



# **ПАК ЯКОРЬ-ИКС**

## **РУКОВОДСТВО АДМИНИСТРАТОРА**

(версия 1.0)

**Авторские права**

Без предварительного письменного разрешения, полученного от ООО «НЕОС», документ и любые выдержки из него, с изменениями и переводом на другие языки, не должны быть воспроизведены или использованы.

## Содержание

1 ОБЩИЕ СВЕДЕНИЯ .....	4
1.1 НАЗНАЧЕНИЕ ДОКУМЕНТА .....	4
1.2 СОСТАВ ДОКУМЕНТА .....	4
1.3 ТЕХНИЧЕСКАЯ ПОДДЕРЖКА.....	5
1.3.1 Производитель .....	5
1.3.2 Служба технической поддержки .....	5
1.4 ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ И СОКРАЩЕНИЯ.....	6
1.5 ИСТОРИЯ ИЗМЕНЕНИЙ .....	8
2 ОПИСАНИЕ СИСТЕМЫ .....	9
2.1 АРХИТЕКТУРА СИСТЕМЫ.....	10
3 НАСТРОЙКА ПАРАМЕТРОВ КОНФИГУРАЦИИ .....	11
3.1 УСЛОВНЫЕ ОБОЗНАЧЕНИЯ.....	11
3.2 НАСТРОЙКА ОСНОВНЫХ ПАРАМЕТРОВ (CONFIG.PROPERTIES).....	13
3.3 НАСТРОЙКА НЕСКОЛЬКИХ ПУ (HOST_MAP) .....	24
3.4 НАСТРОЙКА ПОЛЬЗОВАТЕЛЕЙ (USERS.XML) .....	25
3.5 ХРАНИЛИЩЕ СЕРТИФИКАТОВ И КЛЮЧЕЙ.....	26
3.5.1 Создание хранилища ключей из сертификатов .....	26
3.5.2 Команды утилит keytool и openssl .....	27
4 ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ .....	29
4.1 ДОСТУП С ИСПОЛЬЗОВАНИЕМ КОМАНДЫ SSH .....	29
4.2 УПРАВЛЕНИЕ РАБОТОЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	29
4.3 ПЕРЕЗАПУСК СИСТЕМЫ .....	30
4.4 ПРОВЕРКА РАБОТОСПОСОБНОСТИ .....	30
4.4.1 Проверка работы ПО.....	30
4.4.2 Проверка запуска приложения .....	30
4.4.3 Проверка состояния приложения после запуска .....	31
5 ЖУРНАЛЫ.....	32
5.1 ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ ЖУРНАЛОВ .....	32
5.2 ОПИСАНИЕ ЖУРНАЛА СОБЫТИЙ (COMMON.LOG) .....	32
5.3 ОСНОВНОЙ ЖУРНАЛ РАБОТЫ ПРИЛОЖЕНИЯ (ASN-GW.LOG) .....	35
5.4 ЖУРНАЛ РАБОТЫ КОМПОНЕНТЫ SPRING (SPRING.LOG).....	35
5.5 СИСТЕМНЫЙ ЖУРНАЛ (SYSTEM.LOG) .....	35

## 1 Общие сведения

### 1.1 Назначение документа

Настоящее руководство содержит руководство пользователя по работе с ПАК ЯКОРЬ-ИКС, разработки ООО «НЕОС».

### 1.2 Состав документа

Настоящее руководство состоит из следующих основных частей:

«Общие сведения» — раздел, описывающий назначение и состав документа, содержащий сведения о производителе и технической поддержке;

«Описание системы» — раздел, содержащий сведения о системе, ее характеристики и архитектуру системы;

«Настройка параметров конфигурации» — раздел, содержащий информацию о настройке основных параметров системы с помощью конфигурационных файлов;

«Техническое обслуживание» — раздел, содержащий описание работы пользователя с программным обеспечением;

«Журналы» — раздел, содержащий информацию о журналах системы.

#### **Внимание!**

Перед установкой и началом эксплуатации изделия необходимо внимательно ознакомиться с паспортом изделия и эксплуатационной документацией.

Данный документ должен постоянно находиться при изделии.

## 1.3 Техническая поддержка

Техническая поддержка, а также дополнительное консультирование по вопросам, возникающим в процессе эксплуатации приложения, осуществляются производителем и службой технической поддержки.

### 1.3.1 Производитель

ООО «НЕОС»

197342, Санкт-Петербург

ул. Белоостровская, д. 6, литера А, офис №55

Тел.: (812) 779-13-79

WEB: <https://www.neo-s.ru>

E-mail: [sales@neo-s.com](mailto:sales@neo-s.com)

### 1.3.2 Служба технической поддержки

ООО «НЕОС»

197342, Санкт-Петербург

ул. Белоостровская, д. 6, литера А, офис №55

Тел.: (812) 779-13-79 доп. 1 (круглосуточно)

WEB: <https://www.neo-s.ru>

E-mail: [support@neo-s.com](mailto:support@neo-s.com)

## 1.4 Используемые термины и сокращения

Используемые в настоящем документе термины и сокращения приведены в таблице 1.

Таблица 1 — Используемые термины и сокращения

Термин/Сокращение	Расшифровка
БД	База данных.
ИС БД	Информационные системы, содержащие базы данных.
КПД	Канал передачи данных.
НСД	Несанкционированный доступ.
ОКС-7	(Общий Канал Сигнализации №7) Набор сигнальных телефонных протоколов, используемых для настройки большинства телефонных станций.
ОРМ	Оперативно-розыскные мероприятия.
ОС	Операционная система.
ПО	Программное обеспечение.
ПК	Программный комплекс.
ПУ, ПУ ОРМ	Пульт управления уполномоченного государственного органа, осуществляющего оперативно-розыскную деятельность.
СОРМ	Система технических средств для обеспечения функций оперативно-розыскных мероприятий.
СХД	Система хранения данных.
ТС ОРМ	Технические средства (оборудование транзитных, оконечно-транзитных и оконечных узлов связи с использованием технологии коммутации каналов и (или) коммутации пакетов информации, входящих в состав сети связи общего пользования и выделенных сетей связи фиксированной телефонной связи, включая программное обеспечение), обеспечивающие выполнение установленных действий при проведении оперативно-розыскных мероприятий.
RTP	(Real Time Transport Protocol) протокол передачи данных в реальном времени.
SIP	(Session Initiation Protocol) протокол инициирования сеансов связи.
SNMP	(Simple Network Management Protocol) простой протокол сетевого управления. Протокол для сбора и организации информации об устройствах в IP-сетях.
SSH	(Secure Shell) сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов).
SSL	(Secure Sockets Layer) криптографический протокол с асимметричной криптографией для аутентификации ключей обмена

TCP/IP	(Transmission Control Protocol/Internet Protocol) протокол управления передачей/интернет протокол.
Trap	Сигнал о каком-либо событии, например, ошибке, отправляемый устройством с поддержкой протокола SNMP.
UDP	(User Datagram Protocol) протокол передачи датаграмм пользователей.

## 1.5 История изменений

История изменений настоящего документа фиксируется в таблице 2.

Таблица 2 — История изменений

Дата	Версия документа	Версия продукта	Комментарий
01.03.2025	1.0	Версия ПО: 1.0	Первая версия документа



## 2 Описание системы

Программное обеспечение «ПАК ЯКОРЬ-ИКС» (далее — ПАК ЯКОРЬ-ИКС) производства ООО «НЕОС» предназначено для накопления, хранения, поиска и предоставления информации из информационных систем организаторов распространения информации (ОРИ), собственников или иных владельцев технологических сетей связи имеющих номер автономной системы (АС), провайдеров хостинга, а также информации о пользователях и предоставленных им услугах по запросу от ПУ в соответствии с приказами Минкомсвязи России № 571 от 29.10.2018, Минкомсвязи России № 646 от 05.11.2019, Минцифры России № 935 от 01.11.2023.

Программно-аппаратный комплекс «ЯКОРЬ-ИКС» получает данные с точек съема трафика по протоколу взаимодействия ТС ОРМ с ИС БД ОРМ в соответствии с Приказами Минкомсвязи № 571, № 646, №935 и интегрируется с информационными системами организаторов распространения информации (ОРИ), собственников или иных владельцев технологических сетей связи имеющих номер автономной системы (АС), провайдеров хостинга. Полученные данные «ЯКОРЬ-ИКС» собирает и хранит в собственной базе данных, из которой по запросу предоставляет их на ПУ ОРМ.

Предусмотрены следующие варианты взаимодействия с пультом управления (ПУ):

- Подключение посредством каналов передачи данных № 1–4 с использованием протокола ASN.1 для приема и обработки запросов.

Форматы поисковых запросов регламентируются соответствующими версиями модулей протокола взаимодействия ПУ и ПТС ОРИ, указанных в схемах 2 и 3 Приказа Минцифры России от 29 октября 2018 г. № 571.

Соединение между ПУ и ПТС ОРИ инициируется посредством безопасного транспортного уровня (TLS) последней доступной версии. Процесс аутентификации реализован с применением сертификата формата X.509.

- Организация взаимодействия по единому каналу передачи данных с применением протоколов GraphQL и WebSocket.

Обмен данными в едином канале передачи данных на прикладном уровне должен осуществляться по протоколам HTTP 1.1 и WebSocket.

Собственная база данных ПАК ЯКОРЬ-ИКС может размещаться на внешней системе хранения данных (далее — СХД), удовлетворяющей требованиям Постановления Правительства № 673 от 28.05.2019.

С информационной системой оператора ПАК ЯКОРЬ-ИКС интегрируется через стандартные средства или с использованием брокера сообщений для асинхронной обработки запросов на получение данных.

ПАК ЯКОРЬ-ИКС обеспечивает:

- хранение информации из информационных систем организаторов распространения информации (ОРИ), собственников или иных владельцев технологических сетей связи, имеющих номер автономной системы (АС), провайдеров хостинга по запросу от ПУ в течение 3 лет с даты формирования;
- передачу, доставку и/или обработку в соответствии с требованиями приказов Минкомсвязи России № 571 от 29.10.2018, Минкомсвязи России № 646 от 05.11.2019, Минцифры России № 935 от 01.11.2023

ПАК ЯКОРЬ-ИКС предназначен для выполнения следующих функциональных характеристик:

- накапливать, хранить, осуществлять поиск и предоставлять информацию из информационных систем организаторов распространения информации (ОРИ), собственников или иных владельцев технологических сетей связи, имеющих номер автономной системы (АС), провайдеров хостинга по запросу от ПУ;
- предоставлять информацию о пользователях и предоставленных им услугах по запросу от ПУ;
- принимать и обрабатывать запросы с пульта управления;
- получать данные из информационных систем и сохранять в системе хранения данных (СХД) в структурированном виде;

- извлекать информацию из СХД на основании запросов пульта управления;
- выполнять поисковые запросы от ПУ по согласованным критериям;
- обеспечивать доступ к неформатированным данным (видео, аудио, изображения);
- хранить результаты отложенных запросов и обеспечение их удаления или обновления по запросу ПУ;
- сбор и хранение информации о событиях маршрутизации, логах доступа, а также других критически важных событиях;
- передачу на ПУ по запросу получаемых из ИС данных без дополнительной обработки;
- осуществлять резервное копирование данных;
- временно хранить отчёты по выполненным задачам поиска пульта управления;
- контролировать попытки несанкционированного доступа к системе.

## 2.1 Архитектура системы

Система может использоваться на сети оператора связи.

На рисунке ниже приведена общая архитектура программного обеспечения ПАК ЯКОРЬ-ИКС.

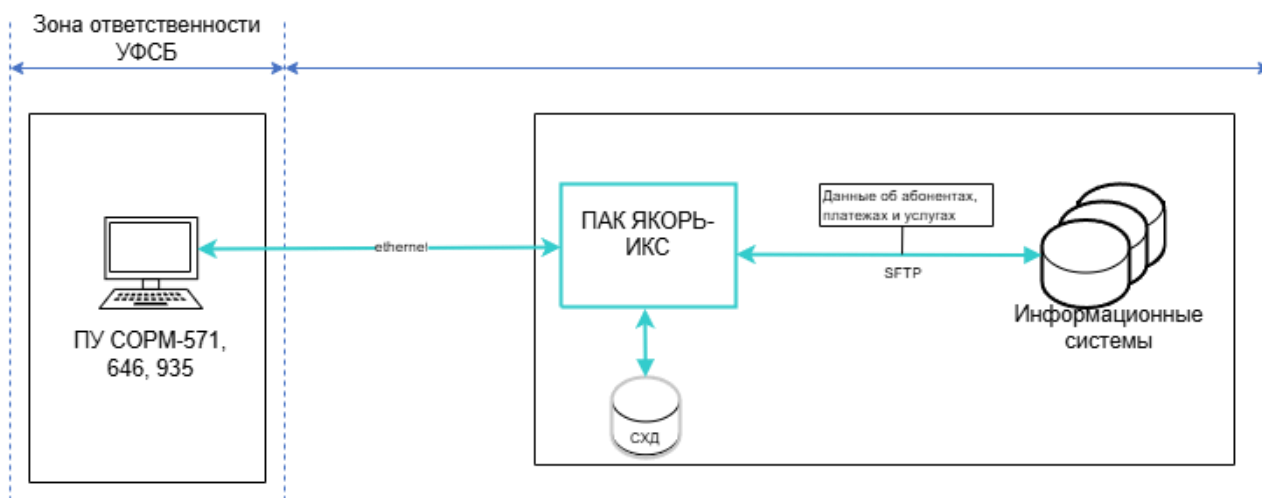


Рисунок 1 — Общая архитектура ПАК ЯКОРЬ-ИКС

## 3 Настройка параметров конфигурации

Настройки управления параметрами конфигурации сохраняются в файлах конфигурации в каталоге — /home/neos/ASN1.GW/config.

- config.properties — файл, содержащий основные настройки параметров приложения;
- host\_map — файл, содержащий настройки подключения нескольких ПУ с разным набором доверенных сертификатов;
- users.xml — файл, содержащий в себе настройки приоритизации пользователей;

В данных файлах представлены стандартные для любой конфигурации секции.

### 3.1 Условные обозначения

При описании конфигурационных файлов используются общепринятые типы данных и набор условных обозначений.

Определения, используемые при описании конфигурации:

#### Условные обозначения

Графа «Значимость параметра/перезапуск» в таблицах конфигурации содержит буквенные коды.

В таблице ниже описаны характеристики параметров, для которых применяются данные обозначения.

Таблица 3 — Буквенные коды

Тип	Описание
O	Optional. Опциональный параметр. Может отсутствовать в конфигурации, в таком случае используется значение по умолчанию.
M	Mandatory. Обязательный параметр. Его отсутствие не позволяет запустить систему, а после перезагрузки конфигурации отображается сообщение об ошибке.
P	Permanent. Параметр не переопределяется динамически, поскольку используется при запуске системы.
R	Reloadable. Параметр, значение которого можно переопределить без перезагрузки.
U	Unique. Параметр, имеющий уникальное в данном контексте значение.
X	Параметр зарезервирован. Использование запрещено.

#### Типы данных

В ходе взаимодействия с сервисом происходит обмен данными определенных типов.

В таблице ниже описаны типы данных, которые применяются во время работы с сервисом.

Таблица 4 — Используемые обозначения для типов данных

Тип	Описание
bool	Логический тип. Используется для задания флага. Принимает только значения 0 или 1, false или true соответственно.
CA (ComponentAdress)	Адрес компоненты. Строка, представляющая собой путь к программной компоненте, состоящий из перечисления узлов дерева компонент, разделенных точкой

Тип	Описание
datetime	Тип для задания даты и времени. Используемые сокращения: YY/YYYY — год, записанный двумя/четырьмя цифрами соответственно; MM — месяц, записанный двумя цифрами; DD — день, записанный двумя цифрами; hh — часы, записанные двумя цифрами; mm — минуты, записанные двумя цифрами; ss — секунды, записанные двумя цифрами; mss — миллисекунды, записанные тремя цифрами. Время задается в формате 24-часового дня.
digit_string	Строковый идентификатор, может содержать только цифры.
enum	Целое знаковое число, элемент перечисления.
float	Дробное число.
format_string	Строковый идентификатор определенного формата (должен указываться формат).
hex	Числовой тип. Задает целое число в формате шестнадцатеричного числа, записанного цифрами 0–9 и буквами A–F. Числу может предшествовать обозначение 0x.  При отсутствии обозначения определяется как строка.
int	Числовой тип. Задает целое 32-битное число, записанное цифрами 0–9 и знаком минуса "-". Диапазон: от $-2^{31}$ до $2^{31}-1$ .
list	Список, содержит несколько значений одной типа или структуры.
object	Кортеж, содержит фиксированное количество параметров различных типов.
regex	Регулярное выражение. Шаблон (маска) номера – формат параметра, задается с помощью регулярных выражений.
string	Строковый тип. Может содержать буквы латинского алфавита, цифры 0–9, спецсимволы и знаки препинания.
templ_selector	Простой шаблон.
unsigned	Целое беззнаковое число

### **Выражения**

Выражения вида: «<[имя\_параметра::]тип>» необходимо заменять на значения соответствующего типа.

Сложный параметр — подсекция конфигурационного файла. Содержит набор параметров.

**Примечание** — Сотрудникам технической поддержки заказчика допускается настраивать только параметры, непосредственно отвечающие за работу заказанной услуги по согласованию с предприятием-изготовителем оборудования.

### 3.2 Настройка основных параметров (config.properties)

Настройки управления конфигурацией сохраняются в файле config.properties.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 5 — Параметры config.properties

Параметр	Значимость параметра/ перезапуск	Описание
Секция [profiles] Настройка профилей.		
profiles	М/Р	Профили заказчика (активирует набор специфичных особенностей для работы с конкретным заказчиком). Тип — string.
Секция [application]		
application.max_hex_print_size	О/Р	Максимальное количество байт ASN структуры, которое будет преобразовано в HEX-формат. Тип — int. Значение по умолчанию: 300
Секция [operator]		
operator.name	М/Р	Имя оператора (заказчика). Значение должно содержать не менее 5 символов. Тип — string.
Секция [task] Настройки БД для сохранения задач и промежуточных результатов, а также настройки обработки задач.		
task.db.username	О/Р	Логин для подключения к БД задач. Тип — string.
task.db.password	О/Р	Пароль для подключения к БД задач. Тип — string.
task.db.driver	М/Р	Полное имя драйвера для БД задач. Тип — string.
task.db.url	М/Р	URL-адрес для подключения к БД задач. Тип — string.

Параметр	Значимость параметра/ перезапуск	Описание
task.db.max_active	O/P	Максимальное количество соединений пула базы данных. Тип — int. Значение должно быть больше 10. Значение по умолчанию: 10
task.db.min_idle	O/P	Минимальное количество «свободных» соединений, поддерживаемых пулом базы данных. Тип — int. Значение не должно быть отрицательным. Значение по умолчанию: 0
task.executors_count	O/P	Максимальное количество одновременно выполняемых задач. Тип — int. Значение должно быть больше 0. Значение по умолчанию: 2
task.max_lifetime_at_days	O/P	Максимальное время жизни задачи (количество дней), если превышен – задача удаляется. Тип — int. Значение по умолчанию: 5
Секция [secure] Настройка параметров безопасности, отвечающих за установление защищенного соединения.		

Параметр	Значимость параметра/перезапуск	Описание
secure.client_auth	O/P	<p>Параметр отвечает за авторизацию клиентов. Если настройка выключена, сертификаты не проверяются, все клиенты рассматриваются как один, возможность подключения нескольких пультов не предусмотрена и не гарантируется.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> <li>true — проверяем CA сертификаты клиентов;</li> <li>false — доверяем всем.</li> </ul> <p>Значение по умолчанию: true</p>
secure.unknown_users_allowed	O/P	<p>Право допуска к работе неизвестных пользователей (которые не заданы в конфигурационном файле users.xml).</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> <li>true — допускать;</li> <li>false — не допускать.</li> </ul> <p>Значение по умолчанию: false</p>
secure.server_key_store.jks_filepath	M/P	<p>Путь к хранилищу серверных ключей.</p> <p>Тип — string.</p>
secure.server_key_store.jks_password	M/P	<p>Пароль для хранилища ключей.</p> <p>Тип — string.</p>
secure.trusted_cert_store.jks_filepath	M/P	<p>Путь к хранилищу доверенных сертификатов.</p> <p>Тип — string.</p>
secure.trusted_cert_store.jks_password	M/P	<p>Пароль для хранилища доверенных сертификатов.</p> <p>Тип — string.</p>
secure.host_map_path	M/P	<p>Путь к файлу, в котором указано соответствие IP-адресов ПУ и имени ключа сервера (псевдоним).</p> <p>Тип — string.</p>

Параметр	Значимость параметра/перезапуск	Описание
secure.default_alias	М/Р	Уникальное имя получаемого ключа (псевдоним) для сервера, которое было сгенерировано системой. Тип — string.
Секция [transport] Настройки доступности портов для КПД1-КПД5.		
transport.channelX.port	О/Р	Настройки портов для организации КПД1-КПД5 (где X – номер соответствующего канала от 1 до 5). Если задано значение равное 0, канал не создается. Тип — int. Значение по умолчанию: 0
transport.channel_connection_timeout	О/Р	Время ожидания создания подключения по каналу (в миллисекундах). Тип — int. Значение по умолчанию: 10000
Секция [management] Настройки интерфейса компоненты мониторинга и конфигурации, который позволяет получать информацию о функционировании ASN1-GW с ПУ.		
management.zabbix.host	М/Р	Хост Zabbix агента. Тип — string. Значение по умолчанию: 127.0.0.1
management.zabbix.port	О/Р	Порт Zabbix агента. Тип — int. Значение по умолчанию: 10050
management.zabbix.connection_timeout	О/Р	Время ожидания подключения к Zabbix агенту (в миллисекундах) Тип — int. Значение по умолчанию: 2000
management.zabbix.response_timeout	О/Р	Время ожидания ответа от Zabbix агента (в миллисекундах). Тип — int. Значение по умолчанию: 5000



Параметр	Значимость параметра/перезапуск	Описание
management.file.path	O/P	Путь до файла с информацией для интерфейса компоненты мониторинга и конфигурации. Тип — string. Значение по умолчанию: /home/neos/management_info
Секция [monitoring] Настройки сервисов по мониторингу.		
monitoring.alarm.listener.max_frame	M/P	Максимальный размер приходящего сообщения с трапами (без учета заголовка размером 13 байт) в байтах. Если пришедшее сообщение превысило указанный порог, выкидывается исключение и пришедшее сообщение не обрабатывается. Тип — int. Значение по умолчанию: 1048576. <b>Примечание</b> — Минимальное значение параметра 1 Кб.
monitoring.alarm.listener.max_worker_count	M/P	Максимальное количество потоков, обрабатывающих трапы. Тип — int.
monitoring.alarm.listener.port	M/P	Порт, на который идёт отправка трапов по протоколу TCP. Тип — int.
monitoring.alarm.service.queue_size	O/P	Максимальный размер очереди, возникших ошибок. Тип — int. Значение по умолчанию: 5000
monitoring.alarm.service.termination_timeout	O/P	Время ожидания завершения внутренних запросов у компонента, обрабатывающего сигналы об авариях и несанкционированном доступе (в секундах). Тип — int. Значение по умолчанию: 30

Параметр	Значимость параметра/ перезапуск	Описание
monitoring.alarm.storage.clean_job.delay	M/P	<p>Время между очистками ошибок (в миллисекундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию: 86400000 (1 день)</p>
monitoring.alarm.storage.clean_job.alarm_outdate_period	M/P	<p>Время устаревания компонента, обрабатывающего сигналы об авариях и несанкционированном доступе, по прошествии которого он будет удалён (в миллисекундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию: 2592000000 (30 дней)</p>
monitoring.alarm.storage.clean_job.termination_timeout	O/P	<p>Время ожидания завершения внутренних запросов у компонента, обрабатывающего сигналы об авариях и несанкционированном доступе (в секундах).</p> <p>Тип — int.</p> <p>Значение по умолчанию: 30</p>
Секция [logic] Настройки КПД.		
logic.disconnect_if_error	O/P	<p>Разрывать соединение при возникновении ошибки на уровне транспорта.</p> <p>Тип — bool.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> <li>true — заполнять;</li> <li>false — не заполнять.</li> </ul> <p>Значение по умолчанию: false</p>
logic.channel5.max_data_length	O/P	<p>Количество байт в каждом блоке при передаче по КПД5.</p> <p>Тип — int.</p> <p>Значение по умолчанию: 10000</p>
Секция [pg] Настройки параметров доступа к БД по приказу Минкомсвязи № 571,646,935		

Параметр	Значимость параметра/ перезапуск	Описание
pg.url	М/Р	URL-адрес для подключения к БД PostgreSQL (справочников, абонентов, платежей, CDR). Тип — string.
pg.driver	О/Р	Полное имя класса JDBC драйвера для БД PostgreSQL. Тип — string. Значение по умолчанию: org.postgresql.Driver
pg.schema	М/Р	Наименование схемы БД PostgreSQL. Тип — string.
pg.username	М/Р	Имя пользователя для подключения к БД PostgreSQL. Тип — string.
pg.password	М/Р	Пароль для подключения к БД PostgreSQL. Тип — string.
pg.log_enabled	О/Р	Необходимость записывать запросы к БД PostgreSQL в лог-файлы. Тип — bool. Возможные значения: true — записывать; false — не записывать. Значение по умолчанию: false
pg.pool.maximum_size	О/Р	Максимальное количество активных соединений к БД PostgreSQL в наборе («пул») Соединения созданы и поддерживаются таким образом, чтобы не было необходимости пересоздавать их для новых запросов. Тип — int. Значение по умолчанию: 10

Параметр	Значимость параметра/ перезапуск	Описание
pg.pool.minimum_idle	O/P	Минимальное количество неактивных соединений к БД PostgreSQL в наборе («пул»).
		Тип — int.
		Значение по умолчанию: 1
pg.pool.connection_timeout	O/P	Время установки соединения с БД PostgreSQL.
		Тип — int.
		Значение по умолчанию: 30000
Секция [pg] Настройки параметров доступа к БД по приказу Минкомсвязи № 646,935,571		
pg.url	M/P	URL-адрес для подключения к БД PostgreSQL.
		Тип — string.
pg.driver	O/P	Полное имя класса JDBC драйвера для БД PostgreSQL.
		Тип — string.
		Значение по умолчанию: org.postgresql.Driver
pg.schema	M/P	Наименование схемы БД PostgreSQL.
		Тип — string.
pg.username	M/P	Имя пользователя для подключения к БД PostgreSQL.
		Тип — string.
pg.password	M/P	Пароль для подключения к БД PostgreSQL.
		Тип — string.
pg.log_enabled	O/P	Необходимость записывать запросы к БД PostgreSQL в лог-файлы.
		Тип — bool.
		Возможные значения:
		true — записывать;
		false — не записывать.
		Значение по умолчанию: false

Параметр	Значимость параметра/перезапуск	Описание
pg.pool.maximum_size	O/P	Максимальное количество активных соединений к БД PostgreSQL в наборе («пул»).
		Тип — int.
		Значение по умолчанию: 10
pg.pool.minimum_idle	O/P	Минимальное количество неактивных соединений к БД PostgreSQL в наборе («пул»).
		Тип — int.
		Значение по умолчанию: 1
pg.pool.connection_timeout	O/P	Время установки соединения с БД PostgreSQL.
		Тип — int.
		Значение по умолчанию: 30000
Секция [content storage] Настройки работы с Voice-storage и Network-storage		
neos.voice_storage.ip	M/P	IP-адрес для доступа по API к Voice-storage.
		Тип — string.
neos.voice_storage.port	O/P	Порт для доступа по API к Voice-storage.
		Тип — int.
neos.network_storage.ip	M/P	IP-адрес для доступа по API к Network-storage.
		Тип — string.
neos.network_storage.port	O/P	Порт для доступа по API к Network-storage.
		Тип — int.
Секция [handlers] Настройки обработчиков данных.		
neos.handlers.data_content.max_request_size	M/P	Максимальное количество записей, которые будут запрошены из хранилища контента за один запрос.
		Тип — int.

Параметр	Значимость параметра/перезапуск	Описание
neos.handlers.connections.pg_switch_id	М/Р	Код коммутатора для статических данных в хранилище (в котором хранятся данные). Тип — string.
neos.handlers.postgres_fetch_size	М/Р	Количество записей, которые будут запрошены из БД PostgreSQL (в которых хранятся данные) за один запрос. Тип — int.
neos.handlers.default_report_limit	М/Р	Определяет максимальное количество строк в ответе на запрос по умолчанию. Тип — int.
neos.handlers.telco_id	М/Р	Идентификатор оператора связи или структурного подразделения для хранилища Тип — int.
neos.handlers.telco_id	М/Р	Идентификатор оператора связи или структурного подразделения для хранилища Тип — int.
neos.task_saver.batched.batch_size	М/Р	Количество строк, которые будут скопированы из таблицы БД за один раз. Тип — int.

Пример файла конфигурации:

```
profiles=production,neos

application.max_hex_print_size=300

operator.name=Neos-Operator

task.db.username=mukhtar
task.db.password=mukhtar

task.db.driver=org.postgresql.Driver
```

```
task.db.url=jdbc:postgresql://127.0.0.1:5432/mukhtar

secure.client_auth=true
secure.unknown_users_allowed=true
secure.server_key_store.jks_filepath=./config/ISBD.jks
secure.server_key_store.jks_password=elephant
secure.trusted_cert_store.jks_filepath=./config/ISBD_trusted.jks
secure.trusted_cert_store.jks_password=elephant
secure.host_map_path=./config/host_map
secure.default_alias=server

transport.channel1.port=10222
transport.channel2.port=10223
transport.channel3.port=10224
transport.channel4.port=10225
transport.channel5.port=10226

management.zabbix.host=127.0.0.1
management.zabbix.port=10050
management.zabbix.connection_timeout=2000
management.zabbix.response_timeout=5000
management.file.path=/home/protei/management_info

monitoring.alarm.listener.max_frame_length=1048576
monitoring.alarm.listener.max_worker_count=5
monitoring.alarm.listener.port=10200
monitoring.alarm.service.alarm_push_verifier_job.delay=30000
monitoring.alarm.storage.clean_job.delay=86400000
monitoring.alarm.storage.clean_job.alarm_outdate_period=2592000000

logic.channel5.max_block_size=10000

pg.url=jdbc:postgresql://172.16.202.72:5432/pg
pg.driver=org.postgresql.Driver
pg.schema=billing
pg.username=pg_user
pg.password=pg_pass
```

```
pg.log_enabled=true

pg.pool.maximum_size=10
pg.pool.minimum_idle=5
pg.pool.connection_timeout=30000

pg.driver=org.postgresql.Driver
pg.url=jdbc:postgresql://172.16.202.152:5432
pg.schema=public
pg.username=sorm
pg.password=elephant

pg.log_enabled=true

pg.pool.maximum_size=10
pg.pool.minimum_idle=5
pg.pool.connection_timeout=30000

neos.voice_storage.ip=172.16.202.152
neos.voice_storage.port=12346

neos.network_storage.ip=127.0.0.1
neos.network_storage.port=8787

neos.handlers.data_content.max_request_size=10
neos.handlers.connections.pg_switch_id=switch#
neos.handlers.postgres_fetch_size=1000
neos.handlers.default_report_limit=999999999999
neos.handlers.telco_id=571
neos.handlers.telco_id=935

neos.task_saver.batched.batch_size=50
logic.channel5.max_data_length=10485760
```

### 3.3 Настройка нескольких ПУ (host\_map)

Настройки, определяющие подключение нескольких ПУ с разным набором доверенных сертификатов, сохраняются в файле host\_map.



В конфигурационном файле указывается соответствие IP-адресов клиентов и имени ключа (псевдоним) для сервера в формате java-properties.

Пример файла конфигурации:

```
192.168.10.1=server_key_1
192.168.10.2=server_key_2
```

### 3.4 Настройка пользователей (users.xml)

Настройки, определяющие приоритет пользователей системы, сохраняются в файле users.xml.

В таблице ниже описаны параметры конфигурационного файла.

Таблица 6 — Параметры users.xml

Параметр	Значимость параметра/ перезапуск	Описание
Корневой элемент UserConfig содержит дочерние узлы User и имеет 2 атрибута:		
defaultPriority	O/P	Приоритет для неизвестных пользователей. Тип — int. Значение по умолчанию: 5
defaultAlias	O/P	Уникальное имя получаемого ключа (псевдоним), которое было сгенерировано системой по умолчанию. Тип — string.
Элемент User — описание пользователя. Может включать в себя список разрешенных Telcold (идентификатор оператора связи или структурного подразделения). Допустимые атрибуты:		
login	M/P	Логин пользователя Тип — string.
priority	O/P	Приоритет пользователя — это приоритет выполнения задач, созданных им. Равномерное распределение с учетом приоритета. Тип — int.
host	O/P	IP-адрес пользователя. Тип — string.
alias	O/P	Уникальное имя серверного ключа (псевдоним). Тип — string.

Пример файла конфигурации:

```
<UserConfig defaultPriority="4" defaultAlias="universal_key">
```

```
<User login="support" priority="6"/>
<User login="admin" priority="1"/>
<User login="worker" priority="1" host="192.168.10.100" alias="worker_key">
  <telcold>123</telcold>
  <telcold>456</telcold>
</User>
</UserConfig>
```

## 3.5 Хранилище сертификатов и ключей

### 3.5.1 Создание хранилища ключей из сертификатов

#### 3.5.1.1 Краткое описание работы хранилища ключей из сертификатов

Результатом конфигурации будет JKS файл (server.jks), который представляет собой хранилище ключей и сертификатов. Данный файл будет содержать один серверный сертификат, собранный с помощью ключа (server.key) и сертификата (server.crt), а также сертификаты центров сертификации (CA.crt).

Механизм установления защищенного соединения:

1. ПУ подключается к ASN1.GW и запрашивает серверный сертификат для проверки того, кем он подписан.
2. ASN1.GW посылает свой сертификат и запрашивает сертификат клиента и, в случае, если клиент доверяет сертификату центра сертификации (далее — CA), которым подписан сертификат сервера, посылает ему свой сертификат для проверки.
3. ASN1.GW проверяет наличие CA, которым подписан сертификат клиента, в своем хранилище как доверенного и, в случае его наличия, считает, что аутентификация успешна и с клиентом можно работать.
4. В случае, если одна из сторон не доверяет другой, то будет отправлено сообщение SSL ALERT и совершен разрыв соединения. В общем случае есть некий SORM\_ROOT\_CA.crt, которым подписан сертификат сервера, в таком случае все подключаемые клиенты, которые доверяют SORM\_ROOT\_CA.crt, будут доверять и сертификату сервера.

#### 3.5.1.2 Импорт серверного сертификата и ключа

Для установления защищенного соединения с ASN1.GW со стороны ПУ присылаются сертификат и закрытый ключ, которые должны быть использованы на серверной стороне (пара файлов — \*.crt и \*.key).

Необходимо импортировать эти файлы (\*.crt и \*.key) в хранилище ключей. Импортирование файлов производится в хранилище типа *pkcs12* с помощью утилиты *openssl*, далее импортируется полученное хранилище *pkcs12* в *keystore*.

Пример импорта на файлах *quantum.crt* и *quantum.key*:

1. Импорт ключа и сертификата в *pkcs12* хранилище:

```
openssl pkcs12 -export -inkey quantum.key -in quantum.crt -name quantum -out quantum.p12
```

Будет предложено ввести пароль для хранилища — необходимо ввести его (далее — пароль 1).

В итоге получилось *pkcs12* хранилище (файл *quantum.p12*) с сертификатом и закрытым ключом внутри.

2. Импорт *pkcs12* хранилища в JKS *keystore*:

```
keytool -importkeystore -destkeystore server.jks -srckeystore quantum.p12 -srcaalias quantum -destalias quantum -
```

```
srcstoretype PKCS12 -deststoretype JKS
```

Файл *server.jks* создаст утилита и предложит ввести пароль для *keystore* (далее — пароль 2, этот пароль необходимо задать в *config.properties#secure.jks\_password*).

Кроме пароля на *keystore* существует также пароль на сам сертификат (доступ к нему), после импорта *pkcs12* хранилища его сертификаты имеют такой же пароль, какой был у хранилища (т.е. пароль 1). Можно либо оставить пароль таким же, тогда пароль сертификата (далее — пароль 3) будет тот же, что пароль 1, либо можно поменять пароль к сертификату.

### 3. Смена пароля к сертификату (опционально):

```
keytool -keypasswd -alias quantum -keystore server.jks
```

Будет предложено ввести пароль от *keystore* (пароль 2), и старый пароль от сертификата (пароль 1). Далее будет предложено ввести новый пароль сертификата. Таким образом в этом случае пароль сертификата (пароль 3) будет отличен от пароля 1.

#### 3.5.1.3 Импорт сертификата клиентов

Сертификаты клиента подписываются удостоверяющим центром. Подпись осуществляется с помощью сертификата этого центра (CA).

Для того, чтобы сервер аутентифицировал клиентов, сервер должен быть уверен в том, что их сертификаты настоящие. Для этого сервер проверяет — кем подписан сертификат клиента.

Чтобы сервер мог проверить это, следует занести сертификат удостоверяющего центра (CA) в хранилище.

В рамках подготовки серверного сертификата создано хранилище ключей (*keystore jks*). В него необходимо добавить сертификат удостоверяющего центра (CA). Например сертификат называется *CA.crt*.

Команда импорта сертификата CA:

```
keytool -import -trustcacerts -keystore server.jks -file CA.crt -alias CA
```

Необходимо ввести пароль от хранилища ключей (пароль 2). Хранилище ключей полностью готово к работе. Можно проверить содержимое хранилища:

```
keytool -list -v -keystore server.jks
```

#### 3.5.2 Команды утилит *keytool* и *openssl*

##### 3.5.2.1 OpenSSL

Генерация пары закрытого-открытого ключа:

```
openssl genrsa -out rootCA.key 2048
```

Создание самоподписного сертификата:

```
openssl req -x509 -new -sha256 -key rootCA.key -days 365 -out rootCA.crt
```

Создание запроса на подпись сертификата и сертификата, подписанного другим сертификатом:

```
openssl req -new -sha256 -key test.key -out test.csr
openssl x509 -req -sha256 -in test.csr -CA rootCA.crt -CAkey rootCA.key -CAcreateserial -out test.crt -days 365
```

Импорт сертификата и закрытого ключа в хранилище *pkcs12*:

```
openssl pkcs12 -export -inkey test.key -in test.crt -name test -out test_store.p12
```

Экспорт сертификата и закрытого ключа из хранилища *pkcs12*:

```
openssl pkcs12 -in keystore.p12 -nodes -nocerts -out private.key  
openssl pkcs12 -in keystore.p12 -nokeys -out public.crt
```

Экспорт открытого ключа в файл:

```
openssl rsa -in rootCA.key -outform PEM -pubout -out public.pem
```

### 3.5.2.2 Java KeyTool (JKS)

Импорт хранилища *pkcs12* в хранилище JKS:

```
keytool -importkeystore -destkeystore test_store.jks -srckeystore test_store.p12 -srcalias test -destalias test -srcstoretype PKCS12 -deststoretype JKS
```

Генерация пары ключей (открытый-закрытый) в хранилище по алгоритму RSA, создание CSR запроса на подпись сертификата, самостоятельная подпись сертификата в хранилище по запросу — создание самоподписного сертификата:

```
keytool -genkey -alias cert_alias -keypass cert_password -keystore keystore.jks -storepass store_password -keyalg RSA  
keytool -certreq -alias cert_alias -file out_request.csr -keystore keystore.jks  
keytool -gencert -alias cert_alias -infile out_request.csr -outfile out_certificate.crt -keystore keystore.jks
```

Импорт сертификата в хранилище как доверенного:

```
keytool -import -trustcacerts -keystore keystore.jks -storepass store_password -noprompt -file out_certificate.crt -alias trust_cert_alias
```

Экспорт хранилища *pkcs12* из хранилища JKS:

```
keytool -importkeystore -srckeystore keystore.jks -destkeystore keystore.p12 -srcstoretype JKS -deststoretype PKCS12 -srcalias alias -destalias alias
```

Экспорт сертификата из хранилища:

```
keytool -export -alias cert_alias -file out_certificate.crt -keystore keystore.jks
```

## 4 Техническое обслуживание

### 4.1 Доступ с использованием команды SSH

Команда SSH служит для установления удаленного сеанса связи.

Для начала работы необходима авторизация пользователя (ввод имени пользователя и пароля). Если номер порта не указан, SSH использует для связи с сервером номер порта по умолчанию (22 порт). Вместо имени сервера может использоваться его IP-адрес.

SSH работает на базе протокола TCP.

При работе в сессии удаленного управления для локализации и решения проблем в основном используются команды, приведённые в таблице ниже.

Таблица 7 — Доступные управляющие команды

Команда	Значение
cd path	переход в каталог path
cd..	переход в родительский каталог
cd /	переход в корневой каталог
ps afx	просмотр запущенных процессов
Pwd	просмотр текущего каталога
ls -l	просмотр содержимого текущего каталога
df -h	информация об использовании места на диске (flash-диск и винчестер)
date	просмотр текущей даты

### 4.2 Управление работой программного обеспечения

Управление работой системы записи переговоров осуществляется скриптами.

Основными скриптами управления работой программного обеспечения являются: start.sh, stop.sh, status.sh, clear.sh и asn1-gw.sh. Использование основных скриптов предоставляет следующие функциональные возможности:

- запуск/остановка системы;
- перезагрузка системы;
- просмотр текущего статуса работы приложения;
- вывод в log-файл информации о текущем состоянии системы.

Основные скрипты, необходимые для управления системой, находятся в корневом каталоге системы: /home/neos/ASN1.GW/bin.

Ниже приведен перечень и описание основных скриптов:

- **start.sh** — штатный запуск системы;
- **stop.sh** — останавливает систему и все утилиты;
- **status.sh** — текущий статус приложения;

- **clear.sh** — очистка лог-файлов;
- **asn1-gw.sh** — основной скрипт. Выводит лог-файл `asn-gw.log` информацию о работе программы и осуществляется запуск остальных скриптов.

## 4.3 Перезапуск системы

При изменении параметров должен производиться перезапуск системы.

**Примечание** — Путем автоматического перезапуска системы также может осуществляться автоматическая локализация аварий (сбой ПО системы).

Для перезапуска используется последовательность команд `./stop.sh;./start.sh`, которая инициализирует загрузку данных и запускает ПО.

Для остановки программного обеспечения используется команда `./stop.sh`.

## 4.4 Проверка работоспособности

В данном разделе описаны операции, доступные пользователю в веб-интерфейсе приложения.

### 4.4.1 Проверка работы ПО

Для проверки рабочего состояния программного обеспечения следует воспользоваться консолью. Откройте консоль и выполните следующие действия:

1. запустить консоль.
2. ввести команду: `ssh hostname`, где `hostname` — IP-адрес сервера, на котором установлено программное обеспечение.
3. ввести регистрационное имя (`login`) и далее пароль.
4. введите команду:

---

```
ps -afx | grep asn1-gw
```

---

Если в системе появится строка с процессом запуска, то система работает корректно.

Все ошибки возникающие в процессе работы системы фиксируются в лог-файле `asn-gw.log` с указанием компоненты, где возникла ошибка, и причиной ее возникновения.

Пример выполнения команды:

---

```
25714 pts/8  Sl   0:15 java -server -Dprotei.sorm-mukhtar-treasury.pidfile=./run/asn1-gw.pid -Dprotei.sorm-mukhtar-treasury.name=asn1-gw -Dlog4j.configurationFile=config/logging.xml -Dres/mukhtar-properties-path=./config/config.properties
```

---

В случае возникновения проблем обратитесь в службу технической поддержки ООО «НЕОС».

### 4.4.2 Проверка запуска приложения

Для проверки успешного (корректного) запуска приложения следует воспользоваться лог-файлами. Для этого с помощью консоли необходимо перейти в директорию, где хранятся лог-файлы: `/home/neos/ASN1.GW/logs`.

**Примечание** — Осуществите запуск консоли и пройдите процедуру регистрации, выполнив пункты 1-3 пункта 4.4.1 «Проверка работы».

В системе ведется лог-файл `asn-gw.log`, в котором после успешной загрузки всех компонент и модулей будет выведена соответствующая информация:

---

```
[2025-02-28 18:02:37,538][DEBUG][main][Validating transport config][ConfigValidator]
```

```
[2025-02-28 18:02:37,557][DEBUG][main][Validating secure config][ConfigValidator]
```

---

```
[2025-02-28 18:02:37,564][DEBUG][main][Validating operator config][ConfigValidator]
[2025-02-28 18:02:40,830][DEBUG][main][Initializing cleaner job][InMemoryAlarmStorage]
[2025-02-28 18:02:41,217][DEBUG][main][Starting transport server][TransportServer]
[2025-02-28 18:02:41,735][DEBUG][main][Transport server successfully started][TransportServer]
```

Если в системе не появятся строки с соответствующей информацией, то приложение работает корректно.

Если в результате выполнения данных действий указанные данные не будут выведены в лог-файле asp-gw.log, то приложение работает не корректно.

**Примечание** — Все ошибки возникающие при запуске приложения и в процессе его работы будут выведены в лог-файле с соответствующим типом сообщения (WARN или ERROR).

В данном сообщении так же будет указано детальное описание причины ее возникновения.

В случае возникновения проблем обратитесь в службу технической поддержки ООО «НЕОС».

#### 4.4.3 Проверка состояния приложения после запуска

Проверить корректность загрузки всех компонент приложения можно отследить по лог-файлу common.log.

Для проверки успешной (корректной) загрузки всех компонент приложения следует воспользоваться лог-файлами. Для этого с помощью консоли необходимо перейти в директорию, где хранятся лог-файлы: /home/neos/ASN1.GW/logs.

**Примечание** — Осуществите запуск консоли и пройдите процедуру регистрации, выполнив пункты 1-3 пункта 4.4.1 «Проверка работы».

В системе ведется лог-файл common.log, в котором после успешного запуска всех компонент и модулей будет выведена соответствующая информация:

```
[2025-02-28 18:02:36,483][INFO][ID:main][Collection property: profiles = [production,
neos]][org.protei.sorm.core.common.props.PropertyBox]

[2025-02-28 18:02:37,534][INFO][ID:main][Integer property : transport.channel1.port =
10222][org.protei.sorm.core.common.props.PropertyBox]

[2025-02-28 18:02:37,535][INFO][ID:main][Integer property : transport.channel2.port =
10223][org.protei.sorm.core.common.props.PropertyBox]

[2025-02-28 18:02:37,535][INFO][ID:main][Integer property : transport.channel3.port =
10224][org.protei.sorm.core.common.props.PropertyBox]

[2025-02-28 18:02:37,535][INFO][ID:main][Integer property : transport.channel4.port =
10225][org.protei.sorm.core.common.props.PropertyBox]

[2025-02-28 18:02:37,535][INFO][ID:main][Integer property : transport.channel5.port =
10226][org.protei.sorm.core.common.props.PropertyBox]

[2025-02-28 18:02:37,535][INFO][ID:main][Integer property : transport.channel_connection_timeout = 10000
(default)][org.protei.sorm.core.common.props.PropertyBox]
```

Если в системе не появятся строки с соответствующей информацией, то приложение работает корректно.

В случае возникновения проблем обратитесь в службу технической поддержки ООО «НЕОС».

## 5 Журналы

### 5.1 Общие принципы организации журналов

Система формирует журналы. Журнал — это именованный информационный поток, с помощью которого отслеживаются результаты работы системы.

В системе обеспечена возможность создания и хранения в течении определенного времени следующих журналов:

- `asn1-gw.log` — основной журнал работы приложения;
- `common.log` — журнал, в котором выводится информация о произошедших событиях;
- `spring.log` — журнал, в котором выводится информация о работе компоненты `spring`, входящий в состав приложения;
- `system.log` — информация о состоянии процесса работы приложения.

Журналы представлены в виде отдельных папок и находятся в корневом каталоге системы: `/home/neos/ASN1.GW/logs`.

Журналы именуются следующим образом: `<наименование журнала>-YYYY-MM-DD-<N>.log`, где `YYYY` — год, `MM` — месяц, `DD` — день месяца начала сбора информации, а `N` — порядковый номер журнала (необходимо, если за день будет сформировано несколько журналов).

Журналы генерируются автоматически с начала нового дня, за исключением первого после старта системы. Для него берется время старта.

Детальное описание данных, выводимых системой в журналах, приведено ниже.

### 5.2 Описание журнала событий (`common.log`)

В данном журнале регистрируется информация по событиям, произошедшим в системе.

Каждая запись (сообщение) выводится в одной строке и имеет фиксированное количество значений. Значения разделяются квадратными скобками и записываются в следующем порядке:

1. Дата и время в формате `YYYY-MM-DD hh-mm-ss,ms`, где `YYYY` — год, `MM` — месяц, `DD` — день, `hh` — часы, `mm` — минуты, `ss` — секунды и `ms` — миллисекунды, а в целом время начала сбора информации.
2. Логин пользователя.
3. Хост-порт пользователя, с которого устанавливается TCP-соединение.
4. Тип канала.
5. Тип события. Возможные значения:
  - `DEBUG` — диагностические сообщения;
  - `INFO` — информационные сообщения;
  - `WARN` — предупреждающие сообщения;
  - `ERROR` — сообщения об ошибках.
6. Идентификатор ASN-сообщений (`sequence`).
7. Направление события.
8. Дополнительные параметры (`"key=value,...,key=value"`). Значения разделены знаком точки с запятой «;».

#### Возможные значения

Тип канала:



- CHANNEL1 — канал управления;
- CHANNEL2 — канал передачи данных (статистические данные);
- CHANNEL3 — канал менеджмента и мониторинга;
- CHANNEL4 — канал передачи сырых (контент) данных.

Направление события:

- RX — прием;
- TX — отправка;
- INTERNAL — внутреннее событие.

Тип события:

- CREATE\_SESSION — создание сессии;
- DESTROY\_SESSION — разрушение сессии;
- UNHANDLED\_MESSAGE — необработанное сообщение;
- DECODE\_ASN\_ERROR — ошибка декодирования asn;
- INTERFACE\_ERROR — ошибка asn-интерфейса (некорректные данные);
- LOGIC\_ERROR — ошибка логики (исключение);
- CONNECT\_REQ — запрос на соединение, RX;
- CONNECT\_RESP — подтверждение соединения, TX;
- ADJUSTMENT\_REQ — запрос на активацию канала, RX;
- ADJUSTMENT\_RESP — подтверждение на активацию канала, TX;
- TRAP\_REQ — посылка уведомления, TX;
- TRAP\_RESP — подтверждение уведомления, RX;
- DISCONNECT\_REQ — запрос на разрыв соединения, RX;
- DISCONNEC\_RESP — подтверждение разрыва соединения, TX;
- SESSION\_INACTIVE\_TIMEOUT\_EXPIRY — истечение таймаута активности сессии;
- REQUEST\_RESPONSE\_TIMEOUT\_EXPIRY — истечение таймаута запрос-ответ;
- CREATE\_TASK\_REQ — Запрос на создания задания (задачи), RX;
- CREATE\_TASK\_RESP — подтверждение создания задания (задачи), TX;
- DATA\_READY\_REQ —запрос о готовности задания, RX;
- DATA\_READY\_RESP — подтверждение готовности задания, TX;
- DATA\_LOAD\_REQ — запрос на начало загрузки данных, RX;
- DATA\_LOAD\_RESP — подтверждение начала загрузки данных, TX;
- DATA\_DROP\_REQ — запрос на удаление задания, RX;
- DATA\_DROP\_RESP — подтверждение удаления задания, TX;
- DATA\_INTERRUPT\_REQ — запрос на прерывание задания (загрузки данных), RX;
- DATA\_INTERRUPT\_RESP — подтверждение прерывания задания (загрузки данных), TX;
- DATA\_ACK\_TIMEOUT\_EXPIRY — истечение таймаута на подтверждение получения данных;
- DATA\_SEND\_REQ — посылка данных отчета, RX;

- DATA\_ACK\_RESP — подтверждение получения данных отчета, TX;
- CHECK\_MODULE\_REQ — запрос на проверку (получения состояния) модулей (компонентов COPM), RX;
- CHECK\_MODULE\_RESP — подтверждение проверки (получения состояния) модулей (компонентов COPM), TX;
- GET\_MODULE\_CONFIG\_REQ — запрос на получение конфигурации модулей (компонентов COPM), RX;
- GET\_MODULE\_CONFIG\_RESP — подтверждение получения конфигурации модулей (компонентов COPM), TX;
- GET\_MODULE\_TYPES\_REQ — запрос на получение типов модулей (компонентов COPM), RX;
- GET\_MODULE\_TYPES\_RESP — подтверждение получения типов модулей (компонентов COPM), TX;
- SET\_MODULE\_CONFIG\_REQ — запрос на изменение конфигурации модулей (компонентов COPM), RX;
- SET\_MODULE\_CONFIG\_RESP — подтверждение изменения конфигурации модулей (компонентов COPM), TX;
- GET\_STRUCTURE\_REQ — запрос на получение всей конфигурации модулей (компонентов COPM), RX;
- GET\_STRUCTURE\_RESP — подтверждение получения всей конфигурации модулей (компонентов COPM), TX;
- UNFORMATTED\_DATA\_TYPES\_REQ — запрос на получение списка типов сырых данных, RX;
- UNFORMATTED\_DATA\_TYPES\_RESP — подтверждение получения списка типов сырых данных, TX;
- UNFORMATTED\_DATA\_START\_REQ — запрос на старт загрузки сырых данных, RX;
- UNFORMATTED\_DATA\_START\_RESP — подтверждение старта загрузки сырых данных, TX;
- UNFORMATTED\_DATA\_STOP\_REQ — запрос на остановку загрузки сырых данных, RX;
- UNFORMATTED\_DATA\_STOP\_RESP — подтверждение остановки загрузки сырых данных, TX;
- UNFORMATTED\_DATA\_SEND\_REQ — посылка неформатированных данных;
- UNFORMATTED\_DATA\_ACK\_RESP — подтверждение получения неформатированных данных;
- UNFORMATTED\_DATA\_ACK\_TIMEOUT\_EXPIRY — истечение таймаута на подтверждение получения неформатированных данных.

Пример содержимого log-файла:

```
[2025-02-28 11:21:53,382][INFO][ID:LogicThread-1][2025-02-28
11:21:53.345;IPU:/172.16.1.5:51263;CHANNEL1;CREATE_SESSION;;INTERNAL;;][event_journal]
[2025-02-28 11:21:53,922][INFO][ID:LogicThread-2][2025-02-28
11:21:53.922;IPU:/172.16.1.5:51264;CHANNEL2;CREATE_SESSION;;INTERNAL;;][event_journal]
[2025-02-28 11:21:54,546][INFO][ID:LogicThread-3][2025-02-28
11:21:54.546;IPU:/172.16.1.5:51265;CHANNEL3;CREATE_SESSION;;INTERNAL;;][event_journal]
[2025-02-28 11:21:54,748][INFO][ID:LogicThread-4][2025-02-28
11:21:54.748;IPU:/172.16.1.5:51266;CHANNEL4;CREATE_SESSION;;INTERNAL;;][event_journal]
[2025-02-28 11:21:55,004][INFO][ID:LogicThread-5][2025-02-28
11:21:55.004;IPU:/172.16.1.5:51267;CHANNEL5;CREATE_SESSION;;INTERNAL;;][event_journal]
```

## 5.3 Основной журнал работы приложения (asn-gw.log)

В данном журнале содержится информация о всех событиях системы, а также предоставляется полная информация о процессах обработки различных запросов, авариях и ошибках.

Каждая запись (сообщение) выводится в одной строке и имеет фиксированное количество значений. Значения разделяются квадратными скобками и записываются в следующем порядке:

1. Дата и время в формате YYYY-MM-DD hh-mm-ss,ms, где YYYY — год, MM — месяц, DD — день, hh — часы, mm — минуты, ss — секунды и ms — миллисекунды, а в целом время начала сбора информации.
2. Тип, выводимой информации. Возможные значения:
  - DEBUG – диагностические сообщения;
  - INFO – информационные сообщения;
  - WARN – предупреждающие сообщения;
  - ERROR – сообщения об ошибках
3. Наименование потока приложения, который формирует данное сообщение.
4. Сообщение, которое выводит приложение.
5. Источник сообщения.

Пример содержимого log-файла:

```
[2025-02-28      11:21:52,432][INFO][nioEventLoopGroup-3-1][CHANNEL1:Channel      active
][TransportServerHandler]
[2025-02-28      11:21:53,309][DEBUG][nioEventLoopGroup-3-1][CHANNEL1:Getting      userId      from
session][TransportServerHandler]
[2025-02-28      11:21:53,340][DEBUG][nioEventLoopGroup-3-1][createAndStartLogic:userId=IPU      type=CHANNEL1
id=1][LogicContainer]
[2025-02-28 11:21:53,924][DEBUG][LogicThread-2][sendEvent: logic of type 'CHANNEL1' of user 'IPU' was found
```

## 5.4 Журнал работы компоненты spring (spring.log)

В данном журнале содержится информация о работе компоненты spring, входящий в состав приложения.

Пример содержимого log-файла:

```
[2025-02-28      11:54:33,258][TRACE][ID:getTaskScheduler-3][Triggering beforeCommit
synchronization][org.springframework.jdbc.datasource.DataSourceTransactionManager]
[2025-02-28      11:54:33,258][TRACE][ID:getTaskScheduler-3][Triggering      beforeCompletion
synchronization][org.springframework.jdbc.datasource.DataSourceTransactionManager]
[2025-02-28      11:54:33,258][DEBUG][ID:getTaskScheduler-3][Initiating      transaction      commit
][org.springframework.jdbc.datasource.DataSourceTransactionManager]
[2025-02-28 11:54:33,258][DEBUG][ID:getTaskScheduler-3][Committing JDBC transaction on Connection
[HikariProxyConnection@826904328      wrapping
org.postgresql.jdbc.PgConnection@3f8e6751]][org.springframework.jdbc.datasource.DataSourceTransactionManager]
```

## 5.5 Системный журнал (system.log)

Системный журнал фиксирует сведения о состоянии приложения и его работе. С помощью данных, выводимых в данном потоке, пользователь может проследить за изменениями, происходящими в определенный момент времени.

Вывод значений осуществляется с помощью глобального параметра SYSTEMLOG, задаваемого в скрипте asn1-gw.sh.

Пример содержимого log-файла:

```
Вт фев 28 10:53:33 MSK 2025 send stop query to pid 25714
Вт фев 28 10:53:46 MSK 2025 started
Process has started with PID = 25714
```