# Аппаратный SDK для СКУД Elsys на базе КСК Elsys-MB-Net. Общие сведения

В настоящем документе описывается программно-аппаратный продукт – SDK для СКУД Elsys (далее – SDK), реализованный на базе коммуникационного сетевого контроллера Elsys-MB-Net II (далее – КСК).

КСК, являющийся аппаратной основой для SDK (далее – главный КСК, или Master-КСК), обеспечивает взаимодействие клиентского программного обеспечения с оборудованием СКУД Elsys и обеспечивает обслуживание до 64 линий связи СКУД Elsys. В каждую линию связи может входить до 63 контроллеров. Линия связи может быть либо физической линией связи RS-485, либо сетевой группой, представляющей собой логическое объединение контроллеров, поддерживающих IP-протоколы и физически подключаемых по интерфейсу Ethernet.

Для организации обмена данными с линиями связи может быть использовано до 31 ведомых (Slave) КСК Elsys-MB-Net или Elsys-MB-Net II.

Таким образом, подсистема СКУД Elsys, обслуживаемая главным КСК, может обслуживать до 64 линий связи, в число которых могут входить:

* до 32 линий связи RS-485 (одна – встроенная, 31 – обслуживаемые каждым из ведомых КСК);
* до 64 сетевых групп.

Обмен данными с контроллерами в сетевых группах может выполнять как главный КСК, так и ведомые КСК (не более одной сетевой группы на каждый ведомый КСК).

Максимальное количество контроллеров, обслуживаемых SDK – 4032.

Поддерживаемые типы контроллеров – Elsys-MB вариантов исполнения Light, Std, Pro, Pro4, Elsys-MB-SM, Elsys-MB-AC, Elsys-IO/MB, Elsys-RM16, Elsys-AC2, Elsys-CDP.

Обмен данными между клиентским ПО и КСК осуществляется по протоколу http через TCP-порт 80 путём передачи POST-запросов в КСК и анализа ответных посылок.

Совместимость с оборудованием СКУД Elsys.

Elsys-MB – версии 2.63 и выше, Elsys-MB-SM – версии 2.24 и выше, Elsys-IP – 2.03 и выше.

# Алгоритм аутентификации

Запросный пакет клиента представляет собой стандартный POST-запрос.

Структура запросного пакета приведена ниже

POST [URI]

HTTP/1.1

ECNC-Auth: Nonce=”[NONCE]”

Created=“[CREATED]”

Digest=”[DIGEST]”

Date:[Date]

Host: [HOST]

Content-Length: [Content-length]

Expect: 100-continue

Connection: Close

[Content]

URI – уникальный текстовый идентификатор ресурса при работе с функциями SDK (всегда имеет значение </xmlapi/std>).

ECNC-Auth – http-заголовок, содержащий аутентификационную информацию.

Content – содержимое пакета, представляющее собой структурированные данные в формате XML.

Пример запросного http-пакета приведён ниже

**Запрос клиента 1.**

POST /xmlapi/std HTTP/1.1

ECNC-Auth: Nonce="tTzDnq9irS8+Mhw+bppUseQ4xm8=", Created="2019-10-10T08:54:57.371Z", Digest="L/4hJy3ZNVBmazVDFLaiqs82VPY="

Date: Thu, 10 Oct 2019 08:54:59 GMT

Host: 192.168.21.161

Content-Length: 92

Expect: 100-continue

Connection: Close

<Envelope>

<Body>

<CID>10001</CID>

<SIDResp>0</SIDResp>

</Body>

</Envelope>

При обмене данными используется аутентификация по алгоритму Hmac-SHA1.

Назначение полей аутентификационного заголовка:

Nonce – случайный 20-байтовый массив, закодированный в формате base64,

Created – дата и время создания запроса в формате yyyy-MM-ddTHH:mm:ss.fffZ,

Digest – 20-байтовая дайджест-строка, вычисляемая по алгоритму Hmac-SHA1 и закодированная в формате base64.

Дайджест вычисляется на основе массива байтов, представляющего собой конкатенацию байтовых массивов <nonce\_bin> + <created> + <method> + <uri> + <content>, где

nonce\_bin – двоичное представление поля Nonce (20 байт),

created – текстовое значение поля Created,

method – текстовое значение метода POST,

uri – текстовое значение uri, без пробелов (в данном случае “/xmlapi/std”).

В качестве пароля для вычисления дайджеста используется строка, являющаяся конкатенацией строки BrandID (в текущей версии – пустая строка “”) и текстового 8-символьного пароля, используемого IP-оборудованием Elsys при обмене данными.

Аутентификационный заголовок читается корректным, если вычисленное сервером значение дайджеста совпадает с переданным в заголовке, а GMT-время, переданное в поле Created, отличается от времени сервера не более чем на 10 секунд.

Сервер проверяет аутентификационный заголовок, и, если он некорректен, выдаёт ошибку 401.

Пример ответа:

**Ответ сервера 1.**

HTTP/1.1 401 Authentication Error

Connection: close

Date: Sat, 16 Feb 2019 22:58:15 GMT

В ответной посылке сервера передаётся его текущее GMT-время.

Клиент должен запомнить разность текущего времени и времени, сообщённого сервером, и, с учётом этого, сформировать корректное значение поля Created.

При дальнейшей работе рекомендуется синхронизировать время сервера путём передачи соответствующей команды, как показано ниже.

Ниже приведён пример дальнейшего информационного обмена.

**Запрос клиента 2.**

Request:

POST /xmlapi/std HTTP/1.1

ECNC-Auth: Nonce="Oj6UU0oHVzEVy8v47b/FonFY22c=", Created="2019-02-16T22:58:15.022Z", Digest="MDWFd2shMC0cbX2EnDK1Z2BGeYI="

Date: Thu, 10 Oct 2019 08:54:59 GMT

Host: 192.168.21.161

Content-Length: 237

Expect: 100-continue

Connection: Close

<Envelope>

<Body>

<CID>10002</CID>

<SIDResp>0</SIDResp>

<SetDateTime>

<LocalTime>2019-10-10T12:54:59.173Z</LocalTime>

<UTCTime>2019-10-10T08:54:59.173Z</UTCTime>

</SetDateTime>

</Body>

</Envelope>

**Ответ сервера 2.**

HTTP/1.1 200 OK

Content-Length: 333

Content-Type: application/octet-stream

Connection: close

Date: Thu, 10 Oct 2019 08:54:59 GMT

<?xml version="1.0" encoding="utf-8"?>

<Envelope>

<Body>

<CIDResp>10002</CIDResp>

<SID>430</SID>

<ConfigGUID/>

<SetDateTimeResponse>

<UTCTime>2019-10-10T08:54:59</UTCTime>

<LocalTime>2019-10-10T12:54:59</LocalTime>

<Result>true</Result>

</SetDateTimeResponse>

</Body>

</Envelope>

**Запрос клиента 3.**

POST /xmlapi/std HTTP/1.1

ECNC-Auth: Nonce="LoQN9hUf1QA/mut/IHsHiTta1VU=", Created="2019-10-10T08:54:59.000Z", Digest="yU/WPT2LOxGpN1afi98ywuXKshY="

Date: Thu, 10 Oct 2019 08:54:59 GMT

Host: 192.168.21.161

Content-Length: 94

Expect: 100-continue

Connection: Close

<Envelope>

<Body>

<CID>10003</CID>

<SIDResp>430</SIDResp>

</Body>

</Envelope>

**Ответ сервера 3**

HTTP/1.1 200 OK

Content-Length: 153

Content-Type: application/octet-stream

Connection: close

Date: Thu, 10 Oct 2019 08:54:59 GMT

<?xml version="1.0" encoding="utf-8"?>

<Envelope>

<Body>

<CIDResp>10003</CIDResp>

<SID>431</SID>

<ConfigGUID/>

</Body>

</Envelope>

# **Структура запросных и ответных XML-пакетов**

Запросный XML-пакет клиента и ответный XML-пакет сервера представляют собой XML-документы, расположенные в содержимом HTTP-пакета.

Тип кодировки – utf-8.

Данные (Content) запросных и ответных пакетов представляют собой XML-пакеты, содержащие корневой узел <Envelope> и вложенный в него обязательный узел <Body>.

В каждом пакете клиент передаёт узлы <CID> и <SIDResp>, а сервер в ответном пакете передаёт узлы <CIDResp> и <SID>.

<CID> - автоинкрементное 32-битное целое, идентификатор запросного пакета, уникальный, в пределах сеанса обмена идентификатор клиентского HTTP-пакета;

<SID> – автоинкрементное 32-битное целое, идентификатор ответного пакета, уникальный, в пределах сеанса обмена идентификатор серверного HTTP-пакета.

<CIDResp> - отклик сервера, должен быть равен CID принятого запросного пакета. Отклик СIDResp является дополнительным подтверждением приёма HTTP-пакета сервером

<SIDResp> - отклик клиента на последний принятый от сервера пакет. Значение этого поля должно быть равно значению, которое отправил сервер в предыдущей информационной посылке. В начале сессии, и до тех пор, пока он не получит от сервера посылку с полем SID, клиент должен передавать в поле SIDResp значение 0. Получение сервером, значения SIDResp, равного значению SID предыдущей посылки, означает, что клиент успешно принял и обработал предыдущую информационную посылку сервера (в этом случае сервер переходит к передаче следующей порции информации – новых событий, состояний и т. д.). Если клиент отправил значение SIDResp, не равное SID предыдущей посылки сервера (в том числе SIDResp = 0), сервер считает, что подтверждение не получено, и продолжает передавать набор данных из предыдущей посылки.

В каждом ответе сервера содержится поле <ConfigGUID>. Это уникальный идентификатор набора конфигурационных параметров, сообщаемый сервером (16 – байтовый массив в HEX-представлении). Это поле сервер получает от клиента во время загрузки аппаратных настроек (дерево устройств). Если это поле пустое, значит, КСК содержит пустую конфигурацию и в него нужно загрузить дерево устройств. Если это поле имеет значение, соответствующее текстовому HEX-представлению 16-байтового (128-битного) массива, значит, в КСК содержится конфигурация системы. Если GUID из базы данных клиента и сообщённый сервером не совпадают, значит, необходимо перезаписать дерево устройств в память КСК (рекомендуется автоматизировать этот процесс, т. к. работа с несовпадающими конфигурациями не имеет смысла). Рекомендуется для вычисления этого поля использовать стандартный алгоритм генерирования GUID.

В информационной посылке клиента может содержаться от одной до нескольких команд, представленных дочерними элементами узла <Body>, или не быть их вовсе. Каждая команда может содержать дополнительные параметры или большие наборы данных или не содержать их.

В ответной посылке сервер на каждую команду возвращает ответ в XML-узлах, имя которых состоит из имени команды с суффиксом Response, например, как показано ниже.

***Команда***

*<SetDateTime>*

*<LocalTime>2019-10-10T12:54:59.173Z</LocalTime>*

*<UTCTime>2019-10-10T08:54:59.173Z</UTCTime>*

*</SetDateTime>*

***Отклик на команду***

*<SetDateTimeResponse>*

*<UTCTime>2019-10-10T08:54:59</UTCTime>*

*<LocalTime>2019-10-10T12:54:59</LocalTime>*

*<Result>true</Result>*

*</SetDateTimeResponse>*

Отклик на команду должен содержать обязательное поле <Result> (true/false), отображающее результат выполнения команды, и может содержать контекстные данные, зависящие от типа команды.

Размер запросных XML пакетов ограничен размером 500 кБайт.

С учётом этого, клиент должен формировать команды загрузки данных, разбивая их, если необходимо, на отдельные пакеты.

Кроме того, в ответной посылке клиента могут содержаться элементы, не связанные непосредственно с управляющими командами. Как правило, содержание этих узлов отображает асинхронные процессы, выполняемые сервером (текущие события системы, изменение состояний устройств, асинхронное выполнение управляющих команд или команд загрузки данных, и т. д.).

# **Команды SDK**

## 4.1 Назначение основных команд

В таблице описано назначение основных команд SDK.

|  |  |  |
| --- | --- | --- |
|  | Команда | Краткое описание |
| 1 | SetDateTime  (отклик – SetDateTimeResponse) | SetDateTimeResponse |
| 2 | GetDateTime  (отклик – GetDateTimeResponse) | GetDateTimeResponse |
| 3 | UpdSysConfig  (отклик – UpdSysConfigResponse) | Команда загрузки дерева устройств.  Процесс загрузки дерева устройств может состоять из одной команды, или из нескольких (если общий объём XML чрезмерно велик). По окончании выполнения серии команд применяются новые настройки дерева устройств с новым GUID и главный КСК перезагружает протокольные модули. |
| 4 | UpdAPBConfig  (отклик - UpdAPBConfigResponse) |  |
| 5 | GetAPBGUID  (отклик - GetAPBGUIDResponse) |  |
| 6 | ControlCmds  (отклик - ControlCmdsResponse) |  |
| 6 | Changes  (отклик - ChangesResponse) |  |
| 7 | BreakChanges  (реализовать стандартный отклик) |  |
| 8 | ClearSDKMode  (реализовать стандартный отклик) |  |
|  |  |  |
| 9 | GetSysConfig  (отклик – GetSysConfigResponse) | Служебная команда, используемая в JAVA-приложении web-интерфейса начальной настройки |
| 10 | SearchUP  (отклик – SearchUPResponse) | Служебная команда, используемая в JAVA-приложении web-интерфейса начальной настройки |
| 11 | UpdateFW  (отклик - UpdateFWResponse) | Служебная команда, используемая в JAVA-приложении web-интерфейса начальной настройки |
|  |  |  |

## 4.2 Наборы данных, сообщаемые клиентом без явного запроса со стороны сервера

В таблице описано назначение основных наборов данных и узлов, которые могут содержаться в ответе сервера.

|  |  |  |
| --- | --- | --- |
|  | Набор данных | Краткое описание |
| 1 | ErrCode | Код ошибки |
| 2 | DevStates |  |
| 3 | MB\_Events |  |
| 4 | ControlCmdsResults |  |
| 5 | ChangesResults |  |
| 6 | OnlineStatus | При первом соединении |
| 7 | ConnectedDevices |  |
| 8 | DisconnectedDevices |  |
| 9 | ConnectedMBNets |  |
| 10 | DisconnectedMBNets |  |

# **SetDateTime и GetDateTime**

# **UpdSysConfig**

# **Общие сведения**

UpdSysConfig – команда загрузки списка устройств с их основными параметрами (дерева устройств). Процесс загрузки дерева устройств может состоять из одной команды, или из нескольких (если общий объём XML чрезмерно велик). По окончании выполнения серии команд применяются новые настройки дерева устройств с новым GUID и главный КСК перезагружает протокольные модули.

Фрагмент команды приведён ниже.

<UpdSysConfig>

<ConfigGUID>8191EBD472224EB69C1E47F7621E8648</ConfigGUID>

<ExchangeMode>1</ExchangeMode>

<CardSize>3</CardSize>

<ExtendedSettings></ExtendedSettings>

<CmdNo>1</CmdNo>

<CmdCount>4</CmdCount>

<MBNets>

<MBNet>…

<Lines>

<Line> … </Line>

…

<Line>… </Line>

</Lines>

</MBNet>

…

<MBNet>…

</MBNet>

</MBNets>

<Lines>

<Line> … </Line>

…

<Line> … </Line>

</ Lines>

<Devices>

<Device>

</Device>

…

<Device> …

</Device>

<Devices>

В каждой команде UpdSysConfig передаются параметры CmdNo и CmdCount, отображающие номер команды и общее число команд в списке (CmdCount >= 1, 1 <=CmdNo <= CmdCount)

По окончании загрузки дерева устройств (при выполнении условия CmdNo = CmdCount ) главный КСК получает весь набор данных, необходимый для обслуживания подсистемы СКУД.

Для каждого из устройств передаются уникальный ID, имя устройства и набор параметров, определяющих принадлежность устройства к той или иной линии связи (адрес линии связи, адрес контроллера, адрес устройства в составе контроллера, тип устройства).

Для КСК, линий связи и контроллеров передаются также ряд дополнительных параметров, необходимых для обмена данными и корректного формирования управляющих и инициализационных команд.

В составе команды <UpdSysConfig> могут быть узлы <CommonSettings>, <MBNets>, <Devices>, <Lines>, <SubDevices>, <MNParts>, <MNParGroups>.

<CommonSettings>, <MBNets>, <Devices> - обязательные узлы.

<MNParts>, <MNParGroups> - необязательные узлы.

Узлы <Lines> и <SubDevices> могут находиться как на корневом уровне (т. е. быть дочерними как для <UpdSysConfig>), так и принадлежать узлам <MBNets> <MBNet> и <Devices><Device> соответственно.

Основные ограничения, которые необходимо учитывать при разбиении данных на отдельные команды:

1. Конфигурация линий связи должна содержаться в узлах <MBNets> и <Lines> и должна быть передана в первой инициализационной команде.
2. Информация о считывателях и дверях, относящихся к одному контроллеру, должна содержаться в одной команде.

Требования к данным дерева устройств:

1. Идентификаторы устройств должны быть уникальными и должны быть в диапазоне 0 < ID <= 2147483647 (должны быть положительными 32-битными числовыми значениями).
2. Узел <MBNet>, имеющий свойство <MBNetType>Master</MBNetType>, должен быть в системе только один.

# **Типы устройств**

# **Общие настройки системы**

# **Список КСК и их свойства**

Ниже приведён фрагмент команды UpdSysConfig, содержащий настройки КСК.

<UpdSysConfig>

…

<MBNets>

<MBNet>

<IPAddr>192.168.21.161</IPAddr>

<MBNetType>Master</MBNetType>

<MNAddr>1</MNAddr>

<ID>1</ID>

<Name>КСК 1</Name>

<VersionNo>770</VersionNo>

<SubnetMask>255.255.255.0</SubnetMask>

<ExchangeEnabled>true</ExchangeEnabled>

<TranslateILocks>false</TranslateILocks>

<Lines>

<Line>…</Line>

<Line>…</Line>

</Lines>

</MBNet>

<MBNet>… </MBNet>

<MBNets>

В качестве дочерних узлов должны быть добавлены линии связи, обслуживаемые этим КСК (узел <Lines>, см. ниже).

**Внимание! Линия RS-485, даже при отсутствии в ней контроллеров, должна быть добавлена в конфигурацию КСК обязательно!**

# **Список линий связи и их свойства**

Список <Lines> может быть дочерним элементом узла <MBNet> (в этом случае он содержит один или два элемента <Line>, элемент с <RS485>true</RS485> обязательный; для него MasterMNAddr и ExchangeMNAddr должны быть равны и совпадать с адресом КСК) либо узла <UpdSysConfig>.

Пример списка <Lines> приведён ниже.

<Lines>

<Line>

<LineAddr>1</LineAddr>

<ID>2</ID>

<MasterMNAddr>1</MasterMNAddr>

<ExchangeMNAddr>1</ExchangeMNAddr>

<Name>Линия связи RS485</Name>

<RS485>true</RS485>

<BaudRate>115200</BaudRate>

<Multimaster>false</Multimaster>

<ForceAPB>false</ForceAPB>

<NoCheckAPB>false</NoCheckAPB>

<ExtendedSettings>

<TransmissionDelay>0</TransmissionDelay>

<ReadInterval>0</ReadInterval>

<ResponseTimeout>0</ResponseTimeout>

<MaxErrorCount>0</MaxErrorCount>

</ExtendedSettings>

</Line>

<Line>

<LineAddr>257</LineAddr>

<ID>3</ID>

<MasterMNAddr>0</MasterMNAddr>

<ExchangeMNAddr>1</ExchangeMNAddr>

<Name>Сетевая группа 1</Name>

<RS485>false</RS485>

<NGAddr>1</NGAddr>

<ForceAPB>false</ForceAPB>

<NoCheckAPB>false</NoCheckAPB>

<ExchangeMode>1</ExchangeMode>

</Line>

</Lines>

Каждая линия связи должна иметь уникальный в пределах всего списка устройств ID (0 < ID <= 2147483647). Кроме того, каждая линия должна иметь уникальный в пределах списка линий связи адрес линии связи LineAddr (0 < LineAddr < 512). Рекомендуется (но не обязательно к выполнению) присваивать линиям связи RS-485 адрес, совпадающий с адресом родительского КСК, а для сетевых групп адрес из диапазона 256.. 511. Для всех устройств системы, описываемых в следующих командах, должна быть корректно описана ссылка на адрес линии связи, которой они принадлежат (поле <LineAddr>).

<RS485> - тип линии связи (true – RS-485, false – сетевая группа)

<MasterMNAddr> - адрес КСК, осуществляющего опрос линии связи (для сетевой группы, опрашиваемой непосредственно это поле должно быть равным нулю; для линии RS-485 должно быть равен адресу родительского КСК);

<ExchangeMNAddr> - адрес КСК, с которым осуществляется межконтроллерный обмен (если для сетевой группы обмен с КСК выключен, это поле должно быть равно нулю; для линии RS-485 должно быть равен адресу родительского КСК);

<ForceAPB> - усиленный антипассбэк;

<NoCheckAPB> - не проверять исправность областей контроля (зон антипассбэка).

Кроме того, есть ряд специфических настроек для линии RS-485 и сетевой группы.

# **Список контроллеров и их свойства**

# **Список устройств и их свойства**

# **ControlCmds**

# **Changes**

# Общие сведения

Команды, предназначенные для КСК и контроллеров доступа Elsys (в частности, содержащие конфигурации контроллеров, содержимое базы персонала, конфигурация охранной подсистемы и информацию об изменении этих настроек), не могут быть выполнены немедленно, поэтому они выполняются асинхронно.

Большинство настроек и данных для КСК и контроллеров доступа Elsys (конфигурации контроллеров, содержимое базы персонала, конфигурация охранной подсистемы) передаются в Master-КСК в виде XML-команд в узле <Changes>. Эти данные не сохраняются в памяти Master-КСК, а используются для формирования списков низкоуровневых команд, предназначенных для асинхронного выполнения.

Клиент передаёт серии команд в виде списка <Changes>, состоящего из элементов Change, каждый из которых состоит из элементов <Item> (в списке <Items>).

<Changes>

<Change>

<ChangeType>CHANGETYPE\_VALUE</ChangeType>

<ID>28891</ID>

<Items>

<Item>

<Action>ACTION\_VALUE</Action>

...

<Item>

</Items>

</Change>

</Changes>

Каждый элемент Change представляет собой элементарную серию команд, имеющую уникальный идентификатор.

КСК принимает эти серии команд и формирует для каждой из серий команд список низкоуровневых команд для КСК и контроллеров.

Результат обработки команды <Changes> сервер сообщает в узле <ChangesResponse> и его дочерних узлах <Result> и <AppendChanges>.

<ChangesResponse>

<Result>true</Result>

<AppendChanges>

<Change>

<ID>28891</ID>

<CmdCount>546</CmdCount>

</Change>

</AppendChanges>

</ChangesResponse>

Здесь <AppendChanges> – список команд, принятых к исполнению.

Эти команды не могут быть выполнены немедленно, поэтому они заносятся в очередь команд, которые последовательно передаются в опрашиваемые устройства.

Master-КСК, имеющий активные списки команд, непрерывно передаёт текущие их текущие состояния (если список команд ещё не стартовал, его состояния не передаются).

Полностью выполненные списки команд Master-КСК автоматически удаляет, убедившись, что клиент их принял (фактом подтверждения является непрерывность информационного обмена, определяемая по равенству SIDResp значению SID, сообщённого сервером в предыдущей посылке).

<ChangesResults>

<ChangesCount>2</ChangesCount>

<MaxChangesCount>128</MaxChangesCount>

<CmdCount>84</CmdCount>

<MaxCmdCount>65536</MaxCmdCount>

<CmdListCount>8</CmdListCount>

<MaxCmdListCount>32768</MaxCmdListCount>

<Changes>

<Change>

<ID>3390</ID>

<Status>1</Status>

<CmdPos>56</CmdPos>

<CmdCount>76</CmdCount>

</Change>

</Changes>

</ChangesResults>

Общая ёмкость списков команд ограничена встроенной памятью Master-КСК.

В некоторых случаях возможна ситуация, при которой Master-КСК вернёт результат

<ChangesResponse>

<Result>false</Result>

<Busy>true</Busy>

<ErrCode>0</ErrCode>

</ChangesResponse>

Признак Busy = true означает, что причина неудачного выполнения команды во временной нехватке памяти под списки команд, поэтому клиент должен дождаться завершения выполнения предыдущих серий команд, после чего повторно передать серию.

**Примечание. Если ChangesCount = 0, CmdCount = 0 и CmdListCount = 0, а сервер вернул Busy = true, то, вероятно, клиентское приложение сформировало слишком большой набор списков команд (при попытке подготовки команд не хватило памяти в главном КСК), и его следует разбить на части.**

# Начальная загрузка базы данных пропусков

### Общие сведения

Общий вид команд приведён ниже.

<Changes>

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3388</ID>

<Devices>

<ID>[ID1]</ID>

<ID>[ID2]</ID>

<ID>…</ID>

</Devices>

<Items>

<Item>

<Action>[Action]</Action>

… [Data]

</Item>

</Items>

</Change>

…

</Changes>

или

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3388</ID>

<Items>

<Item>

<Action>[Action]</Action>

<Devices>

<ID>[ID1]</ID>

<ID>[ID2]</ID>

<ID>…</ID>

</Devices>

… [Data]

</Item>

</Items>

</Change>

</Changes>

Для всех команд загрузки базы бюро пропусков список контроллеров-адресатов <Devices> может присутствовать как на уровне <Change> (в этом случае он общий для всех команд списка <Change><Items><Item>), либо на уровне <Change><Items><Item> (в этом случае для каждого элемента списка команд используется свой список адресатов). Приоритетным является список на уровне <Change> (если он есть, то списки на уровнях <Change><Items><Item> игнорируются, и их вообще не должно быть). Если же список <Change><Devices> отсутствует, то каждый узел <Change><Items><Item> должен иметь список контроллеров-адресатов <Devices>.

*Среди команд начальной загрузки данных есть команды полного удаления данных, но нет команд удаления отдельных элементов (карт, уровней доступа, временных блоков), так как при начальной загрузке данных в них нет необходимости.*

### Удаление всех уровней доступа (DelAllAL)

Пример команды:

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3388</ID>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Items>

<Item>

<Action>DelAllAL</Action>

</Item>

</Items>

</Change>

или

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3388</ID>

<Items>

<Item>

<Action>DelAllAL</Action>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

</Item>

</Items>

</Change>

Во всех примерах команд начальной загрузки данных, приведённых ниже, список устройств передаётся на уровне <Change>.

### Удаление всех карт (DelAllCards)

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3392</ID>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Items>

<Item>

<Action>DelAllCards</Action>

</Item>

</Items>

</Change>

### Удаление всех временных блоков (DelAllTB)

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3390</ID>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Items>

<Item>

<Action>DelAllTB</Action>

</Item>

</Items>

</Change>

### Удаление всех праздничных дней (DelAllHolidays)

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3493</ID>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Items>

<Item>

<Action>DelAllHolidays</Action>

</Item>

</Items>

</Change>

### Добавление уровня доступа (AddAL)

<Changes>

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3389</ID>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Items>

<Item>

<Action>AddAL</Action>

<ALItems>

<ALItem>

<TmBlockNo>1</TmBlockNo>

<TmBlockID>1</TmBlockID>

<RdrID>17</RdrID>

</ALItem>

<ALItem>

<TmBlockNo>1</TmBlockNo>

<TmBlockID>1</TmBlockID>

<RdrID>18</RdrID>

</ALItem>

...

</ALItems>

<ALNo>1</ALNo>

<ALID>1</ALID>

</Item>

</Items>

</Change>

</Changes>

Список адресатов в этой команде косвенно содержится в списке элементов уровня доступа <ALItems>. Низкоуровневые инициализационные команды будут сформированы для контроллеров, которые входят в список Devices, а считыватели которых входят в ALItems этого уровня доступа.

**Примечание. При подготовке данных для SDK рекомендуется для оптимизации трафика сформировать подмножество элементов уровня доступа ALItems, соответствующее списку контроллеров.**

### Добавление пропуска (AddCard)

<Changes>

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3392</ID>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Items>

<Item>

<Action>AddCard</Action>

<CardNo>0x0000000003E9</CardNo>

<ALNo>1</ALNo>

<ALID>1</ALID>

<PINCode>101</PINCode>

<IsTmpCard>false</IsTmpCard>

<NoAPB>false</NoAPB>

<Options>0</Options>

<Priv>0</Priv>

<CardAction>0</CardAction>

<StartDate>2019-10-07T00:00:00</StartDate>

<EndDate>2020-02-07T00:00:00</EndDate>

<DevParams>

<DevParamsItem>

<ID>4</ID>

<Options>0</Options>

<Priv>0</Priv>

<CardAction>0</CardAction>

</DevParamsItem>

</DevParams>

</Item>

<Item>...</Item>

...

<Item>...</Item>

</Items>

</Change>

</Changes>

<DevParams>, <DevParamsItem>, <Options>, <Priv> и <CardAction>.

### Добавление временного блока (AddTB)

<Changes>

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3390</ID>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Items>

<Item>

<Action>AddTB</Action>

<TmBlockID>1</TmBlockID>

<TmBlockNo>1</TmBlockNo>

<Intervals>

<TimeInterval>

<StartTime>0001-01-01T08:00:00</StartTime>

<EndTime>0001-01-01T13:00:00</EndTime>

<ActiveDays>127</ActiveDays>

<ActiveHolidays>1</ActiveHolidays>

<Period>7</Period>

<StartDate>2019-10-07T00:00:00</StartDate>

<EndDate>2020-02-07T00:00:00</EndDate>

<NoUsingHolidays>false</NoUsingHolidays>

</TimeInterval>

<TimeInterval>

<StartTime>0001-01-01T14:00:00</StartTime>

<EndTime>0001-01-01T23:00:00</EndTime>

<ActiveDays>127</ActiveDays>

<ActiveHolidays>1</ActiveHolidays>

<Period>7</Period>

<StartDate>2019-10-07T00:00:00</StartDate>

<EndDate>2020-02-07T00:00:00</EndDate>

<NoUsingHolidays>false</NoUsingHolidays>

</TimeInterval>

</Intervals>

</Item>

...

</Items>

</Change>

</Changes>

### Добавление праздничных дней (AddHolidays)

<Changes>

<Change>

<ChangeType>InitPersDB</ChangeType>

<ID>3493</ID>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Items>

<Item>

<Action>AddHolidays</Action>

<Items>

<Item>

<Date>2019-01-01T00:00:00</Date>

<Type>9</Type>

</Item>

<Item>

<Date>2019-01-02T00:00:00</Date>

<Type>9</Type>

</Item>

...

</Items>

</Item>

</Items>

</Change>

</Changes>

# Загрузка изменений базы данных пропусков

### Общие сведения

Команды, описанные в этом разделе, предназначены для доставки в оборудование изменений в базе данных пропусков. Каждой транзакции бюро пропусков соответствует серия команд (<Change>), которая может состоять из нескольких команд (<Change><Items><Item>). Многие команды загрузки изменений схожи по внутренней структуре с командами начальной загрузки данных (AddCard и UpdCard, AddTB и UpdTB, AddAL и UpdAL, AddHolidays и UpdHolidays) и отличаются способом формирования списков адресатов.

Все команды загрузки изменений имеют следующую структуру:

<Changes>

<Change>

<ChangeType>ChangePersDB</ChangeType>

<ID>ID\_VALUE</ID>

<Items>

<Item>

<Action>ACTION\_VALUE</Action>

<ForAllDevices>..</ForAllDevices>

<Devices>..</Devices>

<Readers>..</Readers>

… (ДАННЫЕ)

и др. ...

</Item>

</Items>

</Change>

</Changes>

Список адресатов каждой команды описывается узлами <ForAllDevices>, <Devices> или <Readers>.

Эти узлы интерпретируются в SDK следующим образом (для всех команд, кроме команды <UpdAL>). Если присутствует узел <ForAllDevices> в значении true, команда формируется для всех контроллеров. Иначе, команда формируется для контроллеров, чьи ID входят в список <Devices>. Если ForAllDevices = false и отсутствует узел Devices, команда формируется для контроллеров, обслуживающих считыватели, чьи ID входят в список Readers.

Для команды обновления уровня доступа <UpdAL> параметр <ForAllDevices> интерпретируется иначе, чем для других команд, а параметры <Devices> и <Readers> отсутствуют (см. ниже, описание команды).

Если набор адресатов команды изменения данных пустой, будет сформирована пустая серия команд с признаком успешного выполнения.

Рекомендуется при создании клиентского программного обеспечения во всех случаях выбрать и использовать только один из вариантов передачи списка адресатов для команд загрузки изменений. Вариант с ForAllDevices = true использовать не рекомендуется, так как он, хотя и несколько упрощает разработку клиентского программного обеспечения и уменьшает объём XML-пакетов, но приводит к формированию избыточного количества команд и снижает эффективность обмена данными с контроллерами.

### Удаление пропуска (DelCard)

Ниже приведены три варианта команды DelCard с разными списками адресатов.

<Changes>

<Change>

<ChangeType>ChangePersDB</ChangeType>

<ID>1002</ID>

<Items>

<Item>

<Action>DelCard</Action>

<ForAllDevices>true</ForAllDevices>

<CardNo>0x0000000003E9</CardNo>

</Item>

</Items>

</Change>

</Changes>

<Changes>

<Change>

<ChangeType>ChangePersDB</ChangeType>

<ID>1003</ID>

<Items>

<Item>

<Action>DelCard</Action>

<Devices>

<ID>4</ID>

<ID>19</ID>

</Devices>

<CardNo>0x0000000003EA</CardNo>

</Item>

</Items>

</Change>

</Changes>

<Changes>

<Change>

<ChangeType>ChangePersDB</ChangeType>

<ID>1004</ID>

<Items>

<Item>

<Action>DelCard</Action>

<Readers>

<ID>173</ID>

<ID>174</ID>

<ID>965</ID>

<ID>966</ID>

<ID>993</ID>

</Readers>

<CardNo>0x0000000003EB</CardNo>

</Item>

</Items>

</Change>

</Changes>

### Добавление (изменение) пропуска (UpdCard)

Команда идентична AddCard, за исключением обработки списка адресатов.

<Changes>

<Change>

<ChangeType>ChangePersDB</ChangeType>

<ID>1003</ID>

<Items>

<AddCard>

<Action>UpdCard</Action>

<Devices>

<ID>4</ID>

<ID>19</ID>

</Devices>

<Options>5</Options>

<Priv>1</Priv>

<CardAction>1</CardAction>

<DevParams>

<DevParamsItem>

<ID>161</ID>

<Options>65</Options>

<Priv>1</Priv>

<CardAction>3</CardAction>

</DevParamsItem>

<DevParamsItem>

<ID>175</ID>

<Options>65</Options>

<Priv>1</Priv>

<CardAction>3</CardAction>

</DevParamsItem>

<DevParamsItem>

<ID>635</ID>

<Options>65</Options>

<Priv>1</Priv>

<CardAction>3</CardAction>

</DevParamsItem>

</DevParams>

<CardNo>0x000000C1ABCD</CardNo>

<ALNo>1</ALNo>

</AddCard>

</Items>

</Change>

</Changes>

В команде вместо узла <Devices> может быть узел <Readers> или <ForAllDevices>true</ForAllDevices>

### Удаление уровня доступа (DelAL)

<Changes>

<Change>

<ChangeType>ChangePersDB</ChangeType>

<ID>1002</ID>

<Items>

<Item>

<Action>DelAL</Action>

<Devices>

<ID>4</ID>

<ID>19</ID>

</Devices>

<ALNo>11</ALNo>

<ALID>11</ALID>

</Item>

</Items>

</Change>

</Changes>

### Изменение уровня доступа (UpdAL)

Ниже приведены два примера примеры команды UpdAL.

Для команды обновления уровня доступа <UpdAL> параметр <ForAllDevices> интерпретируется несколько иначе, чем для других команд, а параметры <Devices> и <Readers> отсутствуют.

Вариант 1 - <ForAllDevices> отсутствует (или = false)

<Changes>

<Change>

<ChangeType>ChangePersDB</ChangeType>

<ID>1001</ID>

<Items>

<Item>

<Action>UpdAL</Action>

<ALItems>

<ALItem>

<TmBlockNo>1</TmBlockNo>

<TmBlockID>1</TmBlockID>

<RdrID>17</RdrID>

</ALItem>

<ALItem>

<TmBlockNo>1</TmBlockNo>

<TmBlockID>1</TmBlockID>

<RdrID>18</RdrID>

</ALItem>

…

<ALNo>1</ALNo>

<ALID>1</ALID>

</Item>

</Items>

</Change>

</Changes>

Вариант 2 - <ForAllDevices> = true

<Item>

<Action>UpdAL</Action>

<ForAllDevices>true</ForAllDevices>

<ALItems>

<ALItem>

<TmBlockNo>1</TmBlockNo>

<TmBlockID>1</TmBlockID>

<RdrID>17</RdrID>

</ALItem>

<ALItem>

<TmBlockNo>1</TmBlockNo>

<TmBlockID>1</TmBlockID>

<RdrID>18</RdrID>

</ALItem>

…

<ALNo>1</ALNo>

<ALID>1</ALID>

</Item>

В первом случае команды обновления уровней доступа формируются для всех контроллеров, обслуживающих считыватели, имеющие ID из списка ALItems.

Во втором случае дополнительно формируются команды удаления уровня доступа для всех контроллеров, где не содержится элементов ALItems.

**Внимание! Очень важно при формировании команд обновления уровня доступа передавать его полное подмножество для каждого контроллера.**

**Например, если контролер обслуживает считыватели с RdrID = 17 и RdrID = 18, и произошло изменение полномочий в считывателе с RdrID = 18 (изменение временного блока с 1 на 2), то следует передавать все элементы уровня доступа, которые хранятся в контроллере (в данном случае – 17 и 18), как показано ниже:**

**<ALItems>**

**<ALItem>**

**<TmBlockNo>1</TmBlockNo>**

**<TmBlockID>1</TmBlockID>**

**<RdrID>17</RdrID>**

**</ALItem>**

**<ALItem>**

**<TmBlockNo>2</TmBlockNo>**

**<TmBlockID>2</TmBlockID>**

**<RdrID>18</RdrID>**

**</ALItem>**

**</ALItems>**

**Если же в команде UpdAL будет присутствовать только один элемент (см. ниже), то такой набор данных SDK воспримет как удаление элемента со считывателем, имеющим RdrID = 17.**

**<ALItems>**

**<ALItem>**

**<TmBlockNo>2</TmBlockNo>**

**<RdrID>18</RdrID>**

**</ALItem>**

**</ALItems>**

### Удаление временного блока (DelTB)

<Changes>

<Change>

<ChangeType>ChangePersDB</ChangeType>

<ID>1005</ID>

<Items>

<Item5>

<Action>DelTB</Action>

<Devices>

<ID>4</ID>

<ID>19</ID>

</Devices>

<TmBlockNo>1</TmBlockNo>

<TmBlockID>1</TmBlockID>

</Item5>

</Items>

</Change>

</Changes>

### Изменение временного блока (UpdTB)

Команда идентична AddTB, за исключением списка адресатов.

<Changes>

<Change>

<ChangeType> ChangePersDB </ChangeType>

<ID>3390</ID>

<Items>

<Item>

<Action>UpdTB</Action>

<TmBlockID>1</TmBlockID>

<TmBlockNo>1</TmBlockNo>

<Devices>

<ID>4</ID>

<ID>34</ID>

<ID>66</ID>

<ID>82</ID>

</Devices>

<Intervals>

<TimeInterval>

<StartTime>0001-01-01T08:00:00</StartTime>

<EndTime>0001-01-01T13:00:00</EndTime>

<ActiveDays>127</ActiveDays>

<ActiveHolidays>1</ActiveHolidays>

<Period>7</Period>

<StartDate>2019-10-07T00:00:00</StartDate>

<EndDate>2020-02-07T00:00:00</EndDate>

<NoUsingHolidays>false</NoUsingHolidays>

</TimeInterval>

<TimeInterval>

<StartTime>0001-01-01T14:00:00</StartTime>

<EndTime>0001-01-01T23:00:00</EndTime>

<ActiveDays>127</ActiveDays>

<ActiveHolidays>1</ActiveHolidays>

<Period>7</Period>

<StartDate>2019-10-07T00:00:00</StartDate>

<EndDate>2020-02-07T00:00:00</EndDate>

<NoUsingHolidays>false</NoUsingHolidays>

</TimeInterval>

</Intervals>

</Item>

...

</Items>

</Change>

</Changes>

### Обновление списка праздников (UpdHolidays)

Команда идентична AddHolidays, за исключением обработки списка адресатов.

<Changes>

<Change>

<ChangeType> ChangePersDB </ChangeType>

<ID>3493</ID>

<Items>

<Item>

<Action>UpdHolidays</Action>

<Devices>

<ID>4</ID>

<ID>34</ID>

</Devices>

<Items>

<Item>

<Date>2019-01-01T00:00:00</Date>

<Type>9</Type>

</Item>

<Item>

<Date>2019-01-02T00:00:00</Date>

<Type>9</Type>

</Item>

...

</Items>

</Item>

</Items>

</Change>

</Changes>

# Приём данных от сервера

# Приём событий

<MB\_Events>

<Item>

<DateTime>2019-10-25T15:13:00</DateTime>

<ID>16</ID>

<DevType>3</DevType>

<MessageCode>14</MessageCode>

<DevState>10</DevState>

<CardNo>0x000000C1ABCD</CardNo>

</Item>

<Item>

<DateTime>2019-10-25T15:13:00</DateTime>

<ID>16</ID>

<DevType>3</DevType>

<MessageCode>56</MessageCode>

<DevState>10</DevState>

<CardNo>0x000000C1ABCD</CardNo>

</Item>

<Item>

<DateTime>2019-10-25T15:13:00</DateTime>

<ID>16</ID>

<DevType>3</DevType>

<MessageCode>119</MessageCode>

<DevState>10</DevState>

<CardNo>0x000000C1ABCD</CardNo>

</Item>

<Item>

<DateTime>2019-10-25T15:13:00</DateTime>

<ID>16</ID>

<DevType>3</DevType>

<MessageCode>27</MessageCode>

<DevState>10</DevState>

<CardNo>0x000000C1ABCD</CardNo>

</Item>

</MB\_Events>

# Приём состояний устройств

<DevStates>

<Item>

<ID>16</ID>

<DevState>10</DevState>

</Item>

<Item>

<ID>17</ID>

<DevState>10</DevState>

</Item>

</DevStates>

# Приём состояний глобальных разделов

<MNPartStates>

<Item>

<ID>1010</ID>

<State>1</State>

</Item>

<Item>

<ID>1009</ID>

<State>1</State>

</Item>

<Item>

<ID>1001</ID>

<State>1</State>

</Item>

<Item>

<ID>1002</ID>

<State>1</State>

</Item>

<Item>

<ID>1003</ID>

<State>1</State>

</Item>

<Item>

<ID>1004</ID>

<State>1</State>

</Item>

</MNPartStates>

# Состояние наличия связи с устройствами

<OnlineStatus>

<MBNets>0300000000000000</MBNets>

<OnlineMBNets>0300000000000000</OnlineMBNets>

<Lines>

<Line>

<LineAddr>1</LineAddr>

<MNAddr>1</MNAddr>

<RS485>1</RS485>

<Devices>0700000000000000</Devices>

<OnlineDevices>0700000000000000</OnlineDevices>

</Line>

<Line>

<LineAddr>257</LineAddr>

<MNAddr>1</MNAddr>

<RS485>0</RS485>

<Devices>8001000000000000</Devices>

<OnlineDevices>8001000000000000</OnlineDevices>

</Line>

<Line>

<LineAddr>2</LineAddr>

<MNAddr>2</MNAddr>

<RS485>1</RS485>

<Devices>0100000000000000</Devices>

<OnlineDevices>0100000000000000</OnlineDevices>

</Line>

<Line>

<LineAddr>258</LineAddr>

<MNAddr>2</MNAddr>

<RS485>0</RS485>

<Devices>0300000000000000</Devices>

<OnlineDevices>0300000000000000</OnlineDevices>

</Line>

</Lines>

</OnlineStatus>

<ConnectedDevices>

<Item>4</Item>

<Item>19</Item>

<Item>112</Item>

<Item>34</Item>

<Item>49</Item>

<Item>66</Item>

<Item>82</Item>

<Item>97</Item>

</ConnectedDevices>

<ConnectedMBNets>

<Item>1</Item>

<Item>64</Item>

</ConnectedMBNets>

# Приём результатов выполнения управляющих команд

# Приём результатов выполнения инициализационных команд

DelAL

UpdAL

UpdALItems - for new CU!

DelCard

UpdCard

DelTB

UpdTB

UpdHolidays