующей лицензии);
- 2) если максимальная суммарная скорость для рассыльщика не выставлена, то по умолчанию она равна 5 submit_sm / сек.

4.19.2. Максимальная скорость отправки SMS для определённого smpp-коннекта

Шаблон конфигурации -

"smpp_defin p0 p1 p2 p3 p4 p5 p6".

smpp_defin = smsc_ip:smc_port:smc_login (host:port:login из блока [CONNECTS]).

p0 (max_speed) – максимальная скорость отправки (submit_sm / сек) через коннект **smpp_defin**.

p1 (is_variative_speed) – параметр включения возможности динамически изменять скорость отправки при получении определённых статусов в ответном **submit_sm_resp**. Если параметр равен "1", то опция включена.

p2 (min_speed) – нижний предел уменьшения скорости отправки при получении целевых статусов, сообщающих о необходимости снизить скорость, в ответном **submit_sm_resp**.

p3 (variative_downgrade_step) – шаг уменьшения скорости рассылки (в submit_sm / sec).

p4 (downgrade_by_one_step_timeout) – таймаут (в секундах) понижения / повышения скорости. Например, был получен статус submit_sm_resp-а (определяется в параметре **p6**), скорость была понижена на значение **p3** и задано время "текущее время + **p4** секунд" (**T0**). Следующее

понижение скорости возможно при двух условиях (одновременно): а) **T0** меньше текущего времени; б) поступление нового submit_sm_resp-а со статусом, определённым в параметре **p6**.

p5 (dngrd_t) – длительность (в минутах) работы при пониженной скорости отправки. Если, например, в resp-статусе был получен код, обозначенный в параметре **p6**, то стартует таймер в **p5** минут, отсчитывающий время до того момента, когда возможно, если отсутствуют ошибки из **p6**, постепенно (в соответствии с параметром **p4**) повышать скорость отправки на шаг **p3**.

p6 (s_limit_rc) – список resp-статусов (через запятую, например "88,89,90"), при получении которых необходимо снижать скорость. Если опираться на спецификацию smpp 3.4., то таким кодом обычно является 88 (ESME has exceeded allowed message limits), хотя возможны и другие варианты, связанные с реализацией SMS-центров от конкретных вендоров.

Пример конфигурации скорости отправки (для smpp-коннектов).

```
#####Konfigurirovanie skorosti rassylki dlya konkretnogo smpp-konnekta
#p0=max_speed. Summa max_speed vseh smpp-konnektov, skonfigurirovannyh na odin potok, ne dolzhna prevyschat max_common_speed,
#vystavlennoye dlya potoka.
##Esli skorost dlya smpp-konnekta ne zadana, to po umolchaniyu ona ravna 1 sms/sec
#p1=is_variative_speed: esli "1", to otslezhivayutsya oshibki prevysheniya propuskn. sposobnosti i skorost vremennno umenshaetsya
#p2=min_speed - minimalnoe znachenie skorosti rassylki, nizhe kotorogo ne proishodit ponizhenie skorosti
#p3=var_downgrd_step=variative_downgrade_step - shag umensheniya skorosti pri oshibkah prevysheniya (s resp-ah)
#p4=downgrade_by_one_step_timeout. V sekundah. Taimaut ponizheniya (pri prihode resp-koda iz p6)/povysheniya skorosti
#Parametr p4 opredelyaet pлавное umenshenie skorosti pri prihode resp-koda iz p6 i pлавное ee povyshenie posle istecheniya
###
#p5=dngrd_t - downgrade timeout. Vremya v minutah, na kotoroe umenshaetsya propusknaya sposobnost pri resp_kodah p6
#p6=s_limit_rc - speed limit resp code. Mozhno cherez zapyatyuy, naprimer, "88,some_code1,some_code2"
#smpp_defin
#p0      #p1      #p2      #p3      #p4      #p5      #p6
192.168.5.6:6677:login1 100      0      0      0      0      0
192.168.5.7:6677:login2 150      1      50     5      15     5      88,89
[CONNECT_SPEED]
```

4.20. [CONNECTS_SYNONYM_TYPE]

Блок конфигурации, завязанный на параметр **syn_params** из блока [CONNECTS]. Позволяет задать режим использования smpp-коннектов для рассылщика с идентификатором **proc_id** (например, sms_sender_1.exe).

Шаблон конфигурации – "**proc_id** **type**".

proc_id – идентификатор экземпляра рассылщика (например, для sms_sender_1.exe proc_id=1).

type – тип обработки файлов-заданий, основанный на сходстве конфигураций различных smpp-коннектов по параметру "**syn_params**" и наличию значения "any_connect" в строке (одна строка в определённом формате – это задание на отправление одного короткого сообщения) файла-задания на рассылку (описание формата строк и имён файлов-заданий будет дано позже). Когда type равен "0", отправка происходит в первый подходящий коннект, например, если условный коннект **A1** имеет настройку syn_params="**dst:int+src:unk**" (разрешено отправлять на федеральные номера, подписываясь короткими номерами), коннект **A2** – "**dst:unk+src:int**", а в задании на отправление dst=79000000000 (т.е. получатель – федеральный номер / int) и src=000123 (т.е. отправитель – короткий номер / unk), то сообщение будет отправлено в коннект **A1**.

Если "**type=1**", то короткие сообщения распределяются примерно поровну между всеми доступными smpp-коннектами. Например, если для sms_sender_1.exe сконфигурированы коннекты **A1** (syn_params="**dst:int+src:int**"), **A2** (syn_params="**dst:unk,int+src:alpha,int**"), **A3**

(**syn_params="dst:int,unk,alpha+src:int,unk,alpha"**), A4 (**syn_params="dst:unk+src:unk"**), а файл-задание содержит некоторое количество заданий на отправку sms, где номера-отправители и номера-получатели – это номера с федеральной нумерацией (int), то смс будут распределены по коннектам **A1-A3**.

Значение по умолчанию (если не конфигурировать рассыльщик с определённым **proc_id** в этом блоке конфигурации) - **0**.

Пример конфигурации.

```
[CONNECTS_SYNONYM_TYPE]
#type: 0 - отправка sms pri podache smpp_defin=any_connect proishodit v pervyj rabochij konnekt, sootvetstvyushij
#tipy dst+src (default value).
#Esli type=1, to nagruzka raspredelyaetsya primerno porovny na kazhdyj konnekt, sootvetstvyushij
#tipy dst(int/unk/alpha)+src(int/unk/alpha).
#Vazhno! Nastroyka dejstvitelna tolko dlya "dst(int/unk/alpha)+src(int/unk/alpha)".
###
#proc_id      #type
1              0
2              1
[CONNECTS_SYNONYM_TYPE]
```

4.21. [CONNECTS_SYNONYM_SRC]

Специальный тип балансировки / резервирования трафика по целевым коннектам (в рамках определённого рассыльщика), основанный на номере отправителя (или маске номера отправителя).

Шаблон конфигурации –

"proc_id src_num/src_mask list_of_smpp_defin choice_smpp_defin_type".

proc_id – идентификатор экземпляра рассыльщика (например, для sms_sender_1.exe **proc_id=1**).

src_num/src_mask – параметр / критерий, определяющий целевой номер отправителя (и / или маску номера отправителя), исходя из которого происходит балансировка / резервирование sms-трафика. Допускается перечисление (через запятую и без пробелов) как масок номеров-отправителей (**src_mask**), так единичных src-номеров (**src_num**), например, **"0011,794*****,NOMER,NUM**,000*23,000123"**, где **"*"** – любой символ.

list_of_smpp_defin – список smpp-соединений (**smpp_defin**), через которые будут отправлены SMS, в которых номер отправителя подпадает под критерий **"src_num/src_mask"**.

choice_smpp_defin_type – параметр, определяющий тип балансировки / резервирования. Если равен **"0"**, то трафик уходит в первый рабочий коннект из списка **"list_of_smpp_defin"**, если же равен **"1"**, то трафик делится (примерно поровну) между всеми рабочими коннектами из списка.

Условия, при которых настройка активна:

1) наличие конфигурационной строки для определённого рассыльщика с **proc_id** (например, sms_sender_1.exe) в блоке **[CONNECTS_SYNONYM_SRC]**;

2) присутствие подстроки "**any_connect_src**" в имени файла-задания или в первом параметре строки (или строк) файла-задания. Также возможно указание в имени файла-задания подстроки вида "**any_connect_src_NUM**", где **NUM** – конкретный номер получателя (например, "**any_connect_src_000012**").

Краткая справка по файлам-заданиям в части выбора целевого smpp-коннекта

Указание на использование целевого smpp-соединения (**smpp_defin**) или одного из соединений при обработке файлов-заданий возможно посредством двух механизмов – через имя файла-задания и посредством первого параметра строки-команды (строки, содержащей параметры отправки конкретного SMS, разделённые посредством ";+").

1. Целевой **smpp_defin** через имя файла-задания (на примере **proc_id=1**, т.е. **sms_sender_1.exe**):

а) **общий шаблон** – "**some_str.subm1**", где **some_str** – любая последовательность символов (за исключением специальных слов). В данном случае используется настройка выбора smpp-соединения посредством первого параметра строки-команды;

б) **src-шаблон** – "**some_str-any_connect_src.subm1**", где **some_str** – любая последовательность символов (за исключением специальных слов), а **any_connect_src** – указание на то, что для **proc_id=1** существует настройка в блоке **[CONNECTS_SYNONYM_SRC]**;

в) **src-num-шаблон** – "**some_str-any_connect_src_NUM.subm1**", где **any_connect_src_NUM** указывает на наличие конфигурации для **proc_id=1** в блоке **[CONNECTS_SYNONYM_SRC]** и обозначает, что в списке рассылки (файле-задании) исключительно команды на отправки SMS с номером отправителя **NUM**;

г) **smpp_defin-шаблон** – "**some_str-ip:port:login.subm1**", где "**ip:port:login**" – указание использовать при отправке SMS конкретный SMS-центр, например, "**192.168.168.168:7788:login1.subm1**";

д) шаблоны из пунктов "б" и "в" верны и для блока конфигурации **[CONNECTS_SYNONYM_DST]**, но с поправкой на dst-адрес (адрес получателя): "**some_str-any_connect_dst.subm1**" (б) и "**some_str-any_connect_dst_NUM.subm1**" (в).

2. Целевой **smpp_defin** через первый параметр строки-команды (в файле-задании):

а) **any_connect** – выбор коннекта через механизм **syn_params** (соответствует одноимённому параметру в блоке **[CONNECTS]**);

б) **any_connect_src** – выбор целевого коннекта на основе настроек из блока **[CONNECTS_SYNONYM_SRC]**;

в) **any_connect_dst** – выбор целевого коннекта на основе настроек из блока **[CONNECTS_SYNONYM_DST]** (аналог конфигурации **CONNECTS_SYNONYM_SRC**, но параметром балансировки / резервирования выступает номер получателя);

г) **ip:port:login** – присвоение первому параметру конкретного smpp-коннекта (существующего в блоке [CONNECTS]).

Пример конфигурации.

```
[CONNECTS_SYNONYM_SRC]
#Specialnyj tip balansirovki/rezervirovaniya trafika, gde kriteriem vystupaet nomer otpravatelya.
#Dlya ispolzovaniya etoj opcii nuzhno, chtoby v imeni faila-ocheredi prisutstvovala libo stroka "any_connect_src",
#libo "any_connect_src_NUM" (s ukazaniem konkretnogo nomera SRC=NUM),
#naprimer, "some_str-any_connect_src.submN"/"some_str-any_connect_src_0001.submN", gde N - nomer sendera/potoka.
#Dругой вариант - v faile ocheredi vida "any_connect" (obshaya ochered, gde v imeni faila net ukazaniya na konkretnyi
#vid balansirovki),
#gde liniya = komanda na otpravku sms (stroka-komanda), yavno ukazat pervym parametrom (vmesto smpp_defin) znachenie "any_connect_src"
###
#src_num/src_mask: esli ispolzuetsya maska (src_mask), to format src_mask imeet vid (primer): 12*23, gde * - lyuboj simvol/cifra,
#t.e. pod masku popadut nomera 12(0)23, 12123, 12223, 12323, 12423, 12523 (i tak dalee).
#Nomera i maski mozno perechislyat cherez zapyatuyu v razlichnykh variaciyah, naprimer, "12345,333*,678*" (bez kavyчек).
###
#list of smpp_defin: "ip1:port:login,ip2:port:login,ip3:port:login" (bez kavyчек), t.e. perechislenie akkauntov, cherez kotorye
#mozno otravlyat sms. Vybora akkaunta osushchestvlyetsya ishodya iz znacheniya parametra #choice_smpp_defin_type.
###
#choice_smpp_defin_type: 0 - otpravka sms v pervyj po spisku rabochij konnekt, 1 - nagruzka raspredelyaetsya primerno porovny
#na kazhdyj konnekt
#proc_id      #src_num/src_mask      #list of smpp_defin      #choice_smpp_defin_type
1              0033              192.168.99.5:5566:login1,192.168.98.5:5566:login2      1
1              0034              192.168.99.7:5566:login12,192.168.99.6:5567:login22      1
1              00077*,0055          192.168.99.7:5466:login62,192.168.99.6:4567:login28      0
[CONNECTS_SYNONYM_SRC]
```

4.22. [CONNECTS_SYNONYM_DST]

Блок конфигурации, аналогичный [CONNECTS_SYNONYM_SRC], но критерием балансировки / резервирования выступает номер получателя (dst).

Пример конфигурации.

```
[CONNECTS_SYNONYM_DST]
#Dannye nastroyki analogichny [CONNECTS_SYNONYM_SRC], no celevym parametrom vystupaet nomer otpravatelya (dst)
###
#Dlya ispolzovaniya etoj opcii nuzhno, chtoby v imeni faila-ocheredi prisutstvovala libo stroka "any_connect_dst",
#libo "any_connect_dst_NUM" (s ukazaniem konkretnogo nomera DST=NUM),
#naprimer, "some_str-any_connect_dst.submN"/"some_str-any_connect_dst_0001.submN", gde N - nomer sendera/potoka.
#Dругой вариант - v faile ocheredi vida "any_connect" (obshaya ochered, gde v imeni faila net ukazaniya na konkretnyi vid balansirovki),
#gde liniya = komanda na otpravku sms, yavno ukazat pervym parametrom (vmesto smpp_defin) znachenie "any_connect_dst".
###
#proc_id      #dst num/dst mask      #list of smpp_defin      #choice_smpp_defin_type
1              9955,440099,775*      10.200.4.5:6789:login1,11.201.5.6:7000:login2      1
[CONNECTS_SYNONYM_DST]
```

4.23. [RX_COMMON_TIMEOUTS]

Конфигурация различных таймаутов, связанных с общей логикой ресивера smpp-трафика (sms_sender_Nrx.exe).

Общий шаблон - "proc_id enq_link_c_t recr_smppiosel_err com_smpp_sta_err".

proc_id – идентификатор экземпляра ресивера (например, для sms_sender_0rx.exe proc_id=0rx).

enq_link_c_t – таймаут (в секундах) проверки существующих smpp-коннектов на необходимость отправки **enquire_link**-а (keep-alive-пакета).

recr_smppiosel_err – таймаут (в секундах) пересоздания селектора коннектов, если произошла ошибка.

com_smpp_sta_err – таймаут (в секундах) проверки smpp-коннектов на необходимость переподключения после получения ошибки.

Значения по умолчанию: **enq_link_c_t** = 1, **recre_smppiosel_err** = 3, **com_smpp_sta_err** = 2.

Пример конфигурации.

```
[RX_COMMON_TIMEOUTS]
#raznye taimauty, svyazannye s obshej logikoj servisa sms_sender_Nrx
#enq_link_c_t - v sekundah. Taimaut prohozhdeniya po spisku konnektov na predmet neobhodimosti otpravki enq_link
#recre_smppiosel_err - recreate_smpp_io_select_if_error. v sekundah. Taimaut peresozdaniya select-var, v kotoruyu pomeshayutsya smpp_h
#com_smpp_sta_err - v sekundah. Taimaut proverki connects na neobhodimost smpp-rekonnekta posle oshibki
#proc_id      #enq_link_c_t  #recre_smppiosel_err  #com_smpp_sta_err
0rx           2            5                               4
[RX_COMMON_TIMEOUTS]
```

4.24. [DSM_COMMON_TIMEOUTS]

Конфигурация различных таймаутов, связанных с общей логикой обработчика отчётов (sms_sender_Ndsm.exe).

Общий шаблон – "**proc_id** **wait_resp_res_t** **fin_save_res_t**".

proc_id – идентификатор экземпляра обработчика отчётов о доставке (например, для sms_sender_0dsm.exe **proc_id**=0dsm).

wait_resp_res_t – таймаут (в секундах) ожидания всех информационных пакетов (submit_sm_resp-ов и отчётов о доставке = deliver_sm-ов), содержащих информацию об отправке и доставке всех частей короткого сообщения (SMS).

fin_save_res_t – таймаут (в секундах) сохранения результатов обработанных заданий (заданий на отслеживание статуса доставки). Означает, что раз в **fin_save_res_t** секунд будет происходить запись результатов обработки в виде файла в директорию **dsm_result_dir** (параметр из блока [MAIN]). Данный алгоритм можно изменить посредством параметра **dsm_result_write_by_hour** (блок конфигурации [MAIN]). Если **dsm_result_write_by_hour**=1, то результаты обработки группируются с почасовой разбивкой.

Значения по умолчанию: **wait_resp_res_t** = 30, **fin_save_res_t** = 60.

Пример конфигурации.

```
[DSM_COMMON_TIMEOUTS]
#raznye taimauty, svyazannye s obshej logikoj servisa sms_sender_Ndsm
###
#wait_resp_res_t (v sekundah) - taimaut ozhidaniya vseh inf-paketov ot N/Nrx, sodержashih informaciyu ob otpravke msg v smpp-konnekt
###
#fin_save_res_t (v sekundah) - taimaut sohraneniya rezultatov obrabotannyh zadaniy (*_track/*_notif/etc)
###
#proc_id      #wait_resp_res_t      #fin_save_res_t
0dsm          35                    68
[DSM_COMMON_TIMEOUTS]
```

4.25. [PROCESSING_DELIVER_SM_PARAMS]

Блок конфигурации определяет логику отправки запросов о статусе доставки сообщений (**query_sm**) для конкретного smpp-коннекта (**smpp_defin**).

Общий шаблон - "**smpp_defin pass_wait_t send_query_sm_t query_sm_send_cnt**".

smpp_defin = smsc_ip:smc_port:smc_login (**host:port:login** из блока [CONNECTS]).

pass_wait_t – время (в минутах) пассивного ожидания отчёта (или отчётов, если отслеживаемое сообщение многосоставное) о доставке (**deliver_sm**, где в содержании сообщения присутствует идентификатор отправленного сообщения).

send_query_sm_t – периодичность (в минутах) отправки запроса статуса доставки сообщения (**query_sm**), если истекло время пассивного ожидания отчёта о доставке.

query_sm_send_cnt – максимальное количество **query_sm**, отправляемых (с периодичностью в **send_query_sm_t** минут) после истечения времени пассивного ожидания отчёта. Если выставить равным "0", то запросы **query_sm** будут отсылаться всё время жизни **SMS** (время жизни сообщения – параметр, который задаётся в строке-команде на отправку **SMS**).

Параметры по умолчанию (если **smpp_defin** отсутствует в данном блоке конфигурации):
pass_wait_t = 60, **send_query_sm_t** = 60, **query_sm_send_cnt** = 0.

Пример конфигурации.

```
[PROCESSING_DELIVER_SM_PARAMS]
#pass wait_t - v minutah. Vremya passivnogo ozhidaniya otheta o dostavke (deliver_sm s sootv. message_id)
#send_query_sm_t - v minutah. Periodichnost otpravki query_sm, kogda isteklo vremya #pass wait_t
#query_sm_send_cnt: esli 0, to query_sm otpravlyautsya vse vremya zhizni submit_sm (validity_period-param)
#smpp_defin      #pass_wait_t      #send_query_sm_t      #query_sm_send_cnt
192.168.77.88:9900:login0      30      10      5
[PROCESSING_DELIVER_SM_PARAMS]
```

4.26. [DSM_NOTIF_FUNC_TEXT]

Конфигурация, определяющая тексты сообщений (сообщений о доставке), отсылаемых на номер отправителя изначального сообщения при доставке / недоставке / частичной доставке получателю.

Общий шаблон конфигурации текста уведомления – "**t_notif_text_some_name=some_text**".

t_notif_text_some_name – имя шаблона ответа. Должно всегда начинаться с "**t_**". Например, имя шаблона ответа может выглядеть так – "**t_for_smsc1**".

Шаблон конфигурации текста уведомления в зависимости от *proc_id* / *smpp_defin* –
"**proc_id/smpp_defin** **t_notif_text_some_name**=some_text".

proc_id – идентификатор экземпляра обработчика отчётов о доставке (например, для sms_sender_0dsm.exe proc_id=0dsm).

smpp_defin = smsc_ip:smc_port:smc_login (**host:port:login** из блока [CONNECTS]).

Особенности:

1) чтобы задействовать механизм отправки сообщений о доставке, необходимо в строке-команде прописать параметр **r_deliv0** (7-й по счёту). Возможные значения **r_deliv0**: **act_track** (активное отслеживание доставки, в т.ч. и посредством query_sm), **psv_track** (пассивное отслеживание доставки), **act_notif** (активное отслеживание статуса доставки и уведомление отправителя о результате), **psv_notif** (пассивное отслеживание доставки и уведомление отправителя), **smpl_track** (простое отслеживание без ожидания deliver_sm и использования query_sm). Блок конфигурации DSM_NOTIF_FUNC_TEXT действует только для **r_deliv0**, равных **act_notif** или **psv_notif**;

2) задания на отправку формируются на стороне сервиса-обработчика отчётов **sms_sender_Ndsm.exe** и передаются сервису-рассыльщику **sms_sender_N.exe**;

3) сообщения, формируемые на стороне сервиса-обработчика отчётов (**dsmSMS**), являются зеркальными по отношению к изначальному SMS в плане src (номер отправителя) и dst (номер получателя). Например, если изначальное сообщение отправляется от 0001 (src) на номер 0002 (dst), то **dsmSMS** будет отправлено от 0002 (src) на номер 0001 (dst);

4) уведомления (или **dsmSMS**) отправляются в трёх случаях: а) при успешной доставке; б) при частичной доставке (если не все части многосоставного смс были успешно доставлены); в) при недоставке сообщения;

5) за каждым событием закреплено стандартное имя шаблона ответа: а) **t_comm_dlv_txt** (если оригинальное сообщение было успешно доставлено); б) **t_comm_prt_dlv_txt** (частичная доставка); в) **t_comm_no_dlv_txt** (сообщение недоставлено);

6) шаблоны сообщений, имеющие более детализированный характер, подставляются в текст уведомления в первую очередь, что отражено в таблице ниже.

Тип шаблона	Описание	Приоритет
DSM_NOTIF		
r_deliv0-direct	Тексты (или шаблоны) уведомлений задаются в строке-команде изначального сообщения посредством добавления подстрок ":dlv_txt=SOME_TEXT=dlv_txt" (при успешной доставке), ":prt_dlv_txt=SOME_TEXT=prt_dlv_txt" (при частичной доставке)	1 (высший)

Тип шаблона	Описание	Приоритет
DSM_NOTIF	<p>и "no_dlv_txt=SOME_TEXT=no_dlv_txt" (при доставке) в параметр r_deliv0 (для значений act_notif / psv_notif), например, act_notif:dlv_txt=SOME_TEXT=dlv_txt. Вместо текста (SOME_TEXT) возможно вписать имя конкретного шаблона ("t_some_tmplt"), определённого в блоке [DSM_NOTIF_FUNC_TEXT].</p> <p>Если для конкретного случая (доставка / частичная доставка / недоставка) подстрока не задана, то используется стандартный шаблон (t_comm_dlv_txt / t_comm_prt_dlv_txt / t_comm_no_dlv_txt), если он задан (для общего случая / для конкретного proc_id / конкретного smpp_defin).</p> <p>При отсутствии сконфигурированных стандартных шаблонов используются текстовки DELIVRD (если SMS доставлено), PRTDELIV (доставлено частично), UNDELIV (недоставлено).</p>	
smpp_defin-tmplt	<p>Заданный в блоке [DSM_NOTIF_FUNC_TEXT] шаблон ответа с привязкой к конкретному smpp-коннекту. Может включать в себя как имена шаблонов, используемых по умолчанию (t_comm_dlv_txt / t_comm_prt_dlv_txt / t_comm_no_dlv_txt), так и специально определяемые, вызов которых возможен через r_deliv0-direct.</p>	2
proc_id-tmplt	<p>Определённый в блоке [DSM_NOTIF_FUNC_TEXT] шаблон ответа с привязкой к конкретному обработчику отчётов с заданным proc_id (например, sms_sender_1dsm.exe, где proc_id=1dsm). Может включать в себя как имена шаблонов, используемых по умолчанию (t_comm_dlv_txt / t_comm_prt_dlv_txt / t_comm_no_dlv_txt), так и специально определяемые, вызов которых возможен через r_deliv0-direct.</p>	3
Общий шаблон / common-tmplt	<p>Определённый в блоке [DSM_NOTIF_FUNC_TEXT] шаблон ответа. Может включать в себя как имена шаблонов, используемых по умолчанию (t_comm_dlv_txt / t_comm_prt_dlv_txt / t_comm_no_dlv_txt), так и специально определяемые, вызов которых возможен через r_deliv0-direct.</p>	4

Пример конфигурации.

```
#Vazhno! Imena shablonov vseгда dolzhny nachinatsya s "t", naprimer, "t_some_tmplt_name"!
##obshie textovye shablony. Podstroka SRC zamenyaetsya na nomer otpravitelya.
#Podstroka DST zamenyaetsya na nomer poluchatelya.
t_comm_dlv_txt=Uvazhaemyj abonent DST, Vash zapros dostavljen abonentu SRC
t_comm_no_dlv_txt=Uvazhaemyj abonent DST, Vash zapros ne byl dostavljen abonentu SRC
t_comm_prt_dlv_txt=Uvazhaemyj abonent DST, Vash zapros byl chastichno dostavljen abonentu SRC
##obshie textovye shablony
##v zavisimosti ot smpp_defin/proc_id
#smpp_defin/dsm-proc_id #txt_tmplt_name=tmplt text
#Pri etom vozmozhno dlya konkretnogo smpp_defin/proc_id pereopredelit kak standartnye shablony
#(t_comm_dlv_txt/t_comm_no_dlv_txt/t_comm_prt_dlv_txt), tak i propisat svoi. Esli dlya kakogo libo smpp_defin/proc_id pereopredeleny
#stand. shablony, to oni ispolzuyutsya v pervuyu ochered, kogda "dlv_txt/no_dlv_txt/prt_dlv_txt" ne zadany v r_deliv0 (act/psv_notif).
#Shablony, zadannye na bolee "nizkom urovne" (urovni takie "obshie shablony->dlya dsm-proc_id->dlya smpp_defin), bolee prioritety,
#chem shablony bolee vysokogo urovnya. Naprimer, shablon, zadannyj dlya smpp_defin (pri imeyushih obshih i zadannyh dlya proc_id)
#budet ispolzovan imenno on.
#192.168.0.3:3333:some_login t_some_tmplt_name_for_smpp_defin=some text
#0dsm t_some_tmplt_name_for_proc_id=some text
###
192.168.55.66:8889:login1 t_comm_dlv_txt=Доставлено!
192.168.55.66:8889:login1 t_comm_no_dlv_txt=Недоставлено!
192.168.55.66:8889:login1 t_comm_prt_dlv_txt=Не все части СМС доставлены!
192.168.55.66:8889:login1 t_alter_dlv=0-1-2-3
###
0dsm t_comm_dlv_txt=Доставлено. 0dsm!
0dsm t_comm_no_dlv_txt=Недоставлено. 0dsm!
0dsm t_comm_prt_dlv_txt=Не все части СМС доставлены. 0dsm!
0dsm t_alter_dlv=0-1-2-3. 0dsm
##v zavisimosti ot smpp_defin/proc_id
[DSM NOTIF FUNC TEXT]
```

4.27. [DSM_NOTIF_REPORT_LISTS]

По умолчанию отправка уведомления о доставке (когда **r_deliv0** равен **act_notif** / **psv_notif**) управляется номеру-отправителю (SRC-адрес) изначального сообщения, но стандартный сценарий отправки возможно изменить посредством добавления подстроки **":report_dst=SOME_DST=report_dst"** в параметр **r_deliv0**, где значение SOME_DST – это либо номер нового получателя, отличного от изначального SRC-адреса, либо список номеров-получателей, разделённых запятой (без пробелов), либо имя списка рассылки, сконфигурированного в блоке **[DSM_NOTIF_REPORT_LIST]**.

Пример строки-команды, когда **report_dst** прописан в блоке конфигурации –

```
"192.168.7.8:8899:login1;+79450000000;+79450000001;+0;+Privet! Как дела?;+12h;
+act_notif:report_list=l_spisok01=report_dst;+0;+0".
```

По аналогии с **[DSM_NOTIF_FUNC_TEXT]** в данном блоке конфигурации можно задать списки для smpp_defin, proc_id (например, 0dsm) и общего случая, при этом приоритет при выборе из списков с одинаковыми именами, но сконфигурированными для разных уровней (общий / уровень proc_id / уровень smpp_defin), отдаётся более детальному уровню (т.е. уровню smpp_defin). Например, если сконфигурировать список с именем l_list01 для smpp_defin=192.168.7.8:8899:login1 и для proc_id=0dsm, то в результате обработки строки-команды (выше) для рассылки уведомлений будет выбран список, заданный для уровня smpp_defin.

Важный момент – все имена списков должны начинаться с "l_" (малая L), например, "l_list000".

Пример конфигурации.

```
[DSM NOTIF REPORT LISTS]
#Vazhno! Imena spiskov vseгда dolzhny nachinatsya s "l ", naprimer, "l_some list name"!
#Priority: obshie_spiski->dlya dsm-proc_id->dlya smpp_defin. Bolee vysokij prioritet imeyut spiski, zadannye dlya konkretnogo
#smpp_defin.
##obshie spiski
#l list_name=num1,num2,etc
##obshie spiski
##v zavisimosti ot smpp_defin/dsm-proc_id
#smpp_defin/dsm-proc_id #list_name=num1,num2,etc
#192.168.0.3:3333:some_login l_some_list_name_for_smpp_defin=some_list
#0dsm l_some_list_name_for_proc_id=some_list
##v zavisimosti ot smpp_defin/dsm-proc_id
l_common_list0=0011,79260000000,1200
192.168.4.5:9999:login1 l_list0=8899,5555,0001
0dsm l_list01=0004,0005,79260000001
[DSM NOTIF REPORT LISTS]
```

4.28 [SMPP_RESP_CODES]

Блок конфигурации, содержащий resp-коды и пояснения к ним (фигурируют в логах рассыльщика). Возможно переопределить существующие пояснения или добавить свои, если, например, используется SMS-центр с некоторым количеством нестандартных resp-кодов, не входящих в спецификацию smpp 3.4.

4.29. [LATIN_CHANGE]

Здесь возможно переопределить символы транслитерации (rus → lat).

5. Отправка SMS. Общие сведения

Отправка SMS-сообщения (или множества SMS-сообщений) подразумевает выполнение ряда простых действий:

- 1) создать файл-задание ("touch send_sms_file"). Для общего случая имя может быть любым;
- 2) заполнить файл-задание командами на отправку SMS, где строка (строка-команда) является командой на отправку одного короткого сообщения (SMS);
- 3) переименовать созданный файл (п.1), добавив расширение ".submN", где N - идентификатор рассыльщика (например, "send_sms_file.subm1");
- 4) переместить файл с расширением ".submN" (например, "send_sms_file.subm1") в директорию **submit_sms_dir** (директория-очередь для файлов-заданий, задаётся в блоке [MAIN]).

5.1. Отправка SMS. Формат строки-команды

В общем виде формат команды на отправку SMS (строки-команды) представляет собой строку файла (файла-задания), где параметры разделены составным символом "+".

smpp_defin;+dst;+src;+is_lat;+sm_full;+valid_time;+r_deliv0;+wfr_t;+brr_cnt

1) smpp_defin – составной идентификатор коннекта. smsc_ip:smc_port:smc_login (host:port:login из блока [CONNECTS]). Также возможны следующие варианты: **any_connect**, **any_connect_src**, **any_connect_dst** и **null**. Использование “any_connect” тесно связано с параметром “syn_params” (блок [CONNECTS]) и блоком конфигурации [CONNECTS_SYNONYM_TYPE], а использование “any_connect_src / any_connect_dst” – с блоками конфигурации [CONNECTS_SYNONYM_SRC] / [CONNECTS_SYNONYM_DST]. Назначение any_connect, any_connect_src и any_connect_dst - балансировка нагрузки и обеспечение резервирования. См. разделы: **SMPP_CONNECTS**, **CONNECTS_SYNONYM_TYPE**, **CONNECTS_SYNONYM_SRC**, **CONNECTS_SYNONYM_DST**.

2) dst – номер получателя. Часто используемые форматы – international (**int**) и unknown (**unk**). Int – формат номера в международном формате (например, 79000000000, где "7" – код РФ), а unk – короткие номера (например, 000111). Вероятно, есть возможность использовать и альфа-нумерацию (alphanumeric / alpha), но это зависит от возможностей / настроек SMS-центра. Со стороны sms_sender-а нет ограничений на использование alpha-номера в качестве номера получателя (dst). Возможность отправки SMS на тот или иной номер (в общем случае) всегда зависит от разрешений, установленных со стороны SMS-центра.

3) src – номер отправителя (или подпись отправителя). Возможные форматы - int / unk / alpha. Возможность отправки SMS с тем или иным src-номером зависит от разрешений на SMS-центре.

4) is_lat - параметр, отвечающий за включение / отключение механизма транслитерации (в соответствии с блоком конфигурации [LATIN_CHANGE]) для конкретного SMS-сообщения. Если “1”, то транслитерация включена.

5) sm_full - текст короткого сообщения. Если требуется, чтобы в коротком сообщении получатель (dst-номер) видел переносы строк, в текст возможно добавить специальное сочетание символов “-nl-” (впоследствии заменяется на символ переноса строки).

6) valid_time - время хранения (в часах) сообщения на SMS-центре в случае невозможности доставки на данный момент (например, абонент-получатель находится вне зоны действия сотовой сети). Возможные значения: def - 10 минут, диапазон целых чисел от 1 до 24 (или от 1h до 24h).

7) r_deliv0 (в общих чертах) - параметр, указывающий на сценарий постобработки данных (статус отправки, статус доставки), полученных в результате отправки короткого сообщения. Возможные значения: **a) act_track**. Включает в себя обработку статуса отправки и активное отслеживание статуса доставки посредством query_sm, если отчёт о доставке не поступил за

время `pass_wait_t` (определяется в блоке [PROCESSING_DELIVER_SM_PARAMS]); **б) *psv_track***. Аналогичен `act_track`, но без использования `query_sm`; **в) *act_notif***. Аналогичен `act_track`, но по факту доставки / недоставки / частичной доставки отправляется sms-уведомление на src-номер (номер получателя уведомления о статусе доставки возможно переопределить) изначального сообщения; **г) *psv_notif***. Аналогичен `act_notif`, но без использования `query_sm`; **д) *smpl_track***. Простое отслеживание статуса отправки.

8) *wfr_t* – время ожидания (в секундах) `resp-a` (`submit_sm_resp`) на отправленный `submit_sm`. Если выставить равным 0, то будет использовано значение по умолчанию, равное 5-ти секундам.

9) *brr_cnt* – количество повторных попыток отправки `submit_sm`, если в `submit_sm_resp` был получен статус, не равный “0” (т.е. отправка по какой-то причине не была успешной). Если параметр не указывать, то по умолчанию кол-во повторных попыток отправки равно “0”.

5.1.1. Описание возможных опций параметра `r_deliv0`

Дополнительные опции (*notif-модификаторы*) в параметре `r_deliv0` (7-й параметр строки-команды) доступны для значений ***act_notif*** (активное отслеживание статуса доставки + уведомление для отправителя) и ***psv_notif*** (пассивное отслеживание статуса доставки + уведомление для отправителя).

В общем виде модификатор выглядит так – “**:MODx=param=MODx**”, где MODx - условное имя модификатора, param – параметр (или параметры) модификатора. В случае использования *notif-модификаторов* `r_deliv0` принимает вид (на примере ***act_notif***) – “**act_notif:MODx0=param0=MODx0:MODx1=param1=MODx1**” (возможно одновременное использование множества опций).

Список *notif-модификаторов*:

1) “**:dlv_txt=some_txt-or-dsm_notif_tmplt=dlv_txt**”. Модификатор текста смс-уведомления (отправка на src-номер изначального сообщения) для случая, когда изначальное сообщение было успешно доставлено адресату. Можно задать как напрямую текст, так и указать имя заранее сконфигурированного шаблона (из блока [DSM_NOTIF_FUNC_TEXT]).

2) “**:no_dlv_txt=some_txt-or-dsm_notif_tmplt=no_dlv_txt**”. Модификатор текста смс-уведомления для случая, когда изначальное сообщение не было доставлено адресату (dst-адрес). Можно задать как напрямую текст, так и указать имя заранее сконфигурированного шаблона (из блока [DSM_NOTIF_FUNC_TEXT]).

3) “**:prt_dlv_txt=some_txt-or-dsm_notif_tmplt=prt_dlv_txt**”. Модификатор текста смс-уведомления для случая, когда изначальное сообщение было частично доставлено абоненту (в случае отправки многосоставного сообщения). Можно задать как напрямую текст, так и указать имя заранее сконфигурированного шаблона (из блока [DSM_NOTIF_FUNC_TEXT]).

4) “**:first_ok**”. Если данный модификатор присутствует в рамках параметра `r_deliv0`, то статус успешной доставки будет определяться по факту доставки 1-й части смс-сообщения.

5) **":report_dst=some_dst / list_of_dst / dsm_notif_report_list=report_dst"**. Позволяет переопределить получателя уведомления о доставке изначального сообщения. Можно как просто переопределить номер, так и указать несколько получателей через запятую (или вписать заранее сконфигурированный список из блока конфигурации **[DSM_NOTIF_REPORT_LISTS]**).

6) **":report_src=some_src=report_src"**. Позволяет переопределить отправителя (номер, от имени которого будет отправлено SMS) уведомления о доставке изначального сообщения.

7) **":sندر=sender_id+smpp_defin=sender"**. По умолчанию отправка уведомления о доставке происходит через тот же smpp-коннект (и sms_sender), через который было отправлено изначальное сообщение. Данный модификатор позволяет переопределить sms_sender и smpp-коннект (smpp_defin) при отправке уведомления о доставке.

8) **":report_spy=smpl_track / act_track / psv_track=report_spy"**. По умолчанию статус доставки / отправки уведомления о доставке изначального сообщения не отслеживается. Использование этого модификатора позволяет переопределить алгоритм по умолчанию. Если вписать значение модификатора, равное smpl_track (простое отслеживание статуса отправки), act_track (отслеживание как статуса отправки, так и активный трэкинг статуса отправки) или psv_track, то соответствующий sms_sender_Ndsm сохранит результаты отправки / доставки смс-уведомления о статусе изначального сообщения.

5.2. Отправка SMS. Формат имени файла-задания

Возможные варианты функциональных модификаторов имени файла-задания (на примере файла-задания, предназначенного для обработки сервисом "sms_sender_1.exe"):

1) без модификаторов, т.е. имя файла задания имеет вид "some_file_name.subm1";

2) модификатор **"send-after"**. Позволяет назначить время начала отправки коротких сообщений (в рамках файла-задания). Формат – "YYYYMMDDHHMISSafter-", где "YYYY" – год, "MM" – месяц, "DD" – день, "HH" – час, "MI" – минута, "SS" – секунда. Должен располагаться в начале имени файла-задания. Пример – "20210424133015after-some_file_name.subm1".

3) модификатор **"smpp-connect"**. Даёт возможность назначить конкретный smpp-коннект (из блока **[CONNECTS]**) при осуществлении отправки SMS в рамках конкретного файла-задания (первый параметр строки-команды при использовании данного модификатора игнорируется). Должен размещаться перед расширением файла-задания, например, "some_file_name-192.168.78.88:9977:login1.subm1" ("192.168.78.88" – ip-адрес SMS-центра, 9977 – порт, login1 – учетная запись для коннекта по протоколу smpp к SMS-центру).

Также (в рамках модификатора "smpp-connect") возможны варианты с балансировкой / резервированием:

a) **"-any_connect_src"**. Наличие этой подстроки в имени файла-задания (перед расширением, например, "some_file_name-any_connect_src.subm1") означает, что строки-команды будут

обработаны в соответствии с правилами балансировки / резервирования, заданными в блоке конфигурации [CONNECTS_SYNONYM_SRC].

b) "-any_connect_dst". Аналогично "-any_connect_src", но применяются правила из блока [CONNECTS_SYNONYM_DST].

c) "-any_connect_src_NUM". Аналогично "-any_connect_src", но применяется правило из блока [CONNECTS_SYNONYM_SRC], соответствующее конкретному номеру отправителя NUM.

d) "-any_connect_dst_NUM". Аналогично "-any_connect_src_NUM", но применяется правило из блока [CONNECTS_SYNONYM_DST].

Допускается совместное использование модификаторов "send-after" и "smpp-connect", например, "20210424133015aftr-some_file_name-192.168.78.88:9977:login1.subm1" или "20210424133015aftr-some_file_name-any_connect_src.subm1".

6. Результат обработки отчётов о доставке

Результат обработки отчетов об отправке и доставке представляет собой файлы (результат-файлы) с расширением ".dsmres", формируемые обработчиком отчётов о доставке (sms_sender_Ndsm.exe) в директории **dsm_result_dir** (параметр из блока конфигурации [MAIN]), где каждая строка представляет собой набор данных о статусе обработки короткого сообщения (SMS) или отдельных его частей.

Типы результат-файлов:

1) файл общих сведений о результате отправки и доставки (общий результат-файл). Содержит совокупные сведения о статусе обработки сообщений без учёта специальных данных по каждой части короткого сообщения (в т.ч. и для многосоставных SMS). Формат имени файла – "yyyymmddhhmiss_dsm-id_comm.dsmres", где yyyymmddhhmiss (yyyy – год, mm – месяц, dd – день, hh – номер часа от 00 до 23, mi – минута, ss – секунда) – время создания файла, а **dsm_id** – идентификатор обработчика отчётов, создавшего общий результат-файл. Например, общий результат-файл от sms_sender_1dsm.exe – это 20210607120000-1_comm.dsmres;

2) файл специальных сведений о результате отправки и доставки (специальный результат-файл). Содержит сведения о статусе обработки каждой части SMS. Формат имени файла – "yyyymmddhhmiss_dsm-id_part.dsmres", где yyyymmddhhmiss – время создания файла, а **dsm_id** – идентификатор обработчика отчётов, создавшего общий результат-файл. Например, специальный результат-файл от sms_sender_1dsm.exe – это 20210607120000-1_part.dsmres.

6.1. Результат обработки отчётов о доставке. Общий результат-файл

Формат файла - "yyyymmddhhmiss_dsm-id_comm.dsmres".

Формат строки в файле - "**begin_dt_op**;**end_time**;**ip**;**port**;**login**;**dst**;**src**;**dt_id**;
n_stream_id;**dsm_id**;**rd_type**;**ton_npi**=***ton_npi**;**cnt**=***cnt**;**sts**=***sts**;**tms**=***tms**;
ntf=***ntf**;**+**".

begin_dt_op – время (в формате yyyymmddhhmiss) попадания SMS в программную очередь рассыльщика (sms_sender_N.exe).

end_time – время окончания формирования отчёта о доставке (в формате yyyymmddhhmiss).

ip, **port** и **login** – ip-адрес, порт и логин для подключения к SMS-центру (блок [CONNECTS]).

dst – номер получателя.

src – номер отправителя.

dt_id – уникальный идентификатор сообщения (односоставного или многосоставного), основанный на времени от начала эпохи.

n_stream_id – идентификатор рассыльщика, через который было отправлено сообщение (например, n_stream_id=1, если SMS было отправлено через sms_sender_1.exe).

dsm_id – идентификатор сервиса обработки отчётов (например, dsm_id=1 для sms_sender_1dsm.exe), сформировавшего отчёт.

rd_type – тип обработки. Возможные значения: **smpl_track** (простое отслеживание статусов отправки каждой части SMS), **act_track** (активное отслеживание статусов отправки и доставки), **psv_track** (пассивное отслеживание статусов отправки и доставки), **act_notif** (активное отслеживание статусов отправки и доставки с последующим SMS-уведомлением на адрес отправителя), **psv_notif** (пассивное отслеживание статусов отправки и доставки с последующим SMS-уведомлением на адрес отправителя).

ton_npi=***ton_npi** – содержит информацию (*) о параметрах **ton** (параметр, сообщающий SMS-центру формат нумерации и тип сети) и **npi** (параметр, указывающий на тип нумерации) как для номера отправителя (src), так и для номера получателя (dst). Полный формат - "**ton_npi**=**dst_ton**|**dst_npi**|**src_ton**|**src_npi**=**ton_npi**" (параметры разделены посредством "|+").

dst_ton – ton для номера получателя.

dst_npi – npi для номера получателя.

src_ton – ton для номера отправителя.

src_npi – npi для номера отправителя.

cnt*=cnt – содержит информацию (*) о счётчиках, параметрах и значениях, использованных при отправке SMS. Полный формат - “**cnt=part_cnt|brr_cnt_b|inf_arr_cnt|sc_sndd_cnt|sc_dlv_cnt=cnt**” (параметры разделены посредством “|+”).

part_cnt – общее количество частей SMS.

brr_cnt_b – начальное количество попыток повторной отправки на каждую часть сообщения в случае, если статус отправки не равен "0" (статус извлекается из ответа от SMS-центра – submit_sm_resp).

inf_arr_cnt – количество полученных от рассыльщика (sms_sender_N.exe) информационных пакетов об отправке частей SMS.

sc_sndd_cnt – количество частей SMS, которые были успешно отправлены.

sc_dlv_cnt – количество успешно доставленных частей SMS.

sts*=sts – содержит информацию (*) о статусах обработки сообщения. Полный формат - “**sts=send_resp_st|deliv_st|is_exp_wfia|is_exp_pw|is_exp_rdt=sts**” (параметры разделены посредством “|+”).

send_resp_st – статус отправки сообщения. Возможные значения: 1 (все части SMS отправлены), 2 (ни одна из частей сообщения не была отправлена), 3 (не все части SMS были отправлены).

deliv_st – статус доставки сообщения. Возможные значения: 0 (статус доставки не отслеживался), 1 (все части SMS доставлены), 2 (ни одна из частей не доставлена), 3 (доставлены не все части сообщения).

is_exp_wfia – сигнализирует о превышении таймаута (wait_resp_res_t из блока [DSM_COMMON_TIMEOUTS]) ожидания всех информационных пакетов (со статусами отправки, извлечённых из ответов от SMS-центра) от сервиса sms_sender_N.exe (или sms_sender_Nrx.exe). Возможные значения: 0 (превышения таймаута не было), 1 (превышение таймаута было).

is_exp_pw – говорит о превышении таймаута пассивного ожидания отчёта о доставке (параметр pass_wait_t из блока [PROCESSING_DELIVER_SM_PARAMS]) от SMS-центра (deliver_sm). Возможные значения: 0 (превышения таймаута не было), 1 (превышение таймаута было).

is_exp_rdt – сообщает о превышении времени жизни сообщения на SMS-центре. Возможные значения: 0 (время жизни SMS не истекло), 1 (время жизни сообщения истекло).

tms*=tms – содержит информацию (*) о таймерах и периодах ожидания, связанных с обработкой короткого сообщения. Полный формат - “**tms=w_all_inf_t|pass_wait_t|reg_deliv_time=tms**” (параметры разделены посредством “|”).

w_all_inf_t – показывает время (в секундах), затраченное сервисом sms_sender_Ndsm.exe на ожидание всех информационных пакетов с информацией о статусах отправки всех частей сообщения.

pass_wait_t – содержит дату окончания пассивного ожидания отчётов о доставке (deliver_sm) сообщения в формате YYYYMMDDHHMISS.

reg_deliv_time – содержит дату устаревания сообщения на SMS-центре в формате YYYYMMDDHHMISS.

ntf*=ntf – содержит информацию (*) о параметрах SMS-уведомления о статусе доставки (адресом получателя выступает адрес отправителя оригинального сообщения). Если в дефолтовую логику не было внесено изменений посредством модификации параметра **r0_deliv**, если **r_deliv0** = “**act_notif / psv_notif**”, описанного в разделе “**Описание возможных опций параметра r_deliv0**”, то данный информационный блок будет иметь вид – “**ntf=1=ntf**” (аналогично и для случаев, когда **r_deliv0** не равен “**act_notif / psv_notif**”). Для случаев, когда в логику отправки SMS-уведомлений о статусе доставки были внесены изменения, формат ntf-параметра примет следующий вид –

“**ntf=0|ntf_first_ok|ntf_dlv_txt|ntf_no_dlv_txt|ntf_prt_dlv_txt|ntf_rep_src|ntf_rep_dst|ntf_sndr_stream|ntf_sndr_smpp|ntf_rep_spy=ntf**” (параметры разделены посредством “|”).

ntf_first_ok – параметр, демонстрирующий порядок отправки SMS-уведомления о статусе доставки. Если равен “1”, то по факту доставки первой части сообщения было отправлено SMS-уведомление, если же “0” – уведомление было отправлено, когда статусы доставки всех частей SMS стали известны.

ntf_dlv_txt – отображает использованный текст SMS-уведомления о статусе доставки, если изначальное сообщение было успешно доставлено. Если равен “0”, то использовался стандартный текст уведомления, соответствующий шаблону **t_comm_dlv_txt**. Если же текст уведомления об успешной доставке был переопределён (через параметр **r_deliv0** при формировании строки-команды на отправку SMS), то параметр **ntf_dlv_txt** будет содержать либо имя нестандартного (не равного **t_comm_dlv_txt**) шаблона из блока [DSM_NOTIF_FUNC_TEXT], либо текст.

ntf_no_dlv_txt – отображает использованный текст SMS-уведомления о статусе доставки, если изначальное сообщение не было доставлено. Если равен “0”, то использовался стандартный текст уведомления, соответствующий шаблону **t_comm_no_dlv_txt**. Если же текст уведомления о неуспешной доставке был переопределён (через параметр **r_deliv0** при формировании строки-команды на отправку SMS), то параметр **ntf_no_dlv_txt** будет содержать либо имя нестандартного (не равного **t_comm_no_dlv_txt**) шаблона из блока [DSM_NOTIF_FUNC_TEXT], либо текст.

ntf_prt_dlv_txt – отображает использованный текст SMS-уведомления о статусе доставки, если изначальное сообщение было доставлено лишь частично (не все части многосоставного сообщения). Если равен "0", то использовался стандартный текст уведомления, соответствующий шаблону **t_comm_prt_dlv_txt**. Если же текст уведомления о частичной доставке был переопределён (через параметр **r_deliv0** при формировании строки-команды на отправку SMS), то параметр **ntf_prt_dlv_txt** будет содержать либо имя нестандартного (не равного **t_comm_prt_dlv_txt**) шаблона из блока [DSM_NOTIF_FUNC_TEXT], либо текст.

ntf_rep_src – отображает использованный адрес отправителя при посылке SMS-уведомления о статусе доставки. Если равен "0", то в качестве номера отправителя использовался номер получателя изначального сообщения.

ntf_rep_dst – отображает использованный адрес получателя при посылке SMS-уведомления о статусе доставки. Если равен "0", то в качестве номера получателя использовался номер отправителя изначального сообщения. Также возможны следующие варианты (помимо обычного переопределения номера получателя): а) список dst-адресов через запятую (SMS-уведомление о статусе доставки было отправлено нескольким получателям); б) имя шаблона из блока конфигурации [DSM_NOTIF_REPORT_LISTS] (SMS-уведомление было отправлено на номера, привязанные к имени шаблона).

ntf_sndr_stream+ntf_sndr_smpp – указывает на экземпляр рассыльщика (stream_id=ntf_sndr_stream) и конкретный smpp-коннект (smpp_defin=ntf_sndr_smpp), использованные для отправки SMS-уведомления о статусе доставки. По умолчанию данный параметр равен "0" (т.е. отправка уведомления происходит через тот же коннект, что и для изначального SMS). Если же параметр, например, равен "1+10.1.2.3:7788:user1", то это означает, что уведомление было отправлено через коннект 10.1.2.3:7788:user1 от сервиса sms_sender_1.exe.

ntf_rep_spy – обозначает использованный тип параметра r_deliv0 для SMS-уведомления о статусе доставки. Если значение равно "0", то статус доставки уведомления не отслеживался.

6.2. Результат обработки отчётов о доставке. Специальный результат-файл

Формат файла – "yyyymmddhhmiss_dsm-id_part.dsmres".

Формат строки в файле – "begin_dt_op;+end_time;+ip;+port;+login;+dst;+src;+dt_id;+mess_id;+n_stream_id;+dsm_id;+rd_type;+part_msg_txt;+p_sm_num;+p_seq_num;+p_brr_cnt_ost;+p_resp_list_str;+p_part_send_resp_st;+p_part_deliv_st;+qsm=*=qsm".

begin_dt_op – время (в формате yyyymmddhhmiss) попадания SMS в программную очередь рассыльщика (sms_sender_N.exe).

end_time – время окончания формирования отчёта о доставке (в формате **yyyymmddhhmiss**).

ip, port и **login** – ip-адрес, порт и логин для подключения к SMS-центру (блок [CONNECTS]).

dst – номер получателя.

src – номер отправителя.

dt_id – уникальный идентификатор сообщения (односоставного или многосоставного), основанный на времени от начала эпохи.

mess_id – уникальный идентификатор submit_sm-сообщения, полученный от SMS-центра в ответном *submit_sm_resp*. Каждая часть многосоставного SMS-сообщения имеет свой уникальный mess_id (или message_id).

n_stream_id – идентификатор рассыльщика, через который было отправлено сообщение (например, n_stream_id=1, если SMS было отправлено через sms_sender_1.exe).

dsm_id – идентификатор сервиса обработки отчётов (например, dsm_id=1 для sms_sender_1dsm.exe), сформировавшего отчёт.

rd_type – тип обработки. Возможные значения: **simpl_track** (простое отслеживание статусов отправки каждой части SMS), **act_track** (активное отслеживание статусов отправки и доставки), **psv_track** (пассивное отслеживание статусов отправки и доставки), **act_notif** (активное отслеживание статусов отправки и доставки с последующим SMS-уведомлением на адрес отправителя), **psv_notif** (пассивное отслеживание статусов отправки и доставки с последующим SMS-уведомлением на адрес отправителя).

part_msg_txt – содержание части сообщения.

p_sm_num – порядковый номер части сообщения.

p_seq_num – порядковый номер отправленного submit_sm в рамках конкретной smpp-сессии. Максимальное значение - 2000000000, после - сбрасывается до 1.

p_brr_cnt_ost – неизрасходованные попытки повторной отправки части сообщения (если на отправленный submit_sm SMS-центр ответил submit_sm_resp-ом, где статус отправки не равен “0”).

p_resp_list_str – строка, содержащая статусы отправки (в рамках пакета submit_sm_resp), полученные от SMS-центра при попытках отправки части сообщения. Строка всегда начинается с “b-> ...”. Например, при успешной отправке с первого раза данный параметр будет равен подстроке “b->0”.

p_part_send_resp_st – итоговый статус отправки части сообщения. Если равно “1”, то часть сообщения была успешно отправлена, если “2” - сообщение отправлено не было.

p_part_deliv_st – итоговый статус доставки части сообщения. Возможные значения: **0** (механизм запроса статуса доставки не был задействован), **1** (сообщение доставлено), **2** (сообщение отправлено, но не доставлено), **3** (сообщение не было отправлено).

qsm=*=qsm – информация о статусе использования запроса *query_sm* в случае, когда **rd_type** равен **"act_track"** (активное отслеживание статуса доставки) или **"act_notif"** (активное отслеживание статуса доставки с последующим уведомлением на номер отправителя). Если **rd_type="psv_track / psv_notif / smpl_track"**, то содержание (*) данного параметра равно **"0"** (итого, **qsm=0=qsm**). Для **rd_type="act_track / act_notif"** подстрока **"qsm=*=qsm"** будет выглядеть так – **"1|p_qsm_b_cnt|p_qsm_cnt_ost|p_qsm_arr_com_st_res|p_qsm_arr_msg_st_res"** (параметры разделены посредством **"|"**).

p_qsm_b_cnt – начальное количество попыток запроса статуса доставки части сообщения (посредством *query_sm*). Выставляется в блоке конфигурации **[PROCESSING_DELIVER_SM_PARAMS]** (параметр **query_sm_send_cnt**) для каждого коннекта (*smpp_defin*). Нулевое значение указывает либо на отсутствие ограничений на количество отправленных *query_sm*, либо на отсутствие необходимости в отправке *query_sm* (т.е. отчёт о доставке был получен без запросов *query_sm*).

p_qsm_cnt_ost – остаток запросов *query_sm* на момент окончания обработки части сообщения. Нулевое значение указывает либо на отсутствие ограничений на количество отправленных *query_sm*, либо на отсутствие необходимости в отправке *query_sm* (т.е. отчёт о доставке был получен без запросов *query_sm*).

p_qsm_arr_com_st_res – содержит информацию о статусах отправки *query_sm*, соответствующих каждому отправленному *query_sm* (*qsm*) в формате **"b->qsm_status1->qsm_status2-> ..."**. Если в рамках обработки части SMS отправка *query_sm* не осуществлялась, то значение данного информационного параметра будет содержать только символ **"b"**.

p_qsm_arr_msg_st_res – содержит информацию о статусах доставки части сообщения, соответствующих каждому запросу *query_sm* (*qsm*) в формате **"b->deliv_status1->deliv_status2-> ..."**. Если в рамках обработки части SMS отправка *query_sm* не осуществлялась, то значение данного информационного параметра будет содержать только символ **"b"**. Расшифровка статусов доставки представлена ниже.

-1 (NOINF_INTERNAL) – выставляется, когда запрос *query_sm* не был успешно отправлен на SMS-центр.

1 (ENROUTE) – сообщение всё ещё на этапе доставки.

2 (DELIVRD) – сообщение успешно доставлено.

3 (EXPIRED) – срок жизни сообщения на SMS-центре истёк.

4 (DELETED) – сообщение было удалено из очереди доставки.

5 (UNDELIV) – сообщение не было доставлено.

6 (ACCEPTD) – аналогичен статусу 2 (DELIVRD).

7 (UNKNOWN) – статус доставки неизвестен.

8 (REJECTD) – сообщение было отклонено SMS-центром.

7. Логирование. Общие сведения

По умолчанию лог-файлы пишутся в директорию, определяемую параметром **log_dir** (блок конфигурации [MAIN]), который возможно переопределить вручную (с условием, что будут выданы права на запись пользователю, под которым планируется запускать сервисы).

Формат имени основного лог-файла - “**DATE_node_id_serv_name.log**”, где *DATE* – текущая дата, *node_id* – одноимённый параметр из блока [MAIN], *serv_name* – имя сервиса без расширения “.exe”. Например, имя лог-файла для sms_sender_1.exe (при node_id=1 и текущей дате равной 20210131) – “20210131_1_sms_sender_1.log”, для sms_sender_0dsm.exe – “20210131_1_sms_sender_0dsm.log”, для sms_sender_0rx.exe – “20210131_1_sms_sender_0rx.log”.

Строка в лог-файле (в общем случае) представляет собой конструкцию вида “**LOG_TYPE;+HH:MI:SS.mmmmmm;+LOG_INFO**” (данные разделены посредством “;+”), где *LOG_TYPE* – тип информационного сообщения (например, info), *HH* – часы, *MI* – минуты, *SS* – секунды, *mmmmmm* – микросекунды, *LOG_INFO* – информация.

Возможные варианты *LOG_TYPE*: info, error, smpp_info, smpp_error, udp_info, udp_error, dgram_info, dgram_error.

info – информация общего характера.

error – ошибки общего характера.

smpp_info – информация о трафике smpp.

smpp_error – ошибки, связанные с smpp-трафиком.

udp_info – информация о трафике UDP.

udp_error – ошибки, связанные с UDP-трафиком.

dgram_info – информация о DGRAM-трафике.

dgram_error – ошибки, связанные с DGRAM-трафиком.

Формат имени STDERR-файла - “**serv_name.stderr**”. Содержит записи об ошибках со стандартного вывода, которые не были отловлены встроенными в сервис механизмами.

Формат имени STDOUT-файла - “**serv_name.stdout**”. Содержит записи о событиях со стандартного вывода, которые не были отловлены встроенными в сервис механизмами.

7.1. Логирование. SMPP-трафик

Формат – “**LOG_TYPE**;**HH:MI:SS.mmmmmm**;**op**;**smpp_defin**;**type**;**status**;**dst**;**src**;**valid_time**;**reg_deliv**;**da_ton**;**da_npi**;**sa_ton**;**sa_npi**;**brr_cnt**;**wfr_t**;**rd_type**;**msg**;**msg_part_num**;**msg_id**;**dcs**;**seq**;**err_txt**” (значения разделены посредством “;”). Если какое-либо значение из представленных (начиная с **op**) не определено, то оно заменяется на символ “-”.

op – имя smpp-операции, например, **submit_sm** или **submit_sm_resp** (в соответствии со спецификацией smpp 3.4).

smpp_defin – описание smpp-коннекта в виде **smc_ip:smc_port:smc_login**.

type – тип трафика, например, “**sys**” (системный) или “**ESME**” (относится к полезному трафику).

status – статус отправки той или иной команды на SMS-центр. Извлекается из resp-ответа, например, при отправке **submit_sm** в ответ поступает **submit_sm_resp** из которого извлекается содержание значения **status**.

dst – номер получателя.

src – номер отправителя.

valid_time – время жизни SMS на SMS-центре в формате, соответствующем спецификации smpp 3.4.

reg_deliv – указывает на то, включен ли механизм уведомления о статусе доставки. Если равен 1, то отчет о доставке был заказан у SMS-центра.

da_ton – Type Of Number (тип номера) для номера получателя (в соответствии со спецификацией smpp 3.4).

da_npi – Numbering Plan Identification (план нумерации) для номера получателя (в соответствии со спецификацией smpp 3.4).

sa_ton – Type Of Number (тип номера) для номера отправителя (в соответствии со спецификацией smpp 3.4).

sa_npi – Numbering Plan Identification (план нумерации) для номера отправителя (в соответствии со спецификацией smpp 3.4).

brr_cnt – количество оставшихся попыток переотправки части сообщения (*submit_sm*), если статус, извлечённый из *submit_sm_resp*, не равен "0" (статус, указывающий на успешную отправку *submit_sm*).

wfr_t – таймаут (в секундах) ожидания *resp*-ответов (например, *submit_sm_resp*).

rd_type – тип отчёта о доставке (*smpl_track* / *act_track* / *psv_track* / *act_notif* / *psv_notif*).

msg – содержание сообщения (или части сообщения, если речь про многосоставное сообщение).

msg_part_num – идентификатор части сообщения.

msg_id – идентификатор части сообщения, присвоенный со стороны SMS-центра.

dcs – кодировка сообщения. Используется только значение "8" (в соответствии со спецификацией *smpp* 3.4).

seq – идентификатор (номер из последовательности), присвоенный части SMS в рамках конкретной *smpp*-сессии.

err_txt – текст ошибки или "OK", если ошибки отсутствуют. Если дело касается ошибок, основанных на статусах *resp*-ов, то подставляются текстовки, обозначенные в блоке конфигурации [*SMPP_RESP_CODES*].

7.2. Логирование. UDP-трафик

Формат - "**LOG_TYPE**;**HH:MI:SS.mmmmmm**;**+op**;**+host**;**+port**;**+udp_msg**;**+err_txt**" (значения разделены посредством ";+").

op – тип операции. Возможные варианты (в рамках сервиса): *send_ans* (отправка ответа на поступивший запрос), *get_req* (получение запроса).

host – адрес хоста, принявшего / отправившего *udp*-пакет. В рамках сервиса значение всегда равно адресу хоста, отправившего *udp*-пакет.

port – порт на стороне хоста, принявшего *udp*-пакет. В рамках сервиса выставляется стандартное значение "N" (т.е. No), т.к. сервис не отправляет запросы, а только принимает и отвечает.

udp_msg – содержание *udp*-пакета.

err_txt – текст ошибки или "OK", если ошибки отсутствуют.

7.3. Логирование. DGRAM-трафик

Формат - "LOG_TYPE;+HH:MM:SS.mmmmmmm;+op;+dgram_file;+dgram_msg;+err_txt"
(значения разделены посредством ";+").

op – тип операции. Возможные варианты: get_req (получение информации), send_str (отправка информации), new_dgram (подключение к dgram-сокету), close_dgram (окончание передачи).

dgram_file – DGRAM-файл (сокет), принявший dgram-пакет (или на который был отправлен dgram-пакет).

dgram_msg – содержание dgram-пакета.

err_txt – текст ошибки или "OK", если ошибки отсутствуют.

8. UDP-запросы к сервисам

1) ping. Отправлять на порт, указанный в блоке конфигурации [UDP_PORTS]. Если сервис активен, то в ответ получим "OK".

2) Получение статистических данных на конкретный момент времени от сервисов рассыльщика (sms_sender_N.exe). Для получения статистики необходимо отправить UDP-пакет, содержащий строку "v_1con_v;+v_0con_v;+v_cque_v" (v_1con_v – запрос кол-ва активных коннектов, v_0con_v – запрос кол-ва неактивных коннектов, v_cque_v – общее кол-во SMS, ожидающих отправки). В ответ поступит udp-пакет с содержанием вида "num1;+num2;+num3", где num1 соответствует результату запроса v_1con_v, num2 – v_0con_v, num3 – v_cque_v.

3) Получение статистических данных на конкретный момент времени от сервисов-ресиверов (sms_sender_Nrx.exe). Механизм аналогичен п.2, но возможен запрос только параметров "v_1con_v;+v_0con_v".

4) Получение статистических данных на конкретный момент времени от сервисов обработки отчётов о доставке (sms_sender_Ndsm.exe). Доступный параметр – v_dsmque_v (общее количество SMS, ожидающих подтверждения доставки).

Пример скрипта для отправки UDP.

```
send udp.pl      [---] 0 L:[ 1+17 18/ 28] *(724 /1036b) 0101 0x065
#!/usr/bin/perl

use IO::Socket::INET;
#####
my ($host_l,$port_l,$send_str_l)=(undef,undef,undef);
if ( defined($ARGV[0]) && $ARGV[0]=~/^\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}$/ ) { $host_l=$ARGV[0]; }
else { print "Host/ip (ARGV-0) is not correct. Exit!\n"; exit; }
#####
if ( defined($ARGV[1]) && $ARGV[1]=~/^\d+$/ ) { $port_l=$ARGV[1]; }
else { print "Port (ARGV-1) is not correct. Exit!\n"; exit; }
#####
if ( defined($ARGV[2]) && length($ARGV[2])>0 ) { $send_str_l=$ARGV[2]; }
else { print "Empty send_str (ARGV-2). Exit!\n"; exit; }
#####
my $request=undef;
my $portaddr=IO::Socket::INET->new(Proto=>'udp',PeerPort=>$port_l,PeerAddr=>$host_l);
$portaddr->send($send_str_l);
eval {
    local $SIG{ALRM}=sub { print "Can't get request from '$host_l:$port_l' (send_str='$send_str_l')\n"; die; };
    alarm(1);
    eval { $portaddr->recv($request,16384,0); };
    alarm(0);
};
alarm(0);
print "Send='$send_str_l' to '$host_l:$port_l': OK\n";
print "Request='$request'\n";
close($portaddr);
```

9. DGRAM-запросы к сервисам

Механика запросов аналогична UDP-запросам, но с учётом специфики взаимодействия с DGRAM-сокетами (необходимо открывать как сокет для отправки данных, так и сокет для получения ответа).

Пример скрипта для отправки DGRAM.

```
send_dgram.pl  [--] 0 L:[ 1+ 0 1/ 30] *(0 /1150b) 0035 0x023
#!/usr/bin/perl
use strict;
use warnings;
use IO::Socket::UNIX;
my ($send_str_l,$send_dgram_name)=(undef,undef);
if ( defined($ARGV[1]) && length($ARGV[1])>0 ) { $send_str_l=$ARGV[1]; }
else { print "Empty send_str (ARGV-1). Exit!\n"; exit; }
if ( defined($ARGV[0]) && $ARGV[0] =~ /^sms_sender/ ) { $send_dgram_name=$ARGV[0]; }
else { print "Empty send_dgram_name (ARGV-0). Exit!\n"; exit; }
my $dgram_file_s='/home/export/services/sms_sender/work/sms_sender_dgram/' . $send_dgram_name;
my $dgram_file_r='/tmp/tmp_sock.dgram';
my $request=undef;
my $portaddr_s=IO::Socket::UNIX->new(Peer=>$dgram_file_s,Type=>SOCK_DGRAM,Proto=>0);
my $portaddr_r=IO::Socket::UNIX->new(Local=>$dgram_file_r,Type=>SOCK_DGRAM,Proto=>0);
$portaddr_s->send($send_str_l.'->' . $dgram_file_r);
print "Send='$send_str_l' to '$dgram_file_s': OK\n";
close($portaddr_s);
eval {
    local $SIG{ALRM}=sub { print "Can't get request from '$dgram_file_r' (send_str='$send_str_l')\n"; die; };
    alarm(1);
    eval {
        $portaddr_r->recv($request,16384,0);
    };
    alarm(0);
};
alarm(0);
print "Get request='$request' from '$dgram_file_r'\n";
close($portaddr_r);
unlink $dgram_file_r;
```

10. Дополнительный функционал на основе ПО

10.1. Простой SMS-чат (simple_chat)

Сервис “SMS-чат” позволяет настроить общение отправителей с получателями (получателями могут быть и отправители) посредством отправки сообщения (SMS) на короткий номер (или номер в international-формате), привязанного к учётной записи (сконфигурированной в основном файле конфигурации - sms_sender.cfg в блоке [CONNECTS]) на СМС-Центре сотового оператора или агрегатора SMS-трафика.

Если по пунктам, то для конечного пользователя услуга выглядит так:

- 1) абонент А (состоящий в списке отправителей) отправляет SMS на короткий номер (или номер в федеральном формате);
- 2) абоненты Б (из списка получателей) получают SMS, подписанную либо номером, на который абонента А отправил SMS, либо именем SMS-чата.

10.1.1. Файл конфигурации simple_chat.cfg (описание)

Файл конфигурации сгенерируется при первом запуске в директории “.../cfg_files”.

10.1.1.1. [CHAT_LIST]

```
[CHAT_LIST]
#chat_name - unique chat name (must be alpha-numeric). You can assign different smpp_defin + chat_dst_num
#to one #chat_name.
###
#smpp_defin - smpp_ip:smpp_port:smpp_login.
#chat_dst_num - short number (or msisdn) for sending messages to this number.
###
#After sending SMS to #chat_dst_num SMS_SENDER resend this message (chat-message) to list of msisdn (CHAT_RECEIVER_LIST).
###
#After configuring and reload_cfg (or restart) at directories ".../cfg_files/simple_chat/sender" and ".../cfg_files/simple_chat/receiver"
#will be created (automatically) files with name #chat_name.
#Put msisdn allowed for send chat-messages to file (sender-file) ".../cfg_files/simple_chat/sender/#chat_name" (one line = one "msisdn;nickname").
#Put msisdn for receive chat-messages to file (receiver-file) ".../cfg_files/simple_chat/receiver/#chat_name" (one line = one msisdn).
###
#If two chats (for example, CHAT1 and CHAT2) use one #chat_dst_num (for example, 0001) and, for example, one msisdn exists at different sender-files, then
#this msisdn must insert "#CHAT_NAME" at the beginning of a message for choose chat.
###
#chat_name      #smpp_defin      #chat_dst_num
##Example (commented)
#CHAT1          192.168.168.158:6789:test_login      000788
#CHAT1          192.168.168.100:6790:test_login0     000789
#CHAT2          192.168.168.150:6770:test_login11    000888
CHAT_TST        192.168.122.2:9900:GSMGW           000111
CHAT_TST0       192.168.122.3:9900:GSMGW           000112
CHAT_TST1       192.168.122.3:9900:GSMGW           000112
[CHAT_LIST]
```

Основной блок конфигурации сервиса, где одна строка - настройка для конкретного чата.

Формат строки - “**chat_name** **smpp_defin** **chat_dst_num**”.

chat_name - имя чата. Должно содержать только латинские символы и цифры (без пробелов).

smpp_defin - smsc_ip:smc_port:smc_login (**host:port:login** из блока [CONNECTS]). Коннект должен быть сконфигурирован в блоке [CONNECTS] либо в режиме приёма-передачи (**TRX**), либо в режиме приёма, т.е. **RX** (если рассылка будет осуществляться через коннект, отличный от указанного в данном блоке конфигурации).

chat_dst_num - короткий номер (либо номер в федеральном формате, например, 79240000000), на который осуществляется приём SMS, которые могут быть отправлены участниками списка отправителей на номер **chat_dst_num**. Номер **chat_dst_num** должен быть прописан на SMS-центре (на стороне оператора сотовой связи или агрегатора SMS-трафика) и привязан к **smpp_defin**.

После внесения изменений в этот блок конфигурации и рестарта ПО (или выполнения команды **reload_cfg**) в директории “**../cfg_files**” будет автоматически создана директория **simple_chat**, содержащая поддиректории **receiver** и **sender**, в которых расположены одноимённые файлы с именем **#chat_name**.

Файл “**../receiver/#chat_name**” - файл для внесения номеров получателей, где одна строка - один номер.

Возможно добавление в список получателей коротких номеров, т.е. возможен вариант, когда получателем сообщений может выступать сервис.

Файл “**../sender/#chat_name**” - файл для внесения номеров отправителей (те, кому разрешено отправлять SMS на номер **chat_dst_num**), где одна строка - один номер.

Также возможно дополнительно указать имя отправителя (nickname), добавив его к номеру через “;+”, например, “**79250000000;+ Иванов Иван Иванович**”.

Возможно добавление в список отправителей коротких номеров, т.е. возможен вариант, когда отправителем выступает сервис.

Примечание 1. Производить рестарт (или перечитывать конфиг) необходимо на тех экземплярах ПО (**sms_sender_n.exe** или **sms_sender_Nrx.exe**), для которых сконфигурированы коннекты (**smpp_defin**) как в блоке [CONNECTS], так и в файле конфигурации **simple_chat.cfg**.

Примечание 2. Основные настройки конфигурируемых SMS-чатов относятся к этому блоку. Остальные параметры (в блоках конфигурации ниже) имеют свои значения по умолчанию.

10.1.1.2. [CHAT_INCOMING_MESSAGES_PARAMS]

```
[CHAT_INCOMING_MESSAGES_PARAMS]
#proc_id - number of sender (1,2,3, etc) or receiver (0rx, 1rx, 2rx, etc).
###
#wait4all_msg_parts_timeout - wait for all parts of message send by participant of sender-file to #chat_dst_num (in seconds).
#Default value (if #chat_name not configured here) = 10.
###
#proc_id      #wait4all_msg_parts_timeout
##Example (commented)
1             15
[CHAT_INCOMING_MESSAGES_PARAMS]
```

Блок конфигурации, позволяющий контролировать процесс приёма всех частей многосоставных SMS.

Строка конфигурации имеет формат вида “**proc_id wait4all_msg_parts_timeout**”.

proc_id - идентификатор процесса рассыльщика (например, для **sms_sender_1.exe** идентификатор = “1”) или ресивера (например, для **sms_sender_2rx.exe** идентификатор = “2rx”).

wait4all_msg_parts_timeout - время ожидания (в секундах) всех частей многосоставного сообщения, отправленного кем-либо из списка “../sender/#chat_name”.

Значение по умолчанию - 10 секунд.

10.1.1.3. [CHAT_MESSAGES_PARAMS_FOR_RECEIVER_LIST]

```
[CHAT_MESSAGES_PARAMS_FOR_RECEIVER_LIST]
#sender_id_for_resend - sms_sender_XX (where XX = sender_id_for_resend) for send chat messages.
#Default value (if #chat_name not configured here) = 1.
###
#smpp_defin_for_resend - smpp_defin for resend chat-message to list of msisdn (receiver-file).
#Possible values: 1) default (smpp_defin from where received chat-message);
#2) smpp_ip:smpp_port:smpp_login (smpp_defin of configured connect);
#3) any_connect (using mechanism "syn_params" of CONNECTS-block);
#4) any_connect_src (using mechanism of [CONNECTS_SYNONYM_SRC]);
#5) any_connect_dst (using mechanism of [CONNECTS_SYNONYM_DST]);
#Default value (if #chat_name not configured here) = default (case 1).
###
#is_lat - transliteration option. Possible values: 1 - transliteration ON, 0 - transliteration OFF.
#Default value (if #chat_name not configured here) = 0.
###
#valid_time - time period on SMSC to resend SMS if first-try-delivery was unsuccessful.
#Possible values:
#1) def (10 min); 2) 1-23 (hours). For example, 5 or 20.
#Default value (if #chat_name not configured here) = def.
###
#deliv_status_mode - enable collecting delivery status of short messages. Possible values:
#1) 0. No delivery status tracking;
#2) act_track. Delivery status active tracking (with using query_sm);
#3) psv_track. Delivery status passive tracking;
#4) smpl_track. Just submit_sm_resp-statuses.
#Default value (if #chat_name not configured here) = 0 (case 1).
###
#wfr_timeout - wait for submit_sm_resp timeout (sec). After this timeout message (submit_sm) can not be handled.
#Default value = 5 (sec).
###
#brr_count - bad resp resend count. Resend count if submit_sm_resp contains status-record with not "0" value.
#Default value (if #chat_name not configured here) = 0 (no resend tries).
###
#chat_name      #sender_id_for_resend  #smpp_defin_for_resend      #is_lat      #valid_time      #deliv_status_mode      #wfr_timeout      #brr_count
##Example (commented)
#CHAT1          1                  192.168.168.150:6689:test_login00      0             def              0                  5              0
#CHAT2          2                  default                                0             def              0                  5              0
#CHAT_TST       1                  default                                0             def              0                  5              0
#CHAT_TST0      2                  any_connect                        0             12              smpl_track        10             10
[CHAT_MESSAGES_PARAMS_FOR_RECEIVER_LIST]
```

Блок конфигурации, позволяющий настроить параметры рассылки на список абонентов из файла “../receiver/#chat_name”.

Формат строки конфигурации -

“chat_name sender_id_for_resend smpp_defin_for_resend is_lat valid_time
deliv_status_mode wr_timeout brr_count”.

chat_name - имя чата. Должно содержать только латинские символы и цифры (без пробелов).

sender_id_for_resend - идентификатор (id) рассылщика (например, с id=1 - sms_sender_1.exe) для рассылки по списку “../receiver/#chat_name”.

Значение по умолчанию - 1.

smpp_defin_for_resend - коннект для рассылки по списку “receiver”. Коннект должен быть сконфигурирован в блоке [CONNECTS] для proc_id = sender_id_for_resend.

Возможные значения: 1) **default**, т.е. используется smpp-коннект/smpp_defin, указанный в параметре **smpp_defin** из блока [CHAT_LIST]). Также является значением по умолчанию;

2) **any_connect**. Используется механизм “syn_params” из блока [CONNECTS] (конфигурационный файл “sms_sender.cfg”);

3) **any_connect_src**. Используется механизм, реализованный через блок конфигурации [CONNECTS_SYNONYM_SRC];

4) **any_connect_dst**. Используется механизм, реализованный через блок конфигурации [CONNECTS_SYNONYM_DST].

is_lat - автоматическая транслитерация. Если параметр равен “1”, то транслитерация включена. Значение по умолчанию - **0**.

valid_time - время хранения (в часах) сообщения на SMS-центре в случае невозможности доставки на данный момент (например, абонент-получатель находится вне зоны действия сотовой сети). Возможные значения: *def* - 10 минут, диапазон целых чисел от 1 до 24 (или от 1h до 24h). Значение по умолчанию - **def**.

deliv_status_mode - параметр, указывающий на сценарий постобработки данных (статус отправки, статус доставки), полученных в результате отправки короткого сообщения.

Возможные значения:

- 1) **0** (т.е. механизм отслеживания статуса доставки не используется);
- 2) **act_track**. Включает в себя обработку статуса отправки и активное отслеживание статуса доставки посредством *query_sm*, если отчёт о доставке не поступил за время *pass_wait_t* (определяется в блоке [PROCESSING_DELIVER_SM_PARAMS]);
- 3) **psv_track**. Аналогичен *act_track*, но без использования *query_sm*;
- 4) **simpl_track**. Простое отслеживание статуса отправки.

Значение по умолчанию - **0**.

wfr_timeout - время ожидания (в секундах) *resp*-а (*submit_sm_resp*) на отправленный *submit_sm*. Если выставить равным 0, то будет использовано значение по умолчанию, равное 5-ти секундам. Значение по умолчанию - **5** секунд.

brr_count - количество повторных попыток отправки *submit_sm*, если в *submit_sm_resp* был получен статус, не равный “0” (т.е. отправка по какой-то причине не была успешной).

Значение по умолчанию - **0**.

10.1.1.4. [CHAT_ANSWER_DST]

```
[CHAT_ANSWER_DST]
#answer_dst_type - mechanism for assign DST_number_for_answer (for resend) after sending SMS to #chat_dst_num.
#Possible values: 1) chat_dst_num (DST_number = chat_dst_num);
#2) chat_name (DST_number = chat_name). In case 2 participants (of sender-file) can not send answers directly.
#Default (if #chat_name not configured here) = chat_dst_num.
###
#chat_name      #answer_dst_type
###Example (commented)
#CHAT1 chat_dst_num
[CHAT_ANSWER_DST]
```

Блок конфигурации, предоставляющий возможность корректировки номера отправителя (и, соответственно, короткий номер получателя при ответном сообщении от другого участника списка “../sender/#chat_name”) при рассылке на список “../receiver/#chat_name”.

Формат строки конфигурации - “**chat_name** **answer_dst_type**”.

chat_name - имя чата. Должно содержать только латинские символы и цифры (без пробелов).

answer_dst_type - механизм выбора номера-подписи (номера для ответа).

Возможные значения:

1) **chat_dst_num** (номер-подпись/номер для ответа = номер, прописанный в параметре chat_dst_num в блоке [CHAT_LIST]);

2) **chat_name** (в качестве подписи используется имя чата).

10.1.1.5. [IF_SENDER_NUM_CAN_USE_MORE_ONE_CHAT]

```
[IF_SENDER_NUM_CAN_USE_MORE_ONE_CHAT]
#Default message if sender num can use more one chat
#(sender num included to different files like ".../cfg_files/simple_chat/sender/#chat name") =
#Dear _SENDER_NUM_, You are included into chats: _CHAT_LIST_. Please, write "#CHAT_NAME:" at the begining of a message for choose chat.
###
# _CHAT_LIST_ = list of available chats.
###
#sender_num=message_if_more_one_chat
79273999534=Uvazhaenyj _SENDER_NUM_, Вы включены в следующие чаты: _CHAT_LIST_. Please, write "#CHAT_NAME:" at the begining of a message for choose chat.
###
[IF_SENDER_NUM_CAN_USE_MORE_ONE_CHAT]
```

Блок конфигурации, позволяющий переопределять текст ответа по умолчанию в случае, когда абонент-отправитель состоит в двух (и более) чатах, но при отправке сообщения не был явно указан чат-получатель в теле сообщения.

10.1.1.6. [CHAT_MSG_TEMPLATES]

Блок конфигурации, позволяющий ограничить количество возможных вариантов текста для отправки списку получателей.

(В РАЗРАБОТКЕ)

10.1.1.7. [CHAT_TARIFICATION_FILES]

Блок конфигурации, позволяющий настраивать параметры тарификации.

(В РАЗРАБОТКЕ)

10.1.1.8. [CHAT_MSG_LIMIT]

Блок конфигурации, позволяющий настраивать лимиты на отправку.

(В РАЗРАБОТКЕ)

10.1.1.9. [CHAT_DEDICATED_LOG]

Блок конфигурации, позволяющий вывести в отдельный лог-файл события, связанные с тем или иным #chat_name (блок [CHAT_LIST]).

(В РАЗРАБОТКЕ)

11. Используемые в ПО модули Perl5

1. Net::SMPP.
2. IO::Socket::INET.
3. IO::Socket::UNIX.
4. IO::Select.
5. Encode.
6. POSIX.

7. Time::HiRes.
8. Time::Local.
9. Data::Dumper.
10. Data::Buffer.
11. Cwd.
12. FileHandle.
13. List::MoreUtils.
14. Safe.
15. Crypt::Blowfish.
16. Crypt::CBC.
17. Crypt::RSA.
18. Crypt::Primes.
19. Crypt::Random.
20. Crypt::RSA::Key::Public.
21. Crypt::RSA::ErrorHandler.
22. Crypt::RSA::Key.
23. Crypt::RSA::DataFormat.
24. Digest::MD2.
25. Digest::MD5.
26. Digest::SHA1.
27. Class::Loader.
28. Convert::ASCII::Armour.
29. Math::Pari.
30. Sort::Versions.
31. Tie::EncryptedHash.
32. Carp.
33. FindBin.
34. Crypt::RSA::ES::OAEP.
35. Async.
36. Storable.
37. Compress::Bzip2.
38. PAR::Packer.