

FiscalPrinter SDK

(ver.1.0.0.9)

User Manual

Table of Contents

Product development history.....	2
Introduction.....	8
Device maintenance.....	8
Language support.....	9
Instalation.....	9
Usage.....	9
Eumerated types.....	12
Properties.....	13
Error status bits.....	13
Device group A.....	13
Device group B.....	14
Device group C.....	16
Informative status bits.....	17
Device group A.....	17
Device group B.....	19
Device group C.....	20
Status Bits for all logical device groups.....	21
Status byte 0.....	21
Status byte 1.....	22
Status byte 2.....	22
Status byte 3.....	23
Status byte 4.....	23
Status byte 5.....	24
Status byte 6.....	24
Status byte 7.....	24
Communication.....	25
Methods.....	26
Error codes and messages.....	27
Disclaimer.....	27
Contacts.....	28

Product development history

Version 1.0.0.9 (08/2020)

Date	Description
08/2020	- Added demo about creating and saving receipt as image (demo for electronic stores) in in C# demo program - "DEMOS/GetEJDocuments_DeviceGroup_A" - about device group A. - Added demo about creating and saving receipt as image (demo for electronic stores) in in C# demo program - "DEMOS/GetEJDocuments_DeviceGroup_B" - about device group B. - Added demo about creating and saving receipt as image (demo for electronic stores) in in C# demo program - "DEMOS/GetEJDocuments_DeviceGroup_C" - about device group C. -----ONLY fot device group C----- *** Fiscal device and payment terminal can connect via bluetooth ONLY if the payment terminal is NOT low energy. *** In that case payment terminal is compatable ONLY with the following fiscal devices from C group - FP-700X , FP-700XE, WP-500X, WP-50X, DP-25X, WP-25X, DP-150X. - All commands for "working with pinpad" in device group C are added to the SDK. - There is also a demo in C# demo program - "DEMOS/SalesDemo_DeviceGroup_C". ----- -Added commands in SDK for device group A: --- config_Set_Logo(string rowNum, string data) --- info_LastErrorExtendedInfo() --- info_ClearLastErrorExtendedInfo() --- info_Get_NRADData_Type() --- config_Set_OptionalErrorText(string option, string offOn) --- config_Set_WorkingWithPinpad(string enable) --- info_Get_DataForLastIssuedReceipt() --- info_Get_DataForLastIssuedReceipt(string docNum) --- info_Get_NotPrintedRowCount(string option) --- info_Get_NRADDocuments(string option) --- info_Get_CashDrawerStatusOrOpenShift(string option) --- info_Get_LastDocumentNumber() --- info_Get_ProgrammedFuel() --- info_Get_SoldQuant_AccumulatedSales(string closure1, string option) --- info_Get_ProgrammedPumps(string closure1, string option) --- info_Get_ProgrammedPumps2(string closure1, string option) --- info_Get_ProgrammedPumps3(string closure1, string option, string pumpNum) --- config_SafetyTurnOffDevice() --- info_Pinpad_CheckLastRecordedRRN() --- info_Pinpad_DataForLastTransaction() --- info_Pinpad_PrintLastSuccesfulTransaction() --- info_Pinpad_Get_DateTime() --- config_Pinpad_Set_DateTime(string dateTime) --- info_Pinpad_TestConnection() --- info_Pinpad_NumberOfEntries() --- info_Pinpad_Information() --- config_Pinpad_SetBatchNumber(string batch)

```

--- config_Pinpad_DeleteBatchNumber()
--- config_Pinpad_DeleteFreezeTransaction()
--- config_Pinpad_SetReceiptNumber(string stan)
--- receipt_Pinpad_PrintByNumber(string number)
--- receipt_Pinpad_PrintLastSuccessfultransaction()
--- receipt_Pinpad_ReturnLastSuccessfultransaction()
--- config_Pinpad_SetRRNFromPPTToFD()
--- info_Pinpad_TestConnectionToServer()
--- report_Pinpad_EndOfDay()
--- info_Pinpad_GetUpdate()
--- report_Pinpad_Reports(string type)
--- info_Pinpad_StatusOfLastTransaction()
--- info_Pinpad_ClearTransactionStatusFlags()
--- config_Pinpad_HeaderPrintingOptions(string option)
--- config_Pinpad_HeaderPrintingOptions(string option,string numReceipts)
--- receipt_Pinpad_PrintReceiptCopyFromEJ_ByNum(string receiptNum)
--- receipt_Pinpad_PrintReceiptCopyFromEJ_ByDate(string date)
--- info_Pinpad_TransactionStatus()

```

Version 1.0.0.8 (02/2020)

Date	Description
02/2020	- Added methods for device group A for command 114, variants about storno receipts --- info_Get_StornoSums(string closure1, string option, string closure2) --- info_Get_Storno_NettoSums(string closure1, string option, string closure2) --- info_Get_Storno_VATSums(string closure1, string option, string closure2) - Also added in documentation pdf – DeviceGroup_A_Doc.

Version 1.0.0.7 (01/2020)

Date	Description
01/2020	- Changed anonymous functions: 1) bool Y(string param) { return !string.IsNullOrEmpty(param); } 2) bool N(string param) { return string.IsNullOrEmpty(param); } 3) bool Y(string param) { return !string.IsNullOrEmpty(param) && !param.Equals("0"); } 4) bool N(string param) { return string.IsNullOrEmpty(param) param.Equals("0"); } - Moved to class FiscalPrinter and implemented like global lambda functions with names and definitions as follows: 1) Name: Y Definition: Func<string,bool> Y = (param) => { return !string.IsNullOrEmpty(param); }; 2) Name: N Definition: Func<string,bool> N = (param) => { return string.IsNullOrEmpty(param); }; 3) Name: Y_0 Definition: public Func<string, bool> Y_0 = (param) => { return !string.IsNullOrEmpty(param) && !param.Equals("0"); };

	4) Name: N_0 Definition: <pre>public Func<string, bool> N_0 = (param) => { return string.IsNullOrEmpty(param) param.Equals("0"); };</pre> Usage is the same as previous versions. - Bug fixed in function Get_CashIn_CashOut in class FDUниверsal_BGR in output parameters names from answer["errorParam"] -> answer["errorCode"] for all 3 device groups A,B,C
--	---

Version 1.0.0.6 (01/2020)

Date	Description
01/2020	- Added new methods in device independent part of SDK. - Fixed some bugs - Added new methods in DEMO program for independent part – Set/Get date and time, Open invoice, X and Z reports, Cash In/Out operation, Information about receipt.

Version 1.0.0.5 (01/2020)

Date	Description
01/2020	- Added new device independent part of SDK, which combines Group A, B and C. -- Contains basic methods (Fiscal receipt and STORNO) and some useful properties. - Added DEMO program for independent part

Version 1.0.0.4 (11/2019)

Date	Description
11/2019	- Added method documentations (in DOCUMENTATION folder in the archive) as follows: -- DeviceGroup_A_Doc - For device group A -- DeviceGroup_B_Doc - For device group B -- DeviceGroup_C_Doc - For device group C

Version 1.0.0.4 (10/2019)

Date	Description
10/2019	Maintain protocol changes about fiscal devices according to ordinance N18/2018. – For device group C, removed commands: -- receipt_StornoOpen_C01 -- receipt_StornoOpen_C02 -- receipt_StornoOpen_C03 -- receipt_StornoOpen_C04 -- receipt_StornoOpen_C05 because they are repeated and useless.

- receipt_StornoOpen_C06 name changed:
THE NEW NAME is: open_StornoReceipt
the functionality is the same as previous versions of SDK.
- For device group A and B:
 - Added open_StornoReceipt command . It combines all of the storno command variants.
- Device group A:
 - Added two NEW variants for command 49.
 - receipt_Sale_TextRow1_WQuan
 - receipt_DSale_TextRow1_WQuan
- For device group B added command 84 - print barcode:
 - receipt_Print_TextBarcode
 - receipt_Print_Barcode
- For device group C:
 - changes in command receipt_Print_Barcode (removed parameter "qRCodeSize" and barcodeType'4' - QR code)
 - added NEW command receipt_Print_Barcode_QR (for barcodeType'4' - QR code)
- Timeout for device when connect is needed just for LAN, so it is only for LAN in this version.

Version 1.0.0.3 (09/2019)

Date	Description
09/2019	Beta version! Maintain protocol changes about fiscal devices according to ordinance N18/2018. – Added status bytes support – Added error messages support – Bug fixed (device_Connected property) – Demo programs demonstrated the new things – Added properties for current status state

Version 1.0.0.2 (07/2019)

Date	Description
07/2019	Beta version! Maintain protocol changes about fiscal devices according to ordinance N18/2018. – Added LAN support

Version 1.0.0.1 (07/2019)

Date	Description
	Beta version! Maintain protocol changes about fiscal devices according to ordinance N18/2018.

07/2019	Added to Device group A missing status bits for errors ---- 1.6 The built-in tax terminal is not responding. ---- 1.3 # Low battery (the real-time clock is in RESET status). ---- 1.2 # The RAM has been reset. ---- 5.5 Fiscal memory read error. ---- 5.2 * The last fiscal memory store operation is not successful. ---- 5.0 * The fiscal memory is set in READONLY mode (locked).
---------	---

Version 1.0.0.0 (07/2019)

Date	Description
07/2019	Beta version! Maintain protocol changes about fiscal devices according to ordinance N18/2018. FiscalPrinter SDK maintains all of the bellow mantioned models: ✓ FP-700 ✓ FP-800 ✓ FP-2000 ✓ FP-650 ✓ FMP-10 ✓ SK1-21F ✓ SK1-31F ✓ DP-05 ✓ DP-15 ✓ DP-25 ✓ DP-35 ✓ DP-150 ✓ WP-50 ✓ FMP-350X ✓ FMP-55X ✓ FP-700X ✓ DP-25X ✓ DP-150X ✓ DP-05C ✓ WP-500X ✓ WP-50X Device models are organized into three logical groups: Device group A: ✓ FP-700 ✓ FP-800 ✓ FP-2000 ✓ FP-650 ✓ FMP-10 ✓ SK1-21F ✓ SK1-31F Device group B: ✓ DP-05 ✓ DP-15 ✓ DP-25
07/2019	

- ✓ DP-35
- ✓ DP-150
- ✓ WP-50

Device group C:

- ✓ FMP-350X
- ✓ FMP-55X
- ✓ FP-700X
- ✓ DP-25X
- ✓ DP-150X
- ✓ DP-05C
- ✓ WP-500X
- ✓ WP-50X

Introduction

This document contains information about software development kit (SDK) “FiscalPrinterSDK”. Is a .dll file, library that manages Datecs fiscal devices for Bulgaria. The SDK allows easy usage and management under OS Windows.

“FiscalPrinterSDK” allows you to manage Datecs fiscal devices for Bulgaria through RS232 serial port communication and TCP IP communication.

Device maintenance

In the table below are all of the devices for Bulgaria which can be maintained with FiscalPrinterSDK.

Model	Type	RS-232	Logical group
FP-650	Fiscal printer	✓	A
FP-700	Fiscal printer	✓	
FP-800	Fiscal printer	✓	
FP-2000	Fiscal printer	✓	
FMP-10	Mobile fiscal printer	✓	
SK1-21F	Building-in fiscal printer	✓	
SK1-31F	Building-in fiscal printer	✓	
DP-05	Mobile cash register	✓	B
DP-15	Cash register	✓	
DP-25	Cash register	✓	
DP-35	Cash register	✓	
DP-150	Cash register	✓	
WP-50	Mobile cash register	✓	
FMP-350X	Mobile fiscal printer	✓	C
FMP-55X	Mobile fiscal printer	✓	
FP-700X	Fiscal printer	✓	
WP-500X	Cash register	✓	
WP-50X	Mobile cash register	✓	
DP-25X	Cash register	✓	
DP-150X	Cash register	✓	
DP-05C	Mobile courier cash register	✓	

Language support

FiscalPrinter SDK supports two languages: English and Bulgarian . You can choose language in a demo program and after that all of the error messages and status bits descriptions will be in the currently selected language. You can see a demonstration of this in any of the demo programs, that comes with the instalation.

Instalation

You don't need to install anything. You can just use the "FiscalPrinterSDK.dll" file which comes with the archive. Additional information how to use it can be found in section "Usage" of this document.

When you download the archive you will see that it contains four folders and one "readme" file.

- **In folder "BINARIES"** are:
 - the library(sdk) FiscalPrinterSDK.dll
 - .exe demo programs files for different logical groups A,B,C.
- **In folder "DEMOS"** you can find the source code and projects as general for all demo programs for that SDK, developed with C# programming language.
- **In folder "DOCUMENTATION"** you can find all the documentation which concerned FiscalPrinterSDK and maintained fiscal devices for Bulgaria.
- **In folder "FiscalPrinterSDK"** you will find the project and source code for the SDK.

Usage

The FiscalPrinterSDK is a **device-dependent** SDK. All of the methods are device-dependent.

Subclasses in the FiscalPrinterSDK project are as follows:

FDGROUP_A_BGR – for devices from group A, 6 status bytes, without commands error codes in output.

FDGROUP_B_BGR – for devices from group B, 6 status bytes, without commands error codes in output.

FDGROUP_C_BGR – for devices from group C, 8 status bytes, with commands error codes in output.

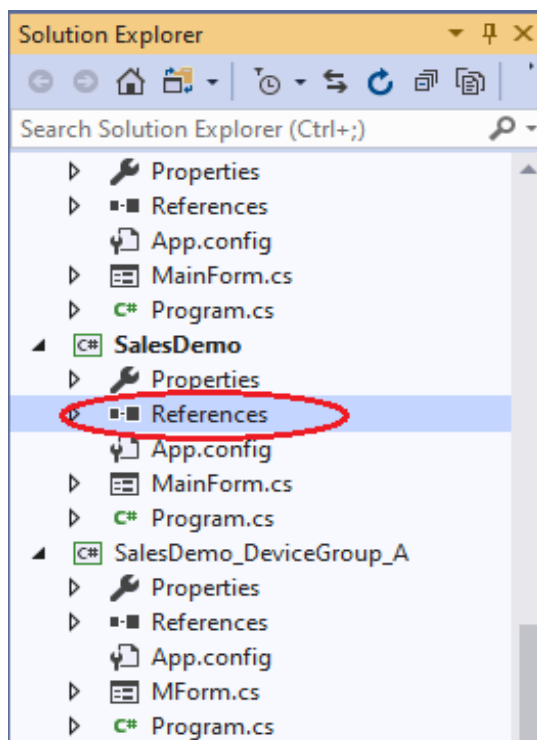
➤ **“Thread safe”?**

This SDK is **NOT** thread safe. If you execute more than one command at the same time the device may not return correct data. So keep that in mind when you develop your program. Write to us for advice or any further information.

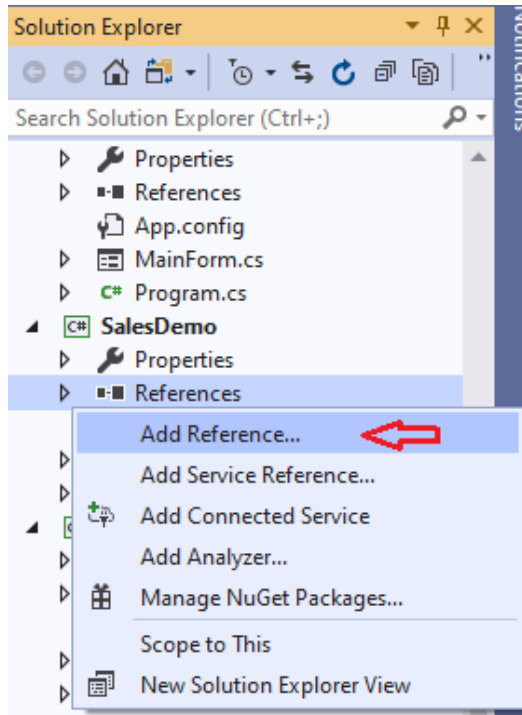
Usage of FiscalPrinterSDK library from Visual studio 2019 in C# program(demo)

It is easy!

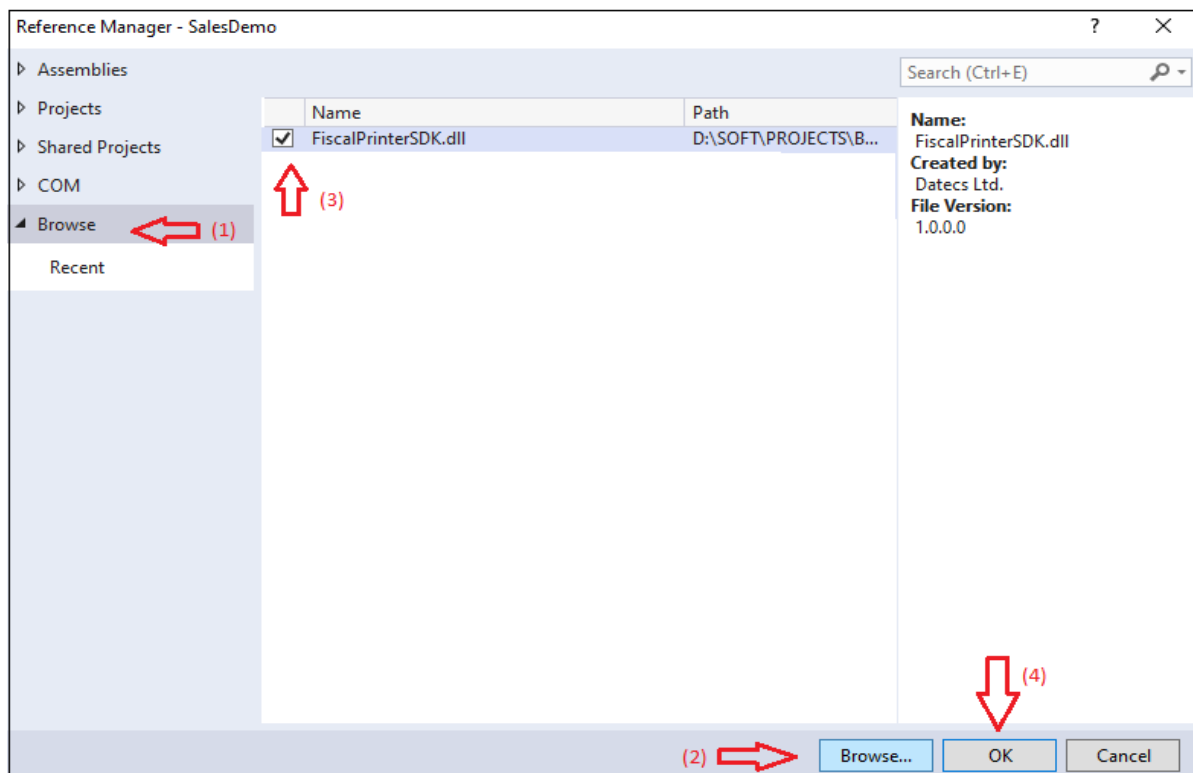
- Open a C# project in Visual studio 2019.
- Go to Solution Explorer\References;



- Click the right button on the mouse and choose “Add Reference”;



- In the dialog window - go to section “Browse” (1), click the button “Browse” (2). In File Explorer open the SDK archive folder and search for “FiscalPrinterSDK.dll“. Then mark it and click the button “Add”. It will show in the dialog window, click on the checkbox to mark it (3) then click the button “OK”(4);



- This is it! Visual Studio will add the interface and you can start using it right away.

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.IO.Ports;
11 using FiscalPrinterSDK;
12
13 namespace GetEJDocuments
14 {
15     3 references
16     public partial class fMain : Form
17     {
18         private int docType = 0;
19         private bool invalidCharacter1;
20         private bool StopReading = false;
21         private FDGROUP_C_BGR fiscal;
22     }
23     1 reference
24     public fMain()
25     {
26         InitializeComponent();
27     }
28 }
12 references
```

- You can find DEMO programs (with source code made with C# programming language on Visual Studio 2019) in the archive folder -> subfolder “DEMOS”.

Eumerated types

Tlanguage

Possible values:

- English
- Bulgarian

FiscalPrinterProtocol - protocol type, depends from packet message type from the device.

Possible values:

- Legacy
- Extended

Properties

- `device_Connected` – return true if the device is connected and false – otherwise.

Fiscal device status

Error status bits

Depending on device models and device packet message, current state is determined by answer and/or status bits current state. Current product provides read only properties, which return current state of particular status bit, raised in state “1”. These flags are raised by the fiscal device in case of an error or in case of a failure(damage) in the device.

You can use this properties in your programs:

- After successful connection with the fiscal device
- After method, command or script implementation

Device group A

Property name	Property type	Status [byte,bit]	Human meaning
eSBit_GeneralError_Sharp	Boolean Read only	[0,5]	General error - this is OR of all errors marked with #
eSBit_PrintingMechanism	Boolean Read only	[0,4]	# Failure in printing mechanism.
eSBit_ClockIsNotSynchronized	Boolean Read only	[0,2]	The real time clock is not synchronized.
eSBit_CommandCodeIsInvalid	Boolean Read only	[0,1]	# Command code is invalid.
eSBit_SyntaxError	Boolean Read only	[0,0]	# Syntax error.
eSBit_BuildInTaxTerminalNot Responding	Boolean Read only	[1, 6]	The built-in tax terminal is not responding.
eSBit_LowBattery	Boolean		# Low battery (the real-time

	Read only	[1, 3]	clock is in RESET status).
eSBit_RamReset	Boolean Read only	[1, 2]	The RAM has been reset.
eSBit_CommandNotPermitted	Boolean Read only	[1, 1]	# The command is not allowed in the current fiscal mode.
eSBit_Overflow	Boolean Read only	[1, 0]	# Overflow during command execution. Status 1.1 will also be set and the command will not change any data in the printer.
eSBit_EJIsFull	Boolean Read only	[2,2]	EJ is full (less than 1 MB free in the EJ).
eSBit_EndOfPaper	Boolean Read only	[2,0]	# No paper. If this flag is raised during a print command, the command is cancelled and does not change the printer status.
eSBit_FM_NotAccess	Boolean Read only	[4,0]	* Error when trying to access data stored in the FM.
eSBit_FM_Full	Boolean Read only	[4,4]	*Fiscal memory is full.
eSBit_GeneralError_Star	Boolean Read only	[4,5]	OR of all errors marked with “*” from Bytes 4 and 5
eSBit_FM_ReadError	Boolean Read only	[5,5]	* Fiscal memory read error
eSBit_LastFMOperation_NotSuccessful	Boolean Read only	[5,2]	* The last fiscal memory store operation is not successful.
eSBit_FM_ReadOnly	Boolean Read only	[5,0]	* The fiscal memory is set in READONLY mode (locked).

Device group B

Property name	Property type	Status [byte, bit]	Human meaning
eSBit_GeneralError_Sharp	Boolean Read only	[0,5]	General error - this is OR of all errors marked with #
eSBit_ClockIsNotSynchronized	Boolean		The real time clock is not

	Read only	[0,2]	synchronized.
eSBit_CommandCodeIsInvalid	Boolean Read only	[0,1]	# Command code is invalid.
eSBit_SyntaxError	Boolean Read only	[0,0]	# Syntax error.
eSBit_BuildInTaxTerminalNot Responding	Boolean Read only	[1, 6]	The built-in tax terminal is not responding.
eSBit_CommandNotPermitted	Boolean Read only	[1, 1]	# The command is not allowed in the current fiscal mode.
eSBit_Overflow	Boolean Read only	[1, 0]	# Overflow during command execution. Status 1.1 will also be set and the command will not change any data in the printer.
eSBit_EJIsFull	Boolean Read only	[2,2]	EJ is full (less than 1 MB free in the EJ).
eSBit_EndOfPaper	Boolean Read only	[2,0]	# No paper. If this flag is raised during a print command, the command is cancelled and does not change the printer status.
eSBit_FM_NotAccess	Boolean Read only	[4,0]	* Error when trying to access data stored in the FM.
eSBit_FM_Full	Boolean Read only	[4,4]	*Fiscal memory is full.
eSBit_GeneralError_Star	Boolean Read only	[4,5]	OR of all errors marked with '*' from Bytes 4 and 5

Device group C

Property name	Property type	Status [byte, bit]	Human meaning
eSBit_GeneralError_Sharp	Boolean Read only	[0,5]	General error - this is OR of all errors marked with #
eSBit_PrintingMechanism	Boolean Read only	[0,4]	# Failure in printing mechanism.
eSBit_ClockIsNotSynchronized	Boolean Read only	[0,2]	The real time clock is not synchronized.
eSBit_CommandCodeIsInvalid	Boolean Read only	[0,1]	# Command code is invalid.
eSBit_SyntaxError	Boolean Read only	[0,0]	# Syntax error.
eSBit_CommandNotPermitted	Boolean Read only	[1, 1]	# The command is not allowed in the current fiscal mode.
eSBit_Overflow	Boolean Read only	[1, 0]	# Overflow during command execution. Status 1.1 will also be set and the command will not change any data in the printer.
eSBit_EJIsFull	Boolean Read only	[2,2]	EJ is full (less than 1 MB free in the EJ).
eSBit_EndOfPaper	Boolean Read only	[2,0]	# No paper. If this flag is raised during a print command, the command is cancelled and does not change the printer status.
eSBit_FM_NotAccess	Boolean Read only	[4,0]	* Error when trying to access data stored in the FM.
eSBit_FM_Full	Boolean Read	[4,4]	*Fiscal memory is full.

	only		
eSBit_GeneralError_Star	Boolean Read only	[4,5]	OR of all errors marked with ‘*’ from Bytes 4 and 5
eSBit_FM_NotFound	Boolean Read only	[4,6]	Fiscal memory not found or damaged.

Fiscal device status

Informative status bits

Depending on device models and device packet message, current state is determined by answer and/or status bits current state. Current product provides read only properties, which return current state of particular status bit, raised in state “1”. These flags are informative, raised by the fiscal device to provide useful information.

You can use this properties in your programs:

- After successful connection with the fiscal device
- After method, command or script implementation

Device group A

Property name	Property type	Status [byte, bit]	Human meaning
iSBit_Cover_IsOpen	Boolean Read only	[0,6]	Cover is open. (Error in device models: FP-700 , FP-2000 ,FP-800 ,FP-650,FMP-10)
iSBit_No_ClientDisplay	Boolean Read only	[0,3]	No client display installed
iSBit_NonFiscal_90Degree	Boolean Read only	[1,5]	A service receipt with 90-degree rotated text printing is open

iSBit_Receipt_Storno	Boolean Read only	[1,4]	Storno receipt is open
iSBit_EJ_VeryNearEnd	Boolean Read only	[2,6]	The end of the EJ is very near (only certain receipts are allowed)
iSBit_Receipt_Nonfiscal	Boolean Read only	[2,5]	Non-fiscal receipt is open
iSBit_EJ_NearlyFull	Boolean Read only	[2,4]	EJ nearly full
iSBit_Receipt_Fiscal	Boolean Read only	[2,3]	Fiscal receipt is open
iSBit_Near_PaperEnd	Boolean Read only	[2,1]	Near paper end
iSBit_Sw7_Status	Boolean Read only	[3,6]	Sw7 status
iSBit_Sw6_Status	Boolean Read only	[3,5]	Sw6 status
iSBit_Sw5_Status	Boolean Read only	[3,4]	Sw5 status
iSBit_Sw4_Status	Boolean Read only	[3,3]	Sw4 status
iSBit_Sw3_Status	Boolean Read only	[3,2]	Sw3 status
iSBit_Sw2_Status	Boolean Read only	[3,1]	Sw2 status
iSBit_Sw1_Status	Boolean Read only	[3,0]	Sw1 status
iSBit_PrintingHead_Overheated	Boolean Read only	[4,6]	The printing head is overheated.

iSBit_LessThan_50_Report	Boolean Read only	[4,3]	There is space for less then 50 reports in Fiscal memory
iSBit_Number_SFM_Set	Boolean Read only	[4,2]	Serial number and number of FM are set
iSBit_Number_Tax_Set	Boolean Read only	[4,1]	Tax number is set
iSBit_VAT_Set	Boolean Read only	[5,4]	VAT are set at least once.
iSBit_Device_Fiscalized	Boolean Read only	[5,3]	Device is fiscalized
iSBit_FM_formatted	Boolean Read only	[5,1]	FM is formatted

Device group B

Property name	Property type	Status [byte, bit]	Human meaning
iSBit_No_ClientDisplay	Boolean Read only	[0,3]	No client display installed
iSBit_TimeoutWithUnsentDocs	Boolean Read only	[1,5]	There are unsent documents to the NRA server
iSBit_Receipt_Nonfiscal	Boolean Read only	[2,5]	Non-fiscal receipt is open
iSBit_EJ_NearlyFull	Boolean Read only	[2,4]	EJ nearly full
iSBit_Receipt_Fiscal	Boolean Read only	[2,3]	Fiscal receipt is open

iSBit_LessThan_50_Report	Boolean Read only	[4,3]	There is space for less then 50 reports in Fiscal memory
iSBit_Number_SFM_Set	Boolean Read only	[4,2]	Serial number and number of FM are set
iSBit_Number_Tax_Set	Boolean Read only	[4,1]	Tax number is set
iSBit_VAT_Set	Boolean Read only	[5,4]	VAT are set at least once.
iSBit_Device_Fiscalized	Boolean Read only	[5,3]	Device is fiscalized
iSBit_FM_formatted	Boolean Read only	[5,1]	FM is formatted

Device group C

Property name	Property type	Status [byte, bit]	Human meaning
iSBit_Cover_IsOpen	Boolean Read only	[0,6]	Cover is open. (Error in device models: FMP-350X, FMP-55X, FP-700X, WP-500X, WP-50X, DP-05C)
iSBit_Receipt_Nonfiscal	Boolean Read only	[2,5]	Non-fiscal receipt is open
iSBit_EJ_NearlyFull	Boolean Read only	[2,4]	EJ nearly full
iSBit_Receipt_Fiscal	Boolean Read only	[2,3]	Fiscal receipt is open
iSBit_Near_PaperEnd	Boolean Read	[2,1]	Near paper end

	only		
iSBit_LessThan_60_Report	Boolean Read only	[4,3]	There is space for less then 60 reports in Fiscal memory
iSBit_Number_SFM_Set	Boolean Read only	[4,2]	Serial number and number of FM are set
iSBit_Number_Tax_Set	Boolean Read only	[4,1]	Tax number is set
iSBit_VAT_Set	Boolean Read only	[5,4]	VAT are set at least once.
iSBit_Device_Fiscalized	Boolean Read only	[5,3]	Device is fiscalized
iSBit_FM_formatted	Boolean Read only	[5,1]	FM is formatted

Status Bits for all logical device groups

Color	Description
Light red	Shows that the status bit is error if it is 1
Light yellow	For errors bits just for some device models
Gray	For informative status bits
Light gold	For warning status bits

Status byte 0

Byte	Bit	Description in logical group A	Description in logical group B	Description in logical group C
0	7	Reserved: always 1	Reserved: always 1	Reserved: always 1
	6	The printer cover is open.(*)	Reserved: always 0	The printer cover is open.(**)
	5	General error: an OR of all errors marked with the '#' character	General error: an OR of all errors marked with the '#' character	General error: an OR of all errors marked with the '#' character
	4	(#)Failure in printing mechanism	Reserved: always 0	(#) Failure in printing mechanism

	3	Customer display is not installed	Customer display is not installed	Reserved: always 0
	2	The clock is not set	The clock is not set	The clock is not set
	1	(#)Invalid command code received	(#)Invalid command code received	(#)Invalid command code received
	0	(#)Incoming data has syntax error	(#)Incoming data has syntax error	(#)Incoming data has syntax error

(*) “The printer cover is open” is an error for models: FP-700,FP-2000,FP-800 ,FP-50, FMP-10 In device group A and it is NOT for: SK1-21F / SK1-31F

(**) “The printer cover is open” is an error for models: FMP-350X, FMP-55X,FP-700X,WP-500X, WP-50X, DP-05C in device group C and it is NOT for: DP-25X, DP-150X

Status byte 1

Byte	Bit	Description in logical group A	Description in logical group B	Description in logical group C
1	7	Reserved: always 1	Reserved: always 1	Reserved: always 1
	6	The built-in tax terminal is not responding	The built-in tax terminal is not responding	Reserved: always 0
	5	A service receipt with 90-degree rotated text printing is open	There are unsent documents to the NRA server	Reserved: always 0
	4	A refund (storno) receipt is open	Reserved: always 0	Reserved: always 0
	3	#Low battery (the real-time clock is in RESET status).	Reserved: always 0	Reserved: always 0
	2	#The RAM has been reset.	Reserved: always 0	Reserved: always 0
	1	(#)Command is not permitted	(#)Command is not permitted	(#)Command is not permitted
	0	(#)Overflow during command execution	(#)Overflow during command execution	(#)Overflow during command execution

Status byte 2

Byte	Bit	Description in logical group A	Description in logical group B	Description in logical group C
2	7	Reserved: always 1	Reserved: always 1	Reserved: always 1
	6	Near EJ end (only certain receipts are allowed)	Not used	Reserved: always 0
	5	Nonfiscal receipt is open	Nonfiscal receipt is open	Nonfiscal receipt is open
	4	EJ nearly full	EJ nearly full	EJ nearly full
	3	Fiscal receipt is open	Fiscal receipt is open	Fiscal receipt is open
	2	EJ is full	EJ is full	EJ is full
	1	Near paper end	Reserved: always 0	Near paper end
	0	(#) End of paper	(#) End of paper	(#) End of paper

Status byte 3

Byte	Bit	Description in logical group A	Description in logical group B	Description in logical group C
3	7	Reserved: always 1	Reserved: always 1	Reserved: always 1
	6	Sw7 status	Reserved: always 0	Reserved: always 0
	5	Sw6 status	Reserved: always 0	Reserved: always 0
	4	Sw5 status	Reserved: always 0	Reserved: always 0
	3	Sw4 status	Reserved: always 0	Reserved: always 0
	2	Sw3 status	Reserved: always 0	Reserved: always 0
	1	Sw2 status	Reserved: always 0	Reserved: always 0
	0	Sw1 status	Reserved: always 0	Reserved: always 0

Status byte 4

Byte	Bit	Description in logical group A	Description in logical group B	Description in logical group C
4	7	Reserved: always 1	Reserved: always 1	Reserved: always 1
	6	The printing head is overheated	Reserved: always 0	Fiscal memory is not found or damaged
	5	OR of all errors marked with the '*' character from bytes 4 and 5	OR of all errors marked with the '*' character from bytes 4 and 5	OR of all errors marked with the '*' character from bytes 4 and 5
	4	(*) The fiscal memory is full	(*) The fiscal memory is full	(*) The fiscal memory is full
	3	There is space for less than 50 records remaining in the FP	There is space for less than 50 records remaining in the FP	There is space for less than 60 records remaining in the FP
	2	Serial number and number of FM are set	Serial number and number of FM are set	Serial number and number of FM are set
	1	Tax number is set	Tax number is set	Tax number is set
	0	(*)Error when trying to access data stored in the FM	(*)Error when trying to access data stored in the FM	(*)Error when trying to access data stored in the FM

Status byte 5

Byte	Bit	Description in logical group A	Description in logical group B	Description in logical group C
5	7	Reserved: always 1	Reserved: always 1	Reserved: always 1
	6	Not used	Reserved: always 0	Reserved: always 0
	5	Fiscal memory read error	Reserved: always 0	Reserved: always 0
	4	VAT are set at least once	VAT are set at least once	VAT are set at least once
	3	Device is fiscalized	Device is fiscalized	Device is fiscalized
	2	(*)The last fiscal memory store operation is not successful	Reserved: always 0	Reserved: always 0
	1	FM is formatted	FM is formatted	FM is formatted
	0	(*)The fiscal memory is set in READONLY mode (locked)	Reserved: always 0	Reserved: always 0

Status byte 6

Byte	Bit	Description in logical group A	Description in logical group B	Description in logical group C
6	7	Reserved: always 1	Reserved: always 1	Reserved: always 1
	6	x	x	Reserved: always 0
	5	x	x	Reserved: always 0
	4	x	x	Reserved: always 0
	3	x	x	Reserved: always 0
	2	x	x	Reserved: always 0
	1	x	x	Reserved: always 0
	0	x	x	Reserved: always 0

Status byte is saved for future use.

Status byte 7

Byte	Bit	Description in logical group A	Description in logical group B	Description in logical group C
7	7	Reserved: always 1	Reserved: always 1	Reserved: always 1
	6	x	x	Reserved: always 0
	5	x	x	Reserved: always 0
	4	x	x	Reserved: always 0
	3	x	x	Reserved: always 0
	2	x	x	Reserved: always 0
	1	x	x	Reserved: always 0
	0	x	x	Reserved: always 0

Status byte is saved for future use.

Communication

➤ Through Serial communication via RS232

In common class “FiscalCommWin” in constructor with parameters:
COM port and Baudrate

In your program initialise variable type “FDGROUP_A_BGR” or
“FDGROUP_B_BGR” or “FDGROUP_C_BGR” for the device group that you use.

Example:

```
private FGROUP_C_BGR fiscal;
```

To initialise and open connection with the device use this:

Example:

```
FiscalCommWin fiscalComm = new FiscalCommWin((string)cbPort.SelectedItem,  
int.Parse((string)cbBaudRate.SelectedItem));  
fiscal = new FGROUP_C_BGR(fiscalComm);  
fiscal.Connect();
```

To close connection with the device use this:
fiscal.Disconnect();

➤ Through TCP/IP communication via LAN/Internet

In common class “FiscalCommWinSocket” in constructor with parameters:
ip Address and tcp_Port

In your program initialise variable type “FDGROUP_A_BGR” or
“FDGROUP_B_BGR” or “FDGROUP_C_BGR” for the device group that you use.

Example:

```
private FGROUP_C_BGR fiscal;
```

To initialise and open connection with the device use this:

Example:

```
FiscalCommWinSocket fiscalComm = new  
FiscalCommWinSocket(txtIPAddress.Text, int.Parse(txtPort.Text));  
fiscal = new FGROUP_C_BGR(fiscalComm);  
fiscal.Connect();
```

To close connection with the device use this:
fiscal.Disconnect();

Methods

General:

CustomCommand - Which execute command from the device with input parameters: (integer) command number and (string) input data.

Get fiscal device status bits through FiscalPrinterSDK methods

- Method „Get_Sbit_State“

With this method you can get current state of the particular status bit that you enter (with parameters of the method – byteIndex, bitIndex).

Status bits states are updated inner in the device after every command execution. For example, if you try to execute command for open a receipt with incorrect input data format – the “Syntax error” status bit will be raised after processing the device answer by the library. You can read it with this method many times and (every time before execution of any other command) status bit “Syntax error” will be raised with value “1”(Method „Get_Sbit_State“ returns **true/false** respectively for state **1/0**). After that, if you execute valid command (with correct input data, ex. Command 74), Current state of this status bit “Syntax error” will return **false**, i.e. there is no syntax error in last executed command.

- Method “Get_Sbit_Description”

With this method you can get text description of the particular status bit that you enter (with parameters of the method – byteIndex, bitIndex). Current language depends on your setting. (Check demo programs)

- Method “Get_Sbit_ErrorChecking”

With this method you can get library behaviour about the current status bit that you choose. If the method returns **true** – this means that according to the library this bit is an error.

- Method “Set_Sbit_ErrorChecking”

With this method you can set the library behaviour about any status bit. For example, if you want you program not to work in case of missing customer display – you can set library behaviour about the status bit “Customer display is not installed”. Set value **true** for this status bit (with method “Set_Sbit_ErrorChecking”) and the library

will start returning error in status bits when execution any of the methods, if customer display is not installed.

Error codes and messages

In device group C (models from this group) the device returns error code after execution of every command. Library processes this code in exception if the error code is not equal to 0 (then the command is executed correctly) in class "FiscalException". If you want to get the error message for particular error code you can use:

- Method "GetErrorMessage" - returns the error message (in previously set language).

In device group A and device group B (models from these groups) the device returns error parameter P/F (**P**ass or **f**ailed execution of a command) after execution of some commands. You should process this in your programs. (Check demo programs)

Disclaimer

The information in this document can be revised without prior announcement at any time and will not be a matter for Datecs Ltd. Datecs Ltd. is not liable for any incomplete or inaccurate information.

However, we will make sure to provide up-to-date information for products, described in this document.

How to send bug reports or problem?

Please, send us an email with the following information:

- SDK version
- Information about Windows and the system you use
- Information about the fiscal device model and if it is possible diagnostic information from the device
- Describe your problem step by step
- Send us log file if it is possible
- Also status bytes states, when you discover your problem, before execution of any other command (because this can cause problem dismiss).
- If the device return error code, send us its value.
- Information about input and output data of the command that you have executed.
- If you send us source code – please, **DO NOT** send whole project source! We do not offer this type of support!
- The support is carried out **ONLY** through email!

Contacts

We will be glad to help.
Send your emails at
dobrin@datecs.bg