

# Эксплуатация

- [Системные требования Windows](#)
- [Установка в windows](#)
- [Конфигурирование](#)
- [Настройки датчиков для работы с агентом](#)
- [Процесс работы автоматического обновления агента](#)
- [Матрица команд](#)
- [Механизм восполнения пробелов в метриках](#)
- [Поддерживаемые типы метрик устройств](#)
- [Кэширование метрик](#)

# Системные требования Windows

Для работы агента СПП необходима 64-битная операционная система windows 8/Windows Server 2012 или более новые версии.

# Установка в windows

1. Распакуйте файлы поставки
2. Отредактируйте [конфиг](#)
3. Запустите **install.bat** (понадобятся права администратора)

# Конфигурирование

Конфигурация в Windows осуществляется через файл **.env**. Конфигурация в Linux осуществляется через файл **.env** ИЛИ через переменные среды (файл **.env** имеет больший приоритет).

Файл **.env** должен располагаться в одном каталоге с запускаемым файлом агента. В файле **.env** задаются параметры агента в формате **ИМЯ\_ПАРАМЕТРА=ЗНАЧЕНИЕ\_ПАРАМЕТРА**. Каждый параметр должен быть описан в отдельной строке (1 параметр - 1 строка).

## Параметры агента:

- **CFG\_API\_PORT** - порт на котором запускается агент. По умолчанию - 8080
- **CFG\_RPC\_PORT** - порт на котором запускается сервер управления агентом. По умолчанию - 8090
- **CFG\_AGENT\_DEVICES\_LOGS\_ENABLED** - Флаг сбора логов у устройств (devices\_logs)
- **CFG\_RPC\_COMMAND\_TIMEOUT** - таймаут в секундах для команд на сервер управления агентом. По умолчанию - 60
- **CFG\_LOGGING\_LEVEL** - уровень логирования агента. По умолчанию - INFO
- **CFG\_COLLECTOR\_SERVICE\_BASE** - адрес публикации сервиса интеграции (общий порт). По умолчанию - <http://127.0.0.1:8080>
- **CFG\_COLLECTOR\_SERVICE\_DATA** - адрес публикации сервиса интеграции (порт для метрик). По умолчанию - <http://127.0.0.1:8084>
- **CFG\_AGENT\_CHECK\_CRON** - "cron" расписание для отправки состояния агента и получения команд. По умолчанию - \*/5 \* \* \* \*
- **CFG\_AGENT\_METRICS\_UPLOAD\_CRON** - "cron" расписание для загрузки метрик в сервис интеграции. По умолчанию - 5,20,35,50 \* \* \* \*
- **CFG\_AGENT\_METRICS\_UPLOAD\_LIMIT** - максимальное кол-во передаваемых метрик в запросе в сервис интеграции. По умолчанию - 100
- **CFG\_AGENT\_METRICS\_COLLECT\_PASSIVE** - "cron" расписание для опроса пассивных устройств. По умолчанию - 0 \*/1 \* \* \*
- **CFG\_AGENT\_DEVICES\_CRON** - "cron" расписание для обновления информации об устройствах от сервиса интеграции. По умолчанию - 0 \*/1 \* \* \*
- **CFG\_AGENT\_TOKEN** - токен авторизации агента (получение токена агента описано ниже). Значение по умолчанию не задано;
- **CFG\_AGENT\_HOST\_ALIASES** - алиасы агента (агентов) для настройки на устройствах: external - адрес, устанавливаемый на устройстве. По умолчанию - []
- **CFG\_AGENT\_UPGRADE\_AUTO** - флаг автоматического обновления агента (только windows сборка). По умолчанию - 1
- **CFG\_AGENT\_UPGRADE\_AUTO\_CRON** - "cron" расписание для автоматического обновления агента. По умолчанию - \*/30 1-6 \* \* \*
- **CFG\_AGENT\_UPGRADE\_FTP\_HOST** - адрес ftp сервера для автоматического

обновления агента. По умолчанию - localhost:21

- **CFG\_AGENT\_UPGRADE\_FTP\_USER** - имя пользователя ftp сервера для автоматического обновления агента. По умолчанию не задано
- **CFG\_AGENT\_UPGRADE\_FTP\_PASSWORD** - пароль ftp сервера для автоматического обновления агента. По умолчанию не задано
- **CFG\_AGENT\_UPGRADE\_FTP\_PATH** - путь директории на ftp сервере для автоматического обновления агента. По умолчанию - /
- **CFG\_AGENT\_PERCENT\_INCORRECT\_DEVICES\_FOR\_ERROR** - процент неактивных устройств, при котором агент отчитается сервису об ошибке. По умолчанию - 50
- **CFG\_AGENT\_MAX\_DEVICE\_INACTIVITY\_TIMEOUT** - максимальное время бездействия активного устройства. По умолчанию - 60
- **CFG\_INSTANCE\_ID** - Опциональный параметр, Id запущенного агента, необходимо при использовании нескольких экземпляров агента (может быть строкой, не использовать кириллицу и пробелы). Важно заполнять этот параметр **ДО** установки экземпляра агента.
- **CFG\_INSTANCE\_DESCRIPTION** - Опциональный параметр, дополнение к описанию запущенного агента (Отображается в службах, не использовать кириллицу)

---

**При конфигурировании .env файла по дефолту будет использоваться СУБД sqlite, но есть возможность использовать Postgres**

- **CFG\_POSTGRES\_MODE** - активация использования СУБД PostgreSQL (1 - активно, 0 - используем sqlite)
- **CFG\_POSTGRES\_DB** - имя БД
- **CFG\_POSTGRES\_USER** - имя пользователя БД
- **CFG\_POSTGRES\_PASSWORD** - пароль от БД
- **CFG\_POSTGRES\_PORT** - порт для подключения
- **CFG\_POSTGRES\_HOST** - хост

Также дополнительные параметры настройки postgres описаны [тут](#)

При возникновении следующей ошибки следует проверить логин, пароль и имя самой БД

```
File "peewee.py", line 3973, in _connect
File "psycopg2\__init__.py", line 122, in connect
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xc2 in position 61: invalid continuation byte
[37452] Failed to execute script 'main' due to unhandled exception!
```

---

**Для конфигурации режима прокси в .env файле необходимо добавить параметры:**

- **CFG\_PROXY\_CRON** - \*/20 \* \* \* \* (Раз в какой тайминг перезапускать все соединения)
- **CFG\_PROXY\_HOST** - host

- CFG\_PROXY\_MODE - 1
  - CFG\_PROXY\_PATH - /app/agent.data/key (ssh ключ агента)
  - CFG\_PROXY\_USER - user
- 

Возможна конфигурация агента через веб интерфейс. Такой вид конфигурации доступен только на запущенном агенте.

Для запуска конфигурации необходимо зайти на страницу *[http://127.0.0.1:CFG\\_API\\_PORT/config](http://127.0.0.1:CFG_API_PORT/config)*

Где **CFG\_API\_PORT** - это порт на котором запущен агент.

По умолчанию отображаются не все настройки. Для доступа ко всем настройкам:

*[http://127.0.0.1:CFG\\_API\\_PORT/config?full\\_config=true](http://127.0.0.1:CFG_API_PORT/config?full_config=true)*

Агент принимает запросы только локальные запросы (из той же системы на которой сам установлен)

# Настройки датчиков для работы с агентом

Для приема метрик с устройств на самих устройствах нужно настроить HTTP выгрузку на агент.

Url для выгрузки с датчиков составляется по схеме:

http://IP:PORT/collector/metrics - для метрик

http://IP:PORT/collector/alarm - для лога

где **IP** - ip адрес устройства, на котором установлен агент

**PORT** - значение параметра CFG\_API\_PORT из [конфига](#) агента

# Процесс работы автоматического обновления агента

## Описание

Автоматические обновления работают только для собранных версий для windows и работающих в качестве службы windows. При **CFG\_AGENT\_UPGRADE\_AUTO=1** агент в соответствии с CRON-расписанием заданным в **CFG\_AGENT\_UPGRADE\_AUTO\_CRON** запускает процесс проверки обновлений.

1. Подключение в FTP серверу.
  1. Адрес сервера:  
**CFG\_AGENT\_UPGRADE\_FTP\_HOST:CFG\_AGENT\_UPGRADE\_FTP\_HOST**
  2. Логин: **CFG\_AGENT\_UPGRADE\_FTP\_USER**
  3. Пароль: **CFG\_AGENT\_UPGRADE\_FTP\_PASSWORD**
2. Получение списка файлов в папке **CFG\_AGENT\_UPGRADE\_FTP\_PATH** на сервере
3. Определение наличия более новых версий агента по полученному списку
4. Скачивание дистрибутивов всех новых версий
5. Обновление

## Настройки агента

- **CFG\_AGENT\_UPGRADE\_AUTO** - флаг автоматического обновления агента (только windows сборка). По умолчанию - 1
- **CFG\_AGENT\_UPGRADE\_AUTO\_CRON** - "cron" расписание для автоматического обновления агента. По умолчанию - \*/30 1-6 \* \* \*
- **CFG\_AGENT\_UPGRADE\_FTP\_HOST** - адрес ftp сервера для автоматического обновления агента. По умолчанию - localhost:21
- **CFG\_AGENT\_UPGRADE\_FTP\_USER** - имя пользователя ftp сервера для автоматического обновления агента. По умолчанию не задано
- **CFG\_AGENT\_UPGRADE\_FTP\_PASSWORD** - пароль ftp сервера для автоматического обновления агента. По умолчанию не задано
- **CFG\_AGENT\_UPGRADE\_FTP\_PATH** - путь директории на ftp сервере для автоматического обновления агента. По умолчанию - /



# Сборка и публикация дистрибутива

Для сборки дистрибутива необходим python версии 3.11. В корне репозитория агента лежат скрипты для сборки агента

- **win\_build.bat** - скрипт для сборки агента с любого текущего коммита
- **win\_build\_prod.bat** - скрипт для сборки продакшен версии агента. работает только в main ветке репозитория

1. Запускаем соответствующий скрипт для сборки дистрибутива
2. В случае успешной сборки ищем архив с дистрибутивом в папке build (например win\_collector.agent\_1.1.16.zip)
3. Копируем архив с дистрибутивом на FTP сервер

# Матрица команд

	3dh	3db	3dx	3dv	3hx	3dm	3dd	3dt	cm
upload_data	✓	✓	✓	✓	✗	✗[1]	✓	✓	✓
reboot	✓	✓	✓	✓	✗	✓	✗	✓	✗[2]
get_config	✓	✓	✓	✓	✗	✓	✓	✓	✗[2]
set_config	✓	✓	✓	✓	✗	✓	✓	✓	✗[2]
reset_auth	✓[3]	✓	✓[3]	✓	✗	✓[3]	✗	✓	✗[2]
flash	✗	✓	✓	✓	✗	✗	✗	✗	✗[2]
screenshot	✓	✓	✓[4]	✓[4]	✗	✓[5]	✗[7]	✗[7]	✗[2]
video_record	✓	✓	✓	✓	✗	✗[6]	✗[7]	✓	✗[2]

- 1. Устройство не поддерживает выгрузку исторических данных
- 2. Нет поддержки в текущей реализации (есть ли поддержка на самом устройстве не проверялось)
- 3. Используется только пароль. Имя пользователя всегда 'admin'
- 4. Поддержка мультисенсора
- 5. Есть поддержка мультисенсора на устройстве, но пока не реализовано
- 6. Нет поддержки выгрузки по расписанию (выгружается постоянно)
- 7. Видео поток транслируется через веб сокет. документации по формату не нашел

# Механизм восполнения пробелов в метриках

## Проблема

Метрики поступают в агент неравномерно. Некоторые устройства не могут прислать метрики из-за проблем с сетью или по причине того что они выключены. Для исключения пробелов в метриках был реализован механизм восполнения пробелов в метриках

## Реализация

Для реализации механизма понадобилось переработать хранение метрик в локальной базе агента - удаление метрик после отправки заменил на установку флага синхронизации. Сами метрики хранятся в локальной базе агента в течении N дней (задается в переменной **CFG\_AGENT\_METRICS\_AGE\_MAX\_DAYS**).

В механизме используются следующие сущности метрики:

- **start\_timestamp** - начальное время данных метрики
- **end\_timestamp** - конечное время данных метрики
- **device\_id** - id устройства с которого пришла метрика

**gap\_checker** производит анализ метрик в локальной базе

- **CHECKER\_PERIOD\_END** - конец промежутка анализируемых метрик - текущий момент
- **CHECKER\_PERIOD\_START** - начало промежутка анализируемых метрик.  
**CHECKER\_PERIOD\_END** - N дней (задается в переменной **CFG\_AGENT\_METRICS\_CHECK\_GAPS\_DAYS** (по умолчанию = 3))

Промежуток времени не содержит метрики если истинно одно из условий

- если от **CHECKER\_PERIOD\_START** минимального **start\_timestamp** нет ни одной метрики
- если от **end\_timestamp** до **start\_timestamp** следующей метрики нет ни одной метрики

Механизм работает следующим образом:

1. По расписанию (задается в переменной **CFG\_AGENT\_METRICS\_CHECK\_GAPS\_CRON**) запускается функция выявления пробелов (далее **gap\_checker**)

2. При **CFG\_AGENT\_METRICS\_CHECK\_GAPS\_ENABLED != 1** выполнение завершается
3. Для каждого зарегистрированного (`is_new == False`) устройства с выключенным пассивным режимом и поддерживающим команду `upload_data` **gap\_checker** просматривает все метрики в локальной базе за промежуток с **CHECKER\_PERIOD\_START** до **CHECKER\_PERIOD\_END**.
4. Если **gap\_checker** находит промежутки времени, не содержащие ни одной метрики
  1. Скачивание исторических данных с устройства по каждому промежутку

# Поддерживаемые типы метрик устройств

Тип датчика	Форматы
3dx	xml - для версий старше 5.3 json - для версий новее 5.3
3dv	xml
3dh	xml
3db	xml
3dt	json
3dm	json
3dd	json

# Кэширование метрик

В агенте реализовано сохранение метрик в кэш перед записью (Ранее часто встречалась ситуация что метрики в базу пишутся по одному, что замедляло общую работу агента), после включения кэша метрики перед отправкой в БД формируют чанки происходит массовая вставка.

Кэш устанавливается через env файл:

- **CFG\_DATABASE\_CACHE\_SIZE=0** (При значении 0 кэш отключен)

## Важно!

Размер кэша не следует ставить больше чем 999 (В таком случае возможны ошибки, детальное тестирование с большим размером кэша не проводилось).

Кэш работает только с теми метриками у которых в момент записи включена опция игнорирования дубликатов, т.е. метрика не может перезаписываться. (Соответственно команда `upload_data` не использует кэширование)