Проект Тип оборудования AccessController

Оглавление

1. ОСНОВНЫЕ ПОНЯТИЯ И ТЕРМИНЫ 4

2. ОПИСАНИЕ ТИПА 5

2.1. Наименование типа и русскоязычный синоним 5

2.2. Краткое описание функциональности типа 5

3. АЛГОРИТМЫ 6

3.1. Общие алгоритмы 6
Создание нового экземпляра оборудования 6
Использование экземпляра оборудования 6
Удаление экземпляра оборудования 7

3.2. Специфические алгоритмы 7
Механизм настройки экземпляра устройства 7
Алгоритм предоставления доступа 7

4. НАСТРОЙКИ 7

4.1. Общие настройки 7
DefaultTimeOut 8
EventLogEnabled 8

4.2. Специфические настройки 8

5. МЕТОДЫ 8

5.1. Служебные методы 8
Init 8
InitProxy1C 9
Open 9
Close 9
GetDeviceInfo 10

5.2. Общие методы 11
GetSettings 11
SetSettings 12
ShowSettingsDlg 12
CheckHealth 13
Enable 13
Disable 14

5.3. Специфические методы 14
GetControllers 14
UploadTimeProfiles 15
UploadHolidays 16
UploadCards 16
DeleteCards 17
SetMode 18
GetEvents 18
DeleteEvents 19
GetStatus 20
GetDateTime 21
SetDateTime 21

6. ФУНКЦИИ ОБРАТНОГО ВЫЗОВА 22
GetAppProperty 22
WriteLog 22
ErrorEvent 23
DataEvent 23

1. Основные понятия и термины

Термин	Описание
Подключаемое оборудование	Любое торговое или промышленное оборудование, использующееся клиентским приложением
Тип оборудования	Регламентированное документацией множество, объединяющее устройства с одинаковым функциональным назначением
Драйвер оборудования, Драйвер	Программный компонент с возможностью отдельной установки или обновления в системе и предоставляющий интерфейс управления и взаимодействия с устройством определенной модели. Предоставляемый драйвером интерфейс должен строго соответствовать описанию типа оборудования по данной технологии, но может иметь расширения (дополнительные методы и/или параметры) для выполнения функций, не регламентированных в документах описания системы
Модель, Модель оборудования	Множество устройств одного типа оборудования, одинаковых или близких по своим характеристикам, управляемых драйвером устройства. Модель оборудования определяется драйвером устройства. В случае, если один драйвер устройства позволяет единообразно управлять одной из нескольких сходных моделей физических устройств, для системы все эти устройства являются одной и той же моделью. Напротив, если для одного и того же физического устройства существует несколько разных драйверов, для системы они являются разными моделями.
Экземпляр оборудования	Совокупность файлов и данных о физическом устройстве и драйвере (идентификатор, наименование, список настроек и т. п.), хранимых и используемых для работы с данным устройством системой управления подключаемым оборудованием и приложением-клиентом.
Клиентское приложение	Любое приложение, использующее для управления подключаемым оборудованием описанные в документации по технологии интерфейсы и алгоритмы взаимодействия с драйверами устройств.
Менеджер оборудования	Часть клиентского приложения, в которой сосредоточены сервисные процедуры и регламентированные документацией интерфейсы взаимодействия с оборудованием.
Настройки устройства	Набор именованных значений, простых типов данных: String, Number, DateTime, Boolean, хранящийся вне клиентского приложения в файловом хранилище настроек. Для каждого определенного в системе устройства имеется свой набор настроек. Настройки устройства определяют параметры и режим функционирования устройства. Настройки доступны клиентскому приложению для чтения и записи с помощью специальных методов.
Уникальный идентификатор	В этом проекте – текстовое представление GUID. Идентификатор содержит 36 символов в верхнем регистре, не заключенных в фигурные скобки. Пример: 8A8910EC-78F5-41A5-9E18-7C28992CF580
Публикуемый метод	Один из определенных в описании интерфейса общих методов, либо методов типа. Все публикуемые методы имеют одинаковую структуру параметров. Входные и возвращаемые методом данные помещаются в два одномерных массива SafeArray
Функции обратного вызова	Регламентируемые данной документацией функции клиентского приложения, доступные для вызова через интерфейс IDIspatch драйвером оборудования. Функции обратного вызова предназначены для информирования приложения о событиях или передачи данных.

Термин	Описание
Событие	Событие возникает как результат изменения состояния устройства. Событие может быть вызвано внешним воздействием, либо внутренними процессами в оборудовании. Для передачи событий приложению драйвер устройства использует функции обратного вызова клиентского приложения
Объект-абстракт	Объект драйвера устройства, не ассоциированный с экземпляром оборудования. Любой объект драйвера является объектом-абстрактом до выполнения метода Open.У объекта-абстракта допустим вызов только тех методов, которые не требуют взаимодействия с физическим устройством. Объект-абстракт не может использовать функции обратного вызова, поскольку не имеет собственного идентификатора в системе.
Объект драйвера	Объект драйвера устройства, созданный клиентским приложением. Для каждого физического устройства создается отдельный экземпляр объекта драйвера
Система контроля и управления доступом (СКУД)	совокупность совместимых между собой аппаратных и программных средств, направленных на ограничение и регистрацию доступа людей, транспорта и других объектов в (из) помещения, здания, зоны и территории

2. Описание типа

2.1. Наименование типа и русскоязычный синоним

Наименование типа англ.: AccessController
Наименование типа рус.: КонтроллерДоступа
Синоним англ.: Access controller
Синоним рус.: Контроллер доступа

2.2. Краткое описание функциональности типа

Система контроля и управления доступом (СКУД) - совокупность совместимых между собой аппаратных и программных средств, направленных на ограничение и регистрацию доступа людей, транспорта и других объектов в (из) помещения, здания, зоны и территории. Контроллеры доступа отвечают за отдельные проходы. Для контроля прохода в любое помещение нужны контроллеры доступа, то есть на каждый проход ставится контроллер доступа и 1-2 считывателя (вход-выход). Контроллер доступа хранит в себе списки временных зон, привилегий, карточек пользователей и т.д.

Отдельный экземпляр устройства данного типа может соответствовать как одному физическому контроллеру так и нескольким. Необходимость подобной неоднозначности обусловлена тем, что взаимодействие с контроллерами может производиться через промежуточные устройства, которые одновременно взаимодействуют с несколькими контроллерами. Экземпляр устройства должен по запросу уметь возвращать список контроллеров, вне зависимости от того, скольким контроллерам он соответствует.

Основная функциональность типа:
- Получить список контроллеров
- Загрузить временные профили
- Загрузить праздники
- Загрузить карты пользователей
- Установить режим
- Получить события
- Получить статус
- Получить время
- Установить время

3. Алгоритмы

3.1. Общие алгоритмы

В этой главе описаны общие для всех типов оборудования алгоритмы взаимодействия приложения с драйвером устройства.

Создание нового экземпляра оборудования

- Приложение создает объект драйвера устройства
- Следующим вызванным методом должен быть Init, инициализирующий драйвер. При этом в параметрах метода драйверу передается ссылка на интерфейс IDispatch клиентского приложения. Драйвер запоминает ссылку на интерфейс для обращения к функциям обратного вызова
- У созданного объекта вызывается метод метод GetDeviceInfo для получения массива DeviceInfo, содержащего константную информацию о драйвере (закладывается разработчиком при реализации драйвера)
- Для получения значений настроек по умолчанию, у драйвера вызывается метод GetSettings.
- Если драйвер имеет собственную форму для настройки оборудования (элемент IsSettingsDlgExist из структуры DeviceInfo), у него вызывается метод ShowSettingsDlg для её отображения. Если форма настроек отсутствует, приложение отображает свою универсальную форму, заполнив её значениями настроек по умолчанию
- Пользователь редактирует значения в форме настроек устройства и закрывает её
- Если пользователь отказался от сохранения настроек, нажав на форме кнопку «Отмена», считается, что он отказался от создания устройства. У драйвера вызывается метод Close и объект драйвера уничтожается. Процедура создания устройства прерывается.
- Если использовалась форма настройки приложения, то набор настроек передается в драйвер через метод SetSettings для проверки корректности их заполнения (валидации)
- Приложение генерирует уникальный идентификатор для нового экземпляра оборудования и создает соответствующий ему индивидуальный каталог в хранилище настроек
- У драйвера запрашивается набор его текущих (проверенных) настроек с помощью метода GetSettnigs. Набор настроек данного экземпляра оборудования сохраняется в индивидуальном каталоге хранилища настроек в файле Settings.xml (см. документ «Установка и обновление системы управления подключаемым оборудованием»)
- У данного объекта драйвера вызывается метод Close, после чего он выгружается из памяти приложения
- Клиентское приложение сохраняет идентификатор созданного устройства для последующего его использования (см. Использование экземпляра оборудования)

Использование экземпляра оборудования

В этом разделе описывается алгоритм работы с ранее созданным экземпляром оборудования.

- Приложение создает объект драйвера устройства. У созданного объекта вызывается метод Init, инициализирующий драйвер. При этом в параметрах метода драйверу передается ссылка на интерфейс IDispatch клиентского приложения (используется драйвером для обращения к функциям обратного вызова)
- Приложение вызывает метод драйвера SetSetiings, передавая в параметрах метода список значений настроек устройства. Драйвер применяет новые настройки.
- Приложение вызывает метод драйвера Open, в параметрах которого передается идентификатор устройства. Данный идентификатор драйвер при обращениях к функциям обратного вызова
- Начальным состоянием любого устройства после выполнения метода Open, является «Выключено». В этом состоянии допускается вызов следующих методов устройства: GetDeviceInfo, GetSettings, SetSettings, ShowSettingsDlg, CheckHealth, Enable, Disable, Close. Все остальные методы будут доступны после включения устройства.
- Перед началом реального использования оборудования приложение вызывает метод Enable. При выполнении метода Enable драйвер должен выполнить все необходимые операции по подготовке к работе: подгрузить необходимые ему внешние библиотеки, открыть, захватить, либо заблокировать необходимые для работы ресурсы: коммуникационные порты, файлы и т.п. Если метод Enable завершается успешно, то устройство переходит в состояние «Включено».

- У оборудования в состоянии «Включено» клиентское приложение последовательно вызывает необходимые ему публикуемые методы драйвера устройства.
- Драйвер устройства может порождать различные события посредством вызова у приложения функций обратного вызова.
- По окончании работы с драйвером устройства, приложение должно отключить его, вызвав метод Disable. При исполнении данного метода драйвер должен привести физическое устройство в исходное состояние, высвободить все захваченные им в процессе работы ресурсы (закрыть открытые им файлы, коммуникационные порты), выгрузить использовавшиеся сторонние библиотеки и т.п.
- Перед уничтожением объекта драйвера устройства, приложение вызывает метод-деструктор Close. Драйвер очищает все свои служебные структуры данных, в том числе очищает ссылку на интерфейс IDispatch клиентского приложения, переданную ему при вызове метода Init.

Удаление экземпляра оборудования

Для удаления экземпляра подключаемого оборудования из системы, достаточно удалить индивидуальный каталог экземпляра оборудования из хранилища настроек (см. описание в документе «Программа установки системы управления подключаемым оборудованием.doc») и очистить в настройках клиентских приложений, использовавших данный экземпляр, все ссылки на его идентификатор.

3.2. Специфические алгоритмы

Механизм настройки экземпляра устройства

При настройке экземпляра устройства, следует учитывать, что
Настройка экземпляра устройства должна включать в себя следующее:
- настройка связи с контроллером или устройством, выполняющим роль «посредника»
- функционал по добавлению, удалению и настройке контроллеров, в случае работы через «посредника»

Алгоритм предоставления доступа

В случае положительного исхода поиска карты пользователя в БД контроллер сравнивает текущее время, полученное от часов контроллера, с временными профилями (далее – ВП), присвоенными данному пользователю.
Порядок выполнения сравнения:
- Если включен режим блокировки, то доступ осуществляется только по привилегии блокировки.
- Если оба ВП не заданы, то ВП круглосуточный и дальше анализ не ведется, т.к. допуск уже разрешен.
- Сравнение текущего дня недели с таблицей праздников. Если текущие месяц и число есть в таблице, то считаем текущий день праздничным. То есть доступ осуществляется по привилегии праздничного дня.
- Сравнение с ВП1.
  - Проверка байта дней недели. Если бит, соответствующий текущему дню недели, равен «0», то нет доступа.
  - Сравнение начала интервала1. Если текущее меньше, то нет доступа.
  - Сравнение конца интервала1. Если текущее меньше, то доступ.
  - Сравнение начала интервала2. Если текущее меньше, то нет доступа.
  - Сравнение конца интервала1. Если текущее меньше, то доступ.
- Сравнение со ВП2.
  - Проверка наличия ВП2. Если нет ВП2, то нет доступа.
  - Далее аналогично сравнению с ВП1

4. Настройки

4.1. Общие настройки

В этом разделе описываются настройки общие для всех моделей данного типа оборудования. Данные настройки подлежат обязательной реализации в любом драйвере подключаемого оборудования.

DefaultTimeOut

Описание:

Значение таймаута выполнения метода по умолчанию в секундах. Если таймаут, преданный в параметрах какого-либо метода, равен нулю, драйвер использует в качестве таймаута значение данной настройки.

Тип	Допустимые значения	Значение по умолчанию	Представление
Number	10..65535	120	Таймаут выполнения метода по умолчанию, целое число (в сек.)

EventLogEnabled

Описание:

Данная настройка предназначена для включения режима записи диагностической информации в журнал событий, либо лог-файл. Используется в целях отладки и диагностики проблем с драйвером устройства. Если установлено значение True, драйвер вызывает функцию обратного вызова WriteLog в те моменты, когда это необходимо (определяется разработчиком). Значение по умолчанию для данной настройки – False

Тип	Допустимые значения	Значение по умолчанию	Представление
Boolean	True/False	False	Запись диагностической информации

4.2. Специфические настройки

Для типа оборудования «Контроллеры доступа» специфические настройки отсутствуют

5. Методы

5.1. Служебные методы

В данном разделе описаны методы, обеспечивающие взаимодействие клиентского приложения и драйвера на начальных и конечных этапах цикла использования оборудования. Данные методы подлежат обязательной реализации в любом драйвере.

Init

Синтаксис:

Init (ApplicationRef: COM-object, ErrorDescription: String): Boolean

Описание:

Метод-конструктор объекта драйвера. При выполнении данного метода происходит начальная инициализация драйвера. Метод вызывается единожды, сразу после создания объекта. Любые другие методы объекта драйвера могут быть вызваны только после успешного выполнения метода Init. Если метод вернул значение False, объект драйвера должен быть уничтожен, оставив систему в исходном состоянии. Если метод выполнен успешно, то с объектом драйвера можно работать как с объектом-абстрактом. Например получить список настроек по умолчанию, вызвав метод GetSettings.

Параметры:

ApplicationRef: COM-object

Ссылка на интерфейс IDispatch приложения-клиента. Используя данную ссылку, драйвер может вызывать у приложения функции обратного вызова (после выполнения метода Open).

ErrorDescription: String

Если метод возвращает False, в данном параметре содержится описание ошибки.

Возвращаемое значение:

True – метод выполнен успешно. False – в противном случае. В случае, если метод вернул результат False, дальнейшая работа с драйвером невозможна и данный объект драйвера должен быть уничтожен

InitProxy1C

Синтаксис:
Init (LibType: Integer, AddinName: String, FileName: String, ModelID: String, ErrorDescription: String): Boolean

Описание:
Дополнительный метод-конструктор объекта драйвера 1С:Совместимо. При выполнении данного метода происходит начальная инициализация драйвера. Метод вызывается единожды, сразу после создания объекта и выполнения метода Init. Любые другие методы объекта драйвера 1С:Совместимо могут быть вызваны только после успешного выполнения метода InitProxy1C. Если метод вернул значение False, объект драйвера должен быть уничтожен, оставив систему в исходном состоянии. Если метод выполнен успешно, то с объектом драйвера можно работать как с объектом-абстрактом. Например - получить список настроек по умолчанию, вызвав метод GetSettings.

Параметры:
LibType: Integer
Тип используемой библиотеки 1С:Совместимо:
- 1 – COM
- 2 – Native API

AddinName: String
ProgID 1С-компоненты для LibType 1. Или имя объекта из библиотеки для LibType 2..
FileName: String
имя файла библиотеки для LibType 2.
ModelID: String
уникальный идентификатор (GUID) модели.
ErrorDescription: String
Если метод возвращает False, в данном параметре содержится описание ошибки.

Возвращаемое значение:
True – метод выполнен успешно. False – в противном случае. В случае, если метод вернул результат False, дальнейшая работа с драйвером невозможна и данный объект драйвера должен быть уничтожен

Open

Синтаксис:
Open (DeviceID: String, ErrorDescription: String): Boolean

Описание:
Завершает инициализацию объекта драйвера и присваивает ему уникальный идентификатор экземпляра. В процессе выполнения данного метода, драйвер не должен захватывать какие-либо неразделяемые ресурсы (например, коммуникационные порты или файлы), и не должен выполнять никаких реальных взаимодействий с устройством. Драйвер проверяет наличие необходимых ему для работы ресурсов (файлов библиотек, их версий и т.п.) и возвращает Ложь, если проверки обнаружили невозможность полноценного функционирования драйвера.
После успешного выполнения метода допускается вызов следующих методов драйвера: ShowSettingsDlg, CheckHealth, Enable, Disable..Все остальные методы будут доступны после включения устройства (метод Enable).
После выполнения метода драйвер может использовать функции обратного вызова.

Параметры:
DeviceID: String (GUID)
Уникальный идентификатор экземпляра оборудования. Идентификатор необходим при использовании функций обратного вызова.
ErrorDescription: String
Если метод вернул False, данный параметр содержит описание ошибки.

Возвращаемое значение:
True – метод выполнен успешно. False – в противном случае. В случае, если метод вернул результат False, дальнейшая работа с драйвером невозможна. Необходимо вызвать метод Close и уничтожить объект драйвера.

Close

Синтаксис:

Close (): Boolean

Описание:
Метод-деструктор драйвера. Метод должен вызываться непосредственно перед уничтожением объекта драйвера. При выполнении данного метода, драйвер устройства должен освободить все захваченные ресурсы, и очистить ссылку на интерфейс IDispatch приложения, переданную в параметрах метода Init. Никакие другие методы драйвера не могут быть вызваны после вызова Close

Параметры:
Нет

Возвращаемое значение:
True

GetDeviceInfo

Синтаксис:
GetDeviceInfo (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean
Описание:
Метод предназначен для получения статической информации о модели устройства. Допустимо вызывать данный метод в любой момент, даже у неинициализированного объекта (то есть до метода Init)

Параметры:
InputParameters: Undefined
Не используется. Необходимо передавать пустое значение.
ResultData: SafeArray
При успешном исполнении метода данный параметр содержит массив SafeArray с информацией о модели устройства. В случае ошибки выполнения метода, параметр содержит массив ErrorInfo
TimeOut: Number
Таймаут ожидания результата выполнения метода. (см. TimeOut)

Возвращаемое значение:
True – метод выполнен успешно. False – в противном случае

Массив ResultData:

Индекс SafeArray	Имя	Тип	Описание
0	TypeName	String	Наименование типа оборудования
1	ModelID	String	GUID модели оборудования
2	ModelName	String	Наименование модели
3	MajorVersion	Number	Мажор-версия системы (контроль совместимости на строгое равенство)
4	MinorVersion	Number	Минор-версия системы (совместимость снизу-вверх)
5	BuildVersion	Number	Текущая версия драйвера (билд), никак не контролируется системой
6	ProcessorName	String	Программный идентификатор драйвера. Используется для создания объекта драйвера
7	ProcessorType	Number	Тип драйвера: 1 = COM объект Win32, 2 = COM объект Win64, 3 = Native API Win32, 4 = Native API Win64, 5 = Native API Linux32, 6 = Native API Linux64
8	IsSettingsDlgExist	Boolean	Наличие собственной формы настроек (см. описание метода ShowSettingsDlg)
9	Description	String	Подробное описание модели
10	DeveloperName	String	Разработчик
11	CI_email	String	Контактная информация
12	CI_www	String	Контактная информация

13	CI_Phones	String	Контактная информация
14	CI_address	String	Контактная информация

5.2. Общие методы

Перечисленные в данном разделе методы являются обязательными для реализации в любом драйвере любого типа оборудования, использующего данную технологию. Совокупность общих и специфических методов называется «публикуемыми» методами. Публикуемые методы доступны для вызова из любого места приложения, в то время как служебные методы должен использовать только менеджер оборудования и строго по описанным алгоритмам взаимодействия. Вызов любого публикуемого метода возможен только у полностью инициализированного объекта драйвера (см. описание метода Open) за исключением методов GetSettings, GetDeviceInfo и ShowSettingsDlg, которые допустимо вызывать у объекта-абстракта. Входные и выходные параметры публикуемых методов всегда помещаются в одномерные массивы SafeArray. Все публикуемые методы всегда имеют три параметра:

InputParameters: SafeArray
Одномерный массив SafeArray с входными параметрами метода, либо Неопределено (Undefined) когда входные параметры отсутствуют.
ResultData: SafeArray
Одномерный массив SafeArray с результатами выполнения метода, либо Неопределенно (Undefined), если возвращаемые параметры отсутствуют. В случае, когда метод возвращает False, в данном параметре содержится массив, содержащий код и описание ошибки (см. описание массива ErrorInfo)
TimeOut: Number
Определяет максимальное время ожидания выполнения метода в секундах. Имеются два зарезервированных предопределенных значения:
0 – время исполнения метода не задано. Драйвер должен использовать значение настройки DefaultTimeOut.
-1 – время исполнения метода не ограничено. Таймаут может быть только целым числом

Массив ErrorInfo:

Индекс в SafeArray	Имя	Тип	Описание
0	ErrorCode	Number	Регламентированный код ошибки (см. документ «Коды ошибок и их диапазоны.doc»)
1	ErrorDescription	String	Описание ошибки оборудования
2	ExtData	SafeArray / String / Number / DateTime / Boolean	Дополнительные данные или Undefined. Тип параметра определяется типом оборудования и вызываемым методом. Необязательный

GetSettings

Синтаксис:
GetSettings (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:
Получение списка значений текущих настроек устройства. Если метод вызывается до первого вызова метода SetSettings, возвращаются значения настроек устройства по умолчанию, заданные разработчиком драйвера.

Параметры:
InputParameters: Undefined
Не используется. При вызове необходимо передавать значение Undefined.
ResultData: SafeArray
Массив значений настроек Settings. В случае ошибки выполнения метода, параметр содержит массив ErrorInfo
TimeOut: Number
Таймаут ожидания результата выполнения метода. (см. TimeOut)

Возвращаемое значение:

True – метод выполнен успешно. False – в противном случае.

Массив ResultData:

Индекс SafeArray	Имя	Тип	Описание
0	Settings	SafeArray	Массив значений настроек устройства

Массив Settings

Двумерный массив значений настроек устройства. Индекс старшего измерения (строки массива) изменяется в диапазоне от 0 до N (N = количество настроек - 1). Индекс младшего измерения (колонки массива) изменяется в диапазоне от 0 до 6

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	ID	String	Идентификатор настройки. Идентификатор должен начинаться с буквы. Допустимо использование только латинских символов.
(0..N, 1)	Value	Number / String / DateTime / Boolean	Значение настройки
(0..N, 2)	Type	String	Строковое представление типа значения настройки. Допустимые варианты: «Number» «String» «DateTime» «Boolean»
(0..N, 3)	Default	Number / String / DateTime / Boolean	Значение настройки по умолчанию
(0..N, 4)	ReadOnly	Boolean	Если равно True, то пользователям запрещается изменять значение данной настройки
(0..N, 5)	Name	String	Пользовательское представление настройки
(0..N, 6)	Description	String	Текстовое описание настройки, ее возможных значений и их воздействия на поведение драйвера

SetSettings

Синтаксис:

SetSettings (InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:

Метод предназначен для установки новых значений настроек устройства. Допустимо вызывать данный метод до вызова метода Open. Метод не должен вызываться приложением, если устройство находится в состоянии «Включено» (см. метод Enable).

Параметры:

InputParameters: SafeArray

Массив значений настроек Settings.

ResultData: SafeArray либо Undefined

Возвращаемые параметры отсутствуют. В случае ошибки выполнения метода, параметр содержит массив ErrorInfo

TimeOut: Number

Таймаут ожидания результата выполнения метода. (см. TimeOut)

Возвращаемое значение:

True – метод выполнен успешно. False – в противном случае

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	Settings	SafeArray	Массив значений настроек устройства

ShowSettingsDlg

Синтаксис:

ShowSettingsDlg (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:

При выполнении данного метода, драйвер отображает в модальном режиме форму для интерактивного изменения настроек устройства. Допускается отсутствие собственной формы настроек при реализации драйвера. В этом случае, метод возвращает False и ошибку с кодом 1003 (см. ErrorInfo), а клиентское приложение отображает собственную универсальную форму для редактирования настроек устройства.

Параметры:

InputParameters: Undefined

Не используется. Необходимо передавать пустое значение.

ResultData: SafeArray

При отсутствии у драйвера формы настроек, возвращается ErrorInfo с кодом ошибки 1003. Если пользователь отказался от внесения изменений (нажал кнопку «Отмена») возвращается ErrorInfo с кодом ошибки 201. В случае, если пользователь нажал кнопку «ОК» и настройки были успешно применены, содержит значение Undefined

TimeOut: Number

Таймаут ожидания результата выполнения метода. Значение данного параметра игнорируется

Возвращаемое значение:

True – метод выполнен успешно, настройки изменены и применены новые значения настроек. False – изменения настроек не произошло.

CheckHealth

Синтаксис:

CheckHealth (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:

Метод используется для проверки работоспособности устройства. Вызов данного метода может приводить к исполнению длительных операций, поэтому в клиентском приложении не рекомендуется использовать данный метод в обычном цикле использования оборудования. Технология никак не регламентирует действия драйвера при выполнении данного метода. Состав производимых проверок определяется разработчиком драйвера.

Параметры:

InputParameters: Undefined

Не используется. Необходимо передавать пустое значение.

ResultData: Undefined либо SafeArray

Возвращаемые параметры отсутствуют. В случае ошибки выполнения метода, параметр содержит массив ErrorInfo

TimeOut: Number

Таймаут ожидания результата выполнения метода. Значение данного параметра игнорируется

Возвращаемое значение:

True – устройство исправно и работоспособно. False – в противном случае

Enable

Синтаксис:

Enable (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:

Включение устройства. Метод должен быть вызван перед началом использования оборудования, до вызова любого специфического метода. При выполнении данного метода драйвер проводит все возможные проверки на готовность устройства к работе и только в случае их успешного завершения, возвращает результат True. Если драйверу требуется для работы монопольный захват каких-либо ресурсов (например, коммуникационный порт), он должен выполнить его при выполнении данного метода.

Параметры:

InputParameters: Undefined

Не используется. Необходимо передавать пустое значение.

ErrorInfo: SafeArray либо Undefined

Возвращаемые параметры отсутствуют. В случае ошибки выполнения метода, параметр содержит массив ErrorInfo

TimeOut: Number

Таймаут ожидания результата выполнения метода. (см. TimeOut)

Возвращаемое значение:
True – устройство включено и готово к работе. False – в противном случае.

Disable

Синтаксис:
Disable (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:
Выключение устройства. Если драйвером были монопольно захвачены какие-либо ресурсы (например, коммуникационный порт), он должен освободить их при выполнении данного метода.

Параметры:
InputParameters: Undefined
Не используется. Необходимо передавать пустое значение.
ErrorInfo: SafeArray
Возвращаемые параметры отсутствуют. В случае ошибки выполнения метода, параметр содержит массив ErrorInfo
TimeOut: Number
Таймаут ожидания результата выполнения метода. (см. TimeOut)

Возвращаемое значение:
True – устройство успешно выключено. False – в противном случае

5.3. Специфические методы

Перечисленные в данном разделе методы являются специфическими методами типа оборудования. Любой из этих методов может быть вызван только у включенного устройства (см. Enable).

GetControllers

Синтаксис:
GetControllers(InputParameters: Undefined, ResultData: SafeArray, TimeOut: Numbers): Boolean

Описание:
Возвращает массив контроллеров, обслуживаемых данным экземпляром устройства.

Параметры:
InputParameters: Undefined
Не используется.
ResultData: SafeArray
Содержит массив контроллеров. В случае ошибки – стандартный массив ErrorInfo.
TimeOut: Numbers
Таймаут исполнения команды в секундах. См. TimeOut

Возвращаемое значение:
True – функция выполнена успешно. False – в противном случае.

Массив ResultData:

Индекс SafeArray	Имя	Тип	Описание
0	Controllers	SafeArray	Массив контроллеров доступа

Массив Controllers:
Содержит массив контроллеров доступа.
Если не задан один из параметров EntranceControl или ExitControl, то они по умолчанию имеют значение True.

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	ControllerID	String, 10	Идентификатор контроллера доступа
(0..N, 1)	ControllerName	String	Наименование контроллера доступа. Необязательный.
(0..N, 2)	EntranceControl	Boolean	Признак наличия считывателя на входе. Необязательный.
(0..N, 3)	ExitControl	Boolean	Признак наличия считывателя на выходе. Необязательный.

UploadTimeProfiles

Синтаксис:
UploadTimeProfiles(InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean

Описание:
Выгружает массив временных профилей в каждый контроллер из массива контроллеров. Временной профиль состоит из дней недели и 2 временных интервалов, в которые разрешен проход для тех карт, которым назначена привилегия прохода по этим профилям. Подробнее см. «Алгоритм предоставления доступа».

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа, тип загрузки и массив временных профилей.
ResultData: Undefined
Не используется. В случае ошибки – стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды в секундах. См. TimeOut

Возвращаемое значение:
True – функция выполнена успешно. False – в противном случае.

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	UploadType	Number	Тип загрузки: 0 – полная загрузка с предварительной очисткой таблиц 1 – обновление таблиц
1	Controllers	SafeArray	Массив контроллеров доступа
2	TimeProfiles	SafeArray	Массив временных профилей

Массив Controllers:
Содержит массив контроллеров доступа.
Если заданы оба временных интервала, то они объединяются.
Если в одном из интервалов один из параметров – BeginTime или EntTime, не заданы, то они имеют значения по умолчанию – начало дня и конца дня соответственно.
Если оба временных интервала не заданы, то проход разрешен в течение всего дня (в те дни, которые разрешен проход).

Индекс SafeArray	Имя	Тип	Описание
(0..N)	ControllerID	String, 10	Идентификатор контроллера доступа

Массив TimeProfiles:
Содержит массив временных профилей.
Если временной профиль с переданным ID уже был записан в устройство, то его значение перезаписывается.

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	TimeProfileID	String, 10	Идентификатор временного профиля
(0..N, 1)	Monday	Boolean	Признак разрешения прохода в понедельник
(0..N, 2)	Tuesday	Boolean	Признак разрешения прохода во вторник
(0..N, 3)	Wednesday	Boolean	Признак разрешения прохода в среду
(0..N, 4)	Thursday	Boolean	Признак разрешения прохода в четверг
(0..N, 5)	Friday	Boolean	Признак разрешения прохода в пятницу
(0..N, 6)	Saturday	Boolean	Признак разрешения прохода в субботу
(0..N, 7)	Sunday	Boolean	Признак разрешения прохода в воскресенье
(0..N, 8)	BeginTime1	Date	Начало первого интервала времени разрешения прохода. Необязательный.
(0..N, 9)	EndTime1	Date	Окончание первого интервала времени разрешения прохода. Необязательный.
(0..N, 10)	BeginTime2	Date	Начало второго интервала времени разрешения прохода. Необязательный.

(0..N, 11)	EndTime2	Date	Окончание второго интервала времени разрешения прохода. Необязательный.

UploadHolidays

Синтаксис:
UploadHolidays(InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean

Описание:
Выгружает массив праздников в каждый контроллер из массива контроллеров.
Праздники представляют собой дни текущего календарного года, в которые не действует проход по временным профилям. В дни праздников, контроллер разрешает проход только для тех карт, которым назначена привилегия прохода по праздникам. Подробнее см. «Алгоритм предоставления доступа».

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа, тип загрузки и массив праздников.
ResultData: Undefined
Не используется. В случае ошибки – стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды в секундах. См. TimeOut

Возвращаемое значение:
True – функция выполнена успешно. False – в противном случае.

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	UploadType	Number	Тип загрузки: 0 – полная загрузка с предварительной очисткой таблиц 1 – обновление таблиц
1	Controllers	SafeArray	Массив контроллеров доступа
2	Holidays	SafeArray	Массив праздников

Массив Controllers:
Содержит массив контроллеров доступа.

Индекс SafeArray	Имя	Тип	Описание
(0..N)	ControllerID	String, 10	Идентификатор контроллера доступа

Массив Holidays:
Содержит массив праздников.

Индекс SafeArray	Имя	Тип	Описание
(0..N)	DayAndMonth	Date	День и месяц праздника (год и время параметра игнорируются)

UploadCards

Синтаксис:
UploadCards(InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean

Описание:
Выгружает массив карт с назначенными им временными профилями и привилегиями в каждый контроллер из массива контроллеров.

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа, тип загрузки и массив карт.
ResultData: Undefined
Не используется. В случае ошибки – стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды в секундах. См. TimeOut

Возвращаемое значение:
True – функция выполнена успешно. False – в противном случае.

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	UploadType	Number	Тип загрузки: 0 – полная загрузка с предварительной очисткой таблиц 1 – обновление таблиц
1	Controllers	SafeArray	Массив контроллеров доступа
2	Cards	SafeArray	Массив карт

Массив Controllers:
Содержит массив контроллеров доступа.

Индекс SafeArray	Имя	Тип	Описание
(0..N)	ControllerID	String, 10	Идентификатор контроллера доступа

Массив Cards:
Содержит массив карт.
Если не задан параметр TimeProfilesPass, то это означает, что разрешен круглосуточный проход во все дни недели, кроме праздников.
Если не задан параметр HolidayPass, то это означает, что разрешен круглосуточный проход в праздничные дни.
Если не задан параметр BlockPass, то это означает, что запрещен проход в режиме блокировки.

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	CardNumber	String, 40	Номер карты
(0..N, 1)	TimeProfilesPass	SafeArray	Массив привилегий по временным профилям. Необязательный.
(0..N, 2)	HolidayPass	Boolean	Признак разрешения прохода в праздничные дни. Необязательный.
(0..N, 3)	BlockPass	Boolean	Признак разрешения прохода при режиме блокировки. Необязательный.

Массив TimeProfilesPass:
Содержит массив привилегий по временным профилям.
Если не задан один из параметров EntrancePass или ExitPass, то они по умолчанию имеют значение True.

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	TimeProfileID	String, 10	Идентификатор временного профиля
(0..N, 1)	EntrancePass	Boolean	Признак разрешения входа по профилю. Необязательный.
(0..N, 2)	ExitPass	Boolean	Признак разрешения выхода по профилю. Необязательный.

DeleteCards

Синтаксис:
DeleteCards(InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean

Описание:
Удаляет массив карт из памяти каждого контроллера из массива контроллеров.

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа, и массив карт.
ResultData: Undefined
Не используется. В случае ошибки – стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды в секундах. См. TimeOut

Возвращаемое значение:
True – функция выполнена успешно. False – в противном случае.

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	Controllers	SafeArray	Массив контроллеров доступа
1	Cards	SafeArray	Массив карт

Массив Controllers:
Содержит массив контроллеров доступа.

Индекс SafeArray	Имя	Тип	Описание
(0..N)	ControllerID	String, 10	Идентификатор контроллера доступа

Массив Cards:
Содержит массив карт.

Индекс SafeArray	Имя	Тип	Описание
(0..N)	CardNumber	String, 40	Номер карты

SetMode

Синтаксис:
SetMode(InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean

Описание:
Устанавливает режим в каждый контроллер из массива контроллеров.

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа и режим, устанавливаемый в контроллер.
ResultData: Undefined
Не используется. В случае ошибки - стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды (в секундах). См. TimeOut

Возвращаемое значение:
True – метод выполнен успешно. False – в противном случае

Массив InputParameters

Индекс SafeArray	Имя	Тип	Описание
0	Controllers	SafeArray	Массив контроллеров доступа
1	Mode	Number	Режимы: 0 – открытая дверь; 1 – закрытая дверь; 2 – дверь заблокирована; 3 – дверь разблокирована; 4 – включен режим охраны; 5 – выключен режим охраны.

Массив Controllers:
Содержит массив контроллеров доступа.

Индекс SafeArray	Имя	Тип	Описание
(0..N)	ControllerID	String, 10	Идентификатор контроллера доступа

GetEvents

Синтаксис:
GetEvents(InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:

Получает массив событий от контроллеров из буфера за указанный период. Необязательный метод

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа, начальную и конечную дату периода.
ResultData: SafeArray
В случае ошибки – стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды в секундах. См. TimeOut

Возвращаемое значение:
True – функция выполнена успешно. False – в противном случае.

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	Controllers	SafeArray	Массив контроллеров доступа
1	BeginDate	DateTime	Начало периода
2	EndDate	DateTime	Конец периода

Массив Controllers:
Содержит массив контроллеров доступа.

Индекс SafeArray	Имя	Тип	Описание
(0..N)	ControllerID	String, 10	Идентификатор контроллера доступа

Массив ResultData:

Индекс SafeArray	Имя	Тип	Описание
0	Events	SafeArray	Массив событий и контроллеров

Массив Events:

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	ControllerID	String, 10	Идентификатор контроллера доступа
(0..N, 1)	DateTime	Date	Дата и время события
(0..N, 2)	EventType	String	Тип события (совпадает с параметрами ErrorName и Event функций ErrorEvent, DataEvent): «Entrance»– Вход «Exit» – Выход «EntranceDenied» – Отказ входа «ExitDenied» – Отказ выхода «Hacked» – Попытка взлома «Broken» - Поломка «NotPowered» – Отключение питания «Discharged» – Разряд батареи
(0..N, 3)	CardNumber	String	Номер карты. Необязательный
(0..N, 4)	Description	String	Описание события. Необязательный

DeleteEvents

Синтаксис:
GetEvents(InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:
Удаляет события от контроллеров из буфера за указанный период. Необязательный метод

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа, начальную и конечную дату периода.
ResultData: SafeArray
Не используется. В случае ошибки – стандартный массив ErrorInfo.

TimeOut: Number
Таймаут исполнения команды в секундах. См. TimeOut

Возвращаемое значение:
True – функция выполнена успешно. False – в противном случае.

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	Controllers	SafeArray	Массив контроллеров доступа
1	BeginDate	DateTime	Начало периода
2	EndDate	DateTime	Конец периода

Массив Controllers:
Содержит массив контроллеров доступа.

Индекс SafeArray	Имя	Тип	Описание
(0..N)	ControllerID	String, 10	Идентификатор контроллера доступа

GetStatus

Синтаксис:
GetStatus(InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:
Получает информацию о его текущем состоянии контроллеров. Необязательный.

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа.
ResultData: SafeArray
Содержит массив статусов контроллеров. В случае ошибки – стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды в секундах. См. TimeOut

Возвращаемое значение:
True – функция выполнена успешно. False – в противном случае.

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	Controllers	SafeArray	Массив контроллеров доступа

Массив Controllers:
Содержит массив контроллеров доступа.

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	ControllerID	String, 10	Идентификатор контроллера доступа

Массив ResultData:

Индекс SafeArray	Имя	Тип	Описание
0	ControllersStatuses	SafeArray	Массив статусов контроллеров

Массив ControllersStatuses:

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	ControllerID	String, 10	Идентификатор контроллера доступа
(0..N, 1)	IsNotReady	Boolean	Признак того, что контроллер или дверь находятся не в нормальном состоянии. Нормальное состояние подразумевает, что все нижеперечисленные флаги установлены в False и нет других поломок или нестандартных режимов
(0..N, 2)	IsBlocked	Boolean	Признак того, что контроллер переведен в

			режим блокировки. Необязательный.
(0..N, 3)	IsOpened	Boolean	Признак того, что дверь открыта. Необязательный.
(0..N, 4)	IsBroken	Boolean	Признак того, что контроллер/дверь/считыватель сломаны. Необязательный.
(0..N, 5)	IsHacked	Boolean	Признак того, что контроллер/дверь/считыватель взломаны. Необязательный.
(0..N, 6)	IsNotPowered	Boolean	Признак того, что контроллер отключен от сети. Необязательный.
(0..N, 7)	IsDischarged	Boolean	Признак того, что аккумулятор разряжен. Необязательный.

GetDateTime

Синтаксис:
GetDateTime(InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean

Описание:
Получает дату и время из контроллера. Необязательный.

Параметры:
InputParameters: SafeArray
Содержит идентификатор контроллера доступа.
ResultData: SafeArray
Содержит дату и время, установленные в контроллере. В случае ошибки - стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды (в секундах). См. TimeOut

Возвращаемое значение:
True – метод выполнен успешно. False – в противном случае

Массив InputParameters:

Индекс SafeArray	Имя	Тип	Описание
0	ControllerID	String, 10	Идентификатор контроллера доступа

Массив ResultData

Индекс SafeArray	Имя	Тип	Описание
0	Date	Date	Дата и время

SetDateTime

Синтаксис:
SetDateTime(InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean

Описание:
Устанавливает дату и время в контроллеры. Необязательный.

Параметры:
InputParameters: SafeArray
Содержит массив контроллеров доступа и дату, устанавливаемую в контроллеры.
ResultData: Undefined
Не используется. В случае ошибки - стандартный массив ErrorInfo.
TimeOut: Number
Таймаут исполнения команды (в секундах). См. TimeOut

Возвращаемое значение:
True – метод выполнен успешно. False – в противном случае

Массив InputParameters

Индекс SafeArray	Имя	Тип	Описание

0	Controllers	SafeArray	Массив контроллеров доступа
1	Date	Date	Дата и время

Массив Controllers:

Содержит массив контроллеров доступа.

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	ControllerID	String, 10	Идентификатор контроллера доступа

6. Функции обратного вызова

В этом разделе описаны функции, подлежащие обязательной реализации в менеджере управления оборудованием любого клиентского приложения. Все функции предназначены для вызова их драйверами устройств.
При вызове метода Init, в его параметрах передается ссылка на интерфейс IDispatch приложения-клиента. Используя эту ссылку, драйвер может вызывать функции обратного вызова, реализованные в приложении.
Для удаленно используемого оборудования вызовы функций будут перенаправляться удаленным клиентским приложениям, через сетевой транспорт.

GetAppProperty

Синтаксис:
GetAppProperty (SourceID: String, Name: String, Value: String): Boolean

Описание:

Функция предназначена для получения свойств системы и приложения-клиента.

Параметры:
SourceID: String (GUID)

Идентификатор устройства, сгенерировавшего событие
Name: String

Наименование требуемого свойства. Допустимые значения:

- «ModelsFolder» - Путь к каталогу моделей оборудования
- «DevicesFolder» - Путь к каталогу экземпляров оборудования
- «Language» - Язык локализации приложения

Value: String

В этом параметре возвращается значение свойства. Для свойства «Language» возвращается двухсимвольный идентификатор языка («ru», «en», «ua» и т. д.)

Возвращаемое значение:

True – функция выполнена успешно, указанное свойство найдено. False – в противном случае

WriteLog

Синтаксис:
WriteLog (SourceID: String, Event: String, Text: String, Type: Number): Boolean

Описание:

Функция предназначена для передачи диагностической и отладочной информации приложению-клиенту. Приложение должно сохранять данную информацию в журнале (лог-файле). Данную функцию можно вызывать в любой момент – на усмотрение разработчика драйвера. Рекомендуется использовать её для диагностики проблем и ошибок при работе с устройством. Если настройка драйвера EventLogEnabled имеет значение False, функция вызываться не должна.

Параметры:
SourceID: String (GUID)

Идентификатор устройства, сгенерировавшего событие
Event: String

Наименование события. Произвольная строка. По наименованию событий можно фильтровать/группировать записи в журнале.
Text: String

Описание события. Произвольная строка
Type: Number

Тип события. Допустимые значения: 0 – Информация; 1 – Ошибка; 2 - Отладка

Возвращаемое значение:
True – функция выполнена успешно, информация записана приложением в журнал. False – в противном случае

ErrorEvent

Синтаксис:
ErrorEvent(SourceID: String, ErrorName: String, Description: String, ExtData: SafeArray): Boolean

Описание:
Функция предназначена для передачи приложению информации о возникновении ошибочного состояния или аварийной ситуации в устройстве
Для типа оборудования «AccessController», функция вызывается драйвером для передачи приложению ршибок работы устройства, зарегистрированных контроллером.

Параметры:
SourceID: String
Идентификатор устройства, сгенерировавшего событие
ErrorName: String
Строка с наименованием типа ошибки. Драйвер оборудования может генерировать ошибки разных типов.
Возможные значения:
«Broken» - сломался контроллер/дверь/считыватель;
«NotPowered» - контроллер не подключен к сети;
«Discharged» - батарея контроллера разряжена.
Description: String
Строка с кратким описанием возникшей ошибки или аварийной ситуации в устройстве
ExtData: String
Идентификатор контроллера.

Возвращаемое значение:
True – функция выполнена успешно, ошибка обработана. False – в противном случае

DataEvent

Синтаксис:
DataEvent (SourceID: String, Event: String, Data: String, ExtData: SafeArray): Boolean

Описание:
Функция предназначена для информирования приложения драйвером устройства о событии и передачи данных. Если событие распознано и обработано приложением, функция возвращает True, иначе возвращается False.
Для типа оборудования «AccessController», функция вызывается драйвером для передачи приложению событий (вход/выход, отказ) и нештатных ситуаций, зарегистрированных контроллером (попыток взлома устройства)

Параметры:
SourceID: String
Идентификатор устройства сгенерировавшее событие
Event: String
«Hacked» - зарегистрирована попытка взлома контроллера/двери/считывателя
«Entrance» - Нормальный вход по карте
«Exit» - Нормальный выход по карте
«EntranceDenied» - Отказано во входе
«ExitDenied» - Отказано в выходе
Data: String
Идентификатор контроллера доступа.
ExtData: SafeArray

Индекс SafeArray	Имя	Тип	Описание
0	ControllerID	String, 10	Идентификатор контроллера доступа
1	CardNumber	String, 40	Номер карты. Передается для всех событий, кроме «Hacked».

2	Description	String	Краткое описание события

Возвращаемое значение:

True – функция выполнена успешно, данные события доставлены в клиентское приложение.
False – событие не было обработано клиентским приложением