

# Сервер передачи ключей Keys Transfer Server (KTS)

## Руководство администратора

Индекс	KTS-AG
Конфиденциальность	Для внутреннего пользования - L1
Ревизия	1.0
Статус	Согласован

## Содержание

1. Аннотация .....	3
2. Термины и сокращения .....	4
3. Обслуживание системы .....	5
4. Настройка bbx_server_go .....	6
4.1. Конфигурационный файл bbx_server_go .....	6
5. Эксплуатация Системы .....	7
5.1. Перенос данных .....	7
5.1.1. Кумулятивный импорт персонализационных данных на KTS сервер (v.2) .....	7
5.1.1.1. Введение .....	7
5.1.1.1.1. Тестовые вектора и валидация ключей .....	7
5.1.1.1.2. Fusemap config .....	8
5.1.1.2. Необходимые условия .....	8
5.1.1.2.1. Необходимые условия (1 вариант: cumulative_import_v2.py, доступ в Internet) .....	8
5.1.1.2.2. Необходимые условия (2 вариант: бинарный файл cumulative_import_v2, без доступа в Internet) .....	8
5.1.1.3. Содержимое кумулятивного архива .....	8
5.1.1.4. Действия при получении вместо кумулятивного архива набора файлов .....	9
5.1.1.5. Предварительные действия .....	10
5.1.1.6. Загрузка "кумулятивных" персонализационных данных .....	10
5.1.1.6.1. Загрузка "кумулятивных" персонализационных данных (1 вариант: cumulative_import_v2.py, доступ в Internet) .....	10
5.1.1.6.2. Загрузка "кумулятивных" персонализационных данных (2 вариант: бинарный файл cumulative_import_v2, без доступа в Internet) .....	11
5.1.2. Удаление данных на KTS сервере .....	12
5.1.3. Формирование и выгрузка отчета о программировании .....	13
6. Ведение Логов .....	15
6.1. Режимы Ведения Логов .....	15
6.2. Форматы логов .....	15
6.3. Логи BBX_Client .....	15
6.4. Просмотр системных логов сервера .....	15

## 1. Аннотация

Данный документ содержит настройки компонентов Сервера передачи ключей Keys Transfer Server (KTS) (далее по тексту - KTS или Система).

Документ предназначен для сотрудников отдела мониторинга и инсталляции, а также для других технических специалистов, в обязанности которых входит настройка системы KTS и поддержание её работоспособности.

## 2. Термины и сокращения

Термин	Определение
FTP (File Transfer Protocol)	Протокол передачи файлов – стандартный протокол, предназначенный для передачи файлов по TCP-сетям (например, Интернет). Протокол построен на архитектуре "клиент-сервер" и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером.
Fusemap Config	Подробное описание области OTP-памяти чипа с адресами, полями и их размерами, включая OTP-ключи и конфигурационные биты.
KGS	Система генерации ключей Keys Generation System (KGS). Продукт ООО "ПЦТ" для работы с ключами: генерация, экспорт, импорт, управление.
KTS сервер	Сервер, на котором развернуты <code>bbx_server_go</code> и база данных (компоненты "Сервера передачи ключей Keys Transfer Server (KTS)").
OTP-ключи (One Time Programmable)	Ключи, которые прошиваются в однократно программируемую область памяти в чипе.
Номер Партии (Part Number, PN)	<p>Сущность, которая характеризует заказы чипов у производителя (чип-вендора). Как правило, в чипах используемое имя <code>part number</code>, выгравированное на корпусе. В KGS сущность кроме имени имеет внутренний идентификатор.</p> <p>В KGS имя задается пользователем и может не совпадать с маркировкой на чипе.</p>
Тип Партии (Part Type, PT, PTPP)	<p>Сущность, которая характеризует набор общих OTP-ключей в чипах одной партии: у всех чипов одной PT одинаковые общие ключи. В чипах используется только идентификатор <code>Part Type ID</code>, в KGS также используется пользовательское имя.</p> <p>Не следует путать Тип Партии с Номером Партии.</p>

Сокращение	Расшифровка
БД	База данных
FMС	FuseMap Config
FMCD	FuseMap Config Description
ID	Identifier
OTP	One-Time Programmable
TDE	Transparent Database Encryption

### 3. Обслуживание системы

Обслуживание системы KTS заключается в выполнении следующих основных действий:

- Изменение настроек KTS (при необходимости).
- Эксплуатация системы: управление данными в базе KTS с помощью специальных скриптов.
- Устранение ошибок на основе логов системы.

## 4. Настройка `bbx_server_go`

Настройка осуществляется с помощью конфигурационного файла `bbx_server_go.cfg`.

### 4.1. Конфигурационный файл `bbx_server_go`

Изначально в системе присутствует шаблон файла конфигурации с расширением `.cfg.dft`, содержащий дефолтные значения параметров.

Создайте копию этого файла с расширением `.cfg`, с которой в дальнейшем будет производиться работа:

```
sudo cp bbx_server_go.cfg.dft bbx_server_go.cfg
```

Таким образом, шаблон с начальными значениями параметров остается неизменным.

Описание параметров, содержащихся в конфигурационном файле, предоставляется по запросу.

## 5. Эксплуатация Системы

### 5.1. Перенос данных

Перенос персонализационных данных (ОТР-ключей), а также прочих данных с FTP-сервера KGS на KTS сервер и обратно осуществляется администратором KTS. Файлы при передаче дополнительно шифруются PGP-ключом администратора.

**⚠ Обратите внимание!** В скриптах загрузка данных осуществляется в одной транзакции - транзакция закрывается, только когда доработает весь блок. Иными словами, пока все ключи не догрузятся, они не будут видны.

Необходимо учитывать, что если начать загружать те же ключи или вектора параллельно, то скрипты упадут с ошибкой (так как эти строки уже будут заблокированы на весь процесс).

#### 5.1.1. Кумулятивный импорт персонализационных данных на KTS сервер (v.2)

##### 5.1.1.1. Введение

Имеется возможность экспорта **единым (кумулятивным) архивом** из KGS данных, необходимых для персонализации чипов (ОТР-ключи, конфигурация партии part type/part number, fusemap config, тестовые вектора для ОТР-валидации).

С помощью скрипта cumulative\_import\_v2.py осуществляется обработка, проверка и импорт этих данных в базу KTS (иными словами, "кумулятивный импорт" либо "импорт кумулятивного архива").

Скрипт запускается под любым пользователем.

**i** Вся обработка и загрузка данных выполняются внутри Python скрипта с использованием курсора базы данных. Данные считываются и загружаются с помощью SQL-запросов.

Существует **две реализации работы со скриптом**:

1. С помощью cumulative\_import\_init.sh и cumulative\_import\_v2.py. Требуется подключение к сети Internet для установки необходимых пакетов.
2. С помощью бинарного файла cumulative\_import\_v2. Требуется openssl (пакет входит в комплект поставки). Используется на машинах **без** подключения к сети Internet.

Особенности этих реализаций (если есть) указаны в разделах ниже.

##### 5.1.1.1.1. Тестовые вектора и валидация ключей

Валидация ключей необходима для проверки 100% чипов, проходящих персонализацию на линии, сразу после записи ключей в ОТР. Валидация направлена на проверку ключей из ОТР. Как правило, само значение ключа из ОТР недоступно для чтения в ПО, но может быть использовано чипом в криптоблоке или лестнице ключей, причем может использовать KDF-функцию. Поэтому валидация выполняется косвенным путем, через проверку на тестовых векторах. Тестовые вектора вычисляются в KGS и передаются в KTS вместе с самими ОТР-ключами.

Скрипт позволяет не только проверять и импортировать тестовые вектора в базу KTS, но удалять тестовые вектора для выбранного Part Type из БД, а также выполнять обе операции одновременно (сначала будет выполнено удаление, затем - импорт).

#### 5.1.1.1.2. Fusemap config

Fusemap Config (FMC) определяет полный список полей в OTP-памяти чипа с размерами и длинами, включая свойства OTP-ключей (без значений) и свойства и значения конфигурационных бит.

Для повышения контроля над FMC, записываемым в чипы, передача FMC может осуществляться через KTS. Скрипт позволяет загрузить в базу либо удалить из нее данный fusemap config file.

#### 5.1.1.2. Необходимые условия

##### 5.1.1.2.1. Необходимые условия (**1 вариант**: cumulative\_import\_v2.py, доступ в Internet)

- Требуется получить (из KGS) и расшифровать (PGP) кумулятивный архив. Имя общего архива: `<PartTypeID>_<Start Device Number>_<End Device Number>_personalization_data.zip`
- На сервере должно быть установлено следующее ПО:
  - python3 (третьей версии)

 python3 входит в состав эталонного образа ОС ("устанавливается из коробки").

- пакет python3-pip
- python3-venv
- пакет libpq-dev
- Перед работой со скриптом должно быть инициализировано виртуальное окружение (см. раздел "Загрузка "кумулятивных" персонализационных данных (1 вариант: cumulative\_import\_v2.py, доступ в Internet)", подраздел "Загрузка "кумулятивного архива"", шаг 2):

```
source cumulative_import_init.sh
```

##### 5.1.1.2.2. Необходимые условия (**2 вариант**: бинарный файл cumulative\_import\_v2, без доступа в Internet)

- Требуется получить (из KGS) и расшифровать (PGP) кумулятивный архив. Имя общего архива: `<PartTypeID>_<Start Device Number>_<End Device Number>_personalization_data.zip`
- На сервере должно быть установлено следующее ПО:
  - python3 (третьей версии)

 python3 входит в состав эталонного образа ОС ("устанавливается из коробки").


- openssl\_1.1.1w-0+deb11u1\_amd64.deb (входит в комплект поставки)

#### 5.1.1.3. Содержимое кумулятивного архива

Кумулятивный архив, экспортируемый из KGS, в общем случае содержит следующие данные:

1. Тестовые вектора:
  - а. Файл с результатами OTP-валидации. Доступ к формату файла предоставляется по запросу.



- i. Имя файла (значение по умолчанию): *<input\_template\_filename>\_processed.<br><input\_template\_extension>*
    - где *<input\_template\_filename>* и *<input\_template\_extension>* - имя с расширением входного файла-шаблона с описанием механизма генерации тестовых векторов, сохраненный в параметрах профиля визарда.
  - ii. **Обратите внимание!** Для получения файла с результатами OTP-валидации нужно предварительно:
    1. Составить входной файл для KGS, для генерации тестовых векторов. Доступ к формату файла предоставляется по запросу. Файл создается вручную.
    2. Указанный файл нужно подать на вход системы KGS. Система KGS обрабатывает этот файл и на его основе составляет файл, описанный в п.1.а.
  - b. Если в импортируемом файле найдено хоть одно ненулевое значение *<RootKeyKDF>*, обработанное в системе KGS без ошибок: **дополнительно** экспортируется файл с параметрами KDF. Доступ к формату файла предоставляется по запросу.
    - i. Имя файла: *<devc\_id>\_<devc\_name>\_rk\_kdf\_config.dat*
  2. Fusemap config (опционально, при выборе пользователем в KGS):
    - a. Файл с fusemap config (FMC):
      - Имя файла: *<src\_filename>.tde*, где:
        - *src\_filename* - исходное имя файла с расширением, которое было сохранено при импорте файла в KGS
        - ".tde" - дописываемое расширение
      - Формат файла с FMC: бинарный файл.
      - При экспорте файл FMC зашифровывается согласно лестнице ключей для выбранного KTS.
    - b. Файл, содержащий хэш лестницы ключей:
      - Имя файла: *<ext\_server\_id>\_key\_ladder\_hash* (без расширения).
      - Доступ к формату файла предоставляется по запросу.
  3. Конфиг партии для KTS:
    - a. Доступ к формату файла предоставляется по запросу. При формировании файла конфигурации дополнительно вычисляется подпись и дописывается в файл. При этом **обязательным условием** является наличие в системе private PGP-ключа, привязанного к выбранному Part Type, этот ключ используется для подписи.
    - b. Имя файла: *<VendorName>\_<PartNumberName>\_bbx\_config.dat*
  4. OTP-ключи:
    - a. Файл с OTP-ключами:
      - i. Доступ к формату файла предоставляется по запросу.
      - ii. Имя файла:  
*<VendorName>\_<BlackBoxName>\_<PartTypeID>\_<StartDeviceNumber>\_<EndDeviceNumber>*
    - b. Файл, содержащий хэш лестницы ключей:
      - i. Имя файла: *<ext\_server\_id>\_key\_ladder\_hash* (без расширения).
      - ii. Доступ к формату файла предоставляется по запросу.
-  Файл перезапишет созданный ранее, при загрузке FMC в базу KTS (см. пункт 2.b выше).
5. Файл с метаданными:
    - a. Доступ к формату файла предоставляется по запросу.
    - b. Имя файла: *metadata.json*

#### 5.1.1.4. Действия при получении вместо кумулятивного архива набора файлов

Персонализационные данные загружаются в БД KTS только посредством кумулятивного архива. Однако в KGS файлы, входящие в состав этого архива, могут быть подготовлены по отдельности.

Т.е. может возникнуть ситуация, когда вместо кумулятивного архива из KGS будут получены несколько файлов: файл с OTP-ключами, файл с тестовыми векторами, файл с fusemap config, файл с конфигурацией (Part type configuration).

В этом случае нужно:

1. Подготовить файл с метаданными (с расширением json, согласно примеру):
  - a. Доступ к формату файла предоставляется по запросу.
  - b. Имя файла: *metadata.json*
  - c. Получить хэш для файла с именем `<file_name>` можно с помощью команды:

```
sha256sum <file_name> | tr 'a-f' 'A-F'
```

2. Собрать в одной папке:
  - a. сформированный *metadata.json*
  - b. полученные ранее файлы с персонализационными данными (файлы должны иметь названия, указанные в *metadata.json*, в соответствующем поле "file"; хэши этих файлов должны иметь значения, указанные в *metadata.json*, в соответствующем поле "sha256").
3. Заархивировать эти файлы. Имя у архива может быть **любым**. Рекомендуется подбирать имя для кумулятивного архива так, чтобы оно было понятно, уникально и удобно при работе.

#### 5.1.1.5. Предварительные действия



Выполняются однократно при установке "с нуля" / обновлении KTS.

Последовательность действий:

1. Удалите *bbx-scripts*, которые использовались до данной версии KTS (во избежание путаницы).
2. Скопируйте *bbx-scripts* (входят в комплект поставки) на KTS сервер:
  - a. ***bbx-scripts-X.X.X.zip***
3. Распакуйте архив.
4. Пользователю, под которым будут запускаться скрипты, предоставьте права полного доступа к папке, содержащей скрипты.

#### 5.1.1.6. Загрузка "кумулятивных" персонализационных данных

5.1.1.6.1. Загрузка "кумулятивных" персонализационных данных (**1 вариант**: *cumulative\_import\_v2.py*, доступ в Internet)

5.1.1.6.1.1. Установка необходимых пакетов

Требуется установить *libpq-dev*, *python3-pip*, *python3-venv*.

5.1.1.6.1.2. Загрузка "кумулятивного архива"

1. Необходимый архив переносится с FTP-сервера KGS в рабочую папку БД KTS и расшифровывается PGP-ключом администратора.

2. Запускается (под любым пользователем) скрипт инициализации виртуального окружения `cumulative_import_init.sh`, входящий в комплект поставки:

```
source cumulative_import_init.sh
```

3. Запускается (под любым пользователем) скрипт `cumulative_import_v2.py`, входящий в комплект поставки.

**Формат команды запуска скрипта с параметрами:**

```
python3 cumulative_import_v2.py [-h] -a "CUMULATIVE_ARCHIVE" [-ho DATABASE_HOST] [-p DATABASE_PORT] [-u DATABASE_USER] [-P DATABASE_USER_PASSWORD] [-d DATABASE_NAME]
```

Параметры запуска:

- a. -h - Вызов справки.
  - b. -a "CUMULATIVE\_ARCHIVE", --archive "CUMULATIVE\_ARCHIVE" - кумулятивный архив. Обязательный параметр.  
**Параметр задаётся "в кавычках"** (т.е. если задан путь к файлу, то в кавычках указывается весь путь, если задано только имя файла (лежит в той же папке, что и скрипт), то в кавычках указывается имя файла).
  - c. -ho DATABASE\_HOST, --host DATABASE\_HOST - имя хоста KTS DB. Значение по умолчанию (задается в файле скрипта) - 'localhost'.
  - d. -p DATABASE\_PORT, --port DATABASE\_PORT - номер порта KTS DB. Значение по умолчанию (задается в файле скрипта) - '5432'.
  - e. -u DATABASE\_USER, --user DATABASE\_USER - имя пользователя KTS DB. Значение по умолчанию (задается в файле скрипта) - 'bbxadmin'.
  - f. -P DATABASE\_USER\_PASSWORD, --pass DATABASE\_USER\_PASSWORD - пароль для доступа к KTS DB. Значение по умолчанию (задается в файле скрипта) - 'bbxadmin'.
  - g. -d DATABASE\_NAME, --database DATABASE\_NAME - имя базы KTS DB. Значение по умолчанию (задается в файле скрипта) - 'bbx'.
4. Скрипт осуществляет проверку и загрузку персонализационных данных в целевые структуры БД.
    - a. В процессе работы скрипта логи пишутся в консоль. Убедитесь, что в логах отсутствуют сообщения об ошибках.
  5. **(Опционально)** по окончании работы с кумулятивным скриптом выполните команду деактивации виртуального окружения:

```
deactivate
```

6. **(Обязательно)** после импорта персонализационных данных в БД необходимо **перезагрузить** `bbx_server_go`.

5.1.1.6.2. Загрузка "кумулятивных" персонализационных данных (**2 вариант:** бинарный файл `cumulative_import_v2`, без доступа в Internet)

5.1.1.6.2.1. Установка пакета OpenSSL



Выполняется однократно при установке "с нуля" / обновлении KTS.

Требуется установить OpenSSL (пакет входит в комплект BBX Scripts).

## 5.1.1.6.2.2. Загрузка "кумулятивного архива"

1. Необходимый архив переносится с FTP-сервера KGS в рабочую папку БД KTS и расшифровывается PGP-ключом администратора.
2. Запускается (под любым пользователем) бинарный исполняемый файл *cumulative\_import\_v2* (в папке *cumulative\_import\_v2*), входящий в комплект поставки.

Формат команды запуска бинарного файла с параметрами:

```
./cumulative_import_v2 -a "CUMULATIVE_ARCHIVE" [-ho DATABASE_HOST] [-p DATABASE_PORT] [-u DATABASE_USER] [-P DATABASE_USER_PASSWORD] [-d DATABASE_NAME]
```

Параметры запуска:

- a. -a "CUMULATIVE\_ARCHIVE", --archive "CUMULATIVE\_ARCHIVE" - кумулятивный архив. Обязательный параметр.  
**Параметр задаётся "в кавычках"** (т.е. если задан путь к файлу, то в кавычках указывается весь путь, если задано только имя файла (лежит в той же папке, что и скрипт), то в кавычках указывается имя файла).
  - b. -ho DATABASE\_HOST, --host DATABASE\_HOST - имя хоста KTS DB. Значение по умолчанию (задается в файле скрипта) - 'localhost'.
  - c. -p DATABASE\_PORT, --port DATABASE\_PORT - номер порта KTS DB. Значение по умолчанию (задается в файле скрипта) - '5432'.
  - d. -u DATABASE\_USER, --user DATABASE\_USER - имя пользователя KTS DB. Значение по умолчанию (задается в файле скрипта) - 'bbxadmin'.
  - e. -P DATABASE\_USER\_PASSWORD, --pass DATABASE\_USER\_PASSWORD - пароль для доступа к KTS DB. Значение по умолчанию (задается в файле скрипта) - 'bbxadmin'.
  - f. -d DATABASE\_NAME, --database DATABASE\_NAME - имя базы KTS DB. Значение по умолчанию (задается в файле скрипта) - 'bbx'.
3. Скрипт осуществляет проверку и загрузку персонализационных данных в целевые структуры БД.
    - a. В процессе работы скрипта логи пишутся в консоль. Убедитесь, что в логах отсутствуют сообщения об ошибках.
  4. **(Обязательно)** после импорта персонализационных данных в БД необходимо **перезагрузить** *bbx\_server\_go*.

## 5.1.2. Удаление данных на KTS сервере

Назначение:

Скрипт *blbx\_del\_data.sh* удаляет **все** данные (например, тестовые вектора, *fusemap config*) из БД KTS для выбранного Part Number, при условии, что ключи ещё не запрашивались.

Скрипт запускается под текущим пользователем.

Необходимые условия:

- имя Part Number, для которого удаляются данные.

Последовательность действий:

1. Запускается (под текущим пользователем) скрипт *blbx\_del\_data.sh*, входящий в комплект поставки. Формат команды запуска скрипта с параметрами:

```
bash blbx_del_data.sh [-help] -part_num <part number> [-h <host>] [-p <port>] [-u <pg_user>] [-P <pg_user_password>] [-d <database>]
```

Параметры запуска:

- a. `-help` - вызов справки.
  - b. `-part_num <part number>` - имя `part number`, для которого удаляются данные. Обязательный параметр.  
**Примечание.** Скрипт удаляет все данные из базы для `<part number>` при условии, что ключи ещё не запрашивались.
  - c. `-mode <fmc, test_vectors>` - два возможных значения: `fmc` (чтобы удалить fusemap config для выбранного `Part number`) либо `test_vectors` (чтобы удалить все тестовые вектора для выбранного `Part number`). Обязательный параметр.
  - d. `-h <host>` - имя хоста KTS DB. Значение по умолчанию (задается в файле скрипта) - "localhost".
  - e. `-p <port>` - номер порта KTS DB. Значение по умолчанию (задается в файле скрипта) - "5432".
  - f. `-u <pg_user>` - имя пользователя KTS DB. Значение по умолчанию (задается в файле скрипта) - "bbxadmin".
  - g. `-P <pg_user_password>` - пароль для доступа к KTS DB. Значение по умолчанию (задается в файле скрипта) - "bbxadmin".
  - h. `-d <database>` - имя базы KTS DB. Значение по умолчанию (задается в файле скрипта) - "bbx".
2. Скрипт осуществляет удаление данных из БД.
  3. По окончании работы скрипта проверяется его лог файл (*blbx\_del\_data.log*) на отсутствие ошибок.

### 5.1.3. Формирование и выгрузка отчета о программировании

В процессе персонализации каждому чипу присваивается один из статусов, характеризующий его состояние:

- a) Неиспользованный (Unused);
- b) ПД для чипа были запрошены (Requested);
- c) Успешно персонализирован (Programmed OK);
- d) Ошибка персонализации (Programming FAILED).


Статусы (c) и (d) устанавливаются клиентским приложением по результатам персонализации.

Данные о статусе персонализации используются для формирования отчета о программировании для выбранной партии чипов, выбранного периода персонализации и пр. Отчет о программировании создается с помощью скрипта *blbx\_make\_report.sh* (входит в комплект поставки).

В качестве аргументов указываются следующие параметры:

1. `-help` - вызов справки.
2. `-f "<output file>"` - выходной файл, в котором будет записан результат. Указывается имя файла с отчетом (имя, с которым в текущей папке будет сохранен файл с отчетом). Доступ к формату файла предоставляется по запросу. **Параметр задаётся "в кавычках"**.
3. `-start_date "<start_date>"` - дата начала периода персонализации - начальная дата периода, за который будет формироваться отчет. **Параметр задаётся "в кавычках"**.
4. `-stop_date "<stop_date>"` - дата окончания периода персонализации - конечная дата периода, за который будет формироваться отчет. **Параметр задаётся "в кавычках"**.

5. `-part_num <part number>` - part number (номер партии чипов), для которого нужно сформировать отчет.
6. `-part_type <part type>` - числовой идентификатор part type (типа партии чипов), для которого нужно сформировать отчет.

 Один из параметров [`-part_num <part number>`] и [`-part_type <part type>`] является обязательным при запуске скрипта: либо должен быть указан `<part number>` (при этом `<part type>` не используется), либо `<part type>` (при этом `<part number>` не используется).

7. `-chip_start <chip start id>` - начальный номер диапазона чипов - наименьший из диапазона ID чипов, данные о персонализации которого попадут в отчет.
8. `-chip_stop <chip stop id>` - конечный номер диапазона чипов - наибольший из диапазона ID чипов, данные о персонализации которого попадут в отчет.
9. `-h <host>` - имя хоста KTS DB. Значение по умолчанию (задается в файле скрипта) - "localhost".
10. `-p <port>` - номер порта KTS DB. Значение по умолчанию (задается в файле скрипта) - "5432".
11. `-u <pg_user>` - имя пользователя KTS DB. Значение по умолчанию (задается в файле скрипта) - "bbxadmin".
12. `-P <pg_user_password>` - пароль для доступа к KTS DB. Значение по умолчанию (задается в файле скрипта) - "bbxadmin".
13. `-d <database>` - имя базы KTS DB. Значение по умолчанию (задается в файле скрипта) - "bbx".

Формат команды запуска скрипта с параметрами:

```
bash blbx_make_report.sh [-help] -f "<output file>" [-start_date "<start_date>"] [-stop_date "<stop_date>"] [-part_num <part number>] [-part_type <part type>] [-chip_start <chip start id>] [-chip_stop <chip stop id>] [-h <host>] [-p <port>] [-u <pg_user>] [-P <pg_user_password>] [-d <database>]
```

### Примеры:

- запуск скрипта с указанием part type:

```
bash blbx_make_report.sh -part_type 131 -p 5433 -f "test_report"
```

- запуск скрипта с указанием part number:

```
bash blbx_make_report.sh -part_num 1 -p 5433 -f "test_report"
```

Файл со сформированным отчетом зашифровывается PGP ключом администратора KTS и переносится на FTP-сервер KGS для последующего хранения.

## 6. Ведение Логов

### 6.1. Режимы Ведения Логов

Компонент ведет логи, информация из которых может быть использована для решения возникающих проблем. Логи могут вестись с разной степенью подробности.

Доступны следующие режимы ведения логов:

- 0 - trace: подробная информация по любым действиям;
- 1 - debug: конфигурационные данные (при запуске системы), другая информация, необходимая для отладки, + сообщения уровня Info;
- 2 - info (значение по умолчанию): базовая информация (сообщения о запуске, работе, выключении системы) + сообщения уровня Warning;
- 3 - warning: системные предупреждения + сообщения уровня Error;
- 4 - error: все ошибки, возникающие в процессе работы, в том числе ошибки уровня Fatal;
- 5 - fatal: критические ошибки, приводящие к сбоям системы.

### 6.2. Форматы логов

Все логи представлены в формате JSON. Каждое сообщение лога представлено в формате отдельной JSON-структуры с исчерпывающим набором полей.

В зависимости от уровня логирования изменяется набор данных внутри JSON либо добавляются новые элементы (записи лога).

Логи для режима **debug** имеют структуру JSON и по формату делятся на следующие виды:

1. Лог для event: "request";
2. Лог для event: "response";

Логи для event: "request" и event: "response" содержат информацию по запросу к Системе и ответу на запрос соответственно.

### 6.3. Логи BBX\_Client

Клиентская библиотека пишет логи ошибок (с пояснениями, если из кода ошибки причина не понятна) в `bbx_chip_client_errors.log`

### 6.4. Просмотр системных логов сервера

Логи также пишутся в системные журналы, который можно посмотреть через `journalctl`.

Используется команда вида:

```
sudo journalctl -u bbx_server_go.service
```

© ООО "ПЦТ", 2023-2024

Документация "Сервер передачи ключей Keys Transfer Server (KTS). Руководство администратора" является объектом авторского права. Воспроизведение всего произведения или любой его части воспрещается без письменного разрешения правообладателя