

Device group A - methods documentation  
(Doc ver. 1.0.0.1)  
(SDK ver. 1.0.0.9)

<b>1 Class Documentation</b>	<b>1</b>
1.1 FiscalPrinterSDK.FDGROUP_A_BGR Class Reference	1
1.1.1 Detailed Description	18
1.1.2 Member Function Documentation	18
1.1.2.1 config_disable_EUR()	18
1.1.2.2 config_disable_PrintSuppression()	19
1.1.2.3 config_enable_EUR()	19
1.1.2.4 config_enable_PrintSuppression()	19
1.1.2.5 config_Pinpad_DeleteBatchNumber()	20
1.1.2.6 config_Pinpad_DeleteFreezeTransaction()	20
1.1.2.7 config_Pinpad_HeaderPrintingOptions() [1/2]	20
1.1.2.8 config_Pinpad_HeaderPrintingOptions() [2/2]	21
1.1.2.9 config_Pinpad_Set_DateTime()	21
1.1.2.10 config_Pinpad_SetBatchNumber()	22
1.1.2.11 config_Pinpad_SetReceiptNumber()	22
1.1.2.12 config_Pinpad_SetRRNFromPPTtoFD()	23
1.1.2.13 config_Restore_PrintSuppression()	23
1.1.2.14 config_SafetyTurnOffDevice()	23
1.1.2.15 config_Set_2GForcedModem()	23
1.1.2.16 config_Set_AdditionalPaymentName()	24
1.1.2.17 config_Set_AutoCutting()	24
1.1.2.18 config_Set_AutoFormat()	25
1.1.2.19 config_Set_AutomaticRetractPaper()	25
1.1.2.20 config_Set_AutomaticZReport()	25
1.1.2.21 config_Set_BarcodeHeight()	26
1.1.2.22 config_Set_CashDrawerPulse()	26
1.1.2.23 config_Set_DateTime()	26
1.1.2.24 config_Set_DepartmentName()	27
1.1.2.25 config_Set_DepartmentNameTwoRows()	27
1.1.2.26 config_Set_ExtraSpace()	27
1.1.2.27 config_Set_FeedButton()	28
1.1.2.28 config_Set_FontSize()	28
1.1.2.29 config_Set_FooterLine()	28
1.1.2.30 config_Set_HeaderLine()	29
1.1.2.31 config_Set_InvoiceRange()	29
1.1.2.32 config_Set_LANSignal()	29
1.1.2.33 config_Set_Logo()	30
1.1.2.34 config_Set_OperatorName()	30
1.1.2.35 config_Set_OperatorPassword()	30
1.1.2.36 config_Set_OptionalErrorText()	31
1.1.2.37 config_Set_PrintDensity()	31
1.1.2.38 config_Set_PrintDepartment()	31

1.1.2.39 config_Set_PrinterReadinessPrintMsgs()	33
1.1.2.40 config_Set_PrintLogo()	33
1.1.2.41 config_Set_PrintTaxDDS()	33
1.1.2.42 config_Set_RollUpPaper()	34
1.1.2.43 config_Set_SIM()	34
1.1.2.44 config_Set_Switches()	34
1.1.2.45 config_Set_UNPPrinting()	35
1.1.2.46 config_Set_WarningTimeRange()	35
1.1.2.47 config_Set_WorkingWithPinpad()	35
1.1.2.48 config_SetLAN()	36
1.1.2.49 display_Clear()	36
1.1.2.50 display_Show_DateTime()	37
1.1.2.51 display_Show_LowerLine()	37
1.1.2.52 display_Show_Text()	37
1.1.2.53 Get_SBit_State()	37
1.1.2.54 Get_sBytesBitInUse()	38
1.1.2.55 info_ClearLastErrorExtendedInfo()	38
1.1.2.56 info_Get_ActiveTaxes()	38
1.1.2.57 info_Get_AdditionalDailyInfo_01()	39
1.1.2.58 info_Get_AdditionalDailyInfo_02()	40
1.1.2.59 info_Get_AdditionalDailyInfo_03()	40
1.1.2.60 info_Get_AdditionalPaymentNames()	41
1.1.2.61 info_Get_AdditionalPayments()	41
1.1.2.62 info_Get_CashDrawerStatusOrOpenShift()	42
1.1.2.63 info_Get_CashIn_CashOut()	42
1.1.2.64 info_Get_CashInCashOut()	43
1.1.2.65 info_Get_Corrections()	43
1.1.2.66 info_Get_CurrentRecieptInfo()	44
1.1.2.67 info_Get_DailyCorrections()	44
1.1.2.68 info_Get_DataForLastIssuedReceipt() [1/2]	45
1.1.2.69 info_Get_DataForLastIssuedReceipt() [2/2]	45
1.1.2.70 info_Get_DateTime()	45
1.1.2.71 info_Get_DateTime_01()	46
1.1.2.72 info_Get_DecimalAndTaxRates()	46
1.1.2.73 info_Get_DepartmentInfo()	46
1.1.2.74 info_Get_DiagnosticInfo()	47
1.1.2.75 info_Get_EIK()	47
1.1.2.76 info_Get_FMRecord_LastDateTime()	48
1.1.2.77 info_Get_FMRecord_LastDateTime_01()	48
1.1.2.78 info_Get_FreeFMRecords()	49
1.1.2.79 info_Get_FTransactionStatus()	49
1.1.2.80 info_Get_InvoiceRange()	49

1.1.2.81 info_Get_LANSettings()	50
1.1.2.82 info_Get_LastDocumentNumber()	50
1.1.2.83 info_Get_LastFiscRecord()	50
1.1.2.84 info_Get_LastFiscRecord_01()	51
1.1.2.85 info_Get_LogoLine()	51
1.1.2.86 info_Get_NettoSums()	52
1.1.2.87 info_Get_NotPrintedRowsCount()	52
1.1.2.88 info_Get_NRADData_Type1()	53
1.1.2.89 info_Get_NRADData_Type2()	53
1.1.2.90 info_Get_NRADData_Type3()	54
1.1.2.91 info_Get_NRADData_Type4()	54
1.1.2.92 info_Get_NRADData_Type5()	55
1.1.2.93 info_Get_NRADData_Type6()	55
1.1.2.94 info_Get_NRADData_Type7()	55
1.1.2.95 info_Get_NRADData_Type8()	56
1.1.2.96 info_Get_NRADDocuments()	56
1.1.2.97 info_Get_Numbers()	57
1.1.2.98 info_Get_OperatorsData()	57
1.1.2.99 info_Get_PrintOption()	58
1.1.2.100 info_Get_ProgrammedFuel()	58
1.1.2.101 info_Get_ProgrammedPumps()	59
1.1.2.102 info_Get_ProgrammedPumps2()	59
1.1.2.103 info_Get_ProgrammedPumps3()	60
1.1.2.104 info_Get_RAMResetDateTime()	60
1.1.2.105 info_Get_SoldQuant_AccumulatedSales()	61
1.1.2.106 info_Get_StatusBytes()	61
1.1.2.107 info_Get_Storno_NettoSums()	62
1.1.2.108 info_Get_Storno_VATsSums()	62
1.1.2.109 info_Get_StornoSums()	63
1.1.2.110 info_Get_Sums()	64
1.1.2.111 info_Get_TaxRates()	64
1.1.2.112 info_Get_TaxRatesByPeriod()	65
1.1.2.113 info_Get_TaxRateValues()	65
1.1.2.114 info_Get_Turnover()	67
1.1.2.115 info_Get_VATsSums()	68
1.1.2.116 info_Get_VoltageAndTemperature()	68
1.1.2.117 info_LastErrorExtendedInfo()	69
1.1.2.118 info_Pinpad_CheckLastRecordedRRN()	69
1.1.2.119 info_Pinpad_ClearTransactionStatusFlags()	69
1.1.2.120 info_Pinpad_DataForLastTransaction()	70
1.1.2.121 info_Pinpad_Get_DateTime()	70
1.1.2.122 info_Pinpad_GetUpdate()	70

1.1.2.123 info_Pinpad_Information()	71
1.1.2.124 info_Pinpad_NumberOfEntries()	71
1.1.2.125 info_Pinpad_PrintLastSuccessfulTransaction()	71
1.1.2.126 info_Pinpad_StatusOfLastTransaction()	72
1.1.2.127 info_Pinpad_TestConnection()	72
1.1.2.128 info_Pinpad_TestConnectionToServer()	72
1.1.2.129 info_Pinpad_TransactionStatus()	73
1.1.2.130 info_Print_Diagnostic_0()	73
1.1.2.131 info_Print_Short_FMReportByDTRange()	73
1.1.2.132 info_Print_Short_MonthlyFMReport()	73
1.1.2.133 info_Print_Short_YearlyFMReport()	74
1.1.2.134 info_Service_Contracts()	74
1.1.2.135 items_Delete_All_Items()	74
1.1.2.136 items_Delete_Item()	75
1.1.2.137 items_Delete_ItemsInRange()	75
1.1.2.138 items_Get_FirstFoundItem()	76
1.1.2.139 items_Get_FirstNotProgrammedItem()	76
1.1.2.140 items_Get_FirstSoldItem()	77
1.1.2.141 items_Get_Item()	77
1.1.2.142 items_Get_ItemsInformation()	78
1.1.2.143 items_Get_LastFoundItem()	78
1.1.2.144 items_Get_LastNotProgrammedItem()	79
1.1.2.145 items_Get_LastSoldItem()	79
1.1.2.146 items_Get_NextItem()	80
1.1.2.147 items_Get_NextSoldItem()	80
1.1.2.148 items_Set_Item()	81
1.1.2.149 items_Set_ItemQuantity()	82
1.1.2.150 items_Set_ItemWithReplace()	82
1.1.2.151 klen_Find_Document()	83
1.1.2.152 klen_Get_Info()	83
1.1.2.153 klen_Get_KLENInfo_ZReport()	84
1.1.2.154 klen_Get_NextTextRow()	84
1.1.2.155 klen_Get_ZReportInfo()	85
1.1.2.156 klen_Print_DocsFromDateToDate()	85
1.1.2.157 klen_Print_DocsInRange()	86
1.1.2.158 klen_Print_DocsInRange_InZReport()	87
1.1.2.159 klen_Print_Document()	87
1.1.2.160 klen_Print_SHA1()	88
1.1.2.161 klen_Print_Validation()	88
1.1.2.162 klen_Set_DocsRange()	88
1.1.2.163 klen_Set_DocsRange_ByDateTime()	89
1.1.2.164 klen_Set_DocsRange_InZReport()	89

1.1.2.165 klen_Set_DocsRange_InZReport_01()	90
1.1.2.166 klen_Set_DocsRange_InZReport_02()	91
1.1.2.167 klen_SInfo_ExV_ByDateRange()	91
1.1.2.168 klen_SInfo_ExV_ByNumber()	92
1.1.2.169 klen_SInfo_ExV_ByNumbersRange()	92
1.1.2.170 klen_SInfo_ExV_GetNext()	93
1.1.2.171 klen_SInfo_V_ByDateRange()	94
1.1.2.172 klen_SInfo_V_ByNumber()	94
1.1.2.173 klen_SInfo_V_ByNumbersRange()	95
1.1.2.174 klen_SInfo_V_GetNext()	96
1.1.2.175 klen_SInfo_W_ByDateRange()	96
1.1.2.176 klen_SInfo_W_ByNumber()	97
1.1.2.177 klen_SInfo_W_ByNumbersRange()	97
1.1.2.178 klen_SInfo_W_GetNext()	97
1.1.2.179 klen_SInfo_Y_ByDateRange()	99
1.1.2.180 klen_SInfo_Y_ByNumber()	99
1.1.2.181 klen_SInfo_Y_ByNumbersRange()	100
1.1.2.182 klen_SInfo_Y_GetNext()	100
1.1.2.183 open_StornoReceipt()	100
1.1.2.184 other_Sound_Signal()	101
1.1.2.185 receipt_CashIn_CashOut()	102
1.1.2.186 receipt_Current_Info()	102
1.1.2.187 receipt_DisplayDepPLU_Sale()	103
1.1.2.188 receipt_DisplayDepPLU_Sale_CByPercent()	103
1.1.2.189 receipt_DisplayDepPLU_Sale_CBySum()	103
1.1.2.190 receipt_DisplayDepPLU_SaleUn()	104
1.1.2.191 receipt_DisplayDepPLU_SaleUn_CByPercent()	104
1.1.2.192 receipt_DisplayDepPLU_SaleUn_CBySum()	105
1.1.2.193 receipt_DisplayDSale()	106
1.1.2.194 receipt_DisplayDSale_CByPercent()	106
1.1.2.195 receipt_DisplayDSale_CBySum()	106
1.1.2.196 receipt_DisplayDSale_Minimum()	107
1.1.2.197 receipt_DisplayDSale_Un()	107
1.1.2.198 receipt_DisplayDSale_UnCByPercent()	108
1.1.2.199 receipt_DisplayDSale_UnCBySum()	108
1.1.2.200 receipt_DisplayPLUSale()	109
1.1.2.201 receipt_DisplayPLUSale_CByPercent()	109
1.1.2.202 receipt_DisplayPLUSale_CBySum()	110
1.1.2.203 receipt_DisplayPLUSale_Un()	110
1.1.2.204 receipt_DisplayPLUSale_UnCByPercent()	111
1.1.2.205 receipt_DisplayPLUSale_UnCBySum()	111
1.1.2.206 receipt_DisplaySale()	112

1.1.2.207 receipt_DisplaySale_CByPercent()	112
1.1.2.208 receipt_DisplaySale_CBySum()	112
1.1.2.209 receipt_DisplaySale_Minimum()	113
1.1.2.210 receipt_DisplaySale_Un()	113
1.1.2.211 receipt_DisplaySale_UnCByPercent()	114
1.1.2.212 receipt_DisplaySale_UnCBySum()	114
1.1.2.213 receipt_Drawer_KickOut()	115
1.1.2.214 receipt_DSale()	115
1.1.2.215 receipt_DSale_CByPercent()	116
1.1.2.216 receipt_DSale_CByPercentWText()	116
1.1.2.217 receipt_DSale_CBySum()	116
1.1.2.218 receipt_DSale_CBySumWText()	117
1.1.2.219 receipt_DSale_Minimum()	117
1.1.2.220 receipt_DSale_TextRow1()	118
1.1.2.221 receipt_DSale_TextRow1_WQuan()	118
1.1.2.222 receipt_DSale_TextRow1CByPercent()	118
1.1.2.223 receipt_DSale_TextRow1CBySum()	119
1.1.2.224 receipt_DSale_TextRow1Un()	119
1.1.2.225 receipt_DSale_TextRow1UnCByPercent()	120
1.1.2.226 receipt_DSale_TextRow1UnCBySum()	120
1.1.2.227 receipt_DSale_TextRow2()	121
1.1.2.228 receipt_DSale_TextRow2CByPercent()	121
1.1.2.229 receipt_DSale_TextRow2CBySum()	122
1.1.2.230 receipt_DSale_TextRow2Un()	122
1.1.2.231 receipt_DSale_TextRow2UnCByPercent()	123
1.1.2.232 receipt_DSale_TextRow2UnCBySum()	123
1.1.2.233 receipt_DSale_Un()	124
1.1.2.234 receipt_DSale_UnCByPercent()	124
1.1.2.235 receipt_DSale_UnCByPercentWText()	125
1.1.2.236 receipt_DSale_UnCBySum()	125
1.1.2.237 receipt_DSale_UnCBySumWText()	126
1.1.2.238 receipt_DSale_UnWText()	126
1.1.2.239 receipt_Fiscal_01_Open()	127
1.1.2.240 receipt_Fiscal_02_Open()	127
1.1.2.241 receipt_Fiscal_Cancel()	128
1.1.2.242 receipt_Fiscal_Close()	128
1.1.2.243 receipt_Fiscal_Text()	128
1.1.2.244 receipt_FiscalOpen_A02()	129
1.1.2.245 receipt_FiscalOpen_A04()	129
1.1.2.246 receipt_Invoice_01_Open()	130
1.1.2.247 receipt_Invoice_Open()	130
1.1.2.248 receipt_NonFiscal_Close()	131

1.1.2.249 receipt_NonFiscal_Open()	131
1.1.2.250 receipt_NonFiscal_Text()	132
1.1.2.251 receipt_NonFiscalRotated_Close()	132
1.1.2.252 receipt_NonFiscalRotated_Open()	132
1.1.2.253 receipt_NonFiscalRotated_Text()	133
1.1.2.254 receipt_Paper_Cut()	133
1.1.2.255 receipt_Paper_Feed()	133
1.1.2.256 receipt_Paper_Feed_InnerContainer()	133
1.1.2.257 receipt_PFiscalText_01()	133
1.1.2.258 receipt_PFiscalText_02()	134
1.1.2.259 receipt_Pinpad_PrintByNumber()	134
1.1.2.260 receipt_Pinpad_PrintLastSuccessfultransaction()	135
1.1.2.261 receipt_Pinpad_PrintReceiptCopyFromEJ_ByDate()	135
1.1.2.262 receipt_Pinpad_PrintReceiptCopyFromEJ_ByNum()	135
1.1.2.263 receipt_Pinpad_ReturnLastSuccessfultransaction()	136
1.1.2.264 receipt_PLU_Sale()	136
1.1.2.265 receipt_PLUDep_Sale()	137
1.1.2.266 receipt_PLUDep_Sale_CByPercent()	137
1.1.2.267 receipt_PLUDep_Sale_CBySum()	137
1.1.2.268 receipt_PLUDep_Sale_Un()	138
1.1.2.269 receipt_PLUDep_Sale_UnCByPercent()	138
1.1.2.270 receipt_PLUDep_Sale_UnCBySum()	139
1.1.2.271 receipt_PLUSale_CByPercent()	139
1.1.2.272 receipt_PLUSale_CBySum()	140
1.1.2.273 receipt_PLUSale_Un()	140
1.1.2.274 receipt_PLUSale_UnCByPercent()	140
1.1.2.275 receipt_PLUSale_UnCBySum()	141
1.1.2.276 receipt_PNonFiscal_Text()	141
1.1.2.277 receipt_Print_Barcode_01()	142
1.1.2.278 receipt_Print_Barcode_02()	142
1.1.2.279 receipt_Print_Duplicate()	143
1.1.2.280 receipt_PrintClientInfo_01()	143
1.1.2.281 receipt_PrintClientInfo_02()	144
1.1.2.282 receipt_PrintClientInfo_03()	144
1.1.2.283 receipt_PrintClientInfo_04()	145
1.1.2.284 receipt_PrintClientInfo_05()	145
1.1.2.285 receipt_PrintClientInfo_06()	145
1.1.2.286 receipt_PrintClientInfo_07()	146
1.1.2.287 receipt_Sale()	147
1.1.2.288 receipt_Sale_CByPercent()	147
1.1.2.289 receipt_Sale_CByPercentWText()	147
1.1.2.290 receipt_Sale_CBySum()	148



1.1.2.291 receipt_Sale_CBySumWText()	148
1.1.2.292 receipt_Sale_Minimum()	149
1.1.2.293 receipt_Sale_TextRow1()	149
1.1.2.294 receipt_Sale_TextRow1_WQuan()	149
1.1.2.295 receipt_Sale_TextRow1CByPercent()	150
1.1.2.296 receipt_Sale_TextRow1CBySum()	150
1.1.2.297 receipt_Sale_TextRow1Un()	151
1.1.2.298 receipt_Sale_TextRow1UnCByPercent()	151
1.1.2.299 receipt_Sale_TextRow1UnCBySum()	152
1.1.2.300 receipt_Sale_TextRow2()	152
1.1.2.301 receipt_Sale_TextRow2CByPercent()	153
1.1.2.302 receipt_Sale_TextRow2CBySum()	153
1.1.2.303 receipt_Sale_TextRow2Un()	154
1.1.2.304 receipt_Sale_TextRow2UnCByPercent()	154
1.1.2.305 receipt_Sale_TextRow2UnCBySum()	155
1.1.2.306 receipt_Sale_Un()	155
1.1.2.307 receipt_Sale_UnCByPercent()	156
1.1.2.308 receipt_Sale_UnCByPercentWText()	156
1.1.2.309 receipt_Sale_UnCBySum()	157
1.1.2.310 receipt_Sale_UnCBySumWText()	157
1.1.2.311 receipt_Sale_UnWText()	158
1.1.2.312 receipt_Separating_Line()	158
1.1.2.313 receipt_Separating_LineW()	158
1.1.2.314 receipt_StornoOpen_A01()	159
1.1.2.315 receipt_StornoOpen_A02()	159
1.1.2.316 receipt_StornoOpen_A03()	160
1.1.2.317 receipt_StornoOpen_A04()	160
1.1.2.318 receipt_StornoOpen_A05()	162
1.1.2.319 receipt_StornoOpen_A06()	163
1.1.2.320 receipt_StornoOpen_A07()	163
1.1.2.321 receipt_StornoOpen_A08()	164
1.1.2.322 receipt_StornoOpen_A09()	165
1.1.2.323 receipt_StornoOpen_A10()	166
1.1.2.324 receipt_StornoOpen_A11()	166
1.1.2.325 receipt_StornoOpen_A12()	167
1.1.2.326 receipt_StornoOpen_A13()	168
1.1.2.327 receipt_StornoOpen_A14()	169
1.1.2.328 receipt_StornoOpen_A15()	170
1.1.2.329 receipt_StornoOpen_A16()	170
1.1.2.330 receipt_Subtotal()	171
1.1.2.331 receipt_Subtotal_CByPercent()	172
1.1.2.332 receipt_Subtotal_CBySum()	172

1.1.2.333 receipt_Total()	173
1.1.2.334 receipt_Total_PAmount()	174
1.1.2.335 receipt_Total_PAmountTextRow1()	174
1.1.2.336 receipt_Total_PAmountTextRow2()	175
1.1.2.337 receipt_Total_PAmountWithoutText()	175
1.1.2.338 receipt_Total_TextRow1()	176
1.1.2.339 receipt_Total_TextRow2()	177
1.1.2.340 receipt_Total_WithoutText()	177
1.1.2.341 report_DailyClosure_01()	178
1.1.2.342 report_DailyClosure_02()	178
1.1.2.343 report_DailyClosureByDepartments_01()	179
1.1.2.344 report_DailyClosureByDepartments_02()	179
1.1.2.345 report_DailyClosureByDepartmentsAndItems_01()	180
1.1.2.346 report_DailyClosureByDepartmentsAndItems_02()	181
1.1.2.347 report_ExtDailyClosure_01()	181
1.1.2.348 report_ExtDailyClosure_02()	182
1.1.2.349 report_FMByDateRange()	182
1.1.2.350 report_FMByDateRange_SHA1()	183
1.1.2.351 report_FMByDateRange_Short()	183
1.1.2.352 report_FMByNumRange()	183
1.1.2.353 report_FMByNumRange_SHA1()	184
1.1.2.354 report_FMByNumRange_Short()	184
1.1.2.355 report_Items()	184
1.1.2.356 report_Items_InRange()	185
1.1.2.357 report_Items_InRangeByGroup()	185
1.1.2.358 report_Operators()	186
1.1.2.359 report_Pinpad_EndOfDay()	186
1.1.2.360 report_Pinpad_Reports()	186
1.1.2.361 report_Service_Contracts()	187
1.1.2.362 service_Disable_Print()	187
1.1.2.363 service_Format_KLEN()	188
1.1.2.364 service_RAM_Reset()	188
1.1.2.365 service_Set_DecimalAndTaxRates()	188
1.1.2.366 service_Set_EIK()	189
1.1.2.367 service_Set_ProductionTestArea()	189
1.1.2.368 service_Set_SerialNumber()	190
1.1.2.369 service_Test_GPRS()	190
1.1.2.370 Set_Sbit_ErrorChecking()	191

# Chapter 1

## Class Documentation

### 1.1 FiscalPrinterSDK.FDGROUP\_A\_BGR Class Reference

Fiscal Devices Group "A" - Bulgaria

#### Public Member Functions

- bool [Get\\_SBit\\_State](#) (int byteIndex, int bitIndex)  
*Get status bit state*
- bool [Get\\_sBytesBitInUse](#) (int byteIndex, int bitIndex)  
*Gets status bits in use*
- void [Set\\_Sbit\\_ErrorChecking](#) (int byteIndex, int bitIndex, bool isError)  
*Get if desired bit is an error*
- void [display\\_Clear](#) ()  
*Clear display*
- void [display\\_Show\\_LowerLine](#) (string textData)  
*Show text on lower line of the display*
- void [display\\_Show\\_DateTime](#) ()  
*Display date and time*
- void [display\\_Show\\_Text](#) (string textData)  
*Show text on the display*
- Dictionary< string, string > [receipt\\_NonFiscal\\_Open](#) ()  
*Open non-fiscal receipt*
- Dictionary< string, string > [receipt\\_NonFiscal\\_Close](#) ()  
*Close non-fiscal receipt*
- void [receipt\\_NonFiscal\\_Text](#) (string inputText)  
*Print free non-fiscal receipt*
- void [receipt\\_PNonFiscal\\_Text](#) (string height, string flags, string inputText)
- Dictionary< string, string > [receipt\\_NonFiscalRotated\\_Open](#) ()  
*Open non-fiscal rotated text*
- void [receipt\\_NonFiscalRotated\\_Text](#) (string inputText)  
*Print 90-degree rotated text*
- Dictionary< string, string > [receipt\\_NonFiscalRotated\\_Close](#) ()  
*Close rotated non-fiscal receipt*

- void [open\\_StornoReceipt](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string uNP, string stornoType, string stornoDocumentNumber, string stornoUNP, string stornoDateTime, string stornoFMNumber, string stornoReason, ref bool isFullStorno)

*Combined variant for Strono receipt*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A01](#) (string operatorNumber, string operatorPassword, string tillNumber, string stornoType, string stornoDocumentNumber)

*Open storno document - variant 1*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A02](#) (string operatorNumber, string operatorPassword, string tillNumber, string uNP, string stornoType, string stornoDocumentNumber)

*Open storno document - variant 2*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A03](#) (string operatorNumber, string operatorPassword, string tillNumber, string stornoType, string stornoDocumentNumber, string stornoReason)

*Open storno document - variant 3*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A04](#) (string operatorNumber, string operatorPassword, string tillNumber, string uNP, string stornoType, string stornoDocumentNumber, string stornoReason)

*Open storno document - variant 4*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A05](#) (string operatorNumber, string operatorPassword, string tillNumber, string stornoType, string stornoDocumentNumber, string stornoUNP, string stornoDate←Time, string stornoFMNumber)

*Open storno document - variant 5*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A06](#) (string operatorNumber, string operatorPassword, string tillNumber, string uNP, string stornoType, string stornoDocumentNumber, string stornoUNP, string stornoDateTime, string stornoFMNumber)

*Open storno document - variant 6*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A07](#) (string operatorNumber, string operatorPassword, string tillNumber, string stornoType, string stornoDocumentNumber, string stornoUNP, string stornoDate←Time, string stornoFMNumber, string stornoReason)

*Open storno document - variant 7*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A08](#) (string operatorNumber, string operatorPassword, string tillNumber, string uNP, string stornoType, string stornoDocumentNumber, string stornoUNP, string stornoDateTime, string stornoFMNumber, string stornoReason)

*Open storno document - variant 8*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A09](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string stornoType, string stornoDocumentNumber)

*Open storno document - variant 9*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A10](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string uNP, string stornoType, string stornoDocument←Number)

*Open storno document - variant 10*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A11](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string stornoType, string stornoDocumentNumber, string stornoReason)

*Open storno document - variant 11*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A12](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string uNP, string stornoType, string stornoDocument←Number, string stornoReason)

*Open storno document - variant 12*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A13](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string stornoType, string stornoDocumentNumber, string stornoUNP, string stornoDateTime, string stornoFMNumber)

*Open storno document - variant 13*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A14](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string uNP, string stornoType, string stornoDocument←Number, string stornoUNP, string stornoDateTime, string stornoFMNumber)

*Open storno document - variant 14*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A15](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string stornoType, string stornoDocumentNumber, string stornoUNP, string stornoDateTime, string stornoFMNumber, string stornoReason)

*Open storno document - variant 15*

- Dictionary< string, string > [receipt\\_StornoOpen\\_A16](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string invoiceNumber, string uNP, string stornoType, string stornoDocumentNumber, string stornoUNP, string stornoDateTime, string stornoFMNumber, string stornoReason)

*Open storno document - variant 16*

- Dictionary< string, string > [receipt\\_Fiscal\\_01\\_Open](#) (string operatorNumber, string operatorPassword, string tillNumber)

*Open fiscal receipt*

- Dictionary< string, string > [receipt\\_Fiscal\\_02\\_Open](#) (string operatorNumber, string operatorPassword, string tillNumber, string uNP)

*Open fiscal receipt with UNP*

- Dictionary< string, string > [receipt\\_Fiscal\\_Close](#) ()

*Close fiscal receipt*

- Dictionary< string, string > [receipt\\_Invoice\\_01\\_Open](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice)

*Open invoice*

- Dictionary< string, string > [receipt\\_FiscalOpen\\_A02](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice)

*Open invoice*

- Dictionary< string, string > [receipt\\_FiscalOpen\\_A04](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string uNP)

*Open invoice with UNP*

- Dictionary< string, string > [receipt\\_Invoice\\_Open](#) (string operatorNumber, string operatorPassword, string tillNumber, string invoice, string uNP)

*Open invoice with UNP*

- void [receipt\\_PrintClientInfo\\_01](#) (string elKType, string elK)

*Print customer information*

- void [receipt\\_PrintClientInfo\\_02](#) (string elKType, string elK, string sellerName)

*Print customer information*

- void [receipt\\_PrintClientInfo\\_03](#) (string elKType, string elK, string sellerName, string receiverName)

*Print customer information*

- void [receipt\\_PrintClientInfo\\_04](#) (string elKType, string elK, string sellerName, string receiverName, string clientName)

*Print customer information*

- void [receipt\\_PrintClientInfo\\_05](#) (string elKType, string elK, string sellerName, string receiverName, string clientName, string taxNo)

*Print customer information*

- void [receipt\\_PrintClientInfo\\_06](#) (string elKType, string elK, string sellerName, string receiverName, string clientName, string taxNo, string address1)

*Print customer information*

- void [receipt\\_PrintClientInfo\\_07](#) (string elKType, string elK, string sellerName, string receiverName, string clientName, string taxNo, string address1, string address2)

*Print customer information*

- void [receipt\\_Sale](#) (string textRow1, string textRow2, string taxGroup, string singlePrice, string quantity)

*Enter a product (sale)*

- void [receipt\\_Sale\\_TextRow1](#) (string textRow1, string taxGroup, string singlePrice, string quantity)

*Enter a product (sale)*

- void [receipt\\_Sale\\_TextRow1\\_WQuan](#) (string textRow1, string taxGroup, string singlePrice)

*Enter a product (sale)*

- void `receipt_Sale_TextRow2` (string textRow2, string taxGroup, string singlePrice, string quantity)

*Enter a product (sale)*

- void `receipt_Sale_Minimum` (string taxGroup, string singlePrice)

*Enter a product (sale)*

- void `receipt_Sale_Un` (string textRow1, string textRow2, string taxGroup, string singlePrice, string quantity, string measure)

*Enter a product (sale)*

- void `receipt_Sale_TextRow1Un` (string textRow1, string taxGroup, string singlePrice, string quantity, string measure)

*Enter a product (sale)*

- void `receipt_Sale_TextRow2Un` (string textRow2, string taxGroup, string singlePrice, string quantity, string measure)

*Enter a product (sale)*

- void `receipt_Sale_UnWText` (string taxGroup, string singlePrice, string quantity, string measure)

*Enter a product (sale)*

- void `receipt_Sale_CByPercent` (string textRow1, string textRow2, string taxGroup, string singlePrice, string quantity, string percent)

*Enter a product (sale)*

- void `receipt_Sale_TextRow1CByPercent` (string textRow1, string taxGroup, string singlePrice, string quantity, string percent)

*Enter a product (sale)*

- void `receipt_Sale_TextRow2CByPercent` (string textRow2, string taxGroup, string singlePrice, string quantity, string percent)

*Enter a product (sale)*

- void `receipt_Sale_CByPercentWText` (string taxGroup, string singlePrice, string quantity, string percent)

*Enter a product (sale)*

- void `receipt_Sale_UnCByPercent` (string textRow1, string textRow2, string taxGroup, string singlePrice, string quantity, string measure, string percent)

*Enter a product (sale)*

- void `receipt_Sale_TextRow1UnCByPercent` (string textRow1, string taxGroup, string singlePrice, string quantity, string measure, string percent)

*Enter a product (sale)*

- void `receipt_Sale_TextRow2UnCByPercent` (string textRow2, string taxGroup, string singlePrice, string quantity, string measure, string percent)

*Enter a product (sale)*

- void `receipt_Sale_UnCByPercentWText` (string taxGroup, string singlePrice, string quantity, string measure, string percent)

*Enter a product (sale)*

- void `receipt_Sale_CBySum` (string textRow1, string textRow2, string taxGroup, string singlePrice, string quantity, string abs)

*Enter a product (sale)*

- void `receipt_Sale_TextRow1CBySum` (string textRow1, string taxGroup, string singlePrice, string quantity, string abs)

*Enter a product (sale)*

- void `receipt_Sale_TextRow2CBySum` (string textRow2, string taxGroup, string singlePrice, string quantity, string abs)

*Enter a product (sale)*

- void `receipt_Sale_CBySumWText` (string taxGroup, string singlePrice, string quantity, string abs)

*Enter a product (sale)*

- void `receipt_Sale_UnCBySum` (string textRow1, string textRow2, string taxGroup, string singlePrice, string quantity, string measure, string abs)

*Enter a product (sale)*

- void [receipt\\_Sale\\_TextRow1UnCBySum](#) (string textRow1, string taxGroup, string singlePrice, string quantity, string measure, string abs)

*Enter a product (sale)*

- void [receipt\\_Sale\\_TextRow2UnCBySum](#) (string textRow2, string taxGroup, string singlePrice, string quantity, string measure, string abs)

*Enter a product (sale)*

- void [receipt\\_Sale\\_UnCBySumWText](#) (string taxGroup, string singlePrice, string quantity, string measure, string abs)

*Enter a product (sale)*

- void [receipt\\_DSale](#) (string textRow1, string textRow2, string department, string singlePrice, string quantity)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow1](#) (string textRow1, string department, string singlePrice, string quantity)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow1\\_WQuan](#) (string textRow1, string department, string singlePrice)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow2](#) (string textRow2, string department, string singlePrice, string quantity)

*Enter a product (sale)*

- void [receipt\\_DSale\\_Minimum](#) (string department, string singlePrice)

*Enter a product (sale)*

- void [receipt\\_DSale\\_Un](#) (string textRow1, string textRow2, string department, string singlePrice, string quantity, string measure)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow1Un](#) (string textRow1, string department, string singlePrice, string quantity, string measure)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow2Un](#) (string textRow2, string department, string singlePrice, string quantity, string measure)

*Enter a product (sale)*

- void [receipt\\_DSale\\_UnWText](#) (string department, string singlePrice, string quantity, string measure)

*Enter a product (sale)*

- void [receipt\\_DSale\\_CByPercent](#) (string textRow1, string textRow2, string department, string singlePrice, string quantity, string percent)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow1CByPercent](#) (string textRow1, string department, string singlePrice, string quantity, string percent)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow2CByPercent](#) (string textRow2, string department, string singlePrice, string quantity, string percent)

*Enter a product (sale)*

- void [receipt\\_DSale\\_CByPercentWText](#) (string department, string singlePrice, string quantity, string percent)

*Enter a product (sale)*

- void [receipt\\_DSale\\_UnCByPercent](#) (string textRow1, string textRow2, string department, string singlePrice, string quantity, string measure, string percent)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow1UnCByPercent](#) (string textRow1, string department, string singlePrice, string quantity, string measure, string percent)

*Enter a product (sale)*

- void [receipt\\_DSale\\_TextRow2UnCByPercent](#) (string textRow2, string department, string singlePrice, string quantity, string measure, string percent)

*Enter a product (sale)*

- void [receipt\\_DSale\\_UnCByPercentWText](#) (string department, string singlePrice, string quantity, string measure, string percent)

*Enter a product (sale)*

- void `receipt_DSale_CBySum` (string textRow1, string textRow2, string department, string singlePrice, string quantity, string abs)

*Enter a product (sale)*

- void `receipt_DSale_TextRow1CBySum` (string textRow1, string department, string singlePrice, string quantity, string abs)

*Enter a product (sale)*

- void `receipt_DSale_TextRow2CBySum` (string textRow2, string department, string singlePrice, string quantity, string abs)

*Enter a product (sale)*

- void `receipt_DSale_CBySumWText` (string department, string singlePrice, string quantity, string abs)

*Enter a product (sale)*

- void `receipt_DSale_UnCBySum` (string textRow1, string textRow2, string department, string singlePrice, string quantity, string measure, string abs)

*Enter a product (sale)*

- void `receipt_DSale_TextRow1UnCBySum` (string textRow1, string department, string singlePrice, string quantity, string measure, string abs)

*Enter a product (sale)*

- void `receipt_DSale_TextRow2UnCBySum` (string textRow2, string department, string singlePrice, string quantity, string measure, string abs)

*Enter a product (sale)*

- void `receipt_DSale_UnCBySumWText` (string department, string singlePrice, string quantity, string measure, string abs)

*Enter a product (sale)*

- void `receipt_DisplaySale` (string textRow, string taxGroup, string singlePrice, string quantity)

*Register and display of a sale*

- void `receipt_DisplaySale_Un` (string textRow, string taxGroup, string singlePrice, string quantity, string measure)

*Register and display of a sale*

- void `receipt_DisplaySale_CByPercent` (string textRow, string taxGroup, string singlePrice, string quantity, string percent)

*Register and display of a sale*

- void `receipt_DisplaySale_UnCByPercent` (string textRow, string taxGroup, string singlePrice, string quantity, string measure, string percent)

*Register and display of a sale*

- void `receipt_DisplaySale_CBySum` (string textRow, string taxGroup, string singlePrice, string quantity, string abs)

*Register and display of a sale*

- void `receipt_DisplaySale_UnCBySum` (string textRow, string taxGroup, string singlePrice, string quantity, string measure, string abs)

*Register and display of a sale*

- void `receipt_DisplaySale_Minimum` (string textRow, string taxGroup, string singlePrice)

*Register and display of a sale*

- void `receipt_DisplayDSale` (string textRow, string department, string singlePrice, string quantity)

*Register and display of a sale*

- void `receipt_DisplayDSale_Un` (string textRow, string department, string singlePrice, string quantity, string measure)

*Register and display of a sale*

- void `receipt_DisplayDSale_CByPercent` (string textRow, string department, string singlePrice, string quantity, string percent)

*Register and display of a sale*

- void `receipt_DisplayDSale_UnCByPercent` (string textRow, string department, string singlePrice, string quantity, string measure, string percent)



*Register and display of a sale*

- void [receipt\\_DisplayDSale\\_CBySum](#) (string textRow, string department, string singlePrice, string quantity, string abs)

*Register and display of a sale*

- void [receipt\\_DisplayDSale\\_UnCBySum](#) (string textRow, string department, string singlePrice, string quantity, string measure, string abs)

*Register and display of a sale*

- void [receipt\\_DisplayDSale\\_Minimum](#) (string textRow, string department, string singlePrice)

*Register and display of a sale*

- void [receipt\\_PLU\\_Sale](#) (string targetPLU, string quantity)

*Registering the sale of a programmed item*

- void [receipt\\_PLUSale\\_CByPercent](#) (string targetPLU, string quantity, string percent)

*Registering the sale of a programmed item*

- void [receipt\\_PLUSale\\_CBySum](#) (string targetPLU, string quantity, string abs)

*Registering the sale of a programmed item*

- void [receipt\\_PLUSale\\_Un](#) (string targetPLU, string quantity, string measure)

*Registering the sale of a programmed item*

- void [receipt\\_PLUSale\\_UnCByPercent](#) (string targetPLU, string quantity, string measure, string percent)

*Registering the sale of a programmed item*

- void [receipt\\_PLUSale\\_UnCBySum](#) (string targetPLU, string quantity, string measure, string abs)

*Registering the sale of a programmed item*

- void [receipt\\_PLUDep\\_Sale](#) (string targetPLU, string department, string quantity)

*Registering the sale of a programmed item*

- void [receipt\\_PLUDep\\_Sale\\_CByPercent](#) (string targetPLU, string department, string quantity, string percent)

*Registering the sale of a programmed item*

- void [receipt\\_PLUDep\\_Sale\\_CBySum](#) (string targetPLU, string department, string quantity, string abs)

*Registering the sale of a programmed item*

- void [receipt\\_PLUDep\\_Sale\\_Un](#) (string targetPLU, string department, string quantity, string measure)

*Registering the sale of a programmed item*

- void [receipt\\_PLUDep\\_Sale\\_UnCByPercent](#) (string targetPLU, string department, string quantity, string measure, string percent)

*Registering the sale of a programmed item*

- void [receipt\\_PLUDep\\_Sale\\_UnCBySum](#) (string targetPLU, string department, string quantity, string measure, string abs)

*Registering the sale of a programmed item*

- void [receipt\\_DisplayPLUSale](#) (string dChar, string targetPLU, string quantity)

*Registering the sale of a programmed item*

- void [receipt\\_DisplayPLUSale\\_CByPercent](#) (string dChar, string targetPLU, string quantity, string percent)

*Registering the sale of a programmed item*

- void [receipt\\_DisplayPLUSale\\_CBySum](#) (string dChar, string targetPLU, string quantity, string abs)

*Registering the sale of a programmed item*

- void [receipt\\_DisplayPLUSale\\_Un](#) (string dChar, string targetPLU, string quantity, string measure)

*Registering the sale of a programmed item*

- void [receipt\\_DisplayPLUSale\\_UnCByPercent](#) (string dChar, string targetPLU, string quantity, string measure, string percent)

*Registering the sale of a programmed item*

- void [receipt\\_DisplayPLUSale\\_UnCBySum](#) (string dChar, string targetPLU, string quantity, string measure, string abs)

*Registering the sale of a programmed item*

- void [receipt\\_DisplayDepPLU\\_Sale](#) (string dChar, string targetPLU, string department, string quantity)

*Registering the sale of a programmed item*

- void `receipt_DisplayDepPLU_Sale_CByPercent` (string dChar, string targetPLU, string department, string quantity, string percent)  
*Registering the sale of a programmed item*
- void `receipt_DisplayDepPLU_Sale_CBySum` (string dChar, string targetPLU, string department, string quantity, string abs)  
*Registering the sale of a programmed item*
- void `receipt_DisplayDepPLU_SaleUn` (string dChar, string targetPLU, string department, string quantity, string measure)  
*Registering the sale of a programmed item*
- void `receipt_DisplayDepPLU_SaleUn_CByPercent` (string dChar, string targetPLU, string department, string quantity, string measure, string percent)  
*Registering the sale of a programmed item*
- void `receipt_DisplayDepPLU_SaleUn_CBySum` (string dChar, string targetPLU, string department, string quantity, string measure, string abs)  
*Registering the sale of a programmed item*
- Dictionary< string, string > `receipt_Subtotal` (string toPrint, string toDisplay)  
*Subtotal*
- Dictionary< string, string > `receipt_Subtotal_CByPercent` (string toPrint, string toDisplay, string percent)  
*Subtotal*
- Dictionary< string, string > `receipt_Subtotal_CBySum` (string toPrint, string toDisplay, string abs)  
*Subtotal*
- Dictionary< string, string > `receipt_Total_PAmount` (string textRow1, string textRow2, string paidMode, string inputAmount)  
*Total*
- Dictionary< string, string > `receipt_Total_PAmountTextRow1` (string textRow1, string paidMode, string inputAmount)
- Dictionary< string, string > `receipt_Total_PAmountTextRow2` (string textRow2, string paidMode, string inputAmount)  
*Total*
- Dictionary< string, string > `receipt_Total_PAmountWithoutText` (string paidMode, string inputAmount)  
*Total*
- Dictionary< string, string > `receipt_Total` (string textRow1, string textRow2)  
*Total*
- Dictionary< string, string > `receipt_Total_TextRow1` (string textRow1)  
*Total*
- Dictionary< string, string > `receipt_Total_TextRow2` (string textRow2)  
*Total*
- Dictionary< string, string > `receipt_Total_WithoutText` ()  
*Total*
- void `receipt_Paper_Feed` (string linesCount)  
*Paper feed*
- void `receipt_Paper_Feed_InnerContainer` (string param)  
*Paper feed (ONLY for models SK1-21F and SK1-31F)*
- void `receipt_Paper_Cut` ()  
*Paper cut*
- void `receipt_Fiscal_Text` (string inputText)  
*Print free fiscal text*
- void `receipt_PFiscalText_01` (string font, string flags, string inputText)  
*Print free fiscal text*
- void `receipt_PFiscalText_02` (string font, string inputText)  
*Print free fiscal text*
- void `receipt_Fiscal_Cancel` ()

- Cancel fiscal receipt*
  - Dictionary< string, string > [receipt\\_CashIn\\_CashOut](#) (string amount)
- Cash in and cash out*
  - void [receipt\\_Print\\_Barcode\\_01](#) (string barcodeType, string alignment, string barcodeData)
- Print barcode*
  - void [receipt\\_Print\\_Barcode\\_02](#) (string barcodeType, string alignment, string barcodeData)
- Print barcode without text*
  - void [receipt\\_Separating\\_Line](#) (string lineType)
- Print separating line*
  - void [receipt\\_Separating\\_LineW](#) (string lineType, string linemask)
- Print separating line. The text "not payable!" with a triple height is added unconditionally*
  - Dictionary< string, string > [receipt\\_Current\\_Info](#) ()
- Information about current receipt*
  - void [receipt\\_Drawer\\_KickOut](#) (string mSec)
- Drawer opening*
  - void [receipt\\_Print\\_Duplicate](#) (string count)
- Print duplicate receipt*
  - Dictionary< string, string > [report\\_Service\\_Contracts](#) (string option)
- Report of the registered service agreements*
  - Dictionary< string, string > [report\\_DailyClosure\\_01](#) (string option)
- Daily closure*
  - Dictionary< string, string > [report\\_DailyClosure\\_02](#) (string option, string withoutClearOpInfo)
- Daily closure*
  - void [report\\_FMByNumRange](#) (string startNumber, string endNumber)
- Full fiscal memory report by block number*
  - void [report\\_FMByNumRange\\_SHA1](#) (string startNumber, string endNumber)
- Full fiscal memory report by block number - prints also the checksum using the SHA-1 algorithm.*
  - void [report\\_FMByDateRange\\_Short](#) (string fromDate, string toDate)
- Summary tax memory report from-to date*
  - void [report\\_FMByDateRange](#) (string fromDate, string toDate)
- Full fiscal memory report by fiscal record date*
  - void [report\\_FMByDateRange\\_SHA1](#) (string fromDate, string toDate)
- Full fiscal memory report by fiscal record date - each Z-report, prints also the checksum using the SHA-1 algorithm.*
  - void [report\\_FMByNumRange\\_Short](#) (string startNumber, string endNumber)
- Summary fiscal memory report for a period*
  - void [report\\_Operators](#) ()
- Operator's report*
  - Dictionary< string, string > [report\\_ExtDailyClosure\\_01](#) (string option)
- Extended daily financial report*
  - Dictionary< string, string > [report\\_ExtDailyClosure\\_02](#) (string option, string withoutClearOpInfo)
- Extended daily financial report*
  - Dictionary< string, string > [report\\_Items](#) (string option)
- Report by PLU*
  - Dictionary< string, string > [report\\_Items\\_InRange](#) (string option, string startPLU, string endPLU)
- Report items in range*
  - Dictionary< string, string > [report\\_Items\\_InRangeByGroup](#) (string option, string startPLU, string endPLU, string group)
- Items in range by by group*
  - Dictionary< string, string > [report\\_DailyClosureByDepartments\\_01](#) (string option)
- Daily financial report with data printed by department*
  - Dictionary< string, string > [report\\_DailyClosureByDepartments\\_02](#) (string option, string withoutClearOpInfo)

*Daily financial report with data printed by department*

- Dictionary< string, string > [report\\_DailyClosureByDepartmentsAndItems\\_01](#) (string option)

*Daily financial report with department and PLU printing*

- Dictionary< string, string > [report\\_DailyClosureByDepartmentsAndItems\\_02](#) (string option, string without← ClearOpInfo)

*Daily financial report with department and PLU printing*

- Dictionary< string, string > [items\\_Get\\_ItemsInformation](#) (string option)

*Get item information*

- Dictionary< string, string > [items\\_Set\\_Item](#) (string option, string taxGroup, string targetPLU, string group, string singlePrice, string quantity, string itemName)

*Program an item*

- Dictionary< string, string > [items\\_Set\\_ItemWithReplace](#) (string option, string taxGroup, string targetPLU, string group, string singlePrice, string replace, string quantity, string itemName)

*Program an item*

- Dictionary< string, string > [items\\_Set\\_ItemQuantity](#) (string option, string targetPLU, string quantity)

*Edit the available PLU quantity.*

- Dictionary< string, string > [items\\_Delete\\_Item](#) (string option, string targetPLU)

*Deletes the specified PLU code, if it has no accumulated sums*

- Dictionary< string, string > [items\\_Delete\\_ItemsInRange](#) (string option, string startPLU, string endPLU)

*Delete items in range*

- Dictionary< string, string > [items\\_Delete\\_All\\_Items](#) (string dOption, string aOption)

*Delete all items*

- Dictionary< string, string > [items\\_Get\\_Item](#) (string option, string targetPLU)

*Read item*

- Dictionary< string, string > [items\\_Get\\_FirstFoundItem](#) (string option, string targetPLU)

*Get first found item*

- Dictionary< string, string > [items\\_Get\\_LastFoundItem](#) (string option, string targetPLU)

*Get last found item*

- Dictionary< string, string > [items\\_Get\\_NextItem](#) (string option)

*Get next item*

- Dictionary< string, string > [items\\_Get\\_FirstSoldItem](#) (string option, string targetPLU)

*Get first sold item*

- Dictionary< string, string > [items\\_Get\\_LastSoldItem](#) (string option, string targetPLU)

*Return the data on the last found PLU item shaving sales*

- Dictionary< string, string > [items\\_Get\\_NextSoldItem](#) (string option, string targetPLU)

*Return the data on the next found sold PLU*

- Dictionary< string, string > [items\\_Get\\_FirstNotProgrammedItem](#) (string option, string targetPLU)

*Find the first not programmed item*

- Dictionary< string, string > [items\\_Get\\_LastNotProgrammedItem](#) (string option, string targetPLU)

*Find the last not programmed PLU*

- Dictionary< string, string > [config\\_SetLAN](#) (string ipAddress, string subnetMask, string portNumber, string defaultGateway, string dHCP, string MACAddress)

*Set LAN settings*

- Dictionary< string, string > [config\\_Set\\_Switches](#) (string switches)

*Store the settings and the switches in the flash memory*

- void [config\\_Set\\_HeaderLine](#) (string item, string value)

*Set header line*

- void [config\\_Set\\_FooterLine](#) (string item, string value)

*Set footer lines*

- void [config\\_Set\\_AutoFormat](#) (string option, string offOn)

*Set auto format*

- void [config\\_Set\\_BarcodeHeight](#) (string option, string value)  
*Set barcode height*
- void [config\\_Set\\_AutoCutting](#) (string option, string offOn)  
*Set auto paper cutting*
- void [config\\_Set\\_PrintDensity](#) (string option, string value)  
*Set print density*
- void [config\\_enable\\_EUR](#) (string option, string on, string rate)  
*Set enabled EUR*
- void [config\\_disable\\_EUR](#) (string option, string off)  
*Set disabled EUR*
- void [config\\_Set\\_FontSize](#) (string option, string value)  
*Set font size*
- void [config\\_Set\\_PrinterReadinessPrintMsgs](#) (string option, string offOn)  
*Enables / disables the printer's readiness print message.*
- void [config\\_Set\\_AutomaticZReport](#) (string option, string offOn, string optZRep)  
*Enable / disable automatic Z-report printing at 23:59:59 o'clock if issuance of a fiscal or service entry for the day.*
- void [config\\_Set\\_UNPPrinting](#) (string option, string offOn)  
*Enable / disable UNP printing of fiscal and reversal receipts.*
- void [config\\_Set\\_PrintLogo](#) (string option, string offOn, string height)  
*Enable/disable printing the logo graphic right before the HEADER*
- void [config\\_Set\\_LANSignal](#) (string option, string offOn)  
*Enable / disable the sound signal when attaching / disconnecting to the device on TCP / IP.*
- void [config\\_Set\\_PrintDepartment](#) (string option, string offOn)  
*Set print department*
- void [config\\_enable\\_PrintSuppression](#) (string option, string on, string lines, string seconds)  
*Enable suppressed printing in a fiscal or non-fiscal receipt*
- void [config\\_disable\\_PrintSuppression](#) (string option, string off)  
*Disable suppressed printing in a fiscal or non-fiscal receipt*
- void [config\\_Set\\_2GForcedModem](#) (string option, string offOn)  
*Enable / disable 2G forced modem operation (for 3G models) - one byte of '1' (enabled) or '0' (disabled).*
- void [config\\_Set\\_ExtraSpace](#) (string option, string value)  
*Set extra space*
- void [config\\_Set\\_SIM](#) (string option, string indexValue)  
*Set SIM*
- void [config\\_Set\\_PrintTaxDDS](#) (string option, string offOn)  
*Enable/disable print tax*
- void [config\\_Set\\_CashDrawerPulse](#) (string option, string offOn)  
*Set cash drawer pulse*
- void [config\\_Set\\_WarningTimeRange](#) (string option, string warnTimeRec, string warnTimeGInfo)  
*Set warning time range*
- void [config\\_Set\\_WorkingWithPinpad](#) (string enable)  
*Set fiscal device to work with a Datecs payment terminal*
- void [config\\_Set\\_FeedButton](#) (string option, string offOn)  
*Disable/enable the FEED button functionality*
- void [config\\_Set\\_OptionalErrorText](#) (string option, string offOn)  
*Disable/enable returning optional text in the command response when an error occurs*
- void [config\\_Set\\_RollUpPaper](#) (string option, string offOn)  
*Enable / disable automatic roll-up of paper in a container after each document. After turning on the printer, the printer does not pick up the paper in a container. ONLY FOR PRINTERS SK1-21F AND SK1-31F.*
- void [config\\_Set\\_AutomaticRetractPaper](#) (string option, string offOn)

*Set time to automatically retract paper in a container after each printed and waiting to takeout document. ONLY FOR PRINTERS SK1-21F AND SK1-31F.*

- void [config\\_Set\\_DateTime](#) (string dateTime)  
*Set date and time*
- Dictionary< string, string > [config\\_Set\\_InvoiceRange](#) (string startValue, string endValue)  
*Set invoice range*
- void [config\\_Restore\\_PrintSuppression](#) (string restoreOption)  
*Force suppressed printing*
- Dictionary< string, string > [config\\_Set\\_AdditionalPaymentName](#) (string option, string additionalPaymentName)  
*Set additional payment names*
- void [config\\_Set\\_DepartmentName](#) (string departmentNumber, string taxGroup, string textRow1)  
*Program a department name*
- void [config\\_Set\\_DepartmentNameTwoRows](#) (string departmentNumber, string taxGroup, string textRow1, string textRow2)  
*Program a department name for two lines*
- void [config\\_Set\\_OperatorPassword](#) (string operatorCode, string oldPassword, string newPassword)  
*Set operator's password*
- void [config\\_Set\\_OperatorName](#) (string operatorCode, string password, string operatorName)  
*Set operator's name*
- void [config\\_Set\\_Logo](#) (string rowNum, string data)  
*Program graphic logo*
- Dictionary< string, string > [config\\_SafetyTurnOffDevice](#) ()  
*Safety turn off the printer or modem*
- Dictionary< string, string > [info\\_LastErrorExtendedInfo](#) ()  
*The command allows to obtain information on the last command completed with an error. This information is saved for successfully completed commands and also after the printer is powered off. It is cleared only on RAM reset and on execution of the command with a "CLEAR" text input.*
- Dictionary< string, string > [info\\_ClearLastErrorExtendedInfo](#) ()  
*Clear extended information about last error*
- Dictionary< string, string > [info\\_Service\\_Contracts](#) ()  
*Service contracts information and report*
- Dictionary< string, string > [info\\_Get\\_LANSettings](#) ()  
*Get last settings*
- Dictionary< string, string > [info\\_Get\\_NRADData\\_Type1](#) ()  
*Read NRA data type 1*
- Dictionary< string, string > [info\\_Get\\_NRADData\\_Type2](#) ()  
*NRA data type 2 - Sales for tax groups for the day*
- Dictionary< string, string > [info\\_Get\\_NRADData\\_Type3](#) ()  
*Read NRA type 3*
- Dictionary< string, string > [info\\_Get\\_NRADData\\_Type4](#) ()  
*Read NRA type 4*
- Dictionary< string, string > [info\\_Get\\_NRADData\\_Type5](#) ()  
*Read NRA type 5*
- Dictionary< string, string > [info\\_Get\\_NRADData\\_Type6](#) (string closure)  
*Read NRA type 6*
- Dictionary< string, string > [info\\_Get\\_NRADData\\_Type7](#) ()  
*Read NRA type 7*
- Dictionary< string, string > [info\\_Get\\_NRADData\\_Type8](#) (string option)  
*Read NRA type 8 - Returns PLU sales data by PLU codes in ascending order*
- Dictionary< string, string > [info\\_Get\\_PrintOption](#) (string option, string targetOption)

- *Get print options*
- Dictionary< string, string > [info\\_Get\\_TaxRatesByPeriod](#) (string startDate, string endDate)
- *Get tax rates by period*
- Dictionary< string, string > [info\\_Get\\_DataForLastIssuedReceipt](#) ()
- *Finds the last issued fiscal or reversal receipt before a daily report with clearing is made /ONLY FOR TYPE 21/*
- Dictionary< string, string > [info\\_Get\\_DataForLastIssuedReceipt](#) (string docNum)
- *Finds a document with document number only if it is an electronic document. /ONLY FOR TYPE 21/*
- Dictionary< string, string > [info\\_Get\\_DateTime](#) ()
- *Get date and time*
- Dictionary< string, string > [info\\_Get\\_DateTime\\_01](#) ()
- *Get date and time separated*
- Dictionary< string, string > [info\\_Get\\_LastFiscRecord](#) ()
- *Get last fiscal record*
- Dictionary< string, string > [info\\_Get\\_LastFiscRecord\\_01](#) ()
- *Get last fiscal record with time*
- Dictionary< string, string > [info\\_Get\\_AdditionalDailyInfo\\_01](#) (string option)
- *Information of current sums for the day*
- Dictionary< string, string > [info\\_Get\\_InvoiceRange](#) ()
- *Get invoice range*
- Dictionary< string, string > [info\\_Get\\_FreeFMRecords](#) ()
- *Get free fiscal memory records*
- Dictionary< string, string > [info\\_Get\\_CashIn\\_CashOut](#) ()
- *Get cash in/ cash out*
- void [info\\_Print\\_Diagnostic\\_0](#) ()
- *Print diagnostic information*
- Dictionary< string, string > [info\\_Get\\_StatusBytes](#) (string option)
- *Depends on option you choose: 'W' - First, waits for the printer to finish printing everything in the print buffer or 'X' - Does not wait for the printer, but responds immediately*
- Dictionary< string, string > [info\\_Get\\_NotPrintedRowsCount](#) (string option)
- *Get not printed rows count*
- Dictionary< string, string > [info\\_Get\\_NRADocuments](#) (string option)
- *Returns data on the customer documents sent to the NRA server*
- Dictionary< string, string > [info\\_Get\\_CashDrawerStatusOrOpenShift](#) (string option)
- *If option is 'D' - Returns cash drawer status (0 - closed drawer, 1 - open drawer) for models that support it(FP-2000 or FP-700) if supported by the connected drawer. If option is 'B' - Returns the status of open shift lock*
- Dictionary< string, string > [info\\_Get\\_FTransactionStatus](#) (string option)
- void [info\\_Print\\_Short\\_FMReportByDTRange](#) (string fromDate, string toDate)
- *Summary fiscal memory report by period*
- void [info\\_Print\\_Short\\_MonthlyFMReport](#) (string startValue)
- *Monthly fiscal memory report*
- void [info\\_Print\\_Short\\_YearlyFMReport](#) (string startValue)
- *Annual fiscal memory report*
- void [info\\_Get\\_VoltageAndTemperature](#) (string voltage, string temperature)
- *Get the supply voltage and temperature*
- Dictionary< string, string > [info\\_Get\\_DecimalsAndTaxRates](#) ()
- *Get decimals and tax rates*
- Dictionary< string, string > [info\\_Get\\_AdditionalPaymentNames](#) (string option)
- *Get additional payment names*
- Dictionary< string, string > [info\\_Get\\_FMRecord\\_LastDateTime](#) (string option)
- *Read date of last fiscal memory record*



- Dictionary< string, string > [info\\_Get\\_FMRecord\\_LastDateTime\\_01](#) (string option)  
*Get date and time from last FM records (separated)*
- Dictionary< string, string > [info\\_Get\\_DepartmentInfo](#) (string departmentNumber)  
*Get department information*
- Dictionary< string, string > [info\\_Get\\_DiagnosticInfo](#) ()  
*Get diagnostic information*
- Dictionary< string, string > [info\\_Get\\_DailyCorrections](#) ()  
*Get daily corrections*
- Dictionary< string, string > [info\\_Get\\_TaxRates](#) ()  
*Get tax rates*
- Dictionary< string, string > [info\\_Get\\_EIK](#) ()  
*Read UIC*
- Dictionary< string, string > [info\\_Get\\_CurrentRecieptInfo](#) ()  
*Get current receipt information*
- Dictionary< string, string > [info\\_Get\\_AdditionalDailyInfo\\_02](#) ()  
*Get additional daily information*
- Dictionary< string, string > [info\\_Get\\_AdditionalDailyInfo\\_03](#) ()  
*Get additional daily information - extended*
- Dictionary< string, string > [info\\_Get\\_OperatorsData](#) (string wpOperator)  
*Get operator's data*
- Dictionary< string, string > [info\\_Get\\_LastDocumentNumber](#) ()  
*Get last document number*
- Dictionary< string, string > [info\\_Get\\_ActiveTaxes](#) (string closureNumber, string option)  
*Get active taxes*
- Dictionary< string, string > [info\\_Get\\_Sums](#) (string closure1, string option, string closure2)  
*Information on the sales for the specified record or period*
- Dictionary< string, string > [info\\_Get\\_StornoSums](#) (string closure1, string option, string closure2)  
*Information on the storno sales for the specified record or period*
- Dictionary< string, string > [info\\_Get\\_NettoSums](#) (string closure1, string option, string closure2)  
*Information on the net amounts for the specified record or period*
- Dictionary< string, string > [info\\_Get\\_Storno\\_NettoSums](#) (string closure1, string option, string closure2)  
*Information on the storno netto sums for the specified record or period*
- Dictionary< string, string > [info\\_Get\\_VATSums](#) (string closure1, string option, string closure2)  
*Information on the VAT assessed for the specified record or period*
- Dictionary< string, string > [info\\_Get\\_Storno\\_VATSums](#) (string closure1, string option, string closure2)  
*Information on the storno VAT assessed for the specified record or period*
- Dictionary< string, string > [info\\_Get\\_Numbers](#) (string closure1, string option)  
*Additional information on the specified record*
- Dictionary< string, string > [info\\_Get\\_TaxRateValues](#) (string closure1, string option)  
*Information on the specified fiscal memory record used to set tax rates*
- Dictionary< string, string > [info\\_Get\\_RAMResetDateTime](#) (string closure1, string option)  
*Information on the specified FM record with RAM reset*
- Dictionary< string, string > [info\\_Get\\_AdditionalPayments](#) (string closure1, string option, string closure2)  
*Information on the amounts by payment type for the specified record or period*
- Dictionary< string, string > [info\\_Get\\_Corrections](#) (string closure1, string option, string closure2)  
*Information on the amounts by payment type for the specified record or period*
- Dictionary< string, string > [info\\_Get\\_CashInCashOut](#) (string closure1, string option, string closure2)  
*Get cash in/ cash out*
- Dictionary< string, string > [info\\_Get\\_Turnover](#) (string closure1, string option)  
*Information on the total sales and VAT before the specified Z-report*
- Dictionary< string, string > [info\\_Get\\_ProgrammedFuel](#) (string closure1, string option)



*ONLY for type 3 and 31 printers! Information on the programmed fuel FTT and names for this Z-report. Only for type 3 and 31 printers.*

- Dictionary< string, string > [info\\_Get\\_SoldQuant\\_AccumulatedSales](#) (string closure1, string option)  
*Information on the sold quantities and accumulated sales for the programmed fuels for this Z-report. Only for type 3 and 31 printers.*
- Dictionary< string, string > [info\\_Get\\_ProgrammedPumps](#) (string closure1, string option)  
*Information on the programmed pumps for this Z-report. Only for type 31 printers.*
- Dictionary< string, string > [info\\_Get\\_ProgrammedPumps2](#) (string closure1, string option)  
*Information on the programmed pumps for this Z-report - 0 - Tank volume at current temperature. Vol15Ci Tank volume at 15 degrees.. Only for type 31 printers*
- Dictionary< string, string > [info\\_Get\\_ProgrammedPumps3](#) (string closure1, string option, string pumpNum)  
*Information on a programmed pump for this Z-report. ONLY for type 31 printers - Nozzle number, Nozzle counter value*
- Dictionary< string, string > [info\\_Get\\_LogoLine](#) (string rowNum)  
*Get logo data from device by line number.*
- void [klen\\_Print\\_Validation](#) (string subcommand01, string subcommand02)  
*Print validation*
- void [klen\\_Print\\_SHA1](#) (string subcommand01, string subcommand02)  
*Print with SHA1 - prints information on each found SHA-1, valid or not*
- Dictionary< string, string > [klen\\_Get\\_KLENInfo\\_ZReport](#) (string subcommand01, string subcommand02, string zReportNumber)  
*Get EJ information for Z report number*
- Dictionary< string, string > [klen\\_Get\\_ZReportInfo](#) (string subcommand01, string subcommand02, string zReportNumber)  
*Reads the actual text of the documents stored in the EJ, calculates the SHA-1 and compares it to the SHA-1 from the Z-report. The command can take a long time to complete, if there are many documents in the daily report.*
- Dictionary< string, string > [klen\\_Get\\_Info](#) (string subcommand01, string subcommand02)  
*Electronic journal information*
- Dictionary< string, string > [klen\\_Find\\_Document](#) (string subcommand01, string documentType, string documentNumber)  
*Read documents*
- Dictionary< string, string > [klen\\_Set\\_DocsRange](#) (string subcommand01, string documentType, string startNumber, string endNumber)  
*Read EJ in range*
- Dictionary< string, string > [klen\\_Set\\_DocsRange\\_InZReport\\_01](#) (string subcommand01, string documentType, string zReportNumber)  
*Read EJ data (Z reports numbers)*
- Dictionary< string, string > [klen\\_Set\\_DocsRange\\_InZReport\\_02](#) (string subcommand01, string documentType, string zReportNumber, string documentNumber)  
*Read EJ data in range (Z reports)*
- Dictionary< string, string > [klen\\_Set\\_DocsRange\\_InZReport](#) (string subcommand01, string documentType, string zReportNumber, string startNumber, string endNumber)  
*Read EJ data in range with document numbers and Z report*
- Dictionary< string, string > [klen\\_Set\\_DocsRange\\_ByDateTime](#) (string subcommand01, string documentType, string fromDateTime, string toDateTime)  
*Read EJ data in range by date and time*
- Dictionary< string, string > [klen\\_Get\\_NextTextRow](#) (string subcommand01)  
*Get next text row*
- Dictionary< string, string > [klen\\_Print\\_Document](#) (string subcommand01, string fontSize, string documentType, string documentNumber)  
*Print EJ data*
- Dictionary< string, string > [klen\\_Print\\_DocsInRange](#) (string subcommand01, string fontSize, string documentType, string startNumber, string endNumber)

*Print EJ data*

- Dictionary< string, string > [klen\\_Print\\_DocsInRange\\_InZReport](#) (string subcommand01, string fontSize, string documentType, string zReportNumber, string startNumber, string endNumber)

*Print EJ data*

- Dictionary< string, string > [klen\\_Print\\_DocsFromDateToDate](#) (string subcommand01, string fontSize, string documentType, string fromDateTime, string toDateTime)

*Print documents by date and time*

- Dictionary< string, string > [klen\\_SInfo\\_W\\_ByNumber](#) (string option, string startNumber)

*Read structured information*

- Dictionary< string, string > [klen\\_SInfo\\_W\\_ByNumbersRange](#) (string option, string startNumber, string end↵Number)

*Read structured information by range*

- Dictionary< string, string > [klen\\_SInfo\\_W\\_ByDateRange](#) (string option, string optionD, string fromDateTime, string toDateTime)

*Read structured information by date and time*

- Dictionary< string, string > [klen\\_SInfo\\_W\\_GetNext](#) (string option)

*Get next text line structured information*

- Dictionary< string, string > [klen\\_SInfo\\_Y\\_ByNumber](#) (string option, string startNumber)

*Return structured information*

- Dictionary< string, string > [klen\\_SInfo\\_Y\\_ByNumbersRange](#) (string option, string startNumber, string end↵Number)

*Get structured information*

- Dictionary< string, string > [klen\\_SInfo\\_Y\\_ByDateRange](#) (string option, string optionD, string fromDateTime, string toDateTime)

*Get structured information*

- Dictionary< string, string > [klen\\_SInfo\\_Y\\_GetNext](#) (string option)

*Get next line structured information*

- Dictionary< string, string > [klen\\_SInfo\\_V\\_ByNumber](#) (string option, string startNumber)

*Return structured information by number*

- Dictionary< string, string > [klen\\_SInfo\\_V\\_ByNumbersRange](#) (string option, string startNumber, string end↵Number)

*Return structured information by number*

- Dictionary< string, string > [klen\\_SInfo\\_V\\_ByDateRange](#) (string option, string optionD, string fromDateTime, string toDateTime)

*Get structured information by date and time*

- Dictionary< string, string > [klen\\_SInfo\\_V\\_GetNext](#) (string option)

*Get next line structured information*

- Dictionary< string, string > [klen\\_SInfo\\_ExV\\_ByNumber](#) (string option, string exOption, string startNumber)

*Get structured information by date and time plus date and time*

- Dictionary< string, string > [klen\\_SInfo\\_ExV\\_ByNumbersRange](#) (string option, string exOption, string start↵Number, string endNumber)

*Return structured information by numbers range plus date and time*

- Dictionary< string, string > [klen\\_SInfo\\_ExV\\_ByDateRange](#) (string option, string exOption, string optionD, string fromDateTime, string toDateTime)

*Get structured information by date and time range plus date and time*

- Dictionary< string, string > [klen\\_SInfo\\_ExV\\_GetNext](#) (string option)

*Get next line structured information*

- Dictionary< string, string > [info\\_Pinpad\\_CheckLastRecordedRRN](#) ()

*Check recently recorded RRN code in fiscal printer with RRN from last transaction in payment terminal*

- Dictionary< string, string > [info\\_Pinpad\\_DataForLastTransaction](#) ()

*Data for last transaction*

- Dictionary< string, string > [info\\_Pinpad\\_PrintLastSuccessfulTransaction](#) ()

- Prints the last successful transaction*

  - Dictionary< string, string > [info\\_Pinpad\\_Get\\_DateTime](#) ()

*Get date and time of payment terminal. The command is forbidden in an open fiscal receipt*

- void [config\\_Pinpad\\_Set\\_DateTime](#) (string dateTime)

*Set date and time. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [info\\_Pinpad\\_TestConnection](#) ()

*Test connection to a payment terminal. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [info\\_Pinpad\\_NumberOfEntries](#) ()

*Number of entries in the payment terminal. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [info\\_Pinpad\\_Information](#) ()

*Payment terminal information. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [config\\_Pinpad\\_SetBatchNumber](#) (string batch)

*Set batch number. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [config\\_Pinpad\\_DeleteBatchNumber](#) ()

*Deletes all transactions from a payment terminal. The command is forbidden in an open fiscal receipt.*

- Dictionary< string, string > [config\\_Pinpad\\_DeleteFreezeTransaction](#) ()

*Deletes (unsuccessful) freeze transactions from a payment terminal, if any. The command is forbidden in an open fiscal receipt.*

- Dictionary< string, string > [config\\_Pinpad\\_SetReceiptNumber](#) (string stan)

*The next receipt will be issued under the assigned number. The command is forbidden in an open fiscal receipt.*

- Dictionary< string, string > [receipt\\_Pinpad\\_PrintByNumber](#) (string number)

*Print a receipt by number. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [receipt\\_Pinpad\\_PrintLastSuccessfultransaction](#) ()

*Printing the last successful transaction in a payment terminal.  
It can only be executed if a receipt is open and the last transaction is completed without a reply from a payment terminal. If the A4h command (164 - [receipt\\_Pinpad\\_ReturnLastSuccessfultransaction](#)) has been executed before, or with card payment it will also be successfully rejected. The receipt is printed automatically after the fiscal receipt has been closed. Requires execution before returning command 148([info\\_Pinpad\\_CheckLastRecordedRRN\(\)](#)) for reference and return data collection.*

- Dictionary< string, string > [receipt\\_Pinpad\\_ReturnLastSuccessfultransaction](#) ()

*Return the last successful transaction. It can only be executed if a receipt is open and the last transaction is completed without a reply from a payment terminal. If the A3h command (163 - [receipt\\_Pinpad\\_PrintLastSuccessfultransaction\(\)](#)) was previously executed or the card payment is successful, it will also be rejected. The receipt is printed automatically after the fiscal receipt has been closed. Requires execution before returning command 148([info\\_Pinpad\\_CheckLastRecordedRRN\(\)](#)) for reference and return data.*

- Dictionary< string, string > [config\\_Pinpad\\_SetRRNFromPPToFD](#) ()

*Establishment of RRN code in fiscal device with RRN code from payment terminal.  
It is used when changing the Fiscal Device and / or payment terminal. The command is forbidden in an open fiscal receipt.*

- Dictionary< string, string > [info\\_Pinpad\\_TestConnectionToServer](#) ()

*Test connection to server. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [report\\_Pinpad\\_EndOfDay](#) ()

*End of day report. The command first prints the (unsuccessful) freeze reversal receipts, then sends data to the server, and finishes with printing a short report in response to the successful or unsuccessful clearing of the registers. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [info\\_Pinpad\\_GetUpdate](#) ()

*Checking and updating of payment terminal software. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [report\\_Pinpad\\_Reports](#) (string type)

*Reports. The command is forbidden in an open fiscal receipt*

- Dictionary< string, string > [info\\_Pinpad\\_StatusOfLastTransaction](#) ()

*Status of last transaction*

- Dictionary< string, string > [info\\_Pinpad\\_ClearTransactionStatusFlags](#) ()  
*Clearing transaction status flags. Clears the flags specified in the command 172([info\\_Pinpad\\_StatusOfLastTransaction\(\)](#)). The command is forbidden in an open fiscal receipt.*
- Dictionary< string, string > [config\\_Pinpad\\_HeaderPrintingOptions](#) (string option)  
*Header printing options. The command is forbidden in an open fiscal receipt.*
- Dictionary< string, string > [config\\_Pinpad\\_HeaderPrintingOptions](#) (string option, string numReceipts)  
*Header printing options. The command is forbidden in an open fiscal receipt.*
- Dictionary< string, string > [receipt\\_Pinpad\\_PrintReceiptCopyFromEJ\\_ByNum](#) (string receiptNum)  
*Printing a copy of an receipt from the EJ by receipt number*
- Dictionary< string, string > [receipt\\_Pinpad\\_PrintReceiptCopyFromEJ\\_ByDate](#) (string date)  
*Printing a copy of an receipt from the EJ by date*
- Dictionary< string, string > [info\\_Pinpad\\_TransactionStatus](#) ()  
*Transaction status (unsuccessful). The command is forbidden in an open fiscal receipt*
- void [other\\_Sound\\_Signal](#) (string soundData)  
*Set sound signal*
- void [service\\_Set\\_DecimalsAndTaxRates](#) (string multiplier, string decimals, string currencyName, string enabledMask, string taxA, string taxB, string taxC, string taxD, string taxE, string taxF, string taxG, string taxH)  
*Set decimals and tax rates*
- Dictionary< string, string > [service\\_Set\\_ProductionTestArea](#) (string option)  
*Set production test area*
- Dictionary< string, string > [service\\_Set\\_SerialNumber](#) (string serialNumber, string fMNumber)  
*Set serial number and fiscal memory number*
- Dictionary< string, string > [service\\_Set\\_EIK](#) (string eikValue, string eikName)  
*Set UIC*
- void [service\\_RAM\\_Reset](#) ()  
*Service RAM reset*
- void [service\\_Disable\\_Print](#) (string disabled)  
*Service disabled printing*
- Dictionary< string, string > [service\\_Format\\_KLEN](#) (string option)  
*Service EJ maintenance*
- Dictionary< string, string > [service\\_Test\\_GPRS](#) (string option)  
*GPRS modem test*

### 1.1.1 Detailed Description

Fiscal Devices Group "A" - Bulgaria

### 1.1.2 Member Function Documentation

#### 1.1.2.1 [config\\_disable\\_EUR\(\)](#)

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_disable_EUR (
    string option,
    string off )
```

Set disabled EUR

**Parameters**

<i>option</i>	'E' - for disable EUR
<i>off</i>	'0' - disabled

**1.1.2.2 config\_disable\_PrintSuppression()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_disable_PrintSuppression (
    string option,
    string off )
```

Disable suppressed printing in a fiscal or non-fiscal receipt

**Parameters**

<i>option</i>	'Q' - one byte for suppressed printing
<i>off</i>	'0' - disables accumulation of unprinted lines in the receipt

**1.1.2.3 config\_enable\_EUR()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_enable_EUR (
    string option,
    string on,
    string rate )
```

Set enabled EUR

**Parameters**

<i>option</i>	'E' - for enable EUR
<i>on</i>	'1' - enabled
<i>rate</i>	currently programmed exchange rate

**1.1.2.4 config\_enable\_PrintSuppression()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_enable_PrintSuppression (
    string option,
    string on,
    string lines,
    string seconds )
```

Enable suppressed printing in a fiscal or non-fiscal receipt

**Parameters**

<i>option</i>	'Q' - one byte for suppressed printing
<i>on</i>	'1' - enables accumulation of unprinted lines in the receipt
<i>lines</i>	Between 4 and 1000 or 0. If the value is 0, printing starts when the print buffer is full.
<i>seconds</i>	Between 2 and 120 or 0. If the value is 0, there is no time limit

**1.1.2.5 config\_Pinpad\_DeleteBatchNumber()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Pinpad_DeleteBatchNumber ( )
```

Deletes all transactions from a payment terminal. The command is forbidden in an open fiscal receipt.

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.6 config\_Pinpad\_DeleteFreezeTransaction()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Pinpad_DeleteFreezeTransaction  
( )
```

Deletes (unsuccessful) freeze transactions from a payment terminal, if any. The command is forbidden in an open fiscal receipt.

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.7 config\_Pinpad\_HeaderPrintingOptions() [1/2]**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Pinpad_HeaderPrintingOptions  
(  
    string option )
```

Header printing options. The command is forbidden in an open fiscal receipt.

**Parameters**

<i>option</i>	number in the range [0 - 4], which determines the printing of the header in repeated receipt information(city, address, po box, phone.): 0 – short repeated receipt; 1 – full information; 2 – full information without phone; 3 - the current status of the print options and the number of repeated receipts; 4 - allows to set the number of repeated receipts
---------------	--

**Returns**

Dictionary with keys:

fAnswer	'P' or current print option
fAnswer1	If present - the current number of reprinted receipts.

**1.1.2.8 config\_Pinpad\_HeaderPrintingOptions() [2/2]**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Pinpad_HeaderPrintingOptions
(
    string option,
    string numReceipts )
```

Header printing options.The command is forbidden in an open fiscal receipt.

**Parameters**

<i>option</i>	number in the range [0 - 4], which determines the printing of the header in repeated receipt information(city, address, po box, phone.): 0 – short repeated receipt; 1 – full information; 2 – full information without phone; 3 - the current status of the print options and the number of repeated receipts; 4 - allows to set the number of repeated receipts
<i>numReceipts</i>	Number representing the number of repeat receipts - max10

**Returns**

Dictionary with keys:

fAnswer	'P' or current print option
fAnswer1	If present - the current number of reprinted receipts.

**1.1.2.9 config\_Pinpad\_Set\_DateTime()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Pinpad_Set_DateTime (
```

```
string dateTime )
```

Set date and time. The command is forbidden in an open fiscal receipt

#### Parameters

<i>dateTime</i>	In format YY-MM-DD HH:MM:SS
-----------------	-----------------------------

#### 1.1.2.10 config\_Pinpad\_SetBatchNumber()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Pinpad_SetBatchNumber (
    string batch )
```

Set batch number. The command is forbidden in an open fiscal receipt

```
///
```

#### Parameters

<i>batch</i>	Number up to 65535
--------------	--------------------

#### Returns

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

#### 1.1.2.11 config\_Pinpad\_SetReceiptNumber()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Pinpad_SetReceiptNumber (
    string stan )
```

The next receipt will be issued under the assigned number. The command is forbidden in an open fiscal receipt.

#### Parameters

<i>stan</i>	Receipt number up to 65535
-------------	----------------------------

#### Returns

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---



**1.1.2.12 config\_Pinpad\_SetRRNFromPPToFD()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Pinpad_SetRRNFromPPToFD ( )
```

Establishment of RRN code in fiscal device with RRN code from payment terminal.

It is used when changing the Fiscal Device and / or payment terminal. The command is forbidden in an open fiscal receipt.

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.13 config\_Restore\_PrintSuppression()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Restore_PrintSuppression (
    string restoreOption )
```

Force suppressed printing

**Parameters**

<i>restoreOption</i>	One byte with the following possible values: '0' - After the end of printing, printing is reenabled—next data will be printed immediately. '1' - After the print buffer is cleared, printing remains suppressed until the end of the receipt.
----------------------	--

**1.1.2.14 config\_SafetyTurnOffDevice()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_SafetyTurnOffDevice ( )
```

Safety turn off the printer or modem

**Returns**

Dictionary with key:

data	'SHUTDOWN' for devices FP-800 / FP-2000 / FP-650 / SK1-21F / SK1-31F or "OFF" for devices FMP-10 / FP-700
------	---

**1.1.2.15 config\_Set\_2GForcedModem()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_2GForcedModem (
```

```
string option,
string offOn )
```

Enable / disable 2G forced modem operation (for 3G models) - one byte of '1' (enabled) or '0' (disabled).

#### Parameters

<i>option</i>	"V" - Enable / disable 2G forced modem operation (for 3G models) - one byte of '1' (enabled) or '0' (disabled).
<i>offOn</i>	'0' - disabled; '1' - enabled

#### 1.1.2.16 config\_Set\_AdditionalPaymentName()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_AdditionalPaymentName (
    string option,
    string additionalPaymentName )
```

Set additional payment names

#### Parameters

<i>option</i>	'I' Additional payment 1; 'J' Additional payment 2; 'K' Additional payment 3; 'L' Additional payment 4. 'i' Additional payment 1. Identical to 'I'; 'j' Additional payment 2. Identical to 'J'; 'k' Additional payment 3. Identical to 'K'; 'l' Additional payment 4. Identical to 'L'; 'm' Additional payment 5; 'n' Additional payment 6; 'o' Additional payment 7; 'p' Additional payment 8 'q' Additional payment 9; 'r' Additional payment 10; 's' Additional payment 11
<i>additionalPaymentName</i>	Name of the respective payment type (up to 24 characters).

#### Returns

Dictionary with key:

<i>errorCode</i>	'P' - pass; 'F' - failed.
------------------	---------------------------

#### 1.1.2.17 config\_Set\_AutoCutting()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_AutoCutting (
    string option,
    string offOn )
```

Set auto paper cutting

#### Parameters

<i>option</i>	'C' - for auto paper cutting
<i>offOn</i>	'0' - disables; '1' - enables auto paper cutting

**1.1.2.18 config\_Set\_AutoFormat()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_AutoFormat (
    string option,
    string offOn )
```

Set auto format

**Parameters**

<i>option</i>	'A' - Automatic invoice-like sales formatting (4 separate lines)
<i>offOn</i>	'0' - disables invoice; '1' - enables invoice

**1.1.2.19 config\_Set\_AutomaticRetractPaper()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_AutomaticRetractPaper (
    string option,
    string offOn )
```

Set time to automatically retract paper in a container after each printed and waiting to takeout document. ONLY FOR PRINTERS SK1-21F AND SK1-31F.

**Parameters**

<i>option</i>	'O' - Set time to automatically retract paper in a container after each printed and waiting to takeout document
<i>offOn</i>	'0' - off; '1' - on

**1.1.2.20 config\_Set\_AutomaticZReport()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_AutomaticZReport (
    string option,
    string offOn,
    string optZRep )
```

Enable / disable automatic Z-report printing at 23:59:59 o'clock if issuance of a fiscal or service entry for the day.

**Parameters**

<i>option</i>	"H" - Enable / disable automatic Z-report printing at 23:59:59 o'clock if issuance of a fiscal or service entry for the day.
<i>offOn</i>	One byte with value of '1' (enabled) or '0' (disabled).

**Parameters**

<i>optZRep</i>	<p>'p': The Z-report is printed (otherwise it is only recorded in the KLEN);</p> <p>'a': Z-report contains data for sales of articles / fuels;</p> <p>'d': The Z-report contains departments data;</p> <p>'N': The Z-report does not reset the operator data;</p> <p>'A': The Z-report does not reset the item data;</p> <p>'D': The Z-report does not reset the departments data.</p>
----------------	--

**1.1.2.21 config\_Set\_BarcodeHeight()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_BarcodeHeight (
    string option,
    string value )
```

Set barcode height

**Parameters**

<i>option</i>	'B'
<i>value</i>	Number setting the barcode height in pixels

**1.1.2.22 config\_Set\_CashDrawerPulse()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_CashDrawerPulse (
    string option,
    string offOn )
```

Set cash drawer pulse

**Parameters**

<i>option</i>	'X' - one byte for cash drawer pulse
<i>offOn</i>	'0' - off; '1' - on

**1.1.2.23 config\_Set\_DateTime()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_DateTime (
    string dateTime )
```

Set date and time

**Parameters**

<i>dateTime</i>	In format DD-MM-YY HH:MM[:SS]
-----------------	-------------------------------

**1.1.2.24 config\_Set\_DepartmentName()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_DepartmentName (
    string departmentNumber,
    string taxGroup,
    string textRow1 )
```

Program a department name

**Parameters**

<i>departmentNumber</i>	Department number. Integer between 1 and 1200.
<i>taxGroup</i>	Tax group associated with the department.
<i>textRow1</i>	Department name or descriptive text. Up to 28 characters.

**1.1.2.25 config\_Set\_DepartmentNameTwoRows()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_DepartmentNameTwoRows (
    string departmentNumber,
    string taxGroup,
    string textRow1,
    string textRow2 )
```

Program a department name for two lines

**Parameters**

<i>departmentNumber</i>	Department number. Integer between 1 and 1200.
<i>taxGroup</i>	Tax group associated with the department.
<i>textRow1</i>	Department name or descriptive text. Up to 28 characters.
<i>textRow2</i>	Department name or descriptive text—second line. Optional parameter: up to 34 characters

**1.1.2.26 config\_Set\_ExtraSpace()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_ExtraSpace (
    string option,
    string value )
```

Set extra space

**Parameters**

<i>option</i>	'R' - for additional letter spacing in pixels
<i>value</i>	Values are between 0 and 4

**1.1.2.27 config\_Set\_FeedButton()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_FeedButton (
    string option,
    string offOn )
```

Disable/enable the FEED button functionality

**Parameters**

<i>option</i>	'c' - one byte for feed button functionality
<i>offOn</i>	'0' - off; '1' - on

**1.1.2.28 config\_Set\_FontSize()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_FontSize (
    string option,
    string value )
```

Set font size

**Parameters**

<i>option</i>	'F' - printer font
<i>value</i>	'0' - 32 dots (4 mm) height, higher letters; '1' - 32 dots(4 mm) height, normal letters; '2' - 24 dots(3 mm) height

**1.1.2.29 config\_Set\_FooterLine()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_FooterLine (
    string item,
    string value )
```

Set footer lines

**Parameters**

<i>item</i>	Between '6' and '7'
<i>value</i>	Text up to 48 symbols

**1.1.2.30 config\_Set\_HeaderLine()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_HeaderLine (
    string item,
    string value )
```

Set header line

**Parameters**

<i>item</i>	Between '0' and '5' for header lines
<i>value</i>	Text up to 48 symbols

**1.1.2.31 config\_Set\_InvoiceRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_InvoiceRange (
    string startValue,
    string endValue )
```

Set invoice range

**Parameters**

<i>startValue</i>	Sets the range start value. Integer up to 10 digits
<i>endValue</i>	Sets the range end value. Integer up to 10 digits

**Returns**

Dictionary with keys:

valueStart	The starting number of the interval. Max 10 digits (1...9999999999).
valueEnd	The ending number of the interval. Max 10 digits (1...9999999999).
valueCurrent	The current invoice receipt number (1...9999999999).

**1.1.2.32 config\_Set\_LANSignal()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_LANSignal (
```

```
string option,
string offOn )
```

Enable / disable the sound signal when attaching / disconnecting to the device on TCP / IP.

#### Parameters

<i>option</i>	"M" - Enable / disable the sound signal when attaching / disconnecting to the device on TCP / IP.
<i>offOn</i>	One byte with value of '1' (enabled) or '0' (disabled).

#### 1.1.2.33 config\_Set\_Logo()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_Logo (
    string rowNum,
    string data )
```

Program graphic logo

#### Parameters

<i>rowNum</i>	The line being programmed. Number between 0 and 95.
<i>data</i>	Graphic data. Set as hexadecimals, two characters for each byte of information. The data length is up to 72 bytes.If it is less than that, the data are automatically padded with 00.

#### 1.1.2.34 config\_Set\_OperatorName()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_OperatorName (
    string operatorCode,
    string password,
    string operatorName )
```

Set operator's name

#### Parameters

<i>operatorCode</i>	Operator code. Between 1 and 16.
<i>password</i>	Password (between 4 and 8 digits).
<i>operatorName</i>	Operator name (up to 24 characters).

#### 1.1.2.35 config\_Set\_OperatorPassword()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_OperatorPassword (
    string operatorCode,
```



```
string oldPassword,
string newPassword )
```

Set operator's password

#### Parameters

<i>operatorCode</i>	Operator code. Between 1 and 16.
<i>oldPassword</i>	Old password (between 4 and 8 digits).
<i>newPassword</i>	New password (between 4 and 8 digits).

#### 1.1.2.36 config\_Set\_OptionalErrorText()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_OptionalErrorText (
    string option,
    string offOn )
```

Disable/enable returning optional text in the command response when an error occurs

#### Parameters

<i>option</i>	'd' - one byte optional error text
<i>offOn</i>	'0' - off; '1' - on

#### 1.1.2.37 config\_Set\_PrintDensity()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_PrintDensity (
    string option,
    string value )
```

Set print density

#### Parameters

<i>option</i>	'D' - for print density
<i>value</i>	Between 1 and 5

#### 1.1.2.38 config\_Set\_PrintDepartment()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_PrintDepartment (
    string option,
    string offOn )
```

Set print department

**Parameters**

<i>option</i>	'N' - one byte for print department
<i>offOn</i>	'0' - disabled; '1' - enabled

**1.1.2.39 config\_Set\_PrinterReadinessPrintMsgs()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_PrinterReadinessPrintMsgs (
    string option,
    string offOn )
```

Enables / disables the printer's readiness print message.

**Parameters**

<i>option</i>	"G" - Enables / disables the printer's readiness print message
<i>offOn</i>	One byte with value of '1' (enabled) or '0' (disabled).

**1.1.2.40 config\_Set\_PrintLogo()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_PrintLogo (
    string option,
    string offOn,
    string height )
```

Enable/disable printing the logo graphic right before the HEADER

**Parameters**

<i>option</i>	'L' - for printing graphic logo
<i>offOn</i>	One byte with value of '1' (enabled) or '0' (disabled).
<i>height</i>	Height

**1.1.2.41 config\_Set\_PrintTaxDDS()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_PrintTaxDDS (
    string option,
    string offOn )
```

Enable/disable print tax

**Parameters**

<i>option</i>	'T' - accumulated VAT for the receipt in a normal (not extended) fiscal receipt.
<i>offOn</i>	'0' - off; '1' - on

**1.1.2.42 config\_Set\_RollUpPaper()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_RollUpPaper (
    string option,
    string offOn )
```

Enable / disable automatic roll-up of paper in a container after each document. After turning on the printer, the printer does not pick up the paper in a container. ONLY FOR PRINTERS SK1-21F AND SK1-31F.

**Parameters**

<i>option</i>	'P' - Enable / disable automatic roll-up of paper in a container after each document. After turning on the printer, the printer does not pick up the paper in a container
<i>offOn</i>	'0' - off; '1' - on

**1.1.2.43 config\_Set\_SIM()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_SIM (
    string option,
    string indexValue )
```

Set SIM

**Parameters**

<i>option</i>	'SIM' - Selects the SIM card to be used by the GPRS modem
<i>indexValue</i>	The value can be either 0 or 1

**1.1.2.44 config\_Set\_Switches()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_Switches (
    string switches )
```

Store the settings and the switches in the flash memory

**Parameters**

<i>switches</i>	8 or 16 bytes with values of '0' or '1': the configuration 'switches'.
-----------------	--

**Returns**

Dictionary with key:

<i>errorCode</i>	"P" - successful recording; "F" - unsuccessful recording of the settings
------------------	--

**1.1.2.45 config\_Set\_UNPPrinting()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_UNPPrinting (
    string option,
    string offOn )
```

Enable / disable UNP printing of fiscal and reversal receipts.

**Parameters**

<i>option</i>	"K" - Enables / disables the printer's readiness print message
<i>offOn</i>	One byte with value of '1' (enabled) or '0' (disabled).

**1.1.2.46 config\_Set\_WarningTimeRange()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_WarningTimeRange (
    string option,
    string warnTimeRec,
    string warnTimeGInfo )
```

Set warning time range

**Parameters**

<i>option</i>	'W' - Sets the time to trigger the warning for unsent data on payment documents and not updated data on the level measuring system
<i>warnTimeRec</i>	Time in minutes, between 10 and 1440 or 0. If it is zero, there is no warning
<i>warnTimeGInfo</i>	Time in minutes, between 10 and 1440 or 0. If it is zero, there is no warning

**1.1.2.47 config\_Set\_WorkingWithPinpad()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.config_Set_WorkingWithPinpad (
```

```
string enable )
```

Set fiscal device to work with a Datecs payment terminal

#### Parameters

<i>enable</i>	0 - disabled, 1 - enabled
---------------	---------------------------

#### 1.1.2.48 config\_SetLAN()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.config_SetLAN (
    string ipAddress,
    string subnetMask,
    string portNumber,
    string defaultGateway,
    string dHCP,
    string mACAddress )
```

Set LAN settings

#### Parameters

<i>ipAddress</i>	4 numbers, between 0 and 255, separated by a dot, representing the IP address of the device
<i>subnetMask</i>	4 numbers, between 0 and 255, separated by a dot, representing the IP address of the device
<i>portNumber</i>	Number, between 1 and 65535, representing the IP port of the device
<i>defaultGateway</i>	4 numbers, between 0 and 255, separated by a dot, representing the Default Gateway address of the device
<i>dHCP</i>	One character: "0" or "1". Enables or disables DHCP (Receiving automatic LAN settings from the server).
<i>mACAddress</i>	Up to 12 hexadecimal characters representing the MAC address of the device. Works only if the service jumper is present!!!

#### Returns

Dictionary with key:

fAnswer	Data for LAN
---------	--------------

#### 1.1.2.49 display\_Clear()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.display_Clear ( )
```

Clear display

**1.1.2.50 display\_Show\_DateTime()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.display_Show_DateTime ( )
```

Display date and time

**1.1.2.51 display\_Show\_LowerLine()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.display_Show_LowerLine (
    string textData )
```

Show text on lower line of the display

**Parameters**

<i>textData</i>	Text, up to 20 symbols,sent directly to the display
-----------------	---

**1.1.2.52 display\_Show\_Text()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.display_Show_Text (
    string textData )
```

Show text on the display

**Parameters**

<i>textData</i>	Text, up to 40 symbols, sent to display
-----------------	---

**1.1.2.53 Get\_SBit\_State()**

```
bool FiscalPrinterSDK.FDGROUP_A_BGR.Get_SBit_State (
    int byteIndex,
    int bitIndex )
```

Get status bit state

**Parameters**

<i>byteIndex</i>	byte index
<i>bitIndex</i>	bit index

**Returns**

Returns current state bool(1 - bit is raised, 0 - bit is not raised)

**1.1.2.54 Get\_sBytesBitInUse()**

```
bool FiscalPrinterSDK.FDGROUP_A_BGR.Get_sBytesBitInUse (
    int byteIndex,
    int bitIndex )
```

Gets status bits in use

**Parameters**

<i>byteIndex</i>	the index of the byte
<i>bitIndex</i>	the index of the bit (go figure)

**Returns**

Returns TRUE if the status bit is in use and FALSE otherwise.

**1.1.2.55 info\_ClearLastErrorExtendedInfo()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_ClearLastErrorExtendedInfo ( )
```

Clear extended information about last error

**Returns**

Dictionary with keys:

cmd	Last error command number.
errorCode	Number: error code.
dateTime	Error date and time in the DD-MM-YY hh:mm:ss format.
errorText	Text description of the last error.

**1.1.2.56 info\_Get\_ActiveTaxes()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_ActiveTaxes (
    string closureNumber,
    string option )
```

Get active taxes



**Parameters**

<i>closureNumber</i>	Fiscal record number
<i>option</i>	'0' - Information on the active tax rates for the Z-report record in question

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
taxRecordNumber	Tax record number.
decimalsCount	Decimal point count for Z report record .
enabled	Enabled tax rates (1 - enabled).
taxRateA	tax rate A in percents.
taxRateB	tax rate B in percents.
taxRateC	tax rate C in percents.
taxRateD	tax rate D in percents.
taxRateE	tax rate E in percents.
taxRateF	tax rate F in percents.
taxRateG	tax rate G in percents.
taxRateH	tax rate H in percents.
dateTime	Date and time (format: DD-MM-YY hh:mm:ss).

**1.1.2.57 info\_Get\_AdditionalDailyInfo\_01()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_AdditionalDailyInfo_01 (
    string option )
```

Information of current sums for the day

**Parameters**

<i>option</i>	'0' - total sales; '1' - Accumulated VAT
---------------	--

**Returns**

Dictionary with keys:

taxA	Turnover for tax group A
taxB	Turnover for tax group B
taxC	Turnover for tax group C
taxD	Turnover for tax group D
taxE	Turnover for tax group E
taxF	Turnover for tax froup F
taxG	Turnover for tax froup G
taxH	Turnover for tax froup H

**1.1.2.58 info\_Get\_AdditionalDailyInfo\_02()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_AdditionalDailyInfo_02 ( )
```

Get additional daily information

**Returns**

Dictionary with keys:

cash	Payed in cash
credit	Payed with credit
debit	Payed with debit
cheque	Payed by check
payment01	Paid by additional payment types 1
payment02	Paid by additional payment types 2
payment03	Paid by additional payment types 3
payment04	Paid by additional payment types 4
closure	Current fiscal closure
nextFReceiptNumber	Next fiscal receipt number

**1.1.2.59 info\_Get\_AdditionalDailyInfo\_03()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_AdditionalDailyInfo_03 ( )
```

Get additional daily information - extended

**Returns**

Dictionary with keys:

cash	Payed in cash
credit	Payed with credit
debit	Payed with debit
cheque	Payed by check
payment01	Paid by additional payment types 1
payment02	Paid by additional payment types 2
payment03	Paid by additional payment types 3
payment04	Paid by additional payment types 4
payment05	Paid by additional payment types 5
payment06	Paid by additional payment types 6
payment07	Paid by additional payment types 7
payment08	Paid by additional payment types 8
payment09	Paid by additional payment types 9
payment10	Paid by additional payment types 10
payment11	Paid by additional payment types 11
closure	Current fiscal closure
nextFReceiptNumber	Next fiscal receipt number

**1.1.2.60 info\_Get\_AdditionalPaymentNames()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_AdditionalPaymentNames (
    string option )
```

Get additional payment names

**Parameters**

<i>option</i>	'I' Additional payment 1; 'J' Additional payment 2; 'K' Additional payment 3; 'L' Additional payment 4. 'i' Additional payment 1. Identical to 'I'; 'j' Additional payment 2. Identical to 'J'; 'k' Additional payment 3. Identical to 'K'; 'l' Additional payment 4. Identical to 'L'; 'm' Additional payment 5; 'n' Additional payment 6; 'o' Additional payment 7; 'p' Additional payment 8 'q' Additional payment 9; 'r' Additional payment 10; 's' Additional payment 11
---------------	--

**Returns**

Dictionary with key:

paymentName	Payment name (up to 24 symbols)
-------------	---------------------------------

**1.1.2.61 info\_Get\_AdditionalPayments()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_AdditionalPayments (
    string closure1,
    string option,
    string closure2 )
```

Information on the amounts by payment type for the specified record or period

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'7' - Information on the amounts by payment type for the specified record or period
<i>closure2</i>	Fiscal memory record number

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
cashpaymentSum	Paid in cash.
debitcardpaymentSum	Paid by debit card.
creditcardpaymentSum	Paid by credit card.
checkpaymentSum	Paid by check.

payment01Sum	Additional payment type 1.
payment02Sum	Additional payment type 2.
payment03Sum	Additional payment type 3.
payment04Sum	Additional payment type 4.
payment05Sum	Additional payment type 5.
payment06Sum	Additional payment type 6.
payment07Sum	Additional payment type 7.
payment08Sum	Additional payment type 8.
payment09Sum	Additional payment type 9.
payment10Sum	Additional payment type 10.
payment11Sum	Additional payment type 11.

#### 1.1.2.62 info\_Get\_CashDrawerStatusOrOpenShift()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_CashDrawerStatusOrOpenShift
(
    string option )
```

If option is 'D' - Returns cash drawer status (0 - closed drawer, 1 - open drawer) for models that support it(FP-2000 or FP-700) if supported by the connected drawer. If option is 'B' - Returns the status of open shift lock

##### Parameters

<i>option</i>	'D' - Returns cash drawer status (0 - closed drawer, 1 - open drawer) for models that support it(FP-2000 or FP-700) if supported by the connected drawer.'B' - Returns the status of open shift lock
---------------	--

##### Returns

Dictionary with key:

status	Returns the status of the drawer (0 - closed drawer, 1 - open drawer) or for option 'B' - Returns 0 - not blocked, 1 - blocked
--------	--

#### 1.1.2.63 info\_Get\_CashIn\_CashOut()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_CashIn_CashOut ( )
```

Get cash in/ cash out

##### Returns

Dictionary with keys:

errorCode	'P' - pass or 'F' - failed
cashSum	Cash in safe sum
servIn	Total sum of cash in operations
servOut	Total sum of cash out operations

**1.1.2.64 info\_Get\_CashInCashOut()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_CashInCashOut (
    string closure1,
    string option,
    string closure2 )
```

Get cash in/ cash out

**Parameters**

<i>closure1</i>	Fiscal record number
<i>option</i>	'9' - Information on the amounts by payment type for the specified record or period
<i>closure2</i>	Fiscal record number

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
cashinCount	Number of cash in operations.
cashinSum	Accumulated cash in sum.
cashoutCount	Number of cash out operations.
cashoutSum	Accumulated cash out sum.

**1.1.2.65 info\_Get\_Corrections()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_Corrections (
    string closure1,
    string option,
    string closure2 )
```

Information on the amounts by payment type for the specified record or period

**Parameters**

<i>closure1</i>	Fiscal record number
<i>option</i>	'8' - Information on the amounts by payment type for the specified record or period
<i>closure2</i>	Fiscal record number

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
discountsCount	Number of discounts.

discountsSum	Accumulated discount sum.
surchargesCount	Number of surcharges.
surchargesSum	Accumulated surcharges sum.
voidedCount	Number of voided operations.
voidedSum	Accumulated voided sum.
canceledCount	Number of canceled receipts.
canceledSum	Accumulated canceled sum.

#### 1.1.2.66 info\_Get\_CurrentRecieptInfo()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_CurrentRecieptInfo ( )
```

Get current receipt information

##### Returns

Dictionary with keys:

canVd	Is it possible return (strono)
taxA	Tax A accumulated sum
taxB	Tax B accumulated sum
taxC	Tax C accumulated sum
taxD	Tax D accumulated sum
taxE	Tax E accumulated sum
taxF	Tax F accumulated sum
taxG	Tax G accumulated sum
taxH	Tax H accumulated sum
inv	Is it invoice open? '0' - No; '1' - Yes
invNum	Next invoice number

#### 1.1.2.67 info\_Get\_DailyCorrections()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_DailyCorrections ( )
```

Get daily corrections

##### Returns

Dictionary with keys:

countDiscounts	Number of discounts after the last Z-report
sumDiscounts	Sum of discounts after the last Z-report
countSurcharges	Number of surcharges after the last Z-report.
sumSurcharges	Sum of surcharges after the last Z-report

**1.1.2.68 info\_Get\_DataForLastIssuedReceipt() [1/2]**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_DataForLastIssuedReceipt (
)
```

Finds the last issued fiscal or reversal receipt before a daily report with clearing is made /ONLY FOR TYPE 21/

**Returns**

Dictionary with keys:

errorParam	'P' - pass; 'F' - failed; '*' - End of receipt + first parameter of the answer
data	Returned data.

**1.1.2.69 info\_Get\_DataForLastIssuedReceipt() [2/2]**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_DataForLastIssuedReceipt (
    string docNum )
```

Finds a document with document number only if it is an electronic document. /ONLY FOR TYPE 21/

**Parameters**

<i>docNum</i>	Document number - only if it is electronic document.
---------------	--

**Returns**

Dictionary with keys:

errorParam	'P' - pass; 'F' - failed; '*' - End of receipt + first parameter of the answer
data	Returned data.

**1.1.2.70 info\_Get\_DateTime()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_DateTime ( )
```

Get date and time

**Returns**

Dictionary with key:

dateTime	Date and time in format: DD-MM-YY HH:MM:SS
----------	--

### 1.1.2.71 info\_Get\_DateTime\_01()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_DateTime_01 ( )
```

Get date and time separated

#### Returns

Dictionary with keys:

day	Day - DD
month	Month - MM
year	Year - YY
hour	Hours - HH
minute	Minutes - mm
second	Seconds - ss

### 1.1.2.72 info\_Get\_DecimalAndTaxRates()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_DecimalAndTaxRates ( )
```

Get decimals and tax rates

#### Returns

Dictionary with keys:

multiplier	One byte, '1' - if receipt is open, '0' - otherwise
decimals	One byte (0...2), digits after decimal point
currencyName	Currency name (3 bytes)
enabledMask	Eight bytes, with 0 or 1, for enabled or disabled tax group
taxA	Tax A value
taxB	Tax B value
taxC	Tax C value
taxD	Tax D value
taxE	Tax E value
taxF	Tax F value
taxG	Tax G value
taxH	Tax H value

### 1.1.2.73 info\_Get\_DepartmentInfo()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_DepartmentInfo (
    string departmentNumber )
```



Get department information

#### Parameters

<i>departmentNumber</i>	Department number. Integer between 1 and 1200. Value 0 returns data on the sales made without department specified. In this case, the tax group is missing.
-------------------------	---

#### Returns

Dictionary with keys:

errorCode	'P' - pass, 'F' - failed
taxGroup	Tax group for department ( - )
recSales	Sales count for departments in receipt.
recSum	Sum for current or last fiscal receipt for department.
totSales	Sales count for department for the day.
totSum	Day sum by department.
line1	Text for department.
line2	Text for department line 2.

#### 1.1.2.74 info\_Get\_DiagnosticInfo()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_DiagnosticInfo ( )
```

Get diagnostic information

#### Returns

Dictionary with keys:

deviceName	Device name
firmware	Firmware version
checkSum	Checksum.
switches	Not in fiscal device
serialNumber	Serial number of device.
fiscalMemoryNumber	Fiscal memory number.

#### 1.1.2.75 info\_Get\_EIK()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_EIK ( )
```

Read UIC

#### Returns

Dictionary with keys:

taxA	Tax rate A
taxB	Tax rate B

#### 1.1.2.76 info\_Get\_FMRecord\_LastDateTime()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_FMRecord_LastDateTime (
    string option )
```

Read date of last fiscal memory record

##### Parameters

<i>option</i>	Blank or 'T' - If present, also returns the time of the last record
---------------	---

##### Returns

Dictionary with key:

dateTime	Format: DD-MM-YYYY or DD-MM-YYYY hh:mm:ss
----------	---

#### 1.1.2.77 info\_Get\_FMRecord\_LastDateTime\_01()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_FMRecord_LastDateTime_01 (
    string option )
```

Get date and time from last FM records (separated)

##### Parameters

<i>option</i>	Blank or 'T' - If present, also returns the time of the last record
---------------	---

##### Returns

Dictionary with keys:

day	Day - DD
month	Month - MM
year	Year - YY
hour	Hours - HH
minute	Minutes - mm
second	Seconds - ss

**1.1.2.78 info\_Get\_FreeFMRecords()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_FreeFMRecords ( )
```

Get free fiscal memory records

**Returns**

Dictionary with keys:

countLogical	Count of logical places for fiscal memory
countTotal	Count of logical places for fiscal memory (repeats the above record)

**1.1.2.79 info\_Get\_FTransactionStatus()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_FTransactionStatus (
    string option )
```

**Parameters**

<i>option</i>	" (the command will return the information on the current status of the bill payable by the customer until now) or leave blank
---------------	--

**Returns**

Dictionary with keys:

open	One byte, '1' - if receipt is open, '0' - otherwise
items	Sales count from current or last fiscal receipt (4 bytes)
amount	Sum of the last fiscal receipt
tender	Sum, payed from next or last receipt

**1.1.2.80 info\_Get\_InvoiceRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_InvoiceRange ( )
```

Get invoice range

**Returns**

Dictionary with keys:

valueStart	Start value of invoice range
valueEnd	End value of invoice range
valueCurrent	Current invoice value

**1.1.2.81 info\_Get\_LANSettings()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_LANSettings ( )
```

Get last settings

**Returns**

Dictionary with keys:

iPAddress	4 numbers, between 0 and 255, separated by a dot, representing the IP address of the device.
subnetMask	4 numbers, between 0 and 255, separated by a dot, representing the Subnet mask of the device.
portNumber	Number, between 1 and 65535, representing the IP port of the device.
defaultGateway	4 numbers, between 0 and 255, separated by a dot, representing the Default Gateway address of the device.
dHCP	One character: "0" or "1". Enables or disables DHCP (Receiving automatic LAN settings from the server).
macAddress	Up to 12 hexadecimal characters representing the MAC address of the device. Works only if the service jumper is present!

**1.1.2.82 info\_Get\_LastDocumentNumber()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_LastDocumentNumber ( )
```

Get last document number

**Returns**

Dictionary with key:

documentNumber	Last issued document number (7 digits).
----------------	---

**1.1.2.83 info\_Get\_LastFiscRecord()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_LastFiscRecord ( )
```

Get last fiscal record

**Returns**

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed
lastFRecordNumber	Last fiscal record
taxA	Turnover for tax group A
taxB	Turnover for tax group B
taxC	Turnover for tax group C
taxD	Turnover for tax group D
taxE	Turnover for tax group E
taxF	Turnover for tax froup F
taxG	Turnover for tax froup G
taxH	Turnover for tax froup H
date	Fiscal record date /DDMMYY/

#### 1.1.2.84 info\_Get\_LastFiscRecord\_01()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_LastFiscRecord_01 ( )
```

Get last fiscal record with time

##### Returns

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed
lastFRecordNumber	Last fiscal record
taxA	Turnover for tax group A
taxB	Turnover for tax group B
taxC	Turnover for tax group C
taxD	Turnover for tax group D
taxE	Turnover for tax group E
taxF	Turnover for tax froup F
taxG	Turnover for tax froup G
taxH	Turnover for tax froup H
date	Fiscal record date /DDMMYY/

#### 1.1.2.85 info\_Get\_LogoLine()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_LogoLine (
    string rowNum )
```

Get logo data from device by line number.

##### Parameters

<i>rowNum</i>	The line being programmed. Number between 0 and 95.
---------------	---

**Returns**

Dictionary with key:

fRow	Data in a row
------	---------------

**1.1.2.86 info\_Get\_NettoSums()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NettoSums (
    string closure1,
    string option,
    string closure2 )
```

Information on the net amounts for the specified record or period

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'2' - Information on the net amounts for the specified record or period
<i>closure2</i>	Fiscal memory record number

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
closureNumber	Closure number for given period.
receiptsCount	Decimal point count for Z report record .
nettoTaxA	Netto amount for tax group A.
nettoTaxB	Turnover for tax group B.
nettoTaxC	Turnover for tax group C.
nettoTaxD	Turnover for tax group D.
nettoTaxE	Turnover for tax group E.
nettoTaxF	Turnover for tax group F.
nettoTaxG	Turnover for tax group G.
nettoTaxH	Turnover for tax group H.

**1.1.2.87 info\_Get\_NotPrintedRowsCount()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NotPrintedRowsCount (
    string option )
```

Get not printed rows count

**Parameters**

<i>option</i>	'L' - Returns the number of lines remaining to be printed
---------------	---

**Returns**

Dictionary with key:

nLines	Number of pending lines in the print buffer. A value of 0 means that there are no pending print data.
--------	---

**1.1.2.88 info\_Get\_NRADData\_Type1()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRADData_Type1 ( )
```

Read NRA data type 1

**Returns**

Dictionary with keys:

errorCode	P if command passes, F - otherwise.
dateTime	Date and time in the "DD-MM-YYYY hh:mm:ss" format.
closure	Next daily report number.
fiscRec	Number of fiscal receipts (customers) for the day.
lastFiscal	Global number of the last issued fiscal receipt.
lastDoc	Global number of the last issued document.
journal	Current journal number.

**1.1.2.89 info\_Get\_NRADData\_Type2()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRADData_Type2 ( )
```

NRA data type 2 - Sales for tax groups for the day

**Returns**

Dictionary with keys:

errorCode	P if command passes, F - otherwise.
s1	Sales for a tax group 1, for the day.
s2	Sales for a tax group 2, for the day.
s3	Sales for a tax group 3, for the day.
s4	Sales for a tax group 4, for the day.
s5	Sales for a tax group 5, for the day.
s6	Sales for a tax group 6, for the day.

s7	Sales for a tax group 7, for the day.
s8	Sales for a tax group 8, for the day.
total	Total sales for the day.
gTotal	Total lifetime sales.

#### 1.1.2.90 info\_Get\_NRADData\_Type3()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRADData_Type3 ( )
```

Read NRA type 3

##### Returns

Dictionary with keys:

errorCode	P if command passes, F - otherwise.
discountN	Number of discounts for the day.
discountS	Total amount of discounts for the day.
markUpN	Number of mark-ups for the day.
markUpS	Total amount of mark-ups for the day.
voidN	Number of adjustments for the day.
voidS	Total amount of adjustments for the day.
cancelN	Number of cancelled receipts for the day.
cancelS	Total amount of cancelled receipts for the day.

#### 1.1.2.91 info\_Get\_NRADData\_Type4()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRADData_Type4 ( )
```

Read NRA type 4

##### Returns

Dictionary with keys:

errorCode	P if command passes, F - otherwise.
cashInN	Number of service deposit operations for the day.
cashInS	Total amount of service deposits for the day.
cashOutN	Number of service withdrawal operations for the day.
cashOutS	Total amount of service withdrawals for the day.
cashSum	Available cash.



**1.1.2.92 info\_Get\_NRAData\_Type5()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRAData_Type5 ( )
```

Read NRA type 5

**Returns**

Dictionary with keys:

errorCode	P if command passes, F - otherwise.
cashPaid	Paid in cash.
credPaid	Paid by credit card.
cardPaid	Paid by debit card.
checkPaid	Paid by check.
paid1	Paid by programmed payment type 1.
paid2	Paid by programmed payment type 2.
paid3	Paid by programmed payment type 3.
paid4	Paid by programmed payment type 4.

**1.1.2.93 info\_Get\_NRAData\_Type6()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRAData_Type6 (
    string closure )
```

Read NRA type 6

**Parameters**

<i>closure</i>	Daily report number
----------------	---------------------

**Returns**

Dictionary with keys:

errorCode	P if command passes, F - otherwise.
dt	Daily report date and time in the "DD-MM-YYYY hh:mm:ss" format.

**1.1.2.94 info\_Get\_NRAData\_Type7()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRAData_Type7 ( )
```

Read NRA type 7

**Returns**

Dictionary with keys:

errorCode	P if command passes, F - otherwise.
dt	Returns the date and time of fiscal registration and time in the "DD-MM-YYYY hh:mm:ss" format.

#### 1.1.2.95 info\_Get\_NRAData\_Type8()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRAData_Type8 (
    string option )
```

Read NRA type 8 - Returns PLU sales data by PLU codes in ascending order

##### Parameters

<i>option</i>	"F" - for the first time, and then "N" until read all and printer returns "F" errorCode
---------------	---

##### Returns

Dictionary with keys:

errorCode	P if command passes, F - otherwise.
PLUcode	PLU code.
PLUtotal	Accumulated sales sum for the PLU.
PLUname	PLU name.

#### 1.1.2.96 info\_Get\_NRADocuments()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_NRADocuments (
    string option )
```

Returns data on the customer documents sent to the NRA server

##### Parameters

<i>option</i>	'R' - returns data on the customer documents sent to the NRA server
---------------	---

##### Returns

Dictionary with keys:

lastPrintDoc	Number of the last printed document.
nLastSentDoc	Number of the last successfully sent document.
dTLastSentDoc	Date and time of the last successfully sent document.
minFromLastSuccessSent	Minutes since the last successfully sent document.
nFirstNotSentDoc	Number of the first unsent document.

dTFirstNotSentDoc	Date and time of the first unsent document.
minFromFirstNotSuccessSent	Minutes since the first unsent document.

### 1.1.2.97 info\_Get\_Numbers()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_Numbers (
    string closure1,
    string option )
```

Additional information on the specified record

#### Parameters

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'4' - Additional information on the specified record

#### Returns

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
closureNumber	Closure number for given period.
taxRecordNumber	Last active tax record number for given period.
resetRecordNumber	Last RAM reset to this fiscal block.
kLENNumber	Number of electronic journal for this fiscal block.
dateTime	Date and time for the data (format: DD-MM-YY hh:mm:ss).

### 1.1.2.98 info\_Get\_OperatorsData()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_OperatorsData (
    string wpOperator )
```

Get operator's data

#### Parameters

<i>wpOperator</i>	Operator ID (between 1 and 16).
-------------------	---------------------------------

#### Returns

Dictionary with keys:

receiptsCount	Receipts count from selected operator
---------------	---------------------------------------

salesCount	Sales count
salesSum	Sales sum
discountsCount	Discounts count
discountsSum	Discounts sum
surchargesCount	Surcharges count
surchargesSum	Surcharges sum
voidsCount	Voids count
voidsSum	Void sum
operatorName	Operator's name
operatorPassword	Operator's password

### 1.1.2.99 info\_Get\_PrintOption()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_PrintOption (
    string option,
    string targetOption )
```

Get print options

#### Parameters

<i>option</i>	'I' - get info
<i>targetOption</i>	All types of command 43 - 'A','B','C','D','E','F','L' ...

#### Returns

Dictionary with key:

fAnswer	Data for selected option
---------	--------------------------

### 1.1.2.100 info\_Get\_ProgrammedFuel()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_ProgrammedFuel (
    string closure1,
    string option )
```

ONLY for type 3 and 31 printers! Information on the programmed fuel FTT and names for this Z-report. Only for type 3 and 31 printers.

#### Parameters

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'11' - Information on the programmed fuel FTT and names for this Z-report. Only for type 3 and 31 printers.

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
data	Data in format: Programmed fuel FTT1,Programmed fuel name ... FFTn,Fuel name n

**1.1.2.101 info\_Get\_ProgrammedPumps()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_ProgrammedPumps (
    string closure1,
    string option )
```

Information on the programmed pumps for this Z-report. Only for type 31 printers.

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'13' - Information on the programmed pumps for this Z-report. Only for type 31 printers.

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
data	Data in format: Disp1 (Programmed pump number), ... ,Disp n

**1.1.2.102 info\_Get\_ProgrammedPumps2()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_ProgrammedPumps2 (
    string closure1,
    string option )
```

Information on the programmed pumps for this Z-report -0 - Tank volume at current temperature. Vol15Ci Tank volume at 15 degrees.. Only for type 31 printers

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'14' - Information on the programmed pumps for this Z-report -0 - Tank volume at current temperature. Vol15Ci Tank volume at 15 degrees

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
data	Data in format: Term1(Programmed tank number),VolAct1(Tank volume at current temperature),Vol15C1(Tank volume at 15 degrees),FL1(Tank fuel level),Tmpr1(Tank fuel temperature),...

### 1.1.2.103 info\_Get\_ProgrammedPumps3()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_ProgrammedPumps3 (
    string closure1,
    string option,
    string pumpNum )
```

Information on a programmed pump for this Z-report. ONLY for type 31 printers - Nozzle number, Nozzle counter value

#### Parameters

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'15' - Information on a programmed pump for this Z-report.
<i>pumpNum</i>	Pump number.

#### Returns

### 1.1.2.104 info\_Get\_RAMResetDateTime()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_RAMResetDateTime (
    string closure1,
    string option )
```

Information on the specified FM record with RAM reset

#### Parameters

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'6' - Information on the specified FM record with RAM reset

#### Returns

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
dateTime	Date and time (format: DD-MM-YY hh:mm:ss).

**1.1.2.105 info\_Get\_SoldQuant\_AccumulatedSales()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_SoldQuant_AccumulatedSales
(
    string closure1,
    string option )
```

Information on the sold quantities and accumulated sales for the programmed fuels for this Z-report. Only for type 3 and 31 printers.

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'12' - Information on the sold quantities and accumulated sales for the programmed fuels for this Z-report.

**Returns**

Dictionary with keys:

<i>errorCode</i>	'P' - Command is executed correctly, 'F' - Command is declined.
<i>data</i>	Data in format: Programmed fuel FTT1,Vol1(fuel sales volume),S1(fuel sales amount) ... F↔ FTn,Voln,Sn

**1.1.2.106 info\_Get\_StatusBytes()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_StatusBytes (
    string option )
```

Depends on option you choose: 'W' - First, waits for the printer to finish printing everything in the print buffer or 'X' - Does not wait for the printer, but responds immediately

**Parameters**

<i>option</i>	'W' - First, waits for the printer to finish printing everything in the print buffer. 'X' - Does not wait for the printer, but responds immediately.
---------------	--

**Returns**

Dictionary with key:

statusBytes	status bytes
-------------	--------------

**1.1.2.107 info\_Get\_Storno\_NettoSums()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_Storno_NettoSums (
    string closure1,
    string option,
    string closure2 )
```

Information on the storno netto sums for the specified record or period

**Parameters**

<i>closure1</i>	Fiscal memory record number.
<i>option</i>	'2*' - Information on the netto sums for the specified record or period for STORNO.
<i>closure2</i>	Fiscal record number

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
closureCnt	Closure count for given period.
refundStornoCnt	Count of STORNO receipts - type REFUND.
operatorErrStornoCnt	Count of STORNO receipts - type OPERATOR'S ERROR.
taxBaseStornoCnt	Count of STORNO receipts - type TAX BASE REDUCTION.
refundSalesCnt	Sales count of STORNO receipts type REFUND.
operatorErrSalesCnt	Sales count of STORNO receipts type OPERATOR'S ERROR.
taxBaseSalesCnt	Sales count of STORNO receipts type TAX BASE REDUCTION.
nettoTaxA	Netto sum for tax group A.
nettoTaxB	Netto sum for tax group B.
nettoTaxC	Netto sum for tax group C.
nettoTaxD	Netto sum for tax group D.
nettoTaxE	Netto sum for tax group E.
nettoTaxF	Netto sum for tax group F.
nettoTaxG	Netto sum for tax group G.
nettoTaxH	Netto sum for tax group H.

**1.1.2.108 info\_Get\_Storno\_VATSums()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_Storno_VATSums (
    string closure1,
    string option,
    string closure2 )
```

Information on the storno VAT assessed for the specified record or period

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'3*' - Information on Storno VAT assessed for the specified record or period
<i>closure2</i>	Fiscal memory record number



**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
closureCnt	Closure count for given period.
refundStornoCnt	Count of STORNO receipts - type REFUND.
operatorErrStornoCnt	Count of STORNO receipts - type OPERATOR'S ERROR.
taxBaseStornoCnt	Count of STORNO receipts - type TAX BASE REDUCTION.
refundSalesCnt	Sales count of STORNO receipts type REFUND.
operatorErrSalesCnt	Sales count of STORNO receipts type OPERATOR'S ERROR.
taxBaseSalesCnt	Sales count of STORNO receipts type TAX BASE REDUCTION.
vATTaxA	Vat (DDS) charged for tax group A.
vATTaxB	Vat (DDS) charged for tax group B.
vATTaxC	Vat (DDS) charged for tax group C.
vATTaxD	Vat (DDS) charged for tax group D.
vATTaxE	Vat (DDS) charged for tax group E.
vATTaxF	Vat (DDS) charged for tax group F.
vATTaxG	Vat (DDS) charged for tax group G.
vATTaxH	Vat (DDS) charged for tax group H.

**1.1.2.109 info\_Get\_StornoSums()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_StornoSums (
    string closure1,
    string option,
    string closure2 )
```

Information on the storno sales for the specified record or period

**Parameters**

<i>closure1</i>	Fiscal memory record number.
<i>option</i>	'1*' - Information on the sales for the specified record or period for STORNO.
<i>closure2</i>	Fiscal record number

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
closureCnt	Closure count for given period.
refundStornoCnt	Count of STORNO receipts - type REFUND.
operatorErrStornoCnt	Count of STORNO receipts - type OPERATOR'S ERROR.
taxBaseStornoCnt	Count of STORNO receipts - type TAX BASE REDUCTION.
refundSalesCnt	Sales count of STORNO receipts type REFUND.
operatorErrSalesCnt	Sales count of STORNO receipts type OPERATOR'S ERROR.
taxBaseSalesCnt	Sales count of STORNO receipts type TAX BASE REDUCTION.

totTaxA	Turnover for tax group A.
totTaxB	Turnover for tax group B.
totTaxC	Turnover for tax group C.
totTaxD	Turnover for tax group D.
totTaxE	Turnover for tax group E.
totTaxF	Turnover for tax group F.
totTaxG	Turnover for tax group G.
totTaxH	Turnover for tax group H.

### 1.1.2.110 info\_Get\_Sums()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_Sums (
    string closure1,
    string option,
    string closure2 )
```

Information on the sales for the specified record or period

#### Parameters

<i>closure1</i>	Fiscal memory record number.
<i>option</i>	'1' - Information on the sales for the specified record or period.
<i>closure2</i>	Fiscal record number

#### Returns

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
closureNumber	Closure number for given period.
receiptsCount	Decimal point count for Z report record .
totTaxA	Turnover for tax group A.
totTaxB	Turnover for tax group B.
totTaxC	Turnover for tax group C.
totTaxD	Turnover for tax group D.
totTaxE	Turnover for tax group E.
totTaxF	Turnover for tax group F.
totTaxG	Turnover for tax group G.
totTaxH	Turnover for tax group H.

### 1.1.2.111 info\_Get\_TaxRates()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_TaxRates ( )
```

Get tax rates

**Returns**

Dictionary with keys:

taxA	Tax rate A
taxB	Tax rate B
taxC	Tax rate C
taxD	Tax rate D
taxE	Tax rate E
taxF	Tax rate F
taxG	Tax rate G
taxH	Tax rate H

**1.1.2.112 info\_Get\_TaxRatesByPeriod()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_TaxRatesByPeriod (
    string startDate,
    string endDate )
```

Get tax rates by period

**Parameters**

<i>startDate</i>	Period start date: DDMMYY (6 bytes).
<i>endDate</i>	Period end date: DDMMYY (6 bytes).

**Returns**

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
aA	Information about tax group A.
bB	Information about tax group B.
cC	Information about tax group C.
dD	Information about tax group D.
eE	Information about tax group E.
fF	Information about tax group F.
gG	Information about tax group G.
hH	Information about tax group H.
dDMMYY	Date of tax rates set

**1.1.2.113 info\_Get\_TaxRateValues()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_TaxRateValues (
    string closure1,
    string option )
```

Information on the specified fiscal memory record used to set tax rates

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'5' - Information on the specified fiscal memory record used to set tax rates

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
decimalsCount	Decimal point count for Z report record .
enabled	Enabled tax rates (1 - enabled).
taxRateA	tax rate A in percents.
taxRateB	tax rate B in percents.
taxRateC	tax rate C in percents.
taxRateD	tax rate D in percents.
taxRateE	tax rate E in percents.
taxRateF	tax rate F in percents.
taxRateG	tax rate G in percents.
taxRateH	tax rate H in percents.
dateTime	Date and time (format: DD-MM-YY hh:mm:ss).

**1.1.2.114 info\_Get\_Turnover()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_Turnover (
    string closure1,
    string option )
```

Information on the total sales and VAT before the specified Z-report

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'10' - Information on the total sales and VAT before the specified Z-report

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
turnoverSum	Accumulated sales up to and including the specified daily report.
turnoverVATSum	Accumulated VAT up to and including the specified daily report.

**1.1.2.115 info\_Get\_VATSum()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_VATSum (
    string closure1,
    string option,
    string closure2 )
```

Information on the VAT assessed for the specified record or period

**Parameters**

<i>closure1</i>	Fiscal memory record number
<i>option</i>	'3' - Information on the VAT assessed for the specified record or period
<i>closure2</i>	Fiscal memory record number

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - Command is declined.
closureNumber	Closure number for given period.
receiptsCount	Decimal point count for Z report record .
vATTaxA	Vat (DDS) charged for tax group A.
vATTaxB	Vat (DDS) charged for tax group B.
vATTaxC	Vat (DDS) charged for tax group C.
vATTaxD	Vat (DDS) charged for tax group D.
vATTaxE	Vat (DDS) charged for tax group E.
vATTaxF	Vat (DDS) charged for tax group F.
vATTaxG	Vat (DDS) charged for tax group G.
vATTaxH	Vat (DDS) charged for tax group H.

**1.1.2.116 info\_Get\_VoltageAndTemperature()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.info_Get_VoltageAndTemperature (
    string voltage,
    string temperature )
```

Get the supply voltage and temperature

**Parameters**

<i>voltage</i>	The supply voltage in volts
<i>temperature</i>	The printing head temperature in degrees

**1.1.2.117 info\_LastErrorExtendedInfo()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_LastErrorExtendedInfo ( )
```

The command allows to obtain information on the last command completed with an error. This information is saved for successfully completed commands and also after the printer is powered off. It is cleared only on RAM reset and on execution of the command with a "CLEAR" text input.

**Returns**

Dictionary with keys:

cmd	Last error command number.
errorCode	Number: error code.
dateTime	Error date and time in the DD-MM-YY hh:mm:ss format.
errorText	Text description of the last error.

**1.1.2.118 info\_Pinpad\_CheckLastRecordedRRN()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_CheckLastRecordedRRN ( )
```

Check recently recorded RRN code in fiscal printer with RRN from last transaction in payment terminal

**Returns**

Returns dictionary with key:

fAnswer	P - passed or F - failed
---------	--------------------------

**1.1.2.119 info\_Pinpad\_ClearTransactionStatusFlags()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_ClearTransactionStatus↵  
Flags ( )
```

Clearing transaction status flags. Clears the flags specified in the command 172([info\\_Pinpad\\_StatusOfLastTransaction\(\)](#)). The command is forbidden in an open fiscal receipt.

**Returns**

Dictionary with key:

fAnswer	'P' - command passed.
---------	-----------------------

**1.1.2.120 info\_Pinpad\_DataForLastTransaction()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_DataForLastTransaction ( )
```

Data for last transaction

**Returns**

Dictionary with keys:

currentRRN	Last RRN code, saved in FD.
transactionNum	Transaction number.
transactionRRN	RRN code of transaction.
AC	Transaction authorization code.
price	Amount of transaction.
cardNo	The last four digits of the card number.

**1.1.2.121 info\_Pinpad\_Get\_DateTime()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_Get_DateTime ( )
```

Get date and time of payment terminal. The command is forbidden in an open fiscal receipt

**Returns**

Dictionary with key:

dateTime	Date and time in format: YYYY-MM-DD HH:MM:SS
----------	--

**1.1.2.122 info\_Pinpad\_GetUpdate()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_GetUpdate ( )
```

Checking and updating of payment terminal software. The command is forbidden in an open fiscal receipt

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---



**1.1.2.123 info\_Pinpad\_Information()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_Information ( )
```

Payment terminal information. The command is forbidden in an open fiscal receipt

**Returns**

Dictionary with keys:

model	Payment terminal model or 'F' - if commandd failed.
serialNum	Payment terminal serial number
softVer	Software version of payment terminal
terminalID	ID of payment terminal
menuType	Menu type of payment terminal

**1.1.2.124 info\_Pinpad\_NumberOfEntries()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_NumberOfEntries ( )
```

Number of entries in the payment terminal. The command is forbidden in an open fiscal receipt

**Returns**

Dictionary with key:

fAnswer	'P' - command passed + Number of entries, 'F' - command failed.
---------	---

**1.1.2.125 info\_Pinpad\_PrintLastSuccessfulTransaction()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_PrintLastSuccessful↵  
Transaction ( )
```

Prints the last successful transaction

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.126 info\_Pinpad\_StatusOfLastTransaction()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_StatusOfLastTransaction  
( )
```

Status of last transaction

**Returns**

Dictionary with keys:

timeout	Flag, if its value is 1, then the previous transaction ended with timeout
voidReceiptPrint	Flag, if its value is 1, then the previous transaction was a reversal and the purchase reversal receipt was not printed.
inTransaction	Flag, if its value is 1, then the printer power was turned off before the end of the printer transaction.

**1.1.2.127 info\_Pinpad\_TestConnection()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_TestConnection ( )
```

Test connection to a payment terminal. The command is forbidden in an open fiscal receipt

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.128 info\_Pinpad\_TestConnectionToServer()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_TestConnectionToServer ( )
```

Test connection to server. The command is forbidden in an open fiscal receipt

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.129 info\_Pinpad\_TransactionStatus()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Pinpad_TransactionStatus ( )
```

Transaction status (unsuccessful). The command is forbidden in an open fiscal receipt

**Returns**

Dictionaty with keys:

reversal	00 – no (unsuccessful) frozen transactions R – (unsuccessful) freeze purchase C – (unsuccessful) freeze reversal or 'F' - if commandd failed.
endDay	'00' – no need to close the day or '01' – day closing required

**1.1.2.130 info\_Print\_Diagnostic\_0()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.info_Print_Diagnostic_0 ( )
```

Print diagnostic information

**1.1.2.131 info\_Print\_Short\_FMReportByDTRange()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.info_Print_Short_FMReportByDTRange (
    string fromDate,
    string toDate )
```

Summary fiscal memory report by period

**Parameters**

<i>fromDate</i>	Start date: 6 bytes (DDMMYY)
<i>toDate</i>	End date: 6 bytes (DDMMYY)

**1.1.2.132 info\_Print\_Short\_MonthlyFMReport()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.info_Print_Short_MonthlyFMReport (
    string startValue )
```

Monthly fiscal memory report

**Parameters**

<i>startValue</i>	Month: 4 bytes (MMYY) for a monthly report
-------------------	--

**1.1.2.133 info\_Print\_Short\_YearlyFMReport()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.info_Print_Short_YearlyFMReport (
    string startValue )
```

Annual fiscal memory report

**Parameters**

<i>startValue</i>	Year: 2 bytes (YY) for an annual report
-------------------	---

**1.1.2.134 info\_Service\_Contracts()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.info_Service_Contracts ( )
```

Service contracts information and report

**Returns**

Dictionary with key:

fAnswer	'F' - failed or 'Left' - integer parameter: number of remaining service agreement registration fields. 'RegDtTm parameter' - Service agreement date and time: "DD-MM-YYYY hh:mm:ss". EndDate parameter - Last service agreement expiration date in the "DD-MM-YYYY" format. 'UIC parameter' 9 or 13 digits: service provider's UIC.
---------	---

**1.1.2.135 items\_Delete\_All\_Items()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Delete_All_Items (
    string dOption,
    string aOption )
```

Delete all items

**Parameters**

<i>dOption</i>	'D' - delete items
<i>aOption</i>	'A' - all items

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.136 items\_Delete\_Item()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Delete_Item (
    string option,
    string targetPLU )
```

Deletes the specified PLU code, if it has no accumulated sums

**Parameters**

<i>option</i>	'D' - delete item
<i>targetPLU</i>	PLU code

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.137 items\_Delete\_ItemsInRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Delete_ItemsInRange (
    string option,
    string startPLU,
    string endPLU )
```

Delete items in range

**Parameters**

<i>option</i>	'D' - delete items
<i>startPLU</i>	Start PLU code without accumulated sum.
<i>endPLU</i>	End PLU code without accumulated sum.

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

### 1.1.2.138 items\_Get\_FirstFoundItem()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_FirstFoundItem (
    string option,
    string targetPLU )
```

Get first found item

#### Parameters

<i>option</i>	'F' - for first found item
<i>targetPLU</i>	PLU code. Between 1 and 999999999

#### Returns

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
PLU	PLU code. Between 1 and 999999999.
taxGroup	Tax group. One byte.
group	Product group. Between 1 and 99.
singlePrice	loading-point number with the current decimal digits for the printer.
total	Accumulated sum for this PLU.
sold	Sold quantity. Floating-point number with 3 decimal digits.
available	Available quantity. Floating-point number with 3 decimal digits.
itemName	PLU name. Up to 36 bytes.

### 1.1.2.139 items\_Get\_FirstNotProgrammedItem()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_FirstNotProgrammedItem (
    string option,
    string targetPLU )
```

Find the first not programmed item

#### Parameters

<i>option</i>	'X' - for first not programmed item
<i>targetPLU</i>	PLU code (between 1 and 999999999).

#### Returns

Dictionary with key:

PLU	PLU code (1 - 999999999)
-----	--------------------------

**1.1.2.140 items\_Get\_FirstSoldItem()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_FirstSoldItem (
    string option,
    string targetPLU )
```

Get first sold item

**Parameters**

<i>option</i>	'f' - return the data on the first found PLU
<i>targetPLU</i>	PLU code. Between 1 and 999999999.

**Returns**

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
PLU	PLU code. Between 1 and 999999999.
taxGroup	Tax group. One byte.
group	Product group. Between 1 and 99.
singlePrice	floating-point number with the current decimal digits for the printer.
total	Accumulated sum for this PLU.
sold	Sold quantity. Floating-point number with 3 decimal digits.
available	Available quantity. Floating-point number with 3 decimal digits.
itemName	PLU name. Up to 36 bytes.

**1.1.2.141 items\_Get\_Item()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_Item (
    string option,
    string targetPLU )
```

Read item

**Parameters**

<i>option</i>	'R' - read PLU item
<i>targetPLU</i>	PLU code. Between 1 and 999999999.

**Returns**

Dictionary with keys:

---

errorCode	'P' - pass; 'F' - failed.
PLU	PLU code. Between 1 and 999999999.
taxGroup	Tax group. One byte.
group	Product group. Between 1 and 99.
singlePrice	floating-point number with the current decimal digits for the printer.
total	Accumulated sum for this PLU.
sold	Sold quantity. Floating-point number with 3 decimal digits.
available	Available quantity. Floating-point number with 3 decimal digits.
itemName	PLU name. Up to 36 bytes.

#### 1.1.2.142 items\_Get\_ItemsInformation()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_ItemsInformation (
    string option )
```

Get item information

##### Parameters

<i>option</i>	'I' - get item information
---------------	----------------------------

##### Returns

Dictionary with keys:

errorCode	'P' - passed, 'F' - failed.
total	Total number of programmable PLUs (3000 for this printer).
prog	Number of programmed PLUs.
len	Maximum PLU name length.

#### 1.1.2.143 items\_Get\_LastFoundItem()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_LastFoundItem (
    string option,
    string targetPLU )
```

Get last found item

##### Parameters

<i>option</i>	'L' - for last found item
<i>targetPLU</i>	PLU code. Between 1 and 999999999



**Returns**

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
PLU	PLU code. Between 1 and 999999999.
taxGroup	Tax group. One byte.
group	Product group. Between 1 and 99.
singlePrice	floating-point number with the current decimal digits for the printer.
total	Accumulated sum for this PLU.
sold	Sold quantity. Floating-point number with 3 decimal digits.
available	Available quantity. Floating-point number with 3 decimal digits.
itemName	PLU name. Up to 36 bytes.

**1.1.2.144 items\_Get\_LastNotProgrammedItem()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_LastNotProgrammedItem (
    string option,
    string targetPLU )
```

Find the last not programmed PLU

**Parameters**

<i>option</i>	'x' - for last not programmed item
<i>targetPLU</i>	PLU code (between 1 and 999999999).

**Returns**

Dictionary with key:

PLU	PLU code (1 - 999999999)
-----	--------------------------

**1.1.2.145 items\_Get\_LastSoldItem()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_LastSoldItem (
    string option,
    string targetPLU )
```

Return the data on the last found PLU item shaving sales

**Parameters**

<i>option</i>	'I' - for last found sold item
<i>targetPLU</i>	PLU code. Between 1 and 999999999.

**Returns**

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
PLU	PLU code. Between 1 and 999999999.
taxGroup	Tax group. One byte.
group	Product group. Between 1 and 99.
singlePrice	floating-point number with the current decimal digits for the printer.
total	Accumulated sum for this PLU.
sold	Sold quantity. Floating-point number with 3 decimal digits.
available	Available quantity. Floating-point number with 3 decimal digits.
itemName	PLU name. Up to 36 bytes.

**1.1.2.146 items\_Get\_NextItem()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_NextItem (
    string option )
```

Get next item

**Parameters**

<i>option</i>	'N' - for next item
---------------	---------------------

**Returns**

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
PLU	PLU code. Between 1 and 999999999.
taxGroup	Tax group. One byte.
group	Product group. Between 1 and 99.
singlePrice	floating-point number with the current decimal digits for the printer.
total	Accumulated sum for this PLU.
sold	Sold quantity. Floating-point number with 3 decimal digits.
available	Available quantity. Floating-point number with 3 decimal digits.
itemName	PLU name. Up to 36 bytes.

**1.1.2.147 items\_Get\_NextSoldItem()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Get_NextSoldItem (
    string option,
    string targetPLU )
```

Return the data on the next found sold PLU

**Parameters**

<i>option</i>	'n' - for next found sold item
<i>targetPLU</i>	PLU code (between 1 and 999999999).

**Returns**

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
PLU	PLU code. Between 1 and 999999999.
taxGroup	Tax group. One byte.
group	Product group. Between 1 and 99.
singlePrice	floating-point number with the current decimal digits for the printer.
total	Accumulated sum for this PLU.
sold	Sold quantity. Floating-point number with 3 decimal digits.
available	Available quantity. Floating-point number with 3 decimal digits.
itemName	PLU name. Up to 36 bytes.

**1.1.2.148 items\_Set\_Item()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Set_Item (
    string option,
    string taxGroup,
    string targetPLU,
    string group,
    string singlePrice,
    string quantity,
    string itemName )
```

Program an item

**Parameters**

<i>option</i>	'P' - program an item
<i>taxGroup</i>	Tax group ('A','B','C','D','E','F','G','H')
<i>targetPLU</i>	PLU code (between 1 and 999999999)
<i>group</i>	Product group (between 1 and 99)
<i>singlePrice</i>	Unit price. Up to 8 significant digits
<i>quantity</i>	Number with up to 3 decimal digits: the available quantity for this PLU.
<i>itemName</i>	PLU name. Up to 36 bytes.

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.149 items\_Set\_ItemQuantity()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Set_ItemQuantity (
    string option,
    string targetPLU,
    string quantity )
```

Edit the available PLU quantity.

**Parameters**

<i>option</i>	'A' - edit qantity
<i>targetPLU</i>	PLU code (between 1 and 999999999).
<i>quantity</i>	Quantity adjustment: a floating-point number with up to 3 decimal digits. Positive numbers increase and negative numbers decrease the quantity.

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.150 items\_Set\_ItemWithReplace()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.items_Set_ItemWithReplace (
    string option,
    string taxGroup,
    string targetPLU,
    string group,
    string singlePrice,
    string replace,
    string quantity,
    string itemName )
```

Program an item

**Parameters**

<i>option</i>	'P' - program an item
<i>taxGroup</i>	Tax group ('A','B','C','D','E','F','G','H')
<i>targetPLU</i>	PLU code (between 1 and 999999999)
<i>group</i>	Product group (between 1 and 99)
<i>singlePrice</i>	Unit price. Up to 8 significant digits
<i>replace</i>	One byte with a value of 'A', available quantity is replaced by 'quantity'
<i>quantity</i>	Number with up to 3 decimal digits: the available quantity for this PLU.
<i>itemName</i>	PLU name. Up to 36 bytes.

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.151 klen\_Find\_Document()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Find_Document (
    string subcommand01,
    string documentType,
    string documentNumber )
```

Read documents

**Parameters**

<i>subcommand01</i>	'R' - Read EJ data.
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>documentNumber</i>	Document number

**Returns**

Dictionary with keys:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
fRow	Text in a row

**1.1.2.152 klen\_Get\_Info()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Get_Info (
    string subcommand01,
    string subcommand02 )
```

Electronic journal information

**Parameters**

<i>subcommand01</i>	'I' - EJ information
<i>subcommand02</i>	'X'

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - No EJ data are found.
sizeTotal	Total journal size in bytes
sizeUsed	Used journal size in bytes.
firstZReportNumber	First Z-report number in the EJ.
lastZReportNumber	Last Z-report number in the EJ.
firstDocumentNumber	First document number in the EJ.
lastDocumentNumber	Last document number in the EJ.

#### 1.1.2.153 klen\_Get\_KLENInfo\_ZReport()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Get_KLENInfo_ZReport (
    string subcommand01,
    string subcommand02,
    string zReportNumber )
```

Get EJ information for Z report number

##### Parameters

subcommand01	'C' - Verify EJ data
subcommand02	'R' - Returns EJ information for the Z-report number
zReportNumber	Z report number

##### Returns

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - No EJ data are found for this Z-report.
fDocNumber	Document number of the Z-report.
dateTime	Date and time of the report in the "DD-MM-YYYY hh:mm:ss" format.
zReportSHA1	40 characters: hexadecimal SHA-1 of the Z-report.

#### 1.1.2.154 klen\_Get\_NextTextRow()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Get_NextTextRow (
    string subcommand01 )
```

Get next text row

##### Parameters

subcommand01	'N' - for next row
--------------	--------------------

**Returns**

Dictionary with keys:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
fRow	Text in a row

**1.1.2.155 klen\_Get\_ZReportInfo()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Get_ZReportInfo (
    string subcommand01,
    string subcommand02,
    string zReportNumber )
```

Reads the actual text of the documents stored in the EJ, calculates the SHA-1 and compares it to the SHA-1 from the Z-report. The command can take a long time to complete, if there are many documents in the daily report.

**Parameters**

subcommand01	'C' - Verify EJ data
subcommand02	'Z' - Reads the actual text of the documents stored in the EJ
zReportNumber	Z report number

**Returns**

Dictionary with keys:

errorCode	'P' - Command is executed correctly, 'F' - No EJ data are found for this Z-report.
documentsCount	Documents.
bytesCount	Data bytes.
sha1KLEN	40 characters: the hexadecimal checksum.
sha1Calculated	40 characters each: the EJ checksum and the calculated checksum, respectively.

**1.1.2.156 klen\_Print\_DocsFromDateToDate()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Print_DocsFromDateToDate (
    string subcommand01,
    string fontSize,
    string documentType,
    string fromDateTime,
    string toDateTime )
```

Print documents by date and time

**Parameters**

subcommand01	'P' - print data
--------------	------------------

**Parameters**

<i>fontSize</i>	One byte with the following allowed values: '>' Print with a normal font size. '<' Print with a font size.
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>fromDateTime</i>	Report start date and time in the DDMMYY[hmmss] format. If the time is omitted, "000000" is assumed, i.e., 00:00:00.
<i>toDateTime</i>	Report end date and time in the DDMMYY[hmmss] format. If the time is omitted, "235959" is assumed, i.e., 23:59:59

**Returns**

Dictionary with key:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
---------	--

**1.1.2.157 klen\_Print\_DocsInRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Print_DocsInRange (
    string subcommand01,
    string fontSize,
    string documentType,
    string startNumber,
    string endNumber )
```

Print EJ data

**Parameters**

<i>subcommand01</i>	'P' - print data
<i>fontSize</i>	One byte with the following allowed values: '>' Print with a normal font size. '<' Print with a font size.
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>startNumber</i>	Start document number
<i>endNumber</i>	End document number

**Returns**

Dictionary with key:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
---------	--



**1.1.2.158 klen\_Print\_DocsInRange\_InZReport()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Print_DocsInRange_InZReport (
    string subcommand01,
    string fontSize,
    string documentType,
    string zReportNumber,
    string startNumber,
    string endNumber )
```

Print EJ data

**Parameters**

<i>subcommand01</i>	'P' - print data
<i>fontSize</i>	One byte with the following allowed values: '>' Print with a normal font size. '<' Print with a font size.
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>zReportNumber</i>	Z report number
<i>startNumber</i>	Start document number
<i>endNumber</i>	End document number

**Returns**

Dictionary with key:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
---------	--

**1.1.2.159 klen\_Print\_Document()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Print_Document (
    string subcommand01,
    string fontSize,
    string documentType,
    string documentNumber )
```

Print EJ data

**Parameters**

<i>subcommand01</i>	'P' - print data
<i>fontSize</i>	One byte with the following allowed values: '>' Print with a normal font size. '<' Print with a font size.
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>documentNumber</i>	Document number

**Returns**

Dictionary with key:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
---------	--

**1.1.2.160 klen\_Print\_SHA1()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.klen_Print_SHA1 (
    string subcommand01,
    string subcommand02 )
```

Print with SHA1 - prints information on each found SHA-1, valid or not

**Parameters**

<i>subcommand01</i>	'C' - Check the validity of the EJ or a part of it
<i>subcommand02</i>	'P' - Prints report on the validity of all SHA-1 checksums for the Z-reports found in the EJ. It compares the EJ and fiscal memory SHA-1 checksums. For each mismatch, prints one line with the Z-report number, date and time

**1.1.2.161 klen\_Print\_Validation()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.klen_Print_Validation (
    string subcommand01,
    string subcommand02 )
```

Print validation

**Parameters**

<i>subcommand01</i>	'C' - Check the validity of the EJ or a part of it
<i>subcommand02</i>	'P' - Prints report on the validity of all SHA-1 checksums for the Z-reports found in the EJ. It compares the EJ and fiscal memory SHA-1 checksums. For each mismatch, prints one line with the Z-report number, date and time

**1.1.2.162 klen\_Set\_DocsRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Set_DocsRange (
    string subcommand01,
    string documentType,
    string startNumber,
    string endNumber )
```

Read EJ in range

**Parameters**

<i>subcommand01</i>	'R' - read EJ data
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>startNumber</i>	Start document number
<i>endNumber</i>	End document number

**Returns**

Dictionary with keys:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
fRow	Text in a row

**1.1.2.163 klen\_Set\_DocsRange\_ByDateTime()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Set_DocsRange_ByDateTime (
    string subcommand01,
    string documentType,
    string fromDateTime,
    string toDateTime )
```

Read EJ data in range by date and time

**Parameters**

<i>subcommand01</i>	'R' - read EJ data
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>fromDateTime</i>	Report start date and time in the DDMMYY[h:mm:ss] format. If the time is omitted, "000000" is assumed, i.e., 00:00:00.
<i>toDateTime</i>	Report end date and time in the DDMMYY[h:mm:ss] format. If the time is omitted, "235959" is assumed, i.e., 23:59:59.

**Returns**

Dictionary with keys:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
fRow	Text in a row

**1.1.2.164 klen\_Set\_DocsRange\_InZReport()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Set_DocsRange_InZReport (
```

```

string subcommand01,
string documentType,
string zReportNumber,
string startNumber,
string endNumber )

```

Read EJ data in range with document numbers and Z report

#### Parameters

<i>subcommand01</i>	'R' - read EJ data
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>zReportNumber</i>	Z report number
<i>startNumber</i>	Start document number
<i>endNumber</i>	End document number

#### Returns

Dictionary with keys:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
fRow	Text in a row

#### 1.1.2.165 klen\_Set\_DocsRange\_InZReport\_01()

```

Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Set_DocsRange_InZReport_01 (
    string subcommand01,
    string documentType,
    string zReportNumber )

```

Read EJ data (Z reports numbers)

#### Parameters

<i>subcommand01</i>	'R' - read EJ data
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>zReportNumber</i>	Z report number

#### Returns

Dictionary with keys:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
fRow	Text in a row

**1.1.2.166 klen\_Set\_DocsRange\_InZReport\_02()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_Set_DocsRange_InZReport_02 (
    string subcommand01,
    string documentType,
    string zReportNumber,
    string documentNumber )
```

Read EJ data in range (Z reports)

**Parameters**

<i>subcommand01</i>	'R' - read EJ data
<i>documentType</i>	'A' All document types; 'F' Fiscal(customer) receipts; 'V' Refund(storno) receipts; 'C' Cancelled(customer) receipts. 'N' Service receipts; 'I' Service deposit receipts; 'O' Service withdrawal receipts.
<i>zReportNumber</i>	Z report number
<i>documentNumber</i>	Document number

**Returns**

Dictionary with keys:

fAnswer	'P' - Command is executed correctly, 'F' - No EJ data are found.
fRow	Text in a row

**1.1.2.167 klen\_SInfo\_ExV\_ByDateRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_ExV_ByDateRange (
    string option,
    string exOption,
    string optionD,
    string fromDateTime,
    string toDateTime )
```

Get structured information by date and time range plus date and time

**Parameters**

<i>option</i>	'V' - for structured information
<i>exOption</i>	'+' - the date and time of the document are added in the format: DD-MM-YYYY HH: MM: SS + pasted tab symbol
<i>optionD</i>	'D' - for dates
<i>fromDateTime</i>	Format is: DDMMYY [hhmmss].If you skip the hour, the start time is 00:00:00
<i>toDateTime</i>	Format is: DDMMYY [hhmmss] format.If you skip the hour, the end time is 23:59:59

**Returns**

Dictionary with keys: The data returned in the order described below is separated by a tab symbol enclosed in quotation marks:

fiscal device ID; kind of receipt - (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); fiscal receipt), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Invoice), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Refund receipt) or (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); global number of receipt; Unique Sale ID; commodity/service - name; commodity/service – single price; commodity/service - quantity; commodity/service - price; total price for the receipt; Fiscal printer programming interface; Invoice number/Credit notification - if the entry is for Invoice or Credit notification; UIC of recipient – if the entry is for Invoice or Credit notification; global number of the refund receipt – if the entry is for Invoice or Credit notification; number of the refunded invoice – if the entry is for Invoice or Credit notification; reason for issue – in case if entry is for refund receipt or Credit notification

**1.1.2.168 klen\_SInfo\_ExV\_ByNumber()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_ExV_ByNumber (
    string option,
    string exOption,
    string startNumber )
```

Get structured information by date and time plus date and time

**Parameters**

<i>option</i>	'V' - for structured information
<i>exOption</i>	'+' - the date and time of the document are added in the format: DD-MM-YYYY HH: MM: SS + pasted tab symbol
<i>startNumber</i>	Number of the first document for the report

**Returns**

Dictionary with keys: The data returned in the order described below is separated by a tab symbol enclosed in quotation marks:

fiscal device ID; kind of receipt - (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); fiscal receipt), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Invoice), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Refund receipt) or (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); global number of receipt; Unique Sale ID; commodity/service - name; commodity/service – single price; commodity/service - quantity; commodity/service - price; total price for the receipt; Fiscal printer programming interface; Invoice number/Credit notification - if the entry is for Invoice or Credit notification; UIC of recipient – if the entry is for Invoice or Credit notification; global number of the refund receipt – if the entry is for Invoice or Credit notification; number of the refunded invoice – if the entry is for Invoice or Credit notification; reason for issue – in case if entry is for refund receipt or Credit notification

**1.1.2.169 klen\_SInfo\_ExV\_ByNumbersRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_ExV_ByNumbersRange (
    string option,
```

```

string exOption,
string startNumber,
string endNumber )

```

Return structured information by numbers range plus date and time

#### Parameters

<i>option</i>	'V' - return structured information
<i>exOption</i>	'+' - the date and time of the document are added in the format: DD-MM-YYYY HH: MM: SS + pasted tab symbol
<i>startNumber</i>	Number of the first document for the report
<i>endNumber</i>	Number of the last document for the report

#### Returns

Dictionary with keys: The data returned in the order described below is separated by a tab symbol enclosed in quotation marks:

fiscal device ID; kind of receipt - (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); fiscal receipt), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Invoice), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Refund receipt) or (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); global number of receipt; Unique Sale ID; commodity/service - name; commodity/service – single price; commodity/service - quantity; commodity/service - price; total price for the receipt; Fiscal printer programming interface; Invoice number/Credit notification - if the entry is for Invoice or Credit notification; UIC of recipient – if the entry is for Invoice or Credit notification; global number of the refund receipt – if the entry is for Invoice or Credit notification; number of the refunded invoice – if the entry is for Invoice or Credit notification; reason for issue – in case if entry is for refund receipt or Credit notification

#### 1.1.2.170 klen\_SInfo\_ExV\_GetNext()

```

Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_ExV_GetNext (
    string option )

```

Get next line structured information

#### Parameters

<i>option</i>	'v' - for next line structured info
---------------	-------------------------------------

#### Returns

Dictionary with keys: The data returned in the order described below is separated by a tab symbol enclosed in quotation marks:

fiscal device ID; kind of receipt - (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); fiscal receipt), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Invoice), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Refund receipt) or (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); global number of receipt; Unique Sale ID; commodity/service - name; commodity/service – single price; commodity/service - quantity; commodity/service - price; total price for the receipt; Fiscal printer programming interface; Invoice number/Credit notification - if the entry is for Invoice or Credit

notification; UIC of recipient – if the entry is for Invoice or Credit notification;  
 global number of the refund receipt – if the entry is for Invoice or Credit notification; number of the refunded invoice – if the entry is for Invoice or Credit notification; reason for issue – in case if entry is for refund receipt or Credit notification

#### 1.1.2.171 klen\_SInfo\_V\_ByDateRange()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_V_ByDateRange (
    string option,
    string optionD,
    string fromDateTime,
    string toDateTime )
```

Get structured information by date and time

##### Parameters

<i>option</i>	'V' - for structured information
<i>optionD</i>	'D' - for dates
<i>fromDateTime</i>	DDMMYY [hhmmss] format.If you skip the hour, the start time is 00:00:00
<i>toDateTime</i>	DDMMYY [hhmmss] format.If you skip the hour, the end time is 23:59:59

##### Returns

Dictionary with keys: The data returned in the order described below is separated by a tab symbol enclosed in quotation marks:

fiscal device ID; kind of receipt - (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); fiscal receipt), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Invoice), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Refund receipt) or (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); global number of receipt; Unique Sale ID; commodity/service - name; commodity/service – single price; commodity/service - quantity; commodity/service - price; total price for the receipt; Fiscal printer programming interface; Invoice number/Credit notification - if the entry is for Invoice or Credit notification; UIC of recipient – if the entry is for Invoice or Credit notification;  
 global number of the refund receipt – if the entry is for Invoice or Credit notification; number of the refunded invoice – if the entry is for Invoice or Credit notification; reason for issue – in case if entry is for refund receipt or Credit notification

#### 1.1.2.172 klen\_SInfo\_V\_ByNumber()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_V_ByNumber (
    string option,
    string startNumber )
```

Return structured information by number



**Parameters**

<i>option</i>	'V' - return structured information
<i>startNumber</i>	Number of the first document for the report

**Returns**

Dictionary with keys:

<i>errorCode</i>	'P' - pass; 'F' - failed.
<i>datafield01</i>	The data returned in the order described below is separated by a tab symbol enclosed in quotation marks: fiscal device ID; kind of receipt - (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); fiscal receipt), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Invoice), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Refund receipt) or (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); global number of receipt; Unique Sale ID; commodity/service - name; commodity/service – single price; commodity/service - quantity; commodity/service - price; total price for the receipt; Fiscal printer programming interface; Invoice number/Credit notification - if the entry is for Invoice or Credit notification; UIC of recipient – if the entry is for Invoice or Credit notification; global number of the refund receipt – if the entry is for Invoice or Credit notification; number of the refunded invoice – if the entry is for Invoice or Credit notification; reason for issue – in case if entry is for refund receipt or Credit notification

**1.1.2.173 klen\_SInfo\_V\_ByNumbersRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_V_ByNumbersRange (
    string option,
    string startNumber,
    string endNumber )
```

Return structured information by number

**Parameters**

<i>option</i>	'V' - return structured information
<i>startNumber</i>	Number of the first document for the report
<i>endNumber</i>	Number of the last document for the report

**Returns**

Dictionary with keys: The data returned in the order described below is separated by a tab symbol enclosed in quotation marks:

fiscal device ID; kind of receipt - (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); fiscal receipt), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Invoice), (fiscal receipt),  
(Invoice), (Refund receipt) or (Credit notification); Refund receipt) or (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); global number of receipt; Unique Sale ID; commodity/service - name; commodity/service – single price; commodity/service - quantity; commodity/service - price; total price for the receipt;  
Fiscal printer programming interface; Invoice number/Credit notification - if the entry is for Invoice or Credit

notification; UIC of recipient – if the entry is for Invoice or Credit notification;  
 global number of the refund receipt – if the entry is for Invoice or Credit notification; number of the refunded invoice – if the entry is for Invoice or Credit notification; reason for issue – in case if entry is for refund receipt or Credit notification

#### 1.1.2.174 klen\_SInfo\_V\_GetNext()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_V_GetNext (
    string option )
```

Get next line structured information

##### Parameters

<i>option</i>	'v' - for next line structured info
---------------	-------------------------------------

##### Returns

Dictionary with keys: The data returned in the order described below is separated by a tab symbol enclosed in quotation marks:

fiscal device ID; kind of receipt - (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); fiscal receipt), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Invoice), (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); Refund receipt) or (fiscal receipt), (Invoice), (Refund receipt) or (Credit notification); global number of receipt; Unique Sale ID; commodity/service - name; commodity/service – single price; commodity/service - quantity; commodity/service - price; total price for the receipt; Fiscal printer programming interface; Invoice number/Credit notification - if the entry is for Invoice or Credit notification; UIC of recipient – if the entry is for Invoice or Credit notification; global number of the refund receipt – if the entry is for Invoice or Credit notification; number of the refunded invoice – if the entry is for Invoice or Credit notification; reason for issue – in case if entry is for refund receipt or Credit notification

#### 1.1.2.175 klen\_SInfo\_W\_ByDateRange()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_W_ByDateRange (
    string option,
    string optionD,
    string fromDateTime,
    string toDateTime )
```

Read structured information by date and time

##### Parameters

<i>option</i>	'W' - return structured information
<i>optionD</i>	'D' - Receipt start and end date and time
<i>fromDateTime</i>	Start date and time in the DD-MM-YYYY hh:mm:ss format
<i>toDateTime</i>	End date and time in the DD-MM-YYYY hh:mm:ss format

**Returns**

Depends on the type. Check protocol for more!

**1.1.2.176 klen\_SInfo\_W\_ByNumber()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_W_ByNumber (
    string option,
    string startNumber )
```

Read structured information

**Parameters**

<i>option</i>	'W' - return structured information
<i>startNumber</i>	Number of the first document for the report.

**Returns**

Depends on the type. Check protocol for more!

**1.1.2.177 klen\_SInfo\_W\_ByNumbersRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_W_ByNumbersRange (
    string option,
    string startNumber,
    string endNumber )
```

Read structured information by range

**Parameters**

<i>option</i>	'W' - return structured information
<i>startNumber</i>	Number of the first document for the report.
<i>endNumber</i>	Number of the last document for the report.

**Returns**

Depends on the type. Check protocol for more!

**1.1.2.178 klen\_SInfo\_W\_GetNext()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_W_GetNext (
    string option )
```

Get next text line structured information

**Parameters**

<i>option</i>	'w' - get structured info
---------------	---------------------------

**Returns**

Depends on the type. Check protocol for more!

**1.1.2.179 klen\_SInfo\_Y\_ByDateRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_Y_ByDateRange (
    string option,
    string optionD,
    string fromDateTime,
    string toDateTime )
```

Get structured information

**Parameters**

<i>option</i>	'Y' - Get structured info
<i>optionD</i>	'D' - date and time
<i>fromDateTime</i>	Start date and time
<i>toDateTime</i>	End date and time

**Returns**

Depends on the type. Check protocol for more!

**1.1.2.180 klen\_SInfo\_Y\_ByNumber()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_Y_ByNumber (
    string option,
    string startNumber )
```

Return structured information

**Parameters**

<i>option</i>	'Y' - return structured information
<i>startNumber</i>	Start document number

**Returns**

Depends on the type. Check protocol for more!

**1.1.2.181 klen\_SInfo\_Y\_ByNumbersRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_Y_ByNumbersRange (
    string option,
    string startNumber,
    string endNumber )
```

Get structured information

**Parameters**

<i>option</i>	'Y' - Get structured info
<i>startNumber</i>	Start document number
<i>endNumber</i>	End document number

**Returns**

Depends on the type. Check protocol for more!

**1.1.2.182 klen\_SInfo\_Y\_GetNext()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.klen_SInfo_Y_GetNext (
    string option )
```

Get next line structured information

**Parameters**

<i>option</i>	'y' - Get next line structured information
---------------	--

**Returns**

Depends on the type. Check protocol for more!

**1.1.2.183 open\_StornoReceipt()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.open_StornoReceipt (
    string operatorNumber,
```

```

string operatorPassword,
string tillNumber,
string invoice,
string invoiceNumber,
string uNP,
string stornoType,
string stornoDocumentNumber,
string stornoUNP,
string stornoDateTime,
string stornoFMNumber,
string stornoReason,
ref bool isFullStorno )

```

Combined variant for Strono receipt

#### Parameters

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - if invoice is wanted
<i>invoiceNumber</i>	Invoice number
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes), for example: DT000600-OP01-0001000). When opening a sales receipt for the first time, the UNP(Unique Sale ID) must be set at least once, if the parameter is then omitted, the FU will increment the unit by the sale number automatically.
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document
<i>stornoUNP</i>	Unique Sale ID of the storno document. The whole Unique Sale ID must be set.
<i>stornoDateTime</i>	Date and time of the reversed document. Format 'DDMMYYhhmmss'.>
<i>stornoFMNumber</i>	Fiscal Memory ID of the FD which has issued the storno receipt
<i>stornoReason</i>	Reason (up to 30 characters.)
<i>isFullStorno</i>	If 'stornoUNP', 'stornoDateTime' and 'stornoFMNumber' are set, this parameter must be set to FALSE, otherwise - TRUE

#### 1.1.2.184 other\_Sound\_Signal()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.other_Sound_Signal (
    string soundData )

```

Set sound signal

#### Parameters

<i>soundData</i>	'C' Do; 'D' Re; 'E' Mi; 'F' Fa; 'G' Sol; 'A' La; 'B' Ti
------------------	---

**1.1.2.185 receipt\_CashIn\_CashOut()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_CashIn_CashOut (
    string amount )
```

Cash in and cash out

**Parameters**

<i>amount</i>	The amount to be registered (up to 10 significant digits). Depending on the sign of the number, it is interpreted as a deposit or withdrawal
---------------	--

**Returns**

Dictionary with keys:

errorCode	'P' The request is executed. If the requested amount is not zero, the printer prints a service receipt to register the transaction. 'F' The request is denied. This happens, if: The available cash is less than the requested service withdrawal; There is an open fiscal or service receipt.
cashSum	Available cash. The amount is increased both with this command and with each cash payment.
servIn	The sum of all "Cash in" commands
servOut	The sum of all "cash out" commands

**1.1.2.186 receipt\_Current\_Info()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Current_Info ( )
```

Information about current receipt

**Returns**

Dictionary with keys:

canVd	Is it possible refund.
taxA	Accumulated sum from tax group A.
taxB	Accumulated sum from tax group B.
taxC	Accumulated sum from tax group B.
taxD	Accumulated sum from tax group B.
taxE	Accumulated sum from tax group B.
taxF	Accumulated sum from tax group B.
taxG	Accumulated sum from tax group B.
taxH	Accumulated sum from tax group B.
inv	Is it open invoice? '0' - no, '1' - yes.
invNumber	Next invoice number (up to 10 digits).



**1.1.2.187 receipt\_DisplayDepPLU\_Sale()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDepPLU_Sale (
    string dChar,
    string targetPLU,
    string department,
    string quantity )
```

Registering the sale of a programmed item

**Parameters**

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well.
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

**1.1.2.188 receipt\_DisplayDepPLU\_Sale\_CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDepPLU_Sale_CByPercent (
    string dChar,
    string targetPLU,
    string department,
    string quantity,
    string percent )
```

Registering the sale of a programmed item

**Parameters**

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well.
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount(depending on the sign), as a percentage of the current sale.The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.189 receipt\_DisplayDepPLU\_Sale\_CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDepPLU_Sale_CBySum (
```

```

string dChar,
string targetPLU,
string department,
string quantity,
string abs )

```

Registering the sale of a programmed item

#### Parameters

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well.
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.190 receipt\_DisplayDepPLU\_SaleUn()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDepPLU_SaleUn (
    string dChar,
    string targetPLU,
    string department,
    string quantity,
    string measure )

```

Registering the sale of a programmed item

#### Parameters

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well.
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"

#### 1.1.2.191 receipt\_DisplayDepPLU\_SaleUn\_CByPercent()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDepPLU_SaleUn_CByPercent (
    string dChar,

```

```

string targetPLU,
string department,
string quantity,
string measure,
string percent )

```

Registering the sale of a programmed item

#### Parameters

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well.
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount(depending on the sign), as a percentage of the current sale.The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

#### 1.1.2.192 receipt\_DisplayDepPLU\_SaleUn\_CBySum()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDepPLU_SaleUn_CBySum (
    string dChar,
    string targetPLU,
    string department,
    string quantity,
    string measure,
    string abs )

```

Registering the sale of a programmed item

#### Parameters

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well.
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount.A discount amount exceeding the value of the sale is not allowed.

**1.1.2.193 receipt\_DisplayDSale()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDSale (
    string textRow,
    string department,
    string singlePrice,
    string quantity )
```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed.
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits

**1.1.2.194 receipt\_DisplayDSale\_CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDSale_CByPercent (
    string textRow,
    string department,
    string singlePrice,
    string quantity,
    string percent )
```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed.
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between -99.00% and 99.00%.

**1.1.2.195 receipt\_DisplayDSale\_CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDSale_CBySum (
    string textRow,
```

```

    string department,
    string singlePrice,
    string quantity,
    string abs )

```

Register and display of a sale

#### Parameters

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed.
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.196 receipt\_DisplayDSale\_Minimum()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDSale_Minimum (
    string textRow,
    string department,
    string singlePrice )

```

Register and display of a sale

#### Parameters

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed.
<i>singlePrice</i>	This is the price, with up to 8 significant digits

#### 1.1.2.197 receipt\_DisplayDSale\_Un()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDSale_Un (
    string textRow,
    string department,
    string singlePrice,
    string quantity,
    string measure )

```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed.
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

**1.1.2.198 receipt\_DisplayDSale\_UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDSale_UnCByPercent (
    string textRow,
    string department,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed.
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between -99.00% and 99.00%.

**1.1.2.199 receipt\_DisplayDSale\_UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayDSale_UnCBySum (
    string textRow,
    string department,
    string singlePrice,
    string quantity,
    string measure,
    string abs )
```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed.
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.200 receipt\_DisplayPLUSale()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayPLUSale (
    string dChar,
    string targetPLU,
    string quantity )
```

Registering the sale of a programmed item

**Parameters**

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits (no more than 3 after the decimal point).

**1.1.2.201 receipt\_DisplayPLUSale\_CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayPLUSale_CByPercent (
    string dChar,
    string targetPLU,
    string quantity,
    string percent )
```

Registering the sale of a programmed item

**Parameters**

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).

**Parameters**

<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.202 receipt\_DisplayPLUSale\_CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayPLUSale_CBySum (
    string dChar,
    string targetPLU,
    string quantity,
    string abs )
```

Registering the sale of a programmed item

**Parameters**

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.203 receipt\_DisplayPLUSale\_Un()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayPLUSale_Un (
    string dChar,
    string targetPLU,
    string quantity,
    string measure )
```

Registering the sale of a programmed item

**Parameters**

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"



**1.1.2.204 receipt\_DisplayPLUSale\_UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayPLUSale_UnCByPercent (
    string dChar,
    string targetPLU,
    string quantity,
    string measure,
    string percent )
```

Registering the sale of a programmed item

**Parameters**

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.205 receipt\_DisplayPLUSale\_UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplayPLUSale_UnCBySum (
    string dChar,
    string targetPLU,
    string quantity,
    string measure,
    string abs )
```

Registering the sale of a programmed item

**Parameters**

<i>dChar</i>	'D' - If present, the sale is shown on the customer display. If the length of the PLU name is more than 20, the characters after 20 are truncated.
<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.206 receipt\_DisplaySale()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplaySale (
    string textRow,
    string taxGroup,
    string singlePrice,
    string quantity )
```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...)
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits

**1.1.2.207 receipt\_DisplaySale\_CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplaySale_CByPercent (
    string textRow,
    string taxGroup,
    string singlePrice,
    string quantity,
    string percent )
```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...)
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>percent</i>	parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between -99.00% and 99.00%..

**1.1.2.208 receipt\_DisplaySale\_CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplaySale_CBySum (
    string textRow,
```

```

    string taxGroup,
    string singlePrice,
    string quantity,
    string abs )

```

Register and display of a sale

#### Parameters

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...)
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed

#### 1.1.2.209 receipt\_DisplaySale\_Minimum()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplaySale_Minimum (
    string textRow,
    string taxGroup,
    string singlePrice )

```

Register and display of a sale

#### Parameters

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...)
<i>singlePrice</i>	This is the price, with up to 8 significant digits

#### 1.1.2.210 receipt\_DisplaySale\_Un()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplaySale_Un (
    string textRow,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure )

```

Register and display of a sale

#### Parameters

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
----------------	---

**Parameters**

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...)
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

**1.1.2.211 receipt\_DisplaySale\_UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplaySale_UnCByPercent (
    string textRow,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...)
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between -99.00% and 99.00%..

**1.1.2.212 receipt\_DisplaySale\_UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DisplaySale_UnCBySum (
    string textRow,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string abs )
```

Register and display of a sale

**Parameters**

<i>textRow</i>	String, up to 20 bytes, containing a line of text describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...)
<i>singlePrice</i>	This is the price, with up to 8 significant digits
<i>quantity</i>	Optional parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed

**1.1.2.213 receipt\_Drawer\_KickOut()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Drawer_KickOut (
    string mSec )
```

Drawer opening

**Parameters**

<i>mSec</i>	The length of the impulse in milliseconds. ( 5...100 )
-------------	--

**1.1.2.214 receipt\_DSale()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale (
    string textRow1,
    string textRow2,
    string department,
    string singlePrice,
    string quantity )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

**1.1.2.215 receipt\_DSale\_CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_CByPercent (
    string textRow1,
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.216 receipt\_DSale\_CByPercentWText()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_CByPercentWText (
    string department,
    string singlePrice,
    string quantity,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.217 receipt\_DSale\_CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_CBySum (
```

```

string textRow1,
string textRow2,
string department,
string singlePrice,
string quantity,
string abs )

```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.218 receipt\_DSale\_CBySumWText()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_CBySumWText (
    string department,
    string singlePrice,
    string quantity,
    string abs )

```

Enter a product (sale)

#### Parameters

<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.219 receipt\_DSale\_Minimum()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_Minimum (
    string department,
    string singlePrice )

```

Enter a product (sale)

**Parameters**

<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign

**1.1.2.220 receipt\_DSale\_TextRow1()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow1 (
    string textRow1,
    string department,
    string singlePrice,
    string quantity )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

**1.1.2.221 receipt\_DSale\_TextRow1\_WQuan()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow1_WQuan (
    string textRow1,
    string department,
    string singlePrice )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign

**1.1.2.222 receipt\_DSale\_TextRow1CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow1CByPercent (
```



```

string textRow1,
string department,
string singlePrice,
string quantity,
string percent )

```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

#### 1.1.2.223 receipt\_DSale\_TextRow1CBySum()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow1CBySum (
    string textRow1,
    string department,
    string singlePrice,
    string quantity,
    string abs )

```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.224 receipt\_DSale\_TextRow1Un()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow1Un (
    string textRow1,
    string department,
    string singlePrice,

```

```

    string quantity,
    string measure )

```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

#### 1.1.2.225 receipt\_DSale\_TextRow1UnCByPercent()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow1UnCByPercent (
    string textRow1,
    string department,
    string singlePrice,
    string quantity,
    string measure,
    string percent )

```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

#### 1.1.2.226 receipt\_DSale\_TextRow1UnCBySum()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow1UnCBySum (
    string textRow1,
    string department,
    string singlePrice,

```

```

    string quantity,
    string measure,
    string abs )

```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.227 receipt\_DSale\_TextRow2()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow2 (
    string textRow2,
    string department,
    string singlePrice,
    string quantity )

```

Enter a product (sale)

#### Parameters

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

#### 1.1.2.228 receipt\_DSale\_TextRow2CByPercent()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow2CByPercent (
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
    string percent )

```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.229 receipt\_DSale\_TextRow2CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow2CBySum (
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
    string abs )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.230 receipt\_DSale\_TextRow2Un()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow2Un (
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
    string measure )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
-----------------	--

**Parameters**

<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

**1.1.2.231 receipt\_DSale\_TextRow2UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow2UnCByPercent (
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.232 receipt\_DSale\_TextRow2UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_TextRow2UnCBySum (
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
    string measure,
    string abs )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.233 receipt\_DSale\_Un()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_Un (
    string textRow1,
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
    string measure )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

**1.1.2.234 receipt\_DSale\_UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_UnCByPercent (
    string textRow1,
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.235 receipt\_DSale\_UnCByPercentWText()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_UnCByPercentWText (
    string department,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.236 receipt\_DSale\_UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_UnCBySum (
    string textRow1,
    string textRow2,
    string department,
    string singlePrice,
    string quantity,
```

```
string measure,
string abs )
```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.237 receipt\_DSale\_UnCBySumWText()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_UnCBySumWText (
    string department,
    string singlePrice,
    string quantity,
    string measure,
    string abs )
```

Enter a product (sale)

#### Parameters

<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.238 receipt\_DSale\_UnWText()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_DSale_UnWText (
    string department,
    string singlePrice,
```



```

    string quantity,
    string measure )

```

Enter a product (sale)

#### Parameters

<i>department</i>	Department number. Integer between 1 and 1200 inclusive. The sale is assigned to the tax group associated with the department when it is programmed
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

#### 1.1.2.239 receipt\_Fiscal\_01\_Open()

```

Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Fiscal_01_Open (
    string operatorNumber,
    string operatorPassword,
    string tillNumber )

```

Open fiscal receipt

#### Parameters

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)

#### Returns

Dictionary with keys:

<i>allReceiptCount</i>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<i>fiscalReceiptCount</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

#### 1.1.2.240 receipt\_Fiscal\_02\_Open()

```

Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Fiscal_02_Open (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string uNP )

```

Open fiscal receipt with UNP

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes), for example: DT000600-OP01-0001000)

**Returns**

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
fiscalReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.241 receipt\_Fiscal\_Cancel()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Fiscal_Cancel ( )
```

Cancel fiscal receipt

**1.1.2.242 receipt\_Fiscal\_Close()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Fiscal_Close ( )
```

Close fiscal receipt

**Returns**

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
fiscalReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.243 receipt\_Fiscal\_Text()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Fiscal_Text (
    string inputText )
```

Print free fiscal text

**Parameters**

<i>inputText</i>	Free text to print. The ':' character is printed at the start and the end of the line. The text may be with any length, but only the characters fitting on the line are used(without raising a truncation error).
------------------	---

**1.1.2.244 receipt\_FiscalOpen\_A02()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_FiscalOpen_A02 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice )
```

Open invoice

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice

**Returns**

Dictionary with keys:

<i>allReceiptCount</i>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<i>fiscalReceiptCount</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.245 receipt\_FiscalOpen\_A04()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_FiscalOpen_A04 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice,
    string uNP )
```

Open invoice with UNP

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)

**Parameters**

<i>invoice</i>	'I' - for invoice
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes), for example: DT000600-OP01-0001000)

**Returns**

Dictionary with keys:

<i>allReceiptCount</i>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<i>fiscalReceiptCount</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.246 receipt\_Invoice\_01\_Open()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Invoice_01_Open (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice )
```

Open invoice

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice

**Returns**

Dictionary with keys:

<i>allReceiptCount</i>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<i>fiscalReceiptCount</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.247 receipt\_Invoice\_Open()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Invoice_Open (
    string operatorNumber,
```

```

string operatorPassword,
string tillNumber,
string invoice,
string uNP )

```

Open invoice with UNP

#### Parameters

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes), for example: DT000600-OP01-0001000)

#### Returns

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
fiscalReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

#### 1.1.2.248 receipt\_NonFiscal\_Close()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_NonFiscal_Close ( )
```

Close non-fiscal receipt

#### Returns

Dictionary with key:

allReceipt	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
------------	---

#### 1.1.2.249 receipt\_NonFiscal\_Open()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_NonFiscal_Open ( )
```

Open non-fiscal receipt

#### Returns

Dictionary with key:

allReceipt	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
------------	---

#### 1.1.2.250 receipt\_NonFiscal\_Text()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_NonFiscal_Text (
    string inputText )
```

Print free non-fiscal receipt

##### Parameters

<i>inputText</i>	One line of text
------------------	------------------

#### 1.1.2.251 receipt\_NonFiscalRotated\_Close()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_NonFiscalRotated_Close ( )
```

Close rotated non-fiscal receipt

##### Returns

Dictionary with key:

rotRec	The sequential number of the closed 90-degree rotated receipt for the day. 4 bytes without a sign.
--------	--

#### 1.1.2.252 receipt\_NonFiscalRotated\_Open()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_NonFiscalRotated_Open ( )
```

Open non-fiscal rotated text

##### Returns

Dictionary with key:

rotRec	The sequential number of the closed 90-degree rotated receipt for the day. 4 bytes without a sign.
--------	--

**1.1.2.253 receipt\_NonFiscalRotated\_Text()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_NonFiscalRotated_Text (
    string inputText )
```

Print 90-degree rotated text

**Parameters**

<i>inputText</i>	Content of next line (up to 100 symbols)
------------------	--

**1.1.2.254 receipt\_Paper\_Cut()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Paper_Cut ( )
```

Paper cut

**1.1.2.255 receipt\_Paper\_Feed()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Paper_Feed (
    string linesCount )
```

Paper feed

**Parameters**

<i>linesCount</i>	The number of lines to feed the paper. Must be a positive number, at least 99 (1 or 2 bytes). If the parameter is missing, the default 1 line is used.
-------------------	--

**1.1.2.256 receipt\_Paper\_Feed\_InnerContainer()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Paper_Feed_InnerContainer (
    string param )
```

Paper feed (ONLY for models SK1-21F and SK1-31F)

**1.1.2.257 receipt\_PFiscalText\_01()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PFiscalText_01 (
    string font,
    string flags,
    string inputText )
```

Print free fiscal text

**Parameters**

<i>font</i>	Integer between 0 and 3: '0' - 32 dots(4 mm) height, higher letters; '1' - 32 dots(4 mm) height, normal letters '2' 24 dots(3 mm) height; '3' - 16 dots(2 mm) height
<i>flags</i>	Between one and 3 letters: 'B', 'H' or 'I'. Each may appear only once. Setting, respectively: B Bold(strong); H High(double height); I Italic(oblique)
<i>inputText</i>	Free text to print. The ':' character is printed at the start and the end of the line. The text may be with any length, but only the characters fitting on the line are used(without raising a truncation error).

**1.1.2.258 receipt\_PFiscalText\_02()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PFiscalText_02 (
    string font,
    string inputText )
```

Print free fiscal text

**Parameters**

<i>font</i>	Integer between 0 and 3: '0' - 32 dots(4 mm) height, higher letters; '1' - 32 dots(4 mm) height, normal letters '2' 24 dots(3 mm) height; '3' - 16 dots(2 mm) height
<i>inputText</i>	Free text to print. The ':' character is printed at the start and the end of the line. The text may be with any length, but only the characters fitting on the line are used(without raising a truncation error).

**1.1.2.259 receipt\_Pinpad\_PrintByNumber()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Pinpad_PrintByNumber (
    string number )
```

Print a receipt by number. The command is forbidden in an open fiscal receipt

**Parameters**

<i>number</i>	Receipt number
---------------	----------------

**Returns**

Dictionary with keys:

fAnswer	'P' - command passed or 'F' - command failed.
errorMessage	Can be text "No data for entered number" or error code.



**1.1.2.260 receipt\_Pinpad\_PrintLastSuccessfultransaction()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Pinpad_PrintLastSuccessfultransaction
( )
```

Printing the last successful transaction in a payment terminal.

It can only be executed if a receipt is open and the last transaction is completed without a reply from a payment terminal. If the A4h command (164 - receipt\_Pinpad\_ReturnLastSuccessfultransaction) has been executed before, or with card payment it will

also be successfully rejected. The receipt is printed automatically after the fiscal receipt has been closed.

Requires execution before returning command 148([info\\_Pinpad\\_CheckLastRecordedRRN\(\)](#)) for reference and return data collection.

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.261 receipt\_Pinpad\_PrintReceiptCopyFromEJ\_ByDate()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Pinpad_PrintReceiptCopy↔
FromEJ_ByDate (
    string date )
```

Printing a copy of an receipt from the EJ by date

**Parameters**

<i>date</i>	Date and time of original receipt in format DDMMYYhhmmss
-------------	--

**Returns**

Dictionary with key:

fAnswer	'P' - command passed or 'F' - command failed.
---------	---

**1.1.2.262 receipt\_Pinpad\_PrintReceiptCopyFromEJ\_ByNum()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Pinpad_PrintReceiptCopy↔
FromEJ_ByNum (
    string receiptNum )
```

Printing a copy of an receipt from the EJ by receipt number

**Parameters**

<i>receiptNum</i>	Global number of original receipt
-------------------	-----------------------------------

**Returns**

Dictionary with key:

fAnswer	'P' - command passed or 'F' - command failed.
---------	---

**1.1.2.263 receipt\_Pinpad\_ReturnLastSuccessfultransaction()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Pinpad_ReturnLastSuccessfultransaction
( )
```

Return the last successful transaction. It can only be executed if a receipt is open and the last transaction is completed without a reply from a payment terminal. If the A3h command (163 - [receipt\\_Pinpad\\_PrintLastSuccessfultransaction\(\)](#)) was previously executed or the card payment is successful, it will also be rejected. The receipt is printed automatically after the fiscal receipt has been closed. Requires execution before returning command 148([info\\_Pinpad\\_CheckLastRecordedRRN\(\)](#)) for reference and return data.

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.264 receipt\_PLU\_Sale()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLU_Sale (
    string targetPLU,
    string quantity )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits (no more than 3 after the decimal point).

**1.1.2.265 receipt\_PLUDep\_Sale()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUDep_Sale (
    string targetPLU,
    string department,
    string quantity )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

**1.1.2.266 receipt\_PLUDep\_Sale\_CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUDep_Sale_CByPercent (
    string targetPLU,
    string department,
    string quantity,
    string percent )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.267 receipt\_PLUDep\_Sale\_CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUDep_Sale_CBySum (
    string targetPLU,
    string department,
    string quantity,
    string abs )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.268 receipt\_PLUDep\_Sale\_Un()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUDep_Sale_Un (
    string targetPLU,
    string department,
    string quantity,
    string measure )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"

**1.1.2.269 receipt\_PLUDep\_Sale\_UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUDep_Sale_UnCByPercent (
    string targetPLU,
    string department,
    string quantity,
    string measure,
    string percent )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well

**Parameters**

<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale.The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

**1.1.2.270 receipt\_PLUDep\_Sale\_UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUDep_Sale_UnCBySum (
    string targetPLU,
    string department,
    string quantity,
    string measure,
    string abs )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>department</i>	Department number. Integer between 1 and 1200 inclusive. If this parameter is present, then the tab separators need to be there as well
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount.A discount amount exceeding the value of the sale is not allowed.

**1.1.2.271 receipt\_PLUSale\_CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUSale_CByPercent (
    string targetPLU,
    string quantity,
    string percent )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

**Parameters**

<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted
----------------	---

**1.1.2.272 receipt\_PLUSale\_CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUSale_CBySum (
    string targetPLU,
    string quantity,
    string abs )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.273 receipt\_PLUSale\_Un()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUSale_Un (
    string targetPLU,
    string quantity,
    string measure )
```

Registering the sale of a programmed item

**Parameters**

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"

**1.1.2.274 receipt\_PLUSale\_UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUSale_UnCByPercent (
```

```

    string targetPLU,
    string quantity,
    string measure,
    string percent )

```

Registering the sale of a programmed item

#### Parameters

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted

#### 1.1.2.275 receipt\_PLUSale\_UnCBySum()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PLUSale_UnCBySum (
    string targetPLU,
    string quantity,
    string measure,
    string abs )

```

Registering the sale of a programmed item

#### Parameters

<i>targetPLU</i>	This is the PLU code. Integer between 1 and 999999999 (up to 9 digits).
<i>quantity</i>	Parameter indicating the product quantity. By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Optional text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg"
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.276 receipt\_PNonFiscal\_Text()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PNonFiscal_Text (
    string height,
    string flags,
    string inputText )

```

**Parameters**

<i>height</i>	Integer between 0 and 3: '0' 32 dots(4 mm) height, higher letters; '1' 32 dots(4 mm) height, normal letters; '2' 24 dots(3 mm) height; '3' 16 dots(2 mm) height
<i>flags</i>	Between one and 3 letters: B Bold (strong);H High(double height); I Italic(oblique)
<i>inputText</i>	One line of text

**1.1.2.277 receipt\_Print\_Barcode\_01()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Print_Barcode_01 (
    string barcodeType,
    string alignment,
    string barcodeData )
```

**Print barcode****Parameters**

<i>barcodeType</i>	<p>'1' EAN8. The data contain only digits and are 7 bytes long. The checksum is calculated by the printer.</p> <p>'2' EAN13. The data contain only digits and are 12 bytes long. The checksum is calculated by the printer.</p> <p>'3' Code 128. The data are characters with ASCII codes between 32 and 127. Their length is between 22 and 42 characters (depends on the contents—the maximum length is achieved when all characters are digits). The checksum is calculated by the printer. When printing on narrow paper, the data length is obviously smaller.</p> <p>'4' ITF (Interleaved 2 of 5). The data contain only digits. '5' ITF (Interleaved 2 of 5). The data contain only digits. The printer automatically generates and prints a checksum. 'D' Two-dimensional barcode Data Matrix. The data are any printable characters (with length up to 140). 'Q' Two-dimensional barcode QR Code. The data are any printable characters (with length up to 140). 'P' Two-dimensional barcode PDF417. The data are any printable characters (with length up to 140).</p>
<i>alignment</i>	One byte: 'L', 'R' or 'C': left-, right-aligned or centred, respectively. It is allowed only in the two-dimensional barcodes. They are printed as text as well.
<i>barcodeData</i>	<p>EAN8 - The data contain only digits and are 7 bytes long; EAN13 - The data contain only digits and are 12 bytes long; Code 128 - The data are characters with ASCII codes between 32 and 127; ITF (Interleaved 2 of 5) - The data contain only digits; ITF (Interleaved 2 of 5). The data contain only digits. The printer automatically generates and prints a checksum</p> <p>'D' Two-dimensional barcode Data Matrix - data are any printable characters (with length up to 140)</p> <p>'Q' Two-dimensional barcode QR Code - data are any printable characters (with length up to 140)</p> <p>'P' Two-dimensional barcode PDF417 - data are any printable characters (with length up to 140)</p>

**1.1.2.278 receipt\_Print\_Barcode\_02()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Print_Barcode_02 (
```



```

string barcodeType,
string alignment,
string barcodeData )

```

Print barcode without text

#### Parameters

<i>barcodeType</i>	<p>'1' EAN8. The data contain only digits and are 7 bytes long. The checksum is calculated by the printer.</p> <p>'2' EAN13. The data contain only digits and are 12 bytes long. The checksum is calculated by the printer.</p> <p>'3' Code 128. The data are characters with ASCII codes between 32 and 127. Their length is between 22 and 42 characters (depends on the contents—the maximum length is achieved when all characters are digits). The checksum is calculated by the printer. When printing on narrow paper, the data length is obviously smaller.</p> <p>'4' ITF (Interleaved 2 of 5). The data contain only digits. '5' ITF (Interleaved 2 of 5). The data contain only digits. The printer automatically generates and prints a checksum. 'D' Two-dimensional barcode Data Matrix. The data are any printable characters (with length up to 140). 'Q' Two-dimensional barcode QR Code. The data are any printable characters (with length up to 140). 'P' Two-dimensional barcode PDF417. The data are any printable characters (with length up to 140).</p>
<i>alignment</i>	One byte: 'L', 'R' or 'C': left-, right-aligned or centred, respectively. It is allowed only in the two-dimensional barcodes.
<i>barcodeData</i>	<p>EAN8 - The data contain only digits and are 7 bytes long; EAN13 - The data contain only digits and are 12 bytes long; Code 128 - The data are characters with ASCII codes between 32 and 127; ITF (Interleaved 2 of 5) - The data contain only digits; ITF (Interleaved 2 of 5). The data contain only digits. The printer automatically generates and prints a checksum</p> <p>'D' Two-dimensional barcode Data Matrix - data are any printable characters (with length up to 140)</p> <p>'Q' Two-dimensional barcode QR Code - data are any printable characters (with length up to 140) 'P' Two-dimensional barcode PDF417 - data are any printable characters (with length up to 140)</p>

#### 1.1.2.279 receipt\_Print\_Duplicate()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Print_Duplicate (
    string count )

```

Print duplicate receipt

#### Parameters

<i>count</i>	Duplicate receipt count (allowed value: 1)
--------------	--

#### 1.1.2.280 receipt\_PrintClientInfo\_01()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PrintClientInfo_01 (

```

```
string eIKType,
string eIK )
```

Print customer information

#### Parameters

<i>eIKType</i>	If preceded by the ':' character, the data are treated as a PIN, if '*' - personal number, if '^' - work number
<i>eIK</i>	Customer's UIC. Between 9 and 14 characters

### 1.1.2.281 receipt\_PrintClientInfo\_02()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PrintClientInfo_02 (
    string eIKType,
    string eIK,
    string sellerName )
```

Print customer information

#### Parameters

<i>eIKType</i>	If preceded by the ':' character, the data are treated as a PIN, if '*' - personal number, if '^' - work number
<i>eIK</i>	Customer's UIC. Between 9 and 14 characters
<i>sellerName</i>	Seller's name. Up to 26 characters.

### 1.1.2.282 receipt\_PrintClientInfo\_03()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PrintClientInfo_03 (
    string eIKType,
    string eIK,
    string sellerName,
    string receiverName )
```

Print customer information

#### Parameters

<i>eIKType</i>	If preceded by the ':' character, the data are treated as a PIN, if '*' - personal number, if '^' - work number
<i>eIK</i>	Customer's UIC. Between 9 and 14 characters
<i>sellerName</i>	Seller's name. Up to 26 characters.
<i>receiverName</i>	Recipient's name. Up to 26 characters

**1.1.2.283 receipt\_PrintClientInfo\_04()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PrintClientInfo_04 (
    string eIKType,
    string eIK,
    string sellerName,
    string receiverName,
    string clientName )
```

Print customer information

**Parameters**

<i>eIKType</i>	If preceded by the ':' character, the data are treated as a PIN, if '*' - personal number, if '^' - work number
<i>eIK</i>	Customer's UIC. Between 9 and 14 characters
<i>sellerName</i>	Seller's name. Up to 26 characters.
<i>receiverName</i>	Recipient's name. Up to 26 characters
<i>clientName</i>	Customer's name. Up to 26 characters

**1.1.2.284 receipt\_PrintClientInfo\_05()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PrintClientInfo_05 (
    string eIKType,
    string eIK,
    string sellerName,
    string receiverName,
    string clientName,
    string taxNo )
```

Print customer information

**Parameters**

<i>eIKType</i>	If preceded by the ':' character, the data are treated as a PIN, if '*' - personal number, if '^' - work number
<i>eIK</i>	Customer's UIC. Between 9 and 14 characters
<i>sellerName</i>	Seller's name. Up to 26 characters.
<i>receiverName</i>	Recipient's name. Up to 26 characters
<i>clientName</i>	Customer's name. Up to 26 characters
<i>taxNo</i>	Customer's VAT number. Between 10 and 14 characters

**1.1.2.285 receipt\_PrintClientInfo\_06()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PrintClientInfo_06 (
    string eIKType,
```

```

string eIK,
string sellerName,
string receiverName,
string clientName,
string taxNo,
string address1 )

```

Print customer information

#### Parameters

<i>eIKType</i>	If preceded by the ':' character, the data are treated as a PIN, if '*' - personal number, if '^' - work number
<i>eIK</i>	Customer's UIC. Between 9 and 14 characters
<i>sellerName</i>	Seller's name. Up to 26 characters.
<i>receiverName</i>	Recipient's name. Up to 26 characters
<i>clientName</i>	Customer's name. Up to 26 characters
<i>taxNo</i>	Customer's VAT number. Between 10 and 14 characters
<i>address1</i>	Customer's address (up to 28 symbols)

#### 1.1.2.286 receipt\_PrintClientInfo\_07()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_PrintClientInfo_07 (
    string eIKType,
    string eIK,
    string sellerName,
    string receiverName,
    string clientName,
    string taxNo,
    string address1,
    string address2 )

```

Print customer information

#### Parameters

<i>eIKType</i>	If preceded by the ':' character, the data are treated as a PIN, if '*' - personal number, if '^' - work number
<i>eIK</i>	Customer's UIC. Between 9 and 14 characters
<i>sellerName</i>	Seller's name. Up to 26 characters.
<i>receiverName</i>	Recipient's name. Up to 26 characters
<i>clientName</i>	Customer's name. Up to 26 characters
<i>taxNo</i>	Customer's VAT number. Between 10 and 14 characters
<i>address1</i>	Customer's address (up to 28 symbols)
<i>address2</i>	Customer's address second line (up to 34 symbols)

**1.1.2.287 receipt\_Sale()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale (
    string textRow1,
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

**1.1.2.288 receipt\_Sale\_CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_CByPercent (
    string textRow1,
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted.

**1.1.2.289 receipt\_Sale\_CByPercentWText()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_CByPercentWText (
    string taxGroup,
```

```

    string singlePrice,
    string quantity,
    string percent )

```

Enter a product (sale)

#### Parameters

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted.

#### 1.1.2.290 receipt\_Sale\_CBySum()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_CBySum (
    string textRow1,
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string abs )

```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

#### 1.1.2.291 receipt\_Sale\_CBySumWText()

```

void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_CBySumWText (
    string taxGroup,
    string singlePrice,
    string quantity,
    string abs )

```

Enter a product (sale)

**Parameters**

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.292 receipt\_Sale\_Minimum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_Minimum (
    string taxGroup,
    string singlePrice )
```

Enter a product (sale)

**Parameters**

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign

**1.1.2.293 receipt\_Sale\_TextRow1()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow1 (
    string textRow1,
    string taxGroup,
    string singlePrice,
    string quantity )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

**1.1.2.294 receipt\_Sale\_TextRow1\_WQuan()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow1_WQuan (
    string textRow1,
```

```
string taxGroup,
string singlePrice )
```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign

#### 1.1.2.295 receipt\_Sale\_TextRow1CByPercent()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow1CByPercent (
    string textRow1,
    string taxGroup,
    string singlePrice,
    string quantity,
    string percent )
```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted.

#### 1.1.2.296 receipt\_Sale\_TextRow1CBySum()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow1CBySum (
    string textRow1,
    string taxGroup,
    string singlePrice,
    string quantity,
    string abs )
```

Enter a product (sale)

#### Parameters

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
-----------------	---



**Parameters**

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.297 receipt\_Sale\_TextRow1Un()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow1Un (
    string textRow1,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

**1.1.2.298 receipt\_Sale\_TextRow1UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow1UnCByPercent (
    string textRow1,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign

**Parameters**

<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted.

**1.1.2.299 receipt\_Sale\_TextRow1UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow1UnCBySum (
    string textRow1,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string abs )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.300 receipt\_Sale\_TextRow2()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow2 (
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
-----------------	--

**Parameters**

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).

**1.1.2.301 receipt\_Sale\_TextRow2CByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow2CByPercent (
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted.

**1.1.2.302 receipt\_Sale\_TextRow2CBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow2CBySum (
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string abs )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**1.1.2.303 receipt\_Sale\_TextRow2Un()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow2Un (
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

**1.1.2.304 receipt\_Sale\_TextRow2UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow2UnCByPercent (
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted.

**1.1.2.305 receipt\_Sale\_TextRow2UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_TextRow2UnCBySum (
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string abs )
```

Enter a product (sale)

**Parameters**

<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount.A discount amount exceeding the value of the sale is not allowed.

**1.1.2.306 receipt\_Sale\_Un()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_Un (
    string textRow1,
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

**1.1.2.307 receipt\_Sale\_UnCByPercent()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_UnCByPercent (
    string textRow1,
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted.

**1.1.2.308 receipt\_Sale\_UnCByPercentWText()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_UnCByPercentWText (
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string percent )
```

Enter a product (sale)

**Parameters**

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>percent</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as a percentage of the current sale. The allowable values are between - 99.00% and 99.00%. Up to 2 decimal digits are accepted.

**1.1.2.309 receipt\_Sale\_UnCBySum()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_UnCBySum (
    string textRow1,
    string textRow2,
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string abs )
```

Enter a product (sale)

**Parameters**

<i>textRow1</i>	Text, up to 42 bytes, containing a line describing the sale
<i>textRow2</i>	Text, up to 42 bytes, containing a second line describing the sale
<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount.A discount amount exceeding the value of the sale is not allowed.

**1.1.2.310 receipt\_Sale\_UnCBySumWText()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_UnCBySumWText (
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure,
    string abs )
```

Enter a product (sale)

**Parameters**

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".
<i>abs</i>	This is an optional parameter indicating the value of the mark-up or discount (depending on the sign), as an amount.A discount amount exceeding the value of the sale is not allowed.

**1.1.2.311 receipt\_Sale\_UnWText()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Sale_UnWText (
    string taxGroup,
    string singlePrice,
    string quantity,
    string measure )
```

Enter a product (sale)

**Parameters**

<i>taxGroup</i>	One byte containing the letter indicating the tax type ('A', 'B', 'C', ...).
<i>singlePrice</i>	This is the unit price, with up to 8 significant digits and sign
<i>quantity</i>	By default, it is 1.000. Length, up to 8 significant digits(no more than 3 after the decimal point).
<i>measure</i>	Name of the unit of measurement. Text for the unit of measurement for the quantity, up to 8 characters, for instance, "kg".

**1.1.2.312 receipt\_Separating\_Line()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Separating_Line (
    string lineType )
```

Print separating line

**Parameters**

<i>lineType</i>	'1' Fill with the '-' character; '2' Fill with alternating '-' and ' ' characters; '3' Fill with the '=' character. '4' Fill with the double-width '*' character
-----------------	---

**1.1.2.313 receipt\_Separating\_LineW()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Separating_LineW (
    string lineType,
    string linemask )
```

Print separating line. The text "not payable!" with a triple height is added unconditionally

**Parameters**

<i>lineType</i>	"W" - mask
<i>linemask</i>	bit mask:number btw 0 and 3 - 1: adds the text "under this receipt"; 2: adds a frame of double-width '*' characters above and below



**1.1.2.314 receipt\_StornoOpen\_A01()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A01 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string stornoType,
    string stornoDocumentNumber )
```

Open storno document - variant 1

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document

**Returns**

Dictionary with keys:

<i>allReceiptCount</i>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<i>stornoReceiptCount</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.315 receipt\_StornoOpen\_A02()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A02 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string uNP,
    string stornoType,
    string stornoDocumentNumber )
```

Open storno document - variant 2

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes), for example: DT000600-OP01-0001000).
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed

**Returns**

Dictionary with keys:

<code>allReceiptCount</code>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<code>stornoReceiptCount</code>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.316 receipt\_StornoOpen\_A03()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A03 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string stornoType,
    string stornoDocumentNumber,
    string stornoReason )
```

Open storno document - variant 3

**Parameters**

<code>operatorNumber</code>	Operator ID (between 1 and 16)
<code>operatorPassword</code>	Operator password (between 4 and 8 digits)
<code>tillNumber</code>	Cash register location number (integer between 1 and 99999)
<code>stornoType</code>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<code>stornoDocumentNumber</code>	(Global) number of the document being reversed
<code>stornoReason</code>	The number of all fiscal receipts since the last closing time until now. (4 bytes)

**Returns**

Dictionary with keys:

<code>allReceiptCount</code>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<code>stornoReceiptCount</code>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.317 receipt\_StornoOpen\_A04()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A04 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string uNP,
    string stornoType,
    string stornoDocumentNumber,
    string stornoReason )
```

Open storno document - variant 4

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>uNP</i>	Leave blank
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoReason</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes)

**Returns**

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
stornoReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.318 receipt\_StornoOpen\_A05()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A05 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string stornoType,
    string stornoDocumentNumber,
    string stornoUNP,
    string stornoDateTime,
    string stornoFMNumber )
```

Open storno document - variant 5

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoUNP</i>	Unique Sale ID of the document being reversed. The whole Unique Sale ID must be set.
<i>stornoDateTime</i>	Date and time of the reversed document. Format "DDMMYYhhmmss".
<i>stornoFMNumber</i>	Fiscal Memory ID of the FD which has issued the storno receipt.

**Returns**

Dictionary with keys:

<code>allReceiptCount</code>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<code>stornoReceiptCount</code>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.319 receipt\_StornoOpen\_A06()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A06 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string uNP,
    string stornoType,
    string stornoDocumentNumber,
    string stornoUNP,
    string stornoDateTime,
    string stornoFMNumber )
```

Open storno document - variant 6

**Parameters**

<code>operatorNumber</code>	Operator ID (between 1 and 16)
<code>operatorPassword</code>	Operator password (between 4 and 8 digits)
<code>tillNumber</code>	Cash register location number (integer between 1 and 99999)
<code>uNP</code>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes)
<code>stornoType</code>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<code>stornoDocumentNumber</code>	(Global) number of the document being reversed
<code>stornoUNP</code>	Unique Sale ID of the document being reversed. The whole Unique Sale ID must be set.
<code>stornoDateTime</code>	Date and time of the reversed document. Format "DDMMYYhhmmss".
<code>stornoFMNumber</code>	Fiscal Memory ID of the FD which has issued the storno receipt.

**Returns**

Dictionary with keys:

<code>allReceiptCount</code>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<code>stornoReceiptCount</code>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.320 receipt\_StornoOpen\_A07()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A07 (
    string operatorNumber,
```

```

string operatorPassword,
string tillNumber,
string stornoType,
string stornoDocumentNumber,
string stornoUNP,
string stornoDateTime,
string stornoFMNumber,
string stornoReason )

```

Open storno document - variant 7

#### Parameters

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoUNP</i>	Unique Sale ID of the document being reversed. The whole Unique Sale ID must be set.
<i>stornoDateTime</i>	Date and time of the reversed document. Format "DDMMYYhhmmss".
<i>stornoFMNumber</i>	Fiscal Memory ID of the FD which has issued the storno receipt.
<i>stornoReason</i>	Reason (up to 30 characters).

#### Returns

Dictionary with keys:

<i>allReceiptCount</i>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<i>stornoReceiptCount</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

### 1.1.2.321 receipt\_StornoOpen\_A08()

```

Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A08 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string uNP,
    string stornoType,
    string stornoDocumentNumber,
    string stornoUNP,
    string stornoDateTime,
    string stornoFMNumber,
    string stornoReason )

```

Open storno document - variant 8

#### Parameters

<i>operatorNumber</i>	Operator ID (between 1 and 16)
-----------------------	--------------------------------

**Parameters**

<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes)
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoUNP</i>	Unique Sale ID of the document being reversed. The whole Unique Sale ID must be set.
<i>stornoDateTime</i>	Date and time of the reversed document. Format "DDMMYYhhmmss".
<i>stornoFMNumber</i>	Fiscal Memory ID of the FD which has issued the storno receipt.
<i>stornoReason</i>	Reason (up to 30 characters).

**Returns**

Dictionary with keys:

<i>allReceiptCount</i>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<i>stornoReceiptCount</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.322 receipt\_StornoOpen\_A09()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A09 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice,
    string invoiceNumber,
    string stornoType,
    string stornoDocumentNumber )
```

Open storno document - variant 9

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>invoiceNumber</i>	Invoice number
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed

**Returns**

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
stornoReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

### 1.1.2.323 receipt\_StornoOpen\_A10()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A10 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice,
    string invoiceNumber,
    string uNP,
    string stornoType,
    string stornoDocumentNumber )
```

Open storno document - variant 10

#### Parameters

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>invoiceNumber</i>	Invoice number
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes)
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed

#### Returns

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
stornoReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

### 1.1.2.324 receipt\_StornoOpen\_A11()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A11 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
```



```

string invoice,
string invoiceNumber,
string stornoType,
string stornoDocumentNumber,
string stornoReason )

```

Open storno document - variant 11

#### Parameters

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>invoiceNumber</i>	Invoice number
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoReason</i>	Reason (up to 30 symbols)

#### Returns

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
stornoReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

#### 1.1.2.325 receipt\_StornoOpen\_A12()

```

Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A12 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice,
    string invoiceNumber,
    string uNP,
    string stornoType,
    string stornoDocumentNumber,
    string stornoReason )

```

Open storno document - variant 12

#### Parameters

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>invoiceNumber</i>	Invoice number

**Parameters**

<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes)
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoReason</i>	Reason (up to 30 symbols)

**Returns**

Dictionary with keys:

<i>allReceiptCount</i>	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
<i>stornoReceiptCount</i>	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.326 receipt\_StornoOpen\_A13()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A13 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice,
    string invoiceNumber,
    string stornoType,
    string stornoDocumentNumber,
    string stornoUNP,
    string stornoDateTime,
    string stornoFMNumber )
```

Open storno document - variant 13

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>invoiceNumber</i>	Invoice number
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoUNP</i>	Unique Sale ID of the document being reversed. The whole Unique Sale ID must be set.
<i>stornoDateTime</i>	Date and time of the reversed document. Format "DDMMYYhhmmss".
<i>stornoFMNumber</i>	Fiscal Memory ID of the FD which has issued the storno receipt.

**Returns**

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
stornoReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.327 receipt\_StornoOpen\_A14()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A14 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice,
    string invoiceNumber,
    string uNP,
    string stornoType,
    string stornoDocumentNumber,
    string stornoUNP,
    string stornoDateTime,
    string stornoFMNumber )
```

Open storno document - variant 14

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>invoiceNumber</i>	Invoice number
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes)
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoUNP</i>	Unique Sale ID of the document being reversed. The whole Unique Sale ID must be set.
<i>stornoDateTime</i>	Date and time of the reversed document. Format "DDMMYYhhmmss".
<i>stornoFMNumber</i>	Fiscal Memory ID of the FD which has issued the storno receipt.

**Returns**

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
stornoReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.328 receipt\_StornoOpen\_A15()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A15 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice,
    string invoiceNumber,
    string stornoType,
    string stornoDocumentNumber,
    string stornoUNP,
    string stornoDateTime,
    string stornoFMNumber,
    string stornoReason )
```

Open storno document - variant 15

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>invoiceNumber</i>	Invoice number
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoUNP</i>	Unique Sale ID of the document being reversed. The whole Unique Sale ID must be set.
<i>stornoDateTime</i>	Date and time of the reversed document. Format "DDMMYYhhmmss".
<i>stornoFMNumber</i>	Fiscal Memory ID of the FD which has issued the storno receipt.
<i>stornoReason</i>	Reason (up to 30 symbols)

**Returns****1.1.2.329 receipt\_StornoOpen\_A16()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_StornoOpen_A16 (
    string operatorNumber,
    string operatorPassword,
    string tillNumber,
    string invoice,
    string invoiceNumber,
    string uNP,
    string stornoType,
    string stornoDocumentNumber,
    string stornoUNP,
    string stornoDateTime,
    string stornoFMNumber,
    string stornoReason )
```

Open storno document - variant 16

**Parameters**

<i>operatorNumber</i>	Operator ID (between 1 and 16)
<i>operatorPassword</i>	Operator password (between 4 and 8 digits)
<i>tillNumber</i>	Cash register location number (integer between 1 and 99999)
<i>invoice</i>	'I' - for invoice
<i>invoiceNumber</i>	Invoice number
<i>uNP</i>	Unique Sale ID (format: FD Serial Number-four digits or Latin letters-sequential number of the sale(seven digits with leading zeroes)
<i>stornoType</i>	'E': operator error; 'R': return/claim; 'T': tax base reduction
<i>stornoDocumentNumber</i>	(Global) number of the document being reversed
<i>stornoUNP</i>	Unique Sale ID of the document being reversed. The whole Unique Sale ID must be set.
<i>stornoDateTime</i>	Date and time of the reversed document. Format "DDMMYYhhmmss".
<i>stornoFMNumber</i>	Fiscal Memory ID of the FD which has issued the storno receipt.
<i>stornoReason</i>	Reason (up to 30 symbols)

**Returns**

Dictionary with keys:

allReceiptCount	The number of all issued receipts (fiscal and service) since the last closing time until now (4 bytes).
stornoReceiptCount	The number of all fiscal receipts since the last closing time until now. (4 bytes).

**1.1.2.330 receipt\_Subtotal()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Subtotal (
    string toPrint,
    string toDisplay )
```

**Subtotal****Parameters**

<i>toPrint</i>	One byte: if it is '1', the subtotal will be printed
<i>toDisplay</i>	One byte: if it is '1', the subtotal will be displayed

**Returns**

Dictionary with keys:

subtotal	The sum until now for the current fiscal receipt (up to 10 bytes)
sumTaxGroupA	Turnover for tax group A (up to 10 bytes)
sumTaxGroupB	Turnover for tax group B (up to 10 bytes)
sumTaxGroupC	Turnover for tax group C (up to 10 bytes)
sumTaxGroupD	Turnover for tax group D (up to 10 bytes)

sumTaxGroupE	Turnover for tax group E (up to 10 bytes)
sumTaxGroupF	Turnover for tax froup F (up to 10 bytes)
sumTaxGroupG	Turnover for tax froup G (up to 10 bytes)
sumTaxGroupH	Turnover for tax froup H (up to 10 bytes)

### 1.1.2.331 receipt\_Subtotal\_CByPercent()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Subtotal_CByPercent (
    string toPrint,
    string toDisplay,
    string percent )
```

Subtotal

Parameters

<i>toPrint</i>	One byte: if it is '1', the subtotal will be printed.
<i>toDisplay</i>	One byte: if it is '1', the subtotal will be displayed.
<i>percent</i>	Parameter indicating the percentage value of the discount or mark-up on the sum accumulated until now

Returns

Dictionary with keys:

subtotal	The sum until now for the current fiscal receipt (up to 10 bytes)
sumTaxGroupA	Turnover for tax group A (up to 10 bytes)
sumTaxGroupB	Turnover for tax group B (up to 10 bytes)
sumTaxGroupC	Turnover for tax group C (up to 10 bytes)
sumTaxGroupD	Turnover for tax group D (up to 10 bytes)
sumTaxGroupE	Turnover for tax group E (up to 10 bytes)
sumTaxGroupF	Turnover for tax froup F (up to 10 bytes)
sumTaxGroupG	Turnover for tax froup G (up to 10 bytes)
sumTaxGroupH	Turnover for tax froup H (up to 10 bytes)

### 1.1.2.332 receipt\_Subtotal\_CBySum()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Subtotal_CBySum (
    string toPrint,
    string toDisplay,
    string abs )
```

Subtotal

**Parameters**

<i>toPrint</i>	One byte: if it is '1', the subtotal will be printed.
<i>toDisplay</i>	One byte: if it is '1', the subtotal will be displayed.
<i>abs</i>	Parameter indicating the value of the mark-up or discount (depending on the sign), as an amount. A discount amount exceeding the value of the sale is not allowed.

**Returns**

Dictionary with keys:

subtotal	The sum until now for the current fiscal receipt (up to 10 bytes)
sumTaxGroupA	Turnover for tax group A (up to 10 bytes)
sumTaxGroupB	Turnover for tax group B (up to 10 bytes)
sumTaxGroupC	Turnover for tax group C (up to 10 bytes)
sumTaxGroupD	Turnover for tax group D (up to 10 bytes)
sumTaxGroupE	Turnover for tax group E (up to 10 bytes)
sumTaxGroupF	Turnover for tax froup F (up to 10 bytes)
sumTaxGroupG	Turnover for tax froup G (up to 10 bytes)
sumTaxGroupH	Turnover for tax froup H (up to 10 bytes)

**1.1.2.333 receipt\_Total()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Total (
    string textRow1,
    string textRow2 )
```

**Total****Parameters**

<i>textRow1</i>	Text, up to 36 bytes, containing the first line
<i>textRow2</i>	Text, up to 36 bytes, containing the second line

**Returns**

Dictionary with keys:

deviceCode	'F' Error; 'E' The calculated subtotal is negative.No payment is made and Amount will contain the negative subtotal 'D' If the amount paid is less than the amount on the receipt.The remaining amount to be paid is returned to Amount. 'R' If the amount paid is more than the sum on the receipt.↵ A "CHANGE" message will be printed and the change is returned to Amount. 'I' The sum under any tax group is negative, causing an error.The current subtotal is returned to Amount.
outputAmount	Up to 9 digits with a sign. Depends on 'deviceCode'

**1.1.2.334 receipt\_Total\_PAmount()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Total_PAmount (
    string textRow1,
    string textRow2,
    string paidMode,
    string inputAmount )
```

Total

**Parameters**

<i>textRow1</i>	Text, up to 36 bytes, containing the first line
<i>textRow2</i>	Text, up to 36 bytes, containing the second line
<i>paidMode</i>	Optional code indicating the payment method: 'P' - Cash payment (default); 'N' - Credit payment; 'C' - Check payment; 'D' - Debit card payment; 'I' - Programmable payment type 1; 'J' - Programmable payment type 2; 'K' - Programmable payment type 3; 'L' - Programmable payment type 4; 'i' - Programmable payment type 1; 'j' - Programmable payment type 2 'k' - Programmable payment type 3; 'l' - Programmable payment type 4; 'm' - Coupons; 'n' - External coupons; 'o' - Packing; 'p' - Internal service 'q' - Damages; 'r' - Bank transfers; 's' - Check
<i>inputAmount</i>	The amount payable (up to 10 significant digits).

**Returns**

Dictionary with keys:

<i>deviceCode</i>	'F' Error; 'E' The calculated subtotal is negative.No payment is made and Amount will contain the negative subtotal 'D' If the amount paid is less than the amount on the receipt.The remaining amount to be paid is returned to Amount. 'R' If the amount paid is more than the sum on the receipt.↔ A "CHANGE" message will be printed and the change is returned to Amount. 'I' The sum under any tax group is negative, causing an error.The current subtotal is returned to Amount.
<i>outputAmount</i>	Up to 9 digits with a sign. Depends on 'deviceCode'

**1.1.2.335 receipt\_Total\_PAmountTextRow1()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Total_PAmountTextRow1 (
    string textRow1,
    string paidMode,
    string inputAmount )
```

**Parameters**

<i>textRow1</i>	Text, up to 36 bytes, containing the first line
<i>paidMode</i>	Optional code indicating the payment method: 'P' - Cash payment (default); 'N' - Credit payment; 'C' - Check payment; 'D' - Debit card payment; 'I' - Programmable payment type 1; 'J' - Programmable payment type 2; 'K' - Programmable payment type 3; 'L' - Programmable payment type 4; 'i' - Programmable payment type 1; 'j' - Programmable payment type 2 'k' - Programmable payment type 3; 'l' - Programmable payment type 4; 'm' - Coupons; 'n' - External coupons; 'o' - Packing; 'p' - Internal service 'q' - Damages; 'r' - Bank transfers; 's' - Check
<i>inputAmount</i>	The amount payable (up to 10 significant digits).



**Returns**

Dictionary with keys:

deviceCode	'F' Error; 'E' The calculated subtotal is negative.No payment is made and Amount will contain the negative subtotal 'D' If the amount paid is less than the amount on the receipt.The remaining amount to be paid is returned to Amount. 'R' If the amount paid is more than the sum on the receipt.↵ A "CHANGE" message will be printed and the change is returned to Amount. 'I' The sum under any tax group is negative, causing an error.The current subtotal is returned to Amount.
outputAmount	Up to 9 digits with a sign. Depends on 'deviceCode'

**1.1.2.336 receipt\_Total\_PAmountTextRow2()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Total_PAmountTextRow2 (
    string textRow2,
    string paidMode,
    string inputAmount )
```

Total

**Parameters**

<i>textRow2</i>	Text, up to 36 bytes, containing the second line
<i>paidMode</i>	Optional code indicating the payment method: 'P' - Cash payment (default); 'N' - Credit payment; 'C' - Check payment; 'D' - Debit card payment; 'I' - Programmable payment type 1; 'J' - Programmable payment type 2; 'K' - Programmable payment type 3; 'L' - Programmable payment type 4; 'i' - Programmable payment type 1; 'j' - Programmable payment type 2 'k' - Programmable payment type 3; 'l' - Programmable payment type 4; 'm' - Coupons; 'n' - External coupons; 'o' - Packing; 'p' - Internal service 'q' - Damages; 'r' - Bank transfers; 's' - Check
<i>inputAmount</i>	The amount payable (up to 10 significant digits).

**Returns**

Dictionary with keys:

deviceCode	'F' Error; 'E' The calculated subtotal is negative.No payment is made and Amount will contain the negative subtotal 'D' If the amount paid is less than the amount on the receipt.The remaining amount to be paid is returned to Amount. 'R' If the amount paid is more than the sum on the receipt.↵ A "CHANGE" message will be printed and the change is returned to Amount. 'I' The sum under any tax group is negative, causing an error.The current subtotal is returned to Amount.
outputAmount	Up to 9 digits with a sign. Depends on 'deviceCode'

**1.1.2.337 receipt\_Total\_PAmountWithoutText()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Total_PAmountWithoutText (
```

```
string paidMode,
string inputAmount )
```

Total

#### Parameters

<i>paidMode</i>	Optional code indicating the payment method: 'P' - Cash payment (default); 'N' - Credit payment; 'C' - Check payment; 'D' - Debit card payment; 'I' - Programmable payment type 1; 'J' - Programmable payment type 2; 'K' - Programmable payment type 3; 'L' - Programmable payment type 4; 'i' - Programmable payment type 1; 'j' - Programmable payment type 2 'k' - Programmable payment type 3; 'l' - Programmable payment type 4; 'm' - Coupons; 'n' - External coupons; 'o' - Packing; 'p' - Internal service 'q' - Damages; 'r' - Bank transfers; 's' - Check
<i>inputAmount</i>	The amount payable (up to 10 significant digits).

#### Returns

Dictionary with keys:

<i>deviceCode</i>	'F' Error; 'E' The calculated subtotal is negative.No payment is made and Amount will contain the negative subtotal 'D' If the amount paid is less than the amount on the receipt.The remaining amount to be paid is returned to Amount. 'R' If the amount paid is more than the sum on the receipt.↔ A "CHANGE" message will be printed and the change is returned to Amount. 'I' The sum under any tax group is negative, causing an error.The current subtotal is returned to Amount.
<i>outputAmount</i>	Up to 9 digits with a sign. Depends on 'deviceCode'

#### 1.1.2.338 receipt\_Total\_TextRow1()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Total_TextRow1 (
    string textRow1 )
```

Total

#### Parameters

<i>textRow1</i>	Text, up to 36 bytes, containing the first line
-----------------	---

#### Returns

Dictionary with keys:

<i>deviceCode</i>	'F' Error; 'E' The calculated subtotal is negative.No payment is made and Amount will contain the negative subtotal 'D' If the amount paid is less than the amount on the receipt.The remaining amount to be paid is returned to Amount. 'R' If the amount paid is more than the sum on the receipt.↔ A "CHANGE" message will be printed and the change is returned to Amount. 'I' The sum under any tax group is negative, causing an error.The current subtotal is returned to Amount.
<i>outputAmount</i>	Up to 9 digits with a sign. Depends on 'deviceCode'

**1.1.2.339 receipt\_Total\_TextRow2()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Total_TextRow2 (
    string textRow2 )
```

Total

**Parameters**

<i>textRow2</i>	Text, up to 36 bytes, containing the second line
-----------------	--

**Returns**

Dictionary with keys:

deviceCode	'F' Error; 'E' The calculated subtotal is negative.No payment is made and Amount will contain the negative subtotal 'D' If the amount paid is less than the amount on the receipt.The remaining amount to be paid is returned to Amount. 'R' If the amount paid is more than the sum on the receipt.↔ A "CHANGE" message will be printed and the change is returned to Amount. 'I' The sum under any tax group is negative, causing an error.The current subtotal is returned to Amount.
outputAmount	Up to 9 digits with a sign. Depends on 'deviceCode'

**1.1.2.340 receipt\_Total\_WithoutText()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.receipt_Total_WithoutText ( )
```

Total

**Returns**

Dictionary with keys:

deviceCode	'F' Error; 'E' The calculated subtotal is negative.No payment is made and Amount will contain the negative subtotal 'D' If the amount paid is less than the amount on the receipt.The remaining amount to be paid is returned to Amount. 'R' If the amount paid is more than the sum on the receipt.↔ A "CHANGE" message will be printed and the change is returned to Amount. 'I' The sum under any tax group is negative, causing an error.The current subtotal is returned to Amount.
outputAmount	Up to 9 digits with a sign. Depends on 'deviceCode'

**1.1.2.341 report\_DailyClosure\_01()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_DailyClosure_01 (
    string option )
```

Daily closure

**Parameters**

<i>option</i>	'0' - Z report or '2' - X report
---------------	----------------------------------

**Returns**

Dictionary with keys:

closure	The fiscal record number: 4 bytes.
fMTotals	The sum of all non-VAT sales: 12 bytes with a sign.
totalsumA	The sums under each tax group A: 12 bytes with a sign.
totalsumB	The sums under each tax group B: 12 bytes with a sign.
totalsumC	The sums under each tax group C: 12 bytes with a sign.
totalsumD	The sums under each tax group D: 12 bytes with a sign.
totalsumE	The sums under each tax group E: 12 bytes with a sign.
totalsumF	The sums under each tax group F: 12 bytes with a sign.
totalsumG	The sums under each tax group G: 12 bytes with a sign.
totalsumH	The sums under each tax group H: 12 bytes with a sign.

**1.1.2.342 report\_DailyClosure\_02()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_DailyClosure_02 (
    string option,
    string withoutClearOpInfo )
```

Daily closure

**Parameters**

<i>option</i>	'0' - Z report
<i>withoutClearOpInfo</i>	'N' - disables the clearing of accumulated data by operator

**Returns**

Dictionary with keys:

closure	The fiscal record number: 4 bytes.
fMTotals	The sum of all non-VAT sales: 12 bytes with a sign.
totalsumA	The sums under each tax group A: 12 bytes with a sign.
totalsumB	The sums under each tax group B: 12 bytes with a sign.

totalsumC	The sums under each tax group C: 12 bytes with a sign.
totalsumD	The sums under each tax group D: 12 bytes with a sign.
totalsumE	The sums under each tax group E: 12 bytes with a sign.
totalsumF	The sums under each tax group F: 12 bytes with a sign.
totalsumG	The sums under each tax group G: 12 bytes with a sign.
totalsumH	The sums under each tax group H: 12 bytes with a sign.

### 1.1.2.343 report\_DailyClosureByDepartments\_01()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_DailyClosureByDepartments_01
(
    string option )
```

Daily financial report with data printed by department

#### Parameters

<i>option</i>	'0' - Z report or '2' - X report
---------------	----------------------------------

#### Returns

Dictionary with keys:

closure	The fiscal record number: 4 bytes.
fMTotal	The sum of all non-VAT sales: 12 bytes with a sign.
totalsumA	The sums under each tax group A: 12 bytes with a sign.
totalsumB	The sums under each tax group B: 12 bytes with a sign.
totalsumC	The sums under each tax group C: 12 bytes with a sign.
totalsumD	The sums under each tax group D: 12 bytes with a sign.
totalsumE	The sums under each tax group E: 12 bytes with a sign.
totalsumF	The sums under each tax group F: 12 bytes with a sign.
totalsumG	The sums under each tax group G: 12 bytes with a sign.
totalsumH	The sums under each tax group H: 12 bytes with a sign.

### 1.1.2.344 report\_DailyClosureByDepartments\_02()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_DailyClosureByDepartments_02
(
    string option,
    string withoutClearOpInfo )
```

Daily financial report with data printed by department

**Parameters**

<i>option</i>	'0' - Z report
<i>withoutClearOpInfo</i>	'N' - disables the clearing of accumulated data by operator

**Returns**

Dictionary with keys:

closure	The fiscal record number: 4 bytes.
fMTotals	The sum of all non-VAT sales: 12 bytes with a sign.
totalsumA	The sums under each tax group A: 12 bytes with a sign.
totalsumB	The sums under each tax group B: 12 bytes with a sign.
totalsumC	The sums under each tax group C: 12 bytes with a sign.
totalsumD	The sums under each tax group D: 12 bytes with a sign.
totalsumE	The sums under each tax group E: 12 bytes with a sign.
totalsumF	The sums under each tax group F: 12 bytes with a sign.
totalsumG	The sums under each tax group G: 12 bytes with a sign.
totalsumH	The sums under each tax group H: 12 bytes with a sign.

**1.1.2.345 report\_DailyClosureByDepartmentsAndItems\_01()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_DailyClosureByDepartments←
AndItems_01 (
    string option )
```

Daily financial report with department and PLU printing

**Parameters**

<i>option</i>	'0' - Z report or '2' - X report
---------------	----------------------------------

**Returns**

Dictionary with keys:

closure	The fiscal record number: 4 bytes.
fMTotals	The sum of all non-VAT sales: 12 bytes with a sign.
totalsumA	The sums under each tax group A: 12 bytes with a sign.
totalsumB	The sums under each tax group B: 12 bytes with a sign.
totalsumC	The sums under each tax group C: 12 bytes with a sign.
totalsumD	The sums under each tax group D: 12 bytes with a sign.
totalsumE	The sums under each tax group E: 12 bytes with a sign.
totalsumF	The sums under each tax group F: 12 bytes with a sign.
totalsumG	The sums under each tax group G: 12 bytes with a sign.
totalsumH	The sums under each tax group H: 12 bytes with a sign.

**1.1.2.346 report\_DailyClosureByDepartmentsAndItems\_02()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_DailyClosureByDepartmentsAndItems_02 (
    string option,
    string withoutClearOpInfo )
```

Daily financial report with department and PLU printing

**Parameters**

<i>option</i>	'0' - Z report
<i>withoutClearOpInfo</i>	'N' - disables the clearing of accumulated data by operator

**Returns**

Dictionary with keys:

closure	The fiscal record number: 4 bytes.
fMTotals	The sum of all non-VAT sales: 12 bytes with a sign.
totalsumA	The sums under each tax group A: 12 bytes with a sign.
totalsumB	The sums under each tax group B: 12 bytes with a sign.
totalsumC	The sums under each tax group C: 12 bytes with a sign.
totalsumD	The sums under each tax group D: 12 bytes with a sign.
totalsumE	The sums under each tax group E: 12 bytes with a sign.
totalsumF	The sums under each tax group F: 12 bytes with a sign.
totalsumG	The sums under each tax group G: 12 bytes with a sign.
totalsumH	The sums under each tax group H: 12 bytes with a sign.

**1.1.2.347 report\_ExtDailyClosure\_01()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_ExtDailyClosure_01 (
    string option )
```

Extended daily financial report

**Parameters**

<i>option</i>	'0' - Z report or '2' - X report
---------------	----------------------------------

**Returns**

Dictionary with keys:

---

closure	The fiscal record number: 4 bytes.
fMTotals	The sum of all non-VAT sales: 12 bytes with a sign.
totalsumA	The sums under each tax group A: 12 bytes with a sign.
totalsumB	The sums under each tax group B: 12 bytes with a sign.
totalsumC	The sums under each tax group C: 12 bytes with a sign.
totalsumD	The sums under each tax group D: 12 bytes with a sign.
totalsumE	The sums under each tax group E: 12 bytes with a sign.
totalsumF	The sums under each tax group F: 12 bytes with a sign.
totalsumG	The sums under each tax group G: 12 bytes with a sign.
totalsumH	The sums under each tax group H: 12 bytes with a sign.

### 1.1.2.348 report\_ExtDailyClosure\_02()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_ExtDailyClosure_02 (
    string option,
    string withoutClearOpInfo )
```

Extended daily financial report

#### Parameters

<i>option</i>	'0' - Z report
<i>withoutClearOpInfo</i>	'N' - disables the clearing of accumulated data by operator

#### Returns

Dictionary with keys:

closure	The fiscal record number: 4 bytes.
fMTotals	The sum of all non-VAT sales: 12 bytes with a sign.
totalsumA	The sums under each tax group A: 12 bytes with a sign.
totalsumB	The sums under each tax group B: 12 bytes with a sign.
totalsumC	The sums under each tax group C: 12 bytes with a sign.
totalsumD	The sums under each tax group D: 12 bytes with a sign.
totalsumE	The sums under each tax group E: 12 bytes with a sign.
totalsumF	The sums under each tax group F: 12 bytes with a sign.
totalsumG	The sums under each tax group G: 12 bytes with a sign.
totalsumH	The sums under each tax group H: 12 bytes with a sign.

### 1.1.2.349 report\_FMByDateRange()

```
void FiscalPrinterSDK.FDGROUP_A_BGR.report_FMByDateRange (
    string fromDate,
    string toDate )
```



Full fiscal memory report by fiscal record date

**Parameters**

<i>fromDate</i>	The start fiscal record date. 6 bytes (DDMMYY).
<i>toDate</i>	The end fiscal record date. 6 bytes (DDMMYY).

**1.1.2.350 report\_FMByDateRange\_SHA1()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.report_FMByDateRange_SHA1 (
    string fromDate,
    string toDate )
```

Full fiscal memory report by fiscal record date - each Z-report, prints also the checksum using the SHA-1 algorithm.

**Parameters**

<i>fromDate</i>	The start fiscal record date. 6 bytes (DDMMYY).
<i>toDate</i>	The end fiscal record date. 6 bytes (DDMMYY).

**1.1.2.351 report\_FMByDateRange\_Short()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.report_FMByDateRange_Short (
    string fromDate,
    string toDate )
```

Summary tax memory report from-to date

**Parameters**

<i>fromDate</i>	Start date: 6 bytes (DDMMYY)
<i>toDate</i>	End date: 6 bytes (DDMMYY)

**1.1.2.352 report\_FMByNumRange()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.report_FMByNumRange (
    string startNumber,
    string endNumber )
```

Full fiscal memory report by block number

**Parameters**

<i>startNumber</i>	Start fiscal block number. 4 bytes
<i>endNumber</i>	End fiscal block number. 4 bytes

**1.1.2.353 report\_FMByNumRange\_SHA1()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.report_FMByNumRange_SHA1 (
    string startNumber,
    string endNumber )
```

Full fiscal memory report by block number - prints also the checksum using the SHA-1 algorithm.

**Parameters**

<i>startNumber</i>	Start fiscal block number. 4 bytes
<i>endNumber</i>	End fiscal block number. 4 bytes

**1.1.2.354 report\_FMByNumRange\_Short()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.report_FMByNumRange_Short (
    string startNumber,
    string endNumber )
```

Summary fiscal memory report for a period

**Parameters**

<i>startNumber</i>	Start fiscal record number.
<i>endNumber</i>	End fiscal record number.

**1.1.2.355 report\_Items()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_Items (
    string option )
```

Report by PLU

**Parameters**

<i>option</i>	<p>'S' Prints only the PLUs having sales for the day. For each PLU, prints the code, tax group, product group, name, unit price, sold quantity and sales.</p> <p>'P' All PLUs with their codes, tax groups, product groups, names, sold quantities, available quantities and unit prices are printed.</p>
---------------	---

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.356 report\_Items\_InRange()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_Items_InRange (
    string option,
    string startPLU,
    string endPLU )
```

Report items in range

**Parameters**

<i>option</i>	<p>'S' Prints only the PLUs having sales for the day. For each PLU, prints the code, tax group, product group, name, unit price, sold quantity and sales.</p> <p>'P' All PLUs with their codes, tax groups, product groups, names, sold quantities, available quantities and unit prices are printed.</p>
<i>startPLU</i>	First PLU code to be included in the report.
<i>endPLU</i>	Last PLU code to be included in the report.

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.357 report\_Items\_InRangeByGroup()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_Items_InRangeByGroup (
    string option,
    string startPLU,
    string endPLU,
    string group )
```

Items in range by by group

**Parameters**

<i>option</i>	'S' Prints only the PLUs having sales for the day. For each PLU, prints the code, tax group, product group, name, unit price, sold quantity and sales. 'P' All PLUs with their codes, tax groups, product groups, names, sold quantities, available quantities and unit prices are printed.
<i>startPLU</i>	First PLU code to be included in the report.
<i>endPLU</i>	Last PLU code to be included in the report.
<i>group</i>	Number between 1 and 99.

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.358 report\_Operators()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.report_Operators ( )
```

Operator's report

**1.1.2.359 report\_Pinpad\_EndOfDay()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_Pinpad_EndOfDay ( )
```

End of day report. The command first prints the (unsuccessful) freeze reversal receipts, then sends data to the server, and finishes with printing a short report in response to the successful or unsuccessful clearing of the registers. The command is forbidden in an open fiscal receipt

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.360 report\_Pinpad\_Reports()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_Pinpad_Reports (
    string type )
```

Reports. The command is forbidden in an open fiscal receipt

**Parameters**

<i>type</i>	1 – short report, 2 – extended report
-------------	---------------------------------------

**Returns**

Dictionary with key:

fAnswer	'P' - command passed, 'F' - command failed.
---------	---

**1.1.2.361 report\_Service\_Contracts()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.report_Service_Contracts (
    string option )
```

Report of the registered service agreements

**Parameters**

<i>option</i>	'P'
---------------	-----

**Returns**

Dictionary with key:

fAnswer	'F' - failed or 'Left' - integer parameter: number of remaining service agreement registration fields. 'RegDtTm parameter' - Service agreement date and time: "DD-MM-YYYY hh:mm:ss". EndDate parameter - Last service agreement expiration date in the "DD-MM-YYYY" format. 'UIC parameter' 9 or 13 digits: service provider's UIC.
---------	---

**1.1.2.362 service\_Disable\_Print()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.service_Disable_Print (
    string disabled )
```

Service disabled printing

**Parameters**

<i>disabled</i>	One byte with the following allowed values: '0' Printing is enabled; '1' Printing is disabled.
-----------------	--

**1.1.2.363 service\_Format\_KLEN()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.service_Format_KLEN (
    string option )
```

Service EJ maintenance

**Parameters**

<i>option</i>	One byte with a value of 'F'.
---------------	-------------------------------

**Returns**

Dictionary with key:

<i>errorCode</i>	'P' - pass; 'F' - failed.
------------------	---------------------------

**1.1.2.364 service\_RAM\_Reset()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.service_RAM_Reset ( )
```

Service RAM reset

**1.1.2.365 service\_Set\_DecimalAndTaxRates()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.service_Set_DecimalAndTaxRates (
    string mutiplier,
    string decimals,
    string currencyName,
    string enabledMask,
    string taxA,
    string taxB,
    string taxC,
    string taxD,
    string taxE,
    string taxF,
    string taxG,
    string taxH )
```

Set decimals and tax rates

**Parameters**

<i>mutiplier</i>	From 0 to 3. Currently deactivated and not used.
<i>decimals</i>	1 byte with value between 0 and 2, shows number of digits after decimal point.
<i>currencyName</i>	Currency name (up to 3 bytes)
<i>enabledMask</i>	8 bytes with value 0 or 1, shows if a tax group A,B ... or H is enabled or not

**Parameters**

<i>taxA</i>	Tax rate A value
<i>taxB</i>	Tax rate B value
<i>taxC</i>	Tax rate C value
<i>taxD</i>	Tax rate D value
<i>taxE</i>	Tax rate E value
<i>taxF</i>	Tax rate F value
<i>taxG</i>	Tax rate G value
<i>taxH</i>	Tax rate H value

**1.1.2.366 service\_Set\_EIK()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.service_Set_EIK (
    string eikValue,
    string eikName )
```

Set UIC

**Parameters**

<i>eikValue</i>	These are up to 14 bytes containing the UIC as text
<i>eikName</i>	This is the comment text before the UIC. By default, it is "UIC".

**Returns**

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

**1.1.2.367 service\_Set\_ProductionTestArea()**

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.service_Set_ProductionTestArea (
    string option )
```

Set production test area

**Parameters**

<i>option</i>	'T' - makes a record in the fiscal memory
---------------	---

**Returns**

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
freeRecords	The number of remaining blocks available to record such blocks. 4 bytes

### 1.1.2.368 service\_Set\_SerialNumber()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.service_Set_SerialNumber (
    string serialNumber,
    string fMNumber )
```

Set serial number and fiscal memory number

#### Parameters

<i>serialNumber</i>	These are 8 bytes: Unique Printer ID containing 2 Latin letters and at least 6 digits
<i>fMNumber</i>	These are 8 bytes: the Fiscal Memory Unit ID. Contains only digits.

#### Returns

Dictionary with key:

errorCode	'P' - pass; 'F' - failed.
-----------	---------------------------

### 1.1.2.369 service\_Test\_GPRS()

```
Dictionary<string, string> FiscalPrinterSDK.FDGROUP_A_BGR.service_Test_GPRS (
    string option )
```

GPRS modem test

#### Parameters

<i>option</i>	One byte with a value of 'M'
---------------	------------------------------

#### Returns

Dictionary with keys:

errorCode	'P' - pass; 'F' - failed.
signal	Signal
iMEI	IMEI
iMSI	IMSI
oper	Operator



**1.1.2.370 Set\_Sbit\_ErrorChecking()**

```
void FiscalPrinterSDK.FDGROUP_A_BGR.Set_Sbit_ErrorChecking (
    int byteIndex,
    int bitIndex,
    bool IsError )
```

Get if desired bit is an error

**Parameters**

<i>byteIndex</i>	Desired byte index
<i>bitIndex</i>	Desired bit index
<i>IsError</i>	If the bit is considered error or not

**Returns**

Returns bool (1 - if the bit is considered error, 0 - is it is informatice)



# Index

config\_disable\_EUR  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 18

config\_disable\_PrintSuppression  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 19

config\_enable\_EUR  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 19

config\_enable\_PrintSuppression  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 19

config\_Pinpad\_DeleteBatchNumber  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 20

config\_Pinpad\_DeleteFreezeTransaction  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 20

config\_Pinpad\_HeaderPrintingOptions  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 20, 21

config\_Pinpad\_Set\_DateTime  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 21

config\_Pinpad\_SetBatchNumber  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 22

config\_Pinpad\_SetReceiptNumber  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 22

config\_Pinpad\_SetRRNFromPPTToFD  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 22

config\_Restore\_PrintSuppression  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 23

config\_SafetyTurnOffDevice  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 23

config\_Set\_2GForcedModem  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 23

config\_Set\_AdditionalPaymentName  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 24

config\_Set\_AutoCutting  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 24

config\_Set\_AutoFormat  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 25

config\_Set\_AutomaticRetractPaper  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 25

config\_Set\_AutomaticZReport  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 25

config\_Set\_BarcodeHeight  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 26

config\_Set\_CashDrawerPulse  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 26

config\_Set\_DateTime  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 26

config\_Set\_DepartmentName  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 27

config\_Set\_DepartmentNameTwoRows  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 27

config\_Set\_ExtraSpace  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 27

config\_Set\_FeedButton  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 28

config\_Set\_FontSize  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 28

config\_Set\_FooterLine  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 28

config\_Set\_HeaderLine  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 29

config\_Set\_InvoiceRange  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 29

config\_Set\_LANSignal  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 29

config\_Set\_Logo  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 30

config\_Set\_OperatorName  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 30

config\_Set\_OperatorPassword  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 30

config\_Set\_OptionalErrorText  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 31

config\_Set\_PrintDensity  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 31

config\_Set\_PrintDepartment  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 31

config\_Set\_PrinterReadinessPrintMsgs  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 33

config\_Set\_PrintLogo  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 33

config\_Set\_PrintTaxDDS  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 33

config\_Set\_RollUpPaper  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 34

config\_Set\_SIM  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 34

config\_Set\_Switches  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 34

config\_Set\_UNPPrinting  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 35

config\_Set\_WarningTimeRange  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 35

config\_Set\_WorkingWithPinpad  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 35

config\_SetLAN  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 36

display\_Clear  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 36

display\_Show\_DateTime  
    FiscalPrinterSDK.FDGROUP\_A\_BGR, 36

- display\_Show\_LowerLine
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, 37
- display\_Show\_Text
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, 37
- FiscalPrinterSDK.FDGROUP\_A\_BGR, 1
  - config\_disable\_EUR, 18
  - config\_disable\_PrintSuppression, 19
  - config\_enable\_EUR, 19
  - config\_enable\_PrintSuppression, 19
  - config\_Pinpad\_DeleteBatchNumber, 20
  - config\_Pinpad\_DeleteFreezeTransaction, 20
  - config\_Pinpad\_HeaderPrintingOptions, 20, 21
  - config\_Pinpad\_Set\_DateTime, 21
  - config\_Pinpad\_SetBatchNumber, 22
  - config\_Pinpad\_SetReceiptNumber, 22
  - config\_Pinpad\_SetRRNFromPPTtoFD, 22
  - config\_Restore\_PrintSuppression, 23
  - config\_SafetyTurnOffDevice, 23
  - config\_Set\_2GForcedModem, 23
  - config\_Set\_AdditionalPaymentName, 24
  - config\_Set\_AutoCutting, 24
  - config\_Set\_AutoFormat, 25
  - config\_Set\_AutomaticRetractPaper, 25
  - config\_Set\_AutomaticZReport, 25
  - config\_Set\_BarcodeHeight, 26
  - config\_Set\_CashDrawerPulse, 26
  - config\_Set\_DateTime, 26
  - config\_Set\_DepartmentName, 27
  - config\_Set\_DepartmentNameTwoRows, 27
  - config\_Set\_ExtraSpace, 27
  - config\_Set\_FeedButton, 28
  - config\_Set\_FontSize, 28
  - config\_Set\_FooterLine, 28
  - config\_Set\_HeaderLine, 29
  - config\_Set\_InvoiceRange, 29
  - config\_Set\_LANSignal, 29
  - config\_Set\_Logo, 30
  - config\_Set\_OperatorName, 30
  - config\_Set\_OperatorPassword, 30
  - config\_Set\_OptionalErrorText, 31
  - config\_Set\_PrintDensity, 31
  - config\_Set\_PrintDepartment, 31
  - config\_Set\_PrinterReadinessPrintMsgs, 33
  - config\_Set\_PrintLogo, 33
  - config\_Set\_PrintTaxDDS, 33
  - config\_Set\_RollUpPaper, 34
  - config\_Set\_SIM, 34
  - config\_Set\_Switches, 34
  - config\_Set\_UNPPrinting, 35
  - config\_Set\_WarningTimeRange, 35
  - config\_Set\_WorkingWithPinpad, 35
  - config\_SetLAN, 36
  - display\_Clear, 36
  - display\_Show\_DateTime, 36
  - display\_Show\_LowerLine, 37
  - display\_Show\_Text, 37
  - Get\_SBit\_State, 37
  - Get\_sBytesBitInUse, 38
  - info\_ClearLastErrorExtendedInfo, 38
  - info\_Get\_ActiveTaxes, 38
  - info\_Get\_AdditionalDailyInfo\_01, 39
  - info\_Get\_AdditionalDailyInfo\_02, 40
  - info\_Get\_AdditionalDailyInfo\_03, 40
  - info\_Get\_AdditionalPaymentNames, 41
  - info\_Get\_AdditionalPayments, 41
  - info\_Get\_CashDrawerStatusOrOpenShift, 42
  - info\_Get\_CashIn\_CashOut, 42
  - info\_Get\_CashInCashOut, 43
  - info\_Get\_Corrections, 43
  - info\_Get\_CurrentReceiptInfo, 44
  - info\_Get\_DailyCorrections, 44
  - info\_Get\_DataForLastIssuedReceipt, 45
  - info\_Get\_DateTime, 45
  - info\_Get\_DateTime\_01, 46
  - info\_Get\_DecimalAndTaxRates, 46
  - info\_Get\_DepartmentInfo, 46
  - info\_Get\_DiagnosticInfo, 47
  - info\_Get\_EIK, 47
  - info\_Get\_FMRecord\_LastDateTime, 48
  - info\_Get\_FMRecord\_LastDateTime\_01, 48
  - info\_Get\_FreeFMRecords, 48
  - info\_Get\_FTransactionStatus, 49
  - info\_Get\_InvoiceRange, 49
  - info\_Get\_LANSettings, 50
  - info\_Get\_LastDocumentNumber, 50
  - info\_Get\_LastFiscRecord, 50
  - info\_Get\_LastFiscRecord\_01, 51
  - info\_Get\_LogoLine, 51
  - info\_Get\_NettoSums, 52
  - info\_Get\_NotPrintedRowsCount, 52
  - info\_Get\_NRADData\_Type1, 53
  - info\_Get\_NRADData\_Type2, 53
  - info\_Get\_NRADData\_Type3, 54
  - info\_Get\_NRADData\_Type4, 54
  - info\_Get\_NRADData\_Type5, 54
  - info\_Get\_NRADData\_Type6, 55
  - info\_Get\_NRADData\_Type7, 55
  - info\_Get\_NRADData\_Type8, 56
  - info\_Get\_NRADDocuments, 56
  - info\_Get\_Numbers, 57
  - info\_Get\_OperatorsData, 57
  - info\_Get\_PrintOption, 58
  - info\_Get\_ProgrammedFuel, 58
  - info\_Get\_ProgrammedPumps, 59
  - info\_Get\_ProgrammedPumps2, 59
  - info\_Get\_ProgrammedPumps3, 60
  - info\_Get\_RAMResetDateTime, 60
  - info\_Get\_SoldQuant\_AccumulatedSales, 61
  - info\_Get\_StatusBytes, 61
  - info\_Get\_Storno\_NettoSums, 61
  - info\_Get\_Storno\_VATsSums, 62
  - info\_Get\_StornoSums, 63
  - info\_Get\_Sums, 64
  - info\_Get\_TaxRates, 64
  - info\_Get\_TaxRatesByPeriod, 65
  - info\_Get\_TaxRateValues, 65

- info\_Get\_Turnover, 67
- info\_Get\_VATSUMs, 67
- info\_Get\_VoltageAndTemperature, 68
- info\_LastErrorExtendedInfo, 68
- info\_Pinpad\_CheckLastRecordedRRN, 69
- info\_Pinpad\_ClearTransactionStatusFlags, 69
- info\_Pinpad\_DataForLastTransaction, 69
- info\_Pinpad\_Get\_DateTime, 70
- info\_Pinpad\_GetUpdate, 70
- info\_Pinpad\_Information, 70
- info\_Pinpad\_NumberOfEntries, 71
- info\_Pinpad\_PrintLastSuccessfulTransaction, 71
- info\_Pinpad\_StatusOfLastTransaction, 71
- info\_Pinpad\_TestConnection, 72
- info\_Pinpad\_TestConnectionToServer, 72
- info\_Pinpad\_TransactionStatus, 72
- info\_Print\_Diagnostic\_0, 73
- info\_Print\_Short\_FMReportByDTRange, 73
- info\_Print\_Short\_MonthlyFMReport, 73
- info\_Print\_Short\_YearlyFMReport, 74
- info\_Service\_Contracts, 74
- items\_Delete\_All\_Items, 74
- items\_Delete\_Item, 75
- items\_Delete\_ItemsInRange, 75
- items\_Get\_FirstFoundItem, 76
- items\_Get\_FirstNotProgrammedItem, 76
- items\_Get\_FirstSoldItem, 77
- items\_Get\_Item, 77
- items\_Get\_ItemsInformation, 78
- items\_Get\_LastFoundItem, 78
- items\_Get\_LastNotProgrammedItem, 79
- items\_Get\_LastSoldItem, 79
- items\_Get\_NextItem, 80
- items\_Get\_NextSoldItem, 80
- items\_Set\_Item, 81
- items\_Set\_ItemQuantity, 82
- items\_Set\_ItemWithReplace, 82
- klen\_Find\_Document, 83
- klen\_Get\_Info, 83
- klen\_Get\_KLENInfo\_ZReport, 84
- klen\_Get\_NextTextRow, 84
- klen\_Get\_ZReportInfo, 85
- klen\_Print\_DocsFromDateToDate, 85
- klen\_Print\_DocsInRange, 86
- klen\_Print\_DocsInRange\_InZReport, 86
- klen\_Print\_Document, 87
- klen\_Print\_SHA1, 88
- klen\_Print\_Validation, 88
- klen\_Set\_DocsRange, 88
- klen\_Set\_DocsRange\_ByDateTime, 89
- klen\_Set\_DocsRange\_InZReport, 89
- klen\_Set\_DocsRange\_InZReport\_01, 90
- klen\_Set\_DocsRange\_InZReport\_02, 91
- klen\_SInfo\_ExV\_ByDateRange, 91
- klen\_SInfo\_ExV\_ByNumber, 92
- klen\_SInfo\_ExV\_ByNumbersRange, 92
- klen\_SInfo\_ExV\_GetNext, 93
- klen\_SInfo\_V\_ByDateRange, 94
- klen\_SInfo\_V\_ByNumbersRange, 95
- klen\_SInfo\_V\_GetNext, 96
- klen\_SInfo\_W\_ByDateRange, 96
- klen\_SInfo\_W\_ByNumber, 97
- klen\_SInfo\_W\_ByNumbersRange, 97
- klen\_SInfo\_W\_GetNext, 97
- klen\_SInfo\_Y\_ByDateRange, 99
- klen\_SInfo\_Y\_ByNumber, 99
- klen\_SInfo\_Y\_ByNumbersRange, 100
- klen\_SInfo\_Y\_GetNext, 100
- open\_StornoReceipt, 100
- other\_Sound\_Signal, 101
- receipt\_CashIn\_CashOut, 101
- receipt\_Current\_Info, 102
- receipt\_DisplayDepPLU\_Sale, 102
- receipt\_DisplayDepPLU\_Sale\_CByPercent, 103
- receipt\_DisplayDepPLU\_Sale\_CBySum, 103
- receipt\_DisplayDepPLU\_SaleUn, 104
- receipt\_DisplayDepPLU\_SaleUn\_CByPercent, 104
- receipt\_DisplayDepPLU\_SaleUn\_CBySum, 105
- receipt\_DisplayDSale, 105
- receipt\_DisplayDSale\_CByPercent, 106
- receipt\_DisplayDSale\_CBySum, 106
- receipt\_DisplayDSale\_Minimum, 107
- receipt\_DisplayDSale\_Un, 107
- receipt\_DisplayDSale\_UnCByPercent, 108
- receipt\_DisplayDSale\_UnCBySum, 108
- receipt\_DisplayPLUSale, 109
- receipt\_DisplayPLUSale\_CByPercent, 109
- receipt\_DisplayPLUSale\_CBySum, 110
- receipt\_DisplayPLUSale\_Un, 110
- receipt\_DisplayPLUSale\_UnCByPercent, 111
- receipt\_DisplayPLUSale\_UnCBySum, 111
- receipt\_DisplaySale, 112
- receipt\_DisplaySale\_CByPercent, 112
- receipt\_DisplaySale\_CBySum, 112
- receipt\_DisplaySale\_Minimum, 113
- receipt\_DisplaySale\_Un, 113
- receipt\_DisplaySale\_UnCByPercent, 114
- receipt\_DisplaySale\_UnCBySum, 114
- receipt\_Drawer\_KickOut, 115
- receipt\_DSale, 115
- receipt\_DSale\_CByPercent, 115
- receipt\_DSale\_CByPercentWText, 116
- receipt\_DSale\_CBySum, 116
- receipt\_DSale\_CBySumWText, 117
- receipt\_DSale\_Minimum, 117
- receipt\_DSale\_TextRow1, 118
- receipt\_DSale\_TextRow1\_WQuan, 118
- receipt\_DSale\_TextRow1CByPercent, 118
- receipt\_DSale\_TextRow1CBySum, 119
- receipt\_DSale\_TextRow1Un, 119
- receipt\_DSale\_TextRow1UnCByPercent, 120
- receipt\_DSale\_TextRow1UnCBySum, 120
- receipt\_DSale\_TextRow2, 121
- receipt\_DSale\_TextRow2CByPercent, 121
- receipt\_DSale\_TextRow2CBySum, 122

- receipt\_DSale\_TextRow2Un, 122
- receipt\_DSale\_TextRow2UnCByPercent, 123
- receipt\_DSale\_TextRow2UnCBySum, 123
- receipt\_DSale\_Un, 124
- receipt\_DSale\_UnCByPercent, 124
- receipt\_DSale\_UnCByPercentWText, 125
- receipt\_DSale\_UnCBySum, 125
- receipt\_DSale\_UnCBySumWText, 126
- receipt\_DSale\_UnWText, 126
- receipt\_Fiscal\_01\_Open, 127
- receipt\_Fiscal\_02\_Open, 127
- receipt\_Fiscal\_Cancel, 128
- receipt\_Fiscal\_Close, 128
- receipt\_Fiscal\_Text, 128
- receipt\_FiscalOpen\_A02, 129
- receipt\_FiscalOpen\_A04, 129
- receipt\_Invoice\_01\_Open, 130
- receipt\_Invoice\_Open, 130
- receipt\_NonFiscal\_Close, 131
- receipt\_NonFiscal\_Open, 131
- receipt\_NonFiscal\_Text, 132
- receipt\_NonFiscalRotated\_Close, 132
- receipt\_NonFiscalRotated\_Open, 132
- receipt\_NonFiscalRotated\_Text, 132
- receipt\_Paper\_Cut, 133
- receipt\_Paper\_Feed, 133
- receipt\_Paper\_Feed\_InnerContainer, 133
- receipt\_PFiscalText\_01, 133
- receipt\_PFiscalText\_02, 134
- receipt\_Pinpad\_PrintByNumber, 134
- receipt\_Pinpad\_PrintLastSuccessfultransaction, 134
- receipt\_Pinpad\_PrintReceiptCopyFromEJ\_ByDate, 135
- receipt\_Pinpad\_PrintReceiptCopyFromEJ\_ByNum, 135
- receipt\_Pinpad\_ReturnLastSuccessfultransaction, 136
- receipt\_PLU\_Sale, 136
- receipt\_PLUDep\_Sale, 136
- receipt\_PLUDep\_Sale\_CByPercent, 137
- receipt\_PLUDep\_Sale\_CBySum, 137
- receipt\_PLUDep\_Sale\_Un, 138
- receipt\_PLUDep\_Sale\_UnCByPercent, 138
- receipt\_PLUDep\_Sale\_UnCBySum, 139
- receipt\_PLUSale\_CByPercent, 139
- receipt\_PLUSale\_CBySum, 140
- receipt\_PLUSale\_Un, 140
- receipt\_PLUSale\_UnCByPercent, 140
- receipt\_PLUSale\_UnCBySum, 141
- receipt\_PNonFiscal\_Text, 141
- receipt\_Print\_Barcode\_01, 142
- receipt\_Print\_Barcode\_02, 142
- receipt\_Print\_Duplicate, 143
- receipt\_PrintClientInfo\_01, 143
- receipt\_PrintClientInfo\_02, 144
- receipt\_PrintClientInfo\_03, 144
- receipt\_PrintClientInfo\_04, 144
- receipt\_PrintClientInfo\_05, 145
- receipt\_PrintClientInfo\_06, 145
- receipt\_PrintClientInfo\_07, 146
- receipt\_Sale, 146
- receipt\_Sale\_CByPercent, 147
- receipt\_Sale\_CByPercentWText, 147
- receipt\_Sale\_CBySum, 148
- receipt\_Sale\_CBySumWText, 148
- receipt\_Sale\_Minimum, 149
- receipt\_Sale\_TextRow1, 149
- receipt\_Sale\_TextRow1\_WQuan, 149
- receipt\_Sale\_TextRow1CByPercent, 150
- receipt\_Sale\_TextRow1CBySum, 150
- receipt\_Sale\_TextRow1Un, 151
- receipt\_Sale\_TextRow1UnCByPercent, 151
- receipt\_Sale\_TextRow1UnCBySum, 152
- receipt\_Sale\_TextRow2, 152
- receipt\_Sale\_TextRow2CByPercent, 153
- receipt\_Sale\_TextRow2CBySum, 153
- receipt\_Sale\_TextRow2Un, 154
- receipt\_Sale\_TextRow2UnCByPercent, 154
- receipt\_Sale\_TextRow2UnCBySum, 154
- receipt\_Sale\_Un, 155
- receipt\_Sale\_UnCByPercent, 155
- receipt\_Sale\_UnCByPercentWText, 156
- receipt\_Sale\_UnCBySum, 157
- receipt\_Sale\_UnCBySumWText, 157
- receipt\_Sale\_UnWText, 158
- receipt\_Separating\_Line, 158
- receipt\_Separating\_LineW, 158
- receipt\_StornoOpen\_A01, 159
- receipt\_StornoOpen\_A02, 159
- receipt\_StornoOpen\_A03, 160
- receipt\_StornoOpen\_A04, 160
- receipt\_StornoOpen\_A05, 162
- receipt\_StornoOpen\_A06, 163
- receipt\_StornoOpen\_A07, 163
- receipt\_StornoOpen\_A08, 164
- receipt\_StornoOpen\_A09, 165
- receipt\_StornoOpen\_A10, 166
- receipt\_StornoOpen\_A11, 166
- receipt\_StornoOpen\_A12, 167
- receipt\_StornoOpen\_A13, 168
- receipt\_StornoOpen\_A14, 169
- receipt\_StornoOpen\_A15, 169
- receipt\_StornoOpen\_A16, 170
- receipt\_Subtotal, 171
- receipt\_Subtotal\_CByPercent, 172
- receipt\_Subtotal\_CBySum, 172
- receipt\_Total, 173
- receipt\_Total\_PAmount, 173
- receipt\_Total\_PAmountTextRow1, 174
- receipt\_Total\_PAmountTextRow2, 175
- receipt\_Total\_PAmountWithoutText, 175
- receipt\_Total\_TextRow1, 176
- receipt\_Total\_TextRow2, 177
- receipt\_Total\_WithoutText, 177
- report\_DailyClosure\_01, 177

- report\_DailyClosure\_02, [178](#)
- report\_DailyClosureByDepartments\_01, [179](#)
- report\_DailyClosureByDepartments\_02, [179](#)
- report\_DailyClosureByDepartmentsAndItems\_01, [180](#)
- report\_DailyClosureByDepartmentsAndItems\_02, [181](#)
- report\_ExtDailyClosure\_01, [181](#)
- report\_ExtDailyClosure\_02, [182](#)
- report\_FMByDateRange, [182](#)
- report\_FMByDateRange\_SHA1, [183](#)
- report\_FMByDateRange\_Short, [183](#)
- report\_FMByNumRange, [183](#)
- report\_FMByNumRange\_SHA1, [184](#)
- report\_FMByNumRange\_Short, [184](#)
- report\_Items, [184](#)
- report\_Items\_InRange, [185](#)
- report\_Items\_InRangeByGroup, [185](#)
- report\_Operators, [186](#)
- report\_Pinpad\_EndOfDay, [186](#)
- report\_Pinpad\_Reports, [186](#)
- report\_Service\_Contracts, [187](#)
- service\_Disable\_Print, [187](#)
- service\_Format\_KLEN, [187](#)
- service\_RAM\_Reset, [188](#)
- service\_Set\_DecimalAndTaxRates, [188](#)
- service\_Set\_EIK, [189](#)
- service\_Set\_ProductionTestArea, [189](#)
- service\_Set\_SerialNumber, [190](#)
- service\_Test\_GPRS, [190](#)
- Set\_Sbit\_ErrorChecking, [191](#)
- Get\_SBit\_State
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [37](#)
- Get\_sBytesBitInUse
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [38](#)
- info\_ClearLastErrorExtendedInfo
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [38](#)
- info\_Get\_ActiveTaxes
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [38](#)
- info\_Get\_AdditionalDailyInfo\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [39](#)
- info\_Get\_AdditionalDailyInfo\_02
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [40](#)
- info\_Get\_AdditionalDailyInfo\_03
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [40](#)
- info\_Get\_AdditionalPaymentNames
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [41](#)
- info\_Get\_AdditionalPayments
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [41](#)
- info\_Get\_CashDrawerStatusOrOpenShift
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [42](#)
- info\_Get\_CashIn\_CashOut
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [42](#)
- info\_Get\_CashInCashOut
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [43](#)
- info\_Get\_Corrections
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [43](#)
- info\_Get\_CurrentReceiptInfo
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [44](#)
- info\_Get\_DailyCorrections
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [44](#)
- info\_Get\_DataForLastIssuedReceipt
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [45](#)
- info\_Get\_DateTime
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [45](#)
- info\_Get\_DateTime\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [46](#)
- info\_Get\_DecimalAndTaxRates
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [46](#)
- info\_Get\_DepartmentInfo
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [46](#)
- info\_Get\_DiagnosticInfo
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [47](#)
- info\_Get\_EIK
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [47](#)
- info\_Get\_FMRecord\_LastDateTime
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [48](#)
- info\_Get\_FMRecord\_LastDateTime\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [48](#)
- info\_Get\_FreeFMRecords
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [48](#)
- info\_Get\_FTransactionStatus
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [49](#)
- info\_Get\_InvoiceRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [49](#)
- info\_Get\_LANSettings
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [50](#)
- info\_Get\_LastDocumentNumber
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [50](#)
- info\_Get\_LastFiscRecord
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [50](#)
- info\_Get\_LastFiscRecord\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [51](#)
- info\_Get\_LogoLine
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [51](#)
- info\_Get\_NettoSums
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [52](#)
- info\_Get\_NotPrintedRowsCount
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [52](#)
- info\_Get\_NRADData\_Type1
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [53](#)
- info\_Get\_NRADData\_Type2
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [53](#)
- info\_Get\_NRADData\_Type3
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [54](#)
- info\_Get\_NRADData\_Type4
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [54](#)
- info\_Get\_NRADData\_Type5
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [54](#)
- info\_Get\_NRADData\_Type6
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [55](#)
- info\_Get\_NRADData\_Type7
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [55](#)
- info\_Get\_NRADData\_Type8
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [56](#)



- info\_Get\_NRADocuments
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [56](#)
- info\_Get\_Numbers
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [57](#)
- info\_Get\_OperatorsData
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [57](#)
- info\_Get\_PrintOption
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [58](#)
- info\_Get\_ProgrammedFuel
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [58](#)
- info\_Get\_ProgrammedPumps
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [59](#)
- info\_Get\_ProgrammedPumps2
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [59](#)
- info\_Get\_ProgrammedPumps3
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [60](#)
- info\_Get\_RAMResetDateTime
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [60](#)
- info\_Get\_SoldQuant\_AccumulatedSales
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [61](#)
- info\_Get\_StatusBytes
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [61](#)
- info\_Get\_Storno\_NettoSums
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [61](#)
- info\_Get\_Storno\_VATSums
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [62](#)
- info\_Get\_StornoSums
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [63](#)
- info\_Get\_Sums
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [64](#)
- info\_Get\_TaxRates
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [64](#)
- info\_Get\_TaxRatesByPeriod
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [65](#)
- info\_Get\_TaxRateValues
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [65](#)
- info\_Get\_Turnover
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [67](#)
- info\_Get\_VATSums
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [67](#)
- info\_Get\_VoltageAndTemperature
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [68](#)
- info\_LastErrorExtendedInfo
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [68](#)
- info\_Pinpad\_CheckLastRecordedRRN
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [69](#)
- info\_Pinpad\_ClearTransactionStatusFlags
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [69](#)
- info\_Pinpad\_DataForLastTransaction
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [69](#)
- info\_Pinpad\_Get\_DateTime
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [70](#)
- info\_Pinpad\_GetUpdate
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [70](#)
- info\_Pinpad\_Information
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [70](#)
- info\_Pinpad\_NumberOfEntries
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [71](#)
- info\_Pinpad\_PrintLastSuccessfulTransaction
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [71](#)
- info\_Pinpad\_StatusOfLastTransaction
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [71](#)
- info\_Pinpad\_TestConnection
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [72](#)
- info\_Pinpad\_TestConnectionToServer
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [72](#)
- info\_Pinpad\_TransactionStatus
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [72](#)
- info\_Print\_Diagnostic\_0
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [73](#)
- info\_Print\_Short\_FMReportByDTRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [73](#)
- info\_Print\_Short\_MonthlyFMReport
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [73](#)
- info\_Print\_Short\_YearlyFMReport
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [74](#)
- info\_Service\_Contracts
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [74](#)
- items\_Delete\_All\_Items
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [74](#)
- items\_Delete\_Item
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [75](#)
- items\_Delete\_ItemsInRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [75](#)
- items\_Get\_FirstFoundItem
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [76](#)
- items\_Get\_FirstNotProgrammedItem
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [76](#)
- items\_Get\_FirstSoldItem
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [77](#)
- items\_Get\_Item
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [77](#)
- items\_Get\_ItemsInformation
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [78](#)
- items\_Get\_LastFoundItem
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [78](#)
- items\_Get\_LastNotProgrammedItem
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [79](#)
- items\_Get\_LastSoldItem
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [79](#)
- items\_Get\_NextItem
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [80](#)
- items\_Get\_NextSoldItem
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [80](#)
- items\_Set\_Item
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [81](#)
- items\_Set\_ItemQuantity
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [82](#)
- items\_Set\_ItemWithReplace
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [82](#)
- klen\_Find\_Document
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [83](#)
- klen\_Get\_Info
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [83](#)
- klen\_Get\_KLENInfo\_ZReport
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [84](#)



- klen\_Get\_NextTextRow
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [84](#)
- klen\_Get\_ZReportInfo
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [85](#)
- klen\_Print\_DocsFromDateToDate
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [85](#)
- klen\_Print\_DocsInRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [86](#)
- klen\_Print\_DocsInRange\_InZReport
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [86](#)
- klen\_Print\_Document
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [87](#)
- klen\_Print\_SHA1
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [88](#)
- klen\_Print\_Validation
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [88](#)
- klen\_Set\_DocsRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [88](#)
- klen\_Set\_DocsRange\_ByDateTime
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [89](#)
- klen\_Set\_DocsRange\_InZReport
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [89](#)
- klen\_Set\_DocsRange\_InZReport\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [90](#)
- klen\_Set\_DocsRange\_InZReport\_02
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [91](#)
- klen\_SInfo\_ExV\_ByDateRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [91](#)
- klen\_SInfo\_ExV\_ByNumber
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [92](#)
- klen\_SInfo\_ExV\_ByNumbersRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [92](#)
- klen\_SInfo\_ExV\_GetNext
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [93](#)
- klen\_SInfo\_V\_ByDateRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [94](#)
- klen\_SInfo\_V\_ByNumber
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [94](#)
- klen\_SInfo\_V\_ByNumbersRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [95](#)
- klen\_SInfo\_V\_GetNext
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [96](#)
- klen\_SInfo\_W\_ByDateRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [96](#)
- klen\_SInfo\_W\_ByNumber
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [97](#)
- klen\_SInfo\_W\_ByNumbersRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [97](#)
- klen\_SInfo\_W\_GetNext
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [97](#)
- klen\_SInfo\_Y\_ByDateRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [99](#)
- klen\_SInfo\_Y\_ByNumber
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [99](#)
- klen\_SInfo\_Y\_ByNumbersRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [100](#)
- klen\_SInfo\_Y\_GetNext
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [100](#)
- open\_StornoReceipt
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [100](#)
- other\_Sound\_Signal
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [101](#)
- receipt\_CashIn\_CashOut
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [101](#)
- receipt\_Current\_Info
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [102](#)
- receipt\_DisplayDepPLU\_Sale
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [102](#)
- receipt\_DisplayDepPLU\_Sale\_CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [103](#)
- receipt\_DisplayDepPLU\_Sale\_CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [103](#)
- receipt\_DisplayDepPLU\_SaleUn
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [104](#)
- receipt\_DisplayDepPLU\_SaleUn\_CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [104](#)
- receipt\_DisplayDepPLU\_SaleUn\_CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [105](#)
- receipt\_DisplayDSale
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [105](#)
- receipt\_DisplayDSale\_CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [106](#)
- receipt\_DisplayDSale\_CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [106](#)
- receipt\_DisplayDSale\_Minimum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [107](#)
- receipt\_DisplayDSale\_Un
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [107](#)
- receipt\_DisplayDSale\_UnCByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [108](#)
- receipt\_DisplayDSale\_UnCBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [108](#)
- receipt\_DisplayPLUSale
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [109](#)
- receipt\_DisplayPLUSale\_CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [109](#)
- receipt\_DisplayPLUSale\_CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [110](#)
- receipt\_DisplayPLUSale\_Un
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [110](#)
- receipt\_DisplayPLUSale\_UnCByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [111](#)
- receipt\_DisplayPLUSale\_UnCBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [111](#)
- receipt\_DisplaySale
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [112](#)
- receipt\_DisplaySale\_CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [112](#)
- receipt\_DisplaySale\_CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [112](#)
- receipt\_DisplaySale\_Minimum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [113](#)
- receipt\_DisplaySale\_Un
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [113](#)
- receipt\_DisplaySale\_UnCByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [114](#)

- receipt\_DisplaySale\_UnCBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [114](#)
- receipt\_Drawer\_KickOut
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [115](#)
- receipt\_DSale
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [115](#)
- receipt\_DSale\_CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [115](#)
- receipt\_DSale\_CByPercentWText
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [116](#)
- receipt\_DSale\_CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [116](#)
- receipt\_DSale\_CBySumWText
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [117](#)
- receipt\_DSale\_Minimum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [117](#)
- receipt\_DSale\_TextRow1
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [118](#)
- receipt\_DSale\_TextRow1\_WQuan
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [118](#)
- receipt\_DSale\_TextRow1CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [118](#)
- receipt\_DSale\_TextRow1CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [119](#)
- receipt\_DSale\_TextRow1Un
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [119](#)
- receipt\_DSale\_TextRow1UnCByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [120](#)
- receipt\_DSale\_TextRow1UnCBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [120](#)
- receipt\_DSale\_TextRow2
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [121](#)
- receipt\_DSale\_TextRow2CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [121](#)
- receipt\_DSale\_TextRow2CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [122](#)
- receipt\_DSale\_TextRow2Un
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [122](#)
- receipt\_DSale\_TextRow2UnCByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [123](#)
- receipt\_DSale\_TextRow2UnCBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [123](#)
- receipt\_DSale\_Un
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [124](#)
- receipt\_DSale\_UnCByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [124](#)
- receipt\_DSale\_UnCByPercentWText
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [125](#)
- receipt\_DSale\_UnCBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [125](#)
- receipt\_DSale\_UnCBySumWText
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [126](#)
- receipt\_DSale\_UnWText
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [126](#)
- receipt\_Fiscal\_01\_Open
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [127](#)
- receipt\_Fiscal\_02\_Open
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [127](#)
- receipt\_Fiscal\_Cancel
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [128](#)
- receipt\_Fiscal\_Close
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [128](#)
- receipt\_Fiscal\_Text
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [128](#)
- receipt\_FiscalOpen\_A02
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [129](#)
- receipt\_FiscalOpen\_A04
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [129](#)
- receipt\_Invoice\_01\_Open
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [130](#)
- receipt\_Invoice\_Open
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [130](#)
- receipt\_NonFiscal\_Close
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [131](#)
- receipt\_NonFiscal\_Open
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [131](#)
- receipt\_NonFiscal\_Text
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [132](#)
- receipt\_NonFiscalRotated\_Close
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [132](#)
- receipt\_NonFiscalRotated\_Open
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [132](#)
- receipt\_NonFiscalRotated\_Text
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [132](#)
- receipt\_Paper\_Cut
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [133](#)
- receipt\_Paper\_Feed
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [133](#)
- receipt\_Paper\_Feed\_InnerContainer
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [133](#)
- receipt\_PFiscalText\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [133](#)
- receipt\_PFiscalText\_02
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [134](#)
- receipt\_Pinpad\_PrintByNumber
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [134](#)
- receipt\_Pinpad\_PrintLastSuccessfultransaction
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [134](#)
- receipt\_Pinpad\_PrintReceiptCopyFromEJ\_ByDate
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [135](#)
- receipt\_Pinpad\_PrintReceiptCopyFromEJ\_ByNum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [135](#)
- receipt\_Pinpad\_ReturnLastSuccessfultransaction
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [136](#)
- receipt\_PLU\_Sale
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [136](#)
- receipt\_PLUDep\_Sale
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [136](#)
- receipt\_PLUDep\_Sale\_CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [137](#)
- receipt\_PLUDep\_Sale\_CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [137](#)
- receipt\_PLUDep\_Sale\_Un
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [138](#)
- receipt\_PLUDep\_Sale\_UnCByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [138](#)

receipt\_PLUDep\_Sale\_UnCBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [139](#)  
 receipt\_PLUSale\_CByPercent  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [139](#)  
 receipt\_PLUSale\_CBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [140](#)  
 receipt\_PLUSale\_Un  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [140](#)  
 receipt\_PLUSale\_UnCByPercent  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [140](#)  
 receipt\_PLUSale\_UnCBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [141](#)  
 receipt\_PNonFiscal\_Text  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [141](#)  
 receipt\_Print\_Barcode\_01  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [142](#)  
 receipt\_Print\_Barcode\_02  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [142](#)  
 receipt\_Print\_Duplicate  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [143](#)  
 receipt\_PrintClientInfo\_01  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [143](#)  
 receipt\_PrintClientInfo\_02  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [144](#)  
 receipt\_PrintClientInfo\_03  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [144](#)  
 receipt\_PrintClientInfo\_04  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [144](#)  
 receipt\_PrintClientInfo\_05  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [145](#)  
 receipt\_PrintClientInfo\_06  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [145](#)  
 receipt\_PrintClientInfo\_07  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [146](#)  
 receipt\_Sale  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [146](#)  
 receipt\_Sale\_CByPercent  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [147](#)  
 receipt\_Sale\_CByPercentWText  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [147](#)  
 receipt\_Sale\_CBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [148](#)  
 receipt\_Sale\_CBySumWText  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [148](#)  
 receipt\_Sale\_Minimum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [149](#)  
 receipt\_Sale\_TextRow1  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [149](#)  
 receipt\_Sale\_TextRow1\_WQuan  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [149](#)  
 receipt\_Sale\_TextRow1CByPercent  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [150](#)  
 receipt\_Sale\_TextRow1CBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [150](#)  
 receipt\_Sale\_TextRow1Un  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [151](#)  
 receipt\_Sale\_TextRow1UnCByPercent  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [151](#)  
 receipt\_Sale\_TextRow1UnCBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [152](#)  
 receipt\_Sale\_TextRow2  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [152](#)  
 receipt\_Sale\_TextRow2CByPercent  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [153](#)  
 receipt\_Sale\_TextRow2CBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [153](#)  
 receipt\_Sale\_TextRow2Un  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [154](#)  
 receipt\_Sale\_TextRow2UnCByPercent  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [154](#)  
 receipt\_Sale\_TextRow2UnCBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [154](#)  
 receipt\_Sale\_Un  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [155](#)  
 receipt\_Sale\_UnCByPercent  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [155](#)  
 receipt\_Sale\_UnCByPercentWText  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [156](#)  
 receipt\_Sale\_UnCBySum  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [157](#)  
 receipt\_Sale\_UnCBySumWText  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [157](#)  
 receipt\_Sale\_UnWText  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [158](#)  
 receipt\_Separating\_Line  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [158](#)  
 receipt\_Separating\_LineW  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [158](#)  
 receipt\_StornoOpen\_A01  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [159](#)  
 receipt\_StornoOpen\_A02  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [159](#)  
 receipt\_StornoOpen\_A03  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [160](#)  
 receipt\_StornoOpen\_A04  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [160](#)  
 receipt\_StornoOpen\_A05  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [162](#)  
 receipt\_StornoOpen\_A06  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [163](#)  
 receipt\_StornoOpen\_A07  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [163](#)  
 receipt\_StornoOpen\_A08  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [164](#)  
 receipt\_StornoOpen\_A09  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [165](#)  
 receipt\_StornoOpen\_A10  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [166](#)  
 receipt\_StornoOpen\_A11  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [166](#)  
 receipt\_StornoOpen\_A12  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [167](#)  
 receipt\_StornoOpen\_A13  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [168](#)  
 receipt\_StornoOpen\_A14  
     FiscalPrinterSDK.FDGROUP\_A\_BGR, [169](#)

- receipt\_StornoOpen\_A15
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [169](#)
- receipt\_StornoOpen\_A16
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [170](#)
- receipt\_Subtotal
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [171](#)
- receipt\_Subtotal\_CByPercent
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [172](#)
- receipt\_Subtotal\_CBySum
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [172](#)
- receipt\_Total
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [173](#)
- receipt\_Total\_PAmount
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [173](#)
- receipt\_Total\_PAmountTextRow1
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [174](#)
- receipt\_Total\_PAmountTextRow2
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [175](#)
- receipt\_Total\_PAmountWithoutText
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [175](#)
- receipt\_Total\_TextRow1
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [176](#)
- receipt\_Total\_TextRow2
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [177](#)
- receipt\_Total\_WithoutText
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [177](#)
- report\_DailyClosure\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [177](#)
- report\_DailyClosure\_02
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [178](#)
- report\_DailyClosureByDepartments\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [179](#)
- report\_DailyClosureByDepartments\_02
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [179](#)
- report\_DailyClosureByDepartmentsAndItems\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [180](#)
- report\_DailyClosureByDepartmentsAndItems\_02
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [181](#)
- report\_ExtDailyClosure\_01
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [181](#)
- report\_ExtDailyClosure\_02
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [182](#)
- report\_FMByDateRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [182](#)
- report\_FMByDateRange\_SHA1
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [183](#)
- report\_FMByDateRange\_Short
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [183](#)
- report\_FMByNumRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [183](#)
- report\_FMByNumRange\_SHA1
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [184](#)
- report\_FMByNumRange\_Short
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [184](#)
- report\_Items
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [184](#)
- report\_Items\_InRange
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [185](#)
- report\_Items\_InRangeByGroup
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [185](#)
- report\_Operators
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [186](#)
- report\_Pinpad\_EndOfDay
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [186](#)
- report\_Pinpad\_Reports
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [186](#)
- report\_Service\_Contracts
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [187](#)
- service\_Disable\_Print
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [187](#)
- service\_Format\_KLEN
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [187](#)
- service\_RAM\_Reset
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [188](#)
- service\_Set\_DecimalAndTaxRates
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [188](#)
- service\_Set\_EIK
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [189](#)
- service\_Set\_ProductionTestArea
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [189](#)
- service\_Set\_SerialNumber
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [190](#)
- service\_Test\_GPRS
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [190](#)
- Set\_Sbit\_ErrorChecking
  - FiscalPrinterSDK.FDGROUP\_A\_BGR, [191](#)