



OLE for Retail POS. Руководство по программированию

редакция 1

ТЕЛЕФОН

**8-495-770-18-94
8-495-770-18-95
8-495-301-05-93**

ФАКС

8-495-301-41-73

ВЕБ

WWW.MULTISOFT.RU

Содержание

Содержание	3
Введение	9
Возможности драйвера OPOS Fiscal Printer	11
Перечень возможностей OPOS Fiscal Printer	11
Свойства	13
Перечень свойств OPOS Fiscal Printer	13
Описание свойств OPOS Fiscal Printer	15
CapAdditionalHeader	15
CapAdditionalLines	15
CapAdditionalTrailer	15
CapAmountAdjustment	15
CapAmountNotPaid	15
CapChangeDue	15
CapCheckTotal	16
CapCompareFirmwareVersion	16
CapCoverSensor	16
CapDoubleWidth	16
CapDuplicateReceipt	16
CapEmptyReceiptIsVoidable	16
CapFiscalReceiptStation	16
CapFiscalReceiptType	17
CapFixedOutput	17
CapHasVatTable	17
CapIndependentHeader	17
CapItemList	17
CapJmEmptySensor	17
CapJmNearEndSensor	17
CapJmPresent	18
CapMultiContractor	18
CapNonFiscalMode	18
CapOnlyVoidLastItem	18
CapOrderAdjustmentFirst	18
CapPackageAdjustment	18
CapPercentAdjustment	18
CapPositiveAdjustment	19
CapPositiveSubtotalAdjustment	19
CapPostPreLine	19
CapPowerLossReport	19
CapPowerReporting	19
CapPredefinedPaymentLines	19
CapReceiptNotPaid	20
CapRecEmptySensor	20
CapRecNearEndSensor	20
CapRecPresent	20
CapRemainingFiscalMemory	20
CapReservedWord	20
CapSetCurrency	20
CapSetHeader	21
CapSetPOSID	21
CapSetStoreFiscalID	21
CapSetTrailer	21
CapSetVatTable	21
CapSlpEmptySensor	21
CapSlpFiscalDocument	21

CapSlpFullSlip.....	22
CapSlpNearEndSensor.....	22
CapSlpPresent.....	22
CapSlpValidation.....	22
CapStatisticsReporting.....	22
CapSubAmountAdjustment.....	22
CapSubPercentAdjustment.....	23
CapSubtotal.....	23
CapTotalizerType.....	23
CapTrainingMode.....	23
CapUpdateFirmware.....	23
CapUpdateStatistics.....	23
CapValidateJournal.....	24
CapXReport.....	24
ActualCurrency.....	24
AdditionalHeader.....	24
AdditionalTrailer.....	25
AmountDecimalPlaces.....	25
AsyncMode.....	25
BinaryConversion.....	25
ChangeDue.....	26
CheckHealthText.....	26
CheckTotal.....	26
Claimed.....	26
ContractorID.....	26
ControlObjectDescription.....	27
ControlObjectVersion.....	27
CountryCode.....	27
CoverOpen.....	27
DateType.....	28
DayOpened.....	28
DescriptionLength.....	28
DeviceDescription.....	28
DeviceEnabled.....	29
DeviceName.....	29
DuplicateReceipt.....	29
ErrorLevel.....	29
ErrorOutID.....	29
ErrorState.....	30
ErrorStation.....	30
ErrorString.....	30
FiscalReceiptStation.....	30
FiscalReceiptType.....	31
FlagWhenIdle.....	31
FreezeEvents.....	31
JmEmpty.....	31
JmNearEnd.....	32
MessageLength.....	32
MessageType.....	32
NumHeaderLines.....	33
NumTrailerLines.....	33
NumVatRates.....	33
OpenResult.....	33
OutputID.....	33
PostLine.....	34
PowerNotify.....	34
PowerState.....	34
PredefinedPaymentLines.....	34
PreLine.....	35
PrinterState.....	35
QuantityDecimalPlaces.....	35

QuantityLength	36
RecEmpty	36
RecNearEnd	36
RemainingFiscalMemory	36
ReservedWord	36
ResultCode	37
ResultCodeExtended	37
ServiceObjectDescription	37
ServiceObjectVersion	37
SlipSelection	38
SlpEmpty	38
SlpNearEnd	38
State	38
TotalizerType	39
TrainingModeActive	39
Методы	41
Перечень методов OPOS Fiscal Printer	41
Описание методов OPOS Fiscal Printer	43
Open	43
Close	43
ClaimDevice	43
ReleaseDevice	43
CheckHealth	43
ClearOutput	43
DirectIO	44
Получить список команд	45
Тест	46
Получить описание для специфической команды	46
Получить описание кода ошибки	48
Записать параметр специфической команды с параметрами	48
Выполнить специфическую команду с параметрами	48
CompareFirmwareVersion	49
ResetStatistics	49
RetrieveStatistics	49
UpdateFirmware	49
UpdateStatistics	49
SetCurrency	49
SetDate	49
SetHeaderLine	50
SetPOSID	50
SetStoreFiscalID	50
SetTrailerLine	50
SetVatTable	51
SetVatValue	51
BeginFiscalReceipt	51
EndFiscalReceipt	51
PrintDuplicateReceipt	52
PrintRecCash	52
PrintRecItem	52
PrintRecItemVoid	53
PrintRecItemAdjustment	53
PrintRecItemAdjustmentVoid	54
PrintRecItemFuel	54
PrintRecItemFuelVoid	54
PrintRecMessage	54
PrintRecNotPaid	54
PrintRecPackageAdjustment	54
PrintRecPackageAdjustVoid	55
PrintRecRefund	55

PrintRecRefundVoid	55
PrintRecSubtotal	56
PrintRecSubtotalAdjustment	56
PrintRecSubtotalAdjustVoid	56
PrintRecTaxID	56
PrintRecTotal	57
PrintRecVoid	57
PrintRecVoidItem	57
BeginFiscalDocument	57
EndFiscalDocument	57
PrintFiscalDocumentLine	57
BeginItemList	58
EndItemList	58
VerifyItem	58
PrintPeriodicTotalsReport	58
PrintPowerLossReport	58
PrintReport	58
PrintXReport	58
PrintZReport	59
BeginInsertion	59
BeginRemoval	59
EndInsertion	59
EndRemoval	59
BeginFixedOutput	59
BeginNonFiscal	59
BeginTraining	59
EndFixedOutput	59
EndNonFiscal	60
EndTraining	60
PrintFixedOutput	60
PrintNormal	60
GetData	60
GetDate	61
GetTotalizer	61
GetVatEntry	61
ClearError	62
ResetPrinter	62
Пример формирования фискального документа	63
Настройки реестра	65
Параметры OPOS Fiscal Printer драйвера в реестре	65
Параметры протокола обмена HID	65
Ссылки на дополнительные материалы	67
Приложение А. Основные положения OLE for Retail POS	69
Что такое OLE for Retail POS?	69
Базовая модель Control-объектов OLE for Retail POS	69
Определения	70
Device Class	70
Control Object or CO	70
Service Object or SO	70
OPOS Control or Control	70
Как приложение использует OPOS Control	71
Когда методы и свойства могут быть доступны	72
Methods	72
Properties	72

Модели Status, Result code и State	73
Модель Status	74
Модель Result Code	74
Модель State	74
Модель доступа к устройству	76
Устройства эксклюзивного использования	76
Устройства с разделяемым доступом	76
События	77
Модель вывода	78
Синхронный вывод	78
Асинхронный вывод	78
Приложение Б. Команды Star Line Mode	79
Разрешенные команды	79
Управление шириной/высотой символов	79
Печать штрих-кодов	80
Выделение текста	80
Отмена выделения текста	80
Управление автоотрезчиком	80
Протяжка бумаги	81
Установка Memory switch	81
Сброс принтера	81
Инициализация сигнала для внешнего устройства	81
Подать сигнал для внешнего устройства	82
Возврат бумаги презентером	82
Приложение В. Команды ESC/POS	83
Разрешенные команды	83
Установка режима работы автоотрезчика	83
Печать в буфер и протяжка бумаги	84
Печать в буфер без протяжки бумаги	84
Установка режима печати символов	84
Инициализация сигнала денежному ящику	85
Управляющий сигнал на первый канал денежного ящика	85
Управляющий сигнал на второй канал денежного ящика	85
Печать штрихкода	86
Приложение Г. Специфические команды DirectIO	87
Специфические команды без параметров	87
Считать данные фискального накопителя	88
Считать программу фискального накопителя	89
Считать дескриптор фискального накопителя	90
Считать программу процессора	91
Печать Z-отчета из буфера	92
Печать оперативного отчета налогового инспектора	93
Активизация ЭКЛЗ	94
Закрытие архива ЭКЛЗ	95
Итог активизации ЭКЛЗ	96
Регистрация пользователя	97
Запись в устройство	98
Запрос данных	99
Установка функции порта	100
Аварийное закрытие смены	101
Запись логотипа	102
Перезагрузка	103
Запрос данных из ЭКЛЗ	104
Запрос состояния ЭКЛЗ	105

Запрос регистрационного номера ЭКЛЗ.....	107
Тест целостности архива ЭКЛЗ.....	108
Запрос итога активизации.....	108
Запрос контрольной ленты.....	109
Запрос документа.....	110
Запрос отчета.....	111
Запрос итогов смен.....	113
Получение данных по запрошенному отчету.....	114
Прекращение.....	119
Специфические команды с параметрами.....	120
Программирование серийного номера.....	120
Фискализация/Перерегистрация.....	121
Инициализация Фискального накопителя.....	122
Отчет ЭКЛЗ по итогам смен в диапазоне дат.....	122
Отчет ЭКЛЗ по итогам смен в диапазоне номеров.....	123
Отчет ЭКЛЗ Документ по номеру КПК.....	124
Отчет ЭКЛЗ Контрольная лента по номеру смены.....	125
Отчет ЭКЛЗ Итоги смены по номеру смены.....	125
Установить пароль на связь.....	126
Установка лицензии.....	127
Типы оплат.....	127
Список таблиц.....	129
Методы и свойства.....	131

Введение

В этом документе описываются свойства и особенности работы контрольно-кассовой техники (ККТ) серии MSTAR и MSPrint (ПТК «MSTAR-ТК», ПТК «MSTAR-650ТК», ПТК «MSTAR-TUP-K», ПТК «MSTAR- Kiosk», ПТК «MSPrint- 80K») и машин электронных регистрирующих (МЭР «MSTAR», МЭР «MSTAR- 650», МЭР «MSPrint- 80»). Руководство предназначено для разработчиков приложений прикладного уровня, знакомых с положениями стандарта UnifiedPOS версии 1.11 в части, касающейся устройств класса Fiscal Printer и его реализацией в рамках «OLE for Retail POS».

В тексте используются следующие понятия и сокращения:

- ККТ – Контрольно-кассовая техника;
- ПТК – Программно-технический комплекс;
- ФП – Фискальная память ККТ (размещена под корпусом ККТ);
- ЭКЛЗ – Электронная контрольная лента защищенная (установлена под корпусом ККТ);
- МЭР – Машина электронная регистрирующая;
- ПС – Память сменных итогов (размещена под корпусом МЭР);
- НИ – Налоговый инспектор;
- ОС – Операционная система;
- ПО – Программное обеспечение;
- Устройство – Под этим термином подразумевается используемая модель принтера MSTAR или MSPrint;
- Фискальный чек – Под этим термином подразумевается чек (документ об оплате), печать которого сопровождается соответствующей записью в ФП устройства (для ККТ MSTAR или ККТ MSPrint-80K) или в ПС (для МЭР MSTAR или МЭР MSPrint-80), соответственно, печать чека должна проводится в фискальном (для ККТ MSTAR или ККТ MSPrint-80K) или учетном (для МЭР MSTAR или МЭР MSPrint-80) режиме; С этим термином связаны: «Фискальная команда», «Фискальный режим»

Возможности драйвера OPOS Fiscal Printer

Перечень возможностей OPOS Fiscal Printer

Перечень возможностей OPOS Fiscal Printer приведен в таблице. (Таблица 1)

Таблица 1. Перечень возможностей OPOS-драйвера.

Свойство	Значение					
	MSTAR-TK	MSTAR-TUP-K	MSTAR-KiosK	МЭР	MSTAR	MSPrint-80K МЭР MSPrint-80
CapAdditionalHeader	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapAdditionalLines	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapAdditionalTrailer	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapAmountAdjustment	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapAmountNotPaid	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
CapChangeDue	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapCheckTotal	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapCompareFirmwareVersion	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapCoverSensor	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapDoubleWidth	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapDuplicateReceipt	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapEmptyReceiptIsVoidable	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapFiscalReceiptStation	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapFiscalReceiptType	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapFixedOutput	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapHasVatTable	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapIndependentHeader	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapItemList	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapJmEmptySensor	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapJmNearEndSensor	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapJmPresent	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapMultiContractor	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapNonFiscalMode	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapOnlyVoidLastItem	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapOrderAdjustmentFirst	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapPackageAdjustment	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapPercentAdjustment	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapPositiveAdjustment	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapPositiveSubtotalAdjustment	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapPostPreLine	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapPowerLossReport	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapPowerReporting	OPOS_PR_STANDARD					
CapPredefinedPaymentLines	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapReceiptNotPaid	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapRecEmptySensor	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapRecNearEndSensor	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapRecPresent	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapRemainingFiscalMemory	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapReservedWord	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
CapSetCurrency	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
CapSetHeader	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapSetPOSID	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

Свойство	Значение					
	MSTAR-TK	MSTAR-TUP-K	MSTAR-KiosK	МЭР	MSTAR	MSPrint-80K МЭР MSPrint-80
CapSetStoreFiscalID	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapSetTrailer	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapSetVatTable	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapSlpEmptySensor	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapSlpFiscalDocument	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapSlpNearEndSensor	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapSlpPresent	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapSlpValidation	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapStatisticsReporting	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapSubAmountAdjustment	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapSubPercentAdjustment	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapSubtotal	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapTotalizerType	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapTrainingMode	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapUpdateFirmware	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
CapUpdateStatistics	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapValidateJournal	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapXReport	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
CapUpdateStatistics	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

Свойства

Перечень свойств OPOS Fiscal Printer

Перечень свойств OPOS Fiscal Printer приведен в таблице. (Таблица 2).

Таблица 2. Перечень свойств OPOS-драйвера.

Свойство	Тип данных	Доступ	Можно использовать после
ActualCurrency	Long	R	Open, Claim & Enable
AdditionalHeader	String	R/W	Open, Claim & Enable
AdditionalTrailer	String	R/W	Open, Claim & Enable
AmountDecimalPlaces	Long	R	Open, Claim & Enable
AsyncMode	Boolean	R/W	Open
BinaryConversion	Long	R/W	Open
ChangeDue	String	R/W	Open
CheckHealthText	String	R	Open
CheckTotal	Boolean	R/W	Open
Claimed	Boolean	R	Open
ContractorID	Long	R/W	Open, Claim & Enable
ControlObjectDescription	String	R	—
ControlObjectVersion	Long	R	—
CountryCode	Long	R	Open, Claim & Enable
CoverOpen	Boolean	R	Open, Claim & Enable
DateType	Long	R/W	Open, Claim & Enable
DayOpened	Boolean	R	Open, Claim & Enable
DescriptionLength	Long	R	Open
DeviceDescription	String	R	Open
DeviceEnabled	Boolean	R/W	Open & Claim
DeviceName	String	R	Open
DuplicateReceipt	Boolean	R/W	Open
ErrorLevel	Long	R	Open
ErrorOutID	Long	R	Open, Claim & Enable
ErrorState	Long	R	Open
ErrorStation	Long	R	Open
ErrorString	String	R	Open
FiscalReceiptStation	Long	R/W	Open, Claim & Enable
FiscalReceiptType	Long	R/W	Open, Claim & Enable
FlagWhenIdle	Boolean	R/W	Open
FreezeEvents	Boolean	R/W	Open
JrnEmpty	Boolean	R	Open, Claim & Enable
JrnNearEnd	Boolean	R	Open, Claim & Enable
MessageLength	Long	R	Open
MessageType	Long	R/W	Open
NumHeaderLines	Long	R	Open
NumTrailerLines	Long	R	Open
NumVatRates	Long	R	Open
OpenResult	Long	R	—
OutputID	Long	R	Open
PostLine	String	R/W	Open, Claim & Enable

Свойство	Тип данных	Доступ	Можно использовать после
PowerNotify	Long	R/W	Open
PowerState	Long	R	Open
PredefinedPaymentLines	String	R	Open
PreLine	String	R/W	Open, Claim & Enable
PrinterState	Long	R	Open, Claim & Enable
QuantityDecimalPlaces	Long	R	Open, Claim & Enable
QuantityLength	Long	R	Open, Claim & Enable
RecEmpty	Boolean	R	Open, Claim & Enable
RecNearEnd	Boolean	R	Open, Claim & Enable
RemainingFiscalMemory	Long	R	Open, Claim & Enable
ReservedWord	String	R	Open
ResultCode	Long	R	—
ResultCodeExtended	Long	R	—
ServiceObjectDescription	String	R	Open
ServiceObjectVersion	Long	R	Open
SlipSelection	Long	R/W	Open, Claim & Enable
SlipEmpty	Boolean	R	Open, Claim & Enable
SlipNearEnd	Boolean	R	Open, Claim & Enable
State	Long	R	—
TotalizerType	Long	R/W	Open, Claim & Enable
TrainingModeActive	Boolean	R	Open, Claim & Enable

Описание свойств OPOS Fiscal Printer

CapAdditionalHeader

Синтаксис: *BOOL CapAdditionalHeader* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство может печатать текст, заданный в свойстве *AdditionalHeader*, в заголовке документа.

Инициализируется: методом *open-claim*

CapAdditionalLines

Синтаксис: *BOOL CapAdditionalLines* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство может печатать на фискальном чеке между итогом и концом чека любой текст, заданный приложением. Печать текста осуществляется методом *PrintRecMessage*

Инициализируется: методом *open-claim*

Смотри также: метод *PrintRecMessage*

CapAdditionalTrailer

Синтаксис: *BOOL CapAdditionalTrailer* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство может печатать текст, заданный в свойстве *AdditionalTrailer*, в конце документа.

Инициализируется: методом *open-claim*

CapAmountAdjustment

Синтаксис: *BOOL CapAmountAdjustment* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство может делать количественные скидки/наценки на операцию продажи.

Инициализируется: методом *open-claim*

CapAmountNotPaid

Свойство не используется, начиная с версии *Control Object 1.11.00*

CapChangeDue

Синтаксис: *BOOL CapChangeDue* {только чтение, доступно после *open*}

Замечания: если *TRUE*, то, при использовании метода *printRecTotal*, текст, который печатается в строке со сдачей, может быть задан в свойстве *ChangeDue*

Инициализируется: методом *open*

CapCheckTotal

Синтаксис: *BOOL CapCheckTotal* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда автоматическое сравнение итога, посчитанного в устройстве, и итога, посчитанного приложением, может быть разрешено или запрещено. Если *FALSE*, тогда автоматическое сравнение не может быть разрешено или запрещено. Это означает, что свойство *CheckTotal* не может быть изменено и доступно только для чтения.

Инициализируется: методом *open-claim*

Смотри также: свойство *CheckTotal*

CapCompareFirmwareVersion

Синтаксис: *BOOL CapCompareFirmwareVersion* {только чтение, доступ после *open*}

Замечания: если *TRUE*, тогда устройство поддерживает сравнение своей версии ПО с версией ПО в файле.

Смотри также: метод *CompareFirmwareVersion*

CapCoverSensor

Синтаксис: *BOOL CapCoverSensor* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство имеет датчик открытия крышки.

Инициализируется: методом *open*

CapDoubleWidth

Синтаксис: *BOOL CapDoubleWidth* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство может печатать символы с двойной шириной.

Инициализируется: методом *open-claim*

CapDuplicateReceipt

Синтаксис: *BOOL CapDuplicateReceipt* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство может распечатать копию фискального чека.

Инициализируется: методом *open-claim*

CapEmptyReceiptIsVoidable

Синтаксис: *BOOL CapEmptyReceiptIsVoidable* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда можно отменить не закрытый документ, в котором нет продаж.

Инициализируется: методом *open-claim*

CapFiscalReceiptStation

Синтаксис: *BOOL CapFiscalReceiptStation* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает печать на станции, определенной в свойстве *FiscalReceiptStation*

Инициализируется: методом *open-claim*

CapFiscalReceiptType

Синтаксис: *BOOL CapFiscalReceiptType* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает печать разных типов фискальных чеков. Тип должен быть определен в свойстве *FiscalReceiptType*

Инициализируется: методом *open-claim*

CapFixedOutput

Синтаксис: *BOOL CapFixedOutput* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает текст фиксированного формата, который можно напечатать с помощью методов *beginFixedOutput*, *printFixedOutput* и *endFixedOutput*

Инициализируется: методом *open-claim*

CapHasVatTable

Синтаксис: *BOOL CapHasVatTable* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство имеет таблицу налоговых ставок.

Инициализируется: методом *open-claim*

CapIndependentHeader

Синтаксис: *BOOL CapIndependentHeader* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает печать заголовка чека до того как поступит первая фискальная команда (см. определение фискального чека в разделе [\(См. "Введение" на стр. 9\)](#)).

Инициализируется: методом *open-claim*

CapItemList

Синтаксис: *BOOL CapItemList* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство может напечатать отчет по списку товаров.

Инициализируется: методом *open-claim*

CapJrnEmptySensor

Синтаксис: *BOOL CapJrnEmptySensor* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда принтер контрольной ленты имеет датчик бумаги.

Инициализируется: методом *open-claim*

CapJrnNearEndSensor

Синтаксис: *BOOL CapJrnNearEndSensor* {только чтение, доступно после *open*}

Замечания: если *TRUE*, тогда принтер контрольной ленты имеет датчик «заканчивается бумага».

Инициализируется: методом *open*

CapJrnPresent

Синтаксис: *BOOL CapJrnPresent* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство имеет принтер контрольной ленты.

Инициализируется: методом *open-claim*

CapMultiContractor

Синтаксис: *BOOL CapMultiContractor* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает больше чем одного владельца. Т.е несколько владельцев могут пользоваться устройством одновременно.

Инициализируется: методом *open-claim*

CapNonFiscalMode

Синтаксис: *BOOL CapNonFiscalMode* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство позволяет работать в не фискальном режиме (см. определение фискального чека в разделе [\(См. "Введение" на стр. 9\)](#)).

Инициализируется: методом *open-claim*

CapOnlyVoidLastItem

Синтаксис: *BOOL CapOnlyVoidLastItem* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда может быть сторнирована только последняя операция в документе.

Инициализируется: методом *open-claim*

CapOrderAdjustmentFirst

Синтаксис: *BOOL CapOrderAdjustmentFirst* {только чтение, доступно после *open-claim*}

Замечания: если *FALSE*, то сначала должен быть выполнен метод *printRecItem*, а затем уже *printRecItemAdjustment*

Если *TRUE*, тогда приложение может вызвать метод *printRecItemAdjustment* до метода *printRecItem*

Инициализируется: методом *open-claim*

CapPackageAdjustment

Синтаксис: *BOOL CapPackageAdjustment* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, то скидка/ наценка может быть применена к группе операций продажи.

Инициализируется: методом *open-claim*

CapPercentAdjustment

Синтаксис: *BOOL CapPercentAdjustment* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство может делать процентные скидки/надбавки на операцию продажи.

Инициализируется: методом *open-claim*

CapPositiveAdjustment

Синтаксис: *BOOL CapPositiveAdjustment* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда метод *printRecItemAdjustment* используется и для скидок и для наценок.

Инициализируется: методом *open-claim*

CapPositiveSubtotalAdjustment

Синтаксис: *BOOL CapPositiveSubtotalAdjustment* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда метод *printRecSubtotalAdjustment* используется и для скидок и для наценок.

Инициализируется: методом *open-claim*

CapPostPreLine

Синтаксис: *BOOL CapPostPreLine* {только чтение, доступно после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает печать дополнительных строк, определенных в свойствах *PostLine* и/или *PreLine*

Инициализируется: методом *open-claim*

CapPowerLossReport

Синтаксис: *BOOL CapPowerLossReport* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство может печатать отчеты о потерях питания с помощью метода *printPowerLossReport*

Инициализируется: методом *open-claim*

CapPowerReporting

Синтаксис: *LONG CapPowerReporting* {только чтение, доступ после *open*}

Замечания: показывает возможности устройства сообщать о потерях питания.

Значения:

- *OPOS_PR_NONE* – *Service Object* не может определить состояние устройства. Поэтому невозможно сообщить о потере питания;
- *OPOS_PR_STANDARD* – *Service Object* может определить и сообщить два состояния – *OFF*, *OFFLINE* и *ONLINE*;
- *OPOS_PR_ADVANCED* – *Service Object* может определить и сообщить все три состояния – *OFF*, *OFFLINE* и *ONLINE*.

Инициализируется: методом *open*

CapPredefinedPaymentLines

Синтаксис: *BOOL CapPredefinedPaymentLines* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, то свойство *PredefinedPaymentLines* содержит список допустимых индексов видов платежей.

Инициализируется: методом *open-claim*

CapReceiptNotPaid

Синтаксис: *BOOL CapReceiptNotPaid* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает метод *printRecNotPaid* чтобы обозначить неоплаченную часть итога.

Инициализируется: методом *open-claim*

CapRecEmptySensor

Синтаксис: *BOOL CapRecEmptySensor* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда чековый принтер имеет датчик отсутствия бумаги.

Инициализируется: методом *open-claim*

CapRecNearEndSensor

Синтаксис: *BOOL CapRecNearEndSensor* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда чековый принтер имеет датчик «бумага скоро закончится».

Инициализируется: методом *open-claim*

CapRecPresent

Синтаксис: *BOOL CapRecPresent* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство имеет принтер чеков.

Инициализируется: методом *open-claim*

CapRemainingFiscalMemory

Синтаксис: *BOOL CapRemainingFiscalMemory* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда в свойстве *RemainingFiscalMemory* будет указано оставшееся количество сменных записей в ФП или ПС. Если *FALSE*, то устройство не поддерживает отчет о количестве оставшихся сменных записей.

Инициализируется: методом *open-claim*

CapReservedWord

Синтаксис: *BOOL CapReservedWord* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство печатает зарезервированное слово (например, "ИТОГ") до значения почитанного итога, и это зарезервированное слово хранится в свойстве *ReservedWord*

Инициализируется: методом *open-claim*

CapSetCurrency

Синтаксис: *BOOL CapSetCurrency* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает смену типов валют с помощью метода *setCurrency*

Инициализируется: методом *open-claim*

CapSetHeader

Синтаксис: *BOOL CapSetHeader* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда можно использовать метод *setHeaderLine* для установки заголовка документа.

Инициализируется: методом *open-claim*

CapSetPOSID

Синтаксис: *BOOL CapSetPOSID* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда можно использовать метод *setPOSID* для инициализации значений *POSID* и *CashierID*

Инициализируется: методом *open-claim*

CapSetStoreFiscalID

Синтаксис: *BOOL CapSetStoreFiscalID* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда можно использовать метод *SetStoreFiscalID* для установления серийного номера.

Инициализируется: методом *open-claim*

CapSetTrailer

Синтаксис: *BOOL CapSetTrailer* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда можно использовать метод *setTrailerLine* для установки текста, печатаемого по окончании фискального чека.

Инициализируется: методом *open-claim*

CapSetVatTable

Синтаксис: *BOOL CapSetVatTable* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда можно использовать методы *setVatValue* и *setVatTable* установки ставок налогов.

Инициализируется: методом *open-claim*

CapSlpEmptySensor

Синтаксис: *BOOL CapSlpEmptySensor* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда у принтера подкладной печати есть датчик "*slip in*"

Инициализируется: методом *open-claim*

Смотри также: свойства *CapSlpPresent*, *CapSlpFiscalDocument*

CapSlpFiscalDocument

Синтаксис: *BOOL CapSlpFiscalDocument* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство имеет принтер подкладной печати и может печатать на нем подкладной документ.

Инициализируется: методом *open-claim*

CapSlpFullSlip

Синтаксис: *BOOL CapSlpFullSlip* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает печать на *slip* принтере *full lengths forms*

Инициализируется: методом *open-claim*

Смотри также: свойства *CapSlpPresent*, *CapSlpFiscalDocument*

CapSlpNearEndSensor

Синтаксис: *BOOL CapSlpNearEndSensor* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда у принтера подкладной печати есть датчик "*slip near end*"

Инициализируется: методом *open-claim*

Смотри также: свойства *CapSlpPresent*, *CapSlpFiscalDocument*

CapSlpPresent

Синтаксис: *BOOL CapSlpPresent* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство имеет принтер подкладной печати.

Инициализируется: методом *open-claim*

Смотри также: свойства *CapSlpFiscalDocument*

CapSlpValidation

Синтаксис: *BOOL CapSlpValidation* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает печать на *slip* принтере проверочную информацию.

Инициализируется: методом *open-claim*

Смотри также: свойства *CapSlpPresent*, *CapSlpFiscalDocument*

CapStatisticsReporting

Синтаксис: *BOOL CapStatisticsReporting* {только чтение, доступ после *open*}

Замечания: если *TRUE*, устройство собирает и может предоставить различную статистику своего использования; иначе статистика не собирается. Для получения статистики используется метод *RetrieveStatistics*

Инициализируется: методом *open*

Смотри также: метод *RetrieveStatistics*

CapSubAmountAdjustment

Синтаксис: *BOOL CapSubAmountAdjustment* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает скидки/наценки на подитог.

Инициализируется: методом *open-claim*

CapSubPercentAdjustment

Синтаксис: *BOOL CapSubPercentAdjustment* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает процентные скидки на подитог.

Инициализируется: методом *open-claim*

CapSubtotal

Синтаксис: *BOOL CapSubtotal* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда можно использовать метод *printRecSubtotal* для печати промежуточного итога.

Инициализируется: методом *open-claim*

CapTotalizerType

Синтаксис: *BOOL CapTotalizerType* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает чтение разных типов счетчиков с помощью метода *getTotalizer*

Инициализируется: методом *open-claim*

CapTrainingMode

Синтаксис: *BOOL CapTrainingMode* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда устройство поддерживает режим обучения.

Инициализируется: методом *open-claim*

CapUpdateFirmware

Синтаксис: *BOOL CapUpdateFirmware* {только чтение, доступ после *open*}

Замечания: если *TRUE*, тогда ПО устройства может быть обновлено с помощью метода *UpdateFirmware*

Смотри также: метод *UpdateFirmware*

CapUpdateStatistics

Синтаксис: *BOOL CapUpdateStatistics* {только чтение, доступ после *open*}

Замечания: если *TRUE*, статистика устройства или другая статистика, может быть сброшена в ноль с помощью метода *ResetStatistics* или обновлена методом *UpdateStatistics*. Если *CapStatisticsReporting* равно *FALSE*, тогда *CapUpdateStatistics* также равно *FALSE*

Инициализируется: методом *open*

Смотри также: свойство *CapStatisticsReporting*, метод *ResetStatistics*, метод *UpdateStatistics*, метод *UpdateFirmware*

CapValidateJournal

Синтаксис: *BOOL CapValidateJournal* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда можно использовать метод *printNormal* для печати на принтере контрольной ленты.

Инициализируется: методом *open-claim*

CapXReport

Синтаксис: *BOOL CapXReport* {только чтение, доступ после *open-claim*}

Замечания: если *TRUE*, тогда можно использовать метод *printXReport* для печати X-отчета.

Инициализируется: методом *open-claim*

ActualCurrency

Синтаксис: *LONG ActualCurrency* {только чтение, доступ после *open-claim-enable*}

Замечания: содержит тип используемой в устройстве валюты. Это свойство действительно, если *CapSetCurrency* равно *TRUE*

Значения:

- *FPTR_AC_BRC* – текущая валюта – Бразильская;
- *FPTR_AC_BGL* – текущая валюта – Болгарская;
- *FPTR_AC_EUR* – текущая валюта – ЕВРО;
- *FPTR_AC_GRD* – текущая валюта – Греческая;
- *FPTR_AC_HUF* – текущая валюта – Венгерская;
- *FPTR_AC_ITL* – текущая валюта – Итальянская;
- *FPTR_AC_PLZ* – текущая валюта – Польская;
- *FPTR_AC_ROL* – текущая валюта – Румынская;
- *FPTR_AC_RUR* – текущая валюта – Российская;
- *FPTR_AC_TRL* – текущая валюта – Турецкая;
- *FPTR_AC_CZK* – текущая валюта – Чешская;
- *FPTR_AC_UAH* – текущая валюта – Украинская;
- *FPTR_AC_OTHER* – текущая валюта – неизвестна.

Значение свойства актуально только когда *DeviceEnabled* = *TRUE*

Смотри также: метод *setCurrency*, свойство *CapSetCurrency*

AdditionalHeader

Синтаксис: *STRING AdditionalHeader* {чтение/запись, доступ после *open-claim-enable*}

Замечания: содержит заголовок, который печатается при вызове метода *beginFiscalReceipt*

Свойство действительно если *CapAdditionalHeader* равно *TRUE*

Инициализируется: пустой строкой и актуально пока *DeviceEnabled* равно *TRUE*

Смотри также: метод *beginFiscalReceipt*, свойство *CapAdditionalHeader*

AdditionalTrailer

Синтаксис: *STRING AdditionalTrailer* {чтение/запись, доступ после *open-claim-enable*}

Замечания: содержит текст, который печатается в конце чека при вызове метода *endFiscalReceipt*

Свойство действительно если *CapAdditionalTrailer* равно *TRUE*

Инициализируется: пустой строкой и актуально пока *DeviceEnabled* равно *TRUE*

Смотри также: метод *endFiscalReceipt*, свойство *CapAdditionalTrailer*

AmountDecimalPlaces

Синтаксис: *LONG AmountDecimalPlaces* {только чтение, доступ после *open-claim-enable*}

Замечания: содержит количество десятичных разрядов, которые используются для вычисления. Значение актуально только тогда, когда *CapAdditionalHeader* равно *TRUE*

AsyncMode

Синтаксис: *BOOL AsyncMode* {чтение/запись, доступ после *open*}

Замечания: если *TRUE*, тогда некоторые методы, такие как *printRecItemAdjustment*, *printRecItem*, *printNormal*, и т.д. могут быть выполнены в асинхронном режиме. Если *FALSE*, все методы выполняются синхронно.

Инициализируется: значением *FALSE* при вызове метода *open*

BinaryConversion

Синтаксис: *LONG BinaryConversion* {чтение/запись, доступ после *open*}

Замечания: определяет формат данных в параметре *Data* метода *PrintNormal*

Допустимые значения:

- *OPOS_BC_NONE* – каждый символ передается в устройство без изменения;
- *OPOS_BC_DECIMAL* – каждый байт преобразуется в 3 символа. (Самый простой способ преобразования).
 - Пример 1: Байтовое значение $154=0x9A$ становится символами $0x31\ 0x35\ 0x34$ (= строка "154").
 - Пример 2: Байтовое значение 8 становится символами $0x30\ 0x30\ 0x38$ (= строка "008").
- *OPOS_BC_NIBBLE* – каждый байт преобразуется в 3 символа. (Самый простой способ преобразования). Байт конвертируется следующим образом:
 - Первый символ = $0x30$ – биты 7-4 байта.
 - Второй символ = $0x30$ – биты 3-0 байта.
 - Пример: Байтовое значение $154=0x9A$ становится символами $0x39\ 0x3A$ (= строкой "9:"). Второй байт преобразован именно в $0x3A$, а не в $0x41$ – символ "A"

Инициализируется: значением *OPOS_BC_NONE* после вызова метода *open*

Возвращаемое значение: устанавливается свойство *ResultCode* в одно из следующих значений:

- *OPOS_SUCCESS* – свойство установлено успешно;
- *OPOS_E_ILLEGAL* – неверное значение.

ChangeDue

Синтаксис: *STRING ChangeDue* {чтение/запись, доступ после *open*}

Замечания: содержит текст, печатаемый в строке со сдачей, при вызове метода *printRecTotal*

Свойство действительно если *CapChangeDue* равно *TRUE*

Инициализируется: пустой строкой после вызова метода *open*

Смотри также: метод *printRecTotal*, свойство *CapChangeDue*

CheckHealthText

Синтаксис: *STRING CheckHealthText* {только чтение, доступ после *open*}

Замечания: содержит результат последнего вызова метода *CheckHealth*. До первого вызова метода *CheckHealth* значение свойства не инициализировано.

CheckTotal

Синтаксис: *BOOL CheckTotal* {чтение/запись, доступ после *open-claim-enable*}

Замечания: если *TRUE*, то производится автоматическое сравнение итога, посчитанного в устройстве и итога, посчитанного приложением. Если *FALSE*, то автоматическое сравнение не производится.

Свойство действительно если *CapCheckTotal* равно *TRUE*

Инициализируется: значением *TRUE* при вызове метода *open-claim-enable*

Смотри также: свойство *CapCheckTotal*

Claimed

Синтаксис: *BOOL Claimed* {только чтение, доступ после *open*}

Замечания: если *TRUE*, то устройство захвачено для эксклюзивного доступа. Если *FALSE*, то устройство отпущено, и другие приложения могут использовать его.

Инициализируется: значением *FALSE* после вызова метода *open*

ContractorID

Синтаксис: *LONG ContractorID* {чтение/запись, доступ после *open-claim-enable*}

Замечания: идентификатор владельца, который в данный момент использует устройство.

Значения:

- *FPTR_CID_FIRST* – первый владелец;
- *FPTR_CID_SECOND* – второй владелец;
- *FPTR_CID_SINGLE* – один владелец.

Инициализируется: значением *FPTR_CID_SINGLE* и актуально пока *DeviceEnabled* равно *TRUE*

Свойство действительно только если *CapMultiContractor* равно *TRUE*

Смотри также: свойство *CapMultiContractor*

ControlObjectDescription

Синтаксис: *STRING ControlObjectDescription* {только чтение, доступно всегда}

Замечания: строка идентифицирует *Control Object* и компанию, которая его произвела.

Это свойство всегда доступно для чтения.

ControlObjectVersion

Синтаксис: *LONG ControlObjectVersion* {только чтение, доступно всегда}

Замечания: версия *Control Object*.

Это свойство всегда доступно для чтения.

CountryCode

Синтаксис: *LONG CountryCode* {только чтение, доступ после *open-claim*}

Замечания: содержит идентификатор страны, поддерживаемой устройством. Идентификаторы могут быть перечислены через логическое *ИЛИ*:

Значения:

- | | |
|---------------------------------|---|
| • <i>FPTR_CC_BRAZIL</i> | – уст-во соответствует Бразильским требованиям; |
| • <i>FPTR_CC_GREECE</i> | – уст-во соответствует Греческим требованиям; |
| • <i>FPTR_CC_HUNGARY</i> | – уст-во соответствует Венгерским требованиям; |
| • <i>FPTR_CC_ITALY</i> | – уст-во соответствует Итальянским требованиям; |
| • <i>FPTR_CC_POLAND</i> | – уст-во соответствует Польским требованиям; |
| • <i>FPTR_CC_TURKEY</i> | – уст-во соответствует Турецким требованиям; |
| • <i>FPTR_CC_RUSSIA</i> | – уст-во соответствует Российским требованиям; |
| • <i>FPTR_CC_BULGARIA</i> | – уст-во соответствует Болгарским требованиям; |
| • <i>FPTR_CC_ROMANIA</i> | – уст-во соответствует Румынским требованиям; |
| • <i>FPTR_CC_CZECH_REPUBLIC</i> | – уст-во соответствует Чешским требованиям; |
| • <i>FPTR_CC_UKRAINE</i> | – уст-во соответствует Украинским требованиям; |
| • <i>FPTR_CC_OTHER</i> | – неизвестные или новые требования. |

CoverOpen

Синтаксис: *BOOL CoverOpen* {только чтение, доступ после *open-claim-enable*}

Замечания: если *TRUE*, тогда крышка принтера открыта.

Если *CapCoverSensor* равно *FALSE*, тогда устройство не имеет датчика открытия крышки и значение этого свойства всегда равно *FALSE*.

Значение свойства актуально только когда *DeviceEnabled* равно *TRUE*

DateType

Синтаксис: *LONG DateType* {чтение/запись, доступ после *open-claim-enable*}

Замечания: определяет тип даты, которая будет запрошена при вызове метода *getDate*

Значения:

- *FPTR_DT_CONF* – дата конфигурации;
- *FPTR_DT_EOD* – дата закрытия последней смены;
- *FPTR_DT_RESET* – дата последнего сброса;
- *FPTR_DT_RTC* – значение часов в устройстве;
- *FPTR_DT_VAT* – дата последнего изменения налоговых ставок;
- *FPTR_DT_START* – дата первого документа в смене.

Смотри также: метод *getDate*

DayOpened

Синтаксис: *BOOL DayOpened* {только чтение, доступ после *open-claim-enable*}

Замечания: если *TRUE*, то смена открыта, иначе смена закрыта.

DescriptionLength

Синтаксис: *LONG DescriptionLength* {только чтение, доступ после *open-claim*}

Замечания: содержит длину строки, которую можно передавать в качестве параметра *description* в различных функциях.

Инициализируется: методом *open-claim*

Смотри также: метод *getData*

DeviceDescription

Синтаксис: *STRING DeviceDescription*

Замечания: строка с описанием устройства.

Инициализируется: методом *open*

DeviceEnabled

Синтаксис: *BOOL DeviceEnabled* {чтение/запись, доступ после *open* и *claim*}

Замечания: когда равно *TRUE*, разрешается работа с устройством. Иначе работа с устройством запрещена. Изменение этого свойства физически не отключает устройство, только программно.

Устройства серии MSTAR являются составными устройствами. Для обеспечения корректной работы модулей, входящих в их состав, может понадобиться установка *DeviceEnabled* в значение *FALSE*. По этому, рекомендуется в режиме ожидания устанавливать свойство *DeviceEnabled* в значение *FALSE*

Инициализируется: значением *FALSE* после вызова метода *open*

Возвращаемое значение: устанавливается свойство *ResultCode* в одно из следующих значений:

- *OPOS_SUCCESS* – свойство установлено удачно;
- *OPOS_E_NOTCLAIMED* – устройство должно быть сначала захвачено.

DeviceName

Синтаксис: *STRING DeviceName*

Замечания: строка с именем устройства.

Инициализируется: методом *open*

DuplicateReceipt

Синтаксис: *BOOL DuplicateReceipt* {чтение/запись, доступ после *open*}

Замечания: если *TRUE*, то можно распечатать копию чека с помощью метода *printDuplicateReceipt*

Свойство действительно если *CapDuplicateReceipt* равно *TRUE*

Инициализируется: значением *FALSE* при вызове метода *open*

ErrorLevel

Синтаксис: *LONG ErrorLevel* {только чтение, доступ после *open*}

Замечания: содержит тип возникшей ошибки.

Значения:

- *FPTR_EL_NONE* – ошибки нет;
- *FPTR_EL_RECOVERABLE* – исправимая ошибка (например: нет бумаги);
- *FPTR_EL_FATAL* – не исправимая ошибка;
- *FPTR_EL_BLOCKED* – аппарат блокирован. Требуется обратиться в тех. сервис.

ErrorOutID

Синтаксис: *LONG ErrorOutID* {только чтение, доступ после *open*}

Замечания: содержит идентификатор события *ErrorEvent*, когда используется асинхронная печать.

Свойство устанавливается перед тем, как будет доставлен *ErrorEvent*

ErrorState

Синтаксис: *LONG ErrorState* {только чтение, доступ после *open*}

Замечания: содержит текущее состояние устройства, когда доставлено *ErrorEvent* в асинхронном режиме.

Свойство устанавливается перед тем, как будет доставлен *ErrorEvent*

Смотри также: свойство *PrinterState*

ErrorStation

Синтаксис: *LONG ErrorStation* {только чтение, доступ после *open*}

Замечания: содержит идентификатор станции, на которой произошла ошибка.

Значения:

- *FPTR_S_JOURNAL;*
- *FPTR_S_RECEIPT;*
- *FPTR_S_SLIP;*
- *FPTR_S_JOURNAL_RECEIPT;*
- *FPTR_S_JOURNAL_SLIP;*
- *FPTR_S_RECEIPT_SLIP.*

Свойство действительно если *ErrorLevel* не равно *FPTR_EL_NONE*

Свойство устанавливается перед тем, как будет доставлен *ErrorEvent*

ErrorString

Синтаксис: *STRING ErrorString* {только чтение, доступ после *open*}

Замечания: содержит описание возникшей ошибки.

FiscalReceiptStation

Синтаксис: *LONG FiscalReceiptStation* {чтение/запись, доступ после *open-claim-enable*}

Замечания: определяет станцию на которой будут печататься чеки.

Допустимые значения:

- *FPTR_RS_RECEIPT* – чековая станция;
- *FPTR_RS_SLIP* – станция подкладной печати.

Свойство действительно если *CapFiscalReceiptStation* равно *TRUE*

Инициализируется: значением *FPTR_RS_RECEIPT* и актуально пока *DeviceEnabled* равно *TRUE*

Смотри также: метод *beginFiscalReceipt*, свойство *CapFiscalReceiptStation*

FiscalReceiptType

Синтаксис: *LONG FiscalReceiptType* {чтение/запись, доступ после *open-claim-enable*}

Замечания: определяет тип следующего чека.

Допустимые значения:

Таблица 3. Допустимые значения FiscalReceiptType для различных устройств.

Поддержка устройством						Допустимое значение	Описание
MSTAR- TK	MSTAR- TUP-K	MSTAR- Kiosk	МЭР MSTAR	MSPrint- 80K	МЭР MSPrint-80		
+	—	+	+	+	+	<i>FPTR_RT_CASH_IN</i>	внесение
+	+	+	+	+	+	<i>FPTR_RT_CASH_OUT</i>	инкассация (выплата)
—	—	—	—	—	—	<i>FPTR_RT_GENERIC</i>	чек общего вида
+	+	+	+	+	+	<i>FPTR_RT_SALES</i>	продажа
+	+	+	+	+	+	<i>FPTR_RT_SERVICE</i>	услуги
—	—	—	—	—	—	<i>FPTR_RT_SIMPLE_ INVOICE</i>	упрощенный счет
+	—	—	+	+	+	<i>FPTR_RT_REFUND</i>	возврат

Свойство действительно если *CapFiscalReceiptType* равно *TRUE*

Смотри также: метод *beginFiscalReceipt*, свойство *CapFiscalReceiptType*

FlagWhenIdle

Синтаксис: *BOOL FlagWhenIdle* {чтение/запись, доступ после *open*}

Замечания: если *TRUE*, событие *StatusUpdateEvent* будет отправлено, когда устройство вернется в *idle state*

Это свойство автоматически сбросится в *FALSE*, когда событие будет доставлено.

Инициализируется: значением *FALSE* при вызове метода *open*

FreezeEvents

Синтаксис: *BOOL FreezeEvents* {чтение/запись, доступ после *open* и *claim*}

Замечания: если *TRUE*, доставка событий от устройства к приложению «замораживается». Если *FALSE*, то события будут доставляться приложению. Приложение может приостановить доставку событий, например, в критической секции кода, которую нельзя прерывать.

Инициализируется: значением *FALSE* после вызова метода *open*

JrnEmpty

Синтаксис: *BOOL JrnEmpty* {только чтение, доступ после *open-claim-enable*}

Замечания: если *TRUE*, то в принтере контрольной ленты отсутствует бумага. Если *CapJrnEmptySensor* равно *FALSE*, тогда значение этого свойства всегда равно *FALSE*

Значение свойства актуально только когда *DeviceEnabled* равно *TRUE*

Смотри также: свойство *JrnNearEnd*

JrnNearEnd

Синтаксис: `BOOL JrnNearEnd` {только чтение, доступ после `open-claim-enable`}

Замечания: если `TRUE`, то в принтере контрольной ленты заканчивается бумага. Если `CapJrnNearEndSensor` равно `FALSE`, тогда значение этого свойства всегда равно `FALSE`

Значение свойства актуально только когда `DeviceEnabled` равно `TRUE`

Смотри также: свойство `JrnEmpty`

MessageLength

Синтаксис: `LONG MessageLength` {только чтение, доступ после `open-claim`}

Замечания: содержит максимальную длину строки, которую можно напечатать с помощью метода `printRecMessage`

Инициализируется: методом `open-claim`

MessageType

Синтаксис: `LONG MessageType` {чтение/запись, доступ после `open-claim-enable`}

Замечания: определяет тип сообщения, которое будет напечатано методом `printRecMessage`

Значения:

- | | |
|---|--|
| • <code>FPTR_MT_ADVANCE</code> | • <code>FPTR_MT_GIVEN_DISCOUNT</code> |
| • <code>FPTR_MT_ADVANCE_PAID</code> | • <code>FPTR_MT_LOCAL_CREDIT</code> |
| • <code>FPTR_MT_AMOUNT_TO_BE_PAID</code> | • <code>FPTR_MT_MILEAGE_KM</code> |
| • <code>FPTR_MT_AMOUNT_TO_BE_PAID_BACK</code> | • <code>FPTR_MT_NOTE</code> |
| • <code>FPTR_MT_CARD</code> | • <code>FPTR_MT_PAID</code> |
| • <code>FPTR_MT_CARD_NUMBER</code> | • <code>FPTR_MT_PAY_IN</code> |
| • <code>FPTR_MT_CARD_TYPE</code> | • <code>FPTR_MT_POINT_GRANTED</code> |
| • <code>FPTR_MT_CASH</code> | • <code>FPTR_MT_POINTS_BONUS</code> |
| • <code>FPTR_MT_CASHIER</code> | • <code>FPTR_MT_POINTS_RECEIPT</code> |
| • <code>FPTR_MT_CASH_REGISTER_NUMBER</code> | • <code>FPTR_MT_POINTS_TOTAL</code> |
| • <code>FPTR_MT_CHANGE</code> | • <code>FPTR_MT_PROFITED</code> |
| • <code>FPTR_MT_CHEQUE</code> | • <code>FPTR_MT_RATE</code> |
| • <code>FPTR_MT_CLIENT_NUMBER</code> | • <code>FPTR_MT_REGISTER_NUMB</code> |
| • <code>FPTR_MT_CLIENT_SIGNATURE</code> | • <code>FPTR_MT_SHIFT_NUMBER</code> |
| • <code>FPTR_MT_COUNTER_STATE</code> | • <code>FPTR_MT_STATE_OF_AN_ACCOUNT</code> |
| • <code>FPTR_MT_CREDIT_CARD</code> | • <code>FPTR_MT_SUBSCRIPTION</code> |
| • <code>FPTR_MT_CURRENCY</code> | • <code>FPTR_MT_TABLE</code> |
| • <code>FPTR_MT_CURRENCY_VALUE</code> | • <code>FPTR_MT_THANK_YOU_FOR_LOYALTY</code> |
| • <code>FPTR_MT_DEPOSIT</code> | • <code>FPTR_MT_TRANSACTION_NUMB</code> |
| • <code>FPTR_MT_DEPOSIT_RETURNED</code> | • <code>FPTR_MT_VALID_TO</code> |
| • <code>FPTR_MT_DOT_LINE</code> | • <code>FPTR_MT_VOUCHER</code> |
| • <code>FPTR_MT_DRIVER_NUMB</code> | • <code>FPTR_MT_VOUCHER_PAID</code> |
| • <code>FPTR_MT_EMPTY_LINE</code> | • <code>FPTR_MT_VOUCHER_VALUE</code> |
| • <code>FPTR_MT_FREE_TEXT</code> | • <code>FPTR_MT_WITH_DISCOUNT</code> |
| • <code>FPTR_MT_FREE_TEXT_WITH_DAY_LIMIT</code> | • <code>FPTR_MT_WITHOUT_UPLIFT</code> |

Инициализируется: значением `FPTR_MT_FREE_TEXT` при вызове метода `open`

Смотри также: метод `printRecMessage`

NumHeaderLines

Синтаксис: *LONG NumHeaderLines* {только чтение, доступ после *open-claim*}

Замечания: содержит число строк в заголовке документа. Каждую строку можно установить с помощью метода *setHeaderLine*

Инициализируется: методом *open-claim*

NumTrailerLines

Синтаксис: *LONG NumTrailerLines* {только чтение, доступ после *open-claim*}

Замечания: содержит число строк, которые можно напечатать в конце каждого документа. Каждую строку можно установить с помощью метода *setTrailerLine*

Инициализируется: методом *open-claim*

NumVatRates

Синтаксис: *LONG NumVatRates* {только чтение, доступ после *open-claim*}

Замечания: содержит количество число налоговых ставок.

Инициализируется: методом *open-claim*

OpenResult

Синтаксис: *LONG OpenResult* {только чтение, доступ после *open*}

Замечания: содержит детальный результат вызова метода *open*

Значения:

- *OPOS_SUCCESS* – прошло успешно;
- *OPOS_OR_ALREADYOPEN* – уже открыто;
- *OPOS_OR_REGBADNAME* – в реестре нет указанного в методе имени;
- *OPOS_OR_REGPROGID* – не получается прочитать ключ реестра с именем устройства;
- *OPOS_OR_CREATE* – не получается создать экземпляр *Service Object*;
- *OPOS_OR_BADIF* – *Service Object* не поддерживает все необходимые методы;
- *OPOS_OR_FAILEDOPEN* – *Service Object* вернул неизвестную ошибку при открытии;
- *OPOS_OR_BADVERSION* – не верная версия *Service Object*;
- *OPOS_OR_NOPORT* – не получилось открыть порт устройства;
- *OPOS_OR_CONFIG* – ошибка конфигурации;
- *OPOS_OR_SPECIFIC* – специфическая ошибка *Service Object*.

Инициализируется: методом *open*

OutputID

Синтаксис: *LONG OutputID* {только чтение, доступ после *open*}

Замечания: содержит идентификатор последнего успешного вывода.

Когда какой-либо метод иницирует асинхронный вывод, *Control* присваивает этому запросу идентификатор. Когда асинхронный вывод окончен, *Control* отправляет приложению событие *OutputCompleteEvent*, содержащее этот идентификатор.

PostLine

Синтаксис: *STRING PostLine* {чтение/запись, доступ после *open-claim-enable*}

Замечания: содержит текст, который может быть напечатан после проведения операции одним из методов *printRec....* Например, после операции продажи. Свойство действительно, если *CapPostPreLine* равно *TRUE*

Инициализируется: пустой строкой и сбрасывается в пустую строку после того, как будет напечатано.

Строка может содержать команды принтера.

Смотри также: метод *printRecSubtotal*, метод *printRecTotal*, свойство *CapPostPreLine*

PowerNotify

Синтаксис: *LONG PowerNotify* {только чтение, доступ после *open*}

Замечания: содержит тип информирования о питании, выбранный приложением.

Значения:

- *OPOS_PN_DISABLED* – события не доставляются;
- *OPOS_PN_ENABLED* – события доставляются, когда свойство *DeviceEnabled* установлено в *TRUE*.

PowerNotify может быть установлено только тогда, когда свойство *DeviceEnabled* установлено в *FALSE*

Инициализируется: значением *OPOS_PN_DISABLED* после вызова метода *open*

PowerState

Синтаксис: *LONG PowerState* {только чтение, доступ после *open*}

Замечания: содержит текущее состояние питания, если оно может быть определено.

Значения:

- *OPOS_PS_UNKNOWN* – невозможно определить состояние питания устройства по одной из следующих причин:
 - *CapPowerReporting* = *OPOS_PR_NONE*
 - *PowerNotify* = *OPOS_PN_DISABLED*
 - *DeviceEnabled* = *FALSE*
- *OPOS_PS_ONLINE* – устройство включено и готово к использованию;
- *OPOS_PS_OFF* – устройство выключено или не присоединено к компьютеру;
- *OPOS_PS_OFFLINE* – устройство включено, но не готово к использованию;
- *OPOS_PS_OFF_OFFLINE* – устройство или выключено или включено, но не готово к использованию.

Инициализируется: значением *OPOS_PS_UNKNOWN* после вызова метода *open*

PredefinedPaymentLines

Синтаксис: *STRING PredefinedPaymentLines* {только чтение, доступ после *open*}

Замечания: содержит список индексов платежей, которые используются в методе *printRecTotal* в параметре *description*

Инициализируется: методом *open*

PreLine

Синтаксис: *STRING PreLine* {чтение/запись, доступ после *open-claim-enable*}

Замечания: содержит текст, который может быть напечатан до проведения операции одним из методов *printRec...*. Например, до операции продажи. Свойство действительно, если *CapPostPreLine* равно *TRUE*. Строка может содержать команды принтера.

Инициализируется: пустой строкой и сбрасывается в пустую строку после того, как будет напечатано.

Смотри также: метод *printRecItem*, метод *printRecItemAdjustment*, метод *printRecRefund*, метод *printRecSubtotalAdjustment*, СВОЙСТВО *CapPostPreLine*

PrinterState

Синтаксис: *LONG PrinterState* {только чтение, доступ после *open*}

Замечания: содержит текущее состояние устройства.

Значения:

- *FPTR_PS_MONITOR* – уст-во находится в ждущем режиме.
В этом режиме можно использовать *begin...* и *set...*;
- *FPTR_PS_FISCAL_RECEIPT* – уст-во в режиме фискального документа.
В этом режиме можно использовать методы *printRec...*;
- *FPTR_PS_FISCAL_RECEIPT_TOTAL* – уст-во ожидает внесения оплаты.
В этом режиме можно использовать методы *printRecTotal* или *printRecNotPaid*;
- *FPTR_PS_FISCAL_RECEIPT_ENDING* – уст-во приняла оплату и ждет завершения документа.
В этом режиме можно использовать методы *printRecMessage* или *endFiscalReceipt*;
- *FPTR_PS_FISCAL_DOCUMENT* – уст-во в процессе печати подкладного документа.
В этом режиме можно использовать методы *printFiscalDocumentLine* или *endFiscalDocument*;
- *FPTR_PS_FIXED_OUTPUT* – уст-во печатает фиксированный текст на одной из своих станций. В этом режиме можно использовать методы *printFixedOutput* или *endFixedOutput*;
- *FPTR_PS_ITEM_LIST* – уст-во работает с базой товаров.
В этом режиме можно использовать методы *verifyItem* или *endItemList*;
- *FPTR_PS_NONFISCAL* – уст-во находится в режиме печати не фискального документа. В этом режиме можно использовать методы *printNormal* или *endNonFiscal*;
- *FPTR_PS_LOCKED* – уст-во заблокирована.
Требуется обратиться к техническому специалисту;
- *FPTR_PS_REPORT* – уст-во в режиме печати фискальных отчетов.
В этом режиме нельзя вызывать никаких методов.

Также есть методы, которые можно вызвать всегда, кроме режима *FPTR_PS_LOCKED*. Это *beginInsertion*, *endInsertion*, *beginRemoval*, *endRemoval*, *getDate*, *getData*, *getTotalizer*, *getVatEntry*, *resetPrinter* и *clearOutput*

QuantityDecimalPlaces

Синтаксис: *LONG QuantityDecimalPlaces* {только чтение, доступ после *open-claim*}

Замечания: содержит количество десятичных знаков в дробной части параметра *Quantity*

QuantityLength

Синтаксис: *LONG QuantityLength* {только чтение, доступ после *open-claim*}

Замечания: количество десятичных знаков в целой и дробной частях параметра *Quantity*

RecEmpty

Синтаксис: *BOOL RecEmpty* {только чтение, доступ после *open-claim-enable*}

Замечания: если *TRUE*, то в чековом принтере нет бумаги.

Если *CapRecEmptySensor* равно *FALSE*, тогда это свойство всегда равно *FALSE*
Значение свойства актуально только когда *DeviceEnabled* равно *TRUE*

Смотри также: свойство *RecNearEnd*

RecNearEnd

Синтаксис: *BOOL RecNearEnd* {только чтение, доступ после *open-claim-enable*}

Замечания: если *TRUE*, то в чековом принтере мало бумаги. Если *CapRecNearEndSensor* равно *FALSE*, тогда это свойство всегда равно *FALSE*.

Значение свойства актуально только когда *DeviceEnabled* равно *TRUE*

Смотри также: свойство *RecEmpty*

RemainingFiscalMemory

Синтаксис: *LONG RemainingFiscalMemory* {только чтение, доступ после *open-claim-enable*}

Замечания: содержит количество оставшихся закрытых смен.

Значение свойства актуально только когда *DeviceEnabled* равно *TRUE* и обновляется после вызова метода *printZReport*

Смотри также: свойство *CapRemainingFiscalMemory*

ReservedWord

Синтаксис: *STRING ReservedWord* {только чтение, доступ после *open*}

Замечания: содержит строку, печатаемую во время вызова метода *printRecTotal*

Свойство действительно если *CapReservedWord* равно *TRUE*

Инициализируется: методом *open*

ResultCode

Синтаксис: *LONG ResultCode* {только чтение, доступ после *open*}

Замечания: это свойство обновляется после вызова каждого метода и содержит результат его выполнения. Оно всегда доступно для чтения. До первого вызова метода *open* содержит значение *OPOS_E_CLOSED*

Значения:

- *OPOS_SUCCESS* – ошибки нет;
- *OPOS_E_CLOSED* – устройство не открыто;
- *OPOS_E_CLAIMED* – устройство захвачено другим приложением;
- *OPOS_E_NOTCLAIMED* – устройство не захвачено для полного доступа;
- *OPOS_E_NOSERVICE* – не удается связаться с *Service Object*;
- *OPOS_E_DISABLE* – свойство *DeviceEnabled* = *FALSE*;
- *OPOS_E_ILLEGAL* – не поддерживаемый метод или неверный параметр;
- *OPOS_E_NOHARDWARE* – устройство не подключено к системе или на нем отсутствует питание;
- *OPOS_E_OFFLINE* – устройство выключено;
- *OPOS_E_NOEXIST* – отсутствует какой-либо параметр;
- *OPOS_E_EXISTS* – такой параметр уже присутствует;
- *OPOS_E_FAILURE* – не получается выполнить команду, хотя устройство подключено и включено;
- *OPOS_E_TIMEOUT* – таймаут;
- *OPOS_E_BUSY* – устройство занято;
- *OPOS_E_EXTENDED* – расширенная ошибка. Подробнее смотреть в свойстве *ResultCodeExtended*;
- *OPOS_E_DEPRECATED* – запрошенная операция устарела и больше не используется.

ResultCodeExtended

Синтаксис: *LONG ResultCodeExtended* {только чтение, доступ после *open*}

Замечания: когда свойство *ResultCode* имеет значение *OPOS_E_EXTENDED*, это свойство содержит расширенную ошибку.

ServiceObjectDescription

Синтаксис: *STRING ServiceObjectDescription* {только чтение, доступно после *open*}

Замечания: строка содержит описание *Service Object*.

Инициализируется: методом *open*

ServiceObjectVersion

Синтаксис: *LONG ServiceObjectVersion* {только чтение, доступно после *open*}

Замечания: содержит версию *Service Object*

Инициализируется: методом *open*

SlipSelection

Синтаксис: *LONG SlipSelection* {чтение/запись, доступ после *open-claim-enable*}

Замечания: определяет тип документа, который будет печататься на принтере подкладной печати.

Значения:

- *FPTR_SS_FULL_LENGTH*
- *FPTR_SS_VALIDATION*

Инициализируется: значением *FPTR_SS_FULL_LENGTH* при вызове метода *claim*

SlpEmpty

Синтаксис: *BOOL SlpEmpty* {только чтение, доступ после *open-claim-enable*}

Замечания: если *TRUE*, отсутствует бумага в принтере подкладного документа.

Если *CapSlpEmptySensor* равно *FALSE*, тогда значение этого свойства всегда равно *FALSE*

Значение свойства актуально только когда *DeviceEnabled* равно *TRUE*

Смотри также: свойство *SlpNearEnd*

SlpNearEnd

Синтаксис: *BOOL SlpNearEnd* {только чтение, доступ после *open-claim-enable*}

Замечания: если *TRUE*, мало бумаги в принтере подкладного документа.

Если *CapSlpNearEndSensor* равно *FALSE*, тогда значение этого свойства всегда равно *FALSE*

Значение свойства актуально только когда *DeviceEnabled* равно *TRUE*

Смотри также: свойство *SlpEmpty*

State

Синтаксис: *LONG State* {только чтение, доступно всегда}

Замечания: содержит текущее состояние *Control'a*.

Значения:

- *OPOS_S_CLOSED* – *Control* закрыт;
- *OPOS_S_IDLE* – *Control* открыт и не занят;
- *OPOS_S_BUSY* – *Control* открыт но занят;
- *OPOS_S_ERROR* – произошла ошибка;

Это свойство всегда доступно для чтения.

TotalizerType

Синтаксис: *LONG TotalizerType* {чтение/запись, доступ после *open-claim-enable*}

Замечания: определяет счетчик, значение которого будет запрошено методом *getTotalizer*

Значения:

- *FPTR_TT_DAY* – по смене;
- *FPTR_TT_RECEIPT* – по чеку;
- *FPTR_TT_GRAND* – общие.

Свойство действительно если *CapTotalizerType* равно *TRUE*

Инициализируется: значением *FPTR_TT_DAY*

Смотри также: метод *getTotalizer*, свойство *CapTotalizerType*

TrainingModeActive

Синтаксис: *BOOL TrainingModeActive* {чтение/запись, доступ после *open-claim-enable*}

Замечания: если *TRUE*, то устройство находится в режиме обучения.

Методы

Перечень методов OPOS Fiscal Printer

Перечень методов OPOS Fiscal Printer приведен в таблице. (Таблица 3).

Таблица 4. Перечень методов OPOS Fiscal Printer.

Название	Можно использовать после
BeginFiscalDocument	Open, Claim & Enable
BeginFiscalReceipt	Open, Claim & Enable
BeginFixedOutput	Open, Claim & Enable
BeginInsertion	Open, Claim & Enable
BeginItemList	Open, Claim & Enable
BeginNonFiscal	Open, Claim & Enable
BeginRemoval	Open, Claim & Enable
BeginTraining	Open, Claim & Enable
CheckHealth	Open, Claim & Enable
ClaimDevice	Open
ClearError	Open, Claim & Enable
ClearOutput	Open & Claim
Close	Open
CompareFirmwareVersion	Open, Claim & Enable
DirectIO	Open
EndFiscalDocument	Open, Claim & Enable
EndFiscalReceipt	Open, Claim & Enable
EndFixedOutput	Open, Claim & Enable
EndInsertion	Open, Claim & Enable
EndItemList	Open, Claim & Enable
EndNonFiscal	Open, Claim & Enable
EndRemoval	Open, Claim & Enable
EndTraining	Open, Claim & Enable
GetData	Open, Claim & Enable
GetDate	Open, Claim & Enable
GetTotalizer	Open, Claim & Enable
GetVatEntry	Open, Claim & Enable
Open	
PrintDuplicateReceipt	Open, Claim & Enable
PrintFiscalDocumentLine	Open, Claim & Enable
PrintFixedOutput	Open, Claim & Enable
PrintNormal	Open, Claim & Enable
PrintPeriodicTotalsReport	Open, Claim & Enable
PrintPowerLossReport	Open, Claim & Enable
PrintRecCash	Open, Claim & Enable
PrintRecItem	Open, Claim & Enable
PrintRecItemAdjustment	Open, Claim & Enable
PrintRecItemAdjustmentVoid	Open, Claim & Enable
PrintRecItemFuel	Open, Claim & Enable
PrintRecItemFuelVoid	Open, Claim & Enable
PrintRecItemVoid	Open, Claim & Enable
PrintRecMessage	Open, Claim & Enable
PrintRecNotPaid	Open, Claim & Enable

Название	Можно использовать после
PrintRecPackageAdjustment	Open, Claim & Enable
PrintRecPackageAdjustVoid	Open, Claim & Enable
PrintRecRefund	Open, Claim & Enable
PrintRecRefundVoid	Open, Claim & Enable
PrintRecSubtotal	Open, Claim & Enable
PrintRecSubtotalAdjustment	Open, Claim & Enable
PrintRecSubtotalAdjustVoid	Open, Claim & Enable
PrintRecTaxID	Open, Claim & Enable
PrintRecTotal	Open, Claim & Enable
PrintRecVoid	Open, Claim & Enable
PrintRecVoidItem	Open, Claim & Enable
PrintReport	Open, Claim & Enable
PrintXReport	Open, Claim & Enable
PrintZReport	Open, Claim & Enable
ReleaseDevice	Open & Claim
ResetPrinter	Open, Claim & Enable
ResetStatistics	Open, Claim & Enable
RetrieveStatistics	Open, Claim & Enable
SetCurrency	Open, Claim & Enable
SetDate	Open, Claim & Enable
SetHeaderLine	Open, Claim & Enable
SetPOSID	Open, Claim & Enable
SetStoreFiscalID	Open, Claim & Enable
SetTrailerLine	Open, Claim & Enable
SetVatTable	Open, Claim & Enable
SetVatValue	Open, Claim & Enable
UpdateFirmwsre	Open, Claim & Enable
UpdateStatistics	Open, Claim & Enable
VerifyItem	Open, Claim & Enable

Описание методов OPOS Fiscal Printer

Open

Синтаксис: *LONG Open (BSTR DeviceName)*

Замечания: метод загружает OPOS драйвер устройства, проверяет правильность его установки и настроек. Никакого обмена с устройством при этом не производится.

Параметры:

- *DeviceName* – название устройства, например "MStarTK".

Пример:

```
Open ("MstarTK")
```

Close

Синтаксис: *LONG Close ()*

Замечания: отключает OPOS драйвер от прикладной программы.

ClaimDevice

Синтаксис: *LONG ClaimDevice (LONG Timeout)*

Замечания: попытка захватить устройство для эксклюзивного использования.

Параметры:

- *Timeout* – значение таймаута при попытке захватить устройство.

Пример:

```
ClaimDevice(0)
```

ReleaseDevice

Синтаксис: *LONG ReleaseDevice ()*

Замечания: отпустить устройство.

CheckHealth

Синтаксис: *LONG CheckHealth (LONG Level)*

Замечания: метод проводит тестирование состояния устройства, затем сохраняет заключение в свойстве *CheckHealthText*

Параметры:

- *Level* – режим тестирования:
 - *OPOS_CH_INTERNAL* – тестирование без изменения состояния устройств;
 - *OPOS_CH_EXTERNAL* – тестирование с печатью чека автотестирования;
 - *OPOS_CH_INTERACTIVE* – не используется.

ClearOutput

Синтаксис: *LONG ClearOutput ()*

Замечания: метод очищает очередь асинхронного вывода.

DirectIO

Команды DirectIO разделяются на 2 части:

- *общие команды*, которые выполняются на всех версиях ПО. Коды этих команд лежат в диапазоне `0x00–0x05` включительно. Не требуют передачи дополнительных параметров.
- *специфические команды*, которые могут меняться от версии к версии. Коды этих команд всегда больше `0x05`. Команды, лежащие в диапазоне `0x06–0xFF`, не требуют передачи дополнительных параметров. Команды с кодами больше `0x100` являются *специфическими командами с параметрами* и требуют передачи дополнительных параметров перед выполнением.

В текущей версии драйвера реализованы следующие *общие команды*:

- 1) `0x00` – получить список поддерживаемых команд
- 2) `0x01` – тест
- 3) `0x02` – получить описание для специфической команды
- 4) `0x03` – получить описание кода ошибки
- 5) `0x04` – записать параметр специфической команды с параметрами
- 6) `0x05` – выполнить специфическую команду с параметрами

С помощью набора общих команд можно получить информацию обо всех поддерживаемых в текущей версии ПО командах, и выполнить любую из них.

Получить список команд

Параметр	In	Out	Примечание
<i>Command</i>	0x00000000	–	
<i>pData</i>	–	–	Любое значение
<i>pString</i>	–	Строка со списком команд.	

Получает список поддерживаемых команд. Ответ выдает в виде XML строки.

Корневым тегом является *ArrayOfCmdInfo*. Описание каждой команды находится в теге *CmdInfo*. Внутри тега *Name* находится название команды, внутри тега *Code* код команды.

Пример ответа:

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfCmdInfo
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <CmdInfo>
    <Name>Команда DirectIO по-умолчанию</Name>
    <Code>0</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Считать данные фискального накопителя</Name>
    <Code>6</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Считать программу фискального накопителя</Name>
    <Code>7</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Считать дескриптор фискального накопителя</Name>
    <Code>8</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Считать программу процессора</Name>
    <Code>9</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Программирование серийного номера</Name>
    <Code>257</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Установка даты ввода режима EBPO</Name>
    <Code>259</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Фискализация/Перерегистрация</Name>
    <Code>258</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Инициализация Фискального накопителя</Name>
    <Code>260</Code>
  </CmdInfo>
  <CmdInfo>
    <Name>Установка котировок валют</Name>
    <Code>261</Code></CmdInfo>
</ArrayOfCmdInfo>
```

Тест

Параметр	In	Out	Примечание
<i>Command</i>	0x00000001		
<i>pData</i>	Тестовое значение	Тестовое значение	Любое значение
<i>pString</i>	Тестовая строка	Тестовая строка	Любое значение

Выводит тестовую строку на печать в отладочной версии.

Получить описание для специфической команды

Параметр	In	Out	Примечание
<i>Command</i>	0x00000002	–	
<i>pData</i>	Код команды	Параметры	
<i>pString</i>	–	Описание параметров	

- *Код команды* содержит код специфической команды, для которой необходимо получить описание.;
- *Параметры* – комбинация через ИЛИ кодов параметров для данной команды. Если запрашивается описание к специфической команде без параметров, то будет возвращен 0. Пример:
Код команды: 0x00000102(фискализация). *Параметры: 0x0000000F*.
 - *0x00000001* – ИНН,
 - *0x00000002* – регистрационный номер,
 - *0x00000004* – пароль налогового инспектора,
 - *0x00000008* – новый пароль НИ.
- *Описание параметров* – строка в формате XML с описанием команды и ее параметров.

Корневой тег – *DirectIOCmd*. Внутри него находятся теги *Cmd*, где указан код команды, *Name*, где указано название команды и *Items*, где описаны параметры команды. Каждый параметр обрамлен тегом *ItemBase*, в котором, в атрибуте *type* находится тип параметра. Тег *Value* хранит значение для инициализации. В теге *Flag* находится код параметра. Тег *Name* хранит название параметра.

Если запрашивается описание специфической команды без параметров, то будет возвращено описание параметров метода *DirectIO* для этой команды в следующем порядке: *Command*, *pData*, *pString*.

В данной реализации драйвера существует 5 типов параметров:

- 1) *ItemString* – Строка
- 2) *ItemLong* – Целое число
- 3) *ItemDateTime* – Дата/Время
- 4) *ItemPassword* – Пароль
- 5) *ItemBool* – Булево значение

Для разных типов параметров возможны дополнительные теги:

- *ReadOnly* – если *true*, то этот параметр только для чтения.
- *Description* – содержит описание параметра
- *MaxLength* – максимальная длина, для параметра типов *ItemString* и *ItemPassword*
- *IsConfirm* – требует ли подтверждения. Для параметра типа *ItemPassword*

Пример XML строки для специфической команды с параметрами:

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

```

xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>258</Cmd>
  <Name>Фискализация/Перерегистрация</Name>
  <Items>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>1</Flag>
      <Name>ИИН</Name>
      <MaxLength>12</MaxLength>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>2</Flag>
      <Name>Регистрационный номер</Name>
      <MaxLength>12</MaxLength>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemPassword">
      <Value xsi:type="xsd:string"/Value>
      <Flag>4</Flag>
      <Name>Пароль НИ</Name>
      <MaxLength>5</MaxLength>
      <IsConfirm>>false</IsConfirm>
    </ItemBase>
    <ItemBase xsi:type="ItemPassword">
      <Value xsi:type="xsd:string"/Value>
      <Flag>8</Flag>
      <Name>Новый пароль НИ</Name>
      <MaxLength>5</MaxLength>
      <IsConfirm>>true</IsConfirm>
    </ItemBase>
  </Items>
</DirectIOCmd>

```

Пример XML строки для специфической команды без параметров:

```

<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>9</Cmd>
  <Name>Считать программу процессора</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">9</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>

```

```

    <ReadOnly>true</ReadOnly>
  </ItemBase><ItemBase xsi:type="ItemPath">
    <Value xsi:type="xsd:string"/>
    <Flag>0</Flag>
    <Name>Сохранить в файл...</Name>
  </ItemBase>
</Items>
</DirectIOCmd>

```

Получить описание кода ошибки

Параметр	In	Out	Примечание
<i>Command</i>	0x00000003	–	
<i>pData</i>	Код ошибки	–	
<i>pString</i>	–	–	

Возвращает текстовое описание для заданного кода ошибки.

Записать параметр специфической команды с параметрами

Параметр	In	Out	Примечание
<i>Command</i>	0x00000004	–	
<i>pData</i>	Флаг параметра	–	
<i>pString</i>	Значение параметра	–	

Флаг параметра должен соответствовать флагу, полученному в результате выполнения команды «Получить список параметров для специфической команды с параметрами».

Выполнить специфическую команду с параметрами

Параметр	In	Out	Примечание
<i>Command</i>	0x00000005	–	
<i>pData</i>	–	–	
<i>pString</i>	–	–	

Перед вызовом данной команды необходимо инициализировать специфическую команду с параметрами и передать все параметры.

Таким образом, выполнение специфическую команду с параметрами сводится к следующей последовательности:

```

DirectIO(cmd_code, 0x0, "");           //Инициализация команды с кодом cmd_code
DirectIO( 4, param_code, "0");         //Запись параметра с кодом param_code
DirectIO( 4, param_code, "значение");  //Запись параметра с кодом param_code
DirectIO( 4, param_code, "10,11");     //Запись параметра с кодом param_code
DirectIO( 5, 0x0, "");                 //Выполнить команду

```

Специфические команды без параметров выполняются следующим образом:

```
DirectIO(cmd_code, int_param, string_param);
```

Где:

- *cmd_code* – код специфической команды без параметров;
- *int_param* – целочисленное значение;
- *string_param* – строковое значение.

Подробнее о специфических командах DirectIO

CompareFirmwareVersion

Синтаксис: `LONG CompareFirmwareVersion (BSTR firmwareFileName, LONG *pResult)`

Замечания: метод не реализован в текущей версии драйвера.

ResetStatistics

Синтаксис: `LONG ResetStatistics (BSTR statisticsBuffer)`

Замечания: метод не реализован в текущей версии драйвера.

RetrieveStatistics

Синтаксис: `LONG RetrieveStatistics (BSTR^ pStatisticsBuffer)`

Замечания: метод не реализован в текущей версии драйвера.

UpdateFirmware

Синтаксис: `LONG UpdateFirmware (BSTR firmwareFileName)`

Замечания: метод не реализован в текущей версии драйвера.

UpdateStatistics

Синтаксис: `LONG UpdateStatistics (BSTR statisticsBuffer)`

Замечания: метод не реализован в текущей версии драйвера.

SetCurrency

Синтаксис: `LONG SetCurrency (LONG NewCurrency)`

Замечания: метод не реализован в текущей версии драйвера.

SetDate

Синтаксис: `LONG SetDate (BSTR Date)`

Замечания: метод изменяет дату и время, хранящееся в устройстве.

Параметры:

Date – дата и время в формате "ддммггггччмм", где

- *дд* – день месяца(01–31);
- *мм* – месяц(01–12);
- *гггг* – год(2005–20xx);
- *чч* – час(0–23);
- *мм* – минута(0–59).

Этот метод можно вызывать только в том случае, если свойство *DayOpened* равно *FALSE*

SetHeaderLine

Синтаксис: `LONG SetHeaderLine (LONG LineNumber ,BSTR Text, BOOL DoubleWidth)`

Замечания: метод изменяет строку заголовка фискального чека.

Параметры:

- *LineNumber* – номер строки заголовка в диапазоне от 1 до *NumHeaderLines*;
- *Text* – текст строки заголовка;
- *DoubleWidth* – печать текста двойной ширины.

Пример:

```
SetHeaderLine(1,"HeaderLine1", TRUE);
```

Этот метод можно вызывать только в том случае, если свойство *DayOpened* равно *FALSE*

SetPOSID

Синтаксис: `LONG SetPOSID (BSTR POSID, BSTR cashierID)`

Замечания: метод регистрирует кассира.

Параметры:

- *POSID* – не используется;
- *CashierID* – номер кассира.

SetStoreFiscalID

Синтаксис: `LONG SetStoreFiscalID (BSTR ID)`

Замечания: устанавливает заводской номер устройства. Этот метод используется однократно при производстве.

Параметры:

- *ID* – новый заводской номер.

SetTrailerLine

Синтаксис: `LONG SetTrailerLine (LONG LineNumber ,BSTR Text, BOOL DoubleWidth)`

Замечания: метод изменяет строку окончания фискального чека.

Параметры:

- *LineNumber* – номер строки заголовка в диапазоне от 1 до *NumTrailerLines*;
- *Text* – текст строки заголовка;
- *DoubleWidth* – печать текста двойной ширины.

Пример:

```
SetTrailerLine(1,"TrailerLine1", TRUE);
```

Этот метод можно вызывать только в том случае, если свойство *DayOpened* равно *FALSE*

SetVatTable

Синтаксис: `LONG SetVatTable ()`

Замечания: метод записывает таблицу значений процентов налоговых ставок, построенную методом `SetVatValue`, в устройстве.

Этот метод можно вызывать только в том случае, если свойство `DayOpened` равно `FALSE`

SetVatValue

Синтаксис: `LONG SetVatValue (LONG VatID, BSTR VatValue)`

Замечания: метод изменяет значения налоговых ставок.

Параметры:

- `VatID` — индекс налоговой ставки;
- `VatValue` — процент налоговой ставки.

Все изменения, внесенные методом `SetVatValue`, кэшируются внутри драйвера. Для сохранения изменений в устройстве используйте метод `SetVatTable`

Этот метод можно вызывать только в том случае, если свойство `DayOpened` равно `FALSE`

Пример:

```
BSTR vat_value=SysAllocString(L"18.5");  
SetVatValue(0,vat_value);
```

BeginFiscalReceipt

Синтаксис: `LONG BeginFiscalReceipt (BOOL PrintHeader)`

Замечания: начало печати фискального чека.

Параметры:

- `PrintHeader` — если `TRUE`, то печатается заголовок чека, если `FALSE`, то не печатается.

Тип чека определяется значением свойства `FiscalReceiptType`

Этот метод можно вызывать только тогда, когда свойство `PrinterState` равно `FPTR_PS_MONITOR`

Если метод успешно выполняется, то `PrinterState` меняется на `FPTR_PS_FISCAL_RECEIPT`

EndFiscalReceipt

Синтаксис: `LONG EndFiscalReceipt (BOOL PrintHeader)`

Замечания: завершение печати фискального чека.

Параметры:

- `PrintHeader` — если `TRUE`, то после окончания текущего чека будет распечатан заголовок следующего.

Тип чека определяется значением свойства `FiscalReceiptType`

Этот метод можно вызывать только в том случае, если свойство `PrinterState` равно `FPTR_PS_FISCAL_RECEIPT_ENDING`

Если метод успешно выполняется, то `PrinterState` становится `FPTR_PS_MONITOR`

PrintDuplicateReceipt

Синтаксис: `LONG PrintDuplicateReceipt ()`

Замечания: печать копии чека.

Этот метод можно вызывать только в том случае, если свойство **PrinterState** равно `FPTR_PS_MONITOR` и **DuplicateReceipt** равно `TRUE`

PrintRecCash

Синтаксис: `LONG PrintRecCash (CURRENCY Amount)`

Замечания: инкассация или внесение денежных средств в устройство.

Параметры:

- *Amount* – вносимая или инкассируемая сумма.

Тип операции определяется свойством **FiscalReceiptType**

Этот метод можно вызывать только в том случае, если свойство **PrinterState** равно `FPTR_PS_FISCAL_RECEIPT` и **FiscalReceiptType** равен `FPTR_RT_CASH_IN` или `FPTR_RT_CASH_OUT`

PrintRecItem

Синтаксис: `LONG PrintRecItem (BSTR Description, CURRENCY Price, LONG Quantity, LONG VatInfo, CURRENCY UnitPrice, BSTR UnitName)`

Замечания: метод осуществляет операцию продажи.

Параметры:

- *Description* – наименование товара;
- *Price* – цена единицы товара; используется, если *UnitPrice* = 0;
- *Quantity* – количество (обратите внимание также на *QuantityLength* и *QuantityDecimalPlaces*). Значение 0 эквивалентно 1000;
- *VatInfo* – индекс налоговой ставки;
- *UnitPrice* – цена единицы товара;
- *UnitName* – единицы измерения.

Этот метод можно вызывать только в том случае, если свойство **PrinterState** равно `FPTR_PS_FISCAL_RECEIPT`

PrintRecItemVoid

Синтаксис: `LONG PrintRecItemVoid (BSTR Description, CURRENCY Price, LONG Quantity, LONG VatInfo, CURRENCY UnitPrice, BSTR UnitName)`

Замечания: метод осуществляет операцию возврата продажи.

Параметры:

- *Description* – наименование товара;
- *Price* – цена единицы товара; используется, если *UnitPrice* = 0;
- *Quantity* – количество (обратите внимание также на *QuantityLength* и *QuantityDecimalPlaces*. Значение 0 эквивалентно 1000;
- *VatInfo* – индекс налоговой ставки;
- *UnitPrice* – цена единицы товара;
- *UnitName* – единицы измерения.

Если *CapOnlyVoidLastItem* равно *TRUE*, то можно отменить только последнюю операцию продажи.

Если *CapPostPreLine* равно *TRUE*, то будут напечатаны дополнительные строки, содержащиеся в свойствах *PostLine* и *PreLine*

PrintRecItemAdjustment

Синтаксис : `LONG PrintRecItemAdjustment (LONG AdjustmentType, BSTR Description, CURRENCY Amount, LONG VatInfo)`

Замечания: метод осуществляет операцию наценки или скидки на последнюю продажу.

Параметры:

- *AdjustmentType* – тип операции:
 - *FPTR_AT_AMOUNT_DISCOUNT* – скидка в денежных единицах;
 - *FPTR_AT_AMOUNT_SURCHAGE* – наценка в денежных единицах;
 - *FPTR_AT_PERCENT_DISCOUNT* – скидка в процентах;
 - *FPTR_AT_PERCENT_SURCHAGE* – наценка в процентах;
- *Description* – описание операции;
- *Amount* – сумма или процент;
- *VatInfo* – индекс налоговой ставки.

Этот метод можно вызывать только в том случае, если свойство *PrinterState* равно *FPTR_PS_FISCAL_RECEIPT*

PrintRecItemAdjustmentVoid

Синтаксис: *LONG PrintRecItemAdjustmentVoid (LONG AdjustmentType, BSTR Description, CURRENCY Amount, LONG VatInfo)*

Замечания: метод отменяет ранее проведенную операцию наценки или скидки на последнюю продажу.

Параметры:

- *AdjustmentType* – тип операции:
 - *FPTR_AT_AMOUNT_DISCOUNT* – скидка в денежных единицах;
 - *FPTR_AT_AMOUNT_SURCHAGE* – наценка в денежных единицах;
 - *FPTR_AT_PERCENT_DISCOUNT* – скидка в процентах;
 - *FPTR_AT_PERCENT_SURCHAGE* – наценка в процентах;
- *Description* – описание операции;
- *Amount* – сумма или процент;
- *VatInfo* – индекс налоговой ставки.

Метод доступен только тогда, когда свойство ***PrinterState*** равно *FPTR_PS_FISCAL_RECEIPT*

PrintRecItemFuel

Синтаксис: *LONG PrintRecItemFuel (BSTR Description, CURRENCY Price, LONG Quantity, LONG VatInfo, CURRENCY UnitPrice, BSTR UnitName, CURRENCY SpecialTax, BSTR SpecialTaxName)*

Замечания: метод не реализован в текущей версии драйвера.

PrintRecItemFuelVoid

Синтаксис: *LONG PrintRecItemFuelVoid (BSTR Description, CURRENCY Price, LONG VatInfo, CURRENCY SpecialTax)*

Замечания: метод не реализован в текущей версии драйвера.

PrintRecMessage

Синтаксис: *LONG PrintRecMessage (BSTR Message)*

Замечания: метод печатает произвольную строку на фискальном чеке после подведения итога и до окончания формирования чека. Строка - параметр метода может содержать команды принтера.

Метод может быть вызван только в том случае, если ***CapAdditionalLines*** рано *TRUE*

Метод доступен только тогда, когда ***PrinterState*** равно *FPTR_PS_FISCAL_RECEIPT_ENDING*

PrintRecNotPaid

Синтаксис: *LONG PrintRecNotPaid (BSTR Description, CURRENCY Amount)*

Замечания: метод не реализован в текущей версии драйвера.

PrintRecPackageAdjustment

Синтаксис: *LONG PrintRecPackageAdjustment (LONG AdjustmentType, BSTR Description, BSTR VatAdjustment)*

Замечания: метод не реализован в текущей версии драйвера.

PrintRecPackageAdjustVoid

Синтаксис: *LONG PrintRecPackageAdjustVoid (LONG AdjustmentType, BSTR VatAdjustment)*

Замечания: метод осуществляет сторнирование операции наценки или скидки на группу проданных товаров.

Параметры:

- *AdjustmentType* – тип операции:
 - *FPTR_AT_AMOUNT_DISCOUNT* – скидка в денежных единицах;
 - *FPTR_AT_AMOUNT_SURCHAGE* – наценка в денежных единицах;
 - *FPTR_AT_PERCENT_DISCOUNT* – скидка в процентах;
 - *FPTR_AT_PERCENT_SURCHAGE* – наценка в процентах;
- *VatAdjustment* – строка, содержащая список скидок или наценок для разных ставок.

Этот метод можно вызывать только в том случае, если свойство *CapPackageAdjustment* равно *TRUE*

PrintRecRefund

Синтаксис: *LONG PrintRecRefund (LONG Description, CURRENCY Amount, LONG VatInfo)*

Замечания: метод осуществляет операцию возврата.

Параметры:

- *Description* – название товара;
- *Amount* – сумма;
- *VatInfo* – индекс налоговой ставки.

Этот метод можно вызывать только в том случае, если свойство *PrinterState* равно *FPTR_PS_FISCAL_RECEIPT*

PrintRecRefundVoid

Синтаксис: *LONG PrintRecRefundVoid (LONG Description, CURRENCY Amount, LONG VatInfo)*

Замечания: метод производит сторнирование в чеке возврата.

Параметры:

- *Description* – название товара;
- *Amount* – сумма;
- *VatInfo* – индекс налоговой ставки.

Этот метод можно вызывать только в том случае, если свойство *PrinterState* равно *FPTR_PS_FISCAL_RECEIPT*

PrintRecSubtotal

Синтаксис: *LONG PrintRecSubtotal (CURRENCY Amount)*

Замечания: метод печатает текущее значение промежуточного итога.

Параметры:

- *Amount* – значение промежуточного итога, посчитанное приложением.

Если свойство **CheckTotal** равно *TRUE*, то устройство производит сравнение промежуточного итога, посчитанного приложением со своим промежуточным итогом. Если значения не совпадут, то будет выведена соответствующая ошибка.

Этот метод можно вызывать только в том случае, если свойство **PrinterState** равно *FPTR_PS_FISCAL_RECEIPT*

PrintRecSubtotalAdjustment

Синтаксис: *LONG PrintRecSubtotalAdjustment (LONG AdjustmentType, BSTR Description, CURRENCY Amount)*

Замечания: мтеод осуществляет операцию наценки или скидки на подитог.

Параметры:

- *AdjustmentType* – тип операции:
 - *FPTR_AT_AMOUNT_DISCOUNT* – скидка в денежных единицах;
 - *FPTR_AT_AMOUNT_SURCHAGE* – наценка в денежных единицах;
 - *FPTR_AT_PERCENT_DISCOUNT* – скидка в процентах;
 - *FPTR_AT_PERCENT_SURCHAGE* – наценка в процентах;
- *Description* – описание операции;
- *Amount* – сумма или процент.

Этот метод можно вызывать только в том случае, если свойство **PrinterState** равно *FPTR_PS_FISCAL_RECEIPT*

PrintRecSubtotalAdjustVoid

Синтаксис: *LONG PrintRecSubtotalAdjustVoid (LONG AdjustmentType, CURRENCY Amount)*

Замечания: метод осуществляет сторнирование наценки или скидки на подитог.

Параметры:

- *AdjustmentType* – тип операции:
 - *FPTR_AT_AMOUNT_DISCOUNT* – сторнирование скидки в денежных единицах;
 - *FPTR_AT_AMOUNT_SURCHAGE* – сторнирование наценки в денежных единицах;
 - *FPTR_AT_PERCENT_DISCOUNT* – сторнирование скидки в процентах;
 - *FPTR_AT_PERCENT_SURCHAGE* – сторнирование наценки в процентах;
- *Amount* – сумма или процент.

Этот метод можно вызывать только в том случае, если свойство **PrinterState** равно *FPTR_PS_FISCAL_RECEIPT*

PrintRecTaxID

Синтаксис: *LONG PrintRecTaxID (BSTR TaxID)*

Замечания: метод не реализован в текущей версии драйвера.

PrintRecTotal

Синтаксис: *LONG PrintRecTotal (CURRENCY Total, CURRENCY Payment, BSTR Description)*

Замечания: метод осуществляет печать итога по чеку и производит оплату.

Параметры:

- *Total* – значение промежуточного итога, посчитанное приложением;
- *Payment* – сумма платежа;
- *Description* – индекс вида платежа.

Если свойство **CheckTotal** равно *TRUE*, то устройство производит сравнение промежуточного итога, посчитанного приложением со своим промежуточным итогом. Если значения не совпадут, то будет выведена соответствующая ошибка.

Если свойство **PrinterState** равно *FPTR_PS_FISCAL_RECEIPT*, то печатается итог по чеку.

Если свойство **PrinterState** равно *FPTR_PS_FISCAL_RECEIPT* или *FPTR_PS_FISCAL_RECEIPT_TOTAL*, то производится оплата и **PrinterState** становится равным *FPTR_PS_FISCAL_RECEIPT_TOTAL*

Если в результате сумма оплаты превышает или равна итогу, то печатается сдача и **PrinterState** становится равным *FPTR_PS_FISCAL_RECEIPT_ENDING*

PrintRecVoid

Синтаксис: *LONG PrintRecVoid (BSTR Description)*

Замечания: выполняет аннулирование чека.

Параметры:

- *Description* – описание операции.

Если метод успешно выполняется, то **PrinterState** становится равным *FPTR_PS_FISCAL_RECEIPT_ENDING*

PrintRecVoidItem

Метод не используется, начиная с версии *Control Object 1.11.00*

BeginFiscalDocument

Синтаксис: *LONG BeginFiscalDocument (LONG DocumentAmount)*

Замечания: метод не реализован в текущей версии драйвера.

EndFiscalDocument

Синтаксис: *LONG EndFiscalDocument ()*

Замечания: метод не реализован в текущей версии драйвера.

PrintFiscalDocumentLine

Синтаксис: *LONG PrintFiscalDocumentLine (BSTR DocumentLine)*

Замечания: метод не реализован в текущей версии драйвера.

BeginItemList

Синтаксис: *LONG BeginItemList (LONG VatID)*

Замечания: метод не реализован в текущей версии драйвера.

EndItemList

Синтаксис: *LONG EndItemList ()*

Замечания: метод не реализован в текущей версии драйвера.

VerifyItem

Синтаксис: *LONG VerifyItem (BSTR ItemName, LONG VatID)*

Замечания: метод не реализован в текущей версии драйвера.

PrintPeriodicTotalsReport

Синтаксис: *LONG PrintPeriodicTotalsReport (BSTR Date1, BSTR Date2)*

Замечания: метод печатает отчет по сменам в указанном периоде дат.

Параметры:

- *Date1* – начальная дата;
- *Date2* – конечная дата;

PrintPowerLossReport

Синтаксис: *LONG PrintPowerLossReport ()*

Замечания: метод печатает отчет по потерям питания.

PrintReport

Синтаксис: *LONG PrintReport (LONG ReportType, BSTR StartNum, BSTR EndNum)*

Замечания: метод печатает отчет по фискальной памяти.

Параметры:

- *ReportType* – тип отчета:
 - *FPTR_RT_ORDINAL* – полный отчет по фискальной памяти
 - *FPTR_RT_DATE* – в диапазоне дат *StartNum* и *EndNum* в формате "ддммггггччмм", где:
 - *дд* – день месяца(01–31);
 - *мм* – месяц(01–12);
 - *гггг* – год(2005–20xx);
 - *чч* – час(0–23);
 - *мм* – минута(0–59);
 - *FPTR_RT_EOD_ORDINAL* – в диапазоне номеров смен *StartNum* и *EndNum*;
- *StartNum* – маркер начала отчета;
- *EndNum* – маркер конца отчета.

PrintXReport

Синтаксис: *LONG PrintXReport ()*

Замечания: метод печатает суточный отчет без гашения (X–отчет).

PrintZReport

Синтаксис: *LONG PrintZReport ()*

Замечания: метод печатает суточный отчет с гашением (Z-отчет).

BeginInsertion

Синтаксис: *LONG BeginInsertion (LONG TimeOut)*

Замечания: метод не реализован в текущей версии драйвера.

BeginRemoval

Синтаксис: *LONG BeginRemoval (LONG TimeOut)*

Замечания: метод не реализован в текущей версии драйвера.

EndInsertion

Синтаксис: *LONG EndInsertion (LONG TimeOut)*

Замечания: метод не реализован в текущей версии драйвера.

EndRemoval

Синтаксис: *LONG EndRemoval (LONG TimeOut)*

Замечания: метод не реализован в текущей версии драйвера.

BeginFixedOutput

Синтаксис: *LONG BeginFixedOutput (LONG Station, LONG DocumentType)*

Замечания: метод не реализован в текущей версии драйвера.

BeginNonFiscal

Синтаксис: *LONG BeginNonFiscal ()*

Замечания: метод начинает печать не фискального документа.

Этот метод можно вызывать только в том случае, если свойство *PrinterState* равно *FPTR_PS_MONITOR*

В случае успешного выполнения свойство *PrinterState* становится равным *FPTR_PS_NON_FISCAL*

BeginTraining

Синтаксис: *LONG BeginTraining ()*

Замечания: метод не реализован в текущей версии драйвера.

EndFixedOutput

Синтаксис: *LONG EndFixedOutput ()*

Замечания: метод не реализован в текущей версии драйвера.

EndNonFiscal

Синтаксис: *LONG EndNonFiscal ()*

Замечания: метод завершает печать не фискального документа.

Этот метод можно вызывать только в том случае, если свойство *PrinterState* равно *FPTR_PS_NON_FISCAL*

В случае успешного выполнения свойство *PrinterState* становится равным *FPTR_PS_MONITOR*

EndTraining

Синтаксис: *LONG EndTraining ()*

Замечания: метод не реализован в текущей версии драйвера.

PrintFixedOutput

Синтаксис: *LONG PrintFixedOutput (LONG DocumentType, LONG LineNumber, BSTR Data)*

Замечания: метод не реализован в текущей версии драйвера.

PrintNormal

Синтаксис: *LONG PrintNormal (LONG Station, BSTR Data)*

Замечания: печать произвольного текста на чеке в режиме не фискального документа.

Строка - параметр метода может содержать команды принтера.

Этот метод можно вызывать только в том случае, если свойство *PrinterState* равно *FPTR_PS_NON_FISCAL*

GetData

Синтаксис: *LONG GetData (LONG DataItem, LONG *OptArgs, BSTR *Data)*

Замечания: получение различной информации о состоянии устройства.

Параметры:

- *DataItem* – индекс запрашиваемой информации:
 - *FPTR_GD_FIRMWARE*
 - *FPTR_GD_PRINTER_ID*
 - *FPTR_GD_CURRENT_TOTAL*
 - *FPTR_GD_RECEIPT_NUMBER*
 - *FPTR_GD_FISCAL_REC*
 - *FPTR_GD_FISCAL_REC_VOID*
 - *FPTR_GD_Z_REPORT*
- *Data* – возвращаемое значение.

GetDate

Синтаксис: `LONG GetDate (BSTR *Date)`

Замечания: получение даты и времени из устройства.

Параметры:

- *Date* – дата и время в виде строки формата "ддммггггччмм", где:
 - *дд* – день месяца(01–31);
 - *мм* – месяц (01–12);
 - *гггг* – год(1997–20xx);
 - *чч* – час(0–23);
 - *мм* – минута(0–59).

Пример:

```
BSTR Date;
GetDate(&Date);
```

GetTotalizer

Синтаксис: `LONG GetTotalizer (LONG VatID, LONG OptArgs, BSTR *Data)`

Замечания: метод получает значение указанного в свойстве **TotalizerType** счетчика. Этот метод можно вызывать только в том случае, если свойство **CapTotalizerType** равно **TRUE**

Параметры:

- *VatID* – идентификатор налоговой ставки;
- *Data* – возвращаемое значение счетчика;
- *OptArgs* – определяет тип запрашиваемого счетчика. Может принимать следующие значения:

• <i>FPTR_GT_DISCOUNT</i>	• <i>FPTR_GT_SUBTOTAL_DISCOUNT</i>
• <i>FPTR_GT_NET</i>	• <i>FPTR_GT_SUBTOTAL_DISCOUNT_VOID</i>
• <i>FPTR_GT_GROSS</i>	• <i>FPTR_GT_SUBTOTAL_SURCHARGES</i>
• <i>FPTR_GT_VAT</i>	• <i>FPTR_GT_SUBTOTAL_SURCHARGES_VOID</i>
• <i>FPTR_GT_VAT_CATEGORY</i>	• <i>FPTR_GT_SURCHARGE</i>
• <i>FPTR_GT_ITEM</i>	• <i>FPTR_GT_SURCHARGE_VOID</i>
• <i>FPTR_GT_ITEM_VOID</i>	• <i>FPTR_GT_REFUND</i>

GetVatEntry

Синтаксис: `LONG GetVatEntry (LONG VatID, LONG OptArgs, LONG *VatRate)`

Замечания: метод считывает значение указанной налоговой ставки. Этот метод можно вызывать только в том случае, если свойство **CapHasVatTable** равно **TRUE**

Параметры:

- *VatID* – идентификатор налоговой ставки;
- *OptArgs* – не используется;
- *VatEntry* – возвращаемое значение налоговой ставки;

Смотри также: свойство **CapHasVatTable**

ClearError

Синтаксис: *LONG ClearError ()*

Замечания: метод очищает ошибку.

ResetPrinter

Синтаксис: *LONG ResetPrinter ()*

Замечания: метод сбрасывает устройство в начальное состояние.

Пример формирования фискального документа

Фискальный документ	Описание строки	Свойства и методы OPOS
МУЛЬТИСОФТ СИСТЕМЗ Россия, 111396, г. Москва ул. Алексея Дикого, 3, т/ф (495) 301-05-93/4173 e-mail: info@multisoft.ru	Установленные значения четырех строк HeaderLine (до 6 строк)	
ККТ:0001650 ИНН: 1326033885 ДОК: 00137 11/08/2011 КАССИР: 00	РЕГ: 44444444 ЧЕК: 0001 14:30	<i>FiscalReceiptType = FPTR_RT_SALES;</i> <i>BeginFiscalReceipt(true);</i>
КАССОВЫЙ ЧЕК	Формируемые кассовым аппаратом строки	
Молоко 25.00*1 шт Срок хранения до 05.09.2011	=25.00A Операция продажи с налоговой ставкой группы А	<i>PostLine = "Срок хранения до 05.04.2011";</i> <i>PrintRecItem("Молоко", 25.00 , 10000 , 0 , 25.00 , "шт");</i>
Специальное предложение: Шоколад 10.00*1 шт	=10.00B Операция продажи с налоговой ставкой группы В	<i>PreLine = "Специальное предложение:";</i> <i>PrintRecItem("Шоколад", 10.00 , 10000 , 1 , 10.00 , "шт");</i>
СКИДКА 15.00%	-1.50 Применение скидки к последней операции продажи	<i>PrintRecItemAdjustment(FPTR_AT_PERCENTAGE_DISCOUNT , "СКИДКА" , 15 , 0);</i>
Хлеб 10.00*1 шт	=10.00C Операция продажи с налоговой ставкой группы С	<i>PrintRecItem("Хлеб", 20.00 , 10000 , 2 , 10.00 , "шт");</i>
СТОРНО Хлеб 10.00*1 шт	=10.00C Отмена последней операции продажи	<i>PrintRecItemVoid("Хлеб", 20.00 , 10000 , 2 , 10.00 , "шт");</i>
ПОДИТОГ А 18.00% В 12.00% С 10.00%	=33.50 3.58 1.07 0.00	Операция подсчета и печати подитога <i>PrintRecSubtotal(33.50);</i>
ПРОДАЖА НАЛИЧНЫМИ СДАЧА	=33.50 =35.00 1.50	Операция подсчета и печати итога <i>CheckTotal = false;</i> <i>PrintRecTotal(33.50 , 35.00 , "НАЛИЧНЫМИ");</i>
Дополнительная информация	Операция печати произвольной строки	<i>PrintRecMessage("Дополнительная информация");</i>
СПАСИБО ЗА ПОКУПКУ	Установленное значение строки TrailerLine (до 3 строк)	<i>EndFiscalReceipt(false);</i>
ФИСКАЛЬНЫЙ ДОКУМЕНТ ® ЭКЛЗ: 1234567891 00001234 #111111	Формируемые кассовым аппаратом строки	

Настройки реестра

Параметры OPOS Fiscal Printer драйвера в реестре

Настройки OPOS драйвера хранятся в разделе реестра

HKLM\SOFTWARE\OLEforRetail\ServiceOPOS\FiscalPrinter\MStarTK

- **Path** – путь к библиотеке драйвера;
- **Options** – тип протокола обмена с устройством. В текущей версии поддерживается только "HID".

Остальные настройки являются системными и не подлежат изменению.

Параметры протокола обмена HID

Настройки параметров протокола HID хранятся в разделе реестра для 32-разрядной ОС:

HKLM\SOFTWARE\OLEforRetail\ServiceOPOS\FiscalPrinter\MstarTK\HID

для 64-разрядной ОС:

HKLM\SOFTWARE\Wow6432Node\OLEforRetail\ServiceOPOS\FiscalPrinter\MstarTK\HID

- **LogPath** – строка, содержащая полные пути к директориям, куда будут записываться файлы контрольных лент по сменам. Пути разделены символом ';'. Пример:
"C:\MSTAR_TK\Journal\Archive1;C:\MSTAR_TK\Journal\Archive2"
- **PartialPrint** – гострочная печать. Если = 1, то используется, иначе нет.
- **PaperRecoveryTimeout** – таймаут захвата бумаги. Определяет в миллисекундах время, через которое чек будет захвачен презентером.
- **PrintFlags** – параметры печати налогов. Содержит результат побитового ИЛИ следующих флагов:
 - *0x00000001* – включить вывод суммы налога в конце чека
 - *0x00000002* – включить вывод суммы по налоговой ставке в конце чека
 - *0x00000004* – включить вывод налогооблагаемой базы в конце чека
 - *0x40000000* – печать налогов в позиции после суммы. (По умолчанию - в начале строки)
- **TaxNames** – строка с названиями налогов. Названия разделены символом ";".
Пример: "НАЛОГ 1;НАЛОГ 2;НАЛОГ 3;НАЛОГ 4;НАЛОГ 5"
- **XReportFlags** – параметры печати X отчета. Содержит результат побитового ИЛИ следующих флагов:
 - *0x00000001* – включить в сводный отчет по кассирам
 - *0x00000002* – включить в сводный отчет по отделам
 - *0x00000004* – не включать в отчет нулевые значения
 - *0x00000008* – отчет по номеру
 - *0x00000010* – послать в ответ на команду расширенный ответ
 - *0x00000020* – отчет по кассирам
 - *0x00000040* – отчет по отделам
 - *0x00000080* – отчет по налогам
- **ZReportFlags** – параметры печати Z отчета. Содержит результат побитового ИЛИ следующих флагов:
 - *0x00000001* – включить в сводный отчет по кассирам
 - *0x00000002* – включить в сводный отчет по отделам
 - *0x00000004* – не включать в отчет нулевые значения
 - *0x00000200* – отчет в буфер
- **CutterInfo** – режим работы отрезчика:
 - *0x00000080* – включить автоматический отрез документа.
 - *0x0000007F* – количество строк протяжки до отреза документа (маска)
- **FiscalReportFlags** – параметры печати фискальных отчетов. Содержит результат

побитового ИЛИ следующих флагов:

- `0x00000002` – полный отчет по фискальной памяти

Остальные настройки являются системными и не подлежат изменению.

ССЫЛКИ НА ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ

1) UnifiedPOS Committee.

<http://www.nrf-arts.org/UnifiedPOS/default.htm>

2) Star Line Mode Command Specifications.

<http://www.star-micronics.co.jp/eng/service/usermanual/linemodepm.pdf>

3) Спецификации команд ESC/POS.

Приложение А. Основные положения OLE for Retail POS

Что такое OLE for Retail POS?

Ole for Retail POS это реализация стандарта *UnifiedPOS* основанная на Microsoft Operation System Software и архитектуре OLE 2.x.

Основные цели *Ole for Retail POS* это:

1. Определение архитектуры доступа к POS устройствам на системах Win32
2. Определение набора интерфейсов POS устройств, достаточного для поддержки набора POS решений

Базовая модель Control-объектов OLE for Retail POS

Control-объекты OLE for Retail POS базируются на спецификации *ActiveX Control*. Приложение может управлять Control-объектом через его методы и свойства. Ответ может быть возвращен приложению через возвращаемые значения и параметры методов, через свойства и через события.

На рисунке 1 представлена архитектура *OLE for Retail POS*.

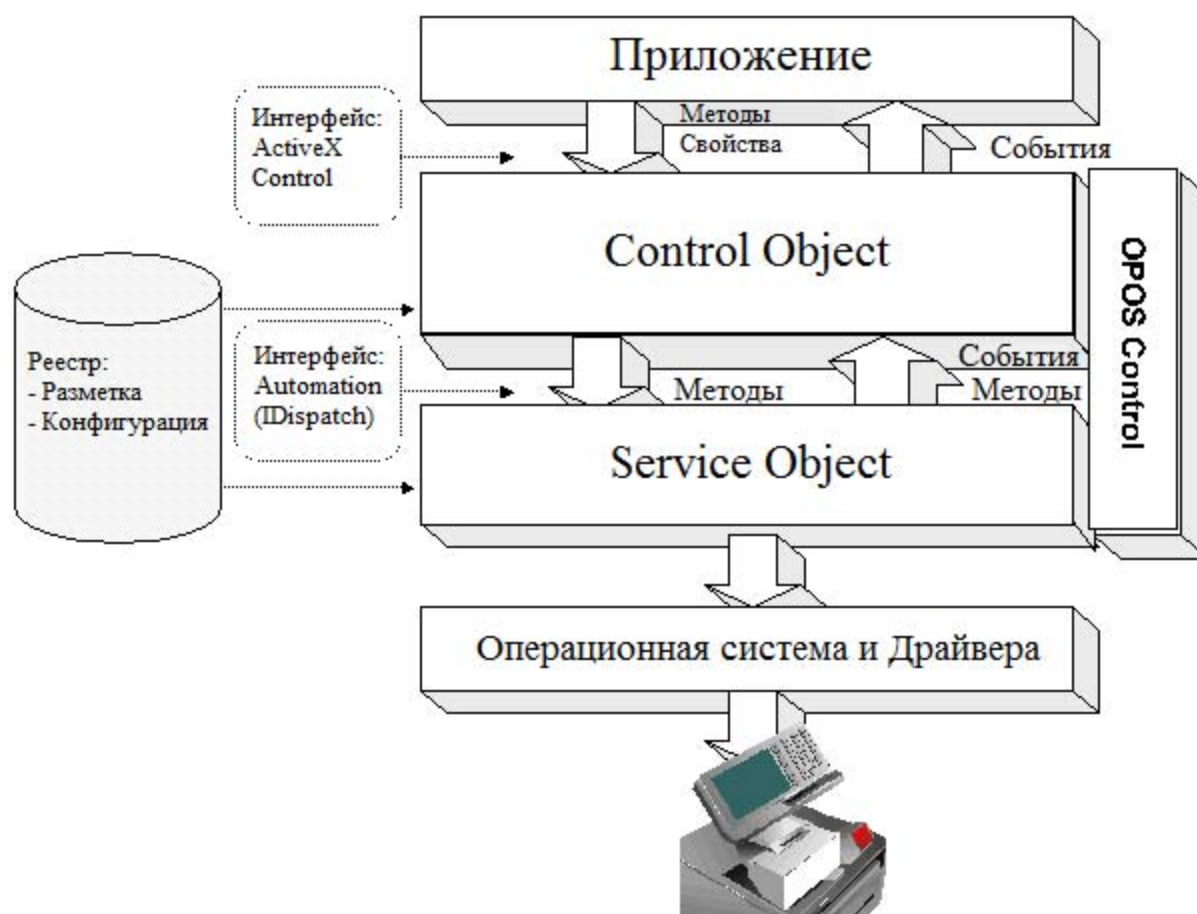


Рисунок 1. Рисунок 1. Архитектура *Ole for Retail POS*

Определения

Device Class

Device Class это класс POS устройств, который предоставляет определенный набор свойств методов и событий. Например, *Cash Drawer* или *POS Printer*.

Некоторые устройства поддерживают больше одного *device class*. Например, некоторые устройства типа *POS Printers* (например, фискальные регистраторы) могут включать в себя также устройства типа *Cash Drawer* (например, денежный ящик).

Control Object or CO

Control Object предоставляет приложению набор свойств, методов, и событий для *device class*, который он реализует.

CO это стандартный *ActiveX* (то есть, OLE 32-bit) *Control*, который не видим во время выполнения. Интерфейсы CO спроектированы таким образом, что все реализации *Control Object* будут совместимы. Это позволяет разрабатывать CO и SO отдельно друг от друга.

Service Object or SO

Service-объект используется *Control*-объектом для работы непосредственно с устройством.

SO реализован как *Automation server*. Он предоставляет набор методов, которые вызываются из CO. Также он может вызывать специальные методы, предоставленные CO для отправки приложению событий. *Service Object* может включать в себя несколько наборов методов, чтобы поддерживать устройства с несколькими классами устройств.

Service-объект обычно реализовывается в виде локального in-proc сервера (в виде DLL). В теории, он также может быть реализован в виде локального out-proc сервера (в виде отдельно запускаемого процесса). Однако на практике было установлено, что *out-proc* сервера не удобны и не всегда работают верно для *OPOS Service Objects*, поэтому не рекомендуется их использовать.

OPOS Control or Control

OPOS Control содержит *Control Object* для *device class*'а (который предоставляет интерфейс) и *Service Object* (который реализовывает интерфейс для конкретного устройства). Это общее название пары CO + SO.

Как приложение использует OPOS Control

Первым делом приложение должно «подхватить» Control, вызвав его метод *Open*. Параметром этого метода является имя, строго определяющее необходимый приложению Control. Метод *Open* совершает следующие шаги:

- 1) Устанавливает связь между *Control Object* и *Service Object*
- 2) Инициализирует свойства *OpenResult*, *Claimed*, *DeviceEnabled*, *DataEventEnabled*, *FreezeEvents*, *AutoDisable*, *DataCount* и *BinaryConversion*. Также, в зависимости от *device class*'а, могут быть инициализированы дополнительные свойства.

OPOS Control-объект одновременно могут открыть несколько приложений. Поэтому, после того как устройство открыто, приложению необходимо вызвать метод *ClaimDevice* для того, чтобы получить эксклюзивный доступ к устройству. Вызов этого метода гарантирует, что никакое другое приложение больше не сможет иметь доступ к этому устройству. Приложение должно вызвать метод *ReleaseDevice* когда оно закончило работу с устройством. После вызова этого метода, другие приложения могут получить доступ к устройству.

Перед тем как использовать устройство, приложение должно присвоить свойству *DeviceEnabled* значение *TRUE*. Эта операция переводит устройство в рабочее состояние, тогда, как присвоение *FALSE* отключает устройство. Например, если Control сканера отключен, тогда устройство будет физически отключено (если возможно). Отключено ли устройство физически или нет, любой обмен с устройством будет невозможен, пока значение свойства *DeviceEnabled* не будет установлено в *TRUE*.

После того, как приложение закончило использовать устройство, необходимо вызвать метод *Close* для того, чтобы освободить устройство и все связанные с ним ресурсы. Если свойство *DeviceEnabled* установлено в *TRUE*, тогда метод *Close* отключит устройство (*DeviceEnabled* = *FALSE*). Если значение свойства *Claimed* равно *TRUE*, тогда метод *Close* автоматически вызовет метод *ReleaseDevice*. Перед завершением своей работы приложение должно закрыть все открытые OPOS Control-объекты.

Таким образом, приложение должно придерживаться следующей основной последовательности:

1. Метод *Open*: устанавливает связь между *Control Object* и *Service Object*.
2. Метод *ClaimDevice*: вызывается для получения эксклюзивного доступа к устройству.
3. Свойство *DeviceEnabled*: установка в *TRUE* переводит устройство в рабочее состояние.
4. Использование устройства.
5. Свойство *DeviceEnabled*: установка в *FALSE* отключает устройство.
6. Метод *ReleaseDevice*: отпускает эксклюзивный доступ. Другие приложения могут обратиться к устройству.
7. Метод *Close*: освобождает устройство и все связанные с ним ресурсы.

Когда методы и свойства могут быть доступны

Methods

Первым всегда должен быть успешно вызван метод **Open**. Любой другой метод до вызова **Open** будет возвращать статус *OPOS_E_CLOSED*

Устройства эксклюзивного использования требуют, чтобы приложение вызвало метод **ClaimDevice** и установило значение свойства **DeviceEnabled** в *TRUE*, перед тем как использовать большинство остальных методов.

Устройства с разделяемым доступом требуют, чтобы приложение установило значение свойства **DeviceEnabled** в *TRUE*, перед тем как использовать большинство остальных методов.

Properties

Перед удачным вызовом **Open**, значение большинства свойств не инициализировано. Попытка установить значение изменяемых свойств будет проигнорирована.

Следующие свойства всегда инициализированы:

Свойство	Значение
State	<i>OPOS_S_CLOSED</i>
ResultCode	<i>OPOS_S_CLOSED</i>
ControlObjectDescription	<i>Control Object</i> зависимая строка
ControlObjectVersion	<i>Control Object</i> зависимое число

Свойства, названия которых начинаются с **Cap...** (свойства, описывающие возможности драйвера) инициализируются после спешного вызова метода **Open**

Устройства эксклюзивного использования требуют, чтобы приложение вызвало метод **ClaimDevice** и установило значение свойства **DeviceEnabled** в *TRUE*, перед тем как некоторые другие свойства будут инициализированы или их значения могут быть изменены.

Устройства с разделяемым доступом требуют, чтобы приложение установило значение свойства **DeviceEnabled** в *TRUE*, перед тем как некоторые другие свойства будут инициализированы или их значения могут быть изменены.

Установка значений изменяемых свойств до того, как будут выполнены все требования, будет игнорироваться. В результате таких действий будет устанавливаться свойство **ResultCode** в значения *OPOS_E_NOTCLAIMED* или *OPOS_E_DISABLED*

Чтение значений неинициализированных свойств будет возвращать следующие значения (кроме специально оговоренных случаев):

Тип свойства	Значение
<i>Boolean</i>	<i>FALSE</i>
<i>Long</i>	0
<i>String</i>	«[Error]» (включая квадратные скобки)

После того как свойства были инициализированы они остаются таковыми до вызова метода **Close**

Модели Status, Result code и State

Модели *status*, *result code* и *state* построены вокруг нескольких основных свойств, событий и методов, указанных в следующей таблице:

Таблица 5. Модели Status, Result code и State

Имя	Значение
<i>State</i>	<p>Свойство, содержащее текущее состояние <i>Control</i>'а:</p> <ul style="list-style-type: none"> • <i>OPOS_S_CLOSED</i> • <i>OPOS_S_IDLE</i> • <i>OPOS_S_BUSY</i> • <i>OPOS_S_ERROR</i>
<i>ResultCode</i>	<p>Свойство, содержащее результат выполнения последнего вызванного метода или последнего изменения изменяемого свойства:</p> <ul style="list-style-type: none"> • <i>OPOS_SUCCESS</i> • <i>OPOS_E_CLOSED</i> • <i>OPOS_E_CLAIMED</i> • <i>OPOS_E_NOTCLAIMED</i> • <i>OPOS_E_NOSERVICE</i> • <i>OPOS_E_DISABLED</i> • <i>OPOS_E_ILLEGAL</i> • <i>OPOS_E_NOHARDWARE</i> • <i>OPOS_E_OFFLINE</i> • <i>OPOS_E_NOEXIST</i> • <i>OPOS_E_EXISTS</i> • <i>OPOS_E_FAILURE</i> • <i>OPOS_E_TIMEOUT</i> • <i>OPOS_E_BUSY</i> • <i>OPOS_E_EXTENDED</i> • <i>OPOS_E_DEPRECATED</i>
<i>ResultCodeExtended</i>	<p>Свойство, содержащее расширенный результат выполнения последнего вызванного метода или последнего изменения изменяемого свойства. Значения зависят от <i>Device Class</i>'а.</p>
<i>StatusUpdateEvent</i>	<p>Событие, отправляемое приложению, когда некоторые специфические свойства или <i>status</i> были изменены.</p>
<i>ErrorEvent</i>	<p>Событие, отправляемое приложению, когда свойство <i>status</i> принимает значение <i>OPOS_S_ERROR</i></p>

Модель Status

Правила модели *status* следующие:

- Единственной особенностью модели *status* является то, что приложение оповещается об изменении статуса с помощью *StatusUpdateEvent*. Эта особенность справедлива для всех классов устройств.
- Каждый *device class* сам определяет, какое изменение его статуса будет оповещать приложение с помощью события. Пример такого определения:
 - Изменение позиции денежного ящика (например, изменение положения из «закрыто» в «открыто»).
 - Изменение в сенсоре POS принтера (например, активация сенсора «присутствия формы», следящего за тем что была установлена форма в принтере подкладной печати).

Модель Result Code

Правила модели *result code* следующие:

- Каждый метод возвращает *result code*. Этот код также помещается в свойство *ResultCode*
- Установка значения изменяемого свойства также ведет к тому, что *result code* будет помещен в свойство *ResultCode*
- Значение *ResultCode* *OPOS_SUCCESS* приравнен к значению ноль. Ненулевые значения означают ошибку или предупреждение.
- *Control*-объект должен выбирать коды результатов по следующим правилам:
 - если *Control* устанавливает значение свойства *ResultCode* в *OPOS_E_EXTENDED*, тогда он должен установить значение свойства *ResultCodeExtended* в одно из значений, определенных документацией его *device class*'а.
 - если *Control* устанавливает значение свойства *ResultCode* в *OPOS_E_EXTENDED*, тогда *Service Object* может установить значение свойства *ResultCodeExtended* в любое специфическое для него значение SO. Описание этих значений необходимо искать в документации к устройству.

Модель State

Правила модели *state* следующие:

- Свойство *State* инициализируется сразу значением *OPOS_S_CLOSED*
- Значение свойства *State* изменяется на *OPOS_S_IDLE*, когда вызван метод *Open* и его результат равен *OPOS_SUCCESS*
- Значение свойства *State* изменяется на *OPOS_S_BUSY*, когда идет процесс вывода. Значение свойства *State* возвращается в *OPOS_S_IDLE*, когда этот процесс завершен.
- Значение свойства *State* устанавливается в *OPOS_S_ERROR* когда:
 - В процессе асинхронного вывода возникла ошибка.
 - Возникла ошибка в процессе ввода, подтверждаемого событиями.

После того, как значение свойства *State* изменилось на *OPOS_S_ERROR*, происходит отправка события *ErrorEvent*. Параметрами этого события являются *result code* и *extended result code*, местоположение ошибки, и указатель на ответ приложения. Местоположение может быть следующим:

- *Output* — ошибка в процессе вывода.
- *InputWithData* — ошибка в процессе ввода данных, когда данные не были до конца обработаны.
- *InputNoData* — ошибка в процессе ввода данных, когда данные были обработаны.

Когда приложение возвращает управление из обработчика события **ErrorEvent**, оно может установить ответ на событие. Ответы могут быть следующими:

- *Retry* – если местоположение - *Output*: повторить асинхронный вывод и выйти из состояния ошибки. Если в процессе повтора снова возникла ошибка, будет сгенерирован еще один **ErrorEvent**. Если местоположение - *Input*: некоторые устройства могут поддерживать повторный ввод. "*Retry*" - это ответ по умолчанию, если местоположение - "*Output*"
- *Clear* – сбросить все буферы вывода (включая асинхронные) или буферы ввода и выйти из состояния ошибки. "*Clear*" – это ответ по умолчанию, если местоположение - "*InputNoData*"
- *Continue* – используется только если местоположение - *InputWithData*. Этот ответ просит продолжить процесс ввода. *Control* остается в состоянии ошибки, и будет продолжать доставлять дополнительные *data events* как указано в свойстве **DataEventEnabled**. Когда все входные данные будут доставлены и значение свойства **DataEventEnabled** будет установлено в *TRUE*, тогда другой **ErrorEvent** с местоположением "*InputNoData*" будет доставлен. "*Continue*" – это ответ по умолчанию, если местоположение - "*InputNoData*"

Control гарантирует, что пока приложение обрабатывает текущий **ErrorEvent**, другой **ErrorEvent** не будет отправлен.

Модель доступа к устройству

Модель доступа к устройству *OLE for Retail POS* поддерживает как устройства, которые могут быть использованы только из одного приложения в одно и то же время, так и устройства, которые могут быть частично или полностью использованы несколькими приложениями. Все *OPOS Control*-объекты могут быть открыты больше чем одним приложением в одно и то же время. Однако, действия, которые приложение может производить над *Control*-объектом, могут быть выполнены только приложением, «захватившим» доступ к устройству.

Устройства эксклюзивного использования

Самый распространенный тип устройств называется "устройства эксклюзивного использования". Например, *POS printer*. Как по физическим так по техническим характеристикам это устройство может быть использовано только одним приложением в одно и то же время. Приложение должно вызвать метод *ClaimDevice* для получения эксклюзивного доступа к устройству перед использованием большинства методов, свойств или событий. Пока устройство не захвачено, вызов методов или установка свойств будет вызывать ошибку *OPOS_E_NOTCLAIMED*, и события не будут доставляться к приложению.

Если несколько приложений хотят разделить доступ к устройству, тогда одно приложение может захватить устройство, выполнить свою последовательность действий, затем отпустить его, для того чтобы другое приложение могло использовать устройство.

Когда снова вызван метод *ClaimDevice*, изменяемые параметры устройства восстанавливают свои значения на момент вызова *ReleaseDevice*. Примером восстановления характеристик может послужить яркость дисплея. Характеристики состояния устройства не восстанавливаются. Например, состояние датчиков принтера. Они обновляются новыми значениями.

Устройства с разделяемым доступом

Некоторые устройства могут быть "разделяемыми устройствами". Например, *keylock*. Разделяемое устройство позволяет множеству приложений вызывать его методы и получать доступ к свойствам. Также оно может доставлять события всем приложениям, использующим его. Разделяемые устройства могут ограничивать количество приложений, которые могут их использовать тем или иным способом.

Подразумевается, что приложение содержит только один процесс.

События

OLE for Retail POS использует механизм событий для информирования о различной активности или изменении *OPOS Control*'а. Существует пять типов событий:

- 1) **DataEvent**: входные данные были опущены в определенные классом устройств свойства.
- 2) **ErrorEvent**: возникла ошибка в процесса ввода или асинхронного вывода.
- 3) **StatusUpdateEvent**: информирует о изменении статуса устройства.
- 4) **OutputCompleteEvent**: асинхронный вывод удачно завершен.
- 5) **DirectIOEvent**: это событие может быть использовано поставщиком драйвера в своих целях.

Когда возникают события, *Service Object* ставит их в конец очереди. Обычно, а работу очереди событий отвечает отдельный поток. Если все условия доставки соблюдены, события, находящиеся в очереди, будут доставлены приложению. Условия, которые могут задержать отправку событий:

- Поток приложения, обрабатывающий события, еще занят предыдущим событием. *OPOS Control*-объекты придерживаются модели *OLE Apartment Threading*. Соответствуя правилам *OLE Apartment Threading*, события будут доставляться потоку, который создал СОМ объект. Этот поток обычно является основным потоком приложения. Если приложение занято обработкой другого события, доставка будет приостановлена, пока этот процесс не завершится.
- Приложение присвоило свойству **FreezeEvents** значение *TRUE*
- Тип события **DataEvent** или **ErrorEvent**, но значение свойства **DataEventEnabled** равно *FALSE*

Если первое в очереди на отправку событие заблокировано, то остальные события также будут заблокированы. Это значит, что очередь событий организована как FIFO (first in first out).

Следует помнить, что свойства, отражающие состояние не блокируются, несмотря на значение **FreezeEvents**. То есть, они будут менять свои значения без отправки сообщений.

Модель вывода

Модель вывода *OLE for Retail POS* содержит вывод двух типов: синхронный и асинхронный. Класс устройств может поддерживать как один, так и оба типа.

Синхронный вывод

Этот тип вывода используется, все операции в устройстве обрабатываются быстро. Его преимущество - простота.

Приложение вызывает метод и *Control* не возвращает управления до тех пор, пока не выполнит операцию.

Асинхронный вывод

Этот тип вывода используется, когда в устройстве присутствуют длинные и медленные операции. Его преимуществом является то, что, пока устройство выполняет операцию, приложение может заниматься другими задачами.

Приложение вызывает метод, затем в свойство *OutputID* устанавливается идентификатор текущего запроса, и приложению сразу возвращается управление. Далее приложение может заниматься другими задачами. Когда операция, инициированная методом, будет выполнена, приложению будет доставлено событие *OutputCompleteEvent*, параметр которого будет содержать идентификатор, выставленный в *OutputID*.

Если в процессе асинхронной работы возникнет ошибка, будет отправлено событие *ErrorEvent*. Приложение может попытаться повторить запрос или сбросить его. *Control* будет находиться в состоянии ошибки, пока обрабатывается событие *ErrorEvent*.

Асинхронный вывод организован по принципу FIFO (first in first out).

Все буферизованные данные, включая асинхронный вывод, могут быть сброшены методом *ClearOutput*. Этот метод также останавливает любой текущий вывод, если такое возможно.

Приложение Б. Команды Star Line Mode

Для устройств серии MSTAR, в свойствах *PreLine* и *PostLine*, а так же в параметрах методов *PrintRecMessage* и *PrintNormal* строки могут содержать *Esc* - последовательности в виде команд *Star Line Mode*. Также команды *Star Line Mode* могут быть переданы через специальную команду *DirectIO*. Детальное описание параметров команд выходит за рамки данного документа. Более подробную информацию можно получить из спецификации Star Line Mode [2].

Разрешенные команды

В следующей таблице указана возможность использования разрешенных команд в различных методах и свойствах.

Таблица 6. Разрешенные команды Star Line Mode.

Название команды	ASCII код команды	Возможно использовать в				
		<i>PreLine</i>	<i>PostLine</i>	<i>PrintRecMessage</i>	<i>PrintNormal</i>	<i>DirectIO</i>
Управление шириной/высотой символов	ESC I	Да	Да	Да	Да	Нет
Печать штрих-кодов	ESC b	Да	Да	Да	Да	Нет
Выделение текста	ESC E	Да	Да	Да	Да	Нет
Отмена выделения текста	ESC F	Да	Да	Да	Да	Нет
Управление автоотрезом	ESC d	Нет	Нет	Нет	Нет	Да
Протяжка бумаги	ESC a	Нет	Нет	Нет	Нет	Да
Установка Memory switch	ESC GS #	Нет	Нет	Нет	Нет	Да
Сброс принтера	ESC ?	Нет	Нет	Нет	Нет	Да
Инициализация сигнала для внешнего устройства	ESC BEL	Нет	Нет	Нет	Нет	Да
Подать сигнал для внешнего устройства	BEL	Нет	Нет	Нет	Нет	Да
Возврат бумаги презентером	ESC SYN	Нет	Нет	Нет	Нет	Да

ВНИМАНИЕ! Для передачи нулевого параметра (n=0) следует использовать строковый ноль (\x30), т.к. цифровой(\x00) будет интерпретирован как конец строки и не будет передан принтеру.

Управление шириной/высотой символов

- Формат команды:
 - ASCII: ESC i n1 n2
 - Hex: 1B 69 n1 n2
 - Decimal: 27 105 n1 n2
- Допустимые значения:
 - $0 \leq n1 \leq 5$;
 - $48 \leq n1 \leq 53$ ("0" ≤ n1 ≤ "5");
 - $0 \leq n2 \leq 5$;
 - $48 \leq n2 \leq 53$ ("0" ≤ n2 ≤ "5").
- Начальные значения:
 - n1 = 0 (Режим удвоенной высоты выключен)
 - n2 = 0 (Режим удвоенной ширины выключен)
- Описание:
 - устанавливает/отменяет двойную высоту/ширину символов. Команда игнорируется, если в качестве ее параметров переданы не допустимые значения.

Печать штрих-кодов

- Формат команды:
 - ASCII: ESC b n1 n2 n3 n4 d1 ... dk RS
 - Hex: 1B 62 n1 n2 n3 n4 d1 ... dk 1E
 - Decimal: 27 98 n1 n2 n3 n4 d1 ... dk 30
- Допустимые значения:
 - $0 \leq n1 \leq 8$;
 - $48 \leq n1 \leq 56$ ("0" ≤ n1 ≤ "8");
 - $1 \leq n2 \leq 4$;
 - $49 \leq n2 \leq 52$ ("1" ≤ n2 ≤ "4");
 - $1 \leq n4 \leq 127$;
 - n3 – режим штрих-кода;
 - k – количество байт данных штрих-кода.
- Описание:
 - печать штрих-кодов выполняется соответственно переданным параметрам. Если n1, n2, n3 и n4 находятся вне области допустимых параметров, команда игнорируется.

Выделение текста

- Формат команды:
 - ASCII: ESC E
 - Hex: 1B 45
 - Decimal: 27 69
- Описание:
 - команда включает режим печати текста с выделением.

Отмена выделения текста

- Формат команды:
 - ASCII: ESC F
 - Hex: 1B 46
 - Decimal: 27 70
- Описание:
 - команда отменяет печать текста с выделением.

Управление автоотрезчиком

- Формат команды:
 - ASCII: ESC d n
 - Hex: 1B 64 n
 - Decimal: 27 100 n
- Допустимые значения:
 - $0 \leq n \leq 3$;
 - $48 \leq n1 \leq 51$ ("0" ≤ n1 ≤ "3");
- Описание:
 - выполняет автоматический отрез бумаги.

Протяжка бумаги

- Формат команды:
 - ASCII: ESC a n
 - Hex: 1B 61 n
 - Decimal: 27 97 n
- Допустимые значения:
 - $1 \leq n \leq 255$;
- Описание:
 - протягивает бумагу на n линий.

Установка Memory switch

- Формат команды:
 - ASCII: ESC GS # m N n1 n2 n3 n4 LF NUL
 - Hex: 1B 1D 23 m N n1 n2 n3 n4 0A 00
 - Decimal: 27 29 35 m N n1 n2 n3 n4 10 0
- Допустимые значения:
 - $m = 87, 84, 44, 43, 45, 64$ ($m = "W", "T", ",", "+", "-", "@"$);
 - $48 \leq n1 \leq 57$ ("0" \leq n1 \leq "9"), $65 \leq n1 \leq 70$ ("A" \leq n1 \leq "F"), $97 \leq n1 \leq 102$ ("a" \leq n1 \leq "f");
 - $48 \leq n2 \leq 57$ ("0" \leq n2 \leq "9"), $65 \leq n2 \leq 70$ ("A" \leq n2 \leq "F"), $97 \leq n2 \leq 102$ ("a" \leq n2 \leq "f");
 - $48 \leq n3 \leq 57$ ("0" \leq n3 \leq "9"), $65 \leq n3 \leq 70$ ("A" \leq n3 \leq "F"), $97 \leq n3 \leq 102$ ("a" \leq n3 \leq "f");
 - $48 \leq n4 \leq 57$ ("0" \leq n4 \leq "9"), $65 \leq n4 \leq 70$ ("A" \leq n4 \leq "F"), $97 \leq n4 \leq 102$ ("a" \leq n4 \leq "f");
- Описание:
 - устанавливает новое значение во внутренних регистрах.

Сброс принтера

- Формат команды:
 - ASCII: ESC ? LF NUL
 - Hex: 1B 3F 0A 00
 - Decimal: 27 63 10 0
- Описание:
 - программный сброс принтера.

Инициализация сигнала для внешнего устройства

- Формат команды:
 - ASCII: ESC BEL n1 n2
 - Hex: 1B 07 n1 n2
 - Decimal: 27 7 n1 n2
- Допустимые значения:
 - $1 \leq n1 \leq 127$;
 - $1 \leq n2 \leq 127$;
- Начальные значения:
 - $n1 = 20$ (Длительность импульса: 200 мс)
 - $n2 = 20$ (Длительность паузы: 200 мс)
- Описание:
 - устанавливает длительность импульса и паузы для генерации сигнала на внешнее устройство:
 - длительность импульса = $10 \times n1$ (мс)
 - длительность паузы = $10 \times n2$ (мс)

Подать сигнал для внешнего устройства

- Формат команды:
 - ASCII: BEL
 - Hex: 07
 - Decimal: 7
- Описание:
 - подает на внешнее устройство сигнал, сконфигурированный командой ESC BEL

Возврат бумаги презентером

- Формат команды:
 - ASCII: ESC SYN 0 n
 - Hex: 1B 16 30 n
 - Decimal: 27 22 48 n
- Допустимые значения:
 - n = 0, n = 48 ("0")
- Описание:
 - в результате выполнения команды презентер возвращает бумагу во внутренний лоток.

Приложение В. Команды ESC/POS

Для устройств серии MSPrint в свойствах *PreLine* и *PostLine*, а также в параметрах методов *PrintRecMessage* и *PrintNormal* строки могут содержать *Esc*-последовательности в виде команд *ESC/POS*. Также команды *ESC/POS* могут быть переданы через специальную команду *DirectIO*. Детальное описание параметров команд выходит за рамки данного документа. Более подробную информацию можно получить из спецификации ESC/POS [3].

Разрешенные команды

В следующей таблице указана возможность использования разрешенных команд в различных методах и свойствах.

Таблица 7. Разрешенные команды ESC/POS.

Название команды	ASCII код команды	Возможно использовать в				
		<i>PreLine</i>	<i>PostLine</i>	<i>PrintRecMessage</i>	<i>PrintNormal</i>	<i>DirectIO</i>
Установка режима работы автоотрезчика	GS V	Нет	Нет	Нет	Нет	Да
Печать в буфер и протяжка бумаги	LF	Да	Да	Да	Да	Нет
Печать в буфер без протяжки бумаги	CR	Да	Да	Да	Да	Нет
Установка режима печати символов	ESC !	Да	Да	Да	Да	Да
Инициализация сигнала денежному ящику	ESC BEL	Нет	Нет	Нет	Нет	Да
Управляющий сигнал на первый канал денежного ящика	ESC 9	Да	Да	Да	Да	Да
Управляющий сигнал на второй канал денежного ящика	ESC p	Да	Да	Да	Да	Да
Печать штрихкода	GS k	Да	Да	Да	Да	Да

ВНИМАНИЕ! Для передачи нулевого параметра (n=0) следует использовать строковый ноль (\x30), т.к. цифровой(\x00) будет интерпретирован как конец строки и не будет передан принтеру.

Установка режима работы автоотрезчика

- Формат команды:
 - ASCII: GS V m
 - Hex: 1D 56 m
 - Decimal: 29 86 m
- Допустимые значения:
 - m=0,1,48,49;
- Описание:
 - устанавливает полный или частичный автоотрез.

Таблица 8. Команды ESC POS. Параметры команды «Установка режима работы автоотрезчика»

Значение m	Функция
0,48	Полный отрез
1,49	Частичный отрез

Печать в буфер и протяжка бумаги

- Формат команды:
 - ASCII: LF
 - Hex: 0A
 - Decimal: 10
- Описание:
 - печатает данные в буфер печати и протягивает бумагу на одну строку.

Печать в буфер без протяжки бумаги

- Формат команды:
 - ASCII: CR
 - Hex: 0D
 - Decimal: 13
- Описание:
 - печатает данные в буфер печати.

Установка режима печати символов

- Формат команды:
 - ASCII: ESC ! n
 - Hex: 1B 21 n
 - Decimal: 27 33 n
- Допустимые значения:
 - $0 \leq n \leq 255$;
- Начальное значение:
 - $n = 0$;
- Описание:
 - устанавливает размер шрифта, устанавливает/отменяет двойную высоту/ширину символов. Значение n задается исходя из таблицы:

Таблица 9. Команды ESC POS. Параметры команды «Установка режима печати символов»

Бит	Значение	Назначение
0	0	Для печати используется шрифт А (12x24)
	1	Для печати используется шрифт В (9x17)
1	0	Игнорируется
	1	Сброс всех изменений (остальные биты при этом игнорируются)
2	0	Зарезервировано
3	0	Не использовать жирный шрифт
	1	Использовать жирный шрифт
4	0	Не использовать двойную высоту символов
	1	Использовать двойную высоту символов
5	0	Не использовать двойную ширину символов
	1	Использовать двойную ширину символов
6	0	Зарезервировано
7	0	Не использовать режим 0x39
	1	Использовать режим 0x39(биты 0,3,4 и 5 установлены в 1), при использовании этого бита, все остальные биты изменения режима печати символов игнорируются

ВНИМАНИЕ! Для отмены изменений следует передавать значение 1 во втором бите ($n=0x02$), т.к. значение $n=0$ будет интерпретировано как конец строки и не передано принтеру.

Инициализация сигнала денежному ящику

- Формат команды:
 - ASCII: ESC BEL n m
 - Hex: 1B 07 n m
 - Decimal: 27 07 n m
- Допустимые значения:
 - $0 \leq n \leq 255$; $0 \leq m \leq 255$;
- Описание:
 - посылает денежному ящику сигнал инициализации последующего управляющего сигнала.
 - n - длительность импульса;
 - m - задержка между импульсами.

Управляющий сигнал на первый канал денежного ящика

- Формат команды:
 - ASCII: ESC 9 m n1 n2
 - Hex: 1B 39 m n1 n2
 - Decimal: 27 57 m n1 n2
- Допустимые значения:
 - $m = 0$; $0 \leq n1 \leq 255$; $0 \leq n2 \leq 255$;
- Начальное значение:
 - $n = 0$;
- Описание:
 - посылает на первый канал денежного ящика управляющий сигнал, соответствующий значениям n1 и n2.

Управляющий сигнал на второй канал денежного ящика

- Формат команды:
 - ASCII: ESC p m n1 n2
 - Hex: 1B 70 m n1 n2
 - Decimal: 27 112 m n1 n2
- Допустимые значения:
 - $m = 0$; $0 \leq n1 \leq 255$; $0 \leq n2 \leq 255$;
- Начальное значение:
 - $n = 0$;
- Описание:
 - посылает на второй канал денежного ящика управляющий сигнал, соответствующий значениям n1 и n2.

Печать штрихкода

- Формат команды:
 - ASCII: GS k m n d₁...d_n
 - Hex: 1D 6B m n d₁...d_n
 - Decimal: 29 107 m n d₁...d_n
- Допустимые значения:
 - $65 \leq m \leq 73$;
 - n и d зависят от используемого стандарта штрихкода;
- Описание:
 - печатает штрихкод заданного стандарта. Значения задаются исходя из следующей таблицы:

Таблица 10. Команды ESC POS. Параметры команды «Печать штрихкода»

Значение m	Стандарт штрихкода	Значения n	Значения d
67	JAN13 (EAN13)	$12 \leq n \leq 13$	$48 \leq d \leq 57$
68	JAN 8 (EAN8)	$7 \leq n \leq 8$	$48 \leq d \leq 57$
69	CODE39	$1 \leq n \leq 255$	$48 \leq d \leq 57, 65 \leq d \leq 90, 32, 36, 37, 43, 45, 46, 47$
73	CODE39	$2 \leq n \leq 255$	$0 \leq d \leq 127$

Приложение Г. Специфические команды DirectIO

Описание как специфической команды без параметров, так и специфической команды с параметрами может быть получено с помощью команды «Получить описание для специфической команды»

Специфические команды без параметров

Таблица 11. Поддержка устройствами специфических команд без параметров.

Название команды	Поддержка устройством					
	MSTAR-TK	MSTAR-TUP-K	MSTAR-Kiosk	МЭР MSTAR	MSPrint-80K	МЭР MPrint-80
Считать данные фискального накопителя	—	—	—	—	—	—
Считать программу фискального накопителя	+	+	+	+	+	+
Считать дескриптор фискального накопителя	+	+	+	+	+	+
Считать программу процессора	+	+	+	+	+	+
Печать Z-отчета из буфера	—	+	+	—	—	—
Печать оперативного отчета налогового инспектора	—	+	+	—	—	—
Активизация ЭКЛЗ	+	+	+	—	+	—
Закрытие архива ЭКЛЗ	+	+	+	—	+	—
Итог активизации ЭКЛЗ	+	+	+	—	+	—
Регистрация пользователя	+	+	+	+	+	+
Запись в устройство	+	+	+	+	+	+
Запрос данных	+	+	+	+	+	+
Установка функции порта	+	+	+	+	+	+
Аварийное закрытие смены	+	+	+	—	+	—
Запись логотипа	+	+	+	+	+	+
Перезагрузка	+	+	+	+	+	+
Запрос данных из ЭКЛЗ	+	+	+	—	+	—

Примечание: Применительно к МЭР «MSTAR», под термином «фискальный накопитель» подразумевается память сменных итогов, а под термином «налоговый инспектор» подразумевается владелец МЭР.

Считать данные фискального накопителя

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	0x00000006	Константа
<i>pData</i>	Параметр команды	0	Константа
<i>pString</i>	Сохранить в файл...	Путь к файлу	—

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>6</Cmd>
  <Name>Считать данные фискального накопителя</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">6</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemPath">
      <Value xsi:type="xsd:string"/>
      <Flag>0</Flag>
      <Name>Сохранить в файл...</Name>
    </ItemBase>
  </Items>
</DirectIOCmd>
```


Считать программу фискального накопителя

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	0x00000007	Константа
<i>pData</i>	Параметр команды	0	Константа
<i>pString</i>	Сохранить в файл...	Путь к файлу	—

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>7</Cmd>
  <Name>Считать программу фискального накопителя</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">7</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemPath">
      <Value xsi:type="xsd:string"/>
      <Flag>0</Flag>
      <Name>Сохранить в файл...</Name>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Считать дескриптор фискального накопителя

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	0x00000008	Константа
<i>pData</i>	Параметр команды	0	Константа
<i>pString</i>	Сохранить в файл...	Путь к файлу	—

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>8</Cmd>
  <Name>Считать дескриптор фискального накопителя</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">8</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemPath">
      <Value xsi:type="xsd:string"/>
      <Flag>0</Flag>
      <Name>Сохранить в файл...</Name>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Считать программу процессора

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	0x00000009	Константа
<i>pData</i>	Параметр команды	0	Константа
<i>pString</i>	Сохранить в файл...	Путь к файлу	—

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>9</Cmd>
  <Name>Считать программу процессора</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">9</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemPath">
      <Value xsi:type="xsd:string"/>
      <Flag>0</Flag>
      <Name>Сохранить в файл...</Name>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Печать Z-отчета из буфера

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	0x0000000A	Константа
<i>pData</i>	Параметр команды	0	Константа
<i>pString</i>	Строка	"" (пустая строка)	Константа

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>10</Cmd>
  <Name>Печать Z-отчета из буфера</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">10</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Строка</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Печать оперативного отчета налогового инспектора

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	0x0000000B	Константа
<i>pData</i>	Параметр команды	0	Константа
<i>pString</i>	Строка	"" (пустая строка)	Константа

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>11</Cmd>
  <Name>Печать оперативного отчета налогового инспектора</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">11</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Строка</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Активизация ЭКЛЗ

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x0000000C</i>	Константа
<i>pData</i>	Параметр команды	<i>0</i>	Константа
<i>pString</i>	Строка	<i>""</i> (пустая строка)	Константа

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>12</Cmd>
  <Name>Активизация ЭКЛЗ</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">12</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Строка</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Заккрытие архива ЭКЛЗ

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x0000000D</i>	Константа
<i>pData</i>	Параметр команды	<i>0</i>	Константа
<i>pString</i>	Строка	<i>""</i> (пустая строка)	Константа

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>13</Cmd>
  <Name>Заккрытие архива ЭКЛЗ</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">13</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Строка</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Итог активизации ЭКЛЗ

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x0000000E</i>	Константа
<i>pData</i>	Параметр команды	<i>0</i>	Константа
<i>pString</i>	Строка	<i>"" (пустая строка)</i>	Константа

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>14</Cmd>
  <Name>Итог активизации ЭКЛЗ</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">14</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Строка</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```


Регистрация пользователя

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x0000000F</i>	Константа
<i>pData</i>	Номер пользователя	<i>Номер пользователя</i>	Константа
<i>pString</i>	Пароль пользователя	<i>Пароль пользователя</i>	Константа

Номер пользователя может принимать значения:

- 0 – налоговый инспектор.

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>15</Cmd>
  <Name>Регистрация пользователя</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">15</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Номер пользователя</Name>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemPassword">
      <Value xsi:type="xsd:string"/>
      <Flag>4</Flag>
      <Name>Пароль пользователя</Name>
      <MaxLength>5</MaxLength>
      <IsConfirm>>false</IsConfirm>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Запись в устройство

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000010</i>	Константа
<i>pData</i>	Код устройства	<i>Код устройства</i>	Константа
<i>pString</i>	Данные	<i>Данные</i>	Константа

Код устройства может принимать значения:

- 0 – прямая запись в принтер. Данные передаются в шестнадцатеричном виде в строковом представлении. Например, команда: *0x1b 0x64 0x03* должна быть передана в виде строки «*1b6403*». Список разрешенных команд приведен в Приложении Б.

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>16</Cmd>
  <Name>Запись в устройство</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">16</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Код устройства</Name>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Данные</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Запрос данных

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000011</i>	Константа
<i>pData</i>	Код данных	<i>Код данных</i>	Константа
<i>pString</i>	Данные	<i>""</i> (пустая строка)	Константа

Код данных может принимать значения:

- *0x00000001* – счетчик: Нарастающий итог
- *0x00000002* – счетчик: Сумма наличных
- *0x00000003* – счетчик: Осталось записей в буфере Z-отчетов
- *0x00000004* – счетчик: Количество записей в буфере Z-отчетов
- *0x00000005* – данные: Установленный лицензионный ключ

Если данные считаны успешно, то в *pData* будет возвращено тоже значение, что и передано, в *pString* – данные в соответствии с кодом, иначе в *pData* будет возвращен ноль.

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>17</Cmd>
  <Name>Запрос данных</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">17</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Код данных</Name>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Данные</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Установка функции порта

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000012</i>	Константа
<i>pData</i>	Номер порта	<i>Номер порта</i>	См. описание
<i>pString</i>	Функция порта	<i>Функция порта</i>	См. описание

Номер порта может принимать значения:

- *0x00000000* – нет порта;
- *0x00000001* – USB-CDC;
- *0x00000002* – COM1;
- *0x00000003* – USB-HID.

Функция порта может принимать значения:

- *"VOID"* – нет функции;
- *"MMSK"* – функция – протокол Меркурий-MSK;
- *"SHTRIH"* – функция – протокол Штрих-M;
- *"MSTAR"* – функция – протокол MSTAR;
- *"BRIDGE"* – функция – мост USB-CDC – COM2.

Если команда выполнена успешно, то на указанном порту устанавливается соответствующая параметру функция. Установка перманентна и не сбрасывается при выключении питания. Описание портов и можно найти в структурной схеме фискальной платы.

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns: xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>18</Cmd>
  <Name>Установка функции порта</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">18</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Номер порта</Name>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Функция порта</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Аварийное закрытие смены

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	0x00000013	Константа
<i>pData</i>	Параметр команды	0	Константа
<i>pString</i>	Строка	"" (пустая строка)	Константа

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>19</Cmd>
  <Name>Аварийное закрытие смены</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">19</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Строка</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Запись логотипа

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000014</i>	Константа
<i>pData</i>	Параметр команды	<i>0</i>	Константа
<i>pString</i>	Файл логотипа...	<i>Путь к файлу</i>	Константа

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>20</Cmd>
  <Name>Запись логотипа</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">20</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Параметр команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemPath">
      <Value xsi:type="xsd:string"/>
      <Flag>0</Flag>
      <Name>Файл логотипа...</Name>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Перезагрузка

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000015</i>	Константа
<i>pData</i>	Параметр команды	<i>0xAACCCFFE</i>	Константа
<i>pString</i>	Строка	"" (пустая строка)	Константа

Если команда выполнена успешно, то произойдет перезагрузка принтера.

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>18</Cmd>
  <Name>Установка функции порта</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">18</Value>
      <Flag>0</Flag>
      <Name>Код команды</Name>
      <ReadOnly>true</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>0</Flag>
      <Name>Номер порта</Name>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>0</Flag>
      <Name>Функция порта</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Запрос данных из ЭКЛЗ

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>pData</i>	
<i>pString</i>	Строка	<i>pString</i>	

Поддерживаются следующие запросы:

- 1) Запрос состояния ЭКЛЗ;
- 2) Запрос регистрационного номера ЭКЛЗ;
- 3) Тест целостности архива ЭКЛЗ;
- 4) Запрос итога активизации;
- 5) Запрос контрольной ленты;
- 6) Запрос документа;
- 7) Запрос отчета;
- 8) Запрос итогов смены;
- 9) Получение данных по запрошенному отчету;
- 10) Завершение отчета.

Подробное описание параметров команды:

- *pData* – в этом параметре передается длина ожидаемого от ЭКЛЗ ответа (в байтах);
- *pString* – в этом параметре передается строка, содержащая сам запуск ЭКЛЗ, формируемый в соответствии со спецификацией "Электронная контрольная лента защищенная (ЭКЛЗ)";

Строка *pString* формируется последовательностью байт, определяющих тип запроса. Для всех запросов общим является следующее содержимое *pString*:

1. Стратовый байт – всегда равен *0x02*;
2. Длина сообщения – число байт в сообщении, исключая стратовый байт, контрольную сумму и байт длины сообщения;
3. Код команды – байт, определяющий тип запроса;
4. Байт контрольной суммы – формируется побайтовым исключающе ИЛИ (*Xor*) всех байт сообщения кроме стартового и байта контрольной суммы;

После получения ответа от ЭКЛЗ, строка *pString* также будет содержать последовательность байт. Для всех ответов данным случае будут общими:

1. Стратовый байт – всегда равен *0x02*;
2. Длина сообщения – число байт в сообщении, исключая стратовый байт, контрольную сумму и байт длины сообщения;
3. Код ответа – байт, отображающий результат выполнения запроса, может принимать следующие значения:
 - *0x00* – нормальное завершение;
 - *0x01* – некорректный формат или параметр команды;
 - *0x02* – некорректное состояние ЭКЛЗ;
 - *0x03* – авария ЭКЛЗ;
 - *0x04* – авария КС (Криптографического процессора) в составе ЭКЛЗ;
 - *0x05* – исчерпан временной ресурс использования ЭКЛЗ;
 - *0x06* – ЭКЛЗ переполнена;
 - *0x07* – неверные дата или время;
 - *0x08* – нет запрошенных данных;
 - *0x09* – переполнение (отрицательный итог документа, слишком много отделов для одного клиента)

Запрос состояния ЭКЛЗ

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x18</i> или <i>0x1E</i>	
<i>pString</i>	Строка	<i>pString</i>	

Формат параметра *pString* для данного запроса - "0202070104" или "0202070207", где:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x02</i>	Длина сообщения
2	<i>0x07</i>	Код команды
3	<i>0x01</i> или <i>0x02</i>	Код запроса
4	<i>0x04</i> или <i>0x07</i>	Контрольная сумма

Коды запроса:

- *0x01* – запрос итога документа, даты, времени, номера последнего КПК и номера ЭКЛЗ;
- *0x02* – запрос итогов текущей смены;

По запросу *0x01* ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x15</i>	Длина сообщения (21)
2	<i>0x00</i>	Код ответа
3	<i>0x01</i>	Итог документа (1324567.99)
4	<i>0x32</i>	
5	<i>0x45</i>	
6	<i>0x67</i>	
7	<i>0x89</i>	
8	<i>0x11</i>	Год
9	<i>0x08</i>	Месяц
10	<i>0x09</i>	День (дата 09.08.2011)
11	<i>0x13</i>	Часы
12	<i>0x32</i>	Минуты (время 13:32)
13	<i>0x01</i>	Номер КПК (01234567)
14	<i>0x23</i>	
15	<i>0x45</i>	
16	<i>0x67</i>	
17	<i>0x10</i>	Номер ЭКЛЗ (1032456789)
18	<i>0x32</i>	
19	<i>0x45</i>	
20	<i>0x67</i>	
21	<i>0x89</i>	
22		биты 0-1 – флаг <i>t</i> ; бит 2 – флаг <i>i</i> ; бит 3 – флаг <i>f</i> ; бит 4 – флаг <i>w</i> ; бит 5 – флаг <i>d</i> ; бит 6 – флаг <i>s</i> ; бит 7 – флаг <i>a</i> .
23		Контрольная сумма

Флаги состояния ЭКЛЗ:

- флаг t – тип документа:
 - 0 – продажа;
 - 1 – покупка;
 - 2 – возврат продажи;
 - 3 – возврат покупки;
- флаг i – открыт Архив;
- флаг f – выполнена активизация;
- флаг w – отчет, устанавливается успешным выполнением команд запроса отчета, итогов смены, итога активизации, контрольной ленты и документа. Сбрасывается успешным выполнением команд запроса данных отчета (когда нет данных) и прекращения, а также флагом a ;
- флаг d – открыт документ;
- флаг s – открыта смена;
- флаг a – неисправимая ошибка устройства.

По запросу $0x02$ ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	$0x02$	STX
1	$0x1B$	Длина сообщения (27)
2	$0x00$	Код ответа
3	$0x00$	Номер смены(56)
4	$0x38$	
5	$0x00$	Итог, накопленный по операциям продажи (1006789.99)
6	$0x01$	
7	$0x00$	
8	$0x67$	
9	$0x89$	
10	$0x99$	
11	$0x10$	Итог, накопленный по операциям покупки (1032456789.99)
12	$0x32$	
13	$0x45$	
14	$0x67$	
15	$0x89$	
16	$0x99$	
17	$0x00$	Итог, накопленный по операциям возврата продажи (6789.99)
18	$0x00$	
19	$0x00$	
20	$0x67$	
21	$0x89$	
22	$0x99$	
23	$0x00$	Итог, накопленный по операциям возврата покупки (6879.99)
24	$0x00$	
25	$0x00$	
26	$0x68$	
27	$0x79$	
28	$0x99$	
29		Контрольная сумма

Запрос регистрационного номера ЭКЛЗ

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x13</i> или <i>0x08</i>	
<i>pString</i>	Строка	<i>pString</i>	

Формат параметра *pString* для данного запроса - "02019495" или "02011415", где:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x01</i>	Длина сообщения
2	<i>0x94</i> или <i>0x14</i>	Код команды
3	<i>0x95</i> или <i>0x15</i>	Контрольная сумма

Коды команды:

- *0x94* – запрос номера ЭКЛЗ в формате строки (кодировка CP-866);
- *0x14* – запрос номера ЭКЛЗ в числовом формате;

По запросу *0x94* ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x11</i>	Длина сообщения (17)
2	<i>0x00</i>	Код ответа
3-18		Строка с номером ЭКЛЗ (16 символов)
19		Контрольная сумма

По запросу *0x14* ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x06</i>	Длина сообщения
2	<i>0x00</i>	Код ответа
3	<i>0x10</i>	Номер ЭКЛЗ (1032456789)
4	<i>0x32</i>	
5	<i>0x45</i>	
6	<i>0x67</i>	
7	<i>0x89</i>	
8		Контрольная сумма

Тест целостности архива ЭКЛЗ

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x04</i>	
<i>pString</i>	Строка	<i>pString</i>	

ЭКЛЗ возвращает код ошибки (в байте кода ответа) *0x03* и устанавливает флаг *a=1* при несправимых ошибках.

Формат параметра *pString* для данного запроса - "0202080109", где:

№ бита	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x02</i>	Длина сообщения
2	<i>0x08</i>	Код команды
3	<i>0x08</i>	Номер теста
4	<i>0x0B</i>	Контрольная сумма

По этому запросу ЭКЛЗ возвращает следующую структуру данных:

№ бита	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x01</i>	Длина сообщения
2	<i>0x00</i>	Код ответа
3	<i>0x01</i>	Контрольная сумма

Запрос итога активизации

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x2C</i> или <i>0x2D</i>	
<i>pString</i>	Строка	<i>pString</i>	

Формат параметра *pString* для данного запроса - "0201A4A5" или "02012425", где:

№ бита	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x01</i>	Длина сообщения
2	<i>0xA4</i> или <i>0x24</i>	Код команды
3	<i>0xA5</i> или <i>0x25</i>	Контрольная сумма

Коды команды:

- *0xA4* – запрос итога активизации ЭКЛЗ в формате строки (кодировка CP-866);
- *0x24* – запрос итога активизации ЭКЛЗ в числовом формате;

По запросу *0xA4* ЭКЛЗ возвращает следующую структуру данных:

№ бита	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x2A</i>	Длина сообщения (41)
2	<i>0x00</i>	Код ответа
3-42		Тип ККМ (40 символов)
43		Контрольная сумма

По запросу $0x24$ ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	$0x02$	STX
1	$0x29$	Длина сообщения (42)
2	$0x00$	Код ответа
3	$0x05$	Код фрагмента
4-43		Тип ККМ (40 символов)
44		Контрольная сумма

После выполнения запроса итога активизации (в любом формате) устанавливается флаг отчета $w=1$, и выполнение любой другой команды кроме "Получение данных по запрошенному отчету" и "Завершение отчета" приведет к ошибке.

Для получения дальнейшей информации по отчету требуется последовательное выполнение команды "Получение данных по запрошенному отчету" до тех пор, пока в коде ответа не появится байт $0x08$ (нет запрошенных данных). В этом случае флаг отчета $w=1$ будет сброшен автоматически.

Для сброса флага до получения всех данных по отчету требуется выполнить команду "Завершение отчета"

Запрос контрольной ленты

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	$0x00000016$	Константа
<i>pData</i>	Параметр команды	$0x2C$ или $0x2D$	
<i>pString</i>	Строка	<i>pString</i>	

Формат параметра *pString* для данного запроса:

№ байта	Значение	Комментарии
0	$0x02$	STX
1	$0x03$	Длина сообщения
2	$0xA0$ или $0x20$	Код команды
3	$0x10$	Номер смены (4130)
4	$0x22$	
5		Контрольная сумма

Коды команды:

- $0xA0$ – запрос контрольной ленты ЭКЛЗ в формате строки (кодировка CP-866);
- $0x20$ – запрос контрольной ленты ЭКЛЗ в числовом формате;

По запросу $0xA0$ ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	$0x02$	STX
1	$0x2A$	Длина сообщения (41)
2	$0x00$	Код ответа
3-42		Тип ККТ (40 символов)
43		Контрольная сумма

По запросу *0x20* ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x29</i>	Длина сообщения (42)
2	<i>0x00</i>	Код ответа
3	<i>0x05</i>	Код фрагмента
4-43		Тип ККМ (40 символов)
44		Контрольная сумма

После выполнения запроса контрольной ленты (в любом формате) устанавливается флаг отчета *w=1*, и выполнение любой другой команды кроме "Получение данных по запрошенному отчету" и "Завершение отчета" приведет к ошибке.

Для получения дальнейшей информации по отчету требуется последовательное выполнение команды "Получение данных по запрошенному отчету" до тех пор, пока в коде ответа не появится байт *0x08* (нет запрошенных данных). В этом случае флаг отчета *w=1* будет сброшен автоматически.

Для сброса флага до получения всех данных по отчету требуется выполнить команду "Завершение отчета"

Запрос документа

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x2C</i> или <i>0x2D</i>	
<i>pString</i>	Строка	<i>pString</i>	

Формат параметра *pString* для данного запроса:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x05</i>	Длина сообщения
2	<i>0xA1</i> или <i>0x21</i>	Код команды
3	<i>0x01</i>	Номер КПК (1345678)
4	<i>0x34</i>	
5	<i>0x56</i>	
6	<i>0x78</i>	
7		Контрольная сумма

Коды команды:

- *0xA1* – запрос документа в формате строки (кодировка CP-866);
- *0x21* – запрос документа в числовом формате;

По запросу *0xA1* ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x2A</i>	Длина сообщения (41)
2	<i>0x00</i>	Код ответа
3-42		Тип ККМ (40 символов)
43		Контрольная сумма

По запросу *0x21* ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x29</i>	Длина сообщения (42)
2	<i>0x00</i>	Код ответа
3	<i>0x05</i>	Код фрагмента
4-43		Тип ККМ (40 символов)
44		Контрольная сумма

После выполнения запроса документа (в любом формате) устанавливается флаг отчета $w=1$, и выполнение любой другой команды кроме "Получение данных по запрошенному отчету" и "Завершение отчета" приведет к ошибке.

Для получения дальнейшей информации по отчету требуется последовательное выполнение команды "Получение данных по запрошенному отчету" до тех пор, пока в коде ответа не появится байт *0x08* (нет запрошенных данных). В этом случае флаг отчета $w=1$ будет сброшен автоматически.

Для сброса флага до получения всех данных по отчету требуется выполнить команду "Завершение отчета"

Запрос отчета

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x2C</i> или <i>0x2D</i>	
<i>pString</i>	Строка	<i>pString</i>	

Поддерживаются следующие типы запросов отчета:

- *0x04* – полный по датам по закрытиям смен;
- *0x05* – краткий по датам по закрытиям смен;
- *0x06* – полный по номерам смен по закрытиям смен;
- *0x07* – краткий по номерам смен по закрытиям смен;

Формат параметра *pString* для запроса отчета по датам:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x08</i>	Длина сообщения
2	<i>0xA2</i> или <i>0x22</i>	Код команды
3	<i>0x04</i>	Код отчета (полный по датам)
4	<i>0x01</i>	Год
5	<i>0x01</i>	Месяц
6	<i>0x11</i>	День (дата начала отчета 01.01.2011)
7	<i>0x01</i>	Год
8	<i>0x08</i>	Месяц
9	<i>0x11</i>	День (дата окончания отчета 01.08.2011)
10		Контрольная сумма

Формат параметра *pString* для запроса отчета по номерам смен:

№ байта	Значение	Комментарии
0	0x02	STX
1	0x06	Длина сообщения
2	0xA2 или 0x22	Код команды
3	0x06	Код отчета (полный по номерам)
4	0x00	Номер смены на начала отчета (1)
5	0x01	
6	0x00	Номер смены на окончания отчета (17
7	0x11	
8		Контрольная сумма

Коды команды:

- 0xA2 – запрос отчета в формате строки (кодировка CP-866);
- 0x22 – запрос отчета в числовом формате;

По запросу 0xA2 ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	0x02	STX
1	0x2A	Длина сообщения (41)
2	0x00	Код ответа
3-42		Тип ККМ (40 символов)
43		Контрольная сумма

По запросу 0x21 ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	0x02	STX
1	0x29	Длина сообщения (42)
2	0x00	Код ответа
3	0x05	Код фрагмента
4-43		Тип ККМ (40 символов)
44		Контрольная сумма

После выполнения запроса отчета(в любом формате) устанавливается флаг отчета $w=1$, и выполнение любой другой команды кроме "Получение данных по запрошенному отчету" и "Завершение отчета" приведет к ошибке.

Для получения дальнейшей информации по отчету требуется последовательное выполнение команды "Получение данных по запрошенному отчету" до тех пор, пока в коде ответа не появится байт 0x08 (нет запрошенных данных). В этом случае флаг отчета $w=1$ будет сброшен автоматически.

Для сброса флага до получения всех данных по отчету требуется выполнить команду "Завершение отчета"

Запрос итогов смен

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x2C</i> или <i>0x2D</i>	
<i>pString</i>	Строка	<i>pString</i>	

Формат параметра *pString* для данного запроса:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x03</i>	Длина сообщения
2	<i>0xA3</i> или <i>0x23</i>	Код команды
3	<i>0x01</i>	Номер смены начала отчета (273)
4	<i>0x11</i>	
5		Контрольная сумма

Коды команды:

- *0xA3* – запрос отчета в формате строки (кодировка CP-866);
- *0x23* – запрос отчета в числовом формате;

По запросу *0xA2* ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x2A</i>	Длина сообщения (41)
2	<i>0x00</i>	Код ответа
3-42		Тип КKM (40 символов)
43		Контрольная сумма

По запросу *0x21* ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x29</i>	Длина сообщения (42)
2	<i>0x00</i>	Код ответа
3	<i>0x05</i>	Код фрагмента
4-43		Тип КKM (40 символов)
44		Контрольная сумма

После выполнения запроса итогов смен (в любом формате) устанавливается флаг отчета $w=1$, и выполнение любой другой команды кроме "Получение данных по запрошенному отчету" и "Завершение отчета" приведет к ошибке.

Для получения дальнейшей информации по отчету требуется последовательное выполнение команды "Получение данных по запрошенному отчету" до тех пор, пока в коде ответа не появится байт *0x08* (нет запрошенных данных). В этом случае флаг отчета $w=1$ будет сброшен автоматически.

Для сброса флага до получения всех данных по отчету требуется выполнить команду "Завершение отчета"

Получение данных по запрошенному отчету

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x2C</i>	
<i>pString</i>	Строка	<i>pString</i>	

Формат параметра *pString* для данного запроса "02010506", где::

№ байта	Значение	Комментарии
0	0x02	STX
1	0x01	Длина сообщения
2	0x05	Код команды
5	0x06	Контрольная сумма

Для отчета в текстовой форме, ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	0x02	STX
1	0x2A	Длина сообщения (41)
2	0x00	Код ответа
3-42		Тип ККМ (40 символов)
43		Контрольная сумма

Для отчета в числовой форме, ЭКЛЗ может вернуть различные ответы, в зависимости от запрашиваемого отчета и кода фрагмента.

На каждый последующий вызов команды "Получение данных по запрошенному отчету" ЭКЛЗ отвечает очередным блоком информации (строкой для текстового запроса или фрагментом для числового запроса).

Возможны следующие фрагменты отчета:

1. Фрагмент "Заголовок документа"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x09	Длина сообщения (9)
2	0x00	Код ответа
3	0x01	Код фрагмента
4	0x10	Код команды (0x10 - продажа, 0x12 - возврат продажи)
5	0x11	Год
6	0x08	Месяц
7	0x09	День (дата 09.08.2011)
8	0x15	Часы
9	0x30	Минуты (время 15:30)
10	0x63	Код оператора (99)
11		Контрольная сумма

2. Фрагмент "Отдел"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x0C	Длина сообщения (12)
2	0x00	Код ответа
3	0x02	Код фрагмента
4	0x0A	Кодотдела (10)
5	0x10	Количество (10324.567)
6	0x32	
7	0x45	
8	0x67	
9	0x15	Сумма (15334567.00)
10	0x33	
11	0x45	
12	0x67	
13	0x00	
14		Контрольная сумма

3. Фрагмент "Итог документа"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x07	Длина сообщения (7)
2	0x00	Код ответа
3	0x06	Код фрагмента
4	0x00	Сумма (324567.89)
5	0x32	
6	0x45	
7	0x67	
8	0x89	
9		Контрольная сумма

4. Фрагмент "Заголовок ККМ"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x19	Длина сообщения (25)
2	0x00	Код ответа
3	0x07	Код фрагмента
4-15	0x00, ..., 0x12, 0x31, 0x23	Заводской номер ККМ (00000000000000000000123123)
16	0x10	Регистрационный номер ЭКЛЗ (1032456789)
17	0x32	
18	0x45	
19	0x67	
20	0x89	
21	0x11	ИНН (112233445566)
22	0x22	
23	0x33	
24	0x44	
25	0x55	
26	0x66	
27		Контрольная сумма

5. Фрагмент "Активизация"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x0A	Длина сообщения (15)
2	0x00	Код ответа
3	0x08	Код фрагмента
4	0x11	Год
5	0x08	Месяц
6	0x07	День (дата 07.08.2011)
7	0x10	Часы
8	0x30	Минуты (время 10:30)
9	0x11	Заводской номер ККМ (112233445566)
10	0x22	
11	0x33	
12	0x44	
13	0x55	
14	0x66	
15	0x00	Номер смены (56)
16	0x38	
17		Контрольная сумма

6. Фрагмент "Закрытие смены"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x1A	Длина сообщения (26)
2	0x00	Код ответа
3	0x09	Код фрагмента
4	0x00	Сумма продажи (1006789.99)
5	0x01	
6	0x00	
7	0x67	
8	0x89	
9	0x99	
10-15	0x00 . . . 0x00	Сумма покупки (0000000000.00)
16	0x00	Сумма возврата продажи (6789.99)
17	0x00	
18	0x00	
19	0x67	
20	0x89	
21	0x99	
22-27	0x00 . . . 0x00	Сумма возврата покупки (0000000000.00)
28		Контрольная сумма

7. Фрагмент "Итог закрытия смены"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x1A	Длина сообщения (26)
2	0x00	Код ответа
3	0x0A	Код фрагмента
4	0x00	Сумма продажи (1006789.99)
5	0x01	
6	0x00	
7	0x67	
8	0x89	
9	0x99	
10-15	0x00...0x00	Сумма покупки (0000000000.00)
16	0x00	Сумма возврата продажи (6789.99)
17	0x00	
18	0x00	
19	0x67	
20	0x89	
21	0x99	
22-27	0x00...0x00	Сумма возврата покупки (0000000000.00)
28		Контрольная сумма

8. Фрагмент "Строка КПК"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x09	Длина сообщения (9)
2	0x00	Код ответа
3	0x0B	Код фрагмента
4	0x01	Номер КПК (1345678)
5	0x34	
6	0x56	
7	0x78	
8	0x01	Значение КПК (12345)
9	0x23	
10	0x45	
11		Контрольная сумма

9. Фрагмент "Заголовок смены"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x0A	Длина сообщения (10)
2	0x00	Код ответа
3	0x0E	Код фрагмента
4	0x00	Номер смены (56)
5	0x38	
6	0x11	Год
7	0x08	Месяц
8	0x09	День (дата 09.08.2011)
9	0x22	Часы
10	0x00	Минуты (время 22:00)
11	0x63	Код оператора (99)
12		Контрольная сумма

10. Фрагмент "Информация о незавершенной смене"

№ байта	Значение	Комментарии
0	0x02	STX
1	0x1A	Длина сообщения (26)
2	0x00	Код ответа
3	0x0A	Код фрагмента
4	0x00	Номер смены (34)
5	0x22	
6	0x00	Сумма продажи (1006789.99)
7	0x01	
8	0x00	
9	0x67	
10	0x89	
11	0x99	
12-17	0x00...0x00	Сумма покупки (0000000000.00)
18	0x00	Сумма возврата продажи (6789.99)
19	0x00	
20	0x00	
21	0x67	
22	0x89	
23	0x99	
24-29	0x00...0x00	Сумма возврата покупки (0000000000.00)
30		Контрольная сумма

Команду "Получение данных по запрошенному отчету" нужно выполнять после запроса какого-либо отчета производить вызов этой команды до тех пор, пока в коде ответа не появится байт 0x08 (нет запрошенных данных)– в этом случае флаг отчета $w=1$ будет сброшен автоматически.

Для сброса флага до получения всех данных по отчету требуется выполнить команду "Завершение отчета"

Прекращение

Параметр	Название	Значение	Примечание
<i>Command</i>	Код команды	<i>0x00000016</i>	Константа
<i>pData</i>	Параметр команды	<i>0x04</i>	
<i>pString</i>	Строка	<i>pString</i>	

Формат параметра *pString* для данного запроса "02010506", где::

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x01</i>	Длина сообщения
2	<i>0x06</i>	Код команды
5	<i>0x07</i>	Контрольная сумма

На эту команду ЭКЛЗ возвращает следующую структуру данных:

№ байта	Значение	Комментарии
0	<i>0x02</i>	STX
1	<i>0x01</i>	Длина сообщения
2	<i>0x00</i>	Код ответа
3		Контрольная сумма

Команда используется для завершения запрошенного в ЭКЛЗ отчета (флаг отчета *w* будет сброшен)

Специфические команды с параметрами

Таблица 12. Поддержка устройствами специфических команд с параметрами.

Название команды	Поддержка устройством					
	MSTAR-TK	MSTAR-TUP-K	MSTAR-Kiosk	МЭР MSTAR	MSPrint-80K	МЭР MPrint-80
Программирование серийного номера	+	+	+	+	+	+
Фискализация/Перерегистрация	+	+	+	+	+	+
Инициализация Фискального накопителя	+	+	+	+	+	+
Отчет ЭКЛЗ по итогам смен в диапазоне дат	+	+	+	—	+	—
Отчет ЭКЛЗ по итогам смен в диапазоне номеров	+	+	+	—	+	—
Отчет ЭКЛЗ «Документ по номеру КПК»	+	+	+	—	+	—
Отчет ЭКЛЗ «Контрольная лента по номеру смены»	+	+	+	—	+	—
Отчет ЭКЛЗ «Итоги смены по номеру смены»	+	+	+	—	+	—
Установить пароль на связь	+	+	+	+	+	+
Установить лицензии	—	+	+	—	—	—

Примечание: Применительно к МЭР «MSTAR», под термином «фискальный накопитель» подразумевается память сменных итогов.

Программирование серийного номера

- Код команды: `0x101`
- Комбинация флагов: `0x1`
- Параметры:

Флаг	Название	Тип	Max length	Read only
1	Серийный номер	<i>ItemString</i>	12	<i>FALSE</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>257</Cmd>
  <Name>Программирование серийного номера</Name>
  <Items>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>1</Flag>
      <Name>Серийный номер</Name>
      <MaxLength>12</MaxLength>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```


Фискализация/Перерегистрация

- Код команды: *0x102*
- Комбинация флагов: *0x7*
- Параметры:

Флаг	Название	Тип	Max length	Read only
1	ИНН	<i>ItemString</i>	12	<i>FALSE</i>
2	Регистрационный номер	<i>ItemString</i>	12	<i>FALSE</i>
4	Новый пароль НИ	<i>ItemPassword</i>	5	

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>258</Cmd>
  <Name>Фискализация/Перерегистрация</Name>
  <Items>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>1</Flag>
      <Name>ИНН</Name>
      <MaxLength>12</MaxLength>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>2</Flag>
      <Name>Регистрационный номер</Name>
      <MaxLength>12</MaxLength>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemPassword">
      <Value xsi:type="xsd:string"></Value>
      <Flag>4</Flag>
      <Name>Новый пароль НИ</Name>
      <MaxLength>5</MaxLength>
      <IsConfirm>true</IsConfirm>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Инициализация Фискального накопителя

- Код команды: *0x104*
- Комбинация флагов: *0x0*
- Параметров нет

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>260</Cmd>
  <Name>Инициализация Фискального накопителя</Name>
  <Items>
  </Items>
</DirectIOCmd>
```

Отчет ЭКЛЗ по итогам смен в диапазоне дат

- Код команды: *0x106*
- Комбинация флагов: *0x7*
- Параметры:

Флаг	Название	Тип
1	Краткий/полный	<i>ItemBool</i>
2	Дата начала отчета	<i>ItemDateTime</i>
4	Дата конца отчета	<i>ItemDateTime</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>262</Cmd>
  <Name>Отчет ЭКЛЗ по итогам смен в диапазоне дат</Name>
  <Items>
    <ItemBase xsi:type="ItemBool">
      <Value xsi:type="xsd:boolean">false</Value>
      <Flag>1</Flag>
      <Name>Краткий/полный</Name>
    </ItemBase>
    <ItemBase xsi:type="ItemDateTime">
      <Value xsi:type="xsd:dateTime">2010-02-01T00:00:00</Value>
      <Flag>2</Flag>
      <Name>Дата начала отчета</Name>
    </ItemBase>
    <ItemBase xsi:type="ItemDateTime">
      <Value xsi:type="xsd:dateTime">2010-02-01T00:00:00</Value>
      <Flag>4</Flag>
      <Name>Дата конца отчета</Name>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Отчет ЭКЛЗ по итогам смен в диапазоне номеров

- Код команды: *0x107*
- Комбинация флагов: *0x7*
- Параметры:

Флаг	Название	Тип	ReadOnly
1	Краткий/полный	<i>ItemBool</i>	
2	Дата начала отчета	<i>ItemLong</i>	<i>false</i>
4	Дата конца отчета	<i>ItemLong</i>	<i>false</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>263</Cmd>
  <Name>Отчет ЭКЛЗ по итогам смен в диапазоне номеров</Name>
  <Items>
    <ItemBase xsi:type="ItemBool">
      <Value xsi:type="xsd:boolean">>false</Value>
      <Flag>1</Flag>
      <Name>Краткий/полный</Name>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">1</Value>
      <Flag>2</Flag>
      <Name>Дата начала отчета</Name>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">1</Value>
      <Flag>4</Flag>
      <Name>Дата конца отчета</Name>
      <ReadOnly>>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Отчет ЭКЛЗ Документ по номеру КПК

- Код команды: *0x108*
- Комбинация флагов: *0x1*
- Параметры:

Флаг	Название	Тип	ReadOnly
<i>1</i>	Номер КПК	<i>ItemLong</i>	<i>false</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>264</Cmd>
  <Name>Отчет ЭКЛЗ Документ по номеру КПК</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>1</Flag>
      <Name>Номер КПК</Name>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Отчет ЭКЛЗ Контрольная лента по номеру смены

- Код команды: *0x109*
- Комбинация флагов: *0x1*
- Параметры:

Флаг	Название	Тип	ReadOnly
<i>1</i>	Номер смены	<i>ItemLong</i>	<i>false</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>265</Cmd>
  <Name>Отчет ЭКЛЗ Документ по номеру смены</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>1</Flag>
      <Name>Номер смены</Name>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Отчет ЭКЛЗ Итоги смены по номеру смены

- Код команды: *0x10B*
- Комбинация флагов: *0x1*
- Параметры:

Флаг	Название	Тип	ReadOnly
<i>1</i>	Номер смены	<i>ItemLong</i>	<i>false</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>267</Cmd>
  <Name>Отчет ЭКЛЗ Итоги смены по номеру смены</Name>
  <Items>
    <ItemBase xsi:type="ItemLong">
      <Value xsi:type="xsd:int">0</Value>
      <Flag>1</Flag>
      <Name>Номер смены</Name>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Установить пароль на связь

- Код команды: *0x10C*
- Комбинация флагов: *0x3*
- Параметры:

Флаг	Название	Тип	ReadOnly
1	Старый пароль	<i>ItemPassword</i>	<i>false</i>
2	Новый пароль	<i>ItemPassword</i>	<i>false</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>268</Cmd>
  <Name>Пароль на связь</Name>
  <Items>
    <ItemBase xsi:type="ItemPassword">
      <Value xsi:type="xsd:string"></Value>
      <Flag>1</Flag>
      <Name>Старый пароль</Name>
      <MaxLength>5</MaxLength>
      <IsConfirm>false</IsConfirm>
    </ItemBase>
    <ItemBase xsi:type="ItemPassword">
      <Value xsi:type="xsd:string"></Value>
      <Flag>2</Flag>
      <Name>Новый пароль</Name>
      <MaxLength>5</MaxLength>
      <IsConfirm>true</IsConfirm>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Установка лицензии

- Код команды: *0x10A*
- Комбинация флагов: *0x1*
- Параметры:

Флаг	Название	Тип	MaxLength	ReadOnly
<i>1</i>	Лицензионный ключ	<i>ItemString</i>	<i>0</i>	<i>false</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>266</Cmd>
  <Name>Установка лицензии</Name>
  <Items>
    <ItemBase xsi:type="ItemString">
      <Value xsi:type="xsd:string"></Value>
      <Flag>1</Flag>
      <Name>Лицензионный ключ</Name>
      <MaxLength>0</MaxLength>
      <ReadOnly>false</ReadOnly>
    </ItemBase>
  </Items>
</DirectIOCmd>
```

Типы оплат

- Код команды: *0x10D*
- Комбинация флагов: *0x1*
- Параметры:

Флаг	Название	Тип	MaxLength	ReadOnly
<i>1</i>	Наименование типа оплаты	<i>ArrayOfString</i>	<i>40</i>	<i>false</i>

Строка XML с описанием команды:

```
<?xml version="1.0" encoding="utf-8"?>
<DirectIOCmd
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Cmd>269</Cmd>
  <Name>Типы оплат</Name>
  <Items>
    <ItemBase xsi:type="ItemStringCollection">
      <Value xsi:type="ArrayOfString">
        <string>НАЛИЧНЫМИ</string>
        <string></string>
        <string></string>
        <string></string>
      </Value>
      <Flag>1</Flag>
      <Name>Наименование типа оплаты</Name>
      <MaxLength>40</MaxLength>
    </ItemBase>
  </Items>
</DirectIOCmd>
```


Список таблиц

Таблица 1. Пречень возможностей OPOS-драйвера.....	11
Таблица 2. Перечень свойств OPOS-драйвера.....	13
Таблица 3. Допустимые значения FiscalReceiptType для различных устройств.....	31
Таблица 4. Пречень методов OPOS Fiscal Printer.....	41
Таблица 5. Модели Status, Result code и State.....	73
Таблица 6. Разрешенные команды Star Line Mode.....	79
Таблица 7. Разрешенные команды ESC/POS.....	83
Таблица 8. Команды ESC POS. Параметры команды «Установка режима работы автоотрезчика».....	83
Таблица 9. Команды ESC POS. Параметры команды «Установка режима печати символов».....	84
Таблица 10. Команды ESC POS. Параметры команды «Печать штрихкода».....	86
Таблица 11. Поддержка устройствами специфических команд без параметров.....	87
Таблица 12. Поддержка устройствами специфических команд с параметрами.....	120

Методы и свойства

A

ActualCurrency	24
AdditionalHeader	24
AdditionalTrailer	25
AmountDecimalPlaces	25
AsyncMode	25

B

BeginFiscalDocument	57
BeginFiscalReceipt	24, 51
BeginFixedOutput	17, 59
BeginInsertion	59
BeginItemList	58
BeginNonFiscal	59
BeginRemoval	59
BeginTraining	59
BinaryConversion	25, 71

C

CapAdditionalHeader	15
CapAdditionalLines	15
CapAdditionalTrailer	15
CapAmountAdjustment	15
CapAmountNotPaid	15
CapChangeDue	15
CapCheckTotal	16
CapCompareFirmwareVersion	16
CapCoverSensor	16, 27
CapDoubleWidth	16
CapDuplicateReceipt	16, 29
CapEmptyReceiptIsVoidable	16
CapFiscalReceiptStation	16
CapFiscalReceiptType	17
CapFixedOutput	17

CapHasVatTable	17, 51
CapIndependentHeader	17
CapItemList	17
CapJmPresent	18
CapJmEmptySensor	17, 31
CapJmNearEndSensor	17, 32
CapMultiContractor	18, 26
CapNonFiscalMode	18
CapOnlyVoidLastItem	18
CapOrderAdjustmentFirst	18
CapPackageAdjustment	18, 55
CapPercentAdjustment	18
CapPositiveAdjustment	19
CapPositiveSubtotalAdjustment	19
CapPostPreLine	19
CapPowerLossReport	19
CapPowerReporting	19, 34
CapPredefinedPaymentLines	19
CapReceiptNotPaid	20
CapRecEmptySensor	20, 36
CapRecNearEndSensor	20, 36
CapRecPresent	20
CapRemainingFiscalMemory	20
CapReservedWord	20
CapSetCurrency	20, 24
CapSetHeader	21
CapSetPOSID	21
CapSetStoreFiscalID	21
CapSetTrailer	21
CapSetVatTable	21
CapSlpEmptySensor	21, 38
CapSlpFiscalDocument	21
CapSlpFullSlip	22
CapSlpNearEndSensor	22, 38
CapSlpPresent	22
CapSlpValidation	22

CapStatisticsReporting	22
CapSubAmountAdjustment	22
CapSubPercentAdjustment	23
CapSubtotal	23
CapTotalizerType	23
CapTrainingMode	23
CapUpdateFirmware	23
CapUpdateStatistics	23
CapValidateJournal	24
CapXReport	24
ChangeDue	26
CheckHealth	43
CheckHealthText	26
CheckTotal	26, 56-57
ClaimDevice	43, 71-72, 76
Claimed	26, 71
ClearError	62
ClearOutput	35, 43, 78
Close	43, 71-72
CompareFirmwareVersion	49
ContractorID	26
ControlObjectDescription	27, 72
ControlObjectVersion	27, 72
CountryCode	27
CoverOpen	27

D

DateType	28
DayOpened	28
DescriptionLength	28
DeviceDescription	28
DeviceEnabled	29, 34, 37, 71-72
DeviceName	29, 43
DuplicateReceipt	29

E

EndFiscalDocument	57
EndFiscalReceipt	25, 51
EndFixedOutput	59
EndInsertion	59
EndItemList	58
EndNonFiscal	60
EndRemoval	59
EndTraining	60
ErrorLevel	29
ErrorOutID	29
ErrorState	30
ErrorStation	30
ErrorString	30

F

FiscalReceiptStation	30
FiscalReceiptType	31, 52
FlagWhenIdle	31
FreezeEvents	31, 77

G

GetData	35, 60
GetDate	28, 35, 61
GetTotalizer	61
GetVatEntry	35, 61

J

JmEmpty	31
JmNearEnd	32

M

MessageLength	32
MessageType	32

N

NumHeaderLines	33
----------------	----

NumTrailerLines	33	PrintRecSubtotalAdjustVoid	56
NumVatRates	33	PrintRecTaxID	56
O		PrintRecTotal	15, 34, 36, 57
Open	33, 43, 71-72	PrintRecVoid	57
OpenResult	33, 71	PrintRecVoidItem	57
OutputID	33, 78	PrintReport	58
P		PrintXReport	24, 58
		PrintZReport	36, 59
PostLine	34, 79, 83	Q	
PowerNotify	34	QuantityDecimalPlaces	35, 52-53
PowerState	34	QuantityLength	36, 52-53
PredefinedPaymentLines	19, 34	R	
PreLine	35, 79, 83	RecEmpty	36
PrintDuplicateReceipt	52	RecNearEnd	36
PrinterState	35	ReleaseDevice	43, 71, 76
PrintFiscalDocumentLine	57	RemainingFiscalMemory	36
PrintFixedOutput	60	ReservedWord	36
PrintNormal	24-25, 60, 79, 83	ResetPrinter	35, 62
PrintPeriodicTotalsReport	58	ResetStatistics	49
PrintPowerLossReport	19, 58	ResultCodeExtended	37, 73
PrintRecCash	52	RetrieveStatistics	49
PrintRecItem	18, 25, 52	S	
PrintRecItemAdjustment	18-19, 25, 53	ServiceObjectDescription	37
PrintRecItemAdjustmentVoid	54	ServiceObjectVersion	37
PrintRecItemFuel	54	SetCurrency	20, 24, 49
PrintRecItemFuelVoid	54	SetDate	49
PrintRecItemVoid	53	SetHeaderLine	21, 50
PrintRecMessage	32, 54, 79, 83	SetPOSID	21, 50
PrintRecNotPaid	20, 35, 54	SetStoreFiscalID	21, 50
PrintRecPackageAdjustment	54	SetTrailerLine	21, 50
PrintRecPackageAdjustVoid	55	SetVatTable	21, 51
PrintRecRefund	55	SetVatValue	21, 51
PrintRecRefundVoid	55	SlipSelection	38
PrintRecSubtotal	23, 56	SlpEmpty	38
PrintRecSubtotalAdjustment	19, 56		

SlpNearEnd		38
State		38, 72-73
	T	
TotalizerType		39
TrainingModeActive		39
	U	
UpdateFirmware		49
UpdateStatistics		49
	V	
VerifyItem		58

ТЕЛЕФОН

(495)-770-18-94
(495)-770-18-95
(495)-301-05-93

ФАКС

8-495-301-41-73

ВЕБ

WWW.MULTISOFT.RU