

# FP3530

(ver.3.0.0.7 – beta version)

## Ръководство да употреба

### История на продукта

### Версия 3.0.0.7 (05/2019)

Описание	
✓ Добавяне на поддръжка за нови модели	
✓ Логическа група A	FP-700
✓ Логическа група C	DP-25X, FMP-55X, FMP-350X, FP-700X, WP-500X, WP-50X, DP-150X, DP-05C
✓ Поправка и корекции	
✓ Логическа група „A“	
✓ 107_items_Set_ItemWithReplace	Wrong delimiter
✓ 058_receipt_PLUxxx_xxx	Възможност за отрицателни стойности
✓ 119_klen_Find_Document	Коригиран грешен синтаксис
✓ 119_klen_Set_DocsRange	Коригиран грешен синтаксис
✓ 119_klen_Set_DocsRange_InZReport_01	Коригиран грешен синтаксис
✓ 119_klen_Set_DocsRange_InZReport_02	Коригиран грешен синтаксис
✓ 119_klen_Set_DocsRange_InZReport	Коригиран грешен синтаксис
✓ 119_klen_Set_DocsRange_ByDateTime	Коригиран грешен синтаксис
✓ 119_klen_Print_Document	Коригиран грешен синтаксис
✓ 119_klen_Print_DocsInRange	Коригиран грешен синтаксис
✓ 119_klen_Print_DocsFromDateToDate	Коригиран грешен синтаксис
✓ 049_receipt_Dsale_TextRow2	Дублирано име – името на втория метод става <b>049_receipt_Dsale_TextRow2Un</b>
✓ 043_config_Set_AutoCutting	Коригиран грешен синтаксис
✓ 087_config_Set_DepartmentName	Дублирано име – името на втория метод <b>..._Set_DepartmentNameTwoRows</b>
✓ 043_info_Get_PrintOption	Коригиран грешен синтаксис
✓ 107_items_Delete_All_Items	Дублирани имена на параметри – Option става <b>D_Option</b> и <b>A_Option</b>
✓ 107_items_Get_xx	Дублирани имена на параметри – всички входящи PLU стават <b>Target_PLU</b>
✓ Логическа група „B“	
✓ 066_config_Set_InvoiceRange	Стрешена проверка на входните данни на команда
✓ 088_info_Get_Department_StornoInfo	Стрешен делимитер
✓ 107_items_Set_ItemWithReplace	Стрешен делимитер
✓ 112_info_read_OperatorsInfo	Няма параметър "ErrorCode" при успешно изпълнение
✓ 112_info_read_OperatorsInfo	Промяна към <b>112_info_Get_StOperatorsData</b>
✓ 057_receipt_PrintClientInfo_14	Дублиране на име на параметър - Address1 става Address2
✓ 058_receipt_PLUxxx_xxx	Възможност за отрицателни стойности
✓ 053_receipt_PinPadTotal_Amount	Промяна към <b>053_receipt_PinPad_Total</b>
✓ 053_receipt_PinPadTotal_AmountTextRow1	Промяна към <b>053_receipt_PinPad_TotalTextRow1</b>
✓ 053_receipt_PinPadTotal_AmountTextRow2	Промяна към <b>053_receipt_PinPad_TotalTextRow2</b>
✓ 087_config_Set_DepartmentName	Дублирано име – името на втория метод <b>..._Set_DepartmentNameTwoRows</b>
✓ 112_info_read_OperatorsInfo	Промяна към <b>112_info_Get_OperatorsStornoData</b>
✓ 107_items_Delete_All_Items	Дублирани имена на параметри – Option става <b>D_Option</b> и <b>A_Option</b>
✓ 107_items_Get_xx	Дублирани имена на параметри – всички входящи PLU стават <b>Target_PLU</b>
✓ 103_info_Get_CurrentReceiptInfo	Промяна на име на параметър – Type става <b>ReceiptType</b>
✓ 083_service_Set_DecimalAndTaxRates	Дублиране на вх./изх. параметри – добавя се inr_/out_ пред името на параметрите
✓ 125_klen_Prepare_DocumentInRange	Типа на параметър „Input_ToDocNumber“ е входящ
✓ Нови методи	
✓ Логическа група „A“	
✓ 119_klen_Print_DocsInRange_InZReport	Отпечатва документи от КЛЕН от номер до номер за даден Z-отчет
✓ 053_receipt_Total_PAmountWithoutText	Продажба с корекция по процент - минимум параметри
✓ 049_receipt_Sale_UnWText	Продажба с употреба на мерна единица – без текст
✓ 049_receipt_Sale_CByPercentWText	Продажба с корекция по процент – без текст
✓ 049_receipt_Sale_UnCByPercentWText	Продажба с корекция по процент и мерна единица – без текст
✓ 049_receipt_Sale_CBySumWText	Продажба с корекция чрез твърда сума – без текст

✓	049_receipt_Sale_UnCBySumWText	С употреба на мерна единица и корекция чрез твърда сума – без текст
✓	049_receipt_DSale_UnWText	Продажба в департамент с мерна единица– без текст
✓	049_receipt_DSale_CByPercentWText	Продажба в департамент с корекция по процент – без текст
✓	049_receipt_DSale_CBySumWText	Продажба в департамент с корекция чрез твърда сума – без текст
✓	Логическа група „B“	
✓	053_receipt_Total_PAmountWithoutText	Команда за Тотал без параметри за печат на текст
✓	053_receipt_Total_WithoutText	Команда за тотал с тип на плащане в брой (без текст)
✓	053_receipt_PinPad_TotalWText	Команда за тотал с тип на плащане с дебитна карта (без текст)
✓	049_receipt_Sale_CByPercentWText	Продажба с корекция по процент - минимум параметри
✓	049_receipt_Sale_CBySumWText	Продажба с корекция чрез твърда сума – без текст
✓	049_receipt_DSale_CByPercentWText	Продажба в департамент с корекция по процент – без текст
✓	049_receipt_DSale_CBySumWText	Продажба в департамент с корекция чрез твърда сума – без текст
✓	Логическа група „C“	Добавени са <b>398</b> нови метода по протокола на устройствата от серия X
✓	Нови типове, методи и характеристики в COM сървър	
✓	TDeviceGroup	Изброен тип, съдържащ логическите групи в едно от които попада устройството
✓	device_Group	Характеристика, показваща логическата група на свързаното фиск. устройство
✓	itls_SummerDT	Характеристика, показваща дали текущото време (на оп.система) е лятно часово време
✓	itls_SummerDTFor	Метод, показващ дали дадено време и час дефинират лятно часово време
✓	Нови демо програми	
✓	Демонстрация на поддръжка на КЛЕН	C#
✓	Локален дейтаснап(Rest) сървър	Демонстрация на управление на фискално устройство през WEB страница

# Версия 3.0.0.6

Дата	Описание
02/2019	<p><b>Бета версия!</b></p> <ul style="list-style-type: none"> <li>✓ Нови полета (<b>properties</b>): <ul style="list-style-type: none"> <li>✓ FR_Invoice – тип: фискална бележка или фактура</li> <li>✓ FR_UNP – УНП на фискалната бележка/фактура</li> <li>✓ can_OpenFiscalReceipt – дали може да се отвори фискална бележка</li> <li>✓ can_OpenInvoiceReceipt – дали може да се отвори фактура</li> </ul> </li> <li>✓ Нови <b>device-independent</b> методи: <ul style="list-style-type: none"> <li>✓ init_FiscalReceiptValues</li> <li>✓ open_FiscalReceipt (команда 48)</li> <li>✓ open_InvoiceReceipt (команда 48 - разширена фискална бележка от тип фактура)</li> </ul> </li> <li>✓ Нови методи във връзка с въвеждането на device-independent метод за команда 48: <ul style="list-style-type: none"> <li>✓ логическа група А: <ul style="list-style-type: none"> <li>✓ 048_receipt_FiscalOpen_A01: <ul style="list-style-type: none"> <li>✓ Отваряне на фискална бележка БЕЗ наличие на поле UNP.</li> <li>✓ Еквивалентна на 048_receipt_Fiscal_01_Open.</li> </ul> </li> <li>✓ 048_receipt_FiscalOpen_A02: <ul style="list-style-type: none"> <li>✓ Отваряне на фактура БЕЗ наличие на поле UNP</li> <li>✓ Еквивалентна на 048_receipt_Invoice_01_Open</li> </ul> </li> <li>✓ 048_receipt_FiscalOpen_A03: <ul style="list-style-type: none"> <li>✓ Отваряне на фискална бележка с наличие на поле UNP</li> <li>✓ Еквивалентна на 048_receipt_Fiscal_02_Open</li> </ul> </li> <li>✓ 048_receipt_FiscalOpen_A04: <ul style="list-style-type: none"> <li>✓ Отваряне на фактура с наличие на поле UNP</li> <li>✓ Еквивалентна на 048_receipt_Invoice_Open</li> </ul> </li> </ul> </li> <li>✓ логическа група Б: <ul style="list-style-type: none"> <li>✓ 048_receipt_FiscalOpen_B01: Отваряне на фискална бележка БЕЗ наличие на поле UNP</li> <li>✓ 048_receipt_FiscalOpen_B02: Отваряне на фактура БЕЗ наличие на поле UNP</li> <li>✓ 048_receipt_FiscalOpen_B03: <ul style="list-style-type: none"> <li>✓ Отваряне на фискална бележка с наличие на поле UNP.</li> <li>✓ Еквивалентна на 048_receipt_Fiscal_Open</li> </ul> </li> <li>✓ 048_receipt_FiscalOpen_B04: <ul style="list-style-type: none"> <li>✓ Отваряне на фактура с наличие на поле UNP</li> <li>✓ Еквивалентна на 048_receipt_Invoice_Open</li> </ul> </li> </ul> </li> </ul> </li> <li>✓ bug fix: Проблем в автоматичното разпознаване на модели "SK1-21F" и "SK1-32F";</li> <li>✓ bug fix: Отстранен проблем при работа по TCP/IP с устройства в логическа група „А“</li> <li>✓ bug fix: Дублиране на имена на функции по HUMAN_NAME_03: <ul style="list-style-type: none"> <li>✓ <b>Промяна на имена в логическа група А:</b> <ul style="list-style-type: none"> <li>✓ 065_info_Get_AdditionalDailyInfo_01: 065_info_Get_AdditionalDailyInfo_01</li> <li>✓ 110_info_Get_AdditionalDailyInfo_01: 110_info_Get_AdditionalDailyInfo_02</li> <li>✓ 110_info_Get_AdditionalDailyInfo_02: 110_info_Get_AdditionalDailyInfo_03</li> <li>✓ 079_info_Print_ShortFMBByDTRange: 079_report_FMBByDateRange_Short</li> <li>✓ 111_report_ItemsWithGroup_InRange: 111_report_Items_InRangeByGroup</li> <li>✓ 080_run_SoundSignal: 080_other_Sound_Signal</li> </ul> </li> <li>✓ <b>Промяна на имена в логическа група Б:</b> <ul style="list-style-type: none"> <li>✓ 065_info_Get_AdditionalDailyInfo_01: 065_info_Get_AdditionalDailyInfo_04</li> <li>✓ 065_info_Get_AdditionalDailyInfo_02: 065_info_Get_AdditionalDailyInfo_05</li> <li>✓ 065_info_Get_AdditionalDailyInfo_03: 065_info_Get_AdditionalDailyInfo_06</li> <li>✓ 065_info_Get_AdditionalDailyInfo_04: 065_info_Get_AdditionalDailyInfo_07</li> <li>✓ 065_info_Get_AdditionalDailyInfo_05: 065_info_Get_AdditionalDailyInfo_08</li> <li>✓ 065_info_Get_AdditionalDailyInfo_06: 065_info_Get_AdditionalDailyInfo_09</li> <li>✓ 065_info_Get_AdditionalDailyInfo_07: 065_info_Get_AdditionalDailyInfo_10</li> <li>✓ 110_info_Get_AdditionalDailyInfo_03: 110_info_Get_AdditionalDailyInfo_11</li> <li>✓ 080_other_write_SoundSignal: 080_other_Sound_Signal</li> </ul> </li> </ul> </li> <li>✓ bug fix: <ul style="list-style-type: none"> <li>✓ Команда 110 в логическа група Б няма параметър "ErrorCode"</li> <li>✓ Команда 112 в логическа група Б няма параметър "ErrorCode"</li> </ul> </li> <li>✓ bug fix: Неизлизане от употреба на делimitери, характерни само за дадени, специфични команди;</li> <li>✓ bug fix: Добавени параметри в отговора на команда 70 'CashIn_CashOut' <ul style="list-style-type: none"> <li>✓ CashSum, ServIn, ServOut</li> </ul> </li> <li>✓ Нови команди в логическа група А: <ul style="list-style-type: none"> <li>✓ 034_report_Service_Contracts</li> <li>✓ 034_info_Service_Contracts</li> <li>✓ 072_service_Fiscalization</li> <li>✓ 083_service_Set_DecimalAndTaxRates</li> <li>✓ 089_service_Set_ProductionTestArea</li> <li>✓ 091_service_Set_SerialNumber</li> <li>✓ 098_service_Set_EIK</li> </ul> </li> </ul>

- ✓ 128\_service\_RAM\_Reset
- ✓ 133\_service\_Disable\_Print
- ✓ 134\_service\_Format\_KLEN
- ✓ 135\_service\_Test\_GPRS
- ✓ 119\_klen\_Get\_Info
- ✓ 119\_klen\_SInfo\_W\_ByNumber
- ✓ 119\_klen\_SInfo\_W\_ByNumbersRange
- ✓ 119\_klen\_SInfo\_W\_ByDateRange
- ✓ 119\_klen\_SInfo\_W\_GetNext
- ✓ 119\_klen\_SInfo\_Y\_ByNumber
- ✓ 119\_klen\_SInfo\_Y\_ByNumbersRange
- ✓ 119\_klen\_SInfo\_Y\_ByDateRange
- ✓ 119\_klen\_SInfo\_Y\_GetNext
- ✓ 119\_klen\_SInfo\_V\_ByNumber
- ✓ 119\_klen\_SInfo\_V\_ByNumbersRange
- ✓ 119\_klen\_SInfo\_V\_ByDateRange
- ✓ 119\_klen\_SInfo\_V\_GetNext
- ✓ 119\_klen\_SInfo\_ExV\_ByNumber
- ✓ 119\_klen\_SInfo\_ExV\_ByNumbersRange
- ✓ 119\_klen\_SInfo\_ExV\_ByDateRange
- ✓ 119\_klen\_SInfo\_ExV\_GetNext
- ✓ Нови команда в логическа група Б:
  - ✓ 072\_service\_Fiscalization
  - ✓ 079\_report\_FMByDateRange\_Short
  - ✓ 083\_service\_Set\_DecimalAndTaxRates
  - ✓ 089\_service\_Set\_ProductionTestArea
  - ✓ 098\_service\_Set\_EIK
  - ✓ 140\_clients\_Set\_ClientData
  - ✓ 140\_clients\_Del\_ClientData
  - ✓ 140\_clients\_Get\_ClientData
  - ✓ 140\_clients\_Set\_SellerName
  - ✓ 140\_clients\_Get\_FirstClientData
  - ✓ 140\_clients\_Get\_NextClientData
  - ✓ 140\_clients\_Del\_AllClientData
  - ✓ 124\_klen\_FindRange\_ByDateTime
  - ✓ 124\_klen\_FindRange\_ByZReports
  - ✓ 124\_klen\_Get\_Info
  - ✓ 125\_klen\_Prepare\_Document
  - ✓ 125\_klen\_Prepare\_DocumentInRange
  - ✓ 125\_klen\_Get\_TextRow
  - ✓ 125\_klen\_Print\_Document
  - ✓ 125\_klen\_Get\_SInfo
- ✓ Корекции в демо програмите:
  - ✓ фикс на някои малки грешки;
  - ✓ редизайн и добавяне на демо страница за **device-independent** методи

# Версия 3.0.0.5

Дата	Описание
02/2019	<p><b>Бета версия!</b></p> <ul style="list-style-type: none"> <li>✓ поддръжка на модел DP-15</li> <li>✓ new property: Operator_Code</li> <li>✓ new property: Operator_Code</li> <li>✓ new property: Operator_Password</li> <li>✓ new property: Till_Number</li> <li>✓ new property: St_Reason_Type</li> <li>✓ new property: St_Doc_Number</li> <li>✓ new property: St_Doc_DateTime</li> <li>✓ new property: St_FM_Number</li> <li>✓ new property: St_Doc_UNP</li> <li>✓ new property: St_ByInvoice</li> <li>✓ new property: St_InvoiceNumber</li> <li>✓ new property: St_Current_UNP</li> <li>✓ new property: can_OpenStornoReceipt</li> <li>✓ new property: AllReceipt_Count</li> <li>✓ new property: StReceipt_Count</li> <li>✓ new property: FiscalReceipt_Count</li> <li>✓ new method: init_StornoValues</li> <li>✓ new method: init_OperatorValues</li> </ul> <p>✓ за устройства от логическа група Б към които е свързан платежен терминал (производство на Датекс) се добавят следните команди:</p> <ul style="list-style-type: none"> <li>✓ 053_receipt_PinPadTotal_Amount</li> <li>✓ 053_receipt_PinPadTotal_AmountTextRow1</li> <li>✓ 053_receipt_PinPadTotal_AmountTextRow2</li> </ul> <p>✓ Забележка: Да се използват единствено при тип на плащане "С" – Плащане с дебитна карта</p>
	<p>✓ bug fix: Имената на различните варианти на команда 46 - дуближ, липси и грешки;</p> <p>✓ Имената се променят съгласно документацията за device-independent метода:</p> <ul style="list-style-type: none"> <li>✓ логическа група А: <ul style="list-style-type: none"> <li>✓ 046_receipt_Storno_06_Open: 046_receipt_StornoOpen_A01</li> <li>✓ 046_receipt_Storno_03_Open: 046_receipt_StornoOpen_A02</li> <li>✓ нов метод: 046_receipt_StornoOpen_A03</li> <li>✓ 046_receipt_Storno_02_Open: 046_receipt_StornoOpen_A04</li> <li>✓ 046_receipt_Storno_05_Open: 046_receipt_StornoOpen_A05</li> <li>✓ 046_receipt_Storno_01_Open: 046_receipt_StornoOpen_A06</li> <li>✓ 046_receipt_Storno_04_Open: 046_receipt_StornoOpen_A07</li> <li>✓ 046_receipt_Storno_Open: 046_receipt_StornoOpen_A08</li> <li>✓ 046_receipt_InvoiceStorno_04_Open: 046_receipt_StornoOpen_A09</li> <li>✓ 046_receipt_InvoiceStorno_03_Open: 046_receipt_StornoOpen_A10</li> <li>✓ нов метод: 046_receipt_StornoOpen_A11</li> <li>✓ 046_receipt_InvoiceStorno_02_Open: 046_receipt_StornoOpen_A12</li> <li>✓ нов метод: 046_receipt_StornoOpen_A13</li> <li>✓ 046_receipt_InvoiceStorno_01_Open: 046_receipt_StornoOpen_A14</li> <li>✓ нов метод: 046_receipt_StornoOpen_A15</li> <li>✓ 046_receipt_InvoiceStorno_Open: 046_receipt_StornoOpen_A16</li> </ul> </li> <li>✓ логическа група Б: <ul style="list-style-type: none"> <li>✓ 046_receipt_Storno_07_Open: 046_receipt_StornoOpen_B01</li> <li>✓ нов метод: 046_receipt_StornoOpen_B02</li> <li>✓ 046_receipt_InvoiceStorno_05_Open: 046_receipt_StornoOpen_B03</li> <li>✓ нов метод: 046_receipt_StornoOpen_B04</li> <li>✓ нов метод: 046_receipt_StornoOpen_B05</li> <li>✓ нов метод: 046_receipt_StornoOpen_B06</li> <li>✓ липсва метод: 046_receipt_Storno_07_Open</li> </ul> </li> </ul> <p>✓ bug fix: 053_receipt_Total_PAmount</p>

## Версия 3.0.0.4

Версия	Дата	Описание
3.0.0.4	02/2019	<b>Бета версия!</b> <ul style="list-style-type: none"> <li>✓ отстранени грешки по текущите команди;</li> <li>✓ добавена е група „info“ - команди, даващи различна информация от страна на фискалното устройство;</li> <li>✓ добавена е група „klen“ - команди за четене на клен на фискалното устройство;</li> <li>✓ нов изброен тип, даващ информация за типа на пакеттираното съобщение;</li> <li>✓ нови “properties”;</li> <li>✓ поправки в документацията;</li> <li>✓ поправки в генерирането на сорскод;</li> <li>✓ добавени енумерирани типове за нови модели;</li> </ul>
3.0.0.3	02/2019	<b>Бета версия!</b> <ul style="list-style-type: none"> <li>✓ отстранени грешки;</li> </ul>
3.0.0.2	02/2019	<b>Бета версия!</b> <ul style="list-style-type: none"> <li>✓ отстранени грешки;</li> </ul>

## Версия 3.0.0.1 и по-стари

Версия	Дата	Описание
3.0.0.1	01/2019	<b>Бета версия!</b> Поддържа промените в протокола на фискалните устройства във връзка с промените в наредба №18 / 2018. Нов интерфейс: ICFD_BGR Поддържа следните модели: <ul style="list-style-type: none"> <li>✓ FP-800</li> <li>✓ FP-2000</li> <li>✓ FP-650</li> <li>✓ FMP-10</li> <li>✓ SK1-21F</li> <li>✓ SK1-31F</li> <li>✓ FP-550</li> <li>✓ DP-05</li> <li>✓ DP-15</li> <li>✓ DP-25</li> <li>✓ DP-35</li> <li>✓ DP-150</li> <li>✓ WP-50</li> </ul>
2.0.0.xx	01/2002 - 2018	Поддържа на всички фискални устройства на Датекс, поддържащи протокола на фискалния принтер. Различните версии и предоставени интерфейси обхващат промените в наредба №18 за описания период, както и логическите модификации на използвания протокол.
1.0.0.xx	01/2001	Поддържа на принтер FP-3530

## Въведение

Този документ съдържа информация относно COM Server "FP3530" който на практика представлява ActiveX библиотека за управление на фискалните устройства на Датекс в България. COM сървъра позволява лесна интеграция, употреба и управление от страна на всеки съвременен език за програмиране под OS Windows.

Ако не сте запознати с COM технологията – накратко казано, COM сървъра е обект, който предоставя на клиентския софтуер имплементация различни видове интерфейси. Клиентския софтуер може да използва имплементацията на всеки от предоставените интерфейси. Има два основни типа COM сървъри, „in-process” и „out-of-process”. "FP3530" е „out-of-process” COM сървър. Същността на COM е така наречения неутрален път за имплементиране на обекти. Те могат да бъдат използвани в среди за разработка – различни от използваната за разработката и създаването на предоставяните обекти.

“FP3530” Ви позволява като програмист да управлявате фискалните устройства на Датекс чрез различни видове на транспортния протокол – RS232, USB или TCP/IP (в зависимост от модела).

## Поддържани устройства

По-долу е даден списък на устройствата на Датекс които могат да се управляват с помощта на "FP3530".

Модел	Тип	RS-232	LAN (TCP/IP)	Логическа група
FP-650	Фискален принтер	Да	Не	А
FP-2000	Фискален принтер	Да	Да	
FP-800	Фискален принтер	Да	Да	
SK1-21F	Фискален принтер за вграждане	Да	Не	
SK1-31F	Фискален принтер за вграждане	Да	Не	
FP-700	Фискален принтер	Да	Да	
FMP-10	Мобилен фискален принтер	Да	Не	
DP-05	Мобилен касов апарат	Да	Не	В
DP-15	Касов апарат	Да	Не	
DP-25	Касов апарат	Да	Не	
DP-35	Касов апарат	Да	Не	
DP-150	Касов апарат	Да	Не	
WP-50	Мобилен касов апарат	Да	Не	
WP-50X	Мобилен касов апарат	Да	Не	С
DP-25X	Касов апарат	Да	Да	
DP-150X	Касов апарат	Да	Не	
WP-500X	Касов апарат	Да	Да	
DP-05C	Мобилен куриерски касов апарат	Да	Не	
FP-700X	Фискален принтер	Да	Да	
FMP-55X	Мобилен фискален принтер	Да	Не	
FMP-350X	Мобилен фискален принтер	Да	Не	

## Поддръжка - езици

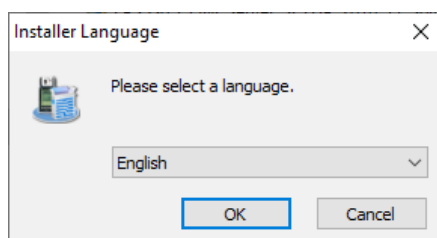
"FP3530" поддържа два езика – Английски и Български. Клиентските програми могат да изберат/настроят езика на СОМ сървъра програмно и след това текстовите съобщения за грешки или текста за статуса на устройството (статус битовите) ще се връщат на съответния език. Можете да видите пример за употребата на тези възможности в демо програмите разпространявани заедно с инсталацията.



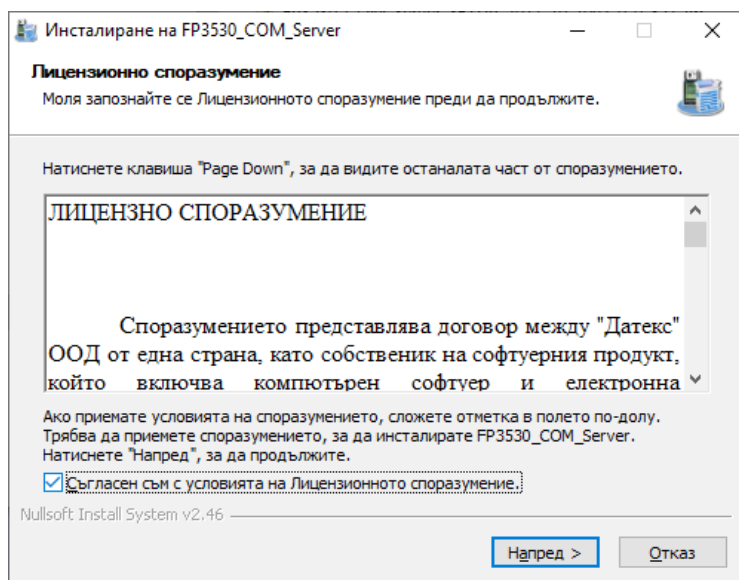
# Инсталация

Инсталацията на "FP3530" е лесна. По-долу е описан процеса стъпка по стъпка.

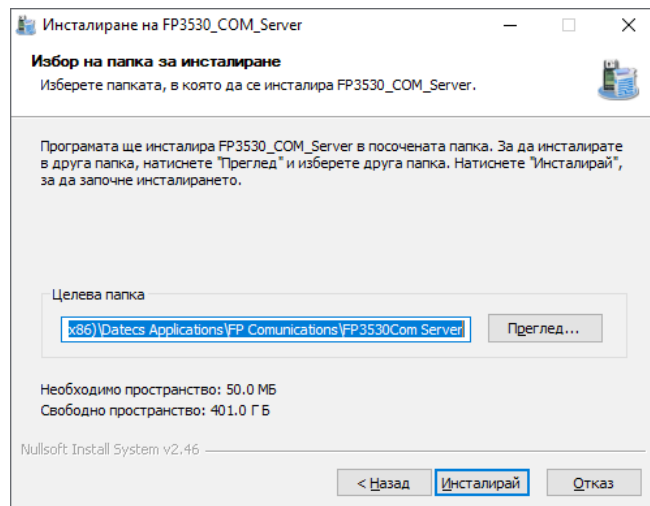
Стартирайте сетъп файла. Ако видите диалогов прозорец от типа на "User account control" (в по-новите версии на Windows), който може да е с различен вид в зависимост от версията на Windows – натиснете бутона "Yes".



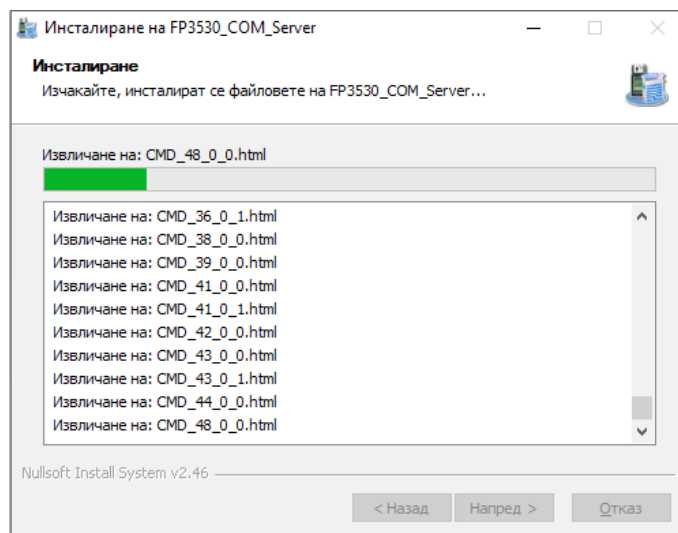
След показването на сплаш картинката, ще видите диалоговия прозорец за избор на език. Изберете предпочитания от Вас език и натиснете бутона "Ok", след което ще видите страницата "License Agreement".



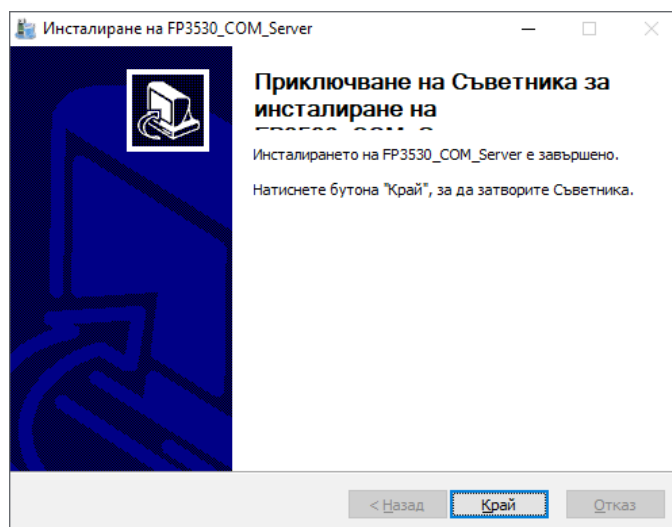
Прочетете лицензното споразумение и ако сте съгласни с условията му - изберете/включете, чекбокса за съгласие и натиснете бутона "Напред".



При желание - посочете папка за инсталиране и натиснете бутона „Инсталирай“



Моля изчакайте, докато приключи инсталацията.



Ако всичко е наред ще видите финалната страница на инсталатора. Натиснете бутона "Finish".

Това е всичко – the "FP3530" е инсталиран. Не трябва да го стартирате или да настройвате нещо. Необходимите настройки на версия 3.0.0.xx се управляват по програмен път, като "FP3530" има собствени настройки по подразбиране. Ако досега не сте използвали този тип интерфейси – разгледайте демо програмите. В някои от тях (на различни езици за програмиране) е показано как може да се управляват настройките. Демо програмите са предоставени както със сорскода си, така и в бинарен вид, така че могат да се използват от крайните потребители за целите на настройката, но екипа на Датекс препоръчва да имплементирате тези няколко метода в клиентските програми, които разработвате. "FP3530" записва сам последните настройки, с които е използван (след успешна връзка с фискалното устройство). Вашия потребителски интерфейс може да се възползва от тази черта в поведението на COM сървъра.

## Няколко думи за протокола на ниско ниво

"FP3530" COM сървър енкапсулира имплементацията на описания по-долу протокол на ниско ниво. Не е необходимо да знаете протокола в детайли, но все пак тук предоставяме информация във връзка с това. Понякога – за нуждите на разработката – може да решите да използвате възможностите за проследяване/лог на комуникацията (tracking mode) на сървъра. Ако изпращате „bug report” свързан с това ниво на комуникация - ще е полезно да имате известни познания по темата.

Ще опиша по-долу, две модификации на протокол за комуникация с някои от най-широко разпространените модели фискални устройства, производство на Датекс ООД.

## Терминология

В този раздел засягам използваната от мен терминология.

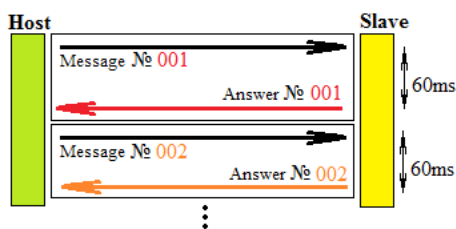
- Значението на [бит](#) и [байт](#);
- Бройни системи:
  - Ще наричам за по-кратко [десетичната бройна](#) система DEC;
  - Ще наричам за по-кратко [шестнайсетичната бройна](#) система HEX;
- Фискалното устройство ще наричаме за по-кратко ФУ или Slave;
- Компютъра (по-точно софтуера, който изпраща съобщенията до ФУ) ще го наричаме Host;
- Транспортен протокол:
  - Под серийна комуникация между Host и Slave ще разбираме свързаност по [RS-232](#) или по [USB](#);
  - Под БТ (BT) комуникация между Host и Slave ще разбираме свързаност по [Bluetooth](#);
  - Под TCP/IP комуникация между Host и Slave ще разбираме свързаност по [TCP/IP](#);
- Специфични, еднобайтови съобщения:
  - Когато говорим, че Slave е изпратил на/към Host, байт NAK ще разбираме байт със стойност 15H ([HEX](#));
  - Когато говорим, че Slave е изпратил на/към Host, байт SYN ще разбираме байт със стойност 16H ([HEX](#));

## Синхронен протокол

Протокола за комуникация с фискалните устройства на Датекс ООД е синхронен.

- Имаме синхронизация в реда на започване, изпращане и получаване на съобщенията;
- Имаме синхронизация по време;
- Имаме синхронизация във формат и размер на съобщенията;

### Синхронизация в реда на започване, изпращане и получаване на съобщенията

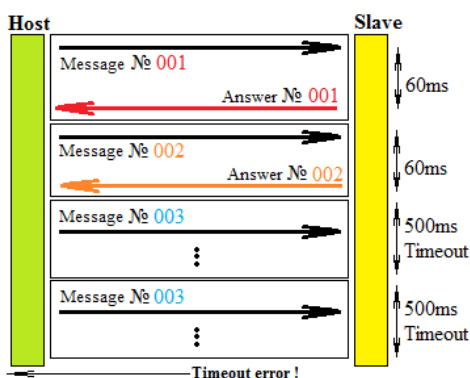


Фискалното устройство (**Slave**) не може да инициира само комуникация. Тя се стартира задължително и винаги от програмата, която се намира на съответния компютър (**Host**).

Ако получавате някаква информация от страна на фискалното устройство по комуникационния канал без да сте я инициирали – това обикновено е признак за дефектирало устройство или за неизчистен буфер.

- Host инициира комуникация, като изпраща пакетирано съобщение, съдържащо команда към Slave;
- В зависимост от текущото си състояние Slave извършва или не исканата операция, след което задължително отговаря с пакетирано съобщение (или в някои случаи с единичен байт);
- Host трябва да изчаква отговора от Slave преди да изпрати друго съобщение, освен ако не изминало времето за таймаут (вж. синхронизацията по време);

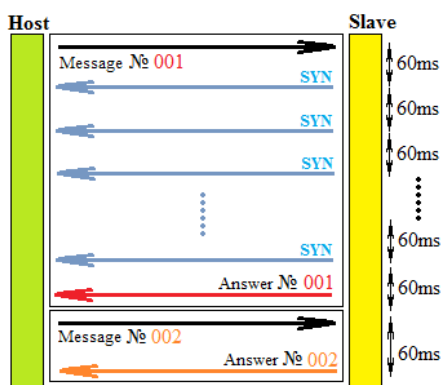
### Синхронизация по време



- На всяка от командите идващи от Host – Slave трябва да отговори не по-късно от 60ms (при това или с пакетирано съобщение или със строго определен еднобайтов код);
- Host има 500ms timeout за получаване на отговор от Slave;

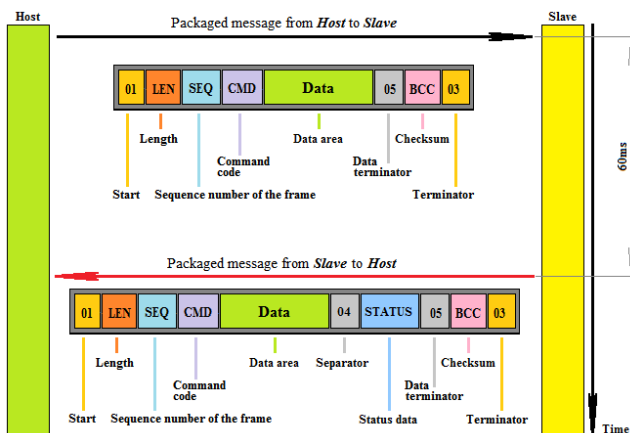
- Ако за 500ms не се получи никакъв отговор, Host трябва да предаде съобщението отново и при това със същия пореден номер и същата команда;
- След два неуспешни опита, Host трябва да индицира, че или няма връзка със Slave или че има хардуерен проблем в устройството;
- Timeout от 500ms се използва в случай, че става дума за комуникация по RS-232 или USB. Ако транспортния протокол е друг (например TCP/IP) - трябва да предвидите по-голям период от време в зависимост от качеството на трасето;

## Поддръжка на нормално забавяне на отговора от страна на Slave



*В случай, че командата е такава, че Slave ще се забави с отговора си, то Slave започва да изпраща по канала байт SYN на всеки 60ms. Така Slave казва на Host, че е зает в момента, но е напълно изправен и функциониращ. Казва му да изчака още малко.*

## Синхронизация по формат на съобщенията



- Всички команди и отговори между Host и Slave (освен описаните по-горе едно-байтови съобщения), са пакетирани в строго определен и предварително известен формат;
- “Случайност” се допуска единствено в областта за логически данни, но мястото където се намират тези данни в пакета, както и максималния им размер са строго определени;
- Форматите на пакетираното съобщение от Slave към Host и обратно са строго

определени, но “различни”, тъй като в отговора от страна на Slave се съдържа и допълнителна информация за статуса на устройството. Дори и ако устройството няма какво да изпрати, като логически данни към Host (т.е. в областта за данни няма нищо), то въпреки това отговаря, за да може програмата (Host) да научи от статус байтовете състоянието на устройството и да провери дали няма някакъв проблем (например, че е

свършила хартията);

- В случай на проблем или шум по трасето на комуникация - напълно възможно е Slave да получи команда с грешна контролна сума (чексума). В този случай Slave отговаря с един байт NAK. С него той казва, че Host трябва да повтори съобщението си без да променя нещо в него - просто да го изпрати отново.

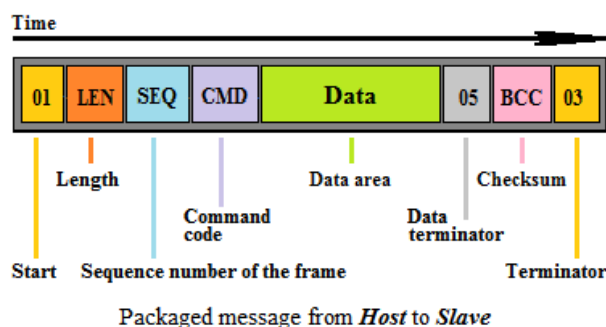
## Модификации на протокола

Описаните по-долу две модификации, могат грубо да се разделят на "стар протокол" и "нов протокол". Така наречения "стар протокол", възниква по време, когато най-масовата операционна система е DOS и съществува заради нуждата от съвместимост със стари, но "живи" продукти. При втората модификация, която можем да наричаме за по-кратко "нов протокол" имаме някои съществени подобрения:

- заделя се по-голям интервал на допустимите команди (т.е. предвижда се появата в бъдещето на нови, неописани към момента команди);
- дава се възможността за прехвърляне на по-голям обем данни в рамките на една команда;
- елементи на стандартизация на командите, като резултат от опита на разработчиците на софтуер и на натрупаните изисквания от страна на различни разработчици на софтуер:
  - код на грешка, като първи параметър в отговора на устройството - с общо значение за всички модели, разработени по този протокол;
  - стандартизиране на сепараторите в областта за данни;
  - заделяне на повече статус байтове в отговора от страна на Slave;

И двата протокола са валидни и еднакво популярни, поради което ще опиша и двете модификации. Разликите са посочени в различен цвят. Стойностите са в HEX

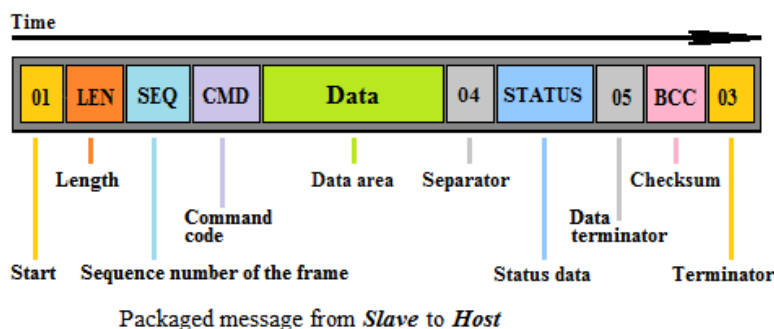
## Общо описание на пакетираното съобщение



- **01 (Start)** - Стартиращ байт.
  - Дължина: 1 байт;
  - Стойност: 01H;
- **LEN** - Дължина на съобщението. Това са броя байтове от стартиращия байт <01> (**но без него**) до терминатора на данни <05> (**включително**), плюс фиксирано отместване от 20H.
  - Стар протокол:
    - Дължина: 1 байт;



- **Стойности:** от 20H до FFH;
- Нов протокол:
  - **Дължина:** 2 байта (word), кодирани в 4 байта;
  - **Стойности:** Тъй като в този протокол възможните стойности са в интервала [0..65535] (DEC), програмата Ви трябва да попълва необходимата стойност в променлива от тип word. Всяка цифра от тези два байта се предава, като към нея се прибави 30H, в резултат на което в пакетираното съобщение за това поле ще участват четири байта. Причината за това кодиране е свързана с вътрешната реализация на проткола във фискалните устройства и не от значение.
  - *Пример: променлива със стойност 1AE3H в пакетираното съобщение се представя като четири байта със стойности съответно 31H, 3AH, 3EH, 33H.*



- **SEQ** - Пореден номер на пакетираното съобщение.
  - Slave записва същия пореден номер в отговора си;
  - Ако Slave получи съобщение със стойност на <SEQ>, същата като при последното получено от него съобщение, то той не извършва действие, а повтаря последното изпратено от него съобщение;
  - Стар протокол:
    - **Стойности:** от 20H до 7FH;
    - **Дължина:** 1 byte;
  - Нов протокол:
    - **Стойности:** от 20H до FFH;
    - **Дължина:** 1 байт;
- **CMD** - Номер на командата.
  - Ако Slave получи несъществуващ или невалиден код, той отговаря с пакетирано съобщение с нулева дължина на полето за данни и вдига в единица статус бита за невалиден код на командата;
  - Когато отговаря на дадено съобщение - Slave записва същия номер <CMD> в отговора си;
  - Горната граница на стойността на допустимите команди зависи от конкретното устройство;
  - Стар протокол:
    - **Дължина:** 1 байт.
    - **Стойности:** от 20H до 7FH;
  - Нов протокол:
    - **Дължина:** 2 байта (word), кодирани в 4 байта;
    - **Стойности:** Тъй като в този протокол възможните стойности са в интервала [0..65535] (DEC), програмата Ви трябва да попълва необходимата стойност в променлива от тип word. Всяка цифра от тези два байта се предава, като към нея се прибави 30H, в резултат на което в пакетираното съобщение за това поле ще участват четири байта. Причината за това кодиране е свързана с вътрешната

реализация на протокола във фискалните устройства и не от значение.

- Пример: променлива със стойност **1AE3H** в пакетираният съобщение се представя като четири байта със стойности съответно **31H, 3AH, 3EH, 33H**.
- Забележка: Списък с конкретните, позволени и валидни стойности на командите за дадено фискално устройство трябва да се взема от документацията на всяко конкретно устройство. Това, че дадена стойност е възможна - не означава, че се използва или че е позволена.

- **Data** - Логически данни.

- Форматът и дължината на областта за данни зависи от конкретната команда;
- Ако командата не изисква изпращане на данни към **Slave** или съответно, ако в отговора към **Host** няма данни, то дължината на това поле е нула;
- Ако при команда от Host към Slave има синтактична грешка в данните, то Slave отговаря с пакетираният съобщение с нулева дължина на полето за данни и вдига в единица статус бита за синтактична грешка;
- Дължина (от Host към Slave): 0 – 213 байта;
- Дължина (от Slave към Host): 0 – 218 байта;
- Стойности: 09H, 0AH и стойностите в интервала от 20H до FFH;

- **04 (Separator)** - Сепаратор. В пакетираният съобщение от Host към Slave - това е байта, който служи за маркер, отделящ логическите данни от данните за статуса на Slave (статус байтовете).

- Дължина: 1 байт;
- Стойност: 04H;

- **STATUS** - Полето за данни, съдържащо информация за текущото състояние на фискалното устройство. В статиите от тук нататък ще наричам байтовете от това поле: "**Статус байтове**";

- Значението на всеки бит от всеки статус байт, трябва да се вземе от документацията на съответното устройство;
- Състоянието на даден бит отговаря логически на някакво състояние на устройството. В статиите от тук нататък - ще наричам всеки такъв бит - "статус бит", а когато става дума за състоянието на даден статус ще имаме предвид стойността на съответния статус бит;
- За съжаление е възможно позицията на даден статус да се мени при различните устройства или дори дадения статус да отсъства поради хардуерни специфики;
- Като цяло статус битовете могат да бъдат разделени на следните групи и подгрупи:
  - такива, на които може да не се обръща внимание:
    - даден статус бит може да бъде запазен за вътрешна употреба (в зависимост от случая - може да бъде винаги със стойност 0 или 1);
    - даден статус бит може да бъде запазен за бъдеща употреба (в зависимост от случая - може да бъде винаги със стойност 0 или 1);
  - такива, на които задължително трябва да обръщате внимание:
    - Информативни статус битове - носят информация за логическо или физическо състояние на устройството. Някои операции не са позволени в дадени ситуации и поради това е добре да ги следите. Например:
      - Ако принтера не е фискализиран - явно не може да издава фискални бележки;
      - Ако принтера няма хартия - не може да печата, каквито и да било документи;
    - Статус битове за грешка - носят обща информация за грешката, която е възникнала. Например: непозволена команда, синтактична грешка и т.н. Понякога (в стария протокол) единствения начин програмата да вземе

правилно решение за разрешаване на ситуацията е да сглоби информацията от всички налични статуси в отговора от страна на Slave;

- Стар протокол:
  - **Дължина:** 6 байта;
  - Стойности за всеки от байтовете: от 80H до FFH;
- Нов протокол:
  - **Дължина:** 8 байта;
  - Стойности за всеки от байтовете: от 80H до FFH
- 05 (Data terminator) - Сепаратор.
  - В пакетираното съобщение от Host към Slave - това е байта, който служи за маркер, отделящ логическите данни от останалата част на пакетираното съобщение.
  - В пакетираното съобщение от Slave към Host - това е байта, който служи за маркер, отделящ статус данните от останалата част на пакетираното съобщение.
    - Дължина: 1 байт;
    - Стойност: 05H;
- ВСС - Контролна сума (чексума). Сумата включва стойностите от <01> без него до <05> включително. Тъй като в този протокол възможните стойности са в интервала [0..65535] (DEC), програмата Ви трябва да попълва необходимата стойност в променлива от тип word. Всяка цифра от тези два байта се предава, като към нея се прибави 30H, в резултат на което в пакетираното съобщение за това поле ще участват четири байта. Причината за това кодиране е свързана с вътрешната реализация на протокола във фискалните устройства и не от значение.
  - Пример: променлива със стойност 1AE3H в пакетираното съобщение се представя като четири байта със стойности съответно 31H,3AH,3EH,33H.
    - Дължина: 2 байта (word), кодирани в 4 байта;
    - Стойности: от 30H до 3FH.
- 03 (Terminator) - Байт, маркиращ края на пакетираното съобщение.
  - Дължина: 1 byte.
  - Стойност: 03H.

# Употреба на FP3530

## Няколко думи във връзка със зависимостта от хардуера

FP3530 и по-специално интерфейс „CFD\_BGR“ има смесени характеристики по отношение на зависимостта от хардуер и фирмуер. Навсякъде където това е било възможно – методите и техните параметри не зависят от конкретния фирмуер, но за съжаление това не винаги е възможно. Някои минимални или максимални стойности (например брой печатими символи) очевидно зависят от конкретното устройство. В други случаи (поради особеностите на Българския пазар) устройствата „наследяват“ предишни версии на дадения модел.

Когато разработвате Вашия проект – имайте предвид, че интерфейс „CFD\_BGR“ ще поддържа всички фискални устройства на Датекс, но като цяло това е зависима от устройството библиотека.

Интерфейс „CFD\_BGR“ предоставя необходимите методи и характеристики с които да Ви улесни максимално възможно и продукта Ви да знае във всеки един момент както какъв е модела, така и какво е неговото състояние.

## Групи фискални устройства в зависимост от типа на пакетираният съобщение

Интерфейс „CFD\_BGR“ предоставя информация за типа на пакетираният съобщение след успешно отваряне на връзката към фискалното устройство. За целта интерфейса предоставя енумериран тип „**TrackedMessageType**“ и property за четене „**type\_PackagedMessage**“. По отношение на типа на пакетираният съобщение – фискалните устройства на Датекс се разделят на две основни групи към днешна дата. Ако използвате интерфейс „CFD\_BGR“, Вие няма да се занимавате с пакетиране и разпакетиране на съобщенията на „ниско“ ниво. Практическата и видима за Вас разлика е в броя на статус битовете които фискалното Ви изпраща при всеки отговор на команда. Вижте таблицата по-долу за повече информация.

## Групи фискални устройства в зависимост от входните данни при изпълнение на различни методи

През годините – фирмуера за различните фискални устройства на Датекс са създавани от различни екипи. Някои от тях външни фирми – по-късно закупени от Датекс. Поради това през годините се получава постепенна разходимост в логиката и параметрите на различните модели фискални устройства. Датекс предприема съответните стъпки за постепенен преход към стандартизация в предлаганите методи, но тъй като трябва да се спазва и условието за наследяването на функционалността (при всяка промяна в законите има преходен период през който стар и непроменен софтуер трябва да работи едновременно с новия) това не е съвсем лесна задача. Логически, ако разделим фискалните устройства в зависимост от поведението и методите които те предлагат – всяко фискално устройство предлагано от Датекс попада в една от три общи групи. Ако използвате интерфейс „CFD\_BGR“, Вие ще

имате достъп както до методите, специфични за дадената група, така и до предлаганите от интерфейса независими от модела методи.

Стойност	Модел	Логическа група „А“	Логическа група „В“	Логическа група „С“	Тип на пакетизираното съобщение	Брой статус битове	Стандартизиран код на грешка в отговора	Стандартизиран делимитер
mc_Unknown	Стойност на property “device_Model” преди отваряне на връзка към фискалното устройство.							
mc_DP_05	DP-05	x	✓	x	pmt_01	6	x	x
mc_DP_15	DP-15	x	✓	x	pmt_01	6	x	x
mc_DP_25	DP-25	x	✓	x	pmt_01	6	x	x
mc_DP_35	DP-35	x	✓	x	pmt_01	6	x	x
mc_DP_150	DP-150	x	✓	x	pmt_01	6	x	x
mc_WP_50	WP-50	x	✓	x	pmt_01	6	x	x
mc_FP_650	FP-650	✓	x	x	pmt_01	6	x	x
mc_FP_800	FP-800	✓	x	x	pmt_01	6	x	x
mc_FP_2000	FP-2000	✓	x	x	pmt_01	6	x	x
mc_SK_21F	SK-21F	✓	x	x	pmt_01	6	x	x
mc_SK_31F	SK-31F	✓	x	x	pmt_01	6	x	x
mc_FMP_10	FMP-10	✓	x	x	pmt_01	6	x	x
mc_FP_550	FP-550	✓	x	x	pmt_01	6	x	x
mc_FP_700	FP-700	✓	x	x	pmt_01	6	x	x
mc_FP_700X	FP-700X	x	x	✓	pmt_02	8	✓	✓
mc_DP_25X	DP-25X	x	x	✓	pmt_02	8	✓	✓
mc_DP_150X	DP-150X	x	x	✓	pmt_02	8	✓	✓
mc_WP_50X	WP-50X	x	x	✓	pmt_02	8	✓	✓
mc_WP_500X	WP-500X	x	x	✓	pmt_02	8	✓	✓
mc_FMP_55X	FMP-350X	x	x	✓	pmt_02	8	✓	✓
mc_FMP_350X	FMP-55X	x	x	✓	pmt_02	8	✓	✓
mc_DP_05C	DP-05C	x	x	✓	pmt_02	8	✓	✓

## “Thread safe” ?

Един от често задаваните въпроси за СОМ сървъра е "Thread-safe" or not?".

Термина "Thread-safe" е малко особен. Трябва да знаем какво всъщност значи в този конкретен случай. Нека погледнем дефиницията: "A piece of code is thread-safe if it functions correctly during simultaneous execution by multiple threads."

***При директна употребата на методите на този СОМ сървър - не бива да смятате, че тази библиотека е "Thread-safe"! Когато извиквате метод с директна комуникация към фискалното устройство (например и условно казано, „Изпълни команда ...“) СОМ сървърът не проверява и не поставя изпълнението на тази команда в опашка. Ако извиквате командите към устройството едновременно от няколко различни нишки (threads) – вероятно изпълнението на командите няма да е коректно. Препоръчваме Ви в зависимост от интерфейса който използвате – да организирате собствена подредба на командите в опашка.***

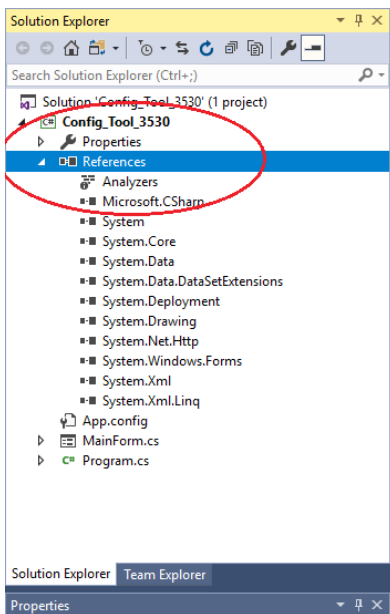
В интерфейс „CFD\_BGR“ е заложен механизъм за транзакционно използване на командите и подреждане в опашка. Ако решите да го използвате – използвайте само този подход, защото ако едновременно с това извиквате от друга нишка команда към фискалното устройство – употребата на интерфейса отново няма да е "Thread-safe".

*При проектирането на Вашия продукт – вземете тази информация под внимание! Пишете ни – ще помогнем със съвет.*

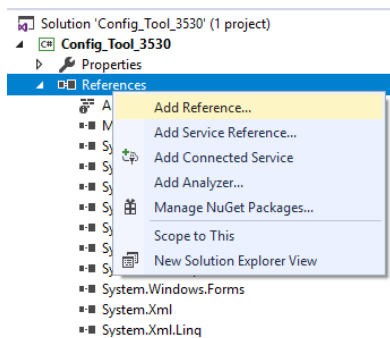
## Използване на интерфейс „CFD\_BGR“ от Visual Studio 2017 (C#)

Лесно е!

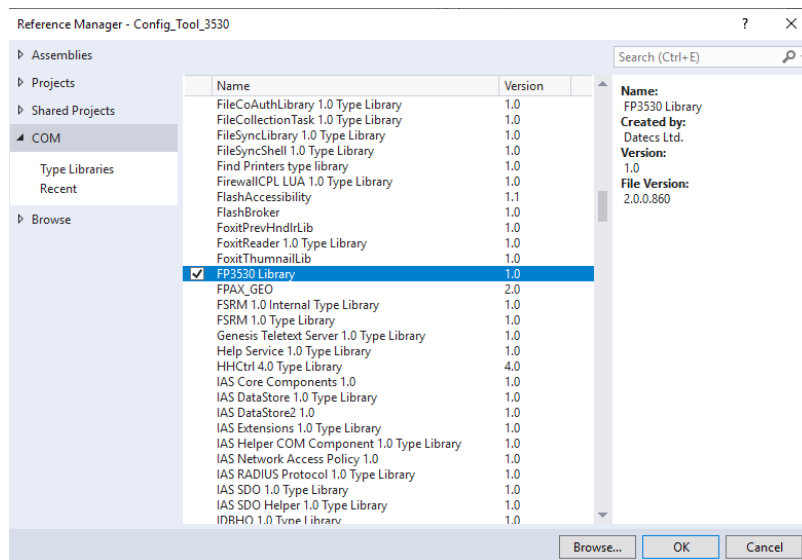
- ✓ Инсталирайте FP3530 COMServer и след това – отидете на "Solution Explorer\References" в проекта Ви;



- ✓ Натиснете десния бутон на мишката и изберете “Add reference”;



- ✓ В диалоговия прозорец – изберете COM и намерете FP3530 в списъка. Натиснете бутона “OK”;



- ✓ Това е всичко – Visual Studio ще добави всички предоставени от сървъра интерфейси и Вие можете да започнете да ги използвате;

```

1  using System;
2  using System.Collections.Generic;
3  using System.Drawing;
4  using System.IO;
5  using System.ComponentModel;
6  using System.Data;
7  using System.Linq;
8  using System.Text;
9  using System.Threading;
10 using System.Windows.Forms;
11 using System.Globalization;
12 using System.Diagnostics;
13 using FP3530;
14 using System.Runtime.InteropServices;
15
16 namespace Config_Tool_3530
17 {
18
19     public partial class fm : Form
20     {
21         private bool inCommand = false;
22         private bool invalidCharacter1;
23         private string command_Name = "";
24         private bool fWeHaveWaitEvent;
25         private byte fSYNCount;
26         private byte fAppMessCount;
27         private FP3530.CFD_BGR serv;
28         private bool fComServer_Started;
29         private TMyLanguages fLanguage;
30         private bool fDevice_Connected;
31         private int fComPort;
32         private bool fLAN_Connected;
33     }
34 }

```

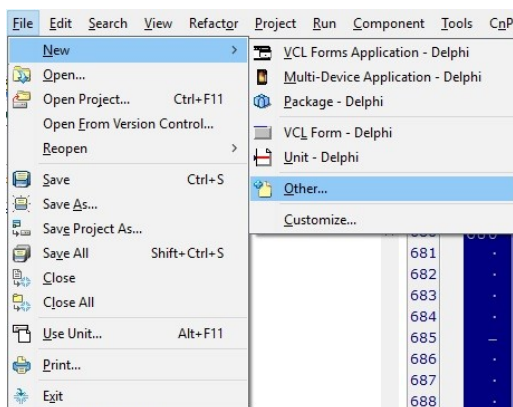
- ✓ Можете да намерите демо програми написани на C# в папките идващи с инсталацията;



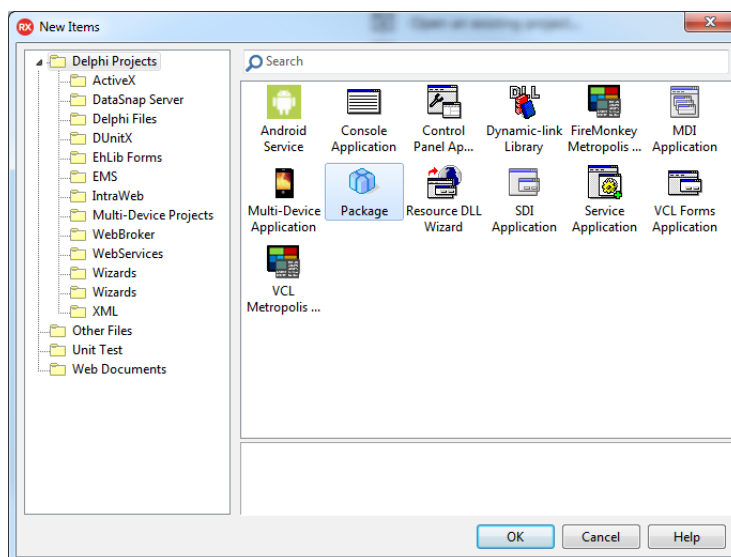
## Употреба на интерфейс „CFD\_BGR“ от Delphi

В Делфи има повече от един начин за употреба на COM сървър – тук показваме препоръчвания от разработчика (и най-лесен) метод. Имайте въпреки това предвид, че при някои версии на Windows е добре да стартирате Delphi като администратор.

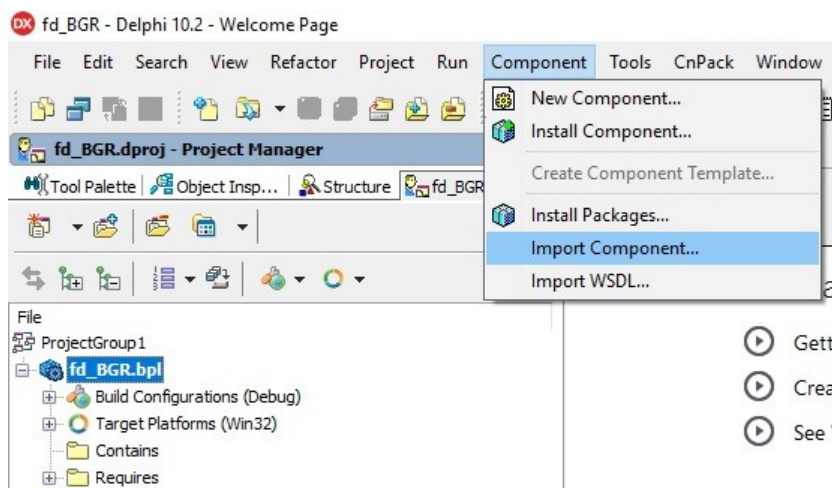
Можете лесно да си направите пакет, в който да добавите „type library“. След компилирането и регистрирането на пакета – ще можете да използвате „CFD\_BGR“ като компонент. Описания по-долу пример е направен под Delphi 10.2 но лесно може да се приложи за всички по-стари версии на Delphi – назад до версия Delphi 7 (автора на документа го изпробвал до тази версия).



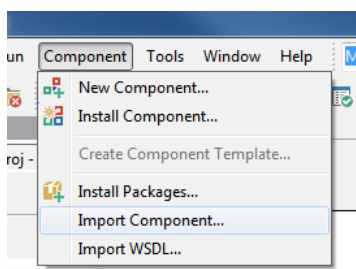
- ✓ Отидете на “File\New” и изберете “Other”;



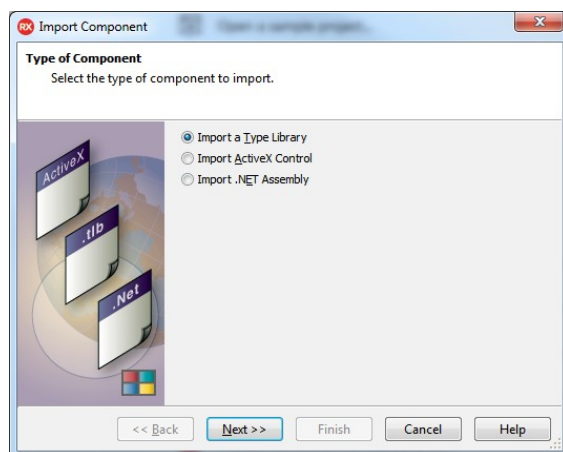
- ✓ Посочете package и натиснете бутона “OK”;



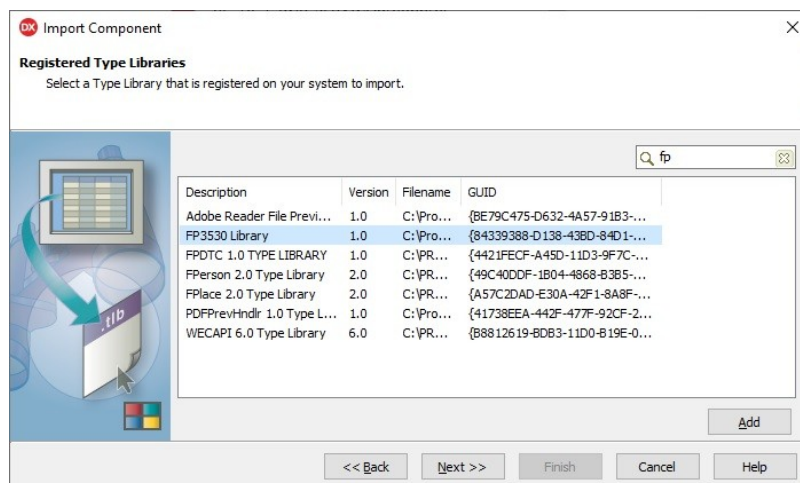
- ✓ Delphi ще направи нов празен проект за „package” - можете да го запишете в зависимост от предпочитанията си – например като “fd\_BGR.dproj” (от фискални устройства – България);



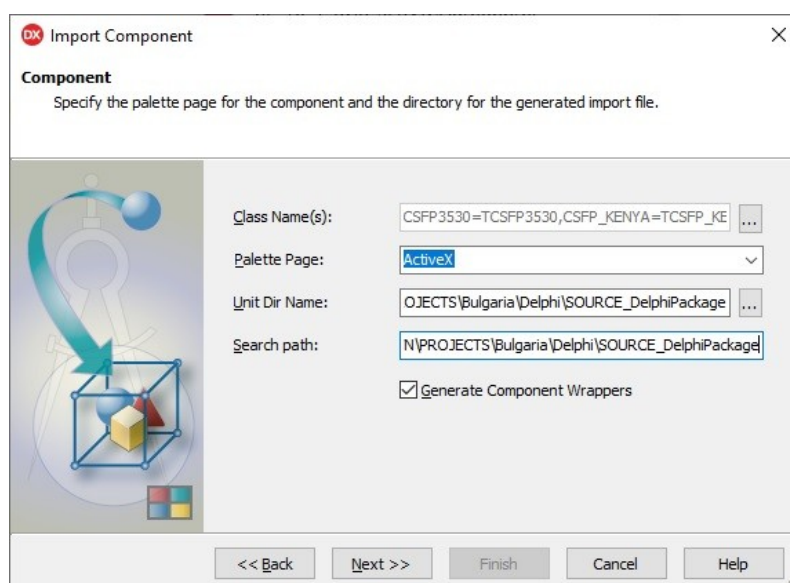
- ✓ В основното меню изберете “Component\Import Component”;



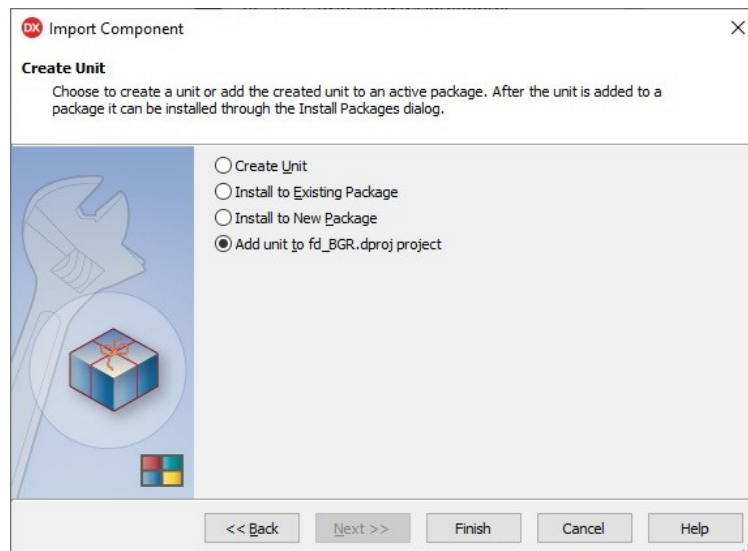
- ✓ Изберете “Import a Type Library” и натиснете бутона “Next”;



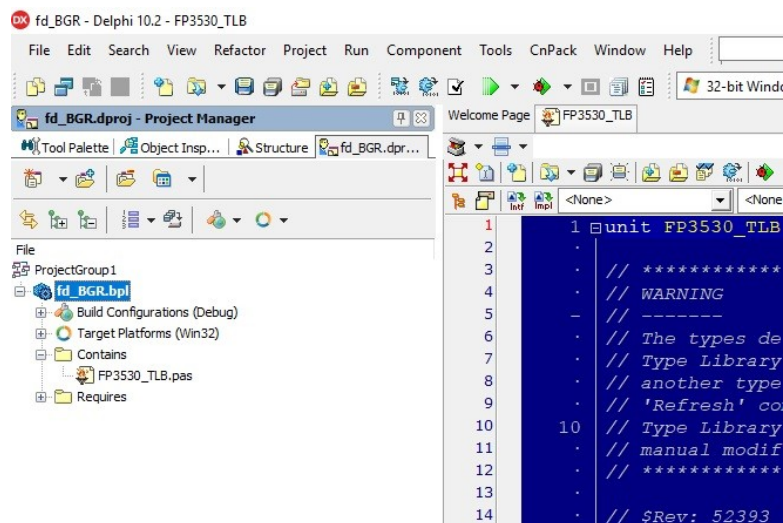
- ✓ На следващата страница – намерете FP3530 и натиснете бутона “Next”;



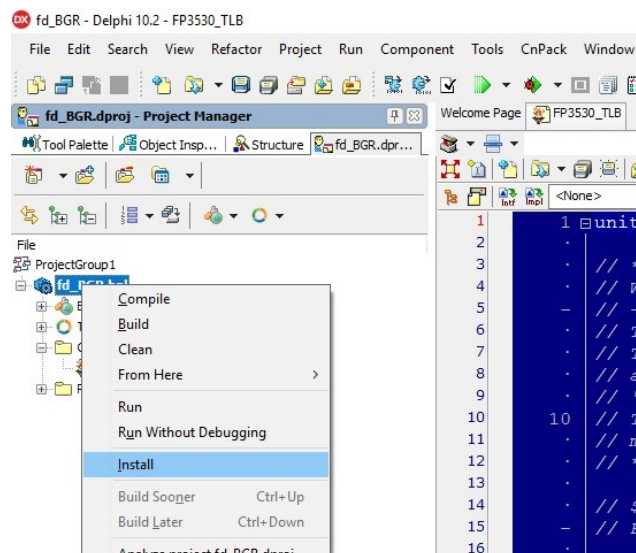
- ✓ На следващата страница – изберете “Palette page”, “Unit dir name” и активирайте „Generate component wrappers”. Натиснете бутона “Next”;



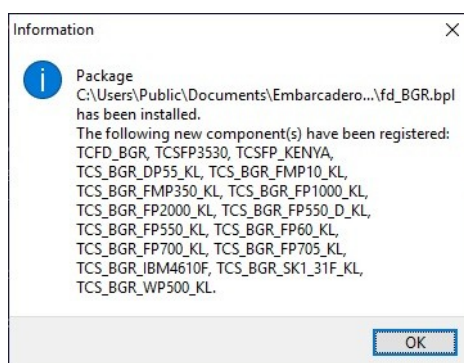
- ✓ Изберете опция “Add unit to fd\_BGR.dproj project” и натиснете бутона “Finish”;



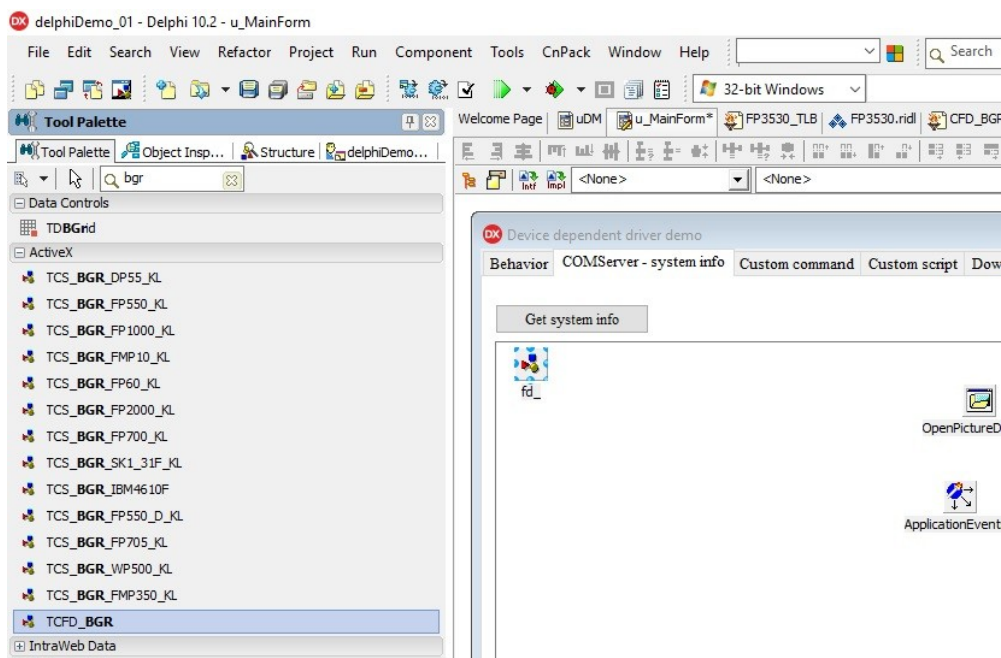
- ✓ Delphi ще създаде за Вас “FP3530\_TLB.pas”. От контекстното меню изберете Save и после Build the project;



- ✓ Инсталирайте пакета в Delphi IDE и можете да започнете да използвате FP3530 в проекта си;



- ✓ Можете да намерите новите компоненти в “Tool Palette”;



- ✓ Можете да намерите демо програми със сорскода в инсталационните папки;
- ✓ Това е всичко;

*Забележка за демо проекта свързан с интерфейс „CFD\_BGR“: е използван отворен сорскод от проекта JVCL/JCL. Сорскода на този проект може да бъде намерен на адрес: <https://github.com/project-jedi>, <https://github.com/TurboPack/SynEdit>, или посредством GetIt Package manager” в по-новите версии на Delphi.*

## Енумерирани типове

FP3530 – COM Server предлага всички enumerated types които му трябва вътрешно и които трябва да се използват при саответната употреба на методи или property.

## TMyLanguages

Клиентския софтуер може да избере/настрои човешкия език за съобщенията от страна на COM сървър по програмен начин. Освено възможните текстови съобщения (LastErrorMessage) тази настройка влияе и на текста който библиотеката връща като описание на статус битовите за даденото фискално устройство.

Възможни стойности:

- English
- Bulgarian

*Забележка: Обърнете внимание на демо проектите идващи с инсталацията.*

## TTransportProtocol

Клиентското приложение трябва да настрои типа на комуникация с устройството преди да опита да отвори връзка към него.

Възможни стойности:

- ctc\_RS232 – serial communication via RS-232 or USB;
- ctc\_TCPIP – TCP/IP communication via LAN/Internet

## TDeviceTypes

След успешна връзка с фискалното устройство, интерфейс „CFD\_BGR“ настройва съответната характеристика, за да можете да я използвате.

Възможни стойности:

- dtc\_FiscalPrinter – фискален принтер;
- dtc\_ECR – касов апарат;

## TDeviceModel

След успешна връзка с фискалното устройство, клиентските програми могат да получат информация от интерфейс „CFD\_BGR“ за модела на фискалното устройство. Това е особено полезно, както заради разликите в хардуера, така и заради разликите в някои от методите на фирмуера.

Възможни стойности:

Стойност	Модел	Логическа група „А“	Логическа група „В“	Логическа група „С“	Тип на пакетизираното съобщение	Брой статус битове	Стандартизиран код на грешка в отговора	Стандартизиран делимитер
<b>mc_Unknown</b>	Стойност на property <b>“device_Model”</b> преди отваряне на връзка към фискалното устройство.							
<b>mc_DP_05</b>	DP-05	x	✓	x	pmt_01	6	x	x
<b>mc_DP_15</b>	DP-15	x	✓	x	pmt_01	6	x	x
<b>mc_DP_25</b>	DP-25	x	✓	x	pmt_01	6	x	x
<b>mc_DP_35</b>	DP-35	x	✓	x	pmt_01	6	x	x
<b>mc_DP_150</b>	DP-150	x	✓	x	pmt_01	6	x	x
<b>mc_WP_50</b>	WP-50	x	✓	x	pmt_01	6	x	x
<b>mc_FP_650</b>	FP-650	✓	x	x	pmt_01	6	x	x
<b>mc_FP_800</b>	FP-800	✓	x	x	pmt_01	6	x	x
<b>mc_FP_2000</b>	FP-2000	✓	x	x	pmt_01	6	x	x
<b>mc_SK_21F</b>	SK-21F	✓	x	x	pmt_01	6	x	x
<b>mc_SK_31F</b>	SK-31F	✓	x	x	pmt_01	6	x	x
<b>mc_FMP_10</b>	FMP-10	✓	x	x	pmt_01	6	x	x
<b>mc_FP_550</b>	FP-550	✓	x	x	pmt_01	6	x	x
<b>mc_FP_700</b>	FP-700	✓	x	x	pmt_01	6	x	x
<b>mc_FP_700X</b>	FP-700X	x	x	✓	pmt_02	8	✓	✓
<b>mc_DP_25X</b>	DP-25X	x	x	✓	pmt_02	8	✓	✓
<b>mc_DP_150X</b>	DP-150X	x	x	✓	pmt_02	8	✓	✓
<b>mc_WP_50X</b>	WP-50X	x	x	✓	pmt_02	8	✓	✓
<b>mc_WP_500X</b>	WP-500X	x	x	✓	pmt_02	8	✓	✓
<b>mc_FMP_55X</b>	FMP-350X	x	x	✓	pmt_02	8	✓	✓
<b>mc_FMP_350X</b>	FMP-55X	x	x	✓	pmt_02	8	✓	✓
<b>mc_DP_05C</b>	DP-05C	x	x	✓	pmt_02	8	✓	✓



# TrpackagedMessageType

Интерфейс „CFD\_BGR“ предоставя информация за типа на пакетизираното съобщение след успешно отваряне на връзката към фискалното устройство.

Възможни стойности:

Стойност	Описание			
<b>pmt_Unknown</b>	Стойност на property за четене <b>“type_PackagedMessage”</b> преди отваряне на връзка към фискалното устройство.			
<b>pmt_01</b>	Стойност на property за четене <b>“type_PackagedMessage”</b> след отваряне на връзка към фискално устройство от следния списък:			
	Модел	Логическа група „А“	Логическа група „В“	Логическа група „С“
	FP-800	✓	х	х
	FP-2000	✓	х	х
	FP-650	✓	х	х
	SK1-21F	✓	х	х
	SK1-31F	✓	х	х
	FMP-10	✓	х	х
	FP-550	✓	х	х
	FP-700	✓	х	х
	DP-05	х	✓	х
	DP-15	х	✓	х
	DP-25	х	✓	х
	DP-35	х	✓	х
	WP-50	х	✓	х
	DP-150	х	✓	х
<b>pmt_02</b>	Стойност на property за четене <b>“type_PackagedMessage”</b> след отваряне на връзка към фискално устройство от следния списък:			
	Модел	Логическа група „А“	Логическа група „В“	Логическа група „С“
	FP-700X	х	х	✓
	DP-25X	х	х	✓
	DP-150X	х	х	✓
	WP-50X	х	х	✓
	WP-500X	х	х	✓
	FMP-350X	х	х	✓
	FMP-55X	х	х	✓
	DP-05C	х	х	✓

## TscriptType

Интерфейс „CFD\_BGR“ предоставя метод чрез който клиентското приложение може да изпълнява предварително подготвени текстови скриптове. Този тип се използва, за да може скрипт енджина на интерфейса да разбере какъв е „езика“ на скрипта.

Възможни стойности:

Стойност	Описание
• DS	Датекс скрипт.

***Забележка:** Датекс си запазва правото да добавя нови скриптови „езици“ и да разширява и подобрява механизма на обработка на скрипта.*

## TCodeType

„CFD\_BGR“ може да изпълнява команди, подадени от клиентското приложение по три начина:

- Чрез употребата на "execute\_Command"
- Чрез употребата на "execute\_Script\_V1"
- Чрез: "execute\_Command\_ByName"

В тази версия на „CFD\_BGR“ - библиотеката може да генерира примерен сорскод на два програмни езика за всички команди експортирани чрез метода "get\_ComandsList".

Възможни стойности:

Стойност	Описание
<b>Delphi</b>	Метода ще генерира сорс код на Delphi (beta)
<b>CSharp</b>	Метода ще генерира сорс код на C# (beta)

***Забележка:** В текущата бета версия на интерфейс „CFD\_BGR“, библиотеката генерира сорскод единствено на Pascal (Delphi) и C#. Генерирания сорскод не е „задължителен“, а служи за ориентиране – как да се използва "execute\_Command\_ByName". Можете да използвате или модифицирате както намерите за добре предложения от нас сорс код. Екипа на Датекс приема препоръки за подобрение, оптимизация и отстраняване на грешки. Ако желаете да ни помогнете, за да добавим и Вашия любим език за програмиране – свържете се с нас. Моля – приемете нашите извинения, ако в генерирания код има грешки към дадения момент.*

# Properties

## Статус на фискалното устройство

### Статус битове за грешка

В зависимост от типа на пакетираното съобщение, текущия статус се определя от отговора и/или от състоянието на статус битовите на устройството. При първи тип на пакетираното съобщение статуса на устройството се кодира в 6 статус байта. При втория тип на пакетираните съобщения, статуса на устройството се определя от кода на грешката върнат при изпълнение на команда и от състоянието на 8 статус байта. Интерфейс „CFD\_BGR“ предоставя „properties“ (само за четене) които връщат състоянието на определени статус битове, вдигнати в състояние 1 при грешка. Тези флагове се вдигат от страна на фискалното устройство в случай на грешка или при някои типове повреди в самото фискално устройство.

Вие трябва (в програмата Ви) да четете и използвате статуса на тези properties:

- След успешно установена връзка с фискалното устройство;
- След изпълнение на метод, команда или скрипт;

Property name	Property type	Status [byte,bit]	Human meaning
eSBit_GeneralError_Sharp	Boolean Read only	[0,5]	General error - this is OR of all errors marked with #
eSBit_PrintingMechanism	Boolean Read only	[0,4]	# Failure in printing mechanism.
eSBit_ClockIsNotSynchronized	Boolean Read only	[0,2]	The real time clock is not synchronized.
eSBit_CommandCodeIsInvalid	Boolean Read only	[0,1]	# Command code is invalid.
eSBit_SyntaxError	Boolean Read only	[0,0]	# Syntax error.
eSBit_CommandNotPermitted	Boolean Read only	[1,1]	# Command is not permitted.
eSBit_Overflow	Boolean Read only	[1,0]	# Overflow during command execution.
eSBit_EJIsFull	Boolean Read only	[2,2]	EJ is full.
eSBit_EndOfPaper	Boolean Read only	[2,0]	# End of paper.
eSBit_FM_NotFound	Boolean Read only	[4,6]	Fiscal memory not found or damaged.
eSBit_FM_NotAccess	Boolean Read only	[4,0]	* Error when trying to access data stored in the FM.
eSBit_FM_Full	Boolean Read only	[4,4]	* Fiscal memory is full.
eSBit_GeneralError_Star	Boolean Read only	[4,5]	OR of all errors marked with ‘*’

## Статус на устройството – информативни статус битове

В зависимост от типа на пакетираното съобщение, текущия статус се определя от отговора и/или от състоянието на статус битовете на устройството. При първи тип на пакетираното съобщение статуса на устройството се кодира в 6 статус байта. При втория тип на пакетираните съобщения, статуса на устройството се определя от кода на грешката върнат при изпълнение на команда и от състоянието на 8 статус байта. Интерфейс „CFD\_BGR“ предоставя „properties“ (само за четене) които връщат състоянието на определени статус битове, вдигнати в състояние 1, но не се смятат за грешка. Това са информативни битове които носят полезна информация за програмата Ви.

Вие трябва (в програмата Ви) да четете и използвате статуса на тези properties:

- След успешно установена връзка с фискалното устройство;
- След изпълнение на метод, команда или скрипт;

Property name	Property type	Status [byte,bit]	Human meaning
iSBit_Cover_IsOpen	Boolean Read only	[0,6]	Cover is open.
iSBit_No_ClientDisplay	Boolean Read only	[0,3]	No client display connected.
iSBit_Receipt_Nonfiscal	Boolean Read only	[2,5]	Nonfiscal receipt is open.
iSBit_EJ_NearlyFull	Boolean Read only	[2,4]	EJ nearly full.
iSBit_Receipt_Fiscal	Boolean Read only	[2,3]	Fiscal receipt is open.
iSBit_Near_PaperEnd	Boolean Read only	[2,1]	Near paper end.
iSBit_LessThan_50_Reports	Boolean Read only	[4,3]	There is space for less than 60 reports in Fiscal memory.
iSBit_Number_SFM_Set	Boolean Read only	[4,2]	Serial number and number of FM are set.
iSBit_Number_Tax_Set	Boolean Read only	[4,1]	Tax number is set.
iSBit_VAT_Set	Boolean Read only	[5,4]	VAT are set at least once.
iSBit_Device_Fiscalized	Boolean Read only	[5,3]	Device is fiscalized.
iSBit_FM_formatted	Boolean Read only	[5,1]	FM is formatted.

## Получаване на статус битовете на фискалното устройство чрез употреба методи

Интерфейс „CFD\_BGR“ предоставя няколко метода чрез които можете не само да получите състоянието и описанието на всеки статус бит, но и сами да определите или промените поведението на библиотеката по отношение на статус битовете.

### ***Брой на статус битовете за текущото фискално устройство***

Освен логически по таблицата с модели на фискалните устройства или от типа на пакетираният съобщение, Вие можете да получите броя на статус битовете от стойността на `read only property` - „`count_StatusBytes`“. Коректната стойност се попълва след успешно установяване на връзка с фискалното устройство.

### ***Метод „get\_Sbit\_State“***

Чрез този метод можете да получите текущото състояние на точно определен от Вас статус бит по индекси съответно за байт и за бит.

Състоянието на статус битовете се обновява вътрешно след изпълнението на всяка команда. Например, ако се опитате да изпълните команда за отваряне на бон с неправилно форматиран входни данни – статус бита за синтактична грешка ще се вдигне след обработката на отговора от страна на библиотеката. Можете да го прочетете с този метод многократно и всеки път (до момента на изпълнение на нова команда) статус бита ще е вдигнат в единица (метода връща за удобство `True/False` съответно за състояние 1/0). Ако след това изпълните валидна команда (например команда 74) – състоянието на този статус бит ще премине в 0, а този метод ще върне `False`, т.е. нямаме синтактична грешка при последната команда.

### ***Метод „get\_Sbit\_Description“***

Чрез този метод можете да получите текстовото описание на точно определен от Вас статус бит по индекси съответно за байт и за бит. Езика на който се връща текста зависи от избраната настройка.

### ***Метод „get\_Sbit\_ErrorChecking“***

Чрез този метод можете да получите какво ще бъде поведението на библиотеката по отношение на дадения статус бит. Ако метода върне **True** за даден статус бит – това означава, че библиотеката ще смята вдигането му в единица за флаг за грешка. Библиотеката ще върне информация, че съответния метод не е правилно изпълнен.

## **Метод „set\_Sbit\_ErrorChecking“**

Чрез този метод можете да определите какво да бъде поведението на библиотеката по отношение на даден статус бит. Например, ако желаете програмата да не работи, ако към фискалното устройство не е свързан дисплей – можете да настроите поведението на библиотеката по отношение на статус бита за свързан дисплей. Изпратете стойност **True** за този статус бит и библиотеката ще започне да връща грешка в статус битовите при изпълнението на който и да било метод, ако към фискалното устройство не е свързан дисплей.

Този метод е удобен и по отношение на „неочаквани“ промени в значението на даден статус бит в бъдещи фискални устройства. Вие ще можете да включвате или изключвате вдигането на грешка в зависимост от конкретната ситуация.

## Статус битове по групи фискални устройства

### Статус байт 0

Байт индекс	Бит индекс	Значение в логическа група „А“	Значение в логическа група „В“	Значение в логическа група „С“
0	7	Резервиран – винаги е 1	Резервиран – винаги е 1	Резервиран – винаги е 0.
	6	Отворен е капакът на принтера	Резервиран – винаги е 0.	Отворен е капакът на принтера
	5	Обща грешка - това е OR на всички грешки, маркирани с '#’.	Обща грешка - това е OR на всички грешки, маркирани с '#’.	Обща грешка - това е OR на всички грешки, маркирани с '#’
	4	(#) Механизмът на печатащото устройство има неизправност	Резервиран – винаги е 0.	(#) Механизмът на печатащото устройство има неизправност
	3	Не е свързан клиентски дисплей	Не е свързан клиентски дисплей.	Резервиран – винаги е 0.
	2	Не е сверен часовника.	Не е сверен часовника.	Не е сверен часовника.
	1	(#) Кодът на получената команда е невалиден	(#) Кодът на получената команда е невалиден	(#) Кодът на получената команда е невалиден
	0	(#) Получените данни имат синтактична грешка	(#) Получените данни имат синтактична грешка.	(#) Получените данни имат синтактична грешка.

### Статус байт 1

Байт индекс	Бит индекс	Значение в логическа група „А“	Значение в логическа група „В“	Значение в логическа група „С“
1	7	Резервиран – винаги е 1	Резервиран – винаги е 1	Резервиран – винаги е 1
	6	Вграденият данъчен терминал не отговаря	Вграденият данъчен терминал не отговаря	Резервиран – винаги е 0.
	5	Отворен е служебен бон за печат на завъртян на 90 градуса текст	Има неизпратени документи за повече от настроеното време за предупреждение	Резервиран – винаги е 0.
	4	Отворен сторно бон	Резервиран – винаги е 0.	Резервиран – винаги е 0.
	3	(#) Слаба батерия (Часовникът за реално време е в състояние RESET)	Резервиран – винаги е 0.	Резервиран – винаги е 0.
	2	(#) Извършено е зануляване на оперативната памет	Резервиран – винаги е 0.	Резервиран – винаги е 0.
	1	(#) Изпълнението на командата не е позволено в текущия фискален режим	(#) Изпълнението на командата не е позволено в текущия фискален режим	(#) Изпълнението на командата не е позволено в текущия фискален режим
	0	При изпълнение на командата се е получило препълване на някои полета от сумите. Статус 1.1 също ще се установи и командата няма да предизвика промяна на данните в принтера	При изпълнение на командата се е получило препълване на някои полета от сумите. Статус 1.1 също ще се установи и командата няма да предизвика промяна на данните в ФУ	При изпълнение на командата се е получило препълване на някои полета от сумите

## Статус байт 2

Байт индекс	Бит индекс	Значение в логическа група „А“	Значение в логическа група „В“	Значение в логическа група „С“
2	7	Резервиран – винаги е 1	Резервиран – винаги е 1.	Резервиран – винаги е 1.
	6	Много близък край на КЛЕН (допускат се само определени бонове).	Не се използва	Резервиран – винаги е 0.
	5	Отворен е служебен бон	Отворен е служебен бон	Отворен е служебен бон
	4	Близък край на КЛЕН (по-малко от 10 МВ от КЛЕН свободни).	Близък край на КЛЕН (по-малко от 10 МВ от КЛЕН свободни).	Близък край на КЛЕН.
	3	Отворен е фискален бон	Отворен е фискален бон	Отворен е фискален бон
	2	Край на КЛЕН (по-малко от 1 МВ от КЛЕН свободни)	Край на КЛЕН (по-малко от 1 МВ от КЛЕН свободни).	Край на КЛЕН (по-малко от 1 МВ от КЛЕН свободни).
	1	Останала е малко хартия	Резервиран – винаги е 0.	Останала е малко хартия
	0	(#) Свършила е хартията. Ако се вдигне този флаг по време на команда, свързана с печат, то командата е отхвърлена и не е променила състоянието на принтера	(#) Свършила е хартията. Ако се вдигне този флаг по време на команда, свързана с печат, то командата е отхвърлена и не е променила състоянието на ФУ.	(#) Свършила е хартията. Ако се вдигне този флаг по време на команда, свързана с печат, то командата е отхвърлена и не е променила състоянието на ФУ.

## Статус байт 3

Байт индекс	Бит индекс	Значение в логическа група „А“	Значение в логическа група „В“	Значение в логическа група „С“
3	7	Резервиран – винаги е 1.	Резервиран – винаги е 1	Резервиран – винаги е 1
	6	Състояние на Sw7	Резервиран – винаги е 0	Резервиран – винаги е 0
	5	Състояние на Sw6	Резервиран – винаги е 0	Резервиран – винаги е 0
	4	Състояние на Sw5	Резервиран – винаги е 0	Резервиран – винаги е 0
	3	Състояние на Sw4	Резервиран – винаги е 0	Резервиран – винаги е 0
	2	Състояние на Sw3	Резервиран – винаги е 0	Резервиран – винаги е 0
	1	Състояние на Sw2	Резервиран – винаги е 0	Резервиран – винаги е 0
	0	Състояние на Sw1	Резервиран – винаги е 0	Резервиран – винаги е 0



## Статус байт 4

Байт индекс	Бит индекс	Значение в логическа група „А“	Значение в логическа група „В“	Значение в логическа група „С“
4	7	Резервиран – винаги е 1.	Резервиран – винаги е 1	Резервиран – винаги е 1
	6	Печатащата глава е прегряла	Не се използва.	Фискалната памет липсва или е повредена
	5	OR на всички грешки, маркирани с ‘*’ от байтове 4 и 5	OR на всички грешки, маркирани с ‘*’ от байтове 4 и 5	OR на всички грешки, маркирани с ‘*’ от байтове 4 и 5
	4	(*) Фискалната памет е пълна	(*) Фискалната памет е пълна	(*) Фискалната памет е пълна
	3	Има място за по-малко от 50 записа във ФП	Има място за по-малко от 50 записа във ФП	Има място за по-малко от 60 записа във ФП
	2	Зададени са индивидуален номер на ФУ и номер на ФП	Зададени са индивидуален номер на ФУ и номер на ФП	Зададени са индивидуален номер на ФУ и номер на ФП
	1	Зададен е ЕИК	Зададен е ЕИК	Зададен е ЕИК
	0	(*) Има грешка при запис във фискалната памет	(*) Грешка при запис във фискалната памет	(*) Грешка при запис във фискалната памет

## Статус байт 5

Байт индекс	Бит индекс	Значение в логическа група „А“	Значение в логическа група „В“	Значение в логическа група „С“
5	7	Резервиран – винаги е 1	Резервиран – винаги е 1	Резервиран – винаги е 1
	6	Не се използва	Резервиран – винаги е 0	Резервиран – винаги е 0
	5	Грешка при четене от фискалната памет	Резервиран – винаги е 0	Резервиран – винаги е 0
	4	Зададени са поне веднъж данъчните ставки	Зададени са поне веднъж данъчните ставки.	Зададени са поне веднъж данъчните ставки.
	3	Устройството е във фискален режим	Устройството е във фискален режим	Устройството е във фискален режим
	2	(*) Последният запис във фискалната памет не е успешен	Резервиран – винаги е 0	Резервиран – винаги е 0
	1	Фискалната памет е форматирана	Фискалната памет е форматирана	Фискалната памет е форматирана
	0	(*) Фискалната памет е установена в режим READONLY (заклучена).	Резервиран – винаги е 0	Резервиран – винаги е 0

## Статус байт 6

Байт индекс	Бит индекс	Значение в логическа група „А“	Значение в логическа група „В“	Значение в логическа група „С“
6	7	x	x	Резервиран – винаги е 1
	6	x	x	Резервиран – винаги е 0
	5	x	x	Резервиран – винаги е 0
	4	x	x	Резервиран – винаги е 0
	3	x	x	Резервиран – винаги е 0
	2	x	x	Резервиран – винаги е 0
	1	x	x	Резервиран – винаги е 0
	0	x	x	Резервиран – винаги е 0

Статус байта е запазен за бъдеща употреба.

## Статус байт 7

Байт индекс	Бит индекс	Значение в логическа група „А“	Значение в логическа група „В“	Значение в логическа група „С“
7	7	x	x	Резервиран – винаги е 1
	6	x	x	Резервиран – винаги е 0
	5	x	x	Резервиран – винаги е 0
	4	x	x	Резервиран – винаги е 0
	3	x	x	Резервиран – винаги е 0
	2	x	x	Резервиран – винаги е 0
	1	x	x	Резервиран – винаги е 0
	0	x	x	Резервиран – винаги е 0

Статус байта е запазен за бъдеща употреба.

## Информация за устройството

Property name	Property type	Human meaning
connected_ToDevice	Boolean Read only	True, ако връзката към устройството е успешно осъществена.
support_RS232	Boolean Read only	True, ако устройството поддържа комуникация по RS232 (or USB).
support_TCPIP	Boolean Read only	True, ако устройството поддържа комуникация по LAN connector (TCP/IP protocol).
device_Type	TDeviceType Read only	<ul style="list-style-type: none"> <li>dt_FiscalPrinter за фискален принтер;</li> <li>dt_ECR за касов апарат;</li> </ul>
device_Number_Serial	WideString/BSTR Read only	Сериен номер на устройството.
device_Number_FMemory	WideString/BSTR Read only	Номер на фискалната памет.
device_Model	TDeviceModel Read only	<ul style="list-style-type: none"> <li>Device model – енумериран тип (модел на устройството);</li> <li>mc_Unknown, ако COM сървър не успява да разпознае модела на устройството;</li> </ul>
device_Model_Group	TdeviceGroup Read only	Всяко от фискалните устройства на Датекс за България попада логически в едно от три групи. Всяко от устройствата в дадена група има общ протокол и поведение с другите устройства от групата. device_Model_Group съдържа логическата група на устройството, след разпознаване на модела.
device_Model_Name	WideString/BSTR Read only	Името на модела (получено от устройството).
codePage	Integer/Long Read only	Стойност на „Code page”, използвана от устройството
device_Firmware_Revision	WideString/BSTR Read only	Ревизия на фирмуера.
device_Firmware_Date	WideString/BSTR Read only	Дата и час на ревизията на фирмуера.
device_Firmware_CheckSum	WideString/BSTR Read only	Чексума на ревизията на фирмуера.

## Комуникация

Тази секция съдържа информация относно properties които са свързани с поведението на COM сървъра при опит за отваряне на връзка или по време на комуникация с фискалното устройство.

Property name	Property type	Human meaning
<b>protocol_TransportType</b>	TTransportProtocol Read only	Клиентското приложение трябва да настрои типа на транспортния протокол чрез изпълнението на метод "set_TransportType" преди опита за отваряне на връзка към фискалното устройство. <ul style="list-style-type: none"> <li>ctc_RS232 – if connection is via (RS-232/USB)</li> <li>ctc_TCPIP – if connection is via LAN (TCP/IP)</li> </ul>
<b>tcpip_Address</b>	WideString/BSTR Read only	Стойност на IP address за връзка с фискалното устройство.
<b>tcpip_Port</b>	Integer/Long Read only	Стойност на TCP/IP port за връзка с фискалното устройство.
<b>rs232_ComPort</b>	Integer/Long Read only	Стойност на COM port за връзка с фискалното устройство.
<b>rs232_BaudRate</b>	Integer/Long Read only	Скорост на комуникация по RS-232 за връзка с фискалното устройство.
<b>read_TimeOutValue</b>	Word Read/Write	Глобален timeout за комуникацията с фискалното устройство. Стойности по подразбиране: <ul style="list-style-type: none"> <li>1000 mSec for RS-232</li> <li>3000 mSec for TCP/IP</li> </ul>
<b>exit_ByReadTimeoutIsOn</b>	Boolean Read/Write	Ако тази х-ка има стойност true и ако при комуникация няма отговор за период по-голям от глобалния таймаут в mSec - "CFD_BGR" ще спре изчакването на отговор от фискалното устройство и ще върне „timeout error“ за текущото изпълнение на командата. По подразбиране стойността е true.
<b>rs232_ReadIntervalTimeout</b>	Integer/Long Read only	Part of commtimeouts structure. The maximum time allowed to elapse before the arrival of the next byte on the communications line, in milliseconds. If the interval between the arrival of any two bytes exceeds this amount, the ReadFile operation is completed and any buffered data is returned. A value of zero indicates that interval time-outs are not used. A value of MAXDWORD, combined with zero values for both the ReadTotalTimeoutConstant and ReadTotalTimeoutMultiplier members, specifies that the read operation is to return immediately with the bytes that have already been received, even if no bytes have been received.
<b>rs232_ReadTotalTimeoutMultiplier</b>	Integer/Long Read only	Part of commtimeouts structure. The multiplier used to calculate the total time-out period for read operations, in milliseconds. For each read operation, this value is multiplied by the requested number of bytes to be read.
<b>rs232_ReadTotalTimeoutConstant</b>	Integer/Long Read only	Part of commtimeouts structure. A constant used to calculate the total time-out period for read operations, in milliseconds. For each read operation, this value is added to the product of the ReadTotalTimeoutMultiplier member and the requested number of bytes. A value of zero for both the ReadTotalTimeoutMultiplier and ReadTotalTimeoutConstant members indicates that total time-outs are not used for read operations.
<b>rs232_WriteTotalTimeoutMultiplier</b>	Integer/Long Read only	Part of commtimeouts structure. The multiplier used to calculate the total time-out period for write operations, in milliseconds. For each write operation, this value is multiplied by the number of bytes to be written.
<b>rs232_WriteTotalTimeoutConstant</b>	Integer/Long Read only	Part of commtimeouts structure. A constant used to calculate the total time-out period for write operations, in milliseconds. For each write operation, this value is added to the product of the WriteTotalTimeoutMultiplier member and the number of bytes to be written. A value of zero for both the WriteTotalTimeoutMultiplier and WriteTotalTimeoutConstant members indicates that total time-outs are not used for write operations.

<b>rs232_OnOpen_Set_DCB</b>	Boolean Read/Write	When this value is true – „CFD_BGR“ will try to set DCB structure during the opening of the COM port. Default value is true.
<b>rs232_OnOpen_Set_DTR_ToFalse</b>	Boolean Read/Write	When this value is true – „CFD_BGR“ will try to set DTR to false during the opening of the COM port. Clears the DTR (data-terminal-ready) signal. Default value is true.
<b>rs232_OnOpen_Set_RTS_ToFalse</b>	Boolean Read/Write	When this value is true – „CFD_BGR“ will try to set RTS to false during the opening of the COM port. Clears the RTS (request-to-send) signal. Default value is true.
<b>connected_ToLAN</b>	Boolean Read only	A return value of TRUE indicates that either the modem connection is active, or a LAN connection is active and a proxy is properly configured for the LAN. It does not guarantee that a connection to a specific host can be established. A return value of FALSE indicates that neither the modem nor the LAN are connected.

## Общо поведение

Property name	Property type	Human meaning
language	TMyLanguages Read only	<ul style="list-style-type: none"> <li>Bulgarian</li> <li>English</li> </ul>
trackingMode	Boolean Read only	If the value is true – „CFD_BGR“ will try to save the communication with the fiscal device in a file according to the values of other tracking properties.
trackingMode_Path	WideString/BSTR Read only	The path to the log file when „CFD_BGR“ is in tracking mode.
trackingMode_FileName	WideString/BSTR Read only	If „CFD_BGR“ is in tracking mode – it will try to create and save the log info a text file with this name.
trackingMode_RowLimit	Integer/Long Read only	This value is a limit of the number of text rows into the log file. „CFD_BGR“ accept values between 100 and 5000.
active_OnBeforeScriptExecute	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event before the execution of the script.
active_OnScriptRowExecute	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after execution of the command from the text row into the script.
active_OnAfterScriptExecute	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after the execution of the script.
active_OnFirstProgress_Init	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event before starting an operation which is a loop of commands. It is usable for initialization of progress bar for example.
active_OnFirstProgress_Loop	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after the execution of a command during to the loop of commands.
active_OnFirstProgress_Complete	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after finishing an operation which is a loop of commands.
active_OnSecondProgress_Init	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event before starting an operation which is a loop of commands. It is usable for initialization of progress bar for example.
active_OnSecondProgress_Loop	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after the execution of a command during to the loop of commands.
active_OnSecondProgress_Complete	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after finishing an operation which is a loop of commands.
active_OnSendCommand	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after the sending of packet message to the fiscal device.
active_OnWait	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after receiving SYN byte from the fiscal device.
active_OnReceiveAnswer	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after receiving of packet message from the fiscal device.
active_OnStatusChange	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event after receiving a packet message from the fiscal device.
active_OnError	Boolean Read only	If this value is true – „CFD_BGR“ will fire an event if an error occurs.

## Други

Property name	Property type	Human meaning
<b>lastError_Code</b>	Integer/Long Read only	Contains the current value of the last error code of the „CFD_BGR“.
<b>lastError_Message</b>	WideString/BSTR Read only	Contains the current value of the last error message of the „CFD_BGR“.
<b>last_AnswerList</b>	WideString/BSTR Read only	Contains the current value of the last answer from the packaged message received from the fiscal device. The values are separated with CRLF. (bytes with the decimal values 13 and 10. "Carriage Return" and "Line Feed").
<b>download_Path</b>	WideString/BSTR Read only	Contains the current value of the path for download which the „CFD_BGR“ driver will try to use in operations for downloading ANAF files from the fiscal device.
<b>DateRange_StartValue</b>	WideString/BSTR Read/Write	For future usages
<b>DateRange_EndValue</b>	WideString/BSTR Read/Write	For future usages

## Събития (Events)

По дизайн събитията които интерфейс „CFD\_BGR“ вдига са предназначени за чисто информативни цели. За удобство по отношение на потребителския интерфейс. По време на изпълнение на команда може да бъдат вдигнати повече от едно събития. Не използвайте кодовете за грешка идващи чрез събития за управление на изпълнението на командите! Възможно е да дойдат асинхронно във времето.

### OnError

„CFD\_BGR“ ще вдигне това събитие при възникване на грешка.

Забележка: Ако изпълнението на командата е неуспешно и метода който използвате връща резултат от тип integer – стойността на **"error\_Code"** най-вероятно ще бъде идентична със стойността на резултата от изпълнението на метода. *Ако изпълнението е успешно – стойността на резултата от метода и стойността на „last error code“ ще бъдат равни на нула, а събитието „OnError“ няма да бъде вдигнато.*

Параметри:

Name	Type	Description
error_Code	Integer	The value of the error code depending of the error.
error_Message	WideString	The text error message.

### OnBeforeScriptExecute event

„CFD\_BGR“ ще вдигне това събитие преди изпълнение на скрипт.

Параметри: **Няма.**



## OnScriptRowExecute event

„CFD\_BGR“ ще вдигне това събитие след изпълнението на команда от текстова линия в даден скрипт.

Параметри:

Name	Type	Description
row_Index	Integer	The index shows the line number depending on the beginning of the script. The index is zero based.
error_Code	Integer	The value of the error code depending of the error. An error code with value 0 means that there no error during the execution of the command.
input_Value	WideString	The value of the executed text line.
output_Value	WideString	The answer from the device side.

## OnAfterScriptExecute event

„CFD\_BGR“ ще вдигне това събитие след изпълнението на скрипт.

Параметри: **None**

## OnSendCommand event

„CFD\_BGR“ ще вдигне това събитие след изпращане на пакетизирано съобщение към фискалното устройство.

Параметри:

Name	Type	Description
Command	WideString	The value of the command in decimal.
DateAndTime	WideString	Date and time of execution in format 'dd.mm.yyyy hh:mm:ss:zzz'
repeat_Value	WideString	By the low level communication protocol – if command failed or if fiscal device asked the host to repeat the command, the host must send the same packet message to the fiscal device. The value of this parameter contains the current index of the attempts.
hex_Header	WideString	The header part from the packet message to the fiscal device in hex values.
hex_Data	WideString	The logical data part from the packet message to the fiscal device in hex values.
hex_Footer	WideString	The footer part from the packet message to the fiscal device in hex values.
human_Data	WideString	The logical data part from the packet message to the fiscal device before to convert into a hex values.

## OnWait event

„CFD\_BGR“ ще вдигне това събитие след получаване на байт със стойност SYN от страна на фискалното устройство.

Параметри:

Name	Type	Description
Value	Byte	Must contain value equal to SYN

## OnReceiveAnswer event

„CFD\_BGR“ ще вдигне това събитие след получаване на пакетизирано съобщение от страна на фискалното устройство.

Параметри:

Name	Type	Description
Command	WideString	The value of the command in decimal.
DateAndTime	WideString	Date and time of execution in format 'dd.mm.yyyy hh:mm:ss:zzz'
repeat_Value	WideString	By the low level communication protocol – if command failed or if fiscal device asked the host to repeat the command, the host must send the same packet message to the fiscal device. The value of this parameter contains the current index of the tries.
hex_Header	WideString	The header part from the packet message from the fiscal device in hex values.
hex_Data	WideString	The logical data part from the packet message from the fiscal device in hex values.
hex_Footer	WideString	The footer part from the packet message from the fiscal device in hex values.
human_Data	WideString	The logical data part from the packet message from the fiscal device as a human text.

## OnStatusChange event

„CFD\_BGR“ ще вдигне това събитие след получаване на пакетизирано съобщение от страна на фискалното устройство. Това събитие е използваемо като тригер. След вдигане на събитието, клиентската програма може да прочете статуса на фискалното устройство от статус битовете (или от статус properties описани по-горе). Това е възможно, защото всички те са вече обновени вътрешно.

Параметри: **Няма**

## OnFirstProgress\_Init event

„CFD\_BGR“ ще вдигне това събитие преди стартиране на операция която се състои от цикъл команди. Удобна е за инициализация на някакъв прогрес (progress bar).

Параметри:

Name	Type	Description
value_Minimum	Integer	The calculated minimum value for the initialization.
value_Maximum	Integer	The calculated maximum value for the initialization.
value_Position	Integer	Current position for the initialization.

## OnFirstProgress\_Loop event

„CFD\_BGR“ ще вдигне това събитие след изпълнението на команда в тялото на вътрешен за библиотеката цикъл.

Параметри:

Name	Type	Description
value_Position	Integer	Current position of the progress.

## OnFirstProgress\_Complete event

„CFD\_BGR“ ще вдигне това събитие след финализиране на вътрешен за библиотеката цикъл.

Параметри: **Няма**

## OnSecondProgress\_Init event

„CFD\_BGR“ ще вдигне това събитие преди стартиране на операция която се състои от цикъл команди. Удобна е за инициализация на някакъв прогрес (progress bar).

Параметри:

Name	Type	Description
value_Minimum	Integer	The calculated minimum value for the initialization.
value_Maximum	Integer	The calculated maximum value for the initialization.
value_Position	Integer	Current position for the initialization.

## OnSecondProgress\_Loop event

„CFD\_BGR“ ще вдигне това събитие след изпълнението на команда в тялото на вътрешен за библиотеката цикъл.

Параметри:

Name	Type	Description
value_Position	Integer	Current position of the progress.

## OnSecondProgress\_Complete event

„CFD\_BGR“ ще вдигне това събитие след финализиране на вътрешен за библиотеката цикъл.

Параметри: **Няма**

## Методи

### Режим проследяване / Tracking mode

За нуждите на разработките „CFD\_BGR“ поддържа вътрешен механизъм за проследяване и записване на комуникацията с фискалното устройство. Ако разработчиците искат да проследят в подробности комуникацията между Host (PC) и Slave (фискалното устройство), те могат да активират този режим. „CFD\_BGR“ ще записва комуникацията в двете посоки в даден файл. Ако има съмнение за грешка или проблем – екипа по поддръжката на библиотеката вероятно ще поиска да активирате този режим, да повторите стъпка по стъпка действията довели до проблема и да изпратите лог файла за анализ. Разбира се – добре е преди това самите разработчици да го разгледат – много често проблема се вижда веднага.

#### set\_TrackingMode\_RowLimit

Изпълнението на този метод ще настрои горната граница на броя текстови редове които ще се записват в лог файла. „CFD\_BGR“ приема стойности между 100 и 5000.

Параметри:

Name	Type	Description
Value	Integer / Long	The limit of the number of text rows into the log file. „CFD_BGR“ accept values between 100 and 5000.
Result	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

#### set\_TrackingMode\_Path

Изпълнението на този метод ще настрои пътя (папката) където лог файла ще се записва когато „CFD\_BGR“ е в режим проследяване.

Параметри:

Name	Type	Description
Value	WideString BSTR	The path to the log file when „CFD_BGR“ is in tracking mode
Result	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## set\_TrackingMode\_FileName

Изпълнението на този метод ще настрои името на лог файла който „CFD\_BGR“ ще се опита да създаде и използва в режим проследяване.

Параметри:

Name	Type	Description
Value	WideString BSTR	The log file name
Result	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## set\_TrackingMode

Изпълнението на този метод ще активира или деактивира режим проследяване. При стойност true – „CFD\_BGR“ ще опитва да записва комуникацията с фискалното устройство във файл и папка съобразно стойностите на описаните по-горе tracking properties.

Параметри:

Name	Type	Description
Value	Boolean	If the value is true, the tracking mode will be activated.
Result	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## Комуникация / Communication

### set\_TransportType

Execution of this function will set the transport protocol type. The client application must set the type of the communication transport protocol before opening of the connection to the fiscal device.

Parameters:

Name	Type	Description
Value	TTransportProtocol	The client application must set the type of the transport protocol before opening the connection to the fiscal device. <ul style="list-style-type: none"><li>ctc_RS232 – if connection is via (RS-232/USB)</li><li>ctc_TCPIP – if connection is via LAN (TCP/IP)</li></ul> For current value – check the value of the corresponding property: "protocol_TransportType".
Result	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the "CFD_BGR" engine.

### set\_TCPIP

Execution of this function will set the values needed for successful execution of the method "open\_Connection" if the chosen protocol type is ctc\_TCPIP. The client application must set the type of the communication transport protocol, the IPAddress and the Port before trying to open the connection to the fiscal device.

Parameters:

Name	Type	Description
IPAddress	WideString BSTR	The IP Address of the fiscal device.
Port	Integer / Long	The Port number for TCP/IP communications of the fiscal device.
Result	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

### set\_RS232

Execution of this function will set the values needed for successful execution of the method "open\_Connection" if the chosen protocol type is ctc\_RS232. The client application must set the type of the communication transport protocol, the ComPort and the BaudRate before trying to open the connection to the fiscal device.



Parameters:

Name	Type	Description
<b>ComPort</b>	Integer / Long	The ComPort.
<b>BaudRate</b>	Integer / Long	The baud rate – must be the same as into the fiscal device.
<b>Result</b>	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## set\_RS232\_Timeouts

Execution of this function will set the values of the commtimeouts structure. Do not use it if you don't know how or if „CFD\_BGR“ is working properly with the default values.

Parameters:

Name	Type	Description
<b>ReadIntervalTimeout</b>	LongWord	Part of commtimeouts structure. The maximum time allowed to elapse before the arrival of the next byte on the communications line, in milliseconds. If the interval between the arrival of any two bytes exceeds this amount, the ReadFile operation is completed and any buffered data is returned. A value of zero indicates that interval time-outs are not used. A value of MAXDWORD, combined with zero values for both the ReadTotalTimeoutConstant and ReadTotalTimeoutMultiplier members, specifies that the read operation is to return immediately with the bytes that have already been received, even if no bytes have been received.
<b>ReadTotalTimeoutMultiplier</b>	LongWord	Part of commtimeouts structure. The multiplier used to calculate the total time-out period for read operations, in milliseconds. For each read operation, this value is multiplied by the requested number of bytes to be read.
<b>ReadTotalTimeoutConstant</b>	LongWord	Part of commtimeouts structure. A constant used to calculate the total time-out period for read operations, in milliseconds. For each read operation, this value is added to the product of the ReadTotalTimeoutMultiplier member and the requested number of bytes. A value of zero for both the ReadTotalTimeoutMultiplier and ReadTotalTimeoutConstant members indicates that total time-outs are not used for read operations.
<b>WriteTotalTimeoutMultiplier</b>	LongWord	Part of commtimeouts structure. The multiplier used to calculate the total time-out period for write operations, in milliseconds. For each write operation, this value is multiplied by the number of bytes to be written.
<b>WriteTotalTimeoutConstant</b>	LongWord	Part of commtimeouts structure. A constant used to calculate the total time-out period for write operations, in milliseconds. For each write operation, this value is added to the product of the WriteTotalTimeoutMultiplier member and the number of bytes to be written. A value of zero for both the WriteTotalTimeoutMultiplier and WriteTotalTimeoutConstant members indicates that total time-outs are not used for write operations.
<b>Result</b>	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the “CFD_BGR“ engine.

## rs232\_COMPortList

This method returns a list of visible for „CFD\_BGR“ engine COM ports. If your COM port is not here – most probably the „CFD\_BGR“ engine isn't using it properly. After the adding of new USB device or hardware (new COM port) – this method must be executed again to refresh the info of the client application.

Parameters:

Name	Type	Description
<b>Result</b>	WideString BSTR	This method returns a list of visible for „CFD_BGR“ engine COM ports.

## open\_Connection

This function opens a connection to the fiscal device according to the other communication properties set before the execution of the command.

Parameters:

Name	Type	Description
<b>Result</b>	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## close\_Connection

This function closes the connection to the fiscal device.

Parameters:

Name	Type	Description
<b>Result</b>	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## Error support

You can find a full list of the possible errors returned from the “CFD\_BGR“ engine or from the fiscal device in the document: "ErrorCodes.xls". This document is located in the folder DOCUMENTATION, which is a sub-folder of the root of the installation folder.

### get\_ErrorMessageByCode

The client applications can get the error message by given code.

Parameters and result:

Name	Type	Description
Value	Integer/Long	The value of the code.
Result	WideString BSTR	The error message for a given code.

### get\_Sbit\_State

The client applications can get the state of any given status bit after each answer of the fiscal device.

Parameters and result:

Name	Type	Description
byteIndex	Byte	The index of the target status byte.
bitIndex	Byte	The index of the target status bit.
Result	Boolean	The result = true – mean that the status bit is raised. <i>Example:</i> <i>if get_SBit_State(0,0) then ShowMessage('Syntax error!');</i>

## get\_Sbit\_Description

The client applications can get the description text for any given status bit.

Parameters and result:

Name	Type	Description
<b>byteIndex</b>	Byte	The index of the target status byte.
<b>bitIndex</b>	Byte	The index of the target status bit.
<b>Result</b>	WideString BSTR	The description text for the given status bit. <i>Example: ShowMessage(get_SBit_Description(0,0));</i>

## get\_Sbit\_ErrorChecking

The client applications can get the description text for any given status bit.

Parameters and result:

Name	Type	Description
<b>byteIndex</b>	Byte	The index of the target status byte.
<b>bitIndex</b>	Byte	The index of the target status bit.
<b>Result</b>	Boolean	If the result of the function is true - this means that the „CFD_BGR“ engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## set\_Sbit\_ErrorChecking

The client application can change behavior of the „CFD\_BGR“ engine according to the needs of the project. If you think that the raising of some status bit is not an error – you can switch off its checking.

*Note: Change of the checking mean that only the “CFD\_BGR“ engine will stop checking given status. The fiscal device will continue raising the given status bit and will continue to return errors less than zero if command is not appropriate or if there is a syntax error, for example. **This function is developed for the needs of service applications and We do not recomend you to use it if you are not sure what we are talking about.** One more time: If the fiscal device "thinks" that given command is an error – it will not execute the command.*

Parameters and result:

Name	Type	Description
byteIndex	Byte	The index of the target status byte.
bitIndex	Byte	The index of the target status bit.
Value	Boolean	True or False if you want to switch on or switch off the checking of given status bit.
Result	Boolean	If the result of the function is true - this mean that the „CFD_BGR“ engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## Behavior

The client applications can change the behavior of the „CFD\_BGR“ in some aspects. For example they can activate or deactivate the raising of the events in the different cases.

### set\_ScriptEvents

With execution of this method the client applications can activate or deactivate the raising of the following event:

- **OnBeforeScriptExecute;**
- **OnScriptRowExecute;**
- **OnAfterScriptExecute;**

Parameters and result:

Name	Type	Description
<b>active_OnBeforeScriptExecute</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnScriptRowExecute</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnAfterScriptExecute</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>save_ToSettings</b>	Boolean	If the value of this parameter is true – the „CFD_BGR“ engine will try to save the changes into its settings file. If the value is not true – the changes take effect only for the current use of the COM server.
<b>Result</b>	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## set\_FirstProgressEvents

With execution of this method the client applications can activate or deactivate the raising of the following event:

- **OnFirstProgress\_Init;**
- **OnFirstProgress\_Loop;**
- **OnFirstProgress\_Complete;**

Parameters and result:

Name	Type	Description
<b>active_OnFirstProgress_Init</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnFirstProgress_Loop</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnFirstProgress_Complete</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>save_ToSettings</b>	Boolean	If the value of this parameter is true – the „CFD_BGR“ engine will try to save the changes into its settings file. If the value is not true – the changes take effect only for the current use of the COM server.
<b>Result</b>	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.



## set\_SecondProgressEvents

With execution of this method the client applications can activate or deactivate the raising of the following event:

- **OnSecondProgress\_Init;**
- **OnSecondProgress\_Loop;**
- **OnSecondProgress\_Complete;**

Parameters and result:

Name	Type	Description
<b>active_OnSecondProgress_Init</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnSecondProgress_Loop</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnSecondProgress_Complete</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>save_ToSettings</b>	Boolean	If the value of this parameter is true – the „CFD_BGR“ engine will try to save the changes into its settings file. If the value is not true – the changes take effect only for the current use of the COM server.
<b>Result</b>	Integer Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## set\_CommunicationEvents

With execution of this method the client applications can activate or deactivate the raising of the following event:

- **OnSendCommand;**
- **OnWait;**
- **OnReceiveAnswer;**
- **OnStatusChange;**
- **OnError;**

Parameters and result:

Name	Type	Description
<b>active_OnSendCommand</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnWait</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnReceiveAnswer</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnStatusChange</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>active_OnError</b>	Boolean	The value of the parameter activates or deactivates the corresponding event.
<b>save_ToSettings</b>	Boolean	If the value of this parameter is true – the „CFD_BGR“ engine will try to save the changes into its settings file. If the value is not true – the changes take effect only for the current use of the COM server.
<b>Result</b>	Integer / Long	Result is equal to zero if method is successfully executed and less than zero if error occurs. If the result is negative – the value contains the last error code from the „CFD_BGR“ engine.

## Services

### *cancel\_Loop*

With execution of this method – you can tell the „CFD\_BGR“ engine to stop the loop process and after that to exit the current execution of the method.

Parameters: **None**

Result: **None**

Note: For future usages

### *upload\_Logo*

This method sends a logo file to the fiscal device. The file must be with proper size, monochrome and with “bmp” extension. The fiscal devices not accept images different than bmp.

During the execution of the methods „CFD\_BGR“ will fire more than one time many of the events, so if the uploading of the logo files from client application must be accelerated as a process – the application can deactivate some of the events in the beginning. After the execution of the method the application can switch them back on to an active state.

Parameters:

Name	Type	Description
<b>FileName</b>	WideString BSTR	The path and file name to the logo file.
<b>Result</b>	WideString BSTR	A system information according to the input search value.

Note: For future usages

## SystemInfo

### *get\_SystemInfoSearchList*

This method returns a list of system values which can be used as a search parameter into the method "**get\_SystemInfo**". These two methods are suggested for the convenience of programmers.

Parameters:

Name	Type	Description
<b>Result</b>	WideString BSTR	A list of system values which can be used as a search parameter with the method "get_SystemInfo".

### *get\_SystemInfo*

The programmers can use this method as it is convenient. It shows information about the current PC configuration, folders and so on in the way that the "CFD\_BGR" sees the system.

Parameters:

Name	Type	Description
<b>SearchValue</b>	WideString BSTR	One of the items returned from the method: " <b>get_SystemInfoSearchList</b> ".
<b>Result</b>	WideString BSTR	A system information according to the input search value.

## Общи методи

### execute\_Command

Common method for sending commands to the fiscal device. The execution of the command can fire one or more of the following events if they are not switched off programmatically:

- OnSendCommand;
- OnWait;
- OnReceiveAnswer;
- OnStatusChange;
- OnError;

Parameters and result:

Name	Type	Description
<b>Command</b>	Integer/Long	The value of the command in the decimal number system.
<b>input_Value</b>	WideString BSTR	<p>The logical data of the command – formatted according to the protocol of the fiscal devices.</p> <p>You can find a detailed description of the commands to the fiscal device in the documents:</p> <ul style="list-style-type: none"><li>• In English: FP_Protocol_EN.pdf</li><li>• In Bulgarian: FP_Protocol_BG.pdf</li></ul> <p>These documents are located in the folder DOCUMENTATION which is a sub-folder of the root of the installation folder.</p>
<b>output_Value</b>	WideString BSTR	Text which contains the logical data from the answer from the fiscal device. The answer is formatted according to the protocol of the fiscal devices. You can parse the answer with your own parser or use the parsed answer from the property "last_AnswerList".
<b>Result</b>	Boolean	If the result of the function is true - this mean that the “CFD_BGR” engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## execute\_Script\_V1

Common method for sending a set of commands to the fiscal device. The client application can execute pre-prepared text scripts through this method. The execution of the command can fire one or more of the following events if they are not switched off programmatically:

- OnBeforeScriptExecute;
- OnScriptRowExecute;
- OnAfterScriptExecute;
- OnSendCommand;
- OnWait;
- OnReceiveAnswer;
- OnStatusChange;
- OnError;

Parameters and result:

Name	Type	Description
<b>scriptType</b>	TScriptType	This value is used from the script engine. "CFD_BGR" must know the language of the incoming script. Possible values: <ul style="list-style-type: none"><li>• DS</li></ul>
<b>input_Value</b>	WideString BSTR	The script text which contains the complete set of needed commands. Format of the text row: <i>&lt;cmd&gt;,&lt;logical data for the command&gt;</i> <ul style="list-style-type: none"><li>• cmd – the value of the command in the decimal number system;</li><li>• logical data of the command – formatted according to the protocol of the fiscal devices.</li></ul> You can find a detailed description of the commands to the fiscal device in the documents: <ul style="list-style-type: none"><li>• In English: FP_Protocol_EN.pdf</li><li>• In Bulgarian: FP_Protocol_BG.pdf</li></ul> These documents are located in the folder DOCUMENTATION which is a sub-folder of the root of the installation folder.
<b>Result</b>	Boolean	If the result of the function is true - this mean that the "CFD_BGR" engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## get\_ComandsList

The “CFD\_BGR“ engine contains an internal description of the commands for the given fiscal device as a collection of objects and the client applications can use them. This method returns a list with “human names” of the commands from the protocol of the fiscal device. In this version of the “CFD\_BGR“ engine – the programmers can choose from three different names for each of the commands and use them according to their own preferences.

Parameters:

Name	Type	Description
<b>Index</b>	Integer/Long	A zero based index for the internal search machine. Use the method with values of index between 0 and 2.
<b>Result</b>	WideString BSTR	This method returns a list of command names accordingly to the used index value.

Format description of the “human names”:

- index [0]: <cmd>\_<group\_name>\_<name>. *Example: 038\_receipt\_NonFiscal\_Open;*
- index [1]: <group\_name>\_<name>. *Example: receipt\_NonFiscal\_Open;*
- index [2]: <name>. *Example: NonFiscal\_Open;*

In some cases the “human names” for index 1 and 2 can be identical. You can easily find the detailed description for a given command in the documentation of the low level protocol by searching for a given **cmd** value and, of course, the **name** value.

## get\_CommandInfo

By using this method the client applications can get from the “CFD\_BGR” engine some general information about the command and its parameters. Look at the source code of the demo programs for a demonstration of the method. This method is developed for the programmers as a fast way of getting information about the command and its input or output parameters. Of course – all of these parameters are described in the low level protocol documentation but here you can find information on the limit values of the parameters used by the “CFD\_BGR” engine.

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	The one of the "human names" of the given command exported from the method “get_CommandsList”.
<b>Value</b>	WideString BSTR	Text with a common information of the command and its input or output parameters.
<b>Result</b>	Integer/Long	If the result of the function is true – this means that the “CFD_BGR” engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## get\_InputParams\_Count

Through this method – the programmers can receive the count of the input parameters for a given command.

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command "get_CommandsList" or from the document "CommandsList.xls".
<b>Value</b>	Integer/Long	The count of the input parameters for a given command.
<b>Result</b>	Integer/Long	If the result of the function is true - this means that the “CFD_BGR” engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.



## get\_InputParams\_Names

Through this method – the programmers can receive the list of the input parameter names for a given command.

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command " <b>get_CommandsList</b> " or from the document " <b>CommandsList.xls</b> ".
<b>Value</b>	WideString BSTR	The list of input parameter names accordingly to the used “ <b>command_Name</b> ” value.
<b>Result</b>	Integer/Long	If the result of the function is true - this mean that the “CFD_BGR” engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## set\_InputParam\_ByIndex

If you want to use the method “**execute\_Command\_ByName**” you must ensure that all input parameters of the command are set properly before the execution of the command. One of the ways to do that is to use this method.

*Notes:*

- *In some cases the value of the input parameter must be empty. For example, look at command 255 in the case of reading of the value from fiscal device.*
- *In other cases the input parameter is optional – so depending on the case, the input parameter may or may not need to be empty.*
- *You must be sure that you set the values of all input parameters properly. The “CFD\_BGR” engine keeps the input parameters into the memory so in some cases you probably do not want to execute the method with no set new values (using the old values).*

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command "get_CommandsList" or from the document "CommandsList.xls".
<b>param_Index</b>	Integer/Long	A zero based index related to the input parameters list of this command.
<b>Value</b>	WideString BSTR	The value of the input parameter. Check the low level protocol to be sure that you will fill the value of this parameter properly.
<b>Result</b>	Integer/Long	If the result of the function is true - this mean that the “CFD_BGR” engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## set\_InputParam\_ByName

If you want to use the method “**execute\_Command\_ByName**” you must ensure that all input parameters of the command are set properly before the execution of the command. One of the ways to do that is to use this method.

*Notes:*

- *In some cases the value of the input parameter must be empty. For example look at command 255 in the case of reading of the value from fiscal device.*
- *In other cases the input parameter is optional – so according to the case, the input parameter may or may not need to be empty.*
- *You must be sure that you set the values of all input parameters properly. The “CFD\_BGR” engine keeps the input parameters in the memory so in some cases you probably do not want to execute the method with no set new values (using the old values).*

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command "get_CommandsList" or from the document "CommandsList.xls".
<b>param_Name</b>	WideString BSTR	The input parameter name.
<b>Value</b>	WideString BSTR	The value of the input parameter. Check the low level protocol to be sure that you will fill the value of this parameter properly.
<b>Result</b>	Integer/Long	If the result of the function is true - this mean that the “CFD_BGR” engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## execute\_Command\_ByName

The “CFD\_BGR“ engine contains an internal description of the commands for the given fiscal device as a collection of objects and the client applications can use them. Set the input parameters of the command properly – before the execution of the method.

The execution of the command can fire one or more of the following events if they are not switched off programmatically:

- OnSendCommand;
- OnWait;
- OnReceiveAnswer;
- OnStatusChange;
- OnError;

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command "get_CommandsList" or from the document "CommandsList.xls".
<b>Result</b>	Integer/Long	If the result of the function is true - this mean that the “CFD_BGR“ engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

After the successful execution of the method – you can receive the output parameters by using one of the following methods:

- get\_OutputParam\_ByIndex;
- get\_OutputParam\_ByName;
- use the values from last\_AnswerList;

*Note: Any of the commands exported into the "human oriented" command names list can be executed with your own input text via method “execute\_Command”. You can execute each of the commands from the low level protocol of the fiscal device via method “execute\_Command”. During the development of the client application **you have more than one way to execute the commands** if:*

- *something goes wrong;*
- *something is not clear;*
- *the command is not exported for the execution by name;*

## get\_OutputParams\_Count

Through this method – the programmers can receive the count of the output parameters for a given command.

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command " <b>get_CommandsList</b> " or from the document " <b>CommandsList.xls</b> ".
<b>Value</b>	Integer/Long	The count of the output parameters for a given command.
<b>Result</b>	Integer/Long	If the result of the function is true - this mean that the “CFD_BGR“ engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## get\_OutputParams\_Names

Through this method – the programmers can receive the list of the output parameter names for a given command.

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	The one of the "human names" of the given command exported from the method “ <b>get_CommandsList</b> ”.
<b>Value</b>	WideString BSTR	The list of output parameter names accordingly to the used “ <b>command_Name</b> ” value.
<b>Result</b>	Integer/Long	If the result of the function is true - this mean that the “CFD_BGR“ engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## get\_OutputParam\_ByIndex

After the successful execution of the method “**execute\_Command\_ByName**” – you can receive the output parameters by using this method.

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command " <b>get_CommandsList</b> " or from the document " <b>CommandsList.xls</b> ".
<b>param_Index</b>	Integer/Long	A zero based index related to the output parameters list of this command.
<b>param_Value</b>	WideString BSTR	The value of the output parameter.
<b>Result</b>	Integer/Long	If the result of the function is true – this mean that the “CFD_BGR“ engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

## get\_OutputParam\_ByName

After the successful execution of the method “**execute\_Command\_ByName**” – you can receive the output parameters by using this method.

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command " <b>get_CommandsList</b> " or from the document " <b>CommandsList.xls</b> ".
<b>param_Name</b>	WideString BSTR	The output parameter name.
<b>param_Value</b>	WideString BSTR	The value of the output parameter.
<b>Result</b>	Integer/Long	If the result of the function is true – this mean that the “CFD_BGR“ engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command

		execution.
--	--	------------

## generate\_SourceCode

In this version of the “CFD\_BGR“ engine – the programmers can use this method to generate a source code which gives an example of the usage of method “**execute\_Command\_ByName**”. The source code can be used and modified as is comfortable for the programmer and without any license restrictions. In simple terms – it is free as a beer. Of course, this source code is a generated from our AI example so if you choose to use it – please ensure that it is implemented and works well on a real fiscal device.

Currently the “CFD\_BGR“ engine can generate a source code on two languages for all commands which are exported from the method "**get\_CommandsList**".

Parameters:

Name	Type	Description
<b>command_Name</b>	WideString BSTR	Use one of the three “human names” possibilities received by the command " <b>get_CommandsList</b> " or from the document " <b>CommandsList.xls</b> ".
<b>code_Type</b>	TCodeType	Possible parameter values: <ul style="list-style-type: none"><li>• Delphi;</li><li>• CSharp;</li></ul>
<b>Result</b>	Integer/Long	If the result of the function is true – this mean that the “CFD_BGR“ engine internally checks the state of the given status bit and if it is raised in the answer of some command – the engine will return an error as a result of the command execution.

***Note:** In this version of “CFD\_BGR“ – the driver generate source code only on Pascal (Delphi) and C#. If you find a bug it would be nice to send an email to the authors of the COM server. We would be glad to fix or resolve the issue in the next version of the engine.*



## Машинно-независими методи (device-independent methods)

### Отваряне на фискален бон

За да се направи отварянето на фискален бон независимо спрямо изискванията на различните устройства – в интерфейс „CFD\_BGR“ са добавени (*Read/Write*) properties които трябва да бъдат попълнени преди извикването на самия метод.

В следващата таблица са изброени поретата (properties) и техните характеристики:

Име	Пояснение			Задължително поле в логическа група „А“	Задължително поле в логическа група „В“	Задължително поле в логическа група „С“
Operator_Code	Код на оператор.			✓	✓	✓
	Логическа група	Стойност	По подразбиране			
	A	от 1 до 16	1			
	B	от 1 до 30	1			
	C	от 1 до 30	1			
Operator_Password	Операторска парола:			✓	✓	✓
	Логическа група	Стойност	По подразбиране			
	A	4 до 8 цифри	0000			
	B	1 до 8 цифри	1			
	C	1 до 8 цифри	1			
Till_Number	Номер на касово място. /цяло число от 1 до 99999/ Ако клиентския софтуер не зададе стойност, сървъра ще се опита да отвори сторно бон със стойност 1.			✓	✓	✓
FR_UNP	УНП на фискалния документ. (Уникален номер продажба)			✗	✗	✗
FR_Invoice	Определя дали ще се отваря фискален бон или разширена клиентска бележка (фактура).			✗	✗	✗
	Стойност	Значение				
	0	Фискален бон				
	1	Разширена клиентска бележка (фактура).				

Полетата оцветени в зелено са задължителни. Фискалните устройства позволяват да бъде отварян фискален бон без подаване на УНП, заради съвместимост със стар софтуер, но за всеки софтуер деклариран като отговарящ на наредба №18 полето е задължително.

Имайте предвид, че отварянето на фактура е възможно само и единствено, ако във фискалното устройство е зададен допустим интервал от фактури и брояча на фактурите не е достигнал края му.

## ***init\_FiscalReceiptValues***

Метода инициализира всички properties необходими за отварянето на фискален бон (или фактура). Където това е възможно – метода попълва стойности по подразбиране.

- ✓ Поле „*can\_OpenFiscalReceipt*“ става със стойност **True**;
- ✓ Поле „*can\_OpenInvoiceReceipt*“ става със стойност **False**.

Този метод се изпълнява автоматично – веднага след успешно отваряне на фискален бон (или фактура), така че стойностите да са инициализирани за следващата употреба, но клиентското приложение може да го вика и самостоятелно – преди попълването на правилните за случая стойности.

## ***can\_OpenFiscalReceipt***

Това поле е само за четене. Ако „*can\_OpenFiscalReceipt*“ е със стойност **False** – това означава, че не са попълнени необходимите за отваряне на фискален бон стойности или че има отворен документ и до неговото затваряне не може да се отвори нов.

Пример за употреба - клиентското приложение може да отвори фискален бон, ако е попълнило стойности в полета:

- ✓ *Operator\_Code*;
- ✓ *Operator\_Password*;
- ✓ *Till\_Number*;
- ✓ *FR\_UNP*;

Погледнете таблицата с необходими за отваряне на фискален бон полета по-горе. Общото правило е че ако сте попълнили поне едно поле от даден цвят – трябва да попълните и останалите полета от същия цвят, за да стане пропърти „*can\_OpenFiscalReceipt*“ със стойност **True** и съответно метод „*open\_FiscalReceipt*“ да може да бъде изпълнен.

При попълване на горните полета, сървърът проверява дали в съответните полета въобще е попълнено нещо и само в някои случаи може да провери дали стойността е правилна.

Пропърти „*can\_OpenFiscalReceipt*“ със стойност **True** не означава, че отварянето на фискален бон ще премине успешно, а че вече сте попълнили достатъчно на брой полета за опит за отваряне чрез съответната комбинация от задължителни и опционални параметри.

## ***can\_OpenInvoiceReceipt***

Това поле е само за четене. Ако „*can\_OpenInvoiceReceipt*“ е със стойност **False** – това означава, че не са попълнени необходимите за отваряне на фискален бон (тип фактура) стойности или че има отворен документ и до неговото затваряне не може да се отвори нов.

Пример за употреба - клиентското приложение може да отвори фискален бон, ако е

попълнило стойности в полета:

- ✓ Operator\_Code;
- ✓ Operator\_Password;
- ✓ Till\_Number;
- ✓ FR\_UNP;
- ✓ FR\_Invoice (със стойност 1)

При попълване на горните полета, сървъра проверява дали в съответните полета въобще е попълнено нещо и само в някои случаи може да провери дали стойността е правилна.

Пропърти „*can\_OpenInvoiceReceipt*“ със стойност *True* не означава, че отварянето на фискален бон (тип фактура) ще премине успешно, а че вече сте попълнили достатъчно на брой полета за опит за отваряне чрез съответната комбинация от задължителни и опционални параметри. Честа срещана грешка – всички полета са попълнени правилно, но не е зададен интервал от номера за фактура (команда 66) или той е изчерпан.

## ***open\_FiscalReceipt* u *open\_InvoiceReceipt***

Ако „*can\_OpenFiscalReceipt*“ е със стойност *True* и стойностите са правилни – COM сървъра ще може да изпълни метод „*open\_FiscalReceipt*“. Ако „*can\_OpenInvoiceReceipt*“ е със стойност *True* и стойностите са правилни – COM сървъра ще може да изпълни метод „*open\_InvoiceReceipt*“.

И в двата случая – отварянето на фискален бон все още зависи от фискалното устройство. Например ако в момента няма хартия – метода ще върне статус бит грешка, а съответния статус бит ще бъде вдигнат след отговора от страна на устройството.

След успешно отваряне на фискален бон – всички *properties* имащи отношение към метода се инициализират вътрешно. Не трябва да забравяте, че трябва да попълвате правилните стойности преди изпълнението на методите.

За логически групи „А“ и „Б“ - метода попълва (read only) *properties*:

- ✓ *AllReceipt\_Count* - Броят на всички издадени бонове (фискални и служебни) от последното приключване на деня до момента;
- ✓ *FiscalReceipt\_Count* - Броят на всички издадени фискални бонове от последното приключване на деня до момента;

За логическа група „С“ - метода попълва стойността на поле **GlobalCounter**;

При успешно изпълнение и отваряне на фискален бон – резултата от изпълнението на метода е със стойност **0**.

При неуспешно изпълнение – резултата съвпада по стойност със стойността на *property* „*lastError\_Code*“.

При отваряне на фискална бележка от тип фактура трябва да се знае, че ако поради някаква причина се наложи да я анулирате (команда 60), при следващото отваряне на фактура устройствата от тип „А“ ще използват същия номер. При устройства от логическа група „Б“ и „С“ отварянето на фактура изхабява текущия номер и при следващо отваряне ще се използва

следващия номер от интервала.

По-долу се вижда какъв документ се отваря при съответната комбинация от въведени полета, както и коя от device-dependent командите се изпълнява вътрешно от страна на COM сървъра.

ID	Operator_Code	Operator_Password	Till_Number	FR_UNP	FR_Invoice	Име на команда	can_OpenFiscalReceipt	can_OpenInvoiceReceipt	Тип документ
A01	✓	✓	✓	✗	0	048_receipt_FiscalOpen_A01	✓	✗	Касов бон
A02	✓	✓	✓	✗	1	048_receipt_FiscalOpen_A02	✗	✓	Фактура
A03	✓	✓	✓	✓	0	048_receipt_FiscalOpen_A03	✓	✗	Касов бон с УНП
A04	✓	✓	✓	✓	1	048_receipt_FiscalOpen_A04	✗	✓	Фактура с УНП
B01	✓	✓	✓	✗	0	048_receipt_FiscalOpen_B01	✓	✗	Касов бон
B02	✓	✓	✓	✗	1	048_receipt_FiscalOpen_B02	✗	✓	Фактура
B03	✓	✓	✓	✓	0	048_receipt_FiscalOpen_B03	✓	✗	Касов бон с УНП
B04	✓	✓	✓	✓	1	048_receipt_FiscalOpen_B04	✗	✓	Фактура с УНП
C01	✓	✓	✓	✗	0	048_receipt_FiscalOpen_C01	✓	✗	Касов бон
C02	✓	✓	✓	✗		048_receipt_FiscalOpen_C02	✗	✓	Фактура
C03	✓	✓	✓	✓		048_receipt_FiscalOpen_C03	✓	✗	Касов бон с УНП
C04	✓	✓	✓	✓		048_receipt_FiscalOpen_C04	✗	✓	Фактура с УНП

**Софтуерните продукти съобразени с промените в наредба №18 трябва да издават фискални документи с УНП.** Останалите методи са оставени за съвместимост или при употреба на фискални устройства от логическа група „А“. В случая на фискални устройства от логическа група „А“ имаме характерна особеност. Ако поне веднъж е издаден фискален документ с УНП – фискалното устройство го запомня и ако използвате команда за отваряне на фискална бележка без да сте попълнили УНП – фискалното устройство ще генерира само следващ УНП номер. За генериране на този УНП, фискалното устройство използва собствения си сериен номер, номера на оператора от поле „**Operator\_Code**“ и стойността на брояча от УНП, увеличена с единица.

## Отваряне на сторно бон

Както бе обяснено по-горе, фискалните устройства, предлагани от Датекс могат да се разделят на три логически групи. За да се направи отварянето на сторно бон независимо спрямо логиката и изискванията на различните устройства – в интерфейс „CFD\_BGR“ са добавени (*Read/Write*) *properties* които трябва да бъдат попълнени преди извикването на самия метод.

В следващата таблица са изброени наличните *properties* и техните характеристики:

Име	Пояснение			Задължително поле в логическа група „А“	Задължително поле в логическа група „В“	Задължително поле в логическа група „С“
Operator_Code	Код на оператор.			✓	✓	✓
	Логическа група	Стойност	По подразбиране			
	A	от 1 до 16	1			
	B	от 1 до 30	1			
C	от 1 до 30	1				
Operator_Password	Операторска парола:			✓	✓	✓
	Логическа група	Стойност	По подразбиране			
	A	4 до 8 цифри	0000			
	B	1 до 8 цифри	1			
C	1 до 8 цифри	1				
Till_Number	Номер на касово място. /цяло число от 1 до 99999/ Ако клиентския софтуер не зададе стойност, сървърът ще се опита да отвори сторно бон със стойност 1.			✓	✓	✓
St_Reason_Type	Сървърът очаква цяло число със стойност от 0 до 2.			✓	✓	✓
	Стойност	Значение				
	0	Операторска грешка (по подразбиране)				
	1	Връщане / Замяна / Рекламация				
2	Намаление на данъчната основа					
St_Doc_Number	Номер на документа (глобален) който се сторнира. (цяло положително число от 1 до 9999999)			✓	✓	✓
St_Doc_DateTime	Дата и час на сторнирания документ.			✗	✓	✓
	Логическа група	Стойност до...				
	A	„DDMMYYhhmmss“				
	B	„DDMMYYhhmmss“				
C	„DD-MM-YY hh:mm:ss DST“					
St_FM_Number	Номер на фискалната памет на ФУ от което е издаден бона, който се сторнира.			✗	✓	✓
St_Doc_UNP	УНП на документа, който се сторнира.			✗	✗	✗
St_ByInvoice				✗	✗	✗
	Стойност	Значение				
	0	Сторниране по фискален бон (по подразбиране)				
1	Сторниране по фактура					
St_InvoiceNumber	Номер на фактурата която се сторнира. (1 - 9999999999)			✗	✗	✗
St_Reason				✗	✗	✗
	Логическа група	Стойност до...				
	A	Причина за сторниране до 30 символа				
	B	Причина за сторниране до 42 символа				
C	Причина за сторниране до 42 символа					
St_Current_UNP	УНП на самия сторно бон. (Само в логическа група А)			✗	-	-

Полетата, оцветени в зелено са минимално изискуемите за съответната логическа група устройства, но имайте предвид, че употребата само на минимално изискуемите properties има своите особености:

- Ако в логическа група „А“ попълните само минимума данни - ФУ ще търси издадения бон в собствената си контролна лента. Ако не го намери, командата приключва неуспешно. Ако го намери, данните се попълват автоматично и се издава сторно бон с всички присъстващи в оригиналния фискален бон данни;
- Ако в логически групи „В“ и „С“ попълните само минимума данни – това означава, че заявявате, че се сторнира документ, за който няма издаден УНП номер;
- Според наредба №18 – сторниране през фискално устройство е възможно до седмо число на следващия месец. Ако подадете дата която не е съобразена с тази особеност, то фискалното устройство ще откаже отварянето на сторно бон;

## ***init\_StornoValues***

Метода инициализира всички properties необходими за отварянето на сторно бон. Където това е възможно – метода попълва стойности по подразбиране.

Пропърти „***can\_OpenStornoReceipt***“ става със стойност False.

Този метод се изпълнява автоматично – веднага след успешно отваряне на сторно бон. Така че стойностите да са инициализирани за следващата употреба, но клиентското приложение може да го вика и самостоятелно – преди попълването на правилните за случая стойности.

## ***can\_OpenStornoReceipt***

Това поле е само за четене. „***can\_OpenStornoReceipt***“ става със стойност **False** след инициализиране с метод „***init\_StornoValues***“ и остава с тази стойност до момента в който клиентското приложение попълни всички необходими за отварянето на сторно бон стойности.

Пример за употреба:

Ако клиентското приложение е попълнило стойности в полета:

- ✓ Operator\_Code;
- ✓ Operator\_Password;
- ✓ Till\_Number;
- ✓ St\_Reason\_Type;
- ✓ St\_Doc\_Number;

Ако фискалното устройство е модел в логическа група „А“ - стойността на „***can\_OpenStornoReceipt***“ ще стане **True**, тъй като това са минимално изискуемите полета за

отваряне на сторно бон в тази група.

Ако фискалното устройство е модел в логическа група „В“ или „С“ - стойността на „*can\_OpenStornoReceipt*“ ще остане в стойност **False**, тъй като в тези групи се изисква попълването на минимум още две полета.

Погледнете таблицата с необходими за отваряне на сторно бон полета по-горе. Общото правило е че ако сте попълнили поне едно поле от даден цвят – трябва да попълните и останалите полета от същия цвят, за да стане пропърти „*can\_OpenStornoReceipt*“ със стойност **True** за устройството към което сте свързани в момента и съответно метод „*open\_StornoReceipt*“ да може да бъде изпълнен.

Това означава, че в момента в който попълните минимално изискуемите полета стойността ще стане **True**, а при попълването на следващото поле е възможно стойността да стане отново **False** до попълването на съответния набор от стойности.

При попълване на горните полета, сървъра проверява дали в съответните полета въобще е попълнено нещо и само в някои случаи може да провери дали стойността е правилна.

Пропърти „*can\_OpenStornoReceipt*“ със стойност **True** не означава, че отварянето на сторно бон ще премине успешно, а че вече сте попълнили достатъчно на брой полета за опит за отваряне чрез съответната комбинация от задължителни и опционални параметри.

## ***open\_StornoReceipt***

Поведението на COM сървър в зависимост от въведените данни по отношение на сторно бона, може да се види по-долу.

Ако „***can\_OpenStornoReceipt***“ е със стойност ***True*** и стойностите са правилни – COM сървър ще изпълни метода, но отварянето на сторно бон все още зависи от фискалното устройство.

Ако клиентското приложение е посочило като причина нещо различно от операторска грешка, фискалното устройство ще провери дали в чекмеджето има достатъчно наличност (логически) и ако няма – ще откаже операцията.

Ако фискалното устройство е в група А и попълнените стойности са минимално изискуемите (до „***St\_Doc\_Number***“ вкл.) - фискалното устройство ще търси издадения бон в контролната си лента. Ако не го намери, командата приключва неуспешно. Ако го намери, данните се попълват автоматично и се издава сторно бон с всички присъстващи в оригиналния фискален бон данни.

След успешно отваряне на сторно бон – всички ***properties*** имащи отношение към метода се инициализират вътрешно с цел предпазване от последващо сторниране с неверни данни.

Метода попълва (read only) ***properties***:

- ✓ ***AllReceipt\_Count*** - Броят на всички издадени бонове (фискални и служебни) от последното приключване на деня до момента;
- ✓ ***StReceipt\_Count*** - Броят на всички издадени фискални СТОРНО бонове от последното приключване на деня до момента;

При успешно изпълнение и отваряне на сторно бон – резултата от изпълнението на метода е със стойност **0**.

При неуспешно изпълнение – резултата съвпада по стойност със стойността на ***property*** „***lastError\_Code***“.

„***St\_Reason***“ е текстово ***property*** в което можете да посочите причината за отварянето на сторно бона. Това поле е независим, опционален параметър в група устройства А. В тази група може да бъде добавен независимо към всяка валидна комбинация от стойности, така че в зависимост от това дали полето е попълнено или не е възможно COM сървър да изпрати една или друга комбинация от входни параметри към фискалното устройство.

Само в логическа група "А" можете да посочите УНП на самия сторно бон като го попълните в текстово ***property*** „***St\_Current\_UNP***“. Това поле също е независим, опционален параметър и може да бъде добавен независимо към всяка валидна комбинация от стойности. В зависимост от това дали полето е попълнено или не – възможно е COM сървър да изпрати една или друга комбинация от входни параметри към фискалното устройство.

При опит за отваряне на сторно бон – метода проверява вътрешно наличността и валидността на стойностите по входните полета. Ако има недостатъчен брой попълнени полета или някоя от стойностите не отговаря на ограниченията – метода ще върне като резултат: „**-43144**“ (Not enough input parameters).

Стойността на ***property*** „***can\_OpenStornoReceipt***“ ще стане ***True*** в комбинациите от правилно попълнени стойности на входните данни, описани по-долу.

Затварянето на сторно бон става само след плащане на **точна сума в брой**.



## Логическа група „А“

ID	Operator_Code	Operator_Password	Till_Number	St_ByInvoice	St_InvoiceNumber	St_Current_UNP	St_Reason_Type	St_Doc_Number	St_Doc_UNP	St_Doc_DateTime	St_FM_Number	St_Reason	Тип сторниран документ
A01	✓	✓	✓				✓	✓					Касов бон
A02	✓	✓	✓			✓	✓	✓					
A03	✓	✓	✓				✓	✓				✓	
A04	✓	✓	✓			✓	✓	✓				✓	
A05	✓	✓	✓				✓	✓	✓	✓	✓		
A06	✓	✓	✓			✓	✓	✓	✓	✓	✓		
A07	✓	✓	✓				✓	✓	✓	✓	✓	✓	
A08	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	
A09	✓	✓	✓	✓	✓		✓	✓					Фактура
A10	✓	✓	✓	✓	✓	✓	✓	✓					
A11	✓	✓	✓	✓	✓		✓	✓				✓	
A12	✓	✓	✓	✓	✓	✓	✓	✓				✓	
A13	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓		
A14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
A15	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	
A16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

### Забележки:

- Колоната „ID“ е добавена към таблицата с цел по-добро описание, а също така в случай, че се съобщава за грешка, проблем или просто при въпрос към поддържащия екип;
- Ако не желаете да използвате метод „open\_StornoReceipt“, а желаете да използвате „execute\_Command\_ByName“, то колоната „ID“ може да бъде свързана с имената на методите за сторниране в метод „execute\_Command\_ByName“ по следния примерен начин – на „ID = A08“ съответстват на имена на методи:
  - 046\_receipt\_StornoOpen\_A08
  - receipt\_StornoOpen\_A08
  - StornoOpen\_A08
- Ако се извършива сторно по фактура – отново е необходимо между командите за тотал и затваряне на бележката да се подаде команда за отпечатване на информация за крайния клиент;

## Логическа група „В“

ID	Operator_Code	Operator_Password	St_Doc_UNP	Till_Number	St_Reason_Type	St_Doc_Number	St_Doc_DateTime	St_FM_Number	St_ByInvoice	St_InvoiceNumber	St_Reason	Тип сторниран документ
B_01	✓	✓		✓	✓	✓	✓	✓				Касов бон
B_02	✓	✓	✓	✓	✓	✓	✓	✓				
B_03	✓	✓		✓	✓	✓	✓	✓	✓	✓		Фактура
B_04	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	
B_05	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
B_06	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

### Забележки:

- Колона „ID“ е добавена към таблицата с цел по-добро описание, а също така в случай, че се съобщава за грешка, проблем или просто при въпрос към поддържащия екип;
- В тази логическа група устройства – сторно бона няма собствен УНП номер;
- Ако не желаете да използвате метод „open\_StornoReceipt“, а желаете да използвате „execute\_Command\_ByName“, то колона „ID“ може да бъде свързана с имената на методите за сторниране в метод „execute\_Command\_ByName“ по следния примерен начин – на „ID = B01“ съответстват на имена на методи:
  - 046\_receipt\_StornoOpen\_B01
  - receipt\_StornoOpen\_B01
  - StornoOpen\_B01
- Ако се извършва сторно по фактура – отново е необходимо между командите за тотал и затваряне на бележката да се подаде команда за отпечатване на информация за крайния клиент;

## Логическа група “С”

ID	Operator_Code	Operator_Password	Till_Number	St_ByInvoice	St_InvoiceNumber	St_Current_UNP	St_Reason_Type	St_Doc_Number	St_Doc_UNP	St_Doc_DateTime	St_FM_Number	St_Reason
C_01	✓	✓	✓				✓	✓		✓	✓	
C_02	✓	✓	✓				✓	✓	✓	✓	✓	
C_03	✓	✓	✓	✓	✓		✓	✓		✓	✓	
C_04	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓
C_05	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	
C_06	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓

### Забележки:

- Колоната „ID“ е добавена към таблицата с цел по-добро описание, а също така в случай, че се съобщава за грешка, проблем или просто при въпрос към поддържащия екип;
- В тази логическа група устройства – сторно бона няма собствен УНП номер;
- Ако се извършва сторно по фактура – отново е необходимо между командите за тотал и затваряне на бележката да се подаде команда за отпечатване на информация за крайния клиент

## Машинно-зависими методи (device-dependent methods)

Интерфейс “CFD\_BGR” предоставя повечето от методите, зависими от съответния модел. Пропуснати са някои методи които се използват при производството, за целите на сервизирането или които не представляват интерес от гледна точка на ежедневната практика.

Най-удобния начин за употреба на машинно-зависим метод е чрез изпълнението на методи:

- ✓ „[set\\_InputParam\\_ByName](#)“;
- ✓ „[execute\\_Command\\_ByName](#)“;
- ✓ „[get\\_OutputParam\\_ByName](#)“

Имената на съответните методи могат да се получат освен от документацията и от самия COM сървър чрез методи:

- ✓ „[get\\_CommandsList](#)“;
- ✓ „[get\\_CommandInfo](#)“;
- ✓ „[get\\_InputParams\\_Count](#)“;
- ✓ „[get\\_InputParams\\_Names](#)“

Всеки от предоставените методи, чиято реализация не е удобна или не работи както трябва може да бъде „заобиколен“ чрез реализация на собствена имплементация използвайки метод „[custom\\_Command](#)“.

Като цяло – интерфейса предоставя три начина за извикване по име:

- По пълно име на командата, включващо номера на командата, името на групата и името на самия метод (Пример: „*100\_display\_Show\_Text*“);
- По име на група и име на метод (Пример: „*display\_Show\_Text*“);
- По име на метод (Пример: „*Show\_Text*“);

За да се избегне повторение или двузначност – при някои методи имената на втория и третия вариант съвпадат. Пример: „*033\_display\_Clear*“, „*display\_Clear*“ и „*display\_Clear*“. За по-голяма яснота от човешка гледна точка машинно-зависимите методи са описани по-долу не по номер на команда, а по групи.

## Машинно-зависими команди в логическа група „А“

### Външен дисплей

Команди към устройства в група „А“	
✓ 033_display_Clear	Изчистване на дисплея
✓ 035_display_Show_LowerLine	Показване на текст (долен ред).
✓ 047_display_Show_UpperLine	Показване на текст (горен ред).
✓ 063_display_Show_DateTime	Показване на датата и часа.
✓ 100_display_Show_Text	Дисплей - пълен контрол

### Нефискален бон

Команди към устройства в група „А“	
✓ 038_receipt_NonFiscal_Open	Отваряне на нефискален бон
✓ 042_receipt_NonFiscal_Text	Печат на нефискален текст
✓ 042_receipt_PNonFiscal_Text	Печат на параметризиран нефискален текст
✓ 039_receipt_NonFiscal_Close	Затваряне на нефискален бон

### Нефискален бон, завъртян на 90 градуса

Команди към устройства в група „А“	
✓ 122_receipt_NonFiscalRotated_Open	Отваряне на завъртян нефискален бон
✓ 123_receipt_NonFiscalRotated_Text	Печат на завъртян нефискален текст
✓ 124_receipt_NonFiscalRotated_Close	Затваряне на нефискален бон отворен с команда 122

### Сторниране

Команди към устройства в група „А“	
<a href="#">(вж. Таблицата)</a>	
✓ 046_receipt_StornoOpen_A01	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_A02	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_A03	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_A04	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_A05	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_A06	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_A07	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_A08	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_A09	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_A10	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_A11	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_A12	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_A13	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_A14	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_A15	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_A16	Вариант на командата за отваряне на сторно бон (по фактура)

## Отваряне на фискален бон и фактура

Команди към устройства в група „А“		
(вж. Таблици)		
✓ 048_receipt_FiscalOpen_A01	Вариант на командата за отваряне на фискален бон	
✓ 048_receipt_FiscalOpen_A02	Вариант на командата за отваряне на фискален бон тип фактура	
✓ 048_receipt_FiscalOpen_A03	Вариант на командата за отваряне на фискален бон с УНП	
✓ 048_receipt_FiscalOpen_A04	Вариант на командата за отваряне на фискален бон тип фактура с УНП	
✓ 048_receipt_Fiscal_01_Open	Стар вариант на команда 048_receipt_FiscalOpen_A01	
✓ 048_receipt_Fiscal_02_Open	Стар вариант на команда 048_receipt_FiscalOpen_A03	
✓ 048_receipt_Invoice_Open	Стар вариант на команда 048_receipt_FiscalOpen_A04	
✓ 048_receipt_Invoice_01_Open	Стар вариант на команда 048_receipt_FiscalOpen_A02	
✓ 056_receipt_Fiscal_Close	Затваряне на фискален бон или фактура	

## Печат на данни за клиента преди команда за затваряне на фактура

Команди към устройства в група „А“		
✓ 057_receipt_PrintClientInfo_01	Вариант на команда за печат на данни за клиента	
✓ 057_receipt_PrintClientInfo_02	Вариант на команда за печат на данни за клиента	
✓ 057_receipt_PrintClientInfo_03	Вариант на команда за печат на данни за клиента	
✓ 057_receipt_PrintClientInfo_04	Вариант на команда за печат на данни за клиента	
✓ 057_receipt_PrintClientInfo_05	Вариант на команда за печат на данни за клиента	
✓ 057_receipt_PrintClientInfo_06	Вариант на команда за печат на данни за клиента	
✓ 057_receipt_PrintClientInfo_07	Вариант на команда за печат на данни за клиента	

## Регистриране на продажба

Команди към устройства в група „А“		
✓ 049_receipt_Sale	Продажба	
✓ 049_receipt_Sale_TextRow1	Продажба с текст в първи ред	
✓ 049_receipt_Sale_TextRow2	Продажба с текст във втори ред	
✓ 049_receipt_Sale_Minimum	Продажба - минимум параметри	
✓ 049_receipt_Sale_Un	Продажба с употреба на мерна единица	
✓ 049_receipt_Sale_TextRow1Un	Продажба с употреба на мерна единица с текст в първи ред	
✓ 049_receipt_Sale_TextRow2Un	Продажба с употреба на мерна единица с текст във втори ред	
✓ 049_receipt_Sale_UnWText	Продажба с употреба на мерна единица – без текст	

## Регистриране на продажба с корекция

Команди към устройства в група „А“		
✓ 049_receipt_Sale_CByPercent	Продажба с корекция по процент	
✓ 049_receipt_Sale_TextRow1CByPercent	Продажба с корекция по процент и текст в първи ред	
✓ 049_receipt_Sale_TextRow2CByPercent	Продажба с корекция по процент и текст във втори ред	
✓ 049_receipt_Sale_CByPercentWText	Продажба с корекция по процент – без текст	
✓ 049_receipt_Sale_UnCByPercent	Продажба с употреба на мерна единица и корекция по процент	
✓ 049_receipt_Sale_TextRow1UnCByPercent	С употреба на мерна единица и корекция по процент (текст в първи ред)	
✓ 049_receipt_Sale_TextRow2UnCByPercent	С употреба на мерна единица и корекция по процент (текст във втори ред)	
✓ 049_receipt_Sale_UnCByPercentWText	Продажба с корекция по процент и мерна единица – без текст	
✓ 049_receipt_Sale_CBySum	Продажба с корекция чрез твърда сума	
✓ 049_receipt_Sale_TextRow1CBySum	Продажба с корекция чрез твърда сума (текст в първи ред)	
✓ 049_receipt_Sale_TextRow2CBySum	Продажба с корекция чрез твърда сума (текст във втори ред)	
✓ 049_receipt_Sale_CBySumWText	Продажба с корекция чрез твърда сума – без текст	
✓ 049_receipt_Sale_UnCBySum	С употреба на мерна единица и корекция чрез твърда сума	
✓ 049_receipt_Sale_TextRow1UnCBySum	С употреба на мерна единица и корекция чрез твърда сума (текст в първи ред)	
✓ 049_receipt_Sale_TextRow2UnCBySum	С употреба на мерна единица и корекция чрез твърда сума (текст във втори ред)	
✓ 049_receipt_Sale_UnCBySumWText	С употреба на мерна единица и корекция чрез твърда сума – без текст	

## Регистриране на продажба в департамент

Команди към устройства в група „А“		
✓	049_receipt_DSale	Продажба в департамент
✓	049_receipt_DSale_TextRow1	Продажба в департамент(текст в първи ред)
✓	049_receipt_DSale_TextRow2	Продажба в департамент(текст във втори ред)
✓	049_receipt_DSale_Minimum	Продажба в департамент - минимум параметри
✓	049_receipt_DSale_Un	Продажба в департамент с употреба на мерна единица
✓	049_receipt_DSale_TextRow1Un	Продажба в департамент с употреба на мерна единица(текст в първи ред)
✓	049_receipt_DSale_TextRow2Un	Продажба в департамент с употреба на мерна единица(текст във втори ред)
✓	049_receipt_DSale_UnWText	Продажба в департамент с мерна единица – без текст

## Регистриране на продажба в департамент с корекция

Команди към устройства в група „А“		
✓	049_receipt_DSale_CByPercent	Продажба в департамент с корекция по %
✓	049_receipt_DSale_TextRow1CByPercent	Продажба в департамент с корекция по %, (текст в първи ред)
✓	049_receipt_DSale_TextRow2CByPercent	Продажба в департамент с корекция по %, (текст във втори ред)
✓	049_receipt_DSale_CByPercentWText	Продажба в департамент с корекция по процент – без текст
✓	049_receipt_DSale_UnCByPercent	Продажба в департамент с корекция по %, употреба на мерна единица
✓	049_receipt_DSale_TextRow1UnCByPercent	Продажба в департамент с корекция по %, употреба на мерна единица(текст в първи ред)
✓	049_receipt_DSale_TextRow2UnCByPercent	Продажба в департамент с корекция по %, употреба на мерна единица(текст във втори ред)
✓	049_receipt_DSale_CBySumWText	Продажба в департамент с корекция чрез твърда сума – без текст
✓	049_receipt_DSale_CBySum	Продажба в департамент с корекция чрез сума
✓	049_receipt_DSale_TextRow1CBySum	Продажба в департамент с корекция чрез сума(текст в първи ред)
✓	049_receipt_DSale_TextRow2CBySum	Продажба в департамент с корекция чрез сума(текст във втори ред)
✓	049_receipt_DSale_CBySumWText	Продажба в департамент с корекция чрез твърда сума – без текст
✓	049_receipt_DSale_UnCBySum	Продажба в департамент с корекция чрез сума, употреба на мерна единица
✓	049_receipt_DSale_TextRow1UnCBySum	Продажба в департамент с корекция чрез сума, употреба на мерна единица(текст в първи ред)
✓	049_receipt_DSale_TextRow2UnCBySum	Продажба в департамент с корекция чрез сума, употреба на мерна единица(текст във втори ред)
✓	049_receipt_DSale_UnCBySumWText	Продажба в департамент с корекция чрез сума, употреба на мерна единица – без текст

## Регистриране на продажба и автоматично показване на външен дисплей

Команди към устройства в група „А“		
✓	052_receipt_DisplaySale	Продажба с авт.показване на външен дисплей
✓	052_receipt_DisplaySale_Un	Продажба с авт.показване на външен дисплей, употреба на мерна единица
✓	052_receipt_DisplaySale_CByPercent	Продажба с авт.показване на външен дисплей и корекция по процент
✓	052_receipt_DisplaySale_UnCByPercent	Продажба с авт.показване на външен дисплей, корекция по процент и употреба на мерна единица
✓	052_receipt_DisplaySale_CBySum	Продажба с авт.показване на външен дисплей и корекция със сума
✓	052_receipt_DisplaySale_UnCBySum	Продажба с авт.показване на външен дисплей, корекция със сума и употреба на мерна единица
✓	052_receipt_DisplaySale_Minimum	Продажба с авт.показване на външен дисплей - минимум параметри

## Регистриране на продажба в департамент и автоматично показване на външен дисплей

Команди към устройства в група „А“		
✓	052_receipt_DisplayDSale	Продажба с авт.показване на външен дисплей
✓	052_receipt_DisplayDSale_Un	Продажба с авт.показване на външен дисплей, употреба на мерна единица
✓	052_receipt_DisplayDSale_CByPercent	Продажба с авт.показване на външен дисплей и корекция по процент
✓	052_receipt_DisplayDSale_UnCByPercent	Продажба с авт.показване на външен дисплей, корекция по процент и употреба на мерна единица
✓	052_receipt_DisplayDSale_CBySum	Продажба с авт.показване на външен дисплей и корекция със сума
✓	052_receipt_DisplayDSale_UnCBySum	Продажба с авт.показване на външен дисплей, корекция със сума и употреба на мерна единица
✓	052_receipt_DisplayDSale_Minimum	Продажба с авт.показване на външен дисплей - минимум параметри

## Регистриране на продажба на артикул

Команди към устройства в група „А“		
✓	058_receipt_PLU_Sale	Продажба на артикул
✓	058_receipt_PLUSale_CByPercent	Продажба на артикул и корекция по процент
✓	058_receipt_PLUSale_CBySum	Продажба на артикул и корекция със сума
✓	058_receipt_PLUSale_Un	Продажба на артикул и употреба на мерна единица
✓	058_receipt_PLUSale_UnCByPercent	Продажба на артикул, корекция по процент и употреба на мерна единица
✓	058_receipt_PLUSale_UnCBySum	Продажба на артикул, корекция със сума и употреба на мерна единица

## Регистриране на продажба на артикул в департамент

Команди към устройства в група „А“		
✓	058_receipt_PLUDep_Sale	Продажба на артикул в департамент
✓	058_receipt_PLUDep_Sale_CByPercent	Продажба на артикул в департамент и корекция по процент
✓	058_receipt_PLUDep_Sale_CBySum	Продажба на артикул в департамент и корекция със сума
✓	058_receipt_PLUDep_Sale_Un	Продажба на артикул в департамент и употреба на мерна единица
✓	058_receipt_PLUDep_Sale_UnCByPercent	Продажба на артикул в департамент, корекция по процент и употреба на мерна единица
✓	058_receipt_PLUDep_Sale_UnCBySum	Продажба на артикул в департамент, корекция със сума и употреба на мерна единица

## Регистриране на продажба на артикул и автоматично показване на външен дисплей

Команди към устройства в група „А“		
✓	058_receipt_DisplayPLUSale	С авт.показване на дисплей
✓	058_receipt_DisplayPLUSale_CByPercent	С авт.показване на дисплей и корекция по процент
✓	058_receipt_DisplayPLUSale_CBySum	С авт.показване на дисплей и корекция със сума
✓	058_receipt_DisplayPLUSale_Un	С авт.показване на дисплей и употреба на мерна единица
✓	058_receipt_DisplayPLUSale_UnCByPercent	С авт.показване на дисплей, корекция с % и употреба на мерна единица
✓	058_receipt_DisplayPLUSale_UnCBySum	С авт.показване на дисплей, корекция със сума и употреба на мерна единица

## Регистриране на продажба на артикул в департамент и авт. показване на външен дисплей

Команди към устройства в група „А“		
✓	058_receipt_DisplayDepPLU_Sale	С авт.показване на дисплей
✓	058_receipt_DisplayDepPLU_Sale_CByPercent	С авт.показване на дисплей и корекция по процент
✓	058_receipt_DisplayDepPLU_Sale_CBySum	С авт.показване на дисплей и корекция със сума
✓	058_receipt_DisplayDepPLU_SaleUn	С авт.показване на дисплей и употреба на мерна единица
✓	058_receipt_DisplayDepPLU_SaleUn_CByPercent	С авт.показване на дисплей, корекция по процент и употреба на мерна единица
✓	058_receipt_DisplayDepPLU_SaleUn_CBySum	С авт.показване на дисплей, корекция със сума и употреба на мерна единица



## Субтотал и Тотал

Команди към устройства в група „А“		
✓	051_receipt_Subtotal	Команда за субтотал
✓	051_receipt_Subtotal_CByPercent	Команда за субтотал с корекция по процент
✓	051_receipt_Subtotal_CBySum	Команда за субтотал с корекция със сума
✓	053_receipt_Total_PAmount	Команда за тотал с тип на плащане (два реда текст)
✓	053_receipt_Total_PAmountTextRow1	Команда за тотал с тип на плащане (текст в първи ред)
✓	053_receipt_Total_PAmountTextRow2	Команда за тотал с тип на плащане (текст във втори ред)
✓	053_receipt_Total_PAmountWithoutText	Команда за тотал с тип на плащане (без текст)
✓	053_receipt_Total	Команда за тотал с тип на плащане в брой (два реда текст)
✓	053_receipt_Total_TextRow1	Команда за тотал с тип на плащане в брой (текст в първи ред)
✓	053_receipt_Total_TextRow2	Команда за тотал с тип на плащане в брой (текст във втори ред)
✓	053_receipt_Total_WithoutText	Команда за тотал с тип на плащане в брой (без текст)

## Други команди свързани с издаването на фискални и нефискални документи

Команди към устройства в група „А“		
✓	044_receipt_Paper_Feed	Придвижване на хартията
✓	045_receipt_Paper_Cut	Отрязване на хартията
✓	054_receipt_Fiscal_Text	Отпечатване на фискален текст
✓	054_receipt_PFiscalText_01	Отпечатване на параметризиран фискален текст
✓	054_receipt_PFiscalText_02	Отпечатване на параметризиран фискален текст
✓	060_receipt_Fiscal_Cancel	Анулиране на фискален бон
✓	070_receipt_CashIn_CashOut	Служебно въвеждане/извеждане на суми
✓	084_receipt_Print_Barcode_01	Отпечатване на баркод
✓	084_receipt_Print_Barcode_02	Отпечатване на баркод
✓	092_receipt_Separating_Line	Отпечатване на разделителна линия
✓	092_receipt_Separating_LineW	Отпечатване на разделителна линия
✓	103_receipt_Current_Info	Информация за текущия бон
✓	106_receipt_Drawer_KickOut	Отваряне на чекмедже
✓	109_receipt_Print_Duplicate	Отпечатване на копие на последния затворен бон с продажби

## Отчети

Команди към устройства в група „А“		
✓	034_report_Service_Contracts	Отчет на регистрираните сервизни договори
✓	069_report_DailyClosure_01	Дневен отчет със или без нулиране
✓	069_report_DailyClosure_02	Дневен отчет със или без нулиране и забрана на изчистването на натрупаните данни по оператори
✓	073_report_FMByNumRange	Отпечатване на пълен отчет на фискалната памет от даден номер до друг
✓	073_report_FMByNumRange_SHA1	Отпечатване на пълен отчет на фискалната памет от даден номер до друг и контролна сума SHA1
✓	079_report_FMByDateRange_Short	Отпечатване на съкратен отчет на фискалната памет от дата до дата
✓	094_report_FMByDateRange	Пълен отчет на фискалната памет за периода между две дати
✓	094_report_FMByDateRange_SHA1	Пълен отчет на фискалната памет за периода между две дати и контролна сума SHA1
✓	095_report_FMByNumRange_Short	Отпечатване на съкратен отчет на фискалната памет от номер до номер
✓	105_report_Operators	Отпечатване на информация за продажбите по оператори
✓	108_report_ExtDailyClosure_01	Отпечатване на дневен отчет със или без нулиране плюс артикули по които има продажби
✓	108_report_ExtDailyClosure_02	Еквивалент на 069_report_DailyClosure_02 плюс артикули (1..10) по които има продажби
✓	111_report_Items	Отчет по артикули
✓	111_report_Items_InRange	Отчет по артикули от номер до номер
✓	111_report_Items_InRangeByGroup	Отчет по артикули от номер до номер и в група
✓	117_report_DailyClosureByDepartments_01	Дневен отчет със или без нулиране и печат на департаменти с продажби за деня
✓	117_report_DailyClosureByDepartments_02	Екв. на 069_report_DailyClosure_02 плюс печат на департаменти с продажби
✓	118_report_DailyClosureByDepartmentsAndItems_01	Екв. на 069_report_DailyClosure_01, печат на департ. с продажби и артикули (1..40)
✓	118_report_DailyClosureByDepartmentsAndItems_02	Екв. на 069_report_DailyClosure_02 плюс печат на департ. с продажби и артикули(1..40)

## Работа с артикули

Команди към устройства в група „А“		
✓	107_items_Get_ItemsInformation	Получаване на обща информация за възможностите на устройството по отн. артикули
✓	107_items_Set_Item	Програмиране на артикул
✓	107_items_Set_ItemWithReplace	Програмиране на артикул с промяна на количеството
✓	107_items_Set_ItemQuantity	Промяна на наличното количество за артикул
✓	107_items_Delete_Item	Изтриване на артикул, ако няма натрупани суми
✓	107_items_Delete_ItemsInRange	Изтриване на артикули в интервал, ако няма натрупани суми за тях
✓	107_items_Delete_All_Items	Изтриване на всички артикули
✓	107_items_Get_Item	Прочитане на данните за артикул
✓	107_items_Get_FirstFoundItem	Прочитане на данните за първия намерен артикул
✓	107_items_Get_LastFoundItem	Прочитане на данните за последния намерен артикул
✓	107_items_Get_NextItem	Прочитане на данните за следващия намерен артикул
✓	107_items_Get_FirstSoldItem	Прочитане на данните за първия намерен продаден артикул
✓	107_items_Get_LastSoldItem	Прочитане на данните за последния намерен продаден артикул
✓	107_items_Get_NextSoldItem	Прочитане на данните за следващия намерен продаден артикул
✓	107_items_Get_FirstNotProgrammedItem	Намиране на ПЛУ за първия непрограмиран(свободен) артикул
✓	107_items_Get_LastNotProgrammedItem	Намиране на ПЛУ за последния непрограмиран(свободен) артикул

## Команди за конфигуриране

Команди към устройства в група „А“		
✓	036_config_SetLAN	Настройка параметри за връзка по LAN
✓	041_config_Set_Switches	Запис на конфигурационни ключета
✓	043_config_Set_HeaderLine	Запис на хедър линии
✓	043_config_Set_FooterLine	Запис на футър линии
✓	043_config_Set_AutoFormat	Автоматично форматиране на продажбите като за фактура
✓	043_config_Set_BarcodeHeight	Настройка на височина на баркода
✓	043_config_Set_AutoCutting	Разрешаване/Забрана на авт. отрязване на хартията
✓	043_config_Set_PrintDensity	Задава плътност на печат
✓	043_config_enable_EUR	Разрешаване на печата на общата сума в EUR при първата команда за плащане
✓	043_config_disable_EUR	Забраняване на печата на общата сума в EUR при първата команда за плащане
✓	043_config_Set_FontSize	Задаване на шрифта за печат на принтера
✓	043_config_Set_PrintLogo	Разрешаване/Забрана печата на графичното лого непосредствено преди HEADER
✓	043_config_Set_PrintDepartment	Разрешаване/Забрана на отпечатването на департамента в началото на всеки фискален бон
✓	043_config_enable_PrintSuppression	Разрешаване на подтискането на печата във фискален или служебен бон
✓	043_config_disable_PrintSuppression	Забраняване на подтискането на печата във фискален или служебен бон
✓	043_config_Set_ExtraSpace	Задава допълнително разстояние в пиксели между буквите в завъртян на 90 градуса служебен бон
✓	043_config_Set_SIM	Избира коя от SIM-картите да използва GPRS модемът
✓	043_config_Set_PrintTaxDDS	Разрешаване/Забрана на отпечатването на натрупания ДДС за бона в нормален фискален бон
✓	043_config_Set_CashDrawerPulse	Разрешаване/Забрана на автоматичното генериране на импулс за отваряне на чекмедже
✓	043_config_Set_WarningTimeRange	Задава времето на което да сработи предупреждението за не предадени данни
✓	043_config_Set_FeedButton	Разрешаване/Забрана на функциите на бутона FEED
✓	043_config_Set_OptionalErrorText	Разрешаване/Забрана на връщането на общ. текст в отговора на командата при настъпила грешка
✓	061_config_Set_DateTime	Задаване на дата и час
✓	066_config_Set_InvoiceRange	Задаване на интервал от номера на фактури
✓	075_config_Restore_PrintSuppression	Възобновява подтиснатия печат в бон в следствие на команда 43, подкоманда 'Q'
✓	085_config_Set_AdditionalPaymentName	Задаване на текст към допълнително плащане
✓	087_config_Set_DepartmentName	Програмиране на име на департамент
✓	087_config_DepartmentNameTwoRows	Програмиране на име на департамент в два реда
✓	101_config_Set_OperatorPassword	Задаване на операторска парола
✓	102_config_Set_OperatorName	Задаване на име на оператор

## Получаване на информация от фискалното устройство

Команди към устройства в група „А“		
✓ 034_info_Service_Contracts	Информация за сервисния договор	
✓ 036_info_Get_LANSettings	Информация за настройките свързани с управление по LAN	
✓ 043_info_Get_PrintOption	Информация за настройките зададени с команда 43	
✓ 050_info_Get_TaxRatesByPeriod	Печат на промените на десетичните знаци и данъчните ставки за посочения период	
✓ 062_info_Get_DateTime	Получаване на дата и час	
✓ 062_info_Get_DateTime_01	Получаване на дата и час, данните са разбити по параметри	
✓ 064_info_Get_LastFiscRecord	Получаване на информацията от последния запис (Z-отчет) във фискалната памет	
✓ 064_info_Get_LastFiscRecord_01	Получаване на информацията от последния запис (Z-отчет) във фискалната памет	
✓ 065_info_Get_AdditionalDailyInfo_01	Получаване на сумите по данъчни групи от последното приключване на деня	
✓ 066_info_Get_InvoiceRange	Получаване на текущия интервал от допустими номера на фактури	
✓ 068_info_Get_FreeFMRRecords	Връща се броя на свободните полета във фискалната памет(брой Z-отчета)	
✓ 070_info_Get_CashIn_CashOut	Касова наличност	
✓ 071_info_Print_Diagnostic_0	Отпечатва диагностична информация	
✓ 074_info_Get_StatusBytes	Получава статусбайтовете на устройството като данни	
✓ 074_info_Get_NotPrintedRowsCount	Връща брой оставащи редове за печат	
✓ 074_info_Get_NRADocuments	Връща данни за изпратените към сървъра на НАП клиентски документи	
✓ 076_info_Get_FTransactionStatus	Статус на фискалната транзакция	
✓ 079_info_Print_Short_FMRReportByDTRange	Изчисляване и отпечатване на съкратен отчет на фискалната памет от дата до дата	
✓ 079_info_Print_Short_MonthlyFMRReport	Изчисляване и отпечатване на месечен съкратен отчет на фискалната памет	
✓ 079_info_Print_Short_YearlyFMRReport	Изчисляване и отпечатване на годишен съкратен отчет на фискалната памет	
✓ 081_info_Get_VoltageAndTemperature	Данни за захранването(напрежение във волтове) и температура на печатащата глава	
✓ 083_info_Get_DecimalAndTaxRates	Валидните за момента десетични знаци, валута и стойност на данъчни ставки	
✓ 085_info_Get_AdditionalPaymentNames	Получаване на текста към допълнителните плащания	
✓ 086_info_Get_FMRRecord_LastDateTime	Датата и часа на последния запис(Z-отчет) във фискалната памет	
✓ 086_info_Get_FMRRecord_LastDateTime_01	Датата и часа на последния запис(Z-отчет) във фискалната памет	
✓ 088_info_Get_DepartmentInfo	Получаване на данни за натрупаните съми по департамент	
✓ 090_info_Get_DiagnosticInfo	Получаване на диагностична информация	
✓ 093_info_Get_DailyCorrections	Получаване на данни за отстъпки и надбавки през деня	
✓ 097_info_Get_TaxRates	Получаване на текущите данъчни ставки	
✓ 099_info_Get_EIK	Получаване на данни за ЕИК на собственика	
✓ 103_info_Get_CurrentReceiptInfo	Информация за натрупаните суми и дали е възможно връщане на регистрирани стоки	
✓ 110_info_Get_AdditionalDailyInfo_02	Информация за разпределението на сумата за деня по начин на плащане	
✓ 110_info_Get_AdditionalDailyInfo_03	Разпределението на сумите за деня по начин на плащане с доп. плащания	
✓ 112_info_Get_OperatorsData	Получаване на информацията за оператор, отпечатвана при отчет	
✓ 113_info_Get_LastDocumentNumber	Получаване на номера на последния отпечатан документ	
✓ 114_info_Get_ActiveTaxes	Информация за активните данъчни ставки за въпросния Z-отчет (запис)	
✓ 114_info_Get_Sums	Информация за оборота за посочения запис или период	
✓ 114_info_Get_NettoSums	Информация за нето-сумите за посочения запис или период	
✓ 114_info_Get_VATSums	Информация за начисления ДДС за посочения запис или период	
✓ 114_info_Get_Numbers	Допълнителна информация за посочения запис	
✓ 114_info_Get_TaxRateValues	Информация за посочения запис от фискалната памет за задаване на данъчни ставки	
✓ 114_info_Get_RAMResetDateTime	Информация за посочения запис от ФП с нулиране на RAM	
✓ 114_info_Get_AdditionalPayments	Информация за сумите по тип плащане за посочения запис или период	
✓ 114_info_Get_Corrections	Информация за сумите за корекции за посочения запис или период	
✓ 114_info_Get_CashInCashOut	Информация за сумите при служебно въвеждане и извеждане на суми	
✓ 114_info_Get_Turnover	Информация за общия оборот и ДДС до посочения Z-отчет	

## Работа с КЛЕН

Команди към устройства в група „А“		
✓ 119_klen_Print_Validation	Проверка валидността на КЛЕН	
✓ 119_klen_Print_SHA1	Проверка валидността на КЛЕН плюс SHA1	
✓ 119_klen_Get_KLENInfo_ZReport	Връща информация от КЛЕН за Z-отчет	
✓ 119_klen_Get_ZReportInfo	Връща информация от КЛЕН за Z-отчет	
✓ 119_klen_Get_Info	Връща обща информация за КЛЕН	
✓ 119_klen_Find_Document	Параметри за търсене в КЛЕН по номер на документ и получава първия текстови ред от КЛЕН	
✓ 119_klen_Set_DocsRange	Параметри за търсене в КЛЕН (от номер до номер) и получава първия текстови ред от КЛЕН	
✓ 119_klen_Set_DocsRange_InZReport_01	Параметри за търсене в КЛЕН - Z-отчет и получава първия текстови ред от КЛЕН	
✓ 119_klen_Set_DocsRange_InZReport_02	Параметри за търсене в КЛЕН - Z-отчет и номер и получава първия текстови ред от КЛЕН	
✓ 119_klen_Set_DocsRange_InZReport	Параметри за търсене в КЛЕН - Z-отчет, от номер до номер и получава първия текстови ред	
✓ 119_klen_Set_DocsRange_ByDateTime	Параметри за търсене в КЛЕН - (от дата до дата) и получава първия текстови ред от КЛЕН	
✓ 119_klen_Get_NextTextRow	Получава следващия текстови ред от КЛЕН по зададените параметри	
✓ 119_klen_Print_Document	Отпечатва документ от КЛЕН по номер	
✓ 119_klen_Print_DocsInRange	Отпечатва документи от КЛЕН от номер до номер	
✓ 119_klen_Print_DocsInRange_InZReport	Отпечатва документи от КЛЕН от номер до номер за даден Z-отчет	
✓ 119_klen_Print_DocsFromDateToDate	Отпечатва документи от КЛЕН от дата до дата	

## Структурирана информация

Команди към устройства в група „А“		
✓	Справки от тип „W“	
✓	119_klen_SInfo_W_ByNumber	Намиране на документ и връщане на информация по номер на документ
✓	119_klen_SInfo_W_ByNumbersRange	Намиране на документ и връщане на информация от номер до номер
✓	119_klen_SInfo_W_ByDateRange	Намиране на документ и връщане на информация от дата до дата
✓	119_klen_SInfo_W_GetNext	Връщане на следващ ред структурирана информация
✓	Справки от тип „Y“	
✓	119_klen_SInfo_Y_ByNumber	Намиране на документ и връщане на информация по номер на документ
✓	119_klen_SInfo_Y_ByNumbersRange	Намиране на документ и връщане на информация от номер до номер
✓	119_klen_SInfo_Y_ByDateRange	Намиране на документ и връщане на информация от дата до дата
✓	119_klen_SInfo_Y_GetNext	Връщане на следващ ред структурирана информация
✓	Справки от тип „V“	
✓	119_klen_SInfo_V_ByNumber	Намиране на документ и връщане на информация по номер на документ
✓	119_klen_SInfo_V_ByNumbersRange	Намиране на документ и връщане на информация от номер до номер
✓	119_klen_SInfo_V_ByDateRange	Намиране на документ и връщане на информация от дата до дата
✓	119_klen_SInfo_V_GetNext	Връщане на следващ ред структурирана информация
✓	Справки от тип „V+“	
✓	119_klen_SInfo_ExV_ByNumber	Намиране на документ и връщане на информация по номер на документ
✓	119_klen_SInfo_ExV_ByNumbersRange	Намиране на документ и връщане на информация от номер до номер
✓	119_klen_SInfo_ExV_ByDateRange	Намиране на документ и връщане на информация от дата до дата
✓	119_klen_SInfo_ExV_GetNext	Връщане на следващ ред структурирана информация

## Сервизни и други команди

Команди към устройства в група „А“		
✓	080_other_Sound_Signal	Издаване на звуков сигнал
✓	072_service_Fiscalization	Само за сервиз - Фискализация
✓	083_service_Set_DecimalAndTaxRates	Само за сервиз - Настройка на десетичен знак, валута и данъчни ставки
✓	089_service_Set_ProductionTestArea	Само за сервиз - Тестване на фискалната памет
✓	091_service_Set_SerialNumber	Само за сервиз - запис на сериен номер и номер на фискалната памет
✓	098_service_Set_EIK	Само за сервиз - установяване на ЕИК
✓	133_service_Disable_Print	Само за сервизи - забрана на печата
✓	135_service_Test_GPRS	Само за сервизи - тест на модема

## Машинно-зависими команди в логическа група „В“

### Външен дисплей

Команди към устройства в група „В“	
✓ 033_display_Clear	Изчистване на дисплея
✓ 035_display_Show_LowerLine	Показване на текст (долен ред).
✓ 047_display_Show_UpperLine	Показване на текст (горен ред).
✓ 063_display_Show_DateTime	Показване на датата и часа.
✓ 100_display_Show_Text	Дисплей - пълен контрол

### Нефискален бон

Команди към устройства в група „В“	
✓ 038_receipt_NonFiscal_Open	Отваряне на нефискален бон
✓ 042_receipt_NonFiscal_Text	Печат на нефискален текст
✓ 042_receipt_PNonFiscal_Text	Печат на параметризиран нефискален текст
✓ 039_receipt_NonFiscal_Close	Затваряне на нефискален бон

### Сторниране

Команди към устройства в група „В“	
<a href="#">(вж. Таблицата)</a>	
✓ 046_receipt_StornoOpen_B01	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_B02	Вариант на командата за отваряне на сторно бон (по касов бон)
✓ 046_receipt_StornoOpen_B03	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_B04	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_B05	Вариант на командата за отваряне на сторно бон (по фактура)
✓ 046_receipt_StornoOpen_B06	Вариант на командата за отваряне на сторно бон (по фактура)

### Отваряне на фискален бон и фактура

Команди към устройства в група „В“	
<a href="#">(вж. Таблицата)</a>	
✓ 048_receipt_FiscalOpen_B01	Вариант на командата за отваряне на фискален бон
✓ 048_receipt_FiscalOpen_B02	Вариант на командата за отваряне на фискален бон тип фактура
✓ 048_receipt_Fiscal_Open	Вариант на командата за отваряне на фискален бон с УНП
✓ 048_receipt_FiscalOpen_B03	Вариант на командата за отваряне на фискален бон с УНП
✓ 048_receipt_FiscalOpen_B04	Вариант на командата за отваряне на фискален бон тип фактура с УНП
✓ 048_receipt_Invoice_Open	Вариант на командата за отваряне на фискален бон тип фактура с УНП
✓ 056_receipt_Fiscal_Close	Затваряне на фискален бон или фактура

## Печат на данни за клиента преди команда за затваряне на фактура

Команди към устройства в група „В“		
✓	057_receipt_PrintClientInfo_08	Вариант на команда за печат на данни за клиента
✓	057_receipt_PrintClientInfo_09	Вариант на команда за печат на данни за клиента
✓	057_receipt_PrintClientInfo_10	Вариант на команда за печат на данни за клиента
✓	057_receipt_PrintClientInfo_11	Вариант на команда за печат на данни за клиента
✓	057_receipt_PrintClientInfo_12	Вариант на команда за печат на данни за клиента
✓	057_receipt_PrintClientInfo_13	Вариант на команда за печат на данни за клиента
✓	057_receipt_PrintClientInfo_14	Вариант на команда за печат на данни за клиента

## Регистриране на продажба

Команди към устройства в група „В“		
✓	049_receipt_Sale	Продажба
✓	049_receipt_Sale_TextRow1	Продажба с текст в първи ред
✓	049_receipt_Sale_TextRow2	Продажба с текст във втори ред
✓	049_receipt_Sale_Minimum	Продажба - минимум параметри

## Регистриране на продажба с корекция

Команди към устройства в група „В“		
✓	049_receipt_Sale_CByPercent	Продажба с корекция по процент
✓	049_receipt_Sale_TextRow1CByPercent	Продажба с корекция по процент и текст в първи ред
✓	049_receipt_Sale_TextRow2CByPercent	Продажба с корекция по процент и текст във втори ред
✓	049_receipt_Sale_CByPercentWText	Продажба с корекция по процент - без текст
✓	049_receipt_Sale_CBySum	Продажба с корекция чрез твърда сума
✓	049_receipt_Sale_TextRow1CBySum	Продажба с корекция чрез твърда сума(текст в първи ред)
✓	049_receipt_Sale_TextRow2CBySum	Продажба с корекция чрез твърда сума(текст във втори ред)
✓	049_receipt_Sale_CBySumWText	Продажба с корекция чрез твърда сума – без текст

## Регистриране на продажба в департамент

Команди към устройства в група „В“		
✓	049_receipt_DSale	Продажба в департамент
✓	049_receipt_DSale_TextRow1	Продажба в департамент(текст в първи ред)
✓	049_receipt_DSale_TextRow2	Продажба в департамент(текст във втори ред)
✓	049_receipt_DSale_Minimum	Продажба в департамент с корекция по процент – без текст

## Регистриране на продажба в департамент с корекция

Команди към устройства в група „В“		
✓	049_receipt_DSale_CByPercent	Продажба в департамент с корекция по %
✓	049_receipt_DSale_TextRow1CByPercent	Продажба в департамент с корекция по %(текст в първи ред)
✓	049_receipt_DSale_TextRow2CByPercent	Продажба в департамент с корекция по %(текст във втори ред)
✓	049_receipt_DSale_CByPercentWText	Продажба в департамент с корекция по процент – без текст
✓	049_receipt_DSale_CBySum	Продажба в департамент с корекция чрез твърда сума
✓	049_receipt_DSale_TextRow1CBySum	Продажба в департамент с корекция чрез твърда сума(текст в първи ред)
✓	049_receipt_DSale_TextRow2CBySum	Продажба в департамент с корекция чрез твърда сума(текст във втори ред)
✓	049_receipt_DSale_CBySumWText	Продажба в департамент с корекция чрез твърда сума – без текст

## **Регистриране на продажба и автоматично показване на външен дисплей**

Команди към устройства в група „В“		
✓	052_receipt_DisplaySale	Продажба с авт.показване на външен дисплей
✓	052_receipt_DisplaySale_CByPercent	Продажба с авт.показване на външен дисплей и корекция по процент
✓	052_receipt_DisplaySale_CBySum	Продажба с авт.показване на външен дисплей и корекция със сума
✓	052_receipt_DisplaySale_Minimum	Продажба с авт.показване на външен дисплей - минимум параметри

## **Регистриране на продажба в департамент и автоматично показване на външен дисплей**

Команди към устройства в група „В“		
✓	052_receipt_DisplayDSale	Продажба с авт.показване на външен дисплей
✓	052_receipt_DisplayDSale_CByPercent	Продажба с авт.показване на външен дисплей и корекция по процент
✓	052_receipt_DisplayDSale_CBySum	Продажба с авт.показване на външен дисплей и корекция със сума
✓	052_receipt_DisplayDSale_Minimum	Продажба с авт.показване на външен дисплей - минимум параметри

## **Регистриране на продажба на артикул**

Команди към устройства в група „В“		
✓	058_receipt_PLU_Sale	Продажба на артикул
✓	058_receipt_PLUSale_CByPercent	Продажба на артикул и корекция по процент
✓	058_receipt_PLUSale_CBySum	Продажба на артикул и корекция със сума

## **Регистриране на продажба на артикул в департамент**

Команди към устройства в група „В“		
✓	058_receipt_PLUDep_Sale	Продажба на артикул в департамент
✓	058_receipt_PLUDep_Sale_CByPercent	Продажба на артикул в департамент и корекция по процент
✓	058_receipt_PLUDep_Sale_CBySum	Продажба на артикул в департамент и корекция със сума

## **Регистриране на продажба на артикул и автоматично показване на външен дисплей**

Команди към устройства в група „В“		
✓	058_receipt_DisplayPLUSale	Продажба на артикул с авт.показване на дисплей
✓	058_receipt_DisplayPLUSale_CByPercent	Продажба на артикул с авт.показване на дисплей и корекция по процент
✓	058_receipt_DisplayPLUSale_CBySum	Продажба на артикул с авт.показване на дисплей и корекция със сума

## Регистриране на продажба на артикул в департамент и авт. показване на външен дисплей

Команди към устройства в група „В“	
✓ 058_receipt_DisplayDepPLU_Sale	Продажба на артикул в департамент с авт.показване на дисплей
✓ 058_receipt_DisplayDepPLU_Sale_CByPercent	Продажба на артикул в департамент с авт.показване на дисплей и корекция по процент
✓ 058_receipt_DisplayDepPLU_Sale_CBySum	Продажба на артикул в департамент с авт.показване на дисплей и корекция със сума

## Субтотал и Тотал

Команди към устройства в група „В“	
✓ 051_receipt_Subtotal	Команда за субтотал
✓ 051_receipt_Subtotal_CByPercent	Команда за субтотал с корекция по процент
✓ 051_receipt_Subtotal_CBySum	Команда за субтотал с корекция със сума
✓ 053_receipt_Total_PAmount	Команда за тотал с тип на плащане (два реда текст)
✓ 053_receipt_Total_PAmountTextRow1	Команда за тотал с тип на плащане (текст в първи ред)
✓ 053_receipt_Total_PAmountTextRow2	Команда за тотал с тип на плащане (текст във втори ред)
✓ 053_receipt_Total_PAmountWithoutText	Команда за тотал с тип на плащане (без текст)
✓ 053_receipt_Total	Команда за тотал с тип на плащане в брой (два реда текст)
✓ 053_receipt_Total_TextRow1	Команда за тотал с тип на плащане в брой (текст в първи ред)
✓ 053_receipt_Total_TextRow2	Команда за тотал с тип на плащане в брой (текст във втори ред)
✓ 053_receipt_Total_WithoutText	Команда за тотал с тип на плащане в брой (без текст)
✓ 053_receipt_PinPad_Total	Команда за тотал с тип на плащане с дебитна карта (два реда текст)
✓ 053_receipt_PinPad_TotalTextRow1	Команда за тотал с тип на плащане с дебитна карта (текст в първи ред)
✓ 053_receipt_PinPad_TotalTextRow2	Команда за тотал с тип на плащане с дебитна карта (текст във втори ред)
✓ 053_receipt_PinPad_TotalWText	Команда за тотал с тип на плащане с дебитна карта (без текст)

## Други команди свързани с издаването на фискални и нефискални документи

Команди към устройства в група „В“	
✓ 044_receipt_Paper_Feed	Придвижване на хартията
✓ 054_receipt_Fiscal_Text	Отпечатване на фискален текст
✓ 060_receipt_Fiscal_Cancel	Анулиране на фискален бон
✓ 070_receipt_CashIn_CashOut	Служебно въвеждане/извеждане на суми
✓ 092_receipt_Separating_Line	Отпечатване на разделителна линия
✓ 103_receipt_Current_Info	Информация за текущия бон
✓ 106_receipt_Drawer_KickOut	Отваряне на чекмедже
✓ 109_receipt_Print_Duplicate	Отпечатване на копие на последния затворен бон с продажби

## Отчети

Команди към устройства в група „В“	
✓ 069_report_DailyClosure_01	Дневен отчет със или без нулиране
✓ 073_report_FMByNumRange	Отпечатване на пълен отчет на фискалната памет от даден номер до друг
✓ 079_report_FMByDateRange_Short	Отпечатване на съкратен отчет на фискалната памет от дата до дата
✓ 094_report_FMByDateRange	Пълен отчет на фискалната памет за периода между две дати
✓ 095_report_FMByNumRange_Short	Отпечатване на съкратен отчет на фискалната памет от номер до номер
✓ 105_report_Operators	Отпечатване на информация за продажбите по оператори
✓ 108_report_ExtDailyClosure_01	Отпечатване на дневен отчет със или без нулиране плюс артикули по които има продажби
✓ 111_report_Items	Отчет по артикули
✓ 111_report_Items_InRange	Отчет по артикули от номер до номер
✓ 111_report_Items_InRangeByGroup	Отчет по артикули от номер до номер и в група
✓ 117_report_DailyClosureByDepartments_01	Дневен отчет със или без нулиране и печат на департаменти с продажби за деня
✓ 118_report_DailyClosureByDepartmentsAndItems_01	Екв. на 069_report_DailyClosure_01 плюс печат на деп. с продажби и артикули



## Работа с артикули

Команди към устройства в група „В“		
✓	107_items_Get_ItemsInformation	Получаване на обща информация за възможностите на устройството по отн. артикули
✓	107_items_Set_Item	Програмиране на артикул
✓	107_items_Set_ItemWithReplace	Програмиране на артикул с промяна на количеството
✓	107_items_Set_ItemQuantity	Промяна на наличното количество за артикул
✓	107_items_Delete_Item	Изтриване на артикул, ако няма натрупани суми
✓	107_items_Delete_ItemsInRange	Изтриване на артикули в интервал, ако няма натрупани суми за тях
✓	107_items_Delete_All_Items	Изтриване на всички артикули
✓	107_items_Get_Item	Прочитане на данните за артикул
✓	107_items_Get_FirstFoundItem	Прочитане на данните за първия намерен артикул
✓	107_items_Get_LastFoundItem	Прочитане на данните за последния намерен артикул
✓	107_items_Get_NextItem	Прочитане на данните за следващия намерен артикул
✓	107_items_Get_FirstSoldItem	Прочитане на данните за първия намерен продаден артикул
✓	107_items_Get_LastSoldItem	Прочитане на данните за последния намерен продаден артикул
✓	107_items_Get_NextSoldItem	Прочитане на данните за следващия намерен продаден артикул
✓	107_items_Get_FirstNotProgrammedItem	Намиране на ПЛУ за първия непрограмиран(свободен) артикул
✓	107_items_Get_LastNotProgrammedItem	Намиране на ПЛУ за последния непрограмиран(свободен) артикул

## Работа с клиенти

Команди към устройства в група „В“		
✓	140_clients_Set_ClientData	Програмиране на данни за клиент
✓	140_clients_Del_ClientData	Изтриване на данни за клиент
✓	140_clients_Get_ClientData	Получаване на данни за програмиран клиент
✓	140_clients_Set_SellerName	Запис на име на продавач
✓	140_clients_Get_SellerName	Получаване на име на продавач
✓	140_clients_Get_FirstClientData	Четене на данните за първия клиент записан в паметта на апарата
✓	140_clients_Get_NextClientData	Четене на данните за следващия клиент записан в паметта на апарата
✓	140_clients_Del_AllClientData	Изтриване на данните за всички клиенти записани в паметта на апарата

## Команди за конфигуриране

Команди към устройства в група „В“		
✓	043_config_Set_HeaderLine	Запис на хедър линии
✓	043_config_Set_FooterLine	Запис на футър линии
✓	043_config_Set_AutoFormat	Автоматично форматиране на продажбите като за фактура
✓	043_config_Set_PrintDensity	Задава плътност на печат
✓	043_config_enable_EUR	Разрешаване на печата на общата сума в EUR при първата команда за плащане
✓	043_config_disable_EUR	Забраняване на печата на общата сума в EUR при първата команда за плащане
✓	043_config_Set_PrintLogo	Разрешаване/Забрана печата на графичното лого непосредствено преди HEADER
✓	043_config_Set_PrintDepartment	Разрешаване/Забрана на отпечатването на името на департамента
✓	043_config_Set_PrintTaxDDS	Разрешаване/Забрана на отпечатването на натрупания ДДС
✓	043_config_Set_CashDrawerPulse	Разрешаване/Забрана на авт. генерирането на импулс за отваряне на чекмедже
✓	061_config_Set_DateTime	Задаване на дата и час
✓	066_config_Set_InvoiceRange	Задаване на интервал от номера на фактури
✓	085_config_Set_AdditionalPaymentName	Задаване на текст към допълнително плащане
✓	087_config_Set_DepartmentName	Програмиране на име на департамент
✓	087_config_Set_DepartmentNameTwoRows	Програмиране на име на департамент в два реда
✓	101_config_Set_OperatorPassword	Задаване на операторска парола
✓	102_config_Set_OperatorName	Задаване на име на оператор

## Получаване на информация от фискалното устройство

Команди към устройства в група „В“		
✓ 043_info_Get_PrintOption	Информация за настройките зададени с команда 43	
✓ 050_info_Get_TaxRatesByPeriod	Печат на промените на десетичните знаци и данъчните ставки за посочения период	
✓ 062_info_Get_DateTime	Получаване на дата и час	
✓ 062_info_Get_DateTime_01	Получаване на дата и час, данните са разбити по параметри	
✓ 064_info_Get_LastFiscRecord	Получаване на информацията от последния запис(Z-отчет) във фискалната памет	
✓ 065_info_Get_AdditionalDailyInfo_04	Оборот по данъчни групи или ДДС по данъчни групи (в зависимост от опцията)	
✓ 065_info_Get_AdditionalDailyInfo_05	Дневни броячи и суми	
✓ 065_info_Get_AdditionalDailyInfo_06	Дневни броячи и суми	
✓ 065_info_Get_AdditionalDailyInfo_07	Дневни броячи и суми	
✓ 065_info_Get_AdditionalDailyInfo_08	Сторно оборот по данъчни групи	
✓ 065_info_Get_AdditionalDailyInfo_09	Сторно ДДС по данъчни групи	
✓ 065_info_Get_AdditionalDailyInfo_10	Суми по плащания СТОРНО	
✓ 066_info_Get_InvoiceRange	Получаване на текущия интервал от допустими номера на фактури	
✓ 068_info_Get_FreeFMRRecords	Броя на свободните полета във фискалната памет(брой Z-отчета)	
✓ 068_info_Get_FreeFMRRecords_01	Броя на свободните полета във фискалната памет(брой Z-отчета)(поддръжка за преходен период)	
✓ 070_info_Get_CashIn_CashOut	Касова наличност	
✓ 071_info_Print_Diagnostic_0	Отпечатва диагностична информация	
✓ 071_info_Print_Diagnostic_1	Отпечатва диагностична информация (вариант)	
✓ 071_info_PrintInfo_GPRS	Печат на тест - GPRS	
✓ 071_info_PrintInfo_TaxTerminal	Печат на информация за данъчния терминал	
✓ 071_info_Get_TaxTerminalInfo	Информация за данъчния терминал	
✓ 071_info_Get_ModemInfo	Информация за модема.	
✓ 071_info_Get_FlashMemoryInfo	Информация за флеш паметта	
✓ 074_info_Get_StatusBytes	Получава статусбайтовете на устройството като данни	
✓ 076_info_Get_FTransactionStatus	Статус на фискалната транзакция	
✓ 079_info_Print_Short_FMRReportByDTRange	Изчисляване и отпечатване на съкратен отчет на фискалната памет от дата до дата	
✓ 079_info_Print_Short_MonthlyFMRReport	Изчисляване и отпечатване на месечен съкратен отчет на фискалната памет	
✓ 079_info_Print_Short_YearlyFMRReport	Изчисляване и отпечатване на годишен съкратен отчет на фискалната памет	
✓ 083_info_Get_DecimalAndTaxRates	Връща валидните за момента стойности на десетични знаци, валута и стойност на данъчни ставки	
✓ 085_info_Get_AdditionalPaymentNames	Получаване на текста към допълнителните плащания	
✓ 086_info_Get_FMRRecord_LastDateTime	Датата и часа на последния запис(Z-отчет) във фискалната памет	
✓ 086_info_Get_FMRRecord_LastDateTime_01	Датата и часа на последния запис(Z-отчет) във фискалната памет, данните са разбити по параметри	
✓ 086_info_Get_Registration_DT_Num	Четене на запис за регистрация	
✓ 086_info_Get_DeRegistration_DT_Num	Четене на запис за deregistration	
✓ 086_info_Get_Registration_Count	Брой на направените до момента регистрации	
✓ 086_info_Get_DeRegistration_Count	Брой на направените до момента deregistration	
✓ 088_info_Get_DepartmentInfo	Получаване на данни за натрупаните суми по департамент	
✓ 088_info_Get_Department_SornoInfo	Получаване на данни за натрупаните сторно суми по департамент	
✓ 090_info_Get_DiagnosticInfo	Получаване на диагностична информация	
✓ 097_info_Get_TaxRates	Получаване на текущите данъчни ставки	
✓ 099_info_Get_EIK	Получаване на данни за ЕИК на собственика	
✓ 103_info_Get_CurrentReceiptInfo	Натрупани суми по данъчни групи и дали е възможно връщане на регистрирани стоки	
✓ 110_info_Get_AdditionalDailyInfo_11	Информация за разпределението на сумата за деня по начин на плащане	
✓ 112_info_Get_OperatorsData	Получаване на информацията за оператор, отпечатвана при отчет	
✓ 112_info_Get_OperatorsSornoData	Информация за сторно операции за оператор	
✓ 113_info_Get_LastDocumentNumber	Получаване на номера на последния отпечатан документ	
✓ 114_info_Get_ActiveTaxes	Информация за активните данъчни ставки за въпросния Z-отчет (запис)	
✓ 114_info_Get_Sums	Информация за оборота за посочения запис или период	
✓ 114_info_Get_NettoSums	Информация за нето-сумите за посочения запис или период	
✓ 114_info_Get_VAT_Sums	Информация за начисления ДДС за посочения запис или период	
✓ 114_info_Get_Numbers	Допълнителна информация за посочения запис	
✓ 114_info_Get_TaxRateValues	Информация за посочения запис от фискалната памет за задаване на данъчни ставки	
✓ 114_info_Get_ReversalTurnover	Информация за оборота от сторно за посочения запис или период	
✓ 114_info_Get_ReversalNetoSums	Информация за нето сумите от сторно за посочения запис или период	
✓ 114_info_Get_ReversalVAT_Sums	Информация за начисления ДДС от сторно за посочения запис или период	

## Работа с КЛЕН и структурирана информация

Команди към устройства в група „В“		
✓ 119_klen_Set_DocsRange_ByDateTime	Задава параметри за търсене в КЛЕН - (интервал от дата до дата) и получава първия текстови ред	
✓ 119_klen_Get_NextTextRow	Четене на следващ текстов ред от КЛЕН	
✓ 124_klen_FindRange_ByDateTime	Намиране на документ за структурирана информация от дата до дата	
✓ 124_klen_FindRange_ByZReports	Намиране на документ за структурирана информация от Z-отчет до Z-отчет	
✓ 124_klen_Get_Info	Връща обща информация за КЛЕН	
✓ 125_klen_Prepare_Document	Подготовка на документа за четене	
✓ 125_klen_Prepare_DocumentInRange	Подготовка на документите за четене	
✓ 125_klen_Get_TextRow	Четене на следващ текстов ред от документа	
✓ 125_klen_Print_Document	Отпечатване на документа в текстов вид на хартия	
✓ 125_klen_Get_SInfo	Четене на данни от фискални бонове по прил.34	

## Сервизни и други команди

Команди към устройства в група „В“	
✓ 080_other_Sound_Signal	Издаване на звуков сигнал
✓ 083_service_Set_DecimalAndTaxRates	Само за сервиз - Настройка на десетичен знак, валута и данъчни ставки
✓ 089_service_Set_ProductionTestArea	Само за сервиз - Тестване на фискалната памет

## Машинно-зависими команди в логическа група „С“

### Външен дисплей

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	033_display_Clear	Изтриване на дисплея
✓	035_display_Show_LowerLine:	Показване на текст на долната линия на дисплея
✓	047_display_Show_UpperLine:	Показване на текст на горната линия на дисплея
✓	063_display_Show_DateTime:	Показване на дата и час

### Печат на фискални и нефискални бележки

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	038_receipt_NonFiscal_Open	Старо име:038_open_NonFiscalReceipt
✓	039_receipt_NonFiscal_Close	Старо име:039_close_NonFiscalReceipt
✓	042_receipt_NonFiscal_Text	Старо име:042_print_NonFiscalText
✓	042_receipt_PNonFiscal_Text	Печат на форматиран нефискален текст
✓	044_receipt_Paper_Feed	Придвижване на хартията
✓	046_receipt_Paper_Cut	Отрязване на хартията
✓	048_receipt_FiscalOpen_C01	Отваряне на документ "Касов бон"
✓	048_receipt_FiscalOpen_C02	Отваряне на документ "Фактура"
✓	048_receipt_FiscalOpen_C03	Отваряне на документ "Касов бон с УНП"
✓	048_receipt_FiscalOpen_C04	Отваряне на документ "Фактура с УНП"
✓	049_receipt_Sale	Продажба
✓	049_receipt_Sale_Un	Продажба с употреба на мерна единица
✓	056_receipt_Fiscal_Close	Затваряне на фискален бон или фактура
✓	057_receipt_PrintClientInfo_15	Печат на данни за клиента във фактура
✓	053_receipt_Total	Тотал на фискален бон или фактура
✓	053_receipt_Total_Currency	Тотал на фискален бон или фактура
✓	054_receipt_Fiscal_Text	Отпечатване на фискален свободен текст
✓	054_receipt_PFiscal_Text	Отпечатване на форматиран фискален свободен текст
✓	058_receipt_PLU_Sale	Продажба на програмиран артикул
✓	060_receipt_Fiscal_Cancel	Анулиране на фискален бон
✓	070_receipt_CashIn_CashOut	Служебен внос/износ в лева или валута
✓	084_receipt_Print_Barcode	Печат на баркод
✓	092_receipt_Separating_Line	Печат на разделителна линия
✓	103_receipt_Current_Info	Информация за натрупаните суми по данъчни групи и вида на фиск. бон
✓	106_receipt_Drawer_KickOut	Отваряне на чекмедже
✓	109_receipt_Print_Duplicate	Отпечатва на копие на последния затворен фискален бон с продажби
✓	122_receipt_PVerticalFiscal_Text	Печат на завъртян на 90гр. параметризиран текст
✓	127_receipt_Print_Stamp	Отпечатване на шампа по име

### Приключване на деня и отчети

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	069_report_DailyClosure_01:	Z и X отчети - печат и получаване на данни
✓	069_report_DailyClosure_02:	Отчети по департаменти и артикули - печат
✓	087_report_Item_Groups:	Печат на информация за стокови групи
✓	088_report_Department_Info:	Печат на информация за департаменти
✓	095_report_FMByNumRange:	Детайлен отчет на фиск.памет от номер до номер
✓	095_report_FMByNumRange:	Съкратен отчет на фиск.памет от номер до номер
✓	105_report_Operators:	Отчет по оператори
✓	111_report_Items:	Отпечатване на отчети по артикули

## Артикули

Команди към устройства в група „С“, налични от версия 3.0.0.7	
✓ 107_items_Get_ItemsInformation	Информация за данните на програмируемия артикул
✓ 107_items_Set_Item	Програмиране на артикул
✓ 107_items_Set_ItemUn	Програмиране на артикул с мерна единица
✓ 107_items_Set_ItemQuantity	Промяна на количеството на програмиран артикул
✓ 107_items_Delete_Item	Изтриване на артикул
✓ 107_items_Delete_ItemsInRange	Изтриване на артикули в интервал
✓ 107_items_Delete_All_Items	Изтриване на всички артикули
✓ 107_items_Get_Item	Информация за артикул
✓ 107_items_Get_FirstFoundItem	Информация за първия намерен програмиран артикул
✓ 107_items_Get_LastFoundItem	Информация за последния намерен програмиран артикул
✓ 107_items_Get_NextItem	Информация за следващия намерен програмиран артикул
✓ 107_items_Get_FirstSoldItem	Информация за първия намерен програмиран артикул с продажби
✓ 107_items_Get_Get_LastSoldItem	Информация за последния намерен програмиран артикул с продажби
✓ 107_items_Get_Get_NextSoldItem	Информация за следващия намерен програмиран артикул с продажби
✓ 107_items_Get_FirstNotProgrammedItem	Информация за първия намерен непрограмиран артикул
✓ 107_items_Get_LastNotProgrammedItem	Информация за последния намерен непрограмиран артикул

## Команди за конфигуриране

### Настройки на устройството

Команди към устройства в група „С“, налични от версия 3.0.0.7	
✓ 061_config_Set_DateTime:	Задаване на дата и час
✓ 066_config_Set_InvoiceRange:	Задаване на интервал номера за фактура
✓ 066_config_Set_InvoiceRange_01:	Разширяване на интервал номера за фактура
✓ 101_config_Set_OperatorPassword:	Промяна на парола на оператор
✓ 255_config_Set_FpComBaudRate	Скорост на комуникация с ПС
✓ 255_config_Set_AutoPaperCutting	Разрешаване/Забрана на авт.отрязване на хартията
✓ 255_config_Set_PaperCuttingType	Разрешаване/Забрана на частично отрязване на хартията
✓ 255_config_Set_BarCodeHeight	Височина на баркода
✓ 255_config_Set_BarcodeName	Разрешаване/Забрана на печат на данни към баркода
✓ 255_config_Set_ComPortBaudRate	Скорост на комуникация със свързани периферни устройства
✓ 255_config_Set_ComPortProtocol	Протокол за комуникация с периферни устройства
✓ 255_config_Set_MainInterfaceType	Комуникационен интерфейс след включване
✓ 255_config_Set_TimeOutBeforePrintFlush	Времеви интервал за буфериране на командите
✓ 255_config_Set_WorkBatteryIncluded	Ф.принтер работи с батерия на основното захранване
✓ 255_config_Set_Dec2xLineSpacing	Намаляване на разстоянието между текстовите линии
✓ 255_config_Set_PrintFontType	Font type
✓ 255_config_Set_FooterEmptyLines	Брой на празните линии до отрязване на хартията
✓ 255_config_Set_HeaderMinLines	Минимален брой линии след футър и преди хедър
✓ 255_config_Set_LogoPrintAfterFooter	Разрешаване/Забрана на печат на лого след отпечатване на футър
✓ 255_config_Set_EnableNearPaperEnd	Разрешаване/Забрана на следене на датчик за близък край на хартията
✓ 255_config_Set_DateFromNAPServDisable	Разрешаване/Забрана на синхронизация на времето със сървъра на НАП
✓ 255_config_Set_AutoPowerOff	Минути до авт.изкл. на касовия апарат при неупотреба
✓ 255_config_Set_BkLight_AutoOff	Минути до авт.изкл. на подсветката на дисплея при неупотреба

### Платежен терминал (PinPad)

Команди към устройства в група „С“, налични от версия 3.0.0.7	
✓ 255_config_Set_PinpadComPort	Номер на COM порт за комуникация с платежен терминал
✓ 255_config_Set_PinpadComBaudRate	Скорост на комуникация с платежен терминал
✓ 255_config_Set_PinpadType	Тип на платежния терминал
✓ 255_config_Set_PinpadConnectionType	Тип на връзка между фис. устр. и сървъра на платежната институция

✓ 255_config_Set_PinpadReceiptCopies	Брой копия ва бележките от платежния терминал
✓ 255_config_Set_PinpadReceiptInfo	Тип и място на бележката от платежния терминал
✓ 255_config_Set_PinpadPaymentMenu	Тип на меню - платежен терминал
✓ 255_config_Set_PinpadLoyaltyPayment	Тип на плащане - платежен терминал

## Лого и щампи

### Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 127_config_Set_StampName:	Задаване на име на щампа, заредена с команда 203
✓ 202_config_Start_LogoLoading:	Старта на процеса по зареждане на лого
✓ 202_config_Logo>Loading:	Зареждане на данни от файл
✓ 202_config_Stop_LogoLoading:	Край на процеса по зареждане на лого
✓ 203_config_Start_StampLoading:	Старта на процеса по зареждане на щампа
✓ 203_config_Stamp>Loading:	Зареждане на данни от файл
✓ 203_config_Stop_StampLoading:	Край на процеса по зареждане на щампа

## Bluetooth

### Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 255_config_Set_BthEnable	Разрешаване/Забрана на bluetooth модула
✓ 255_config_Set_BthDiscoverability	Разрешаване/Забрана на видимост на устройството по bluetooth
✓ 255_config_Set_BthPairing	Тип на пещринга
✓ 255_config_Set_BthPinCode	Пинкод за първане по bluetooth
✓ 255_config_Set_BthVersion	Фирмуерна версия на bluetooth модула
✓ 255_config_Set_BthAddress	Адрес bluetooth модула

## Параметри (само касови апарати)

### Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 255_config_Set_EcrLogNumber	Логически номер на касовия апарат
✓ 255_config_Set_EcrExtendedReceipt	Разрешаване/Забрана на разширен тип на бележката
✓ 255_config_Set_EcrDoveriteli	Разрешаване/Забрана на работа с доверител
✓ 255_config_Set_EcrWithoutPasswords	Разрешаване/Забрана на парола
✓ 255_config_Set_EcrAskForPassword	Разрешаване/Забрана на изискване на парола за всяка бележка
✓ 255_config_Set_EcrAskForVoidPassword	Разрешаване/Забрана на изискване на парола при операция void
✓ 255_config_Set_EcrConnectedOperReport	Разрешаване/Забрана на авт. отчет по оператори при Z-отчет
✓ 255_config_Set_EcrConnectedDeptReport	Разрешаване/Забрана на авт. отчет по департаменти при Z-отчет
✓ 255_config_Set_EcrConnectedPluSalesReport	Разрешаване/Забрана на авт. отчет по артикули при Z-отчет
✓ 255_config_Set_EcrConnectedGroupsReport	Разрешаване/Забрана на авт. отчет по стокови групи при Z-отчет
✓ 255_config_Set_EcrConnectedCashReport	Разрешаване/Забрана на авт. касов отчет при Z-отчет
✓ 255_config_Set_EcrUserPeriodReports	Разрешаване/Забрана на периодичен отчет
✓ 255_config_Set_EcrPluDailyClearing	Авт. нулиране на натр. продажби артикули при Z-отчет
✓ 255_config_Set_EcrSafeOpening	Разрешаване/Забрана на авт. отваряне на чекмедже при команда Тотал
✓ 255_config_Set_EcrScaleBarMask	Баркод маска
✓ 255_config_Set_EcrNumberBarcode	Брой баркодове за програмиран артикул
✓ 255_config_Set_RegModeOnIdle	Време за изчистване на дисплея след приключване на бележката
✓ 255_config_Set_FlushAtEndOnly	Отпечатване на бележката след последното плащане
✓ 255_config_Set_EcrMidnightWarning	Минути преди полунощ за показване на предупреждение за Z-отчет
✓ 255_config_Set_EcrMandatorySubtotal	Задължително натискане на клавиш субтотал
✓ 255_config_Set_Seller	Име на продавач
✓ 255_config_Set_AutoMonthReport	Разрешаване/Забрана на предложение за месечен отчет
✓ 255_config_Set_EcrUnsentWarning	Брой часове за начало на предупреждение за неизпратени документи

## Валута, мерни единици, хедър и футър

### Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 255_config_Set_CurrNameLocal	Име на локална валута
✓ 255_config_Set_CurrNameForeign	Име на валута
✓ 255_config_Set_ExchangeRate	Валутен курс
✓ 255_config_Set_Unit_name	Име на мерна единица
✓ 255_config_Set_Header	Хедър на бележката
✓ 255_config_Set_Footer	Футър на бележката

## Оператори, Програмируемо плащане

### Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 255_config_Set_OperName	Име на оператор
✓ 255_config_Set_OperPasw	Парола на оператор
✓ 255_config_Set_PayName	Име на програмируемо плащане
✓ 255_config_Set_Payment_forbidden	Разрешаване/Забрана на програмируемо плащане

## Управление на бързи клавиши

### Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 255_config_Set_DPxx_PluCode	Обвързване на ПЛУ код с клавиш за бързо набиране
✓ 255_config_Set_KeyNDB_value	Програмиране на надбавка - стойност
✓ 255_config_Set_KeyNDB_percentage	Програмиране на надбавка - процент
✓ 255_config_Set_KeyOTS_value	Програмиране на отстъпка - стойност
✓ 255_config_Set_KeyOTS_percentage	Програмиране на отстъпка - процент
✓ 255_config_Set_KeyOTS_value	Разрешаване/Забрана на клавиша за надбавка
✓ 255_config_Set_KeyOTS_percentage	Разрешаване/Забрана на клавиша за отстъпка

## Сервизни

### Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 255_config_Set_ServPasw	Парола на сервизен техник (изисква сервизен джъмпер)
✓ 255_config_Set_ServMessage	Сервизно съобщение при изтичане на сервизен договор
✓ 255_config_Set_ServiceDate	Дата за сервизно обслужване
✓ 255_config_Set_TAXlabel	Задаване на етикет(текстово име) на ЕИК

## Параметри на бележката

### Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 255_config_Set_PrnQuality	Настройка на контраст на печата
✓ 255_config_Set_PrintColumns	Настройка на брой печатаеми символи на ред
✓ 255_config_Set_EmptyLineAfterTotal	Разрешаване/Забрана на отпечатване на празна линия след Тотал
✓ 255_config_Set_DblHeigh_totalinreg	Разрешаване/Забрана на отпечатване на Тотал реда с двойна ширина
✓ 255_config_Set_Bold_payments	Разрешаване/Забрана на отпечатване на печат с болд при продажба
✓ 255_config_Set_DublReceipts	Разрешаване/Забрана на отпечатване на дубликат
✓ 255_config_Set_IntUseReceipts	Брой на вътрешните бележки
✓ 255_config_Set_BarcodePrint	Разрешаване/Забрана на отпечатване на ПЛУ баркод в бележката
✓ 255_config_Set_LogoPrint	Разрешаване/Забрана на отпечатване на лого
✓ 255_config_Set_DoveriteliPrint	Отпечатване на името на департамента в началото на бележката
✓ 255_config_Set_ForeignPrint	Разрешаване/Забрана на отпечатване на валута в бележката
✓ 255_config_Set_VatPrintEnable	Разрешаване/Забрана на отпечатване на дан.ставки в бележката
✓ 255_config_Set_CondensedPrint	Разрешаване/Забрана на отпечатване на кондензиран печат

## Управление на бутони/клавиатура до дисплея (само за фискални принтери)

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_config_Set_DsblKeyZreport	Разрешаване/Забрана на отпечатване на Z-отчет от клавиатурата
✓	255_config_Set_DsblKeyXreport	Разрешаване/Забрана на отпечатване на X-отчет от клавиатурата
✓	255_config_Set_DsblKeyDiagnostics	Разрешаване/Забрана на отпечатване на диагностична информация от клавиатурата
✓	255_config_Set_DsblKeyFmReports	Разрешаване/Забрана на отпечатване на отчети на фиск. памет от клавиатурата
✓	255_config_Set_DsblKeyJournal	Разрешаване/Забрана на отпечатване на меню електронен журнал от клавиатурата
✓	255_config_Set_DsblKeyDateTime	Разрешаване/Забрана на смяна на дата и час от клавиатурата
✓	255_config_Set_DsblKeyCloseReceipt	Разрешаване/Забрана на ръчно затваряне на бележка от клавиатурата
✓	255_config_Set_DsblKeyCancelReceipt	Разрешаване/Забрана на ръчно анулиране на бележка от клавиатурата

## Модем и LAN параметри

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_config_Set_ModemModel	Настройка на модел на модема
✓	255_config_Set_SimPin	Задаване на ПИН код на СИМ картата
✓	255_config_Set_LanMAC	Задаване на МАК адрес
✓	255_config_Set_DHCPenable	Разрешаване/Забрана на DHCP
✓	255_config_Set_LAN_IP	Задаване на IP адрес (при забранен DHCP)
✓	255_config_Set_LAN_NetMask	Задаване на NetMask адрес (при забранен DHCP)
✓	255_config_Set_LAN_Gateway	Задаване на Gateway адрес (при забранен DHCP)
✓	255_config_Set_LAN_PriDNS	Задаване на Primary DNS адрес (при забранен DHCP)
✓	255_config_Set_LAN_SecDNS	Задаване на Secondary DNS адрес (при забранен DHCP)
✓	255_config_Set_LANport_fpCommands	Задаване на номер на порт за връзка по LAN

## ЕИК, Стокови групи и департамент

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_config_Set_ItemGroups_name	Задаване на име на стокова група
✓	255_config_Set_Dept_price	Задаване на програмирана цена, свързана с департамент
✓	255_config_Set_Dept_name	Задаване на име на департамент

## Само за модел DP-05C

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_config_Set_DHL_Algo	Разрешаване/Забрана на проверка на товарителница с алгоритъм за проверка на DHL
✓	255_config_Set_EIK_validation	Разрешаване/Забрана на проверка на валидност на ЕИК
✓	255_config_Set_EGN_validation	Разрешаване/Забрана на проверка на валидност на ЕГН
✓	255_config_Set_Bonuses	Описание на бонуси
✓	255_config_Set_TextReducedVAT	Текстово описание на причината за редуцирана дан.ставка



## Управление на данни за клиенти (само за касови апарати)

Команди към устройства в група „С“, налични от версия 3.0.0.7

✓ 140_clients_Set_ClientData:	Задаване на данни за клиента
✓ 140_clients_Del_ClientData:	Изтриване на данни за клиента по номер
✓ 140_clients_Del_AllClientData:	Изтриване на данните за всички клиенти
✓ 140_clients_Get_ClientData:	Получаване на данни за клиента по номер
✓ 140_clients_Get_FirstProgrammedClient:	Получаване на данни за първия програмиран клиент
✓ 140_clients_Get_LastProgrammedClient:	Получаване на данни за последния програмиран клиент
✓ 140_clients_Get_NextProgrammedClient:	Получаване на данни за следващия програмиран клиент
✓ 140_clients_Get_ClientByTaxNumber:	Получаване на данни за клиента по данъчен номер
✓ 140_clients_Get_FirstNotProgrammed:	Получаване на данни за първия непрограмиран клиент
✓ 140_clients_Get_LastNotProgrammed:	Получаване на данни за първия непрограмиран клиент

## Получаване на различни видове информация

### Различни видове обща информация

Команди към устройства в група „С“, налични от версия 3.0.0.7	
✓ 050_info_Get_TaxRatesByPeriod:	Получаване на активните данъчни ставки
✓ 062_info_Get_DateTime:	Получаване на текущи дата и час на фискалното устройство
✓ 062_info_Get_DateTime_01:	Получаване на текущи дата и час на фиск.устройство разбити по параметри
✓ 064_info_Get_LastFiscRecord:	Информация за последния фискален запис
✓ 065_info_Get_AdditionalDailyInfo:	Суми по данъчни групи от последното приключване на деня до момента
✓ 066_info_Get_InvoiceRange:	Получаване на стойностите на интервала номера за фактура
✓ 068_info_Get_FreeFMRecords:	Получаване на броя на свободните полета във фискалната памет
✓ 070_info_Get_CashIn_CashOut:	Получаване на информация относно текущата наличност
✓ 071_info_Get_Print_Diagnostic_0:	Печат на диагностична информация
✓ 071_info_Get_Print_Diagnostic_X:	Печат на други типове диагностична информация
✓ 071_info_Get_TaxTerminalInfo:	Получаване на общи данни за връзката със сървър на НАП
✓ 074_info_Get_StatusBytes:	Получаване на статус байтовете
✓ 076_info_Get_FTransactionStatus:	Получаване на статус на фискалната транзакция
✓ 086_info_Get_FMRecord_LastDateTime	Получаване на дата и час на последния фискален запис
✓ 087_info_Get_ItemGroup:	Получаване на информация за стокова група
✓ 088_info_Get_DepartmentInfo:	Получаване на информация за департамент
✓ 090_info_Get_DiagnosticInfo:	Получаване на обща информация за фискалното устройство
✓ 090_info_Get_EIKValue:	Получаване на стойността на ЕИК
✓ 100_info_Get_ErrorDescription:	Текстово описание за дадена код грешка от страна на фиск. устройство
✓ 103_info_Get_CurrentReceiptInfo:	Получаване на информация за вида и статуса на транзакцията
✓ 110_info_Get_AdditionalDailyInfo_C00:	Разпределение на сумата за деня по начин на плащане - продажби
✓ 110_info_Get_AdditionalDailyInfo_C01:	Разпределение на сумата за деня по начин на плащане - сторно
✓ 110_info_Get_AdditionalDailyInfo_C02:	Допълнителна информация за дневните продажби
✓ 110_info_Get_AdditionalDailyInfo_C03:	Допълнителна информация за дневните продажби - отстъпки и надбавки
✓ 110_info_Get_AdditionalDailyInfo_C04:	Допълнителна информация за дневните продажби - вайд и анулиране
✓ 110_info_Get_AdditionalDailyInfo_C05:	Допълнителна информация за дневните продажби - кеш операции (In/Out)
✓ 112_info_Get_OperatorsData:	Информация за дневните продажби и корекции за даден оператор
✓ 123_info_Get_DeviceInfo_01:	Информация за фискалното устройство
✓ 123_info_Get_DeviceInfo_02:	Информация за батерия и статус на мрежата
✓ 123_info_Get_DeviceInfo_03:	Информация за последен фискален бон и Z отчет
✓ 123_info_Get_DeviceInfo_04:	Пълна верификация на електронния журнал
✓ 123_info_Get_DeviceInfo_05:	Информация за нива на батерия и зареждане
✓ 135_info_Get_Modem_IMEI:	Получаване на IMEI на модема
✓ 135_info_Get_Modem_IMSI:	Получаване на IMSI на СИМ картата
✓ 135_info_Get_ModemStatus:	Получаване на данни и статус на модема/връзката
✓ 140_info_Get_ClientsInfo:	Обща информация за база данни клиенти

### Настройки на устройството

Команди към устройства в група „С“, налични от версия 3.0.0.7	
✓ 255_info_Get_FpComBaudRate	Скорост на комуникация с ПС
✓ 255_info_Get_AutoPaperCutting	Разрешаване/Забрана на авт.отрязване на хартията (само при FP-700X)
✓ 255_info_Get_PaperCuttingType	Разрешаване/Забрана на частично отрязване на хартията (само при FP-700X)
✓ 255_info_Get_BarCodeHeight	Височина на баркода
✓ 255_info_Get_BarcodeName	Разрешаване/Забрана на печат на данни към баркода
✓ 255_info_Get_ComPortBaudRate	Скорост на комуникация със свързани периферни устройства
✓ 255_info_Get_ComPortProtocol	Протокол за комуникация с периферни устройства
✓ 255_info_Get_MainInterfaceType	Комуникационен интерфейс след включване
✓ 255_info_Get_TimeOutBeforePrintFlush	Времеви интервал за буфериране на командите
✓ 255_info_Get_WorkBatteryIncluded	Ф.принтер работи с батерия на основното захранване
✓ 255_info_Get_Dec2xLineSpacing	Намаляване на разстоянието между текстовите линии
✓ 255_info_Get_PrintFontType	Font type
✓ 255_info_Get_FooterEmptyLines	Брой на празните линии до отрязване на хартията
✓ 255_info_Get_HeaderMinLines	Минимален брой линии след отпечатване на футър и преди отпечатване на хедър
✓ 255_info_Get_LogoPrintAfterFooter	Разрешаване/Забрана на печат на лого след отпечатване на футър
✓ 255_info_Get_EnableNearPaperEnd	Разрешаване/Забрана на следене на датчик за близък край на хартията
✓ 255_info_Get_DateFromNAPServDisable	Разрешаване/Забрана на синхронизация на времето със сървър на НАП
✓ 255_info_Get_AutoPowerOff	Минути до авт.изкл. на касовия апарат при неупотреба
✓ 255_info_Get_BkLight_AutoOff	Минути до авт.изкл. на подсветката на дисплея при неупотреба

## Платежен терминал (PinPad)

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_PinpadComPort	Номер на COM порт за комуникация с платежен терминал
✓	255_info_Get_PinpadComBaudRate	Скорост на комуникация с платежен терминал
✓	255_info_Get_PinpadType	Тип на платежния терминал
✓	255_info_Get_PinpadConnectionType	Тип на връзката между фискалното устройство и сървъра на платежната институция
✓	255_info_Get_PinpadReceiptCopies	Брой копия на бележките от платежния терминал
✓	255_info_Get_PinpadReceiptInfo	Тип и място на бележката от платежния терминал
✓	255_info_Get_PinpadPaymentMenu	Тип на меню - платежен терминал
✓	255_info_Get_PinpadLoyaltyPayment	Тип на плащане - платежен терминал

## Bluetooth

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_BthEnable	Разрешаване/Забрана на bluetooth модула
✓	255_info_Get_BthDiscoverability	Разрешаване/Забрана на видимост на устройството по bluetooth
✓	255_info_Get_BthPairing	Тип на пееинга
✓	255_info_Get_BthPinCode	Пинкод за първане по bluetooth
✓	255_info_Get_BthVersion	Фирмуерна версия на bluetooth модула
✓	255_info_Get_BthAddress	Адрес bluetooth модула

## Параметри (само касови апарати)

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_EcrLogNumber	Логически номер на касовия апарат
✓	255_info_Get_EcrExtendedReceipt	Разрешаване/Забрана на разширен тип на бележката
✓	255_info_Get_EcrDoveriteli	Разрешаване/Забрана на работа с доверител
✓	255_info_Get_EcrWithoutPasswords	Разрешаване/Забрана на парола
✓	255_info_Get_EcrAskForPassword	Разрешаване/Забрана на изискване на парола за всяка бележка
✓	255_info_Get_EcrAskForVoidPassword	Разрешаване/Забрана на изискване на парола при операция void
✓	255_info_Get_EcrConnectedOperReport	Разрешаване/Забрана на авт. отчет по оператори при Z-отчет
✓	255_info_Get_EcrConnectedDeptReport	Разрешаване/Забрана на авт. отчет по департаменти при Z-отчет
✓	255_info_Get_EcrConnectedPluSalesReport	Разрешаване/Забрана на авт. отчет по артикули при Z-отчет
✓	255_info_Get_EcrConnectedGroupsReport	Разрешаване/Забрана на авт. отчет по стокони групи при Z-отчет
✓	255_info_Get_EcrConnectedCashReport	Разрешаване/Забрана на авт. касов отчет при Z-отчет
✓	255_info_Get_EcrUserPeriodReports	Разрешаване/Забрана на периодичен отчет
✓	255_info_Get_EcrPluDailyClearing	Авт. нулиране на натр. продажи артикули при Z-отчет
✓	255_info_Get_EcrSafeOpening	Разрешаване/Забрана на авт. отваряне на чекмедже при команда Тотал
✓	255_info_Get_EcrScaleBarMask	Баркод маска
✓	255_info_Get_EcrNumberBarcode	Брой баркодове за програмиран артикул
✓	255_info_Get_RegModeOnIdle	Време за изчистване на дисплея след приключване на бележката
✓	255_info_Get_FlushAtEndOnly	Отпечатване на бележката след последното плащане
✓	255_info_Get_EcrMidnightWarning	Минути преди полунощ за показване на предупреждение за Z-отчет
✓	255_info_Get_EcrMandatorySubtotal	Задължително натискане на клавиш субтотал
✓	255_info_Get_Seller	Име на продавач
✓	255_info_Get_AutoMonthReport	Разрешаване/Забрана на предложение за месечен отчет
✓	255_info_Get_EcrUnsentWarning	Брой часове за начало на предупреждение за неизпратени документи

## Валута, мерни единици, хедър и футър

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_CurrNameLocal	Име на локална валута
✓	255_info_Get_CurrNameForeign	Име на валута
✓	255_info_Get_ExchangeRate	Валутен курс
✓	255_info_Get_Unit_name	Име на мерна единица
✓	255_info_Get_Header	Хедър на бележката
✓	255_info_Get_Footer	Футър на бележката

## Оператори, Програмируемо плащане

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_OperName	Име на оператор
✓	255_info_Get_OperPasw	Парола на оператор
✓	255_info_Get_PayName	Име на програмируемо плащане
✓	255_info_Get_Payment_forbidden	Разрешаване/Забрана на програмируемо плащане

## Управление на бързи клавиши

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_DPxx_PluCode	Обвързване на ПЛУ код с клавиш за бързо набиране
✓	255_info_Get_KeyNDB_value	Програмиране на надбавка - стойност
✓	255_info_Get_KeyNDB_percentage	Програмиране на надбавка - процент
✓	255_info_Get_KeyOTS_value	Програмиране на отстъпка - стойност
✓	255_info_Get_KeyOTS_percentage	Програмиране на отстъпка - процент
✓	255_info_Get_KeyOTS_value	Разрешаване/Забрана на клавиша за надбавка
✓	255_info_Get_KeyOTS_percentage	Разрешаване/Забрана на клавиша за отстъпка

## Сервизни

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_ServPasw	Парола на сервизен техник (изисква сервизен джъмпер)
✓	255_info_Get_ServMessage	Сервизно съобщение при изтичане на сервизен договор
✓	255_info_Get_ServiceDate	Дата за сервизно обслужване

## Параметри на бележката

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_PrnQuality	Настройка на контраст на печата
✓	255_info_Get_PrintColumns	Настройка на брой печатаеми символи на ред
✓	255_info_Get_EmptyLineAfterTotal	Разрешаване/Забрана на отпечатване на празна линия след Тотал
✓	255_info_Get_DbIHeigh_totalinreg	Разрешаване/Забрана на отпечатване на Тотал реда с двойна ширина
✓	255_info_Get_Bold_payments	Разрешаване/Забрана на отпечатване на печат с бولد при продажба
✓	255_info_Get_DublReceipts	Разрешаване/Забрана на отпечатване на дубликат
✓	255_info_Get_IntUseReceipts	Брой на вътрешните бележки
✓	255_info_Get_BarcodePrint	Разрешаване/Забрана на отпечатване на ПЛУ баркод в бележката
✓	255_info_Get_LogoPrint	Разрешаване/Забрана на отпечатване на лого
✓	255_info_Get_DoveriteliPrint	Отпечатване или не на името на департамента в началото на бележката
✓	255_info_Get_ForeignPrint	Разрешаване/Забрана на отпечатване на валута в бележката
✓	255_info_Get_VatPrintEnable	Разрешаване/Забрана на отпечатване на дан.ставки в бележката
✓	255_info_Get_CondensedPrint	Разрешаване/Забрана на отпечатване на кондензиран печат

## Управление на бутони/клавиатура до дисплея (само за фискални принтери)

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_DsblKeyZreport	Разрешаване/Забрана на отпечатване на Z-отчет от клавиатурата
✓	255_info_Get_DsblKeyXreport	Разрешаване/Забрана на отпечатване на X-отчет от клавиатурата
✓	255_info_Get_DsblKeyDiagnostics	Разрешаване/Забрана на отпечатване на диагностична информация от клавиатурата
✓	255_info_Get_DsblKeyFmReports	Разрешаване/Забрана на отпечатване на отчети на фиск. памет от клавиатурата
✓	255_info_Get_DsblKeyJournal	Разрешаване/Забрана на отпечатване на меню електронен журнал от клавиатурата
✓	255_info_Get_DsblKeyDateTime	Разрешаване/Забрана на смяна на дата и час от клавиатурата
✓	255_info_Get_DsblKeyCloseReceipt	Разрешаване/Забрана на ръчно затваряне на бележка от клавиатурата
✓	255_info_Get_DsblKeyCancelReceipt	Разрешаване/Забрана на ръчно анулиране на бележка от клавиатурата

## Модем и LAN параметри

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_ModemModel	Настройка на модел на модема
✓	255_info_Get_SimPin	Задаване на ПИН код на СИМ картата
✓	255_info_Get_LanMAC	Задаване на МАК адрес
✓	255_info_Get_DHCPenable	Разрешаване/Забрана на DHCP
✓	255_info_Get_LAN_IP	IP адрес (при забранен DHCP)
✓	255_info_Get_LAN_NetMask	NetMask адрес (при забранен DHCP)
✓	255_info_Get_LAN_Gateway	Gateway адрес (при забранен DHCP)
✓	255_info_Get_LAN_PriDNS	Primary DNS адрес (при забранен DHCP)
✓	255_info_Get_LAN_SecDNS	Secondary DNS адрес (при забранен DHCP)
✓	255_info_Get_LANport_fpCommands	Номер на порт за връзка по LAN

## ЕИК, Стокови групи и департамент

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_ItemGroups_name	Задаване на име на стокова група
✓	255_info_Get_Dept_price	Задаване на прагмирана цена, свързана с департамент
✓	255_info_Get_Dept_name	Задаване на име на департамент

## Само за модел DP-05C

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_DHL_Algo	Разрешаване/Забрана на проверка на товарителница с алгоритъм за проверка на DHL
✓	255_info_Get_EIK_validation	Разрешаване/Забрана на проверка на валидност на ЕИК
✓	255_info_Get_EGN_validation	Разрешаване/Забрана на проверка на валидност на ЕГН
✓	255_info_Get_Bonuses	Описание на бонуси
✓	255_info_Get_TextReducedVAT	Текстово описание на причината за редуцирана дан.ставка

## Информация за/от фискалната памет

Команди към устройства в група „С“, налични от версия 3.0.0.7		
✓	255_info_Get_TAXlabel	Задаване на етикет(текстово име) на ЕИК
✓	255_info_Get_nZreport	Номер на текущия Z-отчет
✓	255_info_Get_nReset	Брой RAM нулирания
✓	255_info_Get_nVatChanges	Брой промени - данъчни ставки
✓	255_info_Get_nIDnumberChanges	Брой промени - сериен номер
✓	255_info_Get_nFMnumberChanges	Брой промени - номер на фискална памет
✓	255_info_Get_nTAXnumberChanges	Брой промени - данъчен номер
✓	255_info_Get_valVat	Текущи стойности - данъчни ставки
✓	255_info_Get_FMDDeviceID	ID - фискална памет
✓	255_info_Get_IDnumber	Сериен номер на устройството
✓	255_info_Get_FMnumber	Номер на фискална памет
✓	255_info_Get_TAXnumber	Данъчен номер
✓	255_info_Get_FmWriteDateTime	Дата и час на запис на блок във фискалната памет
✓	255_info_Get_LastValiddate	Последна валидна дата

## Вътрешни променливи

Команди към устройства в група „С“, налични от версия 3.0.0.7	
✓ 255_info_Get_UNP	Последен уникален номер на продажба
✓ 255_info_Get_StornoUNP	Последен УНП номер на сторниран документ
✓ 255_info_Get_Fiscalized	Устройството е фискализирано или не
✓ 255_info_Get_DFR_needed	Показва дали е издадена фискална бележка след последния Z-отчет
✓ 255_info_Get_DecimalPoint	Брой символи след десетичната запетая
✓ 255_info_Get_nBon	Глобален боряч на бележките (използва се за УНП)
✓ 255_info_Get_nFBon	Глобален боряч на фискалните бележки
✓ 255_info_Get_nInvoice	Текущ номер на фактура издадена от фискалното устройство
✓ 255_info_Get_InvoiceRangeBeg	Начало на интервала за издаване на фактури
✓ 255_info_Get_InvoiceRangeEnd	Край на интервала за издаване на фактури
✓ 255_info_Get_nFBonDailyCount	Брой на фискалните бележки в рамките на деня
✓ 255_info_Get_nLastFiscalDoc	Номер на последната фискална бележка
✓ 255_info_Get_CurrClerk	Номер на текущия оператор
✓ 255_info_Get_EJNewJurnal	Нов електронен журнал
✓ 255_info_Get_EJNumber	Номер на текущия електронен журнал
✓ 255_info_Get_DateLastSucceededSent	Дата и час на последна връзка със сървъра
✓ 255_info_Get_NapRegistered	Устройството е регистрирано в сървъра на НАП
✓ 255_info_Get_DeregOnSever	Устройството е deregистрирано в сървъра на НАП

## Работа с КЛЕН

Команди към устройства в група „С“, налични от версия 3.0.0.7	
✓ 124_klen_Documents_InRange:	Информация за търсене на документи в даден времеви интервал
✓ 125_klen_SetDocument_ToRead:	Посочване на документ за изтегляне от КЛЕН
✓ 125_klen_GetLine_AsText:	Прочитане на линия от документ в клена като текст
✓ 125_klen_GetLine_AsData:	Прочитане на линия от документ в клена като Base64 данни
✓ 125_klen_Print_Document:	Отпечатване на посочения документ
✓ 125_klen_SetCSVDocument_ToRead:	Посочване на документ за получаване на структурирана информация
✓ 125_klen_Get_CSVDData_AsText:	Получаване на структурирана информация за документа

## Сервизни и други

Команди към устройства в група „С“, налични от версия 3.0.0.7	
✓ 083_service_Set_DecimalAndTaxRates:	Програмиране на дан.ставки и десетична точка
✓ 089_service_Test_FiscalMemory:	За вътешна употреба
✓ 091_service_Set_SerialNumber:	За вътешна употреба
✓ 045_other_Check_Connection:	Проверка на връзката
✓ 202_other_Power_Off:	Изключване на фискалното устройство
✓ 202_other_Restart:	Рестарт на фискалното устройство

## Остарели интерфейси (Obsolete interfaces)

Описаните по-долу интерфейси се смятат за остарели от страна на екипа в Датекс и тяхното развитие е прекратено. Ако все пак решите да продължите да ги използвате – за някои от моделите ще Ви трябва консултация по отношение на сторно боновете и статус битовите които се вдигат в този случай.

Моля пишете ни за повече подробности, ако имате проблем – екипа на Датекс ще се опита да го разреши.

Obsolete interfaces
ICSFP3530
ICSFP_KENYA
ICS_BGR_DP55_KL
ICS_BGR_FP550_KL
ICS_BGR_FP1000_KL
ICS_BGR_FMP10_KL
ICS_BGR_FP60_KL
ICS_BGR_FP2000_KL
ICS_BGR_FP700_KL
ICS_BGR_SK1_31F_KL
ICS_BGR_IBM4610F
ICS_BGR_FP550_D_KL
ICS_BGR_FP705_KL
ICS_BGR_WP500_KL
ICS_BGR_FMP350_KL

## Отказ от отговорност

Информацията в този документ подлежи на промяна без предизвестие и не представлява ангажимент от страна на Датекс. Датекс не носи никаква отговорност, ако информацията в този документ е непълна или неточна.

Въпреки това, ние ще положим всички усилия да предоставяме актуална информация за продуктите, описани в този документ.



## Как да се изпращат съобщения за грешки

За да разрешим проблема – често ние трябва да симулираме същите условия като тези при Вас или Вашия клиент. Моля – изпратете ни имейл със следната информация:

- ✓ Версията на COM сървъра, както и кой интерфейс използвате;
- ✓ Информация относно Windows и системата – можете да използвате информацията върната от метода “get\_SystemInfo” или от някоя от демо програмите;
- ✓ Информация относно модела на фискалното устройство. Би било хубаво да отпечатате и сканирате диагностичната информация от самото устройство;
- ✓ Типа на транспортния протокол и ако използвате преходник от USB към RS-232 – информация за модела и драйвера;
- ✓ Ако използвате или става дума за метод “execute\_Command” – изпратете ни стойността на командата, както и стойностите на входноизходните данни, ако има такива;
- ✓ Ако използвате или става дума за “execute\_Script\_V1” – изпратете ни стойността на скрипта;
- ✓ Ако е възможно:
  - активирайте „**tracking mode**” до **true**;
  - опишете възникването на проблема стъпка по стъпка;
  - изпратете ни log файла който генерира самия COM сървър с включен „**tracking mode**”;
- ✓ Ако е възможно – запишете и ни изпратете стойностите на статус битовете след като изпълнението на командата пропадне. Не изпълнявайте други команди, а също не четете други пропъртита след неуспешното изпълнение, защото това може да доведе до препокриване или заличаване на проблема. Просто вземете стойността им и я запишете;
- ✓ Изпратете ни и стойността на последната грешка (last error code);
- ✓ Ако използвате метода “execute\_Command\_ByName” – изпратете ни цялата необходима информация (стойностите на входно/изходните параметри), защото те ще ни трябват при симулирането на същите условия.
- ✓ Всъщност винаги ни изпращайте стойностите на входно/изходните параметри;
- ✓ **Ако ни изпращате сорскод – моля не ни изпращайте сорса на целия проект! Ние не предлагаме такъв тип поддръжка!**

Поддръжката която оказваме е само чрез имейл! Моля не изисквайте какъвто и да било конферентен тип поддръжка.

Ние знаем, че проекта Ви е важен за Вас и клиентите Ви. Повярвайте ни – Вашият успех е важен и за нас. Ще се радваме да помогнем, но поради спецификата на темата, политиката на компанията е да оказва помощ на разработчиците единствено в писмена форма – най-често и предимно чрез имейл.

## Контакти

*Ако откриете грешка – изпратете ни електронно съобщение. Ние ще се радваме да отстраним проблема в следващи версии на имплементацията на интерфейс „CFD\_BGR“. Ако сте програмист – чувствайте се свободен да питате за различните аспекти и начини за употреба на COM сървъра. Ако имате препоръки или желания за доработка на интерфейс „CFD\_BGR“ ще се радваме да го обсъдим. Пишете ни – ще се опитаме да помогнем.*

*Можете да изпращате имейлите със своите съобщения и въпроси до:*

*[dobrin@datecs.bg](mailto:dobrin@datecs.bg)*