

# **Проект Тип оборудования FiscalPrinter**

# ОГЛАВЛЕНИЕ

<b>1. ОСНОВНЫЕ ПОНЯТИЯ И ТЕРМИНЫ .....</b>	<b>4</b>
<b>2. ОПИСАНИЕ СТАНДАРТА .....</b>	<b>6</b>
2.1. Наименование типа и русскоязычный синоним .....	6
2.2. Краткое описание функциональности типа .....	6
<b>3. АЛГОРИТМЫ .....</b>	<b>6</b>
3.1. Общие алгоритмы .....	6
Создание нового экземпляра оборудования .....	6
Использование экземпляра оборудования .....	7
Удаление экземпляра оборудования .....	7
3.2. Специфические алгоритмы .....	8
<b>4. НАСТРОЙКИ .....</b>	<b>8</b>
4.1. Общие настройки .....	8
DefaultTimeOut .....	8
EventLogEnabled .....	8
4.2. Специфические настройки .....	8
MaxReceiptRowLength .....	8
MaxSlipRowLength .....	9
SlipRows .....	9
HeaderText .....	9
FooterText .....	9
FullCutCode .....	9
PartialCutCode .....	9
<b>5. МЕТОДЫ .....</b>	<b>10</b>
5.1. Служебные методы .....	10
Init .....	10
Open .....	10
Close .....	11
GetDeviceInfo .....	11
5.2. Общие методы .....	12
GetSettings .....	12
SetSettings .....	13
ShowSettingsDlg .....	14
CheckHealth .....	14
Enable .....	14
Disable .....	15
5.3. Специфические Методы .....	15
OpenDay .....	15
PrintReceipt .....	16
Encash .....	18

PrintXReport .....	19
PrintExtReport .....	20
PrintZReport .....	20
PrintTemplateDocument .....	21
GetCountersInfo .....	22
GetLastDocInfo .....	23
OpenDrawer .....	24
GetFPStateInfo .....	24
GetDateTime .....	24
SetDateTime .....	25
PrintText .....	25
UploadTemplate .....	26
DownloadTemplate .....	26
<b>6. ФУНКЦИИ ОБРАТНОГО ВЫЗОВА.....</b>	<b>28</b>
GetAppProperty .....	29
ErrorEvent.....	29
WriteLog .....	29
TaskState .....	30

# 1. Основные понятия и термины

Термин	Описание
Внешнее оборудование	Любое торговое или промышленное оборудование, использующееся клиентским приложением
Тип оборудования	Регламентированное стандартом множество, объединяющее устройства с одинаковым функциональным назначением
Драйвер оборудования, Драйвер	Компьютерная программа, с помощью которой система управления внешним оборудованием получает доступ к устройству стандартным образом, с помощью интерфейса соответствующего типа оборудования, описанного в стандарте
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 внешним оборудованием и приложением-клиентом.
Клиентское приложение	Любое приложение, использующее для управления внешним оборудованием описанные в данном стандарте интерфейсы и алгоритмы взаимодействия с драйверами устройств.
Менеджер оборудования	Часть клиентского приложения, в которой сосредоточены сервисные процедуры и регламентированные данным стандартом 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 данным стандартом функции клиентского приложения, доступные для вызова через интерфейс IDIspatch драйвером оборудования. Функции обратного вызова предназначены для информирования приложения о событиях или передачи данных.
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

Термин	Описание
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
Кассовый аппарат, контрольно-кассовая машина	Электронный прибор, снабжённый устройством для печати кассовых чеков. У полнофункционального кассового аппарата помимо этого есть экран, клавиатура. Используется при расчётах за проданные товары и выполненные услуги. Самая важная часть аппарата — фискальная память, данные из которой нельзя удалить. В фискальной памяти накапливаются данные об операциях, совершённых на данной торговой точке и подлежащих налогообложению. ККМ является инструментом контроля со стороны государства за налично-денежным оборотом, полнотой и своевременностью оприходования предприятиями наличной выручки
Фискальный регистратор, ФР	Контрольно-кассовая машина, способная работать только в составе компьютерно-кассовой системы, получая данные через канал связи. Как правило, не имеет экрана и клавиатуры
Кассовый чек	Первичный учетный документ, отпечатанный контрольно-кассовой техникой на бумажном носителе, подтверждающий факт осуществления между пользователем и покупателем (клиентом) наличного денежного расчета и (или) расчета с использованием платежных карт, содержащий сведения об этих расчетах, зарегистрированных программно-аппаратными средствами контрольно-кассовой техники, обеспечивающими надлежащий учет денежных средств при проведении расчетов
Кассовая смена	Промежуток времени, в течение которого разрешена работа на ККМ в фискальном режиме без выдачи очередного фискального отчета (Z-отчета). В соответствии с фискальным законодательством РФ, кассовая смена не может превышать 24 часа. Открытие смены фиксируется либо по окончании процедуры открытия смены, либо в результате пробития первого кассового чека. Кассовые смены нумеруются по порядку.
Контрольная лента	Первичный учетный документ, выполненный контрольно-кассовой техникой на бумажном или электронном носителе, содержащий сведения о контрольно-кассовой технике и наличных денежных расчетах и (или) расчетах с использованием платежных карт
Фискальная память	Комплекс программно-аппаратных средств в составе контрольно-кассовой техники, обеспечивающих некорректируемую ежесуточную(ежесменную) регистрацию и энергонезависимое долговременное хранение итоговой информации, необходимой для полного учета наличных денежных расчетов и (или) расчетов с использованием платежных карт, осуществляемых с применением контрольно-кассовой техники, в целях правильного исчисления налогов
Электронная контрольная лента защищённая (ЭКЛЗ)	Техническое устройство, обеспечивающее защищенную (защита реализована с использованием криптографии) от не обнаруживаемой коррекции регистрацию и

Термин	Описание
	энергонезависимое долговременное хранение информации о каждом из проведенных с использованием ККТ денежных расчетов, необходимое для полного учета доходов в целях правильного исчисления налогов. ЭКЛЗ представляет собой небольшой пластиковый корпус с защитной голограммой и контактным разъемом.
Криптографический проверочный код (КПК)	Совокупность параметров: значение КПК и номер КПК. Значение КПК — контрольная сумма документа, рассчитываемая из его параметров на основании криптографических алгоритмов. Номер КПК — непрерывно нарастающий порядковый номер операции формирования значения КПК.
Фискальный режим	Режим функционирования контрольно-кассовой техники, обеспечивающий регистрацию фискальных данных в фискальной памяти
Фискальные данные	Фиксируемая на контрольной ленте и в фискальной памяти информация о наличных денежных расчетах и (или) расчетах с использованием платежных карт

## 2. Описание стандарта

### 2.1. Наименование типа и русскоязычный синоним

Наименование типа англ.:	FiscalPrinter
Наименование типа рус.:	ФискальныйРегистратор
Синоним англ.:	Fiscal printer
Синоним рус.:	Фискальный регистратор

### 2.2. Краткое описание функциональности типа

Драйвер устройства данного типа умеет выполнять все основные функции фискального регистратора:

открытие смены, печать чека (продажа, возврат, копия), инкассация, снятие X-отчета, снятие Z-отчета, печать текста, установка/считывание даты и времени, открытие денежного ящика и т.д. Помимо этого, драйвер может позволять производить печать чека или произвольного нефискального документа по шаблону. Драйвер фискального регистратора также позволяет получить информацию о текущем состоянии устройства – значения денежных счетчиков, данные о последнем документе, состоянии смены, наличии бумаги и пр.

## 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 [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 `GetSettings`. Набор настроек данного экземпляра оборудования сохраняется в индивидуальном каталоге хранилища настроек в файле `Settings.xml` (см. документ «Установка и обновление системы управления внешним 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 `SetSettings`, 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 (закрывать 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 внешнего оборудования из системы, достаточно удалить индивидуальный каталог экземпляра оборудования из хранилища настроек (см. описание в документе «Программа установки системы управления внешним оборудованием.doc») и очистить в настройках клиентских приложений, использовавших данный экземпляр, все ссылки на его идентификатор.

## 3.2. Специфические алгоритмы

Работа с ФР начинается с открытия кассовой смены. Для этого следует вызвать метод драйвера OpenDay. В течение смены на ФР можно печатать фискальные и нефискальные документы в произвольном порядке. По окончании смены следует вызвать метод PrintZReport. Это приведет к закрытию смены (гашению) на ФР и печати сменного отчета. Длительность смены ККМ может быть любой, но не более 24 часов.

Так же, в течение смены можно распечатать сменный отчет без гашения, получить данные сменных счетчиков сумм и количества документов, данные о последнем документе и состоянии ФР. Для этого служат соответственно методы PrintXReport, GetCountersInfo, GetLastDocInfo, GetFPState.

ФР может использоваться как принтер для печати произвольных нефискальных документов (например, банковских слип-чеков). Для этого предназначен метод драйвера PrintText.

У фискальных документов существует набор реквизитов, идентифицирующих документ: номер документа, номер чека, номер смены, дата документа. Любая команда, печатающая фискальные документы должна возвращать эти реквизиты в выходных параметрах.

## 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 внешнего 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 (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

#### MaxReceiptRowLength

Описание:

Максимальное количество символов, помещающихся в строке чековой ленты. Значение настройки индивидуально для каждой модели ФР. Только для чтения.

Представление	Тип	Допустимые значения	Значение по умолчанию
Длина строки чековой ленты	Number	-	-

## MaxSlipRowLength

### Описание:

Максимальное количество символов, помещающихся в строке подкладного документа. Значение настройки индивидуально для каждой модели ФР. Только для чтения.

Представление	Тип	Допустимые значения	Значение по умолчанию
Длина строки подкладного документа	String	-	-

## SlipRows

### Описание:

Максимальное количество строк, страницы подкладного документа. При превышении этого значения, будет запрошен новый лист, если такая возможность поддерживается моделью ФР. Если равно «0» - количество строк на странице не контролируется.

Представление	Тип	Допустимые значения	Значение по умолчанию
Количество строк подкладного документа	Number	-	0

## HeaderText

### Описание:

Текст заголовка (шапки) документа. Содержит статический текст, выводимый ФР в шапке фискального чека и его копии и сохраняемый в памяти устройства. Может использоваться для вывода запрограммированного текста (например "Добро пожаловать в магазин СтройТоргВсе !") при отсутствии поддержки драйвером или приложением загрузки шаблонов.

Представление	Тип	Допустимые значения	Значение по умолчанию
Текст шапки чека	String	-	-

## FooterText

### Описание:

Текст подвала документа. Содержит статический текст, выводимый ФР в подвале фискального чека и его копии и сохраняемый в памяти устройства. Может использоваться для вывода запрограммированного текста (например "Спасибо за покупку !") при отсутствии поддержки драйвером или приложением загрузки шаблонов.

Представление	Тип	Допустимые значения	Значение по умолчанию
Текст подвала чека	String	-	-

## FullCutCode

### Описание:

Код символа полного отреза бумаги в кодировке ASCII. Встретив данный символ при выполнении метода PrintText, ФР производит полную отрезку бумаги в той строке, где расположен символ. Если ФР не оснащен резаком или не поддерживает отрезку в тексте, символ игнорируется при печати. Настройка предназначена только для чтения приложением из драйвера.

Представление	Тип	Допустимые значения	Значение по умолчанию
Код символа полного отреза	Number	0 - 255	21

## PartialCutCode

### Описание:

Код символа частичного отреза бумаги в кодировке ASCII. Встретив данный символ при выполнении метода PrintText, ФР производит частичную отрезку бумаги в той строке, где расположен символ. Если ФР не оснащен резаком или не поддерживает отрезку в тексте, символ игнорируется при печати. Настройка предназначена только для чтения приложением из драйвера.

Представление	Тип	Допустимые значения	Значение по умолчанию
Код символа частичного отреза	Number	0 - 255	22

## 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-абстрактном. 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

#### 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 <a href="#">ShowSettingsDlg</a> )
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, соответствующего данному стандарту. 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 менеджер оборудования. 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, если 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.

**Массив ErrorInfo:**

Индекс в <code>SafeArray</code>	Имя	Тип	Описание
0	<code>ErrorCode</code>	Number	Код ошибки, регламентированный стандартом (см. документ «Коды ошибок и их диапазоны.doc»)
1	<code>ErrorDescription</code>	String	Описание ошибки оборудования
2	<code>ExtData</code>	<code>SafeArray</code> / <code>String</code> / <code>Number</code> / <code>DateTime</code> / <code>Boolean</code>	Дополнительные данные или <code>Undefined</code> . 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. Необходимо передавать пустое значение.

*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	<a href="#">Settings</a>	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, пользователь не может менять значение настройки
(0..N, 5)	Name	String	Пользовательское представление настройки
(0..N, 6)	Description	String	Описание настройки

**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	<a href="#">Settings</a>	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. Стандарт 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 монополюсный 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 монополюсно 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](#)).

#### OpenDay

**Синтаксис:**

*OpenDay (InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean*

**Описание:**

Открытие смены на фискальном регистраторе. Если смена уже открыта, то метод должен вернуть True. Если для данной модели ФР операция открытия смены сопровождается печатью фискального чека, то данный метод должен вернуть реквизиты чека, указанные в списке возвращаемых параметров. Обязательный метод.

**Параметры:**

*InputParameters: SafeArray*

Идентификатор оператора.

*ResultData: SafeArray*

Содержит реквизиты чека. 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	CashierID	String	Идентификатор оператора. Должен соответствовать одному из списка операторов в памяти ФР. Необязательный
1	CashierName	String	Имя оператора. Используется как альтернативный способ задания имени кассира, например при отсутствии поддержки драйвером ФР списка кассиров. Необязательный
2	PrintDestination	Number	Место печати документа. 0- печать на чековой ленте (по умолчанию); 1 - печать на подкладной бумаге; 2 - печать на контрольной ленте. Если устройство поддерживает только один вариант места печати, параметр игнорируется драйвером. Необязательный

### Массив ResultData

Драйвер должен возвращать приложению максимально возможное количество параметров документа, которые можно получить у ФР (DayNum, RecNum, DocumentNum), несмотря на то, что все эти параметры являются необязательными.

Индекс SafeArray	Имя	Тип	Описание
0	DayNum	Number	Номер смены. Необязательный
1	RecNum	Number	Номер чека. Необязательный
2	DocumentNum	Number	Номер документа. Необязательный
3	RecDate	DateTime	Дата документа

### PrintReceipt

#### Синтаксис:

*PrintReceipt (InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean*

#### Описание:

Печать фискального чека или нефискальной копии чека. Если в методе передан номер шаблона, то драйвер должен сформировать чек, максимально приблизив его внешний вид к шаблону. Обязательный метод.

#### Параметры:

*InputParameters: SafeArray*

Содержит таблицы товаров, оплат и общих настроек печати чека.

*ResultData: SafeArray*

Содержит реквизиты чека. 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	ItemTable	SafeArray	Таблица номенклатуры
1	PaymentTable	SafeArray	Таблица оплат. Номера видов оплат не должны повторяться
2	CommonParameters	SafeArray	Таблица общих параметров чека

### Массив ResultData

Драйвер должен возвращать приложению максимально возможное количество параметров документа, которые можно получить у ФР (DayNum, RecNum, DocumentNum), несмотря на то, что все эти параметры являются необязательными.

Индекс SafeArray	Имя	Тип	Описание
0	DayNum	Number	Номер смены. Необязательный
1	RecNum	Number	Номер чека. Необязательный
2	DocumentNum	Number	Номер документа. Необязательный
3	RecDate	DateTime	Дата документа

#### Массив ItemTable

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	Name	String	Наименование позиции номенклатуры
(0..N, 1)	Code	String	Код номенклатуры (штрихкод). Необязательный
(0..N, 2)	Article	String	Артикул номенклатуры. Необязательный
(0..N, 3)	DepartmentID	Number	Номер отдела
(0..N, 4)	Price	Number	Цена за одну позицию без скидки/наценки
(0..N, 5)	Quantity	Number	Количество позиций номенклатуры
(0..N, 6)	DiscountName	String	Наименование скидки/наценки. Необязательный
(0..N, 7)	DiscountSum	Number	Сумма скидки/наценки
(0..N, 8)	DiscountPercent	Number	Процент скидки/наценки
(0..N, 9)	Sum	Number	Сумма позиции номенклатуры со скидкой
(0..N, 10)	Tax1Code	Number	Номер налога из таблицы соответствий налогов. Необязательный
(0..N, 11)	Tax1Sum	Number	Сумма налога на позицию номенклатуры. Необязательный
(0..N, 12)	Tax1Value	Number	Процент налога. Необязательный
(0..N, 13)	Tax2Code	Number	Номер налога из таблицы соответствий налогов. Необязательный
(0..N, 14)	Tax2Sum	Number	Сумма налога на позицию номенклатуры. Необязательный
(0..N, 15)	Tax2Value	Number	Значение налога. Необязательный
(0..N, 16)	ItemSection	String	Наименование секции, по которой будет производиться форматирование товарной строки. Имеет смысл только при печати чека по шаблону
(0..N, 17)	Description		Дополнительный комментарий. По умолчанию не должен печататься на ФР. Рекомендуется использовать совместно с шаблонами документов для вывода специфического для позиции номенклатуры текста. Необязательный.
(0..N, 18)	DepartmentName		Наименование отдела. Необязательный

#### Массив PaymentTable

При регистрации оплаты в фискальном чеке драйвер должен руководствоваться номером оплаты, который соответствует какому-либо типу оплаты в памяти ФР. При этом номер оплаты 0 всегда соответствует оплате Наличными.

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	PaymentTypeID	Number	Номер вида оплаты, как он задан в ФР
(0..N, 1)	Sum	Number	Сумма оплаты
(0..N, 2)	Description	String	Комментарий к оплате. Необязательный
(0..N, 3)	PaymentTypeName	String	Наименование вида оплаты. Необязательный
(0..N, 4)	PaymentsSection	String	Наименование секции, по которой будет производиться форматирование строки оплаты. Имеет смысл только при печати чека по шаблону. Необязательный

## Массив CommonParameters

Индекс SafeArray	Имя	Тип	Описание
0	RecType	Number	Тип чека: 0 – Продажа; 1 – Возврат.
1	IsFiscal	Boolean	Признак фискального чека (True – фискальный чек; False – копия чека)
2	PrintDestination	Number	0 - печать на чековой ленте (по умолчанию); 1 - печать на подкладной бумаге; 2 - печать на контрольной ленте. Если устройство поддерживает только один вариант печати, то игнорируется. Необязательный
3	TotalWithoutRecDisc	Number	Сумма по чеку без учета скидок и надбавок на чек
4	Total	Number	Сумма по чеку с учетом всех скидок и надбавок
5	DiscountCard	String	Номер дисконтной карты, по которой назначена скидка. Необязательный
6	HeaderText	String	Текст шапки (дополнительный текст шапки, который печатается перед шапкой чека). Необязательный
7	FooterText	String	Текст подвала шапки (дополнительный текст подвала, который печатается после подвала чека). Необязательный
8	DayNum	Number	Номер смены (для копии чека). Необязательный
9	RecNum	Number	Номер чека (для копии чека). Необязательный
10	DocumentNum	Number	Номер документа (для копии чека). Необязательный
11	RecDate	DateTime	Дата документа (для копии чека). Необязательный
12	CashierName	String	Имя кассира (для копии чека). Необязательный
13	CashierID	String	Идентификатор (номер) кассира в памяти ФР. Необязательный
14	TemplateName	Number	Номер шаблона. 0 – для шаблона по умолчанию. Необязательный
15	HeaderSection	String	Наименование секции, по которой будет производиться форматирование шапки чека. Имеет смысл только при печати чека по шаблону. Необязательный
16	FooterSection	String	Наименование секции, по которой будет производиться форматирование подвала чека. Имеет смысл только при печати чека по шаблону. Необязательный

## Encash

### Синтаксис:

*Encash (InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean*

### Описание:

Печать фискального документа Внесения или Изъятия суммы из кассы, с изменением счетчика остатка. Перед выполнением операции изъятия драйвер должен проверить наличие необходимой суммы в кассе, если это возможно, и в случае отсутствия нужной суммы, вернуть соответствующую ошибку. Обязательный метод.

### Параметры:

*InputParameters: SafeArray*

Содержит тип и сумму инкассации.

*ResultData: SafeArray*

Содержит реквизиты чека. 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	EncashmentType	Number	Тип инкассации: 0 – изъятие; 1 – внесение
1	Sum	Number	Сумма инкассации
2	PrintDestination	Number	0 - печать на чековой ленте (по умолчанию); 1 - печать на подкладной бумаге; 2 - печать на контрольной ленте. Если устройство поддерживает только один вариант печати, то игнорируется. Необязательный

### Массив ResultData

Драйвер должен возвращать приложению максимально возможное количество параметров документа, которые можно получить у ФР (DayNum, RecNum, DocumentNum), несмотря на то, что все эти параметры являются необязательными.

Индекс SafeArray	Имя	Тип	Описание
0	DayNum	Number	Номер смены. Необязательный
1	RecNum	Number	Номер чека. Необязательный
2	DocumentNum	Number	Номер документа. Необязательный
3	RecDate	DateTime	Дата документа

### PrintXReport

#### Синтаксис:

*PrintXReport (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean*

#### Описание:

Печать X-отчета (отчет о кассовой смене, не изменяющий состояния ФР). Вид отчета определяется настройками ФР. Обязательный метод.

#### Параметры:

*InputParameters: Undefined*

Не используется. Необходимо передавать пустое значение.

*ResultData: SafeArray*

Содержит реквизиты чека. 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	PrintDestination	Number	0- печать на чековой ленте (по умолчанию); 1 - печать на подкладной бумаге; 2 - печать на контрольной ленте. Если устройство поддерживает только один вариант печати, то игнорируется. Необязательный

### Массив ResultData

Драйвер должен возвращать приложению максимально возможное количество параметров документа, которые можно получить у ФР (DayNum, RecNum, DocumentNum), несмотря на то, что все эти параметры являются необязательными.

Индекс SafeArray	Имя	Тип	Описание
0	DayNum	Number	Номер смены. Необязательный
1	RecNum	Number	Номер чека. Необязательный
2	DocumentNum	Number	Номер документа. Необязательный
3	RecDate	DateTime	Дата документа

## PrintExtReport

### Синтаксис:

*PrintExtReport (InputParameters: SafeArray, ResultData: SafeArray, TimeOut: Number): Boolean*

### Описание:

Печать расширенного отчета. Драйвер ФР может поддерживать не все виды расширенных отчетов либо не поддерживать печать данного типа отчета вовсе. Необязательный метод

### Параметры:

*InputParameters: SafeArray*

Содержит параметры отчета.

*ResultData: SafeArray*

Содержит реквизиты чека. 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	ReportType	Number	Номер типа отчета без гашения. Допустимые значения: 1 – Отчет по датам 2 – Отчет по сменам 3 – Отчет по секциям 4 – Отчет по кассирам 5 – Отчет по товарам 6 – Отчет по часам 7 – Отчет по количеству 8 – Отчет по налогам 9 – Контрольная лента (ЭКЛЗ) 10 – Отчет по сменам по отделам (ЭКЛЗ) 11 – Отчет по датам по отделам (ЭКЛЗ) 12 – Отчет по датам по итогам смен (ЭКЛЗ) 13 – Отчет по сменам по итогам смен (ЭКЛЗ)
1	BeginDate	DateTime	Начальная дата периода выборки
2	EndDate	DateTime	Конечная дата периода выборки
3	BeginDay	Number	Начальный номер смены периода выборки
4	EndDay	Number	Конечный номер смены периода выборки
5	EKLZKPKNumber	Number	Номер КПК ЭКЛЗ
6	PrintDestination	Number	0- печать на чековой ленте (по умолчанию); 1 - печать на подкладной бумаге; 2 - печать на контрольной ленте. Если устройство поддерживает только один вариант печати, то игнорируется. Необязательный

### Массив ResultData

Драйвер должен возвращать приложению максимально возможное количество параметров документа, которые можно получить у ФР (DayNum, RecNum, DocumentNum), несмотря на то, что все эти параметры являются необязательными.

Индекс SafeArray	Имя	Тип	Описание
0	DayNum	Number	Номер смены. Необязательный
1	RecNum	Number	Номер чека. Необязательный
2	DocumentNum	Number	Номер документа. Необязательный
3	RecDate	DateTime	Дата документа

## PrintZReport

### Синтаксис:

*PrintZReport (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean*

**Описание:**

Печать Z-отчета и закрытие смены на ФР. Если смена не была открыта до вызова метода, то драйвер должен вернуть True, не производя никаких действий. Обязательный метод.

**Параметры:**

*InputParameters: Undefined*

Не используется. Необходимо передавать пустое значение.

*ResultData: SafeArray*

Содержит реквизиты чека. 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	PrintDestination	Number	0- печать на чековой ленте (по умолчанию); 1 - печать на подкладной бумаге; 2 - печать на контрольной ленте. Если устройство поддерживает только один вариант печати, то игнорируется. Необязательный

**Массив ResultData**

Драйвер должен возвращать приложению максимально возможное количество параметров документа, которые можно получить у ФР (DayNum, RecNum, DocumentNum), несмотря на то, что все эти параметры являются необязательными.

Индекс SafeArray	Имя	Тип	Описание
0	DayNum	Number	Номер смены. Необязательный
1	RecNum	Number	Номер чека. Необязательный
2	DocumentNum	Number	Номер документа. Необязательный
3	RecDate	DateTime	Дата документа

**PrintTemplateDocument****Синтаксис:**

*PrintTemplateDocument (InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean*

**Описание:**

Печать нефискального документа по заранее загруженному в драйвер шаблону. При отсутствии запрошенного шаблона должен вернуть стандартную ошибку. Документ для печати представляется в виде списка именованных секций, которые последовательно верстаются драйвером при печати согласно шаблону. Каждая секция состоит из произвольного количества именованных параметров. Более подробное описание работы с шаблонами можно посмотреть в проекте Редактор шаблонов. Необязательный метод

**Параметры:**

*InputParameters: SafeArray*

Содержит номер шаблона и параметры шаблона.

*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	TemplateParameters	SafeArray	Печатаемый документ в виде таблицы параметров

1	TemplateName	Number	Номер шаблона
2	PrintDestination	Number	0 - печать на чековой ленте (по умолчанию); 1 - печать на подкладной бумаге; 2 - печать на контрольной ленте. Если устройство поддерживает только один вариант печати, то игнорируется. Необязательный

### Массив TemplateParameters

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	Section	String	Наименование секции, по которой будет производиться форматирование параметров
(0..N, 1)	SectionParameters	SafeArray	Таблица параметров секции

### Массив SectionParameters

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	Name	String	Наименование параметра
(0..N, 1)	Value	Number/ String/ DateTime/ Boolean	Значение параметра

### GetCountersInfo

#### Синтаксис:

*GetCountersInfo (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean*

#### Описание:

Получение значений различных счетчиков из памяти фискального регистратора. Драйвер заполняет в таблице значения только тех полей, которые возможно получить у данной модели ФР

#### Параметры:

*InputParameters: Undefined*

Не используется. Необходимо передавать пустое значение.

*ResultData: SafeArray*

Содержит массив счетчиков и таблицу оборотов в разрезе оплат.

*TimeOut: Number*

Таймаут исполнения команды (в секундах). См. [TimeOut](#)

#### Возвращаемое значение:

True – метод выполнен успешно. False – в противном случае

### Массив ResultData

Индекс SafeArray	Имя	Тип	Описание
0	CountersTable	SafeArray	Массив значений счетчиков
1	PaymentTable	SafeArray	Таблица оборотов оплат

### Массив CountersTable

Если значение счетчика определить невозможно, соответствующий параметр должен содержать значение -1

Индекс SafeArray	Имя	Тип	Описание
0	SalesTotal	Number	Разница продаж и возвратов за смену
1	GrossTotal	Number	Сумма продаж и возвратов за смену (Гросс смены)
2	FullGrossTotal	Number	Сумма продаж и возвратов с момента регистрации ФР (Сумма всех Гроссов)
3	SalesCount	Number	Общее количество чеков продажи за смену
4	RefundCount	Number	Общее количество чеков возврата за смену

5	SalesSum	Number	Общая сумма продаж за смену
6	RefundSum	Number	Общая сумма возвратов за смену
7	SalesDiscountCount	Number	Количество чеков продажи со скидками за смену
8	RefundDiscountCount	Number	Количество чеков возврата со скидками за смену
9	SalesSurchargeCount	Number	Количество чеков продажи с надбавками за смену
10	RefundSurchargeCount	Number	Количество чеков возврата с надбавками за смену
11	SalesDiscountSum	Number	Общая сумма скидков по чекам продажи за смену
12	RefundDiscountSum	Number	Общая сумма скидков по чекам возврата за смену
13	SalesSurchargeSum	Number	Общая сумма надбавок по чекам продажи за смену
14	RefundSurchargeSum	Number	Общая сумма надбавок по чекам возврата за смену
15	CashOutCount	Number	Количество изъятий за смену
16	CashInCount	Number	Количество внесений за смену
17	CashOutSum	Number	Сумма изъятий за смену
18	CashInSum	Number	Сумма внесений за смену
19	CashRemainder	Number	Остаток наличности в кассе

#### Массив PaymentTable

Индекс SafeArray	Имя	Тип	Описание
(0..N, 0)	PaymentTypeID	Number	Номер вида оплаты
(0..N, 1)	PaymentTypeName	Number	Наименование вида оплаты
(0..N, 2)	Count	Number	Количество оплат по данному виду оплаты за смену
(0..N, 3)	SalesSum	Number	Сумма продаж по данному виду оплаты за смену
(0..N, 4)	RefundSum	Number	Сумма возвратов по данному виду оплаты за смену

#### GetLastDocInfo

##### Синтаксис:

*GetLastDocInfo (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean*

##### Описание:

Получение информации о последнем фискальном документе. Необязательный метод

##### Параметры:

*InputParameters: Undefined*

Не используется. Необходимо передавать пустое значение.

*ResultData: SafeArray*

Реквизиты последнего чека. 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 ResultData

Драйвер должен возвращать приложению максимально возможное количество параметров документа, которые можно получить у ФР, несмотря на то, что все эти параметры являются необязательными.

Индекс SafeArray	Имя	Тип	Описание
0	DayNum	Number	Номер смены. Необязательный
1	RecNum	Number	Номер чека. Необязательный
2	DocumentNum	Number	Номер документа. Необязательный
3	RecDate	DateTime	Дата документа. Необязательный
4	RecType	Number	Тип последнего напечатанного фискального документа: 0 – Чек продажи 1 – Чек возврата 2 – Изъятие денег

			3 – Внесение денег 4 – X-отчет 5 – Z-отчет Необязательный
5	RecSum	Number	Сумма документа. Необязательный

## OpenDrawer

### Синтаксис:

*OpenDrawer (InputParameters: Undefined, ResultData: Undefined, TimeOut: Number): Boolean*

### Описание:

Посылает в ФР команду открытия денежного ящика. Необязательный метод

### Параметры:

*InputParameters: Undefined*

Не используется. Необходимо передавать пустое значение.

*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

## GetFPStateInfo

### Синтаксис:

*GetFPStateInfo (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean*

### Описание:

Получение информации о текущем состоянии ФР. Необязательный метод.

### Параметры:

*InputParameters: Undefined*

Не используется. Необходимо передавать пустое значение.

*ResultData: SafeArray*

Содержит массив параметров ФР. 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 ResultData

Индекс SafeArray	Имя	Тип	Описание
0	DrawerState	Number	Состояние денежного ящика. 0 - Закрыт; 1 – Открыт
1	KeyState	Number	Положение ключа. В качестве значение должно вернуться число, идентифицирующее положение ключа
2	RecState	Number	Состояние ЧЛ. 0 – ОК; 1 – Заканчивается; 2 – Отсутствует
3	JrnState	Number	Состояние КЛ. 0 – ОК; 1 – Заканчивается; 2 – Отсутствует
4	SlpEmpty	Number	Состояние ПД. 0 – Вставлен; 1- Отсутствует
5	DayState	Number	Состояние смены. 0 -Смена открыта; 1 - смена закрыта; 2 - истекло 24 часа
6	FMState	Number	Состояние фискальной памяти. 0 - ОК; 1 - Ошибка
7	EKLZState	Number	Состояние ЭКЛЗ. 0 -ОК; 1 - Ошибка ЭКЛЗ; 2 - Отсутствует
7	CoverOpen	Number	Состояние крышки. 0 - Закрыта; 1 - Открыта
8	FPState	Number	Состояние ФР. 0 – ОК; 1 – Занят; 2 – Открытый документ; 3 – Ошибка

## GetDateTime

### Синтаксис:

*GetDateTime (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean*

**Описание:**

Получение текущей даты и времени внутренних часов ФР. Необязательный метод

**Параметры:**

*InputParameters: Undefined*

Не используется. Необходимо передавать пустое значение.

*ResultData: SafeArray*

Содержит дату и время ФР. 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 ResultData**

Индекс SafeArray	Имя	Тип	Описание
0	FPDate	DateTime	Дата и время, считанные из ФР

**SetDateTime****Синтаксис:**

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

**Описание:**

Изменение текущей даты и времени внутренних часов ФР. Необязательный метод

**Параметры:**

*InputParameters: SafeArray*

Содержит дату и время для установки в ФР.

*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	FPDate	DateTime	Дата и время, записываемые в ФР

**PrintText****Синтаксис:**

*PrintText (InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean*

**Описание:**

Печать произвольного текста на ФР. Обязательный метод.

**Параметры:**

*InputParameters: SafeArray*

Содержит текст, для печати на ФР.

*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	Text	String	Текст. Может включать в себя символы перевода каретки (CR/LF) и отреза бумаги (см. <a href="#">FullCutCode</a> , <a href="#">PartialCutCode</a> ).

1	PrintDestination	Number	0- печать на чековой ленте (по умолчанию); 1 - печать на подкладной бумаге; 2 - печать на контрольной ленте. Если устройство поддерживает только один вариант печати, то игнорируется. Необязательный
---	------------------	--------	---

## UploadTemplate

### Синтаксис:

*UploadTemplate (InputParameters: SafeArray, ResultData: Undefined, TimeOut: Number): Boolean*

### Описание:

Загружает в драйвер устройства переданный шаблон. Необязательный метод. Если шаблон с таким номером уже существует, то он перезаписывается. Возвращает «Истину» если шаблон корректно загружен и «Ложь» с описанием ошибки, если шаблон не может быть загружен по какой-либо причине. Используется при печати чека или произвольного текста по шаблону. В случае печати чека, имена секций и параметров шаблона должны совпадать с соответствующими именами параметров таблиц метода печати чека. В случае печати произвольного текста, состав имен секций и параметров определяется приложением. Более подробное описание работы с шаблонами можно посмотреть в проекте Редактор шаблонов.

### Параметры:

*InputParameters: SafeArray*

Содержит параметры шаблона.

*ResultData: Undefined*

Не используется. В случае ошибки - стандартный массив [ErrorInfo](#)

*TimeOut: Number*

Таймаут исполнения команды (в секундах). См. [TimeOut](#)

### Возвращаемое значение:

Истина – метод выполнен успешно. Ложь – в противном случае

### Массив InputParameters

Индекс SafeArray	Имя	Тип	Описание
0	TemplateName	Number	Номер шаблона
1	TemplateDescription	String	Описание шаблона
2	Sections	SafeArray	Массив секций шаблона

## DownloadTemplate

### Синтаксис:

*DownloadTemplate (InputParameters: Undefined, ResultData: SafeArray, TimeOut: Number): Boolean*

### Описание:

Выгружает из драйвера устройства шаблон. Необязательный метод. Возвращает «Истину» если шаблон корректно загружен и «Ложь» с описанием ошибки, если шаблон с данным номером не может быть получен по какой-либо причине. Может быть использован для получения от драйвера ФР шаблона печати чека по умолчанию (TemplateName=0). Если метод доступен, но шаблон с запрашиваемым номером не найден, метод должен вернуть стандартную ошибку. Более подробное описание работы с шаблонами можно посмотреть в проекте Редактор шаблонов.

### Параметры:

*InputParameters: Undefined*

Не используется.

*ResultData: SafeArray*

Содержит параметры шаблона. В случае ошибки - стандартный массив [ErrorInfo](#)

*TimeOut: Number*

Таймаут исполнения команды (в секундах). См. [TimeOut](#)

### Возвращаемое значение:

Истина – метод выполнен успешно. Ложь – в противном случае

### Массив ResultData

Индекс SafeArray	Имя	Тип	Описание
------------------	-----	-----	----------

0	TemplateName	Number	Номер шаблона
1	TemplateDescription	String	Описание шаблона
2	Sections	SafeArray	Массив секций шаблона

Параметры у 2-х вышеописанных методов имеют одинаковую структуру. При загрузке в устройство, они передаются в массиве входных параметров, а при выгрузке – возвращаются в массиве выходных параметров. Описание параметров приведено ниже:

#### Массив Sections

Все атрибуты секции кроме имени (Name) могут быть опущены. При этом Булевы атрибуты приравниваются к «Ложь»

Индекс	Имя	Тип	Описание
(0..N, 0)	Name	String	Имя секции
(0..N, 1)	Fixed	Boolean	Признак того, что секция не может редактироваться и определяется устройством
(0..N, 2)	Predefined	Boolean	Признак того, что секция является предопределенной и не может быть удалена.
(0..N, 3)	CompositeStrings	SafeArray	Массив составных строк

#### Массив CompositeStrings

Все атрибуты могут быть опущены. При этом значения высоты строки считаются равными единице, Булевы атрибуты приравниваются к «Ложь»

Индекс	Имя	Тип	Описание
(0..N, 0)	Fixed	Boolean	Признак того, что строка не может редактироваться и определяется устройством
(0..N, 1)	MinRows	Number	Минимальное количество строк, занимаемых составной строкой
(0..N, 2)	MaxRows	Number	Максимальное количество строк, занимаемых составной строкой
(0..N, 3)	CompositeFields	SafeArray	Массив полей

#### Массив CompositeFields

Все атрибуты могут быть опущены. При этом Строки считаются пустыми, Перечислимые и числовые значения считаются равными нулю, за исключением длины, равной единице, Булевы атрибуты приравниваются к «Ложь»

Индекс	Имя	Тип	Описание
(0..N, 0)	Fixed	Boolean	Признак того, что поле не может редактироваться и определяется устройством
(0..N, 1)	Resizable	Boolean	Признак того, что ширина составного поля имеет «плавающий» размер в зависимости от ширины каретки
(0..N, 2)	Length	Number	Длина составного поля в знаках
(0..N, 3)	HorizontalAlign	Number	Выравнивание текста внутри составного поля по горизонтали 0 – выровнять по верхнему краю 1 – выровнять по нижнему краю 2 – выровнять по центру
(0..N, 4)	VerticalAlign	Number	Выравнивание текста внутри составного поля по вертикали 0 – выровнять по левому краю 1 – выровнять по правому краю

			2 – выровнять по центру
(0..N, 5)	TextPlacement	Number	Способ вывода длинной строки: 0 – Переносить 1 – Обрезать 2 – Забивать
(0..N, 6)	Size	Number	Реквизит задает отклонения от нормального размера шрифта. Нормальным считается размер «0». «-3» самый маленький, «3» самый большой.
(0..N, 7)	Bold	Boolean	Признак того, что поле будет выводиться жирным шрифтом
(0..N, 8)	Italic	Boolean	Признак того, что поле будет выводиться курсивом
(0..N, 9)	Underline	Boolean	Признак того, что поле будет выводиться подчеркнутым шрифтом
(0..N, 10)	Inversion	Boolean	Признак того, что цвета фона и шрифта поля будут инвертированы (например, белым по черному)

### Массив Fields

Все атрибуты поля кроме имени (Name) могут быть опущены. При этом Строки считаются пустыми, Перечислимые и числовые значения считаются равными нулю. Булевы атрибуты приравниваются к «Ложь»

Индекс	Имя	Тип	Описание
(0..N, 0)	Name	String	Имя поля
(0..N, 1)	Fixed	Boolean	Признак того, что поле не будет редактироваться
(0..N, 2)	Predefined	Boolean	Признак того, что поле предопределенным и не может быть удалено.
(0..N, 3)	Type	Number	Тип поля: 0 – Строка; 1 – Число; 2 – Дата;
(0..N, 4)	Format	String	Форматная строка для представления значения поля
(0..N, 5)	IsParameter	Boolean	Признак того что значение поля подставляется из переданных параметров (в противном случае берется из шаблона)
(0..N, 6)	Prefix	String	Строковая константа, подставляемая перед Полям
(0..N, 7)	Postfix	String	Строковая константа, подставляемая после Поля
(0..N, 8)	Value	Number/ String/ DateTime/ Boolean	Если Поле является параметром, содержит тестовое значение, иначе – константное значение

## 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.

Полный список функций содержится в документе «Шаблон типа оборудования».

## 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

## 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

### Параметры:

*SourceID: String (GUID)*

Идентификатор устройства сгенерировавшего событие

*ErrorName: String*

Строка с наименованием типа ошибки. Драйвер оборудования может генерировать ошибки разных типов.

*Description: String*

Строка с кратким описанием возникшей ошибки или аварийной ситуации в устройстве

*ExtData: SafeArray*

Массив, содержащий дополнительные данные об ошибке. Необязательный.

### Возвращаемое значение:

True – функция выполнена успешно, ошибка обработана. 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

**TaskState**

**Синтаксис:**

*TaskState (SourceID: String, Percent: Number, Description: String): Boolean*

**Описание:**

Данная функция предназначена для использования только при выполнении драйвером неопределенно-длительных операций (печать большого количества этикеток, авторизация безналичного платежа и т. п.). Функция служит для уведомления приложения о ходе выполнения текущего метода и запроса на продление таймаута его исполнения. Если менеджер вернул True, отсчет таймаута выполнения метода необходимо начать заново. Если менеджер вернул False, драйвер должен прервать выполнение текущего метода. По умолчанию, менеджер оборудования возвращает True, позволяя драйверу продолжать выполнение метода. Рекомендуется вызывать эту функцию с интервалом от 1/10 до половины от значения таймаута исполнения метода. Это позволит приложению отображать на экране информацию о ходе выполнения команды (прогресс-бар, текст в строке состояния и др.). Не рекомендуется вызывать данную функцию чаще одного раза в секунду, во избежание создания лишней нагрузки на систему.

**Параметры:**

*SourceID: String*

Идентификатор устройства, сгенерировавшего событие

*Percent: Number*

Значение от 0 до 100 – приблизительная степень выполнения текущего метода в процентах.

*Description: String*

Текстовое описание текущего статуса выполнения метода. Например: «Обработано 30% полученных данных».

**Возвращаемое значение:**

True – продолжение выполнения текущего метода разрешено. False – драйверу следует прервать выполнение метода