



Экосистема ECSS-10. Дополнительные сервисы

Руководство по эксплуатации

Содержание

Автоматическое распознавание речи (ASR)	7
Основные понятия	7
Конфигурация сервиса	7
Установка сервиса	7
Запуск сервиса.....	7
Языковые модели.....	8
Использование сервиса.....	8
Проверка лицензии ASR	8
Использование в IVR	9
Транскрибирование разговоров	11
Основные понятия	11
Принцип работы	11
Пример работы сервиса.....	12
Система COPM	14
План подготовки комплекса для сдачи функции COPM.....	14
Настройка посредника COPM.....	16
Подключение потока E1 от ПУ COPM спецслужб к посреднику COPM.....	16
Проверка потока E1 до ПУ COPM	17
Настройка сетевого интерфейса посредника COPM.....	18
Настройка параметров посредника COPM для взаимодействия с пультом COPM.....	20
Настройки SSH	20
Настройки COPM.....	22
Настройки RADIUS	25
Настройка потоков E1	26
Сохранение конфигурации COPM посредника	26
Настройка системы COPM ECSS-10	27
Инсталляция лицензии с функцией COPM, проверка лицензии.....	27
Проверка наличия ключа eToken в системе.....	27
Настройка внутреннего протокола взаимодействия с посредником COPM	28
Настройка системы COPM.....	28
Просмотр настроек	29

Установка значений по умолчанию.....	30
Настройка маршрутизации ECSS-10 в соответствии с требованиями сотрудников спецслужб	30
Проверка соответствия формата выдачи номеров А и Б в ПУ СОРМ формату, требуемому сотрудниками спецслужб.....	32
Подсистема MultiSORM.....	35
Общая работа системы	35
Конфигурирование подсистемы MultiSORM.....	37
Настройка глобальных параметров СОРМ	37
Настройка локальных параметров СОРМ домена.....	38
Создание контекстов маршрутизации	39
Декларация бриджей.....	39
Настройка планов нумерации в доменах	40
Пример настройки подсистемы MultiSORM.....	40
Сервис "Автосекретарь"	49
Описание сервиса.....	49
Архитектура	50
Принцип работы	50
Принцип поиска распознанной строки в телефонной книге.....	51
Ответ от ASR.....	51
По какой книге производится поиск.....	51
Алгоритм поиска.....	52
Пример.....	52
Установка	52
Управление через systemd.....	53
Конфигурация.....	54
Проверка работоспособности.....	55
Настройка на стороне ECSS-10	55
Интеграции.....	59
Интеграция с CRM Битрикс-24	59
Общее описание.....	59
Активация CSTA	60
Управление интеграциями	60
1. Добавление интеграции.....	60

2. Удаление интеграции	61
3. Просмотр списка интеграций	61
4. Замена ключа интеграции.....	62
5. Изменение списка обслуживаемых номеров	63
Добавление разрешения абоненту SSW	64
Установка ecss-crm-server	65
Описание конфигурации ecss-crm-server	66
Установка приложения.....	68
Ручная установка приложения в облачной версии Битрикс-24	68
Регистрация приложения в ecss-crm-server.....	73
Настройка компании в интерфейсе ecss-crm-server	75
Активация click 2 call.....	79
Управление вызовами в интерфейсе Битрикс-24	80
Интеграция с CRM Мегаплан	81
Общее описание.....	82
Активация CSTA	82
Настройка сертификата.....	83
Управление интеграциями	84
1. Добавление интеграции.....	85
2. Удаление интеграции	85
3. Просмотр списка интеграций	85
4. Замена ключа интеграции.....	86
Добавление разрешения абоненту SSW	86
Установка виджета.....	87
Установка из магазина CRM Мегаплан.....	87
Разрешение кросс-доменного запроса для локальных инсталляций	91
Интеграция с сервисом Skype for business (S4B).....	92
Общее описание.....	93
Пример настройки.....	93
Настройка клиента.....	93
Настройка сервиса на ECSS-10	96
Настройка телефонного аппарата.....	99
Интеграции Call-центра	99
Интеграция с Redmine	100

Интеграция с КИС(корпоративной информационной системой)	102
Интеграция с СГС (системой голосового самообслуживания)	105
Интеграция с Elph	106
Установка необходимых для Elph пакетов.....	107
Установка и настройка ecss-elph-api.....	108
Установка чат-сервера.....	113
Настройки на сервере	115
Настройка адресной книги	115
Включение CSTA-API	115
Настройка на домене	116
Настройки медиасервера для работы с WebRTC и активация видеоконференций	117
Настройки для пользователей	119
Пароль для входа.....	119
Тип терминала — smart.....	121
Услуги	121
Включение core_forking.....	121
Настройка чат-сервера при установке из docker-контейнера	123
Конфигурация.....	123
Сетевые окончания.....	125
Настройка аватаров.....	125
Хранение аватаров	126
Изменение адреса аватара в mysql.....	126
Лицензирование Elph	126
Ограничения	127
Команды для работы с лицензионными ограничениями Elph.....	127
Реакция системы при превышении лицензионного лимита.....	130

В экосистему ECSS-10 входят следующие сервисы:

Информацию о других сервисах, с которыми интегрируется Softswitch, можно посмотреть, перейдя на соответствующие страницы:

- [Селекторное совещание \(Teleconference\)](#)
- [Call-центр](#)
- [Сервис "Автообзвон"](#)
- [Приложение IP Phone Eltex](#)

Автоматическое распознавание речи (ASR)

 Услуга доступна, начиная с версии ПО 3.11.2

- [Основные понятия](#)
- [Конфигурация сервиса](#)
 - [Установка сервиса](#)
 - [Запуск сервиса](#)
 - [Языковые модели](#)
- [Использование сервиса](#)
 - [Проверка лицензии ASR](#)
 - [Использование в IVR](#)

Основные понятия

Для более гибкого взаимодействия абонента АТС и IVR сценария, были добавлены функции по управлению голосом.

Конфигурация сервиса

Установка сервиса

Пакет ASR устанавливается из основного репозитория ECSS-10 с использованием следующей команды:

```
sudo apt install ecss-asr
```

В ходе установки будет предложено ответить на несколько [вопросов](#). По умолчанию предлагаются оптимальные настройки. Если по каким-либо причинам данные настройки не подходят, вводятся нужные. Всегда можно их изменить, сделав реконфигурацию пакета:

```
sudo dpkg-reconfigure ecss-asr
```

Запуск сервиса

Для запуска ASR требуется запустить сервис *ecss-asr.service*:

```
sudo systemctl start ecss-asr
```

А так же убедиться, что сервис *systemd-resolved.service* запущен:

```
sudo systemctl status systemd-resolved
• systemd-resolved.service - Network Name Resolution
  Loaded: loaded (/lib/systemd/system/systemd-resolved.service; enabled;
vendor preset: enabled)
  Drop-In: /lib/systemd/system/systemd-resolved.service.d
           └─resolvconf.conf
  Active: active (running) since Вт 2025-04-23 10:09:05 +07; 30 days ago
  Docs: man:systemd-resolved.service(8)
```

Если сервис *systemd-resolved.service* находится в состоянии **Active: inactive**, то сервис следует запустить следующей командой:

```
sudo systemctl start systemd-resolved
```

Языковые модели

Сервис ASR использует специальные модели, описывающие слова звуками. Основная модель хранится по пути:

```
/usr/lib/ecss/ecss-asr/model/ru.dic
```

В этом файле находятся все слова доступные для распознавания речи. При этом, во время работы, для повышения эффективности и точности, ASR использует только часть слов из модели. Эти слова хранятся в файле грамматики языка JSFG:

```
/etc/ecss/ecss-asr/ivr.jsfg
```

Подробнее с структурой файла грамматики можно ознакомиться в официальной [документации](#).

Использование сервиса

Проверка лицензии ASR

Для использования сервиса требуется специальная лицензия. При доступной лицензии, в подробной информации по лицензии будет следующая строка:

```
cluster/storage/ds1/licence/show-licence --details 1
```

```
...
```

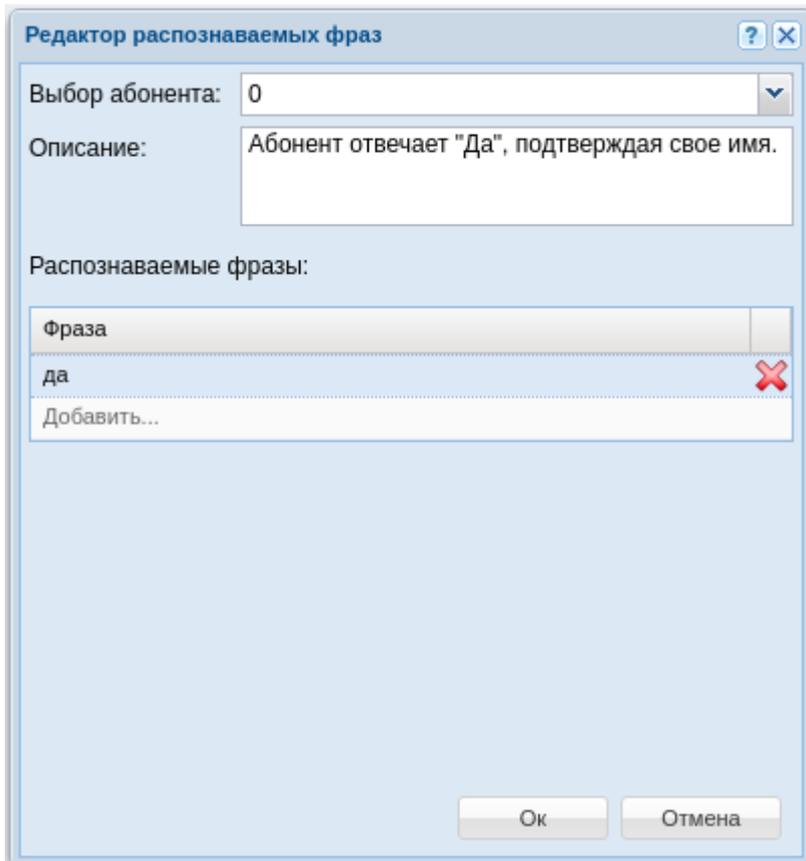
```
Automatic Speech Recognition (ASR) subsystem enabled true
```

```
...
```

Использование в IVR

Сервис ASR предлагается использовать в IVR редакторе. Если сервис ASR включен и активирован, то в таблице *"Список пунктов IVR меню, выбираемых абонентов"* для блока *"Проигрывание интерактивного меню"* в IVR редакторе появится новый столбец *"Распознаваемые фразы"*.

При двойном клике по полю из столбца *"Распознаваемые фразы"* раскрывается модальное окно *"Редактор распознаваемых фраз."*



- **Выбор абонента** — цифра, по нажатию на которую, абонент может продублировать ответ
- **Описание** — текстовое описание данного варианта выбора
- **Распознаваемые фразы** — слово или фраза, которые будут детектироваться в прямой речи абонента. Если слово или фраза будут найдены — скрипт продолжит выполняться по данной ветви.
- **Распознаваемые фразы** — слово или фраза, которые будут детектироваться в прямой речи абонента. Если слово или фраза будут найдены — скрипт продолжит выполняться по данной ветви.

ASR и настройки MSR

Иногда для корректной работы ASR необходимо будет устранить "ложные" срабатывания на посторонние звуки. Для этого необходимо произвести настройку параметра **silence-threshold** путем уменьшения чувствительности (увеличения значения **silence-threshold** например до уровня **-20 дБ**). Прочитать об этом можно в разделе [Настройка конфигурационного файла MSR](#)

Транскрибирование разговоров

 Услуга доступна начиная с версии ПО 3.14.3

- [Основные понятия](#)
- [Принцип работы](#)
- [Пример работы сервиса](#)

Основные понятия

На ECSS-10 имеется возможность транскрибирования записанных разговоров (распознавание текста из голоса). Для этого сервису транскрибирования требуется передать аудио файл в формате WAV, после чего он выдаст результат.

Принцип работы

Для распознавания текста из файла, требуется передать на вход сервиса файл в формате WAV. Существует несколько способов распознавания текста. Описание приведено ниже.

Команда в bash:

ASR_IP:PORT/trans [OPTIONS] – возвращает список распознанных слов с временными метками

Синтаксис:

OPTIONS = { -T[file] }

- -T – файл, передаваемый сервису.

Команда в bash:

ASR_IP:PORT/trans/vtt – возвращает субтитры в формате WEBVTT (расширение .vtt)

Синтаксис:

OPTIONS = { -H[Header] | -T[file] }

- -H – метки добавляемые в выводе утилиты, после транскрибирования;
- -T – файл, передаваемый сервису.

Команда в bash:

ASR_IP:PORT/trans/mono – список распознанных слов с временными метками только для первого канала, игнорируя остальные

Синтаксис:

OPTIONS = { -T[file] }

- -T — файл, передаваемый сервису.

Команда в bash:

ASR_IP:PORT/trans/mono/vtt – распознает только первый канал и конвертирует в субтитры

Синтаксис:

OPTIONS = { -H[Header] | -T[file]}

- -H — метки добавляемые в выводе утилиты, после транскрибирования;
- -T — файл, передаваемый сервису.

Пример работы сервиса

```
$ curl localhost:9000/trans -T test.wav
```

Ответ:

```
[Абонент: А]
decoder-test 1 0.99 0.27 да 0.78
decoder-test 1 2.73 0.21 да 1.00
decoder-test 1 2.94 0.36 да 1.00
decoder-test 1 5.94 0.33 какими 0.93
decoder-test 1 6.27 0.45 деньгами 0.94
```

```
[Абонент: Б]
decoder-test 1 1.65 0.45 алло 0.77
decoder-test 1 3.48 0.15 ну 0.47
decoder-test 1 3.63 0.18 как 0.47
decoder-test 1 3.81 0.15 там 0.47
decoder-test 1 3.96 0.06 с 0.46
decoder-test 1 4.02 0.54 деньгами 1.00
```

```
$ curl localhost:9000/trans/vtt -H "Calling: Алиса" -H "Called: Боб" -T test.wav
```

Ответ:

```
WEBVTT

1
00:00:00.990 --> 00:00:01.260
<v> Алиса>
да
</v>

2
00:00:01.650 --> 00:00:02.100
<v> Боб>
алло
</v>
```

3

00:00:02.730 --> 00:00:03.300

<v> Алиса>

да да

</v>

4

00:00:03.480 --> 00:00:04.560

<v> Боб>

ну как там с деньгами

</v>

5

00:00:05.940 --> 00:00:6.720

<v> Алиса>

какими деньгами

</v>

Примечание: (<v></v>) - voice tag

Система СОРМ

- План подготовки комплекса для сдачи функции СОРМ
- Настройка посредника СОРМ
 - Подключение потока Е1 от ПУ СОРМ спецслужб к посреднику СОРМ
 - Проверка потока Е1 до ПУ СОРМ
 - Настройка сетевого интерфейса посредника СОРМ
 - Настройка параметров посредника СОРМ для взаимодействия с пультом СОРМ
 - Настройки SSH
 - Настройки СОРМ
 - Настройки RADIUS
 - Настройка потоков Е1
 - Сохранение конфигурации СОРМ посредника
- Настройка системы СОРМ ECSS-10
 - Инсталляция лицензии с функцией СОРМ, проверка лицензии
 - Проверка наличия ключа eToken в системе
 - Настройка внутреннего протокола взаимодействия с посредником СОРМ
 - Настройка системы СОРМ
 - Просмотр настроек
 - Установка значений по умолчанию
 - Настройка маршрутизации ECSS-10 в соответствии с требованиями сотрудников спецслужб
 - Проверка соответствия формата выдачи номеров А и Б в ПУ СОРМ формату, требуемому сотрудниками спецслужб
- Подсистема MultiSORM
 - Общая работа системы
 - Конфигурирование подсистемы MultiSORM
 - Настройка глобальных параметров СОРМ
 - Настройка локальных параметров СОРМ домена
 - Создание контекстов маршрутизации
 - Декларация бриджей
 - Настройка планов нумерации в доменах
 - Пример настройки подсистемы MultiSORM

План подготовки комплекса для сдачи функции СОРМ

В комплексе ECSS-10 заложены возможности для выполнения требований к системе технических средств по обеспечению функций оперативно-розыскных мероприятий на электронных АТС, утвержденные приказом Госкомсвязи России от 20.04.1999 № 70 и приказом Минкомсвязи России №268 от 19.11.2012.

Подключение ECSS-10 к ПУ СОРМ спецслужб осуществляется при помощи посредника СОРМ, реализованного на базе шлюза SMG1016M, по соединительным линиям (СЛ). На рисунке 1 приведена схема подключения.



Рисунок 1 – Схема подключения ECSS-10 к ПУ COPM

- ❗ В COPM посреднике реализован основной вариант подключения к ПУ COPM (по потоку E1) и дополнительный вариант подключения (по TCP/IP).

COPM посредник позволяет организовать от 1 до 4 E1 до ПУ COPM (количество потоков E1 определяется абонентской емкостью станции, типом узла и требованиями сотрудников спецслужб).

Один поток E1 COPM содержит 28 разговорных каналов (КСЛ – контрольно соединительных линий) для прослушивания контролируемых абонентов.

- ❗ При совмещенном контроле в разговорный канал потока COPM замешивается звуковой трафик от абонентов А и Б. Смешивание звуковых потоков происходит при помощи трехсторонней конференции на VoIP-субмодуле. Один субмодуль VoIP поддерживает 27 трехсторонних конференций. Таким образом, для обеспечения возможности перехвата одновременно по всем каналам потока E1 необходимо, чтобы в сорм посреднике было установлено не менее 2 субмодулей VoIP (SM-VP-M300).

К одной виртуальной АТС комплекса ECSS-10 можно подключить несколько COPM посредников, каждый из которых будет независимо подключен к отдельному пульту COPM. Каждый из этих COPM посредников будет передавать на контролирующий его пульт данные только о тех абонентах, которые поставлены на контроль с этого пульта.

План подготовки комплекса оборудования ECSS-10 для сдачи функции (протокола) COPM сотрудникам спецслужб:

1. настройка посредника COPM;
2. настройка системы COPM ECSS-10;
3. тестирование соответствия формата выдачи номеров в ПУ COPM с требуемым сотрудниками спецслужб.

Настройка посредника COPM включает в себя:

1. подключение потока E1 от ПУ COPM спецслужб к посреднику COPM (SMG1016M);
2. настройка параметров посредника для взаимодействия с пультом COPM и системой COPM ECSS-10.

Настройка системы COPM ECSS-10 включает в себя три этапа:

1. инсталляция лицензии с функцией COPM, проверка лицензии;
2. настройка внутреннего протокола взаимодействия с посредником COPM;
3. настройка маршрутизации ECSS-10 в соответствии с требованиями сотрудников спецслужб.

Настройка посредника COPM

Порядок подготовки посредника COPM для взаимодействия с ПУ COPM спецслужб и системы ECSS-10:

1. подключение потока E1;
2. проверка потока E1;
3. настройка сетевого интерфейса;
4. настройка программного обеспечения посредника COPM.

Подключение потока E1 от ПУ COPM спецслужб к посреднику COPM

Для SMG-1016M

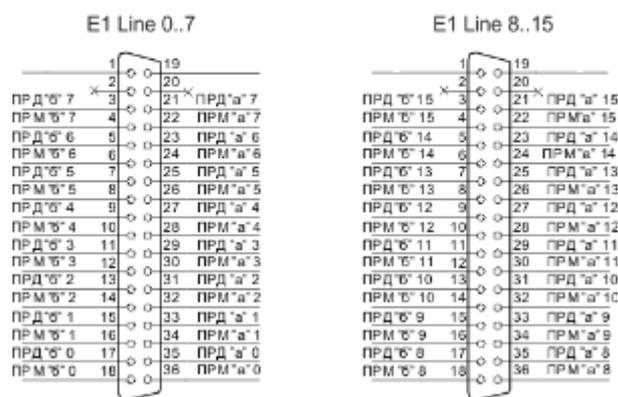


Рисунок 2 – Схема распайки разъема потока E1 для устройства SMG-1016M

- Контакты ПРМ предназначены для приема сигнала из канала в устройство.
- Контакты ПРД предназначены для передачи сигнала из устройства в канал.

Для SMG-2016

Назначение контактов разъемов RJ-48 для подключения потоков E1 E1 Line 0..15 соответствует спецификации ISO/IEC 10173 и приведено в таблице 1.

Таблица 1 – Назначение контактов разъемов RJ-48 для подключения потоков

№ контакта (Pin)	Назначение
1	RCV from network (tip)
2	RCV from network (ring)
3	RCV shield
4	XMT tip
5	XMT ring

№ контакта (Pin)	Назначение
6	XMT shield
7	Не используется
8	Не используется

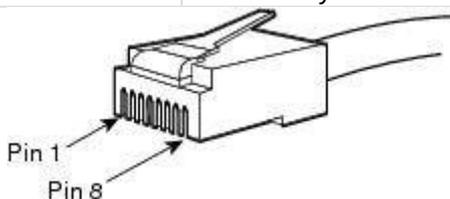


Рисунок 3 — Схема нумерации контактов разъема RJ-48 потока E1 для устройства SMG-2016

Проверка потока E1 до ПУ СОПМ

Для просмотра состояния потоков E1 необходимо:

- подключиться к устройству через web-интерфейс (по умолчанию IP-адрес: 192.168.1.2, login: admin, password: rootpasswd) и открыть вкладку **Мониторинг-Мониторинг потоков E1**;

В разделе отображается информация об установленных чипах на submodule M4E1, а также мониторинг и статистика потоков E1.

Мониторинг потоков E1

Информация о submodule M4E1

№	Name	ID
0	QFALC_v3.1	0x20
1	QFALC_v3.1	0x20
2	QFALC_v3.1	0x20
3	QFALC_v3.1	0x20

Номер потока	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Состояние	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Состояние D канала	WORK															
Состояние D канала	no	up														
Время сбора статистики (сек)	230379	230379	230379	230379	230379	230379	230379	230379	230379	230379	230379	230379	230379	230379	230379	230379
Положительных сбоев	19060	19062	19069	19075	19074	19059	19070	19073	19070	19355	19062	19074	19069	19055	19073	19063
Отрицательных сбоев	2	4	5	4	3	5	4	4	1	6	2	3	1	8	2	2
Принято байт	0	525912	589155	543275	518186	535101	608291	534877	561040	587885	520803	564062	531953	602380	560733	571671
Передано байт	0	1454788	2491339	1788713	1661224	1874488	2690610	1630392	2098463	2317929	1569292	2036677	1811242	1668811	1959113	1941795
Коротких пакетов	0	63652	180355	106104	116786	137927	212125	74104	137916	137971	95502	127326	126406	74269	148538	137916
Больших пакетов	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	1
Переполнений	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ошибок CRC	0	555	565	569	537	532	655	562	594	685	523	627	629	581	588	590
Сбоев передачи	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Code violation counter	1	3	3	0	2	2	2	2	190	179	178	278	20	26	3	3
CRC Error Counter / PRBS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit error rate	5	3	5	4	2	7	2	4	30	72	31	44	2	3	5	6
Выделить	<input type="checkbox"/>															

Возможные статусы потока E1 и расшифровка значений:

- WORK – поток в работе;
- LOS – потеря сигнала;
- OFF – поток физически выключен;
- ABSENT – не установлен submodule;

- AIS – сигнал индикации аварийного состояния (сигнал, содержащий все единицы);
- LOMF – сигнал индикации аварийного состояния сверхцикла;
- RAI – индикация удаленной аварии.

Для корректной работы COPM посредника необходимо, чтобы на потоках, подключенных к ПУ COPM, было состояние WORK.

Настройка сетевого интерфейса посредника COPM

Для настройки сетевого интерфейса необходимо:

- подключиться к устройству через web-интерфейс (по умолчанию IP-адрес: 192.168.1.2, login: admin, password:rootpasswd) и открыть вкладку **Настройки TCP/IP-Сетевые интерфейсы**;

На устройстве есть возможность сконфигурировать 1 основной сетевой интерфейс eth0 и до 9-ти дополнительных интерфейсов. Интерфейсами могут быть интерфейсы VLAN, alias основного интерфейса eth0 либо alias интерфейса VLAN.

Сетевые интерфейсы								
№	Имя интерфейса	Имя сети	IP адрес	Маска сети	DHCP	Сервисы управления	Сервисы телефонии	Профиль firewall
0	eth0	eth0	-	-	+	Web Telnet SSH SNMP	SIP RTP H323 Radius	Не выбран

Alias – это дополнительный сетевой интерфейс, который создается на базе существующего основного интерфейса eth0, либо на базе существующего VLAN-интерфейса.

Для создания, редактирования и удаления правил сетевых интерфейсов используются кнопки:

- «Добавить»;
- «Редактировать»;
- «Удалить».

Сетевые интерфейсы

Сетевой интерфейс 0	
Имя сети	<input type="text" value="eth0"/>
Профиль firewall	Не выбран
Тип	<input type="text" value="Untagged"/>
Использовать DHCP	<input type="checkbox"/>
IP адрес	<input type="text" value="192.168.18.226"/>
Маска сети	<input type="text" value="255.255.255.0"/>
Broadcast	<input type="text" value="192.168.1.255"/>
Шлюз	<input type="text" value="192.168.18.1"/>
Получить DNS автоматически	<input type="checkbox"/>
Получить NTP автоматически	<input type="checkbox"/>
Сервисы	
Управление через Web	<input checked="" type="checkbox"/>
Управление по Telnet	<input checked="" type="checkbox"/>
Управление по SSH	<input checked="" type="checkbox"/>
Использовать SNMP	<input checked="" type="checkbox"/>
Сигнализация SIP	<input checked="" type="checkbox"/>
Передавать RTP	<input checked="" type="checkbox"/>
Сигнализация H.323	<input checked="" type="checkbox"/>
Использовать RADIUS	<input checked="" type="checkbox"/>

Для COPM посредника в интерфейсе как минимум необходимо настроить следующие параметры:

- IPADDR — IP-адрес COPM-посредника;
- NETMASK — сетевая маска;
- GATEWAY — адрес шлюза;
- Флаги управления: Web, SSH, SNMP, RADIUS.

Настройка параметров посредника COPM для взаимодействия с пультом COPM

Для настройки параметров посредника COPM необходимо:

- подключиться к устройству через web-интерфейс и открыть вкладку **COPM посредник**;

В разделе конфигурируются настройки COPM посредника, являющегося посредником для взаимодействия по протоколу COPM между пультом COPM и программным коммутатором ECSS-10. COPM посредник принимает команды от пульта и транслирует их в сторону программного коммутатора ECSS-10, а также транслирует сообщения COPM в обратную сторону. Взаимодействие с пультом осуществляется по протоколу COPM, а взаимодействие с программным коммутатором по протоколам SSH и RADIUS.

Настройки SSH

COPM посредник

SSH
COPM
RADIUS
Потоки E1

Настройки SSH	
IP-адрес сервера	<input type="text" value="192.168.118.52"/>
Резервный IP-адрес	<input type="text" value="0.0.0.0"/>
Порт	<input type="text" value="7777"/>
Имя пользователя	<input type="text" value="admin"/>
Пароль	<input type="text" value="sorm_password"/>

Применить
Отменить

Для взаимодействия COPM посредника с программным коммутатором необходимо настроить следующие параметры:

- **IP-адрес сервера** — IP-адрес программного коммутатора ECSS-10 для взаимодействия по протоколу SSH;

- **Резервный IP-адрес** — IP-адрес резервного программного коммутатора ECSS-10 для взаимодействия по протоколу SSH;
- **Порт** — транспортный порт протокола TCP, открытый на хосте программного коммутатора ECSS-10 для взаимодействия с СОПМ посредником по протоколу SSH. Для настройки работы обычной подсистемы СОПМ параметру необходимо указать значение, настроенное в конфигурации программного коммутатора (**по умолчанию 7777**);
- **Имя пользователя** — имя пользователя для взаимодействия с программным коммутатором по протоколу SSH. В качестве значения параметра указывается **имя виртуальной АТС (домена)**, которая СОПМируется;
- **Пароль** — **пароль пользователя**, взаимодействующего с программным коммутатором по протоколу SSH. Здесь указывается пароль, настроенный для подсистемы СОПМ в соответствующей виртуальной АТС программного коммутатора ECSS-10.

Настройки COPM

SSH	COPM	RADIUS	Потоки E1
-----	------	--------	-----------

Настройки COPM посредника	
Версия ПО	3.5.0
Таймаут, с	0
Передача цифры 0	Передавать цифру 0
Спецификация протокола	Приказ 268 (ЭАТС)
Тип подключенной станции (АТС)	Оконечно-транзитная
Тип линка	X25
Сетевой интерфейс для RTP	eth0 (eth0 192.168.118.203)
Расширенные коды ошибок	<input type="checkbox"/>
TCP порт 1 для передачи данных	1001
TCP порт 2 для передачи данных	1002
Добавление цифры 8 в начало CdPN	<input type="checkbox"/>
Замена кода выхода на МГ с 8 на 7	<input type="checkbox"/>
Замена кода выхода на МГ с 88 на 7	<input type="checkbox"/>
Добавление цифр 810 в начало CdPN	<input type="checkbox"/>

Для взаимодействия COPM посредника с программным коммутатором необходимо настроить следующие параметры:

- **Версия ПО** — версия программного обеспечения программного коммутатора ECSS-10. Для СОРМирования ECSS-10 с версией ПО 3.5.0 и старше **нужно выбрать значение 3.5.0**;
- **Спецификация протокола** — устанавливает спецификацию протокола СОРМ для взаимодействия с пультом: нужно выбрать значение **приказ 268**;
- **Тип подключенной станции (АТС)** — тип узла связи, в качестве которого выступает программный коммутатор ECSS-10. Нужно выбрать тип узла, которым на сети является программный коммутатор: **оконечный, транзитный или транзитно-оконечный**
- **Тип линка** — тип подключения каналов передачи данных КПД1 и КПД2 к пульту СОРМ. При подключении КПД по потоку Е1 используется протокол Х.25, при подключении КПД по IP каналу используется протокол TCP;
- **Сетевой интерфейс для RTP** — сетевой интерфейс, на который от программного коммутатора передается медиа трафик от СОРМируемых вызовов;
- **TCP порт 1 для передачи данных** — номер TCP порта для подключения к ПУ СОРМ канала передачи данных КПД1. Используется при «типе линка» TCP;
- **TCP порт 2 для передачи данных** — номер TCP порта для подключения к ПУ СОРМ канала передачи данных КПД2. Используется при «типе линка» TCP.

Слудующие параметры конфигурируются при необходимости более гибко настроить взаимодействие СОРМ посредника с пультом:

- **Тайм-аут, с** — тайм-аут ожидания приема команд от ПУ СОРМ, в секундах. Если установлено значение "0", то таймер отключен. Если в течение заданного периода времени не поступают команды от ПУ СОРМ, то канал считается неисправным, блокируется передача в ПУ СОРМ. Данные, которые не были отправлены в ПУ СОРМ, удаляются.

Реализовано согласно пункту 1.5 Приказа №70 Госкомсвязи России от 20.04.1999.

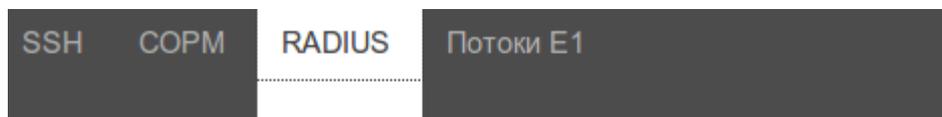
- **Передача цифры 0** — устанавливает режим передачи цифры «0», либо в виде цифры 0, либо в виде символа «А»;
- **Расширенные коды ошибок** — позволяет, при возникновении ошибок обработки команд, передавать на пульт СОРМ детальную информацию о характере возникшей ошибки;

Код	Причина
0x10	команда не принята, неверная длина или неверное количество знаков в номере
0x11	ошибка параметра(-ов)
0x12	неверный тип объекта
0x13	неверный тип номера
0x14	неверная категория
0x15	ошибка приоритета
0x16	команда «Старт СОРМ» не принята по причине «уже стартовал»
0x17	команда не принята по причине «СОРМ не запущен»
0x18	команда не принята по причине «неверный номер СОРМ»
0x25	транк-группа не задана
0x30	группа КСЛ определена другим типом
0x31	таблица объектов переполнена, мониторинг не начат
0x32	в указанной группе КСЛ нет такой КСЛ
0x33	объект уже задан

0x34	неверный номер объекта
0x36	номер уже задан
0x37	номер объекта не подходит для команды
0x38	неверный тип объекта или неверный тип номера
0x39	вывод информации уже завершен (ответ на команду «выдача таблиц»)
0x3A	КСЛ-А уже закреплена
0x3D	неверный номер объекта (при подключении к соединению, отключению от соединения)
0x3E	неверный номер группы КСЛ
0x3F	неверный номер КСЛ-А
0x40	неверный номер КСЛ-Б
0x41	неверный номер КСЛ-А и КСЛ-Б
0x44	неверное количество цифр
0x4E	параметры транка не заданы
0x53	не найден ни номер ни направление
0x54	номер абонента не локальный
0x55	признак номера неверный
0x56	неверный тип объекта для локального номера
0x57	признак локальный нельзя применить, поскольку номер не локальный
0x5F	КСЛ не задана
0x61	ДВО не заданы
0x73	ошибка выделения КСЛ

- **Добавление цифры 8 в начало CdPN** — адаптация, позволяет заменить начальную цифру 7 на цифру 8 в номере абонента В при перехвате по номеру А в вызовах с признаком «абонент станции» и «абонент ТФСООП России»;
- **Замена кода выхода на МГ с 8 на 7** — адаптация, позволяет заменить код выхода на междугородную связь с цифры 8 на цифру 7 в номере абонента В;
- **Замена кода выхода на МГ с 88 на 7** — адаптация, позволяет заменить код выхода на междугородную связь с цифр 88 на цифру 7 в номере абонента В;
- **Добавление цифр 810 в начало CdPN** — адаптация, позволяет заменить начальную цифру 7 на цифры 810 в номере абонента В при перехвате по номеру А в вызовах с признаком «абонент станции» и «абонент ТФСООП России».

Настройки RADIUS

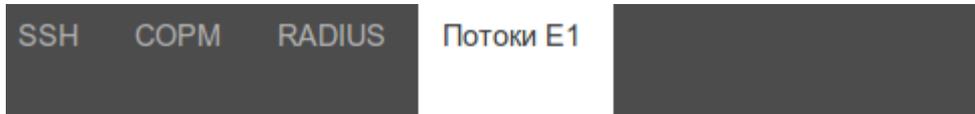


Настройки RADIUS	
IP-адрес сервера	<input type="text" value="192.168.118.203"/>
Порт	<input type="text" value="1813"/>
Секретный ключ	<input type="text" value="2testing123"/>

Для взаимодействия COPM посредника с программным коммутатором необходимо настроить следующие параметры:

- **IP-адрес сервера** — IP-адрес RADIUS сервера COPM посредника, к которому будет подключаться программный коммутатор ECSS-10 по протоколу RADIUS. В качестве IP-адреса нужно указать **адрес сетевого интерфейса COPM посредника**;
- **Порт** — транспортный UDP порт RADIUS сервера COPM посредника, к которому будет подключаться программный коммутатор ECSS-10 по протоколу RADIUS. **Обычно используется порт 1813**, на программном коммутаторе по умолчанию настраивается именно этот порт;
- **Секретный ключ** — ключ-пароль для подключения программного коммутатора ECSS-10 по протоколу RADIUS. Необходимо настроить такой же ключ, как в виртуальной АТС программного коммутатора.

Настройка потоков E1



Аппаратные настройки	
Автономный режим	<input checked="" type="checkbox"/>
Субмодуль M4E1	PCM 0
Канал	Режим
0 <input checked="" type="checkbox"/>	<input checked="" type="radio"/> DTE <input type="radio"/> DCE
1 <input type="checkbox"/>	<input type="radio"/> DTE <input type="radio"/> DCE
2 <input type="checkbox"/>	<input type="radio"/> DTE <input type="radio"/> DCE
3 <input type="checkbox"/>	<input type="radio"/> DTE <input type="radio"/> DCE

- **Автономный режим** — режим используется для устройств, в которых установлен всего один VoIP субмодуль. При установленном флаге все каналы VoIP субмодуля будут использоваться под нужды COPM посредника;

В устройствах с двумя и большим количеством субмодулей VoIP, в зависимости от количества задействованных в сторону пульта COPM потоков E1, субмодули VoIP будут распределяться между COPM посредником и SMG PBX.

Для взаимодействия с программным коммутатором ECSS-10 необходимо настроить следующие параметры:

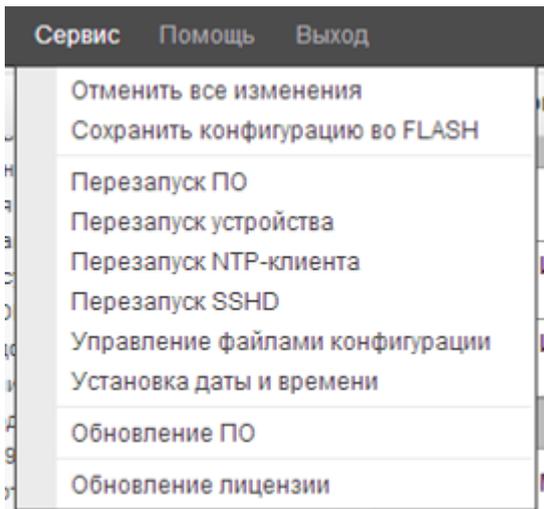
- Субмодуль M4E1 — выбор субмодуля, который будет использоваться для подключения по потокам E1 к пульту COPM;

В таблице Канал/Режим выбираются потоки E1 субмодуля и режим работы протокола X.25:

- DTE — терминальная сторона (Tx cmd=1, Rx cmd=3);
- DCE — линейная сторона (Tx cmd=3, Rx cmd=1).

Сохранение конфигурации COPM посредника

Для сохранения параметров в энергонезависимую память необходимо в меню **Сервис** выбрать пункт **Сохранить конфигурацию во Flash**.



Настройка системы COPM ECSS-10

Инсталляция лицензии с функцией COPM, проверка лицензии

Методика инсталляции лицензии описана в разделе [Начальное конфигурирование системы\(old\)](#).

Проверка наличия активации протокола COPM в лицензии ECSS-10 выполняется командой:

```
/cluster/storage/ds1/licence/current-limits
```

Значение параметра "Sorm enabled" должно быть "true":

```
Sorm enabled           true
```

Связаться с коммерческим отделом компании ЭЛТЕКС, узнать о лицензионных ограничениях и поддержке протокола COPM вашей лицензией. Если при выполнении команды выводится значение "false", необходимо:

1. При необходимости повторить инсталляцию лицензии.
2. Проверить наличие ключа eToken в системе.

Проверка наличия ключа eToken в системе

Проверку можно разделить на два этапа:

1. Проверка определения ключа eToken операционной системой Linux. В командной консоли Linux (shell) нужно выполнить команду:

```
lsusb
```

Пример результата выполнения команды:

```

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 003: ID 18e3:9101 Fitipower Integrated Technology Inc
All-in-1 Card Reader
Bus 002 Device 002: ID 0529:0620 Aladdin Knowledge Systems

```

Наличие строки **Bus 002 Device 002: ID 0529:0620 Aladdin Knowledge Systems** говорит о том, что система определила ключ eToken.

Если приведенной в примере строки нет, смотрите в разделе [Инсталляция библиотек и необходимого ПО для работы eToken](#).

2. Проверка отсутствия ошибок при попытке определить ключ eToken нодой DS. В командной консоли Linux (shell) просмотрите файл **error.log** на наличие ошибок определения ключа eToken нодой DS. Файл расположен по следующему пути: **/var/log/ecss/ds/xxxx_xx_xx_xx_xx_ds_x@ecss_x/**.

Методика решения проблем при определении ключа eToken нодой DS приведена в разделе [Инсталляция библиотек и необходимого ПО для работы eToken](#).

Настройка внутреннего протокола взаимодействия с посредником COPM

Настройка внутреннего протокола взаимодействия между посредником COPM и ECSS-10 осуществляется в командной консоли CLI.

Для настройки системы COPM ECSS-10 у пользователя должны быть права администратора домена "ecss-<DOMAIN_NAME>-domain-admin".

Команды для настройки системы COPM находятся по пути в командной консоли **/domain/<DOMAIN_NAME>/sorm**.

Настройка системы COPM

Для настройки параметров внутреннего протокола взаимодействия между посредником COPM и системой ECSS-10 используются команды:

```
/domain/<DOMAIN_NAME>/sorm/set
```

Для взаимодействия с посредником COPM необходимо произвести настройку следующих параметров:

1. **radius_servers** — IP-адрес и секретный ключ для подключения к посреднику COPM по протоколу RADIUS
Команда для настройки:
/domain/<DOMAIN_NAME>/sorm/set radius_servers add <ADDRESS> <SECRET>

где

<ADDRESS> — IP-адрес посредника COPM;

<SECRET> — секретный ключ для подключения к посреднику COPM.

При необходимости подключить несколько COPM посредников, необходимо настроить несколько подключений по протоколу RADIUS.

2. password — пароль для подключения COPM-посредника к подсистеме COPM на ECSS-10 по протоколу SSH

Команда для настройки:

```
/domain/<DOMAIN_NAME>/sorm/set password <PASSWORD>
```

где

<PASSWORD> — пароль пользователя для подключения к посреднику COPM по протоколу SSH.

 Начиная с версии 3.4.2 логин для подключения COPM посредника равен имени домена к которому он подключается.

3. enabled — включить/выключить подсистему COPMа в определенном домене

Команда для настройки:

```
/domain/<DOMAIN_NAME>/sorm/set enabled <ENABLED>
```

где

<ENABLED> :: true | false — включить/выключить подсистему COPMа в данном домене.

При необходимости можно настроить другие параметры взаимодействия, более подробная информация о командах приведена в разделе [Команды управления конфигурацией посредника COPM](#).

Просмотр настроек

Для просмотра текущих настроек внутреннего протокола используется команда:

```
/domain/<DOMAIN_NAME>/sorm/info
```

Пример выполнения команды:

```
/domain/p.city/sorm/info
```

Property	Domain	Value
adaptation	p.city	ctx_sorm
b_digits_mode	p.city	user
enabled	p.city	true
full_dial_number	p.city	false
password	p.city	123
platform_id	p.city	999
private_phone_prefix	p.city	0
radius_servers	p.city	"192.168.23.235": "2testing123"
send_optional_cf_notification	p.city	true
strict_intercity_numbers	p.city	false
strict_private_numbers	p.city	false

Описание параметров можно посмотреть тут: [Команды управления конфигурацией посредника COPM](#).

Установка значений по умолчанию

Для сброса значения определенного параметра внутреннего протокола на заводское значение используется команда:

```
/domain/<DOMAIN_NAME>/sorm/clean <PARAMETER>
```

где

<PARAMETER> — параметр, значение которого нужно установить по умолчанию.

Настройка маршрутизации ECSS-10 в соответствии с требованиями сотрудников спецслужб

Система маршрутизации вызовов ECSS-10 реализована таким образом, что на этапе маршрутизации можно определить номера вызывающего и вызываемого абонентов (номера А и Б) для дальнейшего их использования системой COPM. Функционал комплекса ECSS-10 позволяет модифицировать номера абонентов А и Б для передачи их посреднику COPM в формате, отличном от используемого при маршрутизации.

Правила отбора номеров А и Б для использования в системе COPM определяются в разделе <conditions> при создании правил контекста маршрутизации. Описание файлов контекстов маршрутизации приведено в разделе [Виртуальная АТС. Маршрутизация телефонных вызовов](#).

Номера А и Б, передаваемые на посредник COPM, определяются в разделе <actions>. Для этого используются теги:

```
<called sorm_digits="value" sorm_ni="value"/>
<calling sorm_digits="value" sorm_ni="value"/>
```

где

- called — тег для обработки номера вызываемого абонента;
- calling — тег для обработки номера вызывающего абонента;
- sorm_digits — номер абонента для СОРМ, который будет анализироваться при СОРМ-ировании вызова;
- sorm_ni — признак абонента для СОРМ, который будет анализироваться при СОРМ-ировании вызова, принимает значения:
 - private — частная сеть;
 - local — местная сеть;
 - zone — зональная сеть;
 - intercity — междугородная сеть;
 - international — международная сеть;
 - emergency — спецслужбы.

Идея модификации номеров заключается в том, чтобы при маршрутизации вызова привести формат номеров А, Б и их признаков к виду, требуемому сотрудниками спецслужб.

Пример создания правила маршрутизации (соответствует требованиям приказ 70)

Для междугородных наборов в зависимости от формата принятого на сети связи может использоваться префикс 7 или 8.

Номера абонента узла связи выдаются в федеральном формате xxxxx-xxxxx (10 знаков без префиксов).

Сотрудники спецслужб вправе потребовать постановку междугородних номеров на контроль по 11 знакам (с префиксом 7 или 8), а номеров локальных абонентов узла связи по короткому номеру.

 Вызов перехватывается в случае **полного** совпадения номера закрепленного на контроль СОРМ и одного из номеров А и Б в результате маршрутизации или номеров А и Б СОРМ (called sorm_digits, calling sorm_digits).

АОН в межстанционном взаимодействии передается в формате 10 цифр, а на контроле СОРМ будет закреплен 11-значный номер (10 цифр + префикс).

Для перехвата вызова необходимо преобразовать АОН до 11 цифр или задать номер А СОРМ 11 знаков на маршрутизации.

Пример

Контекст маршрутизации, удовлетворяющий требованиям телефонии и СОРМ:

```

<rule name="to_8">
  <conditions>
    <cdpn digits="8%"/>
    <cgpn digits="45???">
  </conditions>
  <actions>
    <called sorm_digits="8{" sorm_ni="intercity"/>
    <cdpn digits="7{">
    <calling sorm_digits="{1,2,3,4,5}" sorm_ni="private"/>
    <cgpn digits="38327{1,2,3,4,5}">
  </actions>
  <result>
    <external>
      <direction value="smg_trunk"/>
    </external>
  </result>
</rule>

```

Проверка соответствия формата выдачи номеров А и Б в ПУ СОРМ формату, требуемому сотрудниками спецслужб

Постановка на контроль – получение от ПУ СОРМ команды №5 с номером абонента, который необходимо контролировать, и параметрами для контроля этого номера. При наличии абонента в конфигурации его номер и параметры контроля заносятся во внутреннюю базу данных устройства, при этом на ПУ СОРМ будет отправлено подтверждение успешного выполнения команды. Если номер абонента отсутствует в конфигурации или какие-либо параметры в команде №5 были заданы неверно, на ПУ СОРМ будет отправлено сообщение о невыполнении данной команды.

Перехват вызова – передача ПУ СОРМ сообщений о всех этапах установления соединения абонента, поставленного на контроль (занесенного во внутреннюю базу данных устройства).

Настройка маршрутизации вызовов осуществляется в соответствии с требованиями Приказа Минкомсвязи РФ от 19.11.2012 № 268.

В Приказе Минкомсвязи РФ от 19.11.2012 № 268 описан формат номеров, в котором абоненты должны закрепляться на контроль, и требования к выдаче номеров А и Б в сообщениях о перехвате вызовов абонентов.

- 1) Абоненты станции, местные и междугородные абоненты закрепляются на контроль и передаются в сообщениях перехвата в формате 11 цифр с префиксом 7 в начале номера.
- 2) Международные абоненты закрепляются на контроль и передаются в сообщениях перехвата без префикса выхода на международную сеть (без префикса 810).
- 3) Набор абонента станции, поставленного на контроль, передается на ПУ СОРМ в немодифицированном виде (если абонент станции набрал 6 цифр, то на ПУ СОРМ будет передано 6 цифр набора).

В командной консоли CLI существует инструмент, позволяющий просмотреть формат выдачи номеров А и Б при перехвате вызова пультом управления СОРМ.

Перед сдачей узла связи сотрудникам спецслужб необходимо уточнить формат постановки на контроль и выдачи в ПУ СОПМ следующих номеров:

1. Абонентов узла связи:
 - тип объекта: абонент данной станции;
 - признак номера: абонент данной станции;
 - количество цифр: зависит от нумерации, принятой на сети связи (возможны отступления, необходимо согласовывать с сотрудниками спецслужб);
 - префикс: цифра 7, 8 или не используется (согласовывается с сотрудниками спецслужб).
2. Абонентов местной сети:
 - тип объекта: абонент сети, полный номер;
 - признак номера: местный;
 - количество цифр: зависит от нумерации, принятой на сети связи (возможны отступления, необходимо согласовывать с сотрудниками спецслужб);
 - префикс: цифра 7, 8 или не используется (согласовывается с сотрудниками спецслужб).
3. Междугородный:
 - тип объекта: абонент сети, полный номер;
 - признак номера: междугородный;
 - количество цифр: зависит от нумерации, принятой на сети связи (возможны отступления, необходимо согласовывать с сотрудниками спецслужб);
 - префикс: цифра 7 или 8 (согласовывается с сотрудниками спецслужб).
4. Международный:
 - тип объекта: абонент сети, полный номер;
 - признак номера: международный;
 - количество цифр: зависит от нумерации, принятой на сети связи (возможны отступления, необходимо согласовывать с сотрудниками спецслужб);
 - префикс: цифры 710, 810 или 10 (согласовывается с сотрудниками спецслужб).
5. Спецслужбы:
 - тип объекта: абонент сети, полный номер;
 - признак номера: спецслужбы;
 - количество цифр: зависит от нумерации, принятой на сети связи (возможны отступления, необходимо согласовывать с сотрудниками спецслужб).

❗ В Приказе Минкомсвязи РФ от 19.11.2012 N 268 признаки местный, зонавый, междугородный объединены в один признак ТфССОП России = 04Н.

После настройки маршрутизации необходимо произвести трассирование вызова для проверки формата выдаваемых номеров, используя команду:

```
/domain/<DOMAIN>/routing/trace iface=<IFACE> cdpn.<PARAM>=value [<OPT1>=<VALUE1> [ ... [<OPTN>=<VALUEN>]]]
```

где

<IFACE> — интерфейс вызывающего абонента;

<PARAM> — параметры вызываемого абонента, принимают значения: digits, incomplete, inni, nai, ni, npi;

opt1..N — опциональные параметры, набор входных данных об устанавливаемом телефонном соединении.

Основные тезисы для проверки:

1. Проверить пять основных типов контролируемых абонентов: абонент узла связи, местный, междугородный, международный, спецслужбы.
2. Если набор на междугородные, международные и на другие направления возможен через различных операторов и правила преобразования для данных направлений отличаются, необходимо проверить формат выдачи номеров при вызовах от каждого оператора.
3. Необходимо проверить входящую и исходящую связь.

⚠ Поддержка сообщений 12 ("Передача сообщений электросвязи") в рамках приказа 268, было добавлено на СОРМ посреднике начиная с версии ПО 3.18.0.3953.

Пример

Выполнить проверку набора с абонента узла связи на междугородный номер для следующих требований на станции:

1. Для телефонной связи номер локального абонента узла (абонент А) должен быть преобразован в федеральный формат, набор на междугородный номер (абонент Б) передается без изменений (10 цифр).
2. Для передачи на ПУ СОРМ номера локальных абонентов узла связи должны закрепляться на контроль по 5-ти значному номеру. Междугородные наборы должны перехватываться по 11 цифрам с добавленным к номеру префиксом 8.

Команда для проверки:

```
/domain/test.domain/routing/trace cgnp.digits=28331 cdpn.digits=89001758964
mode=enblock
Traceroute to 89001758964 from 28331(<<"s10p20@msantest.domain">>) at
04.08.2013 22:18:27, final number
Default context is {"test.domain",ctx_from_local}
...
-----
Routing result: external
A: "3832728331"(ni=private, nai=nationalNumber, inc=false, npi=isdntelephony,
apri=presentationAllowed, screening=networkProvided)
B: "9001758964"(ni=intercity, nai=nationalNumber, inc=false, inni=undefined,
npi=undefined)
Options: [{sorm_ni_a,private},
          {sorm_digits_a,"28331"},
          {sorm_ni_b,intercity},
          {sorm_digits_b,"89001758964"}]
...
```

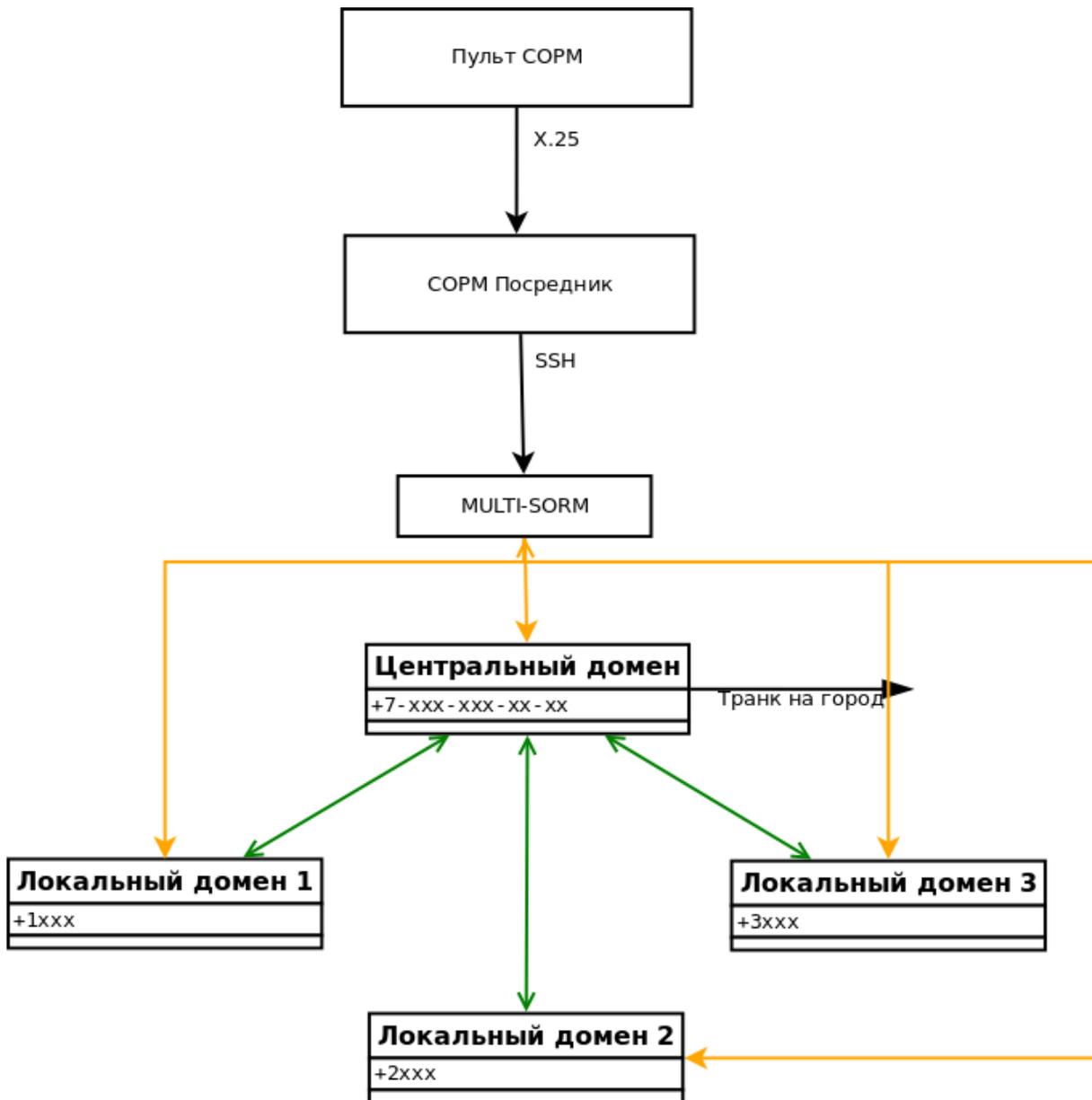
В результате преобразования номера, используемые для дальнейшей маршрутизации вызова, и номера, передаваемые в ПУ СОРМ, отличаются.

В данном наборе в ПУ СОРМ будут переданы номера: А — 28331, признак местный, Б — 89001758964, признак междугородный. Для маршрутизации вызова будут использоваться: А — 3832728331, Б — 9001758964.

Подсистема MultiSORM

Общая работа системы

Подсистема MultiSORM позволяет СОРМ-ировать несколько виртуальных АТС (доменов) как одну, что позволяет скрыть от пульта СОРМ внутреннюю структуру ECSS-10.



Бриджи выхода в центральный домен

Управляющие команды со стороны СОРМ посредника / Нотификации от объектов наблюдения

Рисунок 4 – Организация подключения пульта СОРМ к подсистеме MultiSORM

При использовании подсистемы MultiSORM СОРМ посредник подключается к центральному домену, где авторизуется по имени и паролю. При прохождении процедуры авторизации подсистема MultiSORM проверяет конфигурацию доменов подключенных к подсистеме. Если подсистема MultiSORM не смогла подключить хотя бы один из доменов по причине, что он не существует, либо имеет некорректные настройки для взаимодействия с СОРМ посредником, то авторизация будет неуспешной даже несмотря на то, что авторизационные имя и пароль будут верными. При успешной авторизации подсистема MultiSORM переходит в режим передачи команд полученных от СОРМ посредника в подключенные домены, включая

центральный. Команда транслируется сразу во все подключенные домены и если хотя бы из одного получен положительный ответ, то он передается на СОРМ посредник.

Например, от СОРМ посредника принята команда №5 «постановка на контроль». Она транслируется во все домены, подключенные к подсистеме MultiSORM и если хотя бы один домен поставил принятый в команде номер на контроль, то значит, что команда выполнена успешно. В случае, если от всех доменов получен отрицательный ответ, — на СОРМ посредник будет протранслирован отрицательный ответ.

Абоненты центрального домена ставятся на контроль по полному номеру (11 знаков начиная с семерки согласно 268 приказа) как абоненты сети (абонент сети с полным номером, междугородный), а абоненты остальных доменов, подключенных к центральному, — по локальному короткому номеру со значностью принятой в данном домене, как абоненты данной станции. Подсистема MultiSORM для своей работы требует одного ограничения — локальные номера во всех подключенных доменах должны быть уникальными, то есть номера из одного домена не должны присутствовать в других.

Маршрутизация вызовов между доменами и на внешнее городское направление осуществляется через центральный домен, а маршрутизация между абонентами домена — локально внутри домена. В результате, если вызов идет на город, абонент будет СОРМ-ироваться по полному номеру центрального домена, если вызов осуществляется внутри локального домена или между локальными доменами, то вызов будет СОРМ-ироваться по локальному короткому номеру.

Конфигурирование подсистемы MultiSORM

При конфигурировании подсистемы MultiSORM подразумеваем, что локальные и центральный домены уже созданы. В локальных доменах созданы локальные абоненты с короткой нумерацией, а в центральном домене создан SIP-транк для выхода на городское направление.

Настройка глобальных параметров СОРМ

Для подключения СОРМ посредника подсистема MultiSORM использует ssh порт, на единицу больший, чем указано в параметре `ssh_port`.

Параметр настраивается командой:

```
sorm/set ssh_port <value>
```

- `<value>` — значение порта в диапазоне 1-65535. По умолчанию значение данного параметра равно 7777, то есть по умолчанию подсистема MultiSORM будет использовать порт 7778.

 На СОРМ посреднике в параметре Port вкладки "SORM mediator — SSH" нужно настроить значение `ssh_port+1`

Список доменов, которые будут СОРМ-ироваться и авторизационные данные пользователя подсистемы MultiSORM (имя/пароль) настраиваются командой:

```
sorm/set aggregate_sorm_settings add <User> <Password> <DOMAIN_1> [<DOMAIN_2> ...  
[<DOMAIN_N>]]
```

Для автоматического добавления доменов без повторной постановки абонента на контроль и без перезапуска СОРМ существуют шаблоны доменов. Задаются они следующими командами:

- подключение всех доменов `sorm/set aggregate_sorm_settings add test3 1234 *`
- подключение всех доменов за исключением одного (например, test) `sorm/set aggregate_sorm_settings add test3 1234 * -test`

 Символ звездочка (*) может встречаться только один раз в шаблоне одного домена.

При необходимости для удаления пользователя используется команда:

```
sorm/set aggregate_sorm_settings remove <User>
```

- User — имя пользователя подсистемы MultiSORM;
- Password — пароль пользователя подсистемы MultiSORM;
- DOMAIN_1..DOMAIN_N — имя локального или центрального домена.

Настройка локальных параметров СОРМ домена

В каждом домене, используемом подсистемой MultiSORM необходимо включить функциональность СОРМ и настроить взаимодействие с СОРМ посредником.

Для включения функциональности СОРМ используется параметр `enabled`, которому необходимо установить значение `true`.

Параметр настраивается командой:

```
domain/DOMAIN_N/sorm/set enabled true
```

- DOMAIN_N — имя локального или центрального домена.

Для взаимодействия с СОРМ посредником в каждом домене добавляем RADIUS-сервер и секретный ключ для взаимодействия с ним. IP-адрес СОРМ посредника является IP-адресом RADIUS сервера.

RADIUS сервер добавляется командой:

```
domain/DOMAIN_N/sorm/set radius_servers add <IP address> <secret>
```

- DOMAIN_N — имя локального или центрального домена;
- <IP address> — IP адрес RADIUS сервера;
- <secret> — секретный ключ для доступа к RADIUS серверу.

 На СОРМ посреднике в параметрах *Server IP-address* и *Secret* вкладки "SORM mediator — RADIUS" нужно настроить соответствующие значения IP адреса RADIUS-сервера и секретного ключа.

Создание контекстов маршрутизации

Необходимо для каждого домена создать контексты маршрутизации, в которых учесть планируемую нумерацию и имена бриджей для взаимодействия между центральным и локальными доменами. Сами бриджи будут создаваться позже.

Созданные контексты импортируются командой:

```
domain/DOMAIN_N/routing/import <host> <file>
```

- <host> — имя хоста, на котором необходимо импортировать шаблон контекста маршрутизации;
- <file> — имя файла с шаблоном контекста маршрутизации, который необходимо установить.

Декларация бриджей

Для взаимодействия между центральным и локальными доменами необходимо создать бриджи. Для каждого домена создается один бридж для выхода на городское направление по полной нумерации и один бридж для взаимодействия между доменами через центральный домен по короткой нумерации.

Бридж декларируется командой:

```
/bridge/declare <name> <duplex> <strict> <capacity> <DOMAIN_A> <np_a> <iface_a>  
<tg_a> <routing_ctx_a> <DOMAIN_B> <np_b> <iface_b> <tg_b> <routing_ctx_b>
```

- <name> — уникальное имя bridge-интерфейса (строка, без пробелов) в рамках ECSS-10;
 - <duplex> — режим работы bridge:
 - true — двунаправленный;
 - false — однонаправленный;
 - <strict> — режим проверки номеров, проходящих через бридж:
 - true — проверка включена (при вызове из <DOMAIN_A> -> <DOMAIN_B> будет проверка, что номер абонента B существует в плане нумерации <NP_B> в домене B; при вызове из <DOMAIN_B> -> <DOMAIN_A> будет проверка, что номер абонента A существует в плане нумерации <NP_B> домена B);
 - false — проверка выключена;
 - <capacity> — количество одновременно активных вызовов на заданном bridge-интерфейсе, принимает значения:
 - любое положительное число;
 - unbounded — количество не ограничено;
- <DOMAIN_A> / <DOMAIN_B> — имя первой/второй виртуальной АТС, с которой устанавливает соединение bridge;
- <np_a> / <np_b> — план нумерации первой/второй виртуальной АТС, с которой устанавливает соединение bridge;
- <iface_a> / <iface_b> — имя интерфейса первой/второй виртуальной АТС, через который будет осуществляться взаимодействие с bridge-интерфейсом;
- <tg_a> / <tg_b> — идентификатор транк-группы для интерфейса первой/второй виртуальной АТС;
- <routing_ctx_a> / <routing_ctx_b> — контекст маршрутизации по умолчанию для интерфейса первой/второй виртуальной АТС.

Настройка планов нумерации в доменах

В каждом домене необходимо создать план нумерации. В центральном домене это делается для осуществления вызовов с городского направления в локальные домены, а в локальных доменах для осуществления вызовов на городское направление через центральный домен.

План нумерации добавляется командой:

```
domain/DOMAIN_N/np/declare <numplan>
```

- DOMAIN_N — имя локального или центрального домена;
- <numplan> — имя плана нумерации.

Дальше необходимо добавить полные абонентские номера в план нумерации центрального домена. Данное действие выполняется командой:

```
domain/DOMAIN_N/np/numbers/add <numplan> <numbers>
```

- DOMAIN_N — имя локального или центрального домена;
- <numplan> — имя плана нумерации;
- <numbers> — полные 11 значные абонентские номера.

В центральном домене необходимо присвоить полным абонентским номерам направления на локальные домены, в которых за ними будут закреплены локальные абоненты. Данное действие выполняется командой:

```
domain/DOMAIN_N/np/numbers/bind <numplan> <numbers> --bridge <bridge>
```

- DOMAIN_N — имя локального или центрального домена;
- <numplan> — имя плана нумерации;
- <numbers> — полные 11-значные абонентские номера;
- <bridge> — имя бриджа, через который осуществляется выход на локальный домен.

В локальных доменах необходимо настроить соответствия между полными и локальными номерами. Данное действие выполняется командой:

```
domain/DOMAIN_N/np/numbers/bind <numplan> <numbers> --alias <local_numbers>  
[<group> <IfaceRange>] [--master | --passive]
```

- DOMAIN_N — имя локального или центрального домена;
- <numplan> — имя плана нумерации;
- <number> — полные 11-значные абонентские номера;
- <local_number> — локальные абонентские номера доменов;
- <group> — группа локального абонента, опциональный параметр;
- <IfaceRange> — интерфейс локального абонента, опциональный параметр:
--master — команда определяет локальный номер "мастером", возможен только один "мастер";
--passive — команда определяет локальный номер "не мастером", установлено по умолчанию.

Пример настройки подсистемы MultiSORM

1. Декларируем домены:

```
domain/declare central_voip
domain/declare local.voip
domain/declare voip .local
```

2. Настраиваем сеть для работы протокола SIP (IP_SET voip.local.set1 создан ранее:

```
domain/local.voip/sip/network/set ip_set [voip.local.set1]
domain/voip.local/sip/network/set ip_set [voip.local.set1]
domain/central_voip/sip/network/set ip_set [voip.local.set1]
```

3. В центральном домене настраиваем SIP транк с именем trunk1 на городское направление. IP-адрес городской АТС 192.168.118.119, протокол SIP работает на порте 5060.

```
domain/central_voip/trunk/sip/declare default_routing central_voip
trunk1 voip.local.set1 static 192.168.118.119 5060 sip-proxy 5060
```

4. Создаем локальных пользователей в доменах voip.local и local.voip:

```
domain/local.voip/sip/user/declare default_routing local.voip
300001@local.voip alias-as-user qop_authentication login-as-user 300001
```

```
domain/local.voip/sip/user/declare default_routing local.voip
300000@local.voip alias-as-user qop_authentication login-as-user 300000
domain/local.voip/sip/user/declare default_routing voip.local
200100@voip.local alias-as-user qop_authentication login-as-user 200100
```

```
domain/local.voip/sip/user/declare default_routing voip.local
200119@voip.local alias-as-user qop_authentication login-as-user 200119
```

5. Декларируем пользователя СОРМ посредника с именем multi и паролем multi для работы с мультисормом.МВ подсистему включены три домена:

```
sorm/set aggregate_sorm_settings add multi multi central_voip voip.local
local.voip
```

6. Настраиваем взаимодействие с СОРМ посредником (IP адрес 192.168.118.203, секретный ключ 2testing123) для всех доменов:

```

domain/central_voip/sorm/set enabled true
domain/central_voip/sorm/set radius_servers add 192.168.118.203
2testing123
domain/local_voip/sorm/set enabled true
domain/local_voip/sorm/set radius_servers add 192.168.118.203
2testing123
domain/voip .local/sorm/set enabled true
domain/voip .local/sorm/set radius_servers add 192.168.118.203
2testing123

```

7. Создаем контексты маршрутизации с учетом планируемой нумерации («738327(0-1)????» – полные номера абонентов и «73832?????» – городские номера) и планируемых имен бриджей (c2v, c2l, c2v_inter, c2l_inter, v2c, l2c, v2c_inter, l2c_inter). Импортируем контексты маршрутизации.

Для домена central_voip
ctx_default_routing.xml

```

<?xml version="1.0" encoding="UTF-8"?>
  <context xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
  xs:noNamespaceSchemaLocation="ecss_routing.xsd" name="default_routing"
  digitmap="(*[0-9*][0-9*].#|# [0-9*][0-9*].#|*#[0-9*][0-9*].#|1xx.|
  10xx.)">
    <rule name="to_voip">
      <conditions>
        <cgpn digits="?????"/>
        <cdpn digits="2????"/>
      </conditions>
      <result>
        <external>
          <trunk value="bridge:c2v_inter"/>
        </external>
      </result>
    </rule>
    <rule name="to_local">
      <conditions>
        <cgpn digits="?????"/>
        <cdpn digits="3????"/>
      </conditions>
      <result>
        <external>
          <trunk value="bridge:c2l_inter"/>
        </external>
      </result>
    </rule>
    <rule name="to_local_city_users">
      <conditions>
        <cgpn digits="?????????"/>
        <cdpn digits="738327(0-1)????"/>
      </conditions>
      <actions>
        <called sorm_ni="local"/>
      </actions>
    </rule>
  </context>

```

```

        <calling sorm_ni="local"/>
    </actions>
    <result>
        <continue context="ctx_city"/>
    </result>
</rule>
<rule name="to_gorod">
    <conditions>
        <cgpn digits="?????????"/>
        <cdpn digits="73832?????"/>
    </conditions>
    <actions>
        <calling sorm_ni="local"/>
        <called sorm_ni="local"/>
    </actions>
    <result>
        <external>
            <trunk value="trunk1"/>
        </external>
    </result>
</rule>
</context>

```

ctx_city.xml

```

<?xml version="1.0" encoding="UTF-8"?>
    <context xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="ecss_routing.xsd" name="ctx_city"
np="city" digitmap="(*[0-9*][0-9*].#|#[0-9*][0-9*].#|*#[0-9*][0-9*].#|
1xx.|10xx.)">
        <rule name="to_local_city_users">
            <result>
                <local/>
            </result>
        </rule>
    </context>

```

```

domain/central_voip/routing/import ecss1 ctx_default_routing.xml
domain/central_voip/routing/import ecss1 ctx_city.xml

```

Для домена local.voip ctx_default_routing.xml

```

<?xml version="1.0" encoding="UTF-8"?>
    <context xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="ecss_routing.xsd" name="default_routing"
digitmap="(*[0-9*][0-9*].#|#[0-9*][0-9*].#|*#[0-9*][0-9*].#|1xx.|
10xx.)">
        <rule name="local_calls">
            <conditions>
                <cdpn digits="3?????"/>
                <cgpn digits="??????"/>
            </conditions>
        </rule>
    </context>

```

```

    </conditions>
    <actions>
      <called sorm_ni="private"/>
      <calling sorm_ni="private"/>
    </actions>
    <result>
      <local/>
    </result>
  </rule>
  <rule name="other_departments">
    <conditions>
      <cdpn digits="?????"/>
      <cgpn digits="3????"/>
    </conditions>
    <actions>
      <called sorm_ni="private"/>
      <calling sorm_ni="private"/>
    </actions>
    <result>
      <external>
        <trunk value="bridge:l2c_inter"/>
      </external>
    </result>
  </rule>
  <rule name="to_city_bridge">
    <conditions>
      <cdpn digits="73832?????"/>
      <cgpn digits="3????"/>
    </conditions>
    <actions>
      <called sorm_digits="{1,2,3,4,5,6,7,8,9,10,11}" sorm_ni="local"/
>
      <calling sorm_digits="{1,2,3,4,5,6}" sorm_ni="private"/>
    </actions>
    <result>
      <continue context="to_city"/>
    </result>
  </rule>
  <rule name="local_from_bridge">
    <conditions>
      <cdpn digits="3????"/>
      <cgpn digits="73832?????"/>
    </conditions>
    <actions>
      <called sorm_ni="private"/>
      <calling sorm_ni="local"/>
    </actions>
    <result>
      <local/>
    </result>

```

```

    </rule>
</context>

```

ctx_city.xml

```

<?xml version="1.0" encoding="UTF-8"?>
  <context xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="ecss_routing.xsd" name="to_city" np="city"
digitmap="(*[0-9*][0-9*].#|# [0-9*][0-9*].#|*#[0-9*][0-9*].#|1xx.|
10xx.)">
  <rule name="to_city_bridge">
    <result>
      <external>
        <trunk value="bridge:l2c"/>
      </external>
    </result>
  </rule>
</context>

```

ctx_from_bridge.xml

```

<?xml version="1.0" encoding="UTF-8"?>
  <context xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="ecss_routing.xsd" name="from_l2c"
np="city" digitmap="(*[0-9*][0-9*].#|# [0-9*][0-9*].#|*#[0-9*][0-9*].#|
1xx.|10xx.)">
  <rule name="local_from_bridge">
    <conditions>
      <cdpn digits="73832?????"/>
      <cgpn digits="73832?????"/>
    </conditions>
    <actions>
      <called sorm_ni="private"/>
      <calling sorm_ni="local"/>
    </actions>
    <result>
      <local/>
    </result>
  </rule>
</context>

```

```

domain/local.voip/routing/import ecss1 ctx_default_routing.xml
domain/local.voip/routing/import ecss1 ctx_city.xml
domain/local.voip/routing/import ecss1 ctx_from_bridge.xml

```

Для домена voip.local

ctx_default_routing_multi.xml

```

<?xml version="1.0" encoding="UTF-8"?>
  <context xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="ecss_routing.xsd" name="default_routing"

```

```

digitmap="(*[0-9*][0-9*].#|# [0-9*][0-9*].#|*#[0-9*][0-9*].#|1xx.|
10xx.)">
  <rule name="local_calls">
    <conditions>
      <cgpn digits="?????"/>
      <cdpn digits="2?????"/>
    </conditions>
    <actions>
      <calling sorm_ni="private"/>
      <called sorm_ni="private"/>
    </actions>
    <result>
      <local/>
    </result>
  </rule>
  <rule name="other_departments">
    <conditions>
      <cgpn digits="2?????"/>
      <cdpn digits="3?????"/>
    </conditions>
    <actions>
      <calling sorm_ni="private"/>
      <called sorm_ni="private"/>
    </actions>
    <result>
      <external>
        <trunk value="bridge:v2c_inter"/>
      </external>
    </result>
  </rule>
  <rule name="to_city_bridge">
    <conditions>
      <cgpn digits="2?????"/>
      <cdpn digits="73832?????"/>
    </conditions>
    <actions>
      <calling sorm_digits="{1,2,3,4,5,6}" sorm_ni="private"/>
      <called sorm_digits="{1,2,3,4,5,6,7,8,9,10,11}" sorm_ni="local"/
>
    </actions>
    <result>
      <continue context="to_city"/>
    </result>
  </rule>
  <rule name="local_from_bridge">
    <conditions>
      <cgpn digits="73832?????"/>
      <cdpn digits="2?????"/>
    </conditions>
    <actions>
      <called sorm_ni="private"/>
      <calling sorm_ni="local"/>
    </actions>

```

```

    <result>
      <local/>
    </result>
  </rule>
</context>

```

ctx_to_city_multi.xml

```

<?xml version="1.0" encoding="UTF-8"?>
  <context xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="ecss_routing.xsd" name="to_city" np="city"
digitmap="(*[0-9*][0-9*].#|# [0-9*][0-9*].#|*#[0-9*][0-9*].#|1xx.|
10xx.)">
    <rule name="to_city_bridge">
      <result>
        <external>
          <trunk value="bridge:v2c"/>
        </external>
      </result>
    </rule>
  </context>

```

ctx_from_br_multi.xml

```

<?xml version="1.0" encoding="UTF-8"?>
  <context xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="ecss_routing.xsd" name="from_v2c"
np="city" digitmap="(*[0-9*][0-9*].#|# [0-9*][0-9*].#|*#[0-9*][0-9*].#|
1xx.|10xx.)">
    <rule name="local_from_bridge">
      <conditions>
        <cgpn digits="73832?????"/>
        <cdpn digits="73832?????"/>
      </conditions>
      <actions>
        <called sorm_ni="private"/>
        <calling sorm_ni="local"/>
      </actions>
      <result>
        <local/>
      </result>
    </rule>
  </context>

```

```

domain/voip.local/routing/import ecss1 ctx_default_routing_multi.xml
domain/voip.local/routing/import ecss1 ctx_to_city_multi.xml
domain/voip.local/routing/import ecss1 ctx_from_br_multi.xml

```

8. Декларируем мосты:

```
bridge/declare c2v true true unbounded central_voip city bridge:c2v
tg:c2v default_routing voip.local city bridge:v2c tg:v2c from_v2c
```

```
bridge/declare c2l true true unbounded central_voip city bridge:c2l
tg:c2l default_routing local.voip city bridge:l2c tg:l2c from_l2c
```

```
bridge/declare c2l_inter true false unbounded central_voip default
bridge:c2l_inter tg:c2l default_routing local.voip default
bridge:l2c_inter tg:l2c default_routing
bridge/declare c2v_inter true false unbounded central_voip default
bridge:c2v_inter tg:c2v default_routing voip.local default
bridge:v2c_inter tg:v2c default_routing
```

9. Декларируем планы нумерации во всех доменах:

```
domain/central_voip/np/declare city
domain/local.voip/np/declare city
domain/voip.local/np/declare city
```

10. Добавляем полные абонентские номера в план нумерации центрального домена:

```
domain/central_voip/np/numbers/add city 738327{0-1}000{0-1}
```

11. Присваиваем абонентским номерам направления на домены, в которых за ними будут закреплены локальные абоненты:

```
domain/central_voip/np/numbers/bind city 7383271000{0-1} --bridge c2l
domain/central_voip/np/numbers/bind city 7383270000{0-1} --bridge c2v
```

12. Настраиваем соответствия между полными и локальными номерами:

```
domain/voip.local/np/numbers/bind city 73832700000 --alias 200100
voip.local 053eb7f6f8c37df5 --master
domain/voip.local/np/numbers/bind city 73832700001 --alias 200119
voip.local 053eb7fae721975f --master
domain/local.voip/np/numbers/bind city 73832710000 --alias 300000
local.voip f4c06b4a996e9e6e --master
domain/local.voip/np/numbers/bind city 73832710001 --alias 300001
local.voip f4c068ee409afa6c --master
```

Сервис "Автосекретарь"

- [Описание сервиса](#)
- [Архитектура](#)
- [Принцип работы](#)
- [Принцип поиска распознанной строки в телефонной книге](#)
 - [Ответ от ASR](#)
 - [По какой книге производится поиск](#)
 - [Алгоритм поиска](#)
 - [Пример](#)
- [Установка](#)
 - [Управление через systemd](#)
 - [Конфигурация](#)
 - [Проверка работоспособности](#)
- [Настройка на стороне ECSS-10](#)

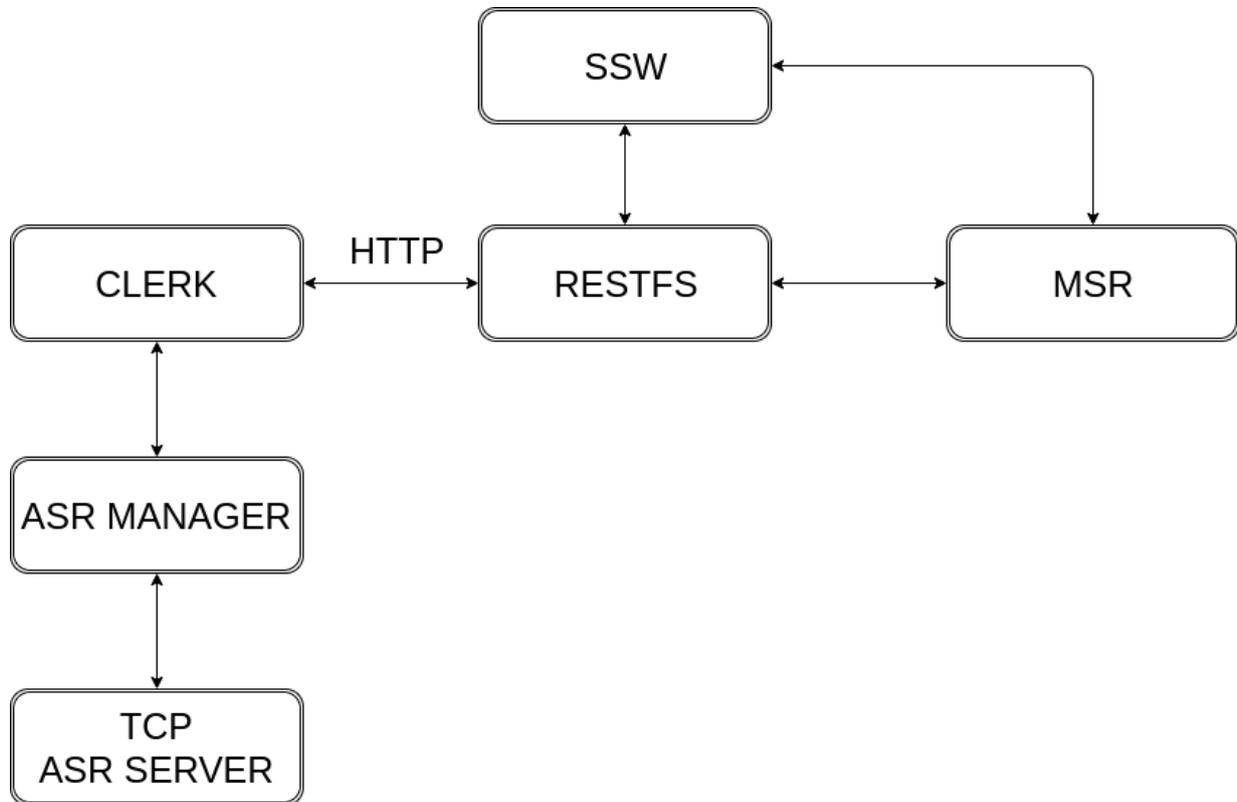
Описание сервиса

Сервис автосекретарь распознает имя и фамилию абонента, которому вы хотите позвонить. После чего производит вызов на данного абонента.

Пример:

 Абонент снимает трубку и набирает номер, на который завязан ivr-скрипт автосекретаря. После приветствия, абонент произносит фразу: Владислав Блинов. Система распознает данные слова и пытается найти в телефонной книге, соответствие имени и фамилии с номером. В случае успеха, система производит вызов на номер абонента.

Архитектура



- Clerk — он же ecss-clerk или автосекретарь. HTTP-сервер (elixir/phoenix), принимающий запросы на распознавание и отдающий результаты в виде JSON;
- Asr Manager — HTTP-сервер (python), который управляет TCP-инстансами Kaldi. Может пересобрать модель, добавив новые слова, держать несколько экземпляров tcp-серверов и управлять ими;
- TCP Asr Server — собственно, один из экземпляров, которым управляет Asr Manager.

Принцип работы

Пользователь снимает трубку и набирает номер, на который завязан ivr-скрипт автосекретаря. После синтезированного приветствия, в котором пользователя просят произнести имя и фамилию, поток с его микрофона отправляется на специальный порт MSR, который работает по следующему принципу: определяет тишину и нарезает по ней чанки. Каждый чанк отправляется отдельным HTTP POST-запросом в сторону ecss-clerk (через restfs) с заголовком "Content-type: Transfer-Encoding". По имени wav-файла ecss-clerk идентифицирует чанк. Каждый чанк отправляется на распознавание, по результату которого phonebook пытается найти подходящий номер. Если не удалось, clerk просит следующий чанк и так до тех пор, пока не получится найти номер или не сработает таймаут в ivr-скрипте.

В случае успеха, json с номером летит на MSR через restfs, MSR отдает json ядру. Промежуточные ответы тоже доходят до ядра. Они имеют HTTP-код 206 и не содержат в прилагавом теле json-а поля с номером, но имеют поле с распознанной строкой. Имея эту

строку, по таймауту ivr может синтезировать ответ для пользователя, уведомив о том, что найти распознанную строку не удалось. Если до таймаута ecss-clerk не пришлёт ответ ядру, то синтезированное уведомление не будет содержать распознанной строки.

Синтез происходит через restfs. Restfs проксирует запрос на настроенный tts-сервер.

После получения PCM ecss-clerk выполняет следующие действия:

- Отправляет rcm на распознавание (через TCP);
- По результату распознавания пытается найти номер в телефонной книге;
- Посылает ответ.

Принцип поиска распознанной строки в телефонной книге

Ответ от ASR

На этапе поиска по телефонной книге уже получен ответ от ASR.

Причем ответ — это не просто строка, а список строк.

Условный пример ответа от ASR:

i алексей
алексей лосев
алексей лысенко

Который преобразуем в список вида:

i ["алексей лысенко", "алексей лосев", "алексей"]

Где **"алексей лысенко"** финальный результат, а остальные промежуточные.

Причина, по которой мы учитываем промежуточные варианты проста — в них может быть правильный вариант распознавания. Вероятность этого мала, но она есть.

По какой книге производится поиск

Поиск производится по xml-книге, взятой с restfs.

Restfs, в свою очередь, формирует xml по данным из одного выбранного источника: ssw, mysql, ldap.

При старте автосекретаря производится импорт xml. По ней формируется структура вида ключ-значение, где ключ — имя, значение — телефон.

Кроме этого, формируется ещё одна подобная структура, содержащая уникальные имена и фамилии из книги, где ключ — уникальное имя, значение — полное имя.

По этим двум мапам — телефонной книге и уникальным именам и производится поиск.

Алгоритм поиска

1. Берем строку из списка;
2. Проверяем, содержится ли данная строка в таблице `hard_aliases`;
3. Раскрываем сокращения (алиасы из таблицы `aliases`) и добавляем их в список, по которому будет производиться рекурсивный поиск;
4. По каждой строке в сформированном списке делаем поиск:
 - a. Формируем список с перестановками для строки:
 - i. Проверяем, содержится ли строка (именно целиком, а не как подстрока) в уникальных именах. Если да — заменяем эту строку на полное имя;
 - ii. Ищем строку в телефонной книге. Если номер найден — поиск закончен.
5. Если в списке из пункта 3 ещё остались строки, то берем следующую.

Пример

Имеем ответ от ASR:

["женя казарцев", "женя казарцев", "женя"]

Двигаемся согласно алгоритму:

1. Берем строку **"женя казарцев"** из списка;
2. Проверяем, содержится ли данная строка в таблице `hard_aliases`. В нашем примере её нет, так что передаем строку дальше как **"женя казарцев"**;
3. Раскрываем сокращения [**"женя казарцев"**, **"божена казарцев"**, **"евгений казарцев"**, **"евгения казарцев"**];
4. По каждой строке в сформированном списке делаем поиск строки;

⚠ Строки **"женя казарцев"** и **"божена казарцев"** пропущены, т.к. пройдя всю цепочку ниже номера для них не были найдены.

Выбираем строку **"евгений казарцев"**:

- a. Формируем список с перестановками для строки [**"евгений казарцев"**, **"казарцев евгений"**]
 - i. Проверяем, содержится ли строка (именно целиком, а не как подстрока) в уникальных именах. Если да — заменяем эту строку на полное имя. Если не содержится, передаем как есть — **"евгений казарцев"**;
 - ii. Ищем строку в телефонной книге. Если не найден — берем следующий элемент из списка с перестановками **евгений казарцев -> 4819**;
5. Если в списке из пункта 3 ещё остались строки, то берем следующую. Здесь мы пропустили строки **"женя казарцев"** и **"божена казарцев"**.

Установка

Установка выполняется из репозитория:

```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/bionic/clerk/
1.0 stable main extras external' > /etc/apt/sources.list.d/eltex-ecss10-
stable.list"
sudo apt update
```

Далее необходимо установить пакет **ecss-clerk**:

```
sudo apt install ecss-clerk
```

Во время установки необходимо ответить на все задаваемые **вопросы**, особенно важно указать правильный url для импорта **телефонной книги** в формате xml и данные mysql пользователя.

Ответы на вопросы более детально описаны ниже.

Во время установки будут применены миграции в бд (создание базы данных, таблицы и т.д.). Во время первого запуска начнут создаваться стандартная база слов (около 6,5 тысяч).

После создания базы рекомендуется запустить **рекомпиляцию asr-модели**.

Ответы на вопросы:

Вопросы	Ответы
Выберите IPv4 для ecss-clerk (Choose IPv4 for ecss-clerk)	IP хоста
Укажите порт для ecss-clerk (Enter ecss-clerk port)	9010
Хотите ли вы сохранить записи? (Do you want save records?)	yes
Введите частоту дискретизации (Enter sample rate)	8000
Введите URL-адрес для импорта xml-телефонной книги (Enter url for import xml phonebook)	http://system.restfs.ecss:9990/ssw?host=book&user_agent=cisco_search&translit=false&skip_no_disp=true
Уровень логирования (Log level)	debug
Как вы хотите настроить пользователя mysql? (How do you want configure mysql user?)	create
Введите логин пользователя MySQL (Enter login for mysql user)	ecss-clerk
Введите размещения пользователя MySQL (Enter mysql user host)	127.0.0.1
Введите пароль для пользователя mysql (Enter password for mysql user)	password

Управление через systemd

Убедитесь, что сервисы запущены:

```
❗ sudo systemctl status ecss-clerk
sudo systemctl status kald-ru
```

В противном случае сервис необходимо включить.

```

❗ sudo systemctl start ecss-clerk
sudo systemctl start kaldi-ru

```

Конфигурация

Конфигурация происходит при установке deb-пакета ecss-clerk. Сам файл с конфигурацией расположен по пути /etc/ecss/ecss-clerk/config.yaml:

```

http:
  url: localhost # ip адрес ecss-clerk
  port: 9010     # порт ecss-clerk
asr:
  manager:
    ip: localhost # ip адрес http сервера kaldi (kaldi-ru)
    port: 9011    # порт http сервера kaldi (kaldi-ru)
  backend: tcp_kaldi # сервис, с помощью чего kaldi-ru запускает модель (сейчас не используется)
  frequency: 8000   # частота, на которой будет запущен asr
pcm:
  save_pcm: true    # сохранять ли приходящие pcm
  pcm_dir: /var/lib/ecss/ecss-clerk/pcm_files # куда сохранять pcm, если нужно
phonebook:
  url_xml: http://system.restfs.ecss:9990/ssw?
  host=book&user_agent=cisco_search&translit=false&skip_no_disp=true # url для импорта xml книги
log:
  level: debug # уровень лога
  console_level: info # уровень консольного лога (видно через journalctl)
  path: /var/log/ecss/ecss-clerk # куда писать лог
database:
  database: ecss_clerk_db_repo # название бд в mysql
  username: ecss-clerk
  password: password
  hostname: localhost
  aliases_dir: /var/lib/ecss/ecss-clerk/default_aliases # директория, в которой хранятся файлы с
дефолтными сокращениями (для миграции)

```

Править конфигурацию руками не рекомендуется, лучше делать это через dpkg-reconfigure

```

❗ sudo dpkg-reconfigure ecss-clerk

```

Проверка работоспособности

Для проверки работоспособности можно обратиться к сервису через curl:

```
curl <host>:9010/test.pcm -T test.pcm
```

Где

- host — адрес хоста, на котором запущен ecss-clerk;
- test.pcm — частота из config.yaml, кодек pcm_s16le, 1 канал (можно с wav заголовком).

Настройка на стороне ECSS-10

Интеграция с ECSS-10 выполнена через IVR-скрипт.

После звонка на ivr, голос от абонента идет на asr_service. Если до таймаута приходит ответ с номером, то происходит звонок на этого абонента. В противном случае мы оказываемся в другой ветке, из которой делается повторный запрос (блок goto, который выполняется 2 раза). Причем, если промежуточный ответ от автосекретаря содержал не пустое поле с распознанной строкой, то она будет произнесена. Кроме этого есть два RPC блока, необходимые для вызова коллбеков на стороне ecss-clerk. Они тегируют запросы как удачные и не удачные. Соответственно, если было падение с таймаутом, то будет вызван негативный коллбек (negative_url). Если в ответе был номер, то будет вызван удачный коллбек (positive_url).

IVR-скрипт

```
{
  "actions": {
    "begin_1": {
      "name": "begin",
      "params": {
        "description": ""
      },
      "pos": {
        "x": 2,
        "y": 0
      },
      "cases": {
        "next": "ivr_2",
        "links": {
          "next": {
            "points": [
              {
                "cx": 450,
                "cy": 70
              },
              {
                "cx": 450,
                "cy": 125
              },
              {
                "cx": 450,
                "cy": 180
              }
            ]
          },
          "text_pos": {
            "cx": 450,
            "cy": 70
          }
        }
      },
      "ivr_2": {
        "name": "ivr",
        "params": {
          "description": ""
        },
        "play": {
          "type": "speech",
          "name": "Произнесите имя и фамилию абонента.",
          "tts": {
            "folder": "b1giqb9no4enfheittff",
            "is_use_folder": false,
            "key": "AgAAAAAJ1M1DAATuwaBa0CTmh006hge0vVDkmtg",
            "is_use_key": false,
            "lang": "ru-RU",
            "is_use_lang": false,
            "voice": "zahar",
            "is_use_voice": false,
            "speed": "1.2",
            "is_use_speed": false,
            "emotion": "good",
            "is_use_emotion": false
          },
          "location": "",
          "group": "",
          "variable_type": ""
        },
        "ivr": [],
        "asr_service": "localhost:9010",
        "playback_interruption": true,
        "volume_interruption": 25,
        "asr_delay": 0,
        "wait_time": 8,
        "extension_dialing": true,
        "direct_call_to_extension": true,
        "max_digits": "3",
        "interdigit_timeout": 2,
        "use_calling_iface": false,
        "pos": {
          "x": 2,
          "y": 1
        },
        "cases": {
          "Extension": "condition_4",
          "No Match": "condition_22",
          "links": {
            "Extension": {
              "points": [
                {
                  "cx": 450,
                  "cy": 220
                },
                {
                  "cx": 450,
                  "cy": 275
                },
                {
                  "cx": 450,
                  "cy": 330
                }
              ]
            },
            "text_pos": {
              "cx": 450,
              "cy": 220
            },
            "No Match": {
              "points": [
                {
                  "cx": 450,
                  "cy": 220
                },
                {
                  "cx": 540,
                  "cy": 350
                },
                {
                  "cx": 630,
                  "cy": 480
                }
              ]
            },
            "text_pos": {
              "cx": 450,
              "cy": 220
            }
          },
          "condition_4": {
            "name": "condition",
            "params": {
              "description": "",
              "conditions": [
                {
                  "code": "0",
                  "condition": "(%ASR_RECOGNIZED%<>\\\"\\\")and(%ASR_NUMBER%<>\\\"\\\")"
                }
              ]
            },
            "pos": {
              "x": 2,
              "y": 2
            },
            "cases": {
              "0": "condition_23",
              "False": "condition_22",
              "links": {
                "0": {
                  "points": [
                    {
                      "cx": 450,
                      "cy": 370
                    },
                    {
                      "cx": 360,
                      "cy": 425
                    },
                    {
                      "cx": 270,
                      "cy": 480
                    }
                  ]
                },
                "text_pos": {
                  "cx": 450,
                  "cy": 370
                },
                "False": {
                  "points": [
                    {
                      "cx": 450,
                      "cy": 370
                    },
                    {
                      "cx": 540,
                      "cy": 425
                    },
                    {
                      "cx": 630,
                      "cy": 480
                    }
                  ]
                },
                "text_pos": {
                  "cx": 450,
                  "cy": 370
                }
              },
              "set_10": {
                "name": "set",
                "params": {
                  "description": ""
                },
                "variables": [
                  {
                    "key": "EXTENSION",
                    "value": "%ASR_NUMBER%"
                  }
                ],
                "pos": {
                  "x": 1,
                  "y": 7
                },
                "cases": {
                  "next": "dial_11",
                  "links": {
                    "next": {
                      "points": [
                        {
                          "cx": 270,
                          "cy": 1120
                        },
                        {
                          "cx": 270,
                          "cy": 1175
                        },
                        {
                          "cx": 270,
                          "cy": 1230
                        }
                      ]
                    },
                    "text_pos": {
                      "cx": 270,
                      "cy": 1120
                    }
                  },
                  "dial_11": {
                    "name": "dial",
                    "params": {
                      "description": "",
                      "numbers": "%EXTENSION%"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

```

", "noanswer_timeout": 55, "use_calling_iface": false, "pos": {"x": 1, "y": 8}, "cases": {"Busy/No
answer": "undefined", "Error": "undefined"}, "links": {"Busy/No answer": {"points": [{"cx": 270, "cy":
1270}, {"cx": 270, "cy": 1325}, {"cx": 270, "cy": 1380}], "text_pos": 0.2, "pos": {"x": 1, "y": 9}}, "Error":
{"points": [{"cx": 270, "cy": 1270}, {"cx": 180, "cy": 1325}, {"cx": 90, "cy": 1380}], "text_pos": 0.2, "pos"
: {"x": 0, "y": 9}}}, "goto_17": {"name": "goto", "params": {"description": "", "max_hops": 2, "goto": "ivr_2"},
"pos": {"x": 4, "y": 9}, "cases": {"Exit": "play_18", "Goto": "ivr_2"}, "links": {"Exit": {"points"
: [{"cx": 810, "cy": 1420}, {"cx": 810, "cy": 1475}, {"cx": 810, "cy": 1530}], "text_pos": 0.2}, "Goto":
{"points": [{"cx": 810, "cy": 1420}, {"cx": 810, "cy": 800}, {"cx": 450, "cy": 180}], "text_pos": 0.2}}, "play_18": {"name": "play", "params": {"description": "", "play": [{"type": "speech", "name": "Исчерпан
лимит попыток. До свидания-."}, "tts": {"folder": "b1g1qb9no4enfheittff", "is_use_folder": false, "key": "AgAAAAAJ1M1DAATuwaBaOCTmh006hge0vVDkmtg", "is_use_key": false, "lang": "ru-RU", "is_use_lang": false, "voice": "zahar", "is_use_voice": false, "speed": "1.2", "is_use_speed": false, "emotion":
"neutral", "is_use_emotion": true}, "location": "", "group": "", "variable_type": ""}], "replay": "1"},
"pos": {"x": 4, "y": 10}, "cases": {"next": "undefined"}, "links": {"next": {"points": [{"cx": 810, "cy":
1570}, {"cx": 990, "cy": 1625}, {"cx": 1170, "cy": 1680}], "text_pos": 0.2, "pos": {"x": 6, "y": 11}}}, "play_19": {"name": "play", "params": {"description": "", "play": [{"type": "speech", "name": "Соединяю."}, "tts": {"folder": "b1g1qb9no4enfheittff", "is_use_folder": false, "key": "AgAAAAAJ1M1DAATuwaBaOCTmh006hge0vVDkmtg", "is_use_key": false, "lang": "ru-RU", "is_use_lang": false, "voice": "zahar", "is_use_voice": false, "speed": "1.2", "is_use_speed": false, "emotion": "good", "is_use_emotion": false}, "location": "", "group": "", "variable_type": ""}], "replay": "1"}, "pos": {"x": 1, "y": 6}, "cases":
{"next": "set_10"}, "links": {"next": {"points": [{"cx": 270, "cy": 970}, {"cx": 270, "cy": 1025}, {"cx":
270, "cy": 1080}], "text_pos": 0.2}}, "rpc_20": {"name": "rpc", "params": {"description": "", "type": "HT
TP", "url": "http://%ASR_SERVICE%/ASR_NEGATIVE_URL%", "method": "HEAD", "request_timeout": 0, "headers": [], "max_bytes": "10000", "expected_encoding": "utf", "rpc_comm": []}, "pos": {"x": 2, "y": 4}, "cases":
{"Error": "set_24"}, "links": {"Error": {"points": [{"cx": 450, "cy": 670}, {"cx": 450, "cy": 725}, {"cx":
450, "cy": 780}], "text_pos": 0.2}}, "rpc_21": {"name": "rpc", "params": {"description": "", "type": "HTT
P", "url": "http://%ASR_SERVICE%/ASR_POSITIVE_URL%", "method": "HEAD", "request_timeout": 0, "headers": [], "max_bytes": "10000", "expected_encoding": "utf", "rpc_comm": []}, "pos": {"x": 0, "y": 4}, "cases":
{"Error": "set_25"}, "links": {"Error": {"points": [{"cx": 90, "cy": 670}, {"cx": 90, "cy": 725}, {"cx": 90,
cy": 780}], "text_pos": 0.2}}, "condition_22": {"name": "condition", "params": {"description": "", "con
ditions": [{"code": "0", "condition": "(%ASR_NEGATIVE_URL%<>\\\"\\\"))"}], "pos": {"x": 3, "y": 3}, "cases":
{"0": "rpc_20", "False": "condition_27"}, "links": {"0": {"points": [{"cx": 630, "cy": 520}, {"cx": 540, "c
y": 575}, {"cx": 450, "cy": 630}], "text_pos": 0.2}, "False": {"points": [{"cx": 630, "cy": 520}, {"cx": 630,
cy": 725}, {"cx": 630, "cy": 930}], "text_pos": 0.2}}, "condition_23": {"name": "condition", "params":
{"description": "", "conditions": [{"code": "0", "condition": "(%ASR_POSITIVE_URL%<>\\\"\\\"))"}], "pos":
{"x": 1, "y": 3}, "cases": {"0": "rpc_21", "False": "play_19"}, "links": {"0": {"points": [{"cx": 270, "cy":
520}, {"cx": 180, "cy": 575}, {"cx": 90, "cy": 630}], "text_pos": 0.2}, "False": {"points": [{"cx": 270, "cy":
520}, {"cx": 270, "cy": 725}, {"cx": 270, "cy": 930}], "text_pos": 0.2}}, "set_24": {"name": "set", "params": {"description": "", "variables": [{"key": "ASR_NEGATIVE_URL", "value": ""}], "pos": {"x": 2, "y": 5}, "cases": {"next": "condition_27"}, "links": {"next": {"points": [{"cx": 450, "cy": 820}, {"cx": 540, "cy":
875}, {"cx": 630, "cy": 930}], "text_pos": 0.2}}, "set_25": {"name": "set", "params": {"description": "", "variables": [{"key": "ASR_POSITIVE_URL", "value": ""}], "pos": {"x": 0, "y": 5}, "cases": {"next": "play_19"}, "links": {"next": {"points": [{"cx": 90, "cy": 820}, {"cx": 180, "cy": 875}, {"cx": 270, "cy": 930}], "text_pos": 0.2}}, "condition_27": {"name": "condition", "params": {"description": "", "conditions":
[{"code": "0", "condition": "(%ASR_RECOGNIZED%<>\\\"\\\"))and(%ASR_NUMBER%=\\\"\\\"))"}], "pos": {"x": 3, "y":
6}, "cases": {"0": "play_28", "False": "play_32"}, "links": {"0": {"points": [{"cx": 630, "cy": 970}, {"cx":
630, "cy": 1025}, {"cx": 630, "cy": 1080}], "text_pos": 0.2}, "False": {"points": [{"cx": 630, "cy": 970},
{"cx": 720, "cy": 1025}, {"cx": 810, "cy": 1080}], "text_pos": 0.2}}, "play_28": {"name": "play", "params": {"description": "", "play": [{"type": "speech", "name": "Не удалось найти абонента
%ASR_RECOGNIZED%"}, "tts": {"folder": "b1g1qb9no4enfheittff", "is_use_folder": false, "key": "AgAAAAAJ1M1DAATuwaBaOCTmh006hge0vVDkmtg", "is_use_key": false, "lang": "ru-RU", "is_use_lang": false, "voice": "

```

```

zahar", "is_use_voice": false, "speed": "1.1", "is_use_speed": true, "emotion": "good",
"is_use_emotion": false}, "location": "", "group": "", "variable_type": ""}, "replay": "1", "pos": {"x":
3, "y": 7}, "cases": {"next": "set_29"}, "links": {"next": {"points": [{"cx": 630, "cy": 1120}, {"cx": 720, "
cy": 1175}, {"cx": 810, "cy": 1230}], "text_pos": 0.2}}, "set_29": {"name": "set", "params":
{"description": "", "variables": [{"key": "ASR_RECOGNIZED", "value": ""}, {"key": "ASR_NUMBER", "value": ""
}], "pos": {"x": 4, "y": 8}, "cases": {"next": "goto_17"}, "links": {"next": {"points": [{"cx": 810, "cy":
1270}, {"cx": 810, "cy": 1325}, {"cx": 810, "cy": 1380}], "text_pos": 0.2}}, "play_32": {"name": "play", "p
arams": {"description": "", "play": [{"type": "speech", "name": "Не удалось распознать абонента.", "tt
s": {"folder": "b1g1qb9no4enfheittff", "is_use_folder": false, "key": "AgAAAAAJ1M1DAATuwaBaOCTmh006h
ge0vVDkmtg", "is_use_key": false, "lang": "ru-RU", "is_use_lang": false, "voice": "zahar", "is_use_voic
e": false, "speed": "1.2", "is_use_speed": false, "emotion": "good", "is_use_emotion": false},
"location": "", "group": "", "variable_type": ""}], "replay": "1", "pos": {"x": 4, "y": 7}, "cases":
{"next": "set_29"}, "links": {"next": {"points": [{"cx": 810, "cy": 1120}, {"cx": 810, "cy": 1175}, {"cx":
810, "cy": 1230}], "text_pos": 0.2}}, "name": "simple_auto_attendant5", "description": "", "version": "
3.14.6.67", "settings": {"speech": {"folder": "b1g1qb9no4enfheittff", "key": "AgAAAAAJ1M1DAATuwaBaO
CTmh006hge0vVDkmtg", "lang": "ru-RU", "voice": "zahar", "speed": "1.2", "emotion": "good", "terminate_if
_tts_failed": false}}, "id": "163xd8a7ecfd21v5"}

```

⚠ После установки ecss-clerk скрипт можно найти в папке /etc/ecss/ecss-clerk/simple_auto_attendant5.json

Импортировать скрипт можно через интерфейс командной строки:

```

domain/<DOMAIN>/ivr/script/.import --id simple_auto_attendant5 --json <IVR-скрипт>

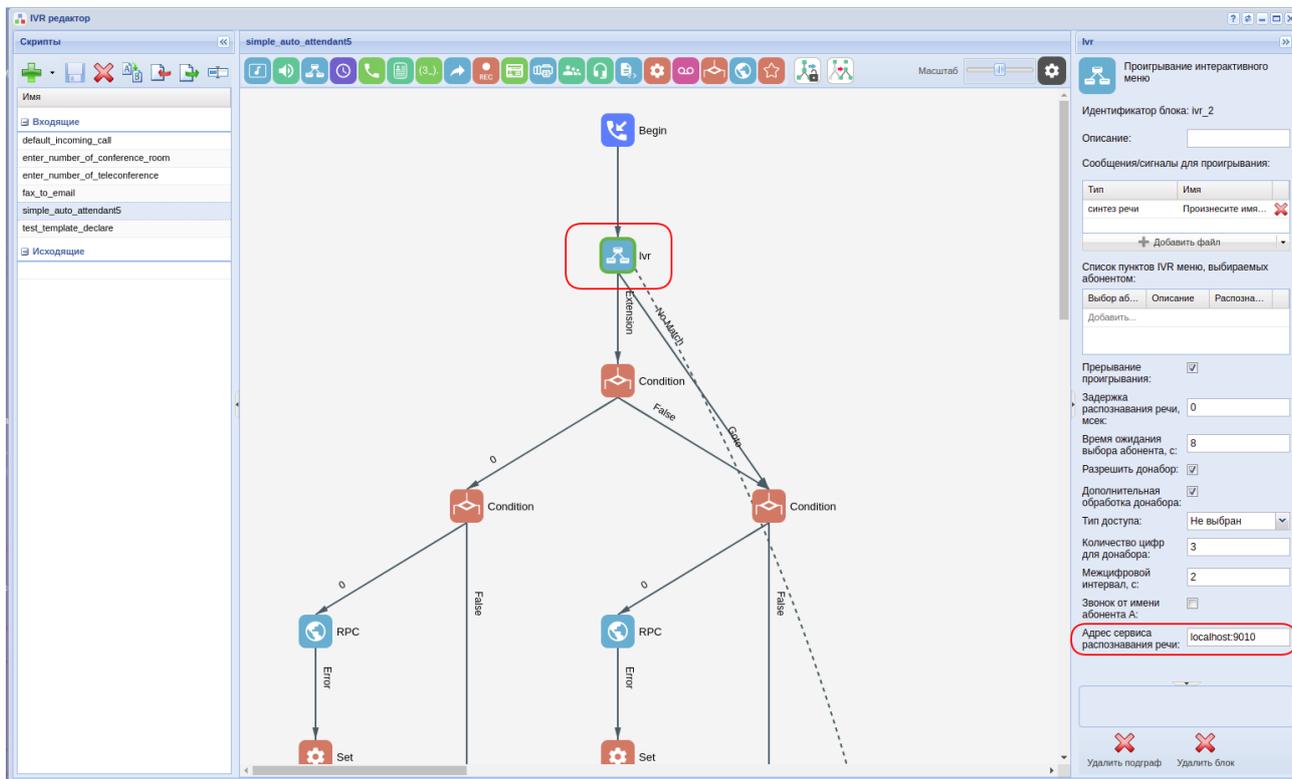
```

Пример

```

admin@[sip1@ecss1#ECSS 010145]:/$ domain/arko/ivr/script/.import --id
simple_auto_attendant5 --json {"actions":{"begin_1":{"name":"begin","params":
{"description":""},"pos":{"x":2,"y":0},"cases":
{"next":"ivr_2"} / ... / "emotion":"good","terminate_if_tts_failed":false}}, "
id":"163xd8a7ecfd21v5"}
Script successfully imported with id <<"simple_auto_attendant5">>

```



После импорта нужно зайти **IVR редактор**, выбрать скрипт "simple_auto_attendant5" и в поле "адрес сервиса распознавания речи" блока "ivr" указать ip-адрес и порт сервера с установленным ecss-clerk.

Затем необходимо настроить нового абонента, включить на нём услугу «personal ivr» и выбрать скрипт, проимпортированный ранее.

Для добавления собственных слов в базу нужно воспользоваться HTTP API сервиса ecss-clerk. Более подробно об этом написано в [данном разделе](#).

Интеграции

В системе ECSS-10 поддерживаются следующие интеграции:

Интеграция с CRM Битрикс-24

- [Общее описание](#)
- [Активация CSTA](#)
- [Управление интеграциями](#)
 - [1. Добавление интеграции](#)
 - [2. Удаление интеграции](#)
 - [3. Просмотр списка интеграций](#)
 - [4. Замена ключа интеграции](#)
 - [5. Изменение списка обслуживаемых номеров](#)
- [Добавление разрешения абоненту SSW](#)
- [Установка ecss-crm-server](#)
 - [Описание конфигурации ecss-crm-server](#)
- [Установка приложения](#)
 - [Ручная установка приложения в облачной версии Битрикс-24](#)
 - [Регистрация приложения в ecss-crm-server](#)
 - [Настройка компании в интерфейсе ecss-crm-server](#)
- [Активация click 2 call](#)
- [Управление вызовами в интерфейсе Битрикс-24](#)

В разделе приведено описание настроек взаимодействия ECSS-10 с системой для ведения проектов, работы с клиентами и настройки бизнес-процессов — [CRM Битрикс-24](#).

Общее описание

Интеграция CRM Битрикс-24 с телефонией ECSS-10 осуществляется с помощью приложения, встраиваемого на веб-страницу CRM Битрикс-24 и сервиса ecss-crm-server. Данное приложение обеспечивает работу таких функций как click to call, управление вызовами (принять/отклонить/отбить), список текущих активных вызовов, всплывающие нотификации, информирующие об активных вызовах и их фазах, а также определение контакта по номеру телефона.

Для того чтобы приложение могло взаимодействовать с системой ECSS-10 посредством ECSS-CRM-SERVER, необходимо:

- активировать CSTA в системе ECSS-10 (без этого приложение не будет отображать вызовы, инициированные не со страницы CRM Битрикс-24);
- добавить в домене интеграцию для аутентификации приложения (см. [Добавление интеграции](#));
- добавить соответствующее разрешение для абонента, номер которого будет использоваться для осуществления вызовов из CRM Битрикс-24 (см. [Интеграция с CRM Битрикс-24](#));
- установить и настроить сервис ecss-crm-server;
- установить приложение и указать настройки подключения к системе ECSS-10.

Активация CSTA

Для того чтобы интеграция могла отображать вызовы, инициированные не со страницы CRM Битрикс-24, нужно убедиться, что в системе активирован функционал [CSTA](#). Это можно сделать командой CoCon:

```
/api/csta/set enabled true
```

И в свойствах домена параметр `vats_type` должен быть выставлен в `private`:

```
admin@mycelium1@ecss1:/$ domain/nick/properties/info
```

Property	Domain	Value
...		
vats_type	nick	private

Управление интеграциями

Интеграция представляет собой пару, состоящую из идентификатора клиента и API ключа. Интеграции используются для аутентификации сторонних сервисов, использующих ECSS CRM SERVER. Команды управления интеграциями приведены в разделе "[Команды управления ключами внешних интеграций](#)".

1. Добавление интеграции

Для добавления интеграции используется команда:

```
/domain/<DOMAIN>/integrations/add <CLIENT_ID> --service --numbers {<RANGE>}
```

Добавить новую интеграцию, API-ключ будет сгенерирован автоматически.

<CLIENT_ID> — идентификатор клиента, для которого будет создан ключ интеграции.

<RANGE> — номера абонентов ECSS-10, обслуживаемых интеграцией.

Пример:

```
admin@[mycelium1@ecss1#test_davidenko]:/$ domain/nick/integrations/add
MyCompanyBitrix24 --service --numbers {101,102,103,110-120}
New integration was added.
API key: wQ7iD7nJ2hQiJDY2wkER2qr3lJYaDw2dE24rh13iHDaQwwwln242rjwHkJPw4DYK
```

 Созданная интеграция должна использоваться строго для одной crm-системы.

2. Удаление интеграции

Для удаления интеграции используется команда:

```
/domain/<DOMAIN>/integrations/remove <CLIENT_ID>
```

Удалить интеграцию.

<CLIENT_ID> — идентификатор клиента, для которого будет удалён ключ интеграции.

```
admin@mycelium1@ecss1:/$ domain/nick/integrations/remove tester4
ok
```

```
[exec at: 22.12.2019 13:20:18, exec time: 16ms, nodes: ds1@ecss1]
```

3. Просмотр списка интеграций

Для просмотра списка интеграций используется команда:

```
/domain/<DOMAIN>/integrations/list
```

Просмотреть список интеграций.

Список интеграций отображается в виде таблицы, пример:

```
admin@[mycelium1@ecss1#test_davidenko]:/$ domain/nick/integrations/list
```

Client id	Type	API Key
Options		
MyCompanyBitrix24	service	wQ7iD7nJ2hQiJDY2wkER2qr3lJYaDw2dE24rh13iHdaQwwwln242rjwHkJPw4DYK
Numbers:		
101		
102		
103		
110		
111		
112		
113		
114		
115		
116		
117		
118		
119		
120		

4. Замена ключа интеграции

Для замены ключа интеграции используется команда:

```
/domain/<DOMAIN>/integrations/replace <CLIENT_ID>
```

Заменить ключ интеграции.

<CLIENT_ID> — идентификатор клиента, для которого будет заменён ключ интеграции.

Пример работы команды:

```
admin@mycelium1@ecss1:/$ domain/nick/integrations/replace tester
```

```
Old key was replaced with new one.
```

```
New API key: wT5neQJ1hykwvntQeJ50yWkoFMizeAJcJwhcowFvV5eVe5zf5C1imCvoMFizAKQz
```

```
[exec at: 22.12.2019 10:15:31, exec time: 28ms, nodes: ds1@ecss1]
```

5. Изменение списка обслуживаемых номеров

Для изменения списка обслуживаемых интеграцией номеров используется команда:

```
/domain/<DOMAIN>/integrations/set <CLIENT_ID> --numbers {<RANGE>}
```

Изменить список обслуживаемых номеров.

<CLIENT_ID> — идентификатор клиента, для которого будет изменен список обслуживаемых номеров.

<RANGE> — новый список номеров абонентов ECSS-10, обслуживаемых интеграцией.

Пример работы команды:

```
admin@[mycelium1@ecss1#test_davidenko]:/$ domain/nick/integrations/set
MyCompanyBitrix24 --numbers {101,102,103,104,105}
Integration properties changed
admin@[mycelium1@ecss1#test_davidenko]:/$ domain/nick/integrations/list
```

Client id	Type	API Key
MyCompanyBitrix24	service	wQ7iD7nJ2hQiJDY2wkER2qr3lJYaDw2dE24rh13iHDaQwww1n242rjwHkJpw4DYK
	Numbers:	
101		
102		
103		
104		
105		

Добавление разрешения абоненту SSW

При выполнении логина в ECSS CRM виджет сообщает ECSS-10 номер телефона абонента, который будет закреплён за этим виджетом. Это требует специального разрешения, активированного у абонента.

Для активации разрешения нужно выставить соответствующему alias значение true для свойства api\call\enabled. Например:

Пример:

```
admin@mycelium1@ecss1:/$ domain/nick/alias/set 101 loc.gr 101@nick
api\call\enabled true
There are aliases within domain nick affected by settings property
api\call\enabled:
  240473 <-> 062743c44271f9e7
```

Legend:

```
a: Setup alias address
i: Setup alias interface
!: Setup broken. Alias already exists
*: Setup broken. Alias not exists
L: Setup broken. No free subscriber licences.
x: Setup broken. Unexpected error
<empty>: Successful setup
```

```
[exec at: 20.12.2019 22:42:09, exec time: 101ms, nodes: ds1@ecss2]
```

В том случае, если данное свойство не будет выставлено, виджет не сможет выполнить логин в ECSS CRM.

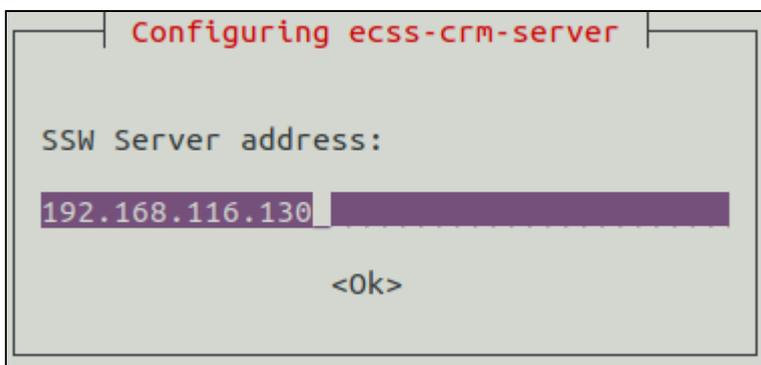
Установка ecss-crm-server

Для установки пакета выполните в консоли ОС команды:

```
sudo apt update
sudo apt install ecss-crm-server
```

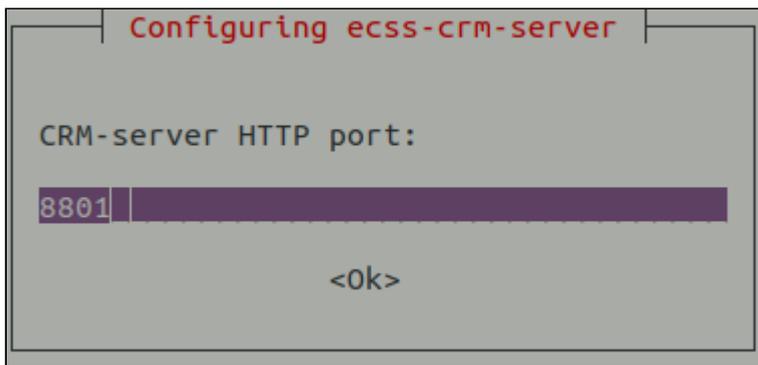
В конце установки пакета проводится его настройка, в ходе которой нужно будет ответить на несколько вопросов, которые приведены в [Приложении Д](#):

Укажите адрес сервера, на котором будет запущен ecss-crm-server. По умолчанию предлагается 127.0.0.1.

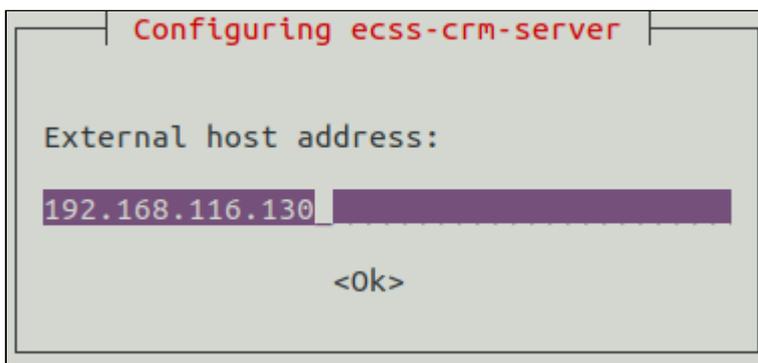


❗ Указание 0.0.0.0 вместо адреса делает невозможным встраивание карточки вызова в интерфейс Битрикс-24.

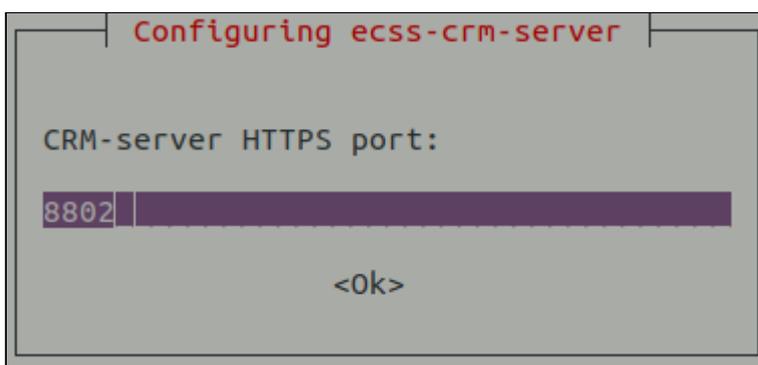
Укажите номера tcp-портов HTTP и HTTPS, на которых будет слушать сервер `ecss-crm-server`. Стандартные порты для HTTP (80) и HTTPS (443) не предлагаются пакетом по умолчанию, так как при установке на один сервер с пакетом `ecss-node`, они будут конфликтовать с веб-конфигуратором ECSS-10. Администратор может самостоятельно указать данные порты при установке пакета, если они не создадут конфликт с другими сервисами.



```
Configuring ecss-crm-server
CRM-server HTTP port:
8801
<Ok>
```



```
Configuring ecss-crm-server
External host address:
192.168.116.130
<Ok>
```



```
Configuring ecss-crm-server
CRM-server HTTPS port:
8802
<Ok>
```

Описание конфигурации `ecss-crm-server`

Настройки, указанные при инсталляции, а также некоторые другие, могут быть изменены после инсталляции пакета в файле `/usr/share/ecss/ecss-crm-server/src/settings.js`.

Ниже представлен пример его содержимого после инсталляции по умолчанию.

```

var settings = {
  path: '/',
  dist: "/usr/share/ecss/ecss-crm-server/frontend/dist",
  // Если указан proxyGatewayURL, то при монтировании карточки звонка, а
  также
  // в redirect_uri будет использоваться данный URL.
  // Если proxyGatewayURL указан, то адрес встроенной карточки звонка будет
  // сконструирован следующим образом:
  // ${settings.proxyGatewayURL}/call/call.html
  // Если proxyGatewayURL НЕ указан, то адрес redirect_uri, и адрес
  встраиваемой
  // карточки звонка, будет сконструирован следующим образом:
  // https://${settings.host}:${settings.proxyPort}/call/call.html
  // proxyGatewayURL: "https://<проxy-origin>:<проxy-port>/bitrix",
  // Если опция активна, то вместо http/https портов будут использованы
  // Unix-сокеты. Используется для работы через проксирующий веб-сервер
  useUnixSockets: false,
  // httpUnixSocket: "/var/lib/ecss/ecss-crm-server/http",
  // httpsUnixSocket: "/var/lib/ecss/ecss-crm-server/https",
  // Запросы на сервер проксируются с помощью nginx.
  host: '127.0.0.1',
  // Адрес SSW задаётся снаружи.
  sswHost: '192.168.116.130',
  hostPort: 8801,
  proxyPort: 8802,
  // Серверу необходимо знать адрес прокси-сервера.
  proxyHost: '192.168.116.130',
  // В случае работы через проxy gateway, т.е по умолчанию, данные
  сертификаты
  // будут использованы для связи с прокси веб-сервером по https
  sslKeyFilename: "/usr/share/ecss/ecss-crm-server/key.pem",
  sslCertFilename: "/usr/share/ecss/ecss-crm-server/cert.pem",
  // bitrixClientId: "some client id",
  // bitrixClientSecret: "some client secret",
  bitrixTokenHost: "https://oauth.bitrix.info",
  logLevel: 'warn',
  // logLevel: 'debug',
  defaultLogFile: '/var/log/ecss/ecss-crm-server/default.log',
  errorLogFile: '/var/log/ecss/ecss-crm-server/error.log',
  logToConsole: false,
  rebindPlacement: false
};
module.exports = settings;

```

где:

- path — HTTP-префикс пути, по которому будет доступен Битрикс-24 коннектор;
- dist — путь, по которому будут доступны страницы и статические файлы сервера;
- serverHost — адрес сервера, указываемый при инсталляции;
- httpPort — HTTP-порт. Игнорируется при работе через proxy gateway;
- httpsPort — HTTPS-порт. Игнорируется при работе через proxy gateway;
- sslKeyFilename, sslCertFilename — сертификаты, которые при установке генерирует пакет ecss-crm;
- bitrixClientId, bitrixClientSecret — если Битрикс-24 коннектор используется для потребностей одной компании, то можно указать параметры, полученные при регистрации приложения прямо в конфиге. Тогда их не нужно будет указывать при регистрации в браузере (см. Регистрация приложения в Битрикс-24 коннекторе);
- bitrixTokenHost — хост для авторизации OAuth2.0;
- logLevel — уровень логов. По умолчанию 'warn'. Поддерживаются следующие уровни логов в порядке возрастания подробности: error, warn, info, http, verbose, debug, silly. На текущий момент используются не все логи. Рекомендуется использовать 'warn' при обычной работе и 'debug' при поиске проблем;
- defaultLogFile — файл, в который пишутся все логи с учетом указанной подробности;
- errorLogFile — файл, в который пишутся только критические ошибки (уровень подробности error);
- logToConsole — выводить логи в stdout;
- rebindPlacement — если из-за какой-то внутренней ошибки не удалось встроить карточку в интерфейс Битрикс-24 при первом входе в компанию, можно вызвать эту процедуру принудительно, выставив значение параметра rebindPlacement в true. Нужно учитывать, что при каждом повторном встраивании карточки страница CRM-Битрикс-24 должна быть обновлена, если она открыта в браузере. После успешного встраивания необходимо выставить значение rebindPlacement обратно в false.

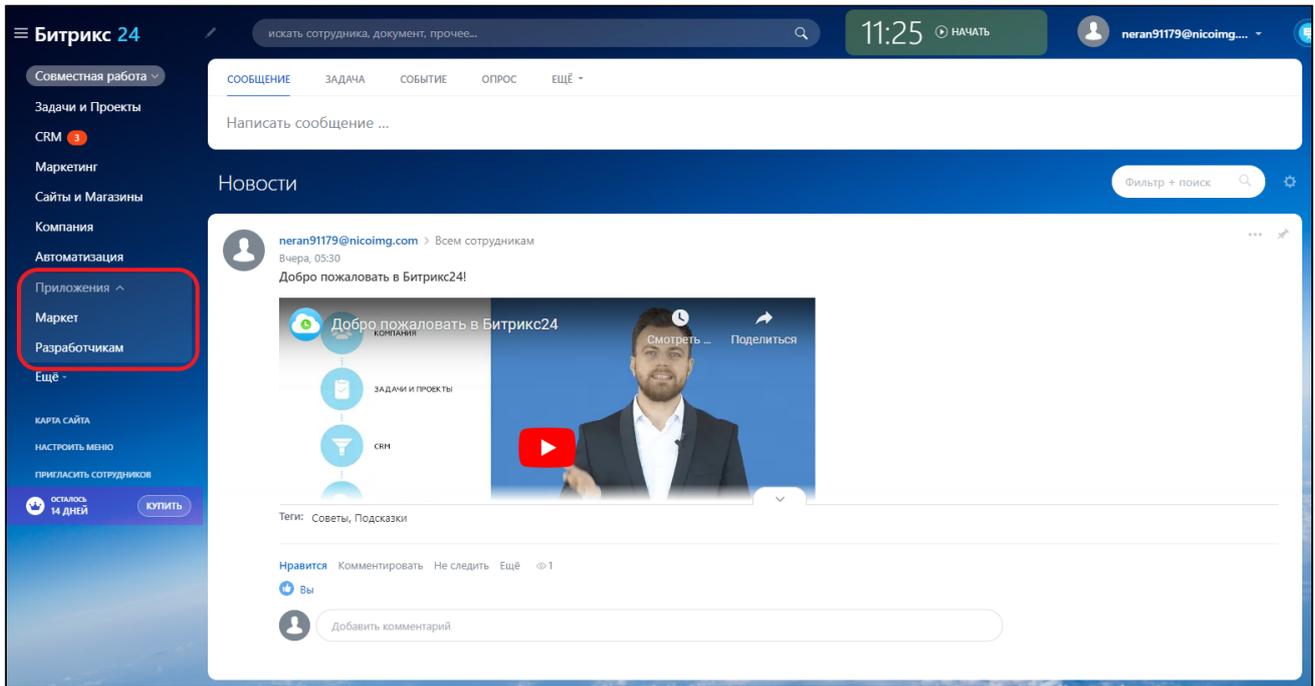
Установка приложения

 Интеграция Битрикс-24 не устанавливается из магазина приложений и может быть установлена только вручную.

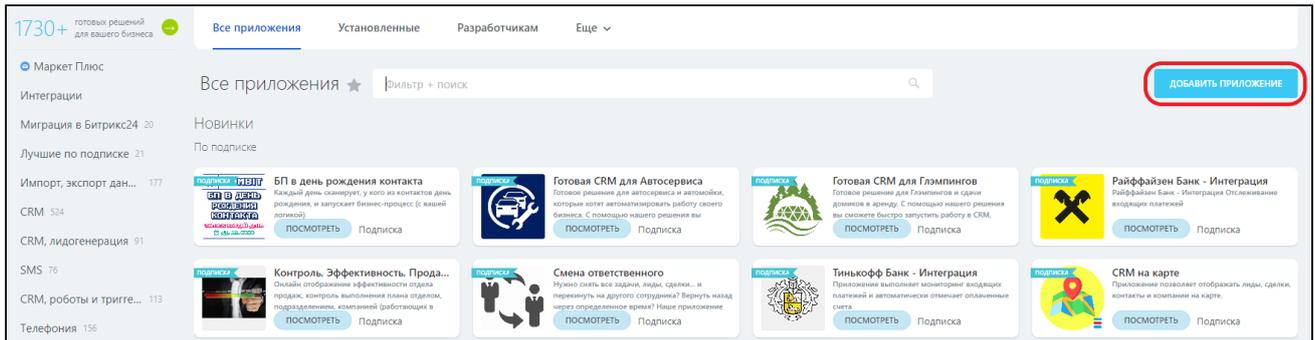
Ручная установка приложения в облачной версии Битрикс-24

Для установки приложения необходимо создать локальное приложение Битрикс-24:

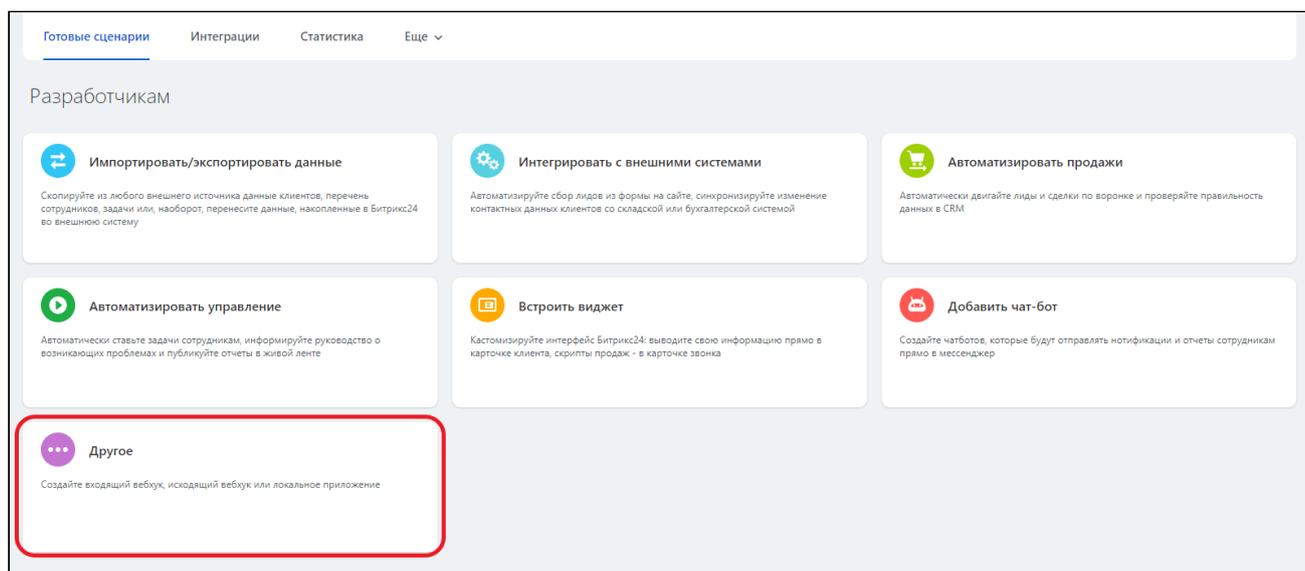
1. Для перехода в меню откройте вкладку **Приложение** в меню Битрикс-24, располагающимся в левой части экрана, и выберите пункт **Маркет** в выпадающем списке.



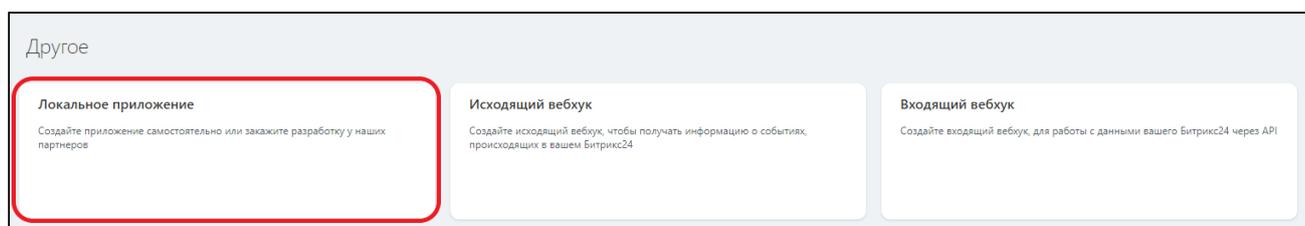
2. В открывшемся меню «Маркет» нажмите кнопку **Добавить приложение.**



3. В меню «Готовые сценарии» нажмите на вкладку **Другое.**



4. В следующем окне нажмите на вкладку **Локальное приложение**.



5. В настройках локального приложения необходимо указать следующие параметры:

1. Название интеграции (по умолчанию «Локальное приложение»).
2. Тип приложения «Серверное».
3. Ссылку на `ecss-crm-server(<IP>:8802/bitrix)`.
4. Галочку «Приложение использует только API».
5. В графе «Настройка прав» выбрать:
 - a. Встраивание приложений (placement);
 - b. Пользователи (user);
 - c. Телефония (telephony);
 - d. Телефония (совершение звонков) (call).

ecss10 1

Создайте приложение самостоятельно или закажите разработку у наших партнеров

Локальное приложение

2 Серверное Статичное

Путь вашего обработчика*

3

Путь для первоначальной установки

4 Использует только API

Настройка прав

5

СОХРАНИТЬ ЗАКРЫТЬ

Ниже представлено окно выбора настроек прав.

Задачи (task)

Задачи (расширенные права) (tasks_extended)

Бизнес-процессы (bizproc)

Форум (forum)

Сайты (landing)

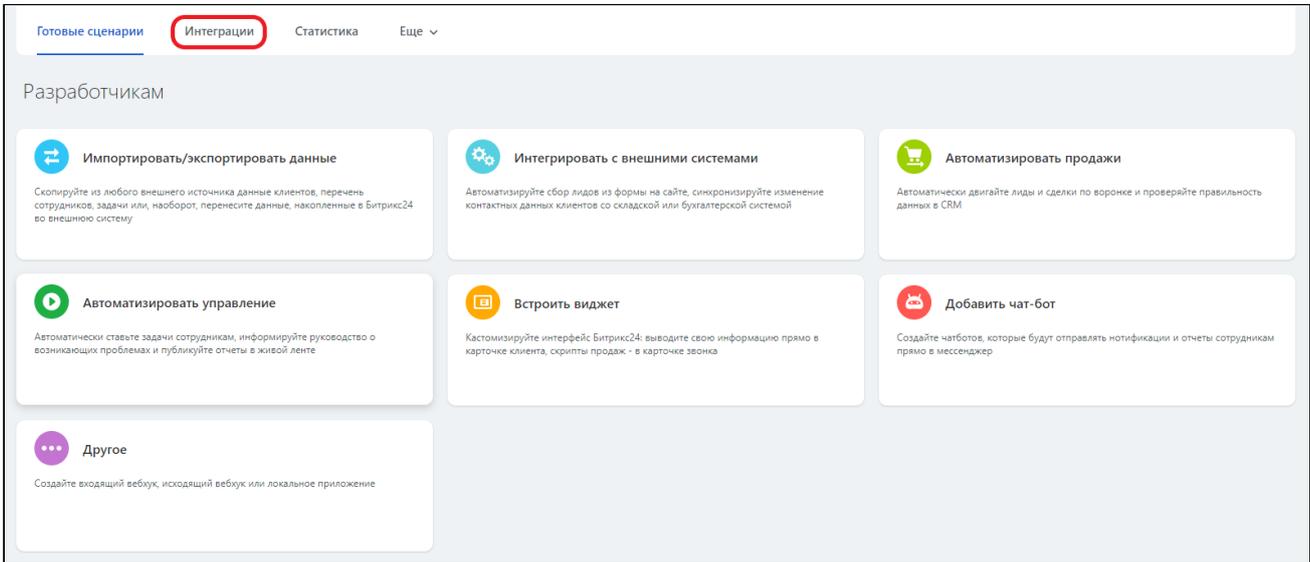
Все

ЗАКРЫТЬ

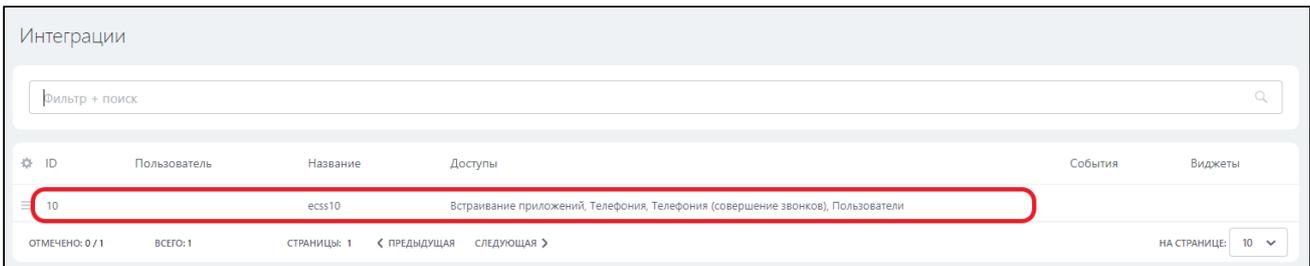
i Обратите внимание: Битрикс-24 игнорирует параметр OAuth2.0 redirect_uri и будет отдавать access_code на адрес, указанный вами на этой странице.

Приложение успешно создано.

6. Вернитесь к предыдущей странице и выберите вкладку «Интеграции».



7. Двойным кликом по строке выберите созданное приложение из предложенного списка.



8. Обратите внимание на значения «Код приложения (client id)» и «Ключ приложения (client secret)», а также на домен компании. Они понадобятся при регистрации компании в ecss-crm-server.

ecss10

Создайте приложение самостоятельно или закажите разработку у наших партнеров

Локальное приложение

Серверное Статичное

Путь вашего обработчика*

Путь для первоначальной установки

Код приложения (client_id)

Ключ приложения (client_secret)

Использует только API

Настройка прав

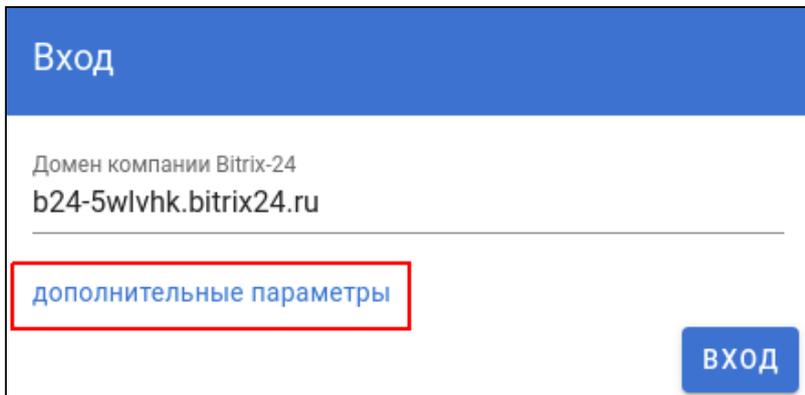
+ выбрать

Регистрация приложения в ecss-crm-server

⚠ Если ecss-crm-server уже установлен, перейдите в браузере по адресу, указанному вами при изначальной настройке. Обратите внимание, что адрес ecss-crm-server, должен совпадать с адресом, настроенным в интерфейсе Битрикс-24 при создании приложения (см. [раздел Ручная установка приложения в облачной версии Битрикс-24](#)).

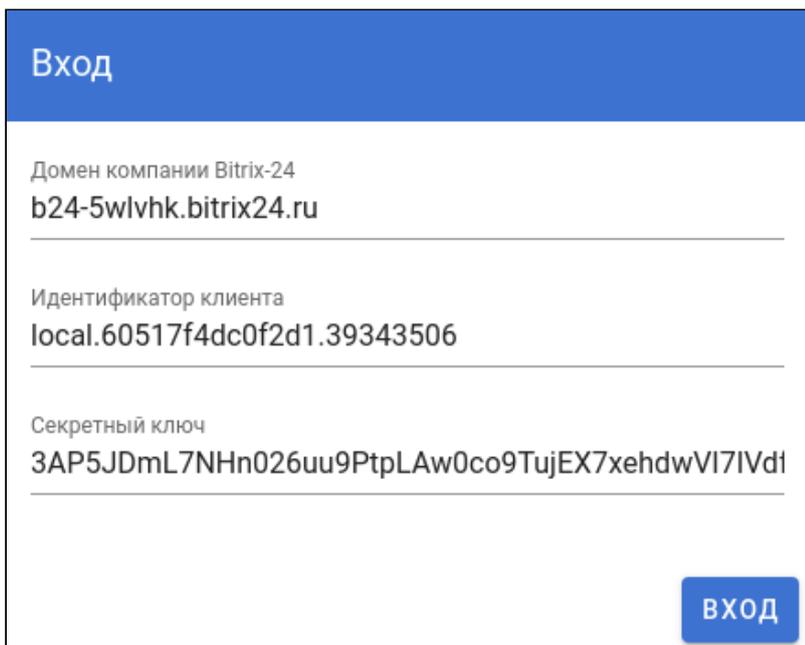
1. Для первого входа в приложение необходимо указать параметры «Код приложения (client id)», «Ключ приложения (client secret)» и домен компании (см. [раздел Ручная установка](#)

приложения в облачной версии Битрикс-24). Для этого необходимо нажать на кнопку **дополнительные параметры**.



The screenshot shows the Bitrix24 login interface. At the top, there is a blue header with the word 'Вход'. Below it, there is a text input field containing the domain 'b24-5wlvhk.bitrix24.ru'. Below the domain field, there is a button labeled 'дополнительные параметры' which is highlighted with a red rectangular border. To the right of this button is a blue button labeled 'ВХОД'.

2. В полях ввода укажите «Идентификатор клиента», что соответствует параметру «Код приложения» в интерфейсе Битрикс-24, и «Секретный ключ», что соответствует параметру «Ключ приложения» в интерфейсе Битрикс-24 (см. [раздел Ручная установка приложения в облачной версии Битрикс-24](#)). Указание этих дополнительных параметров обязательно только на этапе регистрации. В дальнейшем при входе нужно указывать только домен компании Битрикс-24.



The screenshot shows the Bitrix24 login interface with three input fields filled. The first field is the domain 'b24-5wlvhk.bitrix24.ru'. The second field is labeled 'Идентификатор клиента' and contains the value 'local.60517f4dc0f2d1.39343506'. The third field is labeled 'Секретный ключ' and contains the value '3AP5JDmL7NHn026uu9PtpLAW0co9TujEX7xehdwVI7IVdl'. A blue button labeled 'ВХОД' is visible at the bottom right.

⚠ Регистрация компании и последующее управление ей должны осуществляться пользователем с правами администратора компании Битрикс-24.

3. Нажмите кнопку **Вход**, браузер будет переадресован на страницу авторизации Битрикс-24.

После успешного прохождения авторизации, вы попадёте в интерфейс настройки компании ecss-crm-server.

Настройка компании в интерфейсе ecss-crm-server

1. В настройках компании присутствует возможность опционально задать имя компании.

Настройки компании

Домен: b24-5wlvhk.bitrix24.ru

Имя компании
ecss10 [ИЗМЕНИТЬ](#)

Настройки подключения к ECSS-10 ✓

Настройки подключения к Битрикс-24 ^

OAuth2.0 токен: **АВТОРИЗОВАН**

[ПОЛЬЗОВАТЕЛИ](#)

2. Укажите настройки подключения к ECSS-10, полученные у вашего провайдера телефонии.
3. Если вы являетесь администратором домена SSW, то вам необходимо выполнить команду:

```
domain<DOMAIN>integrations/list
```

Пример:

```
admin@[mycelium1@ecss1#ECSS 010145]:/$ domain/nick/integrations/list
```

Client id	Type	API Key
testbitrix	service	8SGuXEJUMs8SGqn6iAVqkMAuSqG1FM6JGkkUS2Ssg7vVuMrGUVqqku22iukqGvuX
Options	Numbers:	
107		
108		

4. Введите необходимые значения «API Key» и «Client id». Каждому значению соответствует указанное значение в приложении Битрикс-24:

- «Идентификатор клиента» --> «Client id»;
- «Ключ API» --> «API Key»;
- «Адрес Call API» --> «https://<IP>:8802»;
- «Домен» --> Используемый домен с интеграцией.

Настройки компании

Домен: b24-5wivhk.bitrix24.ru
Имя компании
ecss10 [ИЗМЕНИТЬ](#)

Настройки подключения к ECSS-10 ✓

Идентификатор клиента
testbitrix

Ключ API
8SGuXEJUMs8SGqn6iAVqkMAuSqGIFM6JGkkUS2Ssg7vVuMrGUVi

Адрес Call API
https://192.168.116.130:8802

Домен
nick

[ПРОВЕРИТЬ](#) Подключено

Настройки подключения к Битрикс-24 ▼

5. Перейдите на вкладку «Настройки подключения к Битрикс-24».

6. На данной вкладке отображается индикатор жизни OAuth2.0 токена (токен продлевается автоматически). Для настройки номеров телефонов пользователей нажмите на кнопку **пользователи**.

Настройки компании

Домен: b24-5wlvhk.bitrix24.ru
Имя компании
ecss10 ИЗМЕНИТЬ

Настройки подключения к ECSS-10 ✓

Настройки подключения к Битрикс-24 ^

OAuth2.0 токен: АВТОРИЗОВАН

ПОЛЬЗОВАТЕЛИ

7. В появившемся диалоговом окне можно настроить номера телефонов пользователей. Измененный список номеров сохраняется автоматически при закрытии диалогового окна. При поступлении вызова коннектор на основании данного списка определяет, какому пользователю необходимо показать карточку звонка в интерфейсе Битрикс-24.

Пользователи

virit76999@naymio.com
телефон
108

ЗАКРЫТЬ

При регистрации компании в ecss-crm-server список пользователей выгружается из аккаунта Битрикс-24. Сервер ecss-crm-server использует номера, указанные в профиле пользователя как «внутренний номер».

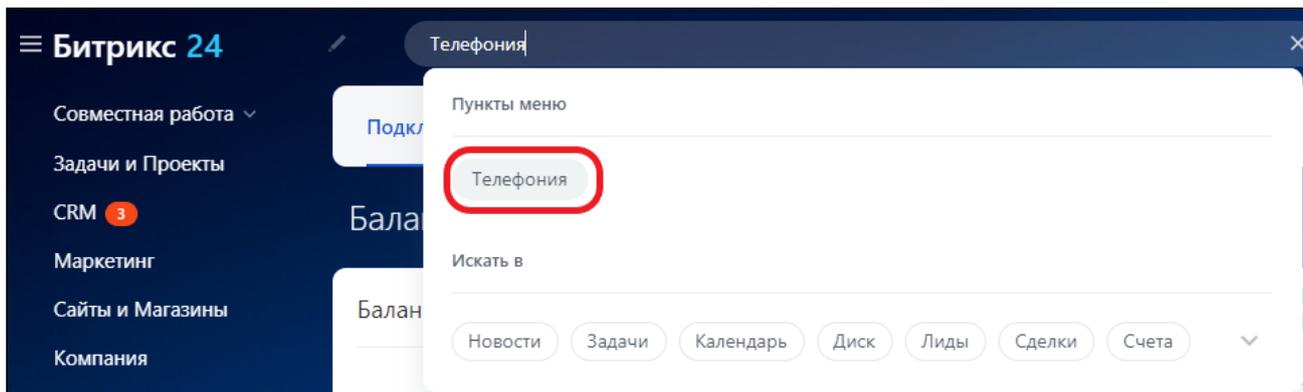
При добавлении нового пользователя в аккаунте компании Битрикс-24 он будет учтен ecss-crm-server при открытии списка пользователей либо при перезапуске сервера ecss-crm-server.

⚠ После того как пользователь Битрикс-24 учтен в ecss-crm-server, последующие изменения поля «внутренний номер» в аккаунте Битрикс-24, учитываться не будут.

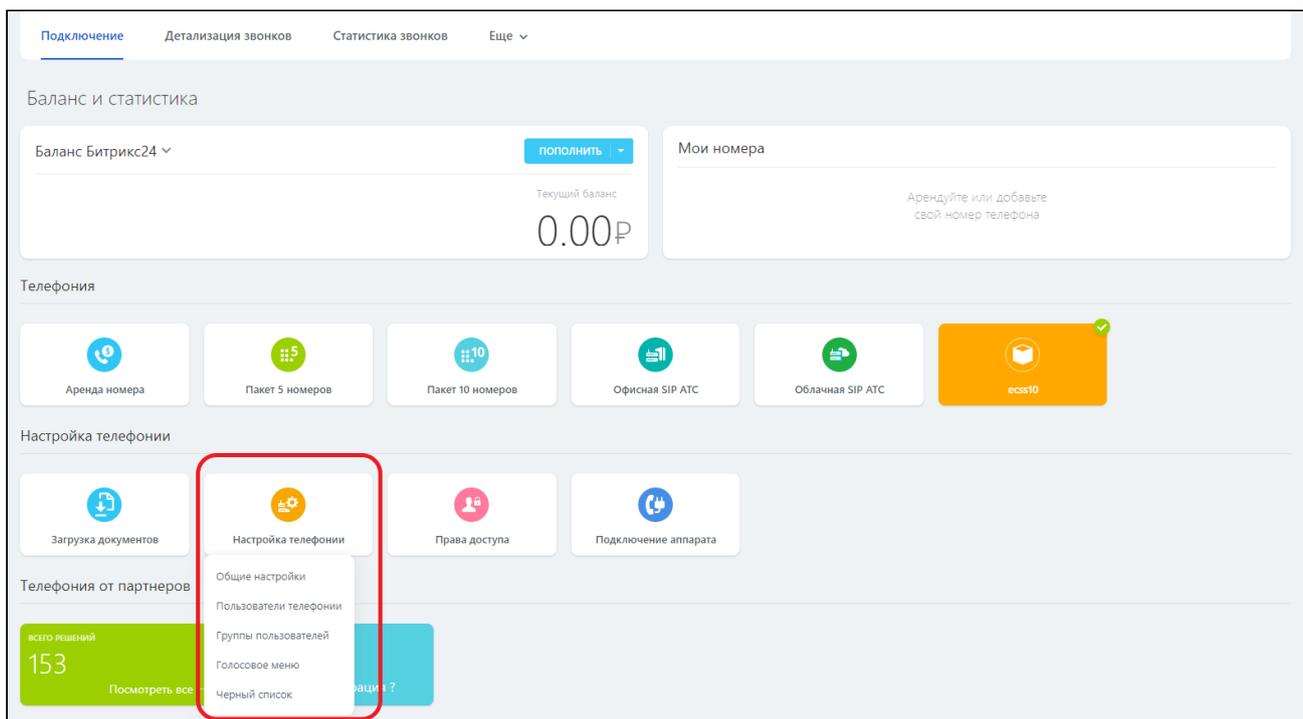
Активация click 2 call

Для включения click 2 call необходимо:

1. В интерфейсе компании в Битрикс-24 в поиске наберите «Телефония» и нажмите на пункт меню **Телефония**.



2. В открывшемся меню нажмите на вкладку **Настройка телефонии** и выберите пункт **Пользователи телефонии**.



3. Перейдите в настройке пользователя телефонии.

Пользователи телефонии		Фильтр + поиск		
ПОЛЬЗОВАТЕЛИ	ВНУТРЕННИЙ НОМЕР	НОМЕР ДЛЯ ИСХОДЯЩЕГО ЗВОНКА	ТЕЛЕФОННЫЙ АППАРАТ	
<div style="border: 1px solid red; padding: 2px;"> Настройка </div>	19@naimio.com	108	Приложение: ecss10	
ТЕЛЕФОННЫЙ АППАРАТ		Не подключен		

ВСЕГО: 1 СТРАНИЦЫ: 1 < ПРЕДЫДУЩАЯ СЛЕДУЮЩАЯ >

4. Выберите в поле «номер для исходящего звонка» созданное приложение.

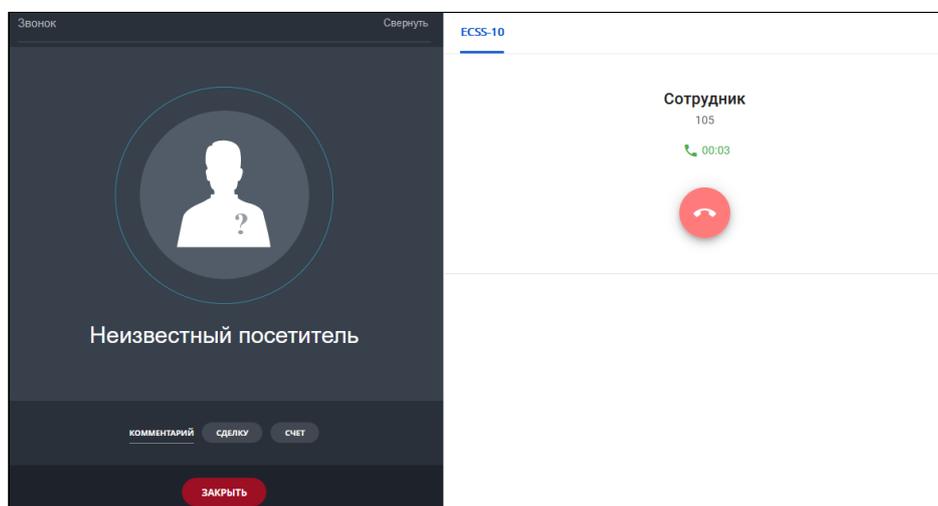
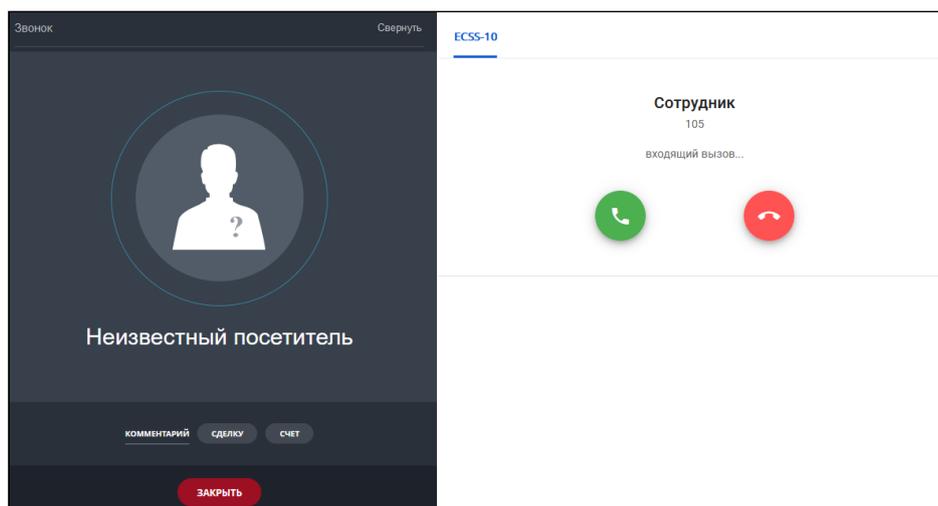
Внутренний номер

Номер для исходящего звонка

Телефонный аппарат

Управление вызовами в интерфейсе Битрикс-24

Если всё настроено правильно, то пользователь при входящем или исходящем вызове должен видеть карточку вызова. Левая часть карточки предоставляется самим Битрикс-24 и содержит интерфейс по управлению контактом. Правая часть карточки принадлежит интеграции с ECSS-10 и содержит номер звонящего (имя, отображаемое на телефонном аппарате) и кнопки управления вызовом: **Отклонить** и, если это входящий вызов, **Принять**.



Кнопка **Отклонить** работает для большинства телефонных аппаратов. Кнопка **Принять** работает для телефонных аппаратов, поддерживающих SIP-answer.

Интеграция с CRM Мегаллан

- [Общее описание](#)
- [Активация CSTA](#)
- [Настройка сертификата](#)
- [Управление интеграциями](#)
 - [1. Добавление интеграции](#)
 - [2. Удаление интеграции](#)
 - [3. Просмотр списка интеграций](#)
 - [4. Замена ключа интеграции](#)
- [Добавление разрешения абоненту SSW](#)
- [Установка виджета](#)
 - [Установка из магазина CRM Мегаллан](#)

- [Разрешение кросс-доменного запроса для локальных инсталляций](#)

В разделе приведено описание настроек взаимодействия ECSS-10 с системой для ведения проектов, работы с клиентами и настройки бизнес-процессов — [CRM Мегаплан](#).

Общее описание

Интеграция CRM Мегаплан с телефонией ECSS-10 осуществляется с помощью виджета, встраиваемого на веб-страницу CRM Мегаплан. Данный виджет обеспечивает работу таких функций как click to call, управление вызовами (принять/отклонить/отбить), список текущих активных вызовов, всплывающие нотификации, информирующие об активных вызовах и их фазах, а также определение контакта по номеру телефона и переход в детали контакта по клику на нотификации.

Для того чтобы виджет мог взаимодействовать с системой ECSS-10 посредством ECSS-10 Call API, необходимо:

- активировать CSTA в системе ECSS-10 (без этого виджет не будет отображать вызовы, инициированные не со страницы CRM Мегаплан);
- настроить сертификат для установки защищенного соединения между виджетом и системой ECSS-10 (см. [Интеграция с CRM Мегаплан](#));
- добавить в домене интеграцию для аутентификации виджета (см. [Добавление интеграции](#));
- добавить соответствующее разрешение для абонента, номер которого будет использоваться для осуществления вызовов из CRM Мегаплан (см. [Интеграция с CRM Мегаплан](#));
- установить виджет и указать настройки подключения к системе ECSS-10 (см. [Интеграция с CRM Мегаплан](#)).



Sorry, the widget is not supported in this export.
But you can reach it using the following URL:

<https://www.youtube.com/watch?v=I0mKtWMwFBg>

Активация CSTA

Для того чтобы виджет мог отображать вызовы, инициированные не со страницы CRM Мегаплан, нужно убедиться, что в системе активирован функционал [CSTA](#). Это можно сделать командой CoCon:

```
/api/csta/set enabled true
```

И в свойствах домена параметр `vats_type` должен быть выставлен в `private`:

```
admin@mycelium1@ecss1:/$ domain/nick/properties/info
```

Property	Domain	Value
...		
vats_type	nick	private

Настройка сертификата

При инсталляции пакета `ecss-user` автоматически генерируется самоподписанный сертификат `ecss10`. Опции `ssl_certificate` размещаются в `/etc/nginx/sites-available/ecss-crm-server.conf`:

```
server {
    ssl on;
    listen 8802 default_server;
    listen [::]:8802 default_server;
    resolver 127.0.0.1;
    ssl_certificate /etc/ecss/ssl/ecss10.crt;
    ssl_certificate_key /etc/ecss/ssl/ecss10.key;
    ...
}
```

Однако, современные браузеры не позволяют по умолчанию устанавливать защищенное соединение с ресурсами, предоставляющими самоподписанные сертификаты. Если в случае обычных страниц это не является серьёзной помехой, так как браузер предлагает добавить исключение безопасности при обращении к ресурсу, то в случае с виджетом ситуация несколько иная. Для работы виджета используются кросс-доменные запросы. В случае кросс-доменного запроса браузер блокирует запрос, не предлагая добавить исключение безопасности. В случае, если браузер блокирует кросс-доменный запрос, виджет отобразит следующую ошибку:

Идёт загрузка настроек виджета ECSS-10. Если данное сообщение отображается более нескольких секунд, значит есть проблемы соединения с API ECSS-10. Проверьте ещё раз введенный адрес. Если адрес указан правильно, но проблема не исчезла, обратитесь в техподдержку виджета ECSS-10, или воспользуйтесь документацией.

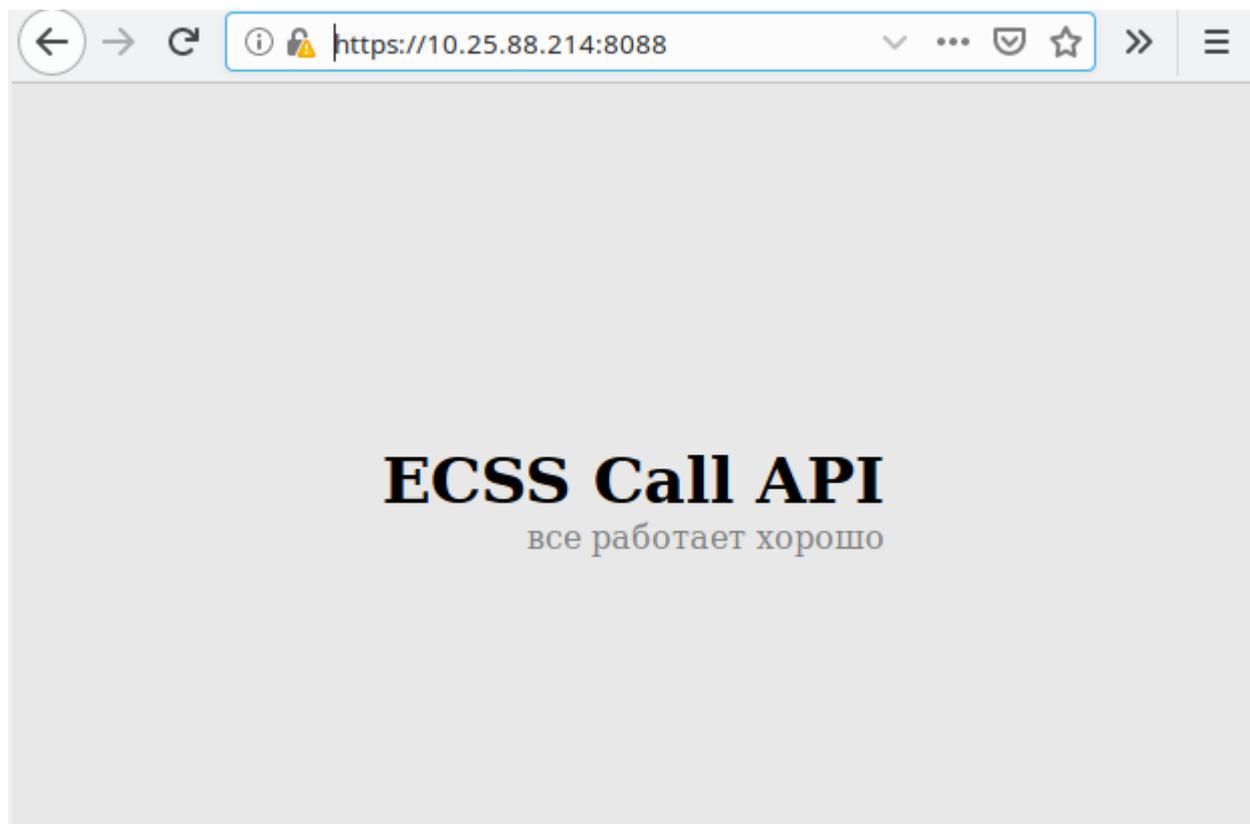
Для того чтобы можно было добавить исключение безопасности для самоподписанного сертификата, существует страница при переходе на которую браузер предложит добавить исключение безопасности. Страница по умолчанию доступна по адресу, на котором был

установлен пакет `ecss-node`. Порт 8802. Таким образом в адресной строке браузера нужно ввести:

```
https://<host>:8802/
```

где `<host>` — имя или адрес ECSS-10.

После добавления исключения безопасности откроется тестовая страница ECSS Call API:



Если страница будет недоступна, то необходимо проверить состояние сервиса `ecss-call-api`:

```
sudo systemctl status ecss-crm-server.service
```

При необходимости включить его:

```
sudo systemctl start ecss-crm-server.service
```

Управление интеграциями

Интеграция представляет собой пару, состоящую из идентификатора клиента и API ключа. Интеграции используются для аутентификации сторонних сервисов, использующих ECSS Call

API. Команды управления интеграциями приведены в разделе "[Команды управления ключами внешних интеграций](#)".

1. Добавление интеграции

Для добавления интеграции используется команда:

```
/domain/<DOMAIN>/integrations/add <CLIENT_ID>
```

Добавить новую интеграцию. API ключ будет сгенерирован автоматически. Для одного клиента, представленного параметром <CLIENT_ID>, может быть добавлена только одна интеграция

<CLIENT_ID> — идентификатор клиента, для которого будет создан ключ интеграции

Пример:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/integrations/add tester
New integration was added.
API key: 2hBqT4acTLhcvFq29qw429TF26cfv9Wqkzn4Ga4chYC2hyhT2WkdknnWC94dFTKC

[exec at: 22.12.2019 10:12:38, exec time: 28ms, nodes: ds1@ecss1]
```

 Созданная интеграции должна использоваться строго для одной crm системы

2. Удаление интеграции

Для удаления интеграции используется команда:

```
/domain/<DOMAIN>/integrations/remove <CLIENT_ID>
```

Удалить интеграцию

<CLIENT_ID> — идентификатор клиента, для которого будет удалён ключ интеграции

```
admin@mycelium1@ecss1:/$ domain/biysk.local/integrations/remove tester4
ok

[exec at: 22.12.2019 13:20:18, exec time: 16ms, nodes: ds1@ecss1]
```

3. Просмотр списка интеграций

Для просмотра списка интеграций используется команда:

```
/domain/<DOMAIN>/integrations/list
```

Просмотреть список интеграций

Список интеграций показывается в виде таблицы, пример:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/integrations/list
```

Client id	API Key
asz	2uKwuT5qxKB21uzPuTb518oxDB2qzuKLebfT17orD0G21zuuLL0efu1D06rrKcBZ
sveta	4lCyT0DH1wfcG41yCytSUDCcfffvWANlyCUL4xcDH4cfyfcWvPNylyCUM1ycOD1H
tester	2hBqT4acTLhcvFq29qW429TF26cfv9Wqkzn4Ga4chYC2hyhT2WkdknnWC94dFTKC

```
[exec at: 22.12.2019 10:14:13, exec time: 12ms, nodes: ds1@ecss1]
```

4. Замена ключа интеграции

Для замены ключа интеграции используется команда:

```
/domain/<DOMAIN>/integrations/replace <CLIENT_ID>
```

Заменить ключ интеграции

<CLIENT_ID> — идентификатор клиента, для которого будет заменён ключ интеграции.

Пример работы команды:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/integrations/replace tester
```

```
Old key was replaced with new one.
```

```
New API key: wT5neQJ1hykwvnTQeJ50ywKoFMizeAJcJwhcowFvV5eVe5zf5C1imCvoMFizAKQz
```

```
[exec at: 22.12.2019 10:15:31, exec time: 28ms, nodes: ds1@ecss1]
```

Добавление разрешения абоненту SSW

При выполнении логина в ECSS Call API виджет сообщает ECSS-10 номер телефона абонента, который будет закреплён за этим виджетом. Это требует специального разрешения, активированного у абонента.

Для активации разрешения нужно выставить соответствующему алиасу значение true для свойства api\call\enabled. Например:

Пример:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/alias/set 240473 loc.gr
240473@biysk.local api\call\enabled true
There are aliases within domain biysk.local affected by settings property
api\call\enabled:
  240473 <-> 062743c44271f9e7
```

Legend:

- a: Setup alias address
- i: Setup alias interface
- !: Setup broken. Alias already exists
- *: Setup broken. Alias not exists
- L: Setup broken. No free subscriber licences.
- x: Setup broken. Unexpected error
- <empty>: Successful setup

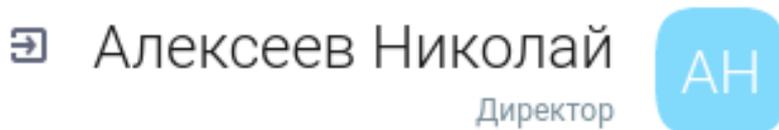
[exec at: 20.12.2019 22:42:09, exec time: 101ms, nodes: ds1@ecss2]

В том случае, если данное свойство не будет выставлено, виджет не сможет выполнить логин в ECSS Call API

Установка виджета

Установка из магазина CRM Мегатлан

Магазин виджетов CRM Мегатлан можно найти в разделе **Профиль -> Расширения**.



-  Профиль
-  Новости
-  Настройки
-  Расширения
-  Помощь

 QR-код для входа в мобильное приложение

Далее в магазине необходимо найти виджет **Телефония ECSS-10**:

<p>телеком услуги.</p> <p>УСТАНОВИТЬ</p>	<p>УСТАНОВИТЬ</p>	<p>менеджером, товаром и данными в дополнительных полях. Пригодится для периодических платежей, чтобы экономить время и правильно учитывать поступления в отчетном периоде.</p> <p>УСТАНОВИТЬ</p>
 <p>Энфорта Оператор, предоставляющий услуги связи на базе беспроводных технологий.</p> <p>УСТАНОВИТЬ</p>	 <p>Деловая сеть - Иркутск Доступ к сети интернет, телефония, создание сетей, регистрация доменов и пр. Тарифы для корпоративных клиентов.</p> <p>УСТАНОВИТЬ</p>	 <p>Телефония Телфин Телфин, виртуальная АТС с широкими настройками, гибкие решения и профессиональная поддержка.</p> <p>УСТАНОВИТЬ</p>
 <p>Инитпрокасса Приложение Инитпрокасса реагирует на изменение статуса оплаты счета и отправляет данные об оплате на сервис Инитпрокасса для фискализации на подключенном оборудовании. После фискализации электронный чек будет отправлен покупателю.</p> <p>УСТАНОВИТЬ</p>	 <p>Телефония ECSS-10 Телефония ECSS-10 для Мегаллан</p> <p>УСТАНОВИТЬ</p>	 <p>PostMonitor С помощью этого приложения вы можете выгрузить данные о клиентах из Мегаллана в аудиторию рекламатента соцсетей, а также автоматически собирать данные из лид-форм соцсетей и передавать их в Мегаллан.</p> <p>ПОДРОБНЕЕ</p>

Для инсталляции виджета необходимо кликнуть на его изображении в магазине.

В открывшемся окне отображается краткая информация о виджете. Для того чтобы перейти к установке, необходимо кликнуть на кнопку установить:

Установка приложения Телефония ECSS-10 для Мегаплан

Описание:

Телефония ECSS-10 для Мегаплан

После установки из магазина, виджет необходимо настроить.

В разделе **Настройки/Интеграция/Телефония/Настройки телефонии/Основное** должен появиться отдельный пункт меню **Параметры подключения к ECSS-10**.

Параметры подключения вы можете узнать у своего провайдера телефонии.

Также необходимо чтобы у пользователя была настроена телефония в разделе

Настройки/Интеграция/Телефония/Настройки телефонии/Пользователи и указан внутренний номер.

Если настройка была выполнена правильно, иконка виджета в левом нижнем углу окна Мегаплан должна стать активной.

Наши специалисты готовы вам помочь:

(383) 272-83-31, techsupp@eltex-co.ru

Виджеты:

Приложение использует виджеты в следующих плейсхолдерах:

- common_telephony_widget
- settings_api_telephony

 Приложение будет иметь полный доступ к данным Вашего аккаунта

УСТАНОВИТЬ

ОТМЕНИТЬ УСТАНОВКУ

В настройках виджета необходимо указать:

- сгенерированные ранее идентификатор клиента и ключ API (см. раздел "[Интеграция с CRM Мегаплан](#)").
- IP-адрес и порт ECSS Call API. Адрес и порт определяется конкретной инсталляцией ECSS-10, однако по умолчанию ECSS Call API устанавливается вместе с пакетом ecss-node и доступно по 0.0.0.0 и порте 8088 на том же самом хосте.
- имя домена (виртуальной АТС) ECSS-10.

Необходимо заполнить предложенные поля и нажать сохранить:

Параметры подключения к ECSS-10

Адрес Eltex ECSS-10 API:

192.168.116.168:8802

Идентификатор клиента:

megaplan

Ключ API:

4TFZrvFB1r7sbA4i3QFTZ67fE

Домен SSW:

nick

Сохранить

Также, необходимо указать сотрудников, которые будут пользоваться данным виджетом. Для этого необходимо кликнуть на поле "Пользователи"

[Телефония](#) → Телефония API

Телефония API Бета

Основное

Пользователи

Настройки сервера

1 сотрудник



Подключение к телефонии по API

Ваш провайдер телефонии или вы самостоятельно можете реализовать взаимодействие с Мегапланом по API.

[Документация](#)

✔ Включено

Выключить

Выбрать сотрудников, которые будут работать с виджетом. В поле "Внутренний номер" необходимо указать номер телефона абонента ECSS-10 . Нажать "Сохранить".

Настроить учетные записи пользователей

 Укажите сотрудников, у которых должна работать телефония в Мегеплане

Сотрудник	Внутренний номер	Удалить
Алексеев Николай:	<input type="text" value="3025"/>	

 Добавить сотрудника

Теперь можно перезагрузить страницу. Если условия, указанные выше в данной документации, соблюдены и параметры виджета заполнены верно, то виджет будет готов к работе.

Разрешение кросс-доменного запроса для локальных инсталляций

В случае, если используется локальная инсталляция Мегеплан (а не сайт megaplan.ru), то необходимо добавить разрешение кросс-доменного запроса от данной инсталляции в конфиг `nginx ecss-call-api.conf`. По умолчанию, строки, отвечающие за обработку кросс-доменных запросов, выглядят так:

```

...

location / {
    # nginx не поддерживает вложенные if, а также более одного условия
    # в рамках одного выражения if, поэтому нам приходится реализовать
    # обработку кросс-домменных запросов в виде lua-модуля
    header_filter_by_lua_block {
        origin = ngx.var.http_origin
        if (origin == nil) then
            -- если origin не заполнен, ничего не делаем, просто проксируем
        elseif (
            origin:match("https://.*%.amocrm%.ru")
            or origin:match("https://.*%.megaplan%.ru")
            -- другие источники (origin) можно добавить здесь. Например:
            -- or origin:match("https://mycrm.example.com")
        ) then
            ngx.header["Access-Control-Allow-Origin"] = origin
            ngx.header["Access-Control-Allow-Headers"] = "content-type"
            ngx.header["Access-Control-Allow-Credentials"] = "true"
        else
            ngx.exit(ngx.HTTP_FORBIDDEN)
        end
    end
}

...

```

Допустим, Мегалплан установлен локально по адресу 192.168.1.2. В этом случае после строчки

```
or origin:match("https://.*%.megaplan%.ru")
```

Нужно добавить строчку

```
or origin:match("https://192%.168%.1%.2")
```

И перезапустить nginx

```
sudo systemctl restart nginx
```

Интеграция с сервисом Skype for business (S4B)

- [Общее описание](#)
- [Пример настройки](#)
 - [Настройка клиента](#)
 - [Настройка сервиса на ECSS-10](#)

- [Настройка телефонного аппарата](#)

В разделе приведено описание настроек взаимодействия ECSS-10 с сервисом Skype for business.

Общее описание

ECSS-10 имеет возможность подписываться на учетные записи сервиса Skype for business (S4B), что позволяет получать BLF/PRESENCE нотификации на телефонный аппарат.

Пример настройки

В сервисе S4B заведена учетная запись с данными параметрами:

 Email — test.eltex@eltex-co.ru
Username — resource\Eltex-test
Пароль — eltex_pass

Настройка клиента

В качестве примера будет использоваться клиент S4B на операционной системе IOS (Apple).

1. На стартовой странице вводим в поле "Адрес для входа организации" наш Email

12:50



Вход

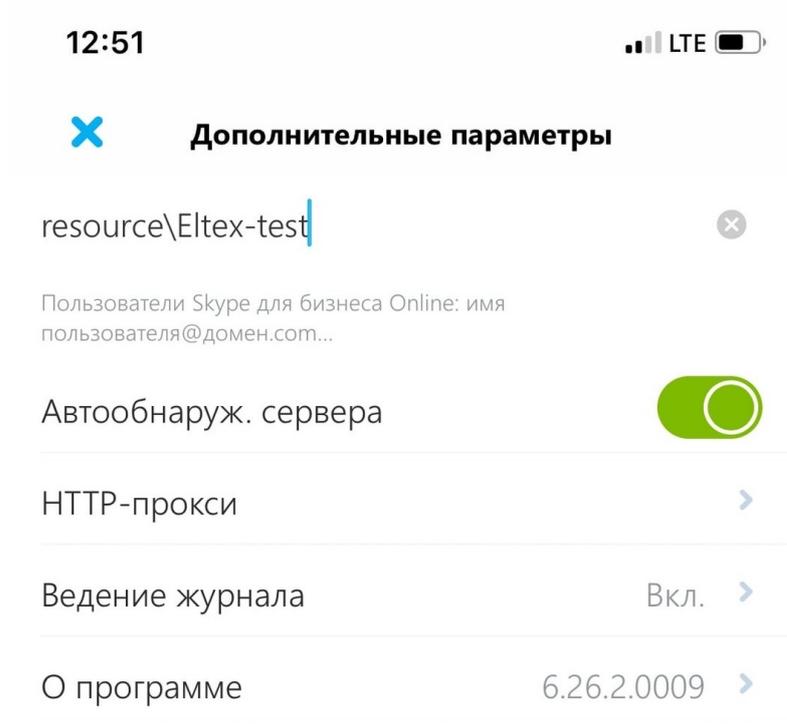
test.eltex@eltex-co.ru

Дополнительные параметры



Помощь по входу

2. Переходим во вкладку "Дополнительные параметры" и вводим в поле "Пользователь" наш Username



3. Выходим из вкладки "Дополнительные параметры" на главную страницу и вводим пароль в появившемся окне

13:21 LTE



Вход

test.eltex@eltex-co.ru

Сохранить пароль

[Дополнительные параметры](#)

Вход

[Помощь по входу](#)

4. Входим в учетную запись

После проделанных манипуляций, можно пользоваться сервисом S4B.

Настройка сервиса на ECSS-10

Для корректной работы сервиса требуется создать абонента, который будет принимать нотификации от S4B, а также на данного абонента можно будет подписаться с телефонного аппарата при помощи BLF/PRESENCE.

1. Создание абонента

```

admin@[ds1@ecss1#ECSS 010070]:/$ domain/test/sip/user/declare default_routing sip.ab 001@te
st alias-as-user no_qop_authentication login-as-user 123
Executed on the sip1@ecss1
Intermediate (incomplete) result:
Declaration for range: 001@test..001@test (1)
...
1 interfaces check for existing ...
[*****] 80mks
1 users interfaces declaration ...
[*****] 7ms
1 users aliases declaration ...
[*****] 60ms
230 interfaces recall to base
[*****] 99ms
Executed on the sip1@ecss1

| declared 1 subscribers |

[exec at: 19.05.2020 13:40:48, exec time: 302ms, nodes: sip1@ecss1]

```

2. Настройка параметров S4B на алиасе:

```

admin@[ds1@ecss1#ECSS 010070]:/$ domain/test/alias/set 001 sip.ab 001@test s4b\email
test.eltex@eltex-co.ru
There are aliases within domain test affected by settings property s4b\email:
001 <-> 0635d482fe60bc0c

Legend:
  a: Setup alias address
  i: Setup alias inteface
  !: Setup broken. Alias allready exists
  *: Setup broken. Alias not exists
  L: Setup broken. No free subscriber licences.
  x: Setup broken. Unexpected error
  <empty>: Successfull setup

[exec at: 19.05.2020 13:42:36, exec time: 68ms, nodes: ds1@ecss1]

```

```
admin@[ds1@ecss1#ECSS 010070]:/$ domain/test/alias/set 001 sip.ab 001@test s4b\username
resource\Eltex-test
```

There are aliases within domain test affected by settings property s4b\username:

```
001 <-> 0635d482fe60bc0c
```

Legend:

```
a: Setup alias address
i: Setup alias inteface
!: Setup broken. Alias allready exists
*: Setup broken. Alias not exists
L: Setup broken. No free subscriber licences.
x: Setup broken. Unexpected error
<empty>: Successfull setup
```

```
[exec at: 19.05.2020 13:43:31, exec time: 66ms, nodes: ds1@ecss1]
```

```
admin@[ds1@ecss1#ECSS 010070]:/$ domain/test/alias/set 001 sip.ab 001@test s4b\password
eltex_pass
```

There are aliases within domain test affected by settings property s4b\password:

```
001 <-> 0635d482fe60bc0c
```

Legend:

```
a: Setup alias address
i: Setup alias inteface
!: Setup broken. Alias allready exists
*: Setup broken. Alias not exists
L: Setup broken. No free subscriber licences.
x: Setup broken. Unexpected error
<empty>: Successfull setup
```

```
[exec at: 19.05.2020 13:44:22, exec time: 69ms, nodes: ds1@ecss1]
```

3. Также дополнительно можно настроить взаимодействие статусов в S4B с активацией услуги DND

Так при выставлении данного значения:

```
admin@[ds1@ecss1#ECSS 010070]:/$ domain/test/ss/s4b-presence/activate 001 --s4b-presence-
activate-dnd-list BeRightBack
```

```
Success: Supplementary service s4b_presence activated for domain "test" address "001"
```

При выставлении в S4B статуса "Скоро вернусь" на номере 001 будет автоматически активирована услуга DND (Услуга заранее должна быть включена на номере 001). Для того, чтобы данная услуга активировалась и на телефоне на нем должен быть настроен as-feature event. Более подробно о настройке написано здесь [Руководство по настройке as-feature-event](#).

Таким же образом можно настроить обратную совместимость. При выставлении на телефоне DND в S4B будет выставлен статус Busy.

```
admin@[ds1@ecss1#ECSS 010070]:/$ domain/test/ss/s4b-presence/activate 001 --dnd-map-to-s4b-presence Busy
Success: Supplementary service s4b_presence activated for domain "test" address "001"
```

4. Также опционально можно отключить отправку уведомлений о смене статуса в S4B. Для этого требуется выставить опцию blf-enable в false:

```
admin@[ds1@ecss1#ECSS 010070]:/$ domain/test/ss/s4b-presence/activate 001 --blf-enable false
Success: Supplementary service s4b_presence activated for domain "test" address "001"
```

После проделанных действий, ECSS-10 получит статус клиента S4B. Для проверки требуется ввести команду

```
admin@[ds1@ecss1#ECSS 010070]:/$ domain/test/s4b/presence/info
```

Address	Presence
001	Online

```
[exec at: 19.05.2020 13:44:26, exec time: 69ms, nodes: ds1@ecss1]
```

Теперь при изменении статуса клиента S4B, будет изменяться статус в команде.

Настройка телефонного аппарата

В качестве телефонного аппарата будет использоваться Yealink T21E2.

Телефонный аппарат требуется подписать на номер 001. Подробная инструкция по настройке BLF описана в данном разделе [Настройка SIP-TA Yealink T21E2](#).

После успешной подписки, на телефонный аппарат Yealink T21E2 начнут приходить уведомления о статусе клиента Skype for business.

Интеграции Call-центра

- [Интеграция с Redmine](#)
- [Интеграция с КИС\(корпоративной информационной системой\)](#)
- [Интеграция с СГС \(системой голосового самообслуживания\)](#)

В разделе приведены описания интеграции Call-центра с внешними системами.

Web-клиент рабочего места оператора поддерживает интеграцию с системами Redmine, корпоративной информационной системой (КИС) и системой голосового самообслуживания (СГС). В случае настроенной интеграции web-клиент будет отображать страницу в специальной области и автоматически выполнять вход в одну из этих систем. Подробнее о настройке см. [команды управления агентами в Call-центре](#) (параметры `integration*`).

Активация интеграции и общие URL задаются при установке или реконфигурации пакета `ecss-cc-ui`.

```
sudo dpkg-reconfigure ecss-cc-ui
```

Вопросы, задаваемые при установке пакета `ecss-cc-ui` приведены в разделе "[Приложение Д. Вопросы, задаваемые при инсталляции пакетов ECSS](#)".

Xsd схема HTTP-API получения списка интеграций оператора приведена на странице [integration_list.xsd](#).

Интеграция с Redmine

Интеграция КЦ ECSS-10 с Redmine позволяет при принятии нового вызова в КЦ открывать страницу Redmine с новой, предзаполненной по определенному шаблону задачей. Для этого в настройках агента КЦ необходимо заполнить следующие поля:

Таблица 1. Параметры агента

Параметр	Значение	Описание
<code>integration\redmine</code>	<code><Login> <Password></code>	учетные данные пользователя Redmine: <code><Login></code> - имя учетной записи; <code><Password></code> - пароль учетной записи
<code>integration\redmine\task\project</code>	string, по умолчанию не задано	имя Redmine проекта по умолчанию. В рамках указанного проекта будут создаваться задачи при поступлении вызова в КЦ

Параметр	Значение	Описание
integration\redmine\task\subject	string, по умолчанию не задано	шаблон темы создаваемой задачи в Redmine при поступлении вызова в КЦ. Поддерживает макро-переменные (макро-переменные указываются через процент, например %CGPN.digits%). Возможные макро-переменные: <ul style="list-style-type: none"> • CGPN.digits - номер вызывающего (А) абонента; • CDPN.digits - номер вызываемого (Б) абонента; • CGPN.display_name - имя вызывающего абонента; • CDPN.display_name - имя вызываемого абонента; • DATE - текущая дата; • TIME - текущее время; • AGENT_ID - идентификатор агента КЦ, на которого распределился вызов; • QUEUE_ID - имя очереди, через которую вызов распределился на агента КЦ (если вызов пришел напрямую на агента КЦ, то данное поле будет пустым); • WORKITEM_ID - внутренний идентификатор заявки в КЦ
integration\redmine\task\body	string, по умолчанию не задано	шаблон тела создаваемой задачи в Redmine при поступлении вызова в КЦ. Поддерживает макро-переменные аналогичные тем, что используются для переменной integration\redmine\task\subject

Параметры выставляются командой [domain/<DOMAIN>/cc/agent/set](#).

Пример шаблона:

Входящий вызов с номера %CGPN% на номер %CDPN%. Время: %DATE% %TIME%. \n
Идентификатор происшествия: %WORKITEM_ID%.

Если WORKITEM_ID не был задан внешней системой, то его нужно генерировать внутри. Для этого необходимо активировать опцию [domain/<DOMAIN>/cc/properties/set generate_workitem_id_by_queue true](#).

Интеграция с КИС(корпоративной информационной системой)

Активация интеграции с КИС и домашняя страница интегрируемой системы также задаются при установке или реконфигурации пакета ecss-cc-ui.

Настройка включает два этапа:

- При установке пакет спрашивает основной URL интегрируемой информационной системы, например <http://mis.nso.ru>. Тип протокола(http или https) важен, в зависимости от этого будут сформированы необходимые настройки nginx, и config.yaml при этом настройка попадет в виде булевого значения, что сервис активирован.
- Необходимо настроить для агента через CLI "остаток" URL, где указано все что после host:port, включая слеш(/). Например, если оператору при поступлении вызова требуется открыть URL подобного вида: http://mis.nso.ru/?page=CALL_CENTER&CALL_NUMBER=3832969921&CALL_ID=sdfs, тогда при настройке пакета URL будет <http://mis.nso.ru>, а для агента CIS Request URI :

```
/?page=CALL_CENTER&CALL_NUMBER=%CDPN.digits%&CALL_ID=%CALL_REF%
```

CDPN.digits и CALL_REF заменится на данный из вызова. Для конкретного оператора параметры выставляются командой `domain/<DOMAIN>/cc/agent/set integration\cis`.

При логине КИС открывается только в том случае, если нужно предварительно ввести логин. Потом ее нужно закрыть самостоятельно. КИС для оператора по конкретной для него ссылке открывается по звонку. В АРМ оператора КЦ есть кнопка "Открыть КИС" Кнопка сделана для того чтобы можно было заранее открыть КИС и пройти авторизацию, и не делать это при ответе на вызов.

Конфигурация находится в файлах:

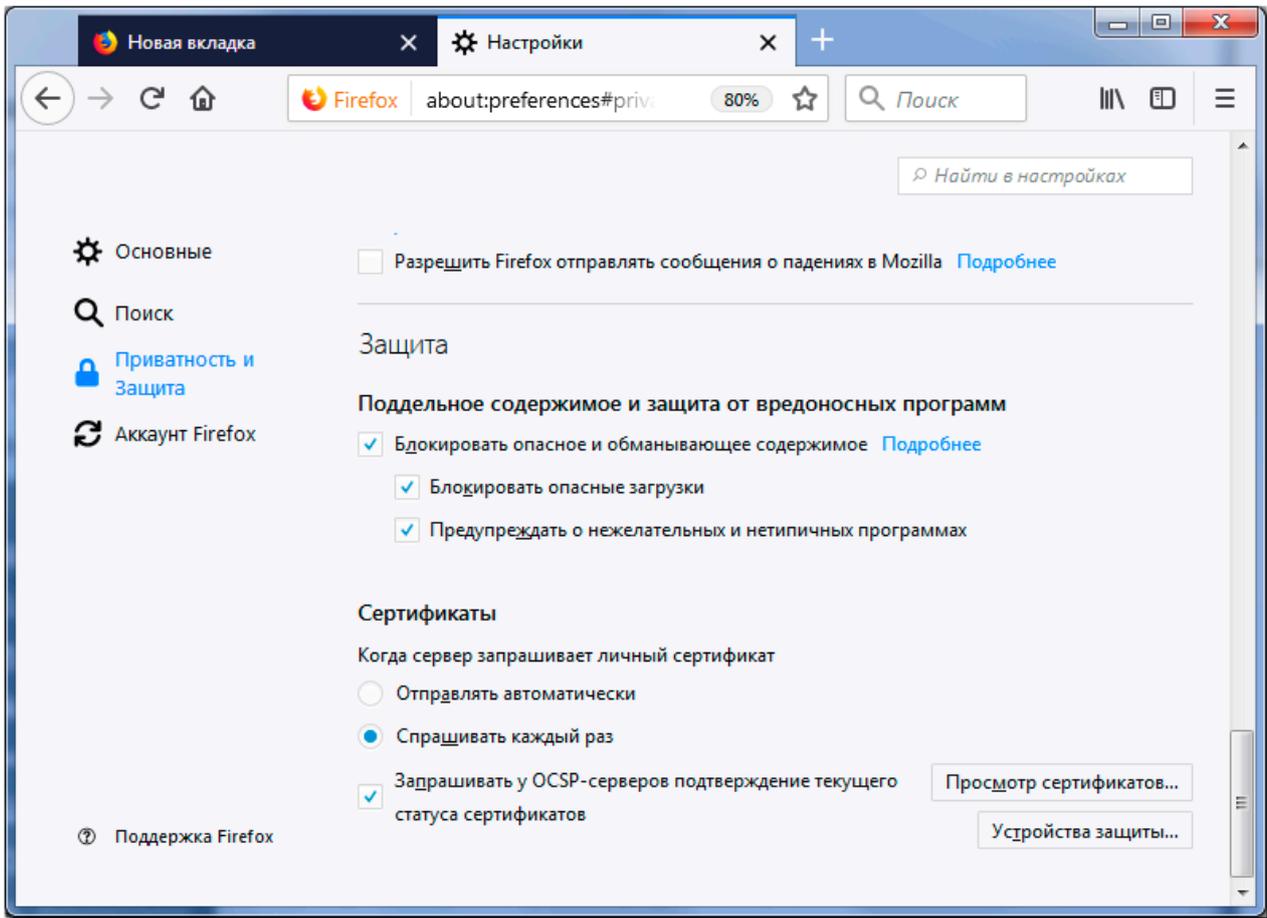
```
/etc/ecss/ecss-cc-ui-api/config.yaml
```

```
/usr/lib/ecss/ecss-cc-ui/cis-proxy.conf;
```

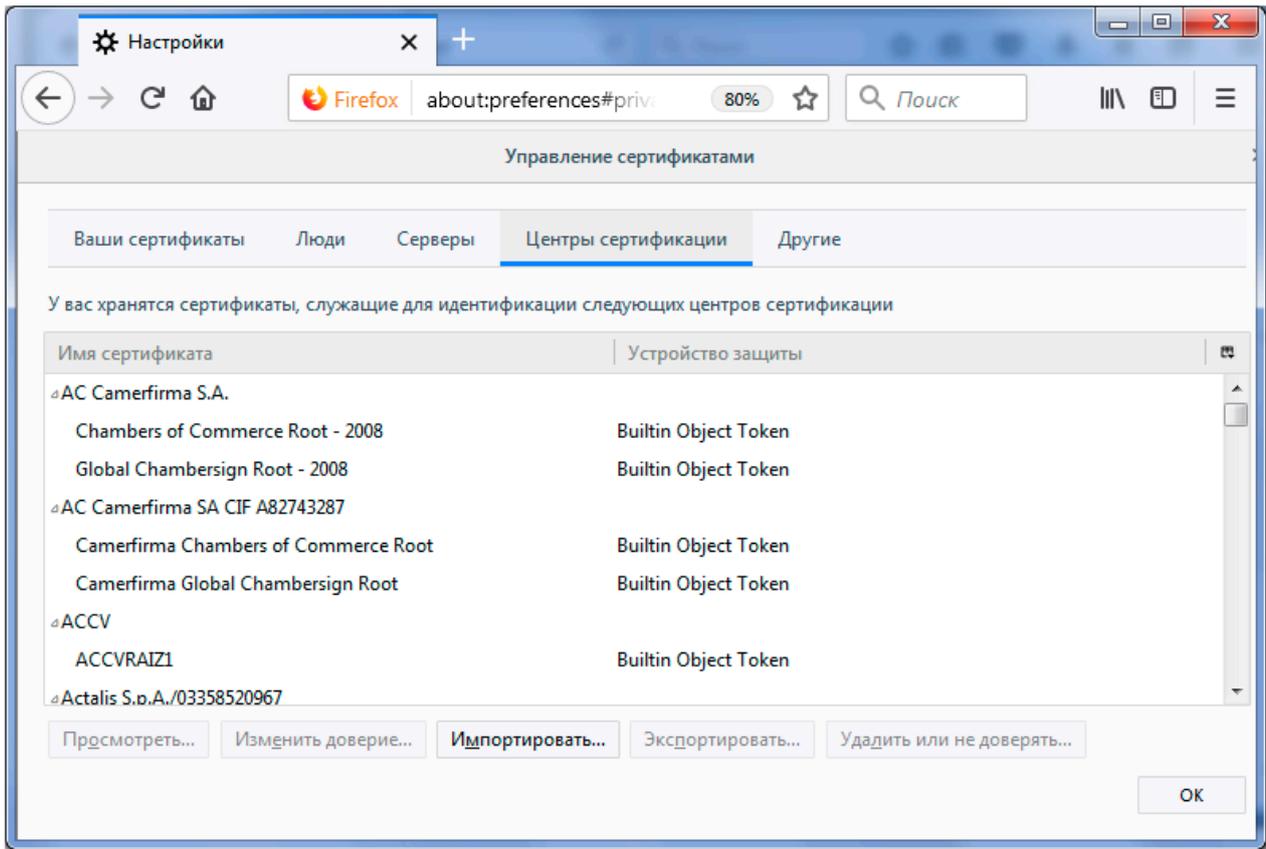
Есть проблема с открытием фрейма https-страницы КИС в браузере Mozilla Firefox - невозможно добавить исключения безопасности . Обойти проблему можно если открыть в новой вкладке адрес прокси IP:8071 и тогда появится возможность применить сертификат.

Если имеется файл [сертификата](#), его можно импортировать в браузер. Ниже показан пример импорта корневого сертификата Элтекс.

Для этого откройте меню *Настройки*. Перейдите в раздел *Приватность и защита - Сертификаты* и нажмите кнопку *"Просмотр сертификатов"*.

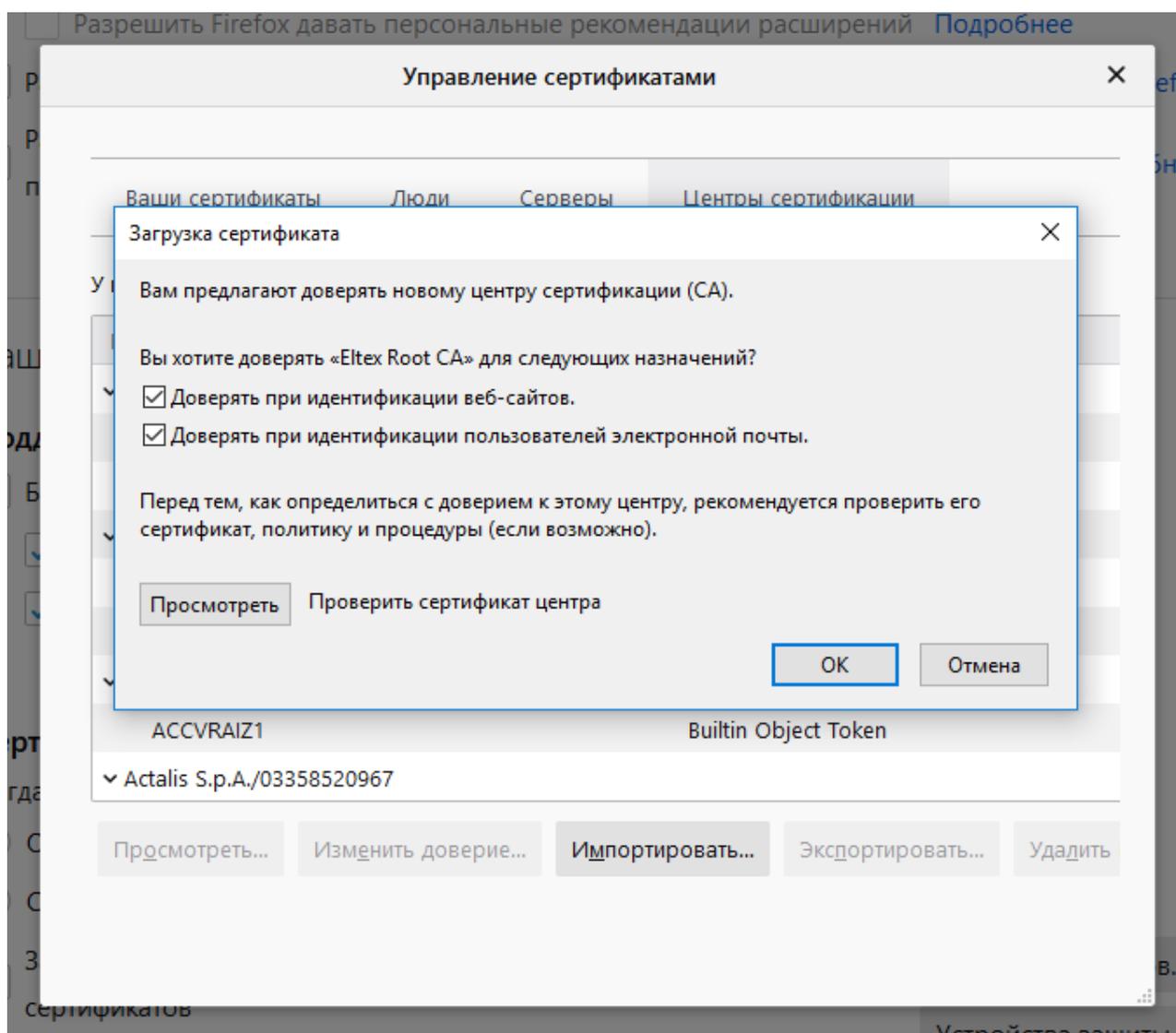


В окне *Менеджера сертификатов* выберите вкладку *Центры сертификации* и нажмите кнопку *"Импортировать..."*



Найдите на жестком диске сохраненный файл сертификата и нажмите кнопку "Открыть".

В окне Загрузка сертификата необходимо выбрать цели, для которых вы импортируете сертификат.



Интеграция с СГС (системой голосового самообслуживания)

По аналогии с КИС Call-центр ECSS-10 возможно также интегрировать с системой голосового самообслуживания(СГС).

При установке пакета также вводится URL в формате `http(s)://host:[port]`.

Для оператора Call-центра параметры выставляются командой `domain/<DOMAIN>/cc/agent/set integration\ssvs`. Например, если для оператора требуется открыть URL вида:

`http://10.0.41.25/api/dialog/transcribe?interaction_id=WORKITEM_ID&key=84eae956ba0a4847752345cb7411af825fcb3cb6d1c05e7e0fbe70b14d1a8c4f,`

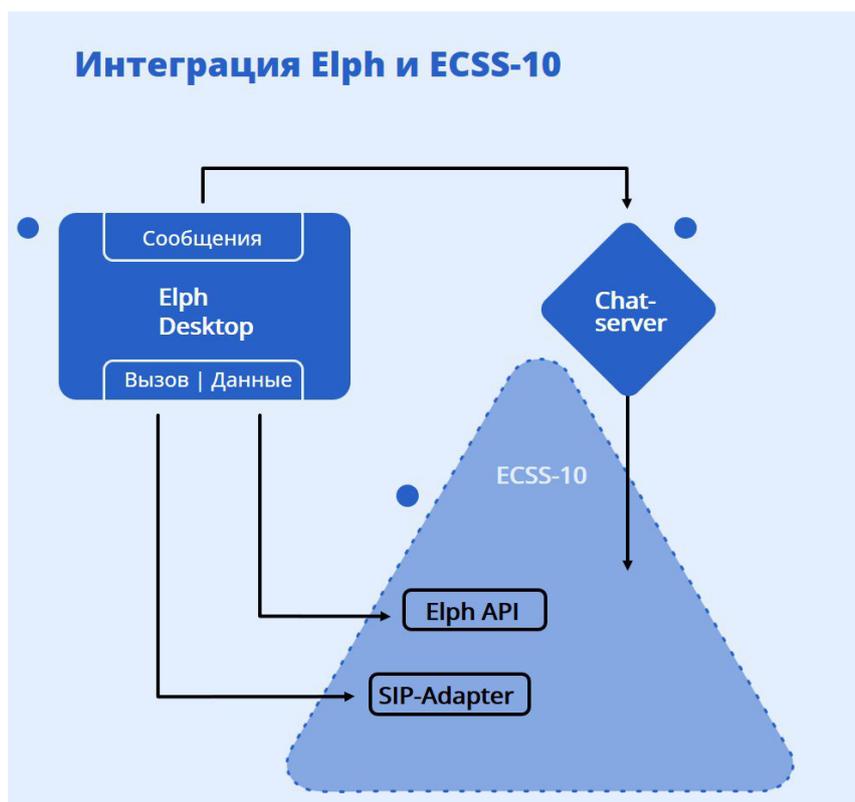
тогда при установке пакета указывается <http://10.0.41.25>, а в настройке для оператора параметр `integration\ssvs` будет:

```
/api/dialog/transcribe?  
interaction_id=%WORKITEM_ID%&key=84eae956ba0a4847752345cb7411af825fcb3cb6d1c0  
5e7e0fbe70b14d1a8c4f
```

WORKITEM_ID заменится на данный из вызова (User-To-User SIP header).

Интеграция с Elph

В данном разделе приведена инструкция по настройке серверной части для работы приложения `Elph-Desktop 2.1.1` и `Elph-Android 2.1.1`

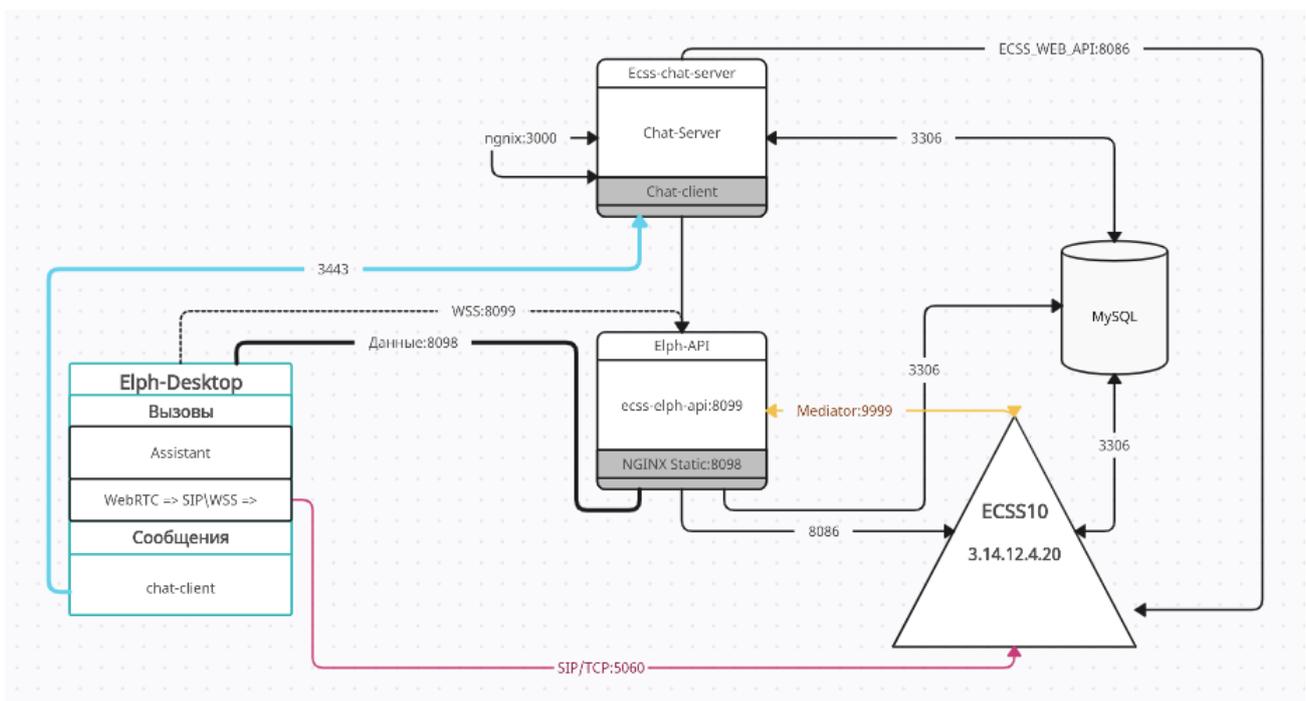


на рисунке представлена упрощённая схема интеграции Elph и ECSS-10

Установка необходимых для Elph пакетов

- Установка и настройка ecss-elph-api
 - Вопросы, задаваемые при установке ecss-elph-api
 - Проверка статуса сервиса
 - Конфигурация
 - Логи
- Установка чат-сервера
 - Разворачиваем докер-образ
 - Инструкция по установке чат-сервера на Astra Linux
 - Обновление чат-сервера

⚠ Для работы приложения **Elph Desktop** на сервере должен быть установлен пакет **ecss-elph-api** и развёрнут **ecss-chat-server**.



Архитектура взаимодействия Elph 2.1.1 с ECSS10

Для начала необходимо подключить репозиторий на ECSS:

```
sudo sh -c "echo 'deb [arch=amd64] http://archive.eltex.org/ssw/bionic/elph/2.1.1 stable main'
>> /etc/apt/sources.list.d/eltex-ecss10-stable.list"
sudo apt update
```

Установка и настройка ecss-elph-api

Установите ecss-elph-api:

```
sudo apt install ecss-elph-api
```

Настройки будут запрошены при установке.

Вопросы, задаваемые при установке ecss-elph-api

Вопрос	
Шаблон вопроса:	ecss-elph-api/core-host
Тип данных:	string
Значение по умолчанию:	localhost
Текст вопроса(en)	IP address(hostname) of ECSS Core:
Текст вопроса(ru)	IP-адрес(хост) на котором развернут ECSS Core:
Описание	Вводится IP или имя хоста, на котором развернут ECSS Core:
Вопрос	
Шаблон вопроса:	ecss-elph-api/core-port
Тип данных:	string
Значение по умолчанию:	8086
Текст вопроса(en)	Port of ECSS Core:
Текст вопроса(ru)	Порт на котором развернут ECSS Core:
Описание	Вводится порт, на котором развернут ECSS Core
Вопрос	
Шаблон вопроса:	ecss-elph-api/mysql-host
Тип данных:	string
Значение по умолчанию:	localhost
Текст вопроса(en)	IP address(hostname) of address book mysql:
Текст вопроса(ru)	IP-адрес(хост) на котором развернут mysql адресной книги:
Описание	Вводится адрес или имя MySQL сервера, на котором находится БД адресной книги. Расположение ecss-mysql относительно ecss-elph-api
Вопрос	
Шаблон вопроса:	ecss-elph-api/mysql-port
Тип данных:	string
Значение по умолчанию:	3306
Текст вопроса(en)	Port of address book mysql:
Текст вопроса(ru)	Порт на котором развернут mysql адресной книги:
Описание	Вводится порт MySQL сервера, на котором находится БД адресной книги
Вопрос	
Шаблон вопроса:	ecss-elph-api/mysql-user
Тип данных:	string
Значение по умолчанию:	address_book
Текст вопроса(en)	Address book mysql username:
Текст вопроса(ru)	Имя пользователя для доступа к mysql адресной книги:
Описание	Вводится логин пользователя БД адресной книги

Вопрос	
Шаблон вопроса:	ecss-elph-api/mysql-password
Тип данных:	password
Значение по умолчанию:	address_book
Текст вопроса(en)	Address book mysql password:
Текст вопроса(ru)	Пароль для доступа к mysql адресной книги:
Описание	Вводится пароль пользователя БД адресной книги. Не путать с "root" - паролем для доступа к mysql по умолчанию
Вопрос	
Шаблон вопроса:	ecss-elph-api/domain
Тип данных:	string
Текст вопроса(en)	Default domain:
Текст вопроса(ru)	Домен по умолчанию:
Описание	Вводится имя домена ECSS, на котором будет использоваться Elph-desktop. Если в поле ничего не вводить, имя домена будет запрошено на странице авторизации приложения.
Вопрос	
Шаблон вопроса:	ecss-elph-api/listen-port
Тип данных:	string
Значение по умолчанию:	8098
Текст вопроса(en)	Port for nginx server:
Текст вопроса(ru)	Порт для сервера nginx:
Описание	Вводится номер порта сервера nginx:
Вопрос	
Шаблон вопроса:	ecss-elph-api/chat-url-enabled
Тип данных:	boolean
Значение по умолчанию:	false
Текст вопроса(en)	Apply chat's URL settings?
Текст вопроса(ru)	Применить настройки URL-адреса чатов?
Описание	При утвердительном ответе будут запрошены настройки чат-сервера
Вопрос	
Шаблон вопроса:	ecss-elph-api/chat-schema
Тип данных:	select
Варианты выбора:	http, https
Значение по умолчанию:	https
Текст вопроса(en)	Chat's URL schema:
Текст вопроса(ru)	Схема URL-адреса чатов:
Описание	Выбирается протокол взаимодействия с чат-серверам
Вопрос	
Шаблон вопроса:	ecss-elph-api/chat-host
Тип данных:	string
Текст вопроса(en)	Chat's IP address(hostname):
Текст вопроса(ru)	IP-адрес(хост) чатов:
Описание	Вводится публичный адрес чат-сервера
Вопрос	
Шаблон вопроса:	ecss-elph-api/chat-port
Тип данных:	string
Значение по умолчанию:	3443
Текст вопроса(en)	Chat's port:

Текст вопроса(ru)	Порт чатов:
Описание	Вводится номер порта чат-сервера
Вопрос	
Шаблон вопроса:	ecss-elph-api/mediator-host
Тип данных:	string
Значение по умолчанию:	localhost
Текст вопроса(en)	IP address(hostname) of ecss-mediator:
Текст вопроса(ru)	IP-адрес(хост) на котором развернут ecss-mediator:
Описание	Вводится IP-адрес или имя хоста, на котором развернут ecss-mediator
Вопрос	
Шаблон вопроса:	ecss-elph-api/mediator-port
Тип данных:	string
Значение по умолчанию:	9999
Текст вопроса(en)	Port of ecss-mediator:
Текст вопроса(ru)	Порт на котором развернут ecss-mediator:
Описание	Вводится номер порта ecss-mediator(порт http-terminal)
Вопрос	
Шаблон вопроса:	ecss-elph-api/mediator-schema
Тип данных:	select
Варианты выбора:	http, https
Значение по умолчанию:	https
Текст вопроса(en)	Mediator's schema:
Текст вопроса(ru)	Схема ecss-mediator:
Описание	Выбирается протокол работы с ecss-mediator
Вопрос	
Шаблон вопроса:	ecss-elph-api/mediator-user
Тип данных:	string
Значение по умолчанию:	admin
Текст вопроса(en)	Mediator's username:
Текст вопроса(ru)	Имя пользователя для доступа к ecss-mediator:
Описание	Вводится имя пользователя для доступа к ecss-mediator
Вопрос	
Шаблон вопроса:	ecss-elph-api/mediator-password-enabled
Тип данных:	boolean
Значение по умолчанию:	false
Текст вопроса(en)	Set password for ecss-mediator?
Текст вопроса(ru)	Установить пароль для ecss-mediator?
Описание	При необходимости можно установить пароль пользователя для доступа к http-terminal
Вопрос	
Шаблон вопроса:	ecss-elph-api/mediator-password
Тип данных:	password
Текст вопроса(en)	Mediator's password:
Текст вопроса(ru)	Пароль для доступа к ecss-mediator:
Описание	Вводится пароль пользователя для доступа к http-terminal

Запомненные ответы на вопросы можно посмотреть в базе debconf. Они будут применены при обновлении пакета.

Пример сохраненных ответов ecss-elph-api:

```

ecss10@ecss1:~$ sudo debconf-get-selections | grep ecss-elph-api
ecss-elph-api ecss-elph-api/mysql-password password address_book
ecss-elph-api ecss-elph-api/mediator-password password
ecss-elph-api ecss-elph-api/chat-host string 192.168.114.122
ecss-elph-api ecss-elph-api/core-port string 8086
ecss-elph-api ecss-elph-api/mediator-schema select http
ecss-elph-api ecss-elph-api/chat-port string 3443
ecss-elph-api ecss-elph-api/mysql-user string address_book
ecss-elph-api ecss-elph-api/mediator-password-enabled boolean false
ecss-elph-api ecss-elph-api/chat-schema select https
ecss-elph-api ecss-elph-api/reinstall boolean false
ecss-elph-api ecss-elph-api/mediator-user string admin
ecss-elph-api ecss-elph-api/mysql-host string localhost
ecss-elph-api ecss-elph-api/mysql-port string 3306
ecss-elph-api ecss-elph-api/listen-port string 8098
ecss-elph-api ecss-elph-api/domain string test.domain
ecss-elph-api ecss-elph-api/mediator-host string localhost
ecss-elph-api ecss-elph-api/core-host string localhost
ecss-elph-api ecss-elph-api/chat-url-enabled boolean true
ecss-elph-api ecss-elph-api/mediator-port string 9999

```

Проверка статуса сервиса

После установки пакета проверьте статус сервиса, он должен быть "Active".

```

ecss10@ecss1:~$ systemctl status ecss-elph-api
● ecss-elph-api.service - Elph desktop websockets API for ecss-10
   Loaded: loaded (/lib/systemd/system/ecss-elph-api.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2023-04-24 17:31:41 +07; 18h ago
     Main PID: 22605 (nodejs)
        Tasks: 11 (limit: 4915)
       Memory: 56.0M
          CGroup: /ecss.slice/ecss-elph-api.service
                 └─22605 /usr/bin/nodejs /usr/share/ecss/ecss-elph-api/dist/websockets/src/main.js

```

Конфигурация

Настройки **ecss-elph-api** записываются в файл **/etc/ecss/ecss-elph-api/config.yaml**, например:

```

default:
  ecss:
    host: localhost
    port: 8086
    domain: test.domain
    user: pubsub
    password: pubsub
  sql:
    host: localhost
    port: 3306
    database: ecss_address_book
    user: address_book
    password: address_book
    types:
      - sip
      - megaco
      - virtual
  chat:
    host: "192.168.114.122"
    port: 3443
    schema: https
  mediator:
    host: localhost
    port: 9999
    schema: https
    user: admin
    password: password

```

По умолчанию создается один профиль — **"default"**. Если вручную добавить еще профили, то в приложении будет запрошено имя профиля. Если домен не задан, то он также будет запрошен.

 user: pubsub, password: pubsub — системные, изменять не нужно!

Вручную править файл конфигурации нежелательно. Если нужно изменить настройки, лучше воспользоваться командой **dpkg-reconfigure**.

```
sudo dpkg-reconfigure ecss-elph-api
```

Если все-таки потребовалось изменить файлы конфигурации, то после изменения необходимо перезапустить сервис **ecss-elph-api**.

Логи

Логи пишутся в системный журнал. При необходимости их можно просмотреть с помощью команд:

```
sudo journalctl -u ecss-elph-api # все логи
sudo journalctl -u ecss-elph-api --since today # логи за сегодня
sudo journalctl -u ecss-elph-api -f # аналог tail -f для просмотра лога в реальном времени
```

Установка чат-сервера

Разворачиваем докер-образ

Скачайте скрипт установки командой:

```
wget https://s3.eltex-co.ru/shared/elph-chat-server/ecss-elph-chat-server.sh
```

Выдайте права на исполнение:

```
chmod +x ecss-elph-chat-server.sh
```

Запустите скрипт, начнется загрузка:

```
./ecss-elph-chat-server.sh download
```

Далее необходимо перелогиниться в терминале, если установка происходит удалённо по ssh, так как пользователь был добавлен в группу docker, после этого необходимо повторить команду:

```
./ecss-elph-chat-server.sh download
```

Авторизуйтесь во внешнем докер-регистре следующей командой (данные авторизации вы можете уточнить в коммерческом отделе):

```
./ecss-elph-chat-server.sh login
```

После этого скачайте образы:

```
./ecss-elph-chat-server.sh pull
```

Теперь необходимо изменить файл конфигурации, подробнее об этом в разделе "[Настройка чат-сервера при установке из docker-контейнера](#)":

```
sudo nano elph-chat-server/stable/.env
```

После сохранения конфигурации можно запускать чат-сервер из корневого каталога:

```
./ecss-elph-chat-server.sh up
```

Инструкция по установке чат-сервера на Astra Linux

Для установки чат-сервера необходимо предварительно установить следующие пакеты:

- [docker.io](#)

⚠ Версия должна быть не ниже 20.10.2

- `docker-compose`

⚠ Версия должна быть не ниже 1.29.2

```
sudo apt install docker.io docker-compose
```

Следующие шаги установки не отличаются от штатных.

Обновление чат-сервера

⚠ Для обновления чат-сервера на конкретную версию, необходимо указать её в поле "TAG" в конфигурационном файле `./elph-chat-server/stable/.env`

Для этого предварительно остановите чат командой:

```
./ecss-elph-chat-server.sh stop
```

Перейдите в директорию `./elph-chat-server/stable/` и отредактируйте конфигурацию, изменив поле "TAG" на версию, которую хотите установить

```
sudo nano .env
```

После этого скачайте образы:

```
./ecss-elph-chat-server.sh pull
```

После завершения процедуры загрузки можно запускать чат-сервер, для этого нужно перейти в папку `./elph-chat-server/stable/` и выполнить команду

```
docker-compose --project-name elph-chat-server-stable up
```

Настройки на сервере

- [Настройка адресной книги](#)
- [Включение CSTA-API](#)
- [Настройка на домене](#)
- [Настройки медиасервера для работы с WebRTC и активация видеоконференций](#)

В данном разделе приведено описание настроек, которые нужно сделать на сервере для работы приложения [Elph-Desktop 2.1](#) и [Elph-Android 2.1](#)

Настройка адресной книги

Предварительно на ECSS должен быть настроен доступ к базе адресной книги. Команды описаны в разделе [/system/address-book/ - системные команды управления адресной книгой абонентов](#).

По умолчанию настройки доступа такие:

```
admin@mycelium1@ecss1:/$ system/address-book/properties/info
```

Property	Value
auto_create_account	true
auto_delete_account	true
database	ecss_address_book
host	address-book.mysql.ecss
password	address_book
port	3306
username	address_book

Включение CSTA-API

Проверьте, что включен [CSTA API](#). Описание команд приведено в разделе [/api/ — команды настройки API CSTA](#).

Пример настроек:

```
admin@mycelium1@ecss1:/$ api/csta/info
```

Property	Value	
auth		
enabled	true	<== включить
ssl_ip	0.0.0.0	
ssl_port	4722	
tcp_ip	0.0.0.0	
tcp_port	4721	
use_ssl	true	
use_tcp	true	

Настройка на домене

В свойствах домена параметр **vats_type** должен быть выставлен в **private**.

```
admin@mycelium1@ecss1:/$ domain/test.domain/properties/info
```

Property	Domain	Value
...		
vats_type	test.domain	private

В свойствах sip необходимо настроить параметр **compact_headers** в режим **force long** для нормальной работы Elph-android:

```
admin@mycelium1@ecss1:/$ domain/test.domain/sip/properties/set compact_headers force_long
Property "compact_headers" successfully changed from:
```

Elph-Android может обрабатывать DTMF в SIP INFO, в формате "application/dtmf-relay", но не может в формате "application/dtmf", поэтому необходимо включить параметр dtmf_relay либо в свойствах sip-домена, либо в sip профиле конкретного абонента, использующего Elph-Android.

Пример настройки для домена:

```
domain/test.domain/sip/properties/set dtmf_relay true
```

Пример настройки для абонента:

```
domain/test.domain/sip/user/set test.group 1000@test.domain dtmf-relay true
```

⚠ Только для работы в **SSW кластере** при включённом параметре **nat_traversal** в sip-профиле абонента, использующего **elph-desktop**, необходимо в свойствах sip-домена активировать параметр **node_control**, также включить **node_control** в sip-профиле **на всех абонентах, использующих elph-desktop** и параметр **nat_traversal** **одновременно**, в остальных случаях **node_control** **следует держать отключенным**, так как это влияет на производительность SSW.

Пример настройки для домена:

```
admin@mycelium1@ecss1:/$ domain/test.domain/sip/user/profile/set with-node-control node_control true
```

Пример настройки для абонента:

```
admin@mycelium1@ecss1:/$ domain/test.domain/sip/user/set 40000 40001@test.domain client-profile with-node-control
```

Настройки медиасервера для работы с WebRTC и активация видеоконференций

Для использования WebRTC и видеозвонков на SSW установите следующие тэги в конфигурационном файле **ecss-media-server**, который находится по пути: `/etc/ecss/ecss-media-server/config.xml`

Параметры, которые необходимо установить для работы с WebRTC.

В секции **general**:

```
use-srtp="optional"
enable-ice-transport="yes"
ice-update="yes"
aggressive-ice="yes"
```

Для прохождения медики за NAT, только если будет использоваться режим `NAT_traversal`, в секции **media** включить параметр:

```
udp-src-check="yes"
```

Для работы видеозвонков в секции **codec**:

```
h264="1"
```

Пример:

```
<?xml version="1.0" encoding="utf-8"?>
<config date="17:31:13 24.04.2023">
  <general log-level="3" log-rotate="yes" max-calls="8192" max-in-group="512" load-sensor="media"
    load-delta="10" calls-delta="100" spool-dir-size="100M" log-name="msr.log" log-path="/var/log/
ecss/media-server" use-srtp="optional" enable-ice-transport="yes" ice-update="yes" aggressive-
ice="yes" stun-server="" suspicious-mode="yes"/>
  <transport bind-addr="127.0.0.1" port="5040" transport="udp+tcp"/>
  <media mixer-clock-rate="8000" use-vad="no" cng-level="0" jb-size="60" rtcp-timeout="0" rtp-
timeout="350" udp-src-check="yes" cn-multiplier="3" port-start="12000" port-range="2048" tias-in-
sdp="no" thread-cnt="2" vid-enc-threads="2" vid-dec-threads="2" video-conf-layout="evenly"
keyframe-interval="1000" vid-decode-delay="100" silence-threshold="-30" dtmf-flash-disable="no"
video-dscp="0" other-dscp="0" dummy-video-src="/usr/share/ecss-media-server/video/
dummy_video.yuv" video-enc-width="1280" video-enc-height="720" finalsilence="1000" rtcp-stat-
dump="yes"/>
  <codec pcma="1" pcmu="2" ilbc="0" gsm="0" g722="3" g729="0" speex="0" l16="0" g7221="0" opus="0
" h264="1" h263-1998="2" t38="1" tel-event-pt="0"/>
  <accounts>
    <!-- <dynamic msr_name="msr.name"
      realm="sip:127.0.0.1:5000"
      dtmf_mode="rfc+inband+info"
      auth_name="user"
      auth_password="password" /> -->
  </accounts>
  <pbyte>
    <mcc bind-addr="127.0.0.1" port="5700"/>
  </pbyte>
  <conf_dir path="/etc/ecss/ecss-media-server/conf.d"/>
  <rtp>
    <auto addr-v4="" />
  </rtp>
</config>
```

Для включения видео в конференциях необходимо переключить режим на медиа-сервере из **audio** в **audio/video**.

Для услуги **conference**:

```
system/media/msr/set --media/conference audio/video
```

Для услуги **teleconference**:

```
system/media/msr/set --media/teleconference audio/video
```

Настройки для пользователей

- Пароль для входа
- Тип терминала — smart
- Услуги
- Включение `core_forking`
 - Описание работы по протоколу WebRTC
 - Настройка на интерфейсе

Elph Desktop может работать как в режиме ассистента к ТА, так и в режиме SIP-клиента.

Пароль для входа

Для пользователей, которые будут работать в приложении Elph, необходимо задать пароль. Данный пароль будет использован так же для аутентификации в чате. Учётная запись чата автоматически создаётся сразу после авторизации абонента в Elph-Desktop.

 Для удобства рекомендуется использовать одинаковые пароли для SIP, логина в Elph и `alias-as-user` при добавлении абонентов.

Настройка пароля производится на `alias`, пример в CLI:

```
domain/test.domain/alias/set 1000 test.group 1000@test.domain password parol
```

Также можно задать пароль через web-конфигуратор:

Карточка абонента > biysk.local > 240007 [240007@biysk.local]

Основные Номера Другие SIP Модификации SIP Лицензионные пакеты ДВО Дополнительные услуги

Добавить Удалить Отменить Домен+Интер...

Имя	Тип	Значение
user_agent	alias	JsSIP 3.9.1
uid	alias	<<"067d66b1075fa17b">>
timezone	global	UTC+07:00
teleconference\password	alias	bXSglt26Wbl2
subscriber_portal\password	alias	8SrMMOY39NYT
subscriber_portal\login	alias	240007
ssvmgm\telephone	global	enabled
password	alias	"pda"
last_incoming_call_info	alias	{rtop_last_incoming_call_info,1,"240470",3703426076844188514,{1664,...
zone	global	<<"default">>
site	global	<<"local">>

Добавить Удалить Отменить изменения и обновить Сохранить

Пароль нужно сообщить пользователю. При авторизации в приложении в качестве логина нужно ввести номер телефона и данный пароль.

Для работы конференции на Elph-android необходимо задать пароль для телеконференции на алиасе:

```
domain/test.domain/alias/set 1000 test.group 1000@test.domain teleconference\password parol
```

И изменить роль на **manager**, чтобы он мог создавать конференции:

```
domain/test.domain/alias/set 1000 test.group 1000@test.domain teleconference\role manager
```

Тип терминала — smart

Для работы с приложением тип терминала абонента должен быть **smart**, пример:

```
domain/test.domain/alias/set 1000 test.group 1000@test.domain terminal_type smart
```

Услуги

Необходимо, чтобы на номере были активны обязательные услуги:

- Удержание вызова (CHOLD);
- Передача вызова (CTR);
- Отображение номера (CLIP);
- Отображение имени (CNIP);
- Мультилайн (ML);
- Конференц-связь с последовательным сбором участников (Conference)

Опционально, при необходимости:

- Голосовая почта (Voicemail);
- Перехват вызова (Pickup).

Возможности использования других услуг также доступны при активации.

Включение core_forking

Описание работы по протоколу WebRTC

На уровень интерфейса был добавлен параметр `core_forking`. Как работает этот параметр, если он выставлен в `true`?

- В `oct_rr_obcsm` делается запрос на SIP-адаптер для получения контактов абонента Б;
- В `oct_rr_forking` делаются параллельные вызовы на контакты абонента Б, полученные на шаге выше. Контакт абонента Б, который ответил раньше остальных, будет проключен с абонентом А.

На уровне `elph-desktp` в SIP-сообщениях REGISTER и INVITE в заголовок Contact был добавлен тег **+sip.ice**. Данный тег служит индикатором того, что удаленная сторона является WebRTC-клиентом.

- Как факт наличия тега **+sip.ice** в SIP-сообщении INVITE используется в SSW? SIP-адаптер формируют занятие на Core с параметром `ice_o` со значением **true**. На Core данный параметр сигнализирует о том, что согласование меди (в `oct_rr_lander`) абонента А нужно парковать с использованием протокола ice <https://datatracker.ietf.org/doc/html/rfc8838>;
- Как факт наличия тега **+sip.ice** в SIP-сообщении REGISTER используется в SSW? SIP-адаптер в ответе на запрос получения контактов абонента Б, помечает контакты как WebRTC. На Core (в `oct_rr_forking`), при формировании параллельных занятий на

контакты абонента Б, в занятии формируется флаг **ice_t** со значением **true**. Данный параметр сигнализирует о том, что согласование медиа (в `oct_rr_lander`) контакта абонента Б нужно делать с использованием протокола ice <https://datatracker.ietf.org/doc/html/rfc8838>.

Параметр **core_forking** со значением **true** перебивает параметр **ice/enabled**.

В случае, если абонент Б использует только WebRTC клиент, то обращаться к SIP-адаптеру для получения единственного WebRTC-контакта пользователя Б не имеет смысла. В данном случае на интерфейсе абоненте Б достаточно будет выставить параметр **ice/enabled**. Параметр **ice/enabled** работает только для абонента Б.

Если простыми словами, то в случае **core_forking**, выставленного в **true**, администратору SSW больше не нужно задумываться об использовании пользователем телефонного аппарата (ТА) и/или Desktop-клиента (WebRTC-клиента, работающего по протоколу ice <https://datatracker.ietf.org/doc/html/rfc8838>).

Настройка на интерфейсе

Установите параметр **core_forking** в значение **true**:

```
domain/<DOMAIN>/iface/user-set <SIP> <GROUP> <USER>@<DOMAIN> core_forking true
```

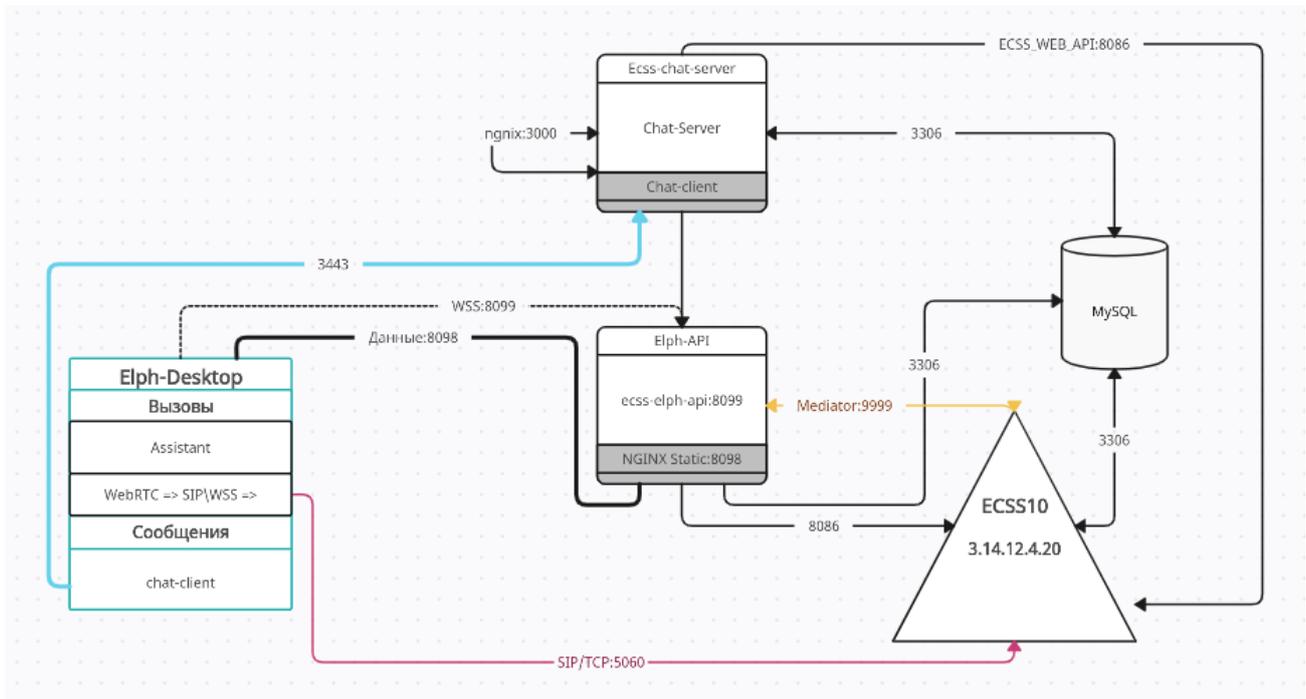
Пример:

```
domain/test.domain/iface/user-set sip1 test.group 1000@test.domain core_forking true
```

На этом настройка абонента закончена. Функционал приложения описан в разделе [Elph Desktop. Руководство по настройке приложения](#)

При обращении пользователя администратор домена/ECSS производит для него вышеприведенные настройки и передает ему параметры подключения и учетные данные для самостоятельной [установки](#).

Настройка чат-сервера при установке из docker-контейнера



Архитектура взаимодействия Elph с ECSS10

Конфигурация

Конфигурация чат сервера хранится в файле `./elph-chat-server/stable/.env`

Например:

```

##### ОБЯЗАТЕЛЬНАЯ КОНФИГУРАЦИЯ #####

# Источник докер образов
DOCKER_REPO="hub-ims.eltex-co.ru"
# Версия докер образов
TAG="2.1.1"

# Публичный адрес сервера
ROOT_URL=https://192.168.114.122:3443

ECSS_WEB_API_HOST=192.168.108.252 # Адрес ядра ECSS10
ECSS_WEB_API_DOMAIN=test.domain

ECSS_ADDRESSBOOK_HOST=192.168.108.252
ECSS_AVATAR_URL_ORIGIN_PREFIX=https://192.168.108.252:8098/loadlocal/ # этот параметр отвечает за
настройку аватаров

##### ОПЦИОНАЛЬНАЯ КОНФИГУРАЦИЯ #####

# Строка подключения к БД
MONGO_URL=mongodb://mongo:27017/elph?replicaSet=rs01
MONGO_OPLUG_URL=mongodb://mongo:27017/local?replicaSet=rs01

ECSS_WEB_API_PORT=8086
ECSS_WEB_API_SERVICE=call

ECSS_ADDRESSBOOK_PORT=3306
ECSS_ADDRESSBOOK_DB_USER=address_book
ECSS_ADDRESSBOOK_DB_PASSWORD=address_book
ECSS_ADDRESSBOOK_DB_NAME=ecss_address_book

UPDATE_ADDRBOOK_AVATAR_EVERY_MS=80000 # Интервал проверки обновления аватаров

NODE_TLS_REJECT_UNAUTHORIZED=0

##### СТАТИЧНАЯ КОНФИГУРАЦИЯ НЕОБХОДИМАЯ ДЛЯ РАБОТЫ #####

## NGINX
NGINX_TAG=1.23.3-${TAG}
MONGO_TAG=4.4-${TAG}
SERVER_TAG=${TAG}

DEPLOY_PLATFORM=deb
PORT=3000
NODE_PATH=/usr/bin/node
NODE_ENV=production

```

В первоначальной настройке необходимо раскомментировать и изменить следующие параметры:

ROOT_URL=https://<ip_address_chat_server>:3443 #Внешний адрес чат-сервера, по которому он будет доступен для пользователей Elph-desktop

ECSS_WEB_API_HOST=<ip_address_elph_api> #Адрес ядра ECSS

ECSS_WEB_API_DOMAIN=<domain_name> #Имя домена ECSS, с которым будет работать чат-сервер

ECSS_ADDRESSBOOK_HOST=<ip_address_mysql> #адрес ecss-mysql

ECSS_AVATAR_URL_ORIGIN_PREFIX=https://<ip_addrss_elph_api>:8098/loadlocal/ #Адрес Elph-api для скачивания аватаров, подробнее об этом в разделе "[Настройка аватаров](#)"

Остальные параметры являются опциональными, изменять их нужно в случае, если используется нестандартная схема развёртывания ECSS-10.

 Обратите внимание на архитектуру, чат-сервер должен иметь доступ к ecss-mysql, иначе не будут работать аватары и имена пользователей в чате, для этого необходимо настроить маску доступа к mysql

делается это командой:

```
dpkg-reconfigure ecss-mysql
```

далее откроется диалоговое окно с вопросом о маске, например для доступа всех ip адресов можно задать маску: %.%.%.%

Сетевые окончания

3443 - https порт для доступа к чат-серверу

Настройка аватаров

В ECSS-10 версии 3.14.12 ещё не реализована единая адресная книга, все данные пользователей (displayname, photo итд) хранятся в Mysql, в таблице **address_book**, поэтому изменения, касаемые фотографий пользователей, необходимо вносить вручную в базу данных.

Для установки или изменения адреса аватара пользователя необходимо узнать его "**uid**", который является уникальным для каждого aliasa. Сделать это можно как с помощью CLI, так и в web-интерфейсе ECSS-10.

Пример для CLI, с помощью команды domain/test.domain/alias/info:

```
admin@[sip1@ecss1#ECSS-CB-DEMO]:/$ domain/test.domain/alias/info 1001
Aliases in domain test.domain
```

Address	Iface	A W	Property	Value
1001	1001@test.domain		user_agent	"Elph-desktop 2.1.136"
			uid	0686370f3e80cc9c

Зная uid пользователя, мы можем задать для него адрес хранения аватара.

Хранение аватаров

Аватары пользователей можно хранить в корне папки `/var/lib/ecss/ecss-elph-api/`

Тогда в таблице Mysql достаточно будет указать только имя файла, также можно ссылаться на внешние источники, указывая прямую ссылку на изображение, например: <http://eltex-co.ru/avatar.jpg>

Поддерживаемые форматы: jpg, png, gif. Размеры и пропорции возможны любые, клиентское приложение само обрежет под нужный размер. Рекомендуются квадратные изображения.

Для удобства рекомендуется задавать такое же имя файла аватара, как и номер телефона пользователя, например **1001.jpg**.

Изменение адреса аватара в mysql

Для начала выполните вход в mysql, пароль по умолчанию: **root**

```
sudo mysql -u root -p
```

Выберите необходимую таблицу:

```
use ecss_address_book;
```

Для изменения адреса аватара пользователя выполните команду **update**, указав **uid** пользователя и название файла. В данном примере файл аватара лежит по пути `/var/lib/ecss/ecss-elph-api/`, поэтому достаточно указать его имя:

```
update contact set photo='1001.jpg' where uuid like '0686370f3e80cc9c';
```

Настройка аватара завершена, он отобразится при следующей авторизации пользователя в Elph-desktop.

Лицензирование Elph

- [Ограничения](#)
- [Команды для работы с лицензионными ограничениями Elph](#)
- [Реакция системы при превышении лицензионного лимита](#)

В ECSS-10 версии 3.14.12 реализована поддержка лицензирования количества активных клиентов Elph-Desktop, Elph-Android и Elph-IOS. При лицензировании данных продуктов учитываются только их sip-регистрации на текущий момент времени.

Ограничения

Имеется 2 типа ограничений:

- количество Elph групп(в текущей версии только 1 группа);
- максимальное количество клиентов Elph группы - лимит участников (клиентов) одной группы.

По умолчанию доступны 10 одновременных SIP-регистраций Elph-клиентов независимо от платформы. То есть, если один и тот же клиент одновременно зарегистрирован на SIP-адаптере ECSS-10 с двух устройств, например с android и desktop, то он занимает 2 лицензии до истечения регистрации или принудительной отмены регистрации.

Команды для работы с лицензионными ограничениями Elph

Просмотр системных ограничений на уровне лицензии, пример:

```
admin@mycelium1@ecss1:/$ cluster/storage/ds1/licence/current-limits
...
Elph
total count of Elph group          1
total count of members of the one Elph group  10
...
```

Для проверки доступных лицензий для Elph используется команда:

```
system/licence/elph/usage --domains
```

Пример:

```
admin@mycelium1@ecss1:/$ system/licence/elph/usage --domains
System usage:
```

Members	System Limit
2	10

```
Licences distributed by domains:
```

Domain	Members	Domain restriction
biysk.local	2	8

Подробную информацию о клиентах, зарегистрированных с использованием лицензии Elph, можно узнать с помощью команды, которая находится по пути:

```
domain/<DOMAIN>/sip/user/elph-registered *
```

Пример:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/sip/user/elph-registered
Executed on the sip1@ecss1
```

Interface	Number	Dynamic	Contact
Registration info			
240470@biysk.local	240470	false	q: 1.0; ip-set: ipset1; on port: 5060; sip
registered on:			registered by node: sip1@ecss1
node: sip1@ecss1 port: tcp 192.168.2.61:5060			registered by node: sip1@ecss1
last via list:			sip transport: tcp
SIP/2.0/TCP 192.168.2.14			user-agent: Elph-desktop 2.1.1+6
240470@192.168.2.14:12702>;expires=529			<sip: last user agent:
Elph-desktop 2.1.1+6			
240474@biysk.local	240474	false	q: 1.0; ip-set: ipset1; on port: 5060; sip
registered on:			registered by node: sip1@ecss1
node: sip1@ecss1 port: tcp 192.168.2.61:5060			registered by node: sip1@ecss1
last via list:			sip transport: tcp
SIP/2.0/TCP 192.168.2.25			user-agent: Elph-desktop 2.1.1+6
240474@192.168.2.25:12702>;expires=601			<sip: last user agent:
Elph-desktop 2.1.1+6			

```
elements: 2
```

Пример установки ограничений для определенного домена:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/properties/restrictions/set
elph\member\limit 8
Restriction elph\member\limit changed from 10 to 8.
```

Набор команд CLI по управлению лицензиями приведен в разделе "[Справочник команд CLI](#)".

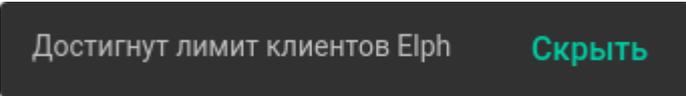
Реакция системы при превышении лицензионного лимита

При попытке регистрации клиента Elph, когда уже достигнут предел лицензионного лимита активных подключений, SIP-адаптер будет отвечать сообщением 403 Forbidden.

Пример:

```
SIP/2.0 403 Forbidden
Via: SIP/2.0/TCP 10.25.111.35:12733;branch=z9hG4bK1251590303
From: "50000" <sip:50000@10.25.88.26:5060>;tag=1251395370
To: "50000" <sip:50000@10.25.88.26:5060>;tag=ecssR168006922693289102911114
Call-ID: 2_1251433770@10.25.111.35
CSeq: 2 REGISTER
Warning: 399 Elph "Licence limit is exceeded!"
User-Agent: Eltex ECSS-10 SIP 3.14.12.957
Content-Length: 0
```

У клиента выйдет сообщение:



Достигнут лимит клиентов Elph [Скрыть](#)

При попытке задать на домене количество клиентов Elph, превышающее системный лимит будет выведено сообщение, например:

```
admin@mycelium1@ecss1:/$ domain/biysk.local/properties/restrictions/set
elph\member\limit 11
Command error: New value 11 for limit elph\member\limit more than licence
limit 10.
```