



ООО "АСТРОН ЛТД"

**Пакет прикладных программ
PowerPOS Версия 5.0**

**Создание драйверов
фискальных принтеров**

Минск – 2015

ООО "Астрон ЛТД"
220113, г. Минск, ул. Мележа 5, корп. 2 оф. 1201
тел. +375 (17) 392-56-00, 01, 02, 03, 04, 05
факс +375 (17) 392-56-06
<http://www.astron.by>

© Copyright Astron Ltd., 2014. All right reserved.

Данная публикация или ее часть не могут быть воспроизведены в любой форме без предварительного письменного разрешения фирмы Астрон ЛТД.

Содержание:

Описание драйвера фискального принтера.....	4
Интерфейс IFiscalPrinter.....	4
Свойства	5
Методы.....	7
Класс PrinterFeatures	10
Свойства	10
Перечисление(Flags) PrinterOptions	10
Значения	10
Перечисление ReceiptType.....	11
Значения	11
Перечисление ReportType	11
Значения	11
Перечисление TaxGrp.....	11
Значения	11
Структура PrinterTaxGrp	12
Конструкторы.....	12
Свойства	12
Перечисление TaxType.....	12
Класс PrinterPayment	12
Свойства	12
Методы.....	12
Перечисление PaymentType	12
Значения	13
Интерфейс IDevice.....	13
Свойства	13
Методы.....	14
Класс Money	14
Конструкторы.....	14
Свойства	14
Класс Quantity	14
Свойства	14
Класс Port.....	15
Свойства	15
Перечисление PortType	15
Значения	15
Класс Parameter	15
Конструкторы.....	15
Свойства	16
Особенности реализации	17

Описание драйвера фискального принтера

Драйвер фискального принтера должен представлять собой сборку для Microsoft .Net 2.0. Класс драйвера должен реализовать интерфейс SoftMarket.Devices.Printers. IFiscalPrinter, находящийся в сборке DriverGlobals.dll.

Сборку, реализующую драйвер, необходимо положить в каталог приложения (по умолчанию C:\Program Files\Astron\PowerPOS\v5).

В проект драйвера нужно включить сборки DriverGlobals.dll и Globals.dll.

Интерфейс IFiscalPrinter

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers. IFiscalPrinter.

Описание: Интерфейс драйвера фискального принтера.

```
public interface IFiscalPrinter : IDevice
{
    void OpenWorkDay();
    void PrintReport(ReportType reportType);
    void PrintReport(ReportType reportType, int beginNum, int endNum);
    void PrintReport(ReportType reportType, DateTime beginDate, DateTime
endDate);
    void PrintExtReport(int reportId, object parameters);

    void BeginReceipt(ReceiptType type, int cashNum, int systemRecNum, string
comment, string recNum);
    void BeginTextDoc();
    void EndReceipt(Money receiptSum, PrinterPayment payment);
    void EndReceipt();
    void EndReceipt(bool printToControlType);
    void CancelReceipt();

    void PrintTemplate(ITemplatePrintInfo template);
    void PrintReceiptItem(int artCode, string artName, Money unitPrice, Quantity
count, string unitName, Money cost, string offerName, Money discount, Money
recDiscount, TaxGrp taxGrp, int artNum, bool cancelled);
    void PrintTextLine(string line);
    void PrintRecDiscount(Money sum, string discountName);
    void PrintClientInfo(string clientName);
    void PrintBottomComment(string[] comments);
    void PrintBarcode(string barcode);

    string NextRecNum { get; }
    string NextRefRecNum { get; }
    string NextZRepNum { get; }
    string LastRefRecNum { get; }
    string LastRecNum { get; }
    string LastZRepNum { get; }

    bool SessionOpened { get; }
    bool FiscalMode { get; }
    bool DemoMode { get; }
    bool IsSknoConnectServer { get; }
    string CashierName { get; set; }
    Money MinMoney { get; set; }
    byte DecimalPlaceMoney { get; set; }
    List<PrinterCurrency> Currencies { get; set; }
    DateTime PrinterTime { get; set; }
    string SerialNum { get; }
    Hashtable Taxes { get; set; }
}
```

```

PrinterFeatures Features { get;}

void Reset(bool reconnect);
void Init(string[] header, string[] footer, string[] firstHeader, string[]
endFooter, string recNumberForNoFiscalReceipt, string
recNumberForNoFiscalRefundReceipt, PrinterOptions options);
void Open(Port port, string deviceVersion);
void SetLogo(Bitmap bmp);
void SetPaymentTypes(Dictionary<PaymentType, string> paymentTypes);

void PrintRecCopy(int? number);
string ReadRecCopy(int? number);
void PrintNullRec();
void OpenDrawer();
void Feed();

void CashIn(Money sum, decimal currencySum = 0, int currencyId = -1);
void CashOut(Money sum, decimal currencySum = 0, int currencyId = -1);
decimal GetCashSum(int currencyId);
Money CashSum { get;}

Money SalesSum { get;}
Money RefundSum { get;}

void ShowPaymentInfo(Money recSum, string recSumHint, Money changeSum, string
changeSumHint);
void ShowItemInfo(string itemName, Quantity itemCount, Money itemCost, Money
recSum);
void ClearDisplay();
void ShowDisplayLine(string line, DisplayLineType displayLineType);

/// <summary>
/// Суммы возвратов по налоговым ставкам в формате "free,a,b,c,d,e"
/// </summary>
string RefundSumByTax
{
    get;
}
/// <summary>
/// Суммы продаж по налоговым ставкам в формате "free,a,b,c,d,e"
/// </summary>
string SalesSumByTax
{
    get;
}
}

```

Свойства

Свойство	Описание
<code>string</code> NextRecNum	Фискальный номер следующего чека
<code>string</code> NextRefRecNum	Фискальный номер следующего чека возврата
<code>string</code> NextZRepNum	Номер следующего номера Z-отчета
<code>string</code> LastRefRecNum	Фискальный номер последнего чека возврата
<code>string</code> LastRecNum	Фискальный номер последнего чека
<code>string</code> LastZRepNum	Номер последнего номера Z-отчета
<code>bool</code> SessionOpened	Признак открытой смены на принтере
<code>bool</code> FiscalMode	Принтер фискализирован
<code>bool</code> DemoMode	Принтер находится в деморежиме (нет лицензии)

<code>bool IsSknoConnectServer</code>	СКНО имеет связь с сервером
<code>string CashierName{get;set;}</code>	Имя кассира
<code>Money MinMoney</code>	Минимальная денежная единица
<code>byte DecimalPlaceMoney</code>	Знаков после запятой для денежной единицы
<code>List<PrinterCurrency> Currencies</code>	Список используемых валют. Валюты которые принтер не может поддерживать, возвращать не будет.
<code>DateTime PrinterTime{get;set;}</code>	Время и дата принтера
<code>string SerialNum</code>	Серийный номер
<code>Hashtable Taxes{get;set;}</code>	Налоговые ставки запрограммированные в принтере
<code>PrinterFeatures Features</code>	Возможности, поддерживаемые принтером
<code>Money CashSum</code>	Сумма в денежном ящике
<code>Money SalesSum</code>	Текущий оборот реализации
<code>Money RefundSum</code>	Текущий оборот возврата
<code>string RefundSumByTax</code>	Суммы возвратов по налоговым ставкам в формате "free, a, b, c, d, e"
<code>string SalesSumByTax</code>	Суммы продаж по налоговым ставкам в формате "free, a, b, c, d, e"

Методы

Метод	Описание
<code>void OpenWorkDay()</code>	Открытие рабочего дня.
<code>void PrintReport(ReportType reportType)</code>	Печать отчет. reportType - Тип отчета.
<code>void PrintReport(ReportType reportType, int beginNum, int endNum)</code>	Печать отчетов по номерам. reportType - Тип отчета beginNum - Начальный номер endNum - Конечный номер
<code>void PrintReport(ReportType reportType, DateTime beginDate, DateTime endDate)</code>	Печать отчетов по датам. reportType - Тип отчета beginDate - Начальная дата endDate - Конечная дата.
<code>void PrintExtReport(int reportId, object parameters)</code>	Печать дополнительных отчетов. reportId - Идентификатор отчета parameters - Параметры отчета.
<code>void BeginReceipt(ReceiptType type, int cashNum, int systemRecNum, string comment, string recNum)</code>	Начало печати документа. type - Тип документа cashNum - Номер кассы, на которой печатается документ systemRecNum - Системный номер чека (должен печататься на чеке принтерами, которые самостоятельно не нумеруют чеки) comment - Комментарий recNum - Номер чека, по которому производится возврат (для чеков возврата).
<code>void BeginTextDoc();</code>	Начало печати текстового нефискального документа
<code>void EndReceipt(Money receiptSum, PrinterPayment payment)</code>	Завершение печати документа (в основном для фискальных документов) receiptSum - Итоговая сумма чека payment - Оплаты чека.
<code>void EndReceipt();</code>	Завершение печати нефискального документа.
<code>void EndReceipt(bool printToControlType)</code>	Завершение печати нефискального документа. printToControlType - Печатать на контрольной ленте
<code>void CancelReceipt();</code>	Отмена документа.
<code>void PrintTemplate(ITemplatePrintInfo template)</code>	Печатает шаблон. template - Шаблон.
<code>void PrintReceiptItem(int artCode, string artName, Money unitPrice, Quantity count, string unitName, Money cost, string offerName, Money discount, Money recDiscount, TaxGrp taxGrp, int artNum, bool cancelled)</code>	Печать позиции чека. artCode - Внутренний код товара artName - Наименование товара unitPrice - Цена за единицу товара count - Количество товара

	<p>unitName - Наименование единицы измерения cost - Стоимость товарной позиции без учета скидок offerName - Наименование скидки на товарную позицию discount - Сумма скидки на товарную позицию без учета скидки на весь чек recDiscount - Часть скидки на чек, попадающая на данную товарную позицию taxGrp - Налоговая ставка artNum - Номер артикулярного регистра (предназначен для поддержки принтеров чеков, ведущих артикулярный учет) cancelled - Признак отмененной товарной позиции</p>
<code>void PrintTextLine(string line);</code>	<p>Печать текстовой строки. line - Текстовая строка.</p>
<code>void PrintRecDiscount(Money sum, string discountName)</code>	<p>Печать строки с информацией о скидке sum - Сумма скидки. discountName - Наименование скидки.</p>
<code>void PrintClientInfo(string clientName)</code>	<p>Печать строки с информацией о покупателе clientName - Имя покупателя.</p>
<code>void PrintBottomComment(string[] comments)</code>	<p>Печатает комментарии в конце чека. comments - Строки с комментариями.</p>
<code>void PrintBarcode(string barcode)</code>	<p>Печатает штрихкод Code128 в конце чека. barcode - Штрихкод Code128</p>
<code>void Reset(bool reconnect)</code>	<p>Перевод принтера чеков в начальное состояние. Если в принтере чеков есть какие-либо открытые документы, они отменяются. reconnect - Переустановка соединения</p>
<code>void Init(string[] header, string[] footer, string[] firstHeader, string[] endFooter, string recNumberForNoFiscalReceipt, string recNumberForNoFiscalRefundReceipt, PrinterOptions options)</code>	<p>Инициализация принтера header - Верхние строки чека footer - Нижние строки чека firstHeader - Шапка чека. endFooter - Подвал чека recNumberForNoFiscalReceipt - "Чек №" для нефискальных чеков recNumberForNoFiscalRefundReceipt - "Чек №" для нефискальных чеков возврата options - Настройки</p>
<code>void Open(Port port, string deviceVersion)</code>	<p>Открыть соединение port - Порт принтера. deviceVersion - Версия прошивки принтера (если драйвер не может установить это самостоятельно)</p>
<code>void SetLogo(Bitmap bmp)</code>	<p>Изменить логотип bmp - Картинка логотипа.</p>
<code>void SetPaymentTypes(Dictionary<PaymentType, string> paymentTypes)</code>	<p>Изменить названия пользовательских типов оплат paymentTypes - названия типов оплат.</p>

<code>void PrintRecCopy(int? number)</code>	Печать копии чека number - номер чека, если не задан то последний чек.
<code>void ReadRecCopy(int? number)</code>	Чтение копии чека number - номер чека, если не задан то последний чек.
<code>void PrintNullRec()</code>	Печать нулевого чека
<code>void OpenDrawer()</code>	Открыть кассовый ящик
<code>void Feed()</code>	Прогон чека на одну строку
<code>void CashIn(Money sum, decimal currencySum = 0, int currencyId = -1)</code>	Служебное внесение sum - Сумма в базовой валюте currencySum - Сумма в валюте currencyId - Идентификатор валюты
<code>void CashOut(Money sum, decimal currencySum = 0, int currencyId = -1)</code>	Служебное вынесение sum - Сумма в базовой валюте currencySum - Сумма в валюте currencyId - Идентификатор валюты
<code>decimal GetCashSum(int currencyId)</code>	Сумма в денежном ящике по валюте currencyId - Идентификатор валюты
<code>void ShowPaymentInfo(Money recSum, string recSumHint, Money changeSum, string changeSumHint)</code>	Отображает информацию об оплате чека на дисплее покупателя recSum - итоговая сумма чека recSumHint - текстовая подсказка для итоговой суммы чека changeSum - сумма сдачи. changeSumHint - текстовая подсказка для суммы сдачи
<code>void ShowItemInfo(string itemName, Quantity itemCount, Money itemCost, Money recSum)</code>	Отображает информацию о товарной позиции на дисплее покупателя itemName - название товара itemCount - количество товара itemCost - стоимость товара. recSum - текущая сумма чека
<code>void ClearDisplay()</code>	Очищает дисплей покупателя.
<code>void ShowDisplayLine(string line, DisplayLineType displayLineType)</code>	Отображает строку на дисплее покупателя. line - Строка. displayLineType - Тип строки дисплея.

Класс PrinterFeatures

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers.PrinterFeatures.

Описание: Возможности поддерживаемые фискальным принтером.

Свойства

Свойство	Описание
<code>bool RefundCashless</code>	Принтер поддерживает возвраты безналичных форм оплат.
<code>bool ArtRegistration</code>	Принтер ведет артикулярный учет.
<code>bool OnlineMode</code>	Принтер может печатать чек в реальном времени (после передачи позиции чека сразу печатает ее).
<code>int MaxRecItemCount</code>	Максимальное количество позиций в чеке.
<code>int TextDocLineLength</code>	Длинна строки чека.
<code>bool SeparateRecCounters</code>	Принтер разделяет номера чеков возврата и чеков продажи.
<code>bool CashBoxControl</code>	Принтер ведет контроль суммы в кассовом ящике.
<code>bool PrintNonFiscalDoc</code>	Принтер может печатать нефискальные чеки.
<code>bool PrintCheckCopy</code>	Принтер может напечатать копию последнего чека.
<code>bool PrintCommentsInCheck</code>	Принтер может печатать комментарии в чеке.
<code>bool PaymentSumControl</code>	Принтер ведет учет оборотов продаж и возвратов.
<code>bool SetTaxes</code>	Принтер может изменять свои налоговые ставки.
<code>int DisplayLen</code>	Длинна сообщения на индикаторе.
<code>bool PrintSavedSalesReport</code>	Принтер может печатать отчет по контролке.
<code>bool SetPaymentTypes</code>	Принтер может задавать названия типов оплат.
<code>int FiscalCommentLineLength</code>	Длина коммента в фискальном чеке.
<code>bool IsReadRecCopy</code>	Принтер может читать содержимое чека.
<code>bool IsMultyCurrency</code>	Принтер может работать с разными валютами.

Перечисление(Flags) PrinterOptions

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers.PrinterOptions.

Описание: Параметры работы принтера, передаваемые ему при инициализации.

Значения

Значение	Описание
<code>PrintOneCount</code>	Печатать количество товара если оно равно 1.
<code>ShowDisplayPrinterInfo</code>	Использовать штатные возможности принтера для показа информации на индикаторе.
<code>OnlineMode</code>	Режим прямой печати (не используется).
<code>PrintArtCode</code>	Печатать перед названием товара его код.
<code>PrintDiscountEachPosition</code>	Печатать скидку для каждого товара.
<code>PrintBarcodeFromGood</code>	Печатать на чеке штрих-коды товаров.
<code>PrintFullGoodName</code>	Печатать на чеке названия товаров целиком, не обрезать название.
<code>MultyCurrencyPayment</code>	Будут осуществляться мультивалютные платежи.

PrintPackNameNewLine	Печатать название упаковки отдельной строкой.
PrintPackNameAfterArtName	Печатать название упаковки после названия товара.
PrintTotalCostItemFromDiscount	Печать итоговой стоимости товарной позиции после применения скидки.
PrintPriceAdditionalCurrency	Печатать цену товара в дополнительной валюте.
PrintSavedSalesEndWorkDay	Печатать контрольную ленту в конце рабочего дня.

Перечисление ReceiptType

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers.ReceiptType.

Описание: Типы документов для печати на фискальном принтере.

Значения

Значение	Описание
Sales	Чек продажи.
Refund	Чек возврата.
Copy	Копия произвольного чека (нефискальный) .
Text	Текстовый нефискальный чек
NoFiscalSales	Нефискальный чек продажи.
NoFiscalRefund	Нефискальный чек возврата.

Перечисление ReportType

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers. ReportType.

Описание: Типы отчетов фискального принтера.

Значения

Значение	Описание
XReport	X-отчет.
ZReport	Z-отчет.
ArtReport	Отчет по артикулам.
SavedSales	Отчет контрольной ленты.

Перечисление TaxGrp

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers.TaxGrp.

Описание: Типы налоговых ставок.

Значения

Значение	Описание
TaxA	
TaxB	
TaxC	

TaxD	
TaxE	
TaxFree	

Структура PrinterTaxGrp

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers.PrinterTaxGrp.

Описание: Класс содержащий информацию о налоговой ставке принтера.

Конструкторы

Конструктор	Описание
PrinterTaxGrp(TaxType type, Percent rate, Percent[] taxes)	

Свойства

Свойство	Описание
TaxType Type	Тип налога
Percent Rate	Значение налога в процентах
Percent[] Taxes	Значения ставок сборов

Перечисление TaxType

Значение	Описание
InTaxType	Вложенный
OverTaxType	Наложный

Класс PrinterPayment

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers.PrinterPayment.

Описание: Клас содержащий информацию об оплате чека.

Свойства

Свойство	Описание
Money TotalSum	Общая сумма всех оплат.

Методы

Метод	Описание
Money GetSum(PaymentType type)	Получить сумму оплаты по типу. type - тип оплаты.

Перечисление PaymentType

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers.PaymentType.

Описание: Типы оплат.

Значения

Значение	Описание
Cash	Оплата наличными.
Card	Оплата карточкой.
Check	Оплата чеком.
Credit	Оплата в кредит.
Ext1	Оплата дополнительным типом оплаты 1.
Ext2	Оплата дополнительным типом оплаты 2.
Ext3	Оплата дополнительным типом оплаты 3.

Интерфейс IDevice

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.IDevice.

Описание: Общий интерфейс устройств.

```
public interface IDevice
{
    String DeviceName{get;}
    String DeviceFriendlyName{get;}
    Port DefaultPort{get;}
    PortType[] PortTypes{get;}
    Parameter[] Parameters{get;set;}
    bool IsOpened{get;}

    void Open(Port port);
    void Close();
    Parameter[] GetDefaultParameters(PortType portType);
    Parameter[] GetParameters(PortType portType);
    bool IsPortSupported(PortType portType);
}
```

Свойства

Свойство	Описание
<code>String</code> DeviceName	Уникальный идентификатор устройства желательно GUID.
<code>String</code> DeviceFriendlyName	Название устройства, отображаемое при настройке кассы. Обычно название протокола и его версия.
<code>Port</code> DefaultPort	Порт устройства по умолчанию.
<code>PortType[]</code> PortTypes	Типы портов, поддерживаемые драйвером.
<code>Parameter[]</code> Parameters	Дополнительные параметры устройства, которые редактируются пользователем в процессе настройки драйвера. Например: скорость СОМ порта и т. д.
<code>bool</code> IsOpened	Признак того, что соединение с устройством установлено.

Методы

Метод	Описание
<code>void Open (Port port)</code>	Устанавливает соединение с устройством. При ошибке соединения с устройством необходимо сгенерировать исключение <code>SoftMarket.Devices.DeviceException</code> с соответствующим описанием.
<code>void Close ()</code>	Разрывает соединение с устройством.
<code>bool IsPortSupported (PortType portType)</code>	Проверяет поддержку устройством указанного типа порта.
<code>Parameter[] GetParameters (PortType portType)</code>	Возвращает дополнительные параметры устройства для указанного порта.
<code>Parameter[] GetDefaultParameters (PortType portType)</code>	Возвращает дополнительные параметры устройства, заданные по умолчанию.

Класс Money

Сборка: `Globals.dll`.

Полное имя: `SoftMarket.Globals.Units.Money`.

Описание: Класс реализующий работу с деньгами.

Конструкторы

Конструктор	Описание
<code>Money (int amount)</code>	<code>Amount</code> - сумма в копейках.
<code>Money (decimal amount)</code>	<code>Amount</code> - сумма в копейках.

Свойства

Свойство	Описание
<code>int Amount {get;}</code>	Сумма в копейках.
<code>decimal DecimalAmount {get;}</code>	Сумма в копейках.

Класс Quantity

Сборка: `Globals.dll`.

Полное имя: `SoftMarket.Globals.Units.Quantity`.

Описание: Класс реализующий работу с количеством.

Свойства

Свойство	Описание
<code>int Amount {get;}</code>	Количество. Для штучных товаров в штуках, для весовых и делимых в граммах.
<code>QuantityTypes Type {get;}</code>	Тип количества. <code>Weight</code> - весовой, <code>NoWeight</code> - штучный, <code>Partial</code> - делимый.

Класс Port

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Port.

Описание: Класс для хранения данных порта устройства (LPT, COM, ETHERNET).

Свойства

Свойство	Описание
<code>int</code> Number{get;}	Номер порта.
<code>PortType</code> Type{get;}	Тип порта.
<code>string</code> IPAddress{get;}	IP адрес.

Перечисление PortType

Значения

Значение	Описание
COM	COM порта.
LPT	LPT порта.
KEYBOARD	Клавиатура
ETHERNET	Сеть Ethernet
VIRTUAL	Виртуального порта.
USB	USB порта.
COMAdapter	COM порта через переходник USB->COM, для Linux

Класс Parameter

Сборка: Globals.dll.

Полное имя: SoftMarket.Globals.Parameter.

Описание: Класс для работы с дополнительными настройками устройств.

Конструкторы

Конструктор	Описание
<code>Parameter(int key, int intValue, int minValue, int maxValue, string description)</code>	key - уникальный идентификатор параметра intValue - значение параметра minValue - минимальное значение параметра maxValue - максимальное значение параметра description - описание параметра
<code>Parameter(int key, int intValue, int minValue, int maxValue, string description, bool isSecure)</code>	key - уникальный идентификатор параметра intValue - значение параметра minValue - минимальное значение параметра maxValue - максимальное значение параметра description - описание параметра isSecure - отображать значение звездочками
<code>Parameter(int key, string stringValue, int minValue, int maxValue, string description)</code>	key - уникальный идентификатор параметра stringValue - значение параметра minValue - минимальная длина параметра

	maxValue - максимальная длина параметра description - описание параметра
<pre>Parameter(int key, string stringValue, int minValue, int maxValue, string description, bool isSecure)</pre>	key - уникальный идентификатор параметра stringValue - значение параметра minValue - минимальная длина параметра maxValue - максимальная длина параметра description - описание параметра isSecure - отображать значение звездочками
<pre>Parameter(int key, int intValue, int[] intEnumValues, string description)</pre>	key - уникальный идентификатор параметра intValue - значение параметра intEnumValues - возможные значения параметра description - описание параметра
<pre>Parameter(int key, string stringValue, string[] stringEnumValues, string description)</pre>	key - уникальный идентификатор параметра stringValue - значение параметра stringEnumValues - возможные значения параметра description - описание параметра
<pre>Parameter(int key, bool boolValue)</pre>	key - уникальный идентификатор параметра boolValue - значение параметра

Свойства

Свойство	Описание
<code>int Key{get;}</code>	Идентификатор параметра.
<code>int IntValue{get;set;}</code>	Целочисленное значение параметра.
<code>string StringValue{get;set;}</code>	Строковое значение параметра.
<code>bool BoolVal{get;set;}</code>	Булево значение параметра.
<code>string Description{get;}</code>	Описание параметра.
<code>int MaxValue{get;}</code>	Максимальное значение/длина параметра.
<code>int MinValue{get;}</code>	Минимальное значения/длина параметра.
<code>bool IsSecure{get;}</code>	Признак отображения значения звездочками.
<code>int[] IntEnumValues{get;}</code>	Список возможных целочисленных значений параметра.
<code>string[] StringEnumValues{get;}</code>	Список возможных строковых значений параметра.

Структура PrinterCurrency

Сборка: DriverGlobals.dll.

Полное имя: SoftMarket.Devices.Printers.PrinterCurrency.

Описание: Структура содержащая информацию о валюте фискального принтера.

Свойства

Свойство	Описание
<code>int Id{get;set;}</code>	Идентификатор валюты
<code>string Name{get;set;}</code>	Наименование валюты
<code>string Code{get;set;}</code>	Код валюты
<code>byte DecimalPlace{get;set;}</code>	Знаков после запятой
<code>decimal MinBanknote{get;set;}</code>	Минимальная денежная единица
<code>decimal Rate{get;set;}</code>	Курс валюты по отношению к базовой
<code>bool IsAddingCurrency{get;set;}</code>	Признак того что валюта является дополнительной
<code>bool IsBaseCurrency{get;set;}</code>	Признак того что валюта является базовой

Особенности реализации

Все исключения, генерируемые драйвером должны быть `SoftMarket.Devices.FiscalPrinterException` (сборка `DriverGlobals.dll`), или быть унаследованы от него. Если возникает исключение другого типа, его желательно перехватывать, логировать при помощи статического класса `Log.Write(err, null);`, и при необходимости генерировать `FiscalPrinterException`.

Пример:

```
try
{
}
catch (Exception err)
{
    Log.Write(err, this);
    throw new FiscalPrinterException(err.Message, err);
}
```

В случаях когда ошибка вызвана тем, что истекла смена кассира необходимо генерировать `Session24HoursException`.

Инициализация

Для инициализации драйвера вызываются два метода: `void Open(Port port)` и `void Init(string[] header, string[] footer, string[] firstHeader, string[] endFooter, string recNumberForNoFiscalReceipt, string recNumberForNoFiscalRefundReceipt, PrinterOptions options)`.

Метод `Open` устанавливает соединение с принтером.

Метод `Init` задает начальные настройки и режим работы принтера. Параметры `firstHeader` и `endFooter` используются для нефискальных принтеров, в них задаются реквизиты владельца.

Параметры `recNumberForNoFiscalRefundReceipt` и `recNumberForNoFiscalRefundReceipt` в качестве надписи «Чек №» для нефискальных чеков. В `options` задаются параметры работы принтера.

При помощи свойства `PrinterFeatures Features` касса запрашивает у драйвера список возможностей принтера. Так что это свойство должно быть инициализировано.

Синхронизация налоговых ставок

Касса может, как запрашивать, так и устанавливать налоговые ставки принтера при помощи свойства `Hashtable Taxes`. Значением свойства является список налоговых ставок в формате `Hashtable` в котором ключ- `TaxGrp` а значение- `PrinterTaxGrp`.

Печать документов

Касса может печатать на принтере шесть видов документов:

1. Фискальные продажи и возвраты.
2. Нефискальные продажи, возвраты и копии.
3. Произвольные текстовые документы.

Нефискальные документы должны при печати выглядеть точно также как и фискальные, но не проходить по фискальной памяти.

Во все документы может быть добавлен штрихкод (при помощи `PrintBarcode(string barcode)`), который должен быть напечатан к концу документа.

Печать документов начинается с `BeginReceipt()`. Для произвольного текстового также может быть использован метод `BeginTextDoc()`.

Передача позиций документа осуществляется методом `PrintReceiptItem(int artCode, string artName, Money unitPrice, Quantity count, string unitName, Money cost, string offerName, Money discount, Money recDiscount, TaxGrp taxGrp, int artNum, bool cancelled)`.

Позиции, помеченные как `cancelled`, на принтере обычно не печатаются.

Если принтер ведет артикулярный учет, то в качестве идентификатора товара необходимо использовать `artNum`.

Скидка на позицию состоит из двух частей:

1. Скидка непосредственно на позицию (`discount`).
2. Скидка на весь чек и соответственно приходящаяся ее часть на позицию (`recDiscount`).

В зависимости от настроек и возможностей принтера скидка может быть назначена принтером тремя способами:

1. Скидка назначается только на позицию (`discount + recDiscount`), и скидка на чек не назначается.
2. Скидка на позицию не назначается. Суммируются `discount + recDiscount` для всех позиций и назначаются на чек.
3. На позицию назначается только `discount`. А на чек назначается суммарный `recDiscount` (можно не подсчитывать так как дополнительно передается в `PrintRecDiscount(Money sum, string discountName)`).

Кроме того в документ могут быть дополнительно добавлены:

1. Информация о клиенте (`PrintClientInfo(string clientName)`).
2. Информация о скидке (`PrintRecDiscount(Money sum, string discountName)`).
3. Штрихкод (`PrintBarcode(string barcode)`).
4. Нижние комментарии (`PrintBottomComment(string[] comments)`).

Для текстового документа добавляются строки `PrintTextLine(string line)`.

Завершение документа осуществляется методом `EndReceipt(Money receiptSum, PrinterPayment payment)` или `EndReceipt()` для произвольного текстового. В параметре `payment` передаются суммы по типам оплат, их нужно получить и передать на принтер в необходимом порядке.

Сначала передаются безналичные оплаты, а затем наличные. Это необходимо, так как сдача может быть получена только с наличных оплат.

В процессе оплаты могут возникать исключительные ситуации следующих типов:

1. Ошибка, после которой чек можно отменить. Генерируется `FiscalPrinterException`, после чего касса повторяет оплату или отменяет документ.
2. Ошибка, после которой чек уже невозможно отменить, а можно только доплатить. Генерируется `CriticalPaymentException`, после чего касса только повторяет оплату.
1. Ошибка, возникающая когда команда об оплате передана на принтер, но связь с ним потеряна. В этой ситуации неизвестно закрылся чек на принтере или нет. Генерируется `LostConnectException`, после чего кассир принимает решение повторить оплату отменить чек или принять его.

Печать шаблонов

Шаблон это подготовленный текстовый документ. Для его печати используется метод `PrintTemplate(ITemplatePrintInfo template)`.

Пример реализации для фискального принтера:

```
public void PrintTemplate(ITemplatePrintInfo template)
```

```

{
    if(!printer.IsOpen)
        throw new FiscalPrinterException(CultureStrings.PrinterNotOpen);

    try
    {
        template.IsTape = true;
        template.PrinterType = PrinterType.String;
        template.PageSize = new Size(Features.TextDocLineLength, 0);

        FontRatio fontRatio = new FontRatio();
        fontRatio.MediumFontRatio.RatioH = 1;
        fontRatio.MediumFontRatio.RatioW = 1;

        fontRatio.LargeFontRatio.RatioH = 1;
        fontRatio.LargeFontRatio.RatioW = 1;

        template.FontRatio = fontRatio;

        BeginTextDoc();

        template.CreatePages();
        foreach(ITemplatePrintInfoPage page in template.Pages)
        {
            foreach(ITemplatePrintInfoPageItem item in page.PageItems)
            {
                if(item.PageItemType == PageItemType.String)
                {
                    PrintTextLine(item.StringItem);
                }
                else if(item.PageItemType == PageItemType.Barcode)
                {
                    barcode = item.StringItem;
                }
            }
        }

        EndReceipt();
    }
    catch(TemplateException err)
    {
        Log.Write(err, this);
        throw new DeviceException(err.Message, err);
    }
}

```